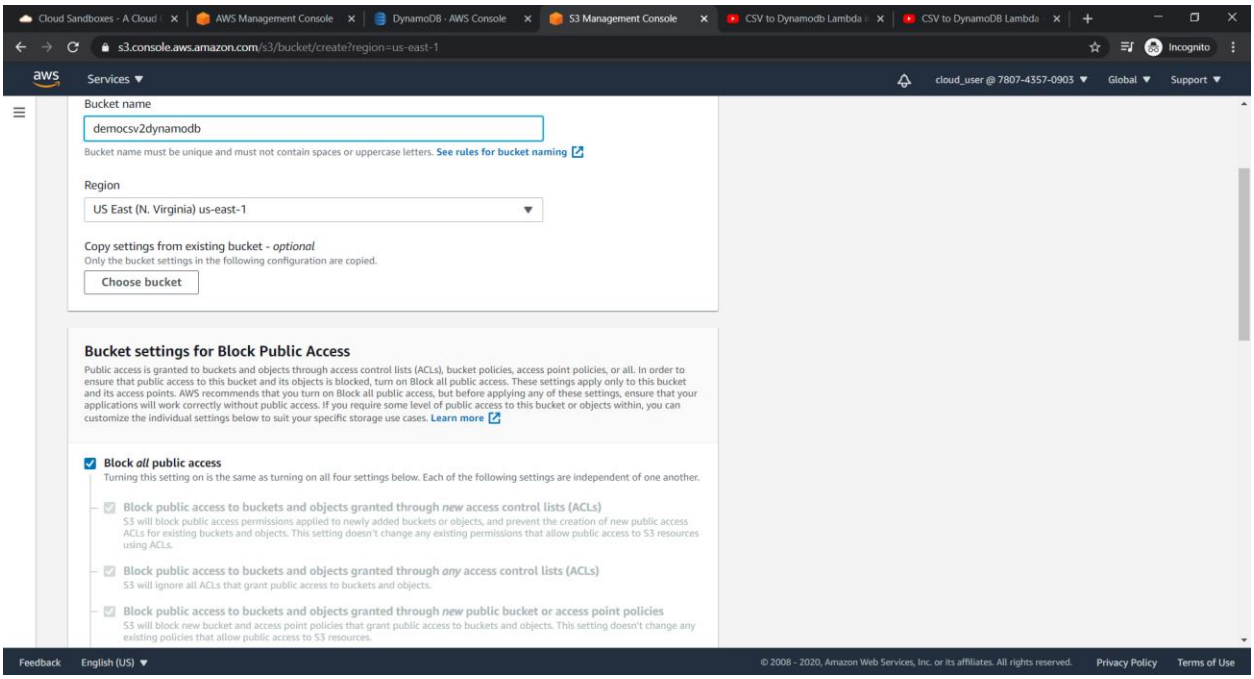
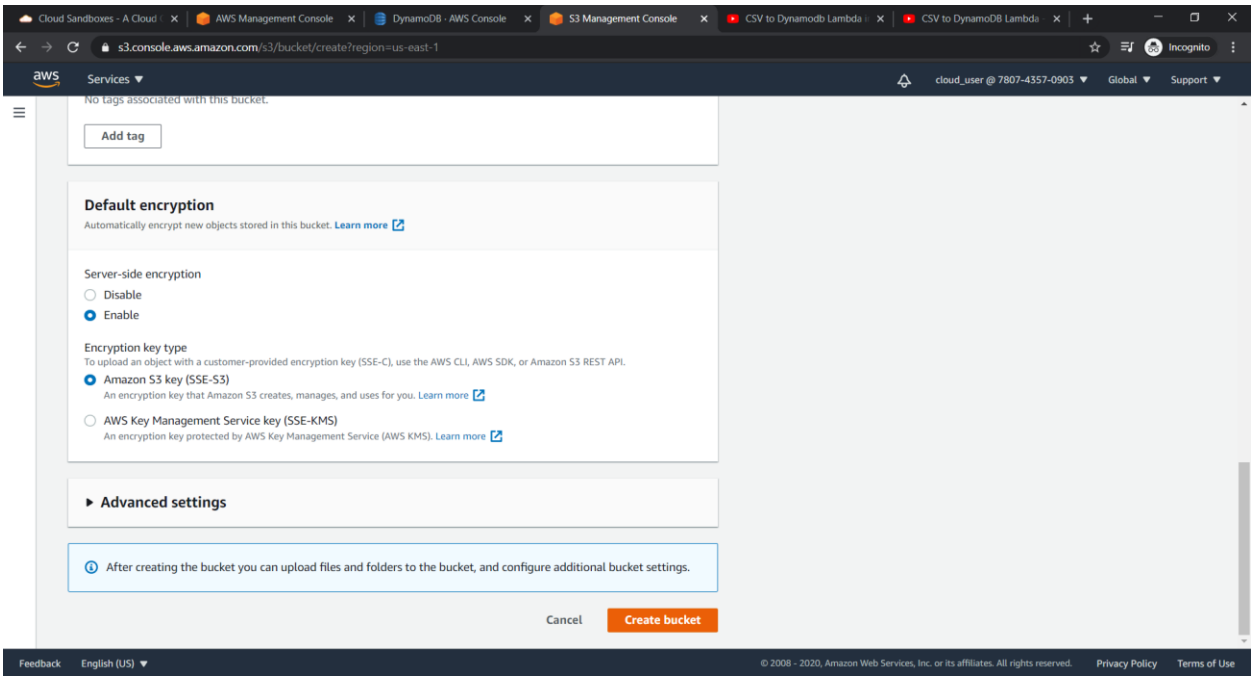
Loading a csv file into dynamodb through Lambda

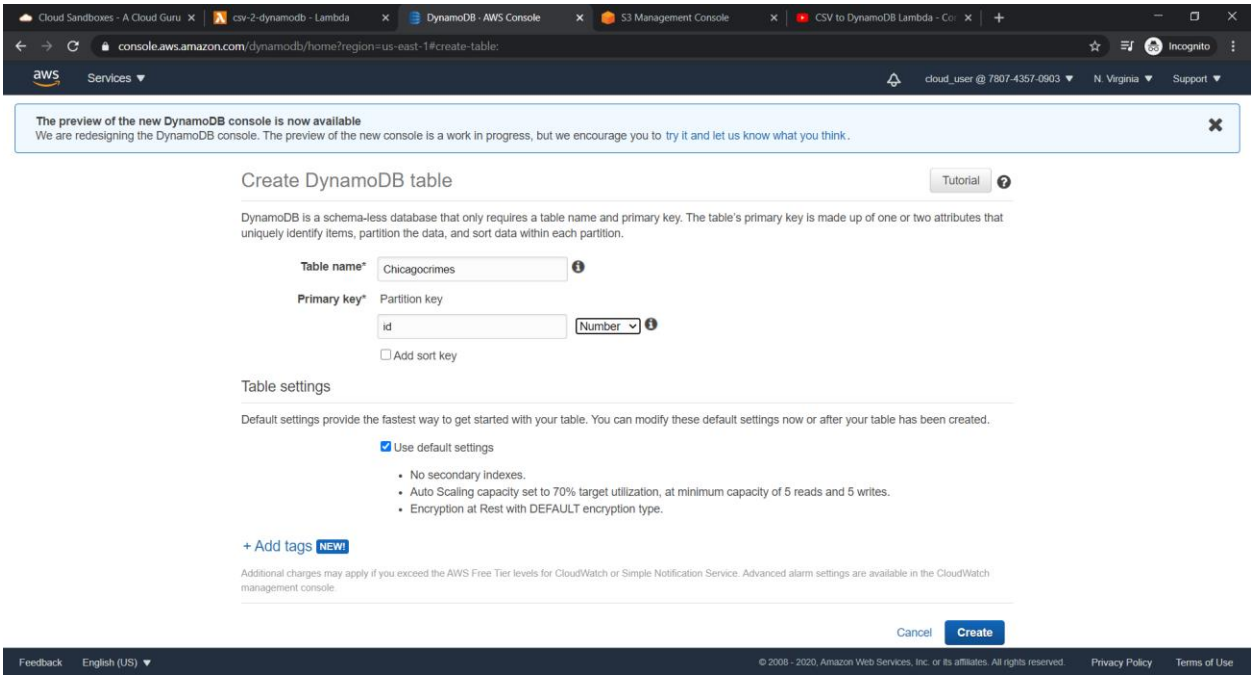
1)Setup an s3 bucket as below

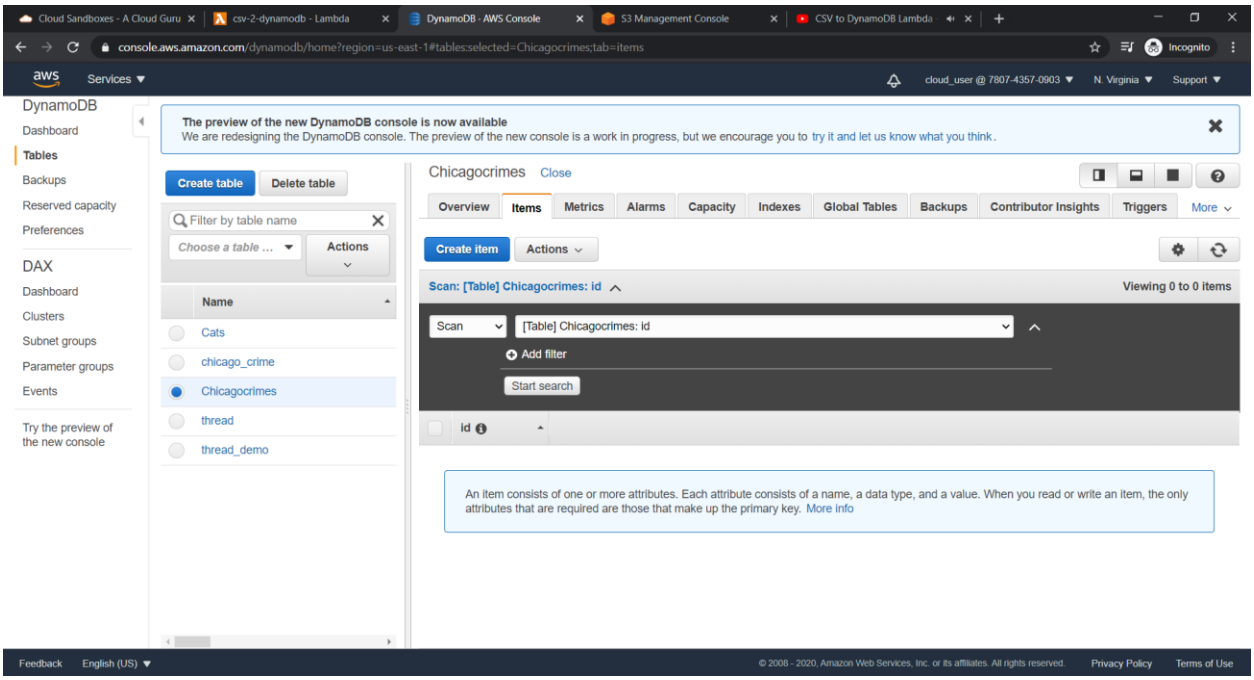




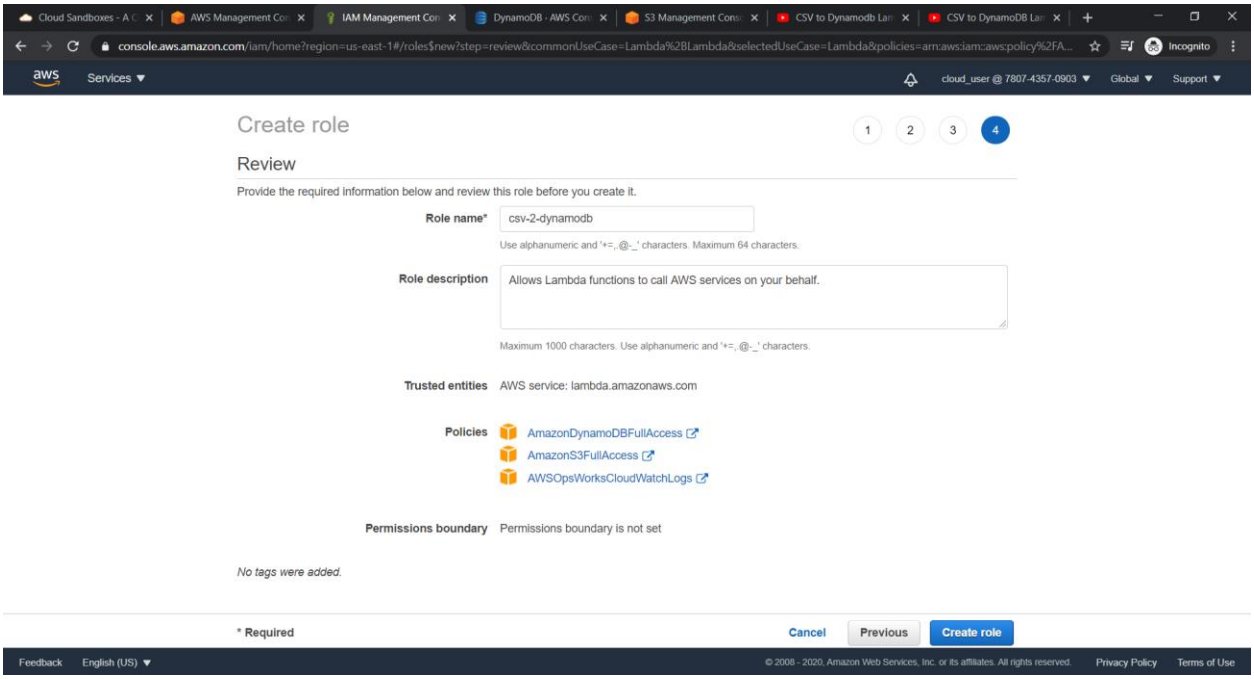
2)Check the crimes data csv file to identify data,primary key and data types

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

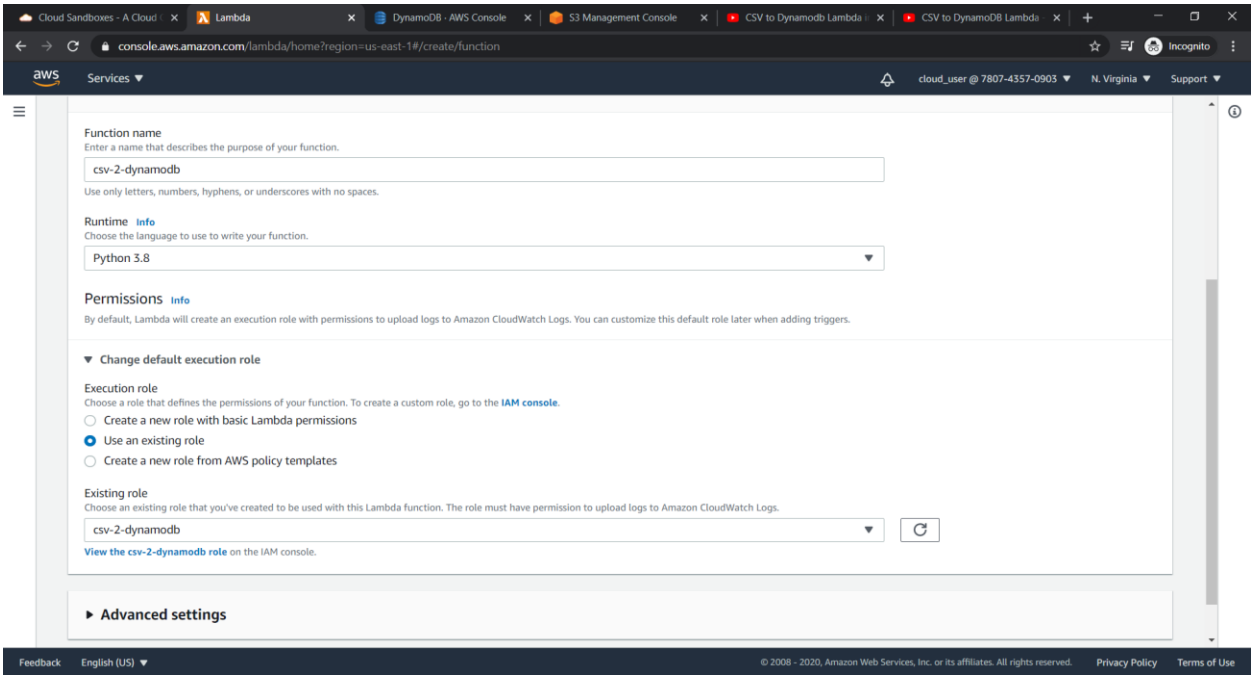




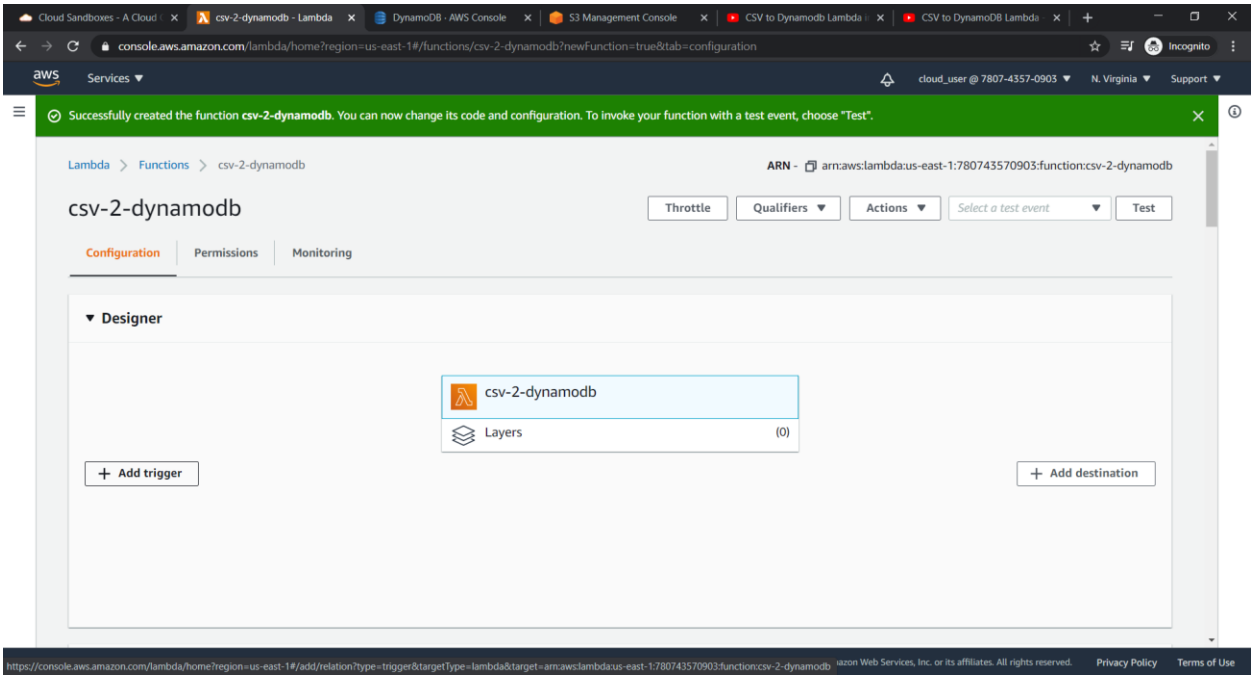
3)Create role for **lambda** to access S3,Dynamodb and CloudWatch for logs

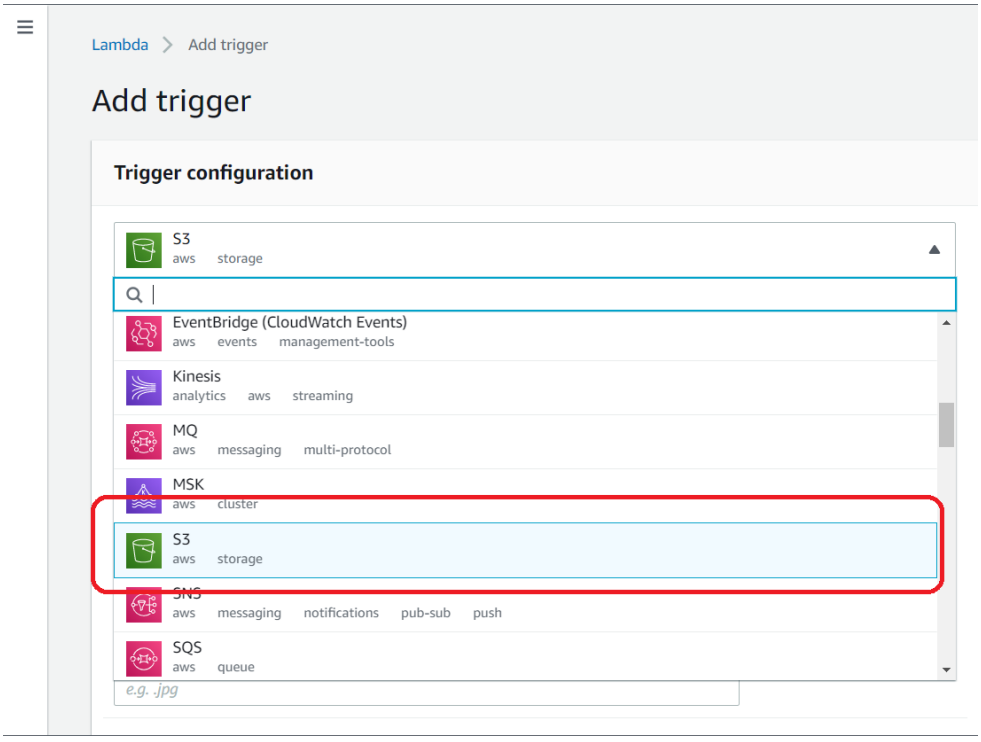


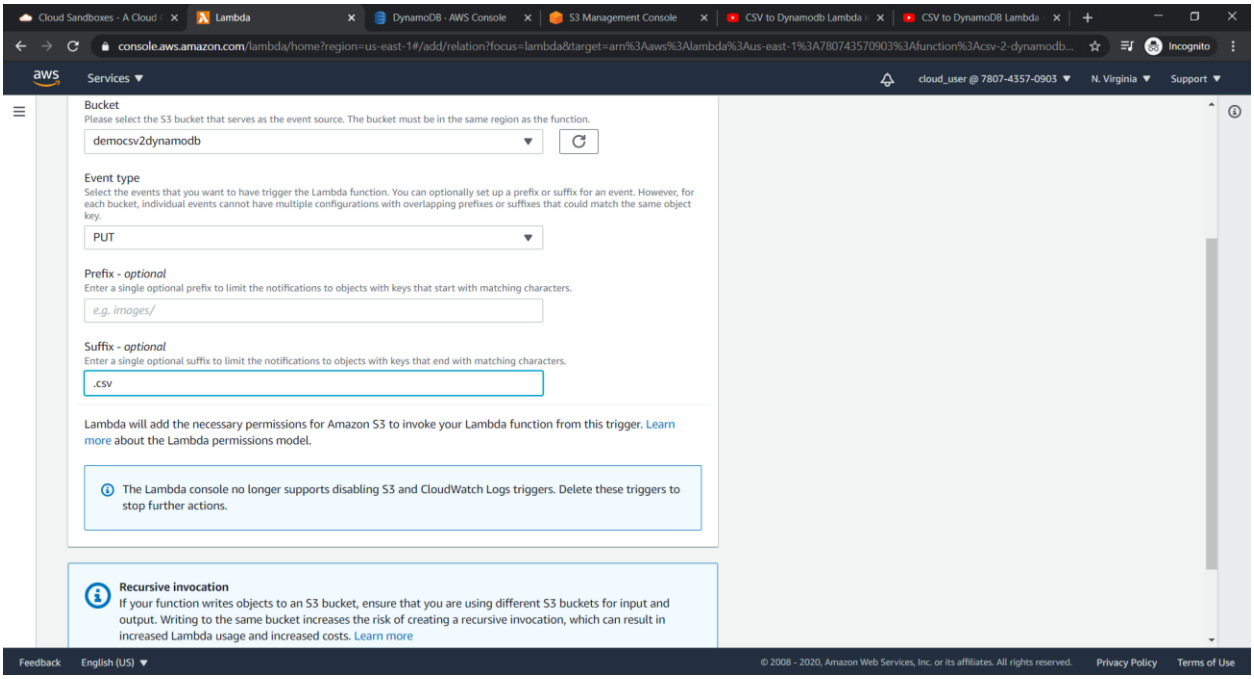
4)Create lambda function

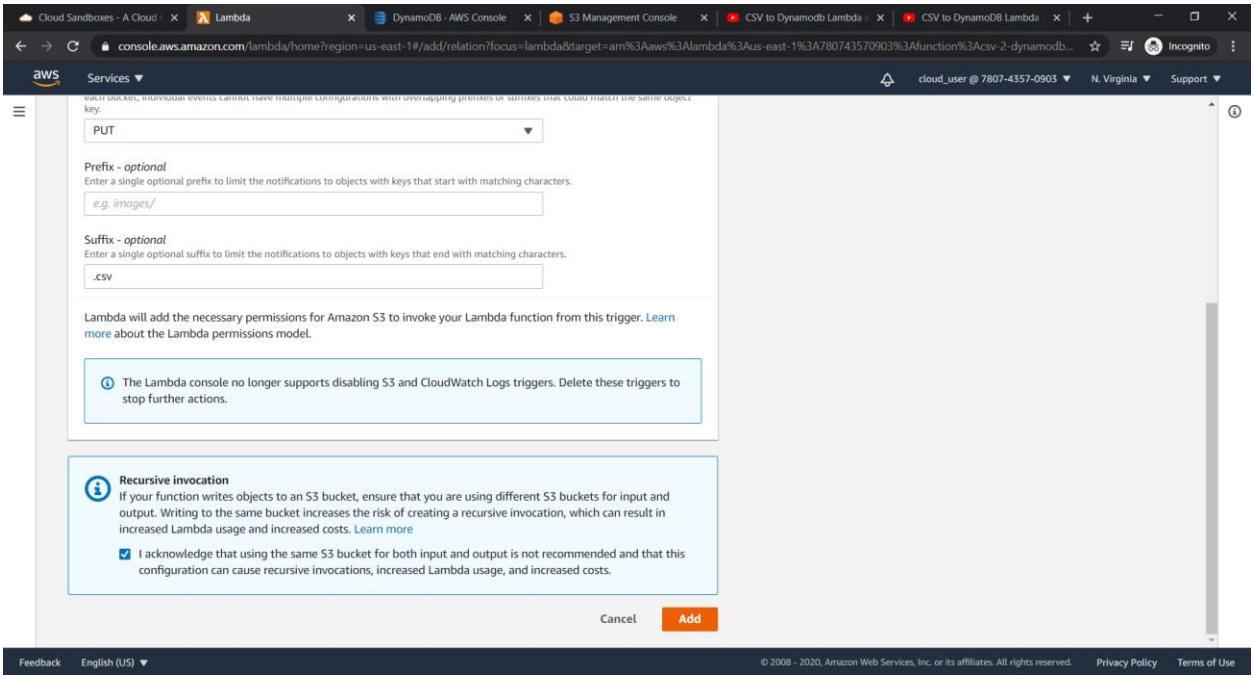


5)Add trigger


Successfully created the function csv-2-dynamodb. You can now change its code and configuration. To invoke your function with a test event, choose "Test".
Lambda > Functions > csv-2-dynamodb
ARN - arn:aws:lambda:us-east-1:780743570903:function:csv-2-dynamodb
csv-2-dynamodb
Throttle
Qualifiers
Actions
Select a test event
Test
Configuration
Permissions
Monitoring
Designer
csv-2-dynamodb
Layers
(0)
+ Add trigger
+ Add destination
Privacy Policy
Terms of Use


Lambda > Add trigger
Add trigger
Trigger configuration
S3
aws storage
EventBridge (CloudWatch Events)
aws events management-tools
Kinesis
analytics aws streaming
MQ
aws messaging multi-protocol
MSK
aws cluster
S3
aws storage
SNS
aws messaging notifications pub-sub push
SQS
aws queue
e.g. .jpg





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.

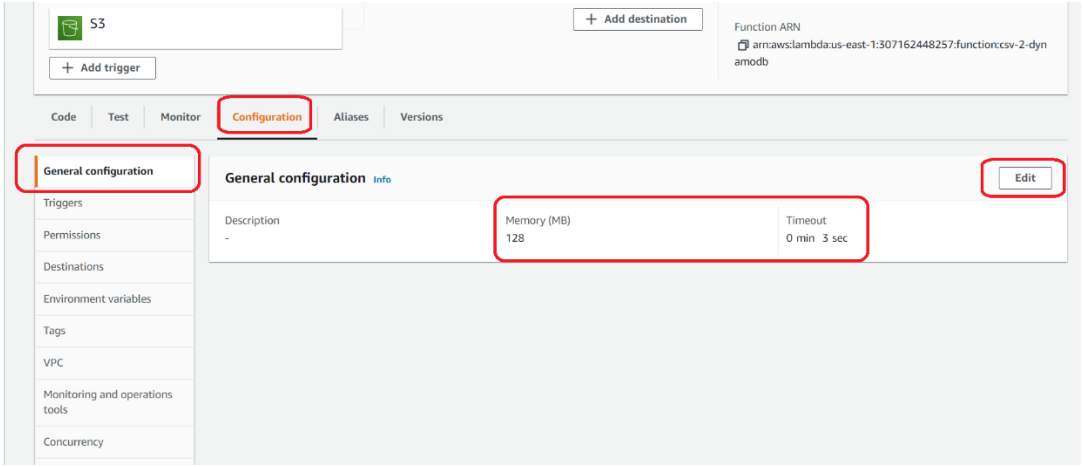
```
import json
import csv
import boto3

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]
```

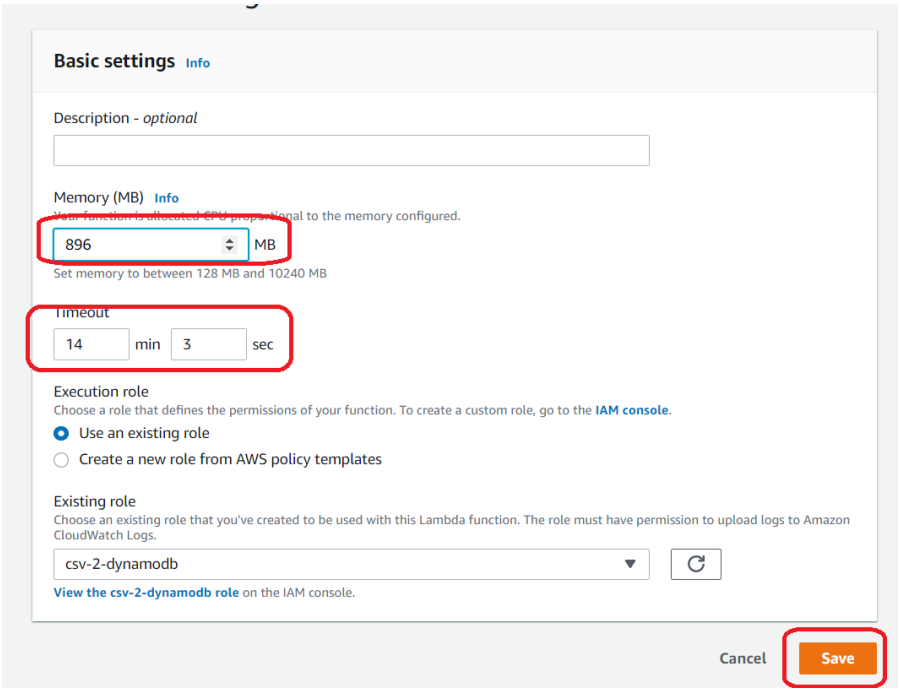
```
        location_desc=row[5]
        arrest=row[6]
        domestic=row[7]
        beat_num=row[8]
        district_code=row[9]
        ward_no=row[10]
        community_code=row[11]
        fbi_code=row[12]
        x_coordinate=row[13]
        y_coordinate=row[14]
        year=row[15]
        date_of_update=row[16]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