DevOps Training

Suryaraj Timsina

Recap

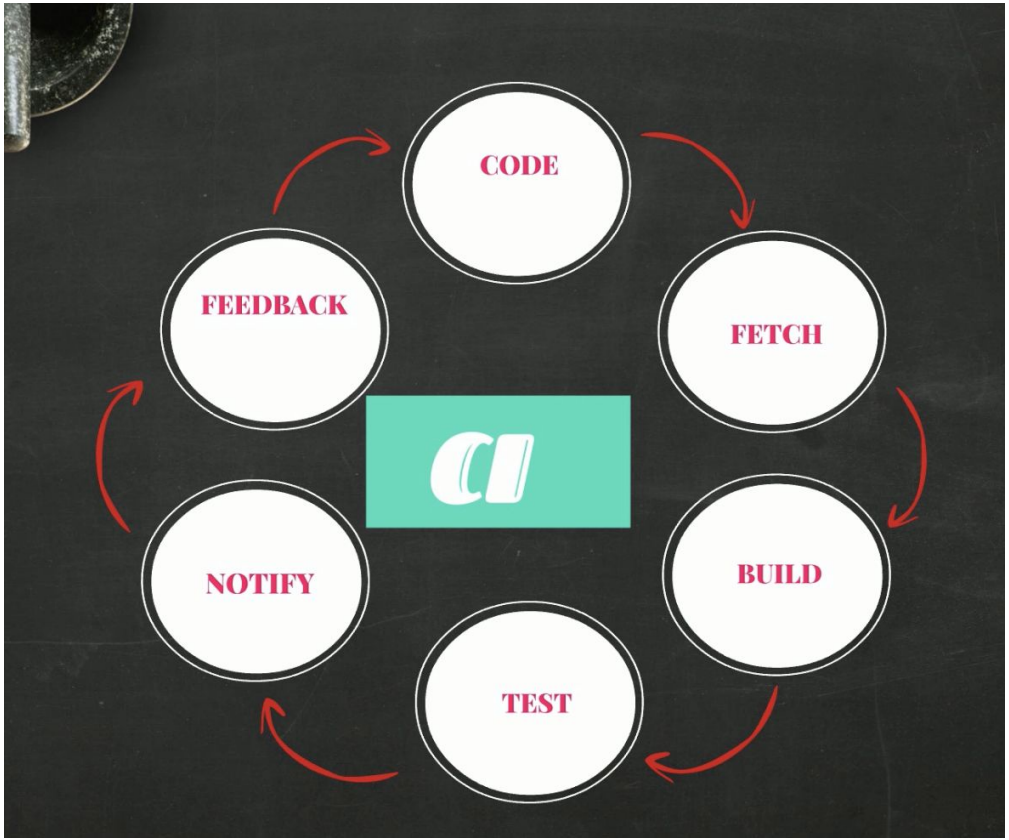
1. Syllabus
2. Introduction
3. Why Learn DevOps?
4. What is DevOps?

Agenda

1. What is Continuous Integration?
2. What is Continuous Delivery?
3. What is Virtualization?
4. VM Setup

What is Continuous Integration?

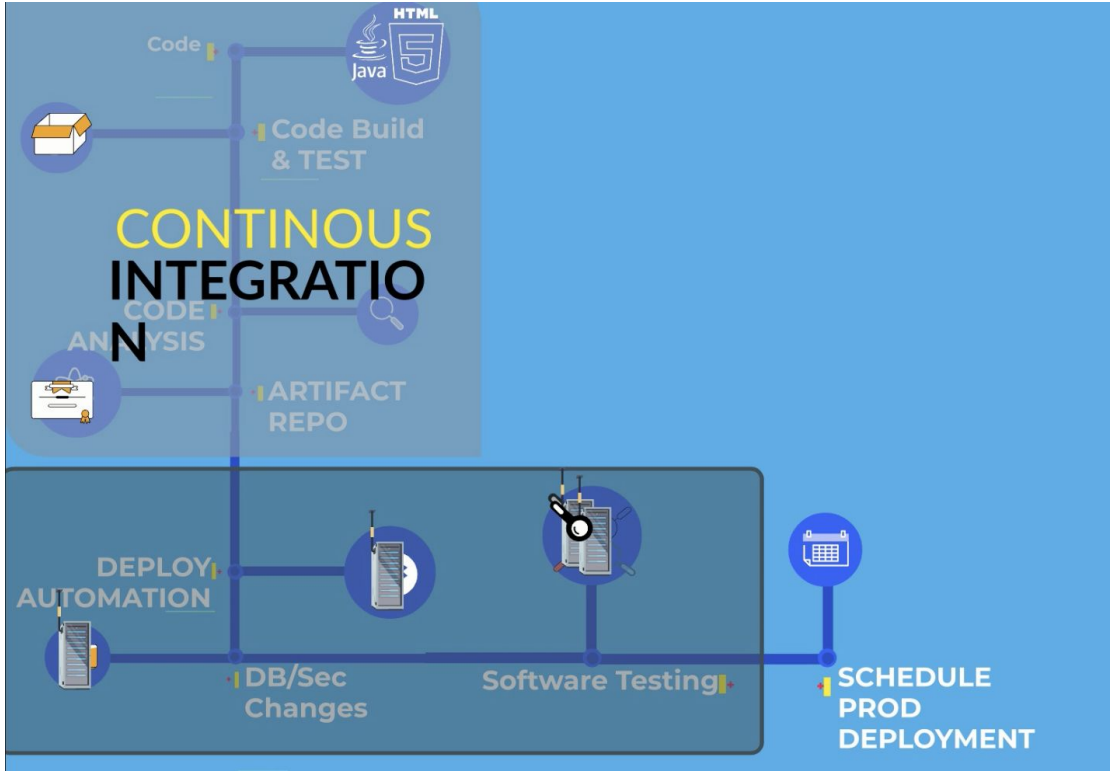
1. An automated process in DevOps which generates software and it's features quickly and efficiently.



What is Continuous Integration?

Goal of CI Process:

To detect the defects at very early stage so it is not multiplied.



It is a software development practice where developers regularly merge their code into a central repository after which automated builds and tests are run.

TOOLS

IDE

Used by developers for coding. These IDE will be integrated with version control system to store and version the code.

- Eclipse
- Visual Studio
- Atom
- Pycharm

TOOLS contd.

Version Control System(VCS)

- GIT
- SVN
- TFS
- PERFORCE

TOOLS contd.

Build Tools

Based on the programming language.

- MAVEN, ANT, GRADLE
- MSBUILD, VISUALBUILD
- IBM URBANCODE
- MAKE
- GRUNT

TOOLS contd.

Software Repository

To store software artifacts

- Sonatype Nexus
- JFROG Artifactory
- Archiva
- CloudSmith Package
- Grunt

TOOLS contd.

CI Tools

Integrates everything

- Jenkins
- CircleCI
- Teamcity
- Bamboo CI
- Cruise Control

What is Continuous Delivery?

1. An automated process of delivery of code changes to servers quickly and efficiently at an enormous phase.
2. It is the extension of continuous integration.
3. Deployment is not just about shipping the software to the servers, it’s more than that.
4. A deployment means also,

- Server provisioning
- Installing dependencies on servers
- Configuration changes
- Network and firewall rule changes.
- Deploy the artifact to the server
- Any other changes.

What is Continuous Delivery?

Problem:
Ops team will be flooded with requests and as CI process will generate faster and regularly.

- Regular deploy requests.
- No clear instructions.
- Already occupied.
- System uptime
- ITIL process driven

After the manual deployment, information will be sent to QA team for testing after conducting testing QA team will send back information.

There is too much of human intervention and manual approval in this process.
Solution:
- Automate it
- Any and every steps in deployment should be automated.

What is Continuous Delivery?

Deployment:

- Server provisioning
- Dependencies
- Config changes
- Network
- Artifact deploy
- Other changes

Tools:

There are alot of automation tools available in the market like

- Ansible, puppet, chef for system automation
- Terraform, cloudformation for cloud infrastructure automation.
- Jenkins, Octopus deploy for CICD automation

What is Continuous Delivery?

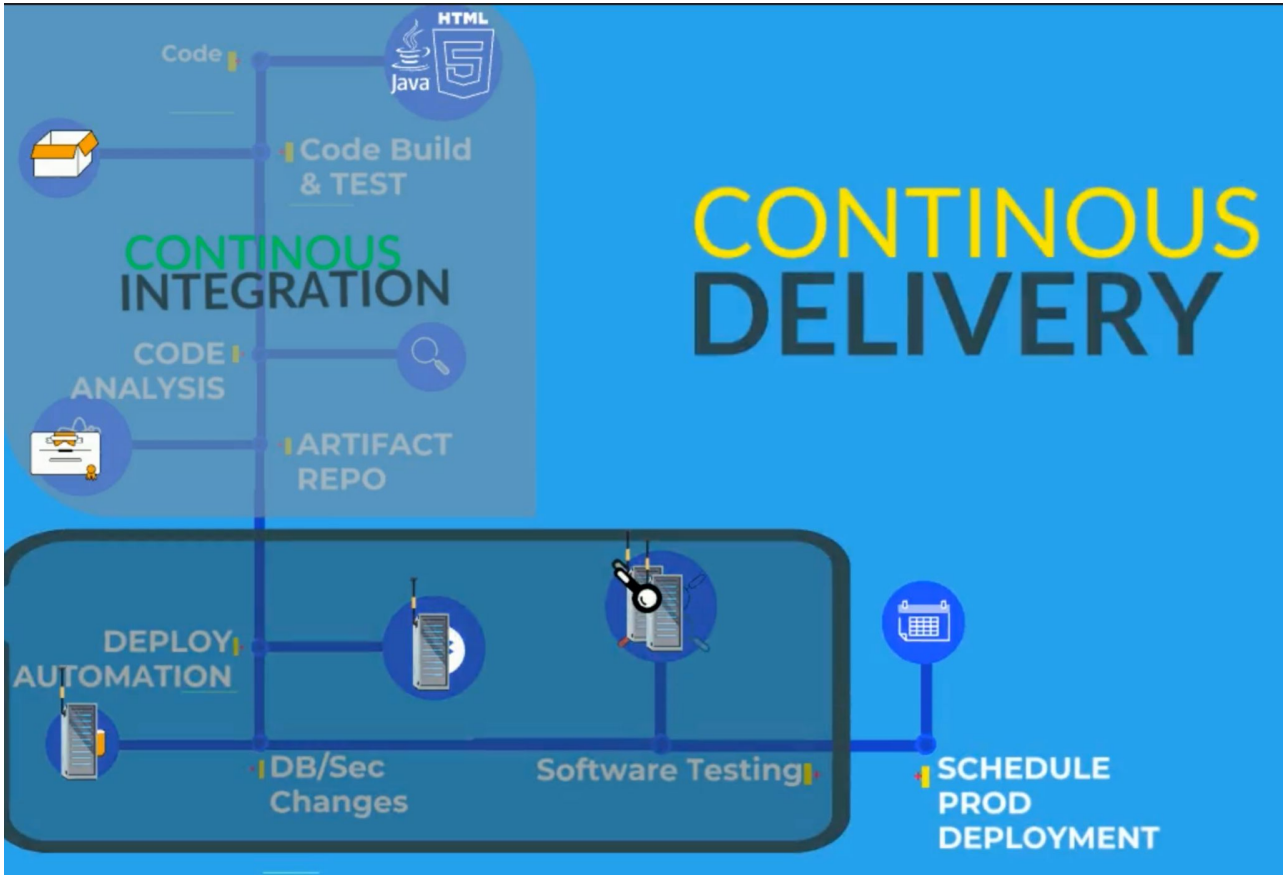
Test Automation

- Software testing also has to be automated.
- Any test process like functional, load, performance, databases, network and security

So Ops team will write automation code for deployment.

Tester will write code for software testing and sync it will developer source code.

What is Continuous Delivery?



What is Virtualization?

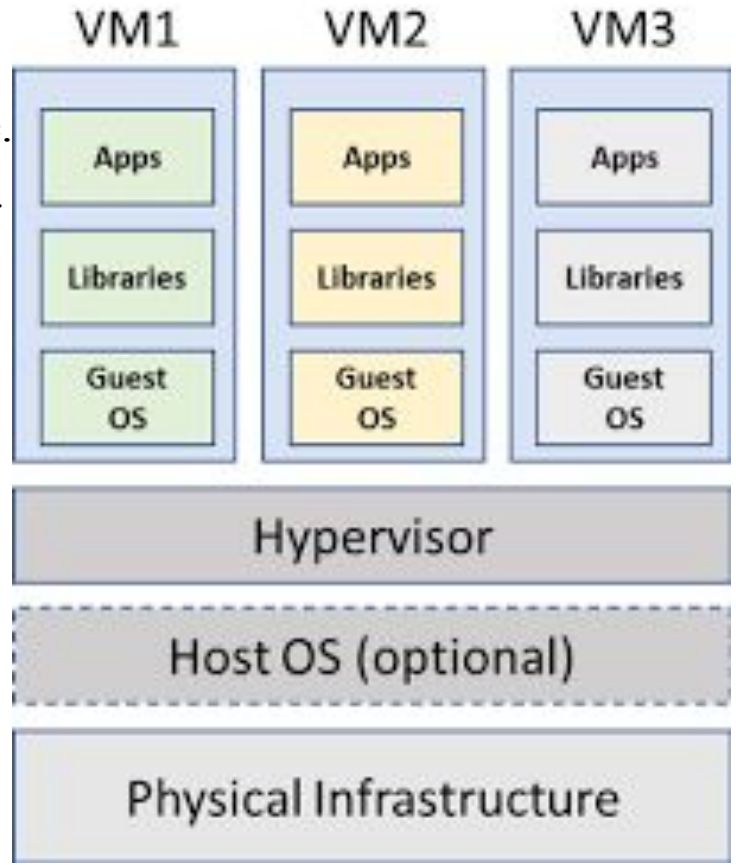
One computer can run multi OS parallelly.

Life before virtualization

- To run App/services we need servers.(Tomcat, apache2, mysql)
- Physical computers(servers in Datacenter)
- One service - One server(Isolation)
- Servers are always overprovisioned. If we need 8gb, 12gb is provisioned
- Server resources mostly underutilized.
- Huge capital expenditure and operating expenditure

What is Virtualization?

- Enter VMWare
- Allow one computer to run multiple OS.
- Partition physical resources in virtual resource.
- Virtual machine runs in isolation environment.
- Each VM needs it’s own OS.
- Server virtualization is the most common Virtualization. But there are other kind of virtualization as well(network, storage).



Terminologies

Host OS

- This is the OS of the physical machine.

Guest OS

- OS of the virtual machine. Virtual machines are also known as guest machine.

VM

- Short form of virtual machine.

Snapshot

- A way of taking backup of VM.

Hypervisor

- Enables virtualization
- Tool or the software that let’s us do or create VM.

Types of Hypervisor

Type-1

- Baremetal
- Runs as a Base OS
- Production
- E.g VMWare EsXI, Xen Hypervisor, Hyper-V

Type-2

- Runs as a software
- Learn and Test Purpose
- E.g Oracle Virtualbox, VMWare server

Creating VM on Virtualbox manually

Prerequisites

- Virtualbox
- Ubuntu 20.04 iso

Creating VM on Virtualbox automatically using Vagrant

What is Vagrant?

Vagrant is an automation tool to manage VM lifecycle, right from creating a virtual machine to making any changes, deleting it, recreating it, provisioning it, anything that we do manually with VMs, we can automate that by using vagrant. Vagrant is a command line tool.

- Create VM Automatically
- Vagrant Commands
- Vagrant Networking
- Provisioning
- RAM, CPU, & Disk
- Multi VM Vagrantfile
- Documentation

Creating VM

VM Management problems

- OS Installations
- Time Consuming
- Manual setup
- Tough Replication for Multi VM
- Documentations for Multi VM

Creating VM

Vagrant for VM's

- No OS installations
- VM Setup through Images(vagrant boxes)
- Images/ Boxes available in Vagrant cloud
- Manage VM's with a file(Vagrantfile)
- VM changes automatic through Vagrantfile
- Vagrant commands to manage VM's
- Provisioning VM/ Executing commands & scripts

Vagrant Setup

- VT(Virtualization Technology) Enabled in BIOS

Creating VM

Vagrant tool

- Hypervisor like Oracle Virtualbox
- CLI(Command line interface) like GIT Bash

VM Setup with Vagrant

- Vagrant Box name from https://app.vagrantup.com/boxes/search
- Project Directory(Folder/Directory at any location of your choice)
- Vagrantfile in Project Directory
- Vagrant commands like 'vagrant up'
- Login with 'vagrant ssh' command for Linux vm's

Linux

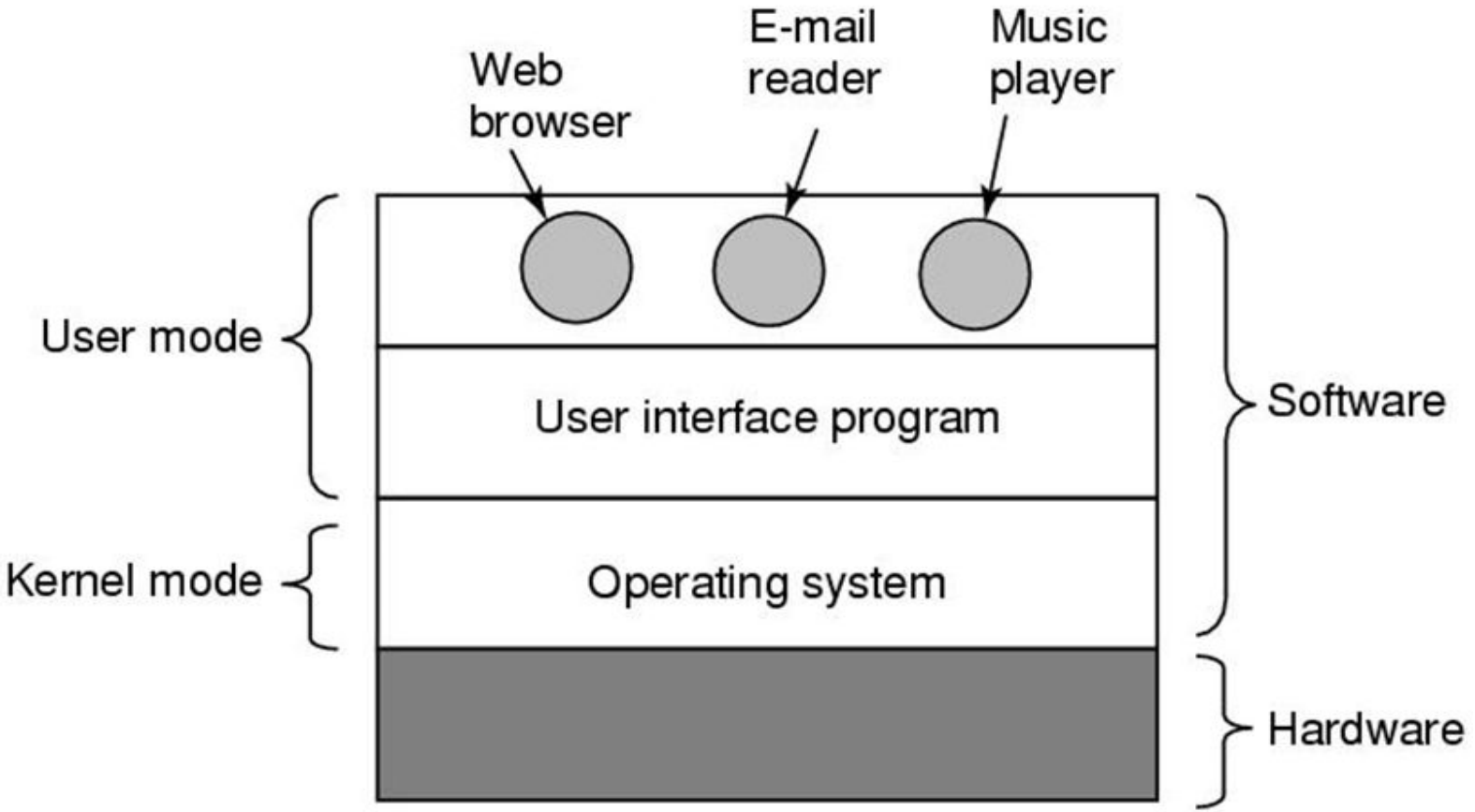
Operating System

- A layer of software required to manage varied components of a computer.
- Acts as interface between user and the computer hardware.
- Processors
- Main memory
- Disks
- Input/output devices

Linux

Operating System

Linux, Where does OS lies?



Linux History:

Linux origin

- 1984: The GNU project and the free software foundation, creates the open source version of UNIX utilities.
- Creates the General Public License(GPL), software license enforcing open source principles.

1991: Linus Torvalds

- Created open source UNIX-like kernel, released under CPL.
- Ports some GNU utilities, solicits, assistance online.

Today:

- Linux Kernel + GNU utilities = complete, open source, UNIX-like operating system.
- Packaged for targeted audiences as distributions.

Linux/ Unix principles

- Everything is a file including hardware.
- Configuration files are in text form.
- Avoid use of GUI.
- Small single purpose program.
- Small programs can be combined to perform complex task.

Open Source Softwares

- Any software that satisfies following criteria are open source software.
- Source code can be download freely.
- Source code can be studied and modified freely.
- Source code can be redistributed without any need of approval.
- E.g Linux, Apache,Mysql, PHP, Perl, Python etc.

Linux Terminology

- Kernel
- Distribution
- Boot loader
- Service
- File system
- X Windows system
- Desktop environment
- Command line

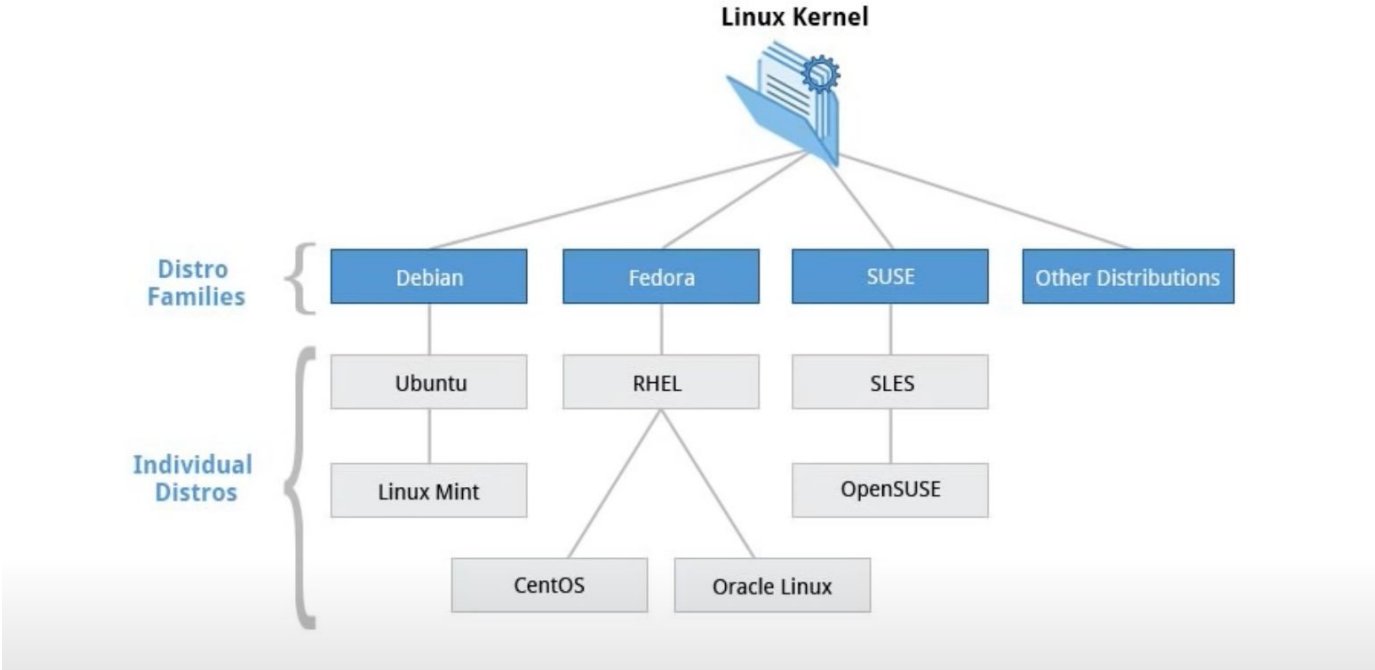
Linux Kernel

- Considered brain of the OS.
- Glue between hardware and application.
- Controls hardware and makes hardware interact with the application.
- E.g Linux Kernel

Linux Distributions

- Collection of programs combined with the linux kernels to make up linux based OS.

  E.g Red hat, Fedora, Ubuntu.



Linux Bootloader

- Program boots the OS.
- E.g GRUB and ISOLINUX.

Linux Service

- Program that runs in the background process.
- E.g httpd, nfsd, ftpd.

Linux File System

- Method for storing and organizing files.
- E.g ext3, ext4, Fat, XFS, NTFS.

Linux X Windows System

- Provides the standard toolkit and protocol to build the graphical user interfaces on nearly all linux distos.

Linux Desktop Environment

- GUI interface on the top of OS.
- E.g: Gnome, KDE, Xfce.

Linux Command Line

- Interface for typing commands on top of OS.

Linux Shell

- Command line interpreter that interprets the command line input and instructs the OS to perform any necessary task and commands.
- E.g bash, tcsh, zsh.

Linux Distros

- Red Hat
- CentOS
- Ubuntu
- Debian
- Kali
- Suse
- Backtrack etc.

CentOS

- Community Enterprise OS.
- Derived from Redhat Enterprise Linux sources.

Ubuntu

- Derivative of debian linux.
- Popular for desktop installation.
- Ubuntu African word which means ‘Humanity to others’, help to others.
- Server and desktop
- Easy to deploy in the cloud: i.e. Amazon EC2, RackSpace Cloud, Custom Cloud, Vmware
- 1-cd <=700 MB which includes GUI, if needed.

/ - Root

- Top-level root directory.
- Every single file and directory starts from the root directory.
- Only root user has write privilege under this directory.

Note: /root is root user’s home directory, which is not same as /.

/bin - User Binaries.

- Contains binary executables.
- Common linux commands you need to use in single-user modes are located under this directory.
- Commands used by all the users of the system are located here.
- For example: ps, ls, ping, grep, cp.

/sbin - System Binaries

- Just like /bin, /sbin also contains binary executables.
- But, the linux commands located under this directory are used typically by system administrator, for system maintenance purpose.
- For example: iptables, reboot, fdisk, ifconfig, swapon.

/etc - Configuration Files

- Contains configuration files required by all programs.
- This also contains startup and shutdown shell scripts used to start/stop individual programs.
- For example: /etc/resolv.conf, /etc/logrotate.conf

/dev - Device Files

- Contains device files.
- These include terminal devices, usb, or any device attached to the system.
- For example: /dev/tty1, /dev/usbmon0.

/proc - Process Information

- Contains information about system process.
- This is a pseudo file system contains information about running process. For example: /proc/{pid} directory contains information about the process with that particular pid.
- This is a virtual filesystem with text information about system resources. For example: /proc/uptime.

/var - Variable Files

- var stands for variable files.
- Content of the files that are expected to grow can be found under this directory.
- This includes — system log files (/var/log); packages and database files (/var/lib); emails (/var/mail); print queues (/var/spool); lock files (/var/lock); temp files needed across reboots (/var/tmp);

/tmp - Temporary Files

- Directory that contains temporary files created by system and users.
- Files under this directory are deleted when system is rebooted.

/usr - User Programs

- Contains binaries, libraries, documentation, and source-code for second level programs.
- /usr/bin contains binary files for user programs. If you can't find a user binary under /bin, look under /usr/bin. For example: at, awk, cc, less, scp
- /usr/sbin contains binary files for system administrators. If you can't find a system binary under /sbin, look under /usr/sbin. For example: atd, cron, sshd, useradd, userdel
- /usr/lib contains libraries for /usr/bin and /usr/sbin
- /usr/local contains users programs that you install from source. For example, when you install apache from source, it goes under /usr/local/apache2

/home - Home Directories

- Home directories for all users to store their personal files.
- For example: /home/srtimsina, /home/student

/boot - Boot Loader Files

- Contains boot loader related files.
- Kernel initrd, vmlinux, grub files are located under /boot
- For example: initrd.img-2.6.32-24-generic, vmlinuz-2.6.32-24-generic

/lib - System Libraries

- Contains library files that supports the binaries located under /bin and /sbin
- Library filenames are either ld* or lib*.so.*
- For example: ld-2.11.1.so, libncurses.so.5.7

/opt - Optional Add-on Applications

- opt stands for optional.
- Contains add-on applications from individual vendors.
- add-on applications should be installed under either /opt/ or /opt/ sub-directory.

/mnt - Mount Directory

- Temporary mount directory where sysadmins can mount filesystems.

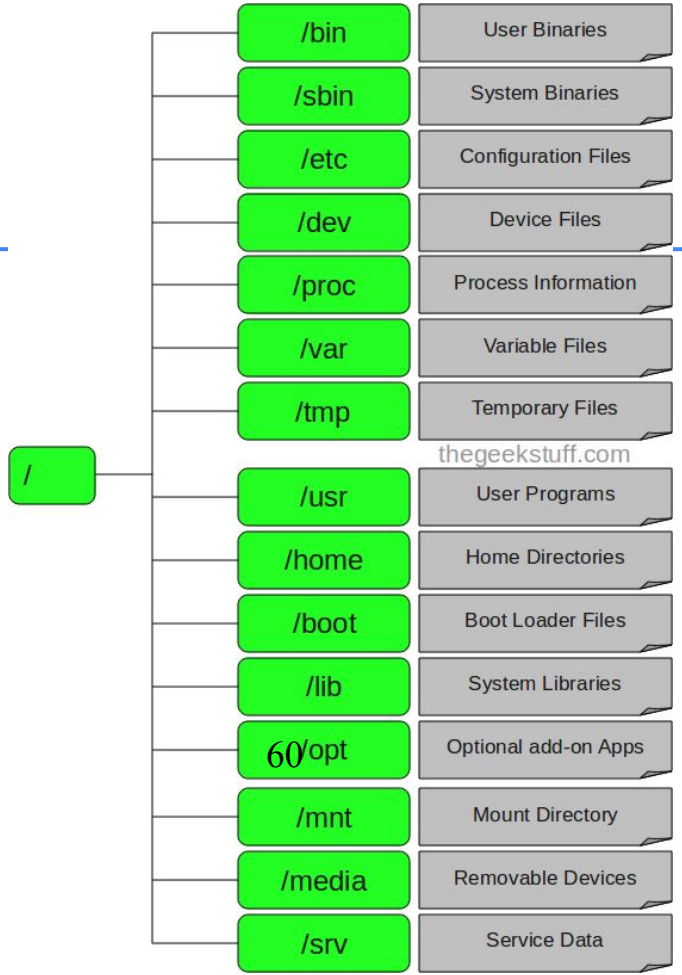
/media - Removable Media Devices

- Temporary mount directory for removable devices.
- For examples, /media/cdrom for CD-ROM; /media/floppy for floppy drives; /media/cdrecorder for CD writer.

/srv - Service Data

- srv stands for service.
- Contains server specific services related data.
- For example, /srv/cvs contains CVS related data.

File System Structure



Login Into Linux

- Need to send username and password.
- Login types
- Graphical //gives desktop interface to supply username and password.
- Simple text //gives shell prompt to supply username and password.

Login Into Linux

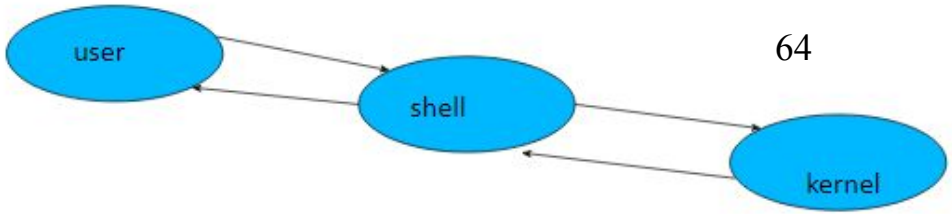
- Shell prompt usually ends in a dollar sign ($)
- [srtimsina@example.com ~]$
- [root@example.com ~]#

Logging Out

- Exit command
- Ctrl+D
- After a logout, new login screen should appear.

Shells

- A shell provides an interface between the user and the operating system kernel
- Either a command interpreter or a graphical user interface
- Traditional Unix shells are command-line interfaces (CLIs)
- Usually started automatically when you log in or open a terminal



Remote Login

- Via ssh.
- SSH server must be running in the machine.
- SSH client is needed in the client machine.
- $sudo apt-get install openssh-server //installs ssh server
- SSH clients
- For windows machine, Xshell, Putty.
- For linux machines

- $sudo apt-get install openssh-client //installs ssh client

Checking The Service Status

- Command Syntax [service] [service_name] [command] or
- [systemctl] [command] [service_name]
- #serivce sshd status
- #systemctl status sshd
- Commands can be, start, restart, reload, status, stop

Text Editors

- vim
- nano
- gedit
- VIM