WEB SOLUTION WITH WORDPRESS

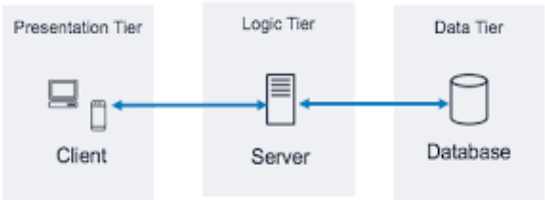
In this documentation we will be preparing storage infrastructure on two Linux servers and implementing a basic web solution using WordPress. WordPress is a free and open-source content management system written in **PHP** and paired with **MySQL** or **MariaDB** as its backend Relational Database Management System (RDBMS).

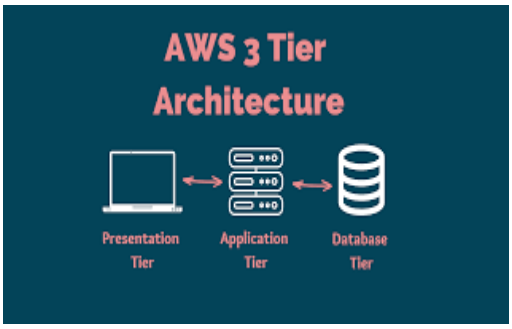
This documentation consists of two parts:

1. Configure storage subsystem for Web and Database servers based on Linux OS. The focus of this part is to give you practical experience of working with disks, partitions and volumes in Linux.
2. Install WordPress and connect it to a remote MySQL database server. This part of the project will solidify your skills of deploying Web and DB tiers of Web solution.

Three-tier Architecture

Three-tier Architecture is a client-server software architecture pattern that comprise of 3 separate layers.





1. **Presentation Layer** (PL): This is the user interface such as the client server or browser on your laptop.
2. **Business Layer** (BL): This is the backend program that implements business logic. Application or Webserver
3. **Data Access or Management Layer** (DAL): This is the layer for computer data storage and data access. Database Server or File System Server such as FTP server, or NFS Server

This documentation also ensures that the disks used to store files on the Linux servers are adequately partitioned and managed through programs such as gdisk and LVM respectively.

Three-Tier Setup

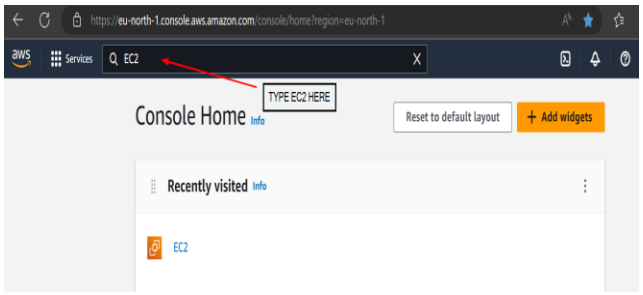
1. A Laptop or PC to serve as a client.
2. An EC2 Linux Server as a web server (This is where you will install WordPress)
3. An EC2 Linux server as a database (DB) server

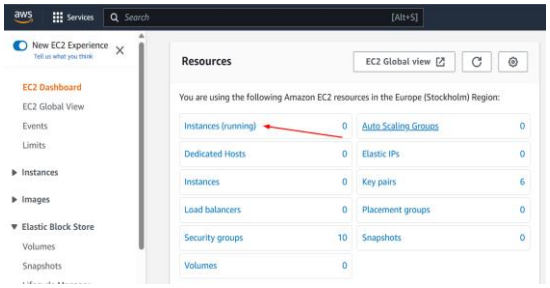
IMPLEMENTATION

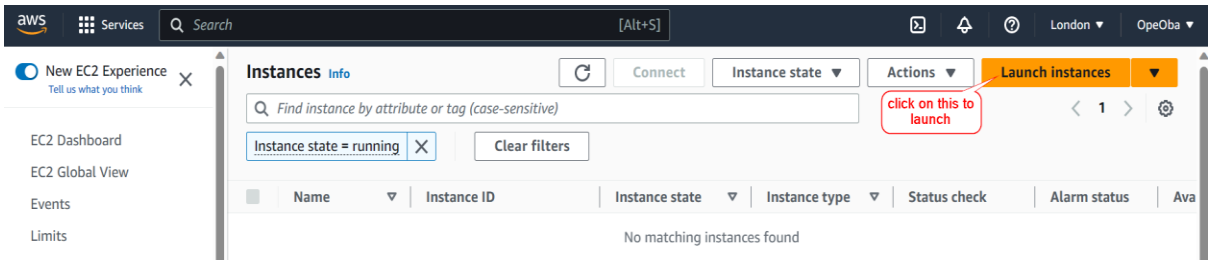
Launching our EC2 Servers

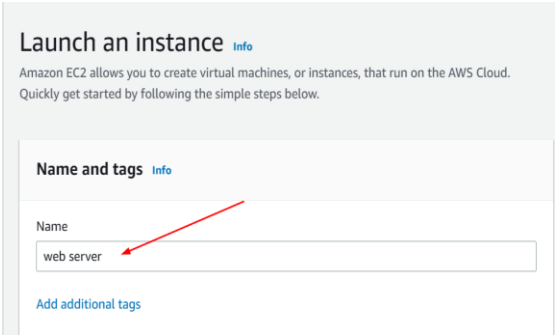
Open your PC browser and login into your account on https://aws.amazon.com/

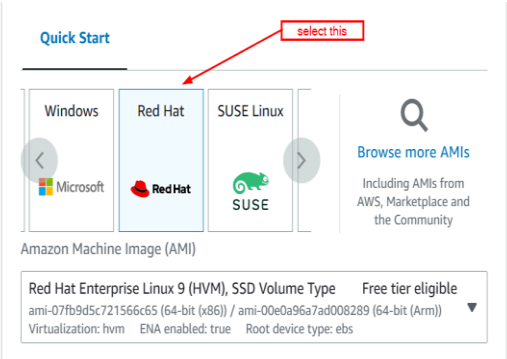
A region is selected by default (change to your closest location if necessary), and from the search bar type EC2 and click.





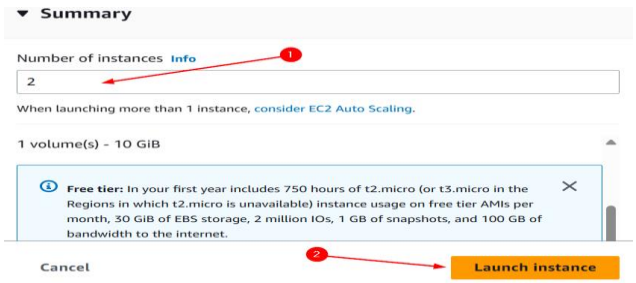


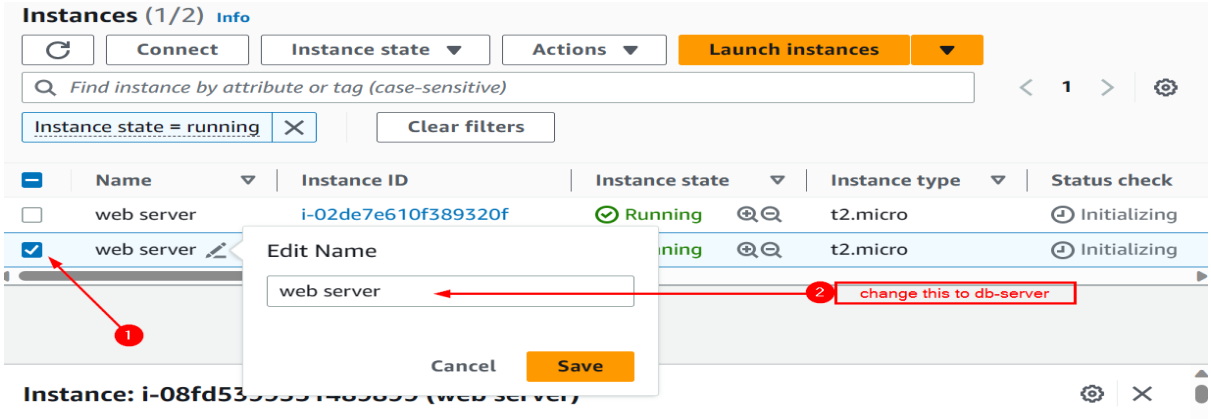




leave the rest as default and move to key pair name and create new key pair or select yours if you have a key pair already. Continue and leave the rest at default settings.

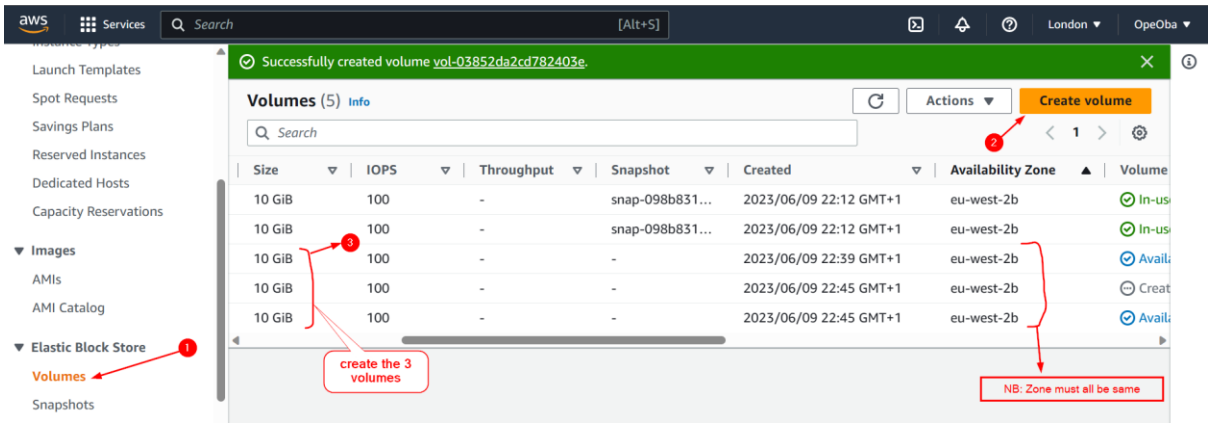
Here below, select 2 instances and then launch.



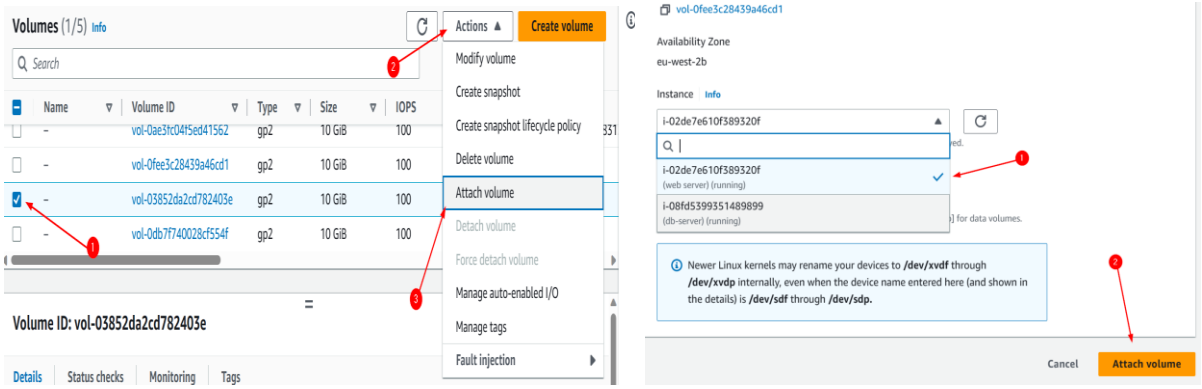


Step 1 — Prepare a Web Server

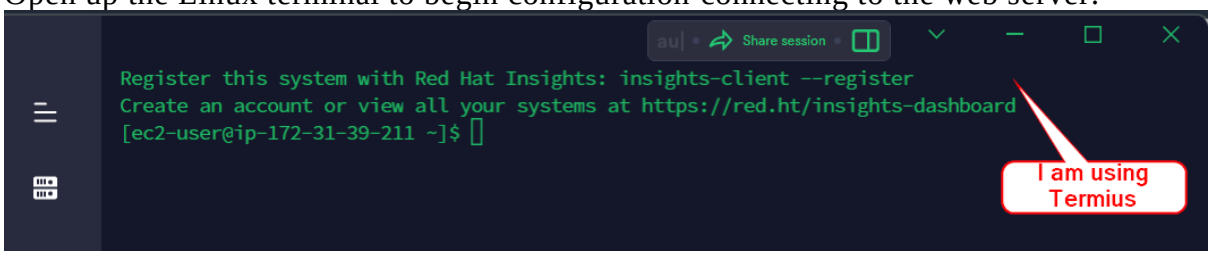
a. Launch an EC2 instance that will serve as "Web Server". Create 3 volumes in the same AZ as your Web Server EC2, each of 10 GiB.



b. Attach all three volumes one by one to your Web Server EC2 instance.



2. Open up the Linux terminal to begin configuration connecting to the web server.



3. Use lsblk 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 xvdf, xvdh, xvdg.

```
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
[ec2-user@ip-172-31-39-211 ~]$ 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-172-31-39-211 ~]$
```

4. Use df -h command to see all mounts and free space on your server.
5. Use gdisk utility to create a single partition on each of the 3 disks.

   sudo gdisk /dev/xvdf

```
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{KMGTP}:
Last sector (2048-20971486, default = 20971486) or {+-}size{KMGTP}:
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): p
Disk /dev/xvdf: 20971520 sectors, 10.0 GiB
Sector size (logical/physical): 512/512 bytes
Disk identifier (GUID): FAED9251-A691-42EE-9267-4E9C331E8628
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 O
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 successfully, exit out of the gdisk console and do the same for the remaining disks.

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

```
[ec2-user@ip-172-31-47-224 ~]$ 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
xvdg    202:96   0   10G  0 disk
└xvdg1 202:97   0   10G  0 part
xvdh    202:112  0   10G  0 disk
└xvdh1 202:113  0   10G  0 part
[ec2-user@ip-172-31-47-224 ~]$
```

7. Install lvm2 package using sudo yum install lvm2. Run sudo lvmdiskscan command to check for available partitions.

   Note: In RedHat/CentOS a different package manager is used to install, so we shall use yum command to install packages.

8. Use pvcreate utility to mark each of 3 disks as physical volumes (PVs) to be used by LVM

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

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

```
[ec2-user@ip-172-31-47-224 ~]$ 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.
  Creating devices file /etc/lvm/devices/system.devices
[ec2-user@ip-172-31-47-224 ~]$ 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-172-31-47-224 ~]$
```

10. Use 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
```

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

```
[ec2-user@ip-172-31-47-224 ~]$ sudo vgs
  VG          #PV #LV #SN Attr   VSize   VFree
  webdata-vg    3   0   0 wz--n- <29.99g <29.99g
[ec2-user@ip-172-31-47-224 ~]$
```

12. Use lvcreate utility to create 2 logical volumes. **apps-lv** (***Use half of the PV size***), and **logs-lv** ***Use the remaining space of the PV size***. **NOTE**: apps-lv will be used to store data for the Website while, logs-lv will be used to store data for logs.

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

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

```
[ec2-user@ip-172-31-47-224 ~]$ 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-172-31-47-224 ~]$
```

16. Verify the entire setup

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

```
sudo lsblk
```

```
  PV Status              allocatable
  Total PE / Free PE     2559 / 509

  PV Name                /dev/xvdf1
  PV UUID                jArl9U-RGGG-Qdnc-3oxP-qVFP-24lO-roh3SP
  PV Status              allocatable
  Total PE / Free PE     2559 / 0

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-logs--lv 253:1    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-apps--lv 253:0    0   14G  0 lvm
[ec2-user@ip-172-31-47-224 ~]$
```

15. Use mkfs.ext4 to format the logical volumes with ext4 filesystem

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

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

    sudo mkdir -p /var/www/html

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

    sudo mkdir -p /home/recovery/logs

18. Mount **/var/www/html** on **apps-lv** logical volume

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

19. 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/
```

20. 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
```

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

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

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

UPDATE THE `/ETC/FSTAB` FILE

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

sudo blkid

```
[ec2-user@ip-172-31-47-224 ~]$ sudo blkid
/dev/xvda4: LABEL="root" UUID="287d9c0b-0e0f-4e92-8534-45733aa3dc68" TYPE="xfs" PARTUUID="6264d
520-3fb9-423f-8ab8-7a0a8e3d3562"
/dev/mapper/webdata--vg-logs--lv: UUID="872e7274-c0d1-440c-98a3-5fbe059026c5" TYPE="ext4"
/dev/xvda2: SEC_TYPE="msdos" UUID="7B77-95E7" TYPE="vfat" PARTUUID="68b2905b-df3e-4fb3-80fa-49d
1e773aa33"
/dev/xvda3: LABEL="boot" UUID="7bc24af7-289d-4bce-b17e-300c3aafe968" TYPE="xfs" PARTUUID="cb07c
243-bc44-4717-853e-28852021225b"
/dev/xvdh1: UUID="bMOip9-3OnP-1EAE-YOFD-x2bd-kdgB-ZYGZEq" TYPE="LVM2_member" PARTLABEL="Linux L
VM" PARTUUID="62ce85db-b3a5-4eae-a33c-3d356d234063"
/dev/xvdf1: UUID="jArl9U-RGGG-Qdnc-3oxP-qVFP-24lO-roh3SP" TYPE="LVM2_member" PARTLABEL="Linux L
VM" PARTUUID="e15467eb-5e65-41da-bf43-63cc2cd8ce40"
/dev/mapper/webdata--vg-apps--lv: UUID="05c9f68c-b409-46e6-9a13-e0e6ac784982" TYPE="ext4"
/dev/xvdg1: UUID="aT0GR1-Og9B-91mm-GR0c-0I44-V1CE-UoWE0N" TYPE="LVM2_member" PARTLABEL="Linux L
VM" PARTUUID="23c223c6-8fb4-4f86-8aef-b42b0a7be2d2"
/dev/xvda1: PARTUUID="fac7f1fb-3e8d-4137-a512-961de09a5549"
[ec2-user@ip-172-31-47-224 ~]$
```

sudo vi /etc/fstab

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

```
UUID=287d9c0b-0e0f-4e92-8534-45733aa3dc68       /       xfs     defaults        0       0
UUID=7bc24af7-289d-4bce-b17e-300c3aafe968       /boot   xfs     defaults        0       0
UUID=7B77-95E7  /boot/efi       vfat    defaults,uid=0,gid=0,umask=077,shortname=winnt  0
 2


# mounts for wordpress webserver
UUID=05c9f68c-b409-46e6-9a13-e0e6ac784982 /var/www/html ext4 defaults 0 0
UUID=872e7274-c0d1-440c-98a3-5fbe059026c5 /var/log      ext4 defaults 0 0
```

1. Test the configuration and reload the daemon

```
sudo mount -a
sudo systemctl daemon-reload
```

2. Verify your setup by running df -h, output must look like this:

```
[ec2-user@ip-172-31-47-224 ~]$ 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/webdata--vg-apps--lv  14G   24K   13G   1% /var/www/html
/dev/mapper/webdata--vg-logs--lv  14G  976K   13G   1% /var/log
[ec2-user@ip-172-31-47-224 ~]$
```

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/.

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 depemdencies
   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
6. Download wordpress and copy wordpress to var/www/html

   mkdir wordpress

   cd wordpress

   sudo yum install wget

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

```
sudo tar xzvf latest.tar.gz
cd wordpress/
sudo cp -R wp-config-sample.php wp-config.php
cd ..
sudo cp -R wordpress/ /var/www/html/
cd /var/www/html
sudo rm -rf wordpress/
sudo rm -rf lost+found/
cd
cd worpress
sudo cp -R wordpress/. /var/www/html
cd /var/www/html
sudo yum install mysql-server
```

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:

```
[ec2-user@ip-172-31-32-222 ~]$ sudo systemctl status mysqld
○ mysqld.service - MySQL 8.0 database server
     Loaded: loaded (/usr/lib/systemd/system/mysqld.service; disabled; preset: disabled)
     Active: inactive (dead)
[ec2-user@ip-172-31-32-222 ~]$
```

```
sudo systemctl restart mysqld
sudo systemctl enable mysqld
```

```
ec2-13-40-76-185.eu-west-2.compute.amazonaws.com
  protobuf-lite-3.14.0-13.el9.x86_64

Complete!
[ec2-user@ip-172-31-13-78 html]$ sudo systemctl start mysqld
[ec2-user@ip-172-31-13-78 html]$ sudo systemctl enable mysqld
Created symlink /etc/systemd/system/multi-user.target.wants/mysqld.service → /usr/lib/systemd/system/
mysqld.service.
[ec2-user@ip-172-31-13-78 html]$ sudo systemctl status mysqld
● mysqld.service - MySQL 8.0 database server
     Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; preset: disabled)
     Active: active (running) since Tue 2023-06-13 22:42:05 UTC; 5min ago
   Main PID: 18694 (mysqld)
     Status: "Server is operational"
      Tasks: 38 (limit: 4421)
     Memory: 391.8M
        CPU: 4.987s
     CGroup: /system.slice/mysqld.service
             └─18694 /usr/libexec/mysqld --basedir=/usr

Jun 13 22:41:57 ip-172-31-13-78.eu-west-2.compute.internal systemd[1]: Starting MySQL 8.0 database s>
Jun 13 22:41:58 ip-172-31-13-78.eu-west-2.compute.internal mysql-prepare-db-dir[18621]: Initializing>
Jun 13 22:42:05 ip-172-31-13-78.eu-west-2.compute.internal systemd[1]: Started MySQL 8.0 database se>
lines 1-14/14 (END)
```

Step 5 — Configure DB to work with WordPress

```
sudo mysql
CREATE DATABASE wordpress;
CREATE USER `mark`@`%` IDENTIFIED WITH mysql_native_password BY
'pass';
GRANT ALL PRIVILEGES ON *.* TO 'mark'@'%' WITH GRANT OPTION;
FLUSH PRIVILEGES;
SHOW DATABASES;
exit
```

`sudo vi /etc/my.cnf` and configure your bind address.

```
ec2-18-170-69-121.eu-west-2.compute.amazonaws.com
#
# 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
~
-- INSERT --
```

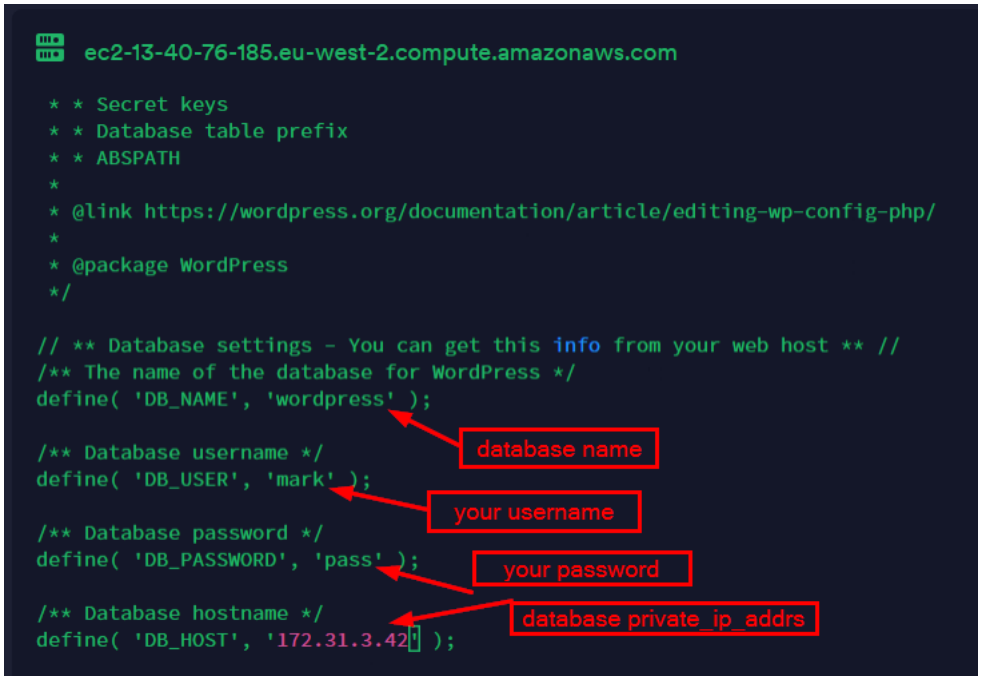
configure your <bind-address>

```
sudo systemctl restart mysqld
```

ON WEBSERVER

Configure your wp-config.php file

```
sudo vi wp-config.php
```

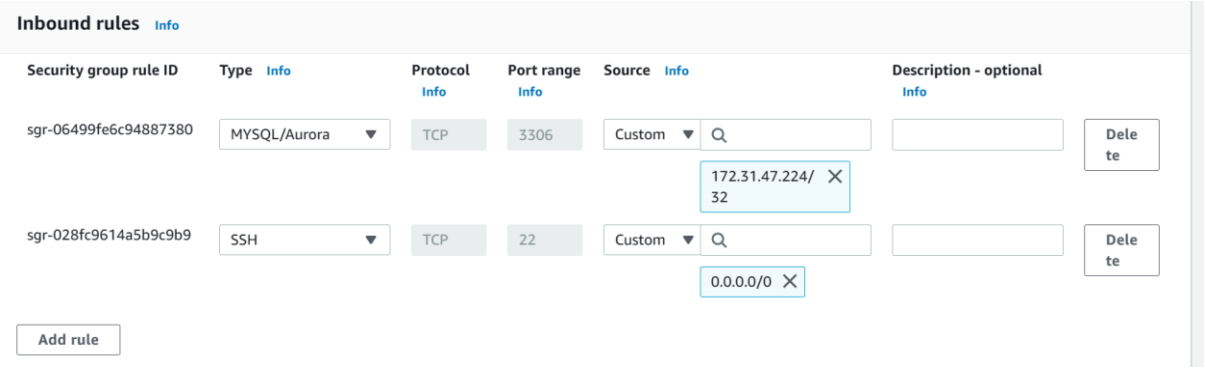



```
sudo systemctl restart httpd
```

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

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 mark -p -h <DB-Server-Private-IP-address>
```

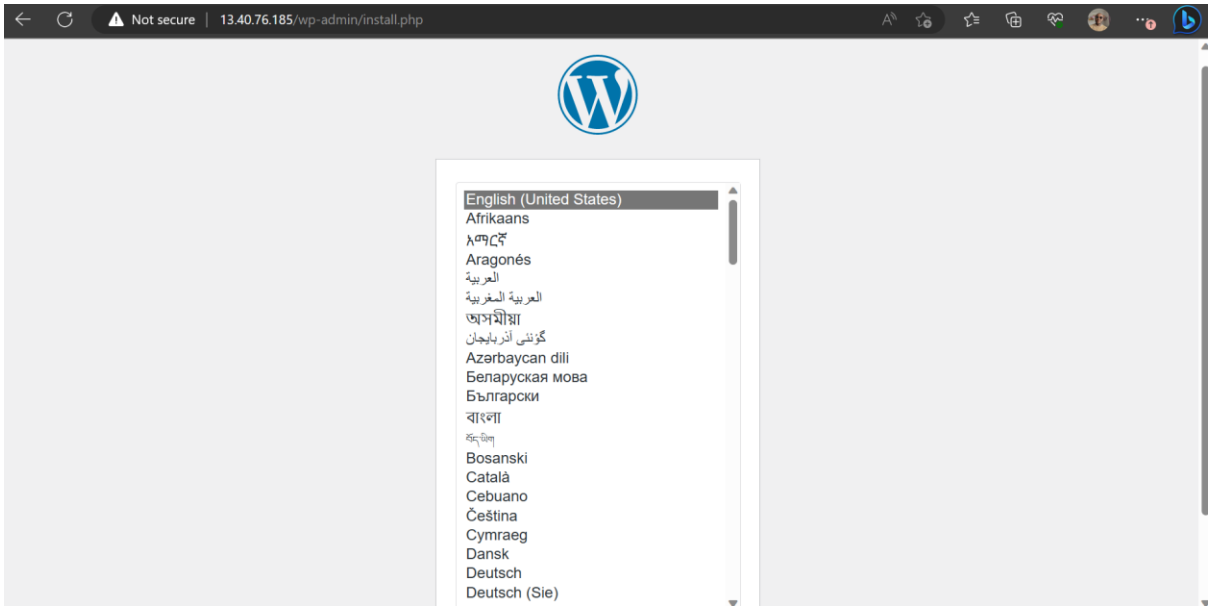
2. Verify if you can successfully execute SHOW DATABASES; command and see a list of existing databases.

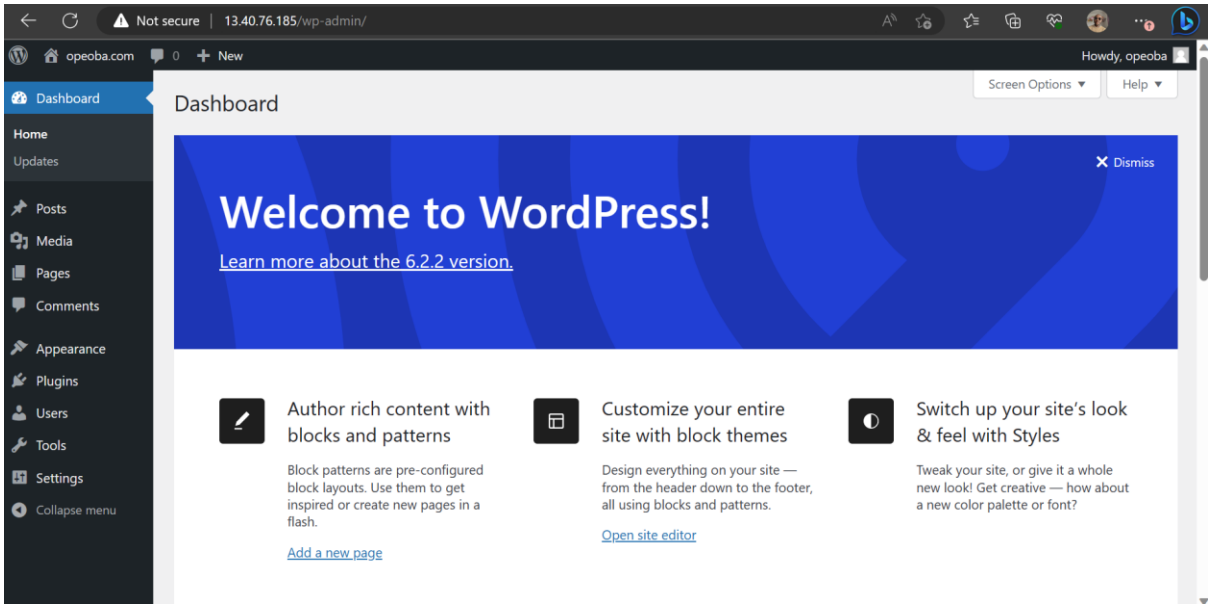
```
ec2-13-40-76-185.eu-west-2.compute.amazonaws.com

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 8.0.32 Source distribution

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| wordpress          |
+--------------------+
5 rows in set (0.01 sec)
```

3. Change permissions and configuration so Apache could use WordPress: Configure SELinux Policies

```
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
```

4. 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)
5. Try to access from your browser the link to your WordPress http://<Web-Server-Public-IP-Address>/


Not secure | 13.40.76.185/wp-admin/install.php
English (United States)
Afrikaans
Aragonés
العربية
العربية المغربية
অসমীয়া
Azərbaycan dili
Беларуская мова
Български
বাংলা
Bosanski
Català
Cebuano
Čeština
Cymraeg
Dansk
Deutsch
Deutsch (Sie)


Not secure | 13.40.76.185/wp-admin/
opeoba.com
New
Howdy, opeoba
Dashboard
Home
Updates
Posts
Media
Pages
Comments
Appearance
Plugins
Users
Tools
Settings
Collapse menu
Screen Options
Help
Dashboard
Dismiss
Welcome to WordPress!
Learn more about the 6.2.2 version.
Author rich content with blocks and patterns
Block patterns are pre-configured block layouts. Use them to get inspired or create new pages in a flash.
Add a new page
Customize your entire site with block themes
Design everything on your site — from the header down to the footer, all using blocks and patterns.
Open site editor
Switch up your site's look & feel with Styles
Tweak your site, or give it a whole new look! Get creative — how about a new color palette or font?


Thank You!!!


CONGRATULATIONS