WEB SOLUTION WITH WORDPRESS

The 3-tier setup requirement.

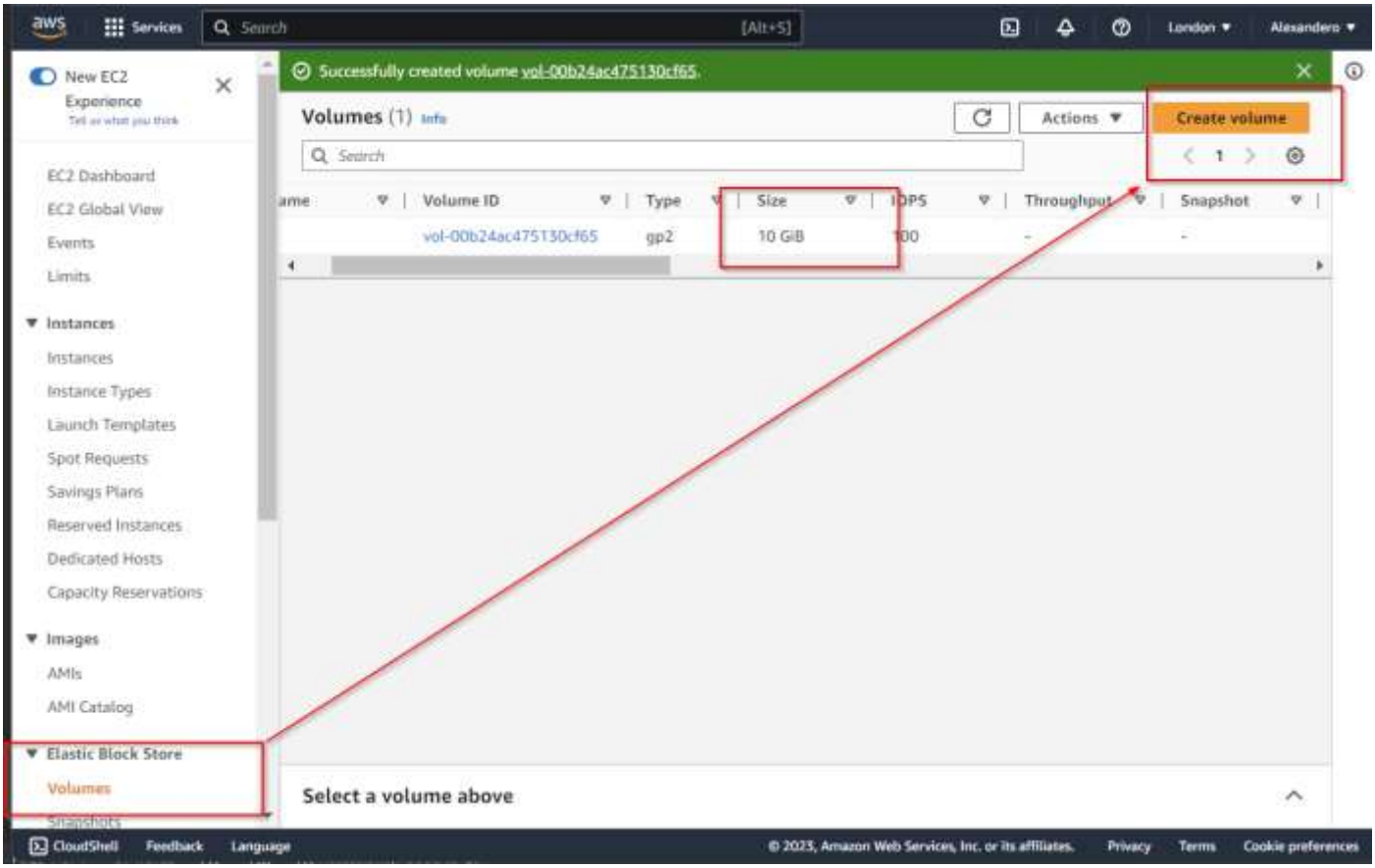
1. A computer to serve as a client.
2. An AWS EC2 Linux server as a web server (on this, we will install the WordPress).
3. An AWS EC2 Linux server as a database (DB) server

We will be making use of the RedHat (Red Hat Enterprise Linux, RHEL) operating system for this project.

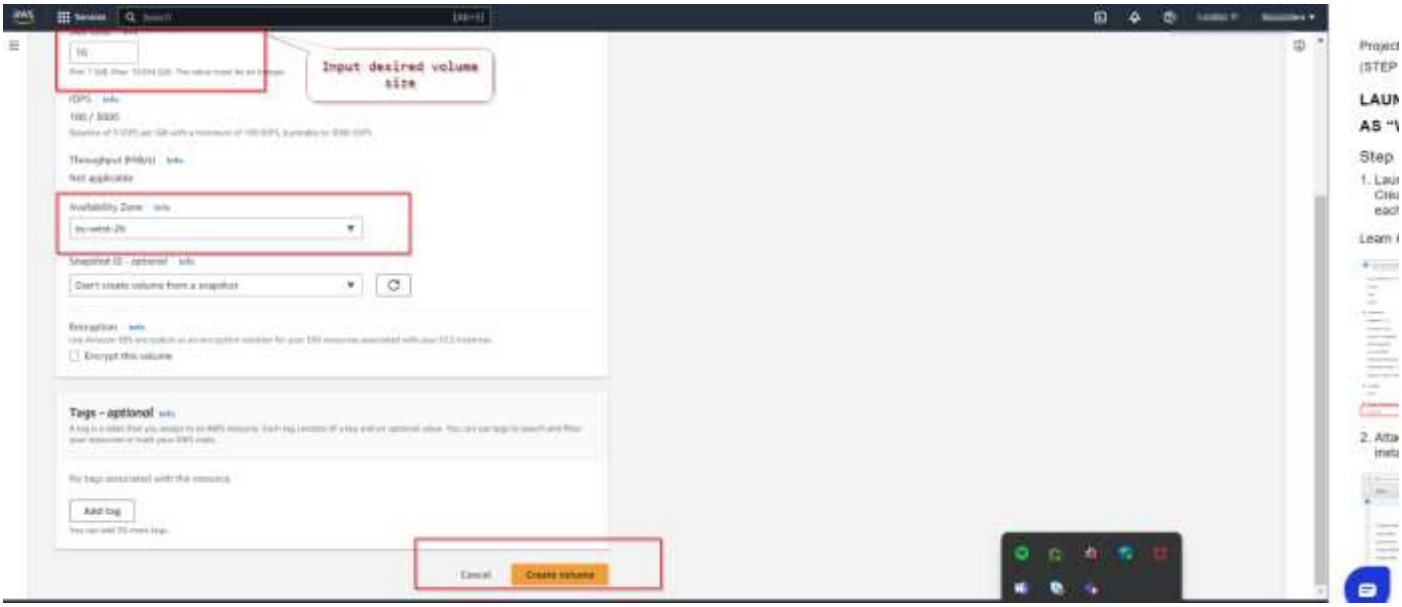
PREPARE THE WEBSERVER

Step 1.

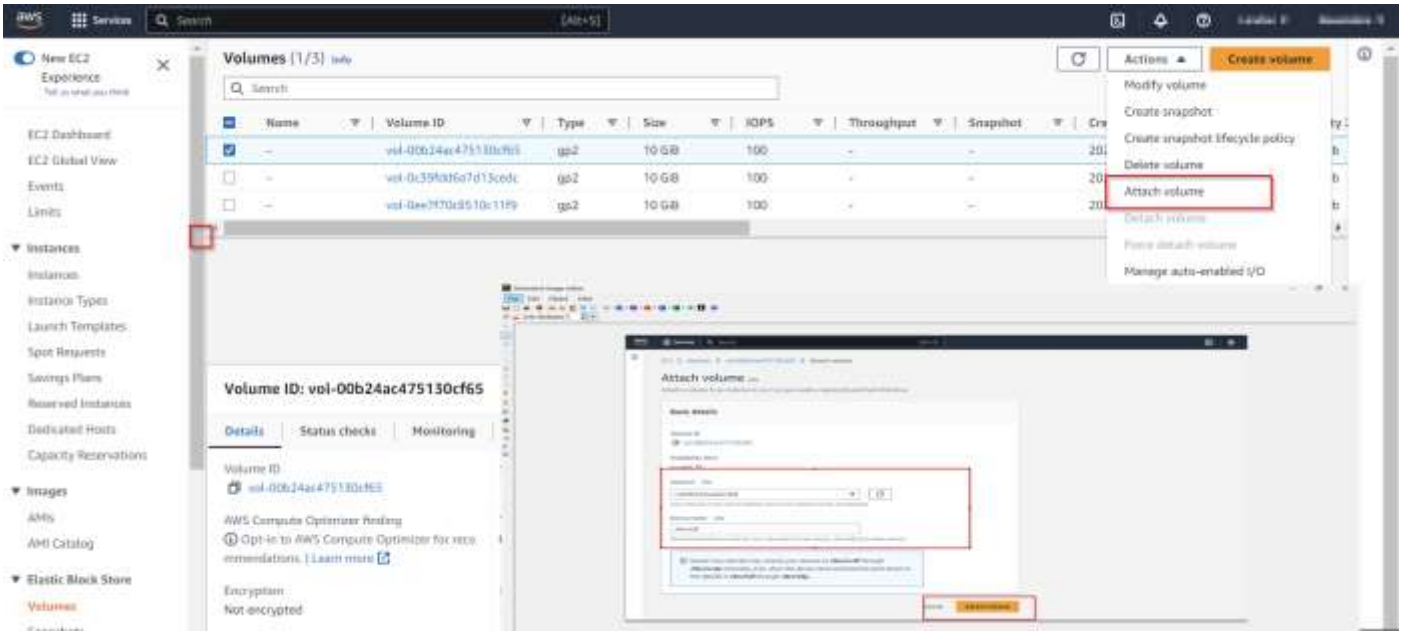
Create 3 volumes of 10GB size on AWS elastic block store (EBS), Lunch an EC2 instance on AWS that will serve as webserver and attach the 3 volumes to the instance.



Ensure the volumes are created on the same availability zone as the server instance.



Attach the volumes.



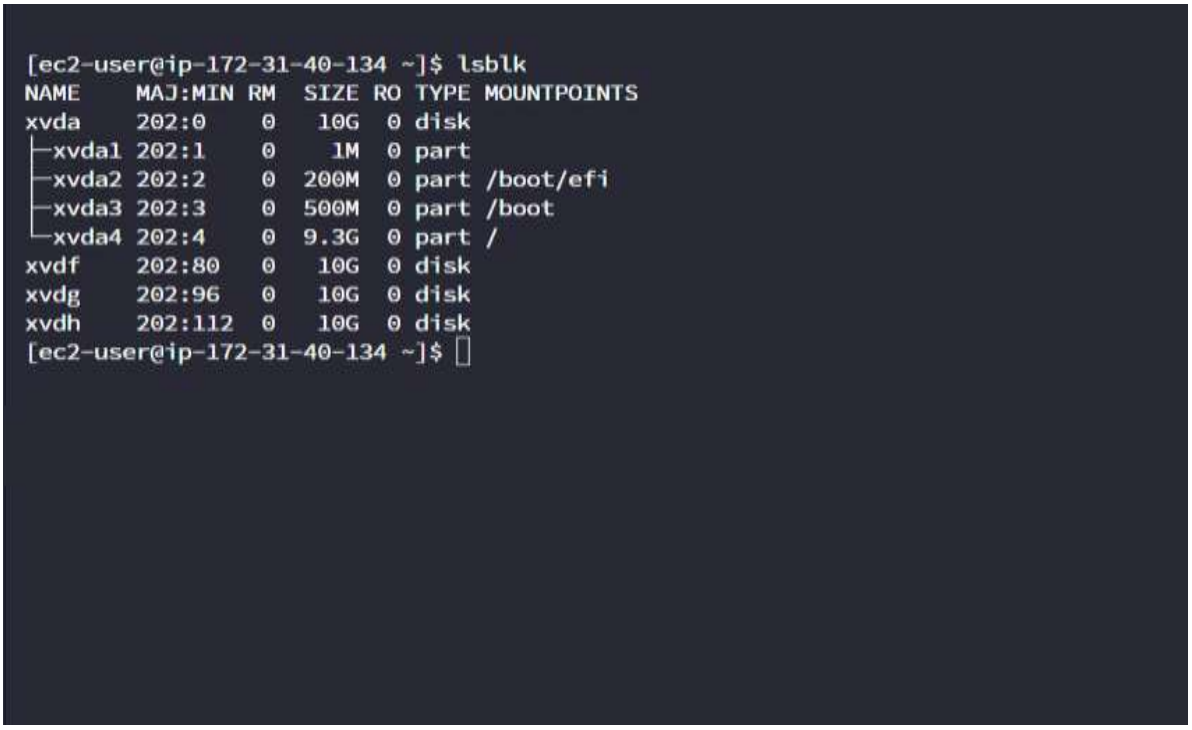
Step 2.

Connect the webserver instance to the terminal and begin configuration.

Firstly, check to confirm the names of the volume blocks that you have just attached to the webserver.

Use code.

$ lsblk



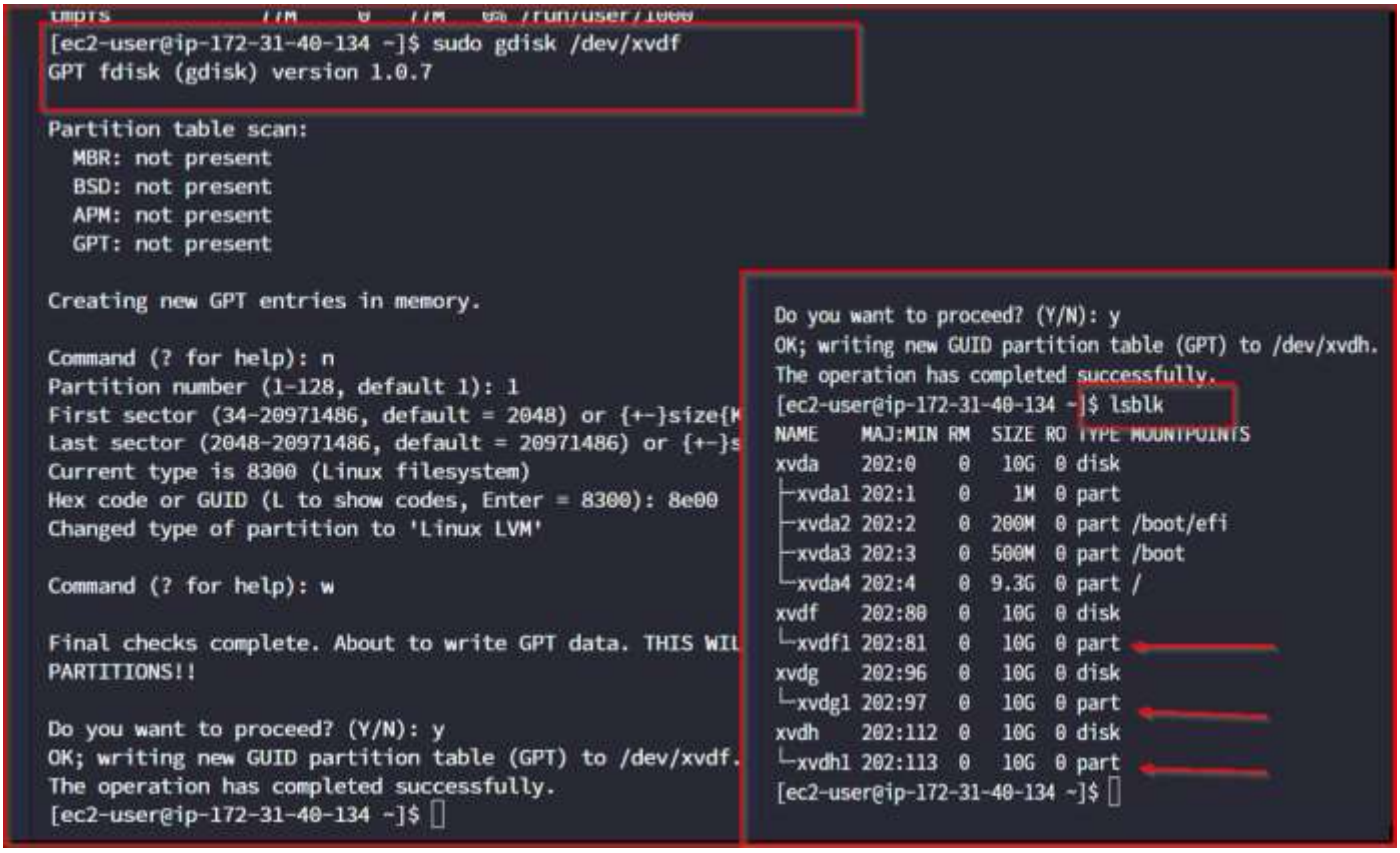
You would see names as xvdf, xvdg. xvdh. Note that these volume blocks (devices) reside in the /dev/ directory.

Step 3.

Create a single partition on each of the **3 disks** using the gdisk utility. Here you will require superuser privileges to use this utility.

$ sudo gdisk /dev/xvdf

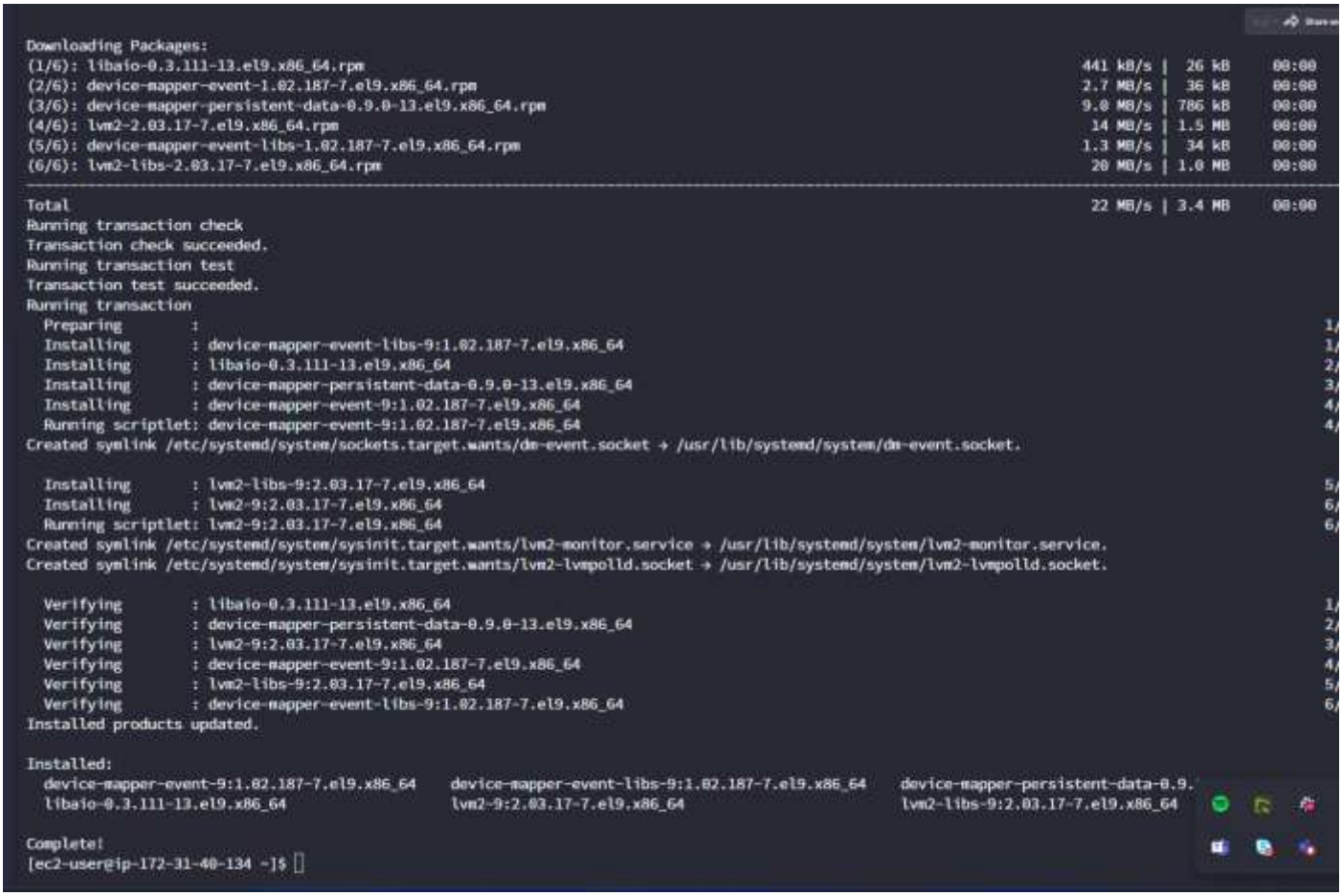
And use the **lsblk** command to view the newly configured partition on the 3 disks.



Step 4.

Install logical volume manager 2 (LVM2). This is a logical volume management system for Linux.

$ sudo yum install lvm2 -y



Mark each of the 3 disks as physical volumes (PVs) to be used by the logical volume manager LVM. Use the pvcreate utility.

$sudo pvcreate /dev/xvdf1 /dev/xvdg1 /dev/xvdh1

Verify the physical volumes (PVs) created by using the **pvs** code.

```
[ec2-user@ip-172-31-40-134 ~]$ sudo pvs
  PV         VG Fmt  Attr PSize   PFree
  /dev/xvdf1    lvm2 ---  <10.00g <10.00g
  /dev/xvdg1    lvm2 ---  <10.00g <10.00g
  /dev/xvdh1    lvm2 ---  <10.00g <10.00g
[ec2-user@ip-172-31-40-134 ~]$
```

Create a volume group (VG) with vgcreate utility and name the volume group webdata-vg. This is needed in order to add the 3 physical volumes (PV) to the VG.

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

Verify the volumes group (VG) created by running the code **sudo vgs.**

```
[ec2-user@ip-172-31-40-134 ~]$ sudo vgcreate webdata-vg /dev/xvdf1 /dev/xvdg1 /dev/xvdh1
  Volume group "webdata-vg" successfully created
[ec2-user@ip-172-31-40-134 ~]$ sudo vgs
  VG         #PV #LV #SN Attr   VSize   VFree
  webdata-vg   3   0   0 wz--n- <29.99g <29.99g
[ec2-user@ip-172-31-40-134 ~]$
```

Step 5.

Now we will go a step further to create 2 logical volumes named **apps-lv** (**app-lv** will be used to store data for the website. In this case, would give it half of the VG size) and **logs-lv** (**logs-lv** will be used to store data for logs. In this case we would allocate the other half of the VG size). Use **lvcreate** utility.

$ sudo lvcreate -n apps-lv -L 14G webdata-vg

$ sudo lvcreate -n logs-lv -L 14G webdata-vg

Verify the logical volume groups (LV) created by running the code **sudo lvs**

```
[ec2-user@ip-172-31-40-134 ~]$ sudo lvcreate -n apps-lv -L 14G webdata-vg && sudo lvcreate -n logs-lv -L 14G webdata-vg
  Logical volume "apps-lv" created.
  Logical volume "logs-lv" created.
[ec2-user@ip-172-31-40-134 ~]$ sudo lvs
  LV      VG         Attr       LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  apps-lv webdata-vg -wi-a----- 14.00g
  logs-lv webdata-vg -wi-a----- 14.00g
[ec2-user@ip-172-31-40-134 ~]$
```

View the set up with the lsblk command.

$sudo lsblk

```
[ec2-user@ip-172-31-40-134 ~]$ lsblk
NAME                      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
xvda                      202:0    0   10G  0 disk
├─xvda1                   202:1    0    1M  0 part
├─xvda2                   202:2    0  200M  0 part /boot/efi
├─xvda3                   202:3    0  500M  0 part /boot
└─xvda4                   202:4    0  9.3G  0 part /
xvdf                      202:80   0   10G  0 disk
└─xvdf1                   202:81   0   10G  0 part
  └─webdata--vg-apps--lv 253:0    0   14G  0 lvm
xvdg                      202:96   0   10G  0 disk
└─xvdg1                   202:97   0   10G  0 part
  ├─webdata--vg-apps--lv 253:0    0   14G  0 lvm
  └─webdata--vg-logs--lv 253:1    0   14G  0 lvm
xvdh                      202:112  0   10G  0 disk
└─xvdh1                   202:113  0   10G  0 part
  └─webdata--vg-logs--lv 253:1    0   14G  0 lvm
[ec2-user@ip-172-31-40-134 ~]$
```

Step 6.

Now let's format the 2 logical volumes (**apps-lv and logs-lv**) with ext4 filesystem. Use mkfs.ext4

$ sudo mkfs -t ext4 /dev/webdata-vg/apps-lv

$ sudo mkfs -t ext4 /dev/webdata-vg/logs-lv

```
[ec2-user@ip-172-31-40-134 ~]$ sudo mkfs -t ext4 /dev/webdata-vg/apps-lv
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 3670016 4k blocks and 917504 inodes
Filesystem UUID: 5a3ca932-8b76-481b-8097-9e564788b719
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

[ec2-user@ip-172-31-40-134 ~]$ sudo mkfs -t ext4 /dev/webdata-vg/logs-lv
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 3670016 4k blocks and 917504 inodes
Filesystem UUID: 4a80c1ad-5c3a-4a22-9cfa-21513c6e01ae
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

Step 7.

Create /var/www/html directory to store website files.

$ sudo mkdir -p /var/www/html

Create /home/recovery/logs directory to store backup of log data.

$ sudo mkdir -p /home/recovery/logs

Mount /var/www/html on app-lv logical volume

$ sudo mount /dev/webdata-vg/apps-lv /var/www/html/

```
[ec2-user@ip-172-31-40-134 ~]$ ^C
[ec2-user@ip-172-31-40-134 ~]$ sudo mkdir -p /var/www/html
[ec2-user@ip-172-31-40-134 ~]$ ls
[ec2-user@ip-172-31-40-134 ~]$ sudo mkdir -p /home/recovery/logs
[ec2-user@ip-172-31-40-134 ~]$ ls
[ec2-user@ip-172-31-40-134 ~]$ sudo mount /dev/webdata-vg/apps-lv /var/www/html/
[ec2-user@ip-172-31-40-134 ~]$
```

Use rsync utility to backup all the files in the log directory /var/log into /home/recovery/logs (This is required before mounting the file system).

$ sudo rsync -av /var/log/. /home/recovery/logs/

Step 8.

mount **/var/log** on **logs-lv** logical volume. (Note that all the existing data on /var/log will be deleted)

$ sudo mount /dev/webdata-vg/logs-lv /var/log

Restore all the log files back into the /var/log directory.

$ sudo rsync -av /home/recovery/logs/. /var/log

Step 9.

Update the /etc/fstab file using the device UUID **so that the mount configuration will persist after restart of the server.**

$ sudo blkid

```
[ec2-user@ip-172-31-40-134 ~]$ sudo blkid
/dev/xvda4: LABEL="root" UUID="287d9c0b-0e0f-4e92-8534-45733aa3dc60" TYPE="xfs" PARTUUID="6264d520-3fb9-423f-8ab8-7a8a8e3d3562"
/dev/mapper/webdata--vg-logs--lv: UUID="4a88c1ad-5c3a-4a22-9cfa-21513c6e01ae" TYPE="ext4"
/dev/xvda2: SEC_TYPE="msdos" UUID="7B77-95E7" TYPE="vfat" PARTUUID="68b2985b-df3e-4fb3-80fa-49d1e773aa33"
/dev/xvda3: LABEL="boot" UUID="7bc24af7-289d-4bce-b17e-308c3aafe968" TYPE="xfs" PARTUUID="cb87c243-bc44-4717-853e-28852021225b"
/dev/xvda1: PARTUUID="fac7f1fb-3e8d-4137-a512-961de09a5549"
/dev/xvdh1: UUID="vAmDfr-0Iym-k7XE-Jel4-67pH-3OB2-E9t71V" TYPE="LVM2_member" PARTLABEL="Linux LVM" PARTUUID="e3a33947-8ba3-4e3a-93d5-96cd4869d462"
/dev/xvdf1: UUID="17vQtP-miAz-IUIr-5tom-svfw-vrhk-N95svt" TYPE="LVM2_member" PARTLABEL="Linux LVM" PARTUUID="66d71501-f9f8-4bd5-a728-6b
/dev/mapper/webdata--vg-apps--lv: UUID="5a3ca932-8b76-481b-8897-9e564788b719" TYPE="ext4"
/dev/xvdg1: UUID="4HwCfh-VmIu-yY8F-B2nP-rMA2-uIxv-tvOA4M" TYPE="LVM2_member" PARTLABEL="Linux LVM" PARTUUID="85eac6ff-b63a-44da-ade4-85
[ec2-user@ip-172-31-40-134 ~]$ ^C
[ec2-user@ip-172-31-40-134 ~]$
```

Sudo vi /etc/fstab

```
UUID=287d9c0b-0e0f-4e92-8534-45733aa3dc68       /       xfs     defaults        0       0
UUID=7bc24af7-289d-4bce-b17e-300c3aafe968       /boot   xfs     defaults        0       0

# mounts for wordpress webserver
UUID=4a88c1ad-5c3a-4a22-9cfa-21513c6e01ae /var/log ext4 defaults 0 0
UUID=5a3ca932-8b76-481b-8097-9e564788b719 /var/www/html ext4 defaults 0 0
```

Test the configuration and reload the daemon.

$ sudo mount -a

$ sudo systemctl daemon-reload

And verify that your setup is running and properly mounted by running the command:

$ df -h

PREPARE THE DATABASE SERVER

Step 10.

We will have to repeat the entire process (**Step 1 to Step 9)** above for a second server, the database server but instead of **apps-lv** create **database-lv** and mount it to **/database** directory instead of **/var/www/html/.**

```
database-vg/ disk/       dm-0        dma_heap/    dri/
[ec2-user@ip-172-31-44-127 ~]$ sudo mkfs -t ext4 /dev/database-vg/database-lv
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 1310720 4k blocks and 327680 inodes
Filesystem UUID: 133a51d4-5014-4f0e-9970-70f427807729
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

[ec2-user@ip-172-31-44-127 ~]$ ls -l /database
total 0
[ec2-user@ip-172-31-44-127 ~]$ sudo mount /dev/database-vg/database-lv /database/
[ec2-user@ip-172-31-44-127 ~]$ df -h
Filesystem                          Size  Used Avail Use% Mounted on
devtmpfs                            4.0M     0  4.0M   0% /dev
tmpfs                               385M     0  385M   0% /dev/shm
tmpfs                               154M  5.8M  149M   4% /run
/dev/xvda4                          9.4G  1.3G  8.1G  14% /
/dev/xvda3                          495M  153M  343M  31% /boot
/dev/xvda2                          200M  8.0K  200M   1% /boot/efi
tmpfs                                77M     0   77M   0% /run/user/1000
/dev/mapper/database--vg-database--lv  4.9G   24K  4.6G   1% /database
[ec2-user@ip-172-31-44-127 ~]$ sudo blkid
/dev/xvda4: LABEL="root" UUID="287d9c0b-0e0f-4e92-8534-45733aa3dc68" TYPE="xfs" PARTUUID="6264d520-3fb9-423f-8ab8-7a0a8e3d3562"
/dev/xvda2: SEC_TYPE="msdos" UUID="7B77-95E7" TYPE="vfat" PARTUUID="68b2905b-df3e-4fb3-80fa-49d1e773aa33"
/dev/xvda3: LABEL="boot" UUID="7bc24af7-289d-4bce-b17e-300c3aafe968" TYPE="xfs" PARTUUID="cb07c243-bc44-4717-853e-28852021225b"
/dev/xvda1: PARTUUID="fac7f1fb-3e8d-4137-a512-961de09a5549"
/dev/xvdh1: UUID="wQKk3g-EIzz-WDpc-0QJE-8In8-0UxM-aWzCfn" TYPE="LVM2_member" PARTLABEL="Linux LVM" PARTUUID="9c7ab100-2d46-483c-9d00-53a58a32a022"
/dev/xvdf1: UUID="kRhzoy-LlGl-7ECk-Ue27-x5fn-mggl-l0Jojs" TYPE="LVM2_member" PARTLABEL="Linux LVM" PARTUUID="b0abd6f0-b42e-45be-a244-10c1ecee06b8"
/dev/mapper/database--vg-database--lv: UUID="133a51d4-5014-4f0e-9970-70f427807729" TYPE="ext4"
/dev/xvdg1: UUID="RR9PHH-y3pX-NLqz-BeiM-Aw06-Nygx-d7oNuw" TYPE="LVM2_member" PARTLABEL="Linux LVM" PARTUUID="1aefcb0b-3385-464d-9297-28a6a2ec468f"
[ec2-user@ip-172-31-44-127 ~]$ ^C
[ec2-user@ip-172-31-44-127 ~]$ sudo vi /etc/fstab
[ec2-user@ip-172-31-44-127 ~]$ sudo mount -a
[ec2-user@ip-172-31-44-127 ~]$ sudo systemctl daemon-reload
[ec2-user@ip-172-31-44-127 ~]$ df -h
Filesystem                          Size  Used Avail Use% Mounted on
devtmpfs                            4.0M     0  4.0M   0% /dev
tmpfs                               385M     0  385M   0% /dev/shm
tmpfs                               154M  5.8M  149M   4% /run
/dev/xvda4                          9.4G  1.3G  8.1G  14% /
/dev/xvda3                          495M  153M  343M  31% /boot
/dev/xvda2                          200M  8.0K  200M   1% /boot/efi
tmpfs                                77M     0   77M   0% /run/user/1000
/dev/mapper/database--vg-database--lv  4.9G   24K  4.6G   1% /database
```

Step 11.

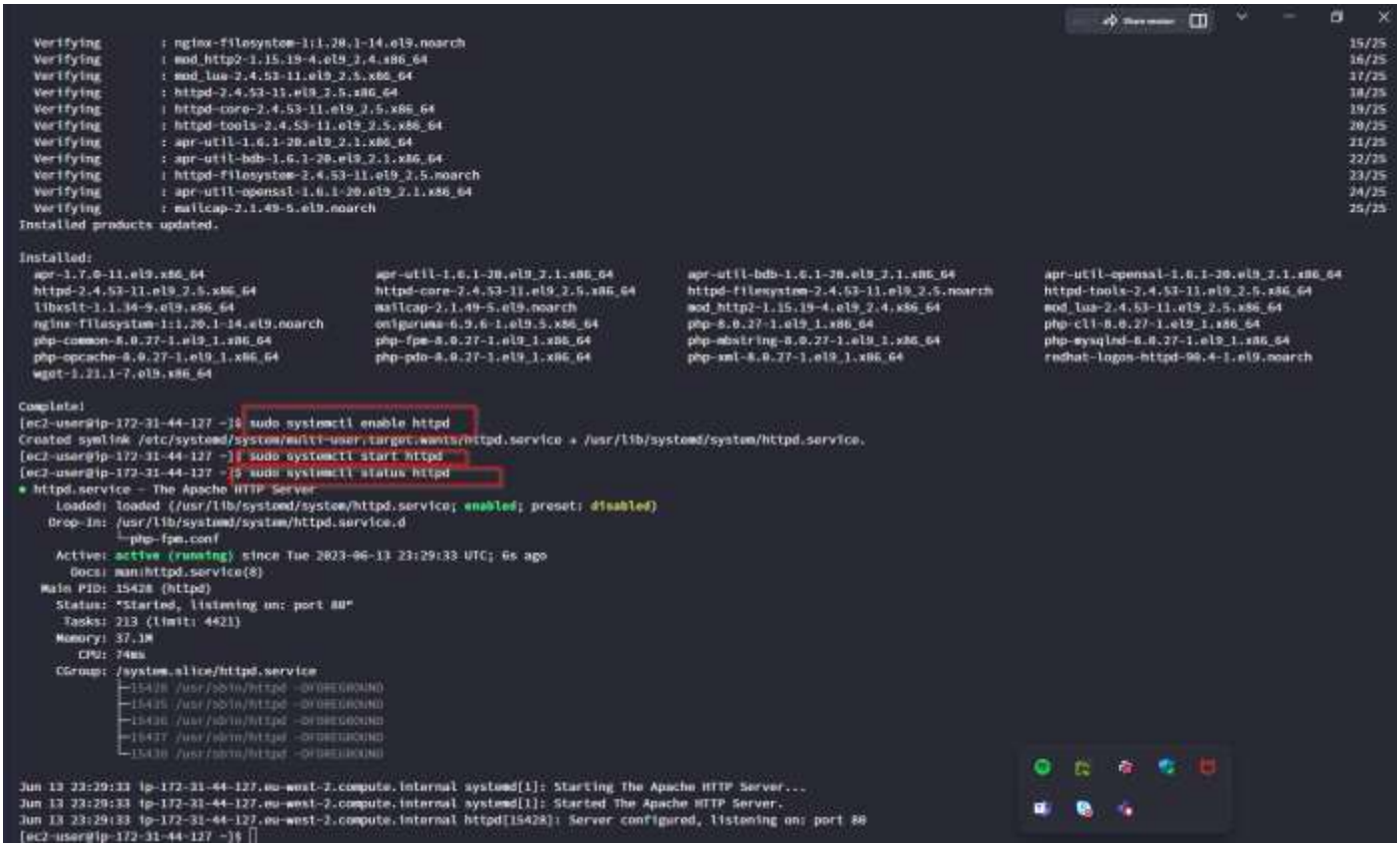
Install wget, Apache and its dependencies.

$ sudo yum -y install wget httpd php php-mysqlnd php-fpm php-json

Start Apache

$ sudo systemctl enable httpd

$ sudo systemctl start httpd



Step 12.

Install PHP and its dependencies.

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

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

sudo yum module list php

sudo yum module reset php

sudo yum module enable php:remi-7.4

sudo yum 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

Step 13.

DOWNLOAD WORDPRESS AND COPY WORDPRESS TO /var/www/html

$ sudo mkdir wordpress

$ cd wordpress

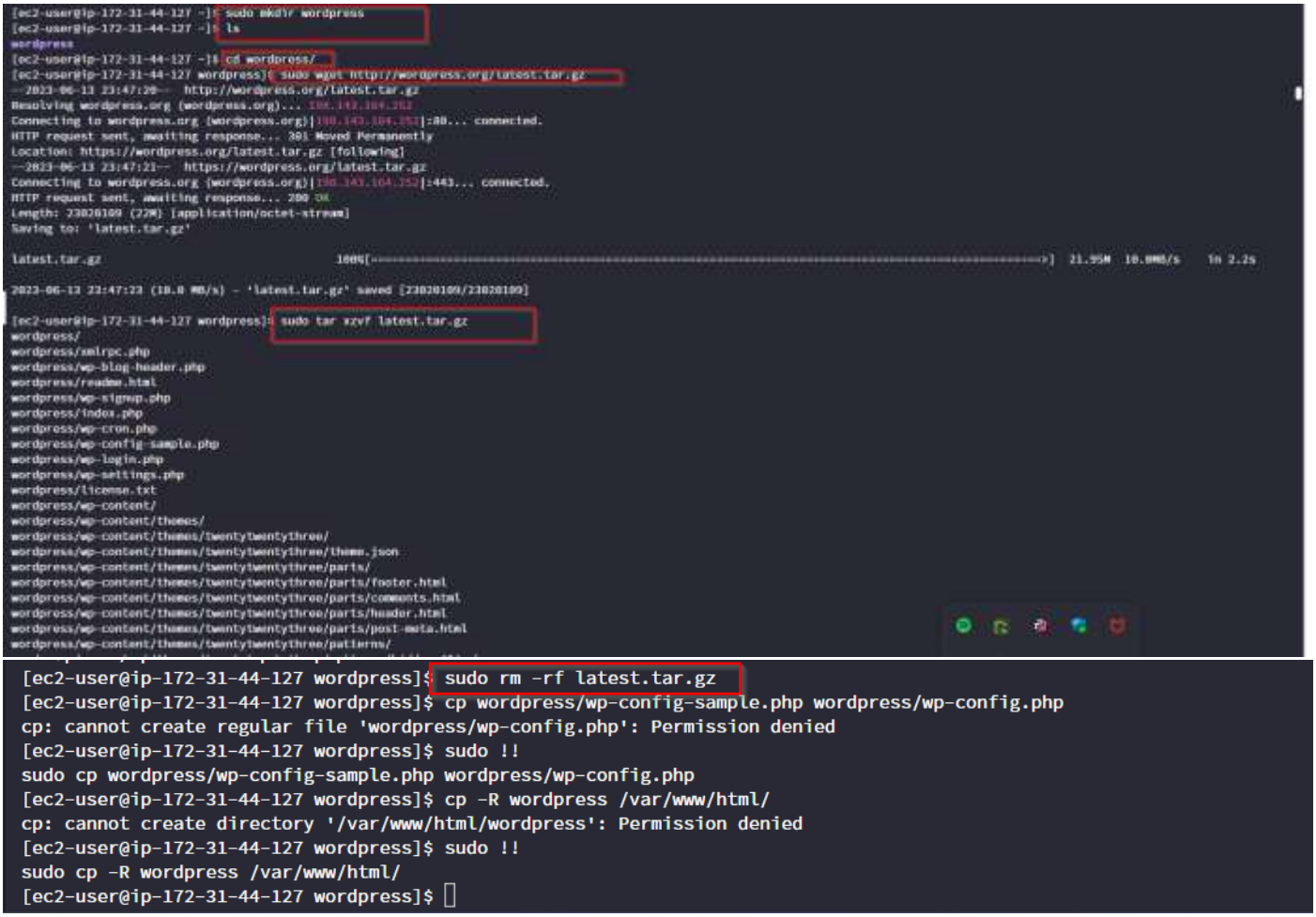
$ sudo wget http://wordpress.org/latest.tar.gz

$ sudo tar xzvf latest.tar.gz

$ sudo rm -rf latest.tar.gz

$ sudo cp wordpress/wp-config-sample.php wordpress/wp-config.php

$ cp -R wordpress /var/www/html/




```
[ec2-user@ip-172-31-44-127 wordpress]$ sudo rm -rf latest.tar.gz
[ec2-user@ip-172-31-44-127 wordpress]$ cp wordpress/wp-config-sample.php wordpress/wp-config.php
cp: cannot create regular file 'wordpress/wp-config.php': Permission denied
[ec2-user@ip-172-31-44-127 wordpress]$ sudo !!
sudo cp wordpress/wp-config-sample.php wordpress/wp-config.php
[ec2-user@ip-172-31-44-127 wordpress]$ cp -R wordpress /var/www/html/
cp: cannot create directory '/var/www/html/wordpress': Permission denied
[ec2-user@ip-172-31-44-127 wordpress]$ sudo !!
sudo cp -R wordpress /var/www/html/
[ec2-user@ip-172-31-44-127 wordpress]$
```

Step 14.

Configure SELinux Policies

$ sudo chown -R apache:apache /var/www/html/wordpress

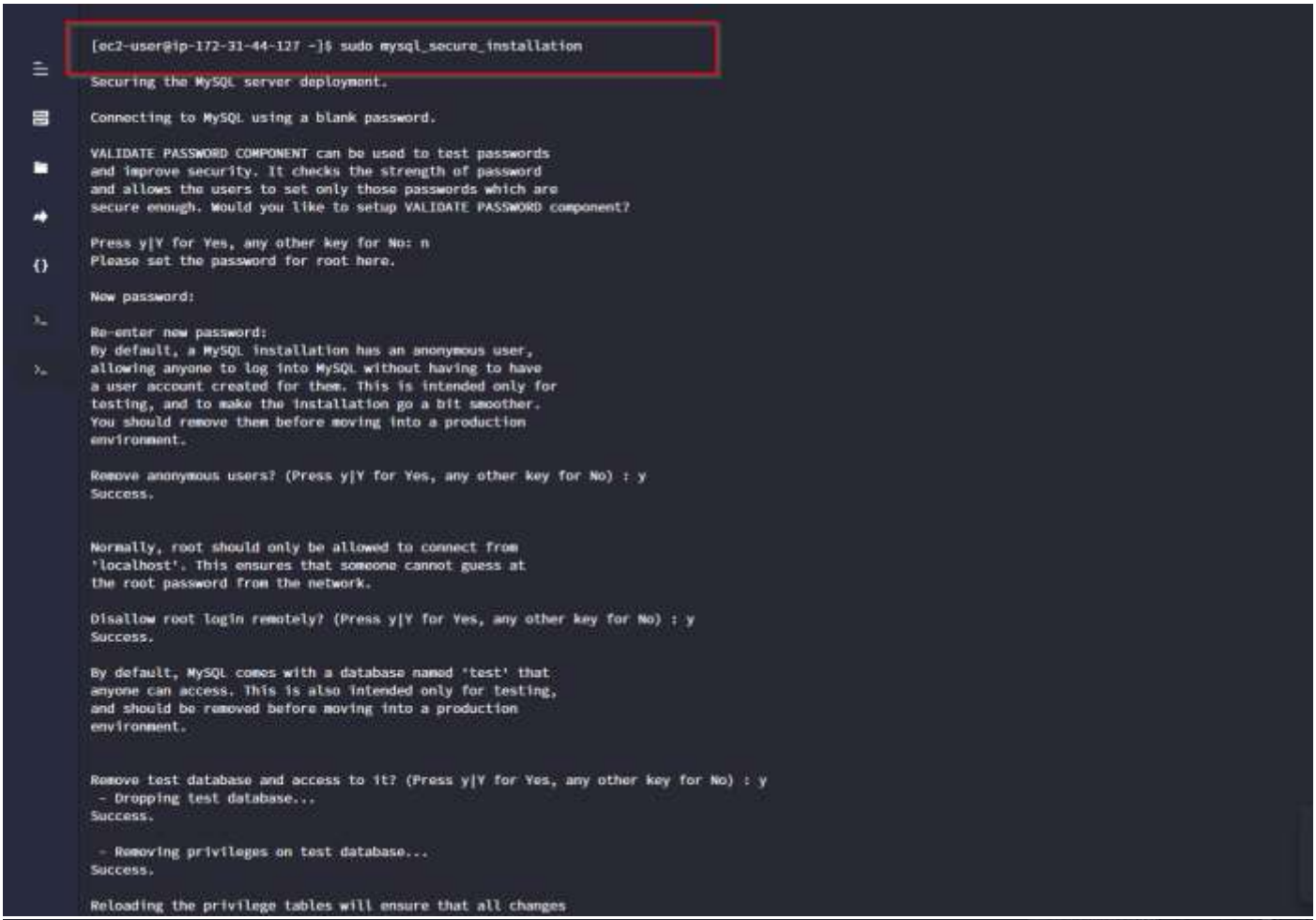
$ sudo chcon -t httpd_sys_rw_content_t /var/www/html/wordpress -R

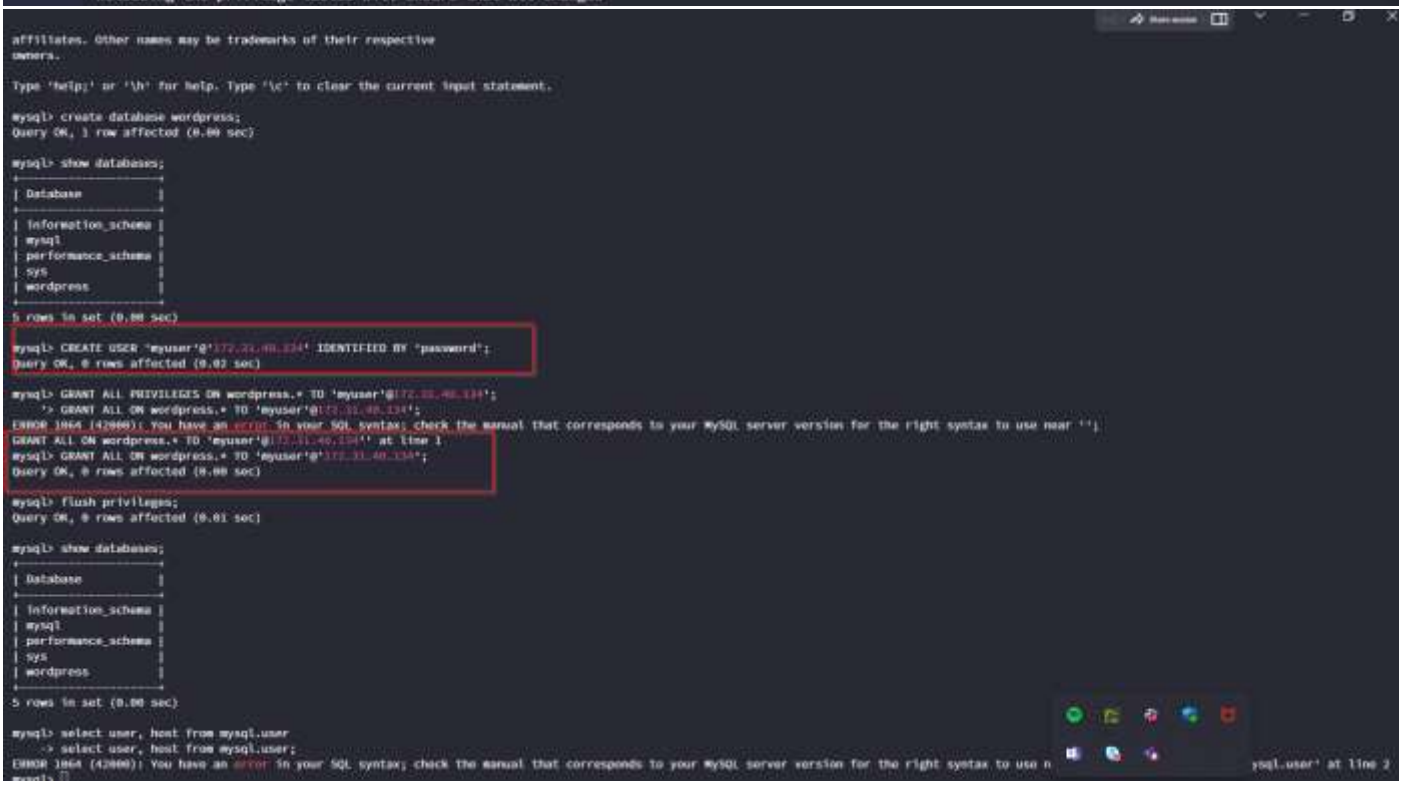
$ sudo setsebool -P httpd_can_network_connect=1

Configure mysql on your Database Server

$ sudo yum update

$ sudo yum install mysql-server





Step 15.

Set mysqld bind address to 0.0.0.0 to allow DB server to receive request from any IP address.

$ sudo vi /etc/my.cnf.d

```
#
# This group is read both both by the client and the server
# use it for options that affect everything
#
[client-server]

#
# include all files from the config directory
#
!includedir /etc/my.cnf.d
[mysqld]
bind-address=0.0.0.0
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

Step 16.

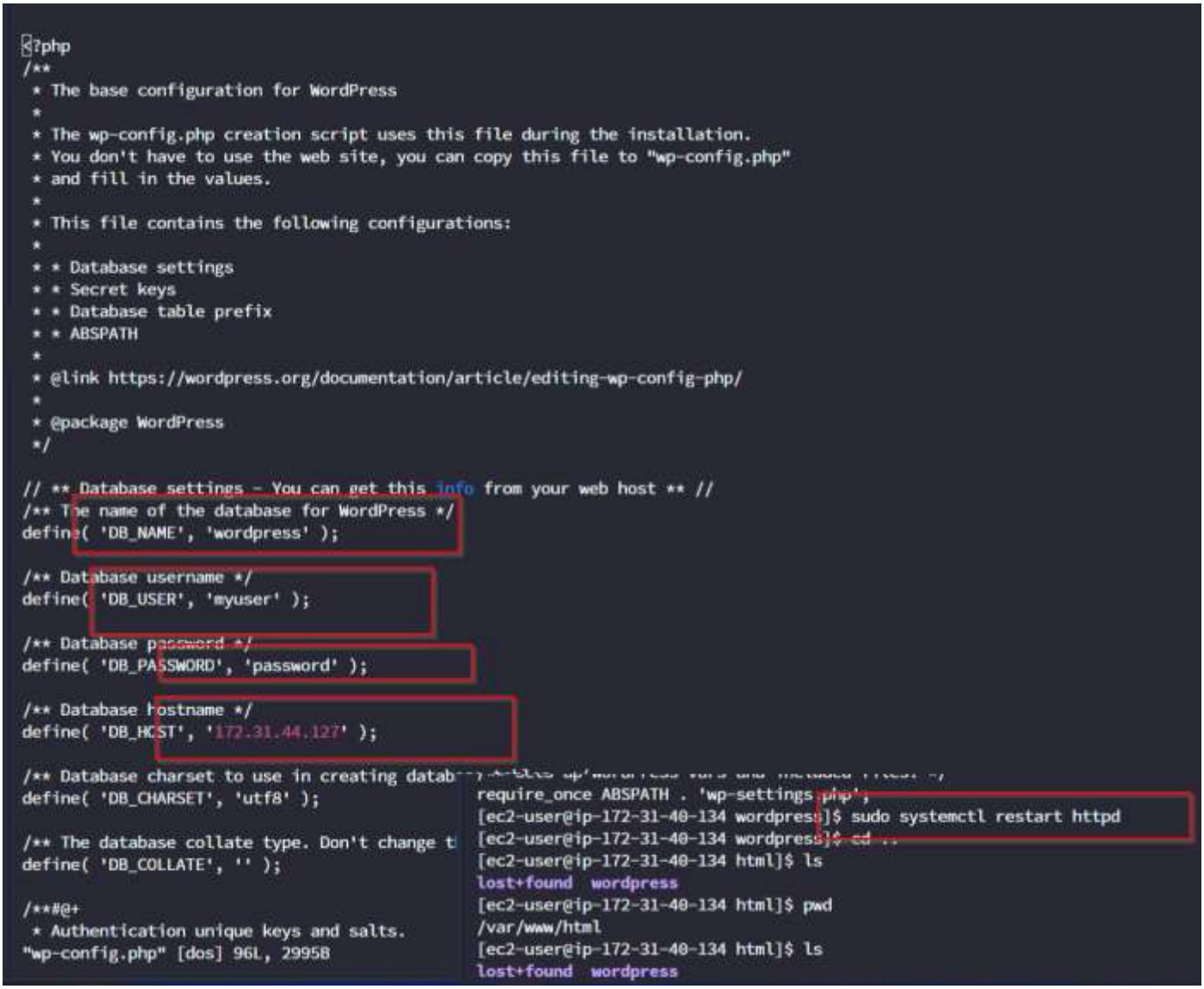
cd into /var/www/html/wordpress and Edit the wp-config.php file with the database credentials and restart the webserver

Database name = wordpress

Username = myuser

Password = password

Database public IP = 172.31.44.127



Step 17.

Configure WordPress to connect to remote database.

Open MySQL port 3306 on DB Server EC2 and allow access to the DB server **ONLY** from the Web Server's private IP address.

Step 18.

Install MySQL client and test that you can connect from your Web Server to your DB server by using mysql-client.

$ sudo yum install mysql

$ sudo mysql -u admin -p -h <DB-Server-Private-IP-address>

verify if you can successfully execute `SHOW DATABASES;` command and see a list of existing databases.

Step 19.

Change permissions and configuration so Apache can use WordPress.

$ sudo chown -R apache:apache /var/www.html

$ sudo chcon -t httpd_sys_rw_content_t /var/www/html/ -R

$ sudo setsebool -P httpd_can_network_connect=1

$ sudo setsebool -P httpd_can_network_connect_db 1

and restart the server.

$ sudo systemctl restart httpd

Enable TCP port 80 in Inbound Rules configuration for your Web Server EC2 (enable from everywhere 0.0.0.0/0 or from your workstation's IP)

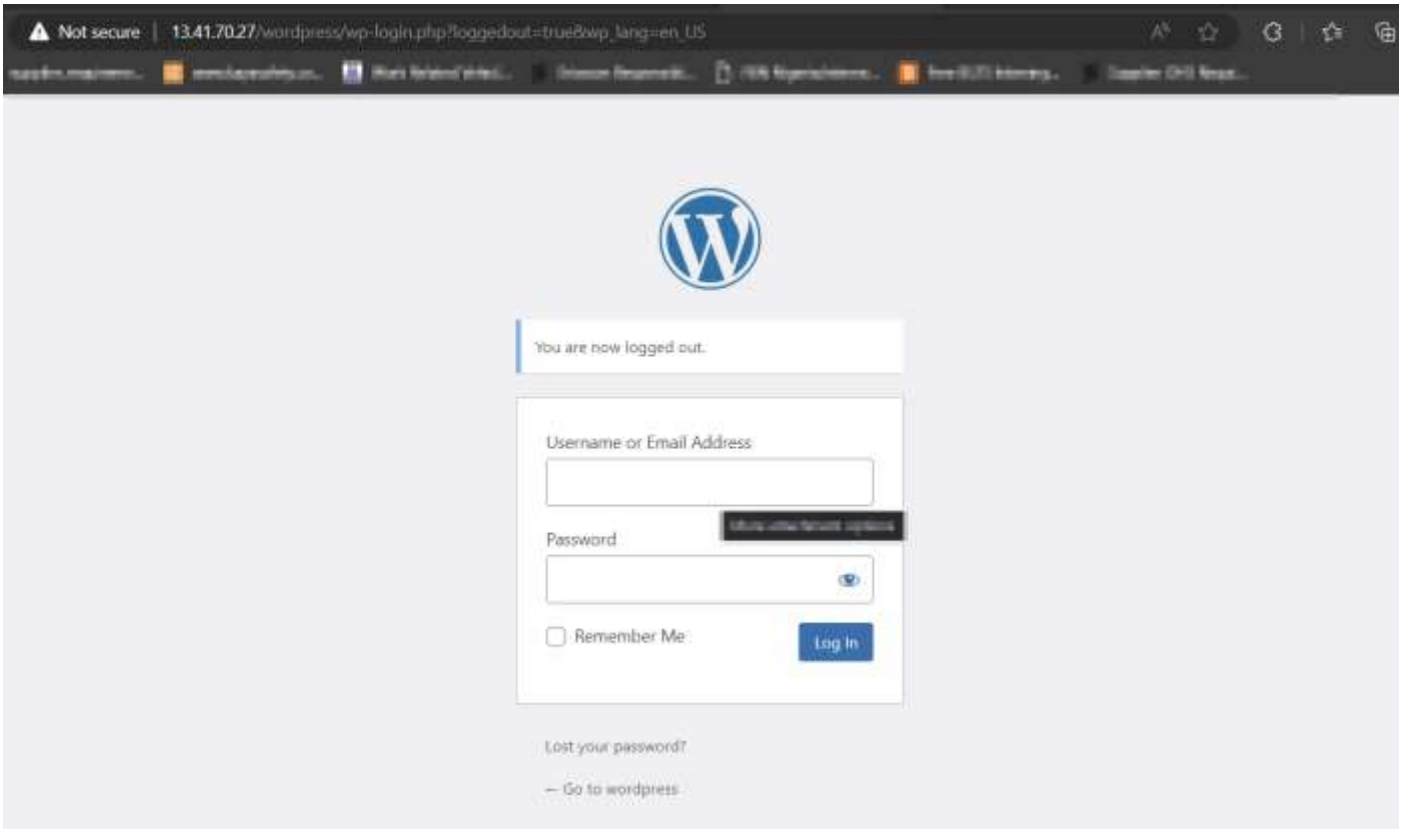
Step 20.

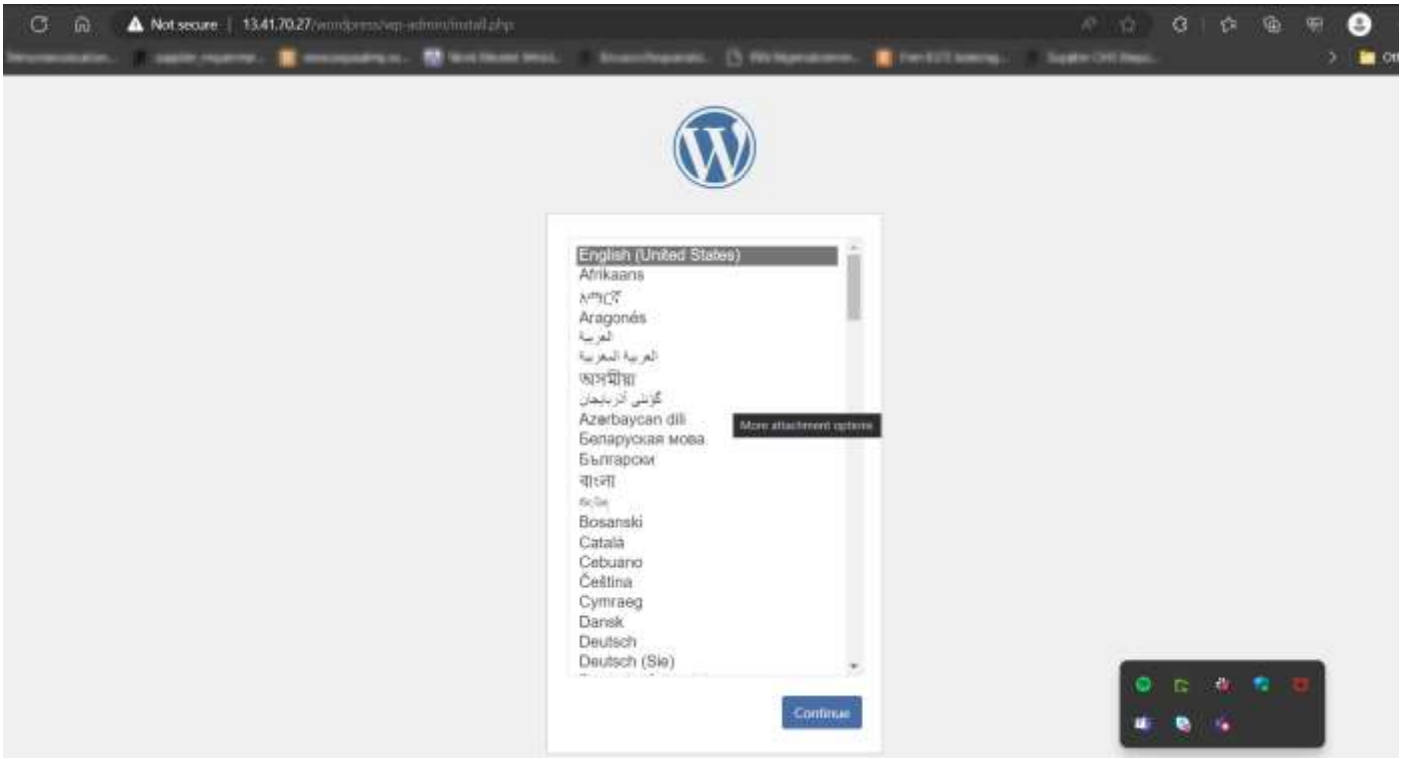
Try to access from your browser the link to your WordPress http://<Web-Server-Public-IP-Address>/wordpress/ and connect to the WordPress DB with the credentials

Database name

Database username

Database password





Congratulations and thank you for following through!!!