

**WELCOME TO:
MODULE 8**

**DISK MANAGEMENT AND
RUN LEVELS**

System Run Level

- System Run Levels

Main Run level

- 0 Shut down (or halt) the system
- 1 Single-user mode; usually aliased as s or S
- 6 Reboot the system

Other Run levels

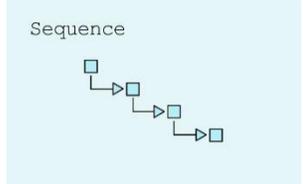
- 2 Multiuser mode without networking
- 3 Multiuser mode with networking
- 5 Multiuser mode with networking and GUI.

Linux Boot Process

BIOS	Basic Input/Output System executes MBR
MBR	Master Boot Record executes GRUB
GRUB	Grand Unified Bootloader executes Kernel
Kernel	Kernel executes /sbin/init
Init	Init executes runlevel programs
Runlevel	Runlevel programs are executed from /etc/rc.d/rc*.d/

Linux Boot Process (Newer Versions)

- The boot sequence changes in CentOS/Redhat 7 and above
- **systemd** is the new service manager in CentOS/RHEL 7 that manages the boot sequence
- It is backward compatible with SysV init scripts used by previous versions of RedHat Linux including RHEL 6
- Every system administrator needs to understand the boot process of an OS in order to troubleshoot effectively



Linux Boot Process (Newer Versions)

BIOS = Basic Input and Output Setting (firmware interface)

POST = Power-On Self-Test started

MBR = Master Boot Record

Information saved in the first sector of a hard disk that indicates where the GRUB2 is located so it can be loaded in computer RAM

GRUB2 = Grand Unified Boot Loader v2

Loads Linux kernel

/boot/grub2/grub.cfg

Kernel = Core of Operating System

Loads required drivers from initrd.img

Starts the first OS process (systemd)

Systemd = System Daemon (PID # 1)

It then starts all the required processes

Reads = /etc/systemd/system/default.target to bring the system to the run-level

Total of 7 run-levels (0 thru 6)

Message of the Day

- Message of the day file location
 - `/etc/motd`

Customize Message of the Day

- Once again, message of the day is the first message users will see when they login to the Linux machine
- Steps:
 - Create a new file in `/etc/profile.d/motd.sh`
 - Add desired commands in motd.sh file
 - Modify the `/etc/ssh/sshd_config` file to edit `#PrintMotd yes` to `PrintMotd no`
 - Restart sshd service
 - `systemctl restart sshd.service`



Disk Partition

- Commands for disk partition
 - `df`
 - `fdisk`

Adding Disk and Creating Partition

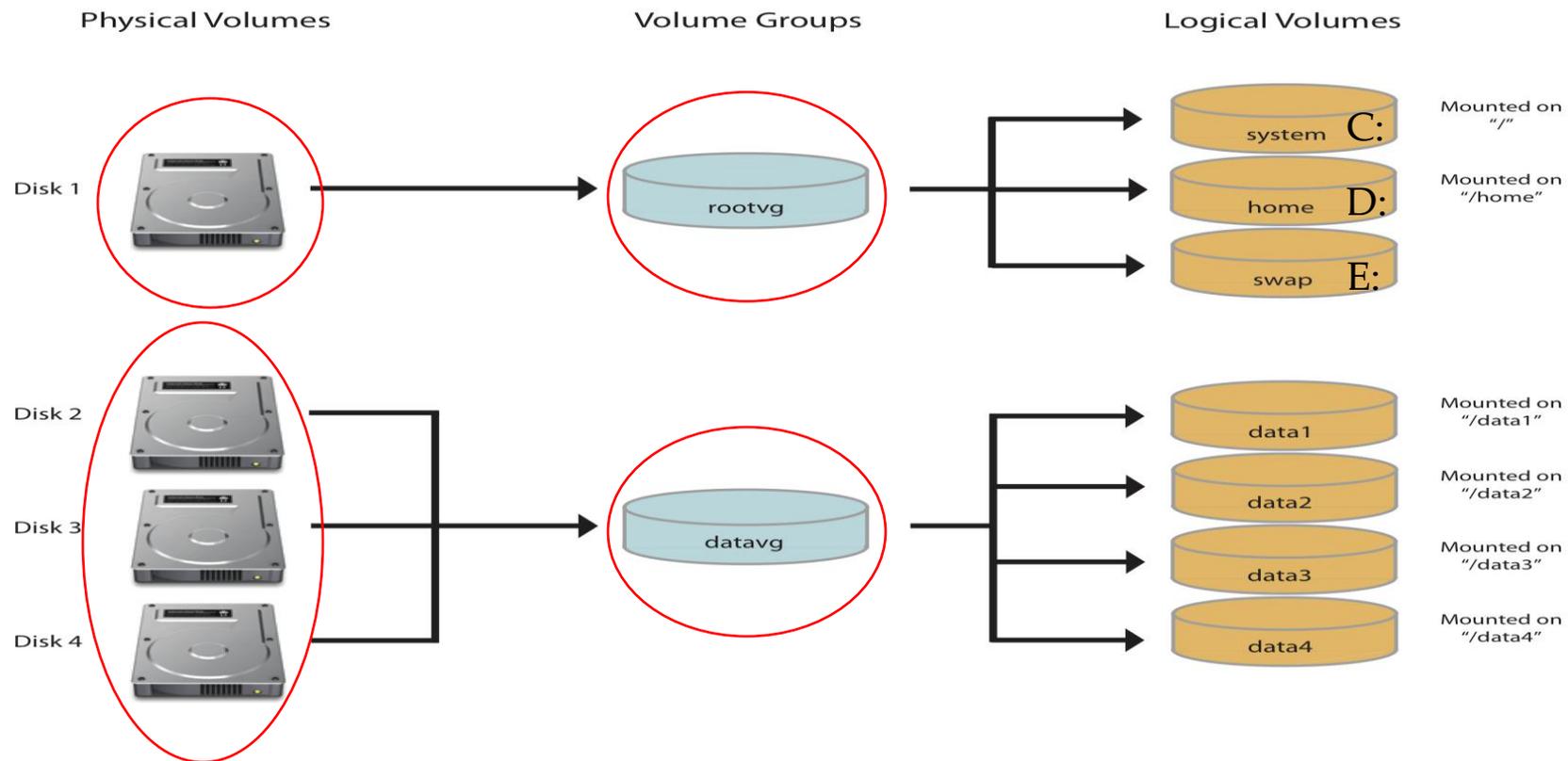
- Purpose? = Out of Space, Additional Apps etc.
- Commands for disk partition
 - `df`
 - `fdisk`

Computer Storage

- Local Storage
 - RAM, HDD, SSD, etc.
- DAS (Direct Attached Storage)
 - CD/DVD, USB flash drive, external disk directly attached with USB or other cables
- SAN (Storage Area Network)
 - Storage attached through iSCSI or fiber cable
- NAS (Network Attached Storage)
 - Storage attached over network (TCP/IP)
 - E.g. Samba, NFS etc.

Logical Volume Management (LVM)

- LVM allows disks to be combined together

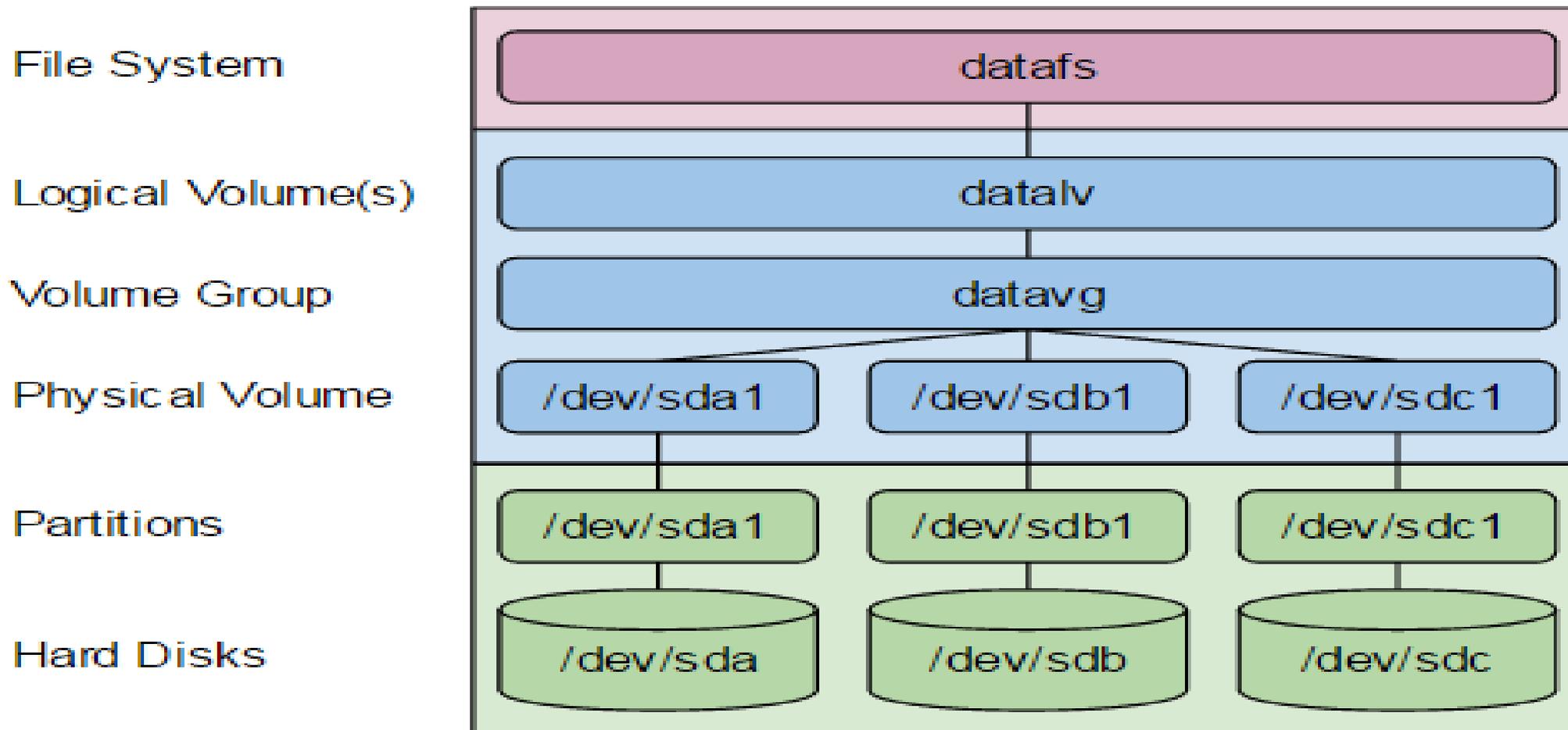


LVM Configuration During Install

- Install Linux CentOS with LVM configuration



Add Disk and Create LVM Partition



ADD AND EXTEND DISK USING LVM

/oracle = 1.0G

/oracle = Full

Few Options:

- Delete older files to free up disk space
- Add new physical disk mount to /oracle2
- Create a new virtual disk and mount to /oracle2
- Or extend /oracle through LVM.

ADD/EXTEND SWAP SPACE

- What is swap? – CentOS.org

Swap space in Linux is used when the amount of physical memory (RAM) is full. If the system needs more memory resources and the RAM is full, inactive pages in memory are moved to the swap space. While swap space can help machines with a small amount of RAM, it should not be considered a replacement for more RAM. Swap space is located on hard drives, which have a slower access time than physical memory

- Recommended swap size = Twice the size of RAM

M = Amount of RAM in GB, and S = Amount of swap in GB, then

If $M < 2$
then $S = M * 2$
Else $S = M + 2$

- Commands

- dd
- mkswap
- swapon or swapoff

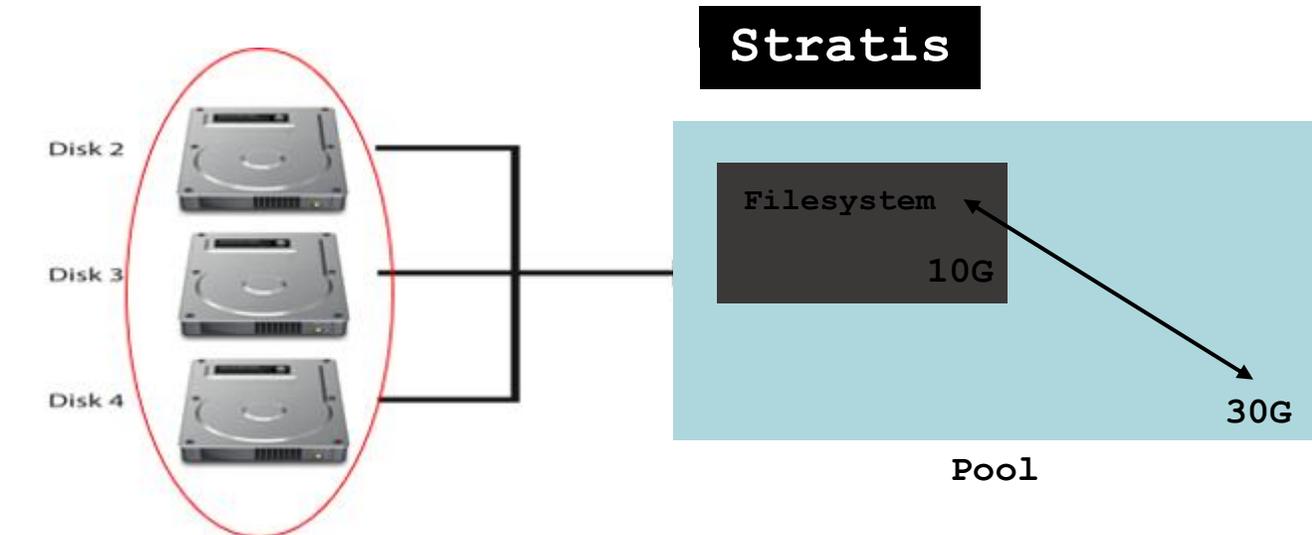
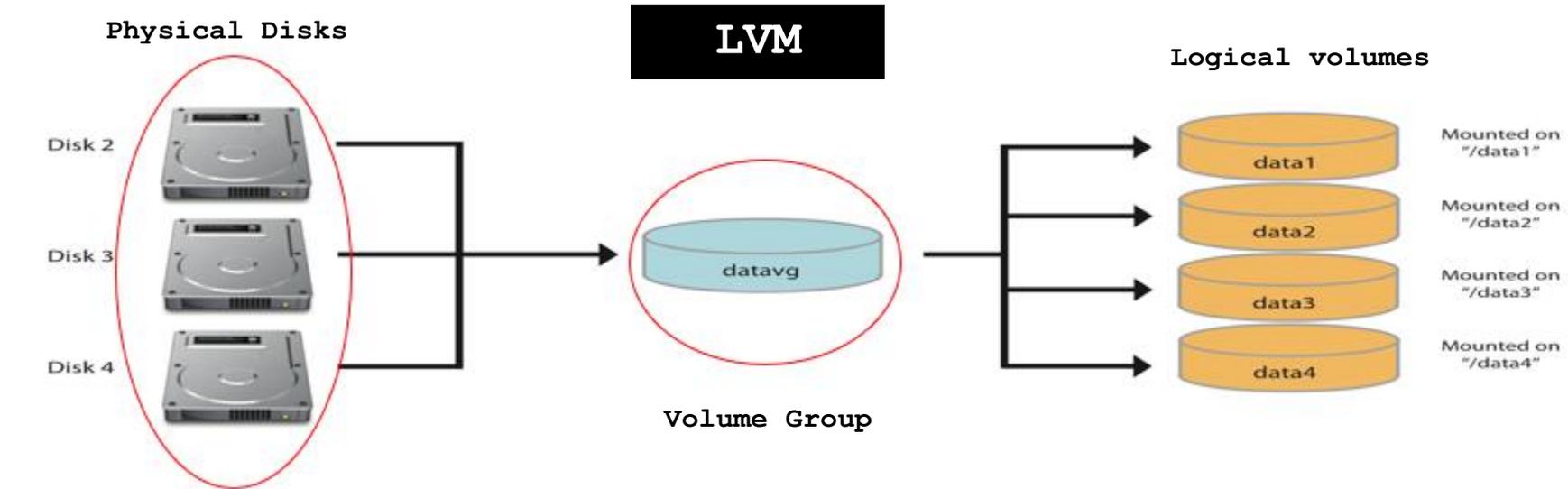
Implement Advanced Storage Features

- Red Hat 8 introduces the next generation volume management solution called Stratis
- It uses thin provisioning by default
- It combines the process of creating logical volume management (LVM) and creation of filesystems into one management
- In LVM if a filesystem system gets full you will have to extend it manually whereas stratis extends the filesystem automatically if it has available space in its pool

In this lesson we will learn...

- How to manage multiple storage layers using Stratis local storage management

Implement Advanced Storage Features



Implement Advanced Storage Features

- Install Stratis package

```
yum/dnf install stratis-cli stratisd
```

- Enable and start Stratis service

```
systemctl enable|start stratisd
```

- Add 2 x 5G new disks from virtualization software and verify at the OS level

```
Oracle virtualbox storage setting  
lsblk
```

- Create a new stratis pool and verify

```
stratis pool create pool1 /dev/sdb  
stratis pool list
```

- Extend the pool

```
stratis pool add-data pool1 /dev/sdc  
stratis pool list
```

Implement Advanced Storage Features

- Create a new filesystem using stratis

```
stratis filesystem create pool1 fs1
```

```
stratis filesystem list           (Filesystem will start with 546 MB)
```

- Create a directory for mount point and mount filesystem

```
mkdir /bigdata
```

```
mount /dev/stratis/pool1/fs1 /bigdata
```

```
lsblk
```

- Create a snapshot of your filesystem

```
stratis filesystem snapshot pool1 fs1 fs1-snap
```

```
stratis filesystem list
```

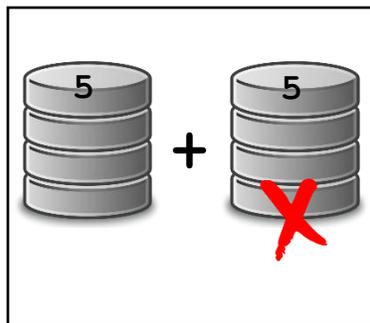
- Add the entry to /etc/fstab to mount at boot

```
UUID="asf-0887afgdja-" /bigdata xfs defaults,x-
```

```
systemd.requires=stratisd.service 0 0
```

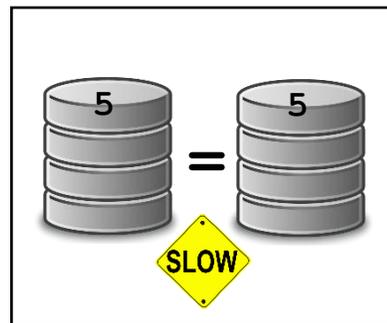
RAID

- RAID (Redundant Array of Independent Disks)
- Type of RAID
 - RAID0
 - RAID1
 - RAID5



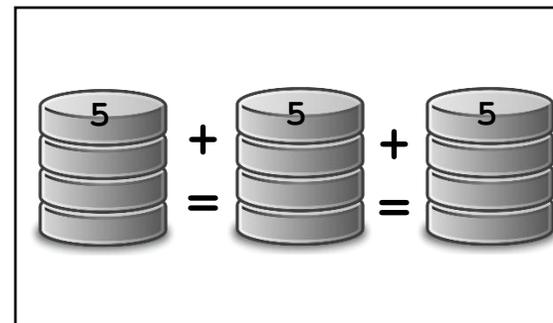
RAID0

5+5=10G



RAID1

5, 5=5G



RAID5

5+5+5=12G

File System Check (fsck and xfs_repair)



- Linux fsck utility is used to check and repair Linux filesystems (ext2, ext3, ext4, etc.)
- Linux xfs_repair utility is used to check and repair Linux filesystems for xfs filesystem type
- Depending on when was the last time a file system was checked, the system runs the fsck during boot time to check whether the filesystem is in consistent state
- System administrator could also run it manually when there is a problem with the filesystems
- Make sure to execute the fsck on an **unmounted** file systems to avoid any data corruption issues.

File System Check (fsck and xfs_repair)...



- Force a filesystem check even if it's clean using option `-f`
- Attempt to fix detected problems automatically using option `-y`
- The `xfs_repair` utility is highly scalable and is designed to repair even very large file systems with many inodes efficiently. Unlike other Linux file systems, `xfs_repair` does not run at boot time
- The following are the possible **exit codes** for `fsck` command
 - 0 – No errors
 - 1 – Filesystem errors corrected
 - 2 – System should be rebooted
 - 4 – Filesystem errors left uncorrected
 - 8 – Operational error
 - 16 – Usage or syntax error
 - 32 – Fsck canceled by user request
 - 128 – Shared-library error

System Backup (dd Command)

5 Different Types of Backups

1. System backup (entire image using tools such as acronis, Veeam, Commvault etc.)
 2. Application backup (3rd party application backup solution)
 3. Database backup (Oracle dataguard, SQL backup etc.)
 4. Filesystem backup (tar, gzip directoris etc.)
 5. Disk backup or disk cloning (dd command)
- dd is a command-line utility for Unix and Unix-like operating systems whose primary purpose is to convert and copy files
 - As a result, dd can be used for tasks such as backing up the boot sector of a hard drive, and obtaining a fixed amount of random data
 - Please note the source and destination disk should be the same size

System Backup (dd Command)...

- To backup or clone an entire hard disk to another hard disk connected to the same system, execute the dd command as shown

```
# dd if=<source file name> of=<target file name> [Options]
```

```
# dd if=/dev/sda of=/dev/sdb
```

- To backup/copy the disk partition

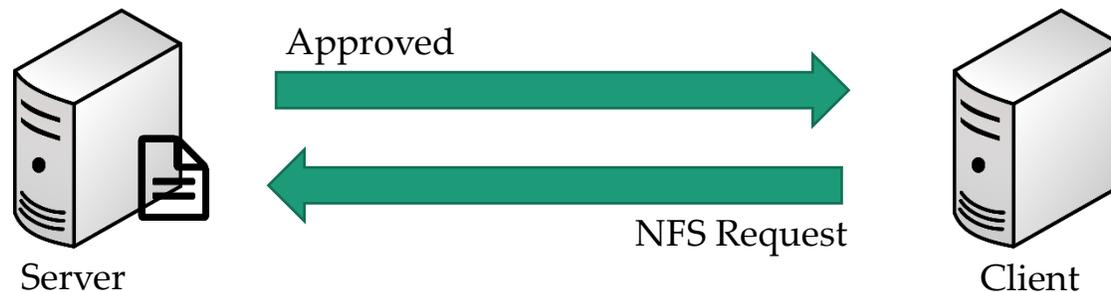
```
# dd if =/dev/sda1 of=/root/sda1.img
```

- Restoring this image file to other machine after copying the .img

```
# dd if=/root/sda1.img of=/dev/sdb3
```

Network File System (NFS)

- NFS stands for Network File System, a file system developed by Sun Microsystems, Inc.
- It is a client/server system that allows users to access files across a network and treat them as if they resided in a local file directory
- For example, if you were using a computer linked to a second computer via NFS, you could access files on the second computer as if they resided in a directory on the first computer. This is accomplished through the processes of exporting (the process by which an NFS server provides remote clients with access to its files) and mounting (the process by which client map NFS shared filesystem)



Network File System (NFS)...

Steps for NFS Server Configuration

- Install NFS packages

```
# yum install nfs-utils libnfsidmap (most likely they are installed)
```

- Once the packages are installed, enable and start NFS services

```
# systemctl enable rpcbind
```

```
# systemctl enable nfs-server
```

```
# systemctl start rpcbind, nfs-server, rpc-statd, nfs-idmapd
```

- Create NFS share directory and assign permissions

```
# mkdir /mypretzels
```

```
# chmod a+rx /mypretzels
```

- Modify `/etc/exports` file to add new shared filesystem

```
# /mypretzels 192.168.12.7(rw, sync, no_root_squash) = for only 1 host
```

Read/write

all changes to the according filesystem are immediately flushed to disk; the respective write operations are being waited for

```
# /mypretzels *(rw, sync, no_root_squash) = for everyone
```

- Export the NFS filesystem

```
# exportfs -rv
```

NFS share

IP address of client machine

root on the client machine will have the same level of access to the files on the system as root on the server.

Network File System (NFS)...

Steps for NFS Client Configuration

- Install NFS packages

```
# yum install nfs-utils rpcbind
```
- Once the packages are installed enable and start rpcbind service

```
# systemctl rpcbind start
```
- Make sure firewalld or iptables stopped (if running)

```
# ps -ef | egrep "firewall|iptables"
```
- Show mount from the NFS server

```
# showmount -e 192.168.1.5 (NFS Server IP)
```
- Create a mount point

```
# mkdir /mnt/kramer
```
- Mount the NFS filesystem

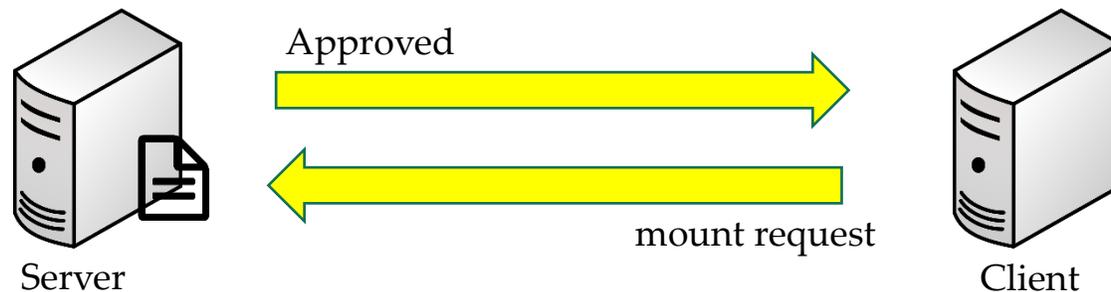
```
# mount 192.168.1.5:/mypretzels /mnt/kramer
```
- Verify mounted filesystem

```
# df -h
```
- To unmount

```
# umount /mnt/kramer
```

Samba

- Samba is a Linux tool or utility that allows sharing for Linux resources such as files and printers to with other operating systems
- It works exactly like NFS but the difference is NFS shares within Linux or Unix like system whereas Samba shares with other OS (e.g. Windows, MAC etc.)
- For example, computer “A” shares its filesystem with computer “B” using Samba then computer “B” will see that shared filesystem as if it is mounted as the local filesystem



Samba (smb vs. CIFS)

- Samba shares its filesystem through a protocol called **SMB** (Server Message Block) which was invented by IBM
- Another protocol used to share Samba is through **CIFS** (Common Internet File System) invented by Microsoft and NMB (NetBios Named Server)
- **CIFS** became the extension of **SMB** and now Microsoft has introduced newer version of SMB v2 and v3 that are mostly used in the industry
- In simple term, most people, when they use either SMB or CIFS, are talking about the same exact thing

Samba Installation and Configuration

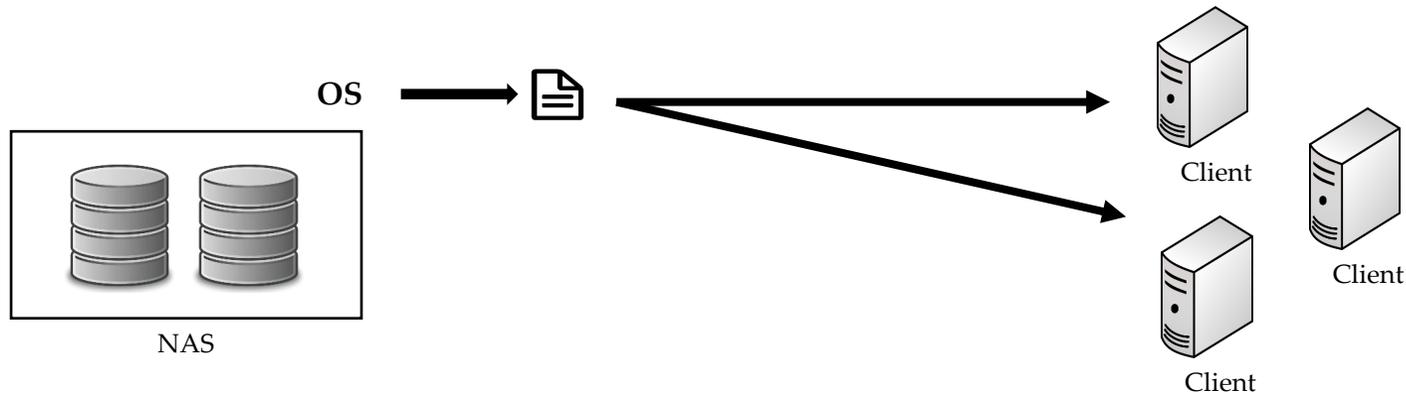
- Take snapshot of your VM
- Install samba packages
- Enable samba to be allowed through firewall (Only if you have firewall running)
- Disable firewall
- Create Samba share directory and assign permissions
- Also change the SELinux security context for the samba shared directory
- Or disable SELinux
- Modify `/etc/samba/smb.conf` file to add new shared filesystem
- Verify the setting
- Once the packages are installed, enable and start **Samba** services (smb and nmb)
- Mount Samba share on Windows client
- Mount Samba share on Linux client
- Additional instructions on creating secure Samba share.

NAS Device for NFS or Samba

- A storage can be carved on a Linux server, and it can be shared with another Linux machine through NFS or to a Windows machine through Samba service



- NFS/Samba or any NAS service can be setup through a dedicated NAS device

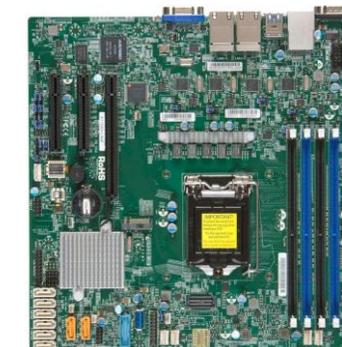
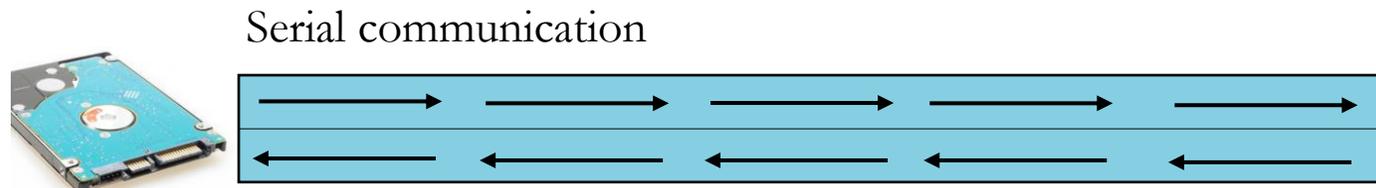
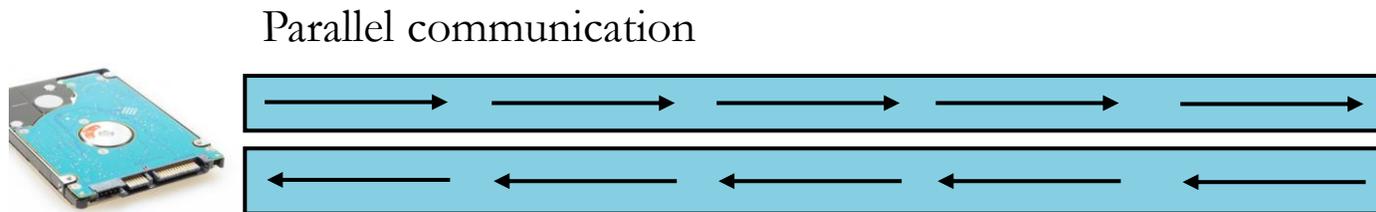


NAS Device for NFS or Samba

- In this video we will learn...
 - Physical layout of a NAS device
 - Setup, configure and manage NAS device
 - Create shared filesystem (NFS and Samba)
 - Mount shared folder from the NAS device to Linux and Windows

SATA and SAS

- **SATA** Stands for Serial Advanced Technology Attachment and **SAS** stands for Serial Attached SCSI (SCSI Stands for Small Computer System Interface, typically pronounced as “scuzzy”)
- Both SAS and SATA utilize serial communication. Serial communication means that the highway has both lanes



SATA and SAS

- The main difference between them is that SAS drives are faster and more reliable than SATA drives
- SAS is generally more expensive, and it's better suited for use in servers or in processing-heavy computer workstations. SATA is less expensive, and it's better suited for desktop file storage
- In a SATA cable, all 4 wires are placed within the same cable. In a SAS cable, the 4 wires are separated into 2 different cables

Why divide the wires between 2 cables???

- So you can connect more devices to one another. With a SATA cable, you can only link the motherboard and the storage drive. You could hook up an expansion device, but that takes up valuable room inside your computer.
- With a SAS cable, you can hook up the motherboard to both a storage drive and another piece of hardware that has SAS connectors.

Here's what the highways look like, metaphorically:

- **SATA cable:** Los Angeles to San Francisco
- **SAS cable:** Los Angeles to San Francisco or Los Angeles to Las Vegas

Difference Between CentOS/RHEL 7 and 8

- Red Hat Enterprise Linux 8 (RHEL 8) is now available for production use with lots of developer-friendly capabilities
- RHEL 8 official release by Red Hat Inc, was announced on **May 7, 2019**
- I will cover only what is changed and what you should in terms of my Linux course

	RHEL 8	RHEL 7
General Availability Date	14-Nov-18	10-Jun-14
Code Name	Ootpa	Maipo
Kernel Version	4.18	3.10.0-123
End of Support	May-2029	30-Jun-2024
Last Minor Release	8.x	7.7
Network Time Synchronization	Only Chrony	Chrony and ntpd
GUI Interface (Desktop)	Gnome 3.28	Gnome 3
Default Database	MySQL 8.0, MariaDB 10.3, PostgreSQL 10 and 9.6, and Redis 5.0	MariaDB

Difference Between CentOS/RHEL 7 and 8

	RHEL 8	RHEL 7
Default Firewall	Firewalld, it uses nftables framework in the backend	Firewalld, it uses Iptables framework in the backend
Max Supported (Individual) File & Filesystem Size	XFS= 1024TB	XFS= 500TB
Package Management	By default both are installed, YUM symbolic link to DNF	By default only YUM and DNF can be installed from the Extra repo
Max. RAM Supported	24 TB on x86_64 architecture	12 TB on x86_64 architecture