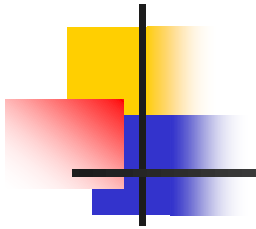
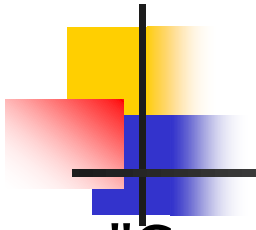
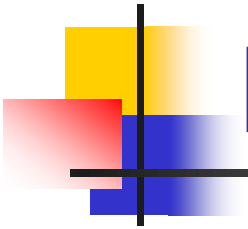


JOB MANAGEMENT (CRON)



- "Cron" is a time-based job scheduler in Unix-like operating systems (Linux, FreeBSD, Mac,solaris OS etc...). And these jobs or tasks are referred to as "Cron Jobs".
- Cron Jobs are used for scheduling tasks to run on the server.
- They're most commonly used for automating system maintenance or administration.
- They are also relevant to web application development and Database and Unix Server Maintenance .
- crontab allows a user who has the right to add jobs to the system chronological tables



Format of crontab entries

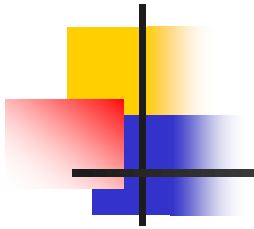
- Minute (0-59)
- Hour (0-23)
- day of month (1-31)
- Month of the year (1-12)
- day of week (0-6) with 0 being Sunday
- Command

Examples :

* * * * * command

To schedule job run at 11:25am on may 25, every week

25 11 25 5 * /tmp/run.sh



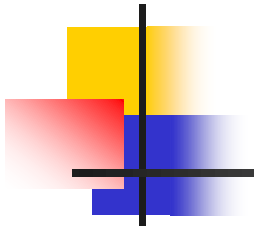
- Cron daemon --- crond

  ps -ef | grep -i cron
- **Log file** /var/spool/cron
- **Config file** /etc/crontab
- **To check the cron status**

  #service crond status

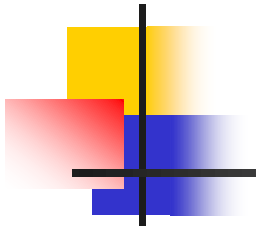
  #service crond start/stop
- chkconfig crond on
- To check service on which run level its running

  chkconfig --list crond



- crontab –e allows the user to edit their entries
- crontab –l allows a listing of current entries
- crontab –r removes all entries for a given user
- crontab file adds the entries in file to your crontab
- Crontab -e

```
* * * * *  /usr/bin/who  >>  /root/who.txt
* * * * *  /usr/bin/uptime  >>  /root/uptime.txt
* * * * *  /bin/date  >>  /root/date.txt
09 14 25 5 *  /bin/df  >>  /root/df.txt
```

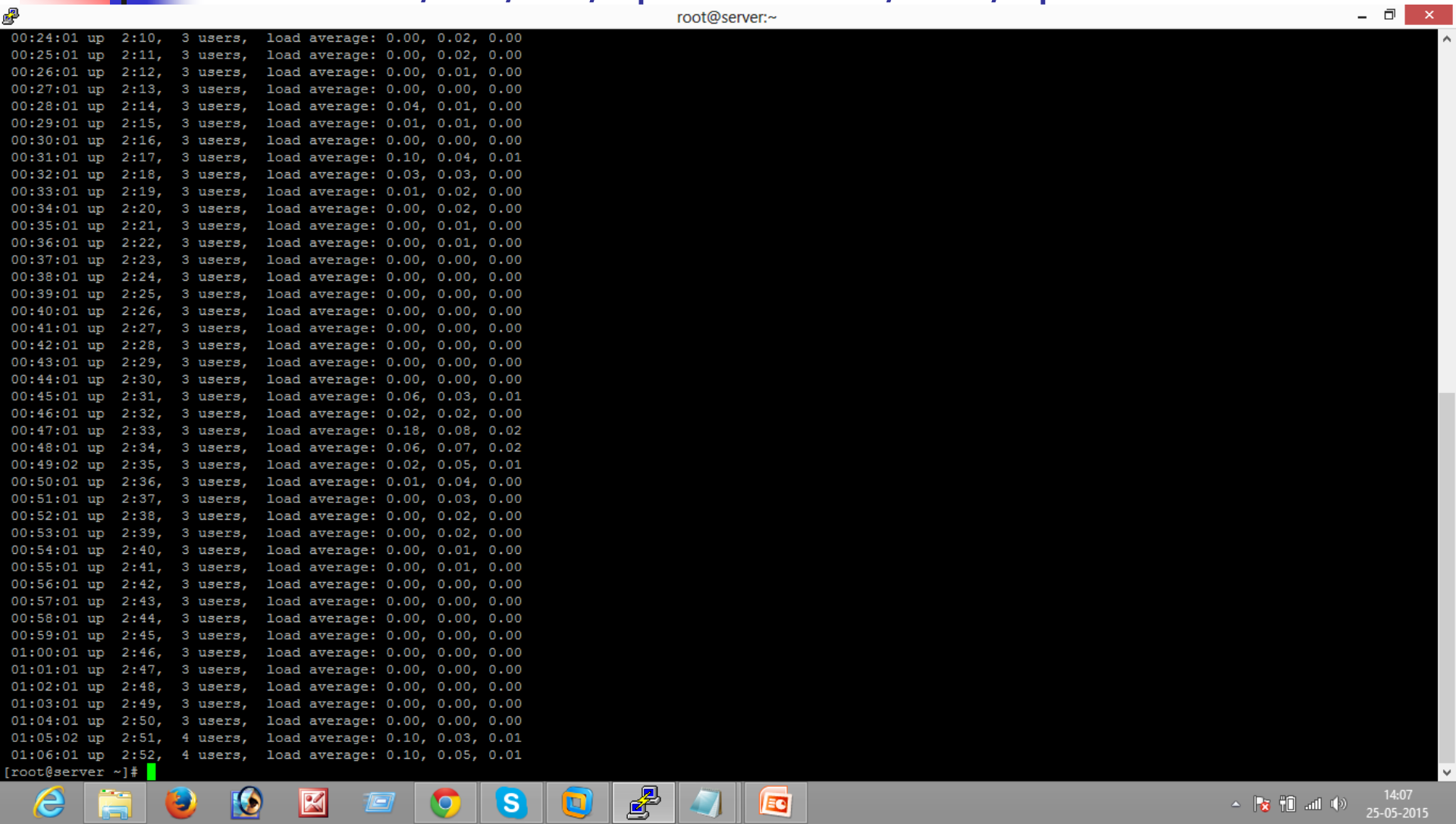


- To view the cron job of user(jeff)

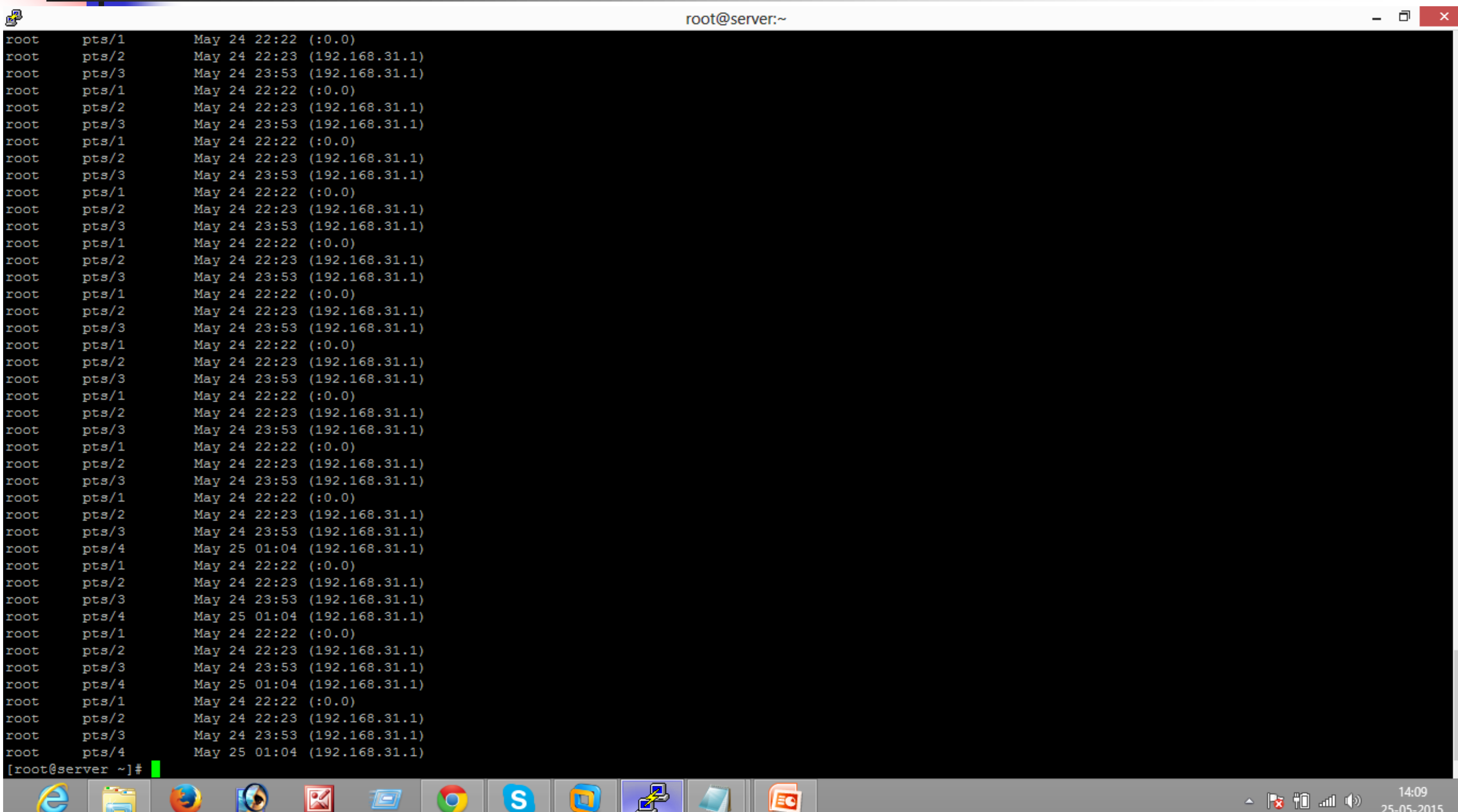
  #crontab -u jeff -l
- To delete the cron job of user

  #crontab -u sandy -r

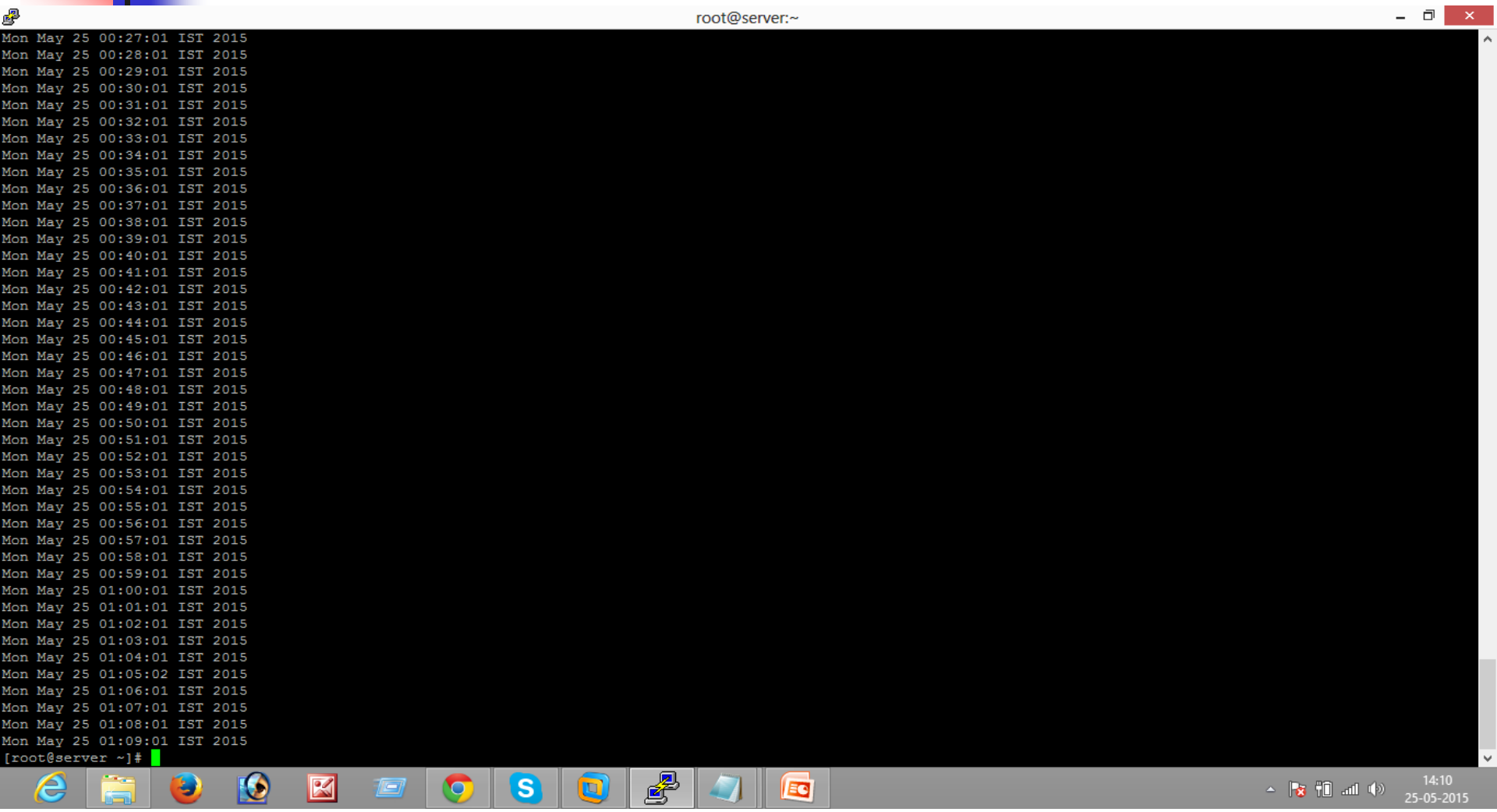
* * * * * /usr/bin/uptime >> /root/uptime.txt



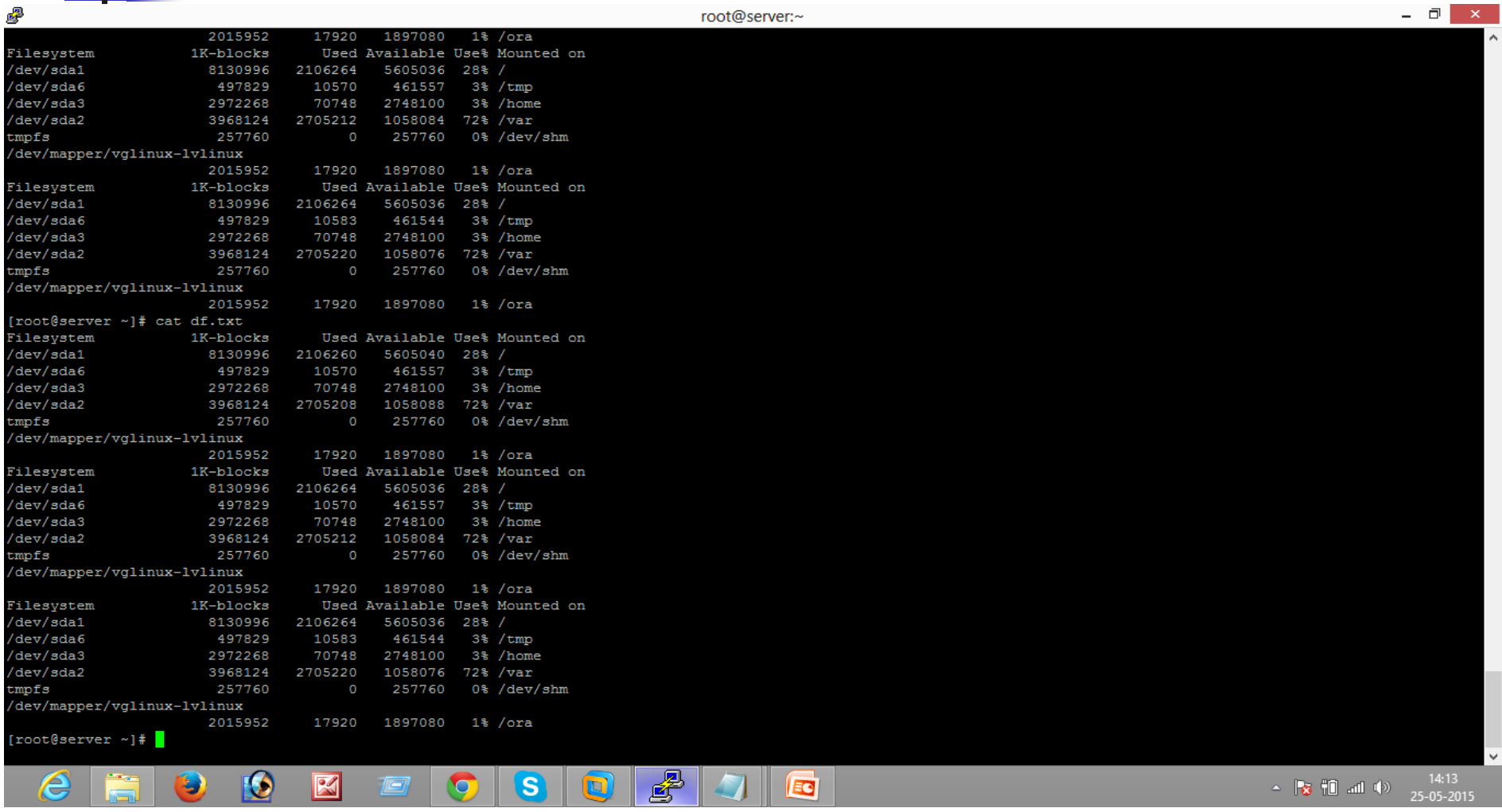
* * * * * /usr/bin/who >> /root/who.txt



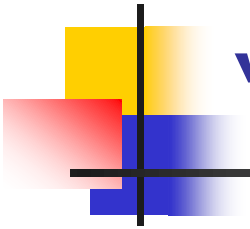
* * * * * /bin/date >> /root/date.txt



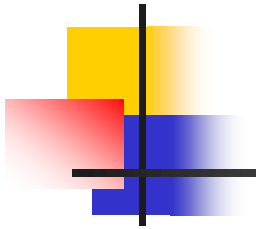
12 14 25 5 * /bin/df >> /root/df.txt



Controlling Access to the "crontab" command

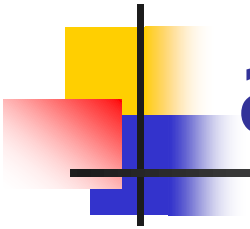


- "root" user can control the access to the "cron" command with "cron.deny" & "cron.allow" files. (Consists usernames one per line)
- The "/etc/cron.d/cron.deny" File:
- /etc/cron.deny
  - This file lists the users who are denied usage of "at" command.
  - This file initially contains "daemon, bin, smtp, nuucp, listen, nobody, noaccess"
  - To deny the "cron" usage for a user add the username to this file



- The “/etc/cron.d/cron.allow” File:
  - This file does not exist by default, so all users except listed in the “cron.deny” file will be allowed to use “crontab” command.
  - By creating this file, you create a list of only those users who are allowed to execute “crontab” command.
  - If you have the user created in cron.allow & cron.deny only it reads cron.deny.

Providing cron and deny access



- /etc/cron.d/cron.allow

  File which allows user cron access
- /etc/cron.d/cron.deny

  File which denies user cron access

How to remove the specified user cron entries

```
[root@testlinux tmp]# date
Wed Mar  2 15:51:38 IST 2016
[root@testlinux tmp]# crontab -l
* * * * * /usr/bin/uptime >> /tmp/uptime.txt
* * * * * /bin/date >> /root/date1.txt
42 10 2 3 * /usr/bin/who >> /root/who.txt
* * * * * /bin/tar -zcf /var/tmp/telnet-0.17-38.el5.i386.rpm.tar /var/tmp

You have new mail in /var/spool/mail/root
[root@testlinux tmp]# crontab -r -u root
[root@testlinux tmp]# crontab -l
no crontab for root
[root@testlinux tmp]#
```

- #crontab –l
- #crontab –r –u username

Cronjob examples

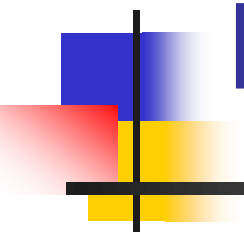
- To run job every min,hour,day ,month

  * * * * * command
- To run top command every day at 7:30am

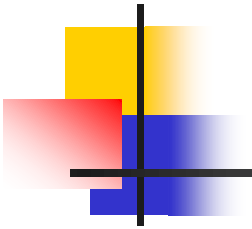
  30 07 * * * /usr/bin/top >> /root/top.txt
- To run a script at 5:44am on dec 25 irrespective of the weekday

  44 05 25 12 * command/script
- To run a script report.sh to run at 10:10 am every Thursday of each week

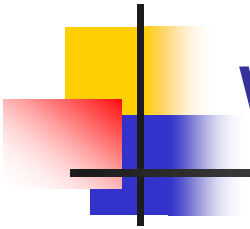
  10 10 * * 5 script



Process Scheduling

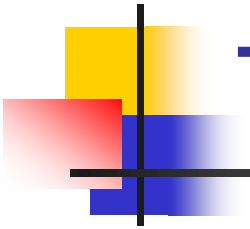


- Process is a program that has been loaded from a long-term storage device, usually a hard disk drive, into system RAM and is currently being processed by the CPU on the motherboard
- Init is the first process started at boot time , always has PID 1.
- Except init every process has a parent.
- Pstree Shows the process ancestry for all the process running on the system.



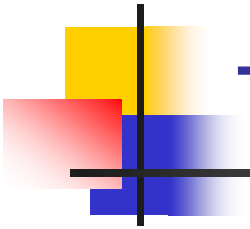
What Exactly Is a Process?

- **Binary executables :** created as a text file using a programming language, such as C or C++. The text file was then run through a compiler to create a binary file that can be processed by the CPU.
- **Internal shell command :** rpm, cd, mkdir,..
- **Shell scripts :**



Types of Processes

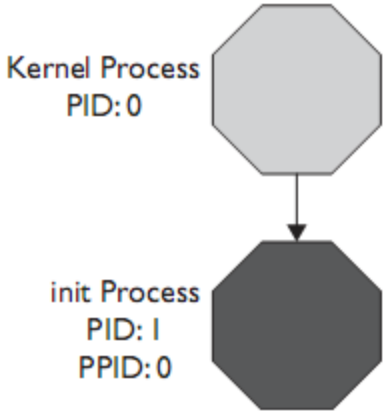
- User Processes: Some processes are created by the end user when he or she executes a command from the shell prompt or though the X Windows graphical interface.
- System processes or daemons: Web server, an FTP server, a file service such as Samba, a print service such as CUPS, a logging service, and so on.
- Ex: cupsd, hald, sshd,...

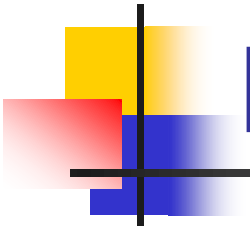


The Heredity of Linux Processes

- ■ **Process ID (PID)** Number This is a number assigned to each process that uniquely identifies it on the system.
- ■ **Parent Process ID (PPID)** Number This is the PID of the process' parent process.

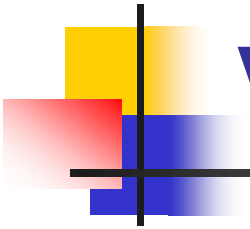
Parent Process ID





Process states

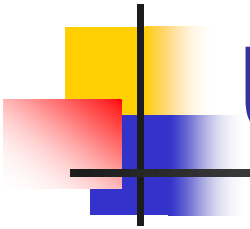
- **`state`** values:
  - **`TASK_RUNNING`** (executing on CPU or runnable).
  - **`TASK_INTERRUPTIBLE`** (waiting on a condition: interrupts, signals and releasing resources may "wake" process).
  - **`TASK_UNINTERRUPTIBLE`** (Sleeping process cannot be woken by a signal).
  - **`TASK_STOPPED`** (stopped process e.g., by a debugger).
  - **`TASK_ZOMBIE`** (terminated before waiting for parent).



Viewing Running Processes

- Using top
- Using ps
- gnome-system-monitor (gui)

Using top



- Top is a powerfull tool
- Dynamic and refreshing tool
- We can see combination of commands
- ■ USER The name of the user that owns the process.
- ■ PR The priority assigned to the process. (We'll discuss process priorities later in this chapter.)
- ■ NI This is the nice value of the process. (We'll talk about what this means later in this chapter.)
- ■ VIRT The amount of virtual memory used by the process.
- ■ RES The amount of physical RAM the process is using (its resident size) in kilobytes.

Using top...

- ■ **%CPU** The percentage of CPU time used by the process.
- ■ **%MEM** The percentage of available physical RAM used by the process.
- ■ **TIME+** The total amount of CPU time the process has consumed since being started.
- ■ **COMMAND** The name of the command that was entered to start the process.

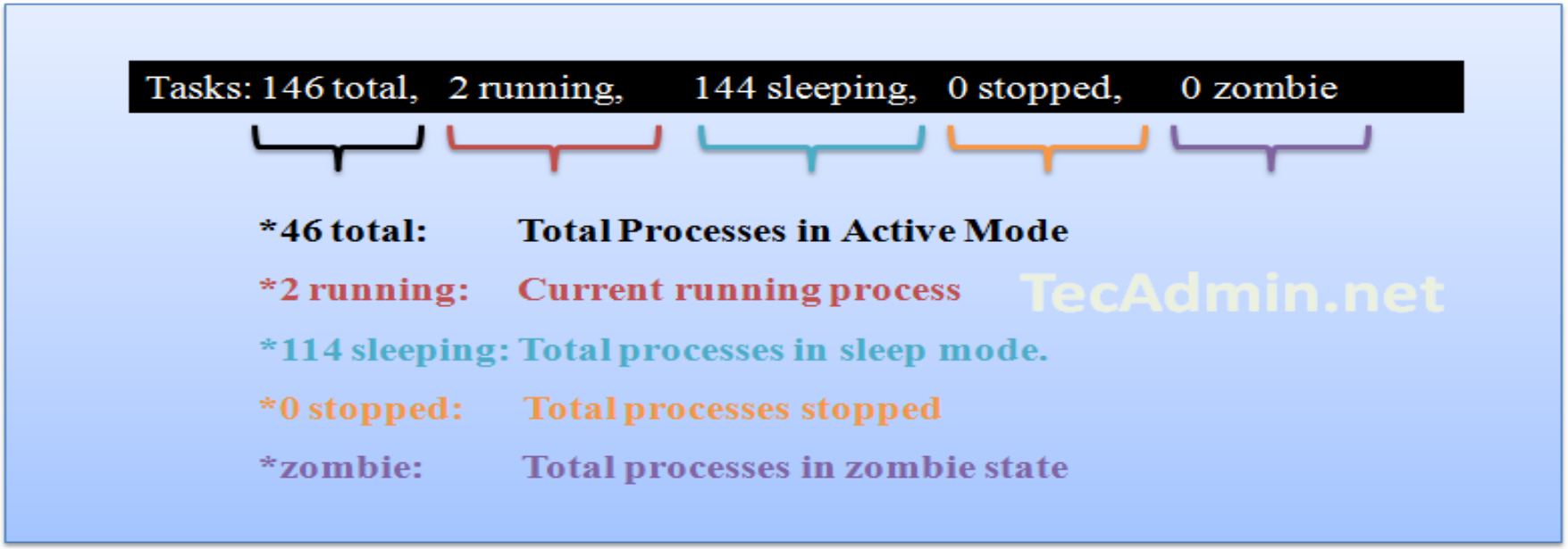
Top – 1st Row

- This first line indicates in order:
- current time (11:37:19)
- uptime of the machine (up 1 day, 1:25)
- users sessions logged in (3 users)
- average load on the system (load average: 0.02, 0.12, 0.07) the 3 values refer to the last minute, five minutes and 15 minutes.

```
top - 11:37:19 up 1 day,  1:25,  3 users,  load average: 0.02, 0.12, 0.07
```

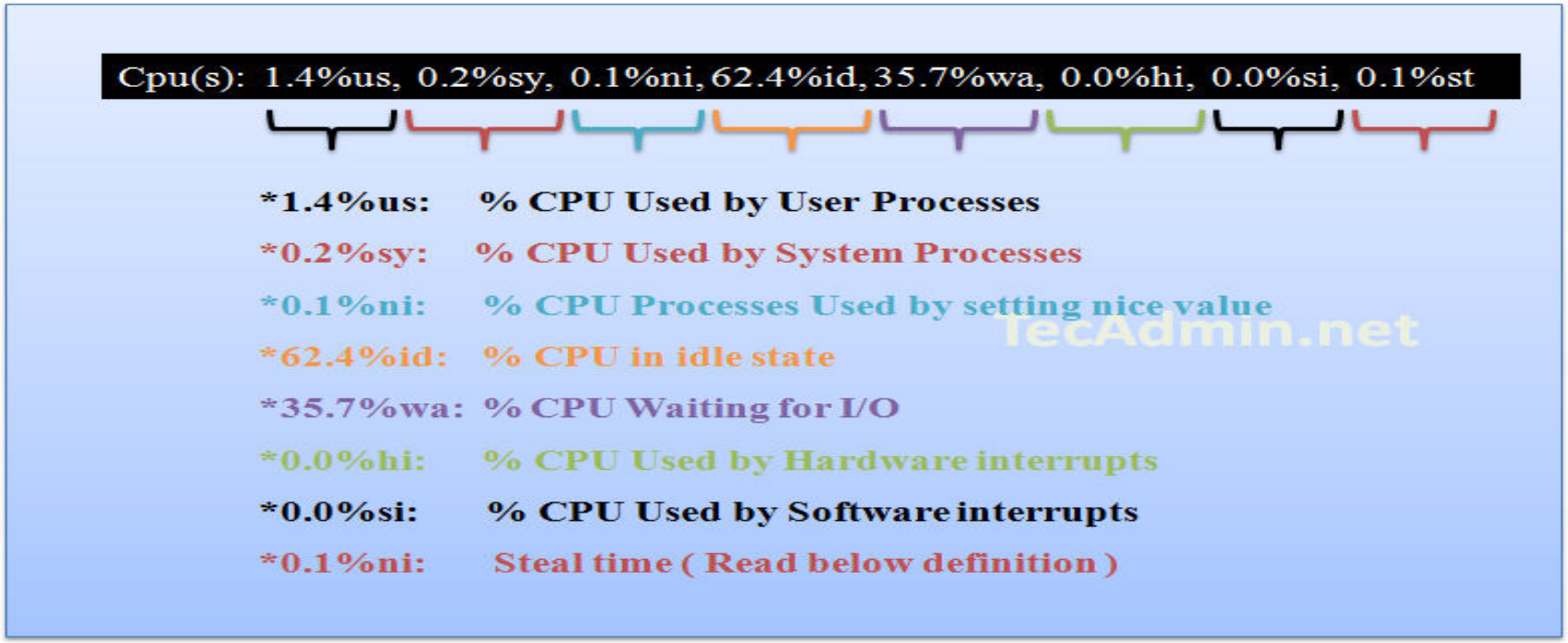
TOP -2nd Row

- The second row gives the following information:
- Row 2 shows the number of process running on server and there state.



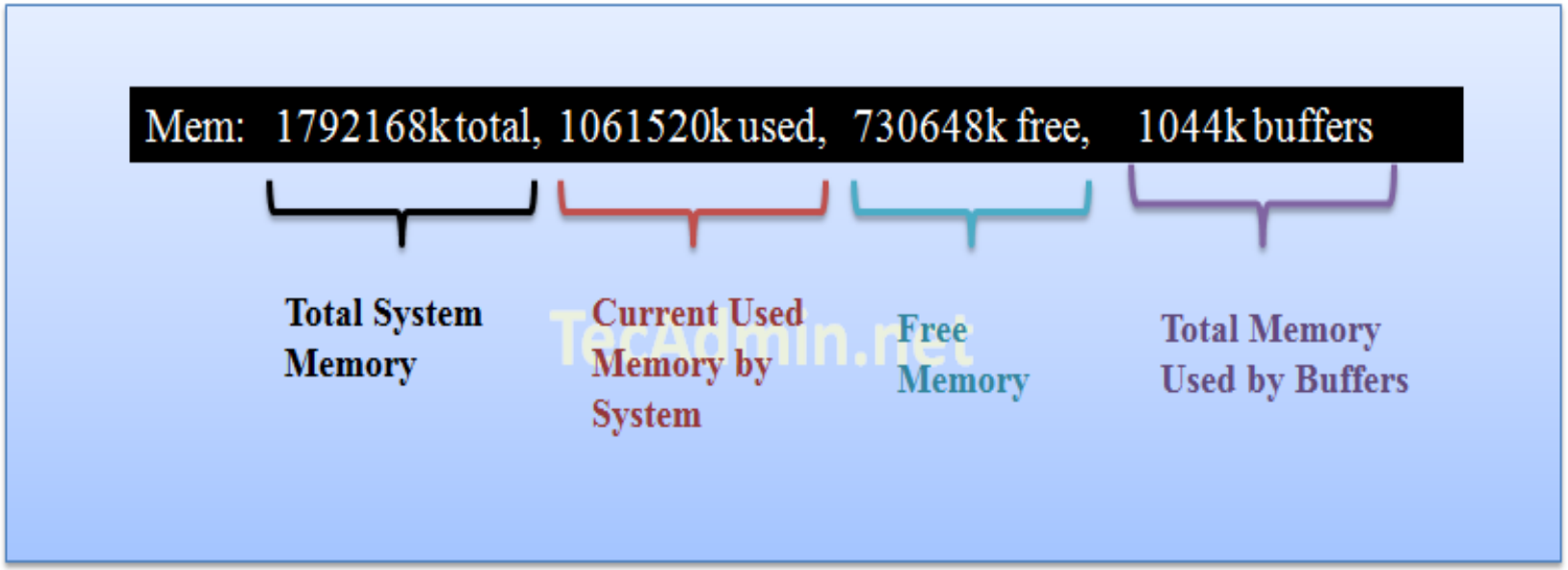
TOP –Row3

- 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.



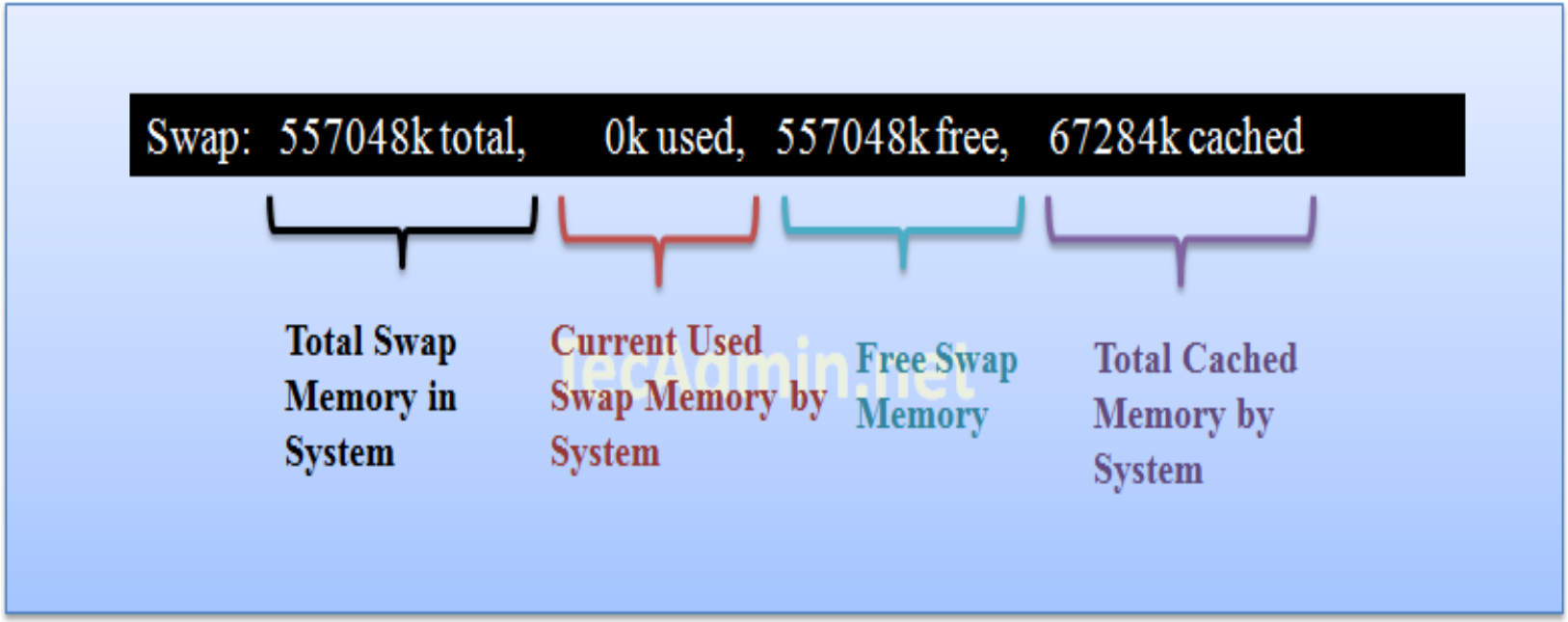
TOP-Row 4

- 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.



TOP-Row 5

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



TOP-Row 6

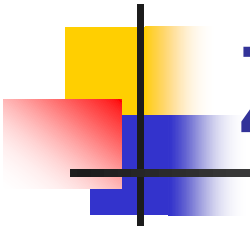
- In this steps you will see all running process on servers and there additional details about them like below.

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
5752	root	20	0	176m	25m	12m	S	6	0.4	3:53.01	jsvc

*PID(5752):	PID of running Process
*USER(root):	User under which process is running
*PR(20):	Priority Of running Process
*NI(0):	Nice Value of running Process
*VIRT(176):	Virtual Memory used by Process
*RES(25m):	Physical Memory used by Process
*SHR(12m):	Shared Memory used by Process
*S(S):	Current Status of Running Process
*%CPU(6):	% CPU Used by this Processes
*%MEM(0.4):	% RAM Used by this Process
*TIME+(3:53:01):	Total time of process running for
*COMMAND(jsvc):	Name of Process

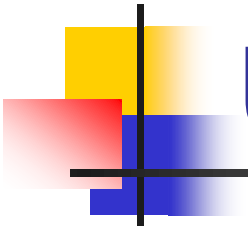
TOP Shortcuts

- **Note:** Press below shortcuts at the time of running top command.
- l –To display or to hide load average line
  t –To display or to hide task/cpu line
  1 –To display or hide all other CPU's
  m –to display or to hide RAM and SWAP details
  s –To change the time interval for updating top results(value is in sec's)
  R –To sort by PID number
  u — Press u then username to get only that user process details
  P –To sort by CPU utilization
  M –To sort by RAM utilization
  c –To display or hide command full path
  r –To renice a process, press r then the PID no then the renice value to renice a process.
  k –To kill a process, press k then PID number then enter to kill a process
  w –To save the modified configuration permanently.
  q –To quit the top command.
  h –for getting help on top command



ZOMBIE

- **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.
- **Zombie process is a process state when the child dies before the parent**
- **Preap**
- **Kill -9**



Using ps..

- ps –e : viewing all processes
- Ps –aux (to check process)
- Usefull options:

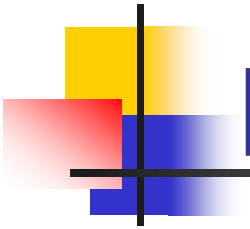
  a = process by all users

  x = process from all terminal

  u = show process owner

  w = include command arguments

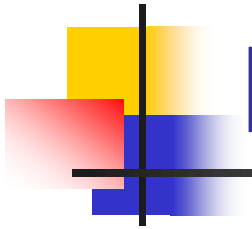
  f = show process anchestry
- ps –f :
  - UID The user ID of the process' owner.
  - PPID The PID of the process' parent process.
  - C The amount of processor time utilized by the process.
  - STIME The time that the process started
  - ps -l



ps -l

- ■ S The state of the process. This column uses the following codes:
  - ■ D Uninterruptible sleep.
  - ■ R Running.
  - ■ S Interruptible sleep.
  - ■ T Stopped or traced.
  - ■ Z Zombied.
- ■ PRI The priority of the process.
- ■ NI The nice value of the process. We'll talk about what this means in the
- next section.
- ■ SZ The size of the process.
- ■ WCHAN The name of the kernel function in which the process is sleeping.
- You will see a dash (–) in this column if the process is currently running.

Ending a Running Process

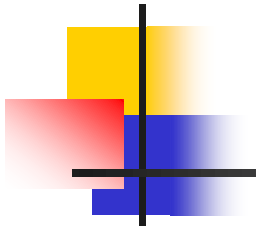


- Kill
  - ■ SIGHUP This is kill signal 1. This signal restarts the process. After
  - a restart, the process will have exactly the same PID that it had before.
  - This is a very useful option for restarting a service for which you've made
  - changes in a configuration file.
  - ■ SIGINT This is kill signal 2. This signal sends a key sequence to
  - the process.
  - ■ SIGKILL This is kill signal 9. This is a brute-force signal that kills the
  - process. If the process was hung badly, this option will force it to stop. However,
  - the process may not clean up after itself if this signal is used. The resources
  - allocated to the process may remain allocated until the system is restarted.
  - ■ SIGTERM This is kill signal 15. This signal tells the process to terminate
  - immediately. This is the default signal sent by kill if you omit a signal in the
  - command line. This signal allows the process to clean up after itself before exiting.
- Killall

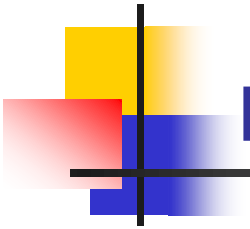
- Exercise 10-1: Working with Linux Processes

Usage of free command

- Total memory
- Used memory
- Free memory
- Shared memory:Indication of how much memory is being shared to 1 process or more.
- Buffers: It's a temporary location to store date for a particular application and this **data is not used** by any other application .
- Cache : It's a memory location to store frequently used data for faster access .this cache used for multiple access.

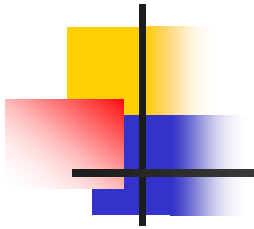


- #free
- #free –m
- #free –g
- #free -c 5 -s 1 (c=count s=time period)
- #free -g -c 5 -s 1 (g=gigabyte)
- #free –l
- **Low memory**: it's a segment memory that reserved for linux kernel.
- **High memory:** is simply reserved for user space programs such as application running on a system .

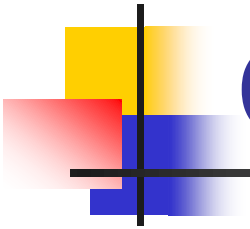


How to clear the cache memory

- #free –m
- sync; echo 3 > /proc/sys/vm/drop_caches

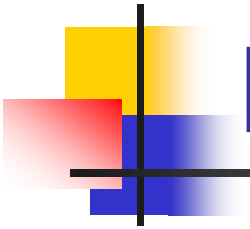


- To terminate a process, highlight it and press Control-C, select the “Kill” option from the options that are available.



Clearing Frozen Processes

- “kill -signal PID”
- “pkill -signal Process”
- Signal Numbers and Names
  - 1 SIGHUP (Hangup)
  - 2 SIGINT (Interrupt)
  - 9 SIGKILL (Kill)
  - 15 SIGTERM (Terminate)

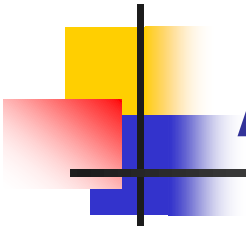


Prioritizing Processes

- nice –n nice_levelcommand.
- Renice n pid

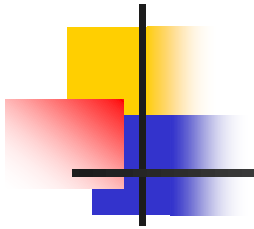
Altering process schedule prority

- Nice is the command to change the priority.
- Process are scheduled with default priority of 0.
- Priority value can range from -20(highest priority) to 19 (lowest)
- Syntax
- #nice [–n adjustment] command
- #nice top
- #nice –n 15 top
- Only super user can change the priority.

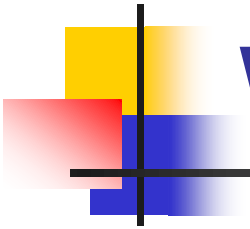


Altering process schedule prority

- Renice changes the priority of a running process.
- Once a priority value is raised, a non privileged user can't lower it.
- If you want to change the the priority of an entire user group, use the –g option.
- To modify the priority of all the process of a particular user
  use –u option.
- #renice 15 –u username command
- #renice -15 –p (pid of user process)
- -15 is high priority
- 15 is low priority.



Run Levels



What is a run level?

- *Run level* is simply a number that Linux uses to distinguish different types of high-level configurations that the machine should be booted into. These run level numbers are on the whole "well-known," in the sense that most of them have very clearly defined meanings.

Init Runlevels

- **The following 7 runlevels are defined in Linux:**
  - 0 - halt (Do NOT set initdefault to this)
  - 1 - Single user mode
  - 2 - Multiuser, without Network (The same as 3, if you do not have networking)
  - 3 – Multiuser with Network
  - 4 - unused
  - 5 – Graphical Mode/ X11
  - 6 - reboot (Do NOT set initdefault to this)

Init Runlevels

- The default runlevel for a system to boot to is configured in **/etc/inittab.**

  id:3:initdefault:

- **In GUI: Applications → System Settings → Server Settings→ Services**
- **Generally, Linux operates in runlevel 3 or 5.**

How to see the run level?

- You can display your previous and current run level using the runlevel command.
- # runlevel

  N 5
- #who –r

run-level 5 2016-03-03 14:50 last=S

Switching or Changing between different runlevels:-

- **Method-1**: Changing run level temporarily without reboot.
- We can use init command to change rune levels without rebooting the system.
- Ex:-if we are currently in run level 3 and want to go to run level 1, just we need to execute
- # init 1
- Or if you want to shutdown a machine you can take help of run level '0' .Just you need to execute
- #init 0

Remember this change is not permanent and on next reboot you will get your default runlevel.

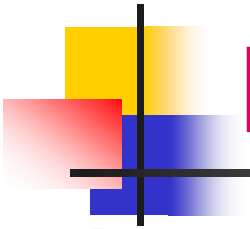
Method-2

- **Method-2: Changing run level permanently**
- If you want to change your default run level then
- Open the file /etc/inittab and edit entry initdefault:
- # vi /etc/inittab
- Let's set initdefault to 5, so that you can boot to X next time when Linux comes up:

  #**id:5:initdefault:**

Method-3

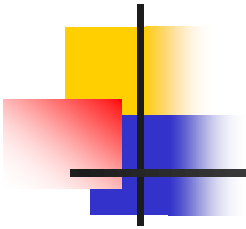
- **Method-3:-Change run level at boot time**
- You can also change the run level at boot time. If your system uses LILO as the boot manager, you can append the run level to the boot command:
- LILO: linux 3 or
  LILO: linux 5
- If your system uses GRUB, you can change the boot runlevel by pressing the `e' key to edit the boot configuration. Append the run level(in our case 5) to the end of the boot command as shown:
- kernel /vmlinuz-2.6.18-164.el5 ro root=LABEL=/ rhgb quiet 5



Linux Services

There are 113 deamons, Out of them, the following are most widely used:

- **apmd** : Power Management
- **autofs** : Automount services
- **crond** : Periodic Command Scheduler
- **cups** : Common Unix Printing System
- **dhcpd** : The DHCP server
- **dovecot** : IMAP (Internet Message Access Protocol) and POP3 (Post Office Protocol) server
- **gpm** : Mouse
- **httpd** : Apache Web server



Linux Services

- **iptables** : Kernel based Packet Filtering firewall
- **kudzu:** Finds new Hardware
- **mysqld** : MySQL server
- **named** : BIND server
- **nfs** : Network File Share
- **nfslock** : NFS file locking
- **ntpd** : NTP (Network Time Protocol) server
- **portmap** : RPC (Remote Procedure Call) support

Linux Services

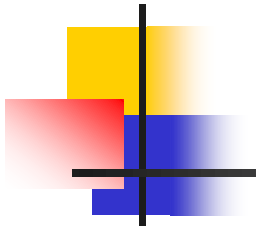
- **sendmail** : Sendmail Mail Server
- **smb** : Samba Network Services
- **snmpd** : Simple Network Management Protocol
- **squid** : Squid Proxy Server
- **sshd** : Open SSH and SFTP server
- **syslog** : System Logging
- **xinetd** : Provides support for telnet, ftp, talk, tftp etc.
- **ypbind** : NIS Server
- **postgresql** : The Postgresql Database Engine

Linux services & Port No

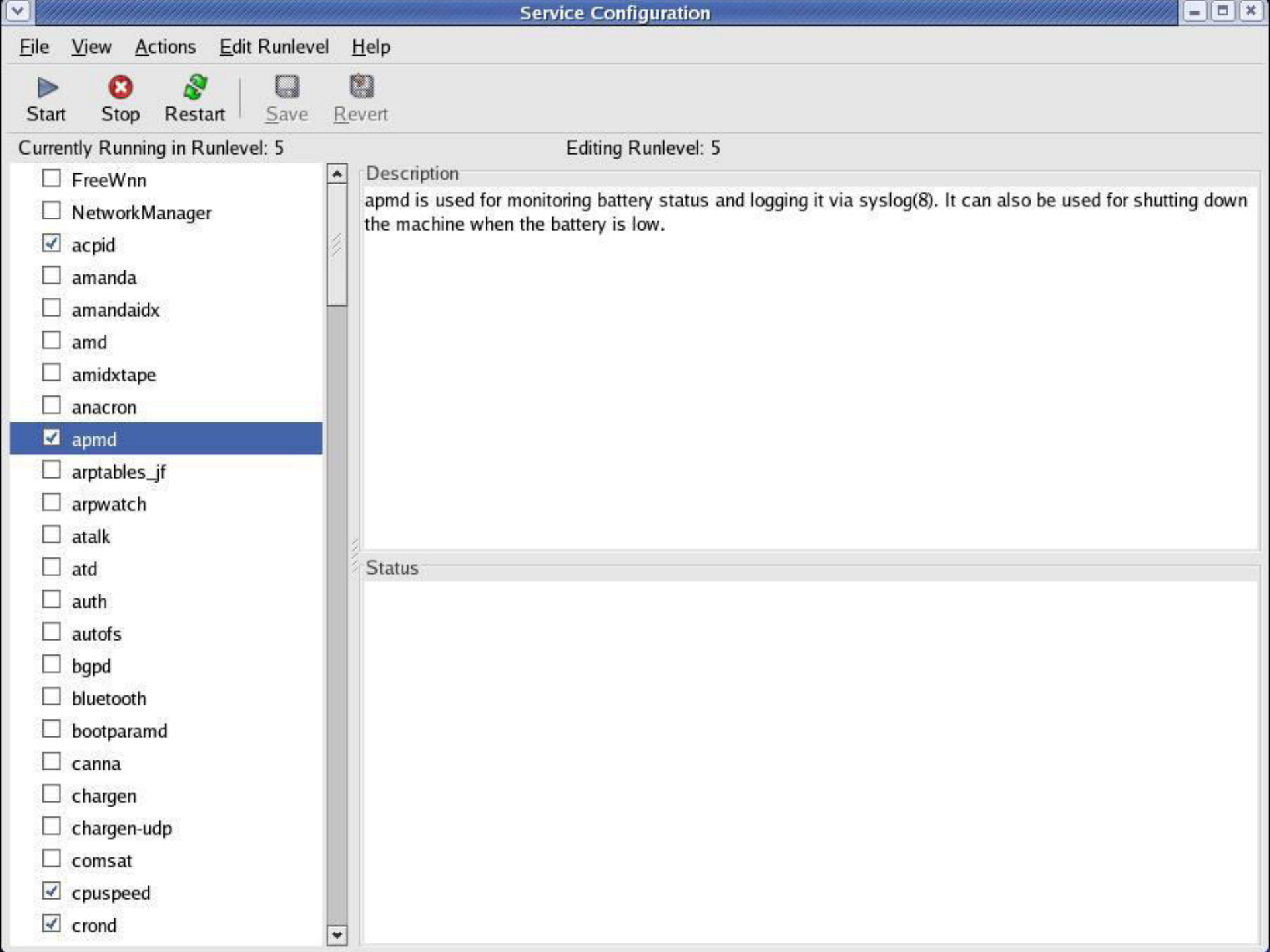
- **The configuration file set for services**

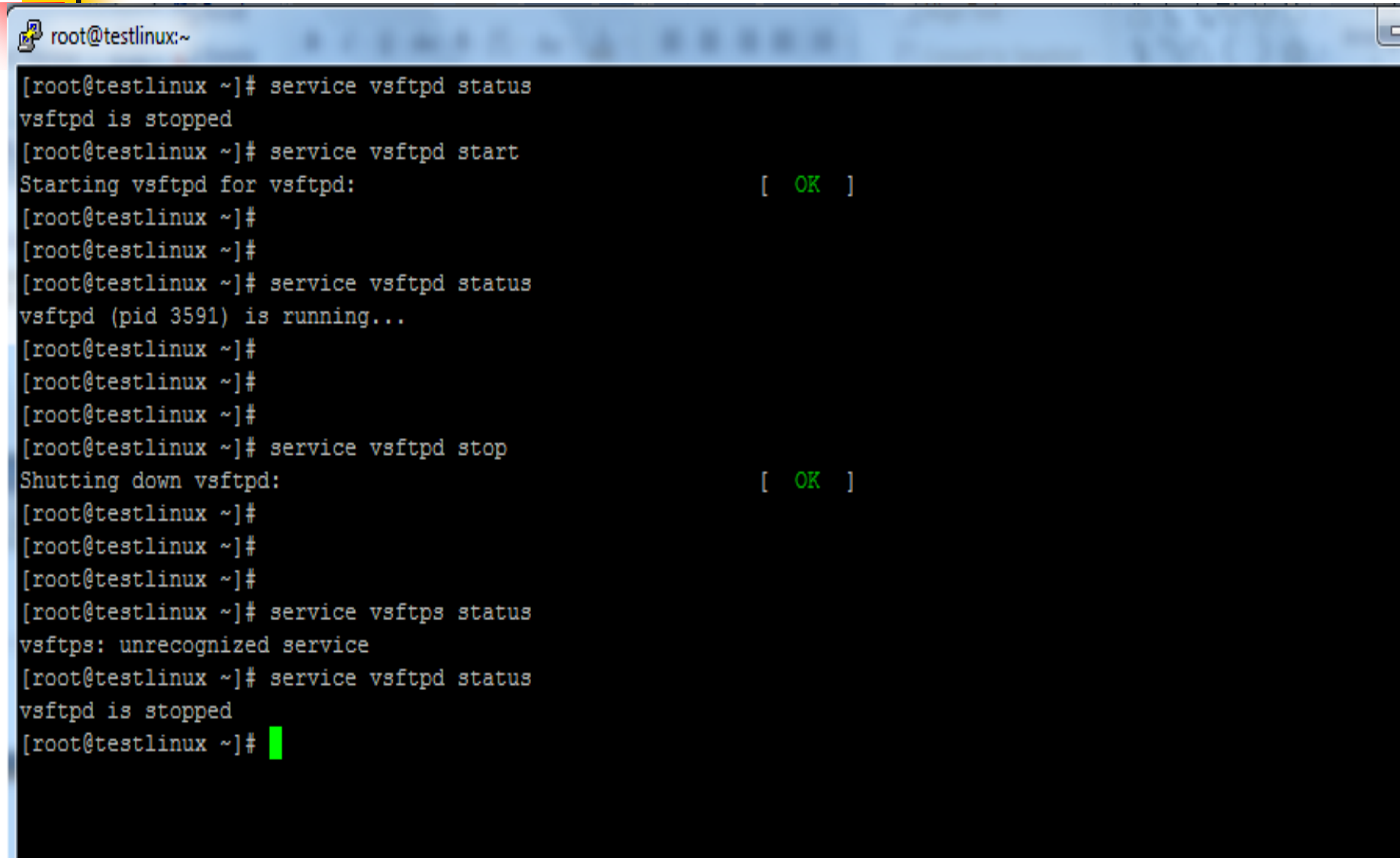
#/etc/services

- **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)
- **67 – Bootp**
- **68 – DHCP**
- **80 – HTTP**/WWW(apache)
- **88 – Kerberos**
- **110 – POP3**(Mail delivery Agent)
- **123 – NTP**(Network time protocol used for time syncing uses UDP protocol)
- **137 – NetBIOS**(nmbd)
- **139 – SMB-Samba**(smbd)



- 143 – IMAP
- 161 – SNMP(For network monitoring)
- 389 – LDAP(For centralized administration)
- 69 – TFTP(Trivial file transfer protocol uses udp protocol for connection less transmission of data)
- 443 – HTTPS(HTTP+SSL for secure web access)
- 514 – Syslogd(udp port)
- 636 – ldaps(both tcp and udp)
- 873 – rsync
- 993 – IMAPS
- 1194 – openVPN
- 1812 – RADIUS
- 995 – POP3s
- 2049 – NFS(nfsd, rpc.nfsd, rpc, portmap)
- 2401 – CVS server
- 3306 – MySql
- 3690 – SVN
- 6000-6063-X11


Service Configuration
File View Actions Edit Runlevel Help
Start Stop Restart Save Revert
Currently Running in Runlevel: 5
Editing Runlevel: 5
FreeWnn
NetworkManager
acpid
amanda
amandaidx
amd
amidxtape
anacron
apmd
arptables_jf
arpwatch
atalk
atd
auth
autofs
bgpd
bluetooth
bootparamd
canna
chargen
chargen-udp
comsat
cpuspeed
crond
Description
apmd is used for monitoring battery status and logging it via syslog(8). It can also be used for shutting down the machine when the battery is low.
Status


root@testlinux:~
[root@testlinux ~]# service vsftpd status
vsftpd is stopped
[root@testlinux ~]# service vsftpd start
Starting vsftpd for vsftpd: [OK]
[root@testlinux ~]#
[root@testlinux ~]#
[root@testlinux ~]# service vsftpd status
vsftpd (pid 3591) is running...
[root@testlinux ~]#
[root@testlinux ~]#
[root@testlinux ~]#
[root@testlinux ~]# service vsftpd stop
Shutting down vsftpd: [OK]
[root@testlinux ~]#
[root@testlinux ~]#
[root@testlinux ~]#
[root@testlinux ~]# service vsftps status
vsftps: unrecognized service
[root@testlinux ~]# service vsftpd status
vsftpd is stopped
[root@testlinux ~]#

How to enable service at boot time ?

#Chkconfig –list vsftpd

(To check)

#chkconfig vsftpd on

(To make it enable at boot time)

```
[root@testlinux ~]# service vsftpd status
vsftpd is stopped
[root@testlinux ~]# chkconfig --list vsftpd
vsftpd          0:off   1:off   2:off   3:off   4:off   5:off   6:off
[root@testlinux ~]#
[root@testlinux ~]#
[root@testlinux ~]# chkconfig vsftpd on
error reading information on service vsftpd: No such file or directory
[root@testlinux ~]# service vsftpd status
vsftpd is stopped
[root@testlinux ~]# service vsftpd start
Starting vsftpd for vsftpd:                                [  OK  ]
[root@testlinux ~]# chkconfig vsftpd on
[root@testlinux ~]# chkconfig --list vsftpd
vsftpd          0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@testlinux ~]# reboot

Broadcast message from root (pts/1) (Thu Mar  3 15:34:33 2016):

The system is going down for reboot NOW!
[root@testlinux ~]#
login as: root
root@192.168.0.10's password:
Last login: Thu Mar  3 15:28:43 2016 from 192.168.0.5
[root@testlinux ~]# bash
[root@testlinux ~]#
[root@testlinux ~]# service vsftpd status
vsftpd (pid 3018) is running...
[root@testlinux ~]# runlevel
N 5
[root@testlinux ~]# chkconfig --list vsftpd
vsftpd          0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@testlinux ~]#
```

Linux Services

- **Start/Stop boot time services in /etc/rc.d/rc3.d or /etc/rc.d/rc5.d**
- All services startup scripts which start with **S will start at boot time** and all startup scripts which **start with K will not start at boot time**. The number after S or K is the priority.

  K95kudzu

  K96pcmcia

  S56xinetd

  S60vsftpd

- **Use**
- **#service --status-all**

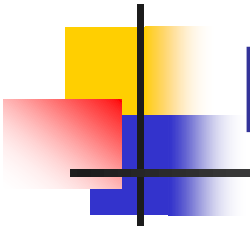
  #service <service name> start/stop/restart

  (to start, stop or restart a service from command line)

  #/etc/init.d/vsftpd status

  #/etc/init.d/vsftpd start/stop

Listing the services



- The **chkconfig** utility is a command-line tool that allows you to specify in which runlevel to start a selected service, as well as to list all available services along with their current setting

  #chkconfig –list

- To display the current settings for a selected service

  #chkconfig --list *service_name*

  # chkconfig --list vsftpd

Enabling service

- To enable a service in runlevels 2, 3, 4, and 5,

  #chkconfig *service_name* on

  #chkconfig vsftpd on

- To enable a service in certain runlevels only
- chkconfig *service_name* on --level *runlevels*

  # chkconfig --list vsftpd
  vsftpd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
  # chkconfig vsftpd on --level 35
  # chkconfig --list vsftpd
  vsftpd 0:off 1:off 2:off 3:on 4:off 5:on 6:off

Disabling service

- To disable a service in runlevels 2, 3, 4, and 5

  #chkconfig *service_name* off

  # chkconfig --list vsftpd

  vsftpd 0:off 1:off 2:on 3:on 4:on 5:on 6:off

  # chkconfig vsftpd off

  # chkconfig --list vsftpd

  vsftpd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
- To disable a service in certain runlevels only, add the --level option followed by numbers from 0 to 6
- chkconfig *service_name* off --level *runlevels*
- *#chkconfig vsftpd off –level 0124*

How to enable service in startup

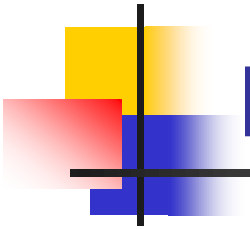
- As an example, lets enable the Apache web server to start in run levels 2, 3, and 5. This is how it is done.

  We first add the service using chkconfig script. Then turn on the service at the desired run levels.

  # chkconfig httpd --add

  # chkconfig httpd on --level 2,3,5
  This will enable the Apache web server to automatically start in the run levels 2, 3 and 5. You

  # chkconfig --list httpd

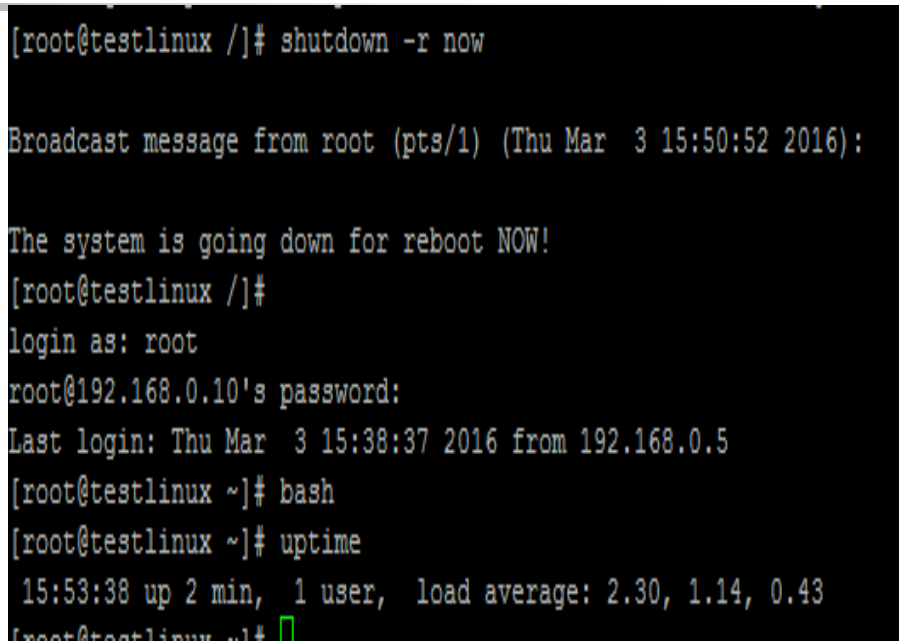


Remove Service from Startup List

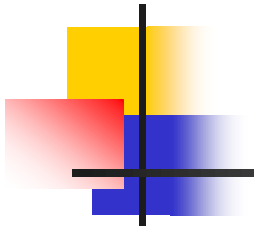
- # chkconfig httpd off
- # chkconfig httpd –del

Reboot & shutdown

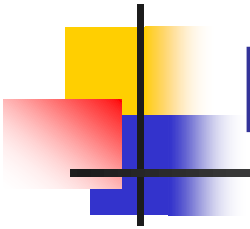
- Any of the following commands will reboot the system from the command line.
- # reboot
- # shutdown -r now
- # init 6



- Shutdown
- Either of the following commands will shut down the system from the command line.
- # shutdown -h now
- # init 0

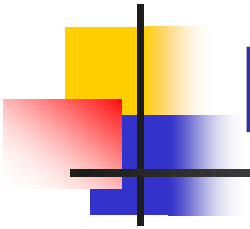


NETWORK FILE SHARING



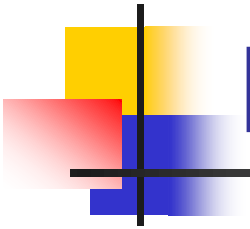
NFS

- Network File System (NFS) protocol allow Linux client to mount remote file systems and interact with those file systems as they are mounted locally.
- NFS stand for Network File System
- NFS is used to share files and printer between Linux / Unix systems
- Red Hat Enterprise Linux 6 supports NFSv2, NFSv3, and NFSv4 clients.
- By default RHEL6 use NFSv4 if the server supports it.



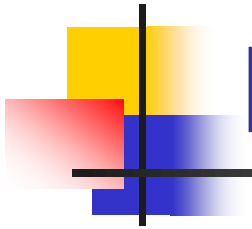
NFSv1

- NFSv1 was the development stage of NFS protocol.
- It was used only for in house experimental purpose.



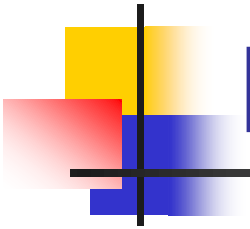
NFSv2

- NFSv2 supports only 32 bit.
- NFSv2 only allowed the first 2 GB of a file to be read
- NFSv2 operated only over UDP



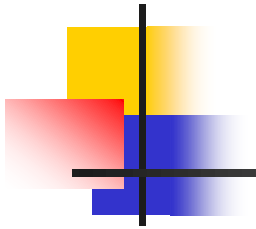
NFSv3

- NFSv3 supports 64 bit file system.
- NFSv3 can handle files larger than 2 GB.
- NFSv3 supports asynchronous writes on the server. asynchronous writes improve write performance.
- NFSv3 supports additional file attributes in many replies, to avoid the need to re-fetch them.
- NFSv3 supports READDIRPLUS operation. READDIRPLUS operation get file handles and attributes along with file names when scanning a directory.
- NFSv3 supports TCP. Using TCP as a transport made NFS over a WAN more feasible.



NFSv4

- NFSv4 retains all NFSv3 advantages.
- NFSv4 supports ACLs.
- NFSv4 uses the virtual file system to present the server's export.
- NFSv4 supports Pseudo file system. Pseudo File System provide maximum flexibility. Exports Pathname on servers can be changed transparently to clients.
- NFSv4 have locking operations as the part of protocol which keep track of open files and delegations.
- NFSv4 works through firewalls and on the Internet.



- NFS port no is 2049
- NFS server config file /etc/exports
- Nfs client config file /etc/fstab
- Nfs shares are mounted at boot time by /etc/rc.d/init.d/netfs

NFS CONFIGURATION LAB

- Create a NFS server machine with
  hostname =server
  ip address =192.168.0.254

- Create a linux client machine with
  hostname =client
  ip address = 192.168.0.253

- make the ip address and hostnames permanent on both server and client side

Configuring NFS SERVER

If we want to configure nfs service we have to install the few Packages like

- **nfs-utils, portmap** , **nfs4-acl-tools xinetd(** rhel 5)
- **nfs-utils, rpcbind** , **nfs4-acl-tools xinetd(** rhel 6)
- check whether the packages installed or not if there are no packages install by using rpm or yum
- rpm -qa | grep nfs

  rpm -ivh nfs* --nodeps --force
- rpm -qa | grep portmap

  rpm -ivh portmap* --nodeps --force
- rpm -qa | grep xinetd*

  rpm -ivh xinetd* --nodeps --force

Configuring NFS SERVER

- restart the xinetd portmap nfs services

  service nfs restart

  service portmap restart

  service xinetd restart

- To enable the service at boot time use chkconfig comm

  chkconfig nfs on

  chkconfig portmap on

  chkconfig xinetd on

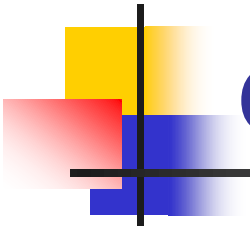
Configuring NFS SERVER

- After rebooting verify the status it must be in running state

  service nfs status

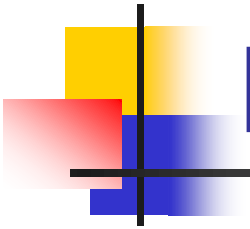
  service portmap status

  service xinetd status
- make a directory /nsfshare and give read write or full permissions to the directory
- Mkdir /nfsshare
- chmod 777 /nfsshare
- now open vi /etc/exports file then add entry in the file
- /nfsshare 192.168.0.253 (rw , sync)
- Save and quit
- Restart the nfs service and restart nfs daemons
- exportfs - avr



exportsfs

- **exportfs** command is used to maintain the current table of exported file systems for NFS.
- This list is kept in a separate file named **/var/lib/nfs/xtab** which is read by **mountd** when a remote host requests access to mount.
- This **xtab** file is initialized with the list of all file systems named in **/etc/exports** by invoking **exportfs –a** .
- administrators can choose to add and delete individual file systems without modifying**/etc/exports** using **exportfs.**
- **exportfs** and it's partner program **mountd** work in one of two modes, a legacy mode which applies to 2.4 and earlier versions of the Linux kernel, and a new mode which applies to 2.6 and later versions providing the **nfsd** virtual filesystem has been mounted at **/proc/fs/nfsd** or**/proc/fs/nfs**. If this filesystem is not mounted in 2.6, the legacy mode is used.



EXPORTFS

- In the new mode, **exportfs** does not give any information to the kernel but only provides it to**mountd** through the **/var/lib/nfs/xtab** file. **mountd** will listen to requests from the kernel and will provide information as needed.
- In the legacy mode, any export requests which identify a specific host (rather than a subnet or netgroup etc) are entered directly into the kernel's export table as well as being written to**/var/lib/nfs/xtab**. Further, any mount points listed in **/var/lib/nfs/rmtab** which match a non host-specific export request will cause an appropriate export entry for the host given in**rmtab** to be entered into the kernel's export table.

Client side configuration

- verify showmount - e 192.168.0.12 (server ip)
- make a directory mkdir /nfs
- mount on /nfs

#mount –t nfs 192.168.0.254 :/nfsshare /nfs

- verify df –h
- mount it permanently in /etc/fstab

192.168.0.254:/nfsshare /nfs nfs defaults 0 0

#df -h

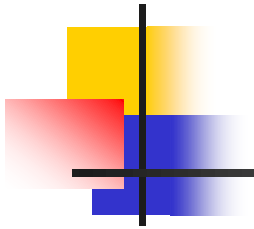
- cd /nfs and create a touch file
- touch client

And verify whether it is shared on server side on not

differences b/w nfsv3 & nfsv4

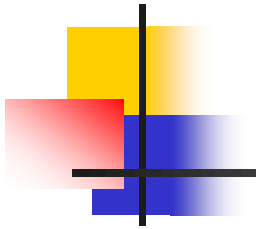
1.Transport protocols

- For NFSv3, the MOUNT service is normally supported over the TCP and UDP protocols.
- For NFSv4, only the TCP protocol is supported.
  NFS v4 is designed for internet use. One unique network port is used on NFSv4. This predetermined port is fixed.
- The default is port 2049. Using NFS v4 through firewalls is easier than with earlier NFS versions.



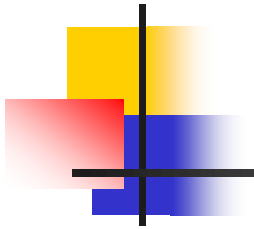
2. Locking operation

- NFS v3 protocol is stateless, so an additional Network Lock Manager (NLM) protocol, an auxiliary protocol for file locking, is required to support locking of NFS-mounted files READ/WRITE. Also NLM is stateful in that the server LOCKED keeps track of locks.
- NFSv4 is stateful. Locking operations(open/read/write/lock/locku/close) are part of the protocol proper. NLM is not used by NFSv4.



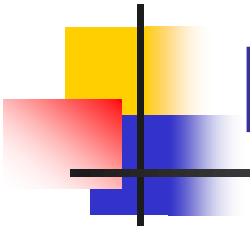
3. Required Services

- NFSv3 relies on Remote Procedure Calls (RPC) to encode and decode requests between clients and servers. NFSv3 depends on portmapper, rpc.mountd, rpc.lockd, rpc.statd.
- NFSv4 has no interaction with portmapper, rpc.mountd, rpc.lockd, and rpc.statd, since protocol support has been incorporated into the v4 protocol. NFSv4 listens on "well-known" TCP port (2049) which eliminates the need for the portmapper interaction. The mounting and locking protocols have been incorpated into the V4 protocol which eliminates the need for interaction with rpc.mountd and rpc.lockd.



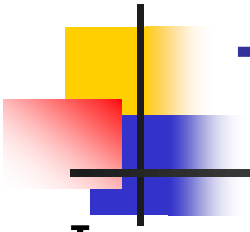
4. Security

- NFS v3 supports export/mount. Thus the host makes the mount request, not a user of the file system.
- With NFSv4, the mandatory security mechanisms are oriented towards authenticating individual users, e.g. by configuring the Kerberos version 5 GSS-API or other security mechanism.



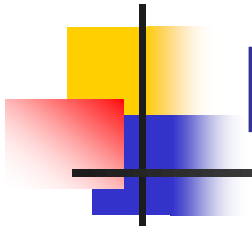
Daemons of nfs-Server

- Nfsd
- Lockd
- Rpciod
- Rpc.{mountd,rquotad,statd}

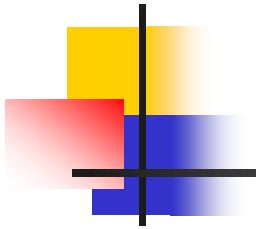


Troubleshooting

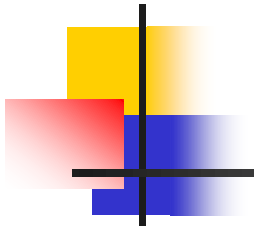
- Issue ----- **df -h is hung**
- check in client df -h , service nfs status, check any nfs mount points
- login to the nfs server
- service nfs status
- service nfs start/restart
- check the mountpoint exists in nfs server
- if it exists ..no issue from server side ..just restart nfs service
- move to client machine mount it
- Mount –t nfs 192.168.0.12:/nfsshare /nfs



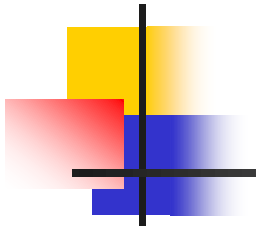
NFS Error codes



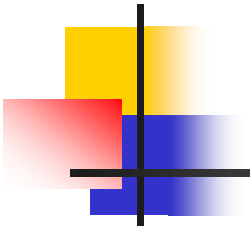
SSH
(SECURED SHELL)



- SSH stand for Secure Shell.
- SSH is a network protocol for secure data communication.
- SSH protocol allows remote command line login.
- SSH protocol enables remote command execution.
- Telnet, rlogin, and ftp transmit unencrypted data over internet.
- OpenSSH encrypt data before sending it over insecure network like internet.
- OpenSSH effectively eliminate eavesdropping, connection hijacking, and other attacks.



- OpenSSH provides secure tunneling and several authentication methods.
- OpenSSH replace Telnet and rlogin with SSH, rcp with scp, ftp with sftp.
-



SSH Tools

- **ssh**

  The ssh [Secure Shell command] is a secure way to log and execute commands in to SSH Server system.

- **sshd**

  The daemon service that implements the ssh server. By default it must be listening on port 22 TCP/IP.

- **scp**

  The Secure Copy command is a secure way to transfer files between computers using the private/public key encryption method.

SSH Packages

- we have to install two packages to configure ssh

1.Openssh

2.Openssh-server

#rpm -qa | grep –i ssh

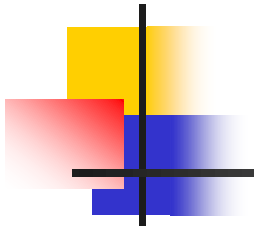
#rpm –qa | grep –i openssh

#yum install openssh* (if there are no packages)

#yum install ssh*

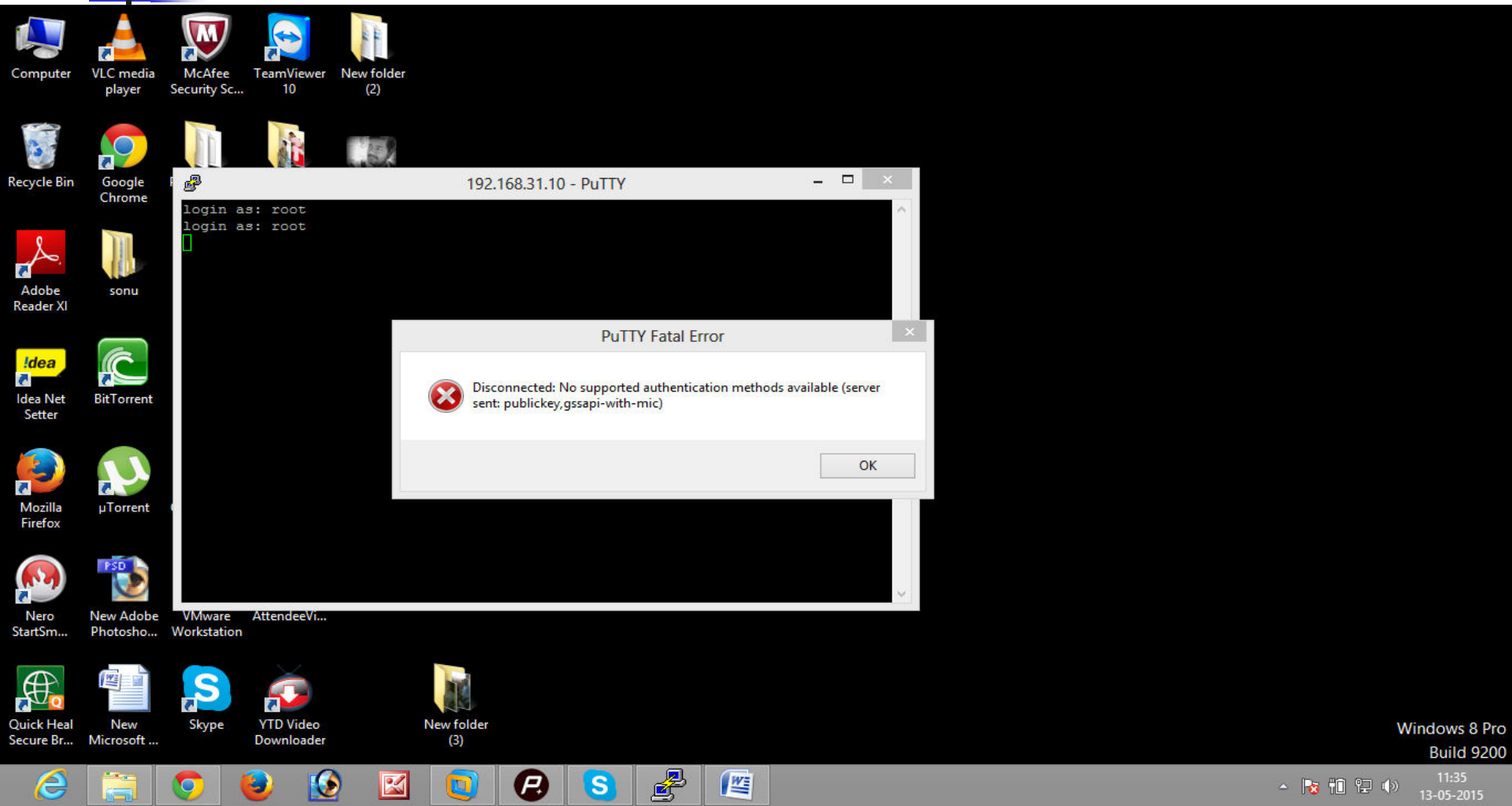
SSH Configuration

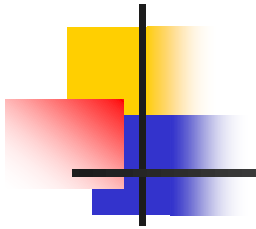
- create two users and assign individual passwords
  useradd user1
  useradd user2
- ssh configuration file
  vi /etc/ssh/sshd_config
- Edit the configuration setting and restart the service
  service network restart



- Check the value of **PasswordAuthentication** directive.
- In order to accept local user password base authentication it must be set to **yes**.
- Set it to **yes** if it is set to **no** and save the file.
- Restart the service if you have made any change in **sshd_config**

  service sshd restart


Computer
VLC media player
McAfee Security Sc...
TeamViewer 10
New folder (2)
Recycle Bin
Google Chrome
Adobe Reader XI
sonu
Idea Net Setter
BitTorrent
Mozilla Firefox
µTorrent
Nero StartSm...
New Adobe Photosho...
VMware Workstation
AttendeeVi...
Quick Heal Secure Br...
New Microsoft ...
Skype
YTD Video Downloader
New folder (3)
192.168.31.10 - PuTTY
login as: root
login as: root
PuTTY Fatal Error
Disconnected: No supported authentication methods available (server sent: publickey,gssapi-with-mic)
OK
Windows 8 Pro
Build 9200
11:35
13-05-2015

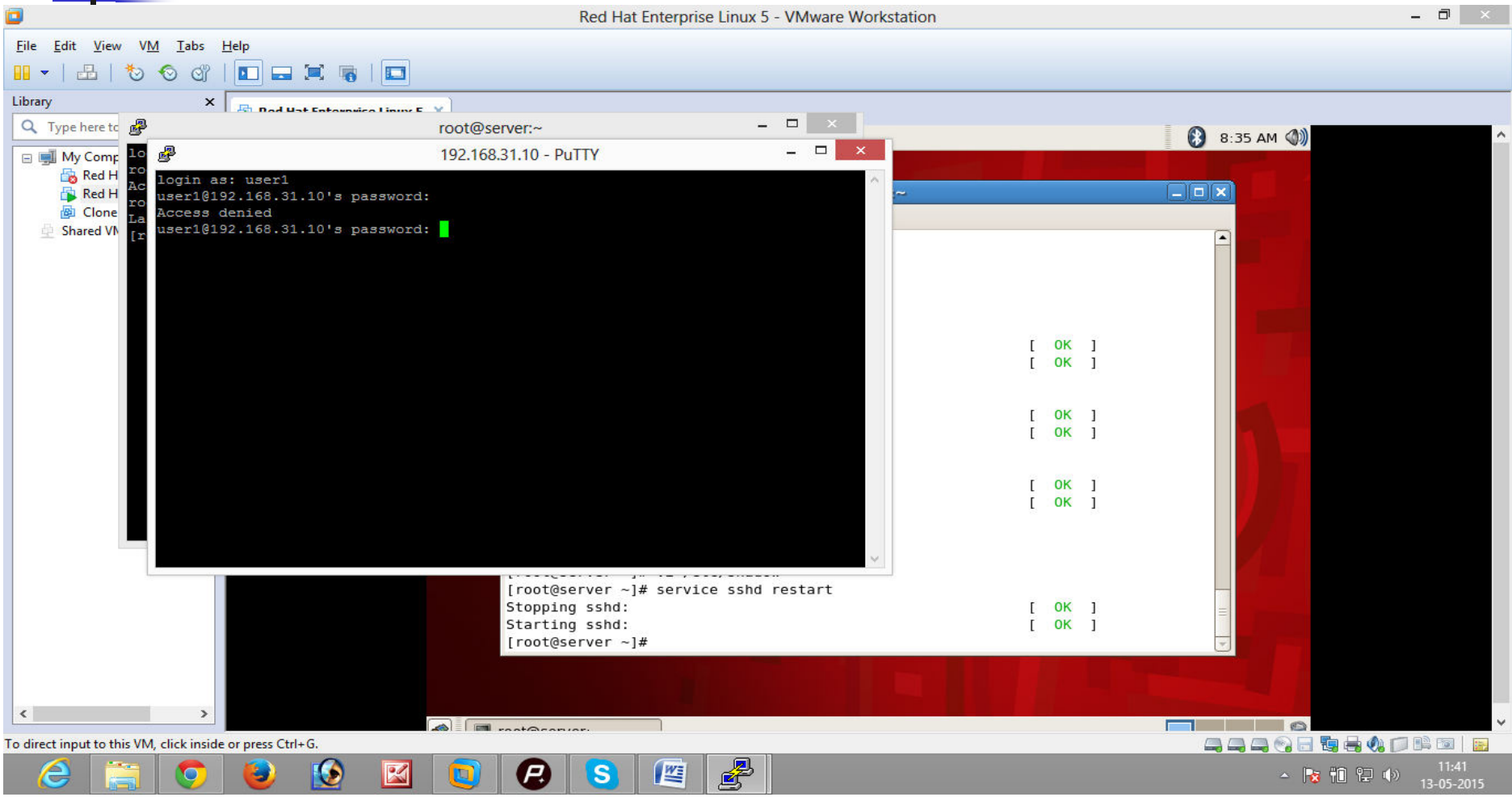


- Following additional directives can be added to **/etc/sshd/sshd_config** file in order to make the ssh server more restrictive.
- **Block empty passwords**

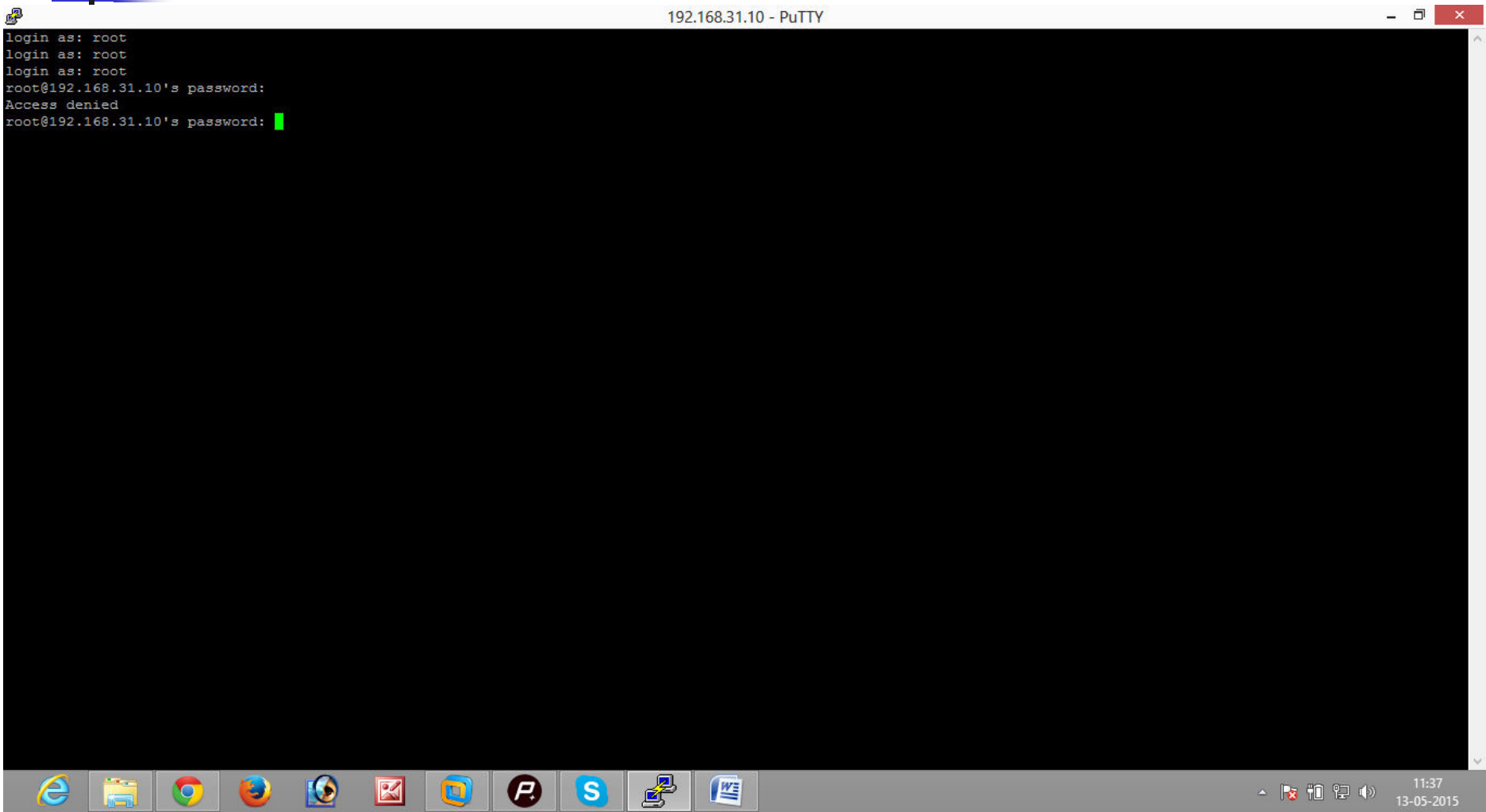
  PermitEmptyPasswords no
- service sshd restart
- **Block root user to log on the system using ssh**.

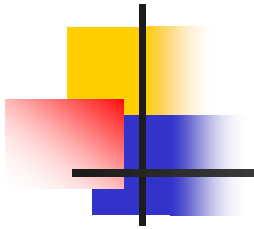
  #PermitRootLogin no

Block empty passwords



Block root user to log on the system using ssh.

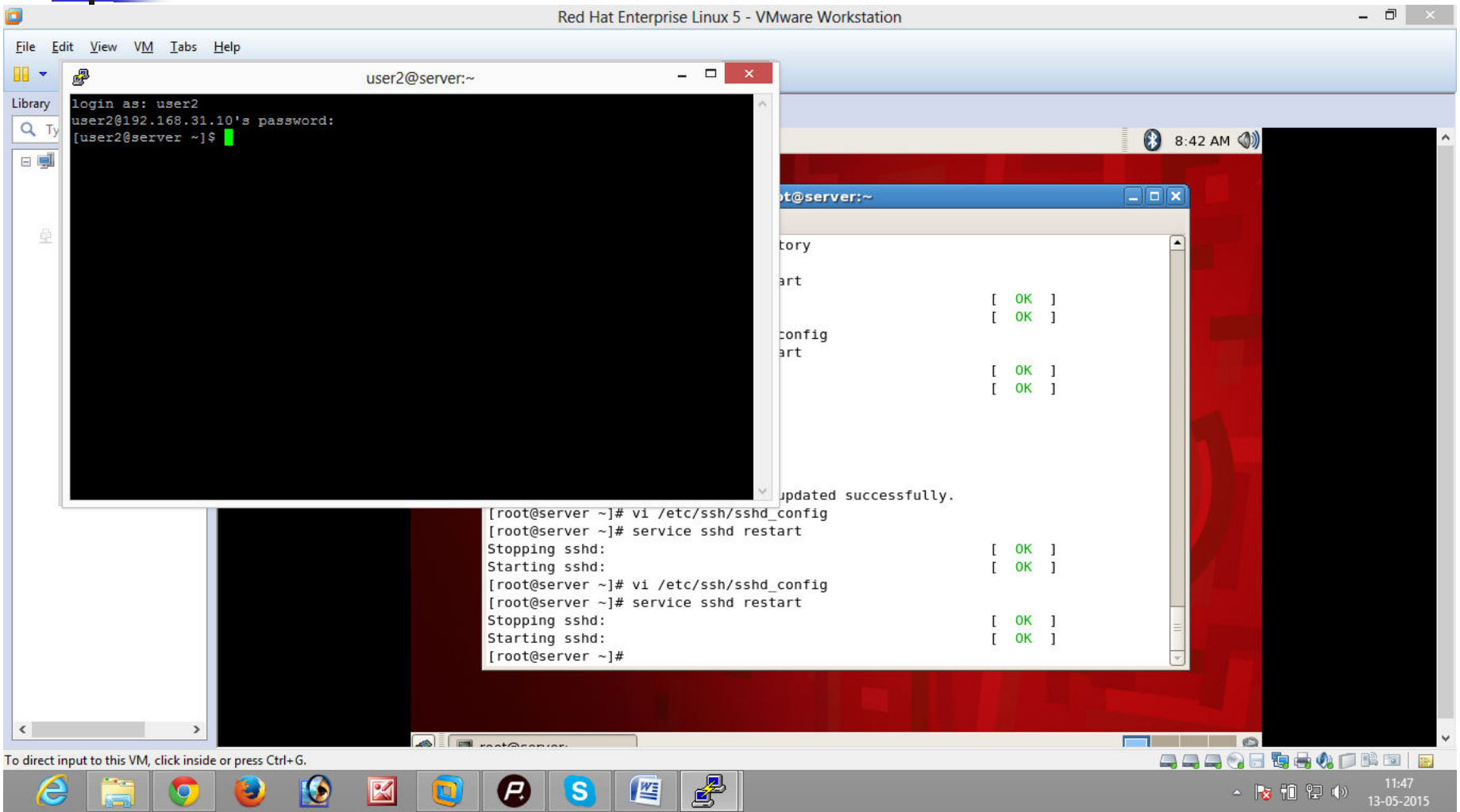




- In addition you can restrict the access to users. In this case all users except ‘nitin' are allowed to connect to the SSH server.
- add in /etc/ssh/sshd_config

  DenyUsers nitin sukesh
- Restart ssh service
- #service sshd restart

Deny users


Red Hat Enterprise Linux 5 - VMware Workstation
File Edit View VM Tabs Help
Library
user2@server:~
login as: user2
user2@192.168.31.10's password:
[user2@server ~]$
8:42 AM
t@server:~
tory
art
[OK]
[OK]
config
art
[OK]
[OK]
updated successfully.
[root@server ~]# vi /etc/ssh/sshd_config
[root@server ~]# service sshd restart
Stopping sshd: [OK]
Starting sshd: [OK]
[root@server ~]# vi /etc/ssh/sshd_config
[root@server ~]# service sshd restart
Stopping sshd: [OK]
Starting sshd: [OK]
[root@server ~]#
To direct input to this VM, click inside or press Ctrl+G.
11:47
13-05-2015