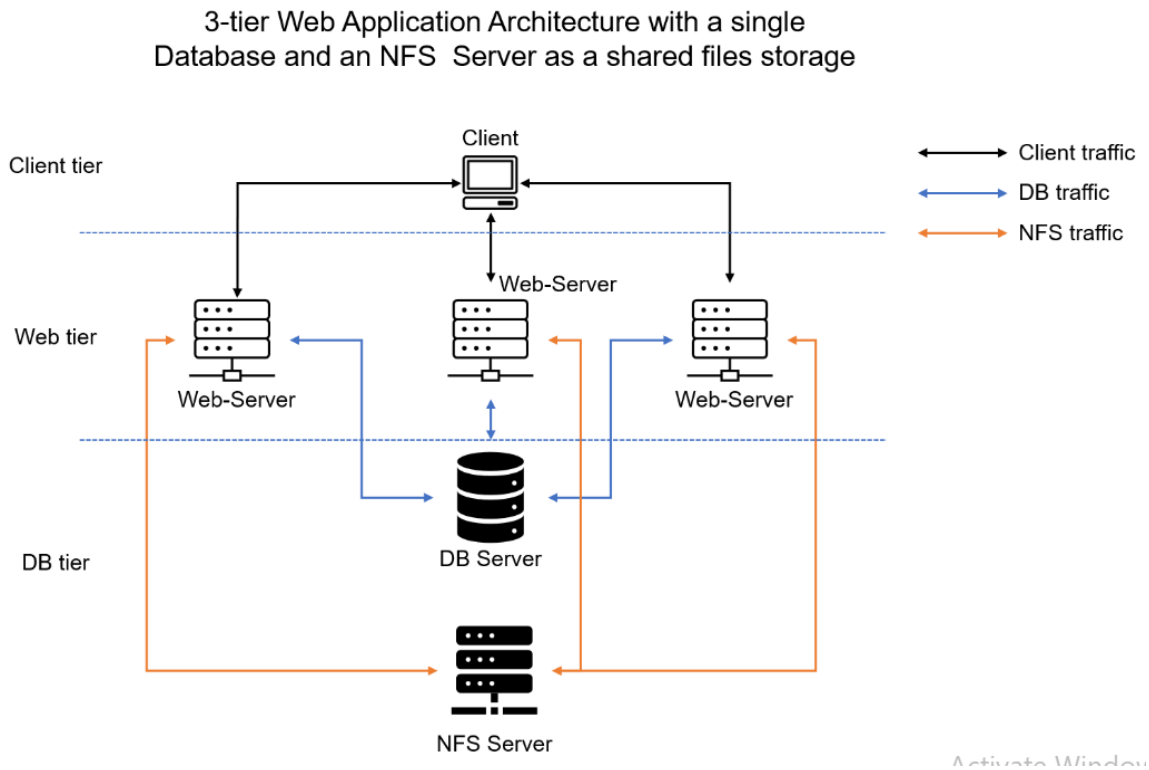
PROJECT -7(Reviewed) : **DEVOPS TOOLING WEBSITE SOLUTION**

In project 6, I implemented a wrdpress based solution ready to serve website/blog content to clients.

In this project 7, file sharing among multiple servers is introduced. Also is the idea of multiple servers sharing one database server.

Below is the network diagram:



This is setup is made up of:

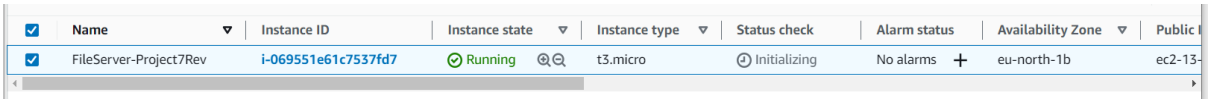
1. Web Server (Rhel 8) x 3
2. Database Sever (ubuntu + mysql)
3. File Server(NFS Server + Rhel 8)

Step 1: **PREPARE NFS SERVER**

- **NFS Configuration:**

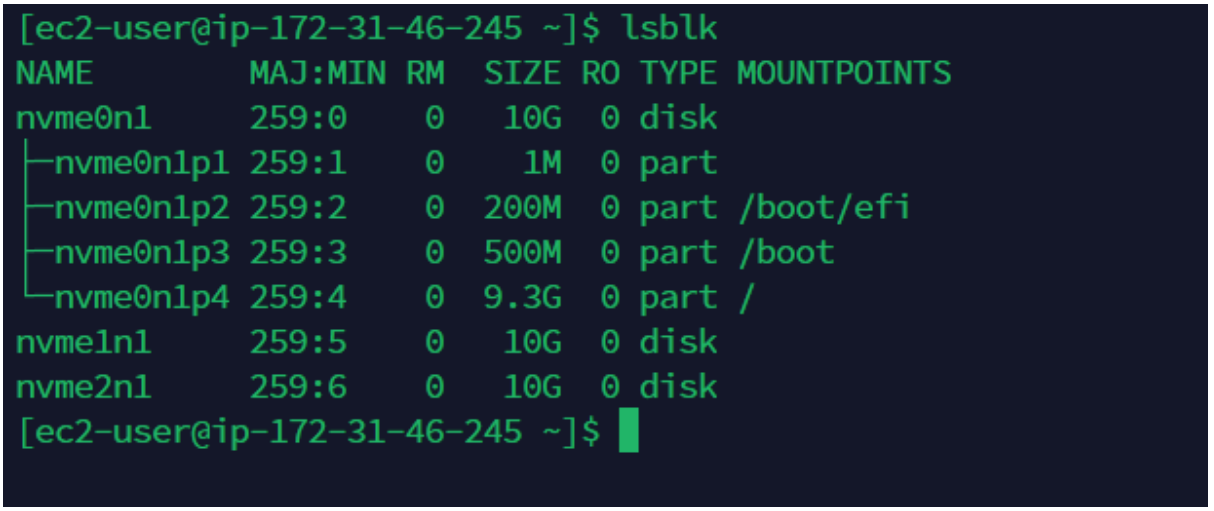
It has two config; the Server side config and the Client side config:

- **Spin up the Server in aws ec2.**

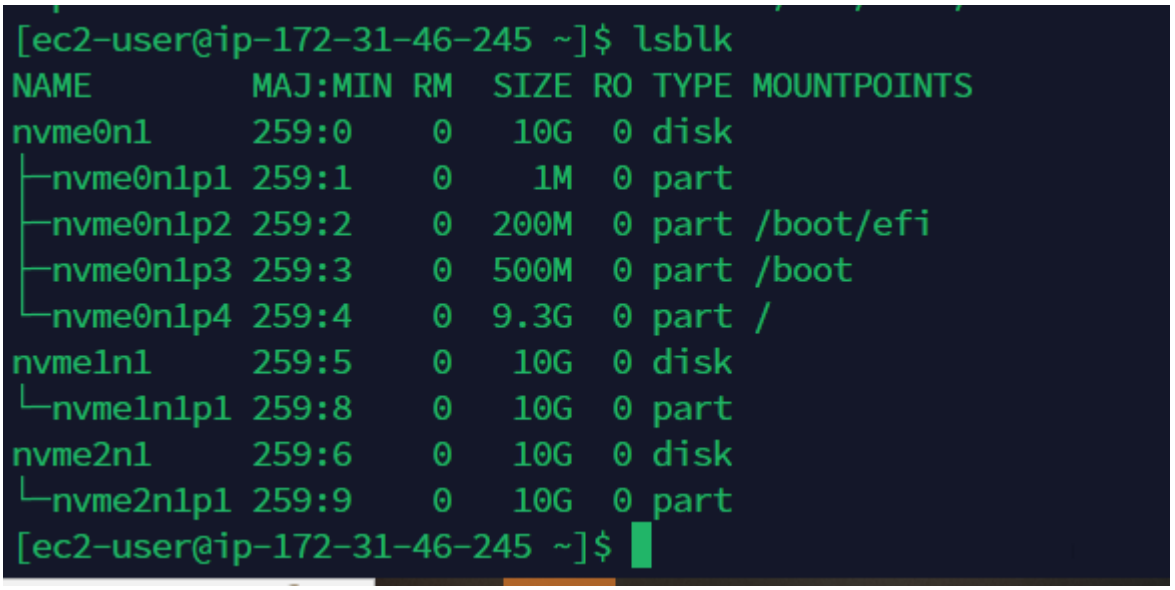


- **Based on the LVM experience I Configure LVM on the File Server.**

I created 2 Volumes of 10G each for the File Server.

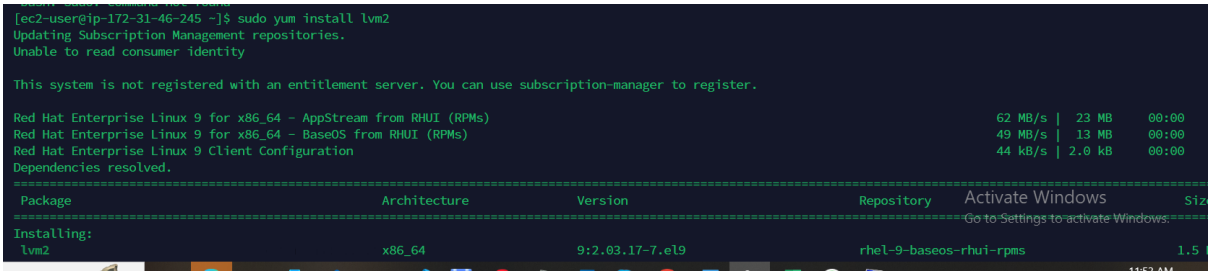


I partitioned each of the disks;



- Install LVM2 package for creating logical volumes on a linux server.

  Sudo yum install lvm2



- Create Physical Volumes on the partitioned disk volumes

  `sudo pvcreate <partition_path>`

```
[ec2-user@ip-172-31-46-245 ~]$ sudo pvcreate /dev/nvme1n1p1 /dev/nvme2n1p1
  Physical volume "/dev/nvme1n1p1" successfully created.
  Physical volume "/dev/nvme2n1p1" successfully created.
  Creating devices file /etc/lvm/devices/system.devices
[ec2-user@ip-172-31-46-245 ~]$
```

- Add up each physical volumes into a volume group called nfs-vg

  `sudo vgcreate <grp_name> <pv_path1> ... <pv_path1000>`

```
[ec2-user@ip-172-31-46-245 ~]$ sudo vgcreate nfs-vg /dev/nvme1n1p1 /dev/nvme2n1p1
  Volume group "nfs-vg" successfully created
[ec2-user@ip-172-31-46-245 ~]$
```

- **I created 3 Logical Volumes- lv-apps, lv-logs and lv-opt.**

`sudo lvcreate -n <lv_name> -L <lv_size> <vg_name>`

```
[ec2-user@ip-172-31-46-245 ~]$ sudo lvcreate -n apps-lv -L 9G nfs-vg
  Logical volume "apps-lv" created.
[ec2-user@ip-172-31-46-245 ~]$ sudo lvcreate -n logs-lv -L 9G nfs-vg
  Logical volume "logs-lv" created.
[ec2-user@ip-172-31-46-245 ~]$
```

- **Verify the entire setup**

```
[ec2-user@ip-172-31-46-245 ~]$ sudo 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         259:8    0   10G  0 part
  └─nfs--vg-apps--lv 253:0    0    9G  0 lvm
nvme2n1             259:6    0   10G  0 disk
└─nvme2n1p1         259:9    0   10G  0 part
  └─nfs--vg-logs--lv 253:1    0    9G  0 lvm
[ec2-user@ip-172-31-46-245 ~]$
```

- **The two logical volumes are ready to be used as filesystems for storing application and log data.**
- **Use `mkfs.ext4` to format the logical volumes with `xfs` filesystem**

```
[ec2-user@ip-172-31-46-245 ~]$ sudo mkfs.xfs /dev/nfs-vg/apps-lv
meta-data=/dev/nfs-vg/apps-lv    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-172-31-46-245 ~]$ sudo mkfs.xfs /dev/nfs-vg/logs-lv
meta-data=/dev/nfs-vg/logs-lv    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-172-31-46-245 ~]$
```

I created mount points for the logical volumes as follows;

```
[ec2-user@ip-172-31-47-23 ~]$ sudo mkdir /mnt/apps
[ec2-user@ip-172-31-47-23 ~]$ sudo mkdir /mnt/logs
[ec2-user@ip-172-31-47-23 ~]$ sudo mkdir /mnt/opt
[ec2-user@ip-172-31-47-23 ~]$
```

`sudo mount /dev/nfs-vg/apps-lv /mnt/apps`

```
[ec2-user@ip-172-31-47-23 ~]$ sudo mount /dev/nfsdata-vg/lv-apps
mount: /dev/nfsdata-vg/lv-apps: can't find in /etc/fstab.
[ec2-user@ip-172-31-47-23 ~]$ sudo mount /dev/nfsdata-vg/lv-apps /mnt/apps
[ec2-user@ip-172-31-47-23 ~]$ sudo mount /dev/nfsdata-vg/lv-logs /mnt/logs
[ec2-user@ip-172-31-47-23 ~]$ sudo mount /dev/nfsdata-vg/lv-opt /mnt/opt
[ec2-user@ip-172-31-47-23 ~]$
```

lv-apps mount on /mnt/apps to be used by the webservers, lv=logs mount on /mnt/logs to be used by webserver logs and finally mount lv-opt on /mnt/opt to be used by jenkins server in the project 8.

- **Install NFS server, configure it to start on reboot and make sure it is running after reboot.**

```
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-172-31-47-23 ~]$ sudo systemctl start nfs-server.service
[ec2-user@ip-172-31-47-23 ~]$ 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-172-31-47-23 ~]$ 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 Sun 2023-07-16 03:53:53 UTC; 45s ago
   Main PID: 52292 (code=exited, status=0/SUCCESS)
        CPU: 35ms

Jul 16 03:53:52 ip-172-31-47-23.eu-north-1.compute.internal systemd[1]: Starting NFS server and services...
Jul 16 03:53:53 ip-172-31-47-23.eu-north-1.compute.internal systemd[1]: Finished NFS server and services.
[ec2-user@ip-172-31-47-23 ~]$
```

- **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
```

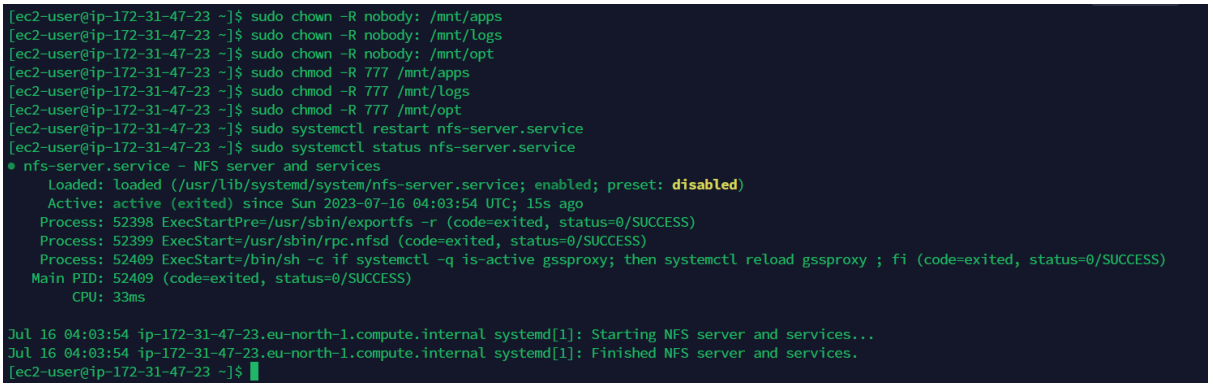
```
sudo chown -R nobody: /mnt/opt
```

```
sudo chmod -R 777 /mnt/apps
```

```
sudo chmod -R 777 /mnt/logs
```

```
sudo chmod -R 777 /mnt/opt
```

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



- **we will be creating our NFS-server, web-servers and database-server all in the same subnet**

```
sudo vi /etc/exports
```

```
/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)
```

```
Esc + :wq!
```

```
sudo exportfs -arv
```

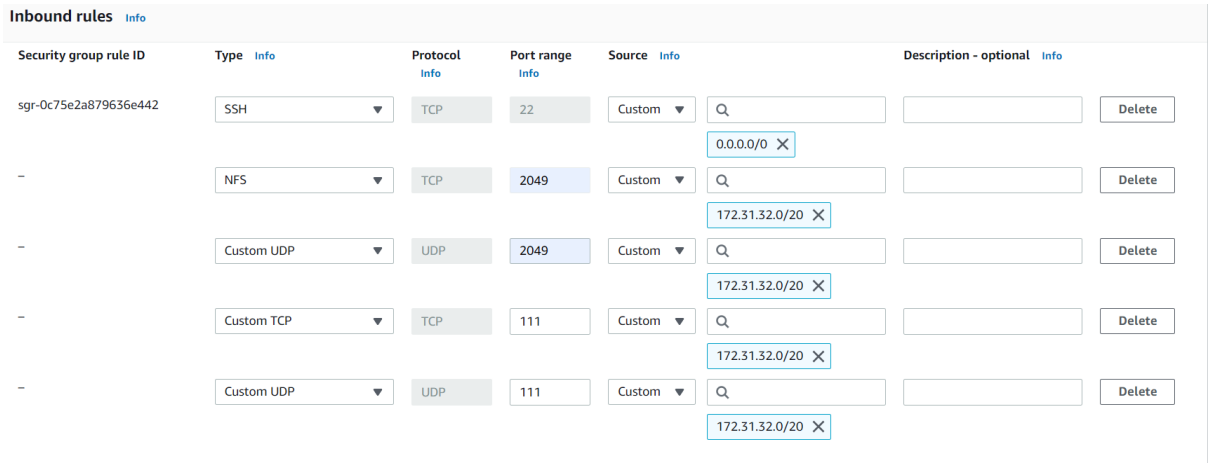
```
[ec2-user@ip-172-31-46-245 ~]$ sudo vi /etc/exports
[ec2-user@ip-172-31-46-245 ~]$ sudo exportfs -arv
exporting 172.31.32.0/20:/mnt/opt
exporting 172.31.32.0/20:/mnt/logs
exporting 172.31.32.0/20:/mnt/apps
[ec2-user@ip-172-31-46-245 ~]$
```

- **check what port is used by NFS so we can open it in security group**

`rpcinfo -p | grep nfs`

```
[ec2-user@ip-172-31-46-245 ~]$ rpcinfo -p | grep nfs
    100003    3   tcp   2049  nfs
    100003    4   tcp   2049  nfs
    100227    3   tcp   2049  nfs_acl
[ec2-user@ip-172-31-46-245 ~]$
```

- **Below ports are to be open on the NFS server**



Step 2: Preparing Database Server

- **Install MySQL server**

Create an Ubuntu Server on AWS which will serve as our Database. Ensure its in the same subnet as the NFS-Server.

```
sudo apt -y update
```

```
sudo apt install -y mysql-server
```

```
To enter the db environment run
```

```
sudo mysql
```

- **Create a database and name it tooling**
- **Create a database user and name it webaccess**
- **Grant permission to webaccess user on tooling database to do anything only from the webservers** `subnet cidr`

create user 'webaccess'@'172.31.32.0/20' identified by 'password';

grant all privileges on tooling.* to 'webaccess'@'172.31.32.0/20';

flush privileges;

```
ubuntu@ip-172-31-34-253:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.33-0ubuntu0.22.04.2 (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 tooling;
Query OK, 1 row affected (0.01 sec)

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

mysql> grant all privileges on tooling.* to 'webaccess'@'172.31.32.0/20';
Query OK, 0 rows affected (0.00 sec)

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 (0.01 sec)

mysql>
```

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            = 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

# This replaces the startup script and checks MyISAM tables if needed
-- INSERT --
```

Step 3; Prepare the Web Servers

- **I created a RHEL EC2 instance on AWS which serves as our web server. And they are in the same subnet.**

Below configurations will be done on the web servers:

- **configuring NFS client**
- **deploying tooling website application**
- **configure servers to work with database**

Installing 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-172-31-40-79 ~]$ sudo mkdir /var/www
[ec2-user@ip-172-31-40-79 ~]$ sudo mount -t nfs -o rw,nosuid 172.31.46.245:/mnt/apps /var/www
[ec2-user@ip-172-31-40-79 ~]$ 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  3.6M  146M   3% /run
/dev/nvme0n1p4           9.4G  1.3G  8.1G  14% /
/dev/nvme0n1p3           495M  153M  343M  31% /boot
/dev/nvme0n1p2           200M  8.0K  200M   1% /boot/efi
tmpfs                     75M     0   75M   0% /run/user/1000
172.31.46.245:/mnt/apps  9.0G   98M  8.9G   2% /var/www
[ec2-user@ip-172-31-40-79 ~]$
```

We then need to ensure that our mounts remain intact when the server reboots. This is achieved by configuring the fstab directory.

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

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

```
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
172.31.46.245:/mnt/apps /var/www nfs defaults 0 0
~
```

Installing Apache

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

```
[ec2-user@ip-172-31-46-245 ~]$ ls -l /mnt/apps
total 0
drwxr-xr-x. 2 root root 6 Apr 28 16:41 cgi-bin
drwxr-xr-x. 2 root root 6 Apr 28 16:41 html
[ec2-user@ip-172-31-46-245 ~]$
```

From webserver1;

```
[ec2-user@ip-172-31-40-79 ~]$ ls -l /var/www
total 0
drwxr-xr-x. 2 root root 6 Apr 28 16:41 cgi-bin
drwxr-xr-x. 2 root root 6 Apr 28 16:41 html
[ec2-user@ip-172-31-40-79 ~]$
```

From File server(NFS SERVER).

```
[ec2-user@ip-172-31-43-214 ~]$ ls -l /var/www
total 0
drwxr-xr-x. 2 root root 6 Apr 28 16:41 cgi-bin
drwxr-xr-x. 2 root root 6 Apr 28 16:41 html
[ec2-user@ip-172-31-43-214 ~]$ history
```

From webserver-2.

```
[ec2-user@ip-172-31-35-30 ~]$ ls -l /var/www
total 0
drwxr-xr-x. 2 root root 6 Apr 28 16:41 cgi-bin
drwxr-xr-x. 2 root root 6 Apr 28 16:41 html
[ec2-user@ip-172-31-35-30 ~]$
```

From the webserver-3;

This shows that the three webservers are in sync with the file server(NFS Server).

I Configured the remaining two servers like this first one;

- **We locate the log folder for Apache on the Web Server and mount it to NFS server's export for logs. Make sure the mount point will persist after reboot.**

```
[ec2-user@ip-172-31-40-79 ~]$ sudo mount -t nfs -o rw,nosuid 172.31.46.245:/mnt/logs /var/log
[ec2-user@ip-172-31-40-79 ~]$ sudo vi /etc/fstab
[ec2-user@ip-172-31-40-79 ~]$ 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  3.6M  146M   3% /run
/dev/nvme0n1p4           9.4G  1.4G  8.0G  15% /
/dev/nvme0n1p3           495M  153M  343M  31% /boot
/dev/nvme0n1p2           200M  8.0K  200M   1% /boot/efi
tmpfs                     75M     0   75M   0% /run/user/1000
172.31.46.245:/mnt/apps  9.0G   98M  8.9G   2% /var/www
172.31.46.245:/mnt/logs  9.0G   98M  8.9G   2% /var/log
[ec2-user@ip-172-31-40-79 ~]$
```

- **On the NFS Server, add web content into the `/mnt/apps` directory. This should contain a html folder. The same content will be present in the `/var/www` directory in the web server.**

On the webserver,

Install git

Do git init

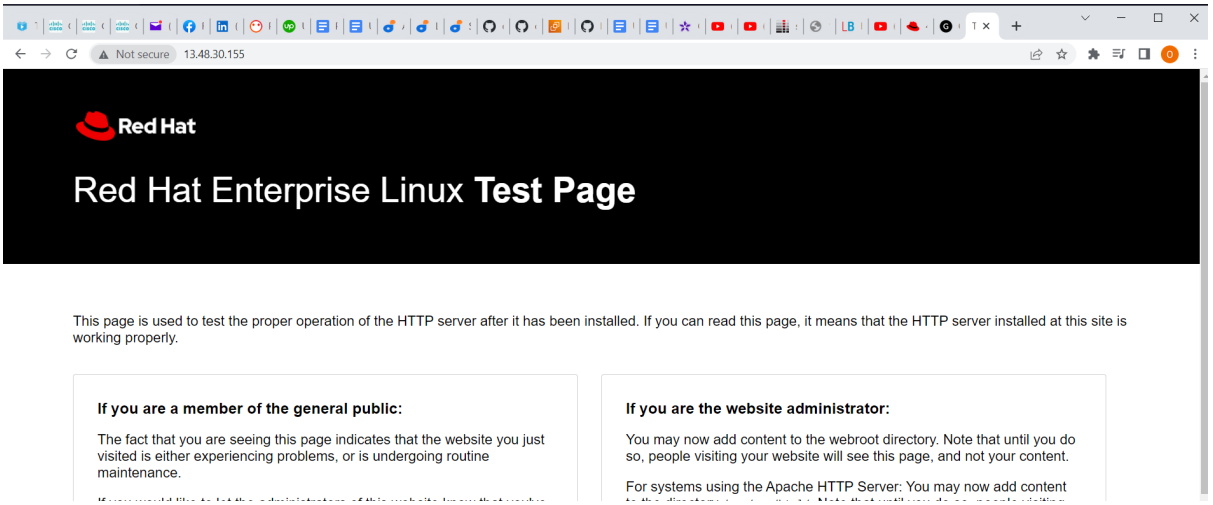
Git clone https://github.com/darey-io/tooling.git

Cd tooling

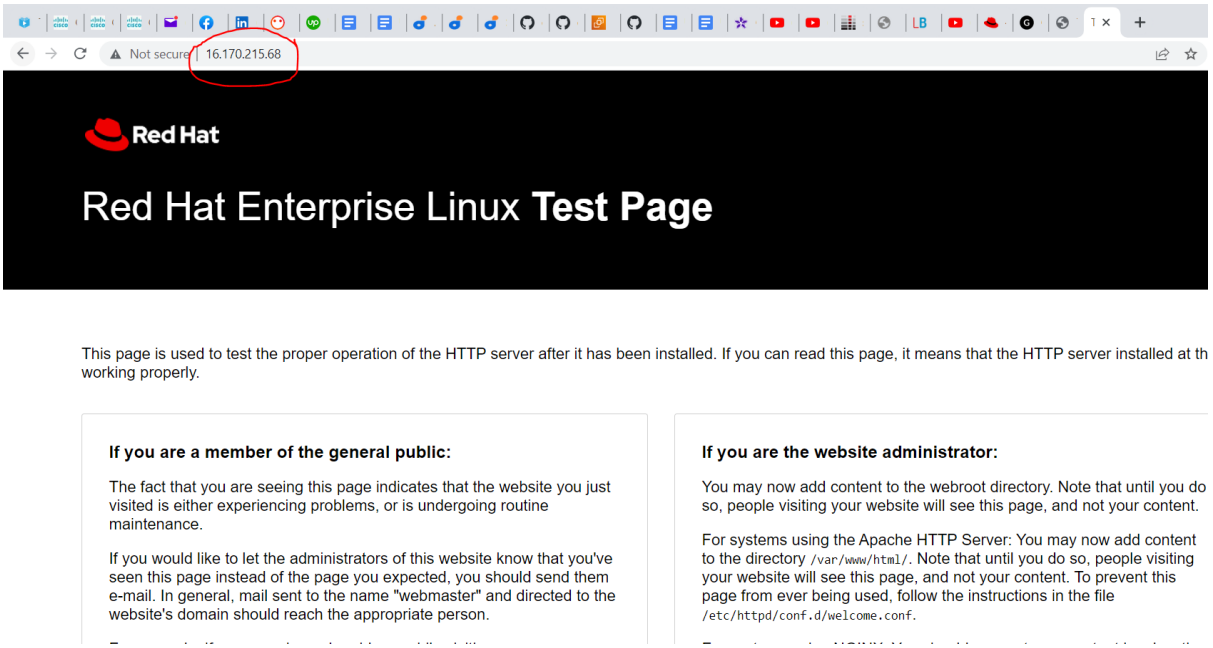
sudo cp -R html/. /var/www/html

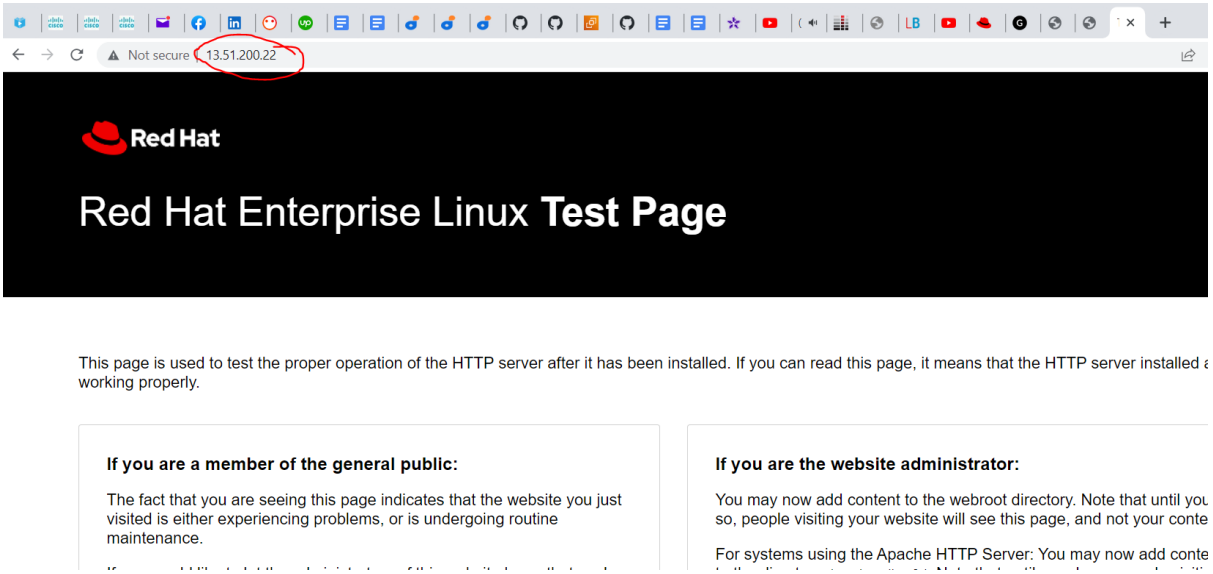
```
[ec2-user@ip-172-31-40-79 tooling]$ ls
apache-config.conf  Dockerfile  html  Jenkinsfile  README.md  start-apache  tooling-db.sql
[ec2-user@ip-172-31-40-79 tooling]$ ls /var/www
cgi-bin  html
[ec2-user@ip-172-31-40-79 tooling]$ sudo cp -R html/. /var/www/html
[ec2-user@ip-172-31-40-79 tooling]$ ls /var/www/html
admin_tooling.php  create_user.php  functions.php  img  index.php  login.php  README.md  register.php  style.css  tooling_stylesheets.css
[ec2-user@ip-172-31-40-79 tooling]$ ls html
admin_tooling.php  create_user.php  functions.php  img  index.php  login.php  README.md  register.php  style.css  tooling_stylesheets.css
[ec2-user@ip-172-31-40-79 tooling]$
```


Open port 80 on the webservers.


Not secure 13.48.30.155
Red Hat
Red Hat Enterprise Linux Test Page
This page is used to test the proper operation of the HTTP server after it has been installed. If you can read this page, it means that the HTTP server installed at this site is working properly.
If you are a member of the general public:
The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.
If you are the website administrator:
You may now add content to the webroot directory. Note that until you do so, people visiting your website will see this page, and not your content.
For systems using the Apache HTTP Server: You may now add content

Webserver1


Not secure 16.170.215.68
Red Hat
Red Hat Enterprise Linux Test Page
This page is used to test the proper operation of the HTTP server after it has been installed. If you can read this page, it means that the HTTP server installed at th
working properly.
If you are a member of the general public:
The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.
If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.
If you are the website administrator:
You may now add content to the webroot directory. Note that until you do so, people visiting your website will see this page, and not your content.
For systems using the Apache HTTP Server: You may now add content to the directory /var/www/html/. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file /etc/httpd/conf.d/welcome.conf.



From webserver3

So access the same site from different IPs; This is not good to have different IPs/URL for the same site. However the NFS is working.

I will need to remove this test page;

- **Update the website's configuration to connect to the database (in `/var/www/html/functions.php` file).**

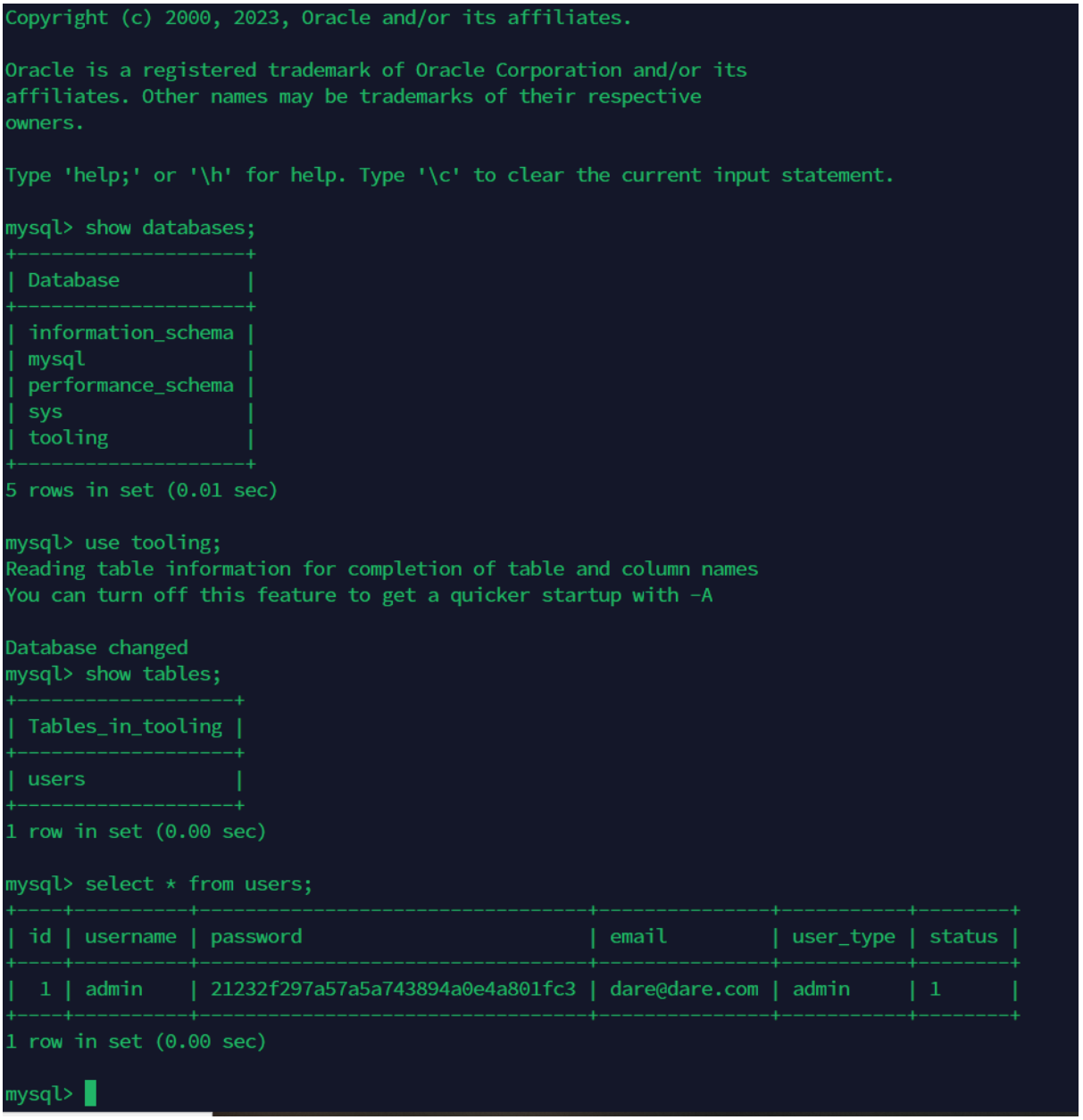
  ```
  sudo vi /var/www/html/functions.php
  ```

```php
<?php
session_start();

// connect to database
$db = mysqli_connect('172.31.34.253', '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");
        }
        if (empty($email)) {
```

- 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`

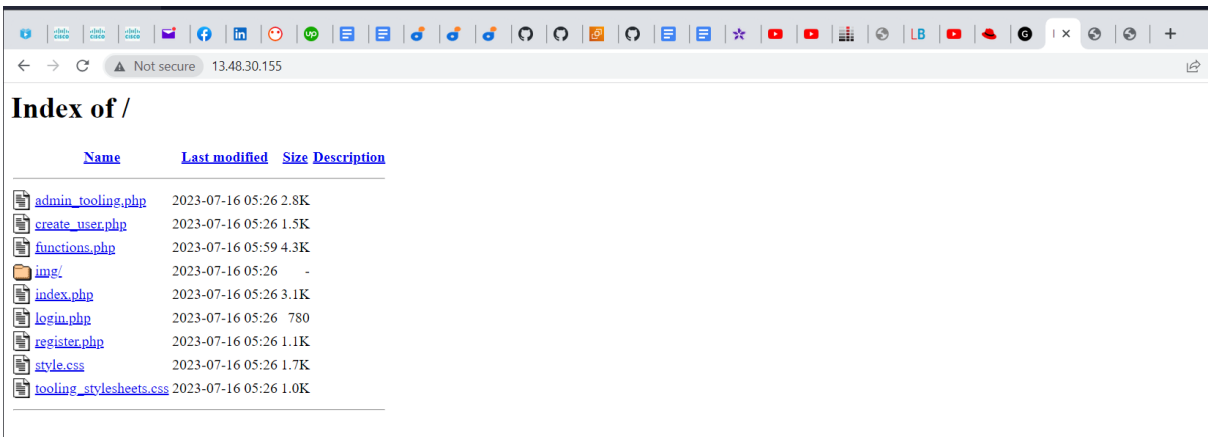
- **Install mysql client on the webservers**

```
Sudo yum install mysql
Cd tooling
mysql -h 172.31.34.253 -u webaccess -p tooling < tooling-db.sql
```



On the webserver remove and backup the test page;

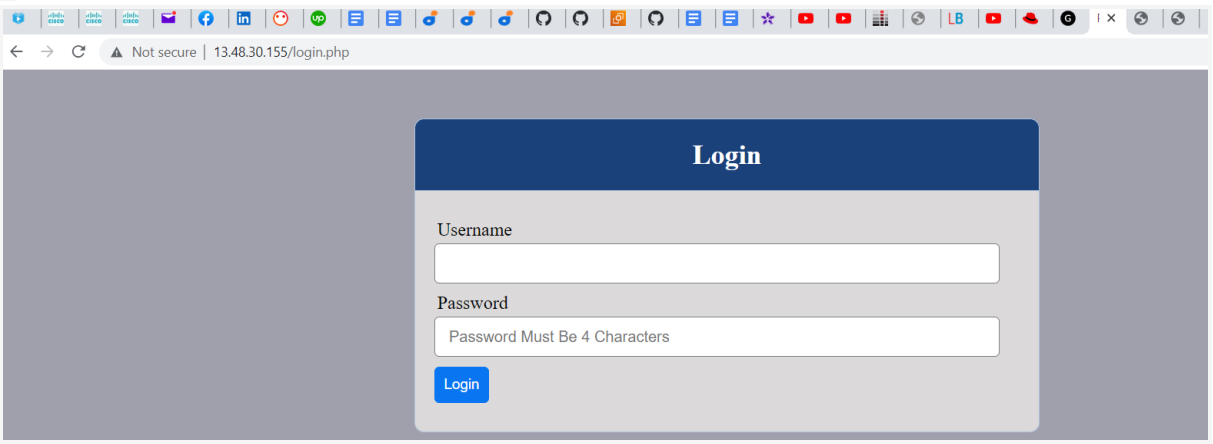
```
sudo mv /etc/httpd/conf.d/welcome.conf /etc/httpd/conf.d/welcome.backup
```



- **Install php**

```
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

sudo systemctl restart httpd
```

.



So using the existing username and password;