***Linux***

* **AWS Account Create**

|  |
| --- |
| Go To Browser ▶ Search [ AWS Free Tier ] ▶ Go to Official Side ▶ Create New Account ▶ Fill All personal Details and submit credit/Debit card details |

|  |
| --- |
| If Bill Showing  Go to ▶ Support Centre ▶ Create Case ▶ Click on [ Account and Billing Support ] ▶ Type – Billing ▶ Category – Dispute a Charge ▶ Subject – Request for waive off my bill ▶ Description – Reason I’m student etc etc ▶ Phone ▶ Give Details After that get call from there side explain problem u get refund that bill. |

# **EC2 – Elastic Cloud Compute**

|  |
| --- |
| **How to Create EC2 Instance**  Go To Browser ▶ Search [ AWS Free Tier ] ▶ Go to Official Side ▶ Go to Home Page ▶ Click on [ My Account ] ▶Select [ AWS Management Console ] ▶ Select [ Root User ] and Login using login details ID- [hsabale5@gmail.com](mailto:hsabale5@gmail.com) PW – Omsairam96@ ▶ Select Region First Where u want to create Server [ E.g. Mumbai ].  Click on [ Services ] ▶ Search and Select [ EC2] ▶ Click on [ Instance Running ] ▶ Click On [ Launch Instance ]  ▶ Give Name and Tag in [ Instance Name ]  ▶ Select Application and OS Image (AMI – Amazon Machine Image) using search box search machine and select Which machine u want.  E.g.-> Amazon Linux 2 AMI (HVM) / Centos  ▶ Select [ Instance Type ] –t2.micro  ▶ Select [ Key Pair ]  –>Create Key Pair  -> Give Key Pair Name E.g. ReX  -> Key Pair Type [RSA]- Recommended because key encrypted  -> Private key file format [.pem / .ppk]  -> Create Key Pair and Download That .pem/.ppk file.  ▶ Network Setting   * Network - vpc-0aa5a59955e4d5528 * Subnet - No preference (Default subnet in any availability zone) * Firewall – Create Security Group – Allow SSH Traffic From – we can set custom IP also   ▶ Configure Storage   * 8,gp2   ▶Lunch Instance Now EC2 Instance Created |

* **Connect Linux Machine**
* **Connect 3 Way**

|  |  |  |
| --- | --- | --- |
| **1st :- On Browser**  ▶ Go to AWS -> Select EC2 Instance -> Connect -> Ec2 Instance Connect ->  Connect -> Now we can use on Browser  **OR**  **2nd :- On CMD**  ▶ Go to that file where we .pem / .ppk file downloaded  ▶ In that folder in Search bar type “CMD” and enter.  ▶ Write [ssh -i ReX.pem [ec2-user@ec2-65-1-147-146.ap-south-1.compute.amazonaws.com](mailto:ec2-user@ec2-65-1-147-146.ap-south-1.compute.amazonaws.com) ]   * ssh = command * ReX.pem = Download File Name * [ec2-user@ec2-65-1-147-146.ap-south-1.compute.amazonaws.com](mailto:ec2-user@ec2-65-1-147-146.ap-south-1.compute.amazonaws.com) = Link from (Go to AWS -> Select Created EC2 Instance -> Connect -> SSH Client -> in Example Select Link after the file name)  |  | | --- | |  |   ▶ Enter  ▶ Give Command “Yes”  ▶ Enter  **OR**  **3rd :- On Git Bash**  ▶ Go to that file where we .pem / .ppk file downloaded  ▶ In that folder in right click and Select “Show more option” and Select “Git Bash here”.  ▶ After Open Git Bash -> Go to the Created instance on AWS site  ▶ Select Instance -> Go to Connect -> go to SSH Client  ▶ Simple copy like this “ssh -i “ReX.pem” [ec2-user@ec2-65-1-147-146.ap-south-1.compute.amazonaws.com](mailto:ec2-user@ec2-65-1-147-146.ap-south-1.compute.amazonaws.com)” from SSH Client and Paste in Git Bash.   |  | | --- | |  |   ▶ Enter  ▶ Give Command “Yes”  ▶ Enter  **OR**  **4th :- On MobaXterm**  ▶ Go to google and download apk of this and install  ▶ Open this app and download key and Copy Created Ec2 Server “Public IP”  ▶ Go to app click on “session” -> click on “ssh” -> Remote host “Public IP”  ▶ Click on “Advanced ssh setting” -> Tick on “Use Private key” and select  ec2 downloaded key and click on ok  ▶ Click on Accept  ▶Login as : “ec2-user” and Enter |

***Linux***

* **Linux – Version 8**
* **Centos – Version 9**

# **Introduction to Linux**

* **What is Operating System?**
* Operating system is an interface between user and the computer hardware.
* The hardware of the computer cannot understand the human readable language as it works on binaries i.e. 0's and 1's.
* Also it is very tough for humans to understand the binary language, in such case we need an interface which can translate human language to hardware and vice-versa for effective communication.
* Operating System’s
* Windows
* Linux – Red Hat, Ubuntu, CentOS, fedora, Kali Linux
* IBM IAX
* Solaris
* HP-UX

## **Types of Operating System:**

1. **Single User Operating System**
2. **Single Tasking Operating System**
3. **Multi-Tasking Operating System**
4. **Multiprogramming Operating System**
5. **Time Sharing Operating System**
6. **Real Time Operating System**
7. **Multiprocessing Operating System**
8. **Distributed Operating System**
9. **Single User Operating System**
10. **Single User - Single Tasking Operating System**

* In this type of operating system only one user can log into system and can perform only one task at a time.
* E.g.: MS-DOS (Disk Operating System)

1. **Single User - Multi tasking operating System**

* This type of O/S supports only one user to log into the system but a user can perform multiple tasks at a time, browsing internet while playing songs, open document etc.
* E.g.: Windows -98, Xp, vista, Seven etc.

1. **Multiprogramming Operating System**

* These type of O/S provides multiple users to log into the system and also each user can perform various tasks at a time.
* In a broader term multiple users can logged in to system and share the resources of the system at the same time.
* E.g. UNIX, Linux, IBM

1. **Time Sharing Operating System**

* Time sharing OS allows the user to perform more than one task at a time, each task getting the same amount of time to execute if the same time elapse or Input Output operation requested, CPU shift the next job waiting and previous put on wait (in case allocated time over) .
* E.g. – BSD Unix, GEnie

1. **Real Time Operating System**

* In real time OS, the jobs have fixed deadlines and jobs have to be completed within their deadline.
* E.g. LynxOS, RT (Real time) Linux

1. **Multiprocessing Operating System**

* Multiprocessor operating systems are used in operating systems to boost the performance of multiple CPUs within a single computer system.

1. **Distributed Operating System**

* Distributed Operating System is a type of model where applications are running on multiple computers linked by communications.
* E.g. Unix , Linux, DEC
* **Linux Origins**

**-** LINUS TORVALDS

a) Finnish college student in 1991

b) Created Linux Kernel

- When Linux Kernel combined with GNU applications, complete free UNIX like OS

was developed.

* **What is Unix**
* The UNIX operating system is a set of programs that act as a link between the computer and the user.
* **Why Linux?**

• Fresh implementation of UNIX APIs

• Open source development model

• Supports wide variety of hardware

• Supports many networking protocols and Configurations

• Fully supported

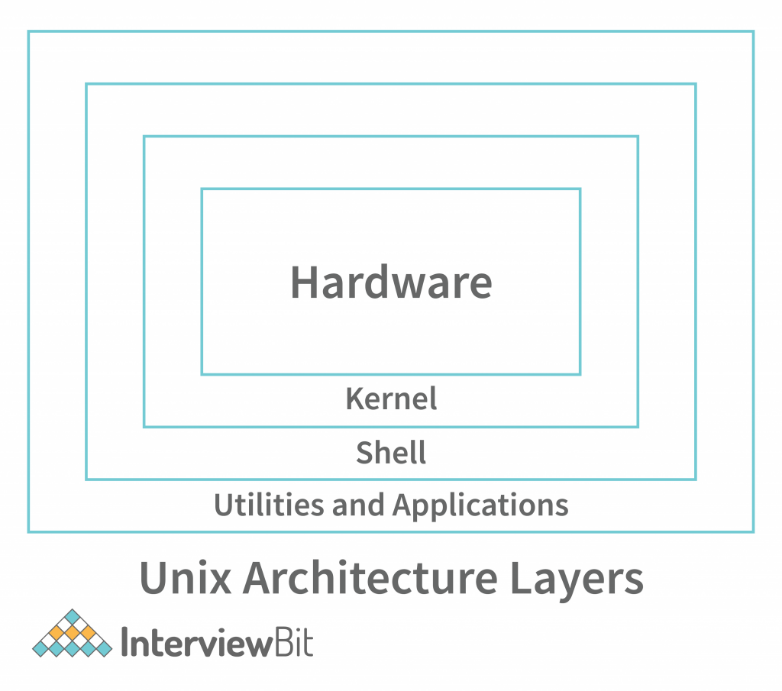
1) Linux is a UNIX like OS: Linux is a similar to UNIX as the various UNIX versions are to each other.

2) Multi-User and Multi-tasking: Linux is a multi-user and multi-tasking operating system. That means that more than one person can be logged on to the same Linux computer at the same time. The same user could even be logged into their account from two or more terminals at the same time; Linux is also Multi-Tasking. A user can have more than one program executing at the same time.

3) Wide hardware support: Red Hat Linux support most pieces modern x86 compatible PC hardware.

4) Fully Supported: Red Hat Linux is a fully supported distribution Red Hat Inc. provides many support programs for the smallest to the largest companies.

* **Unix Architecture**



* **The UNIX architecture has 4 layers. These layers are as shown below:**

1. **Hardware:**

* Hardware is the components that are humanly visible.
* Whatever hardware is connected to a UNIX operating system-based machine, comes in the hardware layer.

1. **Kernel:**

* The kernel is the heart of the operating system.
* The kernel is responsible for acting as an interface between the user and the hardware for the effective utilization of the hardware.
* The kernel is also responsible for process and file management. So, the main 2 features of the kernel are process management and file management.

1. **Process Management:**

* The processes that execute within the operating system require a lot of management in terms of memory being allocated to them, the resource allocation to the process, process synchronization, etc. All this is done by the Kernel in Unix OS

1. **File Management:**

* File management involves managing the data stored in the files. This also includes the transmission of data stored in these files to the processes as and when they request it.

1. **Shell:**

* We understood the importance of the kernel and that it handles most of the important and complex tasks of Unix OS.
* Since the kernel is such an important program of the Unix Operating System, its direct access to the users can be dangerous.
* **Hence, the Shell comes into the picture. Shell is an interpreter program that interprets the commands entered by the user and then sends the requests to the kernel to execute those commands.**
* When the execution of the process is completed, the shell again sends a request to the kernel to display the program/information on the screen to the user.
* **So, Kernel is an interface between the user and the hardware and the Shell is an interface between the user and the Kernel.**
* The shell can be used for opening a file, writing into the files, executing programs, etc. There are 3 types of shells in the Unix Operating system.

1. **Bourne Shell (sh):**

* It is the most widely available shell on Unix OS devices across the world. This was the first shell available in the Unix OS. It is simply called a shell.

1. **C Shell (csh):**

* The University of California (Berkeley) developed C Shell is another Unix shell that removes some of the obsolete features or problems from the Bourne Shell. So, it enhances the performance of the Bourne Shell.

1. **Korn Shell (ksh):**

* The name of the Korn Shell is based on its creator, David Korn. This shell enhances the C shell further by removing the shortcomings of the C shell and also enhancing the user interaction of the Bourne Shell.

1. **Applications/Application Programs:**

* The last layer of the Unix architecture is the Application Program layer. As the name suggests, this outermost layer of Unix Architecture is responsible for executing the application programs.

# **FILESYSTEM HIERARCHY SYSTEM**

**Linux uses single rooted, inverted tree like file system hierarchy**

1. **/ :-**

* This is top level directory It is parent directory for all other directories It is called as ROOT directory It is represented by forward slash (/) C:\ of windows

1. **/root :-**

* it is home directory for root user (super user) It provides working environment for root user C:\Documents and Settings\Administrator

1. **/home :-**

* it is home directory for other users It provide working environment for other users (other than root) c:\Documents and Settings\username

1. **/boot :-**

* it contains bootable files for Linux Like vmlinuz (kernel)..... ntoskrnl Initrd (INITial Ram Disk)and GRUB (GRand Unified Boot loader).... boot.ini, ntldr

1. **/etc :-**

* it contains all configuration files Like /etc/passwd..... User info /etc/resolv.conf... Preferred DNS /etc/dhcpd.conf.... DHCP server C:\windows\system32\dirvers\

1. **/usr :-**

* by default soft wares are installed in /usr directory (UNIX Sharable Resources) c:\program files

1. **/opt :-**

* It is optional directory for /usr It contains third party softwares c:\program files

1. **/bin :-**

* it contains commands used by all users (Binary files)

1. **/sbin :-**

* it contains commands used by only Super User (root) (Super user's binary files)

1. **/dev :-**

* it contains device files Like /dev/hda ... for hard disk /dev/cd rom ... for cd rom Similar to device manager of windows

1. **/proc :-**

* it contain process files Its contents are not permanent, they keep changing It is also called as Virtual Directory Its file contain useful information used by OS like /proc/meminfo ... information of RAM/SWAP /proc/cpuinfo ... information of CPU

1. **/var :-**

* it is containing variable data like mails, log files

1. **/mnt :-**

* it is default mount point for any partition It is empty by default

1. **/media :-**

* it contains all of removable media like CD-ROM, pen drive

1. **/lib :-**

* it contains library files which are used by OS It is similar to dll files of windows Library files in Linux are SO (shared object) files

## **[Linux Directory Structure](https://www.thegeekstuff.com/2010/09/linux-file-system-structure/" \t "_blank)**



* Ec2 Server Key Pair

|  |
| --- |
| * **After Create key**  1. Private Key 2. Public Key   Private key – Our Laptop  Public Key – AWS   * Means if key not match with AWS then permission denied from server. * **For Key create** * RSA – Recommended * Because if we choose RSA then data send with encrypted format. * If we not select RSA data send normal text format is that not secure. * **SSH**   **-** ssh is a protocol, which stands for Secure Shell or Secure Socket Shell. |

# **BASIC COMMANDS**

|  |
| --- |
| **For Identification User**  $ - Normal User  # - Root User |

|  |
| --- |
| * uname = we get OS name * uname –a = You get all system information with users name * uptime = its shows how long the system has been running or how much time pass after reboot * date = shows date and time of system * ls = its list shows files and directories. * ls –ltr = it will provide detailed information about each file or directory with file/folder permission * ls – l= display the contents of the current directories in long format with permission info. * cd = change directories where we want to go * e.g cd /tmp * history = we get all command that we used recently * df –h = The df command tells you the usage of memory details in the Linux system. * wget <https://code.jquery.com/jquery-3.6.0.min.js> * Is a networking command-line tool that lets you download files and interact with REST APIs. |

# **How Switch Normal User to Root User**

|  |
| --- |
| * Switch Normal User to Root User * sudo su - |

|  |
| --- |
| * Switch Root User to Normal User * exit |

|  |
| --- |
| * Which directories we are working * Pwd = present working directories * cd /filename = that type we can change present working directories |

# **Echo Command**

|  |
| --- |
| * echo $? = return the exit status of last command. * You got **127** that is the exit status of last executed command exited with some error. We get 127 if command error. * You got **0** that is the exit Commands on successful completion. We get 0 if our command successful executed * echo $PATH * This output is a list of directories where executable files are stored. * Shows all file system hierarchy of Linux. * E.g- /bin, /root |

# **Environment Command**

|  |
| --- |
| * env = display our current environments * export abc=10 - Set environment variable * env = now we can see in current environments see that variable |

# **Creating File**

|  |
| --- |
| * Create New File * touch Movies * touch = command * Movies = file name * ls = now we can see Movies file created in root folder |

|  |
| --- |
| * Create new File using Vi Editor * **But from “vi” we can create only one file. Using “touch” multiple thousand file can create** * vi Music * vi = command * Music – File name new * Write there we want * Save = Esc + Shift + :wq! * Enter * cat Music * we get what I can write there all data |

## **For Write First Line in File**

|  |
| --- |
| * If we want to write something in that file * cat >Movies * Now we can write in that file * For save = ctrl + d * cat Movies = now we can check what we can write in that file |

## **For Write More Line in File**

|  |
| --- |
| * Add some more lines in same files * cat >>Movies * Now we can write in that file also * For save = ctrl + d * cat Movies = now we can check what we can write in that file |

## **Edit Files Using “vi” editor**

|  |
| --- |
| * Edit something in Previous Lines code or Program we Use Vi Editor * vi Movies * vi – command * Movies – file name * Now for edit there = Esc + O or I * Now we can changes in any line and in between line using all sites arrows(⬇⬆➡⬅) * Save using = Esc + Shift + :wq! And Enter * wq means = write quick * cat Movies = now we can check what we can write and changed1q in that file |

|  |
| --- |
| * Delete any line from program in Vi editor * Go on start of the line that line want to delete you * Esc + dd = now this full line deleted |

|  |
| --- |
| * In Vi Editor you want to exit without save any new changes * Esc + Shift + :q! * Enter |

# **Remove File**

|  |
| --- |
| * Remove single file with confirmation * rm Movies * after the confirmation = yes * now this file deleted |

|  |
| --- |
| * Remove single file without confirmation * rm –f Movies * now this file deleted |

# **Copying File**

|  |
| --- |
| * Create multiple file at a time * touch a b c * now this 3 file created at a time * ls = now we can see files |

|  |
| --- |
| * File copy in different file location * cp a /tmp * cd /tmp * cd /tmp * we are Enter into the “tmp” directories      * ls = now we can see files see here |

## **Copy file With Change file Destination Name**

|  |
| --- |
| * If want to change file destination name * cp a /tmp/myfiles * cd /tmp * cd /tmp * we are Enter into the “tmp” directories      * cat myfiles |

# **Moving File**

|  |
| --- |
| * Move file * mv b /tmp * ls * we cant see this file here now because this file move to the “tmp” folder * cd /tmp * now we are in “tmp” folder * ls * now we can see file come here |

# **Creating Folder**

|  |
| --- |
| * Create New Folder * mkdir videos * mkdir = make directories command * videos = folder name * ls -l = now we can see all files and videos folder created * Blue colour indicate = Folder name * White colour indicate = File name |

## **Removing an Empty directory/folder**

|  |
| --- |
| * **Removing an Empty folder** * Using “rmdir” we only remove empty directories(Folder) * First change directories come to “main directories” after that delete this folder * cd * Now we are in root directories * If videos folder empty they don’t have any files then it will be deleted * rmdir videos * after the confirmation = yes * now this folder deleted * **Removing an Empty directory** * Using “rmdir” we only remove empty directories(Folder) * First change directories come to “main directories” after that delete this folder * cd * Now we are in root directories * If videos folder empty they don’t have any files then it will be deleted * rmdir /videos * after the confirmation = yes * now this folder deleted |

## **Removing an directory/folder with files or directories inside**

|  |
| --- |
| * **Removing a directory with files or directories inside** * If we want to delete directories/folder with Inside Files or directories * First change directories come to “main directories” after that delete this folder * cd * Now we are in main directories * In that folder have directories or not this folder will be deleted now * rm –rf videos * where r stands for recursive and f stands for forcefully * now this folder deleted |

# **Creating Directories**

|  |
| --- |
| * Create New Directories * mkdir /oracle * mkdir = make directories command * /oracle = directories name * ls -ld = now we can see all files and videos folder created * Blue colour indicate = Folder name * White colour indicate = File name |

# **How to Enter Folder, Directories**

|  |
| --- |
| * **How to create and Enter Inside into the Folder** * mkdir Music * cd Music * Check first where are you using “pwd” command * pwd * there we can see [ /root/Music ] means we are in videos folder now * Create multiple files in that folder * touch vdo1 vdo2 vdo3 vdo4 * ls –l * now all files created in this folder * **How to create and Enter inside into the Directories** * mkdir /oracle * cd /oracle * Check first where are you using “pwd” command * pwd * there we can see [ /root/oracle ] means we are in oracle directories now |

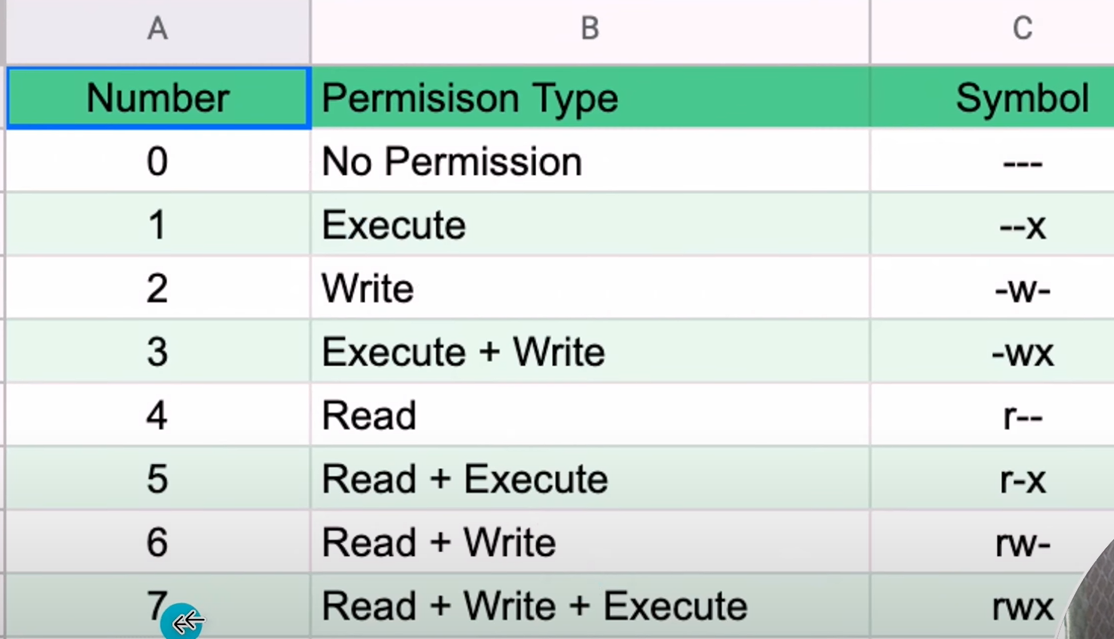
# **Find Any File or Folder or Directories Path**

|  |
| --- |
| * cd * find / -name \*music\* * find / -name \*\* = Command * music = file, folder, directories name |

# **Find Any Word in File Exist or Not**

|  |
| --- |
| * First find path of directories word search and copy there path      * find / -name \*music\* * find / -name \*\* = Command * music = file, folder, directories name * grep -1 'good' /root/music/a * **grep -1 1 ' '** = command * **good** = word which you want to word filter * **/root/music/a** = that files path there we want to find this word * a command used in searching and matching text files contained in the regular expressions. |

# **Changing Permission of file or folder**



* **Types of Permission**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Short form + Value | Symbol |
| 1 | read | r = 4 | r - - |
| 2 | write | w = 2 | - w - |
| 3 | execute | x = 1 | - - x |
|  |  | Total = 4+2+1 = 7 |  |

|  |
| --- |
| Proper Format of **read, write and** **execute** permission |

* **Permission can be set on any file/folder/dir by two methods:-**

**1 - Symbolic method (ugo)**

**2 - Absolute methods (numbers)**

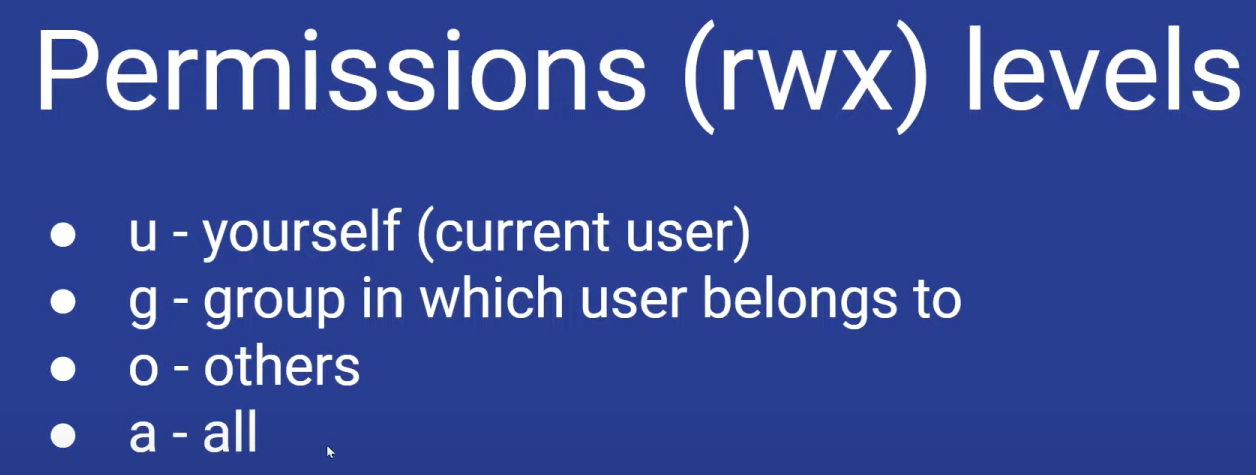
* **If we want to change permission 2 way can change permission**

## **1st way Absolute methods (numbers)**

|  |
| --- |
| For Check Permission   * ls –l   E.g = - --- --- --- root root Music  Value = - 421 421 421 root root Music  Now we want to change read, write and execute permission to this file.   * chmod 222 Music * Ch = change , Mod = mode, 222 – value permission , Music – file name   E.g –  E.g = - --- --- --- root root Music   * chmod 777 Music – rwx rwx rwx   4+2+1 4+2+1 4+2+1  - if hume yaha owner mai read write ki sirf deni permission = rw-  - if hume yaha group owner mai read ki sirf deni permission = r—  - if hume yaha all other user mai write ki sirf deni permission = -w-   * chmod 744 Music – rwx r-- r-- * chmod 700 Music – rwx --- --- * chmod 733 Music – rwx -wx -wx * chmod 222 Music – -w- -w- -w- |

|  |
| --- |
| **For Removing Permissions**   * ls –l   E.g = - --- --- --- root root Music  Value = - 421 421 421 root root Music  Now we want to remove read, write and execute permission to this file.   * chmod 000 Music * Ch = change , Mod = mode, 222 – value permission , Music – file name   E.g –  E.g = - --- --- --- root root Music  E.g = - rw- --- r-x root root Music   * chmod 000 Music – rwx rwx rwx   4+2+1 4+2+1 4+2+1     * chmod 044 Music – --- r-- r-- |

## **2nd way Symbolic method (ugo)**



|  |
| --- |
| * **For Adding Permissions**     For Check Permission   * ls -l   E.g = - --- --- ---  = - **user** **group** **other**   * chmod u+rwx Music * chmod = command * u = youself or owner * + = add permissions * rwx = means all permission deni hai * chmod g+rx Music * chmod = command * g = group of owner * + = add permissions * rx = means read, execute ki permission deni hai * chmod o+rx Music * chmod = command * o = other * + = add permissions * rx = means read, execute ki permission deni hai * chmod a+rx Music * chmod = command * a = at a time all * + = add permissions * rx = means read, execute ki permission deni hai all ko * **For Removing Permissions**   For Check Permission   * ls -l * chmod u-rwx Music * chmod = command * u = youself or owner * - = remove permissions * rwx = means all permission deni hai * chmod g-rx Music * chmod = command * g = group of owner * - = remove permissions * rx = means read, execute ki permission deni hai * chmod o-rx Music * chmod = command * o = other * - = remove permissions * rx = means read, execute ki permission deni hai * chmod a+rx Music * chmod = command * a = at a time all * + = add permissions * rx = means read, execute ki permission deni hai all ko |

# **Change Own Ownership of File or Folder**

|  |
| --- |
| For Check Ownership   * ls -l   E.g – rwx rwx rwx root root Music  Owner Group   * chown ec2-user Music * ch = change, own = ownership, ec2-user – that user we want to give, Music – File/Folder name * ls –l = now we can see ownership change of owner * E.g – rwx rwx rwx ec2-user root Music |

# **Change Group Ownership of File or Folder**

|  |
| --- |
| For Check Ownership   * ls -l   E.g – rwx rwx rwx root root Music  Owner Group   * chgrp ec2-user Music * ch = change, grp = group ownership, ec2-user – that user we want to give, Music – File/Folder name * ls –l = now we can see group ownership change of owner * E.g – rwx rwx rwx ec2-user ec2-user Music |

# **Hyphen (use) in Linux**

|  |
| --- |
| * **Without Hyphen (-)**   Condition – If we changed normal user to root user   * sudo su * Output :- [root@172.31.42.151-ec2-user](mailto:root@172.31.42.151-ec2-user) * Means user will be changed but user id still ec2 * **With Hyphen (-)**   Condition – If we changed normal user to root user   * sudo su - * Output :- [root@172.31.42.151](mailto:root@172.31.42.151) ~ * Then user id also changed and totally converted into root user now. |

# **Create New User in Linux**

* Like = ec2-user, root

|  |
| --- |
| * useradd Ethans * useradd = command * Ethans – new user name as per our think * Id Ethans * Check user created or not * If we want to go inside user * su – Ethans * now we are not a root user now we are Ethans user |

|  |
| --- |
| * If any file created and we want to change that user permissions * touch abc * new “abc” name file created now * changed that file ownership * chown Ethans abc * now file owner ownership changed * e.g - – rwx rwx rwx Ethans root abc |

# **Password Based Authentication to User**

* E.g set password base authentication to = ec2, root, Ethans

|  |
| --- |
| First Create any User   * useradd Ethans * useradd = command * Ethans – new user name as per our think * id Ethans * Check user created or not * passwd Ethans * passwd = command * Ethans = user name there we want to set pass base authentication * After that give any password and Enter * Retype same password |

* **Now we need to Enable Password Authentication**
* Need to changes in configurations file
* Linux Configuration file is “/etc” as per Linux Directory Structure

|  |
| --- |
| * cd /etc * now we are in etc configuration file * cd ssh * now we are in ssh file for configuration * pwd * check present working directories * ls * here we can see we have “sshd\_config” file in that file changes needed * for this file edit we use “vi” editor * vi sshd\_config * this file open now in vi editor * scroll and go down side “PasswordAuthication No” * for edit there use = Esc + o * Remove “No” and write “yes” there * now we changed “no” to “yes” for enable password Authentications * Save = Esc + Shift + :wq! * Enter * Systemctl restart sshd * Enter   **OR**   * vi /etc/ssh/sshd\_config * this file open now in vi editor * scroll and go down side “PasswordAuthication No” * for edit there use = Esc + o * Remove “No” and write “yes” there * now we changed “no” to “yes” for enable password Authentications * Save = Esc + Shift + :wq! * Enter * Systemctl restart sshd * Enter * **Now we going to connect new Created user (Ethans) using Password based Authentications** * Open GitBash Anywhere * Go to AWS free tier “EC2 Instance” and select Created instance and Copy there “Public IPv4 Address” (e.g- 3.111.197.108) * Now write in github and use that IP’s. * ssh Ethans@Public IP’s * E.g = ssh [Ethans@3.111.197.108](mailto:Ethans@3.111.197.108) * Enter * Confirm “Yes” * Write password * Enter * Now we Official entered into the Ethans User * [ethans@3.111.197.108](mailto:ethans@3.111.197.108) ~ |

# **Key Based Authentication to User**

|  |
| --- |
| First Create User   * useradd devuser * User Created * passwd devuser * Set Password   For Change User   * su - devuser * User Switched Now we are in devuser * ssh-keygen * Enter * Enter * Enter * Enter   For See key   * cd .ssh * ls –ltr * now there we get public and private key * Like this e.g- * public key = - rwx rwx rwx devuser devuser 606 25 08:58 id\_rsa.pub * private key = - rwx rwx rwx devuser devuser 2655 25 08:58 id-rsa     For open that Key   * cat id\_rsa.pub * Now key opened * You can give that key to developer user to login my server * Now they can able to login my server without password   How developers can login through my key   * They need for that * Copy key, username , public ip of that EC2 Instance * Now they can login like this * Ssh –I “copy-key” devuser@publicipaddress * E.g - ssh -i Copy Key devuser@2-65-1-147-146 |

# **How many user in your Server**

|  |
| --- |
| * cat /etc/passwd |

# **For all Group Information**

|  |
| --- |
| * Cat /etc/group |

# **Create New Group**

|  |
| --- |
| * groupadd harshal * new group created * cat /etc/group * now you can see group created along with other group |

# **Allotment any Group to New User**

* Every User need Primary and Secondary Group
* For check command – id devuser (file or folder name)
* E.g= uid = 1005 (devuser) gid = 1005 (harshal) groups = 1005 (devgroups)
* uid (devuser) = userid name
* gid (harshal) = primary group name
* groups (devgroups) = secondary group name
* **Primary Group allotment Customize**

|  |
| --- |
| First create any group   * groupadd Harshal * check group info * cat /etc/group   After that copy that groups id (1003) or name  We can allot group any user 2 way  1st :- Using group id   * useradd –g 1003 Rimix * useradd –g = command * 1003 – group id from group information * Rimix – new creating time user that user want to allot group * id Rimix * you clearly see that users group changed. * E.g – uid = 1005 (Rimix) gid = 1005 (harshal) groups = 1005 (harshal)     2nd :- Using group name   * useradd –g sable music * useradd –g = command * sable – group name that we want to allot from group information * music – new creating time user that user want to allot group * **Now new group (sable) allotted to new user (music)** * Check like this * id music * you clearly see that users group changed. * E.g – uid = 1005 (music) gid = 1005 (sable) groups = 1005 (sable) * If I create new user (devuser) that time automatically group id and group created but if I want to change group as per our requirement then we can do like this 2 way. |

* **Secondary Group allotment Customize**

|  |
| --- |
| * We already see primary group allot 2 way * Now secondary group how to allot   After that copy that groups id (1003) or name  We can allot group any user 2 way  1st :- Using group id   * usermod –G 1003 rahul * useradd –g = command * 1003 – group id from group information * rahul – created user that user want to allot group   2nd :- Using group name   * usermod –G harshal rahul * useradd –g = command * harshal – group name that we want to allot from group information * rahul – created user that user want to allot group * **Now new group (harshal) allotted to user (rahul)** * Check like this * id rahul * you clearly see that users group changed. * E.g – uid = 1005 (rahul) gid = 1005 (rahul) groups = 1005 (rahul), 1005 (harshal) * If I create new user (rahul) that time automatically group id and group created but if I want to change group as per our requirement then we can do like this 2 way. |

# **Checked Password Encrypted or Not**

|  |
| --- |
| * Cat /etc/shadow      * Now we can see * “devuser” we are set password so encrypted password showing * But “ec2-user”, “Rimix”, “music”, “rahul” users don’t have set any password so after username showing exclamation mark [ !! ] * !! – not password encrypted * $( encrypted pass) = password encrypted |

# **Linux Processes check**

|  |
| --- |
| * top * Used to show the Linux processes. * Provides a dynamic real-time view of a running system. * vmstat * vmstat (virtual memory statistics) is a system monitor which provides activity information about processes, CPU, memory, block IO, paging and more. * vmstat 5 5 * we get stat of 5 times interval of 5 second |

# **Active Processes in Server**

|  |
| --- |
| * ps –ef * enables you to check the status of active processes on a system, as well as display technical information about the processes. |

# **How many user log into your Server**

|  |
| --- |
| * w * Displays information about all users logged into the current system. |

# **Kill any Process Manually**

|  |
| --- |
| * kill -9 2979 * kill -9 = command * 2973 = process id * a built-in command which is used to terminate processes manually. |

# **For Ignoring Case Sensitive**

|  |
| --- |
| * If I have any user like “devuser” and I filtered that user * Like this * grep Devuser /etc/passwd * but we don’t get our result because our user “devuser” actually starts from small “d” so we not get result. * But      * grep –i Devuser /etc/passwd * after I use –i I getting result because * –i work as a Ignoring case Sensitive |

# **Show Disk Space**

|  |
| --- |
| * df –h * show the file system disk space statistics in “human-readable” format |

# **Display file system usage in KB**

|  |
| --- |
| * df –k * to display file system usage in KB |

# **Nginix Package Install in Linux**

* **What is a package in Linux?**
* A package in Linux is a compressed software archive file containing all the files included with a software application that provides any functionality.
* We can create package 2 way

1. Rpm
2. Yum
3. rpm

* but if we use rpm
* Any package don’t need any dependencies means dependencies already have then using this command package will install.
* But any package need dependencies with installation then using this command not going to install that package.

1. yum

* if we use yum
* If package don’t have any dependencies then using this command it first install all dependencies and install package also.

|  |
| --- |
| * **Install nginx package** * yum install nginx * confirm “y” * Enter |

|  |
| --- |
| * **Remove nginx package** * yum remove nginx |

* **How I can check Package status**

**Note - 1 Package created and running means 1 process also running.**

* **Package = Service**
* systemctl status nginx
* check first status of package “Active = inactive (dead)” means it will not running right now

|  |
| --- |
| * **For Start package** * systemctl start nginx * now check status of package * systemctl status nginx * check first status of package “Active = Active running” means it will running right now * means process started * **For Active Process check** * ps –ef |grep nginx * now we can see process active running of nginx |

|  |
| --- |
| * **For Stop package/Service**      * systemctl stop nginx * now check status of package * ps –ef |grep nginx * now we can see process active running stop of nginx |

## **For Enter Nginx Package**

* cd /usr/share/nginx/ = if we want to enter Nginx Package only

# **Httpd Package Install in Linux**

|  |
| --- |
| * **Install nginx package** * yum install httpd * confirm “y” * Enter |

|  |
| --- |
| * **Remove nginx package** * yum remove httpd |

* **How I can check Package status**

**Note - 1 Package created and running means 1 process also running.**

* **Package = Service**
* systemctl status nginx
* check first status of package “Active = inactive (dead)” means it will not running right now

|  |
| --- |
| * **For Start package** * systemctl start httpd * now check status of package * systemctl status httpd * check first status of package “Active = Active running” means it will running right now * means process started * **For Active Process check** * ps –ef |grep httpd * now we can see process active running of httpd |

|  |
| --- |
| * **For Stop package/Service**      * systemctl stop nginx * now check status of package * ps –ef |grep nginx * now we can see process active running stop of nginx |

## **For Enter Httpd Package**

* cd /var/www/html/ = if we want to enter httpd Package only

## **For Deploy App in Httpd Package**

|  |
| --- |
| * After enter in HTTPD package create html folder and deployed application * vi index.html * Edit = Esc + o * Esc +Shift wq! And Enter * Now copy instance public IP and paste on browser and our application we can see |

# **Increase Disk Size of Created EC2 Machine/Cloud**

|  |
| --- |
| * Check First * fdisk -l * df –l      * we can see only one disk here 8GB = /dev/xvda1 8.0G |

|  |
| --- |
| * Now we increasing through this process * Go on AWS Free tier * Ec2 Instance * Left Side – Click on “Volume” in Elastic Block Store * Click on “Create Volume“ * Set “Size” how many GB want to increase size. * For selecting Availability zone you need to check Where your Server created “EC2 Instance - Availability zone” check and select same Zone. E.g Now my Server (EC2 Instance ) Availability Zone is “ap-south-1b” so same zone I’m selecting here * Create Volume * After Create wait for some minute it status change **“Creating”** to **“Available”** |

|  |
| --- |
| Final Step -   * Now this volume just created and its status is “status - Available” * Means its Created but not attached to your Service yet * We need to change status **“Available”** To **“in Use”** * In “Volume” Select new created Volume * Click on “Action” * Select “Attach Volume” * In “Attach Volume” select Instance there * Attach Volume * Now Status changed to “In use” * fdisk -l * now we can see on default created 8GB disk and now we external created disk added in Server.      * Disk - /dev/xvda 8GiB = already have * Disk - /dev/xvdf 10GiB = new created |

# **Partitions of Disk**

|  |
| --- |
| * After Increase disk size Using Volume, We can partition of Disk * fdisk /dev/xvdf * fdisk = command * /dev/xvdf = path of new created disk that we want to partition * m * this command for help of new command info * in that “n = add a new partition” |

|  |
| --- |
| * n * n for add a new partition |

|  |
| --- |
| * Primary Or Extended * p * p means primary because its fresh new so |

|  |
| --- |
| * Partition Number * 1/2/3/4 or Enter * This will ask for how many partition you want or can enter because default is 1 means 1 partition |

|  |
| --- |
| * First Sector * Enter * Because we created 1 partition |

|  |
| --- |
| * Last Sector * +5/+10/+15 or Enter (All Disk) * Last Sector ask for how many disk size want to partition from selected Disk E.g +5,+10,+15 GiB   E.G – If we have 50Gb Disk so we can set 25GB here partition so 2 partition will be created of 25-25 GB   * If direct Click enter means all disk will be select * **Now partition created 1 by type ‘Linux’ and Size of 10GiB** |

|  |
| --- |
| * For Display * p * Display disk and partition disk |

|  |
| --- |
| * w * Save and exit partition |

|  |
| --- |
| For Disk Info   * fdisk –l * now we get default EC2 disk , new created disk (Using Volume), Partition also      * **first (/dev/xvda) is default disk** * **second (/dev/xvdf) is created disk** * **Third (/dev/xvdf1) is partition** * If we want create that 10G volume disk in 2 file system need to create twice like this * In last sector size select first “+4G” and second time “+4G” then two partition created like this |

## **Create File system for disk and Mount to directories**

|  |
| --- |
| * If we check process we can see only 8 GB Disk * df –h      * but we want see that 10 GB partition also see there then do |

|  |
| --- |
| * **Create File System** * mkfs.xfs /dev/xvdf1 * mkfs.xfs = command , mkfs means – make file system * /dev/xvdf1 = new created partition path * The mkfs. xfs command is used to create a new filesystem |

|  |
| --- |
| * **Create any directories for file system** * Per file system for one folder only * mkdir /oracle * /oracle = new directories name that we want to store file system |

|  |
| --- |
| * **Process Of directories attached to file system** * mount /dev/xvdf1 /oracle * mount = command * /dev/xvdf1 = path of partition disk * /oracle = created directories for file system * The mount command allows users to mount, i.e., attach additional child file systems to a particular mount point on the currently accessible file system.      * See folder attached to the file system |

|  |
| --- |
| * **If we check before that process we can see only 8 GB disk but now we can see 10 GB disk here also added and check using df –h command** * df –h      * now we can see 8 GB & 10 GB both file showing here now |

* **It’s for if any file we upload in that folder capacity changes just we see**

|  |
| --- |
| * **How Enter inside to Oracle directories** * cd /oracle |

* **Example We Create One file in That Directories**

|  |
| --- |
| * **In that Directories we Just create one file for example 1** * dd if=/dev/zero of=./gentoo\_root.img bs=4k iflag=fullblock,count\_bytes count=4G * its take some minute after that this command make file of 4GB |

|  |
| --- |
| * **For Check file created or not** * ls –l      * see here our 1 file created and its size 4GB |

|  |
| --- |
| * **For Check file created or not** * ls –l      * see here our 2 file created and its size 4GB     **After Creating 2 files in the Disk**   * Now because of 4GB 2 files created * so our 10 GB partition file , 8.2 GB used * Now Available disk is 1.9 GB * 82% Disk Used |

## **For Check File Size Uploaded in Folder**

|  |
| --- |
| * **For Check File Size Uploaded in Folder** * du –sh \*      * here our 4 GB 2 files created now we can see and its Size also |

## **Mount Directories Temporary to Permanent**

* Using “mtab” and “fstab”
* If we stop our Ec2 Server and Start again then mounted directories with file system remove because it’s temporary mount.
* Directories Mount Permanent with File System
* **So we use “Mtab” and“Fstab” then this directories mount with file system Permanent.**

|  |
| --- |
| * How to set “/oracle” mount directories Permanent with File system * cat /etc/mtab * in that last line we can see our created folder so copy this full line like this      * Open this vi editor * vi /etc/fstab * shift + d for edit * and paste in that folder after “/dev/mapper/cs-swap” Line      * Remove all till “xfs” after that just write “default 0 0” like this below image      * Like this * Now save this = Esc + Shift + :wq! * Enter * For check * df –h      * /oracle directories mount permanent now * mount –av      * Now our directories Mount with File System Permanent. |

# **For Check Mount Directories Permission and Give Mount Directories Permission**

|  |
| --- |
|   **For Check Mount Folder Permission**     ls –ld /oracle    C:\Users\hsaba\AppData\Local\Microsoft\Windows\Clipboard\HistoryData\{EAEE7A56-73EF-4B62-8E78-8C9A27B16BEC}\{DDD3B60B-E638-40B2-902C-3CFE23B27683}\ResourceMap\{55B7E6F8-1433-4B21-BA4B-5235FBBA77EA}  -        now we can see oracle folder permission    **For Give Mount Folder Permission**     chmod 755 /oracle |

# **Extended Disk Using LVM: - Logical Volume Manger**

* A device mapper framework that provides logical volume management for the Linux kernel.
* Used for Extend Disk Size whenever you want

|  |
| --- |
| Linux Logical Volume Manager Tutorial | Linux Training Academy |

## **Unmounts file System**

|  |
| --- |
| 1. If any file system mounted before that so we need to “unmounts” that file system after that we can create PV.  * df –h * copy partition path      * here we can see /oracle file system mounted * **Unmounts file System**      * umount /oracle * now file system unmounts * Now try to create **Physical Volumes (PV)** |

|  |
| --- |
| * **Create Physical Volumes (PV)** * So that it will be available to LVM as storage capacity. * df –h * copy partition path * Command to create a PV: * pvcreate /dev/xvdf1 * pvcreate = command * /dev/cvdf1 = created partition path * Display PV capacity and additional information:      * pvdisplay   **Important** :-   1. if we get “Command Not found” for create physical volume (PV) then it may be reason some AWS machine E.g. – “Amazon Linux AMI 2023” machine not pre-installed LVM so we need to Install LVM in Machine First then You can we create Physical Volume  * **Install LVM Package** * yum install lvm\* * y * Now we can create physical volume (PV) |

|  |
| --- |
| * **Manage Volume Groups (VGs)** * VG must have at least one member (myvg is our group name and   others are our PVs)   * vgcreate myvg /dev/xvdf1 * vgcreate = command * myvg = create volume group any name as per our think * /dev/cvdf1 = created partition path   To display information for a VG named vg00   * vgdisplay myvg |

|  |
| --- |
| * **Manage Logical Volumes (LVs)** * To create a Logical Volume / Extend Disk Size here * lvcreate -L size(1G or 1T) -n lvname vgname * E.g = lvcreate –L +1.9G –n lvmy myvg * To display information for a LV      * lvdisplay |

* **Now File Extended but we need to Mount that file with Folder**

|  |
| --- |
| * Create File System inode * mkfs.xfs /dev/myvg/lvmy * mkfs.xfs = command * /dev/myvg/lvmy = after “lvdisplay” LV Path |

|  |
| --- |
| * Mount the folder with File System * First Create Oracle folder * mkdir /oracle * if directory already have then we can do next process * Mount Filesystem With folder * mount /dev/myvg/lvmy /oracle |

|  |
| --- |
| * For Check * df –h * but extended file not yet show here * we need to create inode for that command * xfs\_growfs /oracle * Now check * df –h * Now we can see our mount folder also allocated and path also set by new one is “/dev/myvg/ilo10” and File Extended. |

# **Shell Script**

* Execute multiple command at a time.
* “#!” It’s called shebang
* #!/bin/sh -------- Linked With --------🡪 #!/bin/dash ------> is default
* #!/bin/bash

|  |
| --- |
| * **Create Shell Script File** * Vim demo.sh * Vim – Is also vi editor but improved version * Demo – shell script file name * .sh - file format of shell script * Click on “i” means can insert writing * For Exit = Esc + Shift + :wq! and Enter * **Write Shell Script in Vi Editor** * **Run Shell Script** * ./demo.sh **OR** sh demo.sh * “./” OR “sh” = command * Demo.sh = file name * **If permission denied give permission** * ls –ltr * check permission * chmod 777 demo.sh * now can run shell script its run |

* **Write Shell Script**

**E.g.**

|  |
| --- |
| * vim shellscriptfile.sh * shell script file created * **Write In Vim Editor Shell Script**   E.g. ->   * #!/bin/bash * echo "My Name is Harshal"   # Create Folder = this file not print just write for example   * mkdir shellFolder   # Create file in that   * cd shellFolder * touch firstfile secondfile * In that Shell Script One Folder Created after Enter in that folder and In folder 2 file created |