DEVOPS TOOLING WEBSITE SOLUTION

A DevOps team utilizes various tooling solutions in order to help the team carry out their day-to day activities in managing, developing, testing, deploying and monitoring different projects .

In this project we would be implementing a DevOps solution that consist of the following components

Pre-requisite for the projects is the following.

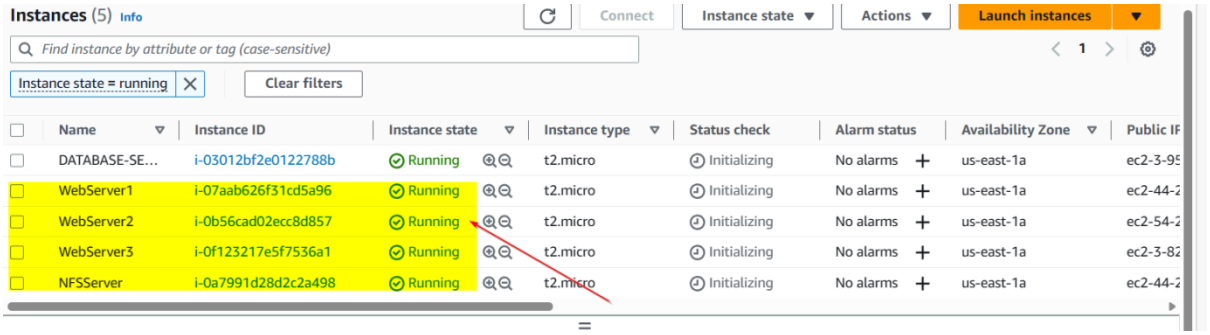
1) Fundamental Knowledge of Installing and downloading software
2) Basic Understanding of Linux Commands
3) AWS account login with EC2 instances
4) Webserver Linux: Red Hat Enterprise Linux 9
5) Database Server: On Ubuntu 22.04+ MySQL
6) Storage Server: Red Hat Enterprise Linux 9 +NFS Server
7) Programming Language: PHP
8) Code Repository
9) Internet connection

IMPLEMENTATION STEPS: Set up of all EC-2 instances.

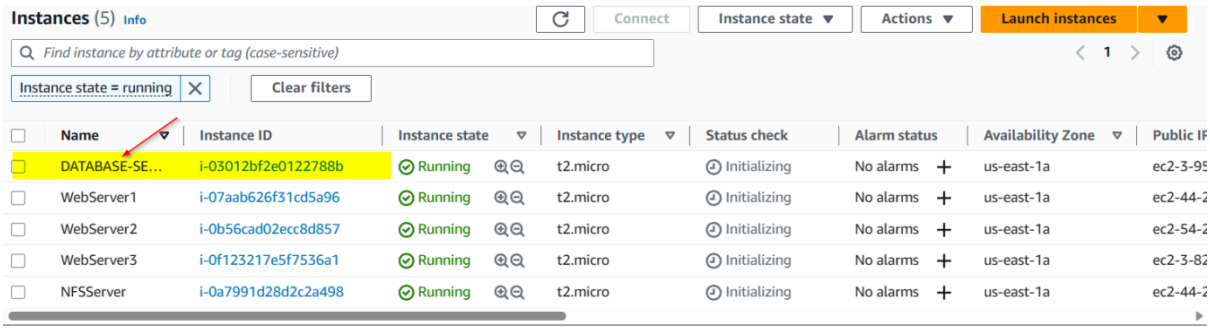
i) Ensure you login with your details to your AWS console via the https://aws.amazon.com
ii) Click on the EC2 link and spin up 5 EC2 instances and make sure they are set up with the operating systems below

4 Red Hat Enterprise Linux 9 Operating system (free tier) comprising of One NFS server and 3 webservers .You can see the instance state that shows all 5 servers are currently running . The names are

NFS Server , WebServer1, WebServer2 WebServer3



1 Ubuntu Operating system comprising of the Database server. The name is DATABASE-SERVERUB



Then we proceed to configuring the NFS Server

NFS SERVER CONFIGURATION

We open git bash on visual studio code or whichever console is convenient to use. We are using git bash here with Visual Studio Code. We are connecting the ssh and typing yes and once the connection is successful, we proceed to naming the nfs server. This is so that we can be able to distinct each server by their names and avoid confusion with other server we would be configuring .The server name was edited as shown below

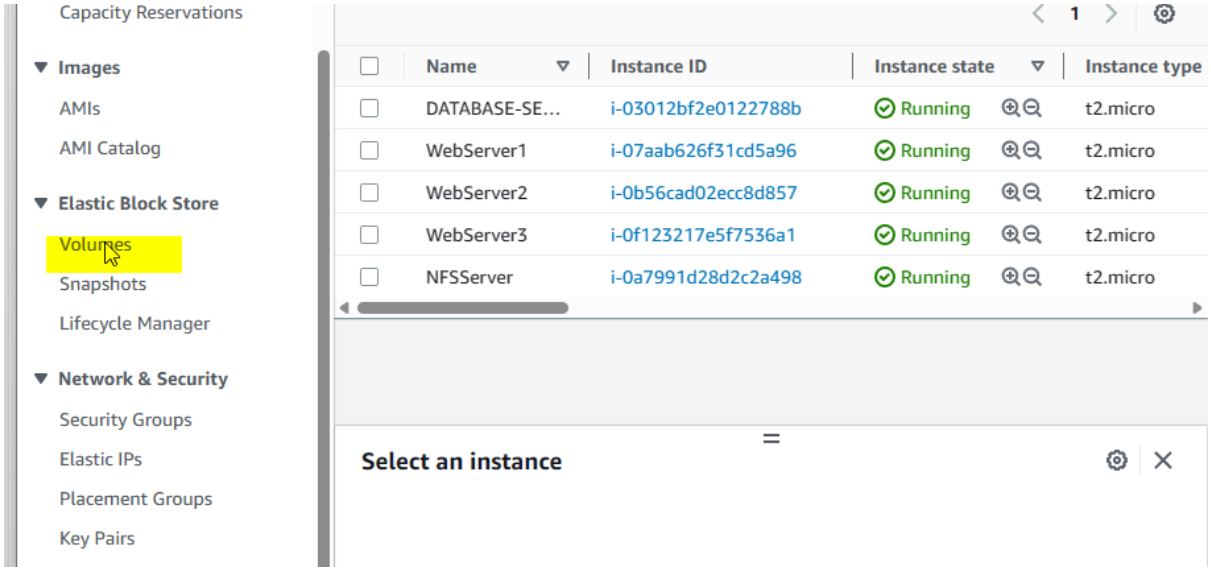
```
oshor@Oshority MINGW64 ~/Downloads (master)
$ ssh -i "NEW-WEBSERVER.pem" ec2-user@ec2-18-233-171-223.compute-1.amazonaws.com
The authenticity of host 'ec2-18-233-171-223.compute-1.amazonaws.com (18.233.171.223)' can't be established.
ED25519 key fingerprint is SHA256:jpq3Y8pZw8+Yrgq4Ko2RjatgIS9bQH3bW6zW53eg+K0.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:99: ec2-3-95-230-183.compute-1.amazonaws.com
    ~/.ssh/known_hosts:111: ec2-52-87-223-18.compute-1.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-233-171-223.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
Last login: Tue Jun 13 17:20:17 2023 from 81.152.238.208
[ec2-user@ip-172-31-85-81 ~]$ sudo hostname nfs-server
[ec2-user@ip-172-31-85-81 ~]$ bash
[ec2-user@nfs-server ~]$ sudo lsblk
NAME                    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
```

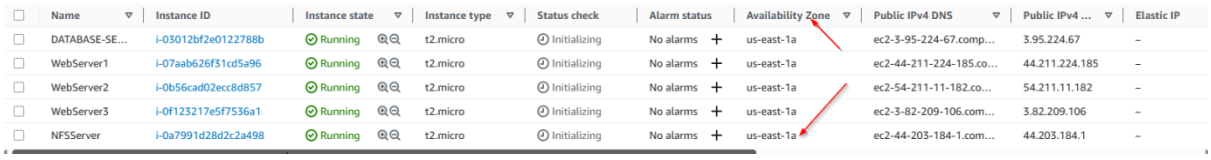
This is also done in the webserver1 and 2 as seen below.

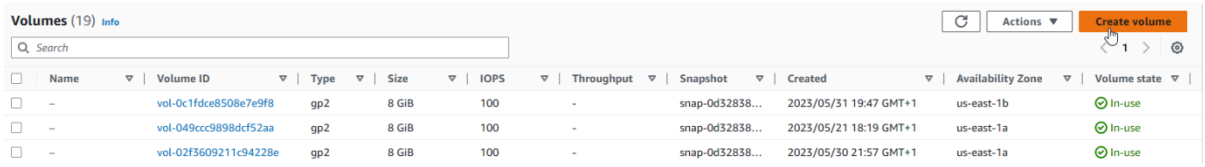
```
oshor@Oshority MINGW64 ~/Downloads (master)
$ ssh -i "NEW-WEBSERVER.pem" ec2-user@ec2-3-86-86-165.compute-1.amazonaws.com
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
Last login: Tue Jun 13 17:22:33 2023 from 81.152.238.208
[ec2-user@ip-172-31-93-237 ~]$ sudo hostname web-server1
[ec2-user@ip-172-31-93-237 ~]$ bash
[ec2-user@web-server1 ~]$ sudo systemctl start httpd.service
```

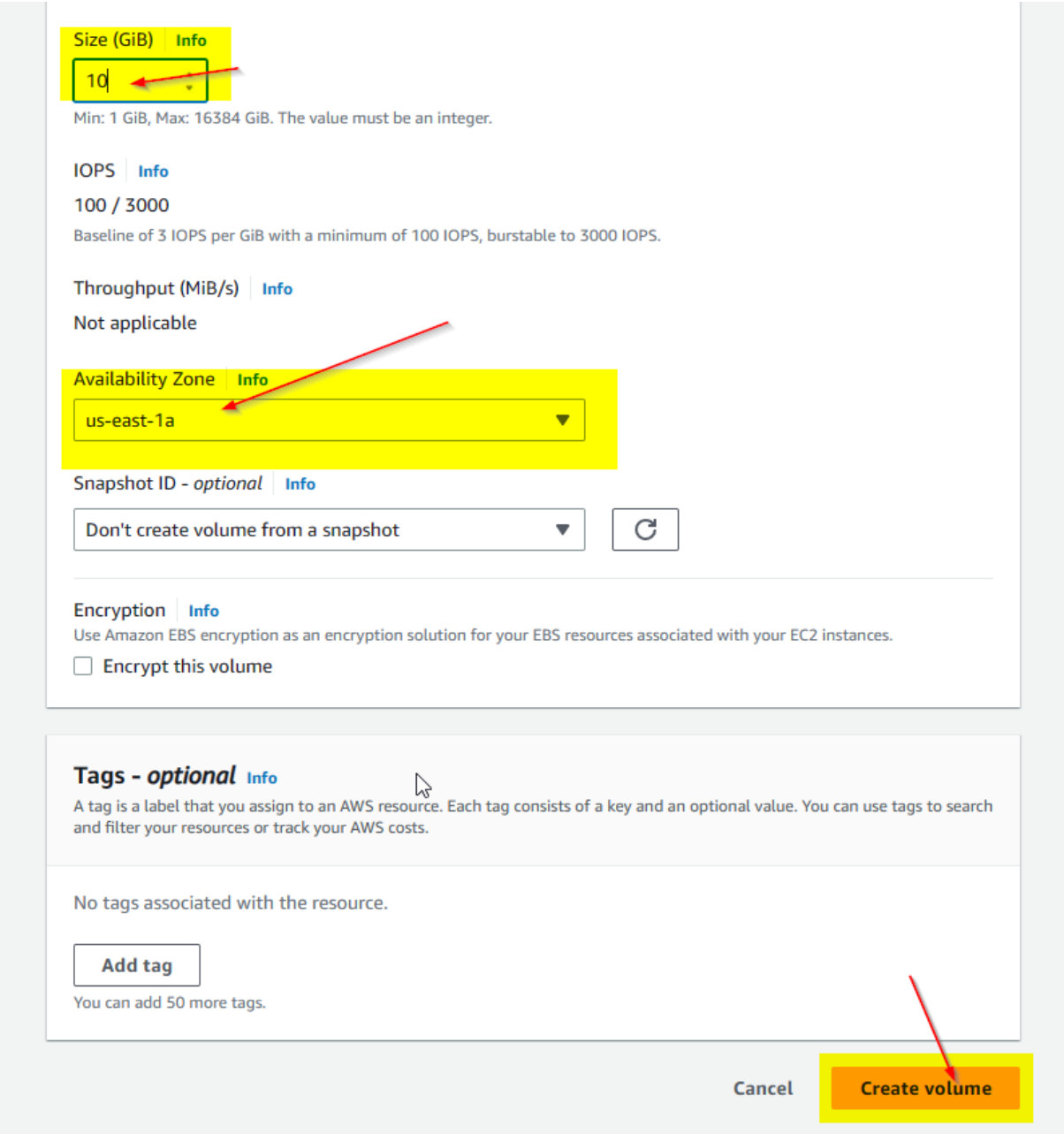
```
oshor@Oshority MINGW64 ~/Downloads (master)
$ ssh -i "NEW-WEBSERVER.pem" ec2-user@ec2-3-83-149-50.compute-1.amazonaws.com
The authenticity of host 'ec2-3-83-149-50.compute-1.amazonaws.com (3.83.149.50)' can't be established.
ED25519 key fingerprint is SHA256:pfHHrQNDQ52t+/4d1KxAe31/7LMHgHsR5SzfPjgGd4g.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-83-149-50.compute-1.amazonaws.com' (ED25519) to the list of known hos
ts.
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
[ec2-user@ip-172-31-86-216 ~]$ sudo hostname web-server2
[ec2-user@ip-172-31-86-216 ~]$ bash
[ec2-user@web-server2 ~]$ sudo yum update -y
```

The next step would be to go back to the AWS console click to create the volumes. You must also check the availability zones as they play a crucial role in the location in which the volumes are created. Ensure the sizes are 10 gig and create volume as seen below.


Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Name
Instance ID
Instance state
Instance type
DATABASE-SE...
i-03012bf2e0122788b
Running
t2.micro
WebServer1
i-07aab626f31cd5a96
Running
t2.micro
WebServer2
i-0b56cad02ecc8d857
Running
t2.micro
WebServer3
i-0f123217e5f7536a1
Running
t2.micro
NFSServer
i-0a7991d28d2c2a498
Running
t2.micro
Select an instance


Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
Public IPv4 ...
Elastic IP
DATABASE-SE...
i-03012bf2e0122788b
Running
t2.micro
Initializing
No alarms
us-east-1a
ec2-3-95-224-67.comp...
3.95.224.67
WebServer1
i-07aab626f31cd5a96
Running
t2.micro
Initializing
No alarms
us-east-1a
ec2-44-211-224-185.co...
44.211.224.185
WebServer2
i-0b56cad02ecc8d857
Running
t2.micro
Initializing
No alarms
us-east-1a
ec2-54-211-11-182.co...
54.211.11.182
WebServer3
i-0f123217e5f7536a1
Running
t2.micro
Initializing
No alarms
us-east-1a
ec2-3-82-209-106.com...
3.82.209.106
NFSServer
i-0a7991d28d2c2a498
Running
t2.micro
Initializing
No alarms
us-east-1a
ec2-44-203-184-1.com...
44.203.184.1


Volumes (19) Info
Search
Actions
Create volume
Name
Volume ID
Type
Size
IOPS
Throughput
Snapshot
Created
Availability Zone
Volume state
vol-0c1fdce8508e7e9f8
gp2
8 GiB
100
snap-0d32838...
2023/05/31 19:47 GMT+1
us-east-1b
In-use
vol-049ccc9898dcf52aa
gp2
8 GiB
100
snap-0d32838...
2023/05/21 18:19 GMT+1
us-east-1a
In-use
vol-02f3609211c94228e
gp2
8 GiB
100
snap-0d32838...
2023/05/30 21:57 GMT+1
us-east-1a
In-use



And you would repeat the same steps for the other 2 volumes. Go to the volume created, Chose the instance you are attaching to (NFS SERVER) and attach volumes and start using them .

Once done we would go back to the terminal and type the lsblk command to see the EBS volumes we attached. The we use the command below to partition the drive and use the help command to see the options available to add a new partition.

```
[ec2-user@nfs-server ~]$ lsblk]
-bash: lsblk]: command not found
[ec2-user@nfs-server ~]$ lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
xvda    202:0    0   10G  0 disk
├─xvda1 202:1    0    1M  0 part
├─xvda2 202:2    0  200M  0 part /boot/efi
├─xvda3 202:3    0  500M  0 part /boot
└─xvda4 202:4    0  9.3G  0 part /
xvdf    202:80   0   10G  0 disk
xvdg    202:96   0   10G  0 disk
xvdh    202:112  0   10G  0 disk
[ec2-user@nfs-server ~]$ sudo gdisk /dev/xvdf
GPT fdisk (gdisk) version 1.0.7

Partition table scan:
  MBR: not present
  BSD: not present
  APM: not present
  GPT: not present

Creating new GPT entries in memory.

Command (? for help): ?
b       back up GPT data to a file
c       change a partition's name
d       delete a partition
i       show detailed information on a partition
l       list known partition types
n       add a new partition
o       create a new empty GUID partition table (GPT)
p       print the partition table
q       quit without saving changes
```

Type “n” to add a new partition,

Choose 1 as the partition number and click enter button for the first and last sector .

Enter :8300 for the default file system as shown below

```
r       recovery and transformation options (experts only)
s       sort partitions
t       change a partition's type code
v       verify disk
w       write table to disk and exit
x       extra functionality (experts only)
?       print this menu

Command (? for help): n
Partition number (1-128, default 1): 1
First sector (34-20971486, default = 2048) or {+-}size{KMGTP}:
Last sector (2048-20971486, default = 20971486) or {+-}size{KMGTP}:
Current type is 8300 (Linux filesystem)
Hex code or GUID (L to show codes, Enter = 8300):
```

You can type “p” to view the partition table as shown below.

```
Total free space is 2014 sectors (1007.0 KiB)

Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048        20971486   10.0 GiB    8300  Linux filesystem

Command (? for help):
```

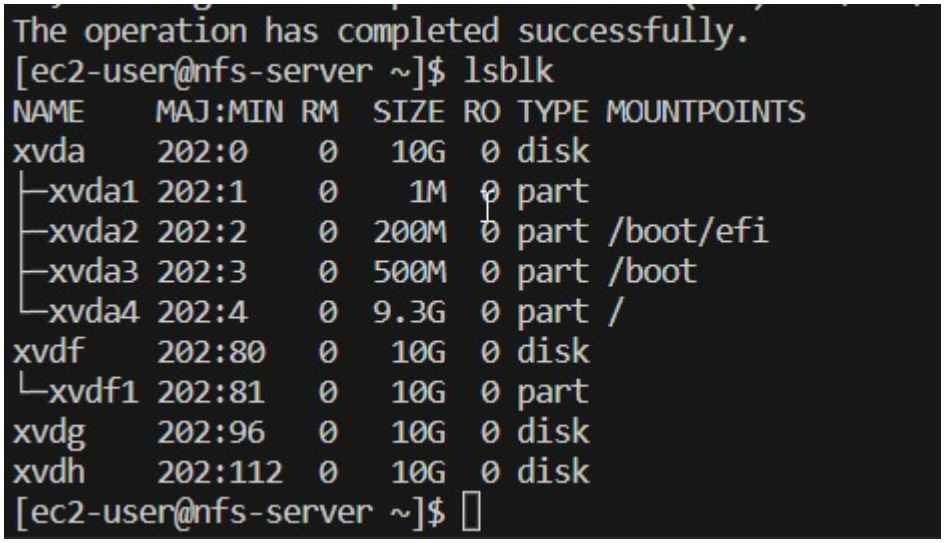
We use “w” to write the table and edit on the disk and type “w” and click enter and type “y” to proceed.

```
Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048        20971486   10.0 GiB    8300  Linux filesystem

Command (? for help): w

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING
PARTITIONS!!

Do you want to proceed? (Y/N): y
```

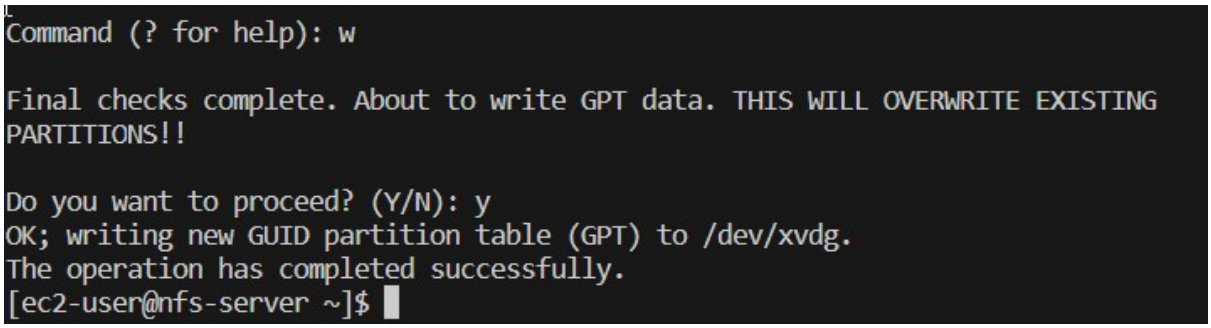
The it states that the operation was successful.

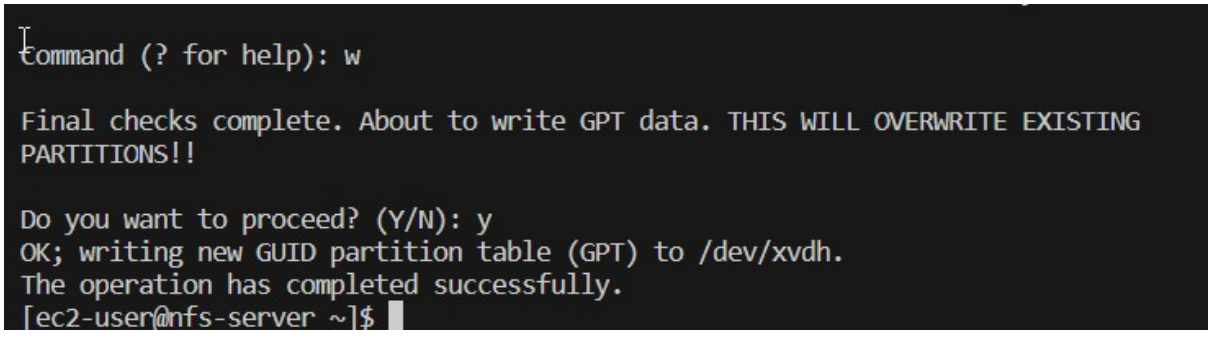
```
Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING
PARTITIONS!!

Do you want to proceed? (Y/N): y
OK; writing new GUID partition table (GPT) to /dev/xvdf.
The operation has completed successfully.
[ec2-user@nfs-server ~]$
```

Type lsblk command to check again and you would see that the xvdf file now has where the partition was created as seen below



Repeat the same steps and create the partition for g and h partitions and the results are shown below





We then proceed to install the lvm2 package.

```
[ec2-user@nfs-server ~]$ sudo yum install lvm2 -y
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Last metadata expiration check: 0:46:13 ago on Mon 12 Jun 2023 03:45:5þ PM UTC.
Dependencies resolved.
================================================================================
 Package                  Architecture  Version                 Repository      Size
================================================================================
Installing:
 lvm2                     x86_64        9:2.03.17-7.el9         rhel-9-baseos-rhui-rpms  1.5 M
Installing dependencies:
 device-mapper-event      x86_64        9:1.02.187-7.el9        rhel-9-baseos-rhui-rpms   36 k
 device-mapper-event-libs x86_64        9:1.02.187-7.el9        rhel-9-baseos-rhui-rpms   34 k
 device-mapper-persistent-data x86_64   0.9.0-13.el9            rhel-9-baseos-rhui-rpms  786 k
 libaio                   x86_64        0.3.111-13.el9          rhel-9-baseos-rhui-rpms   26 k
 lvm2-libs                x86_64        9:2.03.17-7.el9         rhel-9-baseos-rhui-rpms  1.0 M

Transaction Summary
```

After that is successfully installed, we use the lsblk command to check our 3 partitions created

```
  libaio-0.3.111-13.el9.x86_64                                  lvm2-9:

Complete!
[ec2-user@nfs-server ~]$ lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
xvda    202:0    0   10G  0 disk
├─xvda1 202:1    0    1M  0 part
├─xvda2 202:2    0  200M  0 part /boot/efi
├─xvda3 202:3    0  500M  0 part /boot
└─xvda4 202:4    0  9.3G  0 part /
xvdf    202:80   0   10G  0 disk
└─xvdf1 202:81   0   10G  0 part
xvdg    202:96   0   10G  0 disk
└─xvdg1 202:97   0   10G  0 part
xvdh    202:112  0   10G  0 disk
└─xvdh1 202:113  0   10G  0 part
[ec2-user@nfs-server ~]$
```

Next step would be to create a physical volume using the pvcreate command for the xvdf1, xvdg1 and xvdh1 respectively

```
[ec2-user@nfs-server ~]$ sudo pvcreate /dev/xvdf1
  Physical volume "/dev/xvdf1" successfully created.
  Creating devices file /etc/lvm/devices/system.devices
[ec2-user@nfs-server ~]$ sudo pvcreate /dev/xvdg1
  Physical volume "/dev/xvdg1" successfully created.
[ec2-user@nfs-server ~]$ sudo pvcreate /dev/xvdh1
  Physical volume "/dev/xvdh1" successfully created.
[ec2-user@nfs-server ~]$
```

We use the lsblk command to check the 3 physical volumes created

```
  Physical volume "/dev/xvdh1" successfully created.
[ec2-user@nfs-server ~]$ lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
xvda    202:0    0   10G  0 disk
├─xvda1 202:1    0    1M  0 part
├─xvda2 202:2    0  200M  0 part /boot/efi
├─xvda3 202:3    0  500M  0 part /boot
└─xvda4 202:4    0  9.3G  0 part /
xvdf    202:80   0   10G  0 disk
└─xvdf1 202:81   0   10G  0 part
xvdg    202:96   0   10G  0 disk
└─xvdg1 202:97   0   10G  0 part
xvdh    202:112  0   10G  0 disk
└─xvdh1 202:113  0   10G  0 part
```

Use the pvs command to check the 3 physical volumes.

```
[ec2-user@nfs-server ~]$ sudo pvs
  PV         VG Fmt  Attr PSize   PFree
  /dev/xvdf1    lvm2 ---  <10.00g <10.00g
  /dev/xvdg1    lvm2 ---  <10.00g <10.00g
  /dev/xvdh1    lvm2 ---  <10.00g <10.00g
```

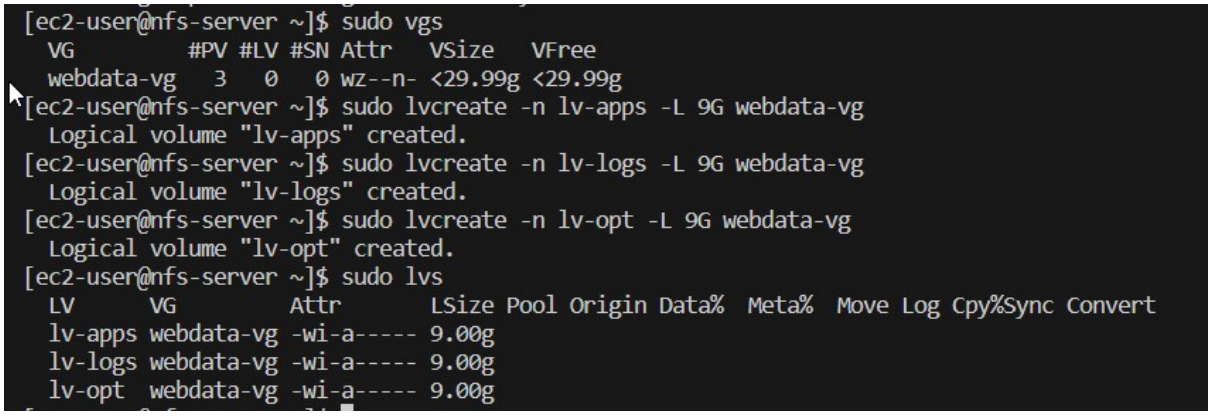
We then use the vg-create command to let the 3 physical volume be seen as 1 logical volume and we name is webdata-vg as shown below

```
[ec2-user@nfs-server ~]$ sudo vgcreate     webdata-vg /dev/xvdh1  /dev/xvdg1 /dev/xvdf1
  Volume group "webdata-vg" successfully created
[ec2-user@nfs-server ~]$
```

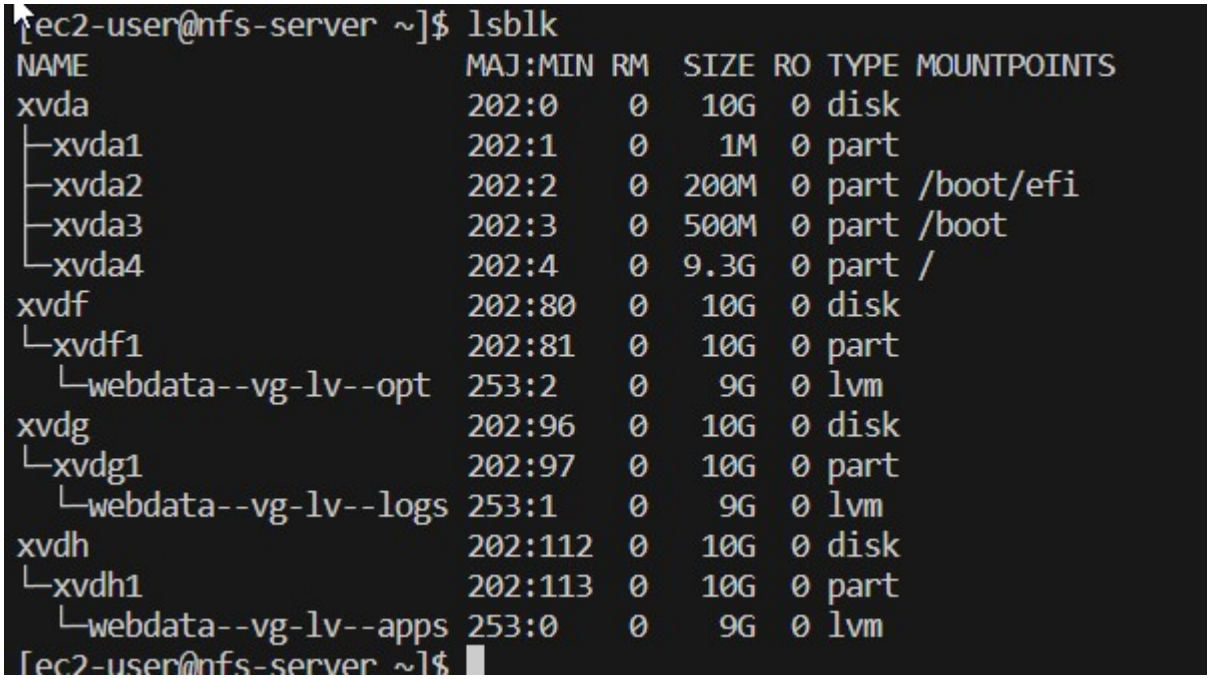
Use “vgs” to check if it was implemented successfully.

```
[ec2-user@nfs-server ~]$ sudo vgcreate     webdata-vg /dev/xvdh1  /dev/xvdg1 /dev/xvdf1
  Volume group "webdata-vg" successfully created
[ec2-user@nfs-server ~]$ sudo vgs
  VG         #PV #LV #SN Attr   VSize   VFree
  webdata-vg   3   0   0 wz--n- <29.99g <29.99g
[ec2-user@nfs-server ~]$
```

Next step would be to create 3 logical volumes apps, logs and opt as shown below and confirm it was implemented successfully

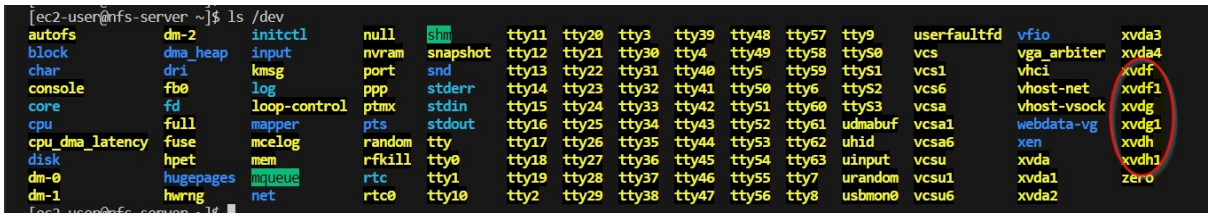


Using the lsblk command to view all we have done since starting the partition creation, we can see the full detailed structure created as shown below



We need to format the extra metadata that came along with this volume with the mkfs command below.

First, we check the devices using the /dev command



And format the 3 logical volume (command apps, logs and opt) with the mkfs command

```
[ec2-user@nfs-server ~]$ sudo mkfs -t xfs  /dev/webdata-vg/lv-apps
meta-data=/dev/webdata-vg/lv-apps isize=512    agcount=4, agsize=589824 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=1 inobtcount=1
data     =                       bsize=4096   blocks=2359296, 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
[ec2-user@nfs-server ~]$ sudo mkfs -t xfs  /dev/webdata-vg/lv-logs
meta-data=/dev/webdata-vg/lv-logs isize=512    agcount=4, agsize=589824 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=1 inobtcount=1
data     =                       bsize=4096   blocks=2359296, 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
[ec2-user@nfs-server ~]$ sudo mkfs -t xfs  /dev/webdata-vg/lv-opt
meta-data=/dev/webdata-vg/lv-opt isize=512    agcount=4, agsize=589824 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=1 inobtcount=1
data     =                       bsize=4096   blocks=2359296, 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
[ec2-user@nfs-server ~]$
```

While in the NFS server we are to create some mount point on the mount directory on the logical volumes. We create the 3 directories as shown below and check it was created successfully.

```
[ec2-user@nfs-server ~]$ sudo mkdir /mnt/apps
sudo mkdir /mnt/logs
sudo mkdir /mnt/opt
[ec2-user@nfs-server ~]$ ls
[ec2-user@nfs-server ~]$ ls/mnt
-bash: ls/mnt: No such file or directory
[ec2-user@nfs-server ~]$ ls /mnt
apps  logs  opt
[ec2-user@nfs-server ~]$
```

We would then mount the 3 logical volume path with the 3 mounting points as shown below

```
[ec2-user@nfs-server ~]$ sudo mount /dev/webdata-vg/lv-apps /mnt/apps
[ec2-user@nfs-server ~]$ sudo mount /dev/webdata-vg/lv-logs /mnt/logs
sudo mount /dev/webdata-vg/lv-opt /mnt/opt
[ec2-user@nfs-server ~]$
```

The next step would be to install the NFS server. Please take note that this is the place we store all the data and by storing it ,the webserver automatically becomes the NFS client.

The installation of the NFS SERVER is done with the commands below

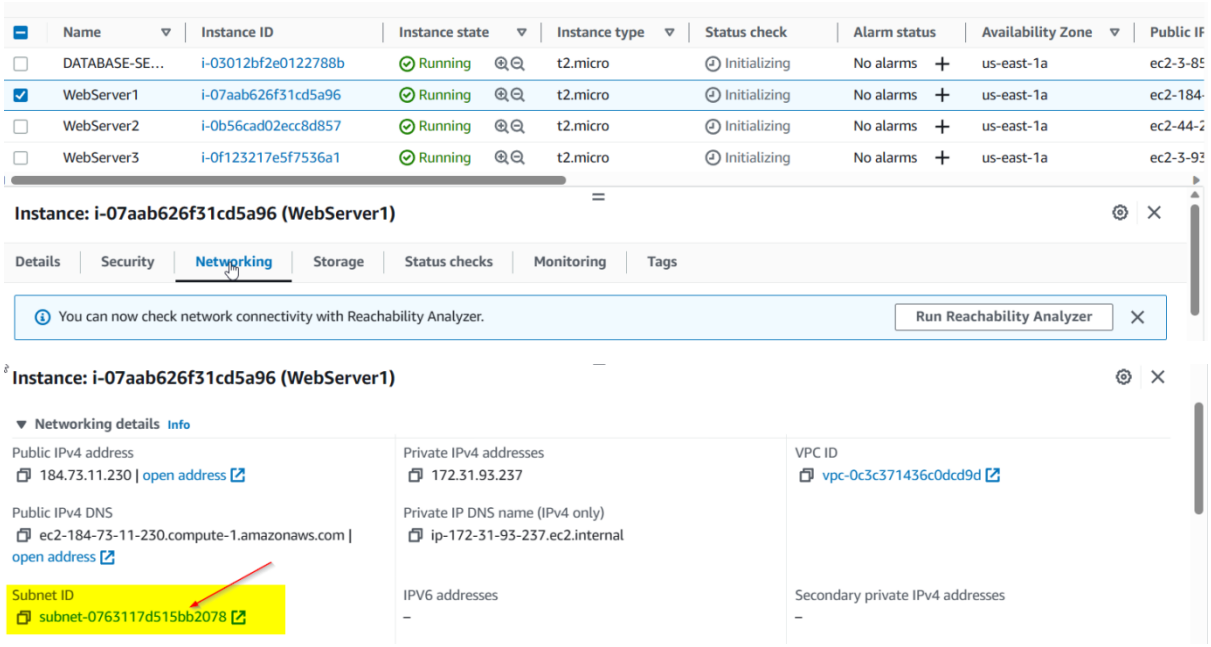
```
[ec2-user@nfs-server ~]$ sudo yum -y update
sudo yum install nfs-utils -y
sudo systemctl start nfs-server.service
sudo systemctl enable nfs-server.service
sudo systemctl status nfs-server.service
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Red Hat Enterprise Linux 9 for x86_64 - AppStream from RHUI (RPMs)                76 kB/s | 4.5 kB     00:00
```

NFS server installation completed, enabled and actively running.

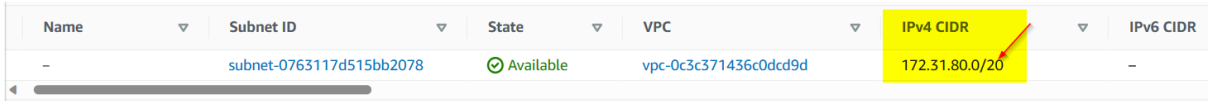
```
Installed:
  gssproxy-0.8.4-5.el9_2.x86_64      keyutils-1.6.3-1.el9.x86_64        libev-4.33-5.el9.x86_64            libnfsidmap-1:2.5.4-18.el9.x86_64
  libtirpc-1.3.3-1.el9.x86_64        libverto-libev-0.3.2-3.el9.x86_64  nfs-utils-1:2.5.4-18.el9.x86_64    quota-1:4.06-6.el9.x86_64
  quota-nls-1:4.06-6.el9.noarch      rpcbind-1.2.6-5.el9.x86_64         sssd-nfs-idmap-2.8.2-2.el9.x86_64

Complete!
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-server.service → /usr/lib/systemd/system/nfs-server.service.
● nfs-server.service - NFS server and services
     Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled; preset: disabled)
     Active: active (exited) since Mon 2023-06-12 17:06:57 UTC; 522ms ago
   Main PID: 15655 (code=exited, status=0/SUCCESS)
        CPU: 29ms

Jun 12 17:06:56 ip-172-31-85-81.ec2.internal systemd[1]: Starting NFS server and services...
Jun 12 17:06:57 ip-172-31-85-81.ec2.internal systemd[1]: Finished NFS server and services.
```

Press q to quit and get back on the terminal

The next step would be to export the mount for webservers from subnet cidr. Click on networking button and then proceed to click the subnet ID link as shown below.



The ipv4 cidr is shown below



Please note: The lower the number of cidr the higher the number of computers that can gain access to that server .This can connect 4096 computers and shows the importance of cidr. Copy the cidr for future use.

We then navigate back to give permissions to allow our webservers access to read, write and execute .We can give permission to root or simply say nobody .

In this case we would say nobody because we want everyone to have access to it and it is not own by anyone.

Before the permission was changed

```
[ec2-user@nfs-server ~]$ ls -larth  /mnt/apps/
total 0
drwxr-xr-x. 2 root root  6 Jun 12 16:51 .
drwxr-xr-x. 5 root root 41 Jun 12 16:56 ..
```

After permission was allowed. see below as permission has been granted to nobody as well as the chmod 777 permission.

```
[ec2-user@nfs-server ~]$ sudo chown -R nobody: /mnt/apps
sudo chown -R nobody: /mnt/logs
sudo chown -R nobody: /mnt/opt

sudo chmod -R 777 /mnt/apps
sudo chmod -R 777 /mnt/logs
sudo chmod -R 777 /mnt/opt
[ec2-user@nfs-server ~]$ ls -larth  /mnt/apps/
total 0
drwxrwxrwx. 2 nobody nobody  6 Jun 12 16:51 .
drwxr-xr-x. 5 root   root   41 Jun 12 16:56 ..
```

We would then proceed to restart our NFS server after mounting of the logical volumes.

We then proceed to configure access to NFS from our clients within the same subnet ipv4 cidr as shown below.

```
drwxr-xr-x. 5 root   root   41 Jun 12 16:56 ..
[ec2-user@nfs-server ~]$ sudo systemctl restart nfs-server.service
[ec2-user@nfs-server ~]$ sudo vi /etc/exports
[ec2-user@nfs-server ~]$ sudo vi /etc/exports
```

Paste the code below in /etc/exports, Save and quit to get back to the terminal

```
/mnt/apps 172.31.80.0/20(rw,sync,no_all_squash,no_root_squash)
/mnt/logs 172.31.80.0/20(rw,sync,no_all_squash,no_root_squash)
/mnt/opt 172.31.80.0/20(rw,sync,no_all_squash,no_root_squash)
~
~
~
~
```

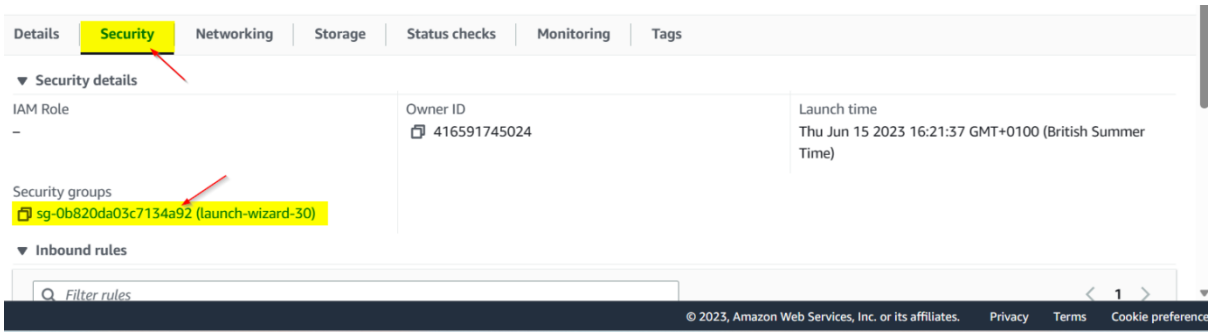
Verify they are properly mounted successfully with the command below.

```
[ec2-user@nfs-server ~]$ sudo exportfs -arv
exporting 172.31.80.0/20:/mnt/opt
exporting 172.31.80.0/20:/mnt/logs
exporting 172.31.80.0/20:/mnt/apps
```

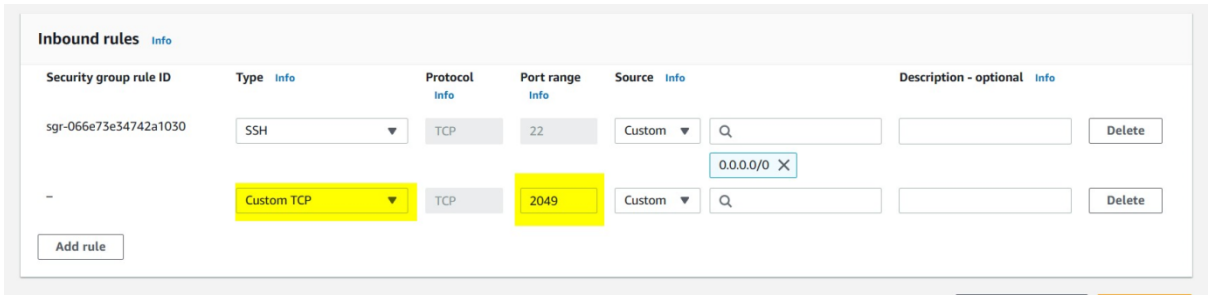
The next process is to go back to AWS security group for NFS server to open the ports tcp and udp via the inbound rules

```
[ec2-user@nfs-server ~]$ rpcinfo -p | grep nfs
    100003    3   tcp   2049  nfs
    100003    4   tcp   2049  nfs
    100227    3   tcp   2049  nfs_acl
[ec2-user@nfs-server ~]$
```

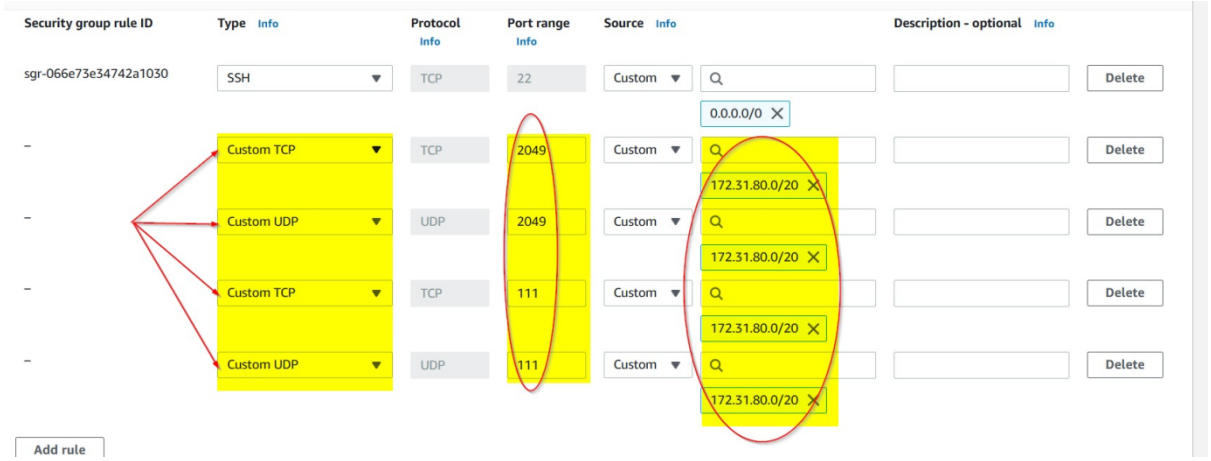
Click security and security group



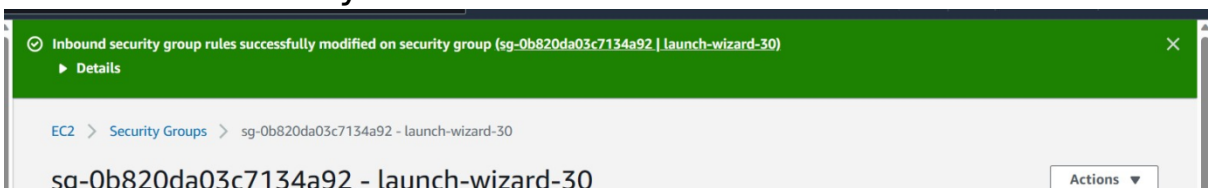
Add all rules.



Including the subnet cidr block of the webservers to access the NFS server.



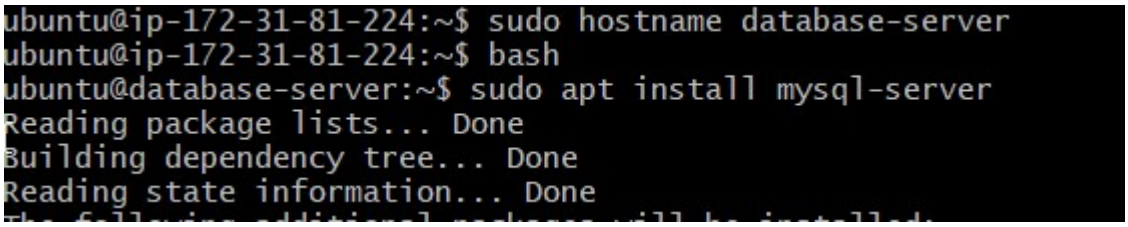
Added successfully



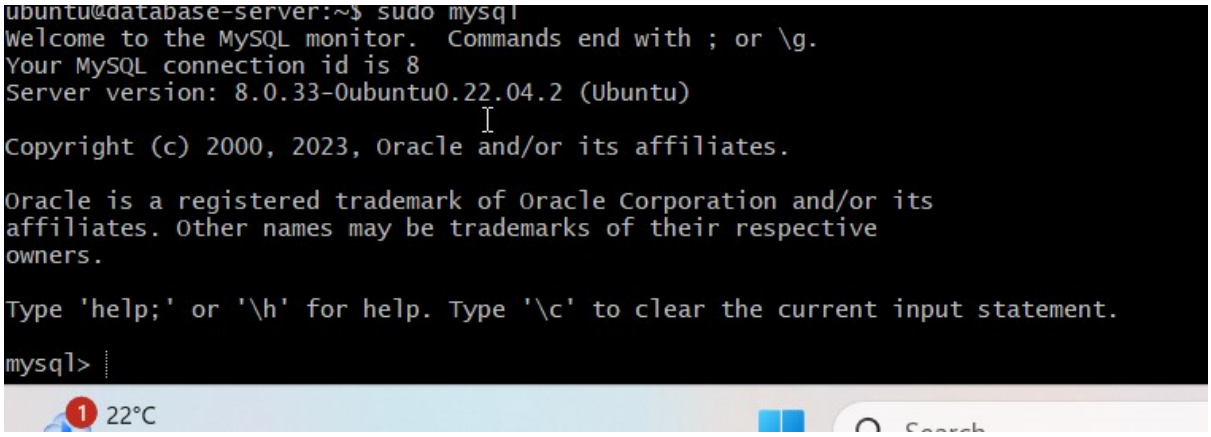
Now we proceed to the next stage to configure the database server.

DATABASE SERVER CONFIGURATION

To configure our database server, we would install MySQL server and also created an hostname for the ip address for easy use.



To access the MySQL environment, we use the command below



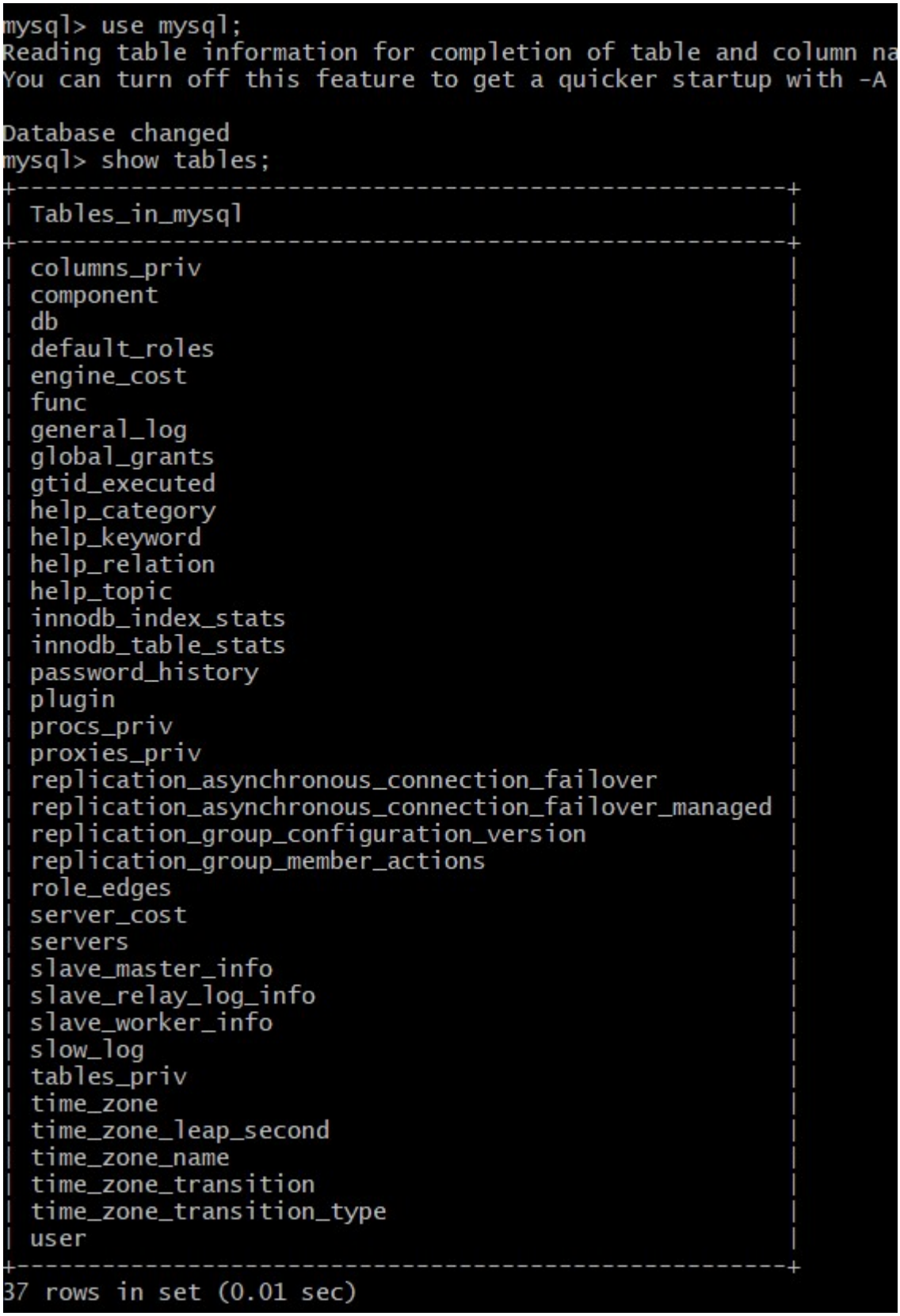
but when we exit out of it, we need to use this

```
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.33-0ubuntu0.22.04.2 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.
```

The next thing would be to create a database. Lets show the current state of the database and it is as shown as below

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.01 sec)
```

To use and show all tables in the MySQL ,



We would be creating the following and ensuring the ipv4 cidr is included when creating the database server.

i)Database name: tooling ii) Database username: webaccess

We would be using authentication by granting all privileges to webaccess from tooling including all tables in it.

```
mysql> create database tooling;
Query OK, 1 row affected (0.02 sec)

mysql> create user 'webaccess'@'172.31.80.0/20' identified by 'password';
Query OK, 0 rows affected (0.04 sec)

mysql> grant all priviledges on tooling.* to 'webaccess'@'172.31.80.0/20';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual t
'' at line 1
mysql> grant all privileges on tooling.* to 'webaccess'@'172.31.80.0/20';
Query OK, 0 rows affected (0.01 sec)
```

Now lets proceed by flushing the privileges and show the database to see if we successfully created the databases.

```
mysql> flush privileges;
Query OK, 0 rows affected (0.01 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| tooling            |
+--------------------+
5 rows in set (0.00 sec)
```

Type: “Use tooling” to activate the database and show tables

```
mysql> use tooling
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

```
mysql> use tooling;
Database changed
mysql> show tables;
Empty set (0.00 sec)

mysql>
```

The new user called webaccess is shown below.

```
mysql> select user,host from user;
+------------------+---------------+
| user             | host          |
+------------------+---------------+
| webaccess        | 172.31.80.0/20 |
| debian-sys-maint | localhost     |
| mysql.infoschema | localhost     |
| mysql.session    | localhost     |
| mysql.sys        | localhost     |
| root             | localhost     |
+------------------+---------------+
6 rows in set (0.00 sec)
```

Now we navigate back to the webservers.

WEBSERVERS

We would be working on the 3 webservers but also note that they would also represent as an NFS client to our NFS server when it stores information. We intend to display the web page on the webserver. We want to be able to read and write on the web page and we intend to gain access to /var/www to retrieve the html

We would proceed with the first step by configuring the NFS CLIENTS on all 3 servers.

We have to update the package manager

```
[ec2-user@web-server2 ~]$ sudo yum update -y
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-m

Last metadata expiration check: 0:03:56 ago on Mon 12 Jun 2023 08:13:56 PM UTC.
```

And install nfs-util

```
[ec2-user@web-server2 ~]$ sudo yum install nfs-utils nfs4-acl-tools -y
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-

Last metadata expiration check: 0:05:18 ago on Mon 12 Jun 2023 08:13:56 PM UTC.
Dependencies resolved.
```

Then we must create the /var/www directory and then mount the path where we would be installing the APACHE server (httpd).

Mounting is where we would be doing the connection linking the NFS CLIENT (Webserver)and the NFS server

We should check the hard disk df -h command to check how much disk available on the computer.

```
Complete!
[ec2-user@web-server2 ~]$ sudo mkdir /var/www/
[ec2-user@web-server2 ~]$ sudo mount -t nfs -o rw,nosuid 172.31.85.81:/mnt/apps /var/www
[ec2-user@web-server2 ~]$ df -h
Filesystem              Size  Used Avail Use% Mounted on
devtmpfs                4.0M     0  4.0M   0% /dev
tmpfs                   385M     0  385M   0% /dev/shm
tmpfs                   154M  5.7M  149M   4% /run
/dev/xvda4              9.4G  1.3G  8.1G  14% /
/dev/xvda3              495M  153M  343M  31% /boot
/dev/xvda2              200M  8.0K  200M   1% /boot/efi
tmpfs                    77M     0   77M   0% /run/user/1000
172.31.85.81:/mnt/apps  9.0G   97M  8.9G   2% /var/www
[ec2-user@web-server2 ~]$
```

To confirm if the mounting was successful we can try an experiment by changing directory to /var/www , and checking its content which should be blank, proceed to create a file name testfile .

```
[ec2-user@web-server1 ~]$ sudo mkdir /var/www/
[ec2-user@web-server1 ~]$ sudo mount -t nfs -o rw,nosuid 172.31.85.81:/mnt/apps /var/www
[ec2-user@web-server1 ~]$ df -h
Filesystem              Size  Used Avail Use% Mounted on
devtmpfs                4.0M     0  4.0M   0% /dev
tmpfs                   385M     0  385M   0% /dev/shm
tmpfs                   154M  5.8M  148M   4% /run
/dev/xvda4              9.4G  1.3G  8.1G  14% /
/dev/xvda3              495M  153M  343M  31% /boot
/dev/xvda2              200M  8.0K  200M   1% /boot/efi
tmpfs                    77M     0   77M   0% /run/user/1000
172.31.85.81:/mnt/apps  9.0G   97M  8.9G   2% /var/www
[ec2-user@web-server1 ~]$ cd /var/www
[ec2-user@web-server1 www]$ ls
[ec2-user@web-server1 www]$ touch testfile
[ec2-user@web-server1 www]$
```

Navigate back to NFS server. Change directory to /mnt and check its content. You would find the file testfile there .

```
[ec2-user@nfs-server ~]$ cd /mnt/apps/
[ec2-user@nfs-server apps]$ ls
[ec2-user@nfs-server apps]$ ls
testfile
```

This confirms that our NFS CLIENT has been successfully connected to store information on the NFS SERVER .

Please note that when you reboot this server it would not retain the connecting state and connection would be lost but to make it connect constantly we need to modify the fstab file .

```
[ec2-user@web-server1 www]$ sudo vi /etc/fstab
[ec2-user@web-server1 www]$ sudo yum install httpd -y
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.
```

```
UUID=287d9c0b-0e0f-4e92-8534-45733aa3dc68       /       xfs     defaults        0       0
UUID=7bc24af7-289d-4bce-b17e-300c3aafe968       /boot   xfs     defaults        0       0
UUID=7B77-95E7  /boot/efi       vfat    defaults,uid=0,gid=0,umask=077,shortname=winnt  0       2
172.31.85.81:/mnt/apps /var/www nfs defaults 0 0
```

Proceed to install Apache and PHP from a third party Remi’s repository shown below

```
sudo yum install httpd -y

sudo dnf install
https://dl.fedoraproject.org/pub/epel/epel-release-
latest-8.noarch.rpm

sudo dnf install dnf-utils
http://rpms.remirepo.net/enterprise/remi-release-8.rpm

sudo dnf module reset php

sudo dnf module enable php:remi-7.4

sudo dnf install php php-opcache php-gd php-curl php-
mysqlnd

sudo systemctl start php-fpm

sudo systemctl enable php-fpm

setsebool -P httpd_execmem 1
```

Install Apache

```
[ec2-user@web-server1 www]$ sudo vi /etc/fstab
[ec2-user@web-server1 www]$ sudo yum install httpd -y
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.
```

Installation continued and type y

```
[ec2-user@web-server1 www]$ sudo dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-
8.noarch.rpm
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.
```

```
[ec2-user@web-server1 www]$ ^[[200~sudo dnf install dnf-utils http://rpms.remirepo.net/enterprise/remi-
release-8.rpm
-bash: $'\E[200~sudo': command not found
[ec2-user@web-server1 www]$ sudo dnf install dnf-utils http://rpms.remirepo.net/enterprise/remi-release
-8.rpm
Updating Subscription Management repositories.
Unable to read consumer identity
```

All our packages are installed then we check if PHP is available on our repository version 8.1 which is a better upgrade compare to 7.4

```
[ec2-user@web-server1 www]$ sudo dnf module list php
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Last metadata expiration check: 0:01:46 ago on Mon 12 Jun 2023 08:47:21 PM UTC.
Red Hat Enterprise Linux 9 for x86_64 - AppStream from RHUI (RPMs)
Name            Stream          Profiles                                Summary
php             8.1             common [d], devel, minimal              PHP scripting language
```

You can still run the command without the: remi-7.4 to get the installation and click yes

```
[ec2-user@web-server1 www]$ sudo dnf module reset php
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Last metadata expiration check: 0:02:07 ago on Mon 12 Jun 2023 08:47:21 PM UTC.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@web-server1 www]$
```

```
 Package                 Architecture          Version                 Repository              Size
Enabling module streams:
 php                                           8.1

Transaction Summary

Is this ok [y/N]: y
Complete!
[ec2-user@web-server1 www]$ sudo dnf install php php-opcache php-gd php-curl php-mysqlnd
```

Then we proceed to install the PHP and type y and click enter

```
[ec2-user@web-server1 www]$ sudo dnf install php php-opcache php-gd php-curl php-mysqlnd
Updating Subscription Management repositories.
```

```
 Package                 Architecture          Version                 Repository              Size
Enabling module streams:
 php                                           8.1

Transaction Summary

Is this ok [y/N]: y
Complete!
[ec2-user@web-server1 www]$ sudo dnf install php php-opcache php-gd php-curl php-mysqlnd
```

We then proceed to start PHP and enable it and then check the status to know if it is running .This can be seen below

```
[ec2-user@web-server1 www]$ sudo systemctl start php-fpm
[ec2-user@web-server1 www]$ sudo systemctl enable php-fpm
Created symlink /etc/systemd/system/multi-user.target.wants/php-fpm.service → /usr/lib/systemd/system/p
hp-fpm.service.
[ec2-user@web-server1 www]$ sudo systemctl status php-fpm
● php-fpm.service - The PHP FastCGI Process Manager
     Loaded: loaded (/usr/lib/systemd/system/php-fpm.service; enabled; preset: disabled)
     Active: active (running) since Mon 2023-06-12 20:54:40 UTC; 1min 3s ago
   Main PID: 15832 (php-fpm)
     Status: "Processes active: 0, idle: 5, Requests: 0, slow: 0, Traffic: 0req/sec"
      Tasks: 6 (limit: 4421)
     Memory: 13.1M
        CPU: 60ms
     CGroup: /system.slice/php-fpm.service
             ├─15832 "php-fpm: master process (/etc/php-fpm.conf)"
             ├─15833 "php-fpm: pool www"
             ├─15834 "php-fpm: pool www"
             ├─15835 "php-fpm: pool www"
             ├─15836 "php-fpm: pool www"
             └─15837 "php-fpm: pool www"

Jun 12 20:54:40 ip-172-31-93-237.ec2.internal systemd[1]: Starting The PHP FastCGI Process Manager...
Jun 12 20:54:40 ip-172-31-93-237.ec2.internal systemd[1]: Started The PHP FastCGI Process Manager.
[ec2-user@web-server1 www]$
```

Next is to set a Boolean and ensure that port 80 is opened.

```
[ec2-user@web-server2 ~]$ sudo setsebool -P httpd_execmem 1
[ec2-user@web-server2 ~]$ curl localhost
curl: (7) Failed to connect to localhost port 80: Connection refused
```

Run the mount command and check the df -h command below

```
[ec2-user@web-server2 ~]$ sudo mount -t nfs -o rw,nosuid 172.31.85.81:/mnt/logs /var/log/httpd
[ec2-user@web-server2 ~]$ sudo df -h
Filesystem              Size  Used Avail Use% Mounted on
devtmpfs                4.0M     0  4.0M   0% /dev
tmpfs                   385M     0  385M   0% /dev/shm
tmpfs                   154M  4.4M  150M   3% /run
/dev/xvda4              9.4G  1.5G  7.9G  17% /
/dev/xvda3              495M  153M  343M  31% /boot
/dev/xvda2              200M  8.0K  200M   1% /boot/efi
172.31.85.81:/mnt/apps  9.4G  1.3G  8.1G  14% /var/www
172.31.85.81:/mnt/logs  9.4G  1.3G  8.1G  14% /var/log/httpd
tmpfs                    77M     0   77M   0% /run/user/1000
```

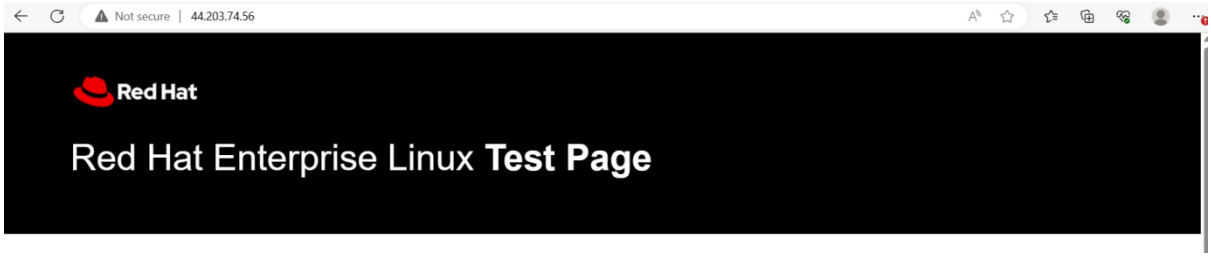
You can now start the httpd and check status as well.

```
[ec2-user@web-server2 ~]$ sudo systemctl start httpd
[ec2-user@web-server2 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
    Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─php-fpm.conf
     Active: active (running) since Wed 2023-06-14 07:33:38 UTC; 7s ago
       Docs: man:httpd.service(8)
   Main PID: 4061 (httpd)
     Status: "Started, listening on: port 80"
      Tasks: 213 (limit: 4421)
     Memory: 28.1M
        CPU: 77ms
     CGroup: /system.slice/httpd.service
```

Enable the httpd

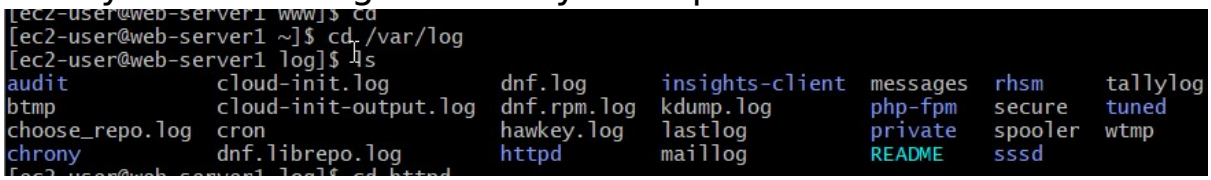
```
[ec2-user@web-server2 ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[ec2-user@web-server2 ~]$ sudo blkid
/dev/xvda4: LABEL="root" UUID="287d9c0b-0e0f-4e92-8534-45733aa3dc68" TYPE="xfs" PARTUUID="6264d520-3fb9-423f-8ab8-7a0a8e3d3562"
/dev/xvda2: SEC_TYPE="msdos" UUID="7B77-95E7" TYPE="vfat" PARTUUID="68b2905b-df3e-4fb3-80fa-49d1e773aa33"
/dev/xvda3: LABEL="boot" UUID="7bc24af7-289d-4bce-b17e-300c3aafe968" TYPE="xfs" PARTUUID="cb07c243-bc44-4717-853e-28852021225b"
/dev/xvda1: PARTUUID="fac7f1fb-3e8d-4137-a512-961de09a5549"
```

We can check the webserver and copy the public Ip address.



We have been able to configure the server successfully.

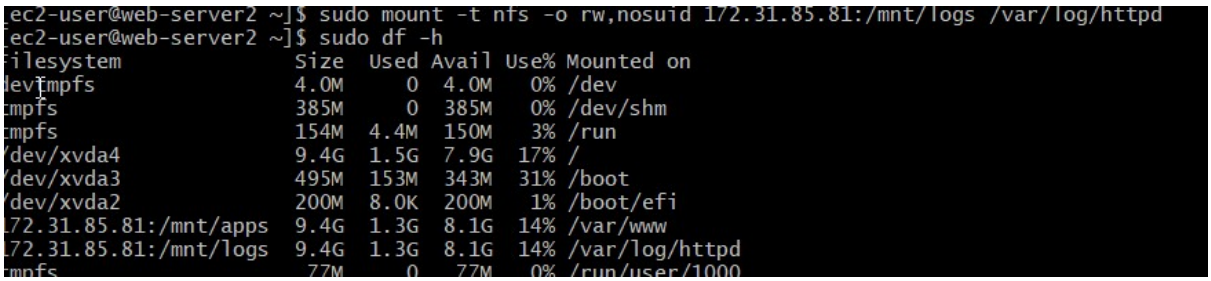
We have to locate the log folder for apache httpd on the webserver .Change directory to /var/log and check its content, then you can change directory to httpd

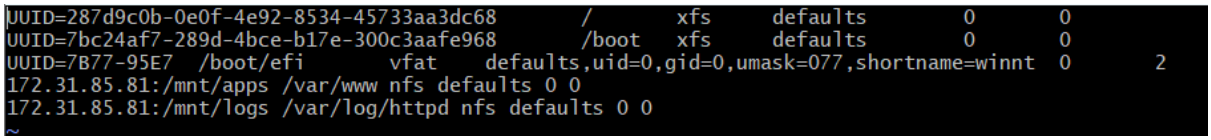


Please note if all troubleshooting fails, always remember selinux to help you out in terms of permission

Run the mount command and check the df -h command below

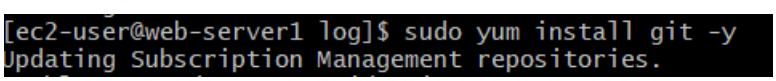
We should make sure we mount the /mnt/logs on /var/log/httpd and add this file to the fstab. This is important for rebooting as well to keep the connection available at all times making sure there is persistence. See below





We now proceed create a fork a repository to retrieve the source code from the git hub account.

First install git on the webserver



Once installed then you clone the https repo.

```
[ec2-user@web-server1 ~]$ git clone https://github.com/eyewande2022/tooling.git
Cloning into 'tooling'...
remote: Enumerating objects: 234, done.
remote: Total 234 (delta 0), reused 0 (delta 0), pack-reused 234
Receiving objects: 100% (234/234), 282.72 KiB | 17.67 MiB/s, done.
Resolving deltas: 100% (130/130), done.
[ec2-user@web-server1 ~]$ ls
```

Then you proceed to deploying the website code into the webserver. Change directory into tooling and check its contents.

Then we should copy all files in the tooling directory into the var/www/html file.

From the tooling directory, Change directory to var/www/html and view the content to the see copied files from tooling shown below.

```
ec2-user@web-server1 ~]$ cd tooling
ec2-user@web-server1 tooling]$ ls
pache-config.conf  Dockerfile  html  Jenkinsfile  README.md  start-apache  tooling-db.sql
ec2-user@web-server1 tooling]$ sudo cp -R . /var/www/html
ec2-user@web-server1 tooling]$ cd
ec2-user@web-server1 ~]$ cd /var/www/html
ec2-user@web-server1 html]$ ls
dmin_tooling.php   Dockerfile     img          login.php     start-apache    tooling_stylesheets.css
pache-config.conf  functions.php  index.php    README.md     style.css
reate_user.php     html           Jenkinsfile  register.php  tooling-db.sql
```

From the html directory change directory into /etc/httpd/conf.d and use the cat command to view the welcome.conf file .

```
[ec2-user@web-server1 html]$ cd /etc/httpd/conf.d/
[ec2-user@web-server1 conf.d]$ ls
autoindex.conf  php.conf  README  userdir.conf  welcome.conf
[ec2-user@web-server1 conf.d]$ cat welcome.conf
#
# This configuration file enables the default "Welcome" page if there
# is no default index page present for the root URL.  To disable the
# Welcome page, comment out all the lines below.
#
# NOTE: if this file is removed, it will be restored on upgrades.
#
```

We would now edit the functions.php file in the html folder.

```
[ec2-user@web-server1 html]$ sudo vi /var/www/html/functions.php
[ec2-user@web-server1 html]$ cd
```

This helps us connect the application to the database. We insert the database private ip, database username ,password and database name as shown below. Please note this only needs to be configured on one of the server and then its connected.

```
<?php
session_start();

// connect to database
$db = mysqli_connect('172.31.81.224', 'webaccess', '[redacted]', 'tooling');

// Check connection
// if (mysqli_connect_errno()) {
// echo "Failed to connect to MySQL: " . mysqli_connect_error();
// exit();
// }
// else{
// echo "connected";
// }

// variable declaration
$username = "";
$email    = "";
```

Then we proceed by applying the tooling-db.sql script to our data base using the command below. When entering the data we can only put the database private ip ,database username and database name but not the password as it is confidential ,so we would leave it blank and would be executed

We are going to install MySQL client because the action would be taking place on the webserver

```
[ec2-user@web-server1 tooling]$ sudo yum install mysql -y
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Last metadata expiration check: 1:11:00 ago on Mon 12 Jun 2023 08:47:21 PM UTC.
Dependencies resolved.
================================================================================
 Package                    Arch       Version            Repository          Size
================================================================================
Installing:
 mysql                      x86_64     8.0.32-1.el9_2     rhel-9-appstream-rhui-rpms  2.8 M
Installing dependencies:
 mariadb-connector-c-config noarch     3.2.6-1.el9_0      rhel-9-appstream-rhui-rpms   11 k
 mysql-common               x86_64     8.0.32-1.el9_2     rhel-9-appstream-rhui-rpms   79 k

Transaction Summary
```

So, we proceed by writing the command

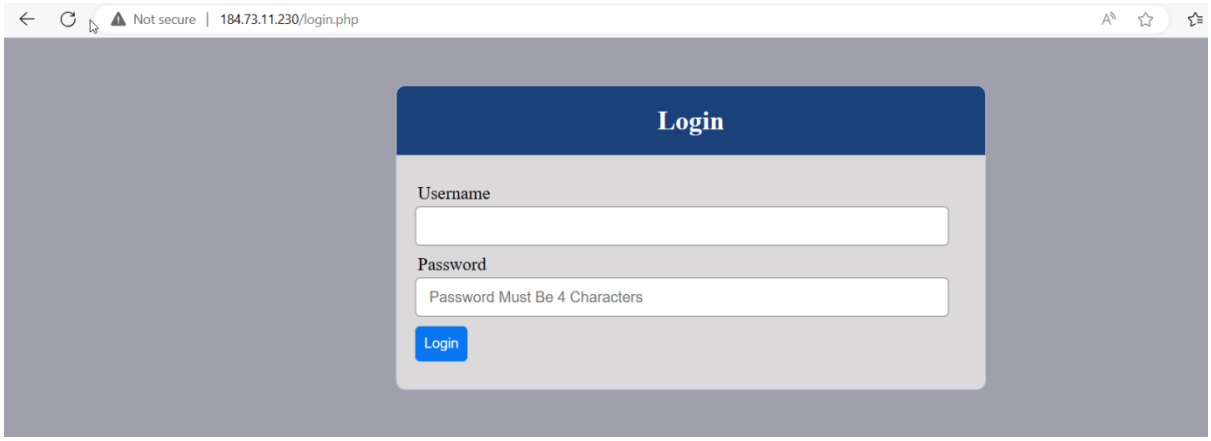
```
[ec2-user@web-server1 tooling]$ mysql -h 172.31.81.224 -u webaccess -p tooling < tooling-db.sql
Enter password:
```

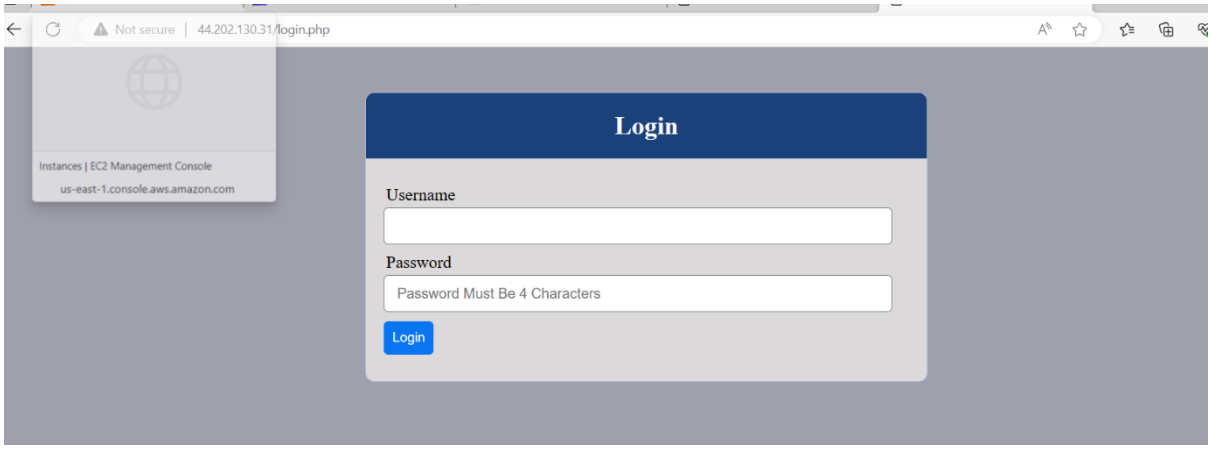
Check blkid to view the id .

```
[ec2-user@web-server1 ~]$ sudo blkid
/dev/xvda4: LABEL="root" UUID="287d9c0b-0e0f-4e92-8534-45733aa3dc68" TYPE="xfs" PARTUUID="6264d520-3fb9
-423f-8ab8-7a0a8e3d3562"
/dev/xvda2: SEC_TYPE="msdos" UUID="7B77-95E7" TYPE="vfat" PARTUUID="68b2905b-df3e-4fb3-80fa-49d1e773aa3
3"
/dev/xvda3: LABEL="boot" UUID="7bc24af7-289d-4bce-b17e-300c3aafe968" TYPE="xfs" PARTUUID="cb07c243-bc44
-4717-853e-28852021225b"
/dev/xvda1: PARTUUID="fac7f1fb-3e8d-4137-a512-961de09a5549"
[ec2-user@web-server1 ~]$ sudo vi /etc/fstab
[ec2-user@web-server1 tooling]$ mysql -h 172.31.81.224 -u webaccess -p tooling < tooling-db.sql
Enter password:
```

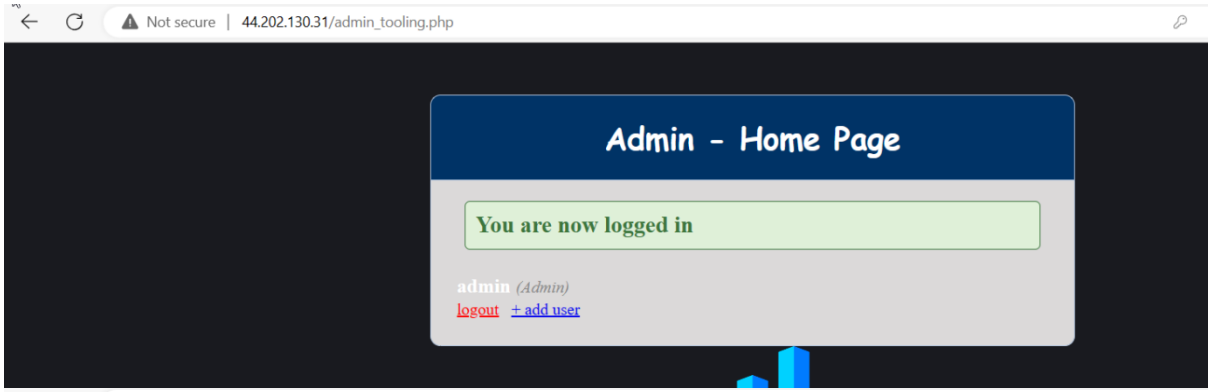
Then we launch out ip addresses for the 2 server and it is

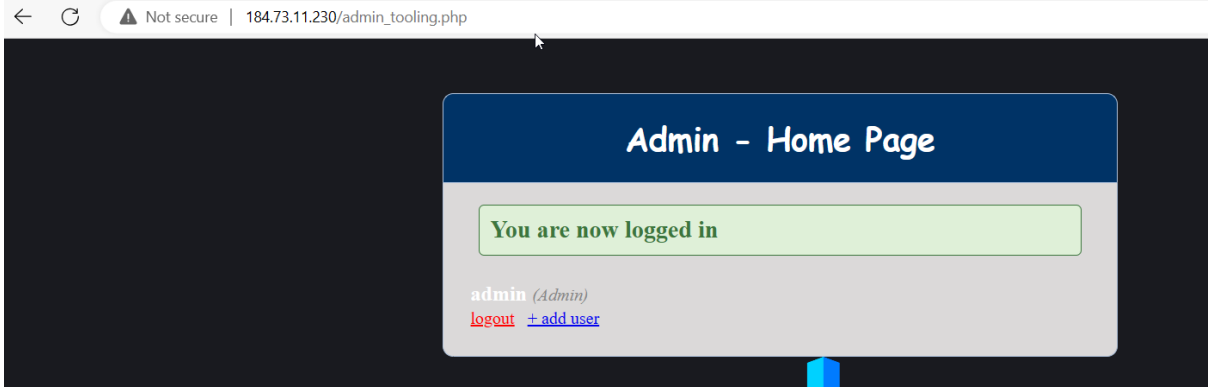
successfully displayed





We would go further to type in the details for the username and password and they both are successfully logged





Finally, we have successfully implemented a web solution using LAMP stack with remote databases and NFS Servers.