

Loading a csv file into dynamodb through Lambda

1)Setup an s3 bucket as below

The first screenshot shows the AWS S3 console 'Create bucket' page. The bucket name is 'democsv2dynamodb' and the region is 'US East (N. Virginia) us-east-1'. The 'Block all public access' checkbox is checked. The second screenshot shows the 'Default encryption' section with 'Server-side encryption' set to 'Enable' and 'Amazon S3 key (SSE-S3)' selected. The 'Create bucket' button is visible at the bottom right.

2)Check the crimes data csv file to identify data,primary key and data types

Create the table “Chicagocrimes” in Dynamodb with primary key “id”

The screenshot displays the AWS DynamoDB console interface. The top navigation bar includes the AWS logo, 'Services' dropdown, and user information. A notification banner at the top states: 'The preview of the new DynamoDB console is now available. We are redesigning the DynamoDB console. The preview of the new console is a work in progress, but we encourage you to try it and let us know what you think.'

Create DynamoDB table

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name* Chicagocrimes

Primary key* Partition key: **Number**

☐ Add sort key

Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

☒ Use default settings

- No secondary indexes.
- Auto Scaling capacity set to 70% target utilization, at minimum capacity of 5 reads and 5 writes.
- Encryption at Rest with DEFAULT encryption type.

[+ Add tags](#)

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service. Advanced alarm settings are available in the CloudWatch management console.

[Cancel](#) [Create](#)

The screenshot then shows the 'Chicagocrimes' table overview. The left sidebar lists navigation options: Dashboard, Tables (selected), Backups, Reserved capacity, Preferences, DAX, Clusters, Subnet groups, Parameter groups, Events, and 'Try the preview of the new console'. The main content area shows the table 'Chicagocrimes' with tabs for Overview, Items, Metrics, Alarms, Capacity, Indexes, Global Tables, Backups, Contributor Insights, Triggers, and More. The 'Items' tab is active, showing a search bar with the filter '[Table] Chicagocrimes: id' and a 'Start search' button. A note at the bottom states: 'An item consists of one or more attributes. Each attribute consists of a name, a data type, and a value. When you read or write an item, the only attributes that are required are those that make up the primary key. [More Info](#)'

3) Create role for **lambda** to access S3, Dynamodb and CloudWatch for logs

Cloud Sandboxes - A Cloud x AWS Management Con x IAM Management Con x DynamoDB - AWS Con x S3 Management Con x CSV to Dynamodb La x CSV to Dynamodb La x + - □ ×

← → ↻ console.aws.amazon.com/iam/home?region=us-east-1#/roles\$new?step=review&commonUseCase=Lambda%2BLambda&selectedUseCase=Lambda&policies=arn:aws:iam::aws:policy%2FA...

aws Services ▼ cloud_user @ 7807-4357-0903 ▼ Global ▼ Support ▼

Create role

1 2 3 4

Review

Provide the required information below and review this role before you create it.

Role name* csv-2-dynamodb
Use alphanumeric and '+', '@', '_' characters. Maximum 64 characters.

Role description Allows Lambda functions to call AWS services on your behalf.
Maximum 1000 characters. Use alphanumeric and '+', '@', '_' characters.

Trusted entities AWS service: lambda.amazonaws.com

Policies

- AmazonDynamoDBFullAccess
- AmazonS3FullAccess
- AWSOpsWorksCloudWatchLogs

Permissions boundary Permissions boundary is not set

No tags were added.

* Required

Cancel Previous **Create role**

Feedback English (US) ▼ © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

4) Create lambda function

Cloud Sandboxes - A Cloud x Lambda x DynamoDB - AWS Console x S3 Management Console x CSV to Dynamodb Lambda x CSV to Dynamodb Lambda x + - □ ×

← → ↻ console.aws.amazon.com/lambda/home?region=us-east-1#/create/function

aws Services ▼ cloud_user @ 7807-4357-0903 ▼ N. Virginia ▼ Support ▼

Create function

Function name
Enter a name that describes the purpose of your function.
csv-2-dynamodb
Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)
Choose the language to use to write your function.
Python 3.8

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**

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

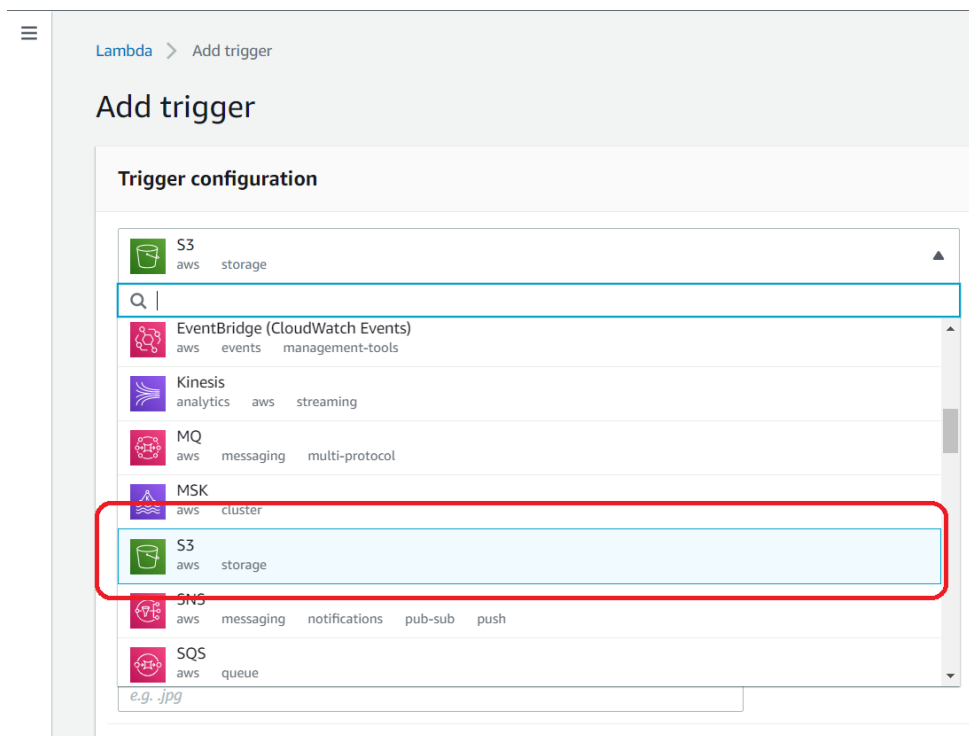
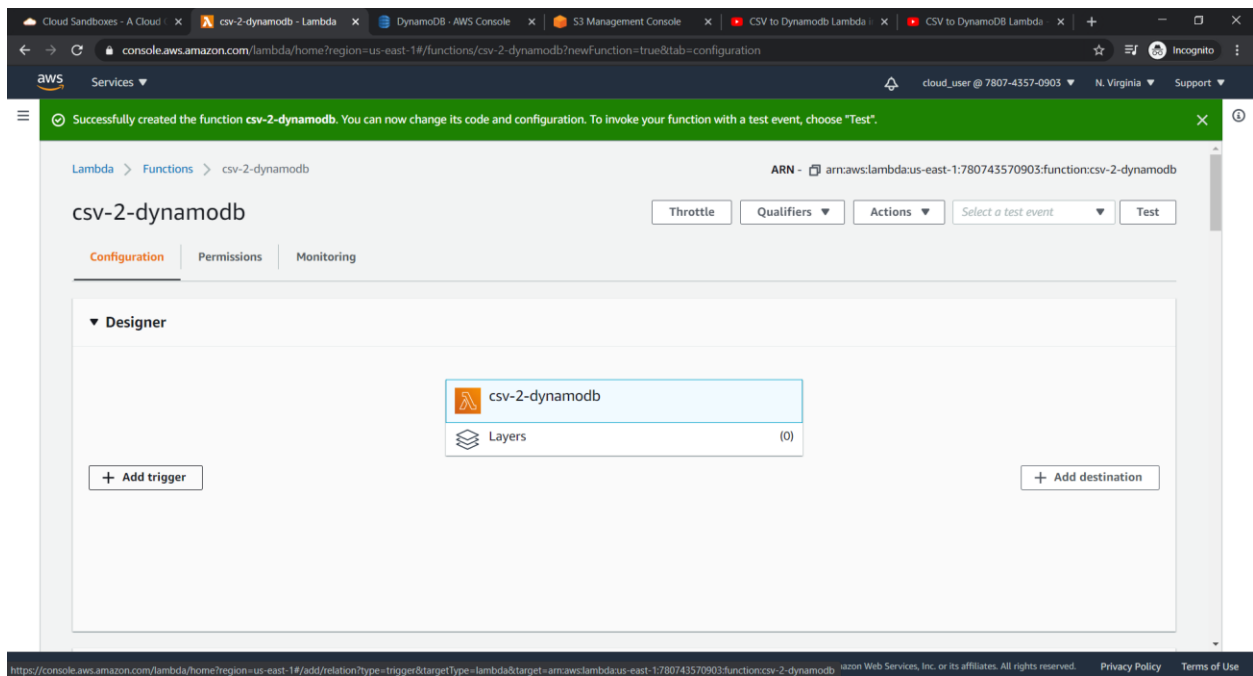
☐ Create a new role with basic Lambda permissions
☒ Use an existing role
☐ Create a new role from AWS policy templates

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.
csv-2-dynamodb [View the csv-2-dynamodb role on the IAM console.](#)

► **Advanced settings**

Feedback English (US) ▼ © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

5) Add trigger



Cloud Sandboxes - A Cloud | x Lambda x DynamoDB - AWS Console x S3 Management Console x CSV to Dynamodb Lambda x CSV to DynamoDB Lambda x + - □ ×

console.aws.amazon.com/lambda/home?region=us-east-1#/add/relation?focus=lambda&target=arn%3Aaws%3Alambda%3Aus-east-1%3A780743570903%3Afunction%3Acsv-2-dynamodb... ☆ Incognito

aws Services ▼ cloud_user @ 7807-4357-0903 N. Virginia Support ▼

Bucket
Please select the S3 bucket that serves as the event source. The bucket must be in the same region as the function.
democsv2dynamodb

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.
PUT

Prefix - optional
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.
e.g. images/

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

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

The Lambda console no longer supports disabling S3 and CloudWatch Logs triggers. Delete these triggers to stop further actions.

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](#)

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Cloud Sandboxes - A Cloud | x Lambda x DynamoDB - AWS Console x S3 Management Console x CSV to Dynamodb Lambda x CSV to DynamoDB Lambda x + - □ ×

console.aws.amazon.com/lambda/home?region=us-east-1#/add/relation?focus=lambda&target=arn%3Aaws%3Alambda%3Aus-east-1%3A780743570903%3Afunction%3Acsv-2-dynamodb... ☆ Incognito

aws Services ▼ cloud_user @ 7807-4357-0903 N. Virginia Support ▼

Bucket
Please select the S3 bucket that serves as the event source. The bucket must be in the same region as the function.
democsv2dynamodb

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.
PUT

Prefix - optional
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.
e.g. images/

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

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

The Lambda console no longer supports disabling S3 and CloudWatch Logs triggers. Delete these triggers to stop further actions.

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. [Learn more](#)

Cancel Add

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Now click on Add button

6)Go to the **Code** tab update the code and click on “Deploy”

Note : Boto3 is the **Amazon Web Services (AWS) Software Development Kit (SDK) for Python**, which allows Python developers to write software that makes use of services like Amazon S3 and Amazon EC2.

```
import json

import csv

import boto3

def lambda_handler(event, context):

    region = 'us-east-1'

    record_list = []

    try:

        s3=boto3.client('s3')

        dynamodb=boto3.client('dynamodb',region_name=region)

        bucket=event['Records'][0]['s3']['bucket']['name']

        key=event['Records'][0]['s3']['object']['key']

        print('Bucket:', bucket,'Key:', key)

        csv_file=s3.get_object(Bucket=bucket,Key=key)

        record_list=csv_file['Body'].read().decode('utf-8').split('\n')

        csv_reader=csv.reader(record_list,delimiter=',',quotechar="''")

        firstrecord=True

        for row in csv_reader:

            if(firstrecord):

                firstrecord=False

                continue

            id=row[0]

            case_number=row[1]

            date=row[2]

            block=row[3]

            iucr_code=row[4]
```

```
location_desc=row[5]
arrest=row[6]
domestic=row[7]
beat_num=row[8]
district_code=row[9]
ward_no=row[10]
community_code=row[11]
fbi_code=row[12]
x_coordinate=row[13]
y_coordinate=row[14]
year=row[15]
date_of_update=row[16]
latitude=row[17]
longitude=row[18]
location=row[19]
```

```
print('id:',id)
```

```
add_to_db=dynamodb.put_item(TableName='Chicagocrimes',Item={'id':{'N':str(id)},'case_number':{'S':str(case_number)},'date':{'S':str(date)},'block':{'S':str(block)},'iucr_code':{'S':str(iucr_code)},'location_desc':{'S':str(location_desc)},'arrest':{'S':str(arrest)},'domestic':{'S':str(domestic)},'beat_num':{'N':str(beat_num)},'district_code':{'N':str(district_code)},'ward_no':{'N':str(ward_no)},'community_code':{'N':str(community_code)},'fbi_code':{'S':str(fbi_code)},'x_coordinate':{'S':str(x_coordinate)},'y_coordinate':{'S':str(y_coordinate)},'year':{'N':str(year)},'date_of_update':{'S':str(date_of_update)},'latitude':{'S':str(latitude)},'longitude':{'S':str(longitude)},'location':{'S':str(location)}})
```

```
except Exception as e:
```

```
print(str(e))
```

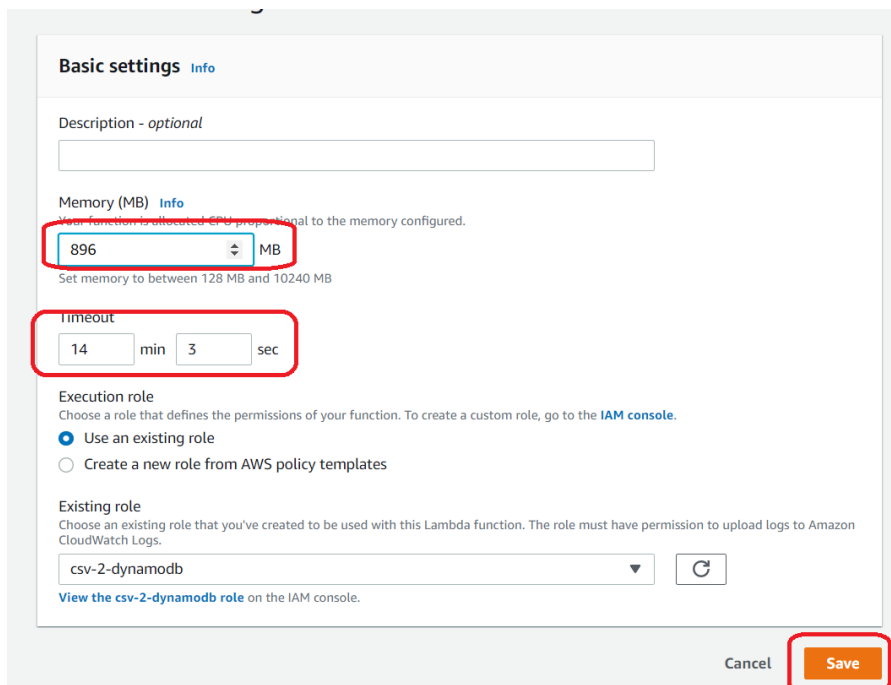
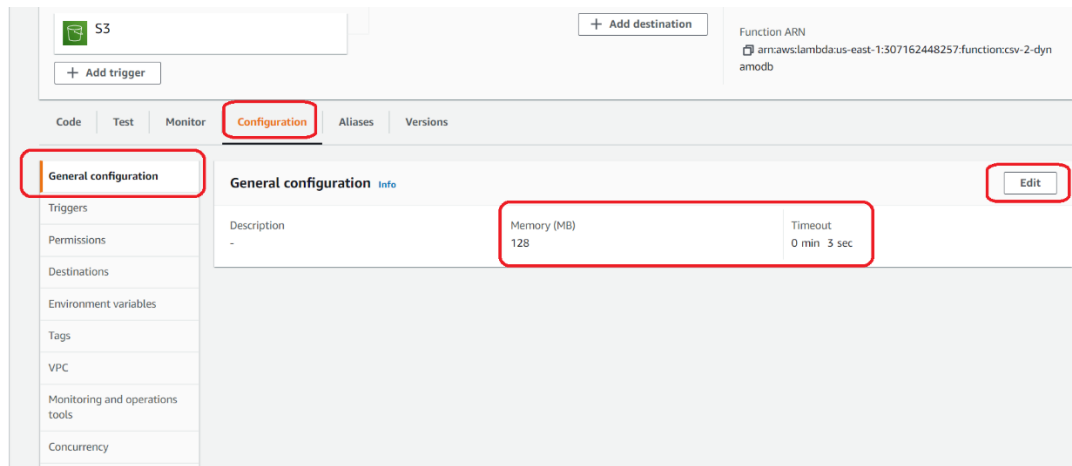
```
return {
```

```
'statusCode': 200,  
  
'body': json.dumps('csv to dynamdb success')  
  
}
```

- Go to the configuration tab > General Configurations > – and Increase the timeout(14min 3 sec:) and memory (896) and “**Deploy**” as shown below

Note : Lambda has a Maximum timeout of **15mins (900 secs)** default is **3 seconds** **Maximum Memory of 10 GB**

Lambda has a **Maximum Memory of 10 GB** default is **128 MB**



7) Upload the crimes data csv file into the bucket and check the dynamodb table

The image shows two screenshots from the AWS console. The top screenshot is the S3 console 'Upload: status' page, indicating a successful upload of a file named 'chicago_crime_dataset.csv' (17.7 MB) to the bucket 's3://democsv2dynamodb'. The bottom screenshot is the AWS DynamoDB console for the 'Chicagocrimes' table, showing a list of 9 items with columns: id, arrest, beat_num, block, case_number, community_cod, date, and date_of_update.

Upload: status

Summary

Destination	Succeeded	Failed
s3://democsv2dynamodb	1 file, 17.7 MB (100.00%)	0 files, 0 B (0%)

Files and folders (1 Total, 17.7 MB)

Name	Folder	Type	Size	Status	Error
chicago_crime_dataset.csv	-	application/vnd.ms-excel	17.7 MB	Succeeded	-

Chicagocrimes

Overview | **Items** | Metrics | Alarms | Capacity | Indexes | Global Tables | Backups | Contributor Insights | Triggers | Access control | Tags

Scan: [Table] Chicagocrimes: id

id	arrest	beat_num	block	case_number	community_cod	date	date_of_update
10508693	TRUE	1022	013XX S SAWYER AVE	HZ250496	29	5/3/2016 23:40	5/10/2016 15:00
10508695	FALSE	313	061XX S DREXEL AVE	HZ250409	42	5/3/2016 21:40	5/10/2016 15:00
10508697	FALSE	1524	053XX W CHICAGO AVE	HZ250503	25	5/3/2016 23:31	5/10/2016 15:00
10508698	FALSE	1532	049XX W FULTON ST	HZ250424	25	5/3/2016 22:10	5/10/2016 15:00
10508699	FALSE	1523	003XX N LOTUS AVE	HZ250455	25	5/3/2016 22:00	5/10/2016 15:00
10508702	FALSE	631	082XX S MARYLAND AVE	HZ250447	44	5/3/2016 22:35	5/10/2016 15:00
10508703	FALSE	133	027XX S STATE ST	HZ250489	35	5/3/2016 22:30	5/10/2016 15:00