Name: Reyes, Alexzander J.	Date Performed: 29/08/2025
Course/Section: CPE212-CPE31S2	Date Submitted: 29/08/2025
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.

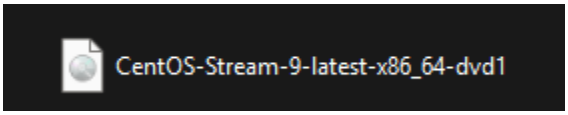
CentOS uses the RPM package format and YUM/DNF as the package manager.

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

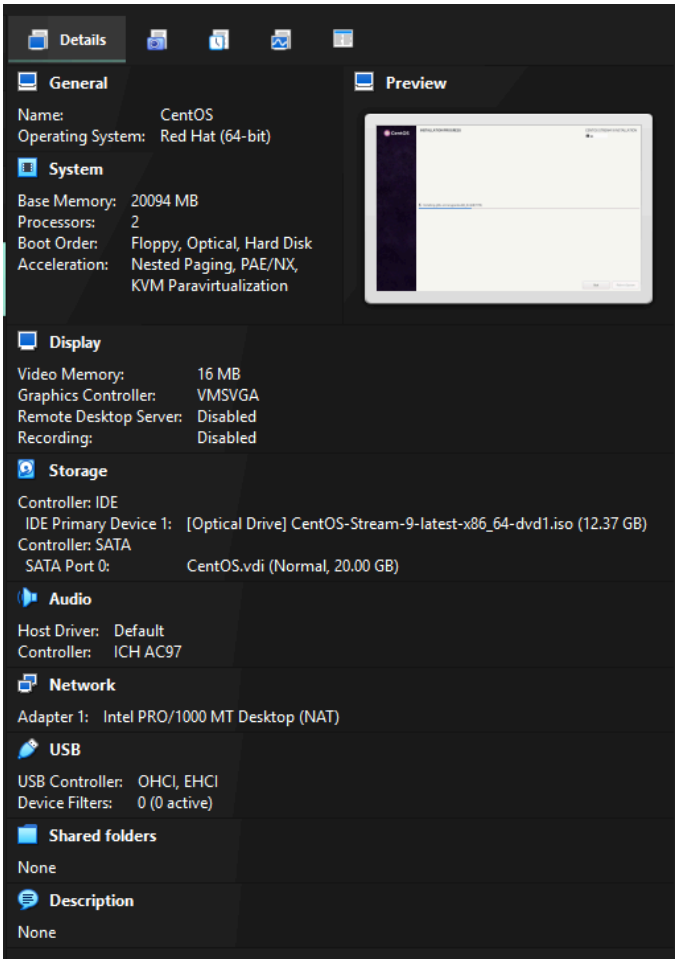
Both offer full-feature package management with network-based repository support, dependency checking and resolution, etc.. If you're familiar with one but not the other, you may have a little trouble switching over, but they're not overwhelmingly different. They both have similar features, just available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

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

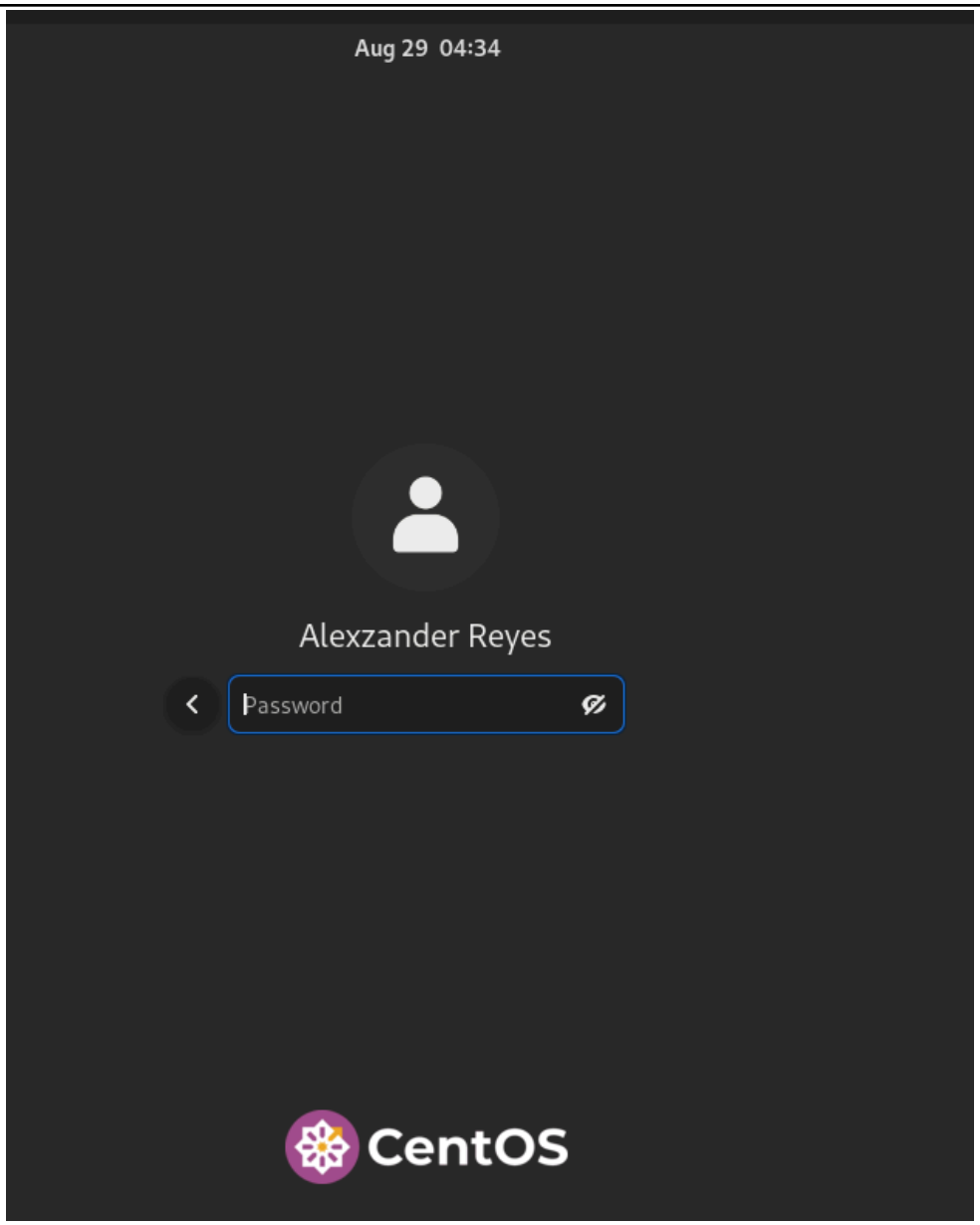
1. Download the image of the CentOS here: http://mirror.rise.ph/centos/7.9.2009/isos/x86_64/



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

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

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 Reyes:
Last metadata expiration check: 0:05:01 ago on Fri 29 Aug 2025 04:35:06 AM PST.
Package openssh-server-8.7p1-46.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

```
[Reyes@localhost ~]$ sudo systemctl start sshd
[Reyes@localhost ~]$ sudo systemctl enable sshd
```

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

```
[Reyes@localhost ~]$ sudo systemctl status sshd
● sshd.service - OpenSSH server daemon
     Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: ena>
     Active: active (running) since Fri 2025-08-29 04:13:58 PST; 26min ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 963 (sshd)
      Tasks: 1 (limit: 123489)
     Memory: 2.8M (peak: 3.1M)
        CPU: 14ms
     CGroup: /system.slice/sshd.service
             └─963 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 29 04:13:58 localhost.localdomain systemd[1]: Starting OpenSSH server daemo>
Aug 29 04:13:58 localhost.localdomain sshd[963]: Server listening on 0.0.0.0 po>
Aug 29 04:13:58 localhost.localdomain sshd[963]: Server listening on :: port 22.
Aug 29 04:13:58 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
lines 1-16/16 (END)
[1]+  Stopped                 sudo systemctl status sshd
```

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

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

```
[Reyes@localhost ~]$ sudo nano /etc/ssh/sshd_config
[Reyes@localhost ~]$ sudo systemctl reload sshd
```

Task 3: Copy the Public Key to CentOS

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

```
vboxuser@Workstation:~$ ssh -V
OpenSSH_9.6p1 Ubuntu-3ubuntu13.13, OpenSSL 3.0.13 30 Jan 2024
```

```
vboxuser@Workstation:~$ ping 192.168.26.13
PING 192.168.26.13 (192.168.26.13) 56(84) bytes of data.
64 bytes from 192.168.26.13: icmp_seq=1 ttl=64 time=1.76 ms
64 bytes from 192.168.26.13: icmp_seq=2 ttl=64 time=1.99 ms
64 bytes from 192.168.26.13: icmp_seq=3 ttl=64 time=0.519 ms
64 bytes from 192.168.26.13: icmp_seq=4 ttl=64 time=1.34 ms
```

   - CentOS ip address

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

```
vboxuser@Workstation:~$ ssh-copy-id Reyes@192.168.26.13
The authenticity of host '192.168.26.13 (192.168.26.13)' can't be established.
ED25519 key fingerprint is SHA256:QFzlPXUYCG38v5IjRcfsf5/gUEt9NvZ1Bsg3egE4IQI.
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: 2 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
Reyes@192.168.26.13's password:

Number of key(s) added: 2

Now try logging into the machine, with:   "ssh 'Reyes@192.168.26.13'"
and check to make sure that only the key(s) you wanted were added.
```

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

```
[Reyes@localhost ~]$ cat ~/.ssh/authorized_keys
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIEZbYyHUvvjdqL/zCUnAiUa/kElD/WHjgbOQx0MH38kr
 vboxuser@Workstation
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDhsJ3at4ZnGt3mCXfedE94iYWbAEmLKX6Gw2+fXyeY
HW3c9bXBlJkR+5FWh0vCvSF+zWM0dQqKk17ciaRxmz8+B2GBHNwFEmjvzMVFe7AsP+T/zeIygBeZHheb
v7RzzSN/aDgf8IkNaIWAVAH0Haiw68UpV+c+3w5/HA/KLoAopnhI2govQj1omky5VWSsw+UAE8V/H8qe
Gj+aq/SK/LxuQxT9silPQ++NyfeBLhL/gQ0lpjWdk5cr22OVX6QJ6dSgdUa+pDSn6yhm52WDygckaw8c
DYnaHIQZ9KAop4eC3oHDI+Xh1yeI6zxU8GexejSzrypdVqPEPHsdkWTm+VP6exR9Q7dmyVn4FQjOySGg
htFzBwpwQtfZlJHChB+ZDMPV609fkVYwQSFKGzaFE/26cNkteVusIXb8QK1r5lwlToxoiHifhFhQRCQk
bt7tQtjbwoHcvbBV8oj/wIGgsLmSqOD57mBMuTFe9HOBIoCsjLo2d7OFUrGbgrcMqxmhQIWn8sqJPUVV
gss3OdXClgs4MakiWf+o2liHltlE8z5TB8GC1/VcXZVCU49cFGxwhDum9pbJwIqq45P8j1w4c3x01WNi
bAU2qFQS3xPsI5l7xKei329BHHKdUyc6NGLydRtDwRIsn2xljNS3dHJkQudCzQYKKVfDEkytzV4HFVbs
uQ== vboxuser@Workstation
```

Task 4: Verify ssh remote connection

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

```
vboxuser@Workstation:~$ ping 192.168.26.13
PING 192.168.26.13 (192.168.26.13) 56(84) bytes of data.
64 bytes from 192.168.26.13: icmp_seq=1 ttl=64 time=1.24 ms
64 bytes from 192.168.26.13: icmp_seq=2 ttl=64 time=2.28 ms
64 bytes from 192.168.26.13: icmp_seq=3 ttl=64 time=0.662 ms
```

```
[Reyes@localhost ~]$ ping 192.168.26.10
PING 192.168.26.10 (192.168.26.10) 56(84) bytes of data.
64 bytes from 192.168.26.10: icmp_seq=1 ttl=64 time=1.06 ms
64 bytes from 192.168.26.10: icmp_seq=2 ttl=64 time=0.987 ms
64 bytes from 192.168.26.10: icmp_seq=3 ttl=64 time=1.59 ms
64 bytes from 192.168.26.10: icmp_seq=4 ttl=64 time=1.43 ms
64 bytes from 192.168.26.10: icmp_seq=5 ttl=64 time=0.413 ms
```

```
vboxuser@Workstation:~$ ssh Reyes@192.168.26.13
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Fri Aug 29 05:19:45 2025
[Reyes@localhost ~]$
logout
Connection to 192.168.26.13 closed.
```

- using ip address

```
vboxuser@Workstation:~$ ssh Reyes@CentOS
The authenticity of host 'centos (192.168.26.13)' can't be established.
ED25519 key fingerprint is SHA256:QFzlPXUYCG38v5IjRcfsf5/gUEt9NvZ1Bsg3egE4IQI.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:10: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? YES
Warning: Permanently added 'centos' (ED25519) to the list of known hosts.
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Fri Aug 29 05:50:00 2025
[Reyes@CentOS ~]$
logout
```

- using hostname

2. Show evidence that you are connected.

```
vboxuser@Workstation:~$ ssh Reyes@192.168.26.13
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Fri Aug 29 05:36:39 2025 from 192.168.26.10
[Reyes@localhost ~]$ hostname
localhost.localdomain
[Reyes@localhost ~]$ whoami
Reyes
```

- using ip address

```
vboxuser@Workstation:~$ ssh Reyes@CentOS
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Fri Aug 29 05:53:59 2025 from 192.168.26.10
[Reyes@CentOS ~]$ hostname
CentOS
[Reyes@CentOS ~]$ whoami
Reyes
```

- using hostname

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 factors to be considered when selecting the optimal distribution between Debian and Red Hat include stability, support, packages handling, and use case. Debian would be the one to choose when a stable system, community oriented, and a huge amount of free software packages are desired. It is suitable in the context where the open-source principles and long-term reliability are relevant. Red Hat, though, is more appropriate in enterprises that engage in professional and commercial assistance, certified hardware compatibility and other enterprise features. You should also take into consideration whether you would like to have a free community supported OS or a subscribing based solution with official vendor support.

2. What are the main differences between Debian and Red Hat Linux distributions?
   - The main differences between Debian and Red Hat lie in their management approach, target audience, and package systems. Debian is a volunteer project based on DEB packages, and the APT package manager, with an emphasis on free software and extensive hardware availability. Red Hat is a commercial delivery targeted at companies, in the form of RPM packages handled with YUM or DNF, professional support, certification, and enterprise quality tools. Also, the Debian releases are more stable and free software oriented and Red Hat is more certified in reliability, security updates and vendor-supported services.