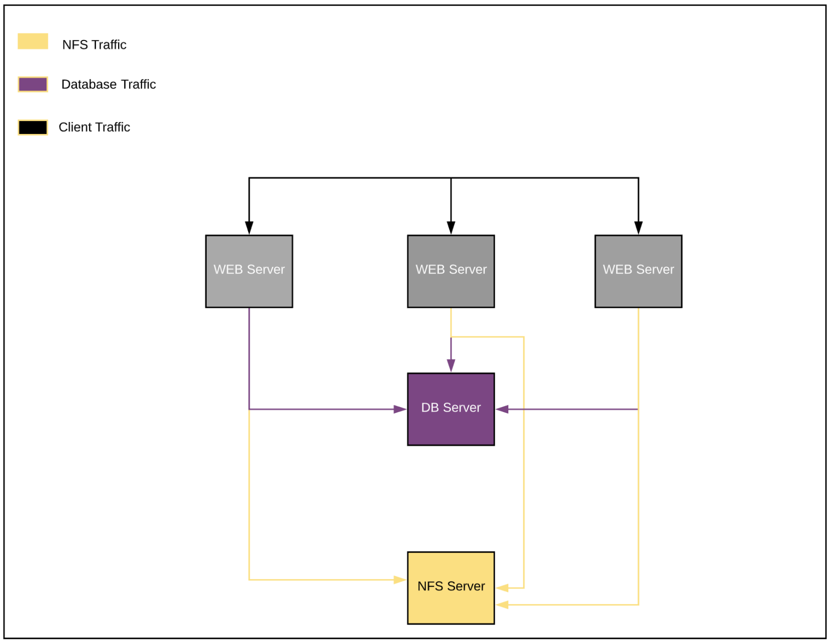
## DevOps Tooling Website Solution

The objective of this project is to develop a means of implementing a tooling website solution which enables easy access to DevOps tools within the corporate infrastructure.

In this project, we will be looking at how the NFS server is used to serve files to the web servers.Also, a database server will be configured for the DevOps tooling website solution. A basic view of the whole setup is illustrated below;



PREREQUISITES:

- Access to an AWS account and five (5) virtual instances with an Ubuntu 20.04 server OS image on one and the rest having the RHEL8.0 Server OS Image installed.
- A laptop or PC to serve as a client.

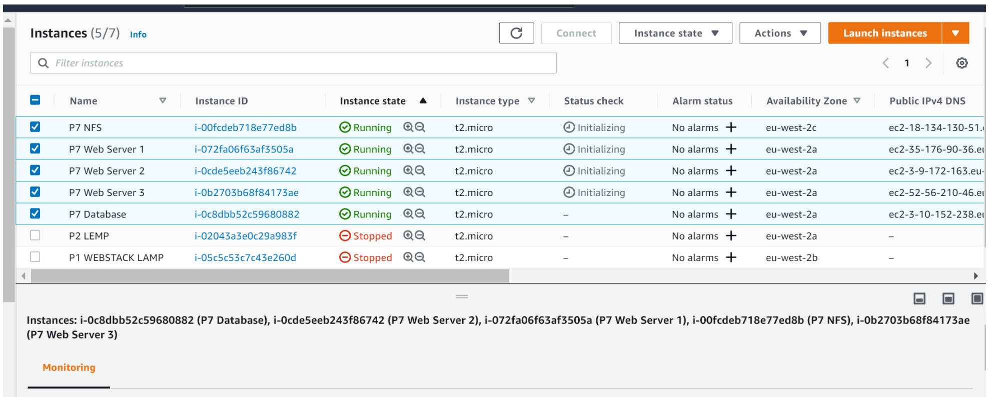
The steps include the below:

**STEP 1**: Launch 4 new EC2 instances with RHEL Linux 8 OS and one with Ubuntu 20.04.

In order to get started, we will require the IP addresses of the five EC2 instances as mentioned above:

NFS Server - **172.31.4.52**
Database Server - **172.31.18.63**

Web Server 1 - **172.31.16.162**
Web server 2 - **172.31.16.41**
Web server 3 - **172.31.30.14**



Instances (5/7) Info

| ☐ | Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 DNS |
|---|---|---|---|---|---|---|---|---|
| ☑ | P7 NFS | i-00fcdeb718e77ed8b | Running | t2.micro | Initializing | No alarms + | eu-west-2c | ec2-18-134-130-51. |
| ☑ | P7 Web Server 1 | i-072fa06f63af3505a | Running | t2.micro | Initializing | No alarms + | eu-west-2a | ec2-35-176-90-36.eu |
| ☑ | P7 Web Server 2 | i-0cde5eeb243f86742 | Running | t2.micro | Initializing | No alarms + | eu-west-2a | ec2-3-9-172-163.eu- |
| ☑ | P7 Web Server 3 | i-0b2703b68f84173ae | Running | t2.micro | Initializing | No alarms + | eu-west-2a | ec2-52-56-210-46.eu |
| ☑ | P7 Database | i-0c8dbb52c59680882 | Running | t2.micro | – | No alarms + | eu-west-2a | ec2-3-10-152-238.eu |
| ☐ | P2 LEMP | i-02043a3e0c29a983f | Stopped | t2.micro | – | No alarms + | eu-west-2a | – |
| ☐ | P1 WEBSTACK LAMP | i-05c5c53c7c43e260d | Stopped | t2.micro | – | No alarms + | eu-west-2b | – |

Instances: i-0c8dbb52c59680882 (P7 Database), i-0cde5eeb243f86742 (P7 Web Server 2), i-072fa06f63af3505a (P7 Web Server 1), i-00fcdeb718e77ed8b (P7 NFS), i-0b2703b68f84173ae (P7 Web Server 3)

## STEP 2: Preparing the NFS Server:

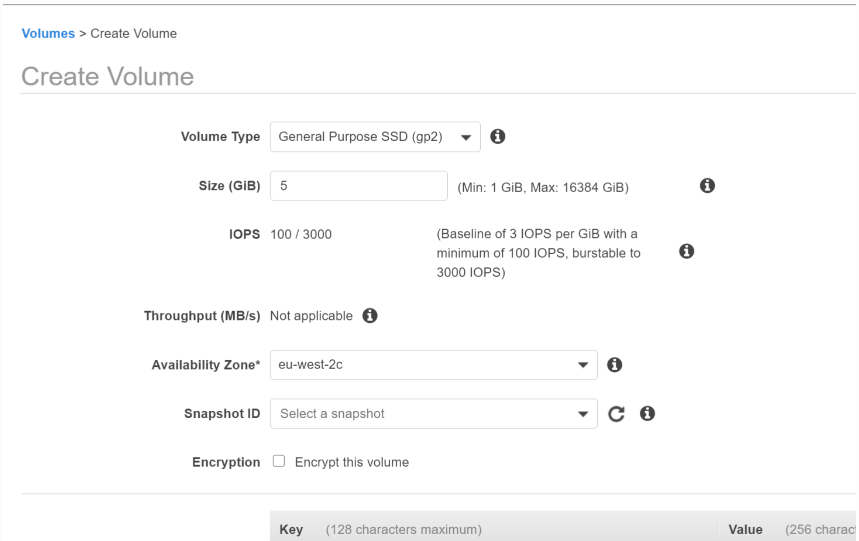
A Network File System is a distributed file system protocol originally developed by Sun Microsystems in 1984, allowing a user on a client computer to access files over a computer network much like local storage is accessed. It definitely has advantages and dis-advantages. You can read more here, but because of its shortcomings, it is not recommended for storing database files. Hence, we will only implement NFS for the website files.

a. Logical Volume Configuration:

We shall be equipping the NFS server with Logical Volume Management configuration. Our NFS server will make use of three (3)  of size 15GB each.

In order to choose the right storage system to implement the solution, we need to identify the suitable storage solution by checking: what data will be stored, in what format, how this data will be accessed, by whom, from where, how frequently, among others.

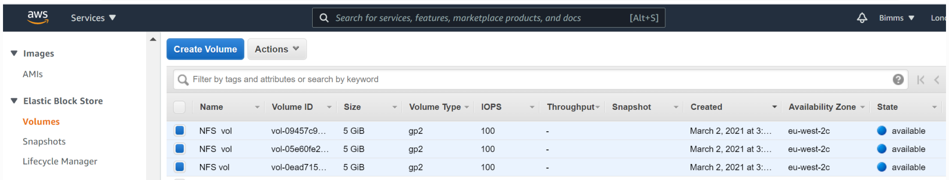
- Create and attach the 3 volumes to NFS instance,
- Locate the EBS volumes section and click on create volume
- Attach the volumes to the NFS server instance



Volumes > Create Volume

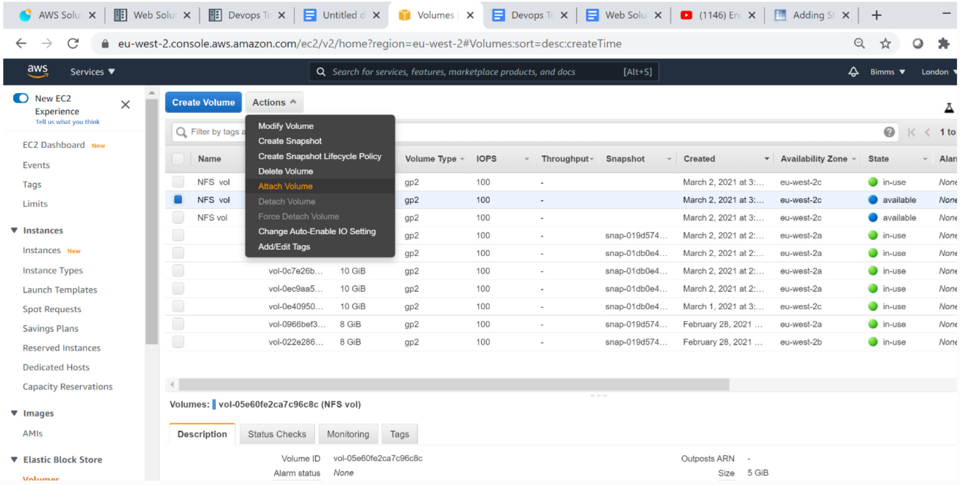
Create Volume

| | |
|---|---|
| Volume Type | General Purpose SSD (gp2) |
| Size (GiB) | 5 (Min: 1 GiB, Max: 16384 GiB) |
| IOPS | 100 / 3000 (Baseline of 3 IOPS per GiB with a minimum of 100 IOPS, burstable to 3000 IOPS) |
| Throughput (MB/s) | Not applicable |
| Availability Zone* | eu-west-2c |
| Snapshot ID | Select a snapshot |
| Encryption | □ Encrypt this volume |



aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
Bimms ▼
Lon
Images
AMIs
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Create Volume
Actions
Filter by tags and attributes or search by keyword

| | Name | Volume ID | Size | Volume Type | IOPS | Throughput | Snapshot | Created | Availability Zone | State |
|---|---|---|---|---|---|---|---|---|---|---|
| ☑ | NFS vol | vol-09457c9… | 5 GiB | gp2 | 100 | - | | March 2, 2021 at 3:… | eu-west-2c | ● available |
| ☑ | NFS vol | vol-05e60fe2… | 5 GiB | gp2 | 100 | - | | March 2, 2021 at 3:… | eu-west-2c | ● available |
| ☑ | NFS vol | vol-0ead715… | 5 GiB | gp2 | 100 | - | | March 2, 2021 at 3:… | eu-west-2c | ● available |



| | Name | | | Volume Type | IOPS | Throughput | Snapshot | Created | Availability Zone | State | Alar |
|---|---|---|---|---|---|---|---|---|---|---|---|
| □ | NFS vol | | | gp2 | 100 | - | | March 2, 2021 at 3:… | eu-west-2c | ● in-use | None |
| ☑ | NFS vol | | | gp2 | 100 | - | | March 2, 2021 at 3:… | eu-west-2c | ● available | None |
| □ | NFS vol | | | gp2 | 100 | - | | March 2, 2021 at 3:… | eu-west-2c | ● available | None |
| □ | | | | gp2 | 100 | - | snap-019d574… | March 2, 2021 at 2:… | eu-west-2a | ● in-use | None |
| □ | | | | gp2 | 100 | - | snap-01db0e4… | March 2, 2021 at 2:… | eu-west-2a | ● in-use | None |
| □ | | vol-0c7e26b… | 10 GiB | gp2 | 100 | - | snap-01db0e4… | March 2, 2021 at 2:… | eu-west-2a | ● in-use | None |
| □ | | vol-0ec9aa5… | 10 GiB | gp2 | 100 | - | snap-01db0e4… | March 2, 2021 at 2:… | eu-west-2a | ● in-use | None |
| □ | | vol-0e40950… | 10 GiB | gp2 | 100 | - | snap-01db0e4… | March 1, 2021 at 3:… | eu-west-2c | ● in-use | None |
| □ | | vol-0966bef3… | 8 GiB | gp2 | 100 | - | snap-019d574… | February 28, 2021 … | eu-west-2a | ● in-use | None |
| □ | | vol-022e286… | 8 GiB | gp2 | 100 | - | snap-019d574… | February 28, 2021 … | eu-west-2b | ● in-use | None |

Volumes: vol-05e60fe2ca7c96c8c (NFS vol)

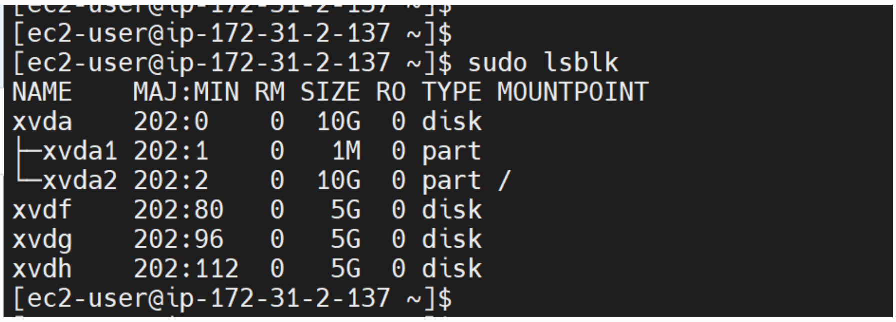
Volume ID vol-05e60fe2ca7c96c8c

Alarm status *None*

Outposts ARN -

Size 5 GiB

- Verify the disks are attached correctly: sudo lsblk

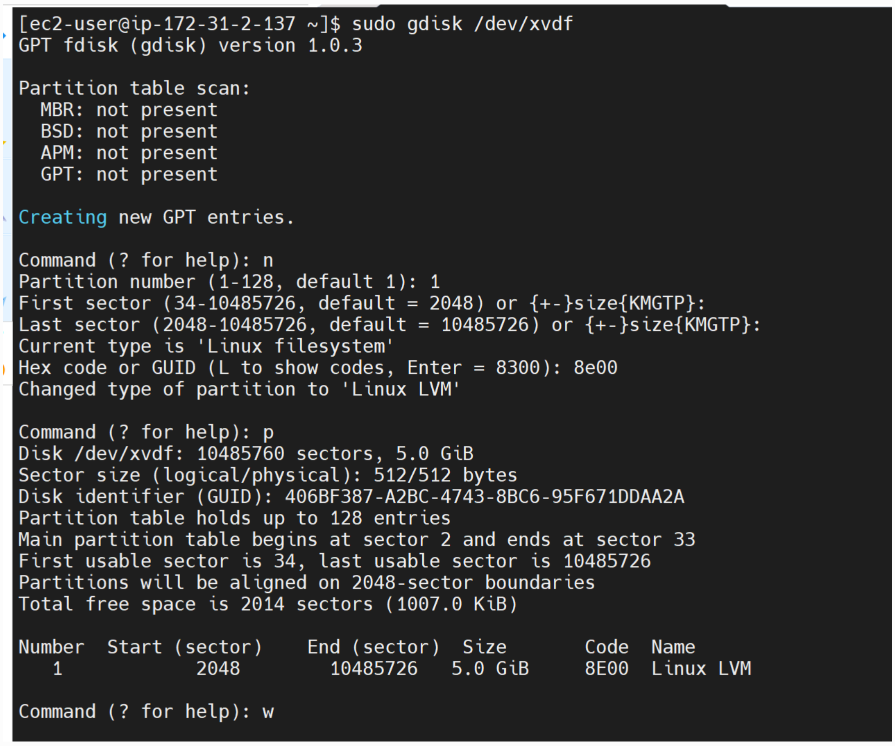


```
[ec2-user@ip-172-31-2-137 ~]$
[ec2-user@ip-172-31-2-137 ~]$
[ec2-user@ip-172-31-2-137 ~]$ sudo lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda    202:0    0  10G  0 disk
├─xvda1 202:1    0   1M  0 part
└─xvda2 202:2    0  10G  0 part /
xvdf    202:80   0   5G  0 disk
xvdg    202:96   0   5G  0 disk
xvdh    202:112  0   5G  0 disk
[ec2-user@ip-172-31-2-137 ~]$
```

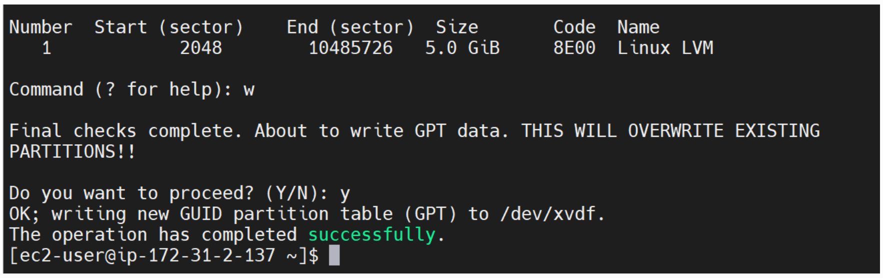
### Partitioning the disks:

Use gdisk utility to create a single partition on each of the 3 disks

```
sudo gdisk /dev/xvdf
```



```
[ec2-user@ip-172-31-2-137 ~]$ sudo gdisk /dev/xvdf
GPT fdisk (gdisk) version 1.0.3

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

Creating new GPT entries.

Command (? for help): n
Partition number (1-128, default 1): 1
First sector (34-10485726, default = 2048) or {+-}size{KMGTP}:
Last sector (2048-10485726, default = 10485726) or {+-}size{KMGTP}:
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300): 8e00
Changed type of partition to 'Linux LVM'

Command (? for help): p
Disk /dev/xvdf: 10485760 sectors, 5.0 GiB
Sector size (logical/physical): 512/512 bytes
Disk identifier (GUID): 406BF387-A2BC-4743-8BC6-95F671DDAA2A
Partition table holds up to 128 entries
Main partition table begins at sector 2 and ends at sector 33
First usable sector is 34, last usable sector is 10485726
Partitions will be aligned on 2048-sector boundaries
Total free space is 2014 sectors (1007.0 KiB)

Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048        10485726   5.0 GiB     8E00  Linux LVM

Command (? for help): w
```

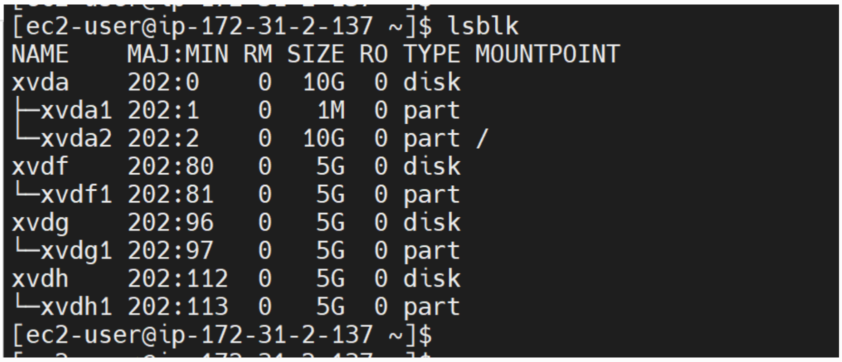


```
Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048        10485726   5.0 GiB     8E00  Linux LVM

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
OK; writing new GUID partition table (GPT) to /dev/xvdf.
The operation has completed successfully.
[ec2-user@ip-172-31-2-137 ~]$ 
```

Repeat the same for the three disks.

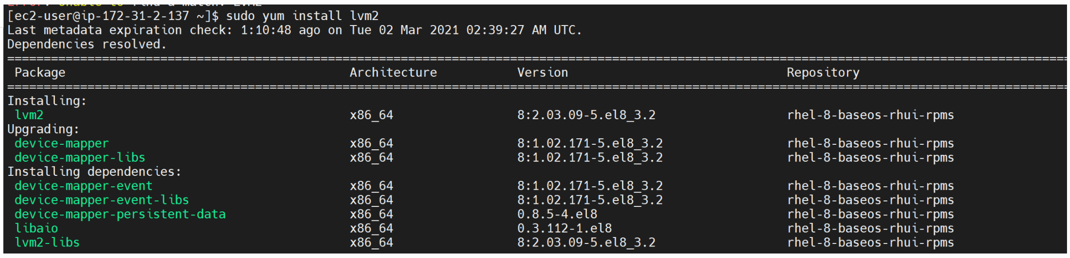
```
sudo gdisk /dev/xvdg
sudo gdisk /dev/xvdh
```

Run lsblk to view the newly configured partition on each of the 3 disks.



```
[ec2-user@ip-172-31-2-137 ~]$ lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda    202:0    0  10G  0 disk
├─xvda1 202:1    0   1M  0 part
└─xvda2 202:2    0  10G  0 part /
xvdf    202:80   0   5G  0 disk
└─xvdf1 202:81   0   5G  0 part
xvdg    202:96   0   5G  0 disk
└─xvdg1 202:97   0   5G  0 part
xvdh    202:112  0   5G  0 disk
└─xvdh1 202:113  0   5G  0 part
[ec2-user@ip-172-31-2-137 ~]$
```

- **Creating LVM logical Volume**
    - Install LVM package: `sudo yum install lvm2`



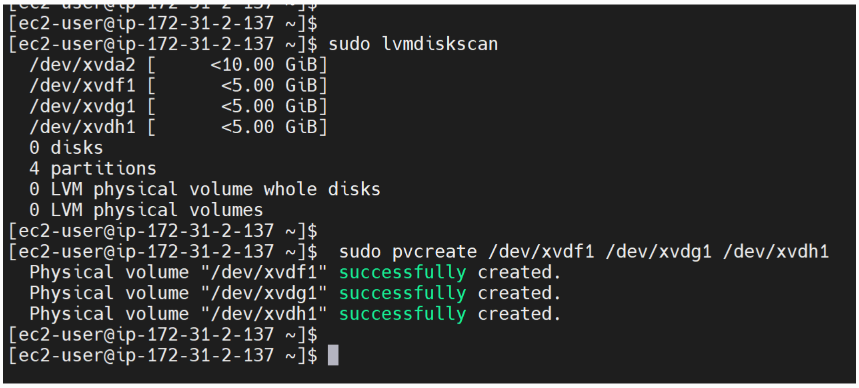
```
[ec2-user@ip-172-31-2-137 ~]$ sudo yum install lvm2
Last metadata expiration check: 1:10:48 ago on Tue 02 Mar 2021 02:39:27 AM UTC.
Dependencies resolved.
==========================================================================================================================================
 Package                                  Architecture          Version                              Repository
==========================================================================================================================================
Installing:
 lvm2                                     x86_64                8:2.03.09-5.el8_3.2                  rhel-8-baseos-rhui-rpms
Upgrading:
 device-mapper                            x86_64                8:1.02.171-5.el8_3.2                 rhel-8-baseos-rhui-rpms
 device-mapper-libs                       x86_64                8:1.02.171-5.el8_3.2                 rhel-8-baseos-rhui-rpms
Installing dependencies:
 device-mapper-event                      x86_64                8:1.02.171-5.el8_3.2                 rhel-8-baseos-rhui-rpms
 device-mapper-event-libs                 x86_64                8:1.02.171-5.el8_3.2                 rhel-8-baseos-rhui-rpms
 device-mapper-persistent-data            x86_64                0.8.5-4.el8                          rhel-8-baseos-rhui-rpms
 libaio                                   x86_64                0.3.112-1.el8                        rhel-8-baseos-rhui-rpms
 lvm2-libs                                x86_64                8:2.03.09-5.el8_3.2                  rhel-8-baseos-rhui-rpms
```

- Use lvmdiskscan utility to scan / check available storage for LVM. It shows the devices that are  suitable to be turned in physical volumes for LVM.

    `sudo lvmdiskscan`

- Create and mark the newly created partitions of the raw storage device as LVM physical volumes using the command below;

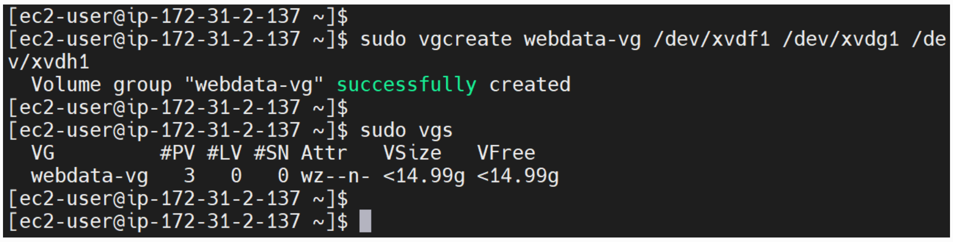
    `sudo pvcreate /dev/xvdf1 /dev/xvdg1 /dev/xvdh1`



```
[ec2-user@ip-172-31-2-137 ~]$
[ec2-user@ip-172-31-2-137 ~]$ sudo lvmdiskscan
  /dev/xvda2 [     <10.00 GiB]
  /dev/xvdf1 [      <5.00 GiB]
  /dev/xvdg1 [      <5.00 GiB]
  /dev/xvdh1 [      <5.00 GiB]
  0 disks
  4 partitions
  0 LVM physical volume whole disks
  0 LVM physical volumes
[ec2-user@ip-172-31-2-137 ~]$
[ec2-user@ip-172-31-2-137 ~]$  sudo pvcreate /dev/xvdf1 /dev/xvdg1 /dev/xvdh1
  Physical volume "/dev/xvdf1" successfully created.
  Physical volume "/dev/xvdg1" successfully created.
  Physical volume "/dev/xvdh1" successfully created.
[ec2-user@ip-172-31-2-137 ~]$
[ec2-user@ip-172-31-2-137 ~]$ █
```

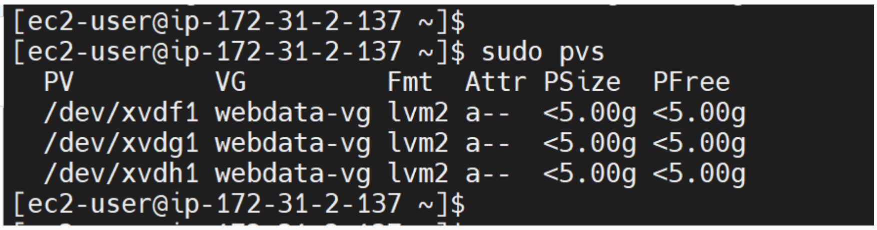
- Create a volume group named **'webdata'** using the physical group already created.

    `sudo vgcreate vg-webdata /dev/xvdf1 /dev/xvdg1 /dev/xvdh1`



```
[ec2-user@ip-172-31-2-137 ~]$
[ec2-user@ip-172-31-2-137 ~]$ sudo vgcreate webdata-vg /dev/xvdf1 /dev/xvdg1 /de
v/xvdh1
  Volume group "webdata-vg" successfully created
[ec2-user@ip-172-31-2-137 ~]$
[ec2-user@ip-172-31-2-137 ~]$ sudo vgs
  VG          #PV #LV #SN Attr   VSize   VFree
  webdata-vg    3   0   0 wz--n- <14.99g <14.99g
[ec2-user@ip-172-31-2-137 ~]$
[ec2-user@ip-172-31-2-137 ~]$ █
```

Confirm the PVs / vgs: sudo pvs

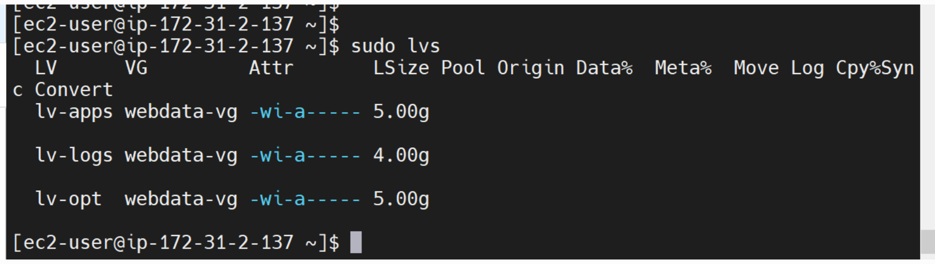


```
[ec2-user@ip-172-31-2-137 ~]$
[ec2-user@ip-172-31-2-137 ~]$ sudo pvs
  PV         VG         Fmt  Attr PSize  PFree
  /dev/xvdf1 webdata-vg lvm2 a--  <5.00g <5.00g
  /dev/xvdg1 webdata-vg lvm2 a--  <5.00g <5.00g
  /dev/xvdh1 webdata-vg lvm2 a--  <5.00g <5.00g
[ec2-user@ip-172-31-2-137 ~]$
```

- Create three logical volumes, lv-opt lv-apps, and lv-logs:

  sudo lvcreate -n lv-opt -L 5G webdata-vg
  sudo lvcreate -n lv-apps -L 5G webdata-vg
  sudo lvcreate -n lv-logs -L 4G webdata-vg

  Confirm LVs have been created: sudo lvs



```
[ec2-user@ip-172-31-2-137 ~]$
[ec2-user@ip-172-31-2-137 ~]$ sudo lvs
  LV      VG         Attr       LSize Pool Origin Data%  Meta%  Move Log Cpy%Syn
c Convert
  lv-apps webdata-vg -wi-a----- 5.00g

  lv-logs webdata-vg -wi-a----- 4.00g

  lv-opt  webdata-vg -wi-a----- 5.00g

[ec2-user@ip-172-31-2-137 ~]$
```

Please note - a reduced space was allocated to lv-logs because the volume group will have to use some of the space it has to store necessary information about itself.

Confirm all the set up with: sudo vgdisplay -v, pvs, lvs and sudo lsblk

**Formatting the logical volumes with xfs file system**

We will use the mkfs utility to achieve this with the command lines below:

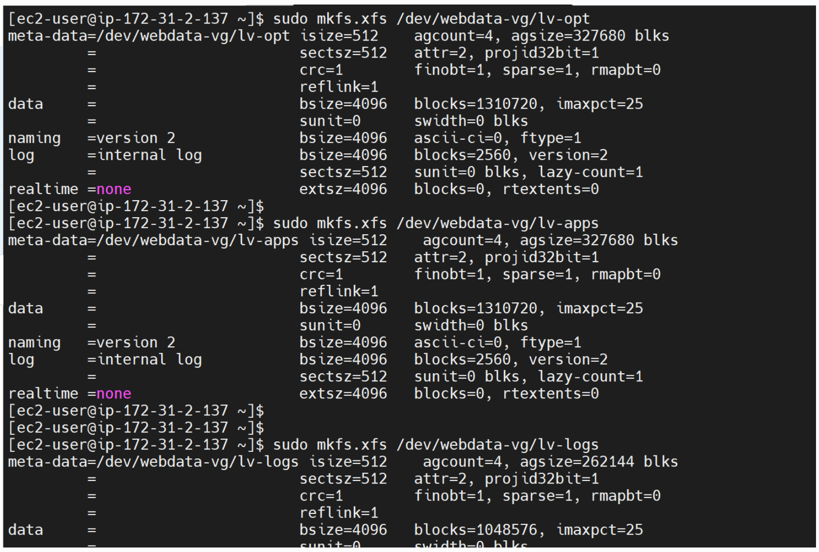
```
sudo mkfs.xfs /dev/webdata-vg/lv-opt
sudo mkfs.xfs /dev/webdata-vg/lv-apps
sudo mkfs.xfs /dev/webdata-vg/lv-logs
```

*Create mount points on /mnt directory for the logical volumes as follows;

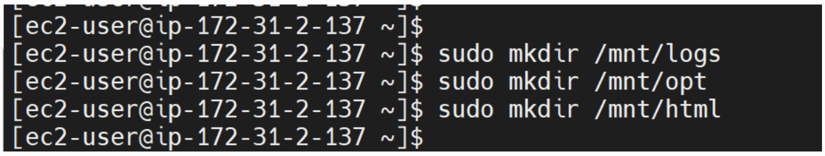
| | |
|---|---|
| - /mnt/logs | sudo mkdir /mnt/logs |
| - /mnt/opt | sudo mkdir /mnt/opt |
| - /mnt/apps | sudo mkdir /mnt/apps |

*Mount lv-apps on /mnt/html - To be used by web servers Mount lv-logs on /mnt/logs - To be used by web server logs Mount lv-opt on /mnt/opt - To be used by Jenkins server in Project 8.

```
sudo mount /dev/webdata-vg/lv-opt /mnt/opt
sudo mount /dev/webdata-vg/lv-apps /mnt/apps
sudo mount /dev/webdata-vg/lv-logs /mnt/logs
```

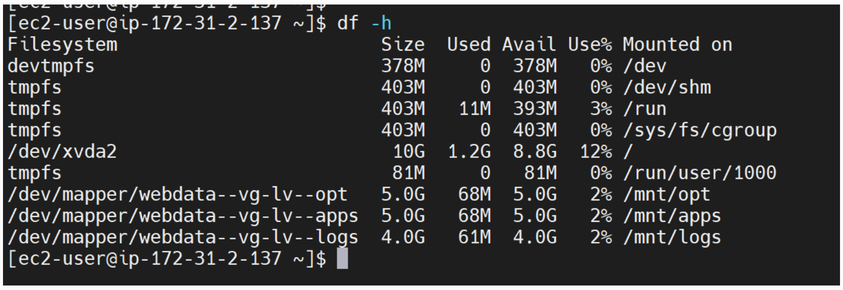


```
[ec2-user@ip-172-31-2-137 ~]$ sudo mkfs.xfs /dev/webdata-vg/lv-opt
meta-data=/dev/webdata-vg/lv-opt isize=512    agcount=4, agsize=327680 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1
data     =                       bsize=4096   blocks=1310720, 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@ip-172-31-2-137 ~]$
[ec2-user@ip-172-31-2-137 ~]$ sudo mkfs.xfs /dev/webdata-vg/lv-apps
meta-data=/dev/webdata-vg/lv-apps isize=512    agcount=4, agsize=327680 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1
data     =                       bsize=4096   blocks=1310720, 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@ip-172-31-2-137 ~]$
[ec2-user@ip-172-31-2-137 ~]$
[ec2-user@ip-172-31-2-137 ~]$ sudo mkfs.xfs /dev/webdata-vg/lv-logs
meta-data=/dev/webdata-vg/lv-logs isize=512    agcount=4, agsize=262144 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1
data     =                       bsize=4096   blocks=1048576, imaxpct=25
         =                       sunit=0      swidth=0 blks
```



```
[ec2-user@ip-172-31-2-137 ~]$
[ec2-user@ip-172-31-2-137 ~]$ sudo mkdir /mnt/logs
[ec2-user@ip-172-31-2-137 ~]$ sudo mkdir /mnt/opt
[ec2-user@ip-172-31-2-137 ~]$ sudo mkdir /mnt/html
[ec2-user@ip-172-31-2-137 ~]$
```

Confirm the mounts: df -h

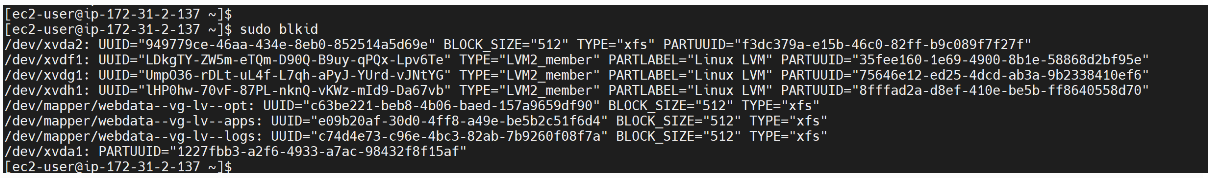


```
[ec2-user@ip-172-31-2-137 ~]$ df -h
Filesystem                         Size  Used Avail Use% Mounted on
devtmpfs                           378M     0  378M   0% /dev
tmpfs                              403M     0  403M   0% /dev/shm
tmpfs                              403M   11M  393M   3% /run
tmpfs                              403M     0  403M   0% /sys/fs/cgroup
/dev/xvda2                          10G  1.2G  8.8G  12% /
tmpfs                               81M     0   81M   0% /run/user/1000
/dev/mapper/webdata--vg-lv--opt    5.0G   68M  5.0G   2% /mnt/opt
/dev/mapper/webdata--vg-lv--apps   5.0G   68M  5.0G   2% /mnt/apps
/dev/mapper/webdata--vg-lv--logs   4.0G   61M  4.0G   2% /mnt/logs
[ec2-user@ip-172-31-2-137 ~]$
```

*Update the ‘/etc/fstab’ file in order to enable the mount configuration to persist upon restart of the server.

We will use the block id command to obtain the UUU code, which the system device’s name.

```
sudo blkid
```

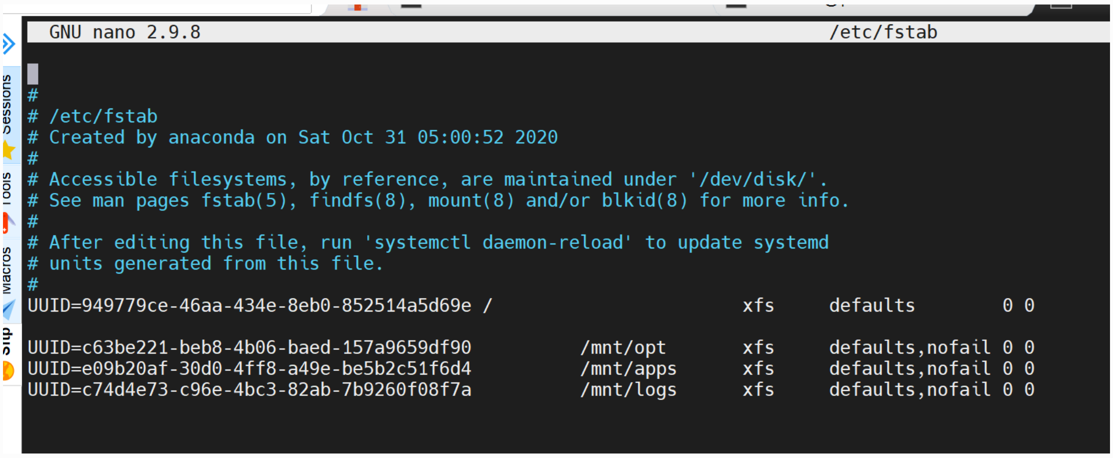


```
[ec2-user@ip-172-31-2-137 ~]$
[ec2-user@ip-172-31-2-137 ~]$ sudo blkid
/dev/xvda2: UUID="949779ce-46aa-434e-8eb0-852514a5d69e" BLOCK_SIZE="512" TYPE="xfs" PARTUUID="f3dc379a-e15b-46c0-82ff-b9c089f7f27f"
/dev/xvdf1: UUID="LDkgTY-ZW5m-eTQm-D90Q-B9uy-qPQx-Lpv6Te" TYPE="LVM2_member" PARTLABEL="Linux LVM" PARTUUID="35fee160-1e69-4900-8b1e-58868d2bf95e"
/dev/xvdg1: UUID="Ump036-rDLt-uL4f-L7qh-aPyJ-YUrd-vJNtYG" TYPE="LVM2_member" PARTLABEL="Linux LVM" PARTUUID="75646e12-ed25-4dcd-ab3a-9b2338410ef6"
/dev/xvdh1: UUID="lHP0hw-70vF-87PL-nknQ-vKWz-mId9-Da67vb" TYPE="LVM2_member" PARTLABEL="Linux LVM" PARTUUID="8fffad2a-d8ef-410e-be5b-ff8640558d70"
/dev/mapper/webdata--vg-lv--opt: UUID="c63be221-beb8-4b06-baed-157a9659df90" BLOCK_SIZE="512" TYPE="xfs"
/dev/mapper/webdata--vg-lv--apps: UUID="e09b20af-30d0-4ff8-a49e-be5b2c51f6d4" BLOCK_SIZE="512" TYPE="xfs"
/dev/mapper/webdata--vg-lv--logs: UUID="c74d4e73-c96e-4bc3-82ab-7b9260f08f7a" BLOCK_SIZE="512" TYPE="xfs"
/dev/xvda1: PARTUUID="1227fbb3-a2f6-4933-a7ac-98432f8f15af"
[ec2-user@ip-172-31-2-137 ~]$
```

```
UUID=c63be221-beb8-4b06-baed-157a9659df90  /mnt/opt
UUID=e09b20af-30d0-4ff8-a49e-be5b2c51f6d4  /mnt/apps
UUID=c74d4e73-c96e-4bc3-82ab-7b9260f08f7a  /mnt/logs
```

**Go into the /etc/fstab/ file and paste the below in:**

```
UUID=c63be221-beb8-4b06-baed-157a9659df90         /mnt/opt        xfs     defaults,nofail, 0 0
UUID=e09b20af-30d0-4ff8-a49e-be5b2c51f6d4         /mnt/apps       xfs     defaults,nofail, 0 0
UUID=c74d4e73-c96e-4bc3-82ab-7b9260f08f7a         /mnt/logs       xfs     defaults,nofail, 0 0
```

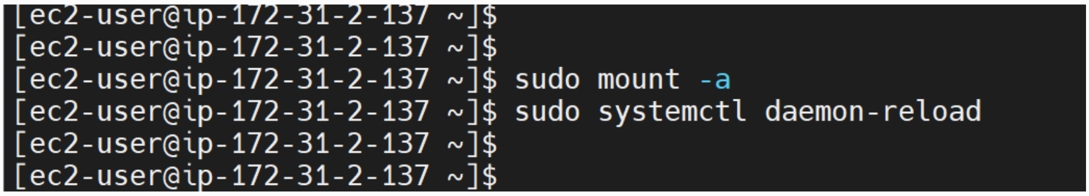


```
  GNU nano 2.9.8                                                     /etc/fstab

#
# /etc/fstab
# Created by anaconda on Sat Oct 31 05:00:52 2020
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
UUID=949779ce-46aa-434e-8eb0-852514a5d69e /                       xfs     defaults        0 0

UUID=c63be221-beb8-4b06-baed-157a9659df90         /mnt/opt        xfs     defaults,nofail 0 0
UUID=e09b20af-30d0-4ff8-a49e-be5b2c51f6d4         /mnt/apps       xfs     defaults,nofail 0 0
UUID=c74d4e73-c96e-4bc3-82ab-7b9260f08f7a         /mnt/logs       xfs     defaults,nofail 0 0
```

Save and close.

Check that the configuration in the etc/fstab is correct: `sudo mount -a`

Update all changes: `sudo systemctl daemon-reload`



```
[ec2-user@ip-172-31-2-137 ~]$
[ec2-user@ip-172-31-2-137 ~]$
[ec2-user@ip-172-31-2-137 ~]$ sudo mount -a
[ec2-user@ip-172-31-2-137 ~]$ sudo systemctl daemon-reload
[ec2-user@ip-172-31-2-137 ~]$
[ec2-user@ip-172-31-2-137 ~]$
```

## Install and Configure the NFS Server.

Firstly, install the NFS utils package

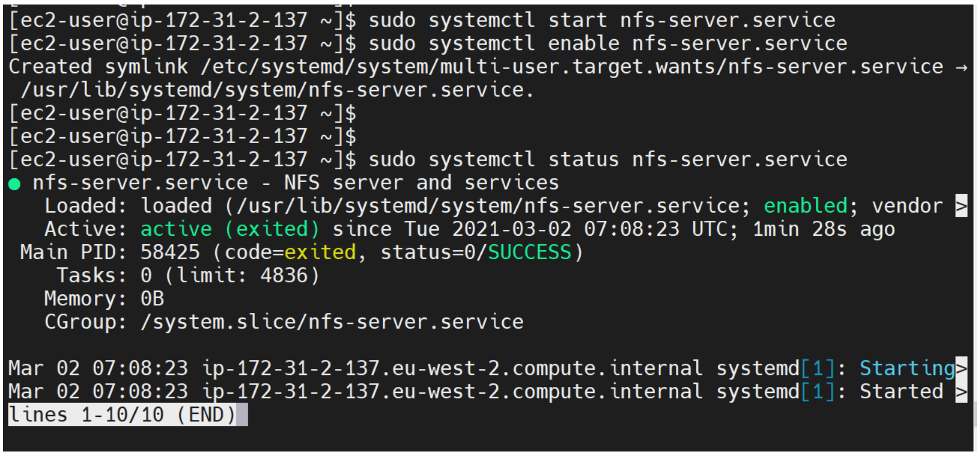
```
sudo yum -y update
sudo yum install nfs-utils -y
```

Start and enable the NFS service

```
sudo systemctl start nfs-server.service
sudo systemctl enable nfs-server.service
```

Check the NFS status

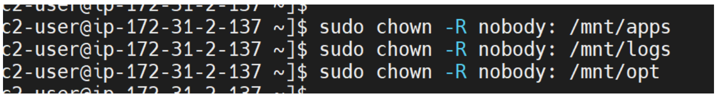
```
sudo systemctl status nfs-server.service
```



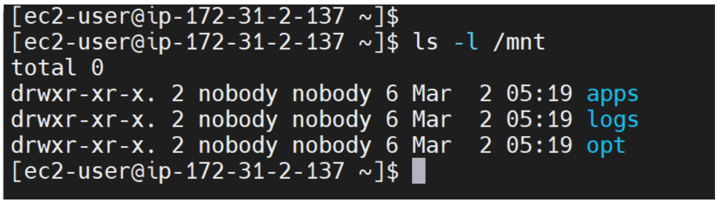
```
[ec2-user@ip-172-31-2-137 ~]$ sudo systemctl start nfs-server.service
[ec2-user@ip-172-31-2-137 ~]$ sudo systemctl enable nfs-server.service
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-server.service →
 /usr/lib/systemd/system/nfs-server.service.
[ec2-user@ip-172-31-2-137 ~]$
[ec2-user@ip-172-31-2-137 ~]$
[ec2-user@ip-172-31-2-137 ~]$ sudo systemctl status nfs-server.service
● nfs-server.service - NFS server and services
   Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled; vendor >
   Active: active (exited) since Tue 2021-03-02 07:08:23 UTC; 1min 28s ago
 Main PID: 58425 (code=exited, status=0/SUCCESS)
    Tasks: 0 (limit: 4836)
   Memory: 0B
   CGroup: /system.slice/nfs-server.service

Mar 02 07:08:23 ip-172-31-2-137.eu-west-2.compute.internal systemd[1]: Starting>
Mar 02 07:08:23 ip-172-31-2-137.eu-west-2.compute.internal systemd[1]: Started >
lines 1-10/10 (END)
```

b. Change the permission of the directories to be exported. The '777' will enable our Web servers to read, write and execute files on NFS. Also note that the 'nobody' implies that even though the permission has been given, only the servers that will be exported will have this access.

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

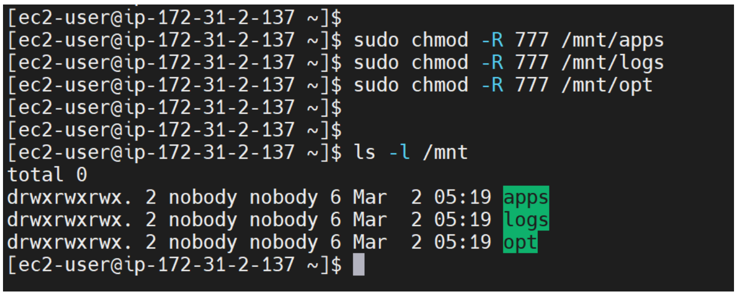


```
c2-user@ip-172-31-2-137 ~]$ sudo chown -R nobody: /mnt/apps
c2-user@ip-172-31-2-137 ~]$ sudo chown -R nobody: /mnt/logs
c2-user@ip-172-31-2-137 ~]$ sudo chown -R nobody: /mnt/opt
```



```
[ec2-user@ip-172-31-2-137 ~]$
[ec2-user@ip-172-31-2-137 ~]$ ls -l /mnt
total 0
drwxr-xr-x. 2 nobody nobody 6 Mar  2 05:19 apps
drwxr-xr-x. 2 nobody nobody 6 Mar  2 05:19 logs
drwxr-xr-x. 2 nobody nobody 6 Mar  2 05:19 opt
[ec2-user@ip-172-31-2-137 ~]$
```

```
sudo chmod -R 777 /mnt/apps
sudo chmod -R 777 /mnt/logs
sudo chmod -R 777 /mnt/opt
OR  sudo chmod 777 /mnt/logs /mnt/opt /mnt/apps
```



```
[ec2-user@ip-172-31-2-137 ~]$
[ec2-user@ip-172-31-2-137 ~]$ sudo chmod -R 777 /mnt/apps
[ec2-user@ip-172-31-2-137 ~]$ sudo chmod -R 777 /mnt/logs
[ec2-user@ip-172-31-2-137 ~]$ sudo chmod -R 777 /mnt/opt
[ec2-user@ip-172-31-2-137 ~]$
[ec2-user@ip-172-31-2-137 ~]$
[ec2-user@ip-172-31-2-137 ~]$ ls -l /mnt
total 0
drwxrwxrwx. 2 nobody nobody 6 Mar  2 05:19 apps
drwxrwxrwx. 2 nobody nobody 6 Mar  2 05:19 logs
drwxrwxrwx. 2 nobody nobody 6 Mar  2 05:19 opt
[ec2-user@ip-172-31-2-137 ~]$
```

Restart the nfs-server to incorporate all the changes

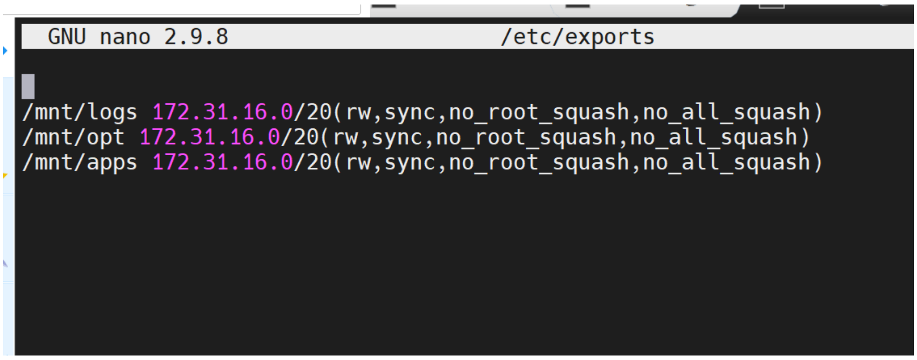
```
sudo systemctl restart nfs-server.service
```

Specify the configuration for the mount exports in the /etc/exports file:

```
sudo nano /etc/exports
```

Subnet for NFS Server: 172.31.0.0/20

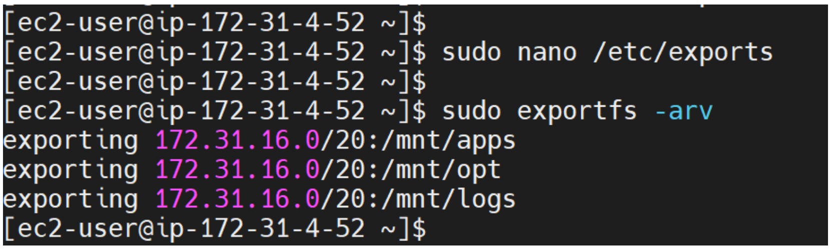
```
/mnt/logs 172.31.16.0/20(rw,sync,no_root_squash,no_all_squash)
/mnt/opt 172.31.16.0/20(rw,sync,no_root_squash,no_all_squash)
/mnt/apps 172.31.16.0/20(rw,sync,no_root_squash,no_all_squash)
```



```
  GNU nano 2.9.8                    /etc/exports

/mnt/logs 172.31.16.0/20(rw,sync,no_root_squash,no_all_squash)
/mnt/opt 172.31.16.0/20(rw,sync,no_root_squash,no_all_squash)
/mnt/apps 172.31.16.0/20(rw,sync,no_root_squash,no_all_squash)
```

Export the mounts for web servers' subnet cidr to connect as clients.:

```
sudo exportfs -arv
```



```
[ec2-user@ip-172-31-4-52 ~]$
[ec2-user@ip-172-31-4-52 ~]$ sudo nano /etc/exports
[ec2-user@ip-172-31-4-52 ~]$
[ec2-user@ip-172-31-4-52 ~]$ sudo exportfs -arv
exporting 172.31.16.0/20:/mnt/apps
exporting 172.31.16.0/20:/mnt/opt
exporting 172.31.16.0/20:/mnt/logs
[ec2-user@ip-172-31-4-52 ~]$
```

Restart NFS service: `sudo systemctl enable nfs-server.service`

***Blocker - At this point, my EC2 instance got corrupted and could not connect to the terminal.**

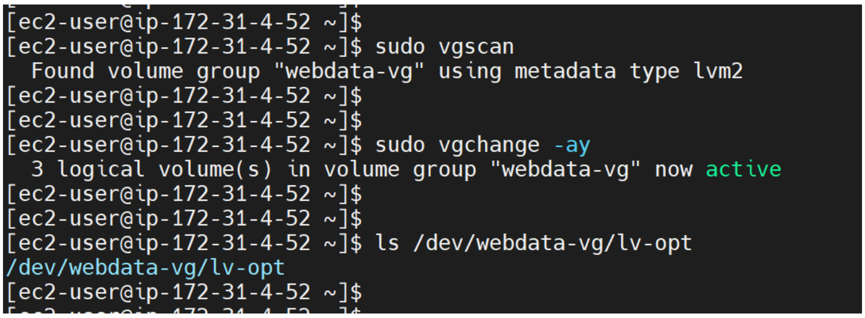
Actions taken to correct:

a. Detected what the issue was through the error in AWS.
    Got to EC2 instance in aws
    Action menu
    Monitor and troubleshoot
    Get system log
    Noticed issues with fstab configuration - input ‘nofails’ instead of ‘nofail’
b. Stopped current instance
c. Detached the volumes
d. Created a new instance
e. Attached the root volume from the initial instance
f. Tried to resolve the issue via terminal before attaching back as `/dev/sda1` but could not get the initial instance to initialise and run again.

Next steps:

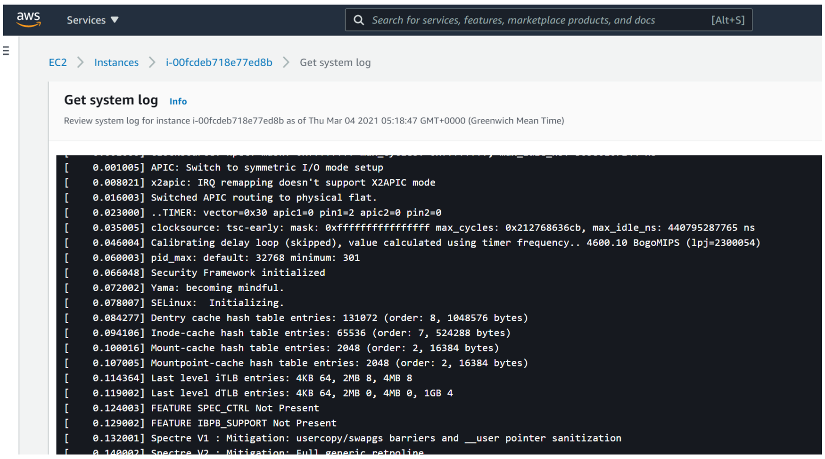
a. Created a new instance
b. Attached all the three volumes (already formatted and partitioned)
    - Connected to the terminal
    - Checked and ensured I could still access all the logical volumes
    - Then activated the logical volumes on my new instance with the command

      ```
      sudo vgscan
      sudo vgchange -ay
      ```



```
[ec2-user@ip-172-31-4-52 ~]$
[ec2-user@ip-172-31-4-52 ~]$ sudo vgscan
  Found volume group "webdata-vg" using metadata type lvm2
[ec2-user@ip-172-31-4-52 ~]$
[ec2-user@ip-172-31-4-52 ~]$
[ec2-user@ip-172-31-4-52 ~]$ sudo vgchange -ay
  3 logical volume(s) in volume group "webdata-vg" now active
[ec2-user@ip-172-31-4-52 ~]$
[ec2-user@ip-172-31-4-52 ~]$
[ec2-user@ip-172-31-4-52 ~]$ ls /dev/webdata-vg/lv-opt
/dev/webdata-vg/lv-opt
[ec2-user@ip-172-31-4-52 ~]$
```

List with: `sudo lvdisplay`
           `sudo lvs`



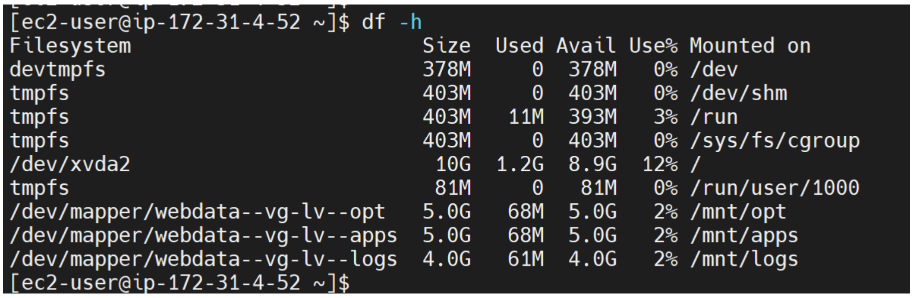
Get system log

Review system log for instance i-00fcdeb718e77ed8b as of Thu Mar 04 2021 05:18:47 GMT+0000 (Greenwich Mean Time)

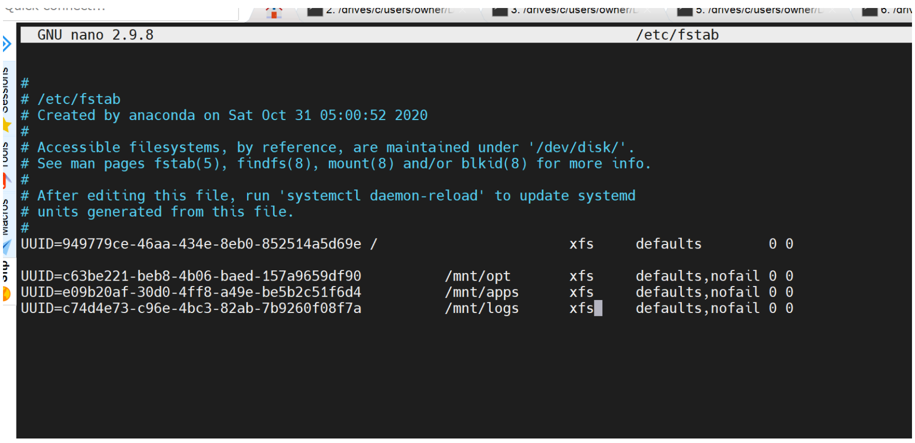
```
[    0.001005] APIC: Switch to symmetric I/O mode setup
[    0.008021] x2apic: IRQ remapping doesn't support X2APIC mode
[    0.016003] Switched APIC routing to physical flat.
[    0.023000] ..TIMER: vector=0x30 apic1=0 pin1=2 apic2=0 pin2=0
[    0.035005] clocksource: tsc-early: mask: 0xffffffffffffffff max_cycles: 0x212768636cb, max_idle_ns: 440795287765 ns
[    0.046004] Calibrating delay loop (skipped), value calculated using timer frequency.. 4600.10 BogoMIPS (lpj=2300054)
[    0.060003] pid_max: default: 32768 minimum: 301
[    0.066048] Security Framework initialized
[    0.072002] Yama: becoming mindful.
[    0.078007] SELinux:  Initializing.
[    0.084277] Dentry cache hash table entries: 131072 (order: 8, 1048576 bytes)
[    0.094106] Inode-cache hash table entries: 65536 (order: 7, 524288 bytes)
[    0.100016] Mount-cache hash table entries: 2048 (order: 2, 16384 bytes)
[    0.107005] Mountpoint-cache hash table entries: 2048 (order: 2, 16384 bytes)
[    0.114364] Last level iTLB entries: 4KB 64, 2MB 8, 4MB 8
[    0.119002] Last level dTLB entries: 4KB 64, 2MB 0, 4MB 0, 1GB 4
[    0.124003] FEATURE SPEC_CTRL Not Present
[    0.129002] FEATURE IBPB_SUPPORT Not Present
[    0.132001] Spectre V1 : Mitigation: usercopy/swapgs barriers and __user pointer sanitization
[    0.140002] Spectre V2 : Mitigation: Full generic retpoline
```

Then repeated the below steps:

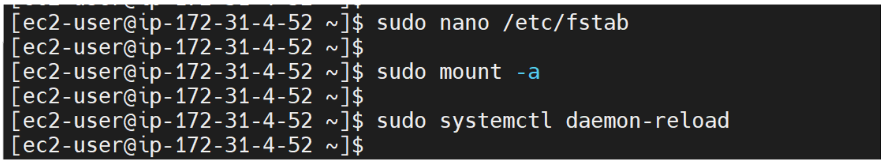
- Formatting the logical volumes with xfs file system and
- Installing and configuring NFS Server.



```
[ec2-user@ip-172-31-4-52 ~]$ df -h
Filesystem                      Size  Used Avail Use% Mounted on
devtmpfs                        378M     0  378M   0% /dev
tmpfs                           403M     0  403M   0% /dev/shm
tmpfs                           403M   11M  393M   3% /run
tmpfs                           403M     0  403M   0% /sys/fs/cgroup
/dev/xvda2                       10G  1.2G  8.9G  12% /
tmpfs                            81M     0   81M   0% /run/user/1000
/dev/mapper/webdata--vg-lv--opt 5.0G   68M  5.0G   2% /mnt/opt
/dev/mapper/webdata--vg-lv--apps 5.0G  68M  5.0G   2% /mnt/apps
/dev/mapper/webdata--vg-lv--logs 4.0G  61M  4.0G   2% /mnt/logs
[ec2-user@ip-172-31-4-52 ~]$
```


2. /drives/c/users/owner/
3. /drives/c/users/owner/
5. /drives/c/users/owner/
6. /driv

```
  GNU nano 2.9.8                                              /etc/fstab


#
# /etc/fstab
# Created by anaconda on Sat Oct 31 05:00:52 2020
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
UUID=949779ce-46aa-434e-8eb0-852514a5d69e /                       xfs     defaults        0 0

UUID=c63be221-beb8-4b06-baed-157a9659df90          /mnt/opt       xfs     defaults,nofail 0 0
UUID=e09b20af-30d0-4ff8-a49e-be5b2c51f6d4          /mnt/apps      xfs     defaults,nofail 0 0
UUID=c74d4e73-c96e-4bc3-82ab-7b9260f08f7a          /mnt/logs      xfs     defaults,nofail 0 0
```



```
[ec2-user@ip-172-31-4-52 ~]$ sudo nano /etc/fstab
[ec2-user@ip-172-31-4-52 ~]$
[ec2-user@ip-172-31-4-52 ~]$ sudo mount -a
[ec2-user@ip-172-31-4-52 ~]$
[ec2-user@ip-172-31-4-52 ~]$ sudo systemctl daemon-reload
[ec2-user@ip-172-31-4-52 ~]$
```

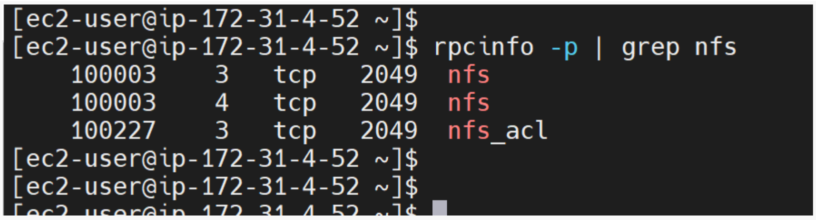
## Set up Firewall rules for NFS Services.

First install the firewall package: `sudo yum install firewalld`

```
sudo systemctl start firewalld
sudo systemctl enable firewalld
```

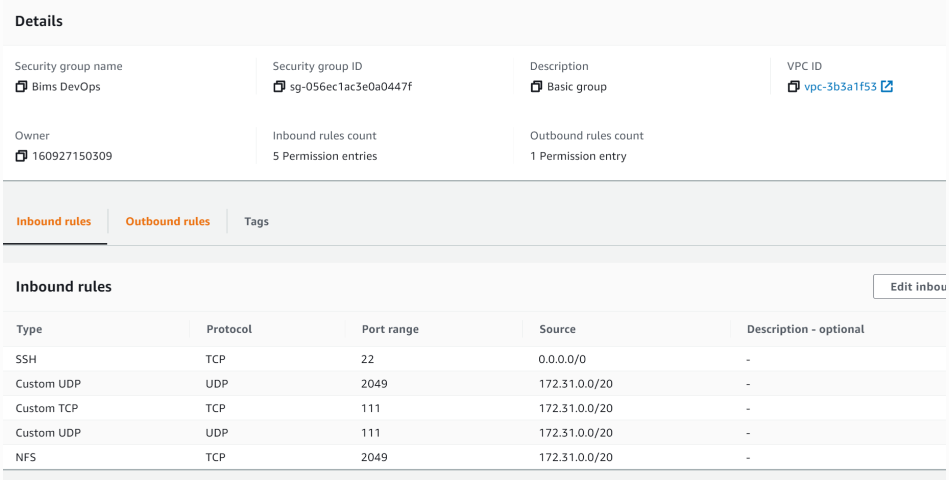
Check which port is used by NFS and open it using Security Groups (add new Inbound Rule)

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



```
[ec2-user@ip-172-31-4-52 ~]$
[ec2-user@ip-172-31-4-52 ~]$ rpcinfo -p | grep nfs
    100003    3   tcp   2049  nfs
    100003    4   tcp   2049  nfs
    100227    3   tcp   2049  nfs_acl
[ec2-user@ip-172-31-4-52 ~]$
[ec2-user@ip-172-31-4-52 ~]$
[ec2-user@ip-172-31-4-52 ~]$
```

**Important note:** In order for NFS server to be accessible from your client, you must also open following ports: TCP 111, UDP 111, UDP 2049



Security group name: Bims DevOps
Security group ID: sg-056ec1ac3e0a0447f
Description: Basic group
VPC ID: vpc-3b3a1f53
Owner: 160927150309
Inbound rules count: 5 Permission entries
Outbound rules count: 1 Permission entry

Inbound rules

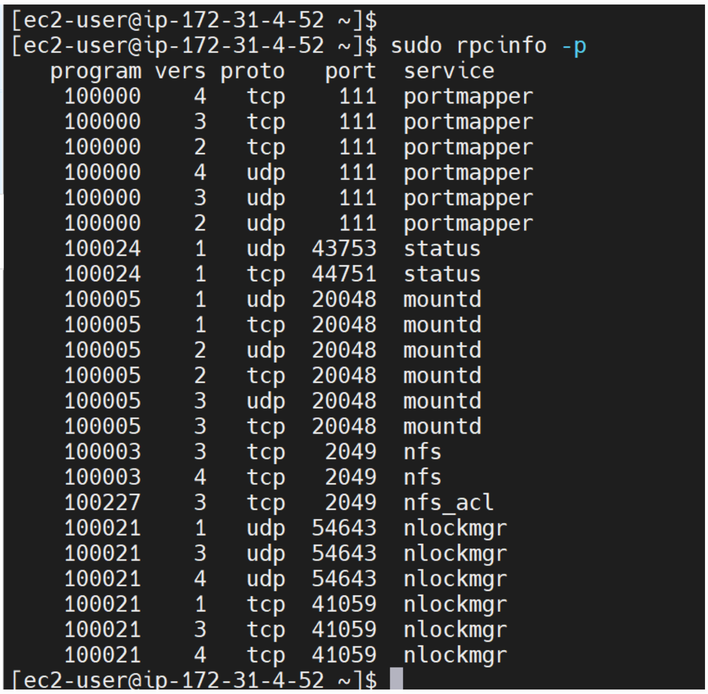
| Type | Protocol | Port range | Source | Description - optional |
|---|---|---|---|---|
| SSH | TCP | 22 | 0.0.0.0/0 | - |
| Custom UDP | UDP | 2049 | 172.31.0.0/20 | - |
| Custom TCP | TCP | 111 | 172.31.0.0/20 | - |
| Custom UDP | UDP | 111 | 172.31.0.0/20 | - |
| NFS | TCP | 2049 | 172.31.0.0/20 | - |

Alternatively, this can also be done through the terminal command line:

```
sudo firewall-cmd --permanent --zone public --add-service mountd

sudo firewall-cmd --permanent --zone public --add-service rpc-bind

sudo firewall-cmd --permanent --zone public --add-service nfs --permanent

sudo firewall-cmd --permanent --zone public --add-source=172.31.0.0/20 --permanent

sudo firewall-cmd --permanent --zone public --add-port=2049/tcp

sudo firewall-cmd --permanent --zone public --add-port=2049/udp

sudo firewall-cmd --reload
```

Confirm that the ports have been opened.

```
sudo rpcinfo -p
```



```
[ec2-user@ip-172-31-4-52 ~]$
[ec2-user@ip-172-31-4-52 ~]$ sudo rpcinfo -p
   program vers proto   port  service
    100000    4   tcp    111  portmapper
    100000    3   tcp    111  portmapper
    100000    2   tcp    111  portmapper
    100000    4   udp    111  portmapper
    100000    3   udp    111  portmapper
    100000    2   udp    111  portmapper
    100024    1   udp  43753  status
    100024    1   tcp  44751  status
    100005    1   udp  20048  mountd
    100005    1   tcp  20048  mountd
    100005    2   udp  20048  mountd
    100005    2   tcp  20048  mountd
    100005    3   udp  20048  mountd
    100005    3   tcp  20048  mountd
    100003    3   tcp   2049  nfs
    100003    4   tcp   2049  nfs
    100227    3   tcp   2049  nfs_acl
    100021    1   udp  54643  nlockmgr
    100021    3   udp  54643  nlockmgr
    100021    4   udp  54643  nlockmgr
    100021    1   tcp  41059  nlockmgr
    100021    3   tcp  41059  nlockmgr
    100021    4   tcp  41059  nlockmgr
[ec2-user@ip-172-31-4-52 ~]$
```

This command sudo 'rpcinfo -p' will list out specific ports based on the **RPC** (Remote procedure calls extend the capabilities of conventional procedure calls across a network and are essential in the development of distributed systems.) tool. head over to the AWS security group and enable inbound connections on the following ports.

## Step 3 – Configure the database server

The MySQL Database simply put, is a storage place for data that will be sent in by application end users. It helps to deploy cloud-native applications.

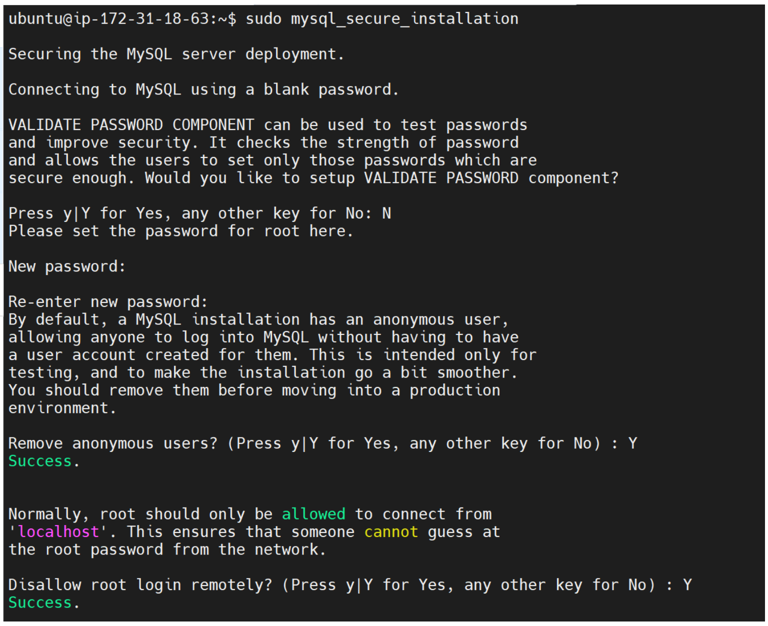
The steps involved in installing and configuring database include the below:

a. Launch an Ubuntu instance in AWS

b. Install MySQL server:
```
sudo apt install mysql-server -y
sudo systemctl start mysql
sudo systemctl enable mysql
```

c. Run the security script: `sudo mysql_secure_installation`

This allows the removal of a broad range of insecure default settings and also performs lockdown access to your database system. This script comes pre-installed with MySQL.



```
ubuntu@ip-172-31-18-63:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: N
Please set the password for root here.

New password:

Re-enter new password:
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : Y
Success.


Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : Y
Success.
```

You will be asked if you want to configure a VALIDATE PASSWORD PLUGIN database feature.

**Note**: This feature is something of a judgment call. Once enabled, passwords which do not match specified criteria will be rejected by MySQL with an error. It is safe to leave validation disabled; however, standard practice is to always use strong, unique passwords for database credentials.

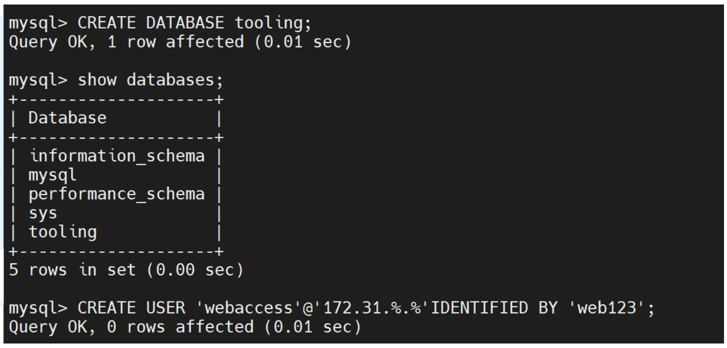
d. Create a database and name it tooling

```
mysql> CREATE DATABASE tooling;
```

```
msql> Show databases;
```

e. Create a new user account called 'webaccess' that will only connect to the remote host. On the database server, start the mysql console using the sudo mysql command then enter the following into the console;

```
mysql> CREATE USER 'webaccess'@'172.31.%.%'IDENTIFIED BY 'web123';
```



```
mysql> CREATE DATABASE tooling;
Query OK, 1 row affected (0.01 sec)

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

mysql> CREATE USER 'webaccess'@'172.31.%.%'IDENTIFIED BY 'web123';
Query OK, 0 rows affected (0.01 sec)
```

f. Confirm user has been created:

```
mysql> select user, host from mysql.user
```

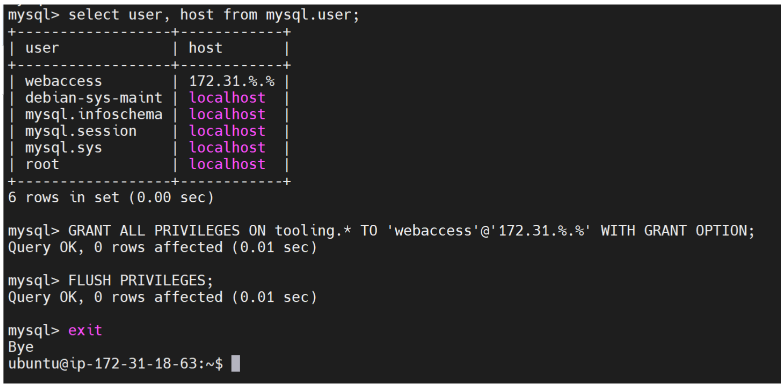
g. Grant privileges to 'webacces' user on 'tooling' database created;

```
GRANT ALL PRIVILEGES ON tooling.* TO 'webaccess'@'172.31.%.%' WITH GRANT OPTION;
```

h. It is good practice to run the 'FLUSH PRIVILEGES' command:

```
mysql> FLUSH PRIVILEGES;
```

i. Exit

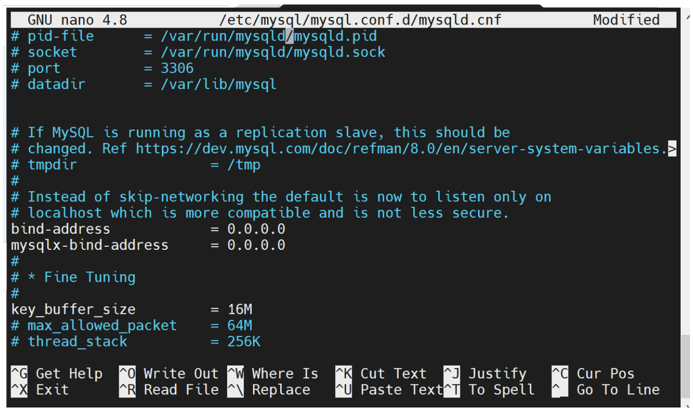


```
mysql> select user, host from mysql.user;
+------------------+-----------+
| user             | host      |
+------------------+-----------+
| webaccess        | 172.31.%.% |
| debian-sys-maint | localhost |
| mysql.infoschema | localhost |
| mysql.session    | localhost |
| mysql.sys        | localhost |
| root             | localhost |
+------------------+-----------+
6 rows in set (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON tooling.* TO 'webaccess'@'172.31.%.%' WITH GRANT OPTION;
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql> exit
Bye
ubuntu@ip-172-31-18-63:~$
```

j. Configure Mysql file to listen to web servers' IP address and modify the bind address in order to enable access to the database.

```
sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

The pre-configured 127.0.0.1, means that the server will only look for local connections. Change this directive to reference an external IP address by setting it to a wildcard IP address: either *, ::, or 0.0.0.0:

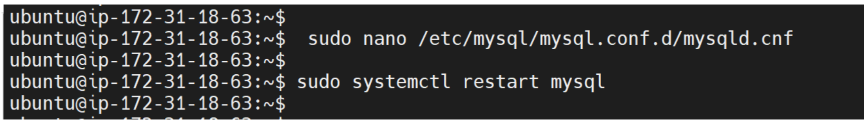


```
  GNU nano 4.8              /etc/mysql/mysql.conf.d/mysqld.cnf             Modified
# pid-file        = /var/run/mysqld/mysqld.pid
# socket          = /var/run/mysqld/mysqld.sock
# port            = 3306
# datadir         = /var/lib/mysql


# 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.>
# 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

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text^T To Spell    ^_ Go To Line
```

Save and close: CTRLX; Y; ENTER.

k. Restart mysql service in order to make the changes effective@

sudo systemctl restart mysql

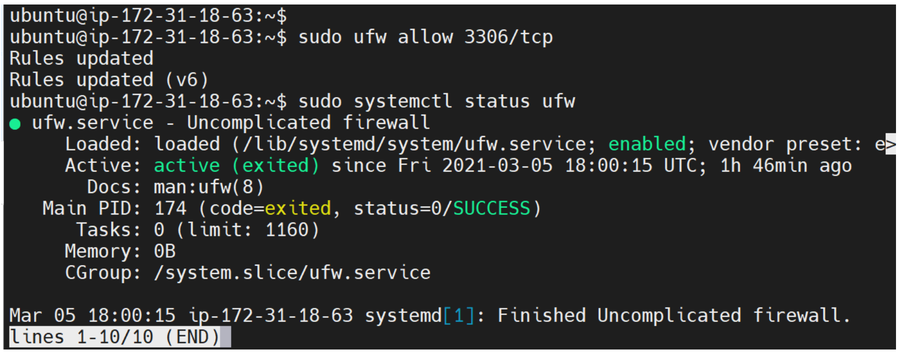


```
ubuntu@ip-172-31-18-63:~$
ubuntu@ip-172-31-18-63:~$  sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
ubuntu@ip-172-31-18-63:~$
ubuntu@ip-172-31-18-63:~$ sudo systemctl restart mysql
ubuntu@ip-172-31-18-63:~$
```

l. Open firewall to llow incoming traffic on port 3306:

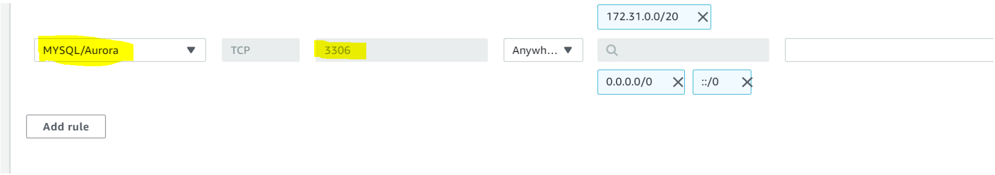
sudo ufw allow 3306/tcp

m. Check status: sudo systemctl status ufw



```
ubuntu@ip-172-31-18-63:~$
ubuntu@ip-172-31-18-63:~$ sudo ufw allow 3306/tcp
Rules updated
Rules updated (v6)
ubuntu@ip-172-31-18-63:~$ sudo systemctl status ufw
● ufw.service - Uncomplicated firewall
     Loaded: loaded (/lib/systemd/system/ufw.service; enabled; vendor preset: e>
     Active: active (exited) since Fri 2021-03-05 18:00:15 UTC; 1h 46min ago
       Docs: man:ufw(8)
   Main PID: 174 (code=exited, status=0/SUCCESS)
      Tasks: 0 (limit: 1160)
     Memory: 0B
     CGroup: /system.slice/ufw.service

Mar 05 18:00:15 ip-172-31-18-63 systemd[1]: Finished Uncomplicated firewall.
lines 1-10/10 (END)
```

n. Add an inbound rule on port 3306 for our AWS security group since the MySQL service listens on that port: MySQL/Aurora



MYSQL/Aurora ▼ TCP 3306 Anywh... ▼

172.31.0.0/20 ✕

0.0.0.0/0 ✕ ::/0 ✕

## Step 4 – Prepare the Web Servers

IP addresses of the servers:

```
NFS Server - 172.31.4.52
Database Server - 172.31.18.63
Web Server 1 - 172.31.16.162
Web server 2 - 172.31.16.41
Web server 3 - 172.31.30.14
```

We need to make sure that our Web Servers can serve the same content from shared storage solutions, in our case - NFS Server and MySQL database. You already know that one DB can be accessed for reads and writes by multiple clients. For storing shared files that our Web Servers will use - we will utilize NFS and mount previously created Logical Volume lv-apps to the folder where Apache stores files to be served to the users (/var/www).

This approach will make our Web Servers stateless, which means we will be able to add new ones or remove them whenever we need, and the integrity of the data (in the database and on NFS) will be preserved.

You can set up one or more Web Servers and point to the same NFS and connect to the same Database Server. During this step we will configure NFS client and deploy a tooling application to our Web Server(s).

**Steps:**

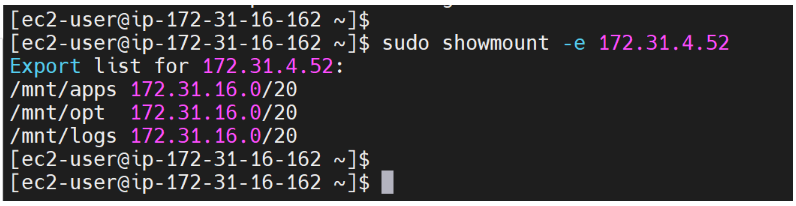
a.
b. Launch a new EC2 instance with RHEL 8 Operating System
c. Install NFS client

The web servers will make use of the NFS server as a backend storage. Based on this requirement, we shall configure the web servers as NFS clients. On instantiating the three (3) instances on the AWS console, we shall run the following commands on each of the servers.This will make necessary installation of the nfs client services.

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

You can view the nfs share that is to be mounted:

```
sudo showmount -e 172.31.4.52
```



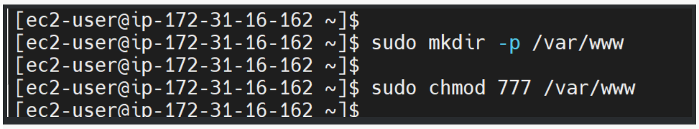
```
[ec2-user@ip-172-31-16-162 ~]$
[ec2-user@ip-172-31-16-162 ~]$ sudo showmount -e 172.31.4.52
Export list for 172.31.4.52:
/mnt/apps 172.31.16.0/20
/mnt/opt  172.31.16.0/20
/mnt/logs 172.31.16.0/20
[ec2-user@ip-172-31-16-162 ~]$
[ec2-user@ip-172-31-16-162 ~]$
```

d. Create mount directory on the web server

```
sudo mkdir -p /var/www
```

e. Grant necessary permission on the web server

```
sudo chmod 777 /var/www
```



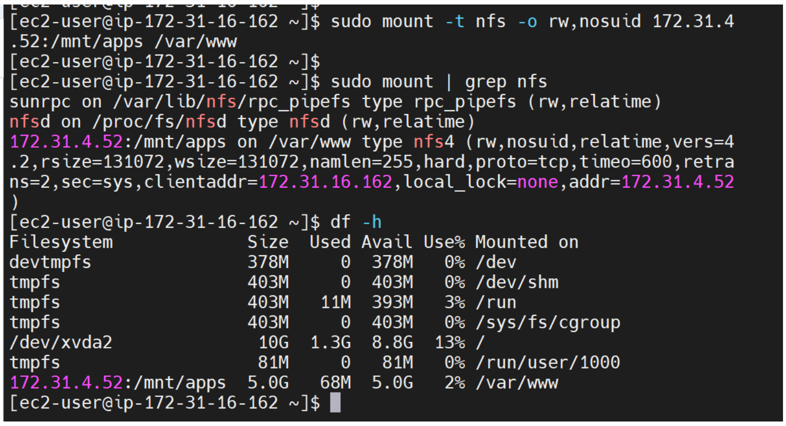
```
[ec2-user@ip-172-31-16-162 ~]$
[ec2-user@ip-172-31-16-162 ~]$ sudo mkdir -p /var/www
[ec2-user@ip-172-31-16-162 ~]$
[ec2-user@ip-172-31-16-162 ~]$ sudo chmod 777 /var/www
[ec2-user@ip-172-31-16-162 ~]$
```

f. Mount the NFS Share on the /var/www directory

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

sudo mount -t nfs -o rw,nosuid 172.31.4.52:/mnt/apps /var/www
```

g. Verify successful mount

```
df -h  OR sudo mount | grep nfs
```



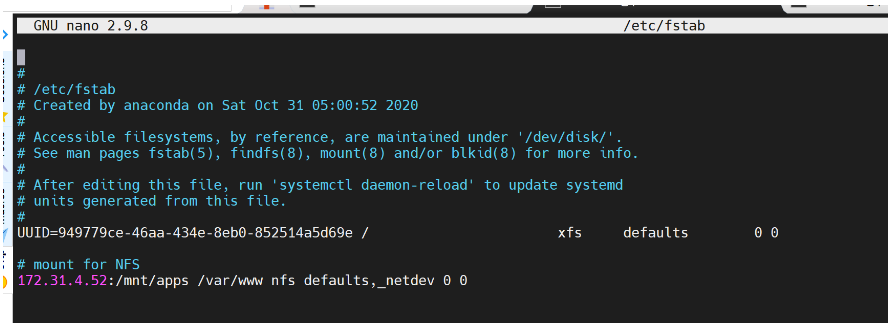
```
[ec2-user@ip-172-31-16-162 ~]$ sudo mount -t nfs -o rw,nosuid 172.31.4
.52:/mnt/apps /var/www
[ec2-user@ip-172-31-16-162 ~]$
[ec2-user@ip-172-31-16-162 ~]$ sudo mount | grep nfs
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
nfsd on /proc/fs/nfsd type nfsd (rw,relatime)
172.31.4.52:/mnt/apps on /var/www type nfs4 (rw,nosuid,relatime,vers=4
.2,rsize=131072,wsize=131072,namlen=255,hard,proto=tcp,timeo=600,retra
ns=2,sec=sys,clientaddr=172.31.16.162,local_lock=none,addr=172.31.4.52
)
[ec2-user@ip-172-31-16-162 ~]$ df -h
Filesystem             Size  Used Avail Use% Mounted on
devtmpfs               378M     0  378M   0% /dev
tmpfs                  403M     0  403M   0% /dev/shm
tmpfs                  403M   11M  393M   3% /run
tmpfs                  403M     0  403M   0% /sys/fs/cgroup
/dev/xvda2              10G  1.3G  8.8G  13% /
tmpfs                   81M     0   81M   0% /run/user/1000
172.31.4.52:/mnt/apps  5.0G   68M  5.0G   2% /var/www
[ec2-user@ip-172-31-16-162 ~]$
```

h. Make the mount persistent by updating the /etc/fstab file on all web servers:

```
sudo nano /etc/fstab
```

i. Add the below in the file:

```
172.31.4.52:/mnt/apps /var/www nfs defaults,_netdev 0 0
```



```
  GNU nano 2.9.8                                                          /etc/fstab

#
# /etc/fstab
# Created by anaconda on Sat Oct 31 05:00:52 2020
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
UUID=949779ce-46aa-434e-8eb0-852514a5d69e /                       xfs     defaults        0 0

# mount for NFS
172.31.4.52:/mnt/apps /var/www nfs defaults,_netdev 0 0
```

From above, the '_netdev' option will help prevent the server from hanging during a scenario where the nfs share is not present and the web servers are booting up.

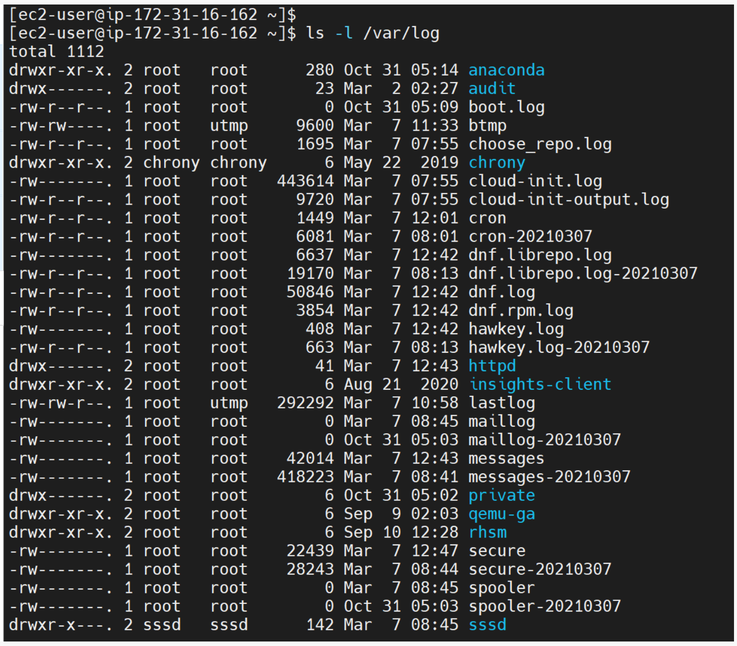
Save and close.

j. **Install Apache** on the web servers. This will help to serve the web content:

```
sudo yum install httpd -y
sudo systemctl start httpd
sudo systemctl enable httpd
```

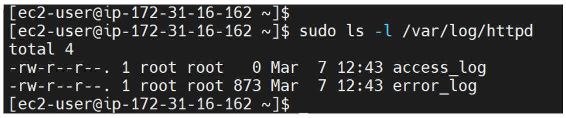
k. Locate the log folder for Apache on the Web Server and mount it to NFS server's export for logs.

```
ls -l /var/log
```



```
[ec2-user@ip-172-31-16-162 ~]$
[ec2-user@ip-172-31-16-162 ~]$ ls -l /var/log
total 1112
drwxr-xr-x. 2 root   root      280 Oct 31 05:14 anaconda
drwx------. 2 root   root       23 Mar  2 02:27 audit
-rw-r--r--. 1 root   root        0 Oct 31 05:09 boot.log
-rw-rw----. 1 root   utmp     9600 Mar  7 11:33 btmp
-rw-r--r--. 1 root   root     1695 Mar  7 07:55 choose_repo.log
drwxr-xr-x. 2 chrony chrony      6 May 22  2019 chrony
-rw-------. 1 root   root   443614 Mar  7 07:55 cloud-init.log
-rw-r--r--. 1 root   root     9720 Mar  7 07:55 cloud-init-output.log
-rw-r--r--. 1 root   root     1449 Mar  7 12:01 cron
-rw-r--r--. 1 root   root     6081 Mar  7 08:01 cron-20210307
-rw-------. 1 root   root     6637 Mar  7 12:42 dnf.librepo.log
-rw-r--r--. 1 root   root    19170 Mar  7 08:13 dnf.librepo.log-20210307
-rw-r--r--. 1 root   root    50846 Mar  7 12:42 dnf.log
-rw-r--r--. 1 root   root     3854 Mar  7 12:42 dnf.rpm.log
-rw-------. 1 root   root      408 Mar  7 12:42 hawkey.log
-rw-r--r--. 1 root   root      663 Mar  7 08:13 hawkey.log-20210307
drwx------. 2 root   root       41 Mar  7 12:43 httpd
drwxr-xr-x. 2 root   root        6 Aug 21  2020 insights-client
-rw-rw-r--. 1 root   utmp   292292 Mar  7 10:58 lastlog
-rw-------. 1 root   root        0 Mar  7 08:45 maillog
-rw-------. 1 root   root        0 Oct 31 05:03 maillog-20210307
-rw-------. 1 root   root    42014 Mar  7 12:43 messages
-rw-------. 1 root   root   418223 Mar  7 08:41 messages-20210307
drwx------. 2 root   root        6 Oct 31 05:02 private
drwxr-xr-x. 2 root   root        6 Sep  9 02:03 qemu-ga
drwxr-xr-x. 2 root   root        6 Sep 10 12:28 rhsm
-rw-------. 1 root   root    22439 Mar  7 12:47 secure
-rw-------. 1 root   root    28243 Mar  7 08:44 secure-20210307
-rw-------. 1 root   root        0 Mar  7 08:45 spooler
-rw-------. 1 root   root        0 Oct 31 05:03 spooler-20210307
drwxr-x---. 2 sssd   sssd      142 Mar  7 08:45 sssd
```

Log folder for apache: sudo  ls -l /var/log/httpd



```
[ec2-user@ip-172-31-16-162 ~]$
[ec2-user@ip-172-31-16-162 ~]$ sudo ls -l /var/log/httpd
total 4
-rw-r--r--. 1 root root   0 Mar  7 12:43 access_log
-rw-r--r--. 1 root root 873 Mar  7 12:43 error_log
[ec2-user@ip-172-31-16-162 ~]$ _
```

Create a backup file for log

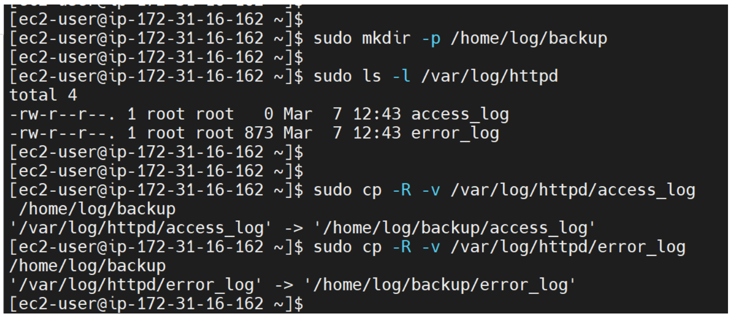
```
sudo mkdir -p /home/log/backup
```

Copy the log file to the backup file

View the file: `ls -l /var/log/httpd`

Then copy:

```
sudo cp -R -v /var/log/httpd/access_log /home/log/backup
sudo cp -R -v /var/log/httpd/error_log /home/log/backup
```



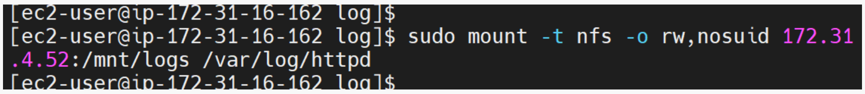
```
[ec2-user@ip-172-31-16-162 ~]$
[ec2-user@ip-172-31-16-162 ~]$ sudo mkdir -p /home/log/backup
[ec2-user@ip-172-31-16-162 ~]$
[ec2-user@ip-172-31-16-162 ~]$ sudo ls -l /var/log/httpd
total 4
-rw-r--r--. 1 root root   0 Mar  7 12:43 access_log
-rw-r--r--. 1 root root 873 Mar  7 12:43 error_log
[ec2-user@ip-172-31-16-162 ~]$
[ec2-user@ip-172-31-16-162 ~]$
[ec2-user@ip-172-31-16-162 ~]$ sudo cp -R -v /var/log/httpd/access_log
 /home/log/backup
'/var/log/httpd/access_log' -> '/home/log/backup/access_log'
[ec2-user@ip-172-31-16-162 ~]$ sudo cp -R -v /var/log/httpd/error_log
/home/log/backup
'/var/log/httpd/error_log' -> '/home/log/backup/error_log'
[ec2-user@ip-172-31-16-162 ~]$
```

*Please note:Change ownership of the log file. If after mounting, the ownership is changed to apache and mode is changed to 777 as per nfs server, there will be a need to change ownership to 700 as a standard rule and also change back to root:root because httpd is meant for root only.

```
sudo chown -R root:root /var/log/httpd
sudo chmod 700 /var/log/httpd
```

Mount /var/log:

```
sudo mount -t nfs -o rw,nosuid <NFS-Server-Private-IP-Address>:/mnt/logs /var/logs
sudo mount -t nfs -o rw,nosuid 172.31.4.52:/mnt/logs /var/log/httpd
```

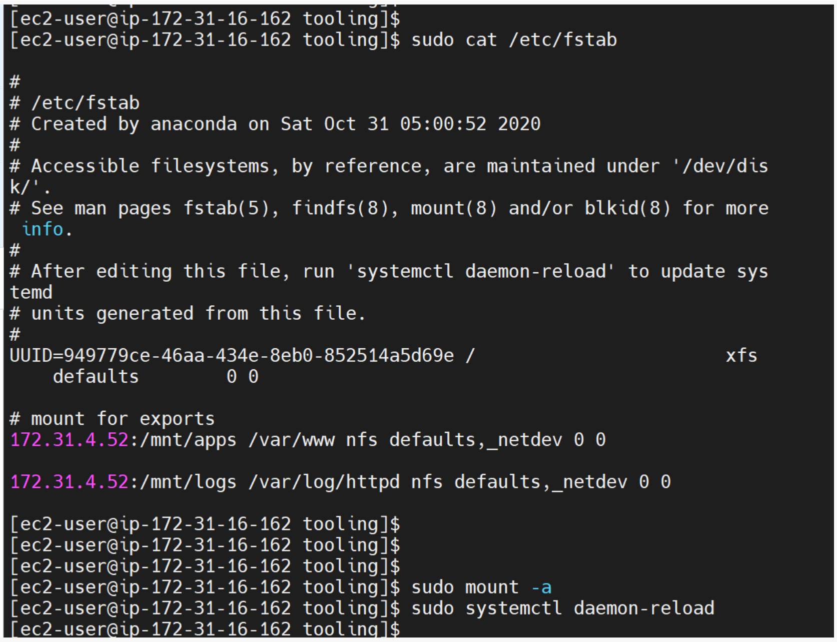


```
[ec2-user@ip-172-31-16-162 log]$
[ec2-user@ip-172-31-16-162 log]$ sudo mount -t nfs -o rw,nosuid 172.31
.4.52:/mnt/logs /var/log/httpd
[ec2-user@ip-172-31-16-162 log]$
```

Make the mount persistent by updating the fstab file with the below:

```
172.31.4.52:/mnt/logs /var/log/httpd nfs defaults,_netdev 0 0
```

```
sudo nano /etc/fstab
```



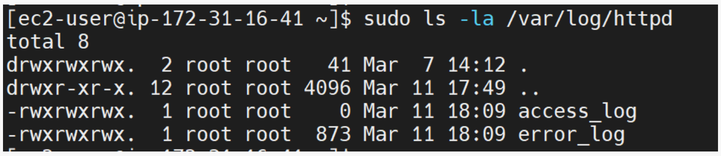
```
[ec2-user@ip-172-31-16-162 tooling]$
[ec2-user@ip-172-31-16-162 tooling]$ sudo cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Sat Oct 31 05:00:52 2020
#
# Accessible filesystems, by reference, are maintained under '/dev/dis
k/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more
 info.
#
# After editing this file, run 'systemctl daemon-reload' to update sys
temd
# units generated from this file.
#
UUID=949779ce-46aa-434e-8eb0-852514a5d69e /                       xfs
    defaults        0 0

# mount for exports
172.31.4.52:/mnt/apps /var/www nfs defaults,_netdev 0 0

172.31.4.52:/mnt/logs /var/log/httpd nfs defaults,_netdev 0 0

[ec2-user@ip-172-31-16-162 tooling]$
[ec2-user@ip-172-31-16-162 tooling]$
[ec2-user@ip-172-31-16-162 tooling]$
[ec2-user@ip-172-31-16-162 tooling]$ sudo mount -a
[ec2-user@ip-172-31-16-162 tooling]$ sudo systemctl daemon-reload
[ec2-user@ip-172-31-16-162 tooling]$
```

Then copy from the back up back into the /var/log/httpd file after the mount.

```
sudo cp -R -v /home/log/backup/. /var/log/httpd
```

View the log file

```
sudo ls -la /var/log/httpd
```



```
[ec2-user@ip-172-31-16-41 ~]$ sudo ls -la /var/log/httpd
total 8
drwxrwxrwx.  2 root root   41 Mar  7 14:12 .
drwxr-xr-x. 12 root root 4096 Mar 11 17:49 ..
-rwxrwxrwx.  1 root root    0 Mar 11 18:09 access_log
-rwxrwxrwx.  1 root root  873 Mar 11 18:09 error_log
```

## Install PHP

There is a need to install the latest PHP version. PHP is the component of our setup that will process code to display dynamic content. It can run scripts, connect to our MySQL databases to get information and hand the processed content over to our web server to display.

Right off the bat, you need to enable the EPEL repository on your system. EPEL, Short for Extra Packages for Enterprise Linux, is an effort from the Fedora team t provides a set of additional packages that are not present by default on RHEL & CentOS.

- To install **PHP 7**, you have to install and enable **EPEL and Remi repository** on your

7 system with the commands below.

```
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
```

- install yum-utils, a collection of useful programs for managing yum repositories a packages. It has tools that basically extend yum's default features.It can be usec managing (enabling or disabling) yum repositories as well as packages without any configuration and so much more.

```
sudo yum install yum-utils
```

- After the successful installation of yum-utils and Remi-packages, search for the PHP modules which are available for download by running the command

```
sudo dnf module list php
```

- The output indicates that the currently installed version of PHP is PHP 7.2. To install the newer release, PHP 7.4, reset the PHP modules.

```
sudo dnf module reset php
```

- Having reset the PHP modules, enable the PHP 7.4 module by running

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

- Finally, install **PHP**, **PHP-FPM** (FastCGI Process Manager) and associated PHP modules using the command.

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

- Verify the version installed

```
php -v
```

- Now PHP 7.4 has now been installed, we will now start and enable PHP-FPM on boot-up.

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

- Check status

```
sudo systemctl status php-fpm
```

- double check the installed version of PHP on your system.

```
sudo php -v
```

- To instruct **SELinux** to allow **Apache** to execute the **PHP** code via PHP-FPM run

```
setsebool -P httpd_execmem 1
sudo setenforce permissive
```

- Finally, restart Apache web server for PHP to work with Apache web server.

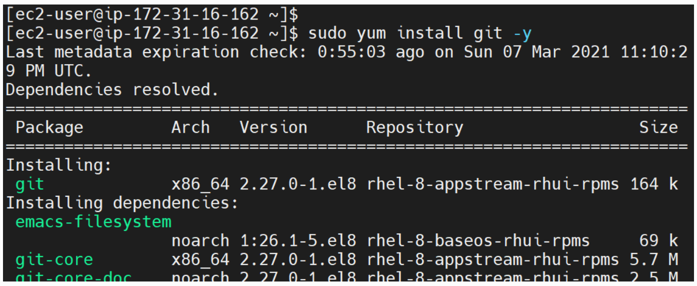
```
sudo systemctl restart httpd
```

Fork the tooling source code from Darey.io Github Account to your Github account. Deploy the tooling website's code to the Web Server. Ensure that the **html** folder from the repository is deployed to /var/www/html

## Install Git:

To get access to the codebase, we'll need to install git. Run the following in the terminal to install git (this should be done on the nfs server)

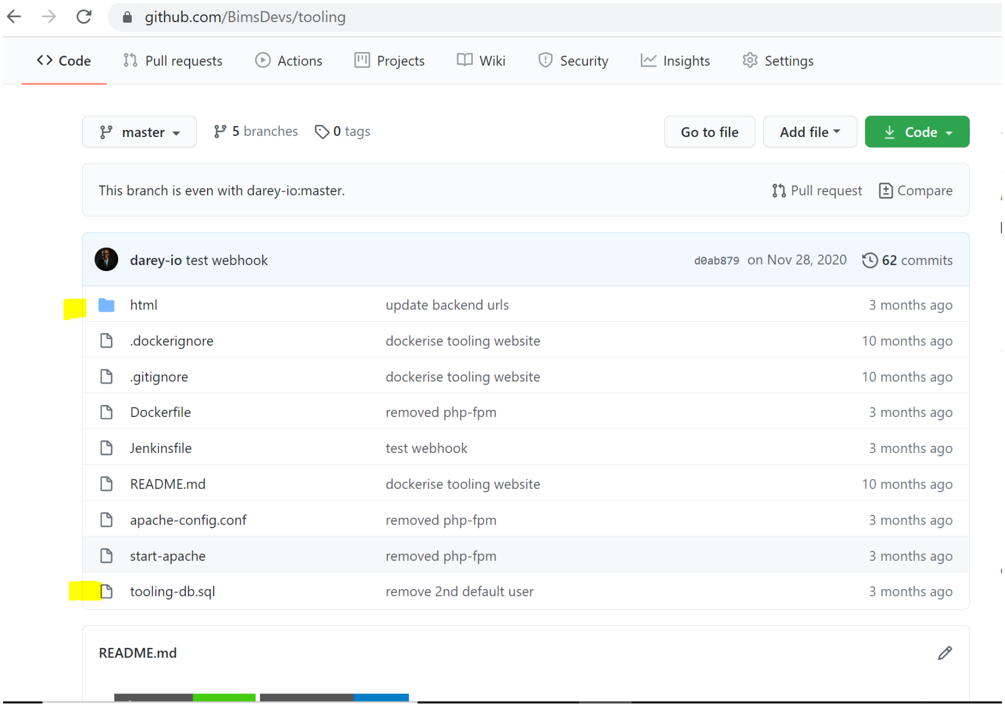
```
sudo yum install git -y
```



```
[ec2-user@ip-172-31-16-162 ~]$
[ec2-user@ip-172-31-16-162 ~]$ sudo yum install git -y
Last metadata expiration check: 0:55:03 ago on Sun 07 Mar 2021 11:10:2
9 PM UTC.
Dependencies resolved.
======================================================================
 Package          Arch   Version      Repository                  Size
======================================================================
Installing:
 git              x86_64 2.27.0-1.el8 rhel-8-appstream-rhui-rpms 164 k
Installing dependencies:
 emacs-filesystem
                  noarch 1:26.1-5.el8 rhel-8-baseos-rhui-rpms     69 k
 git-core         x86_64 2.27.0-1.el8 rhel-8-appstream-rhui-rpms 5.7 M
 git-core-doc     noarch 2.27.0-1.el8 rhel-8-appstream-rhui-rpms 2.5 M
```

Clone the repository for the tooling website;

```
git clone https://github.com/BimsDevs/tooling.git
```



This branch is even with darey-io:master.

darey-io test webhook d0ab879 on Nov 28, 2020 62 commits

| | | |
|---|---|---|
| html | update backend urls | 3 months ago |
| .dockerignore | dockerise tooling website | 10 months ago |
| .gitignore | dockerise tooling website | 10 months ago |
| Dockerfile | removed php-fpm | 3 months ago |
| Jenkinsfile | test webhook | 3 months ago |
| README.md | dockerise tooling website | 10 months ago |
| apache-config.conf | removed php-fpm | 3 months ago |
| start-apache | removed php-fpm | 3 months ago |
| tooling-db.sql | remove 2nd default user | 3 months ago |

cd into the tooling directory and recursively copy the contents of the html directory into /var/www/html

```
      cd tooling ; sudo cp -R html/. /var/www/html
```

Verify successful copying:

```
      sudo ls -la /var/www/html
```

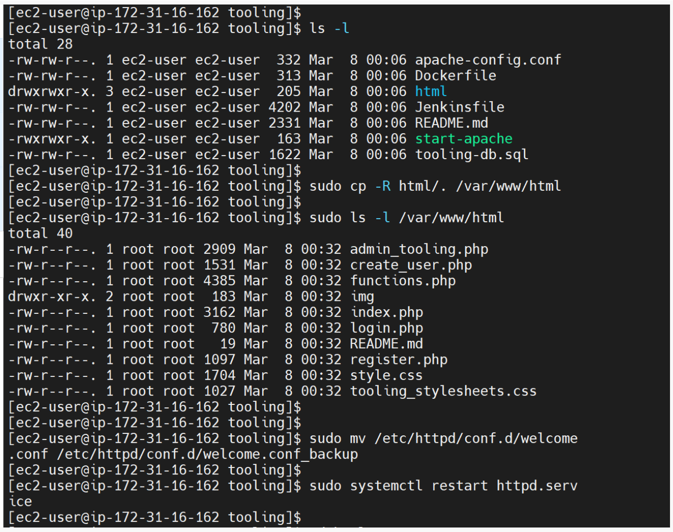
Change directory to /var/www/html

```
      cd /var/www/html
```

Rename the Apache default page:

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

Restart httpd.service: `sudo systemctl restart httpd.service`



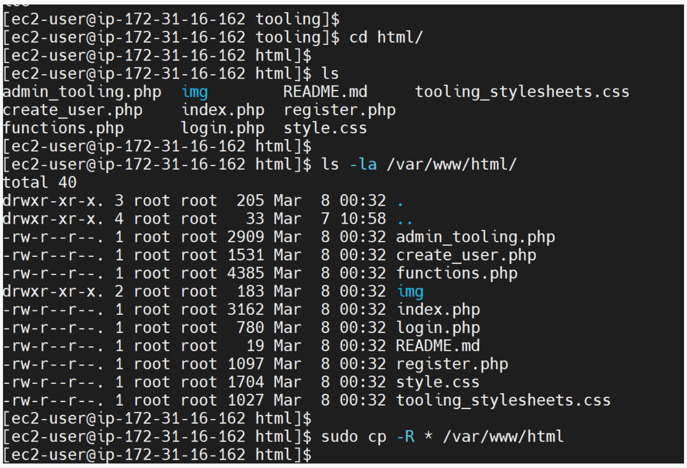
```
[ec2-user@ip-172-31-16-162 tooling]$
[ec2-user@ip-172-31-16-162 tooling]$ ls -l
total 28
-rw-rw-r--. 1 ec2-user ec2-user  332 Mar  8 00:06 apache-config.conf
-rw-rw-r--. 1 ec2-user ec2-user  313 Mar  8 00:06 Dockerfile
drwxrwxr-x. 3 ec2-user ec2-user  205 Mar  8 00:06 html
-rw-rw-r--. 1 ec2-user ec2-user 4202 Mar  8 00:06 Jenkinsfile
-rw-rw-r--. 1 ec2-user ec2-user 2331 Mar  8 00:06 README.md
-rwxrwxr-x. 1 ec2-user ec2-user  163 Mar  8 00:06 start-apache
-rw-rw-r--. 1 ec2-user ec2-user 1622 Mar  8 00:06 tooling-db.sql
[ec2-user@ip-172-31-16-162 tooling]$
[ec2-user@ip-172-31-16-162 tooling]$ sudo cp -R html/. /var/www/html
[ec2-user@ip-172-31-16-162 tooling]$
[ec2-user@ip-172-31-16-162 tooling]$ sudo ls -l /var/www/html
total 40
-rw-r--r--. 1 root root 2909 Mar  8 00:32 admin_tooling.php
-rw-r--r--. 1 root root 1531 Mar  8 00:32 create_user.php
-rw-r--r--. 1 root root 4385 Mar  8 00:32 functions.php
drwxr-xr-x. 2 root root  183 Mar  8 00:32 img
-rw-r--r--. 1 root root 3162 Mar  8 00:32 index.php
-rw-r--r--. 1 root root  780 Mar  8 00:32 login.php
-rw-r--r--. 1 root root   19 Mar  8 00:32 README.md
-rw-r--r--. 1 root root 1097 Mar  8 00:32 register.php
-rw-r--r--. 1 root root 1704 Mar  8 00:32 style.css
-rw-r--r--. 1 root root 1027 Mar  8 00:32 tooling_stylesheets.css
[ec2-user@ip-172-31-16-162 tooling]$
[ec2-user@ip-172-31-16-162 tooling]$
[ec2-user@ip-172-31-16-162 tooling]$ sudo mv /etc/httpd/conf.d/welcome
.conf /etc/httpd/conf.d/welcome.conf_backup
[ec2-user@ip-172-31-16-162 tooling]$
[ec2-user@ip-172-31-16-162 tooling]$ sudo systemctl restart httpd.serv
ice
[ec2-user@ip-172-31-16-162 tooling]$
```

cd into html: `cd /html`

Deploy the entire html folder into /var/www/html;

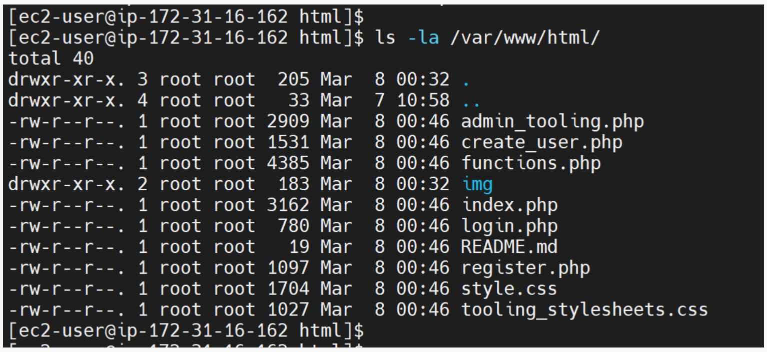
cd into html: `cd /html`

`sudo cp -R * /var/www/html`



```
[ec2-user@ip-172-31-16-162 tooling]$
[ec2-user@ip-172-31-16-162 tooling]$ cd html/
[ec2-user@ip-172-31-16-162 html]$
[ec2-user@ip-172-31-16-162 html]$ ls
admin_tooling.php  img        README.md     tooling_stylesheets.css
create_user.php    index.php  register.php
functions.php      login.php  style.css
[ec2-user@ip-172-31-16-162 html]$
[ec2-user@ip-172-31-16-162 html]$ ls -la /var/www/html/
total 40
drwxr-xr-x. 3 root root  205 Mar  8 00:32 .
drwxr-xr-x. 4 root root   33 Mar  7 10:58 ..
-rw-r--r--. 1 root root 2909 Mar  8 00:32 admin_tooling.php
-rw-r--r--. 1 root root 1531 Mar  8 00:32 create_user.php
-rw-r--r--. 1 root root 4385 Mar  8 00:32 functions.php
drwxr-xr-x. 2 root root  183 Mar  8 00:32 img
-rw-r--r--. 1 root root 3162 Mar  8 00:32 index.php
-rw-r--r--. 1 root root  780 Mar  8 00:32 login.php
-rw-r--r--. 1 root root   19 Mar  8 00:32 README.md
-rw-r--r--. 1 root root 1097 Mar  8 00:32 register.php
-rw-r--r--. 1 root root 1704 Mar  8 00:32 style.css
-rw-r--r--. 1 root root 1027 Mar  8 00:32 tooling_stylesheets.css
[ec2-user@ip-172-31-16-162 html]$
[ec2-user@ip-172-31-16-162 html]$ sudo cp -R * /var/www/html
[ec2-user@ip-172-31-16-162 html]$
```

List the files in the html folder to confirm its contents also has functions.php. `ls -la`



```
[ec2-user@ip-172-31-16-162 html]$
[ec2-user@ip-172-31-16-162 html]$ ls -la /var/www/html/
total 40
drwxr-xr-x. 3 root root  205 Mar  8 00:32 .
drwxr-xr-x. 4 root root   33 Mar  7 10:58 ..
-rw-r--r--. 1 root root 2909 Mar  8 00:46 admin_tooling.php
-rw-r--r--. 1 root root 1531 Mar  8 00:46 create_user.php
-rw-r--r--. 1 root root 4385 Mar  8 00:46 functions.php
drwxr-xr-x. 2 root root  183 Mar  8 00:32 img
-rw-r--r--. 1 root root 3162 Mar  8 00:46 index.php
-rw-r--r--. 1 root root  780 Mar  8 00:46 login.php
-rw-r--r--. 1 root root   19 Mar  8 00:46 README.md
-rw-r--r--. 1 root root 1097 Mar  8 00:46 register.php
-rw-r--r--. 1 root root 1704 Mar  8 00:46 style.css
-rw-r--r--. 1 root root 1027 Mar  8 00:46 tooling_stylesheets.css
[ec2-user@ip-172-31-16-162 html]$
```

## Configure Firewall rules on all web servers:

Open TCP port 80 on aws for all the Web Servers by selecting http in the security section of aws console /instance. We can also allow traffic from MySQL by selecting MySQL Aurora for port 3306. This will allow traffic from http and MySQL.

We can run the command below w=in the terminal.

```
sudo firewall-cmd --permanent --add-service=https
sudo firewall-cmd --permanent --add-service=http
sudo firewall-cmd --permanent --add-service=mysql
sudo firewall-cmd --reload
```

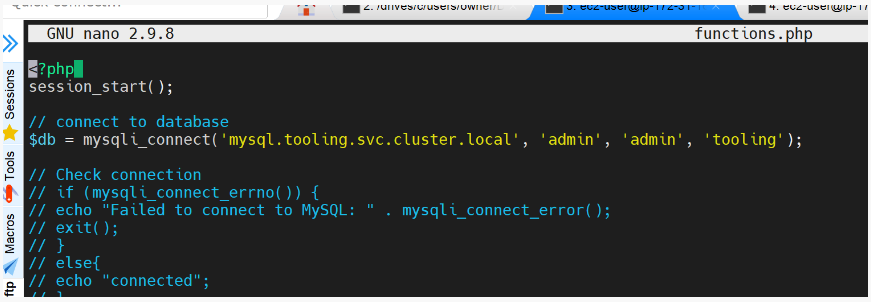
## Configure Apache / website to serve PHP.

Update the website's configuration to connect to the database (in functions.php file) and by applying the tooling-db.sql script.

As the Apache default page has already been renamed in the section above, and we have verified that the 'functions.php' file exists in the var/www/html directory, then we will need to update the tooling website credentials in the function.php folder.

```
sudo nano functions.php
```

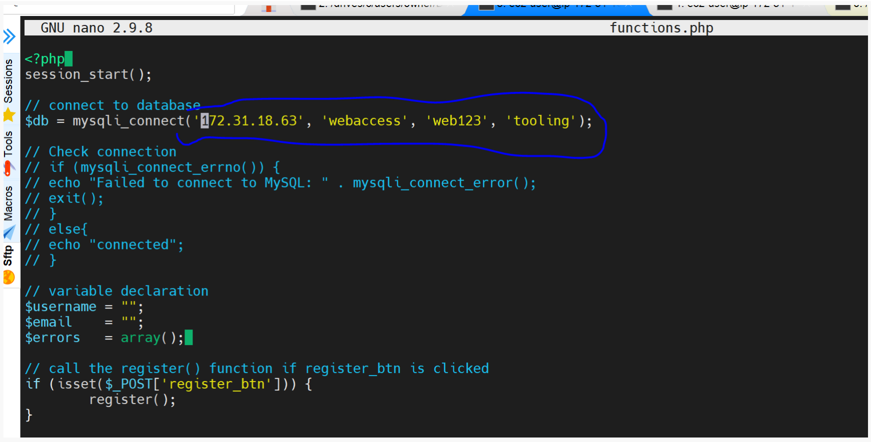
Pre-update:



```
  GNU nano 2.9.8                                              functions.php

<?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";
```

Post update: with the database details and IP - This will set the database credentials to be used when connecting to the tooling database.

```
$db = mysqli_connect('172.31.18.63', 'webaccess', 'web123', 'tooling');
```



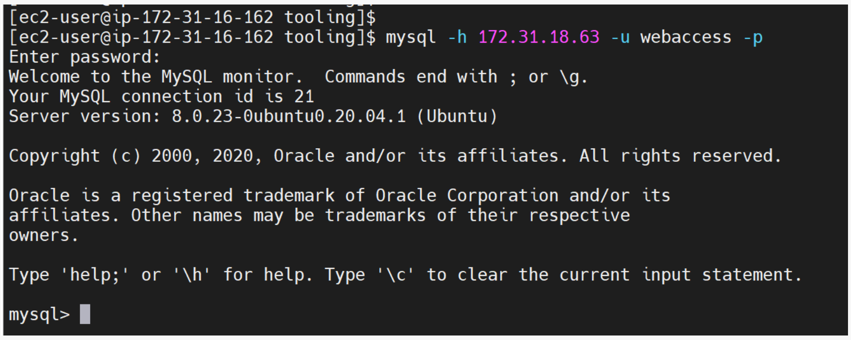
```
GNU nano 2.9.8                                      functions.php

<?php
session_start();

// connect to database
$db = mysqli_connect('172.31.18.63', 'webaccess', 'web123', '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();
}
```

## Install MySQL on the web server

```
sudo yum install mysql -y
mysql_secure_installation
```

Verify that the web server can access the database:

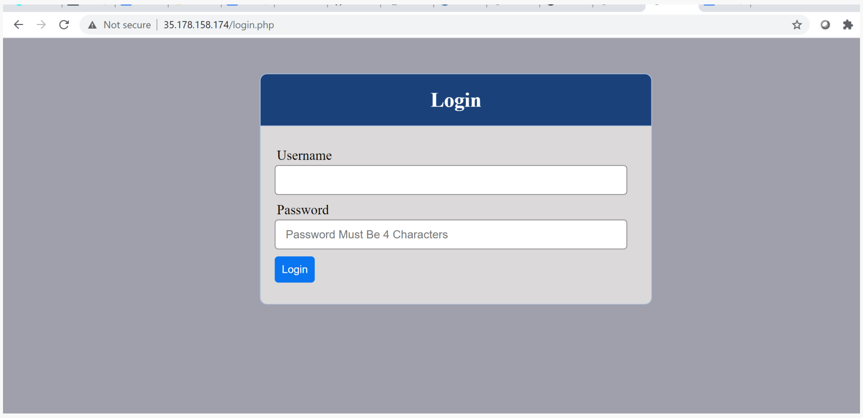
```
mysql -h 172.31.18.63 -u webaccess -p
```



```
[ec2-user@ip-172-31-16-162 tooling]$
[ec2-user@ip-172-31-16-162 tooling]$ mysql -h 172.31.18.63 -u webaccess -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 21
Server version: 8.0.23-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

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> 
```

Restart the httpd service: `sudo systemctl restart httpd.service`

Then test the tooling website on the browser by typing the IP address of the web server on the browser: http://35.178.158.174/login.php



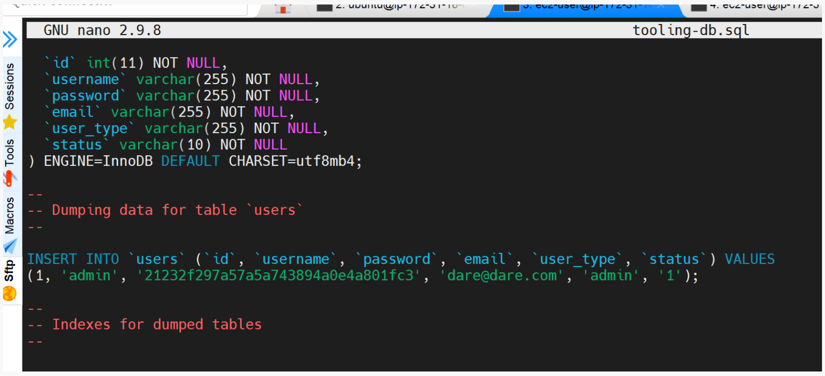
Not secure | 35.178.158.174/login.php
Login
Username
Password
Password Must Be 4 Characters
Login

**Deploy the tooling-db.sql into the database.**

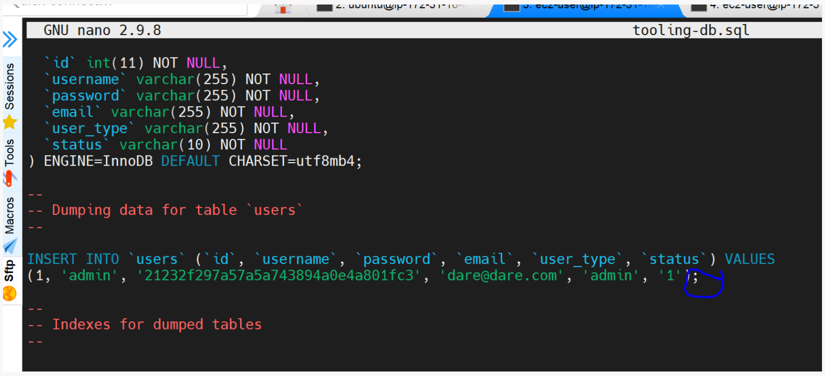
In order to apply the tooling-db.sql, run the below command:

```
cd tooling
sudo nano tooling-db.sql
```

Move into the file and add a column at the end of the ‘INSERT INTO’ line.



```
  GNU nano 2.9.8                                                     tooling-db.sql

  `id` int(11) NOT NULL,
  `username` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `user_type` varchar(255) NOT NULL,
  `status` varchar(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `users`
--

INSERT INTO `users` (`id`, `username`, `password`, `email`, `user_type`, `status`) VALUES
(1, 'admin', '21232f297a57a5a743894a0e4a801fc3', 'dare@dare.com', 'admin', '1');

--
-- Indexes for dumped tables
--
```



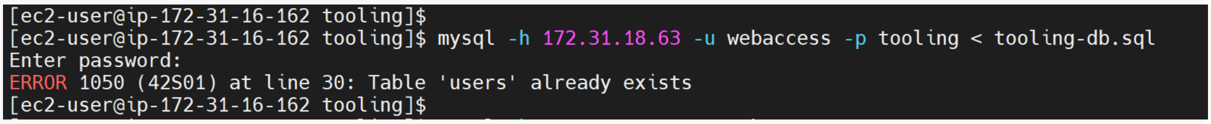
```
  GNU nano 2.9.8                                                     tooling-db.sql

  `id` int(11) NOT NULL,
  `username` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `user_type` varchar(255) NOT NULL,
  `status` varchar(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `users`
--

INSERT INTO `users` (`id`, `username`, `password`, `email`, `user_type`, `status`) VALUES
(1, 'admin', '21232f297a57a5a743894a0e4a801fc3', 'dare@dare.com', 'admin', '1');

--
-- Indexes for dumped tables
--
```

Then run below command so as to export tooling-db into our database:

```
$ mysql -h 172.31.18.63 -u webaccess -p tooling < tooling-db.sql
```

Input the tooling database password

Could not access as the user already exists.



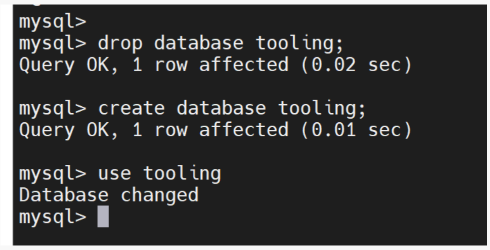
```
[ec2-user@ip-172-31-16-162 tooling]$
[ec2-user@ip-172-31-16-162 tooling]$ mysql -h 172.31.18.63 -u webaccess -p tooling < tooling-db.sql
Enter password:
ERROR 1050 (42S01) at line 30: Table 'users' already exists
[ec2-user@ip-172-31-16-162 tooling]$
```

Steps taken:

Logged in to MySQL of the database server

Dropped the existing database ‘tooling’ and then created a new one in order to remove the conflicts.

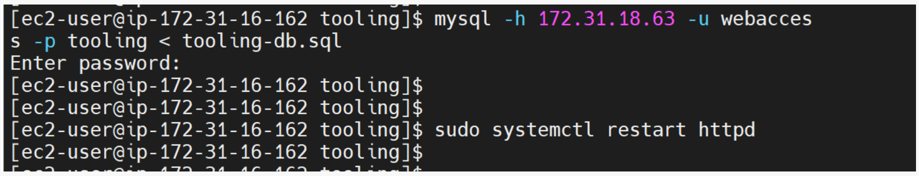
```
mysql> drop database tooling;
mysql> create database tooling;
mysql> use tooling;
```



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

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

mysql> use tooling
Database changed
mysql> █
```

On webserver, try to deploy the tooling-db.sql again

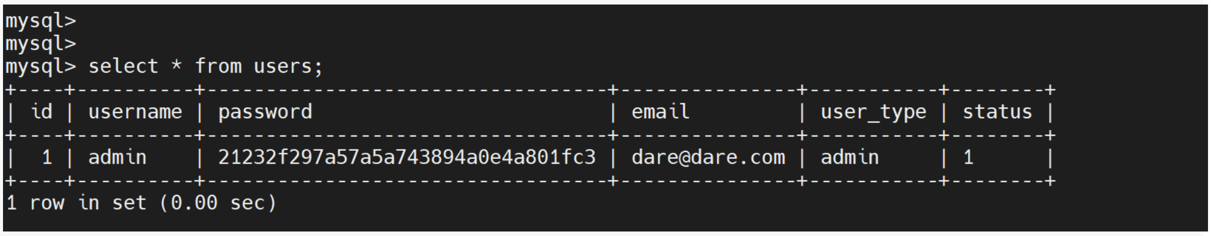
```
$ mysql -h 172.31.18.63 -u webaccess -p tooling < tooling-db.sql
```



```
[ec2-user@ip-172-31-16-162 tooling]$ mysql -h 172.31.18.63 -u webacces
s -p tooling < tooling-db.sql
Enter password:
[ec2-user@ip-172-31-16-162 tooling]$
[ec2-user@ip-172-31-16-162 tooling]$
[ec2-user@ip-172-31-16-162 tooling]$ sudo systemctl restart httpd
[ec2-user@ip-172-31-16-162 tooling]$
```

The above output confirms that the tooling-db.sql has been successfully deployed into the database.

```
mysql> select * from users;
```



```
mysql>
mysql>
mysql> select * from users;
+----+----------+----------------------------------+---------------+-----------+--------+
| id | username | password                         | email         | user_type | status |
+----+----------+----------------------------------+---------------+-----------+--------+
|  1 | admin    | 21232f297a57a5a743894a0e4a801fc3 | dare@dare.com | admin     | 1      |
+----+----------+----------------------------------+---------------+-----------+--------+
1 row in set (0.00 sec)
```

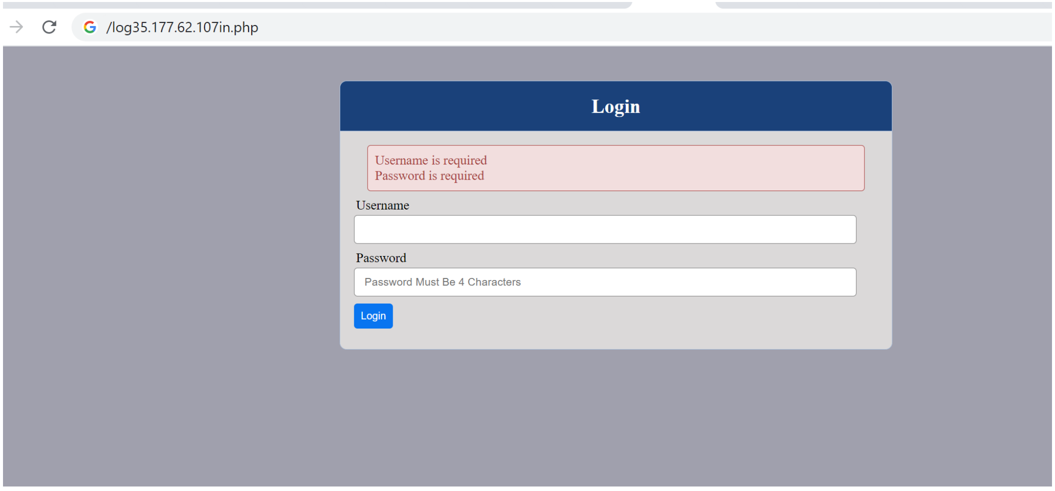
Finally, open the website in your browser

http://<Web-Server-Public-IP-Address-or-Public-DNS-Name>/index.php and make sure you can login into the website with admin user.

Web server 1http://35.177.62.107
Web server 2http://3.8.6.231/

Login with  username: admin
            Password: admin

Blocker:
Unable to log in with the username and password details. This was an issue with the PHP installation as an older version (PHP 7.2) was installed rather than a newer version- PHP 7.4.


/log35.177.62.107in.php

Login

Username is required
Password is required

Username

Password

Password Must Be 4 Characters

**Steps taken to troubleshoot:**

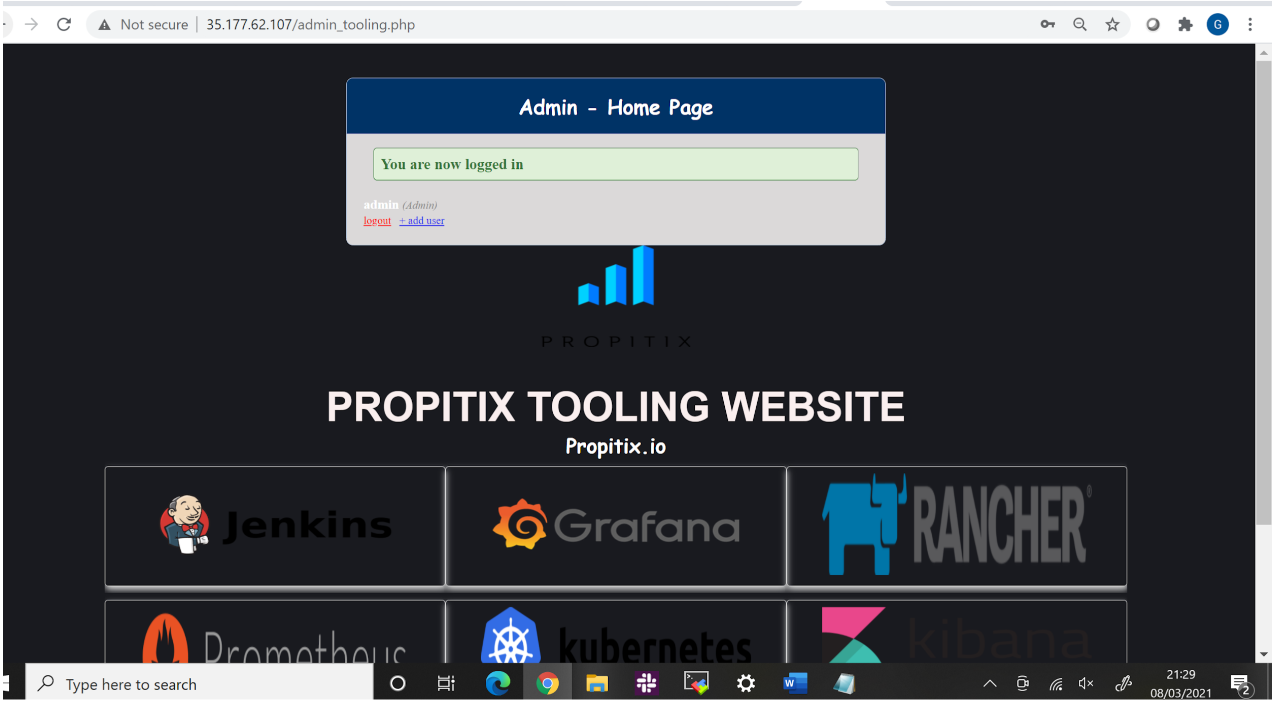
Used error log command to identify where issue was:

```
sudo tail -f /var/log/httpd/error_log
```

Uninstalled the older version by running the command: `yum -y remove php*`

Then installed the latest version PHP 7.4 as done under section PHP above.

```
sudo dnf module install -y php:7.4
sudo dnf install -y php-{mysqlnd,xml,xmlrpc,curl,gd,mbstring,opcache,soap,zip}
sudo systemctl restart httpd
```


Not secure | 35.177.62.107/admin_tooling.php

Admin - Home Page

You are now logged in

admin (Admin)

logout + add user

PROPITIX TOOLING WEBSITE

Propitix.io

Credits:

Darey.io

Network-attached Storage (NAS): https://en.wikipedia.org/wiki/Network-attached_storage
Storage Area Network (SAN): https://en.wikipedia.org/wiki/Storage_area_network
Block Level Storage: https://en.wikipedia.org/wiki/Block-level_storage
Object Storage: https://en.wikipedia.org/wiki/Object_storage
AWS Storage Services: https://dzone.com/articles/confused-by-aws-storage-options-s3-ebs-amp-efs-explained
PHP 7.4: https://www.tecmint.com/install-lamp-on-centos-8
NFS: https://computingforgeeks.com/install-and-configure-nfs-server-on-centos-rhel/
Subnet/CIDR:https://www.digitalocean.com/community/tutorials/understanding-ip-addresses-subnets-and-cidr-notation-for-networking
Firewall Man Page: https://firewalld.org/documentation/man-pages/firewall-cmd.html
NFS Server and Client: https://www.linuxtechi.com/setup-nfs-server-on-centos-8-rhel-8/
Subnet Calculator: https://www.adminsub.net/ipv4-subnet-calculator/10.25.0.0/25
Subnet Sheet: https://www.aelius.com/njh/subnet_sheet.html
NFS Mount: https://www.poftut.com/how-to-mount-nfs-share-in-linux-and-windows/

https://www.codecademy.com/courses/deploy-a-website/lessons/github-pages/exercises/git-init