**Linux practical**

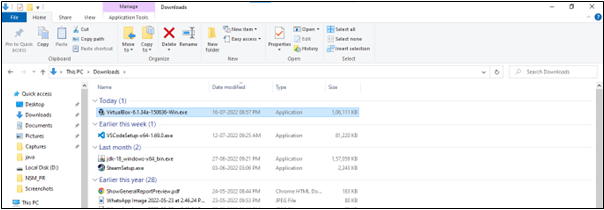
**Practical 0: Installation of RHEL 6.X**

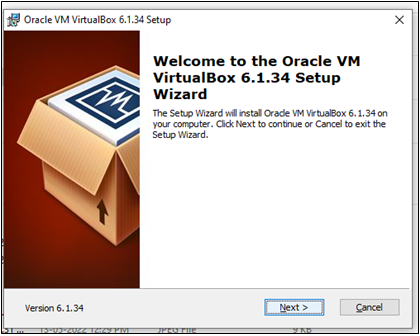
**Installation of Virtual Box.**

**Step 1: Download the VMWARE WORKSTATION 11.0 software from the given link.** [**https://www.virtualbox.org**](https://www.virtualbox.org)

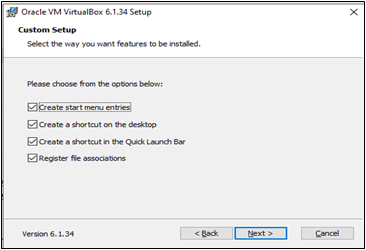


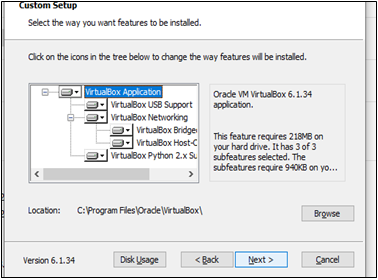
**Step 2: Double click on . exe file . It will show you following dialog box. Click on next button.**





Step 3: Select all the options for getting more features and then click on next.

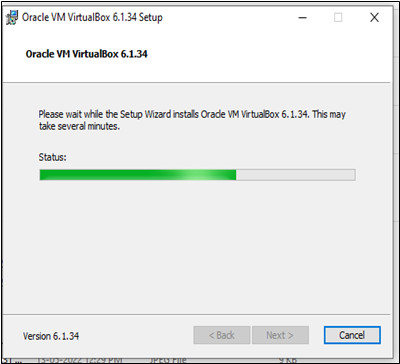
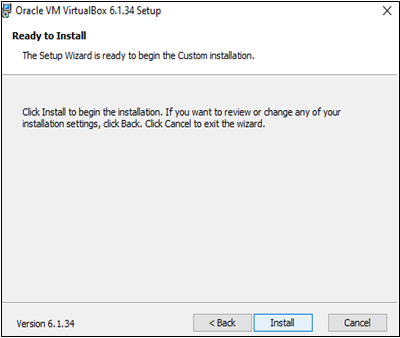




**Step 4: Click on yes button to install the VM VirtualBox software.**

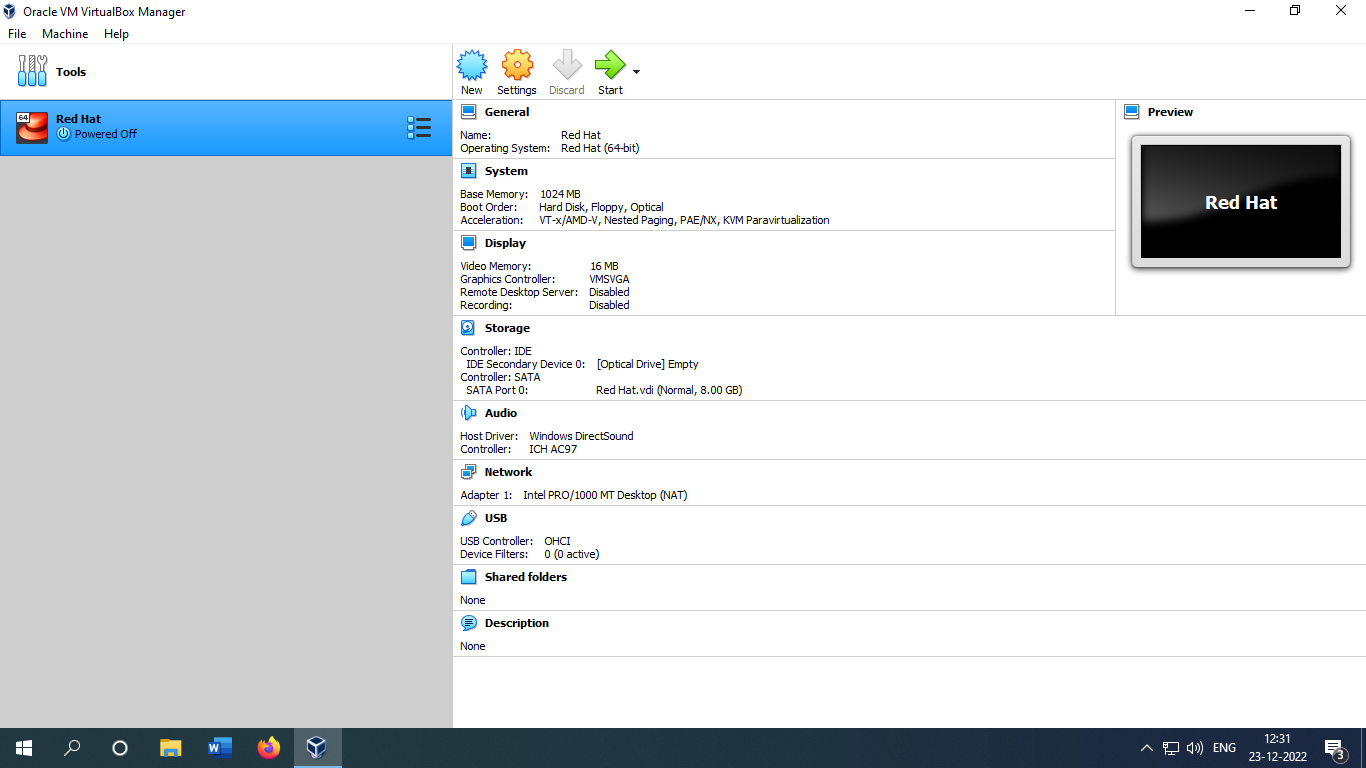


**Step 5: Click on install to start installation process.**

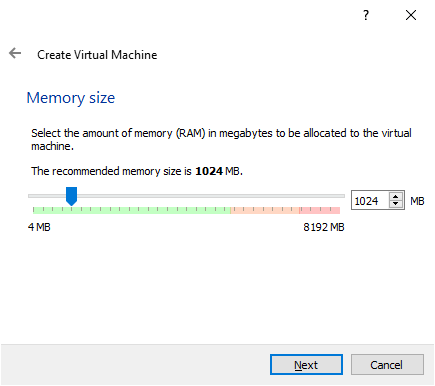
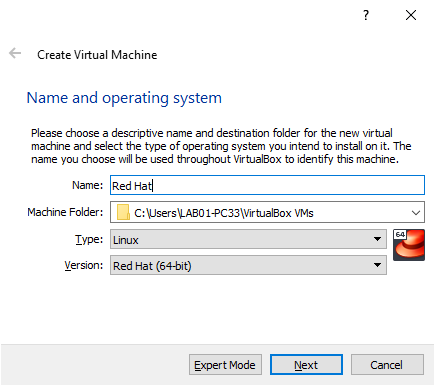


**Steps to install Red hat on virtual box.**

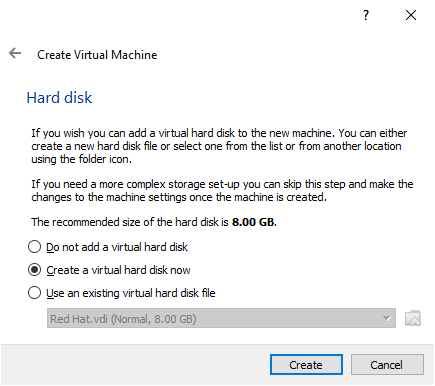
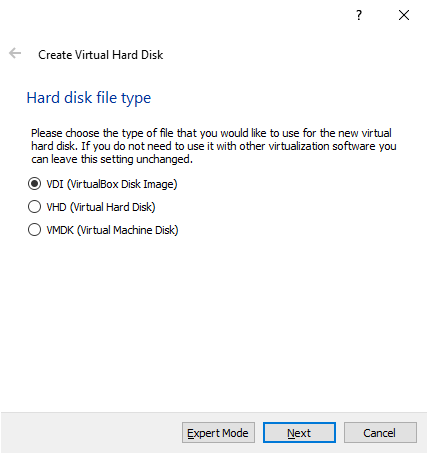
Step 1: click on New to add new virtual machine.



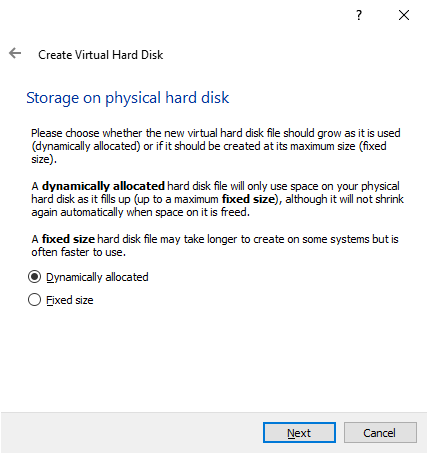
Step 2: Select the name and memory size.

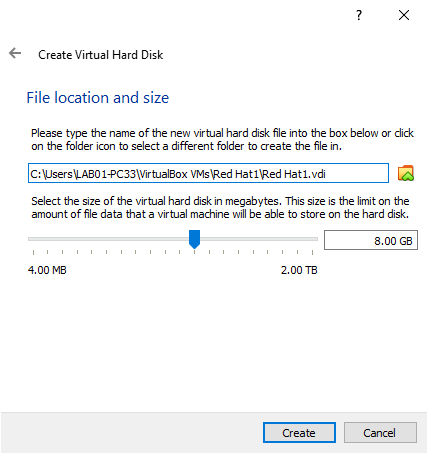


Step 3: select the hard disk and it’s file type to create virtual machine.

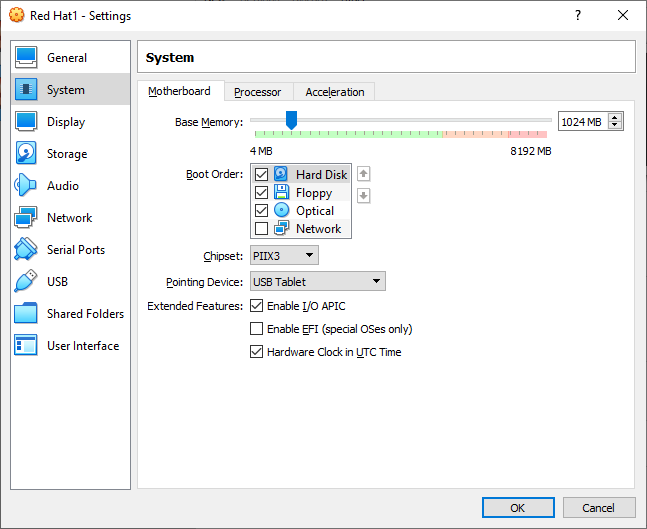
it

Step 4: Select storage on hard disk dynamically allocated and select the executable file location and memory size.

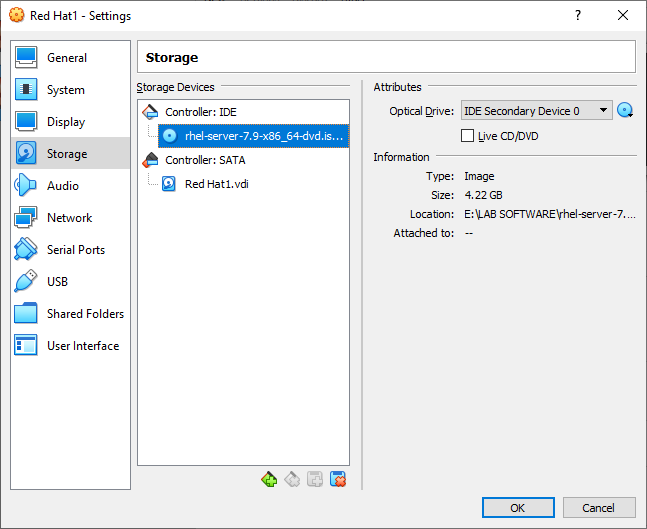




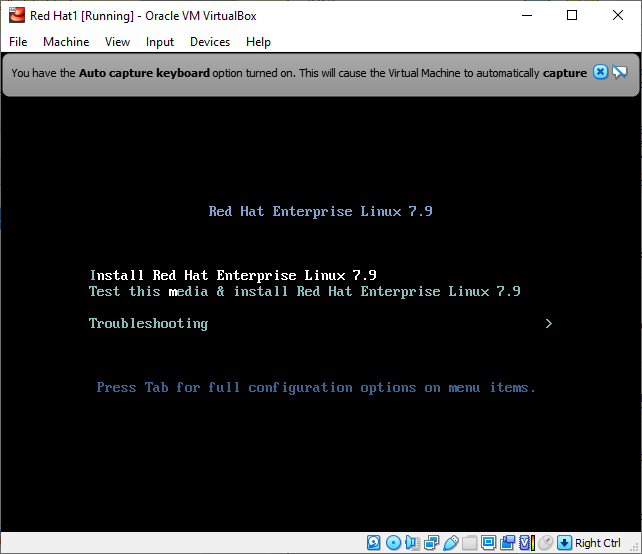
Step 5: Now go to settings option, select the memory size and hard disk as well as pointing device USB tablet and click on OK.

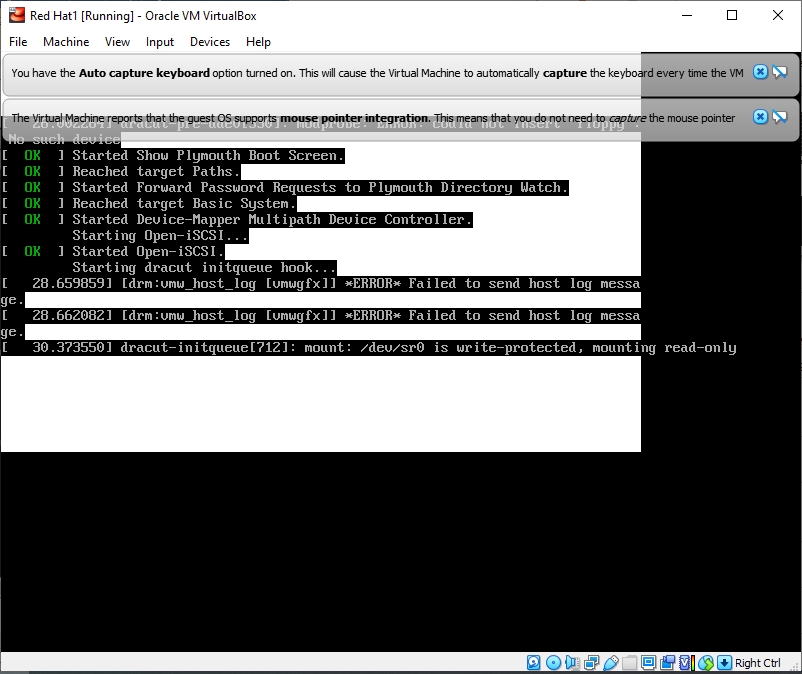


Step 6: Now go to the storage section in the empty file select your red hat executable file and click ok.

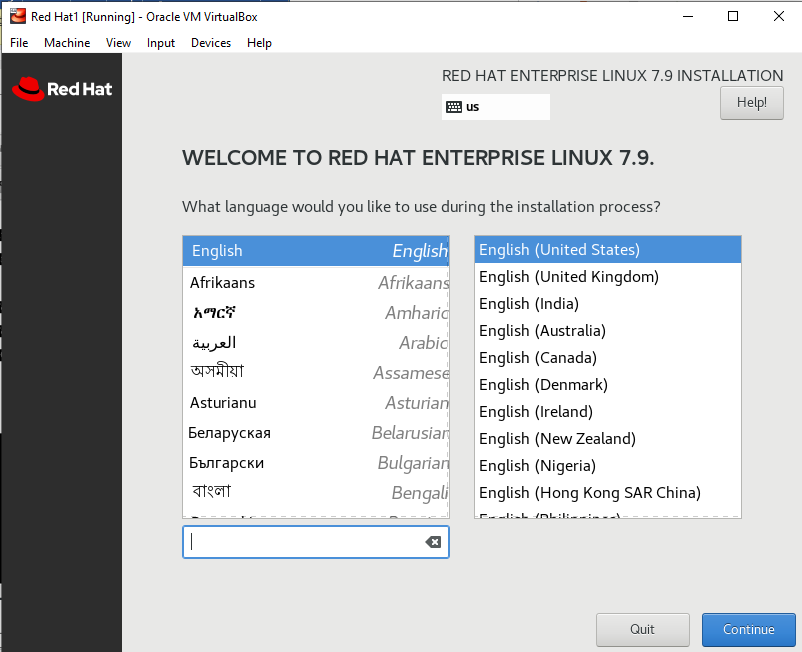


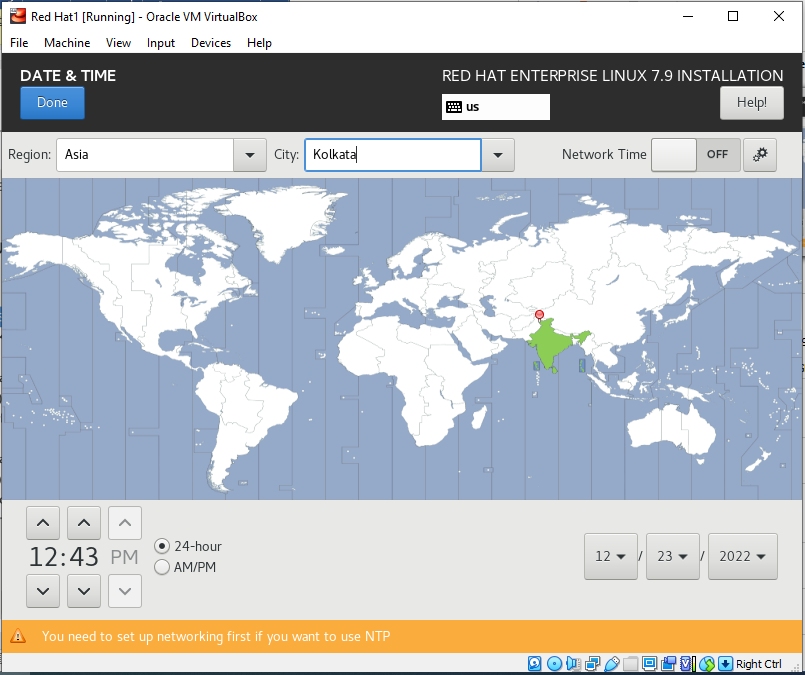
Step 7: click on start (green arrow) to start red hat virtual machine.

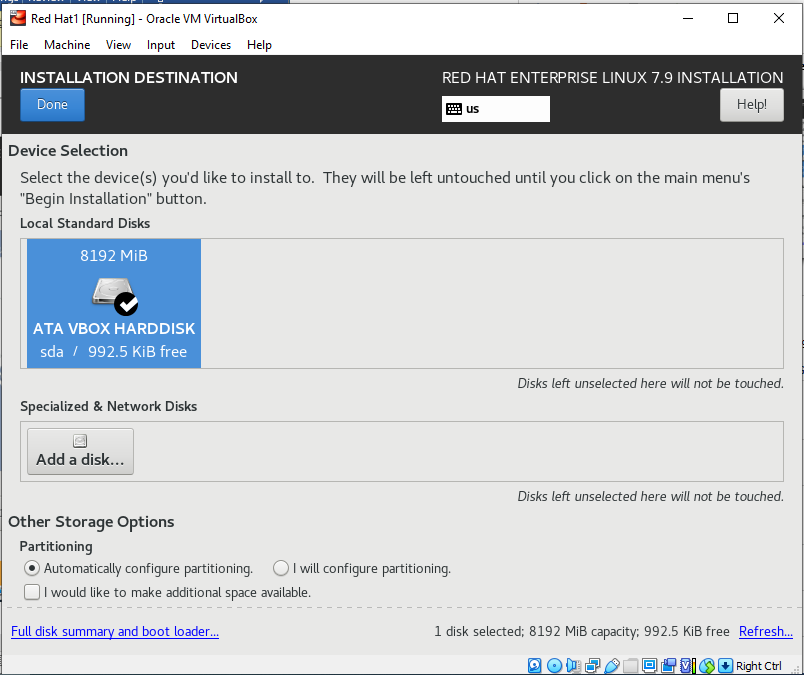


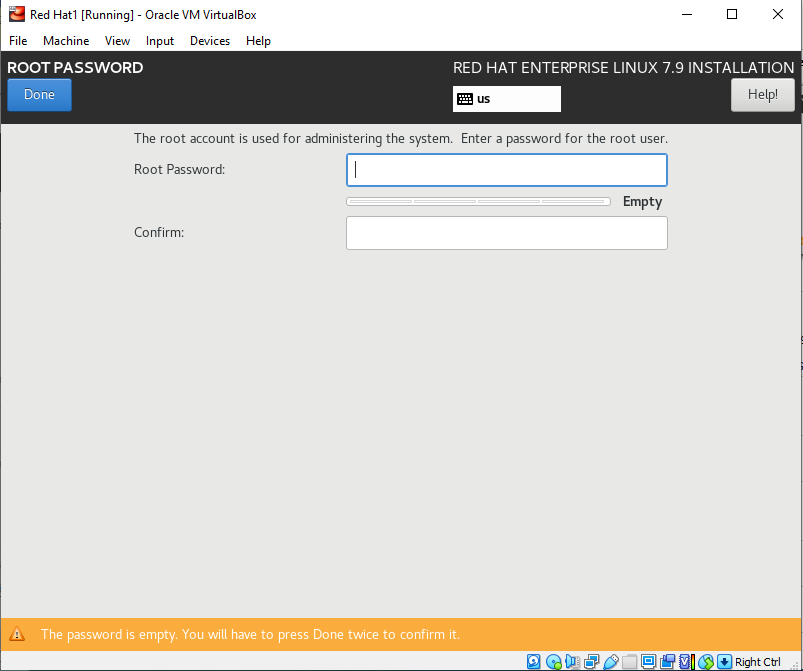


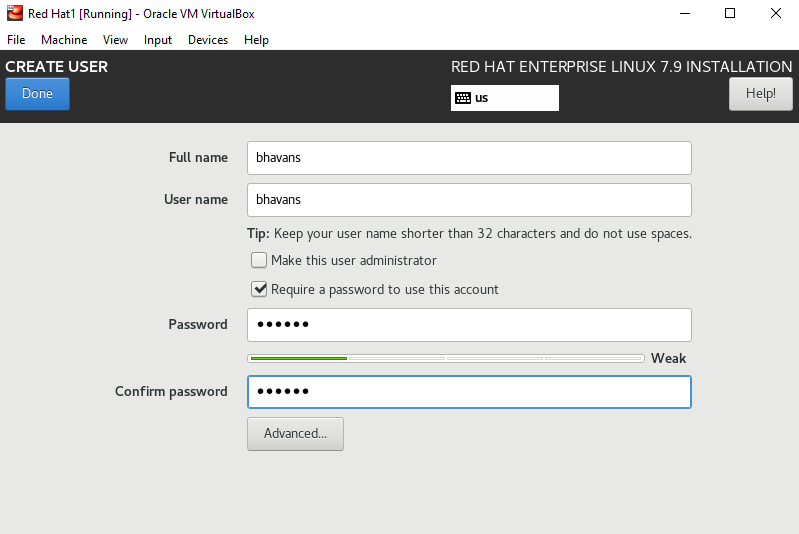
Step 8: select your language (English) and location (mumbai) and click on continue.

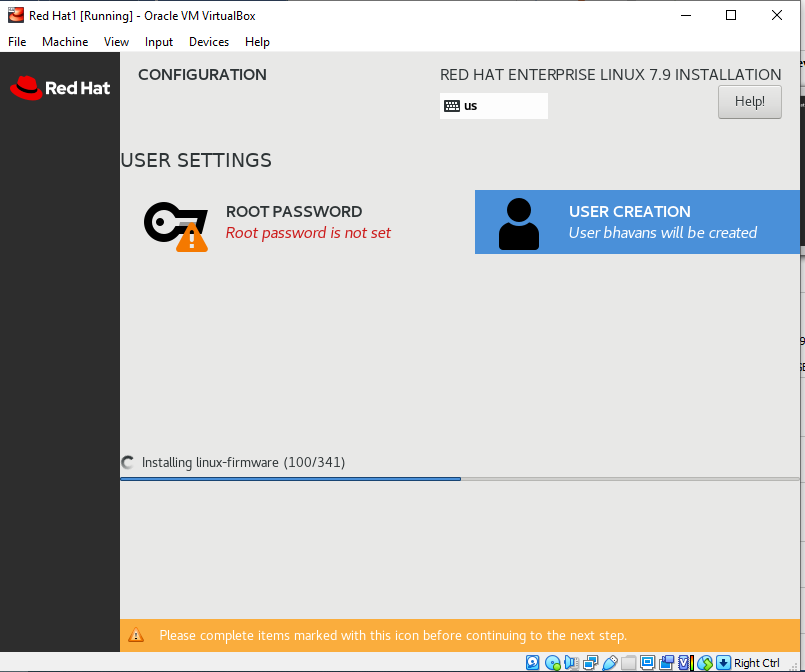


Step 9: select your installation destination and click on Done.

Step 10: Create root password and user account for your system.





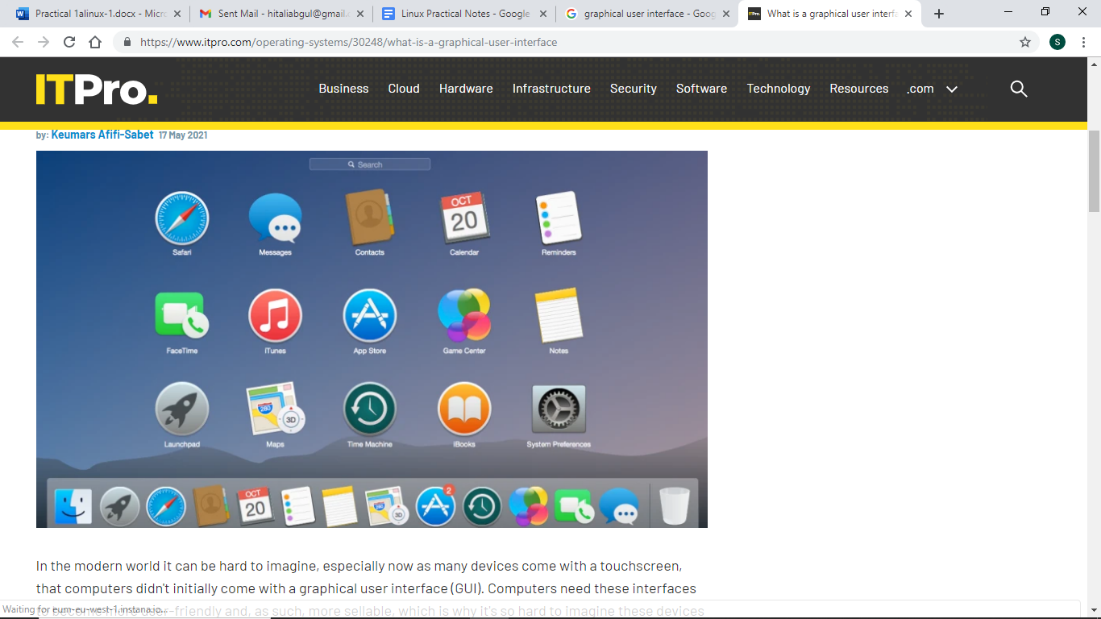


**Practical 1**: **Graphical User Interface and Command Line**

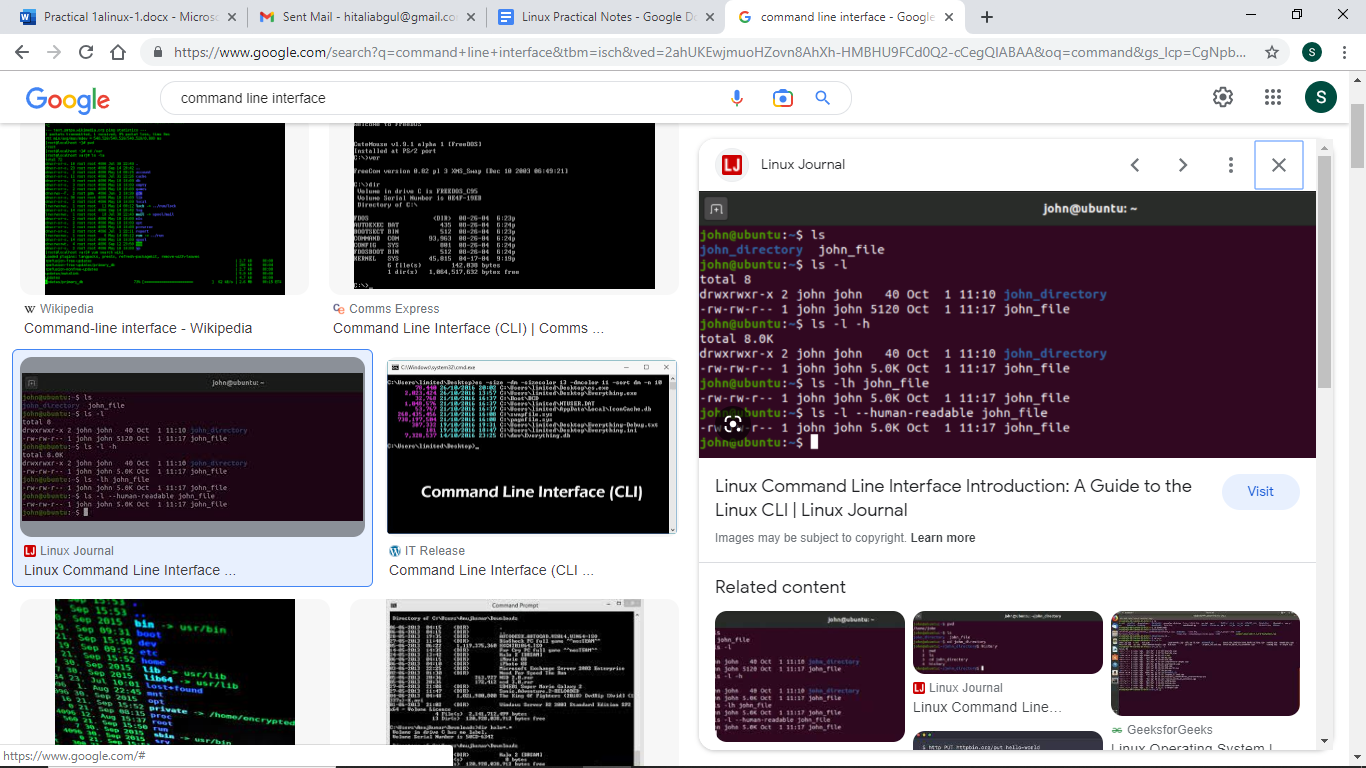
1. **Exploring the Graphical Desktop**

**Definition: A GUI is a human-computer interface that uses windows, icons and menus which can be manipulated by a mouse and often to a limited extent by a keyboard.**

**Example: MS DOS**



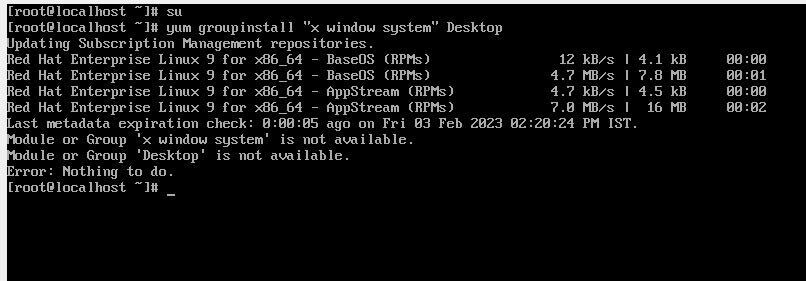
**GUIS stand in sharp contrast to command line interfaces (CLIS) which use only text and are accessed solely by a keyboard.**



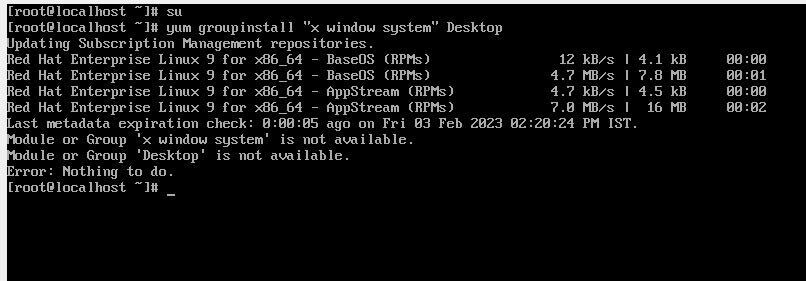
**1. If you are not already root, switch users to the root account:**

**# su**

1. It is possible to enter the system as the root user either for a series of operations or only for one. ...
2. In order to temporarily sign on as another user, you can use the su command.

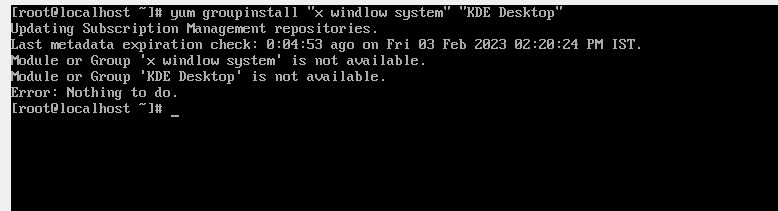
  
**2. If you have not already done so, install the X Window System and a graphical desktop environment. For example, to install the GNOME desktop environment, use this command:**

# yum groupinstall “X window system” Desktop



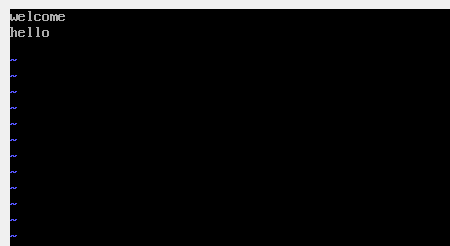
**To install the KDE desktop environment, use:**

# yum groupinstall “X window system” “KDE Desktop”



**3. Run the following command to edit the /etc/inittab file:**

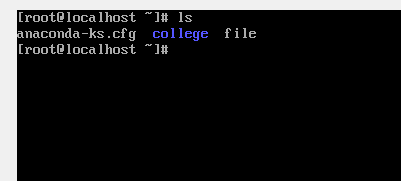
Vi /etc/initab



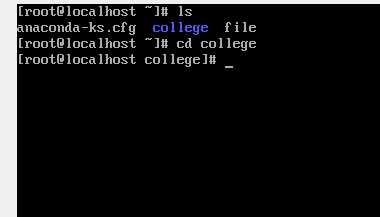
**Practical 1b: Command Line Interface**

**Definition: Abbreviated as CLI, a Command Line Interface connects a user to a computer program or operating system. Through the CLI, users interact with a system or application by typing in text (commands). The command is typed on a specific line following a visual prompt from the computer.**

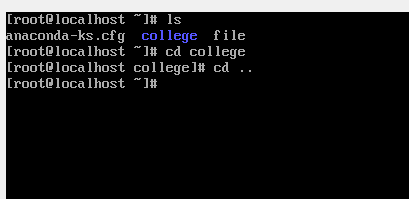
Ls: The ls command is one of the most commonly used commands in daily Linux/UNIX operations. The command is **used in listing contents inside a directory** and is one of the few commands beginners learn from the onset.



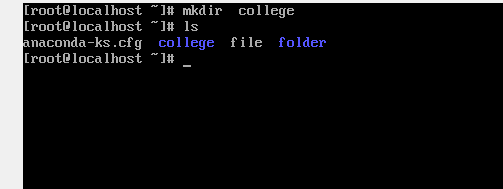
Cd: Linux cd command is **used to change the current working directory** ( i.e., in which the current user is working). The "cd" stands for 'change directory. ' It is one of the most frequently used commands in the Linux terminal.



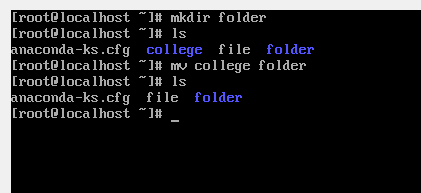
Cd ..: This command is use to go to your home directory.



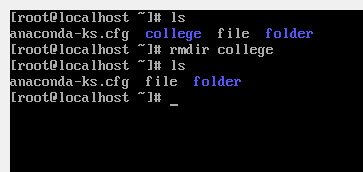
Mkdir: The mkdir stands for 'make directory'. With the help of mkdir command, you can create a new directory wherever you want in your system. Just **type "mkdir <dir name> , in place of <dir name> type the name of new directory, you want to create and then press enter**



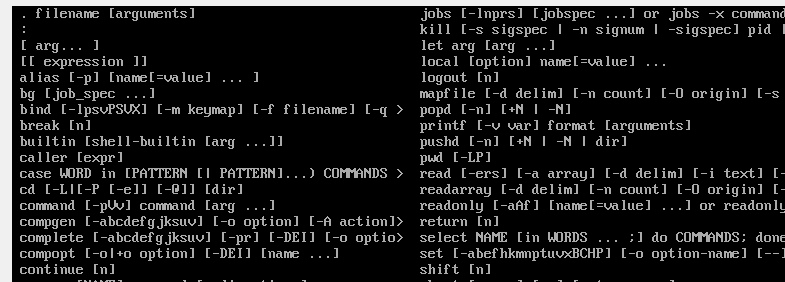
Move: mv stands for move. mv is **used to move one or more files or directories from one place to another in a file system like UNIX**.



Rmdir: The rmdir command **removes the directory, specified by the Directory parameter, from the system**. The directory must be empty before you can remove it, and you must have write permission in its parent directory. Use the ls -al command to check whether the directory is empty.



Help: The help command is a Command Prompt command that's used **to provide more information on another command**. You can use the help command at any time to learn more about a command's usage and syntax, like which options are available and how to actually structure the command to use its various options.



**Practical 1c: Managing Processes**

**Definition: A process in Linux is nothing but a program in execution. It's a running instance of a program. Any command. that you execute starts a process.**

**Types of Processes in Linux:**

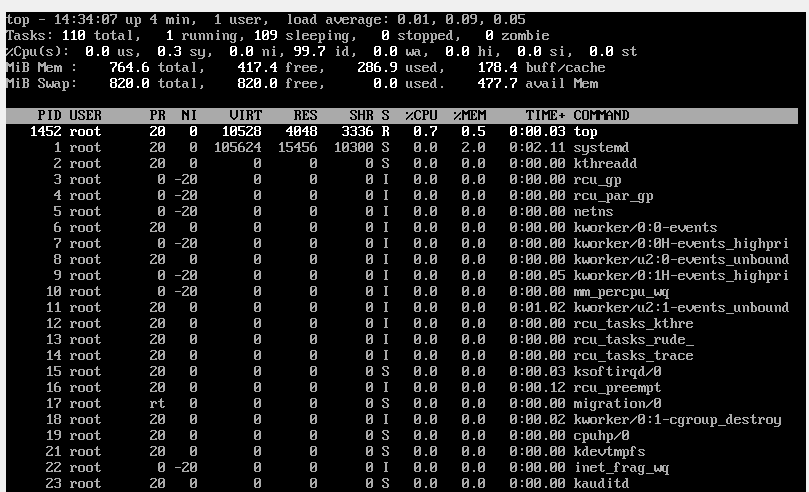
**1. Foreground Processes: depend on the user for input also referred to as interactive processes.**

**2. Background Processes**

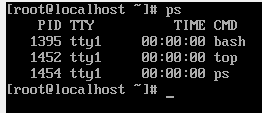
**run independently of the user.**

**Different Commands for Process Management in Linux:**

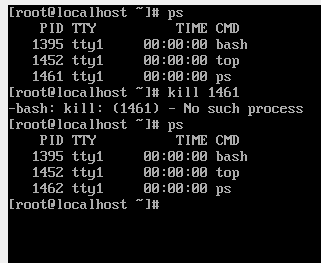
1. **Top:** The top utility is a commonly used tool for **displaying system-performance information**. It dynamically shows administrators which processes are consuming processor and memory resources. Top is incredibly handy.



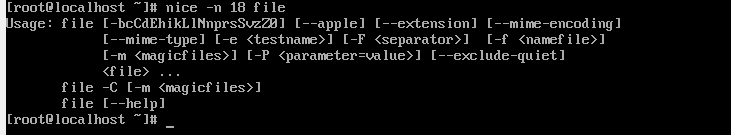
2. Ps: The ps command is **used to view currently running processes on the system**. It helps us to determine which process is doing what in our system, how much memory it is using, how much CPU space it occupies, user ID, command name, etc.



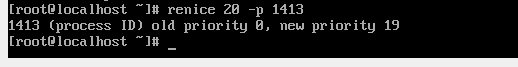
3. The kill command **sends a signal (by default, the SIGTERM signal) to a running process**. This default action normally stops processes. If you want to stop a process, specify the process ID (PID) in the ProcessID variable.



4. Nice: The nice command **lets you run a command at a priority lower than the command's normal priority**. The Command parameter is the name of any executable file on the system. If you do not specify an Increment value the nice command defaults to an increment of 10.



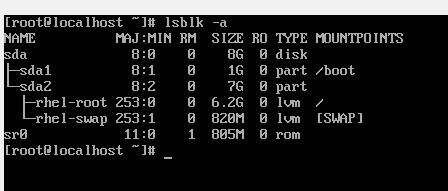
5. Renice: renice **alters the scheduling priority of one or more running processes**.



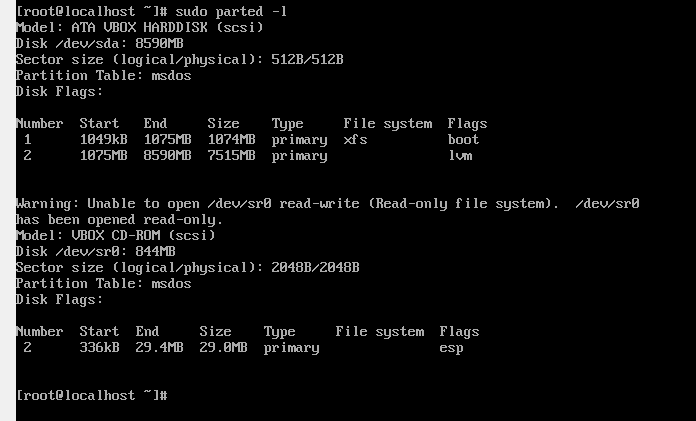
**Practical 2: Storage Devices and Links, Backup** and **Repository**

**a. Working with Storage Devices and** Links

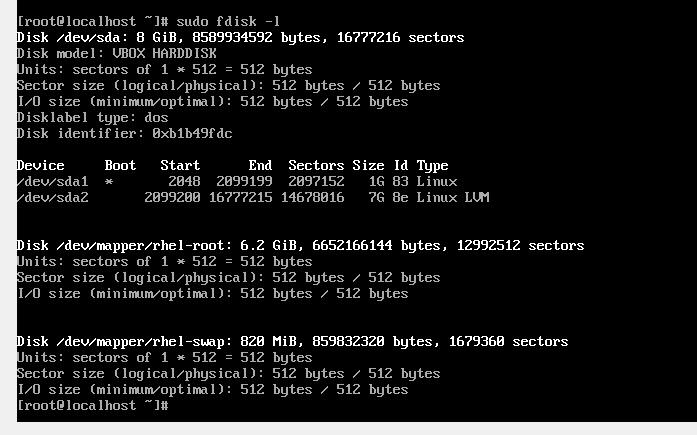
**Lsblk:** lsblk **lists information about all available or the specified block devices**. The lsblk command reads the sysfs filesystem and udev db to gather information.



**Parted –l:** Select the hard disk to be partitioned. Select the disk on which the partition is being created.



**Fdisk –l:** fdisk is **a menu driven command-line utility that allows you to create and manipulate partition tables on a hard disk**.



**Practical 2b: Making a backup**

The GNU tar (short for Tape ARchiver) command is the most widely used archiving utility in Linux systems. Available directly in the terminal, the tar command **helps create, extract, and list archive contents**.

**The general syntax for the tar command is as follows:**

**tar [OPERATION\_AND\_OPTIONS] [ARCHIVE\_NAME] [FILE\_NAME(s)]]**

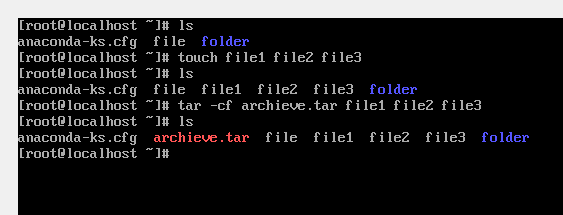
**1. OPERATION - Only one operation argument is allowed and required. The most frequently used operations are:**

**2. OPTIONS- The most frequently used operations are:**

**3. ARCHIVE NAME - The name of the archive.**

**4. FILE\_NAME(s) - The name of the file.**

**For example, to create an archive named archive.tar from the files named file1, file2, file3, you would run the following command: tar -ef archive.tar file1 file2 file3**

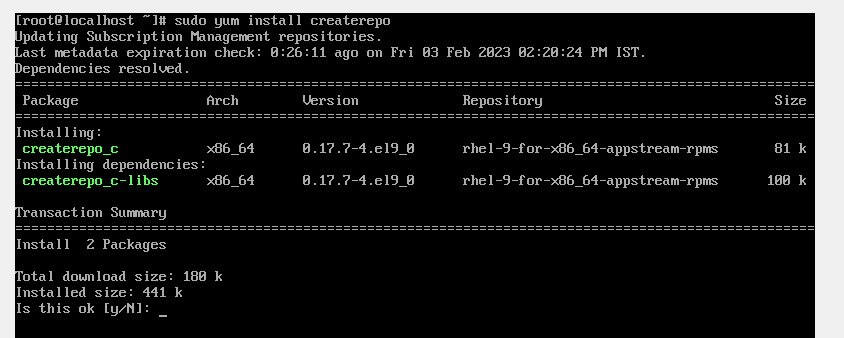


**Practical 2c: creating repository.**

**1. Install createrepo utility**

**To create a yum repository we need to install additional software called "createrepo":**

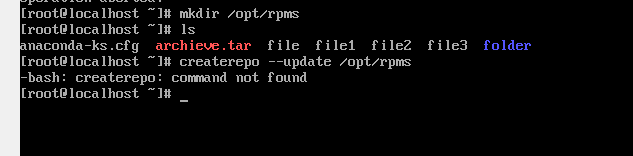
**# sudo yum install createrepo**



**2. Create a repository directory**

**You need to create a new directory that will be the location of your yum repository and will hold the desired rpm package files.**

**$ mkdir /opt/rpms**



**Practical 3: Working** with **RPM's Storage and Networking**

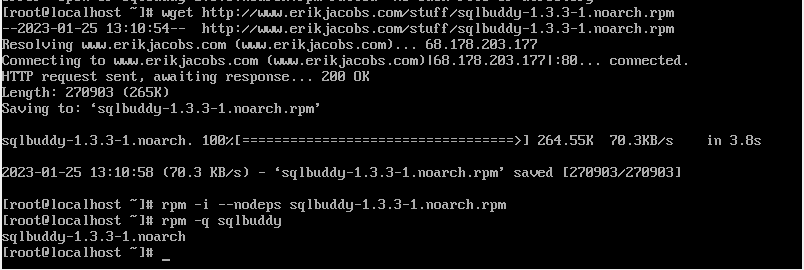
**a. Using Query Options**

**For installing an rpm software package, use the following command with -i option. For**

**example, to install an rpm package called sqlbuddy-1.3.3-1.noarch.rpm.**

**1. The basic syntax for installation with rpm (redhat package manager) is:**

**# http://www.erikjacobs.com/stuff/sqlbuddy-1.3.3-1.noarch.rpm**



2. To List all files of an installed RPM package: To view all the files of an installed rpm packages, use the -ql (query list) with rpm command.

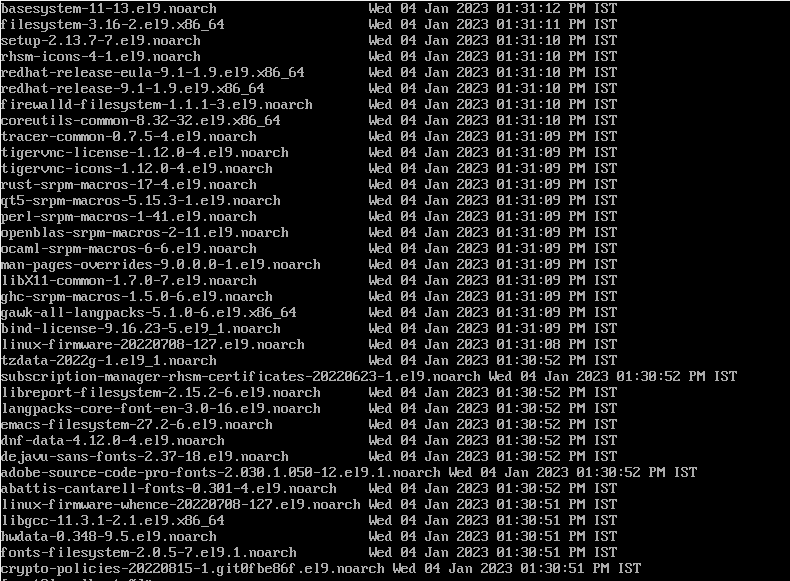
# Rpm -ql sqlbuddy



3. To List Recently Installed RPM Packages: Use the following rpm command with -qa (query all) option, will list

all the recently installed rpm packages.

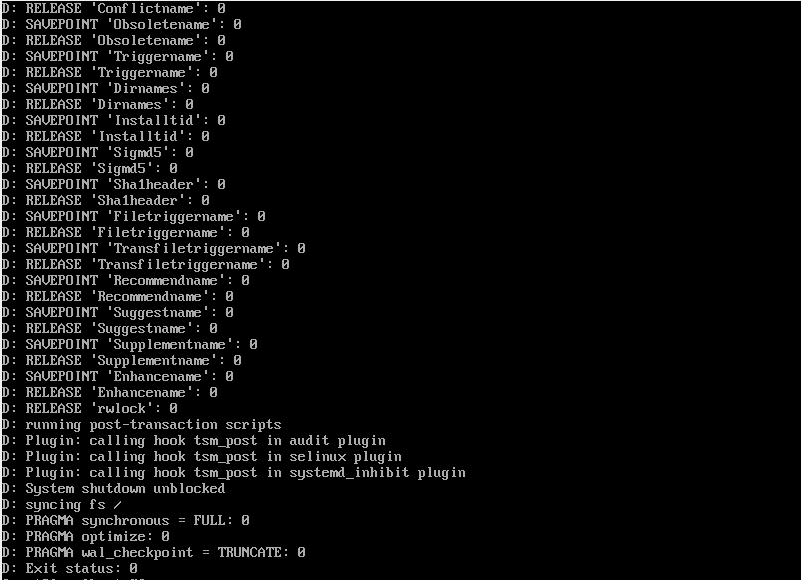
# Rpm -qa --last



4. To remove a RPM Package: To un-install an RPM package, for example we use the package name nx, not the original

package name sqlbuddy-1.3.3-1.noarch.rpm. The -e (erase) option is used to remove package.

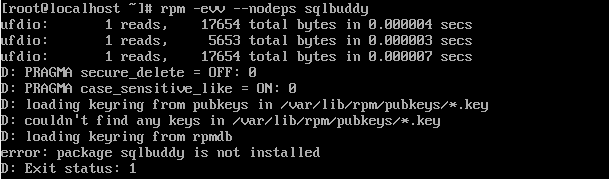
# Rpm -evv sqlbuddy



5. To remove an RPM Package Without Dependencies: The -nodeps (Do not check dependencies) option forcefully

remove the rpm package from the system. But keep in mind removing particular package may break other working applications.

# Rpm -evv –nodeps sqlbuddy



6. To query an Information of Installed RPM Package: Let's say you have installed an rpm package and want to know

the information about the package. The following -qi (query info) option will print the available information of the installed package.

# Rpm -qi sqlbuddy-1.3.3-1.noarch.rpm



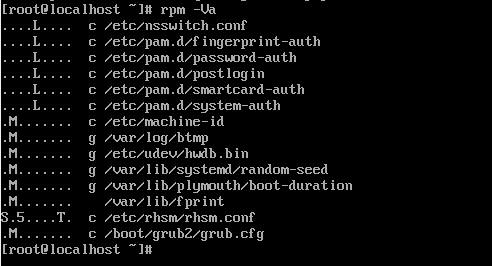
7. To Verify a RPM Package: Verifying a package compares information of installed files of the package against the rpm.database. The -Vp (verify package) is used to verify a package.

# rpm -Vp sqlbuddy-1.3.3-1.noarch.rpm

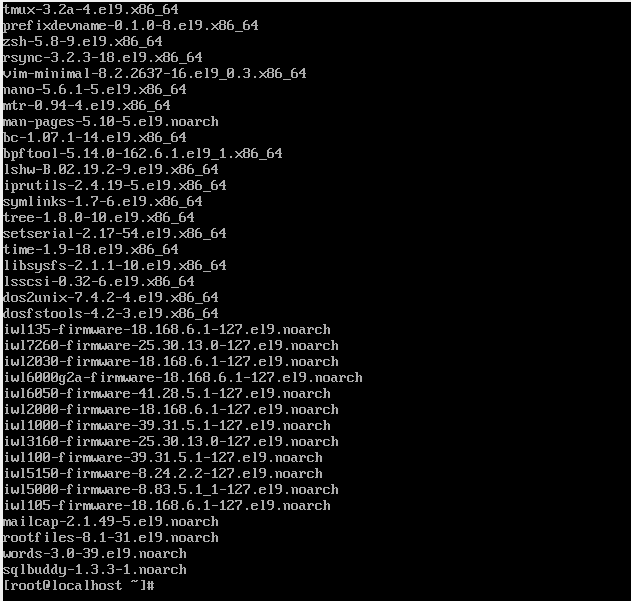


8. To Verify all RPM Packages: Type the following command to verify all the installed rpm packages.

# rpm -Va



Rpm –qa



**Practical 3c: Extracting files from RPM’S**

Every RPM package consists of two parts: the metadata part that describes what is in the package and a cpio archive that contains the actual files in the package.

To extract files from an RPM package you must first extract a cpio archive from the package itself. RedHat provides a utility called rpm2cpio which does exactly that:

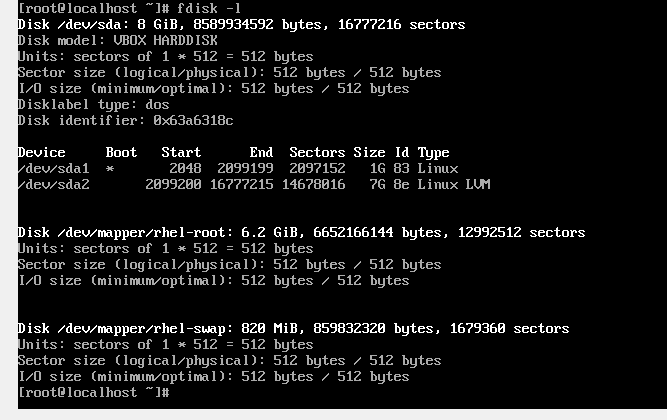
# rpm2cpio sqlbuddy-1.3.3-1.noarch.rpm | cpio -idmv



The cpio command copies files to and from archives. In the example above, we use cpio with the -i flag to extract the files from the archive, -d to create the leading directories where needed, and -m to preserve the file modification times when creating files. The -v flag (verbose) is to list the files processed for the sake of this example.

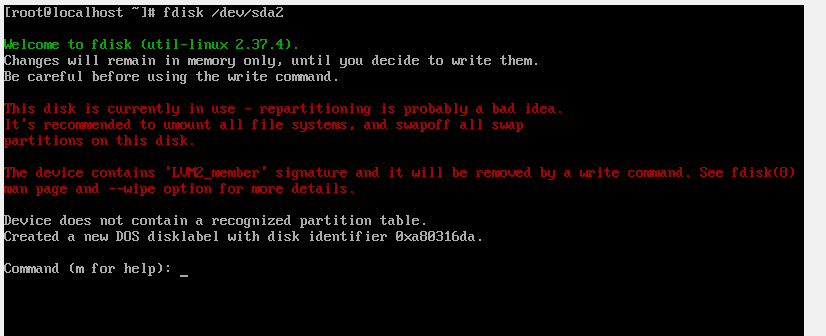
Practica 3c: configuring and managing storage

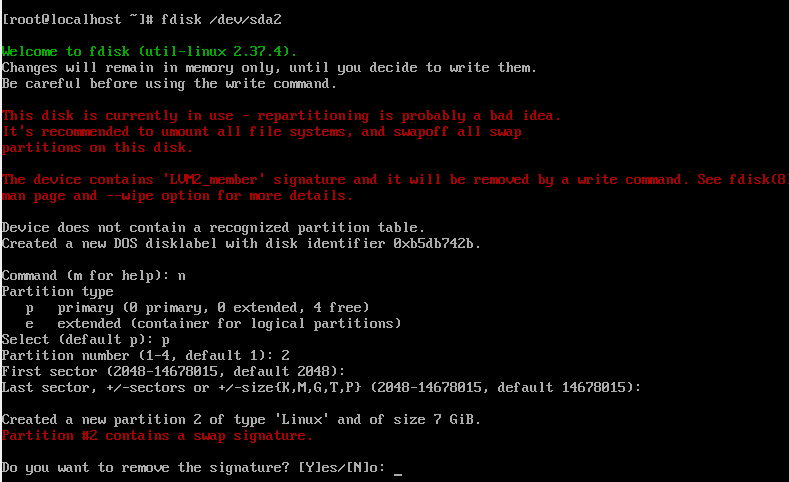
# fdisk –l: This command is used to list all hard drives with partition table.

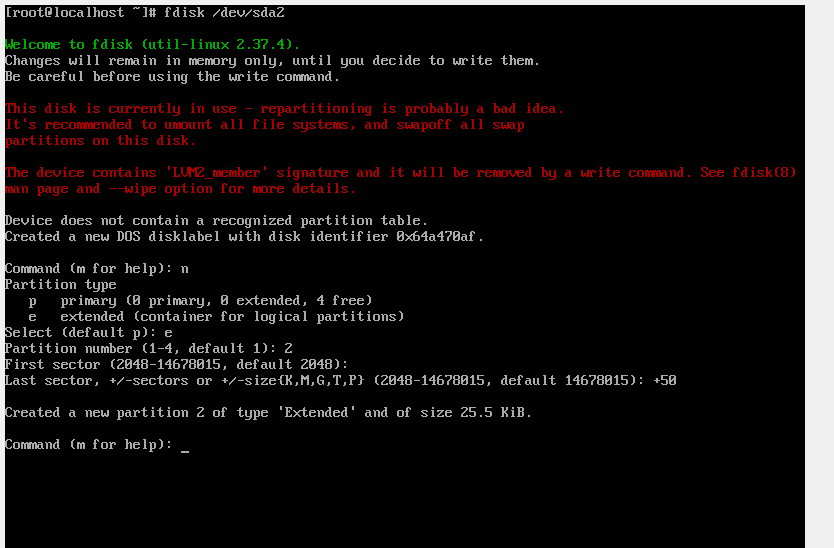
There is a single hard disk which we have partitioned during installation.

Use fdisk /dev/sdb2 to partition second hard disk.

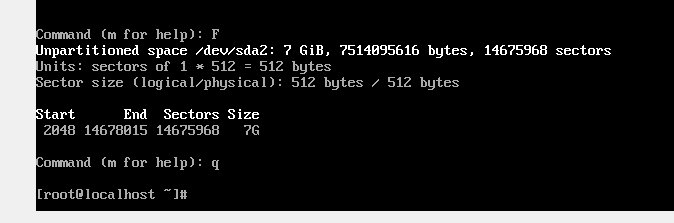
# fdisk /dev/sda2







Command F: To display the partition of disk.



**Practical 4a: Working with Users, Groups and Permissions**

**Ownership of Linux files**

Every file and directory on your Unix/Linux system is assigned 3 types of owner, given below.

**1. User:** A user is the owner of the file.

**2. Group:** A user-group can contain multiple users.

**3. Other:** Any other user who has access to a file. This person has neither created the file, nor he belongs to a usergroup who could own the file.

**Permissions**

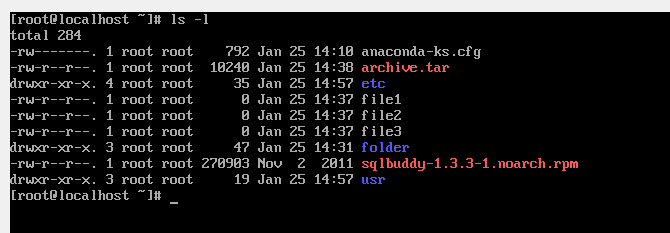
Every file and directory in your UNIX/Linux system has following 3 permissions defined for all the 3 owners discussed above.

**1. Read:** This permission give you the authority to open and read a file. Read permission on a directory gives you the ability to lists its content.

**2. Write**: The write permission gives you the authority to modify the contents of a file.  
**3. Execute:** In Windows, an executable program usually has an extension "exe" and which you can easily run.

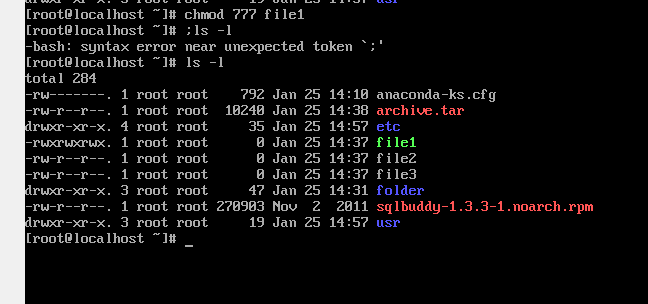
**1. Ls –l:** List all the hidden files present in your system.

# ls -l

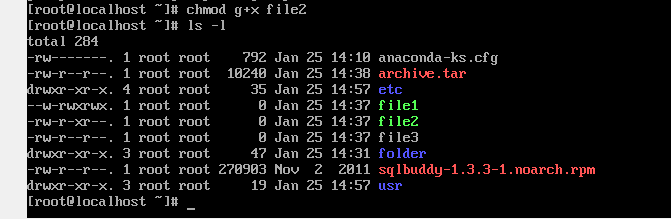


**2. Chmod:** Used to change the permissions for files.

**# chmod [rwx] [file\_name]**

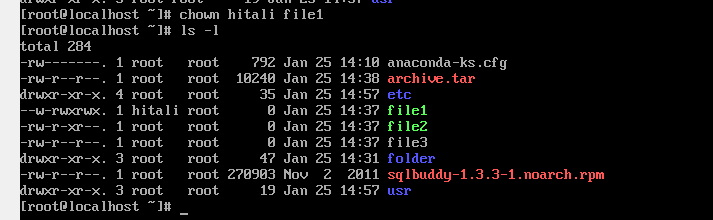


**# chmod [user\_or\_group]+[rwx] [file\_name]**



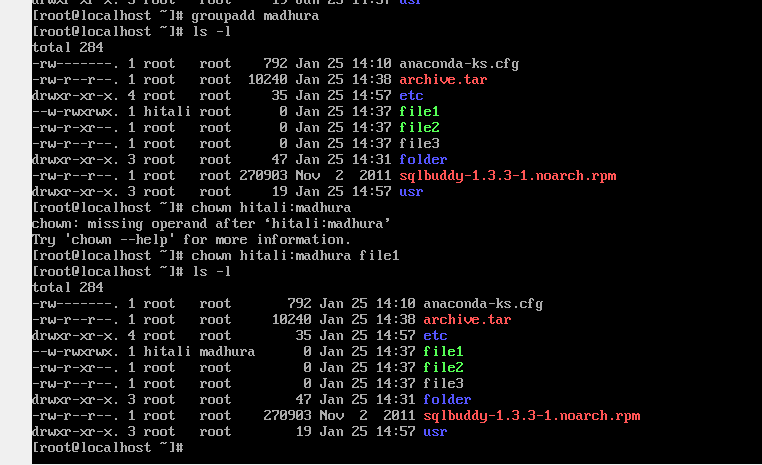
**3. Chown:** Used to change the owner of the file.

**# chmod [groupname\_or\_username] [file\_name]**



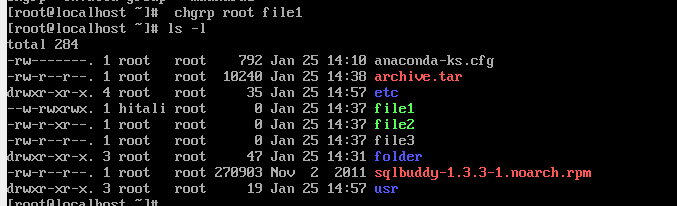
**4. Groupadd:** Used to add new group.

**# groupadd [groupname]**



**5. Chgrp:** Used to change the group owner.

**# chgrp [groupname] [filename]**



**Practical 5: Firewall and cryptographic services**

A Firewall is a network security device that monitors and filters incoming and outgoing network traffic based on an organization’s previously established security policies.

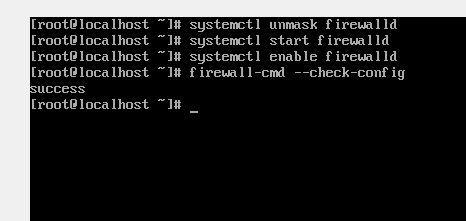
1. **Securing server with iptables**

A Firewall is a network security device that monitors and filters incoming and outgoing network traffic based on an organization's previously established security policies. At its most basic, a firewall is essentially the barrier that sits between a private internal network and the public Internet.

# Systemctl unmask firewalld

# Systemctl start firewalld

# Systemctl enable firewalld



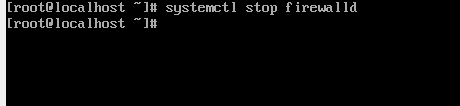
# /sbin/iptables -L



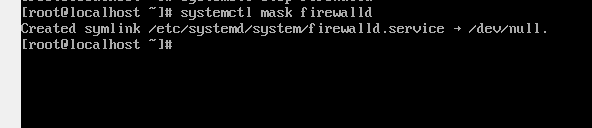
# Systemctl disable firewalld



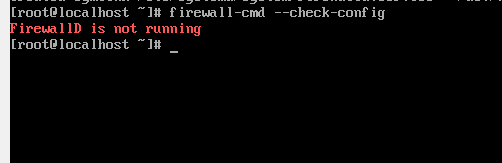
# Systemctl stop firewalld



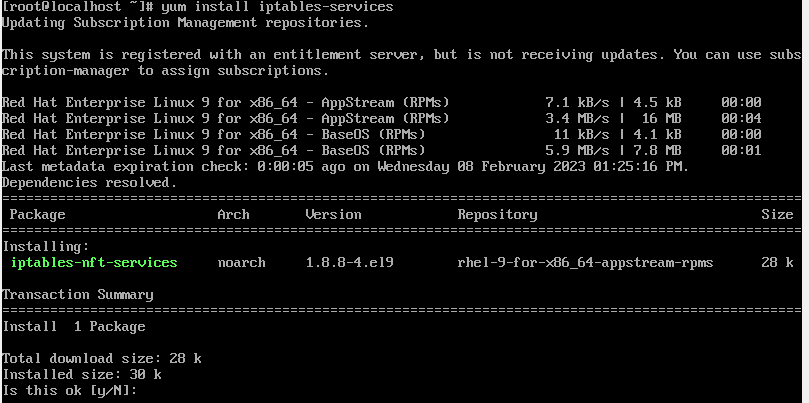
# Systemctl mask firewalld

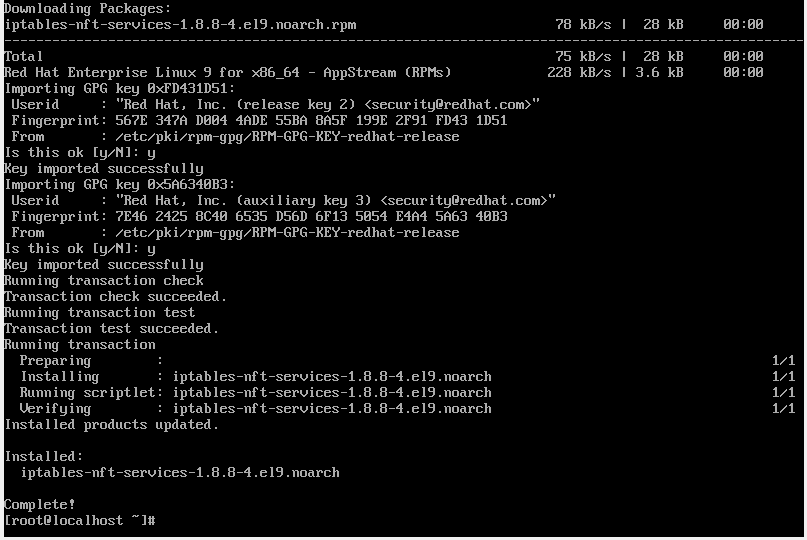


# Firewall-cmd –check-config



# yum install iptables-services



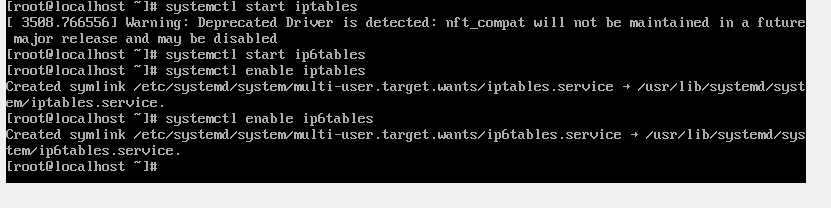


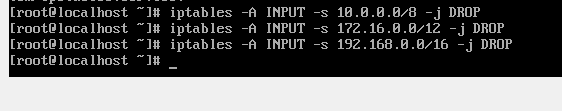
#systemctl start iptables

#systemctl start ip6tables

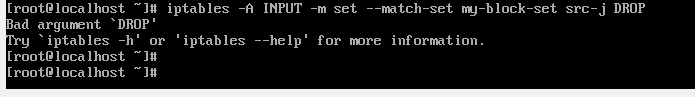
#systemctl enable iptables

#systemctl enable ip6tables



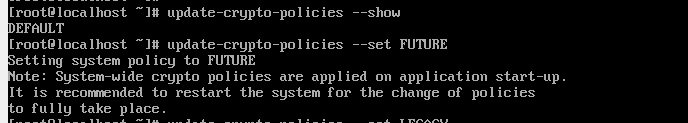




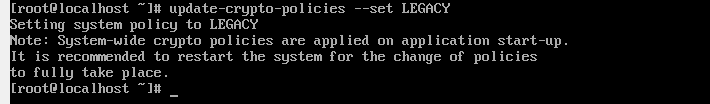


1. **Configuring and managing cryptographic services**

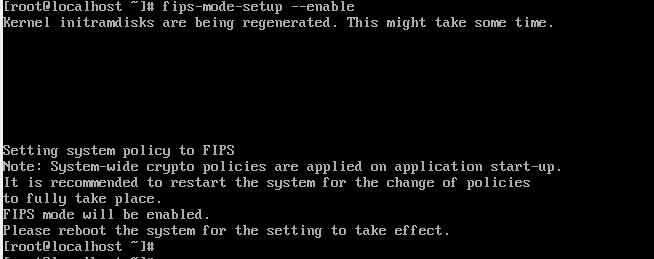
# update-crypto-policies --show

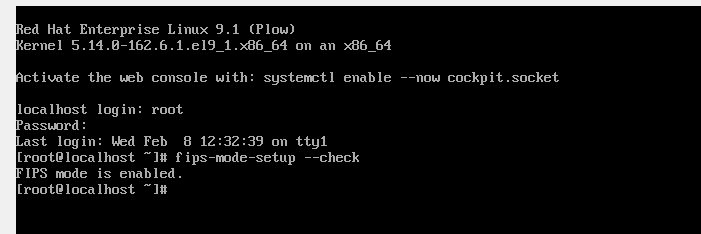


# update-crypto-policies --LEGACY



# update-crypto-policies --enable



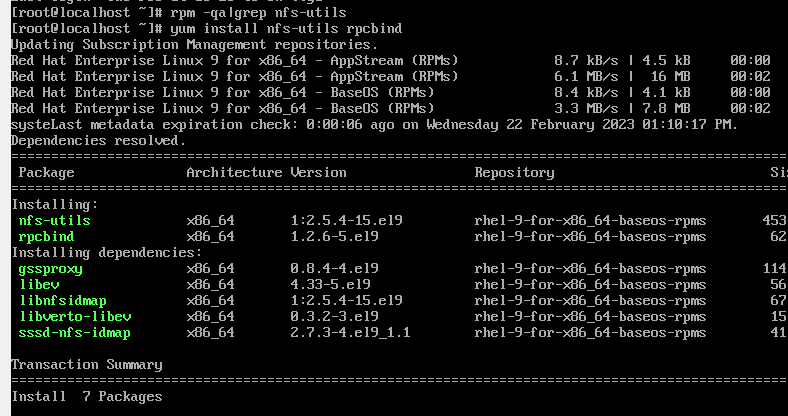


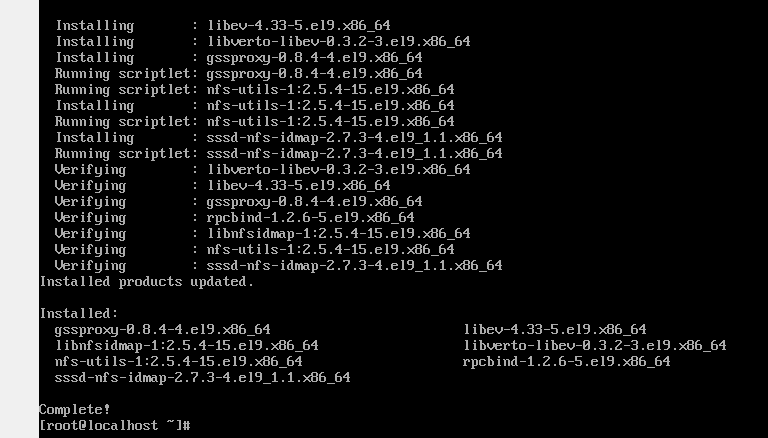
**Practical 6 : configuring servers for file sharing**

**Configuring server side**

NFS enables system administrators to share all or a portion of a file system on a networked server to make it accessible to remote computer users.

# rpm –qa | grep nfs-utils



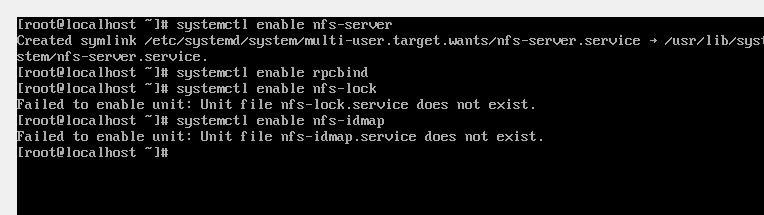


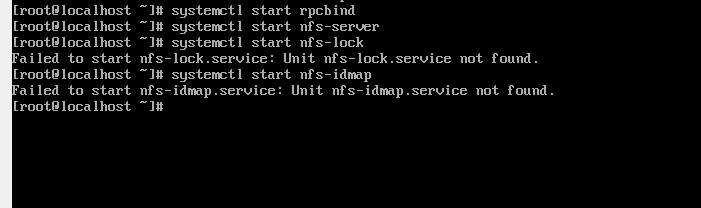
# systemctl enable nfs-server

# systemctl enable rpcbind

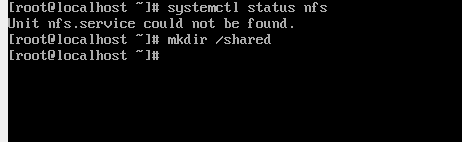
# systemctl enable nfs-lock

# systemctl enable nfs-idmap





# systemctl status nfs



# vi /etc/exports

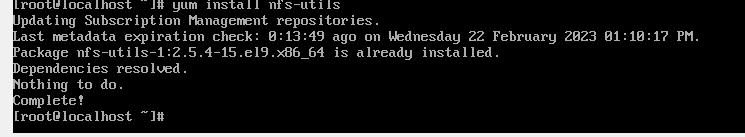
The /etc/exports file controls which file systems are exported to remote hosts and specifies options. Blank lines are ignored, comments can be made by starting





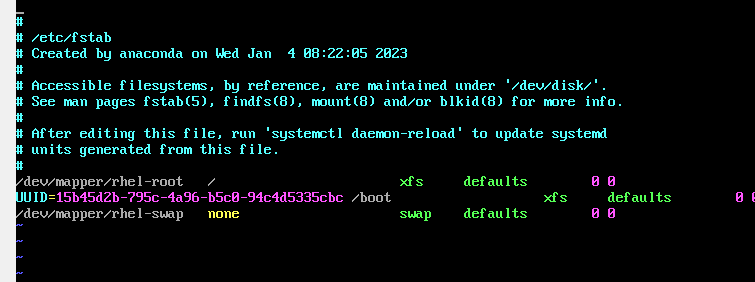
**Clients side configuring**

# yum install nfs-utils



# /etc/fstab

fstab , is a configuration table designed to ease the burden of mounting and unmounting file systems to a machine. It is a set of rules used to control how different filesystems are treated each time they are introduced to a system.



# firewall-cmd –add-service=nfs --zone=internal –permanent

# firewall-cmd –add-service=mountd --zone=internal –permanent

# firewall-cmd –add-service=rpc-bind --zone=internal –permanent

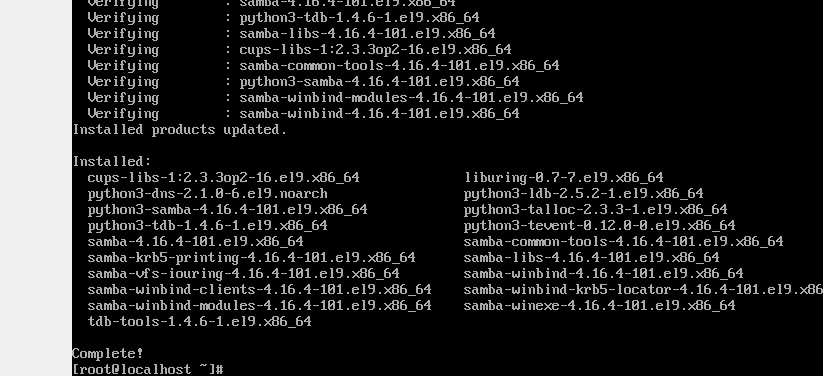


**b. Configuring samba**

Samba is the standard Windows interoperability suite of programs for Linux and Unix. Samba is Free Software licensed under the GNU General Public License.

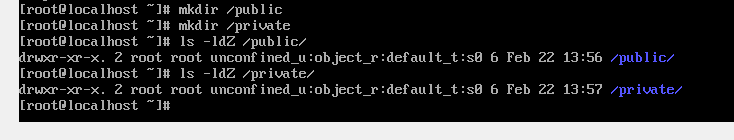
Samba

installation



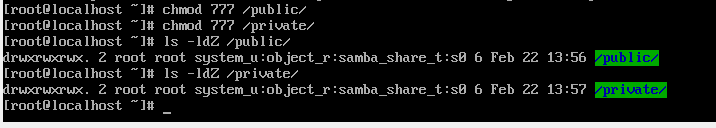
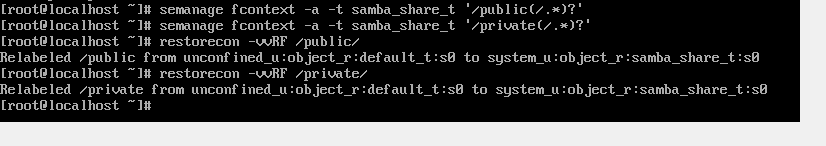
# mkdir /public

# mkdir /private



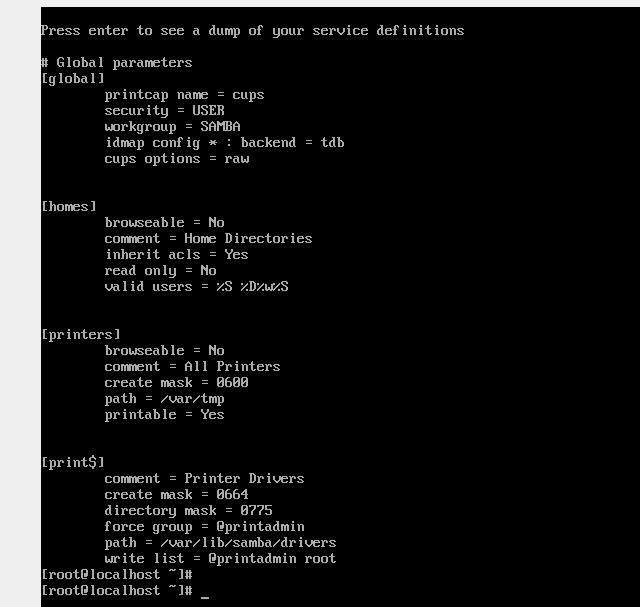
# semanage fcontext –a –t samba\_share\_t ’/public(/.\*)?’

# semanage fcontext –a –t samba\_share\_t ’/private(/.\*)?’



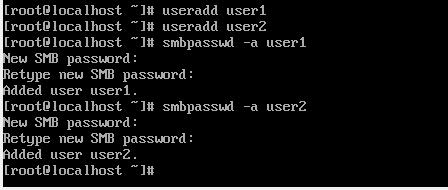


Check whether the connection is done with the Testparm

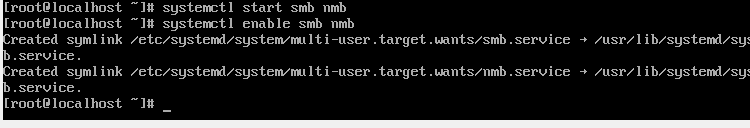


User1-passwd=12345

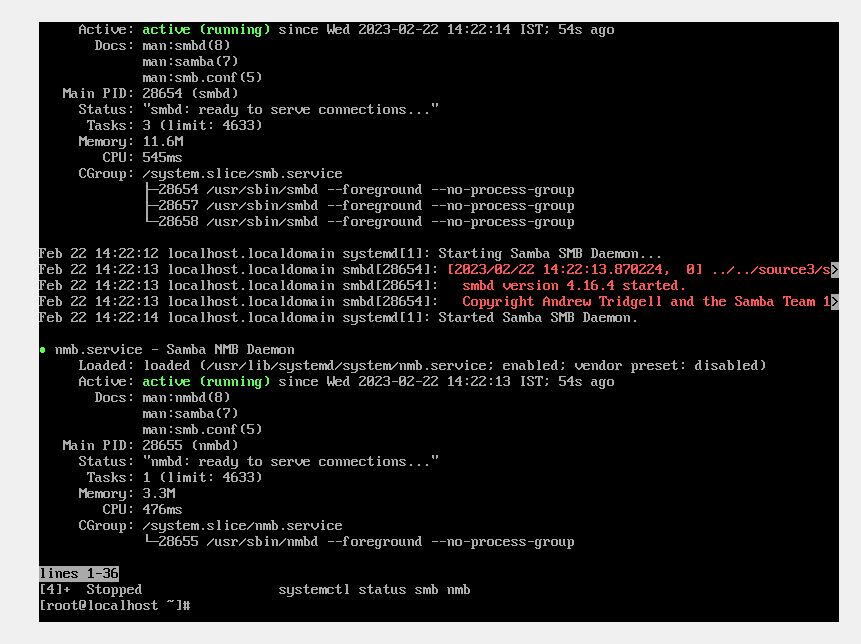
User2-passwd=6789



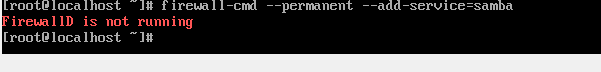
# systemctl start smb nmb

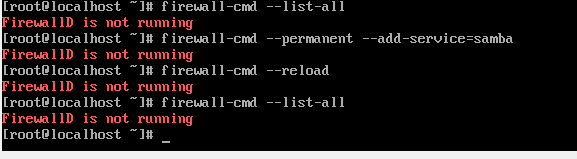


System status

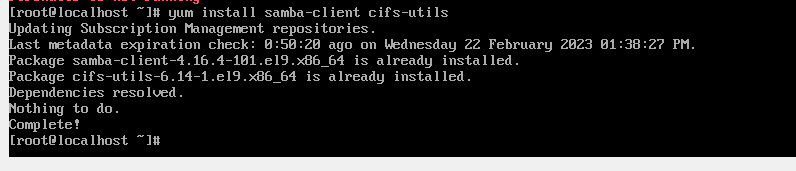


# firewall-cmd –permenent –add-service=samba



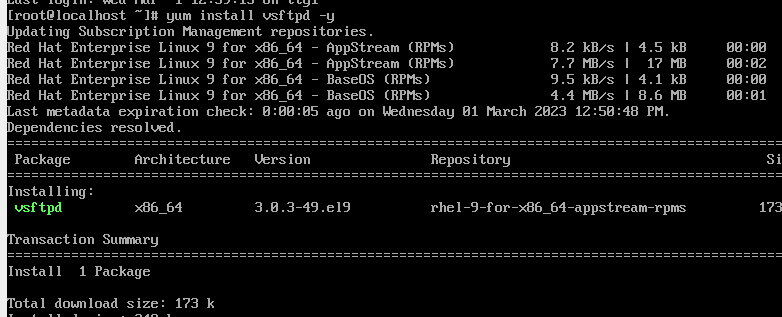


# yum install samba-client cifs-utils



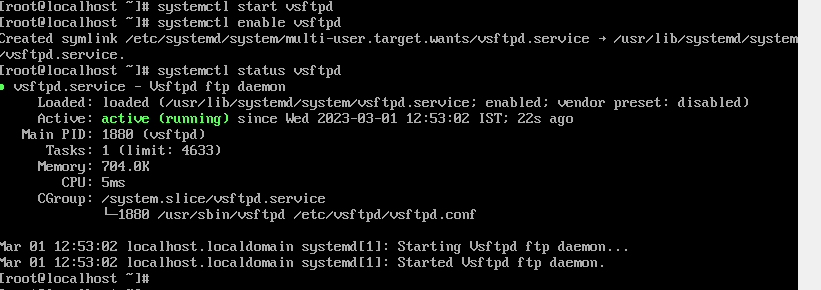
1. **Configuring FTP**

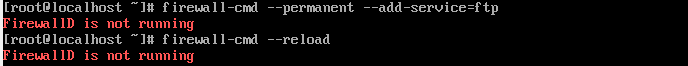
# yum install vsftpd -y



# systemctl start vsftpd

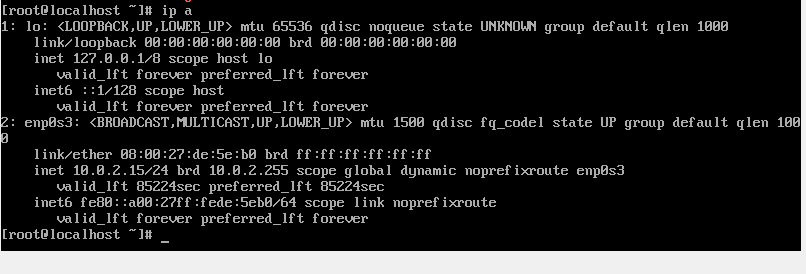
# systemctl enable vsftpd



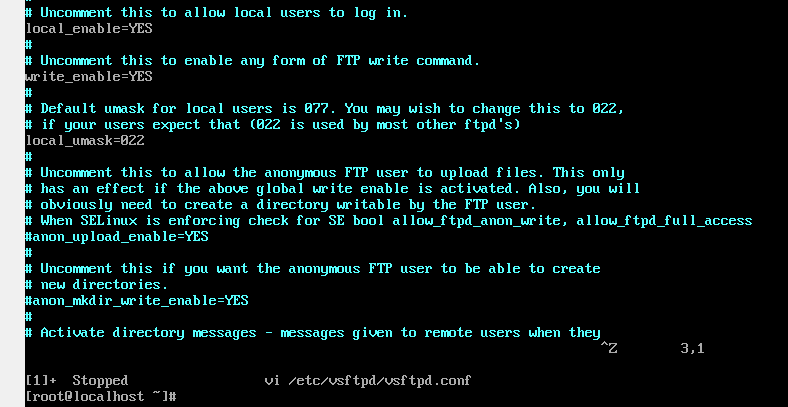




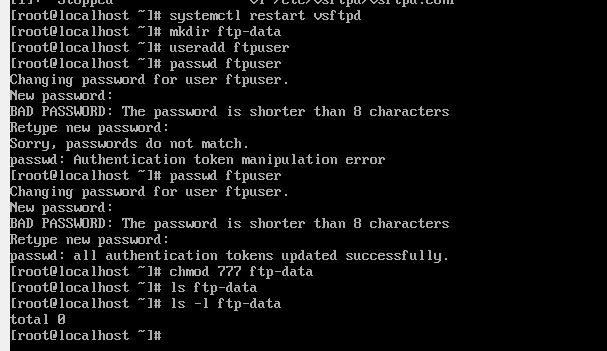
# ip a

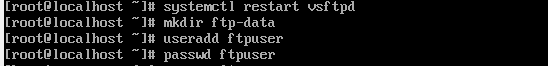


# vi /etc/vsftpd/vsftpd.conf

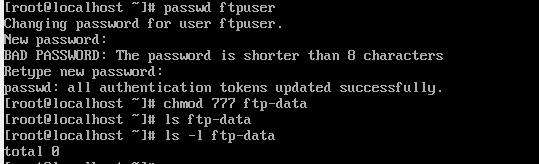


#systemctl restart vsftpd

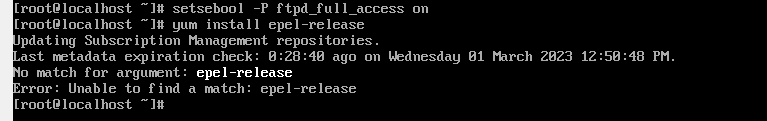


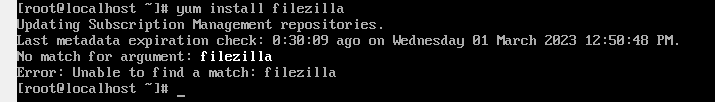


New password- 12345

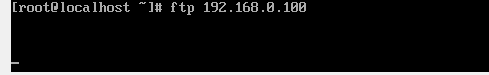
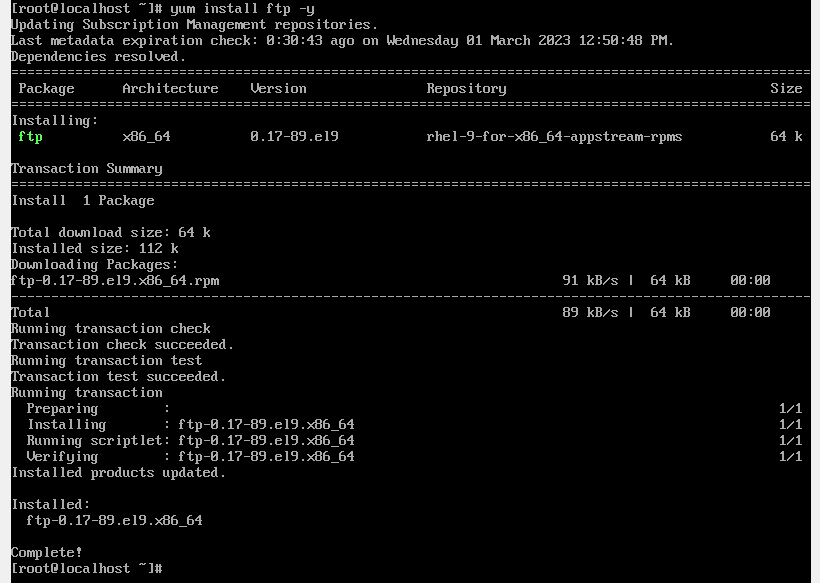


# setsebool –P ftpd\_full\_access on

# yum install epel-release  


# yum install filezilla

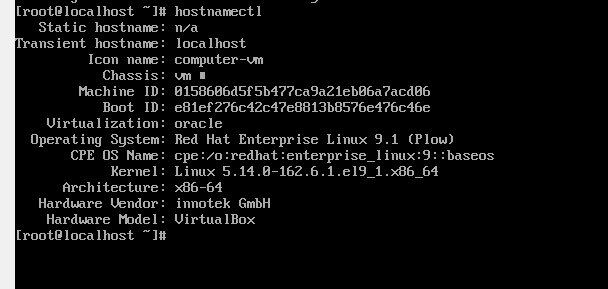
# yum install ftp -y



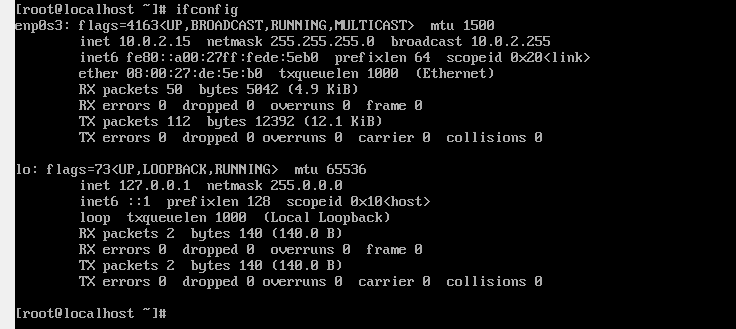
**Practical 7: Configuring DNS, DHCP and Mail Server**

1. **Configuring DNS**

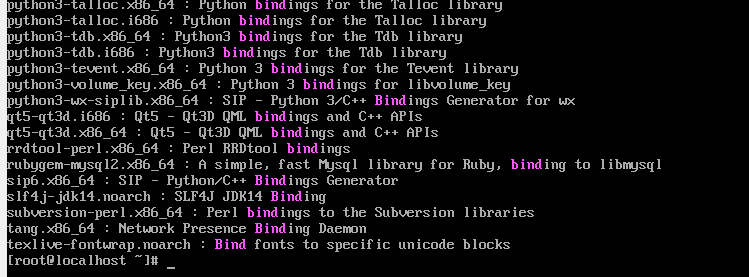
**# hostnamectl**



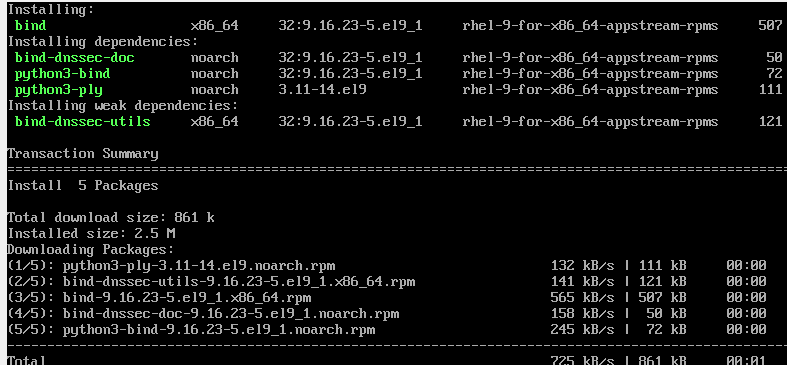
# ifconfig

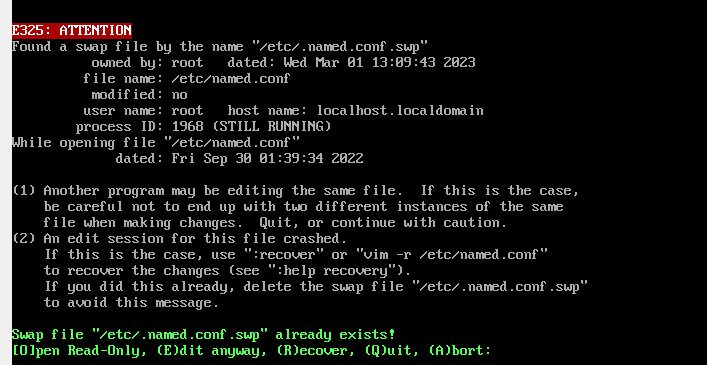


# yum search bind



# yum -y install bind bind-utils

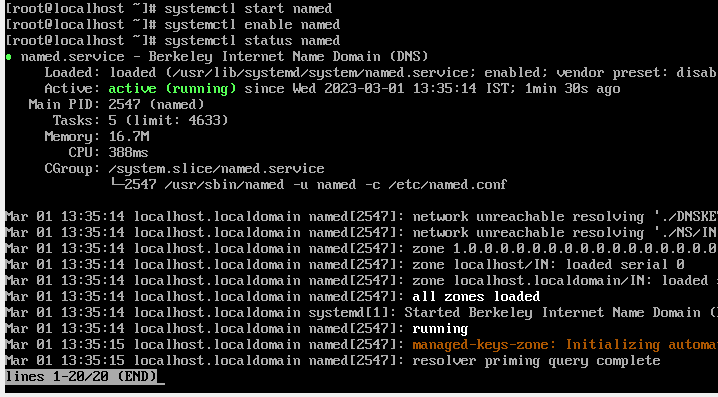
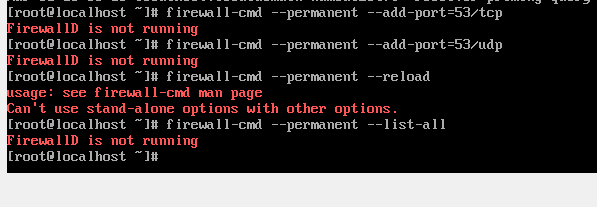


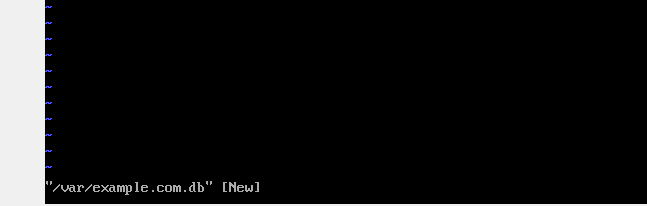
# vi /etc/named.conf

# systemctl start named

# systemctlenable named

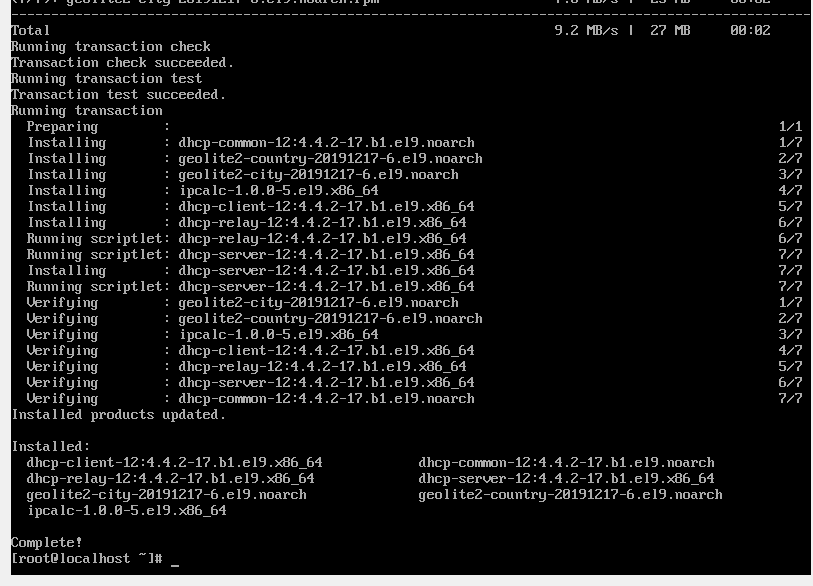
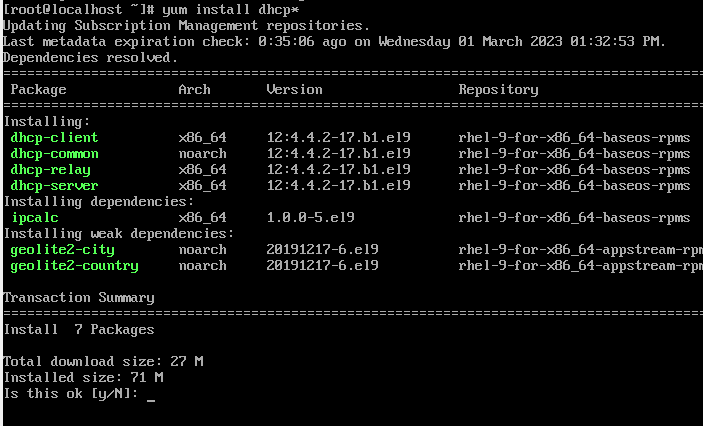
# systemctl staatus named

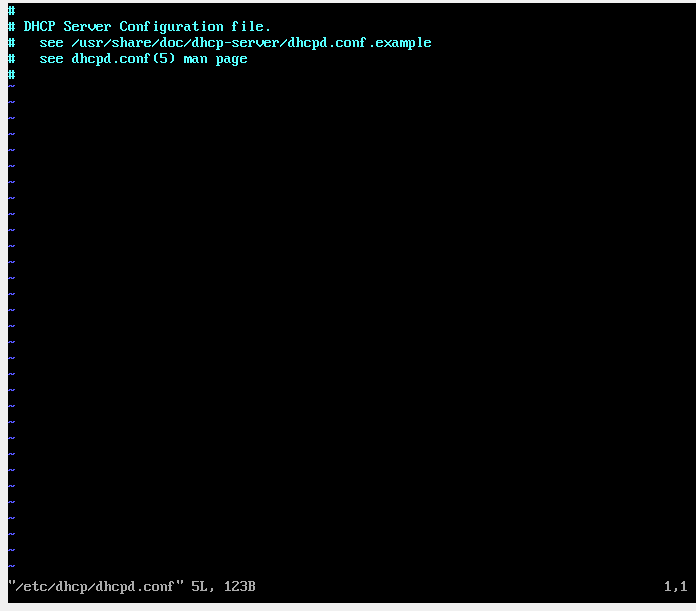
# /var/example.com.db 

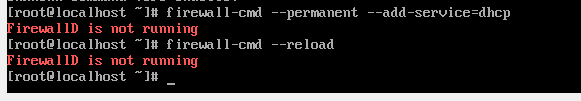
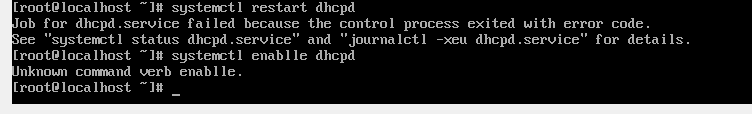
1. **Configuring DHCP**

# yum install dhcp\*

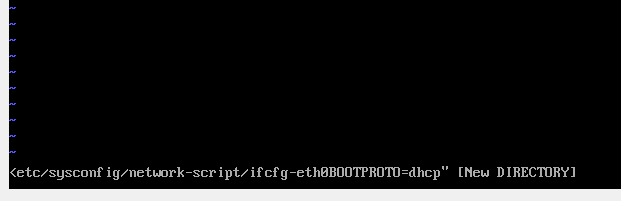


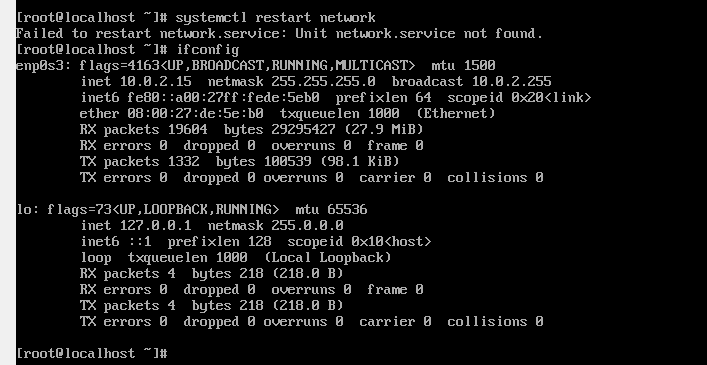
# /etc/dhcp/dhcpd.conf

# systemctl restart dhcpd

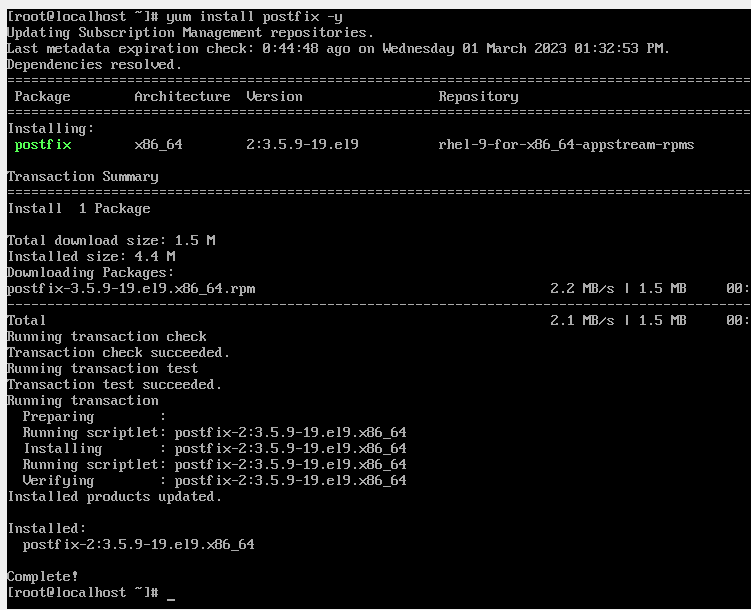


# vi /etc/sysconfig/network-script/ifcgf-eth0**BOOTPROTO=dhcp**

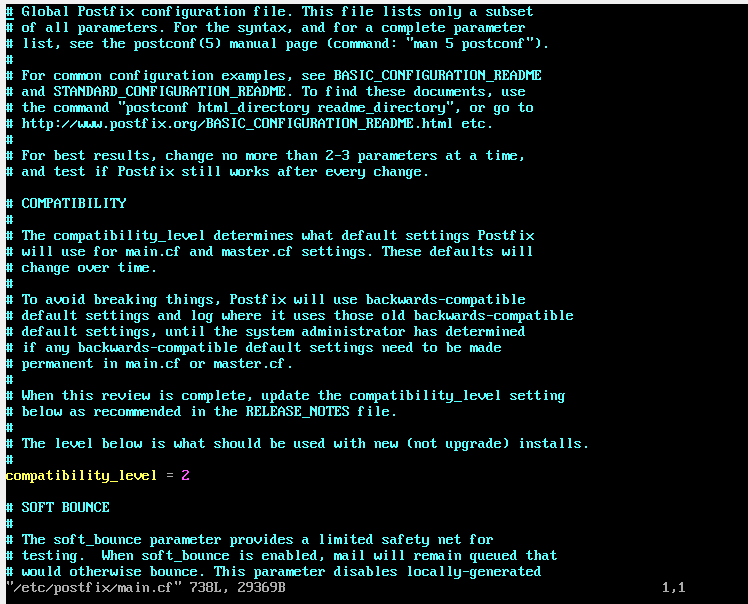
# systemctl restart network



**C. Configuring mail**

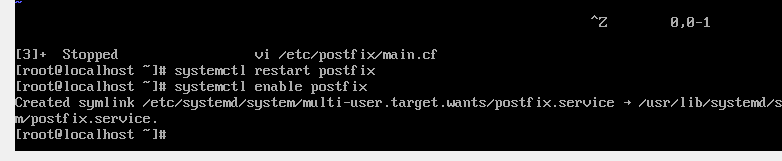


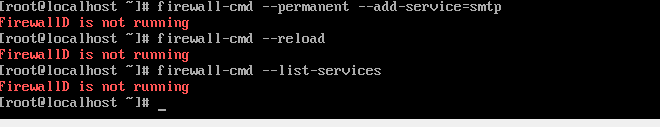
# /etc/postfix/main.cf



# systemctl restart postfix

#systemctl enable postfix





#whoami



#yum -y inatall mailx

