CLOUD COMPUTING PRACTICALS

Name: Saurav Kumar

Submitted to: Mrs. Asmita Patel (Guest Faculty, Ramanujan College)

Submitted by: Saurav Kumar
Roll NO.: 20222745
Semester – VI
Course – B.Voc (Software development)

Differentiate between centralized and decentralized system

Code

one.py > ...

```python
import multiprocessing
import time

def centralized_sum(n):
    return sum(range(1, n + 1))

def calculate_partial_sum(start, end):
    return sum(range(start, end + 1))

def distributed_sum(n, num_workers):
    chunk_size = n // num_workers
    ranges = [
        (i * chunk_size + 1, (i + 1) * chunk_size if i < num_workers - 1 else n)
        for i in range(num_workers)
    ]

    with multiprocessing.Pool(num_workers) as pool:
        results = pool.starmap(calculate_partial_sum, ranges)

    return sum(results)

def main(n, num_workers):
    # Centralized computation
    start_time = time.time()
    centralized_result = centralized_sum(n)
    centralized_time = time.time() - start_time
    print(f"Centralized sum from 1 to {n}: {centralized_result}")
    print(f"Time taken for centralized approach: {centralized_time:.4f} seconds")

    # Distributed computation
    start_time = time.time()
    distributed_result = distributed_sum(n, num_workers)
    distributed_time = time.time() - start_time
    print(f"Distributed sum from 1 to {n}: {distributed_result}")
    print(f"Time taken for distributed approach with {num_workers} workers: {distributed_time:.4f} seconds")
```

```python

if __name__ == "__main__":
    N = 8_000_000
    num_workers = 5
    main(N, num_workers)

```

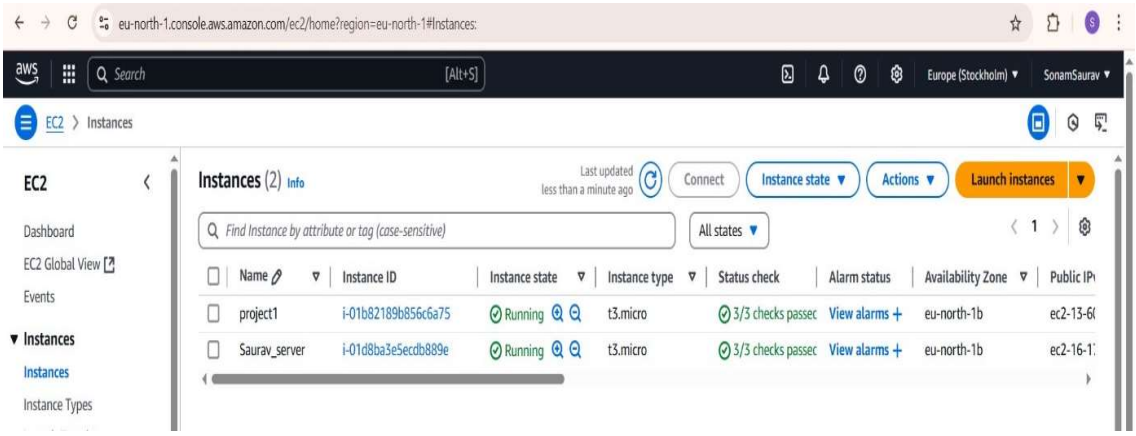
Output

```
PROBLEMS 3   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS C:\Users\HP\OneDrive\Desktop\Cloud_Computing\python>
& C:/Users/HP/AppData/Local/Programs/Python/Python313/python.exe c:/Users/HP/OneDrive/Desktop/Cloud_Computing/python/one.py
Centralized sum from 1 to 8000000: 32000004000000
Time taken for centralized approach: 0.0929 seconds
Distributed sum from 1 to 8000000: 32000004000000
Time taken for distributed approach with 5 workers: 0.2697 seconds
PS C:\Users\HP\OneDrive\Desktop\Cloud_Computing\python>
```

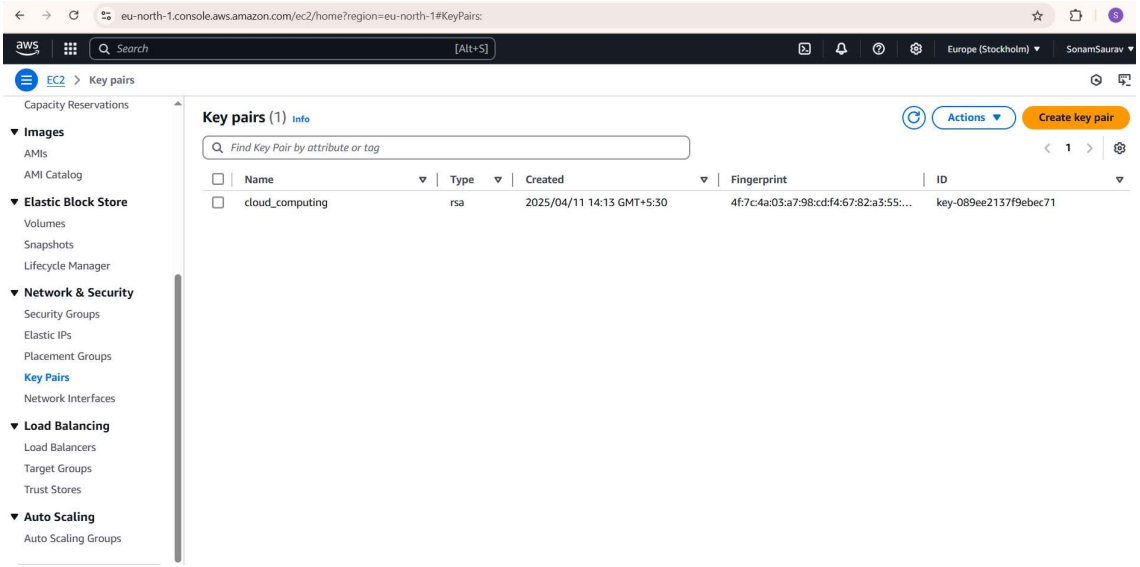
1. Launch an EC2 Instance (Ubuntu)

a. Create a new EC2 instance using an **Ubuntu AMI**.
b. Select a **General Purpose instance type**.
c. Configure key pair, security group, and default network settings.

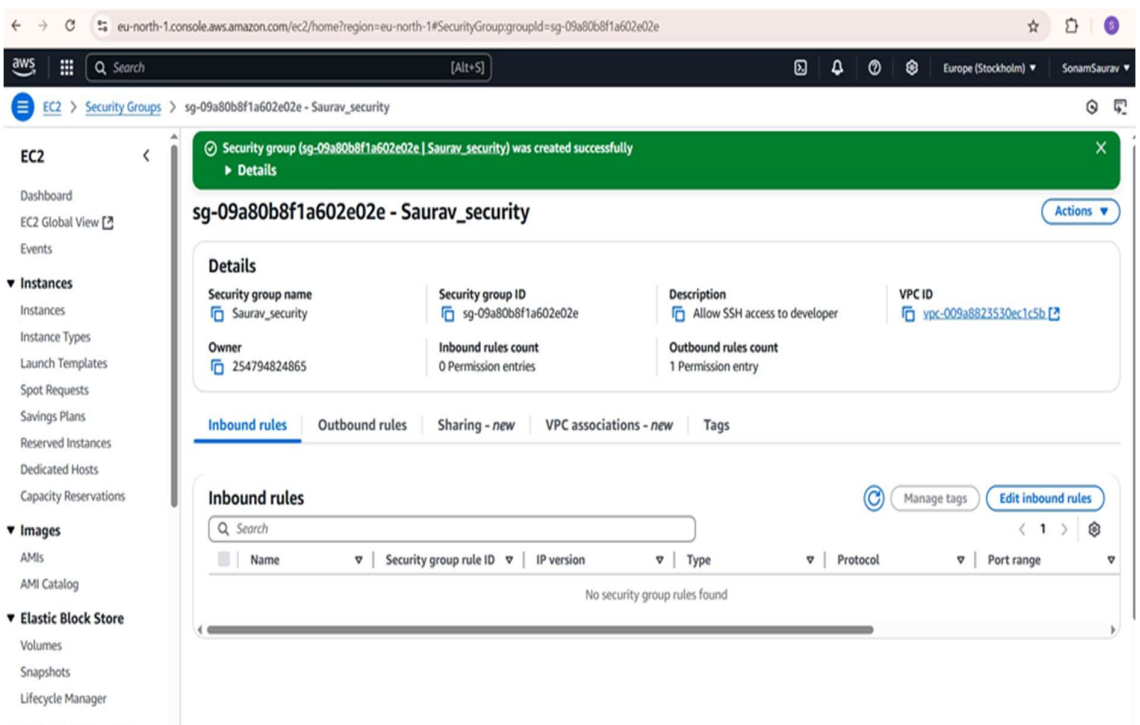
a.



b.



C.



2. *Connect to EC2 via SSH and Run Basic Commands*

a. Use SSH to connect to EC2 instance from local machine.
b. Run the following commands:

$cat /etc/os-release and $free -m

```
 System information as of Fri May  2 13:39:30 UTC 2025

  System load:  0.08              Temperature:           -273.1 C
  Usage of /:   25.2% of 6.71GB   Processes:             107
  Memory usage: 22%               Users logged in:       0
  Swap usage:   0%                IPv4 address for ens5: 172.31.32.218


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Fri May  2 13:33:44 2025 from 13.48.4.203
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-32-218:~$ cat/etc/os-release
-bash: cat/etc/os-release: No such file or directory
ubuntu@ip-172-31-32-218:~$ free -m
               total        used        free      shared  buff/cache   available
Mem:             914         339         366           2         362         574
Swap:              0           0           0
```

$ lscpu

```
ubuntu@ip-172-31-32-218:~$ lscpu
Architecture:            x86_64
  CPU op-mode(s):        32-bit, 64-bit
  Address sizes:         46 bits physical, 48 bits virtual
  Byte Order:            Little Endian
CPU(s):                  2
  On-line CPU(s) list:   0,1
Vendor ID:               GenuineIntel
  Model name:            Intel(R) Xeon(R) Platinum 8259CL CPU @ 2.50GHz
    CPU family:          6
    Model:               85
    Thread(s) per core:  2
    Core(s) per socket:  1
    Socket(s):           1
    Stepping:            7
    BogoMIPS:            4999.95
    Flags:               fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss ht syscall nx pdpe1gb
                         _tsc rep_good nopl xtopology nonstop_tsc cpuid tsc_known_freq pni pclmulqdq ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe
                         e_timer aes xsave avx f16c rdrand hypervisor lahf_lm abm 3dnowprefetch pti fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms in
                         vx512dq rdseed adx smap clflushopt clwb avx512cd avx512bw avx512vl xsaveopt xsavec xgetbv1 xsaves ida arat pku ospke
Virtualization features:
  Hypervisor vendor:     KVM
  Virtualization type:   full
Caches (sum of all):
  L1d:                   32 KiB (1 instance)
  L1i:                   32 KiB (1 instance)
  L2:                    1 MiB (1 instance)
  L3:                    35.8 MiB (1 instance)
```

```
Caches (sum of all):
  L1d:                    32 KiB (1 instance)
  L1i:                    32 KiB (1 instance)
  L2:                     1 MiB (1 instance)
  L3:                     35.8 MiB (1 instance)
NUMA:
  NUMA node(s):           1
  NUMA node0 CPU(s):      0,1
Vulnerabilities:
  Gather data sampling:   Unknown: Dependent on hypervisor status
  Itlb multihit:          KVM: Mitigation: VMX unsupported
  L1tf:                   Mitigation; PTE Inversion
  Mds:                    Vulnerable: Clear CPU buffers attempted, no microcode; SMT Host state unknown
  Meltdown:               Mitigation; PTI
  Mmio stale data:        Vulnerable: Clear CPU buffers attempted, no microcode; SMT Host state unknown
  Reg file data sampling: Not affected
  Retbleed:               Vulnerable
  Spec rstack overflow:   Not affected
  Spec store bypass:      Vulnerable
  Spectre v1:             Mitigation; usercopy/swapgs barriers and __user pointer sanitization
  Spectre v2:             Mitigation; Retpolines; STIBP disabled; RSB filling; PBRSB-eIBRS Not affected; BHI Retpoline
  Srbds:                  Not affected
  Tsx async abort:        Not affected
ubuntu@ip-172-31-32-218:~$
```

$ df -h and $ ip a and $ uname -r and $ ls -lh and $ chmod

```
ubuntu@ip-172-31-32-218:~$ df -h
Filesystem       Size  Used Avail Use% Mounted on
/dev/root        6.8G  1.7G  5.0G  26% /
tmpfs            458M     0  458M   0% /dev/shm
tmpfs            183M  880K  182M   1% /run
tmpfs            5.0M     0  5.0M   0% /run/lock
efivarfs         128K  3.6K  120K   3% /sys/firmware/efi/efivars
/dev/nvme0n1p16  881M   79M  741M  10% /boot
/dev/nvme0n1p15  105M  6.1M   99M   6% /boot/efi
tmpfs             92M   12K   92M   1% /run/user/1000
ubuntu@ip-172-31-32-218:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc mq state UP group default qlen 1000
    link/ether 0a:c9:dc:1d:2a:77 brd ff:ff:ff:ff:ff:ff
    altname enp0s5
    inet 172.31.32.218/20 metric 100 brd 172.31.47.255 scope global dynamic ens5
       valid_lft 3374sec preferred_lft 3374sec
    inet6 fe80::8c9:dcff:fe1d:2a77/64 scope link
       valid_lft forever preferred_lft forever
ubuntu@ip-172-31-32-218:~$ uname -r
6.8.0-1024-aws
ubuntu@ip-172-31-32-218:~$ ls -lh
total 0
ubuntu@ip-172-31-32-218:~$ chmod
chmod: missing operand
Try 'chmod --help' for more information.
```

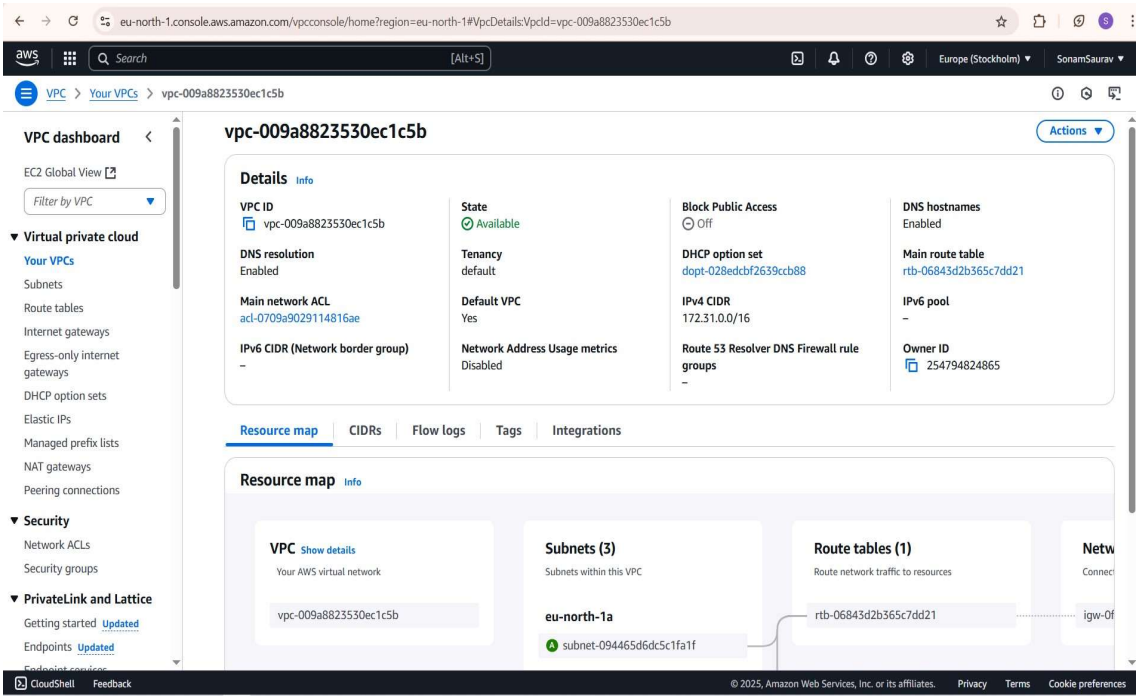
$ls and $ exit

```
ubuntu@ip-172-31-32-218:~$ ls
ubuntu@ip-172-31-32-218:~$ exit
logout
```

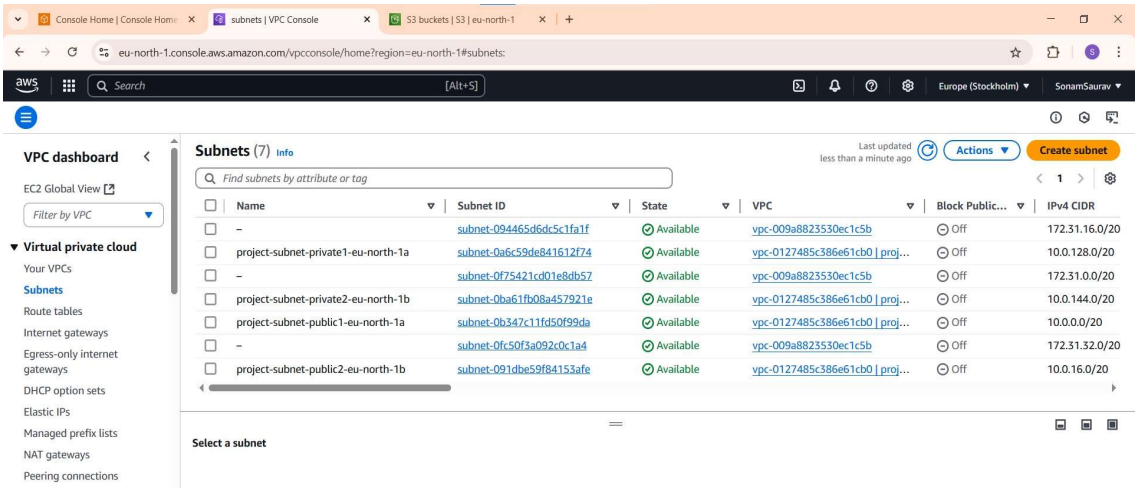
3. Create a VPC with Public and Private Subnets

a. Create a **custom VPC**.
b. Add one **public subnet** and one **private subnet** within the VPC.

a.



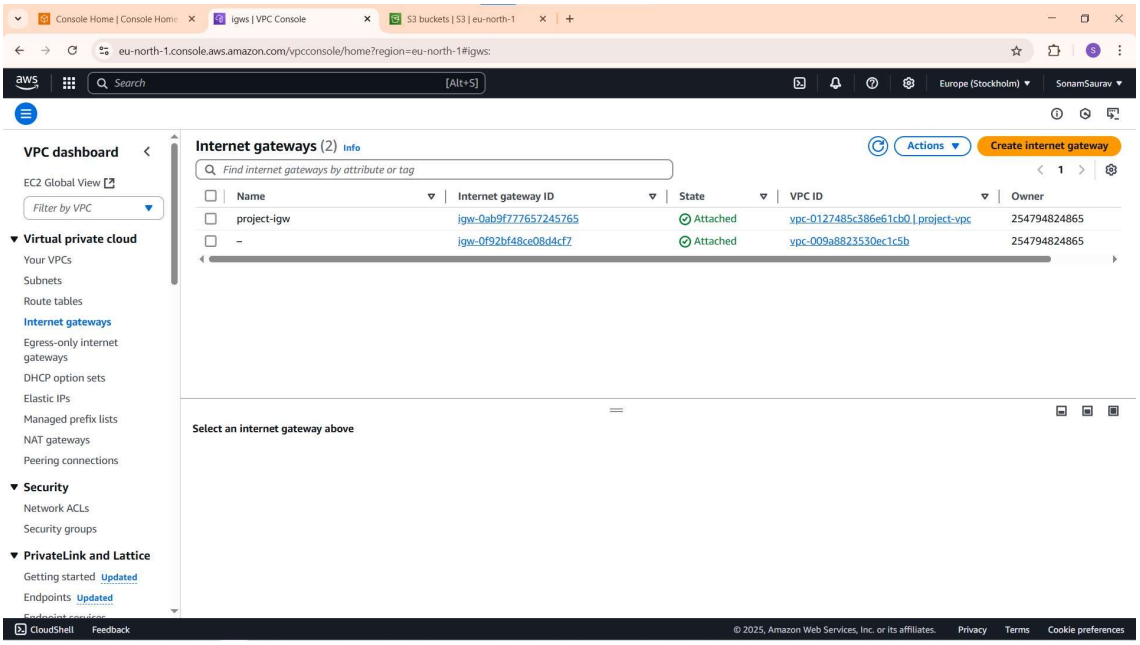
b.



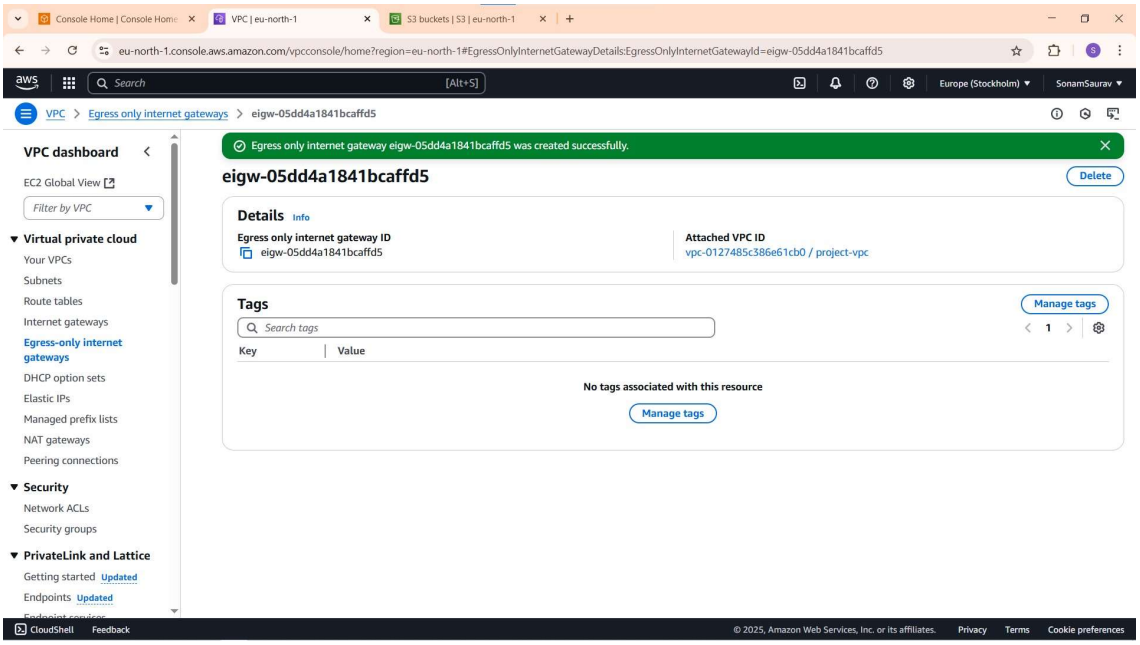
4. Create and Attach an Internet Gateway

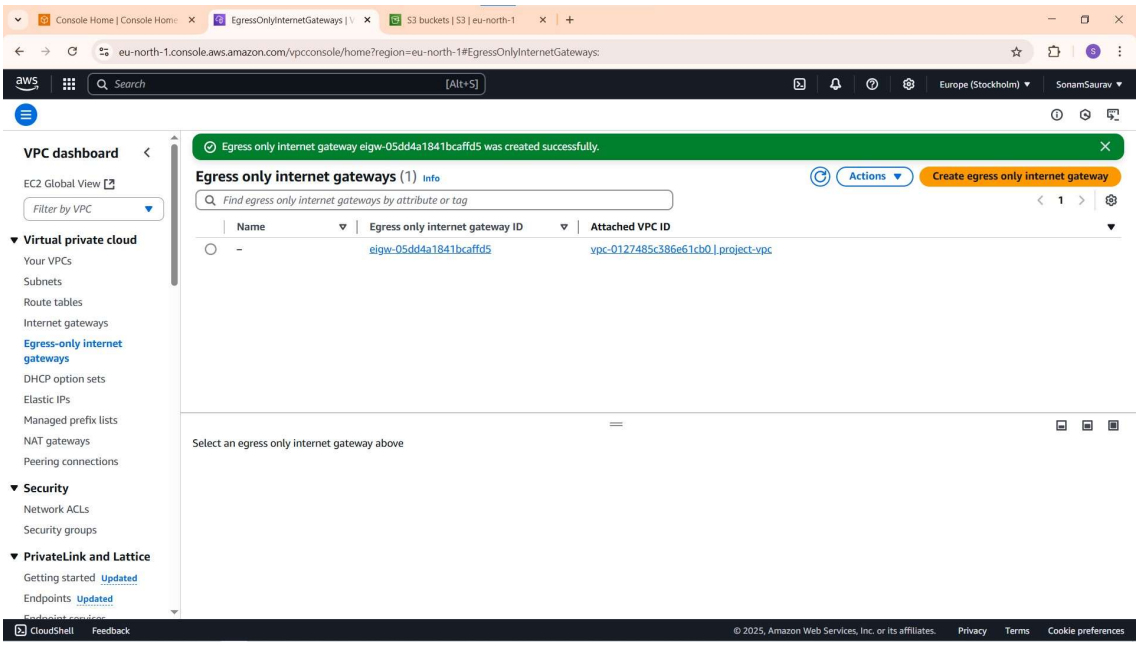
a. Create an **Internet Gateway (IGW)**.
b. Attach the IGW to your VPC to allow internet access for public subnets.

a.



b.





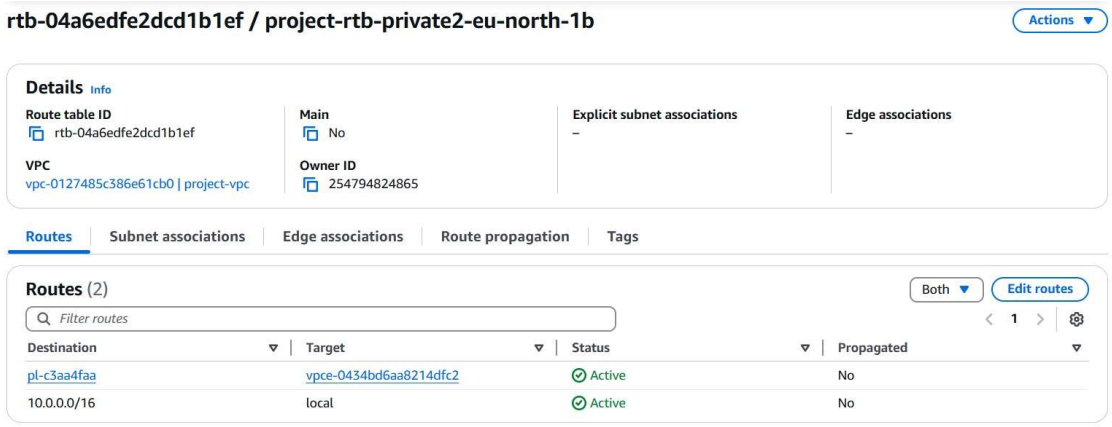
5. *Create and Configure Route Tables*

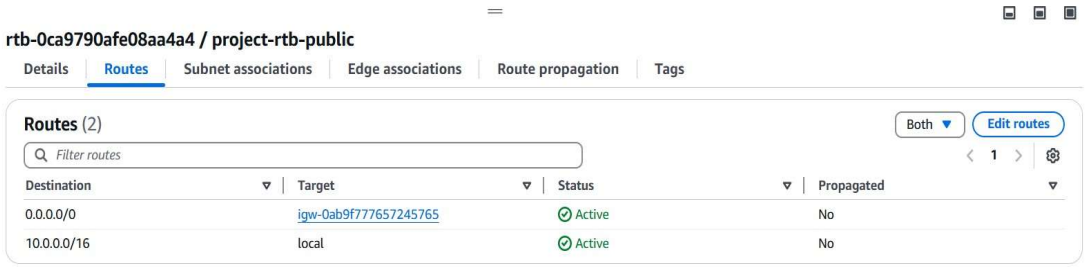
a. Create two route tables:

- A **public route table** with a route to 0.0.0.0/0 via the IGW.
- A **private route table** with no internet route.

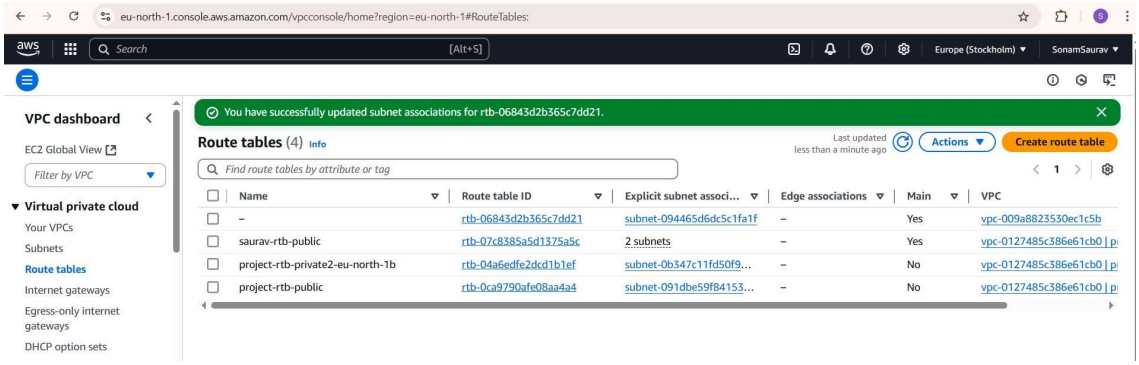
b. Associate each route table with the appropriate subnet.

a.





b.

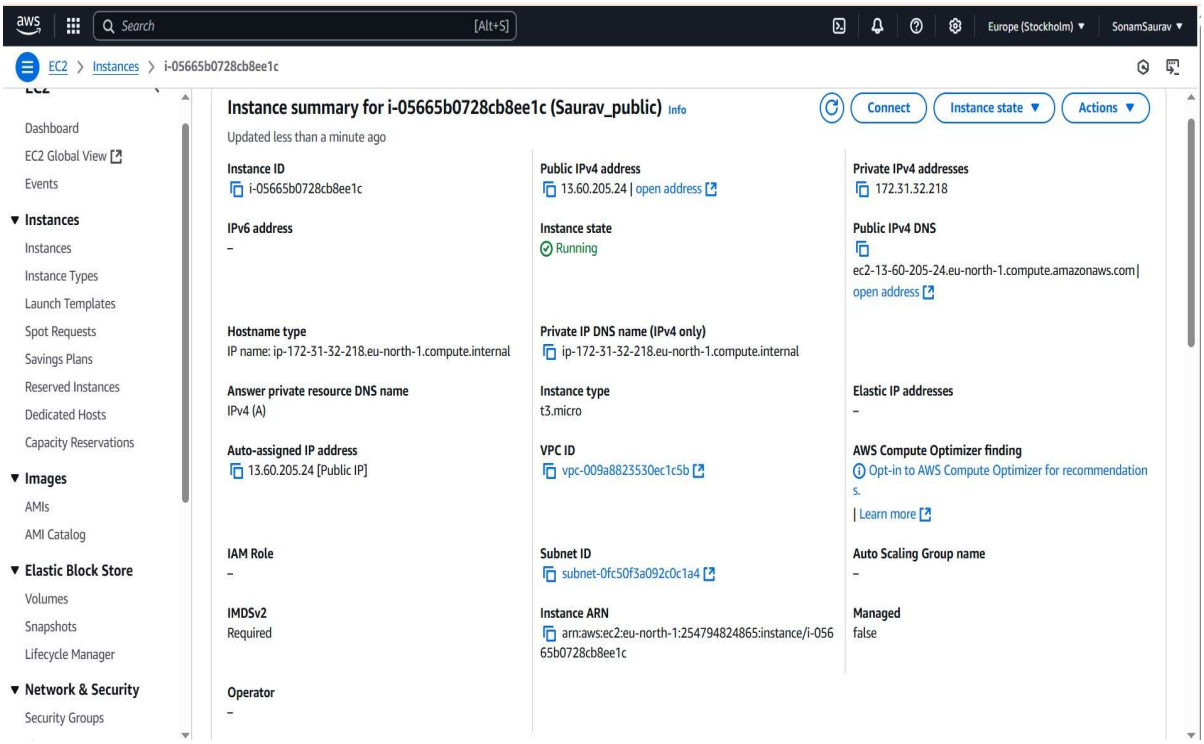


6. *Launch EC2 Instances in Both Subnets*

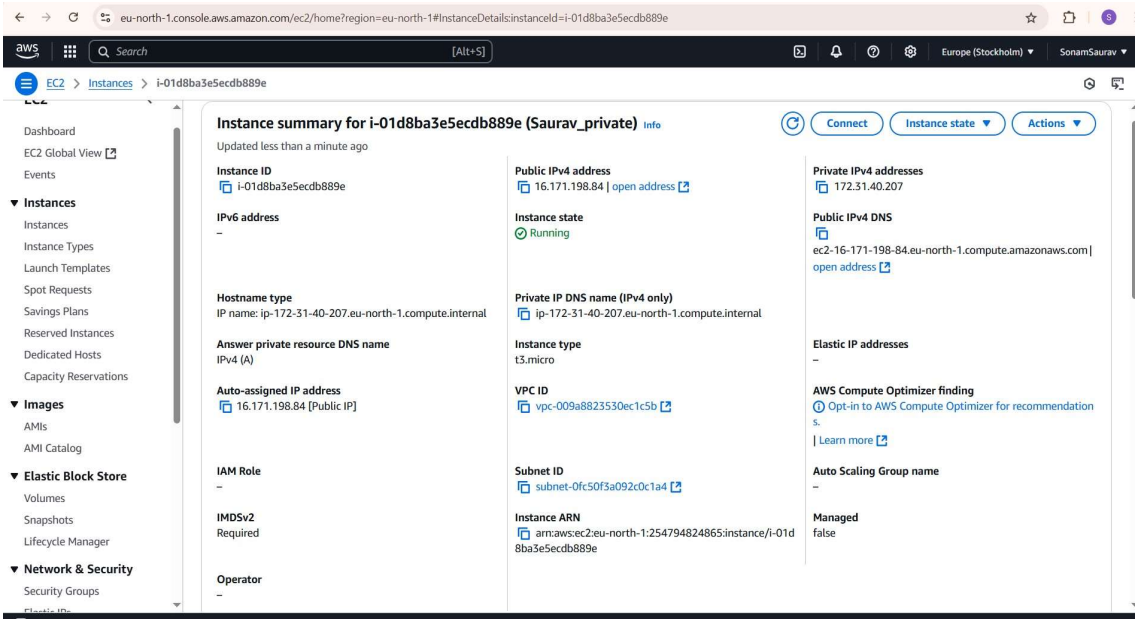
a. Launch one EC2 instance in the **public subnet** and another in the **private subnet**.

b. Ensure the public instance has a public IP for external access.

public subnet



Private subnet



7. Install NGINX and Deploy a Sample Application

a. SSH into the public EC2 instance.

b. Install **NGINX** and deploy a basic HTML page: "Welcome to AWS"

a.

Install nginx

$sudo –i and $apt-get update and $nginx –t

```
*** System restart required ***
Last login: Fri Apr 11 08:58:54 2025 from 13.48.4.202
ubuntu@ip-172-31-43-251:~$ sudo -i
root@ip-172-31-43-251:~# apt-get update
Hit:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1057 kB]
Get:6 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [161 kB]
Get:7 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1060 kB]
Get:8 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [376 kB]
Get:9 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:10 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:11 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7080 B]
Get:12 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [16.4 kB]
Get:13 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [212 B]
Get:14 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:15 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.5 kB]
Get:16 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.2 kB]
Get:17 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:18 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Fetched 3132 kB in 1s (3695 kB/s)
Reading package lists... Done
root@ip-172-31-43-251:~# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

$apt install nginx* and *$ Service nginx status

```
root@ip-172-31-43-251:~# apt-get install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nginx is already the newest version (1.24.0-2ubuntu7.3).
0 upgraded, 0 newly installed, 0 to remove and 48 not upgraded.
root@ip-172-31-43-251:~# service nginx status
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
     Active: active (running) since Fri 2025-04-11 09:03:50 UTC; 3 weeks 0 days ago
       Docs: man:nginx(8)
   Main PID: 3049 (nginx)
      Tasks: 3 (limit: 1077)
     Memory: 2.4M (peak: 3.2M)
        CPU: 18ms
     CGroup: /system.slice/nginx.service
             ├─3049 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             ├─3050 "nginx: worker process"
             └─3051 "nginx: worker process"

Apr 11 09:03:50 ip-172-31-43-251 systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
Apr 11 09:03:50 ip-172-31-43-251 systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
```

$curl localhost

```
root@ip-172-31-43-251:~# curl localhost
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

$ cd/var/www/html/* and *$ ls* and *$cat index.nginx.debian.html

```
root@ip-172-31-43-251:~# cd /var/www/html/
root@ip-172-31-43-251:/var/www/html# ls
index.nginx-debian.html
root@ip-172-31-43-251:/var/www/html# cat index.nginx-debian.html
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

$echo "Welcome to learn AWS">index.html* and *$curl localhost

```
root@ip-172-31-43-251:/var/www/html# echo "Welcome to learn AWS">index.html
root@ip-172-31-43-251:/var/www/html# curl localhost
Welcome to learn AWS
root@ip-172-31-43-251:/var/www/html#
```

8. Use a Bootstrap Script (User Data)

a. While launching a new EC2 instance, include a **user data script** to automatically install software and configure the environment.

a.

User data - *optional* | Info

Upload a file with your user data or enter it in the field.

Choose file

```
#!/bin/bash
# Update the instance
apt-get update -y

# Install Apache (apache2)
apt-get install -y apache2

# Enable and start Apache service
systemctl enable apache2
systemctl start apache2

# Create a sample index.html page
echo "<html><h1>Welcome to your EC2 instance</h1><p>Deployed using EC2 User Data
script!</p></html>" > /var/www/html/index.html
```

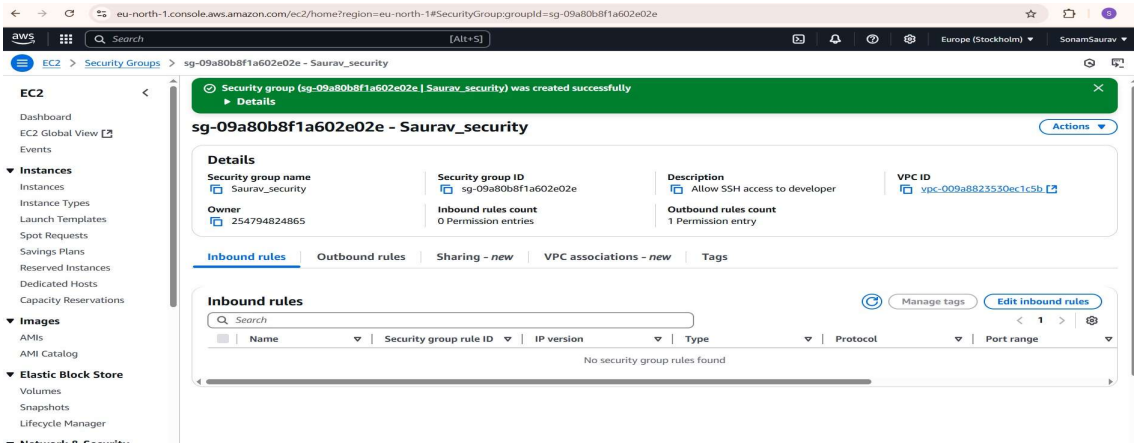
9. Create a Security Group with Inbound and Outbound Rules

a. Create a **custom security group**.

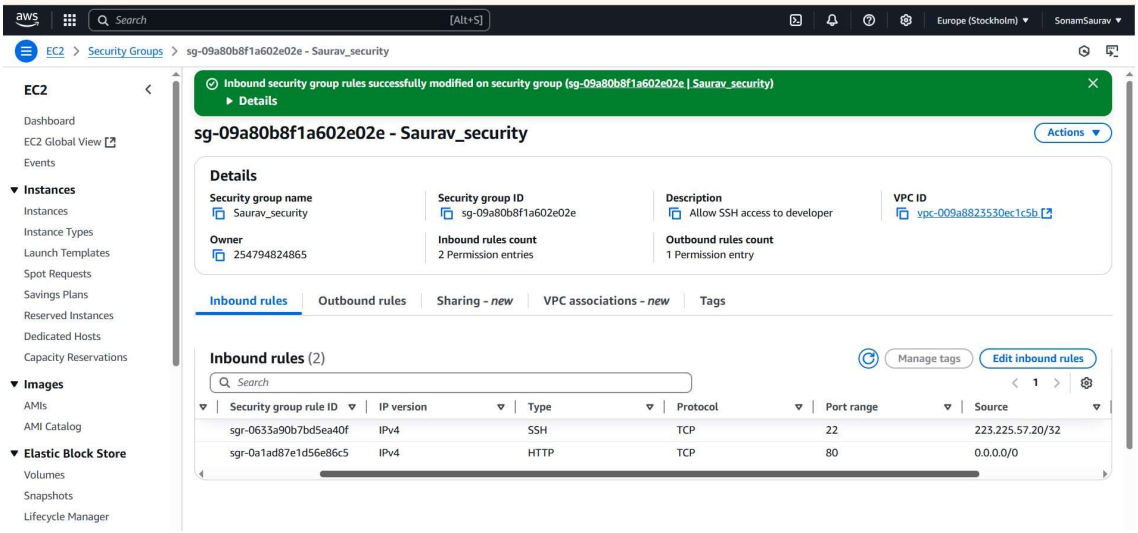
b. Configure:

- Inbound rules: Allow SSH (22) from your IP, HTTP (80) from anywhere.
- Outbound rules: Allow all traffic (default).

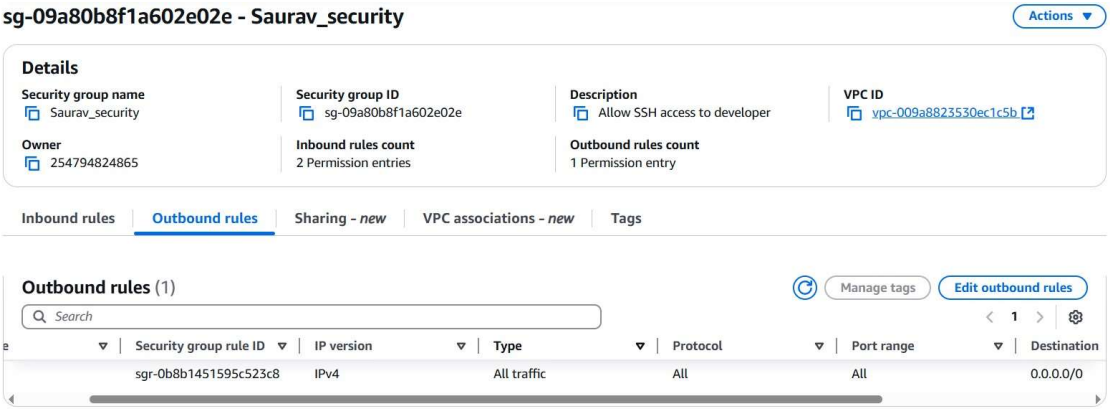
a.



b. Inbound_rule



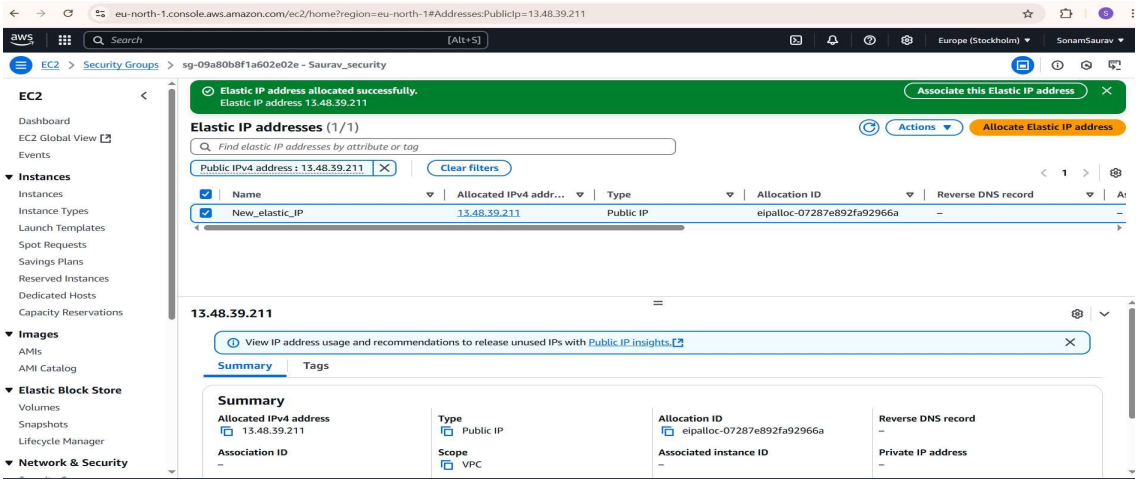
Outbound_rule



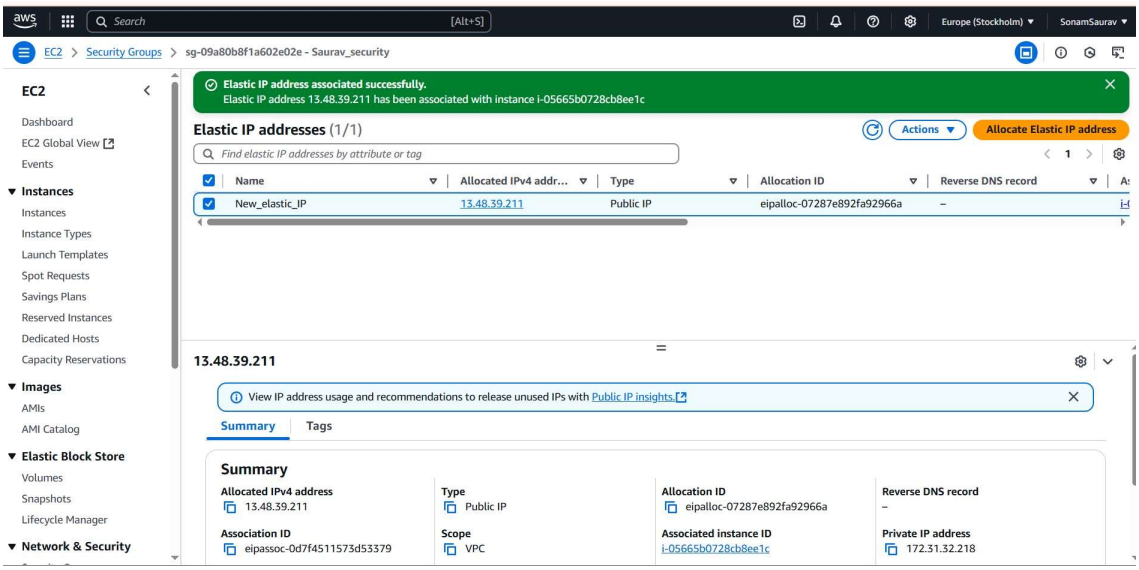
10. Allocate and Attach an Elastic IP (EIP)

a. Allocate a new **Elastic IP address**.
b. Associate the EIP with your public EC2 instance.

a.



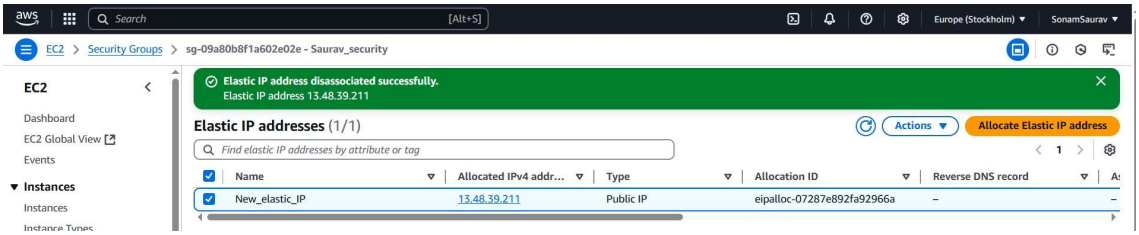
b.



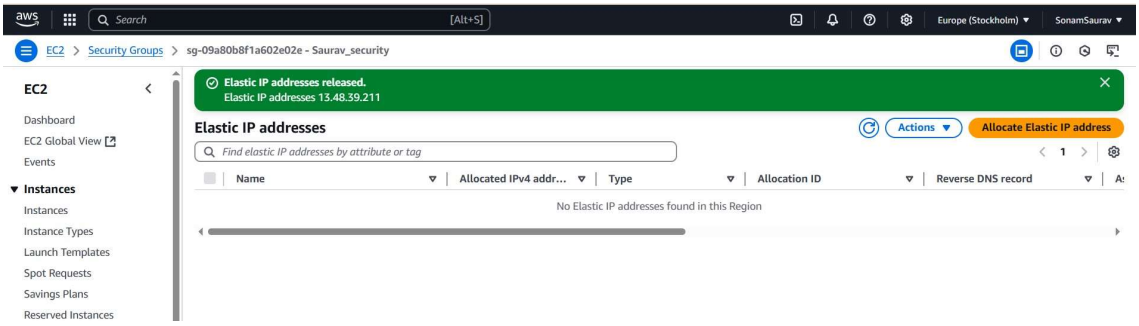
11. Detach and Release Elastic IP

a. Disassociate the EIP from your instance.
b. Release the EIP from your AWS account to avoid unnecessary charges.

a.

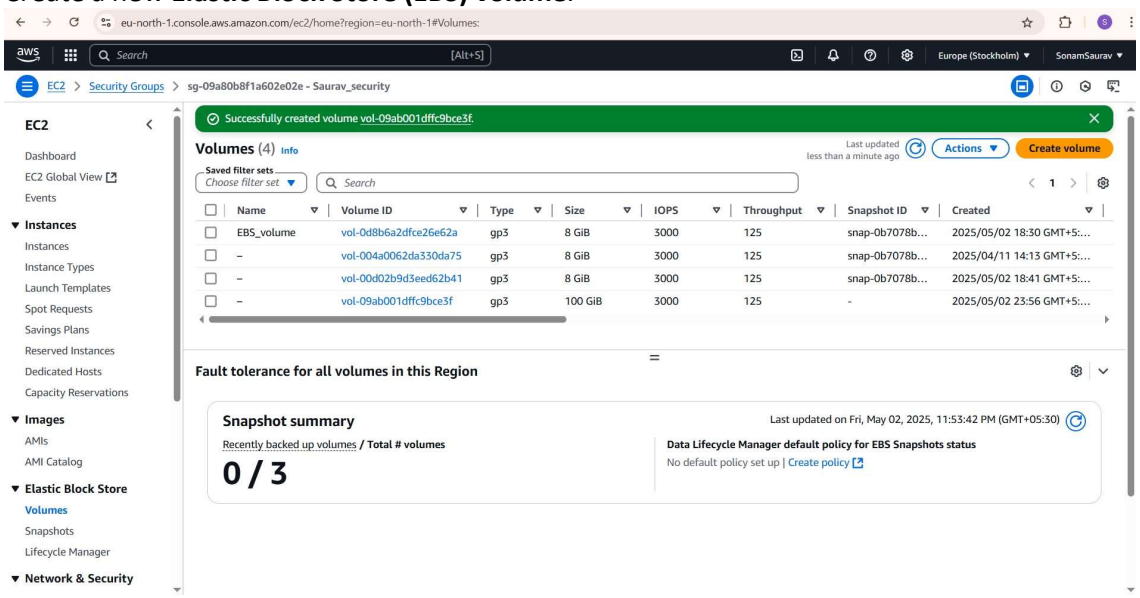


b.

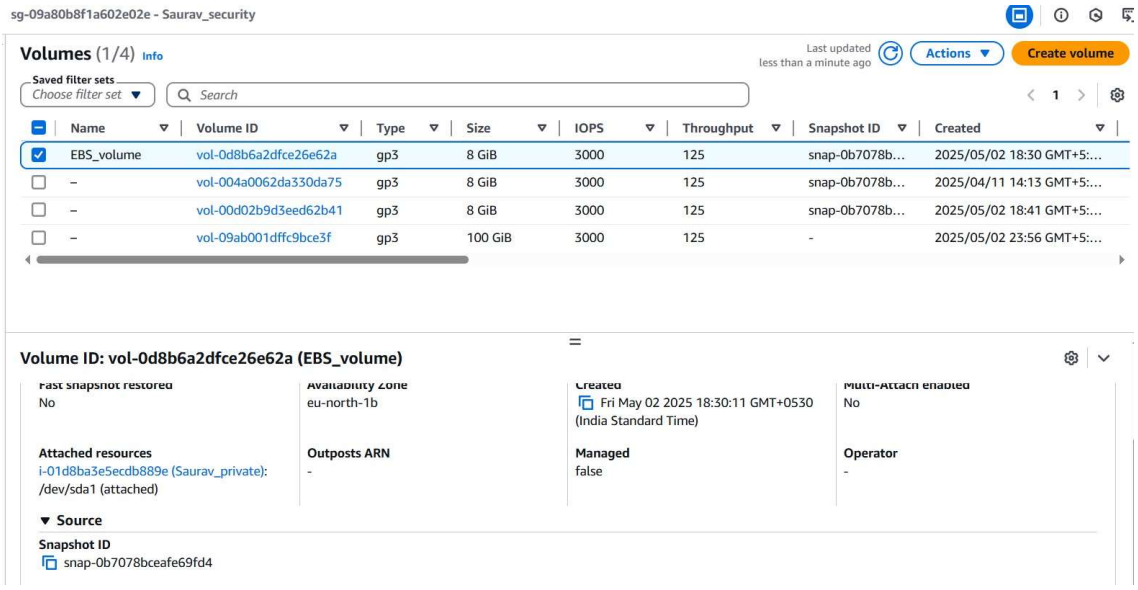


12. *Create and Mount an EBS Volume on EC2*

a. Create a new **Elastic Block Store (EBS) volume**.



b. Attach it to an EC2 instance.



c. Format and mount it on the instance:

sudo mkfs -t ext4 /dev/xvdf

```
ubuntu@ip-12-0-0-150:~$ sudo mkfs -t ext4 /dev/xvdf
mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 26214400 4k blocks and 6553600 inodes
Filesystem UUID: 505189b2-be41-45c8-a98b-739291772bf8
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000, 7962624, 11239424, 20480000, 23887872

Allocating group tables: done
Writing inode tables: done
Creating journal (131072 blocks): done
Writing superblocks and filesystem accounting information: done
```

sudo mkdir /data

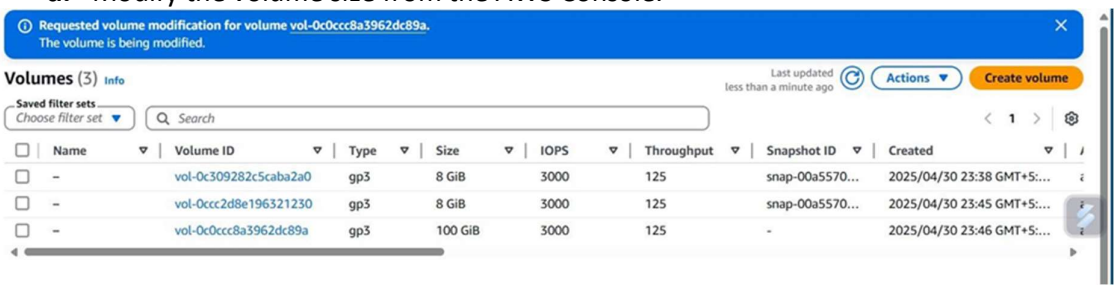
```
Allocating group tables: done
Writing inode tables: done
Creating journal (131072 blocks): done
Writing superblocks and filesystem accounting information: done

ubuntu@ip-12-0-0-150:~$ sudo mkdir /data
ubuntu@ip-12-0-0-150:~$
```

sudo mount /dev/xvdf /data

```
ubuntu@ip-12-0-0-150:~$ sudo mount /dev/xvdf /data
ubuntu@ip-12-0-0-150:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       6.8G  1.7G  5.1G  26% /
tmpfs           479M     0  479M   0% /dev/shm
tmpfs           192M  876K  191M   1% /run
tmpfs           5.0M     0  5.0M   0% /run/lock
/dev/xvda16     881M   79M  741M  10% /boot
/dev/xvda15     105M  6.1M   99M   6% /boot/efi
tmpfs            96M   12K   96M   1% /run/user/1000
/dev/xvdf        98G   24K   93G   1% /data
ubuntu@ip-12-0-0-150:~$
```

13. *Resize EBS Volume and Expand File System*

a. Modify the volume size from the AWS Console.



b. On the EC2 instance, expand the file system:

sudo growpart /dev/xvdf 1

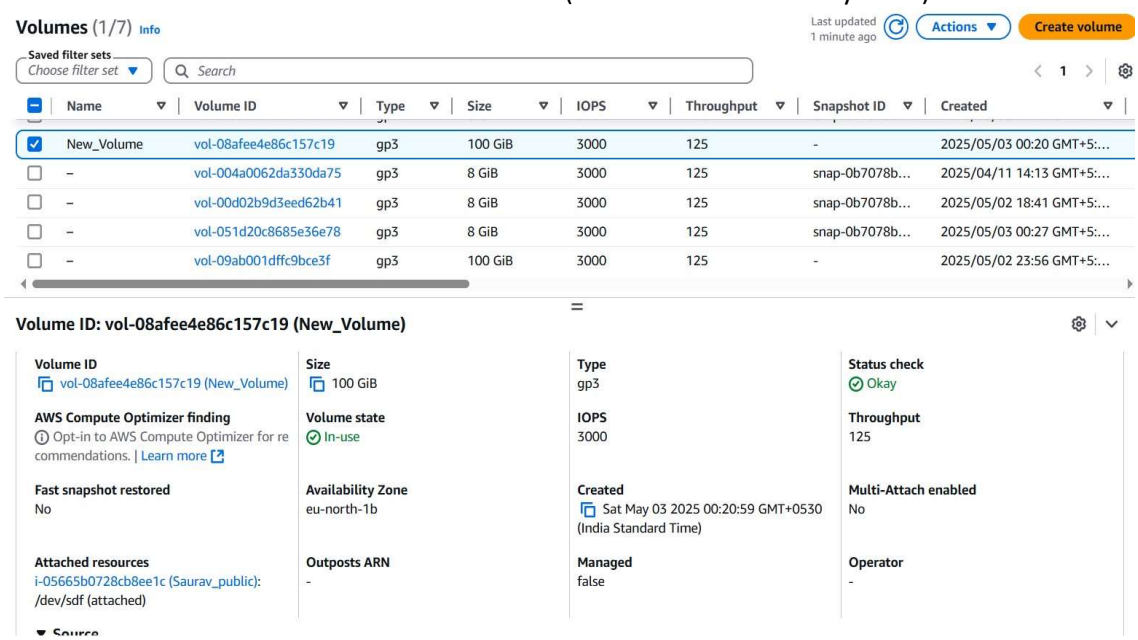
```
ubuntu@ip-12-0-0-150:~$ sudo growpart /dev/xvdf 1
WARN: unknown label
failed [sfd_dump:1] sfdisk --unit=S --dump /dev/xvdf
sfdisk: /dev/xvdf: does not contain a recognized partition table
FAILED: failed to dump sfdisk info for /dev/xvdf
ubuntu@ip-12-0-0-150:~$ ^C
ubuntu@ip-12-0-0-150:~$ ^C
ubuntu@ip-12-0-0-150:~$ sudo resize2fs /dev/xvdf
resize2fs 1.47.0 (5-Feb-2023)
Filesystem at /dev/xvdf is mounted on /data; on-line resizing required
old_desc_blocks = 13, new_desc_blocks = 32
The filesystem on /dev/xvdf is now 65536000 (4k) blocks long.
```

sudo resize2fs /dev/xvdf

```
ubuntu@ip-12-0-0-150:~$ sudo growpart /dev/xvdf 1
WARN: unknown label
failed [sfd_dump:1] sfdisk --unit=S --dump /dev/xvdf
sfdisk: /dev/xvdf: does not contain a recognized partition table
FAILED: failed to dump sfdisk info for /dev/xvdf
ubuntu@ip-12-0-0-150:~$ ^C
ubuntu@ip-12-0-0-150:~$ ^C
ubuntu@ip-12-0-0-150:~$ sudo resize2fs /dev/xvdf
resize2fs 1.47.0 (5-Feb-2023)
Filesystem at /dev/xvdf is mounted on /data; on-line resizing required
old_desc_blocks = 13, new_desc_blocks = 32
The filesystem on /dev/xvdf is now 65536000 (4k) blocks long.
```

14. *Attach One EBS Volume to Multiple EC2 Instances*

a. Attach the same EBS volume to two instances (in the same Availability Zone).

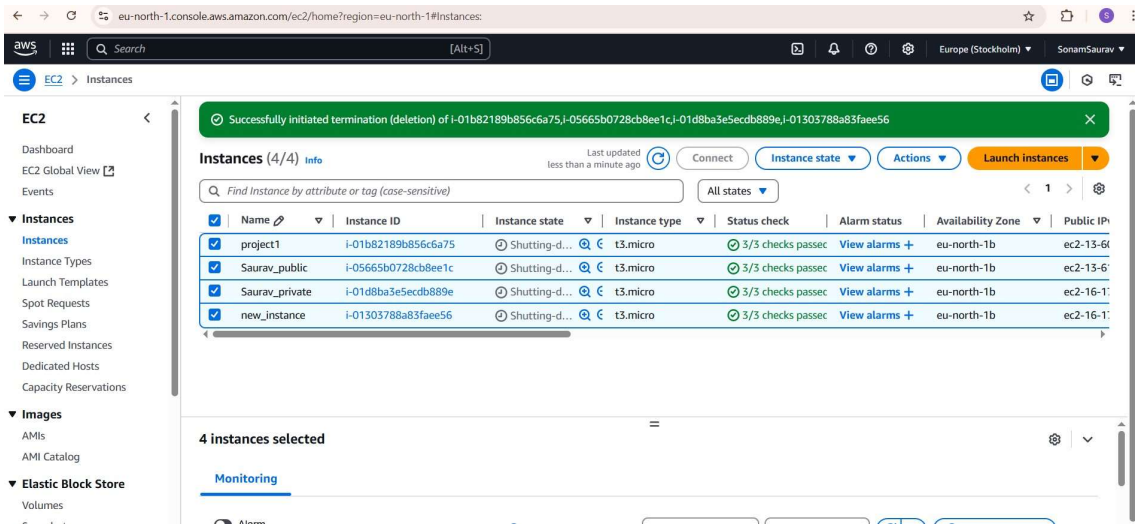


b. Mount in **read-only mode** on one instance to avoid conflicts.

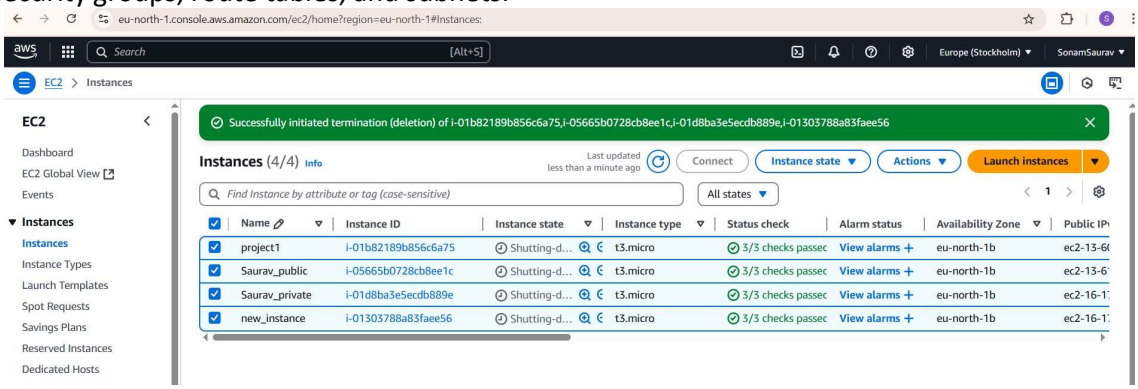
```
ubuntu@ip-12-0-0-150:~$ lsblk
NAME         MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0          7:0    0 26.3M  1 loop /snap/amazon-ssm-agent/9881
loop1          7:1    0 44.4M  1 loop /snap/snapd/23545
loop2          7:2    0 73.9M  1 loop /snap/core22/1748
nvme0n1      259:0    0    8G  0 disk
├─nvme0n1p1  259:2    0    7G  0 part /
├─nvme0n1p14 259:3    0    4M  0 part
├─nvme0n1p15 259:4    0  106M  0 part /boot/efi
└─nvme0n1p16 259:5    0  913M  0 part /boot
nvme1n1      259:1    0  250G  0 disk
nvme2n1      259:6    0  100G  0 disk
ubuntu@ip-12-0-0-150:~$ sudo mkdir /readonly
sudo mount -o ro /dev/nvme1n1 /readonly
mkdir: cannot create directory '/readonly': File exists
ubuntu@ip-12-0-0-150:~$
```

15. Clean Up: Delete EC2 Instances and VPC

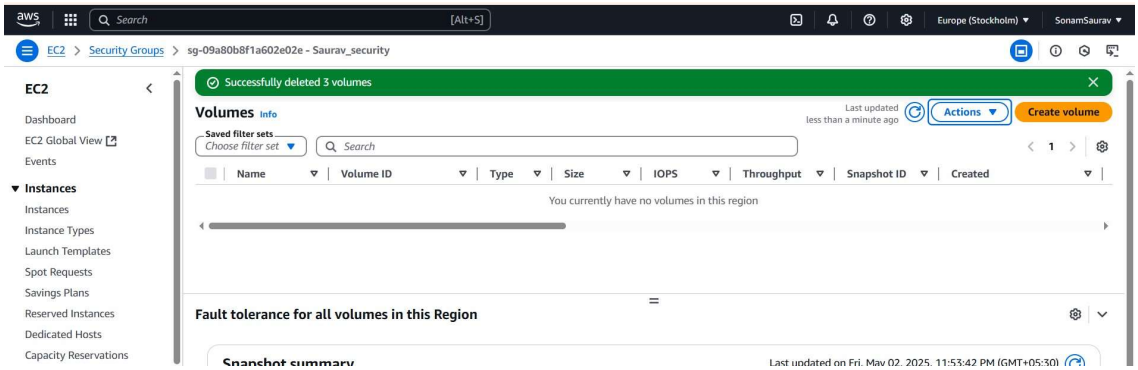
a. Terminate all EC2 instances.



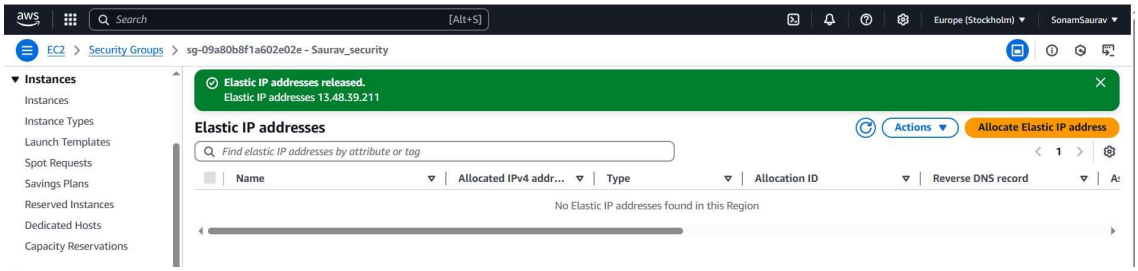
b. Detach and delete associated resources: EBS volumes, Elastic IPs, security groups, route tables, and subnets.



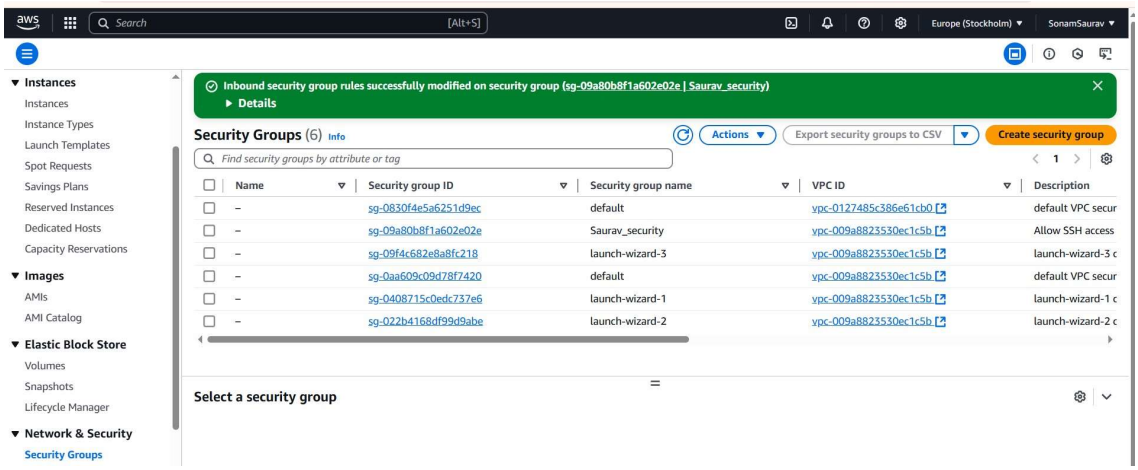
Volumes

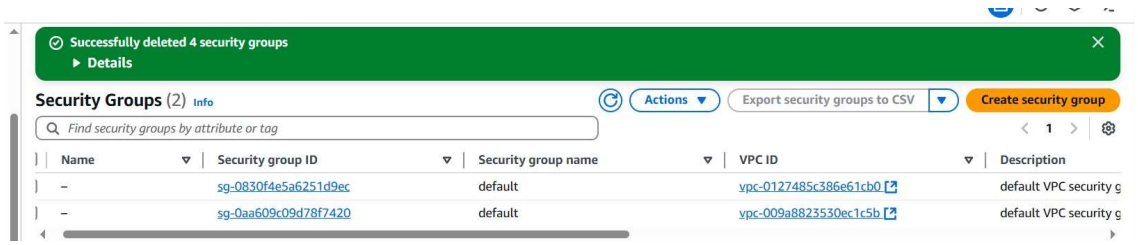


Elastic IP Address

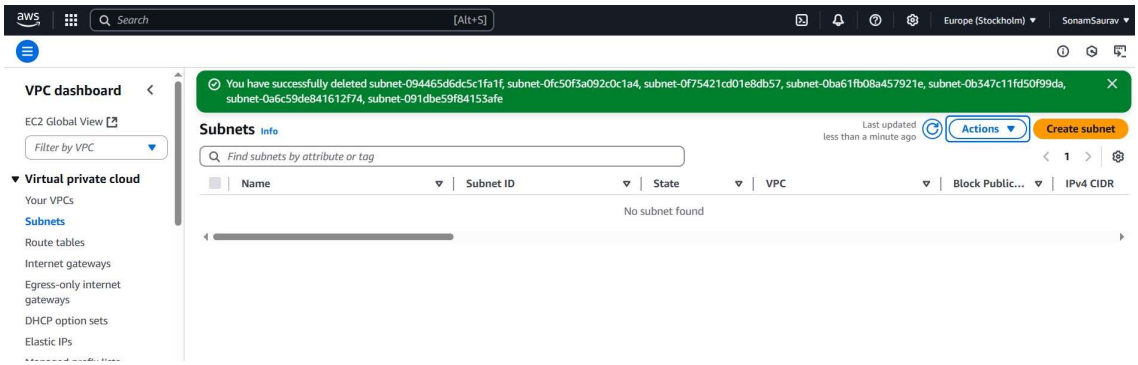


Security Group

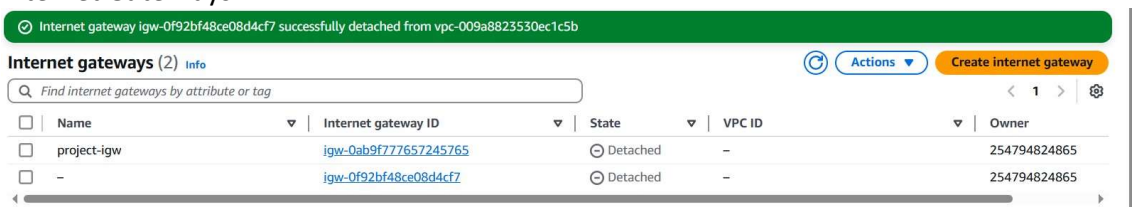


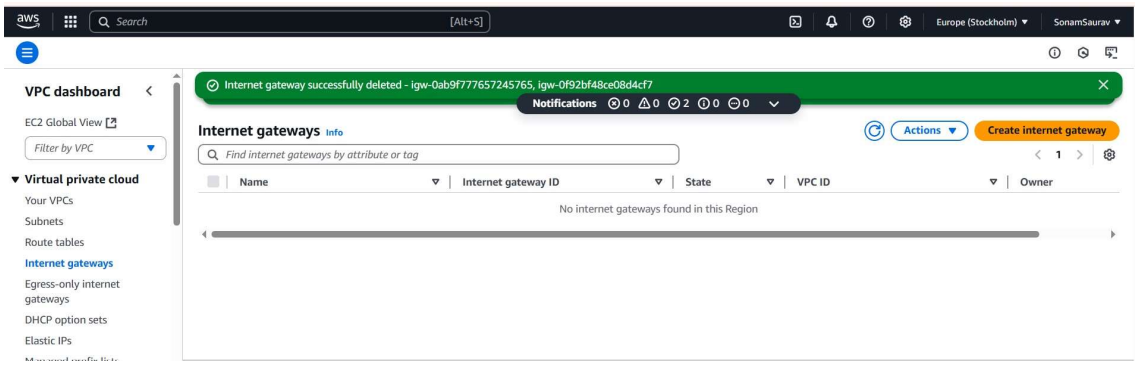


Subnets

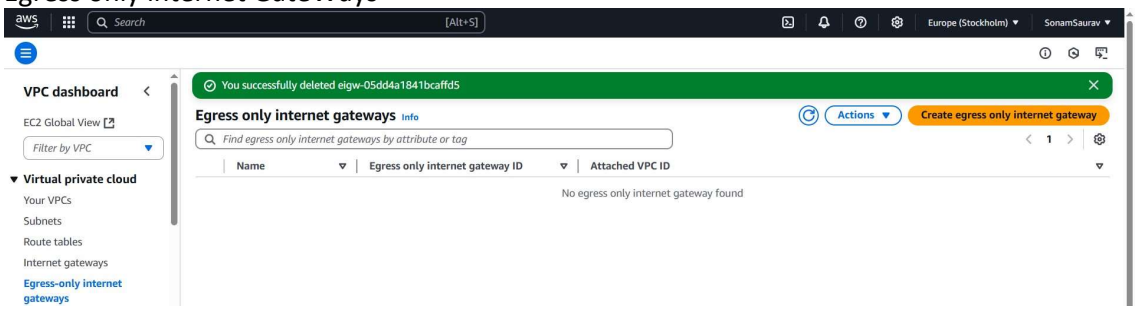


Internet GateWays

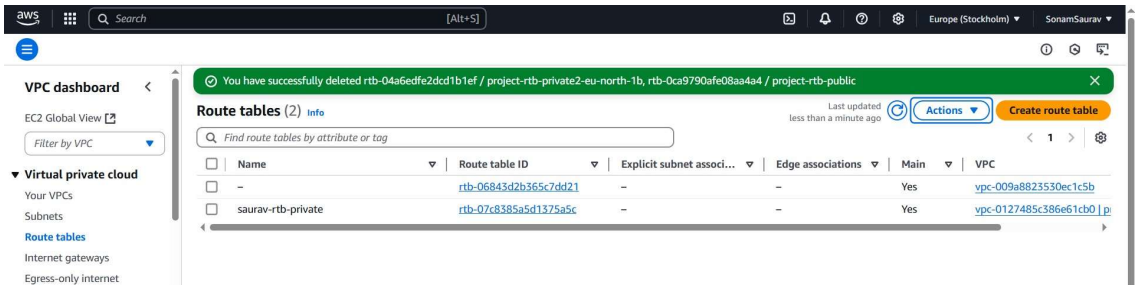


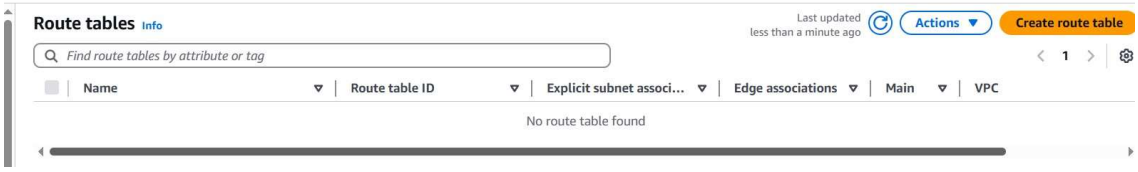


Egress only internet GateWays

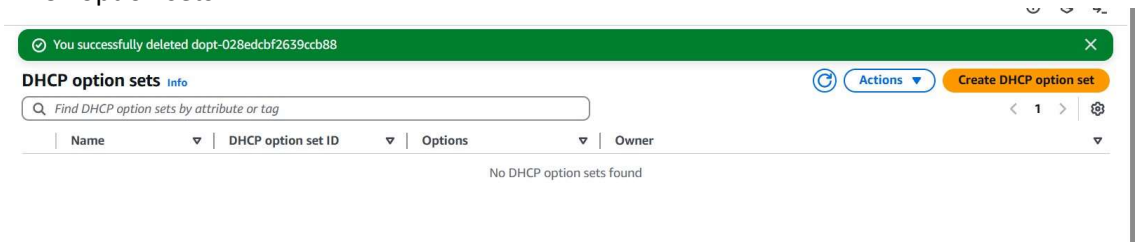
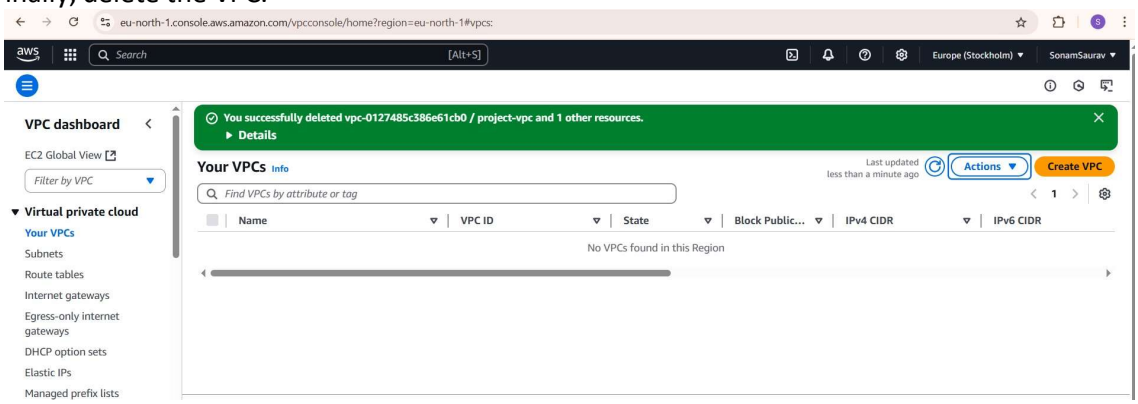


Route Tables





DHCP option sets



c. Finally, delete the VPC.