Name: Buduan, Christian Aaron C.	**Date Performed:** 09/13/2024
Course/Section: CPE31S2	**Date Submitted:** 09/13/2024
Instructor: Engr. Robin Valenzuela	**Semester and SY:** 1st sem 2024-2025

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.

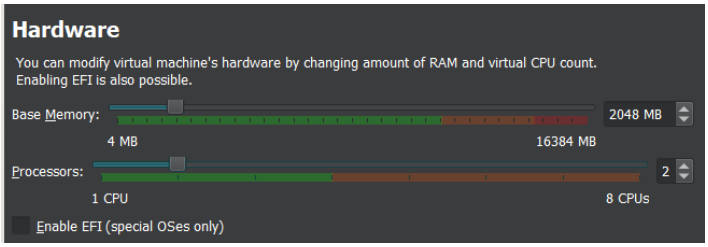
CentOS uses the RPM package format and YUM/DNF as the package manager.

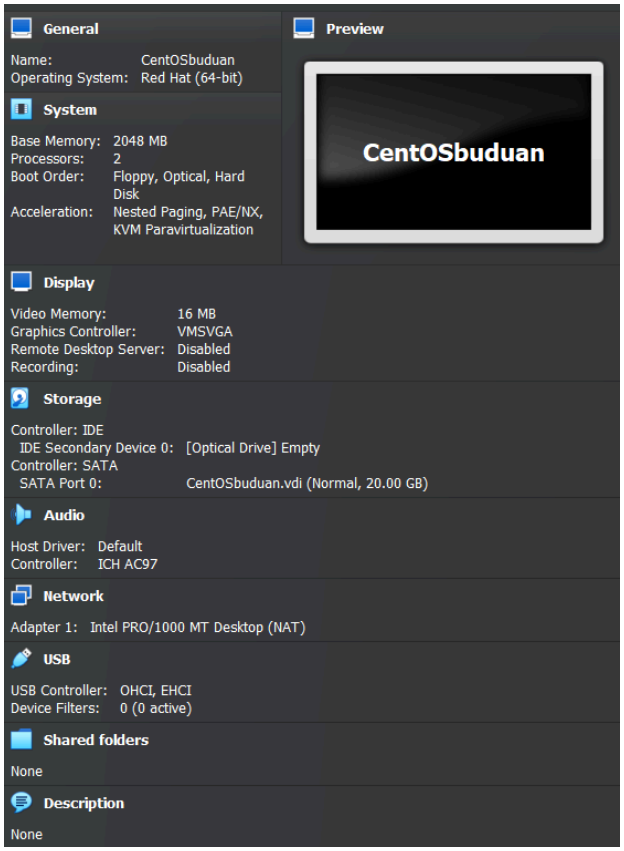
Debian uses the DEB package format and dpkg/APT as the package manager.

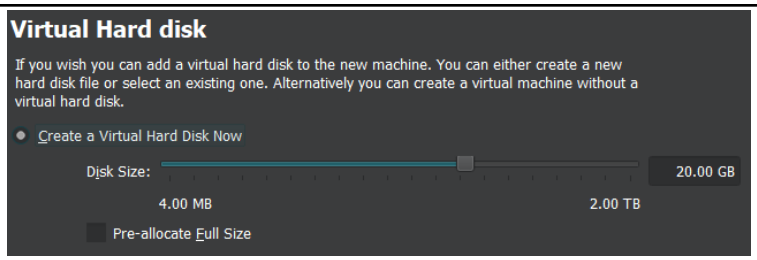
Both offer full-feature package management with network-based repository support, dependency checking and resolution, etc.. If you're familiar with one but not the other, you may have a little trouble switching over, but they're not overwhelmingly different. They both have similar features, just available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

```
[abuduan@localhost ~]$ sudo dnf install openssh-server

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for abuduan:
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                        425 kB/s | 8.2 MB     00:19
CentOS Stream 9 - AppStream                     749 kB/s |  20 MB     00:27
CentOS Stream 9 - Extras packages               1.5 kB/s |  19 kB     00:12
Package openssh-server-8.7p1-43.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

2. Start the *sshd* daemon and set to start after reboot:
   $ systemctl start sshd
   $ systemctl enable sshd
3. Confirm that the sshd daemon is up and running:
   $ systemctl status sshd

```
[abuduan@localhost ~]$ systemctl start sshd
[abuduan@localhost ~]$ systemctl enable sshd
[abuduan@localhost ~]$ systemctl status sshd
● sshd.service - OpenSSH server daemon
     Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: ena>
     Active: active (running) since Fri 2024-09-13 01:02:07 EDT; 1h 38min ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 892 (sshd)
      Tasks: 1 (limit: 10948)
     Memory: 2.2M
        CPU: 23ms
     CGroup: /system.slice/sshd.service
             └─892 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Sep 13 01:02:06 localhost.localdomain systemd[1]: Starting OpenSSH server daemo>
Sep 13 01:02:06 localhost.localdomain sshd[892]: Server listening on 0.0.0.0 po>
Sep 13 01:02:06 localhost.localdomain sshd[892]: Server listening on :: port 22.
Sep 13 01:02:07 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
```

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

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

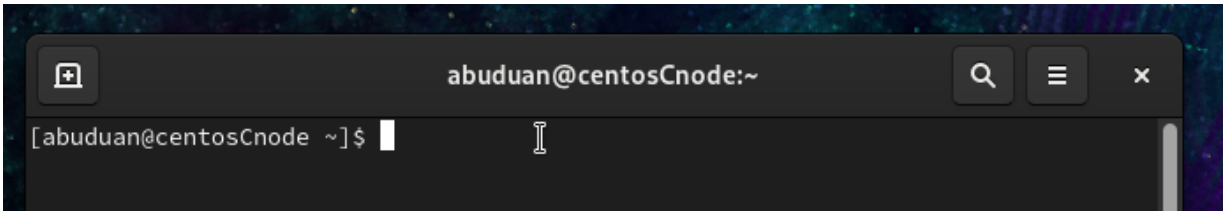
```
[abuduan@localhost ~]$ sudo cat /etc/ssh/sshd_config
[sudo] password for abuduan:
#       $OpenBSD: sshd_config,v 1.104 2021/07/02 05:11:21 dtucker Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.
```

```
#Match User anoncvs
#       X11Forwarding no
#       AllowTcpForwarding no
#       PermitTTY no
#       ForceCommand cvs server
[abuduan@localhost ~]$ systemctl reload sshd
```

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



```
qcacbuduan@Workstation:~$ ssh-copy-id -i ~/.ssh/id_rsa abuduan@centosCnode
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/qcacbuduan
/.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 promp
ted now it is to install the new keys
abuduan@centoscnode's password:

Number of key(s) added: 1

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

Task 4: Verify ssh remote connection

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

```
qcacbuduan@Workstation:~$ ssh abuduan@centosCnode
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Fri Sep 13 03:37:36 2024 from 192.168.56.13
[abuduan@centosCnode ~]$
```

```
[abuduan@centosCnode ~]$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fef2:a13e  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:f2:a1:3e  txqueuelen 1000  (Ethernet)
        RX packets 40  bytes 5405 (5.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 139  bytes 13198 (12.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.16  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::9e9c:87b2:a8cf:bdab  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:c6:cd:74  txqueuelen 1000  (Ethernet)
        RX packets 114  bytes 23129 (22.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 141  bytes 24917 (24.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

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?

- The choice between Debian and RedHat should be based on your specific needs, such as the intended use, budget, required support, and your level of expertise with Linux systems. We should consider these factors to make the right choice for your environment.

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

   - The differences between Debian and RedHat linux distribution is that Debian is known for its stability and is favored for general use, workstations, and servers, utilizing the APT package management system for software installation. When it comes to using Red Hat, it is tailored for enterprise environments, offering professional support and utilizing the RPM and YUM/DNF package management tools.