Linux Commands

1. Ls – list of all directories  
   ls -al list of all hidden files and directories.
2. Which – command is located where.
3. $ for another user & # for root user.
4. Ls --help- list of all help option of this command.
5. Pwd-print work directory(directory in which you are currently in).
6. **Clear or ctrl+l** – clear the terminal.
7. **Ctrl +d**- for exit.
8. Exit-exit the terminal.
9. Ls -ld-list of all directories
10. Ls -ld Desktop/ list of all directories in desktop folder.
11. Ifconfig- know the ip information of server.
12. Use putty terminal for ssh connections in linux – ip information, port no-22 then password .
13. Total terminals tty1-6
14. Chvt – to change the terminal type
15. Chvt 3,4,5,6-for cli terminal type and chvt 2- GUI terminal type and chvt 1 for login shell.
16. W- list of all login terminals
17. Logout-logout the user.
18. Echo- to print anything.
19. Poweroff and reboot-to poweroff and restart the machine.
20. Whoami-name of current user.
21. Id-list the userid, groupid of user.
22. Date-check the date.  
    date +%R-shows time.  
    date +%X-shows time in detail.
23. Cal-calender shows.
24. **Passwd root**-change the password of user.  
    **gpasswd – d username groupname**– delete the user from the group.
25. Man command- description of command.  
    q-quit the command  
    spacebar- come for next page.  
    Reading man pages  
    NAME- Subject name, usually a command or file name.  
    SYNOPSIS- Summary of the command syntax.  
    DESCRIPTION-In depth description of basic understanding of a topic.  
    OPTIONS-Explanations of a command options.  
    EXAMPLES- examples of how to use the command function.  
    FILES-a list of file or directories related to man.  
    SEE ALSO- related information other than man command.  
    BUGS- Know bugs in the software.  
    AUTHOR-Information about the contribution in the topic.
26. W- shows the list of users and what they are doing right now.
27. File /etc/var-check file type.
28. Cat – see the contents of a file.
29. Head -2//head -n 2 – see the first 2 lines of a file.
30. Tail -2// Tail -n 2 – see the bottom 2 lines of a file.
31. Wc- shows line , words and characters in a file.  
    wc -l shows no of lines.  
    wc -w shows no of words.  
    wc -c shows no of characters.
32. Use Tab- for tab competition.
33. Gedit- gnome editor.
34. Stdin/stdout-  
    0- for input  
    1- for output  
    2- for error  
    3+ for filename  
      
    > filename- redirectional or save a output in a file.  
    >> filename- append the output in an existing file.  
    2> filename – save an error in a file.  
    2> /dev/null >filename – temporary save an error in a file and output save in given filename  
    &>filename- save output and error in a filename.  
    &>>filename-append the output in a filename.
35. Pipelines I-join one or two or more commands in a single line.
36. Tee command – ls|tee filename.txt gives the output in a file and also shows the output on screen.
37. VIMEDITOR- Vi command- used to open vim editor   
    vimtutor- used to read the manual and options of vim editor.  
    press esc to exit the command mode.  
    i for insert mode  
    :q! without save  
    :wq continue with save file  
    gg go on first line.  
    shift +g go on last line.  
    0 go on first character on current line  
    $ go on last character on current line  
    managing local users and groups:
38. Sudo for super user  
    id=0,gid=0,Oid=0
39. Id shows id of a user.
40. Note:- superuser id are 0-999  
    localuser- from 1000.
41. Su -userame // su username – if we give hyphen root user are directly go into user home directory otherwise in current directory.
42. Useradd- add a user.
43. Userdel- delete a user.  
    userdel -r delete a user with their roots and directories.
44. Passwd- make a password of user.
45. Su – login in root user.
46. Ll- check who made the directories.
47. /etc/passwd- shows no of users in system and all information about users.
48. Ps - to check the no of process and their process id (PID).  
    ps -au check no of users with their login process id.
49. /etc/passwd- store information about users.  
    /etc/shadow- store info about user password.  
    x- user has password & !- user don’t have password.  
    /etc/group- store info about groups.  
    /etc/gshadow- store info about group password.
50. Grep- to find something in a file or a folder.
51. User properties :  
    user:x:1003:1003::home/user:bin/bash  
    user- username  
    x- userpassword store in encrypted format  
    1003- UID  
    1003- GID  
    ::- display the real name or display name.  
    /home/user- display the home directory of a user.  
    bin/bash- login shell.  
    /sbin/nologin- no login shell.
52. Groupadd-create a new group.
53. Groupdel- delete a group.
54. Groups- check the group of current user.
55. Groupmod- change group properties.  
    gropupmod -- help check options of group properties.  
    gropupmod -- n change groupname  
    gropupmod -- g change groupid.
56. usermod- change user properties.  
    usermod -- help check options of user properties.  
    usermod - c give comments to a user.  
    usermod -s /sbin/nologin restrict the user from no login.  
    usermod - d change user home directory.  
    usermod -m - d change and move user home directory.  
    usermod -g change groupid of a user.  
    SHADOW PASSWORD POLICY
57. password stored in /etc/passwd file.  
    Separated by :  
    1) name of user  
    2) encrypted password   
    3) the day when password was last changed.  
    4) Password elapse details.  
    5)the maximum no of days that can pass without a password change before password expires.  
    6) warning period before password expires.
58. Format of a encrypted password   
    Separated by $  
    1) hashing algorithm  
    2) salt  
    3) encrypted has- mix of salt and algorithm.
59. Chage -password aging  
    chage -d 0 username users needs to change password with login.  
    chage -l username user password aging details.  
    chage -E 2023-08-25 username user password expiry date given.  
    SUPER USER ACCESS
60. Any user give root access from two types :  
    \* add into wheel group- useradd - add a user  
    \* password of it echo “123”|passwd username- add a password to user  
    \* usermod -G wheel username- add a wheel group to user.
61. Add in sudoers file – vim//nano /etc/sudoers
62. username ALL=(ALL) ALL, then save and exit.  
    :wq! – forcefully save and exit.
63. Useradd -r create a system account.  
    FILE CONTROLLING ACCESS
64. R-read, w- write, x -executable.
65. – for regular file  
    d – for directory  
    l- for soft link.
66. Chmod- changing the permissions of a file.  
    \*changing permissions with symbolic method  
    U- user, G- group, o-other, a-all  
    r-read, w-write, x-execute  
    + add permission,- remove permission,= exact permission.  
    for ex-chmod u+rwx  
    g+rw  
    o+r.
67. Changing permission with numeric method  
    start with 0  
    4-read permission  
    2- write permission  
    1-execute permission  
    let’s combine the permission  
    4+2=6= rw  
    4+2+1=7=rwx  
    0= for null permission  
    chmod 777 for all permission  
    chmod 000 for null permission.
68. Umask 001 – for by default permissions  
    for directory-777  
    for file-666   
    for directory-777-001=776 so by default permission of a directory is 776 means full permission for user, full permission for group , read and write permission for others.
69. Umask id for root user -0022  
    for regular user 0002.
70. Chown username filename/directory changing ownership of a file.
71. Chgrp username filename/directory changing group ownership of a file.
72. Chown username:groupname filename change user and group ownership in single way.
73. Special permission denoted by s  
    chmod u+s, g+s to add permissions  
    chmod u-s, g-s to remove permissions.  
    chmod +t for sticky bit  
    MONITORING AND MANAGING LINUX PROCESS
74. Ps- check process command.  
    PID(Process ID) TTY(tele Type terminal) TIME(running time) CMD(command location)  
    ps -a- all process  
    ps -au all process with user details.  
    ps -aux all process with executable services  
    ps -ef extra deomon process and root services also with PPID (Parent Process ID).
75. Kill PID kill the process.  
    kill -9 PID to kill process forcefully.
76. Man ps for different types of options of ps command
77. Top command for showing running processes.  
    f//ff used to check abbrevations of terminologies in command.  
    q go to process and again q to exit from command.  
    h go to help.
78. Pstree process structure in tree architrecture.  
    CONTROLLING JOBS
79. Jobs to see all the current jobs  
    ctrl+C for exit the job  
    ctrl+Z for pause the job.
80. Bg index\_no (come from jobs command). for see jobs in background.  
    fg index\_no (come from jobs command) for see jobs in frontend or foreground.
81. Sleep 10 for sleep the terminal.
82. Pgrep -l -u username- check all running process of user.
83. Pkill -u username kill all running process of a user.
84. Kill all process name- all processes kill related to given process name.
85. Uptime shows uptime.
86. Lscpu hardware information.  
    CONTROLLING SERVICES AND DEAMONS
87. Systemd is the first service that can runn in linux to start deomons and services.
88. Pidof systemd check the process id of a service.
89. Systemctl list-units list of all system services.
90. Systemctl system command.
91. Systemctl list-units --type=service check the list of all services.
92. Systemctl list-sockets check the list of all sockets.
93. Systemctl list-unit-files check the details and state of files related to services and deamons.
94. Systemctl status firewalld.service check the status of service name.  
    LOADED-Wheather the service is load in the memory or not.  
    STATUS- status of service.  
    PID-Process ID of service.
95. Systemctl stop firewalld.service stop the service.
96. Ps -p PID check the service detail on PID no.
97. Systemctl is-failed firewalld.service check the status of service.  
    CONTROLLING SERVICES
98. Reboot -f forecefully reboot the machine.
99. Systemctl restart firewalld.service restart the service and change PID also.
100. Systemctl reload firewalld.service restart or reload the service but still PID same , some services doesn’t support reload command.
101. Systemctl list-dependencies firewalld.service check dependencies and services of given service name.  
     Sudo systemctl list-units check all daemons services.
102. Systemctl mask firewalld.service immediately stop the service and send into /dev/null.
103. Systemctl unmask firewalld.service unmask the service.  
     SSH(SECURE SHELL) SERVICE-REMOTE
104. \*ssh is a terminal based service it can access only through terminal.(service name-sshd.service)
105. Ssh root@IP/hostname then password to access the terminal remotely.
106. Sudo su -l root forcefully login in root user .
107. Ssh-keygen for generate ssh key for one time access   
     by default private key is store in /root/.ssh/ir\_rsa and public key is stored in /root/.ss/id\_rsa.pub .
108. Cat /etc/ssh/sshd\_config  
     **DenyUsers**- usersname  
     **AllowUsers**- usersname   
     NMCLI COMMANDS
109. **Hostname//hostnamectl**- check hostname of server.
110. **Hostnamectl set-hostname redhat-linux** - change hostname.
111. **Name of lancard** – ifconfig then ens 160.
112. for use ssh login first add the hosts in file   
     vi /etc/hosts  
     then **ip of host name**
113. **netstat-** network stuff.
114. **Ss –** session stuff.
115. **Who -**who else logging in system.
116. **Env -**enviroment variables.
117. **Lsusb -**any usb plugged in.
118. **Lsof –** list of all open files.
119. **Apropos –** related commands to the working. **MANAGING LOGS AND MESSAGES**
120. all logs stored in **/var/log**.  
     SOME IMPORTANT:
121. /bin/ -all commands are stored in bin/ folder.
122. /sbin/- Super bin all commands of super user.
123. /usr/- user binary files and libraries are here.
124. /root – root user home directory.
125. /home/- user home directory.
126. /dev- all hardrive CD roms etc in this file.
127. Nmve/sda/vda – hardrive files.
128. **Lsblk**- list out all block devices .
129. /etc/ - network and configurations files.
130. /var/ - all log files in it .
131. /etc/NetworkManager/system-connections – this file give complete network information.
132. /mnt/- mount files that made manually.
133. /media/- gives the mounted flashdrives, pendrives etc in a machine.
134. **Uname -a –** print all the information related to OS .  
     **REDHAT PACKAGE MANAGER**
135. **Sudo yum repolist –** see the repolist of enabled rpm.  
     **sudo yum repolist all** see all the enabled and disabled repositories.
136. **Subscription-manager register-** register your rhel account.
137. **subscription-manager list –available** see the available rhel susbscriptions list.
138. **Cat /etc/yum.repos.d/redhat.repo** - all yum repositories located but from start all are disabled.
139. **Subscription-manager list –consumed** see the current subscription.
140. **Yum What provides yum-config-manager** what provides command to check which package need to run the command yum-config-manager.
141. **sudo subscription-manager repos --enable=<repo-id>** to enable any repository.
142. **sudo subscription-manager repos –disable=<repo-id>** to disable any repository **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
143. **sudo systemctl list-units** list all active daemons and their services.
144. **Sudo systemctl list-units -t service** List all active services.
145. **sudo systemctl list-units –all** list all active and inactive services.
146. **sudo systemctl list-unit-files**  list all the files which are load or unload in memory during boot.
147. **Pgrep <process name>-** combined command of process and grep command to find the PID of any process.
148. **Pkill <process name>-** combined command of process and kill command to kill any process.
149. **Ps -u avinash|grep <process name>**  to find the PID of process.  
     **CLIENT URL**
150. **Curl -o <filename>**  download the file name with given name.
151. **Curl -O –**  save the file with their actual filename on remote server.
152. **Curl -C – -O** continue the fileprocess with same filename and append .
153. **Curl -I** [**www.google.com**](http://www.google.com) check response and header of server.
154. **Alias <name>=”<command>”** set the alias of the command.
155. **Alias -**check all the alias.
156. \*by default alias is temporary once you logout alias are deleted   
     so make it permanent input in .bashrc file   
     **nano .bashrc**.
157. **Ctrl +A**  go to beginning of a line.
158. **Ctrl + E** go to end of a line.
159. **Ctrl + U** delete everything before your cursor.
160. **Ctrl + Y** paste deleted part .
161. **Ctrl + K** delete everything after your cursor.
162. **Alt + backspace** delete the last word only.  
     \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
163. **Mkdir -p /opt/projects/{development,analysis,common}** to make a more than one folder in single click.
164. **~/.ssh/Know\_hosts.old** files provide the information of hosts recently make a connection with the client public key.
165. **/etc/hosts** files works as DNS server tables to store information of hostname with their IP helps in ssh connect with hostname.
166. For given ssh acces to root change it in ssh configuration file  
     **/etc/ssh/ssh\_config** always take backup of the file and then change it.  
     **PermitRootLogin yes** for given access to ssh root login.  
      **PermitRootLogin no** for not given ssh root login.
167. For public keygenrator   
     **ssh-keygen** for key generation. ~/.ssh/id\_rsa.pub create and public key in it.  
     **ssh-copy-id <host ip>** after that there is a file named authorized key **~/.ssh/authrorised key** – automatically copy public key in this folder.
168. **SELinux** (Security Enhanced Linux)– label – context   
     4 types of SELinux-  
     1) user base (ends with \_u)  
     2) rule base (ends with \_r)  
     3) type base (ends with \_t)  
     4) sensitivity base (S0,S1)  
       
     In red hat we use type base context or to find it **ls -Z <filepath>  
     ls -dZ <directory path>**SELinux have 3 modes -  
     1) Enforcing- obey rules and deny access on violation.  
     2) permissive-obey rules and on violation allow access with log generation.  
     3) Disabled -SElinux is disabled.  
     \*to check in which mode currently SELinux is in **getenforce**.  
     to change the SElinux mode temporary   
     **setenforce <choose options> -** temporary change.  
     to change permanent , change in configuration file.  
     file location - **/etc/SElinux/config.**  
     SELinux=enforcing change it to make changes permanent.  
     to change context :-  
     **chcon -t <context name=default\_t> <filename>**to undo the context. **restorecon –Rv <filename>.**
169. **Semanage fcontext -a(add) -t(type) <contextname=default\_t> <filename>** another command to change the context of a file.
170. **SAMBA server-**used to share linux files in one or more operating system over the network.  
     protocols used in it -  
     1) SMB (Server message block)  
     2) CIFS ( Common Internet filesystem)  
     need 3 packages to install samba server-  
     **yum install samba samba-client samba-samba-common**now enable firewalld because firewall block sharing of files over the network-  
     **firewall-cmd -–permanent --zone=public --addservice=samba  
     firewall-cmd --reload**create a directory which you want to share   
     **mkdir -p <filepath=/samba/apps/samba\_server>** set all permission to the file and folder  
     **chmod 777 <filepath>  
     chmod 777 <filepath>**now change the context of file-  
     **chcon -R -t <choosecontext=samba\_share\_t> <filepath//filename>**for private acces – need to create a user or their samba password **smbpasswd -a <username>** now change in configuration file-  
     path - **/etc/samba/smb.conf** first take the backup of this file.  
     then-  
     **SMB FILE CONF**:  
     { [global]

map to guest = Bad User

[folder]

path = /samba

browsable = yes

writable = yes

guest ok = yes

read only = no }

And for protected with samba user

{

map to guest = Never

[folder]

path = /samba

browsable = yes

writable = yes

#guest ok = yes

read only = no

valid users=<username> }

then **systemctl restart smb nmb to start the service.  
  
(for linux to linux – a) make a directory on client machine.  
to mount it---- mount -t cifs //<serverip> /shared\_directory path of client machine (/home/manish/mani) -o username=<username> ,** **vers=3.0  
  
for change the SMB version permanent -  
/etc/samba/smb.conf [global]**

**min protocol = SMB2**

**max protocol = SMB3**

and to check that mounted or not   
**df -Th. (This is temporary)**

1. **(for Permanent) - vi /etc/fstab**then make a entry in it -  
   //192.168.1.26/avi /home/manish/mani defaults,username=<username>,password=<sambapassword> 0 0
2. **For windows to linux – test network with ping command   
   then , //<serverip>  
   smb username -   
   smb password-   
   \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
3. **DISK MANAGEMENT - lsblk -**  check the disk partitions and their usage and space.  
   init 0 = to turn off the linux system  
   init 6 = to restart the system.  
   Select a disk ---  
   then check with **lsblk** command  
     
   then **fdisk /dev/<hardisk dir name sda>**- for SATA **m** for help.  
   n for new partition   
   p- primary , e- extended  
   first select type – **p, e**  
   select partition no – **1,2,3,4**  
   select storage – Space   
   then enter and partition is created   
    **p-**  print the partition table.  
   **l** - list all the partition types.  
   **w** – to save the changes.  
     
   \*when the command is not working when using pendrive then.  
   use command **partprobe.**  
   **\* df -t** command to check partitions in human readable format.  
   \***df – Th**  also gives information of file system and its type  
     
   **blkid** to check id’s of a parttions that file system of that partition is created or not .  
   for make file system of a partition  
   **mkfs.<files ystem name=ext4> path= /dev/sda1  
   mkfs.<files ystem name=ext4> path= /dev/sda2  
   mkfs.<files ystem name=ext4> path= /dev/sda3  
   mkfs.<files ystem name=ext4> path= /dev/sda4**  
   mountpoints with another directory  
   make a dir  
   then **mount <filesystem\_path=/dev/sda1> <location\_path>**   
   (this is temporary mount)  
     
   for permanent mount   
   **add in file /etc/fstab**  
     
   **/dev/mapper/rhel-swap none swap defaults 0 0  
     
   umount -a** to unmount all   
   \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
4. **LVM- (Logical Volume Manager)**helps to build a larger virtual space to combine multiple spaces **called volume group.**A single large space created from this virtual space called logical volume.  
   steps of LVM for adding new space:  
   1) take a disk.  
   2) make partitions.  
   3) From physical volume create volume group.  
   4) create and manage logical volume.  
   5) Apply a filesystem.  
   6) Set a mount point  
   check details of new disk -  
     
   **fdisk -l**make partition with above commands. **\*** to create physical volume   
   **pvcreate /dev/sda1**  
   **pvdisplay** – to display the physical volume.  
     
   \* to create volume group   
   **vgcreate <vgname=vgapps> /dev/sda1**  
   **vgdisplay** – to display the volume group.  
     
   \* to create logicalvolume  
   **lvcreate -L(size – 1T or 1G) -n <name> <name of volume group>**now setup the file system  **mkfs.ext4 /dev/vgapps/vgapps  
     
   mount /dev/vgapps/vgapps-1 /apps-1**  
   **Extending a disk using LVM :**  
   first instal a new hard disk  
     
   then doing procedure for making till physical volume created,  
     
   then, **vgextend <volumegroup\_name=vgapps> <path of physical volume= /dev/sda1>**  
     
   Add space to LV:  
   **lvextend -L +1G <Lvname=/dev/vgapps/vgapps-1 >**  
     
   then resize the partition :  
   **resize2fs /dev/vgapps/vgapps-1 <total size =2G>**
5. **ACCESS CONTROL LIST**ACL is a list of permission that defines access of users on a file or directory.  
     
   **getfacl** get ACL permissions on a file or directory  
   **setfacl** set ACL permissions on a file or directory  
     
   for user permission   
     
   create a file   
   **setfacl -m(modify) u:<name of user>:---(permissions) <filename>**  
     
   then relogin the user   
   check permissions will work.  
     
   same for directory.  
   **setfacl -m(modify) g:<name of group>:---(permissions) <filename>**\*In ACL user permissions are overwrite group permission.  
     
   to delete a single ACL:  
   **setfacl -x u:<username> <filename>**  
     
   to delete all ACL’s:  
   **setfacl -b <filename>  
     
   Default ACL’s -** if you can set ACL on a single file after then all ACL’s are set by default on all the files.  
     
   **setfacl -dm(modify) u:<name of user>:---(permissions) <filename>**  
     
   for copy the ACL’s   
     
   **getfacl --access <filename> | setfacl -M- <new filename>**  
     
   for copy ACL through files.  
   **getfacl file1 | setfacl --set-file=- /abc/file2  
   \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
6. **NFS (Network file system)** -  
     
   packages to install – nfs-utils , rpcbind, libnfs -idmap  
   to check the nfs status  
   **systemctl status nfs-server**Create a directory you want to share -  
     
   now make an entry of that file in **/etc/exports**.  
     
   <share path> 192.168.1.0/24 (for complete network ) 192.168.1.25/24(for particular system)(rw,sync,no\_root\_squash)  
    **(permissions)   
      
    (realtime,sync)   
     
     
    (no\_root\_squash)-same root permissions.  
    (root\_squash)-root permissions aren’t same.**then **:wq** to save and exit.  
     
   start nfs-server with systemctl command.  
   check configurations in export file **exportfs -r** if there is no error then it’s ok.  
   file is share or not check – **showmount -e  
   (please note start nfs-server and rpcbind before it with systemctl command)**then go to client system   
   **check showmount -e <server ip>  
   (please note start nfs-server and rpcbind before it with systemctl command)**then mount (for temporary) **mount -t nfs 192.168.1.26:/data /nfs**for permanent mount entry in /etc/fstab file.  
   \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
7. **LINUX FIREWALL SERVICES**  
   firewalld - firewall Daemon  
   firewall works on the basis of rules , it filter on the basis of network traffic.  
     
   Firewall can be manage of two types :  
   1) GUI   
   2) CLI   
     
   firewall configuration are of two types -  
     
   1) RUNTIME – Temporary   
   2) PERMANENT -   
     
   to install GUI based firewall   
   packages- **firewall-config**  
     
   to work with firewall in CLI always used **firewall-cmd  
     
   firewall -cmd -–get -services-** to list all the services  
   **firewall -cmd -–list -services-** to list all the running services in def zone.   
   **firewall -cmd -–get -zones-** to list all the zones.  
   **firewall -cmd -–get -active -zones** to list all the active zones  
   **firewall -cmd -–get -default -zone=<zone name>** to set the default zone  
   **firewall -cmd -–add -service=http -zone=<zone name>** to add a service in given zone.  
   **firewall -cmd -- list -ports** to check the added ports apart from services ports that were added already.  
   **firewall-cmd -–add -port=<port no >/<protocol name> -zone=<zone name>** to add a service in given zone.  
   **firewall-cmd -–add -interface=<interface no > -zone=<zone name>** to add a interface in given zone.  
     
   \*if something add in default zone , no need to write (**-zone=<zone name>**)  
     
   **firewall-cmd -–reload** reload the firewall.  
     
   \*all changes are runtime or temporary , but for permanent changes  
   always used **firewall-cmd –permanent** before the commands.  
     
   \*to remove the firewall use **remove** keyword at the place of **add**.  
     
   \*add zone is not in CLI mode , you can use GUI mode for it.  
   \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
8. **TAR COMMAND**used for compressed or decompressed files.  
   (an archiving utility)  
     
    **tar -cvf <new archive file name> <list of all files and dir>**

createuse archive file or device archive.

verbosed see the process and files listing.  
 **tar -tf <file name> -** list the files which were archived.  
to extract the files   
  
make a dir mkdir <dir name>  
then go to dir  
  
**tar -xvf <tar file path>**Execute

**for zip files   
  
tar -czvf <filename> <file path>** to make a zip file.  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. **FIND COMMAND**to search anything in linux.  
     
   **find / -type f -name <filename> (**for case sensitive)  
   **find / -type f/d -iname <filename> (**for case in sensitive)

file directory

**amin** access before minutes  
**cmin** created before minutes **mmin** modified before minutes **atime** days **ctime** days **mtime** days  
  
**find / -cmin 5** files search which were before 5 minutes.  
**find . -cmin 5** for current directory.

current directory.  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. **CRON TAB**used to schedule a command , generally used for backups.  
   \*schedule jobs called – cron jobs  
   \* tables store the details called crontables>  
     
   to see the crontab details and formatting  
   go to **vi /etc/crontab** file.  
     
   **crontab -e** for edit the crontab  
   **crontab -r** for remove the crontab  
   **crontab -l** list the cronjobs of running user.  
   **crontab -u <user name> -l** list cronjobs of another user.  
   **crontab -u <user name> -e** edit cronjobs for another user.  
   \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*