Linux Volume Management

1. Introduction to Linux Volume and EBS
2. Physical Volume, Logical Volume, and Volume Group
3. Mounting Volume in Linux
4. What is Logical Volume Management
5. Managing EBS on an EC2 instance
6. Using LVM with EBS for dynamic storage management

1. **A. Linux Volume -** A Linux volume is a flexible storage created using Logical Volume Management (LVM), which enables resizing and easy disk management. It lets you combine, split, extend, or shrink storage without affecting the system.
   B. EBS (Elastic Block Storage) - EBS is AWS's cloud-based block storage service that provides persistent, high-performance virtual disks for EC2 instances. It acts like a physical hard drive that you can attach, detach, and resize.

2. **A. Physical Volume -** A Physical Volume is a real storage device (like an EBS disk or partition) that LVM uses as raw storage. It forms the base layer on which volume groups are built.
   B. Volume Group - A Volume Group is a pool created by combining one or more physical volumes. It provides a large, flexible space from which logical volumes can be allocated.
   C. Logical Volume - A Logical Volume is a virtual partition carved out of a volume group. It acts like a normal disk partition where you can create filesystems and mount directories.

Creating and Attaching Volumes to an EC2 instance:
1. Go to the Elastic Block Storage section
2. Select volumes option
3. Click on create volume and create 3 volumes with 10GB,12GB, and 14GB.
4. Attach all three volumes to the running EC2 instance.

Note: Recommended device names for Linux: /dev/sda1 for the root volume. /dev/sd[f-p] for data volumes. --> /dev/sdf ---> xvdf

To do this process, become the root user in your Linux system.

1. To check the list of all the blocks: → **lsblk**

3. What is Mounting - Mounting means connecting a storage device to a folder so your computer can see and use the files inside it.

4. Logical Volume Management - is a Linux system that lets you manage storage more flexibly.
In simple terms, it allows you to create, resize, and combine storage spaces easily, without worrying about fixed disk partitions.

To perform logical volume management, become the root user and use lvm and do all the processes inside the lvm prompt.

After adding and attaching the volumes to the EC2 instance, see the list of the blocks; it will show you the 3 blocks you attached.


```powershell
$ lsblk
NAME         MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
xvda         202:0    0    30G  0 disk
└─xvda1      202:1    0    30G  0 part /
xvdf         202:80   0    10G  0 disk
xvdg         202:96   0    12G  0 disk
xvdh         202:112  0    14G  0 disk
```

Now, all three volumes are presented as a disk, blocks. To use attached volumes, we need to convert them into physical volumes. We can create a group of volumes.

1. To create a physical volume, we use the **"pvcreate"** command, which is used to **initialise a physical volume (PV).**

```bash
pvcreate /dev/xvdf /dev/xvdg /dev/xvdh
```

2. To check physical volume, use the **"pvs"** command in Linux **to list all physical volumes** that are part of LVM. It shows their size, volume group, free space, and other attributes.

```bash
pvs
```

```bash
PV           VG      Fmt  Attr PSize  PFree
/dev/xvdf            lvm2 ---  10.00g 10.00g
/dev/xvdg            lvm2 ---  12.00g 12.00g
/dev/xvdh            lvm2 ---  14.00g 14.00g
```

3. After creating a physical volume, create the volume group. Use the **"vgcreate"** command and check the volume group with the **"vgs"** command.

```bash
vgcreate myvol_vg /dev/xvdf /dev/xvdg
```

```csharp
Volume group "vol_vg" successfully created
```

```bash
vgs
```

Sample output:

```less
VG        #PV #LV #SN Attr   VSize  VFree
myvol_vg   2   0   0 wz--n- 22.00g 22.00g
```

4. Logical volume in Linux is used to **create a Logical Volume (LV)** from a Volume Group (VG). Use the **"lvcreate"** command and use the **"lvs"** command to check the logical volume.

```bash
lvcreate -L 10G -n myvol_lv myvol_vg
```

```nginx
Logical volume "myvol_lv" created.
```

```css
LV        VG        Attr       LSize
myvol_lv  myvol_vg  -wi-a----- 10.00g
```

How to mount Volumes:

1. First, create a directory.

```bash
mkdir /mnt/myvol_lv_mount
```

2. format logical volumes - means create a file system.

```bash
mkfs.ext4 /dev/myvol_vg/myvol_lv
```

3. Mounting logical volume, the logical volume's storage will be accessible at /mnt/myvol_lv_mount.

```
mount /dev/myvol_vg/myvol_lv /mnt/myvol_lv_mount
```

4. Do **"lsblk"** and check now your physical volumes combined, and make a logical volume with a mount location.

```powershell
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda                202:0    0   30G  0 disk
└─xvda1             202:1    0   30G  0 part /
xvdf                202:80   0   10G  0 disk
xvdg                202:96   0   12G  0 disk
myvol_vg-myvol_lv
                    253:0    0   10G  0 lvm  /mnt/myvol_lv_mount
```

To unmount the logical volume, use the **"umount"** command. It will unmount the volumes.

```bash
umount /mnt/myvol_lv_mount
```

Difference between Volume Attach and Mount Attach:

a. Volume Attach: Attaching a storage device (like an AWS EBS volume) to a server (EC2 instance), at the **cloud/hardware level**.
   Purpose: Makes the disk **visible to the operating system** as a block device.
   Example: In AWS, you attach a 10GB EBS volume to an EC2 instance. After attaching, lsblk will show it as a new disk.

b. Mount Attach: Making a **storage device accessible in the filesystem** by linking it to a directory (mount point) inside the **operating system**.
Purpose: Let you **read/write files** on the disk using a directory path.
Example: After attaching an EBS volume and formatting it, you mount it.

5. Managing EBS on an EC2 instance

A. How to mount Disk Volumes:

```
mkdir /mnt/mydisk
```

```bash
mkfs -t ext4 /dev/xvdh
```

```bash
mount /dev/xvdh /mnt/disk_mount
```
```
loop1               7:1    0 73.9M  1 loop /snap/core22/2133
loop2               7:2    0   74M  1 loop /snap/core22/2163
loop3               7:3    0 50.8M  1 loop /snap/snapd/25202
loop4               7:4    0 50.9M  1 loop /snap/snapd/25577
xvda              202:0    0   20G  0 disk
├─xvda1           202:1    0    7G  0 part /
├─xvda14          202:14   0    1G  0 part
├─xvda15          202:15   0  106M  0 part /boot/efi
└─xvda16          259:0    0  913M  0 part /boot
xvdf              202:80   0   10G  0 disk
└─vol_vg-vol_lv   252:0    0   10G  0 lvm  /mnt/vol_lv_mount
xvdg              202:96   0   12G  0 disk
└─vol_vg-vol_lv   252:0    0   25G  0 lvm  /mnt/vol_lv_mount
xvdh              202:112  0   14G  0 disk /mnt/disk_mount
```

B. Extend the logical volume

```bash
sudo lvextend -L +10G /dev/myvg/mylv
```

```
loop0                7:0    0 27.6M  1 loop /snap/amazon-ssm-agent/11797
loop1                7:1    0 68.9M  1 loop /snap/core22/2133
loop2                7:2    0   74M  1 loop /snap/core22/2163
loop3                7:3    0 50.8M  1 loop /snap/snapd/25202
loop4                7:4    0 50.9M  1 loop /snap/snapd/25577
xvda               202:0    0   20G  0 disk
├─xvda1            202:1    0    7G  0 part /
├─xvda5            202:5    0  106M  0 part /boot/efi
└─xvda16           259:0    0  913M  0 part /boot
xvdf               202:80   0   10G  0 disk
└─vol_vg-vol_lv    252:0    0   15G  0 lvm  /mnt/vol_lv_mount
xvdg               202:96   0   12G  0 disk
└─vol_vg-vol_lv    252:0    0   25G  0 lvm  /mnt/vol_lv_mount
xvdh               202:112  0   20G  0 disk /mnt/disk_mount
```

C. How to reduce the logical volume:

```bash
lvreduce -L 5GB /dev/vol_vg/vol_lv
```

```
lvm> lvreduce -L 5GB /dev/vol_vg/vol_lv
WARNING: Reducing active and open logical volume to 5.00 GiB.
THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce vol_vg/vol_lv? [y/n]: y
Size of logical volume vol_vg/vol_lv changed from 15.00 GiB (3840 extents) to 5.00 GiB (1280 
Logical volume vol_vg/vol_lv successfully resized.
```

```
loop1                 7:1    0  73.9M  1 loop /snap/core22/2133
loop2                 7:2    0    74M  1 loop /snap/core22/2163
loop3                 7:3    0  50.8M  1 loop /snap/snapd/25202
loop4                 7:4    0  50.9M  1 loop /snap/snapd/25577
xvda                202:0    0     8G  0 disk
├─xvda1             202:1    0     7G  0 part /
├─xvda14            202:14   0     4M  0 part
├─xvda15            202:15   0   106M  0 part /boot/efi
└─xvda16            259:0    0   913M  0 part /boot
xvdf                202:80   0    10G  0 disk
xvdg                202:96   0    12G  0 disk
└─vol_vg-vol_lv     252:0    0     5G  0 lvm  /mnt/vol_lv_mount
xvdh                202:112  0    14G  0 disk /mnt/disk_mount
root@ip-172-31-3-85:/home/ubuntu#
```

Happy Learning, Enjoy:)