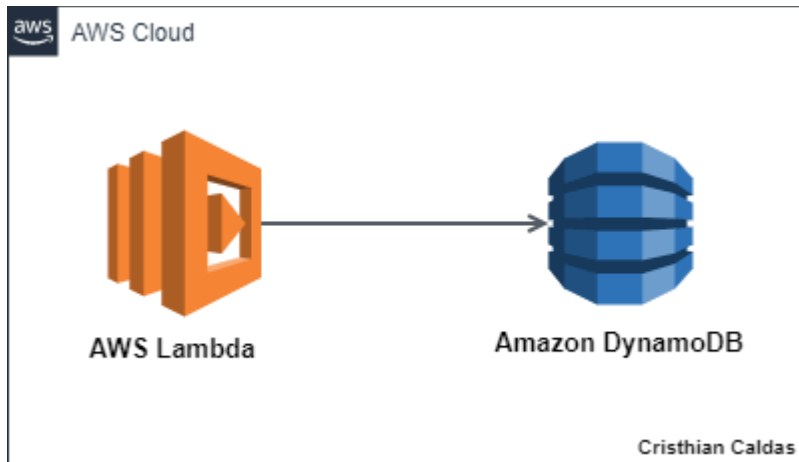


Creando un Backend Serverless

En este taller vamos a utilizar **AWS Lambda** y **Amazon DynamoDB** para crear nuestro backend sin servidor.



Uno de las características de [AWS Lambda](#) es la facilidad con la que se puede integrar con otros servicios de **AWS**.

En este artículo aprenderemos a crear una tabla en **DynamoDB** e implementar una función **Lambda** que nos va a permitir obtener registros de nuestra tabla creada en **DynamoDB**.

A continuación se muestra las tareas que se van a realizar :

- Crear una tabla de **Amazon DynamoDB**.
- Crear un rol de **IAM** para la función Lambda.
- Crear una función **Lambda** para administrar las solicitudes.
- Probar la implementación de nuestra función.

Paso 01 .- Crear una tabla de Amazon DynamoDB.

- **Elija Services > DynamoDB.**
- Seleccionar **Create Table.**
- **Table Name:** Customer
- **Primary Key :** customer_id
- **Table settings :** Seleccione la casilla **Use default settings** y elija **Create.**

Create DynamoDB table

Tutorial ?

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* ⓘ

Primary key* Partition key

ⓘ

☐ 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.
 - Provisioned capacity set to 5 reads and 5 writes.
 - Basic alarms with 80% upper threshold using SNS topic "dynamodb".
 - Encryption at Rest with DEFAULT encryption type.

Desplácese hasta la parte inferior de la sección **Overview** de la nueva tabla y anote el **ARN**. Lo usaremos luego a la hora de crear nuestro Rol.

Overview	Items	Metrics	Alarms	Capacity	Indexes	Global Tables	Backups	Contributor Insights
Table name			Customer					
Primary partition key			customer_id (String)					
Last decrease time			-					
Last increase time			-					
Storage size (in bytes)			0 bytes					
Item count			0 Manage live count					
Region			US East (N. Virginia)					
Amazon Resource Name (ARN)			arn:aws:dynamodb:us-east-1:818929933852:table/Customer					

- Ahora vamos a registrar información en nuestra Tabla.
- Crearemos un archivo **json** con el siguiente esquema:

```
{
  "Customer": [
    {
      "PutRequest": {
        "Item": {
          "customer_id": {
            "S": "5113139"
          },
          "Edad": {
```

```
      "S": "56"
    },
    "Nombre": {
      "S": "Alika Mueller"
    },
    "Pais": {
      "S": "Denmark"
    }
  }
},
{
  "PutRequest": {
    "Item": {
      "customer_id": {
        "S": "25726607"
      },
      "Edad": {
        "S": "56"
      },
      "Nombre": {
        "S": "Michelle Velez"
      },
      "Pais": {
        "S": "Falkland Islands"
      }
    }
  }
},
```

```
{
  "PutRequest": {
    "Item": {
      "customer_id": {
        "S": "33604047"
      },
      "Edad": {
        "S": "56"
      },
      "Nombre": {
        "S": "Xyla Lamb"
      },
      "Pais": {
        "S": "Saint Kitts and Nevis"
      }
    }
  }
},
```

```
{
  "PutRequest": {
    "Item": {
      "customer_id": {
        "S": "50849423"
      },
      "Edad": {
        "S": "56"
      },
      "Nombre": {
        "S": "Taylor Howard"
      }
    }
  }
},
```

```
    },
    "Pais": {
      "S": "Guernsey"
    }
  }
}
},
{
  "PutRequest": {
    "Item": {
      "customer_id": {
        "S": "42848820"
      },
      "Edad": {
        "S": "56"
      },
      "Nombre": {
        "S": "Quynn Kline"
      },
      "Pais": {
        "S": "Anguilla"
      }
    }
  }
},
{
  "PutRequest": {
    "Item": {
      "customer_id": {
```

```
        "S": "18270892"
      },
      "Edad": {
        "S": "56"
      },
      "Nombre": {
        "S": "Lisandra Benjamin"
      },
      "Pais": {
        "S": "Curacao"
      }
    }
  },
  {
    "PutRequest": {
      "Item": {
        "customer_id": {
          "S": "29997643"
        },
        "Edad": {
          "S": "56"
        },
        "Nombre": {
          "S": "Geraldine Silva"
        },
        "Pais": {
          "S": "Pakistan"
        }
      }
    }
  }
}
```

```
    }  
  }  
  },  
{  
  "PutRequest": {  
    "Item": {  
      "customer_id": {  
        "S": "47201639"  
      },  
      "Edad": {  
        "S": "56"  
      },  
      "Nombre": {  
        "S": "Ila Burnett"  
      },  
      "Pais": {  
        "S": "Saint Helena, Ascension"  
      }  
    }  
  }  
},  
{
```

```
{  
  "PutRequest": {  
    "Item": {  
      "customer_id": {  
        "S": "47288747"  
      },  
      "Edad": {  
        "S": "56"
```

```
    },
    "Nombre": {
      "S": "Lillian Mcdaniel"
    },
    "Pais": {
      "S": "Angola"
    }
  }
}

    },
{
  "PutRequest": {
    "Item": {
      "customer_id": {
        "S": "37939202"
      },
      "Edad": {
        "S": "56"
      },
      "Nombre": {
        "S": "Summer Jordan"
      },
      "Pais": {
        "S": "Kyrgyzstan"
      }
    }
  }
}
```

```
]
```


}

- Mediante el siguiente comando vamos a registrar toda la información del archivo en nuestra tabla.(previamente a ver instalado y configurado su [aws cli](#))
- **aws dynamodb batch-write-item — request-items file://customer-data.json**

le.aws.amazon.com/dynamodb/home/region-us-east-1#tables:selected=CustomerTable:Items

Search for services, features, marketplace products, and docs [AWS]

Customer **Close**

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

Create item Actions

Scan: [Table] Customer: customer_id

Scan [Table] Customer: customer_id

Add filter

Start search

customer_id	Edad	Nombre	País
13088979	33	Brian Hood	Kuwait
14410050	56	Arifha Buchanan	Bangladesh
14539403	56	Willa Boone	Bhutan
17276058	67	Wynne Jennings	Romania
18054308	59	Lana Bartlett	Montserrat
18270892	58	Liandra Benjamin	Curacao
18925522	56	Charlotte Hodges	Mexico
23120657	66	Aurora Howard	Oman
23177742	22	Barbara Larsen	Tajikistan
23495622	56	Basia Powell	Canada
25726607	56	Michelle Velez	Falkland Islands
26113080	56	Vesta Brennan	Northern Mariana Islands
29997643	56	Geasline Silva	Pakistan
30451671	56	Kyra Perkins	Wallis and Futuna

© 2007 - 2020 Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Paso 02 .- Crear un rol de IAM para su función Lambda.

- Elija **Services > IAM**
- Seleccionar **Roles>Create rol**
- Seleccionar **Lambda>Next:Permissions**
- **Filter policies : AWSLambdaBasicExecutionRole**

Create policy

Filter policies Showing 34 results

	Policy name	Used as
<input checked="" type="checkbox"/>	AWSLambdaBasicExecutionRole	Permissions policy (1)
<input type="checkbox"/>	AWSLambdaBasicExecutionRole-00ae680f-0e42-4233-88db-216b1cb09523	Permissions policy (1)
<input type="checkbox"/>	AWSLambdaBasicExecutionRole-01637322-6fcb-4b65-97fb-93a3477d8106	Permissions policy (1)
<input type="checkbox"/>	AWSLambdaBasicExecutionRole-0add4924-6f3d-45a7-a685-4f1cfde95f12	Permissions policy (1)
<input type="checkbox"/>	AWSLambdaBasicExecutionRole-197b9347-0b60-4c71-9ad3-47e98780e885	Permissions policy (1)
<input type="checkbox"/>	AWSLambdaBasicExecutionRole-1a6fba4f-0090-4bca-bf95-ae5506ecc633	Permissions policy (1)
<input type="checkbox"/>	AWSLambdaBasicExecutionRole-1c16cc63-1827-4dd4-a057-14aaae41b417	Permissions policy (1)
<input type="checkbox"/>	AWSLambdaBasicExecutionRole-1fad224c-1e6f-421a-941b-9948f2eabe52	Permissions policy (1)

* Required

Cancel Previous **Next: Tags**

- Seleccionar **Next: Tags >Next: Review**
- **Role name:** CustomerLambda
- Seleccionar **Create role.**
- Escriba CustomerLambda en el cuadro de filtro de la pagina de Rol.
- Seleccione el rol y añada **Add inline policy**
- Seleccione **Service** : DynamoDB
- En **Actions** : **Query**
- En **Resources >Specific > Add ARN** y añadir el **ARN** de la tabla creada anteriormente.
- click en **Review policy.**
- **Name** : DynamoDBReadAccess
- **Create policy.**

▼ DynamoDB (1 action) Clone Remove

► Service DynamoDB

► Actions Write
PutItem

▼ Resources close ☒ Specific ☐ All resources

table EDIT Any in this account

[Add ARN to restrict access](#)

► Request conditions [Specify request conditions \(optional\)](#)

[Add additional permissions](#)

Character count: 175 of 10,240. This number character count includes character for all inline policies in this role: CustomerLambda. Cancel Review policy

Paso 03 .- Crear una función Lambda.

- Elija **Services > Lambda**
- **Create Function**
- **Function name** : RequestCustomer
- **Runtime** : Python 3.8
- **Permissions > Change default execution role**
- **Use an existing role > CustomerLambda > Create function**
- Desplácese hasta la sección **Function code** y reemplace el código por el siguiente:

```
import json
```

```

import os

import boto3

from boto3.dynamodb.conditions import Key

def lambda_handler(event, context):

    try:

        customer_id=event['queryStringParameters']['customer_id']

        person= list_customerxID(customer_id)

        return {

            'statusCode': 200,

            'body': person

        }

    except Exception as e:

        print("error en el metodo lambda_handler".format(e))

```

```

def _get_dynamo_table(table_name):

    try:

        dynamo_resource = boto3.resource('dynamodb',
region_name=os.environ.get('DEPLOY_REGION'))

        table = dynamo_resource.Table(table_name)

        return table

    except Exception as e:

        print("error el metodo _get_dynamo_table".format(e))

```

#listando customer x ID

```

def list_customerxID(customer_id):

    try:

```

```

table_name = os.environ.get('DYNAMO_TABLE_NAME')

table = _get_dynamo_table(table_name)

response = table.query(

    KeyConditionExpression=Key('customer_id').eq(customer_id)

)

item = response['Items']

return item

except Exception as e:

    print("error en el metodo list_customerxID".format(e))

```

- Luego en la parte **Environment variables** añadir las siguientes 02 variables que se van a utilizar en la función:
- **DEPLOY_REGION** : us-east-1
- **DYNAMO_TABLE_NAME** : Customer

Paso 04 .- Probar la implementación de nuestra Función

- Ahora vamos a implementar un test para nuestra función.
- En la pantalla principal seleccionar **Configure test events**
- Seleccionar **Create new test event**
- **Event Name** : DemoTest
- Copie y pegue el siguiente evento de prueba en el editor:

```

{
  "body": "eyJ0ZXN0IjoYm9keSJ9",
  "resource": "/{proxy+}",
  "path": "/path/to/resource",
  "httpMethod": "POST",
  "isBase64Encoded": true,
  "queryStringParameters": {
    "customer_id": "25726607"
  },
  "multiValueQueryStringParameters": {

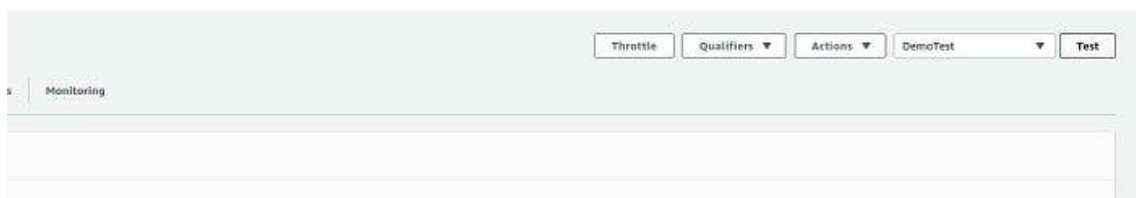
```

```

"foo": [
  "bar"
],
"pathParameters": {
  "proxy": "/path/to/resource"
},
"headers": {
  "Accept":
"text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8",
  "X-Forwarded-For": "127.0.0.1, 127.0.0.2",
  "X-Forwarded-Port": "443",
  "X-Forwarded-Proto": "https"
}
}

```

- Hacer click en **Create**.
- Haga clic en Test con la opción DemoTest seleccionada en el menú desplegable.



- Desplácese hacia la parte superior de la página y amplíe la sección **Details** de la sección **Execution result**.
- Compruebe que la ejecución del test se ha realizado correctamente y que el resultado de la función sea similar al que se muestra en la figura:

RequestCustomer

[Throttle](#)[Qualifiers](#)[Actions](#)[DemoTest](#)[Test](#)

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

▼ Details

The area below shows the result returned by your function execution. [Learn more](#) about returning results from your function.

```
{
  "statusCode": 200,
  "body": {
    {
      "customer_id": "18299",
      "Nombre": "Jose Garcia",
      "Edad": "32",
      "Sexo": "Masculino"
    }
  }
}
```

Summary

Code SHA-256

4fj1dxn0lmqk5Xcd5Ym3wVw7h0umyMf5/TbGnysuI

Duration

212.00 ms

Resources configured

128 MB

Request ID

aba7d773-57ec-4769-bbd2-59682775058

Billed duration

215 ms

Max memory used

75 MB

Log output

The section below shows the logging calls in your code. These correspond to a single row within the CloudWatch log group corresponding to this Lambda function. [Click here](#) to view the CloudWatch log group.

START RequestID: aba7d773-57ec-4769-bbd2-59682775058 Version: \$LATEST

END RequestID: aba7d773-57ec-4769-bbd2-59682775058

REPORT RequestID: aba7d773-57ec-4769-bbd2-59682775058 Duration: 212.00 ms Billed Duration: 215 ms Memory Size: 128 MB Max Memory Used: 75 MB