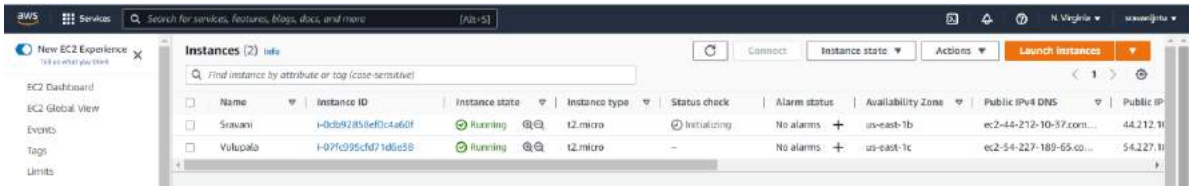
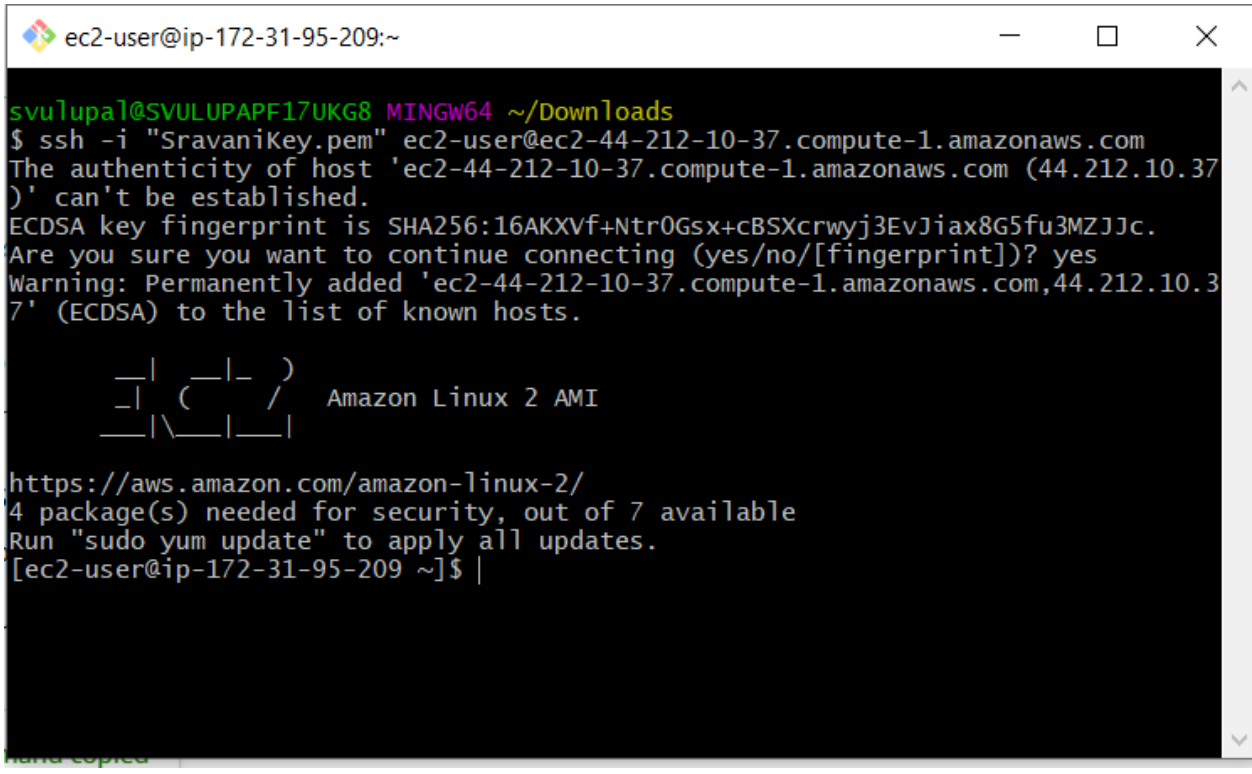
AWS Cloud EC2

1. EC2 instance creation

a. List of instances created with first name and last name:

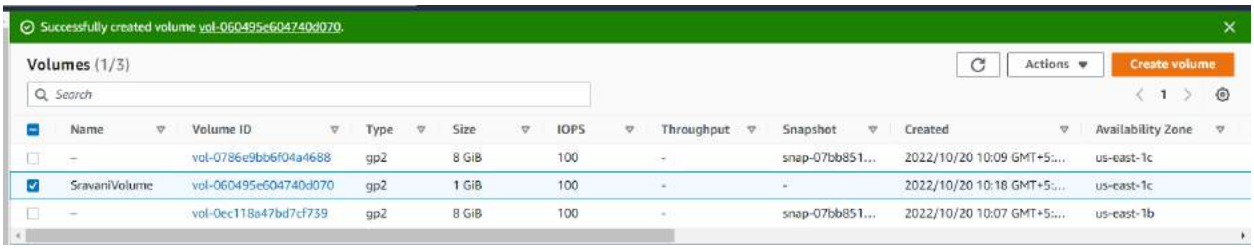


b. Connect to the first instance using key pair:

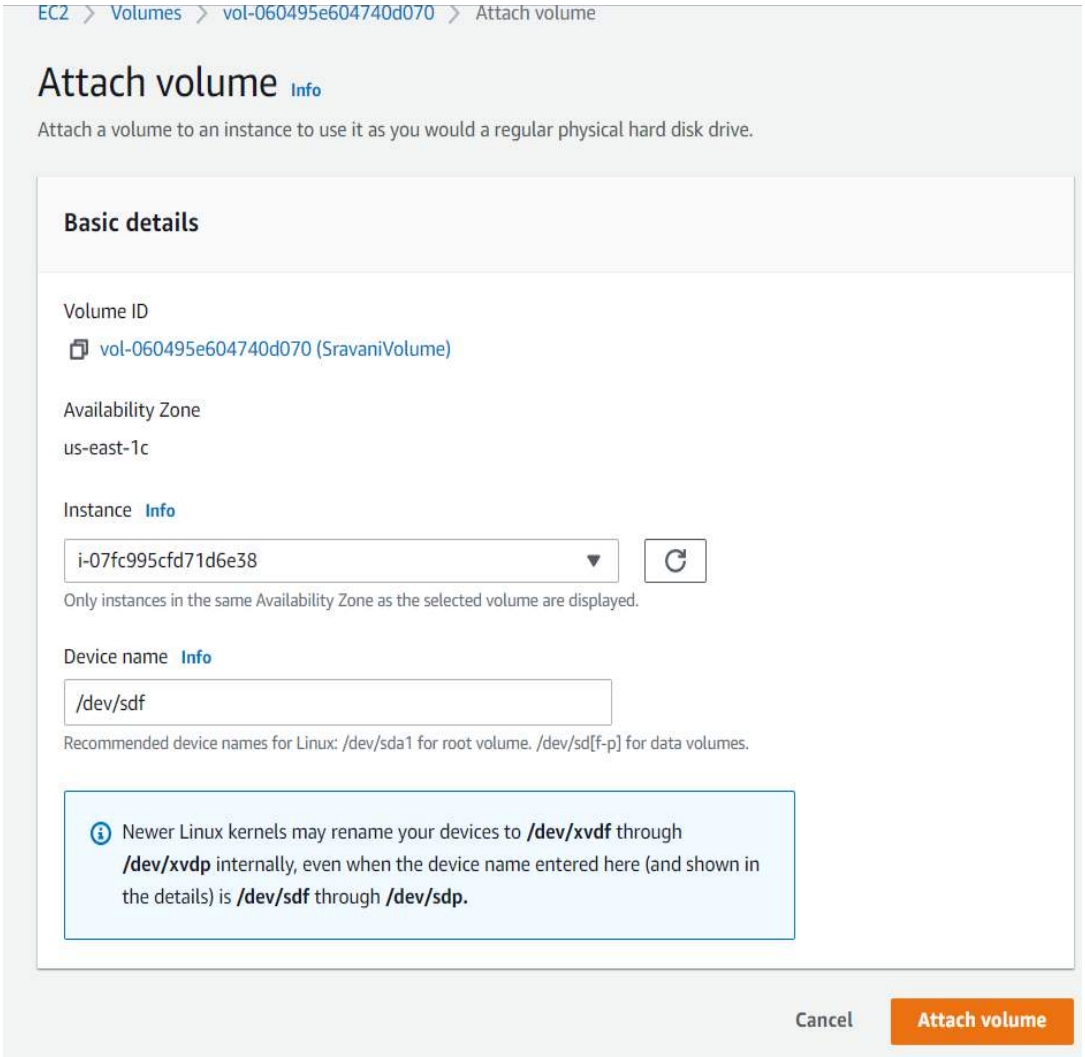


2. EBS (Elastic Block Storage)

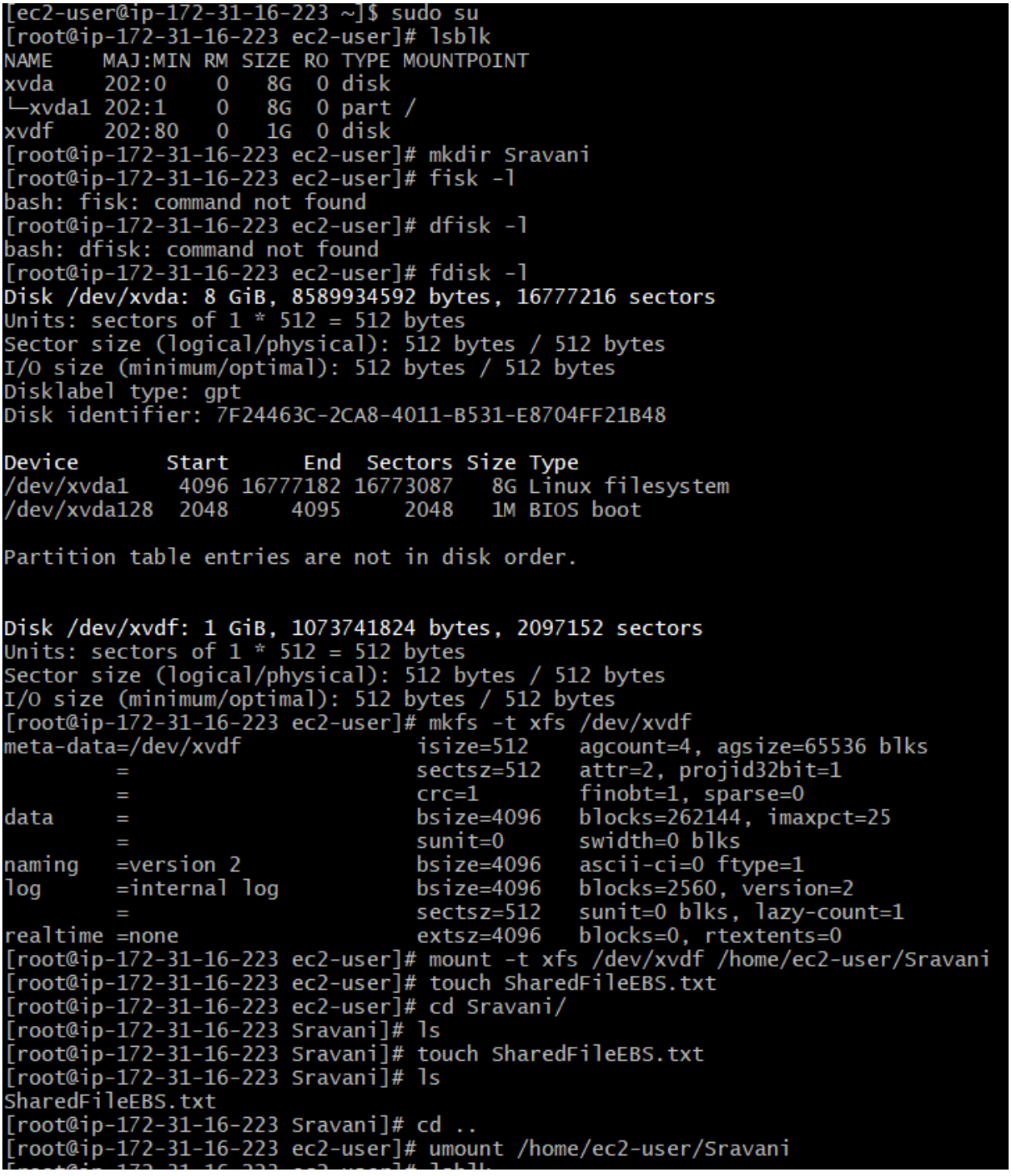
a. Create a Volume from Volumes menu:



b. Attach volume to EC2 instance:

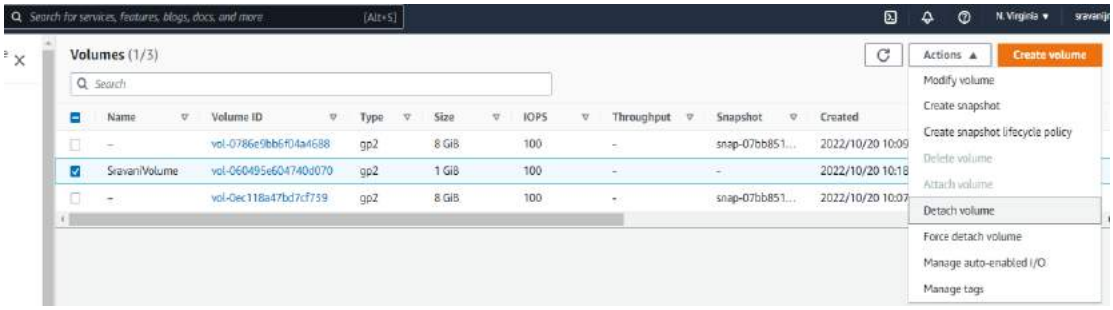


c. Make a file system after attaching the volume, mount file system, create files and un mount :

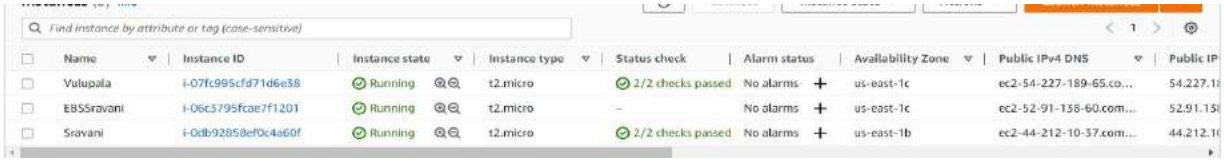


```
[ec2-user@ip-172-31-16-223 ~]$ sudo su
[root@ip-172-31-16-223 ec2-user]# lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda    202:0    0   8G  0 disk
└─xvda1 202:1    0   8G  0 part /
xvdf    202:80   0   1G  0 disk
[root@ip-172-31-16-223 ec2-user]# mkdir Sravani
[root@ip-172-31-16-223 ec2-user]# fisk -l
bash: fisk: command not found
[root@ip-172-31-16-223 ec2-user]# dfisk -l
bash: dfisk: command not found
[root@ip-172-31-16-223 ec2-user]# fdisk -l
Disk /dev/xvda: 8 GiB, 8589934592 bytes, 16777216 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
Disklabel type: gpt
Disk identifier: 7F24463C-2CA8-4011-B531-E8704FF21B48

Device       Start      End  Sectors Size Type
/dev/xvda1    4096 16777182 16773087   8G Linux filesystem
/dev/xvda128  2048     4095     2048   1M BIOS boot

Partition table entries are not in disk order.


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-172-31-16-223 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-172-31-16-223 ec2-user]# mount -t xfs /dev/xvdf /home/ec2-user/Sravani
[root@ip-172-31-16-223 ec2-user]# touch SharedFileEBS.txt
[root@ip-172-31-16-223 ec2-user]# cd Sravani/
[root@ip-172-31-16-223 Sravani]# ls
[root@ip-172-31-16-223 Sravani]# touch SharedFileEBS.txt
[root@ip-172-31-16-223 Sravani]# ls
SharedFileEBS.txt
[root@ip-172-31-16-223 Sravani]# cd ..
[root@ip-172-31-16-223 ec2-user]# umount /home/ec2-user/Sravani
```

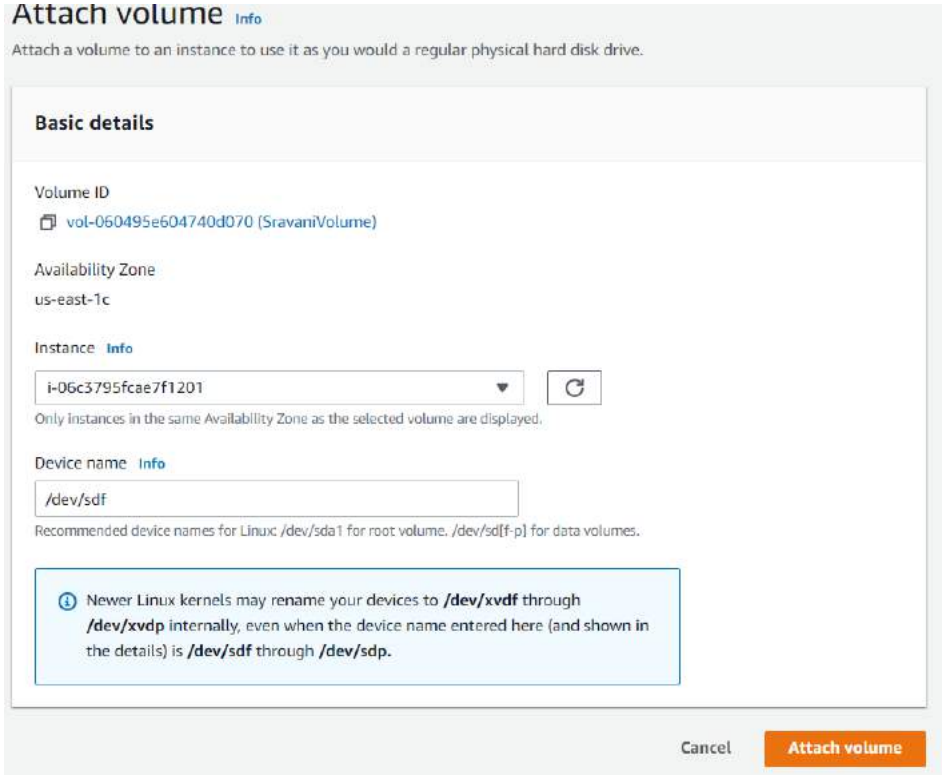
d. Detach the volume from the instance:



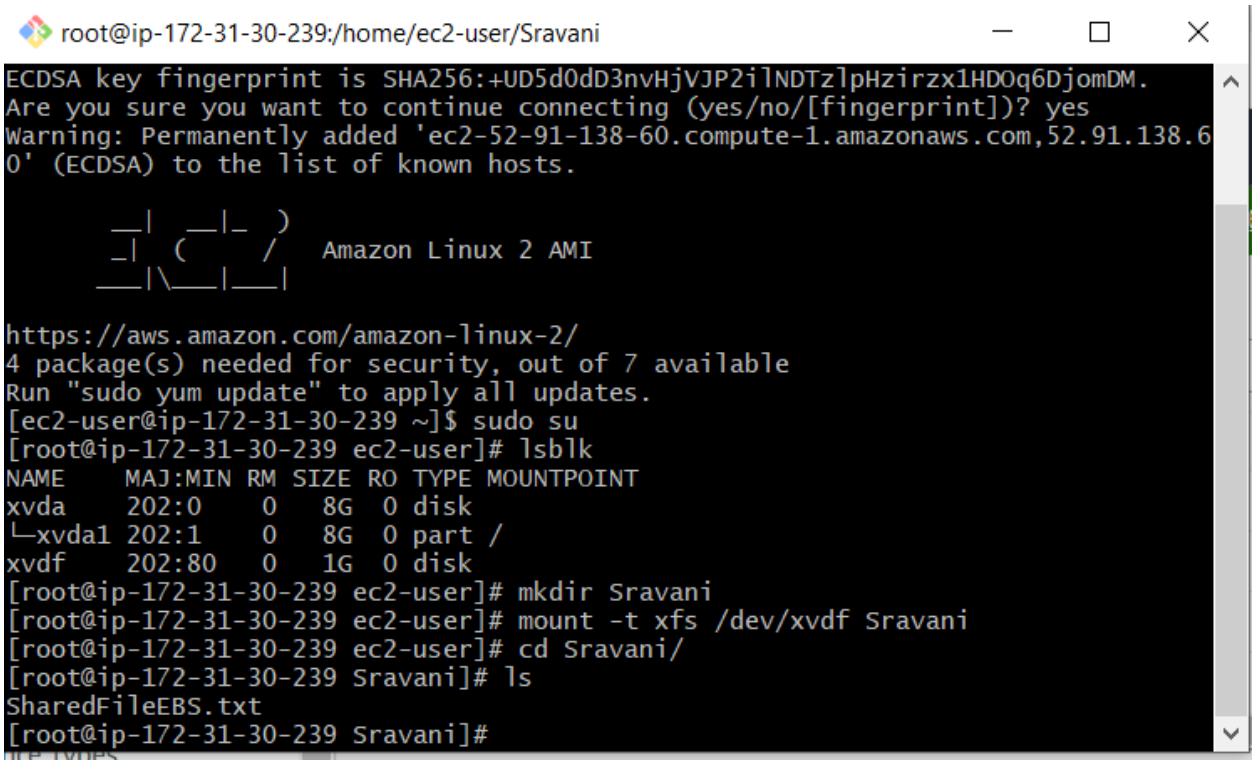
e. Create another instance in the same availability zone:



f. Attach the same volume to this instance :

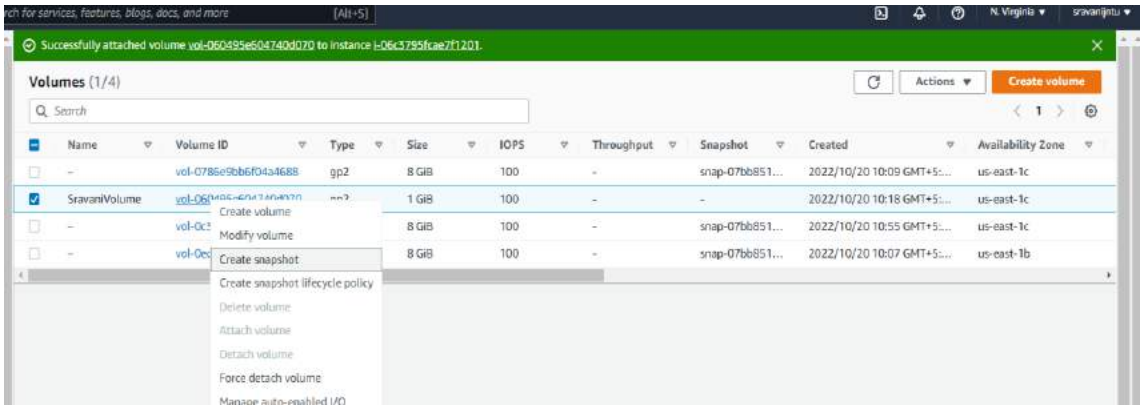


g. Without making a file system, mount the same one and notice the file that we created in the first machine can be accessed here:

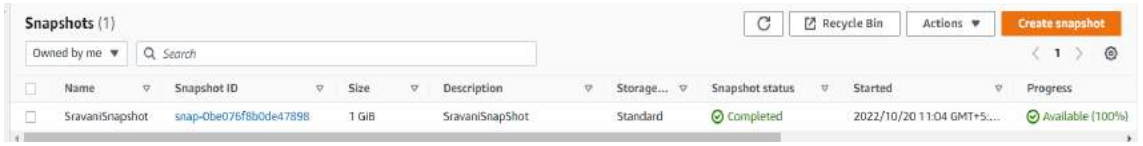


3. Snapshot

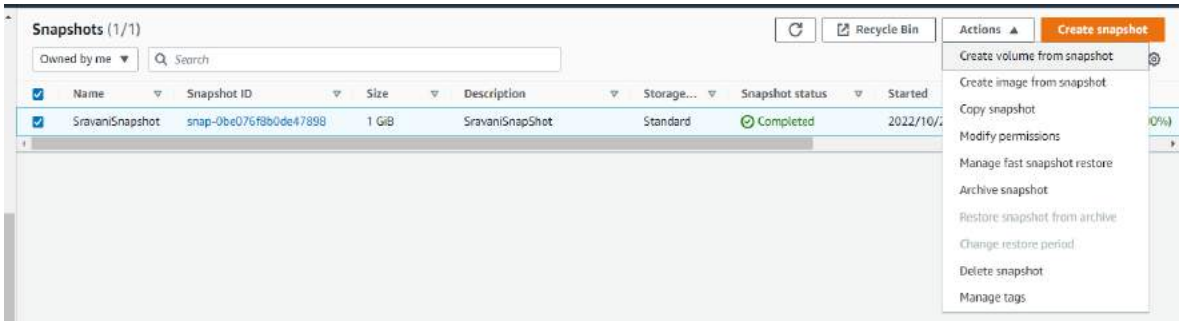
a. Create snapshot from volume:



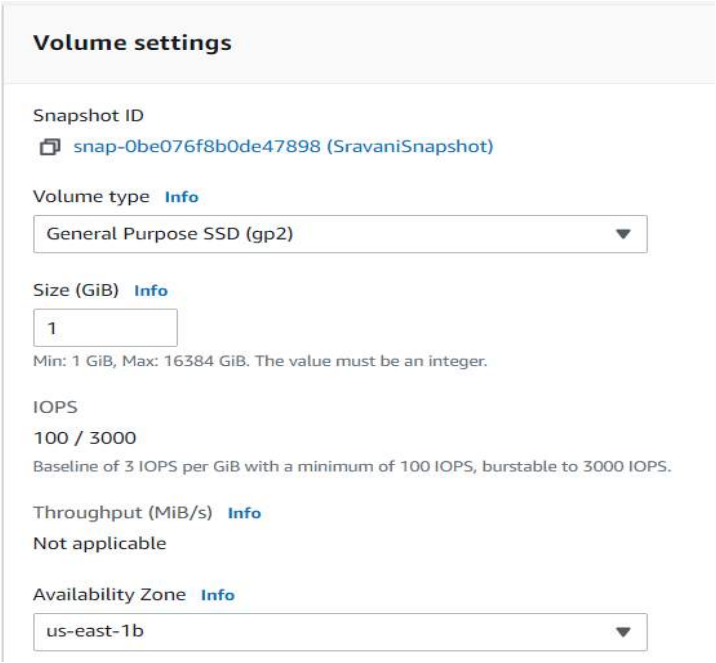
b. Snapshots list:



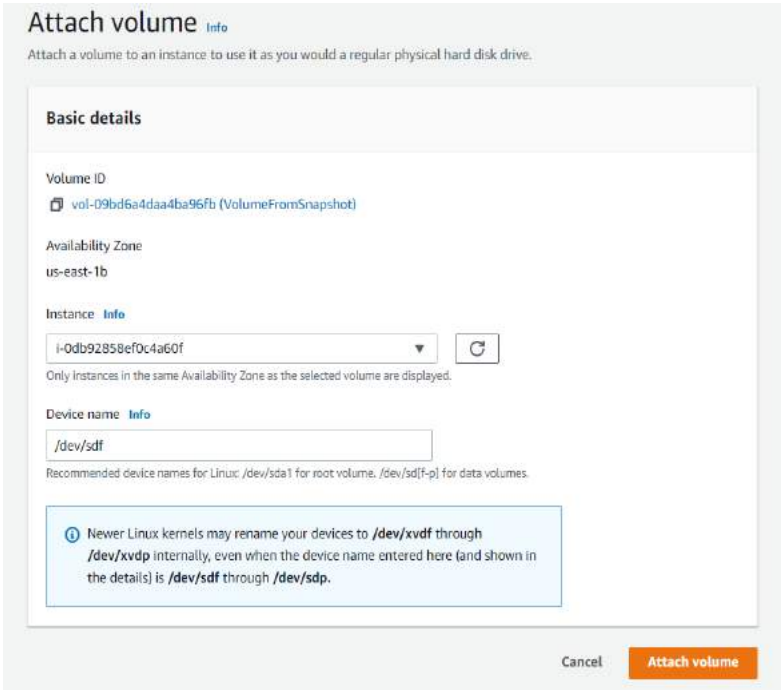
c. Create volume from snapshot:



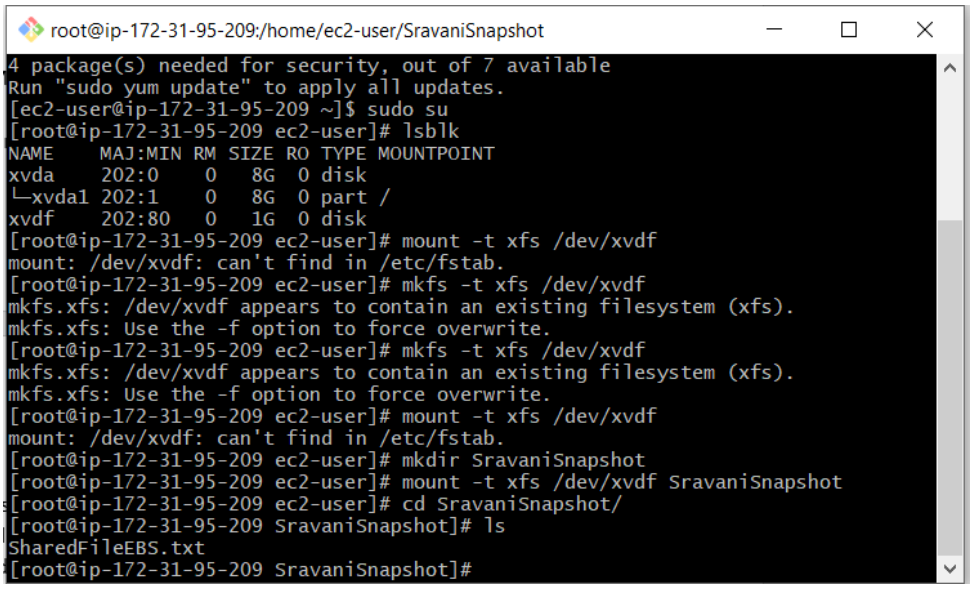
d. Select a different availability zone (selected us-eas-t1b since previously it was us-east-1c):



e. Attach the instance in the same availability zone as above to the volume created from snapshot:

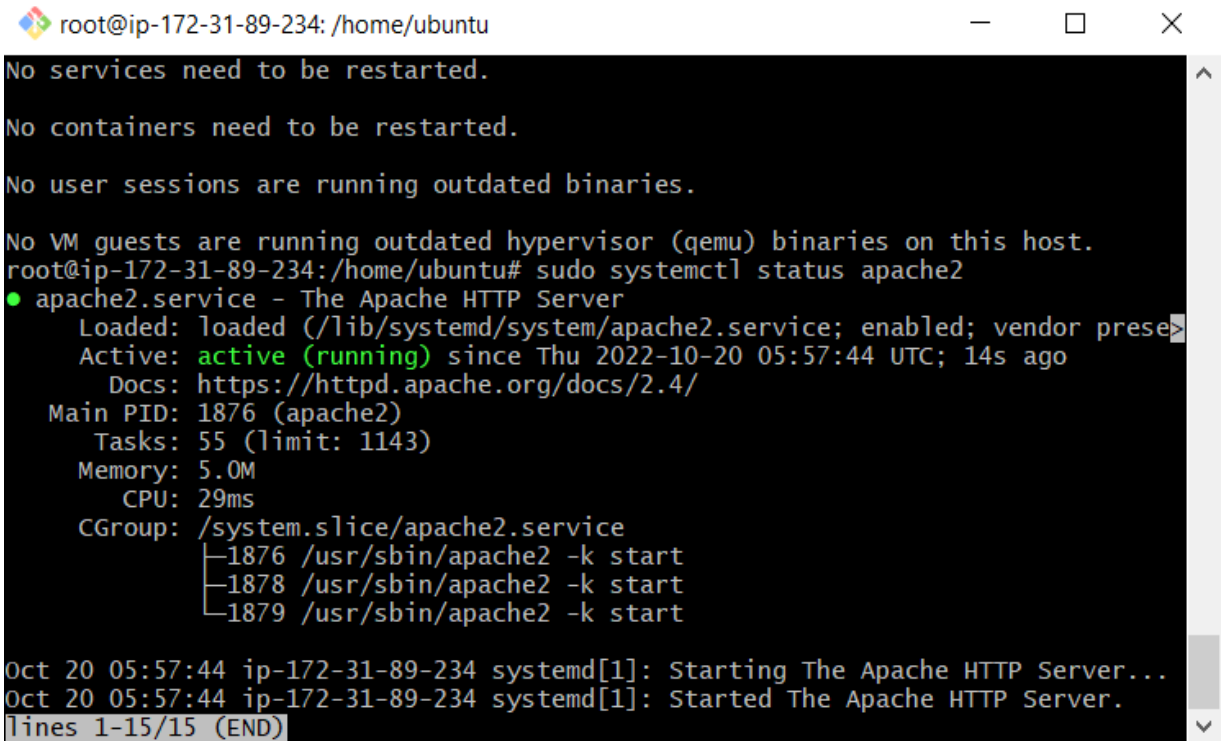


f. Login to the instance in shell and notice the previous files can be accessed without creating the file system again:

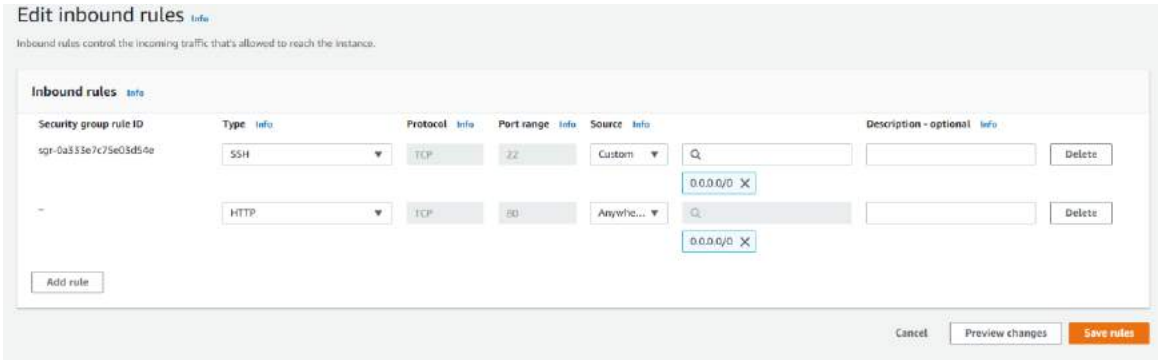


4. AMI (Amazon Machine Image)

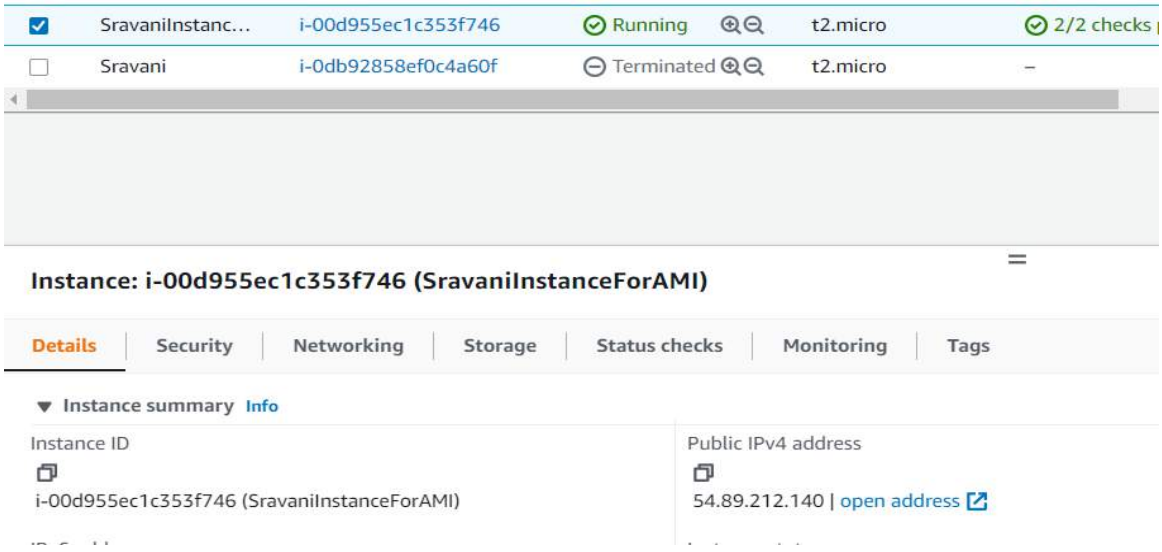
a. Create a ubuntu instance and install apache and check status:



b. Once apache s installed, to connect in browser edit the inbound rules in the security group attached to the instance:

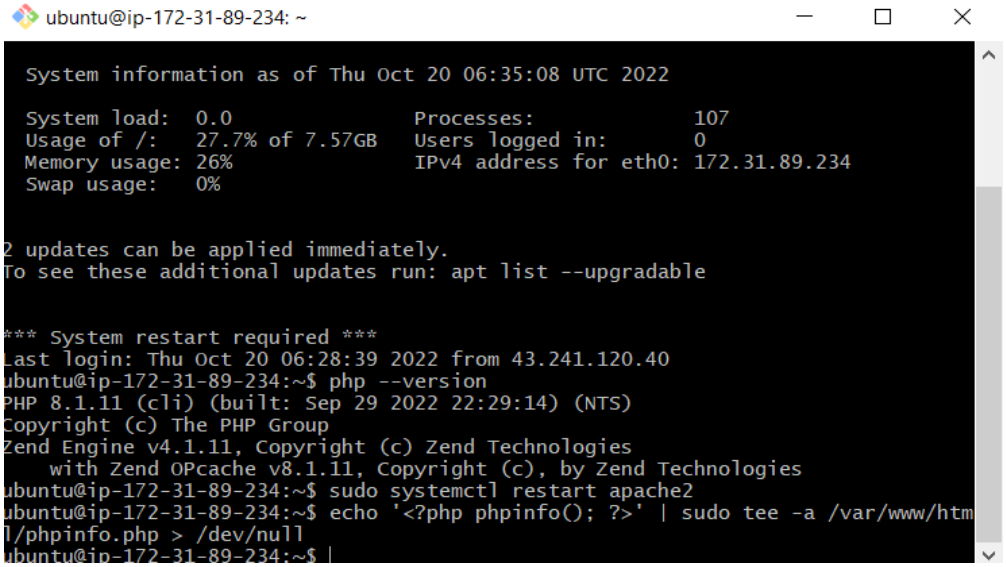


c. Get the public ip of the instance:

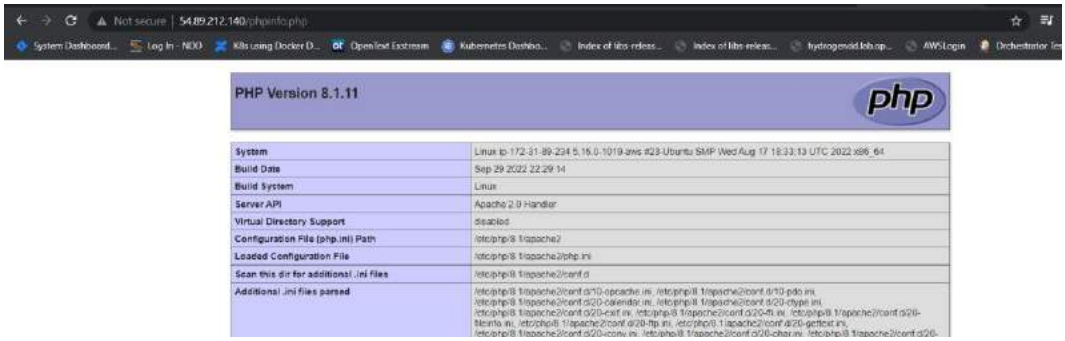


d. Connect in browser http://54.89.212.140. Notice the following is launched:

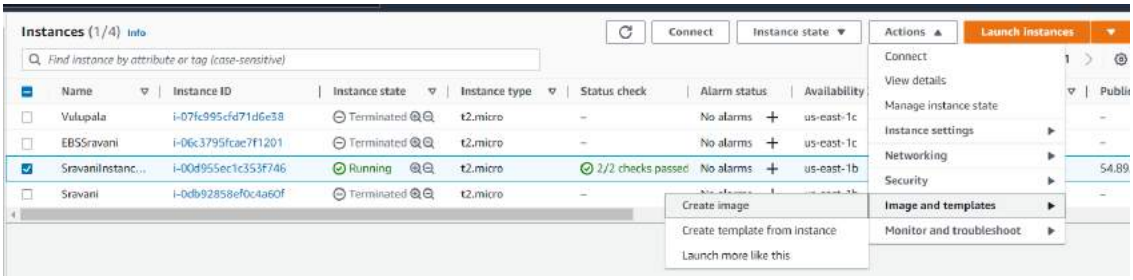
e. Install Php:

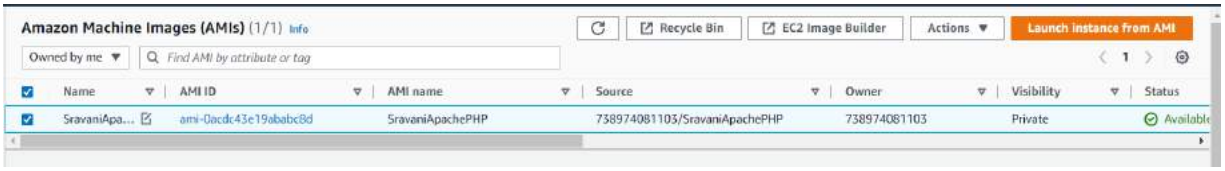


f. Install Php on the same instance and connect to the phpinfo.php page that we created:

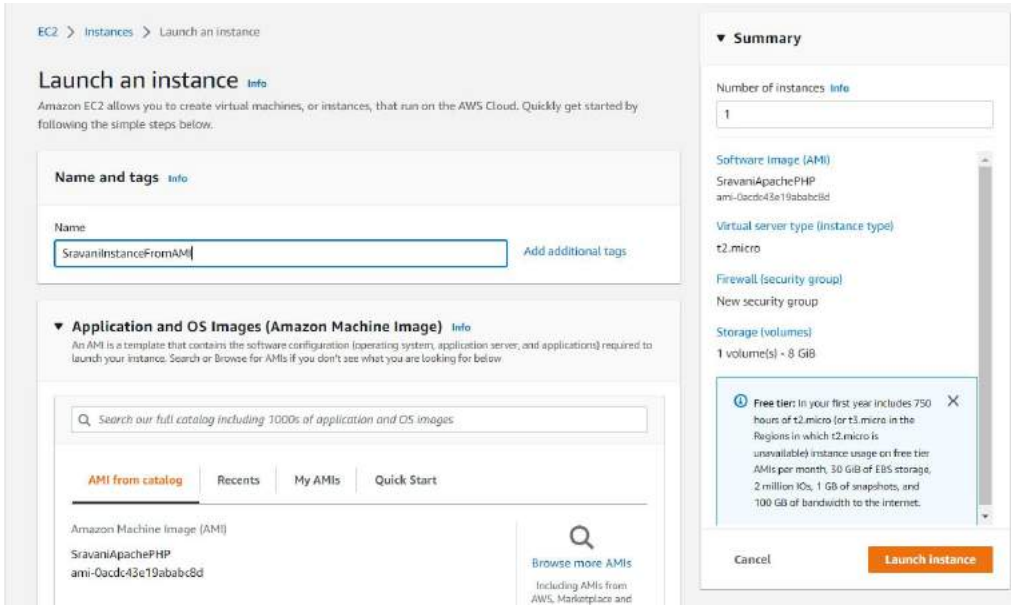


g. Create AMI from the above instance:

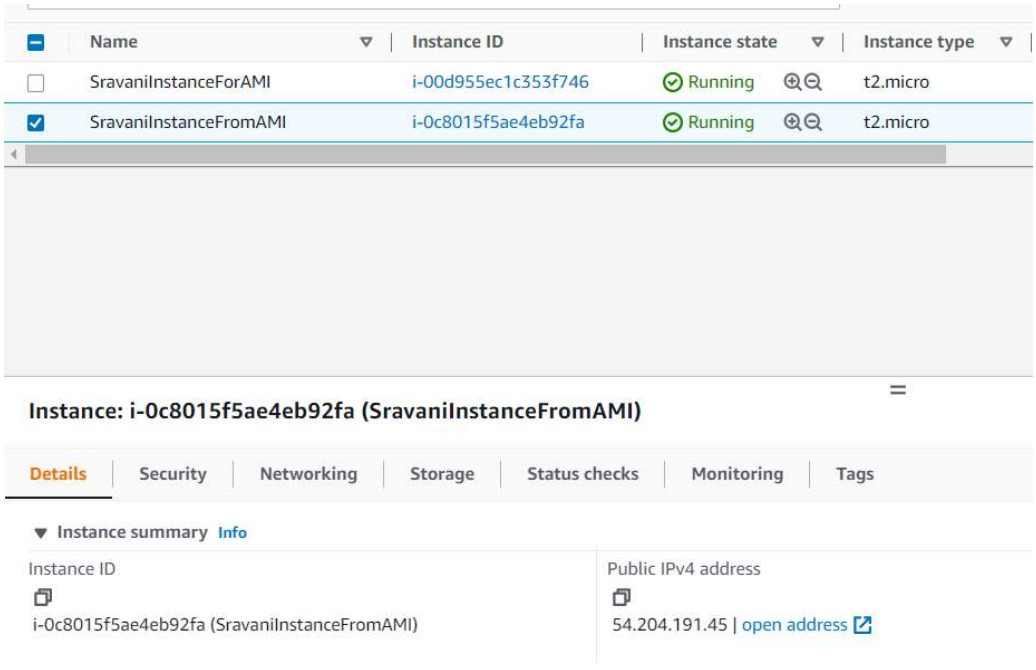




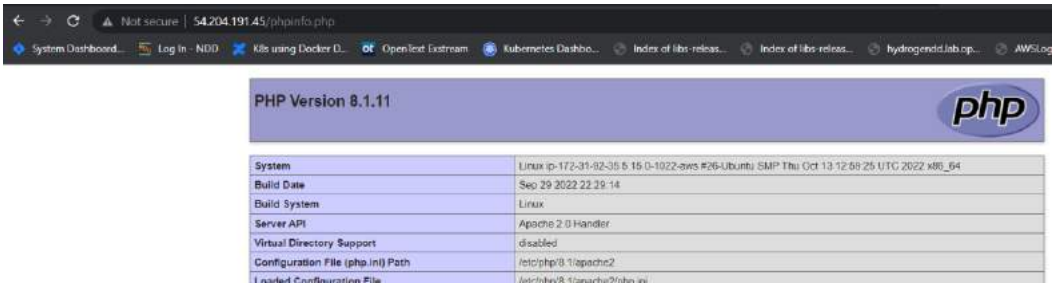
h. Create instance from the above AMI:



i. Get the ip address of the instance created and edit the security group inbound rules to allow http port 80 and connect to apache and phpinfo:



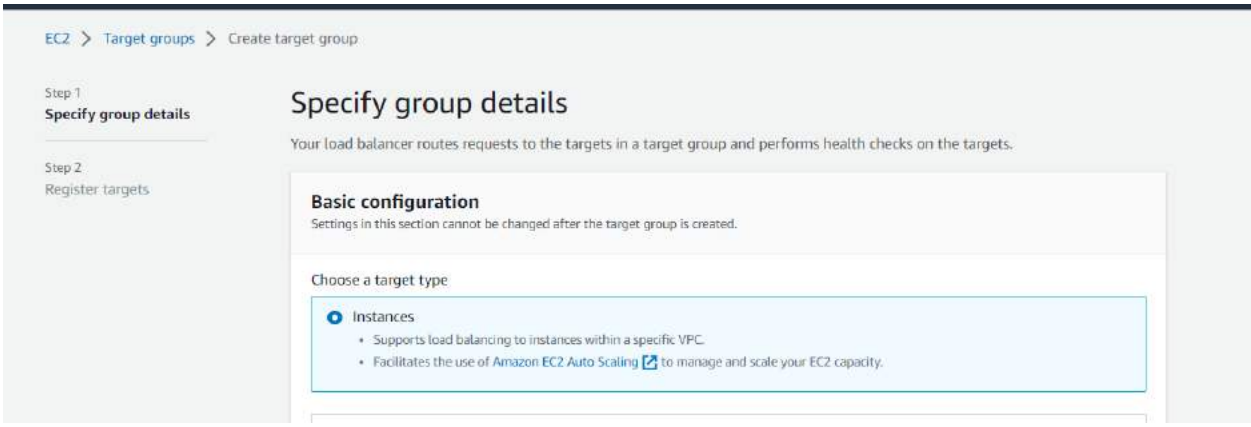

Not secure | 54.204.191.45
System Dashboard...
Log In - NDD
K8s using Docker D...
OpenText Extream
Kubernetes Dashbo...
Index of libs-releas...
Index of libs-releas...
hydrogendd.lab.op...
Apache2 Default Page
Ubuntu
It works!

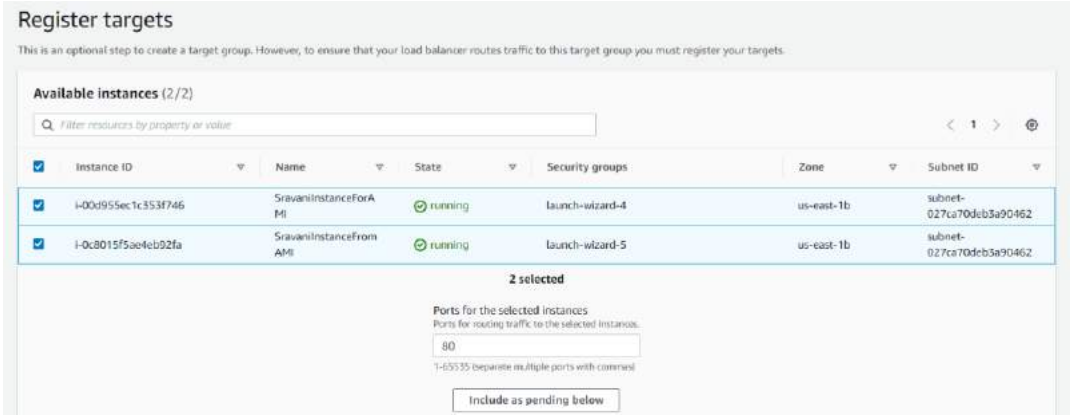

Not secure | 54.204.191.45/phpinfo.php
PHP Version 8.1.11
php
System
Linux ip-172-31-82-35 5.15.0-1022-aws #26-Ubuntu SMP Thu Oct 13 12:58:25 UTC 2022 x86_64
Build Date
Sep 29 2022 22:29:14
Build System
Linux
Server API
Apache 2.0 Handler
Virtual Directory Support
disabled
Configuration File (php.ini) Path
/etc/php/8.1/apache2

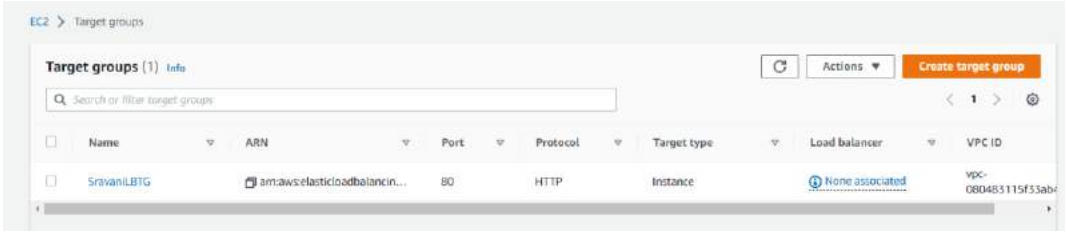
5. Load Balancer

a. Create a security group with inbound rules to allow port 80

b. Create a target group and select the two instances that you have created and click on Include as pending and click on Create target group button at the bottom

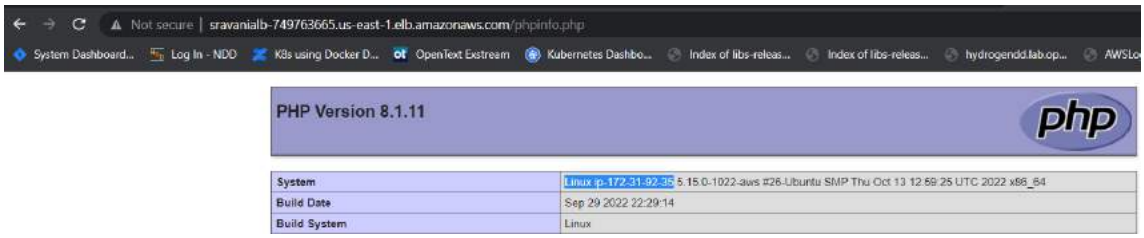


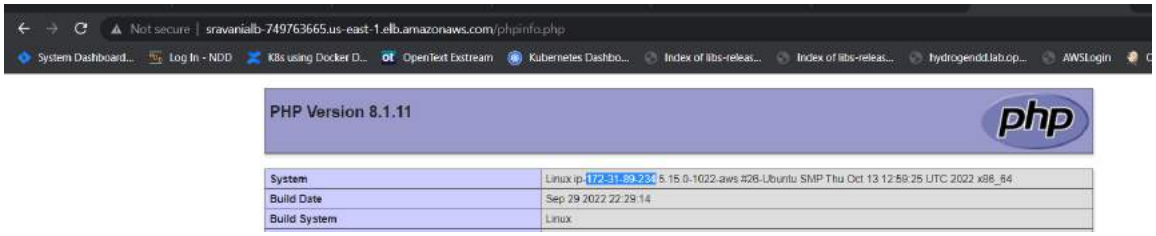




c. Create a load balancer by choosing the security group and target group created above:

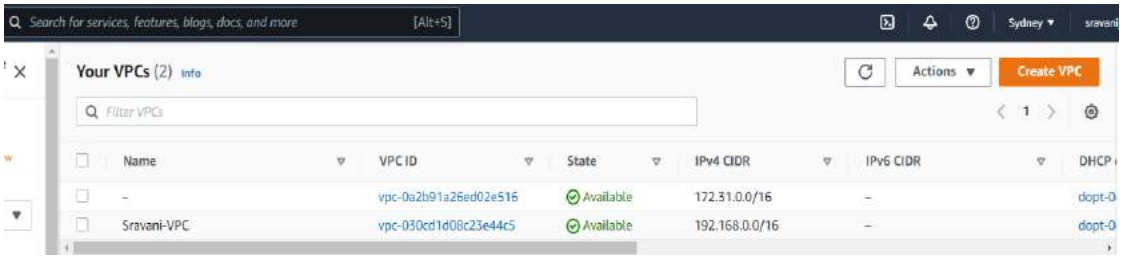
d. Private Ips of two machines : 172.31.89.234 and 172.31.92.35



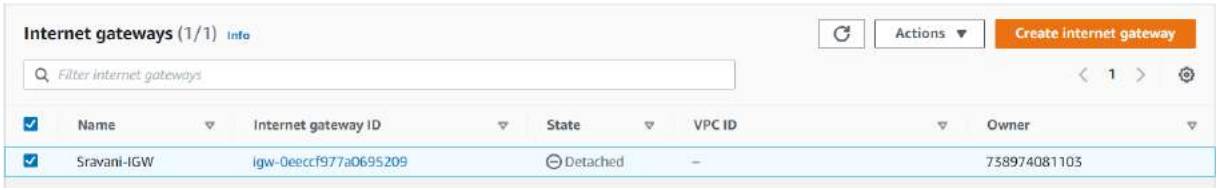


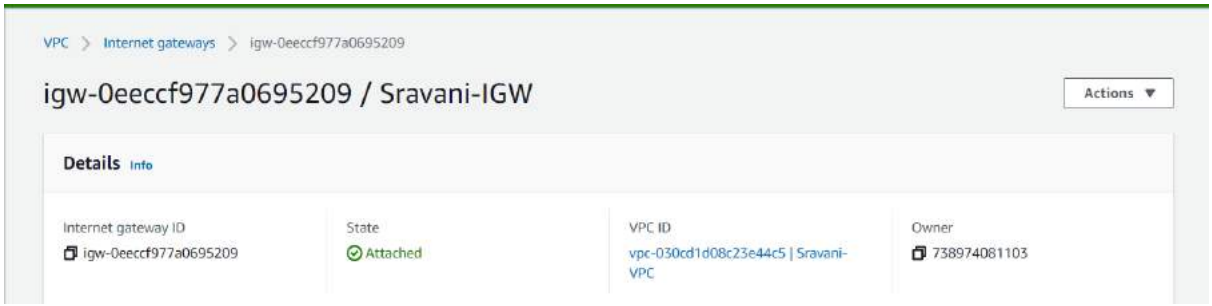
6. VPC

a. Create a VPC:

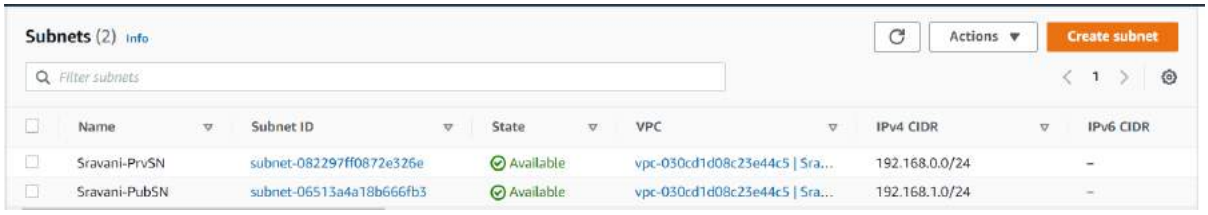


b. Create an internet gateway and attach the VPC:

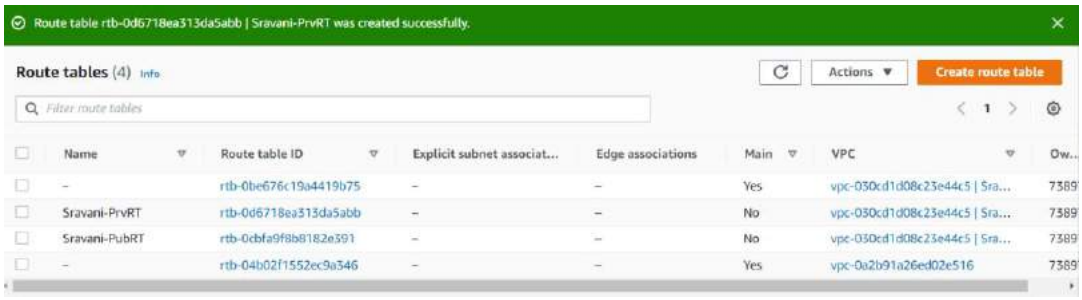




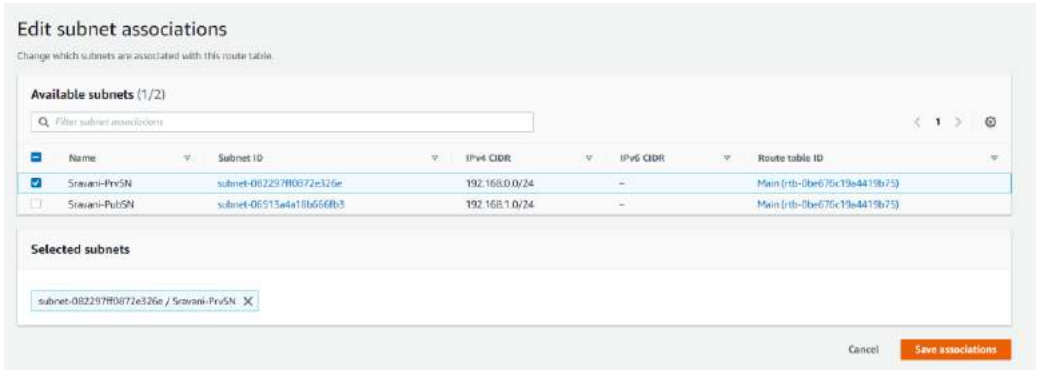
c. Create subnets (one for public subnet and one for private subnet)



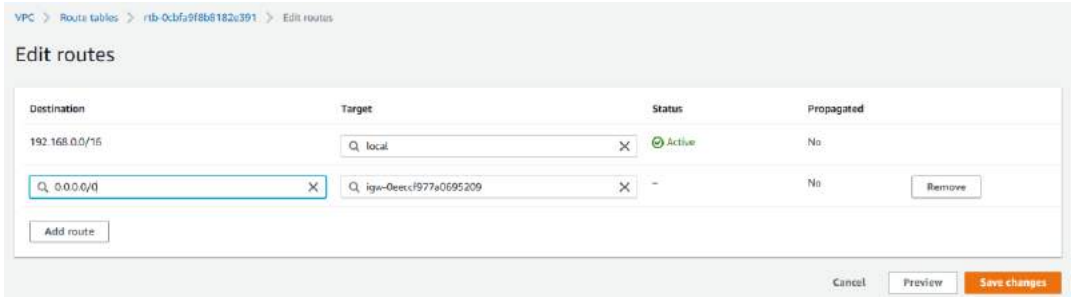
d. Create route tables (Public and Private)



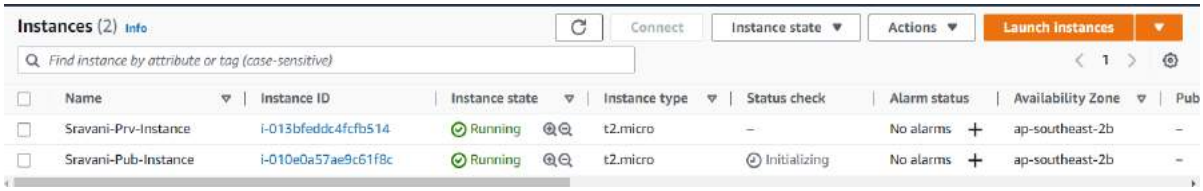
e. Edit subnet associations from route table (Public->public and Private->private)



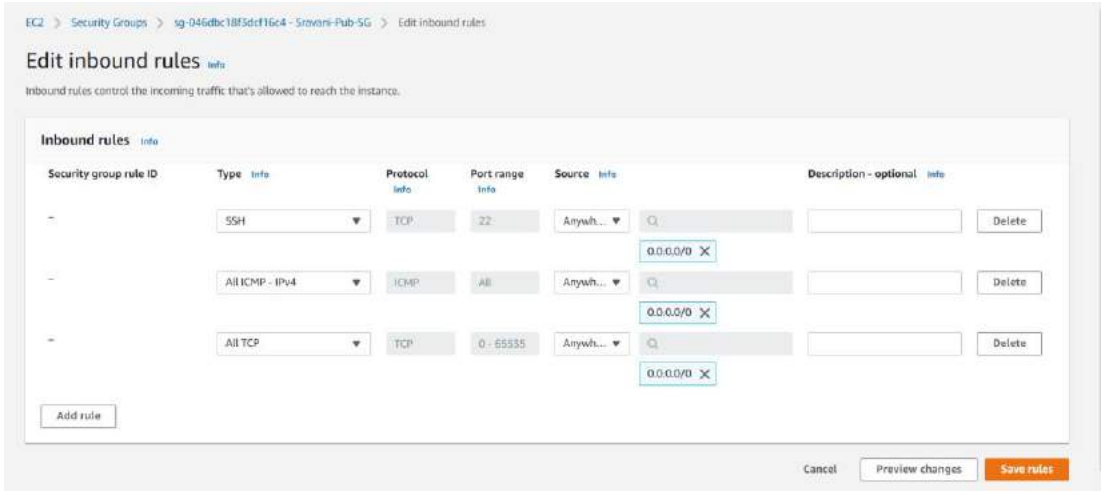
f. Connect public route table to internet gateway:



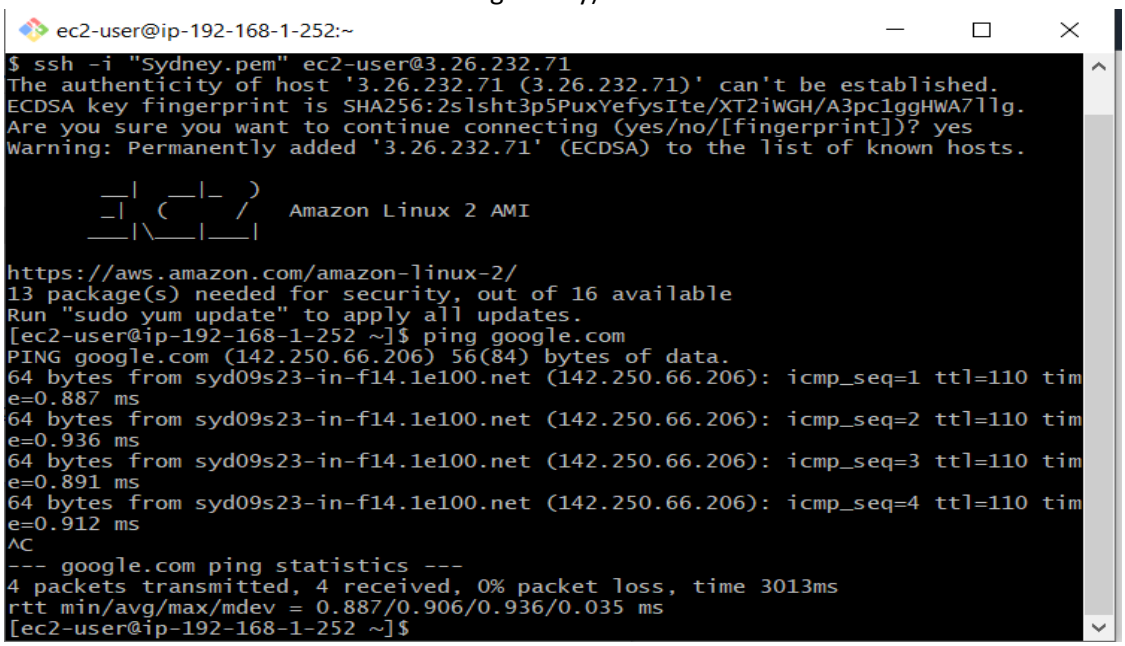
g. Create one instance each in public subnet and private subnet (Select the VPC and also the corresponding public and private subnets)



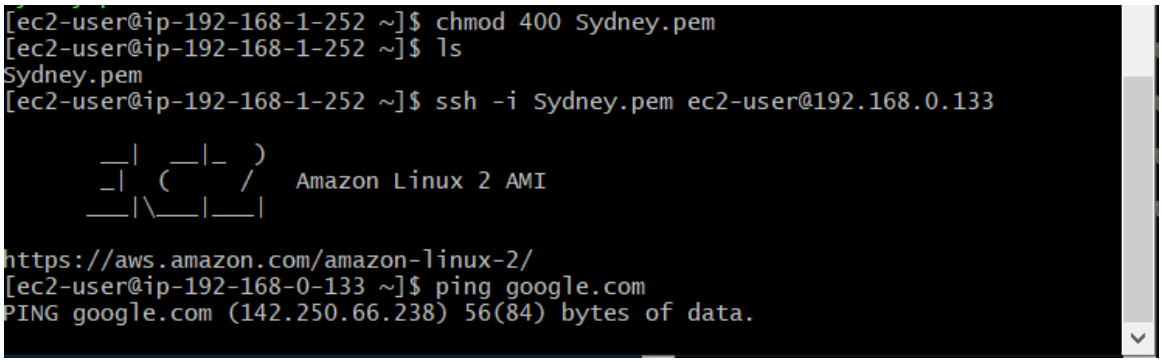
h. Edit the security groups created above to allow all ICMP and all TCP



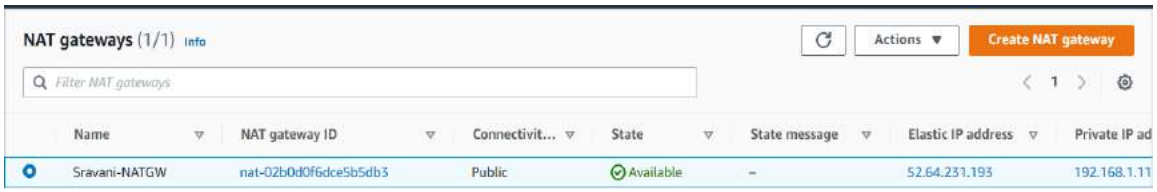
i. Connect to the instance in public subnet1 (We should be able to connect and ping internet since it is connected to internet gateway)



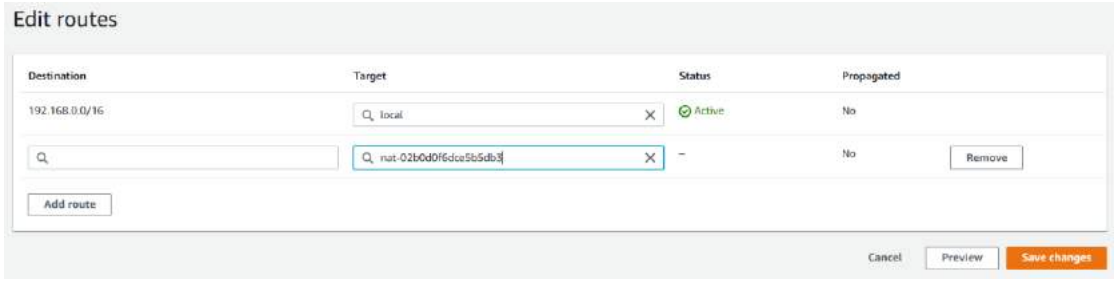
j. To connect to the instance in private subnet1, copy the key file to public subnet and connect to the private instance from there using the public ip of the private subnet instance but you will not be able to connect to internet still since NAT gateway is not created:



k. Create a NAT gateway and associate public subnet



l. Go to the private route table and add NAT gateway



m. Ping google.com from private subnet instance: