Linux System Administrator's Guide

Deployment, Configuration, and Administration of Red Hat / CentOs Linux 7

Legal Notice

Abstract

The System Administrator's Guide documents relevant information regarding the deployment, configuration, and administration of Red Hat Enterprise Linux 7/ CentOs 7. It is oriented towards system administrators with a basic understanding of the system. To expand your expertise, you might also be interested in the Red Hat System Administration I (RH124), Red Hat System Administration II (RH134), Red Hat System Administration III (RH254), or RHCSA Rapid Track (RH199) training courses.

System Administrator (RHCSA)- SA1

Table of Contents

1. Basic Command

In Linux everything is file and to manage these files Linux administrator use command line tools. In this section you will learn how to use command line tools to manage files. We would start from basic and extend it till RHCE exam level. You would learn how to login in command prompt, how to use Linux shell, how to use text editor in command line and many more.

From RHCE7 command line skills are combined with network configuration objectives. In this section we would try to cover those commands with example which requires in RHCE exam.

This section is good for beginner as well. We suggest you to go through with each articles of this section as linux administrators frequently use commands to perform administrative task.

Understanding command prompt

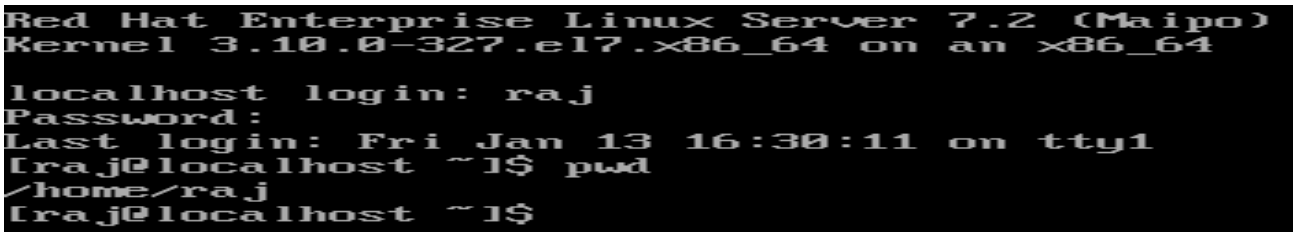
On a default Redhat system terminal console look like this.

Above image contain following information.

Release version of RHEL	Red Hat Enterprise Linux Server release 7.0)
Version number of the kernel	Kernel 3.10.0-123.el7.x86_64 on an x86_64
System hostname	Localhost

Which command line prompt are you going to get, it depends on user type. However, you could customize the prompt. Linux systems have two types of users: - super user known as root user and normal user. For these Linux system have two basic prompts. The following is an example of what you might see when logged in as a normal user:

Note how it includes the username, the hostname of the local system, the current directory, and a $ prompt. The $ prompt is the standard for normal users.



In above example

raj	Username
Localhost	computer name
~	user's home directory in this example it would be /home/raj Every user by default gets a directory in home folder.

Now take a look at a prompt for the root administrative user on the same system. It should look familiar. Except for the name of the account, the only consistent difference is the prompt.

In this example

Root	Username
Localhost	computer name
~	user's home directory in this example it would be /root Root user get its directory separate from other users.

RHEL Terminal Emulation CLI Interface

In the early days of Linux (around1990s) all that was available was a simple text interface to the Linux operating system. This text interface allowed administrators to start programs, control program operations, and move files around on the system.

With the popularity of Microsoft Windows, computer users expected more than the old text interface to work with. This spurred more development in the OSS community, and the Linux graphical desktops emerged.

Back before the days of graphical desktops, the only way to interact with a Unix system was through a text command line interface (CLI) provided by the shell. The CLI allowed text input only, and could only display text and rudimentary graphics output.

As you well know, things are significantly different in today's Linux environment. Just about every Linux distribution uses some type of graphical desktop environment. However, to access the shell you still need a text display to interact with a CLI.

Linux is famous for being able to do things in more than one way, and no place is this more relevant than in graphical desktops.

With all of the new graphical Linux desktop features, sometimes finding a way to get a CLI in a Linux distribution is an easy task.

One way to get to a CLI is to take the Linux system out of graphical desktop mode and place it in text mode. This provides nothing more than a simple shell CLI on the monitor, just like the days before graphical desktops. This mode is called the Linux console, since it emulates the old days of a hard-wired console terminal, and is a direct interface to the Linux system.

By default, 6 command line consoles and one GUI console is available in Redhat system. They're defined in start-ttys.conf file in the /etc/init directory.

To change between consoles, press ALT and the function key associated with the console. For example, the ALT-F5 key combination moves to the five console.

In the RHEL GUI, the ALT-F2 key combination is used to start the Run Application tool; therefore, you'll need to press CTRL-ALT-F2 to move to that second virtual console.

At a text console login, you'd see the above prompt, which depends a bit on the release of RHEL, the version number of the kernel, and the system hostname:

File navigation in command line

New comer in Linux may prefer to use the GUI to manage Linux file system. But Linux administrator/ Linux professional use command line to manage Linux file system because command line tools give you ability to find your way around every Linux distribution.

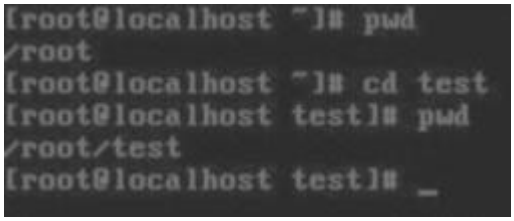
For RHCE certification you need some basic concepts of file navigation to find the important files. In this article I would show some basic command which you need in RHCE certification to navigate the RHEL system.

pwd

On a Linux you could easily find out the name of current directory. The **pwd** command identifies the current directory and prints the name of current directory. You can easily remember it as present working directory.

cd

Simple command to change the directory.



absolute and relatives path before you start playing around the Linux file system be familiar and understand the difference between absolute and relative path.

An absolute path describes the complete directory structure in terms of the top-level directory, root (/).

A relative path is based on the current directory. Relative paths do not include the slash in front.

Let take a simple example to understands distinguish between absolute and relative path. You are logged in from root user and want to go in /home/sanjay directory

From relative path

```
[root@localhost ~]# pwd
/root
[root@localhost ~]# cd /home/sanjay/
[root@localhost sanjay]# pwd
/home/sanjay
[root@localhost sanjay]# _
```

From absolute path

```
[root@localhost Desktop]# useradd sanjay
[root@localhost Desktop]# cd /home/
[root@localhost home]# pwd
/home
[root@localhost home]# cd sanjay/
[root@localhost sanjay]# pwd
/home/sanjay
```

Tilde (~)

On Linux system every user after successful login is taken to his home directory. The tilde (~)is used to represent the home directory of any currently active user.
For example,when user raj logs in, he's taken to his home directory, /home/raj.

```
Red Hat Enterprise Linux Server 7.2 (Maipo)
Kernel 3.10.0-327.el7.x86_64 on an x86_64

localhost login: raj
Password:
Last login: Fri Jan 13 16:30:11 on tty1
[raj@localhost ~]$ pwd
/home/raj
[raj@localhost ~]$
```

In contrast, the home directory of the root administrative user is /root.

```
Red Hat Enterprise Linux Server 7.0 (Maipo)
Kernel 3.10.0-123.el7.x86_64 on an x86_64

server86 login: root
Password:
Last login: Tue Apr 28 12:05:49 on tty3
[root@server86 ~]# pwd
/root
[root@server86 ~]#
```

To return in home directory from anywhere in file system

Use cd command without any argument.

```
[root@localhost var]# pwd
/var
[root@localhost var]# cd
[root@localhost ~]# pwd
/root
[root@localhost ~]# _
```

It's not necessary but you could pass ~ with cd command to return the home directory of user.

```
[root@localhost var]# pwd
/var
[root@localhost var]# cd ~
[root@localhost ~]# pwd
/root
[root@localhost ~]# _
```

Thus, the effect of the cd ~ command depends on your username.
For example, if you've logged in as user sanjay, the cd ~ command navigates to the /home/sanjay directory.
If you've logged in as the root user, this command navigates to the /root directory.

Listing the content of directory

Ls command is helpful to see what files exist in a directory.

```
[root@localhost ~]# ls
anaconda-ks.cfg  Documents  install.log         Music
Desktop          Downloads  install.log.syslog  Pictures
[root@localhost ~]# _
```

Ls command with the right switches, can be quite powerful.

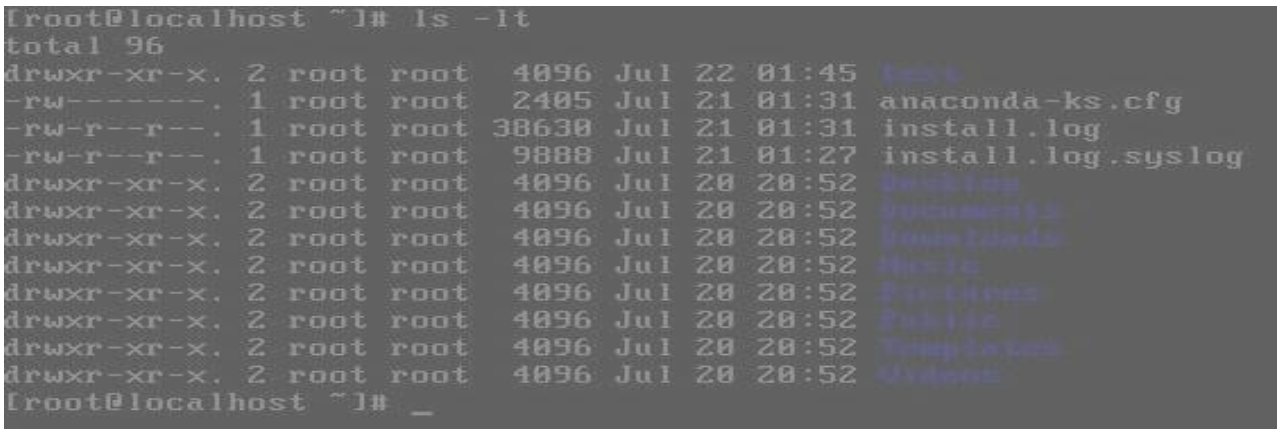
ls -a :- to reveal hidden files

```
[root@localhost ~]# ls -a
                  .dbus     .gnupg              Music
                  Desktop   .gstreamer-0.10     .nautilus
anaconda-ks.cfg   Documents .gtk-bookmarks      Pictures
.bash_history     Downloads .gvfs               Public
.bash_logout      .esd_auth .ICEauthority       .pulse
.bash_profile     .gconf    .local              .pulse-cookie
.bashrc           .gconfd   install.log         .spice-vdagent
.config           .gnome2   install.log.syslog  .ssh
.cshrc            .gnote    .lesshst            .tcshrc
[root@localhost ~]# _
```

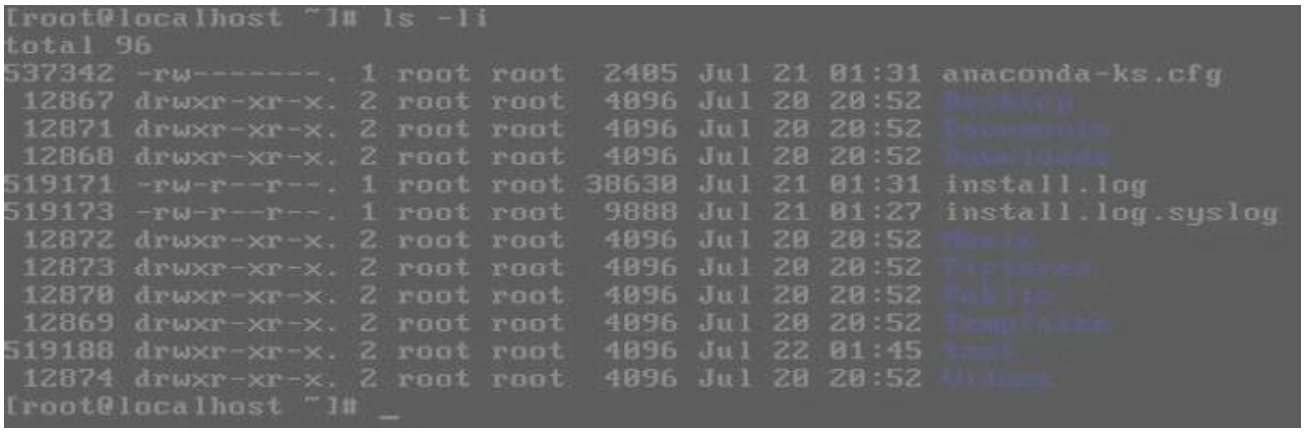
ls -l :- for long listings

```
[root@localhost ~]# ls -l
total 96
-rw-------. 1 root root  2405 Jul 21 01:31 anaconda-ks.cfg
drwxr-xr-x. 2 root root  4096 Jul 20 20:52 Desktop
drwxr-xr-x. 2 root root  4096 Jul 20 20:52 Documents
drwxr-xr-x. 2 root root  4096 Jul 20 20:52 Downloads
-rw-r--r--. 1 root root 38630 Jul 21 01:31 install.log
-rw-r--r--. 1 root root  9888 Jul 21 01:27 install.log.syslog
drwxr-xr-x. 2 root root  4096 Jul 20 20:52 Music
drwxr-xr-x. 2 root root  4096 Jul 20 20:52 Pictures
drwxr-xr-x. 2 root root  4096 Jul 20 20:52 Public
drwxr-xr-x. 2 root root  4096 Jul 20 20:52 Templates
drwxr-xr-x. 2 root root  4096 Jul 22 01:45 test
drwxr-xr-x. 2 root root  4096 Jul 20 20:52 Videos
[root@localhost ~]# _
```

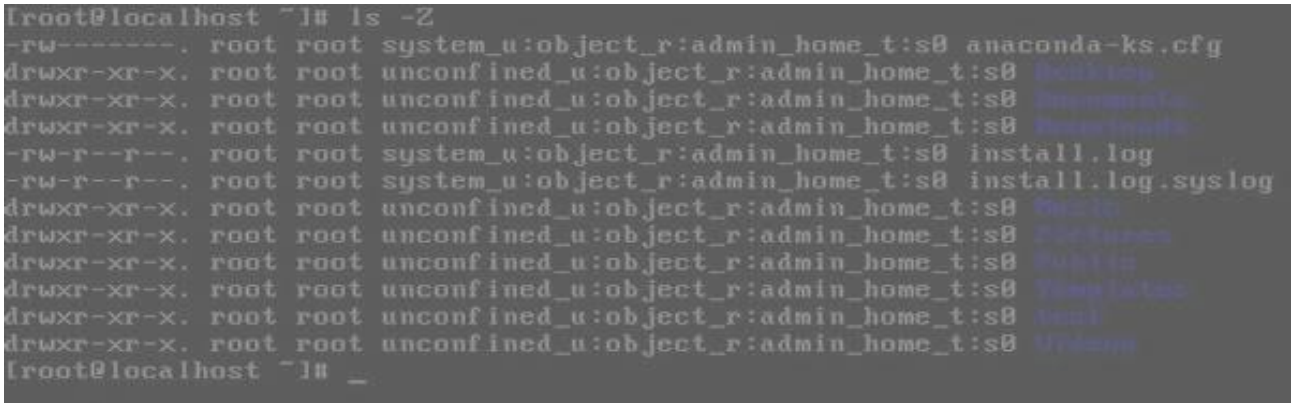
ls -t :- for a time-based list



ls -i :- for inode numbers



ls –Z :- The system_u, object_r, var_t, and s0 output demonstrates the current SELinux contexts of the noted files. During the RHCE you will be expected to configure a system with SELinux enabled.



Basic commands to manage files

Linux and Unix are managed through a series of text files. Linux administrators do not normally use graphical editors to manage these configuration files. Editors such as WordPerfect, OpenOffice.org Writer, Microsoft Word normally either save files in a binary format or add tags. Unless text files are

preserved in their original format, without tags, changes that are made can render a Linux system unbootable.

In RHCE certification you have to perform several tasks on command line. In this article I would some basic commands to create / edit /delete files on Linux system.

How to create and read files on command line in RHEL Linux

Everything in Linux can be reduced to a file.

cat

The most basic command for reading and creating files is **cat**. The cat filename command scrolls the text within the filename file. It also works with multiple filenames; it concatenates the filenames that you might list as one continuous output to your screen. You can redirect the output to the filename of your choice.

To Create a file

```
[sanjay@localhost ~]$ cat > file1
This is 1 line in file1
Now we are in 2 line we can not go back in line1
Press d with ctrl key to save file
Do not press d with ctrl key second time as on
first time file would be saved and you would return
to command prompt and if you press again ctrl d it would

logged you out. ctrl + d is the short cut key of logged out on
command prompt
[sanjay@localhost ~]$ _
```

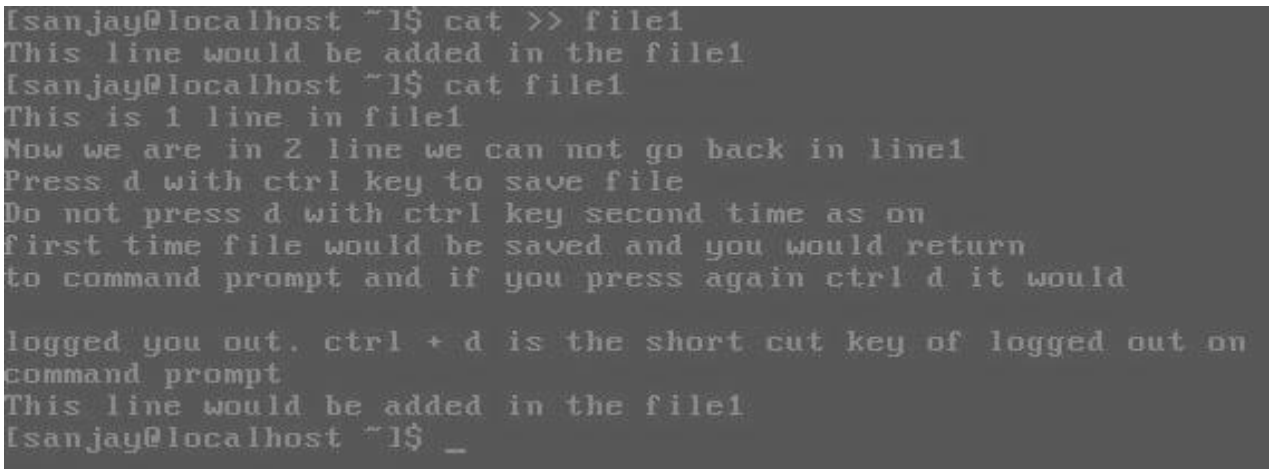
Use ctrl+d to save file

To read the contain of file

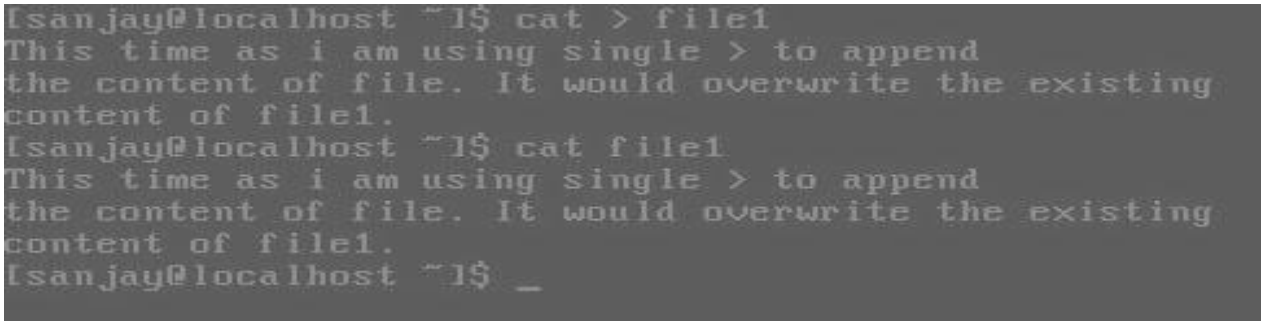
```
[sanjay@localhost ~]$ ls
file1
[sanjay@localhost ~]$ cat file1
This is 1 line in file1
Now we are in 2 line we can not go back in line1
Press d with ctrl key to save file
Do not press d with ctrl key second time as on
first time file would be saved and you would return
to command prompt and if you press again ctrl d it would

logged you out. ctrl + d is the short cut key of logged out on
command prompt
[sanjay@localhost ~]$ _
```

To extend the contain of file use >> with cat command



While extending contents from cat command take care of > . User single > to create new file and >> to extend the contents. If you have used single > to extended the content if would overwrite the existing content of file without any warning.

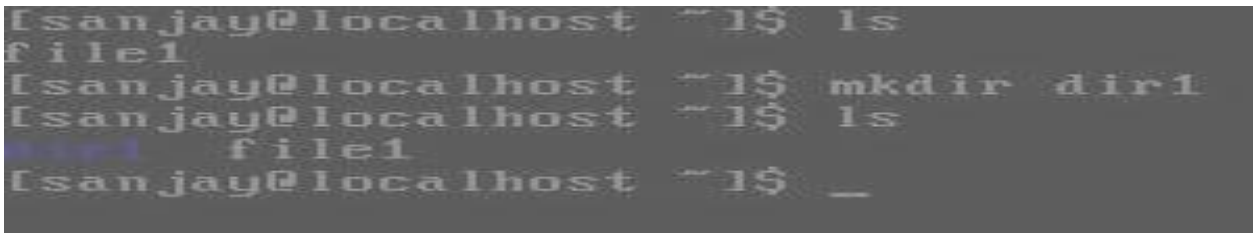


How to create directories on command line in RHEL Linux?

Directories are special types of files that serve as containers for other files.

mkdir

Basic command to create new directory.



To create directory tree use -p

```
[sanjay@localhost ~]$ ls
dir1  file1
[sanjay@localhost ~]$ mkdir -p dir2/dir_a/dir_b/dir_c
[sanjay@localhost ~]$ ls
dir1  dir2  file1
[sanjay@localhost ~]$ cd dir2
[sanjay@localhost dir2]$ ls
dir_a
[sanjay@localhost dir2]$ cd dir_a
[sanjay@localhost dir_a]$ cd dir_b
[sanjay@localhost dir_b]$ cd dir_c
[sanjay@localhost dir_c]$ pwd
/home/sanjay/dir2/dir_a/dir_b/dir_c
[sanjay@localhost dir_c]$ _
```

How to delete files / directories on command line in RHEL Linux

To remove directory use **rmdir** command

```
[sanjay@localhost ~]$ ls
dir1  dir2
[sanjay@localhost ~]$ rmdir dir1
[sanjay@localhost ~]$ ls
dir2
[sanjay@localhost ~]$ rmdir dir2
rmdir: failed to remove `dir2': Directory not empty
[sanjay@localhost ~]$ _
```

To delete file use rm command

```
[sanjay@localhost ~]$ ls
dir1  dir2  file1
[sanjay@localhost ~]$ rm file1
[sanjay@localhost ~]$ ls
dir1  dir2
[sanjay@localhost ~]$ _
```

rmdir command would not delete a directory which contain data inside it
To delete a directory which contain data inside it use rm command with –rf option

```
[sanjay@localhost ~]$ rm -rf dir2
[sanjay@localhost ~]$ ls
[sanjay@localhost ~]$ _
```

Reading the content of file

From our earlier articles of this section you have learnt how to create files and directories. You have also played around the Linux file system. Now its time to read contents of files with some advance options.

less and more

Everything in Linux is file. Sometime you may find its very hard to read the entire file. To navigate in the file Linux, provide some cool commands.

Linux has two of these commands: more and less. With the more filename command,

```
[root@localhost ~]# more /var/log/messages
```

you can scroll through the text of a file, from start to finish, one screen at a time. With the less filename command,

```
[root@localhost ~]# less /var/log/messages
```

you can scroll in both directions. Use PAGE UP and PAGE DOWN keys to scroll.

Press **q** anytime to quit from the output of the commands.

During the exam sometime you need to find some text in files. Like in configuration files you need to find some configuration values. Or you may need to locate some error in error log files.

As the less and more commands do not change files, they are an excellent way to scroll through and search for items.

Let take an example open /var/log/messages files with less command

```
[root@localhost ~]# less /var/log/messages
```

Now I would search the term"pid_max", in forward

```
Jul 21 02:14:31 localhost
 (reserved)
Jul 21 02:14:31 localhost
 (reserved)
Jul 21 02:14:31 localhost
 (reserved)
/pid_max_
```

To search in the reverse direction, substitute a? for the /.

```
Jul 21 02:14:31 localhost
Jul 21 02:14:31 localhost
Jul 21 02:14:31 localhost
Jul 21 02:14:31 localhost
Jul 21 02:14:31 localhost
? pid_max_
```

Found string would be shown like this

```
Jul 21 02:14:31 localhost kernel: pid_max: default: 32768 minimum: 301
Jul 21 02:14:31 localhost kernel: Security Framework initialized
Jul 21 02:14:31 localhost kernel: SELinux:  Initializing.
Jul 21 02:14:31 localhost kernel: Dentry cache hash table entries: 131072
: 8  1048576 bytes)
```

With more command you could only search in forward direction. One more cool feature of less command is it could read the contents in Gzip format compress files without uncompressing them. This features only available in less commands Gzip format, normally shown with the .gz extension.

```
[root@localhost ~]# less /usr/share/man/man1/cat.1.gz
```

Viewing parts of a file Often the data you want to view is located either right at the top or buried at the bottom of a text file. If the information is at the top of a large file, you still need to wait for the cat or more commands to load the entire file before you can view it. If the information is located at the bottom of a file (such as a log file), you need to wade through thousands of lines of text just to get to the last few entries. Fortunately, Linux has specialized commands to solve both of these problems.

head and tail

The **tail** command displays the last group of lines in a file. By default, it will show the last 10 lines in the file.

```
[root@localhost ~]# tail /var/log/messages
Jul 22 05:39:37 localhost NetworkManager[1303]: <info>   gateway 192.168.219.2
Jul 22 05:39:37 localhost NetworkManager[1303]: <info>   nameserver '192.168.219
.2'
Jul 22 05:39:37 localhost NetworkManager[1303]: <info>   domain name 'localdomai
n'
Jul 22 05:39:37 localhost avahi-daemon[1320]: Withdrawing address record for 192
.168.219.130 on eth0.
Jul 22 05:39:37 localhost avahi-daemon[1320]: Leaving mDNS multicast group on in
terface eth0.IPv4 with address 192.168.219.130.
Jul 22 05:39:37 localhost avahi-daemon[1320]: Interface eth0.IPv4 no longer rele
vant for mDNS.
Jul 22 05:39:37 localhost avahi-daemon[1320]: Joining mDNS multicast group on in
terface eth0.IPv4 with address 192.168.219.131.
Jul 22 05:39:37 localhost avahi-daemon[1320]: New relevant interface eth0.IPv4 f
or mDNS.
Jul 22 05:39:37 localhost avahi-daemon[1320]: Registering new address record for
 192.168.219.131 on eth0.IPv4.
Jul 22 05:39:38 localhost NetworkManager[1303]: <info> Policy set 'System eth0'
(eth0) as default for IPv4 routing and DNS.
[root@localhost ~]# _
```

but you can change it with -n switch. For example, to see the 2-line use

```
[root@localhost ~]# tail -2 /var/log/messages
Jul 22 05:39:37 localhost avahi-daemon[1320]: Registering new address record for
 192.168.219.131 on eth0.IPv4.
Jul 22 05:39:38 localhost NetworkManager[1303]: <info> Policy set 'System eth0'
(eth0) as default for IPv4 routing and DNS.
[root@localhost ~]# _
```

The **head** command While not as exotic as the tail command, the head command does what you would expect, it displays the first group of lines at the start of a file. By default, it will display the first 10 lines of text.

```
[root@localhost ~]# head /var/log/messages
Jul 21 02:14:31 localhost kernel: imklog 4.6.2, log source = /proc/kmsg started.
Jul 21 02:14:31 localhost rsyslogd: [origin software="rsyslogd" swVersion="4.6.2
" x-pid="1241" x-info="http://www.rsyslog.com"] (re)start
Jul 21 02:14:31 localhost kernel: Initializing cgroup subsys cpuset
Jul 21 02:14:31 localhost kernel: Initializing cgroup subsys cpu
Jul 21 02:14:31 localhost kernel: Linux version 2.6.32-131.0.15.el6.x86_64 (mock
build@x86-007.build.bos.redhat.com) (gcc version 4.4.4 20100726 (Red Hat 4.4.4-1
3) (GCC) ) #1 SMP Tue May 10 15:42:40 EDT 2011
Jul 21 02:14:31 localhost kernel: Command line: ro root=UUID=287b57d9-4db9-40ff-
9d6a-3a438b280a7b rd_NO_LUKS rd_NO_LVM rd_NO_MD rd_NO_DM LANG=en_US.UTF-8 SYSFON
T=latarcyrheb-sun16 KEYBOARDTYPE=pc KEYTABLE=us crashkernel=auto quiet
Jul 21 02:14:31 localhost kernel: KERNEL supported cpus:
Jul 21 02:14:31 localhost kernel:  Intel GenuineIntel
Jul 21 02:14:31 localhost kernel:  AMD AuthenticAMD
Jul 21 02:14:31 localhost kernel:  Centaur CentaurHauls
[root@localhost ~]# _
```

Similar to the tail command, you can use -n switch

```
[root@localhost ~]# head -n 2 /var/log/messages
Jul 21 02:14:31 localhost kernel: imklog 4.6.2, log source = /proc/kmsg started.
Jul 21 02:14:31 localhost rsyslogd: [origin software="rsyslogd" swVersion="4.6.2
" x-pid="1241" x-info="http://www.rsyslog.com"] (re)start
[root@localhost ~]# _
```

Linux Environment Variables

In this article we would understand linux environment variables. Environmental variable plays significant role in linux system.

We would cover following topics in this article

- What is a linux environment variable
- How to show linux environment variable
- How to set linux environment variable
- How to set linux environmental variable permanently
- List of Linux environmental variable

What is a linux environment variable

Linux environment variable is an object that contains value. In simple terms it is a pair of data object and their respective values. If you are familiar with programming language than you can easily understand it. Linux environment variables do same job which variables do in programming language.

If you are not familiar with programming language you can understand linux variable as a container with name which keeps value inside it. This value could be location of all executable files in the filesystem, the default editor that should be used, or the system locale settings.

Example of linux environmental variable

Let take a simple example of ***ls*** command to understand linux environmental variables. ***ls*** the basic command to list the content of directory. When execute a command in linux, you need to type the full path of that command. Since the ***ls*** command is in the ***/bin*** directory, users should execute the ***/bin/ls*** command to list files in the current directory.

Here comes the magic of linux environmental variables. Linux have a PATH variable. With the help of the PATH variable, full path is not required. The bash shell automatically searches through the directories listed in a user's PATH variable for the command that user just typed at the command line. When a matching command found shell run it. In this way environment variable provides a simple way to share configuration settings between multiple applications and processes in Linux.

How to show linux environment variable

printenv or ***env*** command can be use to list linux environment variables. The ***coreutils*** package contains ***printenv*** and ***env***. Use ***printenv*** command to show linux environmental variables.

$printenv

```
[user1@server ~]$ printenv_
```

The ***env*** utility can also be used to show linux environment variables.

```
$env
```

```
[user1@server ~]$ env
```

printenv to print the names and the values of each. Note that some environment variables are user-specific. Output of ***env*** and ***printenv*** are too big to fit in screen. We can redirect the output of ***printenv*** in a file. To redirect the output of ***printenv*** in a file run following command

```
$printenv > tmp_file
```

```
[user1@server ~]$ printenv > tmp_file
[user1@server ~]$ _
```

now we can read the tmp_file with less command.

```
$less tmp_file
```

```
[user1@server ~]$ printenv > tmp_file
[user1@server ~]$ less tmp_file
```

Use up arrow and down arrow key to scroll. Press ***q*** to exist from file

```
$env
```

```
HOSTNAME=server.example.com
TERM=linux
SHELL=/bin/bash
HISTSIZE=1000
XDG_SESSION_COOKIE=b5bd23c216a40532b967cf
QTDIR=/usr/lib64/qt-3.3
QTINC=/usr/lib64/qt-3.3/include
USER=user1
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi
=40;33;01:or=40;31;01:mi=01;05;37;41:su=3
```

After reading you can simple remove the temporary file

```
$rm tmp_file
```

```
HOME=/home/user1
LOGNAME=user1
QTLIB=/usr/lib64/qt-3.3/lib
CVS_RSH=ssh
LESSOPEN=|/usr/bin/lesspipe.sh %s
G_BROKEN_FILENAMES=1
_=/usr/bin/printenv
[user1@server ~]$ rm tmp_file
[user1@server ~]$
```

How to set linux environment variable

Linux environment can be set in three ways.

- Temporary also know as Session Specific Variables
- permanent locally
- Permanent globally

To set linux environmental variable temporary use ***export*** command.

PATH variable contains location of executable files. To check current ***PATH*** use following command

```
$ echo $PATH
```

```
[user1@server ~]$ echo $PATH
/usr/lib64/qt-3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sb
in:/home/user1/bin
[user1@server ~]$ _
```

Now we would add our directory in this path. Make a directory

```
$mkdir custom_script
```

To add this directory in path run following commands

```
$export PATH="${PATH}:/home/user1/custom_script"
```

Verify that we have successfully added our *custom_script* directory in ***PATH*** variable

```
[user1@server ~]$ echo $PATH
/usr/lib64/qt-3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sb
in:/home/user1/bin
[user1@server ~]$ mkdir custom_script
[user1@server ~]$ export PATH="${PATH}:/home/user1/custom_script"
[user1@server ~]$ echo $PATH
/usr/lib64/qt-3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sb
in:/home/user1/bin:/home/user1/custom_script
[user1@server ~]$ _
```

Now move in our *custom_script* directory and make a sample script

```
[user1@server custom_script]$ cat > simple_script
echo "Hello world "
[user1@server custom_script]$ _
```

make *sample_script* executable and run script directly from command prompt.

```
[user1@server custom_script]$ cat > simple_script
echo "Hello World"
[user1@server custom_script]$ chmod +x simple_script
[user1@server custom_script]$ simple_script
Hello World
[user1@server custom_script]$ _
```

You can also verify that shell run *sample_script* from our *custom_script* directory by which command

```
[user1@server custom_script]$ cat > simple_script
echo "Hello World"
[user1@server custom_script]$ chmod +x simple_script
[user1@server custom_script]$ simple_script
Hello World
[user1@server custom_script]$ which simple_script
~/custom_script/simple_script
[user1@server custom_script]$ ls
simple_script
[user1@server custom_script]$ _
```

Temporary variable only available in current session. To test it log out from current user and login back.

Run *sample_script* again. This time you will get command not found error.

```
[user1@server ~]$ cd custom_script/
[user1@server custom_script]$ ls
simple_script
[user1@server custom_script]$ simple_script
-bash: simple_script: command not found
[user1@server custom_script]$ _
```

How to set linux environmental variable permanently

Defining Variables Locally

As you seen temporary variables are available only on that session. We can make those variables permanent. For security reason you should not define an environment variable globally unless you have sound understanding of linux system. For instance, you might want to add */home/user_name/custom_script* to the ***PATH*** variable for a particular user. In such a case define it locally. As you do not want all other users on your system to have that in their PATH too.

The following files should be used for local environment variables on your system: ~/. profile, ~/.bash_profile, ~/.bash_login and ~/.bash_logout.

```
[user1@server ~]$ ls -a
                                    .pulse-cookie
                    .ICEauthority   .recently-used.xbel
.bash_history
.bash_logout
.bash_profile       .esd_auth       tmp
.bashrc
                    .gtk-bookmarks  .Xauthority
                                    .xsession-errors
                                    .xsession-errors.old
[user1@server ~]$ _
```

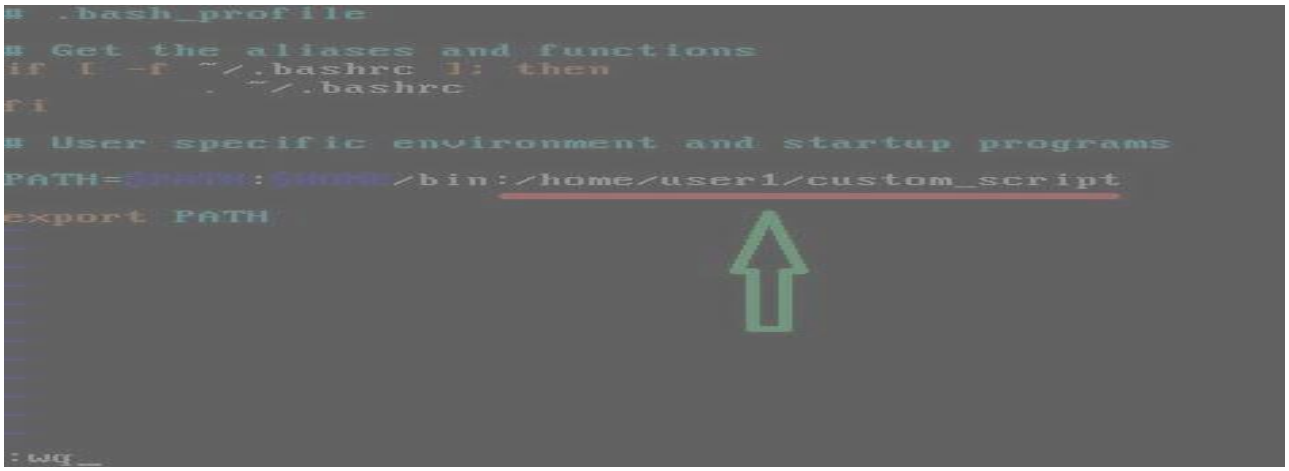
To add our custom_script directory in to the PATH variable for local usage

open .bash_profile file

```
vi ~/.bash_profile
```

```
                                    .recently-used.xbel
                    .ICEauthority
[user1@server ~]$ vi .bash_profile _
```

add our directory */home/user1/custom_script* in PATH variable

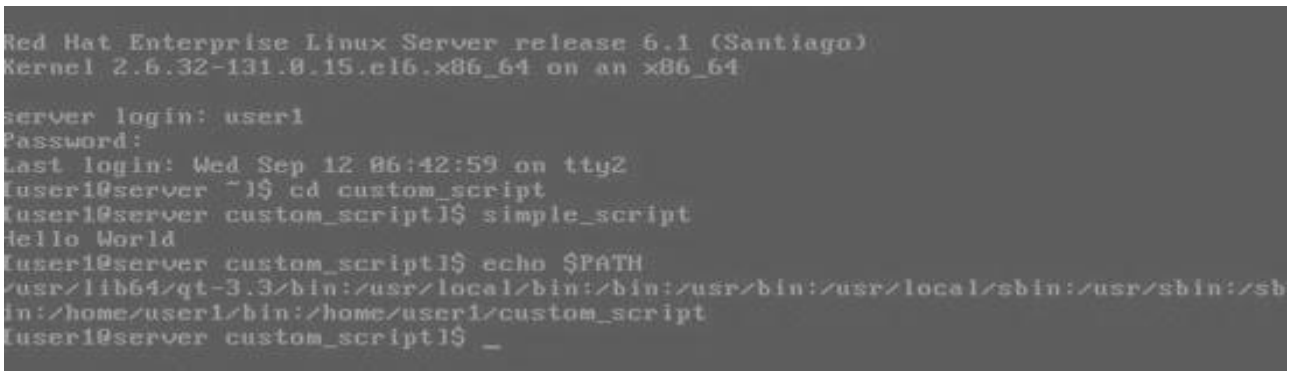


To update the variable, re-login required.

Logout

Login back

Now check that our custom path is available



Permanently set linux environmental variable Globally

root privilege requires to set linux environment variable globally. RHEL maintain and manage the environment variables in numerous files. But you do not need to pay attentions on all files that can contain environment variables. Following the RHEL recommendation

you should only set environmental variables in some particular files. The following files should be used for defining global environment variables on your system: /etc/profile, /etc/bash.bashrc and /etc/environment.

/etc/profile.d Directory is used to define global script.

Login from root and move to /etc

```
Red Hat Enterprise Linux Server release 6.1 (Santiago)
Kernel 2.6.32-131.0.15.el6.x86_64 on an x86_64

server login: root
Password:
Last login: Wed Sep 12 08:29:55 on tty2
[root@server ~]# cd /etc
[root@server etc]# _
```

move to */etc/profile.d* directory

Make a simple test script and make this script executable

```
[root@server etc]# cd profile.d
[root@server profile.d]# cat > global_script.sh
echo "Hello World Globally variable"
[root@server profile.d]# chmod +x global_script.sh
[root@server profile.d]# _
```

This script prints a simple welcome message for all users

To test it logout from root and login back from normal user

```
Red Hat Enterprise Linux Server 7.2 (Maipo)
Kernel 3.10.0-327.el7.x86_64 on an x86_64

localhost login: raj
Password:
Last login: Fri Jan 13 16:30:11 on tty1
[raj@localhost ~]$ pwd
/home/raj
[raj@localhost ~]$
```

We have successfully added global script. After testing to remove it login back from root

remove our test script

```
Red Hat Enterprise Linux Server release 6.1 (Santiago)
Kernel 2.6.32-131.0.15.el6.x86_64 on an x86_64

server login: root
Password:
Last login: Wed Sep 12 08:30:02 on tty2
Hello World Globally variable
[root@server ~]# rm /etc/profile.d/global_script.sh
rm: remove regular file `/etc/profile.d/global_script.sh'? y
[root@server ~]# _
```

List of Linux environmental variable

Variable name	Stored information
DISPLAY	used by the X Window system to identify the display server
DOMAIN	domain name
EDITOR	stores your favorite line editor
HISTSIZE	size of the shell history file in number of lines
HOME	path to your home directory

HOSTNAME	local host name
INPUTRC	location of definition file for input devices such as keyboard
LANG	preferred language
LD_LIBRARY_PATH	paths to search for libraries
LOGNAME	login name
MAIL	location of your incoming mail folder
MANPATH	paths to search for man pages
OS	string describing the operating system
OSTYPE	more information about version etc.
PAGER	used by programs like **man** which need to know what to do in case output is more than one terminal window.
PATH	search paths for commands
PS1	primary prompt
PS2	secondary prompt
PWD	present working directory
SHELL	current shell
TERM	terminal type
UID	user ID
USER(NAME)	user name
VISUAL	your favorite full-screen editor
XENVIRONMENT	location of your personal settings for X behavior
XFILESEARCHPATH	paths to search for graphical libraries

Linux alias command

In this article we would use alias command. Default aliases provide safety features. For a Linux system administrator, it is a handy tool.

How to check default alias

To check default alias run following command

```
$alias
```

```
[user1@server ~]$ alias
alias l.='ls -d .* --color=auto'
alias ll='ls -l --color=auto'
alias ls='ls --color=auto'
alias vi='vim'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-ti
lde'
[user1@server ~]$ _
```

Some of the aliases listed are likely to be system-wide aliases that apply to all users and are created automatically for each new user for a particular shell. Aliases for any other shell can be seen by first switching to that shell and than using the alias command as above.

alias command allows you to launch any command or group of commands with simple names or abbreviations.

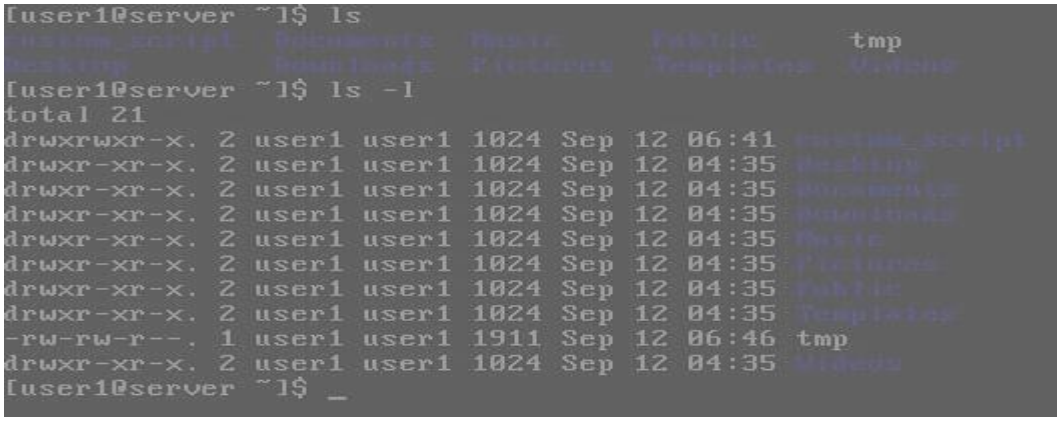
Syntax of alias is

```
alias name="value"
```

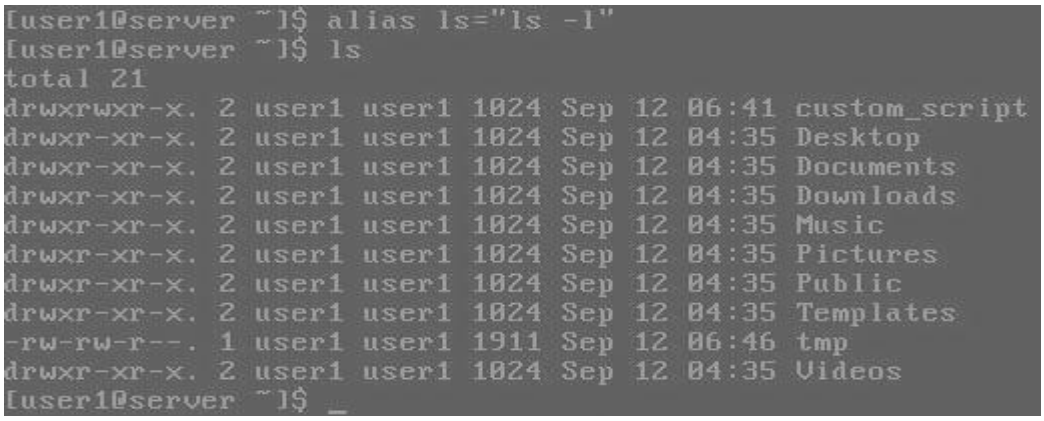
- name is the name of the new alias
- value is the command(s) which it initiates.
- No spaces are permitted before or after the equals sign. Any number of aliases can be created simultaneously by enclosing the name in each name-value pair in quotes.
- The alias name and the replacement text can contain any valid shell input except for the equals sign (=).
- The commands, including any options, arguments and redirection operators, are all enclosed within a single pair of quotation marks, which can be single quotes or double quotes.

Take a simple example of **ls** command. **ls** command lists the content of directory.

With **-l** switch it list content in long format with details

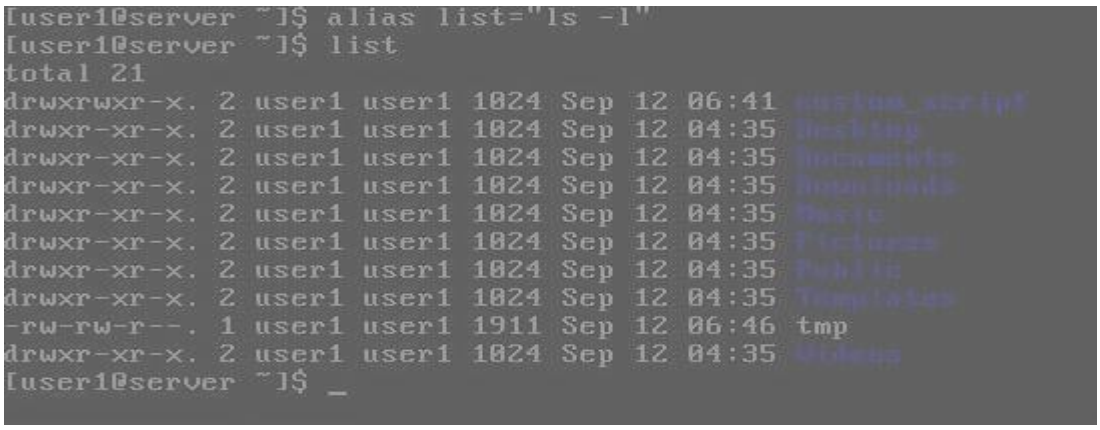


With the use of **alias** command, we can create an alias for **ls** command with **-l** switch so when you run **ls** command it execute with **-l** switch



You can use any simple easy to remember name instead of command and than use them in the same way that ordinary commands are used.

For example, you can use list keyword



alias set in this way are known temporary alias. Temporary alias would not be available after logout.

How to make alias permanent

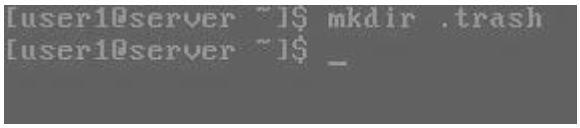
You can make alias permanent locally and globally.

To make an alias permanent on user level edit ***~/.bashrc*** file.

In linux command prompt any file or folder deleted once would be deleted forever. But we can make TRASH folder using alias command.

Make a ***trash*** folder. Keep it hidden. [Put a DOT in front of folder name]

```
$mkdir .trash
```

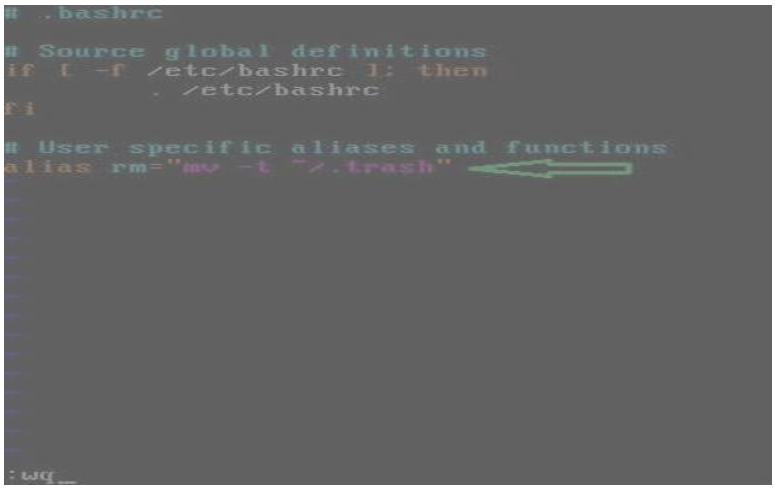


Open ***.bashrc*** file

In the end of file add following command

```
alias rm="mv -t ~/.trash"
```

- rm command used to delete file or folder [with switch]
- ~ In Linux ~ (tilde sign) represent users home directory



.bashrc file initialized when user login.

Logout from current session

login back

Now when a user run **rm** command shell will actually execute **mv** command. Create a test file and delete it with **rm** command

```
[user1@server ~]$ cat > test_file
This file contain important information
What if you delete this file accidently
Once deleted you can not restore it
Linux have no recyle bin
In Linux world once deleted delete forever
But we can custom recyle bin with help of alias command
[user1@server ~]$ rm test_file
[user1@server ~]$ ls
custom_script  Documents  Music     Public     tmp
Desktop        Downloads  Pictures  Templates  Videos
[user1@server ~]$ _
```

you can restore deleted file from trash folder. Restore our deleted file

```
[user1@server ~]$ cd .trash
[user1@server .trash]$ ls
test_file
[user1@server .trash]$ mv test_file ~
[user1@server .trash]$ cd
[user1@server ~]$ ls
custom_script  Documents  Music     Public     test_file  Videos
Desktop        Downloads  Pictures  Templates  tmp
[user1@server ~]$ cat test_file
This file contain important information
What if you delete this file accidently
Once deleted you can not restore it
Linux have no recyle bin
In Linux world once deleted delete forever
But we can custom recyle bin with help of alias command
[user1@server ~]$ _
```

To make an alias permanent on system level login from root and open ***/etc/bashrc***

```
Red Hat Enterprise Linux Server 7.0 (Maipo)
Kernel 3.10.0-123.el7.x86_64 on an x86_64

server86 login: root
Password:
Last login: Tue Apr 28 12:05:57 on tty3
[root@server86 ~]# vim  /etc/bashrc _
```

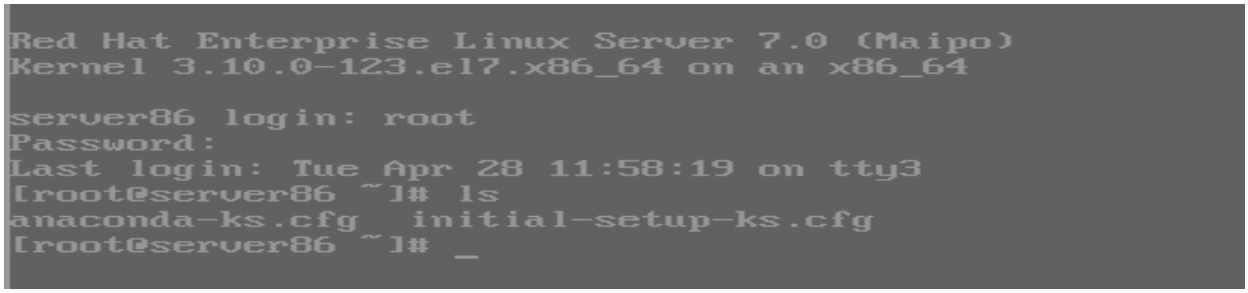
add your custom alias at the bottom of file and save it

```
    unset i
    unset pathmunge
fi
# vim:ts=4:sw=4

# System level custom alias
alias ls="ls -l"

:wq_
```

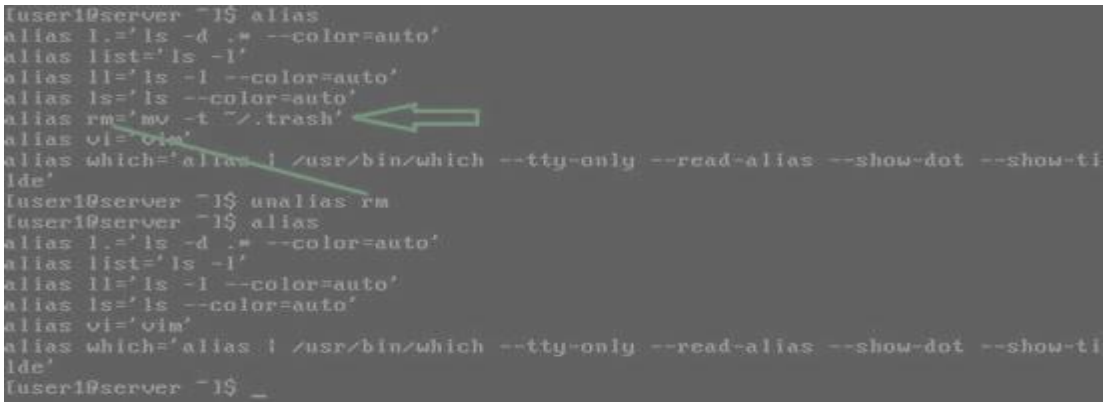
logout from current session login back and test alias



How to unset alias

If you need new alias with same name than best way to remove an alias is by use the alias command to create a new alias with the same name. This overwrites the existing alias with that name.

If you only want to remove alias use ***unalias*** command



If you have created permanent alias then open that file again and remove alias entry from configuration file.

Use of Linux alias command for system administrator

alias command can be used in several ways. Most popular use of alias command among the Linux system administrators are following

Use alias command to reduce the amount of typing

For example, you frequently need to go in directory which have long path. You can create an alias for that directory and use it. Like we have a directory with the path

```
~/custom_script/linux/web_script/php/new_script
```

we can create an alias for it

```
$ alias new_php_script="cd ~/custom_script/linux/web_script/php/new_script"
```

Now whenever we need to go in that directory, we only need to type new_php_script on command prompt

```
$new_php_script
```

Use alias command to specify default options for command.

Like we have specified for ls command in above

Use alias command to create trash on command prompt

As we create in above example

Use alias command for safety of the system

A system administrator use alias to increase the safety of the system by making commands interactive. This forces the user to confirm that it is desired to perform a specific action and thereby reduces the risk from accidental or impulsive abuse of powerful commands.

For example, **cp** command which is used to copy the contents of one file to another file, can also be reduced by making it interactive by default. If the name for the file to be written to does not exist in the specified directory (by default the current directory), it will be created, but if it already exists, its contents will be overwritten.

```
$alias cp="cp -i"
```

Above alias will reduce the chances of an unintended overwriting. Now if it detects any existing file with same name rather than overwriting that file shell would ask for confirmation.

Use alias to correct misspellings of commands.

For example, a user which switched from window platform, he has a habit of typing **dir** instead of **ls**. We can create an alias for it in following way

```
$alias dir="ls"
```

Now user can use dir also to list the content.

Sort command in linux

Sort command allows you to sort the content of file. With sort command you can sort the contents in several ways. By default, the sort command sorts the contents in alphabetical order depending on the first letter in each line.

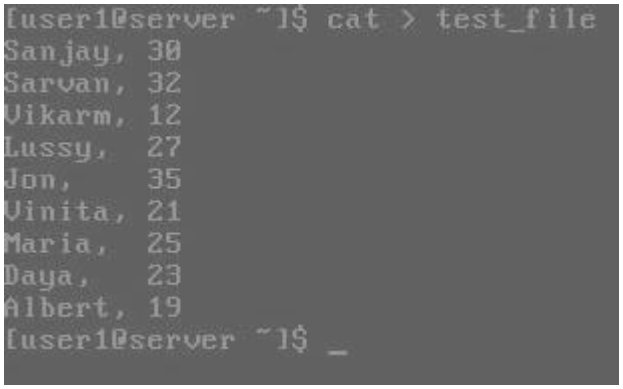
In this article we would cover following topic

- Example of sort command
- How to sort a file in alphabetical order in linux?
- How to sort by column in linux?
- How to sort in reverse order in linux?
- How to merge files with sort?
- How to sort files by size?
- List of options used with sort command

Example of sort command

Create a sample file with dummy names and age

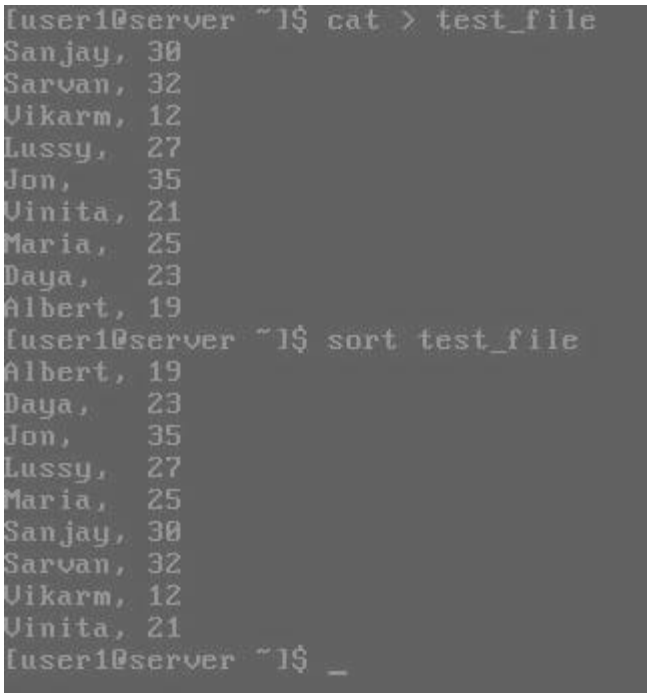
```
$cat > test_file
```



How to sort a file in alphabetical order in linux

To sort this file alphabetically depending on name run following command

```
$sort test_file
```

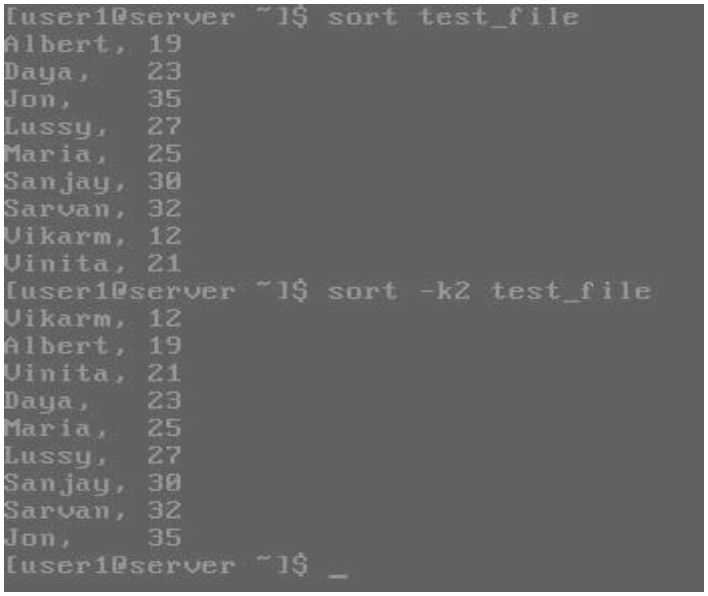


How to sort by column in linux

To sort this file depending on age run following command

```
$sort -k2 test_file
```

-k2 is the option which refers to the second column. You can specify other column also. Suppose that file contain 8 columns and your desired column number is 6 than you should use -k6.



How to sort in reverse order in linux

To sort in reverse order use -r option with sort command. You can also combine it with other options. To sort in reverse order of second column run following command

```
$sort -r -k2 test_file
```

```
[user1@server ~]$ sort -r -k2 test_file
Jon,      35
Jon,      35
Lussy,    34
Sarvan,   32
Sanjay,   30
Sanjay,   30
Lussy,    27
Maria,    25
Daya,     23
Vinita,   21
Vinita,   21
Albert,   19
Vikarm,   12
[user1@server ~]$ _
```

How to merge files with sort

```
[user1@server ~]$ cat test_file
Sanjay,   30
Sarvan,   32
Vikarm,   12
Lussy,    27
Jon,      35
Vinita,   21
Maria,    25
Daya,     23
Albert,   19
[user1@server ~]$ cat sample_file
Sodi,     45
Vakar,    34
Sachin,   35
Rani,     34
[user1@server ~]$ _
```

-m option allows us merge files in a single file.

```
[user1@server ~]$ sort -m sample_file test_file
Sanjay,   30
Sarvan,   32
Sodi,     45
Vakar,    34
Sachin,   35
Rani,     34
Vikarm,   12
Lussy,    27
Jon,      35
Vinita,   21
Maria,    25
Daya,     23
Albert,   19
[user1@server ~]$ _
```

How to save sort output in file?

By default, sort command will print out on standard output. Nothing is going to write in file. To save output in file either use -o option or use redirect.

```
[user1@server ~]$ cat demo_file
Sanjay,  30
Sarvan,  32
Sodi,    45
Vakar,   34
Sachin,  35
Rani,    34
Vikarm,  12
Lussy,   27
Jon,     35
Vinita,  21
Maria,   25
Daya,    23
Albert,  19
[user1@server ~]$ _
```

Use redirect method to save the output of sort command

```
[user1@server ~]$ sort -r demo_file > new_file
[user1@server ~]$ cat new_file
Vinita, 21
Vikarm, 12
Vakar,  34
Sodi,   45
Sarvan, 32
Sanjay, 30
Sachin, 35
Rani,   34
Maria,  25
Lussy,  27
Jon,    35
Daya,   23
Albert, 19
[user1@server ~]$ _
```

Use -o option to save the output of sort command in file

```
[user1@server ~]$ sort -r demo_file -o new_file1
[user1@server ~]$ cat new_file1
Vinita, 21
Vikarm, 12
Vakar,  34
Sodi,   45
Sarvan, 32
Sanjay, 30
Sachin, 35
Rani,   34
Maria,  25
Lussy,  27
Jon,    35
Daya,   23
Albert, 19
[user1@server ~]$ _
```

How to sort number in Linux

Use -n option to sort based on number. Create a simple file with numbers and use default sort order.

It is sorted alphabetically.

```
[user1@server ~]$ cat > number_file
5
56
7
4
6
78
34
12
567
[user1@server ~]$ sort number_file
12
34
4
5
56
567
6
7
78
[user1@server ~]$ _
```

To sort this based on number use -n option

```
[user1@server ~]$ sort -n number_file
4
5
6
7
12
34
56
78
567
[user1@server ~]$ _
```

How to sort files by size?

You can sort files by size with use of sort command. ls command is used to list the contents of directory.

```
[user1@server ~]$ ls -l
total 37
drwxrwxr-x. 2 user1 user1  1024 Sep 12 11:23 archived_file
-rw-rw-r--. 1 user1 user1    34 Sep 12 15:23 blank_space
drwxrwxr-x. 2 user1 user1  1024 Sep 12 11:52 cust_script
drwxr-xr-x. 2 user1 user1  1024 Sep 12 04:35 Desktop
drwxr-xr-x. 2 user1 user1  1024 Sep 12 04:35 Documents
drwxr-xr-x. 2 user1 user1  1024 Sep 12 04:35 Downloads
-rw-rw-r--. 1 user1 user1 10240 Sep 12 13:57 example.tar
drwxr-xr-x. 2 user1 user1  1024 Sep 12 04:35 Music
-rw-rw-r--. 1 user1 user1    24 Sep 12 15:10 number_file
drwxr-xr-x. 2 user1 user1  1024 Sep 12 04:35 Pictures
drwxr-xr-x. 2 user1 user1  1024 Sep 12 04:35 Public
drwxr-xr-x. 2 user1 user1  1024 Sep 12 04:35 Templates
-rw-rw-r--. 1 user1 user1   134 Sep 12 15:35 test_file
drwxr-xr-x. 2 user1 user1  1024 Sep 12 04:35 Videos
[user1@server ~]$ _
```

Use sort command with ls command to sort the files by size. As you can see in above image that size have column no 5. To sort the output of ls command based on size run following command

```
$sort ls -l | sort -k5
```

```
[user1@server ~]$ ls -l | sort -k5
total 37
-rw-rw-r--. 1 user1 user1 10240 Sep 12 13:57 example.tar
drwxr-xr-x. 2 user1 user1  1024 Sep 12 04:35 Desktop
drwxr-xr-x. 2 user1 user1  1024 Sep 12 04:35 Documents
drwxr-xr-x. 2 user1 user1  1024 Sep 12 04:35 Downloads
drwxr-xr-x. 2 user1 user1  1024 Sep 12 04:35 Music
drwxr-xr-x. 2 user1 user1  1024 Sep 12 04:35 Pictures
drwxr-xr-x. 2 user1 user1  1024 Sep 12 04:35 Public
drwxr-xr-x. 2 user1 user1  1024 Sep 12 04:35 Templates
drwxr-xr-x. 2 user1 user1  1024 Sep 12 04:35 Videos
drwxrwxr-x. 2 user1 user1  1024 Sep 12 11:23 archived_file
drwxrwxr-x. 2 user1 user1  1024 Sep 12 11:52 cust_script
-rw-rw-r--. 1 user1 user1   134 Sep 12 15:35 test_file
-rw-rw-r--. 1 user1 user1    24 Sep 12 15:10 number_file
-rw-rw-r--. 1 user1 user1    34 Sep 12 15:23 blank_space
[user1@server ~]$ _
```

List of options used with sort command

-r	Sorts in reverse order
-s	Stabilize sort by disabling last-resort comparison
-t	Use SEP instead of non-blank to blank transition

-u	If line is duplicated only display once
-b	Ignores blank spaces at beginning of the line.
-c	Check whether input is sorted or not.
-d	Use dictionary sort order and ignores the punctuation.
-f	Ignores caps
-k	Start a key at POS1, end it at POS2 (origin 1)
-m	Merges two or more input files into one file.
-M	Treats the first three letters in the line as a month (such as jun.)
-n	Sorts by the beginning of the number at the beginning of the line.
-o	Write result to FILE instead of standard output
-o outputfile	Save the sorted output to a file.

We have listed most frequently options used with sort command. To get a full list of all options with details read man page of sort command

```
$man sort
```

How to find files in linux

In Linux everything is file. Linux system is managed through the several configuration files. Most of configuration files have associated documentation file or sample file. You can use sample files in exam. For RHCE exam you should be able to find the file. It is very common to forget the path of file during the exam. You may know the name of file but not path in that case use these commands to find the file.

- find
- locate

find

find command need two arguments file name and location. Syntax of find command is

```
#find [location] -name  [file name ]
```

- **find**: - command
- **location**: - where you want to search the file
- **-name**: - option to specify the file name
- **file name**: - name of file which you want to search

For example to search **vsftpd.conf** [FTP configuration file] file we would use following command

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

This would start search from top level root directory and list the found.

```
[root@server ~]# find / -name vsftpd.conf
/etc/vsftpd/vsftpd.conf
/usr/share/doc/vsftpd-2.2.2/EXAMPLE/VIRTUAL_USERS/vsftpd.conf
/usr/share/doc/vsftpd-2.2.2/EXAMPLE/INTERNET_SITE_NOINETD/vsftpd.conf
/usr/share/doc/vsftpd-2.2.2/EXAMPLE/INTERNET_SITE/vsftpd.conf
[root@server ~]# _
```

Searching from root directory should be your last resources. When you perform search from root directory **find** command scan the entire Linux system for the desired file. It is time consuming process. Use subdirectories whenever you know it. For example if we know that vsftpd.conf file is located in **/etc** directory we should use following command

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

```
[root@server ~]# find / -name vsftpd.conf
/etc/vsftpd/vsftpd.conf
/usr/share/doc/vsftpd-2.2.2/EXAMPLE/VIRTUAL_USERS/vsftpd.conf
/usr/share/doc/vsftpd-2.2.2/EXAMPLE/INTERNET_SITE_NOINETD/vsftpd.conf
/usr/share/doc/vsftpd-2.2.2/EXAMPLE/INTERNET_SITE/vsftpd.conf
[root@server ~]# find /etc -name vsftpd.conf
/etc/vsftpd/vsftpd.conf
[root@server ~]# _
```

find command accepts wildcard. Wildcard allows us to find a file even we know only few characters of file name. For example our desired file starts from **vs** and have **.conf** in the end but we do not know the middle characters. In this case we would find it in following way

```
#find /etc -name vs*.conf
```

```
[root@server ~]# find /etc -name vsftpd.conf
/etc/vsftpd/vsftpd.conf
[root@server ~]# find /etc -name vs*.conf
/etc/vsftpd/vsftpd.conf
[root@server ~]#
```

Wildcards

*****	Any number of alphanumeric characters
?	Single alphanumeric characters

Example of wildcards

Create a directory and move in it

```
[root@server ~]# mkdir practice_of_find
[root@server ~]# cd practice_of_find/
[root@server practice_of_find]# _
```

Make some blank files for practice of **find** command. Use **touch** command to create files.

```
[root@server practice_of_find]# touch file1.conf
[root@server practice_of_find]# touch demo.conf
[root@server practice_of_find]# touch DeMo.conf
[root@server practice_of_find]# touch deMo.conf
[root@server practice_of_find]# touch fileK.conf
[root@server practice_of_find]# touch 234
[root@server practice_of_find]#
```

Find the files those start from **f** and end with **.conf**

It would return with following error

find: paths must precede expression

```
[root@server practice_of_find]# find /root/practice_of_find -name f*.conf
find: paths must precede expression: fileK.conf
Usage: find [-H] [-L] [-P] [-Olevel] [-D help|tree|search|stat|rates|opt|exec] [
path...] [expression]
[root@server practice_of_find]# _
```

find command expand the wild card while it parse. So if result contain single match it would return without any error. Like in above example we searched for **vs*.conf** and it returned with correct result. But if result contains more than one match it would return with **find: paths must precede expression** error. It is because what find parsing in this case will look like

```
#find /root/practices_of_find  -name file1.conf  filek.conf
```

how to solve find: paths must precede expression error

solution of **find: paths must precede expression** error is very simple. Put the file name in quotes. It would stop the shell (bash) expanding your wildcards.

```
[root@server practice_of_find]# find /root/practice_of_find -name "f*.conf"
/root/practice_of_find/file1.conf
/root/practice_of_find/fileK.conf
[root@server practice_of_find]# _
```

Find the files which

- have **file** in string
- later one character could be anything
- ends with **.conf**

```
[root@server practice_of_find]# find /root/practice_of_find -name "file?.conf"
/root/practice_of_find/file1.conf
/root/practice_of_find/fileK.conf
[root@server practice_of_find]# _
```

locate

find command is too time consuming specially in 2 hour RHCE exam. use **locate** command instead of find in exam. locate command use a database of installed files and directories. locate command database updated only once in a day.
Syntax of locate command is following

```
#locate [file name]
```

locate command search from its database so it does not require path.

```
[root@server practice_of_find]# locate vsftpd.conf
/etc/vsftpd/vsftpd.conf
/usr/share/doc/vsftpd-2.2.2/EXAMPLE/INTERNET_SITE/vsftpd.conf
/usr/share/doc/vsftpd-2.2.2/EXAMPLE/INTERNET_SITE_NOINETD/vsftpd.conf
/usr/share/doc/vsftpd-2.2.2/EXAMPLE/VIRTUAL_USERS/vsftpd.conf
/usr/share/man/man5/vsftpd.conf.5.gz
[root@server practice_of_find]# _
```

Major drawback of locate command is that it update its database only once in a day. For example you can find demo.conf which we created in above example from **find** command but not from **locate** command.

database of **locate** command is updated from **/etc/cron.daily/mlocate.cron** script. We can manually run this script.

```
[root@server practice_of_find]# find /root/practice_of_find -name demo.conf
/root/practice_of_find/demo.conf
[root@server practice_of_find]# locate demo.conf
[root@server practice_of_find]# /etc/cron.daily/mlocate.cron
[root@server practice_of_find]# _
```

Now we can find demo.conf also from **locate** command

```
[root@server practice_of_find]# find /root/practice_of_find -name demo.conf
/root/practice_of_find/demo.conf
[root@server practice_of_find]# locate demo.conf
[root@server practice_of_find]# /etc/cron.daily/mlocate.cron
[root@server practice_of_find]# locate demo.conf
/root/practice_of_find/demo.conf
[root@server practice_of_find]# _
```

In exam

- Update locate command database as soon as possible and use locate command whenever you need to search any file.
- Use find command when locate does not works. Try to specify as much path as you remember when using find command.

how to find difference between two files in linux

diff command finds the difference between files. In this article we would see how **diff** command can help us in RHCE exam.

Everything in Linux is managed through the several configuration files. During the exam you need to change the setting in several configuration files. You should always take backup before making any change in configuration file during the exam.

You can easily find out the changes which you have made if you have backup copy of configuration file.

Example of diff

We would modify **vsftpd.conf** file for practice.

ftp use **vsftpd.conf** configuration file for its setting. RHCE exam objective include ftp so you may have to configure it during the exam.

Create a directory and take backup of **vsftpd.conf** file

```
[root@server ~]# mkdir backup
[root@server ~]# cp /etc/vsftpd/vsftpd.conf /root/backup/
[root@server ~]# ls backup/
vsftpd.conf
[root@server ~]# _
```

Start the ftp service

```
[root@server ~]# chkconfig --list vsftpd
vsftpd          0:off   1:off   2:off   3:off   4:off   5:off   6:off
[root@server ~]# service vsftpd start
Starting vsftpd for vsftpd:                                [  OK  ]
[root@server ~]# service vsftpd status
vsftpd (pid 16245) is running...
[root@server ~]# _
```

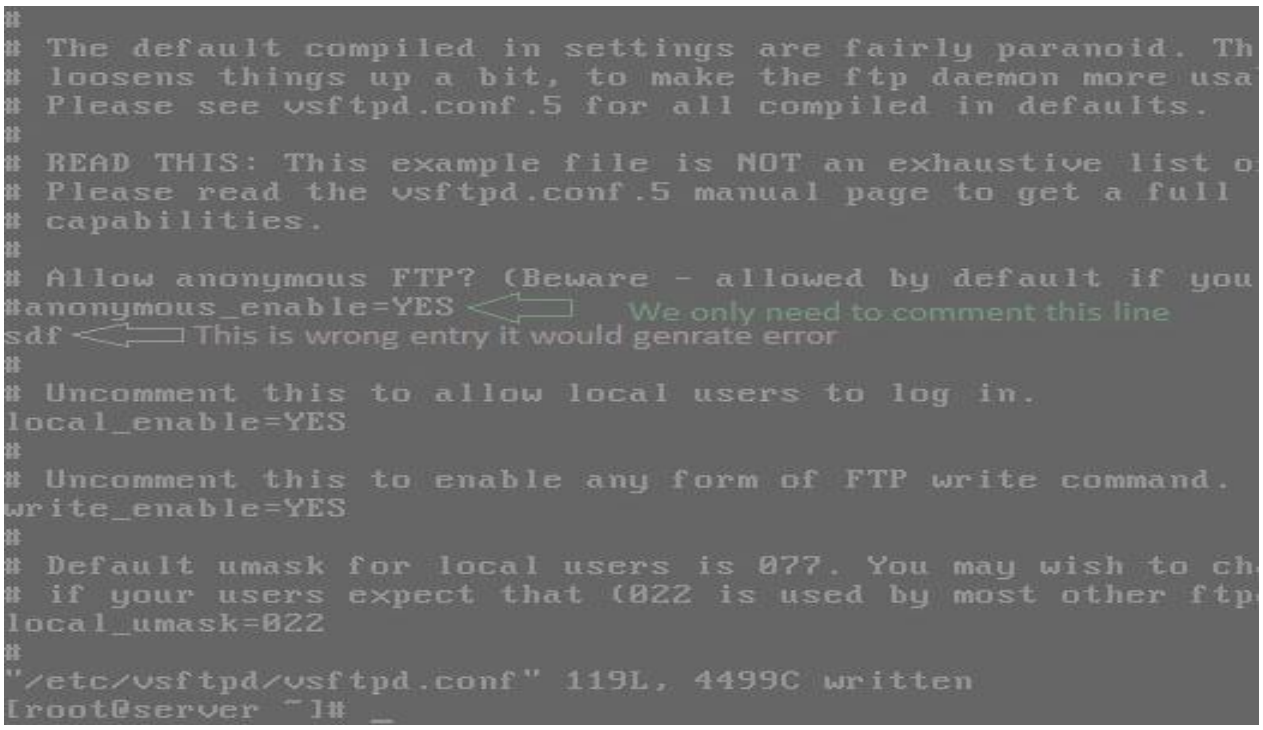
ftp by default allow anonymous login. Open vsftpd.conf

```
[root@server ~]# chkconfig --list vsftpd
vsftpd          0:off   1:off   2:off   3:off   4:off   5:off   6:off
[root@server ~]# service vsftpd start
Starting vsftpd for vsftpd:                                [  OK  ]
[root@server ~]# service vsftpd status
vsftpd (pid 16245) is running...
[root@server ~]# vi /etc/vsftpd/vsftpd.conf _
```

Your task is to disallow anonymous login.

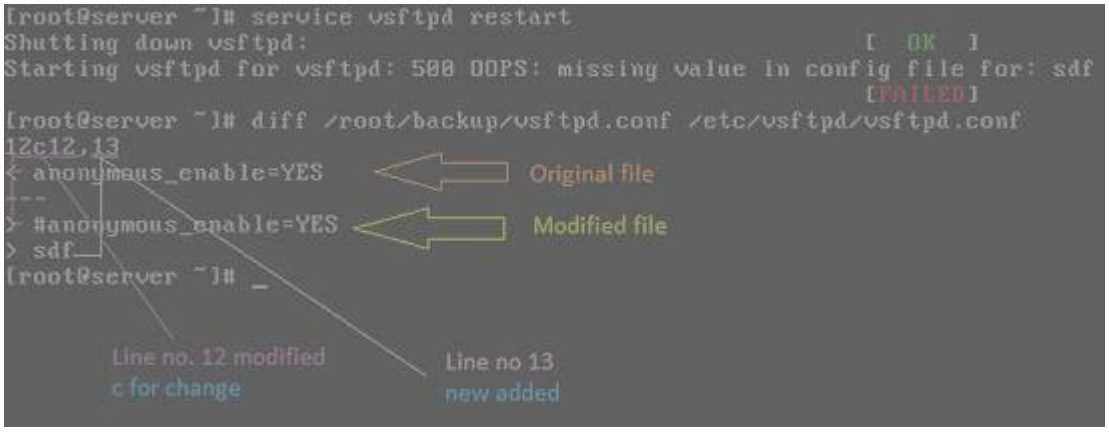
```
# Example config file /etc/vsftpd/vsftpd.conf
#
# The default compiled in settings are fairly paranoid.
# loosens things up a bit, to make the ftp daemon more
# Please see vsftpd.conf.5 for all compiled in defaults
#
# READ THIS: This example file is NOT an exhaustive lis
# Please read the vsftpd.conf.5 manual page to get a fu
# capabilities.
#
# Allow anonymous FTP? (Beware - allowed by default if
anonymous_enable=YES          Setting for anonymous login
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write comman
write_enable=YES
#
# Default umask for local users is 077. You may wish to
# if your users expect that (022 is used by most other
local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upl
"/etc/vsftpd/vsftpd.conf" 118L, 4494C
```

Change configuration setting so ftp disables anonymous login and save the file



Notice that I made wrong setting. It would stop vsftpd service. Try to restart vsftpd service.

vsftpd service failed with error. To troubleshoot this, we need to check what settings we did in configuration file. Here comes the magic of **diff** command. With **diff** command we can easily find the modified settings.



Now we know the modified settings. **diff** command also tells us the line number of modified settings.

Open vsftpd.conf file again

```
[root@server ~]# vi /etc/vsftpd/vsftpd.conf
```

Display line number (To display line number use ESC + : + set nu)

```
# Default umask for local users is 077
# if your users expect that (022 is us
local_umask=022
#
:set nu_ ⟹ Display line number
```

To fix the issue delete line no 13 [contain text sdf] and save the file.

```
 1 # Example config file /etc/vsftpd/vsftpd.con
 2 #
 3 # The default compiled in settings are fairl
 4 # loosens things up a bit, to make the ftp d
 5 # Please see vsftpd.conf.5 for all compiled
 6 #
 7 # READ THIS: This example file is NOT an exh
   ns.
 8 # Please read the vsftpd.conf.5 manual page
   's
 9 # capabilities.
10 #
11 # Allow anonymous FTP? (Beware - allowed by
   out).
12 #anonymous_enable=YES ⟹ Modified line
13 sdf ⟹ New added line
14 #          Delete this line and it would reslove the error
15 # Uncomment this to allow local users to log
16 local_enable=YES
17 #
18 # Uncomment this to enable any form of FTP w
19 write_enable=YES
20 #
:set nu
```

Now restart the vsftpd service again.

```
[root@server ~]# service vsftpd restart
Shutting down vsftpd:                                      [FAILED]
Starting vsftpd for vsftpd:                                [  OK  ]
[root@server ~]# _
```

Restore the original configuration file back after doing this practice.

```
[root@server ~]# cp /root/backup/vsftpd.conf /etc/vsftpd/vsftpd.conf
cp: overwrite `/etc/vsftpd/vsftpd.conf'? y
[root@server ~]# _
```

How to get help for commands

You can use command help options to get more details. During the RHCE exam no internet access is available. But you can use local resources available in RHEL. During the exam whenever you have doubt about the options used with command take help.

command itself

Run command itself without supplying any required options or arguments. For example, we do not know the correct syntax of **grep** command. In this case we should run **grep** command by itself. It would give us hint about the correct syntax of **grep** command.

```
[root@server ~]# grep
Usage: grep [OPTION]... PATTERN [FILE]...
Try `grep --help' for more information.
[root@server ~]# _
```

--help option

Running command by itself does not work for several commands. Like running **cp** command would not give us any information about command syntax or options.

```
[root@server ~]# cp
cp: missing file operand
Try `cp --help' for more information.
[root@server ~]# _
```

Use **--help** option with command to get available help.

```
[root@server ~]# cp
cp: missing file operand
Try `cp --help' for more information.
[root@server ~]# cp --help
```

info

Most of commands in RHEL have info manual. You should take help from info manual whenever it is available. info manual gives you detail about the associated options with short description. Syntax for **info** command is

```
#info [command]
```

```
[root@server ~]# info cp
```

man command

--help option list only most frequently used options. To list all available options with their details, use man page. Linux commands are documented in a format known as the man page. **man** command list the options and settings associated with command. To use **man** command use following syntax

```
$man [command]
```

For example, to get help about **ls** command

```
$man ls
```

```
[root@server ~]# man ls
```

During the exam sometime you may need help about any service or configuration value but do not know which man page is associated with that service. In such a situation, you can use **whatis** and **apropos** command. For example, we want to know information about ntfs but do not know which man page has documentation about ntfs. Run **apropos** command to list all man pages which have nfts in their description.

```
#apropos nfts
```

```
[root@server ~]# apropos ntfs
smbcquotas              (1)  - Set or get QUOTAs of NTFS 5 shares
vfs_acl_tdb             (8)  - Save NTFS-ACLs in a tdb file
vfs_acl_xattr           (8)  - Save NTFS-ACLs in Extended Attributes (EAs)
[root@server ~]# _
```

In same way to list all man page which has **ls** in their title run following command

```
#whatis ls
```

```
[root@server ~]# whatis ls
ls                    (1)  - list directory contents
ls                    (1p)  - list directory contents
[root@server ~]# _
```

Difference between **whatis** and **apropos** is that **whatis** look in title of man page while **apropos** search in description.

During the exam

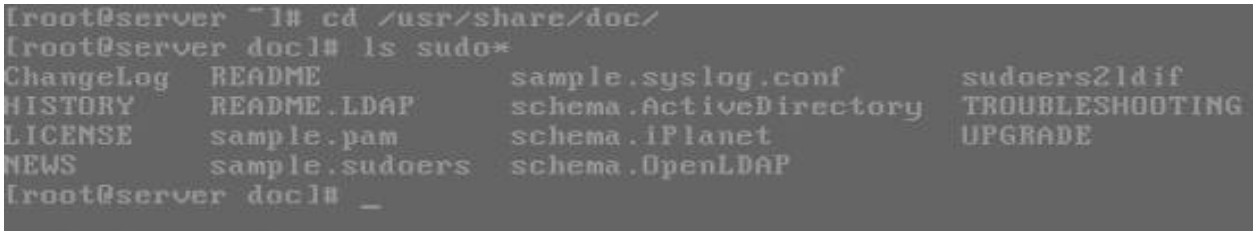
Both **whatis** and **apropos** command depends on a database in the **/var/cache/man** directory. So if you have installed any RPM during the exam these commands would not be for that. Fortunately, you can update the database at any time by running following command

```
# /etc/cron.daily/makewhatis.cron
```

Documentation directory

Path of documentation directory is **/usr/share/doc**. Documentation directory may include sample configuration files. So it is a better idea to check them as well. For example, **sudo-*/** directory contain following sample configuration files.

- sample.syslog.conf
- sample.pam
- sample.sudoers



2. Shell in linux

Before you can dive into working with the Linux command line and shells, it's a good idea to first understand what Linux shell is.

The shell

Linux shell is a special interactive utility. It provides a way for users to start programs, manage files on the filesystem, and manage processes running on the Linux system. The core of the shell is the command prompt. The command prompt is the interactive part of the shell. It allows you to enter text commands, interprets the commands, then executes the commands in the kernel.

The shell contains a set of internal commands that you use to control things such as copying files, moving files, renaming files, displaying the programs currently running on the system, and stopping programs running on the system. Besides the internal commands, the shell also allows you to enter the name of a program at the command prompt. The shell passes the program name off to the kernel to start it.

In RHEL 7 four shells available for users. User have their choice in command line interpreters.

Shell	Description
Bash	The default Bourne-Again shell, based on the command line interpreter originally developed by Stephen Bourne.
Dash	A simpler shell with fewer features than bash, but faster.
Tcsh	A shell that incorporates elements from the C programming language into shell scripts
Zsh	An advanced shell that incorporates features from bash, tcsh, and korn,

- These shells are configured in the /bin directory.
- Default shell in RHEL is bash.

How to access bash shell prompt in GUI

To execute commands, we need shell prompt on Linux. On Redhat system you can get shell prompt easily from a console and within the GUI. By default, when you access shell prompt on Redhat system you would likely to get bash [Bourne-Again shell] shell. However, many Linux administrators use one of the many other available shells but as I previously said in this section we are preparing for RHCE exam and use of the bash shell as it is specified in RHCSA objectives.

Access shell prompt from application menu

To access shell prompt from menu, click on application > click on system tools > and click on terminal

This would launch terminal

Access shell prompt from desktop

To access shell prompt from desktop right click on free space of desktop and click on Open in Terminal

This would launch terminal

Differences between accessing shell prompt from application menu and desktop

While you access terminal from application menu it would open terminal in user home directory. For example, if you are logged in from root user and access terminal from application menu you would get /root home directory of root user as default directory in terminal.

If you have opened terminal from desktop you would desktop as current directory in terminal.

How to access terminal in file system?

You can access terminal anywhere in file system of Linux. Right click on free space and click on open in terminal to access terminal in currently opened directory.

For example, to access terminal in /var directory

3. Grep & Pattern Matching

Grep is an acronym of Global Regular Expression Pattern. Grep command requires file or files name and pattern to search. It scans given file or files line by line and return the lines that contain the specified pattern.

This article enhances your knowledge on grep command which you have acquired from last article.

A regular expression is a way of specifying a pattern in text that can be applied to variable inputs to find all occurrences that match the pattern. A pattern is a sequence of characters.

Create a simple file and put some dummy data in it

how to find text string from grep command

RHCSA exam checks candidate's caliber to find text string from configuration or logs files. Beside exam as a Linux administrator you should know how to use grep to analyze text. In this article I am going to show how to find text string from files in rhel7.

how to find text string?

grep returns any lines that have characters, words, or expressions that match your query. we could use grep in two ways either independent or with any commands.

- To use grep independent run grep with -r switch.
- To use grep with other command use | pipe sign.

In Linux system configuration and log files could contain several thousand lines.

For example, we would like to find whether sshd service start on boot time or not, we would find sshd service status in /var/log/boot.log

```
[root@server ~]# grep -r sshd /var/log/boot.log
Starting sshd:                                             [  OK  ]
[root@server ~]# cat /var/log/boot.log | grep sshd
Starting sshd:                                             [  OK  ]
[root@server ~]# _
```

Take an another example, you would like to find the running process id of firefox use grep with ps command

```
[root@server ~]# ps -ef | grep firefox
root      2082  1896  0 03:22 tty3     00:00:00 grep firefox
[root@server ~]# _
```

Now find in regular text file, first create a test file and then find any string from file

```
[root@server ~]# cat > test_file
This is line one
This is second line
This is third line
This is fourth line
This is fifth line
[root@server ~]# grep -r third test_file
This is third line
[root@server ~]# cat test_file | grep third
This is third line
[root@server ~]# _
```

Finally find in configuration file, check whether userlist_enable (a configuration value which check the access) is enable or not

```
[root@server ~]# grep -r userlist_enable /etc/vsftpd/vsftpd.conf
userlist_enable=YES
[root@server ~]# _
```

Case insensitive grep search

search for *sanjay*

```
[root@server ~]# grep -r sanjay example_file
our main software engineer is sanjay
[root@server ~]# _
```

Default grep command search is case sensitive. To make case insensitive search use **-i** option

```
[root@server ~]# grep -r sanjay example_file
our main software engineer is sanjay
[root@server ~]# grep -i -r sanjay example_file
Sanjay software engineer   $4000
our main software engineer is sanjay
[root@server ~]# _
```

grep for words beginning and ending with

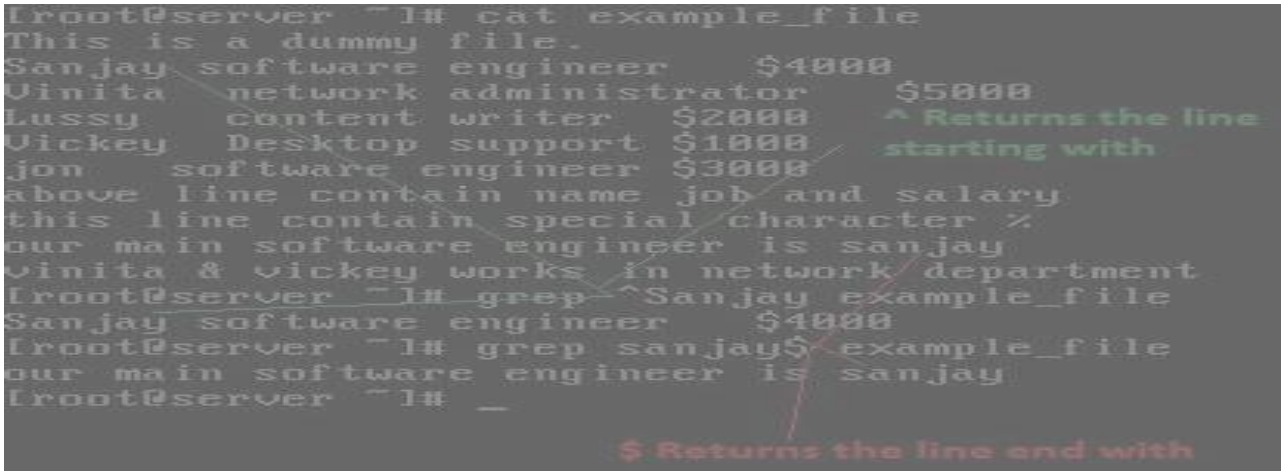
- Use **^** to match only at the start of line.
- Use **$** to match only at the end of line.

grep begins with

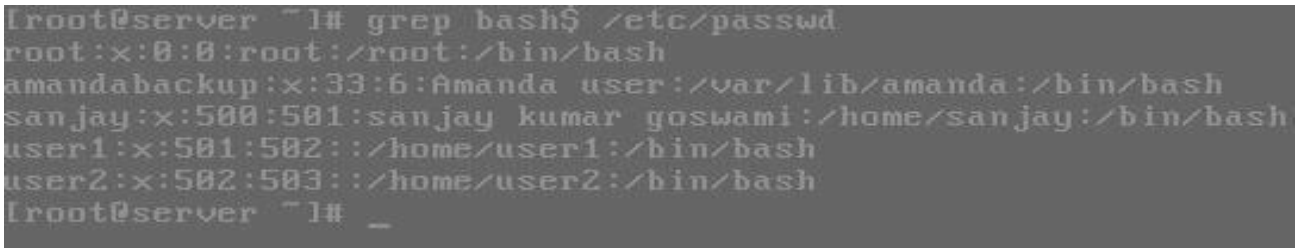
grep with ^ anchor would return all the line that start from given pattern. To match exacts word use **-w** option

```
[root@server ~]# cat > dummy_name
Sanjay
sanjay
sAnJay
sanjaykumar
SaNjAy
sanJay
[root@server ~]# grep ^sanjay dummy_name
sanjay
sanjaykumar
[root@server ~]# grep -w ^sanjay dummy_name
sanjay
[root@server ~]# _
```

grep ends with



Example question: - Find the users which use the bash shell.



-i option makes pattern to case insensitive. That is not useful when we want to make only certain characters' case insensitive from word.

In this case we can use pattern search with [ ]. Pattern search allow us to use any combination. There are already several ready to use combination which are known as character class. Character class is not going to be test in RHCE exam so we are not including it here. Check our Linux study guide for character classes.

For example, we want to search

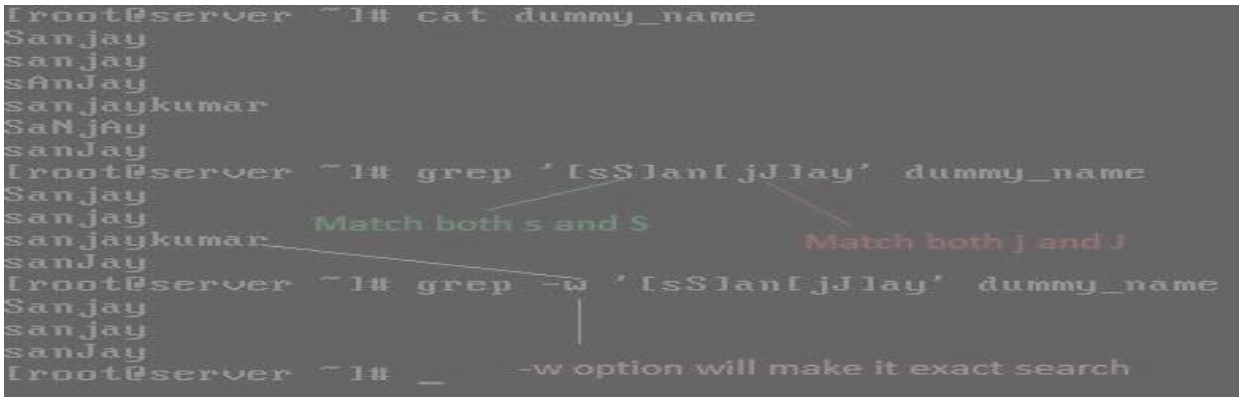
Both s and S

Both j and J

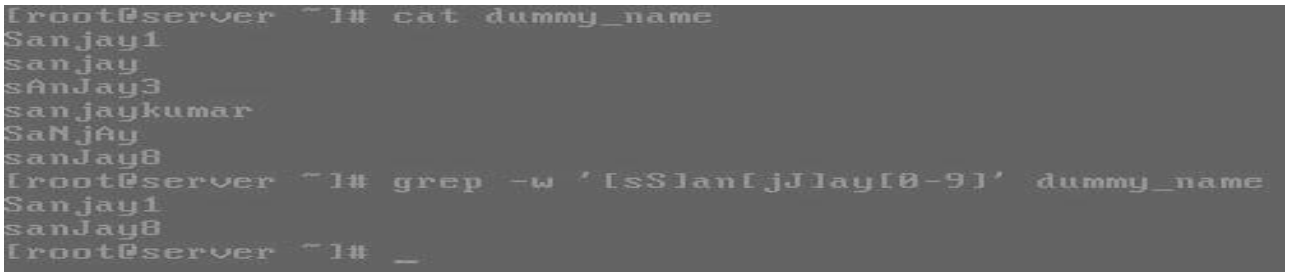
Than we would specify them in following manner

[sS] for both s and S

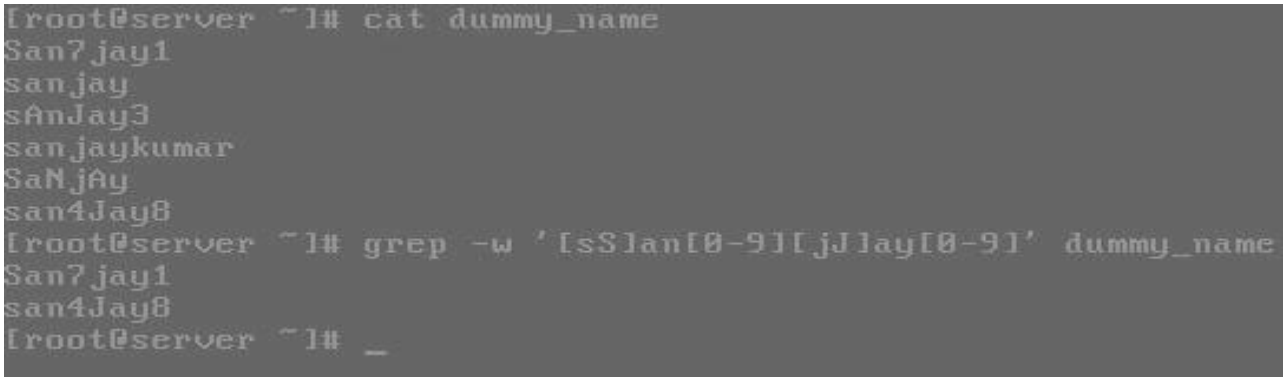
[jJ] for both j and J



pattern search allows us to use digit as well



Search for two digits



Match at least one letter

Match at least one digit

```
[root@server ~]# cat dummy_name
San?jay1
sanjay
sAnJay3
sanjaykumar
SaNjAy
san4Jay8
[root@server ~]# grep [0-9] dummy_name
San?jay1
sAnJay3
san4Jay8
[root@server ~]# _
```

Search for special character

Special characters need to be escaped. For example, to find **&** we need to use it **'\&'**

```
[root@server ~]# cat example_file
This is a dummy file.
Sanjay software engineer   $4000
Vinita  network administrator   $5000
Lussy   content writer  $2000
Vickey  Desktop support $1000
jon   software engineer $3000
above line contain name job and salary
this line contain special character %
our main software engineer is sanjay
vinita & vickey works in network department
[root@server ~]# grep '\&' example_file
vinita & vickey works in network department
[root@server ~]# _
```

Example question: - "/etc/hosts.allow" file contains the access rules for various network services. Check 192.168.1.23 has access or not.

```
[root@server ~]# cat /etc/hosts.allow
#
# hosts.allow   This file contains access rules which are used to
#               allow or deny connections to network services that
#               either use the tcp_wrappers library or that have been
#               started through a tcp_wrappers-enabled xinetd.
#
#               See 'man 5 hosts_options' and 'man 5 hosts_access'
#               for information on rule syntax.
#               See 'man tcpd' for information on tcp_wrappers
#
192.168.1.1
192.168.1.2
192.168.1.23

[root@server ~]# grep '192\.168\.1\.23' /etc/hosts.allow
192.168.1.23
[root@server ~]# _
```

egrep

With **grep** command meta characters [+, ?, |, (,)] loss their special meanings. For example, **+** have a special meaning "one or more times" but if we use it with grep command it would return the line which contains +. To use meta characters with their special meanings use **egrep** [acronym of **Extended Global Regular Expressions Pattern**] command.

```
[root@server ~]# cat test
this is test file
second line ?
third + four
5 ( means five )
[root@server ~]# grep "+" test
third + four
[root@server ~]# egrep "+" test
this is test file
second line ?
third + four
5 ( means five )
[root@server ~]#
```

fgrep

fgrep [acronym of "**Fixed-string Global Regular Expressions Pattern**"] does NOT recognize any regular expression meta-characters as being special. For example, if we want to search the line which contains (**.**) DOT. For grep command dot is a meta character that means 'wild-card, any single character'.

```
[root@server ~]# cat > test
this file contains special characters ?
what if i need to search . in line
[root@server ~]# grep '.' test
this file contains special characters ?
what if i need to search . in line
[root@server ~]# fgrep '.' test
what if i need to search . in line
```

pgrep

pgrep (acronym of **Process-ID Global Regular Expressions Pattern**) scan the currently running processes and lists the process IDs which matches the provided selection criteria. For example, if I want to know the process ID of my sshd process

```
[root@server ~]# pgrep sshd
1418
[root@server ~]#
```

This article is the part of our RHCE Study guide. It is a simplified version of grep command for the RHCE exam candidate. If you are a Linux administrator or need more details about grep command, please check man page.

4. Tar Command In Linux

tar command is used to create archive in Linux. In this article we would cover following topics

- What is tar
- tar file
- Syntax of tar command
- options used with tar command
- How to create tar file?
- How to compress tar file?
- How to extract tar file?
- linux tar command examples

What is tar

tar in linux stand for tape archive. tar was originally developed to write data to sequential I/O devices for backups on magnetic tape. tar is the file format and the name of program used to handle such files. Linux system administrator use tar to collect many files into one larger file for distribution or archiving, while preserving file system information like user and group permissions, dates, and directory structures etc.

tar file

tar file is an archive created from tar command. tar file is a single file that contains any number of individual files plus information to allow them to be restored to their original form. tar file also known as Tarballs. Tarballs are a common way to distribute Linux packages. Tarballs are normally distributed in a compressed format, with a.tar.gz or .tgz file extension, consolidated as a package in a single file.

Syntax of tar command

Syntax of tar command is

```
tar option(s) archive_name file_name(s)
```

- **tar** :- command
- **options** :- switch and parameters used with tar command
- **archive_name** :- name of archived file created from command
- **file_name** :- single or multiple files used to create tar archive file

options used with tar command

tar command requires at least one option to work. tar command has numerous options to use. Most of them are not frequently used. You can check all available options by reading manual page of tar. Run following command for the manual of tar

```
$man tar
```

Compulsory options

At least one option from following options required

Option	Description

-A	append tar files to an archive
-c	create a new archive
-d	find differences between archive and file system
--delete	delete from the archive
-r	append files to the end of an archive
-t	list the contents of an archive
-u	only append files that are newer than copy in archive
-x	extract files from an archive

Additional options

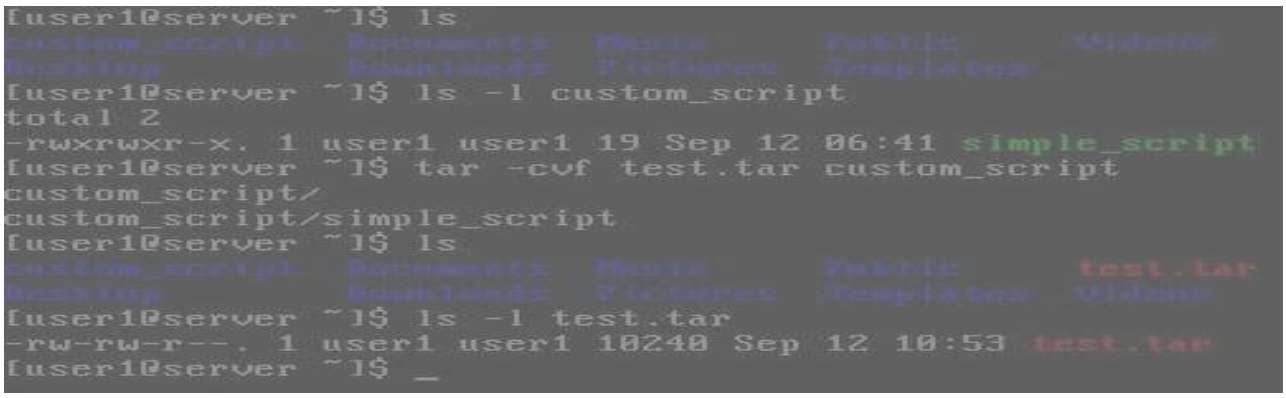
Use as per your requirement

Option	Description
-b	block size of Nx512 bytes (default N=20)
-C	change to directory DIR
-f	use archive file or device
-h	do not dump symlinks; dump the files they point to
-i	ignore blocks of zeros in archive
-j	filter the archive through bzip2
-k	keep existing files; don't overwrite them from archive
-l	stay in local file system when creating an archive
-m	don't extract file modified time
--p	preserve-permissions extract all protection information
-v	verbosely list files processed
-X	exclude files listed in FILE
-Z	filter the archive through compress
-z	filter the archive through gzip

How to create tar file

To create tar file run following command

```
$tar -cvf test.tar custom_folder
```



in above command

- **tar**:- command

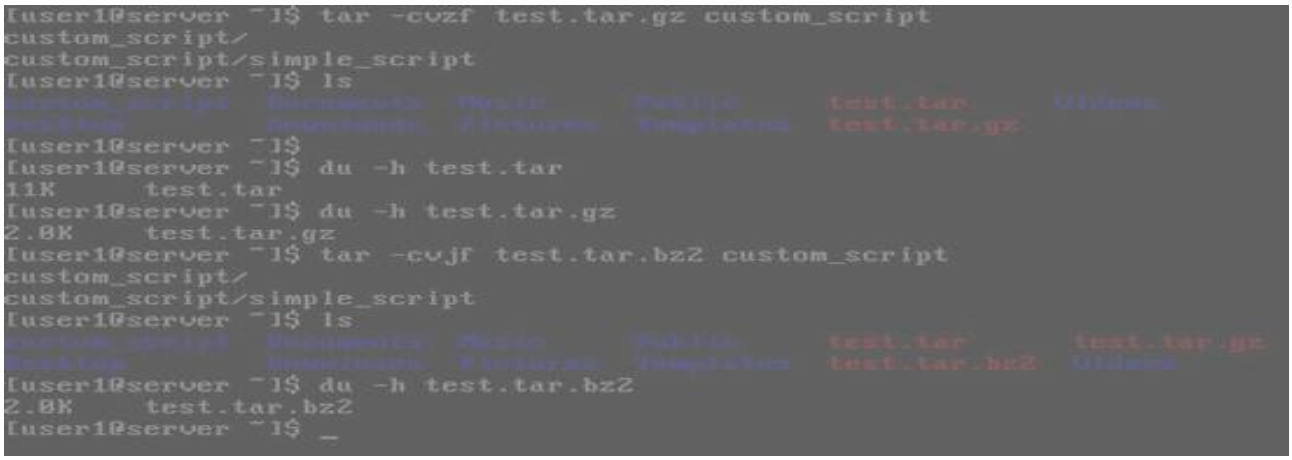
- **c**:- option to creates a new .tar archive file
- **v**:- option to display a list of the files that are included in the archive
- **f**:- option to specifytype of the archive file.You should always use -f option as the final option in sequence otherwise, the system will become confused as to the desired name for the new file and will use the next option in the sequence as the name.
- **test.tar**:- archived file name
- **custom_folder**:- folder which is going to archived

How to compress a tar file

tar command does not have compress or decompress features on its own. tar command is combined with external compression utility.

Option	Compression utility
-j	bzip2
-z	Gzip
-Z	compress

When you use any of above switch to combine compress utility with tar, compression utility compress the new archive file as soon as it has been created.



How to extract tar file

Following steps should be taken before extracting a tar file.

1. Check is tar file compressed or not? You can easily determine it by looking at the filename extension.
2. If the tar file has been compressed, it must first be decompressed using the appropriate decompression program.
3. Move tar file in an empty directory. Create new if you do not have one. It would prevent the reconstituted files from cluttering up the current directory and overwriting any files or directories with same names that are in it.
4. It is also advisable to check sufficient space available before unpacking tar file.

- Use **-x** option to extract
- Use **-v** option to display the list of files during the unpack process
- Use **-f** option to specify file name

Same options can be used to have the compression programs automatically decompress tar files prior to extraction. For example, to uncompress bz2 file use -j switch or to uncompress gzip file use -z switch.

In following example, we would decompress and extract the tar file compressed in above example

make a folder and move all three tar files

```
[user1@server ~]$ ls
[user1@server ~]$ mkdir archived_file
[user1@server ~]$ mv test.tar archived_file
[user1@server ~]$ mv test.tar.gz archived_file
[user1@server ~]$ mv test.tar.bz2 archived_file
[user1@server ~]$ cd archived_file
[user1@server archived_file]$ ls
```

Extract tar file

```
[user1@server archived_file]$ ls
[user1@server archived_file]$ mkdir tar
[user1@server archived_file]$ mv test.tar tar
[user1@server archived_file]$ cd tar
[user1@server tar]$ tar -xvf test.tar
custom_script/
custom_script/simple_script
[user1@server tar]$ ls
[user1@server tar]$ ls -l custom_script
total 2
-rwxrwxr-x. 1 user1 user1 19 Sep 12 06:41 simple_script
[user1@server tar]$ cd ..
[user1@server archived_file]$ _
```

Extract tar file compressed with bz2

```
[user1@server archived_file]$ ls
[user1@server archived_file]$ mkdir tar_with_bz2
[user1@server archived_file]$ mv test.tar.bz2 tar_with_bz2
[user1@server archived_file]$ cd tar_with_bz2
[user1@server tar_with_bz2]$ tar -xvjf test.tar.bz2
custom_script/
custom_script/simple_script
[user1@server tar_with_bz2]$ ls
[user1@server tar_with_bz2]$ ls -l custom_script
total 2
-rwxrwxr-x. 1 user1 user1 19 Sep 12 06:41 simple_script
[user1@server tar_with_bz2]$ cd ..
[user1@server archived_file]$ _
```

Extract tar file compress with gzip

```
[user1@server archived_file]$ ls
[user1@server archived_file]$ mkdir tar_with_gzip
[user1@server archived_file]$ mv test.tar.gz tar_with_gzip
[user1@server archived_file]$ cd tar_with_gzip
[user1@server tar_with_gzip]$ tar -xvzf test.tar.gz
custom_script/
custom_script/simple_script
[user1@server tar_with_gzip]$ ls
[user1@server tar_with_gzip]$ ls -l custom_script
total 2
-rwxrwxr-x. 1 user1 user1 19 Sep 12 06:41 simple_script
[user1@server tar_with_gzip]$ cd ..
[user1@server archived_file]$ _
```

So far in this article we have covered basic of tar with example. Now we will take some advance example of tar command. These examples of tar command helpful for beginner in linux. So let check how Linux system administrator use tar command with example

How to list files of tar tile

Some time you may want to check which files tar file contains without extracting tar file. Use - t option to list the content of tar file without extracting it. -t option works well with compressed tar files as well.

```
[user1@server archived_file]$ ls
test.tar  test.tar.bz2  test.tar.gz
[user1@server archived_file]$ tar -tvf test.tar
drwxrwxr-x user1/user1       0 2012-09-12 06:41 custom_script/
-rwxrwxr-x user1/user1      19 2012-09-12 06:41 custom_script/simple_script
[user1@server archived_file]$ tar -tvf test.tar.bz2
drwxrwxr-x user1/user1       0 2012-09-12 06:41 custom_script/
-rwxrwxr-x user1/user1      19 2012-09-12 06:41 custom_script/simple_script
[user1@server archived_file]$ tar -tvf test.tar.gz
drwxrwxr-x user1/user1       0 2012-09-12 06:41 custom_script/
-rwxrwxr-x user1/user1      19 2012-09-12 06:41 custom_script/simple_script
[user1@server archived_file]$ _
```

How to remove files after creating tar file

By default, tar creates an archive of copies of the original files and/or directories, and the originals are retained. To remove files / directories after creating tar file use --remove-files option.

```
[user1@server ~]$ ls
archived_file  Desktop    Downloads  Music     Public    Videos
custom_script  Documents  folder     Pictures  Templates
[user1@server ~]$ ls -l custom_script
total 6
-rwxrwxr-x. 1 user1 user1 19 Sep 12 11:32 demo_script
-rwxrwxr-x. 1 user1 user1 19 Sep 12 06:41 simple_script
-rwxrwxr-x. 1 user1 user1 19 Sep 12 11:32 test_script
[user1@server ~]$ tar --remove -cvf example.tar custom_script
custom_script/
custom_script/simple_script
custom_script/demo_script
custom_script/test_script
[user1@server ~]$ ls
archived_file  Documents  example.tar  Music     Public    Videos
Desktop        Downloads  folder       Pictures  Templates
[user1@server ~]$ _
```

How to extract single file from tar file

You can extract specific file from tar archive. File name is required for it. You can check file name from -t option as we did in above example. You need to specify full path of file.

To extract single file from tar archive

```
$tar -xvf  [Archived tar file name] [name of file ]
```

```
[user1@server ~]$ ls
[user1@server ~]$ tar -tvf example.tar
drwxrwxr-x user1/user1       0 2012-09-12 11:32 custom_script/
-rwxrwxr-x user1/user1      19 2012-09-12 06:41 custom_script/simple_script
-rwxrwxr-x user1/user1      19 2012-09-12 11:32 custom_script/demo_script
-rwxrwxr-x user1/user1      19 2012-09-12 11:32 custom_script/test_script
[user1@server ~]$ tar -xvf example.tar custom_script/simple_script
custom_script/simple_script
[user1@server ~]$ ls
[user1@server ~]$ ls -l custom_script
total 2
-rwxrwxr-x. 1 user1 user1 19 Sep 12 06:41 simple_script
[user1@server ~]$ _
```

If tar file compressed with bz2 use -j option while extracting

```
$tar -xvjf  [Archived tar file name] [name of file ]
```

If tar file compressed with gzip use -z option while extracting

```
$tar -xvzf [Archived tar file name] [name of file ]
```

How to extract multiple files from tar archive

In above example we extracted single file but you can extract multiple files from tar archive. You can specify multiple files separately

To extract multiple files from tar archive

```
$tar -xvf [Archived tar file name] ["name of file" ] ["name of file" ]
```

```
[user1@server ~]$ ls
[user1@server ~]$ tar -tvf example.tar
drwxrwxr-x user1/user1       0 2012-09-12 11:32 custom_script/
-rwxrwxr-x user1/user1      19 2012-09-12 06:41 custom_script/simple_script
-rwxrwxr-x user1/user1      19 2012-09-12 11:32 custom_script/demo_script
-rwxrwxr-x user1/user1      19 2012-09-12 11:32 custom_script/test_script
[user1@server ~]$ tar -xvf example.tar "custom_script/simple_script" "custom_scr
ipt/demo_script"
custom_script/simple_script
custom_script/demo_script
[user1@server ~]$ ls -l custom_script
total 4
-rwxrwxr-x. 1 user1 user1 19 Sep 12 11:32 demo_script
-rwxrwxr-x. 1 user1 user1 19 Sep 12 06:41 simple_script
[user1@server ~]$ _
```

If tar file compressed with bz2 use -j option while extracting

```
$tar -xvjf [Archived tar file name] ["name of file" ] ["name of file" ]
["name of file" ]
```

If tar file compressed with gzip use -z option while extracting

```
$tar -xvzf [Archived tar file name] ["name of file" ] ["name of file" ]
["name of file" ]
```

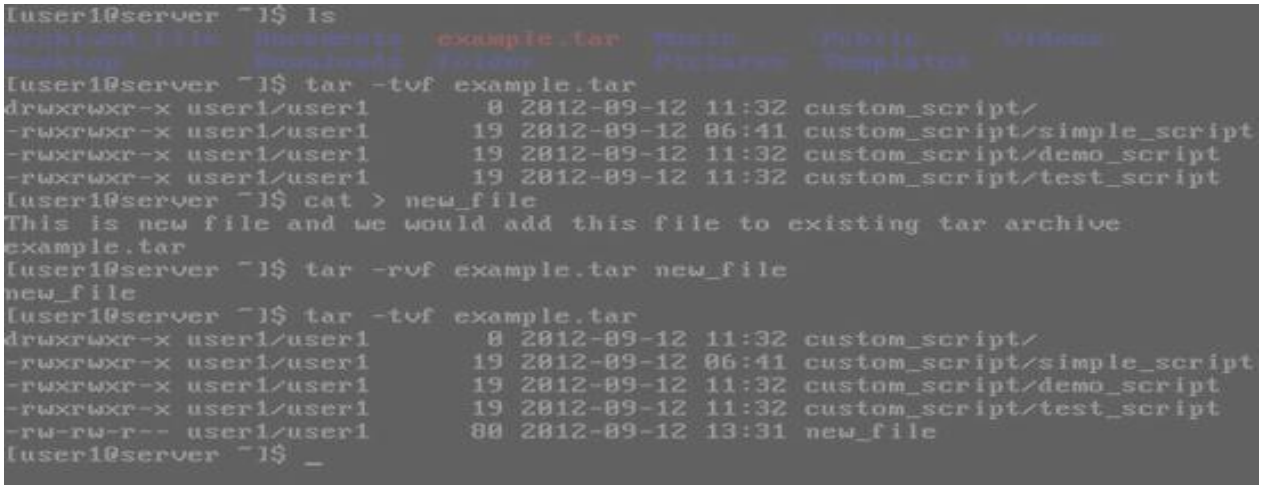
How to extract group files from tar file

You can use wildcard to extract group of files from tar file. Same as above if tar file is compressed you need to use same options [-j for bz2, -z for gzip] as well.

How to add new files in existing tar file

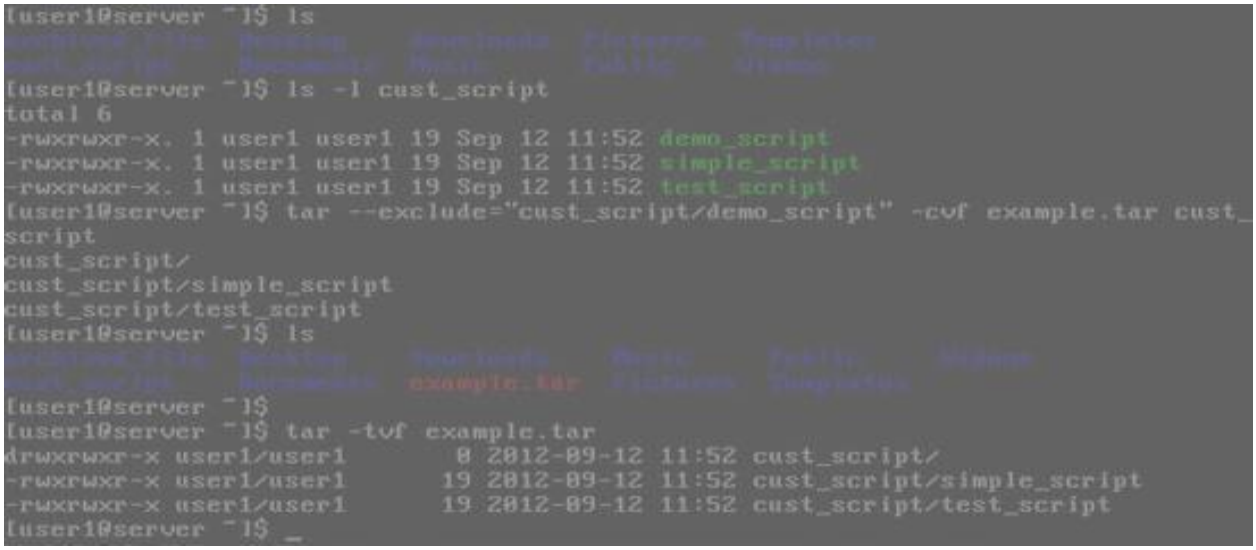
-r option is used to append the existing tar archive.

You can only append tar file if it is not compressed. You cannot append tar file if you have used -j, -z or -Z options during the creation process.



How to exclude files from tar

While creating tar archive some time you need to exclude certain files or directories. With --exclude option you can specify the file or directory



You can use multiple exclude options for tar like

```
$ tar --exclude='file1' --exclude='directory/file2' --exclude='patter*'
```

5. Use of awk & Sed in Linux

1. Print a Text File

'awk '{ print }' /etc/passwd'. here /etc/passwd is an input file

2. Use ":" as the input field separator and print first field only i.e. usernames

'awk -F: '{ print $1 }' /etc/passwd'

3. You can print more that one column from /etc/passwd

'awk -F: '{ print $1 }' /etc/passwd'

here first column of /etc/passwd will be printed.

Introduction to Sed

Sed has several commands, but most people only learn the substitute command

1. Use the command

i) $cat filename

one two three, one two three

four three two one

one hundred

"after using sed"

$sed 's/one/ONE/' filename

"The output would be"

ONE two three, one two three

four three two ONE

ONE hundred

As shown in example it will replace only first word of every line.

ii) To replace every occurence of word in each line use the command

'sed 's/loop/loop the loop/g' filename'

6. Compression Technique in Linux

1. Use gzip command to compress a regular file ie. Example

'[root@server18 ~]# gzip -v lines.txt'

lines.txt: 60.5% -- replaced with lines.txt.gz

It creates a file named "lines.txt.gz"

To uncompress the file "lines.txt.gz"

'[root@server18 ~]# gunzip -v lines.txt.gz '

lines.txt.gz: 60.5% -- replaced with lines.txt

Another compressor which is use to compress the file

2. Use bzip2 to compress any file.

'[root@server18 ~]# bzip2 -v lines.txt'

lines.txt: 1.914:1, 4.180 bits/byte, 47.74% saved, 266 in, 139 out.

To uncompress the same file as above

'[root@server18 ~]# bunzip2 -v lines.txt.bz2

lines.txt.bz2: done

7. USERADD

The useradd command creates a new user account. The new user account will be entered into the system files as below.

1. /etc/passwd
2. /etc/shadow
3. /etc/group
4. /etc/gshadow
5. /var/spool/mail
6. /home
7. Apply security to user's home directory
8. user add contain some user profile set directory and each directory contain some hidden files. these files are known as skeleton files.

location of skeleton file : /etc/skel/

1. **/etc/passwd** - when we create any new user, this file contains information about the newly added user. when we open this file there is some entries such as below.

Syntax: - #vim /etc/passwd

e.g. - Student:x:500:500::/home/Student:/bin/bash

each field have some meaning explain in below.

1. "Student" - this field represent the user name.
2. "x" - pointer or link to password to given user.
3. "500 "- user id, each and every user have their own unique id. Range of UID is 0-65535 and Root user have 0 UID.
4. "500" - Group id, it also has range between 0-65535.
5. ": : "- This field is use to add some string with your name it is known as comment field. Actual name of this field is GECOS. GECOS can store multiple information about any user. This information is known as finger database.

Syntax: - finger <username>

Description: - show the detailed information about user.

Syntax: - chfn <username>

Description: - Above syntax is used for add or update details for the already existing user.

6. "/home/Student": - location of user's home directory.
7. "/bin/bash" - path of shell for user.

Note: -

- Administrator account in Linux is known as root account it has UID 0.
- General account or shell account have UID between 500-60000.
- System accounts and service accounts have UID between 1-499 & 60001-65535. These acounts are never logged-in but they provide some services for the user.

2. /etc/shadow - file stores actual password in encrypted format for user's account with additional properties related to user password.

Syntax:- #vim /etc/shadow

e.g. student:1PIuwih7C$EsvYAfV4kgcPjYnR9iLVA1:15699:0:99999:7:::

1. “student: - It is your log in name.
2. “1PIuwih7C$EsvYAfV4kgcPjYnR9iLVA1” - It your encrypted password. The password should be minimum 6-8 characters long including special characters/digits. if it starts with "1" it means the MD5-based algorithm was used.
3. “15699” - Days since Jan 1, 1970 that password was last changed.
4. “0” - The minimum number of days required between password changes i.e. the number of days left before the user is allowed to change his/her password.
5. “99999” - The maximum number of days the password is valid (after that user is forced to change his/her password).
6. “7” - The number of days before password is to expire that user is warned that his/her password must be changed.
7. “::” - The number of days after password expires that account is disabled.
8. “::” - days since Jan 1, 1970 that account is disabled i.e. an absolute date specifying when the log in may no longer be used.
9. This field is reserved for future use.

3. /etc/group - file is world-readable and contains a list of groups, each on a separate line. Each line is a four field, colon delimited list including the following information.

Syntax: - #vim /etc/group

e.g. student:x:500:student,adhoc,rahul

“student” - Group Name. Used by various utility programs as a human-readable identifier for the group.

1. “x” - Group Password If set, this allows users that are not part of the group to join the group by using the newgrp command and typing the password stored here. If a lower case x is in this field, then shadow group passwords are being used.
2. “500” - Group ID (GID) The numerical equivalent of the group name. It is used by the operating system and applications when determining access privileges.
3. Last field is use to Member list, a comma delimited list of the users belonging to the group.

4. /etc/gshadow - This file is readable only by the root user and contains an encrypted password for each group, as well as group membership and administrator information.

Syntax: - #vim /etc/gshadow

e.g. student:!:rahul: student,Adhoc,rahul

1. “student” - The name of the group.
2. “**!**” - The encrypted password for the group. If set, non-members of the group can join the group by typing the password for that group using the newgrp command. If the value of this field is !, then no user is allowed to access the group using the newgrp command. A value of !! is treated the same as a value of ! — however, it also indicates that a password has never been set before. If the value is null, only group members can log into the group.

Syntax of set group password - #gpasswd <username>

3. Group Administrator listed here (in a comma delimited list) can add or remove group members using the gpasswd command.
4. Group members listed here (in a comma delimited list) are regular, non-administrative members of the group.

Some important Syntax: -

1. #useradd -G <user name> <Group name> “This syntax is use to add a user into already exiting group.”
2. #usermod -G <user name> <Group name> “This syntax is use to change group of existing user.
3. #userdel <user name> “This syntax is use to delete already exist user. It will delete only user. There home directory and mail directory will not be deleted.”
4. #userdel -r <username> “This syntax is also work like as above but it will delete home directory and mail directory of selected user.”
5. #groupadd <group name> “This syntax is use to add a new group.”

5 Useradd Create a Directory of Supplied Username Under /home Directory

6 Useradd Implements Some Security in User’s Home Directory

7 Useradd Creates a Mailbox for Same User Inside /var/spool/mail

8. Chmod permissions In linux

chmod command can be use to change different permission configurations. chmod takes two lists as its arguments: permission changes and filenames.

You can specify the list of permissions in two different ways. One way uses permission symbols and is referred to as the symbolic method. The other uses what is known as a “binary mask” and is referred to as either the absolute or the relative method.

Symbolic Method

The symbolic method of setting permissions uses the characters **r, w, and x** for read, write, and execute, respectively. Any of these permissions can be added or removed. The symbol to add a permission is the **plus sign, +**. The symbol to remove a permission is the **minus sign, -.**

chmod: - File Permissions in Symbolic Method

	Description
R	Read
W	Write
X	Execute (also gives permission to change into a directory)
X	Execute only if it is a directory or has execute permission for some user
S	Set user or group ID on execution
T	Sticky bit
U	Permissions granted to user who owns the file
G	Permissions granted to users in the file's group
O	Permissions granted to owner of the group and users in the file's group

r w x permissions

The first three (r, w, x) are clear. Use them to set read, write, and execute permissions.

s permission

The s permission is used on directories to keep the user or group ID for a file created in the directory. To set the user ID for any new files created in the directory to the owner of the directory, use the chmod u+s <directory> command. To set the group ID for any new files created in the directory to the directory's group, use the chmod g+s <directory> command.

t permission

t is a special permission which provides greater security on directories. Sticky bit is used for directories to protect files within them. Files in a directory with the sticky bit set can only be deleted or renamed by the root user or the owner of the directory.

u g o permission

The last three permissions (u, g, o) are only used with the = operator to set permissions for the owner, group, others, or everyone equal to the existing permissions for the owner, group, others, or everyone. For example, chmod g=u [filename] sets the group permissions to the current permissions for the owner of the file.

Examples of symbolic method

```
[root@localhost ~]# mkdir test
[root@localhost ~]# ls -ld test
drwxr-xr-x 2 root root 4096 Jan 23 03:01 test
[root@localhost ~]# chmod u+rwx test
[root@localhost ~]# ls -ld test
drwxr-xr-x 2 root root 4096 Jan 23 03:01 test
[root@localhost ~]# chmod g+rwx test
[root@localhost ~]# ls -ld test
drwxrwxr-x 2 root root 4096 Jan 23 03:01 test
[root@localhost ~]# chmod o+rwx test
[root@localhost ~]# ls -ld test
drwxrwxrwx 2 root root 4096 Jan 23 03:01 test
[root@localhost ~]# chmod o-rwx test
[root@localhost ~]# ls -ld test
drwxrwx--- 2 root root 4096 Jan 23 03:01 test
[root@localhost ~]# chmod g-rwx test
[root@localhost ~]# ls -ld test
drwx------ 2 root root 4096 Jan 23 03:01 test
[root@localhost ~]# _
```

Absolute Permissions: Binary Masks

The absolute method changes all the permissions at once, instead of specifying one or the other. It uses a binary mask that references all the permissions in each category.

Binary Masks

When dealing with a binary mask, you need to specify three digits for all three categories, as well as their permissions. This makes a binary mask less flexible than the permission symbols.

```
Digits permission
0            none
1            execute
2            write
4            read
3 (1+2)      write and execute
5 (1+4)      read and execute
7 (1+2+4)    read write execute
```

Value	Meaning
777	(rwxrwxrwx) No restrictions on permissions. Anybody may do anything. Generally not a desirable setting.
755	(rwxr-xr-x) The file's owner may read, write, and execute the file. All others may read and execute the file. This setting is common for programs that are used by all users.
700	(rwx——) The file's owner may read, write, and execute the file. Nobody else has any rights. This setting is useful for programs that only the owner may use and must be kept private from others.
666	(rw-rw-rw-) All users may read and write the file.
644	(rw-r–r–) The owner may read and write a file, while all others may only read the file. A common setting for data files that everybody may read, but only the owner may change.
600	(rw——-) The owner may read and write a file. All others have no rights. A common setting for data files that the owner wants to keep private.

Examples of binary masks

```
[root@localhost ~]# chmod 777 test
[root@localhost ~]# ls -ld test
drwxrwxrwx 2 root root 4096 Jan 23 03:01 test
[root@localhost ~]# chmod 755 test
[root@localhost ~]# ls -ld test
drwxr-xr-x 2 root root 4096 Jan 23 03:01 test
[root@localhost ~]# chmod 744 test
[root@localhost ~]# ls -ld test
drwxr--r-- 2 root root 4096 Jan 23 03:01 test
[root@localhost ~]# chmod 700 test
[root@localhost ~]# ls -ld test
drwx------ 2 root root 4096 Jan 23 03:01 test
[root@localhost ~]# chmod 775 test
[root@localhost ~]# ls -ld test
drwxrwxr-x 2 root root 4096 Jan 23 03:01 test
[root@localhost ~]# _
```

Permission	Owner	Group	Other	
Read	x	x	x	777
Write	x	x	x	
Execute	x	x	x	
Permission	Owner	Group	Other	
Read	x	x	x	755
Write	x			
Execute	x	x	x	
Permission	Owner	Group	Other	
Read	x	x	x	744
Write	x			
Execute	x			
Permission	Owner	Group	Other	
Read	x			700
Write	x			
Execute	x			
Permission	Owner	Group	Other	
Read	x	x	x	775
Write	x	x		
Execute	x	x	x	

Permission	Owner	Group	Other
Read - 4	x	x	
Write - 2	x	x	x
Execute - 1	x	x	x
	4+2+1=	4+2+1=	2+1=
	7	7	3

Defaults Permission: umask

Whenever you create a file or directory, it is given default permissions. You can display the current defaults or change them with the **umask** command. The permissions are displayed in binary or symbolic format. The default permissions include any execute permissions that are applied to a directory. Execute permission for a file is turned off by default when you create it because standard data files do not use the executable permissions (to make a file executable like a script, you have to manually set its execute permission). To display the current default permissions, use the umask command with no arguments.

The -S option uses the symbolic format.

```
#umask -S u=rwx,g=rx,o=rx
```

This default umask provides rw-r--r-- permission for standard files and adds execute permission for directories, rwxr-xr-x.

You can set a new default by specifying permissions in either symbolic or binary format. To specify the new permissions, use the -S option. The following example denies others read permission, while allowing user and group read access, which results in permissions of rwxr-x---:

```
#umask -S u=rwx,g=rx,o=
```

When you use the binary format, the mask is the inverse of the permissions you want to set. To set both the read and execute permission on and the write permission off, you use the octal number 2, a binary 010. To set all permissions on, you use an octal 0, a binary 000.

The following example shows the mask for the permission defaults rwx, rx, and rx (rw, r, and r for files):

```
#umask 0022
```

To set the default to only deny all permissions for others, you use 0027, using the binary mask 0111 for the other permissions.

```
#umask 0027
```

Change in umask from RHEL7

From RHEL7 no matter what the value of umask, new files can no longer be automatically created with executable permissions. For example, a umask value of 0454 leads to identical permissions on new files as a umask value of 0545. You need to use commands such as chmod to set executable permissions on a specific file.

Default value of umask is set in /etc/bashrc file.

Chmod Example

In our last article you learnt about permission. Permission can be set by chmod command in two different ways, symbolic and binary masks.

In this article we will practically implements whatever you have learnt so far in file permissions. This article is a sequential of last article if you have missed last article we suggest you to review them before going through this first.

Create 3 user a b c without password. Use for loop despite of creating them separately.

```
#for user in a b c
>do
>useradd $USER
>passwd -d $USER
>done
```

```
[root@localhost ~]# for USER in a b c
> do
> useradd $USER
> passwd -d $USER
> done
Removing password for user a.
passwd: Success
Removing password for user b.
passwd: Success
Removing password for user c.
passwd: Success
[root@localhost ~]# _
```

Now create a group example and add user a and b to it.

```
#groupadd example
#usermod -G example a
#usermod -G example b
```

```
[root@localhost ~]# groupadd example
[root@localhost ~]# usermod -G example a
[root@localhost ~]# usermod -G example b
[root@localhost ~]# _
```

now create a test directory on root partition and change ownership to user a and group to example.

```
[root@localhost ~]# mkdir /test
[root@localhost ~]# chown a /test
[root@localhost ~]# chgrp example /test
[root@localhost ~]# _
```

Now logon in 3 separate terminals from these users.

From root set permission to

```
#chmod 700 /test
```

```
[root@localhost ~]# chmod 700 /test
[root@localhost ~]# ls -ld /test
drwx------ 2 a example 4096 Jan 23 03:25 /test
[root@localhost ~]# _
```

This will set permissions to

```
owner a full
 group example ( a ,b ) none
 other c none
```

to verify these permission, go on the terminals where user a is logged on and run following commands

```
$cd /test
$cat > a_file
This is a file of user a
$ls a_file
```

```
localhost login: a
[a@localhost ~]$ cd /test
[a@localhost test]$ cat > a_file
this file is created by user a
[a@localhost test]$ ls
a_file
[a@localhost test]$ _
```

user a will be able to do all three task read write execute as owner have all three permissions Now try to change /test directory from user b. It will deny. Because user b remains in example group. and group have no permissions.

```
localhost login: b
[b@localhost ~]$ cd /test
-bash: cd: /test: Permission denied
[b@localhost ~]$ _
```

Now try to change /test directory from user c. it will also deny. Because user c is other for this directory and other have no permissions.

```
localhost login: c
[c@localhost ~]$ cd /test
-bash: cd: /test: Permission denied
[c@localhost ~]$ _
```

Now change permission from root to

```
#chmod 710 /test
```

```
[root@localhost ~]# chmod 710 /test
[root@localhost ~]# ls -ld /test
drwx--x--- 2 a example 4096 Jan 23 03:25 /test
[root@localhost ~]# _
```

This will give full permission to owner a. And execute to b (b is in the group of a which is example) User c (other) still have no permissions.

To verify, try change directory from user b to /test it would be success but he will not be able to list the contain of directory.

```
$cd /test $ls
```

```
[b@localhost test]$ ls
a_file
[b@localhost test]$ cat > b_file
-bash: b_file: Permission denied
[b@localhost test]$ _
```

Also verify the permission of c (other) by changing the directory to /test

```
$cd /test
```

```
localhost login: c
[c@localhost ~]$ cd /test
-bash: cd: /test: Permission denied
[c@localhost ~]$ _
```

Now change permission from root to

```
#chmod 751 /test
```

```
[root@localhost ~]# chmod 751 /test
[root@localhost ~]# ls -ld /test
drwxr-x--x 2 a example 4096 Jan 23 03:25 /test
[root@localhost ~]# _
```

This will give full permission to owner a. execute and read to b (b is in the group of a which is example) User c (other) now have execute permissions.

To verify try to list from user b to /test it would be success but he will not be able to write in directory.

```
$ls ; cat > b_file
```

```
[b@localhost test]$ ls
a_file
[b@localhost test]$ cat > b_file
-bash: b_file: Permission denied
[b@localhost test]$ _
```

Also verify the permission of c (other) by changing the directory to /test

```
$cd /test $ls
```

```
[c@localhost ~]$ cd /test
[c@localhost test]$ ls
ls: .: Permission denied
[c@localhost test]$ _
```

Now change permission from root to

```
#chmod 775 /test
```

```
[root@localhost ~]# chmod 775 /test
[root@localhost ~]# ls -ld /test
drwxrwxr-x 2 a example 4096 Jan 23 03:25 /test
[root@localhost ~]# _
```

This will give full permission to owner a b (b is in the group of a which is example) User c (other) now have read and execute permissions.

To verify try make new file from user b to /test it would be success.

```
$cd /test $ls $ cat > b_file This file is created by b
```

```
[b@localhost test]$ ls
a_file
[b@localhost test]$ cat > b_file
this is the file of b user
[b@localhost test]$ _
```

Also verify the permission of c (other) by listing the directory to /test

```
$cd /test $ls
```

```
[c@localhost test]$ ls
a_file  b_file
[c@localhost test]$ cat > c_file
-bash: c_file: Permission denied
[c@localhost test]$ _
```

Now change permission from root to

```
#chmod 777 /test
```

```
[root@localhost ~]# chmod 777 /test
[root@localhost ~]# ls -ld /test
drwxrwxrwx 2 a example 4096 Jan 23 03:25 /test
[root@localhost ~]# _
```

This will give full permission to owner a b and c. User c (other) now have full permissions.

To verify make file from user c

```
$ cat > c_file This file is created by user c
```

```
[c@localhost test]$ ls
a_file  b_file
[c@localhost test]$ cat > c_file
this file created by c user
[c@localhost test]$ _
```

how to set sticky bit

In our previous articles we have discussed about read write and execute permission for file and directory. Now I will show you some special permission which you can set for files and directories.

Sticky Bit Permissions

Sticky Bit is used for directories to protect files within them. Files in a directory with the sticky bit set can only be deleted or renamed by the root user or the owner of the directory.

Sticky Bit Permission Using Symbols

The sticky bit permission symbol is **t**. The sticky bit shows up as a t in the execute position of the other permissions. A program with read and execute permissions with the sticky bit has its permissions displayed as r-t.

```
# chmod +t /home/vinita/data
# ls -l /home/vinita/data -rwxr-xr-t 1 root root 4096 /home/vinita/data
```

Sticky Bit Permission Using the Binary Method

As with ownership, for sticky bit permissions, you add another octal number to the beginning of the octal digits. The octal digit for the sticky bit is 1 (001). The following example sets the sticky bit for the data directory:

```
# chmod 1755 /home/vinita/data
```

To remove sticky bit use minus sign.

```
#chmod o-t /example
```

```
[root@localhost ~]# chmod o-t /example
[root@localhost ~]# ls -ld /example
drwsrwsrwx 2 root root 4096 Jan 23 03:40 /example
[root@localhost ~]# _
```

now Vinita can delete the files owned by nikita verify

```
[vinita@localhost example]$ ls -l
total 8
-rw-rw-r-- 1 nikita root 6 Jan 23 03:36 nikita_file
-rw-rw-r-- 1 vinita root 5 Jan 23 03:36 vinita_file
[vinita@localhost example]$ rm nikita_file
rm: remove write-protected regular file `nikita_file'? y
[vinita@localhost example]$ rm vinita_file
[vinita@localhost example]$ ls
[vinita@localhost example]$ _
```

System Administrator (RHCSA)- SA2

Table of Contents

1. Using RPM

1. **Use the 'rpm -q' command to query the database of installed packages**

'rpm -q telnet'

2. **Instead of specifying the package name, you can use the following options with -q to specify the package(s) you want to query. These are called Package Specification Options.**

3. **list of all installed packages**

'rpm -qa'

4. **To install the package**

'rpm -ivh foo-1.0-1.i386.rpm

5. **Uninstalling a package**

'rpm -e foo'

6. **Upgrading a package**

'rpm -Uvh foo-2.0-1.i386.rpm

7. **Freshening a package is similar to upgrading one.**

'rpm -Fvh foo-1.2-1.i386.rpm'

2.How to configure yum server in Redhat linux

First of all, mount your cd/dvd some where into your system

mount /dev/cdrom1 /mnt/ (Here cdrom1 is my local cdrom device)

1. Go to the mount point of dvd/cd (example)

cd /mnt/RHEL7DVD

2. Go the location where packages are copied

cd Packages

3. # rpm -ivh deltarpm-3.5-0.5.20090913git.el6.i686.rpm

4. # rpm -ivh python-deltarpm-3.5-0.5.20090913git.el6.i686.rpm

5. # rpm -ivh createrepo-0.9.8-5.el6.noarch.rpm

6. # createrepo --database /mnt/RHEL7DVD/Packages

(location of path where rpm packages are listed)

7. # cp /mnt/RHEL7DVD/Packages /var/ftp/pub (For using Yum through vsftpd)

Client side configuration:

1. Create a repo file in your client system as mentioned above in the /etc/yum.repos.d/ . anyfile name but extension must be .repo as shown below

I) vim filename.repo

[id]

baseurl=ftp://192.168.56.101/pub/RHEL7DVD/Packages

gpgcheck=0

Above location "/pub/localyumserver"

NOTE: file DVD/CD is in the same system (client/server both are on the same machine then)

"baseurl=ftp://192.168.56.101/pub/RHEL7DVD/Packages" can be replaced by

"baseurl=file:///var/ftp/pub/RHEL7DVD/Packages"

3. Linux ACL Example

In our previous articles you learnt that how to set read, write, and execute permissions for the owner of the file, the group associated with the file, and for everyone else who has access to the filesystem. These files are visible with the **ls -l command.** These standard file permissions are all that an administrator needs to grant file privileges to users and to prevent unauthorized users from accessing important files.

However, when these basic file permissions are not enough, *access control lists,* or ACLs, can be used on linux file system. ACLs expand the basic read, write, and execute permissions to more categories of users and groups.

Before you start configuration of ACL, you need to enable ACL on filesystem. For testing we would implement ACL on /home partition. Check current status of ACL to confirm that the /home directory is mounted with the acl option, run the mount command alone, without switches or options

```
[root@server ~]# mount
/dev/sda2 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw,rootcontext="system_u:object_r:tmpfs_t:s0")
/dev/sda1 on /boot type ext4 (rw)
/dev/sda5 on /home type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
[root@server ~]# _
```

if directory is mounted with acl it would show in output. As output show acl is not configured on directory. So first we need to remount this directory with ACL

Remount partition with ACL use following command to remount partition with ACL

```
[root@server ~]# mount -o remount -o acl /dev/sda5 /home
[root@server ~]# mount
/dev/sda2 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw,rootcontext="system_u:object_r:tmpfs_t:s0")
/dev/sda1 on /boot type ext4 (rw)
/dev/sda5 on /home type ext4 (rw,acl)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
```

To make sure this is the way /home is mounted on the next reboot, edit /etc/fstab.

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

locate the partition entry and add acl keyword just after the default keyword separated with a comma and save the file.

```
#
# /etc/fstab
# Created by anaconda on Sat Aug 11 19:06:28 2012
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=078a11a3-3070-43d5-b4a4-0ff99f184bc4 /                       ext4    defaul
ts          1 1
UUID=56f99ced-0da2-4244-9976-42f61212ceca /boot                   ext4    defaul
ts          1 2
UUID=55a76f14-28b4-43eb-8d0a-4b216a246646 /home                   ext4    defaul
ts,acl      1 2
UUID=1a2769c8-a5ac-480d-8f21-1b8b99670392 swap                    swap    defaul
ts          0 0
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


:wq_
```

Once the change is made to /etc/fstab, you can activate it with the following command:

```
# mount -o remount /home
```

Or you could reboot the system. after reboot run mount command to check the status of acl

```
[root@server ~]# mount
/dev/sda2 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw,rootcontext="system_u:object_r:tmpfs_t:s0")
/dev/sda1 on /boot type ext4 (rw)
/dev/sda5 on /home type ext4 (rw,acl)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
[root@server ~]# _
```

Now you can start working with ACL commands to set secondary access controls on desired files and directories.

In addition to permissions for the owner and group for the file, ACLs allow for permissions to be set for any user, any user group, and the group of all users not in the group for the user.

Consider a situation where you want to grant write permission only to two users from a group of ten users. If you set permission from chmod all other users from group will get write access on file. In such a situation ACLs works.

Categories of ACLs

There are four categories of ACLs per file:

- For an individual user,
- For a user group,
- Via the effective rights mask
- For users not in the user group associated with the file.

To view the existing ACLs for a file, execute the following:

```
getfacl <file>
```

If ACLs are enabled, the output should look similar

```
# file: accounts # owner: Shweta # group: Shweta user::rwx group::r-x
mask::rwx other::---
```

```
[Shweta@localhost example]$ getfacl accounts
# file: accounts
# owner: Shweta
# group: Shweta
user::rwx
user:Shweta:rwx
user:Vinita:rwx
group::rwx
mask::rwx
other::---

[Shweta@localhost example]$ _
```

To understand acl more clearly let's take a simple example of acl.

Create three users named Shweta Vinita and Niddhi

```
#for USER in Shweta Vinita Niddhi
> do
>useradd $USER
>passwd -d $USER
>done
```

```
[root@localhost ~]# for USER in Shweta Vinita Niddhi
> do
> useradd $USER
> passwd -d $USER
> done
Removing password for user Shweta.
passwd: Success
Removing password for user Vinita.
passwd: Success
Removing password for user Niddhi.
passwd: Success
[root@localhost ~]# _
```

Now make them the member of goswami groups

```
#groupadd goswami
#usermod -G goswami Shweta
#usermod -G goswami Vinita
#usermod -G goswami Niddhi
```

```
[root@localhost ~]# groupadd goswami
[root@localhost ~]# usermod -G goswami Shweta
[root@localhost ~]# usermod -G goswami Vinita
[root@localhost ~]# usermod -G goswami Niddhi
[root@localhost ~]# _
```

Now create a /example directory and change the ownership to Shweta

```
#mkdir /example
#chown Shweta /example
```

```
[root@localhost ~]# mkdir /example
[root@localhost ~]# chown Shweta /example
```

Now logon form Shweta on other terminals and create a folder

```
$cd /example
$mkdir /accounts
```

```
localhost login: Shweta
[Shweta@localhost ~]$ cd /example
[Shweta@localhost example]$ mkdir accounts
[Shweta@localhost example]$ _
```

Now Shweta want to grant write permission only to Vinita. Niddhi will also get writes access on directory if Shewta sets write permission on groups as she is also the member of goswami group. So Shweta will use acl to grant write access to Vinita.

```
$setfacl -m u:Shweta:rwx accounts
$setfacl -m u:Vinita:rwx accounts
$setfacl -m other:--- accounts
$getfacl accounts
```

```
[Shweta@localhost example]$ setfacl -m u:Vinita:rwx accounts
[Shweta@localhost example]$ setfacl -m other:--- accounts
[Shweta@localhost example]$ _
```

To verify execute getfacl commands on accounts folder

```
[Shweta@localhost example]$ getfacl accounts
# file: accounts
# owner: Shweta
# group: Shweta
user::rwx
user:Shweta:rwx
user:Vinita:rwx
group::rwx
mask::rwx
other::---

[Shweta@localhost example]$ _
```

As in output you can see that user Shweta and Vinita have full permission over accounts folder. All other user except Shweta and Vinita have no permission over accounts folder. To verify this acl login form Vinita on other terminal and change directory to example.

```
localhost login: Vinita
Last login: Sat Jan 23 04:50:17 on tty3
[Vinita@localhost ~]$ cd /example
[Vinita@localhost example]$ _
```

Now make a test directory in account folder it should be successful as Vinita user have full permission over account folder.

```
[Vinita@localhost example]$ ls
accounts
[Vinita@localhost example]$ cd accounts/
[Vinita@localhost accounts]$ mkdir test
[Vinita@localhost accounts]$ ls
test
[Vinita@localhost accounts]$ _
```

Now go other terminals and login form user Niddhi and change directory to example

```
localhost login: Niddhi
[Niddhi@localhost ~]$ cd /example/
[Niddhi@localhost example]$ _
```

Try to change directory to account she will be denied as she has no permission over accounts

```
[Niddhi@localhost example]$ ls
accounts
[Niddhi@localhost example]$ cd accounts/
-bash: cd: accounts/: Permission denied
[Niddhi@localhost example]$ _
```

4.Security Enhanced Linux SELinux

SELinux was Developed by the National Security Agency (NSA), it adds protection for different files, applications, processes, and so on. On the Red Hat exams, you are expected to work with SELinux.

The first objective is fundamental to SELinux

Set enforcing/permissive modes for SELinux

The next objective requires that you understand the SELinux contexts defined for different files and processes.

List and identify SELinux file and process contexts

The next objective require that you are able to restore the default file contexts

Restore default file contexts

The last objective require that you configure boolean setting.

Use boolean settings to modify system SELinux settings

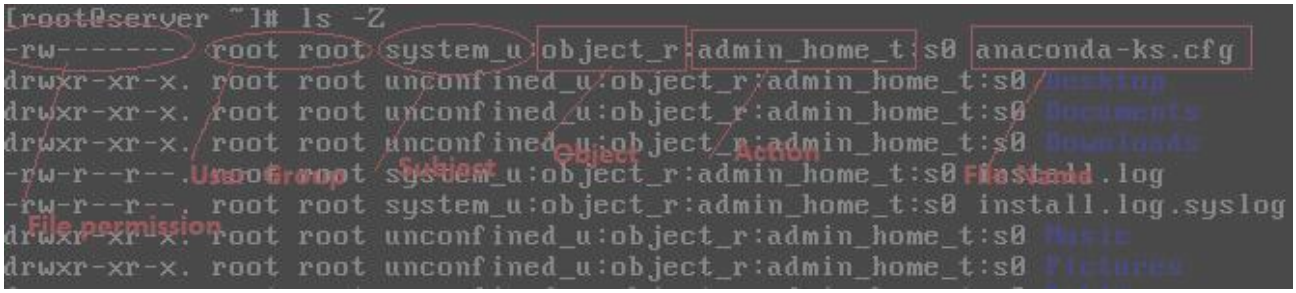
In this article we would start from the fundamental of SELinux.

Understanding SELinux

SELinux can be quite complex. So we would start from basic. Before you start working with SELinux you should understated the terminology used in SELinux. Let's start with some of the basics concept:

- **subject :-** subject is a command, process or application witch want to access any linux file.
- **object :-** object is a linux file or services.
- **action :-** an action is what may be done by the subject to the object.

Each file, folder, and service has an associated label that contains all three contexts.



File Contexts :- SELinux uses four different contexts to enforce security:

- user[take it as subject]
- role[understand it as object]
- domain (also known type, this is action)
- level (new from RHEL7 this level represents the sensitivity level of a file or directory).

for contexts you could use more restrictive values but for RHCE exam you should only focus on following contexts values.

Important context values for RHCE Exam

Contexts	Values	Description
User:	unconfined_u	Unprotected user
	system_u	System user
	user_u	Normal user
Role:	object_r	File
	system_r	Users and processes
Domain:	unconfined_r	Unprotected file or process

Take an example of sshd service check the SELinux labels

- The first field you see here is system_u, which, you can tell from the table, is a system user.
- The second field contains system_r, which again you can reference to see that it is a user or, in this case, a process.
- The third field shows sshd_t as the domain.

The domain is simply a way of categorizing which contexts can do to one another. Let's take an another example of domain context

```
[root@server ~]# ll -Z /etc/ssh/sshd_config
-rw-------. root root system_u:object_r:etc_t:s0       /etc/ssh/sshd_config
[root@server ~]# _
```

From output you could see

```
user[subject]       system_u (a system user)
role[object]        object_r(a file)
domain[action]      etc_t
```

Any service that has access to the etc_t domain is able to access this file. Beside root only system services have access to the /etc directory, so a domain of etc_t makes sense.

Now you have basic understanding of SELinux context.

SELinux commands

In this article we would discuss SELinux commands. Although there are several commands for SELinux but in this article we would only focus on those commands which are required in RHCE Exam.

sestatus

Shows the current status of SELinux

```
[root@server ~]# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /selinux
Current mode:                   enforcing
Mode from config file:          enforcing
Policy version:                 24
Policy from config file:        targeted
[root@server ~]# _
```

Options:

```
-b Displays all Booleans and their statuses
-v Provides verbose output
```

getenforce

Shows the enforcing status of SELinux

```
[root@server ~]# getenforce
Enforcing
[root@server ~]# _
```

setenforce

Changes the enforcing status of SELinux

```
[root@server ~]# getenforce
Permissive
[root@server ~]# setenforce Enforcing
[root@server ~]# getenforce
Enforcing
[root@server ~]# _
```

getsebool

Returns the Boolean value of a service option

```
[root@server ~]# getsebool -a | grep ftp
allow_ftpd_anon_write --> off
allow_ftpd_full_access --> off
allow_ftpd_use_cifs --> off
allow_ftpd_use_nfs --> off
ftp_home_dir --> off
ftpd_connect_db --> off
httpd_enable_ftp_server --> off
sftpd_anon_write --> off
sftpd_enable_homedirs --> off
sftpd_full_access --> off
sftpd_write_ssh_home --> off
tftp_anon_write --> off
[root@server ~]# _
```

setsebool

Sets the Boolean value of a service option

```
[root@server ~]# setsebool allow_ftpd_anon_write 1
[root@server ~]# getsebool -a | grep ftp
allow_ftpd_anon_write --> on
allow_ftpd_full_access --> off
allow_ftpd_use_cifs --> off
allow_ftpd_use_nfs --> off
ftp_home_dir --> off
ftpd_connect_db --> off
httpd_enable_ftp_server --> off
sftpd_anon_write --> off
sftpd_enable_homedirs --> off
sftpd_full_access --> off
sftpd_write_ssh_home --> off
tftp_anon_write --> off
[root@server ~]# _
```

```
-P Makes the changes persistent
```

chcon

Changes the context of a file, directory, or service

```
[root@server ~]# cat > my_file
this is test file
[root@server ~]# ls -Z my_file
-rw-r--r--. root root unconfined_u:object_r:admin_home_t:s0 my_file
[root@server ~]# chcon -vu system_u my_file
changing security context of `my_file'
[root@server ~]# ls -Z my_file
-rw-r--r--. root root system_u:object_r:admin_home_t:s0 my_file
[root@server ~]# _
```

Options:

```
-f Suppresses error messages
-u Sets user context
-r Sets role context
-t Sets type context (domain)
-R Changes recursively
-v Provides verbose output
```

restorecon

Resets the context of an object

```
[root@server ~]# restorecon -F my_file
[root@server ~]# ls -Z my_file
-rw-r--r--. root root system_u:object_r:admin_home_t:s0 my_file
[root@server ~]# _
```

Options:

```
-i Ignores files that don't exist
-p Shows progress
-v Shows changes as they happen
-F Resets context
```

semanage

To review the status of current users, run the semanage login -l command

```
[root@server ~]# semanage login -l

Login Name                SELinux User              MLS/MCS Range

__default__               unconfined_u              s0-s0:c0.c1023
root                      unconfined_u              s0-s0:c0.c1023
system_u                  system_u                  s0-s0:c0.c1023
[root@server ~]# _
```

listing context

To see the context of a particular file, run the ls -Z command.

```
[root@server ~]# ls -Z
-rw-------  root root system_u:object_r:admin_home_t:s0 anaconda-ks.cfg
drwxr-xr-x. root root unconfined_u:object_r:admin_home_t:s0 Desktop
drwxr-xr-x. root root unconfined_u:object_r:admin_home_t:s0 Documents
drwxr-xr-x. root root unconfined_u:object_r:admin_home_t:s0 Downloads
-rw-r--r--. root root system_u:object_r:admin_home_t:s0 install.log
-rw-r--r--. root root system_u:object_r:admin_home_t:s0 install.log.syslog
drwxr-xr-x. root root unconfined_u:object_r:admin_home_t:s0 Music
drwxr-xr-x. root root unconfined_u:object_r:admin_home_t:s0 Pictures
```

To check the SELinux labels associated with service

```
[root@server ~]# ps -ZC sshd
LABEL                             PID TTY          TIME CMD
system_u:system_r:sshd_t:s0-s0:c0.c1023 1427 ? 00:00:00 sshd
[root@server ~]# _
```

how to change SELinux mode

In this article I would cover following RHCSA exam objectives

- **How to set enforcing mode for SELinux**
- **How to set permissive mode for SELinux**
- **How to disable SELinux**

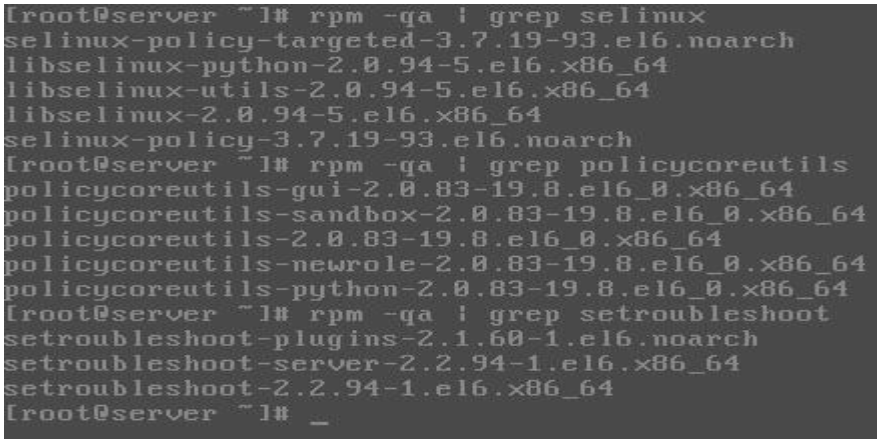
SELinux is including in default installation of RHEL7. When you install RHEL7 SELinux is automatically installed with enforcing mode. But for exam you should know which rpm packages are required for SELinux.

For SELinux following rpm are required.

- selinux
- policycoreutils
- setroubleshoot
- selinux-policy-targeted
- selinux-policy
- libselinux
- libselinux-python
- libselinux-utils
- policycoreutils-python
- setroubleshoot-server
- setroubleshoot-plugins

This article assumes that above packages are installed. If these packages are not installed, install them first. Before going further make sure you have all required packages installed. Use the **rpm -qa | grep selinux, rpm -q policycoreutils, and rpm -qa | grep setroubleshoot** commands to confirm that the SELinux packages are installed.

```
rpm -qa | grep selinux
rpm -qa | grep policycoreutils
rpm -qa | grep setroubleshoot
```



how to check that SELinux is running

To determine the current status of SELinux use sestatus command

```
[root@server ~]# sestatus
SELinux status:                 disabled
[root@server ~]# _
```

As suggested in the RHCSA objectives, you need to know how to “Set enforcing or permissive modes for SELinux.” There are three available modes for SELinux: enforcing, permissive, and disabled.

Disabled	SELinux is turned off and does not restrict any action.
Permissive	In permissive mode any SELinux security violation would be logged only, it means in permissive mode security violation would not be stopped.
Enforcing	In enforcing mode any SELinux security violation would be logged and service would stop. Any action that violate SELinux rule would be denied.

Configuring SELinux

You can change the mode in which SELinux operates by changing the config file. The main config file is /etc/selinux/config.

```
[root@server ~]# cat /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#      enforcing - SELinux security policy is enforced.
#      permissive - SELinux prints warnings instead of enforcing.
#      disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
#      targeted - Targeted processes are protected,
#      mls - Multi Level Security protection.
SELINUXTYPE=targeted


[root@server ~]# _
```

Before SELinux is enabled, each file on the file system must be labeled with a SELinux context. Before this happens, confined domains may be denied access, preventing your system from booting correctly. To prevent this, configure SELINUX=permissive in /etc/selinux/config

open configuration file

```
[root@server ~]# vi /etc/selinux/config _
```

set mode to permissive and save file

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#      enforcing - SELinux security policy is enforced.
#      permissive - SELinux prints warnings instead of enforcing.
#      disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of these two values:
#      targeted - Targeted processes are protected,
#      mls - Multi Level Security protection.
SELINUXTYPE=targeted
~
~
~
~
~
~
~
~
~
:wq_
```

Now reboot the system.

```
[root@server ~]# reboot -f _
```

During the next boot, file systems are labeled. The label process labels all files with a SELinux context. In permissive mode, SELinux policy is not enforced, but denials are still logged for actions that would have been denied if running in enforcing mode.

After reboot you could verify that system is in permissive mode

```
[root@server ~]# getenforce
Permissive
[root@server ~]# _
```

Before changing to enforcing mode run the grep "SELinux is preventing" /var/log/messages command to confirm that SELinux did not deny actions during the last boot.

```
[root@server ~]# cat /var/log/messages | grep "SELinux is preventing"
[root@server ~]# _
```

If SELinux did not deny actions during the last boot, this command does not return any output.

If there were no denial messages in /var/log/messages, open /etc/selinux/config file

```
[root@server ~]# vi /etc/selinux/config _
```

configure SELINUX=enforcing in /etc/selinux/config:

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
#       targeted - Targeted processes are protected,
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
~
~
~
~
~
~
~
~
~
:wq_
```

Reboot your system.

```
[root@server ~]# reboot -f _
```

After reboot, confirm that the getenforce command returns Enforcing:

```
[root@server ~]# getenforce
Enforcing
[root@server ~]# _
```

or you could use sestatus command

```
[root@server ~]# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /selinux
Current mode:                   enforcing
Mode from config file:          enforcing
Policy version:                 24
Policy from config file:        targeted
[root@server ~]# _
```

disabling of SELinux is straightforward

open configuration file

```
[root@server ~]# vi /etc/selinux/config _
```

change the mode to disable in configuration file

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
#       targeted - Targeted processes are protected,
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
~
~
~
~
~
~
~
~
~
~
~
~
:wq_
```

reboot the system

```
[root@server ~]# reboot -f _
```

after reboot confirm the status

```
[root@server ~]# sestatus
SELinux status:                 disabled
[root@server ~]# _
```

5.File System Administration

In this section you would learn how to manage Linux file system. We would start from troubleshooting commands used in file system management. Later you would learn how to create partition from command prompt. We would include basic, LVM and raid partitions. In end of this section you would learn how to secure data with luks.

Useful commands to check file system status

In RHEL7 several commands are available for file system managements. In this article we would discuss only those commands which you may need in RHCE7 exams. Before you start practice of creating and deleting partition it is better to do some practice with these commands.

df

This is handy command to check available free space. Run **df** command

```
[root@server ~]# df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/sda2              8063408   2363988   5289820  31% /
tmpfs                   396264       100    396164   1% /dev/shm
/dev/sda1               198337     27019    161078  15% /boot
/dev/sda5               495844     10524    459720   3% /home
[root@server ~]# _
```

If you feel difficulty in understanding the blocks use **-h** switch with df command

```
[root@server ~]# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda2             7.7G  2.3G  5.1G  31% /
tmpfs                 387M  100K  387M   1% /dev/shm
/dev/sda1             194M   27M  158M  15% /boot
/dev/sda5             485M   11M  449M   3% /home
/dev/sr0               58M   58M     0 100% /media/VMware Tools
[root@server ~]# _
```

Now outputs of df command look more users friendly. You could skip tmpfs and /dev/sr0 as tmpfs stand for temporary space and /dev/sr0 is my media device. This output is very useful when you need to manage disk. You could get an idea about which partition need more space or which partition is having unused free space. Linux LVM gives you an ability to change partition size without losing any data. With LVM you could reduce the size of partition which has unnecessary free space or you could expand the size of partition which requires more space. df command is very helpful when you need to make such a decision. As output of this command show size of my root partition is 7.7G and currently I am using 2.3G and available free space is 5.1G which is fine. Currently none of my partition requires more space. During the practice of LVM we would use this more frequently.

du

This is useful command to check the size of file. While df commands show the available space in partitions, du commands show the size of files in partitions. You could use df command to check the space used by each partitions. if you need more detail about any specific partition like which file is consuming more space then you could use du command. For example we would like to know how much space is used by /boot partition? how much space is available in /boot partition? what is the size of each files and directories in /boot partition? To get the answer of these questions we would first execute **df** command with **-h** switch. It would give us the answer of first and second question. To know the answer of third question, use du command with -h switch.

```
[root@server ~]# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda2             7.7G  2.4G  5.0G  32% /
tmpfs                 387M  100K  387M   1% /dev/shm
/dev/sda1             194M   27M  158M  15% /boot
/dev/sda5             485M   11M  449M   3% /home
/dev/sr0               58M   58M     0 100% /media/VMware Tools
[root@server ~]# du -h /boot
13K     /boot/lost+found
245K    /boot/efi/EFI/redhat
247K    /boot/efi/EFI
249K    /boot/efi
276K    /boot/grub
21M     /boot
[root@server ~]# _
```

You may get confuse from output. As df commands show boot partition is using 27 MB while du command is showing that /boot is using 21 MB so where is remaining 6 MB space?. This space is used by hidden files. You could use du command with -a switch to show the hidden files.

```
[root@server ~]# du -ha /boot_
```

mount

mount is the another helpful command. During the practice we would create and format partitions. mount command would show the file system type of partition. and it also help to know to the type of mount.

```
[root@server ~]# mount
/dev/sda2 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw,rootcontext="system_u:object_r:tmpfs_t:s0")
/dev/sda1 on /boot type ext4 (rw)
/dev/sda5 on /home type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
[root@server ~]# _
```

fstab configuration file

In Linux everything is represented as files. During the boot process directories specified in /etc/fstab are mounted on configured volumes, with the help of the **mount** command. In exam you would create update and delete partition. To keep these changes after reboot we need to update fstab file. Linux normally automatically mount the directory using the /etc/fstab configuration file during the boot process.

Any wrong entry in this file could crash the linux system. Before you start working with fstab file take backup first.

```
[root@server test]# cp /etc/fstab /etc/fstab.bk
[root@server test]# _
```

We suggest you to pay some time in understanding this file.

/etc/fstab explain from let to right

Sr.	Field Name	Description
1	Device	Label or UUID of device to be mounted
2	Mount Point	directory where the filesystem will be mounted
3	Filesystem Format	Describes the filesystem type. Valid filesystem types are ext, ext2, ext3, ext4, msdos, vfat, devpts, proc, tmpfs, udf, iso9660, nfs, smb, and swap. • The **tmpfs** filesystem is a virtual memory filesystem that uses both RAM and swap space. • The **devpts** filesystem relates to pseudo-terminal devices.

		• The **sysfs** filesystem provides dynamic information about system devices. • The **proc** filesystem is especially useful, as it provides dynamically. • **ext3** is the default filesystem for RHEL 5. • **ext4** is the default filesystem for RHEL 6. • **Xfs** is the default filesystem for RHEL7.
4	Mount Options	Available mount points are exec, noatime, noauto, nodev, noexec, nosuid, nouser, remount, ro, rw, suid, sync, user. Use default in this field which contain following option default mount options rw, suid, dev, exec, auto, nouser, and async
5	Dump Value	value for **dump** command. 1 means data is automatically save to disk when you exit Linux.

6	Filesystem Check Order	check order for filesystems during the boot process. **0** not checked during the boot process, default for CD/DVD **1** checked on first place, default for root / system **2** checked on second place, default for local filesystem like /home

UUID

Short for Universally Unique Identifiers.

Every formatted volume has an UUID, a unique 128-bit number. Each UUID represents either a partition, a logical volume, or a RAID array This is new feature added from RHEL6. It is same as RHEL5 LABEL option. UUID is automatically created when the volume is formatted with a command like mkfs.ext4.

fdisk utility

fdisk is available in all major operating system including Microsoft window and Mac Os. But we would discuss only linux version of fdisk. With fdisk utility you could create update and delete partitions. In this article we would explore the fdisk command options. Later in this section we would use fdisk command to manage partitions.

Explore fdisk command options

How to check available switch of fdisk command

To check available switch with fdisk command run **fdisk** command without any switch

```
[root@server ~]# fdisk

Usage:
 fdisk [options] <disk>    change partition table
 fdisk [options] -l <disk> list partition table(s)
 fdisk -s <partition>      give partition size(s) in blocks

Options:
 -b <size>                 sector size (512, 1024, 2048 or 4096)
 -c                        switch off DOS-compatible mode
 -h                        print help
 -u <size>                 give sizes in sectors instead of cylinders
 -v                        print version
 -C <number>               specify the number of cylinders
 -H <number>               specify the number of heads
 -S <number>               specify the number of sectors per track

[root@server ~]# _
```

How to check available disk and partitions with fdisk command

To check available disk and partitions on file system use **-l** switch with **fdisk** command

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

Disk /dev/sda: 12.9 GB, 12884901888 bytes
255 heads, 63 sectors/track, 1566 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000bf3bf

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          26      204800   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2              26        1046     8192000   83  Linux
/dev/sda3            1046        1177     1048576   82  Linux swap
/dev/sda4            1177        1567     3136512    5  Extended
/dev/sda5            1177        1241      512000   83  Linux
[root@server ~]# _
```

output show currently we have five partitions on ***/dev/sda*** disk. To manage disk with fdisk command we need to pass disk location as argument. Whenever you start working with fdisk command, **fdisk -l** command should be first on list. It would give you location of disk which is needed by fdisk command. Now we have mount point of disk so we could start fdisk command.
To start **fdisk** command pass mount point of disk as argument

```
[root@server ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
         switch off the mode (command 'c') and change display units to
         sectors (command 'u').

Command (m for help): _
```

as you could see in output you would get a warning message. Whether you made recommended change or not result would be same. In exam we should focus on result. So simple ignore it. Whether or not recommended changes are made, fdisk provides the same prompt, where you can press **m** to list basic fdisk commands.

press **m** on fdisk command prompt to get the list of all available commands

```
Command (m for help): m
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)

Command (m for help): _
```

During the exam never hesitate to take help. You should use all available resources. We use fdisk in next article so press **q** to quite form fdisk.

```
Command (m for help): q

[root@server ~]# _
```

how to create partition using fdisk

In this article we would use fdisk to create and manage partition. This article assumes that you have a new hard disk (or at least empty space on a current hard drive where you can add a new partition).

create a new partition of 100 MB using fdisk, format it with ext4 filesystem, and configure it on the /test1 directory in /etc/fstab so that the new partition is properly mounted the next time you boot Linux. As you have learnt from previous article ***fdisk*** command need hard disk mount point as argument. Check hard disk mount point

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

Disk /dev/sda: 12.9 GB, 12884901888 bytes
255 heads, 63 sectors/track, 1566 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000bf3bf

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          26      204800   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2              26        1046     8192000   83  Linux
/dev/sda3            1046        1177     1048576   82  Linux swap
/dev/sda4            1177        1567     3136512    5  Extended
/dev/sda5            1177        1241      512000   83  Linux
[root@server ~]# _
```

Start ***fdisk*** command.

```
[root@server ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
         switch off the mode (command 'c') and change display units to
         sectors (command 'u').
```

At the fdisk command line prompt, start with the print command ***(p)*** to print the partition table. This allows you to review the current entries in the partition table. As discuss in previous article it is not necessary to switch off DOS mode. So it is up to you whether you want to follow the recommendations or not. If you want to follow the recommendations execute following command or if you want to ignore the recommendation skip this

```
Command (m for help): p

Disk /dev/sda: 12.9 GB, 12884901888 bytes
255 heads, 63 sectors/track, 1566 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000bf3bf

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          26      204800   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2              26        1046     8192000   83  Linux
/dev/sda3            1046        1177     1048576   82  Linux swap
/dev/sda4            1177        1567     3136512    5  Extended
/dev/sda5            1177        1241      512000   83  Linux

Command (m for help): c
DOS Compatibility flag is not set

Command (m for help): u
Changing display/entry units to sectors

Command (m for help): _
```

you could have up to four primary partitions, which would correspond to numbers 1 through 4. If you need more partitions you could redesignate one partition as an extended partition. After redesignated, you could create logical partitions from extended partition. fdisk now supports the creation of more than 16 partitions on a drive. The remaining partitions are logical partitions, numbered 5 and above. To create new partition type n press enter

If free space is available, fdisk normally starts the new partition at the first available sector or cylinder. The actual size of the partition depends on disk geometry. Press enter of First cylinder line

```
Command (m for help):
Command (m for help): n
First sector (18894848-25165823, default 18894848): _
```

give the size of partition. Keep notice of format size. it is a + sign followed by size. K = Kilobyte M = Megabyte, G= Gigabyte. We want to create 100MB partition so give +100MB and press enter

```
[root@server ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
         switch off the mode (command 'c') and change display units to
         sectors (command 'u').

Command (m for help): n
First cylinder (1241-1567, default 1241):
Using default value 1241
Last cylinder, +cylinders or +size{K,M,G} (1241-1567, default 1567): +100M

Command (m for help): _
```

to save and exit type w and press enter You may get temporary fail error if another partition on that drive has been formatted and mounted.

```
[root@server ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
         switch off the mode (command 'c') and change display units to
         sectors (command 'u').

Command (m for help): n
First cylinder (1241-1567, default 1241):
Using default value 1241
Last cylinder, +cylinders or +size{K,M,G} (1241-1567, default 1567): +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 res
 busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
[root@server ~]# _
```

From command prompt you could try with partprobe command if linux is able to unmount existing partition it would return with success or if it is failed it would return with busy error message.

```
[root@server ~]# partprobe
Warning: WARNING: the kernel failed to re-read the partition table on /dev/sda (
Device or resource busy).  As a result, it may not reflect all of your changes u
ntil after reboot.
[root@server ~]# _
```

If you got failed message reboot system to take effect.

```
[root@server ~]# reboot -f _
```

After reboot login back with root and use ***fdisk*** command with ***-l*** switch

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

Disk /dev/sda: 12.9 GB, 12884901888 bytes
255 heads, 63 sectors/track, 1566 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000bf3bf

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          26      204800   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2              26        1046     8192000   83  Linux
/dev/sda3            1046        1177     1048576   82  Linux swap
/dev/sda4            1177        1567     3136512    5  Extended
/dev/sda5            1177        1241      512000   83  Linux
/dev/sda6            1241        1254      112275+  83  Linux
[root@server ~]# _
```

We have successfully created new partition /dev/sda6 but we would not be able to use it. Because it does not contain any filesystem. To make it useable we need to format it first. Ext4 was the default filesystem of RHEL6. From RHEL7 xfs is the default filesystem. The xfs filesystem reduces fragmentation, guarantees space for files, supports faster checks, and more. It even supports file timestamps in nanoseconds. it is proven technology. Given its speed and reliability, Red Hat even uses xfs as the default filesystem for partitions dedicated to the /boot directory. you can format it to the xfs filesystem using one of the following commands

```
# mkfs -t xfs /dev/sda6
# mke2fs -t xfs /dev/sda6
# mkfs.xfs /dev/sda6
```

Now create a mount point as given in question

```
[root@server ~]# mkdir /test
[root@server ~]# _
```

mount partition and test it. **lost+found** is a special directory and it automatically created on mount point of any partition.

```
[root@server ~]# mount /dev/sda6 /test
[root@server ~]# ls -a /test
.  ..  lost+found
[root@server ~]# _
```

linux maintain filesystem information in ***/etc/fstab*** take its backup first

```
[root@server test]# cp /etc/fstab /etc/fstab.bk
[root@server test]# _
```

to mount this partition permanently open ***/etc/fstab***

```
[root@server test]# vi /etc/fstab_
```

make an entry for this partition in end of file

```
/dev/sda6 /test xfs defaults 0 0
```

to test reboot system and check mounted partition

```
[root@server ~]# cd /test
[root@server test]# ls
lost+found
[root@server test]# _
```

Now we have created and mounted partition. You could use it.

how to delete partition from fdisk command

In our previous article we have created a simple partition of 100MB using fdisk command. Now in this article I would show you how you could delete partition using fdisk command.

Use **mount** command to locate mount point of partition.

```
[root@server ~]# mount
/dev/sda2 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw,rootcontext="system_u:object_r:tmpfs_t:s0")
/dev/sda1 on /boot type ext4 (rw)
/dev/sda5 on /home type ext4 (rw)
/dev/sda6 on /test type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
[root@server ~]# _
```

As output show our newly created partition /dev/sda6 is mounted on /test. Before we remove any partition we need to unmount it. Use **umount** command to unmount it.

```
[root@server ~]# umount /test
[root@server ~]# mount
/dev/sda2 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw,rootcontext="system_u:object_r:tmpfs_t:s0")
/dev/sda1 on /boot type ext4 (rw)
/dev/sda5 on /home type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
[root@server ~]# _
```

Now run **fdisk** command

```
[root@server ~]# fdisk /dev/sda_
```

use **p** at fdisk command prompt to print current file system

```
[root@server ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
         switch off the mode (command 'c') and change display units to
         sectors (command 'u').

Command (m for help): p

Disk /dev/sda: 12.9 GB, 12884901888 bytes
255 heads, 63 sectors/track, 1566 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000bf3bf

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          26      204800   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2              26        1046     8192000   83  Linux
/dev/sda3            1046        1177     1048576   82  Linux swap / Solaris
/dev/sda4            1177        1567     3136512    5  Extended
/dev/sda5            1177        1241      512000   83  Linux
/dev/sda6            1241        1254      112275+  83  Linux

Command (m for help): _
```

We want to delete /dev/sda6 partition use **d** at command prompt

```
[root@server ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
         switch off the mode (command 'c') and change display units to
         sectors (command 'u').

Command (m for help): p

Disk /dev/sda: 12.9 GB, 12884901888 bytes
255 heads, 63 sectors/track, 1566 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000bf3bf

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          26      204800   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2              26        1046     8192000   83  Linux
/dev/sda3            1046        1177     1048576   82  Linux swap / Solaris
/dev/sda4            1177        1567     3136512    5  Extended
/dev/sda5            1177        1241      512000   83  Linux
/dev/sda6            1241        1254      112275+  83  Linux

Command (m for help): d_
```

Now give partition number which we want to delete in our case it would **6**

```
         switch off the mode (command 'c') and change display units to
         sectors (command 'u').

Command (m for help): p

Disk /dev/sda: 12.9 GB, 12884901888 bytes
255 heads, 63 sectors/track, 1566 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000bf3bf

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          26      204800   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2              26        1046     8192000   83  Linux
/dev/sda3            1046        1177     1048576   82  Linux swap / Solaris
/dev/sda4            1177        1567     3136512    5  Extended
/dev/sda5            1177        1241      512000   83  Linux
/dev/sda6            1241        1254      112275+  83  Linux

Command (m for help): d
Partition number (1-6): 6
```

Use **w** at command prompt

```
Disk identifier: 0x000bf3bf

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          26      204800   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2              26        1046     8192000   83  Linux
/dev/sda3            1046        1177     1048576   82  Linux swap / Solaris
/dev/sda4            1177        1567     3136512    5  Extended
/dev/sda5            1177        1241      512000   83  Linux
/dev/sda6            1241        1254      112275+  83  Linux

Command (m for help): d
Partition number (1-6): 6

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 or after you run partprobe(8) or kpartx(8)
Syncing disks.
[root@server ~]# _
```

As we have discussed in our previous article if kernel is unable to unmount the partition; it would return with error code 16. It requires a reboot to locate new partition table. We should remove entry from fstab before reboot. open /etc/fstab

remove entry from fstab

```
#
# /etc/fstab
# Created by anaconda on Sat Aug 11 19:06:28 2012
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=878a11a3-3078-43d5-b4a4-0ff99f184bc4 /                       ext4    defaul
ts          1 1
UUID=56f99ced-0da2-4244-9976-42f61212ceca /boot                   ext4    defaul
ts          1 2
UUID=55a76f14-28b4-43eb-8d8a-4b216a246646 /home                   ext4    defaul
ts          1 2
UUID=1a2769c8-a5ac-488d-8f21-1b8b99670392 swap                    swap    defaul
ts          0 0
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
/dev/sda6               /test                   ext4    defaults        0 0
~
~
~
"/etc/fstab" 17L, 937C
```

after removing entry from fstab save file and quit.

```
#
# /etc/fstab
# Created by anaconda on Sat Aug 11 19:06:28 2012
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=878a11a3-3070-43d5-b4a4-0ff99f184bc4 /                       ext4    defaul
ts          1 1
UUID=56f99ced-0da2-4244-9976-42f61212ceca /boot                   ext4    defaul
ts          1 2
UUID=55a76f14-28b4-43eb-8d0a-4b216a246646 /home                   ext4    defaul
ts          1 2
UUID=1a2769c8-a5ac-480d-8f21-1b8b99670392 swap                    swap    defaul
ts          0 0
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
~
~
~
:wq_
```

now reboot the system

```
[root@server ~]# reboot -f_
```

after restart check the status of mount point. As you have learnt from our previous article that if a partition is mounted on any directory, it would contain lost + found folder.

```
[root@server ~]# cd /test
[root@server test]# ls -a
. ..
[root@server test]# _
```

6.How To Create Swap Partition

RHEL use swap space as overflow for RAM. you could be able to create new swap space. In this article I would create swap space using fdisk command. In previous article you have learnt how to create partitions with fdisk, just one additional step is required to set up that partition for swap space. Before you start make use sure you have enough free and unparted space to create new partition...

Create partition for swap space

Start **fdisk** command.

```
[root@server ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
         switch off the mode (command 'c') and change display units to
         sectors (command 'u').
```

To create new partition type **n** press enter

If free space is available, fdisk normally starts the new partition at the first available sector or cylinder. The actual size of the partition depends on disk geometry. Press enter of First cylinder line

```
Command (m for help):
Command (m for help): n
First sector (18894848-25165823, default 18894848): _
```

give the size of partition. Keep notice of format size. it is a + sign followed by size . K = Kilobyte M = Megabyte, G= Gigabyte. We want to create 100MB partition so give +100MB and press enter

```
[root@server ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
         switch off the mode (command 'c') and change display units to
         sectors (command 'u').

Command (m for help): n
First cylinder (1241-1567, default 1241):
Using default value 1241
Last cylinder, +cylinders or +size{K,M,G} (1241-1567, default 1567): +100M

Command (m for help): _
```

we need to change file type of partition, otherwise fdisk would make it ext4 which is default filesystem for rhel6

type **l** at command prompt

```
Command (m for help): n
First cylinder (1241-1567, default 1241):
Using default value 1241
Last cylinder, +cylinders or +size{K,M,G} (1241-1567, default 1567): +100M

Command (m for help): l_
```

it would list all available file type

```
 2  XENIX root      3c  PartitionMagic  83  Linux           c4  DRDOS/sec (FAT-
 3  XENIX usr       40  Venix 80286     84  OS/2 hidden C:  c6  DRDOS/sec (FAT-
 4  FAT16 <32M      41  PPC PReP Boot   85  Linux extended  c7  Syrinx
 5  Extended        42  SFS             86  NTFS volume set da  Non-FS data
 6  FAT16           4d  QNX4.x          87  NTFS volume set db  CP/M / CTOS / .
 7  HPFS/NTFS       4e  QNX4.x 2nd part 88  Linux plaintext de  Dell Utility
 8  AIX             4f  QNX4.x 3rd part 8e  Linux LVM       df  BootIt
 9  AIX bootable    50  OnTrack DM      93  Amoeba          e1  DOS access
 a  OS/2 Boot Manag 51  OnTrack DM6 Aux 94  Amoeba BBT      e3  DOS R/O
 b  W95 FAT32       52  CP/M            9f  BSD/OS          e4  SpeedStor
 c  W95 FAT32 (LBA) 53  OnTrack DM6 Aux a0  IBM Thinkpad hi eb  BeOS fs
 e  W95 FAT16 (LBA) 54  OnTrackDM6      a5  FreeBSD         ee  GPT
 f  W95 Ext'd (LBA) 55  EZ-Drive        a6  OpenBSD         ef  EFI (FAT-12/16/
10  OPUS            56  Golden Bow      a7  NeXTSTEP        f0  Linux/PA-RISC b
11  Hidden FAT12    5c  Priam Edisk     a8  Darwin UFS      f1  SpeedStor
12  Compaq diagnost 61  SpeedStor       a9  NetBSD          f4  SpeedStor
14  Hidden FAT16 <3 63  GNU HURD or Sys ab  Darwin boot     f2  DOS secondary
16  Hidden FAT16    64  Novell Netware  af  HFS / HFS+      fb  VMware VMFS
17  Hidden HPFS/NTF 65  Novell Netware  b7  BSDI fs         fc  VMware VMKCORE
18  AST SmartSleep  70  DiskSecure Mult b8  BSDI swap       fd  Linux raid auto
1b  Hidden W95 FAT3 75  PC/IX           bb  Boot Wizard hid fe  LANstep
1c  Hidden W95 FAT3 80  Old Minix       be  Solaris boot    ff  BBT
1e  Hidden W95 FAT1

Command (m for help): _
```

type **t** at command prompt

```
Command (m for help): t
Partition number (1-6): _
```

as output of **l** command show the partition ID for swap is **82** so we need to change file type to 82. First press the number of our partition and then type the partition ID for swap partition

```
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 / Solaris)
```

now save with **w** command

```
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 / Solaris)

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
 busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
[root@server test]# _
```

If you did not get error code 16 it means kernel has written new partition table. you could run partprobe command, which would reflect the new change. As we get error code 16, it means kernel could not write new partition table we need to do restart to locate the change in partition table. reboot the system

```
[root@server ~]# reboot -f _
```

After reboot we need to format swap partition. swap volumes are formatted with the mkswap command. format swap partition

```
[root@server ~]# mkswap /dev/sda6
Setting up swapspace version 1, size = 112268 KiB
no label, UUID=ddb77e4c-6751-40d6-a3f8-6c725e17e854
```

activate with the **swapon** command

```
[root@server ~]# mkswap /dev/sda6
Setting up swapspace version 1, size = 112268 KiB
no label, UUID=ddb77e4c-6751-40d6-a3f8-6c725e17e854
[root@server ~]# swapon /dev/sda6
```

If the new swap volume is recognized, you would see it in the /proc/swaps file

```
[root@server ~]# mkswap /dev/sda6
Setting up swapspace version 1, size = 112268 KiB
no label, UUID=ddb77e4c-6751-40d6-a3f8-6c725e17e854
[root@server ~]# swapon /dev/sda6
[root@server ~]# cat /proc/swaps
Filename                                Type            Size    Used    Priority
/dev/sda3                               partition       1048568 0       -1
/dev/sda6                               partition       112264  0       -2
[root@server ~]# _
```

to make it available after reboot open /etc/fstab

```
[root@server test]# vi /etc/fstab_
```

at end of the file

```
_
#
# /etc/fstab
# Created by anaconda on Sat Aug 11 19:06:28 2012
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=078a11a3-3070-43d5-b4a4-0ff99f184bc4 /                       ext4    defaul
ts        1 1
UUID=56f99ced-0da2-4244-9976-42f61212ceca /boot                   ext4    defaul
ts        1 2
UUID=55a76f14-28b4-43eb-8d0a-4b216a246646 /home                   ext4    defaul
ts        1 2
UUID=1a2769c8-a5ac-480d-8f21-1b8b99670392 swap                    swap    defaul
ts        0 0
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
/dev/sda6               swap                    swap    defaults        0 0
~
~
~
"/etc/fstab" 17L, 936C
```

add a line for swap and save it to check it restart the system

```
[root@server ~]# reboot -f _
```

after reboot verify that our swap is on and working properly

```
[root@server ~]# cat /proc/swaps
Filename                                Type            Size    Used    Priority
/dev/sda3                               partition       1048568 0       -1
/dev/sda6                               partition       112264  0       -2
```

As output show our swap volume is mounted and working properly.

- **how to delete swap partition**

In our last article you have seen how could we create swap space in linux using fdisk command. Now in this article we would remove that swap space. check the swap space

```
[root@server ~]# cat /proc/swaps
Filename                                Type            Size    Used    Priority
/dev/sda3                               partition       1048568 0       -1
/dev/sda6                               partition       112264  0       -2
```

we need to turnoff swap before we could remove it, use **swapoff** command to deactivate swap

```
[root@server ~]# cat /proc/swaps
Filename                                Type            Size    Used    Priority
/dev/sda3                               partition       1048568 0       -1
/dev/sda6                               partition       112264  0       -2
[root@server ~]# swapoff /dev/sda6
[root@server ~]# cat /proc/swaps
Filename                                Type            Size    Used    Priority
/dev/sda3                               partition       1048568 0       -1
[root@server ~]# _
```

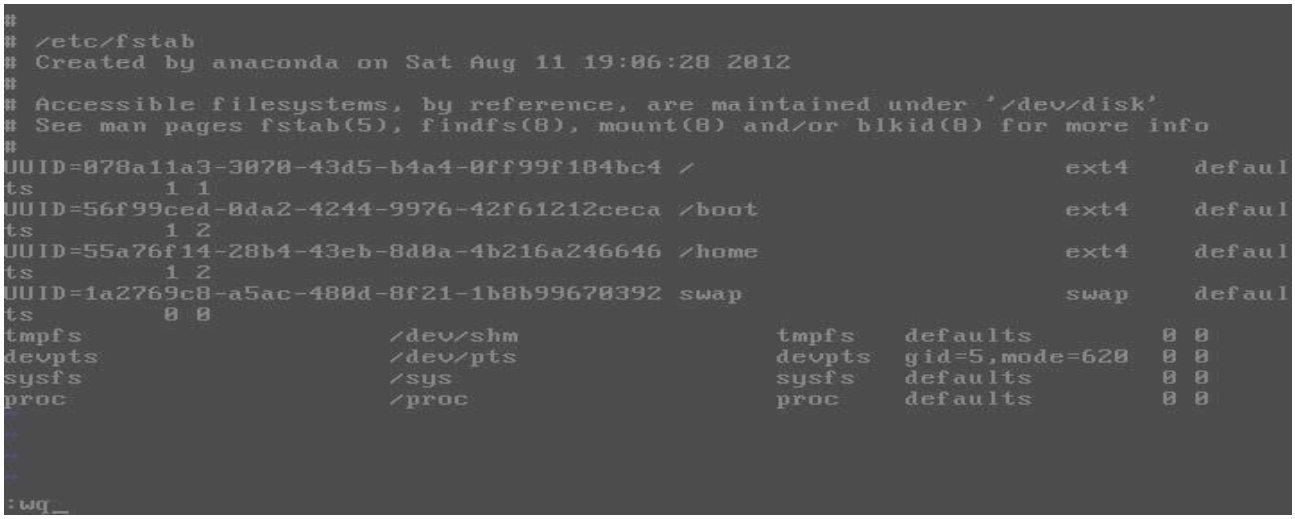
now remove entry from /etc/fstab. Open fstab

```
[root@server test]# vi /etc/fstab_
```

locate the entry for swap, we added it in the end

```
_
#
# /etc/fstab
# Created by anaconda on Sat Aug 11 19:06:28 2012
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=078a11a3-3070-43d5-b4a4-0ff99f184bc4 /                       ext4    defaul
ts          1 1
UUID=56f99ced-0da2-4244-9976-42f61212ceca /boot                   ext4    defaul
ts          1 2
UUID=55a76f14-28b4-43eb-8d0a-4b216a246646 /home                   ext4    defaul
ts          1 2
UUID=1a2769c8-a5ac-480d-8f21-1b8b99670392 swap                    swap    defaul
ts          0 0
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
/dev/sda6               swap                    swap    defaults        0 0
~
~
_
"/etc/fstab" 17L, 936C
```

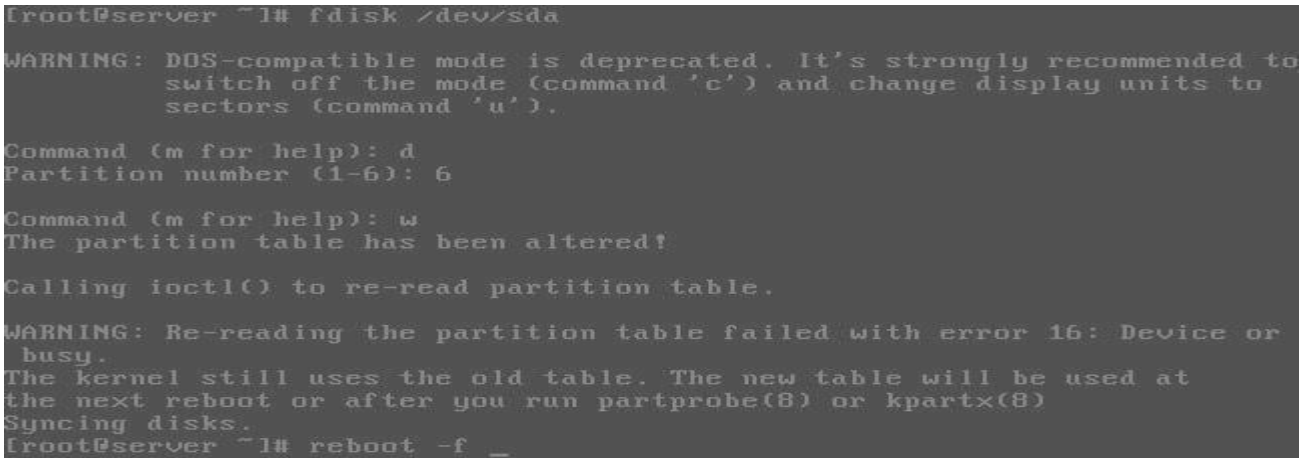
remove our entry for swap partition and save the file



Now run fdisk command. At fdisk command prompt

- use d command to delete the partition
- Give the partition number it would be 6 for us, you can use p command here to list all partition
- save with w command.

as we have discussed in our earlier article we have to do a reboot if kernel is unable to reflect new change in partition table (error code 16). Reboot the system



after reboot verify

as output show we have successfully removed swap space.

7. What is LVM

Logical volume management provides a higher−level view of the disk storage on a computer system than the traditional view of disks and partitions. This gives the system administrator much more flexibility in allocating storage to applications and users.

The logical volume manager also allows management of storage volumes in user−defined groups, allowing the system administrator to deal with sensibly named volume groups such as "development" and "sales" rather than physical disk names such as "sda" and "sdb".

Advantage of Logical Volume Management

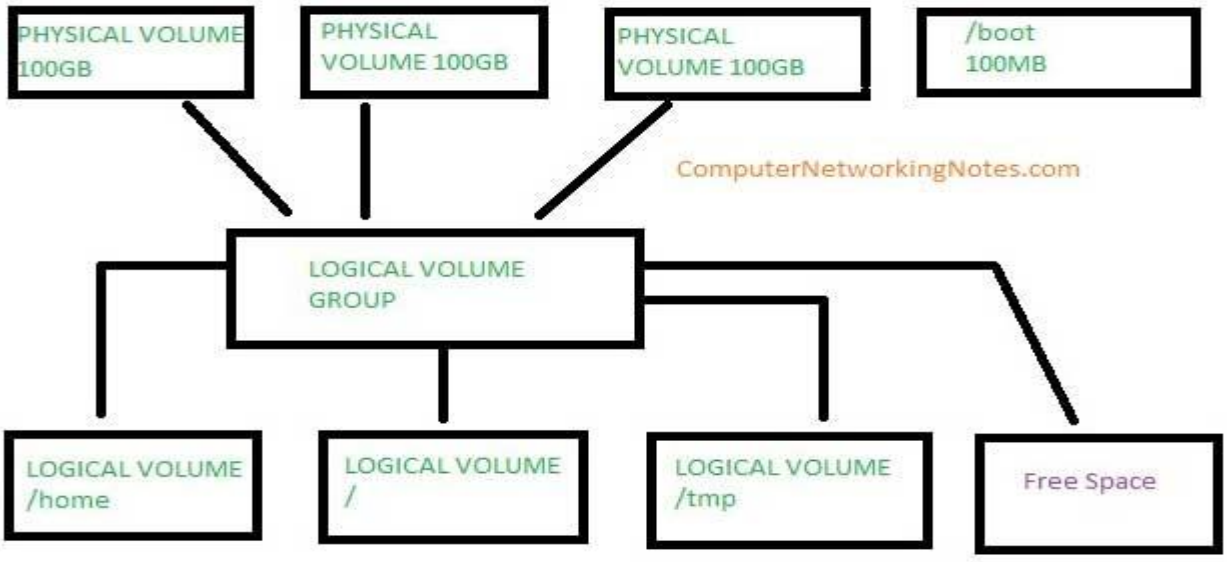
One of the difficult decisions facing a new user installing Linux for the first time is how to partition the disk drive. The need to estimate just how much space is likely to be needed for system files and user files makes the installation more complex than is necessary and some users simply opt to put all their data into one large partition in an attempt to avoid the issue.

Once the user has guessed how much space is needed for /home /usr / (or has let the installation program do it) then is quite common for one of these partitions to fill up even if there is plenty of disk space in one of the other partitions.

With logical volume management, the whole disk would be allocated to a single volume group and logical volumes created to hold the / /usr and /home file systems. If, for example the /home logical volume later filled up but there was still space available on /usr then it would be possible to shrink /usr by a few megabytes and reallocate that space to /home.

Another alternative would be to allocate minimal amounts of space for each logical volume and leave some of the disk unallocated. Then, when the partitions start to fill up, they can be expanded as necessary.

LVM allows administrators to divide hard drive space into physical volumes (PV), which can then be combined into logical volume groups (VG), which are then divided into logical volumes (LV) on which the filesystem and mount point are created.



As shown in image because a logical volume group can include more than one physical volume, a mount point can include more than one physical hard drive, meaning the largest mount point can be

larger than the biggest hard drive in the set. These logical volumes can be resized later if more disk space is needed for a particular mount point. After the mount points are created on logical volumes, a filesystem must be created on them.

LVM is used by default during installation for all mount points except the /boot partition, which cannot exist on a logical volume. But you could create LVM after the installation. In our next article I would show you how to create and update LVM in rhce exam.

Creating LVM in RedHat Linux

LVM (Logical Volume Manager): LVM provides a flexible and high level approach to managing disk space. Instead of each disk drive being split into partitions of fixed sizes onto which fixed size file systems are created, LVM provides a way to group together disk space into logical volumes which can be easily re-sized and moved. In addition, LVM allows administrators to carefully control disk space assigned to different groups of users by allocating distinct volume groups or logical volumes to those users. When the space initially allocated to the volume is exhausted the administrator can simply add more space without having to move the user files to a different file system. LVM have partition id 8e.

Following steps are followed to create LVM.
1. Create PV
2. Create VG
3. Create LVM
4. Format LVM & mount it.

1. Create PV (Physical Volume):

#fdisk -l

Disk /dev/sda: 120.0 GB, 120034123776 bytes
255 heads, 63 sectors/track, 14593 cylinders, total 234441648 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000a808c

Device Boot	Start	End	Blocks	Id	System
/dev/sda1 *	2048	206847	102400	83	Linux
/dev/sda2	206848	57550847	28672000	83	Linux
/dev/sda3	58605120	64227869	2811375	83	Linux
/dev/sda4	57552896	57960446	203775+	83	Linux

In above output there is three Linux partitions, from these given partition we are going to create new physical volume by using sda3 & sda4.

#pvcreate /dev/sda3

#pvcreate /dev/sda4

To show the new physical volume we use following syntax
#pvdisplay

output:

--- Physical volume ---

PV Name	/dev/sda3
VG Name	
PV Size	2.34 GiB / not usable 32.00 MiB
Allocatable	yes
PE Size	4.00 MiB
Total PE	874
Free PE	97
Allocated PE	777
PV UUID	xL33Gt-pBB3-4XdC-7BnK-QSZk-6XyI-T0fSh

--- Physical volume ---

PV Name	/dev/sda4
VG Name	
PV Size	2.68 GiB / not usable 1.48 MiB
Allocatable	yes
PE Size	4.00 MiB
Total PE	686
Free PE	436
Allocated PE	250
PV UUID	0gJH6J-sg0N-XeSh-rwcd-6DK0-CXef-0gt6vw

2. Create VG (Volume Group)

#vgcreate vol0 /dev/sda3 /dev/sda4

in above syntax here vol0 is VG label. To display new volume group

#vgdisplay
Output:

--- Volume group ---

VG Name	vol0
System ID	
Format	lvm2
Metadata Areas	1
Metadata Sequence No	4
VG Access	read/write
VG Status	resizable
MAX LV	0
Cur LV	3
Open LV	2
Max PV	0
Cur PV	1
Act PV	1

VG Size	27.31 GiB
PE Size	32.00 MiB
Total PE	874
Alloc PE / Size	777 / 24.28 GiB
Free PE / Size	97 / 3.03 GiB
VG UUID	v4zfUr-nEh4-JKyu-xcge-RRlp-O0kp-HQVCT3

3. Create LVM

#lvcreate –size 240M –name lv1 vol0

by using above syntax this will create 240Mb size new lvm and here lv1 is label of new lvm. To display information of this volume use following syntax

#lvdisplay

Output:

--- Logical volume ---

LV Name	/dev/vol0/lv1
VG Name	vol0
LV UUID	2sXIPe-AlDg-WFMy-ijzb-UUBS-Tj6H-lKGLDb
LV Write Access	read/write
LV Status	available
# open	1
LV Size	240.00 MiB
Current LE	16
Segments	1
Allocation	inherit
Read ahead sectors	auto
- currently set to	256
Block device	253:2

4. now need to format lvm and mount it.

#mkfs.ext4 /dev/vol0/lv1

Above syntax will format lv1 in ext4 filesystem. Now need to mount this drive.

#mount /dev/vol0/lv1 /media/lv

Note: - We can **extend** and **reduce** lvm size according to our requirement.

1. Extend LVM:

*** extend lvm**

#lvextend --size +40M /dev/vol0/home

Output:

Rounding up size to full physical extent 40.00 MiB

Extending logical volume lv1 to 280.00 MiB

Logical volume home successfully resized

*** resize file system**

#resize2fs /dev/vol0/lv1

Output:

resize2fs 1.41.12 (17-May-2010)

Filesystem at /dev/vol0/lv1 is mounted on /media/lv; on-line resizing required

old desc_blocks = 1, new_desc_blocks = 1

Performing an on-line resize of /dev/vol0/lv1 to 147456 (4k) blocks.

The filesystem on /dev/vol0/lv1 is now 147456 blocks long.

2. Reduce lvm size: To reduce any partition you need to follow these steps.
* unmount partition
* scan file system
* resize file sytem by 60Mb
* lvreduce
* mount reduced partition

*** unmount partition:**

[root@desktop13 ~]# umount /dev/vol0/lv1

*** scan file system:**

[root@desktop13 ~]# e2fsck -f /dev/vol0/lv1

e2fsck 1.41.12 (17-May-2010)

Pass 1: Checking inodes, blocks, and sizes

Pass 2: Checking directory structure

Pass 3: Checking directory connectivity

Pass 4: Checking reference counts

Pass 5: Checking group summary information

/dev/vol0/lv1: 27/40960 files (0.0% non-contiguous), 6787/147456 blocks

*** resize file system by 60Mb**

[root@desktop13 ~]# resize2fs /dev/vol0/lv1 60M

resize2fs 1.41.12 (17-May-2010)

Resizing the filesystem on /dev/vol0/lv1 to 15360 (4k) blocks.

The filesystem on /dev/vol0/lv1 is now 15360 b

*** lvreduce:**

[root@desktop13 ~]# lvreduce --size 60M /dev/vol0/lv1

Rounding up size to full physical extent 64.00 MiB

WARNING: Reducing active and open logical volume to 64.00 MiB

THIS MAY DESTROY YOUR DATA (filesystem etc.)

Do you really want to reduce lv1? [y/n]: y

Reducing logical volume lv1 to 64.00 MiB

Logical volume lv1 successfully resized

*** mount reduced partition**

[root@desktop13 ~]# mount /dev/vol0/lv1 /media/lv

now we have new lvm size of 60Mb.

8.KERNEL

Linux / UNIX Kernel

"kernel is the core part of any operating system which allows every activity which is going to occur in your operating system". For example

accessing any file, playing a movie.

1. Type the following command to see running kernel version:

'$ uname -r'

Output:

2.6.22-14-generic

Where,

* 2 : Kernel version

* 6 : The major revision of the kernel

* 22 : The minor revision of the kernel

* 14 : Immediate fixing / bug fixing for critical error

Redhat Certified Engineer (RHCE)

Table of Contents

1. Configure Telnet Server

Telnet Server in RHEL 7. Telnet protocol allows you to connect to remote hosts over TCP/IP network. Telnet by default does not encrypt any data sent over the connection Telnet was developed in 1969. Telnet was initially developed for private use where security was not primary concern. Telnet protocol has serious security issue. Security expert recommend that the use of Telnet for remote login should be discontinued under all normal circumstances.

• **Telnet Server**

• **Telnet Client**

Telnet Sever

Telnet server software is installed on remote host. You need to configure it before client can connect with it. Telnet Client

Telnet client software allows you to connect telnet server. Once telnet client establishes a connection to the remote host, client becomes a virtual terminal, allowing you to communicate with the remote host from your computer.

Security issue with Telnet .

Anyone who has access to network device located on the network between the two hosts like router, switch, hub or gateway where Telnet is being used can intercept the packets passing by and obtain login, password and whatever else is typed with a packet sniffer software.

Telnet protocol have no implementations that would ensure that communication is carried out between the two hosts is not intercepted in the middle.

In RHEL 7 Telnet is part of the xinetd daemon.

- Telnet use plain text to transmit password.
- root user is not allowed to connect using Telnet.
- Command-line telnet clients are built into all major operating systems.
- RedHat recommends you to use SSH to connect a system instead of Telnet.
- Use Telnet in LAB environment where security is not concern.

Configure Telnet in RHEL 7

Three RPM are required to configure telnet server in linux.

- xinetd
- telnet-server

Login from root user and check required RPM. If you do not have the telnet-server or telnet packages installed, you can install them with the RPMs available from your installation media. The version numbers of the package should not matter, Red Hat Network (RHN) will always provide you with the latest version of the package.

[root@localhost Desktop]# rpm -q xinetd

xinetd-2.3.15-12.el7.x86_64

[root@localhost Desktop]# rpm -q telnet-server

telnet-server-0.17-59.el7.x86_64

Once you have the packages installed, check the **/etc/xinetd.d/telnet** file and if not then create it

[root@localhost Desktop]# cd /etc/xinetd.d/

[root@localhost xinetd.d]# vim telnet (put the content into the file)

Service telnet {

Disable = no

Socket_type = stream

Wait = no

User = root

Server = /usr/sbin/in.telnetd

}

You will need to restart the xinetd service

[root@localhost xinetd.d]# systemctl restart xinetd

To make the xinetd persistant which it will automatically started after reboot

[root@localhost xinetd.d]# systemctl enable xinetd

As I said above root user is not allowed to login from telnet. We need to create a normal user account.

Important: - after making everything correct if from client side telnet show unable to connect or unknow host the add a firewall rule using below method

[root@localhost xinetd.d]# firewall-cmd --add-port=23/tcp

OR

[root@localhost xinetd.d]# firewall-cmd --add-service=telnet

To connect with telnet server, we need specify server IP address which you can check with ifconfig command. You should test telnet configuration before connecting from client computer.

To login from client side using telnet command use below given method

[root@localhost xinetd.d]# telnet 192.168.0.13   # Here IP of Server

Note: To login via root upon telnet server make changes like given below

By default, root user is not allowed to login through the terminal session. To allow root user use below steps

[root@localhost xinetd.d]# vim /etc/securetty

Pts/0

Pts/1

2. Configure SSH Server

SSH server also used to login remotely but having no security issue, to configure SSH server and SSH client in RHEL7.

RHCE 7 configured as below

- Configure key-based SSH authentication.
- Configure additional SSH options described in documentation.

As a Linux administrator you should know

- SSH stand for Secure Shell.
- SSH is a network protocol for secure data communication.
- SSH protocol allows remote command line login.
- SSH protocol enables remote command execution.
- To use SSH you need to deploy SSH Server and SSH Client program respectively.
- OpenSSH is a FREE version of the SSH.
- Telnet, rlogin, and ftp transmit unencrypted data over internet.
- OpenSSH encrypt data before sending it over insecure network like internet.
- OpenSSH effectively eliminate eavesdropping, connection hijacking, and other attacks.
- OpenSSH provides secure tunneling and several authentication methods.
- OpenSSH replace Telnet and rlogin with SSH, rcp with scp, ftp with sftp.

SSH Tools

sshd

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

ssh

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

scp

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

ssh-keygen

This utility is used to create the public/private keys.

ssh-agent

This utility holds private keys used for RSA authentication.

ssh-add

Adds RSA identities to the authentication agent, ssh-agent.

On your laptop or Desktop

- Configure a SSH server and SSH client on RHEL7.
- Create two user **user1** and **user2** and verify that both users can login in SSH server from SSH client.
- Do not allow **root** and **user1** users to login to it and allow the rest of users. To confirm it login from **user2**.
- Re-configure SSH Server to allow login only using public / private keys. Generate keys for **user2** and verify that **user2** can login using keys

How to configure SSH Server in RHEL 7

Two RPM are required to configure and run OpenSSH server.

- openssh-server
- openssh

Before you start configuration make sure that you have necessary RPM packages installed. Install if any RPM is missing.

[root@localhost ~]# rpm -q openssh-server

openssh-server-6.4p1-8.el7.x86_64

Check the current status of **sshd** service, it must be running. If service is stopped start it. Options you need with service command are **start** | **stop** | **restart** | **status**

[root@localhost xinetd.d]# systemctl status sshd

[root@localhost xinetd.d]# systemctl restart sshd

Note: To make ssh persistant after reboot use below command

[root@localhost xinetd.d]# systemctl enable sshd

Important: If during making connection if there is any connectivity issue then add a firewall rule

[root@localhost xinetd.d]# firewall-cmd --add-port=22/tcp

OR

[root@localhost xinetd.d]# firewall-cmd --add-service=ssh

Note: For the making any changes in configuration file which is (sshd_config)

[root@localhost ~]# cd /etc/ssh/

[root@localhost ssh]# ls

moduli sshd_config ssh_host_ecdsa_key.pub ssh_host_rsa_key.pub

To allow or deny users and some other configuration using ssh:

[root@localhost ~]# vim sshd_config

on line 42: uncomment and change the value to 'no'

PermitRootLogin no

on line 65: uncomment the following

PermitEmptyPasswords no
PasswordAuthentication yes
Save & quit then execute the following command.

at any line
Allowusers user1 user2 ## these users only able to login via ssh
denyusers user1 user2 ## these users are not able to login via ssh

[root@localhost xinetd.d]# systemctl restart sshd

How to configure SSH client on RHEL 7:

openssh-clients rpm is required for ssh client.

Check necessary RPM, install if any missing

[root@localhost ~]# rpm -q openssh-clients

openssh-clients-6.4p1-8.el7.x86_64

Go back on **linux client** system and verify that we have user**1** and **root**. Also verify that **user1 and root** able to login in SSH server

[root@localhost ~]# ssh user1@192.168.0.12

3. Configure FTP Server

Your will configure FTP Server on RHEL7. FTP is the most widely used protocol for file transfer. As a linux Administrator you should know

- FTP stand for File Transfer Protocol.
- FTP does not require to login directly into the remote host
- FTP transfer data without encryption
- vsftpd is the only stand-alone FTP distributed with RHEL 7
- vsftpd stand for Very Secure FTP Daemon
- vsftpd is secure, fast and stable version of FTP
- vsftpd efficiently handle large numbers of connection securely
- You should use SFTP instead of FTP while transferring data over public network like Internet

Configure FTP Server on RHEL 7: -

vsftpd package is required for FTP Server. Check whether package is installed or not.

If package is missing install it first.

Step 1:-

[root@localhost ~]# rpm -q vsftpd # If not installed then installed it using yum

[root@localhost ~]# yum install vsftpd

Step 2: Configure **vsftpd** service to start

[root@localhost ~]# systemctl restart vsftpd

Configure **vsftpd** service to start at boot: -

[root@localhost ~]# systemctl enable vsftpd

4. Manage the firewall

FTP Server is by default configured to listen on port 21. Port 21 must be opened if you have configured firewall. The configuration of a firewall for an FTP server is a relatively simple process

[root@localhost xinetd.d]# firewall-cmd --add-port=21/tcp

OR

[root@localhost xinetd.d]# firewall-cmd --add-service=ftp

Go on Server system and open main ftp configuration file **/etc/vsftpd/vsftpd.conf**

FTP non-anonymous server

In this exercise we will configure FTP server that allow local users logins to their home directories. Download/upload must be allowed for these users. Go on server system and open **/etc/vsftpd/vsftpd.conf** file

Comment **anonymous_login=YES**, uncomment **local_enable** and save the file

Enable anonymous user to upload file:

Uncomment the line anon_other_write_enable = yes

And

Chmod o+w /var/ftp/pub

open **/etc/vsftpd/user_list** file

Users listed on **/etc/vsftpd/user_list** are not allowed to login on FTP server. Add user **vikarm** in it. This file also has an entry for root user that why root user is denied from FTP login. If you want to enable root user for ftp session just remove its entry from this file [Enable root for FTP session is not recommended in any circumstances, change at your own risk.

Important: by default, root user is not allowed to login via ftp

To login via root remove name of root from two file.

1. First is /etc/vsftpd/ftpuser
2. Second /etc/vsftpd/user_list

Configure FTP client on RHEL7

[root@localhost ~]# yum install ftp

To connect with server you can use ftp command:

[root@redhat7 ~]# ftp 127.0.0.1

Connected to 127.0.0.1 (127.0.0.1)

220 (vsFTPd 3.0.2)

Name (127.0.0.1:root): ftp

331

Please specify the password.

Password:

230 Login successful.

OR

USE graphical method like web browser and filezilla software

4. Configure NIS Server

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 **192.168.1.222.** 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 tests 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 client.

Network configuration in Linux

- A linux server with ip address 192.168.1.222 and hostname Server
- A linux client with ip address 192.168.1.1 and hostname Client1
- Updated /etc/hosts file on both linux system
- Running portmap service
- Firewall should be off on server

Seven rpm are required to configure nis server. **ypserv, cach, nfs, make, rpcbind**, **ypbind, portmap** check them if not found then install

[root@localhost ~]# rpm -q ypbind

[root@localhost ~]# rpm -q rpcbind

[root@localhost ~]# rpm -q ypserv

Configure Nis server:-

Step 1:

[root@localhost ssh]# yum install ypserv

Step 2:- assign a domain-name to Nis server using below method

[root@localhost ssh]# ypdomainname adhoc.labs.example.com

Note: you can use existing username or can create your own

[root@localhost ssh]# useradd ashu

[root@localhost ssh]# passwd ashu

[root@localhost ssh]# cd /var/yp

[root@localhost ssh]# make

Note: During make command if there is any error like rpcbind is not registered then do the step third first

Step: 3 Now start or restart the service of ypserv

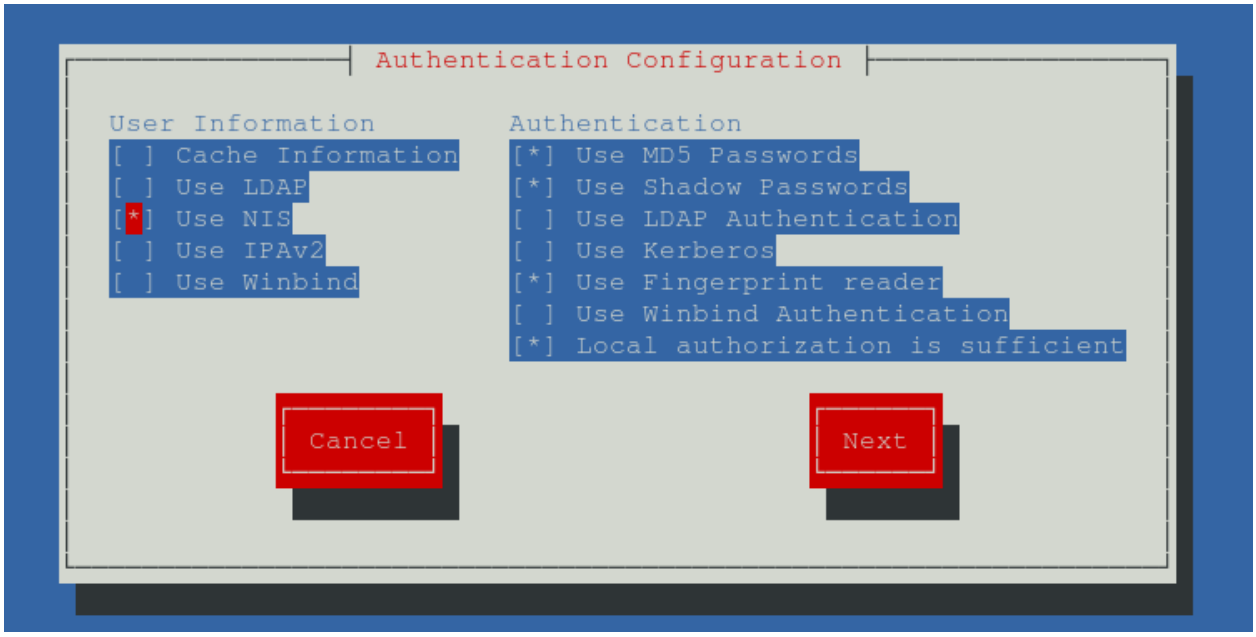
[root@localhost ssh]# systemctl restart ypserv

Configure NIS Client: -

For nis client you have to install ypbind package first and use below steps to use nis

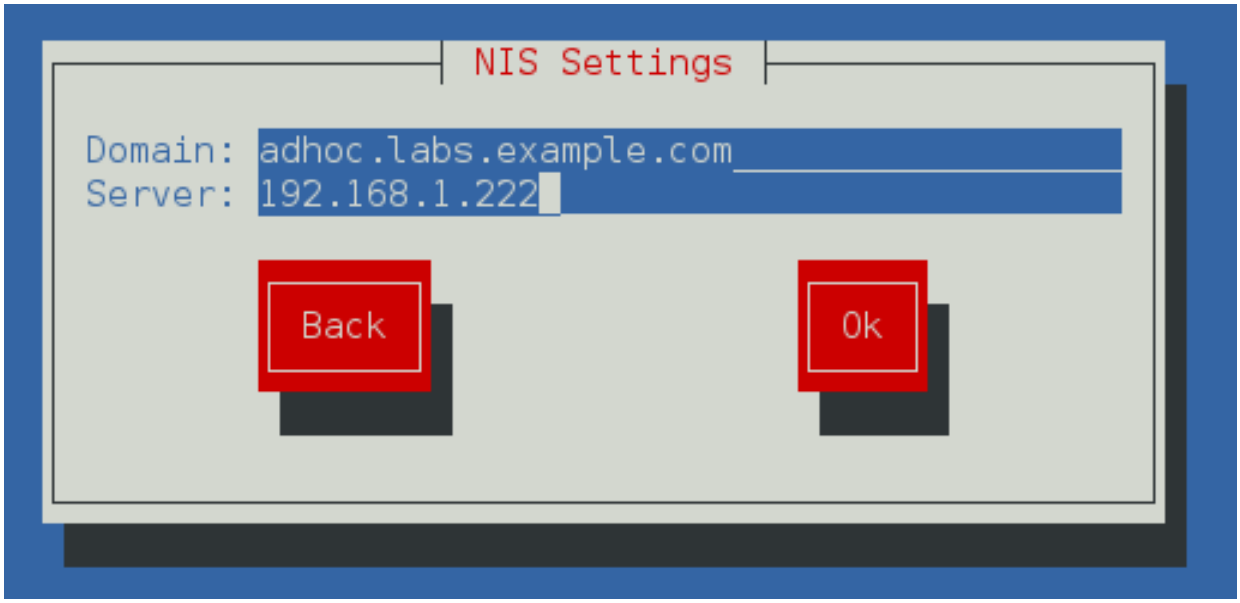
[root@localhost ssh]# yum install ypbind

[root@localhost ssh]# authconfig-tui # when you fire this command it will result like this



Click on Use Nis using space button and press next button as shown in above figure and made some entry as show below firegure

Note: In below figure domain is the same which was assigned in NIS server and IP address of server



NOTE: now you can login via any user which are created in NIS server via command line or graphical

[root@localhost ssh]# su - ashu

[ashu@localhost]$

5. NFS Server

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.

Example 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

Example 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

Example 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 client.

Network configuration in Linux

- 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 service

Step 1: Check the required rpm and if not installed then install it

[root@localhost ~]# rpm -q nfs-utils # If not installed then installed it

OR

[root@redhat7~] yum install nfs-utils

Step 2: configure nfs server

now create a **/data** directory and grant full permission to it

now open **/etc/exports** file

share **data** folder for the network of **192.168.0.254/24** with read and write access

/data 192.168.0.0/24(sync,rw)

Step 3: start or restart the service

[root@localhost ~]# systemctl enable nfs

[root@localhost ~]# systemctl restart nfs

Note: also restart **nfs daemon** with **exportfs**

[root@localhost ~]# exportfs -r

configure client system:

ping form **nfs server** and then check the **share folder by** using showmount command

[root@localhost ~]# showmount -e 192.168.0.12 ## (here 192.168.0.12 is server IP)

now **mount** this share folder on **mnt** mount point. To test this share folder change directory to **mnt** and create a **test file**

[root@localhost ~]# mount -t nfs 192.168.0.12:/data /mnt

After use you should always **unmount** from **mnt** mount point

[root@localhost ~]# **umount /mnt**

6. Web Server

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. If 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.

Example 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** check them for install

Step 1. check for software is installed or not**:**

[root@localhost ~]# rpm -q httpd

[root@localhost ~]# rpm -q httpd-devel

For Installing

[root@localhost ~]# yum install httpd httpd-devel

Now configure the ip address to **192.168.0.12** and check it

Step 2:

start **httpd daemon** and verify its running **status**

[root@localhost ~]# systemctl restart httpd

[root@localhost ~]# systemctl status httpd

Also enable the firewall rule

[root@localhost ~]# firewall-cmd --add-port=80/tcp

[root@localhost ~]# firewall-cmd --add-service=http

Step 3:

Now there are some configuration which are already configured and for extra configuration

1. By default, documentroot is /var/www/html/
2. Page name is index.html

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

[root@localhost ~]# cd /var/www/html/

echo hi >index.html

Note: after doing above step you can test via web browser by entering IP or hostname of web server

Configure virtual hosting:

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

now open **/etc/hosts** file

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

[root@localhost ~]# vim /etc/hosts

127.0.0.1 www.vinita.com

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

[root@localhost ~]# vim /etc/httpd/conf/httpd.conf

NameVirtualHost 192.168.0.12:80

Now go in the end of file and copy last seven line [virtual host tag] and paste them in the end of file.

<virtualhost 192.168.0.12:80**>**

Documentroot /var/www/html/

Servername www.vinita.com

</virtualhost>

<virtualhost 192.168.0.12:80**>**

Documentroot /var/www/virtual/

Servername www.myhome.com

</virtualhost>

[root@localhost ~]# mkdir /var/www/virtual

[root@localhost ~]# echo “hello” >index.html

Note: if you want any security in any documentroot then inside virtualhost

<virtualhost 192.168.0.12:80**>**

Documentroot /var/www/virtual

Servername www.myhome.com

<directory /var/www/virtual/>

Order allow,deny

Allow from 127.0.0.1 192.168.0.0/24

</virtualhost>

Important:

you have done necessary configuration now restart the httpd service:

[root@localhost ~]# systemctl restart httpd

Apache clients:

Important: you can use web browser for apache client to test your web page

7. Samba Server

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.

There are mixed 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.

- 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.

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

Step 1:

samba rpm is required to configure samba server. check them if not found then install

[root@localhost ~]# rpm -q samba

OR

[root@localhost ~]# rpm –qa samba*

For installing packages:

[root@localhost ~]# yum install samba

Step 2:

Create a normal user named **Vinita**

[root@localhost ~]# useradd Vinita

now create **/data** directory and grant it **full permission**

[root@localhost ~]# mkdir /data

[root@localhost ~]# chmod 777 /data

open **/etc/samba/smb.conf** main samba configuration files

[root@localhost ~]# vim /etc/samba/smb.conf

our task is to share **data** folder for **vinita** user so go in the end of file and do editing as shown here

[sharename]

Path = /data

Hosts allow = 192.168.0.0/24

Valid users = Vinita

Public = no

Writable = no

Now save the file

Finally, Now add vinita user to **samba user**

[root@localhost ~]# smbpasswd -a Vinita

Step 3. start the service of smb

[root@localhost ~]# systemctl restart smb

[root@localhost ~]# systemctl enable smb

Client configuration for samba server:

Case 1. If client is linux then install cifs-utils package if not installed and use below method to access

[root@localhost ~]# smbclient -U vinita //192.168.0.12/sharenmae

Note: Furthur you can use the same method like in ftp server to download and upload command like get and put

Case: 2 If client is windows then go to run type \\192.168.0.12 (here ip is server ip)

8. Configure Postfix SMTP Mail Server

There are a number of components that make up a complete email system. Below is a brief description of each one:

Mail User Agent

This is the part of the system that the typical user is likely to be most familiar with. The **Mail User** Agent (MUA), or mail client, is the application that is used to write, send and read email messages. Anyone who has written and sent a message on any computer has used a Mail User Agent of one type or another. Typical Graphical MUA's on Linux are Evolution, Thunderbird and KMail. For those who prefer a text based mail client, there are also the more traditional pine and mail tools.

Mail Transfer Agent

The **Mail Transfer Agent (MTA)** is the part of the email system that does much of the work of transferring the email messages from one computer to another (either on the same local network or over the internet to a remote system). Once configured correctly, most users will not have any direct interaction with their chosen MTA unless they wish to re-configure it for any reason. There are many choices of MTA available for Linux including sendmail, Postfix, Fetchmail, Qmail and **Exim**.

Mail Delivery Agent

Another part of the infrastructure that is typically hidden from the user, the Mail **Delivery Agent** (MDA) sits in the background and performs filtering on the email messages between the Mail Transfer Agent and the mail client (MUA). The most popular form of MDA is a spam filter to remove all the unwanted email messages from the system before they reach the inbox of the user's mail client (MUA). Popular MDAs are **Spamassassin** and Procmail. It is important to note that some Mail User Agent applications (such as Evolution, Thunderbird and KMail) include their own MDA filtering. Others, such as Pine and Basla, do not. This can be a source of confusion to the Linux beginner.

SMTP

SMTP is an acronym for Simple Mail Transport Protocol. This is the protocol used by the email systems to transfer mail messages from one server to another. This protocol is essentially the communications language that the **MTAs** use to talk to each other and transfer messages back and forth.

Configuring an Email System:

Many systems use the Sendmail MTA to transfer email messages and on many Linux distributions this is the default Mail Transfer Agent. Sendmail is, however, a complex system that can be difficult for beginner

and experienced user alike to understand and configure. It is also falling from favor because it is considered to be slower at processing email messages than many of the more recent MTAs available.

Many system administrators are now using Postfix or Qmail to **handle email**. Both are faster and easier to configure than Sendmail. For the purposes of this chapter, therefore, we will look at Postfix as an MTA because of its simplicity and popularity. If you would prefer to use Sendmail there are many books that specialize in the subject and that will do the subject much more justice than we can in this chapter.

Step1: Install postfix via yum (in latest CentOS/Redhat it will be preinstalled, perform a fresh install postfix in Fedora and CentOS5 older

Verify the postfix is installed by default by below command

[root@server ~#]rpm -q postfix
[root@server ~#]yum install postfix (can use this command update to latest if postfix preinstalled)

Step 2: - configure the smtp server

[root@server ~#]vi /etc/postfix/main.cf

in the 75th line, uncomment by removing "#" and specify *your required hostname* (hostname is used with mail id, so email IDs will be look like *user@servercomputing.tech*)
myhostname = servercomputing.tech

#in the 83rd line, uncomment and specify your domain name
mydomain = tech

#in the 99th line, enable it (remove #)
myorigin = $mydomain

#In 116th line, specify interfaces which server allows requests (you can set to all or any specific network interfaces)
inet_interfaces = all

#in the 119th line specify the ip version (ipv4 or ipv6)
inet_protocols = ipv4

#in the 164th line, change like below
mydestination = $myhostname, localhost. $mydomain, localhost, $mydomain

#In the 264th line, enable the line and add your client network (if 192.168.10.0/24)
mynetworks = 127.0.0.0/8, 192.168.10.0/24

#in 418th line, enable the line and edit like below
home_mailbox = Maildir/
#In 545th line, enable the line by removing the "#"

header_checks = regexp:/etc/postfix/header_checks

#In 546th line, add like below

body_checks = regexp:/etc/postfix/body_checks

#In 571th line, add like below

smtpd_banner = $myhostname ESMTP

Step3:

Start/Restart the postfix daemon

[root@server ~#]systemctl restart postfix

Note: you can test by sending an email to local system in case of sending email to real mail server like gmail or yahoo you need a read IP which must be non black listed

OR

Important: for practice purpose you can send or receive mail by mail command in Redhat linux 7

9. Autofs Configuration

The primary configuration file for the autofs is /etc/auto.master, also referred to as the master map. The master map lists autofs-controlled mount points on the system, and their corresponding configuration files or network sources known as automount maps. The format of the master map is as follows:

mount-point map-name options , The variables used in this format are:
mount-point

The autofs mount point e.g /home.

map-name

The name of a map source which contains a list of mount points, and the file system location from which those mount points should be mounted. The syntax for a map entry is described below.

options

If supplied, these will apply to all entries in the given map provided they don't themselves have options specified. This behavior is different from autofs version 4 where options where cumulative. This has been changed to implement mixed environment compatibility.

The following is a sample line from /etc/auto.master file (displayed with cat /etc/auto.master):

/home /etc/auto.misc

The general format of maps is similar to the master map, however the "options" appear between the mount point and the location instead of at the end of the entry as in the master map:

mount-point [options] location
The variables used in this format are:

mount-point

This refers to the autofs mount point. This can be a single directory name for an indirect mount or the full path of the mount point for direct mounts. Each direct and indirect map entry key (mount-point above) may be followed by a space separated list of offset directories (sub directory names each beginning with a "/") making them what is known as a mutli-mount entry

options

Whenever supplied, these are the mount options for the map entries that do not specify their own options.

location

This refers to the file system location such as a local file system path (preceded with the Sun map format escape character ":" for map names beginning with "/"), an NFS file system or other valid file system location.

The following is a sample of contents from a map file (i.e. /etc/auto.misc):

payroll -fstype=nfs personnel:/dev/hda3

sales -fstype=ext3 :/dev/hda4

The first column in a map file indicates the autofs mount point (sales and payroll from the server called personnel). The second column indicates the options for the autofs mount while the third column indicates the source of the mount. Following the above configuration, the autofs mount points will be /home/payroll and /home/sales. The -fstype= option is often omitted and is generally not needed for correct operation.

The autofs will create the directories if they do not exist. If the directories exist before the autofs was started, the autofs will not remove them when it exits. You can start or restart the automount daemon by issuing either of the following two commands:

[root@localhost ~]# systemctl restart autofs

[root@localhost ~]# systemctl status autofs

Using the above configuration, if a process requires access to an autofs unmounted directory such as /home/payroll/2006/July.sxc, the automount daemon automatically mounts the directory. If a timeout is specified, the directory will automatically be unmounted if the directory is not accessed for the timeout period.

You can view the status of the automount daemon by issuing the following command:

[root@localhost ~]# systemctl status autofs

10.NTP Client and Server

The Network Time Protocol (NTP) is used to synchronize a computer's time with another reference time source. Under CentOS / RHEL you can use NTP or OpenNTPD server software. Both package provides client and server software programs for time synchronization.

Install ntp

The ntp package contains utilities and daemons that will synchronize your computer's time to Coordinated Universal Time (UTC) via the NTP protocol and NTP servers. The ntp package includes ntpdate (a program for retrieving the date and time from remote machines via a network) and ntpd (a daemon which continuously adjusts system time). Install the ntp package:

[root@localhost ~]# yum install ntp

How do I configure an NTP Client?

make some entry in below listed configuration file

[root@localhost ~]# vim /etc/ntp.conf

Server 192.168.0.254 # here ip of NTP server

Configure an NTP Server

If you have lots of server and desktop system, configure your own NTP server. Your NTP server contacts a central NTP server,provided by your ISP or a public time server located at ntp.org, to obtain accurate time data. The server then allows other machines on your network to request the time data. Our sample setup:

Step 1.

[root@localhost ~]# yum install ntp

Step 2:

Now open /etc/ntp.conf:

[root@localhost ~]# vim /etc/ntp.conf

restrict default ignore

Above will deny all access to any machine, server or client. However, you need to specifically authorized policy settings. Set it as follows

restrict 202.54.1.5 mask 255.255.255.245 nomodify notrap noquery
server 202.54.1.5

RedHat Certified Engineer II (RHCE)

Table of Contents

1. Grub (GNU Grand Unified Boot loader (GRUB) version 2)

grub version 2 comes with lots of changes if you compare with grub version 1, one more thing when ever you change in your configuration file related to grub every time you need to update the grub.cfg file

Configuring the GRUB 2 Boot Loader: -

To update the GRUB 2 configuration file manually, use the grub2-mkconfig -o command as follows:

[root@redhat7 ~]# grub2-mkconfig -o /boot/grub2/grub.cfg

Important: Here main configuration file is /boot/grub2/grub.cfg but you don't need to touch this file because this update all its setting from two given location

1. /etc/grub.d/ directory

2. /etc/default/grub file

1. list of all files

[root@redhat7 ~]# cd /etc/grub.d/

[root@redhat7 grub.d]# ls

00_header 20_linux_xen 30_os-prober 41_custom

10_linux 20_ppc_terminfo 40_custom README

2. content of all files

[root@redhat7 grub.d]# cat /etc/default/grub

GRUB_TIMEOUT=5

GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"

GRUB_DEFAULT="Red Hat Enterprise Linux Server, with Linux 3.10.0-123.el7.x86_64"

GRUB_DISABLE_SUBMENU=true

GRUB_TERMINAL_OUTPUT="console"

GRUB_CMDLINE_LINUX="rd.lvm.lv=rhel/root vconsole.keymap=us vconsole.font=latarcyrheb-sun16 crashkernel=auto rhgb quiet"

GRUB_DISABLE_RECOVERY="true"

Note 1: Tasks like Passing kernel parameter and changing grub timeout can be managed from /etc/default/grub file as show above file.

Note 2: Tasks like securing grub and making entry of another os like windows can be managed from /etc/grub.d/ directory.

i) Here 10_linux file can be used to secure grub go to last line made some entry as given below

[root@redhat7 grub.d]# vim 10_linux

cat <<EOF

set superusers="ashutoshh"

password ashutoshh redhat

EOF

then update grub.cfg file

[root@redhat7 ~]# grub2-mkconfig -o /boot/grub2/grub.cfg

ii) For making entry for another boot loader

[root@redhat7 grub.d]# vim 40_custom

menuentry "windows 7" {

set root='hd0,msdos1'

chainloader +1

}

then update grub.cfg file

[root@redhat7 ~]# grub2-mkconfig -o /boot/grub2/grub.cfg

2. GPT

Today in the production world size of hard disk is really very big in size which can not be handled by MBR technology because we can only manage hard disk of size 2 TB.

So there is technique GPT partition table which stands for "GUID partition table" and GUID stands for "globally unique identifiers".

Note: MBR works upon -- 32-bit address scheme but GPT works upon 128-bit address scheme.

so when you create partition first of change your partition table style:

For making partition with GPT format you can use any of available tool like "**parted** and **gdisk**"

Important: some points to be remember

1. MBR scheme only can create max 4 primary partition

2. GPT is the scheme where you create unlimited number of primary partition (by default 128) and there is no extended partition.

To scan and which partition table scheme is present follow the below given steps:

[root@redhat7 Desktop]# gdisk /dev/sda

GPT fdisk (gdisk) version 0.8.10

artition table scan:

MBR: MBR only

BSD: not present

APM: not present

GPT: not present

Note: As you can analyze MBR is the scheme which available

Note 2: all the options of fdisk command is working properly in gdisk like p, n, d, w

ii) To check all the partitions

Command (? for help): p

Disk /dev/sda: 625142448 sectors, 298.1 GiB

Logical sector size: 512 bytes

Disk identifier (GUID): 6457798A-B184-4453-8128-AEC7CDBA1136

Partition table holds up to 128 entries

First usable sector is 34, last usable sector is 625142414

Partitions will be aligned on 2048-sector boundaries

Total free space is 18595614 sectors (8.9 GiB)

Number Start (sector) End (sector) Size Code Name

1 2048 411647 200.0 MiB 8300 Linux filesystem

2 411648 102402047 48.6 GiB 8E00 Linux LVM

3 102402048 512002047 195.3 GiB 0700 Microsoft basic data

5 512004096 512393215 190.0 MiB 8300 Linux filesystem

6 520208384 613956430 44.7 GiB 8300 Linux filesystem

7 512395264 512804863 200.0 MiB 8300 Linux filesystem

iii) To create

Command (? for help): n

Partition number (4-128, default 4): 4

First sector (34-625142414, default = 613957632) or {+-}size{KMGTP}:

NOte: As you can analyze that in size option has more options like T, P

So like following these steps to create GPT partition

3. Networking with IPV6

Internet Protocol version 6 (IPv6) is the latest revision of the Internet Protocol (IP) and the first version of the protocol to be widely deployed. IPv6 was developed by the Internet Engineering Task Force (IETF) to deal with the long-anticipated problem of IPv4 address exhaustion. This tutorial will help you in understanding IPv6 and associated terminologies along with appropriate references and examples.

As you all know IPV4 has a 32-bit structure of scheme which means in round figure you only store a number of 2^32 bit long but IPV6 address is made of 128 bits divided into eight 16-bits blocks. Each block is then converted into 4-digit Hexadecimal numbers separated by colon symbol.

For example, the below is 128 bit IPv6 address represented in binary format and divided into eight 16-bits blocks:0010000000000001 0000000000000000 0011001000110100 1101111111100001 0000000001100011 0000000000000000 0000000000000000 1111111011111011

Each block is then converted into Hexadecimal and separated by ':' symbol:

2001:0000:3238:DFE1:0063:0000:0000:FEFB

Even after converting into Hexadecimal format, IPv6 address remains long. IPv6 provides some rules to shorten the address. These rules are:

Rule:1 Discard leading Zero(es):

In Block 5, 0063, the leading two 0s can be omitted, such as (5th block):

2001:0000:3238:DFE1:63:0000:0000:FEFB

Rule:2 If two of more blocks contains consecutive zeroes, omit them all and replace with double colon sign ::, such as (6th and 7th block):

2001:0000:3238:DFE1:63::FEFB

Consecutive blocks of zeroes can be replaced only once by :: so if there are still blocks of zeroes in the address they can be shrink down to single zero, such as (2nd block):

2001:0:3238:DFE1:63::FEFB

Note: Version 6 has slightly complex structure of IP address than that of IPv4. IPv6 has reserved few addresses and address notations for special purposes. See the table below

: :/128 -- unexpected address

: :/0 ---- default route

: :1/128 -- loopback address

1. As shown in the table above 0:0:0:0:0:0:0:0/128 address does not specify to anything and is said to be an unspecified address. After simplifying, all 0s are compacted to ::/128.

2. In IPv4, address 0.0.0.0 with netmask 0.0.0.0 represents default route. The same concept is also applie to IPv6, address 0:0:0:0:0:0:0:0 with netmask all 0s represents default route. After applying IPv6 simplying rule this address is compressed to ::/0.

Loopback addresses in IPv4 are represented by 127.0.0.1 to 127.255.255.255 series. But in IPv6, only 0:0:0:0:0:0:0:1/128 address represents Loopback address. After simplying loopback address, it can be represented as ::1/128.

To add ipv6 address in redhat 7 you can use out of given methods

Method 1: -With the use of ifconfig command

[root@desktop16 ~]# ifconfig enp2s0

enp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 192.168.0.16 netmask 255.255.255.0 broadcast 192.168.0.255

inet6 fe80::beae:c5ff:fec1:5744 prefixlen 64 scopeid 0x20<link>

ether bc:ae:c5:c1:57:44 txqueuelen 1000 (Ethernet)

RX packets 1671723 bytes 1000775554 (954.4 MiB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 2372145 bytes 3021159735 (2.8 GiB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

Note: inet6 fe80::beae:c5ff:fec1:5744 prefixlen 64 this line is for ipv6 and now i am adding one more address

[root@desktop16 ~]# ifconfig enp2s0 inet6 add abcd:123d::23/80

[root@desktop16 ~]# ifconfig enp2s0

enp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 192.168.0.16 netmask 255.255.255.0 broadcast 192.168.0.255

inet6 fe80::beae:c5ff:fec1:5744 prefixlen 64 scopeid 0x20<link>

inet6 abcd:123d::23 prefixlen 80 scopeid 0x0<global>

ether bc:ae:c5:c1:57:44 txqueuelen 1000 (Ethernet)

RX packets 1682227 bytes 1002389460 (955.9 MiB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 2391436 bytes 3048002541 (2.8 GiB)

Note: as you can analyze one more ipv6 address is added

Method 2: - With the use of IP command

i) To check use below given steps

[root@desktop16 ~]# ip addr show dev enp2s0

ii) To change

[root@desktop16 ~]# ip addr add bcad:1287:234a::12ca/64 dev enp2s0

Method 3: - With the use of nmcli

i) To check

[root@desktop16 ~]# nmcli connection show enp2s0 | grep -i ip6

ipv6.ip6-privacy: -1 (unknown)

IP6.ADDRESS[1]: ip = fe80::beae:c5ff:fec1:5744/64, gw = ::

IP6.ADDRESS[2]: ip = bcad:1287:234a::12ca/64, gw = ::

ii) To change

[root@desktop16 ~]# nmcli connection modify enp2s0 ipv6.addresses dacb:234b:8765::43bc/80

4. Networking_with_nmcli

As you used NetworkManager in redhat 6 which was not supported by default, lots of protocol and Modem wifi-device so in redhat 7 a huge enhancedment in NetworkManager program and it is more powerfull than rhedhat 6 for using NetworkManager a great tool known as nmcli will be used.

Note: In redhat 7 you can create your own network profile and activate it as per the requirement and as per place where you are currently.

For doing this we need a preinstalled tool called "nmcli":

1. **To check list of all connection profile use below given command**

[root@redhat7 Desktop]# nmcli connection

NAME UUID TYPE DEVICE

Auto cybertronX a98b3dd0-deec-4981-bc56-0797a73bfca2 802-11-wireless –

Auto cybertronX 426dd791-a058-4caf-b2ca-8fc88cd2e781 802-11-wireless –

Auto CybertroneX 080039c6-0783-498a-9a7b-95d977a6eb9a 802-11-wireless –

newprofile 4e5d4d7c-c002-4392-8608-ae1954613905 802-11-wireless wlp2s0

2. **For Adding a New profile with static IPV4**

[root@redhat7 Desktop]# nmcli connection add con-name cisco1 ifname enp3s0 type ethernet autoconnect yes ip4 192.168.0.200 gw4 192.168.0.254

3. **For Adding a New profile with Dynamic IPV4 allocation**

[root@redhat7 Desktop]# nmcli connection add con-name myadhoc autoconnect yes ifname virbr0 type Ethernet

Note: - Replace ip4 with ip6 and gw4 to gw6 for assigning IPV6 address

4. **Modiy profile**

[root@redhat7 Desktop]# nmcli connection modify virbr0 ipv4.addresses '10.0.2.34/8 10.0.2.254' ipv4.method manual

5. **Up and Down a connection profiles**

[root@redhat7 Desktop]# nmcli connection up docker

Note: here docker is profile name

5. link_aggreation and bonding

Today network load and backup of network like when any link of network is down of cutted off so due to this reason all the network of any organization will shut off so to overcome this problem there are two techniques in the market first is round robin and second is H.A (High availability).

For this purpose, you need to create some network profile and which have to add by nmcli tool

i) To show all existing profile

[root@desktop16 ~]# nmcli connection show

NAME UUID TYPE DEVICE

vnet0 e0537139-fe90-4390-be4a-15255119df50 generic vnet0

virbr0 6d277565-7683-48a2-826c-85207c2fadf6 bridge virbr0

enp2s0 3fa6ba76-1709-4796-a768-6777add0dba9 802-3-ethernet enp2s0

ii) To create profile of team interface

[root@desktop16 ~]# nmcli connection add con-name adhocteam ifname team0 type team config '{"runner" : {"name":"activebackup"}}' ip4 192.168.0.16/24 gw4 192.168.0.254

Note: HERE adhocteam is name network team profile with interface name team0 which will be automatically created

[root@desktop16 ~]# nmcli connection show

NAME UUID TYPE DEVICE

vnet0 e0537139-fe90-4390-be4a-15255119df50 generic vnet0

virbr0 6d277565-7683-48a2-826c-85207c2fadf6 bridge virbr0

enp2s0 3fa6ba76-1709-4796-a768-6777add0dba9 802-3-ethernet enp2s0

adhocteam 1467eddb-1e46-4a4b-935e-5d8e75580738 team team0

Note: As you can see here adhocteam is a new profile created successfully

iii) Now to create a slave profile using real LAN CARD enp2s0

[root@desktop16 ~]# nmcli connection add con-name myteam1 ifname enp2s0 type team-slave master team0

Connection 'myteam1' (92abd88a-c5d5-4b52-94d3-a9bc45b37bf6) successfully added.

iv) Now to create another slave profile using real LAN CARD enp2s0:1

[root@desktop16 ~]# nmcli connection add con-name myteam2 type team-slave ifname enp2s0:1 master team0

Connection 'myteam2' (1cb87429-7a3b-4b3e-a378-b87b945e4d4b) successfully added.

[root@desktop16 ~]# nmcli connection show

NAME UUID TYPE DEVICE

myteam1 92abd88a-c5d5-4b52-94d3-a9bc45b37bf6 802-3-ethernet --

vnet0 e0537139-fe90-4390-be4a-15255119df50 generic vnet0

virbr0 6d277565-7683-48a2-826c-85207c2fadf6 bridge virbr0

enp2s0 3fa6ba76-1709-4796-a768-6777add0dba9 802-3-ethernet enp2s0

adhocteam 1467eddb-1e46-4a4b-935e-5d8e75580738 team team0

myteam2 1cb87429-7a3b-4b3e-a378-b87b945e4d4b 802-3-ethernet --

Now: UP all the three profiles

a) [root@desktop16 Desktop]# nmcli connection up myteam1

b) [root@desktop16 Desktop]# nmcli connection up myteam2

c) [root@desktop16 Desktop]# nmcli connection up adhocteam

6. DHCP SERVER in RHEL 7

For assigning IP address there are two methods

1. Static method
2. Dynamic method

Today we are talking about dynamic method of IP address allocation

DHCP: - dhcp is a protocol for assigning IP address to computer system, mobile phones, router etc by dynamic method OR we can say every device which requires IP address.

DHCP: - stands for “Dynamic Host Configuration protocol “

Configuring dhcp server in Redhat 7

Important: - As you all know that we use three steps for making a server.

Step 1: - check the required software and install it if not installed by using below given steps

[root@localhost ~]# rpm -q dhcp # Here rpm command for checking dhcp package

Dhcp is not installed

[root@redhat7 Desktop]# yum install dhcp

Step 2: configure the dhcp server using configuration file

[root@redhat7 Desktop]# cd /etc/dhcp/

[root@redhat7 dhcp]# ls

dhclient.d dhcpd6.conf dhcpd.conf

Note: here dhcpd.conf is the main configuration file for dhcp server when you open the

file some instruction are given use them if you required or use below configuration steps

For copying manual file you can use below given **cp** command

[root@redhat7 dhcp]# cp /usr/share/doc/dhcp4-2.1/dhcpd.conf.sample /etc/dhcp/dhcpd.conf

OR

Write done necessary configuration

[root@redhat7 dhcp]# vim dhcpd.conf

default-lease-time 600;

max-lease-time 7200;

subnet 20.0.0.0 netmask 255.0.0.0 {

range 20.0.0.10 20.0.0.100;

}

Note: To fix IP of every computer system which is allocated by dhcp you have to bind every

system with their mac (physical address)

host redhat71 {

hardware ethernet 08:00:07:26:c0:a5;

fixed-address 20.0.0.50;

}

host redhat61 {

hardware ethernet 20:CF:30:F3:EB:14;

fixed-address 20.0.0.51;

}

Step 3: start or restart the service of dhcp

[root@redhat7 dhcp]# systemctl restart dhcpd

Important: - To make dhcp service persistant after rebooting your server use below command like in redhat version 6 we use (chkconfig dhcp on)

[root@redhat7 dhcp]# systemctl enable dhcpd

Important: - If during ip allocation for clients if there is any problem then add a firewall rule for dhcp server port:

Dhcp server port: 67

Dhcp Client port : 68

[root@redhat7 dhcp]# firewall-cmd --add-port=67/udp

OR

[root@redhat7 dhcp]# firewall-cmd --add-service=dhcp

Now your dhcp server ready to use.

Dhcp client configuration: -

For dhcp client there is a command known as **dhclient** by using this command you can assign IP address to your system use given method.

```
[root@redhat7 ~]# dhclient  -v  enp2s0
Internet Systems Consortium DHCP Client 4.2.5
Copyright 2004-2013 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/enp2s0/20:cf:30:f2:f4:02
Sending on   LPF/enp2s0/20:cf:30:f2:f4:02
Sending on   Socket/fallback
DHCPDISCOVER on enp2s0 to 255.255.255.255 port 67 interval 6 (xid=0x2807414d)
DHCPREQUEST on enp2s0 to 255.255.255.255 port 67 (xid=0x2807414d)
DHCPOFFER from 20.0.0.2
DHCPACK from 20.0.0.2 (xid=0x2807414d)
bound to 20.0.0.13 -- renewal in 281 seconds.
[root@redhat7 ~]#
```

6. DNS SERVER in RHEL 7

DNS stands for **Domain name system** which is used for resolving your hostname to IP address or IP address to hostname.

There are lots of programe to create your own dns in Redhat linux but we will use BIND programe to create own dns server

STEP 1. use below step to install the software known as bind

[root@redhatv7 ~]# yum install bind

STEP 2. To know the configuration file use below given commad

[root@redhatv7 ~]# rpm -qc bind

/etc/logrotate.d/named

/etc/named.conf

/etc/named.iscdlv.key

/etc/named.rfc1912.zones

/etc/named.root.key

/etc/rndc.conf

/etc/rndc.key

/etc/sysconfig/named

/var/named/named.ca

/var/named/named.empty

/var/named/named.localhost

/var/named/named.loopback

Note: Here /etc/named.conf is the main configuration file

Important: make a backup of main configuration file using below command

[root@redhatv7 ~]# mv /etc/named.conf /etc/named.conf.bak

Important: make your own configuration file and make some necessary entry.

[root@redhatv7 ~]# vim /etc/named.conf

options {

directory "/var/named/";

};

zone "ashu.com" IN {

type master;

file "mdb";

};

Here ashu.com is the name of domain whose record you want to create in you DNS server

Follow given steps to make your own zone file.

Note 1 : Now change your directory to "/var/named/" using below steps

[root@redhatv7 ~]# cd /var/named/

[root@redhatv7 named]# ls

data named.ca named.localhost slaves

dynamic named.empty named.loopback

Note 2: make your zone file for database of DNS using below steps

[root@redhatv7 named]# cp named.localhost mdb # In place of mdb you can choose any file name

Note: 3 change group of mdb file from root to named.

[root@redhatv7 named]# chgrp named mdb

Note: 4 make entry into your zone file to make your database of DNS

Important: Do not delete any thing which is previously present in DNS file just add the given content

[root@redhatv7 named]# vim mdb

ashu.com NS redhat7.ashu.com.

redhat7.ashu.com. A 192.168.0.5

fedora19.ashu.com. A 192.168.0.154

newlinux CNAME fedora19.ashu.com.

Important: above last two lines shows that a particular domain has a particular

i). A stands for (Address record) to resolve name to IP

ii). CNAME stands for (canonical name record) to call a host from other name like aliass

iii). PTR stands for (pointer record) you can also use in zone file which resolve the IP to name.

3. start the service of named

[root@redhatv7 named]# systemctl restart named

Now your Dns has been configured you can use by making your self as DNS client.

DNS CLIENT: - In redhat 6 and 7 you can manage from /etc/resolv.conf

7. Configure targetcli in Redhat 7

Motive behind iscsi protocol is to send hard disk (block device) over the network In redhat 7. we are going to setup a targetcli.

Step 1: To setup target you need to install software which is in version 7 (targetcli)

[root@redhat7 Desktop]# yum install targetcli

step 2: There is no need to start the service or open configuration file so use targetcli command

[root@redhat7 Desktop]# targetcli

targetcli shell version 2.1.fb34

Copyright 2011-2013 by Datera, Inc and others.

For help on commands, type 'help'.

/> # this is the prompt of targetcli

i) To list the architecture of iscsi target

/> ls

o- / …… [...]

o- backstores …… [...]

| o- block …… [Storage Objects: 0]

| o- fileio …… [Storage Objects: 0]

| o- pscsi …… [Storage Objects: 0]

| o- ramdisk …… [Storage Objects: 0]

o- iscsi …… [Targets: 0]

o- loopback …… [Targets: 0]

/>

ii) To create a iscsi target block

/> cd /backstores/block

/backstores/block>

/backstores/block> create newhd /dev/sda7

Created block storage object newhd using /dev/sda7.

/backstores/block>

iii) To create an IQN number use below stpes

/backstores/block> cd /iscsi

/iscsi>

/iscsi> create # create command will automatically IQN number

Created target iqn.2003-01.org.linux-iscsi.redhat7.x8664:sn.66955d1bd65c.

Created TPG 1.

/iscsi>

OR you can give your own IQN number in proper format

/iscsi> create iqn.2014-10.com.example:iscsit

Created target iqn.2014-10.com.example:iscsit.

Created TPG 1.

/iscsi>

Targetcli Client:

1. Install the required software

   [root@redhat7 Desktop]# yum install iscsi-initiator-utils

2. **Discover the target**

[root@redhat7 Desktop]# iscsiadm --mode discoverydb --type sendtargets --portal 192.168.0.200 –discover

Note: - **Here 192.168.0.200 Is the IP of targetcli server after executing command successfully you will find IQN number successfully like given below**

192.168.0.200:3260,1 iqn.2014-07.com.adhoc:tgt1

Login to targetcli server:

[root@redhat7 Desktop]# iscsiadm --mode node --targetname iqn.2014-07.com.adhoc:tgt1 --portal 192.168.0.200:3260 –login

Note: now you can create new partition of new hard disk

iV) Now you need to configure the iscsi hard disk for client with given properities like acls , luns , portals

/iscsi> cd /iscsi/iqn.2014-10.com.example:iscsit/tpg1/

/iscsi/iqn.20…e:iscsit/tpg1>

/iscsi/iqn.20…e:iscsit/tpg1> ls

o- tpg1 …….. [no-gen-acls, no-auth]

o- acls …….. [ACLs: 0]

o- luns …….. [LUNs: 0]

o- portals …….. [Portals: 0]

A) Create a portal so that client can connect and access hard disk

/iscsi/iqn.20…e:iscsit/tpg1> portals create

Using default IP port 3260

Binding to INADDR_ANY (0.0.0.0)

Created network portal 0.0.0.0:3260.

/iscsi/iqn.20…e:iscsit/tpg1>

Note : IT will automatically create 3260 port binded with all ip of ISCSI server

Important: You can create you won port and also can bind with any server IP using below step

/iscsi/iqn.20…e:iscsit/tpg1> portals create 192.168.0.20 1234

Created network portal 192.168.0.20:1234.

/iscsi/iqn.20…e:iscsit/tpg1>

B) You also need to create a Logical unit number or Luns for client hard disk

/iscsi/iqn.20…e:iscsit/tpg1> luns/ create /backstores/block/newhd

Created LUN 0.

/iscsi/iqn.20…e:iscsit/tpg1>

C) By default acls must be implemented in version 7. To two disable acls use below given steps

i)

/iscsi/iqn.20…fb47f492/tpg1> set attribute authentication=0

Parameter authentication is now ’0′.

/iscsi/iqn.20…fb47f492/tpg1>

ii)

/iscsi/iqn.20…fb47f492/tpg1> set attribute generate_node_acls=1

Parameter generate_node_acls is now ’1′.

/iscsi/iqn.20…fb47f492/tpg1>

9. Configure Apache with TLS

As you all know web traffic is no more secure without encryption which is likely to be says without https so to configure https in apache web server follow given steps

step 1: - Install the required software

[root@desktop16 Desktop]# yum install mod_ssl

Step 2: - It will automatically create and configure a new file which can be shown below

[root@desktop16 Desktop]# cd /etc/httpd/

[root@desktop16 httpd]# cd conf.d/

[root@desktop16 conf.d]# ls

autoindex.conf README ssl.conf userdir.conf welcome.conf

[root@desktop16 conf.d]#

Note: - here conf.d is the directory ssl.conf file is created under that file every thing is written about ca,private key , crt .

Step 3: - start the service of apache (httpd) so that it will start the secure port which is 443

[root@desktop16 httpd]# systemctl restart httpd

Step 4: - add the firewall rule

[root@desktop16 conf.d]# firewall-cmd --add-service=https

10. Configure NFS with Kerberos in Redhat 7

NFS (Network File System) As you all know and well familiar fo directory sharing in Network file system but the problem with NFS is there is no security among client and servers because they are working in plain text. To secure both the hosts client and server we are introducing a protocol authentication server known as Kerberos.

TO configure NFS server and client there are steps given below:

There are three system required;

1. kerberos server IP - 192.168.0.254

2. NFS Server IP - 192.168.0.10

3. NFS client IP - 192.168.0.20

Note: - In redhat version 7 kerberos will already configured and which is alos not a part of RHCE you only need to configure your both machine as client and server but here i am also giving steps of Kerberos.

For every node of kerberos client make sure your dns and ntp clients are working properly.

A) Configure the NFS server and make it kerberos client:

Step 1: First make kerberos client

A) Install kerberos client side software

[root@desktop10 Desktop]# yum install krb5-workstation pam_krb5

B)

[root@desktop7 home]# authconfig-tui

Note: here click on use kerberos and make entry of kerberos Realm and IP or hostname of KDC which will be given

step 2: Install the required software of NFs server

[root@desktop10 Desktop]# yum install nfs-utils

step 3: Configure the NFS exported file

Edit the /etc/exports file and add the option sec=krb5

[root@desktop10 Desktop]# vim /etc/exports

/etc *(rw,sec=krb5)

Note:

The sec option accepts four different values: sec=sys (no Kerberos use), sec=krb5 (Kerberos user authentication only), sec=krb5i (Kerberos user authentication and integrity checking), sec=krb5p (Kerberos user authentication, integrity checking and NFS traffic encryption). The higher the level, the more you consume resources.

Step 4: Copy the file from kerberos using any method under /etc directory

Note: For hosts kerberos create a file whree authentication keys are store so Both server client of NFS need to copy their own file and put the to inside there /etc directory and name must be **/etc/krb5.keytab**

Step 4: start the secure service of NFS

[root@desktop10 Desktop]# systemctl restart nfs-secure-server

Step 5: If you know the firewalld make entry of firewall rules other wise stop the firewall by

[root@desktop10 Desktop]# systemctl stop firewalld

NFS client and also kerberos client:

step 1: make it kerberos client using above given steps and also copy the kerberos keyfile in nfs client side named /etc/krb5.keytab

Step 2: Install the required software

[root@desktop10 Desktop]# yum install nfs-utils

Step 3: Start the client side daemon

[root@desktop10 Desktop]# systemctl restart nfs-secure

[root@desktop10 Desktop]# systemctl restart nfs-idmap

Step 4: Now mount the directory in secure medium

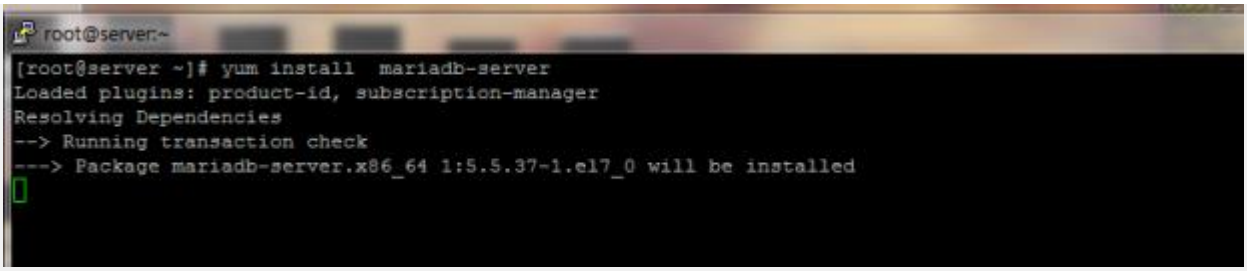
[root@desktop10 Desktop]# mount -o sec=krb5 192.168.0.10:/etc /mnt

Now you can enjoy the NFs with secure medium

11. Mariadb Server

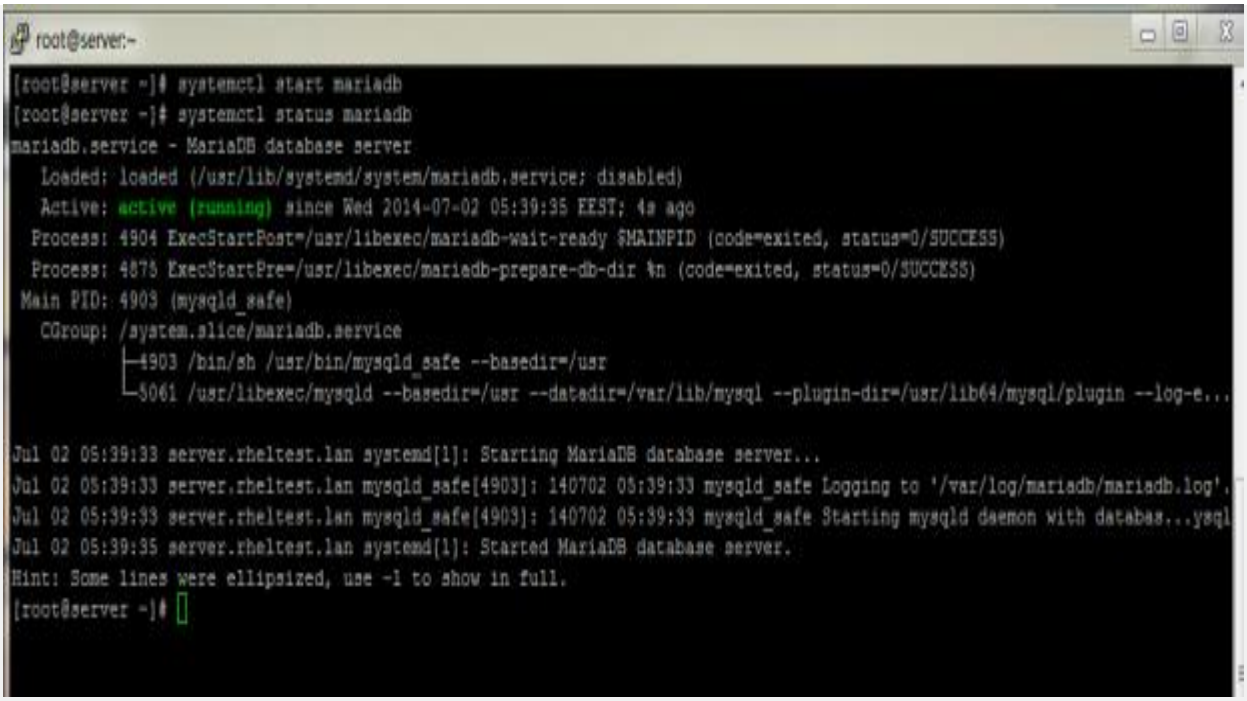
Red Hat Enterprise Linux/CentOS 7.0 switched from MySQL to MariaDB for its default database management system. To install MariaDB database use the following command.

```
# yum install mariadb-server mariadb
```

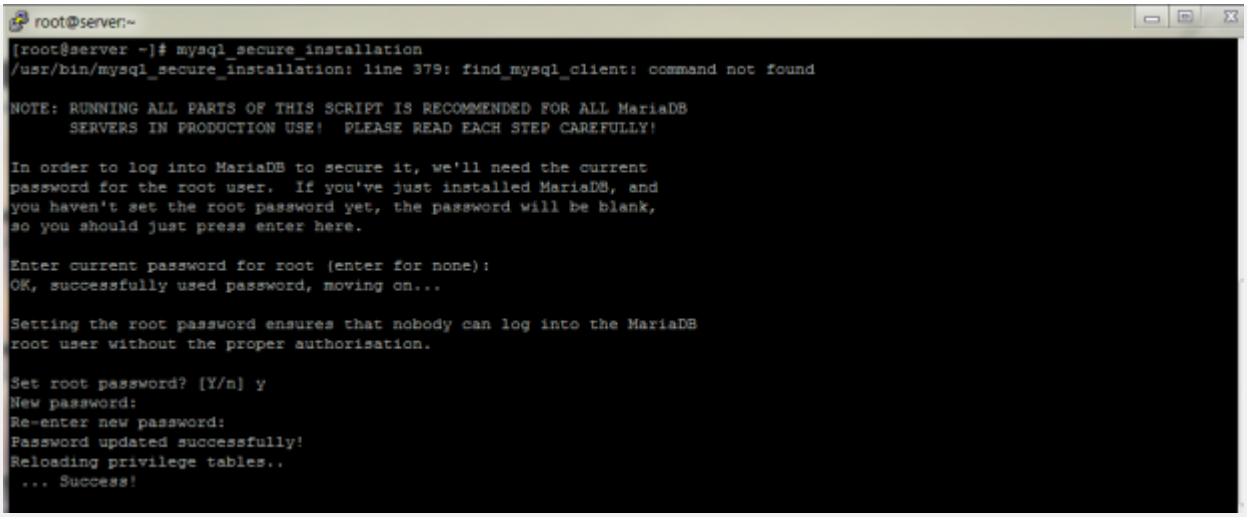


After MariaDB package is installed, start database daemon and use mysql_secure_installation script to secure database (set root password, disable remotely logon from root, remove test database and remove anonymous users).

```
# systemctl start mariadb
```



```
# mysql_secure_installation
```



To test database functionality login to MariaDB using its root account and exit using quit statement.

```
mysql -u root -p


MariaDB > SHOW VARIABLES;


MariaDB > quit
```