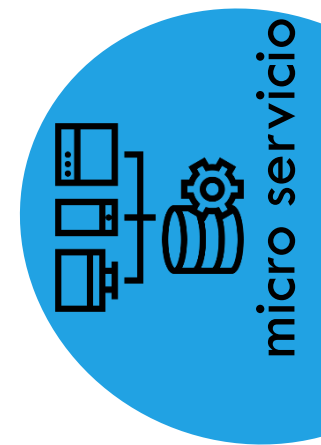
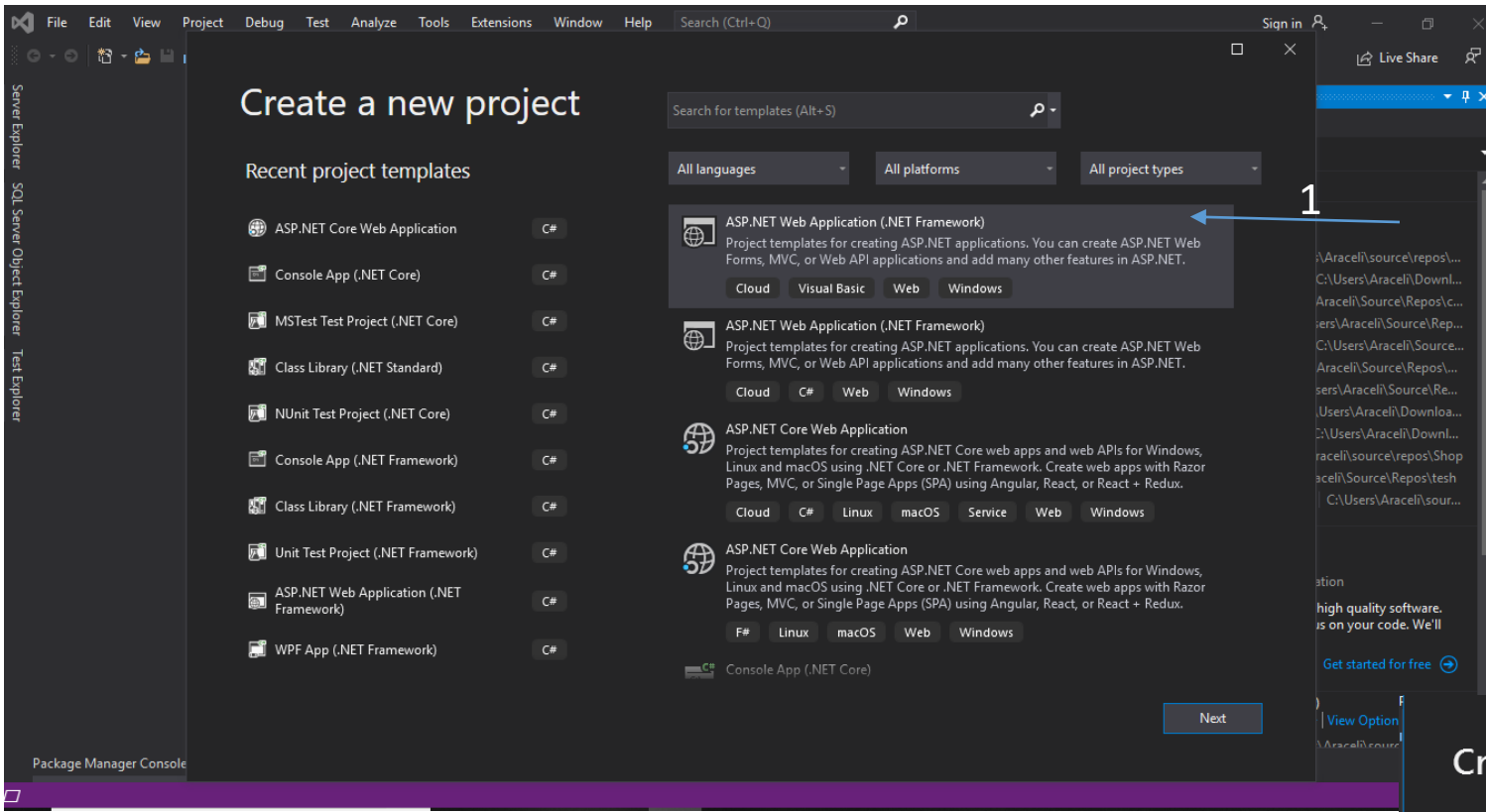
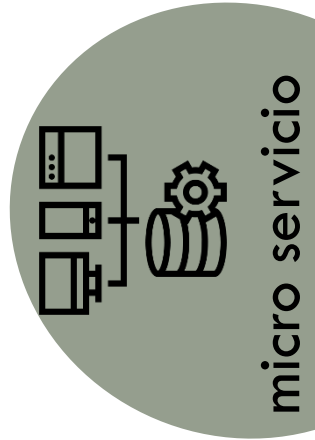


Consumo de Microservicio



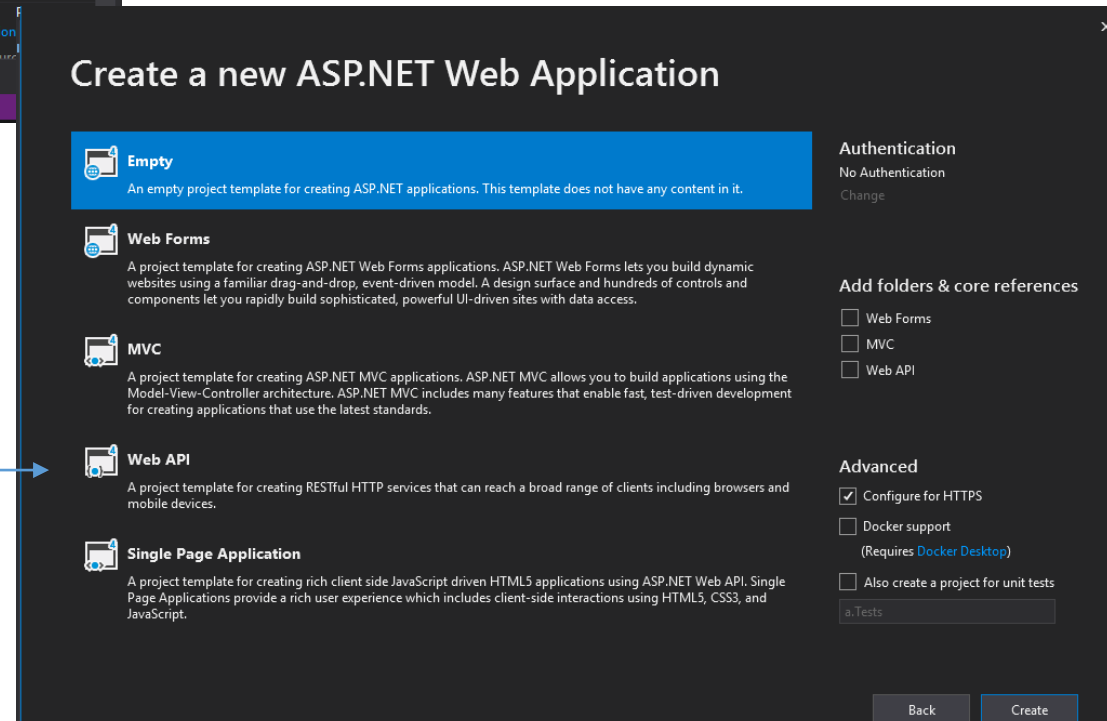


1.-Crear proyecto Web Application



2.-Seleccionar web API

2



AraceliCA

File Edit View Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q)

Debug Any CPU IIS Express

Server Explorer Toolbox SQL Server Object Explorer Test Explorer

HelloController.cs

AraceliCA AraceliCA.Controllers.HelloController Get()

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading.Tasks;
5 using Microsoft.AspNetCore.Http;
6 using Microsoft.AspNetCore.Mvc;
7
8 namespace AraceliCA.Controllers
9 {
10     [Route("api/sps/helloworld/v1")]
11     [ApiController]
12     public class HelloController : ControllerBase
13     {
14         // GET: api/Hello
15         [HttpGet]
16         public IEnumerable<string> Get()
17         {
18             return new string[] { "Araceli", "Carbajal" };
19         }
20
21         // GET: api/Hello/5
22         [HttpGet("{id}", Name = "Get")]
23         public string Get(int id)
24         {
25             return "valor enviado " + id;
26         }
27     }
28 }
29
```

1 Ruta

2 Envía dato

3 Recibe ID

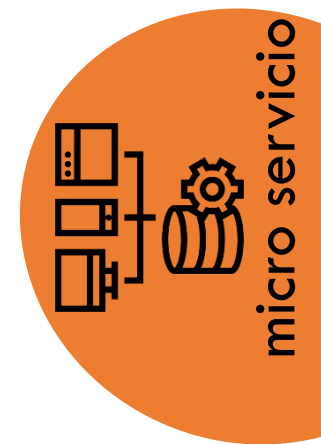
Creación de controller

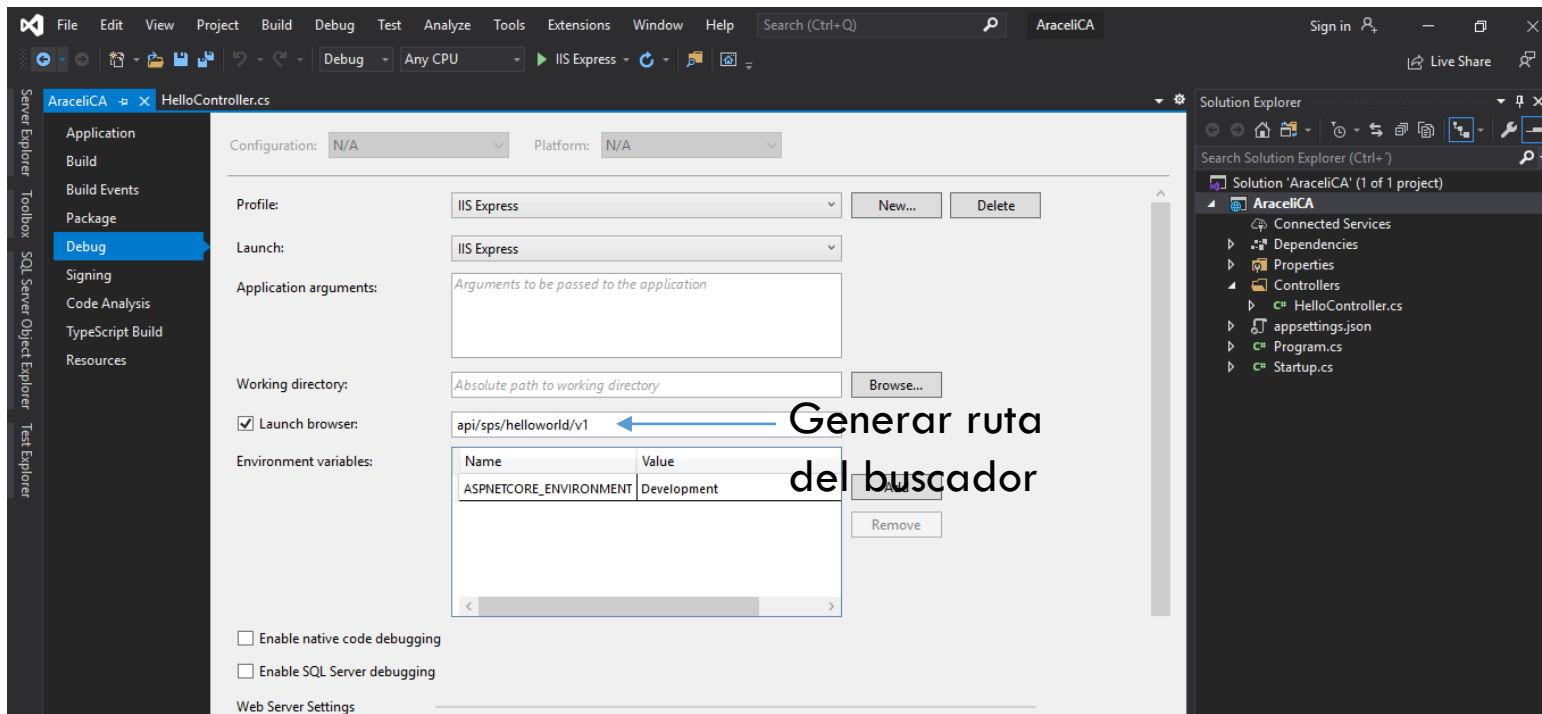
Solution Explorer

Search Solution Explorer (Ctrl+)

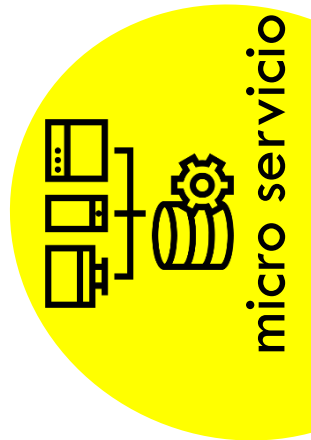
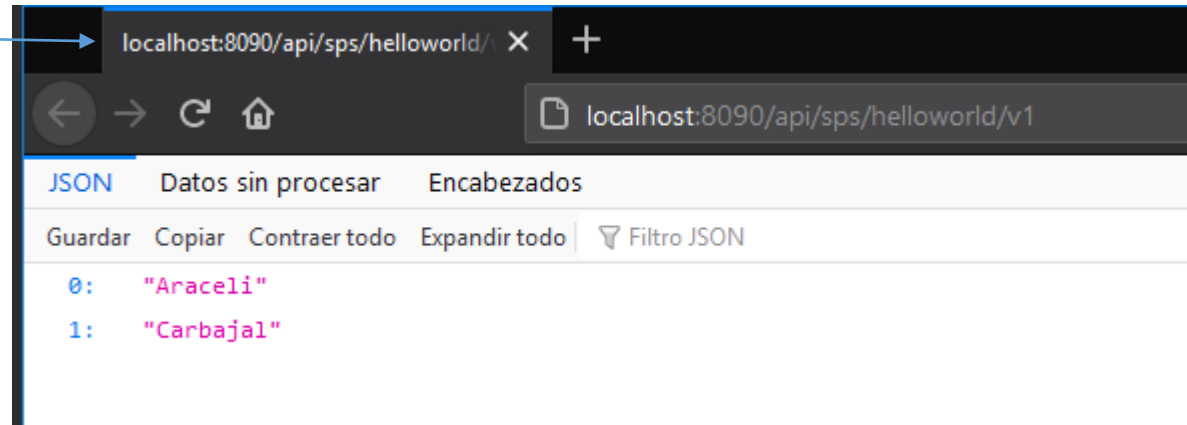
Solution 'AraceliCA' (1 of 1 project)

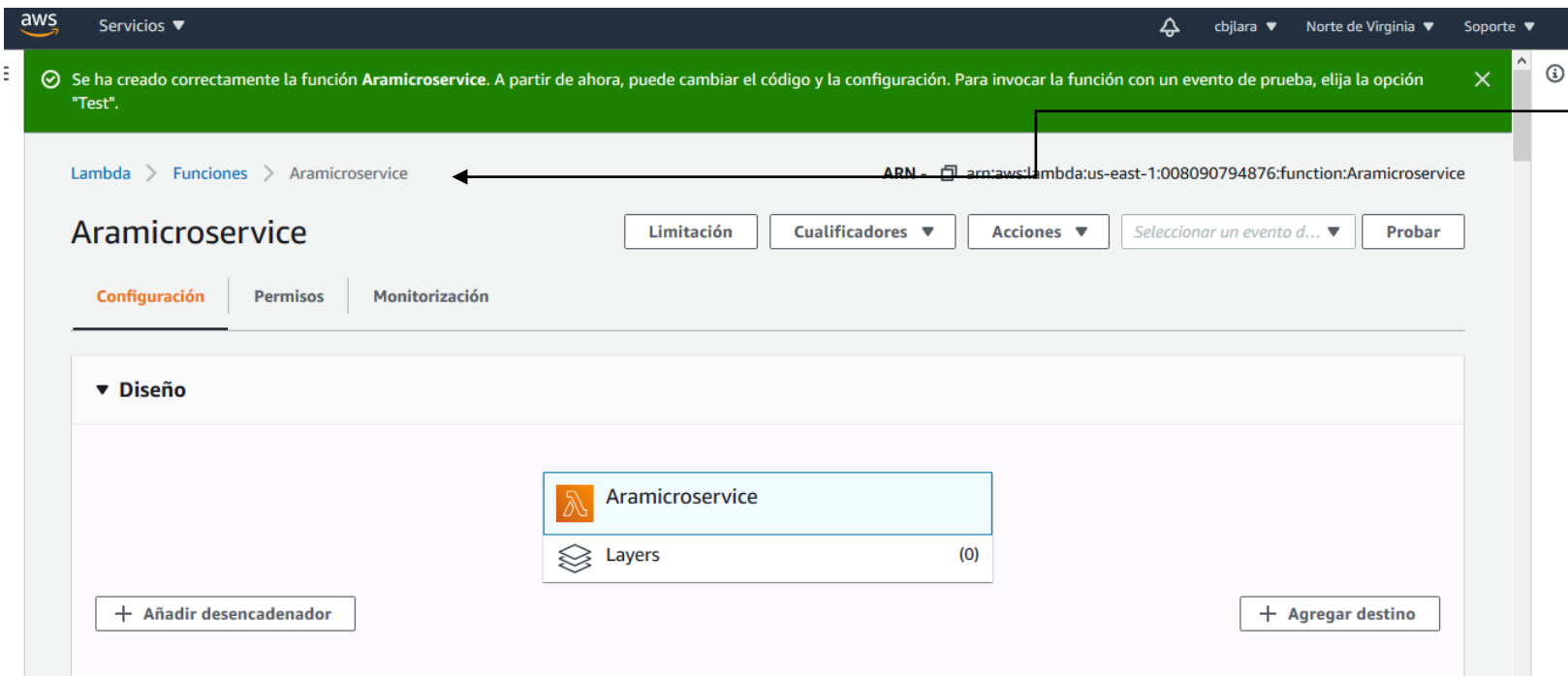
- AraceliCA
 - Connected Services
 - Dependencies
 - Properties
 - Controllers
 - HelloController.cs
 - appsettings.json
 - Program.cs
 - Startup.cs



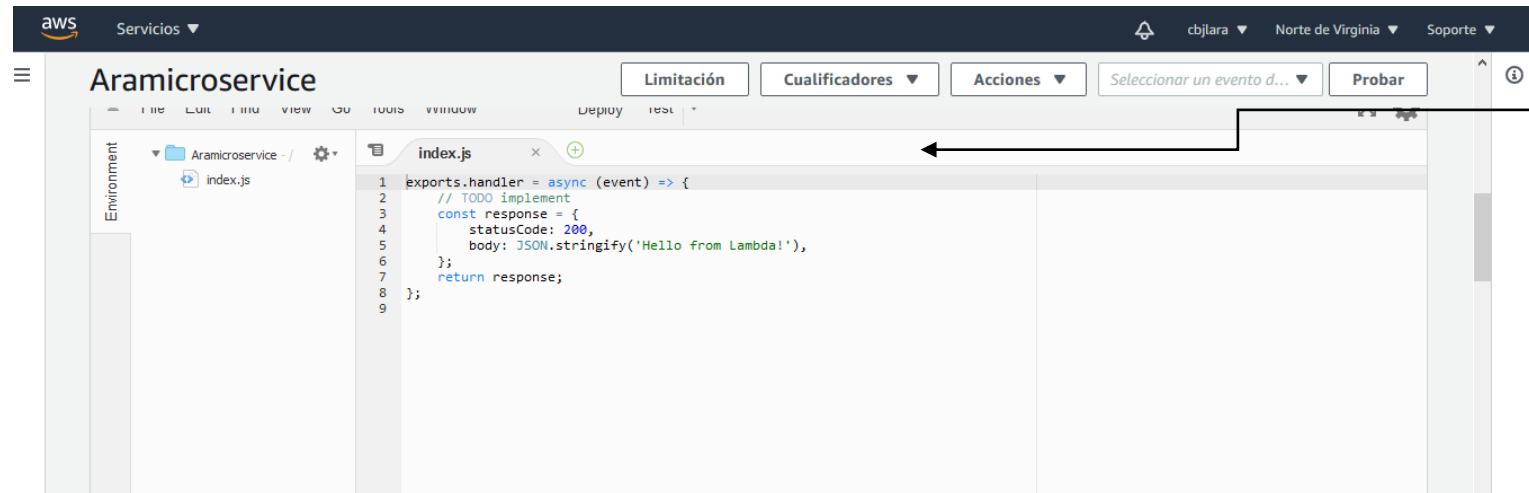


Ubicación de
localhost y ruta

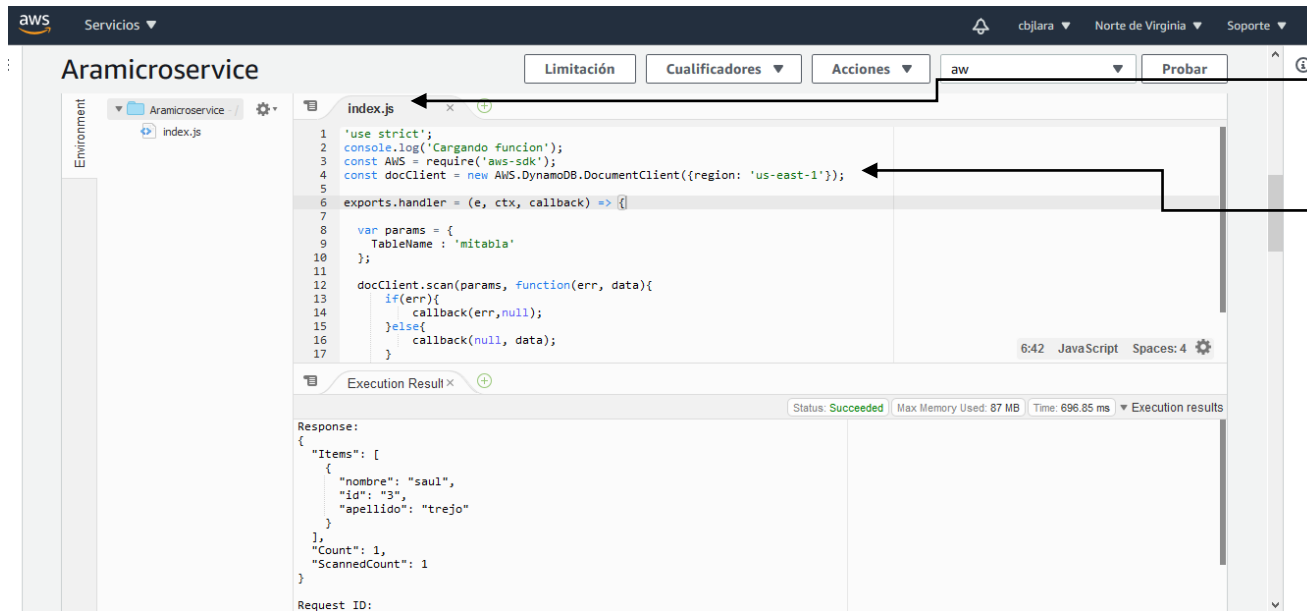




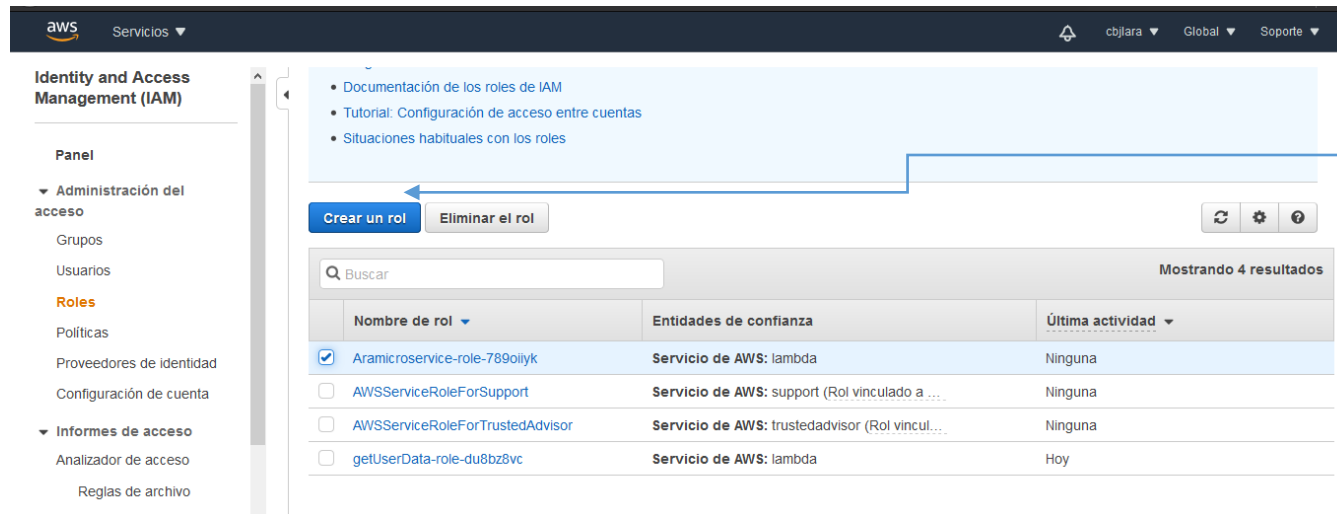
Elastic EC2 Container Service AWS.
Creamos microservicio en Lambda



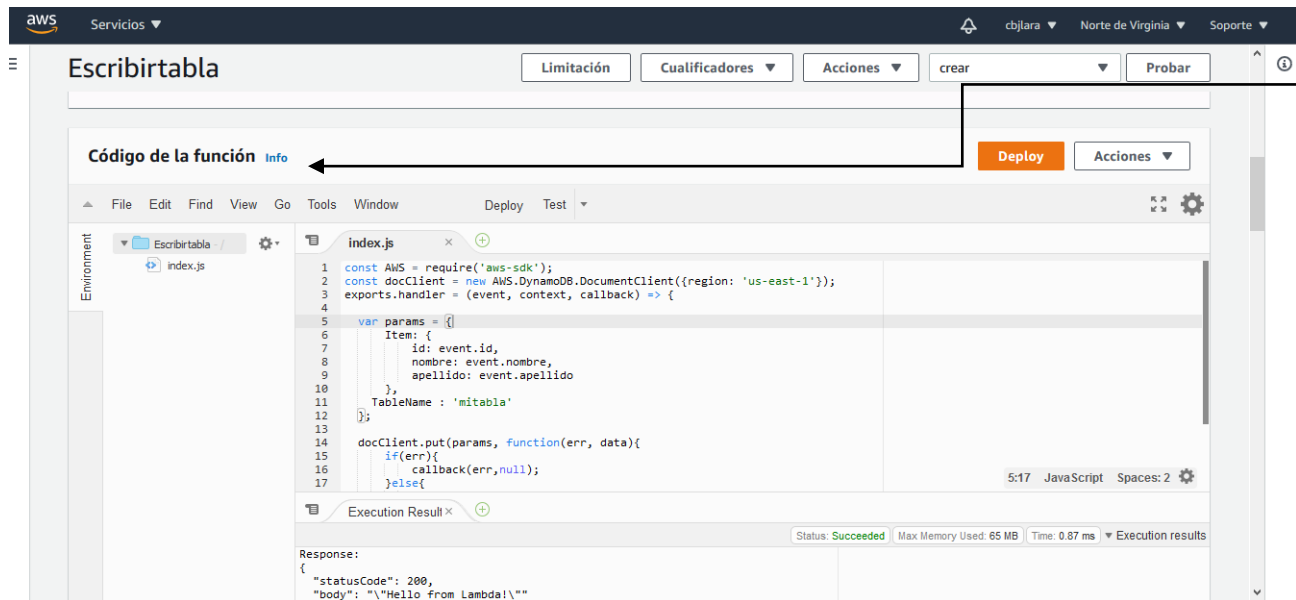
Generado exitosamente



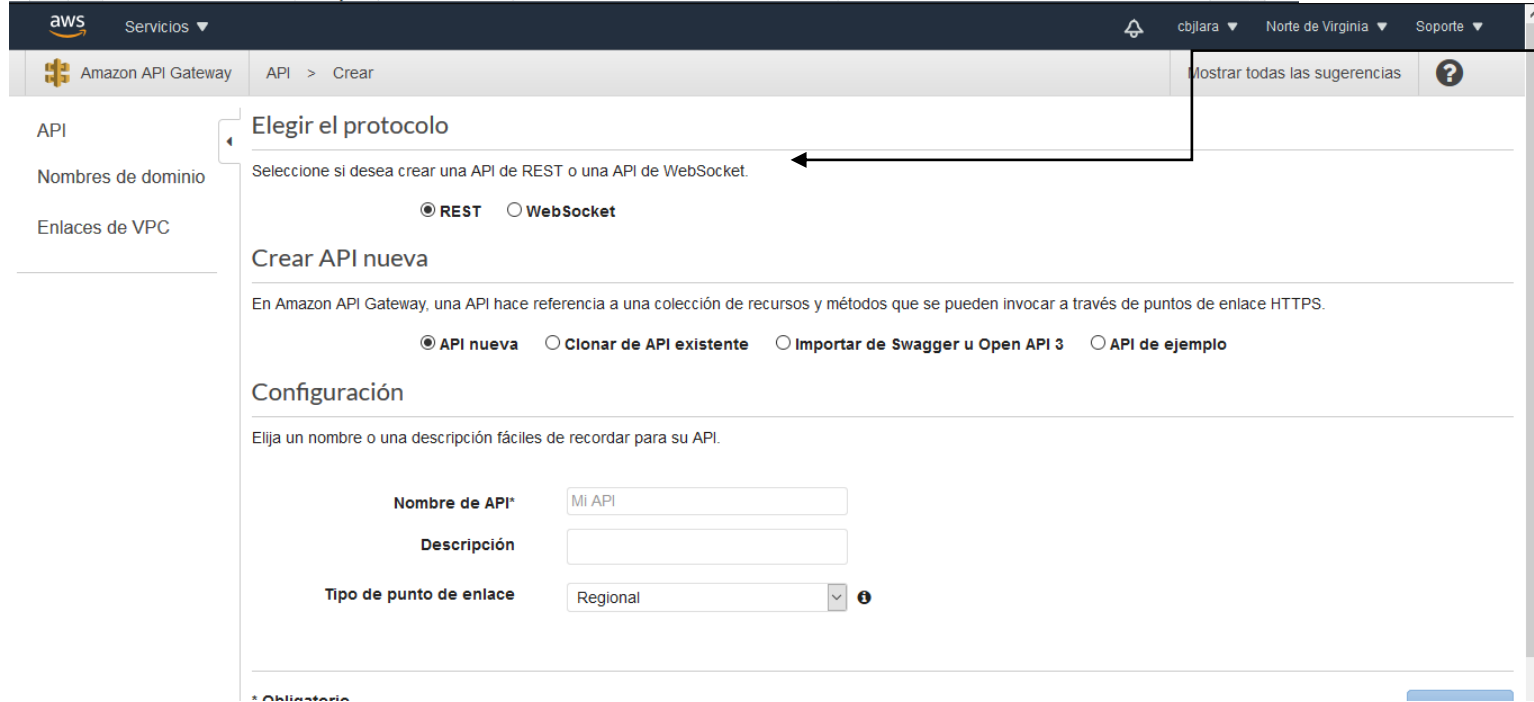
- 1.-Construimos microservice (AWS) en Lambda y con este código leemos los registros,
- 2.-referenciando ubicación geográfica



Nos dirigimos a IAM, click en Roles, seleccionar Rol creado, para poder tener permisos de DB



Creamos en Lambda un servicio para poder agregar nuevo registro



Creamos Api, desde cero

aws Servicios

API

Nombres de dominio

Enlaces de VPC

API: API2

Recursos

Etapas

Autorizadores

Respuestas de gateway

Modelos

Política de recursos

Documentación

Configuración

Planes de uso

Claves de API

Recursos Acciones /tabla - GET - Configuración

Elija el punto de integración del nuevo método.

Tipo de integración ☒ Función Lambda *i*

☐ HTTP *i*

☐ Simulación *i*

☐ Servicio de AWS *i*

☐ Enlace de VPC *i*

Usar la integración de proxy Lambda ☐ *i*

Región Lambda us-east-1

Función Lambda Aramicroservice *i*

Usar tiempo de espera predeterminado ☒ *i*

Guardar

Marcar las propiedades correspondientes

aws Servicios

Amazon API Gateway

API > API2 (yj9hf6k57g) > Recursos > /tabla (bdo17x) > GET

Mostrar todas las sugerencias ?

API

Nombres de dominio

Enlaces de VPC

API: API2

Recursos

Etapas

Autorizadores

Respuestas de gateway

Modelos

Política de recursos

Documentación

Configuración

Planes de uso

Recursos Acciones < Ejecución de método /tabla - GET - Prueba de método

Realice una llamada de prueba a su método con la entrada proporcionada

Ruta

No hay ningún parámetro de ruta para este recurso. Puede definir los parámetros de ruta mediante la sintaxis {myPathParam} en una ruta de recursos.

Cadenas de consulta

(tabla)

param1=value1¶m2=value2

Encabezados

(tabla)

Utilice dos puntos (:) para separar el nombre y el valor del encabezado, y líneas nuevas para declarar varios encabezados; por ejemplo, Accept: application/json.

Solicitud: /tabla

Estado: 200

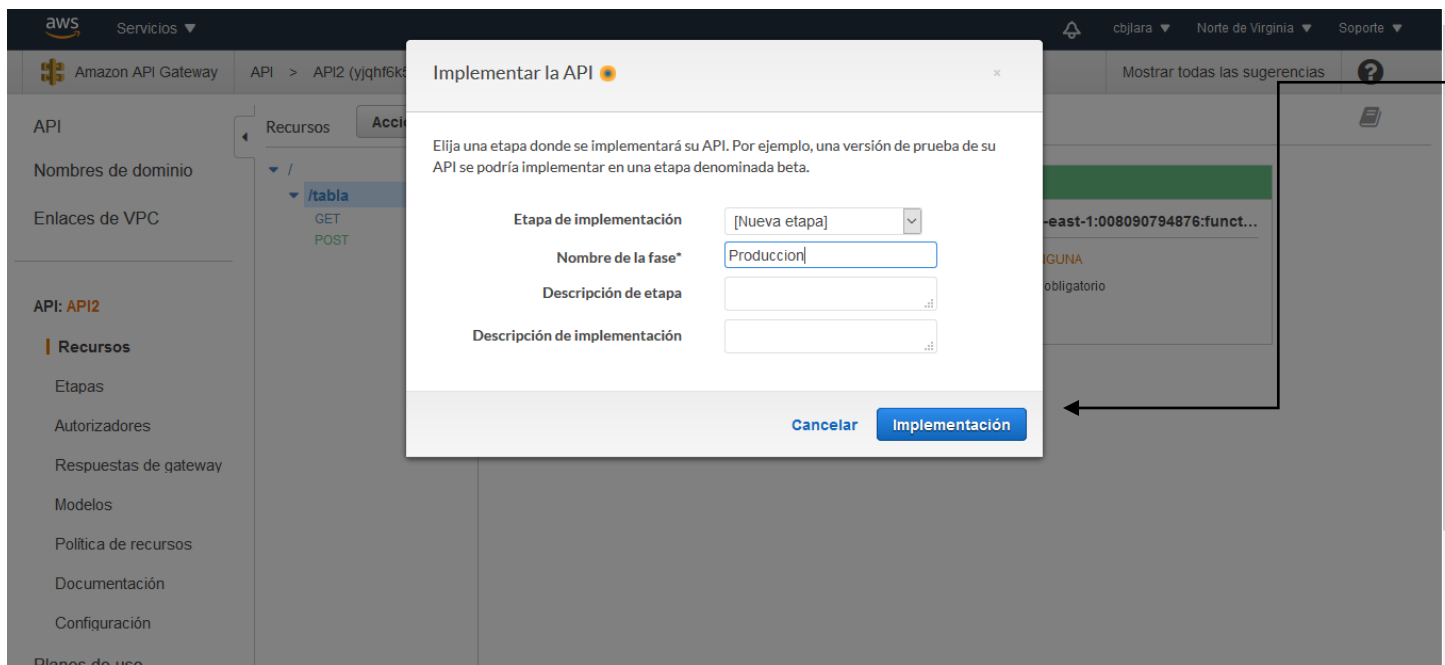
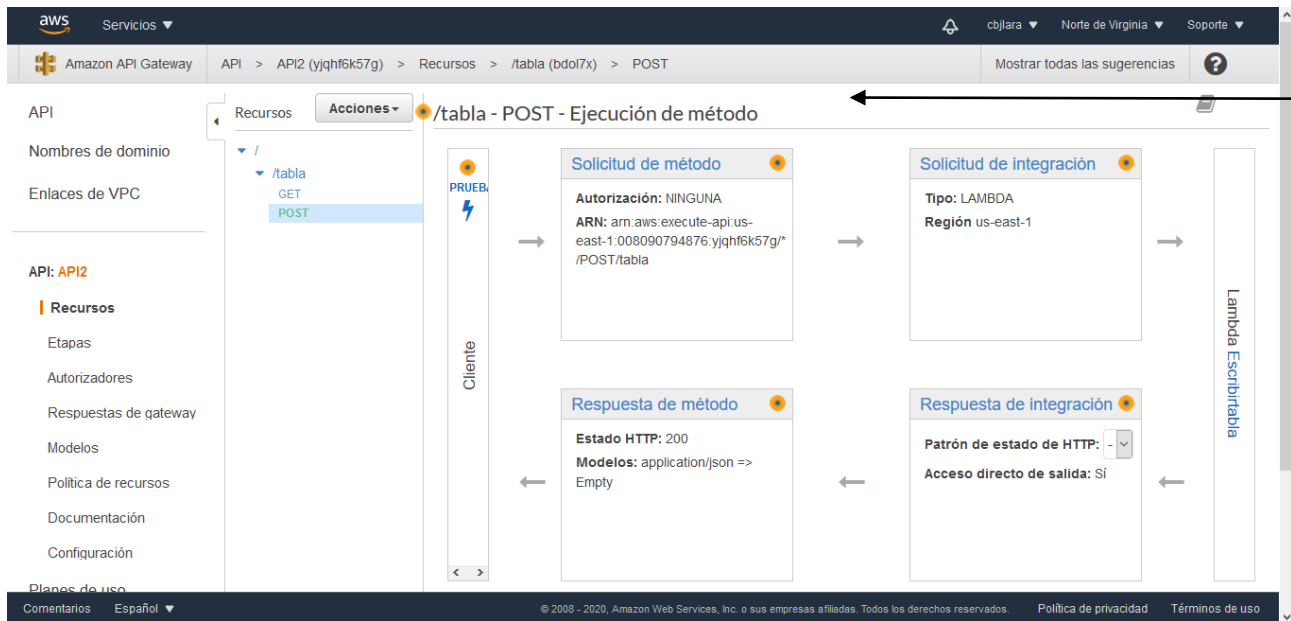
Latencia: 1291 ms

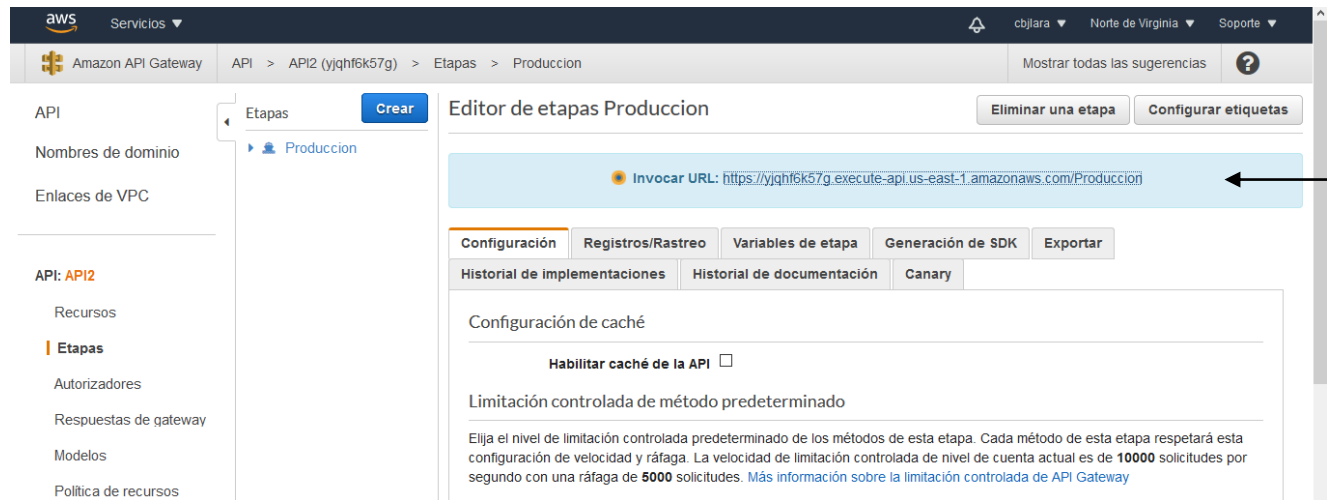
Cuerpo de respuesta

```
{
  "Items": [
    {
      "nombre": "saul",
      "id": "3",
      "apellido": "trejo"
    }
  ],
  "Count": 1,
  "ScannedCount": 1
}
```

Encabezados de respuesta

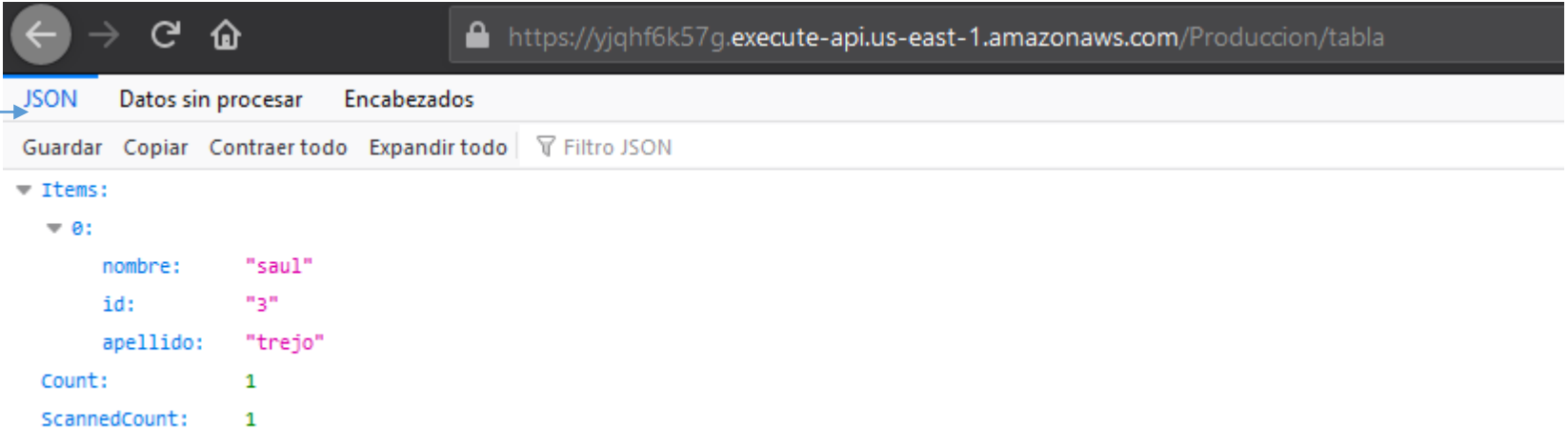
Retornando los registros GET





Alojándose en esta URL nuestra API

En tabla se encuentran alojados nuestros datos, haciendo solicitud de tipo "GET", desde la API



Inicio > Aplicaciones > Advanced REST client



Advanced REST client

Ofrecido por: advancedrestclient.com

★★★★★ 12.023 | [Extensiones](#) | 1.000.000+ usuarios

Con esta App probaremos la función POST

Request

Method POST Request URL <https://yjqhf6k57g.execute-api.us-east-1.amazonaws.com/Prod>

SEND

Parameters ^

Headers

Body

Variables

Body content type

Editor view

Raw input



Raw payload

Data form

Files

```
{
  "id": "8",
  "nombre": "Luis",
  "apellido": "Perez"
}
```

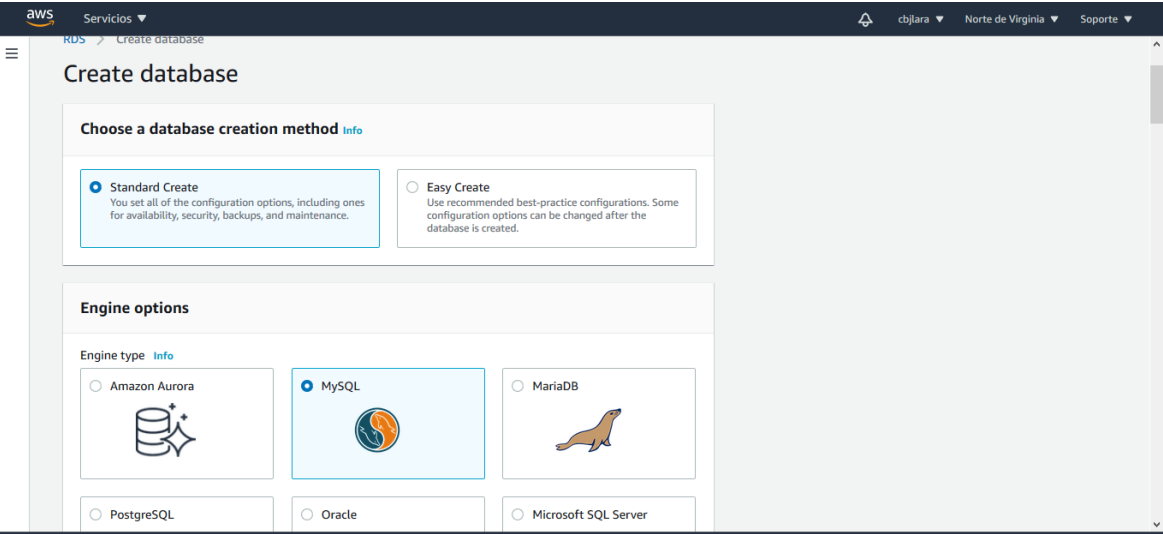
Datos que queremos registrar

200 OK 2161.00 ms

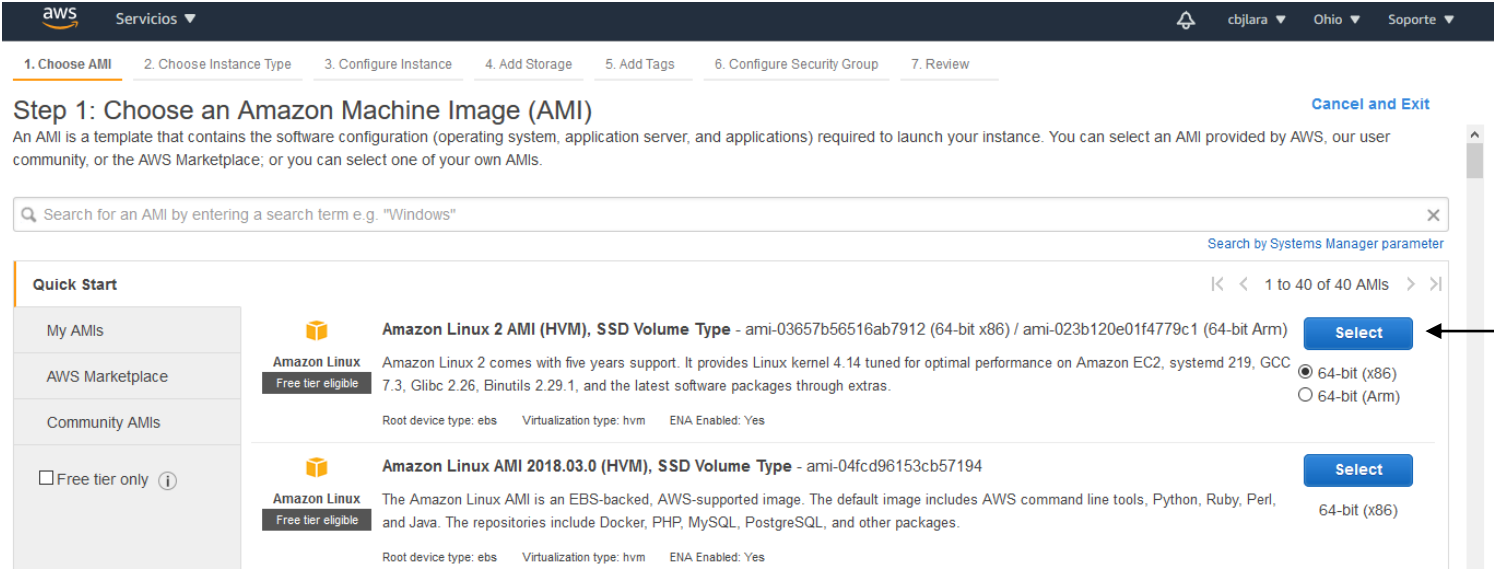
DETAIL

Generando registro satisfactoriamente

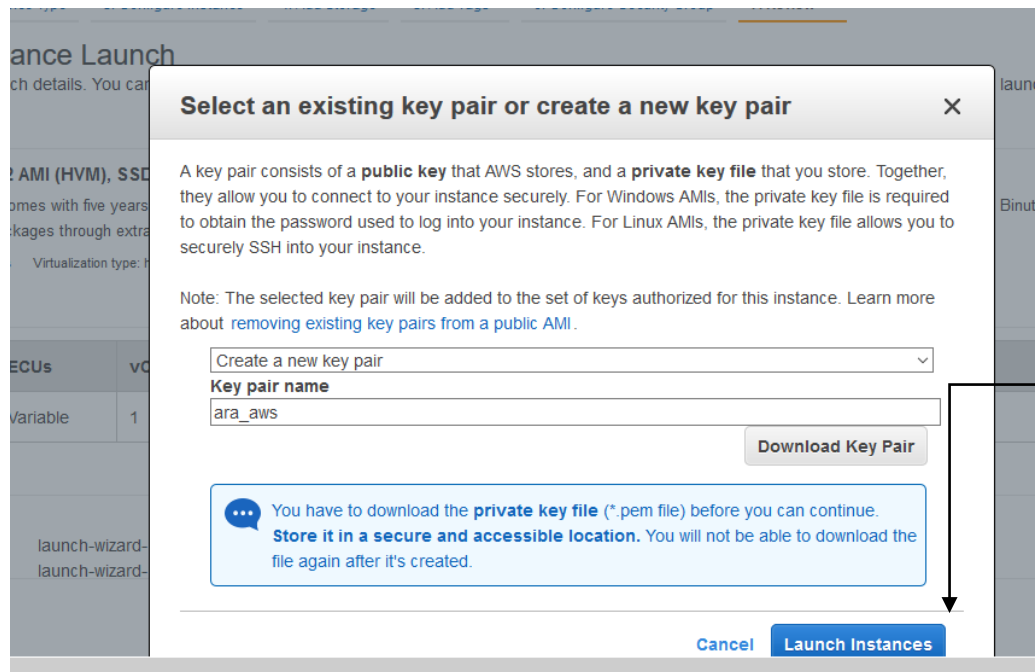
Despliegue de App Nodejs con EC2



Creamos DB en rds, otorgando permisos publicos



Posteriormente creamos instancia EC2



Instancia creada

```
[ec2-user@ip-172-31-67-169 app]$ yum install wget
Failed to set locale, defaulting to C
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
You need to be root to perform this command.
[ec2-user@ip-172-31-67-169 app]$
```

Nos concentraremos a nuestro SSH, creando una carpeta para trabajar con nuestra aplicación

Descomprimiendo en ZIP

```
[ec2-user@ip-172-31-67-169 app]$ wget https://s3.amazonaws.com/cursos-utn/cloud/app_deploy.zip
--2019-05-13 01:00:52-- https://s3.amazonaws.com/cursos-utn/cloud/app_deploy.zip
Resolving s3.amazonaws.com (s3.amazonaws.com)... 52.216.163.61
Connecting to s3.amazonaws.com (s3.amazonaws.com)|52.216.163.61|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7227 (7.1K) [application/zip]
Saving to: 'app_deploy.zip'

100%[=====] 7.1K 260 MB/s in 0s for https://s3.amazonaws.com/cursos-utn/cloud/app_deploy.zip
2019-05-13 01:00:52 (260 MB/s) - 'app_deploy.zip' saved [7227/7227]

[ec2-user@ip-172-31-67-169 app]$ sudo yum install unzip
```

aws Servicios ▾

cbjlara ▾ Norte de Virginia ▾ Soporte ▾

EC2 > Grupos de seguridad > sg-0847e6fce4c3ad70c - launch-wizard-1 > Editar reglas de entrada

Editar reglas de entrada Información

Las reglas de entrada controlan el tráfico entrante que puede llegar a la instancia.

Reglas de entrada Información

Tipo <small>Información</small>	Protocolo <small>Información</small>	Intervalo de puertos <small>Información</small>	Origen <small>Información</small>	Descripción: opcional <small>Información</small>	
SSH ▾	TCP	22	Person... ▾ <input type="text" value="0.0.0.0/0"/>	<input type="text"/>	Eliminar
HTTP ▾	TCP	80	Person... ▾ <input type="text" value="0.0.0.0/0"/>	<input type="text"/>	Eliminar

Agregar regla

Para que tenga acceso cualquier cliente 0.0.0.0/0

aws Servicios ▾

New EC2 Experience Tell us what you think ✕

Panel de EC2 New

- Eventos New
- Etiquetas
- Límites
- ▼ Instancias
 - Instancias
 - Tipos de instancia
 - Plantillas de lanzamiento
 - Solicitudes de spot
 - Savings Plans
 - Instancias reservadas
 - Hosts dedicados New
 - Instancias programadas

Launch Instance ▾ Connect Actions ▾

Filter by tags and attributes or search by keyword

	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
	aramicro	ami-00000000	micro	us-east-1b	running	2/2 checks ...	None	ec2-3-86-64-18.comput...

- Connect
- Get Windows Password
- Create Template From Instance
- Launch More Like This
- Instance State
 - Start
 - Stop
 - Stop - Hibernate
- Instance Settings
 - Reboot
 - Terminate
- Image
- Networking
- CloudWatch Monitoring

Terminamos instancia por motivos de costo de ejecución