Linux/Unix Filesystems and Directories

Several major directories are associated with all modern Unix/Linux operating systems. **These directories organize user files, drivers, kernels, logs, programs, utilities, and more into** different categories. The standardization of the FHS makes it easier for users of other Unix-based operating systems to understand the basics of Linux. Every FHS starts with the root directory, also known by its label, the single forward slash (/). All of the other directories shown in Table are subdirectories of the root directory. Unless they are mounted separately, you can also find their files on the same partition as the root directory.

/	The root directory, the top-level directory in the FHS. All other directories are subdirectories of root, which is always mounted on some partition. All directories that are not mounted on a separate partition are included in the root directory's partition.
/bin	Essential command line utilities. Should not be mounted separately; otherwise, it could be difficult to get to these utilities when using a rescue disk.
/boot	Includes Linux startup files, including the Linux kernel. Can be small; 16MB is usually adequate for a typical modular kernel. If you use multiple kernels, such as for testing a kernel upgrade, increase the size of this partition accordingly.
/etc	Most basic configuration files.
/dev	Hardware and software device drivers for everything from floppy drives to terminals. Do not mount this directory on a separate partition.
/home	Home directories for almost every user.
/lib	Program libraries for the kernel and various command line utilities. Do not mount this directory on a separate partition.
/mnt	The mount point for removable media, including floppy drives, CD-ROMs, and Zip disks.
/opt	Applications such as WordPerfect or StarOffice.
/proc	Currently running kernel-related processes, including device assignments such as IRQ ports, I/O addresses, and DMA channels.
/root	The home directory of the root user.
/sbin	System administration commands. Don't mount this directory separately.
/tmp	Temporary files. By default, Red Hat Linux deletes all files in this directory periodically.
/usr	Small programs accessible to all users. Includes many system administration commands and utilities.
/var	Variable data, including log files and printer spools.

Types of Files Used by Linux

When working with Linux, you need to be aware of the fact that there are a number of different file types used by the file system. This is another area where the Linux file system differs significantly from the Windows file system. With a Windows file system you basically have two entry types in the file system:

- **Directories**
- **Files**

Granted, you can have normal files, hidden files, shortcut files, word processing files, executable files, and so on. However, these are all simple variations of the basic file when working with Windows.
With Linux, however, there are a variety of different file types used by the file system. These include the file types shown in Table

File Type	Description
Regular files	These files are similar to those used by the file systems of other operating systems—for example, executable files, OpenOffice.org files, images, text configuration files, etc.
Links	These files are pointers that point to other files in the file system.
FIFOs	FIFO stands for First In First Out. These are special files used to move data from one running process on the system to another. A FIFO file is basically a queue where the first chunk of data added to the queue is the first chunk of data removed from the queue. Data can only move in one direction through a FIFO.
Sockets	Sockets are similar to FIFOs in that they are used to transfer information between sockets. With a socket, however, data can move bi-directionally.

Some of the Configuration Files in /etc Directory that you should remember

File	Function
/etc/fstab	Lists the partitions and file systems that will be automatically mounted when the system boots.
/etc/group	Contains local group definitions.
/etc/grub.conf	Contains configuration parameters for the GRUB bootloader (assuming it's being used on the system).
/etc/hosts	Contains a list of hostname-to-IP address mappings the system can use to resolve hostnames.

/etc/inittab	Contains configuration parameters for the init process.
/etc/init.d/	A subdirectory that contains startup scripts for services installed on the system. On a Fedora or Red Hat system, these are located in /etc/rc.d/init.d.
/etc/modules.conf	Contains configuration parameters for your kernel modules.
/etc/passwd	Contains your system user accounts.
/etc/shadow	Contains encrypted passwords for your user accounts.
/etc/X11/	Contains configuration files for X Windows.

Virtual Consoles

A virtual console is a command line where you can log into and control Linux. As RHEL is a multi terminal operating system, you can log into Linux, even with the same user ID, several times. It's easy to open a new virtual console. Just use the appropriate ALT-function key combination. For example, pressing ALT-F2 brings you to the second virtual console. You can switch between adjacent virtual consoles by pressing ALT-RIGHT ARROW or ALT-LEFT ARROW. For example, to move from virtual console 2 to virtual console 3, press ALT-RIGHT ARROW.

You can switch between virtual terminals by just press the **ALT+CTRL+Funcation key** combinations.

ALT + CTRL + F1 for terminal 1
ALT + CTRL + F2 for terminal 2
ALT + CTRL + F3 for terminal 3
ALT + CTRL + F4 for terminal 4
ALT + CTRL + F5 for terminal 5
ALT + CTRL + F6 for terminal 6
ALT + CTRL + F7 for terminal 7

Terminal 7 is by default graphic mode beside it all six terminal are CLI based. Open any terminial by press ALT+CTRL+F1 key combinations. root account is automatically created when we install Linux.

Type **root** on login name and press enter key, now give password (no asterisk character like window to guess the password length) When you login from root account you will get # sign at command prompt , and when you login from normal user you will get **$** prompt.

```
Red Hat Enterprise Linux Server release 5 (Tikanga)
Kernel 2.6.18-8.el5 on an i686

Server login: root
Password:
Last login: Sat Dec 19 23:59:53 on tty1
[root@Server ~]# _
```

#clear

This command is used to clear the screen.You have three options to logout .

Press CTRL+D
#exit
#logout

All three commands perform same task.

#pwd
/root

```
[root@Server ~]# pwd
/root
[root@Server ~]# ls
anaconda-ks.cfg  install.log  install.log.syslog  test
[root@Server ~]# ls -a
.                .bash_history  .bashrc   .gconfd       install
..               .bash_logout   .cshrc    .gnome2       .lesshst
anaconda-ks.cfg  .bash_profile  .gconf    install.log   .tcshrc
[root@Server ~]# ls -l
total 68
-rw------- 1 root root   801 Dec 13 17:55 anaconda-ks.cfg
-rw-r--r-- 1 root root 40081 Dec 13 17:55 install.log
-rw-r--r-- 1 root root  4950 Dec 13 17:54 install.log.syslog
drwxr-xr-x 2 root root  4096 Dec 20 00:22 test
[root@Server ~]# ll
total 68
-rw------- 1 root root   801 Dec 13 17:55 anaconda-ks.cfg
-rw-r--r-- 1 root root 40081 Dec 13 17:55 install.log
-rw-r--r-- 1 root root  4950 Dec 13 17:54 install.log.syslog
drwxr-xr-x 2 root root  4096 Dec 20 00:22 test
[root@Server ~]# _
```

Print working directory command will tell you about current location from / partition.

#ls

ls command will list the object in directory. All directory are listed in blue color while files are shown white color.

#ls –a

Normal ls command will not list the hidden files. If you want to list the hidden file use –a switch with ls command to list the hidden files.

#ls –l

Ls command with –l switch will list the objects in long formats . we will discuss more about –l switch in coming sections.

#ll

Same as ls –l . First and major task for any system administrator is user managements. For testing purpose you can perform all task with root account but in real life root account is used for administrative purpose only. Lest create a normal user account for further practical.

#useradd [user name]

Useradd command is used to create user. Several advance options are used with useradd command but you will learn about them in coming article.

```
[root@Server ~]# useradd vinita
[root@Server ~]# passwd vinita
Changing password for user vinita.
New UNIX password:
BAD PASSWORD: it is WAY too short
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[root@Server ~]#
```

#passwd [user name]

In linux no user can be login without password. passwd command is used to assign password for any user. Do not execute this command without user name otherwise it will change root password.

Basic Linux commands cp mv rm mkdir cat cd command example

Command Syntax

$mkdir [directory name]

mkdir command is used to create new directory. Let's create a example directory.

$mkdir example
$ls
example

now create a file. Syntax for creating file is

$cat > [file name]

This command can be used in three way to see the matter of file, to create a new file or to append the matter of file.

$cat [file name] ---------- To see the matter of file
$cat > [file name]---------- To create a file
$cat >> [file name]---------- To append the matter of file

Be little bit careful while using cat > command to create new files. If you accidently used this command with existing file it will overwrite the matter of file. Use CTRL+D to save the matter of file.
Different use of cat command

$cat > test
This is test of file
$cat test
This is test of file
$cat >> test
This is second line
$cat example
This is test of file
This is second line in test file
$cat > test
Now file will over write
$cat test
Now file will overwrite

```
[vinita@Server ~]$ cat > test
this is a test file
[vinita@Server ~]$ cat test
this is a test file
[vinita@Server ~]$ cat >> test
second line in test file
[vinita@Server ~]$ cat test
this is a test file
second line in test file
[vinita@Server ~]$ cat > test
this time matter will overwrite
as we open an existing file
[vinita@Server ~]$ cat test
this time matter will overwrite
as we open an existing file
[vinita@Server ~]$ _
```

$cd [destination directory path]

It is easy to change directories in Linux. Just use cd and name the absolute path of the desired directory. If you use the relative path, just remember that your final destination depends on the present working directory.

```
[vinita@Server ~]$ mkdir example
[vinita@Server ~]$ ls
example  test
[vinita@Server ~]$ cd example
[vinita@Server example]$ pwd
/home/vinita/example
[vinita@Server example]$ cd ..
[vinita@Server ~]$ pwd
/home/vinita
[vinita@Server ~]$ _
```

as you can see in the output of ls command **file** is in white color and **directory** is in blue color.

There are two path concepts associated with Linux directories: absolute paths and relative paths.
An absolute path describes the complete directory structure based on the top level directory, root (/).
A relative path is based on the current directory. Relative paths do not include the slash in front. The difference between an absolute path and a relative one is important To know more about path and directory structure

pwd

In many configurations, you may not know where you are relative to the root (/) directory. The pwd command, which is short for print working directory, can tell you, relative to root (/). Once you know where you are, you can determine whether you need to move to a different directory.

$cd ..

this command is used to exit from current directory.

cp

The cp (copy) command allows you to take the contents of one file and place a copy with the same or different name in the directory of your choice. For example, the ***cp file1 file2*** command takes the contents of file1 and saves the contents in file2. One of the dangers of cp is that it can easily overwrite files in different directories, without prompting you to make sure that's what you really wanted to do.

```
[vinita@Server ~]$ ls
example  test
[vinita@Server ~]$ cp test example
[vinita@Server ~]$ ls
example  test
[vinita@Server ~]$ cd example
[vinita@Server example]$ ls
test
[vinita@Server example]$ cd ..
[vinita@Server ~]$ ls
example  test
[vinita@Server ~]$ mv test example
[vinita@Server ~]$ ls
example
[vinita@Server ~]$ cd example
[vinita@Server example]$ ls
test
[vinita@Server example]$ cd ..
[vinita@Server ~]$ _
```

mv

While you can't rename a file in Linux, you can move it. The mv command essentially puts a different label on a file. For example, the **mv file1 file2** command changes the name of file1 to file2. Unless you're moving the file to a different partition, everything about the file, including the inode number, remains the same.

rm

rm command is used to remove the regular files. It will ask before removing files.

rmdir

it will remove the empty directory. If directory is full then use **rm –rf [directory name]**

halt
This command shuts down the operating system, but can only be run by the root user.

#halt

reboot
This command shuts down and restarts the operating system. It also can only be run by root.

#reboot [will perform simple reboot]
#reboot -f [will perform fast reboot]

init 0
This command also shuts down the operating system, and can only be run by your root user.

#init 0

init 6 This command also shuts down and restarts the operating system. It also can only be run by root

#init 6

man
This command opens the manual page for the command or utility specified. The man utility is a very useful tool. If you are unsure how to use any command, use man to access its manual page. For example, you could enter **man ls** at the shell prompt to learn how to use the ls utility.

#man ls

info
The info utility also displays a help page for the indicated command or utility. The information displayed with info command will be in-depth than that displayed in the man page for the same command.

info ls

su
This command switches the current user to a new user account. For example, if you're logged in as **vickey** and need to change to user account to **vinita,** you can enter su **vinita** at the shell prompt. This command is most frequently used to switch to the superuser root account.
In fact, if you don't supply a username, this utility assumes that you want to change to the root account. If you enter **su -**, then you will switch to the root user account and have all of root's

environment variables applied.
This command require password of the user you want switch.

Looking for Files

There are two basic commands used for file searches: **find** and **locate**

find

The find command searches through directories and subdirectories for a desired file. For example, if you wanted to find the directory with the **grub.conf** linux boot loader file, you could use the following command, which would start the search in the top-level root (/) directory:

```
# find / -name grub.conf
```

But this search took several minutes to get it task done. Alternatively, if you know that this file is located in the /etc subdirectory tree, or /boot/grub/grub.conf you could start in that directory with the following command:

```
# find /etc -name grub.conf
```

locate

If this is all too time-consuming, RHEL 5 includes a default database of all files and directories. Searches with the locate command are almost instantaneous. And locate searches don't require the full file name. The drawback is that the locate command database is normally updated only once each day, as documented in the /etc/cron.daily/mlocate.cron script.

cp

The cp (copy) command allows you to take the contents of one file and place a copy with the same or different name in the directory of your choice. For example, the cp file1 file2 command takes the contents of file1 and saves the contents in file2. One of the dangers of cp is that it can easily overwrite files in different directories, without prompting you to make sure that's what you really wanted to do.

mv

While you can't rename a file in Linux, you can move it. The mv command essentially puts a different label on a file. For example, the mv file1 file2 command changes the name of file1 to file2. Unless you're moving the file to a different partition, everything about the file, including the inode number, remains the same.

echo

This command is used to echo a line of text on the screen. It's frequently used to display environment variables. For example, if you wanted to see the current value of the PATH variable, you could enter
echo $PATH

top

This command is a very useful command that displays a list of all applications and processes currently running on the system. You can sort them by CPU usage, memory usage, process ID number, and which user owns them

which

This command is used to display the full path to a shell command or utility. For example, if you wanted to know the full path to the **ls command,** you would enter
which ls

whoami

This command displays the username of the currently logged-in user.

netstat

This command displays the status of the network, including current connections, routing tables, etc

route

This command is used to view or manipulate the system's routing table.

ifconfig

This command is used to manage network boards installed in the system. It can be used to display or modify your network board configuration parameters. This command can only be run by the root user.

Once you become familiar with these basic command which you need to perform RCHE skill you are ready to move our next series of article focused on RHCE exam

Basic RHCE commands using help for commands

In this article I will show some basic system administration related task which a normal user can perform. To complete this assignment login from normal user which we created in our first assignment.

```
Red Hat Enterprise Linux Server release 5 (Tikanga)
Kernel 2.6.18-8.el5 on an i686

Server login: vinita
Password:
[vinita@Server ~]$ pwd
/home/vinita
[vinita@Server ~]$ _
```

How to count line word and character form a file

$wc [file name]

This command is used to count line words and character of file. Out will first show the line number word and in the end characters.

$wc test
2 4 23 test

```
[vinita@Server ~]$ wc test
 2  4 23 test
[vinita@Server ~]$ _
```

In this example there are 2 lines 4 words and 23 character in test file.

how to display top and bottom line form files

$head –n [number] [file name]

head command is used to display specific number of line from top for given file.

$head –n 4 test

For example this command will show the 4 top most line of file test.

```
[vinita@Server ~]$ head -n 4 test
First line
Second line
third line
four line
[vinita@Server ~]$ tail -n 3 test
seven line
eight line
nine line
[vinita@Server ~]$ _
```

$tail –n [number] [file name]

tail command will display the specific number of line form bottom for given file.

$tail –n 3 test

This command will display the 3 line from bottom of file test

how to find wrong spelling and correct them

$spell [file name]

spell command will display the wrong spelling of files.

$spell test

This command will display the wrong spelling of test file. If there is no spelling mistake no out will show.

```
[vinita@Server ~]$ spell test
spalling
worng
[vinita@Server ~]$ aspell check test_
```

$aspell check [file]

This command is used to correct the spelling related mistake in any given files.

$aspell check test

This command will show the all wrong spelling from test file and there possible corrections. To use correction just press the number shown in front of words.

how display logged in user information and terminal number

$who am i

This command is use to display the username of currently logged.

$who

This command will display all the user currently logged in all terminals.

```
[vinita@Server ~]$ who am i
vinita   tty2         2009-12-20 00:23
[vinita@Server ~]$ who
root     tty1         2009-12-20 00:00
vinita   tty2         2009-12-20 00:23
b        tty3         2009-12-19 22:33
[vinita@Server ~]$ tty
/dev/tty2
[vinita@Server ~]$ date
Sun Dec 20 00:37:35 IST 2009
[vinita@Server ~]$ _
```

$tty

This command is used to display the terminal number of currently logged in terminals.

how to display date time and calendar

$cal

This command will display the calendar of current month. You can see the calendar of any specific year also.

$cal 2010 |more

```
[vinita@Server ~]$ cal
   December 2009
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

[vinita@Server ~]$ cal 2010 |more_
```

This will display the calendar of year 2010. As output will be more than a page so use more switch with commands.

$date

This will the current system times and date.

how to use calculator

$bc

```
[vinita@Server ~]$ bc
bc 1.06
Copyright 1991-1994, 1997, 1998, 2000 Free Software
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
12+23
35
[vinita@Server ~]$ _
```

This command will launch the calculator at command prompt. Write down your calculations and press enter to get the answer. Use CTRL+D key combination to exit from calculator.

how to get help about commands

$info [command]

info command is used to get help about any commands.

$info cat

This will display the help about cat commands. Generally output of info commands is more a then a page. You can quit form this out by just pressing q.

$command - - help

This help options is really very useful when do not want to be read full manual page for help. This will provide very basic help like which switch will work with this command.

$cat - - help

This will show the available switch cat command and a very brief descriptions about these switches.

$man [command]

If want to read the detail about any command use this command. This will give you the complete detail about commands.

$man cat

This command will give the complete details about the cat commands including switches and their usages. Use q to quit from the output of this commands.

$less [file]

When you have a file more than one pages use less command to read the output of file despite of using cat command with more switch. As with more switch you cannot scroll the text in both directions.

$cat [file] |more

If you have a file more than one page than use |more switch with cat to read the output. Without this switch matter of file will scroll too fast that you will see only texts of last pages.

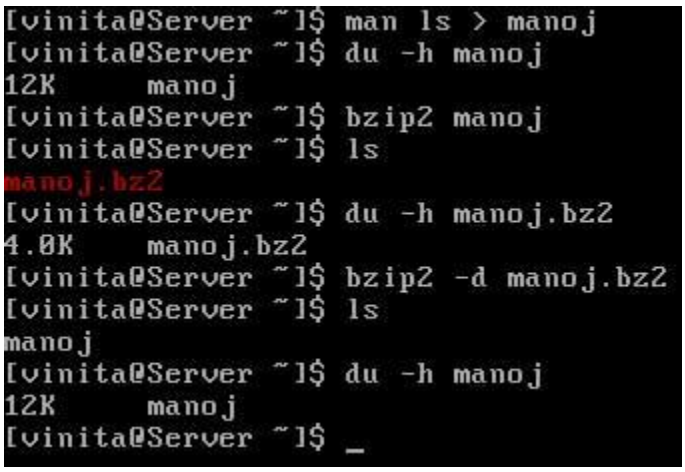
How to compress files to save disk space?

Create a large file and check how much disk space is consumed by this file

$man ls > manoj
$du –h manoj
12k manoj

File manoj is using 12k space on hard disk. For exam prospective you should familiar with two compress utilities.

$bzip2 [file name] **{command syntax}**
$bzip2 manoj
$ls
$du -h manoj.bz2
4k manoj.bz2

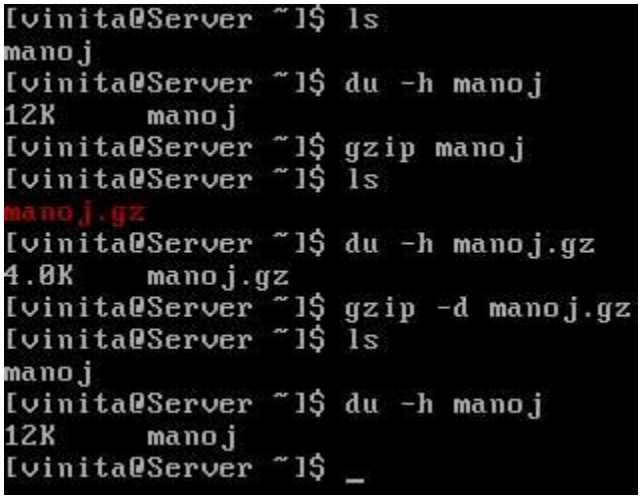


To decompress file

$bzip2 –d manoj.bz2 $ls manoj

as you can show file has been decompressed. Now use other utility to compress the file.

$gzip manoj
$ls
manoj.gz
$du –h manoj.gz
4k manoj.gz
$gzip –d manoj.gz
$ls
manoj



Linux system administrations commands

In our last few assignments you learnt system administration related task which a normal user can perform. In this assignment I will direct you some handy task for root user. To accomplish this assignment login form root account.

Know how much space is consumed

#du

This command will show the usages of disk by files and folder. Output of this command show in bytes. To show it in KB use –h switch.

#du -h [file name]

To know that how much space is consumed by any specific file. For example

#du –h test
12 Kb test

Command is showing that size of test file is 12 kb.

```
[root@Server vinita]# du
16      .
[root@Server vinita]# ls
test
[root@Server vinita]# du -h test
12K     test
[root@Server vinita]# df /
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/sda2              9920624   2557032   6851524  28% /
[root@Server vinita]#
```

Know how much space is available

#df [partition]

df command is used to know the available space on any given partitions. For example to know available space on / partition use this command

#df /

How to find any files

#find [where to find] – name [what to find]

find command is used to find any object in linux. For searching object you can also use locate command but locate command is based on mlocate database. For example to find vinita directory on entire linux use

#find / -name vinita

```
[root@Server vinita]# find / -name vinita
/home/vinita
/root/vinita
/var/run/console/vinita
[root@Server vinita]# find /home -name vinita
/home/vinita
[root@Server vinita]# _
```

Or to find only in /home partition use

#find /home –name vinita

How to use history and clear it

history utility keeps a record of the most recent commands you have executed. The commands are numbered starting at 1, and a limit exists to the number of commands remembered—the default is 500. To see the set of your most recent commands, type history on the command line and press ENTER. A list of your most recent commands is then displayed, preceded by a number.

#history
#history –c

Use –c switch with history command to clear the history.

Check running process and terminate

#ps

The ps (process status) command is used to provide information about the currently running processes, including their process identification numbers (PIDs). A process, also referred to as a task, is an running instance of a program. Every process is assigned a unique PID by the system

```
[root@Server vinita]# ps
  PID TTY          TIME CMD
 4288 tty1     00:00:00 bash
 4845 tty1     00:00:00 ps
[root@Server vinita]# ps -ef_
```

#ps –ef

The -e option generates a list of information about every process currently running. The -f option generates a listing that contains fewer items of information for each process than the -l option. Among the columns displayed by ps -ef, UID contains the username of the account that owns

the process (which is usually the same user that started the process) and STIME displays the time the process started, or the starting date if it started more than 24 hours ago.

#kill [ps number]

The kill command is used on Linux to terminate processes without having to log out or reboot the computer. Thus, it is particularly important to the stability of such systems. Each process is automatically assigned a unique process identification number (PID) when it is created for use by the system to reference the process.

The only argument that is required is a PID, and as many PIDs as desired can be used in a single command. Typically no signal or option is used. Thus, if it is desired to terminate a process with a PID of 485, the following will usually be sufficient:

kill 485

#pstree

pstree command displays the processes on the system in the form of a tree diagram. It differs from the much more commonly used (and more complex) ps program in a number of respects, including that the latter shows the processes in a list rather than a tree diagram but provides more detailed information about them.

how check user set environment

#env

env command will display the environment set for user. A brief description about this output is

EDITOR	Name of editor used.
HOME	The directory that you are first logged into
SHELL	The program you run as your command-line interpreter.
TERM	The type of terminal emulation used
PATH	Listing of directories searched when logging on
MAIL	Location of where the mail is stored
MANPATH	Location of your Manuals.

LOGNAME	The login name
TZ	Time zone of computer

how to check CPU run time status

#top

When you need to see the running processes on your Linux in real time, you have top as your tool for that. top also displays other info besides the running processes, like free memory both physical and swap. use q to quit from the output of top commands.

how to set alias for commands

#alias san=clear

alias command is used to set alias with any command. Mostly alias is used in shell scripting. In our example we set an alias for clear command. Now whenever you need to clear the screen type san instead of clear command. This will work till only you are logged in if want to set alias permanently then do editing in user profile files.

#uname –a

uname command is used to gather the system information's. you can use several switches with commands. Few of them are.

```
[root@Server vinita]# uname -a
Linux Server 2.6.18-8.el5 #1 SMP Fri Jan 26 14:15:21 EST 2007 i686 i686 i386 GNU
/Linux
[root@Server vinita]# uname
Linux
[root@Server vinita]# uname -r
2.6.18-8.el5
[root@Server vinita]# _
```

-a, --all
print all information, in the following order:
-s, --kernel-name
print the kernel name
-n, --nodename
print the network node hostname
-r, --kernel-release
print the kernel release
-v, --kernel-version
print the kernel version
-m, --machine
print the machine hardware name
-p, --processor
print the processor type

-i, --hardware-platform
print the hardware platform
-o, --operating-system
print the operating system
--help
display this help and exit
--version
output version information and exit

how to send message to all logged in user

#wall

wall sends a message to everybody logged in . The message can be given as an argument to wall, or it can be sent to wall's standard input. When using the standard input from a terminal, the message should be terminated with the EOF key (usually Control-D). The length of the message is limited to 20 lines.

To shutdown the system

#halt –p
#init 0

To reboot system

#reboot –f
#init 6
#reboot

System administrations User managements

Linux files responsible for User managements

```
/etc/shadow      store all the Linux password in MD5 encryptions format
/etc/passwd      store all user related information's
/etc/group       store all group related information's
```

back-up files responsible for User managements

In this assignment we will modify these files. So it's better to take back-up before doing this assignment because your little mistake can crash Linux systems.

#mkdir /backup
#cp /etc/passwd /backup
#cp /etc/group /backup
#cp /etc/shadow /backup

```
[root@Server /]# mkdir /backup
[root@Server /]# cp /etc/passwd /backup
[root@Server /]# cp /etc/shadow /backup
[root@Server /]# cp /etc/group /backup
[root@Server /]# cd /backup
[root@Server backup]# ls
group  passwd  shadow
[root@Server backup]# _
```

Create a simple user

userdd is used to create user. Your task is to learn what exactly happens in these files when a new user is added. First observe the last line for these files.

#cat /etc/passwd |more
#cat /etc/shadow |more
#cat /etc/group |more

Now add a simple user.

#useradd vinita
#passwd vinita

```
[root@Server backup]# useradd vinita
[root@Server backup]# passwd vinita
Changing password for user vinita.
New UNIX password:
BAD PASSWORD: it is WAY too short
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[root@Server backup]#
```

Now read these files again with cat command alternate you can use |grep switch to filter the output

#cat /etc/passwd |grep vinita
#cat /etc/shadow |grep vinita
#cat /etc/group |grep vinita
cd /home
#ls –ld vinita

```
[root@Server backup]# cat /etc/passwd |grep vinita
vinita:x:503:504::/home/vinita:/bin/bash
[root@Server backup]# cat /etc/shadow |grep vinita
vinita:$1$Mwn3J3ls$.651UvsB6cEdyroZcJ.8w1:14597:0:99999:7:::
[root@Server backup]# cat /etc/group |grep vinita
vinita:x:504:
[root@Server backup]# cd /home
[root@Server home]# ls -ld vinita
drwx------ 2 vinita vinita 4096 Dec 20 01:14 vinita
[root@Server home]# _
```

User's entry in passwd

All these files are changed when a user is created In passwd files entries are in following formats separated by :

vinita users login name
x password required to login
503 unique user id
504 unique group id
/home/vinita users home directory
/bin/bash user shell

In shadow files entry is straight forwards. Whatever showing beside the user name is the password of user vinita in MD5 encrypt format.

User's entry in group

Whenever you create a normal user, users primary group form same name is automatically created. As you can verify by looking in /etc/group. 504 is the unique group id.

User's home directory

Same as group, users home directory is also created in /home partition and user get the ownership of this directory.

How to create a user without password.

```
[root@Server ~]# useradd nikki
[root@Server ~]# passwd -d nikki
Removing password for user nikki
passwd: Success
[root@Server ~]# _
```

To create a user without password use –d switch .

#useradd nikki
#passwd -d nikki

How to create a group.

To create group use groupadd commands. Group created by this command is called secondary group.

#groupadd test
#cat /etc/group |grep test

How to add user in groups

To add user in this group use usermod commands

#usermod –G test vinita

This command will make vinita user to member of test group.

How to delete secondary group

You can delete a group by groupdel commands

#groupdel test
#cat /etc/group |grep test

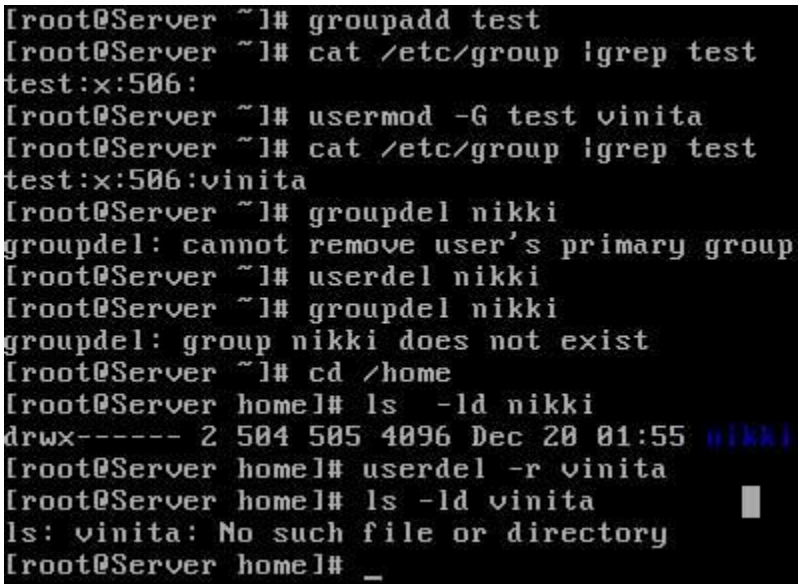
You cannot delete users primary group until user exist for example

#groupdel nikki

How to delete User

userdel command is used to delete user. When a users is deleted user's primary group will automatically be deleted.

#userdel nikki
#groupdel nikki
groupdel: group nikki does not exist.



Whenever you delete user with userdel command. entry of user will be removed from these files. But users home folder and mail folder will not be deleted. As you can see in image. If you want completely remove user including his home folder and mail folder use –r switch with userdel commands.

System administrations User profiles su sudo Shell operations

You discover that

- Files those are responsible for user and group managements
- How to create a normal user
- How to create user without password
- How to create bulk users and groups
- How to delete bulk user and groups

In this assignment we will discuss about user variables and profiles. User's session starting from his login to till exit is controlled by some profile files. These files are located in /etc/skel. When you create a new user script files from this directory are copied in user's home directory. There is only exceptions when user is created with –M switch or user home directoy is already exist.

```
[root@Server ~]# useradd -M vinita
[root@Server ~]# passwd vinita
Changing password for user vinita.
New UNIX password:
BAD PASSWORD: it is WAY too short
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[root@Server ~]# ls /home
a  b  c  babu  lost+found  mamu  nikki  temp
[root@Server ~]# mkdir /home/vinita
[root@Server ~]# chown vinita /home/vinita
[root@Server ~]# cp /etc/skel/.* /home/vinita
cp: omitting directory `/etc/skel/.'
cp: omitting directory `/etc/skel/..'
[root@Server ~]# ls -a /home/vinita
.  ..  .bash_logout  .bash_profile  .bashrc
[root@Server ~]# _
```

In such a situations you need to copy these file manually. These file are hidden and can be seen by –a switch with ls commands.

$ls –a

```
Red Hat Enterprise Linux Server release 5 (Tikanga)
Kernel 2.6.18-8.el5 on an i686

Server login: vinita
Password:
[vinita@Server ~]$ ls -a
.  ..  .bash_logout  .bash_profile  .bashrc
[vinita@Server ~]$ _
```

.bash_profile

```
[vinita@Server ~]$ cat .bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin

export PATH
[vinita@Server ~]$ _
```

This script file instructs user session to check .bashrc file for user aliases and functions. Further its set user command path . if you want add your own directory to your command path. Edit this file. For example user vinita wants her home directory should be check while excuting commands she can add this line in her .bash_profile files.

$vi .bash_profile
PATH=$PATH:$HOME/BIN:/home/vinita

.bashrc

```
[vinita@Server ~]$ cat .bashrc
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# User specific aliases and functions
[vinita@Server ~]$ _
```

This file is used to controls user variable and other profile during his login session. If you want to execute any command automatically on user logon set that command in this file. For example if user vinita wants to clear screen immediately after her login. She need to add clear command at end of this file.

$vi .bashrc

add your command only in the end of file
clear

With this file you can play a funny trick with your friends. Create a user and set exit command in his .bashrc files. Now ask your friends to login with this user. exit command will logout the user as soon as user will login and user will never will be able to login.

.bash_logout

```
[vinita@Server ~]$ cat .bash_logout
# ~/.bash_logout

clear
[vinita@Server ~]$ _
```

This file is used to clear the terminals after the exit of current user.

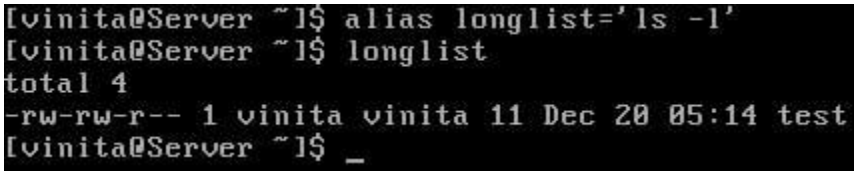
Aliases

The alias command is used to create another name for a command. The alias does not exactly replace the name of the command; it simply gives another name to that command. An alias command begins with the keyword alias and the new name for the command, followed by an equal sign and the command the alias will reference. *No spaces can be around the equal sign used in the alias command.* In the next example, list becomes another name for the ls command:

```
$ alias list=ls
$ ls
Report vickey nikki
$ list
Report vickey nikki
$
```

You can also use an alias to alternate for a command and its option, but you need to enclose both the command and the option within single quotes. Any command you alias that contains spaces must be enclosed in single quotes as well. In the next example, the alias longlist is set for command nls -l

```
$ alias longlist='ls -l'
```



Controlling some important Shell Operations

The BASH shell has several features that enable you to control the way different shell operations work. You need not know all these options for exam. But some hand operations you should always try in exam.

To stop logout form CTRL+D

Several commands in Linux are completed with CTRL+D. for example if you are making file form cat command the CTRL+D is used to save the files. And if you are using calculator on command prompt then CTRL+D is used to exit form calculators. But what if you pressed accidently CTRL+D two times, it will logout you from current session and you have login again.

```
$set –o ignoreeof
```

Now press CTRL+D and you will get a message “Use “logout” to leave the shell.

```
[vinita@Server ~]$ set -o ignoreeof
[vinita@Server ~]$ Use "logout" to leave the shell.
[vinita@Server ~]$ _
```

To stop overwriting of files

Other important shell operations are overwriting. How many times you have overwritten files. For example

$cat > test
Testing file
 $ls
test

now run this command once again

$cat > test
Old matter will overwrite without any message
$ls
$cat test
Old matter will overwrite without any message

Notice how easily Linux can overwrite file. To turnoff this shell feature

$set –o noclobber

Now whenever you will try to overwrite it will stop you with error message.

```
[vinita@Server ~]$ set -o ignoreeof
[vinita@Server ~]$ Use "logout" to leave the shell.
[vinita@Server ~]$ set -o noclobber
[vinita@Server ~]$ cat > test
first file
[vinita@Server ~]$ cat > test
-bash: test: cannot overwrite existing file
[vinita@Server ~]$ _
```

Whatever you set with –o option can be correct with + sign.

$set +o ignoreeof
Now again you can logout with CTRL+D.

Changing shell prompt

By default shell prompt show user name hostname and current working directory. You can change this prompt to following variable.

The following table lists the codes for configuring your prompt:

Prompt	Codes Description
\!	Current history number
\$	Use $ as prompt for all users except the root user, which has the # as its prompt
\d	Current date
\#	History command number for just the current shell
\h	Hostname
\s	Shell type currently active
\t	Time of day in hours, minutes, and seconds
\u	Username
\v	Shell version
\w	Full pathname of the current working directory
\W	Name of the current working directory
\\	Displays a backslash character
\n	Inserts a newline
\[\]	Allows entry of terminal-specific display characters for features like color or bold font
\nnn	Character specified in octal format

Granting root privilege to normal user

Generally in Linux, a system administrator does everything possible as a normal user. It's a good practice to use superuser privileges only when absolutely necessary. But one time when it's appropriate is during the Red Hat exams. Good administrators will return to being normal users when they're done with their tasks. Mistakes as the root user can disable your Linux system. There are two basic ways to make this work:

su
The superuser command, su, prompts you for the root password before logging you in with root privileges.

```
[vinita@Server ~]$ su
Password:
[root@Server vinita]# pwd
/home/vinita
[root@Server vinita]# useradd test
bash: useradd: command not found
[root@Server vinita]# exit
exit
[vinita@Server ~]$ su -
Password:
[root@Server ~]# pwd
/root
[root@Server ~]# useradd test
[root@Server ~]# _
```

su command without any arguments will ask for root password. By giving root password you will get root privilege. To execute any command you should know the exact path of command otherwise you get command not found error. Because you will not get root's command path. To get root's environments and command paths and home directory use – hyphen sign with su commands

Limiting Access to su

First, you will need to add the users who you want to allow access to the su command. Make them a part of the wheel group. By default, this line in /etc/group looks like:

wheel:x:10:root

You can add the users of your choice to the end of this line directly, with the usermod -G wheel [username] command, or with the Red Hat User Manager.

#usermod –G wheel vinita

Next, you will need to make your Pluggable Authentication Modules (PAM) look for this group. You can do so by activating the following command in your /etc/pam.d/su file:

auth required pam_wheel.so use_uid

sudo
The sudo command allows users listed in /etc/sudoers to run administrative commands. You can configure /etc/sudoers to set limits on the root privileges granted to a specific user.

```
[vinita@Server ~]$ who am i
vinita   tty2         2009-12-20 05:21
[vinita@Server ~]$ sudo /sbin/service network restart

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

Password:
Shutting down interface eth0:                              [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface eth0:
Determining IP information for eth0... done.
                                                           [  OK  ]
[vinita@Server ~]$ _
```

To use sudo commands you don't need to give root password. A user with appropriate right from /etc/sudoers can execute root privilege command form his own passwords.

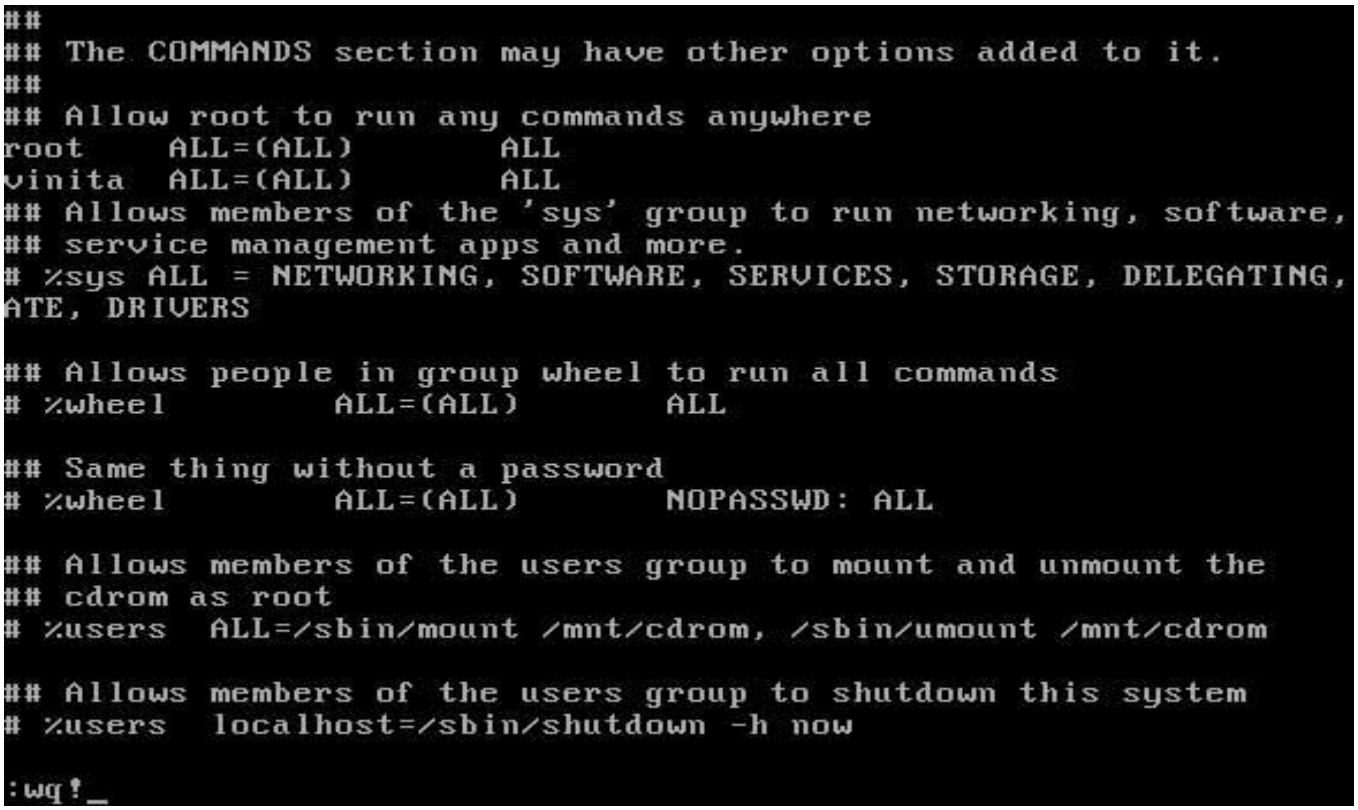
Red Hat Enterprise Linux provides some features that make working as root somewhat safer. For example, logins using the ftp and telnet commands to remote computers are disabled by default.

Limiting Access to sudo

You can limit access to the sudo command. Regular users who are authorized in /etc/sudoers can access administrative commands with their own password. You don't need to give out the administrative password to everyone who thinks they know as much as you do about Linux. To access /etc/sudoers in the vi editor, run the visudo command.

```
[root@Server ~]# vi /etc/sudoers_
```

From the following directive, the root user is allowed full access to administrative commands:

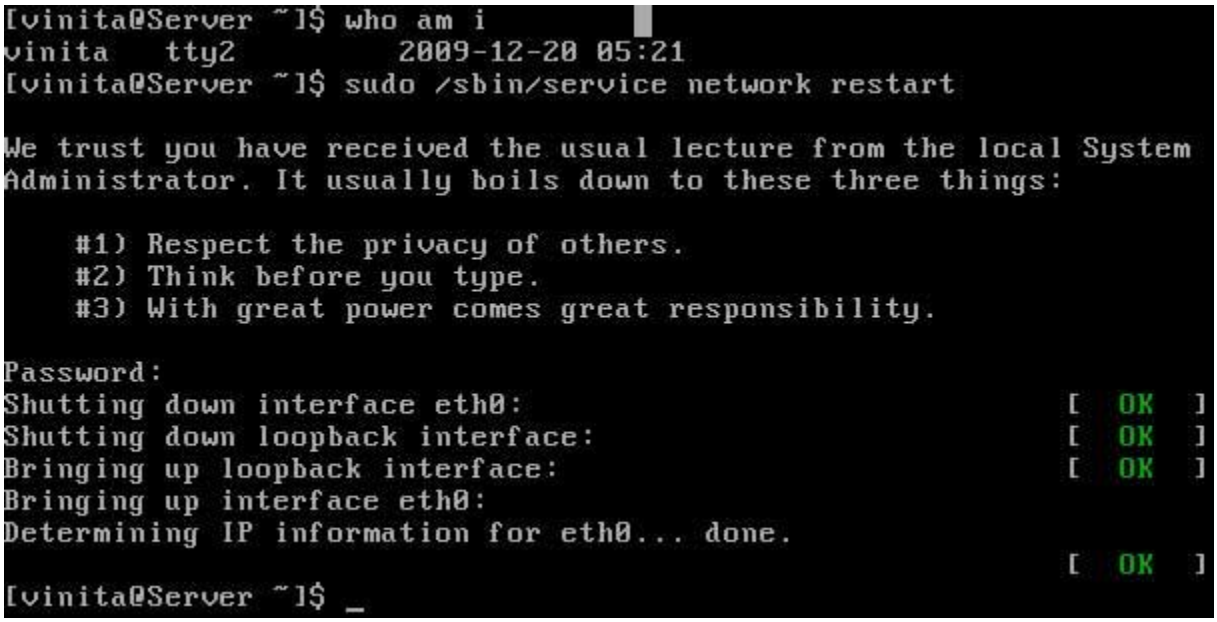


For example, if you want to allow user vinita full administrative access, add the following directive to /etc/sudoers:

root ALL=(ALL) ALL
vinita ALL=(ALL) ALL

In this case, all vinita needs to do to run an administrative command such as starting the network service from her regular account is to run the following command, entering her own user password (note the regular user prompt, $):

$ sudo /sbin/service network restart
Password:



You can even allow special users administrative access without a password. As suggested by the comments, the following directive in /etc/sudoers would allow all users in the wheel group to run administrative commands without a password:

%wheel ALL=(ALL) NOPASSWD: ALL

But you don't have to allow full administrative access. For example, if you want to allow those in the %users group to shut down the local system, you can activate the following directive:

%users localhost=/sbin/shutdown -h now

In our last assignment we discuss about user and group managements. You learnt about the files which are responsible for creating user and groups. You saw what exactly happens when we add new user in these files.

To add a new user, use the useradd command. The basic syntax is

useradd [username]

The username is the only information required to add a new user; however, for exam prospective you should know some additional command-line arguments for useradd. The useradd command creates the account, but the account is locked.

To unlock the account and create a password for the user, use the command passwd [username]. By default, the user's home directory is created and the files from /etc/skel/ are copied into it.

The two exceptions are if the –M option is used or if the home directory already exists.

We have already discussed about these two basic commands in our last article. If you haven't completed our last assignments we suggest you to review it before going with this article as it's the sequential of last assignments.

System administrations User managements Part 1

Create a user with additional command-line arguments.In this example you are going to assign home directory on other locations so first create it and same as create first desired user's secondary group.

#mkdir /test
#groupadd example
#useradd –u 700 –d /test/user1 –g example –c “testing user” –s /bin/sh –m user1
#passwd user1

```
[root@Server ~]# mkdir /test
[root@Server ~]# groupadd example
[root@Server ~]# useradd -u 700 -d /test/user1 -g example -c "testing user" -s /
bin/sh -m user1
[root@Server ~]# passwd user1
Changing password for user user1.
New UNIX password:
BAD PASSWORD: it is WAY too short
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[root@Server ~]# _
```

-c [fullname] Full name of the user (or a comment about the user).
If more than one word is needed, place quotation marks
around the value.
-d [directory] Home directory for the user. The default value is /home/[username]/.
-g [group] Default group for the user specified as a group name or group ID
number. The group name or GID must already exist. The default is
to create a private user group. If a private user group is not
created, the default is the users group.
-m Create a home directory for the user if it doesn't exist. Files from
/etc/skel/ are copied into the home directory.
-s [shell] Specify the user login shell for the user. The default shell if not
specified is /bin/bash.
-u [uid] Integer to use for the user ID. Must be unique unless -o is used.
Values less than 500 are reserved for system users.
-M Do not create a home directory for the user. By default, a home

directory is created unless this option is used or unless the directory already exists.

Now login form this user. And check where did this user logged in and why its shell prompt is looking different.

$pwd
/test/user1

```
Red Hat Enterprise Linux Server release 5 (Tikanga)
Kernel 2.6.18-8.el5 on an i686

Server login: user1
Password:
-sh-3.1$ pwd
/test/user1
-sh-3.1$ _
```

By default user gets bash sell prompts. But we modified this by –s switch and given user to /bin/sh shell. Now change user shell again

#usermod –s /bin/bash user1

```
[root@Server ~]# usermod -s /bin/bash user1
[root@Server ~]# _
```

Verify by login again from user1

```
Red Hat Enterprise Linux Server release 5 (Tikanga)
Kernel 2.6.18-8.el5 on an i686

Server login: user1
Password:
Last login: Sun Dec 20 04:13:40 on tty2
[user1@Server ~]$ _
```

How to manage bulk users

Consider a situation where you need to create more then thousand user. It will be really tedious task if you will do it by simple useradd commands. Here you have to switch to Linux shell scripts.

loop for creating user

for USER in _ _ _ _ _ _ _ _ _ _ _
> do
>useradd $USER
>echo _ _ _ _ |passwd --stdin $USER

>done

Example
(replace users *vinita nikkita niddhi sumit shweta vickey kaushal manoj jai* to your users)

for USER in vinita nikkita niddhi sumit shewta vickey kaushal manoj jai
> do
>useradd $USER
>echo friends |passwd --stdin $USER
>done

This simple for loop will create 9 users and set their defaults passwords to friends.

```
[root@Server ~]# for USER in vinita nikkita niddhi sumit shweta vickey kaushal m
anoj jai
> do
> useradd $USER
> echo friends |passwd --stdin $USER
> done
Changing password for user vinita.
passwd: all authentication tokens updated successfully.
Changing password for user nikkita.
passwd: all authentication tokens updated successfully.
Changing password for user niddhi.
passwd: all authentication tokens updated successfully.
Changing password for user sumit.
passwd: all authentication tokens updated successfully.
Changing password for user shweta.
passwd: all authentication tokens updated successfully.
Changing password for user vickey.
passwd: all authentication tokens updated successfully.
Changing password for user kaushal.
passwd: all authentication tokens updated successfully.
Changing password for user manoj.
passwd: all authentication tokens updated successfully.
Changing password for user jai.
passwd: all authentication tokens updated successfully.
[root@Server ~]# _
```

Loop for creating groups

Now create 3 groups named sales market productions using for loop

#for GROUP in sales market productions
> do
>groupadd $GROUP
>done
 Verify by cat and grep commands

```
[root@Server ~]# for GROUP in sales market poductions
> do
> groupadd $GROUP
> done
[root@Server ~]# cat /etc/group |grep sales
sales:x:500:
[root@Server ~]# cat /etc/group |grep market
market:x:501:
[root@Server ~]# cat /etc/group |grep poductions
poductions:x:502:
[root@Server ~]# _
```

For loop for deleting bulk users

Now remove all the user which we created in pervious example.

#for USER in vinita nikkita niddhi sumit shweta vickey kaushal manoj jai
>do
>userdel -r $USER
>done

```
[root@Server ~]# for USER in vinita nikkita niddhi sumit shweta vickey kaushal m
anoj jai
> do
> userdel -r $USER
> done
[root@Server ~]# _
```

For loop for deleting bulk users

Remove groups which we create in pervious example

#for GROUP in sales market productions
> do
>groupdel $GROUP
>done

```
[root@Server ~]# for GROUP in sales market poductions
> do
> groupdel $GROUP
> done
[root@Server ~]# _
```

Changing Owner and Group chown chgrp commands

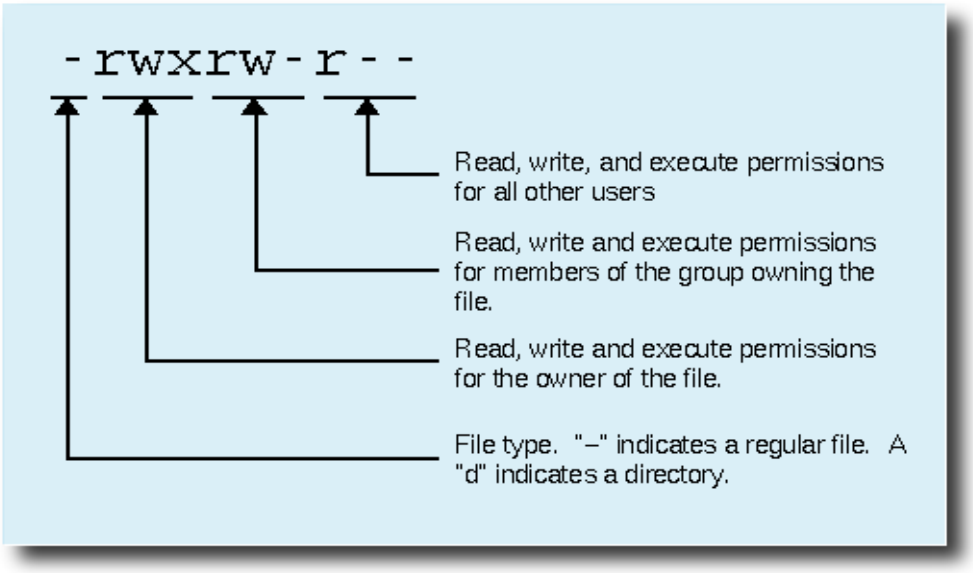
In Red Hat Enterprise Linux, all files have file permissions that determine whether a user is allowed to **read, write, or execute** them. When you issue the command **ls -l,** the first column of information contains these file permissions.Within this first column are places for 10 letters or hyphens.

```
[root@Server ~]# ls -l
total 76
-rw------- 1 root root   801 Dec 13 17:55 anaconda-ks
drwxr-xr-x 2 root root  4096 Dec 14 00:40 backup
-rw-r--r-- 1 root root 40081 Dec 13 17:55 install.log
-rw-r--r-- 1 root root  4950 Dec 13 17:54 install.log
drwxr-xr-x 2 root root  4096 Dec 20 00:22 test
drwxr-xr-x 2 root root  4096 Dec 20 00:50 vinita
[root@Server ~]#
```

The first space is either a hyphen, the letter d, or the letter l.

- A **hyphen** means it is a file.
- If it is the letter **d,** the file is actually a directory.
- If it is the letter **l,** it is a symbolic link to a directory somewhere else on the file system.

The next nine spaces are divided into three sets of three as shown in image.



Files and directories belong to both an owner and a group. A group usually consists of a collection of users, all belonging to the same group. The first set of three is the read, write, and execute permissions for the owner of the file.

A group can also consist of one user, normally the user who creates the file. Each user on the system, including the root user, is assigned his or her own group of which he or she is the only member, ensuring access only by that user. The second set of three is the read, write, and execute permissions for anyone who belongs to the user group for the file.

The last set of permissions is for anyone who has a login to the system.

Ownership

Create a directory /test we will use this for the practical demonstration of permission.

#mkdir /test
#ls –ld /test

```
[root@Server ~]# mkdir /test
[root@Server ~]# ls -ld /test
drwxr-xr-x 2 root root 4096 Dec 20 05:47 /test
[root@Server ~]# _
```

The root user, the system administrator, owns most of the system files that also belong to the root group, of which only the root user is a member. Most administration files, like configuration files in the /etc directory, are owned by the root user and belong to the root group. Only the root user has permission to modify them, whereas normal users can read and, in the case of programs, also execute them.

In this example, the root user owns the fstab file in the /etc directory, which also belongs to the root user group.

-rw-r--r-- 1 root root 621 jan 22 11:03 fstab

Certain directories and files located in the system directories are owned by a service, rather than the root user, because the services need to change those files directly. This is particularly true for services that interact with remote users, such as Internet servers. Most of these files are located in the /var directory. Here you will find files and directories managed by services like the Squid proxy server and the Domain Name Server (named).

In this example, the Squid proxy server directory is owned by the squid user and belongs to the squid group:

drwxr-x--- 2 squid squid 4096 Jan 24 16:29 squid

Changing a File's Owner or Group

Although other users may be able to access a file, only the owner can change its permissions. If you want to give other user to control over one of your file's permissions, you can change the owner of the file from yourself to the other user. The chown command transfers control over a

file to another user. This command takes as its first argument the name of the other user. Following the username, you list the files you are transferring. In our example, we gives control of the /test directory to user a:

 # chown a /test
 # ls -ld /test

```
[root@Server ~]# useradd a
[root@Server ~]# passwd -d a
Removing password for user a.
passwd: Success
[root@Server ~]# chown a /test
[root@Server ~]# ls -ld /test
drwxr-xr-x 2 a root 4096 Dec 20 05:47 /test
[root@Server ~]# _
```

You can also change the group for a file and directories, using the chgrp command. chgrp takes as its first argument the name of the new group for a files or directories.

#chgrp example /test

```
[root@Server ~]# groupadd example
[root@Server ~]# chgrp example /test
[root@Server ~]# ls -ld /test
drwxr-xr-x 2 a example 4096 Dec 20 05:47 /test
[root@Server ~]# _
```

Disk Management simple partition

In this series of article I will demonstrator you necessary disk managements skill for RHCE examinations.

Example Question : -

Add a new logical partition having size 100MB and create the /data directory which will be the mount point for the new partition.

To accomplish this task you must be login form root account. So first login from root and verify your hard disk status with **fdisk –l command** (This command will show that where your hard disk is mounted. You should use the mount point which show in the output of this command. For example if you see **/dev/hda** then you should use **fdisk /dev/hda** in next command. Or if you see **/dev/sdb** then you should use **fdisk /dev/sdb** in next command.

As you can see in image shown below that My hard disk is mounted as **/dev/sda** so I will use ***fdisk /dev/sda***)

```
[root@Sever ~]# fdisk -l

Disk /dev/sda: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          13      104391   83  Linux
/dev/sda2              14        1033     8193150   83  Linux
/dev/sda3            1034        1160     1020127+  82  Linux swap / Solaris
/dev/sda4            1161        1958     6409935    5  Extended
/dev/sda5            1161        1287     1020096   83  Linux
[root@Sever ~]# fdisk /dev/sda

The number of cylinders for this disk is set to 1958.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): _
```

Follow these steps at command prompt in same sequence

n

press enter
+100MB
press enter
w

#partprobe /dev/sda

```
Command (m for help): n
First cylinder (1301-1958, default 1301):
Using default value 1301
Last cylinder or +size or +sizeM or +sizeK (1301-1958, default 1958): +100M

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource
 busy.
The kernel still uses the old table.
The new table will be used at the next reboot.
Syncing disks.
[root@Sever ~]# partprobe /dev/sda
```

fdisk commands is used to create new partitions. partprobe command is used to inform kernel about this change.

Create a /data folder and mount the partition with these commands

```
[root@Sever ~]# mkdir /data
[root@Sever ~]# mount /dev/sda7 /data
[root@Sever ~]# ls -l /data
total 12
drwx------ 2 root root 12288 Oct  2 23:09 lost+found
[root@Sever ~]# vi /etc/fstab _
```

lost+found is a partition specific folder that will appear only in those directory that represent an active partitions.

How to mount simple partition permanently

Create a entry for newly created partition in /etc/fstab so that it can be mount automatically after reboot as shown in image

```
LABEL=/1                /                       ext3    defaults        1 1
LABEL=/boot1            /boot                   ext3    defaults        1 2
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
tmpfs                   /dev/shm                tmpfs   defaults        0 0
LABEL=/home             /home                   ext3    defaults        1 2
proc                    /proc                   proc    defaults        0 0
sysfs                   /sys                    sysfs   defaults        0 0
LABEL=SWAP-sda3         swap                    swap    defaults        0 0
/dev/sda7               /data                   ext3    defaults        0 0
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq_
```

Reboot the system with this command
#reboot -f

If you got no error while system boot then run fdisk -l command to verify that partition has successfully mounted.

```
[root@Sever ~]# fdisk -l

Disk /dev/sda: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          13      104391   83  Linux
/dev/sda2              14        1033     8193150   83  Linux
/dev/sda3            1034        1160     1020127+  82  Linux swap / Solaris
/dev/sda4            1161        1958     6409935    5  Extended
/dev/sda5            1161        1287     1020096   83  Linux
/dev/sda6            1288        1300      104391   83  Linux
/dev/sda7            1301        1313      104391   83  Linux
[root@Sever ~]# _
```

You have effectively accomplished the task given you now its time to remove these partition.

```
[root@Sever ~]# umount /data
[root@Sever ~]# fdisk /dev/sda

The number of cylinders for this disk is set to 1958.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): d
Partition number (1-7): 7

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource
 busy.
The kernel still uses the old table.
The new table will be used at the next reboot.
Syncing disks.
^[Quit
[root@Sever ~]# partprobe /dev/sda_
```

don't forget to remove entry from /etc/fstab files also otherwise you will get error on next boot.

```
[root@Sever ~]# vi /etc/fstab_
```

```
LABEL=/1                /                       ext3    defaults        1 1
LABEL=/boot1            /boot                   ext3    defaults        1 2
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
tmpfs                   /dev/shm                tmpfs   defaults        0 0
LABEL=/home             /home                   ext3    defaults        1 2
proc                    /proc                   proc    defaults        0 0
sysfs                   /sys                    sysfs   defaults        0 0
LABEL=SWAP-sda3         swap                    swap    defaults        0 0
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/etc/fstab" 10L, 610C
```

RHCE Linux Disk Quota Implementation

Disk quotas are commonly used by ISPs, by Web hosting companies, on FTP sites, and on corporate file servers to ensure continued availability of their systems.

Quotas are used to limit a users or a group of users ability to consume disk space. This prevents a small group of users from monopolizing disk capacity and potentially interfering with other users or the entire system.

Without **quotas,** one or more users can upload files on an FTP server to the point of filling a file system. Once the affected partition is full, other users are effectively denied upload access to the disk. This is also a reason to mount different file system directories on different partitions. For example, if you only had partitions for your root (/) directory and swap space, someone uploading to your computer could fill up all of the space in your root directory (/). Without at least a little free space in the root directory (/), your system could become unstable or even crash.

You have two ways to set quotas for users. You can limit users by inodes or by kilobyte-sized disk blocks. Every Linux file requires an inode. Therefore, you can limit users by the number of files or by absolute space. You can set up different quotas for different file systems. For example, you can set different quotas for users on the /home and /tmp directories if they are mounted on their own partitions. Limits on disk blocks restrict the amount of disk space available to a user on your system. Older versions of Red Hat Linux included LinuxConf, which included a graphical tool to configure quotas. Red Hat no longer has a graphical quota configuration tool. Today, you can configure quotas on RHEL only through the command line interface.

Lets look few basic terms used for implementation of disk quota

- **Soft limit**
  This is the maximum amount of space a user can have on that partition. If you have set a grace period, this will act as an alarm. The user will then be notified she is in quota violation. If you have set a grace period, you will also need to set a hard limit. A grace period is the number of days a user is allowed to be above the given quota. After the grace period is over, the user must get under the soft limit to continue. By default grace period have seven days limits.
- **Hard limit**
  Hard limits are necessary only when you are using grace periods. If grace periods are enabled, this will be the absolute limit a user can use. Any attempt to consume resources beyond this limit will be denied. If you are not using grace periods, the soft limit is the maximum amount of available space for each user.
- **GracePeriods**
  Linux has provided the default of seven days for both inodes and block usage. That is, a user may exceed the soft limit on either resource for up to seven days. After that, further requests by that user to use files will be denied.

Reference Table	
Soft Limit	Disk space a user can use
Hard limit	Absolute limit a user can use
Grace Periods	Time duration till user can use hard limit space
1 inode	1 KB
dd	used to create a blank file of specific size
required RPM	quota-3.13-1.2.3.2.el5
/etc/fstab options	usrquota, grpquota
Quota files	aquota.user, aquota.group
Necessary command	mount, quotaon, quotacheck, edquota, quotaoff

Quota Tools

Quota checks can be implemented on the file system of a hard disk partition mounted on your system. The quotas are enabled using the **quotacheck** and **quotaon** programs. They are executed in the ***/etc/rc.d/rc.sysinit*** script, which is run whenever you start up your system. Each partition needs to be mounted with the quota options, usrquota or grpquota. usrquota enables quota controls for users, and grpquota works for groups.

You also need to create *quota.user* and *quota.group* files for each partition for which you enable quotas. These are the quota databases that hold the quota information for each user and group. You can create these files by running the quotacheck command with the **-a** option or the device name of the file system where you want to enable quotas.

edquota

You can set disk quotas using the **edquota** command. With it, you can access the quota record for a particular user and group, which is maintained in the disk quota database. You can also set default quotas that will be applied to any user or group on the file system for which quotas have not been set. edquota will open the record in your default editor, and you can use your editor to make any changes. To open the record for a particular user, use the **-u** option and the username as an argument for edquota

quotacheck, quotaon, and quotaoff

The quota records are maintained in the quota database for that partition. Each partition that has quotas enabled has its own quota database. You can check the validity of your quota database with the quotacheck command. You can turn quotas on and off using the quotaon and quotaoff commands. When you start up your system, quotacheck is run to check the quota databases, and then quotaon is run to turn on quotas.

repquota

As the system administrator, you can use the repquota command to generate a summary of disk usage for a specified file system, checking to see what users are approaching or exceeding

quota limits. repquota takes as its argument the file system to check; the -a option checks all file systems.

Disk Quota Implementation RHCE Linux

In our pervious article we learn about basic necessary concept used for disk quota implementation. Now we will first look at the outline of question that may be asked in RHCE exam.

Example 1
Quota is implemented on /home but not working properly. Find out the Problem and implement the quota to vinita to have a soft limit 50 inodes (files) and hard limit of 100 inodes (files).

Example 2
vinita user tried by:
dd if=/dev/zero of=/home/vinita/test bs=1024 count=50
files created successfully. Again vinita tried to create file having 100k using following command:
dd if=/dev/zero of=/home/vinita/test1 bs=1024 count=100
But she is unable to create the file. Make the user can create the file less then 100K.

Solutions

Example 2 is extremely complicated question from Redhat. Actually question is giving scenario to you to implement quota to vinita user. You should apply the quota to vinita user on /home that vinita user shouldn't occupied space more than 100K.

To solve disk quota question follow these guides

Check the quota RPM (installed by default)

```
[root@Sever ~]# rpm -qa quota
quota-3.13-1.2.3.2.el5
[root@Sever ~]# _
```

Now open /etc/fstab file to enable quota entry

```
[root@Sever ~]# vi /etc/fstab_
```

You can tell Linux to start tracking user quotas by adding the keyword **usrquota** under the options column. Similarly, you can tell Linux to start tracking group quotas with the **grpquota** option.

I add both user and group quotas to the /home directory filesystem

```
LABEL=/1                /                       ext3    defaults        1 1
LABEL=/boot1            /boot                   ext3    defaults        1 2
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
tmpfs                   /dev/shm                tmpfs   defaults        0 0
LABEL=/home             /home                   ext3    defaults,usrquota,grpquo
ta        1 2
proc                    /proc                   proc    defaults        0 0
sysfs                   /sys                    sysfs   defaults        0 0
LABEL=SWAP-sda3         swap                    swap    defaults        0 0
~
~
~
~
~
:wq_
```

Either Reboot the System or remount the partition and verify it

```
[root@Sever ~]# mount -o remount,usrquota,grpquota,rw /home
[root@Sever ~]# mount |grep /home
/dev/sda5 on /home type ext3 (rw,usrquota,grpquota,usrquota,grpquota)
[root@Sever ~]# _
```

The next step is to create quota files. For user and group quotas, you will need the **aquota.user** and **aquota.group** files in the selected filesystem before you can activate actual quotas.
You can do it either manually or the appropriate **quotacheck** command creates them automatically.
For the /home directory described earlier, you would use the following commands:

mount -o remount /home
quotacheck -cugm /home

The options for quotacheck are

- -c Performs a new scan.
- -v Performs a verbose scan.
- -u Scans for user quotas.
- -g Scans for group quotas.
- -m Remounts the scanned filesystem.

This will check the current quota information for all users, groups, and partitions. It stores this information in the appropriate quota partitions. Once the command is run, you should be able to find the aquota.user and aquota.group files in the configured directory.

or you can create these files manually
#touch /home/aquota.group
#touch /home/aquota.user

```
[root@Sever ~]# mount -o remount,usrquota,grpquota,rw /home
[root@Sever ~]# mount |grep /home
/dev/sda5 on /home type ext3 (rw,usrquota,grpquota,usrquota,grpquota)
[root@Sever ~]# quotacheck -auvg
quotacheck: Scanning /dev/sda5 [/home] quotacheck: Cannot stat old user quota fi
le: No such file or directory
quotacheck: Cannot stat old group quota file: No such file or directory
quotacheck: Cannot stat old user quota file: No such file or directory
quotacheck: Cannot stat old group quota file: No such file or directory
done
quotacheck: Checked 3 directories and 2 files
quotacheck: Old file not found.
quotacheck: Old file not found.
[root@Sever ~]# quotacheck -auvg
quotacheck: Scanning /dev/sda5 [/home] done
quotacheck: Checked 3 directories and 4 files
[root@Sever ~]# ls /home
aquota.group  aquota.user  lost+found
[root@Sever ~]# touch /home/aquota.group
[root@Sever ~]# touch /home/aquota.user
[root@Sever ~]# _
```

On this quota by **quotaon** command and create an example user named vinita. use **edquota** command to set quota on vinita user.

```
[root@Sever ~]# quotaon -avug
/dev/sda5 [/home]: group quotas turned on
/dev/sda5 [/home]: user quotas turned on
[root@Sever ~]# useradd vinita
[root@Sever ~]# passwd vinita
Changing password for user vinita.
New UNIX password:
BAD PASSWORD: it is WAY too short
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[root@Sever ~]# edquota vinita_
```

edquota command will edits the **aquota.user** or **aquota.group** file with the vi editor. In our example, we have a user named vinita, and we want to restrict how much disk space she is allowed to use.

Set softlink to 50 and 100 to hard link

```
Disk quotas for user vinita (uid 500):
  Filesystem                   blocks       soft       hard     inodes     soft
    hard
  /dev/sda5                        16          50        100          4        0
       0
~
~
~
~
~
~
:wq_
```

You can also set grace period for user (set only if you are asked by examiner)

```
[root@Sever ~]# edquota -T vinita_
```

We will change default 7 days grace period to 10 days.

```
Times to enforce softlimit for user vinita (uid 500):
Time units may be: days, hours, minutes, or seconds
  Filesystem                         block grace                 inode grace
  /dev/sda5                                10days                    unset
~
~
~
~
:wq_
```

Remount the partition and verify it

```
[root@Sever ~]# mount -o remount,usrquota,grpquota,rw /home
[root@Sever ~]# mount |grep /home
/dev/sda5 on /home type ext3 (rw,usrquota,grpquota,usrquota,grpquota)
[root@Sever ~]# _
```

Verify that you have successfully implemented disk quota

```
[root@Sever ~]# repquota -a
*** Report for user quotas on device /dev/sda5
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      --   17676       0       0              4     0     0
vinita    --      16      50     100              4     0     0


[root@Sever ~]# _
```

Now login from Vinita

```
Red Hat Enterprise Linux Server release 5 (Tikanga)
Kernel 2.6.18-8.el5 on an i686

Sever login: vinita
Password:
[vinita@Sever ~]$ _
```

When testing quotas, there is one useful technique that creates a blank file of the desired size. For example, if you want to create a 100MB file named test in the local directory, run this command
dd if=/dev/zero of=test bs=1k count=100000 .

```
[vinita@Sever ~]$ dd if=/dev/zero of=/home/vinita/test bs=1024 count=50
sda5: warning, user block quota exceeded.
50+0 records in
50+0 records out
51200 bytes (51 kB) copied, 0.000829547 seconds, 61.7 MB/s
[vinita@Sever ~]$ dd if=/dev/zero of=/home/vinita/test1 bs=1024 count=500
sda5: write failed, user block limit reached.
dd: writing `/home/vinita/test1': Disk quota exceeded
29+0 records in
28+0 records out
28672 bytes (29 kB) copied, 0.00138989 seconds, 20.6 MB/s
[vinita@Sever ~]$ dd if=/dev/zero of=/home/vinita/test2 bs=1024 count=5000
dd: writing `/home/vinita/test2': Disk quota exceeded
1+0 records in
0+0 records out
0 bytes (0 B) copied, 0.000932219 seconds, 0.0 kB/s
[vinita@Sever ~]$ ls
test  test1  test2
[vinita@Sever ~]$ du -h
100K    .
[vinita@Sever ~]$ du -h tes*
56K     test
28K     test1
0       test2
[vinita@Sever ~]$ _
```

When user vinita run dd command first time to create a blank file more then 50 Mb she got an warning and the file was created. As she is allowed to exceed her soft limit for 10 days.

Second time she tried to create a file of 500 Mb. As you can see in image she was able only to create a file of 20 Mb. As she can not exceed her hard limit that is set to 100 inode.

In third time she is denied to use any more space as she have already crossed her hard limit.

You can verify the space of created file by du command with -h options.

After successfully completing your practical remove quota entry /etc/fstab

```
LABEL=/1                /                       ext3    defaults        1 1
LABEL=/boot1            /boot                   ext3    defaults        1 2
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
tmpfs                   /dev/shm                tmpfs   defaults        0 0
LABEL=/home             /home                   ext3    defaults        1 2
proc                    /proc                   proc    defaults        0 0
sysfs                   /sys                    sysfs   defaults        0 0
LABEL=SWAP-sda3         swap                    swap    defaults        0 0
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/etc/fstab" 10L, 610C
```

Now turn off quota by quotaoff command and remount home partition for further practical

```
[root@Sever ~]# quotaoff -auvg
/dev/sda5 [/home]: group quotas turned off
/dev/sda5 [/home]: user quotas turned off
[root@Sever ~]# mount -o remount,rw /home
[root@Sever ~]# _
```

.

RHCE Linux lvm partitions

Form the beginning of RHCE exam RedHat always includes a question about LVM partitions. So you must be able to create the LVM partition and mount them properly in /etc/fstab

Explanations of basic definitions

The LVM system organizes hard disks into Logical Volume (LV) groups. Essentially, physical hard disk partitions (or possibly RAID arrays) are set up in a bunch of equal sized chunks known as Physical Extents (PE). As there are several other concepts associated with the LVM system, here we will discuss only some basic definitions those require in rhce:

- **Physical Volume (PV)** is the standard partition that you add to the LVM mix. Normally, a physical volume is a standard primary or logical partition. It can also be a RAID array.
- **Physical Extent (PE)** is a chunk of disk space. Every PV is divided into a number of equal sized PEs. Every PE in a LV group is the same size. Different LV groups can have different sized PEs.
- **Logical Extent (LE)** is also a chunk of disk space. Every LE is mapped to a specific PE.
- **Logical Volume (LV)** is composed of a group of LEs. You can mount a filesystem such as /home and /var on an LV.
- **Volume Group (VG)** is composed of a group of LVs. It is the organizational group for LVM.

Create lvm partition and resize them

Run fdisk /dev/sda to invoke fdisk. Make sure your hard disk status via fdisk -l command before it. If you see /dev/hda in the output of fdisk -l command run fdisk /dev/hda instead of fdisk /dev/sda

```
[root@Sever ~]# fdisk -l

Disk /dev/sda: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          13      104391   83  Linux
/dev/sda2              14        1033     8193150   83  Linux
/dev/sda3            1034        1160     1020127+  82  Linux swap / Solaris
/dev/sda4            1161        1958     6409935    5  Extended
/dev/sda5            1161        1287     1020096   83  Linux
[root@Sever ~]# fdisk /dev/sda

The number of cylinders for this disk is set to 1958.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): _
```

```
Command (m for help): n
First cylinder (1301-1958, default 1301):
Using default value 1301
Last cylinder or +size or +sizeM or +sizeK (1301-1958, default 1958): +100M

Command (m for help): n
First cylinder (1314-1958, default 1314):
Using default value 1314
Last cylinder or +size or +sizeM or +sizeK (1314-1958, default 1958): +100M

Command (m for help): n
First cylinder (1327-1958, default 1327):
Using default value 1327
Last cylinder or +size or +sizeM or +sizeK (1327-1958, default 1958): +100M

Command (m for help): _
```

after creating partition define their file type and save via w command. lvm partitions are denoted as 8e. run these command exactly (caution:- change only the partition you create)

```
Command (m for help): t
Partition number (1-9): 7
Hex code (type L to list codes): 8e

Command (m for help): t
Partition number (1-9): 8
Hex code (type L to list codes): 8e

Command (m for help): t
Partition number (1-9): 9
Hex code (type L to list codes): 8e

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource
 busy.
The kernel still uses the old table.
The new table will be used at the next reboot.
Syncing disks.
^[Quit
[root@Sever ~]# _
```

Now tell kernel about this change run partprobe command

```
[root@Sever ~]# partprobe /dev/sda
[root@Sever ~]# fdisk -l

Disk /dev/sda: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          13      104391   83  Linux
/dev/sda2              14        1033     8193150   83  Linux
/dev/sda3            1034        1160     1020127+  82  Linux swap / Solaris
/dev/sda4            1161        1958     6409935    5  Extended
/dev/sda5            1161        1287     1020096   83  Linux
/dev/sda6            1288        1300      104391   83  Linux
/dev/sda7            1301        1313      104391   8e  Linux LVM
/dev/sda8            1314        1326      104391   8e  Linux LVM
/dev/sda9            1327        1339      104391   8e  Linux LVM
[root@Sever ~]# _
```

Create physical volume from newly created partition and then we will create a volume group to use these physical volumes.

```
[root@Sever ~]# pvcreate /dev/sda{7,8,9}
  Physical volume "/dev/sda7" successfully created
  Physical volume "/dev/sda8" successfully created
  Physical volume "/dev/sda9" successfully created
[root@Sever ~]# vgcreate vg00 /dev/sda7 /dev/sda8 /dev/sda9
  Volume group "vg00" successfully created
[root@Sever ~]# _
```

Create 2 lvm partition from this volume group

```
[root@Sever ~]# lvcreate -L 100M -n lv00 /dev/vg00
  Logical volume "lv00" created
[root@Sever ~]# lvcreate -L 100M -n lv01 /dev/vg00
  Logical volume "lv01" created
[root@Sever ~]# _
```

For format you can use either mke2fs with -j switch or just single command mkfs.ext3
I used both command for illustration

```
[root@Sever ~]# mke2fs -j /dev/vg00/lv01
mke2fs 1.39 (29-May-2006)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
25688 inodes, 102400 blocks
5120 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=67371008
13 block groups
8192 blocks per group, 8192 fragments per group
1976 inodes per group
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345, 73729

Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 37 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
[root@Sever ~]# _
```

```
[root@Sever ~]# mkfs.ext3 /dev/vg00/lv00
mke2fs 1.39 (29-May-2006)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
25688 inodes, 102400 blocks
5120 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=67371008
13 block groups
8192 blocks per group, 8192 fragments per group
1976 inodes per group
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345, 73729

Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 32 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
[root@Sever ~]# _
```

now we will define mount point to use this partitions

```
[root@Sever ~]# mkdir -p /data/lv00
[root@Sever ~]# mkdir /data/lv01
[root@Sever ~]# mount /dev/vg00/lv00 /data/lv00
[root@Sever ~]# mount /dev/vg00/lv01 /data/lv01
[root@Sever ~]# ls -l /data/lv00
total 12
drwx------ 2 root root 12288 Oct  2 22:34 lost+found
[root@Sever ~]# ls -l /data/lv01
total 12
drwx------ 2 root root 12288 Oct  2 22:35 lost+found
[root@Sever ~]# _
```

We can use these partitions until system reboot if you are asked to mount these partition permanently use these steps.

Run vi /etc/fstab

```
[root@Sever ~]# vi /etc/fstab_
```

(fstab :- file contain information about linux partition)
(vi :- editing command if have problem using vi follow this link)

Make entry of newly created lvm partition

/dev/vg00/lv00	/data/lv00	ext3	defaults 0 0
/dev/vg00/lv01	/data/lv01	ext3	defaults 0 0

in the end of files as shown in figure and save the file.

```
LABEL=/1                /                       ext3    defaults        1 1
LABEL=/boot1            /boot                   ext3    defaults        1 2
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
tmpfs                   /dev/shm                tmpfs   defaults        0 0
LABEL=/home             /home                   ext3    defaults        1 2
proc                    /proc                   proc    defaults        0 0
sysfs                   /sys                    sysfs   defaults        0 0
LABEL=SWAP-sda3         swap                    swap    defaults        0 0
/dev/vg00/lv00          /data/lv00              ext3    defaults        0 0
/dev/vg00/lv01          /data/lv01              ext3    defaults        0 0
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq_
```

In this tutorial we learn how to create lvm partition and mount them permanently

Lvm Advance Management Extend /Reduce/ Add/ Remove.

In our pervious article we learnt how to create lvm partition and mount them permanently in **/etc/fstab** . If you missed it we suggest reviewing
RHCE Linux lvm partitions
before going with advance lvm management. In this tutorial guide we will learn about advance lvm management. In our pervious article we have created two lvm partitions. Now we will manage them. First we will extend the size of lv00 our first lvm partition.

Check all these partition are functioning properly

```
[root@Sever ~]# pvdisplay |more_
```

```
  --- Physical volume ---
  PV Name               /dev/sda7
  VG Name               vg00
  PV Size               101.94 MB / not usable 1.94 MB
  Allocatable           yes
  PE Size (KByte)       4096
  Total PE              25
  Free PE               25
  Allocated PE          0
  PV UUID               5wq7qg-4409-zMDA-YGSi-WSKy-2T18-DF3UWP

  --- Physical volume ---
  PV Name               /dev/sda8
  VG Name               vg00
  PV Size               101.94 MB / not usable 1.94 MB
  Allocatable           yes
  PE Size (KByte)       4096
  Total PE              25
  Free PE               25
  Allocated PE          0
  PV UUID               1CZONv-1mfT-aOZ1-t2YG-G1oC-tyAS-NzQEXg

  --- Physical volume ---
  PV Name               /dev/sda9
--More--
```

```
[root@Sever ~]# vgdisplay
  --- Volume group ---
  VG Name               vg00
  System ID
  Format                lvm2
  Metadata Areas        3
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                3
  Act PV                3
  VG Size               300.00 MB
  PE Size               4.00 MB
  Total PE              75
  Alloc PE / Size       0 / 0
  Free  PE / Size       75 / 300.00 MB
  VG UUID               i13DGA-pNiH-4Sch-qybt-Devf-pXKS-FMyQd2

[root@Sever ~]# _
```

Check the current size of **lv00**

```
[root@Sever ~]# lvdisplay /dev/vg00/lv00
  --- Logical volume ---
  LV Name                /dev/vg00/lv00
  VG Name                vg00
  LV UUID                cA4gCg-t7dn-58uR-y0K2-6Lcx-B2El-LO7wW0
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                100.00 MB
  Current LE             25
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           253:0
```

As you can see the current size of **lv00** is 100MB. To extend it with 50MB space run these commands

```
[root@Sever ~]# lvextend -L +50M /dev/vg00/lv00
  Rounding up size to full physical extent 52.00 MB
  Extending logical volume lv00 to 152.00 MB
  Logical volume lv00 successfully resized
[root@Sever ~]# lvdisplay /data/vg00/lv00
  "/data/vg00/lv00": Invalid path for Logical Volume
[root@Sever ~]# lvdisplay /dev/vg00/lv00
  --- Logical volume ---
  LV Name                /dev/vg00/lv00
  VG Name                vg00
  LV UUID                cA4gCg-t7dn-58uR-y0K2-6Lcx-B2El-LO7wW0
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                152.00 MB
  Current LE             38
  Segments               2
  Allocation             inherit
  Read ahead sectors     0
  Block device           253:0

[root@Sever ~]# _
```

Now we will reduce the size of 20MB from lvm partition.
If you have free space in lvm partition no data lose will happen. Lose in data will happen only

when partition is full.

```
[root@Sever ~]# lvreduce -L 20M /dev/vg00/lv00
  WARNING: Reducing active and open logical volume to 20.00 MB
  THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce lv00? [y/n]: y
  Reducing logical volume lv00 to 20.00 MB
  Logical volume lv00 successfully resized
[root@Sever ~]# lvdisplay /dev/vg00/lv00
  --- Logical volume ---
  LV Name                /dev/vg00/lv00
  VG Name                vg00
  LV UUID                cA4gCg-t7dn-58uR-y0K2-6Lcx-B2El-LO7wW0
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                20.00 MB
  Current LE             5
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           253:0

[root@Sever ~]# _
```

at this moment you should be able to **extend** and **reduce** the size of lvm partition now we will remove these partitions

```
[root@Sever ~]# umount /data/lv00
[root@Sever ~]# umount /data/lv01
[root@Sever ~]# lvremove /dev/vg00/lv00
Do you really want to remove active logical volume "lv00"? [y/n]: y
  Logical volume "lv00" successfully removed
[root@Sever ~]# lvremove /dev/vg00/lv01
Do you really want to remove active logical volume "lv01"? [y/n]: y
  Logical volume "lv01" successfully removed
[root@Sever ~]# vgremove /dev/vg00
  Volume group "vg00" successfully removed
[root@Sever ~]# pvremove /dev/sda{7,8,9}
  Labels on physical volume "/dev/sda7" successfully wiped
  Labels on physical volume "/dev/sda8" successfully wiped
  Labels on physical volume "/dev/sda9" successfully wiped
[root@Sever ~]# _
```

don't forget to remove these partitions form **/etc/fstab** also

```
LABEL=/1                /                       ext3    defaults        1 1
LABEL=/boot1            /boot                   ext3    defaults        1 2
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
tmpfs                   /dev/shm                tmpfs   defaults        0 0
LABEL=/home             /home                   ext3    defaults        1 2
proc                    /proc                   proc    defaults        0 0
sysfs                   /sys                    sysfs   defaults        0 0
LABEL=SWAP-sda3         swap                    swap    defaults        0 0
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/etc/fstab" 10L, 610C
```

now use **fdisk** command to delete these partition form hard disk

#fdisk /dev/sda

```
Command (m for help): d
Partition number (1-9): 9

Command (m for help): d
Partition number (1-8): 8

Command (m for help): d
Partition number (1-7): 7

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource
 busy.
The kernel still uses the old table.
The new table will be used at the next reboot.
Syncing disks.
[root@Sever ~]# partprobe /dev/sda
[root@Sever ~]# _
```

run **partprobe** command to tell kernel about change and verify it

```
[root@Sever ~]# fdisk -l

Disk /dev/sda: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          13      104391   83  Linux
/dev/sda2              14        1033     8193150   83  Linux
/dev/sda3            1034        1160     1020127+  82  Linux swap / Solaris
/dev/sda4            1161        1958     6409935    5  Extended
/dev/sda5            1161        1287     1020096   83  Linux
[root@Sever ~]# _
```

Do these practical of lvm as much as you can as RHCE is performance based exam and none of us will want to retake it. We have summarized all important commands in a single table cram it.

Command	Description
partprobe	To notify kernel about the change in device
pvcreate	To create physical volume
pvdisplay	To display the size and other information about physical volume
pvremove	To remove the physical volume
pvresize	To modify the size of physical volume
pvscan	To scan the physical volume
vgextend	to extend the size of volume group
vgreduce	to reduce the size of volume group
vgscan	to scan the volume group for change
vgdisplay	to display the status and information about volume group
lvdisplay	will display the status of lvm
lvreduce	to reduce the size of lvm
lvextend	to extend the size of lvm
lvcreate	to create the lvm

lvremove	to remove the lvm partition	
lvscan	to scan lvm for change	
fdisk	-l	To show the current partition status of hard disk
fdisk	/dev/sda	To invoke fdisk utility on /dev/sda
	n	to create new partition
	t	to define file types
	d	to delete partitions
	w	to save change

Raid partition step by step example and implementations

A ***Redundant Array of Independent Disks (RAID)*** is a series of disks that can save your data even if a terrible failure occurs on one of the disks. While some versions of RAID make complete copies of your data, others use the so-called parity bit to allow your computer to rebuild the data on lost disks

RAID allows an administrator to form an array of several hard drives into one logical drive recognized as one drive by the operating system. It also spreads the data stored over the array of drives to decrease disk access time and accomplish data redundancy. The data redundancy can be used to recover data should one of the hard drives in the array crash.

RAID level 0, or striping,

Means that data is written across all hard drives in the array to accomplish the fast disk performance. No redundancy is used, so the size of the logical RAID drive is equal to the size of all the hard drives in the array. Because there is no redundancy, recovering data from a hard drive crash is not possible through RAID.

RAID level 1, or mirroring,

Means that all data is written to each disk in the array, accomplishing redundancy. The data is “mirrored” on a second drive. This allows for easy recovery should a disk fail. However, it does mean that, for example, if there are two disks in the array, the size for the logical disk is size of the smaller of the two disks because data must be mirrored to the second disk.

RAID level 5

Combines striping and parity. Data is written across all disks as in RAID 0, but parity data is also written to one of the disks. Should a hard drive failure occur, this parity data can be used to recover the data from the failed drive, including while the data is being accessed and the drive is still missing from the array.

RAID level 6

Data is written across all disks as in RAID 5, but two sets of parity data is calculated. Performance is slightly worse than RAID 5 because the extra parity data must be calculated and written to disk. RAID 5 allows for recovery using the parity data if only one drive in the array fails. Because of the dual parity, RAID 6 allows for recovery from the failure of up to two drives in the array.

In real life we never create raid on same hard disk. But its exam and examiner is not going to provide you three spare hard disk so you should be able to create three raid partition on same physical hard disk.

To create raid partition we will use fdisk utility.

Execute fdisk command with -l switch it will show your hard disks mount point

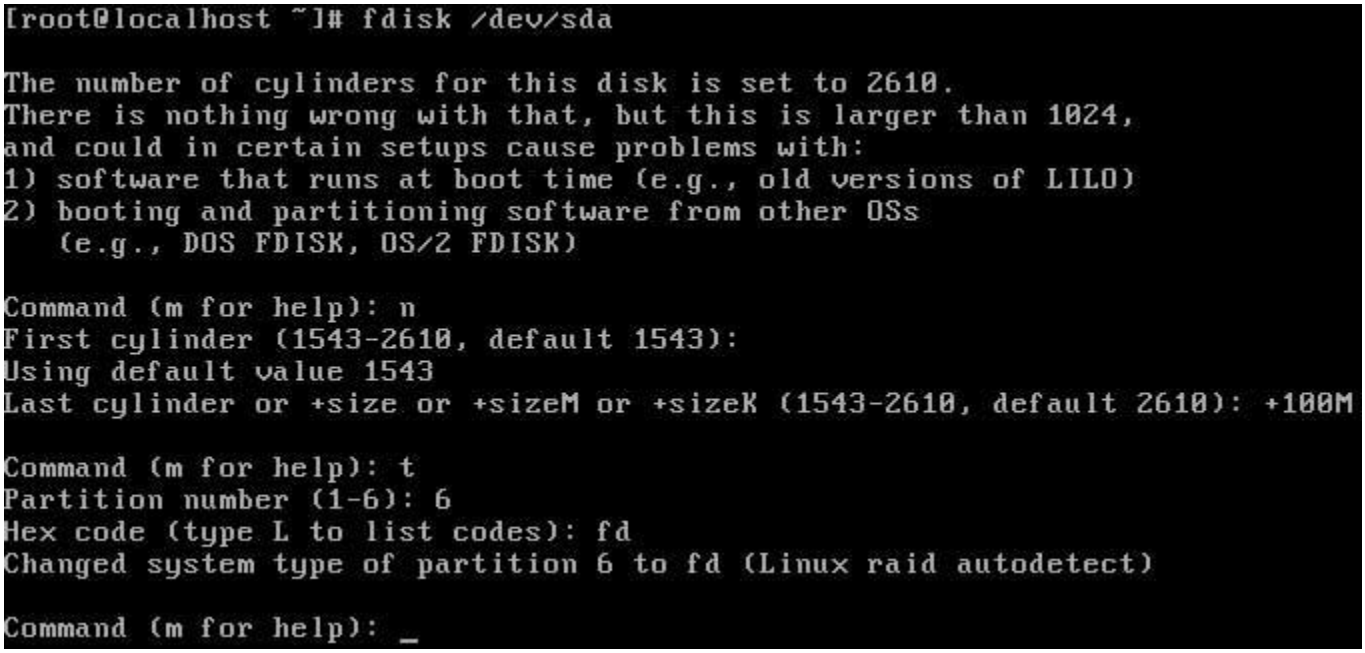
#fdisk -l

now use fdisk commands with proper hard disk options. I am using /dev/sda as you can see in image my hard disk is mounted on /dev/sda. you should the proper hard disk whatever you receive in the output of this commands for example it could be hdd sdb

#fdisk /dev/sda

now create a new partition and assign file type to raid

Command (m for help)n
First cylinder (1543-2610, deaults 1543): press enter
Using deafults value 1543
Last cylender .....): +100M



Now create two more partition repeating the same process don't forget to save with w commands.

```
Command (m for help): n
First cylinder (1556-2610, default 1556):
Using default value 1556
Last cylinder or +size or +sizeM or +sizeK (1556-2610, default 2610): +100M

Command (m for help): t
Partition number (1-7): 7
Hex code (type L to list codes): fd
Changed system type of partition 7 to fd (Linux raid autodetect)

Command (m for help): n
First cylinder (1569-2610, default 1569):
Using default value 1569
Last cylinder or +size or +sizeM or +sizeK (1569-2610, default 2610): +100M

Command (m for help): t
Partition number (1-8): 8
Hex code (type L to list codes): fd
Changed system type of partition 8 to fd (Linux raid autodetect)

Command (m for help): w_
```

Inform kernel about this change by partprobe commands and verify with fdisk -l commands

```
[root@localhost ~]# partprobe /dev/sda
[root@localhost ~]# _
[root@localhost ~]# fdisk -l

Disk /dev/sda: 21.4 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          13      104391   83  Linux
/dev/sda2              14        1288    10241437+  83  Linux
/dev/sda3            1289        1415     1020127+  82  Linux swap
/dev/sda4            1416        2610     9598837+   5  Extended
/dev/sda5            1416        1542     1020096   83  Linux
/dev/sda6            1543        1555      104391   fd  Linux raid
/dev/sda7            1556        1568      104391   fd  Linux raid
/dev/sda8            1569        1581      104391   fd  Linux raid
```

Okey now you have 3 raid partitions and Linux will treat these partition same as three physical hard disks. You can create raid device with these partitions

Create raid 5 device with these partitions

```
[root@localhost ~]# mdadm --create /dev/md0 --level=5 --raid-disk=3 /dev/sda6 /d
ev/sda7 /dev/sda8_
```

Now format this newly created md0 raid device

```
[root@localhost ~]# mke2fs -j /dev/md0
mke2fs 1.39 (29-May-2006)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
52208 inodes, 208640 blocks
10432 blocks (5.00%) reserved for the
First data block=1
Maximum filesystem blocks=67371008
26 block groups
8192 blocks per group, 8192 fragments
2008 inodes per group
```

To use this device you must have specify mount point. we are going to use this raid device on /data directory. So create a /data directory and mount md0 on it

```
[root@Server ~]# mkdir /data
[root@Server ~]# mount /dev/md0  /data
```

This way will mount temporary. To permanently mount make its entry in /etc/fstab file

```
[root@Sever ~]# vi /etc/fstab_
```

```
LABEL=/home             /home                   ext3    defaults        1 2
LABEL=/boot             /boot                   ext3    defaults        1 2
tmpfs                   /dev/shm                tmpfs   defaults        0 0
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
sysfs                   /sys                    sysfs   defaults        0 0
proc                    /proc                   proc    defaults        0 0
LABEL=SWAP-sda3         swap                    swap    defaults        0 0
/dev/md0                /data                   ext3    defaults        1 2
```

Raid partition step by step example and implementations

In our last article we created a raid device md0 from three hard disk partition. In this article we show you that how can you manage a raid device. This guide is the second part of our tutorial on raid devices if you have missed first part then read its first.

Creating a raid devices

First verify all raid partition which we created in our pervious article are working properly and in active state.

#mdadm --detail /dev/md0

```
Number   Major   Minor   RaidDevice State
   0       8       6        0      active sync   /dev/sda6
   1       8       7        1      active sync   /dev/sda7
   2       8       8        2      active sync   /dev/sda8
```

Alternately you can also use cat /proc/mdstat command to see run time status of raid device

```
[root@localhost ~]# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4]
md0 : active raid5 sda7[1] sda8[2] sda6[0]
      208640 blocks level 5, 64k chunk, algorithm 2 [3/3] [UUU]

unused devices: <none>
[root@localhost ~]# _
```

raid devices are mainly used for data backup. Data will be safe even if any hard disk get fail from raid devices. To test it first copy some data in **/data** directory which is the mount point of **md0** raid devices.

#cp * /data
#ls /data

Now assume that one hard disk form raid device gets fail. This can be done by mdadm command with --fail options. Verify data it should be safe

```
[root@localhost ~]# cp * /data
[root@localhost ~]# ls /data
anaconda-ks.cfg  install.log  install.log.syslog  lost+found
[root@localhost ~]# mdadm /dev/md0 --fail /dev/sda7
raid5: Disk failure on sda7, disabling device. Operation continuing
mdadm: set /dev/sda7 faulty in /dev/md0
RAID5 conf printout:
 --- rd:3 wd:2 fd:1
 disk 0, o:1, dev:sda6
 disk 1, o:0, dev:sda7
 disk 2, o:1, dev:sda8
[root@localhost ~]# RAID5 conf printout:
 --- rd:3 wd:2 fd:1
 disk 0, o:1, dev:sda6
 disk 2, o:1, dev:sda8

[root@localhost ~]# ls /data
anaconda-ks.cfg  install.log  install.log.syslog  lost+found
[root@localhost ~]# _
```

you can verify fail device status via mdadm commands with -- detail switch also

```
[root@localhost ~]# mdadm --detail /dev/md0_
 Number   Major   Minor   RaidDevice State
    0       8        6        0      active sync   /dev/sda6
    1       0        0        1      removed
    2       8        8        2      active sync   /dev/sda8

    3       8        7        -      faulty spare   /dev/sda7
```

run time process can be checked via cat /proc/mdstat commands

```
[root@localhost ~]# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4]
md0 : active raid5 sda8[2] sda6[0]
      208640 blocks level 5, 64k chunk, algorithm 2 [3/2]

unused devices: <none>
[root@localhost ~]# _
```

At this point you should have understand what exactly raid device do?. raid device keep your data safe even if hard disk got crashed. its give you enough time to replace faulty hard disk without losing data.

Now remove faulty devices.[Note:- you must have to set fail status before removing it form raid devices]

```
[root@localhost ~]# mdadm /dev/md0 --remove /dev/sda7
mdadm: hot removed /dev/sda7
[root@localhost ~]# _
```

now replace faulty hard disk with good one. To add new disk in raid devices use --add options with mdadm commands

```
[root@localhost ~]# mdadm /dev/md0 --add /dev/sda7
md0: WARNING: sda7 appears to be on the same physical disk as sda8.
     protection against single-disk failure might be compromised.
RAID5 conf printout:
 --- rd:3 wd:2 fd:1
 disk 0, o:1, dev:sda6
 disk 1, o:1, dev:sda7
 disk 2, o:1, dev:sda8
mdadm: re-added /dev/sda7
[root@localhost ~]# RAID5 conf printout:
 --- rd:3 wd:3 fd:0
 disk 0, o:1, dev:sda6
 disk 1, o:1, dev:sda7
 disk 2, o:1, dev:sda8

[root@localhost ~]# _
```

Congratulations you have successful created raid device and changed faulty hard disk. We suggest you to clean these partition before doing further particular.

How to remove raid devices

First un-mount the raid partitions from /data directory

```
[root@localhost ~]# umount /data
[root@localhost ~]# vi /etc/fstab_
```

Now stop /md0 devices and remove it form mdadm commands

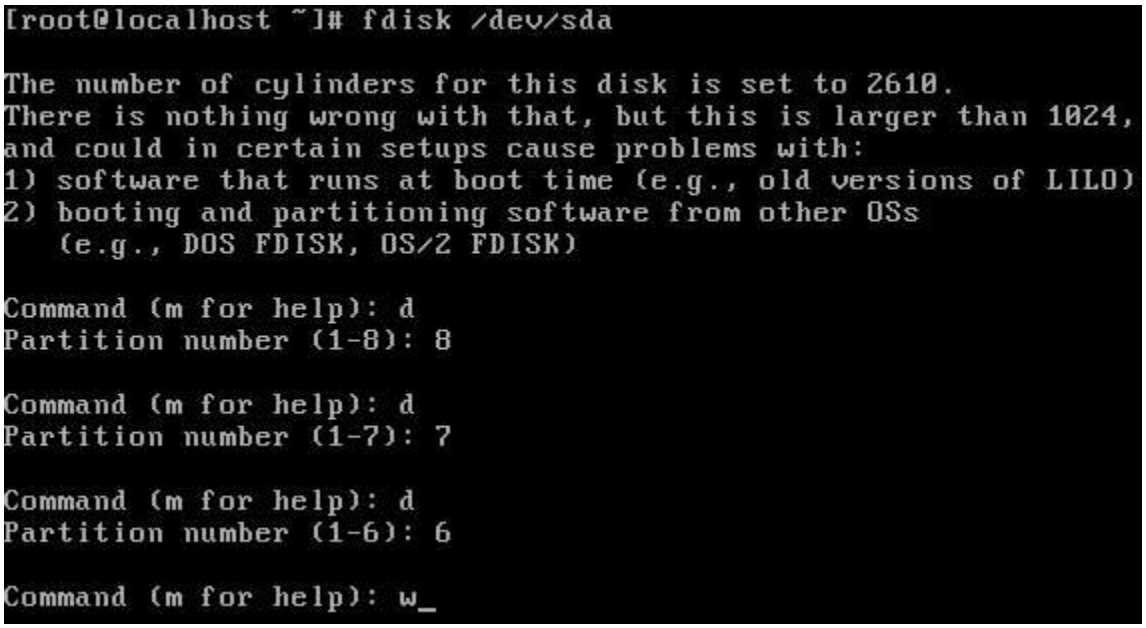
```
[root@localhost ~]# mdadm --stop /dev/md0
mdadm: stopped /dev/md0
[root@localhost ~]# mdadm --remove /dev/md0
[root@localhost ~]# _
```

Now remove entry form /etc/fstab

```
[root@Sever ~]# vi /etc/fstab_
```

```
LABEL=/1                /                       ext3    defaults        1 1
LABEL=/boot1            /boot                   ext3    defaults        1 2
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
tmpfs                   /dev/shm                tmpfs   defaults        0 0
LABEL=/home             /home                   ext3    defaults        1 2
proc                    /proc                   proc    defaults        0 0
sysfs                   /sys                    sysfs   defaults        0 0
LABEL=SWAP-sda3         swap                    swap    defaults        0 0
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/etc/fstab" 10L, 610C
```

use fdisk command to remove raid partition form hard disk


[root@localhost ~]# fdisk /dev/sda
The number of cylinders for this disk is set to 2610.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
(e.g., DOS FDISK, OS/2 FDISK)
Command (m for help): d
Partition number (1-8): 8
Command (m for help): d
Partition number (1-7): 7
Command (m for help): d
Partition number (1-6): 6
Command (m for help): w_

Swap partition step by step example and implementations

Linux uses the swap space configured on one or more hard drive partitions to store infrequently used programs and data. Swap space can extend the amount of effective RAM on your system.

However, if you don't have enough actual RAM, Linux may use the swap space on your hard drive as virtual memory for currently running programs. But you can't just buy extra RAM and eliminate swap space. Linux moves infrequently used programs and data to swap space even if you have gigabytes of RAM.

Normally, Linux (on a 32-bit Intel-style computer) can use a maximum 4GB of swap space in partitions no larger than 2GB. This 4GB can be spread over a maximum of eight partitions. The typical rule of thumb suggests that ***swap space should be two times the amount of RAM.*** For example if you have 1 GB of physical ram then need 2GB space for swap.

Generally Linux create swap during installation but for exam prospective you should be able to create it after installation. There are two method for it via fdisk utility of file method. We will create swap via both method.

Create Swap partition

To accomplish this task you must be login form root account. So first login from root and verify your hard disk status with **fdisk –l command** (This command will show that where your hard disk is mounted. You should use the mount point which show in the output of this command. For example if you see **/dev/hda** then you should use **fdisk /dev/hda** in next command. Or if you see **/dev/sdb** then you should use **fdisk /dev/sdb** in next command.

As you can see in image shown below that My hard disk is mounted as **/dev/sda** so I will use ***fdisk /dev/sda***)

```
[root@Sever ~]# fdisk -l

Disk /dev/sda: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          13      104391   83  Linux
/dev/sda2              14        1033     8193150   83  Linux
/dev/sda3            1034        1160     1020127+  82  Linux swap / Solaris
/dev/sda4            1161        1958     6409935    5  Extended
/dev/sda5            1161        1287     1020096   83  Linux
[root@Sever ~]# fdisk /dev/sda

The number of cylinders for this disk is set to 1958.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): _
```

file type of swap space is 82 so change file type to 82 of this newly created partition and save and exit from fdisk with w options. Run partprobe command to update kernel

```
Command (m for help): t
Partition number (1-6): 6
Hex code (type L to list codes): 82
Changed system type of partition 6 to 82 (Linux swap

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with
 busy.
The kernel still uses the old table.
The new table will be used at the next reboot.
Syncing disks.

[root@Server ~]#
[root@Server ~]# partprobe /dev/sda
```

verify with fdisk -l command that partition is successfully created

```
[root@Server ~]# fdisk -l

Disk /dev/sda: 21.4 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          13      104391   83  Linux
/dev/sda2              14        1288    10241437+  83  Linux
/dev/sda3            1289        1415     1020127+  82  Linux swap
/dev/sda4            1416        2610     9598837+   5  Extended
/dev/sda5            1416        1542     1020096   83  Linux
/dev/sda6            1543        1555      104391   82  Linux swap
[root@Server ~]# _
```

now format swap partition and on it

```
[root@Server ~]# mkswap /dev/sda6
Setting up swapspace version 1, size = 106889 kB
[root@Server ~]# swapon /dev/sda6
[root@Server ~]# vi /etc/fstab _
```

To permanently mount this partition makes its entry in /etc/fstab as shown in image

#vi /etc/fstab

```
LABEL=/                 /                       ext3    defaults        1 1
LABEL=/home             /home                   ext3    defaults        1 2
LABEL=/boot             /boot                   ext3    defaults        1 2
tmpfs                   /dev/shm                tmpfs   defaults        0 0
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
sysfs                   /sys                    sysfs   defaults        0 0
proc                    /proc                   proc    defaults        0 0
LABEL=SWAP-sda3         swap                    swap    defaults        0 0
/dev/sda6               swap                    swap    defaults        0 0
```

Create swap from file method

To use file method for swap space create a blank file of 100MB

```
[root@Server ~]# touch /swap
[root@Server ~]# dd if=/dev/zero of=/swap bs=1M count=100
100+0 records in
100+0 records out
104857600 bytes (105 MB) copied, 0.84868 seconds, 124 MB/s
[root@Server ~]# _
```

Now format this file with mkswap and on this swap space. To keep it on after reboot make its entry to rc.local file

```
[root@Server ~]# mkswap /swap
Setting up swapspace version 1, size = 104853 kB
[root@Server ~]# swapon /swap
#!/bin/sh
#
# This script will be executed *after* all the
# You can put your own initialization stuff in
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local
swapon /swap
```

You have created swap space with both method you can verify its run time status in /proc file system

```
[root@Server ~]# cat /proc/swaps
Filename                                Type            Size    Used
/dev/sda3                               partition       1020116 0
/dev/sda6                               partition       104380  0
/swap                                   file            102392  0
[root@Server ~]# _
```

To remove these partition and files

- ***Turn off swap space with swapoff command***
- ***Remove entry from rc.local***
- ***Remove entry from /etc/fstab***
- ***Reboot system***

RPM Red Hat's Package Manager

In this article from our series of RHCE exam guide we will learn how we can install or remove package in linux.

RPM is a powerful software manager. It can install, remove, query, and verify the software on your system. Rpm is more than a Red Hat specific tool. Many other modern distributions, such as Caldera and SuSe, use rpm too. In this article we will by no means provide comprehensive coverage of rpm. Instead, we will highlight the subset of options we have found useful in the real RHCE Exam.

Querying Your System

The first thing you should do is look and sees what software you have installed on your system. Here is the command to use:

#rpm -qa | more

In case you are unfamiliar with the command line, let me break this command down.

rpm
is the command name.It tells the computer you want to run the rpm program.

(-)
In linux, the set of letters following a dash (-) is called an option or switch.

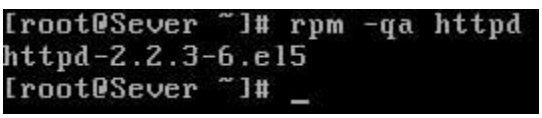
-q
tells rpm you want the query operation.

a following a in the -qa is a modifier for the query option which tells rpm you want to list all the packages.

| **more**
The | more part of the above command is not a feature of rpm at all. It is a standard linux way to show output one page at a time.

package info

Rpm is smart enough to use the package name without the version info.For example see in image



The package info is split into three pieces.

- The first piece is the package name.
- The second is the software version number.
- The third is the package build number.

All three are separated by dashes. The package build number is important incase if there is a more recent rpm build of a program with the same version

Installing New Software

You can install rpm from any location where you have it. In our example we will install it from RHEL dvd.

Command to install package is

#rpm -ivh <package name>

```
[root@Sever ~]# mount /dev/cdrom /mnt
mount: block device /dev/cdrom is write-protected, mounting read-only
[root@Sever ~]# cd /mnt/Server/
[root@Sever Server]# rpm -ivh telnet*
warning: telnet-0.17-38.el5.i386.rpm: Header V3 DSA signature: NOKEY, key ID 370
17186
Preparing...                ########################################### [100%]
        package telnet-0.17-38.el5 is already installed
[root@Sever Server]# _
```

-i is the install switch.
v for verbose messages in case if the installation fails.
h option shows our progress with hash marks.

A variation on an install is an upgrade. An upgrade is used when you want to put a more recent package in place of something that is currently installed . The upgrade syntax is exactly the same as an install, but you replace the **-i** with a **-U.** (Notice it is a capital U) If a new version of **telnet-server** comes out, rpm will take care of removing all the old pieces when you upgrade.

```
[root@Sever Server]# rpm -Uvh telnet-server*
warning: telnet-server-0.17-38.el5.i386.rpm: Header V3 DSA signature: NOKEY, key
 ID 37017186
Preparing...                ########################################### [100%]
   1:telnet-server          ########################################### [100%]
[root@Sever Server]# _
```

Sometimes a package is not removed cleanly. Here is the situation, you try to install something and rpm says its already installed. You then try to remove it, and rpm says that is not installed. What can you do?

#rpm -ivh --force package-1.0-5.i386.rpm

The **--force** option is your solution.It will install rpm in any conditions.

Dependencies are generally regarded as a good thing. Rpm has the capability to know if software has such prerequisites. In the real world, not everything on your system can always be from an rpm. So if you want to install rpm without checking dependencies you can use **--nodeps** options

#rpm -ivh --nodeps package-1.0-5.i386.rpm

```
[root@Sever Server]# rpm -Uvh telnet-server*
warning: telnet-server-0.17-38.el5.i386.rpm: Header V3 DSA signature: NOKEY, key
 ID 37017186
Preparing...                ########################################### [100%]
   1:telnet-server          ########################################### [100%]
[root@Sever Server]# rpm -Uvh telnet-server* --nodeps --force
warning: telnet-server-0.17-38.el5.i386.rpm: Header V3 DSA signature: NOKEY, key
 ID 37017186
Preparing...                ########################################### [100%]
   1:telnet-server          ########################################### [100%]
[root@Sever Server]# _
```

Removing Unwanted Software

A major advantage to a packaging system like rpm is its ease to erase software. Here is how you do it:

#rpm -e telnet-server

```
[root@Sever Server]# rpm -e telnet
[root@Sever Server]# rpm -e telnet-server
[root@Sever Server]# rpm -qa telnet*
[root@Sever Server]# _
```

fstab file define the mount points for partition. Before you can use the files in a directory, you need to mount that directory on a partition formatted to some readable filesystem. Linux normally automates this process using the **/etc/fstab** configuration file. You may encounter problems if connections are lost or media is removed. This cause error and these error are highly tested in RHCE exam.

RHCE Exam Questions

You are giving RHCE exam. Examiner gave you the Boot related problem and told to you that make successfully boot the System. When you started the system, System automatically asking the root password for maintenance. How will you fix that problem?

Troubleshooting of fstab

In this practical we will discuss how a faulty fstab file case error and how can you remove them.

Take back up and Open /etc/fstab file from vi command

```
[root@Server ~]# cp /etc/fstab /root/
[root@Server ~]# vi /etc/fstab _
```

default fstab file look like this

```
LABEL=/1                /                       ext3    defaults        1 1
LABEL=/boot1            /boot                   ext3    defaults        1 2
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
tmpfs                   /dev/shm                tmpfs   defaults        0 0
LABEL=/home             /home                   ext3    defaults        1 2
proc                    /proc                   proc    defaults        0 0
sysfs                   /sys                    sysfs   defaults        0 0
LABEL=SWAP-sda3         swap                    swap    defaults        0 0
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/etc/fstab" 10L, 610C
```

Description of /etc/fstab by Column, Left to Right

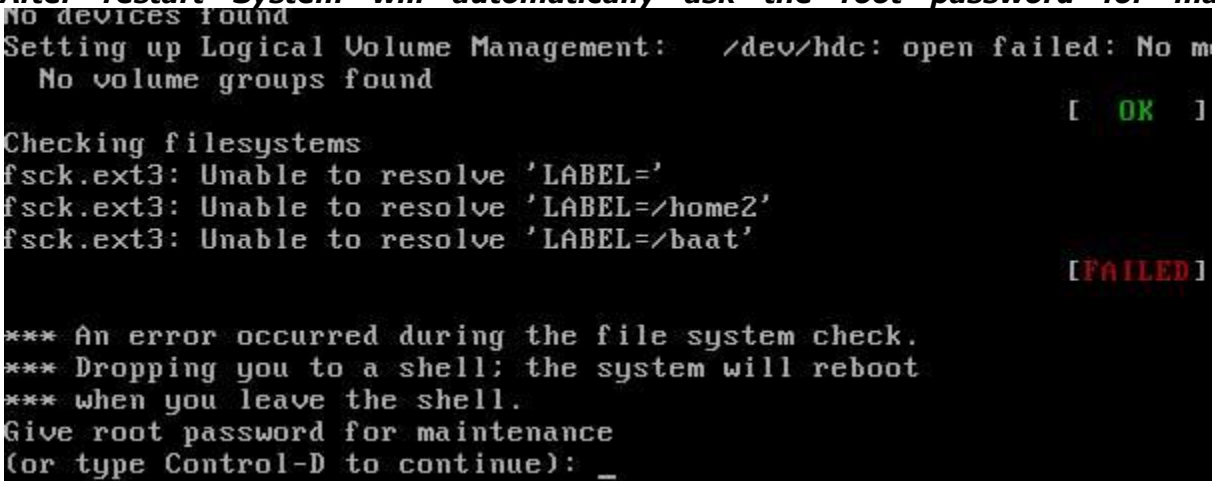
Label	Lists the device to be mounted
Mount Point	Notes the directory where the filesystem will be mounted
Filesystem	Describes the filesystem type. Valid filesystem types include ext, ext2, ext3, msdos,

Format	vfat, devpts, proc, tmpfs, udf, iso9660, nfs, smb, and swap.
Dump Value	Dump Value Either 0 or 1. A value of 1 means that data is automatically saved to disk by the dump(8) command when you exit Linux.
Filesystem Check Order	Filesystem Check Order Determines the order that filesystems are checked by fsck(8) during the boot process. The root directory (/) filesystem should be set to 1, and other local filesystems should be set to 2. Removable filesystems such as /mnt/cdrom should be set to 0, which means that they are not checked during the Linux boot process.

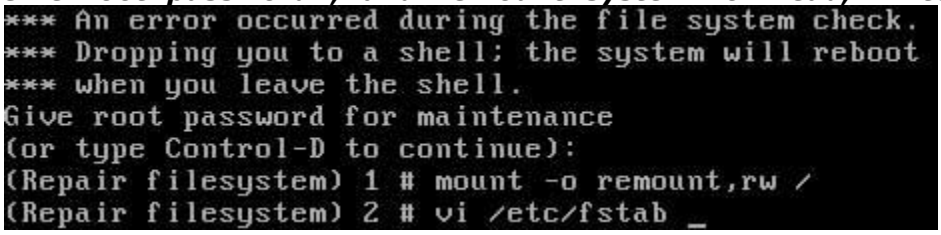
Now make some change in /etc/fstab file so it could be faulty as I did in this file

Save the change and restart the system

After restart System will automatically ask the root password for maintenance



Give root password , and remount system for read, write and open /etc/fstab file



Correct the fstab file and save the change [change will not save if you did not mount the system for read and write]

```
LABEL=/1                /                       ext3    defaults        1 1
LABEL=/boot1            /boot                   ext3    defaults        1 2
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
tmpfs                   /dev/shm                tmpfs   defaults        0 0
LABEL=/home             /home                   ext3    defaults        1 2
proc                    /proc                   proc    defaults        0 0
sysfs                   /sys                    sysfs   defaults        0 0
LABEL=SWAP-sda3         swap                    swap    defaults        0 0
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/etc/fstab" 10L, 610C
```

After saving the change reboot system

```
(Repair filesystem) 4 # reboot -f_
```

This time system will boot without any error.

How to increase virtual terminal in linux.

The **/etc/inittab** file holds instructions for your system on how to manage terminal devices. A line in the **/etc/inittab** file has four basic components: ***an ID, a runlevel, an action, and a process.***

Terminal devices are identified by ID numbers, beginning with 1 for the first device. The runlevel at which the terminal operates is usually 1. The action is usually respawn, which means to run the process continually. The process is a call to the mingetty, mgetty, or agetty with the terminal device name.

Wrong editing in this file could be dangerous even it could crash Linux system. We suggest you to take back up first before editing in this file.

```
#cp /etc/inittab  /root
```

With this file you can change default run level, increase virtual terminals and disable ALT+CTRL+DEL key combination to restart the system.

After taking backup open /etc/inittab file

```
[root@Server ~]# vi /etc/inittab _
```

Change Default Run Level

Linux have seven run levels. Functions of all run level are

```
# Default runlevel. The runlevels used by RHS are:
#   0 - halt (Do NOT set initdefault to this)
#   1 - Single user mode
#   2 - Multiuser, without NFS (The same as 3, if
#   3 - Full multiuser mode
#   4 - unused
#   5 - X11
#   6 - reboot (Do NOT set initdefault to this)
```

During system startup process Linux check this file to determines which runlevel it should be boot by looking at the initdefault directive in **/etc/inittab**. For example, the entry

id:5:initdefault:

shows a default starting point in runlevel 5, which is associated with the GUI

To change this default runlevel locate this tag id:5:initdefault: Now replace the value 5 to 3 as show here to boot system in run level 3 Save the file and restart the system it will boot now in run level 3.

```
#   5 - X11
#   6 - reboot (Do NOT set initdefault to this)
#
id:3:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit
```

How to disable ALT+CTRL+DEL key combination

By default ALT+CTRL+DEL key combination is used to restart the system. This default behavior is also controlled by this tag in /etc/inittab file.

```
# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

By some mean if you don't want to use ATL+CTRL+DEL key to restart the system. Put a hash # sign to this tag and save the file and restart the system. Now you cannot restart the by ATL+CTRL+DEL key.

```
# Trap CTRL-ALT-DELETE
#ca::ctrlaltdel:/sbin/shutdown -t3 -r now_
```

How to increase the number of virtual consoles

Virtual consoles are configured in /etc/inittab. By default, RHEL is configured with six virtual consoles. You can configure up to twelve virtual consoles in /etc/inittab.

Here are the default /etc/inittab entries for the first six virtual consoles:

```
# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

To increase the number of virtual console copy the configuration line of last virtual console and past just below the default line and change the number as shown in image. Save file and restart the system.

```
# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
7:2345:respawn:/sbin/mingetty tty7
8:2345:respawn:/sbin/mingetty tty8
9:2345:respawn:/sbin/mingetty tty9
```

After restart you can login in to increased terminals by pressing ALT+Funcation key combinations.

```
Red Hat Enterprise Linux Server release 5.1
Kernel 2.6.18-53.el5 on an i686

Server login: root
Password:
Last login: Sun Feb 14 21:07:30 on tty1
[root@Server ~]# tty
/dev/tty8
[root@Server ~]# _
```

Other use full commands

#init [run level]

```
[root@Server ~]# init 1_
```

With this command you can switch between run levels. For example to go on run level one type

#init 1 This will take you on run level one.

#telinit 1

```
[root@Server ~]# telinit 1_
```

Same as init. This will also take you on run level one.

#runlevel

```
[root@Server ~]# runlevel
N 3
[root@Server ~]# _
```

To know on which run level are you working now use **runlevel** commands

Linux job scheduling with at commands

at daemon allows you to run the command of your choice, once, at a specified time in the future.

You can set an at job to be run once. The **at** daemon works to the print process; jobs are spooled in the **/var/ spool/at directory** and run at the specified time.

You can use the **at daemon** to run the command or script of your choice. For the purpose of this article we are creating a simple script that will list the contain of current directory and send this output to terminal 2.

From the command line, you can run the **at** time command to start a job to be run at a specified time. That time can be now; in a specified number of minutes, hours, or days; or at the time of your choice. We will take several example to illustrate it more deeply. The **CTRL-D command** exits the at command shell and returns to your original command line interface.

Login from root and create a simple script file test.sh and grant it execute permission

```
[root@Server ~]# tty
/dev/tty2
[root@Server ~]# cat > test.sh
#/bin/sh
ls > /dev/tty2
date > /dev/tty2
[root@Server ~]# chmod +x test.sh
[root@Server ~]# _
```

Check date before scheduling job from at commands

```
[root@Server ~]# date
Sun Feb 14 21:29:40 IST 2010
[root@Server ~]# _
```

To run this script on 21 Feb 2010 after seven days you need to schedule at command as shown here

```
[root@Server ~]# at -f /root/test.sh now +7 days
job 3 at 2010-02-21 21:32
[root@Server ~]# _
```

To run this script after ten minutes you need to schedule at command as shown here

```
[root@Server ~]# at -f /root/test.sh now +10 minutes
job 4 at 2010-02-14 21:42
[root@Server ~]# _
```

To run this script now you need to schedule at command as shown here

```
[root@Server ~]# at -f /root/test.sh now
job 5 at 2010-02-14 21:32
[root@Server ~]# anaconda-ks.cfg  Desktop  dump install.log
 test.sh
Sun Feb 14 21:32:43 IST 2010

[root@Server ~]# _
```

To run this script on 10:15 AM you need to schedule at command as shown here

```
[root@Server ~]# at -f /root/test.sh 10:15 AM
job 2 at 2010-02-15 10:15
[root@Server ~]# _
```

To check the status of your jobs, so you can see if it will work, run the following job queue command:

```
[root@Server ~]# atq
4       2010-02-14 21:42 a root
2       2010-02-15 10:15 a root
3       2010-02-21 21:32 a root
1       2010-02-15 21:30 a root
[root@Server ~]# _
```

If there's a problem with the job, you can remove it with the atrm command. In this example you would remove job number 4 with the following command:

```
[root@Server ~]# atrm 4
[root@Server ~]# atq
2       2010-02-15 10:15 a root
3       2010-02-21 21:32 a root
1       2010-02-15 21:30 a root
[root@Server ~]# _
```

Securing At daemon

You may not want everyone to be able to run a job in the middle of the night. If your system have important security data, someone may download important data or worse, and it could be done before you discover the security violations.

Two files are used to control the behavior of at daemons

- ***/etc/at.allow If present then only users those name are in this file can use at daemons***
- ***/etc/at.deny If present then only user those name are in this file will not be able to use at daemons apart from these user all other can use at daemons***
- ***If both files are not present then only root can access at daemons***

For example create two user Vinita and nikita

```
[root@Server ~]# useradd vinita
[root@Server ~]# useradd nikita
[root@Server ~]# passwd -d vinita
Removing password for user vinita.
passwd: Success
[root@Server ~]# passwd -d nikita
Removing password for user nikita.
passwd: Success
[root@Server ~]# _
```

These files are formatted as one line per user; add user vinita to at.allow

```
[root@Server ~]# cat > /etc/at.allow
vinita
[root@Server ~]# _
```

To test login on other terminal from user vinita and schedule job from at commands

```
[vinita@Server ~]$ at 12:00
at> ls
at> <EOT>
job 7 at 2010-02-15 12:00
[vinita@Server ~]$ _
```

Now login on other terminal from nikita and schedule job form at commands

```
[nikita@Server ~]$ at
You do not have permission to use at.
[nikita@Server ~]$ _
```

Linux job scheduling with cron commands.

The **cron** system is basically a smart alarm clock. When the alarm sounds, Linux runs the commands of your choice automatically. You can set the alarm clock to run at all sorts of regular time intervals.

Linux installs the **cron daemon** (crond) by default. It's configured to check the **/var/spool/cron directory** for jobs by user. It also checks for scheduled jobs for the computer under **/etc/ crontab** and in the **/etc/cron.d directory.**

Login form root and check system date, and run crontab command to schedule job

```
[root@Server ~]# tty
/dev/tty2
[root@Server ~]# date
Sun Feb 14 21:46:09 IST 2010
[root@Server ~]# crontab -e_
```

In open file you can schedule job. There are 6 field in this file 5 for time and one for commands.

Field	Value
minute	0–59
hour	Based on a 24-hour clock; for example, 23 = 11 P.M.
day of month	1–31
month	1–12, or jan, feb, mar, etc.
day of week	0–7; where 0 and 7 are both Sunday; or sun, mon, tue, etc.
command	The command you want to run

If you see an asterisk in any column, cron runs that command for all possible values of that column. For example, an * in the minute field means that the command is run every minute during the specified hour(s). Consider another example, as shown here:

11 5 3 5 * ls

This line runs the ls command every May 3 at 5:11 A.M. The asterisk in the day of week column simply means that it does not matter what day of the week it is; crontab still runs the ls command at the specified time.

For example time in my system is 21:46 and date is 14 Feb Sunday. (See image above). Now I will set cron to display the output of ls commands on tty2 at 21:50

#crontab -e

```
50 21 14 02 * ls > /dev/tty2
```

save file and quit

In real life you do not have to restart cron every time you make a change because cron always checks for changes, But so far exams concern we suggest you to restart cron whenever you made change.

```
[root@Server ~]# service crond restart
Stopping crond:                                            [  OK  ]
Starting crond:                                            [  OK  ]
[root@Server ~]# _
```

Wait for four minute and on 21:50 you will get the output of ls command on tty2

```
[root@Server ~]# Desktop  anaconda-ks.cfg  dump install.log
 test.sh

[root@Server ~]# date
Sun Feb 14 21:50:12 IST 2010
[root@Server ~]# _
```

Setting Up cron for Users

Each user can use the **crontab** command to create and manage cron jobs for their own accounts. There are four switches associated with the **crontab** command:

- ***-u user Allows the root user to edit the crontab of another specific user.***
- ***-l Lists the current entries in the crontab file.***
- ***-r Removes cron entries.***
- ***-e Edits an existing crontab entry. By default, crontab uses vi.***

If you want to set up cron entries on your own account, start with the **crontab -e** command.

Securing cron daemon

You may not want everyone to be able to run a job in the middle of the night. If your system have important security data, someone may download important data or worse, and it could be done before you discover the security violations.

Two files are used to control the behavior of crond daemons

- ***/etc/cron.allow If present then only users those name are in this file can use crond daemons***
- ***/etc/cron.deny If present then only user those name are in this file will not be able to use crond daemons apart from these user all other can use cron daemons***
- ***If both files are not present then only root can access cron daemons***

For example create two user Vinita and nikita

```
[root@Server ~]# useradd vinita
[root@Server ~]# useradd nikita
[root@Server ~]# passwd -d vinita
Removing password for user vinita.
passwd: Success
[root@Server ~]# passwd -d nikita
Removing password for user nikita.
passwd: Success
[root@Server ~]# _
```

These files are formatted as one line per user; add user nikita to cron.allow

```
[root@Server ~]# cat > /etc/cron.allow
nikita
[root@Server ~]# _
```

To test login on other terminal from user nikita and schedule job from cron commands

```
[nikita@Server ~]$ crontab -e_


* * * * * ls > /dev/tty3_


"/tmp/crontab.XXXXh6jT4o" 1L, 25C written
crontab: installing new crontab
[nikita@Server ~]$ _
```

Now login on other terminal from vinita and schedule job form cron commands

```
[vinita@Server ~]$ crontab -e
You (vinita) are not allowed to use this program
See crontab(1) for more information
[vinita@Server ~]$ _
```

Remove root password in Linux

In this article from our series of RHCE exam guide we will learn how to ***remove root password***.

Example Question :-

You are new System Administrator and from now you are going to handle the system and your main task is Network monitoring, Backup and Restore. But you donot know the root password. Change the root password to redhat.

To remove root password in linux follow this guide

When System Successfully boot, it will ask for username and password. But you don't know the root's password. To change the root password you need to boot the system into single user mode. You can pass the kernel arguments from the boot loader.

Reboot system with alt + ctrl + del key combinations
Press space bar on boot loader screen

This will drop you in OS selection screen
Select linux (If you have multibooting then select linux)

From grub boot loader screen select kernel parameter line and press *e*

In kernel editing mode press *space bar* and write *s* in the end of line just after the rhgb quiet and press *enter key* that will bring in grub bootloader screen

```
[ Minimal BASH-like line editing is supported.  For the first word, TAB
  lists possible command completions.  Anywhere else TAB lists the possible
  completions of a device/filename.  ESC at any time cancels.  ENTER
  at any time accepts your changes.]

grub edit> kernel /vmlinuz-2.6.18-8.el5 ro root=LABEL=/1 rhgb quiet s
```

On grub boot loader screen press *b*

```
  GNU GRUB  version 0.97  (638K lower / 522176K upper memory)

 root (hd0,0)
 kernel /vmlinuz-2.6.18-8.el5 ro root=LABEL=/1 rhgb quiet s
 initrd /initrd-2.6.18-8.el5.img
```

This change will tell to boot system in single user mode.
After loading essential modal kernel will drop in rescue mode with root prompt

```
Remounting root filesystem in read-write mode:                 [  OK  ]
Mounting local filesystems:                                    [  OK  ]
Enabling local filesystem quotas:                              [  OK  ]
Enabling /etc/fstab swaps:                                     [  OK  ]
sh-3.1# _
```

now run passwd command to reset root password
and init 5 command to run system in graphic mode or you can just reboot system to on its default run level

```
sh-3.1# passwd
Changing password for user root.
New UNIX password:
BAD PASSWORD: it is too simplistic/systematic
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
sh-3.1# init 5_
```

Recover grub.conf password and remove kernel panic error

The **Grand Unified Bootloader (GRUB)** is a multiboot boot loader used for Linux With GRUB, users can select operating systems to run from a menu interface displayed when a system boots up. Use arrow keys to move to an entry and press ENTER.

As suggested by the Red Hat exam requirements, for the RHCT exam, you need to know how to use the GRUB menu to boot into different runlevels, and diagnose and correct boot failures arising from boot loader errors. We have covered how to boot system in different run level already in pervious article check that for run level related question.

How to boot system in different run level

In this article we will cover two most command booting issue. kernel panic error and grub password

RHCE Exam Questions

You are giving RHCE exam. Examiner gave you the Boot related problem and told to you that make successfully boot the System. While booting system, you saw some error and stop the boot process by displaying some error messages.
Kernel Panic - not syncing: Attempted to kill init!
And no further boot process. What you will do to boot the system.

If you are getting the Kernel panic error, it means it is boot loader related problem. Redhat Enterprise Linux uses the **GRUB boot loader**. You can pass the kernel parameter from the boot loader as well as you can correct the kernel parameter passing from boot loader from GRUB screen at boot time.

RHEL Linux Kernel panic error

For this practical we will modify **grub.conf** So you can understand what exactly case the kernel panic error.

always take back up before modifying **grub.conf** parameter

```
#cp /etc/grub.conf /root
```

open /etc/grub.conf from vi command

```
[root@Server ~]# cp /etc/grub.conf /root
[root@Server ~]# vi /etc/grub.conf _
```

Default grub.conf file look like this We suggest you to cram up this file

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making cha
# NOTICE:  You have a /boot partition.  This means that
#          all kernel and initrd paths are relative to /bo
#          root (hd0,0)
#          kernel /vmlinuz-version ro root=/dev/sda2
#          initrd /initrd-version.img
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux Server (2.6.18-53.el5)
        root (hd0,0)
        kernel /vmlinuz-2.6.18-53.el5 ro root=LABEL=/ rhgb
        initrd /initrd-2.6.18-53.el5.img
```

Now change kernel line as show below [change forward slash / to backward slash \ ]

```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux Server (2.6.18-53.el5)
        root (hd0,0)
        kernel /vmlinuz-2.6.18-53.el5 ro root=LABEL=\_rhgb quiet
        initrd /initrd-2.6.18-53.el5.img
```

Save file with :wq and reboot the system

On Restart you will get kernel panic error

```
setuproot: moving /dev failed: No such file or directory
setuproot: error mounting /proc: No such file or directory
setuproot: error mounting /sys: No such file or directory
switchroot: mount failed: No such file or directory
Kernel panic - not syncing: Attempted to kill init!
```

How to remove kernel panic error

Reboot system and press space bar on boot menu and select kernel line

```
  GNU GRUB  version 0.97  (638K lower / 522176K upper memory)

 kernel /vmlinuz-2.6.18-53.el5 ro root=LABEL=\ rhgb quiet
 initrd /initrd-2.6.18-53.el5.img
```

Now press e for edit and you will see the wrong entry of kernel line in grub.conf

```
   completions of a device/filename.  ESC at any time cancels.  ENTER
   at any time accepts your changes.]

grub edit> kernel /vmlinuz-2.6.18-53.el5 ro root=LABEL=\ rhgb quiet
```

Correct the kernel parameter replace backward slash \ to forward slash / and press enter

to save

This will correct this error temporary.You will get same error after rebooting the system . As change here will not change the default faulty grub.conf so after booting system don't forget to Correct the kernel parameter replace backward slash \ to forward slash /

#vi /etc/grub.conf

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making cha
# NOTICE:  You have a /boot partition.  This means that
#          all kernel and initrd paths are relative to /bo
#          root (hd0,0)
#          kernel /vmlinuz-version ro root=/dev/sda2
#          initrd /initrd-version.img
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux Server (2.6.18-53.el5)
        root (hd0,0)
        kernel /vmlinuz-2.6.18-53.el5 ro root=LABEL=/ rhgb
        initrd /initrd-2.6.18-53.el5.img
```

How remove grub.conf password

By booting system in **single mode** one can easily recovered root password. This could case great security risk. For this every Linux system administrator password protect the **grub.conf** Two types of password can be set on grub.conf one to edit the parameter in **grub.conf** during boot process and another to boot operating system. But what if you lost both root and grub.conf password.

For this practical open grub.conf file

#vi /etc/grub.conf

Set password for editing just below the hidemenu option and Set password for booting the OS just below the title menu

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making
# NOTICE:  You have a /boot partition.  This means tha
#          all kernel and initrd paths are relative to
#          root (hd0,0)
#          kernel /vmlinuz-version ro root=/dev/sda2
#          initrd /initrd-version.img
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
password 123
title Red Hat Enterprise Linux Server (2.6.18-53.el5)
password 123456_
        root (hd0,0)
        kernel /vmlinuz-2.6.18-53.el5 ro root=LABEL=/
        initrd /initrd-2.6.18-53.el5.img
```

Now save file with :wq and restart the system

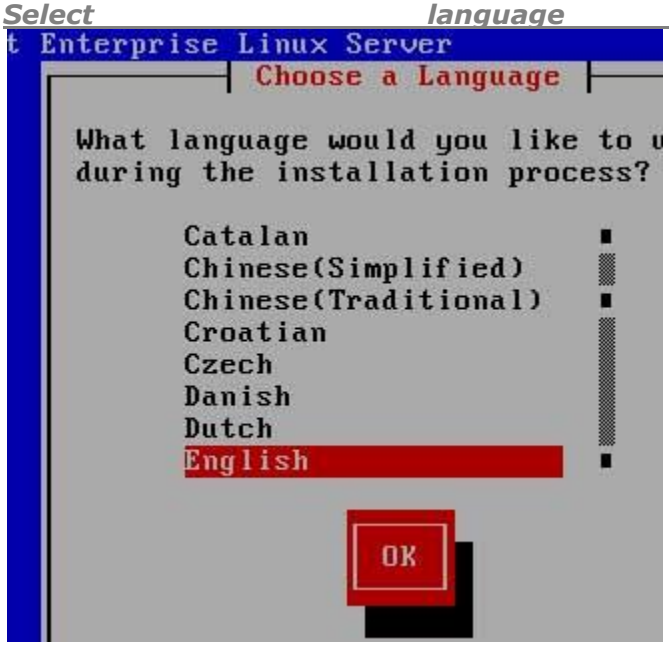
Now press space bar on boot menu and press e to edit It will ask to give the password which you set below the hidemenu

After it on boot screen it will ask OS password which you set under the title menu

Now assume that you lost all three **root, grub.conf** and **boot loader** password. How will you recover these passwords.

Boot system from Linux CD and give linux rescue command on boot screen

Select language to English



Select Keyboard layout to US

Press enter on continue and it will search for linux on hard disk

We don't need networking for this operation so select no

Rescue mode will mount system image under the /mnt/sysimage folder press ok

now change chroot to /mnt/sysimage and open /etc/grub.conf

```
Your system is mounted under the /mnt/sysimage directory.
When finished please exit from the shell and your system

sh-3.1# chroot /mnt/sysimage
sh-3.1# vi /etc/grub.conf _
```

Remove both hidemenu and title password and save file

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making cha
# NOTICE:  You have a /boot partition.  This means that
#          all kernel and initrd paths are relative to /bo
#          root (hd0,0)
#          kernel /vmlinuz-version ro root=/dev/sda2
#          initrd /initrd-version.img
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux Server (2.6.18-53.el5)
        root (hd0,0)
        kernel /vmlinuz-2.6.18-53.el5 ro root=LABEL=/ rhgb
        initrd /initrd-2.6.18-53.el5.img
```

Now reboot the system and remove Linux CD from CDROM

```
sh-3.1# reboot -f_
```

After reboot there should be no password on OS selection screen

```
   GNU GRUB  version 0.97  (638K lower / 522176K upper memory)

 root (hd0,0)
 kernel /vmlinuz-2.6.18-8.el5 ro root=LABEL=/1 rhgb quiet
 initrd /initrd-2.6.18-8.el5.img
```

And on boot screen

```
  Booting 'Red Hat Enterprise Linux Server

root (hd0,0)
 Filesystem type is ext2fs, partition type
kernel /vmlinuz-2.6.18-53.el5 ro root=LABE
   [Linux-bzImage, setup=0x1e00, size=0x1b
```

We have recovered both boot loader and OS selection menu password now you easily recovered **root password** by booting system in single mode.

SERVICES

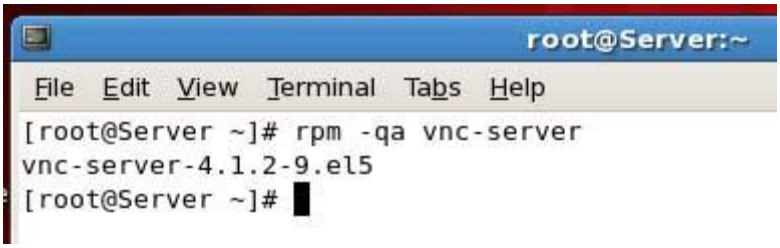
How to configure linux vnc server step by step guide Example and Implementation

VNC server is a used to share desktop with remote computer. VNC works on client server architecture. To share desktop you need **vnc-server** package and to access from other computers you need vnc-viewer. In this tutorials I will show you how to configure **vnc server**.

For demonstration purpose we will use two linux systems. Both systems should have graphics installed.

To configure VNC- Server

Boot system in init 5 or graphic mode. vnc-server rpm is required to configure server check it if not found install it.



now click on preferences from system and select remote desktop

This will launch a new window where you can set sharing and security for remote desktop

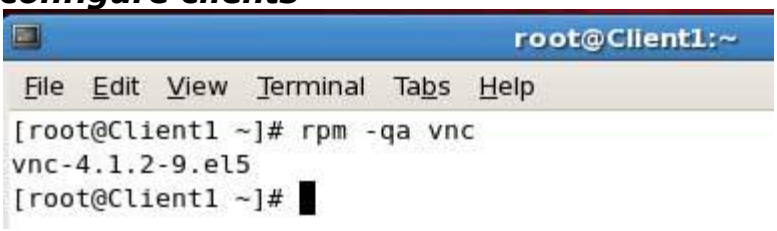
Allow other users to view your desktop check this option if you to present your desktop on other computer

Allow other users to control your desktop Check this options if you

want to grant permission to control user desktop to other user
In security tab you can set password for the user who want to connect with server [Recommended]

Configure Linux client

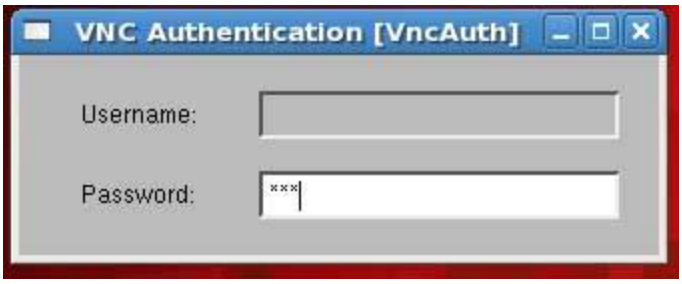
Go on client system and ping server. vnc-viewer rpm is required to configure clients



check it and if not found install
Now select accessories from application menu and click on vnc viewer

this will launch a window Give vnc server ip in it and click on ok

Once connected it will ask for password Give the password which you set on server



On server side it will show a pop up and ask for permission click on allow

After getting permission from server side you can use server desktop on client side

How to configure linux print server step by step guide Example and Implementation

Linux uses the **Common UNIX Printing System**, also known as CUPS. **CUPS** uses the Internet Printing Protocol (IPP) to allow local printing and print sharing. The ***/etc/cups/*** directory stores all the configuration files for printing. However, these files can be easily managed with the Printer Configuration Tool in Linux.

Exam question Raw (Model) printer named printer1 is installed and shared on 192.168.0.254. You should install the shared printer on your PC to connect shared printer using IPP Protocols.

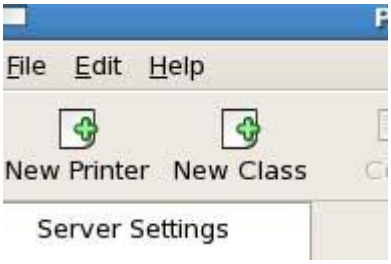
Exam question Raw printer named printerx where x is your station number is installed and shared on server1.example.com. Install the shared printer on your PC to connect shared printer using IPP Protocols. Your server is 192.168.0.254.

> ***Before you can use any printer, you first have to install it on a Linux system on your network. To start the Printer Configuration Tool, go to the System menu on the top panel and select Administration, Printing or execute the command system-config-printer.***

> ***If no printers are available for the system, only the Server Settings view is available for selection. If local printers are configured, a Local Printers menu will available.***

Install new printer

click New Printer on the toolbar.



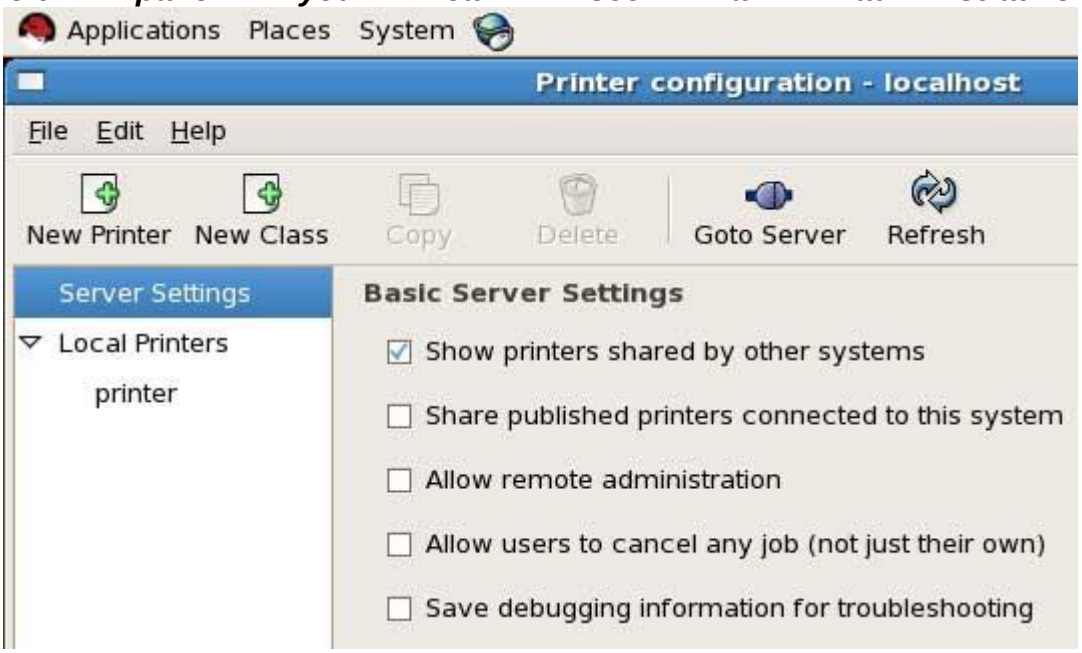
In the dialog window that appears, accept the default queue name or change it to a short, descriptive name that begins with a letter and does not contain spaces. Then select printer from list and click on forward and click on finsh.

spool directories
When your system prints a file, it makes use of special directories called spool directories. The location of the spool directory is obtained from the printer's entry in its configuration file. On Linux, the spool directory is located at /var/spool/cups under a directory with the name of the printer.

print job
A print job is a file to be printed. When you send a file to a printer, a copy of it is made and placed in a spool directory set up for that printer.

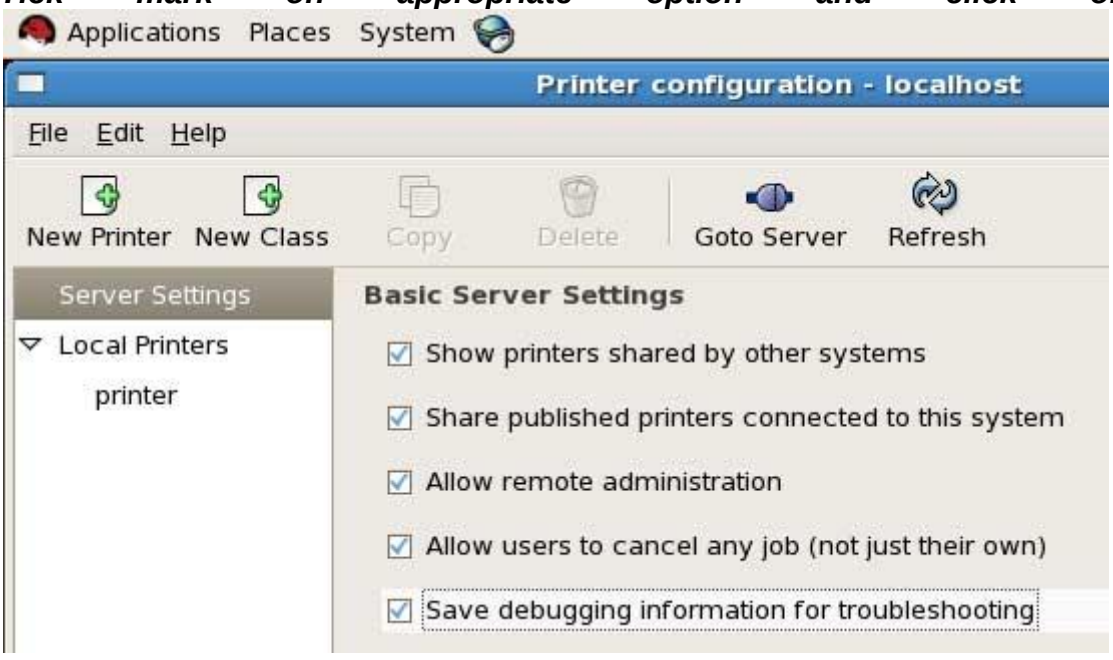
classes
CUPS features a way to let you select a group of printers to print a job instead of selecting just one. That way, if one printer is busy or down, another printer can be automatically selected to perform the job. Such groupings of printers are called classes. Once you have installed your printers, you can group them into different classes.

Once you have successfully installed local printer it will show in right pane. and in left pane you can see all administrative options.



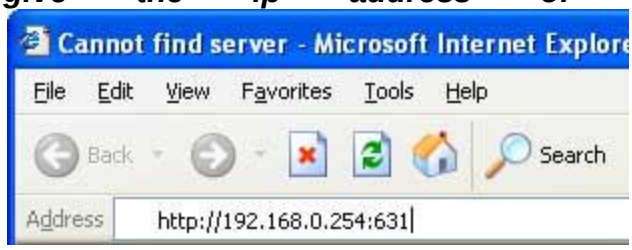
- ***To view shared printer on other system Tick mark on first option***
- ***To share locally attached printer tick mark on second option***
- ***To allow remote administration of this printer check mark on third option***

Tick mark on appropriate option and click on apply

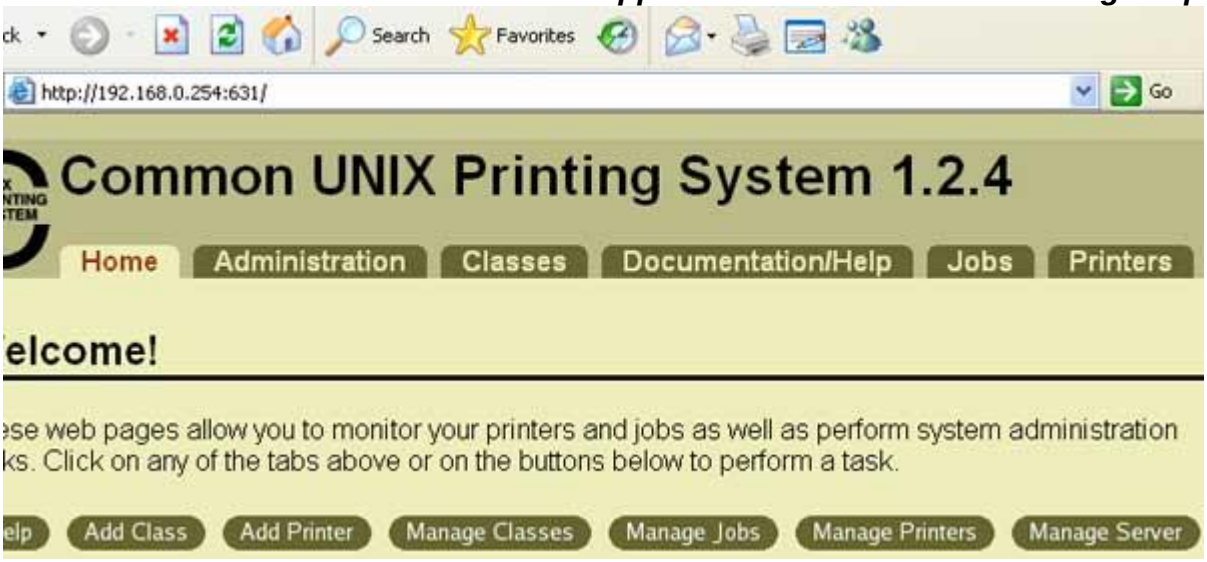


configure window clients

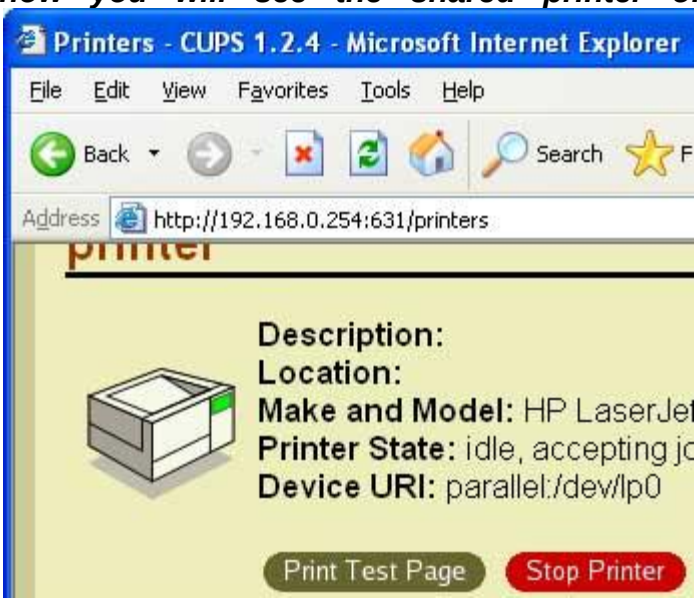
Go on window system and ping from printer server and open internet explorer and give the ip address of server with printer port 631



This will launch CUPS web application click on manage printer



now you will see the shared printer on server click on print test page



A test page will be send on printer server copy this url of printer

click on start button select printer and fax and click on add new printer. this will launch add new printer wizard click next on welcome screen and select network printer

On this screen select internet printer and paste the url which you copied from internet explorer

Install appropriate driver from list or use have disk option you have drive cd and

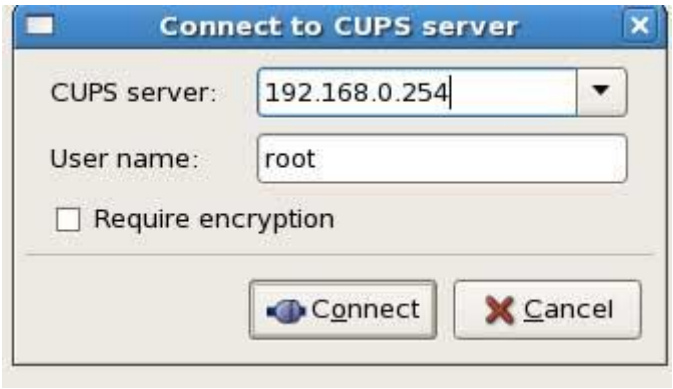
click next. On next screen set this printer defaults and click on next and finish.

Remote administration of print server

Go on linux system and ping from server and click on printing from administration menu

Now click on go to server

Now give print server ip address

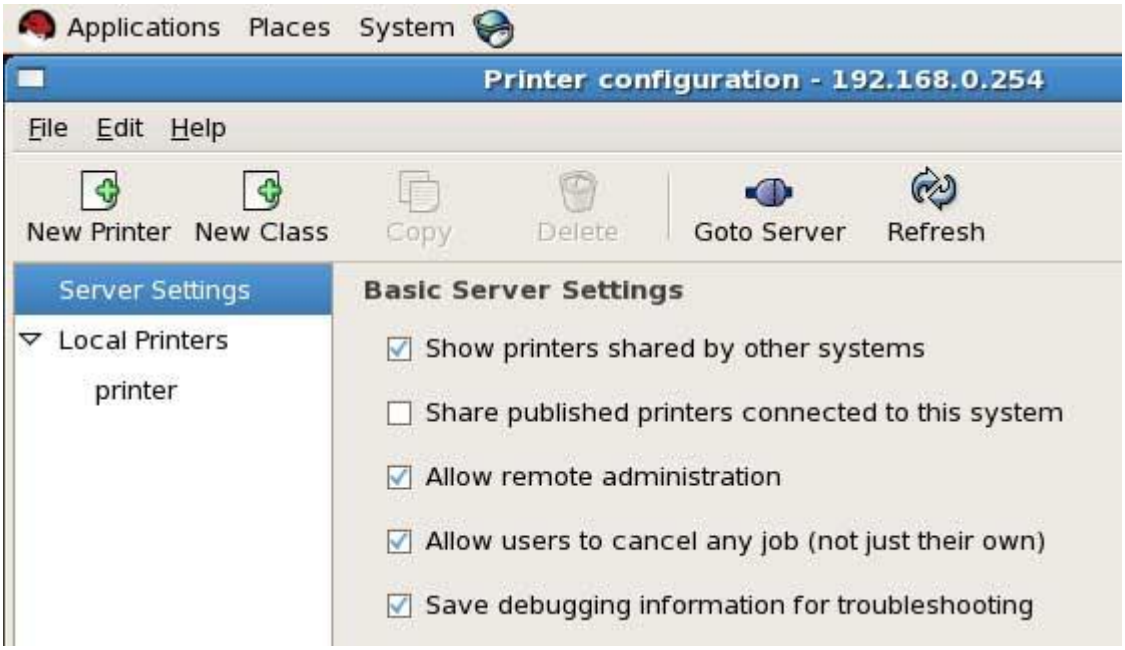


It will take few minute to connect from server depending on network speed

Now give root password to connect printer server

you can see all print administrative Manu in right pane Once you have connected

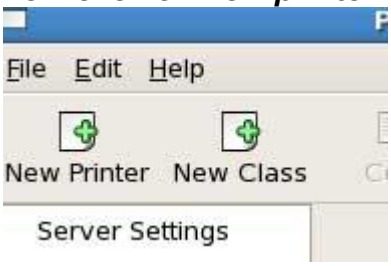
with sever



configure Linux clients

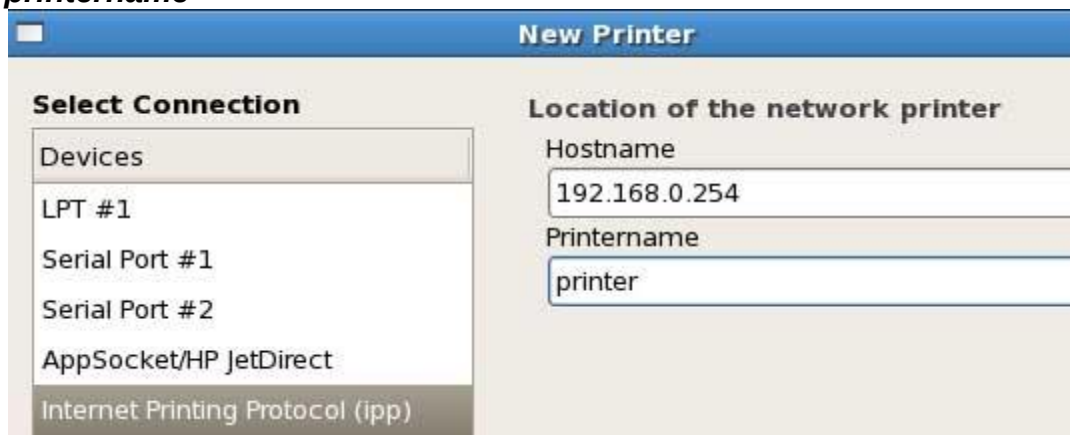
Go on linux system and ping from server and click on printing from administration menu

Now click on new printer



Click on forward In the next New Printer screen, select the type of connection to

internet printing protocols and in hostname give server ip and printer name in printername



select the appropriate model. If multiple drivers are available, select the one most appropriate for your configuration. If you do not want to choose the default and click forward and finish. The main Printer Configuration window should now include the name of your printer.

To print test page click on print test page and a test page will send to print server

Managing Printers from the Command-Line

The **lpadmin** command enables you to perform most printer administration tasks from the command-line.

```
[root@Client1 ~]# lpadmin
Usage:

    lpadmin [-h server] -d destination
    lpadmin [-h server] -x destination
    lpadmin [-h server] -p printer [-c add-class] [-i interface] [-m model
                        [-r remove-class] [-v device] [-D description]
                        [-P ppd-file] [-o name=value]
                        [-u allow:user,user] [-u deny:user,user]
```

lpc **To view all known queues**
lpr **To send print requests to any local print queue**
lpq **To see the print queue**
lprm **To delete the jobs of your choice use it with the job number**
lp **To print any file.**

```
[root@Client1 ~]# lpadmin -p 192.168.0.254
[root@Client1 ~]# lp test
request id is printer-2 (1 file(s))
[root@Client1 ~]#
```

How to configure linux telnet server step by step guide Example and Implementation

telnet server is used to login into another system. You can use the **telnet** command to log in remotely to another system on your network. The system can be on your local area network or available through an Internet connection. **Telnet** operates as if you were logging in to another system from a remote terminal. You will be asked for a login name and password. In effect, you are logging in to another account on another system. In fact, if you have an account on another system, you could use Telnet to log in to it.

You invoke the Telnet utility with the keyword **telnet**. If you know the name of the site you want to connect with, you can enter telnet and the name of the site on the Linux command line.

CAUTION The original version of Telnet is noted for being very insecure. For secure connections over a network or the Internet, you should use the Secure Shell (SSH). We will cover SSH server in next article. SSH operate in the same way as the original but use authentication and encryption to secure the Telnet connection. Even so, it is advisable never to use Telnet to log in to your root account. That why by defaults root account is disable for root login.

Configure telnet server

In this example we will configure a telnet server and will invoke connection from client side.

For this example we are using three systems one linux server one linux clients and one window clients.

- ***A linux server with ip address 192.168.0.254 and hostname Server***
- ***A linux client with ip address 192.168.0.1 and hostname Client1***
- ***A windows xp system with ip address 192.168.0.2 and hostname Client2***
- ***Updated /etc/hosts file on both linux system***
- ***Running portmap and xinetd services***
- ***Firewall should be off on server***

We suggest you to review that article before start configuration of telnet server. Once you have completed the necessary steps follow this guide.

Four rpm are required to configure telnet server. telnet, telnet-server, portmap, xinetd check them if not found then install

```
[root@Server ~]# rpm -qa telnet
telnet-0.17-38.el5
[root@Server ~]# rpm -qa telnet-server
telnet-server-0.17-38.el5
[root@Server ~]# rpm -qa portmap
portmap-4.0-65.2.2.1
[root@Server ~]# rpm -qa xinetd
xinetd-2.3.14-10.el5
[root@Server ~]# _
```

Now check telnet, portmap, xinetd service in system service it should be on

#setup
Select System service from list
[*]portmap
[*]xinetd
[*]telnet

Now restart xinetd and portmap service

```
[root@Server ~]# service portmap restart
Stopping portmap:                                          [  OK  ]
Starting portmap:                                          [  OK  ]
[root@Server ~]# service xinetd restart
Stopping xinetd:                                           [  OK  ]
Starting xinetd:                                           [  OK  ]
[root@Server ~]# _
```

To keep on these services after reboot on then via chkconfig command

```
[root@Server ~]# chkconfig portmap on
[root@Server ~]# chkconfig xinetd on
[root@Server ~]# _
```

After reboot verify their status. It must be in running condition

```
[root@Server ~]# service portmap status
portmap (pid 3430) is running...
[root@Server ~]# service xinetd status
xinetd (pid 3462) is running...
[root@Server ~]# _
```

Create a normal user named vinita

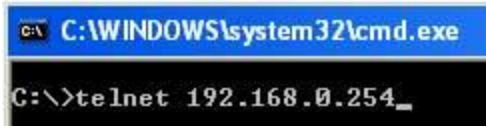
On Linux client

ping from telnet server and run telnet command and give user name and password

```
[root@Client1 ~]# telnet 192.168.0.254
Trying 192.168.0.254...
Connected to 192.168.0.254 (192.168.0.254).
Escape character is '^]'.
Red Hat Enterprise Linux Server release 5.1
Kernel 2.6.18-53.el5 on an i686
login: vinita
Last login: Sun Feb 14 22:56:47 from Client1
[vinita@Server ~]$ _
```

On Window client

ping from telnet server and run telnet command



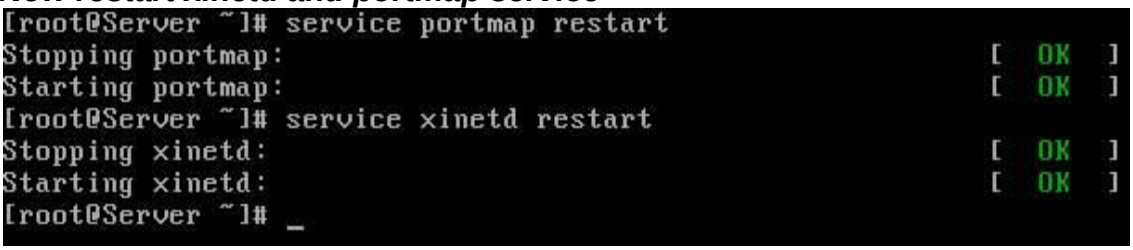
Give user name and password

How to enable root login from telnet server

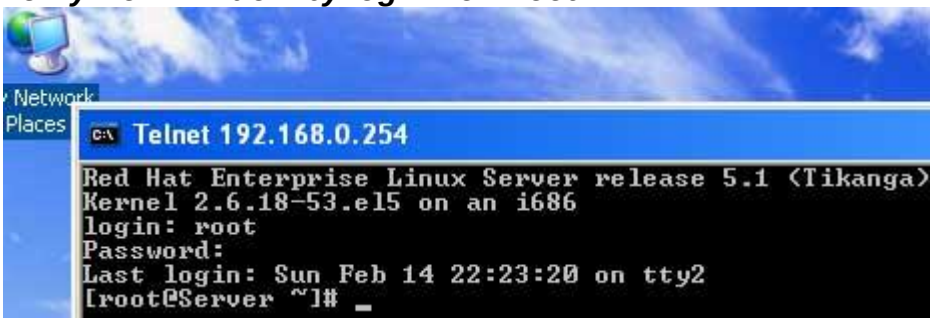
On linux server open file securetty

In the end of file add pts/0 to enable one telnet session for root. if you need to open more telnet session for root and add more pts/1 pts/2 and so on.

Now restart xinetd and portmap service



Verfiy from window by login from root



How to configure linux ssh server step by step guide Example and Implementation

Telnet and FTP are well-known protocol but they send data in plain text format, which can be captured by someone using another system on the same network, including the Internet.

On the other hand, all data transferred using **OpenSSH** tools is encrypted, making it inherently more secure. The OpenSSH suite of tools includes **ssh** for securely logging in to a remote system and executing remote commands, **scp** for encrypting files while transferring them to a remote system, and **sftp** for secure FTP transfers.

OpenSSH uses a server-client relationship. The system being connected to is referred to as the **server**. The system requesting the connection is referred to as the **client**. A system can be both an SSH server and a client. **OpenSSH** also has the added benefits of X11 forwarding and port forwarding.

X11 forwarding, if enabled on both the server and client, allows users to display a graphical application from the system they are logged in to on the system they are logged in from.

Port forwarding allows a connection request to be sent to one server but be forwarded to another server that actually accepts the request.

In this article we will discusses how to use **OpenSSH**, both from the server-side and the client-side.

Configuring the ssh Server

The openssh-server RPM package is required to configure a Red Hat Enterprise Linux system as an OpenSSH server. If it is not already installed, install it with rpm commands as described in our pervious article. After it is installed, start the service as root with the command service sshd start . The system is now an SSH server and can accept connections. To configure the server to automatically start the service at boot time, execute the command chkconfig sshd on as root. To stop the server, execute the command service sshd stop. To verify that the server is running, use the command service sshd status.

Configure ssh server

In this example we will configure a ssh server and will invoke connection from client side.

For this example we are using two systems one linux server one linux clients . To complete these per quest of ssh server Follow this link

per quest of ssh server

- ***A linux server with ip address 192.168.0.254 and hostname Server***
- ***A linux client with ip address 192.168.0.1 and hostname Client1***
- ***Updated /etc/hosts file on both linux system***
- ***Running portmap and xinetd services***
- ***Firewall should be off on server***

We have configured all these steps in our pervious article.

We suggest you to review that article before start configuration of ssh server. Once you have completed the necessary steps follow this guide.

Three rpm are required to configure ssh server. openssh-server, portmap, xinetd check them if not found then install

```
[root@Server ~]# rpm -qa openssh-server
openssh-server-4.3p2-24.el5
[root@Server ~]# rpm -qa portmap
portmap-4.0-65.2.2.1
[root@Server ~]# rpm -qa xinetd
xinetd-2.3.14-10.el5
[root@Server ~]# _
```

Now check sshd, portmap, xinetd service in system service it should be on

#setup
Select System service from list
[*]portmap
[*]xinetd
[*]sshd

Now restart xinetd and portmap and sshd service

```
[root@Server ~]# service portmap restart
Stopping portmap:                                          [  OK  ]
Starting portmap:                                          [  OK  ]
[root@Server ~]# service xinetd restart
Stopping xinetd:                                           [  OK  ]
Starting xinetd:                                           [  OK  ]
[root@Server ~]# _
[root@Server ~]# service sshd restart
Stopping sshd:                                             [  OK  ]
Starting sshd:                                             [  OK  ]
[root@Server ~]# chkconfig sshd on
[root@Server ~]# _
```

To keep on these services after reboot on then via chkconfig command

```
[root@Server ~]# chkconfig portmap on
[root@Server ~]# chkconfig xinetd on
[root@Server ~]# _
```

After reboot verify their status. It must be in running condition

```
[root@Server ~]# service portmap status
portmap (pid 3430) is running...
[root@Server ~]# service xinetd status
xinetd (pid 3462) is running...
[root@Server ~]# _
```

Create a normal user named vinita

```
[root@Server backup]# useradd vinita
[root@Server backup]# passwd vinita
Changing password for user vinita.
New UNIX password:
BAD PASSWORD: it is WAY too short
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[root@Server backup]#
```

On Linux client

ping from ssh server and run ssh command and give root password

```
[root@Client1 ~]# ssh 192.168.0.254
The authenticity of host '192.168.0.254 (192.168.0.254)' can't be established.
RSA key fingerprint is 66:83:74:ed:06:95:15:6c:44:6d:aa:43:ef:87:9e:cf.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.254' (RSA) to the list of known hosts.
root@192.168.0.254's password:
Last login: Sun Feb 14 22:57:07 2010 from Client1
[root@Server ~]# _
```

By default ssh command will enable root session. If you want to login from normal user then specify his name with -l options.

```
[root@Client1 ~]# ssh 192.168.0.254 -l vinita
vinita@192.168.0.254's password:
Last login: Sun Feb 14 22:57:34 2010 from Client2
[vinita@Server ~]$ _
```

With ssh you can run any command on server without login (user password require)

```
[root@Client1 ~]# ssh root@192.168.0.254 ls /root
root@192.168.0.254's password:
anaconda-ks.cfg
Desktop
dump
install.log
install.log.syslog
test.sh
[root@Client1 ~]# _
```

How to configure linux web server step by step guide Example and Implementation

When you view a web page over the Internet, the code to create that page must be retrieved from a server somewhere on the Internet. The server that sends your web browser the code to display a web page is called a web server. There are countless web servers all over the Internet serving countless websites to people all over the world. Whether you need a web server to host a website on the Internet a Red Hat Enterprise Linux server can function as a web server using the **Apache HTTP server**. The Apache HTTP server is a popular, open source server application that runs on many UNIX-based systems as well as Microsoft Windows.

Exam question 1 There are two sites www.vinita.com and www.nikita.com. Both sites are mappings to 192.168.0.X IP address where X is your Host address. Configure the Apache web server for these sites to make accessible on web

Configure web server

In this example we will configure a **web server.**

necessary rpm for web server is httpd, httpd-devel and apr check them for install

```
[root@Server ~]# rpm -qa http*
httpd-manual-2.2.3-6.el5
httpd-2.2.3-11.el5
httpd-2.2.3-6.el5
httpd-manual-2.2.3-11.el5
httpd-devel-2.2.3-6.el5
[root@Server ~]# rpm -qa apr*
apr-docs-1.2.7-11
apr-util-docs-1.2.7-6
apr-1.2.7-11
apr-util-devel-1.2.7-6
apr-devel-1.2.7-11
apr-util-1.2.7-6
[root@Server ~]# _
```

Now configure the ip address to 192.168.0.254 and check it

```
[root@Server ~]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:11:AD:E1
          inet addr:192.168.0.254  Bcast:192.168.0.255
          inet6 addr: fe80::20c:29ff:fe11:ade1/64 Scope
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Met
          RX packets:0 errors:0 dropped:0 overruns:0 fr
          TX packets:51 errors:0 dropped:0 overruns:0 c
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:8466 (8.2 KiB)
          Interrupt:67 Base address:0x2000
```

start httpd daemons and verify its running status

```
[root@Server ~]# chkconfig httpd on
[root@Server ~]# service httpd start
Starting httpd:
[root@Server ~]# service httpd status
httpd (pid 5465 5464 5463 5462 5461 5460 5459 5458 5456) is running
[root@Server ~]# pgrep httpd
5456
5458
5459
5460
5461
5462
5463
5464
5465
[root@Server ~]# _
```

Configure virtual hosting

In this example we will host a website www.vinita.com to apache web server. create a documents root directory for this website and a index page

```
[root@Server ~]# mkdir -p /var/www/virtual/www.vinita.com/html
[root@Server ~]# vi /var/www/virtual/www.vinita.com/html/index.html_
```

for testing purpose we are writing site name in its index page

```
<b> www.vinita.com </b>
```

save file and exit

now open /etc/hosts file

```
[root@Server ~]# vi /etc/hosts_
```

in the end of file bind system ip with www.vinita.com

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1       localhost.localdomain   localhost
::1     localhost6.localdomain6 localhost6
192.168.0.254   Server  Server
192.168.0.1     Client1 Client1
192.168.0.2     Client2 Client2
192.168.0.254   www.vinita.com
```

now open /etc/httpd/conf/httpd.conf main configuration file of apache server

```
[root@Server ~]# vi /etc/httpd/conf/httpd.conf _
```

locate virtual host tag

```
969 # Use name-based virtual hosting.
970 #
971 #NameVirtualHost *:80
```

remove # from the beginning and add the IP of host

```
# Use name-based virtual hosting.
#
NameVirtualHost 192.168.0.254_
#
```

Now go in the end of file and copy last seven line [virtual host tag] and paste them in the end of file. change these seven lines as shown in image

```
984 #<VirtualHost *:80>
985 #    ServerAdmin webmaster@dummy-host.example.com
986 #    DocumentRoot /www/docs/dummy-host.example.com
987 #    ServerName dummy-host.example.com
988 #    ErrorLog logs/dummy-host.example.com-error_log
989 #    CustomLog logs/dummy-host.example.com-access_log common
990 #</VirtualHost>
991
992  <VirtualHost 192.168.0.254>
993     ServerAdmin root@www.vinita.com
994     DocumentRoot /var/www/virtual/www.vinita.com/html
995     ServerName www.vinita.com
996     ErrorLog logs/dummy-www.vinita.com-error_log
997     CustomLog logs/dummy-www.vinita.com-access_log common
998 </VirtualHost>_
```

now save this file and exit from it

you have done necessary configuration now restart the httpd service and test this configuration run links command

```
[root@Server ~]# service httpd restart
Stopping httpd:                                            [  OK  ]
Starting httpd:                                            [  OK  ]
[root@Server ~]# links 192.168.0.254_
```

if links command retrieve your home page

```
                                                  http://192.168.0.254/
www.vinita.com
```

means you have successfully configured the virtual host now test it with site name

```
[root@Server ~]# links www.vinita.com_
```

In output of links command you should see the index page of site

```
                                                  http://www.vinita.com/
www.vinita.com
```

Configure multiple site with same ip address

At this point you have configured one site **www.vinita.com** with the ip address **192.168.0.254.** Now we will configure one more site **www.nikita.com** with **same** ip address

create a documents root directory for www.nikita.com website and a index page

```
[root@Server ~]# mkdir -p /var/www/virtual/www.nikita.com/html
[root@Server ~]# vi /var/www/virtual/www.nikita.com/html/index.html_
```

for testing purpose we are writing site name in its index page

```
<b>www.nikita.com</b>_
```

save file and exit

now open /etc/hosts file and bind system ip with www.nikita.com

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1       localhost.localdomain   localhost
::1     localhost6.localdomain6 localhost6
192.168.0.254   Server  Server
192.168.0.1     Client1 Client1
192.168.0.2     Client2 Client2
192.168.0.254   www.vinita.com
192.168.0.254   www.nikita.com
```

now open /etc/httpd/conf/httpd.conf main configuration file of apache server

```
[root@Server ~]# vi /etc/httpd/conf/httpd.conf _
```

Now go in the end of file and copy last seven line [virtual host tag] and paste them in the end of file. change these seven lines as shown in image

```
 <VirtualHost 192.168.0.254>
    ServerAdmin root@www.vinita.com
    DocumentRoot /var/www/virtual/www.vinita.com/html
    ServerName www.vinita.com
    ErrorLog logs/dummy-www.vinita.com-error_log
    CustomLog logs/dummy-www.vinita.com-access_log common
</VirtualHost>

 <VirtualHost 192.168.0.254>
    ServerAdmin root@www.nikita.com
    DocumentRoot /var/www/virtual/www.nikita.com/html
    ServerName www.nikita.com
    ErrorLog logs/dummy-www.nikita.com-error_log
    CustomLog logs/dummy-www.nikita.com-access_log common
</VirtualHost>_
```

now save this file and exit from it

you have done necessary configuration now restart the httpd service

```
[root@Server ~]# service httpd restart
Stopping httpd:                                            [  OK  ]
Starting httpd:                                            [  OK  ]
[root@Server ~]# _
```

test this configuration run links command

```
[root@Server ~]# links www.nikita.com_
```

In output of links command you should see the index page of site

configure multiple site with multiple ip address

Now we will host multiple sites with multiple ip address. Create a virtual lan card on server and assign its an ip address of **192.168.0.253.** [if you don't know how to create virtual lan card and read our pervious article how to create virtual lan card]. we will create a testing site **www.nidhi.com** and will bind it with ip address of **192.168.0.253**

create a documents root directory for www.nidhi.com website and a index page

```
[root@Server ~]# mkdir -p /var/www/virtual/www.nidhi.com/html
[root@Server ~]# vi /var/www/virtual/www.nidhi.com/html/index.html_
```

for testing purpose we are writing site name in its index page

```
<b>www.nidhi.com</b>_
```

save file and exit

now open /etc/hosts file and bind system ip with www.nidhi.com

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1       localhost.localdomain   localhost
::1     localhost6.localdomain6 localhost6
192.168.0.254   Server  Server
192.168.0.1     Client1 Client1
192.168.0.2     Client2 Client2
192.168.0.254   www.vinita.com
192.168.0.254   www.nikita.com
192.168.0.253   www.nidhi.com_
```

now open /etc/httpd/conf/httpd.conf main configuration file of apache server

```
[root@Server ~]# vi /etc/httpd/conf/httpd.conf _
```

Now go in the end of file and copy last seven line [virtual host tag] and paste them in the end of file. change these seven lines as shown in image

```
 <VirtualHost 192.168.0.254>
    ServerAdmin root@www.nikita.com
    DocumentRoot /var/www/virtual/www.nikita.com/html
    ServerName www.nikita.com
    ErrorLog logs/dummy-www.nikita.com-error_log
    CustomLog logs/dummy-www.nikita.com-access_log common
</VirtualHost>

 <VirtualHost 192.168.0.253>
    ServerAdmin root@www.nidhi.com
    DocumentRoot /var/www/virtual/www.nidhi.com/html
    ServerName www.nidhi.com
    ErrorLog logs/dummy-www.nidhi.com-error_log
    CustomLog logs/dummy-www.nidhi.com-access_log common
</VirtualHost>
_
```

now save this file and exit from it

you have done necessary configuration now restart the httpd service

```
[root@Server ~]# service httpd restart
Stopping httpd:                                            [  OK  ]
Starting httpd:                                            [  OK  ]
[root@Server ~]# _
```

test this configuration run links command

```
root@Server ~]# service httpd restart
topping httpd:                                             [  OK  ]
tarting httpd:                                             [  OK  ]
root@Server ~]# links www.nidhi.com_
```

In output of links command you should see the index page of site

How to create site alias

Now I will show you that how can you use **site alias** to configure more name of same site. we configure a site **www.vinita.com** in stating of example. now we will create **www.goswami.com** site alias for this site so this site can be access with both name.

To create alias first make its entry in /etc/hosts file as shown here

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1       localhost.localdomain   localhost
::1     localhost6.localdomain6 localhost6
192.168.0.254   Server  Server
192.168.0.1     Client1 Client1
192.168.0.2     Client2 Client2
192.168.0.254   www.vinita.com    www.goswami.com_
192.168.0.254   www.nikita.com
192.168.0.253   www.nidhi.com
```

Now open main apache configuration /etc/httpd/conf/httpd.conf

```
[root@Server ~]# vi /etc/httpd/conf/httpd.conf _
```

Now go in the end of file and copy last seven line [virtual host tag] and paste them in the end of file. change these seven lines as shown in image

```
 <VirtualHost 192.168.0.254>
    ServerAdmin root@www.vinita.com
    DocumentRoot /var/www/virtual/www.vinita.com/html
    ServerName www.vinita.com
    ServerAlias www.goswami.com_
    ErrorLog logs/dummy-www.vinita.com-error_log
    CustomLog logs/dummy-www.vinita.com-access_log common
</VirtualHost>
```

now save this file and exit from it

you have done necessary configuration now restart the httpd service and test this configuration run links command

```
[root@Server ~]# service httpd restart
Stopping httpd:                                            [  OK  ]
Starting httpd:                                            [  OK  ]
[root@Server ~]# links www.goswami.com_
```

In output of links command you should see the index page of site

```
                                                  http://www.goswami.com/
www.vinita.com
```

How to configure linux nfs server step by step guide Example and Implementation

NFS, or **Network File System,** is a server-client protocol for sharing files between computers on a common network. NFS enables you to mount a file system on a remote computer as if it were local to your own system. You can then directly access any of the files on that remote file system. The server and client do not have to use the same operating system. The client system just needs to be running an **NFS client** compatible with the **NFS server**.

For example **NFS server** could be a Linux system and Unix could be a client. But it can't be a window system because window is not NFS compatible. The NFS server exports one or more directories to the client systems, and the client systems mount one or more of the shared directories to local directories called mount points. After the share is mounted, all I/O operations are written back to the server, and all clients notice the change as if it occurred on the local filesystem.

A manual refresh is not needed because the client accesses the remote filesystem as if it were local.because access is granted by IP address, a username and password are not required. However, there are security risks to consider because the **NFS server** knows nothing about the users on the client system.

Exam question 1 Some users home directory is shared from your system. Using showmount -e localhost command, the shared directory is not shown. Make access the shared users home directory

Exam question 2 The System you are using is for NFS (Network File Services). Some important data are shared from your system. Make automatically start the nfs and portmap services at boot time

Exam question 3 Share /data directory using NFS only to 192.168.0.0/24 members. These hosts should get read and write access on shared directory.

Configure nfs server

In this example we will configure a nfs server and will mount shared directory from client side.

For this example we are using two systems one linux server one linux clients . To complete these per quest of nfs server Follow this link

- ***A linux server with ip address 192.168.0.254 and hostname Server***
- ***A linux client with ip address 192.168.0.1 and hostname Client1***
- ***Updated /etc/hosts file on both linux system***
- ***Running portmap and xinetd services***
- ***Firewall should be off on server***

We have configured all these steps in our pervious article.

We suggest you to review that article before start configuration of nfs server. Once you have completed the necessary steps follow this guide.

Three rpm are required to configure nfs server. nfs, portmap, xinetd check them if not found then install

```
[root@Server ~]# rpm -qa nfs*
nfs-utils-1.0.9-24.el5
nfs-utils-lib-1.0.8-7.2.z2
[root@Server ~]# rpm -qa portmap*
portmap-4.0-65.2.2.1
[root@Server ~]# rpm -qa xinetd*
xinetd-2.3.14-10.el5
[root@Server ~]# _
```

Now check nfs, portmap, xinetd service in system service it should be on

#setup
Select System service from list
[*]portmap
[*]xinetd
[*]nfs

Now restart xinetd and portmap service

```
[root@Server ~]# service portmap restart
Stopping portmap:                                          [  OK  ]
Starting portmap:                                          [  OK  ]
[root@Server ~]# service xinetd restart
Stopping xinetd:                                           [  OK  ]
Starting xinetd:                                           [  OK  ]
[root@Server ~]# _
```

To keep on these services after reboot on then via chkconfig command

```
[root@Server ~]# chkconfig portmap on
[root@Server ~]# chkconfig xinetd on
[root@Server ~]# _
```

After reboot verify their status. It must be in running condition

```
[root@Server ~]# service portmap status
portmap (pid 3430) is running...
[root@Server ~]# service xinetd status
xinetd (pid 3462) is running...
[root@Server ~]# _
```

now create a /data directory and grant full permission to it

```
[root@Server ~]# mkdir /data
[root@Server ~]# chmod 777 /data
[root@Server ~]# _
```

now open /etc/exports file

```
[root@Server ~]# vi /etc/exports _
```

share data folder for the network of 192.168.0.254/24 with read and write access

```
/data            192.168.0.0/24(rw,sync)_
```

save file with :wq and exit

now restart the nfs service and also on it with chkconfig

```
[root@Server ~]# service nfs restart
Shutting down NFS mountd:                                  [  OK  ]
Shutting down NFS daemon: nfsd: last server has exited
nfsd: unexporting all filesystems
                                                           [  OK  ]
Shutting down NFS quotas:                                  [  OK  ]
Shutting down NFS services:                                [  OK  ]
Starting NFS services:                                     [  OK  ]
Starting NFS quotas:                                       [  OK  ]
Starting NFS daemon: NFSD: Using /var/lib/nfs/v4recovery as the NFSv4
very directory
NFSD: starting 90-second grace period
                                                           [  OK  ]
Starting NFS mountd:                                       [  OK  ]
[root@Server ~]# chkconfig nfs on
[root@Server ~]# _
```

also restart nfs daemons with expotfs

```
[root@Server ~]# exportfs -r
[root@Server ~]# _
```

verify with showmount command that you have successfully shared data folder

```
[root@Server ~]# showmount -e
Export list for Server:
/data 192.168.0.0/24
[root@Server ~]# _
```

configure client system

ping form nfs server and check the share folder

```
[root@Client1 ~]# showmount -e 192.168.0.254
Export list for 192.168.0.254:
/data 192.168.0.0/24
[root@Client1 ~]# _
```

now mount this share folder on mnt mount point. To test this share folder change directory to mnt and create a test file

```
[root@Client1 ~]# mount -t nfs 192.168.0.254:/data /mnt
[root@Client1 ~]# cd /mnt
[root@Client1 mnt]# cat > test
this is test file created on client side
[root@Client1 mnt]# _
```

After use you should always unmount from mnt mount point

```
[root@Client1 mnt]# cd
[root@Client1 ~]# umount /mnt
[root@Client1 ~]# _
```

In this way you can use **shared folder**. But this share folder will be available till system is **up**. It will not be available after **reboot**. To keep it available after reboot make its entry in **fstab**

create a mount point, by making a directory

```
[root@Client1 ~]# mkdir /temp_
```

now open /etc/fstab file

```
[root@Client1 ~]# vi /etc/fstab _
```

make entry for nfs shared directory and define /temp to mount point

```
LABEL=/                 /                       ext3    defaults        1 1
LABEL=/home             /home                   ext3    defaults        1 2
LABEL=/boot             /boot                   ext3    defaults        1 2
tmpfs                   /dev/shm                tmpfs   defaults        0 0
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
sysfs                   /sys                    sysfs   defaults        0 0
proc                    /proc                   proc    defaults        0 0
LABEL=SWAP-sda3         swap                    swap    defaults        0 0
192.168.0.254:/data     /temp                   nfs     defaults        0 0
```

save the with :wq and exit reboot the system with reboot -f command

#reboot -f

after reboot check /temp directory it should show all the shared data

```
[root@Client1 ~]# cd /temp
[root@Client1 temp]# ls
test
[root@Client1 temp]# _
```

How to configure linux ftp server step by step guide Example and Implementation

ftp server is used to transfer files between server and clients. All major operating system supports ftp. ftp is the most used protocol over internet to transfer files. Like most Internet operations, FTP works on a client/ server model. FTP client programs can enable users to transfer files to and from a remote system running an FTP server program.

Any Linux system can operate as an FTP server. It has to run only the server software—an FTP daemon with the appropriate configuration. Transfers are made between user accounts on client and server systems. A user on the remote system has to log in to an account on a server and can then transfer files to and from that account's directories only.

A special kind of user account, named **ftp**, allows any user to log in to it with the username **"anonymous."** This account has its own set of directories and files that are considered public, available to anyone on the network who wants to download them.

The numerous FTP sites on the Internet are FTP servers supporting FTP user accounts with anonymous login. Any Linux system can be configured to support anonymous FTP access, turning them into network FTP sites. Such sites can work on an intranet or on the Internet.

Configuring the ftp Server

The vsftpd RPM package is required to configure a Red Hat Enterprise Linux system as an ftp server. If it is not already installed, install it with rpm commands as described in our pervious article. After it is installed, start the service as root with the command service vsftpd start . The system is now an ftp server and can accept connections. To configure the server to automatically start the service at boot time, execute the command chkconfig vsftpd on as root. To stop the server, execute the command service vsftpd stop. To verify that the server is running, use the command service vsftpd status.

Configure vsftpd server

In this example we will configure a **vsftpd** server and will transfer files from client side.

For this example we are using three systems one linux server one linux clients and one windows xp clients.

- ***A linux server with ip address 192.168.0.254 and hostname Server***
- ***A linux client with ip address 192.168.0.1 and hostname Client1***
- ***A window client with ip address 192.168.0.2 and hostname Client2***

- ***Updated /etc/hosts file on both linux system***
- ***Running portmap and xinetd services***
- ***Firewall should be off on server***

We suggest you to review that article before start configuration of ssh server. Once you have completed the necessary steps follow this guide.

Three rpm are required to configure ssh server. vsftpd, portmap, xinetd check them if not found then install

```
[root@Server ~]# rpm -qa vsftpd
vsftpd-2.0.5-10.el5
[root@Server ~]# rpm -qa portmap
portmap-4.0-65.2.2.1
[root@Server ~]# rpm -qa xinetd
xinetd-2.3.14-10.el5
[root@Server ~]# _
```

Now check vsftpd, portmap, xinetd service in system service it should be on

#setup
Select System service from list
[*]portmap
[*]xinetd
[*]vsftpd

Now restart xinetd and portmap and vsftpd service

```
[root@Server ~]# service portmap restart
Stopping portmap:                                          [  OK  ]
Starting portmap:                                          [  OK  ]
[root@Server ~]# service xinetd restart
Stopping xinetd:                                           [  OK  ]
Starting xinetd:                                           [  OK  ]
[root@Server ~]# _

[root@Server ~]# service vsftpd restart
Shutting down vsftpd:                                      [  OK  ]
Starting vsftpd for vsftpd:                                [  OK  ]
[root@Server ~]# chkconfig vsftpd on
[root@Server ~]# _
```

To keep on these services after reboot on then via chkconfig command

```
[root@Server ~]# chkconfig portmap on
[root@Server ~]# chkconfig xinetd on
[root@Server ~]# _
```

After reboot verify their status. It must be in running condition

```
[root@Server ~]# service portmap status
portmap (pid 3430) is running...
[root@Server ~]# service xinetd status
xinetd (pid 3462) is running...
[root@Server ~]# _
```

Create a normal user named vinita

```
[root@Server backup]# useradd vinita
[root@Server backup]# passwd vinita
Changing password for user vinita.
New UNIX password:
BAD PASSWORD: it is WAY too short
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[root@Server backup]#
```

Login for this user on other terminal and create a test file

```
[vinita@Server ~]$ cat > test
This is test file created on Linux ftp server
[vinita@Server ~]$ _
```

On Linux client

ping from ftp server and run ftp command and give username and password

```
[root@Client1 ~]# ftp 192.168.0.254
Connected to 192.168.0.254.
220 (vsFTPd 2.0.5)
530 Please login with USER and PASS.
530 Please login with USER and PASS.
KERBEROS_V4 rejected as an authentication type
Name (192.168.0.254:root): vinita
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> get test
local: test remote: test
227 Entering Passive Mode (192,168,0,254,240,40)
150 Opening BINARY mode data connection for test (46 bytes).
226 File send OK.
46 bytes received in 0.0026 seconds (17 Kbytes/s)
ftp> quit
221 Goodbye.
[root@Client1 ~]# ls
anaconda-ks.cfg  Desktop  install.log  install.log.syslog  test
[root@Client1 ~]# cat test
This is test file created on Linux ftp server
[root@Client1 ~]# _
```

after login you can download files from the specified directories

Most commonly commands used on ftp prompt are

put To upload files on server
get To download files from server
mput To upload all files
mget To download all files
? To see all available command on ftp prompts
cd To change remote directory

lcd To change local directory

```
ftp> ?
Commands may be abbreviated.  Commands are:

!               delete          literal         prompt
?               debug           ls              put
append          dir             mdelete         pwd
ascii           disconnect      mdir            quit
bell            get             mget            quote
binary          glob            mkdir           recv
bye             hash            mls             remotehelp
cd              help            mput            rename
close           lcd             open            rmdir
ftp> _
```

On window clients

Now go on window clients and create a file. copy con command is used to create files on window. To save use CTRL+Z

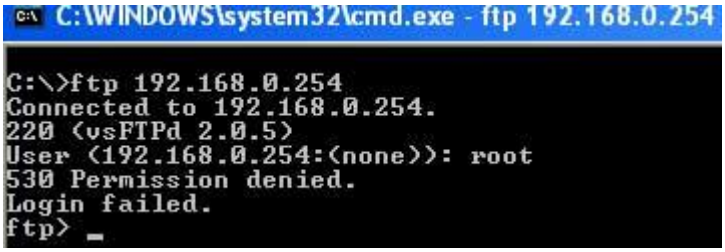
```
C:\WINDOWS\system32\cmd.exe

C:\>copy con win
this is test file created on window
^Z
        1 file(s) copied.

C:\>
```

Now ping from ftp server and invoke ftp session from server, login from user account and download as well as uploads files

```
C:\WINDOWS\system32\cmd.exe

C:\>ftp 192.168.0.254
Connected to 192.168.0.254.
220 (vsFTPd 2.0.5)
User (192.168.0.254:(none)): vinita
331 Please specify the password.
Password:
230 Login successful.
ftp> get test
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for test (46 bytes).
226 File send OK.
ftp: 46 bytes received in 0.00Seconds 46000.00Kbytes/sec.
ftp> put win
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 File receive OK.
ftp: 37 bytes sent in 0.00Seconds 37000.00Kbytes/sec.
ftp> quit
221 Goodbye.

C:\>type test
This is test file created on Linux ftp server

C:\>_
```

Enable root account for ftp session and set permission on user

By default on vsftpd server root account is disable. You cannot login from root account.



Now we will enable root account for ftp session and same time we will disable our normal user vinita to use ftp sessions.

open file /etc/vsftpd/ftpusers . Users whose name are set in this file will not allowed to login from ftp.

```
[root@Server ~]# vi /etc/vsftpd/ftpusers _

# Users that are not allowed to login via ftp
root
bin
daemon
adm
lp
sync
shutdown
```

By default this file have an entry for root that why root are not allowed to use ftp. remove root from list and add user vinita

```
# Users that are not allowed to login via ftp
vinita_
bin
daemon
adm
lp
sync
shutdown
halt
mail
```

Now remove entry form /etc/vsftpd/user_list files. Users whose names are set in this file are also not allowed to login from ftp even they are not prompt for password.

```
[root@Server ~]# vi /etc/vsftpd/user_list _

# vsftpd userlist
# If userlist_deny=NO, only allow users in this file
# If userlist_deny=YES (default), never allow users in this file, and
# do not even prompt for a password.
# Note that the default vsftpd pam config also checks /etc/vsftpd/ftpusers
# for users that are denied.
root
bin
daemon
```

By default this file have an entry for root that way root is denied form login even not

asked for password remove root from list and add user vinita

```
# vsftpd userlist
# If userlist_deny=NO, only allow users in this file
# If userlist_deny=YES (default), never allow users in this file, and
# do not even prompt for a password.
# Note that the default vsftpd pam config also checks /etc/vsftpd/ftpusers
# for users that are denied.
vinita_
bin
daemon
adm
```

After saving change in these files restart the vsftpd service

```
[root@Server ~]# service vsftpd restart
Shutting down vsftpd:                                      [  OK  ]
Starting vsftpd for vsftpd:                                [  OK  ]
[root@Server ~]# chkconfig vsftpd on
[root@Server ~]# _
```

Now go on client system and login from root this time root will login

```
C:\WINDOWS\system32\cmd.exe - ftp 192.168.0.254

C:\>ftp 192.168.0.254
Connected to 192.168.0.254.
220 (vsFTPd 2.0.5)
User (192.168.0.254:(none)): root
331 Please specify the password.
Password:
230 Login successful.
ftp> _
```

Now try to login form user vinita she should not prompt form password also

```
C:\WINDOWS\system32\cmd.exe - ftp 192.168.0.254

C:\>ftp 192.168.0.254
Connected to 192.168.0.254.
220 (vsFTPd 2.0.5)
User (192.168.0.254:(none)): vinita
530 Permission denied.
Login failed.
ftp> _
```

How to set login banner for ftp server

To set login banner open /etc/vsftpd/vsftpd.conf file and search for this tag

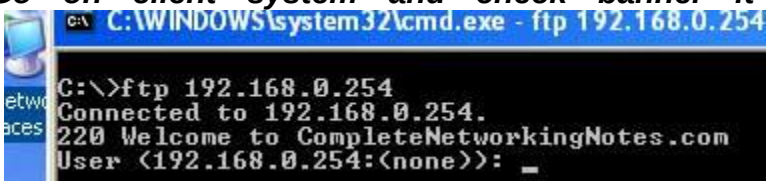
```
# raw file.
# ASCII mangling is a horrible feature of the protocol.
#ascii_upload_enable=YES
#ascii_download_enable=YES
#
# You may fully customise the login banner string:
#ftpd_banner=Welcome to blah FTP service.
```

Uncomment this tag and set your banner and save file , and restart the vsftpd

service

```
# raw file.
# ASCII mangling is a horrible feature of the protocol.
#ascii_upload_enable=YES
#ascii_download_enable=YES
#
# You may fully customise the login banner string:
ftpd_banner=Welcome to CompleteNetworkingNotes.com_
```

Go on client system and check banner it will appear before user login



How to configure linux dhcp server step by step guide Example and Implementation

DHCP, or **Dynamic Host Configuration Protocol,** allows an administrator to configure network settings for all clients on a central server.

The DHCP clients request an IP address and other network settings from the **DHCP server** on the network. The **DHCP server** in turn leases the client an IP address within a given range or leases the client an IP address based on the MAC address of the client's network interface card (NIC). The information includes its IP address, along with the network's name server, gateway, and proxy addresses,including the netmask.

Nothing has to be configured manually on the local system, except to specify the **DHCP server** it should get its network configuration from. If an IP address is assigned according to the MAC address of the client's NIC, the same IP address can be leased to the client every time the client requests one. DHCP makes network administration easier and less prone to error.

Exam Question Configure the DHCP server by matching the following conditions:

- *Subnet and netmask should be 192.168.0.0 255.255.255.0*
- *Gateway Should be 192.168.0.254*
- *DNS Sever Should be 192.168.0.254*
- *Domain Name should be example.com*
- *Range from 192.168.0.10-50*

Exam Question You have DHCP server, which assigns the IP, gateway and DNS server ip to Clients. There is one DNS servers having MAC address (00:50:FC:98:8D:00 in your LAN, But it always required fixed IP address (192.168.0.10). Configure the DHCP server to assign the fixed IP address to DNS server.

Configure dhcp server

In this example we will configure a **dhcp server** and will lease ip address to clients.

For this example we are using three systems one linux server one linux clients and one window clients.

dhcp rpm is required to configure dhcp server. check it if not found then install

```
[root@Server ~]# rpm -qa dhcp
dhcp-3.0.5-7.el5
[root@Server ~]# _
```

Now check dhcpd service in system service it should be on

#setup
Select System service from list
[*]dhcpd

To assign IP to dhcp server

DHCP server have a static a ip address. First configure the ip address **192.168.0.254** with netmask of **255.255.255.0** on server.

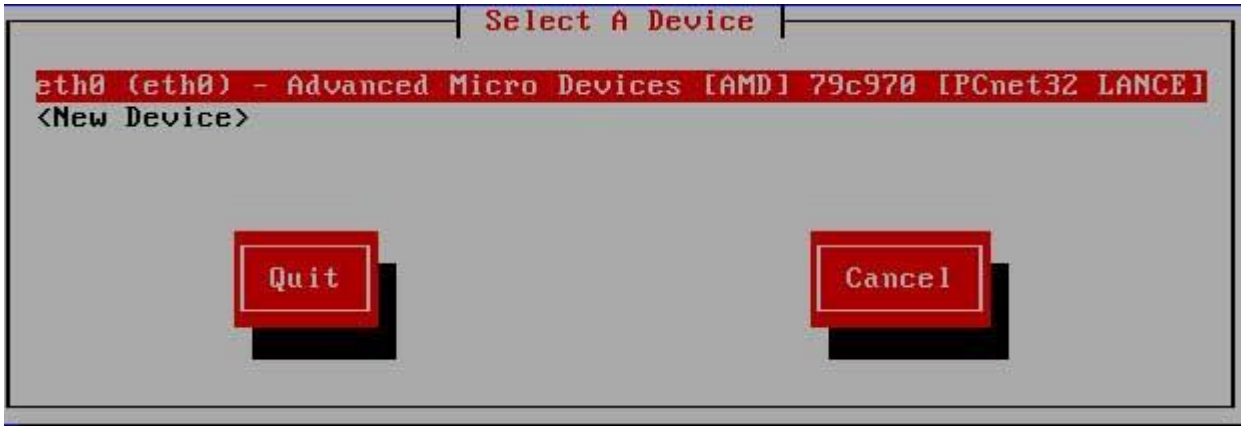
Run setup command form root user

#setup

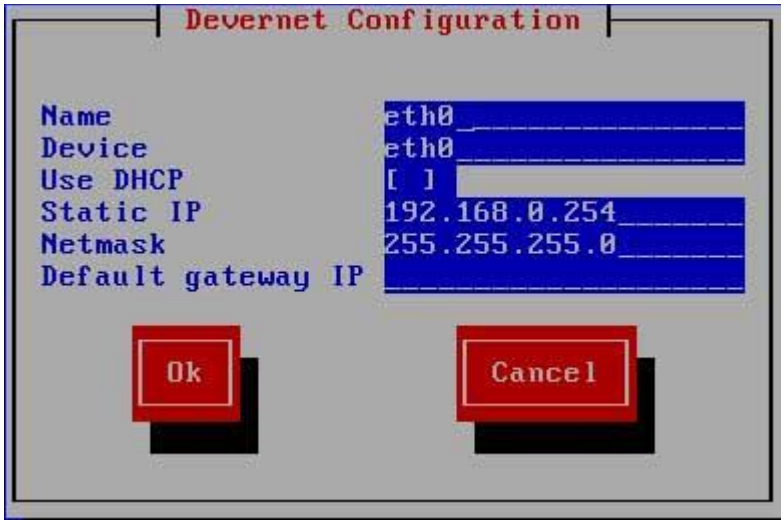
```
[root@localhost Server]# setup_
```

this will launch a new window select network configuration

now a new window will show you all available LAN card select your LAN card (if you don't see any LAN card here mean you don't have install driver)



assign IP in this box and click ok



click on ok, quit and again quit to come back on root prompt.

restart the network service so new ip address can take place on LAN card

#service network restart

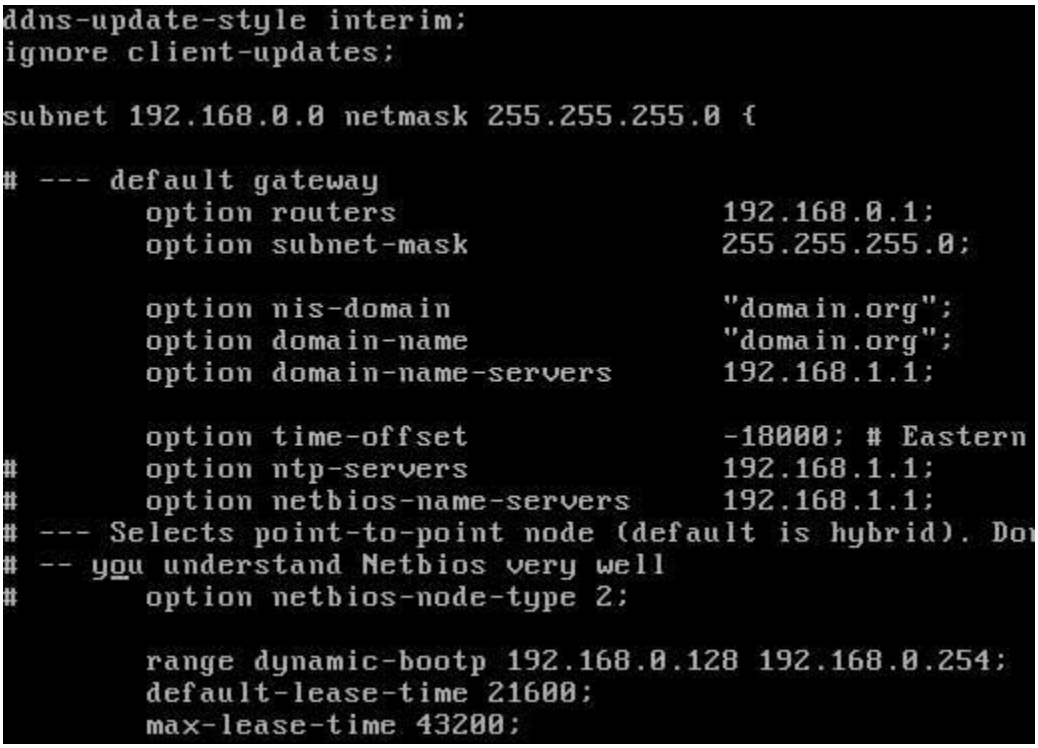
main configuration file of dhcp server is dhcpd.conf. This file located on /etc directory. If this file is not present there or you have corrupted this file, then copy new file first, if ask for overwrite press y

```
[root@Server ~]# cp /usr/share/doc/dhcp-3.0.5/dhcpd.conf.sample /etc/dhcpd.conf
cp: overwrite `/etc/dhcpd.conf'? y
[root@Server ~]# _
```

now open /etc/dhcpd.conf

```
[root@Server ~]# vi /etc/dhcpd.conf _
```

default entry in this file look like this



make these change in this file to configure dhcp server

remove this line
 # - - - default gateway
set option routers to
192.168.0.254
set option subnet-mask to
255.255.255.0
option nis domain to
example.com
option domain-name to
example.com
option domain-name-servers to
192.168.0.254
range dynamic-bootp to
192.168.0.10 192.168.0.50;

After change this file should look like this

```
ddns-update-style interim;
ignore client-updates;

subnet 192.168.0.0 netmask 255.255.255.0 {

        option routers                  192.168.0.254;
        option subnet-mask              255.255.255.0;

        option nis-domain               "example.com";
        option domain-name              "example.com";
        option domain-name-servers      192.168.0.254;

        option time-offset              -18000; # Eastern
#       option ntp-servers              192.168.1.1;
#       option netbios-name-servers     192.168.1.1;
# --- Selects point-to-point node (default is hybrid). Do
# -- you understand Netbios very well
#       option netbios-node-type 2;

        range dynamic-bootp 192.168.0.10 192.168.0.50;
        default-lease-time 21600;
        max-lease-time 43200;
```

how to assign fix ip address to any host

locate this paragraph and change hardware Ethernet to client's mac address and fixed -address to ip address which you want to provide that host

```
# we want the nameserver to appear at a fixed address
host ns {
        next-server marvin.redhat.com;
        hardware ethernet 12:34:56:78:AB:CD;
        fixed-address 207.175.42.254;
}
```

after making necessary change save file and exit

now create a blank file use to store the allocated ip address information

```
[root@Server ~]# touch /var/lib/dhcpd/dhcpd.leases
[root@Server ~]# _
```

Now restart dhcpd service and on it with chkconfig commands

```
[root@Server ~]# service dhcpd restart
Shutting down dhcpd:                                       [FAILED]
Starting dhcpd:                                            [  OK  ]
[root@Server ~]# chkconfig dhcpd on
[root@Server ~]# _
```

Linux Client configuration

Client configuration is very easy and straightforward. All you need to do is set ip address to dynamic in the properties of lan card. In linux

#setup

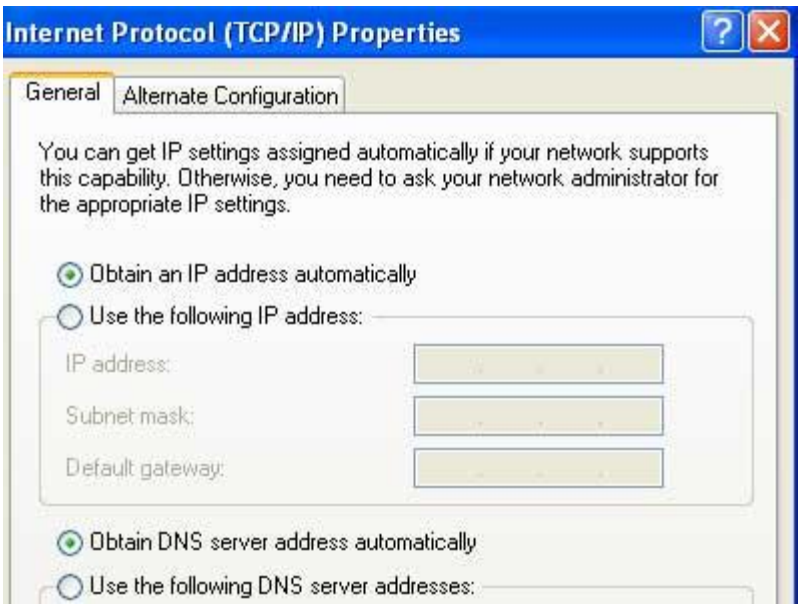
select network configuration from menu list
Select lan card and enter on ok
Select USE DHCP and enter on ok
Now click on quit and quit to come back on root prompt

Now restart the network service to obtain ip from dhcp server

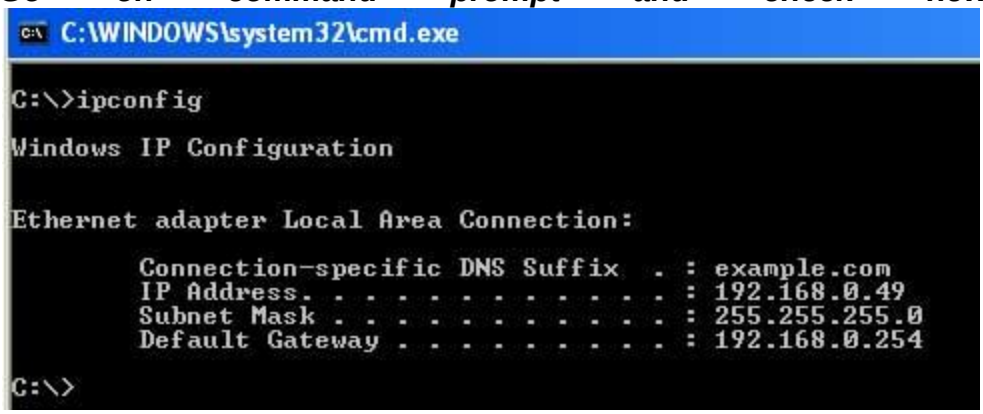
```
[root@Client1 temp]# service network restart
Shutting down interface eth0:                              [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface eth0:
Determining IP information for eth0... done.
                                                           [  OK  ]
[root@Client1 temp]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:62:28:1A
          inet addr:192.168.0.50  Bcast:192.168.0.255  Mask:255.255.
          inet6 addr: fe80::20c:29ff:fe62:281a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:42 errors:0 dropped:0 overruns:0 frame:0
          TX packets:134 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5962 (5.8 KiB)  TX bytes:23484 (22.9 KiB)
          Interrupt:67 Base address:0x2000

[root@Client1 temp]# _
```

Window Client configuration

To configure windows system as dhcp clients open lan card properties and select tcp/ip and click on properties and set obtain ip address automatically



Go on command prompt and check new ip address



Check lease on DHCP server

you can check allocated address on server.

```
[root@Server ~]# cat /var/lib/dhcpd/dhcpd.leases_

lease 192.168.0.50 {
  starts 3 2010/02/17 12:13:27;
  ends 3 2010/02/17 18:13:27;
  binding state active;
  next binding state free;
  hardware ethernet 00:0c:29:62:28:1a;
}
lease 192.168.0.49 {
  starts 3 2010/02/17 12:14:38;
  ends 3 2010/02/17 18:14:38;
  binding state active;
  next binding state free;
  hardware ethernet 00:0c:29:69:d8:2f;
  uid "\001\000\014)i\330/";
  client-hostname "nikki-82617912b";
}
[root@Server ~]# _
```

How to configure linux samba server step by step guide Example and Implementation

Most Linux systems are the part of networks that also run Windows systems. Using Linux **Samba servers**, your Linux and Windows systems can share directories and printers. This is most use full situation where your clients are window native and you want to use the linux security features.

Exam question There are mixed lots of System running on Linux and Windows OS. Some users are working on Windows Operating System. There is a /data directory on linux server should make available on windows to only vinita should have right to connect with samba server . Configure to make available.

Configure samba server

In this example we will configure a **samba** server and will transfer files from client side.For this example we are using two systems one linux server one window clients.

per quest of samba server

- ***A linux server with ip address 192.168.0.254 and hostname Server***
- ***A window client with ip address 192.168.0.2 and hostname Client2***
- ***Updated /etc/hosts file on linux system***
- ***Running portmap and xinetd services***
- ***Firewall should be off on server***

We have configured all these steps in our pervious article.

necessary configuration for samba server

We suggest you to review that article before start configuration of samba server. Once you have completed the necessary steps follow this guide.

samba rpm is required to configure samba server. check them if not found then install

```
[root@Server ~]# rpm -qa samba*
samba-3.0.25b-0.el5.4
samba-common-3.0.25b-0.el5.4
samba-client-3.0.25b-0.el5.4
[root@Server ~]# _
```

Now check smb, portmap, xinetd service in system service it should be on

#setup
Select System service from list
[*]portmap
[*]xinetd
[*]smb

Now restart xinetd and portmap and smb service

```
[root@Server ~]# service portmap restart
Stopping portmap:                                          [  OK  ]
Starting portmap:                                          [  OK  ]
[root@Server ~]# service xinetd restart
Stopping xinetd:                                           [  OK  ]
Starting xinetd:                                           [  OK  ]
[root@Server ~]# _
```

To keep on these services after reboot on then via chkconfig command

```
[root@Server ~]# chkconfig portmap on
[root@Server ~]# chkconfig xinetd on
[root@Server ~]# _
```

After reboot verify their status. It must be in running condition

```
[root@Server ~]# service portmap status
portmap (pid 3430) is running...
[root@Server ~]# service xinetd status
xinetd (pid 3462) is running...
[root@Server ~]# _
```

Create a normal user named vinita

```
[root@Server backup]# useradd vinita
[root@Server backup]# passwd vinita
Changing password for user vinita.
New UNIX password:
BAD PASSWORD: it is WAY too short
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[root@Server backup]#
```

now create /data directory and grant it full permission

```
[root@Server ~]# mkdir /data
[root@Server ~]# chmod 777 /data
[root@Server ~]# _
```

open /etc/samba/smb.conf main samba configuration files

```
[root@Server ~]# vi /etc/samba/smb.conf _
```

By default name of workgroup is MYGROUP in smb.conf file. you can change it with desire name

```
Hosts Allow/Hosts Deny lets you restrict who can
specifiy it as a per share option as well

        workgroup = MYGROUP
        server string = Samba Server Version %v
```

our task is to share data folder for vinita user so go in the end of file and do editing as shown here in this image

```
# Add this line to share
    [data]
 comment = personal share
 path = /data
 public = no
 writable = yes
 printable = no
 browseable = yes
 write list = vinita
```

save file with :wq and exit

Now add vinita user to samba user

```
[root@Server ~]# smbpasswd -a vinita
New SMB password:
Retype new SMB password:
[root@Server ~]# _
```

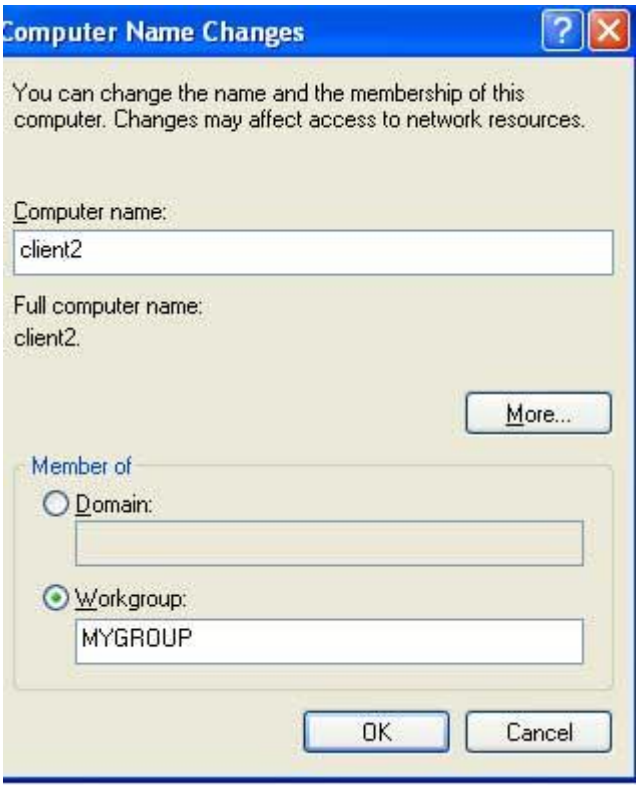
we have made necessary change now on smb service and check it status

```
[root@Server ~]# chkconfig smb on
[root@Server ~]# service smb start
Starting SMB services:
Starting NMB services:
[root@Server ~]# service smb status
smbd (pid 4332 4327) is running...
nmbd (pid 4330) is running...
[root@Server ~]# _
```

if you already have on this service then restart it with service smb restart commands.

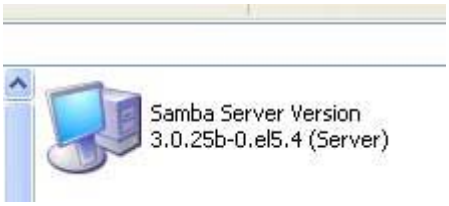
Client configuration for samba server

Go on windows system and ping samba server, change computer name to client2 and workgroup name to MYGROUP



reboot system after changing workgroup name

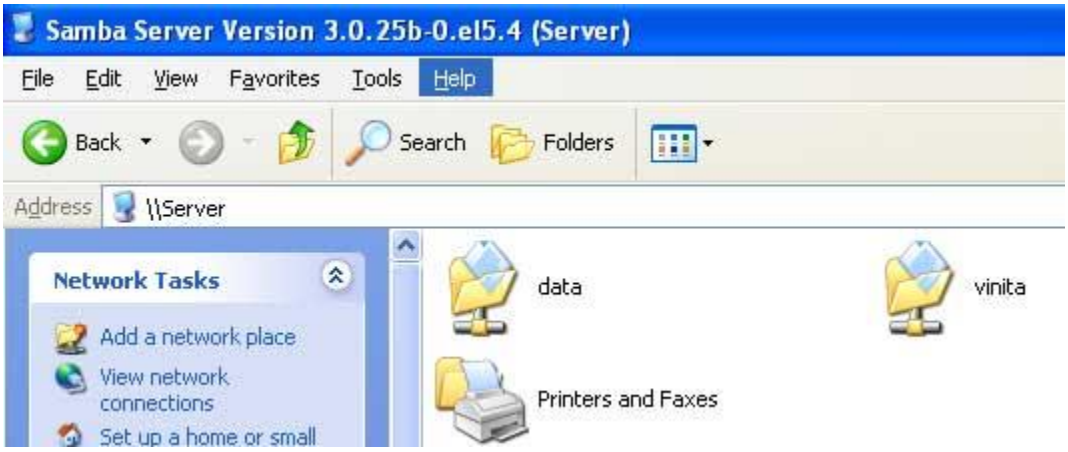
After reboot open my network place here you can see samba server [if not see then click on view workgroup computer in right pane, if still not see then use search button from tool bar and search computer samba server form ip]



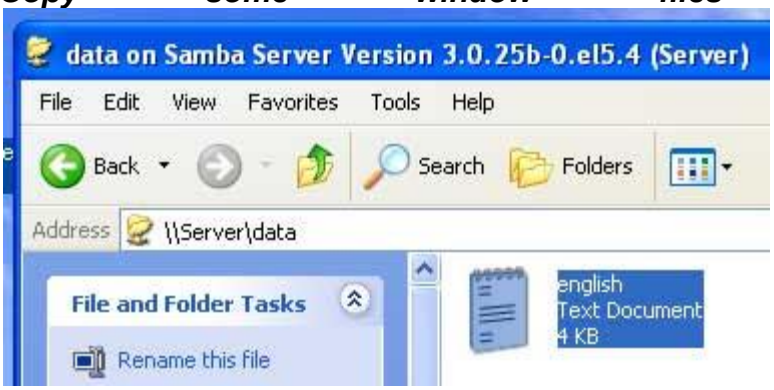
First try to login from user nikita she will not successes as nikita have not permission to login

Now login from user vinita [give the password which you set with smbpasswd command]

As you can see in image user vinita gets the /data folder which we share from samba server

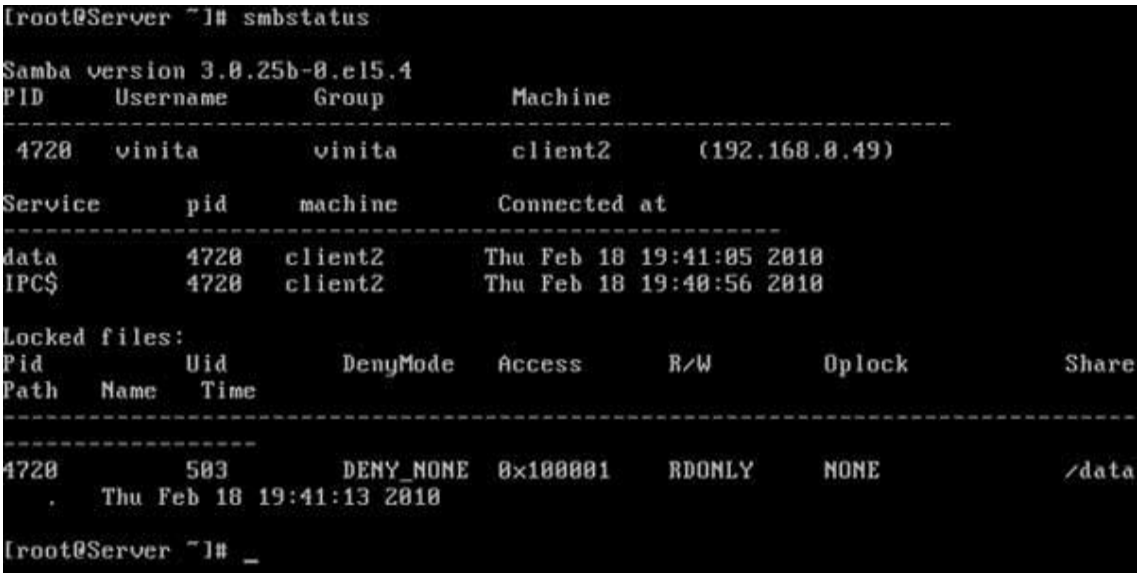


Copy some window files in data folder



Check status on samba server

on samba server you can check runtime status of samba server to check it run smbstatus command



in output you see that one samba shared directory is used on window system

Configure linux nis server step by step guide example and implementation

NIS, or **Network Information Systems,** is a network service that allows authentication and login information to be stored on a centrally located server. This includes the username and password database for login authentication, database of user groups, and the locations of home directories.

RHCE exam questions

One NIS Domain named rhce is configured in your lab, server is 192.168.0.254. nis1, nis2,nis3 user are created on domain server. Make your system as a member of rhce domain. Make sure that when nis user login in your system home directory should get by them. Home directory is shared on server /rhome/nis1.

RHCE exam doesn't ask candidate to configure NIS server. It test only NIS client side configuration. As you can see in example questions. But here in this article we will configure both server and client side for testing purpose so you can get more depth knowledge of nis server

Configure NIS server

In this example we will configure a NIS server and a user nis1 will login from client side.

For this example we are using two systems one linux server one linux clients . To complete these per quest of ssh server Follow this link

per quest of nis server

- ***A linux server with ip address 192.168.0.254 and hostname Server***
- ***A linux client with ip address 192.168.0.1 and hostname Client1***
- ***Updated /etc/hosts file on both linux system***
- ***Running portmap and xinetd services***
- ***Firewall should be off on server***

We have configured all these steps in our pervious article.

We suggest you to review that article before start configuration of nis server. Once you have completed the necessary steps follow this guide.

> ***Seven rpm are required to configure nis server. ypserv, cach, nfs, make, ypbind, portmap, xinetd check them if not found then install***

```
[root@Server ~]# rpm -qa ypserv*
ypserv-2.19-3
[root@Server ~]# rpm -qa ypbind*
ypbind-1.19-8.el5
[root@Server ~]# rpm -qa nfs*
nfs-utils-1.0.9-24.el5
nfs-utils-lib-1.0.8-7.2.z2
[root@Server ~]# rpm -qa make*
make-3.81-1.1
[root@Server ~]# rpm -qa cach*
cachefilesd-0.8-2.el5
caching-nameserver-9.3.3-10.el5
[root@Server ~]# rpm -qa portmap*
portmap-4.0-65.2.2.1
[root@Server ~]# rpm -qa xinetd*
xinetd-2.3.14-10.el5
[root@Server ~]# _
```

Now check nfs,ypserv,yppasswdd,ypbind, portmap, xinetd service in system service it should be on

#setup
Select System service from list
[*]portmap
[*]xinetd
[*]nfs
[*]ypserv
[*]yppasswdd
[*]ypbind

Now open /etc/sysconfig/network file

```
[root@Server ~]# vi /etc/sysconfig/network_
```

Set hostname and NIS domain name as shown here and save file

```
NETWORKING=yes
NETWORKING_IPV6=no
HOSTNAME=Server
NISDOMAIN=rhce
_
```

Now create a user named nis1 and give his home directory on /rhome with full permission

```
[root@Server ~]# mkdir /rhome
[root@Server ~]# useradd -d /rhome/nis1 nis1
[root@Server ~]# passwd nis1
Changing password for user nis1.
New UNIX password:
BAD PASSWORD: it is WAY too short
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[root@Server ~]# chmod 777 /rhome
[root@Server ~]# _
```

Now open /etc/exports file

```
[root@Server ~]# vi /etc/exports _
```

share /rhome/nis1 directory for network

```
/rhome/nis1                *(rw,sync)_
```

save this with :wq and exit

now open /var/yp/Makefile file

```
[root@Server ~]# vi /var/yp/Makefile _
```

and locate line number 109 [use ESC + : +set nu command to show hidden lines or read our vi editor article to know more about vi command line option]

```
104         $(MAKE) -f ../Makefile all
105
106 # If you don't want some of these maps built, feel free to
107 # them out from this list.
108
109 all:  passwd group hosts rpc services netid protocols mail
```

Now remove other entry from this line excepts passwd group hosts netid \ [as shown here]

```
103         $(NOPUSH) || $(MAKE) -f ../Makefile y
104         $(MAKE) -f ../Makefile all
105
106 # If you don't want some of these maps built,
107 # them out from this list.
108
109 all:  passwd group hosts  netid \
```

save this with :wq and exit

Now restart these service

#service portmap restart
#service xinetd restart
#service nfs restart
#service ypserv restart
#service yppasswdd restart

Don't restart **ypbind** service at this time as we haven't updated our database

Now change directory to /var/yp and run make command to create database

```
[root@Server ~]# cd /var/yp
[root@Server yp]# make
gmake[1]: Entering directory `/var/yp/rhce'
Updating netid.byname...
gmake[1]: Leaving directory `/var/yp/rhce'
[root@Server yp]# _
```

now update this database by running this commands [first add server and then add all client machine one by one. After adding press CTRL+D to save, confirm by

pressing y]

```
[root@Server ~]# /usr/lib/yp/ypinit -m

At this point, we have to construct a list of the
servers.  localhost.localdomain is in the list of
inue to add
the names for the other hosts, one per line.  When
list, type a <control D>.
        next host to add:  localhost.localdomain
        next host to add:  server
        next host to add:  client1
        next host to add:
The current list of NIS servers looks like this:

localhost.localdomain
server
client1

Is this correct?  [y/n: y]  y_
```

Now once again restart all these service this time there should be no error

#service portmap restart
#service xinetd restart
#service nfs restart
#service ypserv restart
#service yppasswdd restart
#service ypbind restart

Now set all these service to on with chkconfig so these could be on after restart

#chkconfig portmap on
#chkconfig xinetd on
#chkconfig nfs on
#chkconfig ypserv on
#chkconfig yppasswdd on
#chkconfig ypbind on

Client configuration

Before you start client configuration we suggest you to check proper connectivity between server and client. First try to login on NIS server from telnet. If you can successfully login via telnet then try to mount /rhome/nis1 directory via nfs server. If you get any error in telnet or nfs then remove those error first. You can read our pervious article for configuration related help.

To know how configure telnet server read How to configure linux telnet server step by step guide

To know how configure nfs server read How to configure linux nfs server step by step guide

Once you successfully completed necessary test then start configuration of client sides.

Two rpm are required to configure clients yp-tools and ypbind check them for install

```
[root@Client1 ~]# rpm -qa yp-tools
yp-tools-2.9-0.1
[root@Client1 ~]# rpm -qa ypbind
ypbind-1.19-8.el5
[root@Client1 ~]# _
```

now open /etc/sysconfig/network file

```
[root@Client1 ~]# vi /etc/sysconfig/network_
```

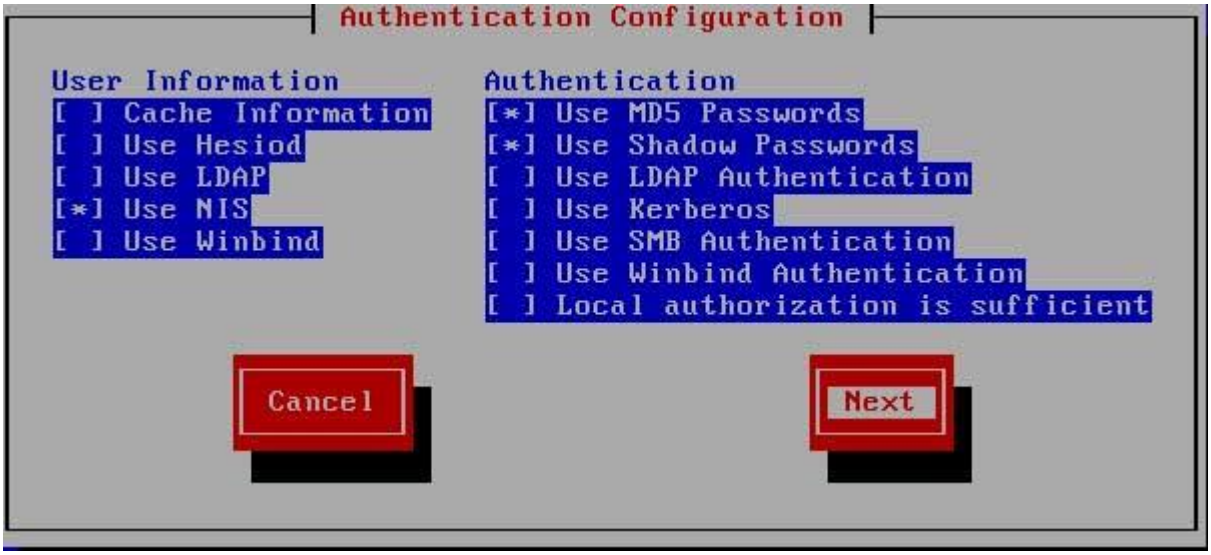
and make change as shown here

```
NETWORKING=yes
NETWORKING_IPV6=no
HOSTNAME=Client1
NISDOMAIN=rhce_
```

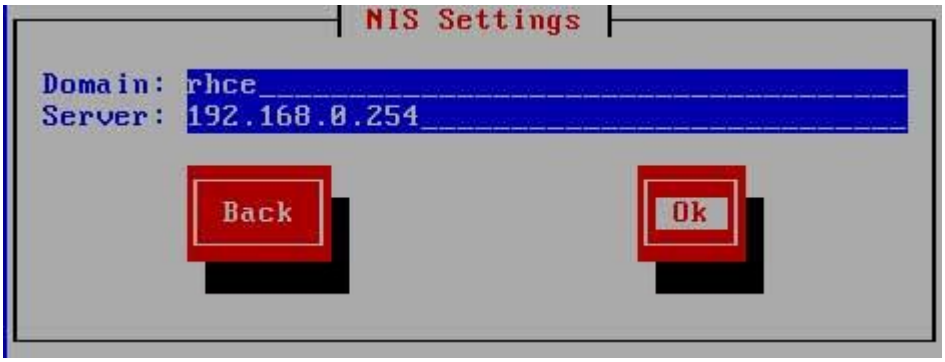
save the file with :wq and exit

now run setup command and select authentication configuration from list #setup

now check mark on NIS and enter on next



Set domain name to rhce and server to 192.168.0.254 and click on ok



No error should be occurred here if you see any error then check all configuration.

no open /etc/auto.master file

```
[root@Client1 ~]# vi /etc/auto.master _
```

in the end of file do editing of /rhome as shown here

```
#
# Note that if there are entries for /net or /misc (as
# above) in the included master map any keys that are the
# same will not be seen as the first read key seen takes
# precedence.
#
+auto.master
/rhome          /etc/auto.misc_
```

save the file with :wq and exit

now open /etc/auto.misc file

```
[root@Client1 ~]# vi /etc/auto.misc _
```

in the end of file do editing of user nis1 as shown here

```
#e2floppy       -fstype=ext2            :/dev/fd0
#jaz            -fstype=ext2            :/dev/sdc1
#removable      -fstype=ext2            :/dev/hdd
nis1            -rw,soft,intr           192.168.0.254:/rhome/nis1_
```

save the file with :wq and exit

now restart autofs and ypbind service

```
[root@Client1 ~]# service autofs restart
Stopping automount:                                        [  OK  ]
Starting automount:                                        [  OK  ]
[root@Client1 ~]# service ypbind restart
Shutting down NIS services:                                [  OK  ]
Binding to the NIS domain:                                 [  OK  ]
Listening for an NIS domain server..
[root@Client1 ~]# _
```

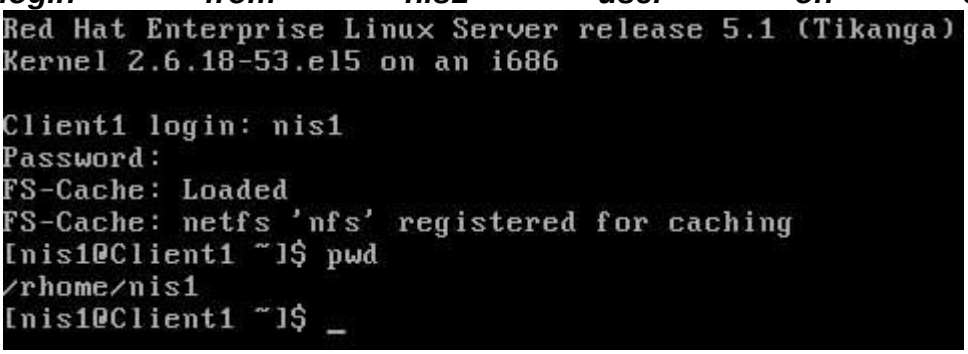
set these service on via chkconfig commands

#chkconfig autofs on
#chkconfig ypbind on

now restart the system

#reboot -f

login from nis1 user on client system



Linux service managements tools chkconfig ntsysv.

Services can be controlled by three programs in linux.

- ***chkconfig (command line only)***
- ***ntsysv (simple text-based application that doesn't require a graphical desktop)***
- ***or the Service Configuration Tool (graphical application).***

It's generally fastest way to control services at the command line. The **chkconfig** command gives you a simple way to maintain different runlevels within the **/etc/rc.d** directory structure. With chkconfig, you can add, remove, and change services; list startup information; and check the state of a particular service.

The **chkconfig** command can be used to configure runlevels and list the current runlevel configuration. It must be run as root if modifying a runlevel. Otherwise commands such as listing whether a service is started at boot time can be run as a non-root user.

Option	Description
--level *runlevel*	Specifies a runlevel to turn on, turn off, or reset a service.
--list *service*	Lists startup information for services at different runlevels. services are just on or off. With no argument, all services are listed.
--add *service*	Adds a service, creating links in the default specified runlevels (or all runlevels, if none are specified).
--del *service*	Deletes all links for the service (startup and shutdown) in all runlevel directories.
service ***on***	Turns a service on, creating a service link in the specified or default runlevel directories
service ***off***	Turns a service off, creating shutdown links in specified or default directories.
service ***reset***	Resets service startup information, creating default links as specified in the chkconfig entry in the service's init.d service script.

To Know about all available switches with chkconfig commands use --help options

```
[root@Server ~]# chkconfig --help
chkconfig version 1.3.30.1 - Copyright (C) 1997-2000 Red Hat, Inc.
This may be freely redistributed under the terms of the GNU Public License.

usage:   chkconfig --list [name]
         chkconfig --add <name>
         chkconfig --del <name>
         chkconfig [--level <levels>] <name> <on|off|reset|resetpriorities>
[root@Server ~]# _
```

To check the status of all services on all runlevel use --list switch with |more options

```
[root@Server ~]# chkconfig --list |more_
```

To check the status of all services on runlevel one use --list switch with |more options

```
[root@Server ~]# chkconfig --list --level 1 |more_
```

To check the status of only vsftpd services on all runlevel one use --list switch with service name

```
[root@Server ~]# chkconfig --list vsftpd
vsftpd          0:off   1:off   2:off   3:off   4:off   5:off   6:off
[root@Server ~]# _
```

To on off vsftpd service on runlevel use on off switch

```
[root@Server ~]# chkconfig vsftpd on
[root@Server ~]# chkconfig --list vsftpd
vsftpd          0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@Server ~]# chkconfig vsftpd off
[root@Server ~]# chkconfig --list vsftpd
vsftpd          0:off   1:off   2:off   3:off   4:off   5:off   6:off
[root@Server ~]# _
```

To deleted vsftpd service use del switch (Note that only service will be delete, not rpm you can add this service again without installing rpm again)

```
[root@Server ~]# chkconfig --list vsftpd
vsftpd          0:off   1:off   2:on    3:on    4:on    5:on
[root@Server ~]# chkconfig --del vsftpd
[root@Server ~]# chkconfig --list vsftpd
service vsftpd supports chkconfig, but is not referenced in any
hkconfig --add vsftpd')
[root@Server ~]# _
```

To add service use add switch (Note rpm must be install first)

```
[root@Server ~]# chkconfig --add vsftpd
[root@Server ~]# chkconfig --list vsftpd
vsftpd          0:off   1:off   2:off   3:off   4:off   5:off   6:off
[root@Server ~]# _
```

The Text Console Service Configuration Tool

If you're managing a large number of services, the command line can be less efficient. You don't need a GUI, just the **ntsysv tool**, which you can open with the command of the same name. However, it affects only services in the current runlevel unless you add an appropriate switch.

For example, if you want to activate several services in runlevels 3 and 5, start ntsysv with the following command:

 # ntsysv --level 35

How to configure linux yum server step by step guide Example and Implementation

YUM stands for **Yellow dog Updater, Modified** because it is based on **YUP**, the **Yellow dog Updater.** Yellow Dog is a version of Linux for the Power Architecture hardware. YUP, and later YUM, were written by the Linux community as a way to maintain an RPM-based system.

Advantages of YUM

Automatic resolution of software dependencies. If a package installation or upgrade request is made and requires the installation or upgrade of additional packages, YUM can list these dependencies and prompt the user to install or upgrade them.

Command-line and graphical versions. The command-line version can be run on a system with a minimal number of software packages. The graphical versions offer ease-of-use and a user-friendly graphical interface to software management.

Multiple software locations at one time. YUM can be configured to look for software packages in more than one location at a time.

Ability to specify particular software versions or architectures. Software locations accessible by YUM can contain multiple versions of the same RPM package and different builds for different architectures such as one for i686 and one for x86_64. yum can easily check the appropriate version and download it.

While it's unlikely that you'll have an Internet connection during the exam, you could have a network connection to a local repository. So you should be ready to use the yum command during the Red Hat exam.

How to create dump of RHEL CD

Whether you perform network installation or create yum repository file you need dump of RHEL CD. It is generally created on server in RHCE exam. Candidate is given a location of this dump to perform network installation. We will create dump of RHEL CD on /var/ftp/pub and use this for network installation or to create yum repository files.

Check how many space is available on /var partition mimimun 4 GB space is required

```
[root@Server ~]# df -h /var
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda2             9.5G  3.6G  5.4G  40% /
[root@Server ~]# _
```

Now mount RHEL dvd on mnt and copy entire disk on /var/ftp/pub

```
[root@Server ~]# mount /dev/dvdwriter /mnt
mount: block device /dev/dvdwriter is write-protected, mounting read-only
[root@Server ~]# cd /mnt
[root@Server mnt]# cp -rf * /var/ftp/pub/
[root@Server mnt]# _
```

Dump is created on /var/ftp/pub now you can umount RHEL dvd

```
[root@Server ~]# umount /mnt
```

Configure yum server

RHCE EXAM QUESTION
Backup of the Redhat Enterprise Linux 5 is taken in /var/ftp/pub on server named Server.example.com. You can install all required packages using yum by creating the repository file.

Pre quest of yum server

we assume that you have completed these pre quest of yum server

- A Linux system with hostname **Server.example.com** and with ip address of **192.168.0.254**
- **Dump** of RHEL disk on **/var/ftp/pub** location

Once you have completed these pre quests follow this guide.

change directory to /var/ftp/pub/Server

```
[root@Server ~]# cd /var/ftp/pub/Server/
[root@Server Server]# _
```

yum and createrepo rpm are required for yum server install them

```
[root@Server Server]# rpm -ivh yum-* --nodeps --force
warning: yum-3.0.1-5.el5.noarch.rpm: Header V3 DSA sig
7186
Preparing...                ###########################
   1:yum-versionlock        ###########################
   2:yum                    ###########################
   3:yum-changelog          ###########################
   4:yum-downloadonly       ###########################
   5:yum-kmod               ###########################
   6:yum-metadata-parser    ###########################
   7:yum-protectbase        ###########################
   8:yum-rhn-plugin         ###########################
   9:yum-security           ###########################
  10:yum-skip-broken        ###########################
  11:yum-updateonboot       ###########################
  12:yum-updatesd           ###########################
  13:yum-utils              ###########################
[root@Server Server]# _
```

Now install createrepo rpm

```
[root@Server Server]# rpm -ivh createrepo-* --nodeps --force
warning: createrepo-0.4.4-2.fc6.noarch.rpm: Header V3 DSA signature:
ID 37017186
Preparing...                ###########################################
   1:createrepo             ###########################################
[root@Server Server]# _
```

After installing necessary package change directory to /var/ftp/pub

```
[root@Server Server]# cd /var/ftp/pub
[root@Server pub]# pwd
/var/ftp/pub
[root@Server pub]# _
```

Now create repository of Server directory

```
[root@Server pub]# createrepo -v Server_
```

repository of all rpm will be created in few minute

```
2156/2159 - freetype-demos-2.2.1-19.el5.i386.rpm
2157/2159 - indent-2.2.9-14.fc6.i386.rpm
2158/2159 - compat-db-4.2.52-5.1.i386.rpm
2159/2159 - libXext-devel-1.0.1-2.1.i386.rpm

Saving Primary metadata
Saving file lists metadata
Saving other metadata
Could not remove old metadata dir: .olddata
Error was [Errno 39] Directory not empty: '/var/
Please clean up this directory manually.
[root@Server pub]#
[root@Server pub]# _
```

Now create repository for VT

```
[root@Server pub]# pwd
/var/ftp/pub
[root@Server pub]# createrepo -v VT_
```

In few second all necessary repository will be created for VT

```
29/31 - virt-manager-0.4.0-3.el5.i386.rpm
30/31 - Virtualization-zh-CN-5.1.0-12.noarch.rpm
31/31 - Virtualization-si-LK-5.1.0-12.noarch.rpm

Saving Primary metadata
Saving file lists metadata
Saving other metadata
Could not remove old metadata dir: .olddata
Error was [Errno 39] Directory not empty: '/var/ft
Please clean up this directory manually.
[root@Server pub]#
[root@Server pub]# _
```

Now create errata directory and repository for it

```
[root@Server pub]# mkdir errata
[root@Server pub]# createrepo -v errata

Saving Primary metadata
Saving file lists metadata
Saving other metadata
[root@Server pub]# _
```

During the process of creating repository two hidden directory with named .olddata is created automatically remove them

```
[root@Server pub]# rm -rf /var/ftp/pub/Server/.olddata
[root@Server pub]# rm -rf /var/ftp/pub/.olddata
[root@Server pub]# _
```

Now check hostname and change directory to /etc/yum.repos.d. copy sample repository file to the file with hostname And open it

```
[root@Server yum.repos.d]# hostname
Server.example.com
[root@Server yum.repos.d]# cd /etc/yum.repos.d/
[root@Server yum.repos.d]# cp -rf rhel-debuginfo.repo Server.example.com.repo
[root@Server yum.repos.d]# vi Server.example.com.repo _
```

Default repository file look like these

```
[rhel-debuginfo]
name=Red Hat Enterprise Linux $releasever - $basearch - Debug
baseurl=ftp://ftp.redhat.com/pub/redhat/linux/enterprise/$rel
arch/Debuginfo/
enabled=0
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
```

Remove defaults line and set new location of Sever and VT as shown here

```
[Server]
name=Server.example.com
baseurl=file:///var/ftp/pub/Server
gpgcheck=0

[VT]
name=Server.example.com
baseurl=file:///var/ftp/pub/VT
gpgcheck=0
_
```

Save file with :wq and exit

Now remove all temporary data file with yum clean all command

```
[root@Server ~]# yum clean all
Loading "protectbase" plugin
Loading "installonlyn" plugin
Loading "rhnplugin" plugin
Loading "security" plugin
Loading "changelog" plugin
Loading "skip-broken" plugin
Loading "kmod" plugin
Loading "downloadonly" plugin
This system is not registered with RHN.
RHN support will be disabled.
Cleaning up Everything
[root@Server ~]# _
```

Congratulation You have successful create yum server

To test yum server remove telnet package

```
[root@Server ~]# yum remove telnet_
```

After checking all dependences it will ask for conformation press y

```
Transaction Summary
================================================
Install       0 Package(s)
Update        0 Package(s)
Remove        1 Package(s)

Is this ok [y/N]: y
Downloading Packages:
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Removing  : telnet

Removed: telnet.i386 1:0.17-38.el5
Complete!
[root@Server ~]# _
```

Now install telnet package from yum server

```
[root@Server ~]# yum install telnet_
```

After checking all dependences it will ask for conformation press y

```
Install       1 Package(s)
Update        0 Package(s)
Remove        0 Package(s)

Total download size: 56 k
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Test
warning: telnet-0.17-38.el5: Header V3
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing: telnet

Installed: telnet.i386 1:0.17-38.el5
Complete!
[root@Server ~]# _
```

PC as router,create virtual lan card.

In this article I will show you that how can you use Linux as a router. Routers are the devices those are used to connect two different networks. Routers are very costly devices. Linux could be a cost effective solution of routing in a small company.

Exam question
Your system is going use as a router for 192.168.0.0/24 and 192.168.1.0/24. Enable the IP forwarding.

Linux as a Router

In this practical we are using three computers. One Linux system will be use for routing and reset two will remain in two different networks. First we will configure the system which is going to play the role of router.

How to create virtual LAN card

Configure server system

You need two LAN card for routing between two networks or you can create virtual LAN card instead of deploying them physically.

To create virtual Ethernet card change directory to /etc/sysconfig/network-scripts

```
[root@Server ~]# cd /etc/sysconfig/network-scripts/
[root@Server network-scripts]# ls
ifcfg-eth0     ifdown-isdn     ifup-aliases  ifup-plip
ifcfg-lo       ifdown-post     ifup-bnep     ifup-plusb
ifdown         ifdown-ppp      ifup-eth      ifup-post
ifdown-bnep    ifdown-routes   ifup-ippp     ifup-ppp
ifdown-eth     ifdown-sit      ifup-ipsec    ifup-routes
ifdown-ippp    ifdown-sl       ifup-ipv6     ifup-sit
ifdown-ipsec   ifdown-tunnel   ifup-ipx      ifup-sl
ifdown-ipv6    ifup            ifup-isdn     ifup-tunnel
[root@Server network-scripts]# _
```

ifcfg-eth0 is the necessary script file for Ethernet 0. Copy this file to the same folder to create new virtual LAN cards.

```
[root@Server network-scripts]# cp ifcfg-eth0 ifcfg-eth0.1
[root@Server network-scripts]# _
```

Now on this newly created virtual LAN card. It could be done by service network restart

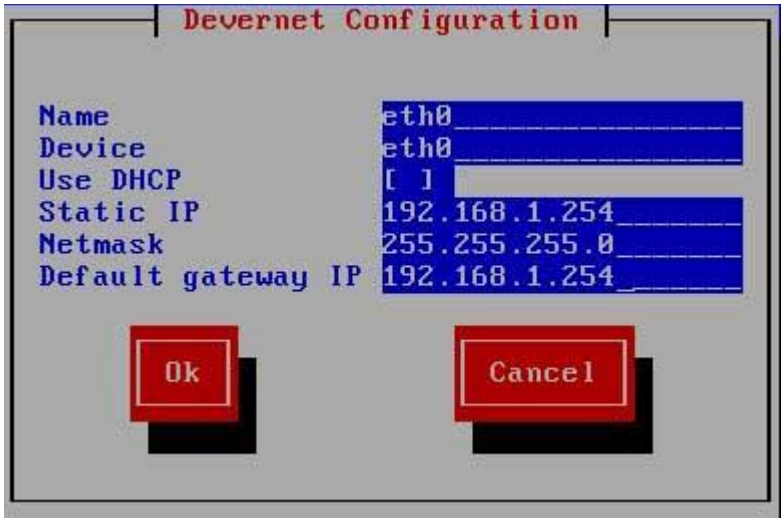
```
[root@Server network-scripts]# service network restart
Shutting down interface eth0:                              [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface eth0:                                [  OK  ]
Bringing up interface eth0.1:                              [  OK  ]
[root@Server network-scripts]# _
```

Run setup command and select network configuration sub window from list

```
[root@Server network-scripts]# setup_
```

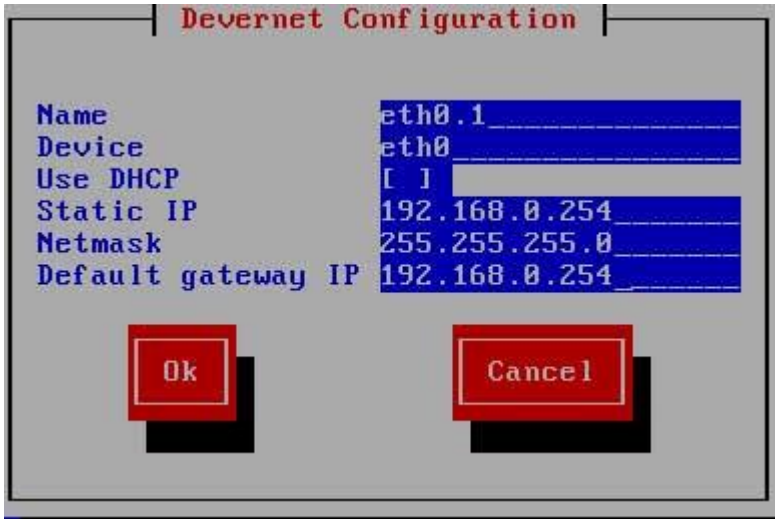
You have two LAN card here, select eth0 from list to assign IP

This Ethernet card will be the default gateway of first network set its IP to 192.168.1.254 and click on ok



Now select eth0.1 (This our virtual LAN card which we create in our last sections)

Set its IP to 192.168.0.254 This will be the default gateway of other network. Click on OK then quit and quit to come back on command prompt



IP forwarding can be enabled by editing in /etc/sysctl.conf file. open this file

```
[root@Server network-scripts]# vi /etc/sysctl.conf _
```

Locate the net.ipv4. ip_forward = 0 tag. and replace the value 0 to 1. This will enable IP forwarding to permanently . But this require a system reboot.

```
# Kernel sysctl configuration file for Red Hat Linux
#
# For binary values, 0 is disabled, 1 is enabled.  See
# sysctl.conf(5) for more details.

# Controls IP packet forwarding
net.ipv4.ip_forward = 1

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1

# Do not accept source routing
net.ipv4.conf.default.accept_source_route = 0
```

If don't want to restart the system you can tell running kernel directly by echo command and kernel will enable the IP forwarding

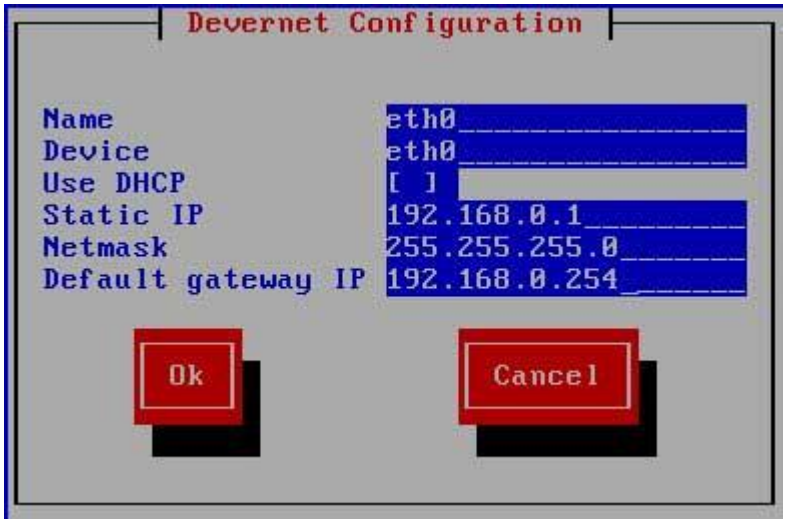
```
[root@Server network-scripts]# echo "1"> /proc/sys/net/ipv4/ip_forward
[root@Server network-scripts]# _
```

now configure our client system. we are using two system one from each network to test the connectivity .

Our first system is a Linux machine run setup command on it

```
[root@Client1 ~]# setup_
```

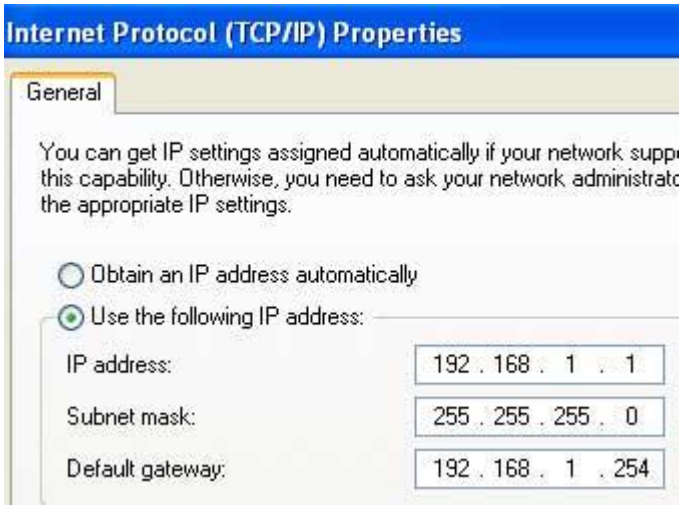
assign its IP address to 192.168.0.1 with a default gateway of 192.168.0.254



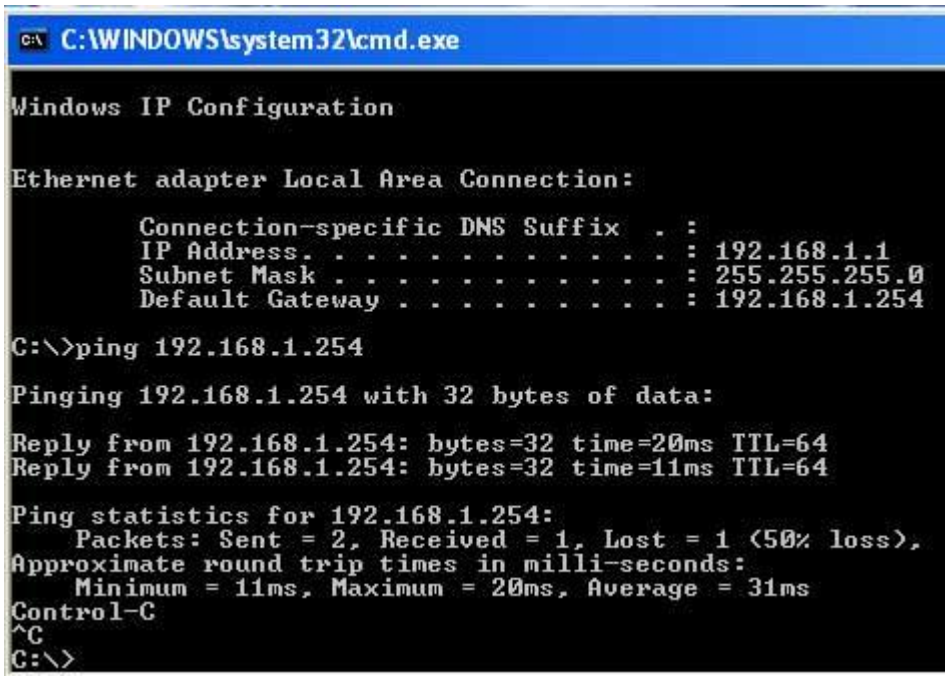
now restart the network service and check connectivity form its default gateway (Server IP)

```
[root@Client1 network-scripts]# service network restart
Shutting down interface eth0:                              [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface eth0:                                [  OK  ]
[root@Client1 network-scripts]# ping 192.168.0.254
PING 192.168.0.254 (192.168.0.254) 56(84) bytes of data.
64 bytes from 192.168.0.254: icmp_seq=1 ttl=64 time=3.97 ms
64 bytes from 192.168.0.254: icmp_seq=2 ttl=64 time=0.282 ms

--- 192.168.0.254 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.282/2.126/3.970/1.844 ms
[root@Client1 network-scripts]# _
```

Now go on our other host which we are using a window machine (You can also use Linux host) and set IP address to 192.168.1.1 with a default gateway to 192.168.1.254



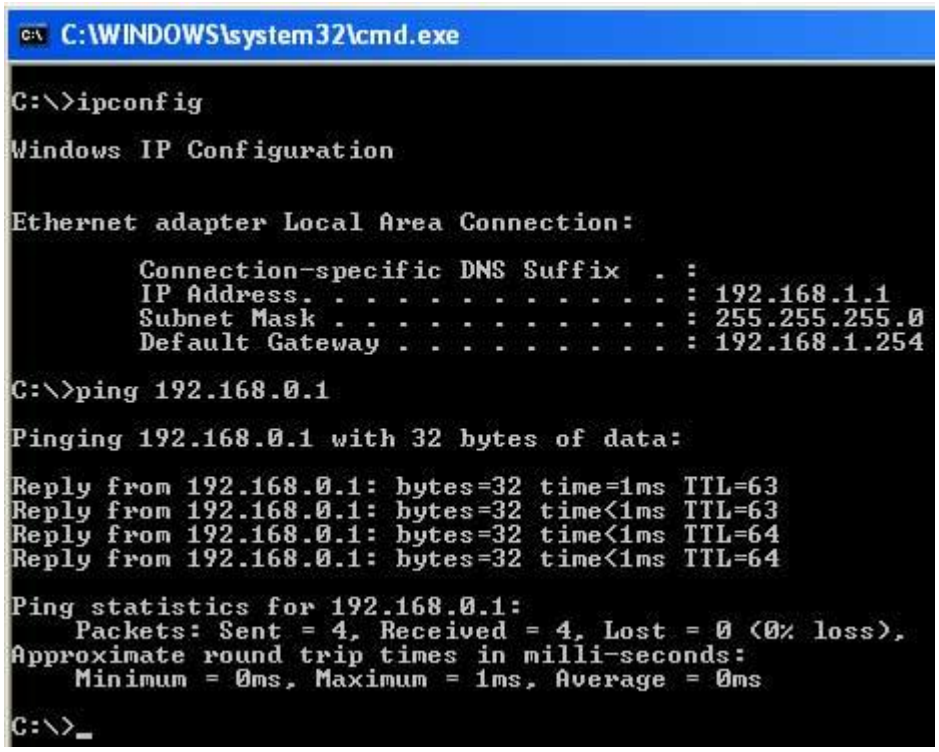
now open command prompt and test connectivity with default gateway



At this point you have completed all necessary step's to enable routing its time to verify this

Test from windows system

ping the Linux host located on other network



Test from Linux system

ping the Window host located on other network

```
[root@Client1 network-scripts]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:62:28:1A
          inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe62:281a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:344 errors:0 dropped:0 overruns:0 frame:0
          TX packets:436 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:46609 (45.5 KiB)  TX bytes:69077 (67.4 KiB)
          Interrupt:67 Base address:0x2000

[root@Client1 network-scripts]# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
From 192.168.0.254: icmp_seq=1 Redirect Host(New nexthop: 192.168.1.1)
64 bytes from 192.168.1.1: icmp_seq=1 ttl=127 time=0.808 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=127 time=0.525 ms

--- 192.168.1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.525/0.666/0.808/0.143 ms
[root@Client1 network-scripts]# _
```