Adding additional storage in EC2 [AWS]

Q.I have an instance with 8G storage, I want to create a directory for user 1 and allocate him 6G and another user 6G. So, the users shouldn't have to worry about the space, minimum they should have the requested size in their working directory. How can u complete this task

A:

1. we want an instance with 8 G of basic storage

2. we will attach total of 12G disk

3. Then we need to split the 12G into 2

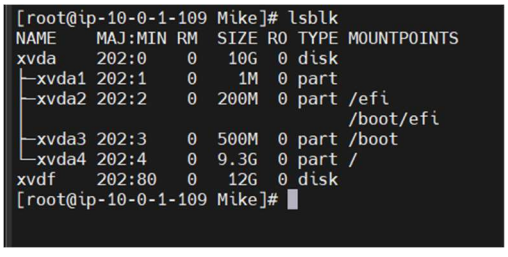
- first attach the 12G EBS then partition it into 2 one 6G and 6G

then attach another EBS for the 2G space needed for Dockerstorage

- Need to make sure the users have their home directory with the size requested and the users should be isolated
- Create IAM user account for the user and give access to ec2 machine
- Each user should login to their respective home directory using SSH key pair authentication method

Phase 1:

Attached the 12G of volume



will partition it into 2 , each having 6G of storage

```
[root@ip-10-0-1-109 Mike]# fdisk /dev/xvdf

Welcome to fdisk (util-linux 2.37.4).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-25165823, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-25165823, default 25165823): +6G

Created a new partition 1 of type 'Linux' and of size 6 GiB.

Command (m for help): n
Partition type
   p   primary (1 primary, 0 extended, 3 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (2-4, default 2): 2
First sector (12584960-25165823, default 12584960):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (12584960-25165823, default 25165823):

Created a new partition 2 of type 'Linux' and of size 6 GiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

[root@ip-10-0-1-109 Mike]#
```

```
[root@ip-10-0-1-109 Mike]# 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 /efi
│                               /boot/efi
├─xvda3 202:3    0  500M  0 part /boot
└─xvda4 202:4    0  9.3G  0 part /
xvdf    202:80   0   12G  0 disk
├─xvdf1 202:81   0    6G  0 part
└─xvdf2 202:82   0    6G  0 part
[root@ip-10-0-1-109 Mike]#
```

Now we will make the both disks as physical volumes

```
[root@ip-10-0-1-109 Mike]# 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 /efi
│                               /boot/efi
├─xvda3 202:3    0  500M  0 part /boot
└─xvda4 202:4    0  9.3G  0 part /
xvdf    202:80   0   12G  0 disk
├─xvdf1 202:81   0    6G  0 part
└─xvdf2 202:82   0    6G  0 part
[root@ip-10-0-1-109 Mike]# pvcreate /dev/xvdf1
  Physical volume "/dev/xvdf1" successfully created.
  Creating devices file /etc/lvm/devices/system.devices
[root@ip-10-0-1-109 Mike]#
```

```
[root@ip-10-0-1-109 Mike]# pvs
  PV         VG Fmt  Attr PSize  PFree
  /dev/xvdf1    lvm2 ---   6.00g  6.00g
  /dev/xvdf2    lvm2 ---  <6.00g <6.00g
```

Now we will create a volume group

```
[root@ip-10-0-1-109 Mike]# vgcreate DB /dev/xvdf1
  Volume group "DB" successfully created
[root@ip-10-0-1-109 Mike]# vgs
  VG #PV #LV #SN Attr   VSize  VFree
  DB   1   0   0 wz--n- <6.00g <6.00g
```

Now we will add other pv to vg

```
[root@ip-10-0-1-109 Mike]# vgextend DB /dev/xvdf2
  Volume group "DB" successfully extended
[root@ip-10-0-1-109 Mike]# vgs
  VG #PV #LV #SN Attr   VSize  VFree
  DB   2   0   0 wz--n- 11.99g 11.99g
[root@ip-10-0-1-109 Mike]#
```

Now we will make logical volumes for new user Kiran in root /Kiran and Ted in /Ted, each having 6G storage

logical volume kiranlv of 6G is created by taking the storage from volume group DB for Kiran

```
  DB   2   0   0 wz--n- 11.99g 11.99g
[root@ip-10-0-1-109 Mike]# lvcreate -L 6G -n kiranlv DB
  Logical volume "kiranlv" created.
[root@ip-10-0-1-109 Mike]#
```

similarly, logical volume tedlv of 6G is created by taking the storage from volume group DB for ted

```
[root@ip-10-0-1-109 Mike]# lvcreate -L 5.99G -n tedlv DB
  Rounding up size to full physical extent 5.99 GiB
  Logical volume "tedlv" created.
[root@ip-10-0-1-109 Mike]# vgs
  VG #PV #LV #SN Attr   VSize  VFree
  DB   2   2   0 wz--n- 11.99g    0
[root@ip-10-0-1-109 Mike]# lvs
  LV      VG Attr       LSize Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  kiranlv DB -wi-a----- 6.00g
  tedlv   DB -wi-a----- 5.99g
[root@ip-10-0-1-109 Mike]#
```

Now we will mount the storage at /ted and /kiran

before:

```
[root@ip-10-0-1-109 Mike]# 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 /efi
                                       /boot/efi
├─xvda3       202:3    0  500M  0 part /boot
├─xvda4       202:4    0  9.3G  0 part /
xvdf          202:80   0   12G  0 disk
├─xvdf1       202:81   0    6G  0 part
│ └─DB-kiranlv 253:0   0    6G  0 lvm
└─xvdf2       202:82   0    6G  0 part
  ├─DB-kiranlv 253:0   0    6G  0 lvm
  └─DB-tedlv   253:1   0    6G  0 lvm
[root@ip-10-0-1-109 Mike]#
```

before mounting we need to format the filesystem of logical volume

```
[root@ip-10-0-1-109 Mike]# useradd Kiran -m -d /Kiran -s /bin/bash
[root@ip-10-0-1-109 Mike]# mount /dev/DB/kiranlv /kiran
mount: /kiran: mount point does not exist.
[root@ip-10-0-1-109 Mike]# mount /dev/DB/kiranlv /Kiran
mount: /Kiran: wrong fs type, bad option, bad superblock on /dev/mapper/DB-kiranlv, miss
ing codepage or helper program, or other error.
[root@ip-10-0-1-109 Mike]#
```

after:

```
[root@ip-10-0-1-109 Mike]# mkfs.ext4 /dev/DB/kiranlv
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 1572864 4k blocks and 393216 inodes
Filesystem UUID: 141134da-c67a-4594-87a2-9afba9c8e0f5
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
```

```
[root@ip-10-0-1-109 Mike]# mount /dev/DB/kiranlv /Kiran
[root@ip-10-0-1-109 Mike]# 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 /efi
│                                      /boot/efi
├─xvda3        202:3    0  500M  0 part /boot
└─xvda4        202:4    0  9.3G  0 part /
xvdf           202:80   0   12G  0 disk
├─xvdf1        202:81   0    6G  0 part
│ └─DB-kiranlv 253:0    0    6G  0 lvm  /Kiran
└─xvdf2        202:82   0    6G  0 part
  ├─DB-kiranlv 253:0    0    6G  0 lvm  /Kiran
  └─DB-tedlv   253:1    0    6G  0 lvm
```

Similarly , we have mounted for Ted

```
[root@ip-10-0-1-109 Mike]# useradd Ted -m -d /Ted -s /bin/bash
[root@ip-10-0-1-109 Mike]# 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 /efi
│                                      /boot/efi
├─xvda3        202:3    0  500M  0 part /boot
└─xvda4        202:4    0  9.3G  0 part /
xvdf           202:80   0   12G  0 disk
├─xvdf1        202:81   0    6G  0 part
│ └─DB-kiranlv 253:0    0    6G  0 lvm  /Kiran
└─xvdf2        202:82   0    6G  0 part
  ├─DB-kiranlv 253:0    0    6G  0 lvm  /Kiran
  └─DB-tedlv   253:1    0    6G  0 lvm
[root@ip-10-0-1-109 Mike]# mkfs.ext4 /dev/DB/tedlv
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 1570816 4k blocks and 393216 inodes
Filesystem UUID: b881024b-e033-4013-8954-f5d86aaf41dc
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

[root@ip-10-0-1-109 Mike]# mount /dev/DB/tedlv /Ted
[root@ip-10-0-1-109 Mike]# 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 /efi
│                                      /boot/efi
├─xvda3        202:3    0  500M  0 part /boot
└─xvda4        202:4    0  9.3G  0 part /
xvdf           202:80   0   12G  0 disk
├─xvdf1        202:81   0    6G  0 part
│ └─DB-kiranlv 253:0    0    6G  0 lvm  /Kiran
└─xvdf2        202:82   0    6G  0 part
  ├─DB-kiranlv 253:0    0    6G  0 lvm  /Kiran
  └─DB-tedlv   253:1    0    6G  0 lvm  /Ted
[root@ip-10-0-1-109 Mike]#
```

Mike have requested an addition of 2G storage for storing the large files but we don't have any other space left in VG to give

```
[root@ip-10-0-1-109 Mike]# vgs
  VG #PV #LV #SN Attr   VSize  VFree
  DB   2   2   0 wz--n- 11.99g    0
[root@ip-10-0-1-109 Mike]#
```

Now we will add another 2G volume for Mike to store the large files in /Home/Mike/Backup

we have attached another 2G of volume

```
[root@ip-10-0-1-109 Mike]# 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 /efi
│                                      /boot/efi
├─xvda3        202:3    0  500M  0 part /boot
└─xvda4        202:4    0  9.3G  0 part /
xvdf           202:80   0   12G  0 disk
├─xvdf1        202:81   0    6G  0 part
│ └─DB-kiranlv 253:0    0    6G  0 lvm  /Kiran
└─xvdf2        202:82   0    6G  0 part
  ├─DB-kiranlv 253:0    0    6G  0 lvm  /Kiran
  └─DB-tedlv   253:1    0    6G  0 lvm  /Ted
xvdg           202:96   0    2G  0 disk
```

Now we will do the similar process for creation of partition and PV

while partitioning we have used entire storage this time

```
[root@ip-10-0-1-109 Mike]# fdisk /dev/xvdg

Welcome to fdisk (util-linux 2.37.4).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x8fe4273e.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-4194303, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-4194303, default 4194303):

Created a new partition 1 of type 'Linux' and of size 2 GiB.
```

```
[root@ip-10-0-1-109 Mike]# 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 /efi
│                                      /boot/efi
├─xvda3        202:3    0  500M  0 part /boot
└─xvda4        202:4    0  9.3G  0 part /
xvdf           202:80   0   12G  0 disk
├─xvdf1        202:81   0    6G  0 part
│ └─DB-kiranlv 253:0    0    6G  0 lvm  /Kiran
└─xvdf2        202:82   0    6G  0 part
  ├─DB-kiranlv 253:0    0    6G  0 lvm  /Kiran
  └─DB-tedlv   253:1    0    6G  0 lvm  /Ted
xvdg           202:96   0    2G  0 disk
└─xvdg1        202:97   0    2G  0 part
```

PV created

```
[root@ip-10-0-1-109 Mike]# pvcreate /dev/xvdg1
  Physical volume "/dev/xvdg1" successfully created.
```

We extended the volume of existing VG named DB

```
[root@ip-10-0-1-109 Mike]# vgextend DB /dev/xvdg1
  Volume group "DB" successfully extended
```

Now VG have free space of 2G

```
[root@ip-10-0-1-109 Mike]# vgs
  VG #PV #LV #SN Attr   VSize   VFree
  DB   3   2   0 wz--n- <13.99g <2.00g
```

Now we will mount 2G volume for Mike to store the large files in /Home/Mike/Backup

```
[root@ip-10-0-1-109 Mike]# lvcreate -L 1.99G -n Mikelv DB
  Rounding up size to full physical extent 1.99 GiB
  Logical volume "Mikelv" created.
[root@ip-10-0-1-109 Mike]# vgs
```

```
[root@ip-10-0-1-109 Mike]# 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 /efi
│                                /boot/efi
├─xvda3       202:3    0  500M  0 part /boot
└─xvda4       202:4    0  9.3G  0 part /
xvdf          202:80   0   12G  0 disk
├─xvdf1       202:81   0    6G  0 part
│ └─DB-kiranlv 253:0    0    6G  0 lvm  /Kiran
└─xvdf2       202:82   0    6G  0 part
  ├─DB-kiranlv 253:0    0    6G  0 lvm  /Kiran
  └─DB-tedlv  253:1    0    6G  0 lvm  /Ted
xvdg          202:96   0    2G  0 disk
└─xvdg1       202:97   0    2G  0 part
  └─DB-Mikelv 253:2    0    2G  0 lvm
```

```
└─DB-tedlv  253:1    0    6G  0 lvm  /Ted
xvdg          202:96   0    2G  0 disk
└─xvdg1       202:97   0    2G  0 part
  └─DB-Mikelv 253:2    0    2G  0 lvm  /home/Mike/Backup
```

df -h will give all the storage details

```
[root@ip-10-0-1-109 Mike]# df -h
Filesystem              Size  Used Avail Use% Mounted on
devtmpfs                4.0M     0  4.0M   0% /dev
tmpfs                   1.8G     0  1.8G   0% /dev/shm
tmpfs                   730M   17M  713M   3% /run
/dev/xvda4              9.4G  1.6G  7.8G  18% /
/dev/xvda3              495M  246M  250M  50% /boot
/dev/xvda2              200M   20K  200M   1% /efi
tmpfs                   365M     0  365M   0% /run/user/1000
tmpfs                   365M     0  365M   0% /run/user/1001
/dev/mapper/DB-kiranlv  5.9G   24K  5.6G   1% /Kiran
/dev/mapper/DB-tedlv    5.9G   24K  5.5G   1% /Ted
/dev/mapper/DB-Mikelv   2.0G   24K  1.9G   1% /home/Mike/Backup
[root@ip-10-0-1-109 Mike]#
```

To make any of the storage as permanent
edit fstab file using vi /etc/fstab

then copy the path of the mounting point the update all details

```
UUID=016e7284-31f6-4c28-bbb1-98c5b25fcdbf       /       xfs     defaults        0       0
UUID=48ebf8a2-a37f-4e53-9bf6-d77493ca7700       /boot   xfs     defaults        0       0
UUID=7B77-95E7  /boot/efi       vfat    defaults,uid=0,gid=0,umask=077,shortname=winnt  0       2
/dev/mapper/DB-kiranlv                          /Kiran  ext4    defaults        0       0
~
~
```

Now we have successfully shared the storage space across the users need by attaching additional volume.

Now we will completely remove the storage that have attached

for that we need to do it in reverse order

1.unmount the storage

2.remove the LVs

3.remove the VGs

4.remove the EBS storage