

# LAB-264

RECURSOS DE RED PARA UNA VPC

SONY ETCHEVERRY, AGUSTÍN ESTECHE, FABRIZIO FIORINA,  
FACUNDO MORALES, AGUSTÍN RODRIGUEZ,  
BENJAMIN SABAÑO, MICHAELA VIAZZI



# Objetivos

- Crear una VPC, una puerta de enlace de Internet, una tabla de enrutamiento, un grupo de seguridad, una lista de acceso de redes y una instancia EC2 para generar una red enrutable dentro de la VPC.

# Ticket del cliente

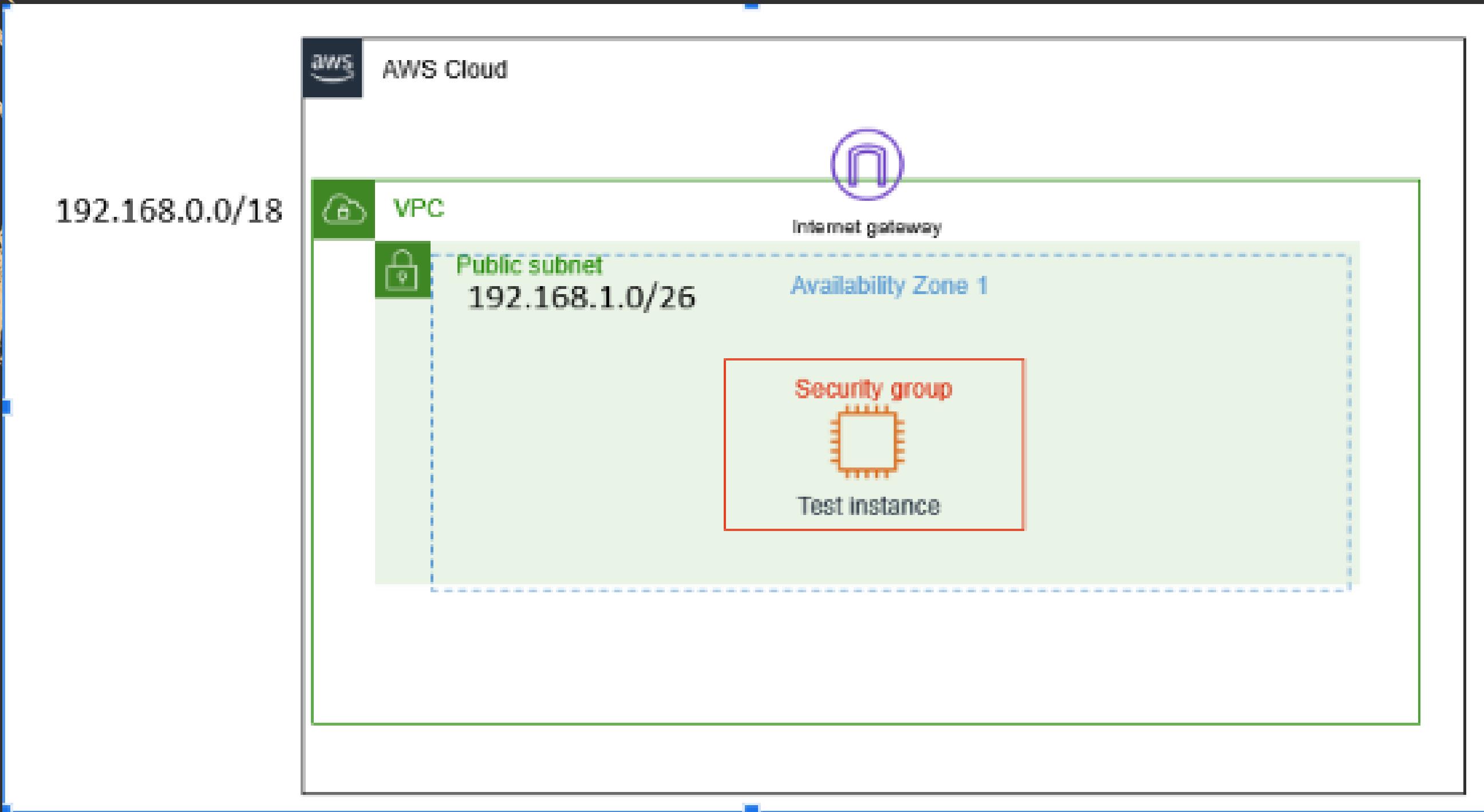
## Correo electrónico del cliente

¡Hola, equipo de soporte en la nube!

Hace unos días, me puse en contacto con ustedes para solicitar ayuda a fin de configurar mi VPC. Pensé que sabía adjuntar todos los recursos para establecer una conexión a Internet, pero ni siquiera puedo hacer ping por fuera de la VPC. ¡Todo lo que necesito es hacer ping! ¿Me pueden ayudar a configurar mi VPC donde tenga conectividad de red y pueda hacer ping? A continuación, se encuentra la arquitectura. ¡Gracias!

Brock, propietario de la empresa emergente

# Arquitectura del cliente



(El cliente nos da dos números IP, uno para la VPC y otro para la subred pública)

Dentro del servicio de VPC, vamos a crear una nueva VPC y la vamos a configurar así:

VPC > Your VPCs > Create VPC

## Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

### VPC settings

Resources to create: Info  
Create only the VPC resource or the VPC and other networking resources.

VPC only     VPC and more

Name tag - optional  
Creates a tag with a key of 'Name' and a value that you specify.  
Test VPC

IPv4 CIDR block: Info  
 IPv4 CIDR manual input  
 IPAM-allocated IPv4 CIDR block

IPv4 CIDR  
192.168.0.0/16

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

IPv6 CIDR block: Info  
 No IPv6 CIDR block  
 IPAM-allocated IPv6 CIDR block  
 Amazon-provided IPv6 CIDR block  
 IPv6 CIDR owned by me

Tenancy: Info  
Default

# Configuración de la subred

**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.  
  
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.

**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.

**IPv4 subnet CIDR block**  
 16 IPs

**Tags - optional**

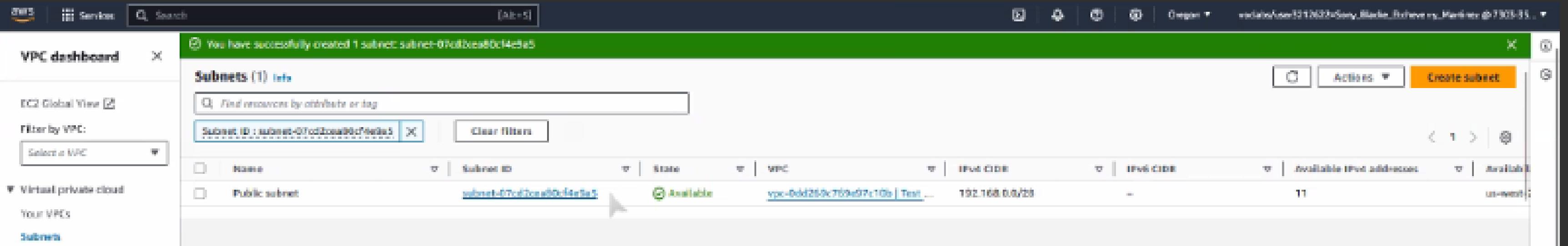
Key	Value - optional
<input type="text" value="Name"/> <a href="#">X</a>	<input type="text" value="Public subnet"/> <a href="#">X</a> <a href="#">Remove</a>

[Add new tag](#)  
You can add 49 more tags.  
[Remove](#)

[Add new subnet](#)

[Cancel](#) [Create subnet](#)

# Creamos las subredes



The screenshot shows the AWS VPC dashboard with a green success message at the top: "You have successfully created 1 subnet: subnet-0701bea60c14c9a5". The main section displays a table of subnets, with one row highlighted. The table columns include Name, Subnet ID, State, VPC, IPv4 CIDR, IPv6 CIDR, Available IPv4 addresses, and Available IPv6 addresses. The highlighted row shows:

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Available IPv4 addresses	Available IPv6 addresses
Public subnet	subnet-0701bea60c14c9a5	Available	vpc-0dd26e7b9d97c10b   Test ...	192.168.0.0/28	-	11	-

# Creamos las "Route Table"

**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.

**VPC**  
The VPC to use for this route table.

**Tags**  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text" value="Name"/> <input type="button" value="X"/>	<input type="text" value="Public route table"/> <input type="button" value="X"/> <input type="button" value="Remove"/>
<input type="button" value="Add new tag"/>	

You can add 49 more tags.

**Route tables (1/3) Info**

#	Name	Route table ID	Origin	Default subnet association	Edge associations	Main	VPC	Owner ID
1	-	rtb-0092961ed5f5bea3	-	-	-	No	vpc-6dd269c766e97fc10b	73035594547
2	-	rtb-0a5a038d72ec555e0	-	-	-	No	vpc-6dd269c766e97fc10b (Test ...)	73035594547
3	Public route table	rtb-0a6b4911931f18620	-	-	-	No	vpc-6dd269c766e97fc10b (Test ...)	73035594547

**rtb-0a6b4911931f18620 / Public route table**

Details | Focus | Subnet associations | Edge associations | Route propagation | Tags

**Details**

Route table ID	<input checked="" type="radio" value="rtb-0a6b4911931f18620"/> rtb-0a6b4911931f18620	From	<input checked="" type="radio" value="No"/> No	Default subnet associations	-
VPC	vpc-6dd269c766e97fc10b (Test VPC)	Owner ID	<input checked="" type="radio" value="73035594547"/> 73035594547	Edge associations	-

# Creamos las Internet Gateway

The image shows two screenshots of the AWS VPC console. The left screenshot is titled 'Create internet gateway' and shows the configuration for a new internet gateway. It includes fields for a 'Name tag' (set to 'IGW test VPC') and optional 'Tags' (one tag named 'Name' with value 'IGW test VPC'). The right screenshot is the 'Internet gateways' dashboard, which lists one gateway with the ID 'igw-09a5f1ffbe001186a'. A red arrow points from the 'Create internet gateway' button in the top right of the dashboard to the 'Create internet gateway' button in the left screenshot.

**Create internet gateway**

An Internet gateway is a virtual router that connects a VPC to the internet. To create a new Internet gateway specify the name for the gateway below.

**Internet gateway settings**

**Name tag**  
Creates a tag with a key of 'Name' and a value that you specify.  
IGW test VPC

**Tags - optional**  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key Value - optional  
Name IGW test VPC Remove

Add new tag You can add 49 more tags.

Cancel Create internet gateway

**VPC dashboard**

**Internet gateways (1)**

Name	Internet gateway ID	State	Actions
-	igw-09a5f1ffbe001186a	Attached	<a href="#">Select a VPC</a>

EC2 Global View Filter by VPC: Select a VPC

**Virtual private cloud**

Your VPCs Subnets Route tables

**Internet gateways**

Egress-only internet gateways Carrier gateways DHCP option sets Elastic IPs Managed prefix lists Endpoints Endpoint services NAT gateways Peering connections

Network ACLs Security groups Rule groups Domain lists

Select an Internet gateway above

# Agregamos la misma a la VPC

VPC > Internet gateways > Attach to VPC (igw-00852c2a1a195ab0d)

### Attach to VPC (igw-00852c2a1a195ab0d) Info

**VPC**  
Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

**Available VPCs**  
Attach the internet gateway to this VPC.

AWS Command Line Interface command

Cancel **Attach Internet gateway**

✓ Updated routes for rtb-0eeb49f7193f18020 / Public route table successfully X

► Details

VPC > Route tables > rtb-0eeb49f7193f18020

### rtb-0eeb49f7193f18020 / Public route table Actions ▾

**Details** Info

Route table ID	rtb-0eeb49f7193f18020	Main	No
VPC	vpc-0dd269c769e97c10b   Test VPC	Owner ID	750335594547
Explicit subnet associations	-	Edge associations	-

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

**Routes (2)** Both ▾ Edit routes

Destination	Target	Status	Propagated
0.0.0.0/0	igw-00852c2a1a19...	Active	No
192.168.0.0/18	local	Active	No

# Editamos las rutas

AWS Services Search [Alt+S] Ore vclabs/user3212622=Sony\_Bla...\_Etcheverry\_Mar

VPC > Route tables > rtb-0eef49f7193f18020 > Edit routes

### Edit routes

**Route 1**

Destination	Target	Status
192.168.0.0/18	local	Active
	<input type="text" value="local"/> <input type="button" value="X"/>	

Propagated: No

---

**Route 2**

Destination	Target	Status
0.0.0.0/0	Internet Gateway	-
	<input type="text" value="igw-00852c2a1a195ab0d"/> <input type="button" value="X"/>	

Propagated: No

Cancel

AWS Services Search [Alt+S] Ore vclabs/user3212622=Sony\_Bla...\_Etcheverry\_Mar

VPC dashboard >

Updated routes for rtb-0eef49f7193f18020 / Public route table successfully

Route tables (1/3) Info Actions Create route table

Name	Route table ID	Explicit subnet assoc...
	rtb-009b9a0ade596bea8	-
	rtb-0a3a058bd2cd953c0	-
<input checked="" type="checkbox"/> Create route table	rtb-0eef49f7193f18020	-

Virtual private cloud

Your VPCs Subnets Route tables Internet gateways Egress-only internet gateways Carrier gateways DHCP option sets Elastic IPs Managed prefix lists Endpoints Endpoint services NAT gateways Peering connections

Security Network ACLs Security groups DNS firewall Rule groups Domain lists

Details

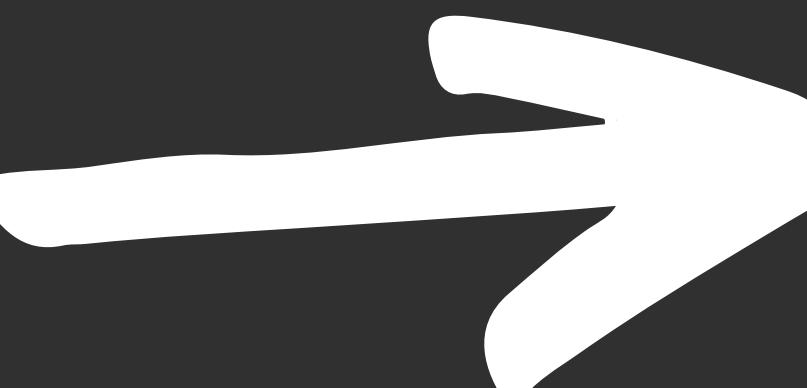
Route table ID	Main	Explicit subnet associations	Edge associations
rtb-0eef49f7193f18020	No	-	-
VPC	73035594547	vpc-0dd269c769e97c	-

The screenshot shows the AWS VPC Route Tables interface. The URL in the address bar is `voclabs/user3212622=Sony_Blacke_Etcheverry_Mar`. The page title is "Edit subnet associations". The main content area is titled "Available subnets (1/1)" and contains a table with one row. The table columns are: Name, Subnet ID, IPv4 CIDR, IPv6 CIDR, and Route tab. The single row shows "Public subnet" with Subnet ID `subnet-07cd2cea80cf4e9a5`, IPv4 CIDR `192.168.0.0/28`, and Route tab `Main (rtb-0eeb49f7193f18020)`. Below this is a "Selected subnets" section containing the same subnet entry. At the bottom are "Cancel" and "Save associations" buttons, with the cursor hovering over the "Save associations" button.

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route tab
Public subnet	<a href="#">subnet-07cd2cea80cf4e9a5</a>	192.168.0.0/28	-	Main (rtb-0eeb49f7193f18020)

Seleccionamos la  
(Public Subnet) y guardamos cambios para que todo se  
asocie correctamente

Vamos a VPC  
==> Route tables



# Creamos un ACL de red con la test VPC.

VPC dashboard

Network ACLs (1/2) [Info](#)

Actions [▼](#) Create network ACL

Find resources by attribute or tag

Name	Network ACL ID	Associated with
-	acl-0f46d3a8a157fd798	subnet-07cd2cea80cf4e9a5 / Public subnet
<input checked="" type="checkbox"/>	acl-05f5512a2b3797612	4 Subnets

Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Endpoints

Endpoint services

NAT gateways

Peering connections

Security

Network ACLs [Details](#)

Security groups

DNS firewall

Rule groups

Domain lists

Details

Inbound rules

Outbound rules

Subnet associations

Tags

Details

Network ACL ID	Associated with	Default	VPC ID
acl-05f5512a2b3797612	4 Subnets	Yes	vpc-02190fc8bf9fdeb
Owner	730335594547		88

VPC > Network ACLs > Create network ACL

### Create network ACL [Info](#)

A network ACL is an optional layer of security that acts as a firewall for controlling traffic in and out of a subnet.

#### Network ACL settings

Name - *optional*  
Creates a tag with a key of 'Name' and a value that you specify.

Public Subnet NACL

VPC  
VPC to use for this network ACL.

vpc-0dd269c769e97c10b (Test VPC)

#### Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key  Name  Value - *optional*  Public Subnet NACL  Remove tag

Add tag

You can add 49 more tags

Cancel [Create network ACL](#)

# Editamos las reglas inbound

VPC > Network ACLs > acl-04c4f15b0f79fcab7 / Public Subnet NACL > Edit inbound rules

**Edit inbound rules** Info

Inbound rules control the incoming traffic that's allowed to reach the VPC.

**Inbound rule 1**

Rule number <small>Info</small>	Type <small>Info</small>	Protocol <small>Info</small>
100	All traffic	All
Port range <small>Info</small>	Source <small>Info</small>	Allow/Deny <small>Info</small>
All	0.0.0.0/0	Allow

**Inbound rule 2**

Rule number <small>Info</small>	Type <small>Info</small>	Protocol <small>Info</small>
*	All traffic	All
Port range <small>Info</small>	Source <small>Info</small>	Allow/Deny <small>Info</small>
All	0.0.0.0/0	Deny

**Add new rule** **Sort by rule number**

Cancel **Preview changes** **Save changes**

VPC > Network ACLs > acl-04c4f15b0f79fcab7 / Public Subnet NACL > Edit outbound rules

**Edit outbound rules** Info

Outbound rules control the outgoing traffic that's allowed to leave the VPC.

**Outbound rule 1**

Rule number <small>Info</small>	Type <small>Info</small>	Protocol <small>Info</small>
100	All traffic	All
Port range <small>Info</small>	Destination <small>Info</small>	Allow/Deny <small>Info</small>
All	0.0.0.0/0	Allow

**Outbound rule 2**

Rule number <small>Info</small>	Type <small>Info</small>	Protocol <small>Info</small>
*	All traffic	All
Port range <small>Info</small>	Destination <small>Info</small>	Allow/Deny <small>Info</small>
All	0.0.0.0/0	Deny

**Add new rule** **Sort by rule number**

Cancel **Preview changes** **Save changes**



## Basic details

Security group name [Info](#)

public security group

Name cannot be edited after creation.

Description [Info](#)

allows public access

VPC [Info](#)

vpc-0dd269c769e97c10b (Test VPC)

Configuramos el grupo de seguridad.  
Añadimos tres reglas

## Inbound rules [Info](#)

### Inbound rule 1

Type [Info](#)

SSH

Protocol [Info](#)

TCP

Port range [Info](#)

22

Source type [Info](#)

Custom

Source [Info](#)

Q

Description - optional [Info](#)

### Inbound rule 2

Type [Info](#)

HTTP

Protocol [Info](#)

TCP

Port range [Info](#)

80

Source type [Info](#)

Custom

Source [Info](#)

Q

Description - optional [Info](#)

### Inbound rule 3

Type [Info](#)

HTTPS

Protocol [Info](#)

TCP

Port range [Info](#)

443

Source type [Info](#)

Custom

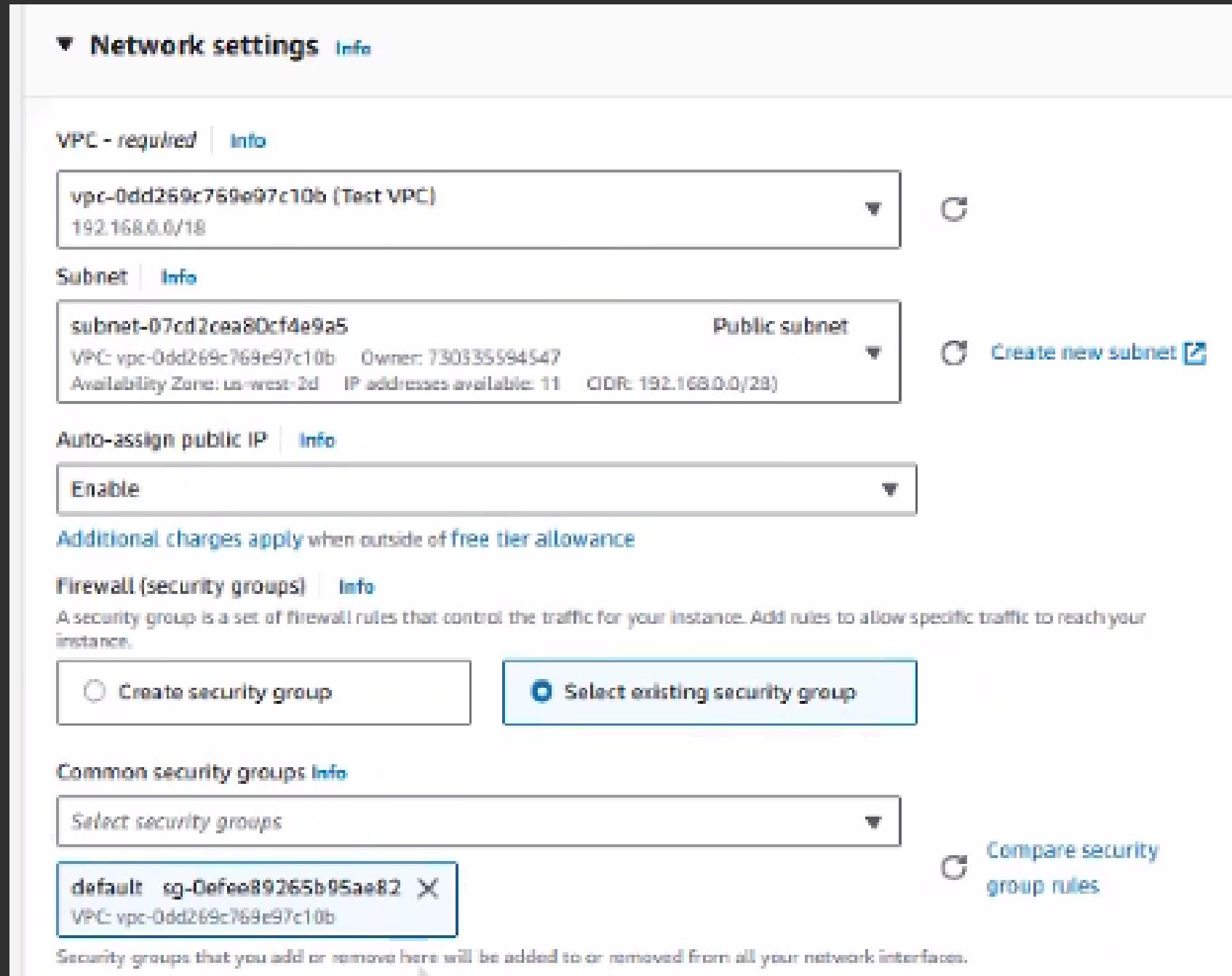
Source [Info](#)

Q

Description - optional [Info](#)

Add rule

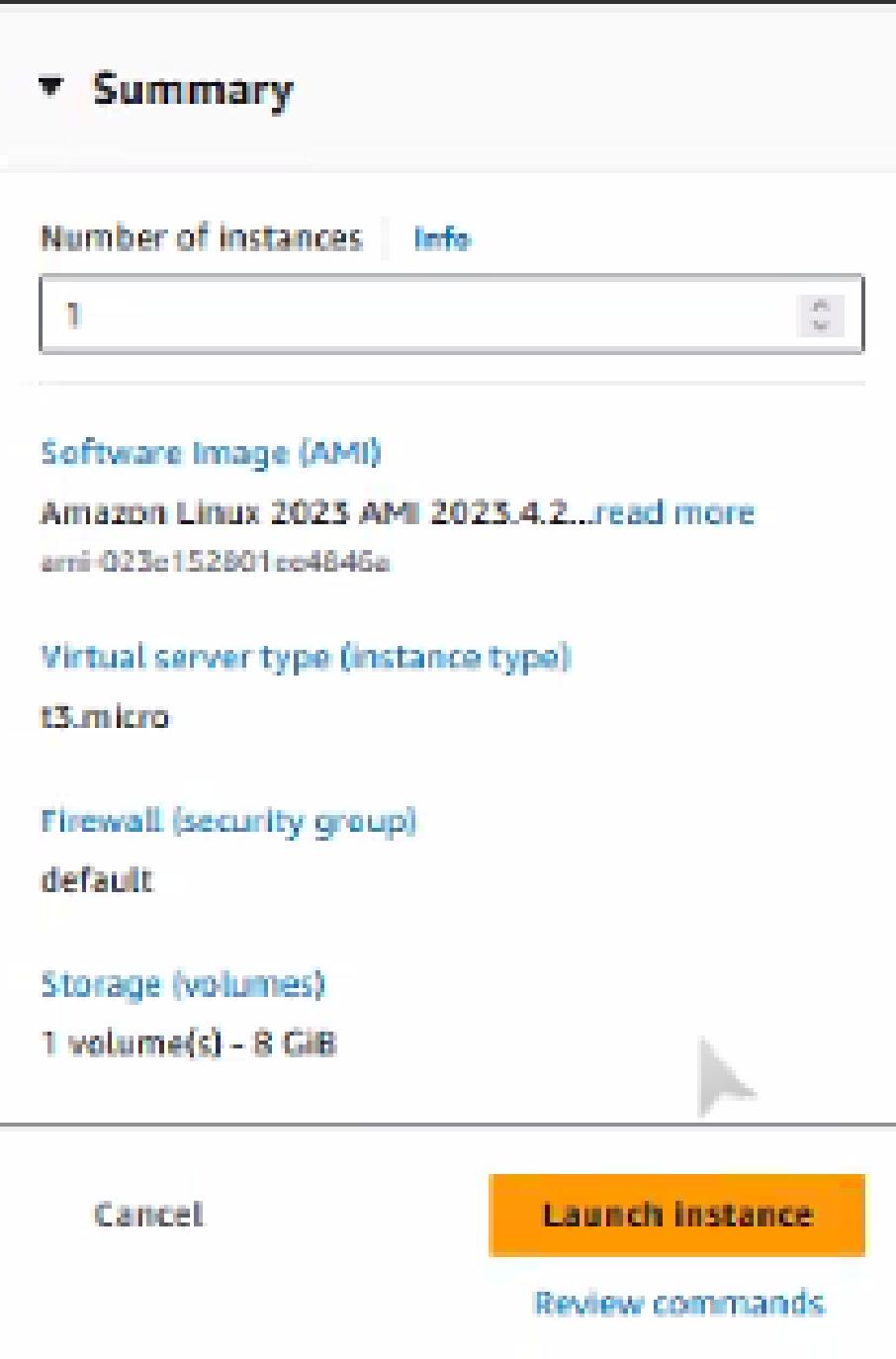
# Tarea 2: Iniciar la instancia EC2 y SSH en la instancia.



Lanzamos una instancia EC2 y procedemos a configurarla, dejando su nombre y etiquetas en blanco, elegimos la AMI de Amazon Linux 2023, volvemos la instancia una de tipo t3.micro y en el par de claves seleccionamos “vokey”.

---

Luego de esto nos dirigimos a la sección “Network settings” donde elegimos como nuestra VPC la opción “Public VPC”, además de seleccionar una subnet pública y permitimos que se asigne de forma automática una IP pública, eligiendo entonces un grupo de seguridad público.



Verificamos que todos nuestros cambios fueron ejecutados correctamente y lanzamos la instancia.

Instances (1) <a href="#">Info</a>						
		<input type="text"/> Find Instance by attribute or tag (case-sensitive)		All states <a href="#">▼</a>		
<input type="checkbox"/>	Name <a href="#">▼</a>	Instance ID	Instance state	Instance type	Status check	Alarm status
<input type="checkbox"/>		I-0fe991a145dd2c90b	<a href="#">Running</a> <a href="#">Q</a> <a href="#">Q</a>	t3.micro	<a href="#">2/2 checks passed</a> <a href="#">View alarms +</a>	us-west-2d

Finalmente volvemos al panel de “Instances” y esperamos a que su estado diga “running”, finalizando entonces este Lab.

MUCHAS  
GRACIAS!