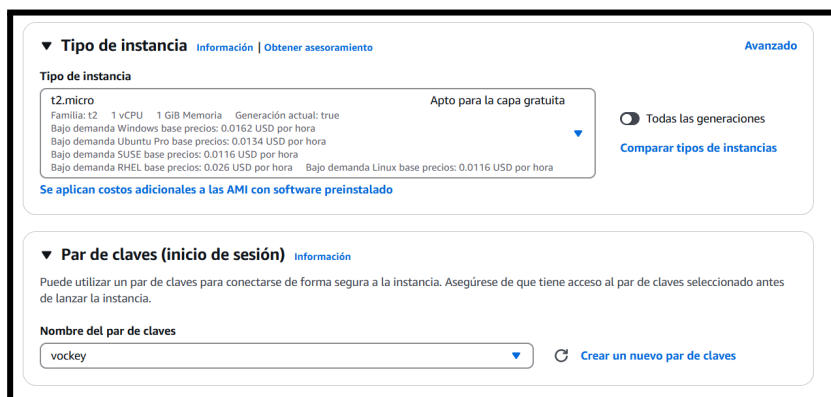
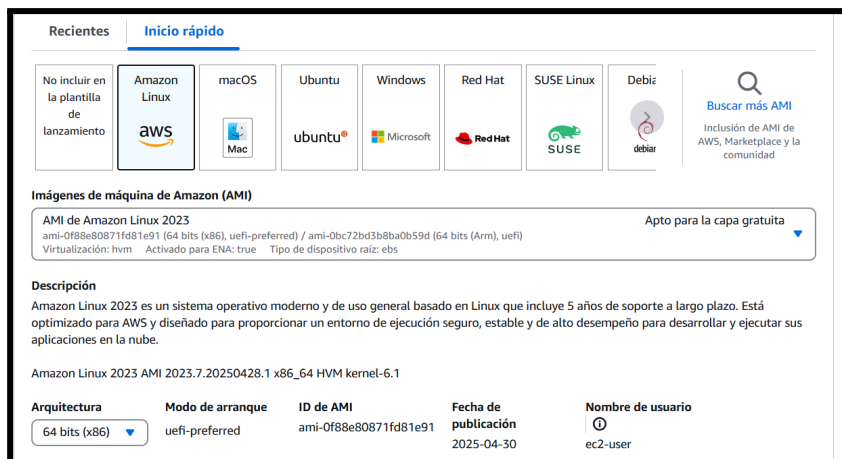
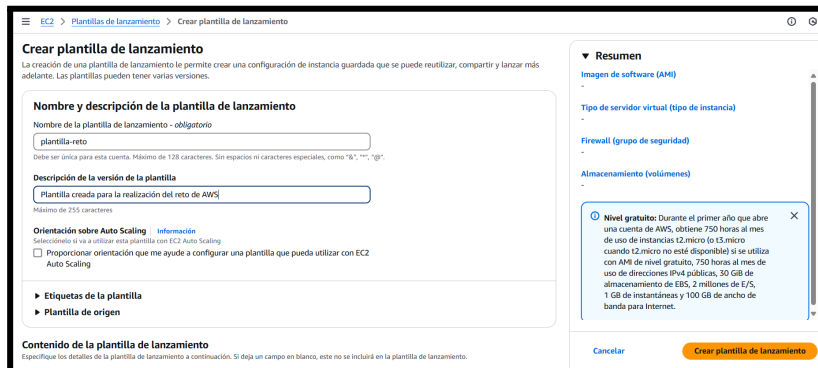
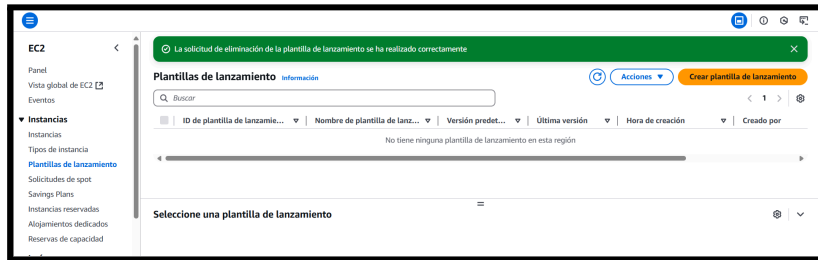


## Reto AWS: Parte 2

### Capturas realizadas para la demostración de la realización del reto:

- Creación del EC2 con load-balancers y autoscaling:
  - Plantilla de lanzamiento:



Configuraciones de red
Información

Subred

Información

No incluir en la plantilla de lanzamiento

[Crear nueva subred](#)

Al especificar una subred, se agrega automáticamente una interfaz de red a la plantilla.

Firewall (grupos de seguridad)

Información

Un grupo de seguridad es un conjunto de reglas de firewall que controlan el tráfico de la instancia. Agregue reglas para permitir que un tráfico específico llegue a la instancia.

☐ Seleccionar un grupo de seguridad existente

☒ Crear grupo de seguridad

Nombre del grupo de seguridad - obligatorio

grupo-seguridad-reto

Este grupo de seguridad se agregará a todas las interfaces de red. El nombre no se puede editar después de crear el grupo de seguridad. La longitud máxima es de 255 caracteres. Caracteres válidos: a-z, A-Z, 0-9, espacios y \_-./!@#%^&\*~:|;'+=,[]{}~

Descripción - obligatorio

Información

Grupo de seguridad asociada a las instancias

VPC

Información

vpc-0d09a848241a95623

(predeterminado)

Reglas de grupos de seguridad de entrada

No hay reglas del grupo de seguridad incluidas actualmente en esta plantilla. Agregue una nueva regla para incluirla en la plantilla de lanzamiento.

(Se me olvidó tomar captura a la parte donde se pone el sudo dnf install)

## - Grupos de autoescalado

Elija la plantilla de lanzamiento o la configuración
Info

Especifique una configuración de lanzamiento que contenga ajustes comunes a todas las instancias EC2 lanzadas por este grupo de Auto Scaling. Si actualmente utiliza configuraciones de lanzamiento, puede considerar la posibilidad de migrar a plantillas de lanzamiento.

Nombre

Nombre del grupo de Auto Scaling

Escriba un nombre para identificar el grupo.

escalado-reto

Debe ser único para esta cuenta en la región actual y no puede superar los 255 caracteres.

Plantilla de lanzamiento

Info

[Cambiar a la configuración de lanzamiento](#)

Plantilla de lanzamiento

Elija una configuración de lanzamiento que contenga la configuración de nivel de instancia, como la imagen de máquina de Amazon (AMI), el tipo de instancia, el par de claves y los grupos de seguridad.

plantilla-reto

[Crear una configuración de lanzamiento](#)

Versión

VPC

Elija la VPC que define la red virtual para el grupo de Auto Scaling.

vpc-0d09a848241a95623

172.31.0.0/16

Default

[Crear una VPC](#)

Zonas de disponibilidad y subredes

Defina qué zonas de disponibilidad y subredes puede utilizar el grupo de Auto Scaling en la VPC elegida.

Seleccionar zonas de disponibilidad y subredes

us-east-1a | subnet-00add74ec1de9d487

172.31.32.0/20

Default

us-east-1b | subnet-03d53a8e2e405efaf

172.31.0.0/20

Default

us-east-1c | subnet-026637f799e2ca8fe

172.31.80.0/20

Default

[Crear una subred](#)

Distribución de zonas de disponibilidad - nueva

El escalamiento automático equilibra automáticamente las instancias en todas las zonas de disponibilidad. Si se produce un problema, el escalamiento automático intentará lanzar instancias en otras zonas de disponibilidad.

☒ Mejor esfuerzo equilibrado

☐ Solo en una zona de disponibilidad

Si los lanzamientos fallan en una zona de disponibilidad, el escalamiento automático intentará lanzarse en otra zona de disponibilidad en buen estado.

Si los lanzamientos fallan en una zona de disponibilidad, el escalamiento automático intentará lanzar instancias en otras zonas de disponibilidad.

Asociar a un nuevo balanceador de carga

Defina un nuevo balanceador de carga para crear para asociarlo a este grupo de Auto Scaling.

Tipo de balanceador de carga

Elija uno de los tipos de balanceador de carga que se ofrecen a continuación. La selección del tipo de balanceador de carga no se puede cambiar después de crearlo. Si necesita un tipo de balanceador de carga distinto del que se ofrece aquí, vaya a la consola de balanceo de carga.

☒ Application Load Balancer

☐ Network Load Balancer

HTTP, HTTPS

TCP, UDP, TLS

Nombre del balanceador de carga

El nombre no se puede cambiar después de crear el balanceador de carga.

reto-alb

Esquema del balanceador de carga

El esquema no se puede cambiar después de crear el balanceador de carga.

☐ Internal

☒ Internet-facing

Asignación de red

El nuevo balanceador de carga se creará con las mismas selecciones de VPC y zona de disponibilidad que el grupo de Auto Scaling. Puede seleccionar subredes diferentes y agregar subredes de zonas de disponibilidad adicionales.

**Agentes de escucha y direccionamiento**

Si necesita agentes de escucha seguros o varios agentes de escucha, puede configurarlos desde la [Consola de balanceo de carga](#) después de crear el balanceador de carga.

**Protocolo** **Puerto** **Direccionamiento predeterminado (reenviar a)**

HTTP 80 Crear un grupo de destino

**Nombre del grupo de destino nuevo**

Se creará un grupo de destino de instancia con la configuración predeterminada.

reto-agentes-escucha

**Comprobaciones de estado**

Las comprobaciones de estado aumentan la disponibilidad reemplazando instancias en mal estado. Cuando se utilizan varias comprobaciones de estado, se evalúan todas y, si se produce un error en al menos una, se lleva a cabo la sustitución de instancias.

**Comprobaciones de estado de EC2**

[Siempre habilitadas](#)

**Tipos de comprobaciones de estado adicionales - opcional** [Info](#)

☒ **Activar las comprobaciones de estado de Elastic Load Balancing** [Recomendado](#)

Elastic Load Balancing puede monitorear si las instancias están disponibles para gestionar solicitudes. Cuando informa de una instancia en mal estado, EC2 Auto Scaling puede sustituirla en la siguiente comprobación periódica.

[EC2 Auto Scaling comenzará a detectar y actuar en función de las comprobaciones de estado realizadas por Elastic Load Balancing. Para evitar terminaciones inesperadas, primero verifique la configuración de estas comprobaciones de estado en la \[consola del equilibrador de carga\]\(#\)](#)

**Tamaño del grupo** [Info](#)

Defina el tamaño inicial del grupo de escalamiento automático. Después de crear el grupo, mediante el escalamiento automático.

**Tipo de capacidad deseado**

Elija la unidad de medida para el valor de capacidad deseado. Las vCPU y la memoria (GiB) solo son compatibles con el tipo de capacidad deseado.

Unidades (número de instancias)

**Capacidad deseada**

Especifique el tamaño de su grupo.

2

**Escalado** [Info](#)

Puede cambiar el tamaño de su grupo de escalamiento automático de forma manual o automática.

**Límites de escalamiento**

Establezca límites sobre cuánto puede aumentarse o disminuirse la capacidad deseada.

**Capacidad deseada mínima** **Capacidad deseada máxima**

1 5

Capacidad igual o inferior a la deseada. Capacidad igual o superior a la deseada.

- Configurar puerto 80 HTTP

**Agentes de escucha y reglas (1)** [Info](#) [Administrar reglas](#)

Un agente de escucha comprueba las solicitudes de conexión en su protocolo y puerto configurados. El tráfico reenvía al grupo de destino predeterminado y cualquier regla adicional.

<input type="checkbox"/>	Protocol:Port	Acción predeterminada	Reglas
<input type="checkbox"/>	HTTP:80 <a href="#">No accesible</a>	<b>Reenviar al grupo de destino</b> <ul style="list-style-type: none"> <li><a href="#">reto-agentes-escucha</a> 1 (100%)</li> <li>Permanencia del grupo de destino: Desactivada</li> </ul>	<a href="#">1 regla</a>

**Reglas de entrada** [Administrar etiquetas](#) [Editar reglas de entrada](#)

<input type="checkbox"/>	Name	ID de la regla del gr...	Versión de IP	Tipo	Protocolo	Intervalo de puertos
No se encontraron reglas de grupo de seguridad						

**Editar reglas de entrada** Información

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

Reglas de entrada	Información	Tipo	Protocolo	Intervalo de puertos	Origen	Descripción: opcional
ID de la regla del grupo de seguridad	Información		Información	Información	Información	
-	TCP personalizado	TCP	80	Anyw...	0.0.0.0/0	

[Agregar regla](#)

⚠ Las reglas cuyo origen es 0.0.0.0/0 o ::/0 permiten a todas las direcciones IP acceder a la instancia. Recomendamos configurar reglas de grupo de seguridad para permitir el acceso únicamente desde direcciones IP conocidas.

[Cancelar](#) [Previsualizar los cambios](#) [Guardar reglas](#)

**Reglas de entrada (1)** [Administrar etiquetas](#) [Editar reglas de entrada](#)

<input type="checkbox"/>	Name	ID de la regla del gr...	Versión de IP	Tipo	Protocolo	Intervalo de puertos
<input type="checkbox"/>	-	sgr-03cdd20bb5aacf471	IPv4	HTTP	TCP	80

(Añadiremos una regla para acceder por SSH para poner en marcha nuestra API-REST)

- Resultado final:



- Creación de BBDD noSQL con DynamoDB:
  - Creación de la tabla “DiasSemana”:

**Crear tabla**

**Detalles de la tabla** Información

DynamoDB es una base de datos sin esquemas que solo requiere un nombre de tabla y una clave principal al crear la tabla.

**Nombre de la tabla**  
Se utilizará para identificar su tabla.  
  
Entre 3 y 255 caracteres. Solo se pueden usar letras, números, guiones bajos (\_) y puntos (.).

**Clave de partición**  
La clave de partición forma parte de la clave principal de la tabla. Se trata de un valor hash que se utiliza para recuperar elementos de la tabla, así como para asignar datos entre hosts por cuestiones de escalabilidad y disponibilidad.

**Clave de ordenación - opcional**  
Puede utilizar una clave de ordenación como segunda parte de la clave principal de una tabla. La clave de ordenación le permite ordenar o buscar entre todos los elementos que comparten la misma clave de partición.

**Configuración de la tabla**

☒ Configuración predeterminada  
La forma más rápida de crear su tabla. Puede modificar la mayor parte de la configuración después de crear la tabla. Para modificar esta configuración ahora, seleccione "Personalizar la configuración".

☐ Personalizar configuración  
Utilice estas características avanzadas para que DynamoDB funcione mejor de acuerdo a sus necesidades.

- Crear los días de la semana en DynamoDB:

**Crear elemento** [Formulario](#) [Vista JSON](#)

Puede agregar, eliminar o editar los atributos de un elemento. Es posible anidar atributos dentro de otros atributos hasta 32 niveles de profundidad. [Más información](#)

**Atributos** [Agregar nuevo atributo](#)

<input type="checkbox"/> Nombre de atributo	Valor	Tipo
Id - Clave de partición	<input type="text" value="0"/>	Número
Días	<input type="text" value="Lunes"/>	Cadena <a href="#">Eliminar</a>

[Cancelar](#) [Crear elemento](#)

(Repetimos esto para los demás días).

**Tabla: DiasSemana: elementos devueltos (7)** Acciones Crear elemento

El análisis inició el mayo 06, 2025, 17:54:30

<input type="checkbox"/>	Id (Número)	Día
<input type="checkbox"/>	<a href="#">3</a>	Jueves
<input type="checkbox"/>	<a href="#">2</a>	Miércoles
<input type="checkbox"/>	<a href="#">4</a>	Viernes
<input type="checkbox"/>	<a href="#">6</a>	Domingo
<input type="checkbox"/>	<a href="#">1</a>	Martes
<input type="checkbox"/>	<a href="#">0</a>	Lunes
<input type="checkbox"/>	<a href="#">5</a>	Sábado

- Entramos a las instancias a copiar el código subido a git:

- Cambiamos los roles IAM de las instancias:

Asocia un rol de IAM a la instancia.

ID de la instancia  
i-0f5e1633b86de242

Rol de IAM  
Selecciona el rol de IAM para asociarlo a la instancia o crea uno nuevo si no ha creado ninguno. El rol que selecciones sustituirá todos los roles que estén asociados actualmente a la instancia.

Crear un nuevo rol de IAM

Cancelar Actualizar rol de IAM

- Nos conectamos a las instancias e instalamos dependencias:

```
[ec2-user@ip-172-31-35-146 ~]$ sudo dnf install git
Last metadata expiration check: 0:41:48 ago on Tue May 6 15:32:26 2025.
Dependencies resolved.

=====
Package                                Architecture          Version
=====
```

```
Complete!
[ec2-user@ip-172-31-35-146 ~]$ sudo dnf install python3 -y
Last metadata expiration check: 0:42:45 ago on Tue May 6 15:32:26 2025.
Package python3-3.9.21-1.amzn2023.0.3.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
```

- Clonamos el repo:

```
[ec2-user@ip-172-31-35-146 ~]$ git clone 'https://github.com/ManuMedinal/reto-aws.git'
Cloning into 'reto-aws'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 10 (delta 0), reused 10 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (10/10), done.
[ec2-user@ip-172-31-35-146 ~]$ ls
reto-aws
[ec2-user@ip-172-31-35-146 ~]$
```

- Instalamos dependencias de python:

```
[ec2-user@ip-172-31-35-146 reto-aws]$ sudo dnf install python3-pip -y
Last metadata expiration check: 0:47:05 ago on Tue May 6 15:32:26 2025.
Dependencies resolved.

=====
Package                                Architecture          Version          Repository
=====
```

```
Complete!
[ec2-user@ip-172-31-35-146 reto-aws]$ pip3 install flask
Defaulting to user installation because normal site-packages is not writeable
Collecting flask
  Downloading flask-3.1.0-py3-none-any.whl (102 kB)
  | 102 kB 15.2 MB/s
Collecting blinker>=1.9
  Downloading blinker-1.9.0-py3-none-any.whl (8.5 kB)
```

```
[ec2-user@ip-172-31-35-146 reto-aws]$ pip3 install boto3
Defaulting to user installation because normal site-packages is not writeable
Collecting boto3
  Downloading boto3-1.38.9-py3-none-any.whl (139 kB)
    |#####| 139 kB 17.1 MB/s
Collecting s3transfer<0.13.0,>=0.12.0
  Downloading s3transfer-0.12.0-py3-none-any.whl (84 kB)
    |#####| 84 kB 6.7 MB/s
```

- Lo ejecutamos:

```
[ec2-user@ip-172-31-35-146 reto-aws]$ python3 api.py
* Serving Flask app 'api'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.31.35.146:5000
Press CTRL+C to quit
```

- Probamos con un curl:

```
[ec2-user@ip-172-31-32-129 reto-aws]$ curl http://127.0.0.1:5000/dias
127.0.0.1 - - [06/May/2025 23:24:50] "GET /dias HTTP/1.1" 200 -
{"dias":[{"dia":"Jueves","id":"3"}, {"dia":"Miercoles","id":"2"}, {"dia":"Viernes","id":"4"}, {"dia":"Domingo","id":"6"}, {"dia":"Martes","id":"1"}, {"dia":"Lunes","id":"0"}, {"dia":"Sabado","id":"5"}]}
```

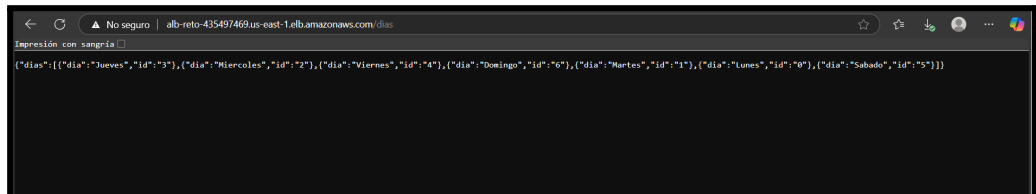
(Se devuelven los días de la semana)

- Desde una máquina local (la mía por ejemplo):

```
Manuel@Manu MINGW64 ~/Desktop/Juegos
$ curl http://alb-reto-435497469.us-east-1.elb.amazonaws.com/dias
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 197 100 197 0 0 793 0 --:--:-- --:--:-- --:--:-- 800{"
dias":[{"dia":"Jueves","id":"3"}, {"dia":"Miercoles","id":"2"}, {"dia":"Viernes","
id":"4"}, {"dia":"Domingo","id":"6"}, {"dia":"Martes","id":"1"}, {"dia":"Lunes","id
":"0"}, {"dia":"Sabado","id":"5"}]}
```


(Se devuelven los días de la semana)

Y en navegador:



The browser window shows the address bar with the URL `alb-reto-435497469.us-east-1.elb.amazonaws.com/dias`. The page content displays the JSON response: `{"dias":[{"dia":"Jueves","id":"3"}, {"dia":"Miercoles","id":"2"}, {"dia":"Viernes","id":"4"}, {"dia":"Domingo","id":"6"}, {"dia":"Martes","id":"1"}, {"dia":"Lunes","id":"0"}, {"dia":"Sabado","id":"5"}]}`.

- Creamos la Api Gateway:
  - Creamos la gateway (se me olvidó sacar captura).
  - Creamos la ruta



The screenshot shows the 'Rutas' (Routes) section in the AWS API Gateway console. It displays a table with one route defined: `/dias` with the HTTP method `GET`. The table has columns for 'Rutas para api-rest-reto', 'Buscar', and 'GET'.

- Definimos la integración de la ruta:

**Detalles de integración**

**Destino de integración**  
 Cuando esta ruta reciba una solicitud, API Gateway la enviará a la URL especificada con el método HTTP definido. "ANY" indica que API Gateway utilizará el mismo método que recibe del intermediario para llamar a la integración.

**Método HTTP** URL

GET http://alb-reto-435497469.us-east-1.elb.amazonaws.com/dias

► Configuración avanzada

Descripción: *opcional*

Cancelar Guardar

- Vamos a stages y hacemos click en el enlace que nos da:

API: api-rest-reto...s411cdvca

**Etapas**

Etapas para api-rest-reto Crear

Buscar

● Sdefault

**Detalles de la etapa** Eliminar Editar

**Detalles**

Nombre	Sdefault	Creado	May 7, 2025 12:50 AM	Última actualización	May 7, 2025 1:43 AM
URL de invocación	https://s411cdvca.execute-api-us-east-1.amazonaws.com				
Descripción	Ninguna				

**Implementación asociada**

Implementación automática ✓ Habilitado

ID de implementación	Implementación creada
u1p2za	May 7, 2025 1:43 AM

**Descripción de la implementación**  
 Automatic deployment triggered by changes to the Api configuration

**Variables de la etapa**

```
Impresión con sangría
{"message": "Not Found"}
```

Nos dará error, debemos de añadirle a la url manualmente el "/dias":

```
https://s411cdvca.execute-api-us-east-1.amazonaws.com/dias

Impresión con sangría
{"dias":[{"dia":"Jueves","id":"3"}, {"dia":"Miercoles","id":"2"}, {"dia":"Viernes","id":"4"}, {"dia":"Domingo","id":"6"}, {"dia":"Martes","id":"1"}, {"dia":"Lunes","id":"0"}, {"dia":"Sabado","id":"5"}]}
```

¡Listo!

- Hubo un pequeño error, y es que hay que definir los puertos en las reglas (y básicamente en todo) al puerto 5000, según lo que definimos en el proyecto python. O podemos cambiar el puerto al 80 en el proyecto y así ahorrarnos trabajo:
  - Para los agentes de escucha (quienes redirigen):

**Revisar destinos**

Destinos (2) Eliminar todos los pendientes

☐ Mostrar solo pendientes

ID de instancia	Nombre	Puerto	Estado	Grupos de seguridad	Zona	Dirección IPv4 privada	ID de subred	Hora de lanzamiento
i-0b338a6e237c24fe7		5000	Ejecutando	grupo-reto	us-east-1a	172.31.32.129	subnet-02b8dfb8f81c0d540	7 de mayo de 2025, 01:06
i-0b338a6e237c24fe7		80	Ejecutando	grupo-reto	us-east-1a	172.31.32.129	subnet-02b8dfb8f81c0d540	7 de mayo de 2025, 01:06

- Para los grupos de seguridad:

**Reglas de entrada (3)** Administrar etiquetas Editar reglas de entrada

<input type="checkbox"/>	Name	ID de la regla del gr...	Versión de IP	Tipo	Protocolo	Intervalo de puertos
<input type="checkbox"/>	-	sgr-0cb27de4104957c5e	IPv4	TCP personalizado	TCP	5000
<input type="checkbox"/>	-	sgr-0265a59dfa56d792a	IPv4	HTTP	TCP	80
<input type="checkbox"/>	-	sgr-08622fd8d9941cde2	IPv4	SSH	TCP	22

Ya por último, a veces da error 503 por sobrecarga de peticiones HTTP. Solo aumentaremos el número de instancias en los grupos de autoescalado.

### **Recursos:**

Link github del proyecto python para simular la API REST:

<https://github.com/ManuMedina1/reto-aws.git>