



# NAT Gateway - subredes privadas

## Reglas encadenadas y NACL



GOBIERNO  
DE ESPAÑA

MINISTERIO  
DE EDUCACIÓN  
Y FORMACIÓN PROFESIONAL



UNIÓN EUROPEA  
Fondo Social Europeo  
El FSE invierte en tu futuro



GENERALITAT  
VALENCIANA  
Conselleria d'Educació, Cultura,  
Universitats i Ocupació



CEFIRE  
FORMACIÓ PROFESSIONAL  
ENSEIGNEMENTS ARTISITIQUES  
I ESPORTIVES



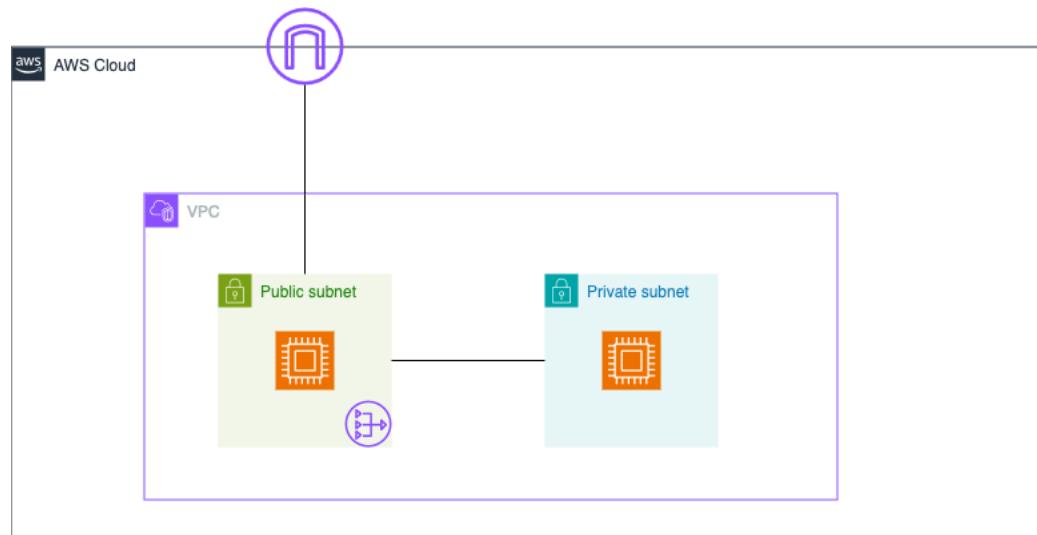
Formació Professional  
Comunitat Valenciana

Escenario .....	3
Conexión SSH.....	7
NACL (network Access control list) .....	11
Escenario.....	11
VPC .....	11
Reglas encadenadas .....	16
Tareas .....	18

## Escenario

El escenario propuesto que vamos a desarrollar es el siguiente, una subred pública y una subred privada, la cual sólo tendrá acceso al exterior pasando primero por la subred pública:

- Direcciónamiento VPC: 172.16.0.0/16
- Direcciónamiento subred pública: 172.16.0.0/20
- Direcciónamiento subred privada: 172.16.64.0/20



[https://docs.aws.amazon.com/es\\_es/vpc/latest/userguide/vpc-nat.html](https://docs.aws.amazon.com/es_es/vpc/latest/userguide/vpc-nat.html)

Primero se montará el escenario utilizando el panel de control y se revisarán las opciones y las tablas de enrutamiento que facilitan esta situación, si queréis ir más rápido o sois fans de la consola también podéis enlazar con el tema anterior y reutilizar el script para montar casi todo, sólo faltará el NAT Gateway y la subred privada.

En nuestro caso podríamos crear una VPC y desde allí ya podríamos crear una subred privada, pero este NO es el caso de los pasos que vamos a seguir.

Aunque en nuestro caso, voy a reutilizar el script y a partir de ahí añadir la subred privada y los elementos que nos hagan falta. Es decir, añadir una subred pública, con un internet gateway y la tabla de enrutamiento para salir al exterior, para montar la VPC y la subred pública podéis usar la interfaz web de la consola de administración. ASUMO a partir de aquí que lo hacéis con la interfaz gráfica.

Resumiendo:

Tenemos una VPC, una subred pública, un IGW y una tabla de enrutamiento.

A continuación, se muestra el direcciónamiento de la VPC:

**Name tag auto-generation** [Info](#)  
 Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

Auto-generate  
 proyecto

**IPv4 CIDR block** [Info](#)  
 Determine the starting IP and the size of your VPC using CIDR notation.

172.16.0.0/16 65.536 IPs

CIDR block size must be between /16 and /28.

**IPv6 CIDR block** [Info](#)  
 No IPv6 CIDR block  
 Amazon-provided IPv6 CIDR block

**Tenancy** [Info](#)  
 Default

Y ahora el direccionamiento de la subred pública (la subred privada la crearemos más adelante)

**Number of public subnets** [Info](#)  
 The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.

0 1

**Number of private subnets** [Info](#)  
 The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.

0 1 2

**Customize subnets CIDR blocks**

**Public subnet CIDR block in us-east-1a**  
 172.16.0.0/20 4096 IPs

**VPC dashboard**

Subnets (7) <a href="#">Info</a>					
Find resources by attribute or tag <input type="text"/>					
	Name	Subnet ID	State	VPC	IPv4
<input type="checkbox"/>	-	subnet-02a189cca22c0a14c	<input checked="" type="checkbox"/> Available	vpc-04ebd867e92ad902e	172
<input type="checkbox"/>	-	subnet-082affb54f4b075a7	<input checked="" type="checkbox"/> Available	vpc-04ebd867e92ad902e	172
<input type="checkbox"/>	projeto-subnet-public1-us-east-1a	subnet-0c14d6cf7f208ef36	<input checked="" type="checkbox"/> Available	vpc-07a5ab0ae36499f26   projeto-vpc	172
<input type="checkbox"/>	-	subnet-0bea8f0be75fda9da	<input checked="" type="checkbox"/> Available	vpc-04ebd867e92ad902e	172

Ahora partiendo de aquí lo primero es crear una subred (que será la privada). Desde el menú de Subnets podremos crearla:

**VPC dashboard**

Subnets (7) <a href="#">Info</a>							
Find resources by attribute or tag <input type="text"/>							
	Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	IPv6 CIDR association
<input type="checkbox"/>	-	subnet-02a189cca22c0a14c	<input checked="" type="checkbox"/> Available	vpc-04ebd867e92ad902e	172.31.48.0/20	-	-
<input type="checkbox"/>	-	subnet-082affb54f4b075a7	<input checked="" type="checkbox"/> Available	vpc-04ebd867e92ad902e	172.31.80.0/20	-	-
<input type="checkbox"/>	projeto-subnet-public1-us-east-1a	subnet-0c14d6cf7f208ef36	<input checked="" type="checkbox"/> Available	vpc-07a5ab0ae36499f26   projeto-vpc	172.16.0.0/20	-	-

Al crearla con las siguientes opciones:

VPC > Subnets > Create subnet

**VPC**

**VPC ID**  
Create subnets in this VPC.  
vpc-07a5ab0ae36499f26 (proyecto-vpc)

**Associated VPC CIDRs**

**IPv4 CIDRs**  
172.16.0.0/16

**Subnet settings**  
Specify the CIDR blocks and Availability Zone for the subnet.

**Subnet 1 of 1**

**Subnet name**  
Create a tag with a key of 'Name' and a value that you specify.  
subredprivada

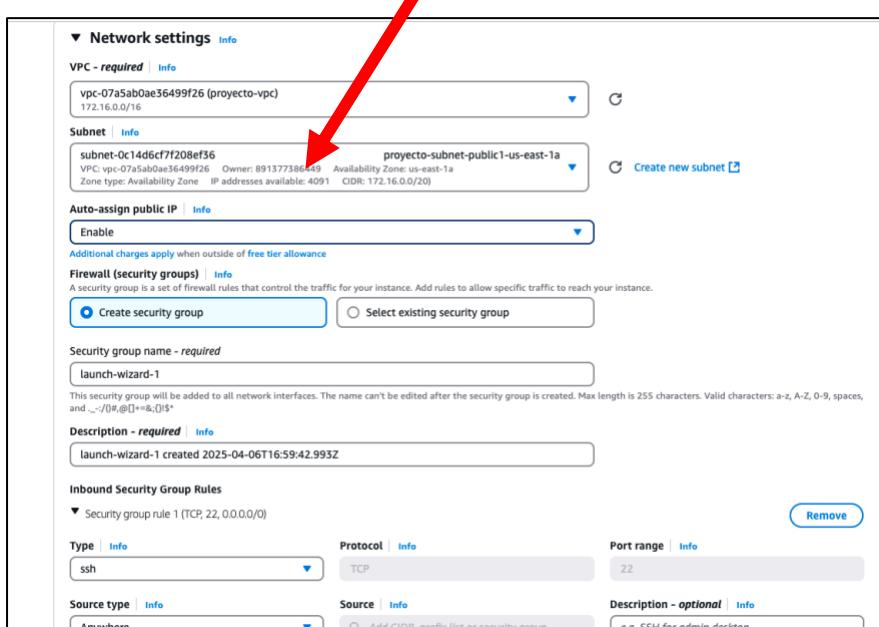
The name can be up to 256 characters long.

**Availability Zone** Info  
Choose the zone in which your subnet will reside, or let Amazon choose one for you.  
No preference

**IPv4 VPC CIDR block** Info  
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.  
172.16.0.0/16

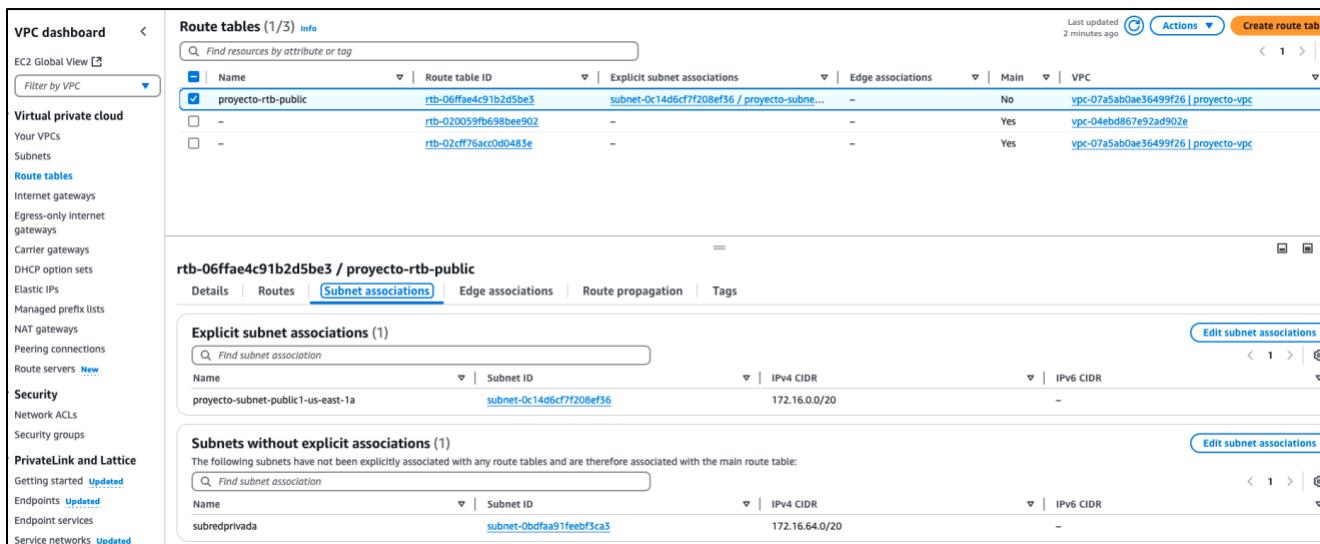
**IPv4 subnet CIDR block**  
172.16.64.0/20 4096 IPs

Y una ec2 para la subred pública y otra para la subred privada. Se muestra la configuración para el ec2 público, para el privado habría que cambiar sólo la subred



Miramos la ip pública del ec2 creado en la subred privada e intentamos hacer un ssh, la respuesta es que NO se puede acceder. NO está asociada la subred en el internet gateway para poder enrutarlo hacia fuera.

Si comprobamos la tabla de enrutamiento desde el menú VPC -> Route tables podremos ver que tenemos una tabla asociada a la VPC en la cual solo hay asociada una subred (siendo la pública solamente).



Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC
proyecto-rtb-public	<a href="#">rtb-06ffae4c91b2d5be3</a>	<a href="#">subnet-0c14d6cf7f208ef36 / proyecto-subne...</a>	-	No	<a href="#">vpc-07a5ab0ae36499f26   proyecto-vpc</a>
-	<a href="#">rtb-020059fb698bee902</a>	-	-	Yes	<a href="#">vpc-04ebd867e92ad902e</a>
-	<a href="#">rtb-02cff76acc0d0483e</a>	-	-	Yes	<a href="#">vpc-07a5ab0ae36499f26   proyecto-vpc</a>

**rtb-06ffae4c91b2d5be3 / proyecto-rtb-public**

Details | Routes | **Subnet associations** | Edge associations | Route propagation | Tags

**Explicit subnet associations (1)**

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
proyecto-subnet-public1-us-east-1a	<a href="#">subnet-0c14d6cf7f208ef36</a>	172.16.0.0/20	-

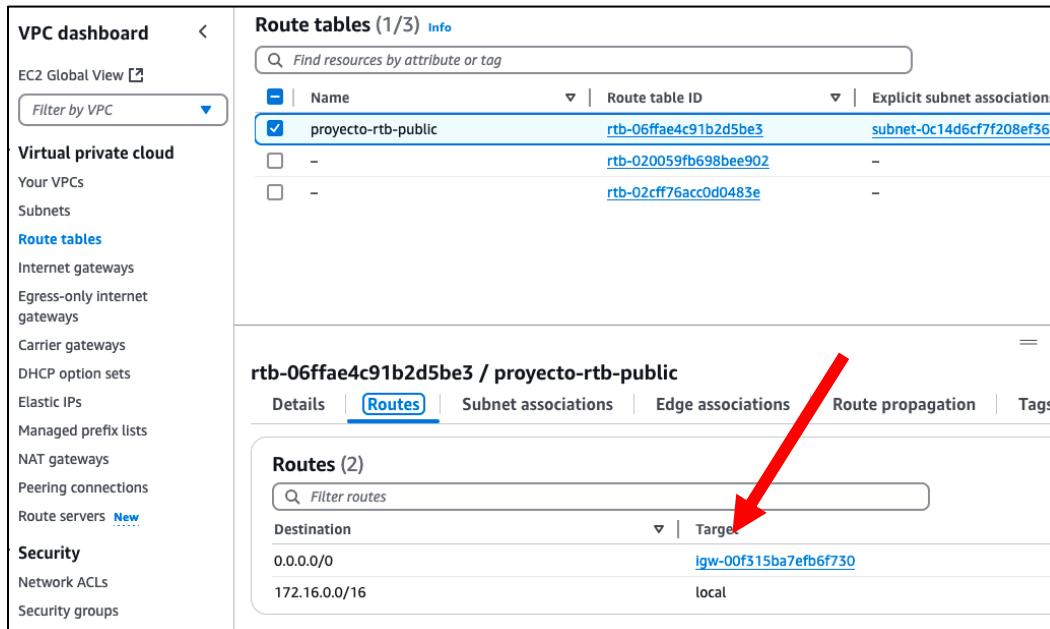
**Subnets without explicit associations (1)**

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
subredprivada	<a href="#">subnet-0bdfaa91feebf3ca3</a>	172.16.64.0/20	-

La subred privada nos aparece en la pestaña “Subnets without explicit associations”, lo que quiere decir que está preparada para ser añadida a la tabla de enrutamiento, pero aún no lo hemos hecho.

Realmente lo que pretendemos es hacer que la subred privada salga a través del a subred pública, para ello cada subred tiene su propia tabla de enrutamiento, la subred pública tiene una ruta hacia el internet gateway para poder salir.



Name	Route table ID	Explicit subnet associations
proyecto-rtb-public	<a href="#">rtb-06ffae4c91b2d5be3</a>	<a href="#">subnet-0c14d6cf7f208ef36 /</a>
-	<a href="#">rtb-020059fb698bee902</a>	-
-	<a href="#">rtb-02cff76acc0d0483e</a>	-

**rtb-06ffae4c91b2d5be3 / proyecto-rtb-public**

Details | **Routes** | Subnet associations | Edge associations | Route propagation | Tags

**Routes (2)**

Destination	Target
0.0.0.0/0	<a href="#">igw-00f315ba7efb6f730</a>
172.16.0.0/16	local

En este punto podríamos conectarnos por ssh a la instancia de la red pública y hacer ping a [www.google.es](http://www.google.es) y funcionaría y desde esta instancia conectarnos por ssh a la instancia de la subred privada, lo que pasa es que aún falta conocer un par de comandos que veremos más adelante para poder conectarnos, pero si alguno se atreve y lo logra el ping desde la subred privada NO funciona.

```
lec2-user@ip-172-16-2-45 ~]$ ping www.google.es
PING www.google.es (142.251.179.94) 56(84) bytes of data.
64 bytes from pd-in-f94.1e100.net (142.251.179.94): icmp_seq=1 ttl=56 time=1.68 ms
64 bytes from pd-in-f94.1e100.net (142.251.179.94): icmp_seq=2 ttl=56 time=1.71 ms
^C
--- www.google.es ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 1.677/1.692/1.708/0.015 ms
[ec2-user@ip-172-16-2-45 ~]$ ssh 172.16.68.71
      #
~\_ #####
~~ \#####\ Amazon Linux 2023
~~  \###|
~~   \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
~~    V~' '-->
~~     /
~~-. /-
~/ /-
/_m/' Last login: Sun Apr  6 17:18:34 2025 from 172.16.2.45
[ec2-user@ip-172-16-68-71 ~]$ ping www.google.es
PING www.google.es (142.250.31.94) 56(84) bytes of data.
^C
--- www.google.es ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3110ms
```

Ahora vamos a crear una tabla de enrutamiento para la subred privada (aunque podríamos usar la tabla que ya hay creada).

Llegados a este punto, vamos a explicar cómo poder hacer estas pruebas y arrastrar las credenciales (labuser.pem) ya que vamos a usar las mismas para las dos instancias, aunque también se podría hacer usando diferentes credenciales.

## Conexión SSH

Seguiremos los pasos indicados para equipos Linux. En primer lugar, comprobamos que las claves privadas tienen los permisos adecuados. Desde el terminal ejecutaremos los siguientes comandos:

- Ejecutamos ssh-agent en 2º plano:

```
eval $(ssh-agent)
```

- Cargamos en memoria la clave privada de la instancia :

```
ssh-add labuser.pem
```

- Podemos ver las claves privadas mediante el comando:

```
ssh-add -l
```

- Nos conectamos en primer lugar a la instancia de la subred pública. Como se puede comprobar, al realizar la conexión también tenemos disponible en memoria, la clave privada del servidor app.

```
ssh -i labuser.pem -A ec2-user@noombrednsdelainstancia
```

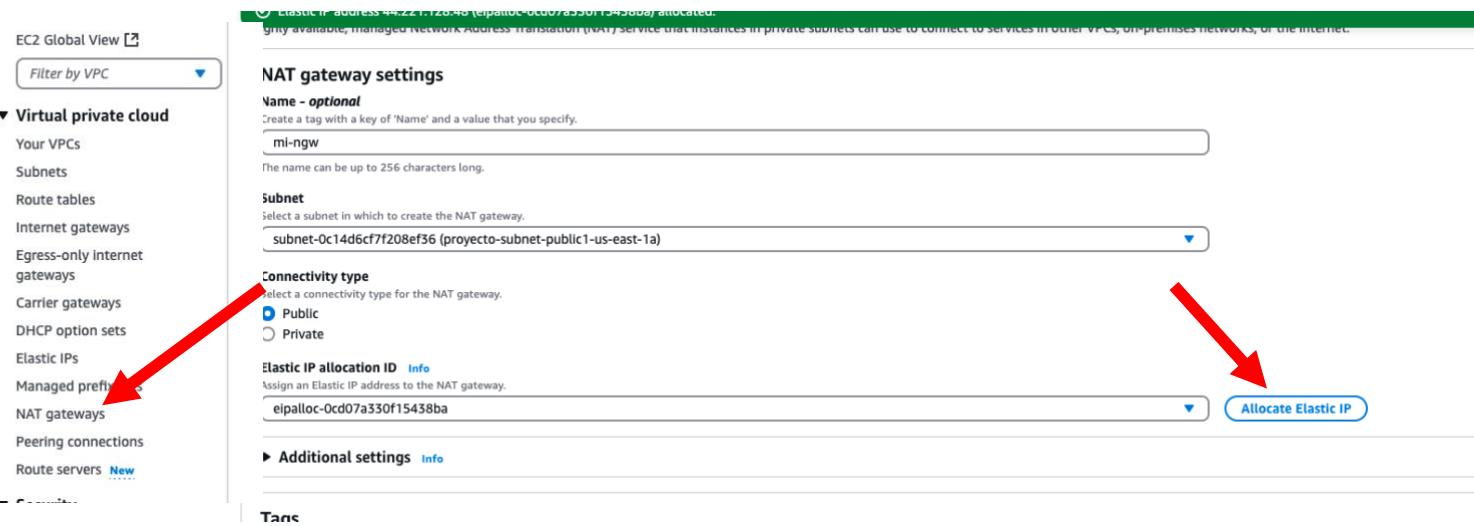
- Seguidamente nos conectamos a la instancia de la subred privada

```
ssh direccionipprivada
```

- Cuando realices la conexión ssh deberás usar el parámetro -A que es el que permite el reenvío de clave privada.

Una vez ya hemos logrado tener acceso a la instancia de la subred privada si intentamos hacer por ejemplo un ping a [www.google.es](http://www.google.es) veremos que no somos capaces. Hasta ahora lo que hemos logrado hacer es llegar hasta la instancia, pero ¿y si la instancia quiere actualizarse o tener acceso a Internet por cualquier motivo? Para esta casuística aparece el servicio NAT Gateway, que lo que nos permite es conectarnos/establecer una conexión con el exterior y que nos devuelvan respuesta, pero si se intentase establecer una conexión desde el exterior esta NO sería posible ya que el dispositivo no capacita para conectarte desde fuera.

Lo primero es crear un nat gateway desde el menú VPC:



EC2 Global View ▾

Filter by VPC ▾

**Virtual private cloud**

- Your VPCs
- Subnets
- Route tables
- Internet gateways
- Egress-only internet gateways
- Carrier gateways
- DHCP option sets
- Elastic IPs
- Managed prefixes
- NAT gateways** (selecciónada)
- Peering connections
- Route servers New

**NAT gateway settings**

**Name - optional**  
Create a tag with a key of 'Name' and a value that you specify.  
**mi-ngw**  
The name can be up to 256 characters long.

**Subnet**  
Select a subnet in which to create the NAT gateway.  
**subnet-0c14d6cf7f208ef36 (proyecto-subnet-public1-us-east-1a)**

**Connectivity type**  
Select a connectivity type for the NAT gateway.  
 **Public**  
 Private

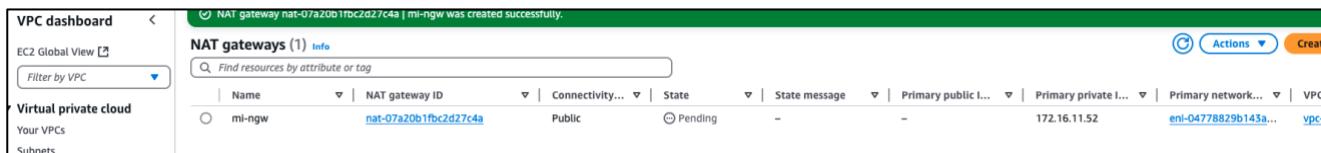
**Elastic IP allocation ID** Info  
Assign an Elastic IP address to the NAT gateway.  
**eipalloc-0cd07a330f15438ba**

**Additional settings** Info

**Allocate Elastic IP**

Aquí aparece por primera vez el concepto de ip elástica, una ip elástica es una ip pública que vamos a poder reutilizar en cualquier servicio, es decir si hubiera un servicio que necesitase de una ip, o bien puedo utilizar las que AWS me da de normalmente de forma dinámica o si ya necesitase una estática que no varíe podría utilizar la ip elástica, por supuesto reservar una ip de esta forma lleva un gasto asociado.

Ya que este servicio necesita de una, si no tenemos una haremos clic en asignar ip elástica y nos creará una (es un coste adicional el tener una ip elástica). La creación del nat gateway lleva un rato y habrá que esperar a que esté en estado “available”. Para este caso el nat gateway lo vamos a crear en la subred pública (ya veremos más adelante por qué).



VPC dashboard

EC2 Global View ▾

Filter by VPC ▾

**NAT gateways (1) Info**

Name	NAT gateway ID	Connectivity...	State	State message	Primary public I...	Primary private I...	Primary network...	VPC
mi-ngw	nat-07a20b1fb2d27c4a	Public	Pending	-	172.16.11.52	eni-04778829b143a...	vpc-	

**NAT gateways (1) Info**

Find resources by attribute or tag

Name: mi-ngw

NAT gateway ID: nat-07a20b1fb2d27c4a

Connectivity: Public

State: Pending

State message: -

Primary public IP: 172.16.11.52

Primary private IP: eni-04778829b143a...

Primary network interface: vpc-

Actions ▾

Create

Ahora vamos a configurar la subred privada para que tenga conexión al exterior. Para ello vamos a crear una tabla de enrutamiento para este dispositivo desde el menú VPC.

### Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the Internet, and your VPN connection.

#### Route table settings

##### Name - optional

Create a tag with a key of 'Name' and a value that you specify.

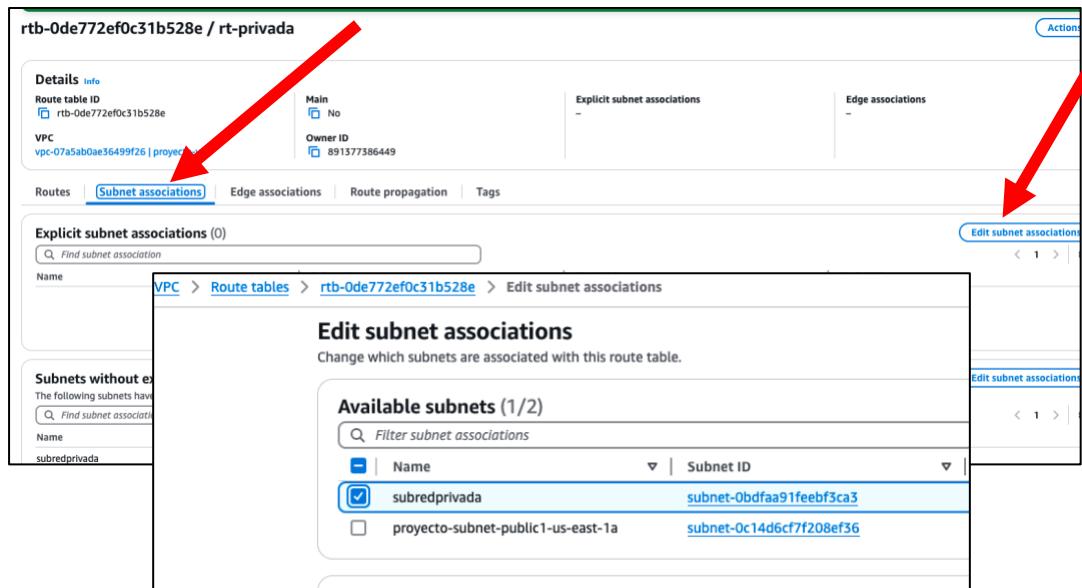
rt-privada

##### VPC

The VPC to use for this route table.

vpc-07a5ab0ae36499f26 ( proyecto-vpc )

Cuando se cree esta tabla le asociaremos la subred privada desde la pestaña "subnet associations" haciendo click en "Edit subnet associations"



rtb-0de772ef0c31b528e / rt-privada

**Details** Info

Route table ID: rtb-0de772ef0c31b528e

VPC: vpc-07a5ab0ae36499f26 | proyecto-vpc

Main: No

Owner ID: 891377386449

Explicit subnet associations: -

Edge associations: -

Actions

Routes | Subnet associations | Edge associations | Route propagation | Tags

**Explicit subnet associations (0)**

Find subnet association

Name

Subnets without explicit associations

The following subnets have no explicit associations:

Find subnet association

Name

subredprivada

**Edit subnet associations**

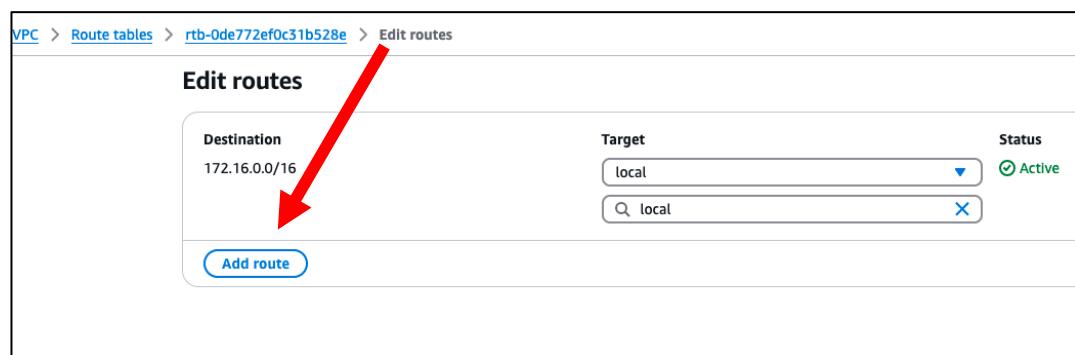
Change which subnets are associated with this route table.

**Available subnets (1/2)**

Filter subnet associations

Name	Subnet ID
<input checked="" type="checkbox"/> subredprivada	subnet-0bdfa91feebf3ca3
<input type="checkbox"/> proyecto-subnet-public1-us-east-1a	subnet-0c14d6cf7f208ef36

Cuando esté disponible iremos a la tabla de enrutamiento de la subred privada (pestaña routes) y podemos ver las rutas preconfiguradas:



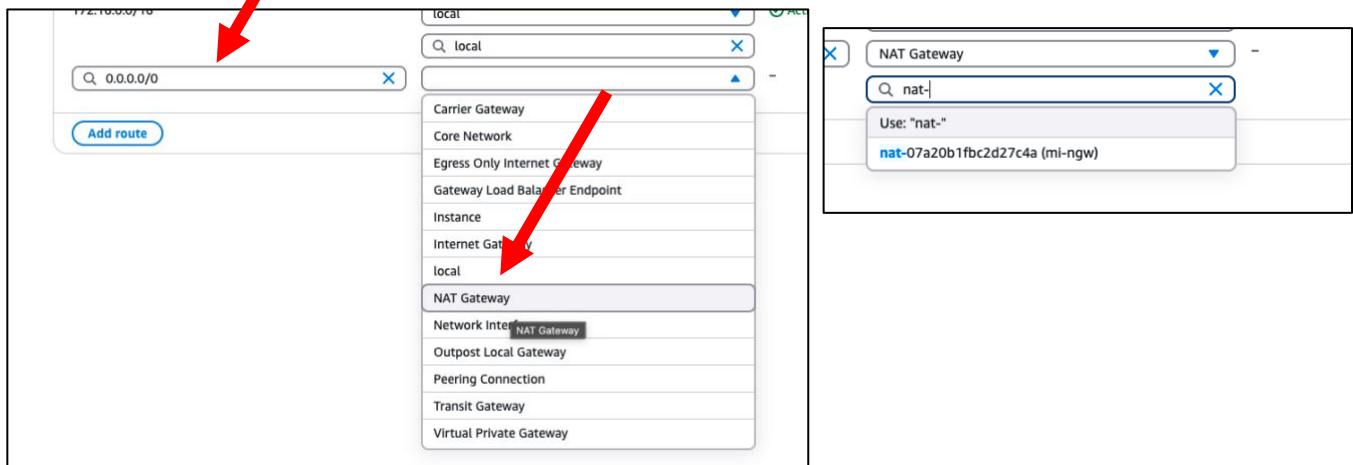
VPC > Route tables > rtb-0de772ef0c31b528e > Edit routes

**Edit routes**

Destination	Target	Status
172.16.0.0/16	local	Active

Add route

Ahora añadiremos una ruta para redirigir el tráfico que vaya a Internet hacia el nat gateway y éste ya se encargará de dar salida.



Ahora podremos hacer dos comprobaciones:

- NO podemos hacer ping desde el exterior:

```
Last login: Sat Apr  5 11:19:14 on ttys003
jorge@mjolnir Downloads % ssh -i labsuser.pem ec2-user@34.232.109.141
```

- Cuando hagamos ping desde la ec2 de la red privada a Google SÍ que nos devolverá respuesta.

```
[ec2-user@ip-172-16-68-71 ~]$ ping www.google.es
PING www.google.es (142.251.179.94) 56(84) bytes of data.
64 bytes from pd-in-f94.1e100.net (142.251.179.94): icmp_seq=1 ttl=105 time=3.20 ms
64 bytes from pd-in-f94.1e100.net (142.251.179.94): icmp_seq=2 ttl=105 time=2.27 ms
64 bytes from pd-in-f94.1e100.net (142.251.179.94): icmp_seq=3 ttl=105 time=2.30 ms
^C
```

## NACL (network Access control list)

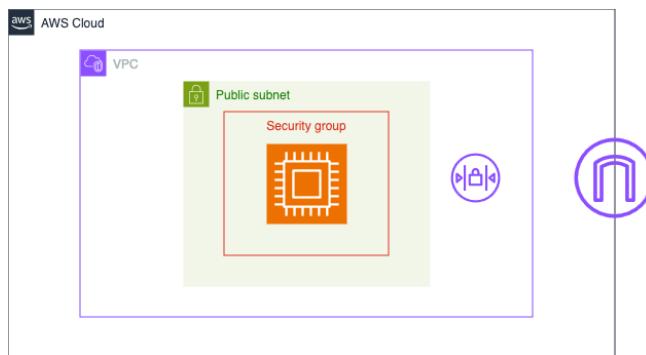
Una NACL es un componente de seguridad que actúa como un firewall (es un firewall) a nivel de subred dentro de una VPC. Hasta ahora hemos estado trabajando con los grupos de seguridad que son reglas con estado, es decir, cuando se establece una conexión ésta se puede devolver y NO es necesario habilitar una regla que lo permita explícitamente. Sin embargo, AWS ofrece otro nivel de seguridad y son las NACL, ACL network o ACL de red.

Es decir, las reglas de seguridad con estado, identificando por ejemplo el caso de un ping, cuando realizamos un ping y añadimos en el grupo de seguridad la regla que permite el ICMP, cuando se realiza el ping la entrada del ping se permite e implícitamente se permite la salida por cualquier puerto ya que es una conexión establecida. Sin embargo, en el caso de las NACL este permiso implícito NO es dado, hay que configurar tanto las reglas de entrada como las de salida.

El objetivo de la práctica es comprobar que los grupos de seguridad son capaces de guardar el estado y la ACLs de red no guardan el estado, para ello montaremos en una VPC una subred pública con su correspondiente EC2, permitiremos la entrada del ping mediante una NACL, pero NO la salida, para comprobar que el ping NO sale y luego más tarde lo permitiremos.

Antes de continuar recuerdo que es recomendable eliminar todo lo que hemos hecho anteriormente para no confundirnos y tener los siguientes conceptos claros. Se puede realizar un reset también del laboratorio, pero dejará éste inaccesible durante un buen rato mientras realiza la limpieza del lab.

### Escenario



### VPC

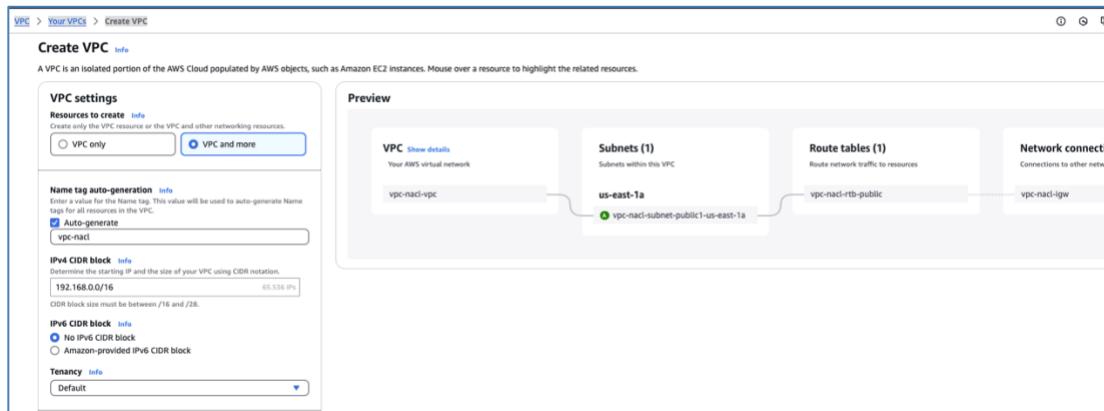
Lo primero es crear la VPC junto con la subred. Para ello seguiremos las siguientes capturas:

AWS, como instancias de Amazon EC2. Deslizar el ratón sobre un recurso para resaltar los recursos relacionados.

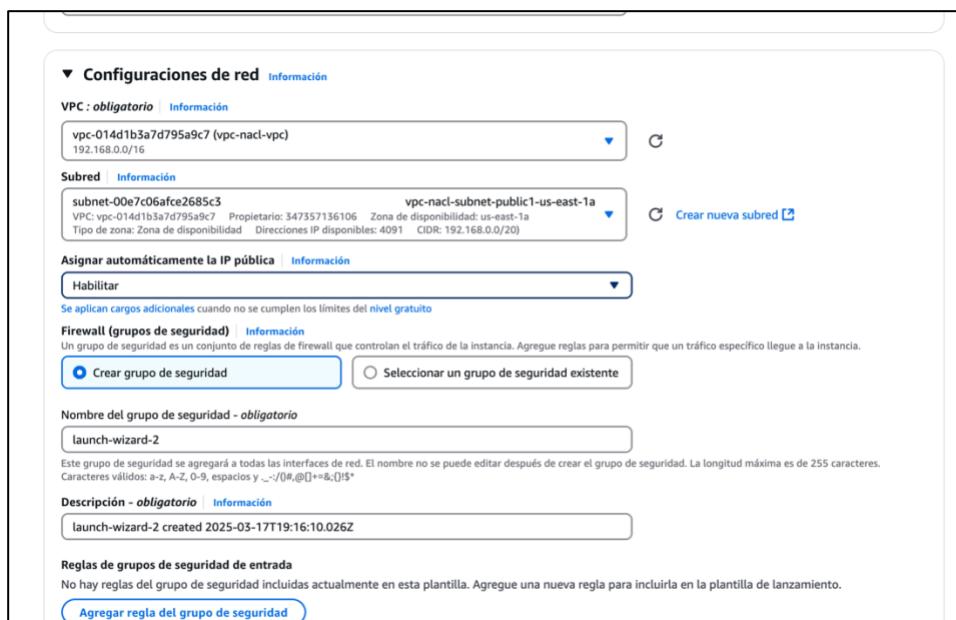
**Vista previa**

VPC	Subredes (1)	Tablas de enrutamiento (1)	Conexiones de red (2)
<a href="#">Mostrar detalles</a> Su red virtual de AWS <b>vpc-nacl-vpc</b>	Subredes dentro de esta VPC <b>us-east-1a</b> <small>vpc-nacl-subnet-public1-us-east-1a</small>	Tablas de enrutamiento (1) Dirigir el tráfico de red a los recursos <b>vpc-nacl-rtb-public</b>	Conexiones de red (2) Conexiones a otras redes <b>vpc-nacl-igw</b> <b>vpc-nacl-vpce-s3</b>

Además, como no vamos a crear redes privadas no nos hará falta crearlas.



Una vez creada la infraestructura pasaremos a lanzar la instancia dentro de la VPC.



Como siempre, podemos entrar por ssh y ver que está hábil la instancia y, ya que estamos, también podemos probar el ping.

```
jorge@mjolnir Downloads % ssh -i labsuser.pem ec2-user@3.92.3.240
The authenticity of host '3.92.3.240 (3.92.3.240)' can't be established.
ED25519 key fingerprint is SHA256:gDAkSdN/hv17zgEaqZB9HREDAD/tfK4zpD3rBHbma9s.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.92.3.240' (ED25519) to the list of known hosts.

# 
Amazon Linux 2023

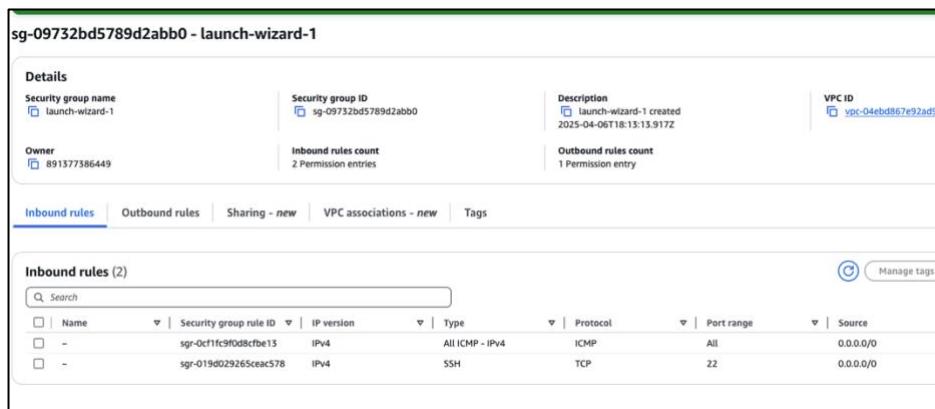
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-192-168-11-23 ~]$ exit
logout
Connection to 3.92.3.240 closed.
jorge@mjolnir Downloads % ping 3.92.3.240
PING 3.92.3.240 (3.92.3.240) 56 data bytes
Request timeout for icmp_seq 0
Request timeout for icmp_seq 1
Request timeout for icmp_seq 2
```

Previamente vamos a modificar el grupo de seguridad y permitir el ping y ver que funciona. Una vez hecha esta comprobación vamos a jugar con las NACL para activar el ping, aunque sea un poco más laborioso.

Hay que tener en cuenta que por defecto TODO el tráfico entrante se deniega. Por lo tanto, daremos permiso al protocolo ICMP en una regla de entrada y salida del ping y luego denegaremos la salida del ping y ver como ya no responde.

El ec2 con el grupo de seguridad configurado para permitir el ping tendría un aspecto como el siguiente:



Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
sgr-0cf1fc9f0d8cfbe13	IPv4	All ICMP - IPv4	ICMP	All	0.0.0.0/0	-
sgr-019d029265ceac578	IPv4	SSH	TCP	22	0.0.0.0/0	-

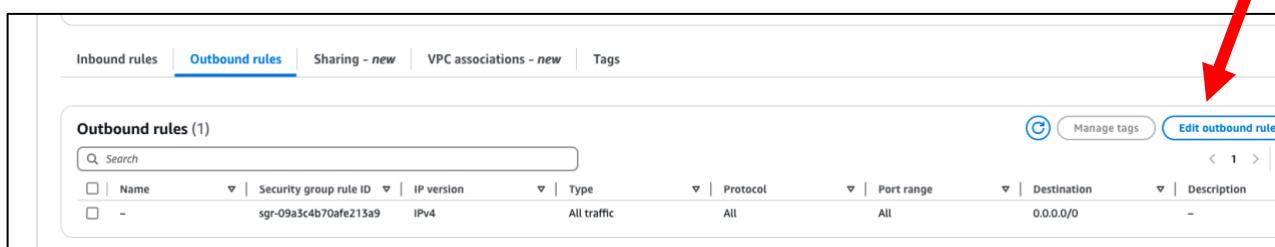
Primero la regla de seguridad con la regla de salida.



Name	Security group rule ID	Port range	Protocol	Source	Security groups	Description
sgr-0cf1fc9f0d8cfbe13	All	ICMP	0.0.0.0/0	-	-	-
sgr-019d029265ceac578	22	TCP	0.0.0.0/0	-	-	-

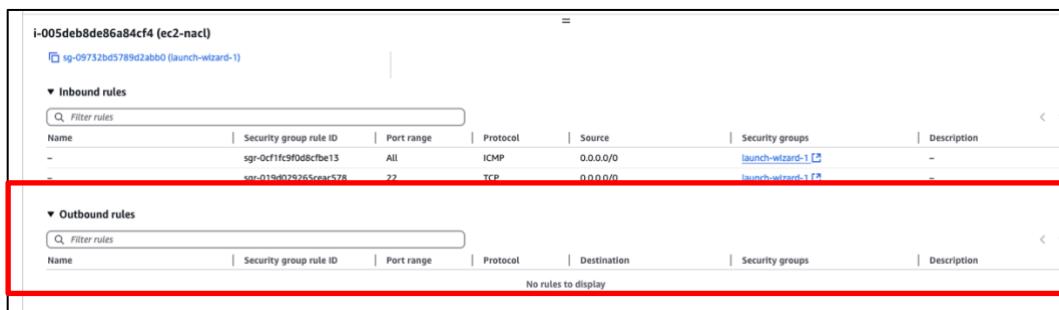
Name	Security group rule ID	Port range	Protocol	Destination	Security groups	Description
sgr-09a3c4b70afe213a9	All	All	0.0.0.0/0	-	-	-

Editamos la regla de seguridad:



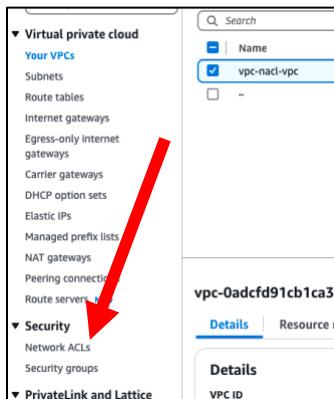
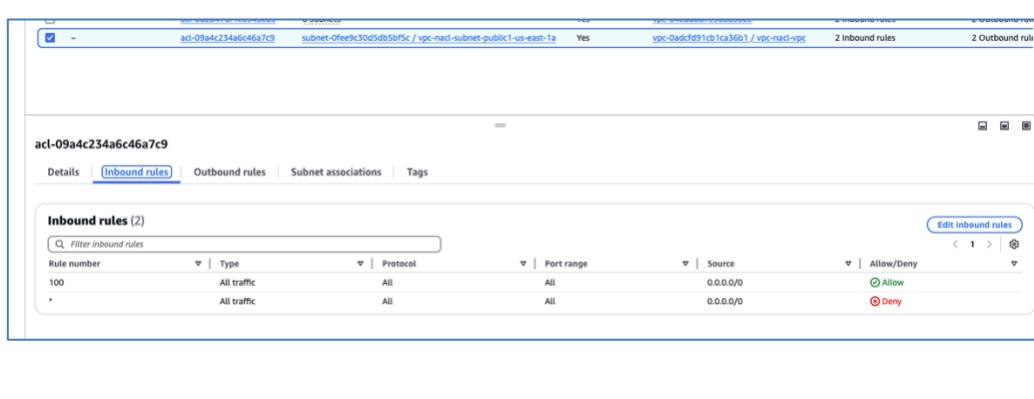
Name	Security group rule ID	Port range	Protocol	Destination	Description
sgr-09a3c4b70afe213a9	IPv4	All traffic	All	All	0.0.0.0/0

Una vez eliminada la regla:



Y podemos comprobar que el ping sigue funcionando:

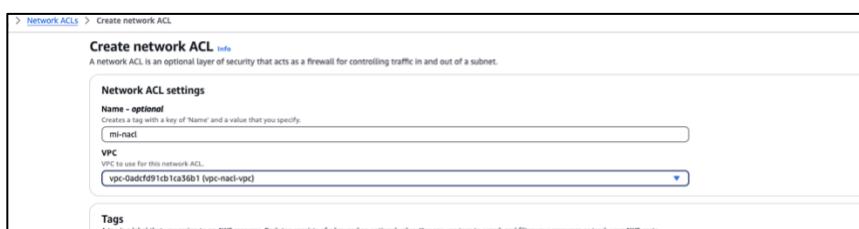
Ahora crearemos la NACL en la VPC con la que estamos trabajando. Se crea justo desde el menú de VPC junto con las reglas de seguridad. En este paso ahora faltaría asociarlo a una subred o conjunto de subredes a controlar.

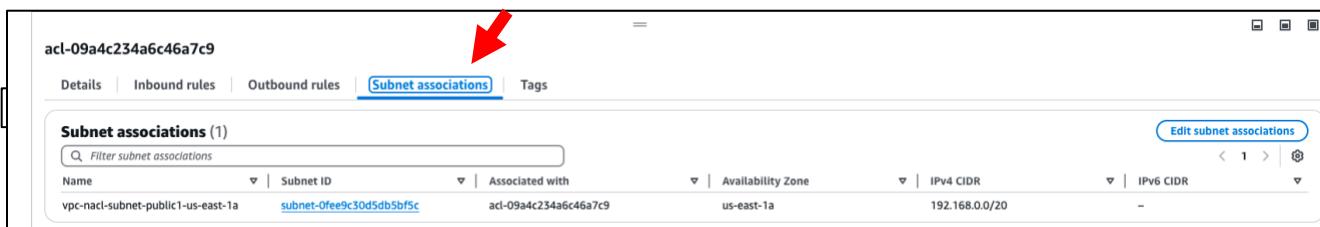
Rule number	Type	Protocol	Port range	Source	Allow/Deny
100	All traffic	All	All	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

Por defecto podemos comprobar que ya existe una NACL asociada a la VPC, que permite todo el tráfico de entrada y salida. Desde el menú de ACL de red seleccionando la regla creada podremos modificar las reglas de entrada y salida.

Ahora crearemos el NACL en la VPC con la que estamos trabajando.



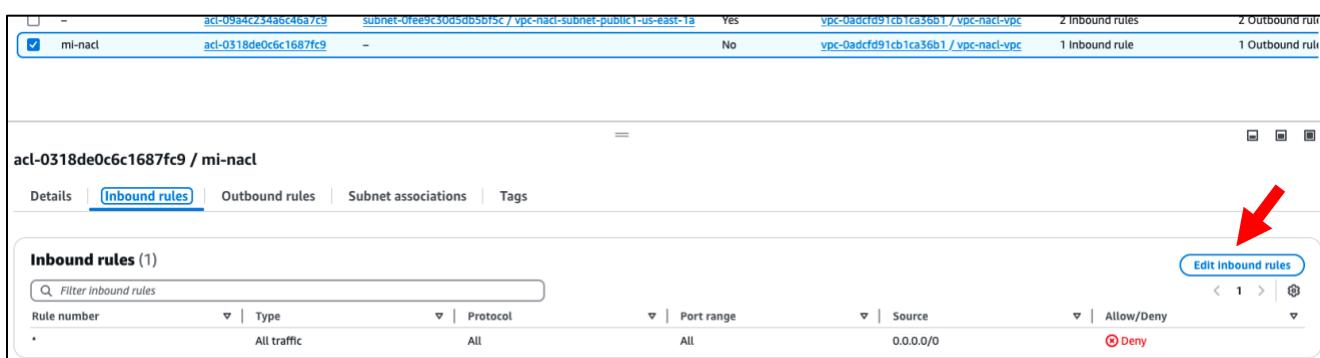
Y comprobamos que tiene asociada la subred pública desde la pestaña "subnet associations"



Name	Subnet ID	Associated with	Availability Zone	IPv4 CIDR	IPv6 CIDR
vpc-nacl-subnet-public1-us-east-1a	subnet-0fee9c30d5db5bf5c	acl-09a4c234a6c46a7c9	us-east-1a	192.168.0.0/20	-

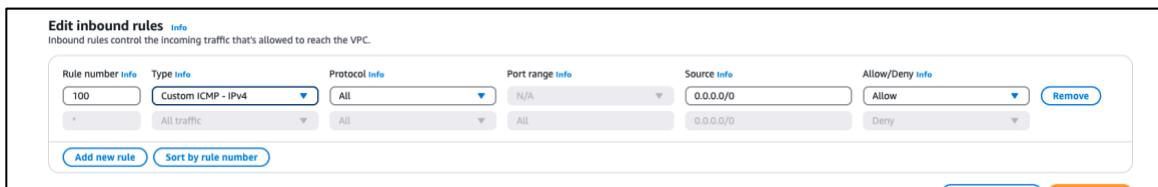
Desde el menú ACL de red, seleccionando la regla creada, podremos modificar las reglas de entrada y salida. Las pruebas serán las siguientes:

- Añadir regla de entrada ICMP. El ping NO funciona. Desde la regla NACL en la pestaña Inbound rules



Rule number	Type	Protocol	Port range	Source	Allow/Deny
*	All traffic	All	All	0.0.0.0/0	Deny

Desde el botón “edit inbound rules” podremos añadir la regla nueva:



Y ahora ya nos aparece la nueva regla:



Cabe destacar que las reglas se evalúan con una prioridad ascendente y en cuanto la satisface ya no se comprueban las de menor prioridad.

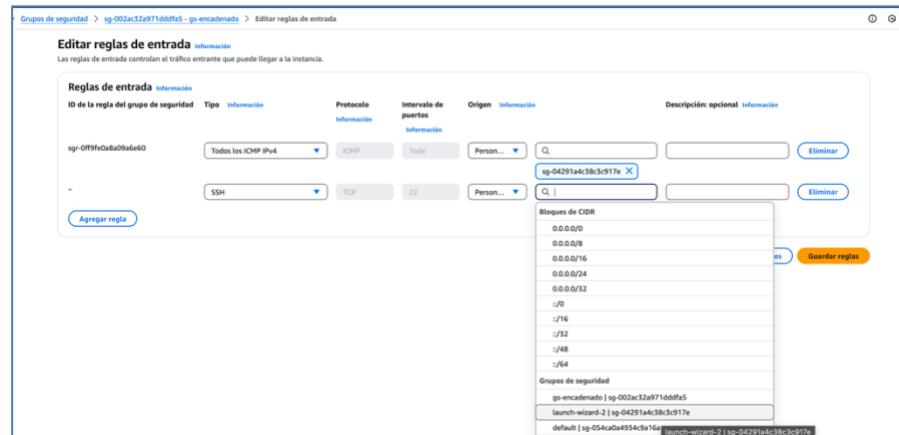
```
jorge@mjolnir Downloads % ping 54.87.133.32
PING 54.87.133.32 (54.87.133.32): 56 data bytes
Request timeout for icmp_seq 0
Request timeout for icmp_seq 1
Request timeout for icmp_seq 2
Request timeout for icmp_seq 3
Request timeout for icmp_seq 4
```

- Ahora añadiremos la regla de salida ICMP en la regla NACL. El ping SI funciona.

```
jorge@mjolnir Downloads % ping 3.92.3.240
PING 3.92.3.240 (3.92.3.240): 56 data bytes
64 bytes from 3.92.3.240: icmp_seq=0 ttl=112 time=107.059 ms
64 bytes from 3.92.3.240: icmp_seq=1 ttl=112 time=115.187 ms
64 bytes from 3.92.3.240: icmp_seq=2 ttl=112 time=108.847 ms
64 bytes from 3.92.3.240: icmp_seq=3 ttl=112 time=105.259 ms
```

## Reglas encadenadas

Ahora lo que vamos a hacer es crear un nuevo grupo de seguridad (con el nombre 'gs-encadenado' por ejemplo) y añadir una nueva regla de entrada, asignándole como origen el grupo de seguridad que tiene la instancia primera.



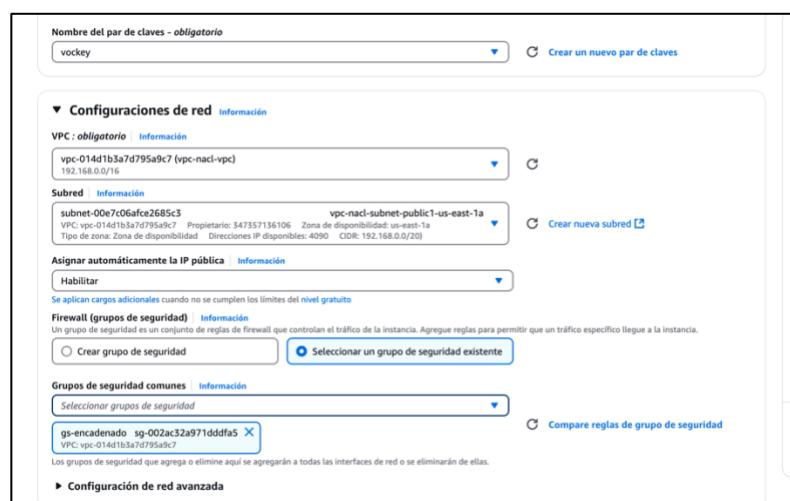
Grupos de seguridad > sg-002ac32a971ddfa5 > gs-encadenado > Editar reglas de entrada

**Reglas de entrada**

ID de la regla del grupo de seguridad	Tipo	Protocolo	Intervalo de puertos	Origen	Descripción (opcional)
sg-0ff9fc0a8e09fe660	Todos los ICMP IPv4	ICMP	Todos	Person... (Q)	sg-04291a4c38c3c917e X
-	SSH	TCP	22	Person... (Q)	

**Guardar regla**

A la hora de crear la instancia ec2 le asignamos al nuevo grupo de seguridad 'encadenado', en lugar de a configurar es una nueva regla de seguridad y lo que le vamos a asignar como



Nombre del par de claves - obligatorio: vockey

Configuraciones de red

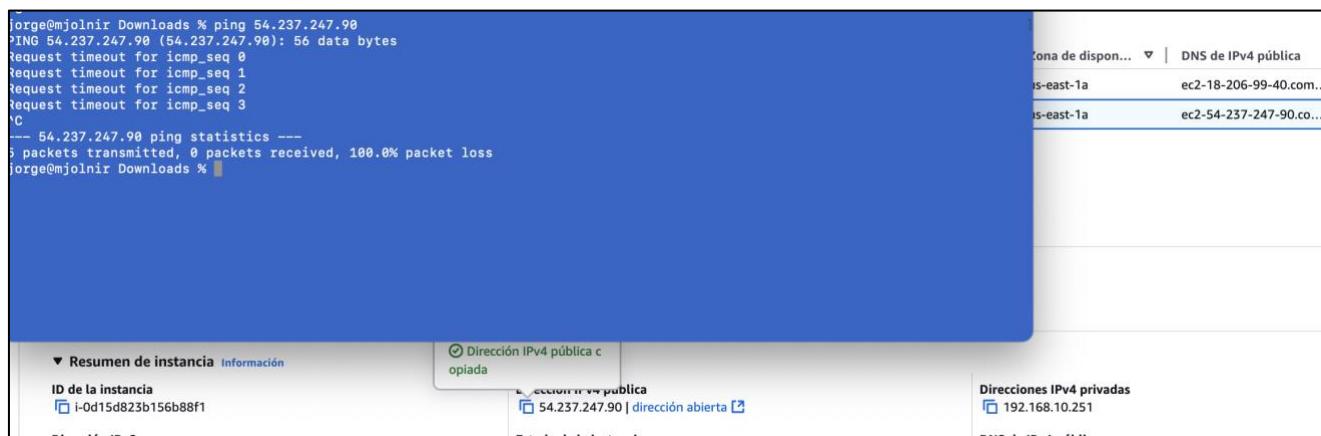
VPC: obligatorio

Subred: subred-0067c66afe2685c3

Asignar automáticamente la IP pública

Grupos de seguridad comunes: gs-encadenado sg-002ac32a971ddfa5

Y cuando queramos por ejemplo hacer un ping desde fuera (desde el ordenador local, por ejemplo):



```
jorge@mjolnir Downloads % ping 54.237.247.98
PING 54.237.247.98 (54.237.247.98): 56 data bytes
Request timeout for icmp_seq 0
Request timeout for icmp_seq 1
Request timeout for icmp_seq 2
Request timeout for icmp_seq 3
'C
--- 54.237.247.98 ping statistics ---
3 packets transmitted, 0 packets received, 100.0% packet loss
jorge@mjolnir Downloads %
```

Resumen de instancia

ID de la instancia: i-0d15d823b156b88f1

Dirección IPv4 pública: 54.237.247.90 | dirección abierta

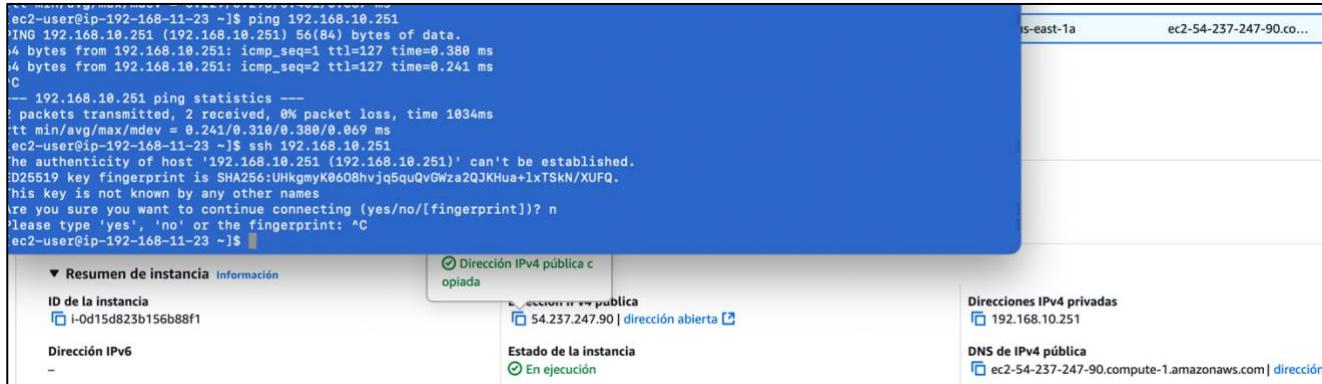
Direcciones IPv4 privadas: 192.168.10.251

Zona de disponibilidad: us-east-1a

DNS de IPv4 pública: ec2-18-206-99-40.com...

DNS de IPv4 privada: ec2-54-237-247-90.co...

Vemos que no lo permite, pero sin embargo si lo realizamos desde dentro de la primera instancia creada, Sí que lo permite:



```
ec2-user@ip-192-168-11-23 ~$ ping 192.168.10.251
PING 192.168.10.251 (192.168.10.251) 56(84) bytes of data.
4 bytes from 192.168.10.251: icmp_seq=1 ttl=127 time=0.380 ms
4 bytes from 192.168.10.251: icmp_seq=2 ttl=127 time=0.241 ms
C
--- 192.168.10.251 ping statistics ---
1 packets transmitted, 2 received, 0% packet loss, time 1034ms
rtt min/avg/max/mdev = 0.241/0.318/0.380/0.069 ms
ec2-user@ip-192-168-11-23 ~$ ssh 192.168.10.251
host authenticity of host '192.168.10.251 (192.168.10.251)' can't be established.
DSA25519 key fingerprint is SHA256:UHkgmyK0e0Bhvjq5qQuQvGwza2QJKHua+lxTSkN/XUFQ.
his key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? n
Please type 'yes', 'no' or the fingerprint: ^C
ec2-user@ip-192-168-11-23 ~$
```

Resumen de instancia [Información](#)

ID de la instancia [i-0d15d823b156b88f1](#)

Dirección IPv6

—

Dirección IPv4 pública [copia](#)

[54.237.247.90 | dirección abierta](#)

Estado de la instancia [En ejecución](#)

Direcciones IPv4 privadas [192.168.10.251](#)

DNS de IPv4 pública [ec2-54-237-247-90.compute-1.amazonaws.com | dirección](#)

Cabe recordar que para conectarse a la primera instancia tendremos que usar el parámetro -A en la conexión y cargar el certificado en memoria.

## Tareas

- La tarea consiste en realizar un despliegue de una típica arquitectura de dos capas, donde en la parte pública tendremos un servidor web, y en la privada un servidor de base de datos:
  - Crea una VPC con dos subredes, una pública y otra privada. Para que desde la subred privada se pueda acceder al exterior, deberás utilizar un NAT Gateway.
  - En la subred pública lanza una instancia EC2 e instala un servidor web Apache, que será accesible desde cualquier equipo externo a la VPC por el puerto 80, utilizando tanto su nombre DNS como su dirección IP pública.
  - En la subred privada, lanza otra instancia EC2 e instala un servicio de MySQL.
  - Los grupos de seguridad asociados a las instancias deben permitir sólo el tráfico necesario, utilizando una regla encadenada en el caso del grupo de seguridad asociado a la instancia de base de datos, de manera que sólo acepte tráfico entrante del protocolo adecuado desde la instancia del servidor web
  - Conéctate con ssh a la instancia de la subred pública, y realiza una prueba de conexión al servidor de base de datos, utilizando para ello la aplicación del cliente de mysql que tendrás que instalar previamente
- Adjunta a la tarea las siguientes capturas: mapa de la VPC, configuración de las reglas de entrada de los grupos de seguridad de las dos instancias, conexión al servidor Apache desde un navegador y conexión al servicio de base de datos desde la instancia del servidor web