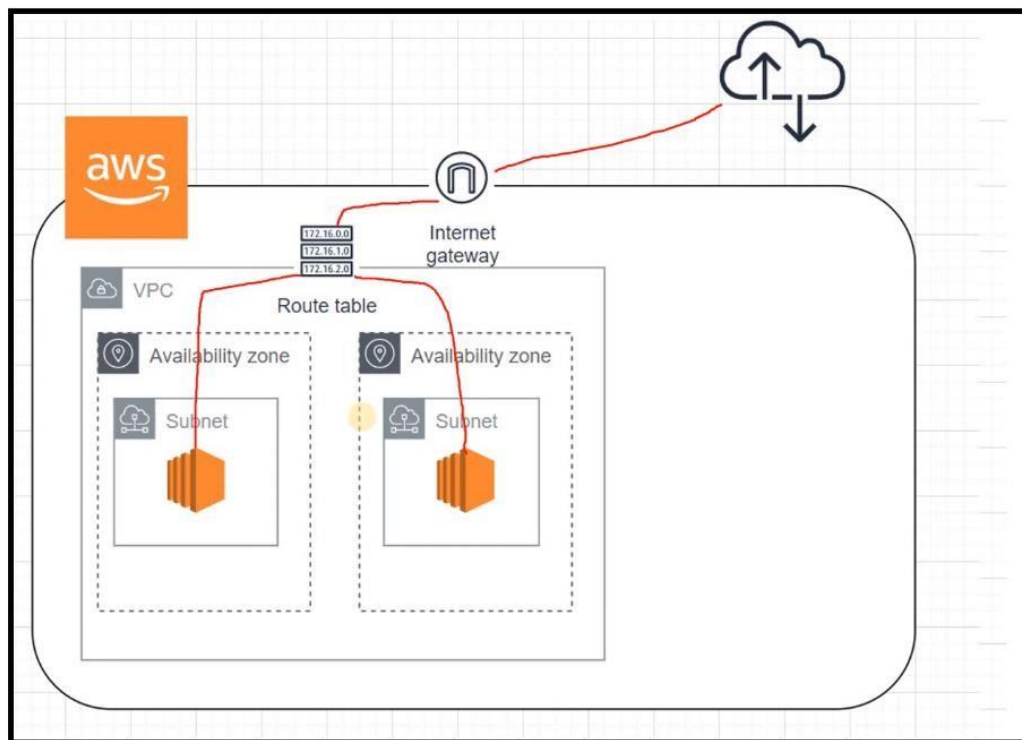


VPC (Virtual Private Cloud)

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.

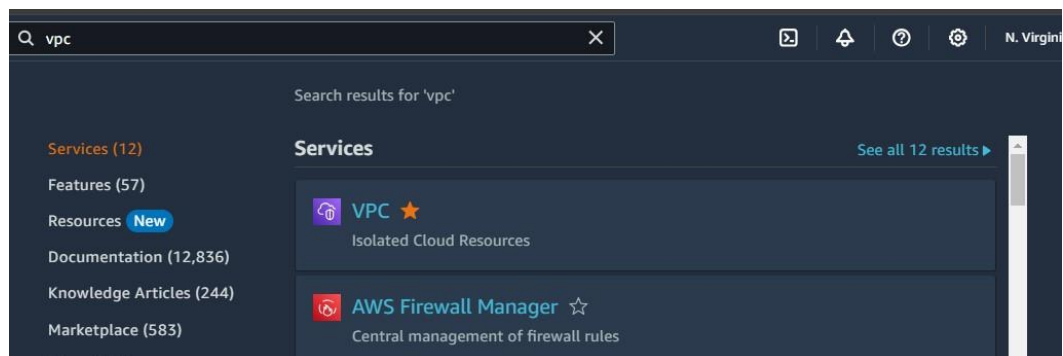
With The help of route table we can communicate with two sub networks.

Internet gateway (IGW)??

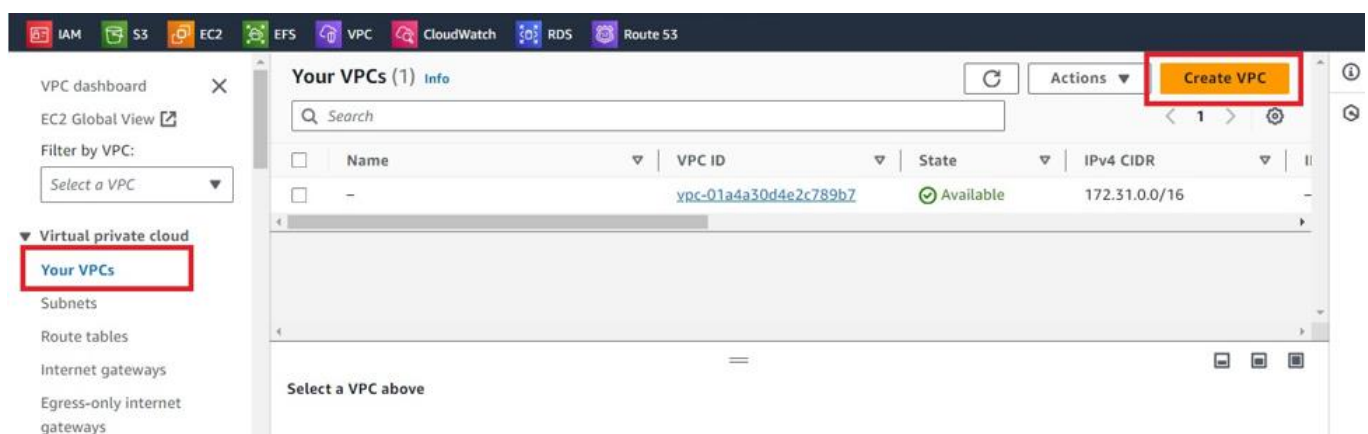
An Internet gateway enables your instances to connect to the internet.

Creating Our Own VPC:-

1. Search VPC service in AWS Dashboard



2. Click on create VPC



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

VPC > Your VPCs > Create VPC

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Resources to create Info
Create only the VPC resource or the VPC and other networking resources.

☒ VPC only ☐ VPC and more

Name tag - optional
Creates a tag with a key of 'Name' and a value that you specify.

Main_network

IPv4 CIDR block Info
☒ IPv4 CIDR manual input ☐ IPAM-allocated IPv4 CIDR block

IPv4 CIDR
192.168.0.0/16

CIDR block size must be between /16 and /28.

4. Click on Create Network

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key Value - optional

You can add 49 more tags

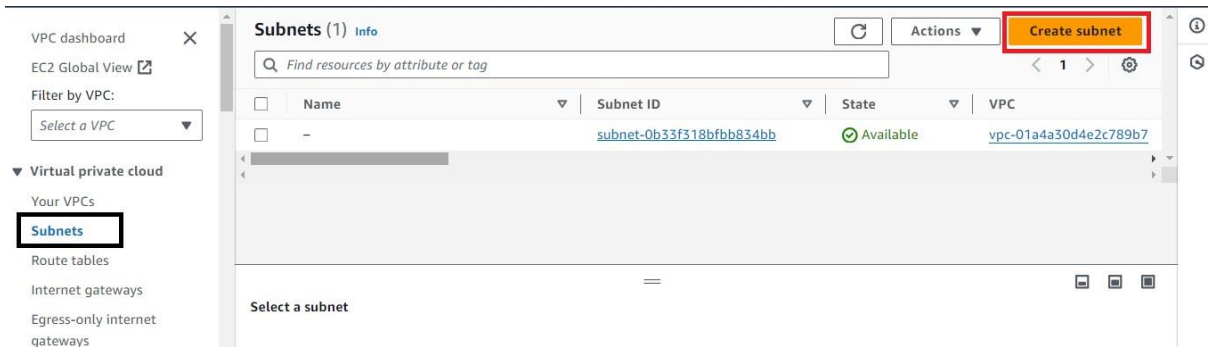
5. VPC Created successfully.....

Your VPCs (2) Info

<input type="checkbox"/>	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
<input type="checkbox"/>	-	vpc-01a4a30d4e2c789b7	Available	172.31.0.0/16	-
<input type="checkbox"/>	Main_network	vpc-0482186802ad848bb	Available	192.168.0.0/16	-

Creating subnets For Our VPC:-

1. Click on Create Subnet



2. Select The VPC Name



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

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 VPC CIDR block [Info](#)
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

IPv4 subnet CIDR block
 256 IPs

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

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 VPC CIDR block [Info](#)
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

IPv4 subnet CIDR block
 256 IPs

▼ Tags - optional

Key	Value - optional	
<input type="text" value="Name"/>	<input type="text" value="my-subnet 2"/>	<input type="button" value="Remove"/>

You can add 49 more tags.

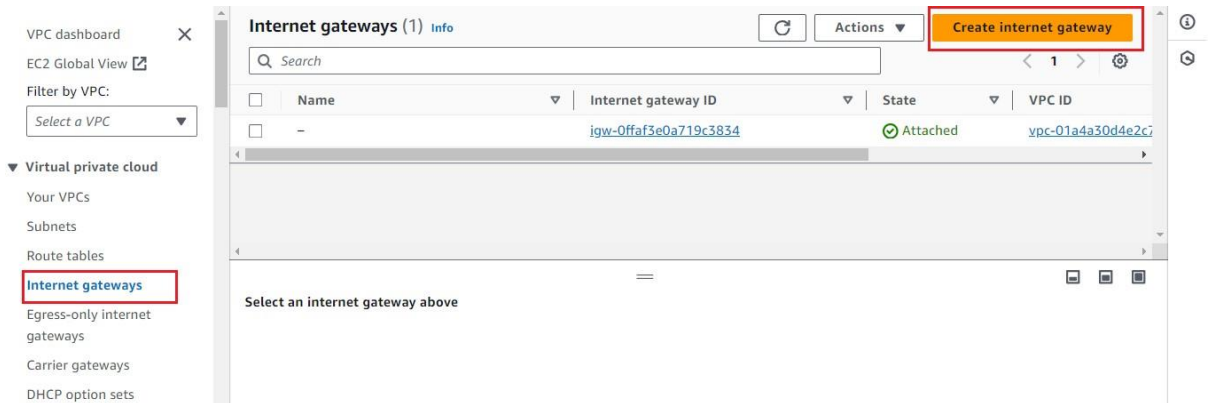
5. Subnet Added Successfully.....

Subnets (5) [Info](#)

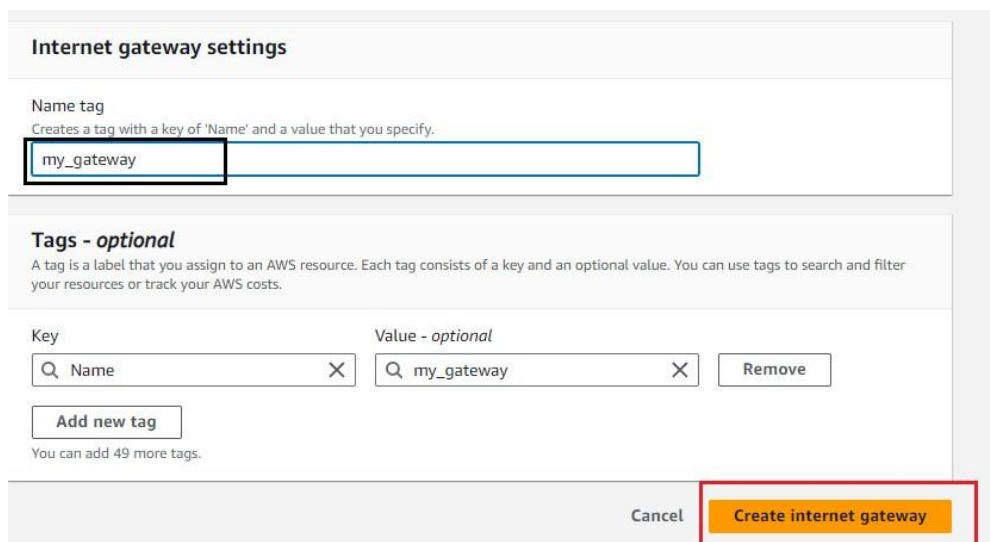
<input type="checkbox"/>	Name	Subnet ID	State	VPC	IPv4 CIDR
<input type="checkbox"/>	-	subnet-0b33f318bfb834bb	Available	vpc-01a4a30d4e2c789b7	172.31.32.0/20
<input type="checkbox"/>	Subnetwork 2	subnet-0f76dc3a8027af4f0	Available	vpc-0d79534b2be880d6d	192.168.2.0/24
<input type="checkbox"/>	Subnetwork-1	subnet-0034f2ce12e44466c	Available	vpc-0d79534b2be880d6d	192.168.1.0/24
<input type="checkbox"/>	My_subnet 1	subnet-07d8c9017e6b67a43	Available	vpc-0482186802ad848bb Main_network	192.168.1.0/24
<input type="checkbox"/>	my-subnet 2	subnet-01a85ebbd6d510c5d	Available	vpc-0482186802ad848bb Main_network	192.168.2.0/24

Creating Internet Gateway for VPC:

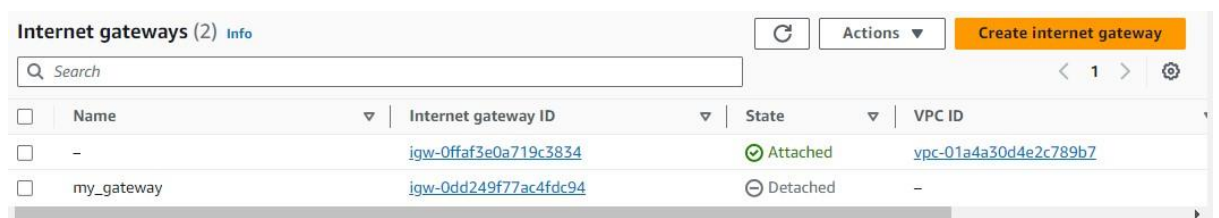
1. Click on **Create internet gateway**



2. Specify The Name As per your choice

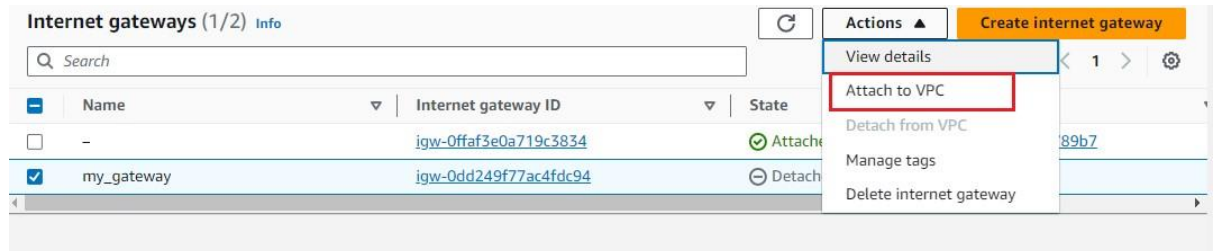


3. Internet Gateway created successfully....

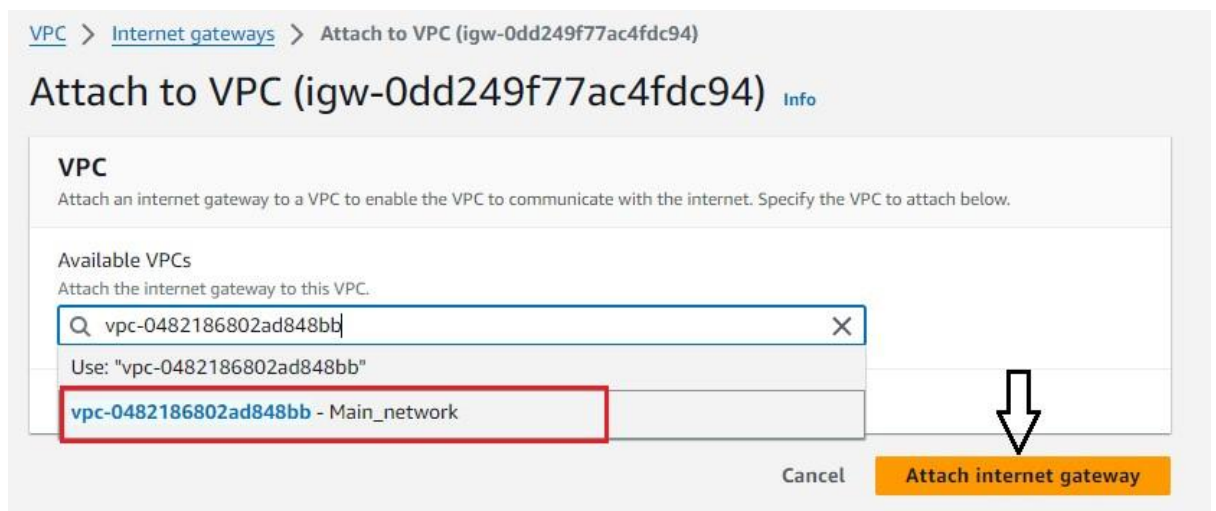


Attach Internet Gateway TO VPC:

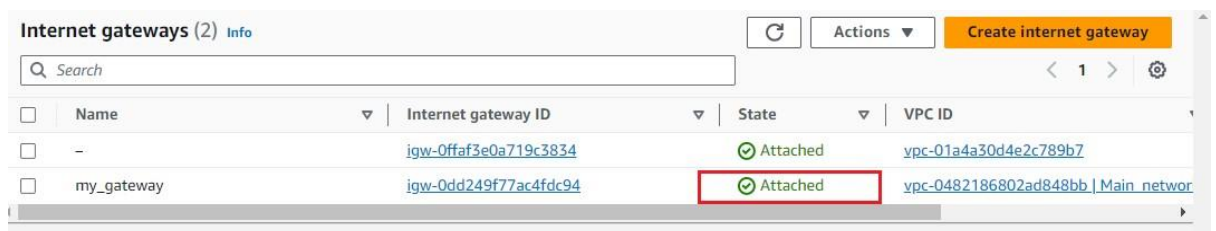
1. Select The Internet Gateway And click on **Attach to VPC** option



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



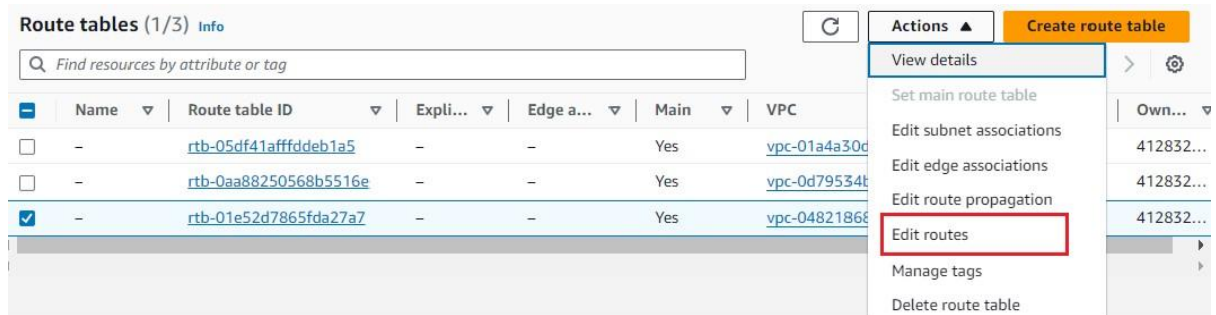
3. Internet Gateway Attached successfully....



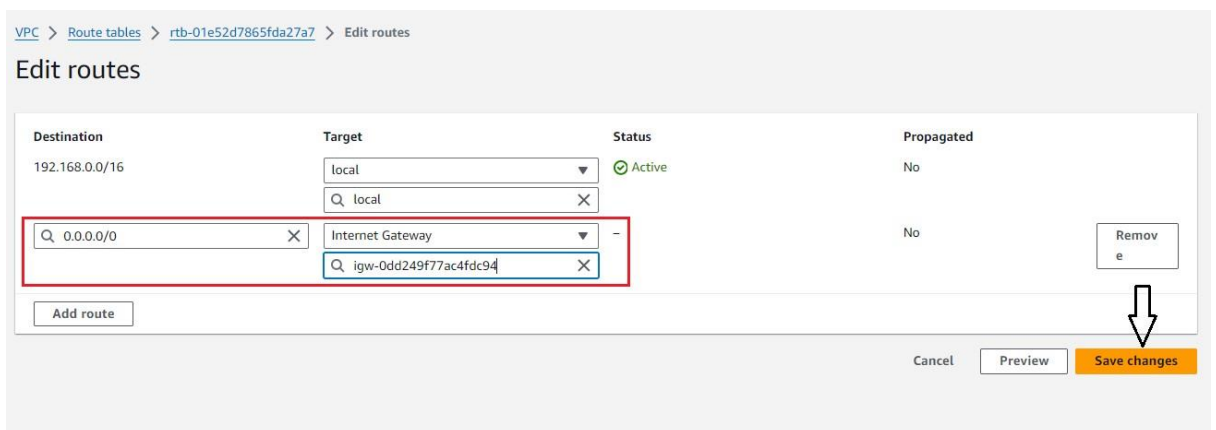
Configuring the Route Tables:

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

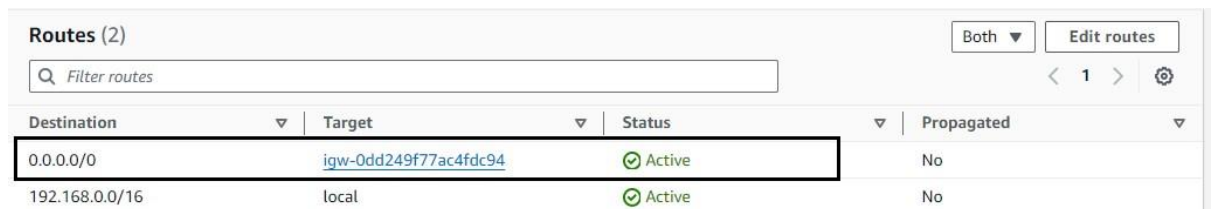
1. Select The Route Table And click on **Edit routes**



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)

The screenshot shows the 'Network settings' step of the AWS 'Launch Instance' wizard. The 'VPC' dropdown is set to 'vpc-0482186802ad848bb (Main_network)' and is highlighted with a red box. The 'Subnet' dropdown is set to 'subnet-07d8c9017e6b67a43 My_subnet 1' and is highlighted with a black box. The 'Auto-assign public IP' dropdown is set to 'Enable' and is highlighted with a blue box. Below these, there are options to 'Create security group' (selected) or 'Select existing security group'. The 'Security group name' field is filled with 'launch-wizard-4'. On the right, the 'Summary' panel shows the instance configuration: 1 instance, Amazon Linux 2023 AMI, t2.micro instance type, and 1 volume of 8 GiB. A blue arrow points to the 'Launch instance' button in the bottom right corner.

Network settings Info

VPC - required Info
vpc-0482186802ad848bb (Main_network)
192.168.0.0/16

Subnet Info
subnet-07d8c9017e6b67a43 My_subnet 1
VPC: vpc-0482186802ad848bb Owner: 412832919872
Availability Zone: us-east-1a IP addresses available: 251 CIDR: 192.168.1.0/24

Create new subnet

Auto-assign public IP Info
Enable

Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

Security group name - required
launch-wizard-4

Description - required Info

Summary

Number of instances Info
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.3.2...read more
ami-0f403e3180720dd7e

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Cancel Launch instance
Review commands

6. Click On launch Instance

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

The screenshot shows the AWS Management Console. The top navigation bar includes the AWS logo, 'Services' menu, a search bar, and a list of services: IAM, S3, EC2, EFS, VPC, CloudWatch, RDS, and Route 53. The main content area displays the 'Launch Instance' wizard completion screen. It shows the instance name 'Amazon Linux 2023' and the URL 'https://aws.amazon.com/linux/amazon-linux-2023'. Below this, there is a terminal window showing the command 'ec2-user@ip-192-168-1-182 ~]\$'.

aws Services Search [Alt+S]

IAM S3 EC2 EFS VPC CloudWatch RDS Route 53

Amazon Linux 2023

https://aws.amazon.com/linux/amazon-linux-2023

ec2-user@ip-192-168-1-182 ~]\$

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:

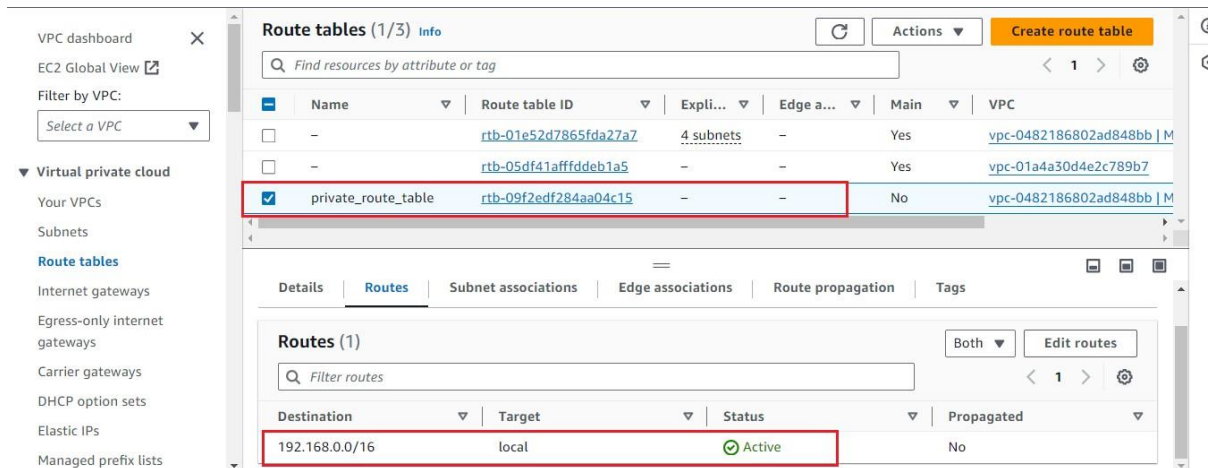
- Databases: Private subnets are suitable for hosting databases or other data storage systems that don't need direct internet access. This provides an additional layer of security by isolating sensitive data from direct exposure to the internet.
- Application Servers: Backend application servers that 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)

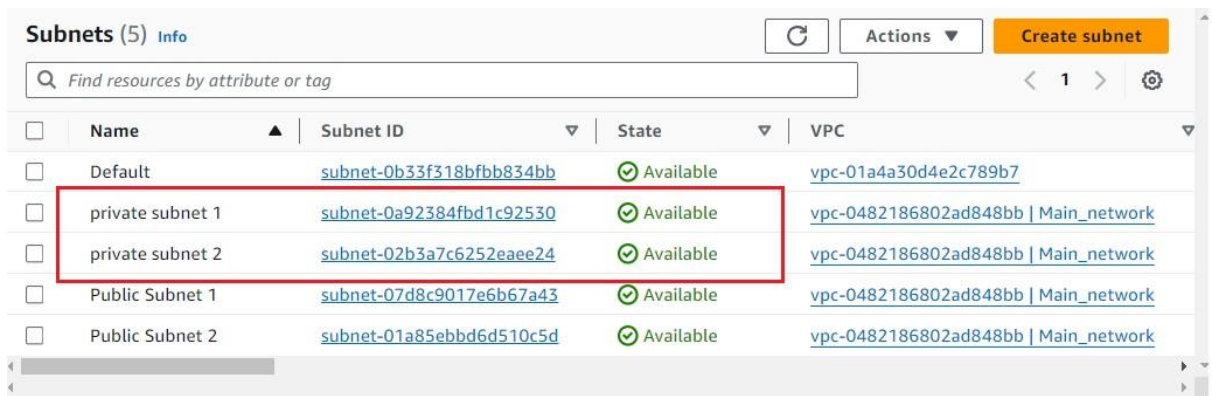


The screenshot shows the AWS VPC console 'Route tables (1/3)' page. A table lists three route tables. The 'private_route_table' (ID: rtb-09f2edf284aa04c15) is selected with a checkbox. Below the table, the 'Routes (1)' section shows a single route with destination 192.168.0.0/16, target 'local', and status 'Active'.

Name	Route table ID	Expli...	Edge a...	Main	VPC
-	rtb-01e52d7865fda27a7	4 subnets	-	Yes	vpc-0482186802ad848bb M
-	rtb-05df41affddeb1a5	-	-	Yes	vpc-01a4a30d4e2c789b7
<input checked="" type="checkbox"/> private_route_table	rtb-09f2edf284aa04c15	-	-	No	vpc-0482186802ad848bb M

Destination	Target	Status	Propagated
192.168.0.0/16	local	Active	No

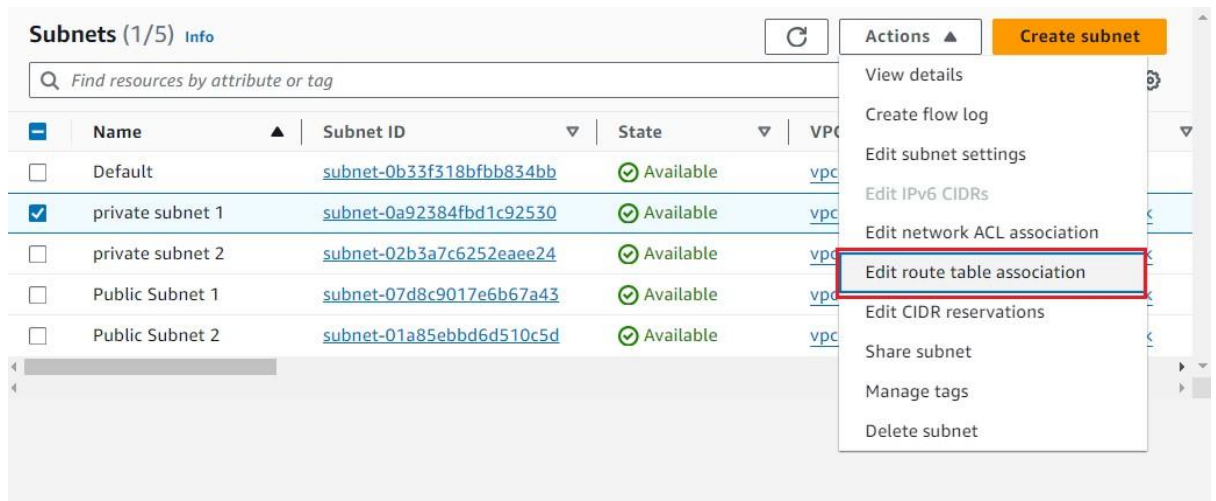
2. Create subnet



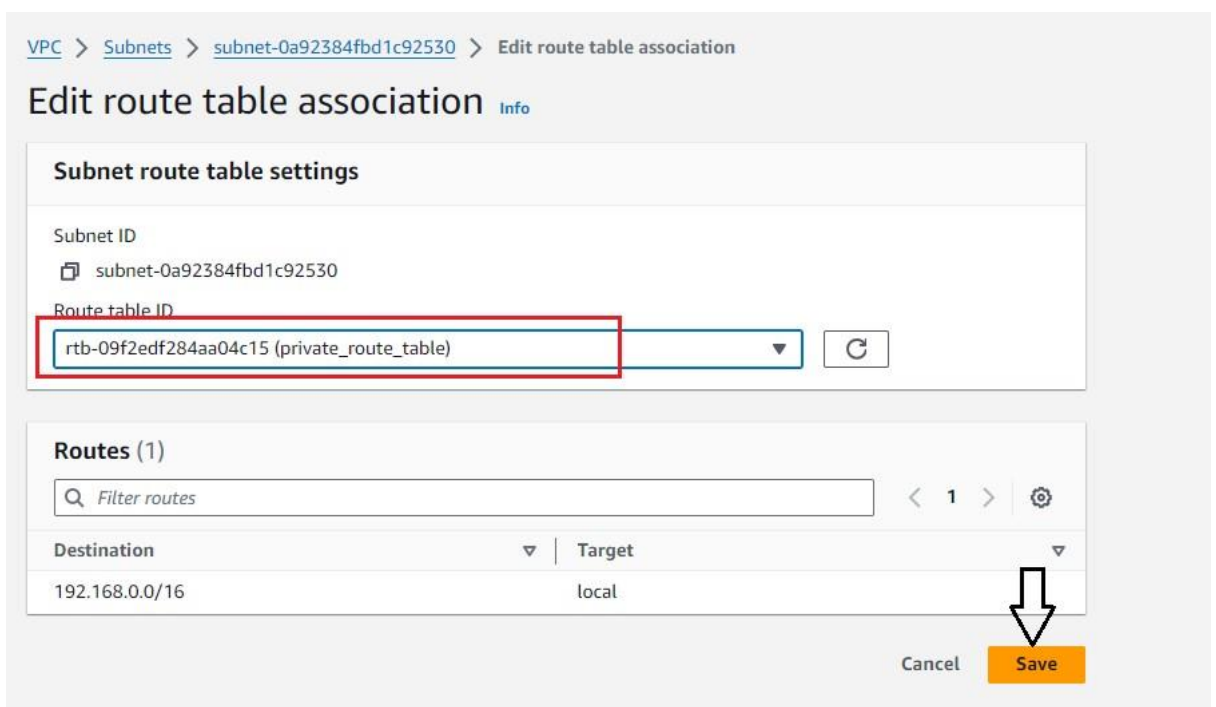
The screenshot shows the AWS VPC console 'Subnets (5)' page. A table lists five subnets. Two private subnets, 'private subnet 1' and 'private subnet 2', are highlighted with a red box. Both are in an 'Available' state and associated with VPC 'vpc-0482186802ad848bb'.

Name	Subnet ID	State	VPC
Default	subnet-0b33f318bfb834bb	Available	vpc-01a4a30d4e2c789b7
private subnet 1	subnet-0a92384fbd1c92530	Available	vpc-0482186802ad848bb Main_network
private subnet 2	subnet-02b3a7c6252eae24	Available	vpc-0482186802ad848bb Main_network
Public Subnet 1	subnet-07d8c9017e6b67a43	Available	vpc-0482186802ad848bb Main_network
Public Subnet 2	subnet-01a85ebbd6d510c5d	Available	vpc-0482186802ad848bb Main_network

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)

▼ Network settings Info

VPC - required Info

vpc-0482186802ad848bb (Main_network)
192.168.0.0/16

Subnet Info

subnet-07d8c9017e6b67a43 Public Subnet 1
VPC: vpc-0482186802ad848bb Owner: 412832919872
Availability Zone: us-east-1a IP addresses available: 250 CIDR: 192.168.1.0/24

Auto-assign public IP Info

Enable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group ☐ Select existing security group

Security group name - required

Public Instance Created successfully....

Instances (2) Info

Find Instance by attribute or tag (case-sensitive)

Any state

	Name	Instance ID	Instance state	Instance type	Status che
<input type="checkbox"/>	public_instance	i-0074b4a2b1502118d	Running	t2.micro	2/2 che
<input type="checkbox"/>	private_instance	i-0bba03ede312d3321	Running	t2.micro	2/2 che

Create private instance:-

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

▼ **Network settings** [Info](#)

VPC - required [Info](#)

vpc-0482186802ad848bb (Main_network)
192.168.0.0/16

Subnet [Info](#)

subnet-0a92384fbd1c92530 private subnet 1
VPC: vpc-0482186802ad848bb Owner: 412832919872
Availability Zone: us-east-1a IP addresses available: 251 CIDR: 192.168.3.0/24

Auto-assign public IP [Info](#)

Disable

Firewall (security groups) [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☐ Create security group ☒ Select existing security group

2. Private Instance Created successfully....

Instances (2) [Info](#) [Refresh](#) [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

Find Instance by attribute or tag (case-sensitive)

Any state

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status checks
<input type="checkbox"/>	public_instance	i-0074b4a2b1502118d	Running	t2.micro	2/2 checks successful
<input type="checkbox"/>	private_instance	i-0bba03ede312d3321	Running	t2.micro	2/2 checks successful

Pinging Private and public Instances:-

Add ALL ICMP – IPV4 rule in security Group...

Inbound rules [Info](#)

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info		
sgr-00be35d1713b93e91	SSH ▼	TCP	22	C... ▼	<input type="text" value="Q"/> <div>0.0.0.0/0 ✕</div>	<input type="text"/>	Del ete
sgr-032f0dc5d1d23a254	All ICMP - IPv4 ▼	ICMP	All	C... ▼	<input type="text" value="Q"/> <div>0.0.0.0/0 ✕</div>	<input type="text"/>	Del ete

Add rule

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

      #
    ~\   ####_~
~~~\___#####\
~~~\____|###|
~~~\___/#/
~~~V~'-'>
~~~~
~~~~-.
~~~~-/_-/_/
~~~~/_m/'-/_/

Amazon Linux 2023

https://aws.amazon.com/linux/amazon-linux-2023

Last login: Sun Mar 10 20:29:50 2024 from 192.168.1.182
[ec2-user@ip-192-168-4-212 ~]$
```