

Aim :- Installation of Redhat Linux Installation.

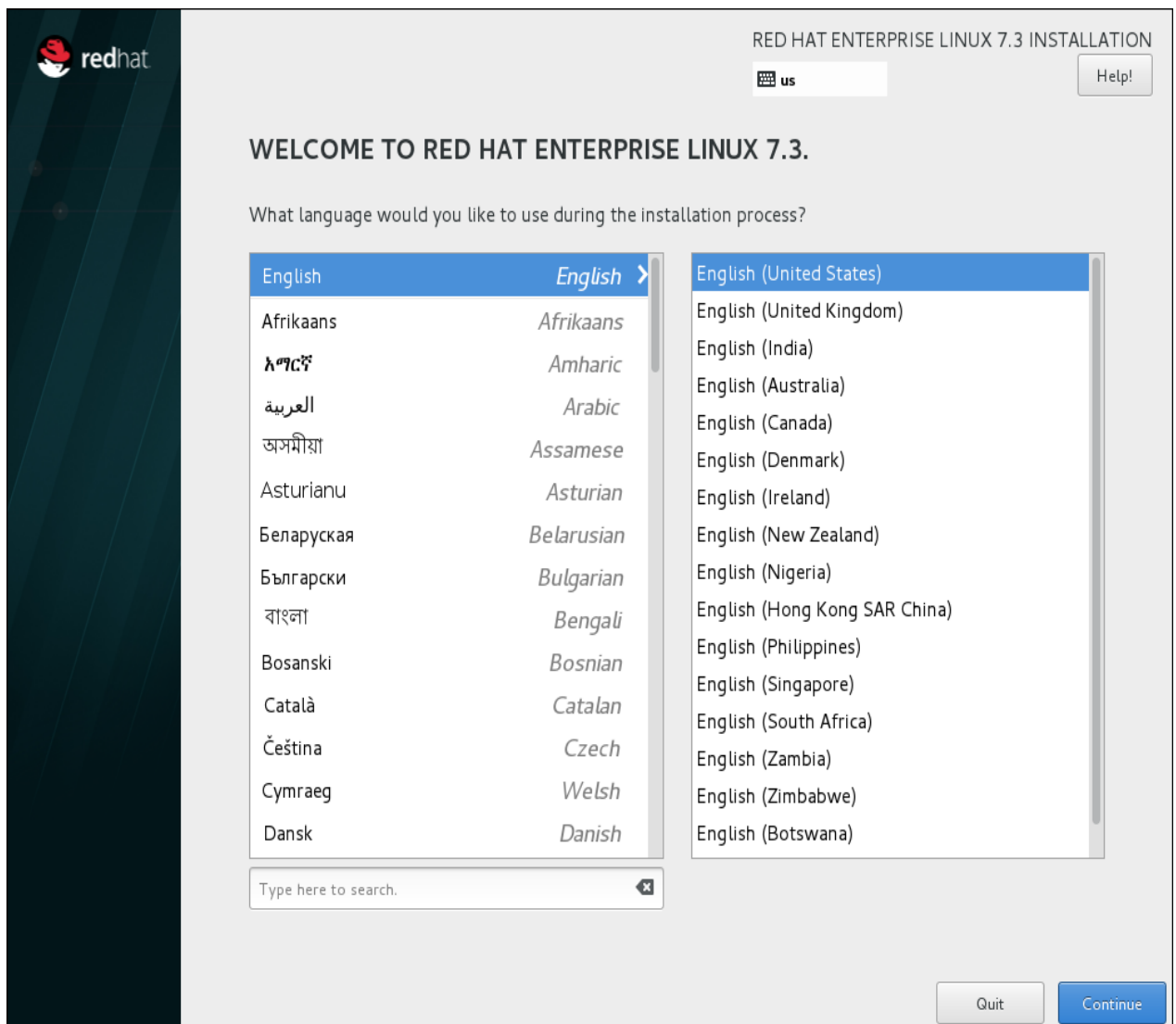
Solution :-

After booting the installation USB drive:

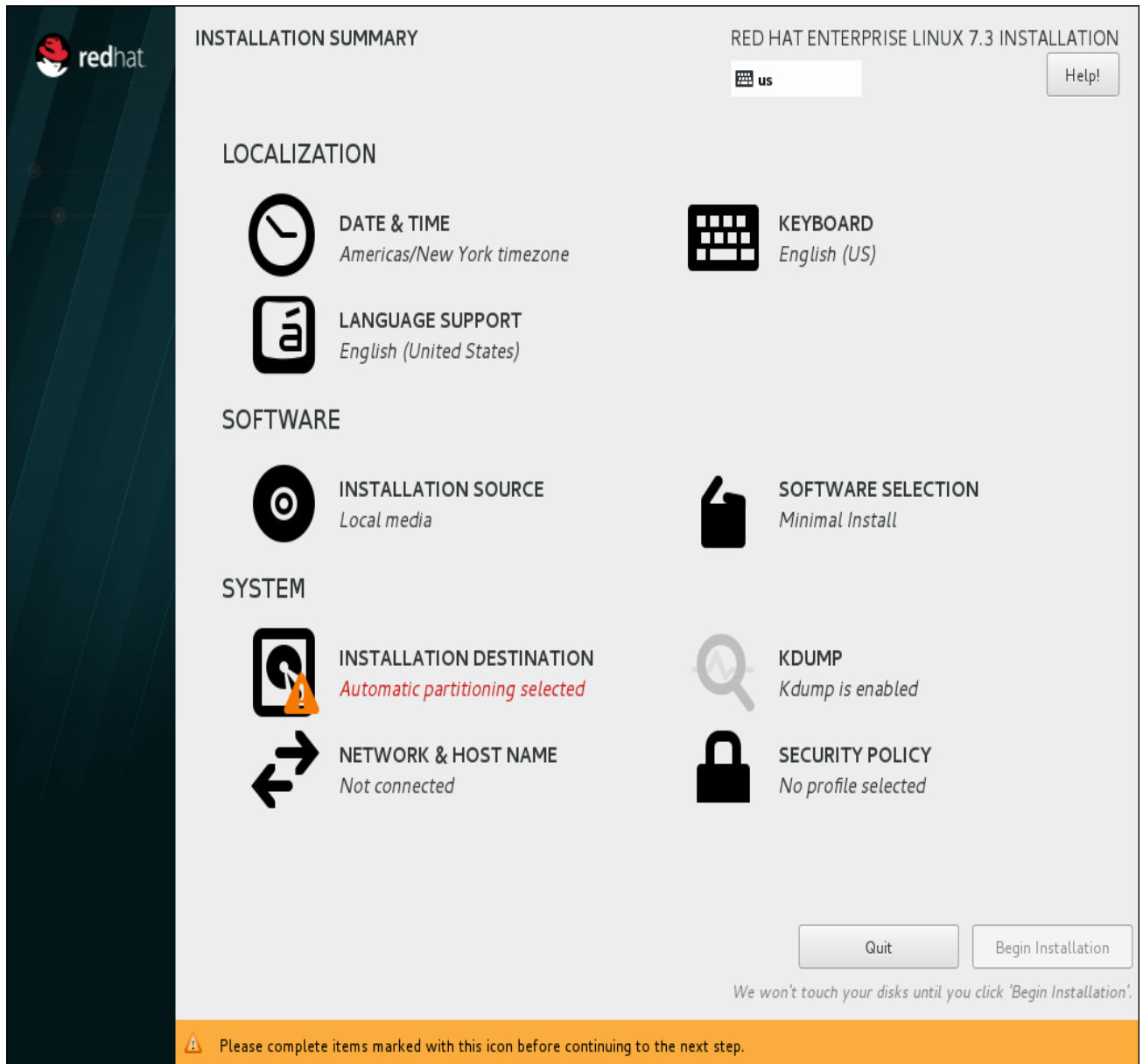
1. Select **Install Red Hat Enterprise Linux** in the boot menu and press **Enter**.



2. After **Anaconda**, the Red Hat Enterprise Linux installer, started, select your language and region, and click **Continue**.



3. The **Installation Summary** is the central screen to set configuration options:



You can display and modify the individual options in any order. If a configuration option was automatically configured correctly, no further action is required. However, if items are marked with an exclamation point icon, you must complete the configuration for these items before you can begin the installation.

#### Note

Nothing will be written to the disk until you click the **Begin Installation** button.

4. Select **Date & Time**:
- Set your region and the nearest city in your time zone.
  - Click **Done** to return to the **Installation Summary**.

DATE & TIME

Done

RED HAT ENTERPRISE LINUX 7.3 INSTALLATION

us

Help!

Region: Americas

City: New York

Network Time

OFF

^

^

^

04:37 PM

24-hour

AM/PM

v

v

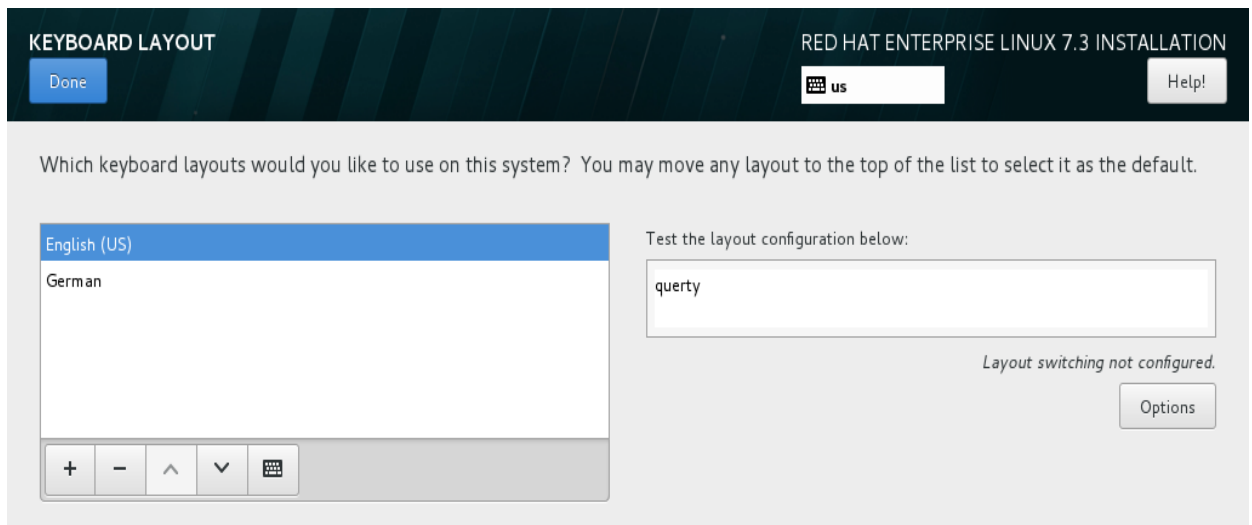
v

04

26

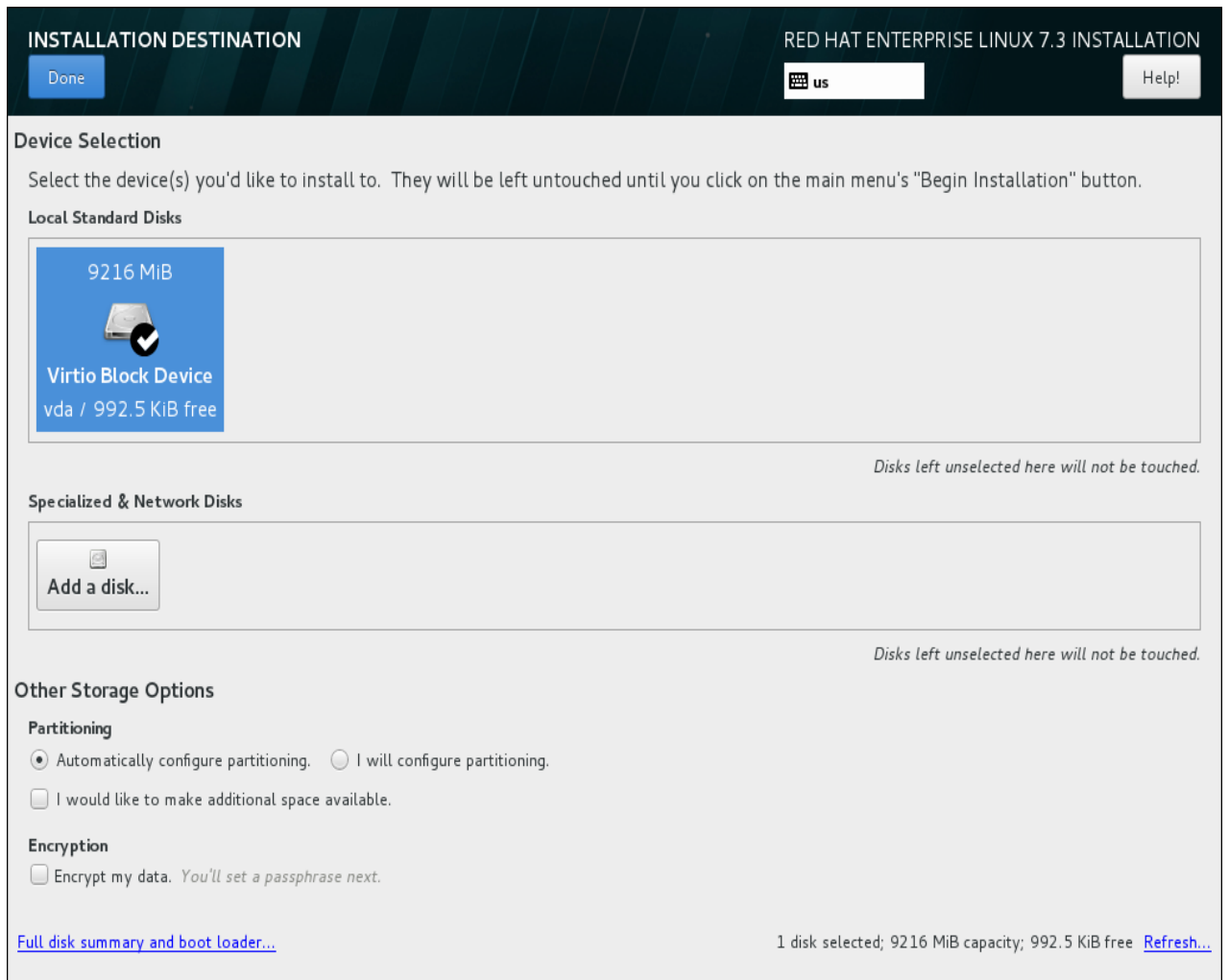
2017

5. Select **Keyboard Layout**:
  - a. Use the + and - buttons to add and remove keyboard layouts.
  - b. If you enable multiple keyboard layouts, move your preferred layout to the top of the list using the ↑ button to set it as default.
  - c. Click **Done** to return to the **Installation Summary**.



6. Select **Installation Destination**:
  - a. Select the target disk. A check mark is displayed next to the selected target.

The selected disk is partitioned automatically.
  - b. Click **Done** to return to the **Installation Summary**.



7. Select **Network and Hostname**:

- a. Click the **Ethernet** sliding switch in the top right corner to enable the network configuration.
- b. Optional, select the device and click **Configure** to update the network interface configuration.
- c. Click **Done** to return to the **Installation Summary**.

NETWORK & HOST NAME

RED HAT ENTERPRISE LINUX 7.3 INSTALLATION

Done

us

Help!

Ethernet (eth0)  
Red Hat, Inc. Virtio network device

+ -

Ethernet (eth0)  
Connected

Hardware Address 52:54:00:27:40:C7

Speed

IP Address 192.168.122.185

Subnet Mask 255.255.255.0

Default Route 192.168.122.1

DNS 192.168.122.1

Configure...

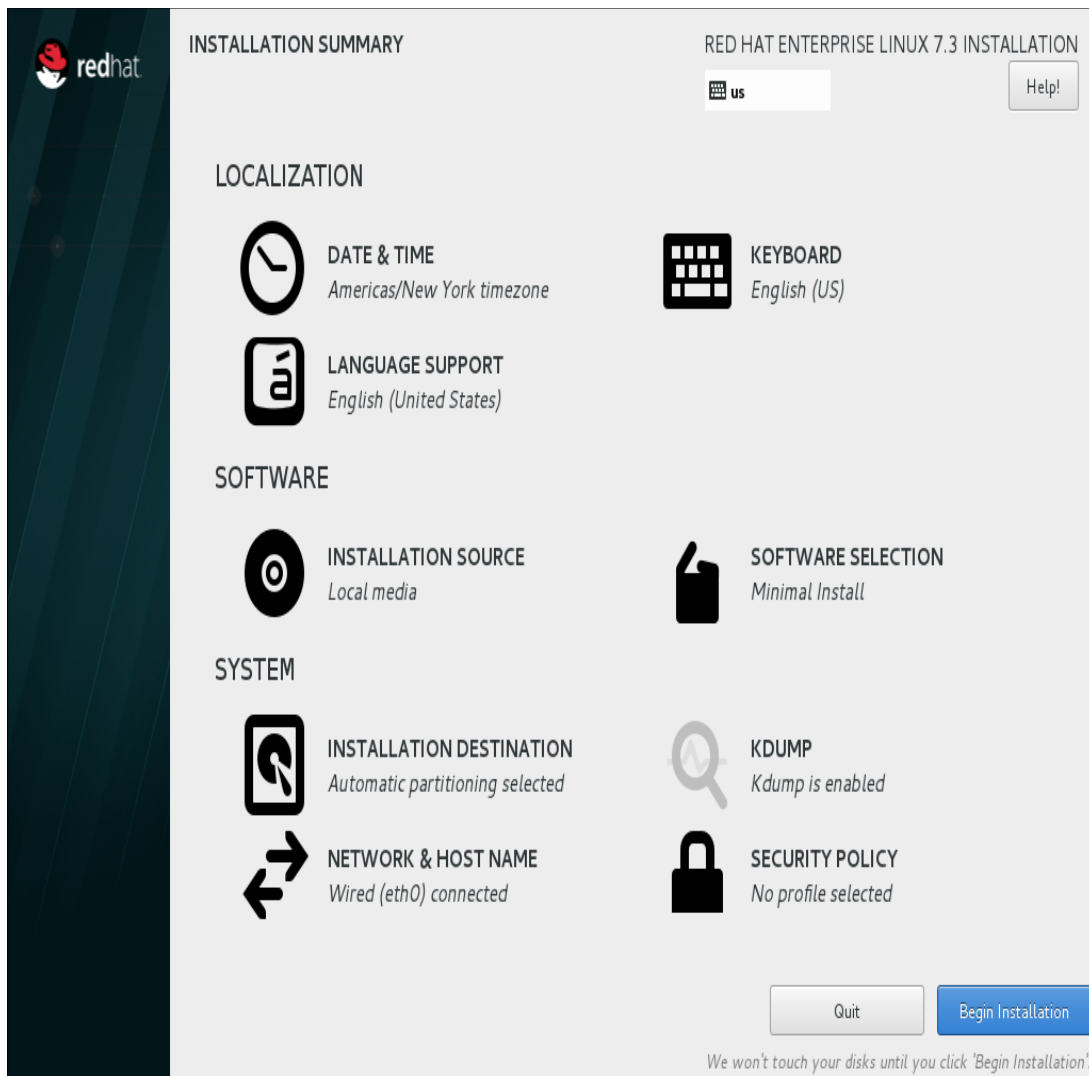
Host name:

Current host name: server.example.com

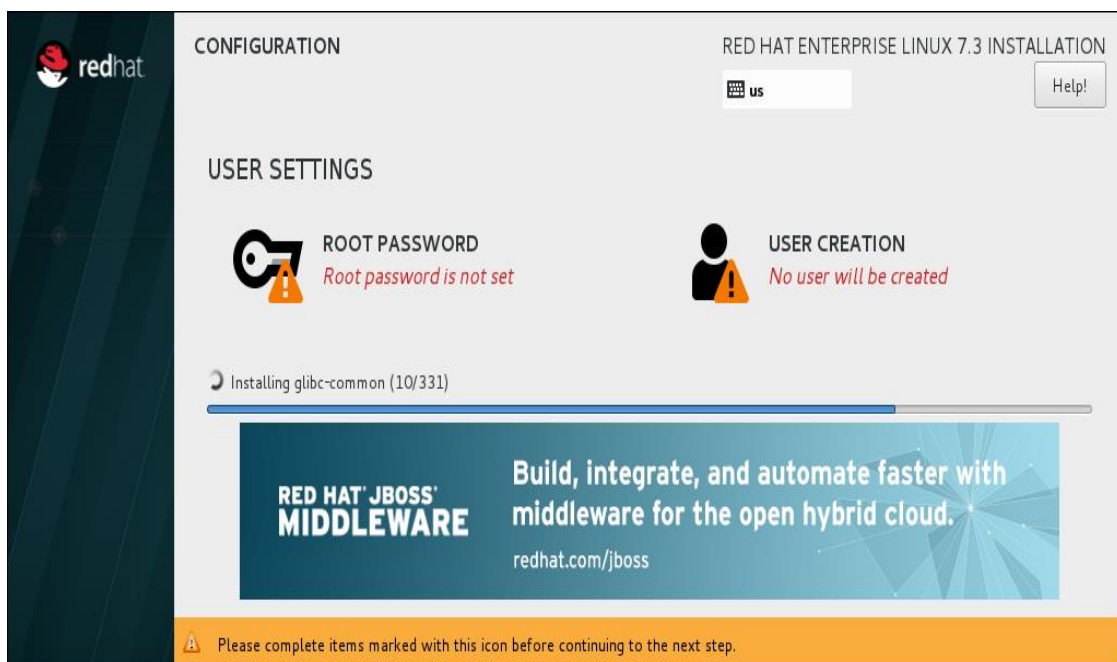
**Note**

**Anaconda** applies the network settings immediately. They are used during the setup and after the installation.

8. On the **Installation Summary** screen, click **Begin Installation**.

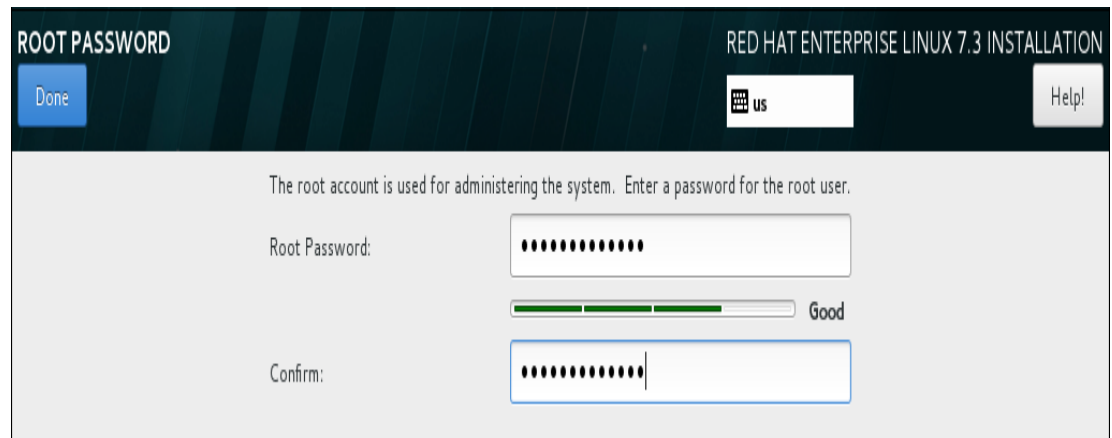


9. The installation starts and the **Configuration** screen is displayed:

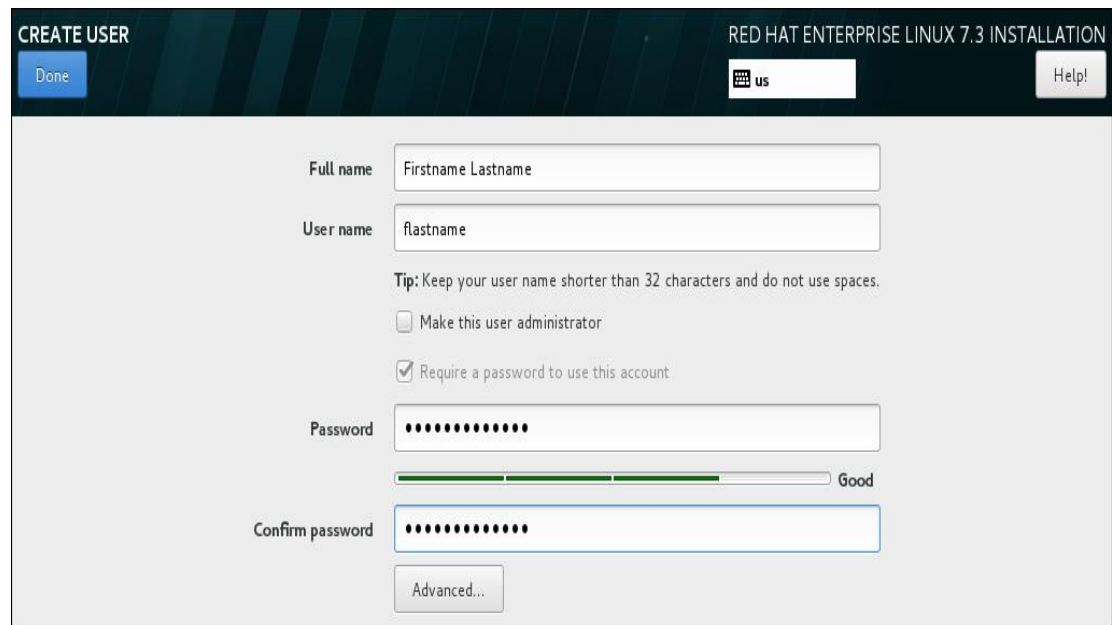


During the installation:

- a. Select **Root Password**:
  - i. Enter the password for the `root` user and confirm it.
  - ii. Click **Done** to return to the **Configuration** screen.



- b. Select **User Creation**:
  - i. Enter the user's full name.
  - ii. Optionally, update the automatically generated user name.
  - iii. Set the password and confirm it.
  - iv. Optionally, check the **Make this user administrator** check box. This will add the user to the `wheel` group and allow this account to use `sudo` without any further configuration.
  - v. Click **Done** to return to the **Configuration** screen.



- c. Wait until the installation completes and click **Reboot**.
10. After the installed system has been started:
    - If you installed the server using the **Server with GUI** base environment, the **Initial Setup** application is started automatically:
      - i. Accept the license agreement.
      - ii. Register the system.

For details, see [Chapter 30, Initial Setup](#).

- If you selected any other base environment during the installation:

- i. Log into the system as the `root` user.

- ii. Register the system and automatically attach a subscription:

- iii. `# subscription-manager register --auto-attach \`

- `--username=user_name --password=password`



Aim :- Working with Grub and init file

Solution :-

Red Hat Enterprise Linux 7 is distributed with version 2 of the GNU GRand Unified Bootloader (GRUB 2), which allows the user to select an operating system or kernel to be loaded at system boot time. GRUB 2 also allows the user to pass arguments to the kernel.

## Introduction to GRUB 2

GRUB 2 reads its configuration from the `/boot/grub2/grub.cfg` file on traditional BIOS-based machines and from the `/boot/efi/EFI/redhat/grub.cfg` file on UEFI machines. This file contains menu information.

The GRUB 2 configuration file, `grub.cfg`, is generated during installation, or by invoking the `/usr/sbin/grub2-mkconfig` utility, and is automatically updated by `grubby` each time a new kernel is installed. When regenerated manually using **grub2-mkconfig**, the file is generated according to the template files located in `/etc/grub.d/`, and custom settings in the `/etc/default/grub` file. Edits of `grub.cfg` will be lost any time **grub2-mkconfig** is used to regenerate the file, so care must be taken to reflect any manual changes in `/etc/default/grub` as well.

Normal operations on `grub.cfg`, such as the removal and addition of new kernels, should be done using the `grubby` tool and, for scripts, using `new-kernel-pkg` tool. If you use `grubby` to modify the default kernel the changes will be inherited when new kernels are installed. For more information on `grubby`, see [Section 26.4, “Making Persistent Changes to a GRUB 2 Menu Using the grubby Tool”](#).

The `/etc/default/grub` file is used by the `grub2-mkconfig` tool, which is used by `anaconda` when creating `grub.cfg` during the installation process, and can be used in the event of a system failure, for example if the boot loader configurations need to be recreated. In general, it is not recommended to replace the `grub.cfg` file by manually running `grub2-mkconfig` except as a last resort. Note that any manual changes to `/etc/default/grub` require rebuilding the `grub.cfg` file.

### *Menu Entries in grub.cfg*

Among various code snippets and directives, the `grub.cfg` configuration file contains one or more `menuentry` blocks, each representing a single GRUB 2 boot menu entry. These blocks always start with the `menuentry` keyword followed by a title, list of options, and an opening curly bracket, and end with a closing curly bracket. Anything between the opening and closing bracket should

be indented. For example, the following is a sample `menuentry` block for Red Hat Enterprise Linux 7 with Linux kernel 3.8.0-0.40.el7.x86\_64:

```
menuentry 'Red Hat Enterprise Linux Server' --class red --class gnu-linux --class gnu --class os
$menuentry_id_option 'gnulinux-simple-c60731dc-9046-4000-9182-64bdcce08616' {

    load_video

    set gfxpayload=keep

    insmod gzio

    insmod part_msdos

    insmod xfs

    set root='hd0,msdos1'

    if [ x$feature_platform_search_hint = xy ]; then

        search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-efi=hd0,msdos1 --hint-
baremetal=ahci0,msdos1 --hint='hd0,msdos1' 19d9e294-65f8-4e37-8e73-d41d6daa6e58

    else

        search --no-floppy --fs-uuid --set=root 19d9e294-65f8-4e37-8e73-d41d6daa6e58

    fi

    echo 'Loading Linux 3.8.0-0.40.el7.x86_64 ...'

    linux16 /vmlinuz-3.8.0-0.40.el7.x86_64 root=/dev/mapper/rhel-root ro rd.md=0 rd.dm=0 rd.lvm.lv=rhel/swap
crashkernel=auto rd.luks=0 vconsole.keymap=us rd.lvm.lv=rhel/root rhgb quiet

    echo 'Loading initial ramdisk ...'

    initrd /initramfs-3.8.0-0.40.el7.x86_64.img

}
```

Each `menuentry` block that represents an installed Linux kernel contains `linux` on 64-bit IBM POWER Series, `linux16` on x86\_64 BIOS-based systems, and `linuxefi` on UEFI-based systems. Then

the `initrd` directives followed by the path to the kernel and the `initramfs` image respectively. If a separate `/boot` partition was created, the paths to the kernel and the `initramfs` image are relative to `/boot`. In the example above, the `initrd /initramfs-3.8.0-0.40.el7.x86_64.img` line means that the `initramfs` image is actually located at `/boot/initramfs-3.8.0-0.40.el7.x86_64.img` when the `root` file system is mounted, and likewise for the kernel path.

The kernel version number as given on the `linux16 /vmlinuz-kernel_version` line must match the version number of the `initramfs` image given on the `initrd /initramfs-kernel_version.img` line of each `menuentry` block. For more information on how to verify the initial RAM disk image, see [Red Hat Enterprise 7 Kernel Administration Guide](#).

#### NOTE

In `menuentry` blocks, the `initrd` directive must point to the location (relative to the `/boot/` directory if it is on a separate partition) of the `initramfs` file corresponding to the same kernel version. This directive is called `initrd` because the previous tool which created initial RAM disk images, `mkinitrd`, created what were known as `initrd` files. The `grub.cfg` directive remains `initrd` to maintain compatibility with other tools. The file-naming convention of systems using the `dracut` utility to create the initial RAM disk image is `initramfs-kernel_version.img`.

For information on using **Dracut**, see [Red Hat Enterprise 7 Kernel Administration Guide](#).

Practical No : 03

Aim :- Basic Linux Commands

Solution :-

## Unix / Linux - Useful Commands

### Files and Directories

These commands allow you to create directories and handle files.

Given below is the list of commands in Files and Directories.

### Manipulating data

The contents of files can be compared and altered with the following commands.

Given below is the list of commands in Manipulating data.

### Compressed Files

Files may be compressed to save space. Compressed files can be created and examined.

Sr.No.	Command & Description
1	<b>compress</b> Compresses files
2	<b>gunzip</b> Helps uncompress gzipped files
3	<b>gzip</b> GNU alternative compression method
4	<b>uncompress</b> Helps uncompress files

### Getting Information

Various Unix manuals and documentation are available on-line. The following Shell commands give information –

Sr.No.	Command & Description
--------	-----------------------

1	<b>info</b> Displays command information pages online
2	<b>man</b> Displays manual pages online

## Network Communication

These following commands are used to send and receive files from a local Unix hosts to the remote host around the world.

Sr.No.	Command & Description
1	<b>ftp</b> File transfer program
2	<b>rnp</b> Remote file copy
3	<b>rlogin</b> Remote login to a Unix host
4	<b>telnet</b> Makes terminal connection to another host
5	<b>ssh</b> Secures shell terminal or command connection

Some of these commands may be restricted at your computer for security reasons.

## Messages between Users

The Unix systems support on-screen messages to other users and world-wide electronic mail –

Sr.No.	Command & Description
1	<b>mail</b> Simple send or read mail program
2	<b>talk</b> Talks to another user
3	<b>write</b> Writes message to another user

Practical No : 04

Aim :- Advanced Linux Commands.

Solution :-

## CAT

Cat(concatenate) command is very frequently used in Linux. It reads data from the file and gives their content as output. It helps us to create, view, concatenate files. So let us see some frequently used cat commands.

### 1) To view a single file

**Command:**

```
$cat filename
```

Output

It will show content of given filename

### 2) To view multiple files

**Command:**

```
$cat file1 file2
```

## Grep, Egrep, Fgrep, Rgrep Commands

These filters output lines matching a given pattern. They read lines from a file or standard input, and print all matching lines by default to standard output.

**Note:** The main program is grep, the variations are simply the same as using specific grep options as below (and they are still being used for backward compatibility):

```
$ egrep = grep -E
```

```
$ fgrep = grep -F
```

```
$ rgrep = grep -r
```

Below are some basic grep commands:

```
tecmint@TecMint ~ $ grep "aaronkilik" /etc/passwd
aaronkilik:x:1001:1001::/home/aaronkilik:
```

```
tecmint@TecMint ~ $ cat /etc/passwd | grep "aronkilik"
aaronkilik:x:1001:1001::/home/aaronkilik:
```

## sort Command

**sort** is used to sort lines of a text file or from standard input.

Below is the content of a file named **domains.list**:

```
tecmint@TecMint ~ $ cat domains.list
tecmint.com
tecmint.com
news.tecmint.com
news.tecmint.com
linuxsay.com
linuxsay.com
windowsmint.com
windowsmint.com
```

You can run a simple [sort command](#) to sort the file content like so:

```
tecmint@TecMint ~ $ sort domains.list
linuxsay.com
linuxsay.com
news.tecmint.com
news.tecmint.com
tecmint.com
tecmint.com
windowsmint.com
windowsmint.com
```

Aim :- Working with Vi Editor.

There are many ways to edit files in Unix. Editing files using the screen-oriented text editor **vi** is one of the best ways. This editor enables you to edit lines in context with other lines in the file.

vi is generally considered the de facto standard in Unix editors because –

- You can use the **vi** editor to edit an existing file or to create a new file from scratch. You can also use this editor to just read a text file.

The following table lists out the basic commands to use the vi editor –

Following is an example to create a new file **testfile** if it already does not exist in the current working directory –

The above command will generate the following output –

You will notice a **tilde** (~) on each line following the cursor. A tilde represents an unused line. If a line does not begin with a tilde and appears to be blank, there is a space, tab, newline, or some other non-viewable character present.

You now have one open file to start working on. Before proceeding further, let us understand a few important concepts.



## Operation Modes

While working with the vi editor, we usually come across the following two modes –

- **Command mode** – This mode enables you to perform administrative tasks such as saving the files, executing the commands, moving the cursor, cutting (yanking) and pasting the lines or words, as well as finding and replacing. In this mode, whatever you type is interpreted as a command.
- **Insert mode** – This mode enables you to insert text into the file. Everything that's typed in this mode is interpreted as input and placed in the file.

vi always starts in the **command mode**. To enter text, you must be in the insert mode for which simply type **i**. To come out of the insert mode, press the **Esc** key, which will take you back to the command mode.

**Hint** – If you are not sure which mode you are in, press the Esc key twice; this will take you to the command mode. You open a file using the vi editor. Start by typing some characters and then come to the command mode to understand the difference.

## Getting Out of vi

The command to quit out of vi is **:q**. Once in the command mode, type colon, and 'q', followed by return. If your file has been modified in any way, the editor will warn you of this, and not let you quit. To ignore this message, the command to quit out of vi without saving is **:q!**. This lets you exit vi without saving any of the changes.

The command to save the contents of the editor is **:w**. You can combine the above command with the quit command, or use **:wq** and return.

The easiest way to **save your changes and exit vi** is with the ZZ command. When you are in the command mode, type **ZZ**. The **ZZ** command works the same way as the **:wq** command.

If you want to specify/state any particular name for the file, you can do so by specifying it after the **:w**. For example, if you wanted to save the file you were working on as another filename called **filename2**, you would type **:w filename2** and return.

## Moving within a File

To move around within a file without affecting your text, you must be in the command mode (press Esc twice). The following table lists out a few commands you can use to move around one character at a time –

Sr.No.	Command & Description
1	<b>k</b> Moves the cursor up one line
2	<b>j</b> Moves the cursor down one line
3	<b>h</b> Moves the cursor to the left one character position
4	<b>l</b> Moves the cursor to the right one character position

The following points need to be considered to move within a file –

- vi is case-sensitive. You need to pay attention to capitalization when using the commands.
- Most commands in vi can be prefaced by the number of times you want the action to occur. For example, **2j** moves the cursor two lines down the cursor location.

There are many other ways to move within a file in vi. Remember that you must be in the command mode (**press Esc twice**). The following table lists out a few commands to move around the file –

Given below is the list of commands to move around the file.

## Control Commands

The following commands can be used with the Control Key to performs functions as given in the table below –

Given below is the list of control commands.

## Editing Files

To edit the file, you need to be in the insert mode. There are many ways to enter the insert mode from the command mode –

Sr.No.	Command & Description
1	<b>i</b> Inserts text before the current cursor location
2	<b>I</b> Inserts text at the beginning of the current line
3	<b>a</b> Inserts text after the current cursor location
4	<b>A</b> Inserts text at the end of the current line
5	<b>o</b> Creates a new line for text entry below the cursor location
6	<b>O</b> Creates a new line for text entry above the cursor location

## Deleting Characters

Here is a list of important commands, which can be used to delete characters and lines in an open file –

Sr.No.	Command & Description
1	<b>x</b> Deletes the character under the cursor location
2	<b>X</b> Deletes the character before the cursor location
3	<b>dw</b> Deletes from the current cursor location to the next word
4	<b>d^</b>

	Deletes from the current cursor position to the beginning of the line
5	<b>d\$</b> Deletes from the current cursor position to the end of the line
6	<b>D</b> Deletes from the cursor position to the end of the current line
7	<b>dd</b> Deletes the line the cursor is on

As mentioned above, most commands in vi can be prefaced by the number of times you want the action to occur. For example, **2x** deletes two characters under the cursor location and **2dd** deletes two lines the cursor is on.

It is recommended that the commands are practiced before we proceed further.

## Change Commands

You also have the capability to change characters, words, or lines in vi without deleting them. Here are the relevant commands –

Sr.No.	Command & Description
1	<b>cc</b> Removes the contents of the line, leaving you in insert mode.
2	<b>cw</b> Changes the word the cursor is on from the cursor to the lowercase <b>w</b> end of the word.
3	<b>r</b> Replaces the character under the cursor. vi returns to the command mode after the replacement is entered.
4	<b>R</b> Overwrites multiple characters beginning with the character currently under the cursor. You must use <b>Esc</b> to stop the overwriting.
5	<b>s</b> Replaces the current character with the character you type. Afterward, you are left in the insert mode.
6	<b>S</b> Deletes the line the cursor is on and replaces it with the new text. After the new text is entered, vi remains in the insert mode.

## Copy and Paste Commands

You can copy lines or words from one place and then you can paste them at another place using the following commands –

Sr.No.	Command & Description
1	<b>yy</b> Copies the current line.
2	<b>yw</b> Copies the current word from the character the lowercase w cursor is on, until the end of the word.
3	<b>p</b> Puts the copied text after the cursor.
4	<b>P</b> Puts the yanked text before the cursor.

## Advanced Commands

There are some advanced commands that simplify day-to-day editing and allow for more efficient use of vi –

Given below is the list advanced commands.

### Word and Character Searching

The vi editor has two kinds of searches: **string** and **character**. For a string search, the / and ? commands are used. When you start these commands, the command just typed will be shown on the last line of the screen, where you type the particular string to look for.

These two commands differ only in the direction where the search takes place –

- The / command searches forwards (downwards) in the file.
- The ? command searches backwards (upwards) in the file.

The **n** and **N** commands repeat the previous search command in the same or the opposite direction, respectively. Some characters have special meanings. These characters must be preceded by a backslash (\) to be included as part of the search expression.

Sr.No.	Character &Description
1	<b>^</b> Searches at the beginning of the line (Use at the beginning of a search expression).
2	<b>.</b> Matches a single character.
3	<b>*</b> Matches zero or more of the previous character.
4	<b>\$</b> End of the line (Use at the end of the search expression).
5	<b>[</b>

	Starts a set of matching or non-matching expressions.
6	<p>&lt;</p> <p>This is put in an expression escaped with the backslash to find the ending or the beginning of a word.</p>
7	<p>&gt;</p> <p>This helps see the '&lt;' character description above.</p>

The character search searches within one line to find a character entered after the command. The **f** and **F** commands search for a character on the current line only. **f** searches forwards and **F** searches backwards and the cursor moves to the position of the found character.

The **t** and **T** commands search for a character on the current line only, but for **t**, the cursor moves to the position before the character, and **T** searches the line backwards to the position after the character.

## Set Commands

You can change the look and feel of your vi screen using the following **:set** commands. Once you are in the command mode, type **:set** followed by any of the following commands.

Sr.No.	Command & Description
1	<p><b>:set ic</b></p> <p>Ignores the case when searching</p>
2	<p><b>:set ai</b></p> <p>Sets autoindent</p>
3	<p><b>:set noai</b></p> <p>Unsets autoindent</p>
4	<p><b>:set nu</b></p> <p>Displays lines with line numbers on the left side</p>
5	<p><b>:set sw</b></p> <p>Sets the width of a software tabstop. For example, you would set a shift width of 4 with this command — <b>:set sw = 4</b></p>
6	<p><b>:set ws</b></p> <p>If <i>wraps</i> is set, and the word is not found at the bottom of the file, it will try searching for it at the beginning</p>
7	<p><b>:set wm</b></p> <p>If this option has a value greater than zero, the editor will automatically "word wrap". For example, to set the wrap margin to two characters, you would type this: <b>:set wm = 2</b></p>
8	<p><b>:set ro</b></p> <p>Changes file type to "read only"</p>

9	<b>:set term</b> Prints terminal type
10	<b>:set bf</b> Discards control characters from input

## Running Commands

The vi has the capability to run commands from within the editor. To run a command, you only need to go to the command mode and type **:!** command.

For example, if you want to check whether a file exists before you try to save your file with that filename, you can type **:! ls** and you will see the output of **ls** on the screen.

You can press any key (or the command's escape sequence) to return to your vi session.

## Replacing Text

The substitution command (**:s/**) enables you to quickly replace words or groups of words within your files. Following is the syntax to replace text –

**:s/search/replace/g**

The **g** stands for globally. The result of this command is that all occurrences on the cursor's line are changed.

Aim :- Working with Users, Groups, and Permissions.

Solution :-

## Linux Users

Linux is designed for multiple users to have access to the same system at the same time. There are three types of users in Linux:

### System users

System users run non-interactive or background processes. These users don't have their own login or home directory.

### Regular users

These users run interactive processes and have their own home directory and login.

### Super user

There is only one root or superuser. This user is the ultimate authority on file ownership and permissions. You create new users as the root user.

## Linux - Viewing all users

You can view a list of all users on the system via:

```
$ cat /etc/passwd
```

This shows every user on the system. Each line represents a user with basic information such as username, user ID, group ID, and the home directory for that user.

## Linux - Switching to the root user

How you switch to the root user may vary depending on your Linux distro. For Ubuntu, you can run:

```
$ sudo su
```

With other distros, you may be able to switch via just `su`. Ubuntu doesn't set a password for the root user by default so `sudo` is required to bypass the password prompt. If you are using Ubuntu and want to set a password for root, you can run `sudo passwd root` (however this is not recommended).

## Linux - Adding a user

As the root user, you can add additional users to the system via:

```
# useradd <username>
```

This will add a new user with the specified name to the system. To add a password login for the new user, run:

```
# passwd <username>
```

This will prompt you to set a password for the new user.

## Linux - Logging in as a new user

Once you've created a password for the new user, you can login via:

```
# su <username>
```

It should prompt you for the password. Once entered correctly, you will be logged in as the new user.

## Linux - Removing a user

As root user, you can remove a user via:

```
# userdel <username>
```

## Linux - Difference between su and sudo

The `su` command is used to switch users. When no username is provided, `su` switches to the root user by default. The `sudo` command allows users and groups to access commands they normally wouldn't have access to. By using `sudo`, you avoid having to switch to the root user to run restricted commands.

In order for a user to run `sudo` commands, the user must be added to the `sudoers` file in `/etc/` directory. Editing the `sudoers` file directly is not recommended. Alternatively, the `visudo` command provides a safe and secure way to edit the `sudoers` file.

## Linux Permissions

Permissions exist to protect a user's files from other users. Run `ls -l` in any directory to list the files and permissions. After running the command, you should see something like:



```
-rwxrwxrwx 1 svc-hadoop-platform-dev cperry    33 May 25 15:39 sample.txt
drw-r--r-- 1 svc-hadoop-platform-dev mjones   461 Aug  4 13:06 sample_files
-rw-rw-r-- 1 svc-hadoop-platform-dev mjones    55 Aug  4 13:06 subs.txt
-rwxrwx--- 1 svc-hadoop-platform-dev terryb   56 Aug 16 17:00 otherfile.txt
```

The first ten characters (-rwxrwxrwx) indicate the permissions for the listed resource. This may seem confusing, but when you break it down it makes more sense.

The first character (d, -) indicates the file type. Normal files are represented by (-) and directories (d). You'll notice the sample\_files directory is the only item listed with a (d) as the first character because it is a directory and not a file like the others.

The next three characters are the permissions for the file's owner. Permissions are represented by three characters `rwX`.

- r - the user has read access
- w - the user has write access
- x - the user has execute permissions

Remember that a hyphen represents restriction of access. Additionally, permissions will always be listed in the same order `rwX`.

The next three characters are permissions for the file's group.

The last three characters are the permissions for all other users.

Taking the subs.txt line sample output (-rw-rw-r--), we infer that:

- (-) this is a regular file
- (rw-) the file owner (mjones) has read/write access to the file
- (rw-) the file's group has read/write access to the file
- (r--) all other users only have read only access to the file

With most implementations, the file owner has the most permissions while the group and other users only have a subset of the owner's permissions.

## Linux Groups

Groups are a way of managing a collection of users. You can see a full list of groups and their members at `/etc/group`.

Every user is automatically associated with a group. This means every file you create as a user will be associated with that user's primary group. To run programs or create files with different groups, you can either:

## Add a new group

```
$ newgrp <marketing>
```

## Change groups

```
chgrp <newgroup>
```

# Linux changing permissions with chmod

As the root user or sudoer, you can change file permissions with `chmod`. For example:

```
chmod +rwx sample.txt
```

This will grant read, write, execute permissions to the file owner for sample.txt. The following will remove the same access.

```
chmod -rwx sample.txt
```

You can also grant specific class permissions with `u`, `g`, and `o`. For example:

```
chmod ug+rw sample.txt //gives read and write access to file owner and group
chmod go-rw sample.txt //denies read and write access to group and outsiders
chmod ugo+rwx sample.txt //gives read, write, execute access to user, group, and outsiders
```

You can also use the octect method, where the numbers 0-7 represent different permissions for each permission class. For example:

```
chmod 444 sample.txt
//results in dr--r--r--
```

This will give read only permissions (4) to the user, the group, and outsiders.

# Linux changing ownership with chown

As the root user or sudoer, you can change file ownership with `chown`:

```
chown mjones sample.txt
```

If you want to change just the group, you can run:

```
chown :newgroup sample.txt
```

Aim :- Setting NFS File Server.

Solution :-

## Setting Up an NFS Server with an NFS Share

Let's see how to set up an NFS server and create an NFS file share, which client machines can mount and access.

### Installing NFS Server

Here is how to install the NFS Kernel—this is the server component that enables a machine to expose directories as NFS shares.

On Ubuntu and Debian:

```
sudo apt-get update
```

```
sudo apt install nfs-kernel-server
```

On CentOS and Fedora:

```
yum -y install nfs-utils
```

```
apt-get install nfs-kernel-server
```

### Create Root NFS Directory

We'll now create the root directory of the NFS shares, this is also known as an export folder.

```
sudo mkdir /mnt/mysharedir
```

Set permissions so that any user on the client machine can access the folder (in the real world you need to consider if the folder needs more restrictive settings).

```
sudo chown nobody:nogroup /mnt/mysharedir #no-one is owner
```

```
sudo chmod 777 /mnt/mysharedir #everyone can modify files
```

### Define Access for NFS Clients in Export File

To grant access to NFS clients, we'll need to define an export file. The file is typically located at `/etc/exports`

Edit the `/etc/exports` file in a text editor, and add one of the following three directives.

All the directives below use the options `rw`, which enables both read and write, `sync`, which writes changes to disk before allowing users to access the modified file, and `no_subtree_check`, which means NFS doesn't check if each subdirectory is accessible to the user.

<b>To enable access to a single client</b>	<code>/mnt/mysharedir</code> <code>{clientIP} (rw,sync,no_subtree_check)</code>
<b>To enable access to several clients</b>	<code>/mnt/mysharedir {clientIP-</code> <code>1} (rw,sync,no_subtree_check)</code> <code>{clientIP-2} (...)</code>

	<code>{clientIP-3}(...)</code>
<b>To enable access to an entire subnet</b>	<code>/mnt/mysharedir</code> <code>{subnetIP}/{subnetMask}(rw, sync, no_subtree_check)</code>

## Make the NFS Share Available to Clients

You can now make the shared directory available to clients using the `exportfs` command. After running this command, the NFS Kernel should be restarted.

```
sudo exportfs -a #making the file share available
```

```
sudo systemctl restart nfs-kernel-server #restarting the NFS kernel
```

If you have a firewall enabled, you'll also need to open up firewall access using the `sudo ufw allow` command.

Aim :- Setting up DNS server.

Solution :-

## Windows

DNS settings are specified in the **TCP/IP Properties** window for the selected network connection.

### Example: Changing DNS server settings on Windows 10

1. Go to the **Control Panel**.
2. Click **Network and Internet > Network and Sharing Center > Change adapter settings**.
3. Select the connection for which you want to configure Google Public DNS. For example:
  - To change the settings for an Ethernet connection, right-click the Ethernet interface and select **Properties**.
  - To change the settings for a wireless connection, right-click the Wi-Fi interface and select **Properties**.
4. If you are prompted for an administrator password or confirmation, type the password or provide confirmation. Select the **Networking** tab. Under **This connection uses the following items**, select **Internet Protocol Version 4 (TCP/IPv4)** or **Internet Protocol Version 6 (TCP/IPv6)** and then click **Properties**.
5. Click **Advanced** and select the **DNS** tab. If there are any DNS server IP addresses listed there, write them down for future reference, and remove them from this window.
6. Click **OK**.
7. Select **Use the following DNS server addresses**. If there are any IP addresses listed in the **Preferred DNS server** or **Alternate DNS server**, write them down for future reference.
8. Replace those addresses with the IP addresses of the Google DNS servers:
  - For IPv4: 8.8.8.8 and/or 8.8.4.4.
  - For IPv6: 2001:4860:4860::8888 and/or 2001:4860:4860::8844.
  - For IPv6-only: you can use Google Public DNS64 *instead of* the IPv6 addresses in the previous point.
9. Test that your setup is working correctly; see Test your new settings.
10. Repeat the procedure for additional network connections you want to change.

## Linux

In most modern Linux distributions, DNS settings are configured through Network Manager.

### Example: Changing DNS server settings on Ubuntu

1. Click **System > Preferences > Network Connections**.
2. Select the connection for which you want to configure Google Public DNS. For example:
  - To change the settings for an Ethernet connection, select the **Wired** tab, then select your network interface in the list. It is usually called `eth0`.
  - To change the settings for a wireless connection, select the **Wireless** tab, then select the appropriate wireless network.
3. Click **Edit**, and in the window that appears, select the **IPv4 Settings** or **IPv6 Settings** tab.
4. If the selected method is **Automatic (DHCP)**, open the dropdown and select **Automatic (DHCP) addresses only** instead. If the method is set to something else, do not change it.
5. In the **DNS servers** field, enter the Google Public DNS IP addresses, separated by a comma:
  - For IPv4: 8.8.8.8 and/or 8.8.4.4.
  - For IPv6: 2001:4860:4860::8888 and/or 2001:4860:4860::8844.
  - For IPv6-only: you can use Google Public DNS64 *instead of* the IPv6 addresses in the previous point.

6. Click **Apply** to save the change. If you are prompted for a password or confirmation, type the password or provide confirmation.
7. Test that your setup is working correctly; see Test your new settings.
8. Repeat the procedure for additional network connections you want to change.

If your distribution doesn't use Network Manager, your DNS settings are specified in `/etc/resolv.conf`.

#### Example: Changing DNS server settings on a Debian server

1. Edit `/etc/resolv.conf`:  

```
sudo vi /etc/resolv.conf
```
2. If any `nameserver` lines appear, write down the IP addresses for future reference.
3. Replace the `nameserver` lines with, or add, the following lines:

For IPv4:

```
nameserver 8.8.8.8
nameserver 8.8.4.4
```

For IPv6:

```
nameserver 2001:4860:4860::8888
nameserver 2001:4860:4860::8844
```

For IPv6-only, you can use Google Public DNS64 *instead of* the above IPv6 addresses.

4. Save and exit.
5. Restart any Internet clients you are using.
6. Test that your setup is working correctly; see Test your new settings.

Additionally, if you are using DHCP client software that overwrites the settings in `/etc/resolv.conf`, you need to configure the DHCP client by editing the client's configuration file.

#### Example: Configuring DHCP client software on a Debian server

1. Back up `/etc/resolv.conf`:  

```
sudo cp /etc/resolv.conf /etc/resolv.conf.auto
```
2. Edit `/etc/dhcp/dhclient.conf` (or `/etc/dhcp3/dhclient.conf`):  

```
sudo vi /etc/dhcp*/dhclient.conf
```
3. If there is a line following `request with only domain-name-servers`, remove that line.
4. If there is a line containing `domain-name-servers` with IP addresses, write down the IP addresses for future reference.
5. Replace that line with, or add, the following line:

For IPv4:

```
prepend domain-name-servers 8.8.8.8, 8.8.4.4;
```

For IPv6:

```
prepend domain-name-servers 2001:4860:4860::8888, 2001:4860:4860::8844;
```

For IPv6-only, you can use Google Public DNS64 *instead of* the above IPv6 addresses.

**Note:** ISC DHCP v4.3.6 or higher is required for IPv6 domain-name-servers to be processed; earlier versions would silently ignore such configuration.

6. Save and exit.
7. Restart any Internet clients you are using.
8. Test that your setup is working correctly; see Test your new settings.

Practical No : 09

Aim :- Setting IP address and connecting to internet.

Solution :-

## Configure Static IP Address in RHEL/CentOS/Fedora:

To configure static IP address in **RHEL / CentOS / Fedora**, you will need to edit:

```
/etc/sysconfig/network  
  
/etc/sysconfig/network-scripts/ifcfg-eth0
```

Where in the above "**ifcfg-eth0**" answers to your network interface **eth0**. If your interface is named "**eth1**" then the file that you will need to edit is "**ifcfg-eth1**".  
Let's start with the first file:

```
# vi /etc/sysconfig/network
```

Open that file and set:

```
NETWORKING=yes  
  
HOSTNAME=node01.tecmint.com  
  
GATEWAY=192.168.0.1  
  
NETWORKING_IPV6=no  
  
IPV6INIT=no
```

Next open:

```
# vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

**Note:** Make sure to open the file corresponding to your network interface. You can find your network interface name with [ifconfig -a command](#).  
In that file make the following changes:

```
DEVICE="eth0"  
  
BOOTPROTO="static"  
  
DNS1="8.8.8.8"  
  
DNS2="4.4.4.4"  
  
GATEWAY="192.168.0.1"
```

```
HOSTNAME="node01.tecmint.com"

HWADDR="00:19:99:A4:46:AB"

IPADDR="192.68.0.100"

NETMASK="255.255.255.0"

NM_CONTROLLED="yes"

ONBOOT="yes"

TYPE="Ethernet"

UUID="8105c095-799b-4f5a-a445-c6d7c3681f07"
```

You will only need to edit the settings for:

1. DNS1 and DNS2
2. GATEWAY
3. HOSTNAME
4. NETMASK
5. IPADDR

Other settings should have already been predefined.

Next edit `resolve.conf` file by opening it with a text editor such as [nano](#) or [vi](#):

```
# vi /etc/resolv.conf

nameserver 8.8.8.8 # Replace with your nameserver ip

nameserver 4.4.4.4 # Replace with your nameserver ip
```

Once you have made your changes restart the networking with:

```
# /etc/init.d/network restart    [On SysVinit]

# systemctl restart network     [On SystemD]
```

## Set Static IP Address in Debian / Ubuntu

To setup static IP address in **Debian/ Ubuntu**, open the following file:

```
# nano /etc/network/interfaces
```

You may see a line looking like this:



```
auto eth0

iface eth0 inet dhcp
```

Change it so it looks like this:

```
auto eth0

iface eth0 inet static

    address 192.168.0.100

    netmask 255.255.255.0

    gateway 192.168.0.1

    dns-nameservers 4.4.4.4

    dns-nameservers 8.8.8.8
```

Save the file and then edit `/etc/resolv.conf` like this:

```
# nano /etc/resolv.conf

nameserver 8.8.8.8 # Replace with your nameserver ip

nameserver 4.4.4.4 # Replace with your nameserver ip
```

Restart the networking on your system with:

```
# /etc/init.d/network restart    [On SysVinit]

# systemctl restart network      [On SystemD]
```

Your static IP address has been configured.