# Linux Volume Management

Before we begin we want to clean up our disk structure to make sure we don't have any left over partitions or mounted drives. They will cause errors.

Unmount all your drives:

```
sudo umount -a
```

If you get error **`unmount: /dev/sdXX: not found`** - don't panic - that drive just doesn't exist, we're going to make it.

Wipe all your drives (this overwrites the MBR where the partition table lives):

```
sudo dd if=/dev/zero of=/dev/sdb bs=512 count=1
sudo dd if=/dev/zero of=/dev/sdc bs=512 count=1
sudo dd if=/dev/zero of=/dev/sdd bs=512 count=1
sudo dd if=/dev/zero of=/dev/sde bs=512 count=1
```

The dd command is a very powerful and useful command. You can easily copy entire hard disks to a file for backup using this command.
The command above is copying from /dev/zero (if=source) to /dev/sdb (of=destination) with block size (bs) of 512 bytes. The command will be run 1 time (count)
The result is an overwrite of the master boot record (which contains all partition information)

```
sudo wipefs --all /dev/sdb /dev/sdc /dev/sdd /dev/sde
```

The result is a completely wiped disk - no data or partition information is left behind (so be careful with this command!)

Read more about the dd command on Wikipedia http://en.wikipedia.org/wiki/Dd_(Unix)

## Use gdisk to delete all partitions on all drives. (option d = delete partition)

# Linux Volume Management

## Partition the drives

Using the gdisk command create a single partition on each drive. The partition should be the entire size of the drive.

```
sudo gdisk /dev/sdb
sudo gdisk /dev/sdc
sudo gdisk /dev/sdd
sudo gdisk /dev/sde
```

If you are unsure of the steps to follow, refer to last week's assignment for step-by-step.
You will need to use gdisk to set the type of partition. We need to set the partition type to **Linux LVM**

```
Command (? for help): n
Partition number (1-128, default 1): [enter]
First sector (34-20971486, default = 2048) or {+-}size{KMGTP}: [enter]
Last sector (2048-20971486, default = 20971486) or {+-}size{KMGTP}: [enter]
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300): L

0700 Microsoft basic data  0c01 Microsoft reserved    2700 Windows RE
3000 ONIE boot             3001 ONIE config           3900 Plan 9
4100 PowerPC PReP boot     4200 Windows LDM data      4201 Windows LDM metadata
4202 Windows Storage Spac  7501 IBM GPFS              7f00 ChromeOS kernel
7f01 ChromeOS root         7f02 ChromeOS reserved     8200 Linux swap
8300 Linux filesystem      8301 Linux reserved        8302 Linux /home
8303 Linux x86 root (/)    8304 Linux x86-64 root (/  8305 Linux ARM64 root (/)
8306 Linux /srv            8307 Linux ARM32 root (/)  8400 Intel Rapid Start
8e00 Linux LVM             a500 FreeBSD disklabel     a501 FreeBSD boot
a502 FreeBSD swap          a503 FreeBSD UFS           a504 FreeBSD ZFS
a505 FreeBSD Vinum/RAID    a580 Midnight BSD data     a581 Midnight BSD boot
a582 Midnight BSD swap     a583 Midnight BSD UFS      a584 Midnight BSD ZFS
a585 Midnight BSD Vinum    a600 OpenBSD disklabel     a800 Apple UFS
a901 NetBSD swap           a902 NetBSD FFS            a903 NetBSD LFS
a904 NetBSD concatenated   a905 NetBSD encrypted      a906 NetBSD RAID
ab00 Recovery HD           af00 Apple HFS/HFS+        af01 Apple RAID
af02 Apple RAID offline    af03 Apple label           af04 AppleTV recovery
af05 Apple Core Storage    bc00 Acronis Secure Zone   be00 Solaris boot
bf00 Solaris root          bf01 Solaris /usr & Mac Z  bf02 Solaris swap
bf03 Solaris backup        bf04 Solaris /var          bf05 Solaris /home
bf06 Solaris alternate se  bf07 Solaris Reserved 1    bf08 Solaris Reserved 2
Press the <Enter> key to see more codes: [enter]
```



```
bf09 Solaris Reserved 3    bf0a Solaris Reserved 4    bf0b Solaris Reserved 5
c001 HP-UX data            c002 HP-UX service         ea00 Freedesktop $BOOT
eb00 Haiku BFS             ed00 Sony system partitio  ed01 Lenovo system partit
ef00 EFI System            ef01 MBR partition scheme  ef02 BIOS boot partition
f800 Ceph OSD              f801 Ceph dm-crypt OSD     f802 Ceph journal
f803 Ceph dm-crypt journa  f804 Ceph disk in creatio  f805 Ceph dm-crypt disk i
fb00 VMWare VMFS           fb01 VMWare reserved       fc00 VMWare kcore crash p
fd00 Linux RAID
Hex code or GUID (L to show codes, Enter = 8300): 8E00
```

Use the gdisk command to list the new partitions and make sure that they are all set to ‘Linux LVM:

```
sudo gdisk -l /dev/sdb
sudo gdisk -l /dev/sdc
sudo gdisk -l /dev/sdd
sudo gdisk -l /dev/sde
```

Each disk should have a partition line that looks like this:

```
Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048        20971486   10.0 GiB    8E00  Linux LVM
```

If it doesn’t say ‘Linux LVM’, re-run gdisk for that partition and change the type using the ‘t’ option.

# Linux Volume Management

## Creating Physical Volumes

Before we can group our disks into Volume Groups, we have to tell LVM which disks we are going to use. We use pvcreate to create new physical volumes:

```
sudo pvcreate /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1
```

```
  Physical volume "/dev/sdb1" successfully created
  Physical volume "/dev/sdc1" successfully created
  Physical volume "/dev/sdd1" successfully created
  Physical volume "/dev/sde1" successfully created
```

And we can verify that they were created with pvdisplay:

```
sudo pvdisplay --short
```

```
  Device "/dev/sde1" has a capacity of 10.00 GiB
  Device "/dev/sdd1" has a capacity of 10.00 GiB
  Device "/dev/sdb1" has a capacity of 10.00 GiB
  Device "/dev/sdc1" has a capacity of 10.00 GiB
```

If you want more detail about each physical volume, use the pvdisplay command without any optons:

```
sudo pvdisplay
```

```
 "/dev/sdb1" is a new physical volume of "10.00 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/sdb1
  VG Name
  PV Size               10.00 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               iGdGsV-61Z8-1wT0-lcoJ-Ou2A-Xfp4-4b1huY
```

None of the volumes are ‘allocatable’ because they are not a member of a Volume Group

# Linux Volume Management

## Creating a Volume Group

The Volume Group will contain our Physical Volumes and from that group we will be able to 'carve' out our Logical Volumes:

```
sudo vgcreate cetyvolgroup /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1
```

```
Volume group "cetyvolgroup" successfully created
```

If we use pvdisplay, we can see that our volumes have been allocated to a Volume Group:

```
sudo pvdisplay /dev/sdb1
```

```
  --- Physical volume ---
   PV Name               /dev/sdb1
   VG Name               cetyvolgroup
   PV Size               10.00 GiB / not usable 2.98 MiB
   Allocatable           yes
   PE Size               4.00 MiB
   Total PE              2559
   Free PE               2559
   Allocated PE          0
   PV UUID               iGdGsV-61Z8-1wT0-lcoJ-Ou2A-Xfp4-4b1huY
```

PE = Physical Extents (smallest allocatable unit)



And, display the Volume Group:

```
sudo vgdisplay cetyvolgroup
```

```
 --- Volume group ---
  VG Name               cetyvolgroup
  System ID
  Format                lvm2
  Metadata Areas        4
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                4
  Act PV                4
  VG Size               39.98 GiB
  PE Size               4.00 MiB
  Total PE              10236
  Alloc PE / Size       0 / 0
  Free  PE / Size       10236 / 39.98 GiB
  VG UUID               07L12G-zCqd-DI6E-14dc-fGP9-fG6t-hJaLAR
```

PE = physical extent
VG = volume group

metadata uses up a small amount of disk space (20 MB in this case). It contains configuration data for the Volume Group.

We can use pvscan to see which Physical Volumes are allocated to which Volume Group:

```
sudo pvscan
```

```
  PV /dev/sdb1   VG cetyvolgroup    lvm2 [10.00 GiB / 10.00 GiB free]
  PV /dev/sdc1   VG cetyvolgroup    lvm2 [10.00 GiB / 10.00 GiB free]
  PV /dev/sdd1   VG cetyvolgroup    lvm2 [10.00 GiB / 10.00 GiB free]
  PV /dev/sde1   VG cetyvolgroup    lvm2 [10.00 GiB / 10.00 GiB free]
  Total: 4 [39.98 GiB] / in use: 4 [39.98 GiB] / in no VG: 0 [0   ]
```

# Linux Volume Management

## Creating Logical Volumes

Now that we have Volume Group space, we can 'carve out' some Logical Volumes to mount:

```
sudo lvcreate --name lv_linuxuser --size 5G cetyvolgroup
```

```
  Logical volume "lv_linuxuser" created
```

Let's take a look at our Volume Group and see how things have changed, now that we've allocated space out of it:

```
sudo vgdisplay cetyvolgroup
```

```
 --- Volume group ---
  VG Name               cetyvolgroup
  System ID
  Format                lvm2
  Metadata Areas        4
  Metadata Sequence No  2
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                1
  Open LV               0
  Max PV                0
  Cur PV                4
  Act PV                4
  VG Size               39.98 GiB
  PE Size               4.00 MiB
  Total PE              10236
  Alloc PE / Size       1280 / 5.00 GiB
  Free  PE / Size       8956 / 34.98 GiB
  VG UUID               07L12G-zCqd-DI6E-14dc-fGP9-fG6t-hJaLAR
```

Because of overhead with metadata and volume border rounding, we don't have 35Gb of space to allocate to our next Logical Volume, we have 34.98 Gb:

```
sudo lvcreate --name lv_media --size 34.98G cetyvolgroup
```

```
  Rounding up size to full physical extent 34.98 GiB
  Logical volume "lv_media" created.
```

# Linux Volume Management

Let's take a look at the Logical Volumes we just created:

```
sudo lvdisplay /dev/cetyvolgroup/lv_media
sudo lvdisplay /dev/cetyvolgroup/lv_linuxuser
```

```
 --- Logical volume ---
  LV Path                /dev/cetyvolgroup/lv_media
  LV Name                lv_media
  VG Name                cetyvolgroup
  LV UUID                YL4uNv-wiu0-py6y-LBcA-jEmd-gNi2-OtwBhY
  LV Write Access        read/write
  LV Creation host, time cushing-dave, 2018-03-18 16:51:40 +0000
  LV Status              available
  # open                 0
  LV Size                34.98 GiB
  Current LE             8955
  Segments               4
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:1

 --- Logical volume ---
  LV Path                /dev/cetyvolgroup/lv_linuxuser
  LV Name                lv_linuxuser
  VG Name                cetyvolgroup
  LV UUID                nORPBW-A2Wh-KTaX-C56s-zc2Q-PDLJ-RlRwjV
  LV Write Access        read/write
  LV Creation host, time cushing-dave, 2018-03-18 16:48:46 +0000
  LV Status              available
  # open                 0
  LV Size                5.00 GiB
  Current LE             1280
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:0
```

# Linux Volume Management

Let's format the volumes, create mount points and mount the volumes:

```
sudo mkfs -t ext4 /dev/cetyvolgroup/lv_linuxuser
sudo mkfs -t ext4 /dev/cetyvolgroup/lv_media

sudo mkdir /home/linuxuser/lvol
sudo mkdir /media/volume2

sudo mount -t ext4 /dev/cetyvolgroup/lv_linuxuser /home/linuxuser/lvol
sudo mount -t ext4 /dev/cetyvolgroup/lv_media /media/volume2
sudo chown linuxuser:linuxuser /home/linuxuser/lvol

mount | grep "cetyvol"
```

```
/dev/mapper/cetyvolgroup-lv_linuxuser on /home/linuxuser/lvol type ext4 (rw)
/dev/mapper/cetyvolgroup-lv_media on /media/volume2 type ext4 (rw)
```