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.

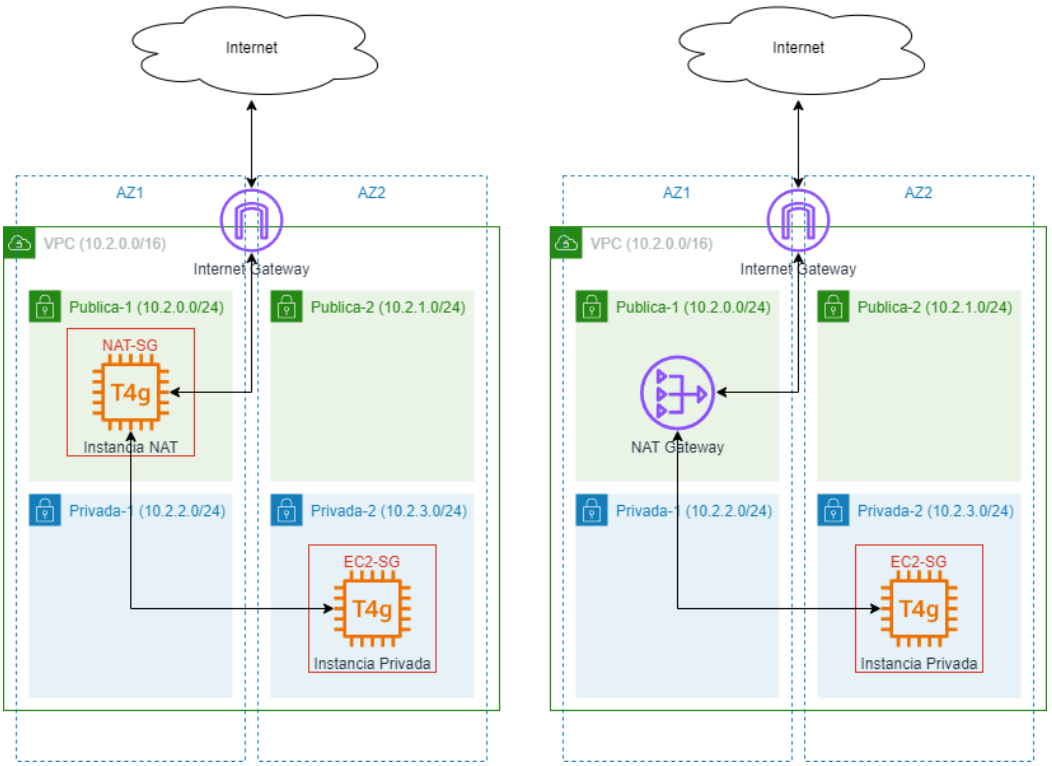
En esta práctica, abordaremos varios aspectos clave de Amazon VPC, desde la creación de una VPC desde cero hasta la configuración de subredes públicas y privadas. También se mostrará cómo lanzar instancias EC2 en una subred pública y, lo que es aún más importante, a establecer una comunicación segura 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:

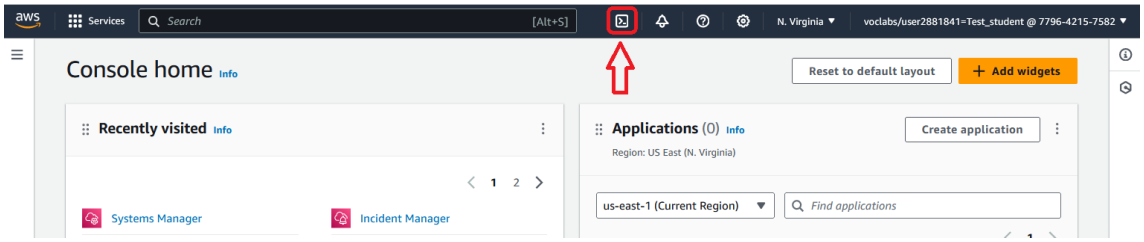


José Emilio Vera

Realización:

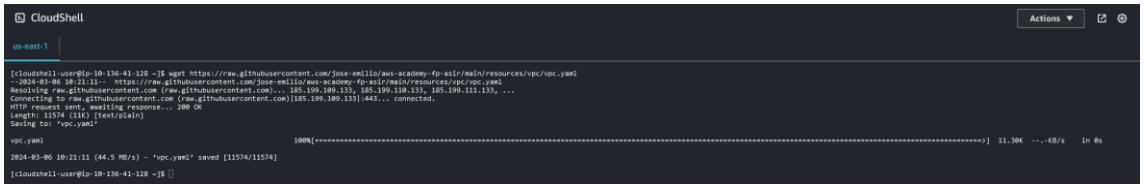
DESPLIEGUE DE LA INFRAESTRUCTURA DE LA PRÁCTICA

1) Para poder desplegar la arquitectura propuesta es necesario crear la infraestructura de red anterior, compuesta por una VPC con dos subredes privadas y dos subredes públicas en cada zona de disponibilidad. Para ello abrimos una sesión en AWS CloudShell, tal y como se muestra en la siguiente figura:



2) Una vez inicializada la sesión de AWS CloudShell, ejecutamos los siguientes comandos para descargar las plantillas de AWS CloudFormation necesarias para desplegar la infraestructura de la práctica:

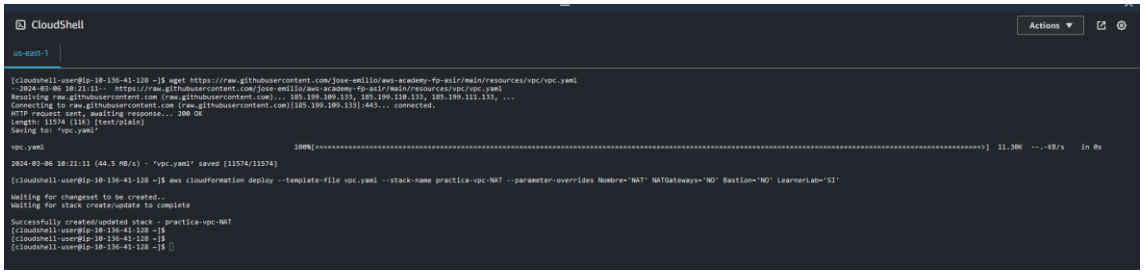
```
wget https://raw.githubusercontent.com/jose-emilio/aws-academy-fp-asir/main/resources/vpc/vpc.yaml
```



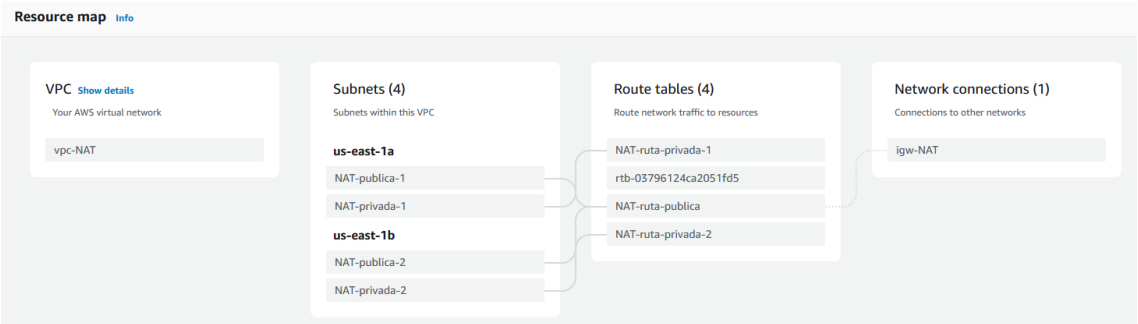
3) Una vez descargadas las plantillas anteriores, ejecutamos la siguiente instrucción para el despliegue:

```
aws cloudformation deploy --template-file vpc.yaml --stack-name practica-vpc-NAT --parameter-overrides
Nombre='NAT' NATGateways='NO' Bastion='NO' LearnerLab='SI'
```

Tras la ejecución de las instrucciones anteriores será necesario esperar unos pocos segundos hasta que la infraestructura de la práctica esté completamente preparada (cuando el servicio AWS CloudFormation devuelva el control del *prompt*).



La infraestructura de red desplegada mediante la plantilla de AWS CloudFormation queda reflejada en el siguiente esquema:

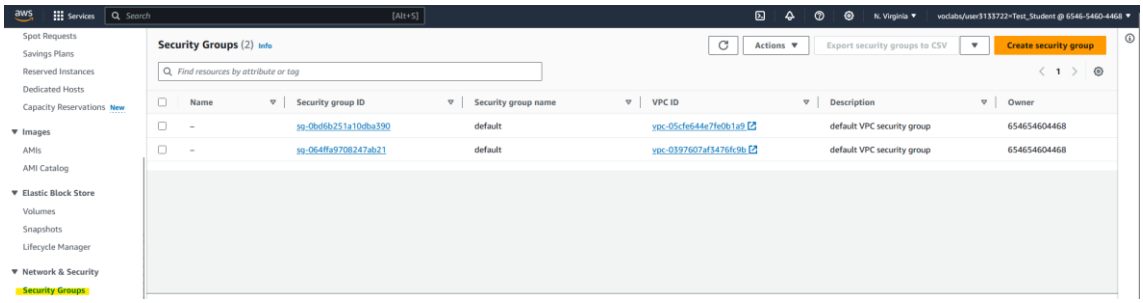


DESPLIEGUE DE UNA INSTANCIA NAT (EC2)

Una de las posibilidades a la hora de implementar una solución NAT dentro de una VPC es crear una instancia EC2 en una subred pública, por ejemplo, mediante Linux, y desplegar un servicio que realice funciones de NAT, como puede ser *iptables*.

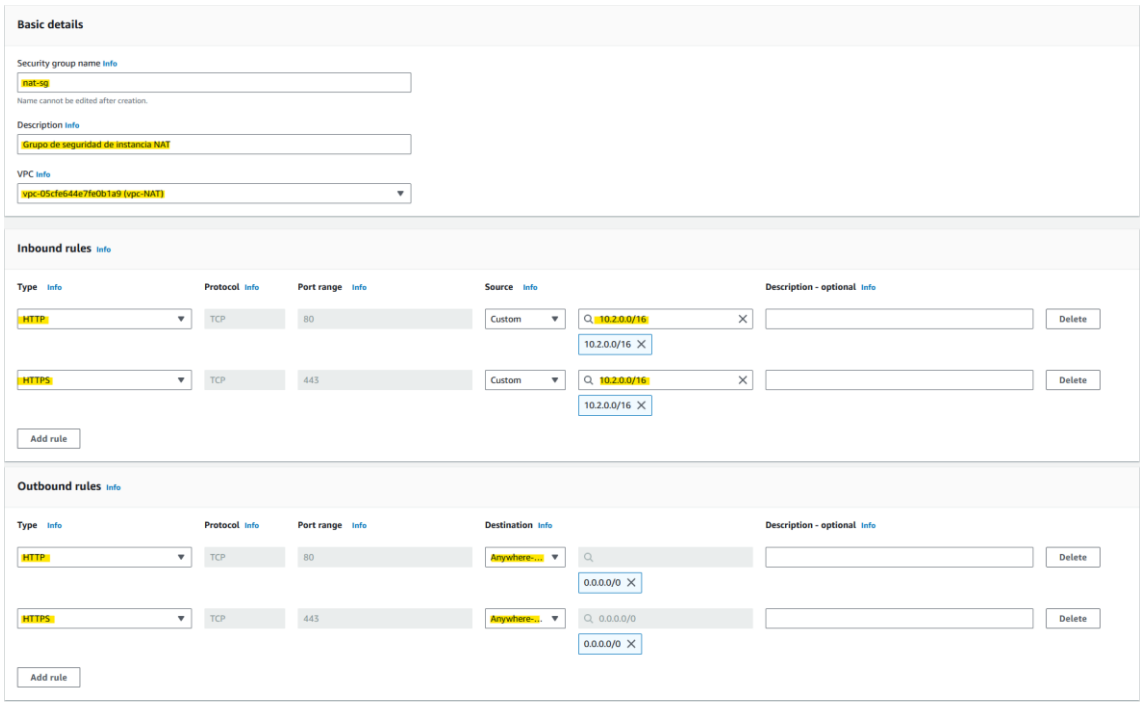
Las instancias NAT presentan la ventaja de su flexibilidad ya que, al tratarse de una instancia EC2, disponemos del control administrativo sobre ella, pudiéndola incluso utilizar para realizar otro tipo de funciones como, por ejemplo, actuar como host bastión para realizar conexiones vía SSH/RDP a instancias ubicadas en subredes privadas.

4) En primer lugar, crearemos un grupo de seguridad para nuestra instancia NAT. En este escenario vamos a restringir el tráfico a HTTP/HTTPS; por ello desde la consola de Amazon EC2 accedemos al menú lateral **Network & Security / Security Groups** y presionamos el botón **Create Security Group**:

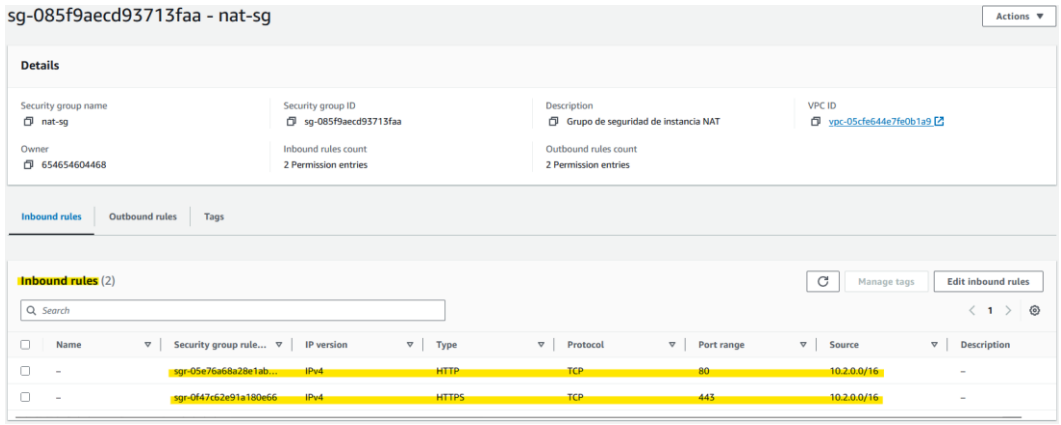


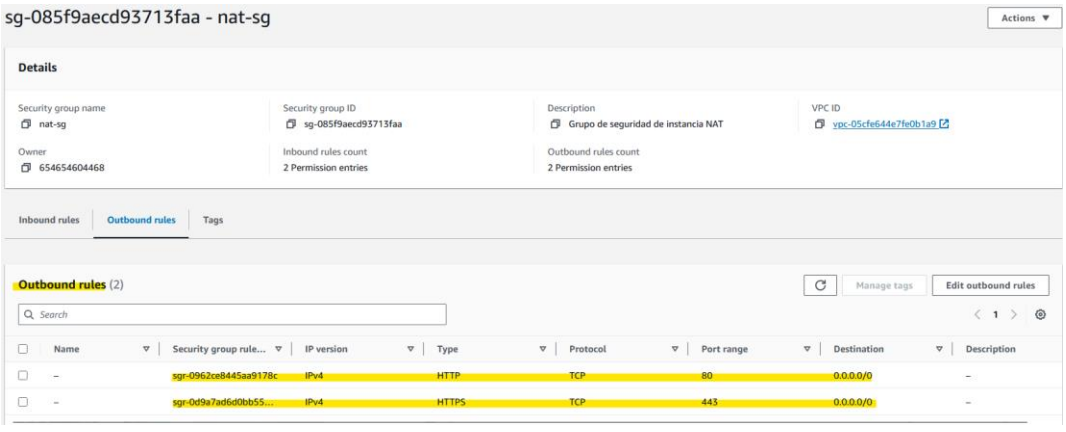
5) En la configuración del grupo de seguridad indicamos la siguiente configuración:

- En el apartado **Basic details**:
  - Introducimos en el campo **Security Group Name** el valor *nat-sg*
  - Introducimos en el campo **Description** el valor *Grupo de seguridad de instancia NAT*
  - Seleccionamos en el campo **VPC** la VPC etiquetada como *vpc-NAT*
- En el apartado **Inbound rules** añadimos una regla que permita el tráfico entrante desde el bloque CIDR de la VPC (10.2.0.0/16) por los puertos 80 TCP y 443 TCP
- En el apartado **Outbound rules**, eliminamos la regla por defecto y añadimos una regla que permita la salida hacia el exterior (0.0.0.0/0) hacia los puertos 80 TCP y 443 TCP



Por último, presionamos el botón **Create security group**. Tras ello, podremos comprobar la creación de nuestro grupo de seguridad y el tráfico permitido tanto de entrada como de salida.





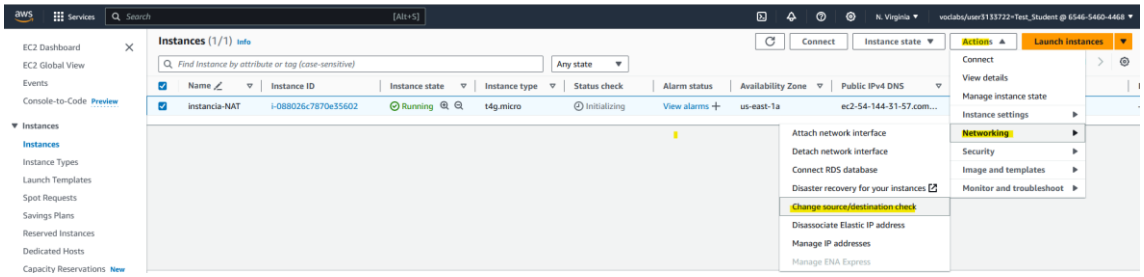
José Emilio Vera

6) Para desplegar una instancia NAT dentro de nuestra VPC, creamos una instancia EC2 con Linux con los siguientes parámetros (véase https://github.com/jose-emilio/aws-academy-fp-ec2/blob/main/Amazon_EC2_Linux.pdf):

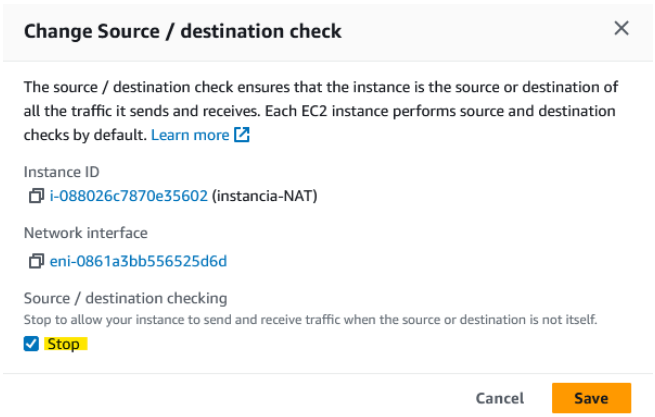
- **Name and tags**: Introducimos el valor *instancia-NAT* en el campo **Name**
- **Application and OS Images (AMI)**:
  - Desde la opción **QuickStart** elegimos una instancia *Amazon Linux 2023 AMI*
  - En el campo **Architecture**, elegimos *64-bit (ARM)*
- **Instance Type**: Seleccionamos la opción *t4g.micro*
- **Key Pair**: Seleccionamos el valor *Proceed without a key pair (Not Recommended)*
- **Network Settings**: Presionamos el botón **Edit** y configuramos las siguientes opciones
  - **VPC**: Seleccionamos la VPC etiquetada como *vpc-NAT*
  - **Subnet**: Seleccionamos la subred etiquetada como *NAT-publica-1*
  - **Firewall (security group)**: Seleccionamos la opción **Select existing security group** y elegimos de la lista desplegable la opción *nat-sg*
- **Advanced settings**:
  - **IAM instance profile**: Seleccionamos el valor *LabInstanceProfile*

Tras el lanzamiento, dispondremos de una instancia EC2 en estado disponible ubicada en una subred pública.

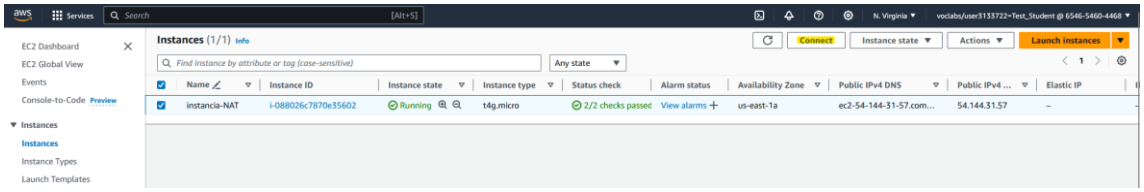
7) Por defecto, las instancias EC2 sólo envían o reciben el tráfico cuyo origen o destino sean ellas mismas. Sin embargo, una instancia NAT debe enviar paquetes en nombre de otros recursos en subredes privadas. Esto implica que hay que desactivar la comprobación de origen/destino que, por defecto, hacen las instancias EC2. Para ello, seleccionamos nuestra instancia EC2 desde la consola de Amazon EC2 y accedemos al botón **Actions / Networking / Change source/destination check**



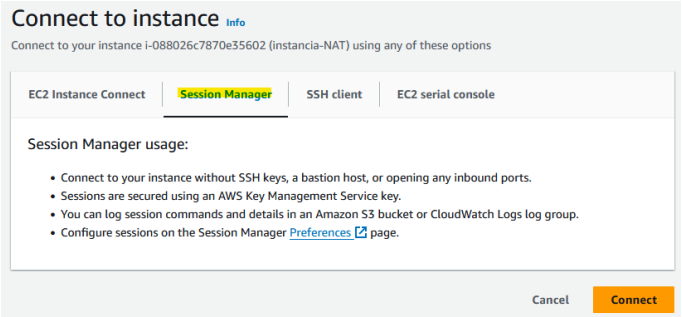
8) En la siguiente ventana, activamos la opción **Stop** en el campo **Source/destination check** y presionamos el botón **Save**:



9) A continuación, accedemos a la instancia EC2 mediante AWS Systems Manager Session Manager. Para ello, seleccionamos la instancia EC2 y presionamos el botón **Connect**:



Seleccionamos la pestaña **Session Manager** y presionamos el botón **Connect**:



10) Una vez abierta la sesión mediante el intérprete de órdenes, debemos configurar en este caso *iptables* para que funcione como servicio y añadir la regla NAT que permitirá que nuestra instancia EC2 enmascare con su propia IP pública, todo el tráfico que provenga desde subredes privadas. Para ello ejecutamos las siguientes órdenes

```
#Se instala el paquete iptables-services, necesario para administrar iptables como servicio Linux
sudo yum install iptables-services -y

#Se habilita para su lanzamiento automático
sudo systemctl enable iptables
sudo systemctl restart iptables

#Se configura el enrutamiento dentro de la instancia EC2
sudo bash -c "echo 'net.ipv4.ip_forward=1' > /etc/sysctl.d/ip-forwarding.conf"
sudo sysctl -p /etc/sysctl.d/ip-forwarding.conf
```

A continuación, debemos conocer el nombre lógico del interfaz de red de nuestra instancia EC2, para lo cual ejecutamos la siguiente orden:

```
netstat -i
```

De esta manera podremos comprobar el nombre adecuado; en este enunciado ha resultado ser *ens5*:

```
sh-5.2$ netstat -i
Kernel Interface table
Iface             MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
ens5             9001    23727      0      0 0          6268      0      0      0 BMRU
lo              65536       12      0      0 0            12      0      0      0 LRU
sh-5.2$
```

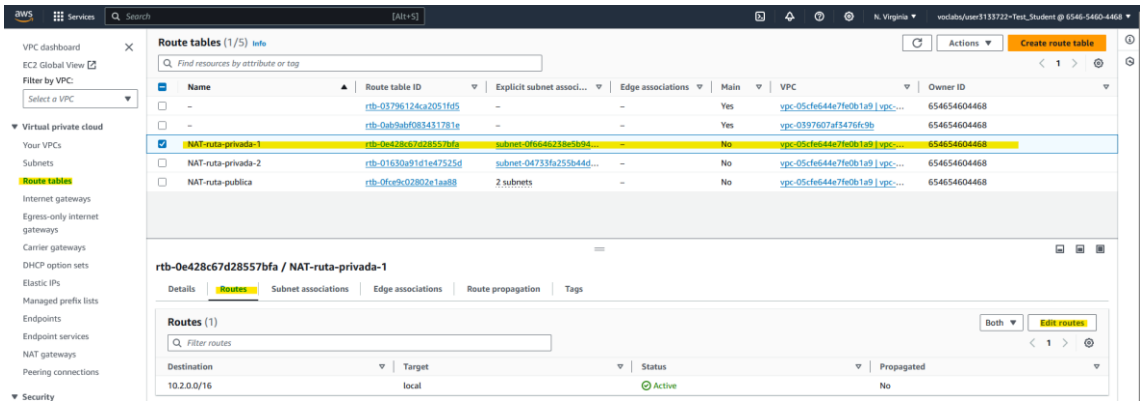
Por último, creamos las reglas necesarias para realizar NAT y salvamos los cambios en *iptables*:

```
#Se agrega la regla NAT de enmascaramiento
sudo iptables -t nat -A POSTROUTING -o ens5 -j MASQUERADE

#Se vacían las reglas de la cadena FORWARD
sudo iptables -F FORWARD

#Se salvan los cambios en el servicio iptables
sudo service iptables save
```

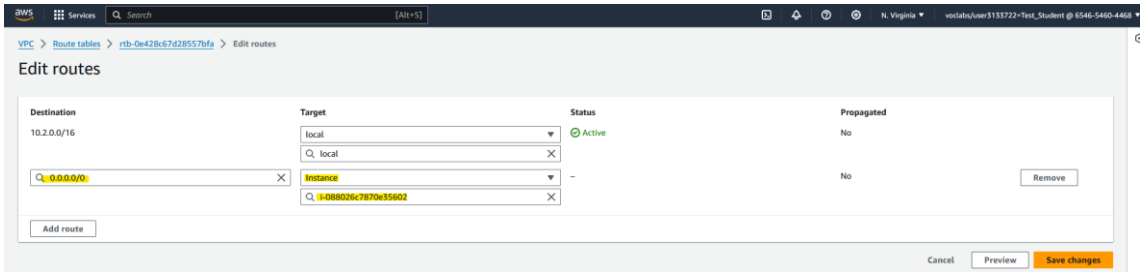
```
sh-5.2$ sudo iptables -t nat -A POSTROUTING -o ens5 -j MASQUERADE
sh-5.2$ sudo iptables -F FORWARD
sh-5.2$ sudo service iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables: [  OK  ]
sh-5.2$
```

Tras la configuración anterior, nuestra instancia NAT estará completamente operativa y disponible para enrutar tráfico enmascarado desde nuestras subredes privadas.

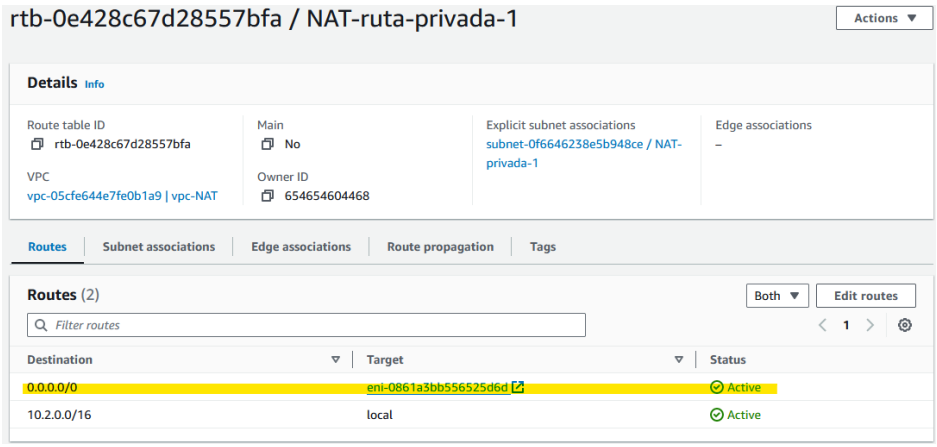
11) A pesar de estar configurada la instancia NAT, aún no es posible enrutar el tráfico ya que aún no hemos actualizado las tablas de rutas para las subredes privadas. En este caso, deberemos actualizar dichas tablas de rutas agregando una entrada para el tráfico no local (0.0.0.0/0) para encaminarlo a nuestra instancia NAT. Para ello, desde la consola de Amazon VPC, accedemos al menú lateral **Virtual Private Cloud / Route Tables** donde podremos comprobar las 3 tablas de rutas creadas para nuestra VPC. En este caso, seleccionamos la tabla de rutas etiquetada como *NAT-ruta-privada-1* y, desde la pestaña **Routes**, presionamos el botón **Edit routes**:



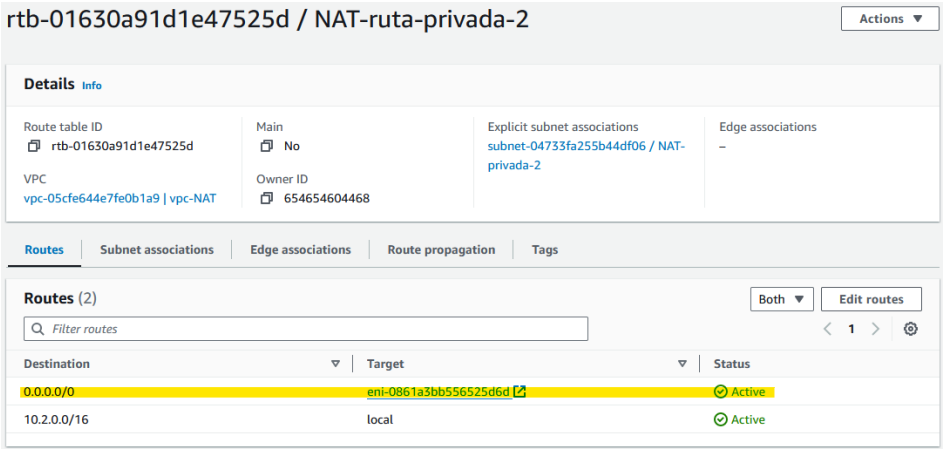
12) En la siguiente ventana, añadimos la entrada en la tabla de rutas para encaminar el tráfico no local hacia nuestra instancia NAT:



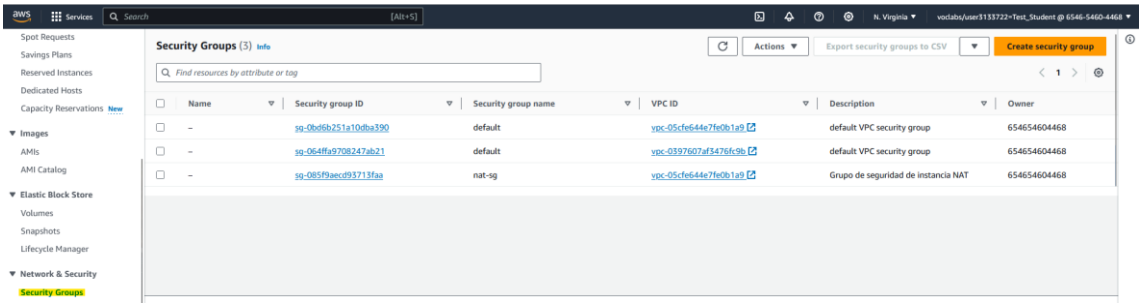
Por último, presionamos el botón **Save changes** y comprobamos los cambios realizados:



13) Repetimos los pasos **11)** y **12)** para la tabla de rutas etiquetada como *NAT-ruta-privada-2*:



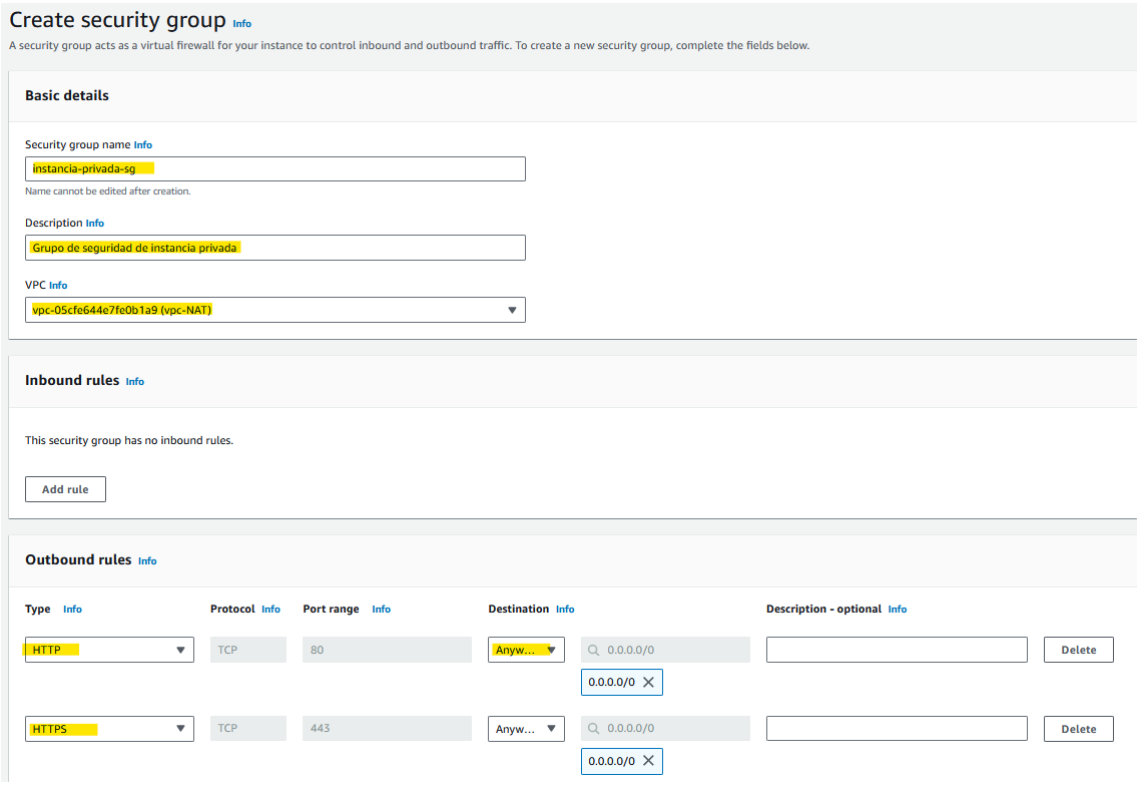
14) Previamente al lanzamiento de la instancia EC2 en una subred privada, crearemos un grupo de seguridad que permita únicamente el tráfico de salida HTTP o HTTPS hacia el exterior (0.0.0.0/0). Desde la consola de Amazon EC2 accedemos al menú lateral **Network & Security / Security Groups** y presionamos el botón **Create Security Group**:



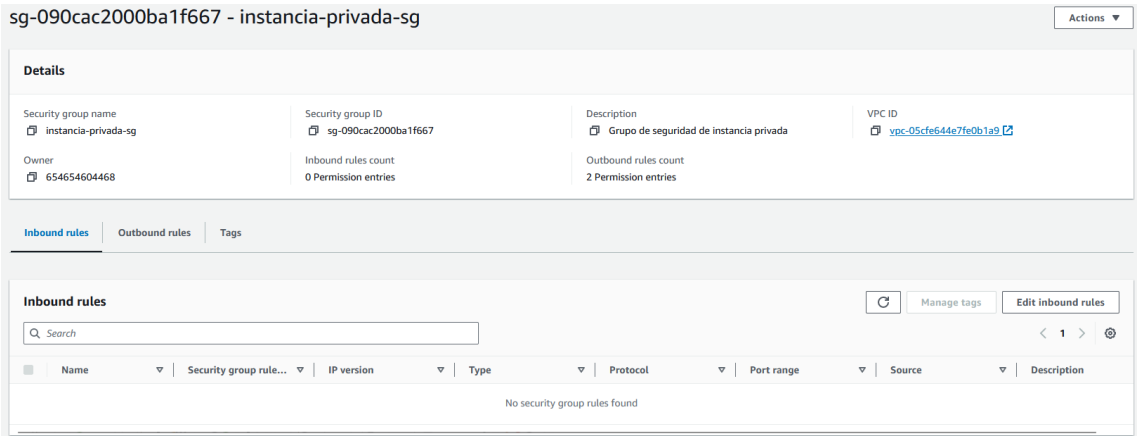
15) En la configuración del grupo de seguridad indicamos la siguiente configuración:

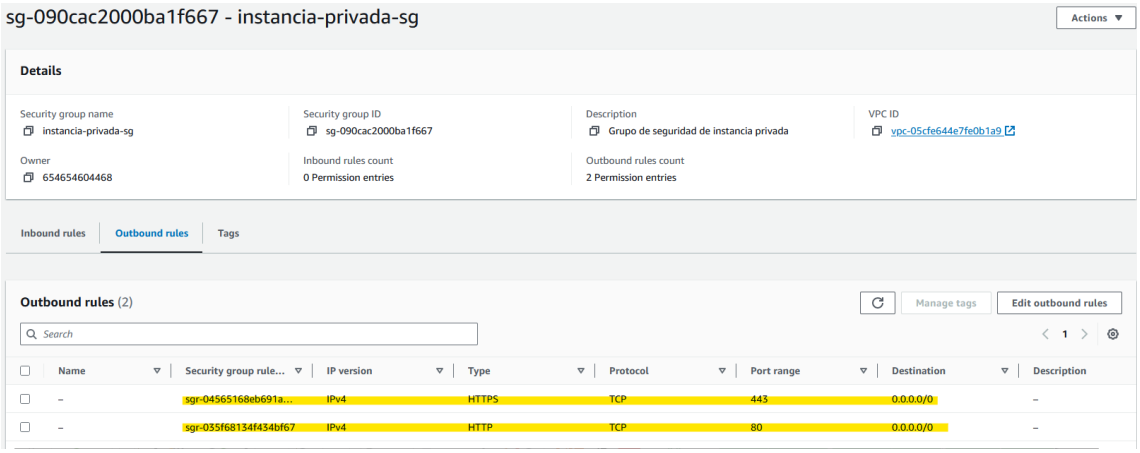
- En el apartado **Basic details**:
  - Introducimos en el campo **Security Group Name** el valor *instancia-privada-sg*
  - Introducimos en el campo **Description** el valor *Grupo de seguridad de instancia privada*
  - Seleccionamos en el campo **VPC** la VPC etiquetada como *vpc-NAT*

- En el apartado **Outbound rules**, eliminamos la regla por defecto y añadimos una regla que permita la salida hacia el exterior (0.0.0.0/0) hacia los puertos 80 TCP y 443 TCP



Por último, presionamos el botón **Create security group**, pudiendo verificar la correcta configuración:

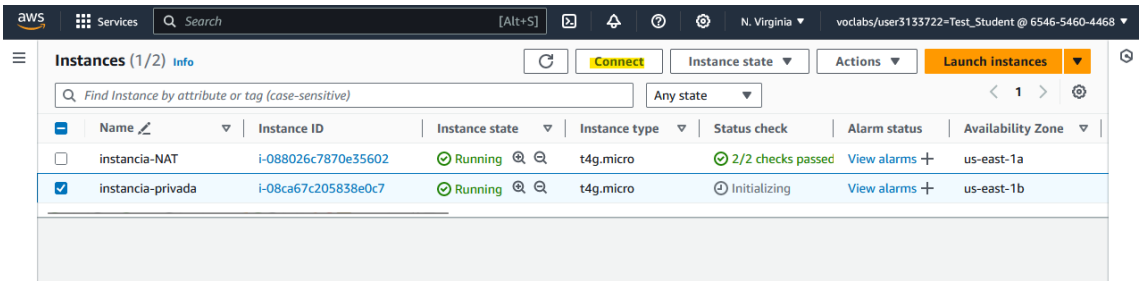




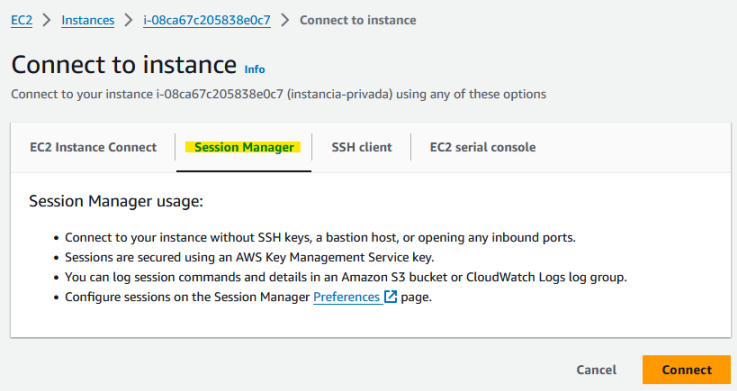
16) A continuación, comprobaremos la conectividad a Internet mediante HTTP o HTTPS lanzando una instancia EC2 en una subred privada. Creamos una instancia EC2 con Linux con los siguientes parámetros (véase https://github.com/jose-emilio/aws-academy-fp-ec2/blob/main/Amazon_EC2_Linux.pdf):

- **Name and tags**: Introducimos el valor *instancia-privada* en el campo **Name**
- **Application and OS Images (AMI)**:
  - Desde la opción **QuickStart** elegimos una instancia *Amazon Linux 2023 AMI*
  - En el campo **Architecture**, elegimos *64-bit (ARM)*
- **Instance Type**: Seleccionamos la opción *t4g.micro*
- **Key Pair**: Seleccionamos el valor *Proceed without a key pair (Not Recommended)*
- **Network Settings**: Presionamos el botón **Edit** y configuramos las siguientes opciones
  - **VPC**: Seleccionamos la VPC etiquetada como *vpc-NAT*
  - **Subnet**: Seleccionamos la subred etiquetada como *NAT-privada-2* (en realidad es indiferente que eliamos *NAT-privada-1*, pues hemos configurado ambas tablas de rutas para que encaminen el tráfico a la instancia NAT)
  - **Firewall (security group)**: Seleccionamos la opción **Select existing security group** y elegimos de la lista desplegable la opción *nat-sg*
- **Advanced settings**:
  - **IAM instance profile**: Seleccionamos el valor *LabInstanceProfile*

17) Tras el proceso anterior, podremos comprobar que la instancia en la subred privada se ha creado correctamente. Para comprobar su funcionamiento intentaremos conectarnos a ella mediante el servicio AWS Systems Manager Session Manager (en realidad con esta prueba ya sería suficiente ya que AWS SSM es un servicio que expone un punto de enlace HTTPS público, y si podemos acceder a nuestra instancia mediante este servicio es porque está enrutando el tráfico hacia la instancia NAT). Para ello, seleccionamos nuestra instancia privada y presionamos el botón **Connect**:



Seleccionamos la pestaña **Session Manager** y presionamos el botón **Connect**:



A continuación, se abrirá la sesión del intérprete de órdenes desde el cual podremos comprobar la navegación HTTP/HTTPS, simplemente ejecutando estas instrucciones:

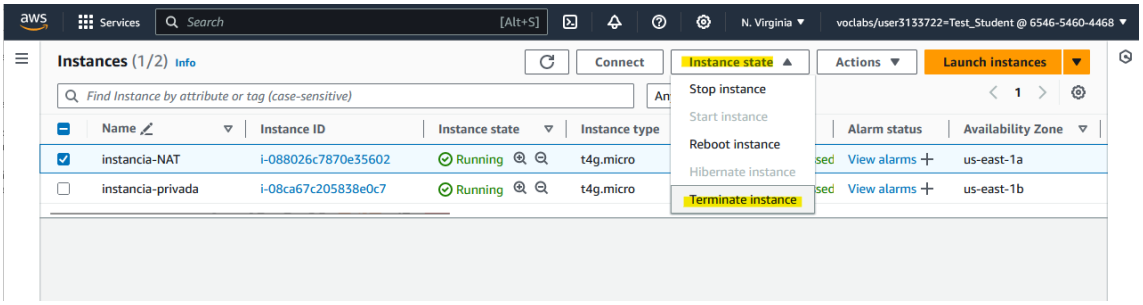
```
sudo yum install lynx -y
lynx https://aws.amazon.com
```

DESPLIEGUE DE UN GATEWAY NAT

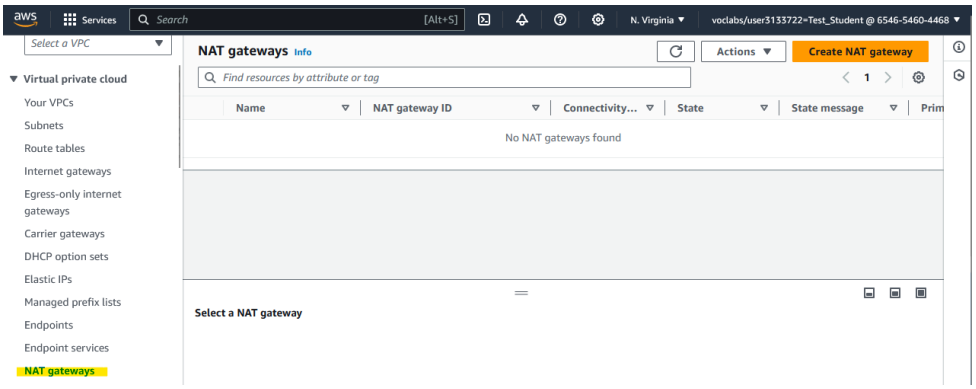
Las instancias NAT son recursos administrados por el usuario en la nube de AWS, por tanto, es necesario que el propio usuario implemente tanto para garantizar la alta disponibilidad como el escalado del servicio NAT.

Sin embargo, en AWS existe la posibilidad de desplegar un Gateway NAT, que es un dispositivo virtual completamente administrado por AWS, escalable horizontalmente y altamente disponible dentro de una zona de disponibilidad. En los siguientes apartados, veremos cómo podemos desplegar un Gateway NAT en nuestra VPC para dar servicio a la instancia privada creada.

18) En primer lugar, terminaremos la instancia NAT que creamos anteriormente, para ello, desde la consola de Amazon EC2, seleccionamos la instancia etiquetada como *instancia-NAT* y presionamos el botón **Instance state / Terminate instance**

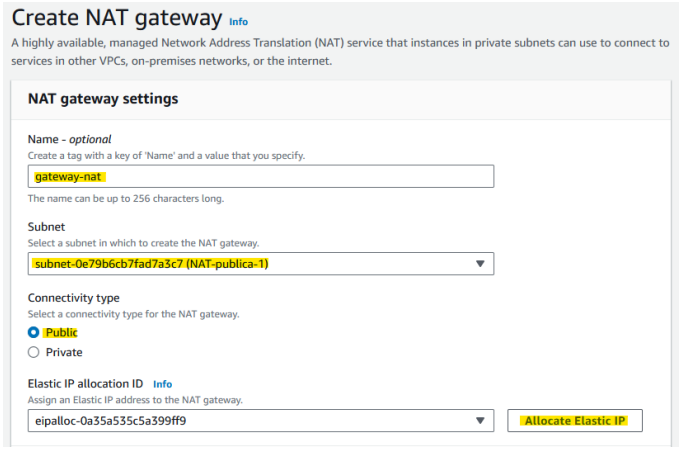


19) Para desplegar el Gateway NAT, accedemos a la consola de Amazon VPC y desde el apartado **Virtual Private Cloud / NAT Gateways** presionamos el botón **Create NAT gateway**:

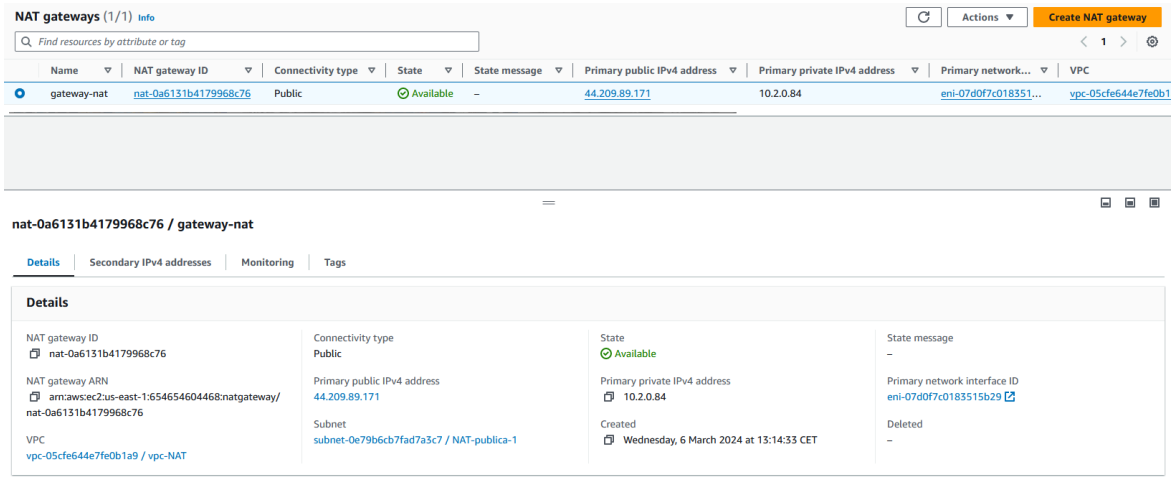


20) En la siguiente ventana, configuramos el Gateway NAT con los siguientes parámetros:

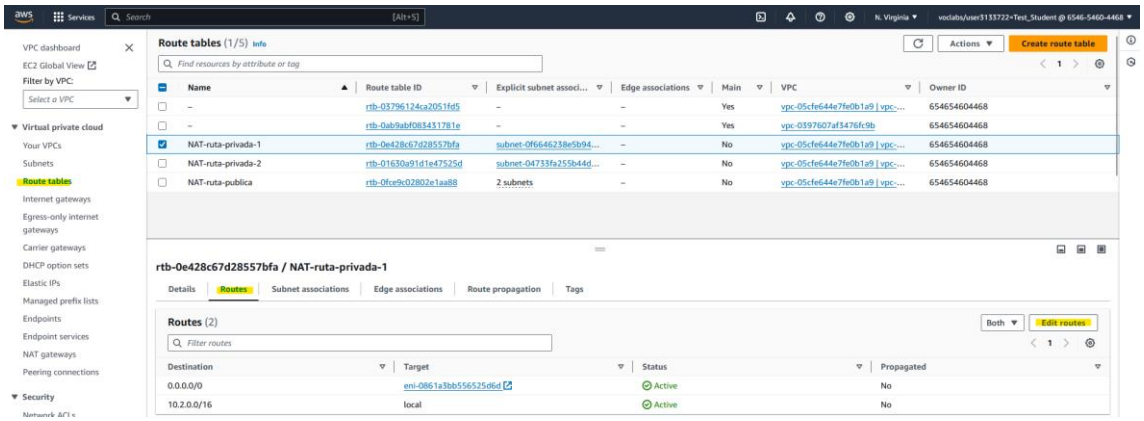
- **Name**: *gateway-nat*
- **Subnet**: Seleccionamos el valor de la subred etiquetada como *NAT-publica-1*
- **Connectivity type**: Seleccionamos el valor *Public*
- **Elastic IP allocation ID**: Presionamos el botón **Allocate Elastic IP**



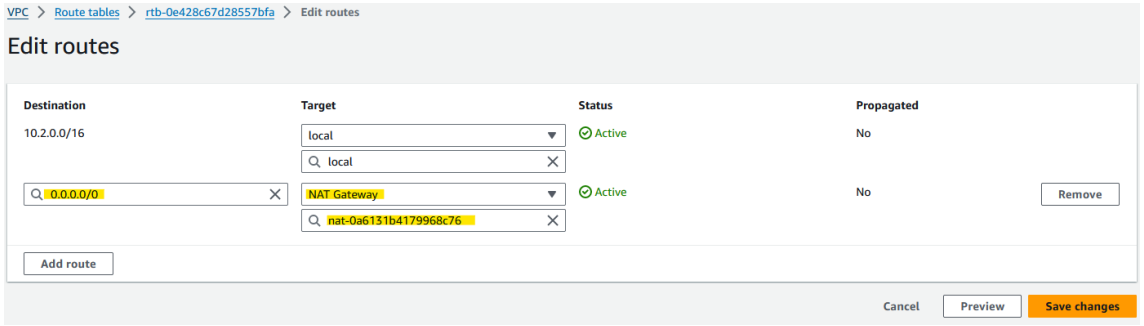
Por último, presionamos el botón **Create NAT gateway**. El dispositivo Gateway NAT tardará unos minutos en aprovisionarse. Procederemos al siguiente paso cuando ya esté disponible:



21) Una vez creado nuestro Gateway NAT, sólo resta modificar las tablas de rutas asignadas a las subredes privadas para que envíen el tráfico no local al Gateway NAT. Para ello, desde la consola de Amazon VPC accedemos al menú lateral **Virtual Private Cloud / Route Tables**, seleccionamos la tabla de rutas etiquetada como *NAT-ruta-privada-1*, seleccionamos la pestaña **Routes** y:presionamos el botón **Edit routes**:



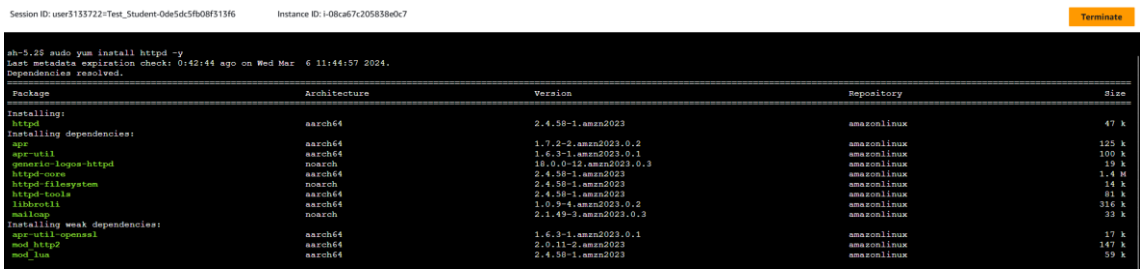
22) Por último, modificamos la entrada de tráfico no local (0.0.0.0/0) existente para redirigir el tráfico al dispositivo Gateway NAT:



Presionamos el botón **Save changes**

23) Repetimos los pasos **21)** y **22)** para la tabla de rutas etiquetada como *NAT-ruta-privada-2*.

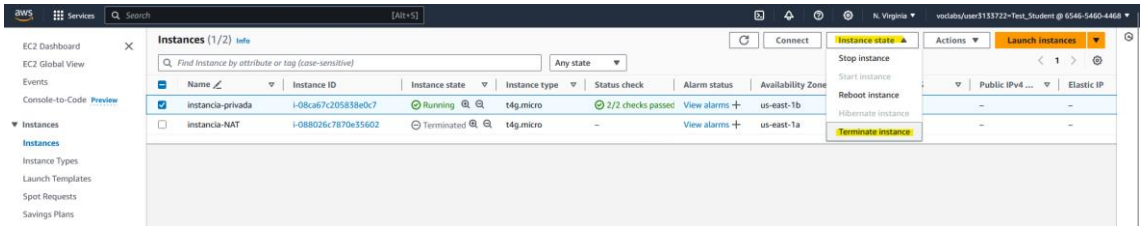
24) A continuación, podremos comprobar que podemos establecer conexión con nuestra instancia privada mediante AWS Systems Manager (ver paso **17)**):



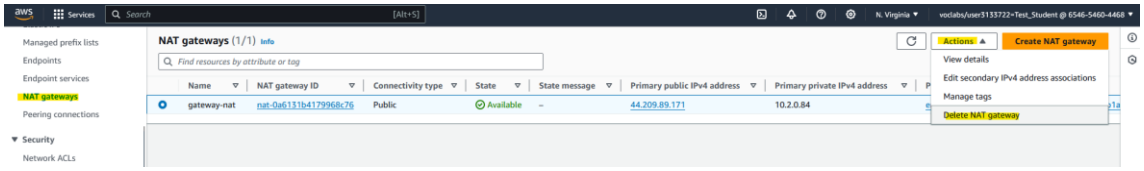
Limpieza de la Práctica:

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:

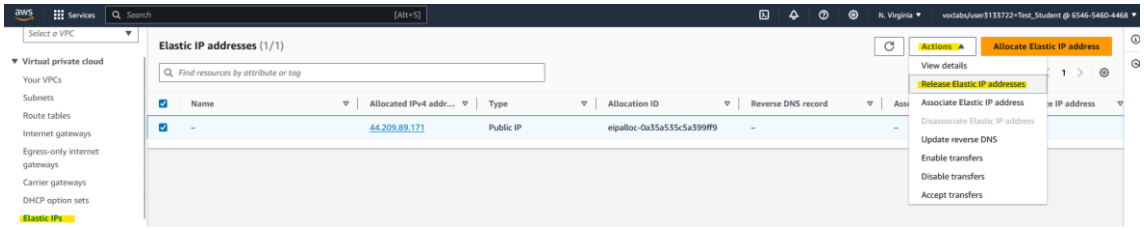
- **Eliminar las instancias EC2**. Para ello, desde la consola de Amazon EC2 seleccionamos la instancia privada y, desde el menú **Instance state** elegimos la opción **Terminate instance**.



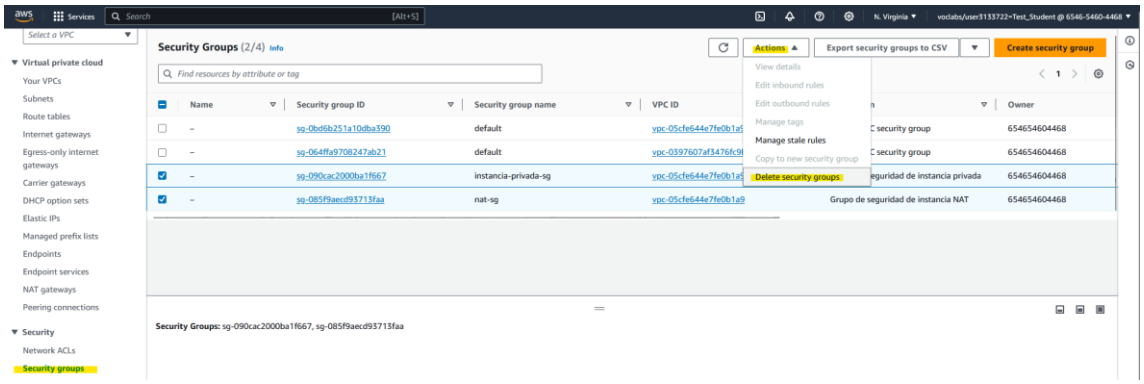
- **Eliminar el Gateway NAT**. Desde la consola de Amazon VPC, accedemos al menú lateral **Virtual Private Cloud / NAT Gateways**, seleccionamos el Gateway NAT creado y presionamos el botón **Actions / Delete NAT Gateway**:



- **Liberar la IP elástica**. Una vez hayamos comprobado que el Gateway NAT se encuentra en estado *Deleted*, desde la consola de Amazon VPC, accedemos al menú lateral **Virtual Private Cloud / Elastic IPs**, seleccionamos la IP creada por el Gateway NAT y presionamos el botón **Actions / Release Elastic IP**:



- **Eliminar los grupos de seguridad**. Desde la consola de Amazon VPC, accedemos al menú lateral **Security / Security groups**, seleccionamos los grupos de seguridad etiquetados como *instancia-privada-sg* y *nat-sg* y presionamos el botón **Actions / Delete security groups**:



- **Desaprovisionar infraestructura de red**. Desde la consola de AWS CloudShell (véase apartados **1)** y **2)**), ejecutar el siguiente comando:

```
aws cloudformation delete-stack --stack-name practica-vpc-NAT
```

La operación anterior, devolverá el prompt automáticamente, pero se habrá comenzado el decomisionamiento de la infraestructura de la VPC creada para esta práctica.