Name: Ruud Van G. Apostol	**Date Performed:** 09/95/25
Course/Section: CPE212 / CPE31S4	**Date Submitted:** 09/05/25
Instructor: Engr. Robin Valenzuela	**Semester and SY:** 1st Sem / 2025-2026

Activity 3: Install SSH server on CentOS or RHEL 8

1. **Objectives:**

1.1 Install Community Enterprise OS or Red Hat Linux OS

1.2 Configure remote SSH connection from remote computer to CentOS/RHEL-8

2. **Discussion:**

CentOS vs. Debian: Overview

CentOS and Debian are Linux distributions that spawn from opposite ends of the candle.

CentOS is a free downstream rebuild of the commercial Red Hat Enterprise Linux distribution where, in contrast, Debian is the free upstream distribution that is the base for other distributions, including the Ubuntu Linux distribution.

As with many Linux distributions, CentOS and Debian are generally more alike than different; it isn't until we dig a little deeper that we find where they branch.

CentOS vs. Debian: Architecture

The available supported architectures can be the determining factor as to whether a distro is a viable option or not. Debian and CentOS are both very popular for x86_64/AMD64, but what other archs are supported by each?

Both Debian and CentOS support AArch64/ARM64, armhf/armhfp , i386 , ppc64el/ppc64le. (Note: armhf/armhfp and i386 are supported in CentOS 7 only.)

CentOS 7 additionally supports POWER9 while Debian and CentOS 8 do not. CentOS 7 focuses on the x86_64/AMD64 architecture with the other archs released through the AltArch SIG (Alternate Architecture Special Interest Group) with CentOS 8 supporting x86_64/AMD64, AArch64 and ppc64le equally.

Debian supports MIPSel, MIPS64el and s390x while CentOS does not. Much like CentOS 8, Debian does not favor one arch over another —all supported architectures are supported equally.

CentOS vs. Debian: Package Management

Most Linux distributions have some form of package manager nowadays, with some more complex and feature-rich than others.

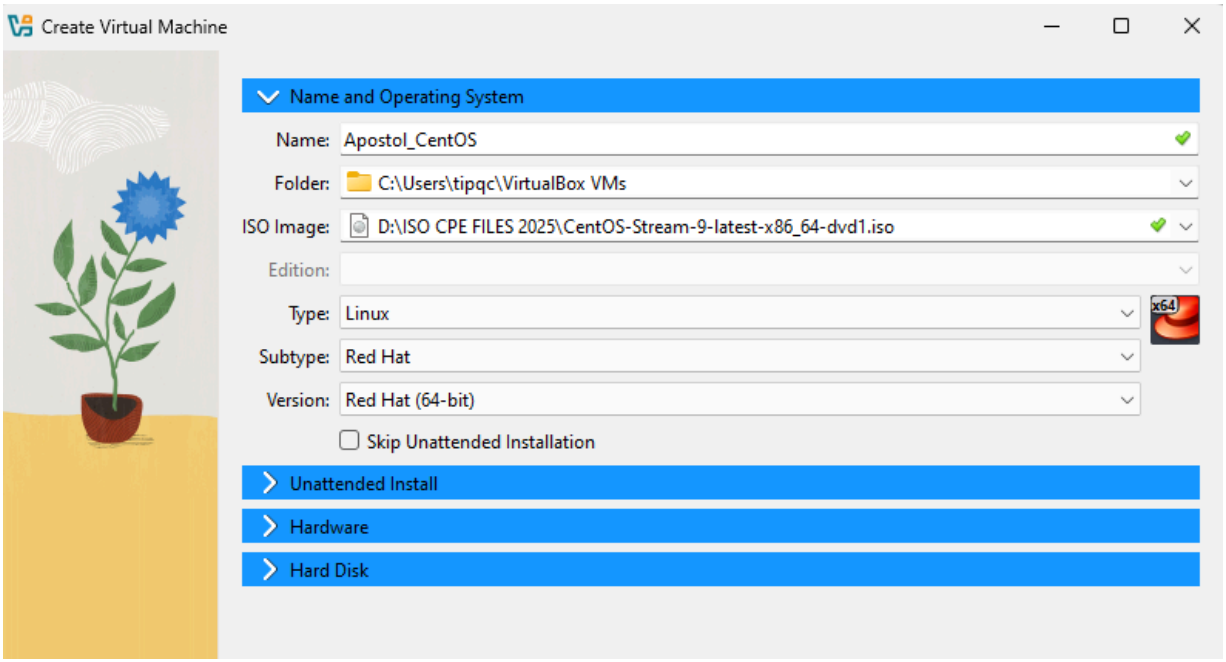
CentOS uses the RPM package format and YUM/DNF as the package manager.

Debian uses the DEB package format and dpkg/APT as the package manager.

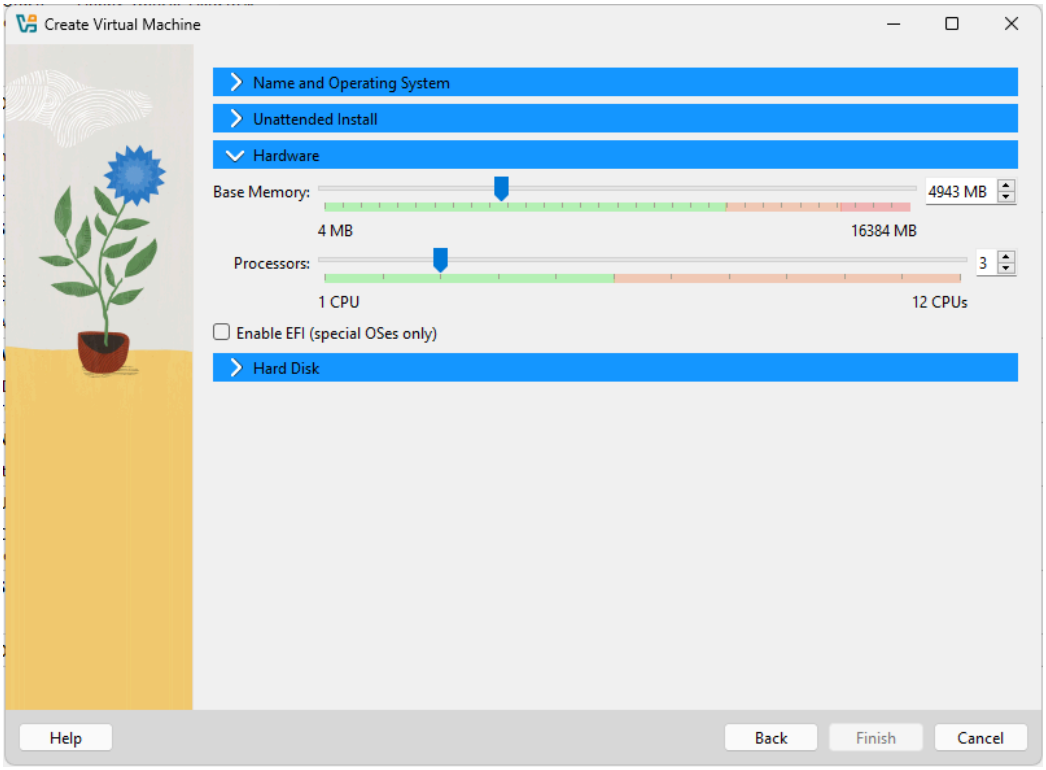
Both offer full-feature package management with network-based repository support, dependency checking and resolution, etc.. If you're familiar with one but not the other, you may have a little trouble switching over, but they're not overwhelmingly different. They both have similar features, just available through a different interface.

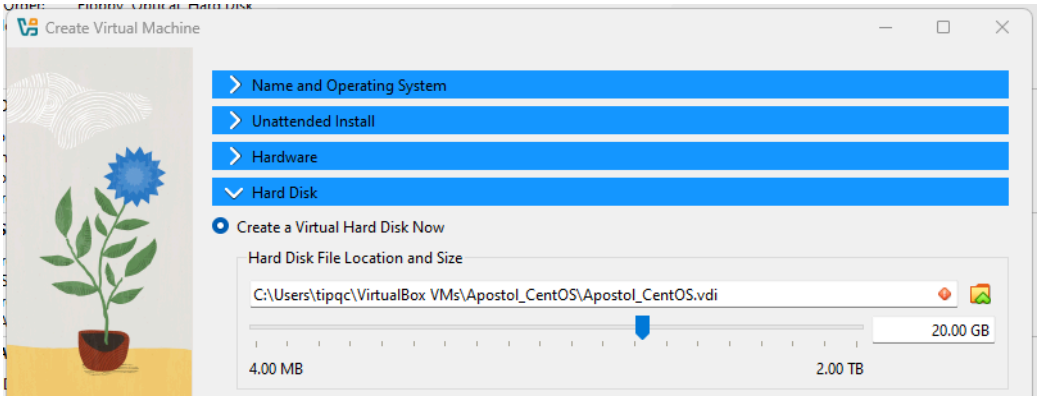
Task 1: Download the CentOS or RHEL-8 image (Create screenshots of the following)

1. Download the image of the CentOS here: http://mirror.rise.ph/centos/7.9.2009/isos/x86_64/

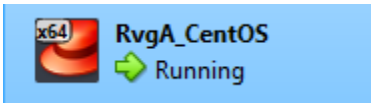


2. Create a VM machine with 2 Gb RAM and 20 Gb HD.





3. Install the downloaded image.
4. Show evidence that the OS was installed already.



Task 2: Install the SSH server package *openssh*

1. Install the ssh server package *openssh* by using the *dnf* command:
   $ dnf install openssh-server

```
[root@vbox Apostol]# dnf install openssh-server
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use "rhc" or "
subscription-manager" to register.

CentOS Stream 9 - BaseOS                        1.2 MB/s | 8.8 MB     00:07
CentOS Stream 9 - Ap 37% [=======-            ] 328 kB/s | 9.3 MB     00:48 ETA
```

2. Start the *sshd* daemon and set to start after reboot:
   $ systemctl start sshd
   $ systemctl enable sshd

```
[root@vbox Apostol]# systemctl start sshd
[root@vbox Apostol]# systemctl enable sshd
[root@vbox Apostol]#
```

3. Confirm that the sshd daemon is up and running:
   $ systemctl status sshd

```
[root@vbox Apostol]# systemctl status sshd
● sshd.service - OpenSSH server daemon
     Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: ena>
     Active: active (running) since Fri 2025-09-05 14:19:14 PST; 1min 11s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 6921 (sshd)
      Tasks: 1 (limit: 20279)
     Memory: 1.4M
        CPU: 10ms
     CGroup: /system.slice/sshd.service
             └─6921 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Sep 05 14:19:14 vbox systemd[1]: Starting OpenSSH server daemon...
Sep 05 14:19:14 vbox sshd[6921]: Server listening on 0.0.0.0 port 22.
Sep 05 14:19:14 vbox sshd[6921]: Server listening on :: port 22.
Sep 05 14:19:14 vbox systemd[1]: Started OpenSSH server daemon.
lines 1-16/16 (END)
[2]+  Stopped                 systemctl status sshd
[root@vbox Apostol]#
```

4. Open the SSH port 22 to allow incoming traffic:
   $ firewall-cmd --zone=public --permanent --add-service=ssh
   $ firewall-cmd --reload

```
[root@vbox Apostol]# firewall-cmd --zone=public --permanent --add-service=ssh
Warning: ALREADY_ENABLED: ssh
success
[root@vbox Apostol]# firewall-cmd --reload
success
[root@vbox Apostol]#
```

5. Locate the ssh server man config file */etc/ssh/sshd_config* and perform custom configuration. Every time you make any change to the */etc/ssh/sshd-config* configuration file reload the *sshd* service to apply changes:
   $ systemctl reload sshd

```
[root@vbox Apostol]# systemctl reload sshd
[root@vbox Apostol]#
```

Task 3: Copy the Public Key to CentOS

1. Make sure that *ssh* is installed on the local machine.

```
[root@vbox Apostol]# ssh -v
usage: ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-B bind_interface]
           [-b bind_address] [-c cipher_spec] [-D [bind_address:]port]
           [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]
           [-i identity_file] [-J [user@]host[:port]] [-L address]
           [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
           [-Q query_option] [-R address] [-S ctl_path] [-W host:port]
           [-w local_tun[:remote_tun]] destination [command]
[root@vbox Apostol]#
```

2. Using the command *ssh-copy-id*, connect your local machine to CentOS.

```
Apostol@ApostolCN:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/Apostol/.ssh/id_rsa):
/home/Apostol/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/Apostol/.ssh/id_rsa
Your public key has been saved in /home/Apostol/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:I4MJfHQGNXVxSLY6PZtXN+U6BBU55XPKpnXXmbkjgmo Apostol@ApostolCN
The key's randomart image is:
+---[RSA 4096]----+
|    oo=...=o. o+.|
| . . o . o.o .o. |
|  o .     . .  o+|
|   o o   o   o oO|
|    o o S o   B=*|
|       o o.+ * ++|
|         .o.o.oo |
|       E.  .. ...|
|      ..         |
+----[SHA256]-----+
```

```
Apostol@ApostolCN:~$ ssh-copy-id -i ~/.ssh/id_rsa Apostol@192.168.56.113
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/Apostol/.ss
h/id_rsa.pub"
The authenticity of host '192.168.56.113 (192.168.56.113)' can't be established.
ED25519 key fingerprint is SHA256:LOdLpQhnO1uPEBnNbX/yjrcKPHgASoQbLoHbqx3iorY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
Apostol@192.168.56.113's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'Apostol@192.168.56.113'"
and check to make sure that only the key(s) you wanted were added.

Apostol@ApostolCN:~$
```

3. On CentOS, verify that you have the *authorized_keys*.

```
[Apostol@vbox ~]$ cat ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDR9/0fMwujyEIizkoxFmawUvt5F/JwOOOprLQfVo61
VYv4RvdDio4jBr/j5JtcUlbqkUEQLp74+LtD8KWum0r4ziSjKFhpYDqcTtEUwrPux5OpXsrxwN0Kd247
lTsZMpyUDu25ZBuTKqqQh+mpbY2vqf42Ox8uyBD5vu0kJSBpjyh0jV4DrD/fMpNX+Z/Iq1xAT50uCI6t
mUUKomZSj9Kuam3qm2kFvAL6G4Zw0kyaMLIp2sDT8oydwByo/7jZ5cafizB0OIxjMqxH1KA8ijZuUSob
8ppJCaQ/0rFYStKLZM51A7nlSZHs0mmVtA0WseN37VaMLMl7bP5hnJy+pI3vYgTrjOoMUSqTzrX4gW6W
0amB3e7yMhA2kOKV+4ApXAjaOS84nsB348qtclcd+go40ozxFDJ9K9F65OhzeNb49E65SM5q32733sud
/OHSISthFDbdX1SMpG/h1k9jwR88q1FLTY4AJy7YuqndWU1a/Ttsj0CAP0sqwKgfQ3H3zqoy/0j61I4P
OYz6iFKsBQH2rNIUMs8A39KMBdS3xNnS0IZn5HRr2CAp/nLFgVfwheXNopJ7Xfp0IWXWu5z4FMtnacet
t9MkpV9lwnjTvbj19M3ZscOyN3usqMPy0J1gXfOMnMCkBaufoSObmesw8HAGqQtVr6pZDo1XHVqAk/62
fw== Apostol@ApostolCN
[Apostol@vbox ~]$
```

Task 4: Verify ssh remote connection

1. Using your local machine, connect to CentOS using ssh.

```
Apostol@ApostolCN:~$ ssh Apostol@192.168.56.113
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Fri Sep  5 15:10:11 2025
[Apostol@vbox ~]$
```

2. Show evidence that you are connected.

```
[Apostol@vbox ~]$ hostname
vbox
[Apostol@vbox ~]$ whoami
Apostol
[Apostol@vbox ~]$
```

Reflections:

Answer the following:

1. What do you think we should look for in choosing the best distribution between Debian and Red Hat Linux distributions?

When choosing between Debian and Red Hat-based distributions, consider factors like cost, community versus commercial support, desired software stability versus newer features, your specific hardware, and the need for long-term enterprise support.

2. What are the main differences between Debian and Red Hat Linux distributions?

The main differences between Debian and Red Hat (specifically Red Hat Enterprise Linux, RHEL) are their package management systems (.deb vs. .rpm), package managers (APT vs. YUM/DNF), and corporate backing (non-profit vs. commercial). Debian is known for its community focus and use of the `.deb` package format with the Advanced Package Tool (APT), while RHEL is a commercial product focused on business use, using `.rpm` packages and the YUM/DNF package managers