Configuring Local Storage & Filesystems

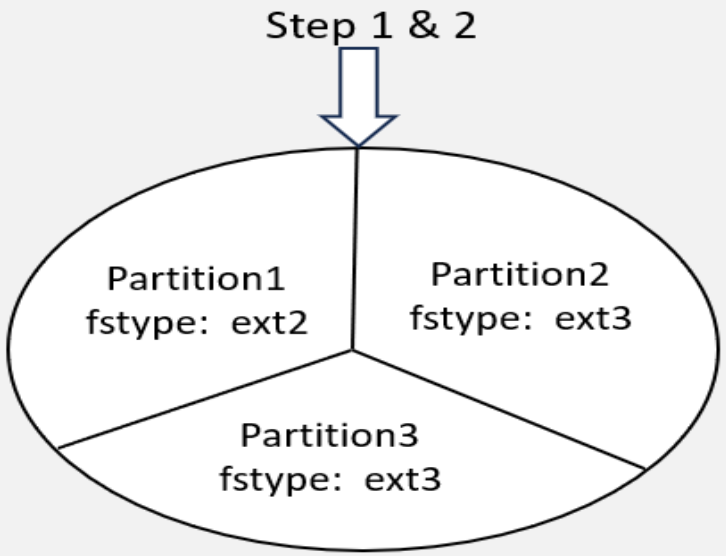
- List, create, delete partitions on MBR and GPT disks
- Create and remove physical volumes
- Assign physical volumes to volume groups
- Create and delete logical volumes
- Extend existing logical volumes
- Create and configure file systems
- Configure systems to mount file systems at boot by universally unique ID (UUID) or label
- Add swap to a system non-destructively
- Mount and unmount network file systems using NFS
- Create and configure set-GID directories for collaboration
- Configuring autofs (Already covered)

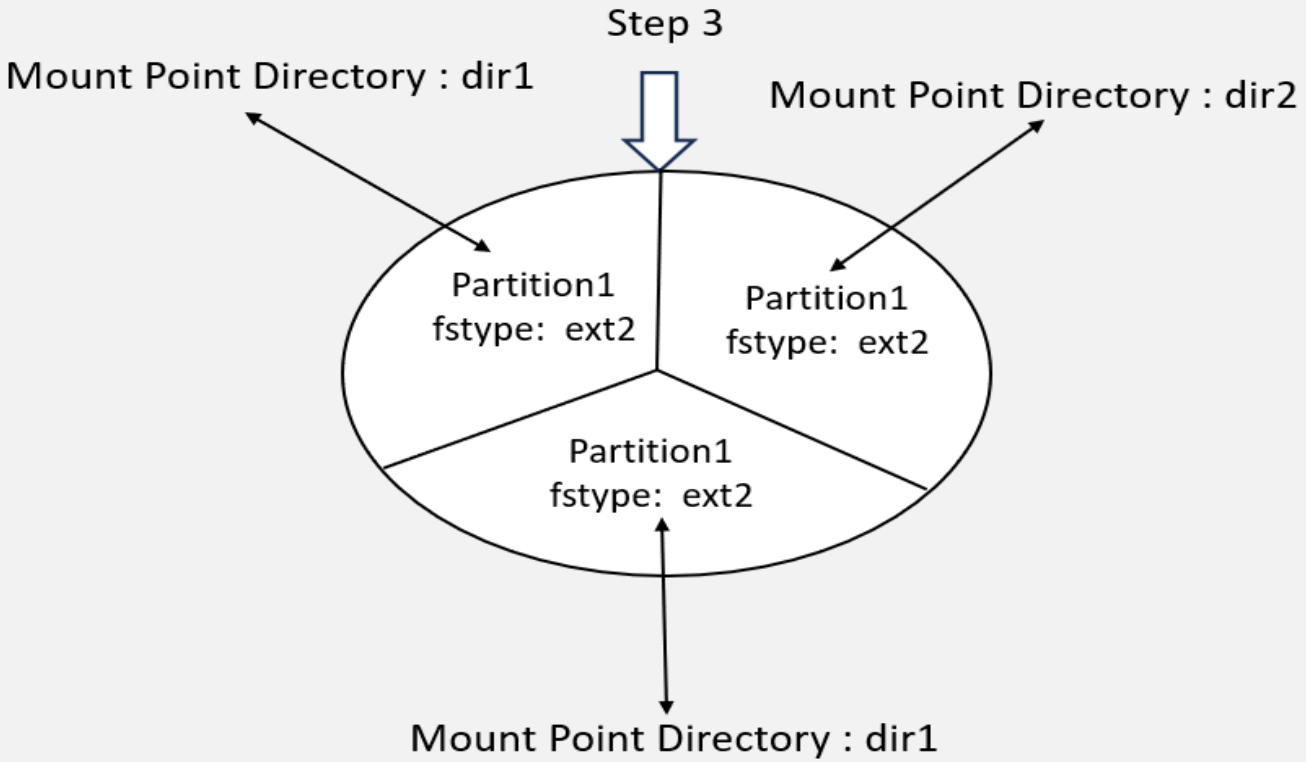
Introducing Standard Partitions and Filesystems

Step1 : Create partition(fdisk)

Step2: Create filesystem(mkfs)

Step3 : Mount filesystem(mount or fstab entry for persistent mount)



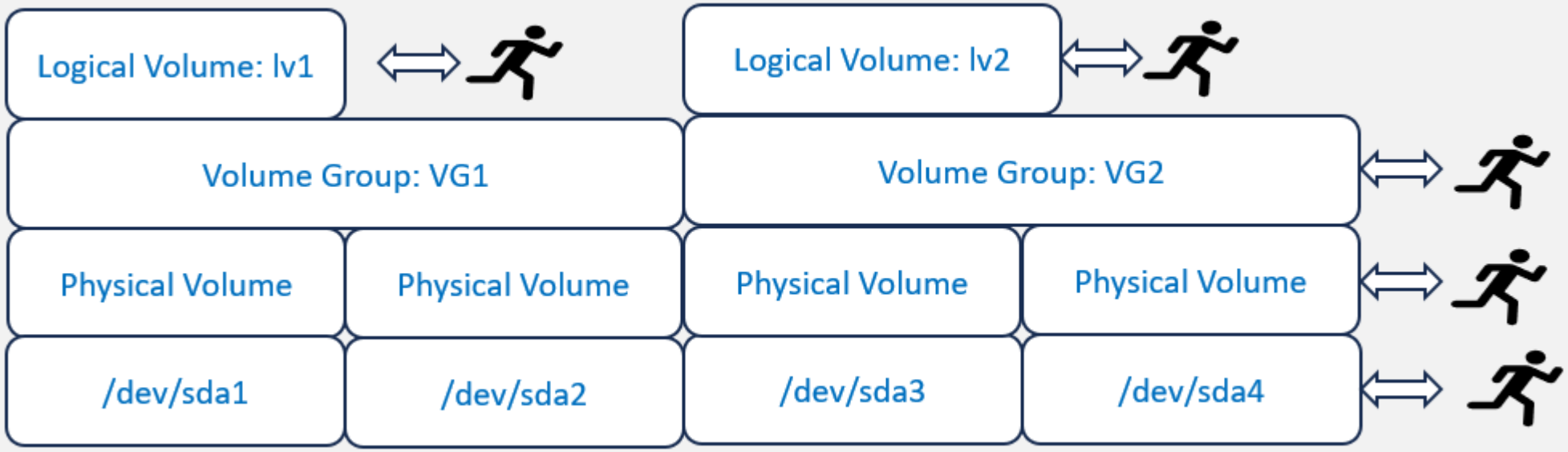


MBR Partitioning Layout - 15 Partitions (3 Primary,12 Logical)

GPT Partitioning Layout - 128 Partitions

Introducing Logical Volumes (LVM's)

Step1: Create physical volume from disks or partitions (pvcreate)
Step2: Create volume group using physical volume(s)(vgcreate)
Step3: Create logical volume (lvcreate)
Step4: Create filesystem on logical volume (mkfs)
Step5: Mount file system on mount point(mount or fstab entry for persistent mount)



Introducing fdisk/gdisk command

fdisk is mainly used to manipulate MBR disk partition table but it also understands GPT and some other partitioning tables too . So, to manage partitions on MBR disk we should use fdisk command line. parted is an alternative to fdisk but fdisk is more convenient to use. We will stick to fdisk to manage MBR disk devices.

gdisk ,on the other hand is used to manipulate GUID partitioning table (GPT) . So, to manage partitions on GPT disks, we must use gdisk.

Both fdisk and gdisk are menu driven and very convenient to use and both have almost same options (commands) to manage disk partitions.

MBR partitioning table(MBR Disk) supports maximum disk size of 2TiB (Tebibytes) and GPT partitioning table (GPT disk) supports maximum disk size of 2.2 ZiB (Zebibytes) and 1 ZiB = 1073741824 TiB.

Hex codes for different partitions types for GPT disk are different than MBR disk (important codes are shown).

MBR Disk Partition types	GPT Disk Partition Types
Linux filesystem (83)	Linux filesystem (8300)
Linux swap (82)	Linux swap (8200)
Linux LVM (8e)	Linux LVM (8e00)

Introducing mkfs command

mkfs command is used to build (create) Linux filesystem.

Example usage :

- mkfs -t fstype device (default filesystem is ext2)
- mkfs -V -t fstype device (To produce verbose output)

mkfs frontend is deprecated in favour of filesystem specific utils.

Example usage:

- mkfs.fstype device
- mkfs.fstype -L VOL_LABEL device

Introducing mount command and fstab file

mount command is used to mount a filesystem in runtime.
Example usage :

- mount -t fstype device dir(mount point) (-t option can be omitted because it detects filesystem automatically)
- mount -o ro device dir

Default mount options are : rw, suid, dev, exec, auto, nouser and async

fstab:
fstab contains static information about filesystems to mount filesystems at boot. So, to mount the filesystem persistently, we must make entry in fstab (/etc/fstab) file.
Each filesystem is described on a separate line with six fields separated by spaces or tabs.

Example entry in fstab is:

/dev/sda5	/mnt/test	ext3	defaults	0	0

➢ Create a standard disk partition of 1 GiB size and build xfs filesystem on the partition.
- Mount this filesystem on /mnt/partition directory.
- Mount should be persistent.

Command	Action/Description
fdisk /dev/sda First input : **n** , Second input : **e** , Two times Enter(To assign remaining space for Logical partitions) ,Third input : **n** , Enter (Default First sector),Fourth input: **+1G** ,**w** (to save and quit)	To create container for extended partition (Container) and to create logical partition of **size 1 GiB**
partprobe	To inform kernel about this partition table changes
mkfs.xfs /dev/sda5	To create mount directory
mkdir /mnt/partition	To create **xfs** filesystem on partition
mount /dev/sda5 /mnt/partition	Mounting filesystem temporarily for testing
mount	To verify mounted filesystem
vim /etc/fstab /dev/sda5 /mnt/partition xfs defaults 0 0 :wq	To mount filesystem persistently (to persist across reboot)
mount -a	To mount through fstab
lsblk or fdisk -l	To list block devices

➢ Create a disk partition of size 200 MiB and mount this for read only access on /mnt/fat directory.
- Use vfat file system for the partition.
- Mount should be persistent.

Command	Action/Description
fdisk /dev/sda First input : **n** ,Enter(Default First sector), Second input : **+200M,** Enter ,**w** (to save and quit)	To create logical partition of **200 MiB** size
partprobe	To inform kernel about this partition
mkfs.vfat /dev/sda6	To create **vfat** filesystem on partition
mkdir /mnt/fat	To create mount point directory
mount -o ro /dev/sda6 /mnt/fat	To test mounting filesystem
mount	To verify mounted filesystem
vim /etc/fstab /dev/sda6 /mnt/fat vfat ro 0 0 :wq	To mount filesystem persistently
mount -a and lsblk --fs	To mount filesystem through fstab and then to list block devices

➢ Configure and add 1 GiB swap space to your System.

Command	Action/Description
fdisk /dev/sda First input : **n** ,Enter, Enter (Default First sector), Second input : **+1G** , Enter, Third input : **t** ,Enter ,Enter(for default Partition),Fourth input : **82** ,Enter, **w** (to save and quit),Enter	To create logical partition of **1 GiB** size of type **Linux swap**
partprobe	To inform kernel about these changes
mkswap /dev/sda7	To configure partition as swap
vim /etc/fstab /dev/sda7 none swap defaults 0 0 :wq	Make entry in **fstab** file
swapon -a	To activate swap as per entry in **fstab**
free -m	To verify added swap memory

➢ Configure logical volume with name lv_volume which should use 200 MiB from volume group vg_group of size 300 MiB.
- ext4 file system should be used.
- Mount this on /mnt/log_vol directory and mount should be persistent.

Command	Action/Description
fdisk /dev/sda First input : **n** ,Enter,Enter (Default first sector) , Second input :**+300M** ,Enter, Third input : **t**,Enter,Enter(for default Partition),Fourth input : **8e** ,Enter,**w** (to save and quit),Enter	To create logical partition of size **300 MiB** with type **Linux LVM**
partprobe	To inform kernel about this partition
pvcreate /dev/sda8	To create physical volume
vgcreate vg_group /dev/sda8	To create volume group from physical volume
lvcreate -n lv_volume -L 200M vg_group	To create logical volume on volume group
mkdir /mnt/log_vol	To create mount directory
mkfs -t ext4 /dev/vg_group/lv_volume	To create **ext4** filesystem on logical volume
mount /dev/vg_group/lv_volume /mnt/log_vol	To mount filesystem temporarily using mount
mount	To verify mounted filesystem
vim /etc/fstab /dev/vg_group/lv_volume /mnt/log_vol ext4 defaults 0 0 :wq	Make entry in **fstab** to mount filesystem persistently
mount -a	To mount through **fstab**

- Configure logical volume with name volume which should use 20 PEs from volume group named group of size 400 MiB.
  - Size of PE should be 16 MiB and file system used must be ext4 filesystem.
  - Mount this on /mnt/volume directory and mount should be persistent.
  - Use UUID to mount this.

Command	Action/Description
fdisk /dev/sda First input : **n** ,Enter,Enter (Default first sector), Second input : **+400M** ,Enter, Third input : **t**, Enter, Enter(for default Partition),Fourth input : **8e** ,Enter, **w** (to save and quit),Enter	To create logical partition
partprobe	To inform kernel about changes
pvcreate /dev/sda9	To create physical volume
vgcreate -s 16M group /dev/sda9	To create volume group with **PE size of 16 MiB**
lvcreate -n volume -l 20 group	To create logical volume using **20 PEs** from volume group
mkdir /mnt/volume	To create mount point
mkfs.ext4 /dev/group/volume	To create **ext4** filesystem on logical volume
mount /dev/group/volume /mnt/volume	To mount filesystem in runtime
mount	To verify mounted filesystem
vim /etc/fstab UUID =? /mnt/volume ext4 defaults 0 0 :wq	To mount persistently through **fstab**
mount -a	To verify mounted filesystem

- Configure logical volume with name lvm from volume group vgroup of size 512 MiB.
  - Logical volume should use total free space on volume group.
  - Create ext4 file system on this volume.
  - Mount the filesystem persistently on /mnt/test-vol using LABEL=test-vol

Command	**Action/Description**
fdisk /dev/sda First input : **n** Enter,Enter(Default first sector), Second input : **+512M** ,Enter, Third input : **t** ,Enter,Enter (Default partition), Fourth input : **8e**,Enter ,**w** (to save and quit),Enter	To create logical partition of type Linux LVM
pvcreate /dev/sda10	To create physical volume
vgcreate vgroup /dev/sda10	To create volume group
lvcreate -n lvm -l 100%FREE vgroup	To create logical volume using all space of volume group
mkfs.ext4 -L test-vol /dev/vgroup/lvm	To create ext4 filesystem
mkdir /mnt/test-vol	Creating mount point directory
vim /etc/fstab LABEL=test-vol /mnt/test-vol ext4 defaults 0 0	Making entry in fstab file to mount filesystem at boot time using LABEL
mount -a	To mount through fstab file

- Resize the lvm lv_volume so that after reboot size should be in between 230 MiB to 260 MiB.
  - Make sure complete logical volume should be usable.

Command	Action/Description
lvdisplay	To display logical volumes
lvextend -r -L +45M /dev/vg_group/lv_volume	To extent logical volume and resize the filesystem

- Extend size of LVM with name lvm by 256 MiB.
  - Create new partition to increase the size of volume group (vgroup) .
  - Format the complete volume with ext4 file system.

Command	Action/Description
fdisk /dev/sda First input : **n** ,Enter, Enter (Default first sector),Second input : **+300M** , Enter,Third input : **t** ,Enter ,Enter(Default partition) Fourth input : **8e** ,Enter, **w** (to save and quit),Enter	To create logical partition of type Linux LVM
pvcreate /dev/sda11	To create physical volume
vgextend vgroup /dev/sda11	To extend volume group
lvextend -r -L +256M /dev/vgroup/lvm	To extend logical volume
lvdisplay	To display logical volume(s)

➢ Create a standard partition of size 100 MiB and format this with ext4 file system.
- Change the file system to xfs and verify same.

Command	Action/Description
fdisk /dev/sda First input : **n** ,Enter,Enter(To select default first sector), Second input :**+100MiB** ,Enter, **w** (to save and quit),Enter	To create partition (Logical partition) of **100 MiB** size
partprobe	To inform kernel about partition table changes
mkfs.ext4 /dev/sda12	To create **ext4** filesystem on partition
mkfs.xfs -f /dev/sda12	To change the filesystem type to **xfs** or to create **xfs** filesystem
lsblk --fs	To verify changes

Mounting filesystems using Systemd mount unit

We already talked about different systemd unit types. systemd mount unit mounts the filesystem as per unit file definition.

Till now we mounted filesystems persistently through fstab and did not talk about background actions which happen to mount the filesystems at boot. This is where systemd-fstab-generator come into action.

systemd-fstab-generator reads fstab file at boot and generates native systemd mount unit files in run time for each filesystem entry in fstab ,so the filesystems are mounted at boot. These unit files are located under /run/systemd/generator/ directory.

It is better approach to mount the filesystems persistently through fstab file because systemd would automatically generate mount unit files for the filesystems with all dependencies.

However, we can mount the filesystem persistently by creating systemd mount unit file at path /etc/system/system with specific filename and .mount extension and then starting/enabling systemd unit.

If mount point directory is /mnt/test ,mount unit file name should be mnt-test.mount .

Introducing systemd and systemd units

systemd is System and Service manager for Linux operating systems and provides tooling for controlling, reporting and system initialization. **systemd** replaced Upstart as system initialization program since RHEL 7.

systemd (Process Id- 1) is the first process which is started during boot process and starts other system resources.

systemd manages the system through different objects ,in general known as **systemd units** which are representation of system resources and services. **systemd** unit consists of a name, type and configuration file that describes particular task .

Different types of systemd units:

- Service : Controls and manages different system services.
- Target : Represents a group of units for specific system state.
- Device : Manages hardware devices
- Mount : Handles file system mounting
- Timer : Manages task scheduling
- ……

systemctl utility is used to manage the **systemd** units .For example, you as system administrator can start/stop system services and enable system services to start at boot. In similar way , same utility can be used to manage other **systemd** units.

systemd configuration files:

- /usr/lib/systemd/system/
  Contains systemd unit files distributed with installed RPM packages.

- /run/systemd/system/
  Contains unit files created during run-time (Precedence over above directory)

- /etc/systemd/system/
  Contains unit files created with **systemctl enable** and unit files added by system admin.(Highest precedence)

Example systemd mount unit file :

vim /etc/systemd/system/mnt-test.mount

```
[Unit]
Description=Mounting filesystem

[Mount]
What=/dev/sda13
Where=/mnt/test
Type=xfs
Options=defaults

[Install]
WantedBy=multi-user.target
```

After creating unit file ,Start and enable systemd unit:

```
systemctl enable --now mnt-test.mount
```

Introducing special permission bits (SUID,SGID & Sticky bit)

SUID (Set User ID) :

SETUID bit configures the special permission at user access level and allows users to execute executables with permissions of owner of executable program regardless of user who is executing this. The use of this permission is restricted because of security concerns.

For example, passwd command can be used by regular user to set his password though this updates the /etc/shadow file which is owned by root user.

```
[root@system ~]# ls -l /usr/bin/passwd
-rwsr-xr-x. 1 root root 32648 Aug 10  2021 /usr/bin/passwd
[root@system ~]#
```

Setting SUID using symbolic representation method:
chmod u+s FILE_PATH

Setting SUID using numeric representation method:
chmod 4644 FILE_PATH (Assuming default permissions on file)

SGID (Set Group ID) :

SGID bit provides access at group level. If it is configured for a file, file can be executed as group owner of the file regardless of user who is executing this.

If set on directory, all future files under this directory will inherit the group ownership of parent directory instead of setting group ownership to the primary group of user who creates file under the directory. This allows file sharing among group members for Group collaboration.

```
[root@system ~]# ls -ld /group_collaboration/
drwxr-sr-x. 2 root root 6 Mar  5 01:07 /group_collaboration/
[root@system ~]#
```

Setting GUID using symbolic representation method:
chmod g+s DIR_PATH

Setting GUID using numeric representation method:
chmod 2755 DIR_PATH (Assuming default permissions on DIR)

Sticky bit :

Sticky bit restricts file deletion at directory level . If set on directory, files under the directory can be deleted by owner of file and root user.

For example, /tmp directory has sticky bit set. Files under /tmp directory can be deleted by root user and file owner. Others can not delete the files even if they have full permissions on file.

```
[root@system ~]# ls -ld /tmp
drwxrwxrwt. 17 root root 4096 Mar  5 01:15 /tmp
[root@system ~]#
```

Setting GUID using symbolic representation method:
chmod +t TEMP_DIR

- Create a directory /projects/docs.
  - Set the group ownership on this directory to group.
  - Give full permissions at group level on this directory.
  - User riya should have no access on this directory.
  - Add users lara and harry to group group.
  - Files created by users lara and harry under this directory should have group ownership set to group.

Command	Action/Description
mkdir -p /projects/docs	Creating directory path **/projects/docs**
chown :group /projects/docs	Changing group ownership
chmod g+rwx /projects/docs	Configuring full permissions at group level
setfacl -m u:riya:- /projects/docs	Removing all permissions for user **riya**
usermod -aG group lara & usermod -aG group harry	Adding users to group **group**
chmod g+s /projects/docs	To set GID bit
getfacl /projects/docs	To display configured access control lists

- Discover the NFS share exported by NFS server ipaserver.example.com.
  - Mount the share /nfsshare on directory /nfs/share and mount should be persistent.
  - NFS version 3 should be used.

Command	Action Description
dnf group install “Network File System Client”	Installing NFS client
showmount -e ipaserver.example.com	Discovering NFS exports
mkdir -p /nfs/share	Creating mount point directory
mount -o _netdev,nfsvers=3 ipaserver.example.com:/nfshare /nfs/share	Mounting NFS share in run time
umount /nfs/share	Unmounting NFS share
vim /etc/fstab ipaserver.example.com:/nfsshare /nfs/share nfs _netdev,nfsvers=3 0 0 :wq	Making entry in **fstab** file for persistent mount
mount -a	Mounting through **fstab**
mount	Verifying the mounted filesystem and **version**

➢ Discover the samba share and mount share samba on /samba/smb1 directory with smb1 user.
Use the password password to mount this share

Command	Action/Description
dnf install samba-client cifs-utils	Installing **Samba** client
smbclient -L ipaserver.example.com	Listing **Samba** shares
mkdir -p /samba/smb1	Creating mount point directory
mount -o _netdev,username=smb1 //ipaserver.example.com/samba /samba/smb1 Enter the Samba user password : *********	Mounting share in runtime
umount /samba/smb1	Unmounting share
vim /etc/fstab //ipaserver.example.com/samba /samba/smb1 cifs _netdev,username=smb1,password=password 0 0 :wq	Making entry in **fstab** file for persistent mount
mount -a	Mounting through **fstab**
mount	Verifying mounted share