BSC-IT

SEM -5 Linux System Administration

NOV-2018 Q.P.Code: 57839

Q1 Attempt any three of the following: (15)

a) Explain piping and Redirecting with proper example. Write a command to print first three lines of the file (5)

Piping:

• The piping is among the most powerful features of the Linux command line.
• Piping executes a command and send the output of that command to the next command so that it can do something with it.
• For example, First you'll execute a command where the output doesn't fit on the screen. Next, by piping this output through less, you can see the output screen by screen.
1. Type the command ps aux. This command provides a list of all the processes that are currently running on your computer. Usually, the list doesn't fit on the screen.
2. To make sure you can see the complete result page by page, use ps aux | less. The output of ps is now sent to less, which outputs it so that you can browse it page by page.
• Another useful example is that, if you want to check whether a user with the name ABC exists in the user database /etc/passwd , we can issue a command cat /etc/passwd | grep ABC
• Here we are piping the contents of the file to the filter grep, which would select all of the lines that contain the string mentioned as an argument of grep.

Redirection:

• Redirection is also one of the powerful features of Linux Command line.
• Redirection sends the output of a command to a file.
• This file doesn't necessarily need to be a regular file, but it can also be a device file.
• For example,
1. From a console window, use the command ps aux. You'll see the output of the command on the current console.
2. Now use ps aux > ~/psoutput.txt. You don't see the actual output of the command, because it is written to a file that is created in your home directory, which is designated by the ~ sign.
3. To show the contents of the file, use the command less ~/psoutput.txt.
• Use ">>" instead of ">" symbol if you do not want to overwrite the content of existing file.
• For example, who > myfile will put the result of the who command (which displays a list of users currently logged in) in a file called myfile.
• If then you want to append the result of another command, for example the free command (which shows information about memory usage on your system), to the same

file myfile, then use free >> myfile.

b) What are the different duties of Linux System Administrator? (5)

Duties of Linux System Administrator

Various duties of Linux System Administrator include:

- Installing and configuring servers
- installing and configuring application software
- Creating and maintaining user accounts
- Backing up and restoring files
- Monitoring and tuning performance
- Configuring a secure system
- Using tools to monitor security

- **Installing and configuring servers**
  - This server runs even on a standalone machine with one user account and it must be configured.
  - Printing in Linux also takes place only after configuring a print server.
  - So these servers need to be installed by the Linux system administrator.
  - But there is no need of servers to use web services, remote FTP access and emailing.
  - Security of the server is also important when it is connected to the machines in outside network.
  - So the various services in the servers are turned off unless specifically enabled and configured.
  - This is the responsibility of the Linux system administrator to know which servers are needed and how to employ them.
- **Installing and Configuring Application Software**
  - Each user has an account on the system.
  - The applications installed by specific users in their home directories will only be available for those users without system administrator's intervention.
  - If the application is to be used by multiple users, it need s to be installed higher up in the Linux File Hierarchy.
  - This job is performed by System administrator only.
  - The administrator can limit which users may use which applications by creating a “group” for that application and enrolling individual users into that group.
  - New software packages might be installed in /opt (This directory is reserved for all the software and add-on packages that are not part of the default installation), if they are likely to be upgraded separately from the Red Hat Linux distribution itself.
  - Some packages may need to go in /usr/local or even /usr
  - /usr is the largest directory on a Linux system, and some people like to have it on a separate partition) , if they are upgrades of packages installed as part of Red Hat Linux.
  - In Red Hat Package Manager (RPM) packages automatically goes where it should.

- **Creating and Maintaining user accounts**
  - In Linux machine an account must be created for each user and system administrator may do this.
  - System administrator can give option to users to change their own password.
  - In red hat enterprise, System administrator can configure a setting to prompt user that they must change password periodically.
  - System Administrator can stop person from accessing account.
  - System administrator can make restriction on web surf like user can surf web but has access to particular sites.
  - System administrator can stop certain websites.
  - Under Linux, every file and program must be owned by a user.
  - Each user has a unique identifier called user Id (UID).
  - Each user must belong to at least one group, a collection of users established by the system administrator.
  - Users may belong to multiple groups and identified by their group IDs (GID).
  - The accessibility of files or programs is based on its UIDs and GIDs.
- **Backing up and restoring files**
  - Backup required to secure data from computer failure, some people may harm other's property or system if system administrator not perfect to handle it.
  - Important files are back-up so that in the event of a failure of hardware, security, or administration, the system can be up and running again with minimal disruption.
  - Backup can be taken in high capacity tape drive and can be stored in disks.
  - System Administrator must decide what to back up and how frequently to take the backup.
  - System Administrator may maintain a series of incremental backups i.e. adding only the files that have changed since the last backup or multiple full backups.
  - System Administrator may use RAID (redundant array of independent disks) for Backup , which is multiple hard drives all containing the same data as insurance against the failure of any one of them, in addition to other backup systems.
  - System administrator should carry restore at least once in non critical time.
  - Need to formulate a plan for bringing the system back up in the event of a failure.
  - System administrator can prevent hardware failure by properly configuring the devices and files
- **Monitoring and Tuning Performance**
  - System tuning is an ongoing process aided by a variety of diagnostic and monitoring tools.
  - Some performance decisions are made at installation time, while others are added later.
  - System administrator does following things to monitor and tune the performance:
    • Proper monitoring
    • Careful system monitoring and diagnostic practices

• Careful system monitoring plus wise use of the built-in configurability of Linux

- **Configuring Secure Systems**
  - The system administrator's first and foremost task is to make certain that no data on the machine or network are likely to become corrupted, whether by hardware or power failure, by mis-configuration or user error or by malicious or inadvertent intrusion from elsewhere.
  - The security can be as simple as to turning off unneeded services, monitoring the Red Hat Linux security mailing list to make sure that all security advisories are followed, and otherwise engaging in good computing practices to make sure the system runs robustly.
  - The system administrator can increase security by :
  - Setting the security permissions
  - Elaborating firewall to protect not only Linux system, but the other nonLinux systems connected to it.
  - Hardening against attacks
  - Making sure that the passwords are strong enough not to be guessed easily.
- **Using Tools To Monitor Security**
  - The Linux development community is quick to find potential exploits and to find ways of slamming shut the door before crackers can enter by making available new, patched versions of packages in which potential exploits have been found.
  - First and best security tool is making sure that whenever a security advisory is issued, you download and install the repaired package.
  - Various distros of Linux provide various tools to system administrator, using which the unauthorized access can be detected and blocked.

c) Explain find command with following options: -name, -user,-exec,-type (5)

find command:

• Finding files is another useful task you can perform on your server.
• find command is used for this purpose.
• This is a very powerful command that helps you find files based on any property the file may have ,such as their names; the access, creation, or modification date; the user who created them; the permissions set on the file; and much more.
• For example, you want to find all files whose name begins with hosts, use find / -name "hosts*“
• find / -user “ABC" to locate all files created by user ABC.

find command with options:

-name: for example (find . -name testfile.txt) it find a file called testfile.txt in current and sub-directories.

-user: for example (find /home -user exampleuser -mtime -7 -iname ".db") it find all .db files (ignoring text case) modified in the last 7 days by a user named exampleuser.

-exec: it is used to execute a command from the bash itself. This command does not create a new process it just replaces the bash with the command to be executed. If the exec command is successful, it does not return to the calling process.

-type: for example (find . -type f -empty) it find an empty file within the current directory.

d) What are different commands for Process Management? (5)

Process Management :

Any application that runs on a Linux system is assigned a process ID or PID. This is a numerical representation of the instance of the application on the system. In most situations this information is only relevant to the system administrator who may have to debug or terminate processes by referencing the PID. Process Management is the series of tasks a System Administrator completes to monitor, manage, and maintain instances of running applications.

• Every command that we start from the shell is managed as a job.

• Every job that we start is not only a job but also a process.

• In addition, when the server boots, many other processes are started to provide services on server.

• These are the daemons, which are processes that are always started in the background and provide services on your server.

• If, for instance, your server starts an Apache web server, this server is started as a daemon.

• Managing processes is an important task for a system administrator.

• Few commands to manage and monitor processes on Linux system are:

ps: Used to show all current processes

kill: Used to send signals to processes, such as asking or forcing a process to stop

pstree: Used to get an overview of all processes, including the relationship between parent and child processes

killall: Used to kill all processes, based on the name of the process

top: Used to get an overview of current system activity

e) Explain the concept of hardlink and symbolic link. Write a command to create hard link and symbolic link. (5)

• It is very useful to be able to access a single file from different locations.
• In a Linux file system, you can use links for this purpose.
• A link appears to be a regular file, but it's more like a pointer that exists in one location to show you how to get to another location.
• There are two different types of links in Linux:
– A symbolic link
– A hard link

Symbolic Link:
• A symbolic link is the most flexible link type you can use.
• It points to any other file and any other directory, no matter where it is.
• There is a difference between the original file and the link.
• If you remove the original file, the symbolic link won't work anymore and thus is invalid.

Hard Link:
• A hard link can be used only to point to a fie that exists on the same device.
• A hard link is more like an additional name you give to a file.
• To get to a file, the file system reads the file's inode in the file system metadata, and from there it learns how to access the block where the actual data of the file is stored.
• With hard links, you only need the original filename to create the hard link.
• Once it has been created, it isn't needed anymore, and the original filename can be removed.

Commands to create symbolic link:
• The ln command is used to create link.
• Use the option -s to create a symbolic link.
• First you put the name of the original file directly after the ln command.
• Next you specify the name of the link you want to create.
• For example,
Use the command ln -s /etc/hosts ~/ users. This creates a symbolic link with the name users in your home directory.

Commands to create Hard link:
• The ln command is used to create link.
• First you put the name of the original file directly after the ln command.
• Next you specify the name of the link you want to create.
• For example,
Use the command ln /etc/hosts ~/ users. This creates a symbolic link with the name users in your home directory.

f) Write a note on RPM and YUM in Linux. (5)

RPM:

- RPMs, the basic package format that is used for software installation.
- RPM stands for Red Hat Package Manager.
- An RPM is basically an archive file.
- It is created with the cpio command.
  (cpio is a general file archiver utility and its associated file format. cpio is a tool for creating and extracting archives, or copying files from one place to another. It handles a number of cpio formats as well as reading and writing tar files. Cpio stands for "copy in, copy out".)
- However, it's no ordinary archive.
- Along with the packages archived with RPM, there is also metadata describing what is in the package and where those different files should be installed.
- Because RPM is so well organized, it is easy for an administrator to query exactly what is happening in it.
- Another benefit of using RPM is that its database is created in the /var/lib/rpm directory.
- This database keeps track of the exact version of files that are installed on the computer.
- Thus, for an administrator, it is possible to query individual RPM files to see their contents.

YUM:

- To standardize software, many programs used on Linux use libraries and other common components provided by other software packages.
- Which means to install package A, package B is required to be present.
- This is known as a software dependency.
- If suppose an administrator who wants to install a given package downloaded from the Internet.
- It's possible that in order to install this package, the administrator would first have to install several other packages.
- This would be indicated by the infamous "Failed dependencies" message.
- Meta Package Handler is a solution to the problem of software dependency.
- Meta Pack age Handler, which in Red Hat is known as yum (Yellowdog Update Manager), works with repositories, which are the installation sources that are consulted whenever a user wants to install a software package.
- In the repositories, all software packages of your distribution are typically available.
- While installing a software package using command yum install somepackage, yum first checks to see whether there are any dependencies.
- If there are, yum checks the repositories to see whether the required software is available in the repositories, and if it is, the administrator will see a list of software that yum wants to install as the required dependencies.
- So yum provides the solution for software dependency problem.

```
[root@hnl ~]# yum install nmap
```

Loaded plugins: product-id, refresh-packagekit, security, subscription-manager

Updating certificate-based repositories.

Setting up Install process

Resolving Dependencies

--> Running transaction check

---> Package nmap.x86_64 2:5.21-4.el6 will be installed

--> Finished Dependency Resolution

Dependencies Resolved

- For using yum utility, we may first have to create and manage the repository.

Q2 Attempt any three of the following: (15)

a) What are different kinds of partitions available in Linux? (5)

- Two popular command-line tools are used to create partitions on RHEL.

– fdisk tool

– parted tool.

- Creating a partition with fdisk is easy to do.
- After starting fdisk, you simply indicate that you want to create a new partition.
- You can create three kinds of partitions:

– Primary Partitions: These are written directly to the master boot record of your hard drive. After creating four primary partitions, you can't add any more partitions—even if there is still a lot of disk space available. There's space for just four partitions in the partition table.

– Extended Partition: Every hard drive can have one extended partition. You cannot create a file system in an extended partition. The only thing you can do with it is to create logical partitions. You'll use an extended partition if you intend to use more than four partitions on a hard drive.

– Logical Partitions: A logical partition is created inside an extended partition. You can have a maximum of 11 logical partitions per disk, and

you can create file systems on top of logical partitions.

• After selecting between primary, extended, or logical partitions, you need to

select a partition type.

• This is an indication to the operating system what the partition is to be used for.

• On RHEL servers, the following are the most common partition types:

– **83:** This is the default partition type. It is used for any partition that is

formatted with a Linux file system.

– **82:** This type is used to indicate that the partition is used as swap space.

– **05:** This partition type is used to indicate that it is an extended partition.

– **8e:** Use this partition type if you want to use the partition as an LVM

physical volume.

• You need to restart your server to activate the new partition.

b) What are different File Systems available on RHEL? (5)

File Systems:

• Once you are done creating one or more partitions, you put a file system on them.
• Several file systems are available on Red Hat Enterprise Linux, but Ext4 is used as the default file system.
• Ext4 provides a very important feature i.e. journaling.
• The journal works as a transaction log in which the file system keeps records of files that are open for modification at any given time.
• The benefit of using a file system journal is that, if the server crashes, it can check to see what files were open at the time of the crash and immediately indicate which files are potentially damaged.
• There is one drawback to using a journal, however: a file system journal takes up disk space—an average of 50MB normally on Ext4.
• Following are different file systems:

Ext2/3 :

• The predecessors of the Ext4 file system. Ext2 doesn't use a file system journal, and therefore it
is a good choice for very small partitions (less than 100MB).
• ext3 provides all the features of ext2, and also features journaling and backward compatibility with ext2.
• The backward compatibility enables you to still run kernels that are only ext2 aware with ext3 partitions. You can upgrade an ext2 file system to an ext3 file system without losing any of your data.

• ext3's journaling feature speeds up the amount of time it takes to bring the file system back to a normal state if it's not been cleanly unmounted (that is, in the event of a power outage or a system crash).
• Under ext2, when a file system is uncleanly mounted, the whole file system must be checked. This takes a long time on large file systems. ext3 keeps a record of uncommitted file transactions and applies only those transactions when the system is brought back up.
• ext3's journaling feature involves a small performance hit to maintain the file system transaction journal.
• Therefore, it's recommended that you use ext3 mostly for your larger file systems, where the ext3 journaling performance hit is made up for in time saved by not having to run fsck on a huge ext2 file system.

Ext4 :

• The default file system on RHEL. It is a general-purpose file system. Ext4 uses file system journaling feature. The file system journal works as a transaction log in which the file system keeps records of files that are open for modification at any given time.
• The benefit of using a file system journal is that, if the server crashes, it can check to see what files were open at the time of the crash and find the damaged files
• The drawback of using a journal is that it takes up disk space. For example, an average of 50MB normally on Ext4. That means it's not a good idea to create a journal on very small file systems because it might leave insufficient space to hold your files.

XFS :

• Provides good performance for very large file systems and very large files.

Btrfs :

• Btrfs is the next generation of Linux file system. It is based on B-tree database, which makes the file system faster. It also has features like Copy on Write, which makes it very easy to revert to a previous version of a file. This file system is easy to grow and shrink. Btrfs is currently available as a tech preview version only, which means that it is not supported and not yet ready for production.

VFAT and MS-DOS :

• Sometimes it 's useful to put files on a USB drive to exchange them among Windows users. This is the purpose of the VFAT and MS-DOS file systems.

GFS :

GFS is Red Hat 's Global File System. It is designed for use in high availability clusters where multiple nodes need to be able to write to the same file system simultaneously.

c) Write a short note on runlevels and services in Linux. (5)

- Many services are typically offered in a RHEL environment.
- A service starts as your server boots.
- The exact services start-up process is determined by the runlevel in which the server boots.
- The runlevel defines the state in which the server boots.
- Every runlevel is referenced by number.
- Given are the different runlevels in Linux:

Runlevel	Meaning
0	Halt - Machine will get shutdown (Do NOT set initdefault to this)
1	Single user mode (Maintanance mode)
2	Multiuser, without NFS (The same as 3, if you do not have networking)
3	Full multiuser mode (GUI will not be here)
4	Unused (Reserved)
5	X11 (Full multiuser mode with GUI)
6	Reboot-Machine will get reboot (Do NOT set initdefault to this)

- In each runlevel, service scripts are started.
- These service scripts are installed in the /etc/init.d directory and managed with the service command.
- The service script starts, either from its configuration file in the /etc directory or from a configuration file that it uses, which is stored in the /etc/sysconfig directory.
- To manage service scripts, two commands are used:
  - The service command: which you can use to start, stop, and monitor all of the service scripts in the /etc/ init.d directory.
  - The chkconfig command, which you can use to enable/disable the service in the runlevel.

1. Open a root shell, and use cd to go to the directory /etc/init.d. Type ls to get a list of all service scripts currently in existence on your server.
2. Type service ntpd status. This should tell you that the ntpd service currently stopped.
3. Type service ntpd start to start the ntpd service. You'll see the message starting ntpd, followed by the text [OK] to confirm that ntpd has started successfully.
4. At this moment, you've started ntpd, but after a reboot it won't be started automatically. Use chkconfig ntpd on to add the ntpd service to the runlevels of your server.
5. To verify that ntpd has indeed been added to your server's runlevels, type chkconfig --list .This command lists all services and their current status. If you want, you can filter the results by adding grep ntpd to the chkconfig --list command.

```
[root@hnl ~]# chkconfig –list
NetworkManager   0:off   1:off   2:on   3:on   4:on   5:on   6:off
Abrt-ccpp        0:off   1:off   2:off  3:on   4:off  5:on   6:off
Abrt-oops        0:off   1:off   2:off  3:on   4:off  5:on   6:off
```

d) What are different steps to enable SSH server on RHEL? (5)

• The Secure Shell (SSH) protocol is the default service that is used to obtain remote access to a server.

• To use SSH, you need an SSH server and an SSH client.

• SSH server is a process that runs on your server.

• On most Linux distributions, the name of this process is sshd.

• To connect to it from a client computer, the ssh client utility is used, if the client is Linux.

• PuTTY is used if the client is Windows based.

Enabling the SSH Server:

• The SSH service is installed on the Red Hat Enterprise Linux server.

• It isn't enabled by default,

• To start it manually use the service sshd start command.

• To make sure that it is also started after a reboot of your server by use chkconfig sshd on.

• After performing these tasks, you can first do a basic connection test and connect to it using the ssh command

Steps to Enabling the SSH Server:

1. From a terminal with root permissions, use the command service sshd start, h the unlikely event that this command shows an error, use yum install openssh-server to install the ssh server package.

2. Use the chkconfig sshd on command to enable the SSH service, and add it to your server's runlevels. This ensures that the SSH server also comes up after rebooting the server.

3. Now it's time to test the SSH server Open a new terminal window, and use ssh root@localhost to open an SSH session where you're logging in as root. Enter the password when prompted

4. You're now in an SSH session. In this example, you tested the connection from your own local machine. You can also test the connection from a remote machine. This will be discussed further in the sections that follow.

5. Type exit to close the SSH sessions.

• An SSH server that has been enabled with all the default settings isn't a secure SSH server.

• To make the SSH server secure, there are at least two modifications to be made to the /etc/ssh/sshd_config file:

– the Port setting

– the AllowRootLogin parameter.

• **Port** : By default, SSH listens on port 22. Every hacker knows this. So if you're directly connected to the Internet, change the SSH port to something less obvious. I like putting it on port 444, for example.

• **PermitRootLogin**: By default, this parameter allows the user root to log in to your SSH server. This is not a good idea. If root is permitted to log in, the potential hacker only has to guess the root password. It's better to switch off root login by giving this parameter the value no. This means you'll have to connect as an ordinary user, and once connected, you'll have to use su - to escalate your privileges to the root level.

• **AllowUsers**: This parameter is not there in sshd_config by default. Everyone should use it and add a list of only those users you want to allow to log in to your SSH server. This makes it really hard for hackers, because they will have to guess the name of that user before starting the evil work.

1. Open a root shed on your server, and use the commands useradd linda and useradd lisa to add two users to your server. Next set the password for these users to pass-word by using passwd linda and passwd lisa.

2. Use vi /etc/ssh/sshd_config to open the sshd configuration file.

3. Change the Port parameter, and give it a value 443. Next set the PermitRootLogin

parameter value to no, and add the parameter AllowUsers, giving it the value linda.

4. close the vi editor using the :wq! command, and restart the sshd process using service sshd restart.

5. Connect as root on SSH port 443 using ssh -p 443 root@localhost. Access should be denied. Try to connect as lisa using ssh -p 443 lisa@localhost. Access should also be denied. Now try to connect as linda using ssh -p 443 linda@localhost. You should be granted access

e) Explain the function of passwd command with its options. (5)

• To access the system, a user needs a password.
• By default, login is denied for the users you create, and passwords are not assigned automatically.
• To enable these users, assign passwords using the passwd command.
• The passwd command is easy to use.
• A user can use it to change his password.
• If that happens, the passwd command will first prompt for the old password and then for the new one.
• The root user can change passwords as well.
• To set the password for a user, root can use passwd followed by the name of the user whose password needs to be changed.
• For example, passwd ABC would change the password for user ABC.
• The user root can use the passwd command in three generic ways:
  – for password maintenance—to change a password, for example.
  – to set password expiry information, which dictates that a password will expire at a particular date (-e).
  – for account maintenance. For example, an administrator can use passwd to lock an account so that login is disabled temporarily (-l).

Performing Account Maintenance with passwd:

• In an environment where many users are using the same server, it is important to perform some basic account maintenance tasks.
• These include:
  – locking accounts when they are not needed for a long time
  – unlocking an account
  – reporting the password status
• An administrator can also force a user to change their password on first use.
• To perform these tasks, the passwd command has some options available as:
  – -l : Enables an administrator to lock an account. For example, passwd -l ABC will lock the account for user ABC.
  – -u : Unlocks an account that has been locked before.
  – -S : Reports the status of the password for a given account.

– -e : Forces the user to change their password on next login.

Managing Password Expiry:

• In a server environment, it makes sense to change passwords occasionally.

• The passwd command has some options to manage account expiry, which are as follows.

– -n min : This rarely used option is applied to set the minimum number of days that a user must use their password. If this option is not used, a user can change their password at any time.

– -x max : This option is used to set the maximum number of days a user can use a password without changing it.

– -c warn : When a password is about to expire, you can use this option to send a warning to the user. The argument for this option specifies the number of days before expiry of the password that the user will receive the warning.

– -i inact : Use this option to expire an account automatically when it hasn't been used for a given period of time. The argument for this option is used to specify the exact duration of this period.

Apart from the passwd command, you can also use chage to manage account expiry.

/ etc/ passwd

• The most important, of all user-related configuration files is /etc/passwd.

• This file is the primary database where user information is stored.

• That is, the most important user properties are stored in this file.

• Different fields are used in /etc/passwd.

• The fields are separated with a colon.

• Following is the short explanation of all the fields used in /etc/passwd:

– Username : The user's login name is stored in the first field in /etc/passwd.

– Password: this field contains 'x' i.e. nothing. Earlier unix OS used to store passwords in this file. but this file is accessible to everyone. So in modern Linux, this field contains nothing.

– UID : this field stores the unique UID for local user. Typically the highest number that is used is 60000 (the highest numbers are reserved for special-purpose accounts).

– GID : Every user has a primary group. The group ID of this primary group is listed there.

– GECOS : The General Electric Comprehensive Operating System (GECOS) field is used to include some additional information about the user. The field can contain anything you like, such as the department where the user works, the user's phone number, or anything else. This makes identifying a user easier for an administrator.

– Home: Directory This field points to the directory of the user's home directory.

– Shell: The last field is used to refer to the program that is started automatically when a user logs in. Most often, this will be /bin/bash.

f) What are the commands used for managing group membership? Write a command for creating two groups and change current group assignment to other existing user. (5)

• There are three commands to manage the groups in your environment:

groupadd, groupdel, and groupmod.

• The basic structure for using the groupadd command is simple: groupadd

somegroup,

where somegroup is the name of the group you want to create.

• When creating groups, the first available group ID (GID) is assigned automatically.

• If you want to specify the GID yourself, you can use the option –g.

• groupmod can be used to make a user a member of some group.

• groupdel is used to delete the group.

• All groups on your system are defined in the configuration file /etc/group.

• To manage group membership, you can use the usermod and groupmod

commands.

• The usermod command provide three different choices to manipulate group

membership.

– -g, --gid : GROUP This option is used to set the primary group for the user.

– -G, --groups: GROUPS Use this option to define a new list of

supplementary groups. Notice that this option replaces the old list of

supplementary groups with the list of new supplementary groups defined

here.

– -a, --append : Use this option together with -G to add new supplementary

groups to the current list of supplementary groups.

Q3 Attempt any three of the following: **(15)**

a) What are firewalls? How it protects the server? **(5)**

Firewall:

• When a hacker breaks through the security wall of a server, it is done by using ports that are actually allowed on the server.

• For instance, the hacker can abuse an Apache web server and have it launch a script that opens connections to external machines.

• A firewall can be used to make sure that no connections are initiated to nodes that haven't specifically been allowed beforehand.

• A firewall works through pack et inspection.

• This means that the firewall screens incoming and outgoing packets to check whether the address, protocol, and port of the packet is either allowed or denied.

• Firewalls works on layer 3,4 & 5 as per OSI RM.

• For additional security of the server, a firewall should be installed on it.

• Netfilter is the default firewall offered through the Linux kernel.

• To configure Netfilter on RHEL, you can use the system-config-firewall tool as a GUI tool.

• The iptables command can be used if you want to work from the command line.

steps to verify the status of the firewall:

• Given are the steps to verify the current status of firewall:

• From the GNOME graphical inter face, select System Administration Firewall.

• Review the warning that tells you that all current configurations will be overwritten, and click Close.

• From the list of trusted services, select DNS, FTP, SSH, and WWW, and click Apply to save the configuration.

• Set ting Up a Firewall with system-config-firewall

• Close system-config-firewall, and open a shell prompt.

• Type chkconfig | grep iptables. This command will display the current status of the iptables service in the runlevels on your server. It should read as follows:

[root@hnl ~]# chkconfig | grep iptables

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

• If the iptables service that implements your firewall isn't listed as being on in runlevels 2, 3, 4, and 5, use chkconfig iptables on to enable it.

• Type service iptables status. This command shows that the current status of iptables is enabled.

• Type iptables -L -v. You'll see a list that displays all of the firewall rules.

b) What are tables, chains, and rules? List common elements of rule. (5)

Tables:

- Tables are the basic building blocks of a Netfilter firewall.
- Specific tables can be created to stipulate the specific functionality of the firewall.
- By default, the filter table is used.
- The NAT table is also frequently used.

Chains:

- A table contains chains.
- A chain consists of a set of rules that is sequentially processed for each packet that enters the firewall until it finds a match.
- The default strategy is "exit on match," which means that the firewall looks no further once the first rule that matches for a specific packet is found.
- The following chains are used in the filter table:
  - o INPUT: Incoming packets are processed in this chain.
  - o OUTPUT: This chain is used for outgoing packets.
  - o FORWARD : This chain is used on routers that forward the packets.
- Of these chains, a server administrator must understand how to use INPUT and OUTPUT.
- Configuring the FORWARD chain is not as important for the server as it is for routers .

Composition of Rules:

- Different elements can be used to specify properties of a packet in each rule.
- Some elements are mandatory, whereas some other elements are used to make the rule more specific.
- Given is the list of elements used in composing rules:

- **Modules:** A module is an optional element that you can use in a rule to enhance the Netfilter firewall. They do that by loading a specific kernel module that adds functionality. A very common module in iptables rules is the state module, which looks at the state of a packet.
- **Interface:** On a server with multiple network cards, it makes sense to apply rules to specific interfaces only. However, if you're configuring your firewall on a typical server with one network card only, you can safely omit the interface specification.
- **IP Addresses:** In a rule, you can allow or deny access to specific IP addresses or IP network addresses.
- **Protocol:** Most rules allow or deny access to specific ports. These ports are always connected to the UDP or TCP protocol. Therefore, if you want to state a specific port, you also need to indicate the protocol that is to be used.
- **Target :** The target is also a mandatory component in a rule. A target specifies what needs to be done with a matching packet. Different targets can be used, of which ACCEPT, DROP, REJECT, and LOG are the most important.

common elements of rule:

- Apart from these common elements found in rules, you need to specify the purpose of the rule.
- There are two options: either you can set a policy or you can add a rule to a chain.
- A policy defines the default behavior.
- If no specific rule is found in the chain that matches an incoming packet, the policy is applied.
- It is always good practice to define a policy that denies all access.
- After defining a policy, you can start working on the chains.
- This typically means you'll append or insert rules in the chain.
- It is very important to be aware that order does matter.
- That is, if you use -A to append a rule, it is entered at the last position in the chain.
- However, if you use -I to insert a rule, you can specify where exactly in the chain you want to insert it.

c) Write a short note on Certificate Authority. (5)

Certificate Authority:

- The use of public/private keys is a great improvement in Internet security.
- There is a challenge of: how can the receiver be sure that the public key that it receives actually comes from a trusted server and not from a hacker who has hijacked the connection?
- This is where the role of CA comes in.
- A CA is used to guarantee the authenticity of a public key.
- The role of the CA is to sign PKI (Public Key Infrastructure) certificates.
- Any server can generate a PKI certificate, and it is the role of the CA to sign these PKI certificates with its private key.

- However, this is useful only if the client that receives the certificate knows the public key of the CA.
- If this is not the case, users will see a message indicating they are using an untrusted connection, and the client application will probably close the connection.
- Thus, for common use on the Internet, make sure that the CA is known to everyone.
- For private internal use, an in-house CA can also be used.

The Trusted Root

- If you want to create your own CA, make sure the users who use it can trust it.
- You can accomplish this by having its certificates signed by a commonly known CA.
- Because the public keys of these commonly known CAs are available in most client applications, the CA that uses it will be transparently accepted.
- The only drawback is that, in general, you need to pay money to the CA that is going to sign your certificates.
- If you don't want or don't need to do that, you can use a self-signed certificate.

Running Your Own CA

- If you do that, you can have its certificates signed by a trusted root.
- Alternatively, you can use self-signed certificates.
- This is the "you can trust me because I say so" type of certificate.
- It's not the kind of security you want to show your customers on the Internet, but it works well for internal use.

d) What are different steps to encrypt and decrypt files? (5)

Encrypting & Decrypting Files with GPG:

• GPG is commonly used to encrypt files.
• The base command to do this is : gpg –e yourfile.
• The gpg command will next ask for a user ID.
• This is the ID of the user to which you want to send the encrypted file.
• This must be a user who is already in your GPG keyring.
• Enter the name of each user for whom you want to encrypt the file on a separate line.
• When you're done, just press Enter on an empty line.
• The receiver of the encrypted file can decrypt it by using the command gpg -d.
• To send it to a new file, make sure to use redirection when specifying the target file.
• The command gpg -d myfile.gpg > myfile will extract the contents of the GPG encrypted file to myfile.
• There is one requirement for extracting a file that has been encrypted with gpg that the user who decrypts the file needs to enter their passphrase to do this.

e) What is NFS? What are advantages and disadvantages of NFS? (5)

NFS :

• NFS, the Network File System, is the most common method for providing file sharing services on Linux and Unix networks.

• It is a distributed file system that enables local access to remote disks and file systems.

• If you have the appropriate network connection, you can access files and directories that are physically located on another system or even different city or country using Linux commands.

• NFS uses a standard client/server architecture.

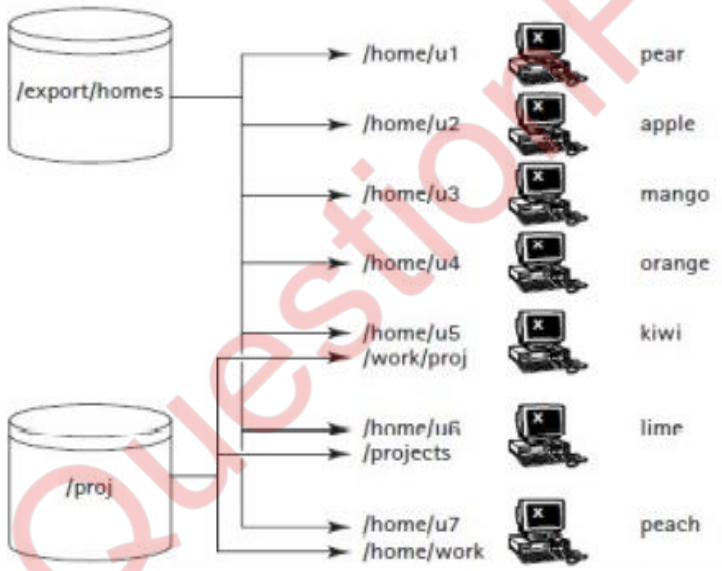
• The server portion consists of the physical disks containing shared file systems and several daemons that make the shared file systems (or entire disks, for that matter) visible to and available for use by client systems on the network.

• This process is normally referred to as exporting a file system.

• NFS clients simply mount the exported file systems known as NFS mounts on their local system just as they would mount file systems on local disks.

• Many sites store users home directories on a central server and use NFS to mount the home directory when users log in or boot their systems.

• In this case, the exported directories must be mounted as /home/username on the local (client) systems, but the export itself can be stored anywhere on the NFS server, say, /exports/users/ username.



Advantages of NFS:

• **Centralized Administration :** NFS stores the file system on a central server. It is much easier, to back up a file system stored on a server than it is to back up /home directories scattered throughout the network. Administration tasks also becomes easy when the files are centrally stored.

• **Easy to update :** It becomes simple to update key configuration files when NFS is used with NIS (Network Information System),because it provide access to shared disk space, or limit access to sensitive data.

• **Conservation of Disk Space :** NFS can also conserve disk space and prevent duplication of resources because file systems that change infrequently or that are usually read-only, such as /usr, can be exported as read-only NFS mounts.
• **Easy up gradation of applications :** It is easy to upgrade applications employed by users throughout a network because it simply becomes a matter of installing the new application and changing the exported file system to point at the new application.
• **Easy logins :** It becomes easier for the end users to log in from any system when NFS is combined with NIS, because users can login even remotely, and still have access to their home directories and see a uniform view of shared data.
• **Security :** Users can protect important or sensitive data that would be impossible or time consuming to recreate by storing it on an NFS mounted file system that is regularly backed up.

Disadvantages of NFS:
• **Heavy network traffic :** NFS is sensitive to network congestion i.e. heavy network traffic slows down NFS performance.
• **Heavy disk activity :** Heavy disk activity on the NFS server badly affects NFS's performance. NFS clients may also run slowly because disk reads and writes take longer time.
• **Security Concerns :** NFS experiences potential security problems because its design assumes a trusted network. NFS implementation based on protocol version 1, 2, and 3 is that they are based on standard (unencrypted) remote procedure calls (RPC).
• **Data unavailability in case of crash :** Data becomes unavailable, if the disk or system exporting vital data or application becomes unavailable due to a disk crash or server failure. No one can access that resource.
• **Never use NFS version 3 and earlier on systems that front the internet** because it becomes easier for packet sniffer to intercept and interpret the data and information.

f) what are steps for setting up samba file server? (5)

Samba:
• Computers running Windows 95 or greater use a protocol called Server Message Block (SMB) to communicate with each other and to share services such as file and print sharing.
• Using a program called Samba, you can emulate the SMB protocol in Linux and connect your Red Hat Network to a Windows network to share files and printers.
• The Common Internet File System (CIFS) is such a protocol, and it is offered by the Linux Samba server.
• The Linux PC icon appears in the Windows Network Neighborhood window, and the files on the Linux PC can be browsed using Windows Explorer.
• The windows file system can be mounted on Linux system and can browsed from Linux PC.

Steps for setting up samba file server:

Step 1: Create a directory on the Linux file system and grant the appropriate permissions to this
directory.

```
# mkdir /sambafiles
# chmod 777 /sambafiles
```

Step 2: Install the Samba packages

```
# yum -y install samba*
```

Step 3: Create the share in Samba.

• Open the file /etc/samba/smb.conf with an editor.

```
# vi /etc/samba/smb.conf
```

• Locate the workgroup parameter, and change it to workgroup = MYSAMBA.

• Go to the bottom of the configuration file, and add the following share configuration:

```
[sambafiles]
comment = samba files
path = /sambafiles
writable = yes
valid user = lucky
```

Step 4: check the syntax of the samba configuration file.

```
#testparm
```

Step 5: Create a Samba account, which makes the Samba server accessible for the Samba users.

```
# useradd lucky
```

• Don't set the password, because Samba users don't need a password on the Linux system.

```
# smbpasswd -a lucky
```

• The above command create Samba user lucky.

Step 6: (re)start the Samba service and make sure that it starts on every server boots.

```
# service smb restart
# chkconfig smb on
```

Q4 Attempt any three of the following: (15)

a) Write a short note on DNS and its hierarchy. (5)

• Domain Name System (DNS) is the system that associates hostnames with IP addresses.

• Because of DNS, users and administrators don't have to remember the IP addresses of computers to which they want to connect but can do so just by entering a name, such as www. example.com.

• To resolve the IP addresses, DNS hierarchy plays an important role.

The DNS Hierarchy:

• DNS is a worldwide hierarchical system.

• In each DNS name, you can see the place of a server in the hierarchy.

• In a name like www.example.com, three parts are involved.

• First, there is the top-level domain (T LD) .com.

• This is one of the top-level domains that have been established by the Internet Assigned Numbers Authority (IANA), the organization that is the ultimate authority responsible for DNS naming.

• Other common top-level domains are .org, .gov, .edu, .mil, and the many top-level domains that exist for countries, such as .uk, .ca, .in, .cn, and .nl.

• Currently, the top-level domain system is changing, and a proposal has been released to make many more top domains available. Each of the top-level domains has a number of name servers.

• Name servers are the servers that have information on the hosts within the domain.

• The most important piece of information that the name servers of the top-level domain have is that relating to the domains that exist within that domain (the subdomain), such as redhat.com, example.com, and so on.

• The name servers of the top-level domains need to know how to find the name servers of these second-tier domains.

• Within the second-tier domains, subdomains can also exist, but often this is the level where individual hosts exist.

• Think of hostnames like www.example.com, ftp.redhat.com, and so on.

• To find these hosts, the second-tier domains normally have a name server that contains resource records for hosts within the domain, which are consulted to find the specific IP address of a host.

• The root domain is at the top of the DNS hierarchy.

• This is the domain that is not directly visible in DNS names but is used to connect all of

the top-level domains together.

• Within DNS, a name server can be configured to administer just the servers within its domain.

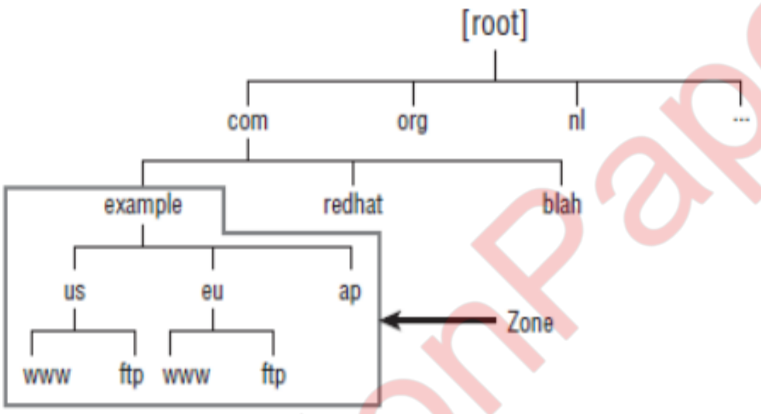
• Often, a name server is also configured to administer the information in subdomains.

• The entire portion of DNS for which a name server is responsible is referred to as a zone.

• There are a few subzones under example.com in this hierarchy.

• This does not mean that each of these subzones needs to have its own name server.

• In a configuration such as this, one name server in the example.com domain can be configured with resource records for all the subzones as well.



• It is also possible to split subzones.

• This is referred to as the delegation of subzone authority.

• This means a subdomain has its own name server, which has resource records for the subdomain.

• In addition, the name server of the parent domain does not know which hosts are in the subdomain.

• This is the case between the .com domain and the example .com domain

b) What are different parameter used for dhcp.conf command? (5)

• The Dynamic Host Configuration Protocol (DHCP) is used to assign IP-related configuration to hosts in your network.

• Using a DHCP server makes managing a network a lot easier.

• To set up a DHCP server, after installing the dhcp package, you need to change common DHCP settings in the main configuration fi le: /etc/dhcp/dhcpd.conf.

• After installing the dhcp package, the file is empty, but there is a good annotated example file in /usr/share/doc/dhcp-<version>/dhcpd.conf.sample.

parameter used for dhcp.conf command:

Some of the most relevant parameters from the dhcpd.conf file and a short explanation of each is:

• **option domain-name:** Use this to set the DNS domain name for the DHCP clients.

• **option domain-name-servers:** This specifies the DNS name servers that should be used.

• **default-lease-time:** This is the default time in seconds that a client can use the IP address that it has received from the DHCP server.

• **max-lease-time:** This is the maximum time that a client can keep on using its assigned IP address. If within the max-lease-time timeout it hasn't been able to contact the DHCP server for renewal, the IP address will expire, and the client can't use it anymore.

• **log-facility:** This specifies which syslog facility the DHCP server uses.

• **subnet:** This is the essence of the work of a DHCP server. The subnet definition specifies the network on which the DHCP server should assign IP addresses. A DHCP server can serve multiple subnets.

• **range:** This is the range of IP addresses within the subnet that the DHCP server can assign to clients.

• **option routers:** This is the router that should be set as the default gateway.

• After giving appropriate values to above parameters, Start the DHCP server by using the command service dhcpd start, and enable it using chkconfig dhcpd on.

• Start the second virtual machine. Make sure that the network card is set to get an IP address from a DHCP server. After starting it, verify that the DHCP server has indeed handed out an IP address.

c) Discuss MTA and MDA in detail. (5)

MTA:

Message Transfer Agent

• The MTA uses the Simple Mail Transfer Protocol (SMT P) to exchange mail messages with

other MTAs on the Internet.

• If a user sends a mail message to a user on another domain on the Internet, it's the responsibility of the MTA to contact the MTA of the other domain and deliver the message there.

• Upon receiving a message, the MTA checks whether it is the final destination.

• If it is, it will deliver the message to the local message delivery agent (MDA), which takes care of delivering the message to the mailbox of the user.

• If the MTA itself is not the final destination, the MTA relays the message to the MTA of the final destination.

• If, for some reason, the MTA cannot deliver the message to the other MTA, it will queue it.

• Queuing means that the MTA stores the message in a local directory and will try to deliver it again later.

• As an administrator, you can flush the queues, which means that you can tell the MTA to send all queued messages now.

• Upon delivery, it sometimes happens that the MTA, which contacted an exterior MTA and delivered the message there, receives it back.

• This process is referred to as bouncing.

• In general, a message is bounced if it doesn't comply with the rules of the receiving MTA, but it can also be bounced if the destination user simply doesn't exist.

• Generally MTA is configured to generate an error if the message couldn't be delivered.

MDA:

Mail Delivery Agent

• Upon receiving a message, the MTA delivers it at the mail delivery agent.

• This is the software component that takes care of delivering the mail message to the destination user.

• The MDA delivers mail to the recipient's local message store, which by default on RHEL is the directory /var/spool/mail/$USER.

• It is common for users to get their messages from a remote desktop on which they are working.

• To facilitate this, you need a POP server that allows users to download messages or an IMAP server that allows users to connect to the mail server and read the messages while they're online.

d) Explain following parameter for secure internet configuration: myhostname, mydomain, myorigin, inet_interfaces, mynetwork. (5)

- There are a few more steps to take to configure a mail server, which is going to handle messages from the Internet.
- You need to make sure your mail server has at least a minimum level of protection against spam and other email abuses. To make a secure Internet configuration, you need to set some additional parameters.
- All of these will be set in the /etc/postfix/main.cf file.
- The following are the relevant parameters:
  - **myhostname :** This parameter specifies the name of this host. If not specified, it is set to the full DNS domain name (FQDN) of this host. This parameter is used as a variable in other parameters in the main.cf file, so it is useful to set it.
  - **mydomain :** This parameter specifies the domain of this host. If not set, the domain name part of the FQDN is used.

- **Myorigin:** This parameter determines the domain seen by the email recipient when receiving messages. The default is to use the FQDN of this host. This means that if user ABC on server xyz.example.com sends a message, the recipient will see a message coming in from ABC@xyz.example.com.
- **inet_interfaces:** This parameter specifies the IP addresses of the mail server to which it binds. By default, it is set to localhost only, which means that our mail server cannot receive messages from the Internet. Use inet_interfaces = all for sending mail to external users.
- **Mynetwork:** This parameter is optional. We can use it to specify the network address from which our MTA accepts messages for relaying without further authentication.

e) Write down the steps to configure Apache for basic website services. (5)

- LAMP is amongst the most common uses of Linux.
- LAMP stands for Linux, Apache, MySQL, and PHP.
- Apache web server is used to provide web services.

Configuring the Apache Web Server

- Apache is one of the most used services on Red Hat Enterprise Linux.
- A basic installation of an Apache website is easy to perform, but by using modules you can make Apache as sophisticated as you need.

Creating a Basic Website

- Configuring an Apache server that services just one website is not hard to do—you just have to install the Apache software and create some content in the Apache document root.
- The default document root is set to /var/www/html on a Red Hat Enterprise Linux server.
- Just put a file in this directory with the name index.html, and it will be served by your Apache server.
- Steps of creating a basic website are:
  - Use yum -y install httpd to install the Apache web server.
  - Use chkconfig httpd on to put the Apache web server in your server's runlevels, and have it start at boot in your runlevels.
  - Open a root shell, and go to the directory /var/www/html. In this directory, create a file with the name index.html. In this file, put the content " welcome to my website" and then use service httpd start to star t the Apache web server.

- Still from the root shell, use elinks http://localhost to access the website you just created. If it works properly, that means your web server is up and running.

f) Write a short note on virtual host. (5)

- One Apache installation can handle more than one Apache website.
- To handle more than one site from an Apache server, you can create virtual hosts.
- A virtual host is a definition of different websites to be served by the Apache web server.
- You can include this definition in the main Apache configuration file /etc/httpd/conf/httpd.conf or in separate files that you will create in the /etc/httpd/conf.d/ directory.
- If you chose the later solution, make sure the name of each of these files ends in .conf.
- When setting up virtual hosts, you have to choose which type to use.
- You can configure following types of virtual hosts:
  - **a name-based virtual host:** Name-based virtual hosts are the default, and they are easier to set up because you can run multiple Apache sites on one IP address.
  - **an IP-based virtual host:** IP-virtual hosts are often used if SSL is needed on a website, because in SSL it is beneficial if a connection can be traced back to its original unique IP address. So, you must set up IP-based virtual hosting to get SSL working.
- Following is an example of setting virtual host:

<VirtualHost *:80>

ServerAdmin webmaster@dummy-host.example.com

DocumentRoot /www/docs/dummy-host.example.com

ServerName dummy-host.example.com

ErrorLog logs/dummy-host.example.com-error_log

CustomLog logs/dummy-host.example.com-access_log common

</VirtualHost>

- The file definition starts by defining the port on which the virtual host should listen.
- This is set to *:80, which means that it is available on any IP address on this machine on port 80.

- The ServerAdmin directive is used to tell users whom to contact if they need to get in touch with the administrator of this web server.
- The most important parameter that is used when defining a virtual host is a DocumentRoot that is specific for that virtual host.
- In this example, a DocumentRoot in /www is used.
- Next the ServerName is specified, and as the last bit of the configuration, some dedicated log files for this virtual host are defined.

Q5 Attempt any three of the following: (15)

a) What are different elements of a shell script? (5)

• A shell script is a text file that contains a sequence of commands.

• Anything that can run a bunch of commands is considered a shell script.

• There are some rules to ensure that you create quality shell scripts.

• The scripts that not only work well for the task for which they are written but that also will be readable by others.

Elements of a Good Shell Script:

• When writing a script, make sure it meets the following elements:

• Has a unique name. If your script has the same name as an existing command, the existing command will be executed and not your script, unless you prefix the name of the script with a backslash (/) character. You can find out whether the name of your script already exists by using the which command. For example, if you want to use the name hello and want to be sure that it's not in use already, type which hello.

• Includes the shebang (#!) to tell the shell which subshell should execute the script. The shebang always starts with #! and is followed by the name of the subshell that should execute the script.

• Includes comments

• Uses the exit command to tell the shell executing the script that it has executed successfully. This command exits the script and then tells the parent shell how

the script has executed. If the parent shell reads exit 0, it knows the script has executed successfully. If it encounters anything other than exit 0, it knows that there was a problem. In more complex scripts, you could even start working with different exit codes like exit 1, exit 2 etc.

• Is executable

• A simple shell script may look like:

Type the following code, and save it with the name hello in your home directory.

```
#!/bin/bash
# this is the hello script
# run it by typing ./hello in the directory where you've found it
clear
echo hello world
exit 0
```

b) Write a script that create directory with a name Mumbai, sets $user and $group as the owner of directory and change permission mode to 770. (5)

```
# !/bin/bash
#
# dirscript
# sets $ USER and $ GROUP as the owners of the directory and change the permission mode to 770

DIRECTORY = $ 1
USER = $ 2
GROUP = $ 3
mkdir /$ Mumbai
Chown $ USER $ Mumbai
Chgrp $ GROUP $ Mumbai
Chmod 770 $ Mumbai
exit 0
```

c) Write a short note on high-availability cluster requirement. (5)

High-Availability Clustering:

• A cluster is a system comprising two or more computers or systems (called nodes) which work together to execute applications or perform other tasks, so that users who use them, have the impression that only a single system responds to them, thus creating an illusion of a single resource (virtual machine).

• A high availability cluster is a group of hosts that act like a single system and provide continuous uptime.

• High availability clusters are often used for load balancing, backup and failover purposes.

• To properly configure a high-availability (HA) cluster, the hosts in the cluster must all have access to the same shared storage.

High-Availability cluster Requirements:

• Several components are involved in setting up an HA cluster.

• Basically, two servers are connected to each other by a network cable.

• Other components are:

• **Multiple nodes:** A typical cluster uses at least two nodes, with a maximum number of 16 nodes. More nodes are deployed to ensure the high availability of a large number of services.

• **Quorum:** It is also an important element in an HA cluster. It is the minimum number of nodes that must be available to continue offering services.
Typically, the quorum consists of half-plus-one nodes. That is, a node needs majority in the cluster to be able to continue offering services.

• **Fence Devices:** To maintain the integrity of the cluster, a failing node must be stopped at all times. A fencing device is typically a hardware device, which is available to shut down another node, even if the node itself has failed.
Common solutions include power switches or integrated management cards,

such as HP ILO or Dell DRAC.

• **A Dedicated Cluster Interface:** To make sure that cluster traffic is not interrupted by anything—for example, a spike in the number of packets sent on the user network—a dedicated cluster network is often created. This ensures that cluster packets will always get through no matter what happens on the user network.

• **Ethernet Bonding:** A network card can always fail. To minimize the impact of this and to get more bandwidth, Ethernet bonding is often used in HA cluster environments. An Ethernet bond is a logical device that groups at least two network cards. The frames are distributed across all network interfaces involved in the bond. However, if one of the interfaces goes down, the other interface is capable of handling all the traffic.

• **Shared Storage:** Most services that run in your cluster will need access to files. iSCSI and fiber channel based Storage area networks are used for this purpose.

d) What are the different steps to create bond device. (5)

• To set up a bonded network interface, you'll have to follow these steps:

a. Identify the physical network cards that you want to configure in the bonding interface.

b. Change the configuration for the physical network cards to make them slaves for the bonding

interface.

c. Create a configuration file for the bonding interface.

d. Make sure the bonding kernel module is loaded.

a. Identify the physical network cards that you want to configure in the bonding interface.

• Assume, identified network card is em1 and em2. These are going to be used in a bonded configuration.

b. Change the configuration for the physical network cards to make them slaves for the bonding

interface.

• Make sure that configuration file of em1 and em2 has following:

DEVICE=em1 (or em2 for second network card)

BOOTPROTO=none

ONBOOT=yes

MASTER=bond0

SLAVE=yes

USERCTL=no

c. Create a configuration file for the bonding interface.

• The bonding interface devices will use names bond0, bond1, and so on, and the configuration file

will be /etc/sysconfig/network-scripts/ifcfg-bond0 for bond device bond0 and so on.

• Configuration file for bond0

DEVICE=bond0

IPADDR=192.168.1.100

PREFIX=24

ONBOOT=yes

BOOTPROTO=none

USERCTL=no

BONDING_OPTS="mode=1 miimon=100"

d. Make sure the bonding kernel module is loaded.

• To do this, create a file with the name /etc/modprobe.d/bonding.conf, and put in the line alias

bond0 bonding.

• After performing all of these steps, your bond device is ready for use. Restart the network, and

use the ip a command to verify the working.

e) Write a short note on TFTP server package. (5)

Setting Up a TFTP and DHCP Server for PXE Boot:

• After setting Network Installation Server, PXE boot is configured.

• Short for Pre-Boot Execution Environment.

• PXE allows networked computers that are not yet loaded with an operating system to be configured and booted remotely by an administrator.

• The PXE code is typically delivered with a new computer on a read-only memory chip or boot disk that allows the computer (a client) to communicate with the network server so that the client machine can be remotely configured and its operating system can be remotely booted.

• Two steps are involved in configuring PXE boot:

1. You need to install a TFTP server and have it provide a boot image to PXE clients.

2. You need to configure DHCP to talk to the TFTP server to provide the boot image to PXE clients.

Installing the TFTP Server :

• Install the TFTP server package using yum -y install tftp-server.

• TFTP is managed by the xinetd service.

• So to tell xinetd that it should allow access to TFTP, you need to open the /etc/xinetd.d/tftp file and change the disabled parameter from Yes to No.

```
service tftp
{
        socket_type             = dgram
        protocol                = udp
        wait                    = yes
        user                    = root
        server                  = /usr/sbin/in.tftpd
        server_args             = -s /var/lib/tftpboot
        disable                 = yes
```

per_source = 11

cps = 100 2

flags = IPv4

}

• Next, restart the xinetd service using service xinetd restart.

• Also make sure to start tftp at booting, using chkconfig tftp on.

f) Write a short note on kick-start file. (5)

Creating a Kickstart File:

• Once an environment is created where everything you need to install your server is available on another server, there is no need to work with optical discs to perform an installation.

• But you still need to answer all the questions which are part of the normal installation process.

• Red Hat offers an excellent solution for this challenge: the kickstart file.

Using a Kickstart File to Perform an Automated Installation :

• When you install a Red Hat system, a file with the name anaconda-ks.cfg is created in the home directory of the root user.

• This file contains most settings that were used while installing your computer.

• To specify that you want to use a kickstart file to install a server, you need to tell the installer where to find the file.

• To use a kickstart file in an automated installation from a TFTP server, you need to add the kickstart file to the section in the TFTP default file that starts the installation.

• Following lines need to added to the default file:

label Linux
menu label ^Install RHEL
menu default
kernel vmlinuz
append initrd=initrd.img
ks=http://server1.example.com/anaconda-ks.cfg