



**ESCUELA COLOMBIANA DE INGENIERIA JULIO GARATIVO**

**AWS SERVERLESS APPLICATION MODEL**

**SECURITY**

**2020**

**Ian Sebastián Martínez Rey**

**Arquitectura y gobernabilidad Tecnológica**

**Maestría en informática**

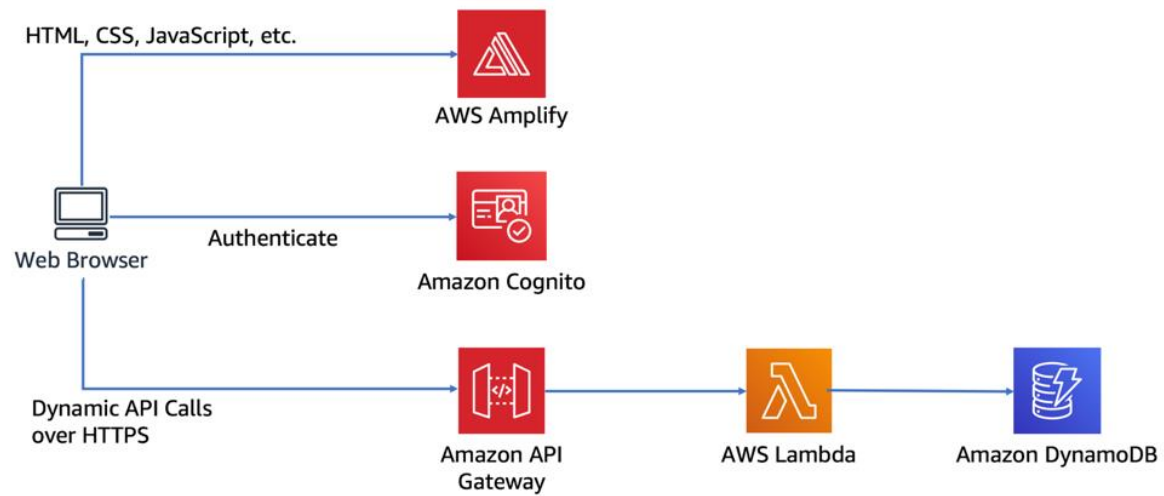
## **INDICE**

- 1. Objetivo**
- 2. Desarrollo actividad**
  - a. Despliegue Front End - Amplify**
  - b. Autenticación – Cognito**
  - c. Despliegue Back End (API Gateway, Lambda, DynamoDB)**

## OBJETIVO

Desarrollar una aplicación serverless con la infraestructura de AWS utilizando los siguientes servicios

- Amplify
- Cognito
- Lambda
- API Gateway
- DynamoDB

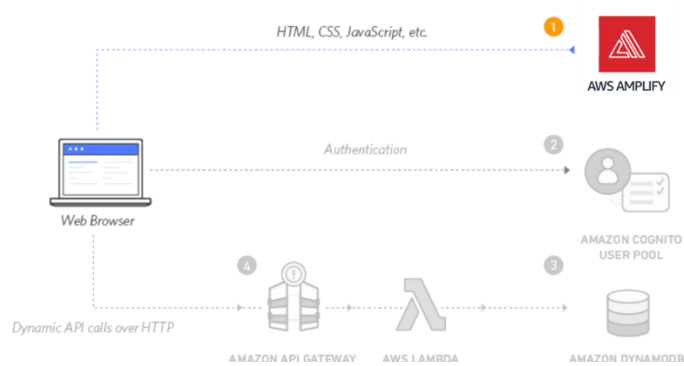


## DESARROLLO ACTIVIDAD

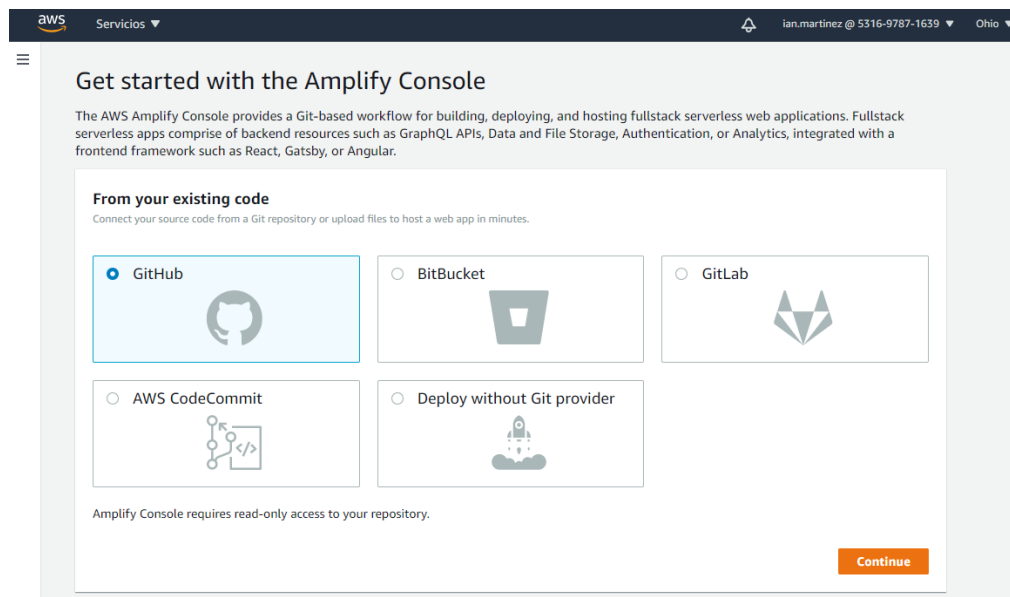
### Archivos del proyecto

<https://github.com/ianmartinezrey/wildrydes-site>

### Despliegue FrontEnd – Amplify



#### 1. Vinculación de la cuenta de AWS con Github



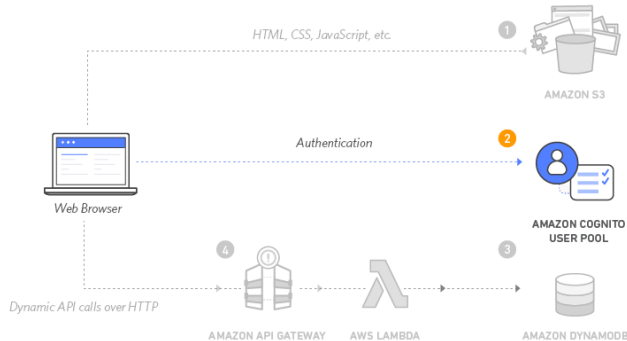
## 2. Se adiciona el repositorio con la aplicación frontend a publicar

The screenshot shows the 'Add repository branch' step in the AWS Amplify console. The interface includes a sidebar with steps: Step 1 (Add repository branch), Step 2 (Configure build settings), and Step 3 (Review). The main content area is titled 'Add repository branch' and features a 'GitHub' section with a success message: 'GitHub authorization was successful.' Below this, the 'Repository service provider' is set to 'GitHub'. The 'Recently updated repositories' section shows a dropdown menu with 'ianmartinezrey/demo\_serverless' selected. The 'Branch' section has a dropdown menu with 'main' selected. There is also a checkbox for 'Connecting a monorepo? Pick a folder.' which is currently unchecked.

## 3. Despliegue de la aplicación en Amplify

The screenshot displays the 'Review' step in the AWS Amplify console. The sidebar shows steps: Step 1 (Add repository branch), Step 2 (Configure build settings), and Step 3 (Review). The main content area is titled 'Review' and contains two sections: 'Repository details' and 'App settings'. The 'Repository details' section shows the repository service as 'GitHub', the repository as 'ianmartinezrey/demo\_serverless', and the branch as 'main'. The 'App settings' section shows the app name as 'demo\_serverless', the framework as 'Web', and the build settings as 'Auto-detected settings will be used'. Below the 'Review' step, the 'demo\_serverless' app page is shown. The sidebar lists various settings: General, Domain management, Build settings, Previews, Notifications, Environment variables, Access control, Access logs, Rewrites and redirects, Documentation, and Support. The main content area shows the app's homepage, a progress bar with four steps (Provision, Build, Deploy, Verify) all marked as complete, and a table with deployment details: Last deployment (29/10/2020 12:37:35), Last commit (This is an autogenerated message | Auto-build | - main), and Previews (Disabled).

## Autenticación – Cognito



### 1. Creación de grupo de usuarios

aws Servicios

Grupos de usuarios | Identidades federadas

## Crear un grupo de usuarios

Cancelar

**Nombre**

¿Qué nombre desea dar a su grupo de usuarios?

Utilice un nombre descriptivo que le permita identificar el grupo de usuarios fácilmente en el futuro.

Nombre de grupo

Users\_appdemo

**¿Cómo quiere crear su grupo de usuarios?**

**Revisar los valores predeterminados**

Comience revisando los valores predeterminados y personalícelos según sus preferencias

**Recorrido por la configuración**

Consulte cada ajuste para elegir las opciones que desee

aws Servicios

Grupos de usuarios | Identidades federadas

## Users\_appdemo

Eliminar el grupo

**Configuración general**

Usuarios y grupos

Atributos

Políticas

MFA y verificaciones

Personalizaciones de mensaje

Etiquetas

Dispositivos

Clientes de aplicación

Desencadenadores

Análisis

**Integración de aplicaciones**

Configuración del cliente de aplicación

Nombre del dominio

Personalización de la interfaz de usuario

Servidores de recursos

**Federación**

Proveedores de identidades

Mapeo de atributos

Su grupo de usuarios se ha creado correctamente.

**Id de grupo** us-east-2\_wNeibioSM

**ARN de grupo** arn:aws:cognito-idp:us-east-2:531697871639:userpool/us-east-2\_wNeibioSM

**Número estimado de usuarios** 0

**Atributos obligatorios** email

**Atributos de alias** ninguna

**Atributos de nombre de usuario** ninguna

**¿Desea habilitar la indistinción de mayúsculas y minúsculas?** Sí

**Atributos personalizados** Elegir atributos personalizados...

**Longitud mínima de la contraseña** 8

**Política de contraseñas** letras mayúsculas, letras minúsculas, caracteres especiales, números

**¿Se permiten los registros de usuario?** Los usuarios pueden inscribirse por sí solos

**dirección de correo electrónico del REMITENTE** Predeterminada

**entrega de correo electrónico a través de Amazon SES** No

Nota: Ha elegido que Cognito envíe mensajes de correo electrónico en su nombre. Las prácticas recomendadas sugieren que los clientes envíen mensajes de correo electrónico a

## 2. Creación de aplicaciones clientes para el grupo creado

The screenshot shows the AWS IAM console interface for the 'Users\_appdemo' group. The left sidebar contains a navigation menu with options like 'Configuración general', 'Usuarios y grupos', 'Atributos', 'Políticas', 'MFA y verificaciones', 'Seguridad avanzada', 'Personalizaciones de mensaje', 'Etiquetas', 'Dispositivos', 'Clientes de aplicación' (highlighted), 'Desencadenadores', 'Análisis', 'Integración de aplicaciones', 'Configuración del cliente de aplicación', 'Nombre del dominio', 'Personalización de la interfaz de usuario', 'Servidores de recursos', 'Federación', and 'Proveedores de identidad'. The main content area is titled '¿Qué clientes de aplicación tendrán acceso a este grupo de usuarios?' and includes a sub-header 'Los clientes de aplicación que añada recibirán un ID único y una clave secreta opcional para obtener acceso a este grupo de usuarios.' Below this, there are three sections for configuring application clients: 'Nombre del cliente de aplicación' (with a text input field containing 'appdemo'), 'Actualizar el vencimiento del token' (with input fields for '30' días and '0' minutos, and a note 'Deben estar comprendidos entre 60 minutos y 3650 días'), and 'Vencimiento del token de acceso' (with input fields for '0' días and '60' minutos, and a note 'Deben estar comprendidos entre 5 minutos y 1 día. No pueden ser mayores que el vencimiento del token de actualización.'). There is also a 'Vencimiento del ID de token' section with input fields for '0' días and '60' minutos, and a note 'Deben estar comprendidos entre 5 minutos y 1 día. No pueden ser mayores que el vencimiento del token de actualización.'

This screenshot shows the same AWS IAM console interface as the previous one, but with the 'ID de cliente de aplicación' field populated with the value '674lom5dukkcmhtfc4i5q9o11'. The 'Nombre del cliente de aplicación' field still contains 'appdemo'. Below the input fields, there is a 'Mostrar detalles' button. At the bottom of the main content area, there are two links: 'Añadir otro cliente de aplicación' and 'Volver a los detalles del grupo'.

## 3. Se registran los datos de UserPoolId y userPoolClientId en la aplicación front end

The screenshot shows a code editor with a file named 'config.js'. The code defines a 'window.config' object with a 'cognito' property. The 'cognito' object contains the following fields: 'userPoolId' (set to 'us-east-2\_wNeibioSM'), 'userPoolClientId' (set to '674lom5dukkcmhtfc4i5q9o11'), and 'region' (set to 'us-east-2'). There is also an 'api' property with an 'invokeUrl' field set to 'https://rc7nvt4tql.execute-api.us-west-2.amazonaws.com/prod'. The code is as follows:

```
1 window.config = {
2   cognito: {
3     userPoolId: 'us-east-2_wNeibioSM', // e.g. us-east-2_uXboG5pAb
4     userPoolClientId: '674lom5dukkcmhtfc4i5q9o11', // e.g. 25ddkmj4v6hfsfvrhpf17n4hv
5     region: 'us-east-2' // e.g. us-east-2
6   },
7   api: {
8     invokeUrl: 'https://rc7nvt4tql.execute-api.us-west-2.amazonaws.com/prod',
9   }
10 };
11
```

4. Se realiza el login desde la aplicación front end y observación la solicitud de creación y confirmación de esta

The screenshot shows the AWS IAM console interface. The left sidebar contains navigation links for 'Configuración general' and 'Integración de aplicaciones'. The main content area is titled 'Users\_appdemo' and has tabs for 'Usuarios' and 'Grupos'. Under the 'Usuarios' tab, there are buttons for 'Importar usuarios' and 'Crear un usuario', and a search bar labeled 'User name'. Below these is a table of users.

Nombre de usuario	Habilitado	Estado de la cuenta	Correo electrónico verificado	Número de teléfono verificado	Actualizado	Creado
sebastian.martinez.rey-at-gmail.com	Enabled	UNCONFIRMED	false	-	Oct 29, 2020 9:08:39 PM	Oct 29, 2020 9:08:39 PM

This screenshot is similar to the previous one, but the user 'sebastian.martinez.rey-at-gmail.com' now has a status of 'CONFIRMED' and 'Correo electrónico verificado' is 'true'. The 'Actualizado' timestamp is also updated to '9:10:29 PM'.

Nombre de usuario	Habilitado	Estado de la cuenta	Correo electrónico verificado	Número de teléfono verificado	Actualizado	Creado
sebastian.martinez.rey-at-gmail.com	Enabled	CONFIRMED	true	-	Oct 29, 2020 9:10:29 PM	Oct 29, 2020 9:08:39 PM

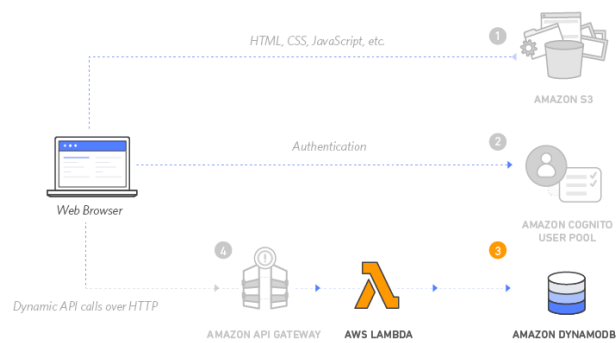
The screenshot shows a verification message box with the following text:

main.dxku8wb2e4fua.amplifyapp.com dice  
Verification successful. You will now be redirected to the login page.

There is a blue button labeled 'Aceptar' at the bottom right of the message box.



## Despliegue Backend (API Gateway, Lambda, DynamoDB)



## DynamoDB

### 1. Creación de tabla

La imagen muestra la consola de AWS Management Console con la pestaña 'Rides' seleccionada. Se visualizan los detalles de la tabla DynamoDB:

- Nombre de la tabla:** Rides
- Clave de partición principal:** RideId (Cadena)
- Clave de ordenación principal:** -
- Recuperación a un momento dado:** DESHABILITADO [Habilitar](#)
- Tipo de cifrado:** PREDETERMINADO [Administrar el cifrado](#)
- ARN de clave principal de KMS:** No aplicable
- Estado de cifrado:** DESHABILITADO [Administrar Contributor Insights](#) [Nuevo](#)
- Atributo de tiempo de vida:** DESHABILITADO [Administrar TTL](#)
- Estado de la tabla:** Activo
- Fecha de creación:** 29 de octubre de 2020, 16:25:49 UTC-5
- Modo de capacidad de lectura/escritura:** Aprovisionado
- Último cambio al modo bajo demanda:** -
- Unidades de capacidad de lectura aprovisionada:** 5 (Auto Scaling Deshabilitados)
- Unidades de capacidad de escritura aprovisionada:** 5 (Auto Scaling Deshabilitados)
- Tiempo de disminución más reciente:** -
- Tiempo de aumento más reciente:** -
- Tamaño de almacenamiento (en bytes):** 0 bytes
- Recuento de elementos:** 0 [Administrar recuento dinámico](#)
- Región:** US East (Ohio)
- Nombre de recurso de Amazon (ARN):** `arn:aws:dynamodb:us-east-2:531697871639:table/Rides`

El tamaño de almacenamiento y el recuento de elementos no se actualizan en tiempo real. Se actualizan de forma periódica, alrededor de cada seis horas.

El formulario 'Añadir ARN' permite especificar el ARN para la tabla. Los campos y valores mostrados son:

- Especificar el ARN para table:** `arn:aws:dynamodb:us-east-2:531697871639:table/Rides`
- Region:** ☒ `us-east-2` ☐ Cualquiera
- Account:** `531697871639` ☐ Cualquiera
- Table name:** `Rides` ☐ Cualquiera

Botones: [Cancelar](#) [Añadir](#)

## 2. Creación de Rol para acceso a la tabla

The screenshot shows the AWS IAM console interface. On the left is a navigation menu with sections like 'Administración del acceso' and 'Informes de acceso'. The main content area is titled 'Resumen' for the role 'WildRydesLambda'. It displays key information: ARN of the role, description, instance profile ARN, path, creation time, last activity, and session duration. Below this, there are tabs for 'Permisos', 'Relaciones de confianza', 'Etiquetas (1)', 'Access Advisor', and 'Revocar las sesiones'. The 'Permisos' tab is active, showing 'Políticas de permisos (1 política aplicada)'. It includes a table with one policy: 'AWSLambdaBasicExecutionRole', which is an AWS managed policy. There is also a section for 'Límite de permisos (no definido)'.

## 3. Se añade una política insertada con el “arn” de la tabla creada

The screenshot shows the 'Crear una política' (Create a policy) page in the AWS IAM console. It has a progress indicator with steps 1 and 2. The page explains that a policy defines AWS permissions and can be created in the visual editor or JSON. The 'Editor visual' tab is selected. It shows a configuration for a policy named 'DynamoDB (1 acción)'. Under 'Servicio', 'DynamoDB' is selected. Under 'Acciones', 'Escritura' (Write) is selected with the action 'PutItem'. Under 'Recursos', 'Específico' (Specific) is selected. A resource ARN is entered: 'arn:aws:dynamodb:us-east-2:53169787:'. There is a checkbox for 'Cualquiera de esta cuenta' (Any of this account). At the bottom, there is a section for 'Condiciones de solicitud' (Request conditions) and a button to 'Añadir permisos adicionales' (Add additional permissions).

#### 4. Se asignan permisos al rol creado para acceso a la tabla

aws

Servicios

ian.martinez @ 5316-9787-1639

Global

Soporte

Identity and Access Management (IAM)

Panel

Administración del acceso

Grupos

Usuarios

**Roles**

Políticas

Proveedores de identidad

Configuración de cuenta

Informes de acceso

Analizador de acceso

Reglas de archivo

Analizadores

Configuración

Informe de credenciales

Actividad de la organización

Políticas de control de servicios (SCP)

Buscar en IAM

Roles > WildRydesLambda

Resumen

Eliminar el rol

ARN de rol

arn:aws:iam::531697871639:role/WildRydesLambda

Descripción del rol

Allows Lambda functions to call AWS services on your behalf. | [Editar](#)

ARN del perfil de instancia

Ruta

/

Hora de creación

2020-10-29 16:36 EST

Última actividad

No se ha accedido en el periodo de seguimiento

Duración máxima de la sesión

1 hora [Editar](#)

Permisos

Relaciones de confianza

Etiquetas (1)

Access Advisor

Revocar las sesiones

Políticas de permisos (2 políticas aplicadas)

Asociar políticas

Añadir una política insertada

Nombre de la política	Tipo de política	
AWSLambdaBasicExecutionRole	Política administrada por AWS	
DynamoDBWriteAccess	Política insertada	

Limite de permisos (no definido)

# Lambda

## 1. Creación de servicio y asociación del rol creado

The screenshot shows the 'Create new function' wizard in the AWS Lambda console. It has three tabs: 'Crear desde cero' (selected), 'Utilice un proyecto', and 'Examine el repositorio de aplicaciones sin servidor'. The 'Crear desde cero' tab is active, showing the 'Información básica' section. The function name is 'RequestUnicorn'. The runtime is set to 'Node.js 12.x'. The permissions section shows the 'Cambiar el rol de ejecución predeterminado' dropdown set to 'Uso de un rol existente', with 'WildRydesLambda' selected. A green notification bar at the top states: 'Se ha creado correctamente la función RequestUnicorn. A partir de ahora, puede cambiar el código y la configuración. Para invocar la función con un evento de prueba, elija la opción "Test".'

**Crear desde cero**  
Empiece con un sencillo ejemplo "Hello World".

**Utilice un proyecto**  
Cree una aplicación Lambda utilizando un código de muestra y los ajustes de configuración predefinidos de casos de uso comunes.

**Examine el repositorio de aplicaciones sin servidor**  
Implemente una aplicación Lambda de ejemplo desde AWS Serverless Application Repository.

**Información básica**

**Nombre de la función**  
Escriba un nombre para describir el propósito de la función.  
RequestUnicorn

Utilice exclusivamente letras, números, guiones o guiones bajos. No incluya espacios.

**Tiempo de ejecución** [Info](#)  
Seleccione el lenguaje que quiere utilizar para escribir la función.  
Node.js 12.x

**Permisos** [Info](#)  
De forma predeterminada, Lambda creará un rol de ejecución con permisos para cargar registros en Amazon CloudWatch Logs. Puede personalizar este rol predeterminado más adelante al agregar los disparadores.

**▼ Cambiar el rol de ejecución predeterminado**

**Rol de ejecución**  
Seleccione un rol que defina los permisos de la función. Para crear un rol personalizado, vaya a la [consola de IAM](#).

☐ Creación de un nuevo rol con permisos básicos de Lambda

☒ Uso de un rol existente

☐ Creación de un nuevo rol desde la política de AWS templates

**Rol existente**  
Seleccione un rol existente que haya creado para usarlo con esta función de Lambda. El rol debe tener permiso para cargar registros en Amazon CloudWatch Logs.

WildRydesLambda

[Consulte el rol WildRydesLambda en la consola de IAM.](#)

## 2. Se añade el código correspondiente de la función

The screenshot shows the AWS Lambda console with the 'RequestUnicorn' function selected. A green notification bar at the top states: 'Se ha creado correctamente la función RequestUnicorn. A partir de ahora, puede cambiar el código y la configuración. Para invocar la función con un evento de prueba, elija la opción "Test".'

**RequestUnicorn**

Limitación Cualificadores Acciones Seleccionar un evento d... Probar

Layers (0)

+ Añadir desencadenador + Agregar destino

**Código de la función** [Info](#) **Deploy** Acciones

File Edit Find View Go Tools Window Test Deploy

Environment

RequestUnicorn / index.js

```
43 const requestBody = JSON.parse(event.body);
44
45 const pickupLocation = requestBody.PickupLocation;
46
47 const unicorn = findUnicorn(pickupLocation);
48
49 recordRide(rideId, username, unicorn).then(() => {
50   // You can use the callback function to provide a return value from your Node.js
51   // Lambda functions. The first parameter is used for failed invocations. The
52   // second parameter specifies the result data of the invocation.
53
54   // Because this Lambda function is called by an API Gateway proxy integration
55   // the result object must use the following structure.
56   callback(null, {
57     statusCode: 201,
58     body: JSON.stringify({
59       RideId: rideId,
60       Unicorn: unicorn,
61       Eta: '30 seconds',
62       Rider: username,
63     }),
64     headers: {
65       'Access-Control-Allow-Origin': '*',
66     },
67   });
68 });
```

### 3. Se crea un evento para realizar la prueba de la función y observar su respuesta

### Configurar un evento de prueba

Una función puede tener un máximo de 10 eventos de prueba. Los eventos se conservan hasta que cambia de equipo o navegador web y prueba la función con los mismos eventos.

☒ Crear un evento de prueba nuevo  
☐ Editar eventos de prueba guardados

Plantilla de eventos  
hello-world

Nombre del evento  
TestRequestEvent

```
1 {  
2   "path": "/ride",  
3   "httpMethod": "POST",  
4   "headers": {  
5     "Accept": "*/*",  
6     "Authorization": "eyJraWQ1O1JLTzRVWmZs",  
7     "content-type": "application/json; charset=UTF-8"  
8   },  
9   "queryStringParameters": null,  
10  "pathParameters": null,  
11  "requestContext": {  
12    "authorizer": {  
13      "claims": {  
14        "cognito:username": "the_username"  
15      }  
16    }  
17  },  
18  "body": "{\"PickupLocation\":{\"Latitude\":\"47.6174755835663\",\"Longitude\":\"-122.288370666501\"}}"  
19 }
```

Cancelar Formato JSON Crear

Se ha creado correctamente la función RequestUnicorn. A partir de ahora, puede cambiar el código y la configuración. Para invocar la función con un evento de prueba, elija la opción "Test".

RequestUnicorn

Limitación Cualificadores Acciones TestRequestEvent Probar

Resultado de la ejecución: correcta (Registros)

Detalles

En el área siguiente, se muestra el resultado devuelto por la ejecución de la función. Obtenga más información sobre cómo devolver los resultados de la función.

```
{  
  "statusCode": 201,  
  "body": "{\"rideId\":\"w-rxK5sSHRFND_8ikQyVNA\",\"Unicorn\":  
{\"Name\":\"Shadowfax\",\"Color\":\"White\",\"Gender\":\"Male\"},\"Eta\":\"30 seconds\",\"Rider\":\"the_username\"}\",  
  \"headers\": {  
    \"Access-Control-Allow-Origin\": \"*\"  
  }  
}
```

Resumen

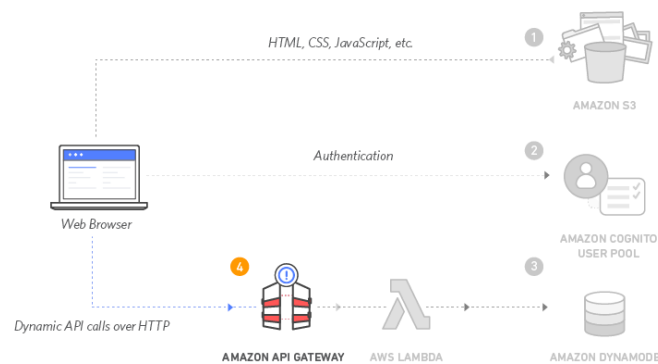
Código SHA-256	ID de solicitud
D485KOABIHbtuH1bXhMs4FhzqbPwvyKouy2Dwvyvq64=	195a79ad-61bb-48ad-811c-91f70062bbc6
Duración	Duración facturada
396.33 ms	400 ms
Recursos configurados	Memoria máx. utilizada
128 MB	86 MB

Resultado de registro

En la siguiente sección, se muestran las llamadas de registro del código. Se corresponden con una sola fila del grupo de registros de CloudWatch para esta función de Lambda. Haga clic aquí para ver el grupo de registros de CloudWatch.

```
START RequestId: 195a79ad-61bb-48ad-811c-91f70062bbc6 Version: $LATEST  
2020-10-29T21:55:16.178Z 195a79ad-61bb-48ad-811c-91f70062bbc6 INFO Received event ( w-rxK5sSHRFND_8ikQyVNA ): {  
  path: '/ride',  
  httpMethod: 'POST',  
  headers: {  
    Accept: '*/*',  
    Authorization: 'eyJraWQ1O1JLTzRVWmZs',  
    'content-type': 'application/json; charset=UTF-8'  
  },  
  ...
```

## API GATEWAY



### 1. Creamos el API Gateway

aws Servicios

Amazon API Gateway API > Crear

### Crear API nueva

En Amazon API Gateway, una API hace referencia a una colección de recursos y métodos que se pueden invocar a través de una URL.

☒ API nueva ☐ Importar de Swagger u Open API 3 ☐ API de ejemplo

### Configuración

Elija un nombre o una descripción fáciles de recordar para su API.

Nombre de API\* WildRydes

Descripción

Tipo de punto de enlace Optimizado para límites ⓘ

### 2. Se crea una autorización para acceder al servicio indicando que se utilizara el servicio Cognito para la validación de usuarios.

aws Servicios

Amazon API Gateway API > WildRydes (twmg2nm4r5) > Autorizadores

### Autorizadores

Los autorizadores le permiten controlar el acceso a sus API con los grupos de usuarios de Cognito.

[+ Crear autorizador nuevo](#)

### Crear autorizador

Nombre \* WildRydes

Tipo \* ⓘ

☐ Lambda ☒ Cognito

Grupo de usuarios de Cognito \* ⓘ

us-east-2 WildRydes

Origen de token \* ⓘ Validación de token ⓘ

Authorization (Autorización)

### 3. Se crea un nuevo recurso el cual recibirá las peticiones que se realicen desde el front end

The screenshot shows the AWS API Gateway console. The breadcrumb trail is: API > WildRydes (twmg2nm4r5) > Recursos > / (09n2f1uorj) > Crear. The left sidebar shows the API 'WildRydes' and the 'Recursos' (Resources) section. The main content area is titled 'Nuevo recurso secundario' (New secondary resource). It contains a text box for the resource name, which is 'ride', and a text box for the resource path, which is '/ride'. There is a checkbox for 'Configurar como recurso de proxy' (Configure as proxy resource) which is unchecked. Below the path field, there is a note: 'Puede agregar parámetros de ruta utilizando llaves. Por ejemplo, la ruta de recurso {username} representa un parámetro de ruta denominado "username". Al configurar /(proxy+) como un recurso de proxy, se recuperan todas las solicitudes en sus recursos secundarios. Por ejemplo, funciona en una solicitud GET a /foo. Para controlar las solicitudes a /, agregue un nuevo método ANY en el recurso /.' At the bottom, there is a checkbox for 'Habilitar API Gateway CORS' (Enable API Gateway CORS) which is checked.

The screenshot shows the AWS API Gateway console. The breadcrumb trail is: API > WildRydes (twmg2nm4r5) > Recursos > /ride (kyv7kn). The left sidebar shows the API 'WildRydes' and the 'Recursos' (Resources) section. The main content area is titled '/ride Métodos' (ride Methods). It shows a list of methods for the resource '/ride', with 'OPTIONS' selected. The 'OPTIONS' method details are shown on the right, including the 'Punto de enlace de simulación' (Simulation endpoint) and the 'Autorización' (Authorization) set to 'NINGUNA' (None).

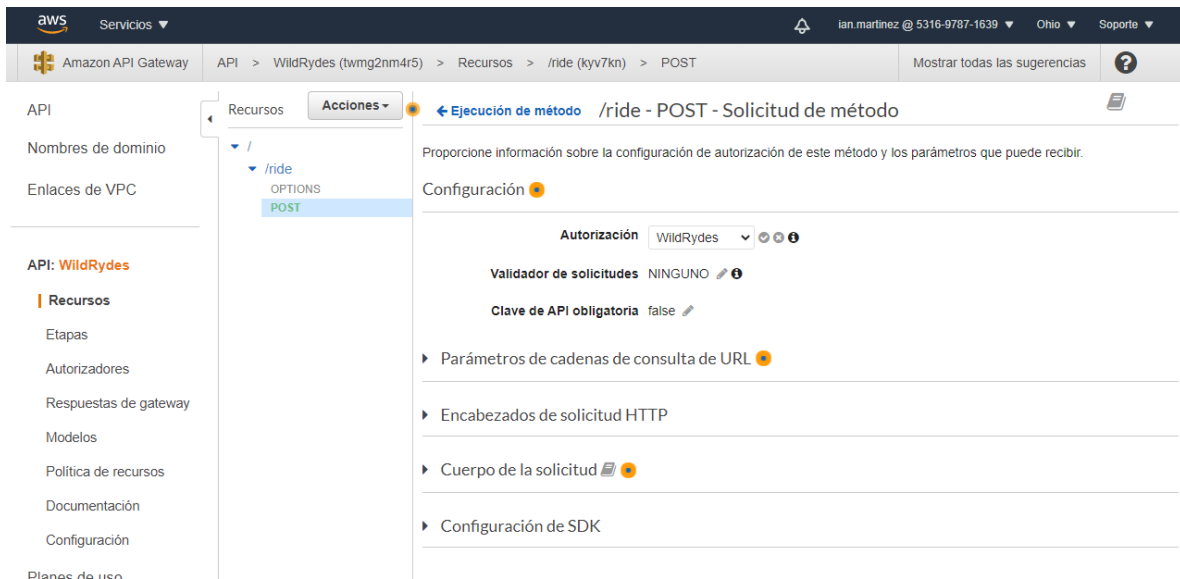
### 4. Agregamos un nuevo método POST al recurso creado

The screenshot shows the AWS API Gateway console. The breadcrumb trail is: Recursos > Acciones > /ride - POST - Configuración. The left sidebar shows the API 'WildRydes' and the 'Recursos' (Resources) section. The main content area is titled '/ride - POST - Configuración'. It contains a text box for the integration point, which is 'RequestUnicorn'. There are several checkboxes: 'Tipo de integración' (Integration type) with 'Función Lambda' (Lambda function) selected; 'Usar la integración de proxy' (Use proxy integration) which is checked; 'Usar tiempo de espera predeterminado' (Use default timeout) which is checked. The 'Región Lambda' (Lambda region) is set to 'us-east-2'.

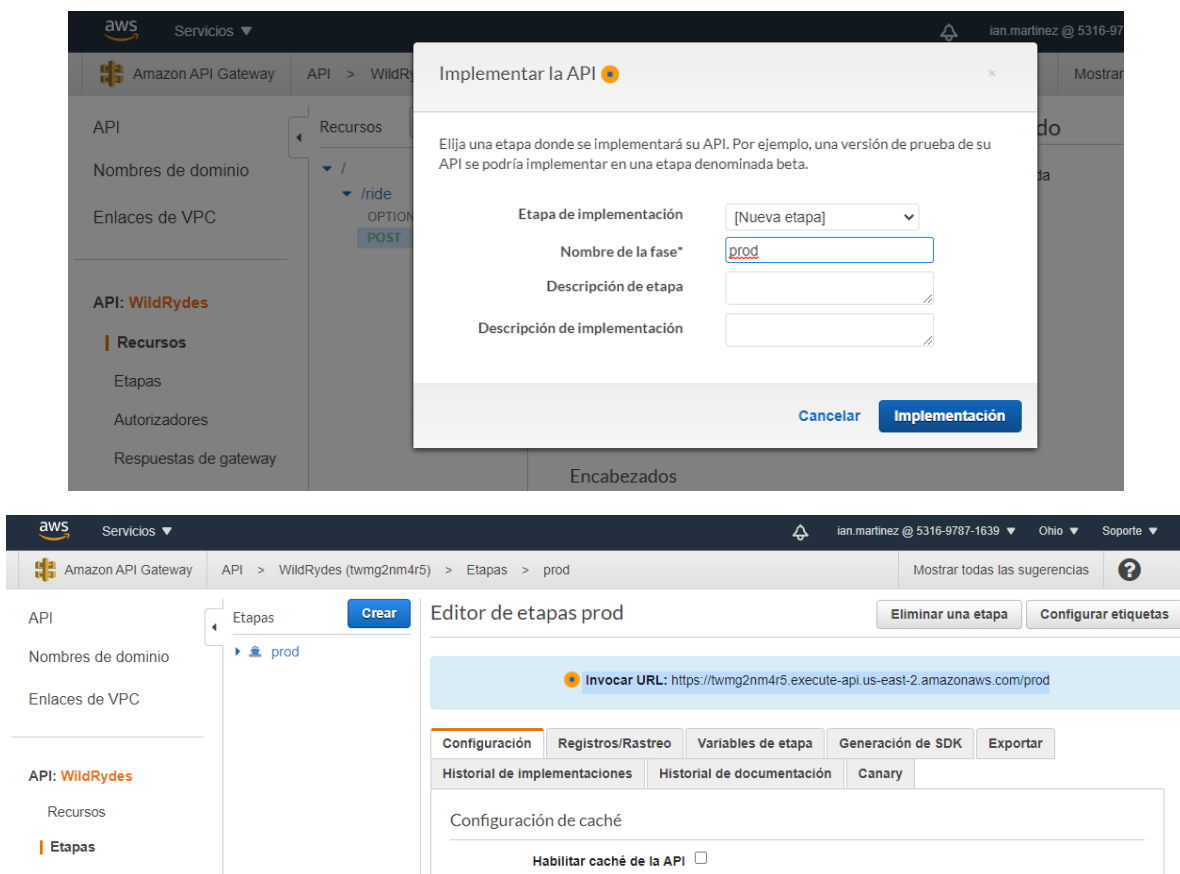
Agregar permiso a la función Lambda

Va a conceder permiso a API Gateway para invocar la función Lambda:  
arn:aws:lambda:us-east-2:531697871639:function:RequestUnicorn

5. Una vez creado el método, indicamos que la autorización se realizará por el grupo creado en el servicio Cognito.



6. Por último, se realiza la implementación del API para producción la cual nos indicara la url que se debe invocar para utilizar el servicio





7. Finalmente, se actualiza la URL de invocación en la aplicación front end culminando el proceso

```
1 window._config = {  
2   cognito: {  
3     userPoolId: 'us-east-2_wNeibioSM', // e.g. us-east-2_uXboG5pAb  
4     userPoolClientId: '674lom5dukkcmhtflc4i5q9oll', // e.g. 25ddkmj4v6hfsfvruhpf17n4hv  
5     region: 'us-east-2' // e.g. us-east-2  
6   },  
7   api: {  
8     invokeUrl: 'https://twmq2nm4r5.execute-api.us-east-2.amazonaws.com/prod'  
9   }  
10  };
```