

Basic Commands:

-> root user Prompt

\$ -> user working Prompt

\$logname -> Displays current username

\$pwd -> Present working directory

Date -> Display date & time

Cal -> Display current month Calender

\$cal 2020 -> Display 2020 calender

\$who -> To display information of all the the users currently logged in

\$Man -> Man stands for manual

\$Man ls

\$whoami -> It displays current username

\$finger -> It displays complete information of all the the users currently logged in

\$uptime -> How long server up and running , how many users connected and load avg time

\$which -> Gives command location

\$Which date

\$whereis -> similar to which

\$whereis date

\$tty -> terminal position

\$df -> Displays disk free size

\$du -> Displays Disk usage information

\$clear -> To clear the screen

\$echo -> outputs or displays a string

\$echo "sample test"

Output: sample test

\$poweroff : poweroff the server

Or

\$shutdown

\$wget : To Download any file from online

\$wget softwarelink

\$curl: To Download any file from online

\$curl -o newname softwarelink

\$history : To get the history of commands.

\$history

\$history 10

\$history -c (lock the history)

\$history -u (unlock the history)

\$rm .bash_history (Removes the history)

\$sudo su : switch to super user

\$su : switch to super user with password

\$su username : switch to other user

Managing files and directories in Linux

Creating files:

Cat (Concatenate) :

It is used to create a file and display, appending the contents of a file

To create a file:

```
$cat > filename
```

```
Hello world
```

```
Ctrl+d ( To save the file)
```

To display the content of the file:

```
$cat filename
```

Or

```
$cat < filename
```

To append data in the existing file:

```
$cat >> filename
```

```
Hello world
```

```
Ctrl + d ( To save)
```

Touch : To create a file, but are empty

```
$touch filename
```

To create multiple files

```
$touch file1 file2
```

ls : Displays the contents of a directory

Sys: \$ ls [options]

-r reverse

-l inode (index number)

-a hidden files

-R -> recursively

-l -> Lognlist

-h -> human readable

Mkdir : Create Directory

\$mkdir <directory_name>

To create multiple directories:

\$mkdir dir1 dir2 dir3

To create nested directory

\$mkdir -p world/asia/india

rmdir : Remove directory

\$rmdir directory

To remove directory containing files

\$rmdir -r directoryname

tree :To check directory structure

\$tree world

Or

ls -r

Navigation commands:

Cd : changes the current location

\$Cd directory_name

Cd .. -> To go one level back

Cd ../.. -> To go two levels back

Cp: copies files or directories from one location to another

\$cp [options] source destination

-R -> copies files recursively

-f -> Copies files forcefully

-v -> provides verbose output

\$cp file1 /opt/file1 (copy file1 to /opt directory

\$cp /var/log/file1 . (Here . means current directory

mv :

Moves or renames the files or directories

Renaming:

\$mv file_old_name file_new_name

Moving:

```
$mv file1 /opt/file1
```

Rm : Deletes the files

```
$ rm [options] file
```

-i interactive

-r recursive

-f forcefully

Delete file:

```
$rm filename
```

file : Displays the type of a file

```
$file test1
```

Output: test1 : file

head : Displays top 10 lines of the file

```
$head filename
```

Display top 5 lines

```
Head -5 filename
```

tail : Displays last 10 lines of the file

```
$tail filename
```

Display last 5 lines

\$tail -5 filename

wc : Provides a word or line count

\$wc [options] file

-l lines

-w words

-c characters

\$wc filename (Display lines words and characters)

\$wc -l filename (only lines)

\$wc -w filename (only words)

less : To display less file contents in Page-wise

\$ less filename

more : To display more contents of file in Page-wise

\$more filename

Shell commands:

To know available shells:

```
$ls /bin/*sh
```

To check current working shell:

```
$echo $0
```

To shift from bash to sh shell:

```
$sh
```

To shift from sh to bash:

```
$bash
```

exit : exit from the shell

Important Ports:

- 20 – FTP Data (For transferring FTP data)
- 21 – FTP Control (For starting FTP connection)
- 22 – SSH (For secure remote administration which uses SSL to encrypt the transmission)
- 23 – Telnet (For insecure remote administration)
- 25 – SMTP (Mail Transfer Agent for e-mail server such as SEND mail)
- 53 – DNS (Special service which uses both TCP and UDP)
- 68 – DHCP
- 80 – HTTP/WWW(Apache)
- 123 – NTP (Network time protocol used for time syncing uses UDP protocol)
- 161 – SNMP (For network monitoring)
- 443 – HTTPS (HTTP+SSL for secure web access)
- 3306 – MySql
- 3389 - Windows RDP
- 3690 – SVN
- 8080 - Jenkins and Tomcat

Networking commands

ifconfig:

view the IP Address, Hardware / MAC address

```
$ifconfig
```

ifconfig -a

List all interfaces

```
$ifconfig
```

Ping:

ping (Packet INternet Groper) is a utility normally used for testing connectivity between two systems on a network (Local Area Network (LAN) or Wide Area Network (WAN)). It uses ICMP (Internet Control Message Protocol) to communicate to nodes on a network.

```
$ ping google.com
```

To limit ping for specified number of icmp requests

```
$ping -c 4 google.com
```

traceroute:

traceroute command in Linux prints the route that a packet takes to reach the host.

```
$traceroute google.com
```

if traceroute not installed .

Install using **\$yum install traceroute -y**

netstat

Linux netstat command refers to the network statistics.

```
$netstat
```

dig

Linux dig command stands for Domain Information Groper. This command is used in DNS lookup to query the DNS name server. It is also used to troubleshoot DNS related issues.

```
$dig google.com
```

If dig not found install dig using **\$yum install bind-utils -y**

nslookup

Linux nslookup is also a command used for DNS related queries. It is the older version of dig.

```
$nslookup google.com
```

host

Linux host command displays the domain name for a given IP address and IP address for a given hostname. It is also used to fetch DNS lookup for DNS related query.

```
$host google.com
```

route

Linux route command displays and manipulates the routing table existing for your system.

A router is basically used to find the best way to send the packets across to a destination.

\$route

```
[root@localhost bin]# route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default 10.0.2.2 0.0.0.0 UG 100 0 0 enp0s3
10.0.2.0 0.0.0.0 255.255.255.0 U 100 0 0 enp0s3
```

hostname

Linux hostname is the simple command used to view and set the hostname of a system.

Syntax:

\$hostname

```
[root@jenkinsserver vagrant]# hostname
jenkinsserver
```

To change hostname

Syntax:

\$hostname <requiredname>

Example:

\$hostname jenkinsserver

To make changes exit and relogin

```
[root@localhost bin]# hostname jenkinsserver
[root@localhost bin]# exit
exit
[vagrant@localhost ~]$ sudo su
[root@jenkinsserver vagrant]#
```

Process Management

Ps -> To list running processes

\$ps

Ps -u : List running processes with consumed cpu and memory details

\$ps -u

Stopping processes:

\$kill pid

Example:

\$kill 6738

top Command

The top command is a very useful tool for quickly showing processes sorted by various criteria.

\$top

\$sleep : Sleep command pauses for a specified time.

\$sleep 60 (stopping for 1 min = 60 seconds)

User and Group management commands

User Management commands:

<i>Purpose</i>	<i>Command</i>
<ul style="list-style-type: none">• Adding a user	useradd dan
<ul style="list-style-type: none">• Assigning password to user	passwd dan
<ul style="list-style-type: none">• Changing home directory for user	useradd dan -d /home/new
<ul style="list-style-type: none">• Setting expiry for user	useradd dan -e 2017-11-25
<ul style="list-style-type: none">• Adding inactive period before expiry	useradd dan -f 2
<ul style="list-style-type: none">• Changing default shell	useradd dan -s /bin/sh
<ul style="list-style-type: none">• Removing user	userdel dan
<ul style="list-style-type: none">• Removing user with home directory	userdel -r dan
<ul style="list-style-type: none">• Setting expiry for user	usermod -e 2017-11-25 dan
<ul style="list-style-type: none">• Changing home directory	usermod -d /home/new dan
<ul style="list-style-type: none">• Changing default shell	usermod -s /bin/sh dan
<ul style="list-style-type: none">• Locking an account	usermod -L dan
<ul style="list-style-type: none">• Unlocking a locked account	usermod -u dan

Group management Commands:

Following are the commands for managing groups

- Adding a group `groupadd linuxgroup`
- Adding user to group `usermod -aG linuxgroup dan`
- Changing owner & group of a file `chown dan:linuxgroup newfile.txt`
- Changing only owner of a file `chown dan: newfile.txt`
- Changing only group of a file `chown :linuxgroup newfile.txt`
- Deleting a group `groupdel linuxgroup`

Command to Set the Password for the Group: Below command is used to set the password of the group. After executing the command we have to enter the new password which we want to assign to the group. The password has to be given twice for confirmation purposes.

```
gpasswd group_name
```

Command to Display the Group Password File: The below command gives us the password file as output. The file is present in a form such that no information about the file is open for the viewers. Instead of this try: “cat /etc/group” to get more information about the groups.

```
cat /etc/gshadow
```

Command to Add a User to an Existing Group: Below command is used to add a user to an existing group. The users which may be present in any primary or secondary group will exit the other groups and will become the part of this group.

```
usermod -G group_name username
```

Command to Add User to Group Without Removing From Existing Groups: This command is used to add a user to a new group while preventing him from getting removed from his existing groups.

```
usermod -aG *group_name *username
```

Command to Add Multiple Users to a Group at once:

```
gpasswd -M *username1, *username2, *username3 ..., *username  
*group_name
```

Command to Delete a User From a Group: Below command is used to delete a user from a group. The user is then removed from the group though it is still a valid user in the system but it is no more a part of the group. The user remains part of the groups which it was in and if it was part of no other group then it will be part of its primary group.

```
gpasswd -d *username1 *group_name
```


Package Management

Distributions: Centos / Redhat / Fedora

Using rpm tool:

1. Installing Package:

```
# rpm -ivh MySQL-client-3.23.57-1.i386.rpm
```

```
Preparing... ##### [100%]
```

```
1:MySQL-client #####  
[100%]
```

rpm command and options

- i : install a package

- v : verbose

- h : print hash marks as the package archive is unpacked.

2. Query all the RPM Packages using rpm -qa

You can use rpm command to query all the packages installed in your system.

```
# rpm -qa
```

```
cdrecord-2.01-10.7.el5
```

```
bluez-libs-3.7-1.1
```

```
setarch-2.0-1.1
```

```
.
```

```
.
```

- q query operation

- a queries all installed packages

3. Query a Particular RPM Package using rpm -q

The above example lists all currently installed package. After installation of a package to check the installation, you can query a particular package and verify as shown below.

```
# rpm -q MySQL-client
```

```
MySQL-client-3.23.57-1
```

```
# rpm -q MySQL
```

```
package MySQL is not installed
```

4. Uninstalling a RPM Package using rpm -e

To remove an installed rpm package using -e as shown below. After uninstallation, you can query using rpm -qa and verify the uninstallation.

```
# rpm -ev MySQL-client
```

rpm

Task	Command
Install a .rpm file	rpm -i package-file-name.rpm
Remove a .rpm file	rpm --erase package-name(s)
Listing Installed Files	rpm -qa
Check if a given package is installed	rpm --query package-name(s)

Using yum tool: High level

yum list installed :

Lists packages installed via YUM

\$yum list installed

yum list all :

Lists all currently available packages

\$yum list all

yum group list :

Lists grouped packages

\$yum group list

yum info :

Provides detailed information about a package

\$yum info git

yum search :

Searches package descriptions and names

\$yum search git

localinstall : Installs a local rpm package

yum remove :

Removes the installed package

\$yum remove git

yum clean all :

Cleans /var/cache/yum to free disk-space

\$yum clean all

man yum :

Like all linux commands, the help file

yum

Task	Command
Installs the package(s) with dependency	yum install package-name(s)
Remove the package(s) but not dependency	yum erase package-name(s)
Update Package List	yum update
search Packages	yum search search-pattern
get info about a package	yum info package-name(s)

ubuntu:

Installing package: `$apt-get install git`

Removing package: `$apt-get remove git`

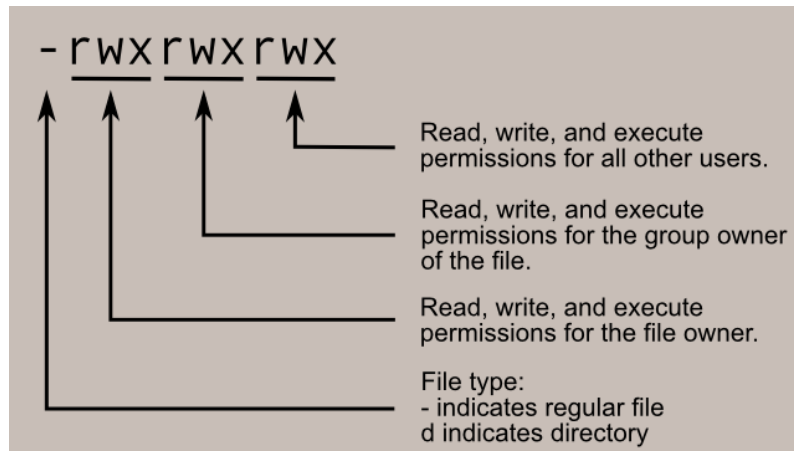
Get info about package : `$apt show telnet`

Update package list: `$apt-get update`

apt

Task	Command
Installs the package(s) with dependency	<code>apt-get install package-name(s)</code>
Remove the package(s) but not dependency	<code>apt-get remove package-name(s)</code>
Remove unused dependency	<code>apt-get autoremove</code>
Update Package List	<code>apt-get update</code>
search Packages	<code>apt search search_string</code>
get info about a package	<code>apt show package</code>

Permissions



$\text{rwx} = 111 \text{ in binary} = 7$

$\text{rw-} = 110 \text{ in binary} = 6$

$\text{r-x} = 101 \text{ in binary} = 5$

$\text{r--} = 100 \text{ in binary} = 4$

Value	Meaning
777	(rwxrwxrwx) No restrictions on permissions. Anybody may do anything. Generally not a desirable setting.
755	(rwxr-xr-x) The file's owner may read, write, and execute the file. All others may read and execute the file. This setting is common for programs that are used by all users.
700	(rwx-----) The file's owner may read, write, and execute the file. Nobody else has any rights. This setting is useful for programs that only the owner may use and must be kept private from others.
666	(rw-rw-rw-) All users may read and write the file.
644	(rw-r--r--) The owner may read and write a file, while all others may only read the file. A common setting for data files that everybody may read, but only the owner may change.
600	(rw-----) The owner may read and write a file. All others have no rights. A common setting for data files that the owner wants to keep private.

Important Commands:

chmod - modify file access rights

```
$ chmod 600 some_file
```

Giving read and write permissions to file

chown - change file or directory ownership

We can change the owner of a file by using the chown command. Here's an example: Suppose we wanted to change the owner of some_file from "me" to "you". We could:

```
$ chown username filename
```

chgrp - change a file's group ownership

The group ownership of a file or directory may be changed with chgrp. This command is used like this:

```
$ chgrp group_name file_name
```