Name:Ramirez, Kiel Louis A.	Date Performed: 9-5-25
Course/Section: CPE 212-CPE31S2	Date Submitted: 9-5-25
Instructor: Sir Rbbin	Semester and SY: 1st 2025-2026

<div align="center">

Activity 3: Install SSH server on CentOS or RHEL 8

</div>

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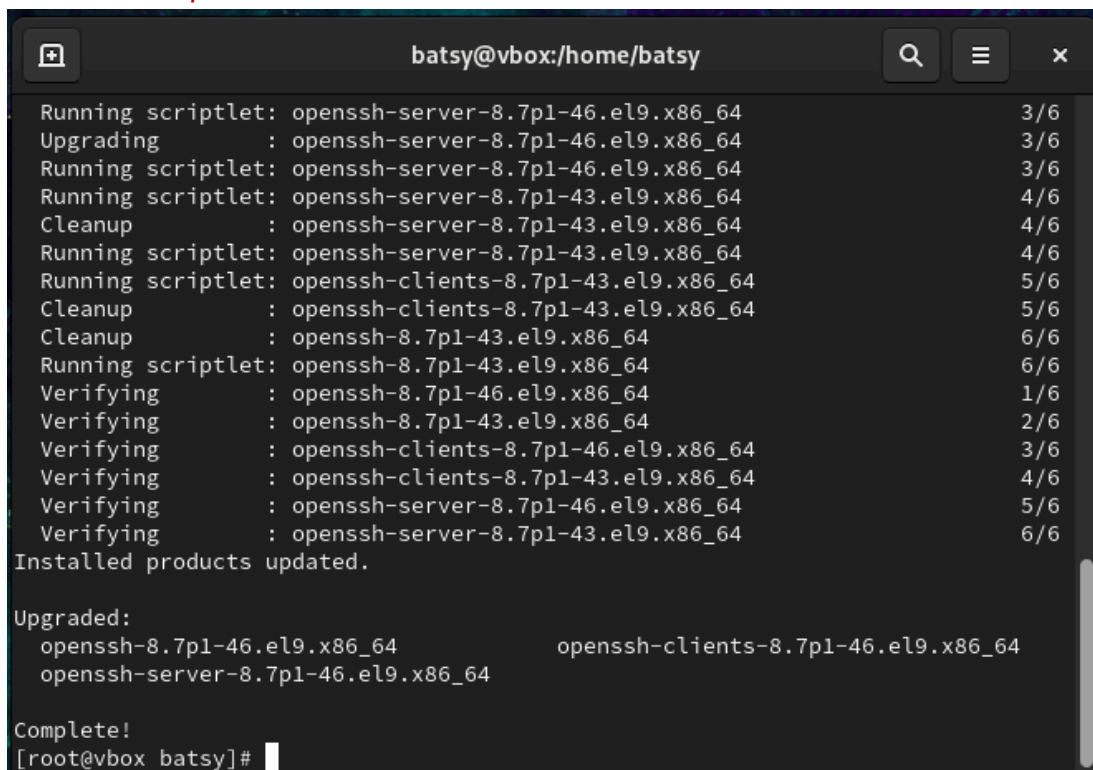
Task 1: Download the CentOS or RHEL-8 image (Create screenshots of the following)

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


```
┌────────────────────────────────────────────────────────────────────────────────┐
│  ⊞                        batsy@vbox:/home/batsy              Q   ≡   ✕          │
├────────────────────────────────────────────────────────────────────────────────┤
│   Running scriptlet: openssh-server-8.7p1-46.el9.x86_64                    3/6   │
│   Upgrading       : openssh-server-8.7p1-46.el9.x86_64                     3/6   │
│   Running scriptlet: openssh-server-8.7p1-46.el9.x86_64                    3/6   │
│   Running scriptlet: openssh-server-8.7p1-43.el9.x86_64                    4/6   │
│   Cleanup         : openssh-server-8.7p1-43.el9.x86_64                     4/6   │
│   Running scriptlet: openssh-server-8.7p1-43.el9.x86_64                    4/6   │
│   Running scriptlet: openssh-clients-8.7p1-43.el9.x86_64                   5/6   │
│   Cleanup         : openssh-clients-8.7p1-43.el9.x86_64                    5/6   │
│   Cleanup         : openssh-8.7p1-43.el9.x86_64                            6/6   │
│   Running scriptlet: openssh-8.7p1-43.el9.x86_64                           6/6   │
│   Verifying       : openssh-8.7p1-46.el9.x86_64                            1/6   │
│   Verifying       : openssh-8.7p1-43.el9.x86_64                            2/6   │
│   Verifying       : openssh-clients-8.7p1-46.el9.x86_64                    3/6   │
│   Verifying       : openssh-clients-8.7p1-43.el9.x86_64                    4/6   │
│   Verifying       : openssh-server-8.7p1-46.el9.x86_64                     5/6   │
│   Verifying       : openssh-server-8.7p1-43.el9.x86_64                     6/6   │
│ Installed products updated.                                                      │
│                                                                                  │
│ Upgraded:                                                                        │
│   openssh-8.7p1-46.el9.x86_64              openssh-clients-8.7p1-46.el9.x86_64   │
│   openssh-server-8.7p1-46.el9.x86_64                                             │
│                                                                                  │
│ Complete!                                                                        │
│ [root@vbox batsy]# █                                                             │
└────────────────────────────────────────────────────────────────────────────────┘
```
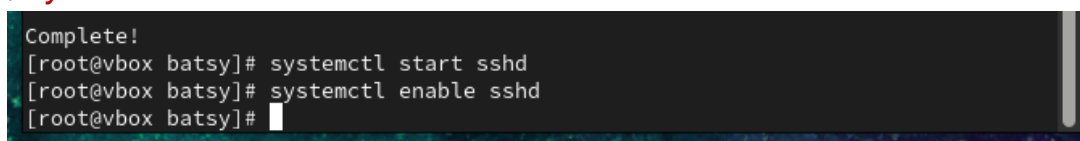
2. Start the *sshd* daemon and set to start after reboot:
   $ systemctl start sshd
   $ systemctl enable sshd

```
Complete!
[root@vbox batsy]# systemctl start sshd
[root@vbox batsy]# systemctl enable sshd
[root@vbox batsy]# █
```

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

```
Complete!
[root@vbox batsy]# systemctl start sshd
[root@vbox batsy]# systemctl enable sshd
[root@vbox batsy]# systemctl status sshd
● sshd.service - OpenSSH server daemon
     Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: ena>
     Active: active (running) since Fri 2025-09-05 17:29:44 PST; 4min 47s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 3226 (sshd)
      Tasks: 1 (limit: 10949)
     Memory: 1.4M
        CPU: 7ms
     CGroup: /system.slice/sshd.service
             └─3226 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Sep 05 17:29:44 vbox systemd[1]: Starting OpenSSH server daemon...
Sep 05 17:29:44 vbox sshd[3226]: Server listening on 0.0.0.0 port 22.
Sep 05 17:29:44 vbox sshd[3226]: Server listening on :: port 22.
Sep 05 17:29:44 vbox systemd[1]: Started OpenSSH server daemon.
^X
[1]+  Stopped                 systemctl status sshd
[root@vbox batsy]#
```

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

```
tmpfs                        181720       96     181624    1% /run/user/1000
[root@vbox batsy]# firewall-cmd --zone=public --permanent --add-service=ssh
Warning: ALREADY_ENABLED: ssh
success
[root@vbox batsy]# firewall-cmd --reload
success
[root@vbox batsy]#
```

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

```
success
[root@vbox batsy]# systemctl reload sshd
[root@vbox batsy]# 
```

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 *authorize`d_keys*.

```
vboxuser@LocalMachine:~$ ssh batsy@192.168.56.112
sign_and_send_pubkey: signing failed: agent refused operation
batsy@192.168.56.112's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Fri Sep  5 17:53:16 2025
[batsy@localhost ~]$
```

Task 4: Verify ssh remote connection
1. Using your local machine, connect to CentOS using ssh.
2. Show evidence that you are connected.

```
vboxuser@LocalMachine:~$ ssh batsy@192.168.56.112
sign_and_send_pubkey: signing failed: agent refused operation
batsy@192.168.56.112's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Fri Sep  5 17:53:16 2025
[batsy@localhost ~]$ hostname
localhost.localdomain
[batsy@localhost ~]$ whoami
batsy
[batsy@localhost ~]$ uname -a
Linux localhost.localdomain 5.14.0-505.el9.x86_64 #1 SMP PREEMPT_DYNAMIC Thu Se
p 5 07:54:07 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux
[batsy@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?
   -Consider your use case. Debian is better for community driven projects and servers, while Red Hat is great for companies that need paid support. Check your

package management preferences. Debian uses APT, and Red Hat uses YUM/DNF. Also, think about whether you need stability or the latest software, as well as if commercial support or certifications are important to you.

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

   -Debian is a free, community-supported distribution that has a slower release cycle and focuses on stability. It uses APT for package management. Red Hat is a commercial distribution aimed at enterprises. It offers paid support, a long term lifecycle, and certification, using YUM/DNF for packages. Debian serves as the source for many distributions, such as Ubuntu, while Red Hat maintains Fedora as its source project.