Name: Shaniah Rose Hope M. Sumaoang	Date Performed: September 5, 2023
Course/Section: CPE 232-CPE31S5	Date Submitted: September 6, 2023
Instructor: Engr. Roman Richard	Semester and SY: 1st Semester, SY 2023-2024

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.

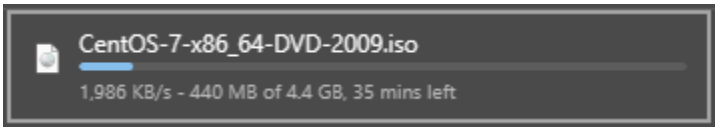
CentOS uses the RPM package format and YUM/DNF as the package manager.

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

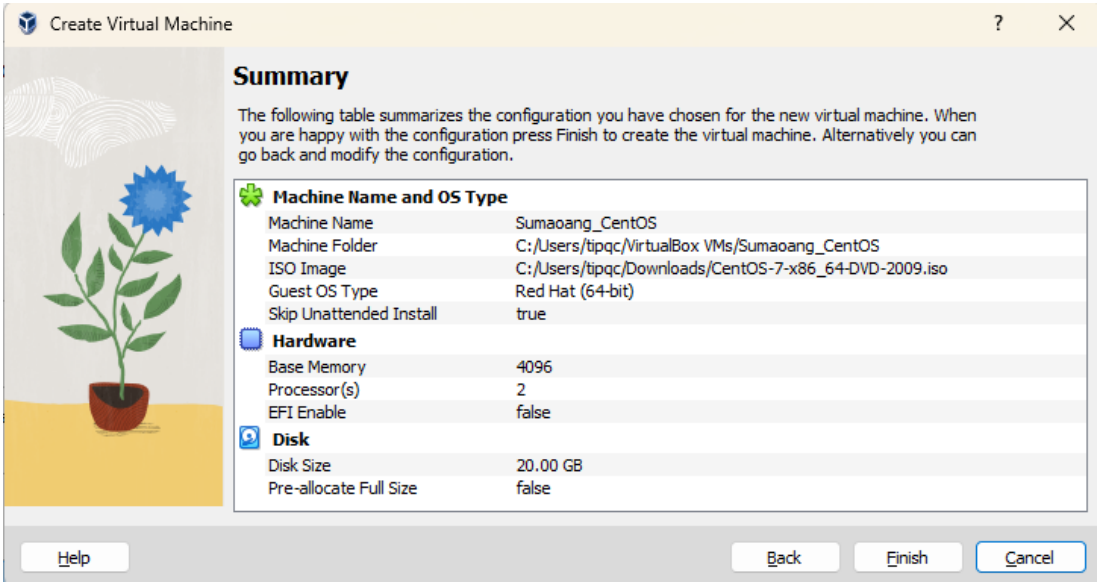
Both offer full-feature package management with network-based repository support, dependency checking and resolution, etc.. If you're familiar with one but not the other, you may have a little trouble switching over, but they're not overwhelmingly different. They both have similar features, just available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

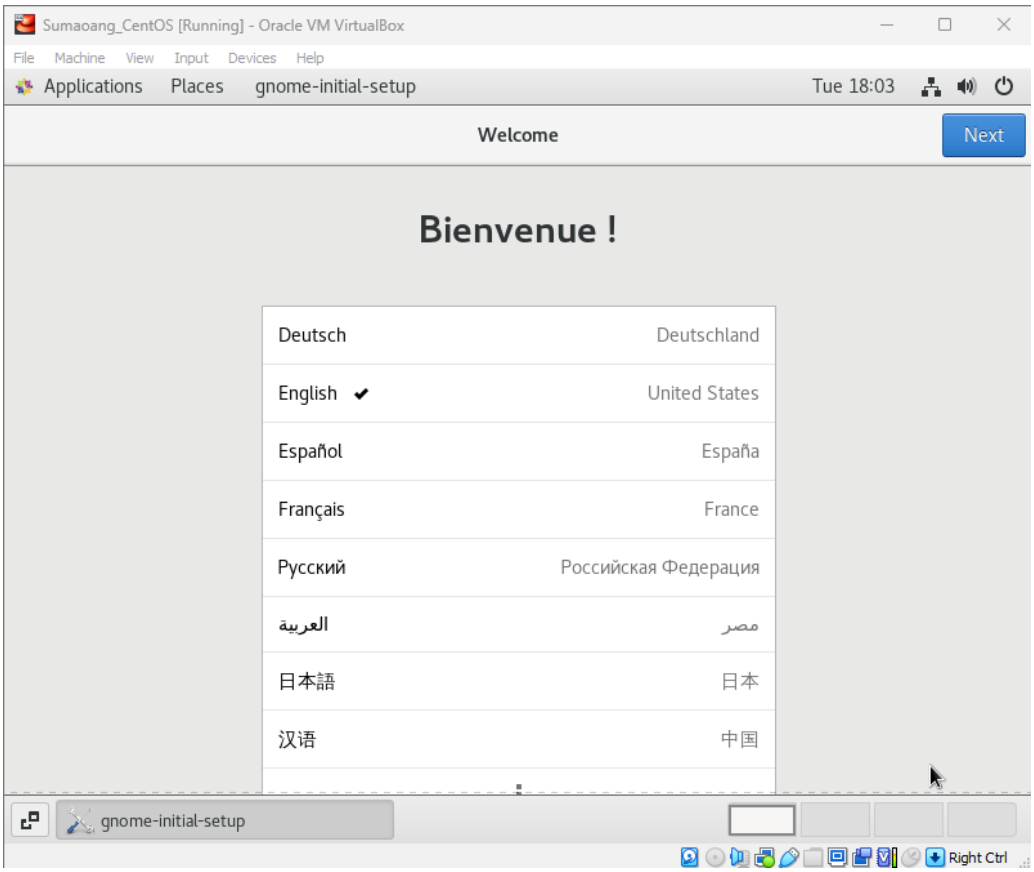


2. Create a VM machine with 2 Gb RAM and 20 Gb HD.



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

```
[srhmshan@localhost ~]$ su -
Password:
[root@localhost ~]# yum install dnf

Dependency Installed:
  dnf-data.noarch 0:4.0.9.2-2.el7_9          libcomps.x86_64 0:0.1.8-14.el7
  libdnf.x86_64 0:0.22.5-2.el7_9             libmodulemd.x86_64 0:1.6.3-1.el7
  librepo.x86_64 0:1.8.1-8.el7_9             libsolv.x86_64 0:0.6.34-4.el7
  python-enum34.noarch 0:1.0.4-1.el7         python2-dnf.noarch 0:4.0.9.2-2.el7_9
  python2-hawkey.x86_64 0:0.22.5-2.el7_9     python2-libcomps.x86_64 0:0.1.8-14.el7
  python2-libdnf.x86_64 0:0.22.5-2.el7_9

Complete!
```

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

   $ systemctl start sshd

```
[srhmshan@localhost ~]$ systemctl start sshd
[srhmshan@localhost ~]$
```

   $ systemctl enable sshd

```
[srhmshan@localhost ~]$ systemctl enable sshd
[srhmshan@localhost ~]$
```

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

   $ systemctl status sshd

```
[srhmshan@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 Tue 2023-09-05 22:15:51 PST; 49min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 1234 (sshd)
   CGroup: /system.slice/sshd.service
           └─1234 /usr/sbin/sshd -D

Sep 05 22:15:51 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Sep 05 22:15:51 localhost.localdomain sshd[1234]: Server listening on 0.0.0.0 port 22.
Sep 05 22:15:51 localhost.localdomain sshd[1234]: Server listening on :: port 22.
Sep 05 22:15:51 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
Hint: Some lines were ellipsized, use -l to show in full.
```

4. Open the SSH port 22 to allow incoming traffic:

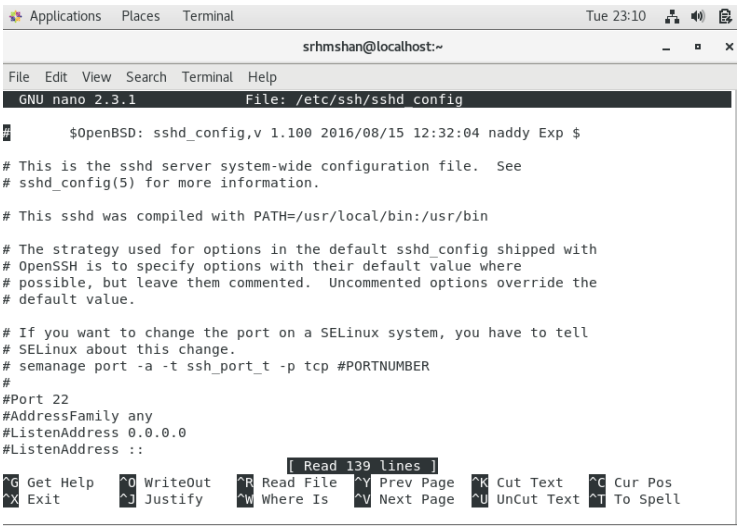
   $ firewall-cmd --zone=public --permanent --add-service=ssh

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

   $ firewall-cmd --reload

```
[srhmshan@localhost ~]$ firewall-cmd --reload
success
```

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




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

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.

```
srhmshan@localmachine:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/srhmshan/.ssh/id_rsa):
Created directory '/home/srhmshan/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/srhmshan/.ssh/id_rsa
Your public key has been saved in /home/srhmshan/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:PR5sQPnY7gZ0mO4cIAZkqtuE1NzuGroHIEzkVXcanGY srhmshan@localmachine
The key's randomart image is:
+---[RSA 4096]----+
|.oo....ooo       |
|.+= . .E=        |
|+o + .o..*       |
|=o  + . =++      |
|= .. o +So*      |
| =   .   +o.o    |
|. o. . o +.      |
|  ..o   o o      |
| oo.     .       |
+----[SHA256]-----+
```

```
srhmshan@localmachine:~$ ssh-copy-id srhmshan@192.168.56.106
The authenticity of host '192.168.56.106 (192.168.56.106)' can't be established.
ED25519 key fingerprint is SHA256:q7HNMi4GwfWQ00ZMt9i6OwEHOERc0vUHNoRUYJ20fxA.
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
srhmshan@192.168.56.106's password:
Ubuntu Software ) added: 1

Now try logging into the machine, with:   "ssh 'srhmshan@192.168.56.106'"
and check to make sure that only the key(s) you wanted were added.
```

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

```
[srhmshan@CentOS ~]$ cd ~/.ssh
[srhmshan@CentOS .ssh]$ ls
authorized_keys
[srhmshan@CentOS .ssh]$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQC1VXePRWj6CHVer7k0bsHMn4qtBuG5Yx5V9dZmaDdiYXnX9E2
FKDfDQbLnBVt45TKz6A2V3acMWJHZYL0FAqZFOqDOBFVgo++FJSELtqdAI9D+LjeX2WiONcY09BtTvEb0DBqGfi
lyuYW1yPTk0u+xBD8XIamidoMqr//GL6PZw84uxPxc1UxbEBMxyHkm1U61T6oSDvfRWkBl5AkWae/lfjaq2FihD
X1nJ8guF47FZhOO2LeEjv77BabbM7hCyxniBhaTI3G0qU+6HwPty5ZBH/hK75Ykd/u8aWpviy8TAn63ivnr9CMC
qAQpLCcb+h7WaznhzPlm3NgjE8Yagz3b/4PoI5ZpzRHvt1YzAqYTs/YHvZUcPZOu5k2f2AbDu4nxskZliE5bd6v
tgRk1ppCZsjgTDrLPEqvV6UlsBb5tDf+FhqvSmMHf/RMNLtrHBAA/70P3FbZKJnZd2u/woEZ8AmPT0uDBqy6D6B
wud4cdIgHMRGJh3h0FOGaGzQGz5zgvTPCnNVVAwxYlgvHfuB7tEsgnFWqcqWTiuTTDh9xmv43GlciSS9PYoLz2j
JVQUsHOGQm1K5/3LOuYuNXvaJ4M+hULY6tjo7yUPhkhBFWv3UPDYKhBbONSOQxhTmcbFjjbq76I3HSr96BR5vJA
JAd5BL5iFONjPZ9s8Q1j3SaYEw== srhmshan@localmachine
```

Task 4: Verify ssh remote connection

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

```
srhmshan@localmachine:~$ ssh srhmshan@192.168.56.106
Last login: Wed Sep  6 00:10:36 2023
[srhmshan@CentOS ~]$
```

2. Show evidence that you are connected.

```
[srhmshan@CentOS ~]$ who -a
           system boot  2023-09-06 00:10
           run-level 5  2023-09-06 00:10
srhmshan ? :0           2023-09-06 00:10   ?          2049 (:0)
srhmshan + pts/0        2023-09-06 00:10   .          2810 (:0)
srhmshan + pts/1        2023-09-06 00:33 00:06        3461 (192.168.56.107)
```

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 choosing the best distribution between Debian and Red Hat Linux, we must know the Architecture, Package Management, Filesystems, Kernel, Upgrading and Support. Choosing the best distribution to use, you must consider the technical requirements, internal resources, support options and business decisions.

2. What are the main differences between Debian and Red Hat Linux distributions?
   Debian is well-known for its community-driven development, stability, and dedication to free software principles, whereas Red Hat-based distributions such as RHEL and CentOS are preferred in enterprise environments due to their commercial support, long-term stability, and compatibility with enterprise-grade software and hardware.