

Linux Command

1) ls ↵

To list files and their contents.

Options:

- **-A**: Display all files, including hidden ones.
 - **-l**: Long listing format with detailed information.
 - **-s**: Print the allocated size of each file.
 - **-c**: Sort by change time (ctime).
 - **-h**: Display sizes in human-readable format (e.g., K, M, G).
 - **-r**: Sort in reverse order.
 - **-S**: Sort files by size.
 - **-R**: List subdirectories recursively.
 - **-i**: Show inode numbers.
 - **-lrt**: Sort files by modification date, with the most recent last.
 - **-lshr**: Sort files by size in human-readable format, displaying the smallest first.
 - **ls -lt**: Lists files in long format, sorted by modification time, with the most recently modified files first.
 - **ls -lt --time=creation**: Lists files in long format, sorted by creation time, with the most recently created files first.
 - **ls -lc**: Lists files in long format, sorted by change time (ctime).
-

2) cal ↵

To display a calendar.

Format:

```
{cal day month year}
```

Options:

- **1**: Display a 12-month calendar of the specified year.
 - **-1**: Display the current month.
 - **-3**: Display the previous, current, and next month.
 - **-s**: Display Sunday as the first day of the week.
 - **-m**: Display Monday as the first day of the week..
 - **-j**: Display Julian dates.
 - **-y**: Display the calendar of the current year.
-

3) date ↵

To display or set the system date and time.

Options:

- **-r**: Display the last modification time of a file.
-

4) mv ↵

To move (rename) files.

Usage:

Linux Command

(Source)	Destination)
<code>mv /home/abhijeet/file1</code>	<code>/home/afridi/file1 ↵</code>
<code>mv file1 file2 ...</code>	<code>dir1 ↵ (Multiple files)</code>

Options:

- **-f:** Do not prompt before overwriting.
 - **-i:** Prompt before overwrite.
-

5) cp ↵

To copy files and directories.

Usage:

(Source)	(Destination)
<code>cp -R /Abhifile1/files</code>	<code>/home/user/ ↵</code>

6) touch <filename> ↵

To create a file (it will create a zero-byte file).

Examples:

- `touch {file1,file2} ↵`
 - `touch dir1/file1 ↵`: Create a file inside the directory.
 - `touch File{1..10} ↵`: Create file1 to file10.
-

7) mkdir <directory name> ↵

To create directories.

Options:

- **-p:** Make parent directories.
 - **Usage:**
 - `mkdir -p dir1/1/2/3 ↵`
 - `mkdir {dir1,dir2}; touch {file1,file2} ↵`: To create directories as well as files.
-

8) rm <files> ↵ or rm -r <dir> ↵

To remove files and directories.

Options:

- **-r:** Remove directories and their contents recursively.
- **-f:** Never prompt before removing.
- **-d:** To remove empty directories.

Examples:

- `rm -rf *` ↵: Remove everything.
 - `rm -rf *file` ↵: Remove everything that has or ends with "file".
-

Linux Command

9) rmdir <dir name> ↵

To remove an empty directory.

Options:

- **-p**: To remove the directory and its ancestors.
-

10) echo "Any line of message" ↵

To display a line of text.

Options:

- **-n**: Print text/line without a newline at the end.

Examples:

- `echo $SHELL`: Display the current shell.
 - `echo $?`: Check if the last command executed successfully (returns 0 if successful).
 - `echo $DISPLAY`: Display information about connected displays..
-

11) Input/Output Redirection ↵

Redirection Operators:

- **'>'**: Output redirection (overwrites).
- **'<'**: Input redirection.
- **'>>'**: Append output to a file without overwriting.
- **'2>'**: Redirect standard error.
- **'&>'**: Redirect standard error and standard output.
- **'2>&1'**: Redirect standard error to standard output.

Examples:

- `echo "Msg" > file1`: Redirect the message into file1.
 - `echo "msg2" >> file1`: Append message 2 into file1.
-

12) cat file1 file2 ↵

To concatenate files and print on the standard output.

Usage:

- `cat file1 file2 ↵`: Display the contents of file1 and file2 together.
- `cat file1 ↵`: Display only file1's content.

Options:

- **-n**: Print line numbers.

Linux Command

13) find : Search for files and directories in a specified location.

Basic Usage:

```
find <path> [options] [expression]
```

Common Options:

- **-name** : Search for files by name.
- **-user** : Search for files owned by a specific user.
- **-uid** : Search for files using user ID.
- **-size** : Search for files by size.
- **-mtime** : Search for files modified in the last n days.
- **-atime** : Search for files accessed in the last n days.
- **-mmin** : Search for files modified in the last n minutes.

Useful Examples:

1. **Find a file named passwd:**

```
find / -name passwd
```
2. **Find files owned by user abhijeet:**

```
find / -user abhijeet
```
3. **Find the top 10 largest files:**

```
find / -type f -exec du -sh {} + 2> /dev/null | sort -rh | head -n 10
```
4. **Find all .log files:**

```
find /var/log -type f -name "*.log"
```
5. **Search for files by size:**

```
find /root/ -size <n>          # Exact size
find /root/ -size -<n>        # Less than n
find /root/ -size +<n>        # More than n
```
6. **Search for files modified within a specific time frame:**
 - **By modification time:**

```
find /root/ -mtime <n>          # Exactly n days ago
```
 - **By access time:**

```
find /root/ -atime <n>          # Exactly n days ago
```
 - **By modified minutes:**

```
find /root/ -mmin -<n>          # Modified within the last n minutes
find /root/ -mmin +<n>          # Modified more than n minutes ago
```
7. **Delete empty directories:**

```
find /path/to/directory -type d -empty -delete
```
8. **Find files containing a specific string:**

```
find /path/to/search -type f -exec grep -l "search_string" {} +
```

14) last : Display a list of last logged-in users.

Basic Usage:

```
last [options] [username]
```

Common Use Cases:

- **Display all logins:**

Linux Command

`last`

- **Show last login of a specific user:**

`last abhijeet`

- **Show system reboots:**

`last reboot`

- **Limit the number of entries:**

`last -n 5`

Displays the last 5 login entries.

Data Source:

- **Entries are retrieved from `/var/log/wtmp`:**
 - This file records all login sessions, including logins, logouts, reboots, and shutdowns.
 - Each entry logs when a user logs in or out, providing a history of user activity.

15) lastb : Display a list of last failed login attempts.

Basic Usage:

`lastb [options]`

Common Use Cases:

- **Display all failed login attempts:**
`sudo lastb`
- **Limit the number of entries (for example, to show the last 5 failed attempts):**
`sudo lastb -n 5`
- **Show failed attempts for a specific user:**
`sudo lastb | grep username`
- **Check failed password attempts in the secure log:**
`sudo grep "Failed password" /var/log/secure`

Data Source:

- **Entries are retrieved from `/var/log/btmp`:**
 - This file records all failed login attempts, logging details such as username, terminal, host, and timestamp.
 - Each entry provides information on unsuccessful login attempts, helping identify potential security issues.

16) tail : Display the last part of files.

Basic Usage:

- **Display last 10 lines:**
`tail <filename>`

Options:

- **-n: Print last n lines.**
`tail -n 20 <filename>`

Linux Command

- **-f**: Follow file growth. This option is useful for monitoring log files in real-time. It keeps the terminal open and displays new lines as they are added to the file.

```
tail -f <filename>
```

- **Example**: If you have a log file (e.g., `app.log`) that is constantly being updated, using `tail -f app.log` will show the last few lines of the file and automatically update the display as new log entries are added. This way, you can monitor the file without needing to refresh it manually.

- **Redirect output to another file:**

```
tail <filename> > /root/dest/file1
```

17) grep : Search for a pattern in a file.

Basic Usage:

```
grep "pattern" <filename>
```

Displays lines matching the pattern.

Common Options:

- **\$**: Match lines ending with a pattern.

```
grep 'test$' <filename>
```
- **#**: Match lines containing #.

```
grep '#' <filename>
```
- **^#**: Match lines starting with #.

```
grep '^#' <filename>
```
- **-i**: Ignore case.

```
grep -i "pattern" <filename>
```
- **-v**: Display non-matching lines.

```
grep -v "pattern" <filename>
```
- **-e**: Search for multiple patterns.

```
grep -e <string1> -e <string2> -e <string3> <filename>
```

Examples:

1. **Exclude lines starting with #:**

```
grep -v '^#' /etc/ssh/sshd_config
```
2. **Display lines starting with #:**

```
grep '^#' /etc/ssh/sshd_config
```

Linux Command

18) vim <filename> : To edit a text file (It is a text/notepad editor).

Vim Has Three Modes:

1. **Command Mode:** For navigating, copying, pasting, and deleting text.
2. **Insert Mode:** For inserting text. Enter by pressing `i`, and exit by pressing `Esc`.
3. **Visual Mode:** For selecting text.

Basic Commands:

- `:q` : Quit Vim.
- `:q!` : Quit without saving.
- `:w` : Save (write) the file.
- `:wq!` : Save the file and quit.

Editing Commands:

- `yy` : Copy (yank) the current line.
- `n yy` : Copy (yank) `n` lines.
- `dd` : Cut (delete) the current line.
- `n dd` : Cut (delete) `n` lines.
- `p` : Paste below the cursor (lowercase `p`).
- `P` : Paste above the cursor (uppercase `P`).
- `G` : Go to the bottom of the file.
- `nG` : Go to line number `n`.
- `cw` : Change (delete and enter insert mode) from the cursor to the end of the current word.
- `cc` : Replace the entire line.
- `ctrl + u` : To undo the changes
- `:%d` : To delete all the lines.

Searching and Replacing:

- `:/word` : Search for a word.
- `:?word` : Search backwards for a word.
- `:%s/old-word/new-word` : Replace the first occurrence of `old-word` with `new-word`.
- `:%s/old-word/new-word/g` : Replace all occurrences of `old-word` with `new-word`.

Miscellaneous:

- `!:<command>` : Run an external command (e.g., `!:ls` to list files).
- `:set nu` : Display line numbers.
- `:set nonu` : Remove line numbers.

Example:

`seq 100 > bigdata.txt` : Create a file `bigdata.txt` with numbers 1 to 100.

19) sed : Stream editor for filtering and transforming text.

`sed` works on line numbers and words.

Linux Command

Syntax Explanation:

- s/Search Pattern/Replacement/Flags
 - / : Delimiter
 - i or I : Ignore case.
 - g : Global (replace all occurrences in each line).

Basic Usage:

1. Delete the 1st line:

```
sed '1d' /etc/passwd
```

This will delete the 1st line of the file and print the output.

2. Replace the first occurrence of a pattern in each line:

```
sed 's/Ravish/Aavesh/' /etc/passwd
```

This replaces the first occurrence of Ravish with Aavesh in each line and prints the output.

3. Replace all occurrences in each line:

```
sed 's/Ravish/Aavesh/g' /etc/passwd
```

This replaces all occurrences of Ravish with Aavesh globally in each line.

4. Replace the 3rd occurrence in each line:

```
sed 's/Ravish/Aavesh/3' /etc/passwd
```

This replaces the 3rd occurrence of Ravish with Aavesh in each line.

5. Ignore case while replacing:

```
sed 's/ravish/Aavesh/i' text
```

The i flag ignores case, so both ravish and Ravish will be replaced.

6. Replace and create a backup:

```
sed -i.backup 's/Ravish/Aavesh/' /etc/passwd
```

This creates a backup of the original file (/etc/passwd.backup), replaces the first occurrence of Ravish with Aavesh, and saves the changes to the original file.

7. Remove blank lines:

```
sed '/^$/d' /etc/passwd
```

This removes all blank lines from the file.

8. Remove commented lines:

```
sed '/^#/d' /etc/passwd
```

This removes lines starting with # (comments).

9. Remove blank and commented lines:

```
sed '/^#/d; /^$/d' /etc/passwd
```

This removes both blank lines and lines starting with #.

10. Using multiple expressions with -e:

```
sed -e '/^#/d' -e '/^$/d' -e 's/LogicOps/Ravish/' name6.txt
```

This removes comments, blank lines, and replaces LogicOps with Ravish in name6.txt.

11. Modifying two files simultaneously:

```
sed 's/Ravish/Aavesh/' /etc/passwd /etc/ssh/ssh_config
```

This replaces the first occurrence of Ravish with Aavesh in both /etc/passwd and /etc/ssh/ssh_config.

```
sed 's#/sbin/nologin#/bin/bash#g' passwd : To replace /sbin/nologin to /bin/bash
```


Linux Command

20) awk : A column-based text processing tool.

- **-F** : Specifies the delimiter (field separator).

Examples:

1. **Filter users with specific UID and login shell:**

```
awk -F: '$3 >= 1000 && $7 == "/bin/bash" {print $1}' /etc/passwd
```

This command will display the usernames of users whose user ID (\$3) is greater than or equal to 1000, and whose login shell is /bin/bash.

2. **Print lines matching a regex pattern:**

```
awk '/root/ {print}' /etc/passwd
```

This prints all lines that contain the word root from /etc/passwd.

3. **Print only the last column:**

```
awk '{print $NF}' data.txt
```

This prints the last column from data.txt (\$NF refers to the last field).

4. **Print lines with more than 3 fields:**

```
awk 'NF > 3 {print}' file.txt
```

This prints all lines from file.txt that have more than 3 fields.

5. **List files and directories in MB and GB:**

```
du -sh * | sort -hr | awk 'BEGIN {IGNORECASE=1} $1 ~ /[GM]/ {print $0}'
```

This command lists only the files and directories whose sizes are in megabytes (M) or gigabytes (G).

21) tar -cvf myarchive.tar files..... ↵ : To create and unpack tar file

-c : To create a new tar file

-v : Display details/changes in standard output

-f : Represent the specific name of tar file

-t : To list the contain of the archive file

-z : To compress / decompress tar file in creation time.

-x : To extract / unpack .tar file

-d : To see the changes of the file

Linux Command

Ex : tar -cvzf mytar.tar.gz file1 file2 file3 ↵

tar -zxvf <Mytar.gz> <Specific file name> ↵ : To extract any specific file from tar file.

Note : It overwrite the updated file when we extract the archive file.

22) star -c -f=myarchive.tar filenames ↵ : To create tar file.

-t : To list the contain of archive file.

-c : To create tar file

-f : Specific tar file name

-x : To extract the tar file

-z : To compress and decompress file without making .gz file

Ex: star -cvz -f=My.tar file1 file2 file3 ↵

23) su - <username> ↵ : To switch user

Ex: su - abhijeet ↵ : abhijeet is the user name

‘-‘ represent all privillage of the user / or login with home directory

su - ↵ : To login as root user

zcat

unzip

24) gzip <filename> ↵ : To compress a file

-d : To decompress file

-l : It provide compression info

Linux Command

Compression Levels:

- -1: Fastest compression (least compressed).
- -9: Slowest compression (most compressed).
- Default is -6 if no level is specified.

Example Commands:

1. Fastest Compression (Level 1)

```
bash
Copy code
gzip -1 filename.txt
```

2. Maximum Compression (Level 9)

```
bash
Copy code
gzip -9 filename.txt
```

25) `gunzip <filename>` ↵ : To decompress a file

26) Hard Links and Soft Links

Creating Links Between Files

To create a hard link or soft link between files, use the following commands:

- **For a hard link:**
`ln <source> <file link name or address>`
- **For a soft link:** Use the `-s` flag:
`ln -s <source> <file link name or address>`

Example:

```
ln -s /1/2/3/filename <file link name or address>
```

Hard Links:

1. A hard link is like giving an existing file a **second name**.
2. It refers to the **same file content** as the original file because it shares the same inode (file information on disk).
3. All hard links are **equal**, meaning there's no main file—each hard link is treated the same as the others.
4. Any **changes** made to one hard link will appear in all other hard links, as they all share the same data.
5. **Deleting** a hard link does **not** delete the file as long as at least one hard link still exists. The file is only removed when **all** hard links are deleted.
6. You **cannot** create hard links for directories and they usually cannot work across different filesystems.

Linux Command

Steps to Find Files with the Same Inode Number and Hard Links:

1. **Get the inode number of the source file:**

```
ls -li <source>
```

- o Note the inode number (it's the first column of the output).

2. **Search for all files with the same inode in the filesystem or a specific directory:**

```
find / -inum <inode_number>
```

- o Replace / with a specific directory path if you want to narrow down the search.

Soft Links (Symbolic Links):

1. A soft link, or symbolic link, is like a **shortcut** to another file or directory.
2. It points to the **file name** or path, not the inode, meaning it acts as a reference to the original file.
3. Soft links can point to **both files and directories**.
4. If the original file is deleted, the soft link becomes a **broken link** and no longer points to a valid file.
5. Soft links can span across **different filesystems**, allowing for linking files in separate locations.
6. Soft links have their own **inode**, which is different from the inode of the original file.

Inode Number:

The inode number holds the metadata (information) about a file or folder in a Linux filesystem, but it does not contain the actual content or the name of the file.

It tracks important details like:

- **File Size:** The size of the file in bytes.
- **Permissions:** The read, write, and execute permissions for the owner, group, and others.
- **Owner and Group:** The user ID (UID) and group ID (GID) that own the file.
- **Timestamps:** Important dates related to the file, such as creation, modification, and last access times.
- **Location of Data on Disk:** Pointers to where the file's actual content is stored on the disk.

27) Changing File and Directory Permissions

To change the permissions of a file or directory, use the following command:

```
chmod <permission> <filename>
```

Example:

```
chmod ugo+rwX <filename>
```

Permission Breakdown:

- **u** = owner (user)
- **g** = group owner
- **o** = other users

Permission Values:

- **read** = **r** = 4
- **write** = **w** = 2
- **execute** = **x** = 1

Special Permissions:

- **setuid** = **u+s** (4) — can be set for files only and is applicable for users; **x** is replaced by **s**.

Linux Command

- **setgid** = $g+s$ (2) — can be set for directories only and is applicable for groups; x is replaced by s .
- **sticky bit** = $o+t$ (1) — can be set for directories only and is applicable for others; x is replaced by t .

Symbols:

- **+** : To add permission
- **-** : To remove permission
- **a** : All (user, group, and other)

Recursive Options:

- **-R** : To change permission recursively (uppercase 'R').
- **-x** : To add execute permission only for directories recursively (uppercase 'X'). it won't work while removing the permission also need to use -R. ex. `chmod -R u+X /root/Data_Disk`

Chattr Command:

To add or remove chattr permission, preventing even the root user from deleting a file:

```
chattr +i <filename>
```

```
chattr -i <filename>
```

Check Chattr Status:

Use the following command to see if chattr is applied:

```
lsattr
```

Additional Examples:

- **Give read, write, and execute permissions to all:**
`chmod a+rwX file1`
- **Set all permissions for all users:**
`chmod 777 <filename>`
- **Set all permissions for all users and setuid:**
`chmod 4777 <filename>`
- **Set all permissions for all users (setuid, setgid, and sticky bit):**
`chmod 7777 <filename>`
- **Set read, write, and execute permissions for the user:**
`chmod u+rwX <filename>`
- **Remove read, write, and execute permissions for the user:**
`chmod u-rwX <filename>`
- **Set chattr:**
`chmod +i <filename>`

28) Changing File and Directory Owner and Group

To change the owner and group of a file or directory, use the following command:

```
chown ownername:groupname <filename>
```

Example:

```
chown root:linuxgroup <File1>
```

Linux Command

Changing Only the Owner:

To change only the owner while keeping the current group, use:

```
chown ownername <filename>
```

Example:

```
chown root <File1>
```

Changing Only the Group:

To change only the group while keeping the current owner, use:

```
chown :groupname <filename>
```

Example:

```
chown :linuxgroup <File1>
```

29) Changing Default File and Directory Permissions

To change the default permissions for newly created files and directories, use the `umask` command:

```
umask <value>
```

Example:

```
umask 022
```

Default Permission Values:

- **File value:** 666
- **Directory value:** 777
- **Normal user umask:** 002
- **Root user umask:** 022

Permission Calculation:

- **File:** $666 - 022 = 644$ (rw-r--r--)
- **Directory:** $777 - 022 = 755$ (drwxr-xr-x)

Default Umask Value:

The default umask value is set in `/etc/bashrc`.

To apply the change after update the configuration file, use:

```
source /etc/bashrc
```

or

```
exec bash
```

30) sudo : Execute a command with superuser privileges.

Basic Usage:

```
sudo <command>
```

Linux Command

Common Use Cases:

- **Run a command as root:**
`sudo <command>`
- **Run a command as a specific user:**
`sudo -u <username> <command>`
- **List the commands a user can run with sudo:**
`sudo -l -u <username>`
- **Edit a file with root privileges:**
`sudo vim <filename>`
- **Switch to root user:**
`sudo su -`

31) stat <filename> ↵ : To show the modification details of file.

32) id <username> ↵ : It show the id of the user.

Ex : `id root ↵` // It will show user id, group id of root user.

33) which <command name> ↵ : Show the full path of shell command.

Ex: `which passwd ↵`

Output : `/usr/bin/passwd`

34) whereis <commandname> ↵ : To locate the binary, source and the location of manual page of the file.

-b : Search only for binary.

-m : search manual page file location.

-s : Search only for source.

Linux Command

35) `whatis <command name>` ↵ : To display manual page description.

36) `locate <filename>` ↵ : To find file by name

`-i` : Ignore case

`/var/lib/mlocate/mlocatedb`

`updatedb` ↵ // To update the mlocate database

37) `updatedb` ↵ : To update database for locate / mlocate.

38) `info <commandname>` ↵ To display the command documents

`-n` : For next page

39) `wc <filename>` ↵ : To count line, word, character of text file.

`-l` : count new line.

`-w` : count word.

`-c` : count Character

40) `shutdown` ↵ : To power off the system

`init 0`

`power off`

`shutdown -P now` (Uppercase P)

`shutdown -P +5`

`shutdown -P 00:00`

`shutdown -c`

`systemctl poweroff -i` // Other user can shutdown the system using this system.

41) `restart` ↵ : To restart the system.

`init 6`

`reboot`

`systemctl reboot`

`shutdown -r 00:00`

`shutdown -r now`

Linux Command

shutdown -r +5
shutdown -c

42) gpasswd <option> <groupname> ↵ : to assign password to the group.

gpasswd <group name> // To assign password to group

-a : add user to the group

gpasswd -a <username> <group name> // To add user to a group

-d : del user from tht group

-r : remove password from the group

gpasswd -r <gname> // remove the password from group

-R: restrict to access that group.

gpasswd -R <group name> // We cant not use that group

A: set the list of sdministrative users

-m: to set the list list of group mamber

gpasswd -d <username> <groupname> // To delete the user from the group

43) halt ↵ : To halt the system.

systemctl halt ↵

halt ↵

shutdown -h +5 ↵

44) useradd <username> ↵ : To add new user.

-M : without create home dir (Uppercase M).

-d : To specify particular home dir of user manually on creation time

-u : Top add user id on creation time

useradd -o -u 0 -g root <username> //Create duplicate root user ##use root as primary group

45) userdel <username> ↵ : To delete user

-f : forcefull

Linux Command

-r : remove home directory and mail spool

46) passwd <username> ¶ : To give password

-l : to lock the user(Lowercase l)

-u : to unlock the user(Lowercase u)

47) groupadd <groupname> ¶ : To add new group

-g : group id

-r : system group

-R : root group

-n : Change group name

48) groupdel <groupname> ¶ : To remove group

-R : To delete group and apply changes to the root dir.

49) w ¶ : To show who is logged on and what they are doing

50) who ¶ : To show who is logged on.

-r : To show run level

-a : To see system boot time

-b : To see system last boot time.

cat /etc/passwd | grep user1 // To see current login shell of particular user.

51) usermod <option> <username> : To modify user account

-c : Modify comment

-md : to modify user home directory

-L : To lock the user (Uppercase L)

-U : To unlock the user (Uppercase U)

Linux Command

-g : To add primary group
-G : To add secondary group
-u : To change uid
-l : To change user name.
-s : To change login shell
-aG : to append secondary group

Ex : usermod -G <groupname> <username> //To add the secondary group
usermod -u <uid> <username> // to change the uid of the username
usermod -l <new user name> <old user name> // To rename the user name
usermod -s </sbin/nologin or /bin/bash> <user name> // To change the user login shell
usermod -G "" <username> //To remove all secondary remove
usermod -aG <Secondary group name> <user name> // To append secondary group means more than one secondary group.
usermod -o -u 0 <username> // Duplicate root user

PS : To check the details of process.

ps -cf :- display process information with PPID

ps -u <u.name> //to view the process of user

Ex:-ps axo pid,comm,nice | grep httpd

ps -G <g.name>/ //to view the process that is running by the group

Ex:-ps -ef |grep init

ps -ely cat // to view the nice value of the command

ps -p <pid> // to view the process name

pidof <p.name> //to view the process id

53. Kill : To terminate a process

- kill <process id> // To kill the process by pid

Linux Command

(2373)

- kill %1 :- It terminate the job 1.
 <job no.>
- kill -9 pid // To kill process immediately
- kill -SIG STOP/19 %1 :- It stop job 1.
- kill -SIG CONT/18 %1 :- It continue job 1.

Note : “kill” terminate the process based on PID where as “pkill” terminate the process based on name or other attributes.

- 1 -SING UP
- 9 -SIG KILL // Kill the process forcefully
- 15 -SIGTERM // Kill the process gracefully
- 18 -SIGCONT
- 19 -SIG STOP

55) ip : To show ip address and show / manipulate routing , device policy routing and tunnels.

- ip addr // It show ip address
- ip addr show ens37 // Show ip address of device ens37.
- ip -s link show ens 37 // show sending and receiving packets.
- ip a s eth0 // Show the pass of eth0

Uptime : to check the system uptime

Uptime -p : it will show runiing duration

56) top : To display linux process and load average

-20 :- High priority

19 :- Low priority

Shift +n or cat/proc/loadavg :- short it high to low of PID

1 :- To check the no of CPU

r :-To renice by PID

k :-To kill process by PID

Linux Command

-n 2 :- To refresh screen 2(n) times and comes into terminal number.

-d 3 :- It set the time for refresh display.

u :- To see the process owned by the user

shift + m or M = it sort the memory utilization top to bottom

shift +p or P = It sort cpu utilization top to bottom

Priority (PR) = Nice (NI) + 20

For example:

- If **NI = 0** (default), the **PR = 20** (default priority).
 - If **NI = -5**, the **PR = 15** (higher priority).
 - If **NI = +10**, the **PR = 30** (lower priority).
-
- **2938**: The **Process ID (PID)**. This is the unique identifier for the process, in this case, 2938.
 - **root**: The **user** that owns the process. In this case, it's `root`, meaning the process is being run by the root user.
 - **20**: The **priority (PR)** of the process. In this case, it's 20, which is the default priority for a regular user process. The lower the priority value, the higher the priority of the process.
 - **0**: The **nice value (NI)**. This value can be adjusted by users to change the priority of a process. A value of 0 means the process is running with the default nice value.
-
- **1541576**: The **virtual memory size (VIRT)** in kilobytes. This is the total amount of memory the process can access, including both the memory in RAM and the swap. In this case, it is 1,541,576 KB (or approximately 1.54 GB).
 - **307096**: The **resident set size (RES)** in kilobytes. This is the actual physical memory used by the process, excluding swap. Here, it's using 307,096 KB (about 307 MB) of physical memory.
 - **73984**: The **shared memory size (SHR)** in kilobytes. This is the amount of memory the process shares with other processes. In this case, it's 73,984 KB (or about 73 MB).
 - **S**: The **state** of the process. In this case, s stands for **Sleeping**, meaning the process is idle, waiting for an event like input/output (I/O).

Linux Command

57) nice : To run/set a program with modified scheduling priority.

-20 :- Most Favouritable

19 :- Least Favouritable

nice -n 0 <process name> // To set priority '0' to the process httpd
httpd

It cannot done if processs is running

kill -9 <process name> //To kill process

58) renice : To change the priority of running process.

Ex. renice -n 10 <pid> :- It change the priority of pid .
Priority

-n <Priority> \$(pgrep httpd) :- It change the priority of process

Ex. renice -n 10 \$(pgrap httpd)

59). time : To display the time taken to complete a process

Ex :- time nice -n 10 tar -cvf mytar.tar // It display the time to create tar file.

du -sh * //estimate file space uses.

Note : du (command) for check file/directory size

60. df : To display the file system and disk space usages.

-T :- To view the particular file system type.

-h :- It show the disk space/in human readable

Linux Command

1) du : To estimate file space usage

-a : To display all file sizes

-h : human readable

du -sh *

2) dd : To create a file

dd if=/dev/zero of=/filename bs=1MB count=2048 // It will create 2GB empty file

61. chage : To change user password expiry information Or to change user password policy.

-l :- show user password information

i.e. chage -l <username>

-E :- To change account expire date

i.e. change -E 2015-06-05 <username> ## yy-mm-dd format

-1 :- To remove account expire

i.e. chage -E -1 <username>

-M :- To modify password expire date .

i.e. chage -M 90 <username> // (Uppercase M)

days format

-m :- minimum days ## Minimum days between password change

-d :- It expire everything

i.e. chage -d 0 <username>

-W :- To modify warning days // (Uppercase W)

-I :-set password inactive after expitation to inactive

Note : date -d "+40 days" +%F ## To find specific date

vim /etc/passwd

Linux Command

<user name>: x:<uid>:<gid>:<comment>: <users home directory>:<login shell>

vim /etc/shadow

Username:password :last changed:min.day:max.days:Warn days : inactive days :expire days:reserved for future.

Inactive days: after few days of passwd expired, password will inactive then user have to contact admin team if user miss this one also then after that account will expire.

Note: Minimum days always should be 0.

Note:-/etc/login.defs // Default password policy configuration file

passwd -x 1 <user name> // It put never expiry of user

passwd -S <uname> // (to see the status of the password of that user. if root user is not assigned
// the password then the password status is locked)

passwd -d <uname> // To delete the password from user.

passwd <uname> // To set password of user

63) fdisk : To manipulate MBR based disk partition / It means create, delete and modify the partition.

-l : To list all details of disk

fdisk -l /dev/sdb

fdisk /dev/sdb : To create, delete and modify partition

Options :- n : new partition, d : delete partition, t : type of partition

Partition id :- 83 : Linux File System, 82 : Swap, 8e : LVM, fd : Raid

Note : fdisk can create only 4 partition "3 Primary partition and 1 extended partition".

gdisk can create 128 partition

fdisk : 32Bit

gdisk : 64Bit

Linux Command

64) `gdisk` : To interactive GUID partition table (GPT) manipulator.

Support UEFI Device

8300 - Linux Filesystem

8200 - Swap Filesystem

8e00 - LVM Filesystem

65) `mkfs` : To build a linux filesystem // By default it will create ext2 file system

`-t` : Specify the type of file system to be build.

`-f` : Forcefully

`mkfs -t xfs <partition name>`

`mkfs.ext2/ext3/ext4/etc <device name>`

ex. `mkfs.ext2 -f /dev/sdb1` : It will change the file system type by formatting the partition with a particular file system type

`mkfs.ext4 -m <no.> <partition no.>` // To format a partition with a specific reserve % for superblock. By default it will take 5-10% superblock.

66) `blkid` : To print block device attributes or to see block id.

67) `mount` : To mount a file system

`-a` : To mount all

For temporary mount

`mount <device name> <mount Point>`

e.g. `mount /dev/sdb1 /mnt/mymount`

Linux Command

For permanent mount

vim /etc/fstab //By make an entry in fstab file

<device name> <mount point> <filesystem type> <mount options> <take a backup or not> <fsck value>

e.g. /dev/sdb1 /mnt/oracle xfs defaults 0 0 ## After make entry run mount -a command or reboot

mountpoint <Directory location> //To check particular directory is mount point or not

vim /etc/mtab ##Is the file that keep record of all recent mount point

mount // Will show all mounted device

68) umount : To unmount the file system.

-a : To unmount all

-f : To unmount forcefully

umount <mountpoint> //To unmount a file

umount /mnt/mymount

69) lsblk : To list block device of disk / device in tree view.

70) partprobe : To update partition table information

-s : Show summary of device and their partition.

e.g. partprobe -s /dev/sdb ## only disk name not partition name

71) pvcreate /dev/sdb1 /dev/sdc1 : To initialize physical volume(s) for use by LVM

Linux Command

72) `pvremove /dev/sdb1 /dev/sdc1` : To rename LVM label(s) from physical volume(s).

`pvremove <device name>`

73) `pvdisplay` or `pvs` or `pvscan` : To display the physical volume.

`-m` : To display mapping of physical extents

74) `vgcreate <vgname> /dev/sdb1 /dev/sdc1` : To create a volume group

`vgcreate <vgname> /dev/sdb1 /dev/sdc1`

Note : Vg is the composite of logical volume and physical volume.

75) `vgdisplay` or `vgs` or `vgscan` : To display volume group

Note : To reduce Lvm partition formatted with XFS filesystem

- `xfsdump -f /location/name.dump <Mount Point>`
- Then unmount, then reduce, then formate with `mkfs`, then mount
- `xfsrestore -f /location/name.dump <mount point>`

76) `vgremove <vgname>` : To remove volume group

77) `lvcreate -n <lvname> -L <size> <vgname>` : To create a logical volume

`-n` : To give lv name

`-L` : To give size

e.g. `lvcreate -n lv1 -L 500M vgname1`

Linux Command

`lvcreate -l 200 -n <lvname> /dev/vg1`

`lvcreate -n <Lv name> -l 100% FREE <Vg Location>`

78) `lvdisplay` or `lvs` or `lvscan` : To display information about a logical volume

79) `lvremove /dev/vg1/lv_vg1` : To remove logical volume

LVM Snapshot

`lvcreate -s -n <Lvm_snapshot> -L <20M> <Lv Location>`

`umount <file system>`

`lvconvert --merge </lvm_location>`

`lvchange -ay <lvname>`

`mount -a`

80) `vgextend <vgname> <device name>` : To add physical volume to a volume group

e.g. `vgextend vg1 /dev/sdd3`

81) `vgreduce <vgname> <device name>` : To remove physical volume from a volume group

i.e. `vgreduce vg1 /dev/sdd3`

83) `lvextend -L <size> <Location of lv>` : To add or extend space to the logical volume.

i.e. `lvextend -L 5G /dev/vg/lv`

`lvextend -L +5G /dev/vg/lv`

`lvextend -l +50% FREE /dev/vg/lv`

Linux Command

84) `lvreduce` (for ext file system)

To reduce the size of logical volume

`lvreduce -L -<size> <Lv name>`

`lvreduce -L -1G /dev/vg1/lv1`

`lvreduce -l 20/dev/vg1/Lv1//To remove using LEs`

`-r =resizefs`

`lvreduce -l-500m-r <lv name> reduce ext file system`

85) `pvmove` <location whose data is store>

To move extents from one physical volume to another

`Pvmove <old pv>`

86) `xfs_growfs` <mount point>

To refresh/Expand on xfs file system

`Xfs_growfs <mount point>`

e.s. `xfs_growfs /mnt/my volume`

`xfs_repair /dev/vg1/lv1//to repair xfs file system`

87) `resize2fs` <mount point>

Ext2/ext3/ext4 file system resize

e.s. `resize2fs /mnt/my volume`

88) `tune2fs`

To change parameters of an ext2/ext3/ext4 file system.

`-l :-To set level name`

e.s. `tune2fs -l ext4label /dev/sdc1`

`-l <device name> :- To verify label name`

Linux Command

89) xfs_admin

To change parameter of an xfs file system

-l :- label name

Xfs_admin -l <new label name> <partition name>

e.s. xfs_admin -l label1 /dev/sdd1

To verify

-l <device name> :- label name of device

90) xfs_repair <device name>

To repair an xfs file system

Xfs_repair /dev/sdc1

1) vgcfgrestore : To restore volume group information

vgcfgrestore --list <Vg Name> //To see Vg backup list

vgcfgrestore -f <paste the backup file name> <vg name>

2) vgcfgbackup : To backup volume group configuration

vgcfgbackup <volume group> //To take a backup

3) vgchange : To change volume group attributes/parameter

-l : Logical volume number

vgchange -l <no> <vg name>

-P : Max physical volume number

vgchange -p <No.> <vg name>

-s : To set physical extents size / block size

Linux Command

`vgchange -s <no.> <vgname>`

-a : Activate (n:no, y:yes)

`vgchange -an <Volume groups> //Activate, No`

`vgchange -ay <Volume groups> //Activate, Yes`

4) `lvchange` : To change the attributes/ parameter of logical volume(s)

-a : y|n

-p : for permission : rw|r

`lvchange -Pr <lv name> //To put the logical volume read only mode`

`lvchange -Prw <lvname> // To put the logical volume read, write mode`

5) `vgrename` : To rename volume group

`vgrename <existing vg> <new Vg name>`

6) `lvrename` : To rename logical volume

`lvrename <existing lv> <new lvname>`

7) `cdrecord` : To write the cd/dvd

`Cdrecord /root/desktop/rhel.iso`

8) `eject` : To eject removable media

Linux Command

9) strings <command name> : to convert binary language into human readable language

string mkfs

91) fsck <device name or partition name>

To change consistency and repair a linux file system.

fsck /dev/sdb1

92 free :- To see the memory and swap space

To display amount of free and used memory in the system.

-m :- display mega byte (mb)

e.g. free -m

-g :-display GB

-h :- display in human readable.

93) mkswap <device or partition name>

To make a swap file system

e.g. mkswap /dev/battlestar/swap

94) swapon/off <swap device partition name>

To enable/disable swap space

Linux Command

-a :-all

i.e. swap on -a //on

swap off -a //off

i.e. swap /dev/battlestar/swap

-s :-show summary of swap

Swap -s //To see the swap space information

You can check the current **swappiness** value with:

cat /proc/sys/vm/swappiness

- A value of **0** tells the kernel to avoid swapping as much as possible, only using swap space when absolutely necessary.
- A value of **100** tells the kernel to aggressively swap pages out of RAM.

95) getfacl <file/dir/name>

Get file access list (ACL)

i.e. getfacl file1

96) setfacl

To set access control lists (ACL)

- -m u:username:permission filename // To modify ACL

i.e. setfacl -m u:abhi:rw file1

- -b <filename> //remove all ACL permission
- i.e. setfacl -b file1
- -x u:abhi file1//rename ACL permission of user.
- --remove-default <file/dirname> //To remove all default ACL permission.
- i.e. setfacl --remove-default file1
- -d : for giving default ACL permission
- -d -m u:abhi:rw dir1
- Setfacl: -x d:v:<username> dirname// To remove default permission
- Setfacl -m m::r file2//To update mask permission

Linux Command

- `Setfacl -m m::- file` //remove all mask permission of the file
- `Setfacl -x m:: filename` // remove perm mask from file //mask perm. will be remove after remove all ACL permission
- `Setfacl -R -m g:filename:rw ; u:abhi:rw dir1`
- `Getfacl filename | setfacl --set-file= - file2` // copy filename1 permission to file
We give default permission only to the dir
The acl mask defines the maximum effective permission for any entry in acl.
CP command does not perserve ACL rules mv command preserve ACL rules

m: max permission

using we assign to some particular user to access the file and directoroes.

97) nmcli : Command line tool for controlling network manager.

`nmcli dev status` // Show device network status

`nmcli dev show` // To show the device

`nmcli con show` // Show all connection

`nmcli con add con-name "Mycon" autoconnect yes type ethernet ifname ens33 ipv4.method auto` //

To add new connection

`nmcli con up <con-name>` // To up the connection

`nmcli con down <con-name>` // To down the connection

`nmcli con del <con-name>` // To delete the connection

`nmcli con mod <con-name>` // To modify the connection

e.g. `nmcli con mod ens33 autoconnect yes` // To update autoconnect as yes of ens33 connection

Note : `/etc/sysconfig/network-scripts/` - Centos7

`/etc/NetworkManager/system-connections/` - Centos9 // device configuration file

Linux Command

Network Teaming

1) teamdctl : Team demon control tool

teamdctl lateam0 state : It will show the status of team "lateam0"

##Process to create teaming

nmcli con add con-name lateam0 type team ifname lateam0 config '{"runner":{"name": "activebackup" or "roundrobin" }}'

nmcli con mode lateam0 ipv4.address 192.168.0.10/24

nmcli con mode lateam0 ipv4.method manual

##Now add the port with lateam0

nmcli con add type team-slave con-name lateam0_port1 ifname ens33 master lateam0

nmcli con add type team-slave con-name lateam0_port2 ifname ens34 master lateam0

nmcli con up lateam0/lateam0_port1/ lateam0_port1

Some other things related o teaming

Assign IP address to team0

nmcli con add con-name "name" type ethernet ifname ens33 autoconnect yes ipv4.address 192.168.1.15/24 ipv4.method manual gw4 <gateway id> ipv4.dns <dns Id> //Assign static ip

nmcli con mod dynamic ip ipv4.ignore-auto-dns yes

nmcli con mod dynamic ip ipv4.dns <DNS Server ip>

PEERDNS=no

+ add dns (we can have up to 3 nameserver configured)

- remove dns, without sign replace the current dns

98) nmtui : Text user interface for controlling network manager

99) nm-connection-editor : It show GUI for managing devices.

100) ss : To display listening port and establish connection. Or utility to investigate sockets.

Linux Command

- a : To display all sockets
- t : To display tcp sockets
- n : To display port number
- l : To display listening socket

101) netstat : To print network connection, routing tables, interface statistics.

netstat -ntulp // To check listening port (tcp and UDP)

netstat -ntp - // To check established connection

foreign : From where service is accessing s

-r : Display kernel routing tables

-n : Show numerical addresses

t = Tcp u = udp, n = port number, l = port is listening or not, p = process id

Network monitoring connection both for incoming & outgoing as well as view routing table interface statistics etc.

netstat : -at : TCP port

netstat : -l : Listing port

netstat : -s : statistic

netstat : -r : routing table

netstat -an | grep 22 : To find all process running on 22

netstat -ap | grep ssh : To find ssh running process

lsof -i :9090 : To check which process/service is using this port

sudo ss -tuln | grep :9090 : Same

systemctl list-sockets | grep 9090 : To see the socket

Linux Command

Purpose of ListenAddress:

- `ListenAddress` defines which network interfaces (or IP addresses) *on the SSH server* (Server A) will accept SSH connections.
- For example, if Server A has multiple IP addresses (e.g., `192.168.100.1` and `10.0.0.1`), setting `ListenAddress 192.168.100.1` would mean SSH will only listen for connections on `192.168.100.1`, and ignore any connection attempts on `10.0.0.1`.

1) ip route : Routing table management

`Ip route list` : list all route

`Ip route add 216.58.217.0/24 via 192.168.1.11 dev ens33` : To add ip route

##static route

`Vim /etc/sysconfig/static-route`

`
`any net 173.194.205.0 netmask 255.255.255.0 gw 162.242.253.1 dev eth0`

2) nslookup : To query internet name servers interactively

`nslookup google.com`

3) nmap : To check the open port

`nmap -p <port nm.> <ip> //` To check particular port for particular ip is open or not

`nmap <ip> //` To see all open port of the ip

Linux Command

4) exportfs : To maintain table of exported NFS file system

-a : Export or unexport all dir

-v : List of exported dir

-r : Re-export all directories

i.e. exportfs -avr

5) dig : DNS lookup utility

dig -x <ip.address> // To query of ip address

dig <FQDN> // To query of domain

101) traceroute : To print the route packets trace to network host

- traceroute <ip address>

102) ping : Send ICMP ECHO_Request to network host.

-c 5 : Show 5 ping and quit

ping <ip address> : To ping ip address

Bash : Is the command language interpreter for GNU OS.

Shell : Is an interactive interface that allows user to execute command and utilities in linux os.

103) hostnamectl : To control the system hostname

hostnamectl set-hostname <host name> // To change or set hostname

hostnamectl status // To see hostname information

Linux Command

hostname // It display the hostname

hostname -i // To show ip

exec bash // To apply the changes without reboot

104) getent : To get entries from name service switch libraries.

105) timedatectl : To control the system time and date.

- list-time zones // To display time zone
- set-time zones 'Asia/kolkata' // To set the Kolkata time zone
- set-time 12:32:15 // Set the time
- timedatectl // It display current date and time
- set-ntp false/true // To enable and disable ntp

106) tzselect : To select a time zone

107) at : To schedule task

- at now +10minutes // Schedule a task after 10 min

at > init 6

ctrl + D - Two time to excute above command ?? It will execute the command after 10 minutes

- at 12:00 am // Schedule the job for the given time

- atq // To list the pending job

- atrm <job No.> // to remove job

vim /etc/at.deny // Here we can restrict user to use at utility by enter username in the file

vim /etc/at.allow // Here we can allow the user to use at utility by default this file not created

-r : To delete the jobs

-l : To list the jobs

Try below

at 5:20 pm

Linux Command

at > logger "The system current uptime is \$(uptime)"

ctrl+D (2times)

journalctl -xn

108) yumdownloader : to download rpm package

yumdownloader <packagename> //It will install the rpm package in local storage

yum localinstall <rpm package location> // Then we can use this step to install above downloaded package

109) yum : yellowdog updater modifier REDHAT package management tool

yum check-update // To check all package updates

yum info <package name> // To gether info of the package

yum install <package name> // To install package

yum list installed // It show all install package

yum update <Packagename> // To update package

yum update -y //It update all

yum remove <packagename> // It remove the package

yum autoremove // To remove unnecessary package or dependency

yum clean all // to clean temporary, cache file.

Yum config-manager --disable/--enable <repos id> // To enable/disable repos

yum localinstall <package name> // To install downloaded package

yum update kernel //To update kernel

yum deplist <packagename> // To check dependency

yum install --downloadonly --downloader=downloads php

##How to roll back

yum history list all // check the history

yum history info <id> //Find the info id and check

yum history undo <id> // Then by entering previous id we can roll back to previous

Linux Command

Note : Yum download the package along with dependency.

110) rpm : Redhat package manager

rpm -ivh <package name>

-i <package name> // To install package

-e <package name> // To erase or remove package

-u <package name> // To upgrade package

-q <package name> // To query about package

-qa <package name> // All query

-ql <package name> // To show all file of the package

-R <package name> // To list package dependent **##NOT Sure**

e.x.

rpm -i httpd

rpmquery httpd

rpm -qa httpd

rpm -e httpd

rpm -qf <filename> // To check which package is responsible for the file

rpm -qa kernel --last // To see kernel installation date time

rpm -ivh --test <package name> // To check the package consistency

Note : rpm not able to download the dependency along with package

110) uname : To print system information

-a : Print all information

-r : Print kernel version

Linux Command

-m : System architecture current bit

-o : O/S

112) firewall-cmd : Firewall command line client

- firewall-cmd --get-zones // It display all available zone
- firewall-cmd --get-default-zone //To show default zones
- firewall-cmd --list-all //To show default zone details
- firewall-cmd --zone=home --add-source=192.168.1.0/24 // To add ip address to the source of the zone
- firewall-cmd --reload or --complete-reload //To reload firewall configuration
- firewall-config // To show GUI based configuration
- firewall-cmd --zone=home --permanent --add-source=192.168.1.0/24 //To apply permanent we have to reload the configuration after apply
- firewall-cmd --zone=public --add-port=80/tcp // To add for runtime after reboot it will not survive
- firewall-cmd --panic-on // It looked everything on local console
- firewall-cmd --query-panic // To see panic mode status or query
- firewall-cmd -state // To see firewall status

To enable the logs.

```
sudo firewall-cmd --set-log-denied=all
```

```
sudo systemctl restart firewalld
```

```
tail -f /var/log/messages
```

```
"Nov 21 20:19:44 masternode kernel: filter_IN_public_REJECT: IN=ens33 OUT=
```

```
MAC=00:0c:29:49:45:56:00:50:56:c0:00:08:08:00 SRC=192.168.100.1 DST=192.168.100.163 LEN=52
```

```
TOS=0x00 PREC=0x00 TTL=128 ID=38997 DF PROTO=TCP SPT=56947 DPT=80 WINDOW=64240
```

```
RES=0x00 SYN URGP=0"
```

Linux Command

113) ssh : To access remote system using ssh

ssh <username@ip address>

ssh [root@192.168.1.12](#) // To access the system "root=username" & "192.168.1.12=Remote system address"

ssh [root@192.168.1.12](#) ls // It execute the ls command and exit

114) scp : To secure copy (Remote file copy program)

scp <source location> <destination location>

scp /root/file1 [root@192.168.1.12:~/](#) // It will copy "file1" to the remote address given file location

114.a) rsync : fast remote and local file copying tool

It use delta technique to transfer the file

-P : resume the transfer from interrupted point

If we want to copy a file that is reside, before the it 1st check or matches the both file size and creation/updation time of the file and then only copy the changes or new content.

rsync <source location> <destination location>

rsync <source file location> [root@192.168.1.12:~/](#) ## Upload

rsync [root@192.168.1.12:~/](#) <destination> ##Downloading

rsync file1 [root@192.168.1.11:/root/Documents](#)

-a = all (Copy the file with all permission except SELinux and ACL permission)

-aA = Synchronize ACL permission

-aAx = Synchronize ACL permission and selinux permission also

115) sftp : To secure file transfer program

sftp [root@192.168.1.12](#)

then

sftp > ls : Display content

sftp > get <filename with loaction> : To download or copy file from remote location to local

sftp > put <Filenme with location> : To upload or send file from local to remote location

bye : To exit

Linux Command

116) ssh-keygen : Authentication key generation.

-t : Key type "rsa algorithm or dsa algorithm"

ssh-keygen : By default It will create rsa type key

ssh-keygen -t dsa : It will create dsa key

/root/.ssh/ // Is the location of pub and private key

id_rsa and id_rsa.pub

Note : public key share with remote machine and compare with private key of local machine if all ok then we can take access

117) ssh-copy-id : To copy the key to authorize login on a remote machine

ssh-copy-id root@192.168.1.11 // It copy the generated key by "ssh-keygen" to the remote IP address

// It will copy the public key "id_rsa.pub" to the remote

118) ssh-add : Add private key identifies to the authenticate agent.

Process to use key based authentication

Step 1 : ssh-keygen or ssh-keygen -t dsa // It will create the key file "id_rsa & id_rsa.pub" in /username/.ssh/ directory.

Linux Command

Step 2 : `ssh-copy-id root@192.168.1.11` // It will copy the key "id_rsa.pub" to the authorized_key to the remote ip address.

Step 3 : `ssh-agent bash`

Step 4 : `ssh-add` // It will add the identity to the id_rsa file

119) `getenforce` : To display current mode of SELinux

120) `setenforce` : To change the mode of SELinux

0 // Permissive mode // Disable mode but it will notify for unwanted access

1 //Enforcing mode ## Means enable

`setenforce 0`

`setenforce 1`

`vim /etc/selinux/config`

`SELINUX=enforcing/disabled` // It will apply the changes for permanent after that need to reboot

To completely disable the SELinux we need to make changes in configuration file as

"SELINUX=disabled " then reboot

Or we can use directly sed command

`sed -i 's/enforcing/disabled'`

2) `semanage` : SELinux policy management tool

`semanage port -l` // List all port

Linux Command

`semanage port -l | grep 80`

`semanage port -a -t httpd_port_t -p tcp 8282 // To add port 8282 in Hpptd_port_t`

`semanage fcontext -l : To list all context`

`semanage fcontext -at httpd_sys_content_t "/mnt/check(/.*)?" // To change the context of a file or dir.`

`semanage fcontext -d "/mnt/file(/.*)?"`

`-d : To restore default file context.`

3) `restorecon` : Restore file default SELinux context.

`restorecon -R : Change selinux file context label recursively.`

`restorecon -V : Show changes.`

e.x. `restorecon -Rv /mnt/check`

3a) `chcon` : To change file context "SELinux security context"

`chcon -R -t samba_share_t /file location`

`-R : Recursive`

`-t : type`

4) `getsebool` : get selinux Boolean value

`-a : show all SELinux Booleans value`

`getsebool -a | grep nfs_export`

5) `setsebool` : set SELinux Boolean value

`setsebool -P nfs_export_all_ro=1`

`-P : To set Boolean value for permanent`

`1=enable 0=disable`

Linux Command

121) sysctl : To configure kernel parameter at runtime

-a : Display all parameter value currently available

-w kernel.sysrq=32 :- change parameter value

```
sysctl -a | grep kernel
```

```
cat /proc/sys/kernel/sysrq
```

1) grub2-set-default <option> : To switch between the kernel

0 : Means newly updated

1: Means previous one

grub2-set-default 0/1 // To switch between the kernel reboot is require

package-cleanup --oldkernels --count=1 // Remove old kernel

```
vim /boot/grub/grub.conf // To load default kernel
```

122) elinks : elinks is a text mode www browser.

```
elinks https://localhost
```

configuration file of httpd.

```
/etc/httpd/conf/httpd.conf
```

```
/var/www/html
```

123) systemctl : Control the systemd system and service manager

Linux Command

`systemctl status <service name> // To check status`

`systemctl start/stop <service name> // to start and stop the service`

`systemctl enable/disable <service name> // To enable and disable the service`

`systemctl restart <service name> // To restart the service.`

`systemctl is-enabled / is-active httpd // to check httpd service is enable or not / active or not`

`systemctl list-unit-files // to display enable and disable service`

`systemctl isolate <target.name> // To change target for runtime`

`systemctl list-units --type=target // To display all available target`

`systemctl get-default // to show default target`

`systemctl set-default <target name> // to set the default target`

`systemctl list-dependencies <service name> // To list dependencies of particular services.`

if we enable any service it will create a symlink between `'/usr/lib/systemd/system/httpd.service'` to `'/etc/systemd/system/multi-user.target.wants/httpd.service'`

124) mdadm : Multiple Disk and device administrator

It is used to manage and monitor software RAID devices.

- `mdadm -Cv /dev/md0 -n 2 /dev/sdb /dev/sdc -l 0 //To create raid 0 using /dev/sd[b-c] "Uppercase C" .`
- `mdadm -E /dev/sdb1 /dev/sdb2 //Enquiry about the device`
- `mdadm -D /dev/md0 //Details about the raid device`
- `mdadm /dev/md0 -a /dev/sdb1 //To add disk`
- `mdadm --stop /dev/md0 //To stop the /dev/md0 raid device`
- `mdadm /dev/md0 -r /dev/sdb1 //To remove disk`
- `mdadm --assemble /dev/md0 /dev/sdb /dev/sdc //To restart the /dev/md0 raid device`
- `mdadm --zero-superblock /dev/sdb1 //To remove md superblock from the partition or any specific Raid.`

Linux Command

- mdadm -S /dev/md0 //To stop Raid0

125) smbpasswd : Change users smb password

smbpasswd -a <username> //Add samba password to user

smbpasswd -u <uname> //change password

-d : Disable samba user

-e : Enable samba user

-x : Delete samba user password

-r : remove samba user password

-n : null password

126) cpio : Copy files to and from archives

o : To take backup

i : To restore the cpio backup

v : verbose (Display the output)

t : To list the cpio content

We use cpio to take backup as a archive.

*.cpio is the extensions of cpio archive

e.x.

ls | cpio -o > /root/backup.cpio //To take backup

cpio i < /root/backup.cpio //It restore the backup at the current PWD

cpio t < /root/backup.cpio // To list the content of the cpio archive

127) modprobe : Add and remove modules from the linux kernel

modprobe <module name> // To install and re-install module name

modprobe -r <module name> // To remove module

Linux Command

`modinfo <mod name> // To see the specified module information`

`lsmod // To list the modules`

`extension .ko`

128) `sar` : To check system activity report ---- We can check current and previous dated system activity report.

`sar 2 10 // system report for every 2 sec upto 10 times`

`-b` : Report input output statistics

`sar -b 2 5`

`-s` : Report swap statistics

`sar -s -1 //To check swap information of previous date.`

`sar -s 2 3 //To check swap information of current time`

`-p` : Report cpu usages

`-q` : Load average

`-r` : Memory usages

`-d` : Disk static

`sar -q -f /var/log/sa/sa17 // To see the back dated log file of load average`

configuration : `Vim /etc/sysconfig/sysstat`

`"HISTORY=28"`

log : `/var/log/sa/sa....`

129) `vmstat` : It gives complete information about virtual memory like no of process, memory usage, paging memory, disk.

`vmstat -d // To see the statistics of disk used.`

`vmstat -a // Active inactive process`

Linux Command

- fields

procs : r : Running process

b : Busy Process

memory : swap : How much virtual memory used

free : How much memory is free

buff : How much temporary memory is using

cache : How much caching still using

swap : si : How much data transfer from RAM to SWAP

so : How much data transfer from SWAP to RAM

IO : bi : How much block input

bo : How much block output

130) sosreport : To store the system configuration and diagnostic report.

system : in : The number of interrupts per second.

cs : the number of context switches per second

cpu : us : The cpu time spent in user space. i.e. Normal process

sy : The cpu time spent in kernel space

id : CPU idle time

wa : CPU time spent waiting for I/O

131) iotop : To monitor the I/O statistic

-o : High consumption I/O

-p : PID

-u : user

132) iostat : It reports cpu statistics and I/O statistics for device and partitions.

-c : Display CPU utilization report : iostat -c

-d : Display disk utilization report : iostat -d

-m : display disk utilization in MB format. : iostat -m

-p : Display static of disk in specified format : iostat -p or iostat -p <device or partition>

Linux Command

-N : Display Lvm static : iostat -N
-t : Display time stamp : iostat -t
-X : Display extended information of the disk : iostat -x

133) dmidecode : It gives hardware related information of the sys.

dmidecode -t memory // To see memory info
dmidecode -t bios // BIOS's info
dmidecode -t system // system information
dmidecode -t processor // Processor info

134) crontab -e // To edit the cron jobs

```
*/2 * * * * touch file {1..2}
```

root user it will create file every two minutes.

<minutes>	<hours>	<day of the month>	<month of the year>	<day of the week>
(0 - 59)	(0 - 23)	(1 - 31)	(1 - 12 or jan, feb, ...)	(0 - 6 or sun, mon, ...)

// five star representing this time respectively.

*	Is treated as a wild card. Meaning any possible value.
*/ 5	Is treated as every 5 minutes, hours, days or months. Replacing the 5 with any numerical value will change this option.
2, 4, 6	Treated as an OR, so if placed in the hours, this could mean at 2, 4 or 6 o'clock
9-17	Treats for any value between 9 and 17. So if placed in day of the month this would be days 9 through 17 or if put in hours, it would be between 9 AM and 5 PM.

crontab -r // To remove cron jobs
crontab -l // To list cron jobs
crontab -u <user name> -e or -r or -l

How to reboot the system using crontab

Linux Command

`crontab -e`

`43 12 * * * /sbin/shutdown/ -r // It will reboot the system every day at 12:43`

`vim /etc/crontab ## configuration file`

- The cronjob `0 0 * * * echo 3 > /proc/sys/vm/drop_caches` will execute the command `echo 3 > /proc/sys/vm/drop_caches` at **midnight (00:00) every day**.

In Context of a Cronjob:

- This cronjob is scheduled to run every day at midnight.
- It is used to clear the system's cache daily, which might be intended to free up memory in systems where memory usage by the cache needs periodic flushing (e.g., in environments with limited memory).

...

```
cat <<EOF | sudo tee /etc/cron.d/hourly > /dev/null
```

```
# Run the hourly jobs
```

```
SHELL=/bin/bash
```

```
PATH=/sbin:/bin:/usr/sbin:/usr/bin
```

```
MAILTO=root
```

```
01 * * * * root run-parts /etc/cron.hourly
```

```
EOF
```

...

The cron job defined in `/etc/cron.d/hourly` will run all the scripts located in the `/etc/cron.hourly` directory at **1 minute past every hour**.

What this does:

- **root**: The command is executed by the **root** user.
- **run-parts**: This is a utility that runs all executable scripts in a specified directory (in this case, `/etc/cron.hourly`).
- **/etc/cron.hourly**: This directory contains scripts that are executed on an hourly basis.
- **Don't use** `.sh` at the end of script.

135) `dump` : To take backup

`-o` : full backup

`-(1-9)` : Incremental backup

`dump -ouvf /opt/full.dump //To take backup`

Linux Command

dump -ouvf /destination
restore -tf /destination : To list the backup contain
restore -rf /destination ; To restore the file

u : update
v : verbose
f : used to specify the dump or backup files
t : list the contains
r : to restore

-u : update the entries in /etc/dumpdates conf file.

restore -rf /opt/full.dump // to restores

Some Extra Commands

growpart /dev/sdb 2 // to apply the changes in the partition size.
lldpctl // To find the VLAN of the interface (yum install lldpd -y)
ifdown eth1 // to down the interface
ifup eht1 // To up the interface
ifenslave -c bond0 p6p1 // To change the active interface.

smartctl // To check the hard disk health
smartctl -i /dev/sdb // to check the info about the disk
smartctl -H /dev/sdb // to check the disk health

pam_tally2 -u <uname> // To check failed login
pam_tally2 -r <uname> // To reset user

/etc/pam.d/password-auth // pam_tally2 configuration files.

SUDO

Visudo /etc/sudoers
Go to 98 line

Username ALL=ALL NOPASSWD: ALL // It won't ask root password to user

:wq!

Linux Command

Allows people in the group wheel to run all commands
%wheel ALL=(ALL) ALL

It means all the users that belong to the wheel group can execute all the commands like root

raju ALL=NOPASSWD:/usr/sbin/useradd, /usr/sbin/usermod

abhijeet ALL=(ALL) NOPASSWD: ALL

RAID

Raid stands for redundant array of independent or inexpensive disks, is a storage device that provides fault tolerance, load balancing using stripping, mirroring and parity concepts.

There are two types of Raid available

1) Hardware Raid 2) Software Raid

There are mainly five types of Raid which are generally used

1) Raid-0 (Stripping)

- In Raid-0 data is divided into multiple blocks and each of those blocks is stored on different disk. So, minimum 2 disk is required and partition id is "fd".
- It provide high read/write speed, but there is no redundancy and fault tolerance because if one of the disk fails, the entire data is lost.
- We can use 100% disk space of the total disk space.

2) Raid-1 (Mirroring)

- In Raid-1 the same data is stored in two disk, as like take the backup of the data into another disks. Minimum 2 disks are required and partition id is "fd"
- Write speed is slow as compared to Raid-0.
- It provide fault tolerance and more redundancy.
- But we can use only 50% space of the total disk space.

Linux Command

3) Raid-5 (Striping with parity)

- It striped the data across the multiple disks and on one disks a parity checksum of all the block data is written.
- Minimum 3 disks is required to configure raid-5 and partition id is "fd".
- It provide high read speed but write speed is little bit slow because of parity checksum.
- If one of the disk fail we can recover the data from parity but if 2 disk fails, then we cannot recover the data. So, there is less fault tolerance.
- we can use approximately 70% space of the total disk space and remaining space is reserved for the parity.

4) Riad-6 -----(Striping with double parity)

- It striped the data across the multiple disks with dual parity.
- Minimum 4 disks are required and partition id is "fd" but we can use only 50% space of the total space because of dual parity.
- It provide more load balancing and fault tolerance as compare to Raid-5.
- R/W speed is also fast.
- if two disks are fail on the same time, then we can also recover the data but if 3 disks are fails at the same time then we cannot recover the data.

5) Raid-10 ----- (Mirroring + Stripping)

- Raid-10 is the combination of Raid-1 & Raid-0, To provide data mirroring with data stripping to protect our data.
- It requires minimum of four disks.
- We can use only 50% space of total space.
- Read Write speed is very fast.
- It provides fault tolerance and load balancing.

If one disk failed then we can recover the data but if two disk failed then we cannot recover the data.

Linux Command

Network Teaming

1st we must have more than one NIC card for network teaming.

```
nmcli con add con-name team0 ifname team type team config '{"runner": {"name": "roundrobin" or "active backup" or "broadcasting"}}'
```

Add two NIC card

- nmcli con add con-name port1 ifname ens33 type team-slave master team
- nmcli con add con-name port2 ifname ens34 type team-slave master team

Add ip address

- nmcli con mod team ipv4.address <ip address/netmask> ipv4.method static

```
nmcli con up team
```

```
teamdctl team state
```

```
ping -I team0 <Ip address>
```

Linux Command

FTP (File Transfer Protocol)

Ftp is used for transfer file between one host to another host in a network.

Server :

```
yum install vsftpd * -y
systemctl start vsftpd
firewall-cmd --permanent --add-service=ftp
firewall-cmd --complete-reload

vim /etc/vsftpd/vsftpd.conf
    anonymous_enable = no
    chroot_local_user = YES
    allow_writeable_chroot = YES
    writeable_enable = YES
    anon_upload_enable = YES // Do not use for secure ftp
    ftp_banner = "welcome to ftp"
    max_clients=<no>
```

:wq!

```
systemctl restart vsftpd
```

```
mkdir /etc/ftp/pub/files
chcon -t public_content_t "file(/.*)?"
setsebool -P ftpd_full_access=1
```

```
chmod 770 files
```

Client :

```
yum install -y ftp* lftp* filezilla(optional)
ftp <ip> or <hostname>
ftp > ls
    > get <filename> // downloading
    > put <file name> // uploading
    > mget f1 f2 // Multiple file get
    > buy
```

```
vim /etc/host.allow
vim /etc/host.deny
vim /etc/vsftpd/ftpusers
```

Linux Command

`vim /etc/vsftpd/user_list`

What is DHCP :

Dynamic Host configuration protocol assigns ip address to hosts. DHCP operations fall into four phases : (1) Discovery (2) Offer (3) Request (4) acknowledgement.

How does DHCP client find dhcp server?

The dhcp client discovers a DHCP server by broadcasting a discovery message to the limited broadcasting address on the local subnet. And if a router is present and configured to behave as Bootp relay agent, the request is passed to their DHCP servers on different subnets.

We have to use static NIC card parameter to configure dhcp server. and also close the vmware local dhcp server for dynamic NAT.

Broadcasting : In computer networking broadcasting refers to transmitting packet that will be received by every device on the network.

For ex. : A radio station broadcasting a signal to many listeners, and digital t.v. subscribers receive signal that is broadcast by their T.V provider.

Firewall

- It allows certain packets which we want from authorized sources or genuine applications and it blocks the packets or communication from unknown sources. In that manner it tries to protect our system from unauthorized access of our services and resources.

- The service which controls the firewall is known as firewalld.
- Firewall is a set of scripts and daemons that manages our netfilter.

Netfilter : Netfilter (kernel) is a framework provided by the Linux kernel that allows various networking-related operations to be implemented in the form of customized handlers.

Linux Command

Zones : Is a set of pre-define rule that establish the security borders of a network. Whatever interface of the server in that zone those rule are apply for that interface.

/usr/lib/firewalld/services : Is the firewall services configuration file.

Rich Rule : Allow the administrator to specify custom rule for the zone.

firewall-cmd --get-active-zones // To show active zones

firewall-cmd --zone=home --list-all // To know about the rule of the zone.

firewall-cmd --zone=home --change-interface ens37 // To change the zone

firewall-cmd --zone=home --add-source=ip.addr // It allow every traffic from the host

firewall-cmd --get-zone-of-interface=ens33 // It display the specified device zone.

NAT (Network Address translation) : Natting means that I am hiding my private id and if any request is going on to the network external sites it will be shown public id.

firewall-cmd --version // Check version of firewall

firewall-cmd --get-zone-of-interface=ens33 //To check the zone of interface

Masquerade : Masquerading is assuming we have two network cards, moving traffic from one card to another card and in the process connecting two network. The other assumption is that we have an internal network with private ip address that we will change to an external address on the machine.

firewall-cmd --zone=<zone name> --add-masquerade //Add masquerade

firewall-cmd --zone=<zone name> --query-masquerade // Check masquerade status on particular zone

Masquerade should be on before port forwarding

firewall-cmd --permanent --zone=public \ "press enter"

--add-forward-port=port=22:proto=tcp:toport=24 //It redirect the incoming port 22 into the port 24

firewall-cmd --permanent --zone=public \ "press enter"

--add-forward-port=port=22:proto=ssh:toaddr=<redirect ip addr> //It redirect the incoming port 22 into the port 24

IP Addr + Port No. = Socket

Ip is an unique address which is provide for our machine.

Linux Command

port : port number is a unique id of a protocol in a computer network, all the protocol in a computer network are having a unique port number.

when our system is not connected with the internet still we have an ip which is 127.0.0.1, is nothing but our local host.

By using port host machine connect to the require protocol / services.

A port number is always associated with an IP address of a host and the protocol type of the communication. It completes the destination or origination network address of a message. specific port number are commonly reserved to identify specific services, so that an arriving packet can be easily forwarded to a running application.

Kerberos (KDC)

KDC : A key distribution centre is responsible for providing keys to users in a network that shares sensitive or private data.

- Each time a connection is established between two computers in a network, they both request to KDC to generate a unique password which can be used by the and system users for verification.
- The KDC system should be dedicated machine. The machine needs to be secure if possible, it should not run any services other than the KDC.

Confluence (Inventory Management) : Store the server's information like ipaddress, name, console ip.

Jump Server : It used in linux as a gateway to access other Linux machines on a private network.

Server name Console Port Expansion name

- 1) DELL : DRAC or iDRAC : Dell Remote Access Control Integrated remote Access Controllers.
- 2) HP : ILO : Integrated Light Out (BL460)
- 3) IBM : HMC : Hardware Management Port
- 4) CISCO : CIMC

Why we need SSL Certificate or https

- To encrypts the traffic between web server and user web browsers and enhance data security.

SSL : Secure Socket Layer

CSR : Certificate Sign Request

CA : Certificate Authority

CRT : Certificate Extension Name ".crt"

key : Key of the certificate "/etc/pki/tls/private/<domain name>.key"

TLS : Transport Layer Security

Linux Command

Virtual Host : Means using one Ip address to host, multiple websites or multiple websites hosting on same machine.

Web Server : The job of web server is accept request from client and send responses to those request. web server get a URL, translate it to a file name (For static requests) & send that file back over the internet from local hdd, or is translate it to a programe name (For dynamic request). ex: Ticket booking website changing is always going on, to connect the database and send the ouyput back to the request client.

In case webserver is not able to process requests or website is not available it will send error message.

- It controlled by the httpd daemon.
- Main configuration file `/etc/httpd/conf/httpd.conf` `http:80` `https:443`

Quota

There are two types of quota : 1. Block Quota (Disk Quota) 2. Inode Quota (File Quota)

`/dev/<disk name>` `//` Create a partition

`mkfs.file system <device>` `//` Give file system

`vim /etc/fstab` `//` For permanent mount

`<Device name>` `<Mount Point>` `<FileSystem>` `defaults,usrquota/grpquota 0 0` `//` XFS doesn't support quota

`mount -o remount,usrquota /mnt/quota/` `//` Remount

`usrquota` : for user

`grpquota` : for group

`quotacheck -cu <mountPoint>` `//` To create quota database for user

`quotaon <mount point>`

`edquota -u <username>`

`repquota -a` `//` To display report of the quota.

`edquota -t` `//` To change the grace period.

Linux Command

Boot Process

There are 6 high level stages of a typical linux boot process.

BIOS : Basic I/O system executes MBR.

MBR : Master Boot record executes GRUB.

GRUB : Grand unified Bootloader Executes kernel.

Kernel : Kernel executes /sbin/init

init : Init executes run level programs

Run Level : Run level programs are executed from /etc/rc.d.rc*.d/

1. BIOS : Stand for Basic Input/output system.

- Performs some system integrity checks.
- Searches, loads, and executes the boot loader program.
- It looks for boot loader in hard drive, floppy, or cd-rom
- Once the boot loader program is detected and loaded into the memory, BIOS gives the control to it.
- So, in simple terms BIOS loads and executes the MBR boot loader.

2. MBR : Master Boot Record

- It is located in the 1st sector of the bootable disk. Typical /dev/had, or /dev/sda
- MBR is less than 512 bytes in size. This has three components
 - a. Primary boot loader info in 1st 446 bytes.
 - b. Partition table info in next 64 bytes.
 - c. MBR validation check in last 2 bytes.
- It contains information about GRUB.
- So in simple terms MBR loads and executes the GRUB boot loader.

3. GRUB : GRUB stands for grand unified bootloader.

- If we have multiple kernel images installed on our system, then we can choose which one to be executed.
- GRUB displays a splash screen, and waits for a few seconds, if we don't enter anything. If we don't enter anything it loads the default kernel image as specified in the grub configuration file.

Linux Command

- GRUB configuration file is /boot/grub2/grub.conf .
- So, in simple terms GRUB just loads and executes kernel and initrd images.

4. Kernel : Mounts the root file systems as specified in the "root=" in grub.conf

- Kernel executes the system program since the system is the 1st program to be executed by linux kernel, it has process id(PID) of 1. Do a "ps -ef | grep init" and check the PID.

- initrd stands for initrd Ram Disk. Initrd is used by kernel as a temporary root file system until kernel is booted and the root file system is mounted. It also contains necessary drivers compiled inside, which helps it to access the hard drive partitions, and other hardware.

5. systemd : Looks at the /etc/systemd/system/default.target file to decide the linux run level.

- following are the available run levels.
- 0 : halt / Info/shutdown.
- 1 : Single user mode
- 2 : multi user without NFS.
- 3 : Full multiuser mode.
- 4 : Unused
- 5 : GUI
- 6 : reboot

systemd identifies the default init level from /etc/inittab and uses that to load all appropriate programs.

who -r : To check the default run level.

6. run level programs : When the linux system is booting up, we might see various services getting started. For example it might say starting to send mail..... ok. Those are the runlevel programs executed from the run level directory as defined by our run level.

- depending on our default init level setting. the system will execute the programs from one of the following directories.

- Run level 0 - /etc/rc.d/rc0.d/
- Run level 1 - /etc/rc.d/rc1.d/
- Run level 2 - /etc/rc.d/rc2.d/
- Run level 3 - /etc/rc.d/rc3.d/
-
- Run level 6 - /etc/rc.d/rc6.d/

- There are also symbolic links available for these directory under /etc directory. So, /etc/rc0.d is linked to /etc/rc.d/rc0.d

- Under the /etc/rc.d/rc*.d/ directories we would see programs that start with S and K.

- Programs starts with S are used during startup. S for startup.

- Programs starts with k are used during shutdown. K for kill

- There are numbers right next to S and K in the program names. Those are the sequence number in which the programs should be started to killed.

- For ex. S12 syslog is to start the syslog daemon, which has the sequence number of 12.

Linux Command

- S80 sendmail is to start the sendmail daemon, which has the sequence number of 80. So, syslog programs will be started before send mail.

Autofs

```
yum install -y autofs
```

```
vim /etc/auto.master  
/root/mnt/samba /etc/auto.misc --timeou=300
```

```
vim /etc/auto.misc  
user_mount -fstype=cifs,multiuser,sec=ntlmssp,cred=/config/smb.txt :/192.168.86.131/S_Share
```

Logrotate

In IT log rotation is an automated process used in system administration in which log files are compressed, moved, renamed or deleted once they are too old or too big.

Conf File : /etc/logrotate.conf include /etc/logrotate.d

```
vim /etc/logrotate.d/samba
```

```
    /var/log/samba {  
        daily  
        create 0644 root root  
        maxsize 1M  
        rotate 2  
    }
```

```
/var/log/messages {  
    weekly  
    rotate 4  
    dateext  
    delaycompress  
    compress  
    missingok  
    notifempty  
    postrotate
```

Linux Command

```
/bin/systemctl reload rsyslog > /dev/null 2>&1 || true
endscript
}
```

```
/var/log/sshd.log {
    daily
    rotate 2
    dateext
    compress
    missingok
    copytruncate
}
```

For forcefully rotate

```
logrotate -f /etc/logrotate.d/samba
```

Custom log rotate file

```
vim custom
/root/test/file {           // Here we specified the file location
    daily/monthly/weakly    // This is the rotate duration
    rotate 2                // 2 is the rotate number means how many file it makes
    create                  // Create the file
}
```

```
logrotate -f /root/test/desktop/custom // This is the custom file location
```

Local Repo

Mount the CD or File.

```
mount /dev/sr0 /root/mnt/repo
```

```
cd /etc/yum.repos.d/ : move all repo file into another file
```

then, vim test.repo

```
[test.repo]
name = centos local repo
baseurl = file:///root/mnt/repo
enabled=1/0
gpgcheck=0/1
:wq!
```

Linux Command

Then try,

```
yum remove httpd  
yum install httpd
```

Reset/Change root password

- 1) Reboot
- 2) Press 'e' to edit grub configuration in boot menu.
- 3) rd.break then ctrl+x to continue
- 4) mount -o remount,rw /sysroot // It gives read write permission to the file sysroot.
- 5) chroot /sysroot // Change root into sysroot
- 6) then type passwd // command
- 7) touch /.autorelabel // The file carry autoreload when we reboot.
- 8) exit
- 9) exit

How to scan a harddisk

```
echo "--" > /sys/class/scsi_host/host0 or host1 or host2/scan
```

or

```
echo "--" | sudo tee /sys/class/scsi_host/host*/scan
```

To scan new harddisk where hostname may be different into different system.

-- indicates " = channel" " = scsi target id" " = lun"

How to flash or clear Ram

Linux Command

echo 1> /proc/sys/vm/drop_caches //To clear page cache only

echo 2> /proc/sys/vm/drop_caches // To clear dentries and inodes

echo 3> /proc/sys/vm/drop_caches //To clear cache, dentries & inode ##This is not recommend to run

SELinux

Security Enhance Linux (SELinux) is a security that enhance security that allows users and administrator more control over which applications and users can access which resources such as file.

If httpd or apache is compromised then attacker could access of file and folder qwned that have rw, permission for apache, that means anythings in web serving traffic.

If the SELinux on then the risk is limited to the compromised directories, files, services.

1. Document Root : It is a location of file which are accessible by clients. By default all request taken from this directory “/var/www/html”.
2. ServerRoot : It simplfy where is web server are install. /etc/httpd
3. CGI (Common Gateway Interface) : CGI is an interface between web client and web server.
4. Scripts Alias : The control which directory control script server /var/www/cgi_bin
5. TLS : Transport layer Security is a upper Layer protocol (ULP) that runs over TCP.
6. Http : Http/Https are used to transfer hyper text documents on the www or http is the protocol used to transfer data over the web.
7. Httpd : Is a software program that runs in the background of a web server and wait for incoming server requests. The daemon answers the request automatically.
8. DNS (Domain name system) : DNS is a larger database which resides on various computers that contains the names and ip address of various hosts/domain.
9. NFS : Network file system is a way of mounting linux directories over a network. The main use of NFS is to share out data on a central server to all the PCS In the network.
10. SELinux : Security enhanced linux is a set of security policies which are going to apply on the machine to improve the overall security of the machine.

Linux Command

11. Bin(Binary) : /bin contains the executable (i.e. ready to run) programs that must be available in order to attain minimal functionality for the purpose of booting (i.e. starting) and repairing a system.
12. Repository (repo) : Is a storage location or contain the configuration of repos (i.e. disk location / web location) of disk or web from which our system retrieves and install OS updates and applications.
Each repository is a collection of software hosted on remote server and intended to be used for installing and updating software packages on linux.
13. SELinux Context : Is the mechanism used by SELinux to classify resources, such as files, on a SELinux-enabled system.
14. Boolean : Is a conditional rule that allows run time modification of the security policy without having to load a new policy.
15. Port : A port is a logical entity which acts as a endpoints of communication to identify a given process or application.
(It is a 16 bit number 0-65535)
16. gpgcheck : gpgcheck stands for signature verification from the central database. If signature verification is successful then we are sure about the security if we set the value of gpgcheck is ! then it asks for signature verification else it does not.
17. / - The root directory : Everything on our Linux system is located under the / directory, known as root directory.
18. /bin – Essential user binaries : The /bin directory contains the essential user binaries (Programs) that must be present when the system is mounted in single-user mode.
19. /boot – static boot files ; The /boot directory contains the files needed to boot the system. like GRUB boot loader's files and our linux kernels are stored here.
20. /cdrom – Historical mount point for CD-Roms : It is temporary location for CD-Roms inserted in the system.
21. /dev – device file ; Linux exposes devices as files, and the /dev dir contains number of specified files that represent device.
22. /etc - configuration file : the /etc directory contains configuration files, which can generally be edited by hand in a text editor.
23. /home - Home folders : The /home dir contains a home folder for each user.
24. /lib- Essential shared libraries : The / lib dir contains libraries needed by the essential binaries in the /bin and /sbin folder.
25. /lost+found - Recovery files
26. /media - Removable media
27. /mnt - Temporary mount point

Linux Command

- 28. /opt - Optional packages
- 29. /proc - kernel & process
- 30. /root - root home dir
- 31. /run - Applicatio state files
- 32. /sbin - System administration Binaries
- 33. /selinux - SELinux virtual file system
- 34. /srv - Service data
- 35. /tmp - Temporary files
- 36. /usr - Use binaries & Read-only data
- 37. /var - Variable data file
- 38. demon : Gives service to the process.
- 39. Kernel : Establish relationship between hardware and software.
- 40. dmidecode : desktop management interface table decoder “Retrives system hardware related information such as processor, RAM, BIOS details, Memory.”
- 41. /root : It is root user home directory
- 42. /etc : It contains all system configuration files for linux.
- 43. /opt : This file system holds additional software installed on the system.
- 44. /bin : It contains command used by all users.
- 45. /sbin : It contains commands used by root user.
- 46. /var : contains data that frequently changes while the system is operational.
- 47. /proc : It contain process file.

server.hp.com
<hostname> <Domain name>
<Fully Qualified Domain Name>

Logical Volume management (LVM)

LVM means the combination of 2 or more physical disk in order to make a big logical disk.

- S1. Create / Delete a hard disk.

Linux Command

S2. Make partition using fdisk and gdisk in the partition time give the partition label linux LVM '8e00' in gdisk '8e' in fdisk.

S3. Create Physical volumes

```
## pvcreate /dev/sdb1 /dev/sdc1
```

```
## pvdisplay ; To display physical group
```

S4. Create Volume group

```
## vgcreate <vg name> /dev/sdb1 /dev/sdc
```

```
## vgdisplay : To display the volume group
```

S5. Create logical volume

```
## lvcreate -n <lv name> -L <size> <vgname>
```

```
## lvdisplay : To display logical volume
```

S6. Then make file system of the LV.

```
## mkfs -t xfs <lv name>
```

S7. Then mount the device to any directory

```
## mount /dev/vg/lv /mnt/mymount
```

For remove

1st unmount the file system, then lvremove then vgrename and then pvremove.

How many types of files.

-- device file, symbolic file, regular file, network file, binary file, executable file.

Port Numbers

1. 21 - FTP
2. 22 - SSH
3. 23 - TELNET
4. 25 - SMTP (Send mail)
5. 53 - Domain (DNS)
6. 80 - HTTP
7. 443 - HTTPS
8. 110 - POP3
9. 143 - IMAP

Linux Command

10. 631 - PRINTER
11. 2049 - NFS Server
12. 1028 - NFS Client
13. 3260 - ISCSI_lqn
14. DHCP server - 67
15. DHCP Client – 68
16. Backup port no. - 13782
17. NTP – 123

18. 139 - Samba Port Number : To establish connection
19. 137 - Samba Port Number : To verify share name
20. 138 - Samba Port Number : To data transfer
21. 445 - Samba Port Number : For Authentication

Linux Log Management

1. Logs Captured by Journald

When **journald** is running, it captures logs from:

- **Kernel logs:** Logs from the Linux kernel (`/dev/kmsg`).
- **Systemd services:** Logs from units managed by `systemd` (e.g., system services).
- **Applications:** Any application configured to send logs to `stdout`, `stderr`, or `/dev/log`.
- **Boot logs:** Logs generated during the boot process.
- **Audit logs:** Journald can capture SELinux and other audit-related logs.

Use Case:

- **System troubleshooting:** Quickly analyze logs from services and applications using `journalctl`.
 - **Centralized systemd logging:** Simplifies log management in modern Linux systems.
-

Linux Command

2. Logs Captured by Rsyslog

When `rsyslog` is running, it captures logs from:

- **Kernel logs:** If explicitly configured to read from `/proc/kmsg`.
- **Applications:** Apps configured to log via `/dev/log` or other syslog-compatible mechanisms.
- **Custom sources:** Rsyslog can also capture logs from files, network sources, or custom inputs.
- **Forwarded logs:** Logs forwarded by other servers or `journald` (if `ForwardToSyslog=yes` is set).

Use Case:

- **Remote logging:** Forward logs to a central server for analysis.
- **Advanced filtering and storage:** Store specific logs in customized files for compliance or analysis.

3. If Journald Is Disabled

If `journald` is not running, here's how logs are handled:

- **Rsyslog still collects logs directly:**
 - Kernel logs: Rsyslog can read directly from `/proc/kmsg`.
 - Application logs: Applications log directly to `/dev/log` (handled by `rsyslog`).
 - Custom logs: Rsyslog configurations determine how and where to capture additional logs.

Outcome:

- You lose some **systemd-specific logs** (like service state changes) because those rely on `journald`.
- Logs will still be written to plaintext files (e.g., `/var/log/messages`), as managed by `rsyslog`.

4. If Rsyslog Is Disabled

If `rsyslog` is not running:

- **Journald captures all system logs:**
 - Journald collects logs from systemd services, the kernel, and applications.
 - Logs are stored in the journal directory (`/run/log/journal` or `/var/log/journal`).

Linux Command

Outcome:

- **No plaintext log files:** Logs are stored in binary format, accessible only via `journalctl`.
 - Remote logging or custom filtering won't be available unless another tool (e.g., Fluentd, Logstash) is used.
-

5. Combined Scenario

- **Journald Active, Rsyslog Active:**
 - Journald captures all logs, then forwards them to Rsyslog (if `ForwardToSyslog=yes`).
 - Rsyslog processes and stores these logs into traditional files or forwards them to a remote server.
 - **Journald Active, Rsyslog Disabled:**
 - Logs are stored only in the binary journal format, accessible via `journalctl`.
 - **Journald Disabled, Rsyslog Active:**
 - Rsyslog captures logs directly from the kernel, applications, and custom inputs.
 - Logs are stored in plaintext files.
-

Summary

- **If only journald is running:** Logs are collected, but they are stored in binary format, and advanced log forwarding/filtering is unavailable.
- **If only rsyslog is running:** Logs are collected from sources directly, but systemd-specific metadata may be missing.
- **If both are running:** Logs are collected by journald, forwarded to rsyslog, and then processed or stored.

Would you like a step-by-step guide to configure either tool for a specific scenario?

You said:

So if both is running it won't to duplication of logs like same logs will capture and send to rsyslog and rsyslog will capture by self as well

ChatGPT said:

ChatGPT

Linux Command

Good question! When **both** `journald` and `rsyslog` are **running**, there is **no duplication of logs** by default because of the way they interact and are configured. Here's how it works and how duplication can be avoided:

How Logging Works with Both Journald and Rsyslog

1. Journald Collects All Logs:

- `Journald` captures logs from the kernel, `systemd` services, and applications.
- It acts as the central collector and stores logs in its binary journal files.

2. Forwarding Logs to Rsyslog:

- If `ForwardToSyslog=yes` is set in `/etc/systemd/journald.conf` (default in most setups), `journald` forwards a copy of its logs to `rsyslog`.
- `Rsyslog` does **not independently capture logs from sources like `/dev/log` or `/proc/kmsg` unless explicitly configured to do so.**

3. Rsyslog Processes Forwarded Logs:

- `Rsyslog` takes logs forwarded by `journald` and processes or stores them in traditional plaintext files (e.g., `/var/log/messages`).

Default Behavior Prevents Duplication

- By default, `Rsyslog` relies on `journald` for log forwarding.
- If `Rsyslog` is configured to independently read sources like `/dev/log`, `/proc/kmsg`, or custom inputs **in addition to receiving forwarded logs**, then duplication can occur.

Avoiding Duplication

To ensure no duplication:

1. Use Only Forwarding from Journald:

- Confirm that `rsyslog` does not duplicate log sources manually:
 - In `/etc/rsyslog.conf`, ensure you're not loading `imuxsock` (for `/dev/log`) or `imklog` (for `/proc/kmsg`) **if `journald` is forwarding logs.**

bash

Copy code

Linux Command

- Only process logs forwarded from journald.

2. Avoid Forwarding from Journald (Optional):

- If you prefer Rsyslog to collect logs directly, disable forwarding in `/etc/systemd/journald.conf`:
plaintext
Copy code
- Configure Rsyslog to read logs from `/dev/log`, `/proc/kmsg`, or other sources directly.

Why Use Both Together?

Running both together is common for these reasons:

- **Journald** captures logs and provides rich metadata (e.g., service states).
- **Rsyslog** provides advanced processing, file-based storage, or remote forwarding.

When configured correctly, journald handles log collection and forwards a single copy to Rsyslog for further action, avoiding duplication.

Summary

- **No duplication occurs by default** if Rsyslog relies only on forwarded logs from Journald.
- Duplication can happen if Rsyslog is also configured to independently collect logs from the same sources.
- Proper configuration in `/etc/systemd/journald.conf` and `/etc/rsyslog.conf` ensures logs are handled efficiently.

Difference Between `/dev/log` and `/proc/kmsg`

Aspect	<code>/dev/log</code>	<code>/proc/kmsg</code>
Source	Applications, system services	Linux kernel
Type of Logs	User-space logs	Kernel-level logs

Linux Command

Aspect	/dev/log	/proc/kmsg
Access	Logging daemons (e.g., <code>journald</code> , <code>rsyslog</code>)	Privileged processes (e.g., <code>dmesg</code>)
Purpose	Captures logs from apps/services	Captures kernel activity
Example Logs	Application errors, service failures	Boot messages, driver errors

Use Cases

- `/dev/log`: Logs from applications (e.g., Apache, SSHD) and services.
- `/proc/kmsg`: Logs from the Linux kernel (e.g., hardware or driver issues).

Would you like configuration examples?

You said:

ok if syslog forward in no the imjournal is also there right to pull the logs from journald

ChatGPT said:

ChatGPT

Yes, that's correct! If `journald` **does not forward logs** to `rsyslog` (e.g., `ForwardToSyslog=no` in `/etc/systemd/journald.conf`), `rsyslog` can still pull logs from `journald` using the `imjournal` module. Here's how it works:

How imjournal Works

- **Purpose:** `imjournal` allows `rsyslog` to read logs directly from `journald`'s binary journal files.
- **Configuration:**
 - You must enable the `imjournal` module in `rsyslog`'s configuration.
 - The module reads logs directly from the `journald` journal files located in `/run/log/journal/` (volatile storage) or `/var/log/journal/` (persistent storage).

Linux Command

When to Use imjournal

- When ForwardToSyslog=no in journald.conf (journald does not forward logs automatically).
- To process and store logs from journald using rsyslog, even if forwarding is disabled.
- To leverage journald's structured logs while still using rsyslog for advanced processing or forwarding.

Example Configuration

In /etc/rsyslog.conf (or a file in /etc/rsyslog.d/):

plaintext
Copy code

Additional options for imjournal can include:

plaintext
Copy code

- StateFile: Ensures rsyslog keeps track of the last read log entry to avoid duplication.

Comparison: Forwarding vs. Pulling

Feature	Forwarding (ForwardToSyslog=yes)	Pulling (imjournal module)
Log Source	Journald forwards logs to /dev/log.	Rsyslog pulls logs from journal files.
Setup Complexity	Simple (enabled by default).	Requires configuring imjournal.
Efficiency	Directly sends logs to rsyslog.	May be slower for large logs due to reading journal files.
Structured Data Support	Limited (syslog format).	Full access to journald's structured data.

Linux Command

Summary

- If `ForwardToSyslog=no`, use `imjournal` to pull logs from `journald`.
- This setup gives `rsyslog` access to all `journald` logs without duplication or direct forwarding.

Here's how you can set up and configure the `imjournal` module in `rsyslog` to pull logs from `journald`:

Steps to Configure `imjournal` for `Rsyslog`

1. Disable Forwarding in `Journald`

- Edit the `journald` configuration file:

```
bash
Copy code
sudo nano /etc/systemd/journald.conf
```

- Ensure the following line is set:

```
plaintext
Copy code
ForwardToSyslog=no
```

- Restart `journald` to apply the changes:

```
bash
Copy code
sudo systemctl restart systemd-journald
```

2. Enable the `imjournal` Module in `Rsyslog`

- Edit the `Rsyslog` configuration file:

```
bash
Copy code
sudo nano /etc/rsyslog.conf
```

- Add the following lines near the top (before other inputs):

```
plaintext
Copy code
module(load="imjournal" StateFile="/var/lib/rsyslog/imjournal.state")
```

Linux Command

- **StateFile**: Ensures Rsyslog keeps track of where it left off when reading logs, avoiding duplication.
- If you're using Rsyslog configuration snippets, you can alternatively create a file:

```
bash
Copy code
sudo nano /etc/rsyslog.d/01-imjournal.conf
```

- Add the same module configuration inside the file.

3. Configure Additional Options (Optional)

- You can customize `imjournal` with additional parameters:
 - **ReadFromJournal="on|off"**: Enable/disable reading directly from the journal (default is `on`).
 - **StateFile**: Location of the file storing the state of the journal read process.
 - **IgnorePreviousMessages="on|off"**: Start from the current log entries, ignoring old logs.

Example:

```
plaintext
Copy code
module(
    load="imjournal"
    StateFile="/var/lib/rsyslog/imjournal.state"
    IgnorePreviousMessages="on"
)
```

4. Restart Rsyslog

- Restart the Rsyslog service to apply changes:

```
bash
Copy code
sudo systemctl restart rsyslog
```

5. Verify Logs

- Check that logs are being pulled from Journald:

```
bash
Copy code
sudo tail -f /var/log/messages
```

- Ensure no duplication or missing entries.
-

Linux Command

Testing and Troubleshooting

1. Check Rsyslog Status:

```
bash
Copy code
sudo systemctl status rsyslog
```

- o Ensure there are no errors related to `imjournal`.

2. Verify `imjournal` Usage:

- o Check the `journal` state file:

```
bash
Copy code
sudo cat /var/lib/rsyslog/imjournal.state
```

- This file should exist and update as logs are read.

3. Debugging Logs:

- o Increase verbosity in `/etc/rsyslog.conf`:

```
plaintext
Copy code
global(debug="on")
```

4. Check `Journal` Logs:

```
bash
Copy code
journalctl -u systemd-journal
```

Summary

- With `imjournal`, `rsyslog` pulls logs directly from `journal`'s journal files.
- This setup avoids the need for `ForwardToSyslog` and provides structured access to logs.

How to Fix inode Issue

Identify and Remove Unnecessary Small Files:

1. Find directories with a large number of files:

```
find /mnt -xdev -type f | wc -l
```

2. Identify which directories are consuming the most inodes:

```
for dir in /mnt/*; do echo "$(find "$dir" -type f | wc -l) files in $dir"; done | sort -nr | head
```

Linux Command

3. Delete unnecessary files:

```
rm -rf /mnt/path/to/unwanted/files
```