



CloudOps Guild
Together, towards mastery in Cloud and DevOps

🚀 Crea un API RESTful sin servidores con API Gateway + Lambda + DynamoDB









🎯 **Objetivo:** Diseñar una API funcional sin servidores utilizando arquitectura serverless con AWS API Gateway y AWS Lambda.

🧠 **Nivel:** Intermedio (ideal para personas que están empezando en el mundo cloud)



CloudOps Guild
Together, towards mastery in Cloud and DevOps

Tabla de contenidos

-  Escenario del mundo real
-  ¿Qué es una API RESTful serverless?
-  Características principales
-  Paso a paso: Crea tu propia API sin servidores
 - Paso 1: Requisitos previos
 - Paso 2: Crear la función Lambda
 - Paso 3: Crear la API en API Gateway
 - Paso 4: Configurar el Mapping Template (opcional)
 - Paso 5: Desplegar la API
 - Paso 6: Probar la API desde Windows CMD o MAC terminal
-  Retos comunes y cómo resolverlos
-  Conclusiones y próximos pasos



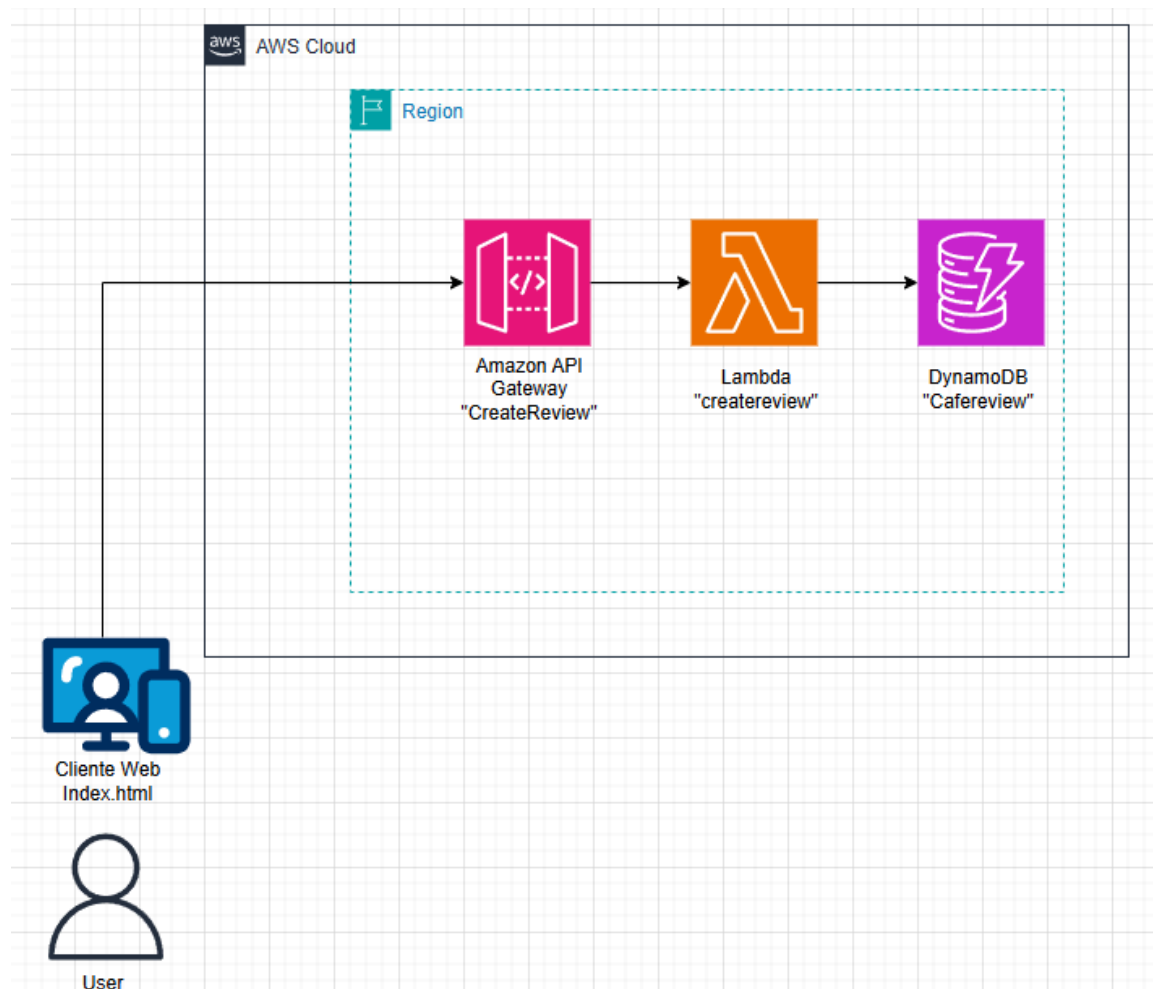
CloudOps Guild
Together, towards mastery in Cloud and DevOps

Escenario del mundo real

CloudFizz es una startup que quiere lanzar una app de reseñas de cafeterías locales. Necesitan una API para que los usuarios puedan crear reseñas, pero no quieren manejar servidores, ni infraestructura.

La solución ideal: una API *serverless* con AWS API Gateway + AWS Lambda.

Diagrama de arquitectura del laboratorio





CloudOps Guild
Together, towards mastery in Cloud and DevOps

Paso a paso: Crea tu propia API sin servidores

♦ Paso 1: Requisitos previos

Antes de comenzar, asegúrate de tener:

- ☒ Una cuenta activa de AWS
- ☒ Acceso a la consola de AWS (<https://console.aws.amazon.com/>)
- ☒ Permisos para usar Lambda, API Gateway e IAM
- ☒ Instalado **curl** o Postman para probar la API
- ☒ Conocimientos básicos de Python (muy básicos)

♦ Paso 2: Crear la función Lambda

1. Ve al servicio Lambda en la consola de AWS



CloudOps Guild

Together, towards mastery in Cloud and DevOps

Lambda > Functions > Create function

Create function Info

Choose one of the following options to create your function.

☒ **Author from scratch**
Start with a simple Hello World example.

☐ **Use a blueprint**
Build a Lambda application from sample code and configuration presets for common use cases.

☐ **Container image**
Select a container image to deploy for your function.

Basic information

Function name
Enter a name that describes the purpose of your function.

createreview

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Python 3.13

Architecture Info
Choose the instruction set architecture you want for your function code.

☒ x86_64

☐ arm64

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 [\[i\]](#)

☒ Create a new role with basic Lambda permissions

☐ Use an existing role

☐ Create a new role from AWS policy templates

2. Haz clic en “Crear función”

3. Elige:

- ☐ Autor desde cero (Author from scratch)
- ☐ Nombre: **createreview**

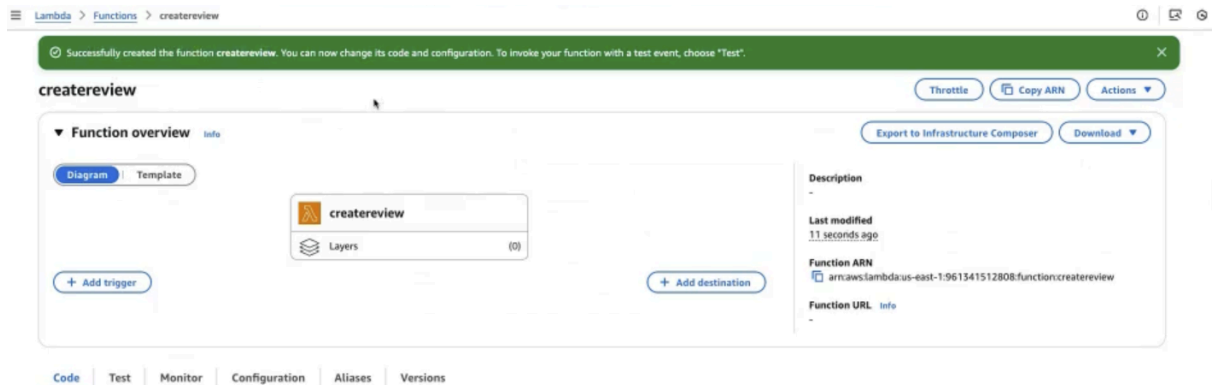
4. Runtime: **Python 3.13**

5. En chage default execution role seleccionar la opción **Create a new role with basic Lambda permissions**

6. Haz clic en Crear función



CloudOps Guild
Together, towards mastery in Cloud and DevOps



7. Reemplaza el código por este:

```
import json

def lambda_handler(event, context):

    try:

        response = {

            "mensaje": "¡Reseña creada exitosamente!",

            "reseña": event

        }

        return {

            'statusCode': 201,

            'body': json.dumps(response, ensure_ascii=False)

        }

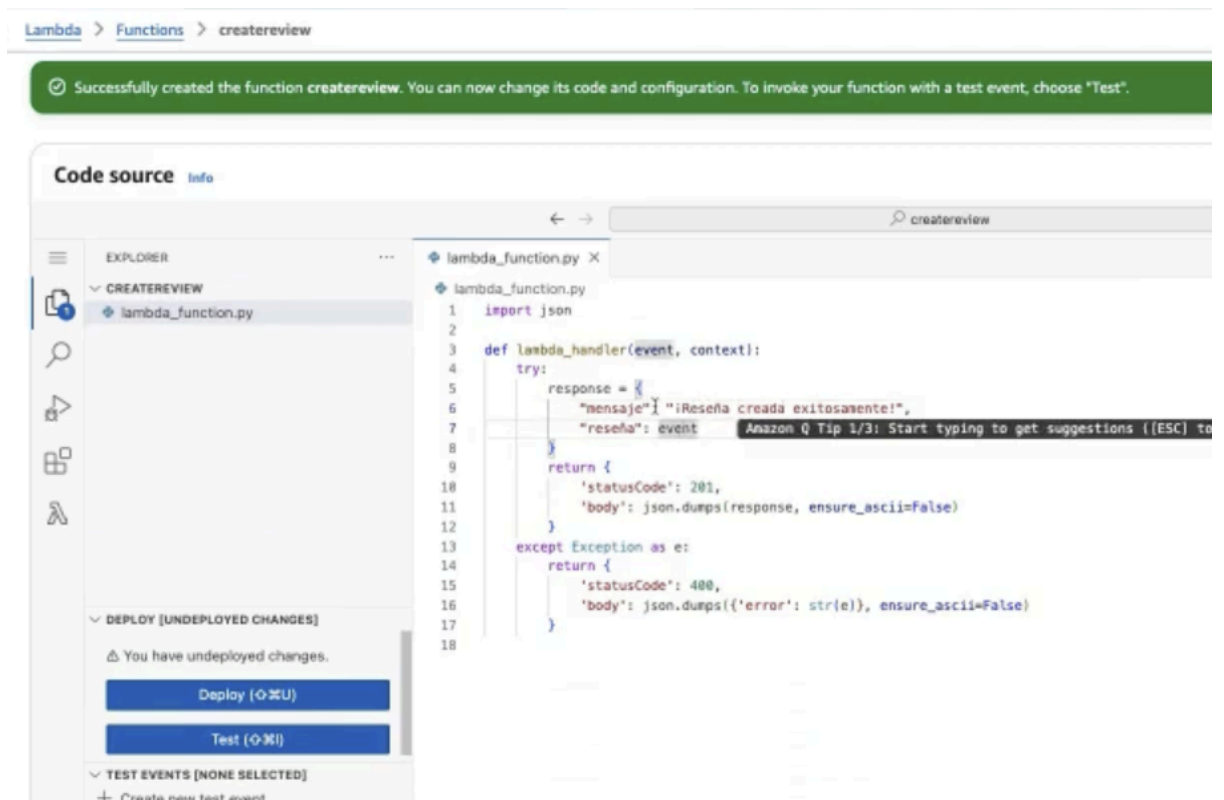
    except Exception as e:
```



CloudOps Guild
Together, towards mastery in Cloud and DevOps

```
return {  
    'statusCode': 400,  
    'body': json.dumps({'error': str(e)}, ensure_ascii=False)  
}
```

6. Haz clic en “Deploy” o “Implementar” para guardar el código





CloudOps Guild
Together, towards mastery in Cloud and DevOps

♦ Paso 3: Crear la API en API Gateway

API Gateway > APIs > Create API > Create REST API

Create REST API Info

API details

☒ **New API**
Create a new REST API.

☐ **Clone existing API**
Create a copy of an API in this AWS account.

☐ **Import API**
Import an API from an OpenAPI definition.

☐ **Example API**
Learn about API Gateway with an example API.

API name
CreateReview

Description - optional
API use

API endpoint type
Regional APIs are deployed in the current AWS Region. Edge-optimized APIs route requests to the nearest CloudFront Point of Presence. Private APIs are only accessible from VPCs.
Regional

IP address type Info
Select the type of IP addresses that can invoke the default endpoint for your API.
☒ **IPv4**
Supports only edge-optimized and Regional API endpoint types.
☐ **Dualstack**
Supports all API endpoint types.

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

1. Ve al servicio API Gateway
2. Haz clic en Crear API
3. Selecciona API REST → clic en Crear (Build)
4. Nombra tu API: **CreateReview**
5. En el menú izquierdo, haz clic derecho en / y selecciona “Crear recurso”
 - Nombre del recurso: **review**
 - Ruta del recurso: **/review**
 - Copia la Invoke URL
6. Con el recurso seleccionado, haz clic en “Crear método” y elige **POST**
7. En integración, selecciona:
 - Tipo de integración: Función Lambda



CloudOps Guild
Together, towards mastery in Cloud and DevOps

- **Escoger Region y ARN de la lambda createreview**
- **opción Usar Lambda Proxy Integration (deja desmarcada)**
- **Nombre de la función: **createreview****
- **Haz clic en Guardar**
- **Activar CORS desde el recurso de la API.**
- **Asegurarse que desde la configuración de la petición de integración (integration request) en el método POST escoger la opción When there are no templates defined(recommended) en request body passthrough**

Method details

Method type
POST

Integration type

- ☒ **Lambda function**
Integrate your API with a Lambda function.
- ☐ HTTP
Integrate with an existing HTTP endpoint.
- ☐ Mock
Generate a response based on API Gateway mappings and transformations.
- ☐ AWS service
Integrate with an AWS Service.
- ☐ VPC link
Integrate with a resource that isn't accessible over the public internet.

☒ **Lambda proxy integration**
Send the request to your Lambda function as a structured event.

Lambda function
Provide the Lambda function name or alias. You can also provide an ARN from another account.

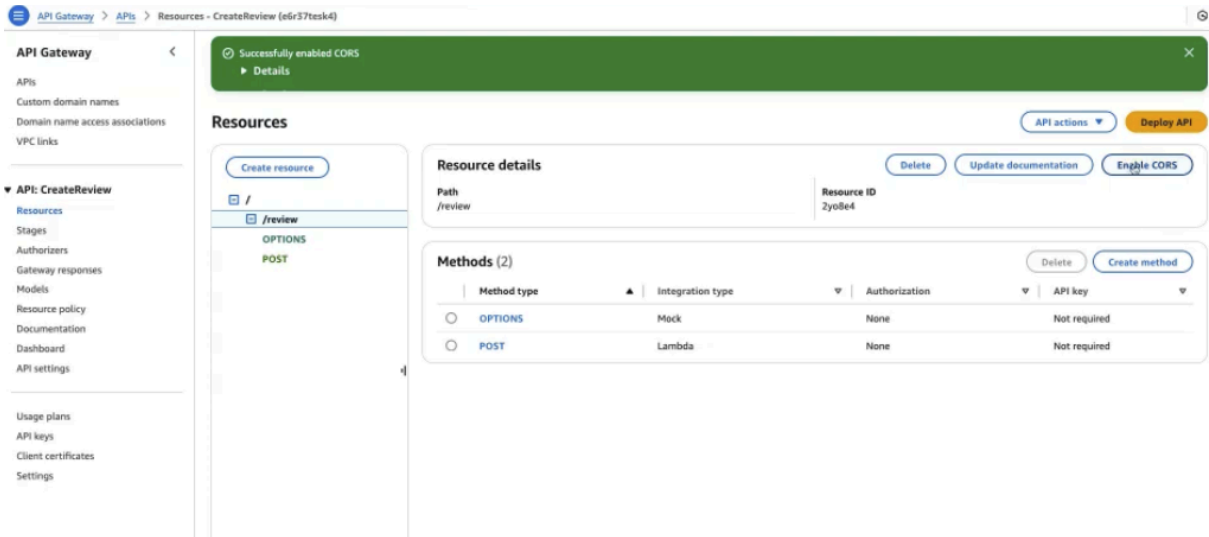
us-east-1

Grant API Gateway permission to invoke your Lambda function. To turn off, update the function's resource policy yourself, or provide an invoke role that API Gateway uses to invoke your function.



CloudOps Guild

Together, towards mastery in Cloud and DevOps



♦ Paso 4: Configurar el Mapping Template (opcional)

Este paso es clave para que Lambda reciba los datos correctamente.

1. Dentro del método POST, haz clic en “Integration Request”
2. Ve a la sección Mapping Templates
3. En Content-Type, escribe: **application/json** y haz clic en el ✓
4. En el editor del template, borra lo que aparece y pega esto:

json

```
#set($inputRoot = $input.path('$'))
```

```
$input.json('$')
```



CloudOps Guild
Together, towards mastery in Cloud and DevOps

✓ Esto asegura que el cuerpo de la solicitud se pase correctamente a Lambda como JSON

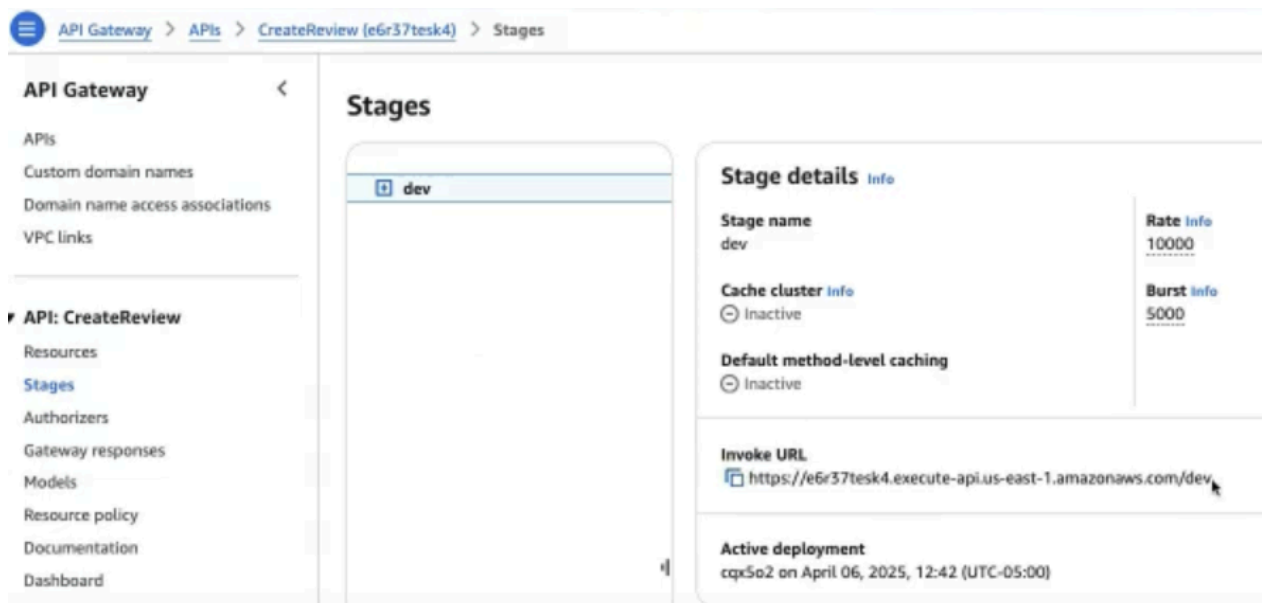
5. En la parte superior, haz clic en Guardar

♦ Paso 5: Desplegar la API

1. En el panel izquierdo, haz clic en Acciones → Desplegar API

2. Selecciona Nueva etapa

- Nombre de etapa: **dev**
- Haz clic en Desplegar





CloudOps Guild
Together, towards mastery in Cloud and DevOps

Obtendrás una URL similar a:

Ejemplo Invoke url

https://API_ID.execute-api.us-east-1.amazonaws.com/stage/resource

(Esa es tu API pública ✨)

♦ Paso 6: Probar la API desde Windows CMD

1. Abre la terminal de CMD en Windows
2. Ejecuta este comando en una sola línea:

Comando Curl desde Windows

curl -X POST
https://<tu-api-id>.execute-api.<region>.amazonaws.com/dev/resenas
-H "Content-Type: application/json" -d "{\"usuario\": \"Ana\",
\"comentario\": \"¡Excelente café artesanal!\"}"

Opcional desde un MAC o Linux



CloudOps Guild
Together, towards mastery in Cloud and DevOps

```
curl -X POST
https://https://API_ID.execute-api.us-east-1.amazonaws.com/stage/resource \

-H "Content-Type: application/json" \

-d '{"usuario": "Ana", "comentario": "¡Excelente café artesanal!"}'
```

📌 Asegúrate de reemplazar **<tu-api-id>**, **<region>**, **<stage>** y **<resource>** con los valores reales de tu API

✅ Respuesta esperada

```
{
  "mensaje": "¡Reseña creada exitosamente!",
  "reseña": {
    "usuario": "Ana",
    "comentario": "¡Excelente café artesanal!"
  }
}
```

🎉 ¡Tu API sin servidores está funcionando!



```
mario.serrano@RENTING00021 ~ % curl --version
curl 8.7.1 (x86_64-apple-darwin23.0) libcurl/8.7.1 (SecureTransport) LibreSSL/3.3.6 zlib/1.2.12 nghttp2/1.61.0
Release-Date: 2024-03-27
Protocols: dict file ftp ftps gopher gophers http https imap imaps ipfs ipns ldap ldaps mqtt pop3 pop3s rtsp smb smbs smtp smtps telnet tftp
Features: alt-svc AsynchDNS GSS-API HSTS HTTP2 HTTPS-proxy IPv6 Kerberos Largefile libz MultiSSL NTLM SPNEGO SSL threadsafe UnixSockets
mario.serrano@RENTING00021 ~ %
```

Problema	Solución
Missing Authentication Token	Verifica que estés usando el método POST y la ruta correcta
403 Forbidden	Verifica los permisos del rol de ejecución de Lambda
El cuerpo no se recibió (body)	Asegúrate de haber configurado correctamente el Mapping Template
Could not parse payload into json	Usa <code>\$input.json('\$')</code> en el Mapping Template (sin comillas extras)
Caracteres raros como <code>\u00e1</code> en la respuesta	Usa <code>json.dumps(..., ensure_ascii=False)</code> en la función Lambda



CloudOps Guild
Together, towards mastery in Cloud and DevOps

🔑 Conclusiones y próximos pasos

- ✓ Lograste construir una API RESTful sin servidores
 - ✓ Usaste Lambda + API Gateway paso a paso
 - ✓ Aprendiste a resolver errores reales que surgen en el camino
 - ✓ ¡Tienes una base para construir un backend completo sin preocuparte por servidores!
-

🚀 ¿Qué sigue?

- Agregar nuevos endpoints (**GET**, **DELETE**, **PUT**)
- Conectar tu API a DynamoDB para guardar los datos
- Crear un frontend que consuma esta API
- Subir este proyecto a GitHub y compartirlo con tu comunidad