WEB SOLUTION WITH WORDPRESS

In this project we would be implementing a basic web solution called WORDPRESS while creating storage infrastructures on 2 Linux servers

This project consists of two parts:

1) Configuration of storage systems focusing on working with disks, partitioning and creating volumes on Web and Data servers using the Linux Operating system

2) Deployment of Web and Database tier of Web solutions by connecting WORDPRESS to a remote MySQL server

It's important to have web server installed separated with WordPress as the CLIENT while we install the database server separately this helps us in

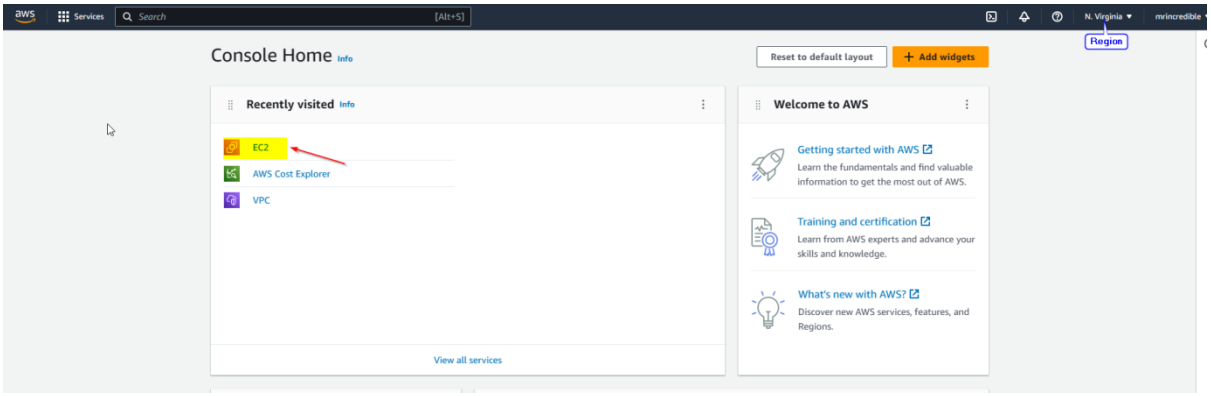
separating the 2 servers just to enhance security and avoid natural disaster recovery in case of any unforeseen circumstance so as to void losing so much information.

Pre-requisite for the projects is the following.

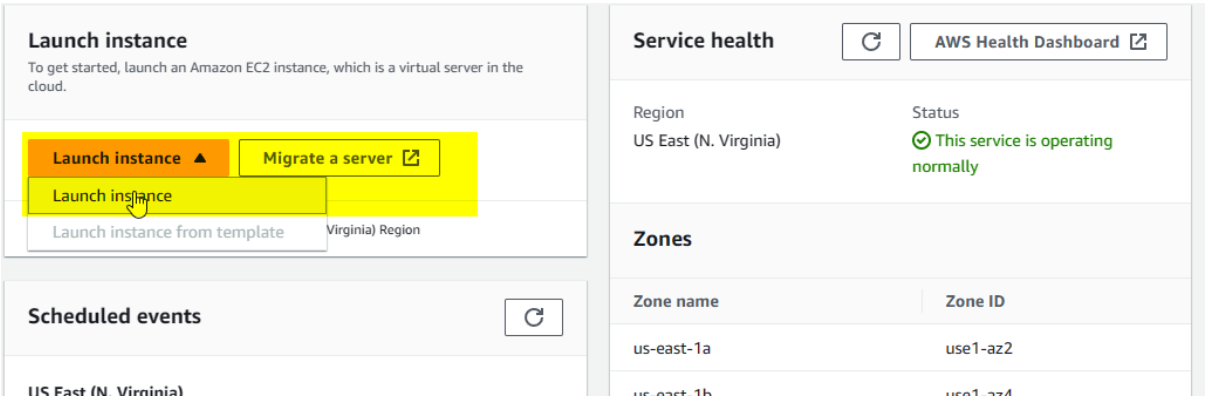
1) Fundamental Knowledge of Installing and downloading software
2) Basic Understanding of Linux Commands
3) AWS account login with 2 EC2 instances (Red Hat)
4) Webserver (WORDPRESS
5) Laptop or PC to serve as a client.
6) Database Server (MYSQL database server)
7) Internet connection

IMPLEMENTATION STEPS:

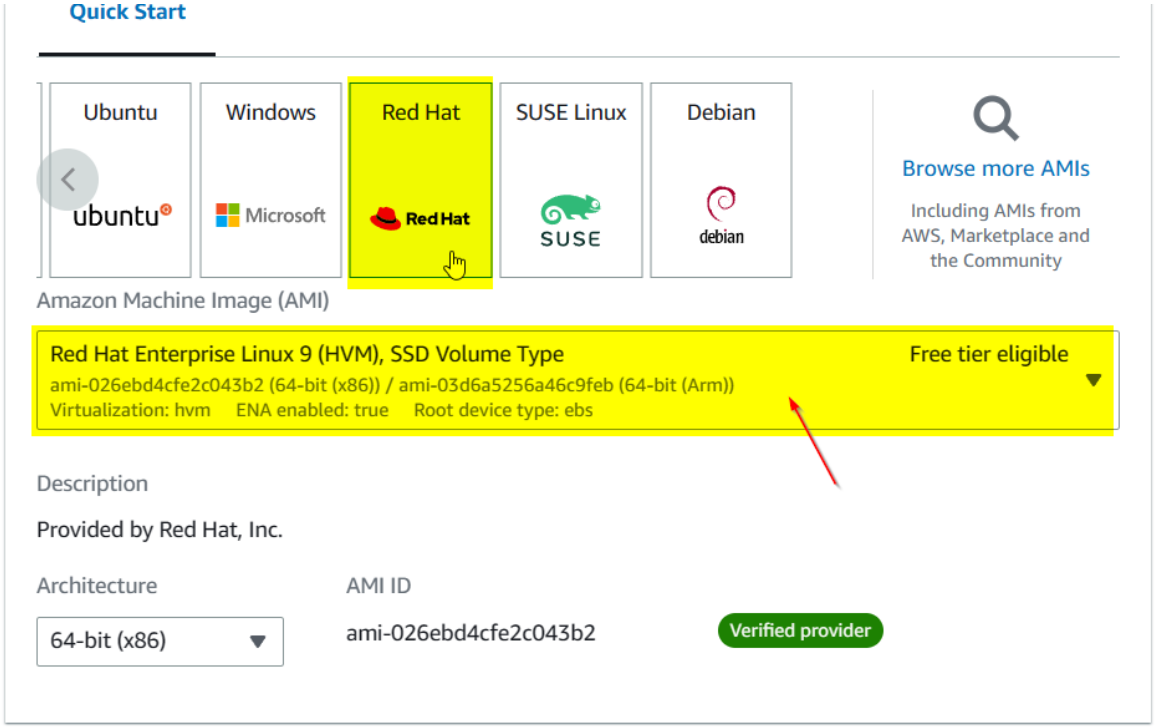
i) Ensure you login with your details to your AWS console via the https://aws.amazon.com
ii) Click on the EC2 link to create instances.



iii)Click on launch instance dropdown button and select launch instance.

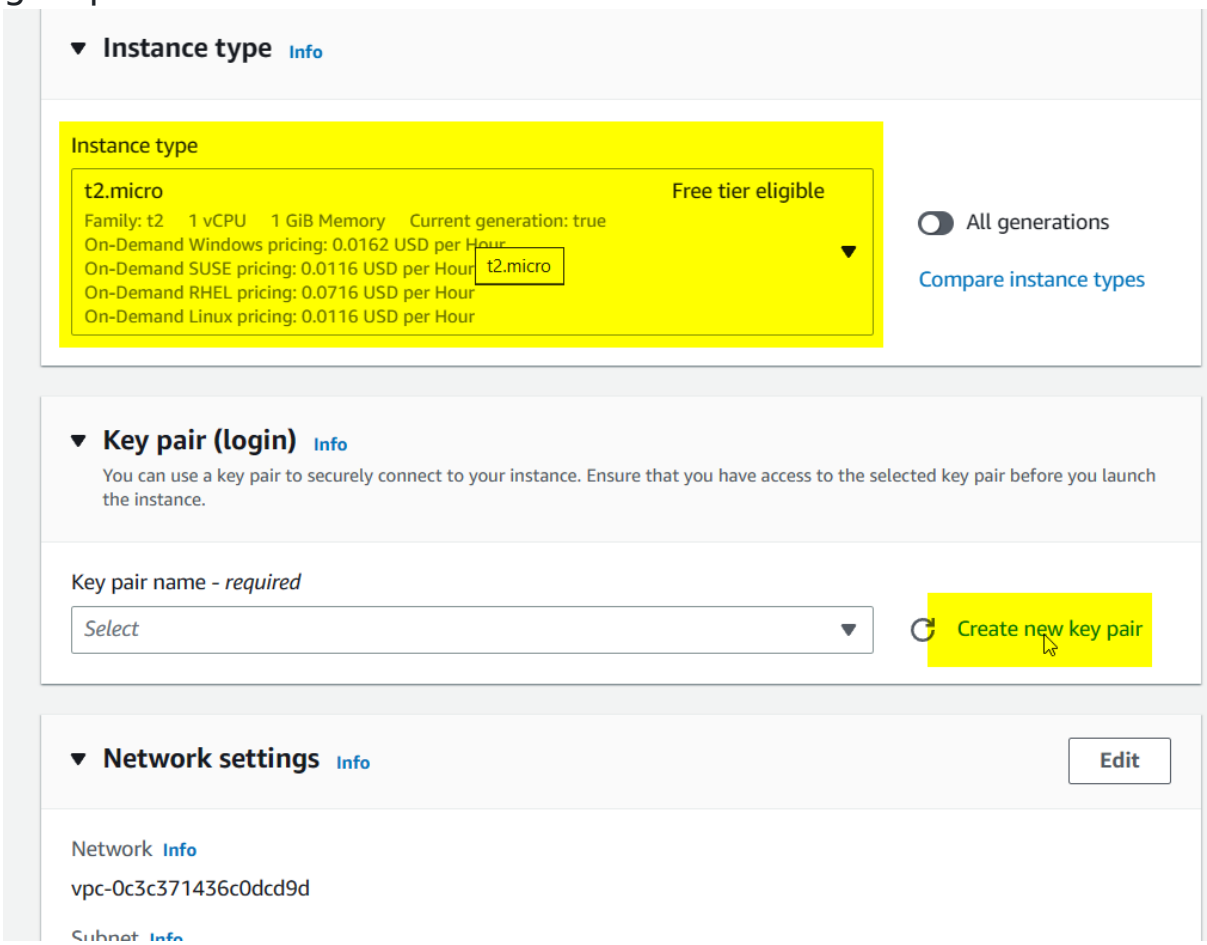


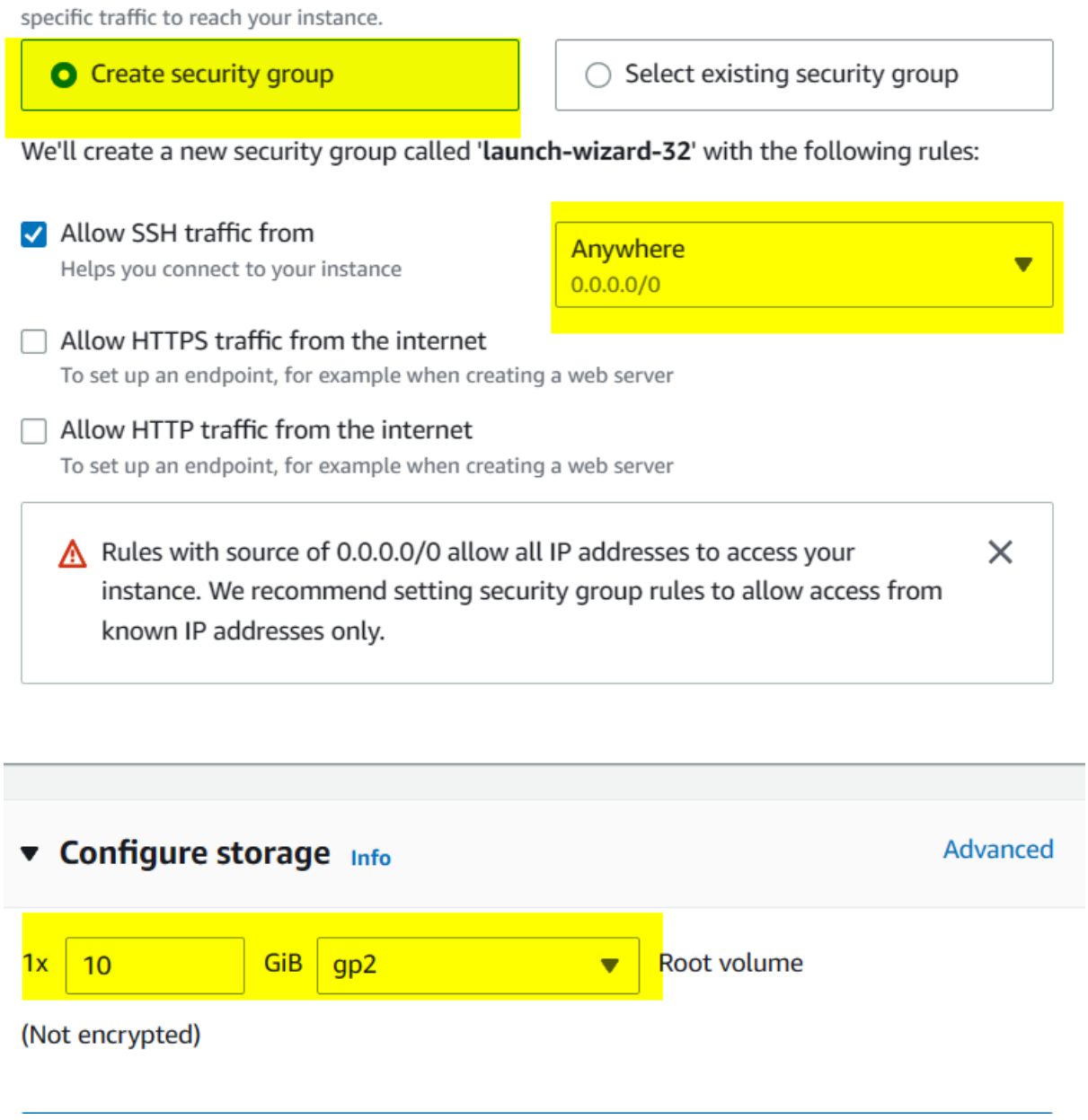
Select Red-Hat from the quick start option and note that amazon machine image selection varies from user to user .Select red hat enterprise Linux 9 HVM SSD Volume type .



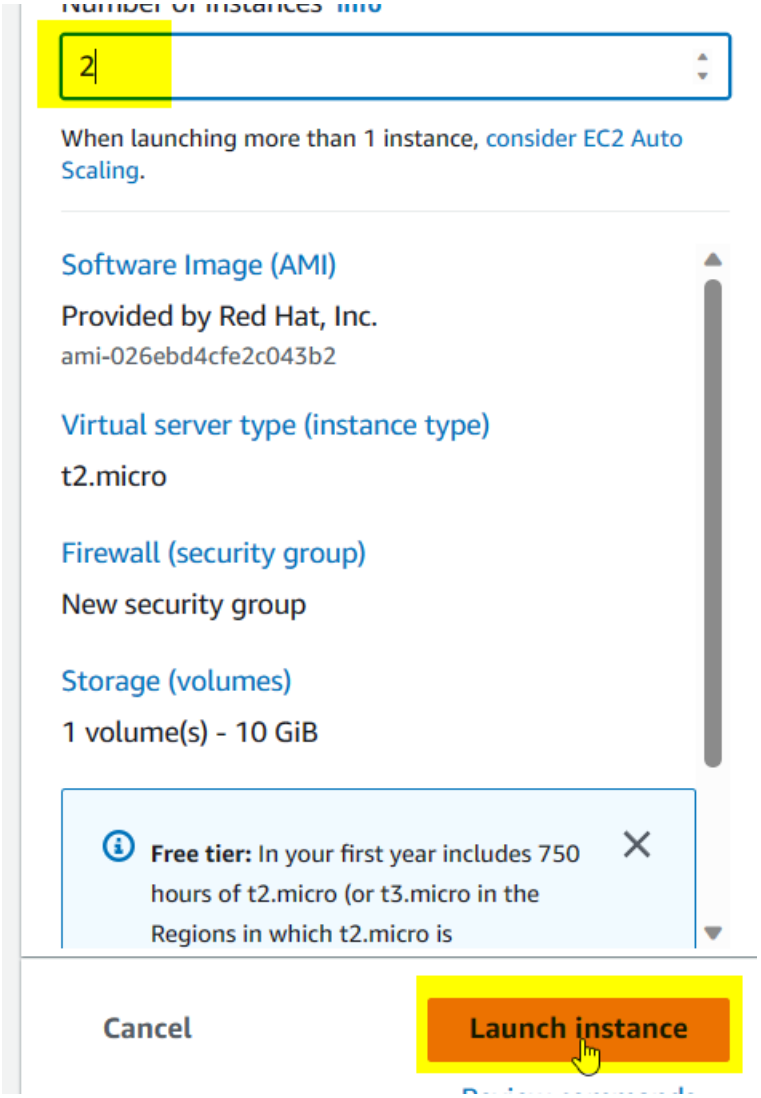
Click on the “Create new key pair” link and ensure the Checkbox remains unchanged on the “Create security

group”

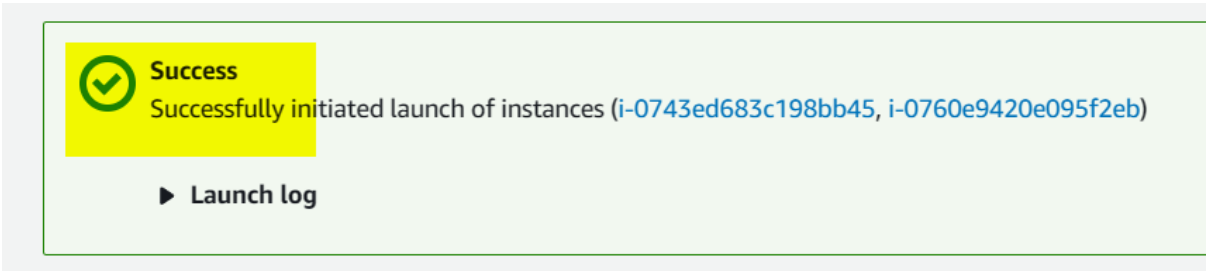

Instance type Info
Instance type
t2.micro
Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows pricing: 0.0162 USD per Hour
On-Demand SUSE pricing: 0.0116 USD per Hour
t2.micro
On-Demand RHEL pricing: 0.0716 USD per Hour
On-Demand Linux pricing: 0.0116 USD per Hour
All generations
Compare instance types
Key pair (login) Info
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.
Key pair name - required
Select
Create new key pair
Network settings Info
Edit
Network Info
vpc-0c3c371436c0dcd9d
Subnet Info



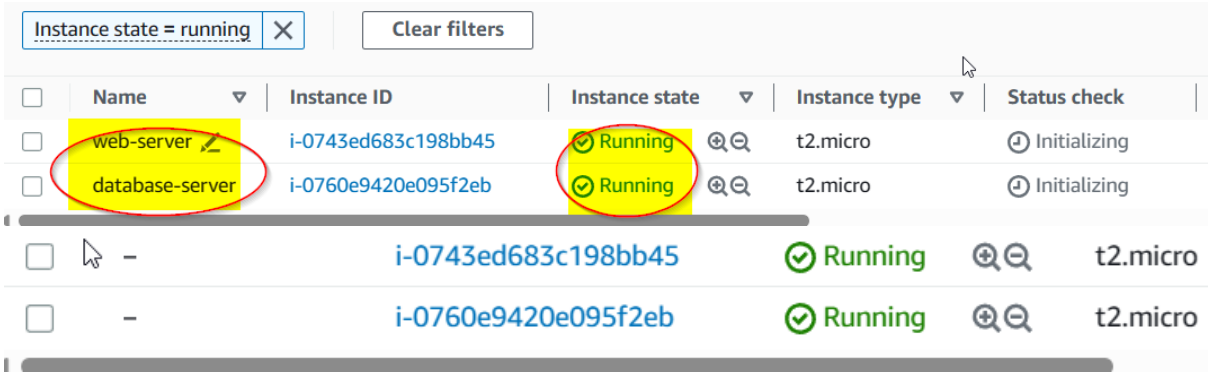
Select 2 instances and Launch both instances



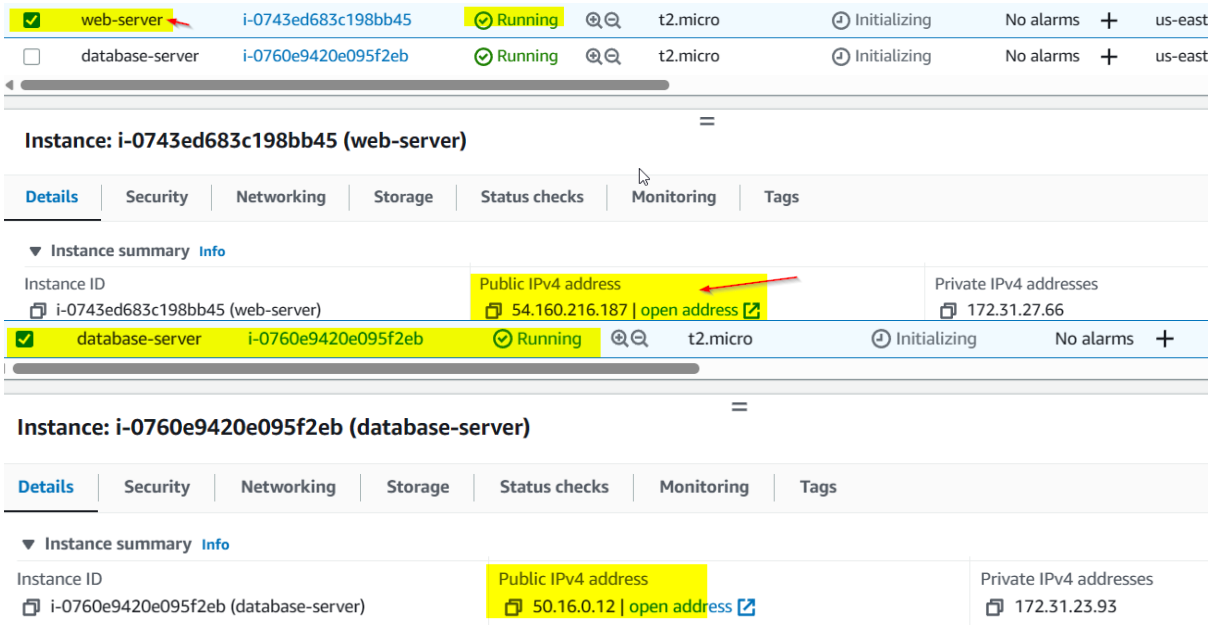
Instances are successfully launched as shown below



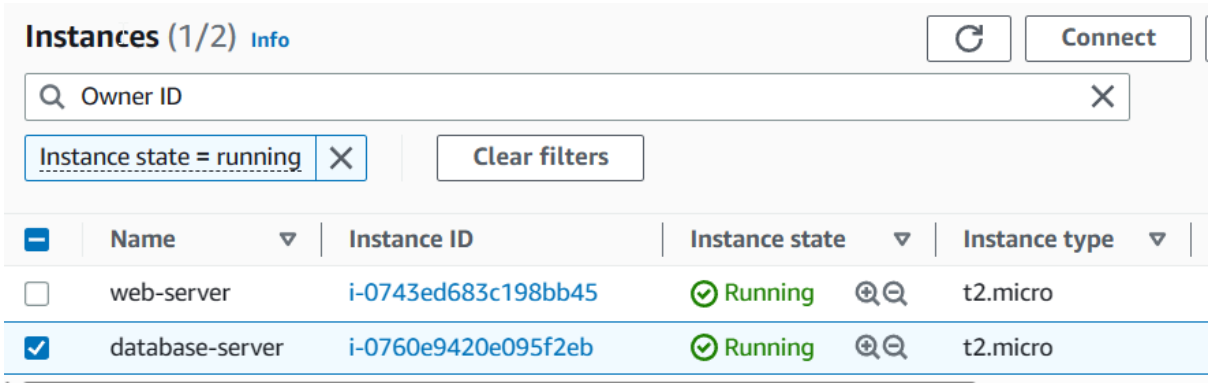
Then we name the 2 instance webserver and database respectively .



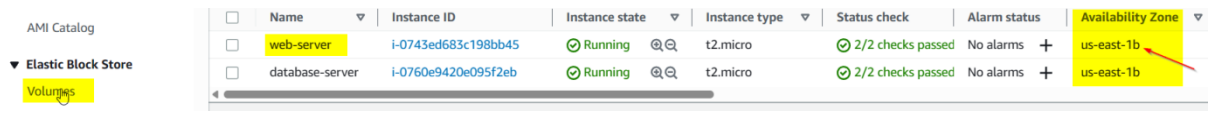
The public ip address of both servers are displayed below

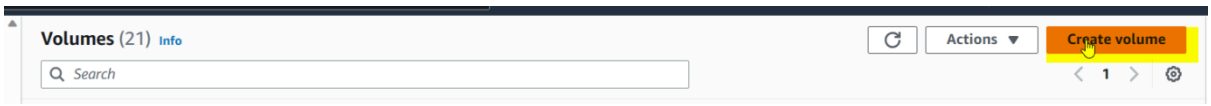


Click to connect to ssh

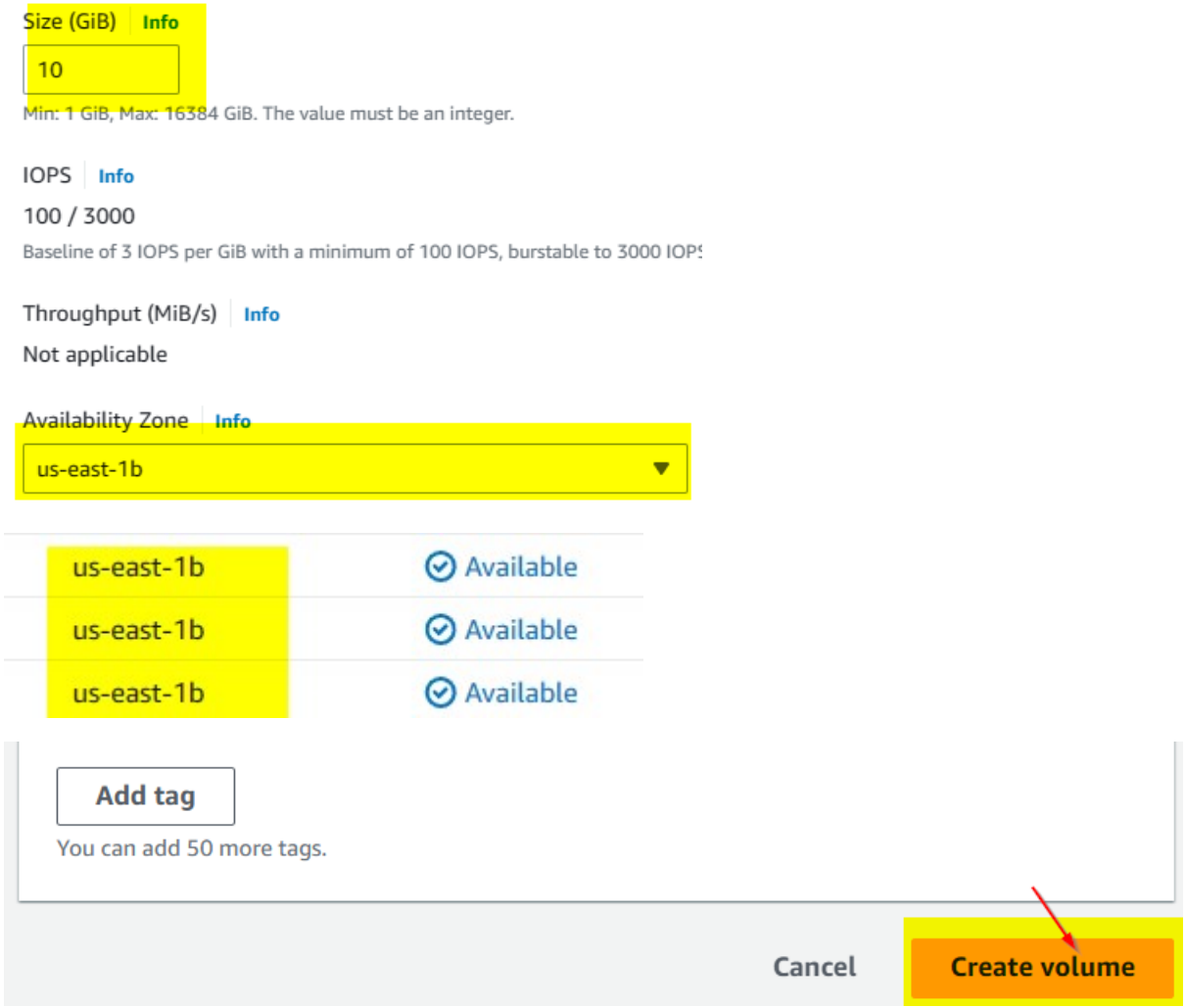


We then proceed to check the availability zone of our server and click add volumes.We are to create 3 volumes



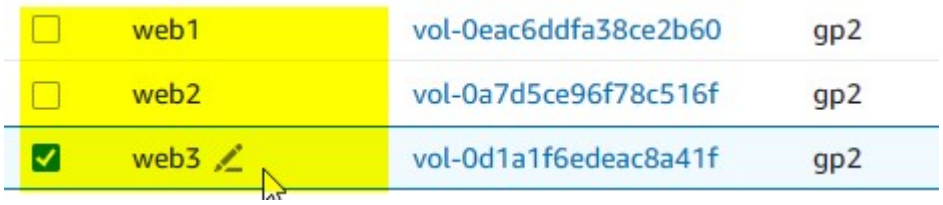


Select 10 Gib and the availability zone and click to create volume

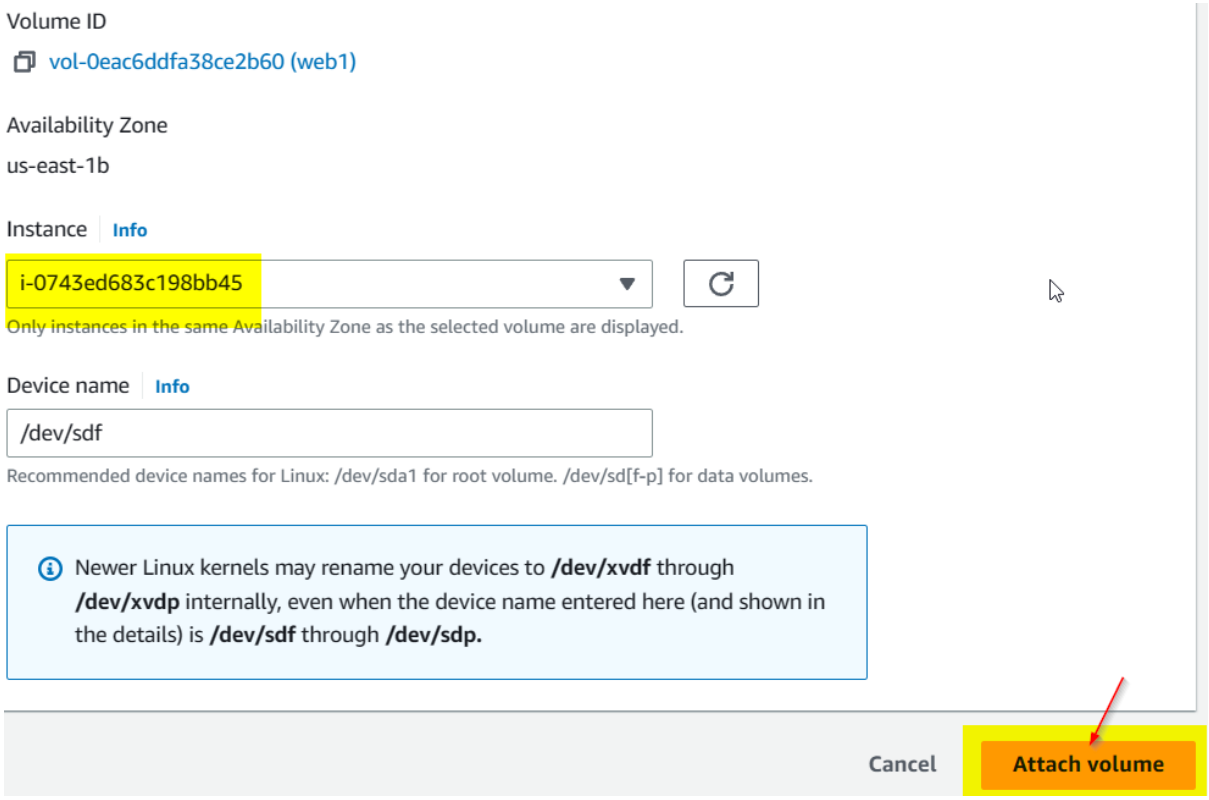


After creating the 3 volumes we refresh and can see them below.We name them web1,web2,web3 respectively

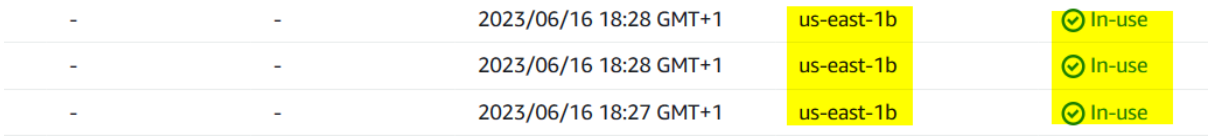
100	-	-	2023/06/16 18:08 GMT+1	us-east-1a	Available
100	-	-	2023/06/16 18:10 GMT+1	us-east-1a	Available
100	-	-	2023/06/16 18:10 GMT+1	us-east-1a	Available



Then we now attach each of the 3 volumes to the webserver as seen below



All 3 attachments are seen below and are now ready for use



WEBSERVER CONFIGURATION

Open git bash on visual studio code or whichever console is convenient to use. We are using git bash here with Visual Studio Code

We rename the ip address as webserver as seen below

```
oshor@Oshority MINGW64 ~/Downloads (master)
$ ssh -i "webanddbserver.pem" ec2-user@ec2-54-160-216-187.compute-1.amazonaws.
The authenticity of host 'ec2-54-160-216-187.compute-1.amazonaws.com (54.160.2
shed.
ED25519 key fingerprint is SHA256:9uTMOrwxpJ62mvxVhOjh2BqXwlnI7tyqkUOHTUTP6JU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

```
ec2-user@ip-172-31-27-66 ~]$ sudo hostname web-server
ec2-user@ip-172-31-27-66 ~]$ bash
ec2-user@web-server ~]$
```

Once all volumes have been attached you should run the lsblk command and you would be able to see all the 3 disks that have been created xvdf, xvdg and xvdh

```
[ec2-user@web-server ~]$ 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
```

With the df -h command we can see the mount point available

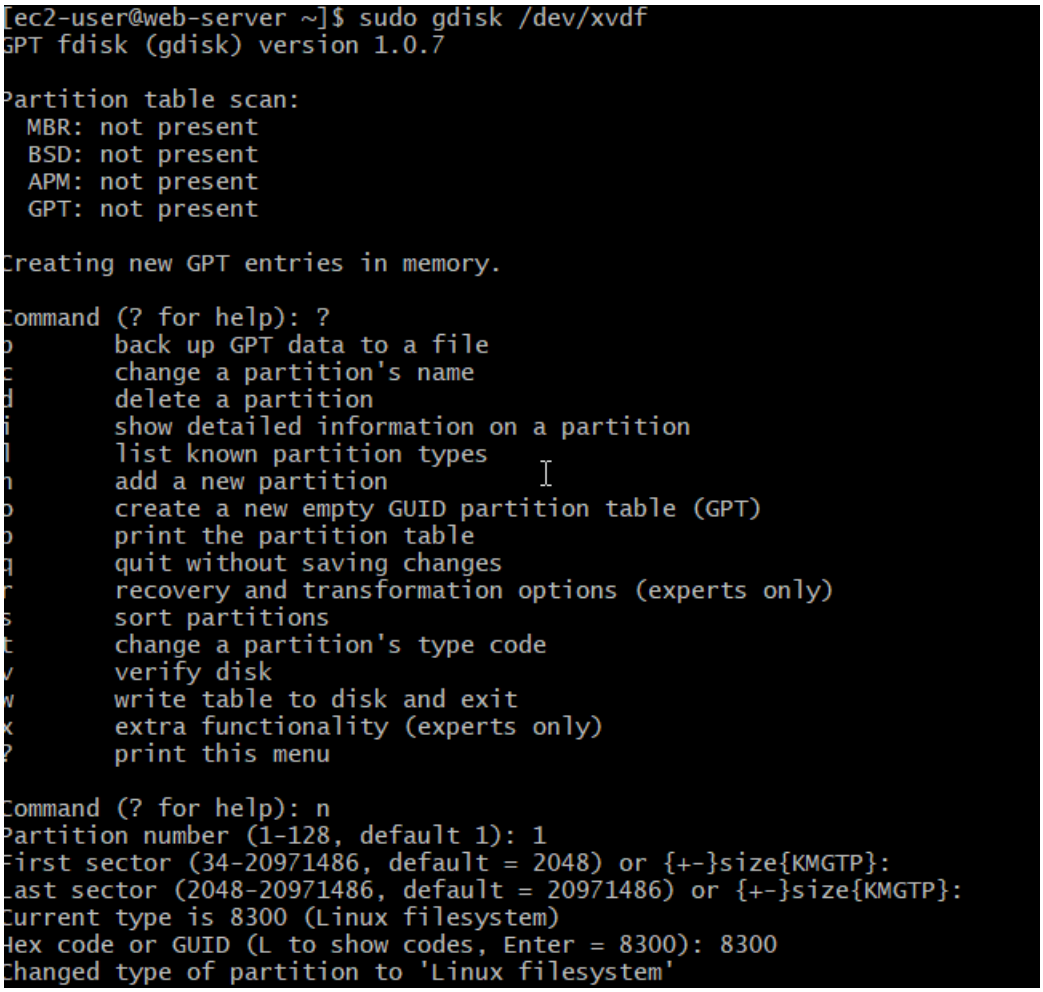
```
ec2-user@web-server ~]$ df -h
ilesystem      Size  Used Avail Use% Mounted on
levtmpfs       4.0M     0  4.0M   0% /dev
mpfs           385M     0  385M   0% /dev/shm
mpfs           154M  5.7M  149M   4% /run
/dev/xvda4     9.4G  1.3G  8.1G  14% /
/dev/xvda3     495M  153M  343M  31% /boot
/dev/xvda2     200M  8.0K  200M   1% /boot/efi
mpfs            77M     0   77M   0% /run/user/1000
ec2-user@web-server ~]$
```

We have to create a partition on the physical disk. We use the gdisk function to create a single partition on xvdf, xvdg and xvdh.Please note all the devices are stored in /dev .

```
[ec2-user@web-server ~]$ sudo ls /dev
autofs           hugepages     ppp       tty1   tty23  tty37  tty50  tty7       vcs6        xvda2
block            hwrng         ptmx      tty10  tty24  tty38  tty51  tty8       vcsa        xvda3
char             initctl       pts       tty11  tty25  tty39  tty52  tty9       vcsa1       xvda4
console          input         random    tty12  tty26  tty4   tty53  ttyS0      vcsa6       xvdf
core             kmsg          rfkill    tty13  tty27  tty40  tty54  ttyS1      vcsu        xvdg
cpu              log           rtc       tty14  tty28  tty41  tty55  ttyS2      vcsu1       xvdh
cpu_dma_latency  loop-control  rtc0      tty15  tty29  tty42  tty56  ttyS3      vcsu6       zero
disk             mapper        shm       tty16  tty3   tty43  tty57  udmabuf    vfio
dma_heap         mcelog        snapshot  tty17  tty30  tty44  tty58  uhid       vga_arbiter
dri              mem           snd       tty18  tty31  tty45  tty59  uinput     vhci
fb0              mqueue        stderr    tty19  tty32  tty46  tty6   urandom    vhost-net
fd               net           stdin     tty2   tty33  tty47  tty60  usbmon0    vhost-vsock
full             null          stdout    tty20  tty34  tty48  tty61  userfaultfd xen
fuse             nvram         tty       tty21  tty35  tty49  tty62  vcs        xvda
hpet             port          tty0      tty22  tty36  tty5   tty63  vcs1       xvda1
[ec2-user@web-server ~]$
```

We use the gdisk command as show below , Type “n” to add a new partition,

Choose 1 as the partition number and click enter button for the first and last sector .Enter :8300 for the default file system ,



Type “p” to view the partition table. Use “w” to write the table and edit on the disk and type “w” and click enter and type “y” to proceed.Then it can be seen that the operation was successful.

```
Command (? for help): p
Disk /dev/xvdf: 20971520 sectors, 10.0 GiB
Sector size (logical/physical): 512/512 bytes
Disk identifier (GUID): E6C147E7-9BFE-4806-986B-3CB71979A24E
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    8300  Linux filesystem

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@web-server ~]$
```

Repeat the same steps and create the partition for g and h partitions and the results are shown below

```
Do you want to proceed? (Y/N): y
OK; writing new GUID partition table (GPT) to /dev/xvdg.
The operation has completed successfully.
Do you want to proceed? (Y/N): y
OK; writing new GUID partition table (GPT) to /dev/xvdh.
The operation has completed successfully.
```

Type lsblk command to check again and you would see that the xvdf1,xvdg1,xvdh1 files has been created .

```
ec2-user@web-server ~]$ lsblk
AME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
vda    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 /
vdf    202:80   0   10G  0 disk
-xvdf1 202:81   0   10G  0 part
vdg    202:96   0   10G  0 disk
-xvdg1 202:97   0   10G  0 part
vdh    202:112  0   10G  0 disk
-xvdh1 202:113  0   10G  0 part
```

We then proceed to install the lvm2 package.

```
[ec2-user@web-server ~]$ sudo yum install lvm2 -y
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register

Last metadata expiration check: 0:00:35 ago on Fri 16 Jun 2023 06:36:51 PM UTC.
Dependencies resolved.
================================================================================
 Package                  Arch       Version              Repository          Size
================================================================================
Installing:
 lvm2                     x86_64     9:2.03.17-7.el9      rhel-9-baseos-rhui-rpms  1.5 M
Installing dependencies:
 device-mapper-event      x86_64     9:1.02.187-7.el9     rhel-9-baseos-rhui-rpms   36 k
 device-mapper-event-libs x86_64     9:1.02.187-7.el9     rhel-9-baseos-rhui-rpms   34 k
 device-mapper-persistent-data x86_64 0.9.0-13.el9       rhel-9-baseos-rhui-rpms  786 k
 libaio                   x86_64     0.3.111-13.el9       rhel-9-baseos-rhui-rpms   26 k
 lvm2-libs                x86_64     9:2.03.17-7.el9      rhel-9-baseos-rhui-rpms  1.0 M
```

Next step is to create a physical volume using the pvcreate command for the xvdf1, xvdg1 and xvdh1 respectively

```
[ec2-user@web-server ~]$ 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
```

We use the lsblk command to check the 3 physical volumes created

```
[ec2-user@web-server ~]$ 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
```

Use the pvs command to check the 3 physical volumes.

```
[ec2-user@web-server ~]$ 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
```

Volume groups is used to add together all physical volumes andmake them whole .We then use the vg-create command to let the 3 physical volume be seen as 1 logical volume and we name is webdata-vg as shown below

```
[ec2-user@web-server ~]$ sudo vgcreate webdata-vg /dev/xvdh1 /dev/xvdg1 /dev/xvdf1
  Volume group "webdata-vg" successfully created
```

Use “vgs” to check if it was implemented successfully.

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

The reason the VSize is not 30g is because some little amount has been reserved for the disk itself if space are needed in the future

From this volume group we can now create 2 logical volume which we give to our servers to use on apps and logs and confirm it was implemented successfully

```
[ec2-user@web-server ~]$ sudo lvcreate -n lv-apps -L 9G webdata-vg
  Logical volume "lv-apps" created.
[ec2-user@web-server ~]$ sudo lvcreate -n lv-logs -L 9G webdata-vg
  Logical volume "lv-logs" created.
[ec2-user@web-server ~]$
```

```
[ec2-user@web-server ~]$ sudo lvs
  LV      VG         Attr       LSize Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  lv-apps webdata-vg -wi-a----- 9.00g
  lv-logs webdata-vg -wi-a----- 9.00g
```

We can add 5g to the apps and logs by using this command for both of them as shown below and make them 14 gig each .Check and confirm it was implemented successfully

```
[ec2-user@web-server ~]$ sudo lvcreate -n lv-logs -L 5G webdata-vg^C
[ec2-user@web-server ~]$ sudo lvextend --size +5G webdata-vg/lv-logs
  Size of logical volume webdata-vg/lv-logs changed from 9.00 GiB (2304 extents) to 14.00 GiB (3584 ext
ents).
  Logical volume webdata-vg/lv-logs successfully resized.
[ec2-user@web-server ~]$ sudo lvextend --size +5G webdata-vg/lv-apps
  Size of logical volume webdata-vg/lv-apps changed from 9.00 GiB (2304 extents) to 14.00 GiB (3584 ext
ents).
  Logical volume webdata-vg/lv-apps successfully resized.
[ec2-user@web-server ~]$ sudo lvs
  LV      VG         Attr       LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  lv-apps webdata-vg -wi-a----- 14.00g
  lv-logs webdata-vg -wi-a----- 14.00g
[ec2-user@web-server ~]$
```

Use pvs to check whats left of the gig size (1.99)

```
[ec2-user@web-server ~]$ sudo pvs
  PV         VG         Fmt  Attr PSize   PFree
  /dev/xvdf1 webdata-vg lvm2 a--  <10.00g <1.99g
  /dev/xvdg1 webdata-vg lvm2 a--  <10.00g     0
  /dev/xvdh1 webdata-vg lvm2 a--  <10.00g     0
[ec2-user@web-server ~]$
```

And format the 2 logical volumes (command apps and logs) with the mkfs command

```
[ec2-user@web-server ~]$ sudo mkfs -t xfs  /dev/webdata-vg/lv-apps
meta-data=/dev/webdata-vg/lv-apps isize=512    agcount=4, agsize=917504 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=1 inobtcount=1
data     =                       bsize=4096   blocks=3670016, 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@web-server ~]$ sudo mkfs -t xfs  /dev/webdata-vg/lv-logs
meta-data=/dev/webdata-vg/lv-logs isize=512    agcount=4, agsize=917504 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=1 inobtcount=1
data     =                       bsize=4096   blocks=3670016, 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@web-server ~]$
```

Next, we are creating a mount point for our devices but to create a directory called www. we need confirm it is not existing already so we type the command below and we can confirm its not there and proceed to create the www directory

```
[ec2-user@web-server ~]$ ls -l /var
total 12
drwxr-xr-x.  2 root root    6 Aug  9  2021 adm
drwxr-xr-x. 11 root root  145 May  3 09:00 cache
drwxr-xr-x.  2 root root    6 Mar 10 07:13 crash
drwxr-xr-x.  3 root root   18 May  3 09:00 db
drwxr-xr-x.  2 root root    6 Aug  9  2021 empty
drwxr-xr-x.  2 root root    6 Aug  9  2021 ftp
drwxr-xr-x.  2 root root    6 Aug  9  2021 games
drwxr-xr-x.  3 root root   18 May  3 09:00 kerberos
drwxr-xr-x. 27 root root 4096 Jun 16 17:27 lib
drwxr-xr-x.  2 root root    6 Aug  9  2021 local
lrwxrwxrwx.  1 root root   11 May  3 09:00 lock -> ../run/lock
drwxr-xr-x.  9 root root 4096 Jun 16 17:12 log
lrwxrwxrwx.  1 root root   10 Aug  9  2021 mail -> spool/mail
drwxr-xr-x.  2 root root    6 Aug  9  2021 nis
drwxr-xr-x.  2 root root    6 Aug  9  2021 opt
drwxr-xr-x.  2 root root    6 Aug  9  2021 preserve
lrwxrwxrwx.  1 root root    6 May  3 09:00 run -> ../run
drwxr-xr-x.  7 root root   68 May  3 09:00 spool
drwxrwxrwt.  6 root root 4096 Jun 16 19:24 tmp
drwxr-xr-x.  2 root root    6 Aug  9  2021 yp
[ec2-user@web-server ~]$
```

We can see the folder and file created .

```
[ec2-user@web-server ~]$ sudo mkdir -p  /var/www/html
[ec2-user@web-server ~]$ ls -l /var
total 12
drwxr-xr-x.  2 root root    6 Aug  9  2021 adm
drwxr-xr-x. 11 root root  145 May  3 09:00 cache
drwxr-xr-x.  2 root root    6 Mar 10 07:13 crash
drwxr-xr-x.  3 root root   18 May  3 09:00 db
drwxr-xr-x.  2 root root    6 Aug  9  2021 empty
drwxr-xr-x.  2 root root    6 Aug  9  2021 ftp
drwxr-xr-x.  2 root root    6 Aug  9  2021 games
drwxr-xr-x.  3 root root   18 May  3 09:00 kerberos
drwxr-xr-x. 27 root root 4096 Jun 16 17:27 lib
drwxr-xr-x.  2 root root    6 Aug  9  2021 local
lrwxrwxrwx.  1 root root   11 May  3 09:00 lock -> ../run/lock
drwxr-xr-x.  9 root root 4096 Jun 16 17:12 log
lrwxrwxrwx.  1 root root   10 Aug  9  2021 mail -> spool/mail
drwxr-xr-x.  2 root root    6 Aug  9  2021 nis
drwxr-xr-x.  2 root root    6 Aug  9  2021 opt
drwxr-xr-x.  2 root root    6 Aug  9  2021 preserve
lrwxrwxrwx.  1 root root    6 May  3 09:00 run -> ../run
drwxr-xr-x.  7 root root   68 May  3 09:00 spool
drwxrwxrwt.  6 root root 4096 Jun 16 19:30 tmp
drwxr-xr-x.  3 root root   18 Jun 16 19:30 www
drwxr-xr-x.  2 root root    6 Aug  9  2021 yp
```

We create another directory and folder (/home/recovery/logs) and mount the apps-lv device and place it on the directory (/var/www/html)

Please note that mounting means the data exist in both places .But you should always check the existing file to know its empty before performing the mount operation because it might lead to loss of data if you don’t check the file content .

```
[ec2-user@web-server ~]$ sudo mkdir -p  /home/recovery/logs
[ec2-user@web-server ~]$ sudo mount /dev/webdata-vg/apps-lv  /var/www/html^C
[ec2-user@web-server ~]$ ls -l /var/www/html
total 0
[ec2-user@web-server ~]$
```

You can now proceed to mount and use the df -h command

```
[ec2-user@web-server ~]$ sudo mount /dev/webdata-vg/lv-apps  /var/www/html
[ec2-user@web-server ~]$ 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  5.7M  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-lv--apps  14G  133M   14G   1% /var/www/html
```

We have to check again for the logs file to investigate its content before mounting .As you can see there are 496 files which could have been lost if we proceed by mounting which makes it very mandatory to check as they are very important

to our machine .

```
[ec2-user@web-server ~]$ sudo ls -l /var/log
total 496
drwx------. 2 root   root       23 Jun 16 16:47 audit
-rw-rw----. 1 root   utmp    28416 Jun 16 18:17 btmp
-rw-r--r--. 1 root   root     1017 Jun 16 17:27 choose_repo.log
drwxr-x---. 2 chrony chrony      6 Oct 13  2022 chrony
-rw-r--r--. 1 root   root   162484 Jun 16 16:48 cloud-init.log
-rw-r-----. 1 root   adm      4059 Jun 16 16:48 cloud-init-output.log
-rw-------. 1 root   root     2489 Jun 16 19:01 cron
-rw-r--r--. 1 root   root     7829 Jun 16 18:37 dnf.librepo.log
-rw-r--r--. 1 root   root    17784 Jun 16 18:37 dnf.log
-rw-r--r--. 1 root   root     1768 Jun 16 18:37 dnf.rpm.log
-rw-r--r--. 1 root   root      600 Jun 16 18:37 hawkey.log
drwx------. 2 root   root        6 Nov 11  2022 insights-client
-rw-------. 1 root   root      722 Jun 16 16:48 kdump.log
-rw-rw-r--. 1 root   utmp   292292 Jun 16 16:58 lastlog
-rw-------. 1 root   root        0 May  3 09:00 maillog
-rw-------. 1 root   root   147657 Jun 16 19:40 messages
drwx------. 2 root   root        6 May  3 09:00 private
lrwxrwxrwx. 1 root   root       39 May  3 09:00 README -> ../../usr/share/doc/systemd/README.logs
drwxr-xr-x. 2 root   root       43 Jun 16 16:48 rhsm
-rw-------. 1 root   root    53746 Jun 16 19:40 secure
-rw-------. 1 root   root        0 May  3 09:00 spooler
drwxr-x---. 2 sssd   sssd        6 Jan 16 15:20 sssd
-rw-------. 1 root   root        0 May  3 09:00 tallylog
drwxr-xr-x. 2 root   root       23 Jun 16 16:48 tuned
-rw-rw-r--. 1 root   utmp     2688 Jun 16 16:58 wtmp
```

The solution to this is by copying the files in /var/log into the /home/recovery/log file and you can confirm it by simply checking its content as shown below

```
[ec2-user@web-server ~]$ sudo rsync -av /var/log  /home/recovery/logs
sending incremental file list
log/
log/README -> ../../usr/share/doc/systemd/README.logs
log/btmp
log/choose_repo.log
log/cloud-init-output.log
log/cloud-init.log
log/cron
log/dnf.librepo.log
log/dnf.log
log/dnf.rpm.log
log/hawkey.log
log/kdump.log
log/lastlog
log/maillog
log/messages
log/secure
log/spooler
log/tallylog
log/wtmp
log/audit/
log/audit/audit.log
log/chrony/
log/insights-client/
log/private/
log/rhsm/
log/rhsm/rhsm.log
log/rhsm/rhsmcertd.log
log/sssd/
log/tuned/
log/tuned/tuned.log

sent 1,310,936 bytes  received 458 bytes  2,622,788.00 bytes/sec
total size is 1,308,853  speedup is 1.00
[ec2-user@web-server ~]$ sudo ls -l /home/recovery/logs
total 4
drwxr-xr-x. 9 root root 4096 Jun 16 17:12 log
[ec2-user@web-server ~]$
```

And with this you can see that the folder has been backed up

```
[ec2-user@web-server ~]$ sudo ls -l /home/recovery/logs/log
total 736
drwx------. 2 root   root       23 Jun 16 16:47 audit
-rw-rw----. 1 root   utmp    28416 Jun 16 18:17 btmp
-rw-r--r--. 1 root   root     1017 Jun 16 17:27 choose_repo.log
drwxr-x---. 2 chrony chrony      6 Oct 13  2022 chrony
-rw-r--r--. 1 root   root   162484 Jun 16 16:48 cloud-init.log
-rw-r-----. 1 root   adm      4059 Jun 16 16:48 cloud-init-output.log
-rw-------. 1 root   root     2489 Jun 16 19:01 cron
-rw-r--r--. 1 root   root     7829 Jun 16 18:37 dnf.librepo.log
-rw-r--r--. 1 root   root    17784 Jun 16 18:37 dnf.log
-rw-r--r--. 1 root   root     1768 Jun 16 18:37 dnf.rpm.log
-rw-r--r--. 1 root   root      600 Jun 16 18:37 hawkey.log
drwx------. 2 root   root        6 Nov 11  2022 insights-client
-rw-------. 1 root   root      722 Jun 16 16:48 kdump.log
-rw-rw-r--. 1 root   utmp   292292 Jun 16 16:58 lastlog
-rw-------. 1 root   root        0 May  3 09:00 maillog
-rw-------. 1 root   root   147939 Jun 16 19:42 messages
drwx------. 2 root   root        6 May  3 09:00 private
lrwxrwxrwx. 1 root   root       39 May  3 09:00 README -> ../../usr/share/doc/systemd/README.logs
drwxr-xr-x. 2 root   root       43 Jun 16 16:48 rhsm
-rw-------. 1 root   root    54082 Jun 16 19:42 secure
-rw-------. 1 root   root        0 May  3 09:00 spooler
drwxr-x---. 2 sssd   sssd        6 Jan 16 15:20 sssd
-rw-------. 1 root   root        0 May  3 09:00 tallylog
drwxr-xr-x. 2 root   root       23 Jun 16 16:48 tuned
-rw-rw-r--. 1 root   utmp     2688 Jun 16 16:58 wtmp
```

Proceed to mounting

```
[ec2-user@web-server ~]$  sudo mount /dev/webdata-vg/lv-logs /var/log
[ec2-user@web-server ~]$ sudo ls -l /var/log
total 0
[ec2-user@web-server ~]$ sudo rsync -av /home/recovery/logs/  /var/log
sending incremental file list
./
log/
log/README -> ../../usr/share/doc/systemd/README.logs
log/btmp
log/choose_repo.log
log/cloud-init-output.log
log/cloud-init.log
log/cron
log/dnf.librepo.log
log/dnf.log
log/dnf.rpm.log
log/hawkey.log
log/kdump.log
log/lastlog
log/maillog
```

The command below shows that it has been properly mounted

```
[ec2-user@web-server ~]$ 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  5.7M  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-lv--apps  14G  133M   14G   1% /var/www/html
/dev/mapper/webdata--vg-lv--logs  14G  134M   14G   1% /var/log
[ec2-user@web-server ~]$
```

We should know that the mount is temporary and once the server is rebooted it would loose connection. We should ensure to make the connection persist and that would be by editing the fstab file and adding some crucial command to make it permanently stable

We check the blkid to check the commands we need to populate the fstab

```
[ec2-user@web-server ~]$ sudo blkid
/dev/xvda4: LABEL="root" UUID="287d9c0b-0e0f-4e92-8534-45733aa3dc68" TYPE="xfs" PARTUUID="6264d520-3fb9
-423f-8ab8-7a0a8e3d3562"
/dev/mapper/webdata--vg-lv--logs: UUID="0429602f-c0b9-4d3c-bc2e-f9e2a98c4f1e" TYPE="xfs"
/dev/xvda2: SEC_TYPE="msdos" UUID="7B77-95E7" TYPE="vfat" PARTUUID="68b2905b-df3e-4fb3-80fa-49d1e773aa3
a"
/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="9jcF6W-h3Le-MfwY-Teum-JyOv-rte7-L5TICu" TYPE="LVM2_member" PARTLABEL="Linux filesyste
m" PARTUUID="a77253fc-36b7-4fe7-a3e3-7ce4eb8118da"
/dev/xvdf1: UUID="jTeR51-329y-t0mj-P6Uy-z33T-2SuY-CttJ9K" TYPE="LVM2_member" PARTLABEL="Linux filesyste
m" PARTUUID="1e9886c3-7c9a-4d91-9e89-d2ee872ea5ea"
/dev/mapper/webdata--vg-lv--apps: UUID="1c420fbe-1783-4315-b693-656f7bb511d4" TYPE="xfs"
/dev/xvdg1: UUID="l3M75S-5iMD-kcYR-zmmC-44wc-yjOV-dtHxRc" TYPE="LVM2_member" PARTLABEL="Linux filesyste
m" PARTUUID="74a288ef-d355-4170-83c4-4be791c9c899"
```

We are going to copy the UUID for the lv-apps and lv -logs and go back to our terminal.

```
[ec2-user@ip-172-31-27-66 ~]$ sudo vi  /etc/fstab

#MOUNT FOR WORDPRESS SERVER
/dev/mapper/webdata--vg-lv--logs: UUID=0429602f-c0b9-4d3c-bc2e-f9e2a98c4f1e   /var/www/html  xfs  default,nofail 0  0

/dev/mapper/webdata--vg-lv--apps: UUID=1c420fbe-1783-4315-b693-656f7bb511d4  /var/logs  xfs  default,nofail 0  0
```

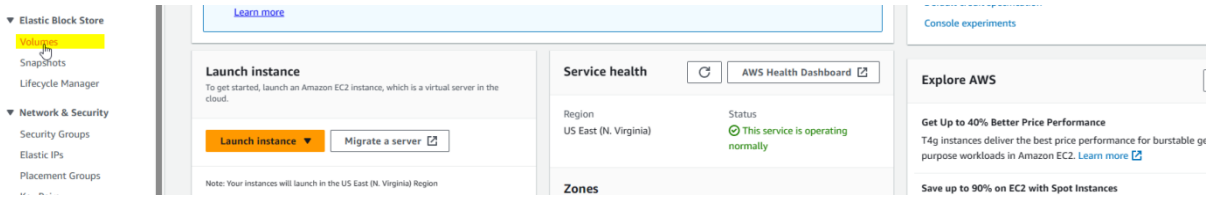
We have update the fstab and also reloaded the daemon

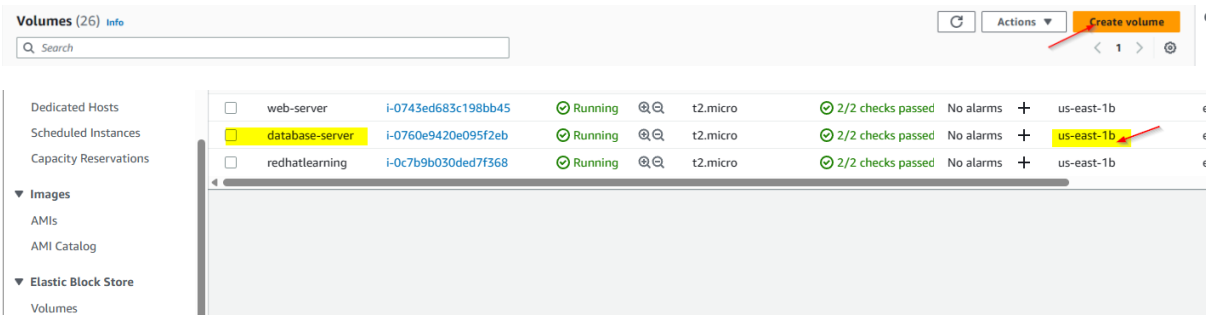
```
[ec2-user@ip-172-31-27-66 ~]$ df -h
Filesystem                      Size  Used Avail Use% Mounted on
devtmpfs                        4.0M     0  4.0M   0% /dev
tmpfs                           385M     0  385M   0% /dev/shm
tmpfs                           154M  9.0M  145M   6% /run
/dev/xvda4                      9.4G  1.3G  8.1G  14% /
/dev/xvda3                      495M  153M  343M  31% /boot
/dev/xvda2                      200M  8.0K  200M   1% /boot/efi
/dev/mapper/webdata--vg-lv--apps   14G  133M   14G   1% /var/www/html
/dev/mapper/webdata--vg-lv--logs   14G  134M   14G   1% /var/log
tmpfs                           77M     0   77M   0% /run/user/1000
[ec2-user@ip-172-31-27-66 ~]$ sudo systemctl daemon-reload
[ec2-user@ip-172-31-27-66 ~]$ sudo mount -a
[ec2-user@ip-172-31-27-66 ~]$
```

DATABASE SERVER PREPARATION

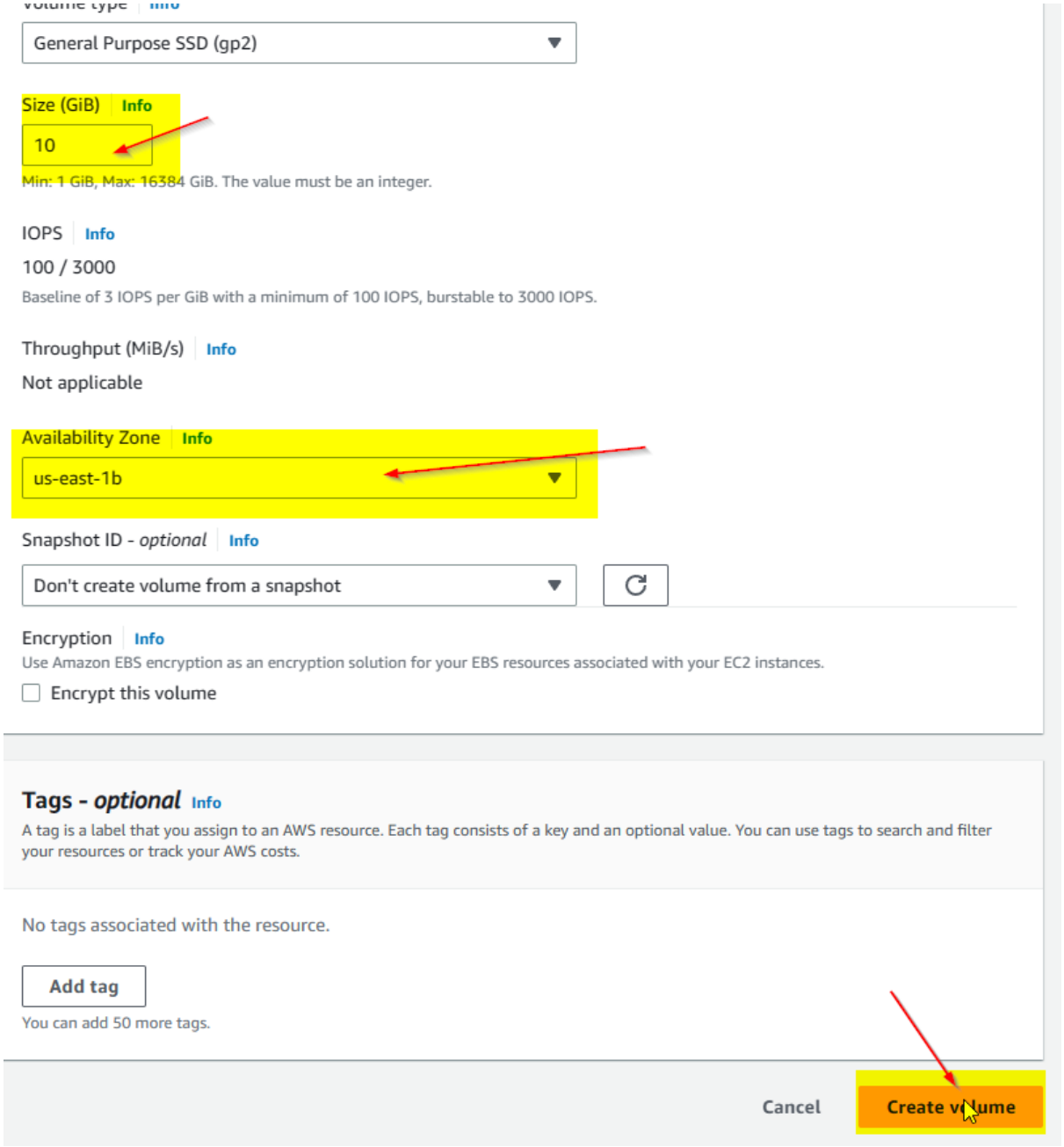
In this preparation we would launch a red hat instance for the DB server and repeat all steps and create a db-lv and mount It to a /db

We would create volume and also take note of the availability zone of the database server and create 3 volumes

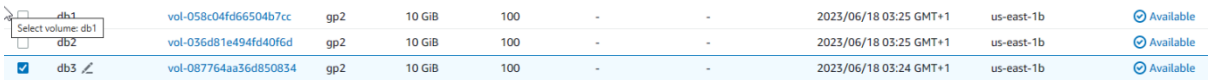




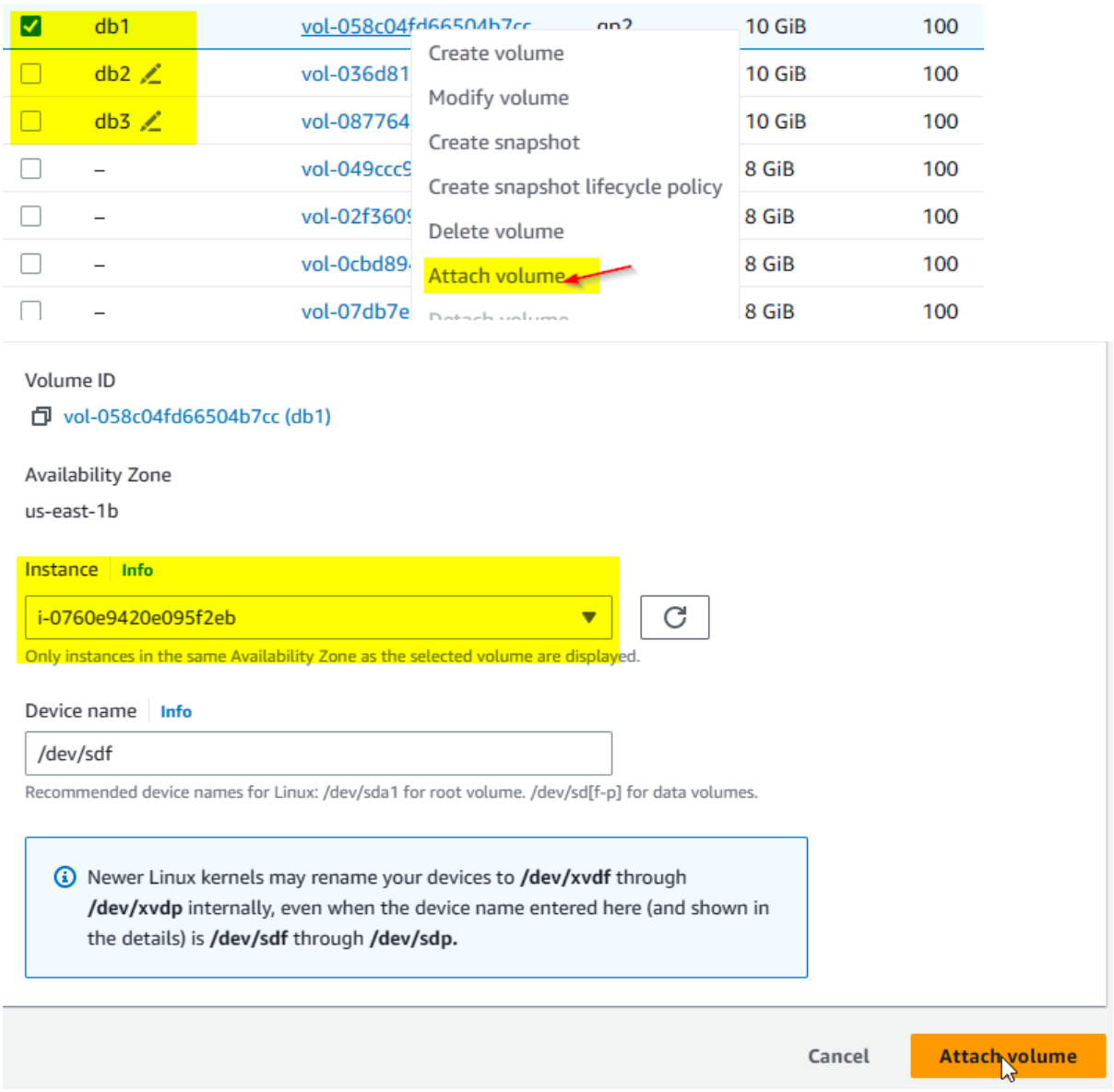
Select 10 Gib and the availability zone and click to create volume



Refresh to see all volumes created and name them db1,db2,db3 respectively and see they are available



Then we now attach each of the 3 volumes to the webserver as seen below



Repeat this steps for the rest of the volumes as seen below and all 3 volumes are now ready for use by the database server

DATABASE SERVER CONFIGURATION

Open git bash on visual studio code or whichever console is convenient to use. We are using git bash here with Visual Studio Code

We rename the ip address as webserver as seen below.

```
oshor@O[illegible]hority MINGW64 ~/Downloads (master)
$ ssh -i "webanddbserver.pem" ec2-user@ec2-50-16-0-12.compute-1.amazonaws.com
The authenticity of host 'ec2-50-16-0-12.compute-1.amazonaws.com (50.16.0.12)'
ED25519 key fingerprint is SHA256:ggv9knR6Gny6JOlVFWgVR+gLbXtDsXDELh4Ck83miqE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

```
[ec2-user@ip-172-31-23-93 ~]$ sudo hostname database-server
[ec2-user@ip-172-31-23-93 ~]$ bash
[ec2-user@database-server ~]$
```

Once all volumes have been attached you should run the lsblk command and you would be able to see all the 3 disks that have been created xvdf, xvdg and xvdh

```
[ec2-user@database-server ~]$ 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@database-server ~]$
```

With the df -h command we can see the mount point available

```
[ec2-user@database-server ~]$ 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  6.6M  148M   5% /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
```

We use the gdisk command as show below , Type “n” to add a new partition,

Choose 1 as the partition number and click enter button for the first and last sector .Enter :8300 for the default file system ,

```
[ec2-user@database-server ~]$ 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): ?
b       back up GPT data to a file
c       change a partition's name
d       delete a partition
i       show detailed information on a partition
l       list known partition types
n       add a new partition
o       create a new empty GUID partition table (GPT)
p       print the partition table
q       quit without saving changes
r       recovery and transformation options (experts only)
s       sort partitions
t       change a partition's type code
v       verify disk
w       write table to disk and exit
x       extra functionality (experts only)
?       print this menu

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): 8300
Changed type of partition to 'Linux filesystem'
```

Type “p” to view the partition table. Use “w” to write the table and edit on the disk and type “w” and click enter and type “y” to proceed. Then it can be seen that the operation was successful.

```
Command (? for help): p
Disk /dev/xvdf: 20971520 sectors, 10.0 GiB
Sector size (logical/physical): 512/512 bytes
Disk identifier (GUID): 1F6F2769-5A5B-4A27-9467-2E9427C75B95
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    8300  Linux filesystem

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@database-server ~]$
```

Type lsblk command to check again and you would see that the xvdf1,xvdg1,xvdh1 files has been created

```
[ec2-user@database-server ~]$ 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@database-server ~]$
```

Repeat the same steps and create the partition for g and h partitions and the results are shown below

```
Do you want to proceed? (Y/N): y
OK; writing new GUID partition table (GPT) to /dev/xvdh.
The operation has completed successfully.
```

```
Do you want to proceed? (Y/N): y
OK; writing new GUID partition table (GPT) to /dev/xvdg.
The operation has completed successfully.
[ec2-user@database-server ~]$
```

We then proceed to install the lvm2 package.

```
[ec2-user@database-server ~]$ sudo yum install lvm2 -y
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Last metadata expiration check: 0:47:35 ago on Sun 18 Jun 2023 02:10:54 AM UTC.
Dependencies resolved.
================================================================================
 Package                     Arch       Version              Repository      Size
================================================================================
Installing:
 lvm2                        x86_64     9:2.03.17-7.el9      rhel-9-baseos-rhui-rpms  1.5 M
Installing dependencies:
 device-mapper-event         x86_64     9:1.02.187-7.el9     rhel-9-baseos-rhui-rpms   36 k
 device-mapper-event-libs    x86_64     9:1.02.187-7.el9     rhel-9-baseos-rhui-rpms   34 k
 device-mapper-persistent-data x86_64   0.9.0-13.el9         rhel-9-baseos-rhui-rpms  786 k
 libaio                      x86_64     0.3.111-13.el9       rhel-9-baseos-rhui-rpms   26 k
 lvm2-libs                   x86_64     9:2.03.17-7.el9      rhel-9-baseos-rhui-rpms  1.0 M
```

Next step is to create a physical volume using the pvcreate command for the xvdf1, xvdg1 and xvdh1 respectively

```
[ec2-user@database-server ~]$ 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@database-server ~]$
```

We use the lsblk command to check the 3 physical volumes created

```
[ec2-user@database-server ~]$ 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@database-server ~]$
```

Use the pvs command to check the 3 physical volumes.

```
[ec2-user@database-server ~]$ 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
```

Volume groups is used to add together all physical volumes and make them whole .We then use the vg-create command to let the 3 physical volume be seen as 1 logical volume and we name is database-vg as shown below

```
[ec2-user@database-server ~]$ sudo vgcreate database-vg  /dev/xvdf1  /dev/xvdg1  /dev/xvdh1
  Volume group "database-vg" successfully created
[ec2-user@database-server ~]$
```

Use “vgs” to check if it was implemented successfully.

```
[ec2-user@database-server ~]$ sudo vgs
  VG          #PV #LV #SN Attr   VSize   VFree
  database-vg   3   0   0 wz--n- <29.99g <29.99g
[ec2-user@database-server ~]$
```

The reason the VSize is not 30g is because some little amount has been reserved for the disk itself if space is needed in the future

From this volume group we can now create 2 logical volume which we give to our servers to use on apps and logs and confirm it was implemented successfully

```
[ec2-user@database-server ~]$ sudo lvcreate -n lv-db -L 20G database-vg
  Logical volume "lv-db" created.
[ec2-user@database-server ~]$
```

```
[ec2-user@database-server ~]$ sudo lvs
  LV    VG          Attr       LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  lv-db database-vg -wi-a----- 20.00g
```

Use pvs to check what’s left of the gig size (9.99)

```
[ec2-user@database-server ~]$ sudo pvs
  PV         VG          Fmt  Attr PSize   PFree
  /dev/xvdf1 database-vg lvm2 a--  <10.00g     0
  /dev/xvdg1 database-vg lvm2 a--  <10.00g     0
  /dev/xvdh1 database-vg lvm2 a--  <10.00g <9.99g
[ec2-user@database-server ~]$
```

Next step is to create the mount point .We create a directory called “db”

and create the file system using the mkfs command and then mount

```
[ec2-user@database-server ~]$ sudo pvs
  PV         VG          Fmt  Attr PSize   PFree
  /dev/xvdf1 database-vg lvm2 a--  <10.00g     0
  /dev/xvdg1 database-vg lvm2 a--  <10.00g     0
  /dev/xvdh1 database-vg lvm2 a--  <10.00g <9.99g
[ec2-user@database-server ~]$ sudo mkdir /db
[ec2-user@database-server ~]$ sudo mkfs -t xfs /dev/database-vg/lv-db
meta-data=/dev/database-vg/lv-db isize=512    agcount=4, agsize=1310720 bl
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=1 inobtcount=1
data     =                       bsize=4096   blocks=5242880, 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
```

We know we created the file but lets check to see if it has any content .As you can see below it has no content

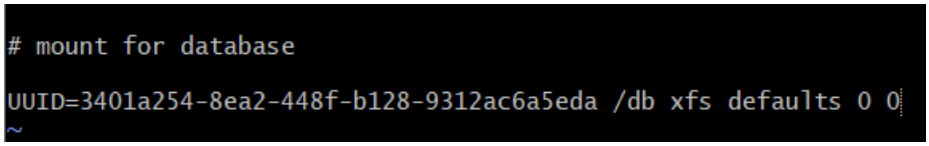
```
[ec2-user@database-server ~]$ ls -l /db
total 0
```

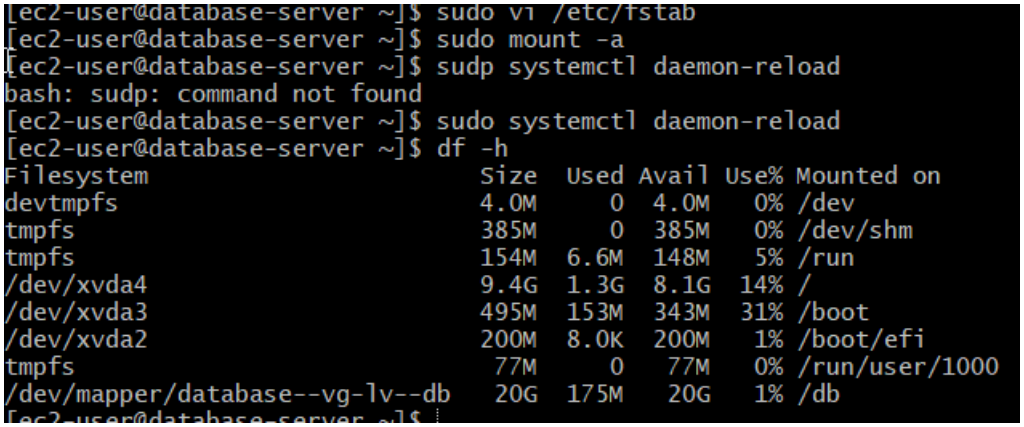
Then we mount it and also confirm If it has been mounted with the df -h command

```
[ec2-user@database-server ~]$ sudo mount /dev/database-vg/lv-db  /db
[ec2-user@database-server ~]$ 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  6.6M  148M   5% /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-lv--db   20G  175M   20G   1% /db
[ec2-user@database-server ~]$
```

(Using blkid command)

The next thing to do is to make sure that the connection is persisted and consistent even after reboot. We would be editing the /etc/fstab file to perform this action and confirm if success with the command below and have a system reload .Also check the df -h command to confirm its there in the information provided



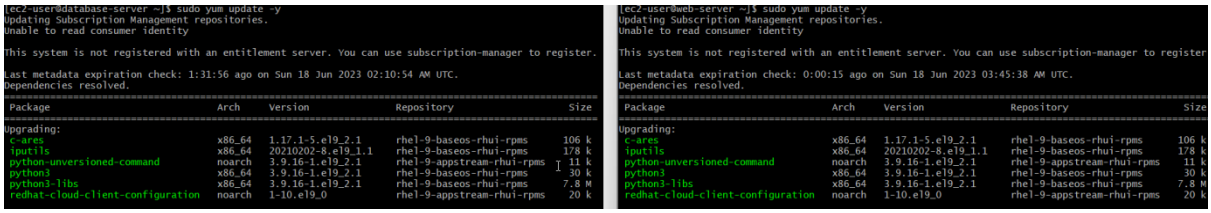


Next step is to install word press on our webserver.

WORDPRESS INSTALLATION ON WEBSERVER

This installation should be done in the webserver .We have to run an update on the webserver and the database server .

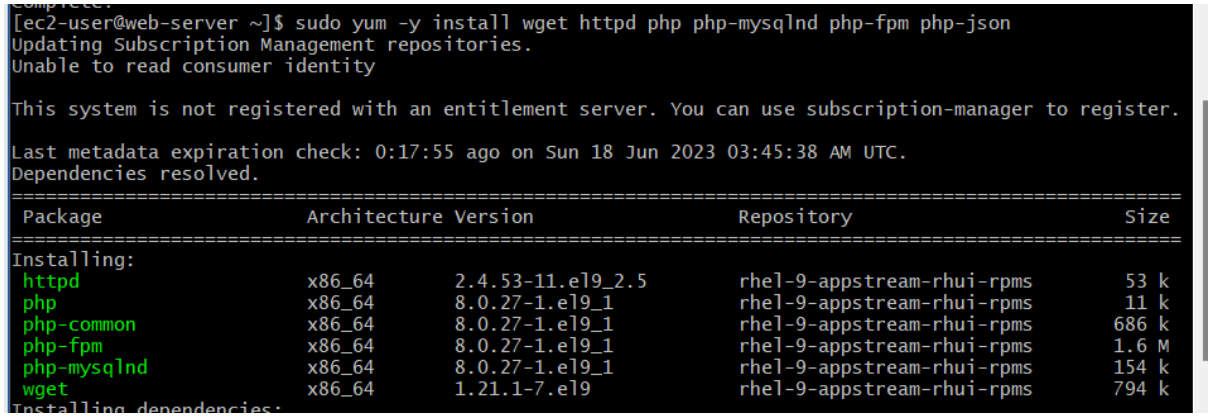
But before then we have to update both servers as shown below



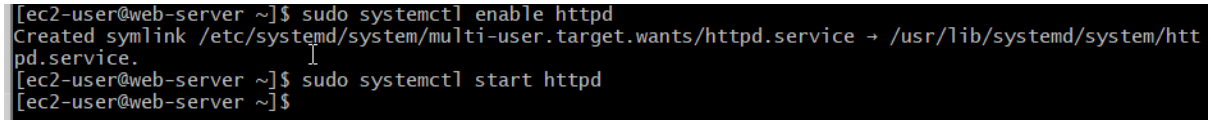
We should check the security group of both instances to ensure they are all open to the traffic we want it to be.

We then proceed to install WordPress on our webserver.

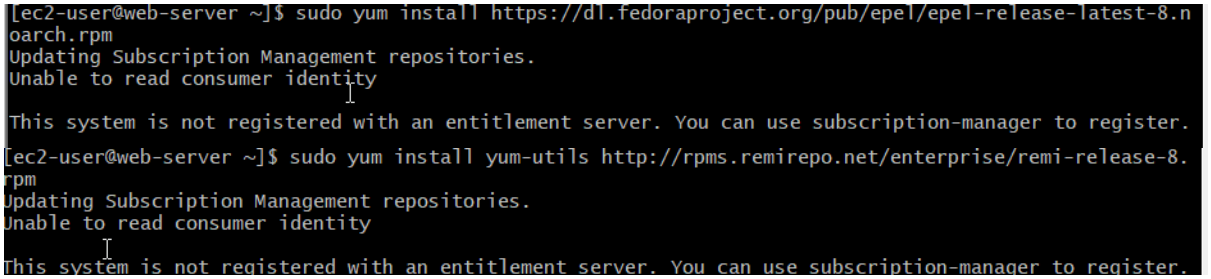
First ,Install wget,apache and its dependencies



Then we enable and start apache httpd



Then we proceed to install the PHP and its dependencies



We then proceed to list and reset and enable the PHP

```
[ec2-user@web-server ~]$ sudo yum module list php
Updating Subscription Management repositories.
Unable to read consumer identity
```

```
[ec2-user@web-server ~]$ sudo yum module reset php
Updating Subscription Management repositories.
Unable to read consumer identity
```

```
[ec2-user@web-server ~]$ sudo yum module enable php
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Last metadata expiration check: 0:01:41 ago on Sun 18 Jun 2023 04:07:27 AM UTC.
Dependencies resolved.
================================================================================
 Package                Architecture          Version                Repository              Size
================================================================================
Enabling module streams:
 php                                          8.1

Transaction Summary
================================================================================

Is this ok [y/N]: y
```

Install other dependencies of php

```
[ec2-user@web-server ~]$ sudo yum install php php-opcache php-gd php-curl php-mysqlnd
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.
```

Start, enable and set Boolean value for Apache as shown below

```
[ec2-user@web-server ~]$ sudo systemctl start php-fpm
[ec2-user@web-server ~]$ sudo systemctl enable php-fpm
Created symlink /etc/systemd/system/multi-user.target.wants/php-fpm.service → /usr/lib/systemd/system/p
hp-fpm.service.
[ec2-user@web-server ~]$ setsebool -P httpd_execmem 1
Cannot set persistent booleans, please try as root.
[ec2-user@web-server ~]$ sudo setsebool -P httpd_execmem 1
[ec2-user@web-server ~]$
```

Restarting Apache

```
ec2-user@web-server ~]$ sudo systemctl restart httpd
ec2-user@web-server ~]$
```

Apache status and confirm there is no file on the /var/www/html location

```
[ec2-user@web-server ~]$ 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 Sun 2023-06-18 04:16:18 UTC; 2min 56s ago
       Docs: man:httpd.service(8)
   Main PID: 22645 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec
      Tasks: 213 (limit: 4421)
```

```
[ec2-user@web-server ~]$ sudo ls -l  /var/www/html
total 0
[ec2-user@web-server ~]$
```

We proceed to create and change directory to WordPress and install it with its dependencies.

```
[ec2-user@web-server ~]$ mkdir wordpress
[ec2-user@web-server ~]$ cd wordpress/
[ec2-user@web-server wordpress]$   sudo wget http://wordpress.org/latest.tar.gz
--2023-06-18 04:27:44--  http://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://wordpress.org/latest.tar.gz [following]
--2023-06-18 04:27:44--  https://wordpress.org/latest.tar.gz
```

Downloading in progress and finally completed .Configure selinux policies as well

```
[ec2-user@web-server 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
```

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

Go back to your database server to complete the installation and configuration.

Installing MySQL on database server, restart, enable and check status

```
Installed size: 236 M
Is this ok [y/N]: y
Downloading Packages:
(1/67): perl-Digest-MD5-2.58-4.el9.x86_64.rpm
[ec2-user@database-server ~]$ sudo yum install mysql-server
Updating Subscription Management repositories.
Unable to read consumer identity

[ec2-user@database-server ~]$ sudo systemctl restart mysqld
[ec2-user@database-server ~]$ sudo systemctl enable mysqld
Created symlink /etc/systemd/system/multi-user.target.wants/mysqld.service → /usr/lib/systemd/system/my
sqld.service.
[ec2-user@database-server ~]$ 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 Sun 2023-06-18 04:38:56 UTC; 18s ago
   Main PID: 19782 (mysqld)
```

Configure DB to work with WordPress

We would create a database named wordpress and create my user with your credentials and ensure you grant all permission and flush out privileges .Once done check the database tables and exit it

```
[ec2-user@database-server ~]$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
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> CREATE DATABASE wordpress;
Query OK, 1 row affected (0.01 sec)

mysql> CREATE USER `myuser`@`54.160.216.187` IDENTIFIED BY 'mypass';
Query OK, 0 rows affected (0.02 sec)

mysql> GRANT ALL ON wordpress.* TO 'myuser'@'54.160.216.187';
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES
    -> SHOW DATABASES
    -> exit
    -> FLUSH PRIVILEGES;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MyS
QL server version for the right syntax to use near 'SHOW DATABASES
exit
FLUSH PRIVILEGES' at line 2
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| wordpress          |
+--------------------+
5 rows in set (0.00 sec)
```

Please note that when we check the user you would get this

```
+------------------+----------------+
| user             | host           |
+------------------+----------------+
| myuser           | 54.160.216.187 |
| mysql.infoschema | localhost      |
| mysql.session    | localhost      |
| mysql.sys        | localhost      |
| root             | localhost      |
+------------------+----------------+
5 rows in set (0.00 sec)
```

```
mysql> exit
Bye
[ec2-user@database-server ~]$
```

We need to set the bind address :

```
#
[client-server]

!includedlr /etc/my.conf.d
[mysqld]
bind address=0.0.0.0
~
~
~
~
~
```

The next step would be to configure WORDPRESS which is the webserver to act as a client to connect to the remote database

```
[ec2-user@database-server ~]$ sudo vi /etc/my.cnf.d
[ec2-user@database-server ~]$ sudo systemctl restart mysqld
[ec2-user@database-server ~]$
```