

# To update data into DynamoDB using Lambda Function (triggered on S3 Event)

## 1. Role and Policies attached

[IAM](#) > [Roles](#) > [roleforlambdaS3toDDB](#)

### roleforlambdaS3toDDB

Delete

Allows Lambda functions to call AWS services on your behalf.

#### Summary

Edit

Creation date

February 01, 2023, 12:27 (UTC+05:30)

Last activity

✓ 54 minutes ago

ARN

arn:aws:iam::461292457774:role/roleforlambdaS3toDDB

Maximum session duration

1 hour

Permissions

Trust relationships

Tags

Access Advisor

Revoke sessions

#### Permissions policies (3) [Info](#)

You can attach up to 10 managed policies.



Simulate

Remove

Add permissions ▼

🔍 Filter policies by property or policy name and press enter.

< 1 > ⚙️

<input type="checkbox"/>	Policy name <a href="#">🔗</a>	Type	Description
<input type="checkbox"/>	<a href="#">+ AmazonDynamoDBFullAccess</a>	AWS managed	Provides full access to Amazon DynamoDB via the AWS Management Console.
<input type="checkbox"/>	<a href="#">+ AmazonS3ReadOnlyAccess</a>	AWS managed	Provides read only access to all buckets via the AWS Management Console.
<input type="checkbox"/>	<a href="#">+ AWSLambdaBasicExecutionRole</a>	AWS managed	Provides write permissions to CloudWatch Logs.

## 2. S3 Bucket

Amazon S3 > Buckets > source-emp

source-emp [Info](#)

Objects

Properties

Permissions

Metrics

Management

Access Points

### Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)



Copy S3 URI

Copy URL

Download

Open

Delete

Actions ▼

Create folder

Upload

Find objects by prefix

< 1 > ⚙

<input type="checkbox"/>	Name ▲	Type ▼	Last modified ▼	Size ▼	Storage class ▼
<input type="checkbox"/>	<a href="#">employeedata.csv</a>	csv	February 1, 2023, 12:34:09 (UTC+05:30)	358.0 B	Standard


### 3. Lambda Function


Lambda > Functions > automateCSVtoDDB


## automateCSVtoDDB

Throttle Copy ARN Actions ▼

▼ Function overview [Info](#)

 automateCSVtoDDB

 Layers (0)

 S3

+ Add trigger

+ Add destination


Description

-

Last modified

9 minutes ago

Function ARN

 arn:aws:lambda:ap-south-1:461292457774:function:automat  
eCSVtoDDB

Function URL [Info](#)

-

# Function Code

**Code source** [Info](#)

File Edit Find View Go Tools Window **Test** Deploy

Go to Anything (Ctrl-P)

Environment

- automateCSVtoDDB
  - lambda\_function.py

```
1 import boto3
2 import os
3
4 bucket_name = os.getenv('bucket_name')
5
6 s3_client = boto3.client("s3")
7 dynamodb= boto3.resource('dynamodb')
8 table= dynamodb.Table('employees')
9
10 def lambda_handler(event, context):
11     # bucket_name = event ['Records'][0]['s3']['bucket']['name']
12     s3_file_name = event ['Records'][0]['s3']['object']['key']
13
14     if s3_file_name == 'employeeedata.csv':
15         resp= s3_client.get_object(Bucket=bucket_name, Key=s3_file_name)
16         data = resp['Body'].read().decode("utf-8")
17         employees = data.split("\n")
18         for i in range(1,len(employees)):
19             items=employees[i].split(",")
20             ddmmyyyy=items[4].split("-")
21             ddmmyyyy.reverse() #converts to yyyymmdd
22             # Add it to dynamodb
23             table.put_item(
24                 Item = {
25                     "id" : int(items[0]),
26                     "dept" : items[1],
27                     "userName" : items[2]+" "+items[3],
28                     "DOB" : "-".join(ddmmyyyy)
29                 }
30             )
31     else :
32         print("Check the file name!!")
```

# Execution Results post Deployment

lambda\_function. x

Execution result: x

+

▼ Execution results

Status: SucceededMax memory used: 73 MBTime: 715.68 ms

<b>Test Event Name</b> testing01	
<b>Response</b> null	
<b>Function Logs</b> START RequestId: cd4606b8-cc3d-459b-a29b-48e491626c1d Version: \$LATEST END RequestId: cd4606b8-cc3d-459b-a29b-48e491626c1d REPORT RequestId: cd4606b8-cc3d-459b-a29b-48e491626c1d   Duration: 715.68 ms Billed Duration: 716 ms Memory Size: 128 MB Max Memory Used: 73 MB   Init Duration: 420.76 ms	
<b>Request ID</b> cd4606b8-cc3d-459b-a29b-48e491626c1d	

## 4. DynamoDB items

### employees

▼ Scan or query items

☒ Scan

☐ Query

Select a table or index  
Table - employees

► Filters

Run

Reset

✔ Completed. Read capacity units consumed: 0.5

Items returned (10)

<input type="checkbox"/>	id	dept	DOB	userName
<input type="checkbox"/>	7	Finance	1998-03-01	Mayur Sinha
<input type="checkbox"/>	8	IT	1985-08-26	Amit Sharma
<input type="checkbox"/>	10	Admin	1994-05-28	Smita Raina
<input type="checkbox"/>	3	IT	1998-02-01	Maya Patil
<input type="checkbox"/>	2	Finance	1992-05-28	Suresh Mishra
<input type="checkbox"/>	9	Admin	1981-03-12	Prajakta Patil
<input type="checkbox"/>	4	IT	1989-08-27	Prashant Jadhav
<input type="checkbox"/>	6	IT	1993-05-28	Pratik Kumar
<input type="checkbox"/>	1	Admin	1998-01-01	Deepti Sharma
<input type="checkbox"/>	5	Finance	1998-02-01	Deepak Kumar