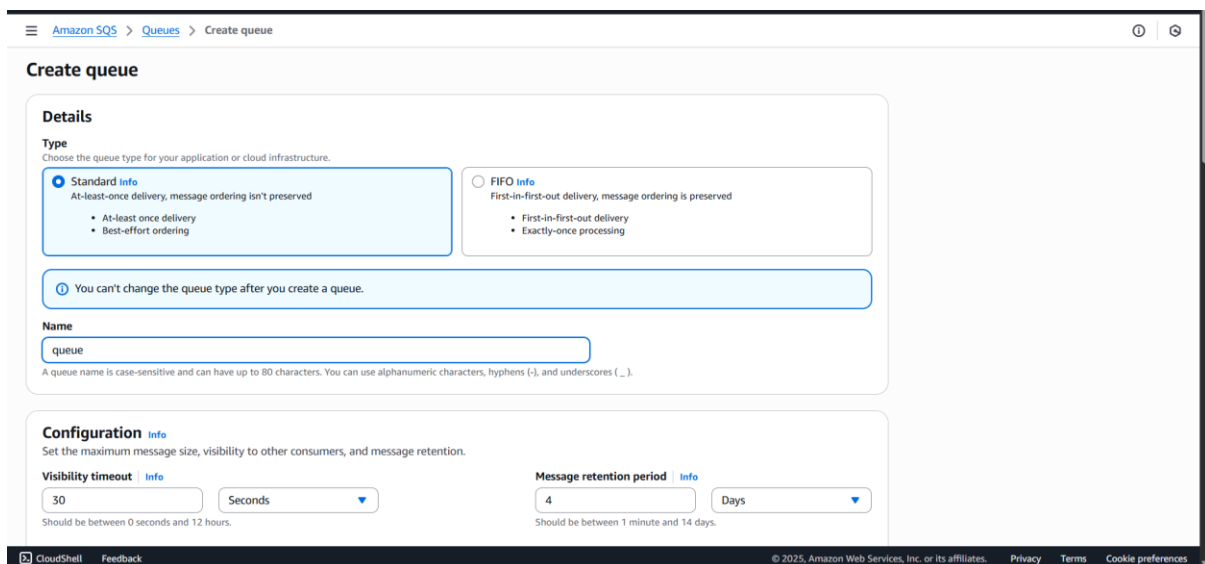


# SIMPLE QUEUE SERVICE

Amazon SQS (Simple Queue Service) is a fully managed message queuing service that enables decoupling and reliable communication between distributed systems and microservices. It allows applications to send, store, and receive messages asynchronously, improving scalability, fault tolerance, and overall system performance. By managing the messaging infrastructure, SQS helps simplify integration and ensures that messages are not lost even if one part of the system fails.

Go to the Amazon SQS Console, click "Create queue", select "Standard" as the queue type, enter a name for your queue (e.g., MyStandardQueue), under Configuration leave the Visibility timeout as default (usually 30 seconds), Delivery delay as default (0 seconds), and Message retention period as default (4 days), leave all other settings as default, and finally click "Create queue"



The screenshot displays the 'Create queue' interface in the Amazon SQS console. The breadcrumb navigation at the top reads 'Amazon SQS > Queues > Create queue'. The main heading is 'Create queue'. Below this, the 'Details' section is active, showing two queue type options: 'Standard' (selected) and 'FIFO'. The 'Standard' option is described as 'At-least-once delivery, message ordering isn't preserved' with sub-points 'At-least once delivery' and 'Best-effort ordering'. The 'FIFO' option is described as 'First-in-first-out delivery, message ordering is preserved' with sub-points 'First-in-first-out delivery' and 'Exactly-once processing'. A warning message states: 'You can't change the queue type after you create a queue.' Below this, the 'Name' field contains the text 'queue'. A note specifies: 'A queue name is case-sensitive and can have up to 80 characters. You can use alphanumeric characters, hyphens (-), and underscores (\_).' The 'Configuration' section is also visible, with the instruction 'Set the maximum message size, visibility to other consumers, and message retention.' It includes two settings: 'Visibility timeout' set to '30' seconds and 'Message retention period' set to '4' days. Both settings have links to 'info' and range constraints (0-12 hours for visibility, 1 minute-14 days for retention). The footer of the console shows 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. or its affiliates, along with links for 'Privacy', 'Terms', and 'Cookie preferences'.

Should be between 0 and 20 seconds.

**Encryption** [Info](#)  
Amazon SQS provides in-transit encryption by default. To add at-rest encryption to your queue, enable server-side encryption.

**Server-side encryption**

☐ Disabled

☒ Enabled

**Encryption key type**

☒ Amazon SQS key (SSE-SQS)  
An encryption key that Amazon SQS creates, manages, and uses for you.

☐ AWS Key Management Service key (SSE-KMS)  
An encryption key protected by AWS Key Management Service (AWS KMS).

**Access policy** [Info](#)  
Define who can access your queue.

**Choose method**

☒ Basic  
Use simple criteria to define a basic access policy.

☐ Advanced  
Use a JSON object to define an advanced access policy.

**Define who can send messages to the queue**

☒ Only the queue owner  
Only the owner of the queue can send messages to the queue.

☐ Only the specified AWS accounts, IAM users and roles

**JSON (read-only)**

```
{
  "Version": "2012-10-17",
  "Id": "_default_policy_id",

```

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

**Redrive allow policy** - [Optional](#) [Info](#)  
Identify which source queues can use this queue as the dead-letter queue.

**Select which source queues can use this queue as the dead-letter queue.**

☒ Disabled

☐ Enabled

**Dead-letter queue** - [Optional](#) [Info](#)  
Send undeliverable messages to a dead-letter queue.

**Set this queue to receive undeliverable messages.**

☒ Disabled

☐ Enabled

**Tags** - [Optional](#) [Info](#)  
A tag is a label assigned to an AWS resource. Use tags to search and filter your resources or track your AWS costs.

**Key**

**Value - optional**

Remove

Add new tag

You can add 49 more tags.

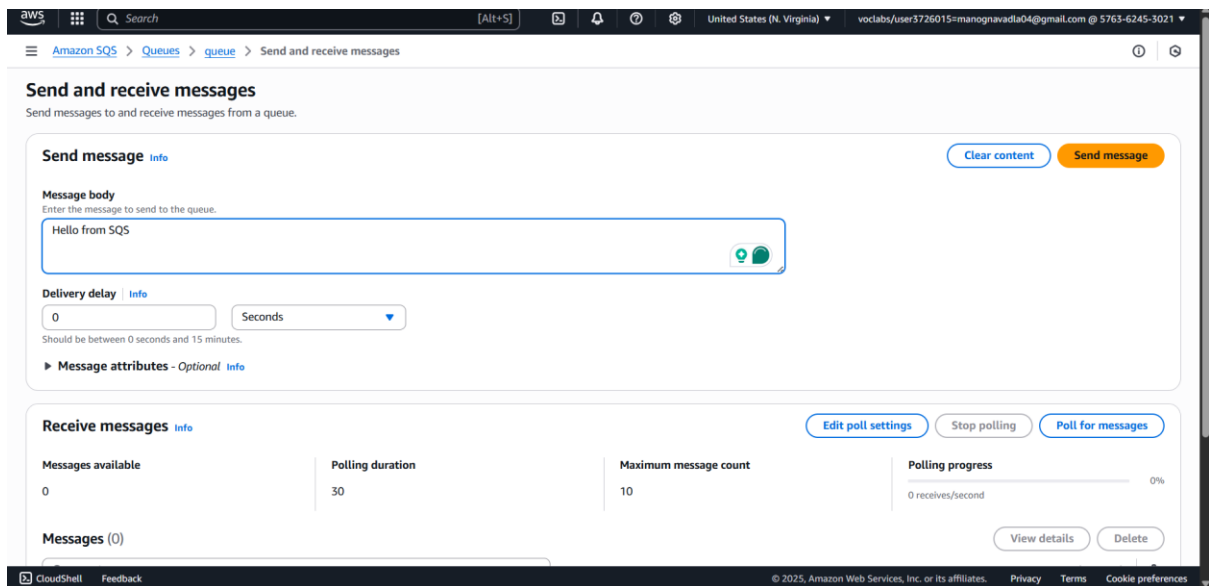
CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Go to the “Send and receive messages” tab.

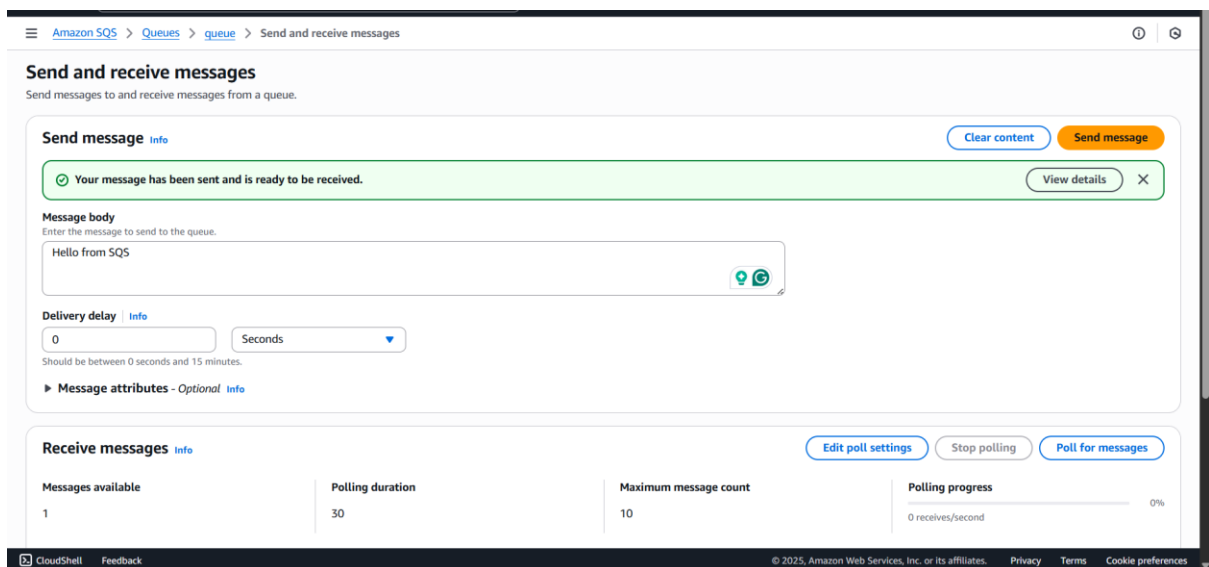
Under “Send a message”:

Clear the content box, type a message (e.g., "Hello from SQS").

Click “Send message”



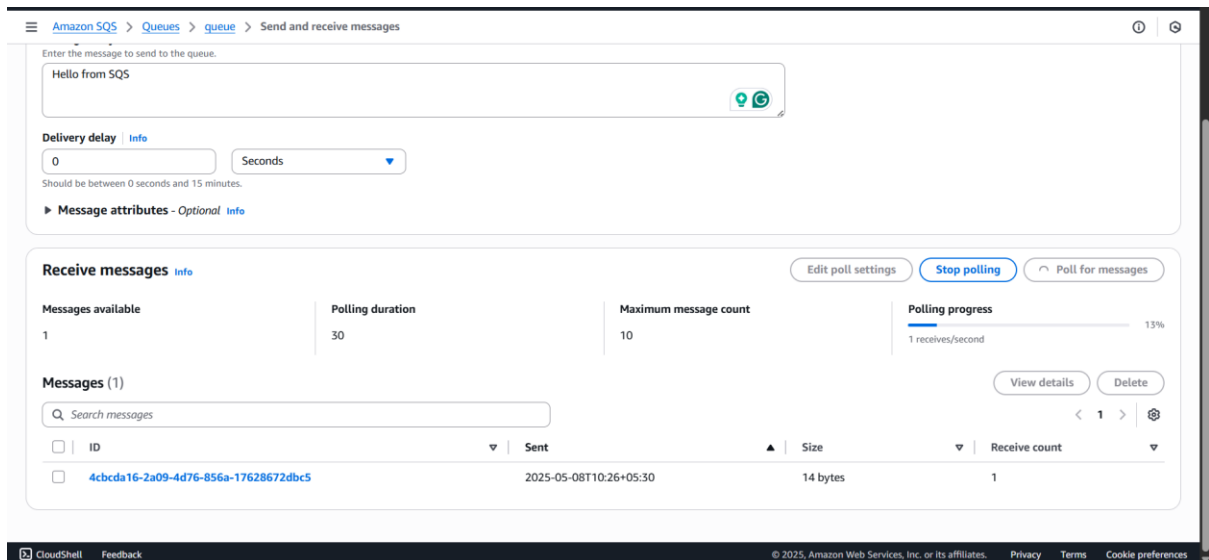
The message has been successfully sent



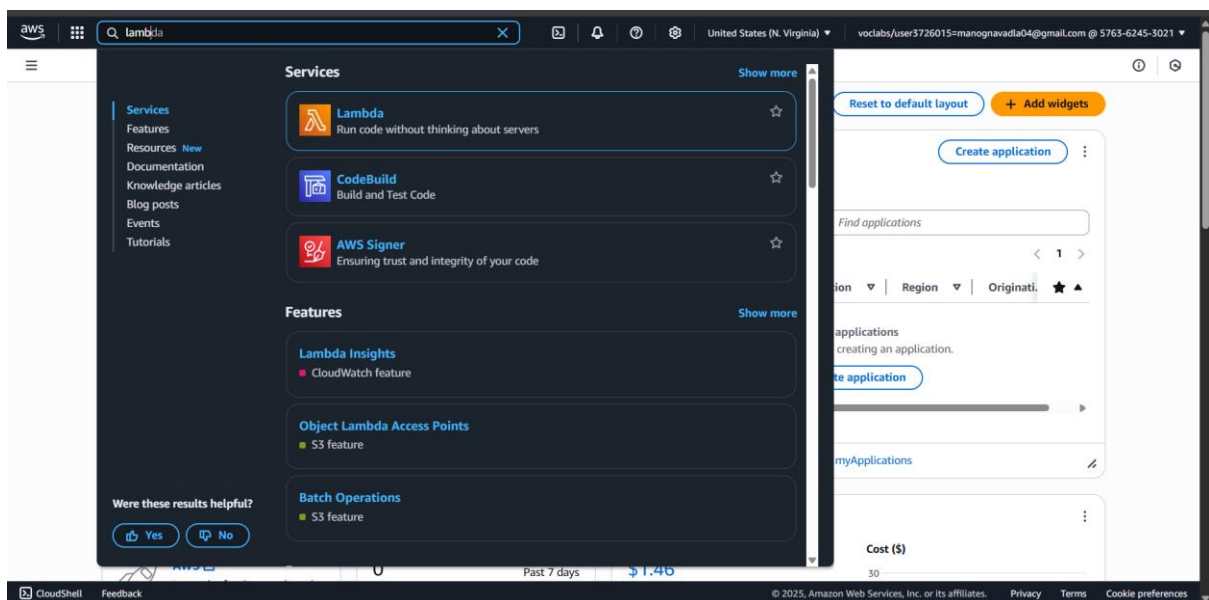
Scroll down to “Receive messages”:

Click “Poll for messages”.

You should see your message appear in the list.



Search for Lambda service and click on create function



Give the name for the lambda function.

Click create with blueprint Execution role :

Use existing role- labRole

Lambda > Functions > Create function

Create function

Choose one of the following options to create your function.

☐ Author from scratch  
Start with a simple Hello World example.

☒ Use a blueprint  
Build a Lambda application from sample code and configuration presets for common use cases.

☐ Container image  
Select a container image to deploy for your function.

Basic information

Blueprint name

Process messages in an SQS queue  
An Amazon SQS trigger that logs messages in a queue.

nodejs18.x

Function name

myFunctionName

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (\_).

Runtime

nodejs18.x

Architecture

x86\_64

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.

☐ Create a new role with basic Lambda permissions

☐ Use an existing role

☒ Create a new role from AWS policy templates

CloudShell

Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws

Search

[Alt+S]

United States (N. Virginia)

voclabs/user3726015=manognavadta04@gmail.com @ 5763-6245-3021

Lambda > Functions > Create function

Create function

Choose one of the following options to create your function.

☒ Use an existing role

☐ Create a new role from AWS policy templates

Existing role

Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

LabRole

View the LabRole role on the IAM console.

Lambda function code

Code is preconfigured by the chosen blueprint. You can configure it after you create the function. Learn more about deploying Lambda functions.

```
1 console.log('Loading function');
2
3 export const handler = async (event) => {
4   //console.log('Received event:', JSON.stringify(event, null, 2));
5   for (const { messageId, body } of event.Records) {
6     console.log('SQS message %s: %j', messageId, body);
7   }
8   return `Successfully processed ${event.Records.length} messages.`;
9 };
10
```

1:1 JavaScript

CloudShell

Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

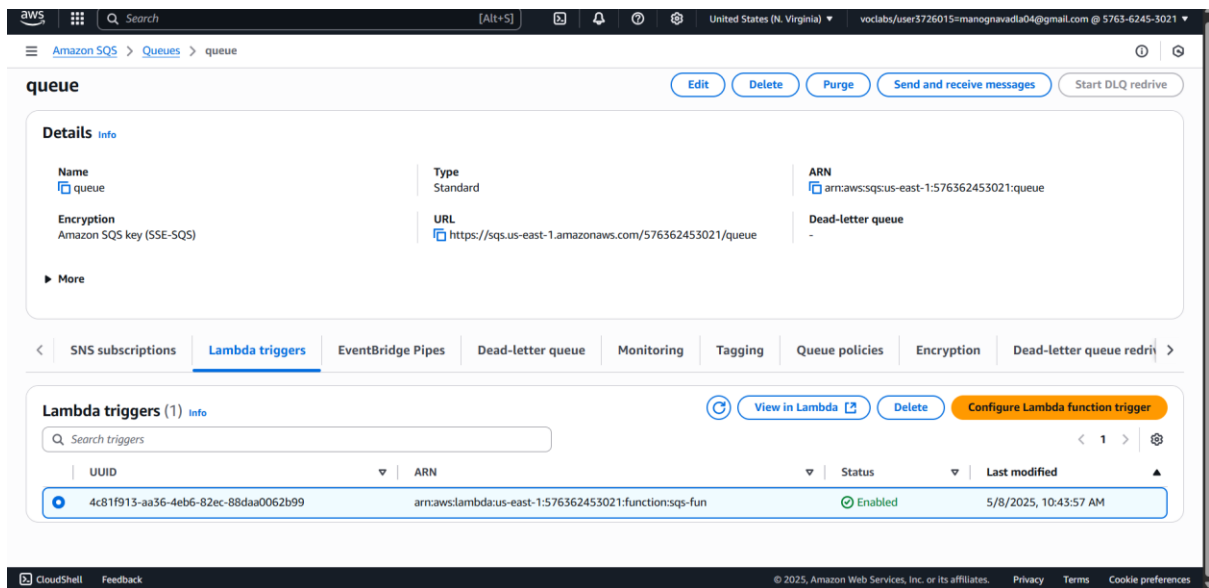
## Select Triggers

The screenshot shows the 'Create function' page in the AWS Lambda console. The 'SQS trigger' section is active, showing a dropdown menu with 'SQS' selected. Below it, the 'SQS queue' field contains the ARN 'arn:aws:sqs:us-east-1:576362453021:queue'. The 'Event poller configuration' section has 'Activate trigger' checked, 'Enable metrics' unchecked, and 'Batch size - optional' set to 10. The 'Batch window - optional' is set to 0. A 'Remove' button is in the top right corner of the trigger section.

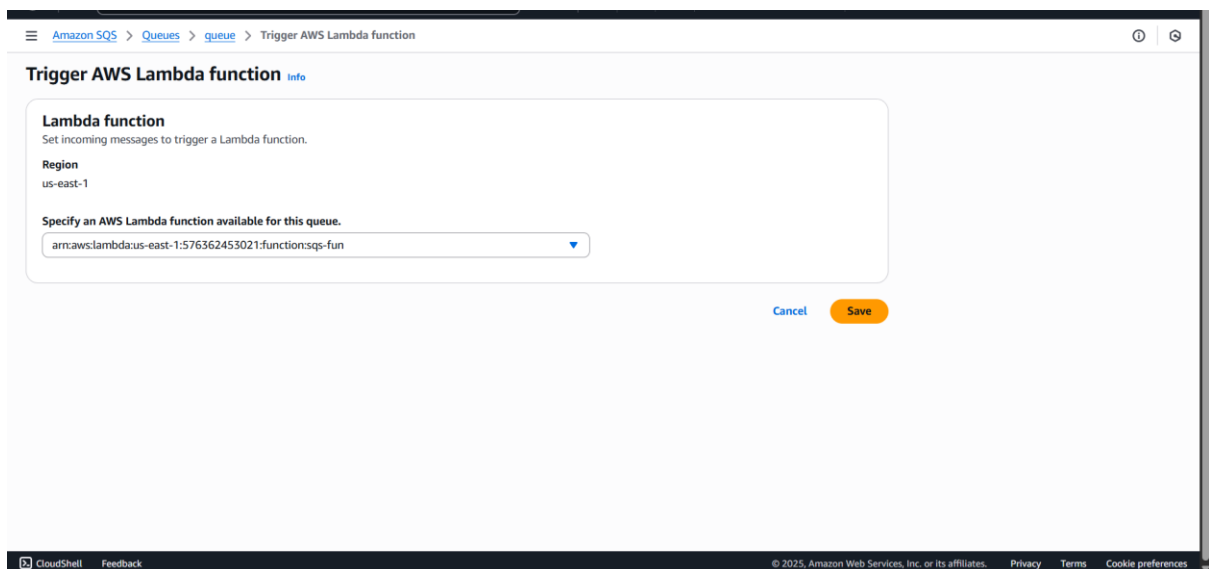
## Created a Lambda function

The screenshot shows the 'sqs-fun' function overview page in the AWS Lambda console. A green notification bar at the top states: 'Congratulations! Your Lambda function "sqs-fun" has been successfully created and configured with queue as a trigger. Choose Test to input a test event and test your function.' The 'Function overview' section shows a diagram with 'sqs-fun' and 'SQS' connected. The 'Description' field contains 'An Amazon SQS trigger that logs messages in a queue.' The 'Last modified' field shows '3 seconds ago'. The 'Function ARN' is 'arn:aws:lambda:us-east-1:576362453021:function:sqs-fun'. The 'Function URL' is '-'. The 'Code source' section is visible at the bottom.

Go to the SQS queue and navigate to Lambda Triggers, you will be able to see the function you created



Click on configure lambda trigger



Create an event and test the lambda function

Navigation: Lambda > Functions > sqs-fun

### sqs-fun

Buttons: Throttle, Copy ARN, Actions, Export to Infrastructure Composer, Download

**Function overview**  
**Diagram** | Template  
Diagram showing sqs-fun triggered by SQS.  
+ Add trigger  
+ Add destination

**Description**  
An Amazon SQS trigger that logs messages in a queue.  
**Last modified**  
10 minutes ago  
**Function ARN**  
arn:aws:lambda:us-east-1:576362453021:function:sqs-fun  
**Function URL**  
Info

Code | **Test** | Monitor | Configuration | Aliases | Versions

**Executing function: succeeded** (logs 2)  
**Details**  
"Successfully processed 1 messages."  
**Summary**

**Executing function: succeeded** (logs 1)  
**Details**  
"Successfully processed 1 messages."  
**Summary**

<b>Code SHA-256</b> q23mAeJ+VjhxGXhO/sW3gXsUrXyhmSMilpgmIK7Pl6c=	<b>Execution time</b> 2 minutes ago
<b>Function version</b> \$LATEST	<b>Request ID</b> 56257aa5-be8b-477b-8d57-ab5084f25ae5
<b>Duration</b> 77.12 ms	<b>Billed duration</b> 78 ms
<b>Resources configured</b> 128 MB	<b>Max memory used</b> 68 MB

**Log output**

The area below shows the last 4 KB of the execution log. [Click here](#) to view the corresponding CloudWatch log group.

```
START RequestId: 56257aa5-be8b-477b-8d57-ab5084f25ae5 Version: $LATEST
2025-05-08T05:23:17.741Z      56257aa5-be8b-477b-8d57-ab5084f25ae5    INFO    SQS message 19d0b57-b21e-4ac1-bd88-01bb068cb78: "Hello from SQS!"
END RequestId: 56257aa5-be8b-477b-8d57-ab5084f25ae5
REPORT RequestId: 56257aa5-be8b-477b-8d57-ab5084f25ae5  Duration: 77.12 ms   Billed Duration: 78 ms   Memory Size: 128 MB   Max Memory Used: 68 MB
```

**Test event** info  
Delete | CloudWatch Logs Live Tail | Save | **Test**

The lambda function is activated



Navigation: Lambda > Functions > sqs-fun

+ Add trigger

Code | Test | Monitor | **Configuration** | Aliases | Versions

General configuration  
**Triggers**  
Permissions  
Destinations  
Function URL  
Environment variables  
Tags  
VPC  
RDS databases  
Monitoring and operations tools  
Concurrency and recursion detection

**Triggers (1)** Info

Find triggers

Trigger

**SQS: queue**  
arn:aws:sqs:us-east-1:576362453021:queue  
state: **Enabled**

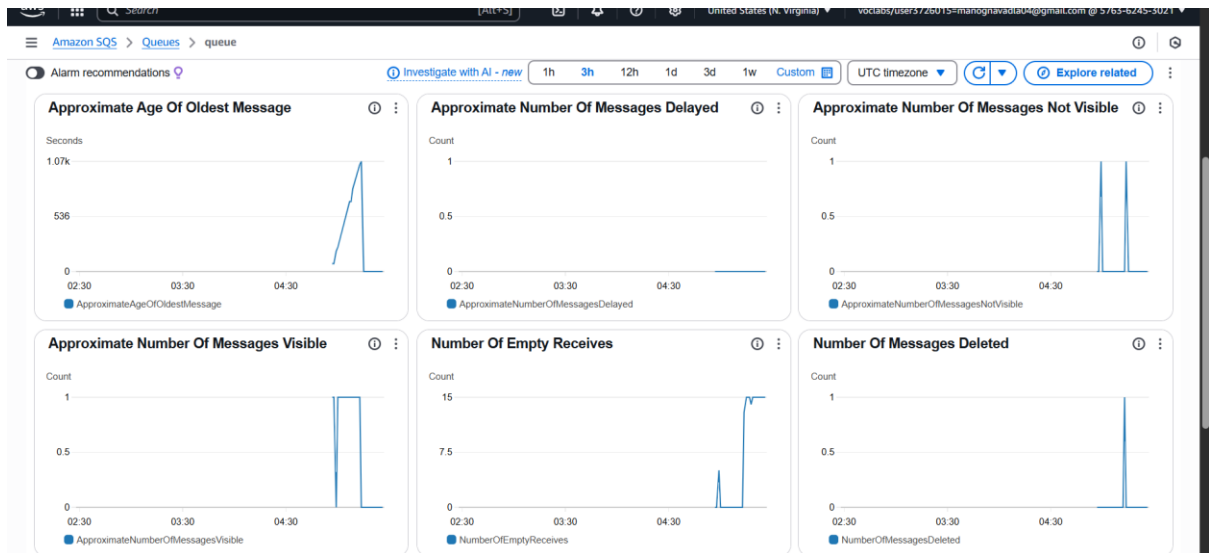
▼ Details

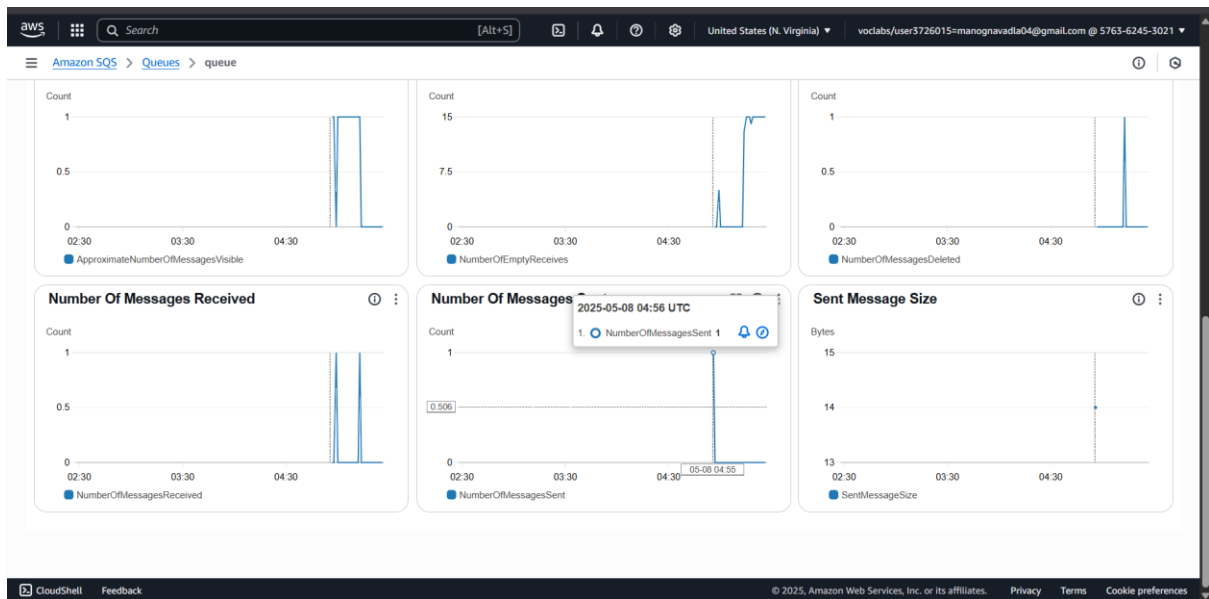
Activate trigger: **Yes**  
Batch size: **10**  
Batch window: **None**  
Event source mapping ARN: **arn:aws:lambda:us-east-1:576362453021:event-source-mapping:4c81f913-aa36-4eb6-82ec-88daa0062b99**  
Metrics: **None**  
On-failure destination: **None**  
Report batch item failures: **No**  
Tags: [View](#)  
UUID: **4c81f913-aa36-4eb6-82ec-88daa0062b99**

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Go to the monitoring tab in queue





Navigate to cloud watch through lambda function, you can see the message that was sent

