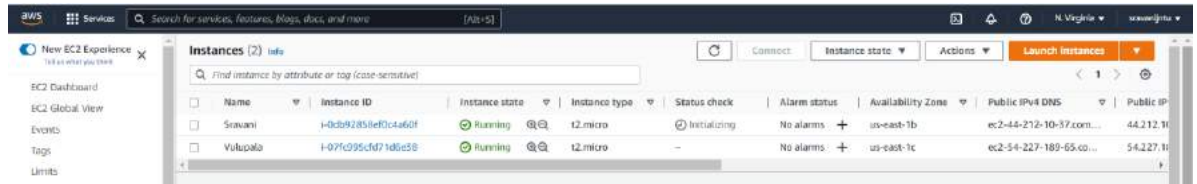


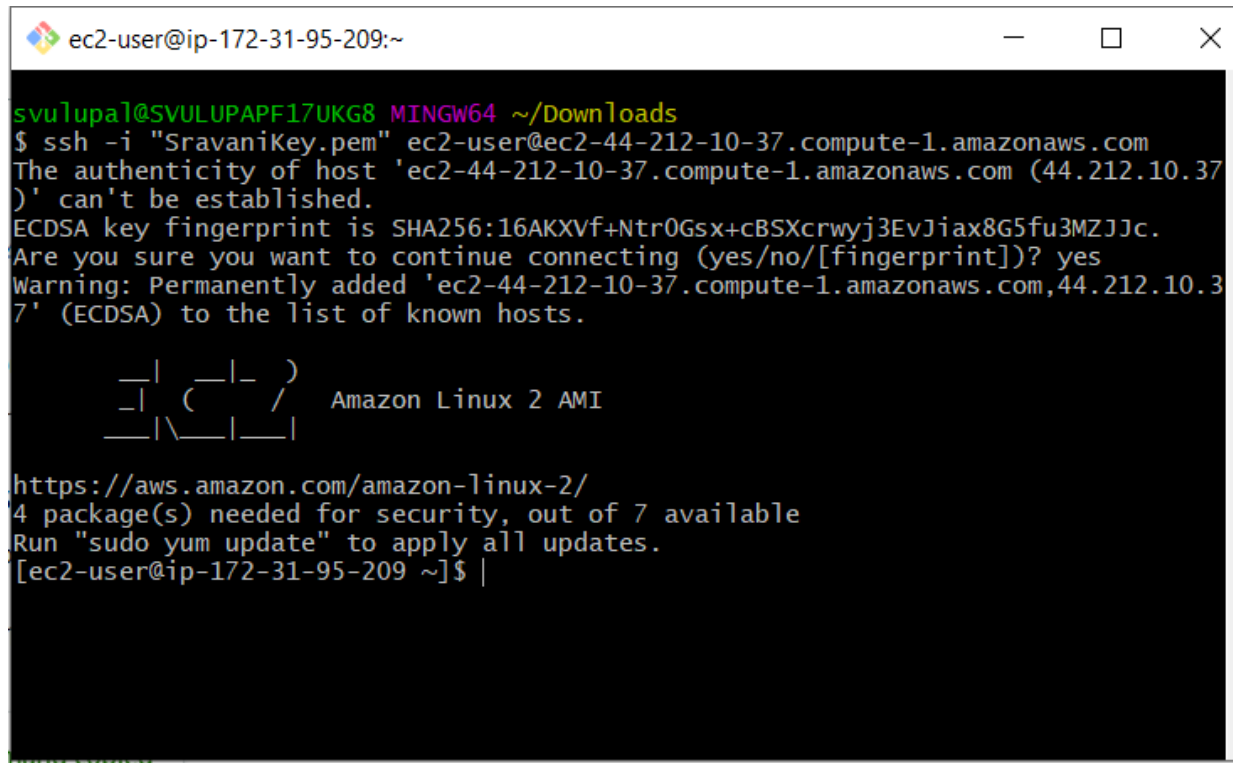
# AWS Cloud EC2

## 1. EC2 instance creation

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

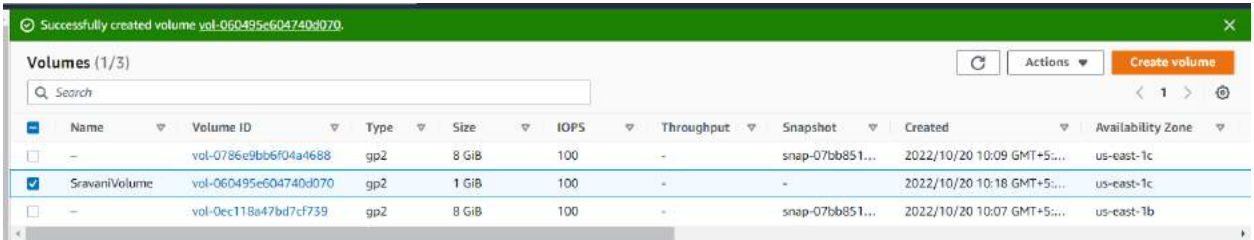


- b. Connect to the first instance using key pair:

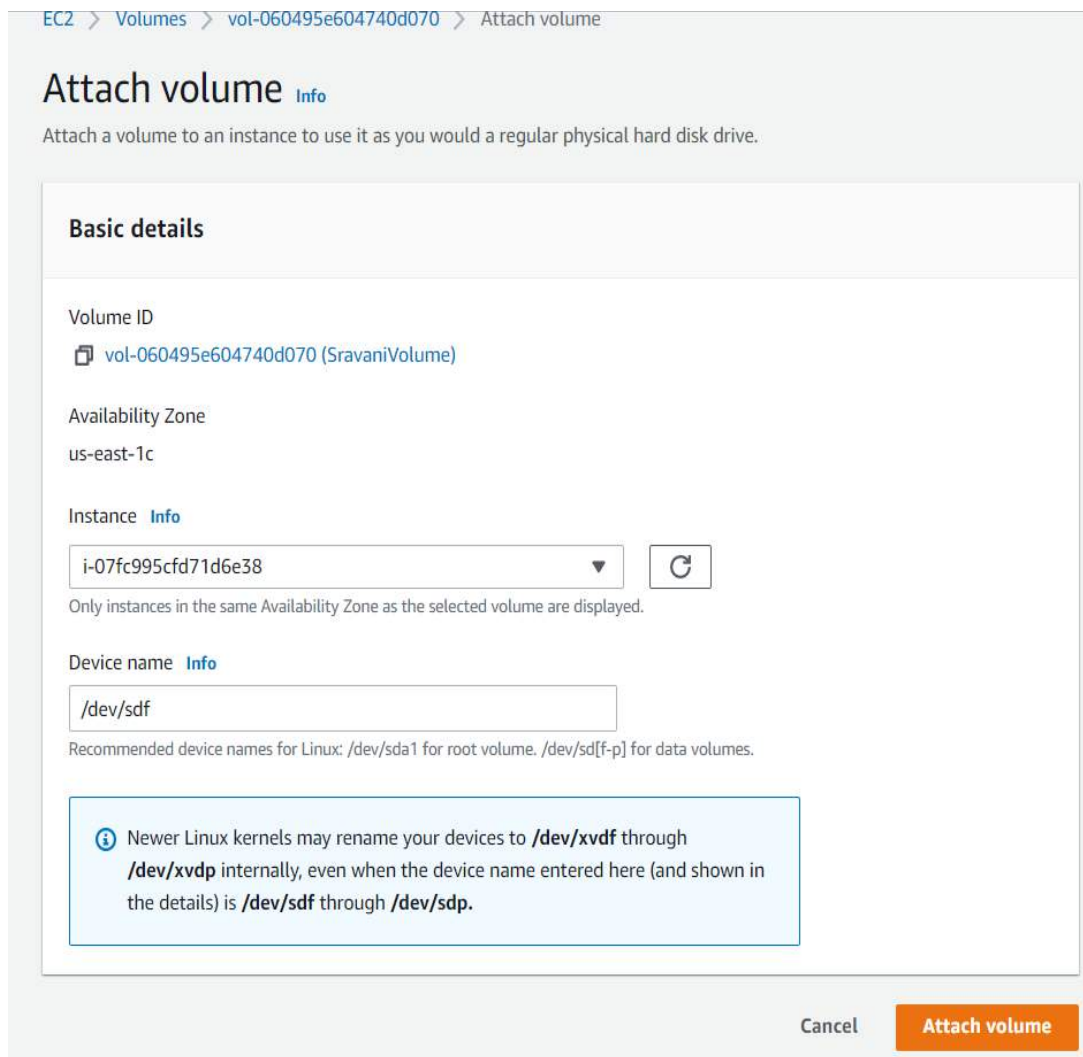


## 2. EBS (Elastic Block Storage)

- 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 unmount :

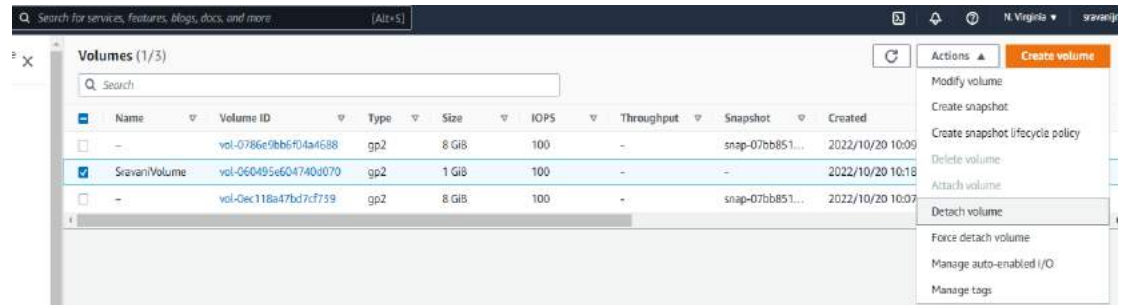
```
[ec2-user@ip-172-31-16-223 ~]$ sudo su
[root@ip-172-31-16-223 ec2-user]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda        202:0    0   8G  0 disk
└─xvda1     202:1    0   8G  0 part /
xvdf        202:80   0   1G  0 disk
[root@ip-172-31-16-223 ec2-user]# mkdir Sravani
[root@ip-172-31-16-223 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

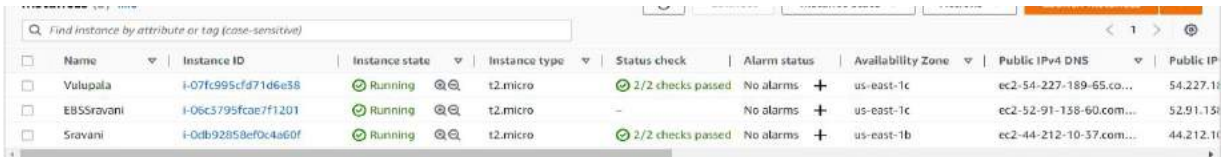
Partition table entries are not in disk order.

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
[root@ip-172-31-16-223 ec2-user]# lsblk
```

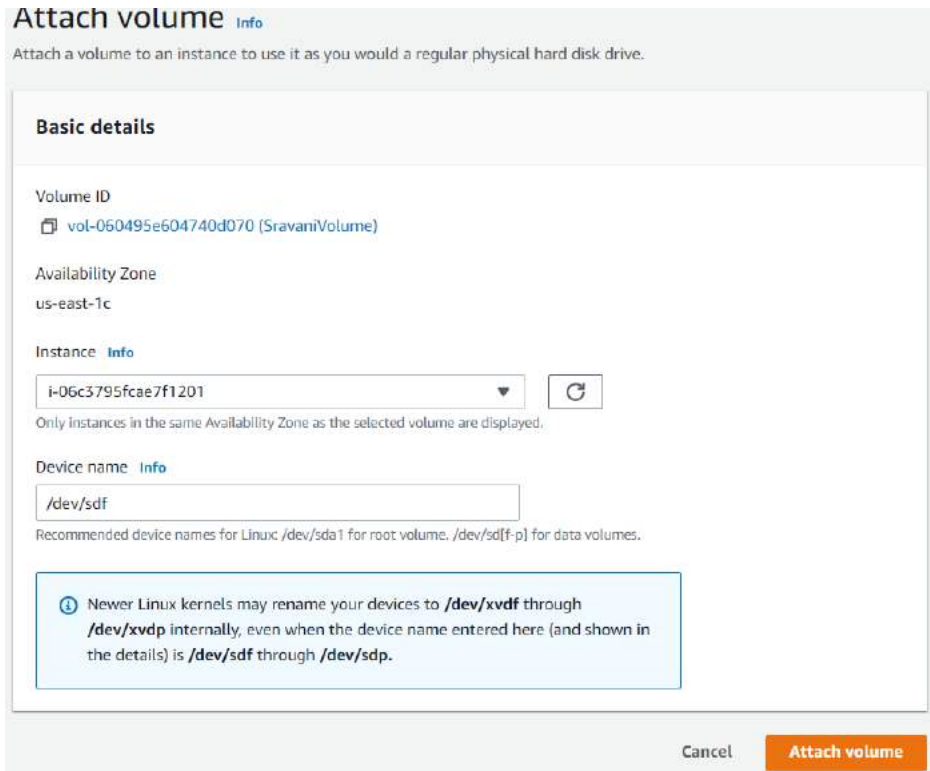
d. Detach the volume from the instance:



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:

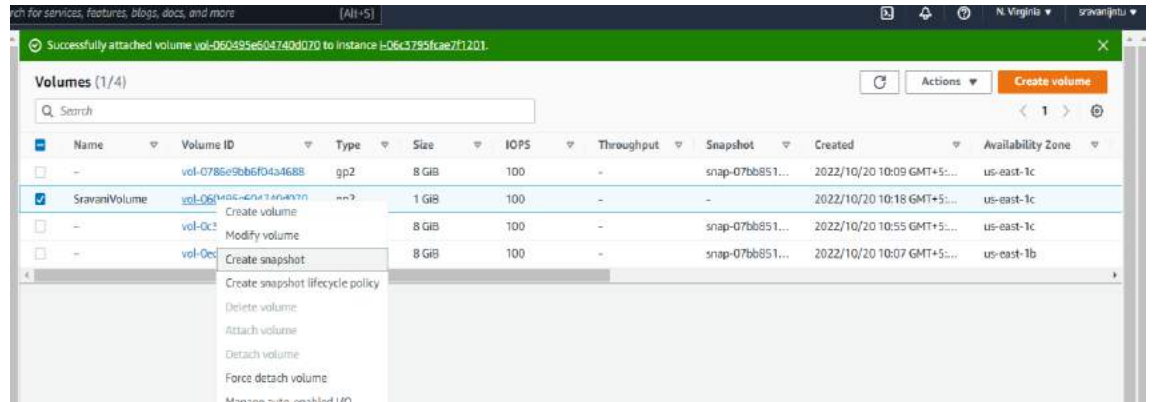
```
root@ip-172-31-30-239:/home/ec2-user/Sravani
ECDSA key fingerprint is SHA256:+UD5d0dD3nvHjVJP2ilNDTzlpHzirzx1HD0q6DjomDM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-52-91-138-60.compute-1.amazonaws.com,52.91.138.60' (ECDSA) to the list of known hosts.

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

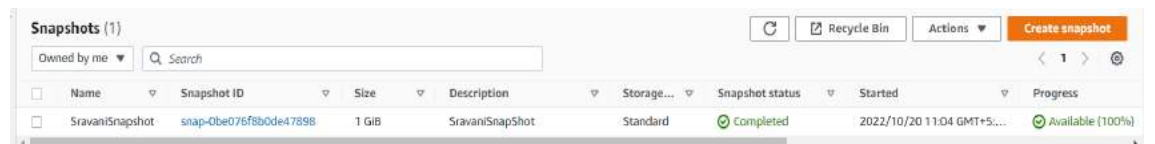
https://aws.amazon.com/amazon-linux-2/
4 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-30-239 ~]$ sudo su
[root@ip-172-31-30-239 ec2-user]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda         202:0    0   8G  0 disk
└─xvda1      202:1    0   8G  0 part /
xvdf         202:80   0   1G  0 disk
[root@ip-172-31-30-239 ec2-user]# mkdir Sravani
[root@ip-172-31-30-239 ec2-user]# mount -t xfs /dev/xvdf Sravani
[root@ip-172-31-30-239 ec2-user]# cd Sravani/
[root@ip-172-31-30-239 Sravani]# ls
SharedFileEBS.txt
[root@ip-172-31-30-239 Sravani]#
```

### 3. Snapshot

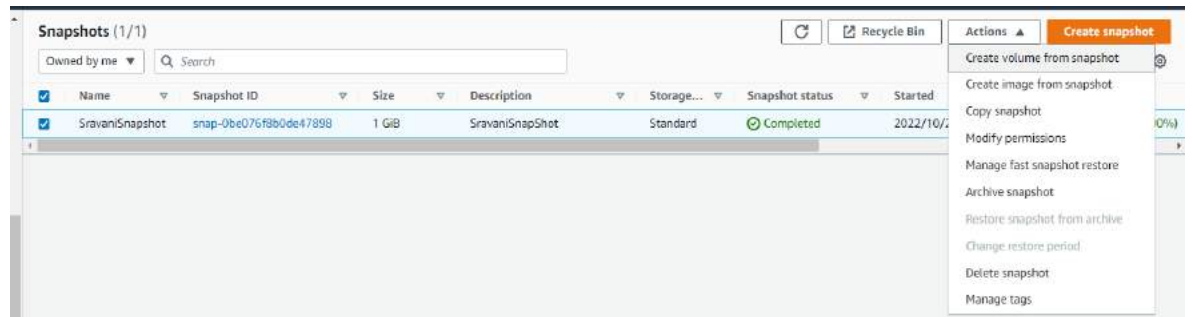
a. Create snapshot from volume:



b. Snapshots list:



c. Create volume from snapshot:



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

### Volume settings

Snapshot ID  
snap-0be076f8b0de47898 (SravaniSnapshot)

Volume type [Info](#)  
General Purpose SSD (gp2)

Size (GiB) [Info](#)  
1  
Min: 1 GiB, Max: 16384 GiB. The value must be an integer.

IOPS  
100 / 3000  
Baseline of 3 IOPS per GiB with a minimum of 100 IOPS, burstable to 3000 IOPS.

Throughput (MiB/s) [Info](#)  
Not applicable

Availability Zone [Info](#)  
us-east-1b

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


### Attach volume [Info](#)

Attach a volume to an instance to use it as you would a regular physical hard disk drive.

#### Basic details

Volume ID  
vol-09bd6a4daa4ba96fb (VolumeFromSnapshot)


Availability Zone  
us-east-1b

Instance [Info](#)  
i-0db92858ef0c4a60f 

Only instances in the same Availability Zone as the selected volume are displayed.

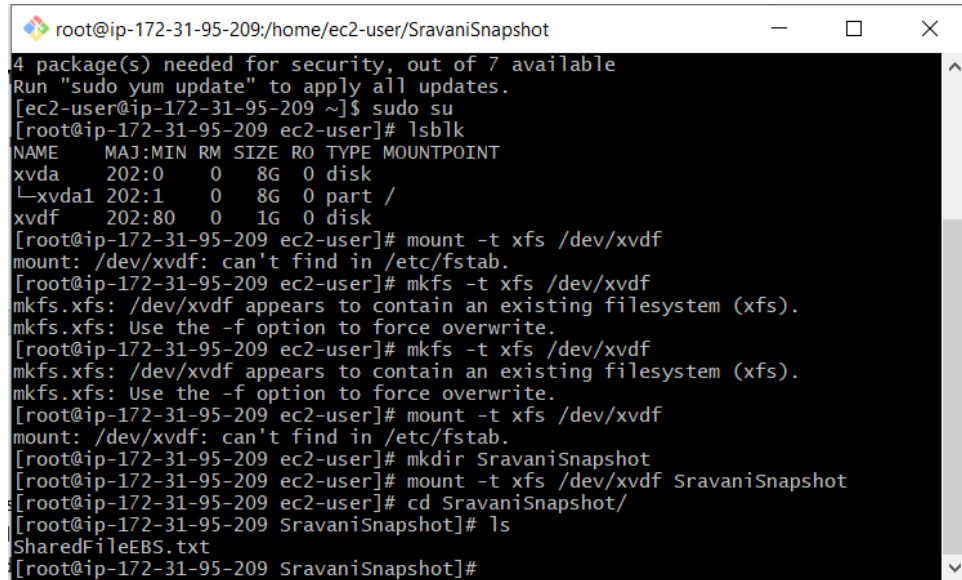
Device name [Info](#)  
/dev/sdf

Recommended device names for Linux: /dev/sda1 for root volume, /dev/sd[f-p] for data volumes.

 Newer Linux kernels may rename your devices to `/dev/xvdf` through `/dev/xvdp` internally, even when the device name entered here (and shown in the details) is `/dev/sdf` through `/dev/sdp`.

Cancel **Attach volume**

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



```
root@ip-172-31-95-209:/home/ec2-user/SravaniSnapshot
4 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-95-209 ~]$ sudo su
[root@ip-172-31-95-209 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-172-31-95-209 ec2-user]# mount -t xfs /dev/xvdf
mount: /dev/xvdf: can't find in /etc/fstab.
[root@ip-172-31-95-209 ec2-user]# mkfs -t xfs /dev/xvdf
mkfs.xfs: /dev/xvdf appears to contain an existing filesystem (xfs).
mkfs.xfs: Use the -f option to force overwrite.
[root@ip-172-31-95-209 ec2-user]# mkfs -t xfs /dev/xvdf
mkfs.xfs: /dev/xvdf appears to contain an existing filesystem (xfs).
mkfs.xfs: Use the -f option to force overwrite.
[root@ip-172-31-95-209 ec2-user]# mount -t xfs /dev/xvdf
mount: /dev/xvdf: can't find in /etc/fstab.
[root@ip-172-31-95-209 ec2-user]# mkdir SravaniSnapshot
[root@ip-172-31-95-209 ec2-user]# mount -t xfs /dev/xvdf SravaniSnapshot
[root@ip-172-31-95-209 ec2-user]# cd SravaniSnapshot/
[root@ip-172-31-95-209 SravaniSnapshot]# ls
SharedFileEBS.txt
[root@ip-172-31-95-209 SravaniSnapshot]#
```



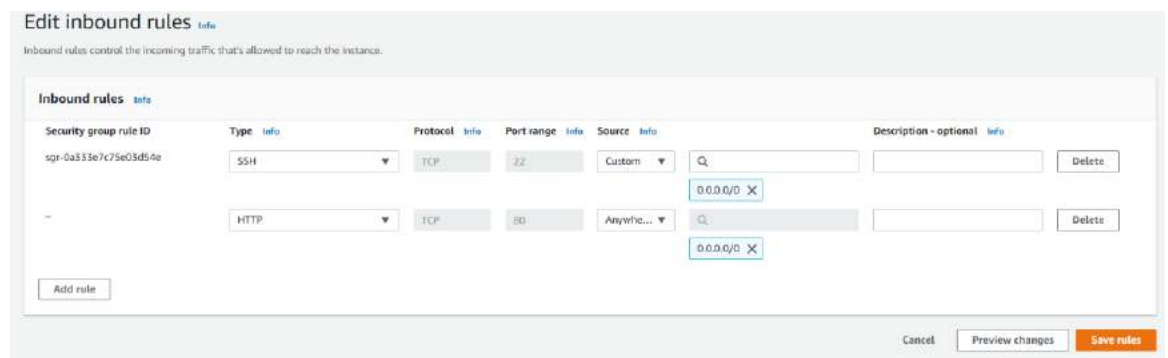
## 4. AMI (Amazon Machine Image)

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

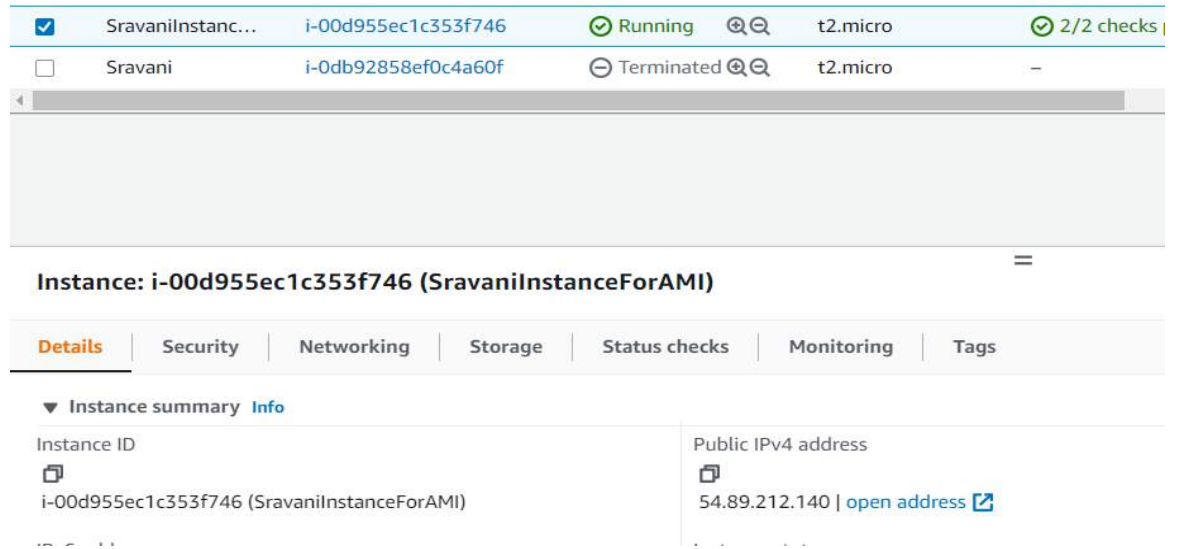
```
root@ip-172-31-89-234: /home/ubuntu
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-89-234:/home/ubuntu# sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese
   Active: active (running) since Thu 2022-10-20 05:57:44 UTC; 14s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 1876 (apache2)
     Tasks: 55 (limit: 1143)
    Memory: 5.0M
       CPU: 29ms
    CGroup: /system.slice/apache2.service
           └─1876 /usr/sbin/apache2 -k start
             └─1878 /usr/sbin/apache2 -k start
               └─1879 /usr/sbin/apache2 -k start

Oct 20 05:57:44 ip-172-31-89-234 systemd[1]: Starting The Apache HTTP Server...
Oct 20 05:57:44 ip-172-31-89-234 systemd[1]: Started The Apache HTTP Server.
lines 1-15/15 (END)
```

- b. Once apache is 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:



The screenshot shows the AWS Management Console interface. At the top, there is a table of EC2 instances:

Instance State	Instance Name	Instance ID	Status	Instance Type	Checks
<input checked="" type="checkbox"/>	SravanInstanceForAMI	i-00d955ec1c353f746	Running	t2.micro	2/2 checks passed
<input type="checkbox"/>	SravanInstance	i-0db92858ef0c4a60f	Terminated	t2.micro	-

Below the table, the details for the selected instance **Instance: i-00d955ec1c353f746 (SravanInstanceForAMI)** are shown. The **Details** tab is active, displaying the **Instance summary** section:

- Instance ID:** i-00d955ec1c353f746 (SravanInstanceForAMI)
- Public IPv4 address:** 54.89.212.140 | [open address](#)

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



The screenshot shows a web browser window with the address bar displaying <http://54.89.212.140>. The page content is the Apache2 Default Page on Ubuntu, featuring the Ubuntu logo and the text "Apache2 Default Page" and "It works!". Below this, there is a paragraph of text explaining that this is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems.

e. Install Php:

```

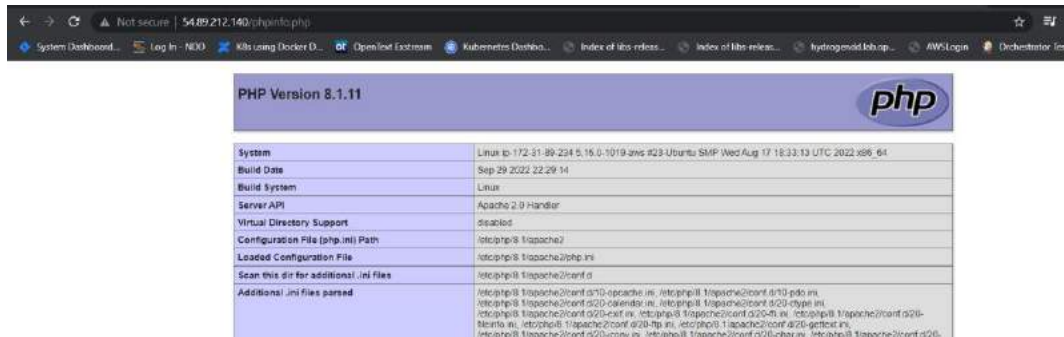
ubuntu@ip-172-31-89-234: ~
System information as of Thu Oct 20 06:35:08 UTC 2022

System load: 0.0          Processes:           107
Usage of /:  27.7% of 7.57GB  Users logged in:   0
Memory usage: 26%        IPv4 address for eth0: 172.31.89.234
Swap usage:  0%

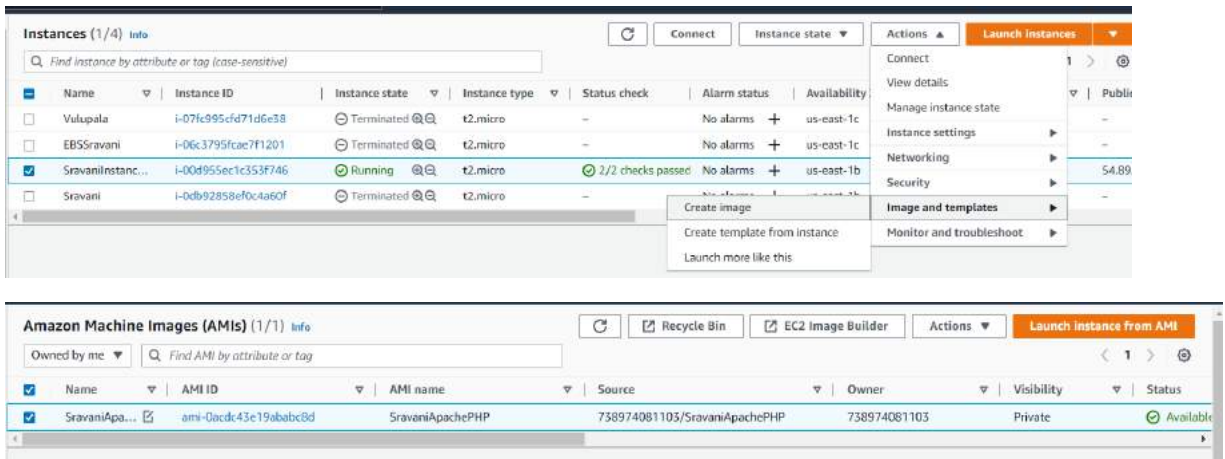
2 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

*** System restart required ***
Last login: Thu Oct 20 06:28:39 2022 from 43.241.120.40
ubuntu@ip-172-31-89-234:~$ php --version
PHP 8.1.11 (cli) (built: Sep 29 2022 22:29:14) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.11, Copyright (c) Zend Technologies
with Zend OPcache v8.1.11, Copyright (c), by Zend Technologies
ubuntu@ip-172-31-89-234:~$ sudo systemctl restart apache2
ubuntu@ip-172-31-89-234:~$ echo '<?php phpinfo(); ?>' | sudo tee -a /var/www/html/phpinfo.php > /dev/null
ubuntu@ip-172-31-89-234:~$
  
```

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:

EC2 > Instances > Launch an instance

### Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

**Name and tags Info**

Name:  [Add additional tags](#)

**Application and OS Images (Amazon Machine Image) Info**

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

[AMI from catalog](#) [Recents](#) [My AMIs](#) [Quick Start](#)

Amazon Machine Image (AMI)  
SrvaniApachePHP  
ami-0acdc43e19ababc8d

[Browse more AMIs](#)  
Including AMIs from AWS, Marketplace and

**Summary**

Number of instances Info:

Software image (AMI): SrvaniApachePHP  
ami-0acdc43e19ababc8d

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

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

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet.

[Cancel](#) [Launch instance](#)

- 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:

Name	Instance ID	Instance state	Instance type
SrvaniInstanceForAMI	i-00d955ec1c353f746	Running	t2.micro
<b>SrvaniInstanceFromAMI</b>	<b>i-0c8015f5ae4eb92fa</b>	<b>Running</b>	<b>t2.micro</b>

**Instance: i-0c8015f5ae4eb92fa (SrvaniInstanceFromAMI)**

[Details](#) | [Security](#) | [Networking](#) | [Storage](#) | [Status checks](#) | [Monitoring](#) | [Tags](#)

**Instance summary Info**

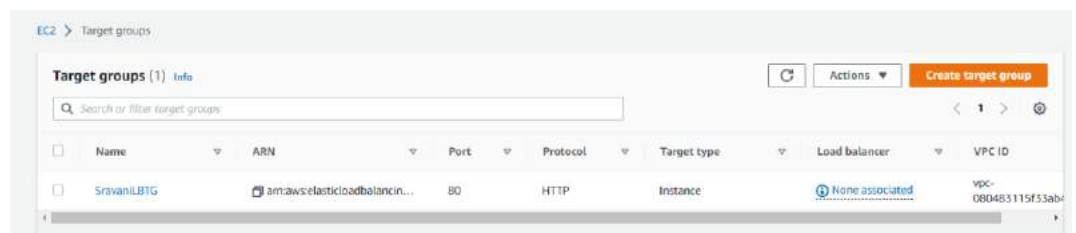
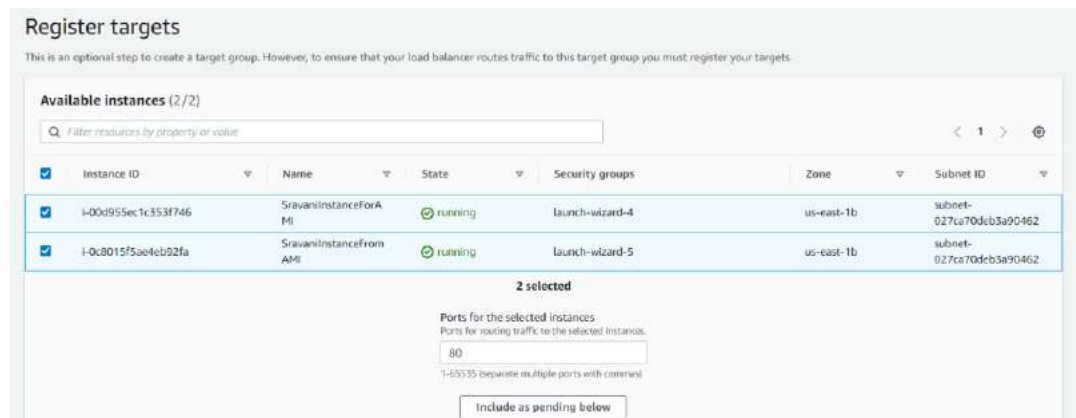
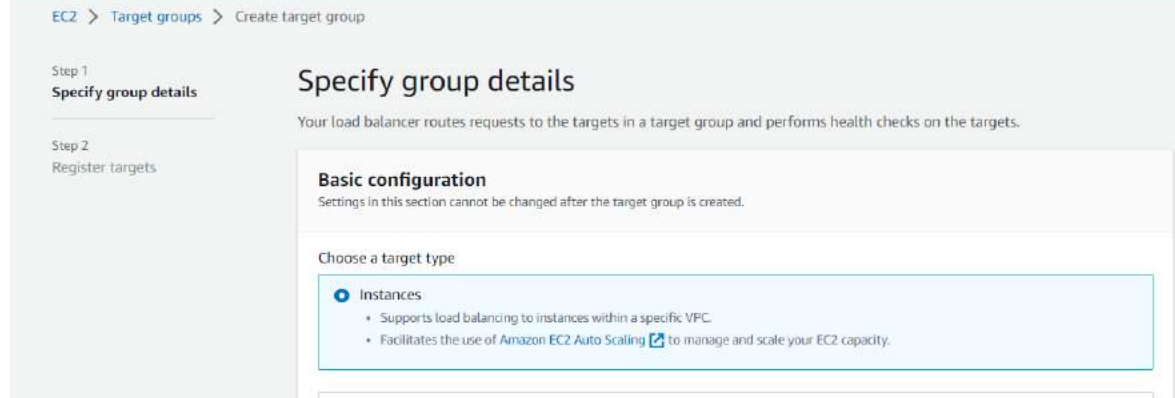
Instance ID: i-0c8015f5ae4eb92fa (SrvaniInstanceFromAMI)

Public IPv4 address: 54.204.191.45 | [open address](#)

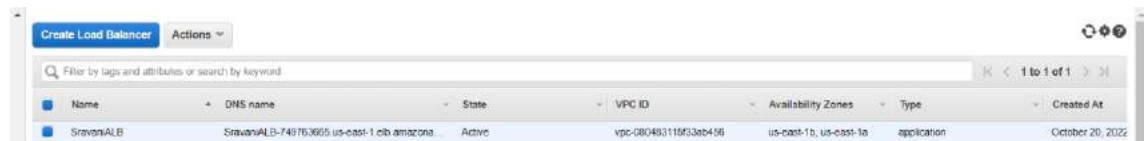


## 5. Load Balancer

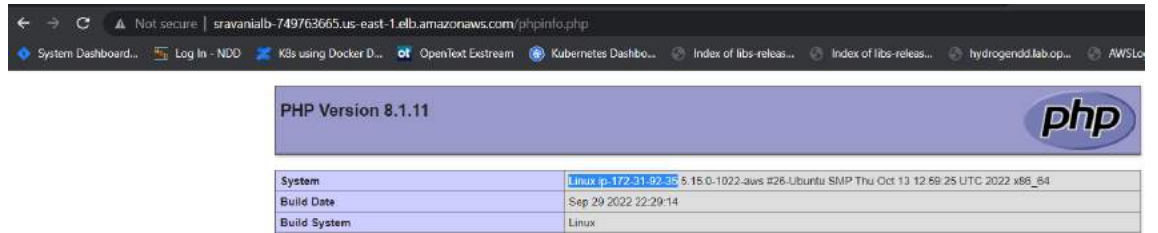
- Create a security group with inbound rules to allow port 80
- Create a target group and select the two instances that you have created and click on Include as pending and click on Create target group button at the bottom



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

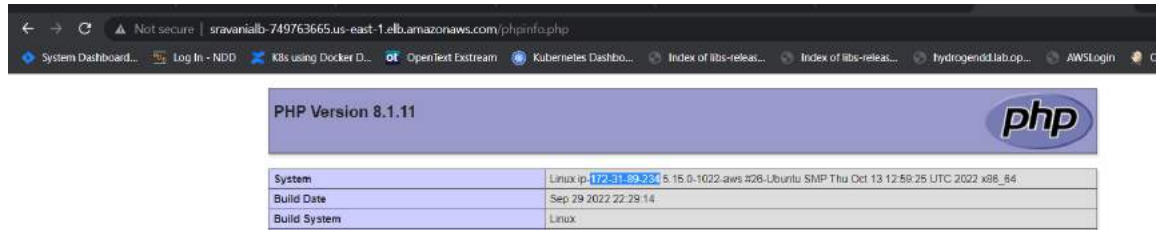


d. Private Ips of two machines : 172.31.89.234 and 172.31.92.35



A screenshot of a web browser displaying the PHPinfo.php page. The browser's address bar shows the URL "sraivialb-749763665.us-east-1.elb.amazonaws.com/phpinfo.php". The page header indicates "PHP Version 8.1.11" with the PHP logo. Below the header is a table with the following information:

System	Linux ip:172.31.92.35 5.15.0-1022-aws #26-Ubuntu SMP Thu Oct 13 12:59:25 UTC 2022 x86_64
Build Date	Sep 29 2022 22:29:14
Build System	Linux

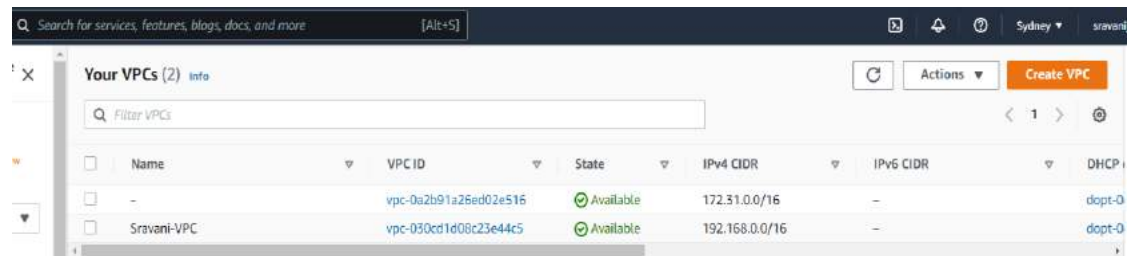


A screenshot of a web browser displaying the PHPinfo.php page. The browser's address bar shows the URL "sraivialb-749763665.us-east-1.elb.amazonaws.com/phpinfo.php". The page header indicates "PHP Version 8.1.11" with the PHP logo. Below the header is a table with the following information:

System	Linux ip:172.31.89.234 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

## 6. VPC

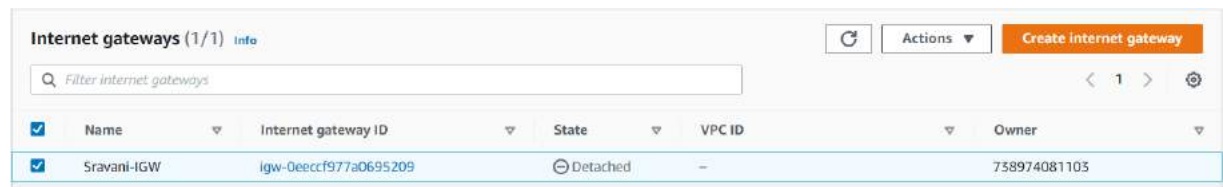
- a. Create a VPC:



The screenshot shows the AWS Management Console interface for 'Your VPCs (2)'. It includes a search bar, a refresh button, and an 'Actions' dropdown menu. A table lists the VPCs with columns for Name, VPC ID, State, IPv4 CIDR, IPv6 CIDR, and DHCP. Two VPCs are listed: one with ID vpc-0a2b91a25ed02e516 and another named 'Sravani-VPC' with ID vpc-030cd1d08c23e44c5.

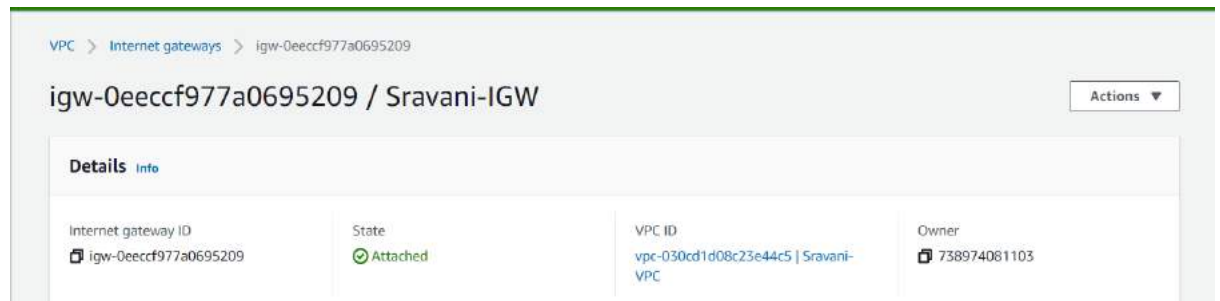
Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP
-	vpc-0a2b91a25ed02e516	Available	172.31.0.0/16	-	dopt-0
Sravani-VPC	vpc-030cd1d08c23e44c5	Available	192.168.0.0/16	-	dopt-0

- b. Create an internet gateway and attach the VPC:



The screenshot shows the AWS Management Console interface for 'Internet gateways (1/1)'. It includes a search bar, a refresh button, and an 'Actions' dropdown menu. A table lists the internet gateways with columns for Name, Internet gateway ID, State, VPC ID, and Owner. One internet gateway is listed: 'Sravani-IGW' with ID igw-0eecf977a0695209.

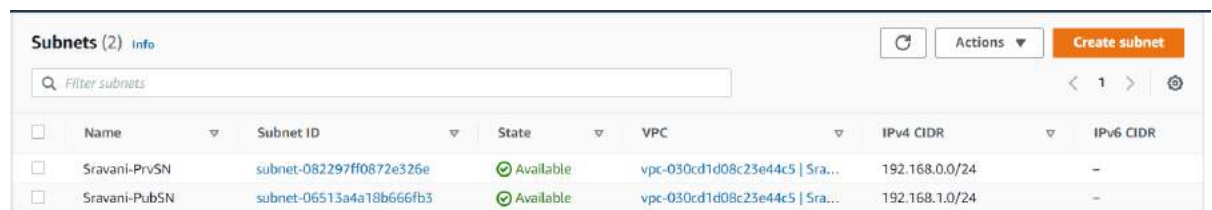
Name	Internet gateway ID	State	VPC ID	Owner
Sravani-IGW	igw-0eecf977a0695209	Detached	-	738974081103



The screenshot shows the details page for the internet gateway 'igw-0eecf977a0695209 / Sravani-IGW'. It includes a search bar, a refresh button, and an 'Actions' dropdown menu. The details section shows the following information:

Internet gateway ID	State	VPC ID	Owner
igw-0eecf977a0695209	Attached	vpc-030cd1d08c23e44c5   Sravani-VPC	738974081103

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

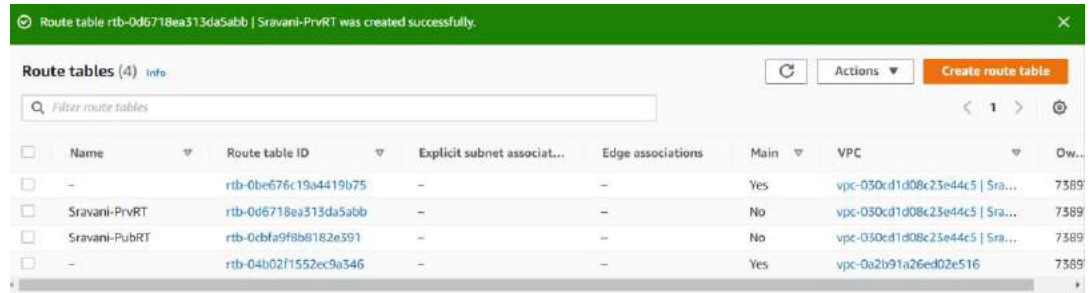


The screenshot shows the AWS Management Console interface for 'Subnets (2)'. It includes a search bar, a refresh button, and an 'Actions' dropdown menu. A table lists the subnets with columns for Name, Subnet ID, State, VPC, IPv4 CIDR, and IPv6 CIDR. Two subnets are listed: 'Sravani-PrivSN' with ID subnet-082297ff0872e326e and 'Sravani-PubSN' with ID subnet-06513a4a18b666fb3.

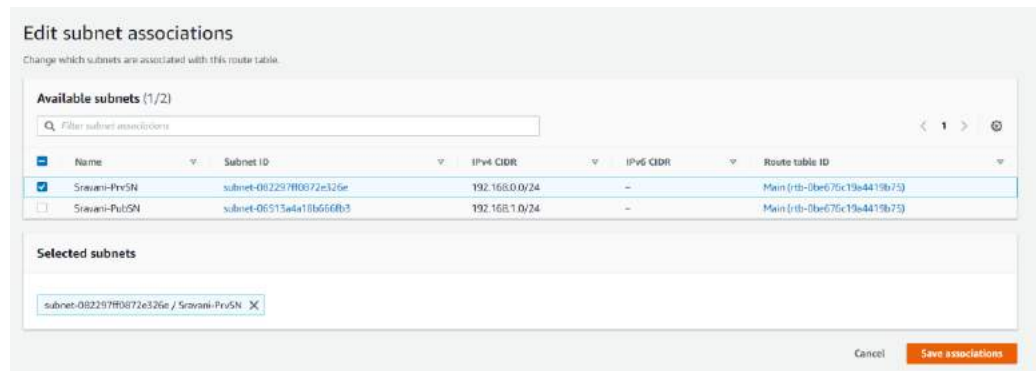
Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR
Sravani-PrivSN	subnet-082297ff0872e326e	Available	vpc-030cd1d08c23e44c5   Sra...	192.168.0.0/24	-
Sravani-PubSN	subnet-06513a4a18b666fb3	Available	vpc-030cd1d08c23e44c5   Sra...	192.168.1.0/24	-



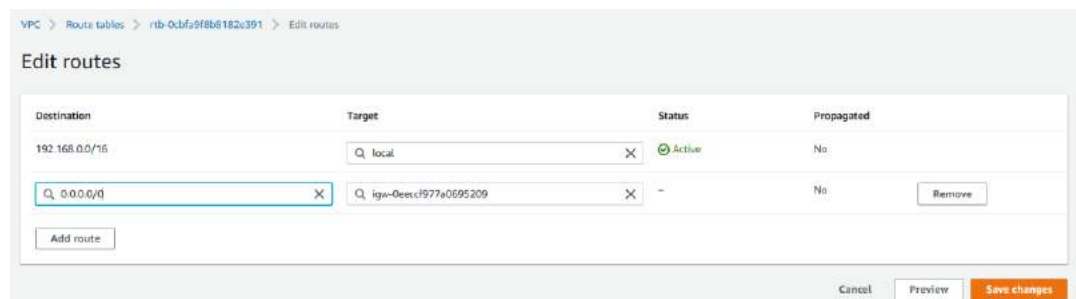
- d. Create route tables (Public and Private)



- 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)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Pub
Sravani-Priv-Instance	i-013bfeddc4fcfb514	Running	t2.micro	-	No alarms	ap-southeast-2b	-
Sravani-Pub-Instance	i-010e0a57ae9c61f8c	Running	t2.micro	Initializing	No alarms	ap-southeast-2b	-

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

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
-	SSH	TCP	22	Anywh...	
-	All ICMP - IPv4	ICMP	All	Anywh...	
-	All TCP	TCP	0 - 65535	Anywh...	

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

```

ec2-user@ip-192-168-1-252:~
$ ssh -i "Sydney.pem" ec2-user@3.26.232.71
The authenticity of host '3.26.232.71 (3.26.232.71)' can't be established.
ECDSA key fingerprint is SHA256:2slsht3p5PuxYefysIte/XT2iWGH/A3pc1ggHWA7llg.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.26.232.71' (ECDSA) to the list of known hosts.

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

https://aws.amazon.com/amazon-linux-2/
13 package(s) needed for security, out of 16 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-192-168-1-252 ~]$ ping google.com
PING google.com (142.250.66.206) 56(84) bytes of data.
64 bytes from syd09s23-in-f14.1e100.net (142.250.66.206): icmp_seq=1 ttl=110 tim
e=0.887 ms
64 bytes from syd09s23-in-f14.1e100.net (142.250.66.206): icmp_seq=2 ttl=110 tim
e=0.936 ms
64 bytes from syd09s23-in-f14.1e100.net (142.250.66.206): icmp_seq=3 ttl=110 tim
e=0.891 ms
64 bytes from syd09s23-in-f14.1e100.net (142.250.66.206): icmp_seq=4 ttl=110 tim
e=0.912 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3013ms
rtt min/avg/max/mdev = 0.887/0.906/0.936/0.035 ms
[ec2-user@ip-192-168-1-252 ~]$

```

- 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:

```
[ec2-user@ip-192-168-1-252 ~]$ chmod 400 Sydney.pem
[ec2-user@ip-192-168-1-252 ~]$ ls
Sydney.pem
[ec2-user@ip-192-168-1-252 ~]$ ssh -i Sydney.pem ec2-user@192.168.0.133

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

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-192-168-0-133 ~]$ ping google.com
PING google.com (142.250.66.238) 56(84) bytes of data.
```

- k. Create a NAT gateway and associate public subnet

Name	NAT gateway ID	Connectivity...	State	State message	Elastic IP address	Private IP ad
Sravani-NATGW	nat-02b0d0f6dce5b5db3	Public	Available	-	52.64.231.193	192.168.1.11

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

Destination	Target	Status	Propagated
192.168.0.0/16	local	Active	No
	nat-02b0d0f6dce5b5db3	-	No

m. Ping google.com from private subnet instance:

```
64 bytes from syd09s23-in-f14.1e100.net (142.250.66.206): icmp_seq=181 ttl=110 t
ime=1.02 ms
64 bytes from syd09s23-in-f14.1e100.net (142.250.66.206): icmp_seq=182 ttl=110 t
ime=0.978 ms
64 bytes from syd09s23-in-f14.1e100.net (142.250.66.206): icmp_seq=183 ttl=110 t
ime=0.952 ms
64 bytes from syd09s23-in-f14.1e100.net (142.250.66.206): icmp_seq=184 ttl=110 t
ime=0.952 ms
64 bytes from syd09s23-in-f14.1e100.net (142.250.66.206): icmp_seq=185 ttl=110 t
ime=0.976 ms
^C
--- google.com ping statistics ---
185 packets transmitted, 185 received, 0% packet loss, time 184673ms
rtt min/avg/max/mdev = 0.918/0.982/1.434/0.074 ms
[ec2-user@ip-192-168-0-133 ~]$
```