Name: Mark Andrei Ponayo	**Date Performed:** Sep 6, 2023
Course/Section: BSCPE31S5	**Date Submitted:** Sep 6, 2023
Instructor: Engr. Roman Richard	**Semester and SY:** 3rd year 1st sem

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.

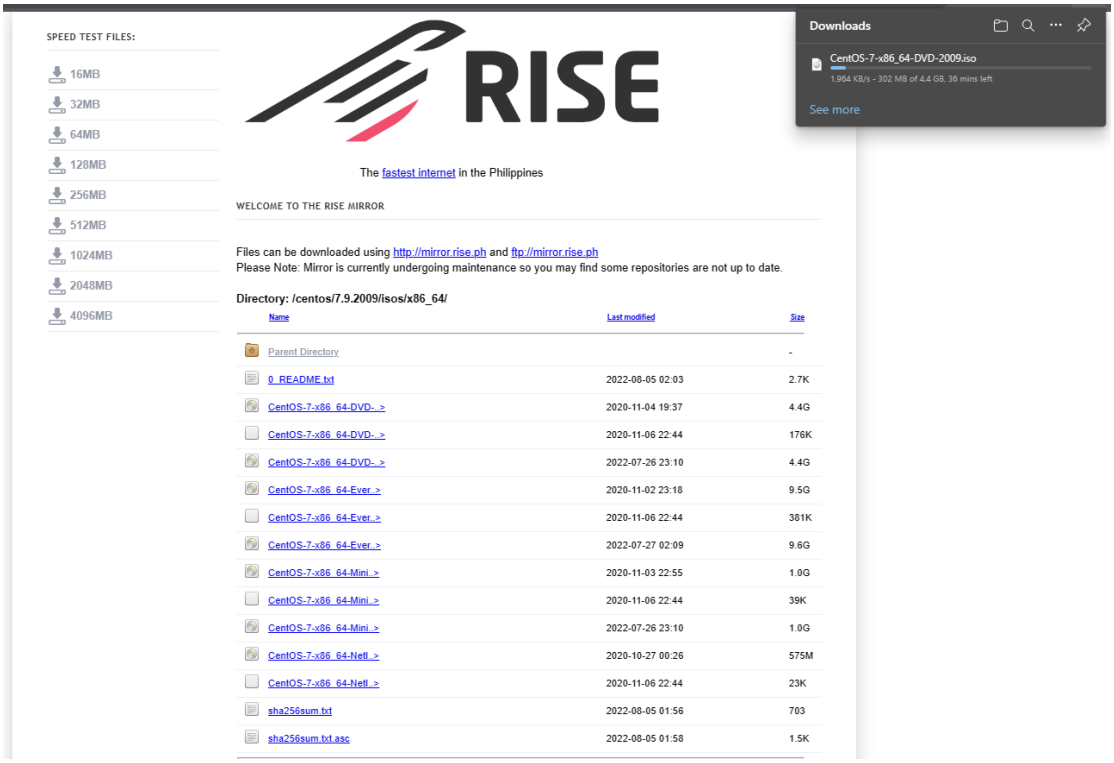
CentOS uses the RPM package format and YUM/DNF as the package manager.

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.

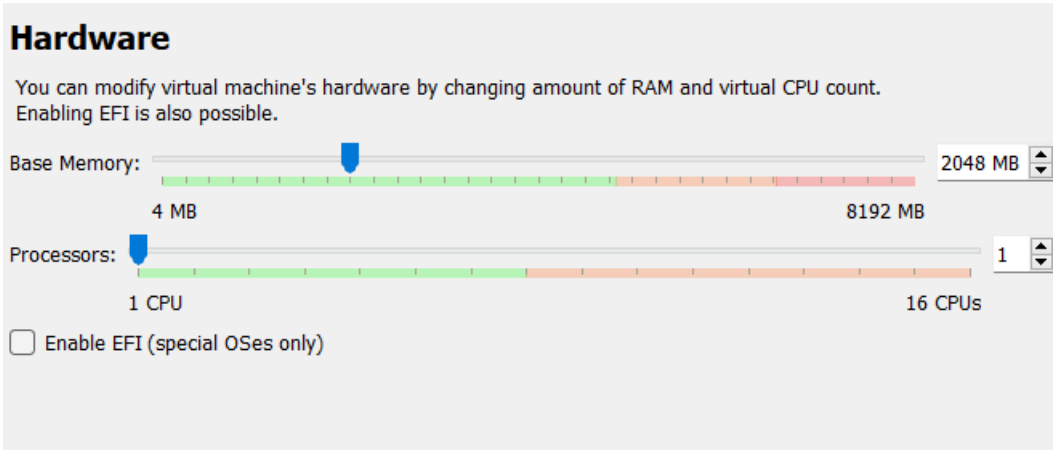
Task 1: Download the CentOS or RHEL-8 image (Create screenshots of the following)

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

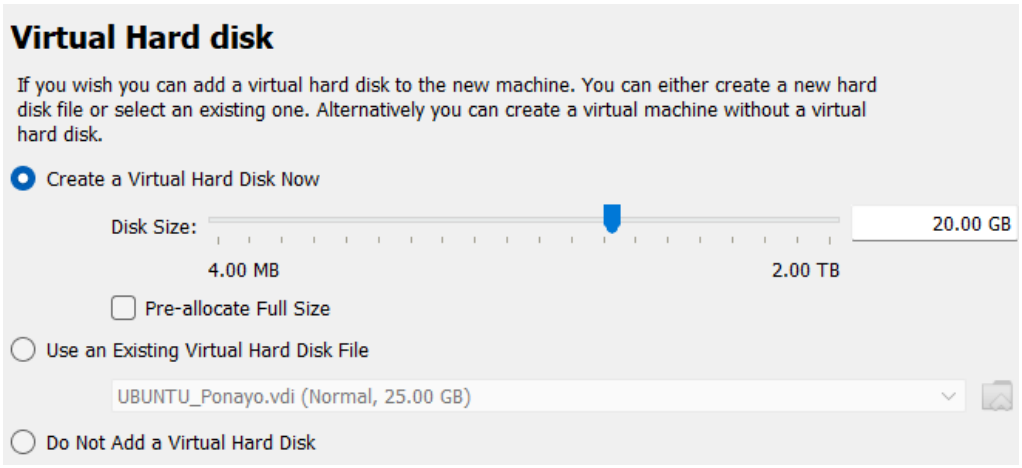


2. Create a VM machine with 2 Gb RAM and 20 Gb HD.

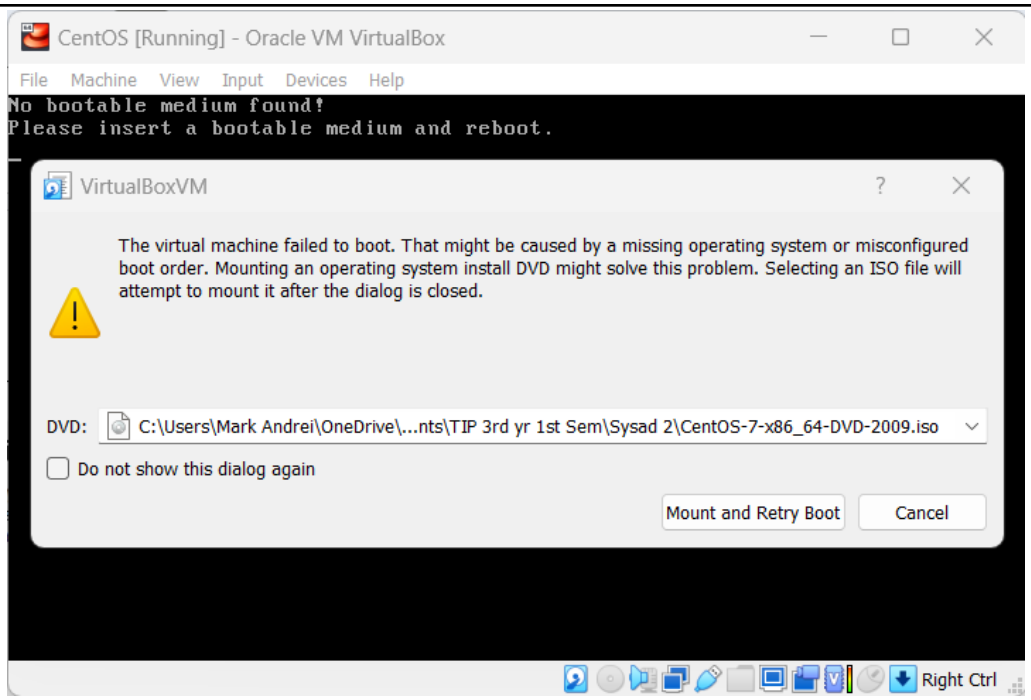
RAM:



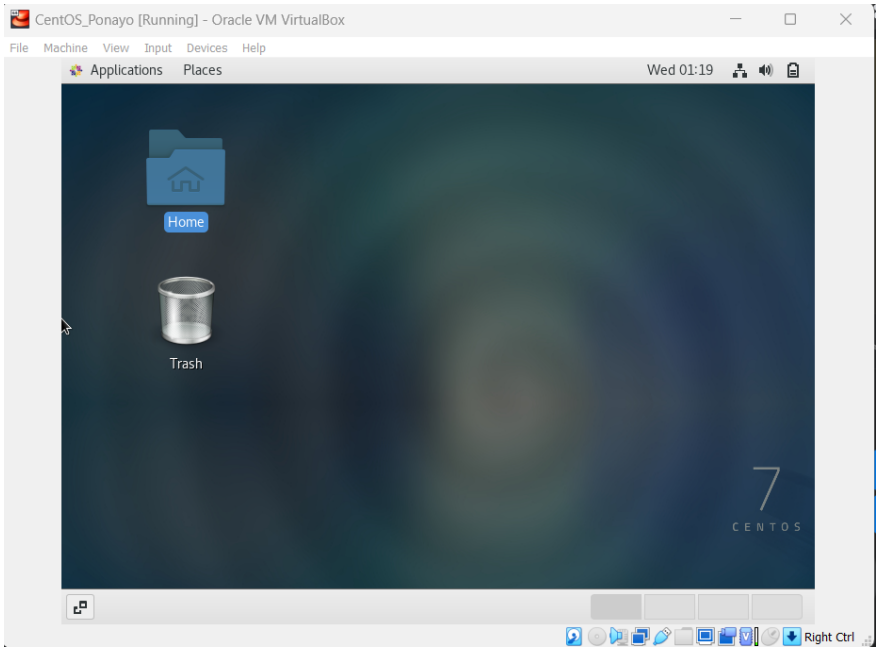
HD:



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

```
[ponayo@localhost ~]$ dnf install openssh-server
Error: This command has to be run under the root user.
[ponayo@localhost ~]$ su
Password:
[root@localhost ponayo]# dnf install openssh-server
CentOS-7 - Base                                  8.1 MB/s |  10 MB     00:01
CentOS-7 - Updates                               11 MB/s |  28 MB     00:02
CentOS-7 - Extras                                880 kB/s | 360 kB     00:00
Package openssh-server-7.4p1-21.el7.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@localhost ponayo]#
```

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

```
[ponayo@localhost ~]$ systemctl start sshd
[ponayo@localhost ~]$ systemctl enable sshd
[ponayo@localhost ~]$
```

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

```
[ponayo@localhost ~]$ systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enable
d)
   Active: active (running) since Wed 2023-09-06 02:13:56 PST; 6min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 1203 (sshd)
   CGroup: /system.slice/sshd.service
           └─1203 /usr/sbin/sshd -D

Sep 06 02:13:55 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Sep 06 02:13:55 localhost.localdomain sshd[1203]: Server listening on 0.0.0.0 port 22.
Sep 06 02:13:56 localhost.localdomain sshd[1203]: Server listening on :: port 22.
Sep 06 02:13:56 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
Hint: Some lines were ellipsized, use -l to show in full.
[ponayo@localhost ~]$
```

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

```
[ponayo@localhost ~]$ firewall-cmd --zone=public --permanent --add-service=ssh
Warning: ALREADY_ENABLED: ssh
success
[ponayo@localhost ~]$ firewall-cmd --reload
success
[ponayo@localhost ~]$
```

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

Task 3: Copy the Public Key to CentOS

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

```
ponayo@Workstation:~$ sudo apt install ssh
[sudo] password for ponayo:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ssh is already the newest version (1:8.9p1-3ubuntu0.3).
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2 libllvm13
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
ponayo@Workstation:~$
```

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

```
ponayo@Workstation:~$ ssh-copy-id ponayo@192.168.56.101
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ponayo/.ssh/id_
rsa.pub"
/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 prompted n
ow it is to install the new keys
ponayo@192.168.56.101's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'ponayo@192.168.56.101'"
and check to make sure that only the key(s) you wanted were added.

ponayo@Workstation:~$ ssh ponayo@192.168.56.101
Last login: Wed Sep  6 03:44:32 2023 from 192.168.56.102
[ponayo@localhost ~]$
```

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

```
[ponayo@localhost ~]$ cd ~/.ssh
[ponayo@localhost .ssh]$ ls
authorized_keys
[ponayo@localhost .ssh]$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCe285F+r2g9rdm4gwxaa0XBL2VU7hNHPAynUJvZlQ/C8026Od
XVTGw1OKNeH//L5leXMP2ZzFQX+nsK8IQagdAXcvbsfnPOuZxojkhwPLdD7CiKCj1YmQJQW7TBOPBVU+GHpLc4F
Tate3+RoONoPO5vEXqzdcNCu/DpvkDfWB61cp8An8ohKbBUPNVavgtRR1wVPO85aIJFRTtFs32qLXM8nsne/1v2
0r6rzV1Vx/ODuUxpOQzP/tijFQp3daBNxcfpi7w8Y9o5wZnBhluCvbyWdtp59Po6UgxQGFLS+AMJgrN1js7Chay
QT4Tmp1RrhTckMf3uaTIJH8hhwch/pmjWClD5JznCfyDfuAZIhnHj3cTmp1og6lTDf+6aMN0DFW+9t8pOEZKAbK
3fLeh0eKl7Wo3osVav1QCyQAbuvsNAmXBbujw2nTpG5Rqdvb9XoYEXm7SczfLdqSIY+oQhCqUhe0cAymMl8KI+J
yWwSHEFnpSV0xiM4f2hEu5be4m268= ponayo@Workstation
[ponayo@localhost .ssh]$
```

Task 4: Verify ssh remote connection

1. Using your local machine, connect to CentOS using ssh.
2. Show evidence that you are connected.

```
ponayo@Workstation:~$ ssh ponayo@192.168.56.101
The authenticity of host '192.168.56.101 (192.168.56.101)' can't be established.
ED25519 key fingerprint is SHA256:+rL/vlsIQCgivW6otszceC8jcMy3Z3cCM0FdJfmMAaI.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.101' (ED25519) to the list of known hosts.
ponayo@192.168.56.101's password:
Last login: Wed Sep  6 03:26:24 2023
[ponayo@localhost ~]$
```

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?
   - For my personal preference, I prefer Debian linux distributions over red hat. Using debian linux is much more easy to use than Red Hat in a way that you don't need to click the capture every time you're going to input a code. Aside from that, using Debian Linux is more practical to us since it is more modern and we can easily find or navigate the applications that we need.
2. What are the main diffence between Debian and Red Hat Linux distributions?
   - While using both Debian and RedHat Linux distributions. They have the same way of encoding syntax on the terminal but the interface has a big difference. While using Red Hat, it was a hassle to encode since you have to click the capture mode to encode freely inside the distribution while in debian you don't need to use capture mode. Aside from that, there was a huge difference when it came to the interface. Since RedHat or CentOS have 3 options: the application, places and terminal on the ribbon area and that's how you find what application you want to use. While in the debian, it was more easy since when you open the OS the all application will appear and it was way more organized than Red Hat