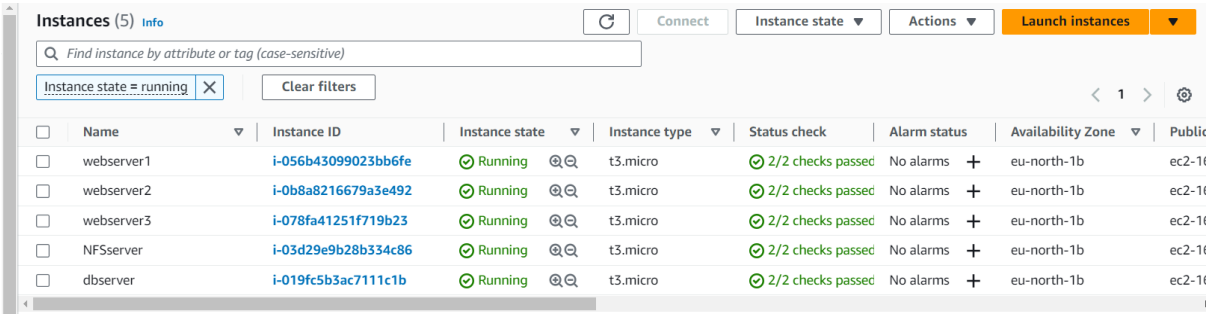
PROJECT 7

Step1:

- Spin up servers



Instances (5) Info

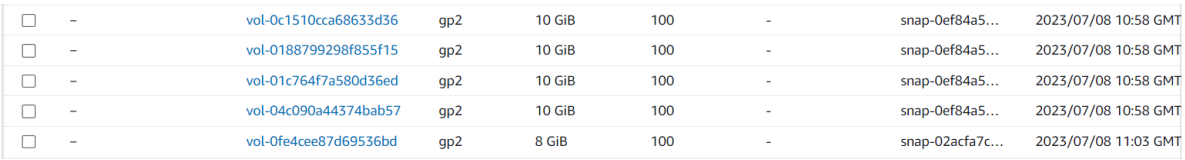
Connect | Instance state | Actions | Launch instances

Find instance by attribute or tag (case-sensitive)

Instance state = running | Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public
webserver1	i-056b43099023bb6fe	Running	t3.micro	2/2 checks passed	No alarms	eu-north-1b	ec2-1
webserver2	i-0b8a8216679a3e492	Running	t3.micro	2/2 checks passed	No alarms	eu-north-1b	ec2-1
webserver3	i-078fa41251f719b23	Running	t3.micro	2/2 checks passed	No alarms	eu-north-1b	ec2-1
NFSserver	i-03d29e9b28b334c86	Running	t3.micro	2/2 checks passed	No alarms	eu-north-1b	ec2-1
dbserver	i-019fc5b3ac7111c1b	Running	t3.micro	2/2 checks passed	No alarms	eu-north-1b	ec2-1

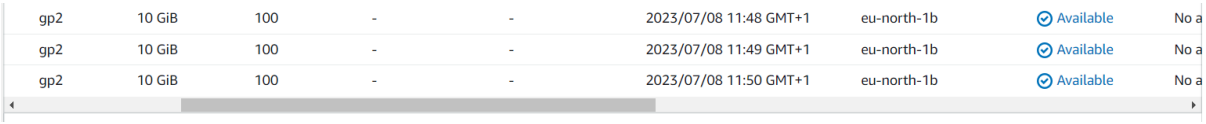
- Configure LVM on the servers



–	vol-0c1510cca68633d36	gp2	10 GiB	100	-	snap-0ef84a5...	2023/07/08 10:58 GMT
–	vol-0188799298f855f15	gp2	10 GiB	100	-	snap-0ef84a5...	2023/07/08 10:58 GMT
–	vol-01c764f7a580d36ed	gp2	10 GiB	100	-	snap-0ef84a5...	2023/07/08 10:58 GMT
–	vol-04c090a44374bab57	gp2	10 GiB	100	-	snap-0ef84a5...	2023/07/08 10:58 GMT
–	vol-0fe4cee87d69536bd	gp2	8 GiB	100	-	snap-02acfa7c...	2023/07/08 11:03 GMT

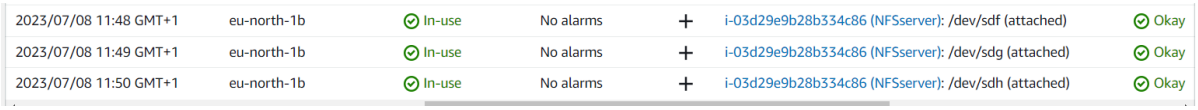
This is the created EBS Volumes automatically when I created the EC2 Server instances.

We will create volumes:
NB: We will create 3 logical volumes.



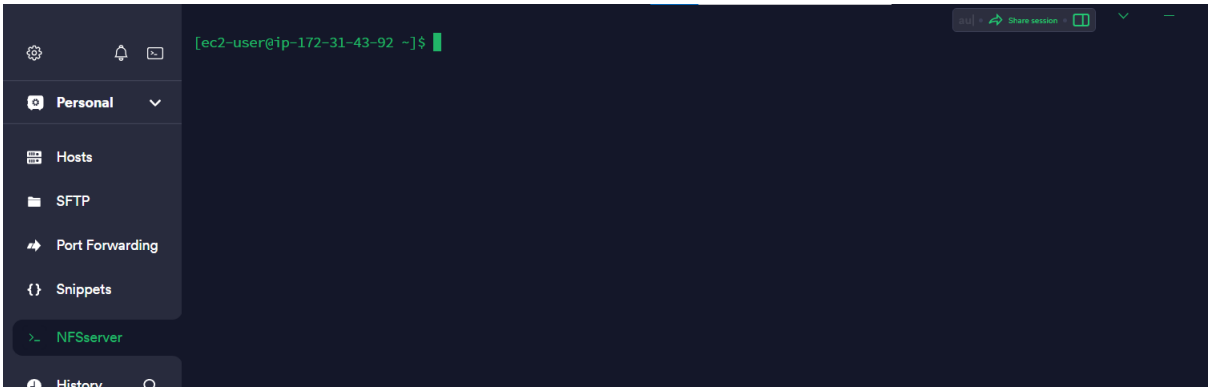
gp2	10 GiB	100	-	-	2023/07/08 11:48 GMT+1	eu-north-1b	Available	No a
gp2	10 GiB	100	-	-	2023/07/08 11:49 GMT+1	eu-north-1b	Available	No a
gp2	10 GiB	100	-	-	2023/07/08 11:50 GMT+1	eu-north-1b	Available	No a

Then attach them to the NFS Server.

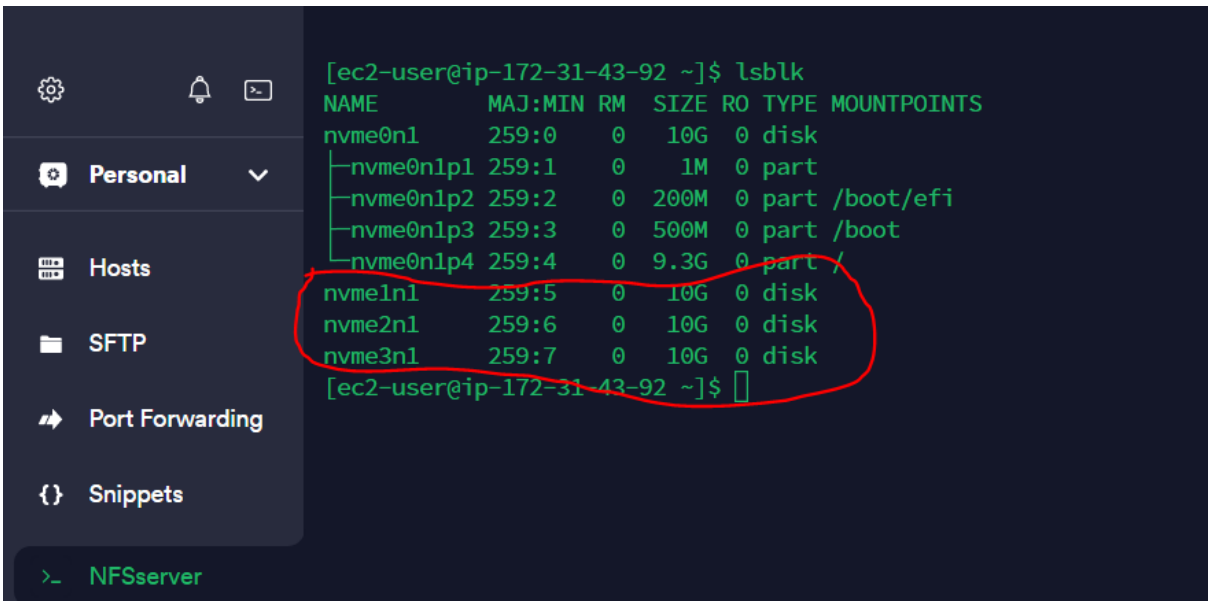


2023/07/08 11:48 GMT+1	eu-north-1b	In-use	No alarms	+	i-03d29e9b28b334c86 (NFSserver): /dev/sdf (attached)	Okay
2023/07/08 11:49 GMT+1	eu-north-1b	In-use	No alarms	+	i-03d29e9b28b334c86 (NFSserver): /dev/sdg (attached)	Okay
2023/07/08 11:50 GMT+1	eu-north-1b	In-use	No alarms	+	i-03d29e9b28b334c86 (NFSserver): /dev/sdh (attached)	Okay

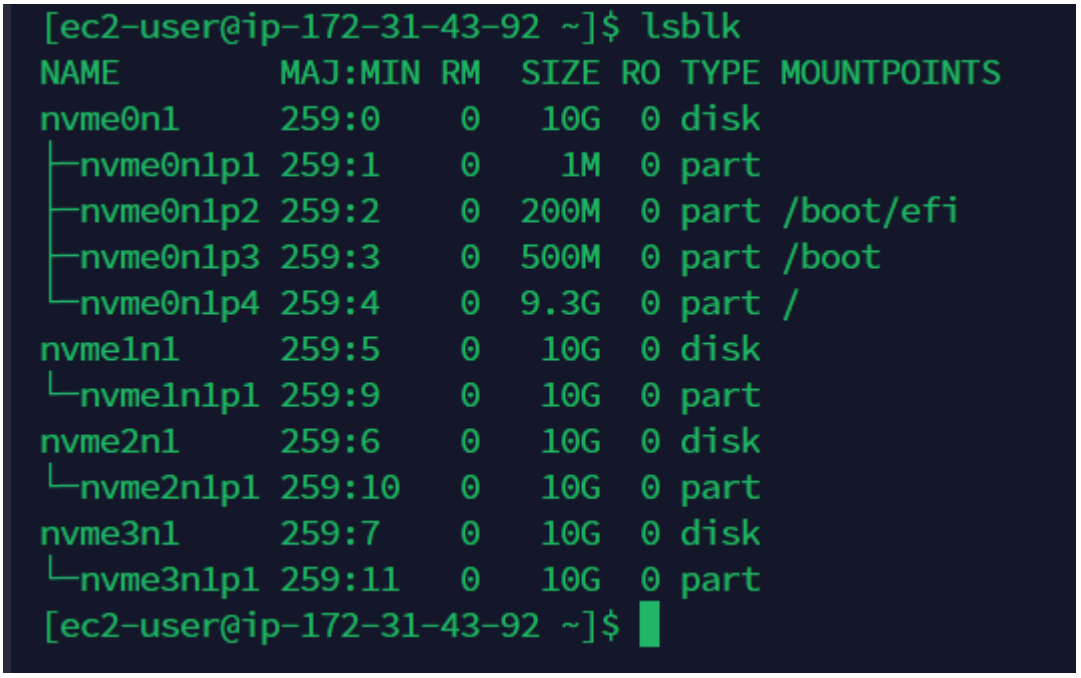
Then connect to the NFSserver:



Then we need to list all the block devices that are attached to the NFSserver



Then use gdisk utility to create 1 partition in each of the attached disks



The next thing is to install the LVM Package
[ec2-user@ip-172-31-43-92 ~]$ sudo yum install lvm2 -y

We should check for available partition:

```
[ec2-user@ip-172-31-43-92 ~]$ sudo lvmdiskscan
  /dev/nvme0n1p2 [     200.00 MiB]
  /dev/nvme0n1p3 [     500.00 MiB]
  /dev/nvme0n1p4 [       9.31 GiB]
  /dev/nvme1n1p1 [     <10.00 GiB]
  /dev/nvme2n1p1 [     <10.00 GiB]
  /dev/nvme3n1p1 [     <10.00 GiB]
  0 disks
  6 partitions
  0 LVM physical volume whole disks
  0 LVM physical volumes
[ec2-user@ip-172-31-43-92 ~]$
[ec2-user@ip-172-31-43-92 ~]$
[ec2-user@ip-172-31-43-92 ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
nvme0n1     259:0    0   10G  0 disk
├─nvme0n1p1 259:1    0    1M  0 part
├─nvme0n1p2 259:2    0  200M  0 part /boot/efi
├─nvme0n1p3 259:3    0  500M  0 part /boot
└─nvme0n1p4 259:4    0  9.3G  0 part /
nvme1n1     259:5    0   10G  0 disk
└─nvme1n1p1 259:9    0   10G  0 part
nvme2n1     259:6    0   10G  0 disk
└─nvme2n1p1 259:10   0   10G  0 part
nvme3n1     259:7    0   10G  0 disk
└─nvme3n1p1 259:11   0   10G  0 part
[ec2-user@ip-172-31-43-92 ~]$
```

We need to create physical volume to be used by lvm using pvcreate utility:

```
[ec2-user@ip-172-31-43-92 ~]$ sudo pvcreate /dev/nvme1n1p1
  Physical volume "/dev/nvme1n1p1" successfully created.
  Creating devices file /etc/lvm/devices/system.devices
[ec2-user@ip-172-31-43-92 ~]$ sudo pvcreate /dev/nvme2n1p1
  Physical volume "/dev/nvme2n1p1" successfully created.
[ec2-user@ip-172-31-43-92 ~]$ sudo pvcreate /dev/nvme3n1p1
  Physical volume "/dev/nvme3n1p1" successfully created.
[ec2-user@ip-172-31-43-92 ~]$
```

Then check if the physical volume are there:

```
[ec2-user@ip-172-31-43-92 ~]$ sudo pvs
  PV             VG Fmt  Attr PSize   PFree
  /dev/nvme1n1p1    lvm2 ---  <10.00g <10.00g
  /dev/nvme2n1p1    lvm2 ---  <10.00g <10.00g
  /dev/nvme3n1p1    lvm2 ---  <10.00g <10.00g
[ec2-user@ip-172-31-43-92 ~]$
```

Then we are going to use vgcreate utility to create volume group:

```
[ec2-user@ip-172-31-43-92 ~]$ sudo pvs
  PV             VG Fmt  Attr PSize   PFree
  /dev/nvme1n1p1    lvm2 ---  <10.00g <10.00g
  /dev/nvme2n1p1    lvm2 ---  <10.00g <10.00g
  /dev/nvme3n1p1    lvm2 ---  <10.00g <10.00g
[ec2-user@ip-172-31-43-92 ~]$ sudo vgcreate /dev/nvme3n1p1 /dev/nvme2n1p1 /dev/nvme1n1p1
  /dev/nvme3n1p1: already exists in filesystem
  Run `vgcreate --help' for more information.
[ec2-user@ip-172-31-43-92 ~]$ sudo vgcreate webdata-vg /dev/nvme3n1p1 /dev/nvme2n1p1 /dev/nvme1n1p1
  Volume group "webdata-vg" successfully created
[ec2-user@ip-172-31-43-92 ~]$
```

Check the volume group was created and the size of the disk sum up:

```
[ec2-user@ip-172-31-43-92 ~]$ sudo vgs
  VG         #PV #LV #SN Attr   VSize   VFree
  webdata-vg   3   0   0 wz--n- <29.99g <29.99g
[ec2-user@ip-172-31-43-92 ~]$
```

NB: Its less than 30G bcos the file system has used up some space already.

Then we will need to create logical volumes; lv-opt, lv-apps and lv-logs

```
[ec2-user@ip-172-31-43-92 ~]$ sudo lvcreate -n lv-apps -L 9G webdata-vg
  Logical volume "lv-apps" created.
[ec2-user@ip-172-31-43-92 ~]$ sudo lvcreate -n lv-logs -L 9G webdata-vg
  Logical volume "lv-logs" created.
[ec2-user@ip-172-31-43-92 ~]$ sudo lvcreate -n lv-opt -L 9G webdata-vg
  Logical volume "lv-opt" created.
[ec2-user@ip-172-31-43-92 ~]$
```

Check if they are created and their size

```
[ec2-user@ip-172-31-43-92 ~]$ sudo lvs
  LV      VG         Attr       LSize Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  lv-apps webdata-vg -wi-a----- 9.00g
  lv-logs webdata-vg -wi-a----- 9.00g
  lv-opt  webdata-vg -wi-a----- 9.00g
[ec2-user@ip-172-31-43-92 ~]$
```

```
[ec2-user@ip-172-31-43-92 ~]$ lsblk
NAME                     MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
nvme0n1                  259:0    0   10G  0 disk
├─nvme0n1p1              259:1    0    1M  0 part
├─nvme0n1p2              259:2    0  200M  0 part /boot/efi
├─nvme0n1p3              259:3    0  500M  0 part /boot
└─nvme0n1p4              259:4    0  9.3G  0 part /
nvme1n1                  259:5    0   10G  0 disk
└─nvme1n1p1              259:9    0   10G  0 part
  └─webdata--vg-lv--opt  253:2    0    9G  0 lvm
nvme2n1                  259:6    0   10G  0 disk
└─nvme2n1p1              259:10   0   10G  0 part
  └─webdata--vg-lv--logs 253:1    0    9G  0 lvm
nvme3n1                  259:7    0   10G  0 disk
└─nvme3n1p1              259:11   0   10G  0 part
  └─webdata--vg-lv--apps 253:0    0    9G  0 lvm
[ec2-user@ip-172-31-43-92 ~]$
```

Next step is to format the disks as xfs and not ext4;

```
[ec2-user@ip-172-31-43-92 ~]$ sudo mkfs -t xfs /dev/webdata-vg/lv-apps
meta-data=/dev/webdata-vg/lv-apps isize=512    agcount=16, agsize=147456 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=1      swidth=1 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=1 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
[ec2-user@ip-172-31-43-92 ~]$ sudo mkfs -t xfs /dev/webdata-vg/lv-logs
meta-data=/dev/webdata-vg/lv-logs isize=512    agcount=16, agsize=147456 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=1      swidth=1 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=1 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
[ec2-user@ip-172-31-43-92 ~]$ sudo mkfs -t xfs /dev/webdata-vg/lv-opt
meta-data=/dev/webdata-vg/lv-opt isize=512    agcount=16, agsize=147456 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=1      swidth=1 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=1 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
[ec2-user@ip-172-31-43-92 ~]$
```

We need to create the mount point for the three logical volumes; apps, logs and opt;

```
[ec2-user@ip-172-31-43-92 ~]$ sudo mkdir /mnt/apps
[ec2-user@ip-172-31-43-92 ~]$ sudo mkdir /mnt/logs
[ec2-user@ip-172-31-43-92 ~]$ sudo mkdir /mnt/opt
[ec2-user@ip-172-31-43-92 ~]$
```

Carry out the mounting now;

```
[ec2-user@ip-172-31-43-92 ~]$ sudo mount /dev/webdata-vg/lv=apps /mnt/apps
mount: /mnt/apps: special device /dev/webdata-vg/lv=apps does not exist.
[ec2-user@ip-172-31-43-92 ~]$ sudo mount /dev/webdata-vg/lv-apps /mnt/apps
[ec2-user@ip-172-31-43-92 ~]$ ^C
[ec2-user@ip-172-31-43-92 ~]$ sudo mount /dev/webdata-vg/lv-logs /mnt/logs
[ec2-user@ip-172-31-43-92 ~]$ sudo mount /dev/webdata-vg/lv-opt /mnt/opt
[ec2-user@ip-172-31-43-92 ~]$
```

The next step is to install 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
```

```
[ec2-user@ip-172-31-43-92 ~]$ sudo systemctl start nfs-server.service
[ec2-user@ip-172-31-43-92 ~]$ sudo systemctl enable nfs-server.service
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-server.service → /usr/lib/systemd/system/nfs-server.service.
[ec2-user@ip-172-31-43-92 ~]$ sudo systemctl status 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 Sat 2023-07-08 17:37:54 UTC; 50s ago
   Main PID: 53042 (code=exited, status=0/SUCCESS)
        CPU: 33ms

Jul 08 17:37:54 ip-172-31-43-92.eu-north-1.compute.internal systemd[1]: Starting NFS server and services...
Jul 08 17:37:54 ip-172-31-43-92.eu-north-1.compute.internal systemd[1]: Finished NFS server and services.
[ec2-user@ip-172-31-43-92 ~]$
```

The nfs server is active and running now.

- Make sure we set up permission that will allow our Web servers to read, write and execute files on NFS:

```
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
```

```
sudo systemctl restart nfs-server.service
```

- Configure access to NFS for clients within the same subnet (example of Subnet CIDR – `172.31.32.0/20`:

```
sudo vi /etc/exports

Paste:
/mnt/apps <Subnet-CIDR>(rw,sync,no_all_squash,no_root_squash)
/mnt/logs <Subnet-CIDR>(rw,sync,no_all_squash,no_root_squash)
/mnt/opt <Subnet-CIDR>(rw,sync,no_all_squash,no_root_squash)
```

- Export it so that the webservers will see it when they try to connect;

```
sudo exportfs -arv
```

```
[ec2-user@ip-172-31-43-92 ~]$ sudo exportfs -arv
exporting 172.31.32.0/20:/mnt/opt
exporting 172.31.32.0/20:/mnt/logs
exporting 172.31.32.0/20:/mnt/apps
[ec2-user@ip-172-31-43-92 ~]$
```

As can be seen, it has exported all of these

- Check which port is used by NFS and open it using Security Groups (add new Inbound Rule)

```
rpcinfo -p | grep nfs
```

```
[ec2-user@ip-172-31-43-92 ~]$ rpcinfo -p | grep nfs
    100003    3   tcp   2049  nfs
    100003    4   tcp   2049  nfs
    100227    3   tcp   2049  nfs_acl
[ec2-user@ip-172-31-43-92 ~]$
```

Step2:

· Install mysql server

```
ubuntu@ip-172-31-34-206:~$ which mysql
/usr/bin/mysql
ubuntu@ip-172-31-34-206:~$
```

· Log into mysql and create a database called tooling

```
ubuntu@ip-172-31-34-206:~$ which mysql
/usr/bin/mysql
ubuntu@ip-172-31-34-206:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.33-0ubuntu0.20.04.2 (Ubuntu)

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

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database toolin;
Query OK, 1 row affected (0.01 sec)

mysql>
```

· Create a database user and name it webaccess

```
mysql> create user 'webaccess'@'172.31.32.0/20' identified by 'password';
Query OK, 0 rows affected (0.02 sec)

mysql>
```

· Grant permission to webaccess user on database tooling;

```
mysql> grant all privileges on tooling.* to 'webaccess'@'172.31.32.0/20';
Query OK, 0 rows affected (0.01 sec)

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

Step3: Prepare the Web Servers

We are to mount the logical volumes created on the webservers now.

- Install NFS client

Now we gonna install nfs client without which we can not access the nfs server at all.

```
sudo yum install nfs-utils nfs4-acl-tools -y
```

- Mount `/var/www/` and target the NFS server's export for apps

```
sudo mkdir /var/www
sudo mount -t nfs -o rw,nosuid
<NFS-Server-Private-IP-Address>:/mnt/apps /var/www
```

NB: This /mnt/apps is located in the nfs serve while the /var;www is located locally on the webserver1

```
[ec2-user@ip-172-31-32-224 ~]$ sudo mount -t nfs -o rw,nosuid 172.31.43.92:/mnt/apps /var/www
[ec2-user@ip-172-31-32-224 ~]$ df -h
Filesystem              Size  Used Avail Use% Mounted on
devtmpfs                4.0M     0  4.0M   0% /dev
tmpfs                   372M     0  372M   0% /dev/shm
tmpfs                   149M  8.8M  140M   6% /run
/dev/nvme0n1p4          9.4G  1.5G  7.9G  16% /
/dev/nvme0n1p3          495M  266M  229M  54% /boot
/dev/nvme0n1p2          200M  8.0K  200M   1% /boot/efi
tmpfs                    75M     0   75M   0% /run/user/1000
172.31.43.92:/mnt/apps  9.0G   98M  8.9G   2% /var/www
[ec2-user@ip-172-31-32-224 ~]$
```

- Verify that NFS was mounted successfully by running `df -h` Make sure that the changes will persist on Web Server after reboot:

```
sudo vi /etc/fstab
```

add following line

```
<NFS-Server-Private-IP-Address>:/mnt/apps /var/www nfs defaults 0 0
```

- Next is to install Apache

```
sudo yum install httpd -y
```

NB: Without Apache, this webserver is not webserver bcos it will not be able to serve content to the end users.

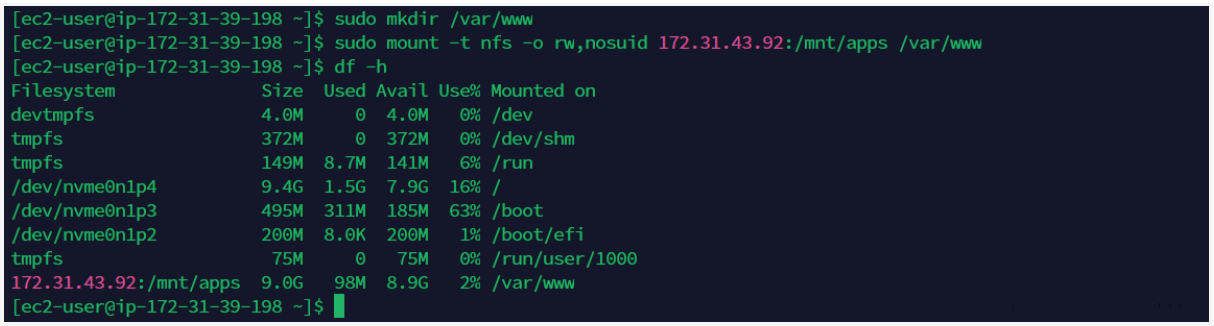
Webserver2 configuration

- Launch the ec2 webserver2
- Install NFS Client

```
sudo yum install nfs-utils nfs4-acl-tools -y
```

- Mount `/var/www/` and target the NFS server's export for apps

```
sudo mkdir /var/www
sudo mount -t nfs -o rw,nosuid
<NFS-Server-Private-IP-Address>:/mnt/apps /var/www
```



```
[ec2-user@ip-172-31-39-198 ~]$ sudo mkdir /var/www
[ec2-user@ip-172-31-39-198 ~]$ sudo mount -t nfs -o rw,nosuid 172.31.43.92:/mnt/apps /var/www
[ec2-user@ip-172-31-39-198 ~]$ df -h
Filesystem              Size  Used Avail Use% Mounted on
devtmpfs                4.0M     0  4.0M   0% /dev
tmpfs                   372M     0  372M   0% /dev/shm
tmpfs                   149M  8.7M  141M   6% /run
/dev/nvme0n1p4          9.4G  1.5G  7.9G  16% /
/dev/nvme0n1p3          495M  311M  185M  63% /boot
/dev/nvme0n1p2          200M  8.0K  200M   1% /boot/efi
tmpfs                    75M     0   75M   0% /run/user/1000
172.31.43.92:/mnt/apps  9.0G   98M  8.9G   2% /var/www
[ec2-user@ip-172-31-39-198 ~]$
```

```
sudo vi /etc/fstab
```

add following line

```
<NFS-Server-Private-IP-Address>:/mnt/apps /var/www nfs defaults 0 0
```

- Install Remi's repository, Apache and PHP

```
sudo yum install httpd -y
```

Webserver3 configuration

Repeat same like webserver1

```
[ec2-user@ip-172-31-38-213 ~]$ sudo mkdir /var/www
[ec2-user@ip-172-31-38-213 ~]$ sudo mount -t nfs -o rw,nosuid 172.31.43.92:/mnt/apps /var/www
[ec2-user@ip-172-31-38-213 ~]$ df -h
Filesystem              Size  Used Avail Use% Mounted on
devtmpfs                4.0M     0  4.0M   0% /dev
tmpfs                   372M     0  372M   0% /dev/shm
tmpfs                   149M  7.3M  142M   5% /run
/dev/nvme0n1p4          9.4G  1.5G  7.9G  16% /
/dev/nvme0n1p3          495M  266M  229M  54% /boot
/dev/nvme0n1p2          200M  8.0K  200M   1% /boot/efi
tmpfs                    75M     0   75M   0% /run/user/1000
172.31.43.92:/mnt/apps  9.0G   98M  8.9G   2% /var/www
[ec2-user@ip-172-31-38-213 ~]$
```

7. Locate the log folder for Apache on the Web Server and mount it to NFS server's export for logs. Repeat step №4 to make sure the mount point will persist after reboot.

```
sudo mount -t nfs -o rw,nosuid
<NFS-Server-Private-IP-Address>:/mnt/logs /var/log/httpd
```

```
[ec2-user@ip-172-31-32-224 ~]$ sudo mount -t nfs -o rw,nosuid 172.31.43.92:/mnt/logs /var/log/httpd
[ec2-user@ip-172-31-32-224 ~]$
```

```
sudo vi /etc/fstab
```

add following line

```
<NFS-Server-Private-IP-Address>:/mnt/logs /var/log/httpd nfs defaults 0
0
```

8. Fork the tooling source code from Darey.io Github Account to your Github account. (Learn how to fork a repo here)

https://github.com/darey-io/tooling.git

```
[ec2-user@ip-172-31-32-224 ~]$ git clone https://github.com/darey-io/tooling.git
Cloning into 'tooling'...
remote: Enumerating objects: 243, done.
remote: Total 243 (delta 0), reused 0 (delta 0), pack-reused 243
Receiving objects: 100% (243/243), 283.48 KiB | 2.58 MiB/s, done.
Resolving deltas: 100% (137/137), done.
[ec2-user@ip-172-31-32-224 ~]$
```

```
[ec2-user@ip-172-31-32-224 ~]$ ls
tooling
[ec2-user@ip-172-31-32-224 ~]$
```

```
[ec2-user@ip-172-31-32-224 ~]$ cd tooling
[ec2-user@ip-172-31-32-224 tooling]$ ls
apache-config.conf  Dockerfile  html  Jenkinsfile  README.md  start-apache  tooling-db.sql
[ec2-user@ip-172-31-32-224 tooling]$
```

9. Deploy the tooling website's code to the Webserver. Ensure that the html folder from the repository is deployed to `/var/www/html`

```
sudo cp -r html/. /var/www/html
```

```
[ec2-user@ip-172-31-32-224 tooling]$ ls /var/www/html
admin_tooling.php  functions.php  index.php  README.md     style.css
create_user.php    img            login.php  register.php  tooling_stylesheets.css
[ec2-user@ip-172-31-32-224 tooling]$
```

Do not forget to open TCP port 80 on the Web Server.

If you encounter 403 Error – check permissions to your

`/var/www/html`
folder and also disable SELinux

```
sudo setenforce 0
```

To make this change permanent – open following config file

```
sudo vi /etc/sysconfig/selinux
```
and set

```
SELINUX=disabled
```
then restrt httpd.

10. Update the website's configuration to connect to the database (in `/var/www/html/functions.php` file). Apply `Tooling-db.sql` script to your database using this command `mysql -h <databse-private-ip> -u <db-username> -p <db-pasword> < tooling-db.sql`

Sudo vi `/var/www/html/functions.php`

NB: Edit the circled

```
<?php
session_start();

// connect to database
$db = mysqli_connect('mysql.tooling.svc.cluster.local', 'admin', 'admin', 'tooling');

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

// variable declaration
$username = "";
$email    = "";
$errors   = array();

// call the register() function if register_btn is clicked
if (isset($_POST['register_btn'])) {
        register();
}

// REGISTER USER
function register(){
        // call these variables with the global keyword to make them available in function
        global $db, $errors, $username, $email;

        // receive all input values from the form. Call the e() function
    // defined below to escape form values
        $username    =  e($_POST['username']);
        $email       =  e($_POST['email']);
        $password_1  =  e($_POST['password_1']);
        $password_2  =  e($_POST['password_2']);

        // form validation: ensure that the form is correctly filled
        if (empty($username)) {
                array_push($errors, "Username is required");
"/var/www/html/functions.php" 178L, 4385B
```

Edited copy

```php
<?php
session_start();

// connect to database
$db = mysqli_connect('172.31.34.206', 'webaccess', 'password', 'tooling');

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

// variable declaration
$username = "";
$email    = "";
$errors   = array();

// call the register() function if register_btn is clicked
if (isset($_POST['register_btn'])) {
        register();
}

// REGISTER USER
function register(){
        // call these variables with the global keyword to make them available in function
        global $db, $errors, $username, $email;

        // receive all input values from the form. Call the e() function
    // defined below to escape form values
        $username    =  e($_POST['username']);
        $email       =  e($_POST['email']);
        $password_1  =  e($_POST['password_1']);
        $password_2  =  e($_POST['password_2']);

        // form validation: ensure that the form is correctly filled
        if (empty($username)) {
                array_push($errors, "Username is required");
-- INSERT --
```

Sudo yum install mysql

mysql -h 172.31.34.206 -u webaccess -p tooling < tooling-db.sql

Open up mysql port in the dbserver

sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf

```
#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

# Here is entries for some specific programs
# The following values assume you have at least 32M ram

[mysqld]
#
# * Basic Settings
#
user            = mysql
# pid-file      = /var/run/mysqld/mysqld.pid
# socket        = /var/run/mysqld/mysqld.sock
# port          = 3306
# datadir       = /var/lib/mysql


# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
# tmpdir                = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 127.0.0.1
mysqlx-bind-address     = 127.0.0.1
#
# * Fine Tuning
#
key_buffer_size         = 16M
# max_allowed_packet    = 64M
# thread_stack          = 256K

# thread_cache_size       = -1
"/etc/mysql/mysql.conf.d/mysqld.cnf" 78L, 2220C
```

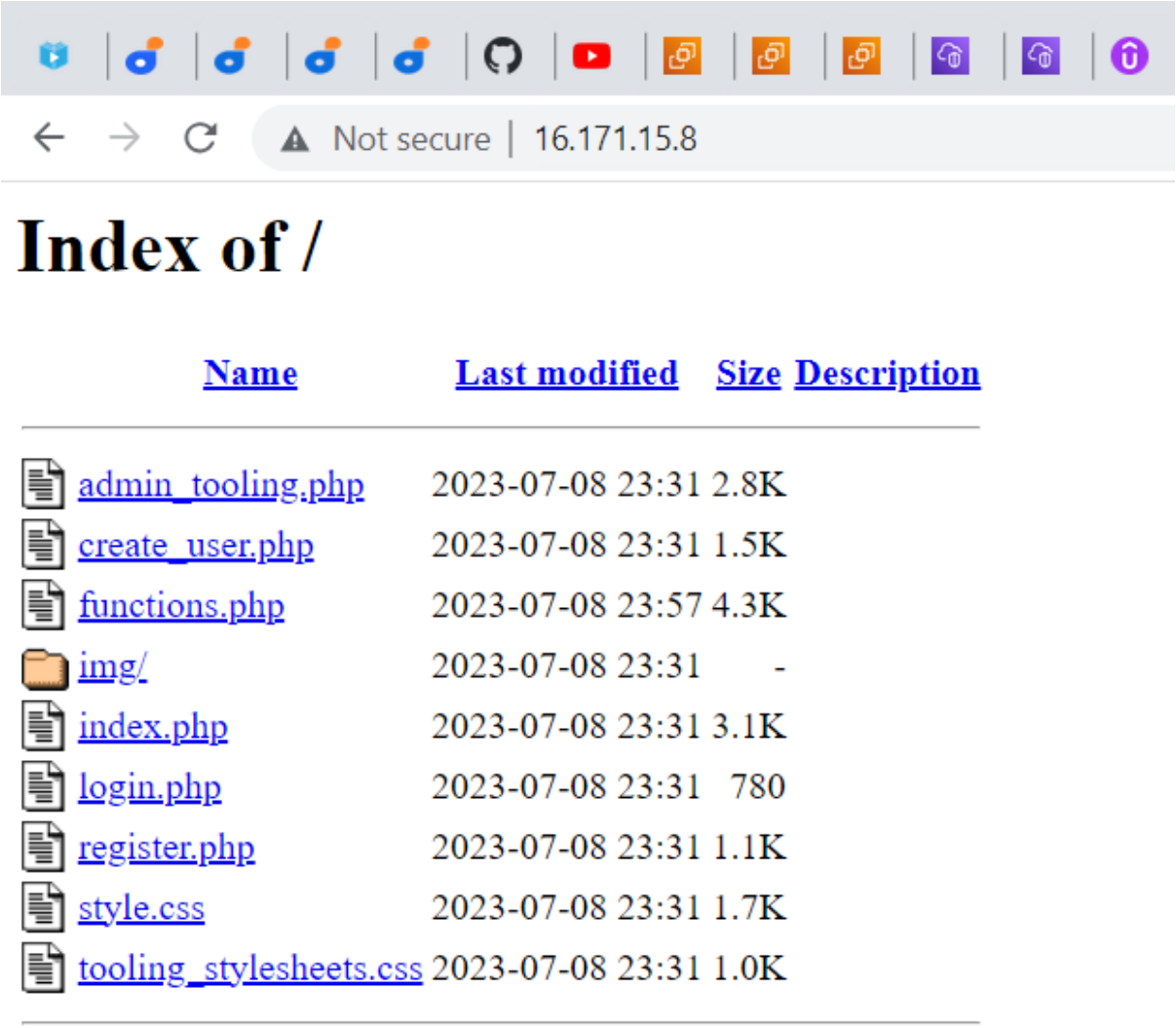
year ago Peer To Peer Live Mentoring Sessions (DevOps Technical Implementations)

So change the two bind address to 0.0.0.0

```
#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

# Here is entries for some specific programs
# The following values assume you have at least 32M ram

[mysqld]
#
# * Basic Settings
#
user            = mysql
# pid-file      = /var/run/mysqld/mysqld.pid
# socket        = /var/run/mysqld/mysqld.sock
# port          = 3306
# datadir       = /var/lib/mysql


# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
# tmpdir                = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 0.0.0.0
mysqlx-bind-address     = 0.0.0.0
#
# * Fine Tuning
#
key_buffer_size         = 16M
# max_allowed_packet    = 64M
# thread_stack          = 256K

# thread_cache_size       = -1
-- INSERT --
```

Sudo systemctl restart httpd

sudo mv /etc/httpd/conf.d/welcome.conf /etc/httpd/conf.d/welcome.backup



Fine this is our website, but we will need to install php dependencies which will help us display all these things:

```
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
```

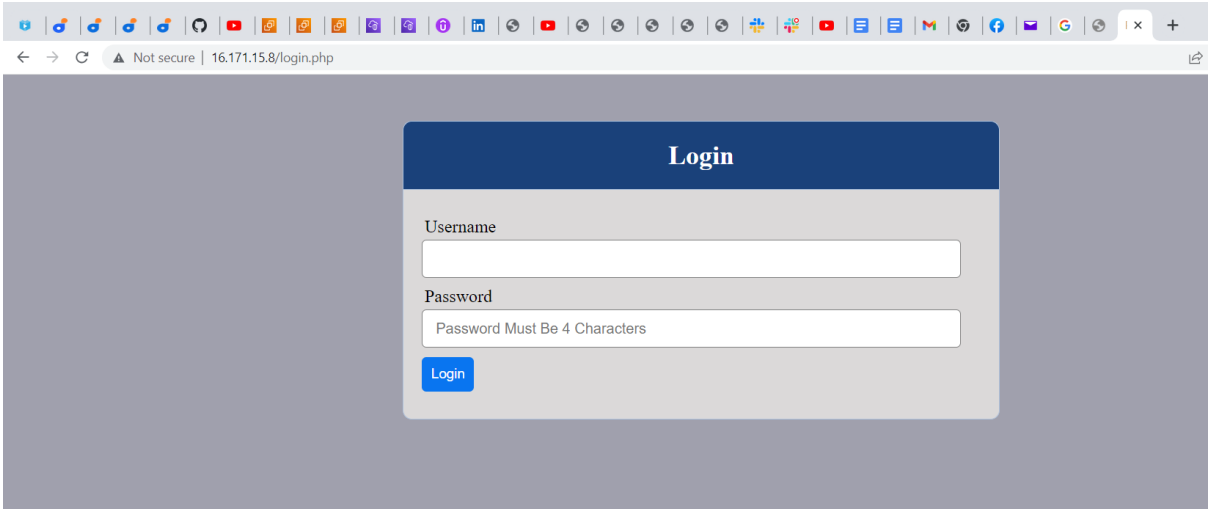
```
sudo dnf module enable php:remi-7.4
```

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

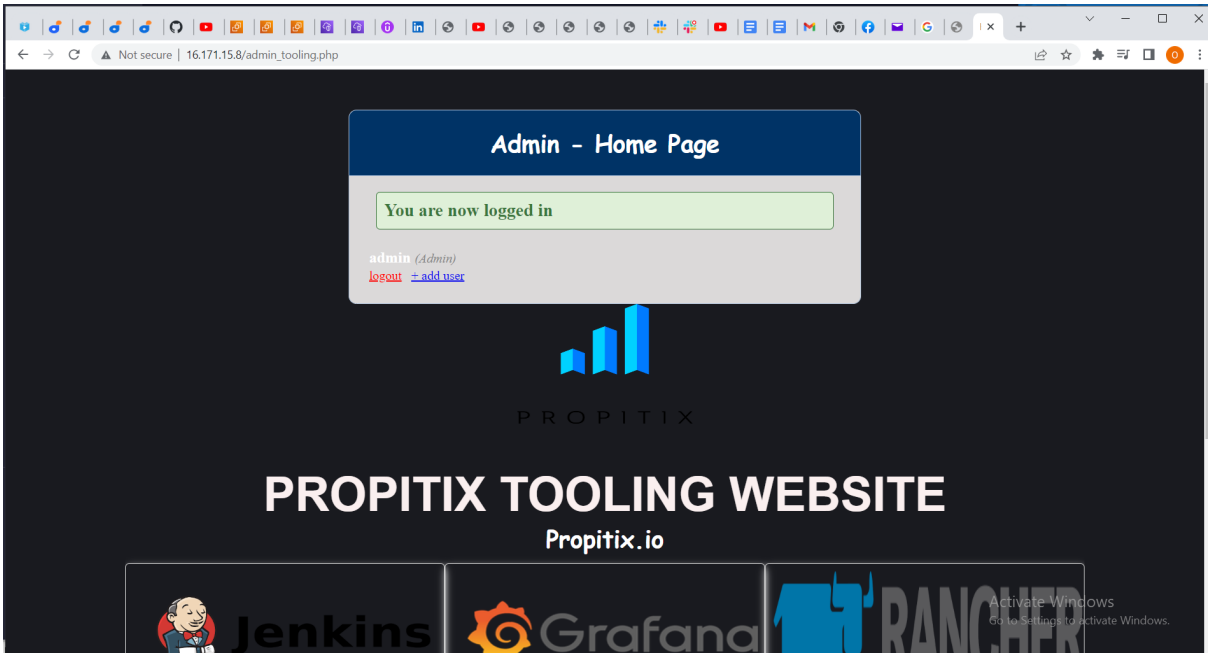
```
sudo systemctl start php-fpm
```

```
sudo systemctl enable php-fpm
```

```
setsebool -P httpd_execmem 1
```



Finished, great job. To God be the glory.


Not secure | 16.171.15.8/admin_tooling.php
Admin - Home Page
You are now logged in
admin (Admin)
logout + add user
PROPITIX
PROPITIX TOOLING WEBSITE
Propitix.io
Jenkins
Grafana
RANCHER
Activate Windows
Go to Settings to activate Windows.