

Name: BONIFACIO, REDJ GUILIAN F.	Date Performed: September 5, 2025
Course/Section: CPE31S4	Date Submitted: September 5, 2025
Instructor: Engr. VALENZUELA, ROBIN	Semester and SY: 2nd Semester SY 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.

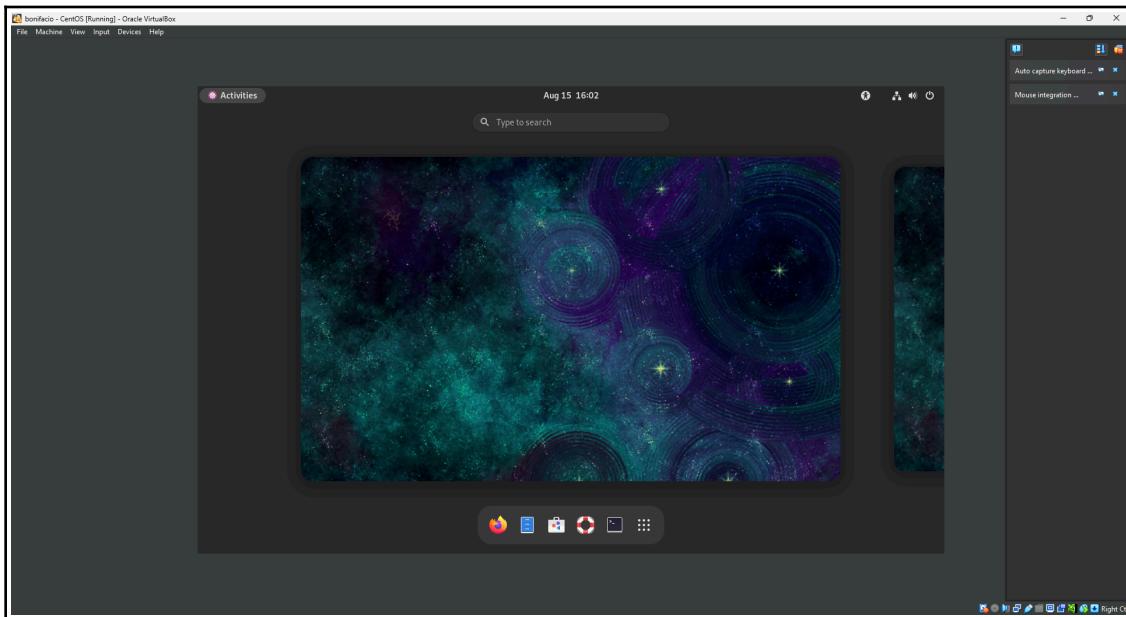
CentOS uses the RPM package format and YUM/DNF as the package manager.

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

```
[rdjbnfc@vbox ~] $ dnf install openssh-server
Not root, Subscription Management repositories not updated
Error: This command has to be run with superuser privileges (under the root user
on most systems).
[rdjbnfc@vbox ~] $ 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 rdjbnfc:
Sorry, try again.
[sudo] password for rdjbnfc:
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           3.3 kB/s | 4.5 kB   00:01
CentOS Stream 9 - BaseOS           1.1 MB/s | 8.8 MB   00:07
```

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
```

```
[rdjbnfc@vbox ~] $ sudo systemctl start sshd
[rdjbnfc@vbox ~] $ sudo systemctl enable sshd
[rdjbnfc@vbox ~] $ 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 2025-08-15 16:12:01 PST; 1min 2s ago
      Docs: man:sshd(8)
             man:sshd_config(5)
    Main PID: 3239 (sshd)
       Tasks: 1 (limit: 32189)
      Memory: 1.4M
        CPU: 8ms
       CGroup: /system.slice/sshd.service
               └─3239 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 15 16:12:01 vbox systemd[1]: Starting OpenSSH server daemon...
Aug 15 16:12:01 vbox sshd[3239]: Server listening on 0.0.0.0 port 22.
Aug 15 16:12:01 vbox sshd[3239]: Server listening on :: port 22.
Aug 15 16:12:01 vbox systemd[1]: Started OpenSSH server daemon.
lines 1-16/16 (END)
```

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

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

```
$ firewall-cmd --reload
```

```
rdjbnfc@vbox:~ — sudo systemctl ... x rdjbnfc@vbox:~ x
[rdjbnfc@vbox ~]$ sudo firewall-cmd --zone-public --permanent -- add-service-ssh
[sudo] password for rdjbnfc:
usage: 'firewall-cmd --help' for usage information or see firewall-cmd(1) man page
firewall-cmd: error: unrecognized arguments: --zone-public -- add-service-ssh
[rdjbnfc@vbox ~]$ sudo firewall-cmd --reload
success
[rdjbnfc@vbox ~]$
```

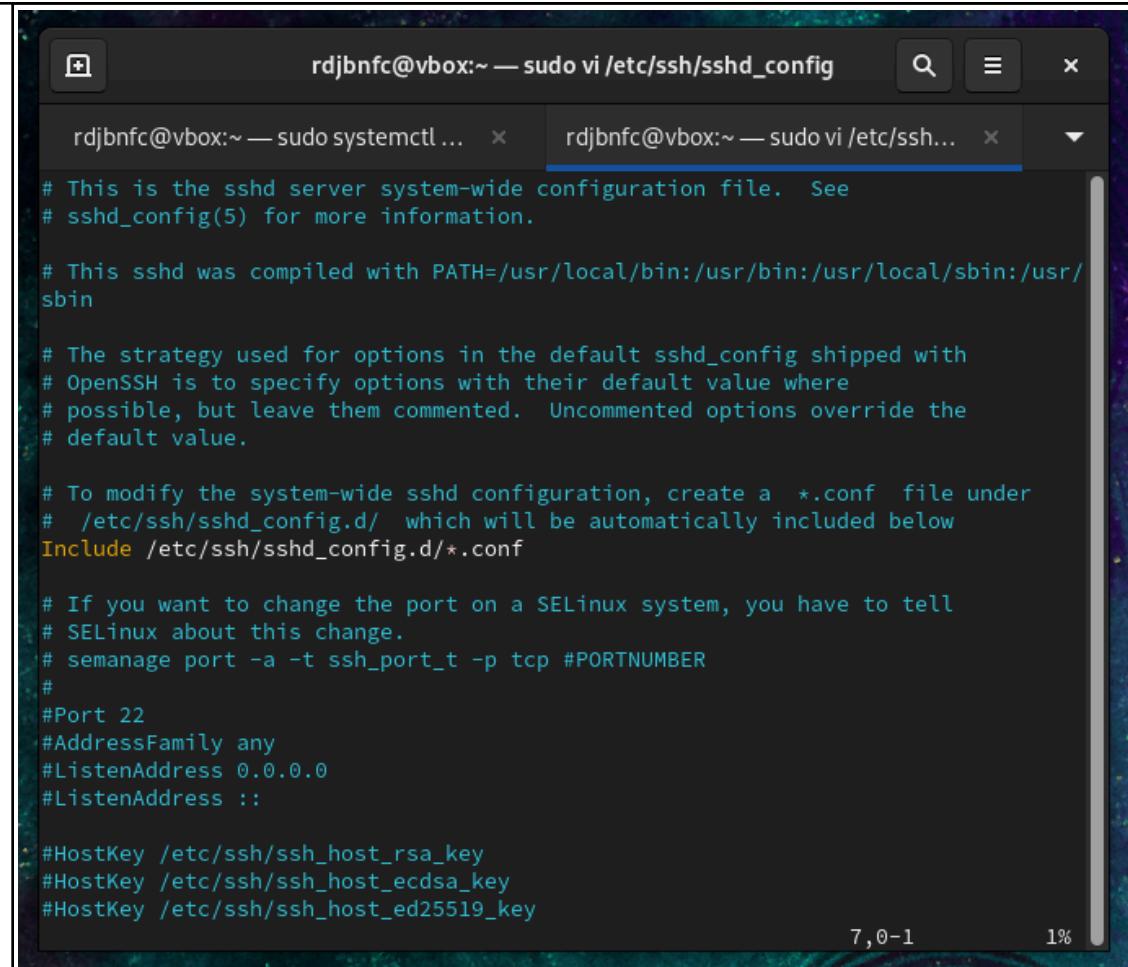
- Locate the ssh server man config file `/etc/ssh/sshd_config` and perform custom configuration. Every time you make any change to the `/etc/ssh/sshd-config` configuration file reload the `sshd` service to apply changes:

```
$ systemctl reload sshd
```

```
rdjbnfc@vbox:~ — sudo vi /etc/ssh/sshd_config x rdjbnfc@vbox:~ — sudo vi /etc/ssh... x
E325: ATTENTION
Found a swap file by the name "/etc/ssh/.sshd_config.swp"
    owned by: root    dated: Fri Aug 15 16:20:16 2025
    file name: /etc/ssh/sshd_config
    modified: YES
    user name: root    host name: vbox
    process ID: 34074
While opening file "/etc/ssh/sshd_config"
    dated: Tue Jul 22 23:50:16 2025

(1) Another program may be editing the same file. If this is the case,
    be careful not to end up with two different instances of the same
    file when making changes. Quit, or continue with caution.
(2) An edit session for this file crashed.
    If this is the case, use ":recover" or "vim -r /etc/ssh/sshd_config"
    to recover the changes (see ":help recovery").
    If you did this already, delete the swap file "/etc/ssh/.sshd_config.swp"
    to avoid this message.

Swap file "/etc/ssh/.sshd_config.swp" already exists!
[O]pen Read-Only, (E)dit anyway, (R)ecover, (D)elete it, (Q)uit, (A)bort: x
```



The screenshot shows a terminal window with two tabs open. The active tab is titled "rdjbnfc@vbox:~ — sudo vi /etc/ssh/sshd_config". It displays the configuration file for the sshd server. The file contains several comments and configuration options, including port settings, host keys, and SELinux rules. The bottom right corner of the terminal shows the status "7,0-1" and "1%".

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

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/
sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

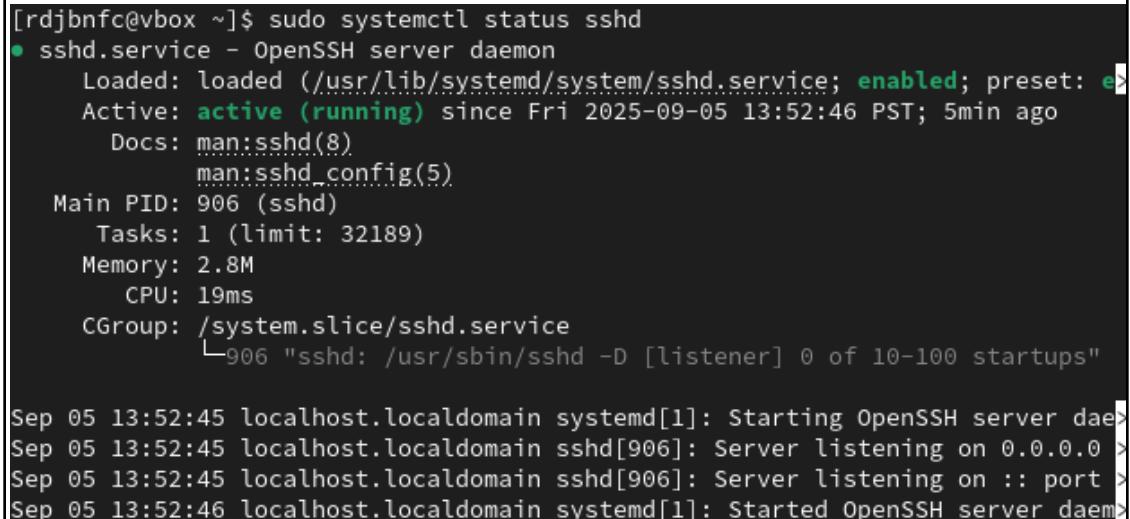
# To modify the system-wide sshd configuration, create a *.conf file under
# /etc/ssh/sshd_config.d/ which will be automatically included below
Include /etc/ssh/sshd_config.d/*.conf

# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key
```

Task 3: Copy the Public Key to CentOS

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



The screenshot shows a terminal window displaying the output of the command "sudo systemctl status sshd". It shows that the sshd service is active and running. The terminal also displays log messages from the sshd daemon starting up and listening on port 0.0.0.0.

```
[rdjbnfc@vbox ~]$ sudo systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-09-05 13:52:46 PST; 5min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 906 (sshd)
      Tasks: 1 (limit: 32189)
     Memory: 2.8M
        CPU: 19ms
      CGroup: /system.slice/sshd.service
              └─906 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Sep 05 13:52:45 localhost.localdomain systemd[1]: Starting OpenSSH server dae>
Sep 05 13:52:45 localhost.localdomain sshd[906]: Server listening on 0.0.0.0 >
Sep 05 13:52:45 localhost.localdomain sshd[906]: Server listening on :: port >
Sep 05 13:52:46 localhost.localdomain systemd[1]: Started OpenSSH server daem>
```

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

```
bonifacio@workstation:~/CPE232_RedjBonifacio$ ssh-copy-id rdjbnfc@192.168.56.116
/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 prompted now it is to install the new keys
rdjbnfc@192.168.56.116's password:

Number of key(s) added: 2

Now try logging into the machine, with: "ssh 'rdjbnfc@192.168.56.116'" and check to make sure that only the key(s) you wanted were added.
```

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

```
bonifacio@workstation:~/CPE232_RedjBonifacio$ ssh rdjbnfc@192.168.56.116
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Fri Sep 5 16:11:16 2025 from 192.168.56.116
[rdjbnfc@vbox ~]$ ls -l ~/.ssh
total 20
-rw----- 1 rdjbnfc rdjbnfc 1419 Sep 5 16:16 authorized_keys
-rw----- 1 rdjbnfc rdjbnfc 2602 Sep 5 13:48 id_rsa
-rw-r--r-- 1 rdjbnfc rdjbnfc 566 Sep 5 13:48 id_rsa.pub
-rw----- 1 rdjbnfc rdjbnfc 840 Sep 5 16:06 known_hosts
-rw-r--r-- 1 rdjbnfc rdjbnfc 96 Sep 5 16:06 known_hosts.old
[rdjbnfc@vbox ~]$ cat ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDrhs2omR6brOzJGlZ0eaYAecgkA0g0CopaekqIJKAT4e3PuOS4N5z3efn8equ7fVWl60mVKcVbNC9Z1U0A+Iz55+VG7ewWWjZ88NZr7ljnFURjmEoiv89VVi0qS23Vxdn2eYAXCPVmSp/98Z3h3EVgPf2ajK29VJzua7N8BGMFVXmr4eq6tqPambztIsp5J20zcd5HkCeJSg9Nil6gWOBVf07CJNkRFeGvCNTXCtsu4Z1H5eChZmKKnHX591C+YZ+a+ssiODhMxyueK99751AHS33AcUJde1168hJWRxabQRMH9lKw/R1B6q6wz4S1SFMxEyoa/1Ph1r1l7jABFjVortJn6EgPb3uTWPdoLHdaHdsnou1iVfLu9DN3JRyUXHZwMY2PoyWPPK6b9VzRbHdxTDFxSztj/fd9u+PWRVe61Bv1b6s+1T3fzbCCzSwTh+i33ONtIDv9GvRe9nbxiH6rUmFvb7AWTiemk= rdibnfc@vbox
```

Task 4: Verify ssh remote connection

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

```
[rdjbnfc@vbox ~]$ ssh rdjbnfc@192.168.56.116
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Fri Sep 5 16:17:39 2025 from 192.168.56.111
```

2. Show evidence that you are connected.

```
Last login: Fri Sep  5 16:17:39 2025 from 192
[rdjbnfc@vbox ~]$ whoami
hostname
cat /etc/redhat-release
rdjbnfc
vbox
CentOS Stream release 9
[rdjbnfc@vbox ~]$
```

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?
 - When deciding between Debian-based and Red Hat-based Linux distributions, it is important to take into account factors like system stability, the type of package management used, whether support is mainly community-driven or enterprise-level, the approach to security updates, overall ease of system administration, hardware support, and long-term maintenance options.
2. What are the main difference between Debian and Red Hat Linux distributions?
 - The key distinction is that **Debian** is community-driven, emphasizing stability and free software while using the **APT (dpkg)** package manager. In contrast, **Red Hat (RHEL)** targets enterprise needs, providing professional support, certifications, and utilizing the **YUM/DNF (RPM)** package manager. Debian is commonly chosen for its flexibility and suitability for servers, whereas Red Hat is preferred in enterprise settings that demand official support and certifications.