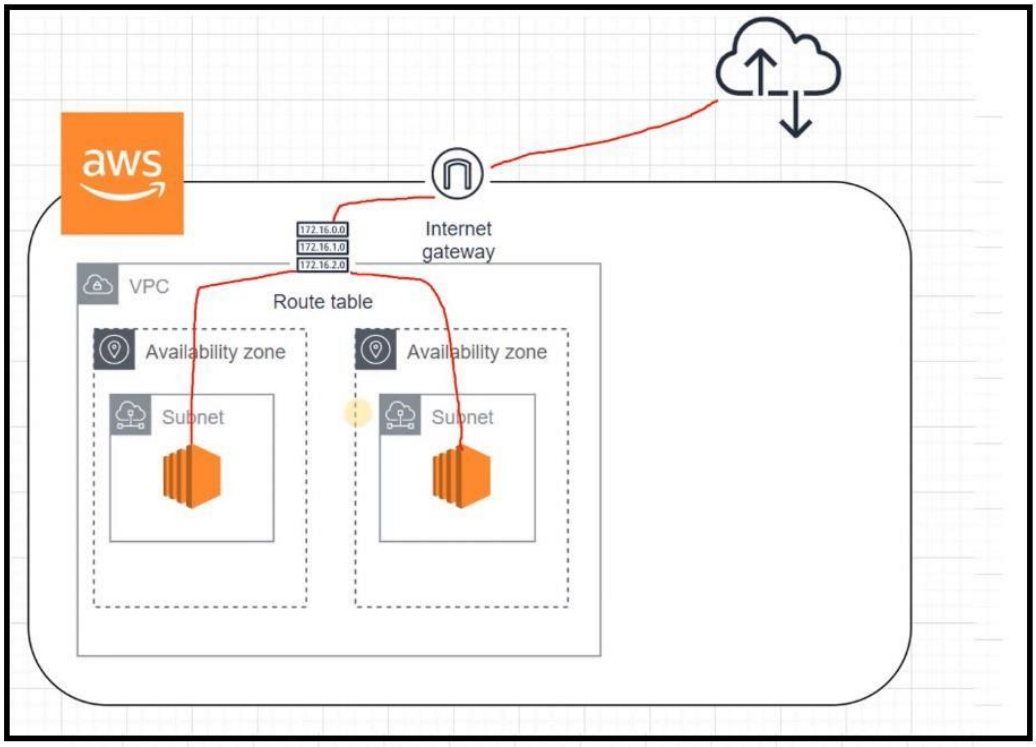
VPC (Virtual Private Cloud)

What is VPC??

A Virtual Private Cloud (VPC) is like your own private, isolated section of the internet in the cloud. It's a virtual network that you can set up in a cloud computing environment (like AWS, Azure, or Google Cloud) to run your applications and services.

Imagine it as creating your own little corner of the internet where you can place your servers, databases, and other resources. With a VPC, you have control over the network settings, like IP addresses, subnets, and routing. It helps you keep your resources secure and organized, and you can even connect your VPC to the internet or other VPCs if needed.



What is Subnet??

Subnet is a range of IP address In Your VPC. With The help of subnet we can identify which portion of IP is Network portion and which portion is Host Portion.

Route Table??

A Route table contains a set of rules, called routes that are used to determine where network traffic from your VPC is directed.

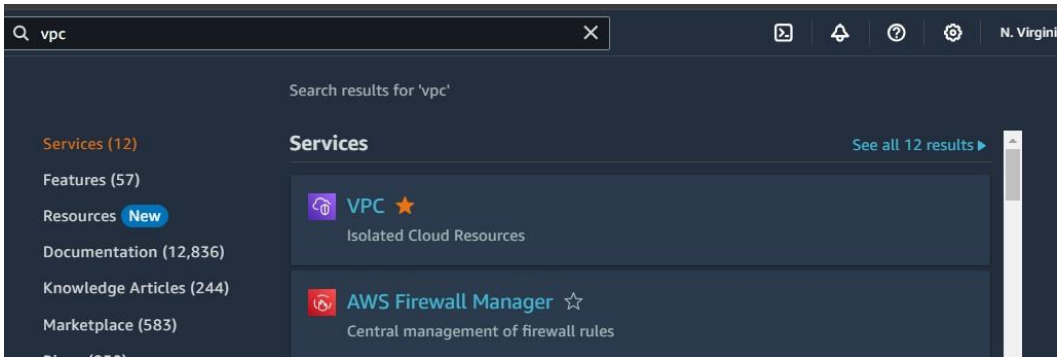
With The help of route table we can communicate with two sub networks.

Internet gateway (IGW)??

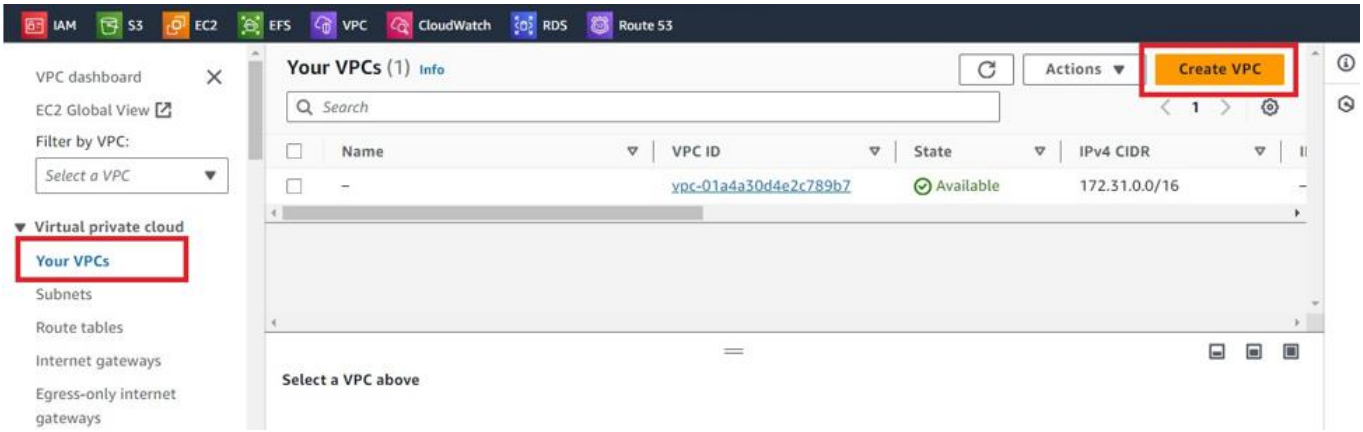
An Internet gateway enables your instances to connect to the internet.

Creating Our Own VPC:-

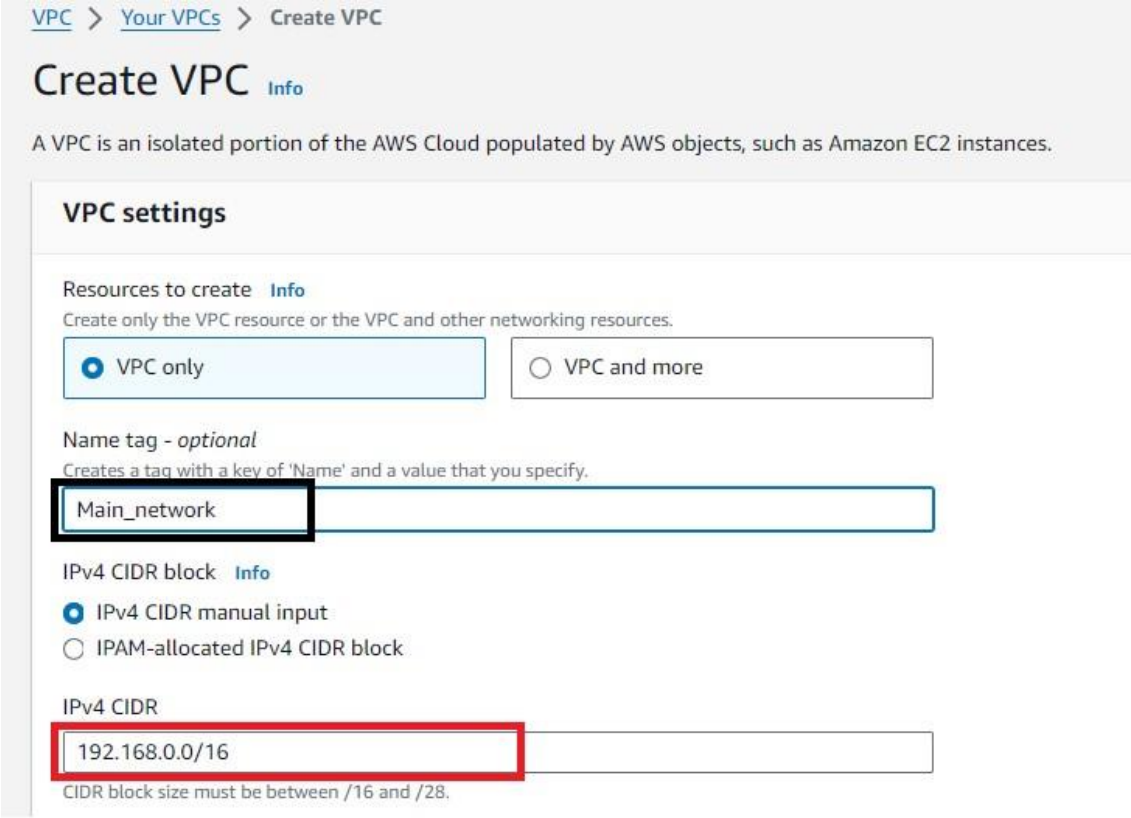
1. Search VPC service in AWS Dashboard



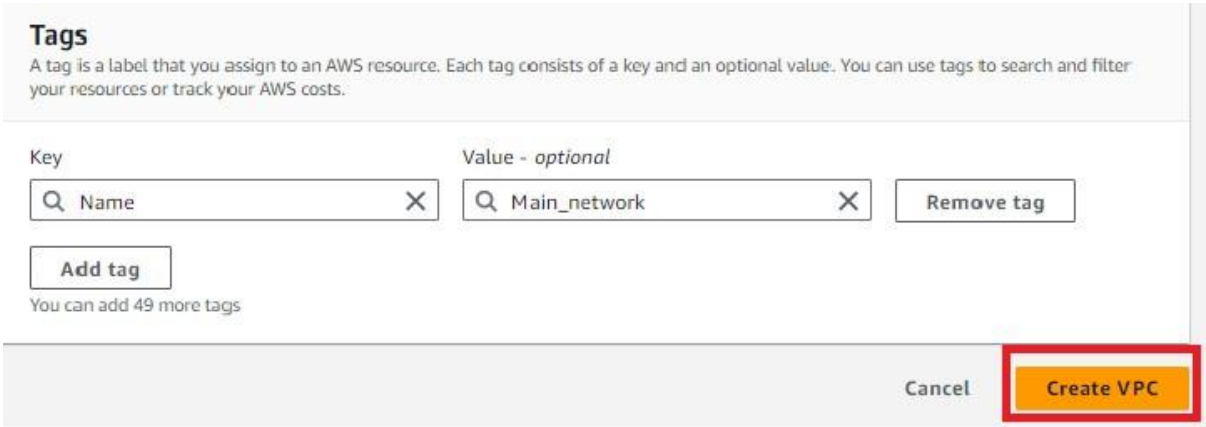
2. Click on create VPC



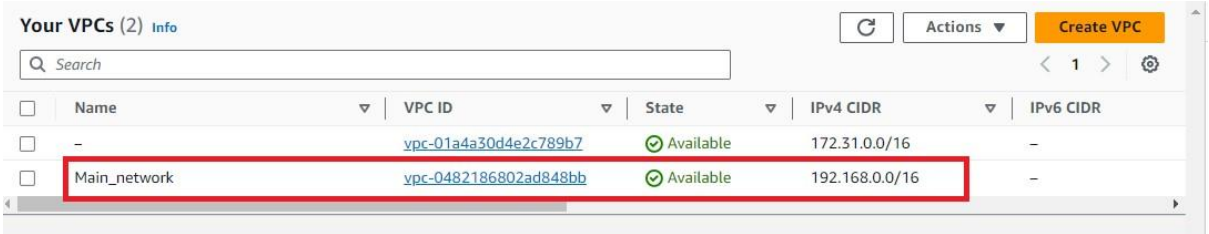
3. Assign Name And **main network Range (CIDR)**



4. Click on Create Network

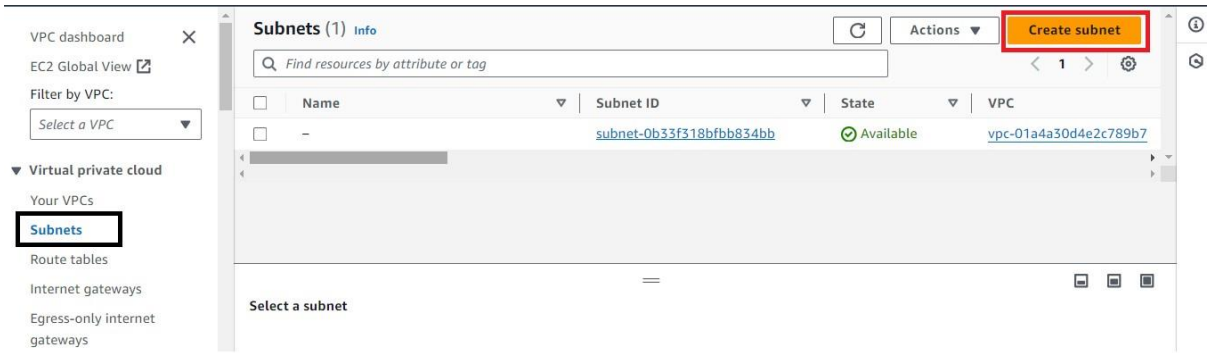


5. **VPC Created successfully……**

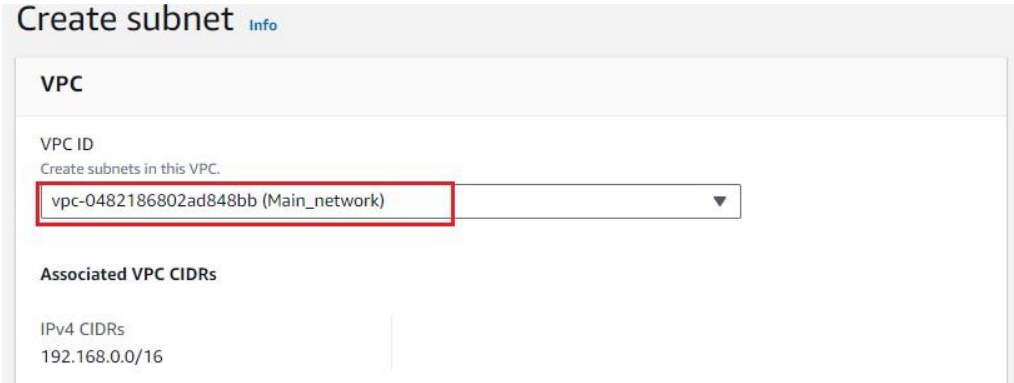


Creating subnets For Our VPC:-

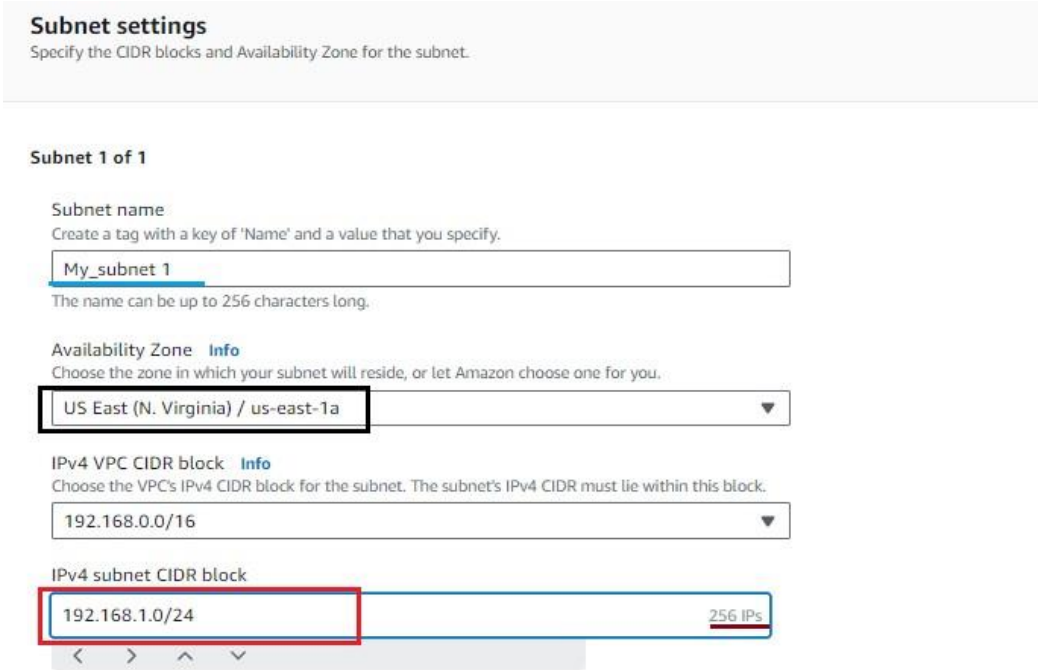
1. Click on Create Subnet



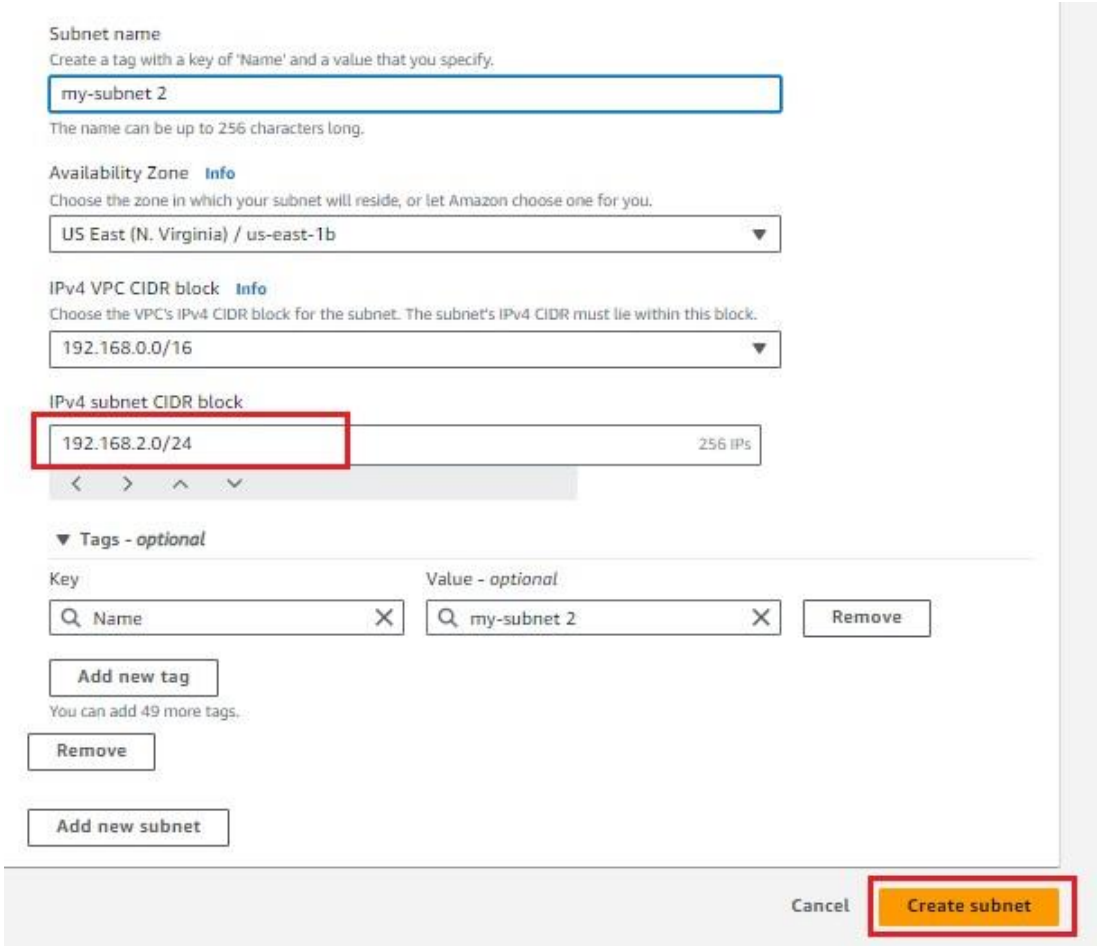
2. Select The VPC Name



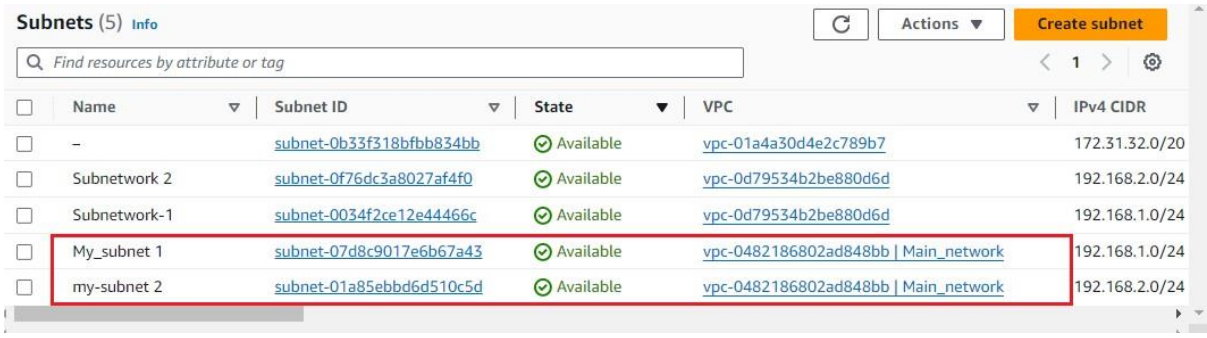
3. Select the **Subnet Range**, **Availability Zone**, **Subnet name**



4. Add Subnets as per your requirements and click on Create **subnet….**



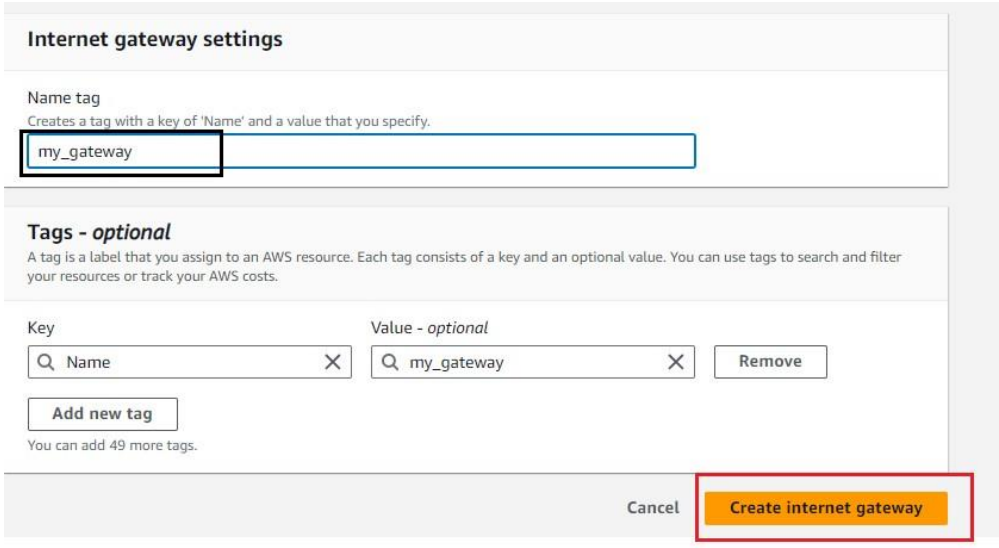
5. **Subnet Added Successfully……**



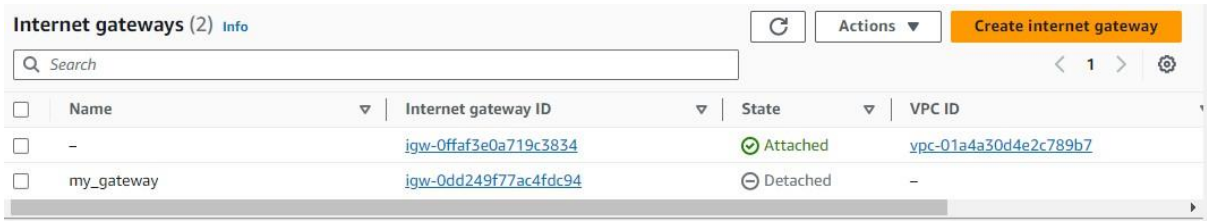
Creating Internet Gateway for VPC:

1. Click on **Create internet gateway**

2. Specify The Name As per your choice

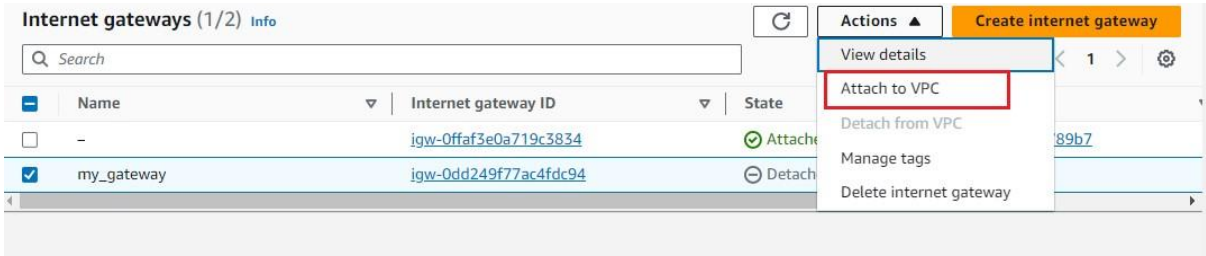


3. **Internet Gateway created successfully….**

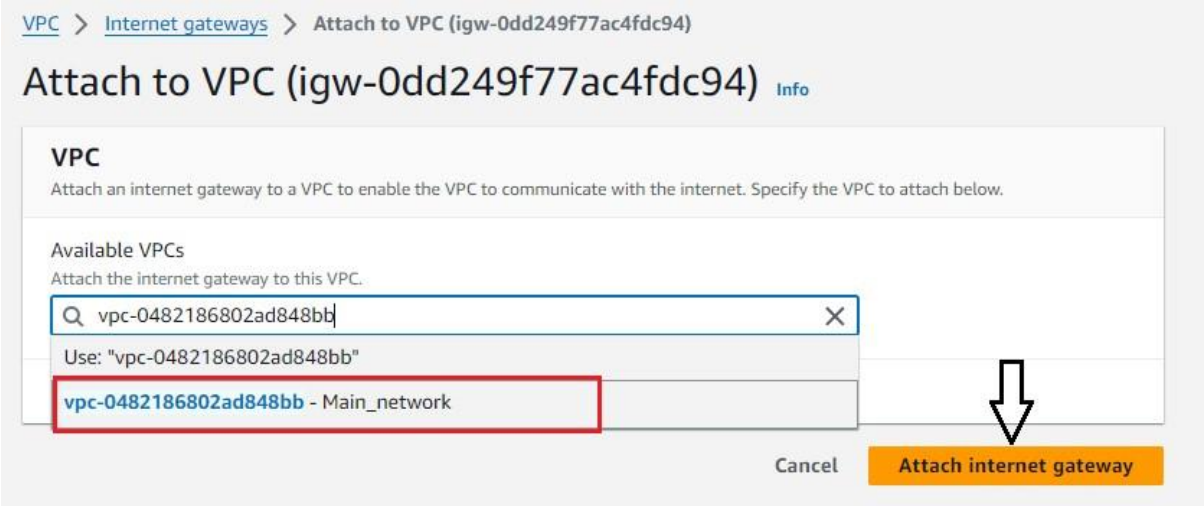


Attach Internet Gateway TO VPC:

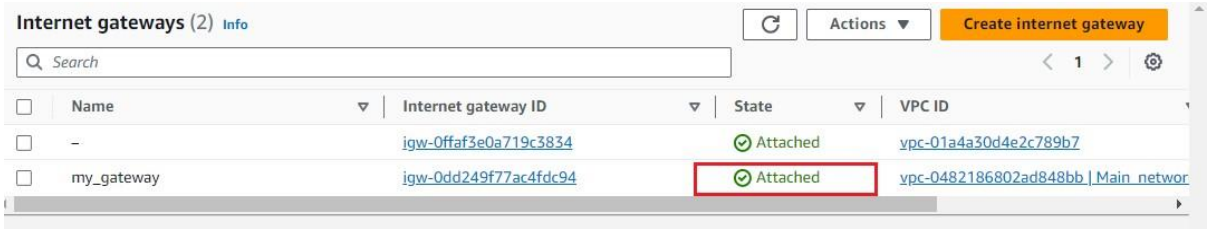
1. Select The Internet Gateway And click on **Attach to VPC** option



2. Select The VPC and click on **Attach internet gateway** option



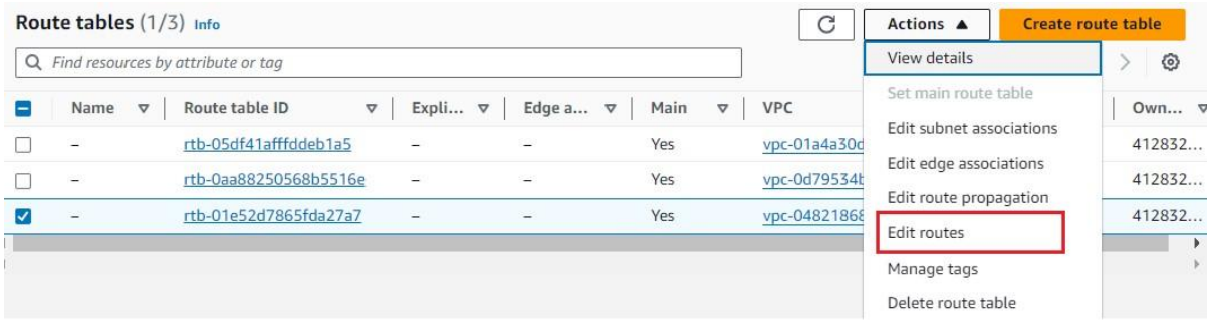
3. **Internet Gateway Attached successfully….**



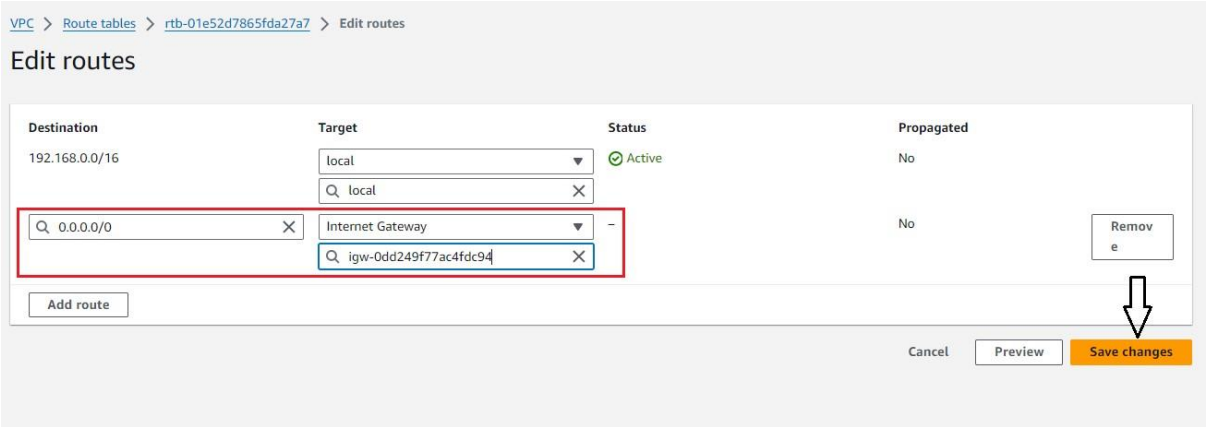
Configuring the Route Tables:

Note: - (After Creating VPC the Route Table is created automatically)

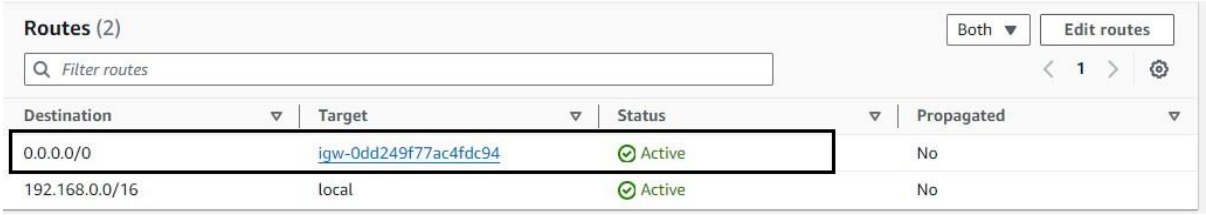
1. Select The Route Table And click on **Edit routes**



2. Select the Internet Gateway and Destination (0.0.0.0/0) **means all ips** can access to the internet…..

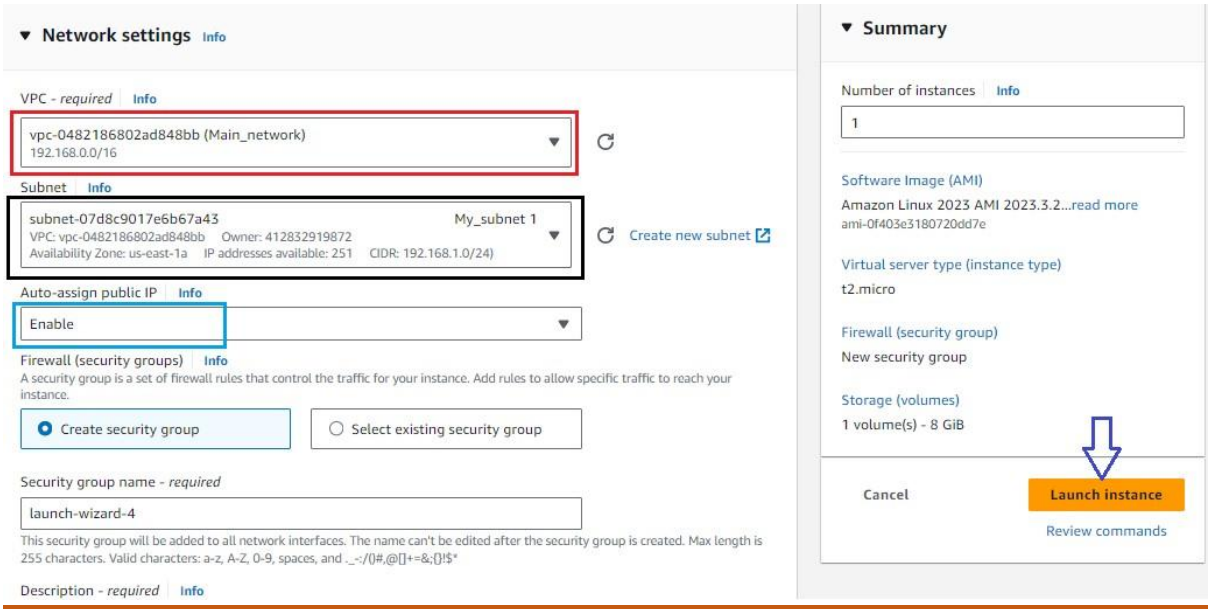


3. **Route Table Configured successfully…**



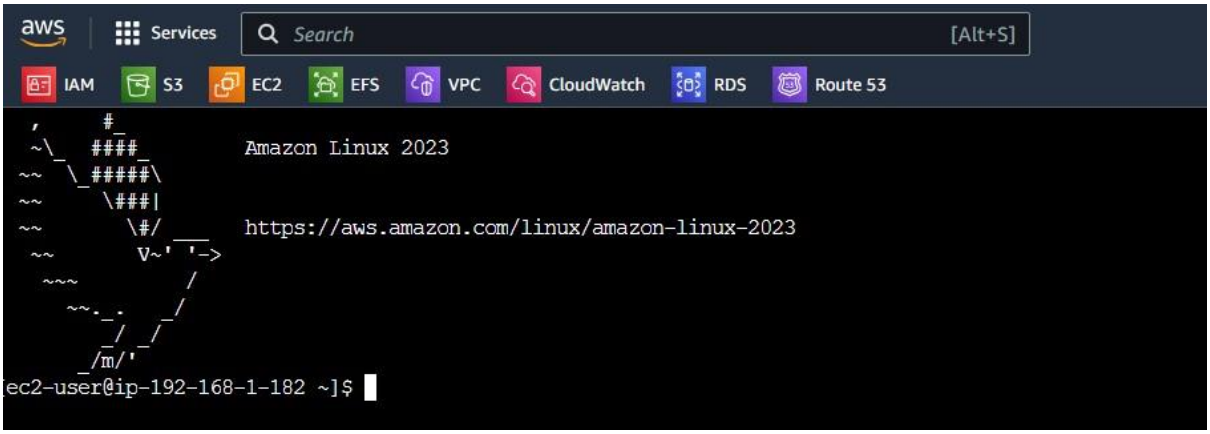
Launch Instance in Private VPC

1. Click on Launch Instance
2. Select Ami
3. Select Instance Type
4. Key pair
5. Network Settings (select VPC name, select subnet , And select Enable Auto-assign public IP)



6. Click On launch Instance

7. **We successfully launch our Instance in Private VPC…**



Creating Public And Private Subnets:-

Public Subnets:

- Web Servers: Public subnets are commonly used for hosting web servers or any other services that need to be directly accessible from the internet. These resources are assigned public IP addresses and can handle incoming requests from users or clients over the internet.
- Load Balancers: Load balancers, which distribute incoming network traffic across multiple servers to ensure no single server is overwhelmed, are often placed in public subnets to efficiently handle internet-facing traffic.
- Content Delivery Networks (CDNs): CDNs, which cache and distribute content globally to improve performance, may use resources in public subnets to ensure efficient delivery of content to end-users.

Private Subnets:

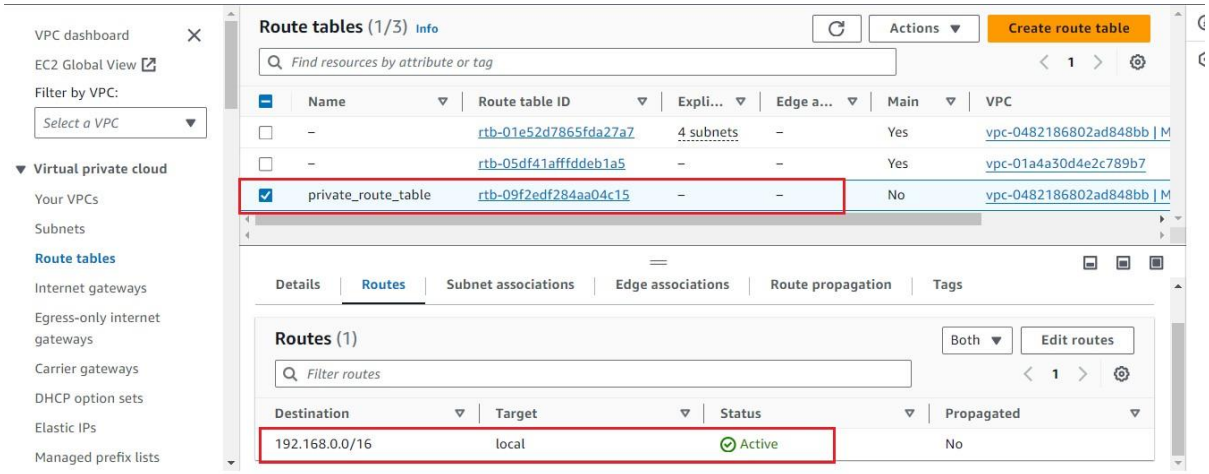
- Databases: Private subnets are suitable for hosting databases or other data storage systems that don't need direct internet access. This provides an additional layer of security by isolating sensitive data from direct exposure to the internet.
- Application Servers: Backend application servers that process business logic or handle sensitive transactions can be placed in private subnets. This helps protect them from direct access by external entities while allowing controlled communication with other components.
- Internal Services: Services that are used internally within an organization and don't need to be accessed from the internet can be deployed in private subnets. This includes various backend services and internal communication mechanisms.
- Secure File Storage: Private subnets are suitable for storing sensitive files or documents that should not be directly accessible from the internet.

How to create public Subnets:-

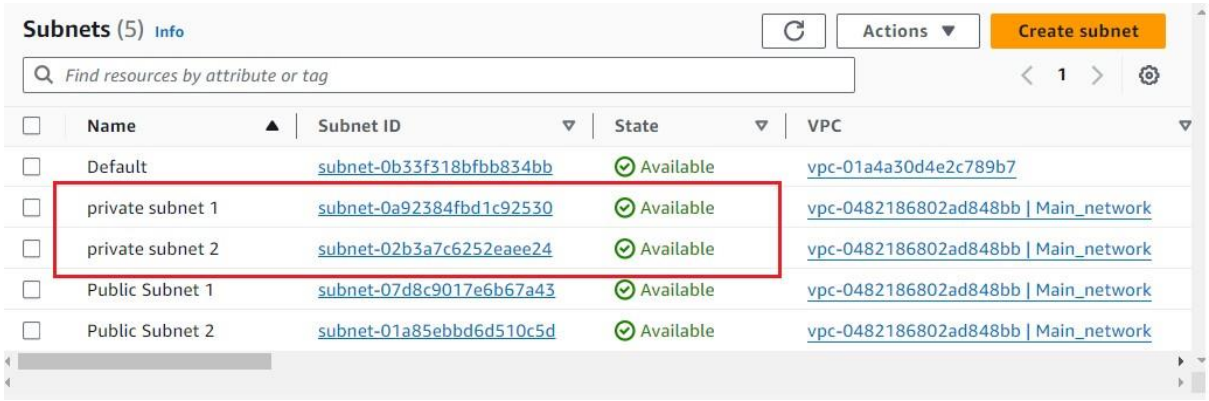
1. **Create subnets**
2. **Assign Internet Gateway to VPC**
3. **Assign Internet Gateway to Route Table**

How to Create Private Subnets:-

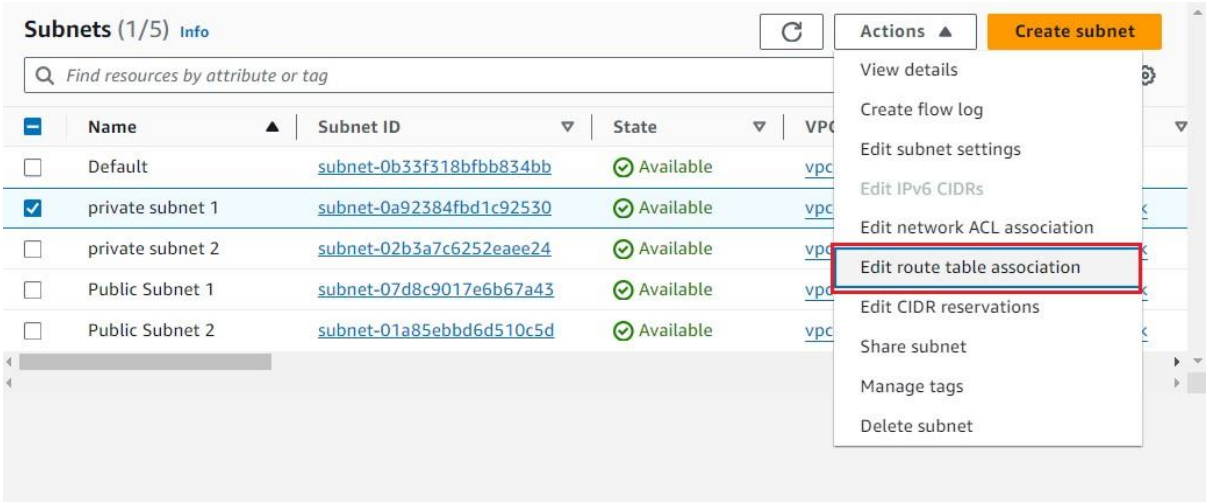
1. Create route Table And do not assign any Internet gateway To that Route Table (Remain as it is)



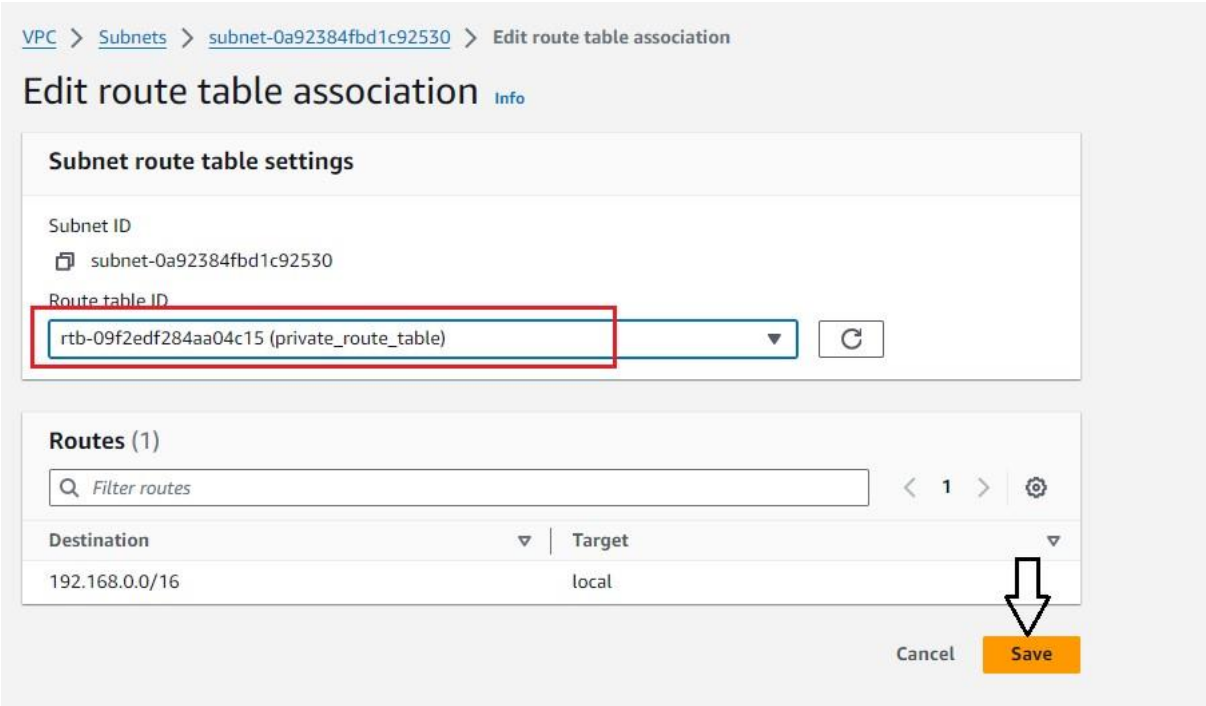
2. Create subnet



3. Select The subnet and click on edit route table association



4. Select The private route table (which we haven't give public access) and click on save option

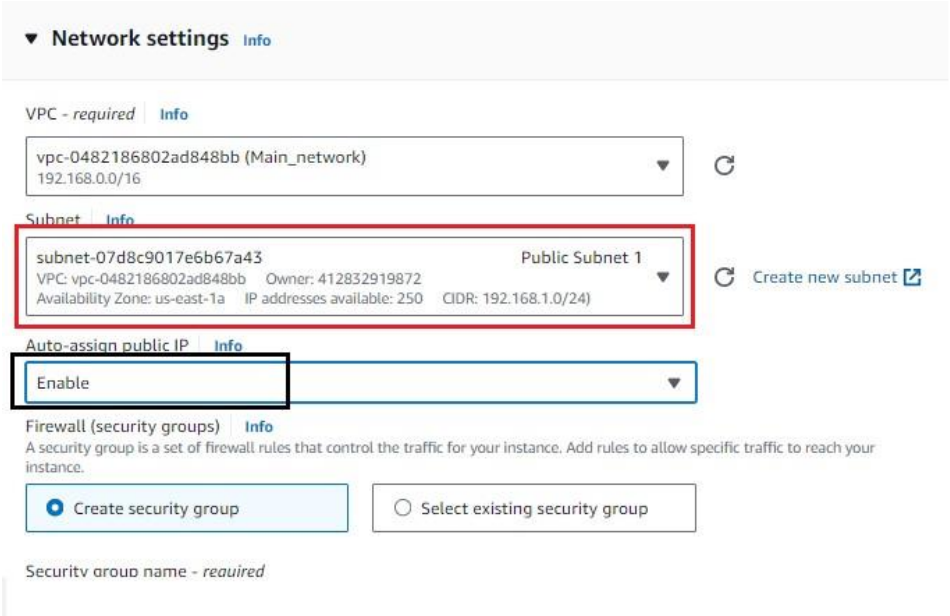


5. **Private subnets configuration done successfully….**

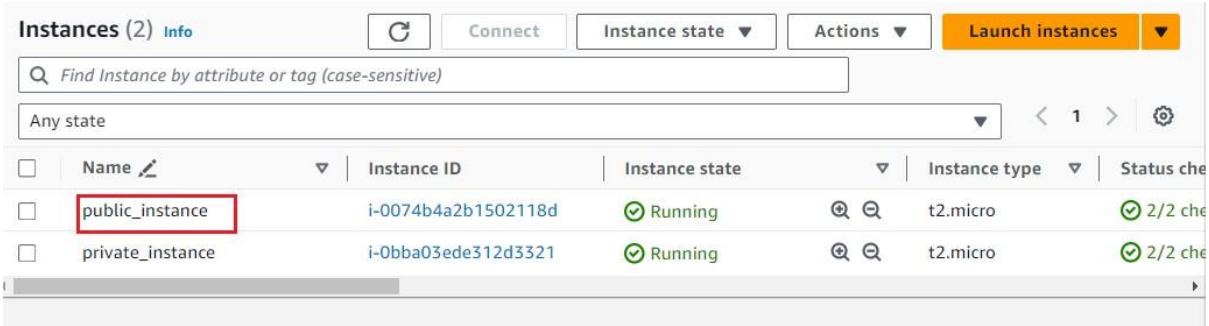
Internally Pinging Private and public Instance:-

Create public instance:-

1. While creating instance select the **public subnet** (which we have created)

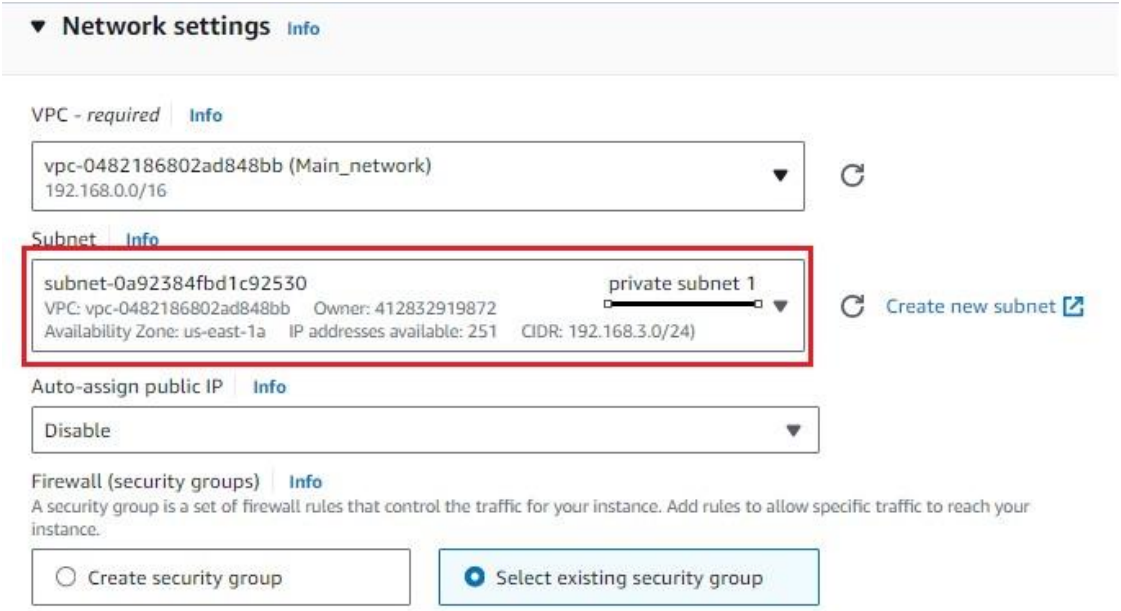


Public Instance Created successfully….

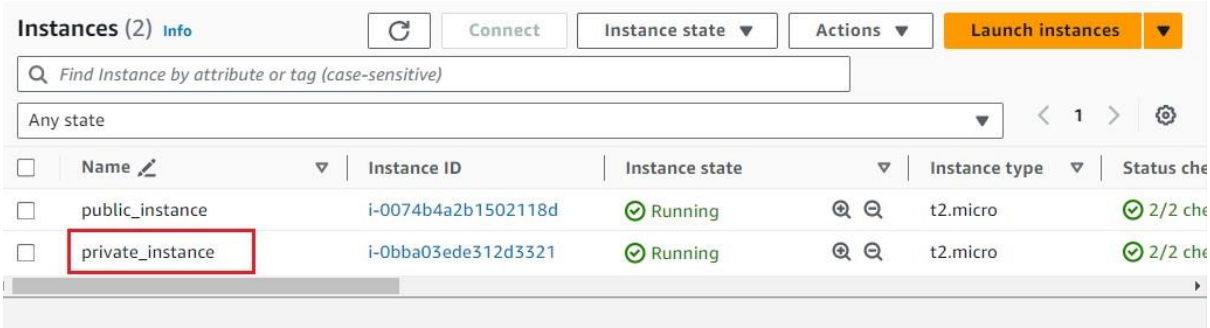


Create private instance:-

1. While Creating Instance select the private subnet (which we have created)

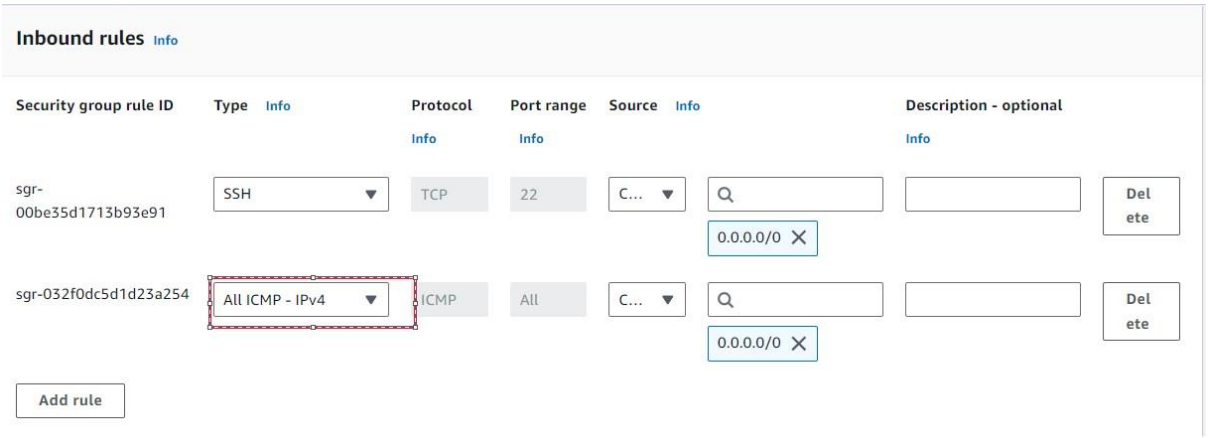


2. **Private Instance Created successfully….**



Pinging Private and public Instances:-

Add **ALL ICMP – IPV4** rule in security Group…



After Adding Rule in both (public and private) instances we can able to ping each other…..

```
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Sun Mar 10 18:51:59 2024 from 18.206.107.27
[ec2-user@ip-192-168-1-182 ~]$ ping  192.168.4.212
PING 192.168.4.212 (192.168.4.212) 56(84) bytes of data.
64 bytes from 192.168.4.212: icmp_seq=1 ttl=127 time=1.48 ms
64 bytes from 192.168.4.212: icmp_seq=2 ttl=127 time=1.09 ms
64 bytes from 192.168.4.212: icmp_seq=3 ttl=127 time=1.09 ms
^C
--- 192.168.4.212 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 1.089/1.220/1.482/0.185 ms
[ec2-user@ip-192-168-1-182 ~]$
```

Getting ssh (Public to private instance…)

1. **Forward or copy** your private instance key (private key) in public instance

```
[root@ip-192-168-1-182 ec2-user]# vim private.pem
```

2. Paste your Private instance private key….

3. Change the permission (**chmod 600 private.pem**)

4. Connect to the instance

   Ssh –i private.pem ec2-user@<your private mahine ip>

5. **We successfully able to access the private instance…..**

```
[root@ip-192-168-1-182 ec2-user]# ssh -i  private.pem  ec2-user@192.168.4.212^C
[root@ip-192-168-1-182 ec2-user]# ssh -i  private.pem  ec2-user@192.168.4.212
       #_
       ~\_  ####_        Amazon Linux 2023
      ~~  \_#####\
      ~~     \###|
      ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
       ~~       V~' '->
        ~~~         /
          ~~._.   _/
             _/ _/
           _/m/'
Last login: Sun Mar 10 20:29:50 2024 from 192.168.1.182
[ec2-user@ip-192-168-4-212 ~]$
```