

TOP 50 LINUX COMMANDS FOR DEVOPS



System Info Commands

1

System Info Commands

hostname – shows the name of the system host.

→ ~ **hostname**
localhost

hostid – shows the host id of the system assigned by the OS.

→ ~ **hostid**
0a123456

date - shows the current date and time in UTC format.

→ ~ **date**
Wed Jan 19 12:34:56 UTC 2024

1

System Info Commands

uptime - shows the elapsed time duration since the machine logged in.

→ ~ `uptime`

```
12:34:56 up 1 day, 3:45, 2 users, load average: 0.25, 0.20, 0.18
```

uname – unix name.

→ ~ `uname`

```
Linux
```

1

System Info Commands

clear - clears the screen.

→ ~ clear

history - lists all the commands executed until now.

→ ~ history

1 ls

2 cd Documents

3 nano file.txt

4 gcc program.c -o program

5 ./program

6 history

1

System Info Commands

sudo - Super User Do.

→ ~ `sudo su -
USERNAME`

`echo $?` - shows the exit status of the last executed command (0 - success, 1-255 - error/failure).

→ ~ `echo $?
127`

1

System Info Commands

shutdown -r now - restart the machine immediately (-r restart).

```
→ ~ sudo shutdown -r now  
Broadcast message from user@hostname  
(/dev/pts/0) at 12:34 ...  
  
The system is going down for reboot NOW!
```

printenv - displays all the environment variables of the Linux system.

```
→ ~ printenv  
TERM=xterm-256color  
SHELL=/bin/bash  
USER=your_username  
...
```


1

System Info Commands

last - shows previous logins in the Linux system.

→ ~ last

root pts/0 Wed Jan 19 12:34 still logged in

reboot system boot 5.4.0-96-generic Wed Jan 19 12:33 still running

systemctl — System Control: Manage system services using systemd.

System Info Commands

→ ~ systemctl status sshd

- sshd.service - OpenBSD Secure Shell server

Loaded: loaded (/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)

Active: active (running) since Wed 2024-01-19 12:34:56 UTC; 1 day 3h ago

Docs: man:sshd(8)

man:sshd_config(5)

Process: 1234 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)

Main PID: 5678 (sshd)

Tasks: 1 (limit: 1234)

Memory: 2.3M

CPU: 12ms

CGroup: /system.slice/sshd.service

└─5678 /usr/sbin/sshd -D

Jan 19 12:34:56 hostname systemd[1]: Starting OpenBSD Secure Shell server...

Jan 19 12:34:56 hostname sshd[5678]: Server listening on 0.0.0.0 port 22.

Jan 19 12:34:56 hostname sshd[5678]: Server listening on :: port 22.

Jan 19 12:34:56 hostname systemd[1]: Started OpenBSD Secure Shell server.

2

File Commands

2

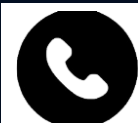
File Commands

touch - creates an empty file or updates timestamp of the existing file.

- **touch <fileName>** - creates a single empty file.
- **touch <file1> <file2>** - creates file1, file2 empty files.

cat - concatenates and displays the contents of files.

- **cat <fileName>** - displays the contents of the file.
- **cat > <fileName>** - creates a new file, allows to input content interactively and redirects inputted content to the created file (> redirection operator).



2

File Commands

head <fileName> - displays first 10 lines of the file by default.

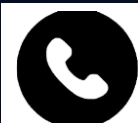
- **head -n 5 <fileName>** - displays first 5 lines of the file (-n number)

→ ~ head -n 5 help.txt

1. Commands shortcut

....

5. huddle - Connect to Syncup Call



2

File Commands

tail <fileName> - displays the last 10 lines of the file by default.

- **tail -F <fileName>** - displays contents of the file in real-time even when the file is rotated or replaced (used for log file monitoring).

→ ~ **tail -F** mySystem.logs

echo "I love DevOps"

echo "Best Linux commands"

....



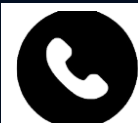
2

File Commands

less <fileName> - used to view large files (log files) in a paginated manner.

rm - remove command.

- **rm <fileName>** - removes the file.
- **rm -r <dirName>** - removes files & folders of directory recursively (-r recursive).
- **rm -rf <dirName>** - force remove the files & folders of directory recursively (-f force).
- Example: `rm -r ./test`



2

File Commands

cp - copy command.

- **cp <source> <destination>** - copy the files and folders from source to destination.
- **cp -r <dir1> <dir2>** - copy dir1 directory to dir2 directory recursively (-r recursive).
- Example: `cp -r ./sourceDir ./destiDir`



3

File Permission Commands

3

File permission Commands

ls -l <pathOfFileName> - shows the permissions of the file.

• → ~ ls -l .

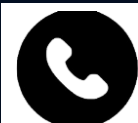
total 1016

-rw-r--r-- 1 vinodhakumara staff 48 Jan 19 21:06 crazy.sh

-rw-r--r-- 1 vinodhakumara staff 2463 Jan 2 11:25 help

-rw-r--r-- 1 vinodhakumara staff 48 Jan 19 22:14 mySystem.logs

drwxr-xr-x@ 8 vinodhakumara staff 256 Dec 20 12:51 observability-signoz



3 File permission Commands

ls -ld <dirNamePath> - shows the permissions of the directory.

→ ~ ls -ld Downloads

```
drwx-----@ 53 vinodhakumaral staff 1696 Jan 19 21:00 Downloads
```

chmod <octalNumber> <fileName> - changes mode/permissions of the file.

- Example: `chmod 742 test.txt`

chmod <octalNumber> -R <dirName> - changes mode/permissions of the directory recursively.

3 File permission Commands

ls -ld <dirNamePath> - shows the permissions of the directory.

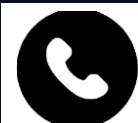
→ ~ ls -ld Downloads

```
drwx-----@ 53 vinodhakumara staff 1696 Jan 19 21:00 Downloads
```

chmod <octalNumber> <fileName> - changes mode/permissions of the file.

- Example: `chmod 742 test.txt`

`chmod <octalNumber> -R <dirName>` - changes mode/permissions of the directory recursively.



3 File permission Commands

chown <newUser> <fileName> - changes the user ownership of a file.

- **Example:** `chown rocky test.txt`

chown <newUser>:<newGroup> <fileName> - changes the user & group ownerships of a file

chgrp <groupName> <fileName/dirName> - updates the group name for file/directory.

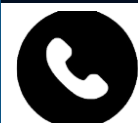


3 File permission Commands

- Example: `chgrp mygroup ./test`

getfacl <fileName/dirName> - shows the file/directory access control list

```
→ ~ getfacl filename.txt
# file: filename.txt
# owner: user1
# group: group1
user::rw-
```



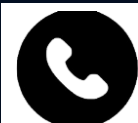
3 File permission Commands

```
group::r—  
other::r--
```

setfacl -m u:<userName>:rwx <fileName/dirName> -
modifies the current acl of the file/directory.

setfacl -x u:<userName>: <fileName/dirName> - removes the
acl permissions for the file/directory.

setfacl -m g:<groupName>:rwx <fileName/dirName> -
modifies the group acls for the file/directory.



3 File permission Commands

setfacl -x g:<groupName>: <fileName/dirName> - removes the group acl permissions for the file/directory.





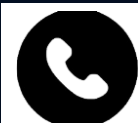
File Permission Octal Number

4

File Permission Octal Number

read (r) — 4, write (w)- 2, execute (x) — 1

- *Sum the numbers to generate an octal number for setting permissions on a file or directory.*



5

User Management Commands

5

User Management Commands

ac — Total connect time for all users or specified users.

- The `ac` command reads the `/var/log/wtmp` file, which contains binary data about every login, logout, system event, and current status on the system. It gets its data from the `wtmp` file.

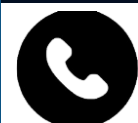
- Display total login time of a specific user.

`ac john`

- Display total login time for each user.

`ac -p`

- Display total login time for each day.



5

User Management Commands

ac — Total connect time for all users or specified users.

- **ac -d**

- Display total login time for the current day.

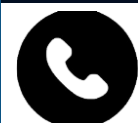
ac -d -p

- Display login time from a specific log file.

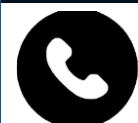
ac -f /var/log/wtmp

useradd - Creates a user account.

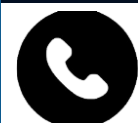
useradd <userName> - Creates user account without home & mail spool directories



- Example: `useradd bot`
- Example: `useradd -m bot`
- `useradd -m <userName>` - Creates user account with home & mail spool directories.
- `ususerdel` - Deletes User Account.
- `userdel <userName>` - deletes the user from the system.
- `userdel -r <userName>` - deletes the user from the system along with home and mail spool directories.
- Example: `userdel -r bot`
- `/etc/passwd` - Stores information about user accounts.



- **passwd <userName>** - The system generates a password for the user and then stores it in the **/etc/shadow** file.
- **cat /etc/passwd** - displays the complete list of users on that machine.





Group Management Commands

6

Group Management Commands

groupadd <groupName> - creates the group.

groupdel <groupName> - delete the group.

/etc/group - stores the information of the groups.

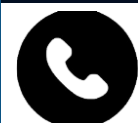
cat /etc/group - displays the complete list of groups on that machine.

gpasswd <groupName> - creates a password for the group.

- **gpasswd -a <userName> <groupName>** - adds the user to the group.

- **gpasswd -d <userName> <groupName>** - removes the user from the group.

52



- **gpasswd -M <userName1>,<userName2>,<userName3>**
<groupName> - adds multiple users to the group and removes the existing ones of the group.

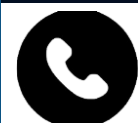


Searching Commands

Searching Commands

find — Search for files/directories based on their names.

- **find . -name <fileName>** - finds the mentioned file if available in the current directory (.(period) represents current directory).
- **find <dirName> -name <fileName>** - finds the mentioned file in the directory.
- **find <dirName> -perm 754** - finds the files in the directory having 754 permission.



7

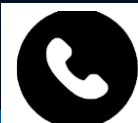
Searching Commands

locate is faster for finding files by name due to its pre-built database, while **find** is more versatile, allowing complex searches based on various criteria in real-time.

locate - Search for files/directories based on their names.

- **locate <fileName/dirName>** - locates the file/directory and displays the path.

- **Example:** locate crazy.txt





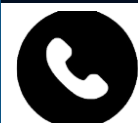
**GREP Command_Global
Regular Expression Print**

8

GREP Command — Global Regular Expression Print

find — Search for files/directories based on their names.

- **grep <textToSearch> <fileName>** - used to find text patterns within files.
- **grep -i <textToSearch> <fileName>** - used to find text patterns within the file ignoring the case (-i ignore case).
- **grep -v <textToSearch> <fileName>** - used to find non matching lines of text patterns (-v invert-match).



8

GREP Command — Global Regular Expression Print

find — Search for files/directories based on their names.

- `grep -l <textToSearch> <fileNames>` - used to display the matching string file names.
- Example: `grep -l welcome crazy.txt`
- There are additional commands related to grep.
- `egrep` (or `grep -E`)



8

GREP Command — Global Regular Expression Print

find — Search for files/directories based on their names.

- fgrep (or `grep -F`)
- zgrep (for compressed files)
- zegrep (or `zgrep -E` for compressed files)
- egrep (or `grep -E`)
- bzgrep (for compressed files)

40





Hardware Information Commands

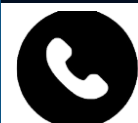
9

Hardware Information Commands

free -h - Display system memory information in human-readable format (-h). names.

df -h - It displays the disk space usage of mounted file systems.

du - Disk usage.



9

Hardware Information Commands

- **du -h** - Display disk usage information in human-readable format.
- **du -sh** - Display the total size of the directory in human-readable format, summarizing the size instead of listing individual file sizes.
- **du -sh <fileName/dirName>** - Displays the total size of the file/directory.

10

Connection To Remote System

10 Connection To Remote System

ssh - Secure Shell: Connect to a remote server securely.

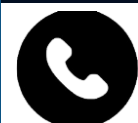
Example: `ssh user@remote_host`

scp - Securely Copy Files: Copy files between local and remote systems using SSH.

Example: `scp file.txt user@remote_host:/path`

rsync - Remote Sync: Synchronize files and directories between systems.

Example: `rsync -avz local_folder/
user@remote_host:remote_folder/`



11

Network Commands

11

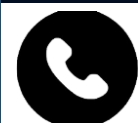
Network Commands

- nc — Simple tcp proxy, network daemon testing

- Example: `nc -vz google.com 443`

- ping **<hostName>** - tests the reachability & responsiveness of the remote host.

- Example: `ping google.com -c 2` (-c pings 2 times)



Network Commands

• **dig** `<domainName>` - Shows DNS information of the domain.

- Example: `dig medium.com`

• **wget** `<url>` - Used to retrieve/download files from the internet.

- **curl** - client URL.

Network Commands

- **curl <url>** - Used to retrieve/download files from the internet.
- **ifconfig** - Display available network interfaces.
- **ip addr** - Display and manipulate network interface info.
- **curl ifconfig.me** - Shows the public ip address of the machine.
- **netstat -antp** - shows all tcp open ports (-a all, t-tcp, n-active, p protocol).
- **traceroute <url>** - traces the route using packets from source to destination host.

