# Basic Linux commands used by DevOps engineers in day-to-day activities

Rahul Desharaj · Follow
7 min read · Apr 25, 2024

Listen Share

This article will help in understanding most of the important and majorly used Linux commands that would be required for a DevOps Engineer.

## System Info Commands :-

**`hostname`** - shows the name of the system host.



**`hostid`** - shows the host id of the system assigned by the OS

```
[ ec2-user@ip-172-31-27-200 ~ ]$ hostid
1facc81b
```

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

```
[ ec2-user@ip-172-31-27-200 ~ ]$ date
Thu Apr 25 04:33:09 UTC 2024
```

**`whoami`** - shows the currently logged-in username of the terminal.

```
[ ec2-user@ip-172-31-27-200 ~ ]$ whoami
ec2-user
```

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

```
[ ec2-user@ip-172-31-27-200 ~ ]$ uptime
 04:34:08 up 14 min,  1 user,  load average: 0.00, 0.01, 0.02
```

**`uname`** - unix name.

```
[ ec2-user@ip-172-31-27-200 ~ ]$ uname
Linux
```

**`clear`** - clears the screen.

**`history`** - lists all the commands executed until now.

```
[ ec2-user@ip-172-31-27-200 ~ ]$ history
    1  25/04/24 04:30:05 clear
    2  25/04/24 04:30:12 hostname
    3  25/04/24 04:32:24 hostid
    4  25/04/24 04:33:09 date
    5  25/04/24 04:33:36 whoami
    6  25/04/24 04:34:08 uptime
    7  25/04/24 04:34:38 uname
    8  25/04/24 04:35:05 clear
    9  25/04/24 04:35:21 history
```

**`sudo`** - Super User Do.

```
[ ec2-user@ip-172-31-27-200 ~ ]$ sudo su -

3.85.210.8 | 172.31.27.200 | t2.micro | null
[ root@ip-172-31-27-200 ~ ]#
```

**`echo $?`** - shows the exit status of the last executed command (0 - success, 1–127 - error/failure).

```
[ root@ip-172-31-27-200 ~ ]# echo $?
0
```

## Directory Commands :-

**`pwd`** - To check where you are currently in the system we use `pwd` command.

```
[ root@ip-172-31-27-200 ~ ]# pwd
/root
```

**`cd`** To change the working directory from one location to another we use `cd` command

**`cd ..`** In Linux, we refer the parent directory with `..` , So in case if we want to go to parent path then simply we can go with.

```
[ root@ip-172-31-27-200 ~ ]# cd sample/

3.85.210.8 | 172.31.27.200 | t2.micro | null
[ root@ip-172-31-27-200 ~/sample ]# cd ..

3.85.210.8 | 172.31.27.200 | t2.micro | null
[ root@ip-172-31-27-200 ~ ]#
```

**`mkdir`** - make directory.

```
[ root@ip-172-31-27-200 ~ ]# mkdir rahul

3.85.210.8 | 172.31.27.200 | t2.micro | null
[ root@ip-172-31-27-200 ~ ]# cd rahul/

3.85.210.8 | 172.31.27.200 | t2.micro | null
[ root@ip-172-31-27-200 ~/rahul ]#
```

## File Commands :-

`touch` This Command by default creates an empty file.

```
[ root@ip-172-31-27-200 ~/rahul ]# touch text
```

`ls -l` To check the file created.

```
[ root@ip-172-31-27-200 ~/rahul ]# touch text

3.85.210.8 | 172.31.27.200 | t2.micro | null
[ root@ip-172-31-27-200 ~/rahul ]# ls -l
total 0
-rw-r--r-- 1 root root 0 Apr 25 04:45 text
```

`vim` : This is a text editor used in Linux. It stands for “Vi Improved”.

- Normal mode: This is the default mode in which vim starts. In normal mode, you can use various commands to navigate and edit the text.
- Insert mode: In insert mode, you can type text into the file. To enter insert mode, press the “i” key. To exit insert mode and return to normal mode, press the “Esc” key.
- Command mode: In command mode, you can enter commands to perform various actions, such as saving the file or quitting vim. To enter command mode, press the “:” key.

`cat` - concatenates and displays the contents of files.

```
[ root@ip-172-31-27-200 ~/rahul ]# cat text
this is sample file
```

`rm` - remove command.

- `rm -f <fileName>` - removes the file.

```
[ root@ip-172-31-27-200 ~/rahul ]# rm -f text
```

- `rm -rf <dirName>` - force remove the files & folders of directory recursively (-f force).

```
[ root@ip-172-31-27-200 ~/rahul ]# cd

3.85.210.8 | 172.31.27.200 | t2.micro | null
[ root@ip-172-31-27-200 ~ ]# rm -rf rahul/

3.85.210.8 | 172.31.27.200 | t2.micro | null
[ root@ip-172-31-27-200 ~ ]#
```

`cp` - copy command.

`cp <source> <destination>` - copy the files and folders from source to destination.

`mv` - move or rename command.

`mv <fileName> <newFileName>` - renames the file to new name.

```
[ root@87387e37d4b5 ~ ]# mv notes.txt note.txt
[ root@87387e37d4b5 ~ ]# ls -l
total 20
-rw------- 1 root root 4115 Feb 14 04:11 anaconda-ks.cfg
-rw-r--r-- 1 root root   95 Feb 14 04:11 anaconda-post-nochroot.log
-rw-r--r-- 1 root root  483 Feb 14 04:11 anaconda-post.log
-rw-r--r-- 1 root root    0 Apr 25 05:01 note.txt
-rw------- 1 root root 3850 Feb 14 04:11 original-ks.cfg
```

## Network Commands :-

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

```
[ root@ip-172-31-27-200 ~ ]# hostname
ip-172-31-27-200.ec2.internal

3.85.210.8 | 172.31.27.200 | t2.micro | null
[ root@ip-172-31-27-200 ~ ]# ping ip-172-31-27-200.ec2.internal
PING ip-172-31-27-200.ec2.internal (172.31.27.200) 56(84) bytes of data.
64 bytes from ip-172-31-27-200.ec2.internal (172.31.27.200): icmp_seq=1 ttl=64 time=0.016 ms
64 bytes from ip-172-31-27-200.ec2.internal (172.31.27.200): icmp_seq=2 ttl=64 time=0.055 ms
64 bytes from ip-172-31-27-200.ec2.internal (172.31.27.200): icmp_seq=3 ttl=64 time=0.050 ms
```

`ifconfig` - display available network interfaces.

```
[ root@ip-172-31-27-200 ~ ]# ifconfig
enX0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
        inet 172.31.27.200  netmask 255.255.240.0  broadcast 172.31.31.255
        inet6 fe80::8ff:e6ff:fe0a:7843  prefixlen 64  scopeid 0x20<link>
        ether 0a:ff:e6:0a:78:43  txqueuelen 1000  (Ethernet)
        RX packets 2493  bytes 224019 (218.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2866  bytes 305409 (298.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 132  bytes 11088 (10.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 132  bytes 11088 (10.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

**`netstat -lntp`** - shows all tcp open ports (-a all, t-tcp, n-active, p protocol).

```
[ root@ip-172-31-27-200 ~ ]# netstat -lntp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1246/sshd: /usr/sbi
tcp6       0      0 :::22                   :::*                    LISTEN      1246/sshd: /usr/sbi
```

**Nslookup:** This stands for “Name server Lookup”. This is a tool for checking DNS hostname to Ip or Ip to Hostname. This is very helpful while troubleshooting.

```
[ root@ip-172-31-27-200 ~ ]# nslookup google.com
Server:         172.31.0.2
Address:        172.31.0.2#53

Non-authoritative answer:
Name:   google.com
Address: 172.253.63.139
Name:   google.com
Address: 172.253.63.100
Name:   google.com
Address: 172.253.63.101
Name:   google.com
Address: 172.253.63.102
Name:   google.com
Address: 172.253.63.113
Name:   google.com
Address: 172.253.63.138
Name:   google.com
Address: 2607:f8b0:4004:c08::64
Name:   google.com
Address: 2607:f8b0:4004:c08::71
Name:   google.com
Address: 2607:f8b0:4004:c08::8a
Name:   google.com
Address: 2607:f8b0:4004:c08::8b
```

## Process Information Commands :-

- `ps` - shows the currently running process.

```
[ root@ip-172-31-27-200 ~ ]# ps
    PID TTY          TIME CMD
   1421 pts/0    00:00:00 sudo
   1422 pts/0    00:00:00 su
   1423 pts/0    00:00:00 bash
   1641 pts/0    00:00:00 ps
```

- `ps -ef` - shows all the processes of the system.

```
[ root@ip-172-31-27-200 ~ ]# ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 04:19 ?        00:00:01 /usr/lib/systemd/systemd --switched-root --system --deserialize 31
root           2       0  0 04:19 ?        00:00:00 [kthreadd]
root           3       2  0 04:19 ?        00:00:00 [rcu_gp]
root           4       2  0 04:19 ?        00:00:00 [rcu_par_gp]
root           5       2  0 04:19 ?        00:00:00 [slub_flushwq]
root           6       2  0 04:19 ?        00:00:00 [netns]
root           8       2  0 04:19 ?        00:00:00 [kworker/0:0H-events_highpri]
root           9       2  0 04:19 ?        00:00:00 [kworker/u30:0-events_unbound]
root          10       2  0 04:19 ?        00:00:00 [mm_percpu_wq]
root          12       2  0 04:19 ?        00:00:00 [rcu_tasks_kthre]
root          13       2  0 04:19 ?        00:00:00 [rcu_tasks_rude_]
```

- **`top`** - shows the real-time, dynamic view of the running processes of a system.

```
top - 05:17:43 up 58 min,  1 user,  load average: 0.45, 0.15, 0.05
Tasks: 150 total,   1 running, 149 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.3 us,  0.3 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :    754.0 total,    433.8 free,    220.6 used,    206.5 buff/cache
MiB Swap:   2048.0 total,   2021.5 free,     26.5 used.    533.3 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   1615 root      20   0       0      0      0 I   0.3   0.0   0:00.33 kworker/0:2-events
      1 root      20   0  103880   6664   3356 S   0.0   0.9   0:01.49 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flushwq
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
```

- **`kill <pid>`** - gracefully terminates the process pid.

```
[ root@ip-172-31-27-200 ~ ]# kill 1
```

- **`df -h`** (disk free) command will have an account of available disk space, used by file system.

```
[ root@ip-172-31-27-200 ~ ]# df -h
Filesystem                 Size  Used Avail Use% Mounted on
devtmpfs                   4.0M     0  4.0M   0% /dev
tmpfs                      377M     0  377M   0% /dev/shm
tmpfs                      151M  2.5M  149M   2% /run
/dev/mapper/RootVG-rootVol 6.0G  2.0G  4.0G  33% /
tmpfs                      377M     0  377M   0% /tmp
/dev/mapper/RootVG-homeVol 960M   40M  921M   5% /home
/dev/mapper/RootVG-varVol  2.0G  342M  1.7G  18% /var
/dev/mapper/RootVG-logVol  2.0G   73M  1.9G   4% /var/log
/dev/mapper/RootVG-varTmpVol 2.0G  47M  1.9G   3% /var/tmp
/dev/mapper/RootVG-auditVol 4.4G  64M  4.3G   2% /var/log/audit
/dev/xvda3                 424M  320M  105M  76% /boot
/dev/xvda2                 122M  7.1M  115M   6% /boot/efi
tmpfs                       76M     0   76M   0% /run/user/1001
```

## Red Hat Package related Commands

Yum - Package Manager for RHEL Linux distributions.

- **yum** - a newer version of the package manager with colorized output, progress bar and additional functions.

**`yum update -y`** - updates the package list.

```
[ root@ip-172-31-27-200 ~ ]# yum update -y
Last metadata expiration check: 0:16:40 ago on Thu Apr 25 05:33:40 2024.
Dependencies resolved.
Nothing to do.
Complete!
```

**`yum list --installed`** - lists all the installed packages.

```
[ root@ip-172-31-27-200 ~ ]# yum list --installed
Installed Packages
NetworkManager.x86_64                    1:1.44.0-5.el9_3                @rhel-9-baseos-rhui-rpms
NetworkManager-libnm.x86_64              1:1.44.0-5.el9_3                @rhel-9-baseos-rhui-rpms
NetworkManager-team.x86_64               1:1.44.0-5.el9_3                @rhel-9-baseos-rhui-rpms
NetworkManager-tui.x86_64                1:1.44.0-5.el9_3                @rhel-9-baseos-rhui-rpms
acl.x86_64                               2.3.1-3.el9                     @System
alternatives.x86_64                      1.24-1.el9                      @System
amazon-libdnf-plugin.x86_64              1.0.1-1.el9                     @System
amazon-ssm-agent.x86_64                  3.2.2222.0-1                    @System
audit.x86_64                             3.0.7-104.el9                   @System
audit-libs.x86_64                        3.0.7-104.el9                   @System
authselect.x86_64                        1.2.6-2.el9                     @System
```

## Service Management

**sudo systemctl list-units -t service** : To list all the services which are running in the Operating System

```
[ root@ip-172-31-27-200 ~ ]# sudo systemctl list-units -t service
  UNIT                          LOAD   ACTIVE SUB     DESCRIPTION
  amazon-ssm-agent.service      loaded active running amazon-ssm-agent
  auditd.service                loaded active running Security Auditing Service
  chronyd.service               loaded active running NTP client/server
  cloud-config.service          loaded active exited  Apply the settings specified in cloud-config
  cloud-final.service           loaded active exited  Execute cloud user/final scripts
  cloud-init-local.service      loaded active exited  Initial cloud-init job (pre-networking)
  cloud-init.service            loaded active exited  Initial cloud-init job (metadata service crawler)
  crond.service                 loaded active running Command Scheduler
  dbus-broker.service           loaded active running D-Bus System Message Bus
  dracut-shutdown.service       loaded active exited  Restore /run/initramfs on shutdown
  getty@tty1.service            loaded active running Getty on tty1
  import-state.service          loaded active exited  Import network configuration from initramfs
  kmod-static-nodes.service     loaded active exited  Create List of Static Device Nodes
```

**sudo systemctl start nginx** : To start a service
**syntax : sudo systemctl start <service name>**

```
[ root@ip-172-31-27-200 ~ ]# sudo systemctl start nginx
```

**sudo systemctl status nginx :** check the status of the service

```
[ root@ip-172-31-27-200 ~ ]# sudo systemctl status  nginx
● nginx.service - The nginx HTTP and reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; preset: disabled)
     Active: active (running) since Thu 2024-04-25 06:31:16 UTC; 2min 44s ago
    Process: 27622 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
    Process: 27623 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
    Process: 27624 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
   Main PID: 27625 (nginx)
      Tasks: 2 (limit: 4300)
     Memory: 2.2M
        CPU: 89ms
     CGroup: /system.slice/nginx.service
             ├─27625 "nginx: master process /usr/sbin/nginx"
             └─27627 "nginx: worker process"
```

**sudo systemctl restart nginx :** To restart a service

```
[ root@ip-172-31-27-200 ~ ]# sudo systemctl restart nginx

3.85.210.8 | 172.31.27.200 | t2.micro | null
```

**Other Commands :-**

**Grep:** This command searches for a particular string/ word in a text file. This is similar to “Ctrl+F” but executed via a CLI.

```
[ root@ip-172-31-27-200 ~ ]# ps -ef | grep nginx
root       27371    1423  0 05:33 pts/0    00:00:00 grep --color=auto nginx
```

**Tail:** This command prints the last N number of data of the given input. By default, it prints 10 lines.

We can specify the number of lines, that we want to display.

```
[ root@ip-172-31-27-200 ~ ]# ps -ef | tail -5
root       27184       2  0 05:32 ?        00:00:00 [kworker/0:0-xfs-sync/dm-4]
root       27216       2  0 05:33 ?        00:00:00 [kworker/u30:0-writeback]
root       27217       2  0 05:33 ?        00:00:00 [kworker/u30:1]
root       27384    1423  0 05:35 pts/0    00:00:00 ps -ef
root       27385    1423  0 05:35 pts/0    00:00:00 tail -5
```

**Head:** This command prints the first N number of data of the given input. By default, it prints 10 lines.

We can specify the number of lines, that we want to display.

```
[ root@ip-172-31-27-200 ~ ]# ps -ef | head -5
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 04:19 ?        00:00:02 /usr/lib/systemd/systemd --system --deserialize 39
root           2       0  0 04:19 ?        00:00:00 [kthreadd]
root           3       2  0 04:19 ?        00:00:00 [rcu_gp]
root           4       2  0 04:19 ?        00:00:00 [rcu_par_gp]
```

**Free:** This command displays the total amount of free space available along with the amount of memory used and swap memory in the system, and also the buffers used by the kernel.

```
[ root@ip-172-31-27-200 ~ ]# free
               total        used        free      shared  buff/cache   available
Mem:          772076      234796      208888        2160      451100      537280
Swap:        2097148       45824     2051324
```

**ssh-keygen:** This command is used to generate a public/private authentication key pair.

This process of authentication allows the user to connect remote server without providing a password.

```
[ root@ip-172-31-27-200 ~ ]# ssh-keygen -f rahul
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in rahul
Your public key has been saved in rahul.pub
The key fingerprint is:
SHA256:j9vRH6m1nF8+L6CgLIvCyECZ7/fCuX4FYl2LVKSdYok root@ip-172-31-27-200.ec2.internal
The key's randomart image is:
+---[RSA 3072]----+
|        oo       |
|      ..+..      |
|  o  Eo=oo.      |
| +   o.+..       |
|. . . . S        |
|.  .    .+ ..  . |
|= . ....o.o...+ .|
|oo o.=o. o.. =.*.|
| .. +==.. . . =o*|
+----[SHA256]-----+
```

**CURL:** Curl is a tool used for transferring data to or from a server, using various protocols, such as HTTP, HTTPS, FTP, and more. Basic example:

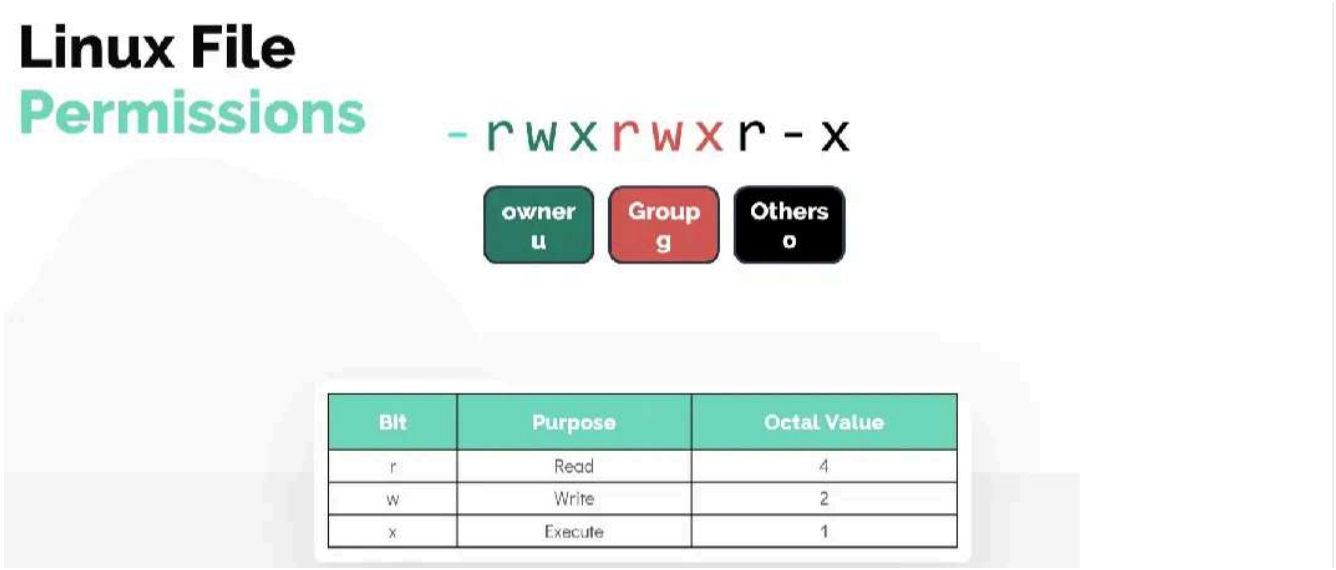
Syntax: curl <url>

```
[ root@ip-172-31-27-200 ~ ]# curl google.com
<HTML><HEAD><meta http-equiv="content-type" content="text/html;charset=utf-8">
<TITLE>301 Moved</TITLE></HEAD><BODY>
<H1>301 Moved</H1>
The document has moved
<A HREF="http://www.google.com/">here</A>.
</BODY></HTML>
```

**tar:** is used to create, maintain, modify, and extract files from archives, often called “tarballs”. It can bundle multiple files and directories into a single archive file, preserving permissions, timestamps, and directory structures.

**cron and crontab:** cron is a system daemon used to execute scheduled tasks at predefined times. crontab is the command-line interface to manage these scheduled tasks for individual users.
Example Usage:
Editing user's crontab: crontab -e
Specifying a cron job to run every hour: 0 * * * * /path/to/script.sh

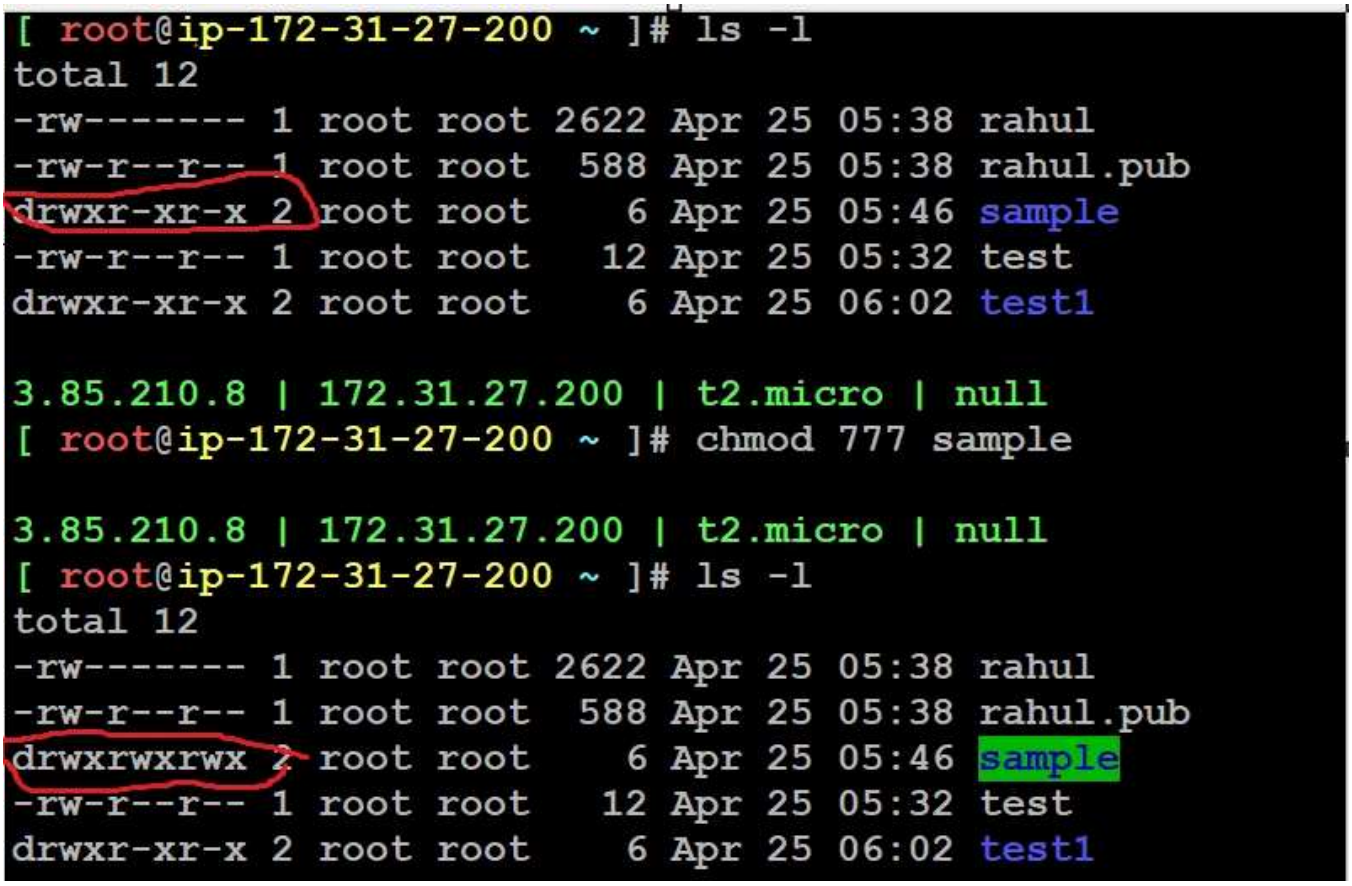
## File Permission Octal Numbers




```
[ root@ip-172-31-27-200 ~ ]# ls -l
total 12
-rw------- 1 root root 2622 Apr 25 05:38 rahul
-rw-r--r-- 1 root root  588 Apr 25 05:38 rahul.pub
drwxr-xr-x 2 root root    6 Apr 25 05:46 sample
-rw-r--r-- 1 root root   12 Apr 25 05:32 test
drwxr-xr-x 2 root root    6 Apr 25 06:02 test1
```

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

Syntax : chmod 777 test1.txt




This blog page covers most of the majorly used Linux commands for DevOps Warriors and is a growing document. Commands will be added as the days go and when I find some interesting commands on the same.

Do follow me for more such content related to DevOps world.

**Reach me** 👨‍💻

- Linkedin -🟦

Follow

## Written by Rahul Desharaj

164 Followers