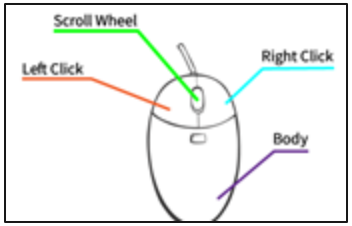
DEVOPS EASY LEARNING

Linux For DevOps Engineer

Keyboard Options

Keyboard Symbols



DEVOPS EASY LEARNING

Keyboard Symbols

DEVOPS EASY LEARNING

Symbol	Explanation
~	Tildle
`	Back quote
!	Exclamation mark, exclamation point, or bang.
@	At symbol
&	Ampersand
*	**Wildcard**, Asterisk, Star, multiplication
|	Pipe
$	Dollar Sign

Symbol	Explanation
?	Question Mark
(	Open or left parenthesis
)	Close or right parenthesis
{	Open brace or curly bracket
}	Close brace or curly bracket
[	Open bracket
]	Close bracket
.	Period or dot

Keyboard Symbols

DEVOPS EASY LEARNING

Symbol	Explanation
\	Backslash
/	Forward slash
!	Exclamation mark, exclamation point, or bang.
:	Colon
"	Double quote
'	Single quote
;	Semicolon
,	Comma

Symbol	Explanation
_	Underscore
+	Plus
-	Hyphen or dash or minus
=	equal
==	Double equal
#	Pound or hash
%	Percent
Backspace	Backspace (or Backspace) key

Keyboard Symbols

DEVOPS EASY LEARNING

Symbol	Explanation
<	Less than
>	Greater than
>>	Greater than or equal to
<<	Less than equal to
!=	Not equal to
ctrl+l	Clear Linux Terminal
ctrl+shift+t	Open google page that you close it
ctrl+c	Cancel
ctrl+z	undo

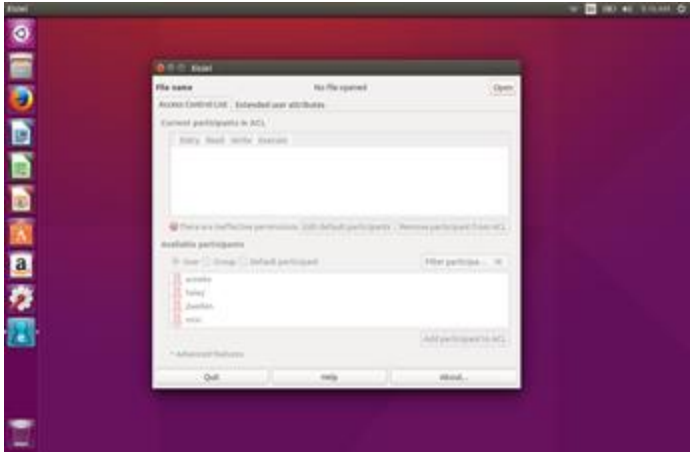
Symbol	Explanation
tab	Tabulation or tab key
ctrl+c	Copy clipboard
ctrl+v	Past from clipboard
esc	Esc (escape) key
arrows	Up, down, left, right Arrow keys.
Backspace	Backspace (or Backspace) key
Num Lock	Num Lock key
ctrl+s	save
ctrl+y	redo

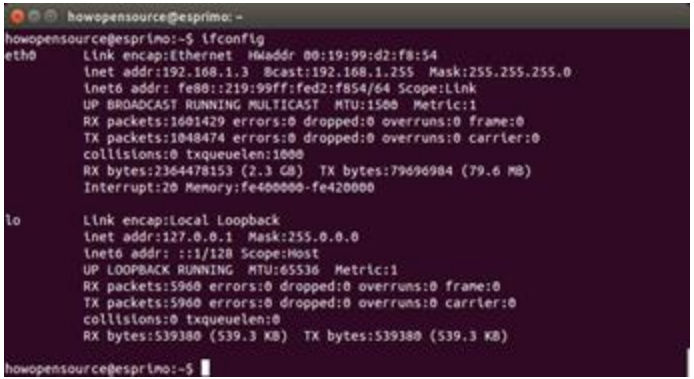
GUI and CLI

GUI Vs CLI

In IT world, ***GUI*** stand for graphical user interface; It is an interface through which a user interacts with electronic devices such as computers and smartphones through the use of icons, menus and other visual indicators or representations (graphics)

In IT world, ***CLI*** stand for command line interface. It is a text-based interface that is used to operate software and operating systems while allowing the user to respond to visual prompts by typing single commands into the interface and receiving a reply in the same way.

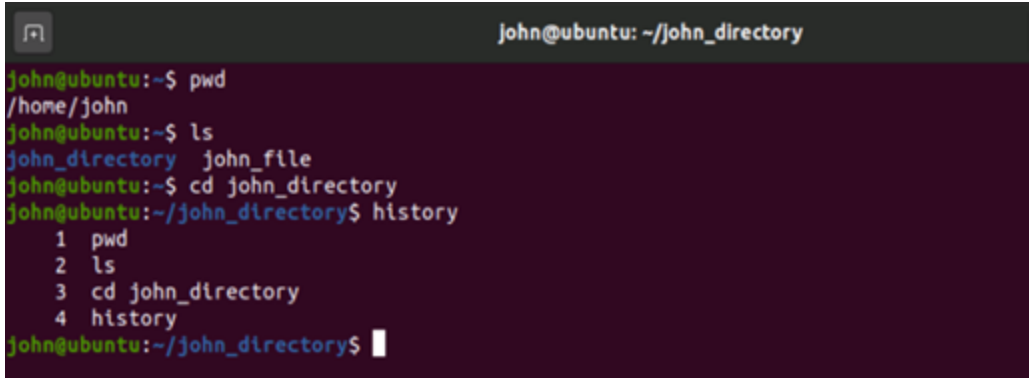




DEVOPS EASY LEARNING

GUI Vs CLI

- **CLI OS**
  - Unix (Solaris, AIX or HP-UX)
  - Linux (Redhat, CentOS, Ubuntu, Fedora)



- **GUI OS**
  - Windows
  - MacOS

DEVOPS EASY LEARNING

GUI Vs CLI

CLI	GUI
Interaction is by typing commands	Interaction with devices is by graphics and visual components and icons
Commands need to be memorized	Visual indicators and icons are easy to understand
Use a lot less memory to store programs	Use large amount of memory due to graphics
Use of keyboard for commands makes CLI quicker	Use of mouse for interaction makes it slow
Only keyboard is used	Mouse and keyboard is used
Command line interface does not change, remains same over time	Structure and design can change with updates
With a CLI, users have all the control over the filesystem and operating system, and the tasks become simple. You can create a script that contains a few lines of command and it will do the work for you	Although GUI's can create shortcuts, they do not readily support scripting or automation. For commontasks, a user must repeat each action within the GUI manually

- **GUI** = Graphical user interface
- **CLI** = Command line interface

DEVOPS EASY LEARNING

Linux Distributions

What is Linux?

- Just like Windows, iOS, and Mac OS, **Linux** is an operating system.
- In fact, one of the **most popular platforms on the planet**, Android, is powered by the Linux operating system.
- Most applications there we used out there are being hosted on the Linux platform
- An operating system is software that manages all of the hardware resources associated with your desktop or laptop.
- To put it simply, the operating system manages the communication between your software and your hardware. Without the operating system (OS), the software wouldn't function.

DEVOPS EASY LEARNING

When Linux?

In 1969, Ken Thompson and Dennis Ritchie Created the first version of Unix OS (Ken Thompson and Dennis Ritchie are considered the inventors of Unix.).

The system was multitasking and multi-users which means it could support multi-user at the same time. This makes it very popular and a lot of companies start creating their own version. For example, **HP created HP-UX, IBM created AIX, solaris created Oracle, Apple created MAC OS.**

Unix was a close source and **Linus Torvalds** was not satisfied, and he started developing his own version of software, and he came out with **Linux software 1991**. He released the Linux project under the **GNU** project or Licence or GPL(general public licence). Since then, they have been a lot of modifications of the linux software and this gave birth to some **Linux flavor** or distributions that we are using today such as: CentOS which is the clone of Redhat, Ubuntu, Fedora, Debian, Opensuse, Linux Mint, Kali and so on.

UNIX Distributions (close source): Apple Mac OS (created by Apple), Oracle Solaris (created by Oracle), Oracle Solaris (created by Oracle), AIX (created by IBM)

Linux creation announcement by Torvalds

"Starting this Thanksgiving I am going to write a complete Unix-compatible software system called GNU (for Gnu's Not Unix), and give it away free to everyone who can use it. Contributions of time, money, programs and equipment are greatly needed."

DEVOPS EASY LEARNING

Why Use Linux?

This is the one question that most people ask. Why bother learning a completely different computing environment, when the operating system that ships with most desktops, laptops, and servers works just fine?

To answer that question, I would pose another question. Does that operating system you're currently using really work "just fine"? Or, d**o you find yourself battling obstacles like viruses, malware, slow downs, crashes, costly repairs, and licensing fees?**

If you struggle with the above, **Linux might be the perfect platform for you**. Linux has evolved into one of the most reliable computer ecosystems on the planet. Combine that reliability with zero cost of entry and you have the perfect solution for a desktop platform.

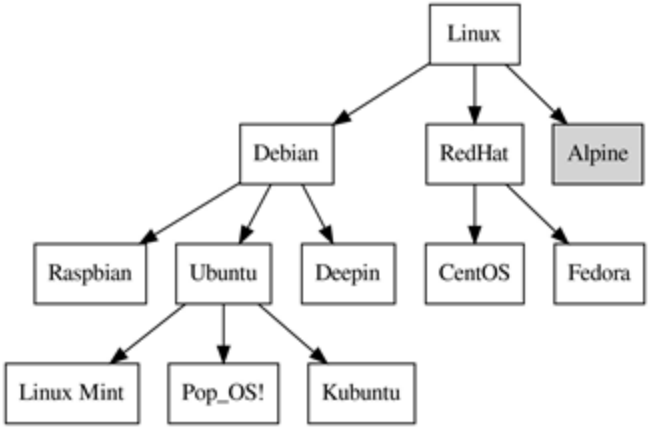
That's right, zero cost of entry… as in free. You can install Linux on as many computers as you like **without paying a cent for software or server licensing.**

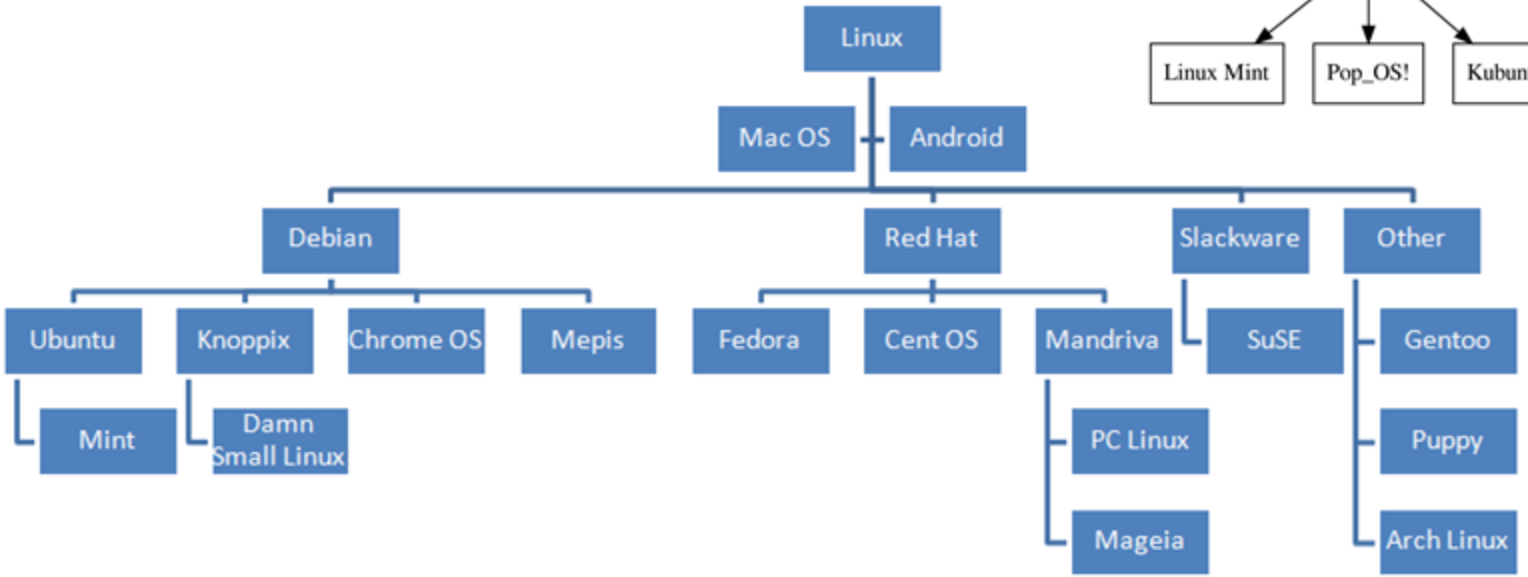
Let just say thanks to Torvalds

DEVOPS EASY LEARNING

What is a Distribution?

Linux has a number of different versions to suit any type of user, you'll find a **"flavor"** of Linux to match your needs. These versions are called **distributions** (or, in the short form, **"distros"**). Nearly every distribution of Linux can be downloaded for free, burned onto disk (or USB thumb drive), and installed (on as many machines as you like).

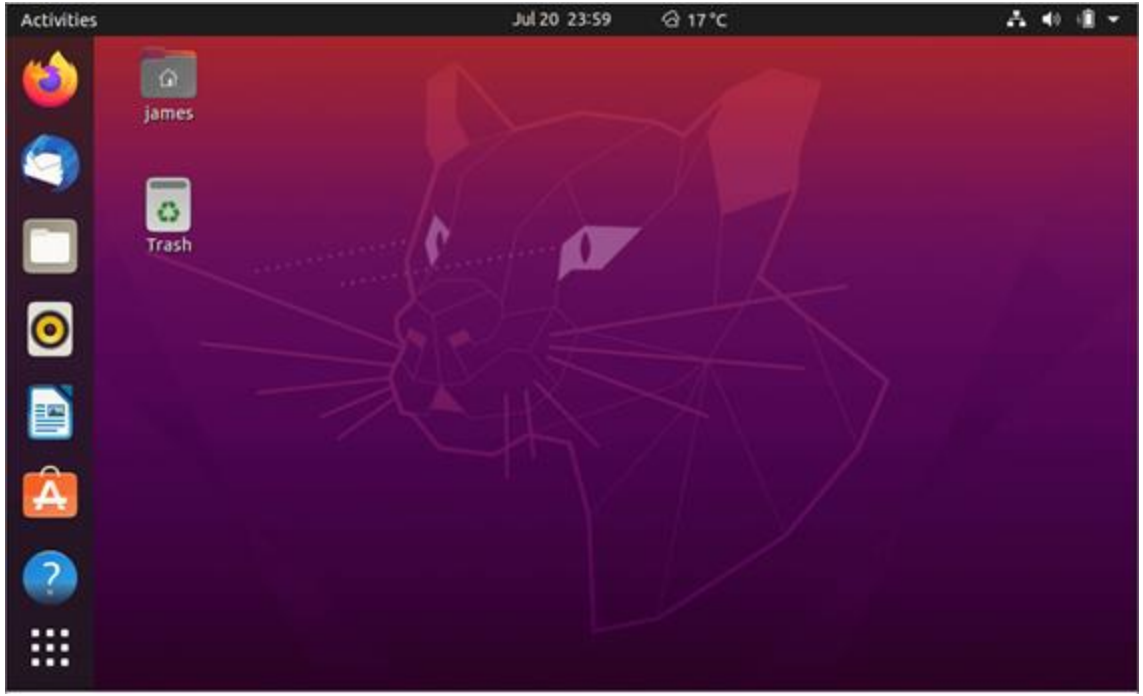




DEVOPS EASY LEARNING

What are the most Popular Linux Distro?

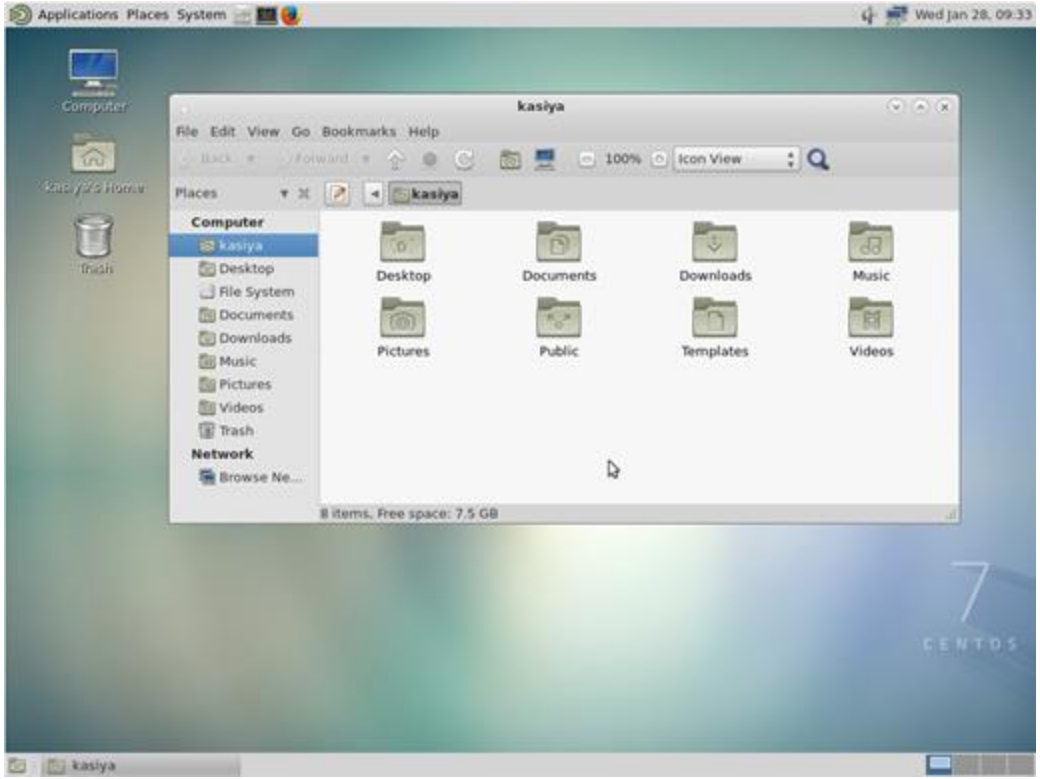
- Ubuntu is one of the most popular Linux distributions use out there
- We Will be using Ubuntu throughout this course
- All our server are ubuntu servers



DEVOPS EASY LEARNING

What are the most Popular Linux Distro?

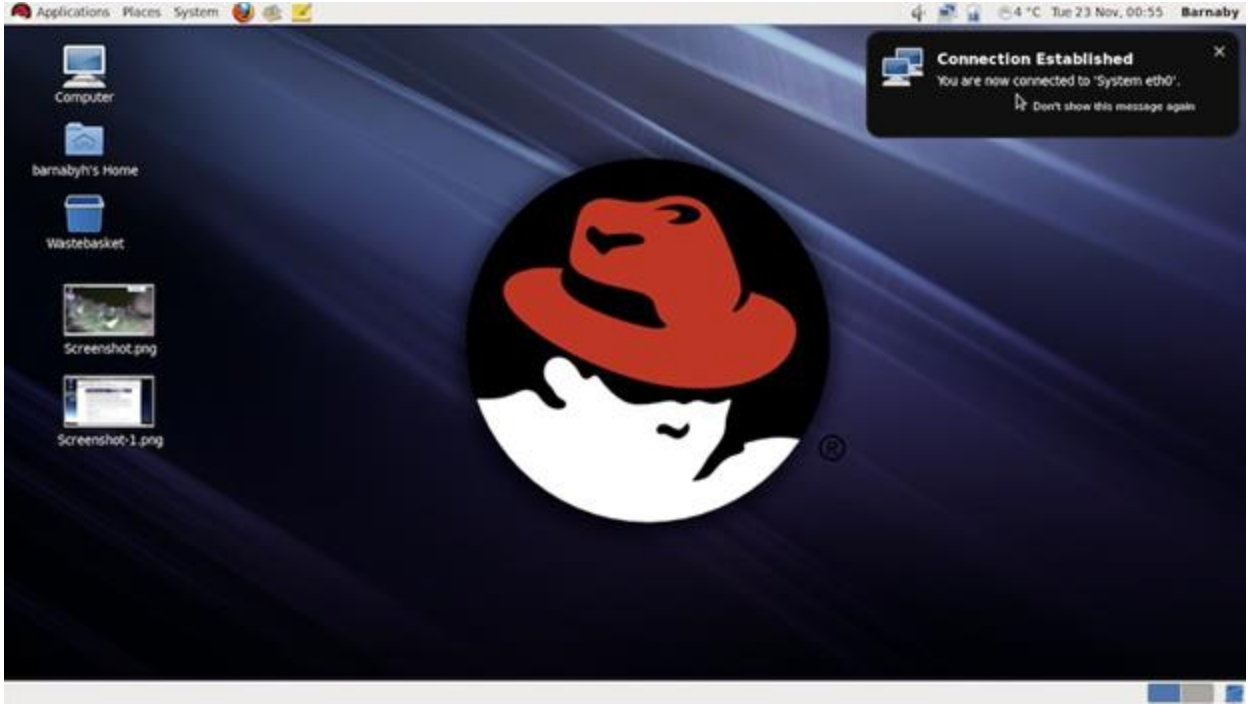
- Ubuntu is also one of the most popular Linux distributions use out there
- We will start using CentOs when we start **Docker**



DEVOPS EASY LEARNING

What are the most Popular Linux Distro?

- RedHat is also one of the most popular Linux distributions use out there
- We will start using CentOs when we start **Docker**



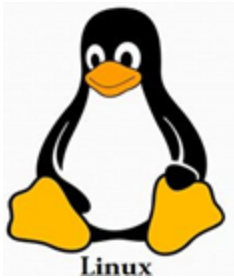
DEVOPS EASY LEARNING

Linux Vs Windows Vs Unix

Linux Vs Windows Vs Unix

What is Linux?

It is an open-source and free operating system (OS) that provides its users with c compatibility with the user interface and programming interface. It is based on the Unix standards and consists of many elements that are developed separately. It is mostly used for servers. It was created in created in **1991 by Torvalds**



What is Windows?

It is a licensed OS with an inaccessible source code. Windows works pretty well for all users who have **very little to no computer knowledge**. It is good for commercial use among businesses as well as for personal use. It is very straightforward and easy to use. Windows is software owned by **Microsoft** and it was created in **1985**. No matter how many talented developers Microsoft hires, finding exploits will never be as fast as in Linux since there is a **big community behind it.**

What is Unix?

Unix is a powerful and multitasking operating system that behaves like a bridge between the user and the computer. It allows the user to perform specific functions. This operating system was launched in **1960**, and was released by **AT&T Bell Labs.**

DEVOPS EASY LEARNING

Linux Vs Windows

Linux	Windows
Linux is a open source operating system	While windows are the not the open source operating system
Linux is free of cost	While it is costly
Linux was created in 1991 by Torvalds	Windows was created in 1985 by Microsoft
Linux in case-sensitive	Windows do not care about case-sensitive
Linux is more efficient in comparison of windows	While windows are less efficient
Linux can run for a very long time without restarting	While Microsoft will even force and restart windows users PC for updates
There is a forward slash **(/)** is used for Separating the directories.	While there is backslash **(\ )** is used for Separating the directories.
Linux is highly secure as compared to Windows	While it provides less security than linux
Linux is widely used in hacking purpose based systems	While windows does not provide much efficiency in hacking

DEVOPS EASY LEARNING

Linux Vs Windows

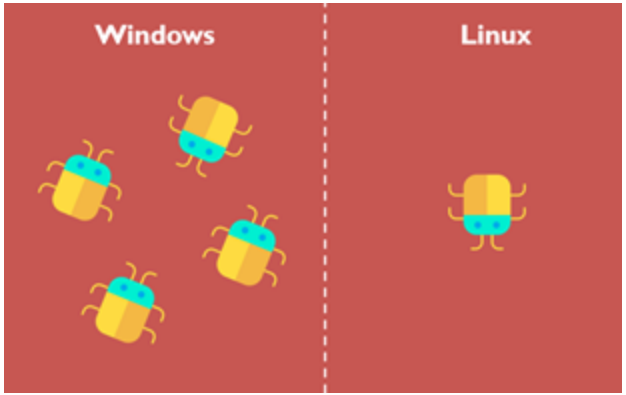
Linux	Windows
There are 3 types of user account – (1) Regular , (2) Root , (3) Service account	There are 4 types of user account – (1) Administrator , (2) Standard , (3) Child , (4) Gues
Root user is the super user and has all administrative privileges	Administrator user has all administrative privileges of computers.
Linux is used for servers	While Windows is mostly used for home PC
Linux file naming convention in case sensitive. Thus, sample and SAMPLE are 2 different files in Linux/Unix operating system	In Windows, you cannot have 2 files with the same name in the same folder

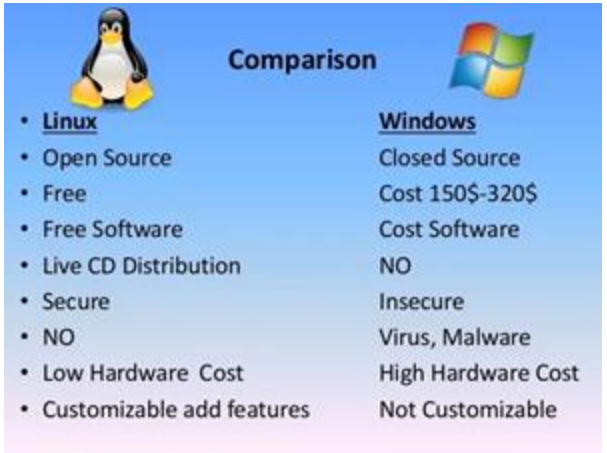
DEVOPS EASY LEARNING

Linux Vs Windows

Why Linux is more secure than Windows?

Linux is based on a multi-user architecture, making it way more stable than a single-user OS like Windows. As Linux is community-driven with regular **monitoring by the developers from every corner of the earth**, any new problem raised can be solved within a few hours and the necessary patch can be ready for supply.





DEVOPS EASY LEARNING

Linux Vs Unix

Features	Linux	Unix
Launched by	This operating system was launched by Linus Torvalds at the University of Helsinki in 1991.	This operating system was launched in 1960 and released by AT&T Bell Labs.
Written in	C and other programming languages	C and other programming languages
Usage	It is used in several systems like desktop, smartphones, mainframes and servers	Unix is majorly used on workstations and servers.
Examples	Some examples of Linux are: Fedora, Debian, Red Hat, Ubuntu, Android, etc.	Some examples of unix are IBM AIX, Darwin, Solaris, HP-UX, macOS X, etc.
Security	Linux provides higher security	Unix is also highly secured
Price	Linux is free and its corporate support is available at any time by the community	Unix is not totally free. There are some Unix versions that are not free such as Mac OS, other than that UNIX is expensive.

DEVOPS EASY LEARNING

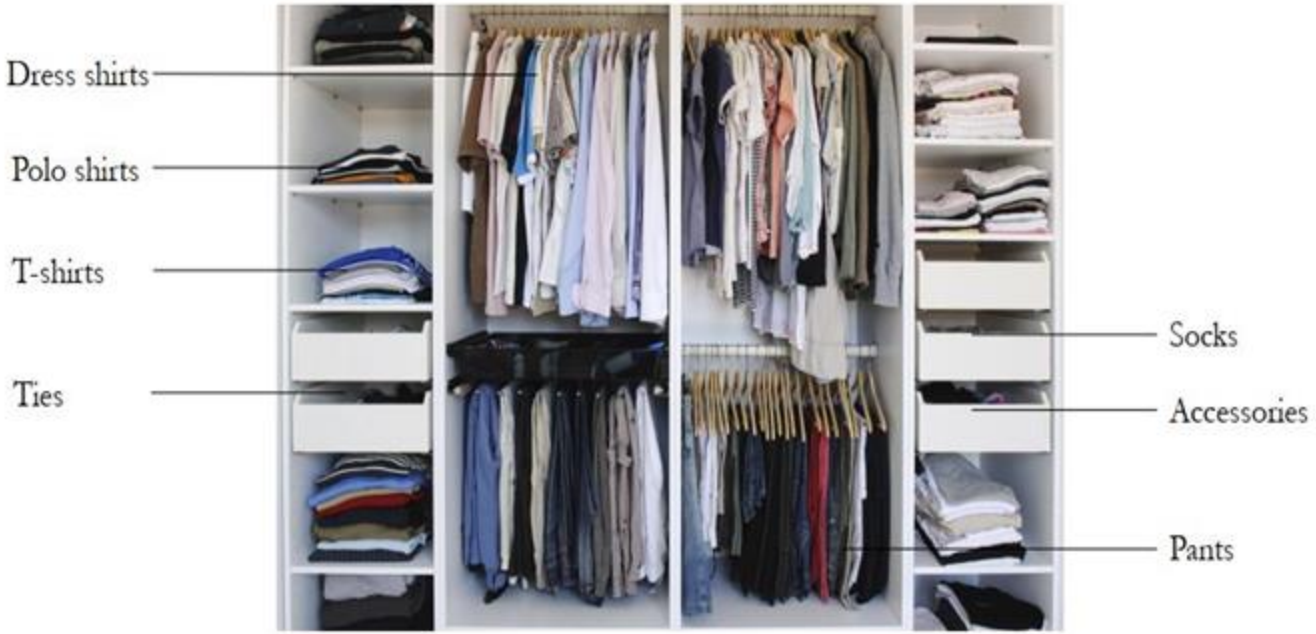
Linux File System

Linux File System

- File system is how files/data is stored or organized in computer
- Without a file system, data placed in a storage would be one large body of data with no way to tell where one piece of data stops and the next begins
- Example of a closet without any organisation

DEVOPS EASY LEARNING

Linux File System

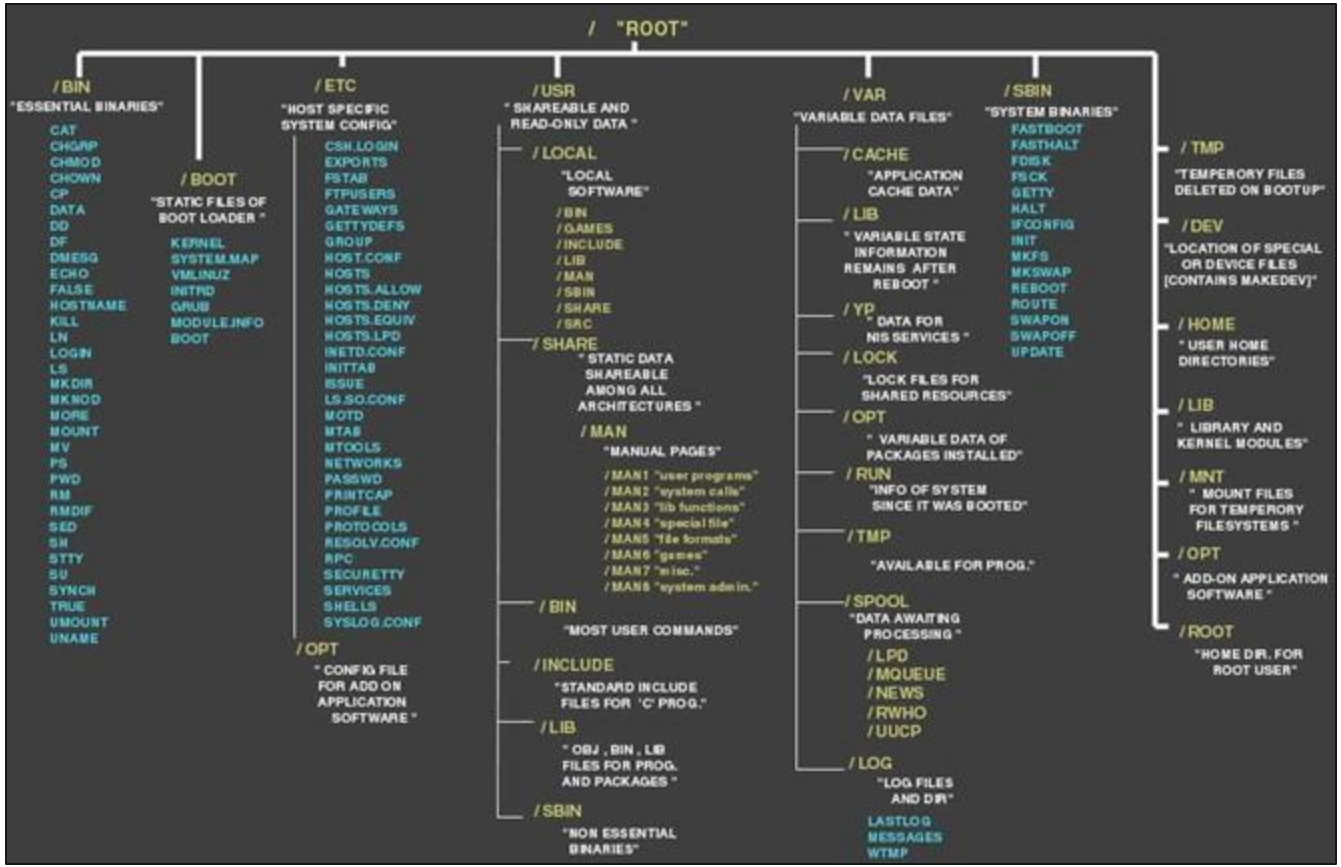


This is a well organized and structured closet. Linux file system looks the same

DEVOPS EASY LEARNING

Linux File System

- The filesystem starts with a **root** represented by a **slash (/)**.
- It is the same as a C drive in Windows.
- The root is the highest level of the organizational hierarchy of the filesystem. Each system only has **one filesystem**.



DEVOPS EASY LEARNING

Linux File System

ROOT (/) it's like a C drive-in window. It can read, write or execute any program

HOME: That's where the regular user's home folders are stored. The ***ROOT*** is where the root's home folder is stored.

ETC: it is where most programs keep their configuration files

BIN: it is where regular user commands are stored (executable programs are stored in **BIN AND SBIN**)

SBIN: it is where root user commands are stored (executable programs are stored in BIN AND SBIN)

DEV: it is where the system keeps references to all the hardware it has such as hard drives, memory, CPUs, and everything else.

MNT: You may also need to work in the mnt folder which is where local or network filesystems are mounted into the overall system or the media folder where removable file systems like USB drives and optical drives are mounted.

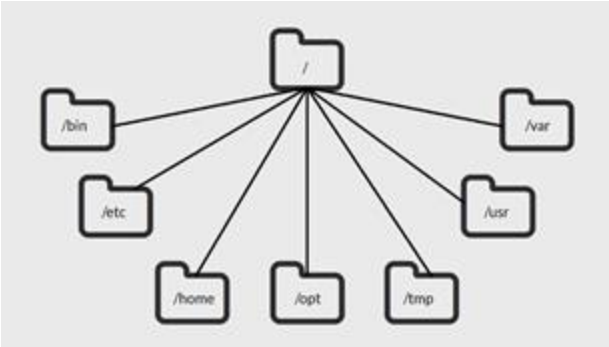
TMP: it is a temporary directory. It contains temporary data that will be erased when you reboot your computer

USR: user directory. It contains user data and programs

VAR: variable data directory. It is where the system must be able to write files and folders during operation. It's the system LOG

LIB: it is where 32-bit programs are installed

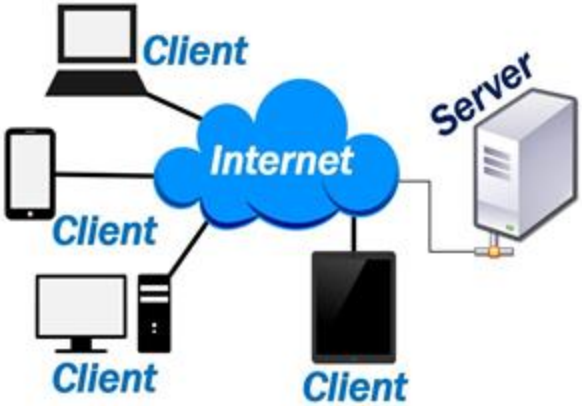
LIB64: it is where 64-bit programs are installed



DEVOPS EASY LEARNING

Terminology

What is Client and server machine?



Client Machine:

- It a computer that we use to connect to the remote server
- It a computer that connects and requests information from a server.
- We used a client machine to connect or request information or download contents from the remote server

Server or Node or Host:

- It is a computer accessed by a user working at a remote location.
- It a computer that is responsible for responding to requests made by a client program
- All information requested by the client are stirred on the server
- All student will used their **client machines (Mac/Windows/Linux)** to connect to our **remote servers** in the Cloud to do the hand on.

DEVOPS EASY LEARNING

What is a Datacenter?

DEVOPS EASY LEARNING

Where does company store all the servers?

- It is the CEO room or house or office or girlfriend house?
- It is in the CEO garage?
- It is in the company corporate office?
- Where is store then?

Datacenter:

It is where we have all the servers for an organization or where we have a massive server or **node** for a company or heard of the business.

It a good idea to store servers in the datacenter today

- If yes, why?
- If no, why?

What is a Cloud Computing?

Simply put, **cloud computing** is the delivery of computing services including servers, storage, databases, networking, software over the Internet **("the cloud")** to offer faster innovation, flexible resources, and **economies of scale**.

Some issues before cloud computing:

- Scalability issues
- Maintenance issue
- We lose money because some servers are idle

Pros of cloud computing:

- We can scale easily
- Cost saving because we pay just for what we used
- Solved servers maintenance issues
- Auto-scaling and high availability (data is store in multiple availability zones)

What is On premise?

On Promise:

- On-premises means a software & a hardware infrastructural setup deployed & running from within the confines of your organization.
- You have the complete control over the **infrastructural setup**.
- Data stays in your private network, nobody other than your team has access to the information.
- In the present industry trend when everyone is moving **towards the cloud,** deploying their software on AWS or Google cloud. **Why would you want to deploy your code on your private on-premises infrastructure?**
- **This why we are use cloud also guys**

What is a hybrid cloud?

- Hybrid cloud is IT infrastructure that connects at least one public cloud and at least one private cloud
- Hybrid cloud refers to a cloud environment made up of both private cloud resources (on promise for instance) and public cloud resources (on AWS for instance)

DEVOPS EASY LEARNING

What is a File, Folder and Directory?

A ***file*** Also known as a "document" is an object on a computer that stores data, information, settings, or commands used with a computer program.

***Directory*.** Also known as a "folder", a *directory* is a container that stores files and directories.

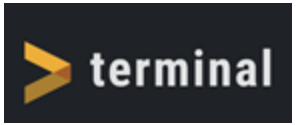
PS: A folder is called directory in Linux.

DEVOPS EASY LEARNING

What is a SSH?

How can I connect to a remote server using a client machine?

- We have multiples that we can use to connect to a remote machine or server
- If you are a Mac user, you can just use your build tools call **terminal**
- In Windows, we have some build in tools such a **CMD** and **Powershell**
- If you don't want to use the build in tools in Windows, you can install a third-party such:
  - **Git Bash**
  - **MobaXterm**
  - **Putty**

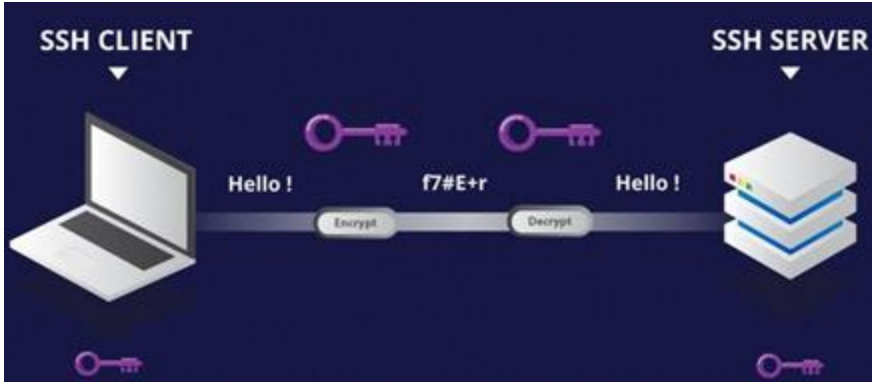


What is SSH?

- **It a secure shell**
- It is used to securely login or connect to a remove server
- The communication going back and forth between the client and the server is **encrypted** when you connect to the server through SSH
- **When you login to the server using ssh, you are no more woking on you client machine**

PuTTY

How to Connect to a Server Using SSH?



You need the following to connect to a remote server:

- A client machine (Windows / Mac / Linux / Unix)
- Any SSH command line of your choice
- An account on the server (username and password)
- SSH port (22 is the default port for SSH and it can be changed to enhance security)
- Server IP address (we do not use server IP at DEL because it is not professional) or DNS (domain name service)

Command to login using SSH:

- ssh [username]@server-ip
- ssh [username]@dns
- ssh -p [port] [username]@dns
- ssh [username]@[@server-ip] -i [key path]

What is a port?

- To get into a server, we need to open the door
- We open the door before we get into our house
- The same thing happens with port. We need to open before we get into the server and that door is call **port**

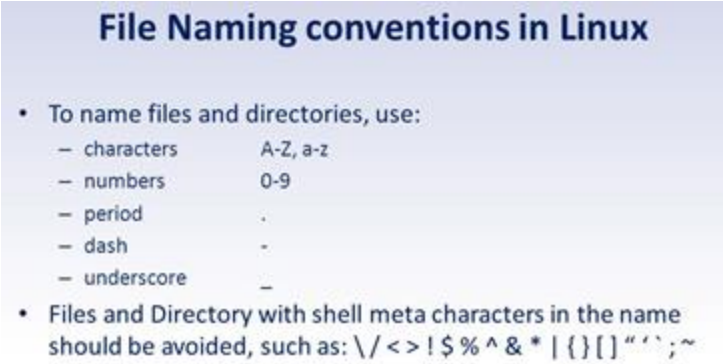
ssh-keygen -R server3.anomicatech.com

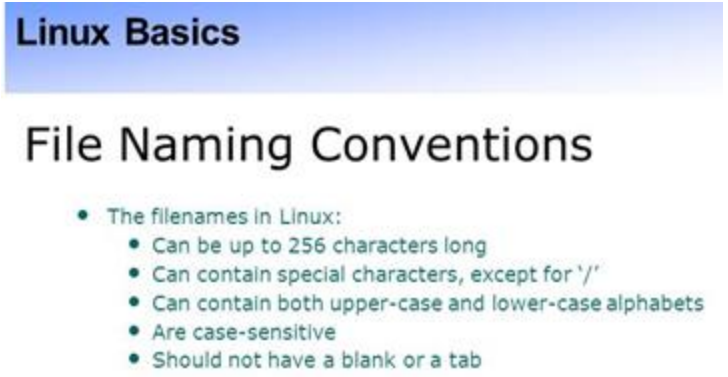
DEVOPS EASY LEARNING

File and Directory Naming Convention in Linux or In IT

How can I name file and directories in Linux or in IT

- Windows does not really care about the naming convention
- This is a big deal in Linux and it
- Let say I want to create a directory call **"apha website development for covid19"** for instance
- You need do separate each word either by underscore **(_)** or hyphen **(-)** or start each word by capital letter
  - apha_website_development_for_covid19
  - apha-website-development-for-covid19
  - AphaWebsiteDevelopmentForCovid19
  - aphaWebsiteDevelopmentForCovid19





DEVOPS EASY LEARNING

What is Case Sensitive in Linux Or IT?

Case Sensitive in Linux

The Linux operating system provides you with multiple commands that you can run in the terminal to find a specific file. Although, most of these commands are case sensitive, meaning that you need to know the exact name of your file and whether it is in **lower-case** or **upper-case** letters or a combination of both

Linux is case sensitive because **'a'** and **'A'**

"Computer" and **"computer"** are two different words because the **"C"** is uppercase in the first example and lowercase in the second example

DEVOPS EASY LEARNING

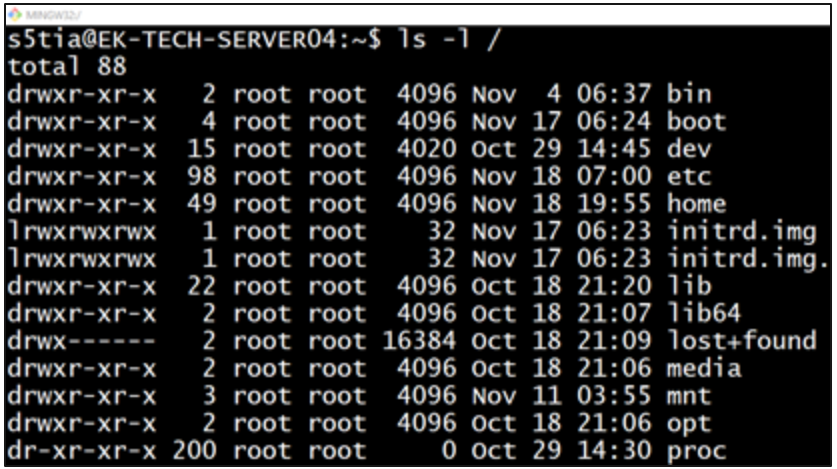
What is a Terminal?

A Terminal

It is an interface in linux that a user interact with the operating system and received an output.

Here are couples of nickname or aliases in linux environment that when pronounced refer to the terminal**:**

- back environment
- black room
- dark room
- back zone



DEVOPS EASY LEARNING

What is a Command and Option In Linux?

A Command:

- In Linux the word ***"command"*** is an instruction given by a user telling a computer to do something.
- For example when you type the ls command, you are instructing the computer to list contains of a directory.
- Everything that we type on a linux terminal is a command

An Option

- We use option to change the default behavior or output of a command
- Let take the **ls** command as an example. This command has multiple option base on what you want to accomplish
- You can Alway check Linux manual to get all option that a command can take
- We will teach you how to the want that are widely used

Options:

- ll,
- ls -a,
- ls -ltr,
- ls -la -lh,
- ls -lh,
- ls -h,

DEVOPS EASY LEARNING

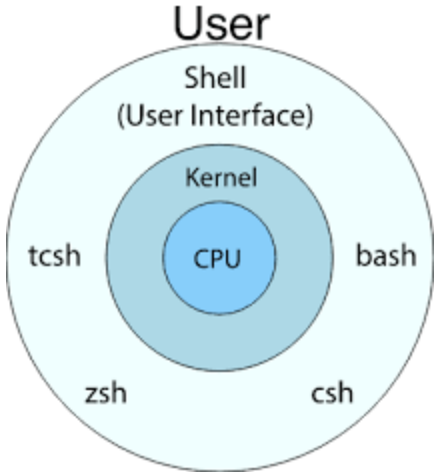
Kernel and Shell In Linux

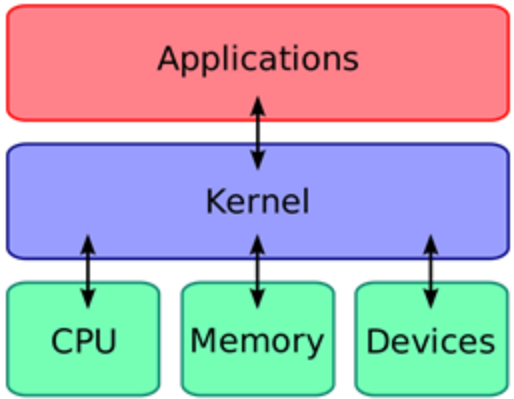
Kernel

- **kernel** is the main component of a Linux operating system (OS) and is the core interface between a computer's hardware and its processes
- It provides the interfaces needed for users and applications to interact with the computer.
- It launches and manages applications.
- It manages the underlying system hardware devices.

Shell

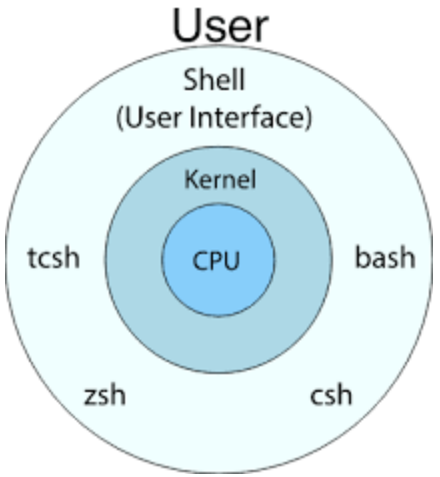
- **kernel** is the main component of a Linux operating system (OS)
- It is the core interface between a computer's hardware and its processes





DEVOPS EASY LEARNING

Type of Shell In Linux



Types of Shell

There are two main shells in Linux:

The **Bourne Shell** (The prompt for this shell is $) and The **C shell**.

Bourne Shell derivatives are listed below:

- Bourne shell (sh)
- Korn shell (ksh)
- Bourne Again Shell also knew as bash (most popular)

Why bash is a default shell in Linux?

- Bash shell has some feature that other shell don't have such as:
  - Command aliases
  - Auto completion
  - Input/output redirection
  - Shell variables for customizing your environment
  - Command-line editing (using vi)
  - Access to previous commands
  - Integer arithmetic
  - Arithmetic expressions
- It is an enhanced version of the Bourne shell. It was created by Steven Bourne

root@EK-TECH-SERVER04:~# cat /etc/shells

/bin/sh

/bin/bash

/bin/rbash

/bin/dash

/usr/bin/tmux

/usr/bin/screen

DEVOPS EASY LEARNING

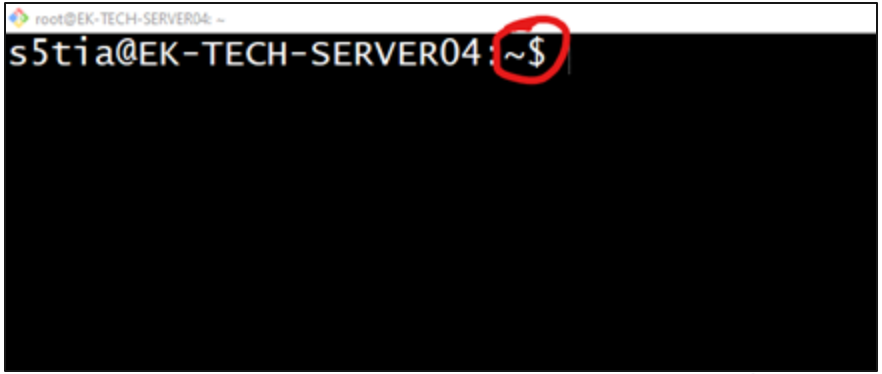
Root User and Regular User In Linux

Root user or admin user or power user or superuser

- The superuser, or root (also known as admin account), is a unique user account used for system administration purposes on Linux
- It is typically has the **highest access rights** on the system.
- It the default account that get created when you launch or install a Linux operating system

Regular user or non root user or normal user

- Normal users on Linux run with reduced permissions or limited **privileges**
- There are some task that a regular user cannot do such as:
  - Install, remove and update packages
  - Create, delete and modify users and groups
  - Manage file system permissions for users and groups
- We can give a regular user a privilege to execute command as a root user using **sudo**

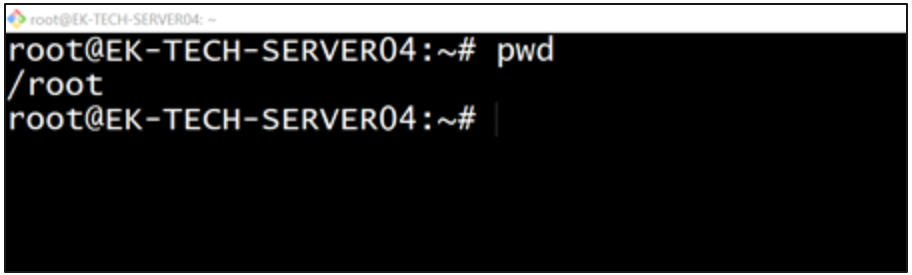


DEVOPS EASY LEARNING

User and Root Home directory In Linux

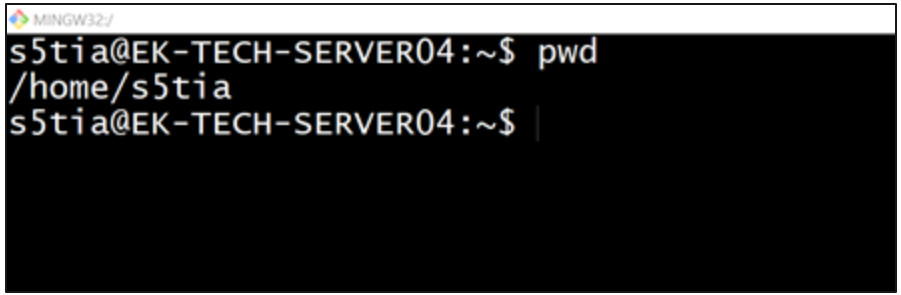
Root user home directory

- It is where a root files and directories are stored
- When you login as a regular user, your default home directory is **/root**
- A root home directory is located at **/root**



Regular home directory

- It is where users files and directories are stored
- When you login as a regular user, your default home directory is **/home/[user_name]**
- A user home directory is located at **/home/[user_name]**



DEVOPS EASY LEARNING

Help in Linux

How can I get Help in Linux?

I need help

- Linux has a lot of command and some time, it is very hard to remember all the commands
- You will need to google some Linux commands base on what you what to accomplish
- All Linux commands are in Google and you just need to know how to google and use them
- You will no memorize all the commands as one. It will takes time
- It you want to master linux commands quickly, please **practice everyday** and you will get used to those commands
- **Even at work**, we google command base on what we want to accomplish
- Please **learn how to google** because google will alway provide answer base on what you ask.
- You need to master how to google to be more efficient in this journey

DEVOPS EASY LEARNING

How can I get Help in Linux?

man

- It stands for manual in Linux.
- It is a command line tool that we use to get help about specific command and their options in Linux
- It is Linux dictionary where we can find help about all Linux commands
- Please press **q** on the keyboard to to exit out of the command

man command

Example: # **man** [command]

man ls

Options: # **man** test

DEVOPS EASY LEARNING

How can I get Help in Linux?

info

- It is used to get help about a specific command in Linux
- It is similar to the man command
- Please press **q** on the keyboard to to exit out of the command

info command

Example: # **info** [command]

info ls

Options: none

DEVOPS EASY LEARNING

DevOps Linux Commands

What are some command that you need as a feature DevOps?
As DevOps, You do not need to know all linux Commands because you are not a **Linux system admin**. Linux help you to get your job done and you need to know the basic or the foundation.

Day to day activity commands for DevOps Engineer

- ls
- cd
- pwd
- touch
- mkdir
- rm
- man
- vim
- exit
- cat
- less or more or less
- grep
- cp
- mv
- clear or ctrl + l
- ctrl + c
- sudo
- history

DevOps use This commands some time

- ps and top
- free
- du
- unzip
- tar
- awk
- find
- yum or apt
- chmod
- echo
- scp
- Ping
- crontab

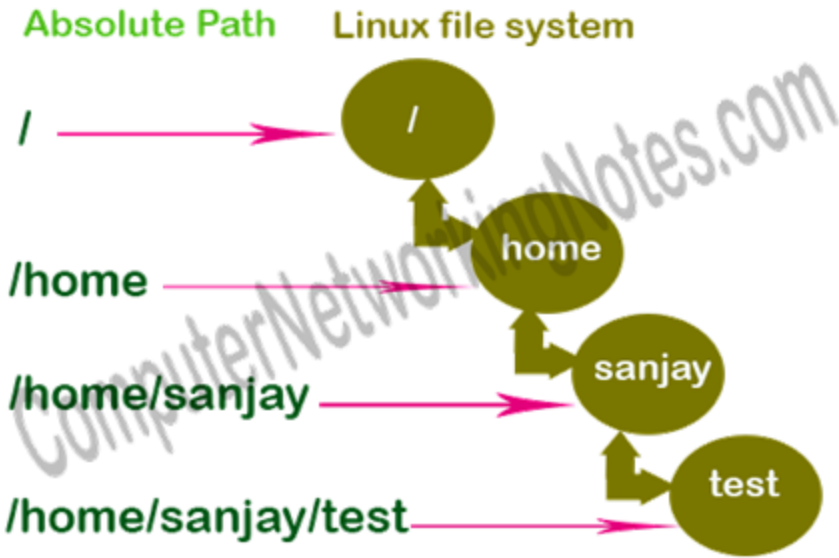
These are the Linux commands that you need to know to get your job done as a DevOps Engineer

DEVOPS EASY LEARNING

What is a Relative Path?

Relative path

Relative path starts from the **current directory** and goes up to the actual object. Relative path depends on the current directory. When we change the directory, relative path also changes

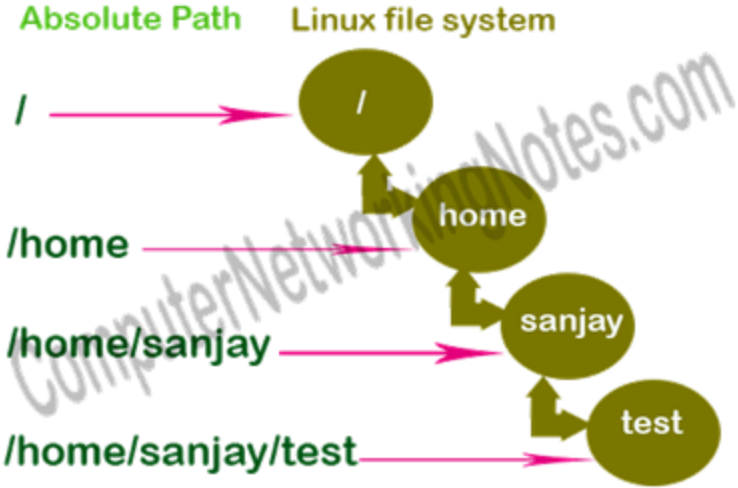


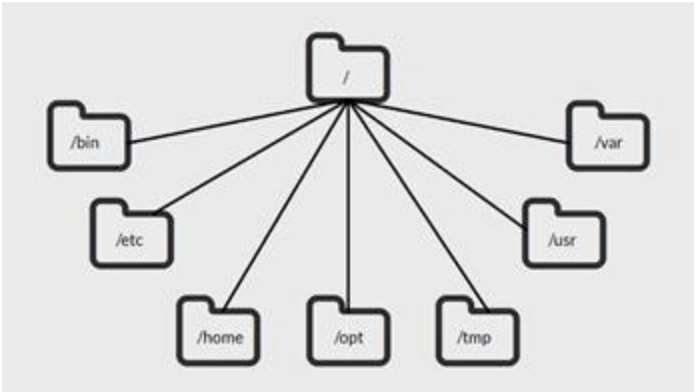
DEVOPS EASY LEARNING

What is an absolute Path?

Absolute Path

Absolute path starts from the directory **root (/)** and goes up to the actual object (file or directory). It contains the names of all directories that come in the middle of the directory root and the actual object. In this, name of the parent directory is written in the left.





DEVOPS EASY LEARNING

Absolute and Relative Path

Command	Description	Path
cat ./dir1/abc	Print the contents of the file abc.	Use relative path. *Include current directory*
cat dir1/abc	Print the contents of the file abc.	Use relative path. *Skip current directory*
cd ./dir1	Change current directory to dir1	Use relative path.
cd ..	Change current directory to parent directory	Use relative path.
cd /home/sanjay/dir1	Change current directory to dir1	Use absolute path.
cp ./dir1/abc .	Copy the file abc in current directory	Use relative path.
./simple.sh	Run script from current directory	Use relative path

Linux Basic Commands

Linux Commands

clear

- It is use to clear you screen
- You also ctrl + l as a shortcut
- You will used this command daily as a DevOps

clear command

Example: # **clear**

ctrl + l

Options: none

DEVOPS EASY LEARNING

Linux Commands

history

- The **history** command is used to view the previously executed command.
- It is one of the most basic and frequently used commands in Linux.
- You will used this command daily as a DevOps

history command

Example: # **history**

Options: none

DEVOPS EASY LEARNING

Linux Commands

exit

- The **exit** command in linux is used to exit out of the shell where you are currently working
- When you are done with you hand on, instead of just closing your terminal, use the exit command instead
- It is one of the most basic and frequently used commands in Linux.
- You will used this command daily as a DevOps

exit command

Example: # **history**

Options: none

DEVOPS EASY LEARNING

Linux Commands

ls

- The **ls** command stands for list directory content
- and it is used to list files or directories (folders) in Linux.
- It is one of the most basic and frequently used commands in Linux
- You will used this command daily as a DevOps

ls command

Example: *# **ls** ----- will list content of current directory*

*# **ls** [directory_name] ---will list the content of [directory_name]*

Options:

- *ll*
- *ls -la*
- *ls -la -lh*
- *ls -lh*
- *ls -lh*

- **ll** = long listing
- **ls -la** = long listing with hidden files
- **ls -la -lh** = long listing with hidden files and human readable such Megabyte, Gigabyte etc.
- **ls -lh** = long listing human readable such Megabyte, Gigabyte etc.
- **ls -h** = short listing human readable such Megabyte, Gigabyte etc.

DEVOPS EASY LEARNING

Linux Commands

cd

- The command **cd** stands for change directory in Linux.
- It is used to move from directory to another
- It is one of the most basic and frequently used commands in Linux.
- You will used this command daily as a DevOps

cd command

Example: *# **cd** [directory_name or path]*

*# **cd /root/tmp** ----- to move to tmp directory*
*# **cd ~/Download** ----- to move to download directory*
*# **cd /home/Download** ----- to move to download directory*

Options:

- cd
- cd ~
- cd -
- cd ..
- cd ../..
- cd ../../..
- cd ../../../..
- cd ../../../../../../../../../../../../../..

Options:

- **cd** = move to home
- **cd ~** = move to home
- **cd -** = move to the previous directory
- **cd ..** = move 1 step back
- **cd ../..** = move 2 steps back
- **cd ../../..** = move 3 steps back
- **cd ../../../..** = move 4 steps back
- **cd ../../../../../../../../../../../../../..** = move **n** steps back

DEVOPS EASY LEARNING

Linux Commands

touch

- The **touch** command is commonly used to create an empty file in Linux
- It is one of the most basic and frequently used commands in Linux.
- You will used this command daily as a DevOps

Example: # **touch** [file_name]

touch engineering.txt

Ps: *you don't really need to put the file extension because Linux do not care about file extension if it as a regular file*

Options: none

DEVOPS EASY LEARNING

Linux Commands

touch

- The **mkdir** stands for make directory in Linux. It allows users to create new directories or folder.
- It is one of the most basic and frequently used commands in Linux.
- You will used this command daily as a DevOps

mkdir command

Example: # **mkdir** [directory_name]

mkdir manager

Options: # **mkdir -p** [path]

mkdir /tmp/tia ⇒ This will help create a directory with the path.

The parent directory must exist first

DEVOPS EASY LEARNING

Linux Commands

cat

- The **cat** command is used to view or to open the content of a file in Linux
- It is useful to view a content of the file that **doesn't have** lot of content
- It is one of the most basic and frequently used commands in Linux.
- You will used this command daily as a DevOps

Example: # **cat** [directory_name]

cat [file_path]

cat logs.txt

Options: none

DEVOPS EASY LEARNING

Linux Commands

less

- The **less** command is used to view or to open the content of a file in Linux
- It is useful to view a content of the file that **have** a lot of content
- You will type **q** to exit out of the file
- It is one of the most basic and frequently used commands in Linux.
- You will used this command daily as a DevOps

Example: # **less** [directory_name]

less [file_path]

less logs.txt

Options: none

DEVOPS EASY LEARNING

Linux Commands

more

- The **more** command is used to view or to open the content of a file in Linux
- It is useful to view a content of the file that **have** a lot of content
- You will exit out of the file **automatically** at the end of the fole
- It is one of the most basic and frequently used commands in Linux.
- You will used this command daily as a DevOps

more command

Example: # **more** [directory_name]

more [file_path]

more logs.txt

Options: none

DEVOPS EASY LEARNING

Linux Commands

nl

- The **nl** command is used to view or to open the content of a file in Linux
- It is useful when you want to display the line numbers
- It is one of the most basic and frequently used commands in Linux.
- You will used this command daily as a DevOps

Example: # **nl** [directory_name]

nl [file_path]

nl logs.txt

Options: none

DEVOPS EASY LEARNING

Linux Commands

cp

- The **cp** command is used to copy files or directory in Linux from one location to another.
- It is one of the most basic and frequently used commands in Linux.
- You will used this command daily as a DevOps

Example: # **cp -r** [directory_name or directory_name]

cp [file_path or directory_path]

cp logs.txt /tmp ⇒ copy logs.txt to tmp directory

cp -r ~/home/tia/manager /tmp ⇒ copy manager in tia home directory to the tmp directory

Options: # **cp -r**

r = recursive. It means copy the parent and all the subdirectories

DEVOPS EASY LEARNING

Linux Commands

cp

- The **cp** command is used to copy files or directory in Linux from one location to another.
- It is one of the most basic and frequently used commands in Linux.
- You will used this command daily as a DevOps

cp options

Options:

 # cp -r manager /tmp ⇒ copy the directory manager to the tmp directory

 # cp * /tmp ⇒ copy everything in my current working directory to the tmp directory

 # cp -r manager/* /tmp ⇒ copy everything in manager to the tmp directory

 # cp *.txt /tmp ⇒ copy all files that ends with **.txt** in in my current working directory to tmp

 # cp *.sh /tmp ⇒ copy all files that ends with **.sh** in in my current working directory to tmp

 # cp *a /tmp ⇒ copy all files that ends with **a** in in my current working directory to tmp

 # cp a* /tmp ⇒ copy all files that starts with **a** in in my current working directory to tmp

r = recursive. It means copy the parent and all the subdirectories

DEVOPS EASY LEARNING

Linux Commands

cp

- The **cp** command is used to copy files or directory in Linux from one location to another.
- It is one of the most basic and frequently used commands in Linux.
- You will used this command daily as a DevOps

cp options with path

Options:

 # cp -r ~/Desktop/manager /tmp ⇒ copy the directory manager from Desktop to the tmp directory

 # cp -r ~/home/tia/* /tmp ⇒ copy everything in my tia directory to the tmp directory

 # cp -r manager/* /tmp ⇒ copy everything in manager directory to the tmp directory

 # cp -r manager/* . ⇒ copy everything in manager directory and put in my current working directory

 # cp /tia/script/*.txt /tmp ⇒ copy all files that ends with **.txt** in script directory to the tmp directory

 # cp *.sh /home/Desktop/scripts ⇒ copy all files that ends with **.sh** in the script directory on my desktop

 # cp ~/*.sh /home/Desktop/scripts ⇒ copy all files that ends with **.sh** in tom home directory on my desktop

 # cp /root/*a /tmp ⇒ copy all files that ends with **a** in root home directory to tmp directory

r = recursive. It means copy the parent and all the subdirectories

DEVOPS EASY LEARNING

Linux Commands

cp, and mv

The command **mv** stands for move. The mv is used to move one or more files or directories from one place to another. It is also used to rename files and directories

*Example : # **mv** file1 file2 ------ this will rename file1 to file2*

*# **mv** file1 folder1 ----this will move file1 to directory folder1*

The **cp** command is used to copy files or directory in Linux from one place to another.

*Example : # **cp** file1 folder1 ------this copy file1 into directory folder1*

Options:

- cp -r
- cp * /tmp
- cp test/* /tmp
- cp *.txt /tmp
- cp *.sh /tmp
- cp *a /tmp
- cp a* /tmp

DEVOPS EASY LEARNING

Linux Commands

rm , rm -rf and echo

The **rm (remove)** command is used to delete files only

The **rm -rf** command is used to delete files and or directories

*Example : # **rm** [file_name]*

*Example : # **rm -rf** [file_name] or # **rm -rf** [directory_name]*

The **echo** is used to display or print text (string) on the terminal

*Example : # **echo** ["text_to_display"]*

Options:

- rm -rf
- rm -rf *
- rm -rf *.sh
- rm -rf *.txt
- rm -rf *a
- rm -rf a*

DEVOPS EASY LEARNING

Linux Commands

env, export and unset

The **env** command in Linux is used to print environment variables or system variables
*Example : # **env***

The **export** command is a built-in utility of Linux Bash shell. It is used to set the environment variables or system variables in Linux
*Example : # **export*** [env_name]=[value]

The **unset** command is a built-in utility of Linux Bash shell. It is used to delete or remove the environment variables or system variables in Linux
*Example : # **unset*** [env_name]

Options:

- export DB_PASSWORD=”DeVops@”
- export DB_USER_NAME=”admin”
- echo $DB_PASSWORD
- echo $DB_USER_NAME

- unset DB_PASSWORD
- unset DB_USER_NAME
- echo $DB_PASSWORD
- echo $DB_USER_NAME

DEVOPS EASY LEARNING

Linux Commands

head and tail

The **head** command in Linux is used to view the first 10 lines of a file by default

*Example : # **head** [file_name]*

The **tail** command in Linux is used to view last 10 lines of a file by default

*Example : # **tail** [file_name]*

Options:

- head -n20
- head -n5
- head -n2
- tail -n20
- tail -n5
- tail -n2

DEVOPS EASY LEARNING

Linux Commands

wc, whoami and who

The **wc** command is used to count words, lines and characters in the file
*Example : # **wc** [argument]*

The **whoami** is a compound of the words "Who am I?" and prints the name of the user that is currently login
*Example : # **whoami***

The **w or who** command will display the name of all the users that are currently login
*Example : # **w***

***su** is the command in linux used to switch for one user to another or also run command on behalf of other user*
*Example # **su** - [username]*
*# **su** - **root***

Options:

- wc -l = number of lines
- wc -w = number of words
- wc -c = number of characters

DEVOPS EASY LEARNING

Linux Commands

Uname, hostname, nproc , arch

The **uname** command is used to display the system information such as the kernel version, the system architecture etc.

*Example : # **uname***

The **hostname** command is used to display the hostname name of the server.

*Example : # **hostname***

The **arch or getconf LONG_BIT** is used to get the current computer architecture.

*Example : # **arch***

The **nproc** command is used to get the the number of CPU that the server is using.

*Example : # **nproc***

Options:

- uname -a
- uname -r
- uname -m

DEVOPS EASY LEARNING

Linux Commands

append and redirect

“>“: Overwrites the existing file, or creates a file if the file of the mentioned name is not present in the directory.

“>>“: Appends the existing file, or creates a file if the file of the mentioned name is not present in the directory.

While making modifications in a file and you want to overwrite the existing data, then use the **“>”** operator. If you want to append something to that file, use the **“>>”** operator

The **“>”** is an output operator that overwrites the existing file, while **“>>”** is also an output operator but appends the data in an already existing file. Both operators are often used to modify the files in Linux.

> file.txt: to empty a file **cat >> demo.txt** **cat > demo.txt**

ctrl + d: to save

DEVOPS EASY LEARNING

Linux Commands

tar, zip and unzip

The **zip** command is used to compress a file in Linux
*Example : # **zip -r my_file.zip tia-devops***

The **unzip** command is used to extract the zip file in Linux
*Example : # **unzip my_file.zip***

The **tar** command is used to compress a file in Linux
Example : # tar cf [file_name.tar] [file_name]
tar -cpzf tia-devops.tar tia-devops
tar -cpzf tia-devops.gz tia-devops

tar -xpzf tia-devops.tar
tar -xpzf tia-devops.gz

- **--c** = create new file (overwrites old file)
- **--v** = verbose
- **--x** = extract
- **--p** = preserve permissions
- **--z** = compress
- **--f** = filename (very important)

Naming Files with time = filename-$(date +%F-%T)

Options:
- free -h
- r = recursive

PS: use -r for zip to avoid zipping and empty directories

DEVOPS EASY LEARNING

Linux Commands

scp command

The **scp** is a command-line utility that allows you to copy files and directories from one server to another

Copy a file or a directory from a remote server

scp -r [username]@[server IP address or domain name]:[remote path] [destination path]

scp -r tia@server4.anomicatech.com:/tmp/devops /home/tia/Download/devops-course

scp -r tia@server4.anomicatech.com:/tmp/devops .

Copy a file or a directory from local to a remote server

scp -r *[directory name or path] [username]@[server IP address]:[destination path][directory name]*

scp -r *~/Download/devops-course tia*@server4.anomicatech.com*:/tmp/devops*

scp -r *devops-course tia*@server4.anomicatech.com*:/tmp/devops*

Copy all directories and files from a remote server

scp -r *tia*@server4.anomicatech.com*:/tmp/* /home/tia/Download/devops-course*

scp -r *tia*@server4.anomicatech.com*:/tmp/** .

DEVOPS EASY LEARNING

Linux Commands

vimdiff , dif and alias

The **vimdiff** is a command-line utility that allows you to compare two files line by line.

*Example : # **vimdiff** [file1] [file2]*

The **alias** command is like custom shortcuts used to represent a command (or set of commands) executed with or without custom options

*Example : # **alias** [alias_name]=[command]*

*# **alias** c="clear"*

The ***diff*** is a command-line utility that allows you to compare two files

*Example : # **diff** [file1] [file2]*

vim ~/.bashrc

source ~/.bashrc

alias c='clear'

alias u='uname -a'

alias g='git'

Click here to learn more

DEVOPS EASY LEARNING

Linux Commands

free , df and du

The **free** command provides information about the total amount of the physical and swap memory, as well as the free and used memory.

*Example : # **free***

The **df** (short for disk free) is used to show the amount of free disk space available on Linux

*Example : # **df***

The **du** command stand for disk usage, is used to estimate file space usage

*Example : # **du***

 # du -sh tia-devops

Options:

- free -h
- df -h
- du -h
- du -sh
- s = size
- h = human readable (G, M)

DEVOPS EASY LEARNING

Linux Commands

The **top** command is used to check all current running processes in Linux.

ps: *to exit the* ***top*** *command, press* ***q*** *in your keyboard*

Example : # ***top***

The **ps** command is used to check all current running processes in Linux.

Example : # ***ps***

The **kill** command is used to kill or terminate a process in Linux.

Example : # ***kill -9*** *[process_id]*

The **uptime** command is used to check how long the system has been up.

Example : # ***uptime***

free , df and du

Options:

- kill -9
- kill -15
- ps -aux
- uptime -p

kill -9 = Kill the parent and the children processes at once

kill -15 = Kill the children processes before killing the parent process (soft kill)

DEVOPS EASY LEARNING

Vim Editor in Linux

Vim In Linux

The **vim** command is used to edit or add contain into **files**, vim can also help create a file that does not exist yet and add content inside.

How to install vim?

*Example : **# yum install vim -y or apt install vim***

How to open or create a file in vim?

*Example : **# vim** [file_name]*

We have 3 mode in vim:

- Command line mode
- Insert mode
- And visual mode

DEVOPS EASY LEARNING

Vim In Linux

Options:

- **vim**: open vim
- **q!**: quit a file without saving
- ***esc***: command line mode
- ***i***: insert mode
- ***o***: insert mode
- ***:w***: write

Options:

- ***yy or cc***: copy a line the whole line
- ***p***: paste
- ***:wq or :x***: write and quit
- ***:wq!***: force write and quit
- ***u***: undo
- ***ctrl +r***: redo
- ***shift + v***: visual mode

Options:

- ***dd***: delete a line
- ***y***: copy
- ***[[***: move to the beginning of the file
- ***]]***: move to the end of the file
- ***:set nu***: enable line number
- ***:set nonu***: disable line number
- ***:set background***: change background
- ***:set noai*:** to disable auto-indent
- ***:set background*:** dark to set a dark background

DEVOPS EASY LEARNING

Users and Groups Management

User and Group Management

useradd , userdel, id, groups and passwd

The **useradd** command is used to create a new user account in Linux
*Example : # **useradd** [user_name]*

The **userdel** is used to delete users in Linux
*Example : # **userdel** [user_name]*

The **id** command is a command-line utility that prints the real and effective user and group IDs
*Example : # **id** [user_name]*

The **groups** command is used to check the group that the user belongs to in Linux
*Example : # **groups** [user_name]*
The **passwd** command is used assign or reset a user account password in Linux
*Example : # **groups** [user_name]*

Options:

- userdel -r = delete a user with is home directory
- PS: Please avoid deleting a user with his home directory

DEVOPS EASY LEARNING

User and Group Management

groupadd and groupdel

The **groupadd** command is used to creates a new group in Linux

*Example : # **groupadd** [group_name]*

*# **groupadd** sysadmin*

*# **groupadd** finance; groupadd HR; groupadd manager*

The **groupdel** command is used to delete a existing group in Linux.

PS: This will delete all members of the group.

*Example : # **groupdel** [group_name]*

*# **groupdel** sysadmin*

*# **groupdel** finance; groupadd HR; groupadd manager*

Options: None

DEVOPS EASY LEARNING

User and Group Management

groupadd and groupdel

The **usermod** command is a command in Linux that is used to change the properties of a user in Linux through the command line or to modify a user account

*Example : # **usermod** [options] [user_name]*

Options:

usermod -aG [group_name] [username] = Add a user to a secondary group or supplementary group

usermod -s /sbin/nologin [username] = Modify a user default shell

usermod -L = Lock a user account

usermod -U = Unlock a user account

usermod -l (new username + old username) = Change the login name

usermod -c (usermod -c "Tom Smith" tom) = Add a full name or comment

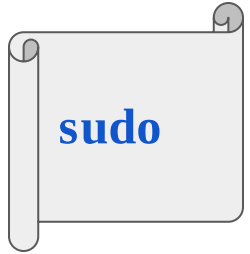
Options:

- -aG
- -L
- -U
- -s
- -c

usermod -s /sbin/nologin -aG manager,finance,sysadmin -c "Tim Smith" tim

DEVOPS EASY LEARNING

User and Group Management



The **sudo** command is used to allow a permitted user to execute a command as the superuser (root)

*Example : # **sudo** [command]*

The **visudo** command is used to to open the sudoers file

*Example : # **visudo***

 # sudo -i

username ALL=(ALL:ALL) NOPASSWD:ALL = This user will run all commands as root without a sudo password

%group_name ALL=(ALL:ALL) ALL NOPASSWD:ALL = All users in this group will run all commands as root without a sudo password

username ALL=(ALL:ALL) ALL = This user will run all commands as root with a sudo password

%group_name ALL=(ALL:ALL) ALL = All users in this group will run all commands as root with a sudo password

PS: do not use vim to open or edit the sudoers file **(It will locked the system)** === >> **vim /etc/sudoers**

How to change the default sudo timeout?

sudo visudo

Defaults:USER timestamp_timeout=30

Defaults:USER timestamp_timeout=0

The default is 5 minutes

DEVOPS EASY LEARNING

User and Group Management

passwd and chage

The **passwd** command is used to create or update existing passwords of users

*Example : # **passwd** [user_name]*

*# **passwd --expire [user_name]** = expire a user password*

*# **passwd -e [user_name]** = expire a user password*

*# **passwd -e [user_name]** = expire a user password*

*# **chage -l [user_name]** = check if the user password is expired*

*# **passwd -S [user_name]** = check if the user password is locked*

*# **passwd -S tom***

*tom **L** 08/15/2022 0 99999 7 -1 (**L means the** Password locked.)*

*tom **P** 08/15/2022 0 99999 7 -1: it is unlock (**P means the** Password locked.)*

*If the password starts with **!** in /etc/shadow, the account is locked*

***passwd -l [user_name]**= lock the user account*

***passwd -u [user_name]** = unlock the user account*

***usermod -L [user_name]** = lock the user account*

***usermod -U [user_name]** = unlock the user account*

DEVOPS EASY LEARNING

Package manager in Linux

Package Manager In Linux

apt and yum

apt command in Linux is used to install and delete package in Ubuntu or Debian operating system

*Example : # **apt** install -y [package name]*

*# **apt-get** install -y [package name]*

*# **apt** remove -y [package name]*

yum command in Linux is used to install and delete package in CentOS or RedHat operating system

*Example : # **yum** install -y [package name]*

*# **yum** remove -y [package name]*

The **which** command is used to identify the location of a given executable that is executed when you type the executable name (command) in the terminal prompt.

*Example : **# which** [command name or package name]*

Options:

- apt-get install -y
- apt install -y
- apt remove -y
- yum install -y
- yum remove -y

DEVOPS EASY LEARNING

Service In Linux

service and systemctl

The **systemctl** command is used manage services or daemons in Linux.

*Example : # **systemctl** **start** [package name]*

*# **systemctl** **stop** [package name]*

*# **systemctl** **status** [package name]*

*# **systemctl** **restart** [package name]*

*# **systemctl** **enable** [package name]*

The **service** command is used manage services or daemons in Linux.

*Example : # **service** [package name] **start***

*# **service** [package name] **stop***

*# **service** [package name] **status***

*# **service** [package name] **restart***

*# **service** [package name] **enable***

A **process** is simply an application or a script which can be running in the foreground or the background. **Service or systemctl** is a command which allows you start, stop or restart services running in the background

DEVOPS EASY LEARNING

Service In Linux

The **systemctl** command is used manage services or daemons in Linux.

Example : ***# systemctl*** ***start*** *[package name]*

 # systemctl ***stop*** *[package name]*

 # systemctl ***status*** *[package name]*

 # systemctl ***restart*** *[package name]*

 # systemctl ***enable*** *[package name]*

The **service** command is used manage services or daemons in Linux.

Example : ***# service*** *[package name]* ***start***

 # service *[package name]* ***stop***

 # service *[package name]* ***status***

 # service *[package name]* ***restart***

 # service *[package name]* ***enable***

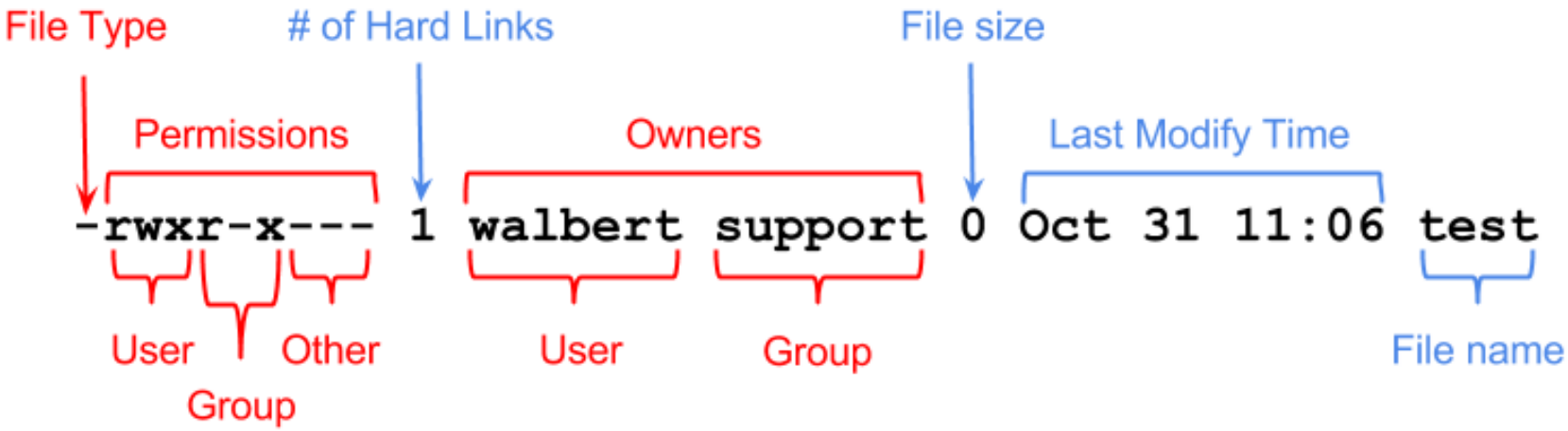
service and systemctl

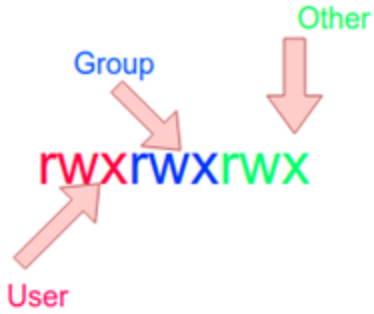
```
apt update -y
apt install wget -y
apt install unzip -y
apt install apache2 -y
cd /var/www/html/

wget https://linux-devops-course.s3.amazonaws.com/WEB+SIDE+HTML/covid19.zip
unzip covid19.zip
cp -R covid19/* .
rm -rf covid19.zip
rm -rf covid19

systemctl start apache2
systemctl status apache2
systemctl enable apache2

wget https://linux-devops-course.s3.amazonaws.com/articles.zip
unzip articles.zip
cp -R articles/* .
rm -rf articles.zip
rm -rf articles
```

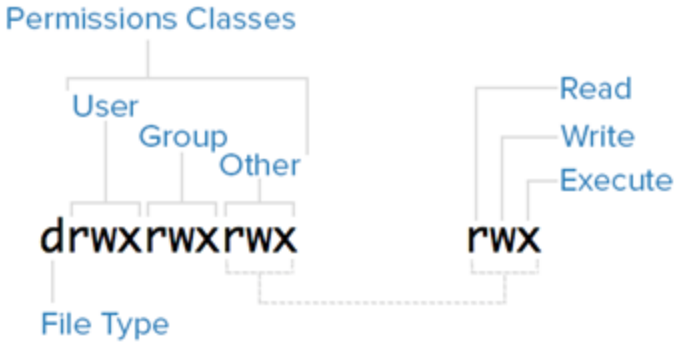
DEVOPS EASY LEARNING

Linux File and Directories Permission

File and directory permissions

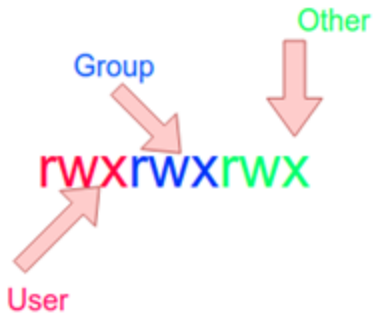






DEVOPS EASY LEARNING

File and directory permissions



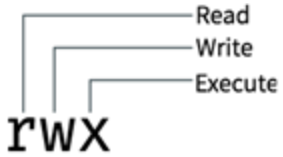
Ownership of Linux files

Every file and directory on your Unix/Linux system is assigned 3 types of the owner, as given below.

- **User:** A user is the owner of the file. By default, the person who created a file becomes its owner. Hence, a user is also sometimes called an owner.
- **Group:** A group can contain multiple users. All users who are in the same group will have the same Linux group permissions access to the file. Suppose you have a project where a number of people require access to a file. Instead of manually assigning permissions to each user, you could add all users to a group, and assign group permission to file such that only these group members and no one else can read or modify the files.
- **Other:** Other Any other user who has access to a file. This person has neither created the file nor belongs to a user group that could own the file. Practically, it means everybody else. Hence, when you set the permission for others, it is also referred as set **permissions for the world**.

DEVOPS EASY LEARNING

File and directory permissions

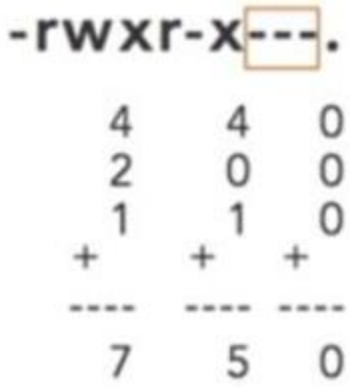


Read	=	4
Write	=	2
Execute	=	1

Permissions

Every file and directory in your UNIX/Linux system has following 3 permissions defined for all the 3 owners discussed above.

- **Read:** This permission give you the authority to open and read a file. Read permission on a directory gives you the ability to lists its content.
- **Write:** The write permission gives you the authority to modify the contents of a file. The write permission on a directory gives you the authority to add, remove and rename files stored in the directory. Consider a scenario where you have to write permission on file but do not have write permission on the directory where the file is stored. You will be able to modify the file contents. But you will not be able to rename, move or remove the file from the directory.
- **Execute:** In Windows, an executable program usually has an extension .exe and which you can easily run. In Unix/Linux, you cannot run a program unless the execute permission is set. If the execute permission is not set, you might still be able to see/modify the program code, but not run it.



DEVOPS EASY LEARNING

File and directory permissions

Read	=	4
Write	=	2
Execute	=	1

Changing file/directory permissions with 'chmod' command

- We can use the chmod command which stands for change mode.
- Using the command, we can set permissions (read, write, execute) on a file/directory for the owner, group and the world.

Syntax:

chmod [permissions] [file/directory name]

There are 2 ways to use the command

- Absolute(Numeric or Octal) Mode
- Symbolic Mode

This is what you will used the a lot in the real world

chmod [scrpt_name]

DEVOPS EASY LEARNING

File and directory permissions

Octal Value	Symbolic Value	Result
777	a+rwx	rwxrwxrwx
755	u+rwx,g=rx,o=rx	rwxr-xr-x
644	u=rw,g=r,o=r	rw-r--r--
700	u=rwx,g-rwx,o-rwx	rwx------

DEVOPS EASY LEARNING

File and directory permissions

Original	Symbolic Value	Result
rw-r--r--	+x	rwxr-xr-x
rwxrwxrwx	g=w,o=r	rwx-w-r--
rwxr-xr-x	o-rx	rwxr-x---
rwxrwxrwx	a-x	rw-rw-rw-

File and directory permissions

	User (u)	Group (g)	Others (o)
Read (r)	4	4	4
Write (w)	2	2	2
Execute (x)	1	1	1
Total	7	7	7

File and directory permissions

Number	Permission Type	Symbol
0	No Permission	---
1	Execute	--x
2	Write	-w-
3	Execute + Write	-wx
4	Read	r--
5	Read + Execute	r-x
6	Read +Write	rw-
7	Read + Write +Execute	rwx

DEVOPS EASY LEARNING

Crontab in Linux

Crontab or Cron Job

Cron Job

- It is a tool used in Linux to schedule tasks.
  This task can be a single command or a whole script
- The crontab is used to run a specific task at a regular interval.
- Crontab executes jobs automatically in the backend at a specified time and interval.

To schedule a crontab

crontab -e

List a cron jobs

crontab -l

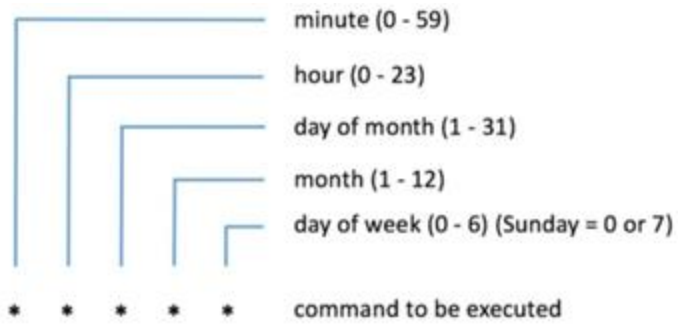
To delete a cron job

crontab -r

Click here to learn more

crontab guru
https://crontab.guru/

field	allowed values
minute	**0-59**
hour	**0-23**
day of month	**1-31**
month	**1-12** (or names; see example below)
day of week	**0-7** (**0** or **7** is Sunday, or use names; see below)



DEVOPS EASY LEARNING

Crontab or Cron Job

Cron Job

- It is a tool used in Linux to schedule tasks.
  This task can be a single command or a whole script
- The crontab is used to run a specific task at a regular interval.
- Crontab executes jobs automatically in the backend at a specified time and interval.

To schedule a crontab

crontab -e

List a cron jobs

crontab -l

To delete a cron job

crontab -r

Click here to learn more

crontab guru
https://crontab.guru/

Special string	Meaning
@reboot	Run once, at startup.
@yearly	Run once a year, "0 0 1 1 *".
@annually	(same as @yearly)
@monthly	Run once a month, "0 0 1 * *".
@weekly	Run once a week, "0 0 * * 0".
@daily	Run once a day, "0 0 * * *".
@midnight	(same as @daily)
@hourly	Run once an hour, "0 * * * *".

DEVOPS EASY LEARNING

Crontab or Cron Job

Schedule a cron to execute on every 10 minutes.

If you want to run your script on a 10 minutes interval, you can configure it like below. These types of crons are useful for monitoring.

***/10 * * * * sh /scripts/monitor.sh**

Schedule a cron to execute every minute.

Generally, we don't require any script to execute every minute but in some cases, you may need to configure it.

*** * * * * sh /scripts/script.sh**

This will echo Hello every minutes

*** * * * * echo "Hello" >> /tmp/test.txt**

crontab -e

crontab -r

Schedule tasks to execute on Weekly (@weekly).

@weekly timestamp is similar to 0 0 1 * mon. It will execute a task in the first minute of the week. It may be useful to do weekly tasks like the cleanup of the system etc.

@weekly sh /bin/script.sh

Schedule tasks to execute on hourly (@hourly).

@hourly timestamp is similar to 0 * * * *. It will execute a task in the first minute of every hour, It may be useful to do hourly tasks.

@hourly sh /scripts/script.sh

crontab -e

* * * * * rm -rf /tmp/*

DEVOPS EASY LEARNING

Advance Commands in Linux

Advance Command Linux

cut, grep and pipe

The **cut** command is a command-line utility for cutting sections from each line of a file. It writes the result to the standard output. It can be used to cut parts of a line by delimiter, position, and character

The **pipe** command takes the output of command and send it as input to another command.

Example : # cut [option] [file]

```
# head -n10 /etc/passwd | cut -d: -f1-3
# uname -a | cut -d" " -f3 | cut -d- -f1 | cut -d. -f1-2
# cat /etc/*release |grep DISTRIB_ID  |cut -d= -f2
        # cut -d: -f1-3 /etc/passwd
# head -n10 /etc/passwd | cut -d: -f1-7
```

The **grep** (Global Regular Expression) command is used to search for a particular string or word in the file

Options:

- cut -d

DEVOPS EASY LEARNING

Advance Command Linux

awk, grep and pipe

The **awk** command is a command-line utility for cutting sections from each line of a file. It writes the result to the standard output. It can be used to cut parts of a line by delimiter, position, and character.

The **pipe** command takes the output of command and send it as input to another command.

Example : # awk [option] [file]

 # free -h |grep "Mem:" |awk '{print$2}'

 # lscpu |grep "CPU MHz:" |awk '{print$3}' |awk -F. '{print$1}'

 # df -mh |grep "/dev/" |head -1 |awk '{print$5}' |awk -F% '{print$1}'

 # cat /etc/ssh/sshd_config |grep -i ^PasswordAuthentication |awk '{print$2}'

 # ls -l |awk -F" " '{print$9}'

 # ifconfig |grep netmask |head -1 |awk -F" " '{print$2}'

 # netstat -ano |grep 22 |head -1 |awk -F" " '{print$4}' |awk -F: '{print$2}'

*# cat /etc/*release |grep DISTRIB_DESCRIPTION |awk -F'="' '{print$2}' |awk -F" " '{print$1}'*

The **grep** (Global Regular Expression) command is used to search for a particular string or word in the file

Options:

- cut -d

DEVOPS EASY LEARNING

Advance Command Linux

sed, grep and pipe

The **sed** command in UNIX stands for stream editor and it can perform a lot of functions on files like searching, finding and replacing, insertion or deletion.

The **pipe** command takes the output of command and send it as input to another command.

Example :

```
# sed -i '/^%admin/d' /etc/sudoers
# sed -i '/ALL=(ALL:ALL) ALL/a%admin  ALL=(ALL)      ALL' /etc/sudoers
sed -i '/# Allow members of group sudo to execute any command/a%admin
ALL=(ALL)      ALL' sudoers
sed -i '/^s4tia/d' sudoers
sed -i '/^%admin/d' sudoers
sed -i '/# User privilege specification/as4tia  ALL=(ALL)      ALL' sudoers
sed -i '/# User privilege specification/a%admin  ALL=(ALL)      ALL' sudoers
# cat /etc/ssh/sshd_config |grep -i ^PasswordAuthentication |awk '{print$2}'
# sed -i '/PasswordAuthentication no/d' sshd_config
# sed -i '/PasswordAuthentication yes' sshd_config
```

The **grep** (Global Regular Expression) command is used to search for a particular string or word in the file

Options:

- a = append
- d = delete

DEVOPS EASY LEARNING

Linux Commands

find, reboot and shutdown

The **find** command is used to filter objects (files and directories) in the file system

Example : # ***find [where to start searching from] [-options] [what to find]***

find / -iname [file name]

find / -iname [directory name]

find / -type f -iname [file name]

find / -type d -iname [directory name]

find . -iname tecmint.txt

The **reboot** command is used to reboot or restart the system

Example : # **reboot**

The **shutdown** command is used to restart or shutdown the system

Example : # **reboot**

shutdown -h +5 "Server is going down for upgrade. Please save your work."

shutdown -h + number of minute + [message]

shutdown -h +10 "Server is going down for upgrade. Please save your work."

shutdown -r +5 "Server will restart in 5 minutes. Please save your work."

-r = restart

-h = shutdown

reboot and init 6 = restart

init 0, halt, poweroff = stop the system

DEVOPS EASY LEARNING

Links

Links

Practice directory
Click here

Linux Commands
Click here

relative path and absolute path in linux
click here

DEVOPS EASY LEARNING

Congratulations

DEVOPS EASY LEARNING