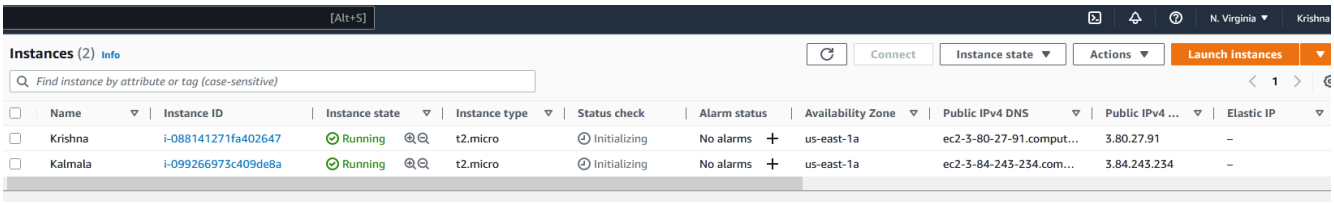
1. Creating Instances:

- As per below screenshot, Created 2 instances with names as

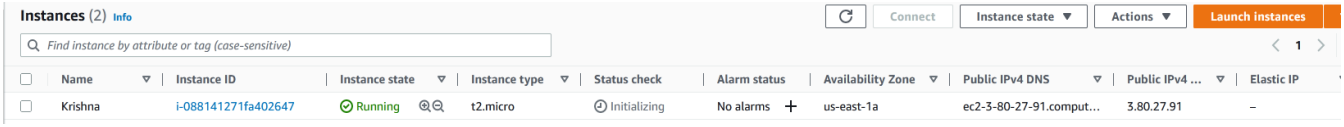
  Instance1 - "**Krishna**" & Instance2 -"**Kalmala**".



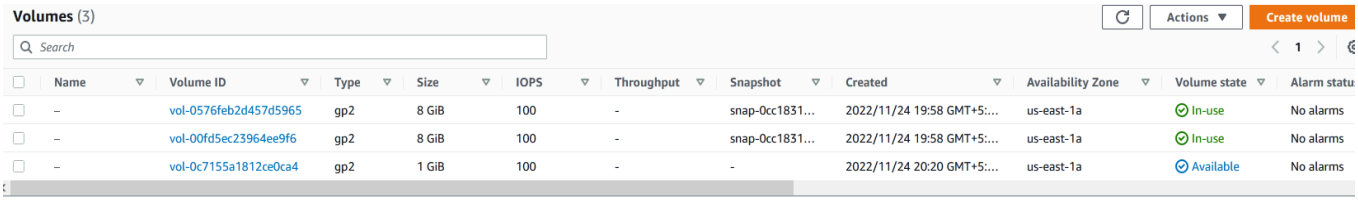
---

2. Creating Elastic Block Store (EBS):

- Initially created an instance with name as "Krishna"

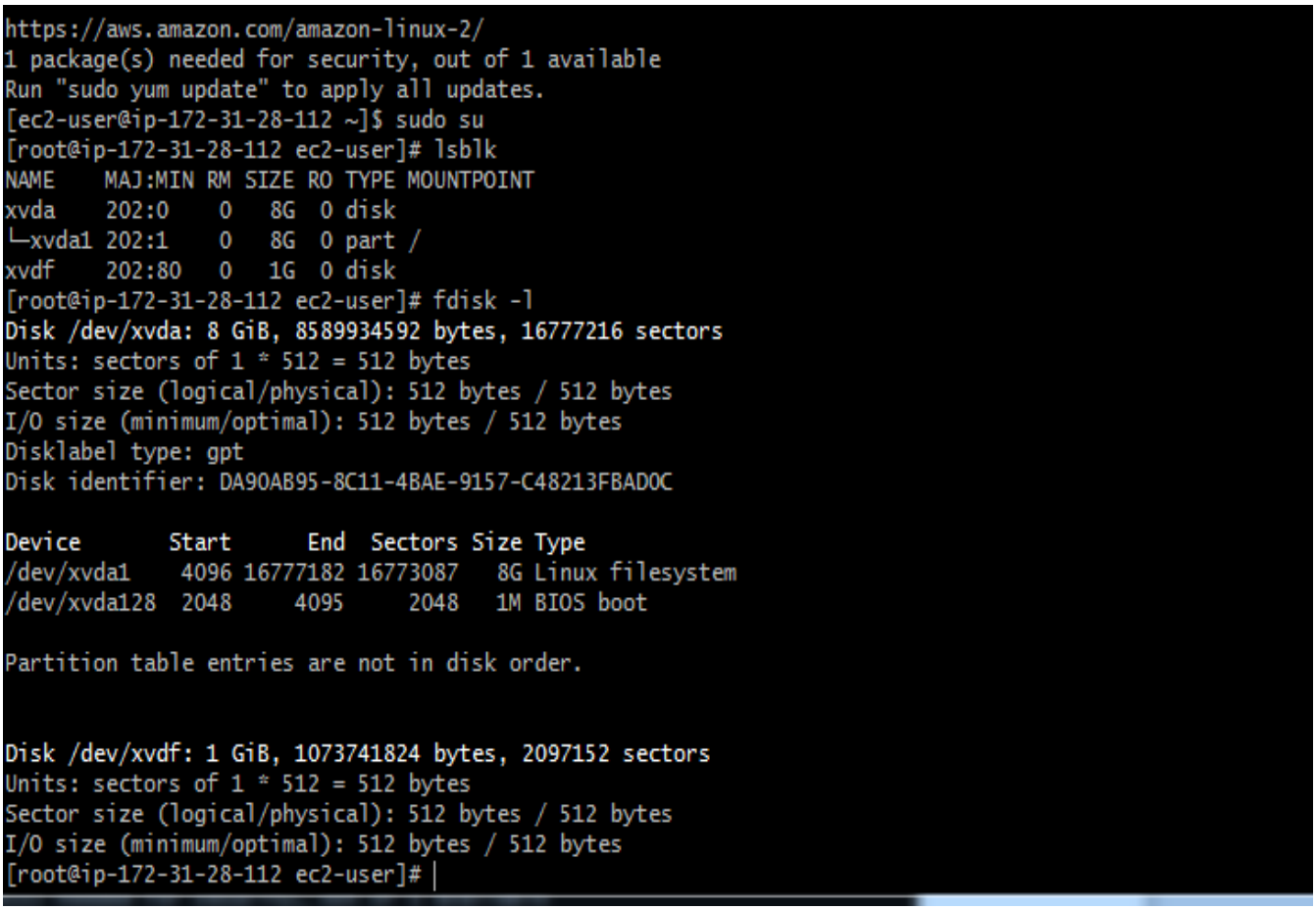


- then, created volume of 1GB



- now its attached to instance "Krishna" & then through terminal checking commands like

"lsblk" (lists all file systems)



- Making directory & Mounting

```
Disk /dev/xvdf: 1 GiB, 1073741824 bytes, 2097152 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
[root@ip-172-31-28-112 ec2-user]# mkdir testdir
[root@ip-172-31-28-112 ec2-user]# mkfs -t xfs /dev/xvdf
meta-data=/dev/xvdf              isize=512    agcount=4, agsize=65536 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=0
data     =                       bsize=4096   blocks=262144, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal log           bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
[root@ip-172-31-28-112 ec2-user]# mount -t xfs /dev/xvdf /home/ec2-user/testdir
[root@ip-172-31-28-112 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-28-112 ec2-user]# lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda    202:0    0   8G  0 disk
└─xvda1 202:1    0   8G  0 part /
xvdf    202:80   0   1G  0 disk /home/ec2-user/testdir
[root@ip-172-31-28-112 ec2-user]#
```

- Creating file under dir

```
[root@ip-172-31-28-112 ec2-user]# cd testdir
[root@ip-172-31-28-112 testdir]# touch testfile.txt
[root@ip-172-31-28-112 testdir]# ls
testfile.txt
[root@ip-172-31-28-112 testdir]#
```

- Now, created another instance "Kalmala" & dettached volume from instance Krishna & attached to instance "Kalmala"

```
[ec2-user@ip-172-31-18-204 ~]$ sudo su
[root@ip-172-31-18-204 ec2-user]# lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda    202:0    0   8G  0 disk
└─xvda1 202:1    0   8G  0 part /
xvdf    202:80   0   1G  0 disk
[root@ip-172-31-18-204 ec2-user]#
```

- now from instance2 we can view the file system which was created through instance1

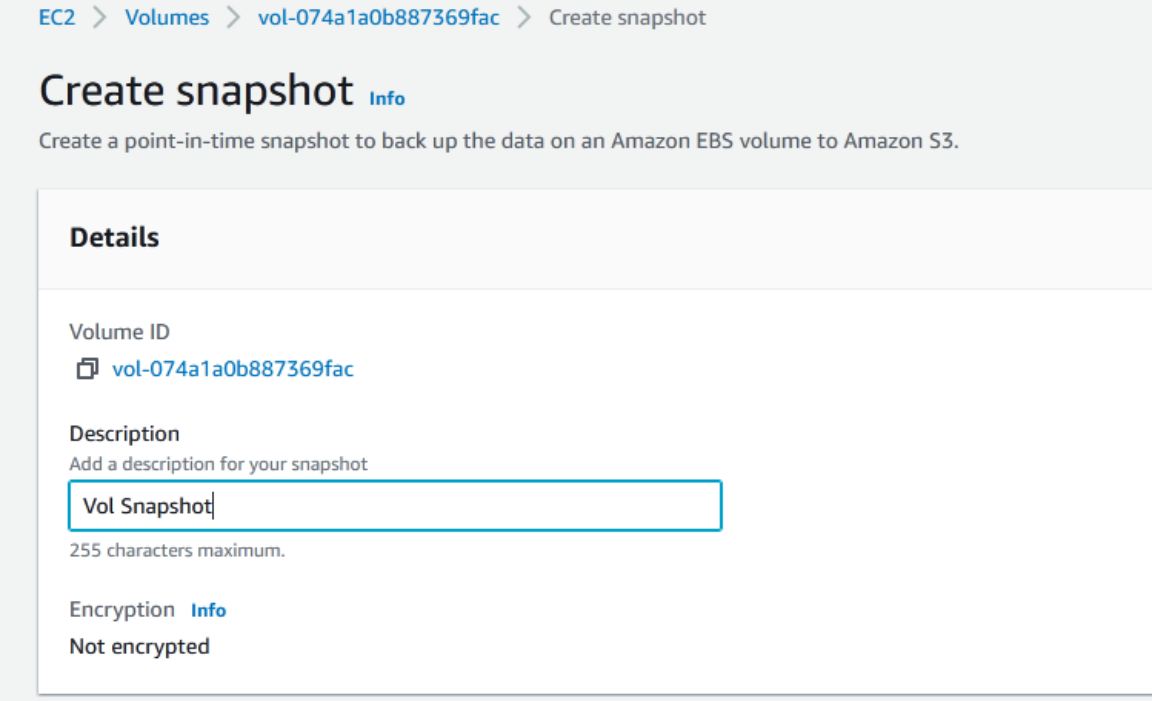
```
[root@ip-172-31-18-204 ec2-user]# mkdir test2
[root@ip-172-31-18-204 ec2-user]# mount -t xfs /dev/xvdf test2
[root@ip-172-31-18-204 ec2-user]# cd test2
[root@ip-172-31-18-204 test2]# ls
testfile.txt
[root@ip-172-31-18-204 test2]#
```

---

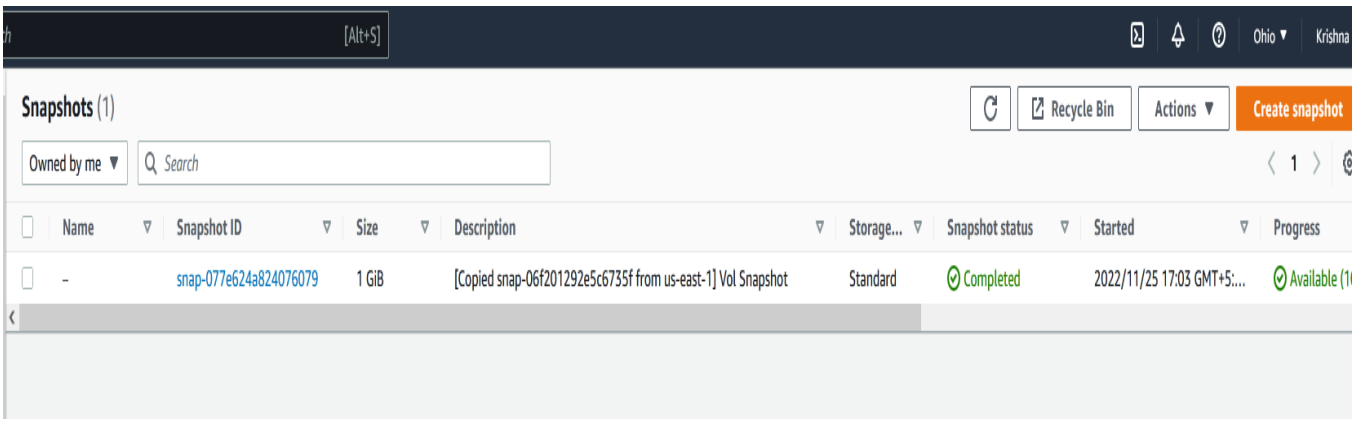
3. Creating SNAPHSHOT:

- We can create snapshot from snapshot section or from volume section

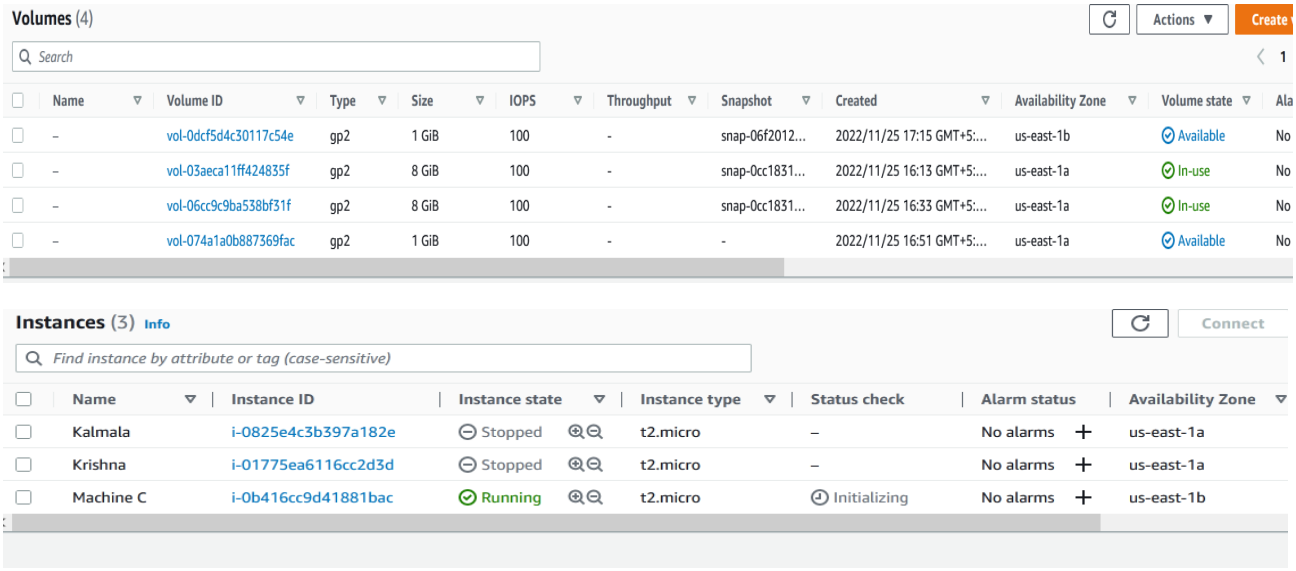
- Creating snapshot from volume



- Snapshot created in another region (Ohio) from region (N Virginia)



& Through snapshot we have created similar volume in different availability zone & same with instance - "Machine-C"



Volumes (4)

Actions ▼ Create

Search

1

Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot	Created	Availability Zone	Volume state	Ala
–	vol-0dcf5d4c30117c54e	gp2	1 GiB	100	-	snap-06f2012...	2022/11/25 17:15 GMT+5:...	us-east-1b	Available	No
–	vol-03aeca11ff424835f	gp2	8 GiB	100	-	snap-0cc1831...	2022/11/25 16:13 GMT+5:...	us-east-1a	In-use	No
–	vol-06cc9c9ba538bf31f	gp2	8 GiB	100	-	snap-0cc1831...	2022/11/25 16:33 GMT+5:...	us-east-1a	In-use	No
–	vol-074a1a0b887369fac	gp2	1 GiB	100	-	-	2022/11/25 16:51 GMT+5:...	us-east-1a	Available	No

Instances (3) Info

Connect

Find instance by attribute or tag (case-sensitive)

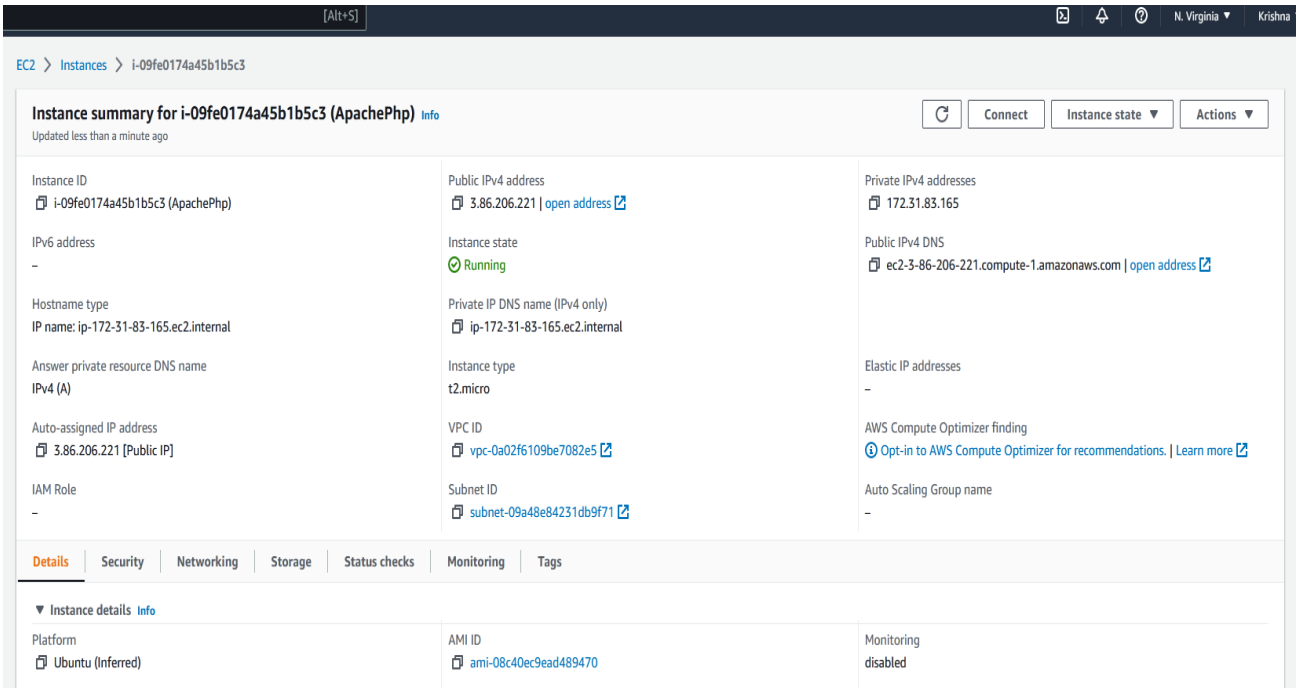
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Kalmala	i-0825e4c3b397a182e	Stopped	t2.micro	–	No alarms	us-east-1a
Krishna	i-01775ea6116cc2d3d	Stopped	t2.micro	–	No alarms	us-east-1a
Machine C	i-0b416cc9d41881bac	Running	t2.micro	Initializing	No alarms	us-east-1b

now by connecting to Machine-C we can use already created filesystem files in volume.

---

4. AMI (Amazon Machine Image):

- Created instance with Ubuntu server & followed instructions as step-by-step



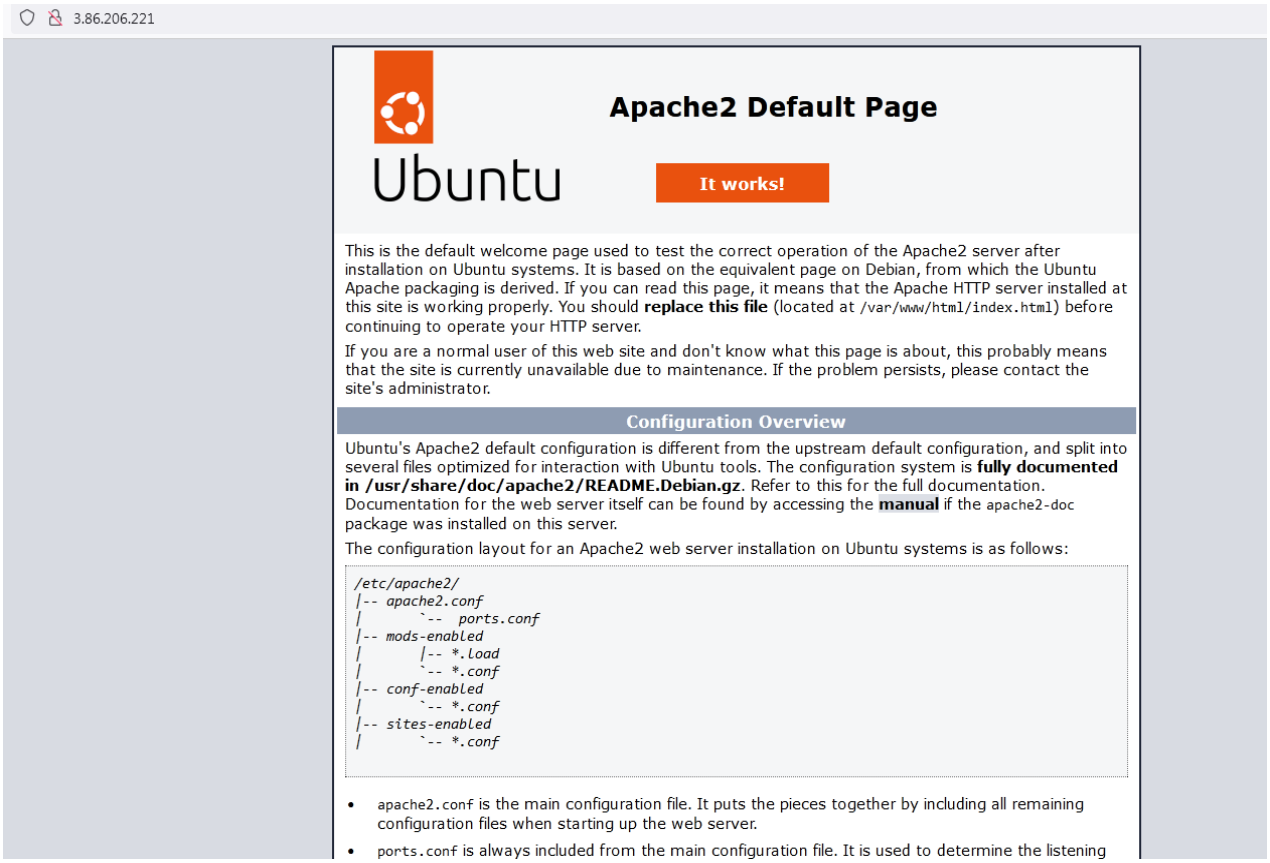

```
38 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-172-31-83-165:/home/ubuntu# ufw allow ssh
Rules updated
Rules updated (v6)
root@ip-172-31-83-165:/home/ubuntu# ufw allow 80
Rules updated
Rules updated (v6)
root@ip-172-31-83-165:/home/ubuntu# ufw allow 443
Rules updated
Rules updated (v6)
root@ip-172-31-83-165:/home/ubuntu# ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
root@ip-172-31-83-165:/home/ubuntu#
```

- Now, from below screenshot we can check apache2 service is active & running

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-83-165:/home/ubuntu# systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2022-11-26 08:11:16 UTC; 33s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2397 (apache2)
      Tasks: 55 (limit: 1143)
     Memory: 4.9M
        CPU: 30ms
     CGroup: /system.slice/apache2.service
             ├─2397 /usr/sbin/apache2 -k start
             ├─2399 /usr/sbin/apache2 -k start
             └─2400 /usr/sbin/apache2 -k start

Nov 26 08:11:16 ip-172-31-83-165 systemd[1]: Starting The Apache HTTP Server...
Nov 26 08:11:16 ip-172-31-83-165 systemd[1]: Started The Apache HTTP Server.
root@ip-172-31-83-165:/home/ubuntu#
```

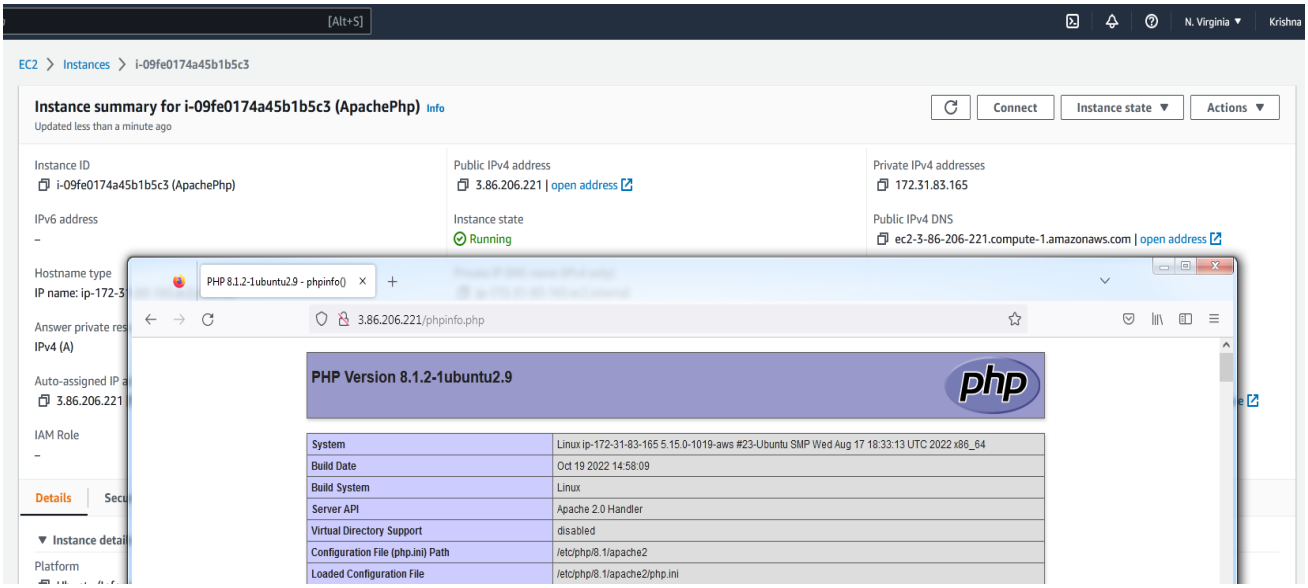
- now by searching with Instance public IP address(added HTTP-80 Port in Inbound Rule) in web we can view Default Apache2 page as below



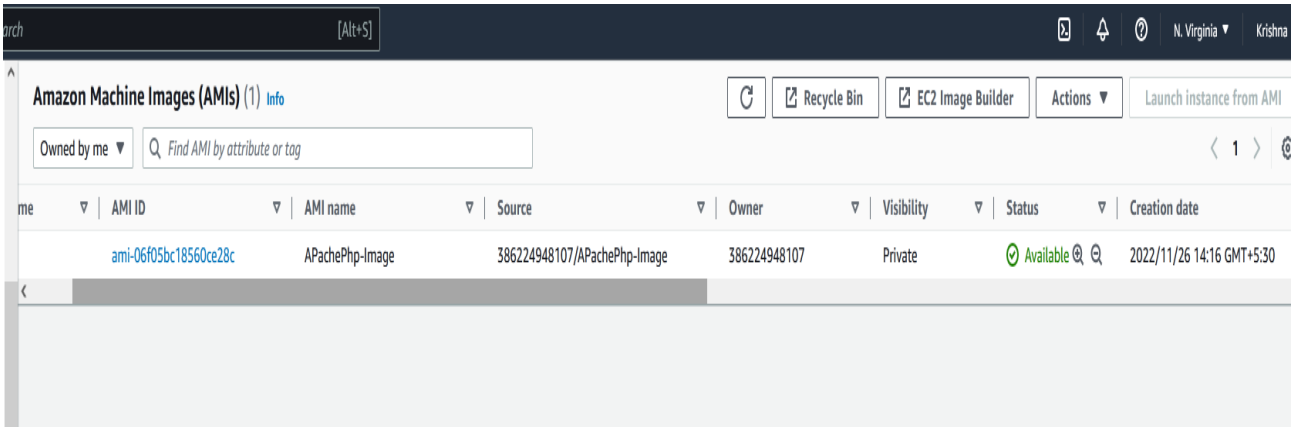
- installing PHP 8.1.2 version

```
root@ip-172-31-83-165:/# php --version
PHP 8.1.2-1ubuntu2.9 (cli) (built: Oct 19 2022 14:58:09) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.2-1ubuntu2.9, Copyright (c), by Zend Technologies
root@ip-172-31-83-165:/#
```

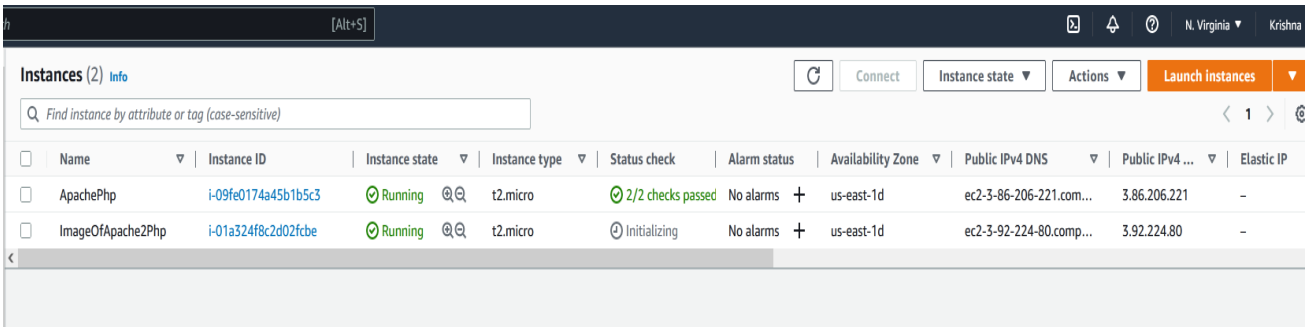
- we can view PHP in web with instance private IP address



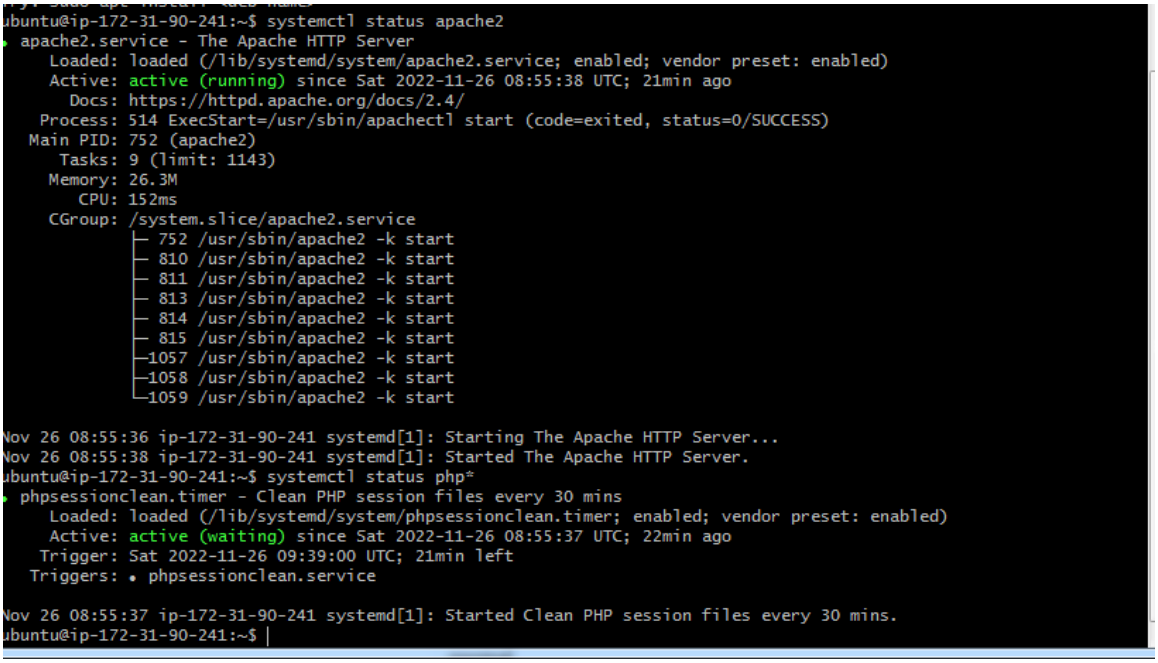
- Now, Creating Image from above Instance (applications with PHP & Apache2)



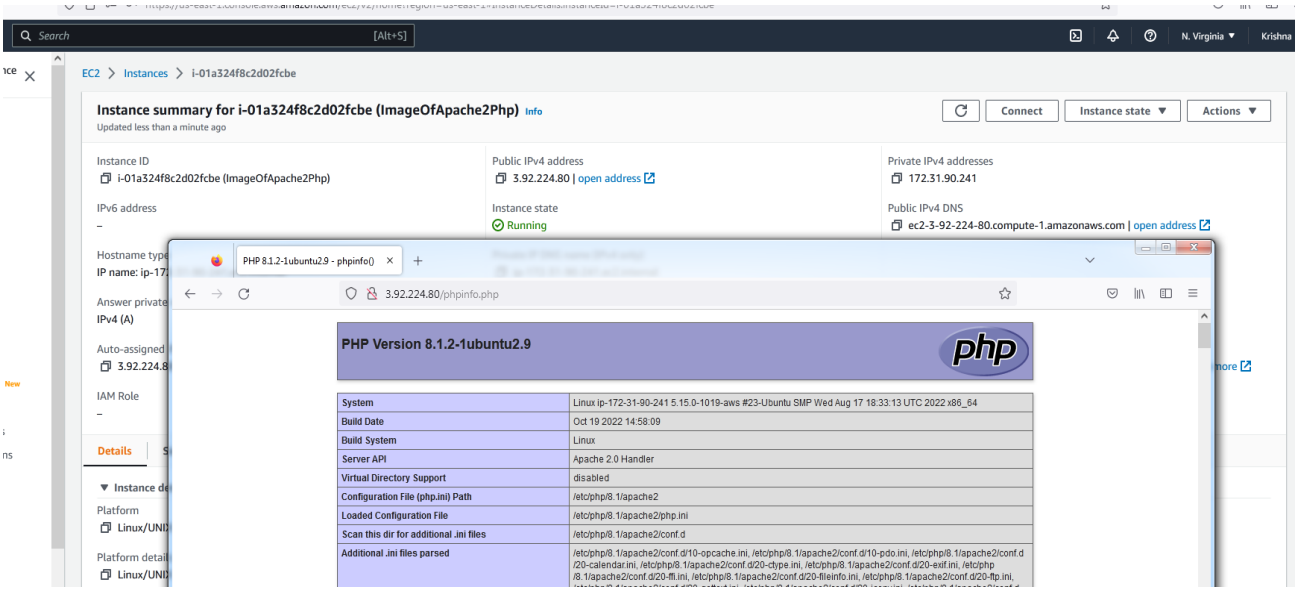
- And from above image we have created Instance (which comes with PHP & Apache2)



- Now from below screenshot we can see Instance created with Image is loaded with PHP & Apache2 in it



- & also we can ping in web with above created instance IP

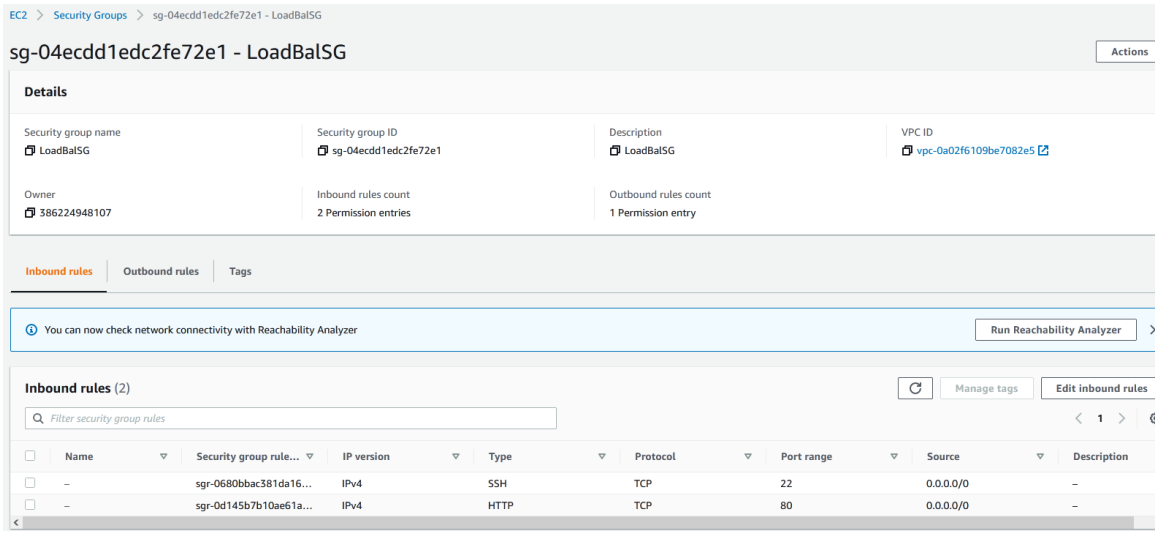


---

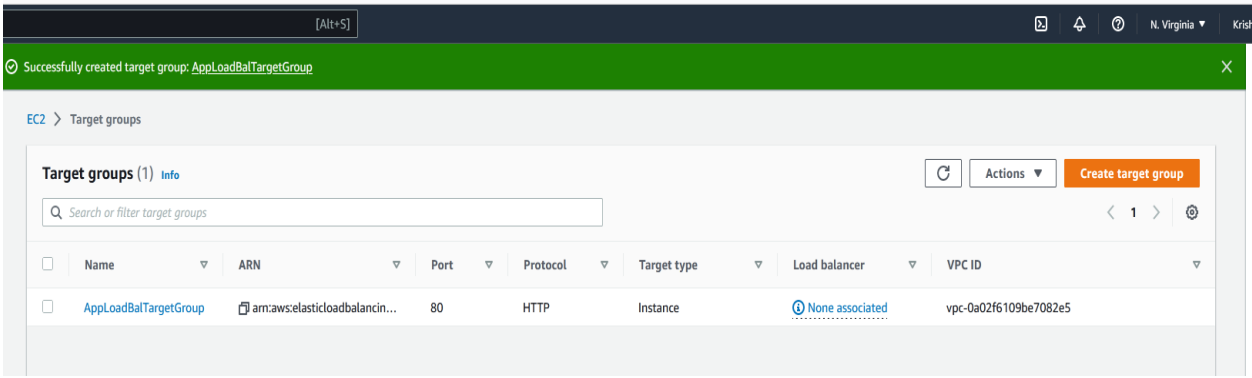
5. Load Balancer:

- Created Load Balancer with Customized Security Group -

  - Edited Inbound rules as SSH & HTTP Ports



& Created Target Group which included 2 instances

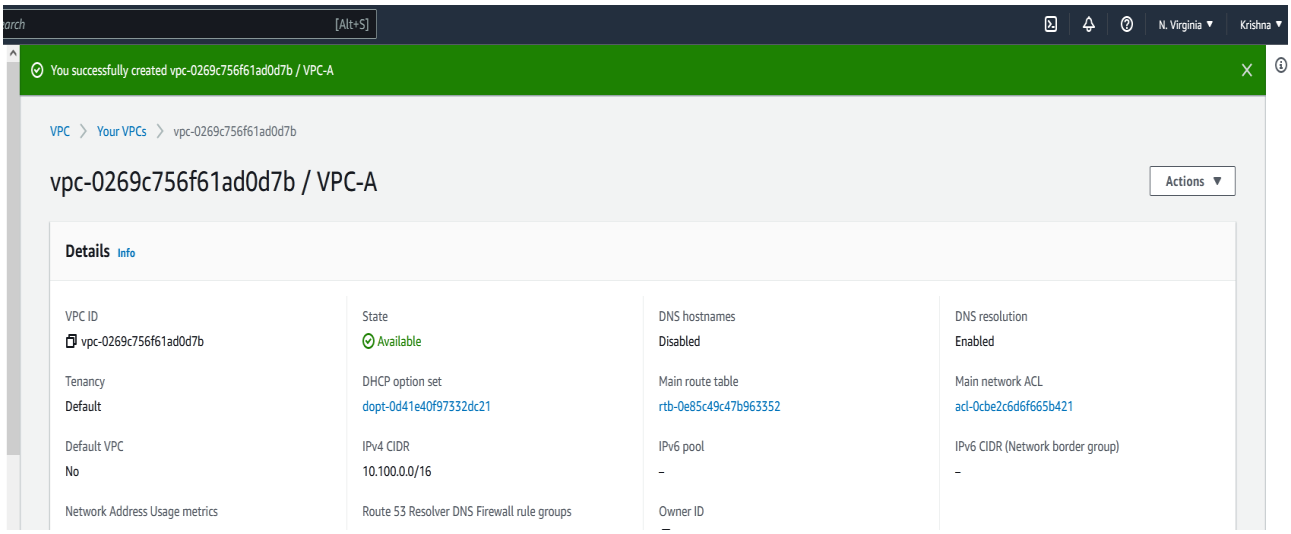


- Now by Copy & Pasting the DNS of Load Balancer in web, we can see 2Instance's Private IP addresses changing by refreshing web tab.

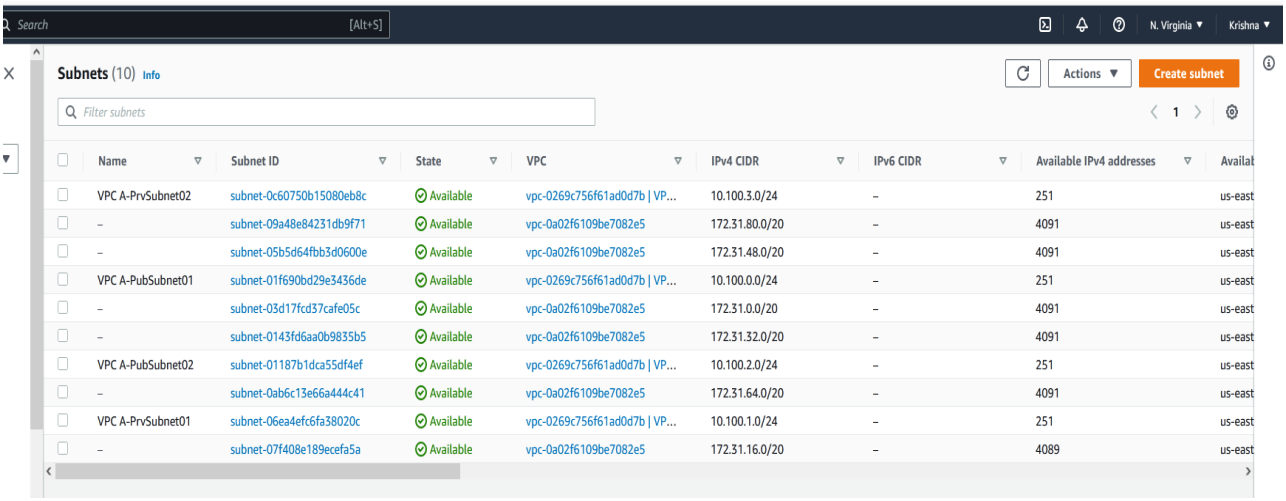
---

6. Virtual Private Cloud (VPC):

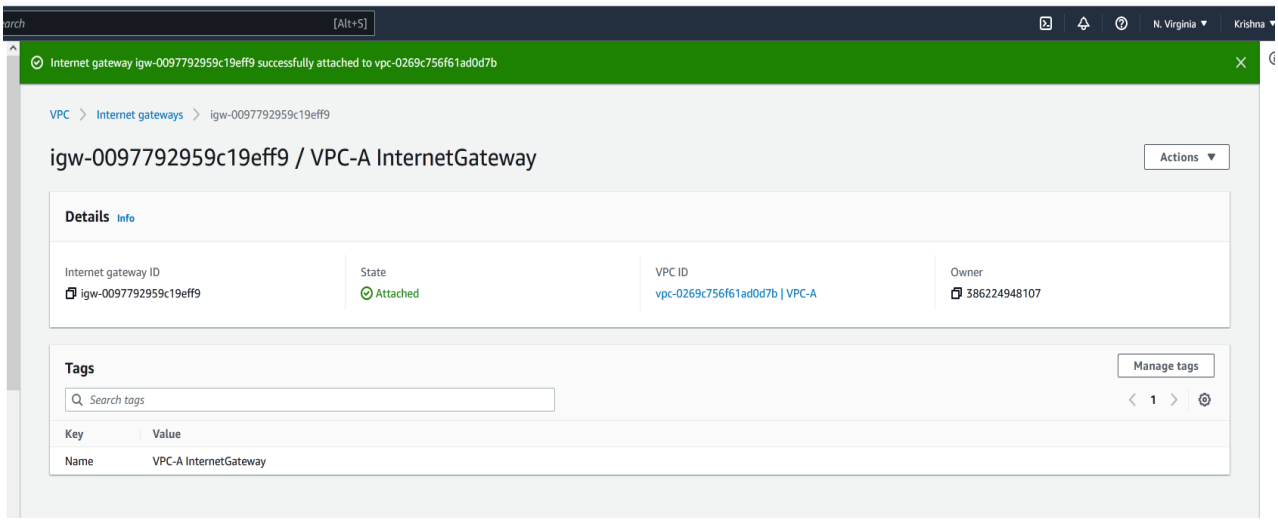
- As per below screenshot, Created VPC with name "VPC-A"



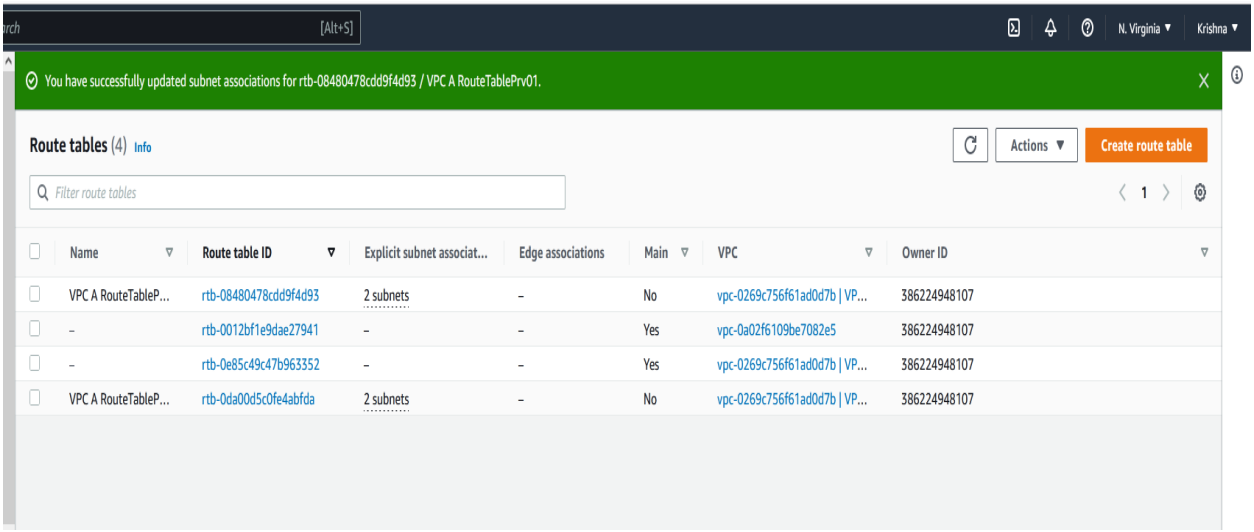
- And Created 4subnets (2-Public Subnets & 2-Private Subnets)



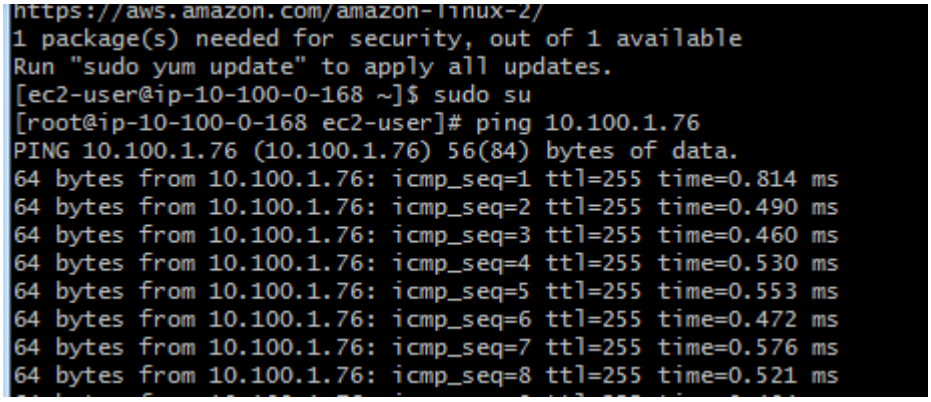
- Created Internet gateway & attached it to VPC-A



- Created 1Public route table & 1Private route table & associated with respective subnets
- Now, Public route table is connected to internet gateway



- Created SG with Inbound rules (SSH, All ICMP-IPv4, All TCP)
- Created 2instances (Public & Private)
- we can ping from public to private but not from private to public



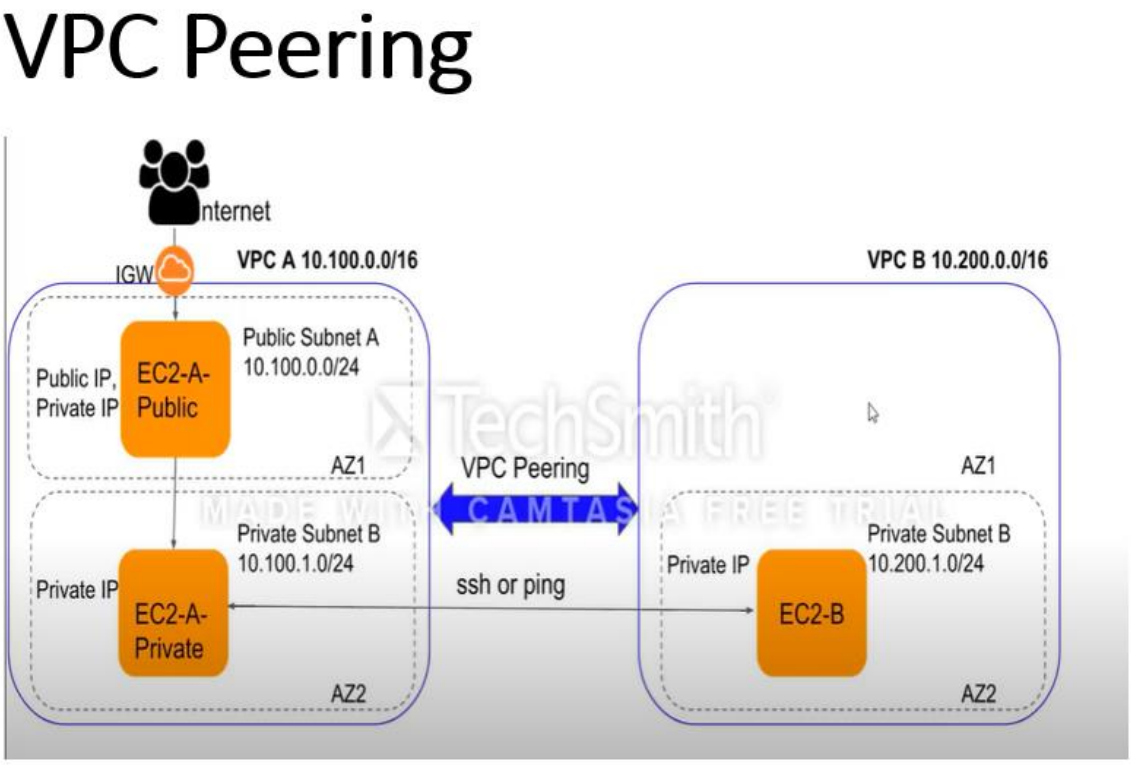
- By using command " scp -i keypair -r keypair ec2-user@PublicIPAddress:/home/ec2-user"

  we can copy keypair from Outside to Ec2-user of Public-Instance & to Private instance

  chmod - gives read/write/execute permissions

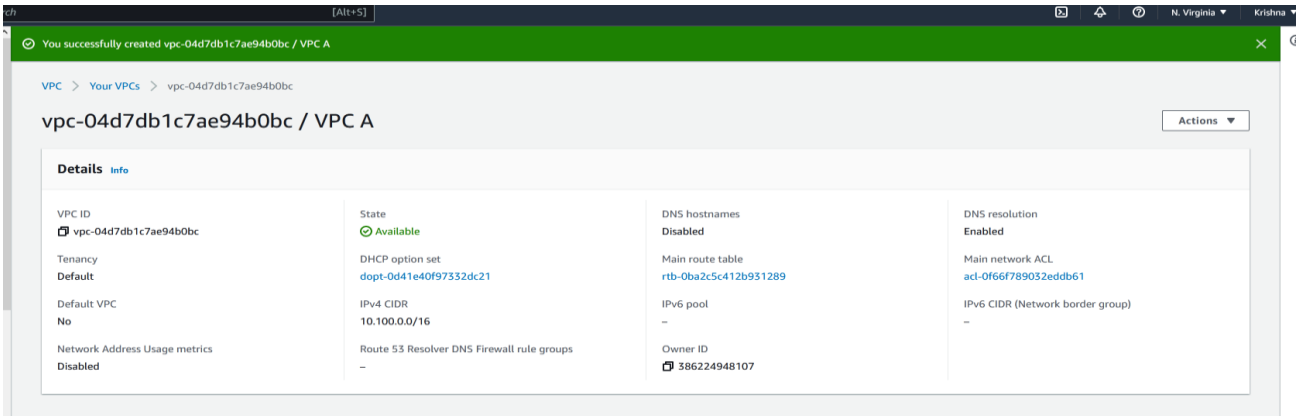
- to access internet from Private subnet we need Nat gateway
- Created NAT Gateway in public subnet by allocating Elastic IP address & edited private route table by adding Nat gateway at edit routes
- Created Nat gateway at Public Subnet & associated it with Private route table
- Now by doing ssh from public instance with Private subnet of Private IP address we can access internet.

7. VPC Peering:



- For the purpose of connecting & to slove any issue in Private subnet Instance of One VPC from Private subnet Instance of Other VPC we use VPC Peering

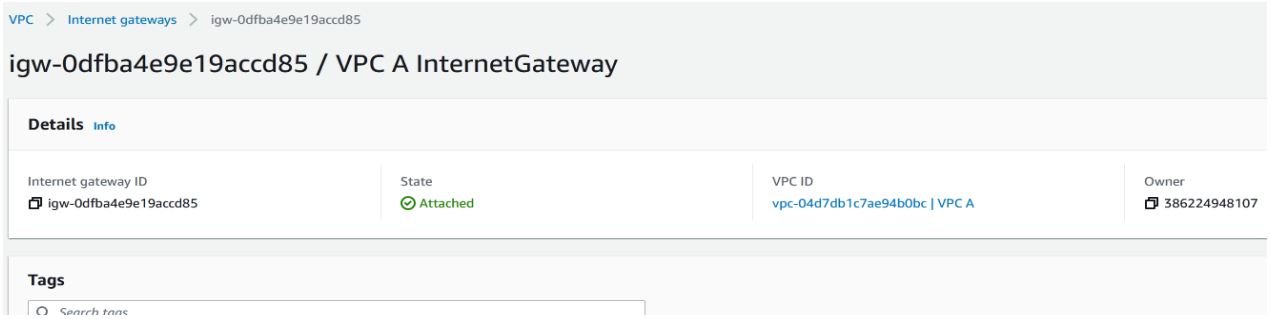
- Created a VPC - "VPC A"



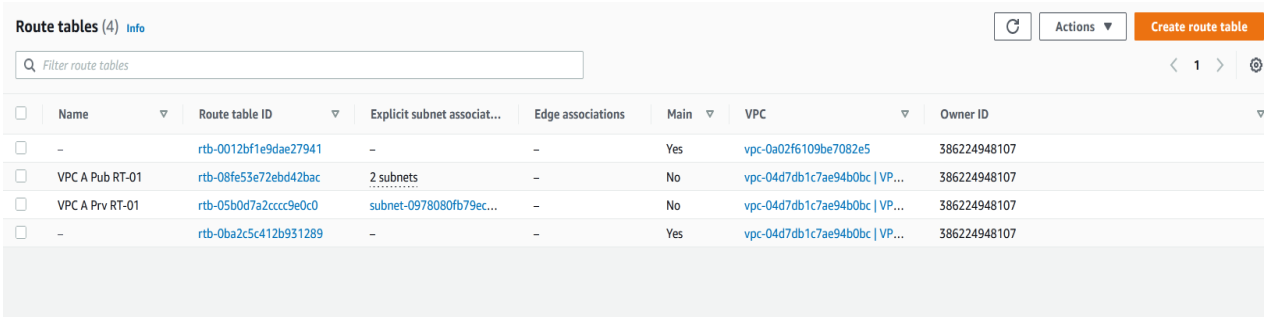
- Created 2 Subnets under VPC A (from below screenshot 1st & 3rd - 1Public & 1Private)

VPC A-PubSubnet01	subnet-010b4b43c7556d504	Available	vpc-04d7db1c7ae94b0bc | VP...	10.100.0.0/24	-	251	us-
VPC A-PubSubnet02	subnet-0b4d60550a1f3ea8d	Available	vpc-04d7db1c7ae94b0bc | VP...	10.100.2.0/24	-	251	us-
VPC A-PrvSubnet01	subnet-0978080fb79ec3179	Available	vpc-04d7db1c7ae94b0bc | VP...	10.100.1.0/24	-	251	us-

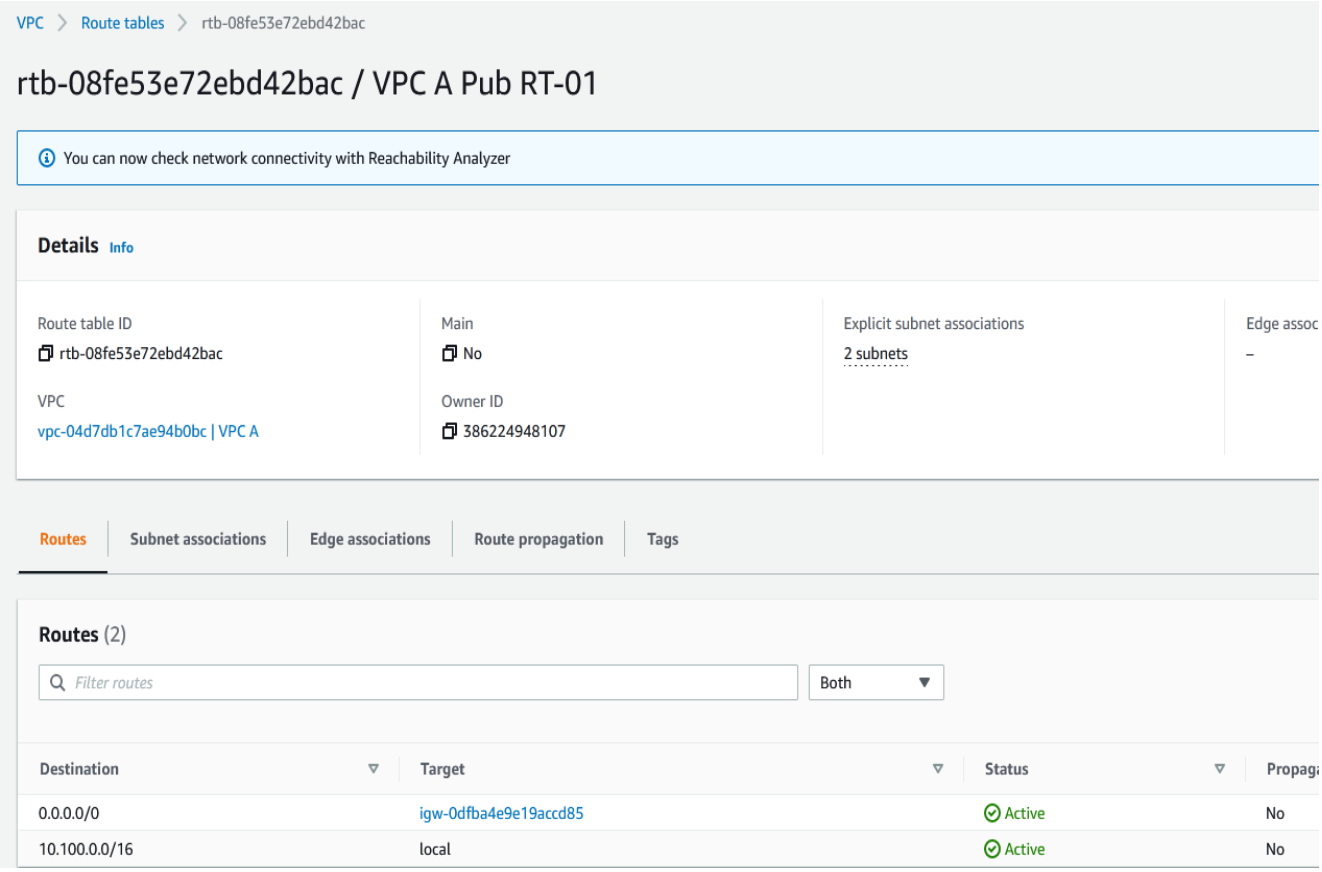
- Created Internet gateway & attached to VPC A



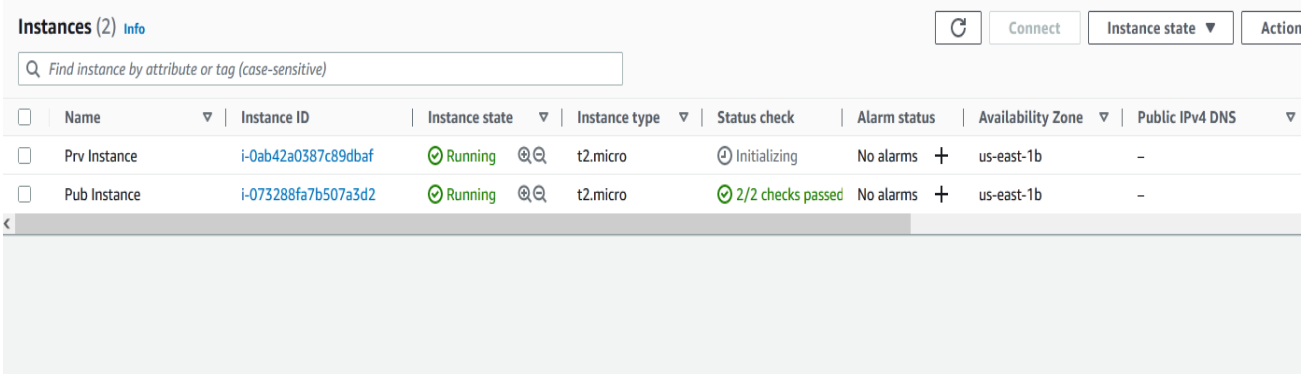
- 2Route Tables created & associated with respective subnets (i.e., public route table to public subnet & Private route table to private subnet)



- Now associated Public Route Table to Internet Gateway



- Now Created 2-Instances (1-Public & 1-Private) by associating public instance with public subnet & private instance with private subnet



- Now connect the public instance……it pings if we try to ping private subnet instance (#ping Private IP Address) with private IP address (below screenshot is copied one)

```
[ec2-user@ip-10-100-0-207 ~]$ sudo su
[root@ip-10-100-0-207 ec2-user]# ping 10.100.1.176
PING 10.100.1.176 (10.100.1.176) 56(84) bytes of data.
64 bytes from 10.100.1.176: icmp_seq=1 ttl=255 time=1.29 ms
64 bytes from 10.100.1.176: icmp_seq=2 ttl=255 time=0.718 ms
64 bytes from 10.100.1.176: icmp_seq=3 ttl=255 time=0.719 ms
64 bytes from 10.100.1.176: icmp_seq=4 ttl=255 time=0.707 ms
64 bytes from 10.100.1.176: icmp_seq=5 ttl=255 time=0.703 ms
^C
```

- Now need to copy the key file to do SSH

- We can connect Private instance from public instance

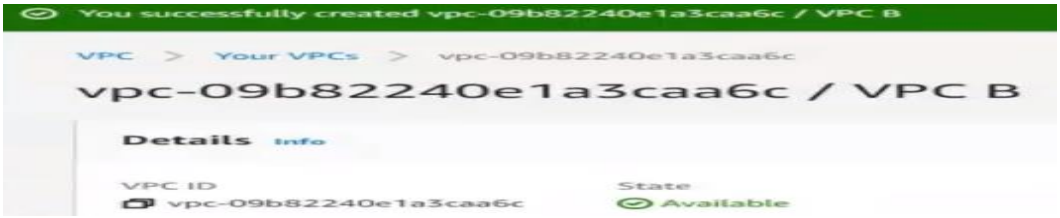
```
[root@ip-10-100-0-207 ec2-user]# ls -lt
total 4
-rwxrwxrwx 1 ec2-user ec2-user 1674 Oct
[root@ip-10-100-0-207 ec2-user]# ssh -i

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-100-1-176 ~]$
```

- We can't ping from private subnet as we don't have internet gateway connection

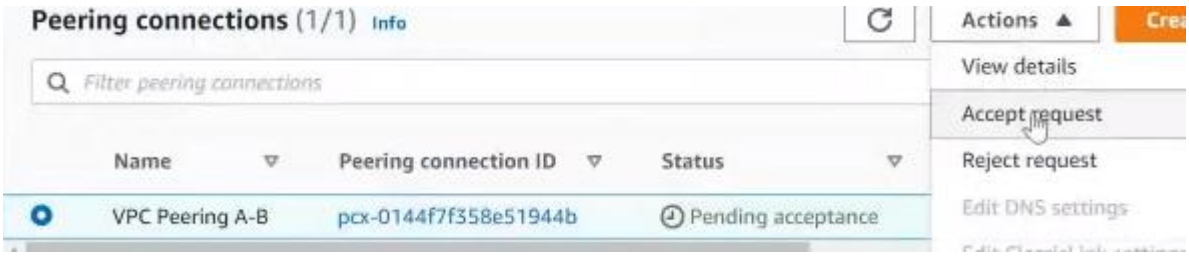
- Now create another VPC - VPC B



- Create a Private subnet & Private route table on VPC B

- Create an Instance of VPC B

- To establish a connection between Private instances of VPC A & VPC B we need to create Peering connection by selecting requester as VPC A & Peer with as VPC B & accept peering connection



- Need to edit route tables of private instances to establish a connection between them & by copying pem file from public instance to private instance of VPC A
- Now connection can be established by doing SSH from private instance of VPC A to private instance of VPC B.