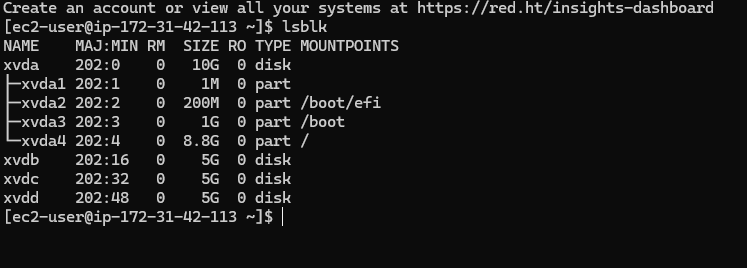
#### **Web Solution With WordPress**

#### Step 1 — Prepare a Web Server

1. Launch an EC2 instance that will serve as "Web Server". Create 3 volumes in the same AZ as your Web Server EC2.
2. Attach all three volumes one by one to your Web Server EC2 instance.
3. Open up the Linux terminal to begin configuration
4. Use [lsblk](https://man7.org/linux/man-pages/man8/lsblk.8.html) command to inspect what block devices are attached to the server. Notice names of your newly created devices. All devices in Linux reside in /dev/ directory. Inspect it with ls /dev/ and make sure you see all 3 newly created block devices there - their names will likely be xvdb, xvdc, xvdd.



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

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 branch segun-edits: p

Disk /dev/xvdf: 20971520 sectors, 10.0 GiB

Sector size (logical/physical): 512/512 bytes

Disk identifier (GUID): D936A35E-CE80-41A1-B87E-54D2044D160B

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 20971486

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 20971486 10.0 GiB 8E00 Linux LVM

Command (? for help): w

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING

PARTITIONS!!

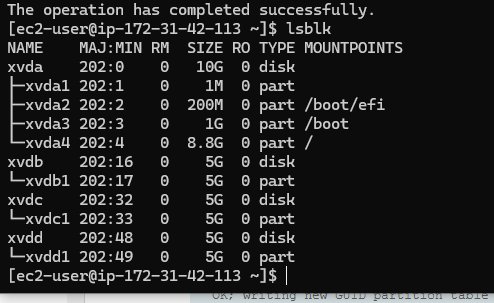
Do you want to proceed? (Y/N): yes

OK; writing new GUID partition table (GPT) to /dev/xvdf.

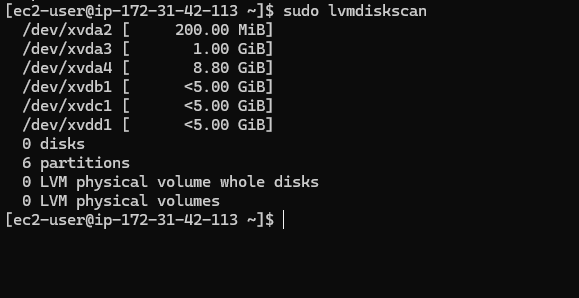
The operation has completed successfully.

Now, your changes has been configured succesfuly, exit out of the gdisk console and do the same for the remaining disks.

1. Use lsblk utility to view the newly configured partition on each of the 3 disks



1. .Install [lvm2](https://en.wikipedia.org/wiki/Logical_Volume_Manager_(Linux)) package using sudo yum install lvm2. Run sudo lvmdiskscan command to check for available partitions.



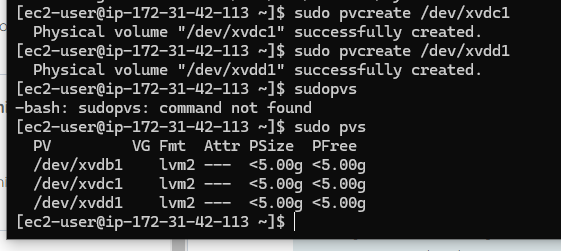
1. Use [pvcreate](https://linux.die.net/man/8/pvcreate) utility to mark each of 3 disks as physical volumes (PVs) to be used by LVM

sudo pvcreate /dev/xvdb1

sudo pvcreate /dev/xvdc1

sudo pvcreate /dev/xvdd1

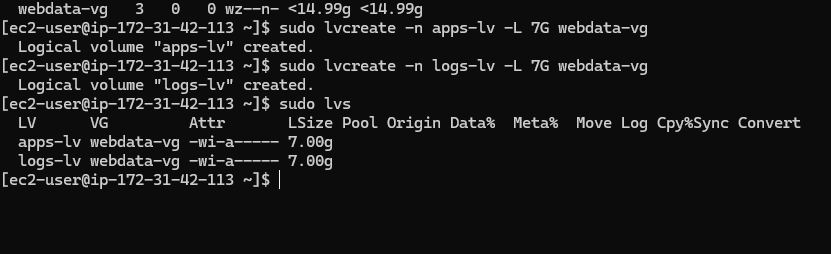
1. Verify that your Physical volume has been created successfully by running sudo pvs.



1. Use [vgcreate](https://linux.die.net/man/8/vgcreate) utility to add all 3 PVs to a volume group (VG). Name the VG **webdata-vg**

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

1. Verify that your VG has been created successfully by running sudo vgs



1. store data for logs.

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

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

1. Verify that your Logical Volume has been created successfully by running sudo lvs

Use mkfs.ext4 to format the logical volumes with [ext4](https://en.wikipedia.org/wiki/Ext4) filesystem

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

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

1. Create **/var/www/html** directory to store website files sudo mkdir -p /var/www/html
2. Create **/home/recovery/logs** to store backup of log data sudo mkdir -p /home/recovery/logs
3. Mount **/var/www/html** on **apps-lv** logical volume

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

1. Use [rsync](https://linux.die.net/man/1/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/

1. Mount **/var/log** on **logs-lv** logical volume. (*Note that all the existing data on /var/log will be deleted. That is why step 15 above is very important*)

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

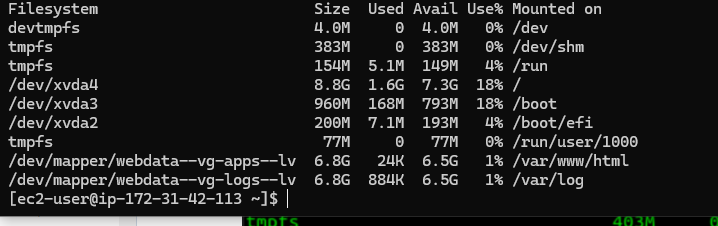
1. Restore log files back into **/var/log** directory

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

Verify the entire setup

sudo vgdisplay -v #view complete setup - VG, PV, and LV

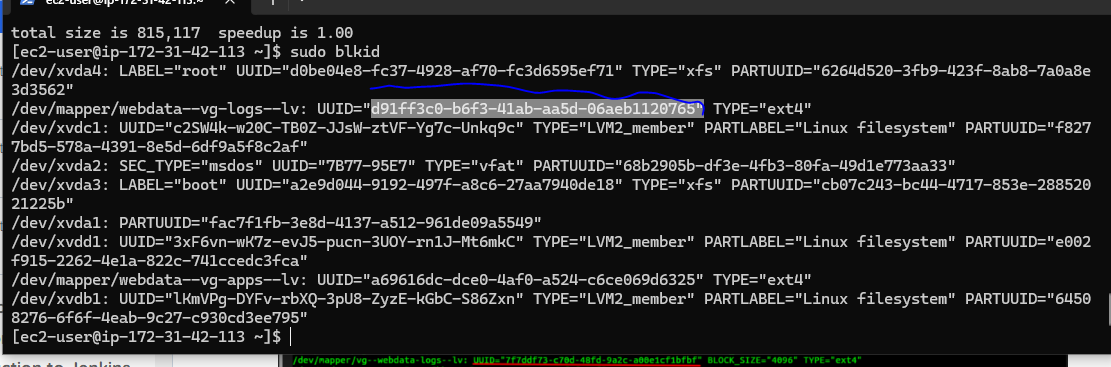
sudo lsblk



1. Update /etc/fstab file so that the mount configuration will persist after restart of the server.

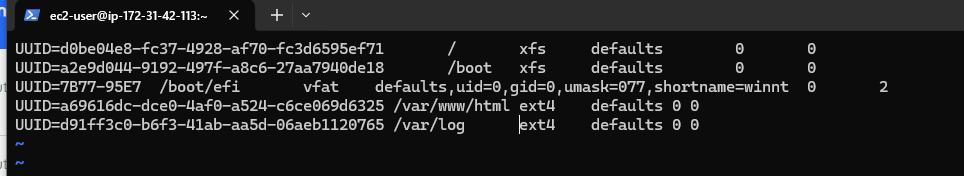
The UUID of the device will be used to update the /etc/fstab file;

sudo blkid



sudo vi /etc/fstab

Update /etc/fstab in this format using your own UUID and rememeber to remove the leading and ending quotes.

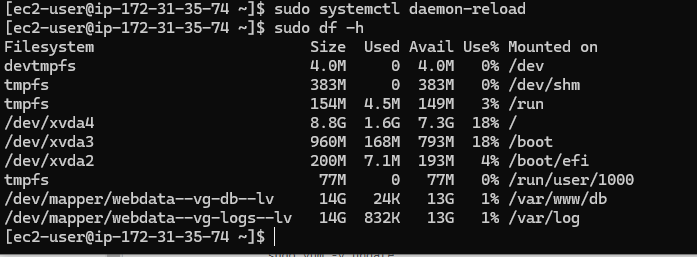


1. Test the configuration and reload the daemon  
   sudo mount -a  
   sudo systemctl daemon-reload
2. Verify your setup by running df -h,

#### Step 2 — Prepare the Database Server

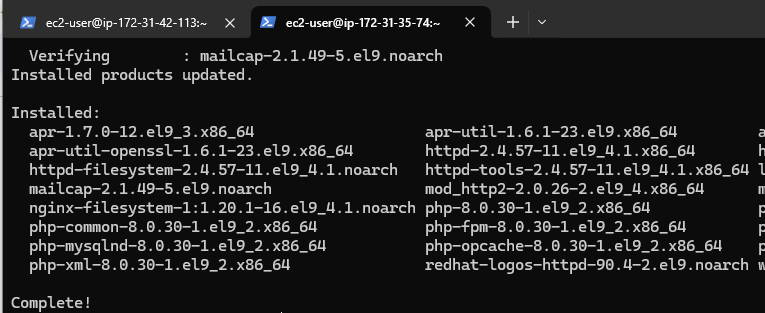
Launch a second RedHat EC2 instance that will have a role - 'DB Server' Repeat the same steps as for the Web Server, but instead of apps-lv create db-lv and mount it to /db directory instead of /var/www/html/.

After repeating the process, your output will be like this.



#### Step 3 — Install WordPress on your Web Server EC2

1. Update the repository  
   sudo yum -y update
2. Install wget, Apache and it's dependencies  
   sudo yum -y install wget httpd php php-mysqlnd php-fpm php-json
3. Start Apache  
   sudo systemctl enable httpd sudo systemctl start httpd
4. To install PHP and it's 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
5. Restart Apache  
   sudo systemctl restart httpd



1. Download wordpress and copy wordpress to /var/www/html  
   mkdir wordpress  
   cd wordpress  
   sudo wget http://wordpress.org/latest.tar.gz  
   sudo tar -xzvf latest.tar.gz  
   sudo rm -rf latest.tar.gz  
   cp wordpress/wp-config-sample.php wordpress/wp-config.php  
   cp -R wordpress /var/www/html/
2. 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

Use your browser to access the public IP of your webserver,if successful you will the red hat enterprise linux page.



#### Step 4 — Install MySQL on your DB Server EC2

sudo yum update

sudo yum install mysql-server

Verify that the service is up and running by using sudo systemctl status mysqld, if it is not running, restart the service and enable it so it will be running even after reboot:

sudo systemctl restart mysqld

sudo systemctl enable mysqld

#### Step 5 — Configure DB to work with WordPress

sudo mysql

CREATE DATABASE wordpress;

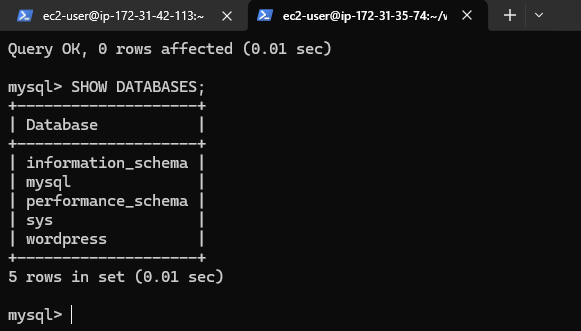
CREATE USER 'myuser'@'172.31.42.113' IDENTIFIED BY 'mypass';

GRANT ALL ON wordpress.\* TO 'myuser'@'172.31.42.113';

FLUSH PRIVILEGES;

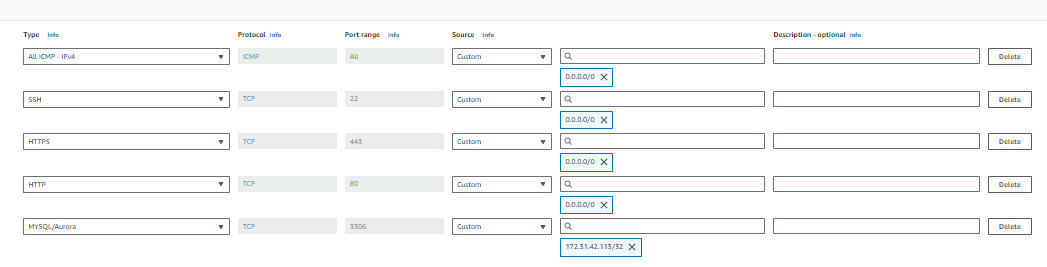
SHOW DATABASES;

exit



#### Step 6 — Configure WordPress to connect to remote database.

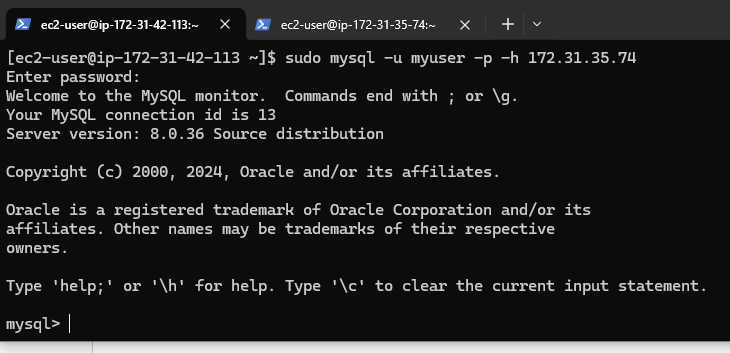
**Hint:** Do not forget to open MySQL port 3306 on DB Server EC2. For extra security, you shall allow access to the DB server **ONLY** from your Web Server's IP address, so in the Inbound Rule configuration specify source as /32

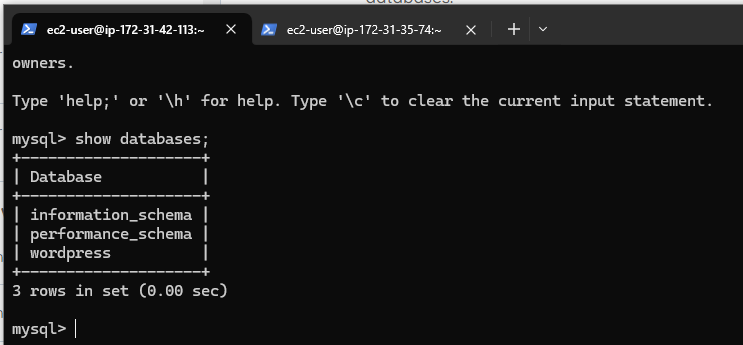


1. Install MySQL client and test that you can connect from your Web Server to your DB server by using mysql-client

sudo yum install mysql

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



1. Verify if you can successfully execute SHOW DATABASES; command and see a list of existing databases.
2. Change permissions and configuration so Apache could use WordPress:

sudo find //var/www/htm/ -type f -exec chmod 644 {} \;

sudo find //var/www/htm/ -type d -exec chmod 755 {} \;

sudo chmod 440 /var/www/html/wp-config.php

**Verify wp-config.php Settings**

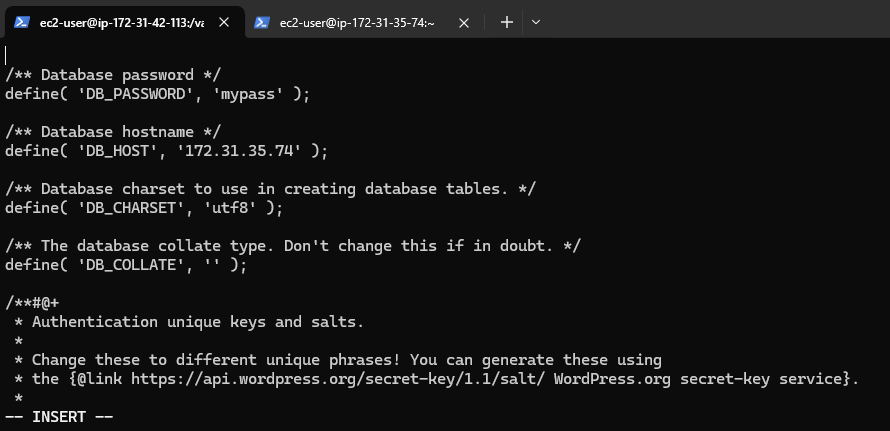
* Open your wp-config.php file located in the root of your WordPress installation.
* Ensure the database credentials (DB\_NAME, DB\_USER, DB\_PASSWORD, DB\_HOST) are correct.
* Sudo vi /var/www/html/wp-config.php

define('DB\_NAME', 'your\_database\_name');

define('DB\_USER', 'your\_username');

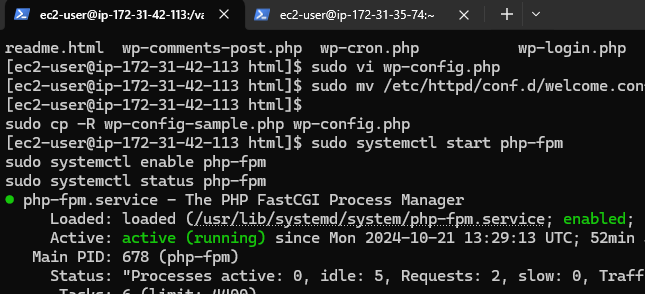
define('DB\_PASSWORD', 'your\_password');

define('DB\_HOST', 'localhost'); // or the correct host

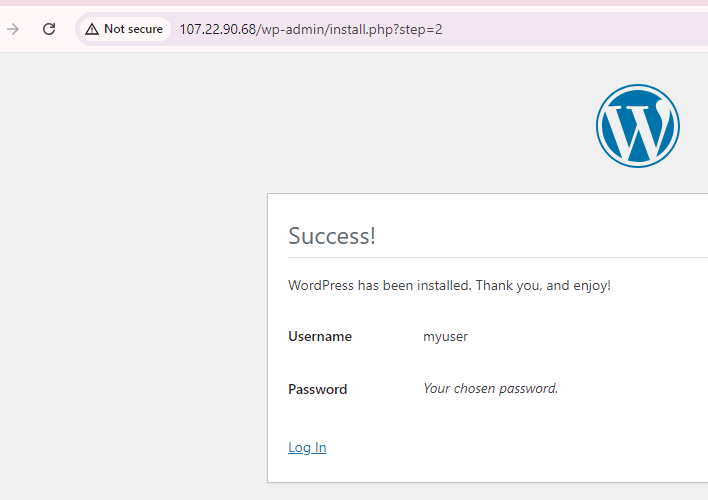


1. Enable TCP port 80 in Inbound Rules configuration for your Web Server EC2 (enable from everywhere 0.0.0.0/0 or from your workstation's IP)
2. **Disable the Apache default page:**the default page can be renamed.

sudo mv /etc/httpd/conf.d/welcome.conf /etc/httpd/conf.d/welcome.conf\_backup



1. Try to access from your browser the link to your WordPress http://<Web-Server-Public-IP-Address>/wordpress/



If you see this message - it means your WordPress has successfully connected to your remote MySQL database

