

AMAZON VPC: CREACIÓN DE UNA INFRAESTRUCTURA DE RED PERSONALIZADA

Amazon Virtual Private Cloud (Amazon VPC) es un servicio esencial en la nube de Amazon Web Services (AWS) que permite a las organizaciones crear su propia red virtual privada en el entorno de la nube. Con Amazon VPC, se puede diseñar y personalizar una red de manera eficiente, lo cual brinda un control total sobre la infraestructura de red en AWS. Esta práctica tiene como objetivo proporcionar una comprensión profunda de cómo configurar y utilizar Amazon VPC para crear una infraestructura de red segura y escalable.

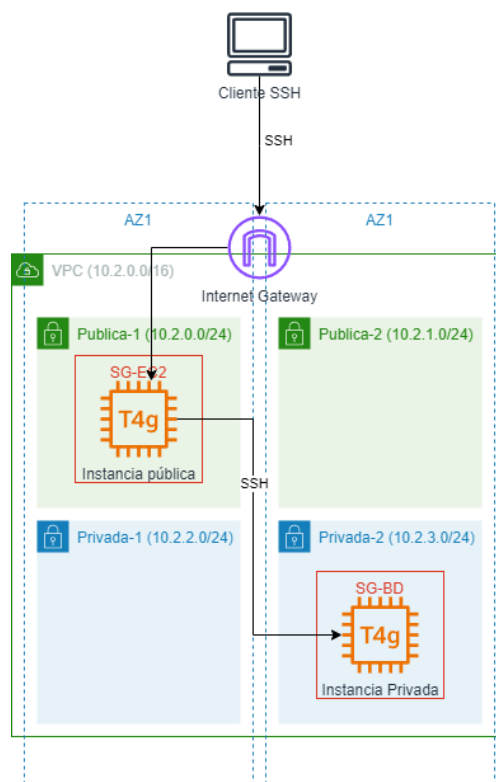
En esta práctica, abordaremos varios aspectos clave de Amazon VPC, desde la creación de una VPC desde cero hasta la configuración de subredes públicas y privadas. También se mostrará cómo lanzar instancias EC2 en una subred pública y, lo que es aún más importante, a establecer una comunicación segura con instancias EC2 ubicadas en subredes privadas. La habilidad de separar las instancias en subredes públicas y privadas es fundamental para la seguridad y el rendimiento de las aplicaciones en la nube, ya que permite controlar qué recursos pueden ser accesibles desde Internet y cuáles se mantienen aislados.

En resumen, esta práctica servirá como una guía paso a paso para comprender y utilizar Amazon VPC de manera efectiva, lo que te permitirá crear una infraestructura de red sólida en la nube, manteniendo un alto nivel de seguridad y escalabilidad para tus aplicaciones y servicios alojados en AWS.

Requerimientos:

- Disponer de acceso a los recursos de AWS a través de un *sandbox* de AWS Academy

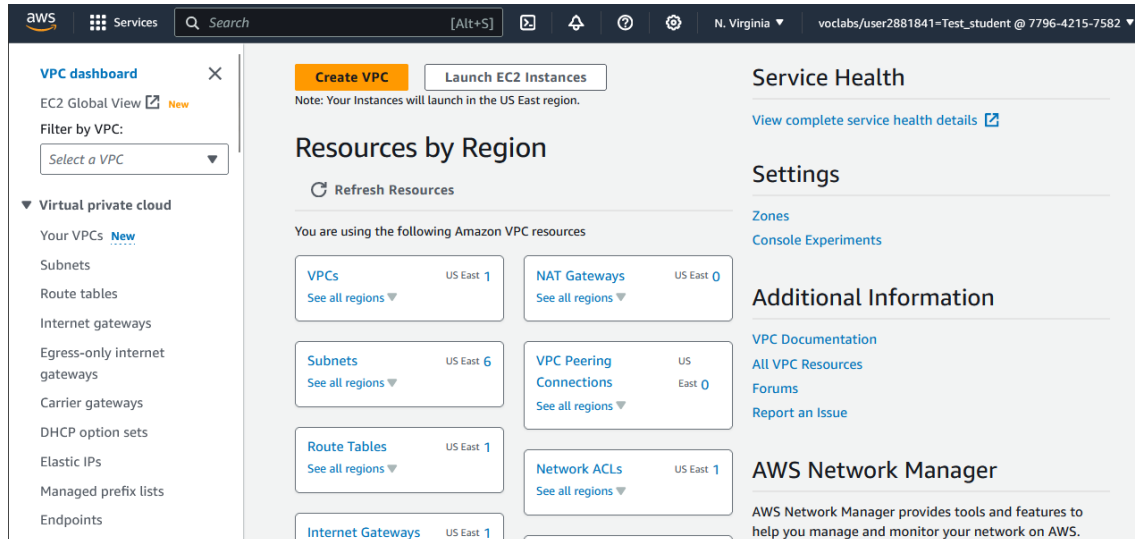
Arquitectura propuesta:



Realización:

CONFIGURACIÓN DE LA INFRAESTRUCTURA DE RED

- 1) Para crear la infraestructura de red necesaria para esta práctica, accedemos a la consola del servicio de Amazon VPC y presionamos el botón **Create VPC**:



La creación de una nube privada virtual (VPC) puede realizarse mediante un asistente (*wizard*) o personalizándola desde el principio. En esta práctica se explicarán ambas formas, si bien se recomienda crearla desde el inicio, para conocer más en profundidad el proceso de la creación de una VPC.

En nuestro caso, crearemos una VPC que abarcará dos zonas de disponibilidad (AZs, *Availability Zones*), y en cada una de ellas crearemos una subred pública y una subred privada. Las subredes públicas tendrán acceso directo a Internet y además, todos los recursos que se desplieguen en ellas podrán ser, potencialmente, accedidos desde la Internet pública. Las subredes privadas, por el contrario, no tendrán acceso directo a Internet, aunque podría proporcionarse mediante un dispositivo que actúe a modo de NAT (*Network Address Translation*), como un Gateway NAT o una instancia NAT.

Creación de una Amazon VPC mediante el asistente

- 2) Una vez en la ventana de la consola de administración del servicio de Amazon VPC, en el apartado **Resources to create**, se selecciona la opción **VPC and more**. A continuación, parametrizamos nuestra VPC, indicando los siguientes valores:
 - **Name tag auto-generation**: Indicamos el valor *wizard* y nos aseguramos de que la casilla de verificación **Auto-generate** esté marcada.
 - **IPv4 CIDR block**: Indicamos el bloque CIDR de nuestra red, en esta práctica será *10.2.0.0/16*
 - **IPv6 CIDR block**: Marcaremos la opción *No IPv6 CIDR block* (aunque si se desea, se puede activar una pila de red *dualstack* para permitir tanto tráfico IPv4 como IPv6)
 - **Tenancy**: Indica la tenencia por defecto de nuestras instancias EC2 que se lancen en esta VPC. Seleccionaremos la opción *Default*

- **Number of Availability Zones (AZs):** Indicaremos el valor 2 y ampliaremos la opción **Customize AZs**, indicando como primera zona de disponibilidad *us-east-1a* y como segunda zona de disponibilidad *us-east-1b*
- **Number of public subnets:** Seleccionaremos el valor 2
- **Number of private subnets:** Seleccionaremos el valor 2
- **Customize subnets CIDR blocks:** Indicaremos los bloques CIDR de nuestras cuatro subredes:
 - **Public subnet CIDR block in us-east-1a:** 10.2.0.0/24
 - **Public subnet CIDR block in us-east-1b:** 10.2.1.0/24
 - **Private subnet CIDR block in us-east-1a:** 10.2.2.0/24
 - **Private subnet CIDR block in us-east-1b:** 10.2.3.0/24
- **NAT gateways:** Indicaremos el valor *None*. Los NAT Gateway son dispositivos completamente administrados, escalables horizontalmente y altamente disponibles en una zona de disponibilidad que permiten que los recursos desplegados en subredes privadas puedan acceder a Internet. En esta práctica, omitiremos su uso debido al alto coste económico que puede repercutir para la implementación de la práctica.
- **VPC endpoints:** Indicaremos el valor *None*.
- **DNS options:**
 - **Enable DNS hostnames:** Este atributo determina si las instancias lanzadas en nuestra VPC recibirán un nombre DNS de host público que se resuelva a su IP pública. Marcamos la casilla de verificación
 - **Enable DNS resolution:** Este atributo determina si se debe resolver las solicitudes DNS a través del servidor de Amazon en la VPC. Marcamos la casilla de verificación

Por último, presionamos el botón **Create VPC**, tras lo cual aparecerá una ventana de progresión en la que, automáticamente se habrán creado los siguientes elementos:

Create VPC workflow

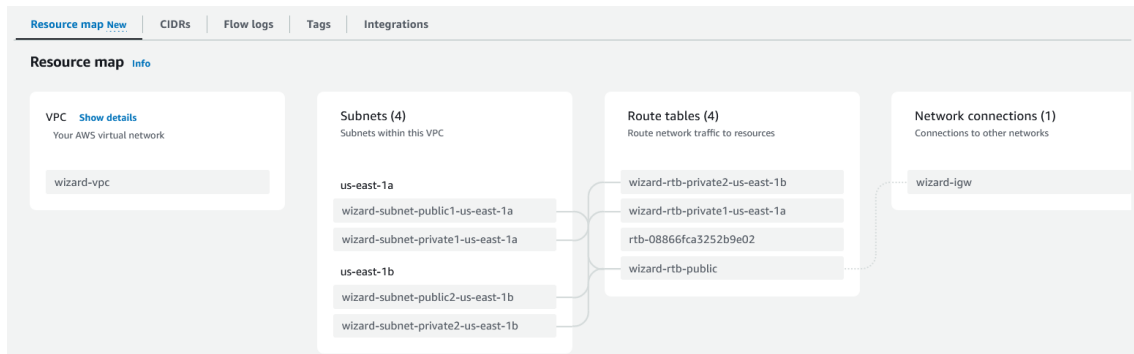
✓ Success

▼ Details

- ✓ Create VPC: [vpc-0610f158a5d262ad7](#)
- ✓ Enable DNS hostnames
- ✓ Enable DNS resolution
- ✓ Verifying VPC creation: [vpc-0610f158a5d262ad7](#)
- ✓ Create subnet: [subnet-04c2ea739127bf159](#)
- ✓ Create subnet: [subnet-0cafbf159aa5f7066](#)
- ✓ Create subnet: [subnet-008a937dddbab75e3](#)
- ✓ Create subnet: [subnet-0dabdc90c3d52177b](#)
- ✓ Create internet gateway: [igw-010ea83c09d3dedfd](#)
- ✓ Attach internet gateway to the VPC
- ✓ Create route table: [rtb-0a31dff556cdebe5f](#)
- ✓ Create route
- ✓ Associate route table
- ✓ Associate route table
- ✓ Create route table: [rtb-017f150fe9be8f305](#)
- ✓ Associate route table
- ✓ Create route table: [rtb-04043f890da14f9cd](#)
- ✓ Associate route table
- ✓ Verifying route table creation

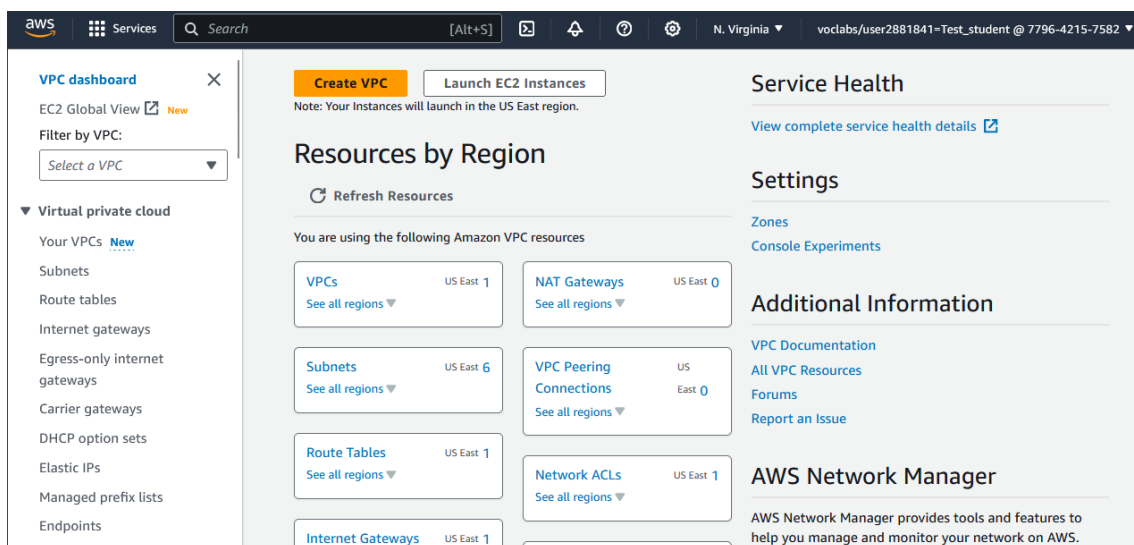
Todos los elementos y acciones realizadas mediante el asistente pueden crearse también de forma manual, lo cual nos da una mayor flexibilidad a la hora de personalizar nuestro entorno de red.

- 3) Si, a continuación, presionamos el botón **View VPC** podremos observar las propiedades de la VPC creada, así como su estructura mostrada mediante un mapa de recurso. En esta estructura, se puede observar cómo nuestra VPC está dividida en cuatro subredes (dos públicas y dos privadas), ubicadas en dos zonas de disponibilidad diferentes, además de haberse creado cuatro tablas de rutas (una tabla de rutas por defecto, una tabla de rutas para las subredes públicas de todas las zonas de disponibilidad, y una tabla de rutas para cada zona de disponibilidad para las subredes privadas) y un dispositivo Internet Gateway, que permite la conectividad con Internet desde las subredes públicas:



Creación de una Amazon VPC personalizada

- 4) Para crear una VPC personalizada debemos acceder de nuevo a la consola de Amazon VPC y presionar el botón **Create VPC**:



- 5) En la siguiente pantalla seleccionamos en el parámetro **Resource to create** la opción **VPC only** y completamos el formulario con los datos siguientes:

- **Name tag:** Introducimos el valor *custom-vpc*
- **IPv4 CIDR block:** Seleccionamos el valor *IPv4 CIDR manual input* y en el cuadro de texto **IPv4 CIDR** indicamos el valor *10.2.0.0/16*
- **IPv6 CIDR block:** Seleccionamos la opción *No IPv6 CIDR block*
- **Tenancy:** Seleccionamos el valor *Default*

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

custom-vpc

IPv4 CIDR block Info
☒ IPv4 CIDR manual input ☐ IPAM-allocated IPv4 CIDR block

IPv4 CIDR
10.2.0.0/16
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

Por último, presionamos el botón **Create VPC**.

- 6) A continuación, habilitaremos el soporte de resolución de DNS mediante los servidores de Amazon, así como la asignación de nombres de DNS para los recursos que despluguemos en la VPC. Para ello, desde la ventana de información de nuestra VPC, presionamos el botón **Actions** y seleccionamos la acción **Edit VPC settings**:

VPC > Your VPCs > vpc-003aa24db44e44353

vpc-003aa24db44e44353 / custom-vpc

Details Info

VPC ID	State	DNS hostnames
vpc-003aa24db44e44353	Available	Disabled
Tenancy	DHCP option set	Main route table
Default	dopt-072a9614b14b1e9d7	rtb-07cf295bcca1776e4
Default VPC	IPv4 CIDR	IPv6 pool
No	10.2.0.0/16	-
Network Address Usage metrics	Route 53 Resolver DNS Firewall rule groups	Owner ID
Disabled	Failed to load rule groups	779642157582

Actions

- Create flow log
- Edit VPC settings**
- Edit CIDRs
- Manage middlebox routes
- Manage tags
- Delete VPC

- 7) En la siguiente ventana, dentro del grupo de opciones **DNS Settings**, marcamos las casillas de verificación correspondientes a las opciones **Enable DNS resolution** y **Enable DNS hostnames**. Presionamos el botón **Save**:

VPC > Your VPCs > vpc-003aa24db44e44353 > Edit VPC settings

Edit VPC settings [Info](#)

VPC details

VPC ID
vpc-003aa24db44e44353

Name
custom-vpc

DHCP settings

DHCP option set [Info](#)
dopt-072a9614b14b1e9d7

DNS settings

☒ [Enable DNS resolution](#) [Info](#)

☒ [Enable DNS hostnames](#) [Info](#)

- 8) A continuación, vamos a crear un **Internet Gateway**. Se trata de un dispositivo completamente administrado, altamente disponible y escalable horizontalmente que permitirá que nuestros recursos desplegados en subredes públicas puedan acceder a Internet y ser accedidos desde la Internet pública. Para ello, desde el menú lateral de la consola de Amazon VPC, buscamos el grupo de opciones **Virtual private cloud** y seleccionamos la opción **Internet gateways**. Desde aquí, presionamos el botón **Create internet gateway**:

aws Services Search [Alt+S] N. Virginia voclabs/user2881841=Test_student @ 7796-4215-7582

VPC dashboard EC2 Global View [New](#)

Filter by VPC:
[Select a VPC](#)

Virtual private cloud

- Your VPCs [New](#)
- Subnets
- Route tables
- Internet gateways**

Internet gateways (2) [Info](#)

[Search](#) [Refresh](#) [Actions](#) [Create internet gateway](#)

<input type="checkbox"/>	Name	Internet gateway ID	State	VPC ID
<input type="checkbox"/>	wizard-igw	igw-010ea83c09d3dedfd	Attached	vpc-0610f158a5d262ad7 v
<input type="checkbox"/>	-	igw-0e2ac298ed49da5f4	Attached	vpc-0e7f90de0523449f3

- 9) En la ventana siguiente, introducimos en el campo **Name tag** el valor **custom-igw** y presionamos el botón **Create internet gateway**:

VPC > Internet gateways > Create internet gateway

Create internet gateway [Info](#)

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.

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.

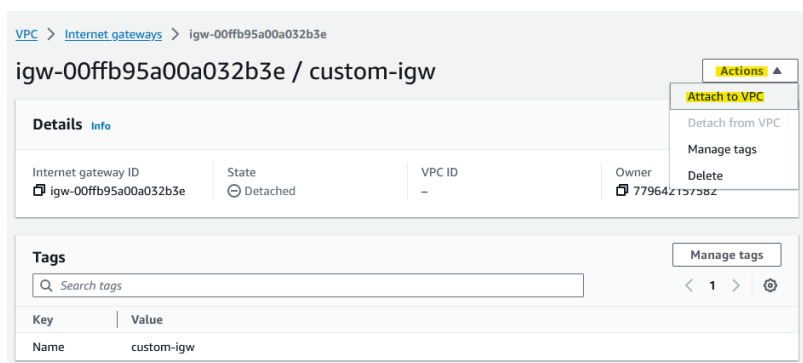
Key	Value - optional	
<input type="text" value="Name"/>	<input type="text" value="custom-igw"/>	Remove

[Add new tag](#)

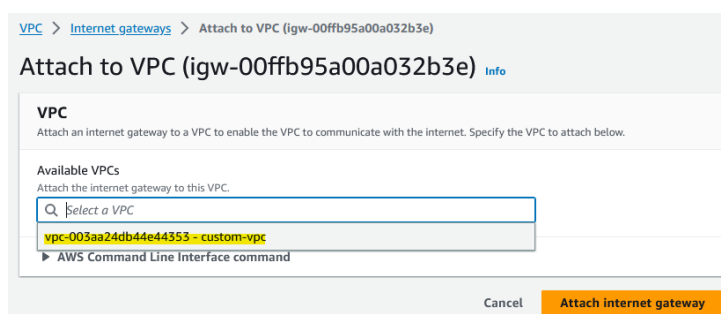
You can add 49 more tags.

[Cancel](#) [Create internet gateway](#)

- 10) Una vez creado el dispositivo virtual, lo asociamos a la VPC creada. Para ello, desde la pantalla de propiedades de nuestro Internet Gateway, presionamos el botón **Actions** y seleccionamos la opción **Attach to VPC**:



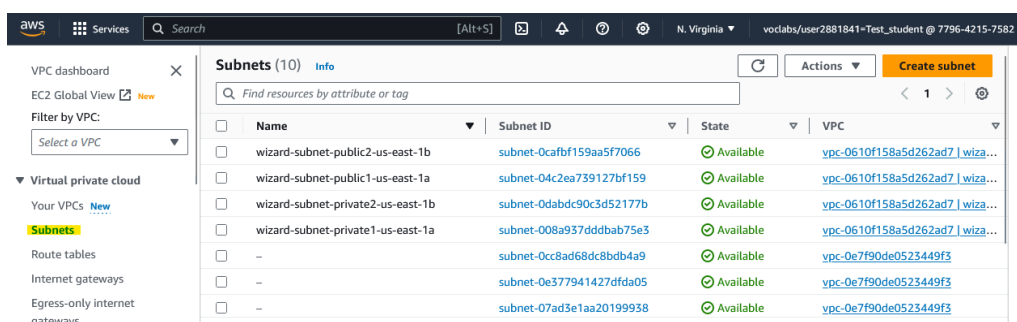
- 11) Seleccionamos a continuación la VPC etiquetada como *custom-vpc* dentro del campo **Available VPCs** y presionamos el botón **Attach internet gateway**:



- 12) Ha llegado el momento de definir las subredes de nuestra VPC. Tal y como se plantea en la práctica hay que crear cuatro subredes con los siguientes bloques CIDR y características:

Nombre de la subred	Zona de disponibilidad	Bloque CIDR
custom-subnet-public1-us-east-1a	us-east-1a	10.2.0.0/24
custom-subnet-public2-us-east-1b	us-east-1b	10.2.1.0/24
custom-subnet-private1-us-east-1a	us-east-1a	10.2.2.0/24
custom-subnet-private2-us-east-1b	us-east-1b	10.2.3.0/24

Para crear las subredes, accedemos al menú lateral de la consola de Amazon VPC y, dentro del grupo de opciones **Virtual private cloud** seleccionamos la opción **Subnets** y presionamos el botón **Create subnet**:



- 13) En la siguiente pantalla seleccionamos dentro del menú desplegable **VPC ID** la VPC etiquetada como *custom-vpc*

VPC > Subnets > Create subnet

Create subnet Info

VPC

VPC ID
Create subnets in this VPC.

vpc-003aa24db44e44353 (custom-vpc)

Associated VPC CIDRs

IPv4 CIDRs
10.2.0.0/16

A continuación, se completa añadiendo la información de cada una de las subredes, introduciendo en los campos **Subnet name**, **Availability zone** y **IPv4 subnet CIDR block** la información reflejada en la tabla del apartado 12:

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.

custom-subnet-public1-us-east-1a
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.

US East (N. Virginia) / us-east-1a

IPv4 VPC CIDR block Info
Choose the IPv4 VPC CIDR block to create a subnet in.

10.2.0.0/16

IPv4 subnet CIDR block
10.2.0.0/24 256 IPs

Una vez completada la información de la primera subred, presionamos el botón **Add new subnet** y procedemos añadiendo la información de las tres subredes restantes. Por último, presionamos el botón **Create subnet**. Tras esta operación nuestras subredes estarán creadas y podremos visualizarlas en la pantalla siguiente:

Subnets (4) Info

Find resources by attribute or tag

Subnet ID : subnet-0944979fe3133590b Subnet ID : subnet-039be73e19982ea05 Subnet ID : subnet-011cea0aae384abd Show more (+1)

Clear filters

	Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 ...	Avail...	Availability Zone
<input type="checkbox"/>	custom-subnet-public1-us-east-1a	subnet-0944979fe3133590b	Available	vpc-003aa...	10.2.0.0/24	-	251	us-east-1a
<input type="checkbox"/>	custom-subnet-public2-us-east-1b	subnet-039be73e19982ea05	Available	vpc-003aa...	10.2.1.0/24	-	251	us-east-1b
<input type="checkbox"/>	custom-subnet-private2-us-east-1b	subnet-0b4131ab76d3a2d06	Available	vpc-003aa...	10.2.3.0/24	-	251	us-east-1b
<input type="checkbox"/>	custom-subnet-private1-us-east-1a	subnet-011cea0aae384abd	Available	vpc-003aa...	10.2.2.0/24	-	251	us-east-1a

Nótese en la imagen anterior que en la columna **Available IP addresses** aparece el valor 251. Esto es debido a que la subred tiene una máscara /24 y por tanto 256 posibles IPs. Sin embargo, hay cinco de ellas que están reservadas: la primera para la IP de la subred, la segunda para el router interno de la VPC, la tercera para el servicio DNS de la subred, la cuarta queda reservada para un futuro, y la última para la dirección de *broadcast* de la subred. Por ejemplo, para la subred con el bloque CIDR 10.2.2.0/24 están reservadas las siguientes IPs:

- 10.2.2.0 → IP de subred
- 10.2.2.1 → IP router de la VPC
- 10.2.2.2 → IP servicio DNS de la VPC
- 10.2.2.3 → Reservada para uso futuro
- 10.2.2.255 → IP de broadcast

14) Para configurar que los recursos desplegados en subredes públicas tomen, por defecto, una IPv4 pública, desde la ventana anterior, seleccionamos la subred etiquetada como *custom-subnet-public1-us-east-1a* y presionamos el botón **Actions**, y activamos la opción **Edit subnet settings**:

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 ...
<input checked="" type="checkbox"/> custom-subnet-public1-us-east-1a	subnet-0944979fe3133590b	Available	vpc-003aa...	10.2.0.0/24	-
<input type="checkbox"/> custom-subnet-public2-us-east-1b	subnet-039be73e19982ea05	Available	vpc-003aa...	10.2.1.0/24	-
<input type="checkbox"/> custom-subnet-private2-us-east-1b	subnet-0b4131ab76d3a2d06	Available	vpc-003aa...	10.2.3.0/24	-
<input type="checkbox"/> custom-subnet-private1-us-east-1a	subnet-011cea0aae384abd	Available	vpc-003aa...	10.2.2.0/24	-

15) A continuación, en la siguiente ventana marcamos la casilla de verificación **Enable auto-assign public IPv4 address** y presionamos el botón **Save**:

Edit subnet settings

Subnet

Subnet ID: subnet-0944979fe3133590b
Name: custom-subnet-public1-us-east-1a

Auto-assign IP settings

Enable the auto-assign IP settings to automatically request a public IPv4 or IPv6 address for a new network interface in this subnet.

☒ **Enable auto-assign public IPv4 address**

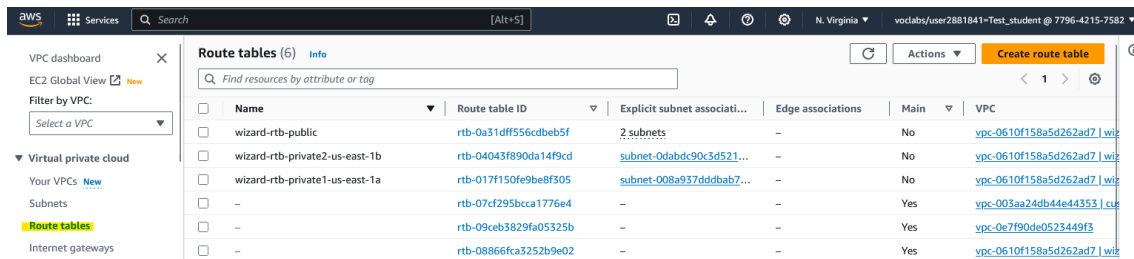
☐ Enable auto-assign customer-owned IPv4 address

Repetimos la operación de los pasos 14 y 15 para la subred etiquetada como *custom-subnet-public2-us-east-1b*.

16) Ahora sólo nos falta crear las tablas de rutas personalizadas para nuestra VPC. Siguiendo las buenas prácticas no utilizaremos la tabla de rutas por defecto (que se crea en el momento del despliegue de la VPC), sino que definiremos tres nuevas tablas de rutas:

- **custom-rtb-public.** Enrutará todo el tráfico no local hacia el dispositivo Internet Gateway. Se asignará a las subredes públicas en ambas zonas de disponibilidad
- **custom-rtb-private1-us-east-1a.** No permitirá el tráfico saliente hacia Internet. Se asignará a la subred privada de la zona de disponibilidad *us-east-1a*
- **custom-rtb-private1-us-east-1b.** No permitirá el tráfico saliente hacia Internet. Se asignará a la subred privada de la zona de disponibilidad *us-east-1b*

Para crear la tabla de rutas pública, accedemos a la consola de Amazon VPC y, desde el menú lateral, seleccionamos la opción **Route tables** dentro del grupo de opciones **Virtual private cloud**. A continuación, presionamos el botón **Create route table**:



- 17) Dentro de la pantalla siguiente, indicamos en el campo **Name** el valor *custom-rtb-public* y seleccionamos, en el menú desplegable **VPC**, la VPC etiquetada como *custom-vpc*. Por último presionamos el botón **Create 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.

vpc-003aa24db44e44353 (custom-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	Value - optional	
Name	custom-rtb-public	Remove

[Add new tag](#)

You can add 49 more tags.

Cancel
Create route table

- 18) Para la tabla de rutas que se asignará a las subredes públicas, hay que añadir una entrada que redirija todo el tráfico no local (*0.0.0.0/0*) hacia el dispositivo Internet Gateway. Para ello, desde la ventana de configuración de nuestra tabla de rutas, presionamos el botón **Edit routes**:

VPC > Route tables > rtb-028054a205eab340e

rtb-028054a205eab340e / custom-rtb-public

Actions ▾

Details info

Route table ID rtb-028054a205eab340e	Main No	Explicit subnet associations -	Edge associations -
VPC vpc-003aa24db44e44353 custom-vpc	Owner ID 779642157582		

Routes Subnet associations Edge associations Route propagation Tags

Routes (1)

Filter routes

Destination	Target	Status	Propagated
10.2.0.0/16	local	Active	No

- 19) Desde la siguiente ventana, presionamos el botón **Add route**, introducimos en el campo **Destination** el valor **0.0.0.0/0** y, desde el menú desplegable **Target** elegimos la opción **Internet Gateway** y, si todo va bien, deberá aparecer nuestro dispositivo que seleccionaremos.

VPC > Route tables > rtb-028054a205eab340e > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.2.0.0/16	local	Active	No
0.0.0.0/0	Internet Gateway	-	No

Remove

Add route

Cancel Preview **Save changes**

Por último, presionamos el botón **Save changes**, tras lo cual podremos comprobar que se ha añadido nuestra nueva entrada en la tabla de rutas:

VPC > Route tables > rtb-028054a205eab340e

rtb-028054a205eab340e / custom-rtb-public

Actions ▾

Details info

Route table ID rtb-028054a205eab340e	Main No	Explicit subnet associations -	Edge associations -
VPC vpc-003aa24db44e44353 custom-vpc	Owner ID 779642157582		

Routes Subnet associations Edge associations Route propagation Tags

Routes (2)

Filter routes

Destination	Target	Status	Propagated
0.0.0.0/0	igw-00fb95a00a032b3e	Active	No
10.2.0.0/16	local	Active	No

- 20) Ahora debemos asociar nuestra tabla de rutas con ambas subredes públicas. Para ello, desde la ventana anterior seleccionamos la pestaña **Subnet associations** y, dentro del apartado **Explicit subnet associations** presionamos el botón **Edit subnet associations**:

Routes Subnet associations Edge associations Route propagation Tags			
Explicit subnet associations (0) Edit subnet associations			
<input type="text" value="Find subnet association"/>			
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
No subnet associations You do not have any subnet associations.			
Subnets without explicit associations (4) Edit subnet associations			
The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:			
<input type="text" value="Find subnet association"/>			
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
custom-subnet-public1-us-east-1a	subnet-0944979fe3133590b	10.2.0.0/24	–
custom-subnet-public2-us-east-1b	subnet-039be73e19982ea05	10.2.1.0/24	–
custom-subnet-private2-us-east-1b	subnet-0b4131ab76d3a2d06	10.2.3.0/24	–
custom-subnet-private1-us-east-1a	subnet-011cea0aae384abd	10.2.2.0/24	–

- 21) A continuación, marcamos las casillas de verificación correspondientes a nuestras subredes públicas, tal y como se muestra en la siguiente figura y presionamos el botón **Save associations**:

VPC > Route tables > [rtb-028054a205eab340e](#) > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (2/4)

	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
<input checked="" type="checkbox"/>	custom-subnet-public1-us-east-1a	subnet-0944979fe3133590b	10.2.0.0/24	–
<input checked="" type="checkbox"/>	custom-subnet-public2-us-east-1b	subnet-039be73e19982ea05	10.2.1.0/24	–
<input type="checkbox"/>	custom-subnet-private2-us-east-1b	subnet-0b4131ab76d3a2d06	10.2.3.0/24	–
<input type="checkbox"/>	custom-subnet-private1-us-east-1a	subnet-011cea0aae384abd	10.2.2.0/24	–

Selected subnets

Cancel **Save associations**

- 22) Para crear la primera tabla de rutas para la primera subred privada, procedemos de igual forma que en el apartado 17, tal y como se muestra en la figura:

VPC > Route tables > Create route table

Create route table

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

You can add 49 more tags.

- 23) A continuación, no añadiremos ninguna ruta adicional a la tabla de rutas, sino que procederemos a asignarla directamente a la subred etiquetada como *custom-subnet-private-1-us-east-1a*, tal y como se hizo en los apartados 20-21:

VPC > Route tables > rtb-08043711bfb76c8b > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/4)

Filter subnet associations

	Name	Subnet ID	IPv4 CIDR
<input type="checkbox"/>	custom-subnet-public1-us-east-1a	subnet-0944979fe3133590b	10.2.0.0/24
<input type="checkbox"/>	custom-subnet-public2-us-east-1b	subnet-039be73e19982ea05	10.2.1.0/24
<input type="checkbox"/>	custom-subnet-private2-us-east-1b	subnet-0b4131ab76d3a2d06	10.2.3.0/24
<input checked="" type="checkbox"/>	custom-subnet-private1-us-east-1a	subnet-011cea0aae384abd	10.2.2.0/24

Selected subnets

subnet-011cea0aae384abd / custom-subnet-private1-us-east-1a X

- 24) Por último, repetimos el mismo proceso para la segunda tabla de rutas privada:

VPC > Route tables > Create route table

Create route table

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.

custom-rtb-private2-us-east-1b

VPC
The VPC to use for this route table.

vpc-003aa24db44e4353 (custom-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 X **Value - optional** custom-rtb-private2-us-east-1b X Remove

Add new tag

You can add 49 more tags.

VPC > Route tables > rtb-0bbf826ac85c256b > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/4)

Filter subnet associations

	Name	Subnet ID	IPv4 CIDR
<input type="checkbox"/>	custom-subnet-public1-us-east-1a	subnet-0944979fe3133590b	10.2.0.0/24
<input type="checkbox"/>	custom-subnet-public2-us-east-1b	subnet-039be73e19982ea05	10.2.1.0/24
<input checked="" type="checkbox"/>	custom-subnet-private2-us-east-1b	subnet-0b4131ab76d3a2d06	10.2.3.0/24
<input type="checkbox"/>	custom-subnet-private1-us-east-1a	subnet-011cea0aae384abd	10.2.2.0/24

Selected subnets

subnet-0b4131ab76d3a2d06 / custom-subnet-private2-us-east-1b X

- 25) Tras todo el proceso, desde la opción **Route tables** del menú lateral podremos comprobar que nuestras tres tablas de rutas están creadas y asignadas a sus correspondientes subredes:

Name	Route table ID	Explicit subnet associations	Edge associations	Main
-	rtb-07cf295bcca1776e4	-	-	Yes
-	rtb-09ceb3829fa05325b	-	-	Yes
-	rtb-08866fca3252b9e02	-	-	Yes
custom-rtb-private1-us-east-1a	rtb-08043711bfbd76c8b	subnet-011cea0aae384abd / custom-subnet-private1-us-east-1a	-	No
custom-rtb-private2-us-east-1b	rtb-0bbf826aac85c256b	subnet-0b4131ab76d3a2d06 / custom-subnet-private2-us-east-1b	-	No
custom-rtb-public	rtb-028054a205eab340e	2 subnets	-	No
wizard-rtb-private1-us-east-1a	rtb-017f150fe9be8f305	subnet-008a937dddbab75e3 / wizard-subnet-private1-us-east-1a	-	No

LANZAMIENTO DE INSTANCIAS DE AMAZON EC2 EN LA VPC PERSONALIZADA

En este apartado, se lanzarán dos instancias EC2, una en la subred pública etiquetada como *custom-subnet-public1-us-east-1a* (10.2.0.0/24) y otra en la subred etiquetada como *custom-subnet-private2-us-east-1b*. A continuación, lanzaremos una conexión SSH contra la instancia ubicada en la subred pública, utilizando su IP pública y, desde dicha instancia, a su vez lanzaremos otra conexión SSH contra la IP privada de la segunda instancia ubicada en la subred privada. Para ello será necesario crear un grupo de seguridad que permita el tráfico desde la IP pública del aula hasta la instancia ubicada en la subred pública.

Para la conectividad entre ambas instancias EC2, será suficiente con utilizar el grupo de seguridad *default*. Este grupo de seguridad deniega todo el tráfico entrante a las instancias EC2 que no provenga de otra instancia que tenga asignado el mismo grupo de seguridad (*default*). La configuración de las instancias quedaría como sigue:

Nombre Instancia	Subred	Grupos de seguridad
Publica	custom-subnet-public1-us-east-1a	ssh-sg, default
Privada	custom-subnet-private2-us-east-1b	default

Grupo de seguridad	Entrada		Salida	
	Protocolo/Puerto	Origen	Protocolo/Puerto	Destino
ssh-sg	22 TCP	MyIP	Todo	Todo
default	Todo	ssh-sg	Todo	Todo

- 26) A continuación, crearemos en nuestra VPC el grupo de seguridad *ssh-sg*. Para ello, accederemos a la consola del servicio de Amazon VPC y, desde el menú lateral, en el grupo de opciones **Security**, activamos la opción **Security Groups** y presionamos el botón **Create security group**:

Name	Security group ID	Security group name	VPC ID	Description	Owner
-	sg-0103dfb04ab21c953	default	vpc-0610f158a5d262ad7	default VPC security gr...	779642157582
-	sg-045c513f52d709fcd	default	vpc-0e7f90de0523449f3	default VPC security gr...	779642157582
-	sg-0358b8e33ac5f10b2	default	vpc-003aa24db44e44353	default VPC security gr...	779642157582
mysql-sg	sg-01c19a85f0708dbd7	mysql-sg	vpc-0e7f90de0523449f3	Created by RDS manag...	779642157582

- 27) En la siguiente ventana, completamos con los datos del grupo de seguridad *ssh-sg*, tal y como se muestra en la siguiente figura y presionamos el botón **Create security group**:

VPC > Security Groups > Create security group

Create security group [Info](#)

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name [Info](#)

Name cannot be edited after creation.

Description [Info](#)

VPC [Info](#)

Inbound rules [Info](#)

Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info
SSH	TCP	22	My IP	Permite el trafico entrante desde mi IP

[Add rule](#)

Outbound rules [Info](#)

Type Info	Protocol Info	Port range Info	Destination Info	Description - optional Info
All traffic	All	All	Cust...	

[Add rule](#)

- 28) A continuación, procedemos a lanzar, en nuestra VPC, dos instancias de Amazon EC2 con sistema operativo Amazon Linux, asegurándonos que se lanzan en las subredes indicadas y tienen asignados los grupos de seguridad que se muestran en la tabla al inicio de este apartado. Las características de las instancias que se lanzan son las siguientes:

- **AMI:** Amazon Linux 2023 AMI
- **Arquitectura:** 64-bit (ARM)
- **Tipo de instancia:** *t4g.micro*
- **Nombre del par de claves:** *vockey*
- **Configuración de red:**
 - **VPC:** *custom-vpc*
 - **Subred:** *custom-subnet-public1-us-east-1a* o *custom-subnet-private2-us-east-1b*
 - **Grupos de seguridad:** *Select existing security group*

En la siguiente figura, se muestra cómo quedaría la configuración de red para la instancia ubicada en la subred pública:

VPC - required [Info](#)

vpc-003aa24db44e44353 (custom-vpc)
10.2.0.0/16

Subnet [Info](#)

subnet-0944979fe3133590b custom-subnet-public1-us-east-1a
VPC: vpc-003aa24db44e44353 Owner: 779642157582
Availability Zone: us-east-1a IP addresses available: 251 CIDR: 10.2.0.0/24

Auto-assign public IP [Info](#)

Enable

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☐ Create security group ☒ Select existing security group

Common security groups [Info](#)

Select security groups

ssh-sg sg-03eb8123203e10e1c X default sg-0358b8e33acf510b2 X
VPC: vpc-003aa24db44e44353 VPC: vpc-003aa24db44e44353

[Hide all selected](#)

Security groups that you add or remove here will be added to or removed from all your network interfaces.

La configuración de red para la instancia desplegada en la subred privada quedaría como se muestra a continuación:

VPC - required [Info](#)

vpc-003aa24db44e44353 (custom-vpc)
10.2.0.0/16

Subnet [Info](#)

subnet-0b4131ab76d3a2d06 custom-subnet-private2-us-east-1b
VPC: vpc-003aa24db44e44353 Owner: 779642157582
Availability Zone: us-east-1b IP addresses available: 251 CIDR: 10.2.3.0/24

Auto-assign public IP [Info](#)

Disable

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☐ Create security group ☒ Select existing security group

Common security groups [Info](#)

Select security groups

default sg-0358b8e33acf510b2 X
VPC: vpc-003aa24db44e44353

[Compare security group rules](#)

Security groups that you add or remove here will be added to or removed from all your network interfaces.

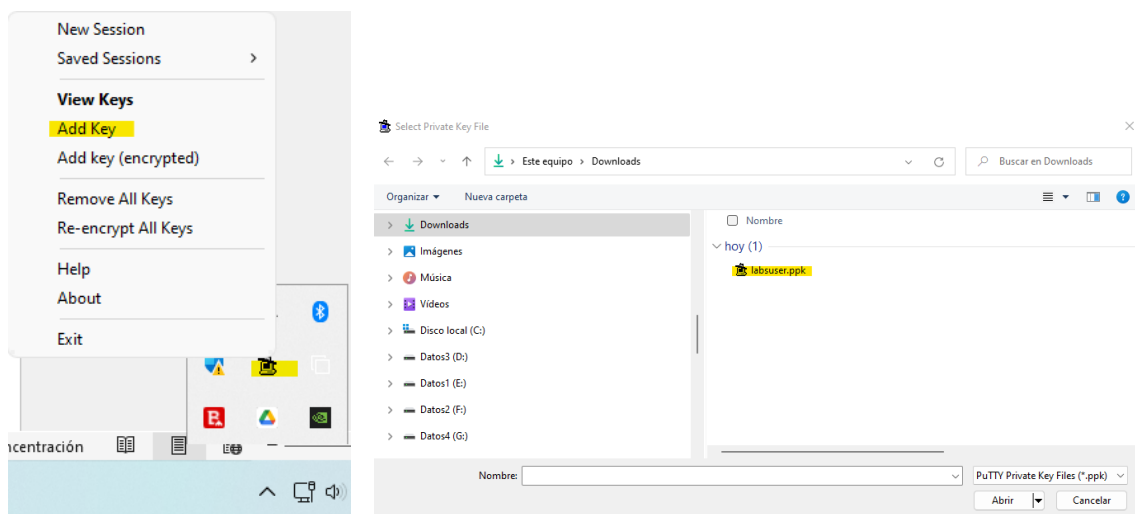
Para más información sobre el lanzamiento de instancias de Amazon EC2 en AWS Academy Learner Lab, consulta la siguiente práctica:

https://github.com/jose-emilio/aws-academy-fp-ec2/blob/main/Amazon_EC2_Linux.pdf

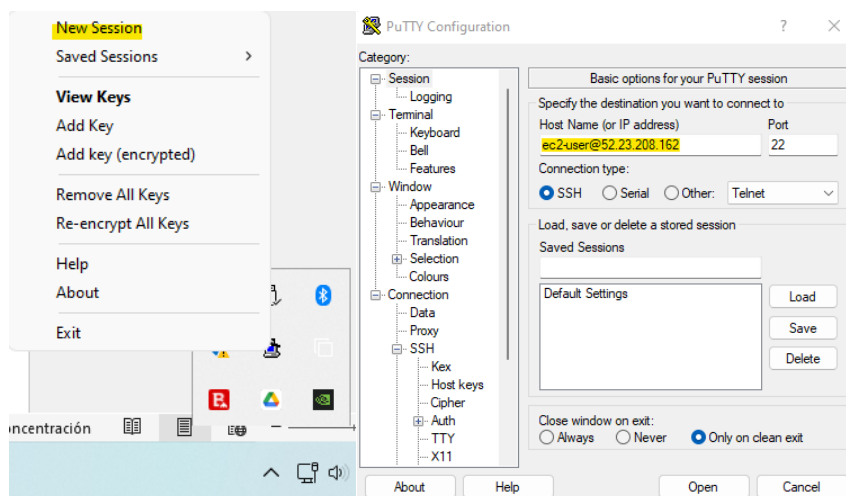
- 29) Tras el lanzamiento de las instancias, podremos comprobar desde la consola de Amazon EC2 que dichas instancias se han creado y que la instancia *Publica* tiene asignada una IP pública:

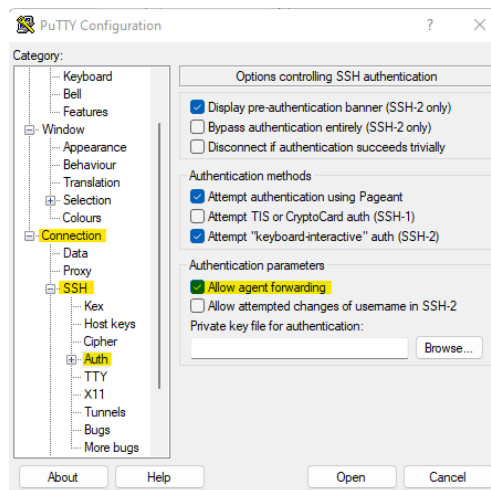
	Name	Instance ID	Instance state	Instance type	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	Private IP address
<input type="checkbox"/>	Publica	i-040ebf5f88e1a1cbb	Running	t4g.micro	us-east-1a	ec2-52-23-208-162.co...	52.23.208.162	-	10.2.0.89
<input type="checkbox"/>	Privada	i-0065a5fa5448cb19d	Running	t4g.micro	us-east-1b	-	-	-	10.2.3.9

- 30) A continuación, lanzaremos una conexión SSH contra la IP pública de la instancia *Publica* utilizando **Putty**. En este caso, para poder conectarnos a su vez a la instancia *Privada* es necesario habilitar *Agent Forwarding* en la configuración de la conexión, para lo cual utilizaremos la utilidad **Pageant** (incluida en Putty) tal y como se muestra en las figuras siguientes:



- 31) Para lanzar la sesión utilizando *Agent Forwarding*, creamos una nueva sesión desde **Pageant** de la forma siguiente, indicando la IP pública de nuestra instancia, y habilitando el agente desde el menú **Connection / SSH / Auth** con la opción indicada en las imágenes:





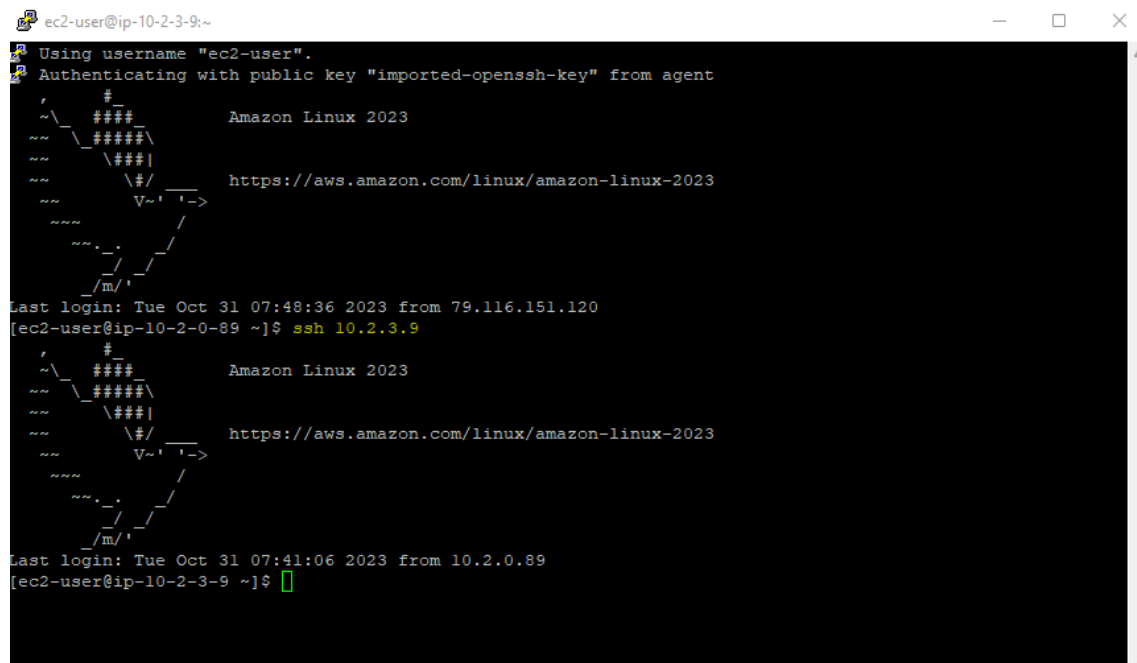
- 32) Una vez conectados a nuestra instancia *Publica* podemos conectarnos mediante SSH a la instancia *Privada* simplemente ejecutando la siguiente orden, sustituyendo el *placeholder* por la IP privada de nuestra instancia:

```
ssh ec2-user@<ip-privada>
```

O directamente:

```
ssh <ip-privada>
```

Podremos comprobar que hemos podido conectar a nuestra instancia privada a través de nuestra instancia pública:



- 33) Una forma alternativa de conectar directamente desde nuestra máquina en el aula a la instancia EC2 privada es, precisamente, utilizando reenvío de puertos local (*Local Port Forwarding*). Para ello,

```
ssh -i <clave-privada> -L 2022:<ip-privada>:22 ec2-user@<ip-publica>
```

[illegible]

```
ssh -i <clave-privada> ec2-user@localhost -p 2022
```

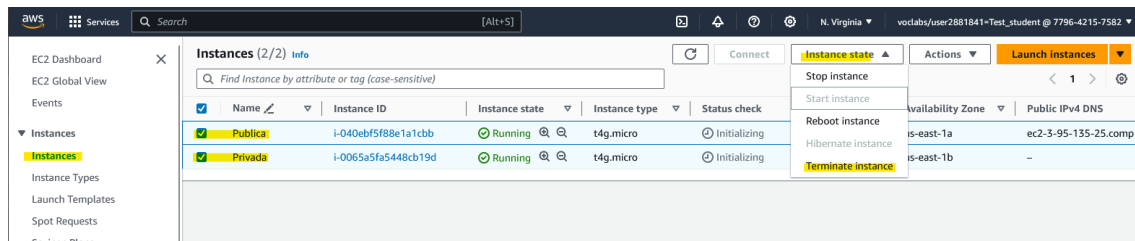
```
PS C:\Users\Jose> ssh -i .\Downloads\labsuser.pem ec2-user@localhost -p 2022
The authenticity of host '[localhost]:2022 ([::1]:2022)' can't be established.
ECDSA key fingerprint is SHA256:B/Yh3imcdLhkOoh2EEoQpfMVOLNvW7HTz8YvhByNk+A.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[localhost]:2022' (ECDSA) to the list of known hosts.
```

```
#
#####          Amazon Linux 2023
###
##### \
### |
### /#|         https://aws.amazon.com/linux/amazon-linux-2023
### Vm'->
      ^
     ^
    ^
   m/'
last login: Tue Oct 31 07:59:35 2023 from 10.2.0.89
[ec2-user@ip-10-2-3-9 ~]$
```

Limpieza de la Práctica:

Para terminar esta práctica y liberar los recursos creados, evitando así el consumo de créditos de AWS Academy Learner Labs, simplemente debemos dar los siguientes pasos:

- **Eliminar las instancias EC2.** Para ello, desde la consola de Amazon EC2 seleccionamos ambas instancias y, desde el menú **Instance state** elegimos la opción **Terminate instance**.



- **Eliminar las VPCs.** En realidad, los recursos creados mediante el servicio de Amazon VPC no representan ningún coste económico que repercuta contra los créditos del AWS Academy Learner Lab. Sin embargo, pueden eliminarse desde la consola del servicio de Amazon VPC, seleccionando cada VPC por separado y, desde el menú **Actions** activar la opción **Delete VPC**.

