

First Name: **SaiGopi**

Last Name: **Pachipala**

## Contents

1. EC2 Instances .....	2
Steps:.....	2
2. EBS volume .....	3
Steps:.....	3
3. Snapshot .....	6
Steps:.....	6
4. AMI.....	8
Steps:.....	8
5. Load Balancer .....	12
Steps:.....	12
6. VPC with 2 public subnets & 2 private subnet having Internet gateway and NAT gateway .....	16
Steps:.....	16
7. VPC Peering.....	20
Steps:.....	20

# 1.EC2 Instances

## Steps:

1. Open AWS and under search box select EC2
2. Click on launch instance
3. Now enter name of your machine (Machine1-SaiGopi)
4. under AMI select Amazon linux
5. under instance type select t2.micro
6. under key pair >> create a new key pair (TOKYO.pem)
7. under firewall security group click on create security group
8. and click on launch instance
9. Repeat the same process and create one more instance (Machine1-Pachipala)

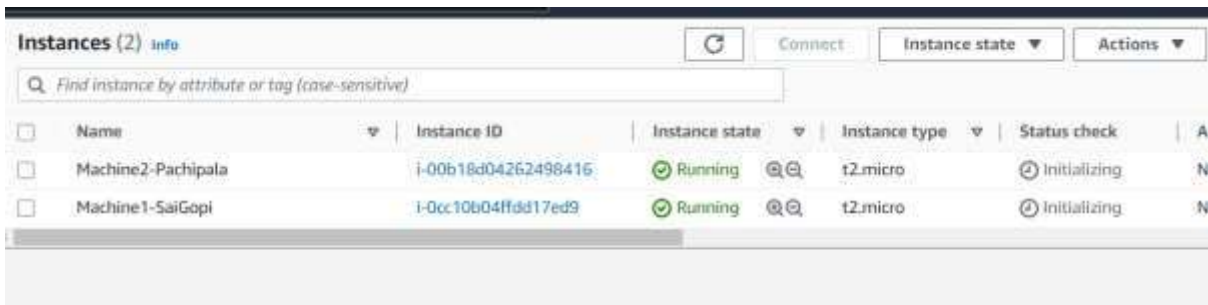


Fig1:Ec2 instances

## 10.Steps to connect to ec2 machines.

- Select machine1 under instances tab and click on connect.
- Under connect to instance select SSH client
- Now copy the ssh command shown under example
- Now go to the .pem file location directory and open command prompt terminal
- Now paste the ssh command and click on enter
- Type yes to connect and you can see your Machine1-Saigopi instance running



Fig2.Machine1-SaiGopi

```
Microsoft Windows [Version 10.0.22890.1000]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sai Gopi\Downloads>ssh -i "TOKYO.pem" ec2-user@ec2-18-183-154-27.ap-northeast-1.compute.amazonaws.com
The authenticity of host 'ec2-18-183-154-27.ap-northeast-1.compute.amazonaws.com (18.183.154.27)' can't be established.
ECDSA key fingerprint is SHA256:1T91JIEJlhfVZvc+0ZG6kPR8a/iPhzRlJ6GRfUbx0Po.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-183-154-27.ap-northeast-1.compute.amazonaws.com,18.183.154.27' (ECDSA) to the list of known hosts.

   _   _          _   _
  / \   \        / \   \
 (___)___)      (___)___)
  / \   \        / \   \
  \_/   /        \_/   /

Amazon Linux 2 AMI

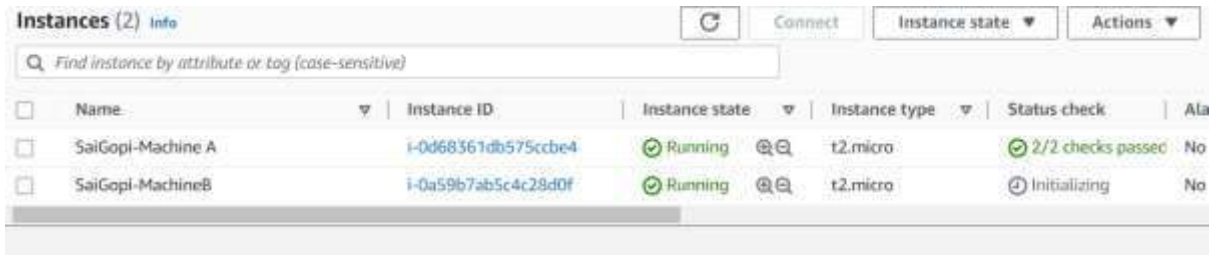
https://aws.amazon.com/amazon-linux-2/
13 package(s) needed for security, out of 16 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-11-166 ~]$ sudo su
[root@ip-172-31-11-166 ec2-user]#
```

Fig3.Machine2-Pachipala

## 2.EBS volume

### Steps:

1. When ever you want to provide an extra storage to your machine you opt for this Elastic Block store (EBS).
2. Created Two machines SaiGopi-Machine A and SaiGopi-Machine B in sydney region.



Name	Instance ID	Instance state	Instance type	Status check
SaiGopi-Machine A	i-0d68361db575ccbe4	Running	t2.micro	2/2 checks passed
SaiGopi-MachineB	i-0a59b7ab5c4c28d0f	Running	t2.micro	Initializing

Fig4: Instances for EBS

3. Under EBS select volumes and you can see default storage allocated for your EC2 machines.
4. Now click on create volume

- Under volume type select any type you want (General purpose SSD (gp2))
- Under size select the amount of GB (1GB)
- Under Availability zone you can select available zone in which your instance got created.
- Now click on create volume.
- Now click on volumes and you can see all volumes and newly created EBS.
- Now select the EBS and click on actions and click on attach volume.
- Under Basic details select your instance and click on attach volume.

### Volumes (3)

<input type="checkbox"/>	Name	Volume ID	Type	Size	IOPS
<input type="checkbox"/>	-	vol-0c6587eaa8c9a76a6	gp2	8 GiB	100
<input type="checkbox"/>	SaiGopi-EBS	vol-028550c673eb755d3	gp2	1 GiB	100
<input type="checkbox"/>	-	vol-02269764fa5037ee8	gp2	8 GiB	100

Fig5: EBS volume of 1GB

6. Now log on to SaiGopi-MachineA and make a file system and mount it.

- lsblk – to list all file systems
- mkdir <directory name> to create a storage directory
- mkfs -t xfs /dev/sdf
- mount -t xfs /dev/sdf storage
- created a storage directory named SaiGopi-Storage
- mounted it to file system and created ten .txt files in it
- umounted the file system.

```

Microsoft Windows [Version 10.0.22000.1098]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sai Gopi\Downloads>ssh -i "SaiGopi.pem" ec2-user@ec2-3-25-72-20.ap-southeast-2.compute.amazonaws.com
The authenticity of host 'ec2-3-25-72-20.ap-southeast-2.compute.amazonaws.com (3.25.72.20)' can't be established.
ECDSA key fingerprint is SHA256:rQ/xEYjYHLejzb4t2PQbByEeR9nbvuBtB+2jydZ/IdV.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-25-72-20.ap-southeast-2.compute.amazonaws.com,3.25.72.20' (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-172-31-7-91 ~]$ sudo su
[root@ip-172-31-7-91 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        202:00   0   1G  0 disk
[root@ip-172-31-7-91 ec2-user]# df -HT
Filesystem      Type      Size  Used Avail Use% Mounted on
devtmpfs        devtmpfs  474M   0  474M   0% /dev
tmpfs            tmpfs     483M   0  483M   0% /dev/shm
tmpfs            tmpfs     483M  412K  482M   1% /run
tmpfs            tmpfs     483M   0  483M   0% /sys/fs/cgroup
/dev/xvda1      xfs       8.0G  1.6G  6.5G  20% /
tmpfs            tmpfs     97M   0   97M   0% /run/user/1000
[root@ip-172-31-7-91 ec2-user]# mkdir SaiGopi-Storage
[root@ip-172-31-7-91 ec2-user]# ls
SaiGopi-Storage
[root@ip-172-31-7-91 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-172-31-7-91 ec2-user]# mount -t xfs /dev/xvdf /home/ec2-user/SaiGopi-Storage/

```

Fig 6: File system created for SaiGopi-MachineA and mounted it

```

[root@ip-172-31-7-91 ec2-user]# mount -t xfs /dev/xvdf /home/ec2-user/SaiGopi-Storage/
[root@ip-172-31-7-91 ec2-user]# df -hT
Filesystem      Type      Size  Used Avail Use% Mounted on
devtmpfs        devtmpfs  474M   0  474M   0% /dev
tmpfs           tmpfs     483M   0  483M   0% /dev/shm
tmpfs           tmpfs     483M  412K  482M   1% /run
tmpfs           tmpfs     483M   0  483M   0% /sys/fs/cgroup
/dev/xvda1      xfs       8.0G  1.6G  6.5G  20% /
tmpfs           tmpfs     97M   0   97M   0% /run/user/1000
/dev/xvdf       xfs      1014M   34M  981M   4% /home/ec2-user/SaiGopi-Storage
[root@ip-172-31-7-91 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-7-91 ec2-user]# ls
SaiGopi-Storage
[root@ip-172-31-7-91 ec2-user]# cd SaiGopi-Storage/
[root@ip-172-31-7-91 SaiGopi-Storage]# touch {1..10}.txt
[root@ip-172-31-7-91 SaiGopi-Storage]# ls
10.txt 1.txt 2.txt 3.txt 4.txt 5.txt 6.txt 7.txt 8.txt 9.txt
[root@ip-172-31-7-91 SaiGopi-Storage]# umount -t xfs /dev/xvdf /home/ec2-user/SaiGopi-Storage/
umount: /home/ec2-user/SaiGopi-Storage: target is busy.
umount: /home/ec2-user/SaiGopi-Storage/: target is busy.
[root@ip-172-31-7-91 SaiGopi-Storage]# cd ..
[root@ip-172-31-7-91 ec2-user]# umount -t xfs /dev/xvdf /home/ec2-user/SaiGopi-Storage/
umount: /home/ec2-user/SaiGopi-Storage/: not mounted.
[root@ip-172-31-7-91 ec2-user]# umount /home/ec2-user/SaiGopi-Storage/
umount: /home/ec2-user/SaiGopi-Storage/: not mounted.
[root@ip-172-31-7-91 ec2-user]# df -hT
Filesystem      Type      Size  Used Avail Use% Mounted on
devtmpfs        devtmpfs  474M   0  474M   0% /dev
tmpfs           tmpfs     483M   0  483M   0% /dev/shm
tmpfs           tmpfs     483M  412K  482M   1% /run
tmpfs           tmpfs     483M   0  483M   0% /sys/fs/cgroup
/dev/xvda1      xfs       8.0G  1.6G  6.5G  20% /
tmpfs           tmpfs     97M   0   97M   0% /run/user/1000
[root@ip-172-31-7-91 ec2-user]# cat

```

Fig7: Created 10 files in SaiGopi-Storage and unmounted it

7. Now detach the EBS volume from machine A and attach it to Machine B
8. Now connect to Machine B, create a new directory and mount the same to it.
9. SaiGopi-MachineB EBS volume contains all the ten txt files.

```

    _|_  _|_  )
    _| (  _| /  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-172-31-8-108 ~]$ sudo su
[root@ip-172-31-8-108 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-172-31-8-108 ec2-user]# df -hT
Filesystem      Type      Size  Used Avail Use% Mounted on
devtmpfs        devtmpfs  474M   0  474M   0% /dev
tmpfs           tmpfs     483M   0  483M   0% /dev/shm
tmpfs           tmpfs     483M 412K  482M   1% /run
tmpfs           tmpfs     483M   0  483M   0% /sys/fs/cgroup
/dev/xvda1      xfs       8.0G  1.6G  6.5G  20% /
tmpfs           tmpfs     97M   0   97M   0% /run/user/1000
[root@ip-172-31-8-108 ec2-user]# ls
[root@ip-172-31-8-108 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: 2330CCC2-270B-42AA-8CB6-AB640F80B1B4

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-172-31-8-108 ec2-user]# mkdir SaiGopi-Attach
[root@ip-172-31-8-108 ec2-user]# mount -t xfs /dev/xvdf /home/ec2-user/SaiGopi-Attach/
[root@ip-172-31-8-108 ec2-user]# cd SaiGopi-Attach/
[root@ip-172-31-8-108 SaiGopi-Attach]# ls
10.txt 1.txt 2.txt 3.txt 4.txt 5.txt 6.txt 7.txt 8.txt 9.txt
[root@ip-172-31-8-108 SaiGopi-Attach]# cat

```

Fig8:SaiGopi-MachineB EBS

### 3.Snapshot

#### Steps:

1. Under EC2 Elastic Block store click on Snapshot
2. Click on create snapshot
3. Under volume id select your volume (SaiGopi-EBS) in Sydney region
4. Under description enter name of snapshot
5. Now click on create snapshot

6. Now click on snapshots and you can able to see your created snapshot
7. Select your snapshot and click on actions and click on copy snapshot
8. In settings page of copy snapshot ,under Destination region select the region where you want to create Tokyo (ap-northeast1)
9. Now click on copy snapshot

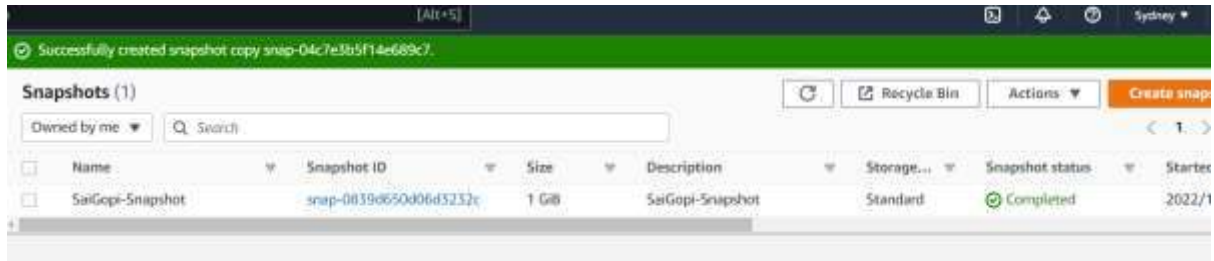


Fig 9: Snapshot created in Tokyo region from SaiGopi-EBS volume

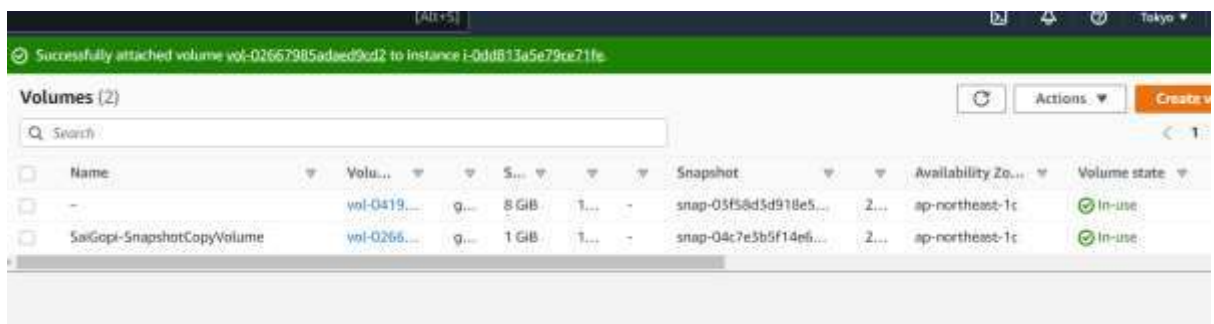


Fig 10: volume created from copy snapshot in Tokyo region

10. Create a SaiGopi-Machine C in Tokyo region and attach the EBS volume created from Snapshot copy
11. Now connect to SaiGopi-Machine C and create a new storage directory named SaiGopi-SnaphotVolume and mount it.
12. Switch to the SaiGopi-SnaphotVolume directory and check the list of files in it.

```

  _ | _ | _ )
  _ | ( _ | /
  _ | \ | _ |

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-172-31-9-249 ~]$ sudo su
[root@ip-172-31-9-249 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-172-31-9-249 ec2-user]# df -hT
Filesystem      Type      Size  Used Avail Use% Mounted on
devtmpfs        devtmpfs  474M   0  474M   0% /dev
tmpfs           tmpfs     483M   0  483M   0% /dev/shm
tmpfs           tmpfs     483M 412K  482M   1% /run
tmpfs           tmpfs     483M   0  483M   0% /sys/fs/cgroup
/dev/xvda1      xfs       8.0G  1.6G  6.5G  20% /
tmpfs           tmpfs     97M   0   97M   0% /run/user/1000
[root@ip-172-31-9-249 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: 2330CCC2-270B-42AA-8CB6-AB640F80B1B4

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-172-31-9-249 ec2-user]# mkdir SaiGopi-SnapshotVolume
[root@ip-172-31-9-249 ec2-user]# mount -t xfs /dev/xvdf /home/ec2-user/SaiGopi-SnapshotVolume/
[root@ip-172-31-9-249 ec2-user]# cd SaiGopi-SnapshotVolume/
[root@ip-172-31-9-249 SaiGopi-SnapshotVolume]# ls
10.txt 1.txt 2.txt 3.txt 4.txt 5.txt 6.txt 7.txt 8.txt 9.txt
[root@ip-172-31-9-249 SaiGopi-SnapshotVolume]#

```

Fig 11: SaiGopi-Machine C SnapshotVolume

## 4.AMI

An Amazon Machine Image (AMI) is a template that contains a software configuration (for example, an operating system, an application server, and applications). From an AMI, you launch an instance, which is a copy of the AMI running as a virtual server in the cloud.

### Steps:

- Created an SaiGopi-Machine1 Instance and in the security-groups add inbound rule http port 80 for this machine.
- Connect to the above instance and perform the below commands



Name	Instance ID	Instance state	Instance type	Status check	Alarm status
<input checked="" type="checkbox"/> SaiGopi-Machine1	i-03b981391c8f996bf	Running	t2.micro	2/2 checks passed	No alarms
<input type="checkbox"/> SaiGopi-Machine1	i-0b22d7ee72260d012	Terminated	t2.micro	-	No alarms

Instance: i-03b981391c8f996bf (SaiGopi-Machine1)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

▼ Instance summary info

Instance ID	Public IPv4 address	Private IPv4 addresses
i-03b981391c8f996bf (SaiGopi-Machine1)	54.238.72.24 open address	172.31.32.200

Fig12: SaiGopi-Machine1

- Preparing your Ubuntu server
  1. sudo apt update
  2. sudo ufw allow ssh
  3. sudo ufw allow 80
  4. sudo ufw allow 443
  5. sudo ufw enable

```

firewall is active and enabled on system startup
root@ip-172-31-32-200:/home/ubuntu# apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.3-0 mailcap mime-support ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser bzip2-doc
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.3-0 mailcap mime-support ssl-cert
0 upgraded, 13 newly installed, 0 to remove and 79 not upgraded.
Need to get 2138 kB of archives.
After this operation, 8501 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libapr1 amd64 1.7.0-0ubuntu1 [187 kB]
Get:2 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libaprutil1 amd64 1.6.1-5ubuntu4 [92.4 kB]
Get:3 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.1-5ubuntu4 [11.1 kB]
Get:4 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libaprutil1-ldap amd64 1.6.1-5ubuntu4 [9162 B]
Get:5 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 liblua5.3-0 amd64 5.3.6-1build1 [140 kB]
Get:6 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-bin amd64 2.4.52-1ubuntu4.1 [1367 kB]
Get:7 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-data all 2.4.52-1ubuntu4.1 [165 kB]
Get:8 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-utils amd64 2.4.52-1ubuntu4.1 [89.1 kB]
Get:9 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 mailcap all 3.70+emailubuntu1 [23.8 kB]

```

Fig13: Preparing your Ubuntu server

- Installing and testing Apache2
  1. sudo apt install apache2
  2. sudo systemctl status apache2
  3. <http://YOURSERVERIPADDRESS/>

```

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-32-200:/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 Sun 2022-11-13 17:35:41 UTC; 15s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2463 (apache2)
     Tasks: 55 (limit: 1143)
   Memory: 4.9M
     CPU: 20ms
   CGroup: /system.slice/apache2.service
           └─2463 /usr/sbin/apache2 -k start
             └─2465 /usr/sbin/apache2 -k start
               └─2466 /usr/sbin/apache2 -k start

Nov 13 17:35:41 ip-172-31-32-200 systemd[1]: Starting The Apache HTTP Server...
Nov 13 17:35:41 ip-172-31-32-200 systemd[1]: Started The Apache HTTP Server.
root@ip-172-31-32-200:/home/ubuntu#

```

Fig 14: Testing apache2

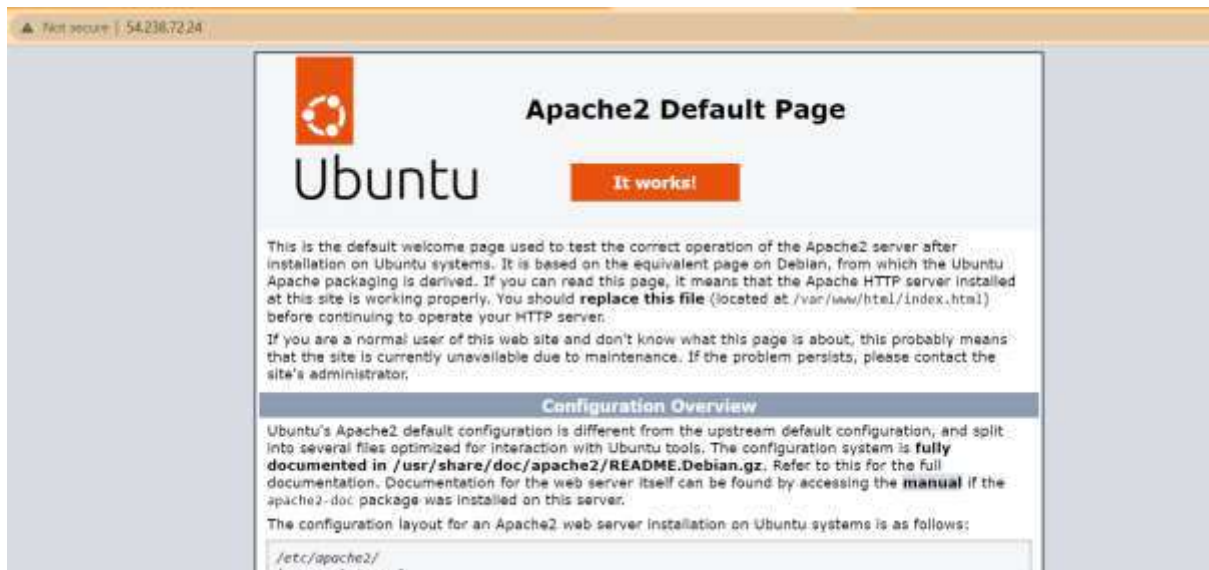


Fig 15: Testing apache2 on browser

- Installing and testing PHP
  1. `sudo apt install php8.1`
  2. `php --version`
  3. `sudo systemctl restart apache2`
  4. `echo '<?php phpinfo(); ?>' | sudo tee -a /var/www/html/phpinfo.php > /dev/null`
  5. <http://YOURSERVERIPADDRESS/phpinfo.php>



Fig 16: Testing PHP on browser

- Now create AMI from existing instance
- From this AMI create a new instance and in the security-group add inbound rule for http port 80
- Connect to this instance and try to access the ubuntu and php on browser

```

C:\Users\Sai Gopi\Downloads>ssh -i "TOKYO.pem" root@ec2-18-176-55-205.ap-northeast-1.compute.amazonaws.com
The authenticity of host 'ec2-18-176-55-205.ap-northeast-1.compute.amazonaws.com (18.176.55.205)' can't be established.
ED25519 key fingerprint is SHA256:oryzIHyjXsD6kCy2RyngH2uEapj5R/kq34hFYFz4bIM.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-176-55-205.ap-northeast-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Please login as the user "ubuntu" rather than the user "root".

ssh -i "TOKYO.pem" root@ec2-18-176-55-205.ap-northeast-1.compute.amazonaws.comConnection to ec2-18-176-55-205.ap-northeast-1.com

C:\Users\Sai Gopi\Downloads>ssh -i "TOKYO.pem" ubuntu@ec2-18-176-55-205.ap-northeast-1.compute.amazonaws.com
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1019-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sun Nov 13 18:57:58 UTC 2022

System load:  0.3046875      Processes:    112
Usage of /:   22.4% of 7.576G Users logged in:  0
Memory usage: 21%          IPv4 address for eth0: 172.31.32.196
Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

  https://ubuntu.com/aws/pro

84 updates can be applied immediately.
54 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Sun Nov 13 17:33:41 2022 from 175.101.107.225
ubuntu@ip-172-31-32-196:~$ sudo su
root@ip-172-31-32-196:/home/ubuntu# cat

```

Fig 17: SaiGopi-Machine2



Fig 18: Testing ubuntu for SaiGopi-Machine2

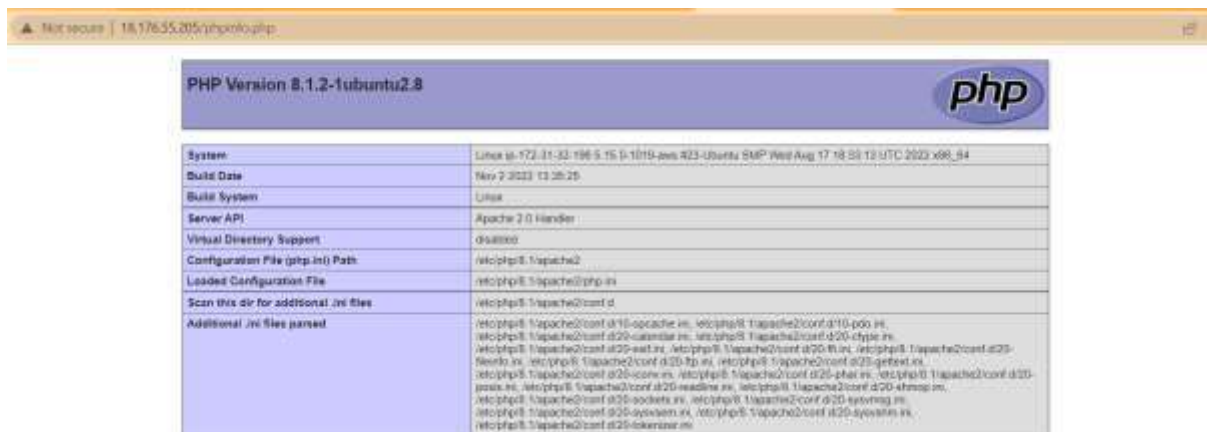


Fig 19: Testing ubuntu for SaiGopi-Machine2

## 5. Load Balancer

### Steps:

- Create a EC2 machine (SaiGopi-A) and add security group with inbound rule allowing SSH and HTTP port.
- Prepare your ubuntu server and install and test apache2
- Install and test PHP8.1
- Create an AMI and create two instances from AMI with security group allowing inbound rule for SSH and HTTP port.

The screenshot shows the AWS Management Console 'Instances' page. A table lists three instances: SaiGopi-A, SaiGopi-2, and SaiGopi-1. SaiGopi-A is selected. Below the table, the details for instance i-04b05bf28e24549cd (SaiGopi-A) are shown, including its public IPv4 address (34.222.88.140) and private IPv4 addresses (172.31.0.166).

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input checked="" type="checkbox"/> SaiGopi-A	i-04b05bf28e24549cd	Running	t2.micro	2/2 checks passed	No alarms	us-west-2c
<input type="checkbox"/> SaiGopi-2	i-09cb0533eaccd3a27	Running	t2.micro	2/2 checks passed	No alarms	us-west-2c
<input type="checkbox"/> SaiGopi-1	i-0c0fddef876af708e	Running	t2.micro	2/2 checks passed	No alarms	us-west-2c

**Instance: i-04b05bf28e24549cd (SaiGopi-A)**

Instance ID: i-04b05bf28e24549cd (SaiGopi-A)  
Public IPv4 address: 34.222.88.140 | open address  
Private IPv4 addresses: 172.31.0.166

Fig20: SaiGopi-A

The screenshot shows the AWS Management Console 'Instances' page. A table lists three instances: SaiGopi-A, SaiGopi-2, and SaiGopi-1. SaiGopi-1 is selected. Below the table, the details for instance i-0c0fddef876af708e (SaiGopi-1) are shown, including its public IPv4 address (18.237.143.53) and private IPv4 addresses (172.31.0.139).

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/> SaiGopi-A	i-04b05bf28e24549cd	Running	t2.micro	2/2 checks passed	No alarms	us-west-2c
<input type="checkbox"/> SaiGopi-2	i-09cb0533eaccd3a27	Running	t2.micro	2/2 checks passed	No alarms	us-west-2c
<input checked="" type="checkbox"/> SaiGopi-1	i-0c0fddef876af708e	Running	t2.micro	2/2 checks passed	No alarms	us-west-2c

**Instance: i-0c0fddef876af708e (SaiGopi-1)**

Instance ID: i-0c0fddef876af708e (SaiGopi-1)  
Public IPv4 address: 18.237.143.53 | open address  
Private IPv4 addresses: 172.31.0.139

Fig21: SaiGopi-1

Find instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
SaiGopi-A	i-04b05bf28e24549cd	Running	t2.micro	2/2 checks passed	No alarms	us-west-2c
<b>SaiGopi-2</b>	<b>i-09cb0533eaccd3a27</b>	Running	t2.micro	2/2 checks passed	No alarms	us-west-2c
SaiGopi-1	i-0c0fdddf876af708e	Running	t2.micro	2/2 checks passed	No alarms	us-west-2c

---

**Instance: i-09cb0533eaccd3a27 (SaiGopi-2)**

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

▼ Instance summary info

Instance ID: i-09cb0533eaccd3a27 (SaiGopi-2)

Public IPv4 address: 34.223.247.113 | open address

Private IPv4 addresses: 172.31.10.222

Fig22: SaiGopi-22

Amazon Machine Images (AMIs) (1) info

Owned by me | Find AMI by attribute or tag

Name	AMI ID	AMI name	Source
SaiGopi-Image	ami-0eca4011541e9e759	SaiGopi-Image	013620209347/SaiGopi-Image

Fig23: AMI

- Under Load balancing from EC2 service click on Load Balancer and click on create a load balancer.
- Click on create Application load balancer and Give name to your load balancer (SaiGopi-APLB) and select all mappings under Network Mapping.
- Under security groups create a new security group allowing inbound rules for SSH and HTTP port.
- Under Listeners and routing, need to create a new target group (SaiGopi-APLBTG) and include your target machines under it.
- Now connect your Target Group to your Load balancer and click on create.

Target groups (1/1) [info](#) Actions Create target group

Search or filter target groups

Name	ARN	Port	Protocol	Target type	Load balancer
SaiGopi-APLBTG	arn:aws:elasticloadbalancing:us-west-2:1166845986:targetgroup/SaiGopi-APLBTG/297fd192e6756e5b	80	HTTP	Instance	SaiGopi-APLB

target group: SaiGopi-APLBTG

Registered targets (2) Deregister Register targets

Filter resources by property or value

Instance ID	Name	Port	Zone	Health status	Health status details
i-09cb0533eaccd3a27	SaiGopi-2	80	us-west-2c	healthy	
i-0c0fddefb76af708e	SaiGopi-1	80	us-west-2c	healthy	

Fig24: Target Group

Filter by tags and attributes or search by keyword

Name	DNS name	State	VPC ID	Availability Zones	Type
SaiGopi-APLB	SaiGopi-APLB-1166845986.us-west-2.elb.amazonaws.com	Active	vpc-0c10b116b4625493	us-west-2c, us-west-2b	application

Load balancer: SaiGopi-APLB

Description Listeners Monitoring Integrated services Tags

Basic Configuration

Name	SaiGopi-APLB
ARN	arn:aws:elasticloadbalancing:us-west-2:1166845986:loadbalancer/app/SaiGopi-APLB/297fd192e6756e5b
DNS name	SaiGopi-APLB-1166845986.us-west-2.elb.amazonaws.com (A Record)
State	Active

Fig25: Load Balancer

- Now connect to your Load balancer by copying the DNS name and pasting in the browser.
- You can also check to which machine it is being connected using DNS name/phpinfo.php and you can check the ip address of your machine to which it is being connected.



Fig26: Connecting to Load balancer



Fig 27: Connecting to Machine1 using LB



Fig 28: Connecting to Machine2 using LB

## 6. VPC with 2 public subnets & 2 private subnet having Internet gateway and NAT gateway

### Steps:

- Create a VPC with 192.168.0.0/16 range.
- Create an Internet Gateway and attach it to the VPC created

Name	VPC ID	State	IPv4 CIDR
SaiGopi-VPC A	vpc-0d69682daf9b795a0	Available	192.168.0.0/16
-	vpc-0ebc6663e8377aeb3	Available	172.31.0.0/16

Fig 29: VPC

- Create Two private and two public subnets with the range:
  1. SaiGopi-VPCA- PRVSN1 >> 192.168.0.0/24
  2. SaiGopi-VPCA- PRVSN2 >> 192.168.1.0/24
  3. SaiGopi-VPCA- PUBSN1>> 192.168.2.0/24
  4. SaiGopi-VPCA- PUBSN2 >> 192.168.3.0/24

Name	Subnet ID	State	VPC	IPv4 CIDR
SaiGopi-VPCA-PUBSN1	subnet-043b29725e2b0186a	Available	vpc-0d69682daf9b795a0   Sai...	192.168.2.0/24
SaiGopi-VPCA-PUBSN2	subnet-0e9ee3e03e21580ed	Available	vpc-0d69682daf9b795a0   Sai...	192.168.3.0/24
-	subnet-0426a3f3a7ede3100	Available	vpc-0ebc6663e8377aeb3	172.31.16.0/20
SaiGopi-VPCA-PRVSN1	subnet-0329b24ec6e173b7b	Available	vpc-0d69682daf9b795a0   Sai...	192.168.0.0/24
-	subnet-00549a52040bbcd36	Available	vpc-0ebc6663e8377aeb3	172.31.0.0/20
SaiGopi-VPCA-PRVSN2	subnet-007b8487b82f840ce	Available	vpc-0d69682daf9b795a0   Sai...	192.168.1.0/24
-	subnet-06444667c2ef027e4	Available	vpc-0ebc6663e8377aeb3	172.31.32.0/20

Fig 30: Subnets

- Create Two route tables i.e one public and one private
  1. SaiGopi-VPCA-PUBRT
  2. SaiGopi-VPCA-PRVRT

Name	Route table ID	Explicit subnet associat...	Edge associations	Main
-	rtb-059c7bca3f0d09da6	-	-	Yes
SaiGopi-VPCA-PUBRT	rtb-0e92b9434c306719e	2 subnets	-	No
SaiGopi-VPCA-PRVRT	rtb-0c593935f03f6a2b6	2 subnets	-	No
-	rtb-0d1f9c94e43390491	-	-	Yes

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-043b29725e2b0186a / SaiGopi-VPCA-PUBSN1	192.168.2.0/24	-
subnet-0e9ee3e03e21580ed / SaiGopi-VPCA-PUBSN2	192.168.3.0/24	-

Fig 31: Public Route Table

- Associate the private route table to the existing private subnets
- Associate the public route table to the existing public subnets and the Internet Gateway



<input type="checkbox"/>	-	rtb-059c7bca3f0d09da6	-	-
<input type="checkbox"/>	SaiGopi-VPCA-PUBRT	rtb-0e92b9434c306719e	2 subnets	-
<input checked="" type="checkbox"/>	SaiGopi-VPCA-PRVRT	rtb-0c593935f03f6a2b6	2 subnets	-
<input type="checkbox"/>	-	rtb-0d1f9c94e43390491	-	-

Subnet ID	IPv4 CIDR
subnet-0329b24ec6e173b7b / SaiGopi-VPCA-PRVSN1	192.168.0.0/24
subnet-007b8487b82f840ce / SaiGopi-VPCA-PRVSN2	192.168.1.0/24

Fig 32: Private Route Table

- Create a New Security Group and add rules in the Inbound Traffic allowing SSH, All ICMP IPV4 and All TCP types .
- Now create Two EC2 instances with the newly created security group and under network edit option choose the Newly created VPC and the corresponding subnets
  1. SaiGopi-PRVSN1
  2. SaiGopi-PUBSN1

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
SaiGopi-PUBSN1	i-003f12cd7885add34	Running	t2.micro	2/2 checks passed	No alarms
SaiGopi-PRVSN1	i-0b569355d39e0184d	Terminated	t2.micro	-	No alarms
SaiGopi-PRVSN1	i-0bb3e64ea7a67e822	Running	t2.micro	2/2 checks passed	No alarms
SaiGopi-PUBSN1	i-02a69528bd26496c6	Terminated	t2.micro	-	No alarms

**Instance: i-0bb3e64ea7a67e822 (SaiGopi-PRVSN1)**

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary info

Instance ID: i-0bb3e64ea7a67e822 (SaiGopi-PRVSN1)

Public IPv4 address: 43.206.95.247 | open address

Private IPv4 address: 192.168.0.150

Instance state: Running

Public IPv4 DNS: -

Fig 33: EC2 machines

- Now connect to the EC2 Public machine from SSH client and ping from it
- We can able to ping from this machine
- Now connect to the EC2 Private machine from SSH client and try to ping
- We can see we are not able to connect to the machine and also we are not able to ping from it.

```

ECDSA key fingerprint is SHA256:qpuztVrAP7t8YS/bTzN1JN/Dkcoz/lm/4dUFAKEoDW8.
ECDSA key fingerprint is MD5:71:3f:e7:f5:0f:78:32:d0:ca:47:4a:9a:b1:54:86:db.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.150' (ECDSA) to the list of known hosts.

```

```

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

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-192-168-0-150 ~]$ sudo su
[root@ip-192-168-0-150 ec2-user]# ping google
ping: google: Name or service not known
[root@ip-192-168-0-150 ec2-user]# ping google.com
PING google.com (142.250.207.14) 56(84) bytes of data.
^C
--- google.com ping statistics ---
17 packets transmitted, 0 received, 100% packet loss, time 16360ms

[root@ip-192-168-0-150 ec2-user]# ssh -i "SaiGopi.pem" ec2-user@43.206.95.247
Warning: Identity file SaiGopi.pem not accessible: No such file or directory.
^C
[root@ip-192-168-0-150 ec2-user]# ping google.com
PING google.com (142.250.207.14) 56(84) bytes of data.
64 bytes from nrt13s54-in-f14.1e100.net (142.250.207.14): icmp_seq=1 ttl=102 time=4.78 ms
64 bytes from nrt13s54-in-f14.1e100.net (142.250.207.14): icmp_seq=2 ttl=102 time=4.11 ms
64 bytes from nrt13s54-in-f14.1e100.net (142.250.207.14): icmp_seq=3 ttl=102 time=4.05 ms
64 bytes from nrt13s54-in-f14.1e100.net (142.250.207.14): icmp_seq=4 ttl=102 time=4.03 ms
64 bytes from nrt13s54-in-f14.1e100.net (142.250.207.14): icmp_seq=5 ttl=102 time=4.02 ms
64 bytes from nrt13s54-in-f14.1e100.net (142.250.207.14): icmp_seq=6 ttl=102 time=4.04 ms
64 bytes from nrt13s54-in-f14.1e100.net (142.250.207.14): icmp_seq=7 ttl=102 time=4.01 ms
64 bytes from nrt13s54-in-f14.1e100.net (142.250.207.14): icmp_seq=8 ttl=102 time=4.03 ms
^C
--- google.com ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7008ms
rtt min/avg/max/mdev = 4.019/4.137/4.780/0.246 ms
[root@ip-192-168-0-150 ec2-user]# cat

```

Fig 34: Connecting to Public machine

- So We try to ping the Private machine from public machine , which will give an error of key not exists.
- So we copy the pem file into the Public machine using secure copy  
Scp -i .\SaiGopi.pem -r .\SaiGopi.pem ec2-user@public machine private ip address :/present working directory of Public machine.
- Grant the permission to the pem file chmod 700
- Now you can able to connect the private machine from the public machine.
- Now you are able to connect to the private machine but still you are not able to ping since its not connected to any internet Gateway.
- So for this purpose we use Nat Gateway in order to provide internet for the private machines.
- Create a Nat Gateway and allocate an elastic Ip .
- Now go to the Private route table and under routes add the Nat Gateway and now connect to this machine and ping from it

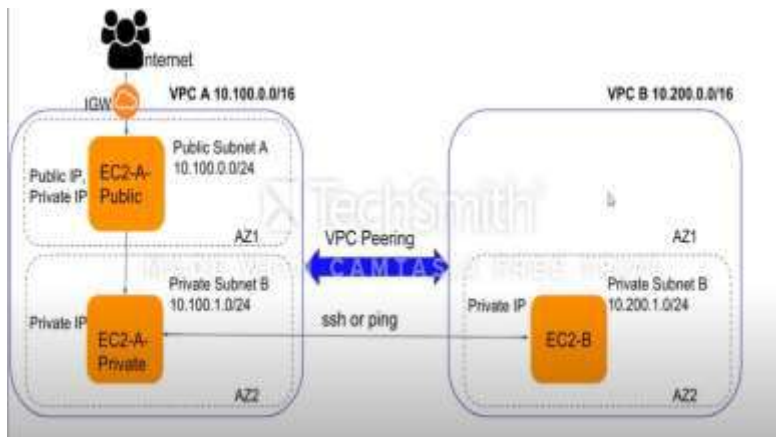
```
C:\Users\Sai Gopi\Downloads>ssh -i "SaiGopi.pem" ec2-user@35.77.9.139
The authenticity of host '35.77.9.139 (35.77.9.139)' can't be established.
ED25519 key fingerprint is SHA256:Z6JctaKbnoMbU8zSUS0i4sqSpjQTde1Dq7Jg1LL1fes.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '35.77.9.139' (ED25519) to the list of known hosts.
```

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

```
https://aws.amazon.com/amazon-linux-2/
1 package(s) needed for security, out of 1 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-192-168-2-140 ~]$ sudo su
[root@ip-192-168-2-140 ec2-user]# ping 192.168.0.150
PING 192.168.0.150 (192.168.0.150) 56(84) bytes of data.
64 bytes from 192.168.0.150: icmp_seq=1 ttl=255 time=0.983 ms
64 bytes from 192.168.0.150: icmp_seq=2 ttl=255 time=0.511 ms
64 bytes from 192.168.0.150: icmp_seq=3 ttl=255 time=0.485 ms
64 bytes from 192.168.0.150: icmp_seq=4 ttl=255 time=0.567 ms
64 bytes from 192.168.0.150: icmp_seq=5 ttl=255 time=0.557 ms
64 bytes from 192.168.0.150: icmp_seq=6 ttl=255 time=0.561 ms
64 bytes from 192.168.0.150: icmp_seq=7 ttl=255 time=0.499 ms
64 bytes from 192.168.0.150: icmp_seq=8 ttl=255 time=0.537 ms
64 bytes from 192.168.0.150: icmp_seq=9 ttl=255 time=0.603 ms
64 bytes from 192.168.0.150: icmp_seq=10 ttl=255 time=0.506 ms
64 bytes from 192.168.0.150: icmp_seq=11 ttl=255 time=0.422 ms
64 bytes from 192.168.0.150: icmp_seq=12 ttl=255 time=0.532 ms
64 bytes from 192.168.0.150: icmp_seq=13 ttl=255 time=0.528 ms
64 bytes from 192.168.0.150: icmp_seq=14 ttl=255 time=0.525 ms
64 bytes from 192.168.0.150: icmp_seq=15 ttl=255 time=0.650 ms
64 bytes from 192.168.0.150: icmp_seq=16 ttl=255 time=0.505 ms
64 bytes from 192.168.0.150: icmp_seq=17 ttl=255 time=0.524 ms
^C
--- 192.168.0.150 ping statistics ---
17 packets transmitted, 17 received, 0% packet loss, time 16348ms
rtt min/avg/max/mdev = 0.422/0.558/0.983/0.118 ms
```

Fig 35: Connecting to Private machine

## 7. VPC Peering



### Steps:

1. Creation of two VPC's
  - VPC A >> 10.100.0.0/16
  - VPC B >> 10.200.0.0/16

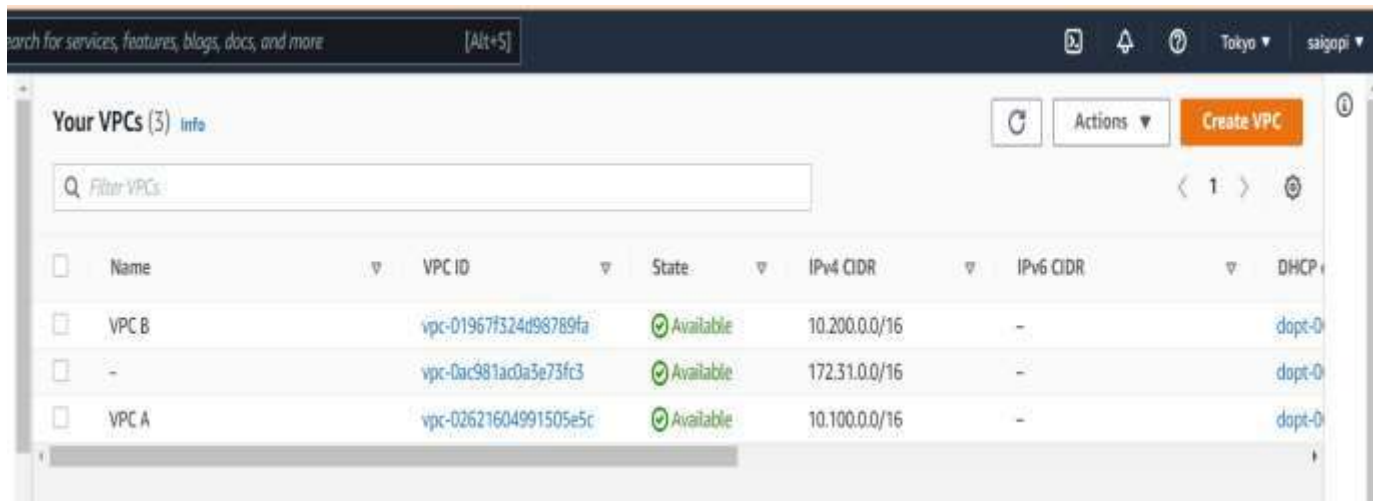


Fig 36: VPCs

2. Creation of Route Tables
  - VPC A-PUB RT
  - VPC A-PRV RT
  - VPC B PRV RT

Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC	Over
-	rtb-0edfa5f763882d09f	-	-	Yes	vpc-01967f324d98789fa   VPC B	01
VPC A PUB RT	rtb-05386b6632bb3ea94	subnet-0d23e2ebfb7b8...	-	No	vpc-02621604991505e5c   VP...	01
VPC B PRV RT	rtb-Dc2309b385a6b1a34	subnet-025f688bd2b8c...	-	No	vpc-01967f324d98789fa   VPC B	01
-	rtb-0693d6f4583bd52e3	-	-	Yes	vpc-0ac981ac0a3e73fc3	01
VPC A PRV RT	rtb-01d7c305c2e6064f8	subnet-0295007b1f1b5...	-	No	vpc-02621604991505e5c   VP...	01
-	rtb-07fb64939e0d75998	-	-	Yes	vpc-02621604991505e5c   VP...	01

Fig 37: Route tables

### 3. Creation of Subnets

- VPC A-PUBSN A >>10.100.0.0/24
- VPC A-PRVSN B >>10.100.1.0/24
- VPC B PRV B >>10.200.1.0/24

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR
-	subnet-01a99032e38138ede	Available	vpc-0ac981ac0a3e73fc3	172.31.16.0/20	-
VPC A-PUBSN A	subnet-0d23e2ebfb7b8bbcc	Available	vpc-02621604991505e5c   VP...	10.100.0.0/24	-
VPC A -PRVSN B	subnet-0295007b1f1b59cb4	Available	vpc-02621604991505e5c   VP...	10.100.1.0/24	-
-	subnet-00e93c3650aae642c	Available	vpc-0ac981ac0a3e73fc3	172.31.0.0/20	-
-	subnet-0f3d187ba70710d24	Available	vpc-0ac981ac0a3e73fc3	172.31.32.0/20	-
VPC B PRV B	subnet-025f688bd2b8c1aec	Available	vpc-01967f324d98789fa   VPC B	10.200.1.0/24	-

Fig 38: Subnets

### 4. Creation of Internet Gateway

- VPC A IGW

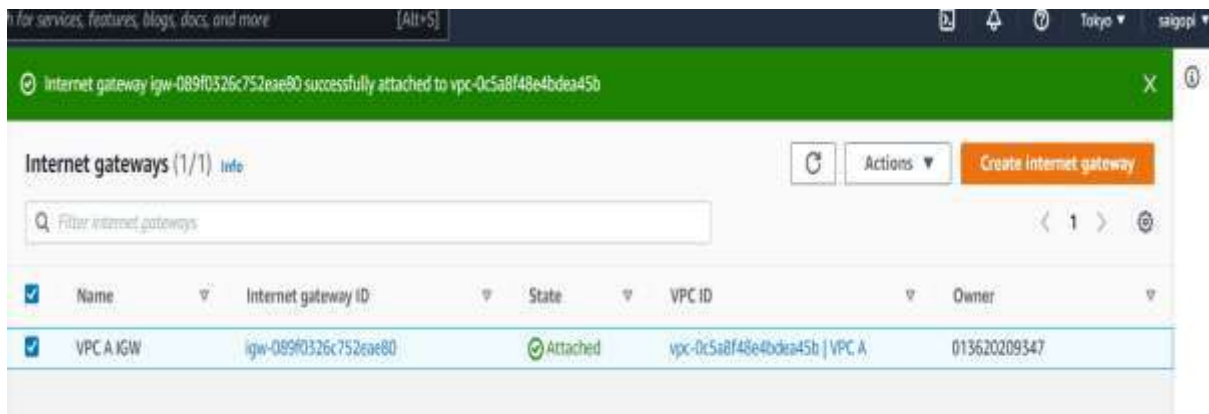


Fig 39: Internet Gateway

5. Attaching the subnets to corresponding route tables.
  - VPC A-PUBSN A to VPC A-PUB RT
  - VPC A-PRVSN B to VPC A-PRV RT
  - VPC B PRV B to VPC B PRV RT
6. Attaching Internet gateway to VPC
  - VPC A IGW to VPC A
7. Attaching route table to internet gateway
  - VPC A PUB RT to VPC A IGW
8. Creation of EC2 instances
  - OPEN EC2 instances and click on launch instance
  - Give EC2 instance a name and create a new key pair and select t2.micro instance type.
  - Under Network settings select the related VPC ,subnet and enable the public IP
  - Create a new security group and later edit the inbound and outbound rules to enable all IPV4 addresses.
  - And click on create instance.
  - EC2 VPCA-PUB
  - EC2 VPCA-PRV
  - EC2 VPCB-PRV

Find instance by attribute or tag (case-sensitive)								
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Avail		
EC2-VPCA-PUB	i-00d07574c709e4ef0	Terminated	t2.micro	-	No alarms	ap-nc		
EC2-VPCA-PRV	i-0c7b91d5960903217	Running	t2.micro	2/2 checks passed	No alarms	ap-nc		
EC2-VPCA-PUB	i-0019ff7968d023900	Running	t2.micro	2/2 checks passed	No alarms	ap-nc		
EC2-VPC B PRV	i-03b3e5267aa132403	Running	t2.micro	-	No alarms	ap-nc		

Fig 40: EC2 machines

9. Connecting to first EC2 machine >> EC2 VPCA-PUB

```

C:\ ec2-user@ip-10-100-0-154:~
Microsoft Windows [Version 10.0.22000.1098]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sai Gopi\Downloads>ssh -i "TOKYO.pem" ec2-user@3.113.5.151
The authenticity of host '3.113.5.151 (3.113.5.151)' can't be established.
ECDSA key fingerprint is SHA256:2JIm1u6LQbhvTDhfQ644Ih5E/NWQU/CJ5ewRU/WfYgY.
Are you sure you want to continue connecting (yes/no/[fingerprint])? YES
Warning: Permanently added '3.113.5.151' (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-10-100-0-154 ~]$

```

Fig 41: connecting to Public machine

10. Now trying to ping the EC2 VPCA-PRV machine from EC2 VPCA-PUB machine by using private IP of EC2 VPCA-PRV machine

```

C:\ root@ip-10-100-0-154:/home/ec2-user
[root@ip-10-100-0-154 ec2-user]# ping 10.100.1.226
PING 10.100.1.226 (10.100.1.226) 56(84) bytes of data:
64 bytes from 10.100.1.226: icmp_seq=1 ttl=255 time=0.425 ms
64 bytes from 10.100.1.226: icmp_seq=2 ttl=255 time=2.65 ms
64 bytes from 10.100.1.226: icmp_seq=3 ttl=255 time=0.544 ms
64 bytes from 10.100.1.226: icmp_seq=4 ttl=255 time=0.416 ms
64 bytes from 10.100.1.226: icmp_seq=5 ttl=255 time=0.382 ms
64 bytes from 10.100.1.226: icmp_seq=6 ttl=255 time=0.446 ms
64 bytes from 10.100.1.226: icmp_seq=7 ttl=255 time=0.449 ms
64 bytes from 10.100.1.226: icmp_seq=8 ttl=255 time=0.471 ms
^C
--- 10.100.1.226 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7128ms
rtt min/avg/max/mdev = 0.382/0.723/2.656/0.732 ms
[root@ip-10-100-0-154 ec2-user]#

```

Fig 42: connecting to private machine using VPC peering

11. I am unable to ssh the machine as it requires the .pem file .so copying the .pem file to EC2 VPCA-PUB 10-100-0-151 ec2user.

- scp -i .\TOKYO.pem -r .\TOKYO.pem ec2-user@publicmachine ip address :/Present working directoty.
- Connect to the machine and ping from it.

```

64 bytes from 10.100.1.226: icmp_seq=2 ttl=255 time=0.487 ms
64 bytes from 10.100.1.226: icmp_seq=3 ttl=255 time=0.589 ms
64 bytes from 10.100.1.226: icmp_seq=4 ttl=255 time=0.532 ms
^C
--- 10.100.1.226 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3049ms
rtt min/avg/max/mdev = 0.392/0.500/0.589/0.072 ms
[root@ip-10-100-0-154 ec2-user]# ssh -i "TOKYO.pem" ec2-user@54.178.163.32
^C
[root@ip-10-100-0-154 ec2-user]# ls -l
total 4
-rw-rw-r-- 1 ec2-user ec2-user 1674 Oct 27 18:26 TOKYO.pem
[root@ip-10-100-0-154 ec2-user]# chmod 700 TOKYO.pem
[root@ip-10-100-0-154 ec2-user]# ls-l
bash: ls-l: command not found
[root@ip-10-100-0-154 ec2-user]# ls -l
total 4
-rwx----- 1 ec2-user ec2-user 1674 Oct 27 18:26 TOKYO.pem
[root@ip-10-100-0-154 ec2-user]# ^Ch -i "TOKYO.pem" ec2-user@54.178.163.32
[root@ip-10-100-0-154 ec2-user]# ssh -i TOKYO.pem ec2-user@10.100.1.226

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

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-100-1-226 ~]$

```

Filter peering connections

Name	Peering connection ID	Status	Requester VPC	Accepter VPC
Peering VPC	pcx-0a10168be305db0ae	Active	vpc-02621604991505e5c / VP...	vpc-01967f324d98789fa / VP...

Fig 43: VPC Peering

12. Now connecting to the private EC2 machine from VPC peering



```

64 bytes from 10.100.1.226: icmp_seq=2 ttl=255 time=0.487 ms
64 bytes from 10.100.1.226: icmp_seq=3 ttl=255 time=0.589 ms
64 bytes from 10.100.1.226: icmp_seq=4 ttl=255 time=0.532 ms
^C
--- 10.100.1.226 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3049ms
rtt min/avg/max/mdev = 0.392/0.500/0.589/0.072 ms
[root@ip-10-100-0-154 ec2-user]# ssh -i "TOKYO.pem" ec2-user@54.178.163.32
^C
[root@ip-10-100-0-154 ec2-user]# ls -l
total 4
-rw-rw-r-- 1 ec2-user ec2-user 1674 Oct 27 18:26 TOKYO.pem
[root@ip-10-100-0-154 ec2-user]# chmod 700 TOKYO.pem
[root@ip-10-100-0-154 ec2-user]# ls -l
bash: ls-l: command not found
[root@ip-10-100-0-154 ec2-user]# ls -l
total 4
-rwx----- 1 ec2-user ec2-user 1674 Oct 27 18:26 TOKYO.pem
[root@ip-10-100-0-154 ec2-user]# ^Ch -i "TOKYO.pem" ec2-user@54.178.163.32
[root@ip-10-100-0-154 ec2-user]# ssh -i TOKYO.pem ec2-user@10.100.1.226

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

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-100-1-226 ~]$ sudo su
[root@ip-10-100-1-226 ec2-user]# ping google
ping: google: Name or service not known
[root@ip-10-100-1-226 ec2-user]# ping 10.200.1.12
PING 10.200.1.12 (10.200.1.12) 56(84) bytes of data.
64 bytes from 10.200.1.12: icmp_seq=1 ttl=255 time=2.23 ms
64 bytes from 10.200.1.12: icmp_seq=2 ttl=255 time=1.88 ms
64 bytes from 10.200.1.12: icmp_seq=3 ttl=255 time=1.88 ms
64 bytes from 10.200.1.12: icmp_seq=4 ttl=255 time=1.96 ms
64 bytes from 10.200.1.12: icmp_seq=5 ttl=255 time=1.95 ms
64 bytes from 10.200.1.12: icmp_seq=6 ttl=255 time=1.99 ms
^C
--- 10.200.1.12 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5006ms
rtt min/avg/max/mdev = 1.880/1.985/2.234/0.118 ms
[root@ip-10-100-1-226 ec2-user]#

```

Fig 44: EC2 machine by VPC peering