

## 1. CREACIÓN DE LA POLÍTICA CUSTOM PARA SECRET MANAGER

- Crear la política custom para secrets manager: esta política permitirá a la función Lambda leer secretos de AWS Secrets Manager.

### Política JSON (SecretsManagerReadOnlyPolicyTuNombre):

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue",
        "secretsmanager:DescribeSecret"
      ],
      "Resource": "*"
    }
  ]
}
```

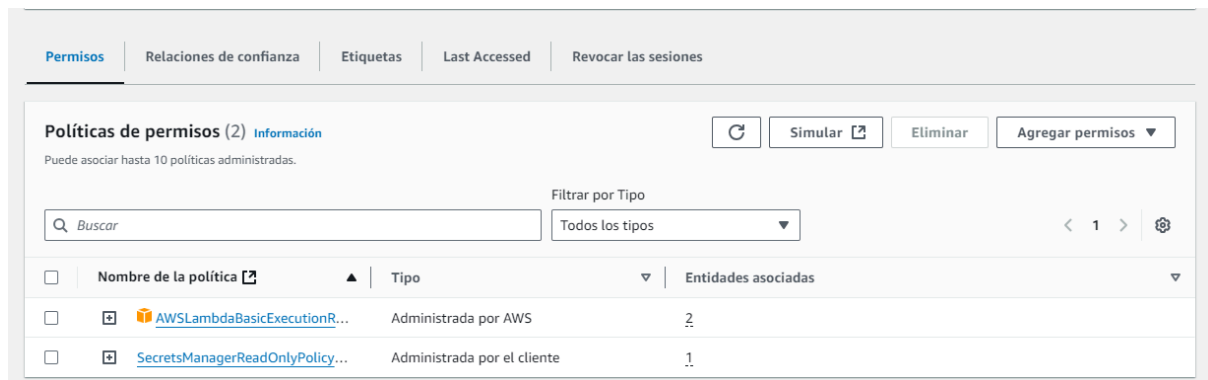
## 2. Crear el Rol para la Lambda: Este rol permitirá que la función Lambda acceda a AWS Secrets Manager y escriba logs en Amazon CloudWatch.

### Pasos:

- Accede a la consola de IAM: En la consola de AWS, ve a IAM.
- Crear un rol para Lambda: En el menú lateral, selecciona Roles. Haz clic en Crear rol. Selecciona Lambda como tipo de entidad de confianza. Haz clic en Siguiente.

The screenshot shows the AWS IAM console interface for a role named 'LambdasecretsmanagerRoleYuriangel'. The breadcrumb navigation at the top reads 'IAM > Roles > LambdasecretsmanagerRoleYuriangel'. The role name is displayed with an 'Información' link and an 'Eliminar' button. Below this, a description states: 'Allows Lambda functions to call AWS services on your behalf.' A 'Resumen' section contains an 'Editar' button and a table with the following details:

Fecha de creación October 03, 2024, 18:44 (UTC-05:00)	ARN arn:aws:iam::137068251696:role/LambdasecretsmanagerRoleYuriangel
Última actividad hace 2 horas	Duración máxima de la sesión 1 hora



### 3. Crear el Secreto en Secrets Manager: Aquí almacenarás la API Key para la API pública que estás usando.

#### Pasos:

- Accede a la consola de Secrets Manager: Ve a la consola de AWS y busca Secrets Manager.

- Crear un nuevo secreto: Haz clic en Store a new secret.

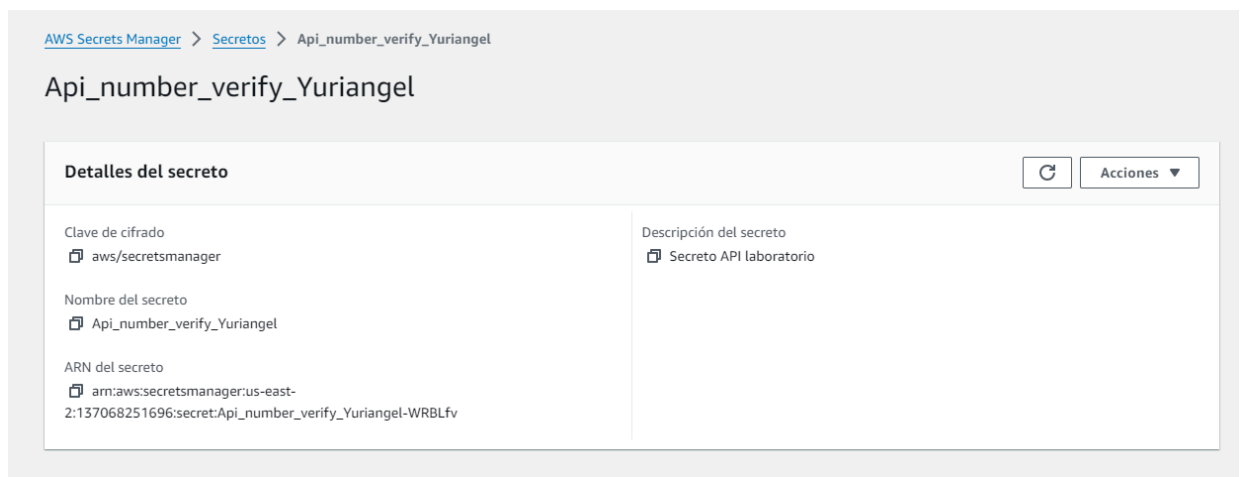
En Secret type, selecciona Other type of secret.

En Key/value pairs, añade: Key: api\_key

Value: El valor que obtuviste de la página de NumVerify.

Para obtener este value debes registrarte en <https://numverify.com/>

Seleccionas el plan free



### 4. Crear la Función Lambda: Esta será la función Lambda que utilizará el secreto y hará la solicitud HTTP a la API pública.

#### Pasos:

- Accede a la consola de Lambda: En la consola de AWS, busca Lambda.

- Crear una nueva función: Haz clic en Create function.

Selecciona Author from scratch.

Dale un nombre, por ejemplo: ValidatePhoneNumberTuNombre.

En Runtime, selecciona Node.js 20.x.  
 En Permissions, selecciona Use an existing role y elige el rol  
 LambdaSecretsManagerRoleTuNombre (creado previamente).  
 Haz clic en Create function.

The screenshot shows the AWS Lambda console's 'Código fuente' (Source code) tab. The function name is 'validatePhone\_Number\_Yuriangel'. The code is written in JavaScript (index.mjs) and uses the AWS SDK for Node.js to interact with the AWS Secrets Manager. The code imports 'SecretsManagerClient' and 'GetSecretValueCommand', creates a client for the 'us-east-2' region, and defines an async handler. The handler logs an attempt to retrieve a secret named 'Api\_number\_verify\_Yuriangel', obtains the secret, parses it as JSON, and extracts the 'api\_key'. It then uses these values to construct a response object with 'phoneNumber', 'countryCode', and 'formatParam' fields. The code is as follows:

```

1 import { SecretsManagerClient, GetSecretValueCommand } from "@aws-sdk/client-secrets-manager";
2 const client = new SecretsManagerClient({ region: "us-east-2" });
3
4 export const handler = async (event) => {
5   const secretName = 'Api_number_verify_Yuriangel';
6   console.log('Attempting to retrieve API secret');
7   try {
8     // Obtener la clave de API de Secrets Manager
9     const command = new GetSecretValueCommand({ SecretId: secretName });
10    const data = await client.send(command);
11    console.log('API secret retrieved successfully');
12    const secret = JSON.parse(data.SecretString);
13    const accessKey = secret.api_key;
14
15    // Extraer parámetros dinámicos del evento
16    const phoneNumber = event.phoneNumber || "14158586273";
17    const countryCode = event.countryCode || "";
18    const formatParam = event.formatParam || 1;
  
```

The screenshot shows the 'Probar' (Test) tab in the AWS Lambda console. It indicates that the function was executed successfully without errors. Below this, there is a section for 'Detalles' (Details) showing the last 4 KB of the execution log. The log contains a JSON object representing the test event. Below the log, there is a 'Resumen' (Summary) section with the following information:

Resumen	
Código SHA-256	Tiempo de ejecución
0W8VM6z2Bg+49TfQsLud2P4c4C7rA5XTUjjiyp18n7QU=	hace 53 segundos
ID de solicitud	Versión de la función
96538e7b-00a5-4545-b2fe-286639566f6e	\$LATEST

**Probar la función Lambda:** Crea un evento de prueba con el siguiente JSON:

```

{
  "phoneNumber": "3114868041",
  "countryCode": "co"
}
  
```

## Evento JSON

```

1 {
2   "phoneNumber": "3114868041",
3   "countryCode": "CO"
4 }
  
```

Ejecuta la función y revisa los logs en Amazon CloudWatch para verificar que todo esté funcionando correctamente

<

Flujos de registros

Etiquetas

Detección de anomalías

Filtros de métricas

Filtros de suscripción

Flujos de registros (5)

Eliminar

Crear flujo de registros

Buscar en todas las secuencias

Filtrar flujos de registro o probar la búsqueda de prefijos

☐ Coincidencia exacta

☐ Mostrar caducado

[Información](#)

<input type="checkbox"/>	Secuencia de registro	Hora del último evento
<input type="checkbox"/>	<a href="#">2024/10/04/[\$LATEST]de6485ae4ddb4187baae846a3aac16f0</a>	2024-10-04 03:28:28 (UTC)
<input type="checkbox"/>	<a href="#">2024/10/04/[\$LATEST]bcf7d60e9aba48cc9d2499a288e4cf47</a>	2024-10-04 00:34:07 (UTC)
<input type="checkbox"/>	<a href="#">2024/10/04/[\$LATEST]c62a984793c7400395d45a437947fd51</a>	2024-10-04 00:22:44 (UTC)
<input type="checkbox"/>	<a href="#">2024/10/04/[\$LATEST]c798671da9ce4f14aa103d4b7af3fd0f</a>	2024-10-04 00:18:15 (UTC)
<input type="checkbox"/>	<a href="#">2024/10/04/[\$LATEST]b68ffcb756041d69d11bb7d10b010eb</a>	2024-10-04 00:17:12 (UTC)

▶

Marca temporal

Mensaje

Hay eventos antiguos para cargar. [Cargar más.](#)

▼

2024-10-04T00:18:14.335Z

INIT\_START Runtime Version: nodejs:20.v39 Runtime Version ARN: arn:aws:lambda:us-east-2::runtime:ad...  
INIT\_START Runtime Version: nodejs:20.v39 Runtime Version ARN: arn:aws:lambda:us-east-2::runtime:ad...  
Expandir fila Expandir el evento de registro Thu Oct 03 2024 19:18:14 GMT-0500 (Colombia Standard Time)

▶

2024-10-04T00:18:14.476Z

2024-10-04T00:18:14.476Z undefined ERROR Uncaught Exception {"errorType":"Runtime.UserCodeSyntaxEr...

▶

2024-10-04T00:18:14.495Z

INIT\_REPORT Init Duration: 160.57 ms Phase: init Status: error Error Type: Runtime.UserCodeSyntaxEr...

▶

2024-10-04T00:18:15.681Z

2024-10-04T00:18:15.681Z undefined ERROR Uncaught Exception {"errorType":"Runtime.UserCodeSyntaxEr...

▶

2024-10-04T00:18:15.960Z

INIT\_REPORT Init Duration: 1447.96 ms Phase: invoke Status: error Error Type: Runtime.UserCodeSynt...

▶

2024-10-04T00:18:15.960Z

START RequestId: 667f5f8b-de84-4348-948d-2675fd6fd2a5 Version: \$LATEST

▶

2024-10-04T00:18:15.972Z

END RequestId: 667f5f8b-de84-4348-948d-2675fd6fd2a5

▶

2024-10-04T00:18:15.973Z

REPORT RequestId: 667f5f8b-de84-4348-948d-2675fd6fd2a5 Duration: 1459.59 ms Billed Duration: 1460 ...

No hay eventos recientes en este momento. *Reintento automático en pausa.* [Reanudar](#)

Volver arriba ^