Server Administration Basics

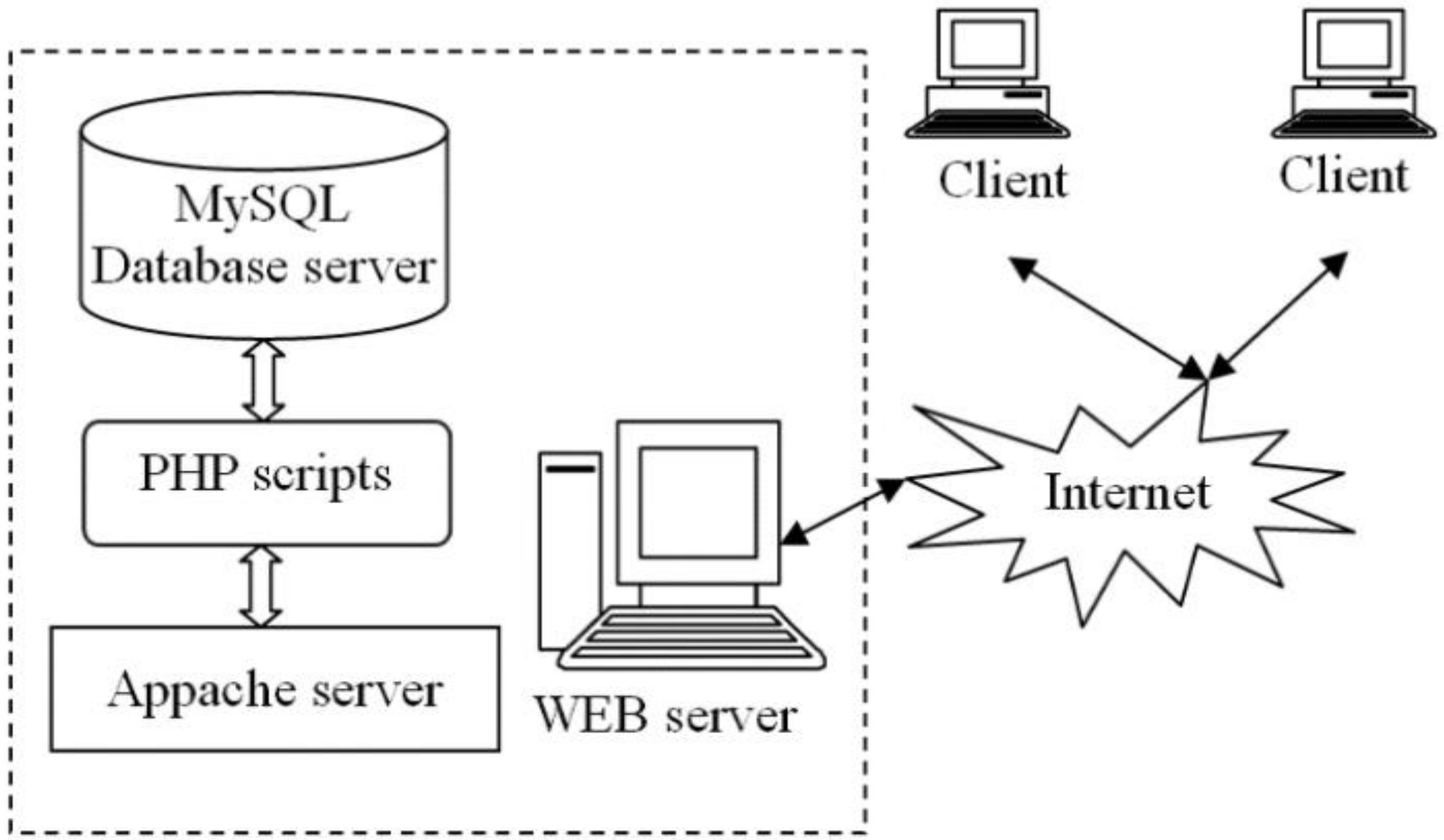
Compiled by: Hiranya Prasad Bastakoti

Contents

Open Source Server and Client Installation

- In a client-server architecture, clients request services from a server. Servers provide the requested service to their clients.
- However, clients and servers are separate programs running on different machines.
- Clients act as the front end, and servers remain at the back end.
- The responsibility of delivering data to the user lies with the clients, whereas the servers are responsible for the storage and management of data.

Client Server Architecture



Client

- A **client** is a piece of software that runs on a device and gives the user access to a server. The server provides the requested services upon request from the customer.
- A web browser, like Google Chrome or Mozilla Firefox, is a customer that asks a web server for webpages.

Server

- A **server** is a piece of software that runs on a device and makes services available to other devices which is connected to a network.
- It receives customer requests, and handles them, also provides the requested data or issues.
- A web server is a piece of software which responds to the customer requests by storing and delivering web pages and other sites via the internet.
- Example:**Apache HTTP Server, Nginx Web Server, Lighttpd Web Server, Caddy Web Server**

Advantages of Client/Server Architecture

- The data is centralized within the system that is maintained in a single place.
- The model is efficient in delivering resources to the client and also requires low-cost maintenance.
- It is easy to manage, and the data can be easily delivered to the client.
- As the data is centralized, this system is more secure and serves added security to the data.
- Within this type of model, more clients and servers can be embedded into the server, which makes the performance outstanding and increases the model’s overall flexibility.

Open source System/Software

- Open source development is an approach to system/software development in which the source code of a software system is published and volunteers are invited to participate in the development process
- Open Source software is software that can be freely accessed, used, changed, and shared (in modified or unmodified form) by anyone.
- The best-known open source product is the Linux operating system which is widely used as a server system and, increasingly, as a desktop environment.
- Other important open source products are Java, the Apache web server and the mySQL database management system

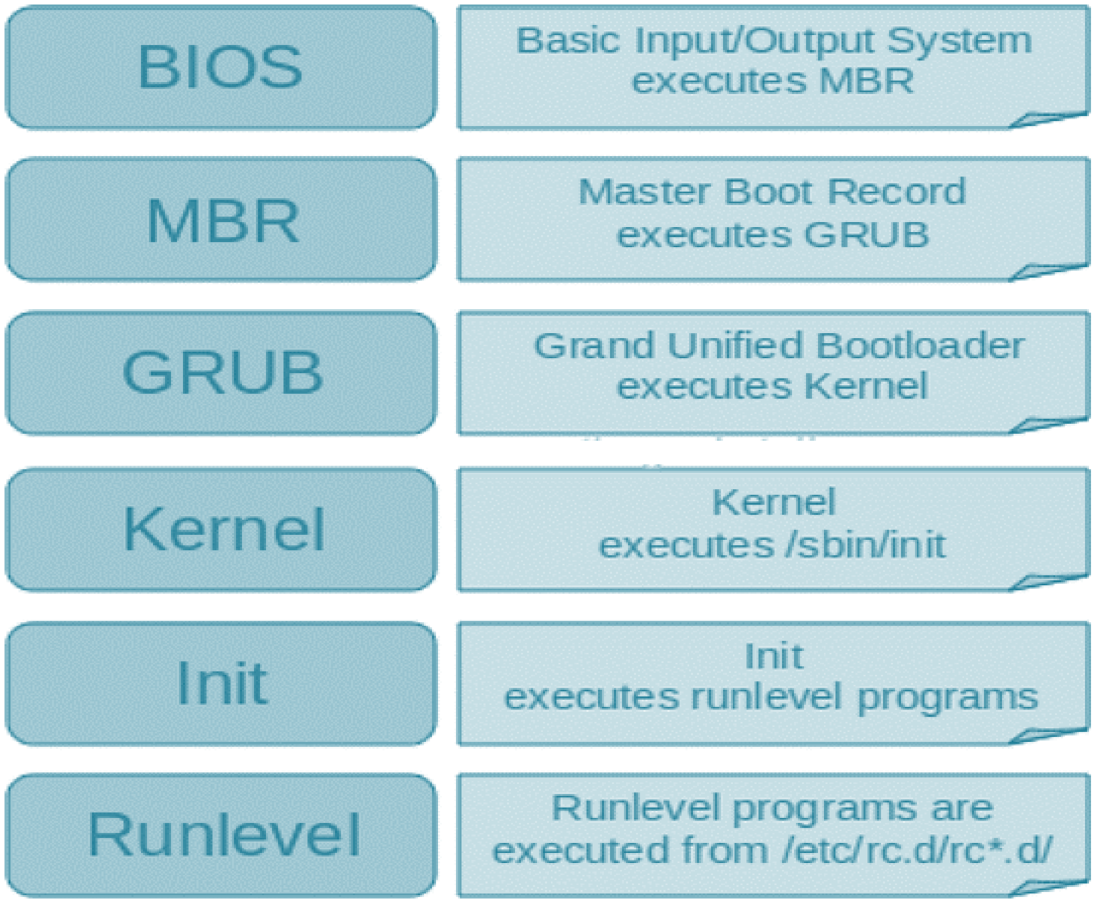
Linux Booting Process

- **Booting is a process of switching on the computer and starting the operating system.**
- **Press the power button on your system, and after few moments you see the Linux login prompt**

Six steps of the booting process are:

- **BIOS and Setup Program,**
- **The Power- On-Self-Test (POST),**
- **The Operating system Loads,**
- **System Configuration**
- **System Utility Loads**
- **Users Authentication.**

The six high level stages of a typical 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/

Contd..

BIOS

- BIOS stands for Basic Input / Output System.
- Performs some system integrity checks.
- Searches, loads, and executes the boot loader program.
- It looks for boot loader in floppy, cd-rom, or hard drive. You can press a key (typically F12 of F2, but it depends on your system) during the BIOS startup to change the boot sequence.
- Once the boot loader program is detected and loaded into the memory, BIOS gives the control to it.
- So, in simple terms BIOS loads and executes the MBR boot loader.

Contd..

MBR

- MBR stands for Master Boot Record.
- It is located in the 1st sector of the bootable disk. Typically /dev/hda, or /dev/sda
- MBR is less than 512 bytes in size. This has three components 1) primary boot loader info in 1st 446 bytes 2) partition table info in next 64 bytes 3) mbr validation check in last 2 bytes.
- It contains information about GRUB (or LILO in old systems).
- So, in simple terms MBR loads and executes the GRUB boot loader.

Contd..

GRUB

- GRUB stands for Grand Unified Bootloader.
- If you have multiple kernel images installed on your system, you can choose which one to be executed.
- GRUB displays a splash screen, waits for few seconds, if you do not enter anything, it loads the default kernel image as specified in the grub configuration file.
- GRUB has the knowledge of the filesystem (the older Linux loader LILO didn’t understand filesystem).
- Grub configuration file is /boot/grub/grub.conf (/etc/grub.conf is a link to this). The following is a sample grub.conf of CentOS.

Kernel

- A kernel is the lowest level of software that can interface with computer hardware
- Mounts the root file system as specified in the “root=” in grub.conf
- Kernel executes the /sbin/init program.
- Since init was the 1st program to be executed by Linux Kernel, it has the process id (PID) of 1. Do a ‘ps -ef | grep init’ and check the pid.
- initrd stands for Initial RAM Disk.
- initrd is used by kernel as temporary root file system until kernel is booted and the real root file system is mounted. It

- **Init**

- Looks at the /etc/inittab file to decide the Linux run level.
- Following are the available run levels
  - 0 – halt
  - 1 – Single user mode
  - 2 – Multiuser, without NFS
  - 3 – Full multiuser mode
  - 4 – unused
  - 5 – X11
  - 6 – reboot
- Init identifies the default initlevel from /etc/inittab and uses that to load all appropriate program.
- Execute ‘grep initdefault /etc/inittab’ on your system to identify the default run level
- If you want to get into trouble, you can set the default run level to 0 or 6. Since you know what 0 and 6 means, probably you might not do that.
- Typically you would set the default run level to either 3 or 5

Runlevel programs

- When the Linux system is booting up, you might see various services getting started.
  For example, it might say "starting sendmail …. OK". Those are the runlevel programs, executed from the run level directory as defined by your run level.

- Depending on your default init level setting, the system will execute the programs from one of the following directories.

  Run level 0 – /etc/rc.d/rc0.d/
  Run level 1 – /etc/rc.d/rc1.d/
  Run level 2 – /etc/rc.d/rc2.d/
  Run level 3 – /etc/rc.d/rc3.d/
  Run level 4 – /etc/rc.d/rc4.d/
  Run level 5 – /etc/rc.d/rc5.d/
  Run level 6 – /etc/rc.d/rc6.d/

Runlevel

- Please note that there are also symbolic links available for these directory under /etc directly. So, /etc/rc0.d is linked to /etc/rc.d/rc0.d.
- Under the /etc/rc.d/rc*.d/ directories, you would see programs that start with S and K.
- Programs starts with S are used during startup. S for startup.
- Programs starts with K are used during shutdown. K for kill.
- There are numbers right next to S and K in the program names. Those are the sequence number in which the programs should be started or killed.

Contd.

Init

- Looks at the /etc/inittab file to decide the Linux run level.
- Following are the available run levels
  - 0 – halt
  - 1 – Single user mode
  - 2 – Multiuser, without NFS
  - 3 – Full multiuser mode
  - 4 – unused
  - 5 – X11
  - 6 – reboot
- Init identifies the default initlevel from /etc/inittab and uses that to load all appr program.
- Execute 'grep initdefault /etc/inittab' on your system to identify the default run leve
- If you want to get into trouble, you can set the default run level to 0 or 6. Sin know what 0 and 6 means, probably you might not do that.
- Typically you would set the default run level to either 3 or 5.

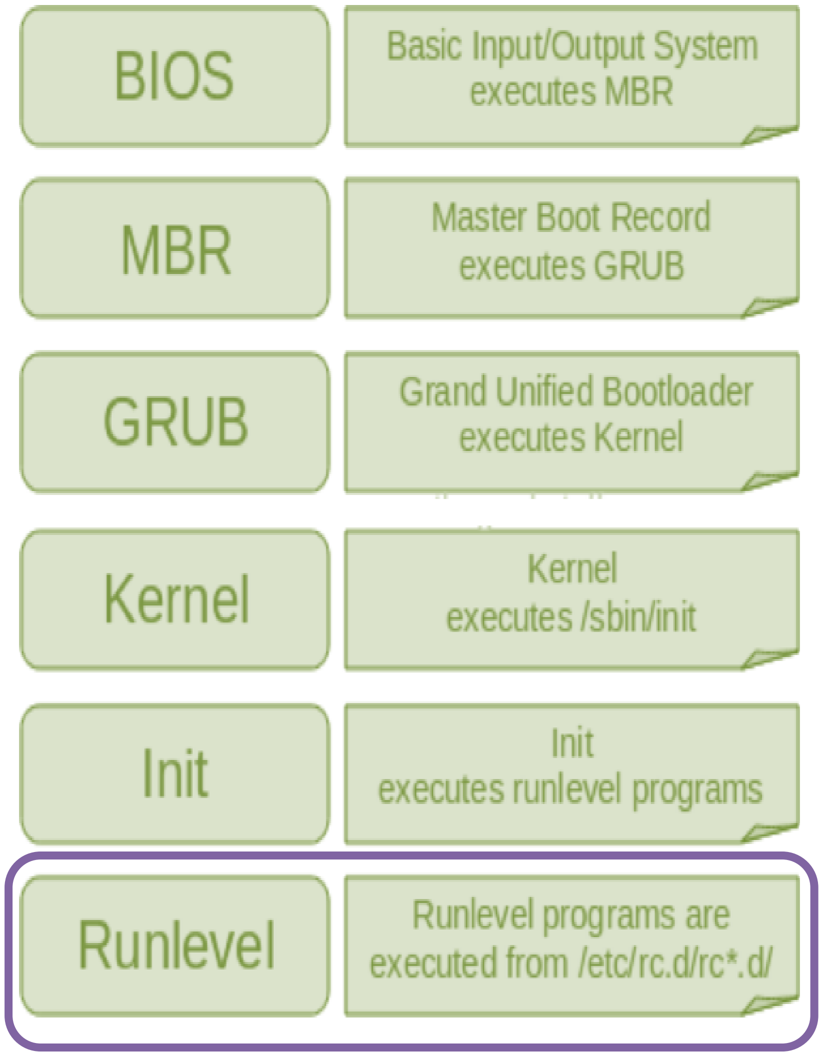
Runlevel programs

- When the Linux system is booting up, you might see various services getting started.

  For example, it might say "starting sendmail …. OK". Those are the runlevel programs, executed from the run level directory as defined by your run level.

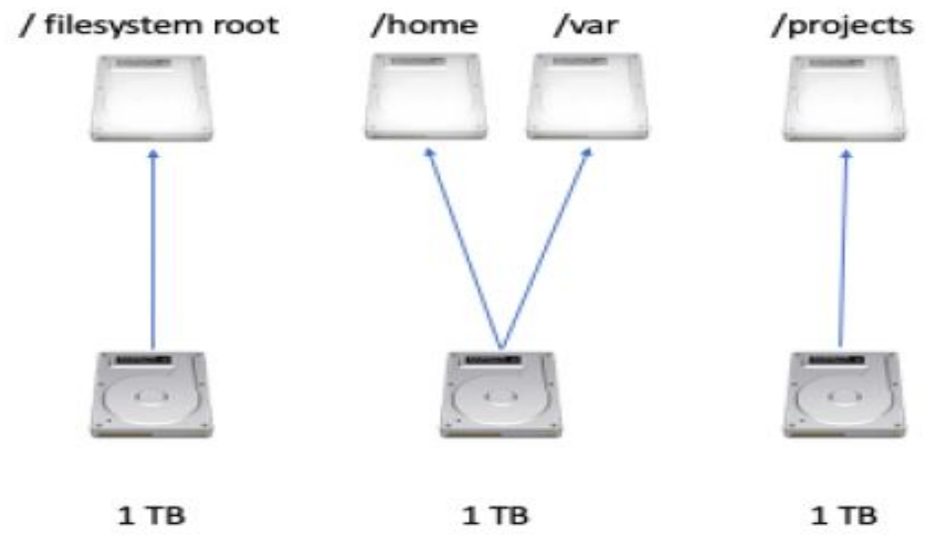
- Depending on your default init level setting, the system will execute the programs from one of the following directories.

  Run level 0 – /etc/rc.d/rc0.d/
  Run level 1 – /etc/rc.d/rc1.d/
  Run level 2 – /etc/rc.d/rc2.d/
  Run level 3 – /etc/rc.d/rc3.d/
  Run level 4 – /etc/rc.d/rc4.d/
  Run level 5 – /etc/rc.d/rc5.d/
  Run level 6 – /etc/rc.d/rc6.d/


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/

Traditional storage management

- Traditional storage capacity is based on individual disk capacity.
- Storage space is typically managed based on the maximum capacity of individual hard disk drives.



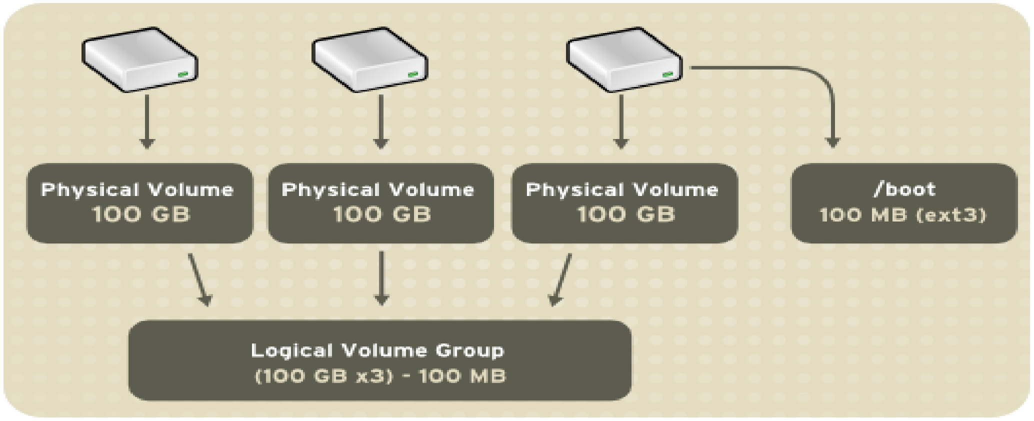
Three 1 TB hard drives with partitions and mount points. The partitions are entirely contained on the individual hard disk drives.

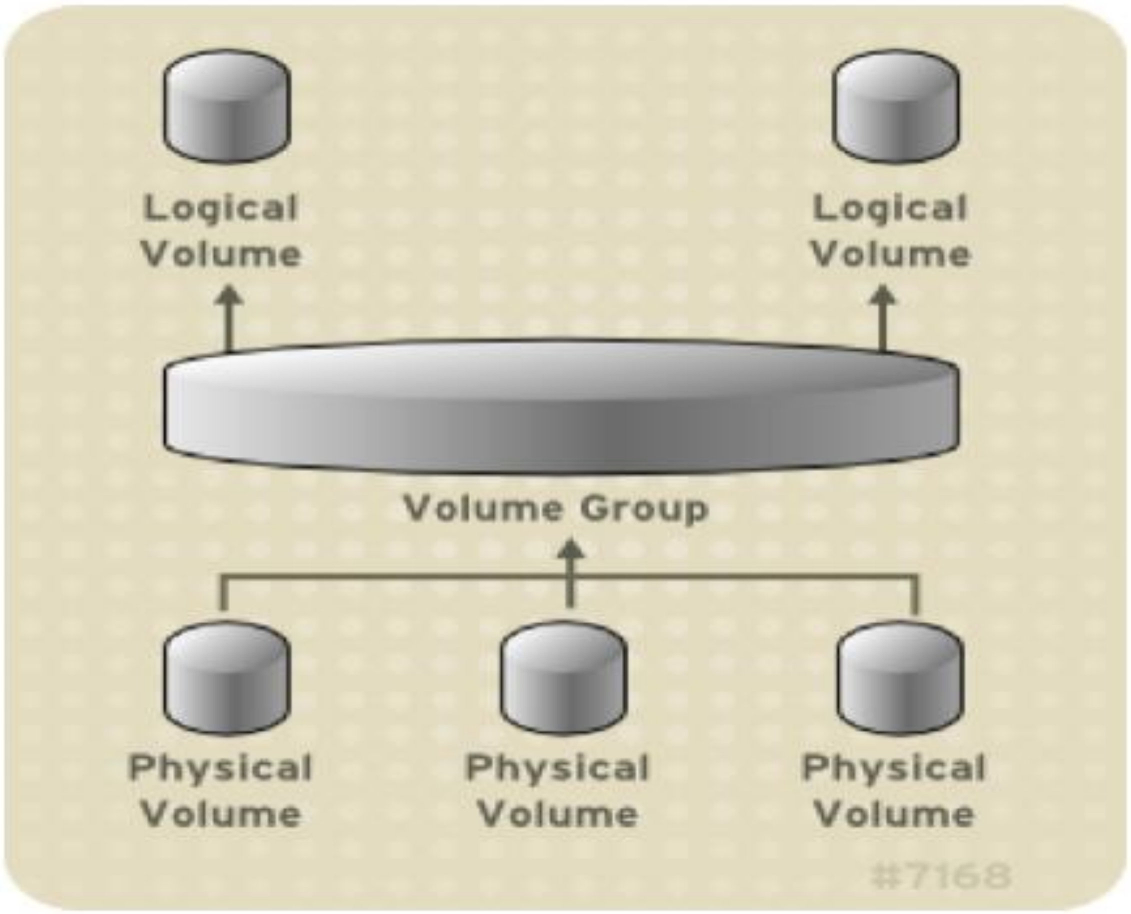
Logical Volume Manager (LVM)

- LVM is a tool for logical volume management which includes allocating disks, striping, mirroring and resizing logical volumes.
- With LVM, a hard drive or set of hard drives is allocated to one or more *physical volumes*.
- LVM physical volumes can be placed on other block devices which might span two or more disks.
- The physical volumes are combined into *logical volumes*, with the exception of the /boot partition.
- The /boot partition cannot be on a logical volume group because the boot loader cannot read it. If the root (/) partition is on a logical volume, create a separate /boot partition which is not a part of a volume group.

Contd..

- With LVM, those same three disks are considered to be 3 TB of aggregated storage capacity.
- This is accomplished by designating the storage disks as *Physical Volumes* (PV), or storage capacity useable by LVM.





Three hard disk drives are combined into one volume group that is then carved into two logical volumes.

Startup Services(System Daemon): Xinetd/Inetd

- A daemon is a computer program that is usually initiated at the operating system startup and runs as a background process.
- The purpose of the daemon is to respond to network, hardware, or system requests to perform certain tasks.
- Unix-like systems typically run numerous daemons, mainly to accommodate requests for services from other computers on a network, but also to respond to other programs and to hardware activity.

Contd..

- Daemons are usually instantiated as processes. A process is an executing (i.e., running) instance of a program.
- Processes are managed by the kernel (i.e., the core of the operating system), which assigns each a unique process identification number (PID).
- There are three basic types of processes in Linux: interactive, batch and daemon

Inted/Xinetd

- Many daemons are now started only as required and by a single daemon, xinetd (which has replaced inetd in newer systems), rather than running continuously. xinetd, which is referred to as a TCP/IP super server, itself is started at boot time, and it listens to the ports assigned to the processes listed in the /etc/inetd.conf or in /etc/xinetd.conf configuration file.
- Examples of daemons that it starts include crond (which runs scheduled tasks), ftpd (file transfer), lpd (laser printing), rlogind (remote login), rshd (remote command execution) and telnetd (telnet)..

Inted

- Inetd (internet service daemon) is a super-server daemon on many Unix systems that provides Internet services.
- For each configured service, it listens for requests from connecting clients.
- Requests are served by spawning a process which runs the appropriate executable, but simple services such as echo are served by inetd itself.
- External executables, which are run on request, can be single- or multi-threaded. First appearing in 4.3BSD, it is generally located at /usr/sbin/inetd.

- Often called a super-server, inetd listens on designated ports used by Internet services such as FTP, POP3, and telnet. When a TCP packet or UDP packet arrives with a particular destination port number, inetd launches the appropriate server program to handle the connection
- The inetd daemon starts by default each time you start your system. When the daemon starts, it reads its configuration information from the file specified in the ConfigurationFile parameter. If the parameter is not specified, the inetd daemon reads its configuration information from the /etc/inetd.conf file.
- The ftpd, rlogind, rexecd, rshd, talkd, telnetd, and uucpd daemons are started by default by inetd.

Xinetd(Extended Internet Service Daemon)

- **Xinetd** is an open-source daemon that manages the Internet connections on many modern Unix and Unix-like servers.
- It offers improved security over the traditional inetd daemon, and has replaced inetd in most modern versions of Linux, BSD, and macOS.
- Xinetd performs a similar function to its predecessor.
- It listens for incoming network traffic and launches the appropriate service, such as FTP, HTTP, or UDP, to handle the traffic.
- xinetd runs constantly and listens on all ports for the services it manages. When a connection request arrives for one of its managed services, xinetd starts up the appropriate server for that service.
- The configuration file for xinetd is /etc/xinetd.conf, but the file only contains a few defaults and an instruction to include the /etc/xinetd.d directory.

Managing accounts: users, groups and other privileges

- User management includes everything from creating a user to deleting a user on your system. User management can be done in three ways on a Linux system.
- **Graphical tools** are easy and suitable for new users.
- **Command line tools** includes commands like useradd, userdel, passwd, etc. These are mostly used by the server administrators.
- Third and very rare tool is to **edit the local configuration files** directly using vi.
- *The local user database in Linux is: /etc/passwd directory.*

:

1. Use Account info :/etc/passwd file:

- User account information is stored in the /etc/passwd file.
- This information includes the account name, home directory location, and default shell, among other values

2.Create, modify, and delete user accounts

- The process for managing user accounts is very straightforward. Sysadmins either add, modify, or delete users, and the related commands:
- useradd
- usermod
- userdel

3. Manage password requirements

- Two common ways of managing password settings are using the /etc/login.defs file or Pluggable Authentication Module (PAM) setting.

4. Managing groups:

- It's more efficient to group user accounts with similar access requirements than to manage permissions on a user-by-user basis.
- Therefore, sysadmins need to be comfortable with the process of creating, modifying, and deleting groups.

i. /etc/group file

- /etc/group file contains group account information.
- This information can be essential for troubleshooting, security audits, and ensuring users can access the resources they need

Ii. Create, modify, and delete groups:

- groupadd
- groupmod
- groupdel

iii. Manage group membership

- Adding users to a group simplifies permissions management.
- Adding a user to a group modifies the user, not the group. Therefore, the necessary command is the usermod command.

Some commands to display group information:

- usermod: Update group membership
- id: Display a list of groups the user is a member of
- cat /etc/group: Show a list of existing groups, with membership displayed in the last field

File System

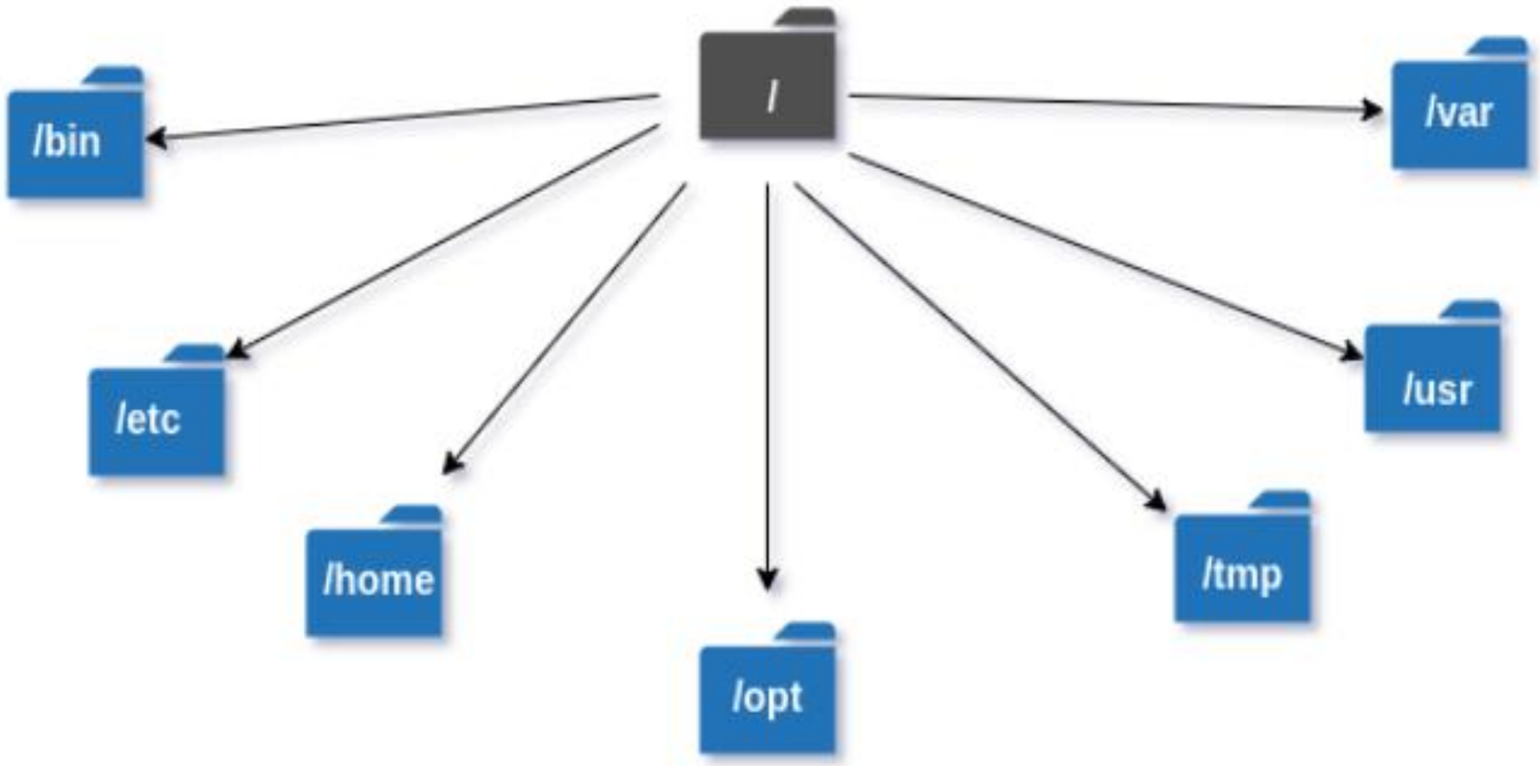
- The file system structure is the most basic level of organization in an operating system.
- The way an operating system interacts with its users, applications, and security model nearly always depends on how the operating system organizes files on storage devices.
- Providing a common file system structure ensures users and programs can access and write files.

- File systems break files down into two logical categories:
- Shareable versus unshareable files
- Variable versus static files
- *Shareable* files can be accessed locally and by remote hosts
- *unshareable* files are only available locally
- *Variable* files, such as log files, can be changed at any time;
- *Static* files, such as binaries, do not change without an action from the system administrator.

Types of File

- **General Files** – It is also called ordinary files. It may be an image, video, program, or simple text files. These types of files can be in ASCII or Binary format. It is the most commonly used file in the Linux system.
- **Directory Files** – These types of files are a warehouse for other file types. It may be a directory file within a directory (subdirectory).
- **Device Files –** In a Windows-like operating system, devices like CD-ROM, and hard drives are represented as drive letters like F: G: H whereas in the Linux system device are represented as files. As for example, /dev/sda1, /dev/sda2 and so on.

Tree-like structure starting with the root directory



Common top-level directories associated with the root directory:

- **/bin** – binary or executable programs.
- **/etc** – system configuration files.
- **/home** – home directory. It is the default current directory.
- **/opt** – optional or third-party software.
- **/tmp** – temporary space, typically cleared on reboot.
- **/usr** – User related programs.
- **/var** – log files.

Others directory

- **/boot-** It contains all the boot-related information files and folders such as conf, grub, etc.
- **/dev –** It is the location of the device files such as dev/sda1, dev/sda2, etc.
- **/lib –** It contains kernel modules and a shared library.
- **/lost+found –** It is used to find recovered bits of corrupted files.
- **/media –** It contains subdirectories where removal media devices inserted.
- **/mnt –** It contains temporary mount directories for mounting the file system.
- **/proc** – It is a virtual and pseudo-file system to contains info about the running processes with a specific process ID or PID.
- **/run –** It stores volatile runtime data.
- **/sbin –** binary executable programs for an administrator.
- **/srv –** It contains server-specific and server-related files.
- **/sys –** It is a virtual filesystem for modern Linux distributions to store and allows modification of the devices connected to the system.

Quota Management

- Disk space can be restricted by implementing disk quotas which alert a system administrator before a user consumes too much disk space or a partition becomes full.
- Disk quotas can be configured for individual users as well as user groups.
- A Quota is a built-in feature of the Linux kernel that is used to set a limit of how much disk space a user or a group can use.
- It is also used to limit the maximum number of files a user or a group can create on Linux. The file system where you want to use quota must also support quota.

Hard quota and Soft quota

- Each computer account has both a hard and a soft quota.
- The soft quota is the point at which you are warned that you are approaching your hard quota.
- The hard quota is the absolute maximum amount of disk space the system grants your account.

Usage quota and File quota

- There are two basic types of disk quotas.
- The first, known as a usage quota or block quota, limits the amount of disk space that can be used.
- The second, known as a file quota or inode quota, limits the number of files and directories that can be created.

Common Unix Disk Quota Utilities

- quota - display a user's file system disk quota and usage;
- edquota - Edit user quotas for file system;
- repquota - Summarize quotas for a file system\
- quotacheck - File system quota consistency checker;
- quotaon - Turn file system quotas on and off;
- /etc/fstab (Linux) or /etc/vfstab (Solaris) - list of default parameters for each file system including quota status.

Implementing disk quotas

- Enable quotas per file system by modifying the /etc/fstab file.
- Remount the file system(s).
- Create the quota database files and generate the disk usage table.
- Assign quota policies.

Job scheduling

- Job scheduling is the process of allocating system resources to many different tasks by an operating system (OS).
- The system handles prioritized job queues that are awaiting CPU time and it should determine which job to be taken from which queue and the amount of time to be allocated for the job.
- This type of scheduling makes sure that all jobs are carried out fairly and on time

Job Scheduling with cron

- Cron is a system that helps Linux users to schedule any task. However, a cron job is any defined task to run in a given time period.
- It can be a shell script or a simple bash command. Cron job helps us automate our routine tasks, it can be hourly, daily, monthly, etc.
- *It is a service that enables you to schedule running a task, often called a job, at regular times.*
- *A cron job is only executed if the system is running on the scheduled time.*
- Cron is a scheduling daemon that executes tasks at specified intervals. These tasks are called cron jobs and are mostly used to automate system maintenance or administration.

Cron job syntax

- **crontab -a <filename>**: create a new **<filename>** as crontab file
- **crontab -e**: edit our crontab file or create one if it doesn’t already exist
- **crontab -l**: show up our crontab file
- **crontab -r**: delete our crontab file
- **crontab -v**: show up the last time we have edited our crontab file
- /etc/cron.hourly/
- /etc/cron.daily/
- /etc/cron.weekly/
- /etc/cron.monthly/

Crontab

- Crontab (cron table) is a text file that specifies the schedule of cron jobs.
- There are two types of crontab files.
- The system-wide crontab files and individual user crontab files.
- Users' crontab files are named according to the user's name, and their location varies by operating systems.
- In Red Hat based distributions such as CentOS, crontab files are stored in the /var/spool/cron directory, while on Debian and Ubuntu files are stored in the /var/spool/cron/crontabs directory.

Crontab syntax

- crontab -e - Edit crontab file, or create one if it doesn’t already exist.
- crontab -l - Display crontab file contents.
- crontab -r - Remove your current crontab file.
- crontab -i - Remove your current crontab file with a prompt before removal.
- crontab -u <username> - Edit other user crontab file. This option requires system administrator privileges.

Anacron

- **Anacron** is used to run commands periodically with a frequency defined in days.
- It works a little different from **cron**; assumes that a machine will not be powered on all the time.
- It is appropriate for running daily, weekly, and monthly scheduled jobs normally run by cron, on machines that will not run 24-7 such as laptops and desktops machines.
- if you use **anacron**, you can be assured that the next time you power on the desktop/laptop again, the backup script will be executed.

- anacron jobs are listed in **/etc/anacrontab** and jobs can be scheduled using the format below (comments inside anacrontab file must start with **#**).

```
period  delay  job-identifier
```

- **period** – this is the frequency of job execution specified in days or as @daily, @weekly, or @monthly for once per day, week, or month. You can as well use numbers: 1 – daily, 7 – weekly, 30 – monthly and N – number of days.
- **delay** – it's the number of minutes to wait before executing a job.
- **job-id** – it's the distinctive name for the job written in log files.

Comparison(Cron vs Anacron)

Cron	Anacron
It’s a daemon	It’s not a daemon
Appropriate for server machines	Appropriate for desktop/laptop machines
Enables you to run scheduled jobs every minute	Only enables you to run scheduled jobs on daily basis
Doesn’t executed a scheduled job when the machine if off	If the machine if off when a scheduled job is due, it will execute a scheduled job when the machine is powered on the next time
Can be used by both normal users and root	Can only be used by root unless otherwise (enabled for normal users with specific configs)

Log Analysis

- **Log analysis** is a process that gives visibility into the performance and health of IT infrastructure and application stacks, through the review and interpretation of logs that are generated by network, operating systems, applications, servers, and other hardware and software components.
- Logs typically contain time-series data that is either streamed using collectors real-time or stored for review at a later time.
- Log analysis offers insight into system performance and can indicate possible problems such as security breaches or imminent hardware failure.

Log Analysis

- Linux has a special directory for storing logs called /var/log.
- This directory contains logs from the OS itself, services, and various applications running on the system.
- System log file analysis is one of the most important tasks when analyzing the system
- The system log files should be the first thing to do when maintaining or troubleshooting a system.
- Examples:
- /var/log/syslog and /var/log/messages store all global system activity data, including startup messages. Debian-based systems like Ubuntu store this in /var/log/syslog, while Red Hat-based systems like RHEL or CentOS use /var/log/messages.
- /var/log/auth.log and /var/log/secure store all security-related events such as logins, root user actions, and output from pluggable authentication modules (PAM). Ubuntu and Debian use /var/log/auth.log, while Red Hat and CentOS use /var/log/secure.

- /var/log/kern.log stores kernel events, errors, and warning logs, which are particularly helpful for troubleshooting custom kernels.
- /var/log/cron stores information about scheduled tasks (cron jobs). Use this data to verify your cron jobs are running successfully.
- Some applications also write log files to this directory. For example, the Apache web server writes logs to the /var/log/apache2 directory (on Debian), while MySQL writes logs to the /var/log/mysql directory.

Importance of Log Analysis

- Compliance with governance and regulatory mandates as well as internal policies
- Tracking security breaches and data exfiltration to determine responsible parties and act to remediate breaches.
- Aid in diagnosing and troubleshooting the full stack
- Tracking user behavior anomalies to detect malicious intent or compromised systems
- Assistance in forensic investigation of malware attacks, exfiltration, or employee theft

Process controlling and management

- Process management on Linux simply operates i.e continue, stop, or terminate , commands that are already running, just started, or already completed.
- A process is a running program.
- This process is created when a command is about to be executed, so we can call it a running instance of a running program.
- Process optimization or control is called process management.

Process can be run in two ways

- **Foreground process**: By default, All the processes are run in the foreground. When a process is run in foreground, no other process can be run on the same terminal until the process is finished or killed. When issuing this type of process, the system receives input from the keyboard(stdin) and gives output to the screen(stdout).
- **Background process**: Adding '&' to a foreground command makes it a background process. A background process runs on its own without input from the keyboard(stdin) and waits for input from the keyboard. While the process runs in the background, other processes can be run in the foreground.
- The background process will be in stop state till input from the keyboard is given (usually 'Enter' key) then becomes a foreground process and gets executed. Only after the background process becomes a foreground process, that process gets completed else it will be a stop state.

Process States

A process in Linux can go through different states after it's created and before it's terminated. These states are:

1. **Running:**A process in **running** state means that it is running or it's ready to run.
2. **Sleeping:** The process is in a **sleeping** state when it is waiting for a resource to be available
   - **Interruptible sleep:**
   - **Uninterruptible sleep**
   - A process in **Interruptible sleep** will wakeup to handle signals, whereas a process in **Uninterruptible sleep** will not.
3. **Stopped:**A process enters a **stopped** state when it receives a stop signal.
4. **Zombie: Zombie** state is when a process is dead but the entry for the process is still present in the table

Commands

- There are two commands available in Linux to track running processes. These two commands are **Top** and **Ps**.
- To stop a process in Linux, use the '**kill**' command. kill command sends a signal to the process.
- There are different types of signals that you can send. However, the most common one is ‘kill -9’ which is ‘**SIGKILL**’.
- **bg -** put suspended process into background
- **fg -** bring process into foreground
- **jobs -** list processes

Online Server upgrade/update process

- Keep configuration backup, including the files in boot partitions. Check software compatibility with new version of OS
- Keep data backup
- Keep mount information s
- Perform yum upgrade or apt-get update depends on OS
- Run dracut*(Dracut is a set of tools that provide enhanced functionality for automating the Linux boot process*)
- Reboot the server
- Check the functionality
- If it works,it is ok. Else restore it to older kernel image from backup and older configuration

Centos/RedHat

- To update your Red Hat-based distro, use the following command while logged into the server via SSH.

```
yum update
```

- Please note that this will prompt you to confirm the package installations that occur during the update process by asking you to press 'y'. If you'd like to avoid this, you may use the aforementioned command with the -y option. The command would be as follows.

```
yum update -y
```

Ubuntu

- To update your Debian or Ubuntu-based distribution, execute the following command while logged into the server via SSH.

```
apt-get update && apt-get dist-upgrade
```

- These are actually two commands:

```
apt-get update
apt-get dist-upgrade
```

Administrating Database

- System administration is a process of setting up, configuring, and managing a computer system.
- System administration involves creating a user account, taking reports, performing backup, updating configuration files, documentation, and performing recovery actions.
- A database administrator (DBA) is the information technician responsible for directing and performing all activities related to maintaining a successful database environment.
- DBA makes sure an organization's databases and related applications operate functionally and efficiently.
- DBA is responsible for understanding and managing the overall database environment. By developing and implementing a strategic blueprint to follow when deploying databases within their organization

Create MySQL Database on Linux

Step 1:

- Log into the MySQL server from the command line with the following command, specifying the user *root* with the *-u* flag, and prompt for a password using the *-p* flag. Enter your current password once prompted to complete the login.

```
mysql -u root –p
```

- We can change the MySQL password for the root user or any other user in the database via the command line.
- A prompt displays like the one below once you log in.

```
mysql>
```

Step 2:

- To create a database with the name *Prime_database*, type the following command.

```
CREATE DATABASE prime_database;
```

Step 3:

View All MySQL Databases

- Use the following command to view the database.

```
SHOW DATABASES;
```

Administering Web

- A web server administrator, also known as a system administrator, is responsible for the maintenance, configuration, and reliable operation of computer systems, especially web servers.
- They are responsible for ensuring that the web server is secure, reliable, and performing at its best.
- Web server administrators must have a strong understanding of computer networks, operating systems, and web server software.
- They must also be able to troubleshoot any issues that arise with the web server.
- Leading web servers include Apache, Microsoft's Internet Information Services (IIS) and Nginx -- pronounced engine X.
- Other web servers include Novell's NetWare server, Google Web Server (GWS) and IBM's family of Domino servers.

What does a web administrator do?

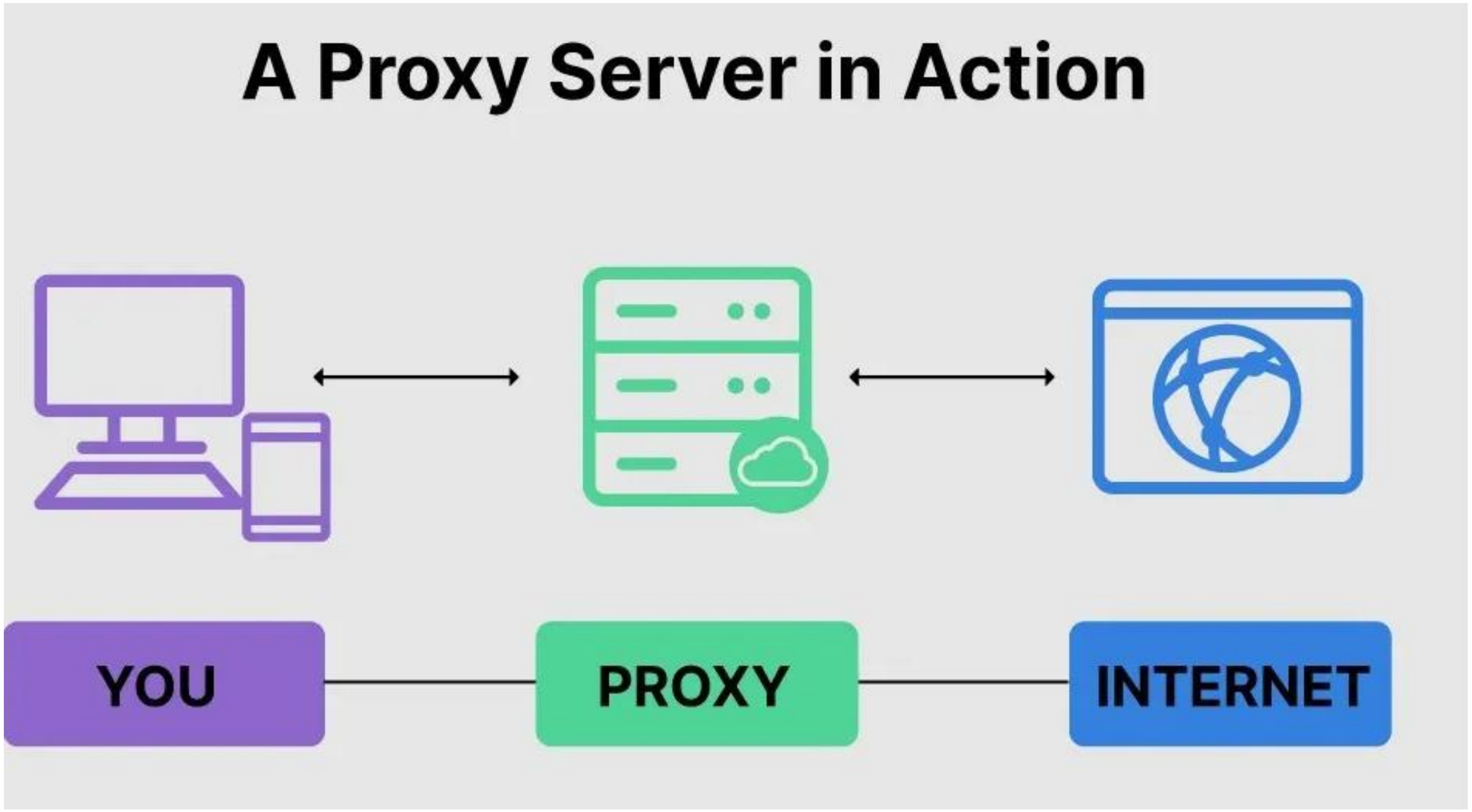
- User Accounts
- Web Security
- Web Servers
- Web Software

❑ CGI,PHP,JSP,Databases

- Log Analysis
- Content Management

Administrating Proxy

- A **proxy server** is a system or router that provides a gateway between users and the internet.
- A proxy server is a server that acts as an intermediary for requests from clients seeking resources on the internet or an external network.


A Proxy Server in Action
YOU
PROXY
INTERNET

Features

- Improve security
- Secure employees' internet activity from people trying to snoop on them
- Balance internet traffic to prevent crashes
- Control the websites employees and staff access in the office
- Save bandwidth by caching files or compressing incoming traffic

QA ??