

Linux Basic & Admin

Linux Admin Commands

Basics:

- # -- root user
- \$ -- normal user
- #whoami – gives your login name
- #who
- #date – gives date in UTC format
- #pwd – print working dir
- #cal – displays current month with highlighting on today's date
- #cal 2017 – displays all months in 2017
- #cal may 2017 – displays 2017 may month
- #clear – for clearing console
- #hostname – to get hostname (short name)
- #hostname -f – to get fully hostname

- #hostname -i – for displaying ip address of instance

To get Memory details:

- #free -m (RAM)
- #df (HDD)
- #df -h (human readable format)
- #df -m (file system in MB)
- #du – directory usage

Process Management

- #ps - shows all the process details
- #ps aux – lists all the process of all users
- #kill <process id> - to kill a process which is running
- #pkill <process name> - to kill a process by name
 - #pkill java
 - #pkill tomcat

File Management:

- `#touch <file name>` – to create a file ex : `#touch demo.txt`
- `#ls` – to list all files and dir
- `#vi demo.txt` – to open a file
- Press `i` – for insert mode to enter data
- `:wq!` – write and quite (close)
- `#cat <filename>` - to display content of file
- `#ls -i` – to list inode values of all files
- `#ls -r` – to list all files in reverse order
- `#ls -t` – to list by time of modification
- `#ls -s` – to list by sizes
- `#cat <filename> > sample` – to copy data to sample
- `#cp <filename> sample` (cp will ask if file content is getting override)
- `#cat <filename> >> sample` – to append data in sample
- `#cat sample` – to show content in sample

- #ls -a – to list all hidden files
- #rm <filename>
- #ls -l – for displaying long list(all info about each file)

-	rwX	rwX	rwX	1	root	root	10	Dec 12	02:12	<file name>
file	Owner	group	other	no.	owner	group	momory	date & time		file name
type	permi	permi	permi	links	(user)					

File Type

- -> Regular file(text, doc, image)
d -> directory
l -> link file
s -> socket file
c -> char. special file
b -> Block special file

Permissions

r -> read permission
w -> write permission
e -> execute permission
- -> no permission

- `#touch file{1..5}` – for creating group of files at a time

To manage Directories

- `#mkdir <dir name>` -> creating directory
- `#mkdir <dir name> <dir name>` -> creating two directory at a time
- `#mkdir <dir name>{1..5}` -> creating five directory at a time
- `#cp demo.txt <dir>/` -> to copy file from present to <dir>
- `#cd <dir name>` -> for changing directory
- `#cd ..` -> to move to parent directory
- `#cd /` -> to move to root directory
- `#cd ~` -> to move to home directory
- `#cd` -> to move to home directory
- `#cd -` -> to move to previous working directory
- `#rmdir <dir>` -> to delete empty directory (if you try to delete directory which is not empty, will throw error)
- `#rm -rf <dir>` -> to delete empty/non-empty directory
- `#rm -rf *` -> to delete all sub directories

Soft Link and Hard Link

Soft Link : “Shortcut” (different inode values)

```
#ln -s <filename> <slinkname>
```

```
#ls
```

```
#ls -i -> for listing different inode values
```

Hard Link : “replica” (maintains same inode values)

```
#ln <file name> <hlinkname>
```

```
#ls -is
```

```
#rm <filename>
```

```
#cat <slinkname> -> can't access data
```

```
#cat <hlinkname> -> can access data
```

File Security

- **chmod**

different file permissions

read(r) – 4

r & w – 4 + 2 = 6

write(w) – 2

r & x – 4 + 1 = 5

execute(x) – 1

r,w & x – 4 + 2 + 1 = 7

no permi – 0

syntax-1

#chmod [permission] <file name>

Ex:

- #chmod 421 <file name>
- #chmod 536 <file name>
- #chmod 777 <file name>
- #chmod 000 <file name>

Syntax – 2

- `#chmod [who][+/-/=[permi] <filename>`

who

u – user

g – group

o – others

+ - grant permission

- - revoke permission

= - grant a specific permission & revoke all other permissions

- `#chmod u+r demo.txt` -- giving read permission to demo.txt
- `#chmod g=r demo.txt` -- granting read permission and revoking all other permissions to demo.txt

User and Group Management

- `#sudo su -`
- `#sudo -i` -- to switch to root user
- `#cat /etc/passwd` -- to list all users
- `#cat /etc/group` -- to list all groups
- `#groupadd <group name>` -- to add a group
- `#groupdel <group name>` -- to delete a group
- `#useradd <user name>` -- to add a user `#useradd raja`
- `#passwd <user name>` -- to set a password `#passwd raja`
- `#userdel <user name>` -- to delete a user
- `#groups <user name>` ----- shows `<user name> : <group name>`
- `#groupmod -n <new name> <old name>` -- to change group name
- `#usermod -g <group name> <user name>` -- to modify primary group for a user
- `#members <group name>`
- `Id <user name>`

To change ownership

#chown <username> <filename> -- to change owner

#chgrp <group name> <file name> -- to change group

#chown <username>:<groupname> <filename> -- to change owner & grp

File Compression & Extraction

File Compression methods		100MB
GunZip	----> ¼ size	25MB
BunZip	----> 1/8 size	12 MB

File data : char, numbers, alphanumeric, binary, raw, duplicate

For Compression

```
#tar -cf <name>.tar <name>.txt (tar)
#tar -czf <name>.tar.gz <name> (gunzip)
#tar -cjf <name>.tar.bz2 <name>
```

For Extraction

```
#tar -xf <name>.tar
#tar -xzf <name>.tar.gz
#tar -xjf <name>.tar.bz2
```

HEAD

Syntax:

#head [option] <file name>

Ex:

#head <file name> ----> list first 10 lines

#head -n 5 <file name> ----> list first 5 lines

TAIL

Syntax:

#tail [option] <filename>

Ex:

#tail <file name> ----> list last 10 lines

#tail -n 3 <file name> ----> list last 3 lines

#head -n 4 <file name> | tail -n 1 ----> list 4th line

#head -n 4 <file name> | tail -n 2 ----> list 4 & 5 line

GREP

Syntax:

```
#grep [option] <string> <file name>
```

Ex:

```
#grep -e Hello demo.txt
```

```
#grep -e Hello -e Devops demo.txt
```

```
#grep -i Hello demo.txt -----> ignore cases
```

```
#grep -l Hello *.txt -----> list all files
```

```
#grep -v Hello demo.txt ----> other lines
```

PASTE

Syntax:

```
#paste <file1> <file2>
```

Ex:

```
#paste file1 file2
```

```
#paste file1 file2 > file3
```

```
#paste -d '$' file1 file2
```

```
#echo '<content>' &>><file name> ----- to append content at  
last
```

```
#cat <file name 1> >> <file name 2> ----> copies data from file1 and appends data  
into file2
```