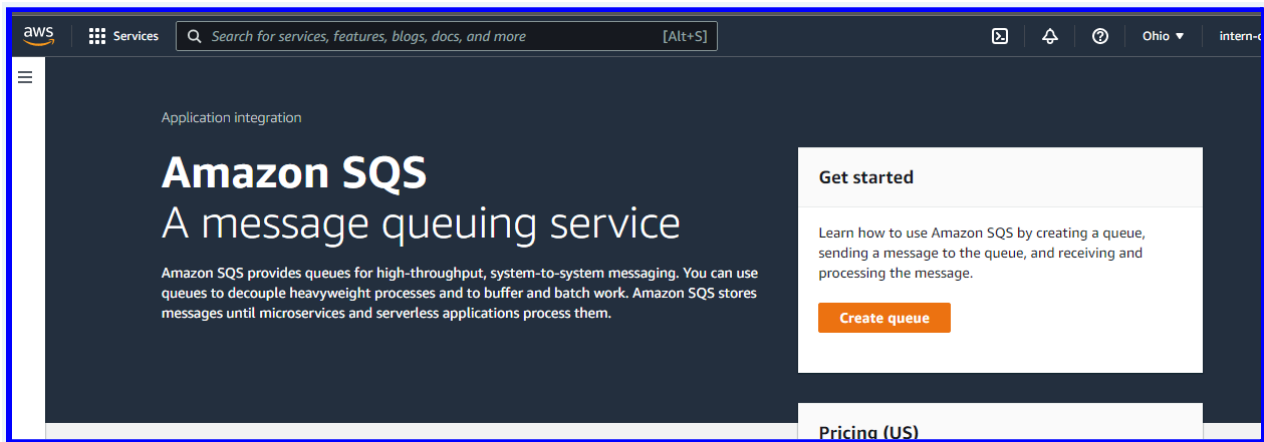


(Optional)

- Create standard SQS.
- Add this SQS as target in above created Event Bridge rule (in addition to existing SNS)
- Add lambda trigger in SQL to sendEmail lambda function.

SQS Home Page



Creating Standard Queue

Amazon SQS > Queues > Create queue

Create queue

Details

Type
Choose the queue type for your application or cloud infrastructure.

You can't change the queue type after you create a queue.

☒ **Standard** [Info](#)

At-least-once delivery, message ordering isn't preserved

- At-least once delivery
- Best-effort ordering

☐ **FIFO** [Info](#)

First-in-first-out delivery, message ordering is preserved

- First-in-first-out delivery
- Exactly-once processing

Name

Team-D-SQS

A queue name is case-sensitive and can have up to 80 characters. You can use alphanumeric characters, hyphens (-), and underscores (_).

Using Default COnfiguration

Configuration

Set the maximum message size, visibility to other consumers, and message retention. [Info](#)

Visibility timeout [Info](#)

Seconds ▼

Should be between 0 seconds and 12 hours.

Message retention period [Info](#)

Days ▼

Should be between 1 minute and 14 days.

Delivery delay [Info](#)

Seconds ▼

Should be between 0 seconds and 15 minutes.

Maximum message size [Info](#)

KB

Should be between 1 KB and 256 KB.

Receive message wait time [Info](#)

Seconds

Should be between 0 and 20 seconds.

Using Basic Method

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
Only the specified AWS account IDs, IAM users and roles can send messages to the queue.

Define who can receive messages from the queue

☒ Only the queue owner
Only the owner of the queue can receive messages from the queue.

☐ Only the specified AWS accounts, IAM users and roles
Only the specified AWS account IDs, IAM users and roles can receive messages from the queue.

JSON (read-only)

```
{
  "Version": "2008-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
      "Sid": "__owner_statement",
      "Effect": "Allow",
      "Principal": {
        "AWS": "949263681218"
      },
      "Action": [
        "SQS:*"
      ],
      "Resource": "arn:aws:sqs:us-east-2:949263681218:Team-D-SQS"
    }
  ]
}
```

Standard Queue Created

Queue Team-D-SQS created successfully
You can now send and receive messages.

Amazon SQS > Queues > Team-D-SQS

Team-D-SQS

EditDeletePurgeSend and receive messagesStart DLQ redrive

DetailsInfo

Name	Type	ARN
Team-D-SQS	Standard	arn:aws:sqs:us-east-2:949263681218:Team-D-SQS
Encryption	URL	Dead-letter queue
Disabled	https://sqs.us-east-2.amazonaws.com/949263681218/Team-D-SQS	-
More		

Updating Event Bridge with SQS as a target

EventBridge - Learning content
Tell us what topics you would like to see more learning material for (tutorials, videos, blog posts, etc.).

Amazon EventBridge > Rules > Team-D-Event-Bridge

Team-D-Event-Bridge

EditDeleteDisable

Rule details

Rule name	Status
Team-D-Event-Bridge	Enabled
Description	Event bus name
Team-D-Event-Bridge	default
Rule ARN	Event bus ARN
arn:aws:events:us-east-2:949263681218:rule/Team-D-Event-Bridge	arn:aws:events:us-east-2:949263681218:event-bus/default
	Monitoring
	Metrics for the rule

Adding SQS as Target too alongside with SNS

Target

Remove

Select target(s) to invoke when an event matches your event pattern or when schedule is triggered (limit of 5 targets per rule).

SNS topic▼

Topic

Team-D-SNS▼

► Configure input

► Retry policy and dead-letter queue

Target

Remove

Select target(s) to invoke when an event matches your event pattern or when schedule is triggered (limit of 5 targets per rule).

SQS queue▼

Queue*

Team-D-SQS▼

► Configure input

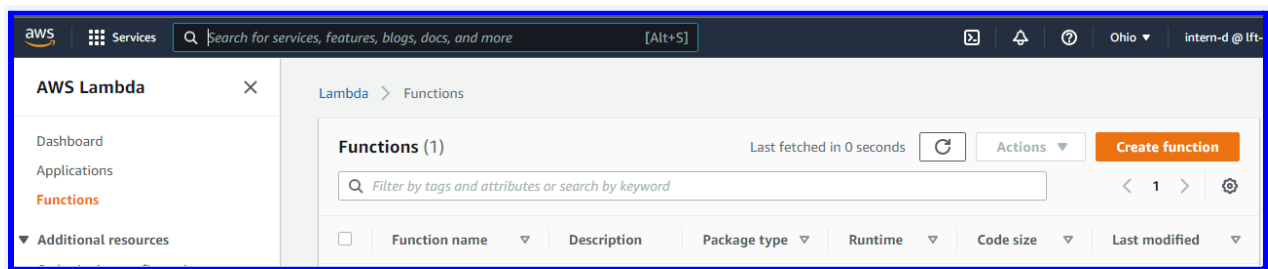
► Retry policy and dead-letter queue

Add target

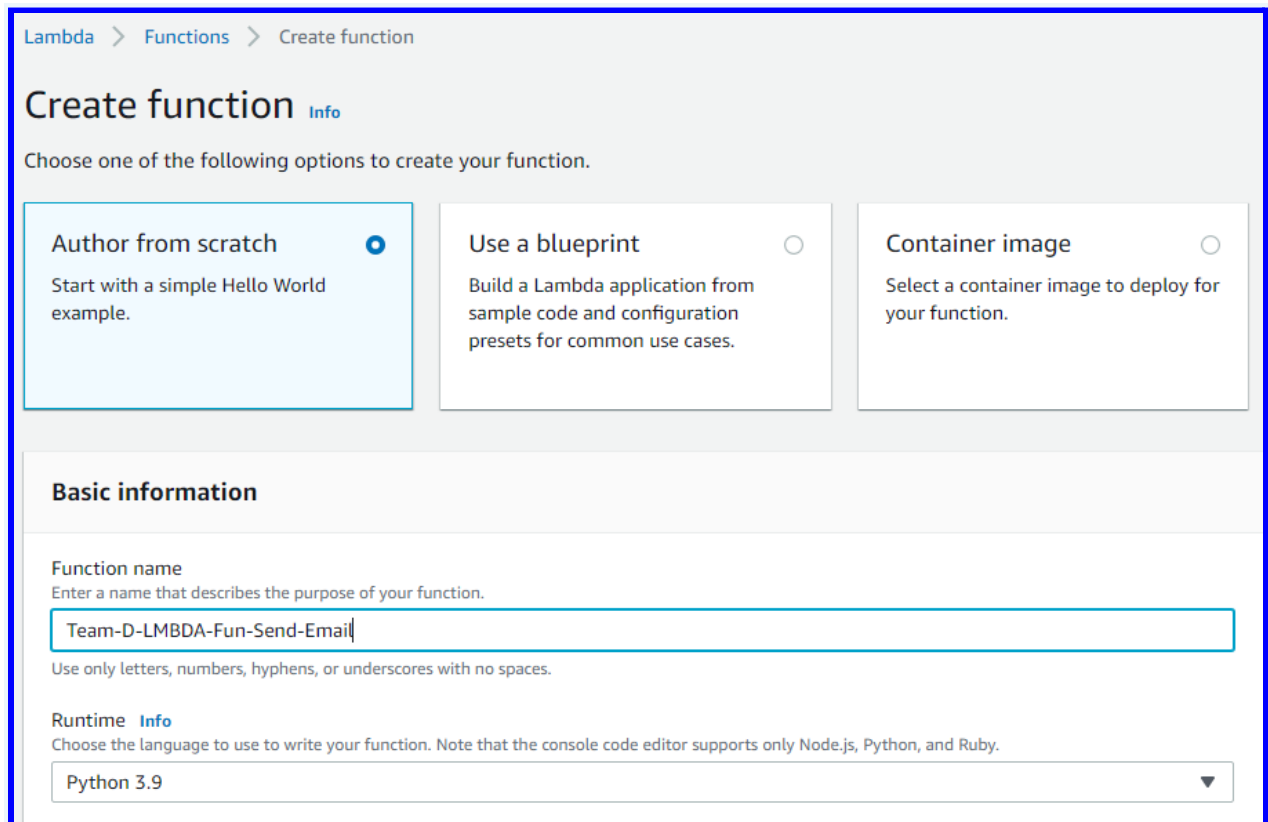
We can see two targets in Event Bridge

Event pattern			
<pre>{ "source": ["aws.ec2"], "detail-type": ["EC2 Instance State-change Notification"] }</pre>			
Target(s) (2) <div>View details</div>			
	Name▼	Type▼	ARN
<input type="radio"/>	Team-D-SNS	SNS topic	arn:aws:sns:us-east-2:949263681218:Team-D-SNS
<input type="radio"/>	Team-D-SQS	SQS queue	arn:aws:sqs:us-east-2:949263681218:Team-D-SQS

Lambda function Home Page



Creating Send Email Function



Function name
Enter a name that describes the purpose of your function.

Team-D-LMBDA-Fun-Send-Email

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Python 3.9

Architecture [Info](#)
Choose the instruction set architecture you want for your function code.

☒ x86_64
☐ arm64

Permissions [Info](#)
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

► [Change default execution role](#)

Using the same code used by Amit Joshi Dai and deployed

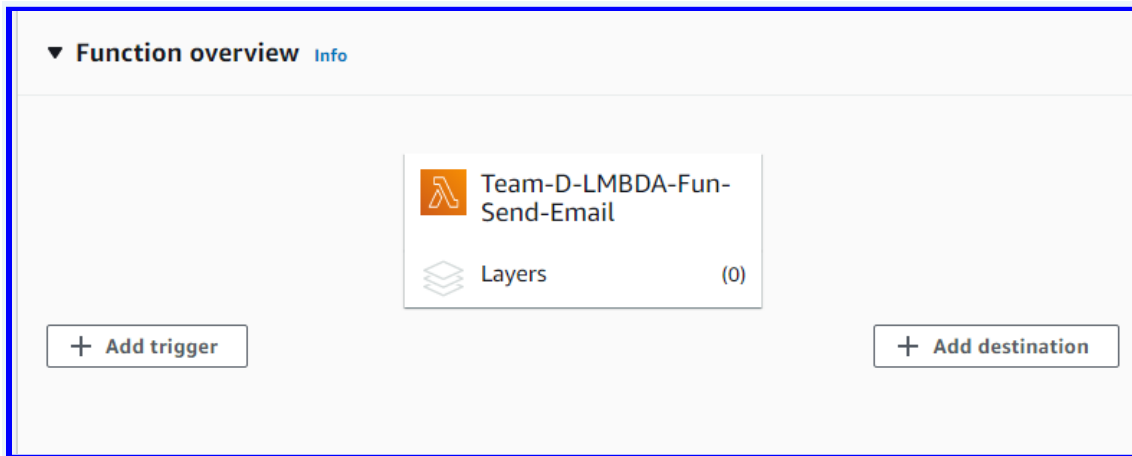
Where email address is changed

```

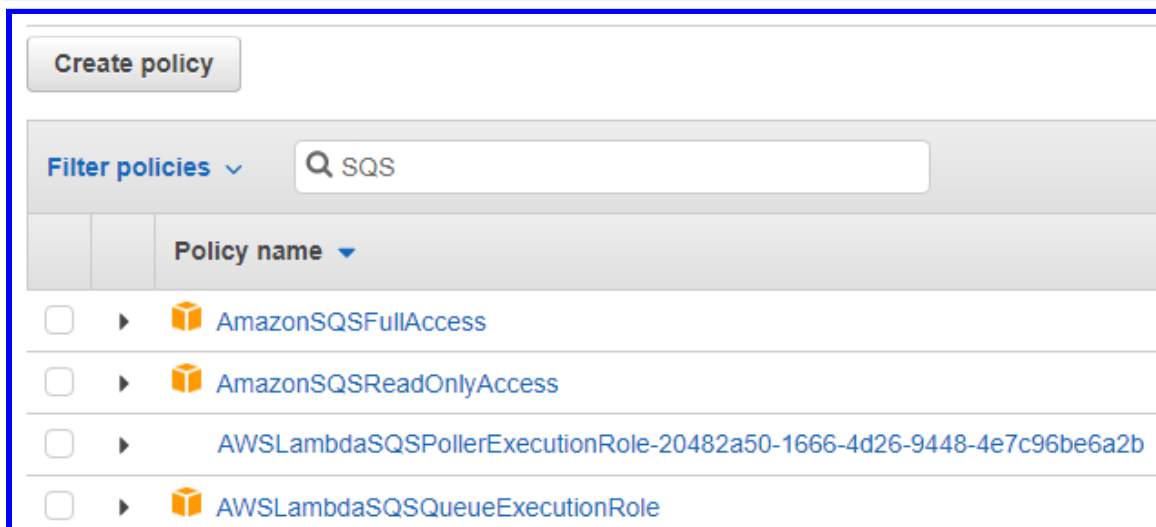
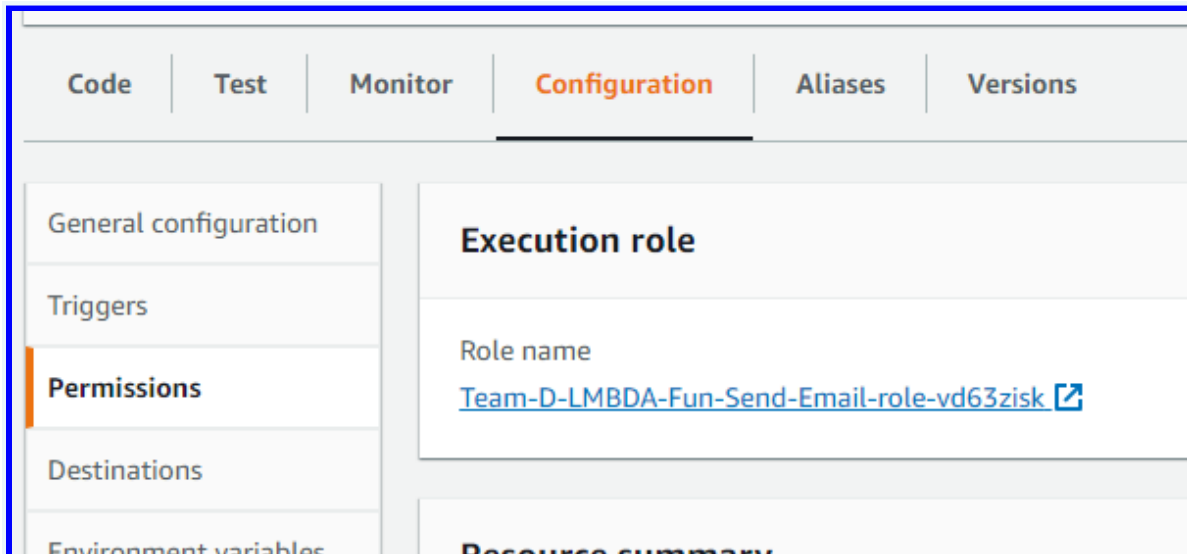
1 import json
2
3 import boto3
4 import os
5
6 ses = boto3.client('ses')
7
8 email_from = 'deesirouss@gmail.com'
9 email_to = 'deesirouss@gmail.com'
10 # email_to_env = os.getenv("TOEMAIL").split(",")
11 # if len(email_to_env) > 0:
12 #     email_to = email_to_env
13 email_subject = 'Subject'
14 email_body = 'Body'
15
16 def lambda_handler(event, context):
17     # TODO implement
18     print(event)
19     email_subject = "Event Occured"
20     email_body = json.dumps(event)
21     response = ses.send_email(
22         Source = email_from,
23         Destination={
24             'ToAddresses': [
25                 email_to,
26             ]
27         },
28         Message={
29             'Subject': {
30                 'Data': email_subject
31             },
32             'Body': {
33                 'Text': {
34                     'Data': email_body,
35                 },

```

Adding SQS Trigger for SQS



Giving the permission of SQS Execution to the Execution Role




Selecting SQS as trigger

Lambda > Add trigger

Add trigger

Trigger configuration

 **SQS**
aws queue


SQS queue
Choose or enter the ARN of an SQS queue.

Batch size
The maximum number of messages to retrieve in a single batch.

Batch window
The maximum amount of time to gather records before invoking the function, in seconds.

▶ **Additional settings - optional**

Creating SQS Trigger

 **SQS**

Function ARN
`arn:aws:lambda:us-east-2:949263681218:function:Team-D-LMBDA-Fun-Send-Email`

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

General configuration

Triggers

Permissions

Destinations


Environment variables

Tags

VPC

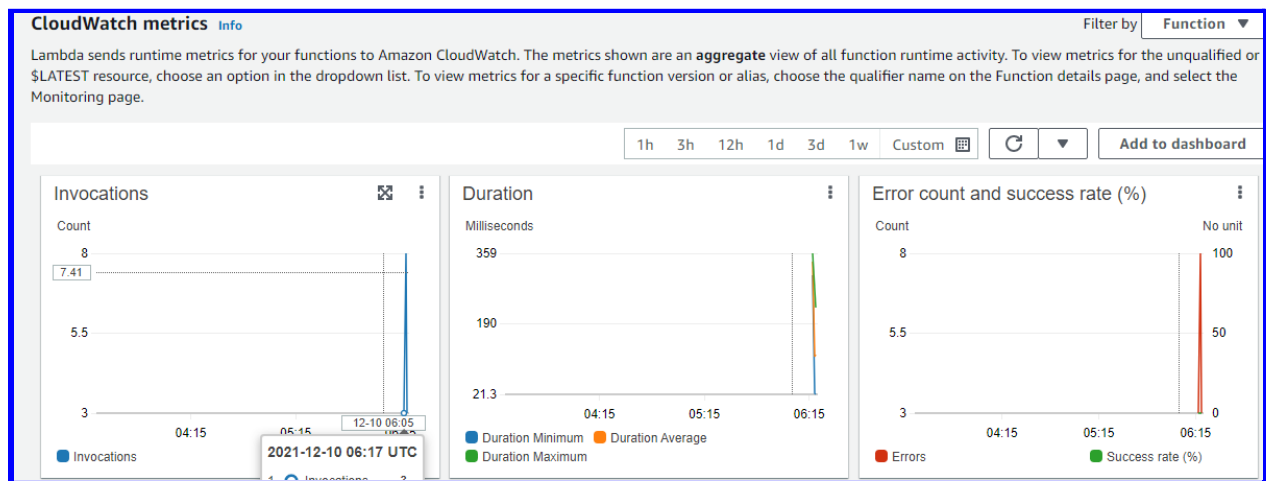
Triggers (1)

☐ Trigger

 **SQS: Team-D-SQS (Creating)**
`arn:aws:sqs:us-east-2:949263681218:Team-D-SQS`
▶ Details

After Successful Enabled Message, we can view metrics, logs for events

CloudWatch Metrics in Monitor Tab



If we see the log Stream for this function in Log Groups Of Cloud Watch

The screenshot displays the CloudWatch Log Groups console for a specific function. The left sidebar shows the 'CloudWatch' navigation menu with options like Favorites, Dashboards, Alarms, Logs, Log groups, Logs Insights, Metrics, Events, Application monitoring, Insights, Settings, and Getting Started. The main area shows the 'Log events' section for the log group `/aws/lambda/Team-D-LMBDA-Fun-Send-Email`. The log events are displayed in a table with columns for Timestamp and Message.

Timestamp	Message
2021-12-10T12:02:53.961+05:45	START RequestId: 2c5424ad-7c77-5b7f-b002-7e818f26d26e Version: \$LATEST
2021-12-10T12:02:53.962+05:45	{'Records': [{'messageId': '3ca75660-75bb-4a05-8600-2b5327677ac8', 'receiptHand
2021-12-10T12:02:54.320+05:45	[ERROR] ClientError: An error occurred (AccessDenied) when calling the SendEmail operation: User 'arn:aws:sts::949263681218:assumed-role/Team-D-LMBDA-Fun-Send-Email-role-vd63zisk/Team-D-LMBDA-Fun-Send-Email' is not authorized to perform 'ses:SendEmail' on resource 'arn:aws:ses:us-east-2:949263681218:identity/deesirouss@gmail.com'

I.e AccessDenied for sending Email, not authorized to perform SES:SendEmail

Again we have to give permission of SES to Execution Role

Add permissions to Team-D-LMBDA-Fun-Send-Email-role-vd63zisk

Attach Permissions

Create policy

Filter policies

	Policy name	Type
<input checked="" type="checkbox"/>	AmazonSESFullAccess	AWS managed
<input type="checkbox"/>	AmazonSESReadOnlyAccess	AWS managed
<input type="checkbox"/>	AWSOpsWorksRegisterCLI_OnPremises	AWS managed
<input type="checkbox"/>	ElementalActivationsGenerateLicenses	AWS managed

Now we don't see any error logs in CloudWatch

```
2021-12-10T12:20:38.091+05:45    START RequestId: f7bda0cb-51a2-5f59-96ff-fae7a6d15a36 Version: $LATEST
2021-12-10T12:20:38.095+05:45    {"Records": [{"messageId": "3a4a7330-4d75-4fd9-bd21-ba3069761515", "receiptHandle": "AQEBHq41AnEx...
{"Records": [{"messageId": "3a4a7330-4d75-4fd9-bd21-ba3069761515", "receiptHandle":
"AQEBHq41AnExgz9hMTAKiUdohFauGOnLoWhya4iGpIrIps7MzhHCgZdqQMZLbtHXW2DDM6158o3NZgtqAppNfLeZV5TWUDr588d0C0qIJ/itquS47SagayISrAjNI2SgN4A9g5QTXKMsG8T3tz+UScR3q/+4Gu74nIJ6EJC1GRjDggyfcerMO9T074SaUTUXaIHsW3KV+q0eT3IYhPHBFnt1gbX2QH/rktpdrLUPkvkg0cVmfT0Ecp4i+9CzNS40Hmf61rqVZzrosfnFcys2IVVYM9mEAtuAW0H34I2jL9kV6yeCj1n5NEjHj1e2tv8d0SqAr5grGtJTTaKL9GNcLVxIrvX/gVDupkrYORbJs6J1sDb5JpYYjnFTViIFDf41V4L8VH3LPpTvmw4GpdpQ==", "body": '{"version":"0","id":"856f1275-1a4c-11b9-a5dc-6a0034959e7c","detail-type":"EC2 Instance State-change Notification","source":"aws.ec2","account":"949263681218","time":"2021-12-10T06:15:58Z","region":"us-east-2","resources":[{"arn:aws:ec2:us-east-2:949263681218:instance/i-03ad64dd899b45fab"}],"detail":{"instance-id":"i-03ad64dd899b45fab","state":"pending"}}', "attributes":{"ApproximateReceiveCount": "36", "SentTimestamp": "1639116959554", "SenderId": "AIDAIMTEBMVXFWR5XIKCK", "ApproximateFirstReceiveTimestamp": "1639116959554"}, "messageAttributes": {}, "md5OfBody": "06308ffe395e83abd9f4f08371e58ee2", "eventSource": "aws:sqs", "eventSourceARN": "arn:aws:sqs:us-east-2:949263681218:Team-D-SQS", "awsRegion": "us-east-2"}]}
2021-12-10T12:20:38.306+05:45    END RequestId: f7bda0cb-51a2-5f59-96ff-fae7a6d15a36
```

And we get the success message of Event Occurred Via Lambda Trigger

Event Occured Inbox x

deesirouss@gmail.com via amazonses.com 12:20 PM (0 minutes ago)

Be careful with this message

This may be a spoofed message. The message claims to have been sent from your account, but Gmail couldn't verify the actual source. Avoid clicking links or replying with sensitive information, unless you are sure you actually sent this message. (No need to reset your password, the real sender does not actually have access to your account)

Report spam Looks safe

```
[{"Records": [{"messageId": "3a4a7330-4d75-4fd9-bd21-ba3069761515", "receiptHandle": "AQEBHq41AnExgz9hMTAKiUdohFauGOnLoWhya4iGpIrIps7MzhHCgZdqQMZLbtHXW2DDM6158o3NZgtqAppNfLeZV5TWUDr588d0C0qIJ/itquS47SagayISrAjNI2SgN4A9g5QTXKMsG8T3tz+UScR3q/+4Gu74nIJ6EJC1GRjDggyfcerMO9T074SaUTUXaIHsW3KV+q0eT3IYhPHBFnt1gbX2QH/rktpdrLUPkvkg0cVmfT0Ecp4i+9CzNS40Hmf61rqVZzrosfnFcys2IVVYM9mEAtuAW0H34I2jL9kV6yeCj1n5NEjHj1e2tv8d0SqAr5grGtJTTaKL9GNcLVxIrvX/gVDupkrYORbJs6J1sDb5JpYYjnFTViIFDf41V4L8VH3LPpTvmw4GpdpQ==", "body": '{"version":"0","id":"856f1275-1a4c-11b9-a5dc-6a0034959e7c","detail-type":"EC2 Instance State-change Notification","source":"aws.ec2","account":"949263681218","time":"2021-12-10T06:15:58Z","region":"us-east-2","resources":[{"arn:aws:ec2:us-east-2:949263681218:instance/i-03ad64dd899b45fab"}],"detail":{"instance-id":"i-03ad64dd899b45fab","state":"pending"}}', "attributes":{"ApproximateReceiveCount": "36", "SentTimestamp": "1639116959554", "SenderId": "AIDAIMTEBMVXFWR5XIKCK", "ApproximateFirstReceiveTimestamp": "1639116959554"}, "messageAttributes": {}, "md5OfBody": "06308ffe395e83abd9f4f08371e58ee2", "eventSource": "aws:sqs", "eventSourceARN": "arn:aws:sqs:us-east-2:949263681218:Team-D-SQS", "awsRegion": "us-east-2"}]}]
```