

# WEB SOLUTION WITH WORDPRESS

The 3-tier setup requirement.

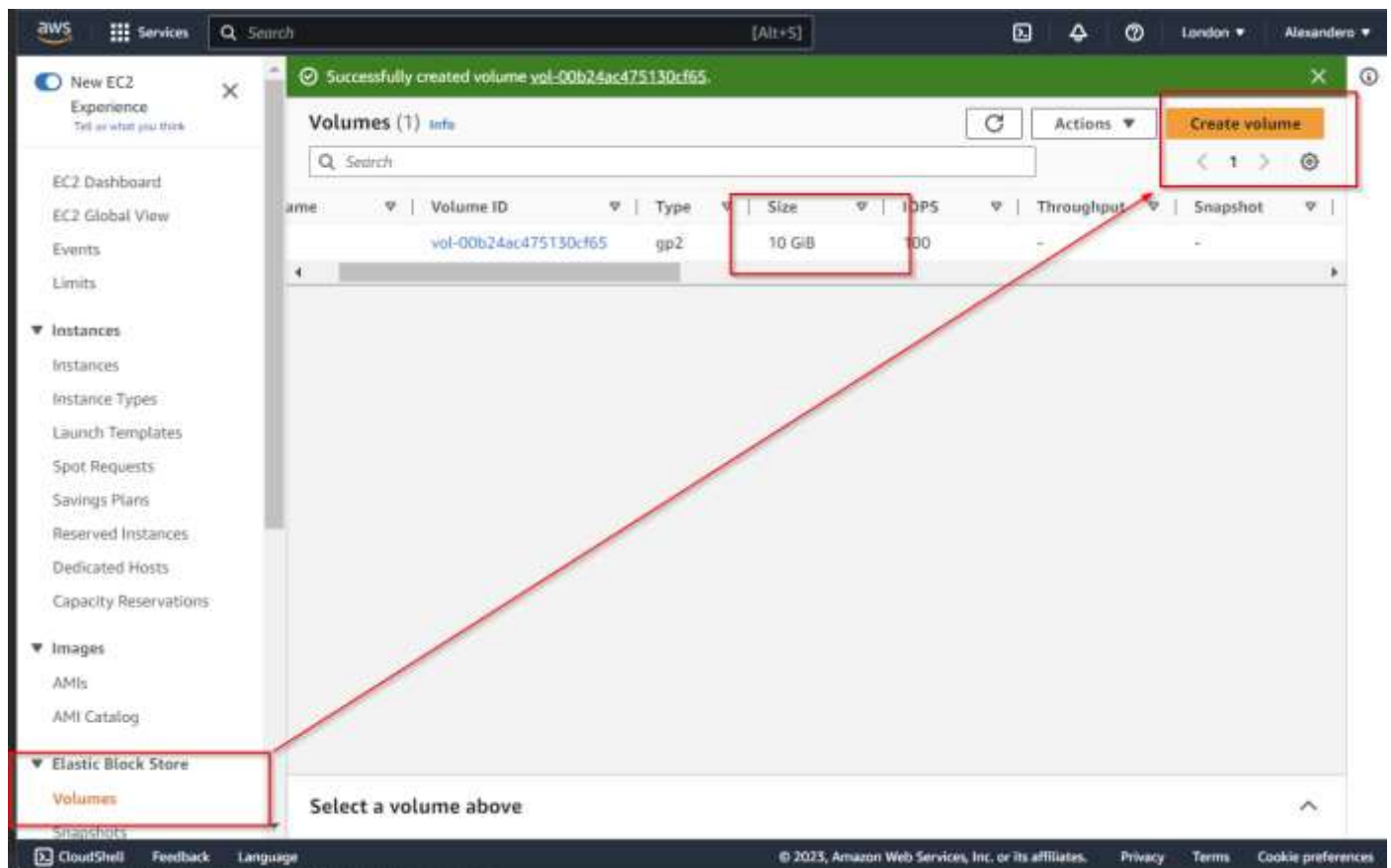
1. A computer to serve as a client.
2. An AWS EC2 Linux server as a web server (on this, we will install the WordPress).
3. An AWS EC2 Linux server as a database (DB) server

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

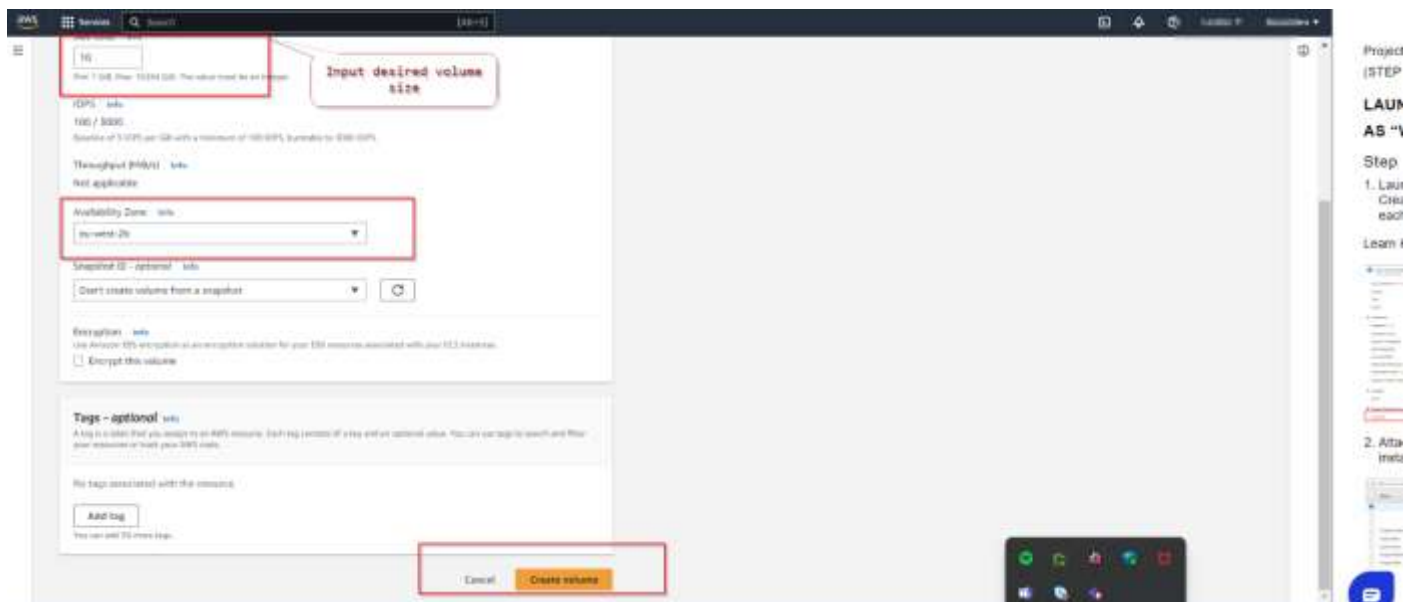
## PREPARE THE WEBSERVER

### Step 1.

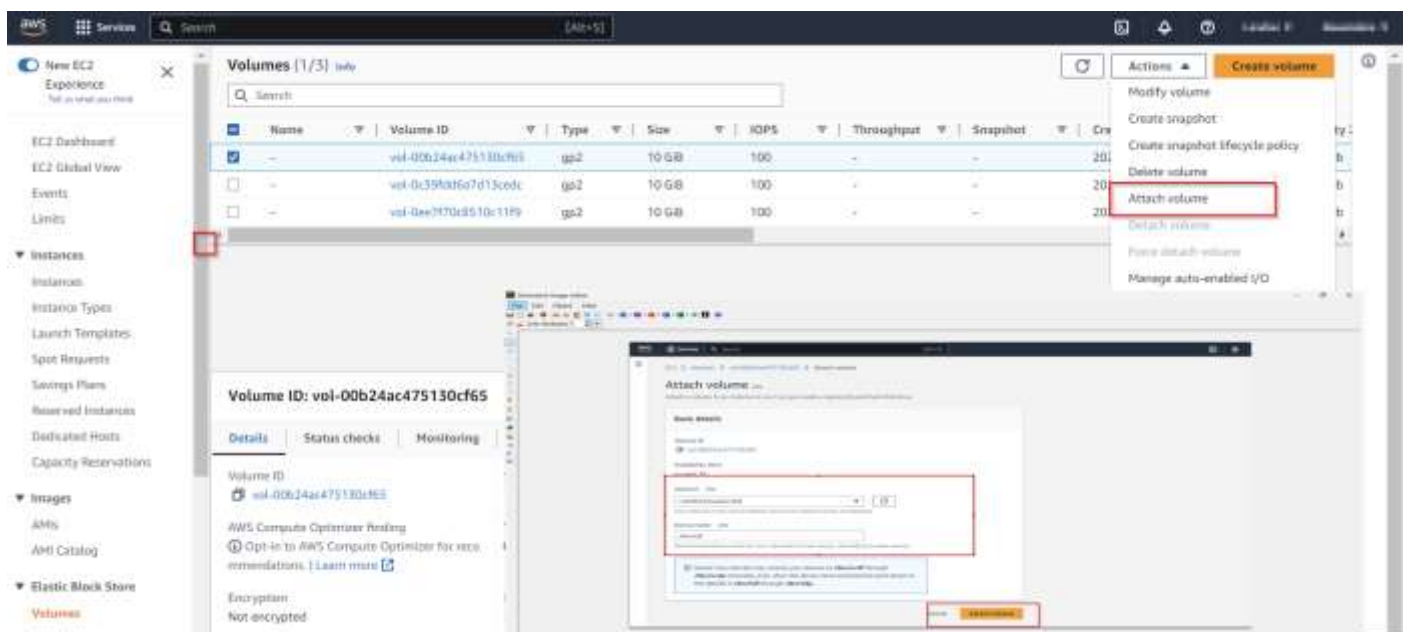
Create 3 volumes of 10GB size on AWS elastic block store (EBS), Launch an EC2 instance on AWS that will serve as webserver and attach the 3 volumes to the instance.



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
```

```
[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
xvdg        202:96    0   10G  0 disk
xvdh        202:112  0   10G  0 disk
[ec2-user@ip-172-31-40-134 ~]$
```

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.

```

[ec2-user@ip-172-31-40-134 ~]$ sudo gdisk /dev/xvdf
GPT fdisk (gdisk) version 1.0.7

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

Creating new GPT entries in memory.

Command (? for help): n
Partition number (1-128, default 1): 1
First sector (34-20971486, default = 2048) or {+}-size{M}:
Last sector (2048-20971486, default = 20971486) or {+}-size{M}:
Current type is 8300 (Linux filesystem)
Hex code or GUID (L to show codes, Enter = 8300): 8e00
Changed type of partition to '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-172-31-40-134 ~]$

```

```

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-172-31-40-134 ~]$ 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    1G  0 part
├─xvdg      202:96   0    1G  0 disk
├─xvdg1     202:97   0    1G  0 part
└─xvdh      202:112  0    1G  0 disk
    └─xvdh1  202:113  0    1G  0 part
[ec2-user@ip-172-31-40-134 ~]$

```

#### Step 4.

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

```
$ sudo yum install lvm2 -y
```

```

Downloading Packages:
(1/6): libaio-0.3.111-13.el9.x86_64.rpm                               441 kB/s | 26 kB   00:00
(2/6): device-mapper-event-1.02.187-7.el9.x86_64.rpm                2.7 MB/s | 36 kB   00:00
(3/6): device-mapper-persistent-data-0.9.0-13.el9.x86_64.rpm        9.0 MB/s | 786 kB   00:00
(4/6): lvm2-2.03.17-7.el9.x86_64.rpm                                14 MB/s | 1.5 MB   00:00
(5/6): device-mapper-event-libs-1.02.187-7.el9.x86_64.rpm           1.3 MB/s | 34 kB   00:00
(6/6): lvm2-libs-2.03.17-7.el9.x86_64.rpm                           20 MB/s | 1.0 MB   00:00
Total                                                                22 MB/s | 3.4 MB   00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing                : 1/6
  Installing               : device-mapper-event-libs-9:1.02.187-7.el9.x86_64 1/6
  Installing               : libaio-0.3.111-13.el9.x86_64                  2/6
  Installing               : device-mapper-persistent-data-0.9.0-13.el9.x86_64 3/6
  Installing               : device-mapper-event-9:1.02.187-7.el9.x86_64    4/6
  Running scriptlet: device-mapper-event-9:1.02.187-7.el9.x86_64          4/6
Created symlink /etc/systemd/system/sockets.target.wants/dm-event.socket → /usr/lib/systemd/system/dm-event.socket.

  Installing               : lvm2-libs-9:2.03.17-7.el9.x86_64              5/6
  Installing               : lvm2-9:2.03.17-7.el9.x86_64                  6/6
  Running scriptlet: lvm2-9:2.03.17-7.el9.x86_64                          6/6
Created symlink /etc/systemd/system/sysinit.target.wants/lvm2-monitor.service → /usr/lib/systemd/system/lvm2-monitor.service.
Created symlink /etc/systemd/system/sysinit.target.wants/lvm2-lvmpolld.socket → /usr/lib/systemd/system/lvm2-lvmpolld.socket.

  Verifying                : libaio-0.3.111-13.el9.x86_64                  1/6
  Verifying                : device-mapper-persistent-data-0.9.0-13.el9.x86_64 2/6
  Verifying                : lvm2-9:2.03.17-7.el9.x86_64                  3/6
  Verifying                : device-mapper-event-9:1.02.187-7.el9.x86_64    4/6
  Verifying                : lvm2-libs-9:2.03.17-7.el9.x86_64              5/6
  Verifying                : device-mapper-event-libs-9:1.02.187-7.el9.x86_64 6/6
Installed products updated.

Installed:
  device-mapper-event-9:1.02.187-7.el9.x86_64  device-mapper-event-libs-9:1.02.187-7.el9.x86_64  device-mapper-persistent-data-0.9.0-13.el9.x86_64
  libaio-0.3.111-13.el9.x86_64                lvm2-9:2.03.17-7.el9.x86_64                lvm2-libs-9:2.03.17-7.el9.x86_64

Complete!
[ec2-user@ip-172-31-40-134 ~]$

```

Mark each of the 3 disks as physical volumes (PVs) to be used by the logical volume manager LVM. Use the `pvcreeate` 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: 4a80clad-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="2d7d9c8b-8e0f-4e92-8534-45733aa3dc68" TYPE="xfs" PARTUUID="6264d520-3fb9-423f-8ab8-7a8a8e3d3562"
/dev/mapper/webdata--vg--logs--lv: UUID="4a88c1ad-5c3a-4a22-9cfa-21513c6e91ac" 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-398c3aaf9e68" TYPE="xfs" PARTUUID="cb87c243-bc44-4717-853e-28852021225b"
/dev/xvda1: PARTUUID="fac7f1fb-3e8d-4137-a512-961de89a5549"
/dev/xvdb1: UUID="vAm0fr-0Tym-k7XE-Je14-67pH-3C82-E9t71V" TYPE="LVM2_member" PARTLABEL="Linux LVM" PARTUUID="e3a33947-8ba3-4e3a-93d5-96cd4869d462"
/dev/xvdf1: UUID="17vQtP-n1Az-IUIr-5tom-svfw-vrhk-M95svt" TYPE="LVM2_member" PARTLABEL="Linux LVM" PARTUUID="66d71501-f9f8-4bd5-a728-6b"
/dev/mapper/webdata--vg--apps--lv: UUID="5a3ca932-8b76-481b-8697-9e564788b719" TYPE="ext4"
/dev/xvdp1: UUID="4HwCfh-VeIu-yY8F-8ZnF-rMA2-u1Xv-tvQA4M" TYPE="LVM2_member" PARTLABEL="Linux LVM" PARTUUID="65eac6ff-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=4a80clad-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-b      dm-a heap/      dr1/
[ec2-user@ip-172-31-44-127 ~]$ sudo mkfs -t ext4 /dev/database-vg/database-lv
mkfs: 1.46.5 (38-Dec-2021)
Creating filesystem with 1310720 4k blocks and 327680 inodes
Filesystem UUID: 133a51d4-5b14-4f8e-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-8ab8-7a0a0e3d3562"
/dev/xvda2: SEC_TYPE="msdos" UUID="7877-95E7" 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-BQJE-0InR-0UxM-aWzCfn" TYPE="LVM2_member" PARTLABEL="Linux LVM" PARTUUID="9c7ab100-2d46-483c-9d08-53a58a32a022"
/dev/xvdf1: UUID="kKhzy0-LIG1-7ECK-Ue27-xSfn-megl-10Jois" TYPE="LVM2_member" PARTLABEL="Linux LVM" PARTUUID="b0abd6f0-b42e-45be-a244-10c1ec0e9608"
/dev/mapper/database--vg--database--lv: UUID="133a51d4-5b14-4f8e-9970-70f427807729" TYPE="ext4"
/dev/xvdg1: UUID="R89PHH-y3pX-NLqz-BetM-Aw06-NyGX-d7oNuw" TYPE="LVM2_member" PARTLABEL="Linux LVM" PARTUUID="1a0fc0b0-3385-464d-9297-20a6a2ec468f"
[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
```

```
Verifying      : nginx-filesystem-1:1.20.1-14.el9.noarch
Verifying      : mod_http2-1.15.19-4.el9_2.5.x86_64
Verifying      : mod_lua-2.4.53-11.el9_2.5.x86_64
Verifying      : httpd-2.4.53-11.el9_2.5.x86_64
Verifying      : httpd-core-2.4.53-11.el9_2.5.x86_64
Verifying      : httpd-tools-2.4.53-11.el9_2.5.x86_64
Verifying      : apr-util-1.6.1-20.el9_2.1.x86_64
Verifying      : apr-util-bdb-1.6.1-20.el9_2.1.x86_64
Verifying      : httpd-filesystem-2.4.53-11.el9_2.5.noarch
Verifying      : apr-util-openssl-1.6.1-20.el9_2.1.x86_64
Verifying      : mailcap-2.1.49-5.el9.noarch
Installed products updated.

Installed:
apr-1.7.0-11.el9.x86_64
httpd-2.4.53-11.el9_2.5.x86_64
libxslt-1.1.34-9.el9.x86_64
nginx-filesystem-1:1.20.1-14.el9.noarch
php-common-8.0.27-1.el9_1.x86_64
php-opcache-8.0.27-1.el9_1.x86_64
wget-1.21.1-7.el9.x86_64

apr-util-1.6.1-20.el9_2.1.x86_64
httpd-core-2.4.53-11.el9_2.5.x86_64
mailcap-2.1.49-5.el9.noarch
oniguruma-6.9.6-1.el9_3.x86_64
php-fpm-8.0.27-1.el9_1.x86_64
php-pdo-8.0.27-1.el9_1.x86_64

apr-util-bdb-1.6.1-20.el9_2.1.x86_64
httpd-filesystem-2.4.53-11.el9_2.5.noarch
mod_http2-1.15.19-4.el9_2.4.x86_64
php-8.0.27-1.el9_1.x86_64
php-abstract-8.0.27-1.el9_1.x86_64
php-xml-8.0.27-1.el9_1.x86_64

apr-util-openssl-1.6.1-20.el9_2.1.x86_64
httpd-tools-2.4.53-11.el9_2.5.x86_64
mod_lua-2.4.53-11.el9_2.5.x86_64
php-cli-8.0.27-1.el9_1.x86_64
php-mysqlnd-8.0.27-1.el9_1.x86_64
redhat-logos-httpd-90.4-1.el9.noarch

Complete!
[ec2-user@ip-172-31-44-127 ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service + /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-172-31-44-127 ~]$ sudo systemctl start httpd
[ec2-user@ip-172-31-44-127 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Drop-In: /usr/lib/systemd/system/httpd.service.d
            └─php-fpm.conf
   Active: active (running) since Tue 2023-06-13 23:29:33 UTC; 6s ago
     Docs: man:httpd.service(8)
    Main PID: 15428 (httpd)
   Status: "Started, listening on: port 80"
     Tasks: 213 (limit: 4421)
    Memory: 37.1M
       CPU: 74ms
    CGroup: /system.slice/httpd.service
            └─15428 /usr/sbin/httpd -DFOREGROUND
              └─15435 /usr/sbin/httpd -DFOREGROUND
                └─15436 /usr/sbin/httpd -DFOREGROUND
                  └─15437 /usr/sbin/httpd -DFOREGROUND
                    └─15438 /usr/sbin/httpd -DFOREGROUND

Jun 13 23:29:33 ip-172-31-44-127.eu-west-2.compute.internal systemd[1]: Starting The Apache HTTP Server...
Jun 13 23:29:33 ip-172-31-44-127.eu-west-2.compute.internal systemd[1]: Started The Apache HTTP Server.
Jun 13 23:29:33 ip-172-31-44-127.eu-west-2.compute.internal httpd[15428]: Server configured, listening on: port 80
[ec2-user@ip-172-31-44-127 ~]$
```

## 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 ~]$ sudo mkdir wordpress
[ec2-user@ip-172-31-44-127 ~]$ ls
wordpress
[ec2-user@ip-172-31-44-127 ~]$ cd wordpress/
[ec2-user@ip-172-31-44-127 wordpress]$ sudo wget http://wordpress.org/latest.tar.gz
--2023-06-13 23:47:20-- http://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 157.140.101.253
Connecting to wordpress.org (wordpress.org)|157.140.101.253|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://wordpress.org/latest.tar.gz [following]
--2023-06-13 23:47:21-- https://wordpress.org/latest.tar.gz
Connecting to wordpress.org (wordpress.org)|157.140.101.253|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 23020109 (22M) [application/octet-stream]
Saving to: 'latest.tar.gz'

latest.tar.gz                100%[=====] 21.95M  10.0MB/s   in 2.25s

2023-06-13 23:47:23 (10.0 MB/s) - 'latest.tar.gz' saved [23020109/23020109]

[ec2-user@ip-172-31-44-127 wordpress]$ sudo tar xzvf latest.tar.gz
wordpress/
wordpress/xmlrpc.php
wordpress/wp-blog-header.php
wordpress/readme.html
wordpress/wp-signup.php
wordpress/index.php
wordpress/wp-cron.php
wordpress/wp-config-sample.php
wordpress/wp-login.php
wordpress/wp-settings.php
wordpress/license.txt
wordpress/wp-content/
wordpress/wp-content/themes/
wordpress/wp-content/themes/twentytwentythree/
wordpress/wp-content/themes/twentytwentythree/theme.json
wordpress/wp-content/themes/twentytwentythree/parts/
wordpress/wp-content/themes/twentytwentythree/parts/footer.html
wordpress/wp-content/themes/twentytwentythree/parts/comments.html
wordpress/wp-content/themes/twentytwentythree/parts/header.html
wordpress/wp-content/themes/twentytwentythree/parts/post-meta.html
wordpress/wp-content/themes/twentytwentythree/patterns/

[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
```

```
[ec2-user@ip-172-31-44-127 ~]$ 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.

By default, MySQL comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y

- Dropping test database...

Success.

- Removing privileges on test database...

Success.

Reloading the privilege tables will ensure that all changes

affiliates. Other names may be trademarks of their respective owners.

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

```
mysql> create database wordpress;
Query OK, 1 row affected (0.00 sec)
```

```
mysql> show databases;
```

Database
information_schema
mysql
performance_schema
sys
wordpress

5 rows in set (0.00 sec)

```
mysql> CREATE USER 'myuser'@'172.31.40.134' IDENTIFIED BY 'password';
```

```
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> GRANT ALL PRIVILEGES ON wordpress.* TO 'myuser'@'172.31.40.134';
```

```
mysql> GRANT ALL ON wordpress.* TO 'myuser'@'172.31.40.134';
```

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''

```
mysql> GRANT ALL ON wordpress.* TO 'myuser'@'172.31.40.134' at line 1
```

```
mysql> GRANT ALL ON wordpress.* TO 'myuser'@'172.31.40.134';
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> flush privileges;
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> show databases;
```

Database
information_schema
mysql
performance_schema
sys
wordpress

5 rows in set (0.00 sec)

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

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

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use n

```
mysql>
```

mysql.user' at line 2

## 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
```

[illegible]

### 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

```

?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the installation.
 * You don't have to use the web site, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * Database settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://wordpress.org/documentation/article/editing-wp-config-php/
 *
 * @package WordPress
 */

// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** Database username */
define( 'DB_USER', 'myuser' );

/** Database password */
define( 'DB_PASSWORD', 'password' );

/** Database hostname */
define( 'DB_HOST', '172.31.44.127' );

/** Database charset to use in creating database tables */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this under any circumstances. If you are not sure,
    leave the default and do not use this option. */
define( 'DB_COLLATE', '' );

/**#@+
 * Authentication unique keys and salts.
 * wp-config.php [dos] 96L, 29958
 */
require_once ABSPATH . 'wp-settings.php';
[ec2-user@ip-172-31-40-134 wordpress]$ sudo systemctl restart httpd
[ec2-user@ip-172-31-40-134 wordpress]$ cd ..
[ec2-user@ip-172-31-40-134 html]$ ls
lost+found  wordpress
[ec2-user@ip-172-31-40-134 html]$ pwd
/var/www/html
[ec2-user@ip-172-31-40-134 html]$ ls
lost+found  wordpress

```

## 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)

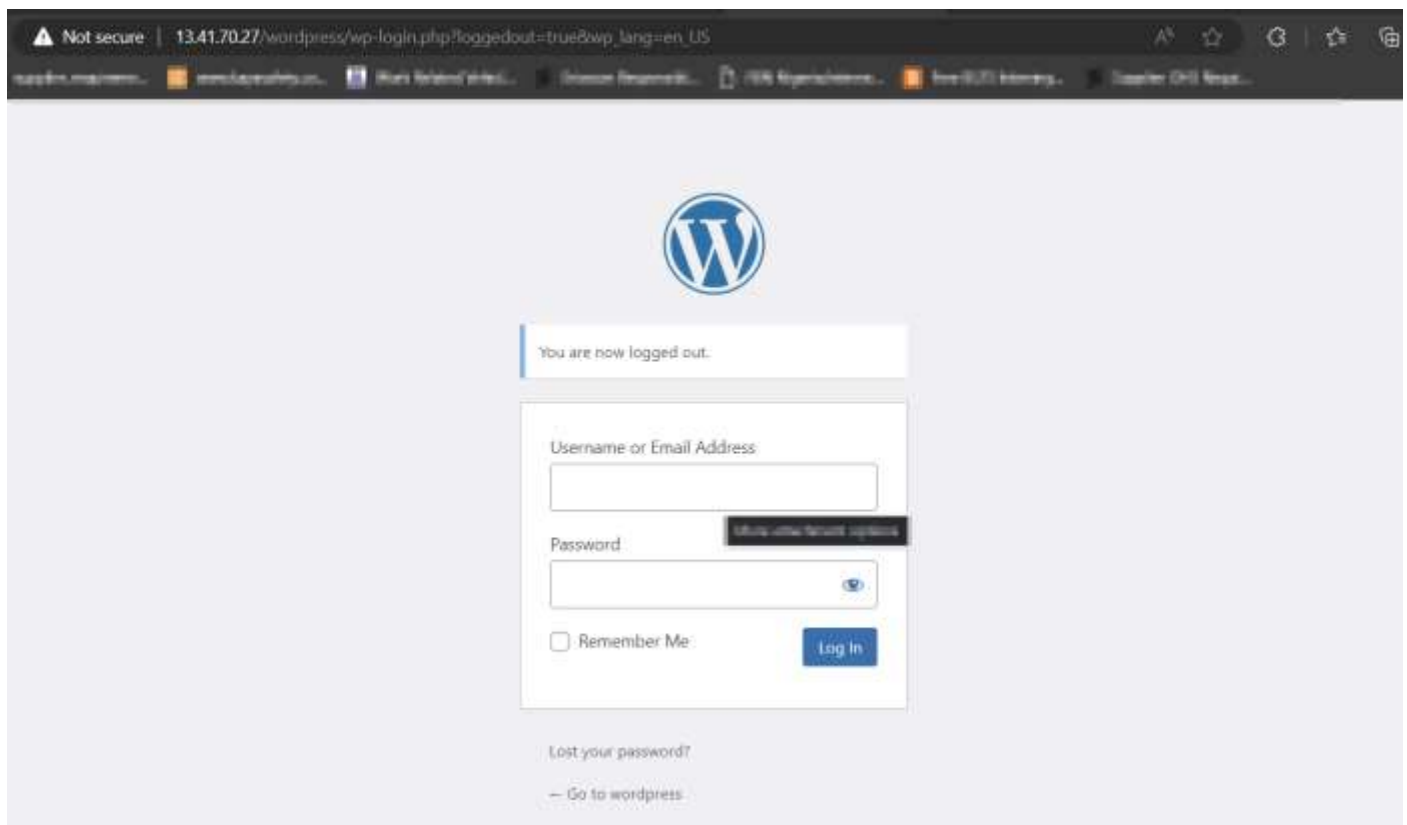
## Step 20.

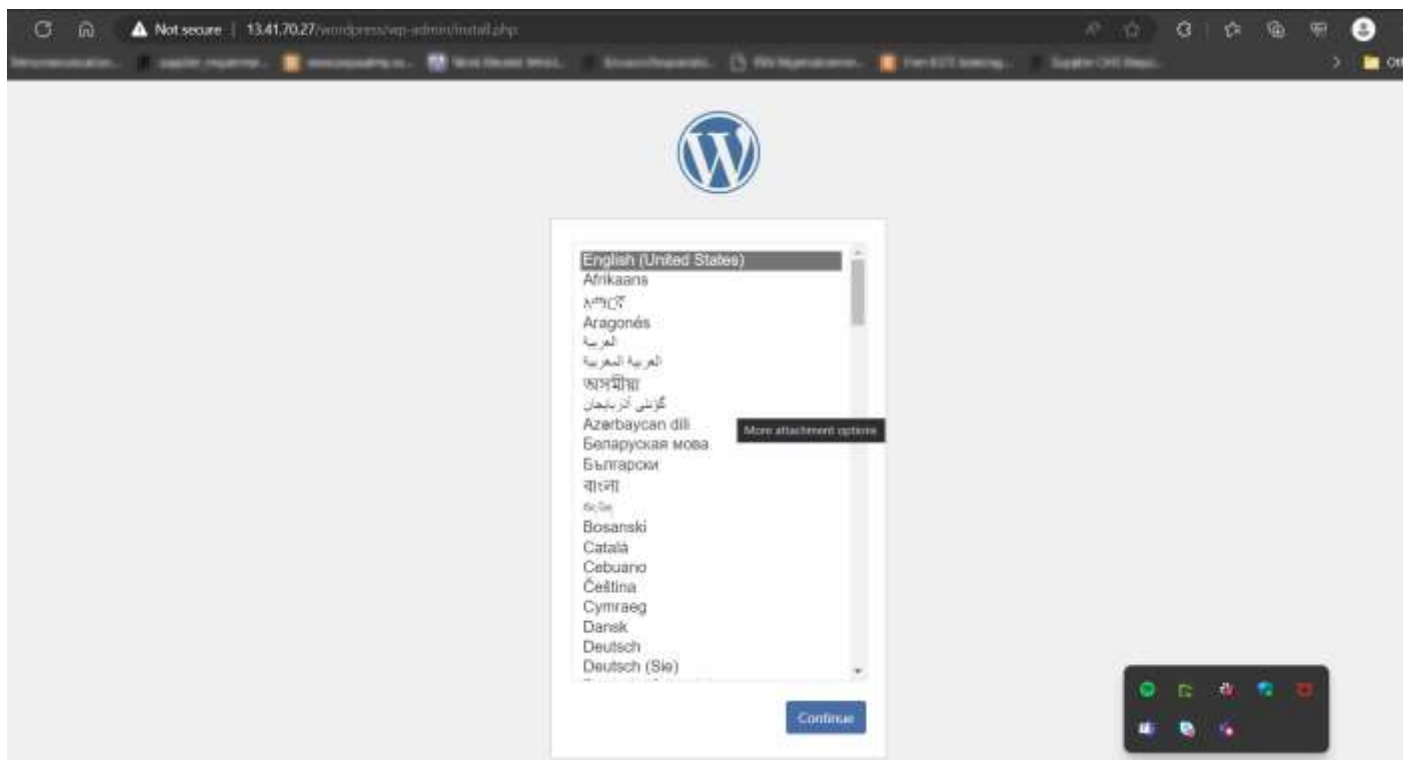
Try to access from your browser the link to your WordPress <http://<Web-Server-Public-IP-Address>/wordpress/> and connect to the WordPress DB with the credentials

Database name

Database username

Database password





Congratulations and thank you for following through!!!