Linux Administration

Linux Basic Commands :

- **ls :**
  - Used to lists directory contents of files and directories.
  - # ls -l (show long listing)
  - # ls -a (show hidden files)
- **cd :**
  - used to changing one directory to another.
  - # cd /var/log/
- **pwd :**
  - Used to display the present working directory.
  - # pwd
- **cp :**
  - Used to copy file/directory from source to destination.
  - # cp source_path destination_path
- **mv :**
  - Used to move file/directory from source to destination.
  - mv source_path destination_path
- **rm :**
  - Used to delete a file or directory.
  - # rm -rf file_name
- **rmdir :**
  - Used to delete an empty directory.
  - # rmdir dir_name

- **mkdir :**
    - Used to create a directory.
    - # mkdir dir_name
- **cat :**
    - cat command allows us to view contain of file, concatenate files and redirect output in terminal or files.
    - # cat file_name (to view the file)
    - # cat file_1 > file_2
- **touch :**
    - Used to create a new empty file.
    - # touch file_name
- **chmod :**
    - Used to Change access permissions.
    - # chmod 644 file_name
- **chown :**
    - Used to Change file owner and group.
    - # chown owner_name file_name
    - # chown owner_name:group_name file_name
- **chgrp :**
    - Used to Change group ownership.
    - # chgrp user_name file_name
- **date :**
    - Used to display date and time.
    - # date
- **df :**
    - Used to display free disk space.

- # df –h

- **du :**
  - Used to the amount of disk space used by the specified files and for each subdirectory.
  - # du -h
- **echo :**
  - Used to display messages on the screen.
  - # echo “message”
- **fdisk :**
  - Used to list the partition tables of the disk.
  - # fdisk -l
- **find :**
  - Used to search a folder hierarchy for filename(s) that meet a desired criteria: Name, Size, File Type etc.
  - # find / -name file_name
- **free :**
  - Used to display memory usage.
  - # free -m
- **grep :**
  - Used to search file(s) for lines that match a given pattern.
  - Grep -i word/pattern file_name.
- **useradd :**
  - Used to create a user.
  - # useradd user_name
- **passwd :**
  - Used to create/change password of the user.
  - # passwd user_name
- **userdel :**

- Used to delete a user.
- # userdel user_name

- **groupadd :**
  - Used to create a group.
  - # groupadd group_name
- **groupdel :**
  - Used to delete a group.
  - # groupdel group_name
- **groups :**
  - Used to check a user belongs which group.
  - # groups user_name
- **history :**
  - Used to display the executed commands history.
  - # History
- **hostname :**
  - Used to print and change the host name/system name.
  - # hostname
- **head :**
  - Used to display first part of the file.
  - # head file_name (display first 10 lines)
  - # head -5 file_name (display first 5 lines)
- **tail :**
  - Used to print last part of the file.
  - # tail file_name (display last 10 lines)
  - # tail -5 file_name (display last 5 lines)
- **more :**
  - Using more we can only toggle to the next page.
  - # more file_name

- **less :**
    - Using less we can toggle to next as well as previous page.

    # less file_name
- **id :**
    - Used to display id and group details of an user.
    - # id user_name
- **ifconfig :**
    - Used to display ip address and ethernet card details.
    - # ifconfig
- **kill :**
    - Used to kill a process.
    - # kill signal(-9 or -15) pid
- **lsof :**
    - Used to display all the opened ports.
    - # lsof
- **netstat :**
    - Used to display network status and listening ports.
    - # netstat -tulpn
- **ping :**
    - Used to check the network connection.
    - # ping host_ip
- **ps :**
    - Used to display and monitor the process status. Ps will display the static information.
    - # Ps aux
- **top :**
    - Used to monitor the performance of the system and process. It will display information dynamically.
    - # top

- **scp :**
    - Used to copy files securely to the remote machine.
    - # scp source_path dest_path
- **rsync :**
    - Used to synchronize a file to the remote machine.
    - # rsync -avz source_path dest_path
- **ssh :**
    - Used to login into a remote system securely.
    - # ssh _address
- **stat :**
    - Used to display status of the file.
    - # stat file_name
- **su :**
    - Used to switch from one user to another.
    - # su user_name
- **sudo :**
    - Sudo allows a permitted user to execute a command as the superuser.
    - #sudo command
- **tar :**
    - Used to create and extract an archive file.
    - # tar -cvf file_name.tar file1 file2 file3
    - # tar -xvf file_name.tar
- **uptime :**
    - Used to display from how long the machine is running,how many users logged in and the load average of the machine.
    - #uptime
- **vmstat :**
    - Used to display virtual memory statistics.

- # vmstat

- **w :**
  - It will display all currently logged in users and their status.
  - # w
- **who :**
  - Used to Print all user names currently logged in.
  - # who
- **whoami :**
  - It will print the currently logged in user name.
  - # whoami
- **man :**
  - Used to display all the details of a command with all the options.

  # man ps

1.1 VIM Editor :

- Vim file_name : To edit the existing file or to create a new file
- **i :** To insert text to the editor.
- **h :** Moves the cursor to the left.
- **l :** Moves the cursor to the right.
- **k :** Moves the cursor up.
- **j :** Moves the cursor down.
- **$:** Moves the cursor end of the current line.
- **0 (Zero) :** Moves the cursor beginning of the current line.
- **shift+g :** Moves cursor to the last line.
- **shift+h :** Moves cursor to the first line.
- **esc+wq :** Save and exit from the editor.
- **esc+wq! :** Save and quit forcefully.

- **esc+q :** Quit without saving the file.
- **esc+q! :** Quit without saving the file forcefully.
- **esc+u :** Undo the changes.
- **dd :** Delete the current line.
- **yy :** Copy the current line.
- **10yy :** Copy 10 lines.
- **d :** cut the selected / current line.
- **p :** Paste the copied lines.
- **Esc : 3 :** Directly go to the line number 3.

1.2 Connectivity Issue :(User not able to connect to the server/host)

1. Check the machine (Server) by pinging. Use the below command.

   # ping Server_IP

2. Check the user's age parameter. Use the below command.

   # chage –l user_name

3. Check the user having /bin/bash or not. Follow the below steps.

   # grepuser_name /etc/passwd

   User_name:x:502:502::/home/sudhir:/bin/bash

   - **To add a shell for an user :**

   # usermod –s /bin/bash user_name

4. Check the user is locked. If you find " ! " symbol before encrypted password then the account is locked.

   # grep user1 /etc/shadow

   user1:!!6ciJaxxxxxxxxxxxxxkEvF29ITpef0Sxxxxxxxxxxx6mRAHee4tZT0r11:16299:0:99999:7:::

- **To unlock the user use the below command:**

passwd –u user_name

usermod –U user_name

- **To Lock an user use the below command :**

usermod –L user_name

passwd –l user_name

5. Check the 22 port is listening or not. Use the below command.

sudonetstat –tulpn | grepssh

6. Check the VM is running or not. Login into AWS Console and check.

1.3 Performance Monotoring :

1. CPU Utilization : Load average (top and ps)1min 5min 15min

- check which process is using more cpu. (top or ps)

top

ps –aux

- check the number of processor and cores present in the VM.

lscpu

- check the zombie processes :

ps aux | grep -w defunct

- Kill the processes if not necessary :

#kill -15 pid (It will stop the process in a safe mode)

2**. Memory utilization : (free and top)**

- check which application using more memory.

free -h

top

- check the services

service service_name status

- To check all the available processes

service --status-all

- check the memory leakage and To resolve the same use the below command.

free –h

If you will found more memory in "buffers" and "cached" then you will get to know that there is memory leakage is going on. To clear the buffers and cached run the below command.

sync; echo "1" /proc/sys/vm/drop_caches

3. **Disk space : (df –h and du -h)**

- check the disk utilization.

  # df -h

- **To check how many disk are attached and where they are mounted use the below command.**

  # lsblk

4**. I/O operation :**

- check the io operation by (vmstat)

  #vmstat

- **To understand the output of "vmstat" follow the below notes :**

  Proc:
  ----
  r: How many processes are waiting for CPU time.
  b: Wait Queue - Process which are waiting for I/O (disk, network, user input,etc..)

  Memory:
  ----
  swpd: shows how many blocks are swapped out to disk (paged). Total Virtual memory usage.

  Note: you can see the swap area configured in server using "cat proc/swaps"

  free: Idle Memory
  buff: Memory used as buffers, like before/after I/O operations

  cache: Memory used as cache by the Operating System

  Swap:
  ----
  si: How many blocks per second the operating system is swapping in. i.e Memory swapped in from the disk (Read from swap area to Memory)
  so: How many blocks per second the operating system is swaped Out. i.e Memory swapped to the disk (Written to swap area and cleared from Memory)

  In Ideal condition, We like to see si and so at 0 most of the time, and we definitely don't like to see more than 10 blocks per second.

  IO:

------
bi: Blocks received from block device - Read (like a hard disk)
bo: Blocks sent to a block device - Write

System:
----
in: The number of interrupts per second, including the clock.
cs: The number of context switches per second.

CPU:
----
us: percentage of cpu used for running non-kernel code. (user time, including nice time)
sy: percentage of cpu used for running kernel code. (system time - network, IO interrupts, etc)
id: cpu idle time in percentage.
wa: percentage of time spent by cpu for waiting to IO.

If you used to monitor this data, you can understand how is your server doing during peak usage times.

Note: the memory, swap, and I/O statistics are in blocks, not in bytes. In Linux, blocks are usually 1,024 bytes (1 KB).

5. **Network packet drop:**

#check network packet drop.

- **Capture Packets from source IP**

tcpdump -i eth0 src 192.168.0.2

tcpdump -i eth0 dst 50.116.66.139

1.1 User & Group Management :

List of important files for managing user and group in Linux.

/etc/passwd (Contains all user's information)

/etc/shadow (Contains encrypted password of all users)

/etc/group (Contains group names and group ids)

/etc/login.defs (Contains all login parameters of an user)

1. **To create a user :**

adduseruser_name

2. **To set and change a password :**

passwduser_name

3. **To create a group :**

   # groupaddgroup_name

4. **To delete a user :**

   # userdel –r user_name

5. **To delete a group :**

   # groupdelgroup_name

6. **To check which user belongs to which group :**

   # groups user_name

7. **To check user’s parameters :**

   # id user_name

8. **To add a user to another group :**

   # usermod -a –G group_nameuser_name

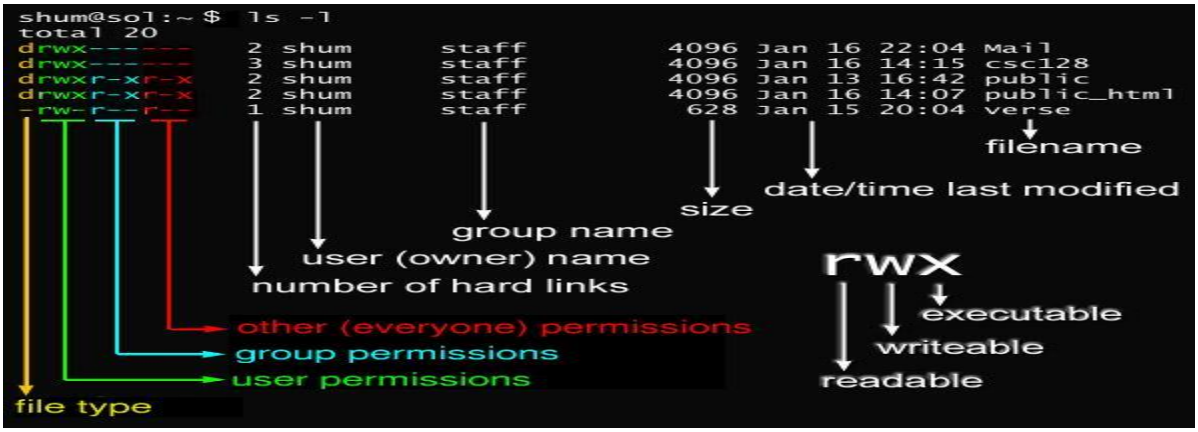
9. **To Change file owner and group :**

   # chown owner_name:group_namefile_name

   # chown owner_namefile_name

1.2 File and permission Management :

- **File permission types and Levels :**



1. **Types of permissions :**

   a. Read (r) numeric value (4)

   b. Write (w) numeric value (2)

   c. Execute (x) numeric value (1)

2. **Levels of permission :**

   a. Owner/ User

   b. Group

   c. Other

3. **Commands need to set/change the permissions :**

   a. # chmod 644 file_name (read and write for owner, read for group and others)

b. # Setfacl -m u:user_name:rw file_name (used to set a special permission for a group or user)

4. Special permissions in linux :

a. Sticky bit (t)

If the directory has the sticky bit set, a file can be deleted only by the owner of the file, the owner of the directory, or by root. This special permission prevents a user from deleting other users' files from public directories such as /tmp:

e.g. drwxrwxrw**t** 7 root sys 400 Sep 3 13:37 tmp

b. Suid (s) numeric value (4)

This permission allow an user to run an executable file as the owner of that file.

e.g. -r-**s**r-sr-x 3 root sys 104580 Sep 16 12:02 /usr/bin/passwd
chmod u+s file_name

c. Sgid (s) numeric value (2)

This permission is same as setuid and it will effect to the users of an entire group.

e.g. -r-x--**s**--x 1 root mail 63628 Sep 16 12:01 /usr/bin/mail
chmod g+s file_name

- **To check the file permissions :**

  # ls –l file_name

- **To change file permission :**

  #chmod 644file_name

- **To set ACL permission for a user :**

#setfacl -m u:user_name:permissions (rwx) file_name

e.g : # setfacl -m u:sudhir:r-x file1 (Here “Sudhir” is the user and having “read” and “execute” permission on “file1” but don’t have “write permission”)

- To check the status of a file :

stat file_name

1.3 Bulk User ID Creation :

newuser reads following syntax.

<Username>:<Password>:<UID>:<GID>:<User Info>:<Home Dir>:<Default Shell>

Username: User login name

Password: User password

UID: User Identifier

GID: Group Identifier for user’s primary group

User Info: User information like full name, contact information etc.

Home Dir: User’s home directory

Default Shell: User’s default shell

So, now let’s take an example to create multiple users with help of newusers command.

Steps:-

[root@CentOS6 ~]# vim users.txt

user1:user1@123:601:601:User 1:/home/user1:/bin/bash

user2:user2@123:602:602:User 2:/home/user2:/bin/bash

user3:user3@123:603:603:User 3:/home/user3:/bin/bash

user4:user4@123:604:604:User 4:/home/user4:/bin/bash

user5:user5@123:605:605:User 5:/home/user5:/bin/bash

save and exit.

You can verify entry by cat command.

Now, let’s execute above file.

[root@CentOS6 ~]# newusers users.txt