

Objective:- Subnetting in Networking

- VPC :- 172.21.10.12

Class B 13-12 =1

Subnet mask:- 255.255.192.0

Block Size :- 256-192 = 64

IPs	SUBNET1	SUBNET2
Network	172.21.0.0 / 18	172.21.64.0 / 18
Broadcasting	172.21.63.255 / 18	172.21.127.255 / 18

- SUBNET :- 172.21.32.0 / 24

Class B 24-12 =12 Bits borrow from host

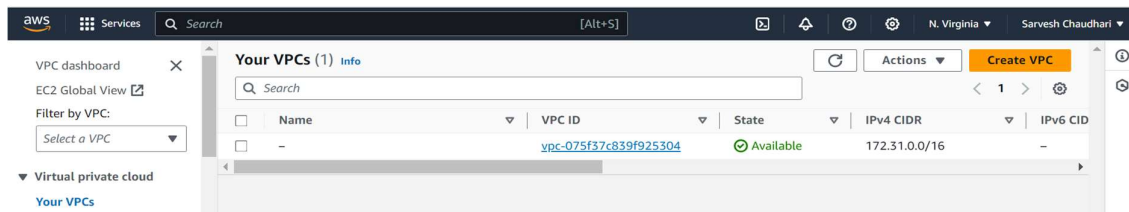
Subnet mask:- 255.255.192.0

Block Size :- 256 - 255 = 1

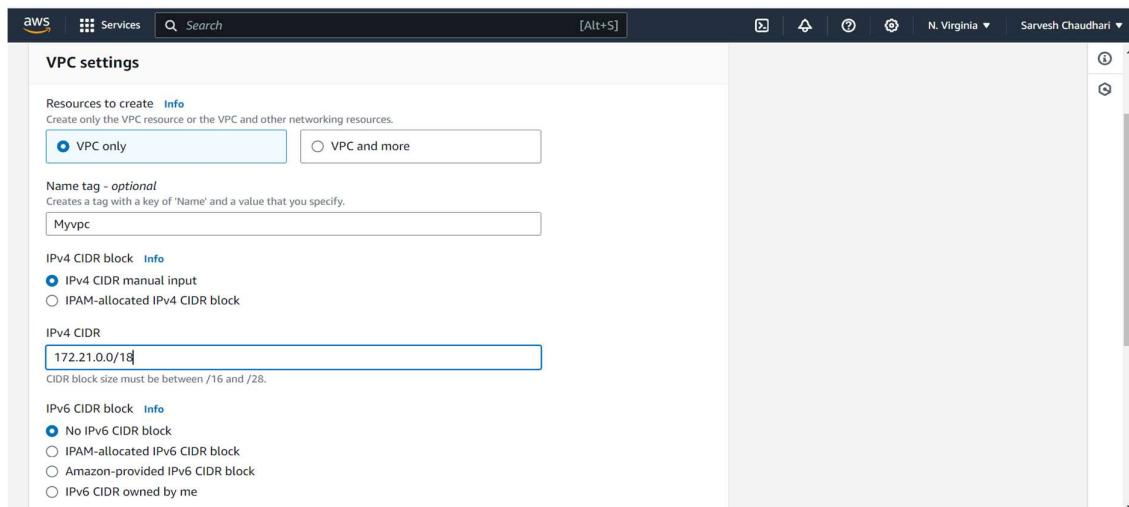
IPs	SUBNET1	SUBNET2
Network	172.21.32.0 / 18	172.21.33.0 / 18
Broadcasting	172.21.32.255 / 18	172.21.33.255 / 18

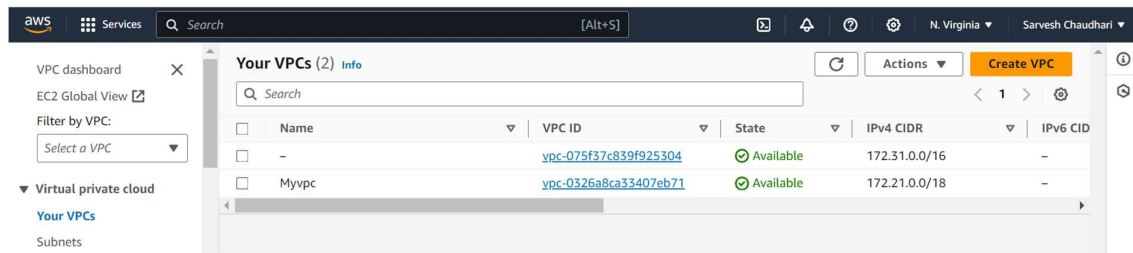
➤ Steps for Subnetting

S1) Click on create VPC



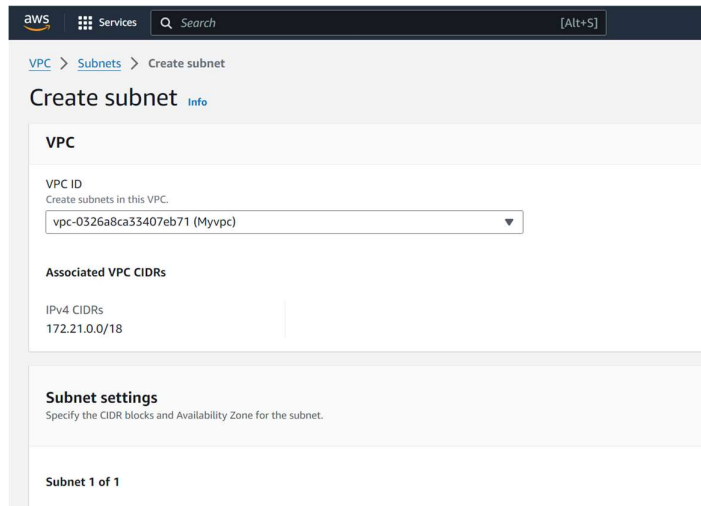
Now type name for VPC & IPv4 CIDR, then click on create VPC





Now create subnet for Publicsubnet

Select VPC



Now in subnet setting give name, Availability zone, then give IPv4 subnet CIDR block

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.
Publicsubnet
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.
US East (N. Virginia) / us-east-1a

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.
172.21.0.0/18

IPv4 subnet CIDR block
172.21.32.0/24 256 IPs

Then click on create. Now similarly create private subnet with subnet2 IP address.

Now create Internet gateway

VPC > Internet gateways > Create internet gateway

Create internet gateway [Info](#)

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

Name tag

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

MyGTW

Tags - optional

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

Q Name X

Q MyGTW X

Remove

Add new tag

You can add 49 more tags.

Now select Internet gateway then using action Attach to VPC

aws Services Search [Alt+S]

EC2 Global View

Filter by VPC: Select a VPC

Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Internet gateways (1/2) Info

Search

Name	Internet gateway ID
-	igw-087876bdd07f45270
MyGTW	igw-01a838dbd399cd063

Actions

Create internet gateway

View details

Attach to VPC

Detach from VPC

Manage tags

Delete internet gateway

Now create Route Table for Public and private

aws Services Search [Alt+S]

connection.

Route table settings

Name - optional

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

PublicRT

VPC

The VPC to use for this route table.

vpc-0326a8ca33407eb71 (Myvpc)

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

Q Name X

Q PublicRT X

Remove

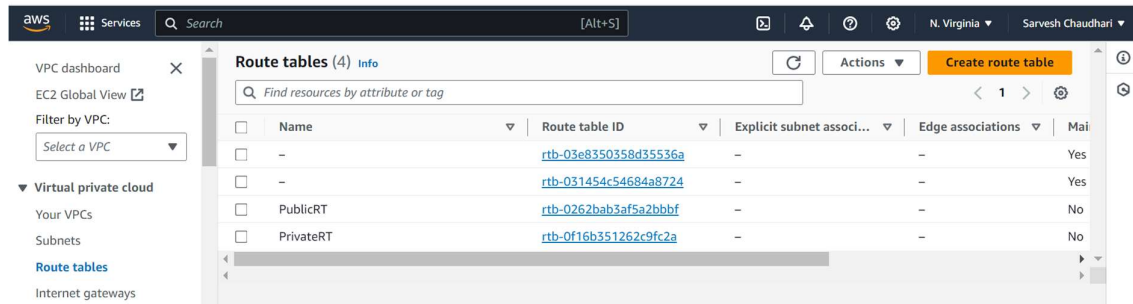
Add new tag

You can add 49 more tags.

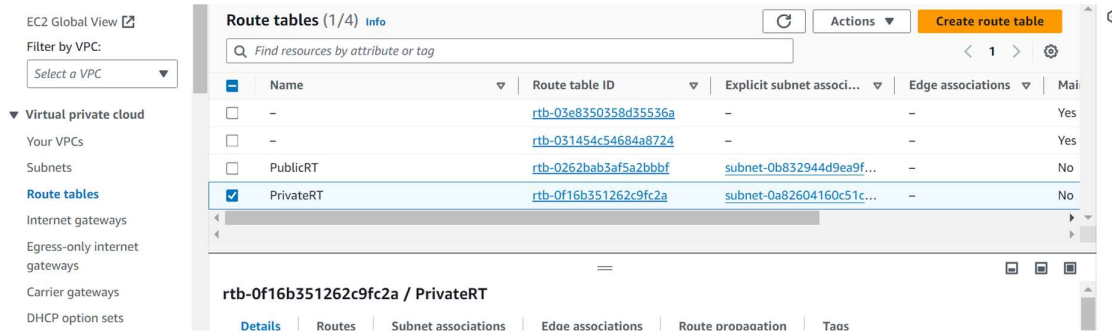
Cancel

Create route table

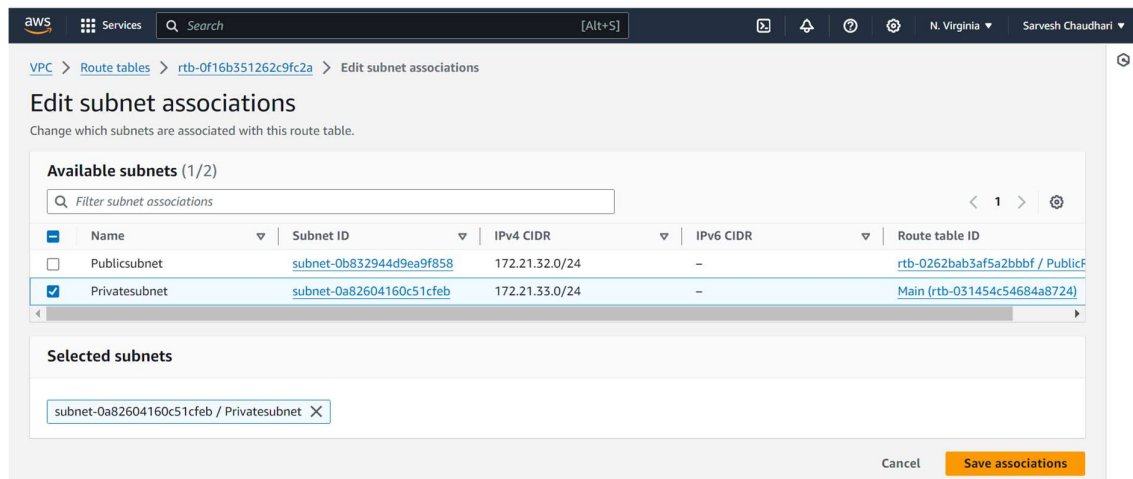
Both Route tables created



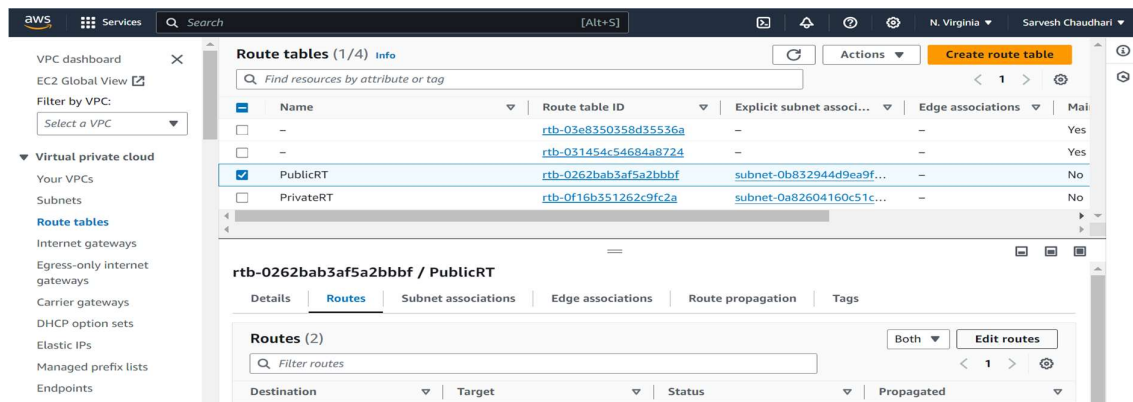
Now select PublicRT then go to Subnet Association > Edit Subnet association



Now Select publicsubnet and privatesubnet for there respective RT



Now edit route.



Now go to PublicRT then go to Routes > edit route > add route > Select created Getway then save changes

VPC > Route tables > rtb-0262bab3af5a2bbbf > Edit routes

Edit routes

Destination	Target	Status	Propagated
172.21.0.0/18	local	Active	No
0.0.0.0/0	Internet Gateway	-	No

Buttons: Add route, Cancel, Preview, Save changes

Now create instance Webserver & DBserver

Launch an instance

Name and tags

Name: Webserver

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.3.2

Virtual server type (instance type): t2.micro

Now select VPC, subnet and auto-assign public IP enable for Public

For private disable public IP

Network settings

VPC - required: vpc-0326a8ca33407eb71 (Myvpc)

Subnet: subnet-0b832944d9ea9f858 (Publicsubnet)

Auto-assign public IP: Enable

Firewall (security groups): Create security group

Security group name - required: securitygp1

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.3.2

Virtual server type (instance type): t2.micro

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

Buttons: Cancel, Launch instance

```
aws Services Search [Alt+S] N. Virginia Sarvesh Chaudhari
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-172-21-32-90 ~]$ sudo su
[root@ip-172-21-32-90 ec2-user]# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=58 time=3.88 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=58 time=1.24 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=58 time=1.26 ms
^Z
[1]+  Stopped                  ping 8.8.8.8
[root@ip-172-21-32-90 ec2-user]#
```

```
64 bytes from 8.8.8.8: icmp_seq=70 ttl=58 time=1.24 ms
64 bytes from 8.8.8.8: icmp_seq=71 ttl=58 time=1.26 ms
64 bytes from 8.8.8.8: icmp_seq=72 ttl=58 time=1.29 ms
64 bytes from 8.8.8.8: icmp_seq=73 ttl=58 time=1.28 ms
64 bytes from 8.8.8.8: icmp_seq=74 ttl=58 time=1.27 ms
64 bytes from 8.8.8.8: icmp_seq=75 ttl=58 time=1.24 ms
64 bytes from 8.8.8.8: icmp_seq=76 ttl=58 time=1.38 ms
64 bytes from 8.8.8.8: icmp_seq=77 ttl=58 time=1.30 ms
64 bytes from 8.8.8.8: icmp_seq=78 ttl=58 time=1.23 ms
64 bytes from 8.8.8.8: icmp_seq=79 ttl=58 time=1.35 ms
^Z
[1]+  Stopped                  ping 8.8.8.8
[root@ip-172-21-32-156 ec2-user]# vi dbkey.pem
[root@ip-172-21-32-156 ec2-user]# ls
```

```
[root@ip-172-21-32-156 ec2-user]# ls
dbkey.pem
[root@ip-172-21-32-156 ec2-user]# chmod 700 dbkey.pem
[root@ip-172-21-32-156 ec2-user]# ssh -i dbkey.pem e2-user@172.21.33.30
The authenticity of host '172.21.33.30 (172.21.33.30)' can't be established.
ED25519 key fingerprint is SHA256:0IscRTTA8BCczitrCJ6UxjXNqSs6eCymPa4B+0VvdU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.21.33.30' (ED25519) to the list of known hosts.
e2-user@172.21.33.30: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[root@ip-172-21-32-156 ec2-user]#
```

Here when we ping public ip of Webserver it runs but Ping of private ip of DBserver is not running.

To do that as we have go to back to our machine

```
64 bytes from 172.21.33.40: icmp_seq=3 ttl=127 time=0.034 ms
64 bytes from 172.21.33.40: icmp_seq=4 ttl=127 time=0.035 ms
64 bytes from 172.21.33.40: icmp_seq=5 ttl=127 time=0.035 ms
^Z
[2]+  Stopped                  ping 172.21.33.40
[ec2-user@ip-172-21-33-40 ~]$ exit
logout
There are stopped jobs.
[ec2-user@ip-172-21-33-40 ~]$ exit
logout
Connection to 172.21.33.40 closed.
[root@ip-172-21-32-90 ec2-user]# sudo su
[root@ip-172-21-32-90 ec2-user]# exit
exit
[root@ip-172-21-32-90 ec2-user]# logout
bash: logout: not login shell: use 'exit'
[root@ip-172-21-32-90 ec2-user]# exit
exit
[ec2-user@ip-172-21-32-90 ~]$ ping 172.21.33.40
PING 172.21.33.40 (172.21.33.40) 56(84) bytes of data.
^Z
[1]+  Stopped                  ping 172.21.33.40
```

In last command we ping private ip but it was not done to do this go to instances then from security go to security groups then GO to Inbound rules in type select “All ICMP IPv4” type then as source select anywhere.

aws Services Search [Alt+S] N. Virginia Sarvesh Chaudhari

Inbound rules info

Security group rule ID	Type	Protocol	Port range	Source	Description - optional	
sgr-06ede2110ada921c2	All ICMP - IPv4	ICMP	All	Cu... 0.0.0.0/0		Delete
sgr-0e30cc1d22b0217a0	SSH	TCP	22	Cu... 0.0.0.0/0		Delete

Add rule

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Preview changes Save rules

Do the same thing for the other instance and it will allow access.

Now as we change inbound rule it allows ping private-ip

```
[root@ip-172-21-32-90 ec2-user]# exit
exit
[root@ip-172-21-32-90 ec2-user]# logout
bash: logout: not login shell: use `exit'
[root@ip-172-21-32-90 ec2-user]# exit
exit
[ec2-user@ip-172-21-32-90 ~]$ ping 172.21.33.40
PING 172.21.33.40 (172.21.33.40) 56(84) bytes of data.
^Z
[1]+  Stopped                  ping 172.21.33.40
[ec2-user@ip-172-21-32-90 ~]$ ping 172.21.33.40
PING 172.21.33.40 (172.21.33.40) 56(84) bytes of data.
64 bytes from 172.21.33.40: icmp_seq=1 ttl=127 time=0.511 ms
64 bytes from 172.21.33.40: icmp_seq=2 ttl=127 time=0.642 ms
64 bytes from 172.21.33.40: icmp_seq=3 ttl=127 time=0.533 ms
64 bytes from 172.21.33.40: icmp_seq=4 ttl=127 time=0.523 ms
64 bytes from 172.21.33.40: icmp_seq=5 ttl=127 time=0.536 ms
^Z
[2]+  Stopped                  ping 172.21.33.40
[ec2-user@ip-172-21-32-90 ~]$
```