

Experiment 07 - Cloud Computing

January 18, 2023

Aim

To create and deploy lambda function to transfer csv data to DynamoDB along with adding triggers for S3 and destination as SNS topic for the lambda function.

Theory

- **Lambda** : AWS Lambda is an event-driven, serverless computing platform provided by Amazon as a part of Amazon Web Services. It is a computing service that runs code in response to events and automatically manages the computing resources required by that code.
- **Triggers** : A trigger is a resource you configure to allow another AWS service to invoke your function when certain events or conditions occur. Your function can have multiple triggers. Lambda Functions can be triggered in different ways: an HTTP request, a new document upload to S3, a scheduled Job, an AWS Kinesis data stream, or a notification from AWS Simple Notification Service (SNS).
- **SNS** : Amazon Simple Notification Service (Amazon SNS) is a managed service that provides message delivery from publishers to subscribers (also known as producers and consumers). Publishers communicate asynchronously with subscribers by sending messages to a topic, which is a logical access point and communication channel.

Results

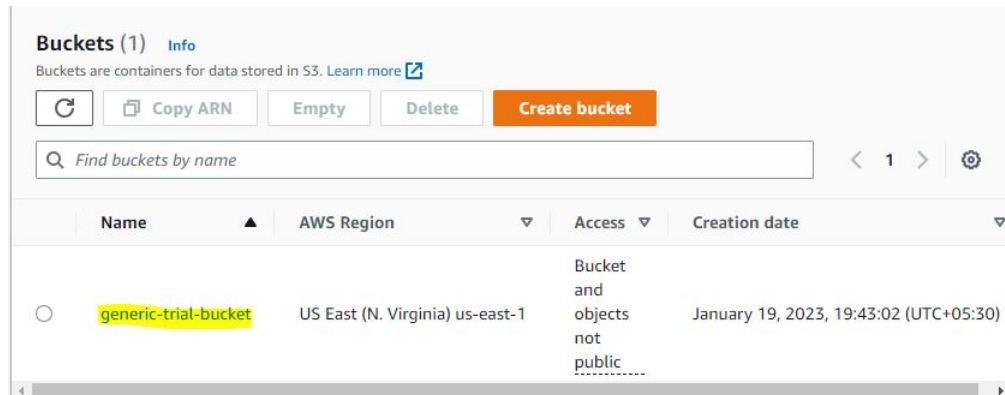


Figure 1: Start by creating S3 Bucket with name you prefer and default settings

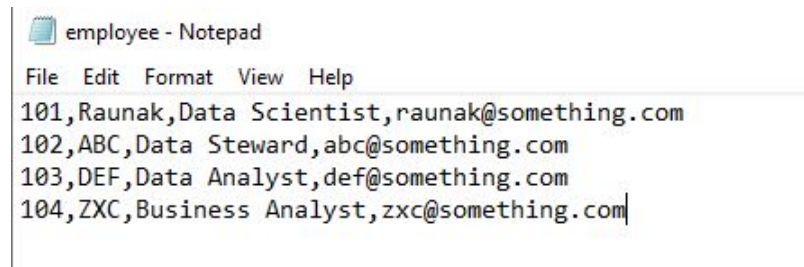


Figure 2: Create a random comma separated value file with arbitrary data

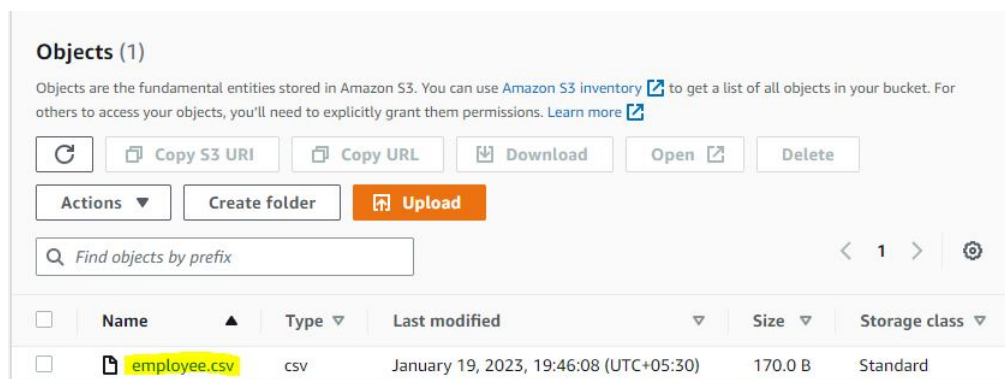


Figure 3: Upload the created CSV file in the bucket you had created

Tables (1) [Info](#)

Find tables by table name

Any table tag

<

1

>

Name

Status

Partition key

Sort key

Indexes

Read capacity mode

employeeData

Active

empID (N)

-

0

Provisioned with auto scaling (5)

Figure 4: Create a table in DynamoDB with empID as its primary key

IAM > Roles > Create role

Select trusted entity [Info](#)

Trusted entity type

☒ **AWS service**
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

☐ **AWS account**
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

☐ **Web identity**
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

☐ **SAML 2.0 federation**
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

☐ **Custom trust policy**
Create a custom trust policy to enable others to perform actions in this account.

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Common use cases

☐ **EC2**
Allows EC2 instances to call AWS services on your behalf.

☒ **Lambda**
Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:

[Cancel](#)
[Next](#)

Figure 5: Create an IAM role that supports the Lambda service

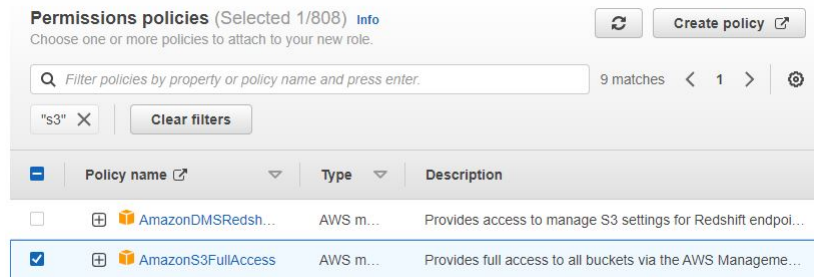


Figure 6: Enable the S3 Full Access Function

Permissions policy summary

Policy name	Type	Attached as
AWSLambdaBasicExecutionRole	AWS managed	Permissions policy
AWSLambda_FullAccess	AWS managed	Permissions policy
AWSLambdaDynamoDBExecutionRole	AWS managed	Permissions policy
AmazonDynamoDBFullAccess	AWS managed	Permissions policy
AmazonS3FullAccess	AWS managed	Permissions policy

Figure 7: Similarly follow all the displayed services respectively

Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.

Maximum 64 characters. Use alphanumeric and '+', '@', '-' characters.

Figure 8: Create the name for role details and proceed creating the role

Basic information

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

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.

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

☒ x86_64

☐ arm64

Figure 9: Create a Lambda Function and give name of your choice along with select runtime as Python

▼ Change default execution role

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

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.

[View the genericLambdaService role](#) on the IAM console.

► Advanced settings

Cancel **Create function**

Figure 10: Change the default execution role by choosing an existing service and selecting the IAM role you created initially from the dropdown menu and create the function

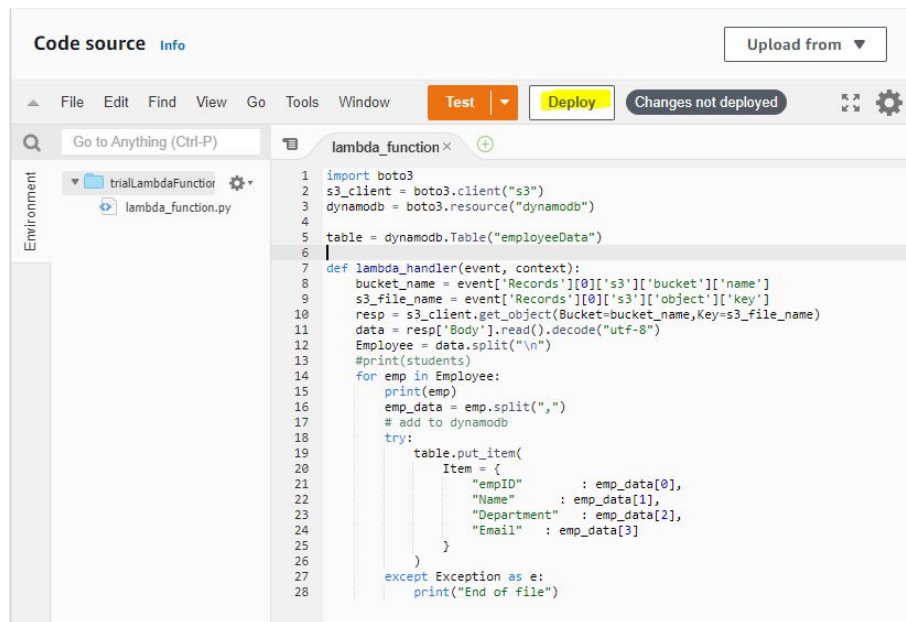


Figure 11: In code source type the following script that takes the data from bucket and puts in the dynamo DB table. Do not forget to deploy once completed to proceed with execution

Test event Info Save Test

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

☒ Create new event ☐ Edit saved event

Event name

basicTest

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings

☒ Private
This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)

☐ Shareable
This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

Template - optional

s3-put

Figure 12: Create a test event to test if the function is working effectively

Event JSON

Format JSON

```
10      "principalId": "EXAMPLE"
11    },
12    "requestParameters": {
13      "sourceIPAddress": "127.0.0.1"
14    },
15    "responseElements": {
16      "x-amz-request-id": "EXAMPLE123456789",
17      "x-amz-id-2": "EXAMPLE123/5678abcdefghijklmbdaisawesom/mnopqrstuvwxyzABCDEFGH"
18    },
19    "s3": {
20      "s3SchemaVersion": "1.0",
21      "configurationId": "testConfigRule",
22      "bucket": {
23        "name": "generic-trial-bucket",
24        "ownerIdentity": {
25          "principalId": "EXAMPLE"
26        },
27        "arn": "arn:aws:s3:::generic-trial-bucket"
28      },
29      "object": {
30        "key": "employee.csv",
31        "size": 1024,
32        "eTag": "0123456789abcdef0123456789abcdef",
33        "sequencer": "0A1B2C3D4E5F678901"
34      }
35    }
36  }
37 }
38 }
```

Figure 13: Select the S3-Put option and make required changes in the Event JSON file in the place by selecting name of your bucket with its respective ARN and employee file that was uploaded in the bucket

Test event

Info

Delete

Save

Test

To invoke your function without saving an event, modify the event, then choose Test. Lambda uses the modified event to invoke your function, but does not overwrite the original event until you choose Save changes.

Test event action

☐ Create new event

☒ Edit saved event


Event name

basicTest

▼

↺

Figure 14: After saving the JSON proceed by testing

**Execution result: succeeded** ([logs](#))

▼ Details

The area below shows the last 4 KB of the execution log.

null

Summary

Code SHA-256	Request ID
JhCtOHQO5cxMjylQgbACbA02ICiRWsKHet2z+ZSA6FU=	b1ab3389-4b1b-41e3-bcb7-254fe3534a53
Init duration	Duration
385.94 ms	548.90 ms
Billed duration	Resources configured
549 ms	128 MB
Max memory used	
73 MB	

Log output

The section below shows the logging calls in your code. [Click here](#) to view the corresponding CloudWatch log group.

START RequestId: b1ab3389-4b1b-41e3-bcb7-254fe3534a53 Version: \$LATEST
101,Raunak,Data Scientist,raunak@something.com
102,ABC,Data Steward,abc@something.com
103,DEF,Data Analyst,def@something.com
104,ZXC,Business Analyst,zxc@something.com
END RequestId: b1ab3389-4b1b-41e3-bcb7-254fe3534a53
REPORT RequestId: b1ab3389-4b1b-41e3-bcb7-254fe3534a53 Duration: 548.90 ms Billed Duration: 549 ms Memory Size: 128 MB Max Memory Used: 73 MB Init Duration: 385.94 ms


Figure 15: Successful execution of the test that was created

Items returned (4)				
<input type="checkbox"/>	ID	Departme...	Email	Name
<input type="checkbox"/>	103	Data Analyst	def@somet...	DEF
<input type="checkbox"/>	102	Data Steward	abc@somet...	ABC
<input type="checkbox"/>	104	Business An...	zxc@somet...	ZXC
<input type="checkbox"/>	101	Data Scientist	raunak@so...	Raunak


Figure 16: Successful addition of the records

▼ Function overview

Info



trialLamb
daFunction



Layers (0)

+ Add trigger

+ Add destination

Description

-

Last modified

3 minutes ago

Function ARN




arn:aws:lambda:us-east-1:303164066091:function:trialLambdaFunction

Figure 17: Adding the triggers for function

Trigger configuration [Info](#)

 **S3**
aws storage

Bucket

Please select the S3 bucket that serves as the event source. The bucket must be in the same region as the function.

× ↺

Bucket region: us-east-1

Event type

Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.

Prefix - optional

Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.

Suffix - optional

Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters.

Recursive invocation

If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. [Learn more](#)

☒ I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.

Lambda will add the necessary permissions for AWS S3 to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

Cancel Add

Figure 18: Follow the respective information for triggers related to S3

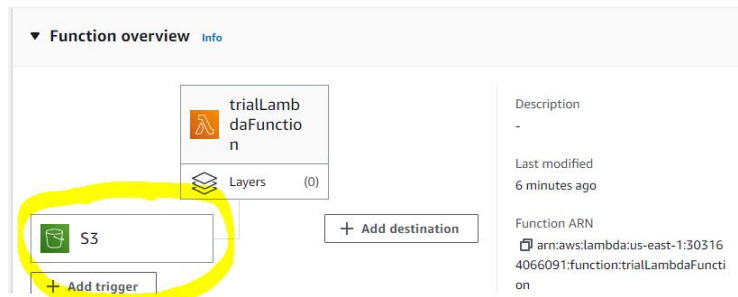


Figure 19: Successful addition of the trigger for function

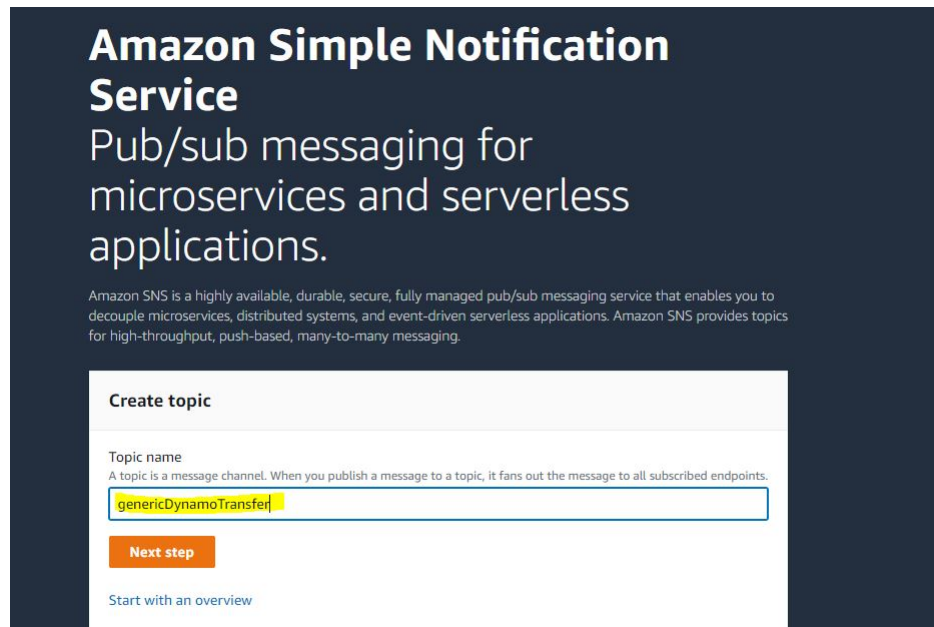


Figure 20: Create the AWS SNS service by the name of your choice

Create subscription

Details

Topic ARN

Protocol

The type of endpoint to subscribe

Email

Endpoint

An email address that can receive notifications from Amazon SNS.

Figure 21: Create the subscription

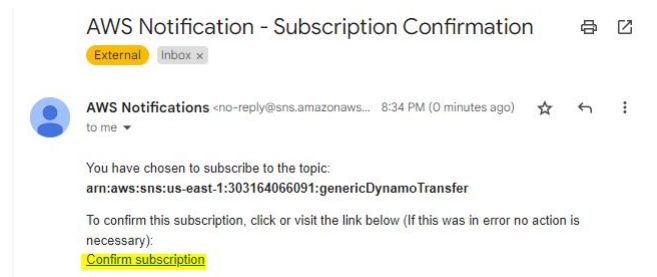


Figure 22: Confirm the email notification

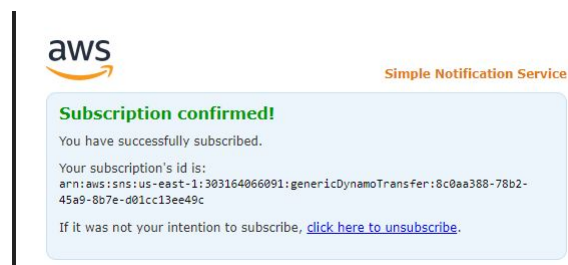


Figure 23: Successful acceptance of subscription

Destination configuration

Send invocation records to a destination when your function is invoked asynchronously, or if your function processes records from a stream.

Source

The type of invocation that maps to the destination.

☒ Asynchronous invocation

☐ Stream invocation

Condition

The condition for using the destination.

☐ On failure

☒ On success

Destination type

An SQS queue, SNS topic, Lambda function, or EventBridge event bus.

SNS topic

ⓘ Your function's execution role doesn't have permission to send result to the destination. By clicking save we'll attempt to add permission to the role for you.

Destination

genericDynamoTransfer

Cancel

Save

Figure 24: Change the configuration for the lambda service for SNS

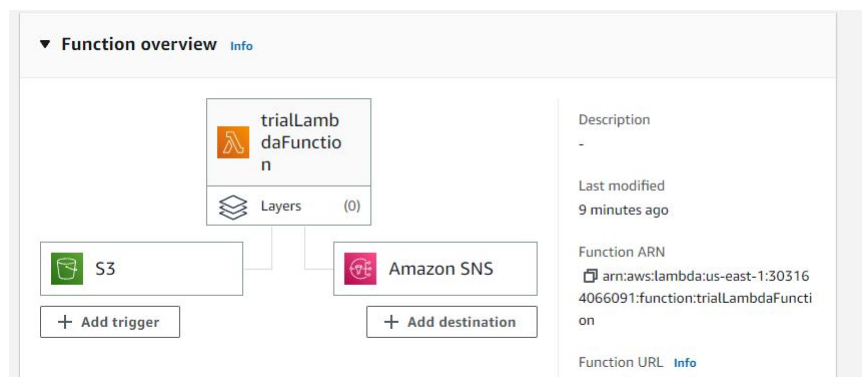


Figure 25: Successful addition of the SNS service for Lambda function

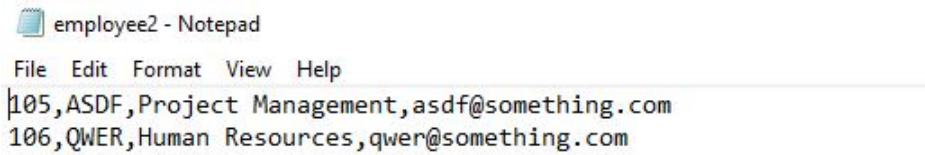


Figure 26: Create one more employee file with more records and upload to S3 bucket

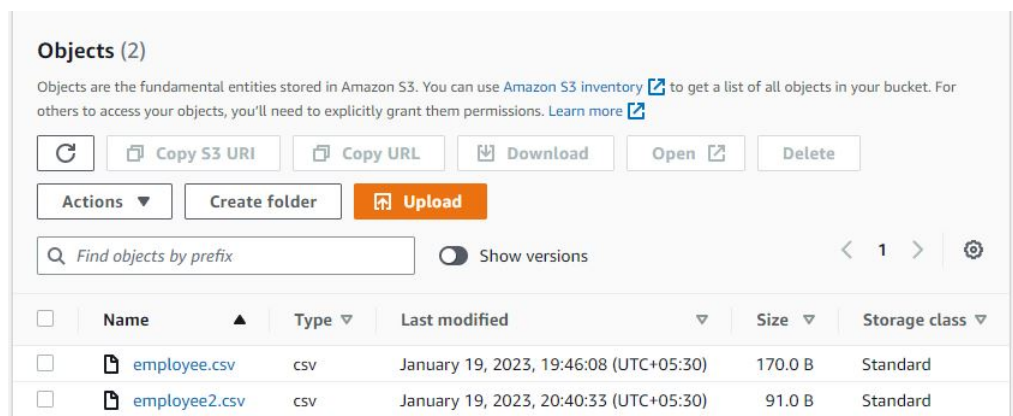


Figure 27: Successful uploading the newly created CSV in the bucket

Items returned (6)				
<input type="checkbox"/>	ID	Departme...	Email	Name
<input type="checkbox"/>	101	Data Scientist	raunak@so...	Raunak
<input type="checkbox"/>	102	Data Steward	abc@somet...	ABC
<input type="checkbox"/>	103	Data Analyst	def@somet...	DEF
<input type="checkbox"/>	104	Business An...	zxc@somet...	ZXC
<input type="checkbox"/>	105	Project Man...	asdf@some...	ASDF
<input type="checkbox"/>	106	Human Reso...	qwer@som...	QWER

Figure 28: After running the test again successful you can see addition of newly created records

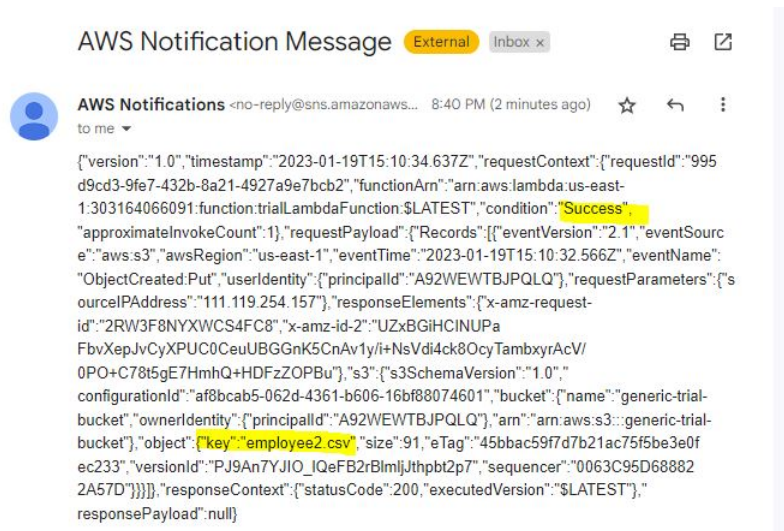


Figure 29: Success email retrieval using SNS is also successful

Conclusion

This experiment is successful demonstration of the AWS Lambda service that considers the content from the AWS S3 bucket and uses for adding the data in the AWS DynamoDB Table. The Lambda basically acts as the server that executes the following service and creates a serverless application. Along with all the said, execution of the AWS SNS is also used that retrieves the success message on successful execution of every service call. The execution of the all the process has been successfully shown in the results section of this experiment respectively.