## Linux Filesystem Hierarchy: -

* bin -> usr/bin
* boot
* dev
* etc
* home
* lib -> usr/lib
* lib64 -> usr/lib64
* local
* media
* mnt
* opt
* proc
* root
* run
* sbin -> usr/sbin
* srv
* sys
* tmp
* usr
* Var

## Root: -

LINUX

## Home: -

**Usr: -**

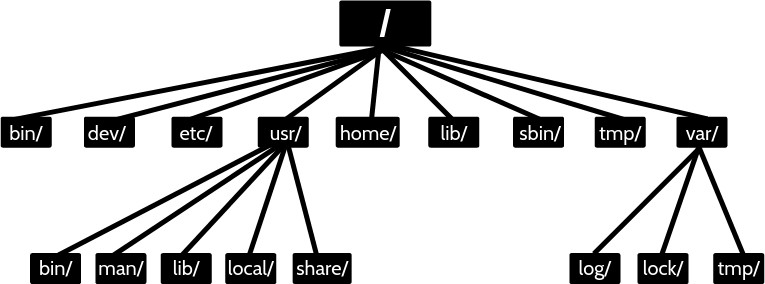
* It is a home directory for root user (Super User).
* It provides working environment for root user.
* It is a home directory for other users.
* It provides working environment for other users (other than root).
* By default, All the software’s are installed in this directory.

**Bin: -**

**Sbin: -**

**Var: -**

* It contains commands used by all users except root.
* It contains commands used by only super user (root).
* It is containing variables data like mails, log files of services.



**Basic commands: -**

|  |  |
| --- | --- |
| **Commands** | **Usage** |
| date | Show the current date and time |
| cal | Show this month's calendar |
| uptime | Show current uptime (how much time you are used) |
| whoami | It will show the current user |
| finger | Display information about user |
| Users/id | Show on which user you logged in/ Current user info |
| man | Shows manual of command |
| username | Shows your user name |
| who/w | It shows the login user details IP, IDLE time…etc |

**Create & delete a file or directory: -**

|  |  |
| --- | --- |
| **Commands** | **Usage** |
| touch | Creates a 0 bites file |
| cat > filename | Create file and allow to write |
| nano | Create a file if filename doesn't exit |
| vi | Create a file if filename doesn't exit |
| rm | Remove a file |
| mkdir | Create a directory |
| rmdir | Remove an empty directory |
| rm -rf | Remove a directory if it has any data |

**Managing files or Directories: -**

|  |  |
| --- | --- |
| **Commands** | **Commands** |
| cp | Copy a file |
| mv | Move a file |
| find | Find a file |
| grep | Search for a pattern in a file |
| cd | Switch between directories |
| diff | Find content difference in 2 files |
| sed | Search and replace particular pattern |
| chmod | Change file permissions |
| chown | Change ownership of a file |
| file | Show what kind of file it is |

**Copy Command: - Syntax: -**

* + cp <source file> <destination file>
  + cp <source path> <destination path>

## Example: -

* + cp file1 file2
  + cp file1 /home/file2
  + cp /home/file2 /etc/file1

## Move Command: - Syntax: -

* + mv <source file> <destination file>
  + mv <source path> <destination path>

## Example: -

* + mv file1 file2
  + mv file1 /home/file2
  + mv /home/file1 /etc/file200

## Note: -

* + Move a file is used to rename the file name, Move the file from one place to another place.
  + If the file is created it will overwrite the data if not the file is created and write the data in that file.

## Find Command: -

**Syntax: -**

* + find / -option filename

|  |  |
| --- | --- |
| **Option** | **Usage** |
| -name | For searching a file with its name |
| -user | For files whose owner is a particular user |
| -group | For files belonging to particular group |

## Example: -

* + find / -name file1 >>It is searching for file1 in the entire Linux.

## Diff Command: -

**Syntax: -**

* + diff <filename> <filename>

## Example: -

* + diff file1 file2 >>It shows the difference between two files

## Grep Command: -

* + It is used to pick out the required expression from the file and print the output.

## Syntax: -

* + grep <patron> filename

## Example: -

* + grep Gayathri filename >> If the word is same in the file then it is print.
  + grep Gateway /etc/ssh/sshd\_config
  + grep -i gateway /etc/ssh/sshd\_config >> It is print the case-sensitive.

### Grep command will work on files only not on directories.

* + ls -l | grep dir1 >>then it works on directories.
  + ls -l | grep ^d >>It shows the first letter d starting files/directories.

## Sed Command: -

* Which is used to search a word in the file and replace it with the word required to be in the output.

**Note: -** It will modify the output, but there will no change in the original file

## Syntax: -

* + sed ‘s/old\_text/new\_text/’ file\_name >> Replace the text if the text is

separate from the other word.

* + sed ‘s/old\_text/new\_text/g’ file\_name >> Replace the text if the text is part of

the word.

* + sed -i ‘s/old\_text/new\_text/’ file\_name >>Replace the text and update the file.
  + sed -n ‘5,10p, file\_name >>It will display the lines from 5-10 only
  + sed ’10,20d’ file\_name >>It will delete the lines from 10,20 and print.

## Example: -

* + sed 's/Gayathri/Sarvani/' file2
  + sed 's/thri/Sarvani/g' file2
  + Sed 's/thri/Sarvani/ig' file2 >> i means ignore the case sensitive.
  + Sed -i 's/thri/Sarvani/ig' file2
  + sed -i 's/Gayathri/Sarvani/' file2
  + sed -n '1,2p' file2
  + sed -n '1p' file2
  + sed '1d' file2 >> It will delete in the output not actual file
  + sed ‘1,2d’ file2 >> It will delete in the output not actual file

## User Management

## 

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Example** | **Home Directory** | **Shell** |
| Super User | Root | root | bin/bash |
| System User | ftp, ssh, apache | var/ftp, etc | sbin/nologin |
| Normal User | Visitor, ec2-user | home/username | bin/bash |

**User Add and Permissions: - Syntax: -**

* + useradd <option> <username>

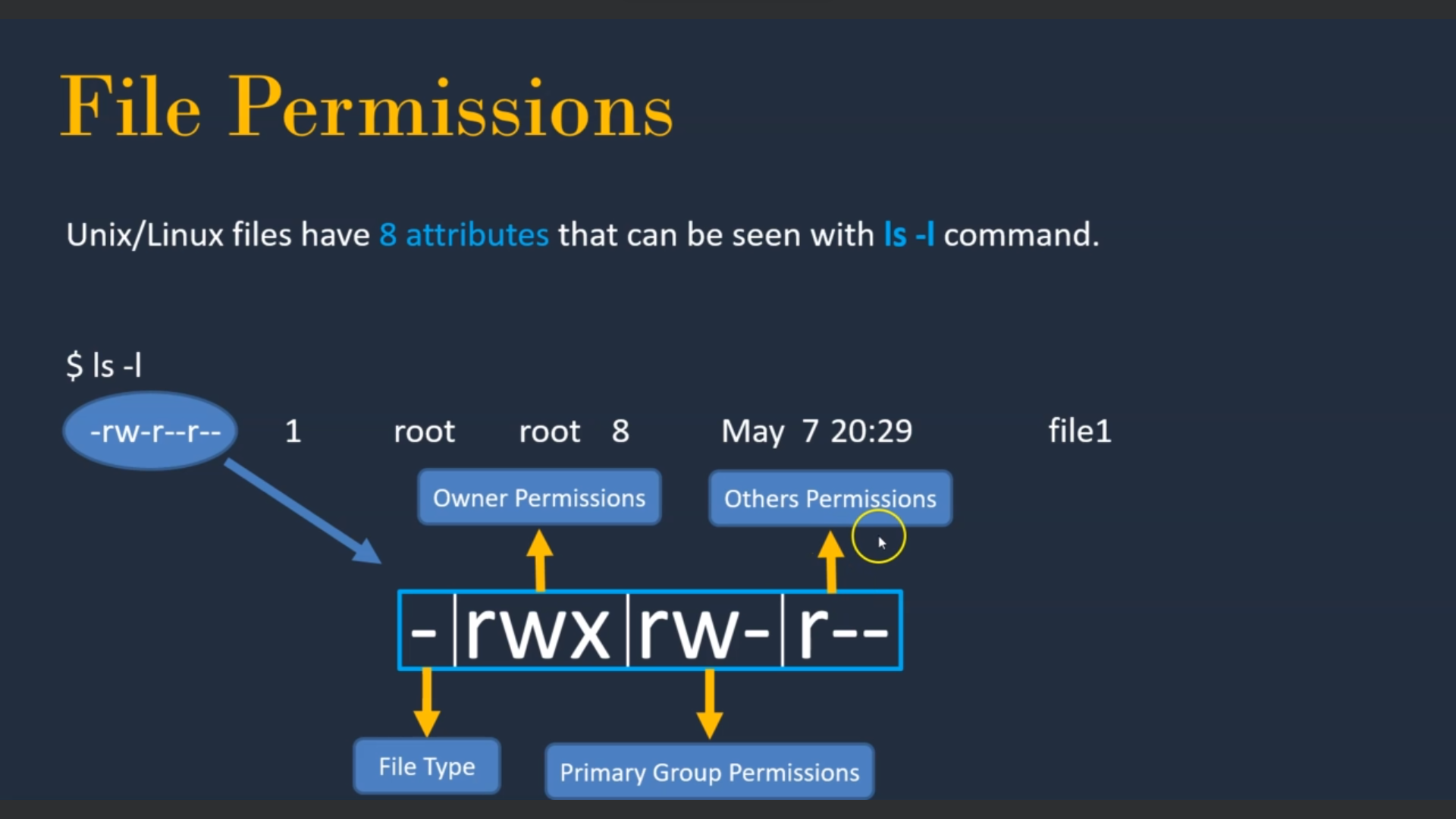
## Example: -

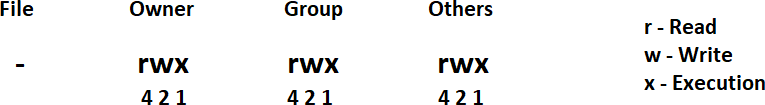
* + useradd nani
  + Working directory for nani user is **/home/nani**. All the user’s working directory is

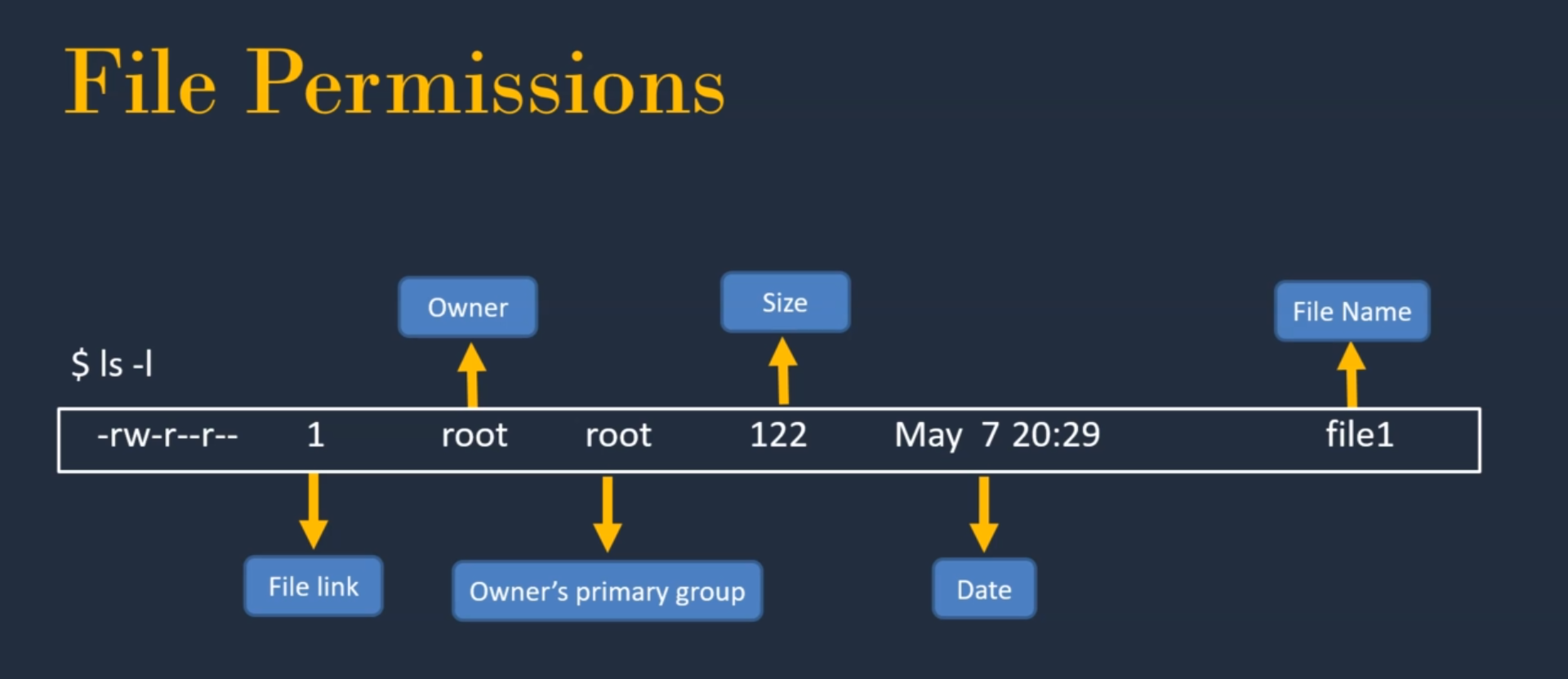
### /home

* + If we created a user automatically a group has been created with user name uid=1001(nani) gid=1001(nani) groups=1001(nani)
  + Enable the password base authentication in **/etc/ssh/sshd\_config** file as yes and reload or restart the sshd service. **service sshd restart/reload**.
  + If you want see the users **cd/home** or **cat /etc/passwd**

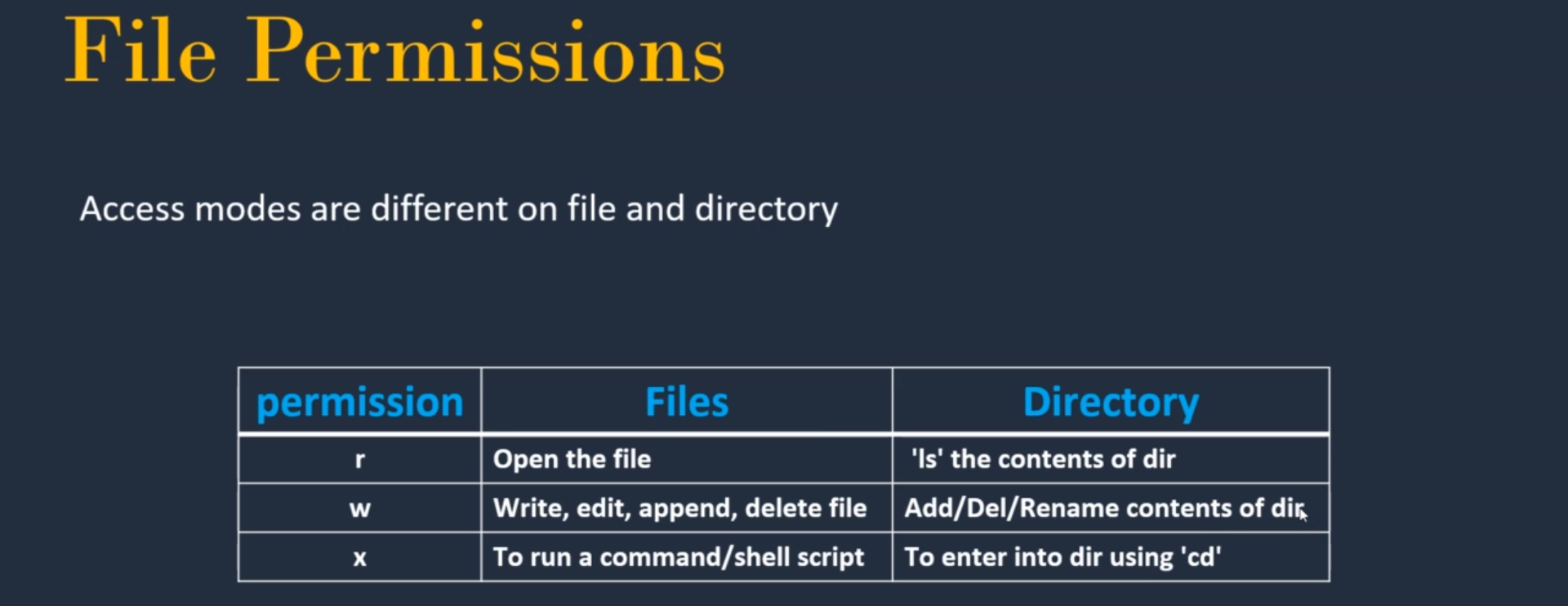
## Permissions to a file/directories: -

****



****

**File permissions on Directory: -**

****

**Chown Command: -**

* It is changing the ownership of the file. It is executable by **root** user only.

## Syntax: -

chown <username> <filename> It is changing the owner of a file.

## Example: -

chown gayathri file1

* It is changing the owner and group access of the file.

## Syntax: -

chown <username> <groupname> <filename> It is changing the permission of the owner.

## Example: -

Sudo chown gayathri:gayathri file1 >>Now the file owner and group owned by **Gayathri User Modifications: -**

### Options: -

* + -u user id
  + -G Secondary group id
  + -g Primary group id
  + -d Home directory
  + -c Comment
  + -s Shell

## Example: -

* usermod -G gayathri nani

uid=1002(gayathri) gid=1002(gayathri) groups=1002(gayathri),1001(nani) Means nani user added to the gayathri group.

* usermod -g nani gayathri

uid=1002(gayathri) gid=1001(nani) groups=1001(nani),1002(gayathri) Means gid and groups are changed to nani user.

## Removing the Groups user: -

* gpasswd -d <Owner User> <Remove User>

## Example: -

* uid=1002(gayathri) gid=1002(gayathri) groups=1002(gayathri),1001(nani)
* gpasswd -d nani gayathri

## Error: -

Removing user nani from group gayathri gpasswd: user 'nani' is not a member of 'gayathri'

* gpasswd -d gayathri nani

## File Permissions: -

* We will use the numbers and also symbols. The permissions edited either owner or the root user.

|  |  |  |
| --- | --- | --- |
| Read | - | 4 |
| Write | - | 2 |
| Execute | - | 1 |

**Example: -**

|  |  |  |
| --- | --- | --- |
| * chmod | 777 | file1 |
| * chmod | 765 | file2 |

## System Management Commands

|  |  |
| --- | --- |
| **Command** | **Description** |
| history | List all commands executed by a user |
| free | Free memory of a server |
| /proc/meminfo | Display memory information |
| /proc/cpuinfo | Display CPU information |
| uname -a | Show kernel information |
| du | Show directory space usage |
| whereis | Show possible locations of app |
| which | Show which app will be run by default |

* + Free command will show the memory details in KB’s. If you want in MB’s then

execute **free –m** command.

total used free shared buff/cache available

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Mem: | 989140 | 134168 | 306204 | | 680 | 548768 | 712948 |
| Swap: | 0 | 0 | 0 | |  |  |  |
|  | total | used | free | shared | buff/cache | | available |
| Mem: | 965 | 131 | 298 | 0 | 535 | | 696 |
| Swap: | 0 | 0 | 0 |  |  | |  |

* + One more command for memory information **cat /proc/meminfo**
  + If we want to know the CPU information then use the command

### cat /proc/cpuinfo

* + du command shows the usage of the directories in Kb’s

4 ./.ssh

40 .

* + Whereis and which will shows that the commands executing path

### whereis ls

ls: /usr/bin/ls /usr/share/man/man1/ls.1.gz /usr/share/man/man1p/ls.1p.gz

### which ls

alias ls='ls --color=auto'

/usr/bin/ls

* + df command shows the file systems in the server. File system means like c-drive, d-drive…..etc and **df –h** will show the disk utilization.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Filesystem | 1K-blocks | Used | Available | Use% | Mounted on |
| devtmpfs | 485296 | 0 | 485296 | 0% | /dev |
| tmpfs | 494568 | 0 | 494568 | 0% | /dev/shm |
| tmpfs | 494568 | 808 | 493760 | 1% | /run |
| tmpfs | 494568 | 0 | 494568 | 0% | /sys/fs/cgroup |
| /dev/xvda1 | 8376300 | 1595524 | 6780776 | 20% | / |
| tmpfs | 98916 | 0 | 98916 | 0% | /run/user/1000 |
| tmpfs | 98916 | 0 | 98916 | 0% | /run/user/1002 |
| tmpfs | 98916 | 0 | 98916 | 0% | /run/user/1001 |

## Software Management

Yum is the primary tool for getting, installing, deleting, querying and managing RedHat Enterprise Linux RPM software packages from official RedHat software repositories, as well as other third-party repositories.

## Commands: -

* + yum install <package name>
  + yum remove <package name>
  + yum update <package name>
  + yum info <package name>
  + yum list available
  + yum list installed

## Networking

|  |  |
| --- | --- |
| **Command** | **Description** |
| hostname | lists host name of the server |
| ping <ip> | availability of destination server over the network |
| wget | download packages/software’s onto Linux system |
| ifconfig | lists IP addresses of the server |
| telnet | connect to remote host/check port available status |
| curl | access the application as from browser |

* hostname >> ip-10-1-1-74.ec2.internal

It is stored in **cat /etc/hostname** we will edit the host name as we need and restart the server. **init 6** is the command to restart the system. Don’t change this name in real-time because some DNS name are effected.

* ping [www.google.com](http://www.google.com/)

64 bytes from bk-in-f99.1e100.net (142.251.111.99): icmp\_seq=1 ttl=51 time=1.72 ms 64 bytes from bk-in-f99.1e100.net (142.251.111.99): icmp\_seq=2 ttl=51 time=1.84 ms

It is used that the destination/target server has available or not.

* wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.56/bin/apache-tomcat- 9.0.56.tar.gz

wget command is used to download the packages.

* curl <https://tomcat.apache.org/download-90.cgi>

curl is working as the browser in Linux. It is used that once deploy the application will check whether the server accessible or not.

|  |  |
| --- | --- |
| **Port Number** | **Service** |
| 21 | FTP |
| 22 | SSH |
| 23 | TELNET |
| 25 | SMTP |
| 53 | DNS |
| 80 | HTTP |
| 443 | HTTPS |

If you want to know which ports are used or which ports are running then use the command

### netstat -tulpn

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Proto** | **Recv-Q** | **Send-Q** | **Local Address** | **Foreign Address** | **State** | **PID/Program name** |
| tcp | 0 | 0 | 0.0.0.0:**111** | 0.0.0.0:\* | LISTEN | 2558/rpcbind |
| tcp | 0 | 0 | 0.0.0.0:**22** | 0.0.0.0:\* | LISTEN | 3217/sshd |
| tcp | 0 | 0 | 127.0.0.1:**25** | 0.0.0.0:\* | LISTEN | 2989/master |
| tcp6 | 0 | 0 | :::**111** | :::\* | LISTEN | 2558/rpcbind |
| tcp6 | 0 | 0 | :::**22** | :::\* | LISTEN | 3217/sshd |

* telnet is used that the port numbers are used or not for localhost or the target system/server.

### telnet localhost 8080

Trying 127.0.0.1...

telnet: connect to address 127.0.0.1: Connection refused

### telnet localhost 80

Trying 127.0.0.1...

Connected to localhost.

# Services

* This controls the starting and stopping of services.
* Service <name of the service> status
* Service <name of the service> start
* Service <name of the service> stop
* Service <name of the service> reload
* Service <name of the service> restart

# Chkconfig

* This controls which services are set to start after reboot the server. It is mainly used while creating the AMI in AWS
* chkconfig --list >> To check the availability of services
* chkconfig <service> on >> to make the services available after restart.
* chkconfig <service> off >> to make the services unavailable after restart.

**chkconfig httpd on**

# Process Management

* When you start a program or running an application in Linux, it actually run as a process
* A Linux process (a daemon), running in foreground or in the background, uses memory and CPU resources.

|  |  |
| --- | --- |
| **Command** | **Description** |
| ps -ef | List the process which are running in the system |
| kill / kill -9 | Kill a process or Service |
| fg | Run the program in the foreground |
| bg | Run the service in the back group |
| top | List top 20 process which are consuming more CPU |

* ps –ef is like taskbar in windows.
* kill command has kill the process >> kill <processed> kill 11256
* kill -9 is sure kill the process. In the worst cases will use the kill command. kill -9 17564
* bg command run the programs in the backend. Sleep 500 >> Curser waiting for 500 seconds Ctrl+Z >> stop the service

Jobs >> is the command that what are the jobs are executing. bg %<number>

* fg command will run the bg programs in the front-end. fg %<number>

# Archiving files and directories

|  |  |
| --- | --- |
| **Command** | **Description** |
| gzip | Create a compressed file |
| gunzip | Unzip a file |
| tar | Extract tar file |

* gzip menas reduce the size of the file.
* tar command will works on directories. If you want to zip the directory then use the command **tar -cvf <destination file name> <source filename>**

**c = compress**

**v = verbose**

**f = file**

### tar -cvf 789.tar 789

* Then it is created a compressed verbose file and still we have the source directory also. The cvf file will be converted as a zip file because it is a file.
* If you want to unzip the directory then use the command **tar –xvf 789.tar**
* If you want to unzip and also extract the file then we will use the command

x = extract

v = verbose

f = file

**tar -xvzf apache-tomcat-9.0.56.tar.gz > It will do unzip and extract**

# Crontab

* It is used for scheduling a job.

### Commands

crontab -l crontab -e

Execute a job at 8:30 on everyday morning 30 8 \* \* \* <command>

Execute a job at 2:00 PM on every Saturday 00 14 \* \* 6 <command>

Execute a job at 12:00 AM on 1st July 00 00 01 06 \* <command>

Execute a job at 3:30 PM on Every month 25th 30 15 25 \* \* <command>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Min | Hours | Date | Month | Day of the week |

# Link Files

There are two types of link files, one is **Soft link** and the other one is **Hard link.**

|  |  |
| --- | --- |
| **Soft Link** | **Hard Link** |
| Shortcut File | Backup File |
| Size of link file is equal to no.of characters in the name  of original file | Size of both file is same |
| If original file is deleted, link is broken and data is lost | If original file is deleted then also link will  contain data |
| Command: ln -s <sourcefile> <destinationfile> | Command: ln <sourcefile>  <destinationfile> |

* If we want link file then use the command **ln -s file1 file1.link.** Then it creates a file, reduced the size and mapping to the source file. If we delete the source file then the linked file also deleted.
* If we want link file then use the command **ln file1 file1.link.** Then it creates a link file, with same size. If we delete the source file then the linked file also deleted.

# Copy file between Servers

* Windows to Linux

Tools: Mobaxterm or winscp >>drag and drop the file

* Linux to Linux

SCP(secure copy) is a command-line utility that allows you to securely copy files and directories between two systems.

### scp <source file name> <username@destination host>:<destination folder>

* **Example:-**

scp file1 root@10.1.1.100:/tmp

scp root@10.1.1.100:/tmp /home/ec2-user/ scp file5 nani@10.1.1.100:/home/nani

scp file2 ec2-user@10.1.1.200:/home/ec2-user >>this will execute remote to local scp nani@10.1.1.200:/home/nani/file2 /home/ec2-user >>pull request from local

The above two commands are same.

* If you want to copy the directory then use the command scp -r dir1 nani@10.1.1.100:/home/nani/

SSH Authentication

Generate keys

Ssh-keygen

Ssh-keygen -t rsa -b 4096 >> generating more secure key

**The End….!!**

scp -i Trust\_pvt\_openssh /home/ubuntu/test.pcap ubuntu@10.0.1.250:/home/ubuntu/test.pcap