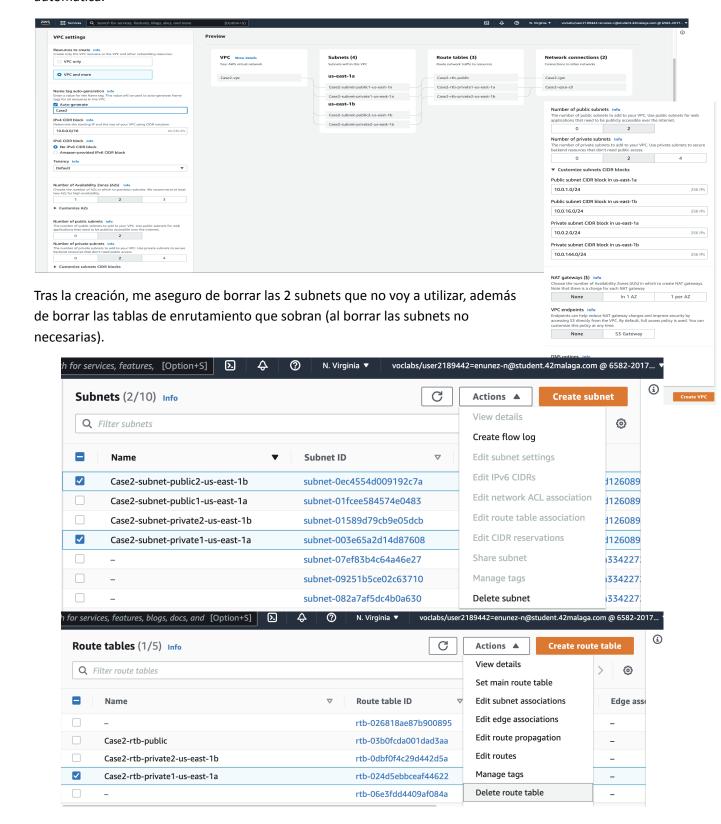
## Emilio Núñez Nieto <enunez-n> enunez-n@student.42malaga.com

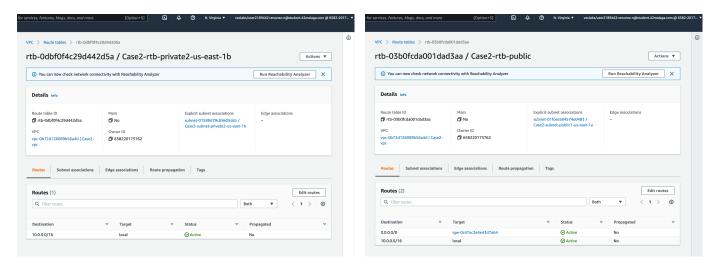
## **CASO PRÁCTICO 2**

Tarea 1 (50 puntos): Creación de la VPC y las subredes.

Para la **creación de la VPC** y las **subredes**, he hecho uso del wizard, el cual me da la opción de generar la VPC, con 2 AZ y 2 subnets, generando el IGW y las tablas de enrutamiento de forma automática.

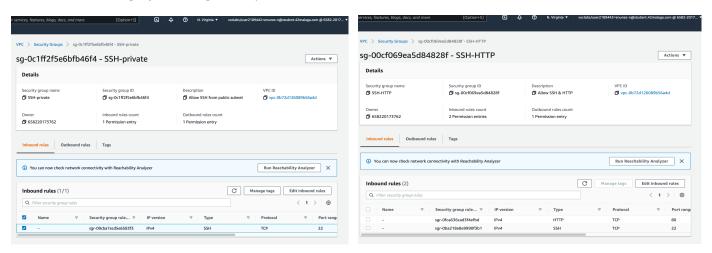


Así quedarían las tablas de enrutamiento de la subnet pública y de la privada:



Tarea 2 (50 puntos): Creación de las instancias EC2 y los Security Groups.

Defino ambos grupos de seguridad para las dos subredes:



El grupo para la subnet privada, con reglas de acceso interno SSH desde las IPs de la subnet pública, y el grupo para la subnet pública, con reglas de acceso SSH y HTTP abierto a todas las IPs entrantes.

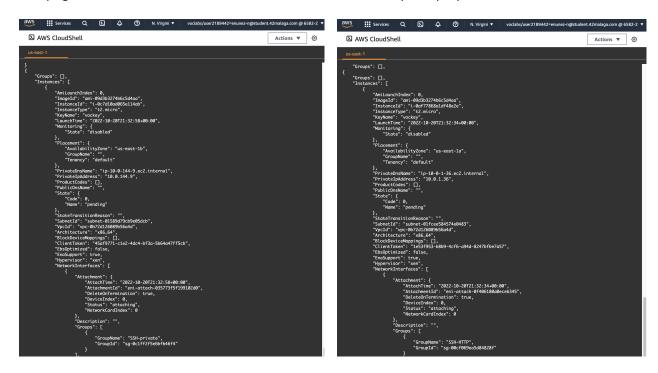
Una vez creado todo lo anterior, abro CLI para lanzar las dos instancias:

```
aws
        Services
                      a
                            \Sigma
                                   Д
                                          @
                                                 N. Virgi ▼
                                                              voclabs/user2189442=enunez-n@student.42malaga.com @ 65 ▼
 Actions ▼
                                                                                                           0
 us-east-1
#!/bin/bash
##EC2 instance in public subnet with webserver
aws ec2 run-instances
                  ami-09d3b3274b6c5d4aa \
                       t2.micro \
                         d sg-00cf069ea5d84828f \
                   subnet-01fcee584574e0483
                   file://userdata-web.sh \
               && \
##EC2 instance in private subnet
aws ec2 run-instances \
                  ami-09d3b3274b6c5d4aa \
                       t2.micro \
                  vockey \
                           sg-0c1ff2f5e6bfb46f4 \
                   subnet-01589d79cb9e05dcb
```

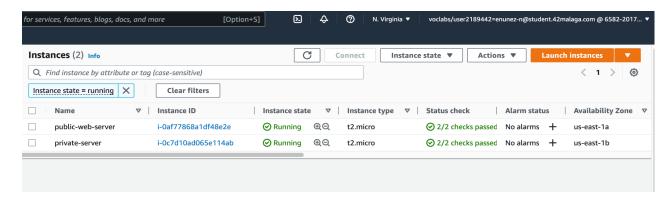
El script se compone de las siguientes partes:

- aws ec2 run-instances: comando para lanzar instancias ec2 al que le doy los siguientes flags con los detalles de cada instancia:
  - $\circ$  --image-id  $\rightarrow$  ID del AMI a montar.
  - --instance-type → el tipo de procesador de la instancia.
  - --count → cantidad de instancias a montar.
  - --key-name → nombre de la llave de encriptación para el acceso a la instancia.
  - --security-group-id → ID del grupo de seguridad asignado a la instancia.
  - o --subnet-id → ID de la subnet donde será montada la instancia.
  - --user-data → script (en este caso archivo con script) que será lanzado al inicio del primer montaje de la instancia.
  - --associate-public-ip-address → indica que se le asigne una IP pública a la instancia (en caso de necesitar ser accesible desde internet).
  - $\circ$  --tag-specifications  $\rightarrow$  flag con el que poder asignar diferentes tags a la instancia.

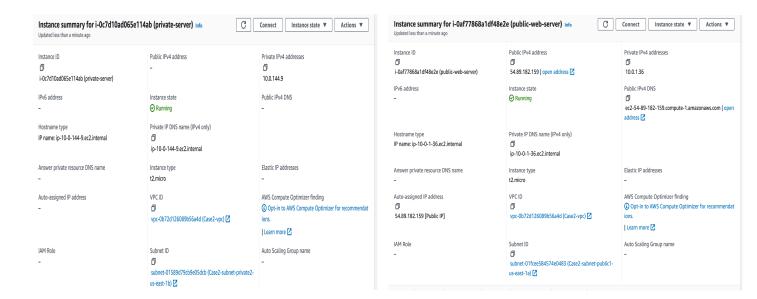
El script genera ambas instancias, cada una con las IDs de subnet y SG apropiadas.



Instancias en estado activo:



## Detalle de las dos instancias:



La instancia en la subnet pública tiene instalado un web server, por lo que podemos acceder a ella desde internet debido a que dispone de una IP pública:



## Hello AWS, this is a simple web server created by enunez-n!

Del mismo modo, podemos acceder mediante SSH a esta subnet pública con su IP a la que le hemos dado derechos mediante su SG:

Compruebo que se puede acceder a la subnet privada desde la subnet pública mediante SSH y no existe ese acceso desde IPs externas a la subnet pública:

```
aws chmod 400 labsuser.pem
→ aws ssh -i labsuser.pem ec2-user@54.89.182.159
The authenticity of host '54.89.182.159 (54.89.182.159)' can't be established.
ECDSA key fingerprint is SHA256:p1HodwQjn0rwwkuEc4nPvOB1Vtl1As0767WqEjOLJb0.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.89.182.159' (ECDSA) to the list of known hosts.
      __| __|_ )
_| ( / Amazon Linux 2 AMI
___|\__|
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-1-36 ~]$ ls
[ec2-user@ip-10-0-1-36 \sim]$ vim labsuser.pem
[ec2-user@ip-10-0-1-36 ~]$ chmod 400 labsuser.pem
[ec2-user@ip-10-0-1-36 ~]$ ssh -i labsuser.pem ec2-user@10.0.144.9
The authenticity of host 10.0.144.9 (10.0.144.9) can't be established.
ECDSA key fingerprint is SHA256:YD/dMUFCSrU5TAGLhYaw4PnsmnCmvM53UInVVewBX2I.
ECDSA key fingerprint is MD5:a6:53:0b:ef:be:95:89:91:ab:1d:21:75:bc:d4:08:28.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.144.9' (ECDSA) to the list of known hosts.
      __| __|_ )
_| ( / Amazon Linux 2 AMI
___|\__|
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-144-9 ~]$
```

```
[ec2-user@ip-10-0-144-9 ~]$ exit
logout
Connection to 10.0.144.9 closed.
[ec2-user@ip-10-0-1-36 ~]$ exit
logout
Connection to 54.89.182.159 closed.
→ aws ssh -i labsuser.pem ec2-user@10.0.144.9
ssh: connect to host 10.0.144.9 port 22: No route to host
→ aws
```