**23** : telnet port number

**22** : ssh/scp

**hostname** : To find hostname

**uname -r** : check kernel version

**uname -m or arch** : check given os is 32 bit or 64bit

**cat /etc/redhat-release** : check linux version

**ping ipaddress**or**hostname** : to check connectivity between two boxes **telnet hostname portnumber** : to check port is opened or not in remote box **who** : to find users logged in to the box

**who -q** : count number of users logged in

**last reboot** – to check reboot history or **who -b ifconfig** : Check ipaddress

**ifconfg -a** : check nic cards are enabled or not

**man command** : It’s a help document how to use a command and its switches

**&** : give & at the end of the command to run a process in the back ground

**fg**: bring a process to foreground

**cat>.f1.txt** : create a hidden file f1.txt

**cat /dev/null>sysout.txt** : nullify a log file

**ls -a** : To find hidden files

**ls -l** : To find long list format

**ls -lrt** : To find recent modified files in reverse order

**ls -lt** : Display the files based on timestamp

**df -h**: free disk space

**du -h** : lists all files and their sizes in present directory and total directory size, to find disk a usage.

**find -type f -exec du -sh {} + | sort -rh | head -n 5 :** To find top 5 files based on the size

**find . -user rama -type f -exec du -m {} \;|awk '{ s = s+$1 } END { print "Total used: ",s }'**: disk usage based on user or **du -sh ~username** or **du ~username**

**free -m** : to find free memory or ram

**chmod numericalnumber filename** : read 4 write 2 exceute 1 : File permissions

**chmod -R 755 directory** : file permissions change to entire directory

**useradd username** : to add a user

**passwd username** : change password

**chownnewownername filename** : change Ownership to a file **chown -R newownername dirname** : change ownership to directory **chown :newgroupname filename** : change groupname to a file

**chown -R :newgroupname dirname** : change groupname to a directory

**sudo command** : To execute with special user or admin user

**su username** : switch user to other user

**ssh[username@192.168.42.130](mailto:username@192.168.42.130)** : switch to user account which is in remote box

**mv oldfilenewfile** : rename file or directory

**rm filename** : To delete a file

**rm -r dirname**: To delete a directory

**tail -6 filename** : To display the last 6 lines in a file

**tail -f filename** : To read runtime logs

**head -6 filename** : To display top 6 lines in a file

**find /tmp -type f -name "\*.txt"** : find which ends with .txt

**find /tmp -type f -name "krishna\*" -print -exec rm {} \;** find and delte files which starts with krishna

**find /tmp -mtime +10** : search files modified older than 10 days

**find /tmp -mtime +10 -exec rm -r {} \;** : search files and dir’s modified older than 10 days and delete

**find /tmp -type d -mtime +10 -exec rmdir {} \;** find and delete dir’s older than 10 day

**find /tmp -type f -name "\*.txt" -mtime +50** : find which ends with .txt older than 50 day

**find /tmp -type f -name "\*.txt" -mtime +50 -exec rm {} \;** find ends with .txt older than 50 days and delete **find /tmp -type f -name "\*.txt" -mtime +50 -size +4M -exec rm {} \;** find ends with .txt older than 50 days , size more than 4mb and delete

**find /tmp -type f -empty** : search empty files

**find /tmp -type d -empty** : search empty directories

**find /tmp -type f -empty -exec rm {} \;** search empty files and delete **find /tmp -user krishna** : search the files owned by a particular user **grep “krishna” f1.txt** : search a string krishna in file f1.txt

**grep -r “krishna” /tmp** : search a string in multiple file

**find . -type f -print -exec sed -i 's/km\_test/km\_multi/g' {} \;** find and replace in multiple files

**sed ‘s/oldstring//g’ input.file>output.file** delete a string in a file

**sed -i ‘s/oldstring/newstring/g’ f1.txt** : find and replace a string in f1.txt

**sed -i ‘s/oldstring/newstring/g’ f1.txt f2.txt :** find and replace a string in f1.txt and f2.txt

**sftp** [**remoteuser@192.168.42.129**](mailto:remoteuser@192.168.42.129) **(Remote IP)** : connect to sftp **sftp> put localfile\_f1.txt** : upload from local to remote **sftp>mput \*.txt** : upload multiple files from local to remote **sftp>get f1.txt** : download from remote to local

**sftp>mget \*.txt** : download multiple files from remote to local host

**runlevel** : check runlevel

1. : halt the system
2. : single user mode
3. : local multiuser with networking but without network service (like NFS)
4. : full multiuser with networking
5. : not used
6. : full multiuser with networking and X windows (GUI)
7. : Reboot

**wc -l filename** – display number of lines in a file

**nl filename** or **set number** in vi : to set line numbers

**o (small o)** Insert blank line next line to cursor

**O (capital O)** insert blank line before the current line

**ps -ef | grep server1** : to check server1 is running or not and get info about that server

**ps -ef | grep http** : check webserver is running or not

**ps -ef | grep java** : to check all running java processes are running or not and get info about that processes

**ps -u username** : command to check process running with a particular user

**ps -auxf|sort -nr -k3|head -15** : command to find top 15 processes which consumes high cpu usage

**ps -aux --sort -rss | head -15** : command to find top 15 processes which consumes high memory usage

**pgrep java | wc -l** : to count number of java process that are currently running **sshremotehost 'pgrep java' :** command to check running process on remote host **ptree (in solaris) pstree (in linux)** : check how each process is related to other process **pfiles** : In solaris to see list of open files with pid, usage: pfilespid

**lsof -i:portnumber** : to find pid with a port number

**netstat -antp | greppid** or**netstat -nlp | greppid** : To find port numbers with pid or pid with port number

**netstat -plnt** : to check listening ports

**diff vscmp**: diff is text comparison line by line but cmp is the byte comparison

# export JAVA\_HOME=/usr/java/jdk1.6/bin/java export PATH=$PATH:/usr/java/jdk1.6/bin

**awk**: It will read the file and we can print specific rows and columns for easy understanding purpose

**sar**: monitor cpu usage

**pkill java** : pkill will kill all the processes matching to that string

**pkill -u username** : kill all process belongs to a particular user

**killall**: If too many instances for a particular process or too many child threads are there at that time we will use killall, usage: killall -15 processname

**kill -3 pid**:To generate thread dumps

**kill -9 pid**: It’s a immediate kill / forcible kill

**kill -15 pid**: It will wait until all the child process are killed the parent process will be killed

**rpm -ivhrpmfilename**: install rpm file

**tar -tvf filename.tar** Display the tar files without extraction

**tar -cvf filename.tar location where tar to be created** Create tar file

**tar -xvf filename.tar** Extract tar file

**tar -xzvf filename.tar** Extract tar.gz file

**zip filename.zip file1 file2 folder1 folder2** : zip the files

**zipinfo -1 filename.zip** : display zip file content

**unzip filename.zip** : unzip a file

**scp f1.tx username@remotehostname:/some/remotedir** : copy files from local box to remote box **scpusrename@remotehostname:f1.txt /some/localdir** : copy files from remote box to local box **scp -r dir1 username@remotehostname:/some/remote/dir2** : copy dir from local to remote

**scp username@remotehost1:/file/to/send username@remotehost2:/where/to/put** Copy from remote to remote

**nohup command &** run the process even after logout from the shell prompt

**find vs grep** : find is used to search files and grep is used to search pattern (string) in files

**echo $?** To check last command executed successfully or not. If the output is 0 – success, non zero - failed

**cat /proc/mounts** check all mounted devices

**mount -t type device destination\_dir** : Mount a device or filesystem

**unmount /directory** : unmount syntax

**lsof -p pid** : to check list of opened file with a specific process

**lsof -u username** : to check list of opened files with a specific user

**grep "23/Jan/2013:06" example.com | cut -d[ -f2 | cut -d] -f1 | awk -F: '{print $2":"$3}' | sort - nk1 -nk2 | uniq -c | awk '{ if ($1 > 10) print $0}' –** Command to check number of requests

<http://www.inmotionhosting.com/support/website/server-usage/view-level-of-traffic-with-apache-access-log>

# crontab syntax :

Minute hour Day of Month Month day of Week command

0-59 0-23 1-31 1-12 or jan-dec 0-6 or sun – sat command

**30 0 \* \* \* find /tmp -type -empty -delete** : delete empty files under /tmpeveryday at 12:30 am

**\*/5 \* \* \* \* /home/krishna/abc.sh** : abc.sh will execute for every 5 min

**0 \*/5 \* \* \* /home/krishna/abc.sh** : abc.sh will execute for every 5 hours

**0 17 \* \* \* /home/krishna/abc.sh :** abc.sh will execute everyday at 5 pm [**http://www.crontab-generator.org/**](http://www.crontab-generator.org/) - It’s a nice application for cron tab syntax generation **lscpu**: diplay hardware info like 32 bit/64 bit, cpu, model, cpu family

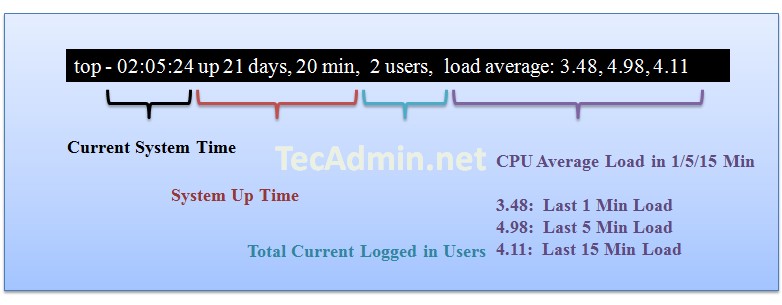
**last** : to find out users login and logout time

**more** vs **less**: more is only forward direction, less is both forward and backward direction

**top:** is used find cpu utilization, memory utilization of the processes Below is the sample output and explained in detail

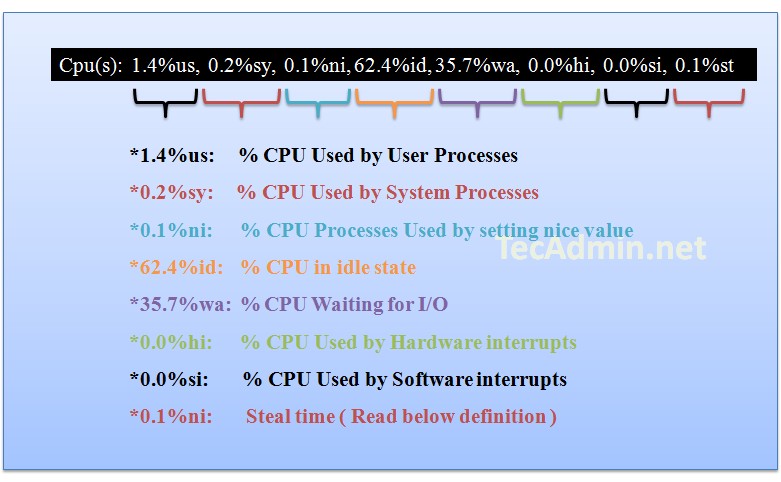
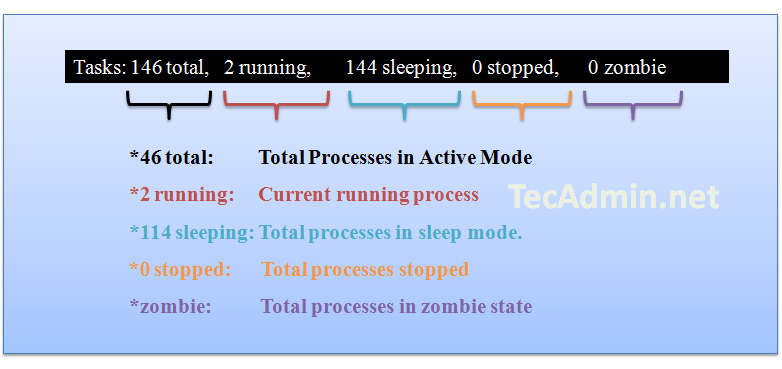
Result Row #1:

Row 1 results shows about server up time from last reboot, currently logged in users and cpu load on server. The same output you can find using linuxuptime command.



*ResultRow 2 shows the number of process running on server and there state.*

Zombie process or defunct process is a process that has completed execution but still has an entry in the

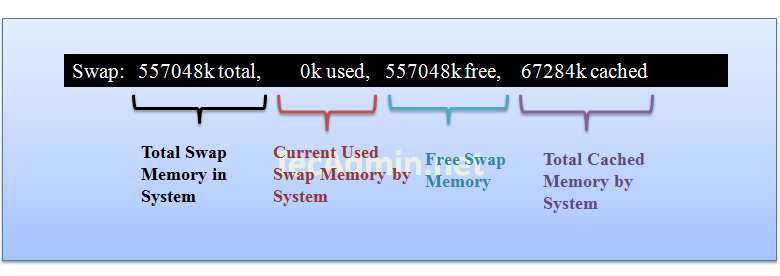
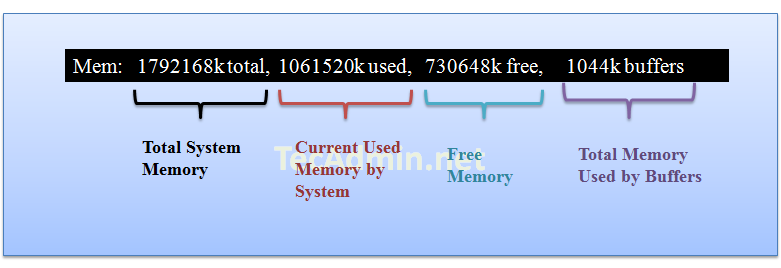


process table. This entry is still needed to allow the parent process to read its child’s exit status.

Row three shows the cpu utilization status on server, you can find here how much cpu is free and how much is utilizing by system.

Steal time is the time that a virtual CPU waits for a real CPU while the hypervisor is servicing another virtual processor.

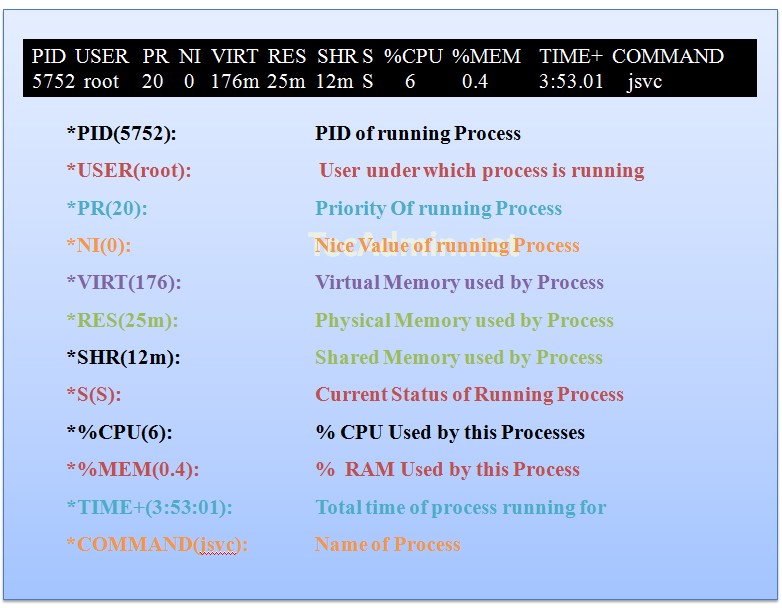
Row 4 shows the memory utilization on server, you can find here how much memory is used, the same results you can find using free command.



Row 5 shows the swap memory utilization on server, you can find here how much swap is being used,



*Result Row #6 ( Running Processes ):In this steps you will see all running process on servers and additional details about them like below.*



If you want to sort output on basis of any fields, you can use following key combinations.Press "SHIFT + F" and Select your choice below and press ENTER.

**Zombie process** or **defunct process:** is a process that has completed execution but still has an entry in the process table. This entry is still needed to allow the parent process to read its child’s exit status

***Swap memory or Swap space***: swap space in Linux is used when the amount of physical memory (RAM) is full. If the system needs more memory resources and the RAM is full, inactive pages in memory are moved to the swap space. While swap space can help machines with a small amount of RAM, it should not be considered a replacement for more RAM. Swap space is located on hard drives, which have a slower access time than physical memory.

**Sticky Bit** is mainly used on folders in order to avoid deletion of a folder and its content by other users though they are having write permissions on the folder contents. If Sticky bit is enabled on a folder, the folder contents are deleted by only owner who created them and the root user. No one else can delete other users data in this folder(Where sticky bit is set). This is a security measure to avoid deletion of critical folders and their content(sub-folders and files), though other users have full permissions.

Ex: chmod 1755 /tmp/test123

Here in 1755, 1 indicates Sticky Bit set, 7 for full permissions for owner, 5 for read and execute permissions for group, read and execute permissions for others.

**Symbolic link** or **soft link** or**symlink**: A symbolic link is a special kind of file that points to another file or reference, much like a shortcut in windows. It is used to link between files

Usage: ln -s <file1><file2>

**tar -xzvf httpd-2.2.31.tar.gz -C /root/Desktop/softwares/apache/apache\_ext**: extract to different dir **find . -type f -print | xargs du -sm | sort -nr | head -5**: Command to find the top 5 disk usage files **find /tmp/abc ! -name ‘f1.py’ -type f -exec rm -f {} +**: delete all files in /tmp/abc except f1.py

**find . -type f -exec grep -H 'rama' {} \;**Search a string rama in multiple filesby using find command

**sed –n ‘200,400p’ f1.txt** : To display lines from 200-400

**df -Ph /tmp | tail -1 | awk '{print $4}'** : Find disk space allocated for a director

**cat /proc/net/dev\_mcast**:to check multicast is supporting or not

<http://vim.wikia.com/wiki/Ranges> - vi editor

1. What is port tunnelling
2. Explain netstat –an output
3. How to move a cursor to end of the file in Vi editor.
4. How to move a cursor to end of the line in Vi editor.
5. How to move a cursor to start of the file in Vi editor.
6. How to move a cursor to start of the line in Vi editor.
7. What is the difference between 32 bit OS and 64 bit OS
8. What is the difference between find and grep commands.
9. What is the use of UMASK and what is the default UMASK value<https://docs.oracle.com/cd/E19683-01/817-1658/userconcept-95347/index.html>
10. What is the difference between softlink and hardlink
11. In netstat –an what is the difference between listening and established
12. What is the use of umask
13. Command to delete the lines from 50-70 in a particular file – 50,70d
14. How to change timestamp of a file
15. How to copy the lines from 30 to 40 in a particular file
16. What is the use of nslookup, syntax and what is the output
17. What is the difference between grep, egrep, fgrep and pgrep
18. What is the difference between grep, egrep, fgrep and pgrep
19. How to delete a line in VI
20. How to save and quit in VI

:wq

Rsync

How to delete ctrl m character

Ulimit –a will be used while installation

# Networking Commands

curl -v telnet://192.168.48.128:9990

<https://gist.github.com/Khoulaiz/41b387883a208d6e914b#file-gistfile1-md> apache concurrent users

# netstat -nt | grep :80 | wc -l

# netstat -plan|grep :80 | wc -l

# netstat -an | grep 'EST' | wc -l

netstat -an |grep:80|wc-l

crontab -e

\*/2 \* \* \* \* /opt/EAP-7.0.0/bin/standalone.sh

awk

awk '{print $0}' test\_file.txt -- print all the columns awk '{print $2}' test\_file.txt -- print second column awk '{print $NF}' test\_file.txt -- print last column

awk '{print $2 $4}' test\_file.txt - print 2nd and 4th column

awk '{print $2, "\t", $4}' test\_file.txt - Print 2nd and 4th column with tab seperator awk '{$3=""; print $0}' test\_file.txt - print all the columns but not 3

awk '{$3=$4=""; print $0}' test\_file.txt - print all the columns but not 3 and 4

You can tell awk how fields are separated using the -F option on the command line. awk 'NR==3 {print $2}' test\_file.txt - prints 3rd row 2nd column

awk 'NR==4 {$3=$4=""; print $0}' test\_file.txt awk 'NR==1 {print $4} || NR==2 {print $5}' file awk 'NR==5 || NR==9' "file"