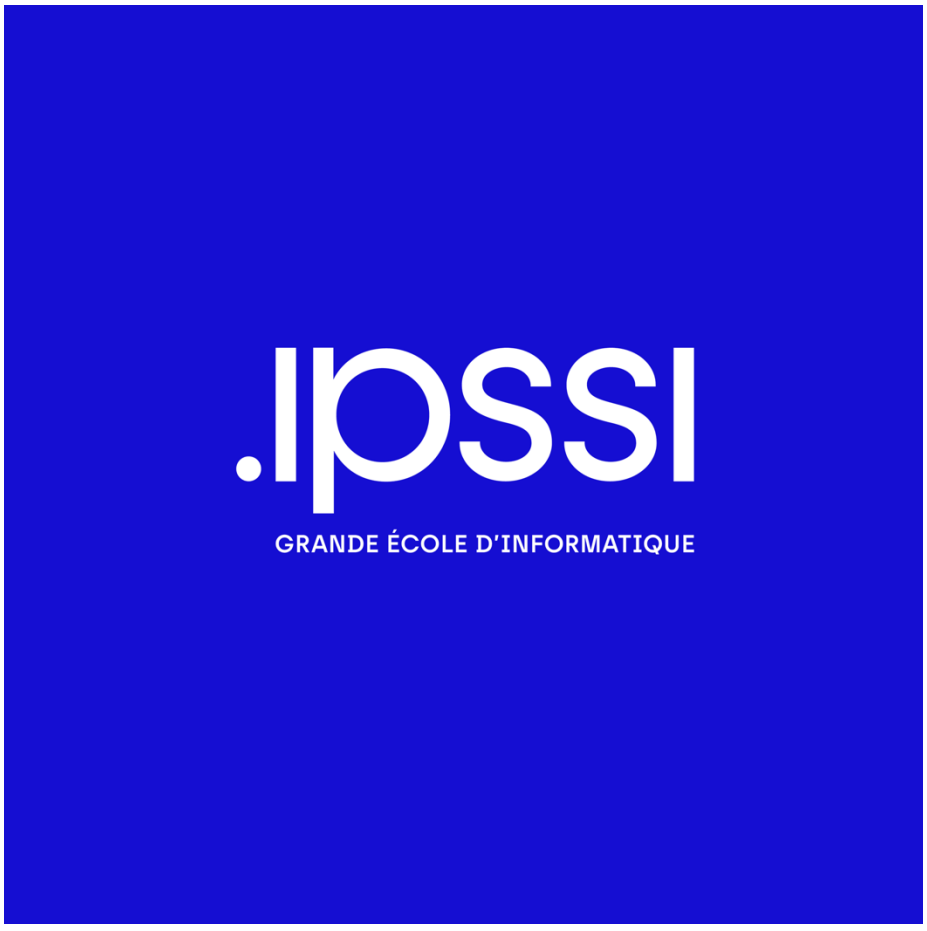
TP AWS

Amine MAHI

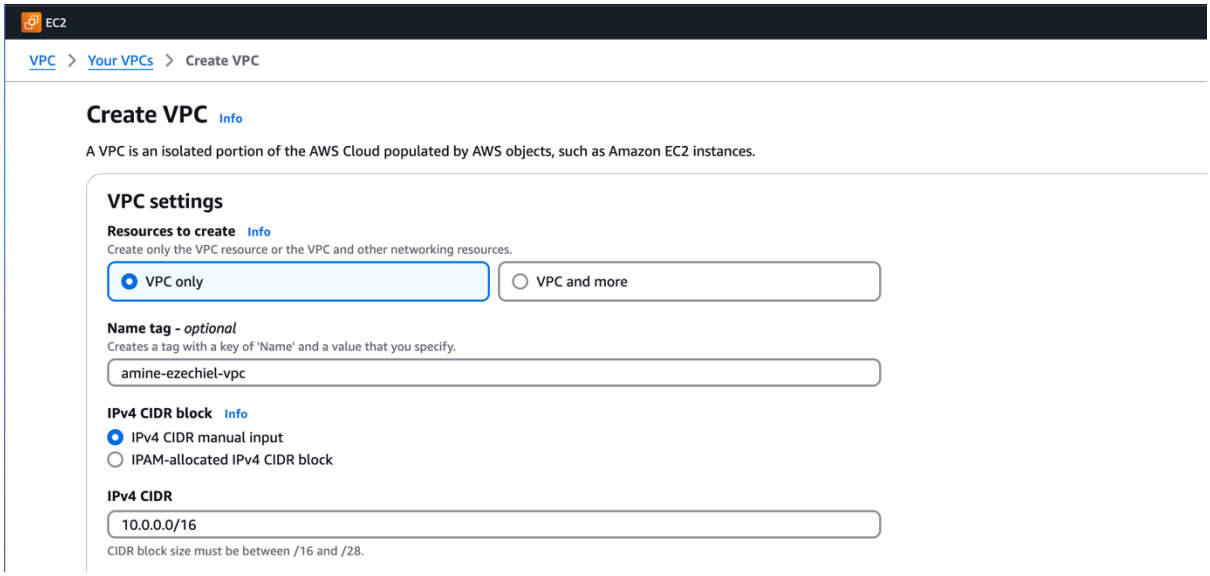
Ézéchiel BATCHI

BTC 26.1

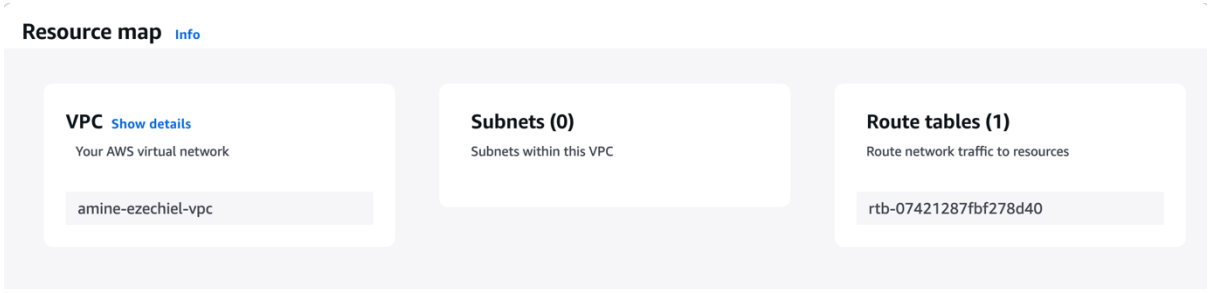


CONFIGURATION VPC

Nous commençons par crée notre VPC.



Pour l'instant nos VPC est vide



Sous-Réseaux

On crée alors nos sous réseaux en sachant qu'on aura 3 sous réseaux pour chaque instance. On choisit le CIDR block du VPC et en respectant les plages réseaux.

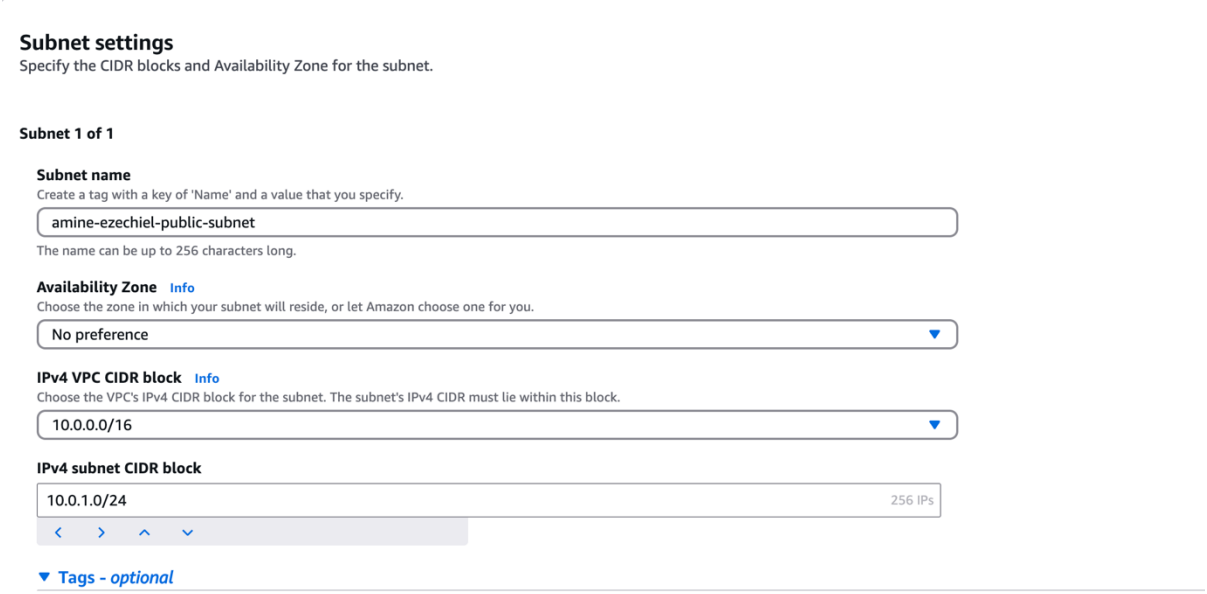
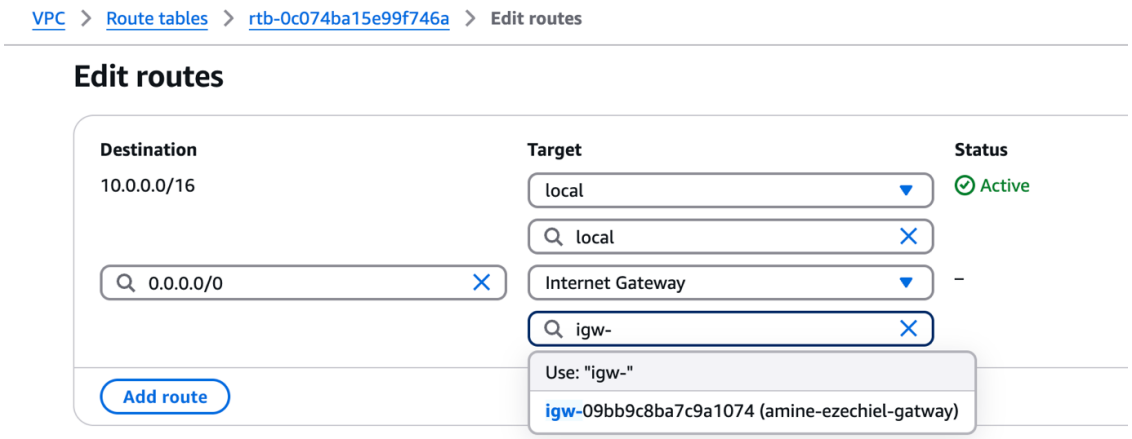
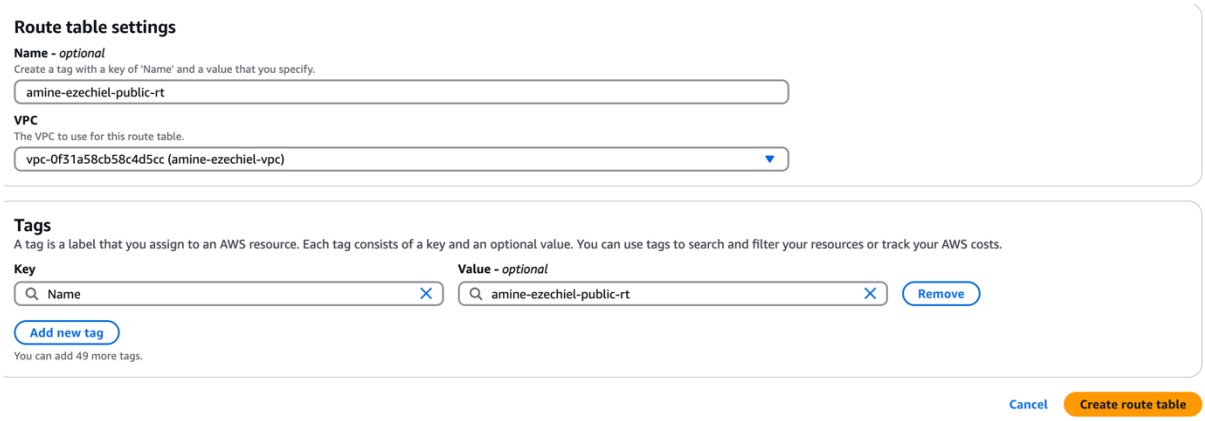


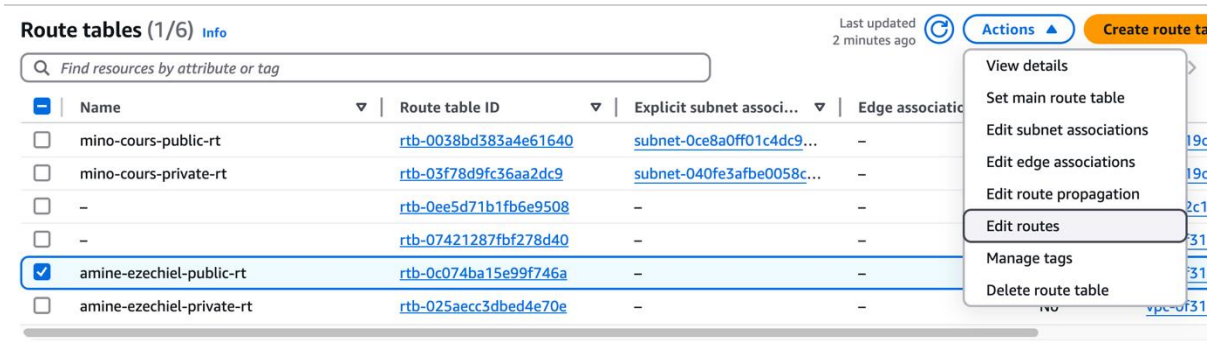
Table de routage

Nous aurons ensuite besoin de nos deux tables de routages, une pour la publique et une pour la privée qu'on associe au VPC.

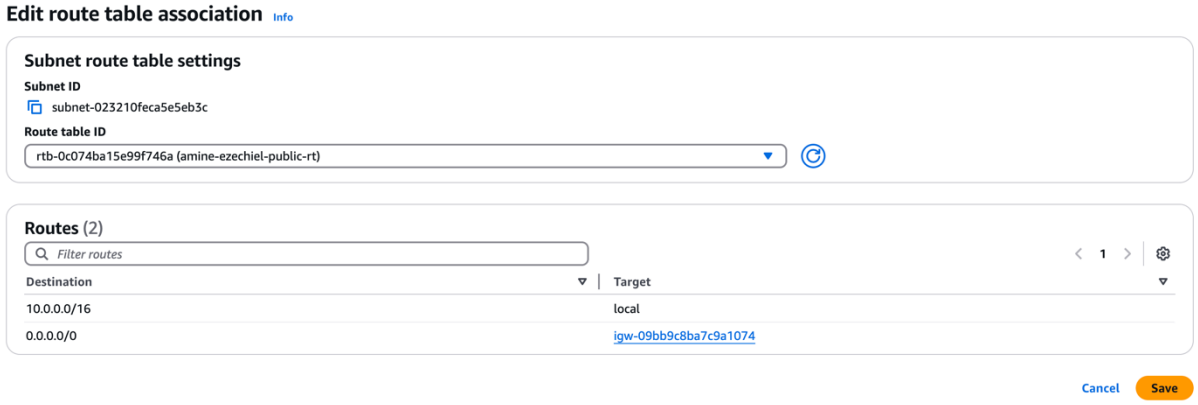




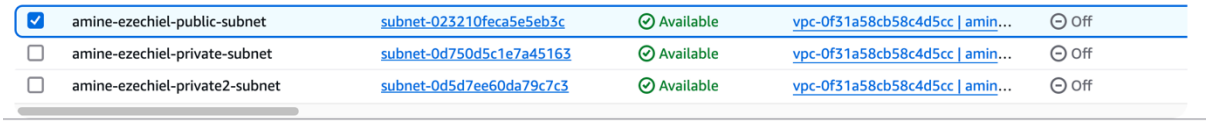
On edit ensuite les routes

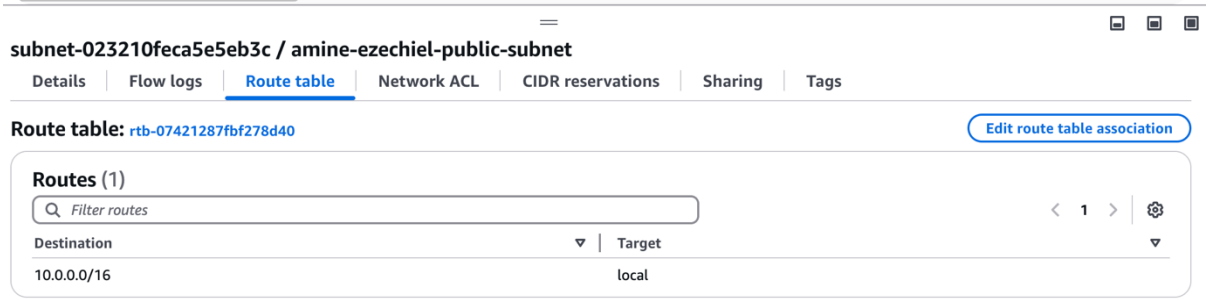


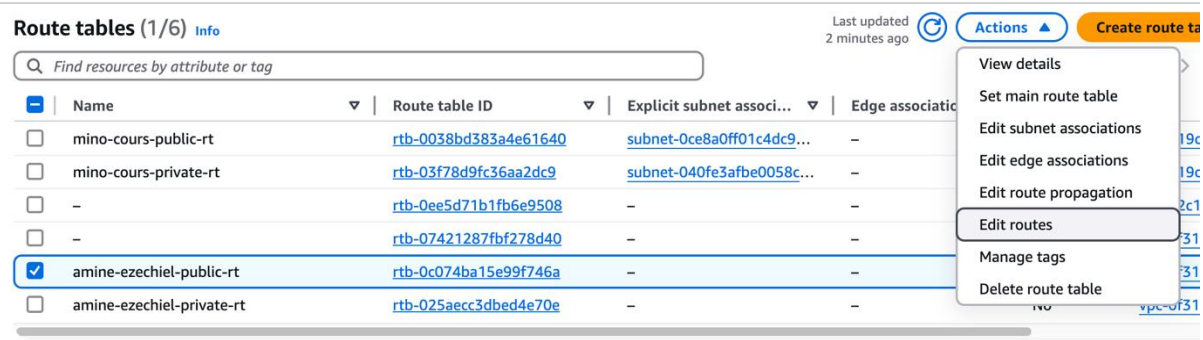
On associe nos tables de routages pour qu'ils soient connecté à notre vpc.



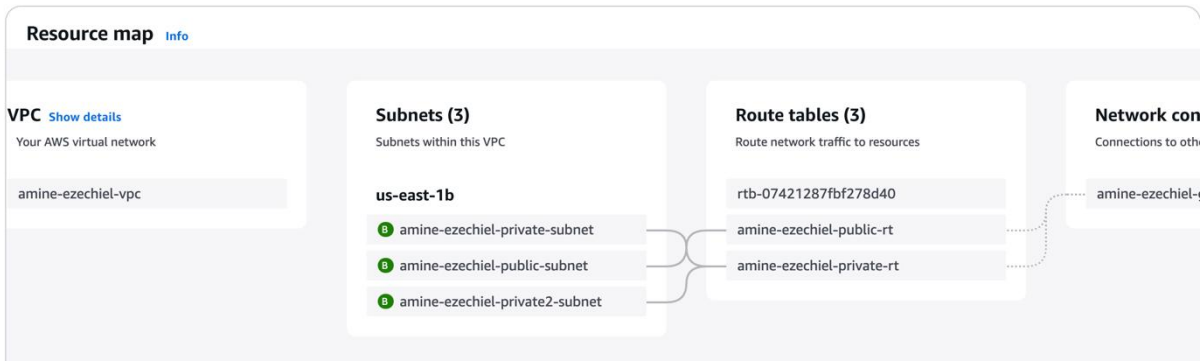
Une fois cela connecter on connecte nos sous réseaux à nos tables de routages.

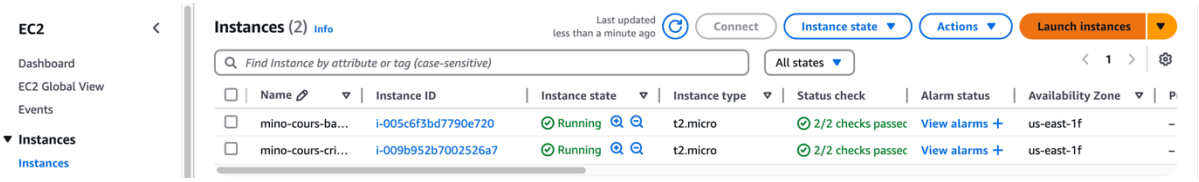






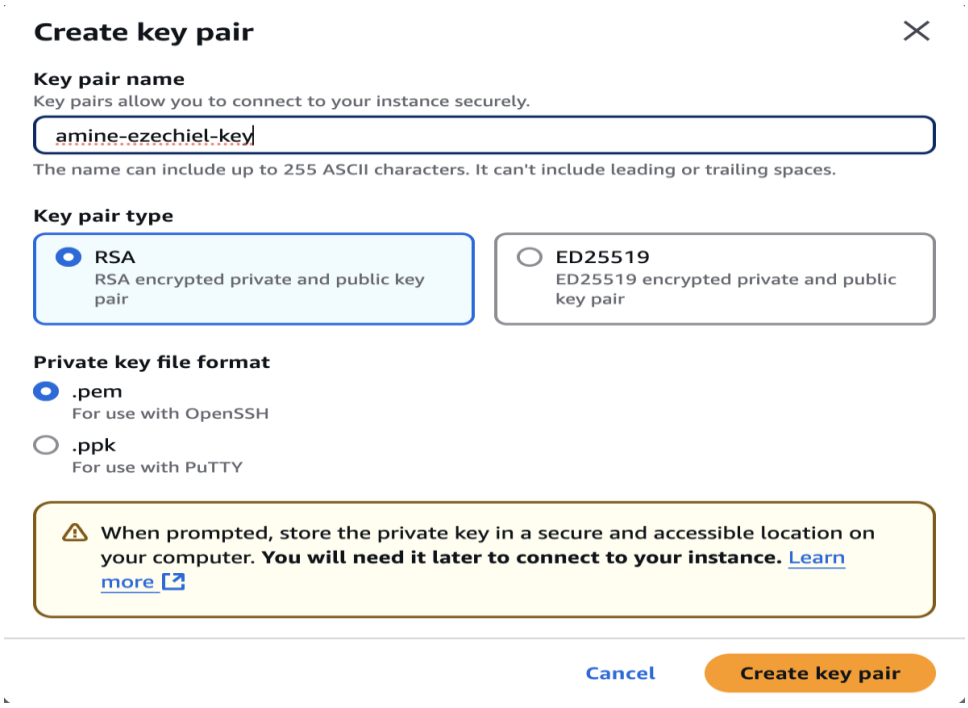
On voit ensuite nos différents sous réseaux et nos différentes tables de routages.

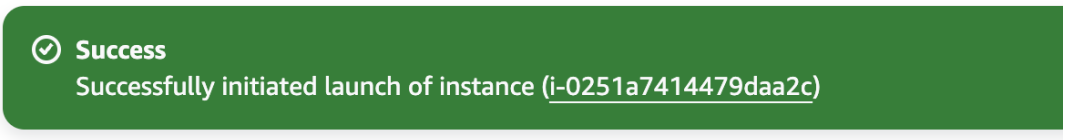




Génération de la paire de Clé

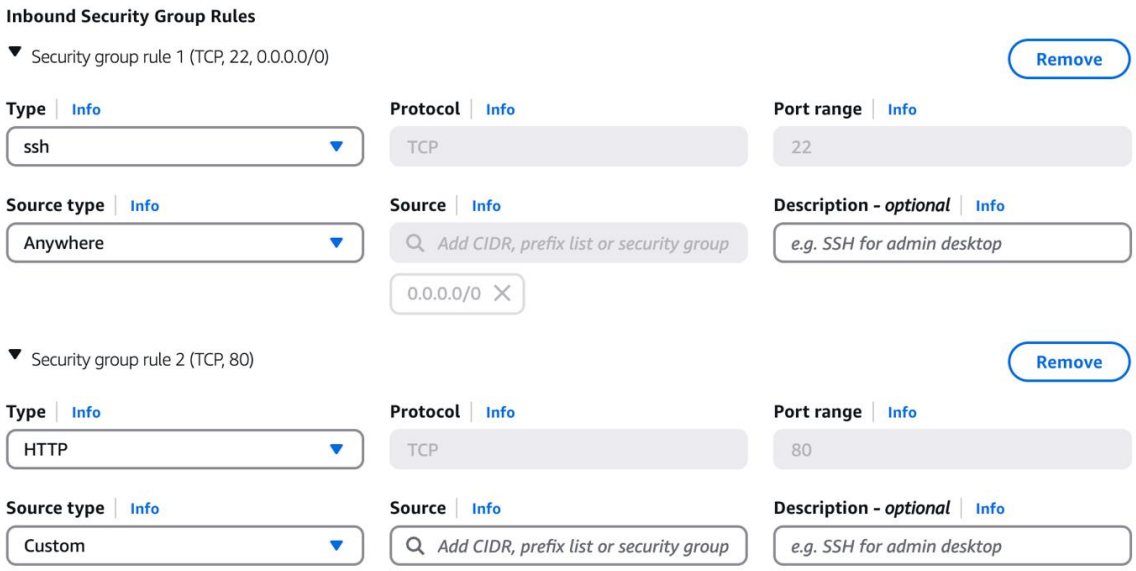
On crée une clef RSA qui permettra de se connecter en ssh de manière sécuriser avec un format .pem



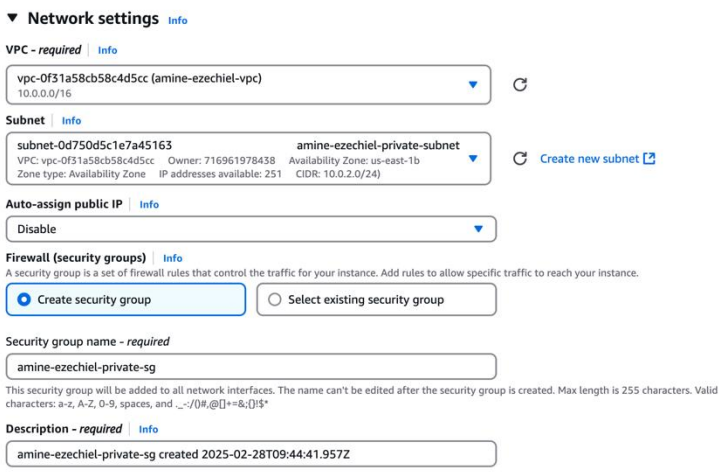


▶ Launch log

On donne les services voulus pour l’instance. Ssh pour la connexion et http pour la connexion au reverse proxy.



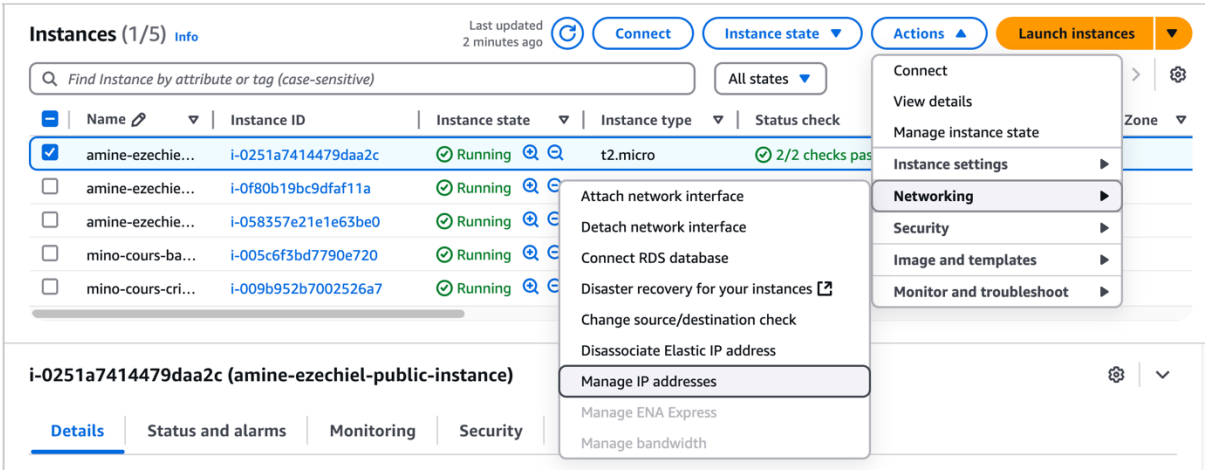
On choisit ensuite le bon sous réseaux crée et le bon VPC pour lier l’instance et les configurations faites précédemment.



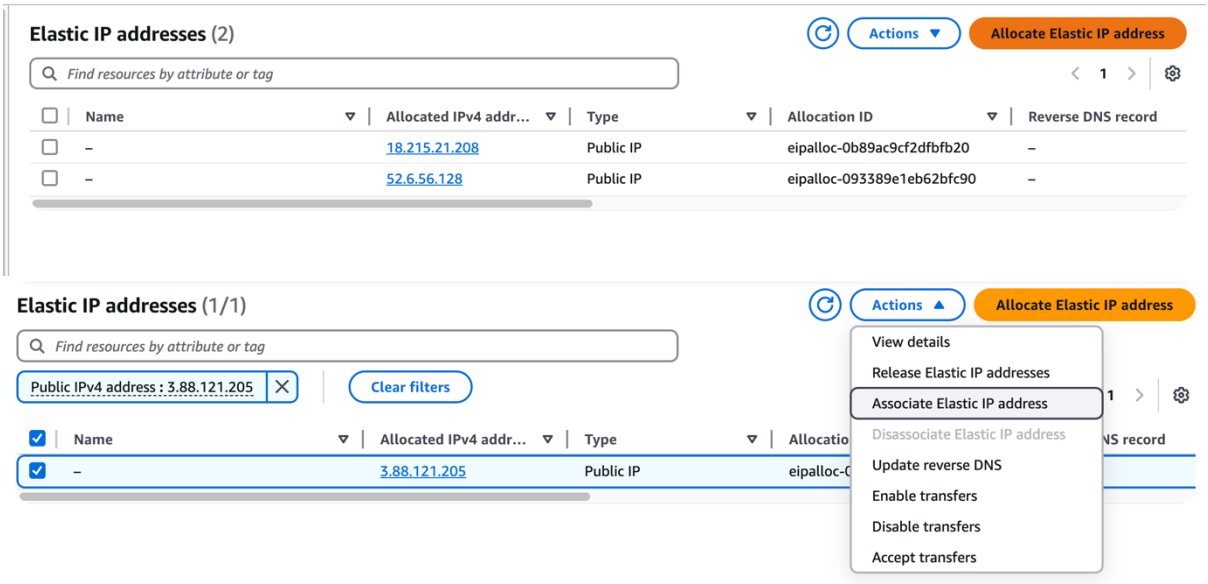
Voici le script qui permet d'avoir le service apache directement à l'instalation de la machine

```
yum install -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "<h1> Serveur Web 1<h1>" > /var/www/html/index.html
```

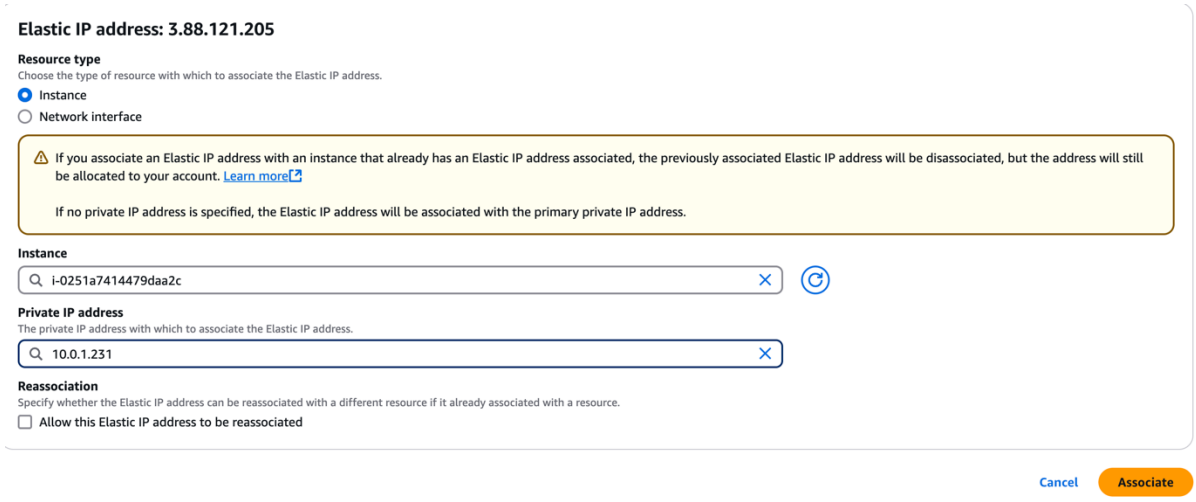
On va donner une IP publique à l'instance publique



Pour cela nous allons donner une IP elastic a l'instance



On choisis la bonne IP prive et la bonne instance.



On peut se connecter en ssh à l'instansce publique via la clef privées

```
➜  ~ sudo ssh ec2-user@3.88.121.205 -i .ssh/Mino\ Key\ EC2.pem
Password:
The authenticity of host '3.88.121.205 (3.88.121.205)' can't be established.
ED25519 key fingerprint is SHA256:gRLVSCri4TLW3eiAP9nF2FtZvtNjUP3MySxCanij7ck.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.88.121.205' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-1-231 ~]$
```

On a donc installer haProxy sur l'admin

```
Installing:
 haproxy            x86_64            2.8.3-1.amzn2023.0.1              amazonlinux            2.4 M

Transaction Summary
================================================================================================================
Install  1 Package

Total download size: 2.4 M
Installed size: 7.6 M
Downloading Packages:
haproxy-2.8.3-1.amzn2023.0.1.x86_64.rpm                                 21 MB/s | 2.4 MB     00:00
----------------------------------------------------------------------------------------------------------------
Total                                                                   16 MB/s | 2.4 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                        1/1
  Running scriptlet: haproxy-2.8.3-1.amzn2023.0.1.x86_64                                    1/1
  Installing       : haproxy-2.8.3-1.amzn2023.0.1.x86_64                                    1/1
  Running scriptlet: haproxy-2.8.3-1.amzn2023.0.1.x86_64                                    1/1
  Verifying        : haproxy-2.8.3-1.amzn2023.0.1.x86_64                                    1/1

Installed:
  haproxy-2.8.3-1.amzn2023.0.1.x86_64

Complete!
[ec2-user@ip-10-0-1-231 ~]$
```

```
[ec2-user@ip-10-0-1-231 ~]$ sudo nano /etc/haproxy/haproxy.cfg
[ec2-user@ip-10-0-1-231 ~]$ sudo systemctl restart haproxy
[ec2-user@ip-10-0-1-231 ~]$ sudo systemctl enable haproxy
Created symlink /etc/systemd/system/multi-user.target.wants/haproxy.service → /usr/lib/systemd/system/haproxy.service.
[ec2-user@ip-10-0-1-231 ~]$ sudo systemctl status haproxy
● haproxy.service - HAProxy Load Balancer
     Loaded: loaded (/usr/lib/systemd/system/haproxy.service; enabled; preset: disabled)
     Active: active (running) since Fri 2025-02-28 10:17:53 UTC; 36s ago
   Main PID: 26917 (haproxy)
     Status: "Ready."
      Tasks: 2 (limit: 1111)
     Memory: 5.9M
        CPU: 31ms
     CGroup: /system.slice/haproxy.service
             ├─26917 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -f /etc/haproxy/conf.d -p /run/haproxy.pid
             └─26919 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -f /etc/haproxy/conf.d -p /run/haproxy.pid

Feb 28 10:17:53 ip-10-0-1-231.ec2.internal systemd[1]: Starting haproxy.service - HAProxy Load Balancer...
Feb 28 10:17:53 ip-10-0-1-231.ec2.internal haproxy[26917]: [NOTICE]   (26917) : haproxy version is 2.8.3-86e043a
Feb 28 10:17:53 ip-10-0-1-231.ec2.internal haproxy[26917]: [NOTICE]   (26917) : path to executable is /usr/sbin/haproxy
Feb 28 10:17:53 ip-10-0-1-231.ec2.internal haproxy[26917]: [ALERT]    (26917) : config : parsing [/etc/haproxy/haproxy.c>
Feb 28 10:17:53 ip-10-0-1-231.ec2.internal haproxy[26917]: [NOTICE]   (26917) : New worker (26919) forked
Feb 28 10:17:53 ip-10-0-1-231.ec2.internal systemd[1]: Started haproxy.service - HAProxy Load Balancer.
Feb 28 10:17:53 ip-10-0-1-231.ec2.internal haproxy[26917]: [NOTICE]   (26917) : Loading success.
Feb 28 10:17:53 ip-10-0-1-231.ec2.internal haproxy[26919]: [WARNING]  (26919) : Server web_servers/web1 is DOWN, reason:>
Feb 28 10:17:54 ip-10-0-1-231.ec2.internal haproxy[26919]: [WARNING]  (26919) : Server web_servers/web2 is DOWN, reason:>
Feb 28 10:17:54 ip-10-0-1-231.ec2.internal haproxy[26919]: [ALERT]    (26919) : backend 'web_servers' has no server avai>
```

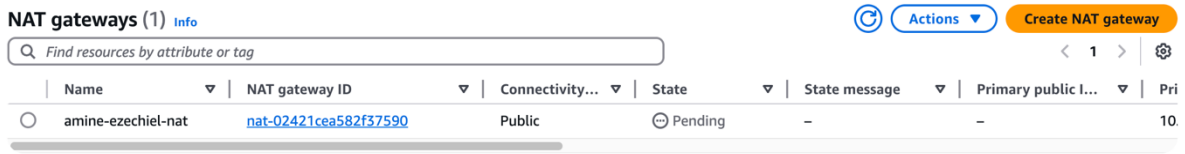
Voici le fichier de configuration haproxy pour que le load balancing est lieu avec les ip de nos serveurs

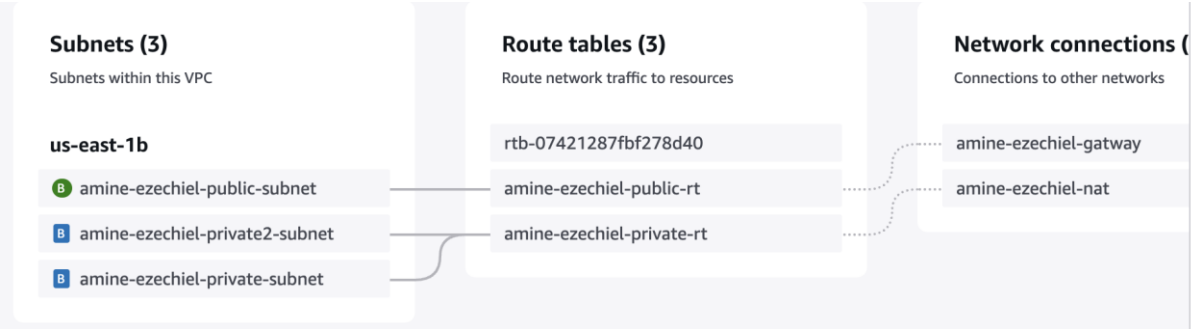
```
global
    log /dev/log local0
    log /dev/log local1 notice
    chroot /var/lib/haproxy
    stats socket /var/lib/haproxy/stats mode 660 level admin
    stats timeout 30s
    user haproxy
    group haproxy
    daemon

defaults
    log global
    mode http
    option httplog
    option dontlognull
    timeout connect 5000ms
    timeout client 50000ms
    timeout server 50000ms

frontend http_front
    bind *:80
    default_backend web_servers

backend web_servers
    balance roundrobin
    server web1 10.0.2.198:80 check
    server web2 10.0.3.33:80 check
```

On a peut donc se connecter aux instances privées en envoyant la clef dans l’instance publique

```
[ec2-user@ip-10-0-1-231 .ssh]$ sudo ssh ec2-user@10.0.2.198 -i ~/.ssh/Minoclef
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-2-198 ~]$
```

Nous avions un problème de connexion avec nos instances privées car elle n'était pas connectée un NAT Gateway pour acceder à internet etexectuer le script.





Le fichier index.html n'était pas créé dans les serveurs donc on l'as crée et personalise avec un numéro de serveur

```
[ec2-user@ip-10-0-3-33 ~]$ sudo touch /var/www/html/index.html
[ec2-user@ip-10-0-3-33 ~]$ sudo nano /var/www/html/index.html
```

```
  GNU nano 5.8                  /var/www/html/index.html
<h1>Server2<h2>
```

```
[ec2-user@ip-10-0-3-33 ~]$ yum update -y
Error: This command has to be run with superuser privileges (under the root user
 on most systems).
[ec2-user@ip-10-0-3-33 ~]$ sudo yum update -y
Amazon Linux 2023 repository                 40 MB/s |  32 MB     00:00
Amazon Linux 2023 Kernel Livepatch repository  110 kB/s |  14 kB     00:00
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-10-0-3-33 ~]$ sudo yum install httpd -y
Last metadata expiration check: 0:00:13 ago on Fri Feb 28 11:21:50 2025.
Dependencies resolved.
```

Test

On a bien les deux sevreurs quand on refresh la page