Name: Issa Victoria Peñas	**Date Performed:** 08/30/2022
Course/Section: CPE31S2	**Date Submitted:** 08/30/2022
Instructor: Dr. Jonathan Taylar	**Semester and SY:** 1st Semester (SY: 2022-2023)

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/

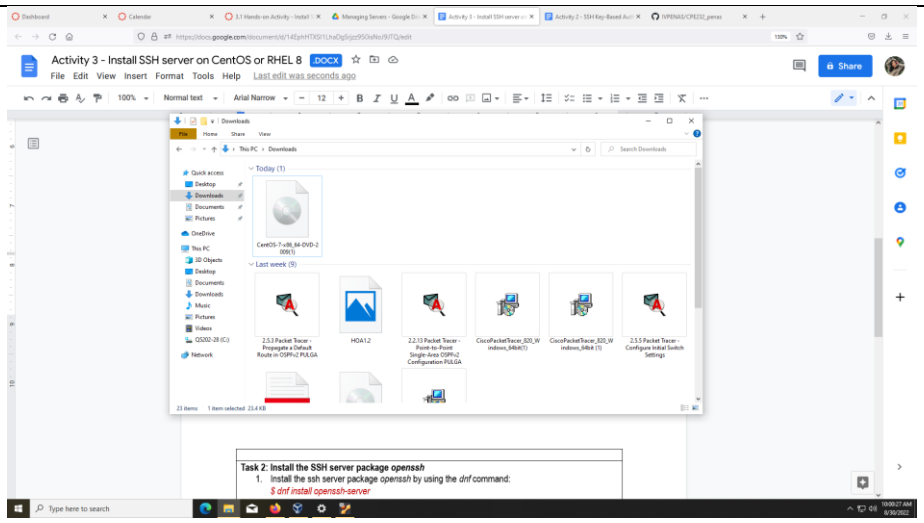

Figure 1.1. Shows the Downloaded .iso File of the CentOS

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

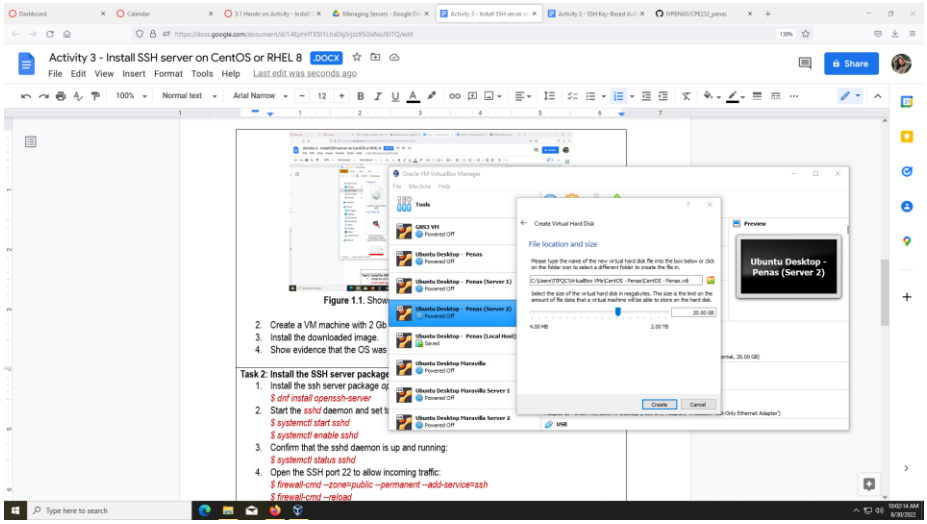

Figure 1.2. Shows the creation of a new VM Machine with its customized settings

3. Install the downloaded image.

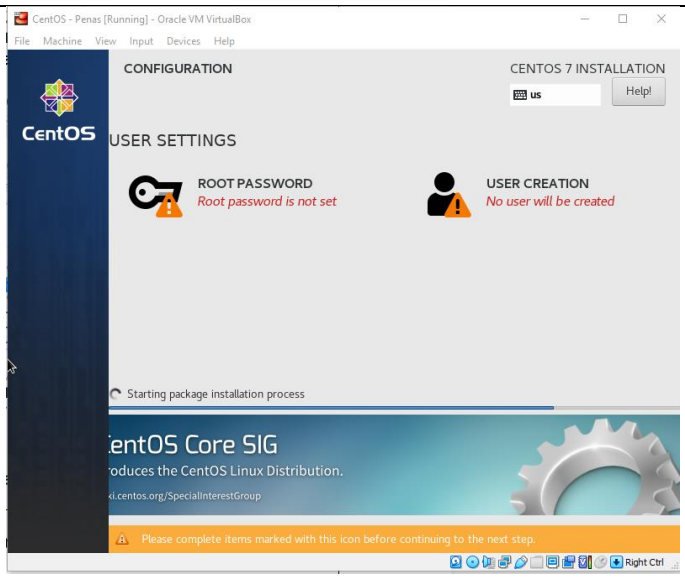

Figure 1.3. Installation is on-going

4. Show evidence that the OS was installed already.

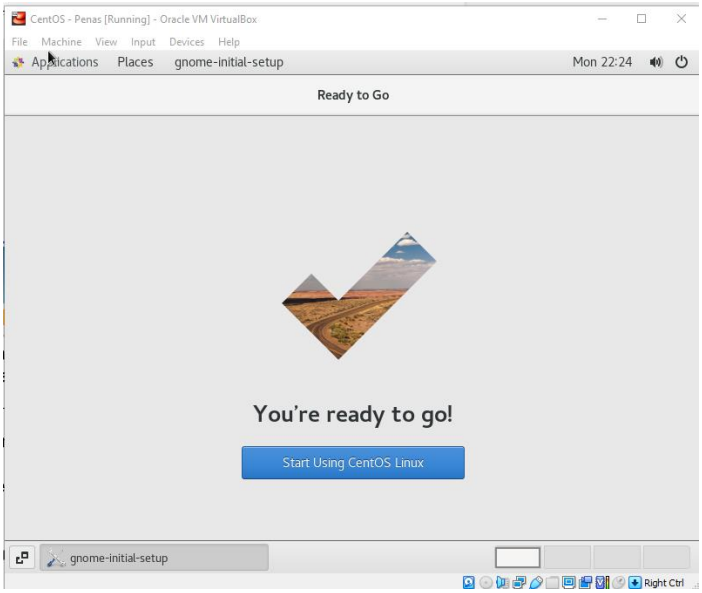

Figure 1.4. Installation Completed

Task 2: Install the SSH server package *openssh*

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

   $ dnf install openssh-server

```
[penas@localhost ~]$ dnf install openssh-server
Error: This command has to be run under the root user.
[penas@localhost ~]$ sudo dnf install openssh-server
[sudo] password for penas:
CentOS-7 - Base                                643 kB/s |  10 MB     00:16
CentOS-7 - Updates                             1.0 MB/s |  21 MB     00:21
CentOS-7 - Extras                               21 kB/s | 331 kB     00:16
Last metadata expiration check: 0:00:01 ago on Mon 29 Aug 2022 11:03:48 PM EDT.
Package openssh-server-7.4p1-21.el7.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[penas@localhost ~]$
```

Figure 1.5. Installing the openssh-server in the Terminal

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

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

Figure 1.6. Inputting the commands in the Terminal

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

```
[penas@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 Mon 2022-08-29 22:39:24 EDT; 25min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 1122 (sshd)
    Tasks: 1
   CGroup: /system.slice/sshd.service
           └─1122 /usr/sbin/sshd -D

Aug 29 22:39:24 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Aug 29 22:39:24 localhost.localdomain sshd[1122]: Server listening on 0.0.0.0 port 22.
Aug 29 22:39:24 localhost.localdomain sshd[1122]: Server listening on :: port 22.
Aug 29 22:39:24 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
Hint: Some lines were ellipsized, use -l to show in full.
[penas@localhost ~]$
```

Figure 1.7. The Status was activated

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

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

Figure 1.8. The Firewall was activated

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

```
[penas@localhost ~]$ systemctl reload sshd
```

Figure 1.9. Inputted the given command

Task 3: Copy the Public Key to CentOS

1. Make sure that *ssh* is installed on the local machine.
2. Using the command *ssh-copy-id*, connect your local machine to CentOS.

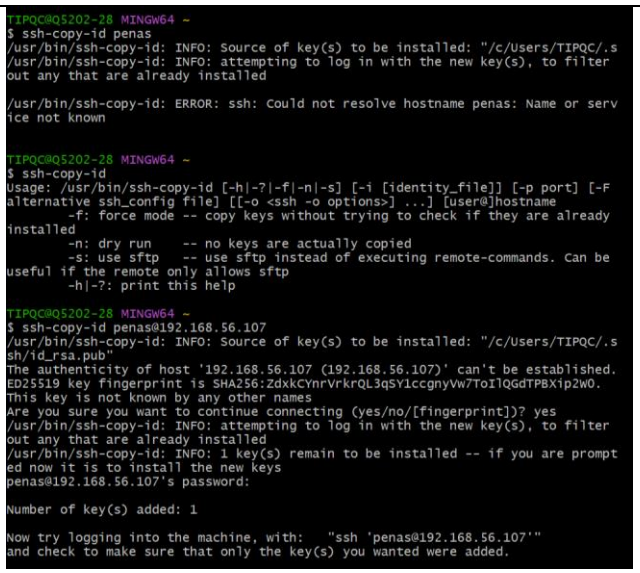

Figure 1.9. Copying the Public Key of the IP Address from the CentOS

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

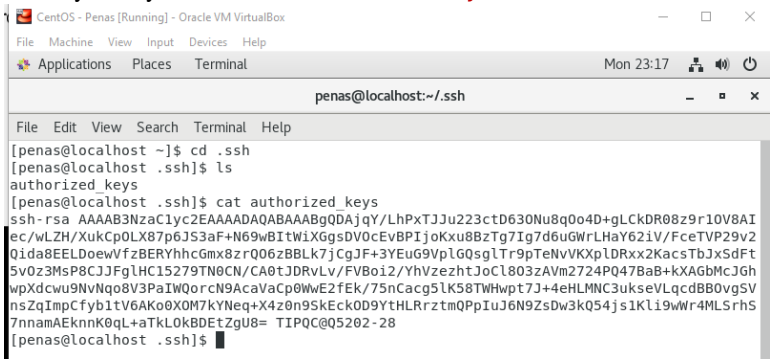

Figure 1.10. Verifying the authorized_keys within the CentOS Terminal

Task 4: Verify ssh remote connection

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

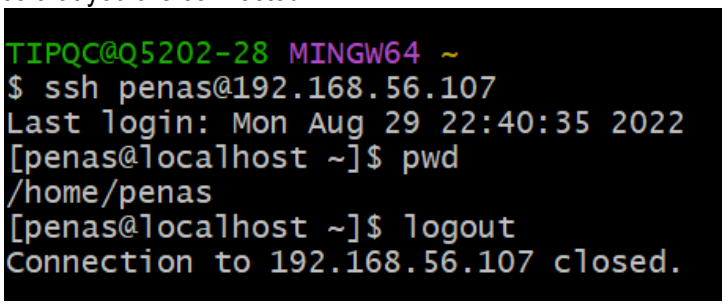

Figure 1.11. Connected to the Local Machine which is in the Git Bash

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?
    - In simple terms, Debian is commonly on **general use** as everyone can use it, some might find it confusing due to its stale GUI in Terminal which is hard to identify its previous commands and outputs unlike in Ubuntu Linux there are Identifications where it separates the new line and its output. Then RedHat is commonly used on Large Enterprises as they sell commercial distributions on Linux

2. What are the main difference between Debian and Red Hat Linux distributions?
    - Most of their main difference was their architecture and functions used where the table displayed below are only their MAIN key points. But in the end of the day it'll depend on the programmer, environment and its own preferred details

Debian	RedHat
Free Software	Open-Source Software
Their OS Family is under Unix-like	Their OS Family belongs to Linux
Is not under Commercial Linux Distribution	Is under Commercial Linux Distribution
Debian uses the apt-get dependency resolver	RedHat uses a yum dependency resolver.
Compatible in other Platforms (Linux,Windows, MacOS, more…)	More features than Debian such as their commands

Table 2.1. Here are some of the key differences between Debian and RedHat