Unit 4: Linux Administration

4.1 Installing Linux
4.2 Installation of Open-Source Software
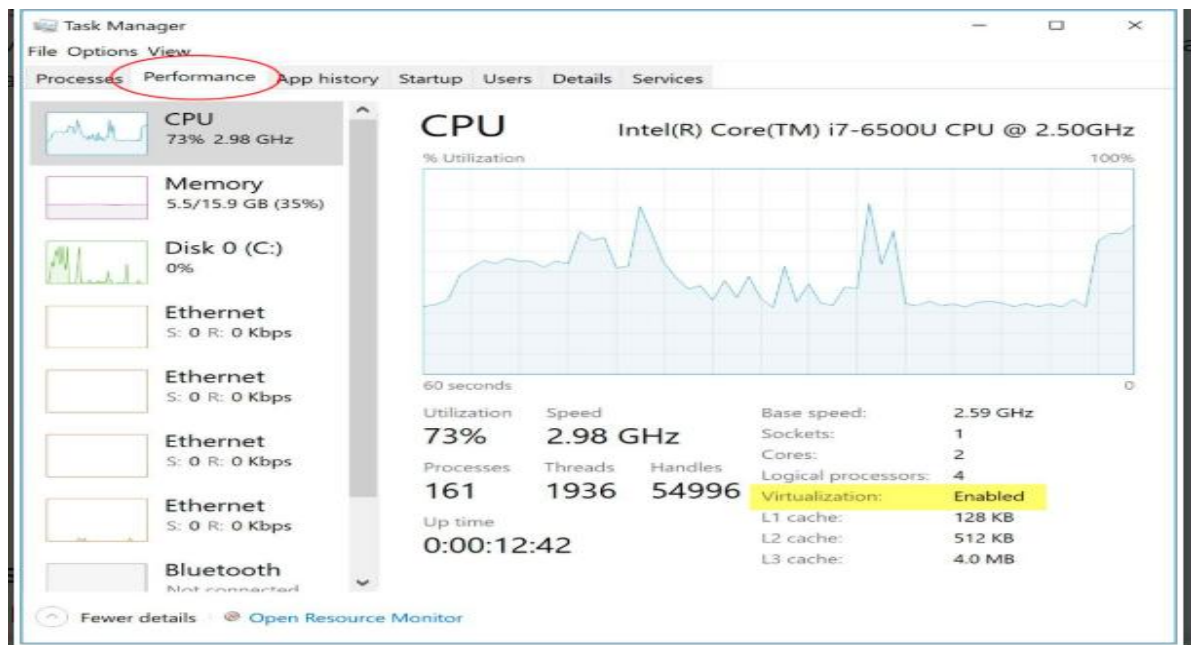**4.3 Maintaining User Accounts**
4.4 System Config Services (Package)

4.1 Installing Linux

(For Practical Point of view)
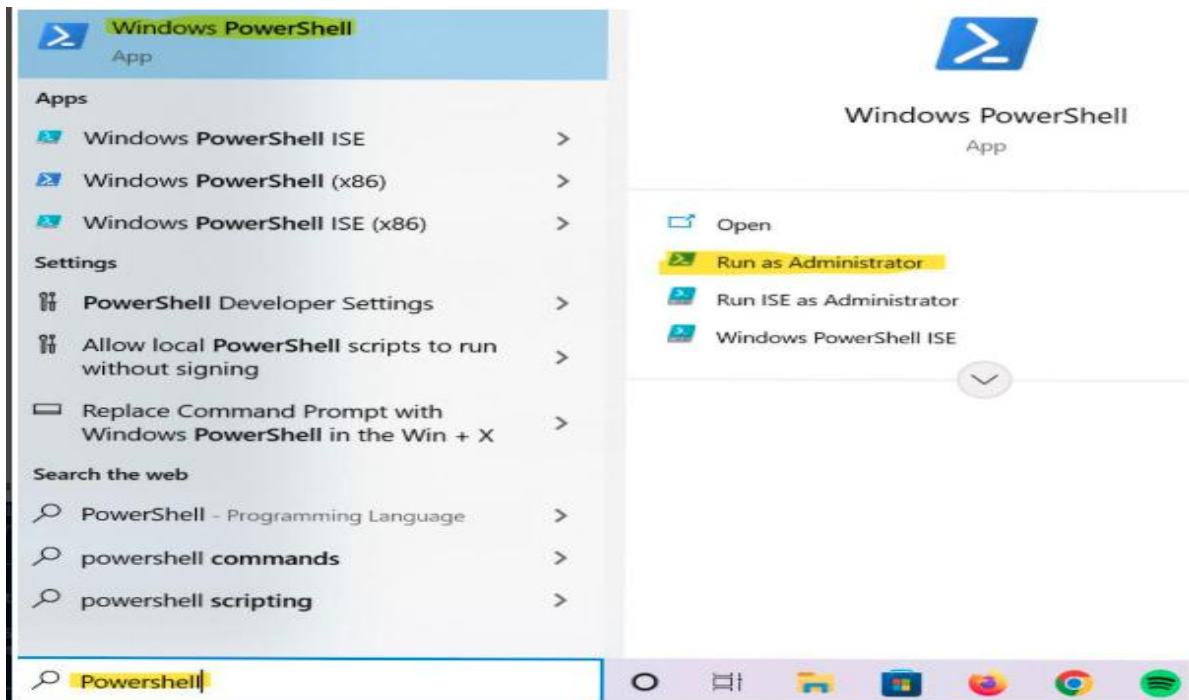**How to install and Run Linux Software in Windows 10?**
- Windows 10 OS with version 1903 or higher or Windows 11 OS.
- BIOS Level Virtualization support should be enabled, you can check it in Task Manager > Performance Tab.

Step 1: Enable WSL Feature.

To install WSL distros (Linux OS) we need to have the WSL feature enabled.
Open Powershell as Administrator and run the below command.

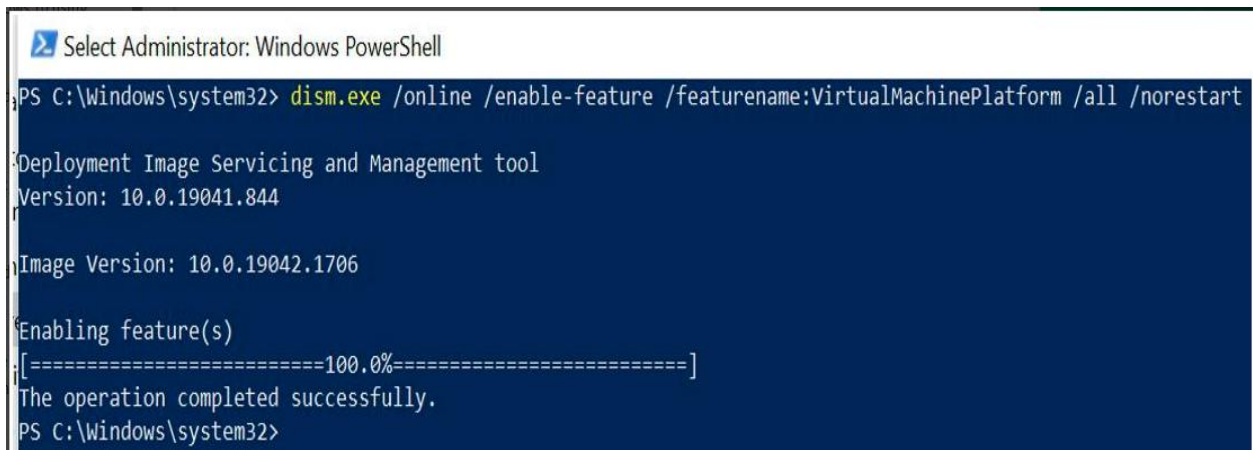*dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart*

Step 2: Enable Virtual Machine Platform.

WSL also depends on a Virtual Machine Platform, so we must enable it before installing the Linux distros (OS).

Open Powershell as Administrator and run the below command.

dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart

Step 3: Restart Your PC
Restart your PC to make the above changes to become active.

Step 4: Download and install the Linux kernel update package.
Updating the WSL kernel package is recommended before installing Linux distros, you might face issues if this is not up to date.
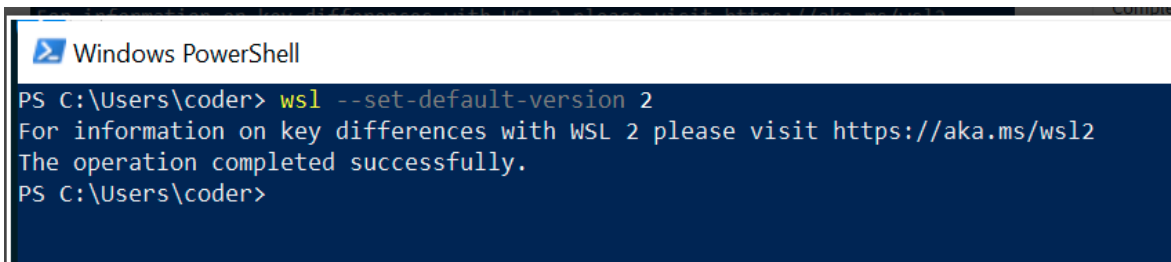
- You can download the Linux kernel update package from here.
- Double-click the downloaded file to update the kernel.
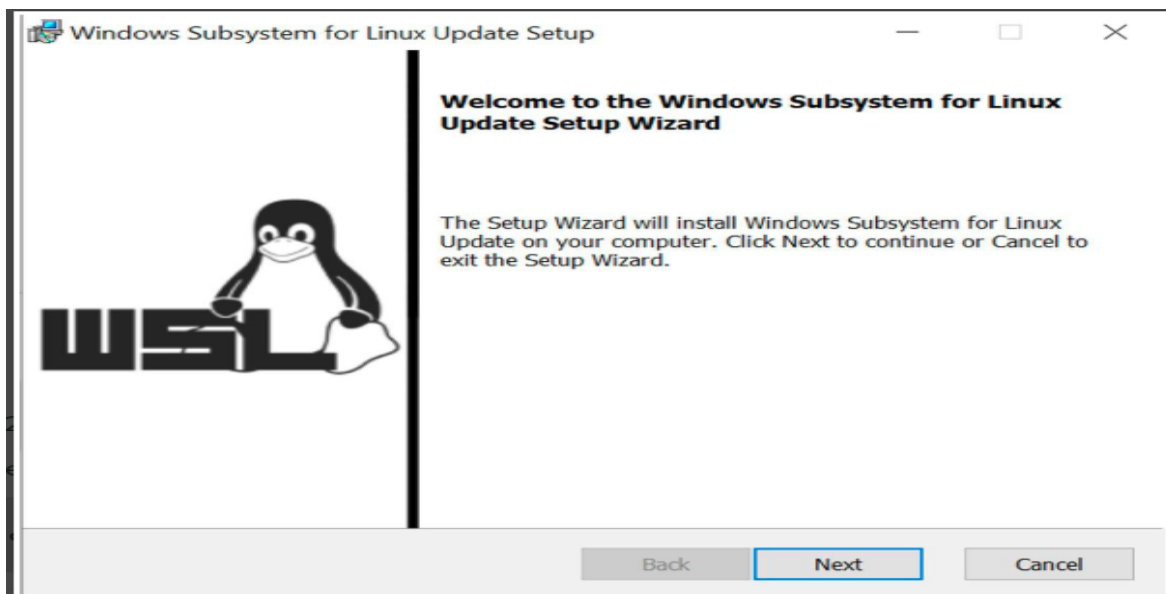
Step 5: **Set v2 as the WSL default version.**
WSL2 is preferred over version 1 for better performance.

```
wsl --set-default-version 2
```

Step 6: Install Linux distribution (Ubuntu).
Now we can install the required Linux distros, here in this example where we are installing
Ubuntu 20.04.

- Open Powershell and run the below command.

wsl --install -d Ubuntu-20.04

- Once installation is completed, open "Ubuntu" from the start menu.

- Set username, and password for WSL environment (Ubuntu).

Step 7: Run sample commands.

(For Theory Point of view)

Write the step by step installation of Linux in Ubuntu

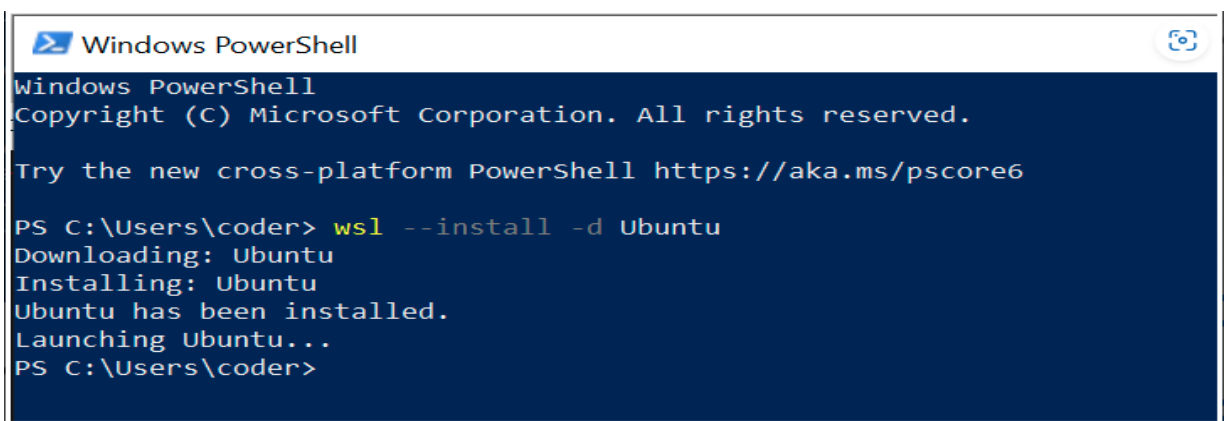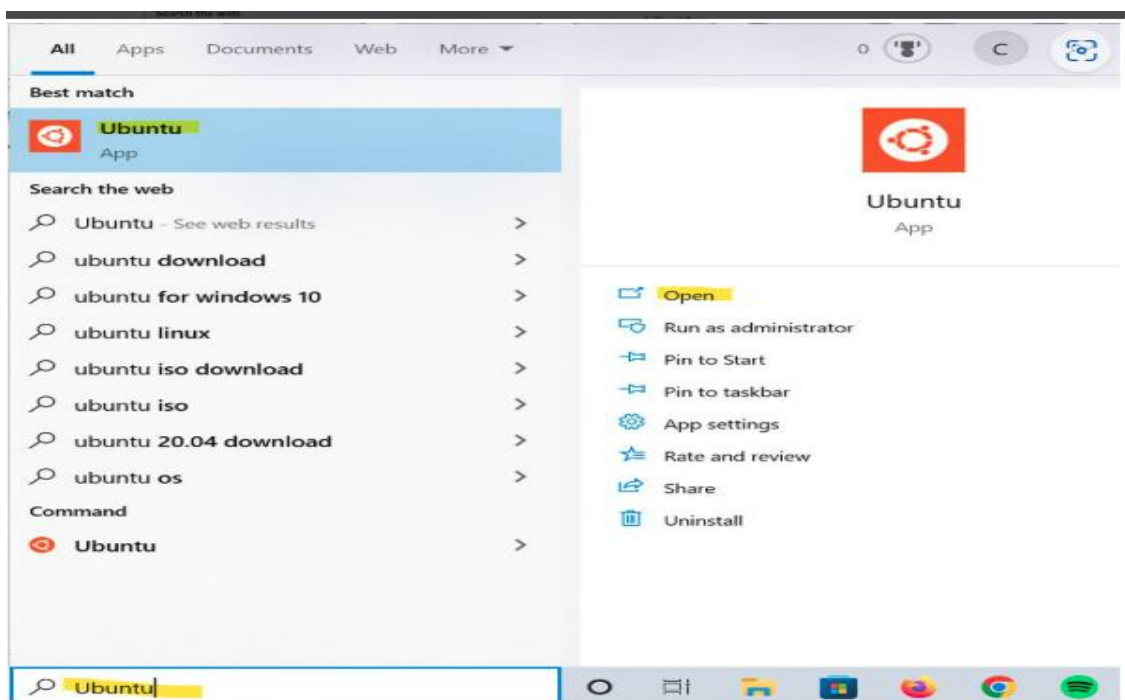If you want to install a new Linux distribution or even another version of Ubuntu on your existing Ubuntu system, follow these step-by-step instructions to do so effectively. This guide assumes you want to **install a fresh version of Linux**, not just update or upgrade your current Ubuntu installation.

Step-by-Step Guide: Installing Linux on Ubuntu

Step 1: Download the Linux Distribution ISO

1. **Choose a Linux Distribution**: For this guide, we'll assume you're installing Ubuntu, but you can follow similar steps for other distributions like Linux Mint, Fedora, etc.
2. **Download the ISO**:
    o Go to the official Ubuntu website: <u>Download Ubuntu</u>.
    o Download the appropriate version (usually, the **desktop** version for most users).

Step 2: Create a Bootable USB Drive on Ubuntu

To install Linux, you will need to create a bootable USB stick that you can boot from to start the installation process.

1. **Insert a USB Drive**: Insert a USB stick with at least 4 GB of space into your computer.
2. **Open Startup Disk Creator**:
    o Press the **Super key (Windows key)**, search for **Startup Disk Creator**, and open it.
3. **Select the ISO File**:
    o In the Startup Disk Creator, click **Other** to select the downloaded ISO file of the Linux distribution.
4. **Select the USB Drive**:
    o Choose your USB stick from the list of available drives.
5. **Create the Bootable Disk**:
    o Click **Make Startup Disk** to begin creating the bootable USB.
    o Wait for the process to complete (it may take several minutes depending on the USB speed).

Step 3: Backup Your Data (If Necessary)

- If you are installing Linux on a system with existing data or another OS (e.g., Windows), **backup your important files** before proceeding. Installing Linux may erase your data if you're not careful.

Step 4: Restart and Boot from USB

1. **Restart Your Computer**: Insert the bootable USB drive and restart your computer.
2. **Access BIOS/UEFI**:

- o As your computer restarts, press the BIOS/UEFI access key (this is often **F2**, **F10**, **DEL**, or **ESC**, depending on the manufacturer).
- o Consult your laptop/PC manual if you're unsure which key to press.
3. **Set USB as Boot Device**:
    - o In the BIOS/UEFI menu, go to the **Boot Options** section.
    - o Select the **USB drive** as the primary boot device.
4. **Save and Exit**: Save the settings and exit the BIOS. Your system should now boot from the USB drive.

Step 5: Begin the Installation Process

1. **Select "Install Ubuntu"**:
    - o When the system boots from the USB drive, you'll see a menu. Choose the option **"Install Ubuntu"** (or the appropriate option for your Linux distro).
2. **Choose Language**:
    - o Select the language you want to use during the installation (e.g., English) and click **Continue**.

Step 6: Choose Keyboard Layout

1. Select your **keyboard layout** (most users can choose **English (US)**).
2. Click **Continue**.

Step 7: Set Up Network Connection

1. **Connect to Wi-Fi (if needed)**:
    - o If you're using a laptop, connect to a Wi-Fi network by selecting it and entering the password.
    - o If you're using a wired Ethernet connection, it should be automatically detected.
2. Click **Continue**.

Step 8: Choose Installation Type

1. **Erase Disk and Install Linux**:
    - o This option will **erase all data** on your disk, so make sure to back up everything important before proceeding.
    - o Choose this option if you want to perform a **clean installation** of Linux.
2. **Install Alongside Existing OS (Dual Boot)**:
    - o If you want to **dual-boot** Linux alongside your existing OS (such as Windows), choose this option.
    - o The installer will resize your existing partitions and install Linux alongside your current system.
3. **Manual Partitioning ("Something Else")**:
    - o If you want to manage partitions manually, choose the **"Something else"** option.
    - o Here, you can create and modify partitions according to your needs, such as creating separate partitions for /, /home, and swap space.
4. **Confirm Your Selection**:

o After selecting your preferred installation type, click **Install Now**. A warning will appear asking if you're sure you want to proceed (as this could erase data), so confirm by clicking **Continue**.

Step 9: Choose Your Time Zone

1. The installer will ask you to select your **time zone**.
    o You can either click on your location on the map or type the city in the search box.
2. Click **Continue**.

Step 10: Create Your User Account

1. Enter your **name**, **username**, and **password**. This will be your login information.
2. You can also select whether you want **automatic login** (no password needed on startup) or require the password every time you log in.
3. Click **Continue**.

Step 11: Wait for the Installation to Complete

- The installation process will begin. This may take 10-30 minutes depending on your system's speed.
- The installer will copy files and set up the system. During this time, you may see additional prompts about installing software, third-party drivers, or updates.

Step 12: Restart Your Computer

1. Once the installation finishes, you'll be asked to **remove the installation media** (the USB drive).
2. **Click "Restart Now"**.
3. Your system will restart and boot into the newly installed Linux OS.

Step 13: First Boot into Linux

1. **Log in** using the username and password you created during the installation.
2. After logging in, you'll be in your new Linux environment. Explore the system, and make sure everything is working as expected.

Step 14: Update Your System

1. It's a good idea to update your system to the latest software versions and security patches.
2. Open a **Terminal** (press **Ctrl** + **Alt** + **T**), and run the following commands:

sudo apt update
sudo apt upgrade

    o This will update your package list and install any available updates.

4.2 Installation of Open Source Software

Once you have decided to migrate to open source software, you will need to do some basic installing. Installing open source software depends on your operating system. This is a how-to compilation for multiple operating systems; read the appropriate section for your OS.
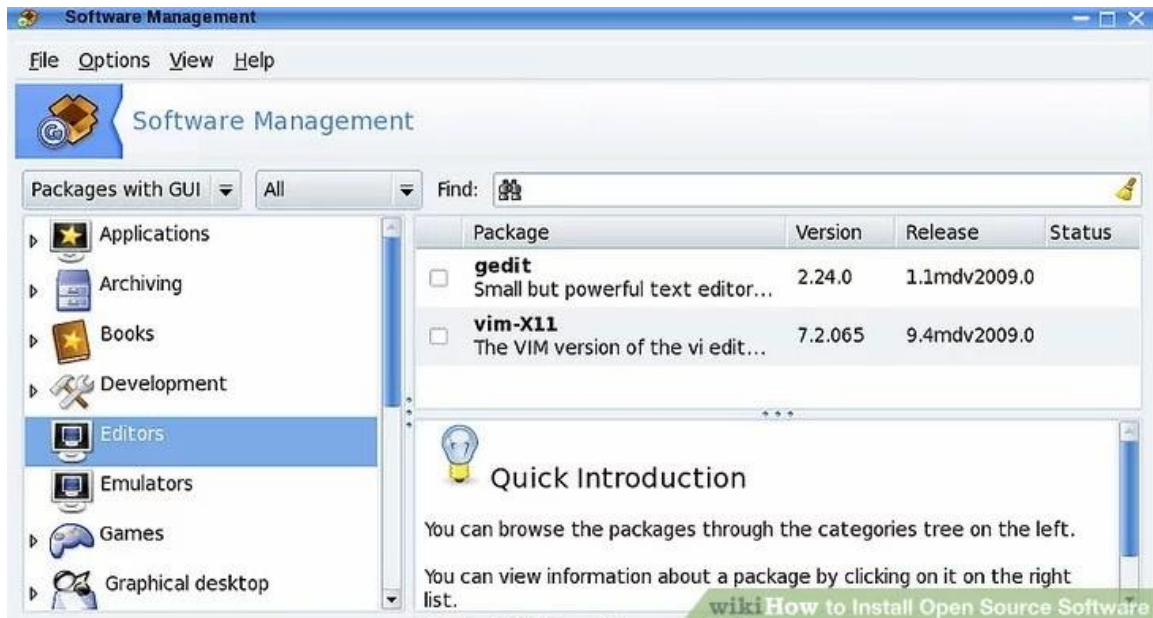
Linux/Unix/Unix-Like Systems

For most such systems, you can probably use the OSs package manager to install a pre-built binary package. This is always the recommended method.

Alternatively, you could follow these steps:

- o Download and uncompress the source code.
- o In the terminal, move into the extracted directory.
- o Run "./configure" to configure the software.
- o Run "make" to compile the software.
- o Run "make install" to install the software.

4.3 Maintaining User Accounts

Linux user

- A user or account of a system is uniquely identified by a numerical number called the UID (unique identification number).
- There are two types of users –

1. The root or super user.
2. Normal users.

- A root or super user can access all the files, while the normal user has limited access to files.
- A super user can add, delete and modify a user account. The full account information is stored in the /etc/passwd file and a hash password is stored in the file /etc/shadow. Some operations on a user account are discussed below.
- Creating a user with a default setting: A user can be added by running the user add command at the command prompt. After creating the user, set a password using the passwd utility.
    o The system automatically assigns a UID, creates the home directory (/home/<username>) and sets the default shell to /bin/bash.
    o The useradd command creates a user private group whenever a new user is added to the system and names the group after the user.
- Locking and unlocking a user: A super user can lock and unlock a user account.
    o To lock an account, one needs to invoke passwd with the -l option.
    o To unlock an account, one needs to invoke passwd with the -u option.
- Changing a user name: The –l option with the usermod command changes the login (user)
- Removing a user: Combining userdel with the –r option drop a user and the home directory associated with that user.

Linux group
- Linux group is a mechanism to organize a collection of users. Like the user ID, each group is also associated with a unique ID called the GID (group ID).
- There are two types of groups –

1. a primary group
2. a supplementary group.

- Each user is a member of a primary group and of zero or 'more than zero' supplementary groups.
- The group information is stored in /etc/group and the respective passwords are stored in the /etc/gshadow file.
- Some operations such as creating, deleting and modifying on a group are below.
- Creating a group with default settings: To add a new group with default settings, run the groupadd command as a root user.
- If you wish to add a password, then type gpasswd with the group name.
- **Creating a group with a specified GID**: To explicitly specify the GID of a group, execute the groupadd command with the –g option.

- **Removing group password:** To remove a group password, run gpasswd –r with the relevant group name
- **Changing the group's name:** To change the group's name, run the groupmod command with the -n option as a super user
- **Changing the group's GID:** To change the GID of a group, run the groupmod command with –g
- **Deleting a group:** Before deleting a primary group, delete the users of that primary group. To delete a group, run the groupdel command with the group name

4.4 System Config Services (Package)

- Name: system-config-services - Service Configuration Utility
- Synopsis: system-config-services
- Description : This is a graphical tool for enabling and disabling services (including xinetd services). Functionality to start, stop, and restart services is also included.
- Options :None
- Files:/usr/bin/system-config-services
  - o /usr/share/system-config-services/*
- To run this program simply type: system-config-services
- Bug :
  - o Some services will not start or stop properly if started anywhere but the console (system-config-services will appear to hang in these instances). This is not a bug in system-config-services, but in the individual services.
  - o Some services are incredibly hard to detect if they are running or not. While there are workarounds present to deal with these, it can't be guaranteed that they're detected properly. Please file bugs against the system-config-services component at http://bugzilla.redhat.com if you encounter such services.
  - o Some configuration files run a set of commands upon startup. A common convention is for such files to have "rc" in their name, typically using the name of the program then an "(.)rc" suffix e.g. ".xinitrc", ".vimrc", ".bashrc", "xsane.rc". S

There are various methods for managing access to system services:
a)   /etc/init.d/service
b)   rcconf
c)   update-rc.d etc