

## **Api GW + lambdas (CRUD) + DB + Serverless**

### **Estudiantes:**

Johan Calixto Ramírez Acosta

Gian Lucas Figueroa Felizola

Leandro Alfonso Maldonado Ospino

### **Docente:**

Salomón Segundo de la hoz

### **Electiva:**

Cloud Computing

Facultad de Ingenierías y Tecnológicas

Universidad Popular del Cesar

Valledupar – Cesar

2022

## Api GW + lambdas (CRUD) + DB + Serverless

En AWS es importante crear un usuario en Identity and Access Management (IAM)

Identity and Access Management (IAM)

Panel

▼ Administración del acceso

- Grupos de usuarios
- Usuarios**
- Roles
- Políticas
- Proveedores de identidad
- Configuración de cuenta

IAM > Usuarios

**Usuarios (1)** Información

Un usuario de IAM es una identidad con credenciales válidas a largo plazo que se utiliza para interactuar con AWS en una cuenta.

Eliminar Agregar usuarios

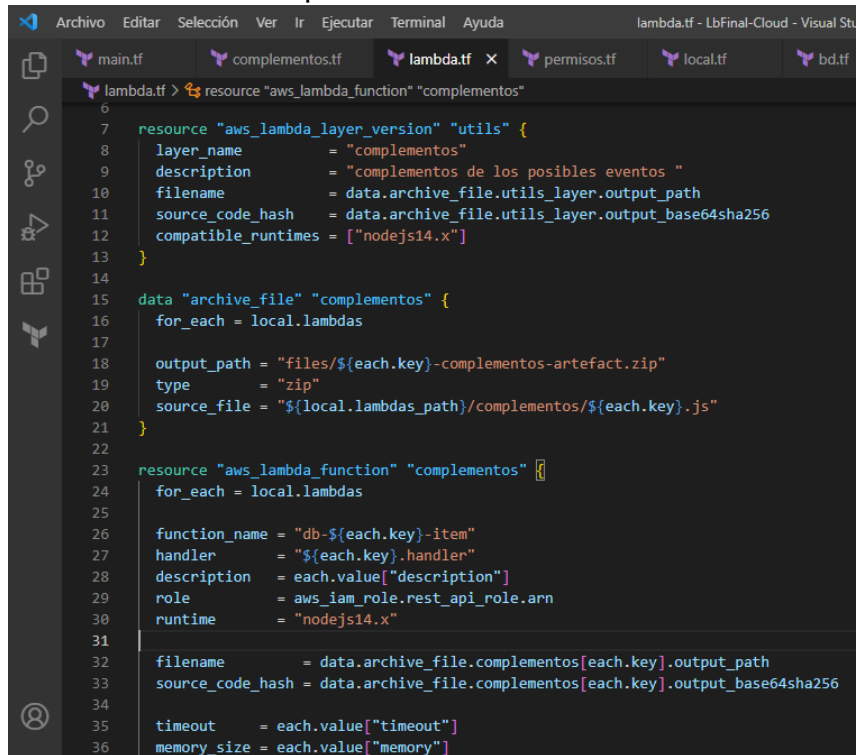
Buscar usuarios por nombre de usuario o clave de acceso

	Nombre de usuario	Grupos	Última activi...	MFA	Antigüedad de la contra
<input type="checkbox"/>	Jack	Ninguno	✓ hace 3 minutos	Ninguno	Ninguno

Escribimos el siguiente código en Visual Code

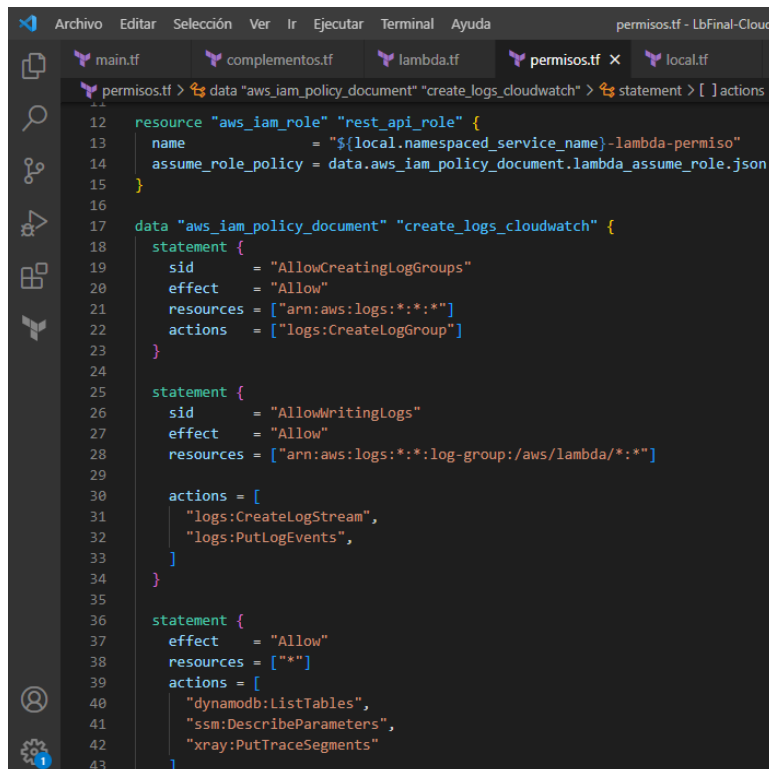
```
main.tf > ...
1
2 provider "aws" {
3   region = var.aws_region
4   profile = var.aws_profile
5
6   default_tags {
7     tags = {
8       Project = "Rest Api GW"
9       CreatedAt = "2022-12-08"
10      ManagedBy = "Terraform"
11      Owner = "Johan Ramirez"
12      Env = var.env
13    }
14  }
15 }
16
```

## Creamos el archivo que tiene informacion de las lambdas



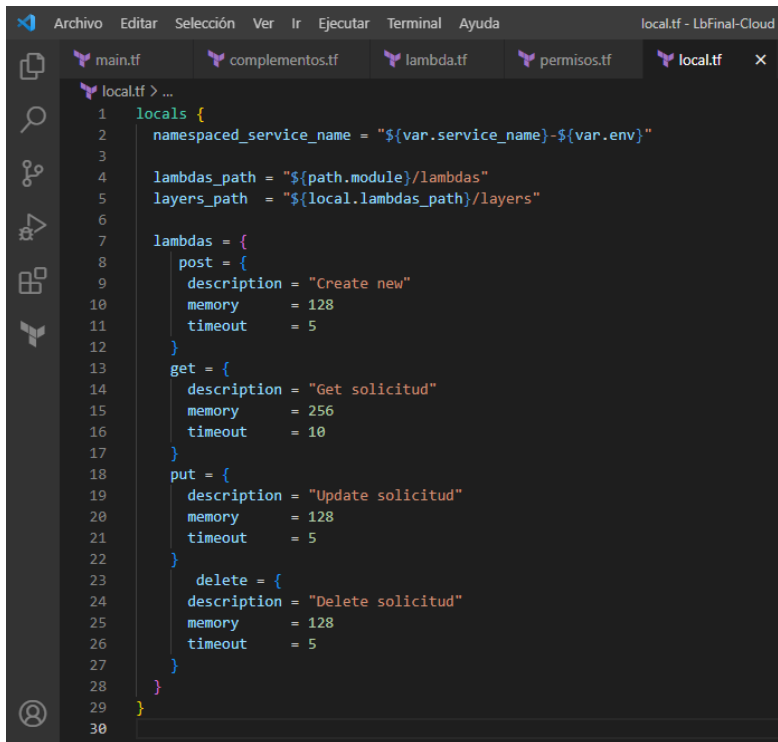
```
1  resource "aws_lambda_function" "complementos"
2
3
4
5
6
7  resource "aws_lambda_layer_version" "utils" {
8    layer_name       = "complementos"
9    description      = "complementos de los posibles eventos "
10   filename         = data.archive_file.utils_layer.output_path
11   source_code_hash  = data.archive_file.utils_layer.output_base64sha256
12   compatible_runtimes = ["nodejs14.x"]
13 }
14
15 data "archive_file" "complementos" {
16   for_each = local.lambdas
17
18   output_path = "files/${each.key}-complementos-artefact.zip"
19   type        = "zip"
20   source_file = "${local.lambdas_path}/complementos/${each.key}.js"
21 }
22
23 resource "aws_lambda_function" "complementos" {
24   for_each = local.lambdas
25
26   function_name = "db-${each.key}-item"
27   handler       = "${each.key}.handler"
28   description   = each.value["description"]
29   role          = aws_iam_role.rest_api_role.arn
30   runtime       = "nodejs14.x"
31
32   filename         = data.archive_file.complementos[each.key].output_path
33   source_code_hash = data.archive_file.complementos[each.key].output_base64sha256
34
35   timeout        = each.value["timeout"]
36   memory_size    = each.value["memory"]
37 }
```

## Definimos todos los permisos que utilizaremos



```
1  data "aws_iam_policy_document" "create_logs_cloudwatch" {
2    statement {
3      actions = ["logs:CreateLogGroup"]
4      effect   = "Allow"
5      resources = ["arn:aws:logs:*:*:*"]
6    }
7  }
8
9  resource "aws_iam_role" "rest_api_role" {
10   name               = "${local.namespaced_service_name}-lambda-permiso"
11   assume_role_policy = data.aws_iam_policy_document.lambda_assume_role.json
12 }
13
14 data "aws_iam_policy_document" "create_logs_cloudwatch" {
15   statement {
16     sid      = "AllowCreatingLogGroups"
17     effect   = "Allow"
18     resources = ["arn:aws:logs:*:*:*"]
19     actions  = ["logs:CreateLogGroup"]
20   }
21
22   statement {
23     sid      = "AllowWritingLogs"
24     effect   = "Allow"
25     resources = ["arn:aws:logs:*:*:log-group:/aws/lambda/*:*"]
26     actions  = [
27       "logs:CreateLogStream",
28       "logs:PutLogEvents",
29     ]
30   }
31
32   statement {
33     effect   = "Allow"
34     resources = ["*"]
35     actions  = [
36       "dynamodb:ListTables",
37       "ssm:DescribeParameters",
38       "xray:PutTraceSegments"
39     ]
40   }
41 }
42
43 }
```

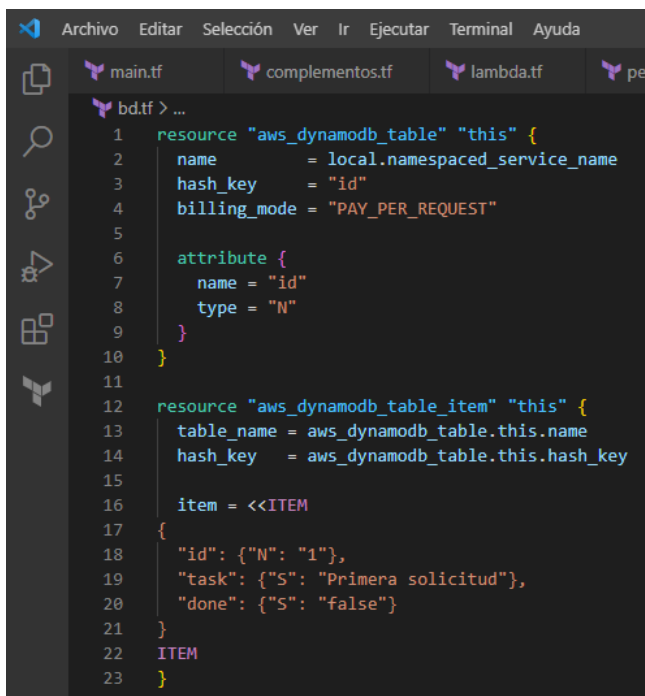
Creamos las solicitudes que se trabajan de manera local



The screenshot shows a code editor with a dark theme. The top menu bar includes 'Archivo', 'Editar', 'Selección', 'Ver', 'Ir', 'Ejecutar', 'Terminal', and 'Ayuda'. The title bar indicates the file is 'local.tf' in a workspace named 'LbFinal-Cloud'. The editor has several tabs open: 'main.tf', 'complementos.tf', 'lambda.tf', 'permisos.tf', and 'local.tf'. The 'local.tf' tab is active, showing a Terraform configuration. The configuration defines a 'locals' block with variables for 'namespaced\_service\_name', 'lambdas\_path', and 'layers\_path'. It also defines a 'lambdas' resource with four functions: 'post', 'get', 'put', and 'delete'. Each function has a 'description', 'memory', and 'timeout' attribute.

```
1 locals {
2   namespaced_service_name = "${var.service_name}-${var.env}"
3
4   lambdas_path = "${path.module}/lambdas"
5   layers_path = "${local.lambdas_path}/layers"
6
7   lambdas = {
8     post = {
9       description = "Create new"
10      memory      = 128
11      timeout     = 5
12    }
13    get = {
14      description = "Get solicitud"
15      memory      = 256
16      timeout     = 10
17    }
18    put = {
19      description = "Update solicitud"
20      memory      = 128
21      timeout     = 5
22    }
23    delete = {
24      description = "Delete solicitud"
25      memory      = 128
26      timeout     = 5
27    }
28  }
29 }
30
```

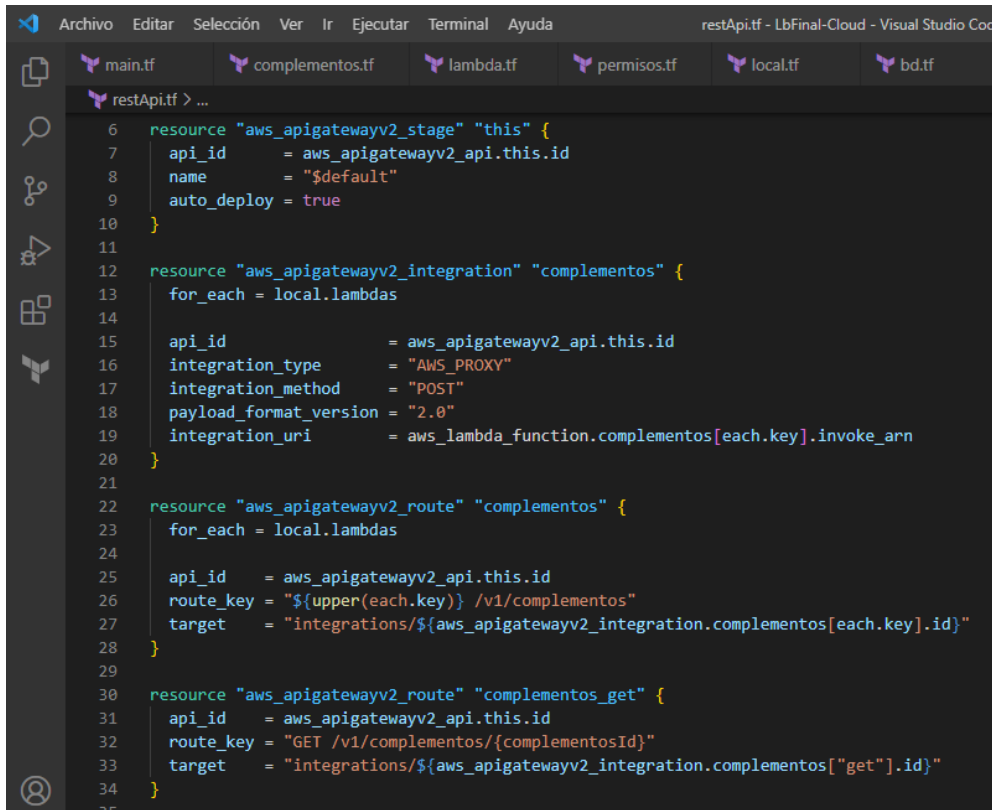
Archivo para la creación de la base de datos



The screenshot shows a code editor with a dark theme. The top menu bar includes 'Archivo', 'Editar', 'Selección', 'Ver', 'Ir', 'Ejecutar', 'Terminal', and 'Ayuda'. The title bar indicates the file is 'bd.tf' in a workspace named 'LbFinal-Cloud'. The editor has several tabs open: 'main.tf', 'complementos.tf', 'lambda.tf', 'permisos.tf', and 'bd.tf'. The 'bd.tf' tab is active, showing a Terraform configuration. The configuration defines two resources: 'aws\_dynamodb\_table' and 'aws\_dynamodb\_table\_item'. The 'aws\_dynamodb\_table' resource has attributes for 'name', 'hash\_key', and 'billing\_mode'. The 'aws\_dynamodb\_table\_item' resource has attributes for 'table\_name', 'hash\_key', and 'item'.

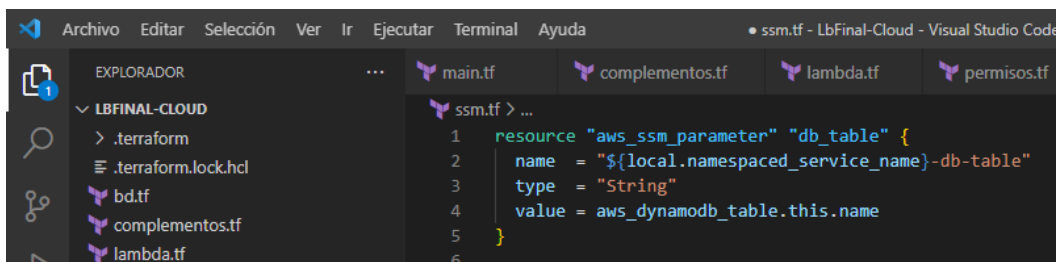
```
1 resource "aws_dynamodb_table" "this" {
2   name       = local.namespaced_service_name
3   hash_key   = "id"
4   billing_mode = "PAY_PER_REQUEST"
5
6   attribute {
7     name = "id"
8     type = "N"
9   }
10 }
11
12 resource "aws_dynamodb_table_item" "this" {
13   table_name = aws_dynamodb_table.this.name
14   hash_key   = aws_dynamodb_table.this.hash_key
15
16   item = <<ITEM
17   {
18     "id": {"N": "1"},
19     "task": {"S": "Primera solicitud"},
20     "done": {"S": "false"}
21   }
22   ITEM
23 }
24
```

Definimos la configuración del archivo el cual va a completar e integral los servicios



```
6 resource "aws_apigatewayv2_stage" "this" {
7   api_id      = aws_apigatewayv2_api.this.id
8   name        = "$default"
9   auto_deploy = true
10 }
11
12 resource "aws_apigatewayv2_integration" "complementos" {
13   for_each = local.lambdas
14
15   api_id            = aws_apigatewayv2_api.this.id
16   integration_type  = "AWS_PROXY"
17   integration_method = "POST"
18   payload_format_version = "2.0"
19   integration_uri    = aws_lambda_function.complementos[each.key].invoke_arn
20 }
21
22 resource "aws_apigatewayv2_route" "complementos" {
23   for_each = local.lambdas
24
25   api_id      = aws_apigatewayv2_api.this.id
26   route_key   = "${upper(each.key)} /v1/complementos"
27   target       = "integrations/${aws_apigatewayv2_integration.complementos[each.key].id}"
28 }
29
30 resource "aws_apigatewayv2_route" "complementos_get" {
31   api_id      = aws_apigatewayv2_api.this.id
32   route_key   = "GET /v1/complementos/{complementosId}"
33   target       = "integrations/${aws_apigatewayv2_integration.complementos["get"].id}"
34 }
35 }
```

Creamos un archivo para la configuración del Parameter Store



```
1 resource "aws_ssm_parameter" "db_table" {
2   name = "${local.namespaced_service_name}-db-table"
3   type = "String"
4   value = aws_dynamodb_table.this.name
5 }
6
```

Inicializamos el directorio de trabajo que contiene la configuración de Terraform

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL

PS C:\Users\johan\OneDrive\Escritorio\LbFinal-Cloud> terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Finding latest version of hashicorp/archive...
- Installing hashicorp/aws v4.46.0...
- Installed hashicorp/aws v4.46.0 (signed by HashiCorp)
- Installing hashicorp/archive v2.2.0...
- Installed hashicorp/archive v2.2.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

```
PS C:\Users\johan\OneDrive\Escritorio\LbFinal-Cloud> terraform plan

data.archive_file.utils_layer: Reading...
data.archive_file.Complementos["put"]: Reading...
data.archive_file.Complementos["delete"]: Reading...
data.archive_file.Complementos["post"]: Reading...
data.archive_file.Complementos["get"]: Reading...
data.archive_file.utils_layer: Read complete after 0s [id=b5c5e65ba64a6338a9cd9cdade2babe9dad4367c]
data.archive_file.Complementos["delete"]: Read complete after 0s [id=50b5c31c691419b48c1a6b3d10b8a473a75d505d]
data.archive_file.Complementos["put"]: Read complete after 0s [id=3b0e2a0fb7cacf8b05ee15140a752f0f1fd3fdcb]
data.archive_file.Complementos["get"]: Read complete after 0s [id=dc85576a41cc751b7e49c62f6b5c7d134bbccd0b]
data.archive_file.Complementos["post"]: Read complete after 0s [id=57256bea045aebce8739577e35544ca85aef7517]
data.aws_iam_policy_document.lambda_assume_role: Reading...
data.aws_iam_policy_document.lambda_assume_role: Read complete after 0s [id=3693445097]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
following symbols:
+ create
<= read (data resources)

Terraform will perform the following actions:

# data.aws_iam_policy_document.create_logs_cloudwatch will be read during apply
# (depends on a resource or a module with changes pending)
<= data "aws_iam_policy_document" "create_logs_cloudwatch" {
  + id   = (known after apply)
  + json = (known after apply)

  + statement {
    + actions = [
```

Por último, aplicamos

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run
"terraform apply" now.
PS C:\Users\johan\OneDrive\Escritorio\LbFinal-Cloud> terraform apply -auto-approve
data.archive_file.Complementos["put"]: Reading...
data.archive_file.utils_layer: Reading...
data.archive_file.Complementos["post"]: Reading...
data.archive_file.Complementos["get"]: Reading...
data.archive_file.Complementos["delete"]: Reading...
data.archive_file.Complementos["put"]: Read complete after 0s [id=3b0e2a0fb7cacf8b05ee15140a752f0f1fd3fdcb]
data.archive_file.utils_layer: Read complete after 0s [id=b5c5e65ba64a6338a9cd9cdade2babe9dad4367c]
data.archive_file.Complementos["post"]: Read complete after 0s [id=57256bea045aebce8739577e35544ca85aef7517]
data.archive_file.Complementos["get"]: Read complete after 0s [id=dc85576a41cc751b7e49c62f6b5c7d134bbccd0b]
data.archive_file.Complementos["delete"]: Read complete after 0s [id=50b5c31c691419b48c1a6b3d10b8a473a75d505d]
data.aws_iam_policy_document.lambda_assume_role: Reading...
aws_lambda_layer_version.utils: Refreshing state... [id=arn:aws:lambda:us-east-1:476786095040:layer:Complementos:2]
aws_apigatewayv2_api.this: Refreshing state... [id=d94e6u12hk]
aws_dynamodb_table.this: Refreshing state... [id=complementos-dev]
data.aws_iam_policy_document.lambda_assume_role: Read complete after 0s [id=3693445097]
aws_iam_role.rest_api_role: Refreshing state... [id=complementos-dev-lambda-permiso]
aws_apigatewayv2_stage.this: Refreshing state... [id=$default]
aws_dynamodb_table.item.this: Refreshing state... [id=complementos-dev|id||1]
aws_ssm_parameter.db_table: Refreshing state... [id=complementos-dev-db-table]
data.aws_iam_policy_document.create_logs_cloudwatch: Reading...
```

Luego podemos verificar la creación en AWS

The screenshot shows the AWS Management Console interface. On the left, there is a navigation pane with 'Gateway de API' selected. The main content area displays the details for the API 'complementos-dev-api'. At the top right, there is a status bar with 'Etapa: -' and an 'Implementación' button. Below this, there is an 'Editar' button. The 'Detalles de la API' section contains a table with the following information:

ID de API	Protocolo	Creado
d94e6u12hk	HTTP	2022-12-09
Descripción	Punto de enlace predeterminado	
No Description	Habilitado	

aws

Servicios

Búsqueda

[Alt+S]

Norte de Virginia

johan

Gateway de API

API

Nombres de dominio personalizados

Enlaces de VPC

API: complementos-de... (d94e6u12hk)

Desarrollar

Rutas

Autorización

Integraciones

CORS

Volver a importar

Exportar

Etapas para complementos-dev-api

Buscar recursos

Nombre de etapa	URL de invocación	Implementación asociada	Implementación automática	Última actualización
\$default	https://d94e6u12hk.execute-api.us-east-1.amazonaws.com	o89ovm	enabled	2022-12-09

Etiquetas (5)

Buscar recursos

Clave	Valor
Project	Rest Api GW
Owner	Johan Ramirez
ManagedBy	Terraform
Env	dev

Comentarios

¿Busca la selección de idiomas? Encuéntrala en la nueva Unified Settings

© 2022, Amazon Web Services, Inc. o sus filiales.

Privacidad

Términos

Preferencias de cookies

aws

Servicios

Búsqueda

Gateway de API

API

Nombres de dominio personalizados

Enlaces de VPC

API: complementos-de... (d94e6u12hk)

Desarrollar

Rutas

Autorización

Integraciones

CORS

Volver a importar

Exportar

Rutas

Rutas para complementos-dev-api

Create

Buscar

/v1

/complementos

/[{ComplementosId}]

GET

/Complementos

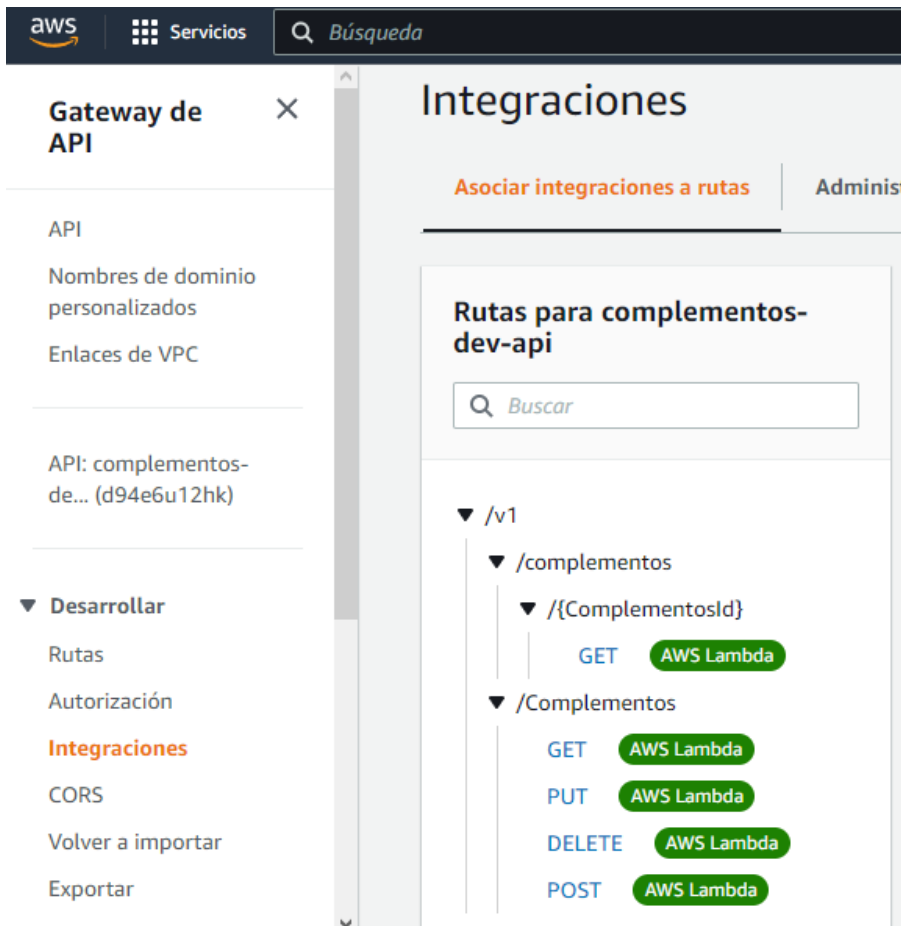
GET

PUT

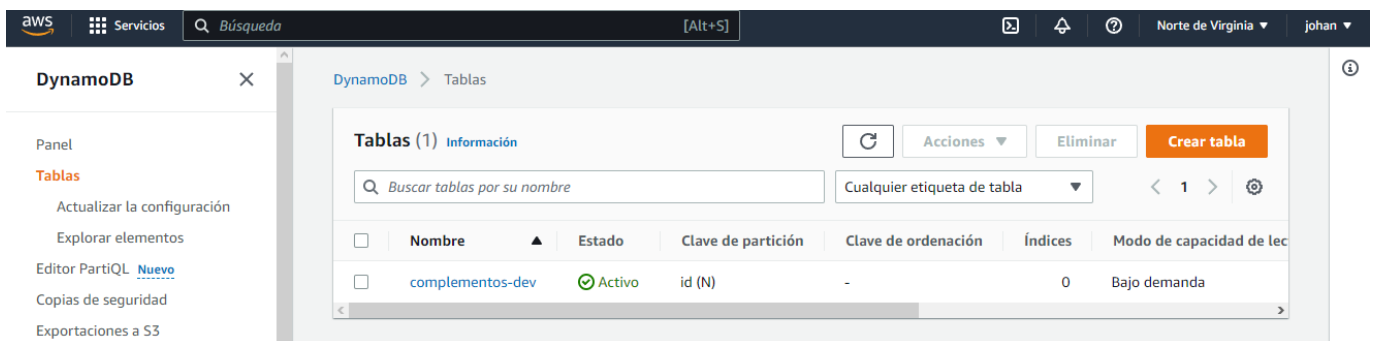
DELETE

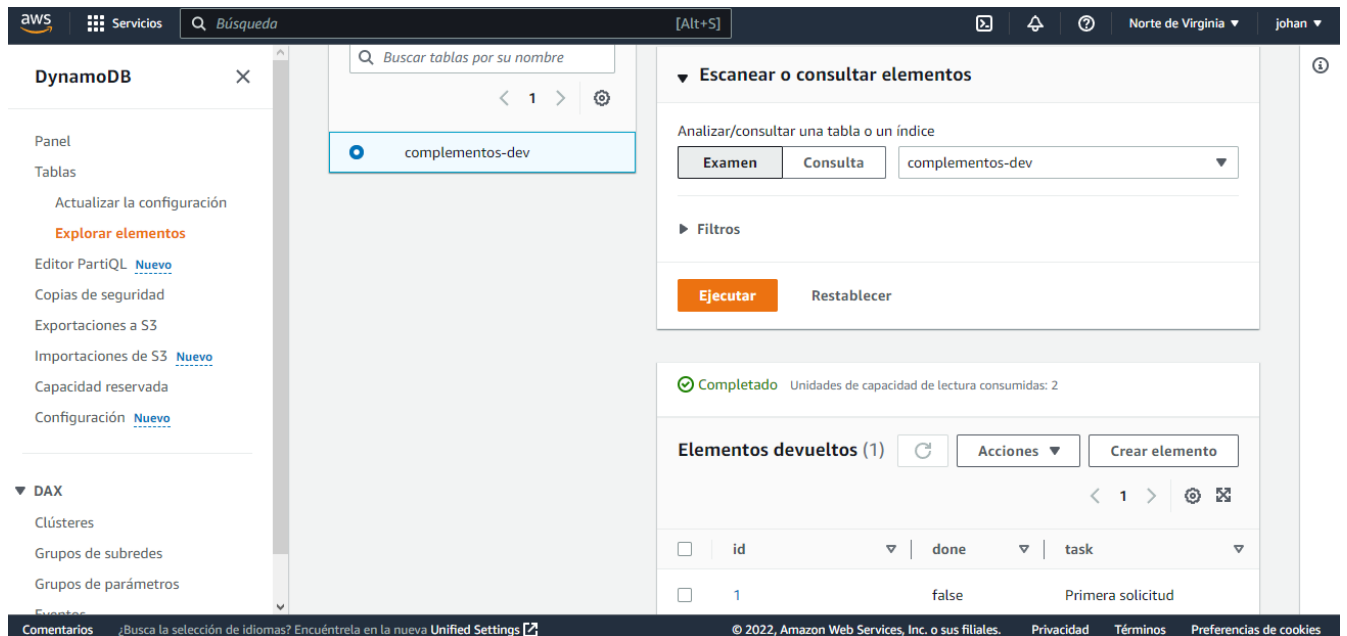
POST





También podemos observar la tabla que creamos en la base de datos





## Serverless

Procedemos a instalar serverless

```
Microsoft Windows [Versión 10.0.19044.2251]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\johan>npm install -g serverless
npm WARN deprecated querystring@0.2.1: The querystring API is considered Legacy. new code should use the URLSearchParams API instead.
npm WARN deprecated querystring@0.2.0: The querystring API is considered Legacy. new code should use the URLSearchParams API instead.
npm WARN deprecated querystring@0.2.0: The querystring API is considered Legacy. new code should use the URLSearchParams API instead.
npm WARN deprecated superagent@7.1.6: Please downgrade to v7.1.5 if you need IE/ActiveXObject support OR upgrade to v8.0.0 as we no longer support IE and published an incorrect patch version (see https://github.com/visionmedia/superagent/issues/1731)

added 406 packages, and audited 407 packages in 1m

66 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
npm notice
npm notice New major version of npm available! 8.19.2 -> 9.2.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v9.2.0
npm notice Run npm install -g npm@9.2.0 to update!
npm notice
C:\Users\johan>
```

Ya teniendo Serverless podemos crear nuestro proyecto

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL

PS C:\Users\johan\OneDrive\Escritorio\LbFinal-Cloud\serverless> serverless

Creating a new serverless project

? What do you want to make? AWS - Node.js - HTTP API
? What do you want to call this project? lambda-crud

✓ Project successfully created in lambda-crud folder

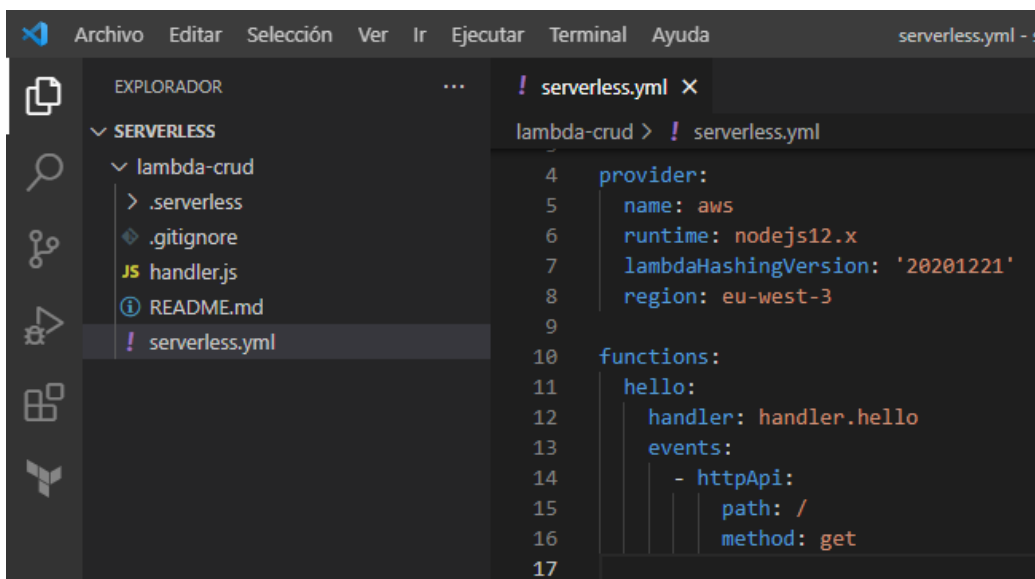
? Do you want to login/register to Serverless Dashboard? No

? Do you want to deploy now? (Y/n) No
? Do you want to deploy now? No

What next?
Run these commands in the project directory:

serverless deploy    Deploy changes
serverless info       View deployed endpoints and resources
serverless invoke     Invoke deployed functions
serverless --help     Discover more commands
PS C:\Users\johan\OneDrive\Escritorio\LbFinal-Cloud\serverless> 
```

Definimos la región en el archivo serverless.yml

The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane shows a project structure with a folder named 'SERVERLESS' containing a subfolder 'lambda-crud'. Inside 'lambda-crud', there are files: '.serverless', '.gitignore', 'handler.js', 'README.md', and 'serverless.yml'. The 'serverless.yml' file is selected and its content is displayed in the main editor. The configuration in 'serverless.yml' defines the provider as 'aws', sets the runtime to 'nodejs12.x', the lambdaHashingVersion to '20201221', and the region to 'eu-west-3'. It also defines a function named 'hello' that uses the 'handler.hello' handler and is triggered by an 'httpApi' event with a GET method on the root path.

```
serverless.yml
1
2
3
4 provider:
5   name: aws
6   runtime: nodejs12.x
7   lambdaHashingVersion: '20201221'
8   region: eu-west-3
9
10 functions:
11   hello:
12     handler: handler.hello
13     events:
14       - httpApi:
15         path: /
16         method: get
17
```

## Ejecutamos el serverless

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL

PS C:\Users\johan\OneDrive\Escritorio\LbFinal-Cloud\serverless\lambda-crud> serverless deploy --verbose

Deploying lambda-crud to stage dev (eu-west-3)

Packaging
Excluding development dependencies for service package
Retrieving CloudFormation stack
Creating CloudFormation stack
Creating new change set
Waiting for new change set to be created
Change Set did not reach desired state, retrying
Executing created change set
  CREATE_IN_PROGRESS - AWS::CloudFormation::Stack - lambda-crud-dev
  CREATE_IN_PROGRESS - AWS::S3::Bucket - ServerlessDeploymentBucket
  CREATE_IN_PROGRESS - AWS::S3::Bucket - ServerlessDeploymentBucket
  CREATE_COMPLETE - AWS::S3::Bucket - ServerlessDeploymentBucket
  CREATE_IN_PROGRESS - AWS::S3::BucketPolicy - ServerlessDeploymentBucketPolicy
  CREATE_IN_PROGRESS - AWS::S3::BucketPolicy - ServerlessDeploymentBucketPolicy
  CREATE_COMPLETE - AWS::S3::BucketPolicy - ServerlessDeploymentBucketPolicy
  CREATE_COMPLETE - AWS::CloudFormation::Stack - lambda-crud-dev
Uploading
Uploading CloudFormation file to S3
Uploading State file to S3
```

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  powershell

✓ Service deployed to stack lambda-crud-dev (125s)

endpoint: GET - https://e5b4uv4t2k.execute-api.eu-west-3.amazonaws.com/
functions:
  hello: lambda-crud-dev-hello (1.7 kB)

Stack Outputs:
  HelloLambdaFunctionQualifiedArn: arn:aws:lambda:eu-west-3:476786095040:function:lambda-crud-dev-hello:1
  HttpApiId: e5b4uv4t2k
  ServerlessDeploymentBucketName: lambda-crud-dev-serverlessdeploymentbucket-1n8hp48xpemxv
  HttpApiUrl: https://e5b4uv4t2k.execute-api.eu-west-3.amazonaws.com

1 deprecation found: run 'serverless doctor' for more details

Need a better logging experience than CloudWatch? Try our Dev Mode in console: run "serverless --console"
PS C:\Users\johan\OneDrive\Escritorio\LbFinal-Cloud\serverless\lambda-crud> 
```

Podemos observar el objeto Json

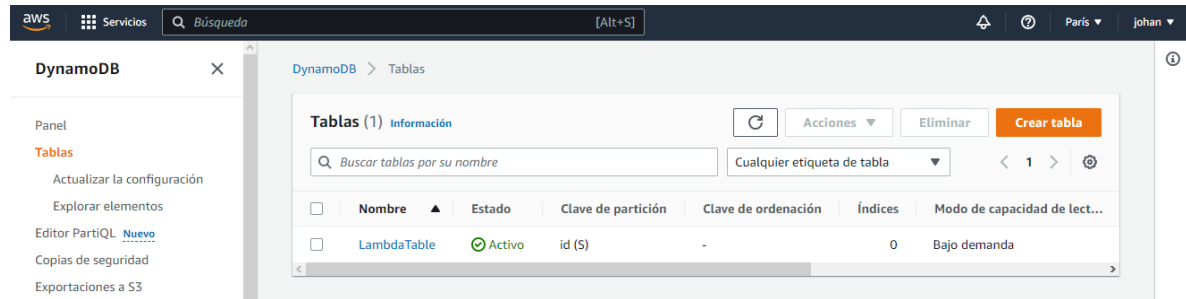
```
e5b4uv4t2k.execute-api.eu-west-3.amazonaws.com

{
  "message": "Go Serverless v3.0! Your function executed successfully!",
  "input": {
    "version": "2.0",
    "routeKey": "GET /",
    "rawPath": "/",
    "rawQueryString": "",
    "headers": {
      "accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9",
      "accept-encoding": "gzip, deflate, br",
      "accept-language": "es-ES,es;q=0.9",
      "content-length": "0",
      "host": "e5b4uv4t2k.execute-api.eu-west-3.amazonaws.com",
      "sec-ch-ua": "\"Opera GX\";v=\"93\", \"Not(A)Brand\";v=\"8\", \"Chromium\";v=\"107\"",
      "sec-ch-ua-mobile": "?0",
      "sec-ch-ua-platform": "Windows",
      "sec-fetch-dest": "document",
      "sec-fetch-mode": "navigate",
      "sec-fetch-site": "none",
      "sec-fetch-user": "?1",
      "upgrade-insecure-requests": "1",
      "user-agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36 OPR/93.0.0.0",
      "x-amzn-trace-id": "Root=1-63939dee-674bcf5e61a488096405898e",
      "x-forwarded-for": "179.13.6.93",
      "x-forwarded-port": "443",
      "x-forwarded-proto": "https"
    },
    "requestContext": {
      "accountId": "476786095040",
      "apiId": "e5b4uv4t2k",
      "domainName": "e5b4uv4t2k.execute-api.eu-west-3.amazonaws.com",
      "domainPrefix": "e5b4uv4t2k",
      "http": {
        "method": "GET",
        "path": "/",
        "protocol": "HTTP/1.1",
        "sourceIp": "179.13.6.93",
        "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36 OPR/93.0.0.0"
      }
    }
  }
}
```

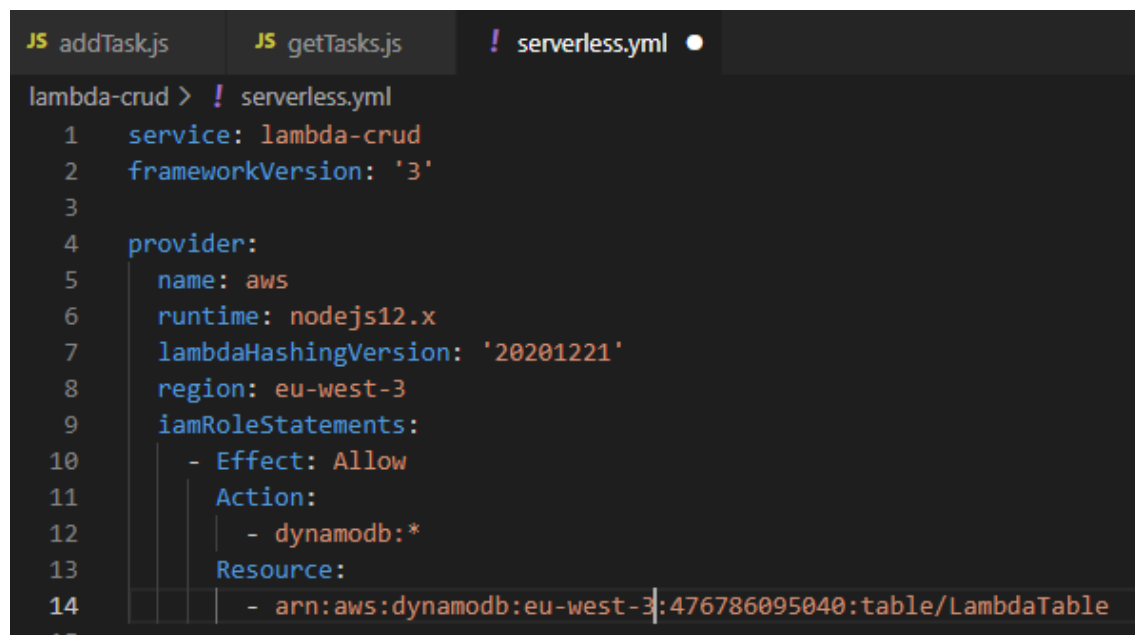
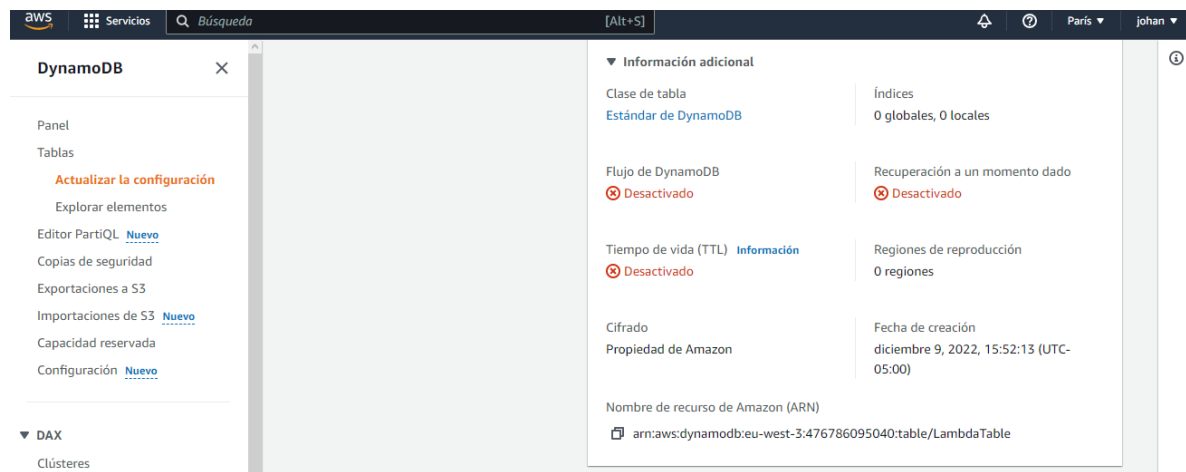
Configuramos el DynamoBd en el mismo archivo

```
serverless.yml
lambda-crud > serverless.yml
53
54 resources:
55   Resources:
56     LambdaTable:
57       Type: AWS::DynamoDB::Table
58       Properties:
59         TableName: LambdaTable
60         BillingMode: PAY_PER_REQUEST
61         AttributeDefinitions:
62           - AttributeName: id
63             AttributeType: S
64         KeySchema:
65           - AttributeName: id
66             KeyType: HASH
67
```

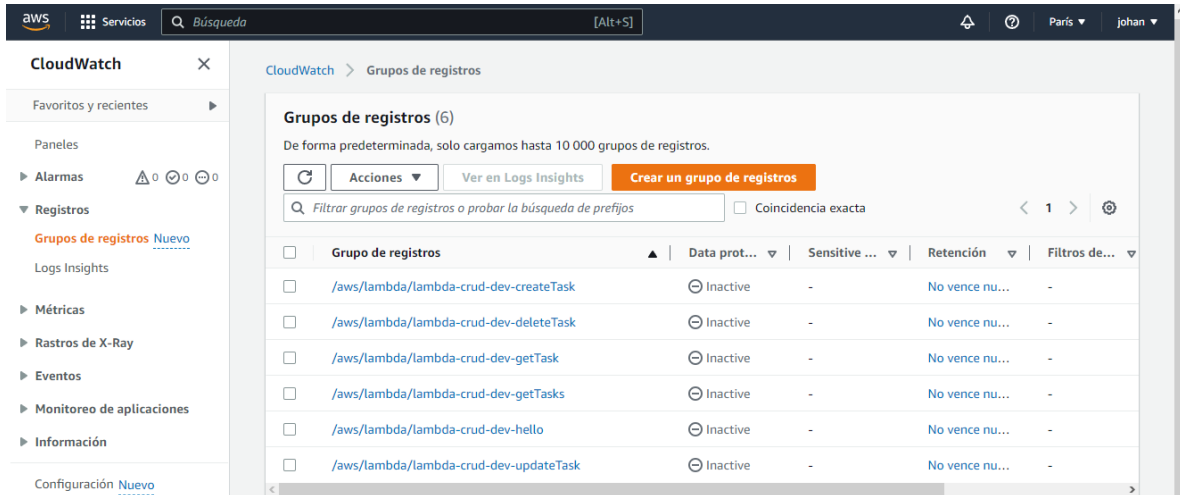
Luego de subir los cambios podemos observar que en AWS nuestra tabla fue creada



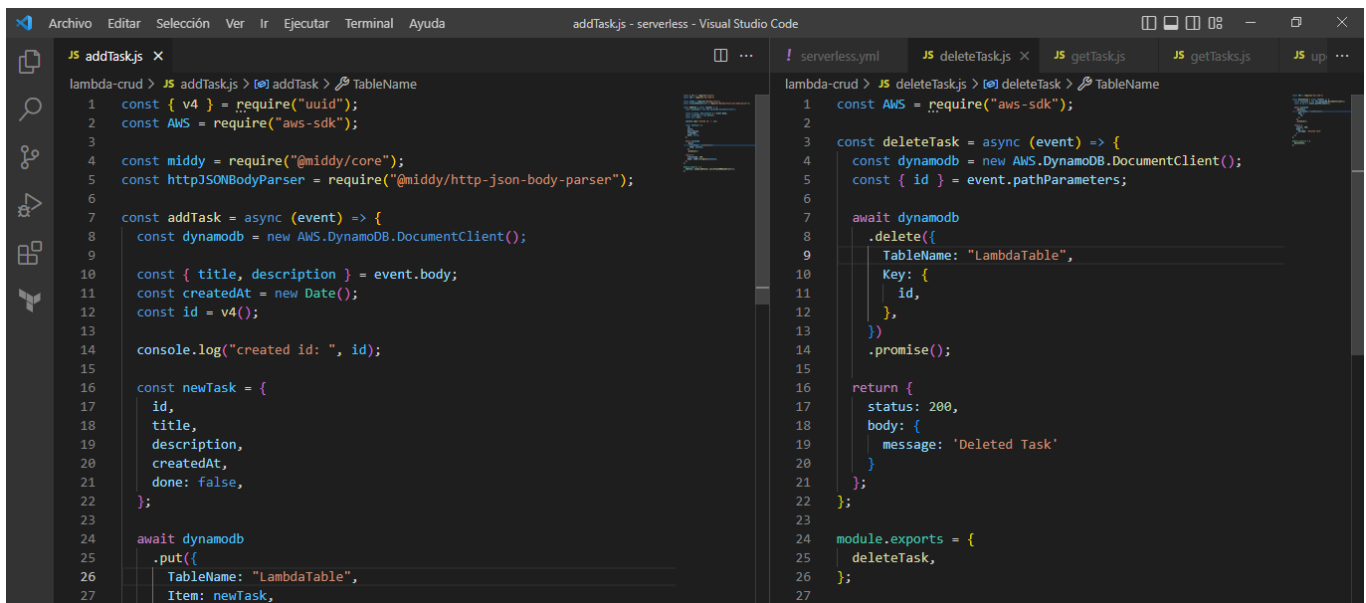
Con la información que nos proporciona AWS completamos nuestro archivo `Serveless.yml`

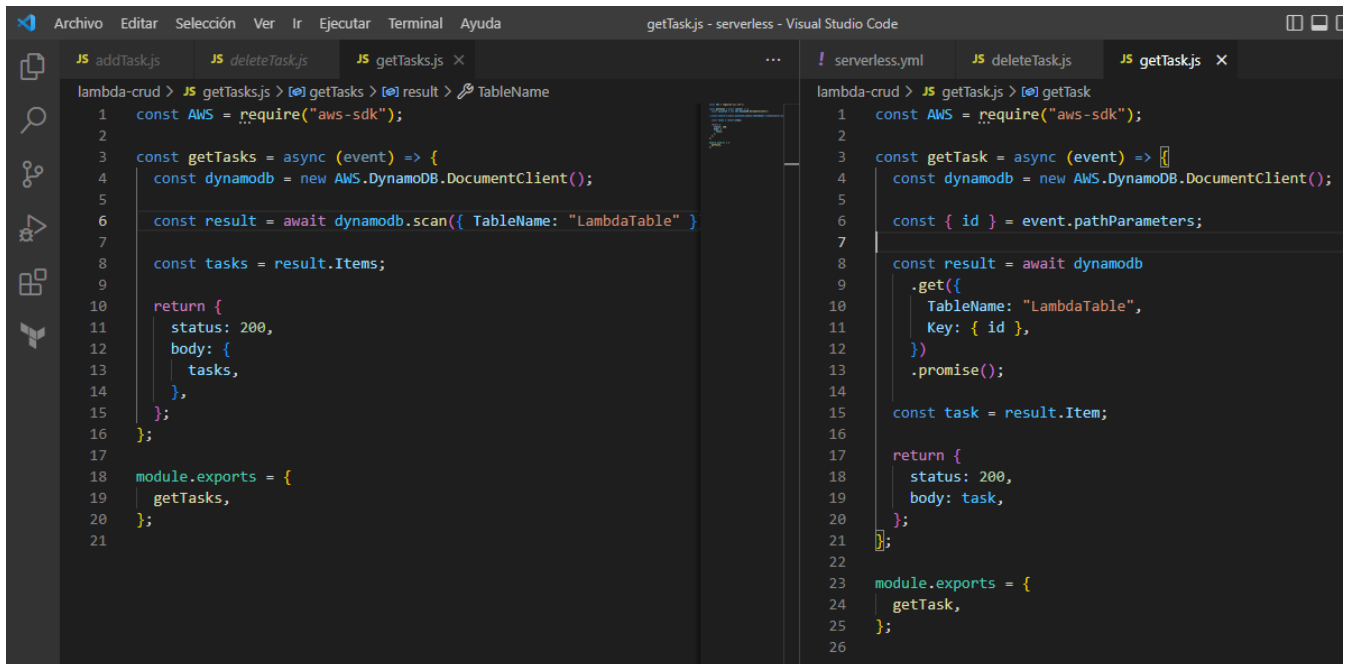


Ahora con el servicio CloudWatch podemos ver los mensajes que salen de nuestro servidor



Procedemos a codificar el crud

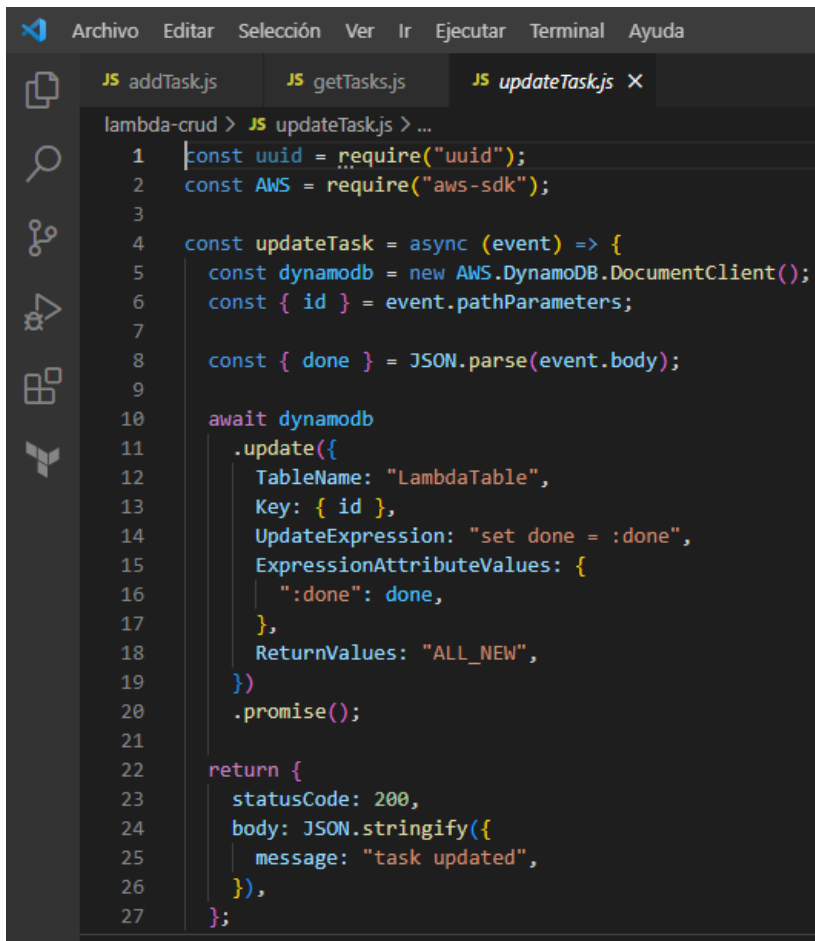




```
getTasks.js
1  const AWS = require("aws-sdk");
2
3  const getTasks = async (event) => {
4    const dynamodb = new AWS.DynamoDB.DocumentClient();
5
6    const result = await dynamodb.scan({ TableName: "LambdaTable" });
7
8    const tasks = result.Items;
9
10   return {
11     status: 200,
12     body: {
13       tasks,
14     },
15   };
16 };
17
18 module.exports = {
19   getTasks,
20 };
21
```

```
getTask.js
1  const AWS = require("aws-sdk");
2
3  const getTask = async (event) => {
4    const dynamodb = new AWS.DynamoDB.DocumentClient();
5
6    const { id } = event.pathParameters;
7
8    const result = await dynamodb
9      .get({
10        TableName: "LambdaTable",
11        Key: { id },
12      })
13      .promise();
14
15    const task = result.Item;
16
17    return {
18      status: 200,
19      body: task,
20    };
21 };
22
23 module.exports = {
24   getTask,
25 };
26
```





```
lambda-crud > JS updateTask.js > ...
1  const uuid = require("uuid");
2  const AWS = require("aws-sdk");
3
4  const updateTask = async (event) => {
5      const dynamodb = new AWS.DynamoDB.DocumentClient();
6      const { id } = event.pathParameters;
7
8      const { done } = JSON.parse(event.body);
9
10     await dynamodb
11         .update({
12             TableName: "LambdaTable",
13             Key: { id },
14             UpdateExpression: "set done = :done",
15             ExpressionAttributeValues: {
16                 ":done": done,
17             },
18             ReturnValues: "ALL_NEW",
19         })
20         .promise();
21
22     return {
23         statusCode: 200,
24         body: JSON.stringify({
25             message: "task updated",
26         }),
27     };
28 }
```

Finalmente volvemos a ejecutar el serverless para subir los cambios y con esto tendríamos el crud completo para las peticiones http

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL

Retrieving CloudFormation stack
Removing old service artifacts from S3

✓ Service deployed to stack lambda-crud-dev (56s)

endpoints:
GET - https://e5b4uv4t2k.execute-api.eu-west-3.amazonaws.com/
POST - https://e5b4uv4t2k.execute-api.eu-west-3.amazonaws.com/tasks
GET - https://e5b4uv4t2k.execute-api.eu-west-3.amazonaws.com/tasks
GET - https://e5b4uv4t2k.execute-api.eu-west-3.amazonaws.com/tasks/{id}
PUT - https://e5b4uv4t2k.execute-api.eu-west-3.amazonaws.com/tasks/{id}
DELETE - https://e5b4uv4t2k.execute-api.eu-west-3.amazonaws.com/tasks/{id}

functions:
hello: lambda-crud-dev-hello (3.8 kB)
createTask: lambda-crud-dev-createTask (3.8 kB)
getTasks: lambda-crud-dev-getTasks (3.8 kB)
getTask: lambda-crud-dev-getTask (3.8 kB)
updateTask: lambda-crud-dev-updateTask (3.8 kB)
deleteTask: lambda-crud-dev-deleteTask (3.8 kB)

Stack Outputs:
GetTasksLambdaFunctionQualifiedArn: arn:aws:lambda:eu-west-3:476786095040:function:lambda-crud-dev-getTasks:3
DeleteTaskLambdaFunctionQualifiedArn: arn:aws:lambda:eu-west-3:476786095040:function:lambda-crud-dev-deleteTask:3
UpdateTaskLambdaFunctionQualifiedArn: arn:aws:lambda:eu-west-3:476786095040:function:lambda-crud-dev-updateTask:3
CreateTaskLambdaFunctionQualifiedArn: arn:aws:lambda:eu-west-3:476786095040:function:lambda-crud-dev-createTask:3
HelloLambdaFunctionQualifiedArn: arn:aws:lambda:eu-west-3:476786095040:function:lambda-crud-dev-hello:3
```

## Resultados finales

Servicios

[Alt+S]

París

johan

AWS Lambda

Panel

Aplicaciones

Funciones

▼ Recursos adicionales

Configuraciones de la firma de código

Capas

Réplicas

▼ Recursos de AWS relacionados

Máquinas de estado de Step Functions

Lambda > Funciones

Funciones (6)

Última obtención ahora

Acciones

Crear una función

< 1 >

<input type="checkbox"/>	Nombre de la función	Descripción	Tipo de paquete	Tiempo de ejecución	Última modificación
<input type="checkbox"/>	lambda-crud-dev-deleteTask	-	Zip	Node.js 12.x	hace 3 minutos
<input type="checkbox"/>	lambda-crud-dev-getTask	-	Zip	Node.js 12.x	hace 3 minutos
<input type="checkbox"/>	lambda-crud-dev-updateTask	-	Zip	Node.js 12.x	hace 3 minutos
<input type="checkbox"/>	lambda-crud-dev-hello	-	Zip	Node.js 12.x	hace 3 minutos
<input type="checkbox"/>	lambda-crud-dev-getTasks	-	Zip	Node.js 12.x	hace 3 minutos
<input type="checkbox"/>	lambda-crud-dev-createTask	-	Zip	Node.js 12.x	hace 3 minutos

## **Bibliografía**

<https://www.youtube.com/watch?v=da6E0R2grfo>

<https://github.com/hashicorp/terraform-provider-aws/issues/14873>

[https://repost.aws/questions/QU\\_m8virJwQKacrrt6YZa-Pg/access-denied-to-ssm-parameter-but-policy-in-place](https://repost.aws/questions/QU_m8virJwQKacrrt6YZa-Pg/access-denied-to-ssm-parameter-but-policy-in-place)

<https://www.youtube.com/watch?v=SkbcSbF0uq4&t=938s>

<https://www.youtube.com/watch?v=vvux4WOU5dc>