Name: Alexis Neil D. Deniega	**Date Performed:** August 15th, 2025
Course/Section: CPE 212 - CPE31S4	**Date Submitted:** August 15th, 2025
Instructor: Engr. Robin Valenzuela	**Semester and SY:** SY 2025-26 1st Sem

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.

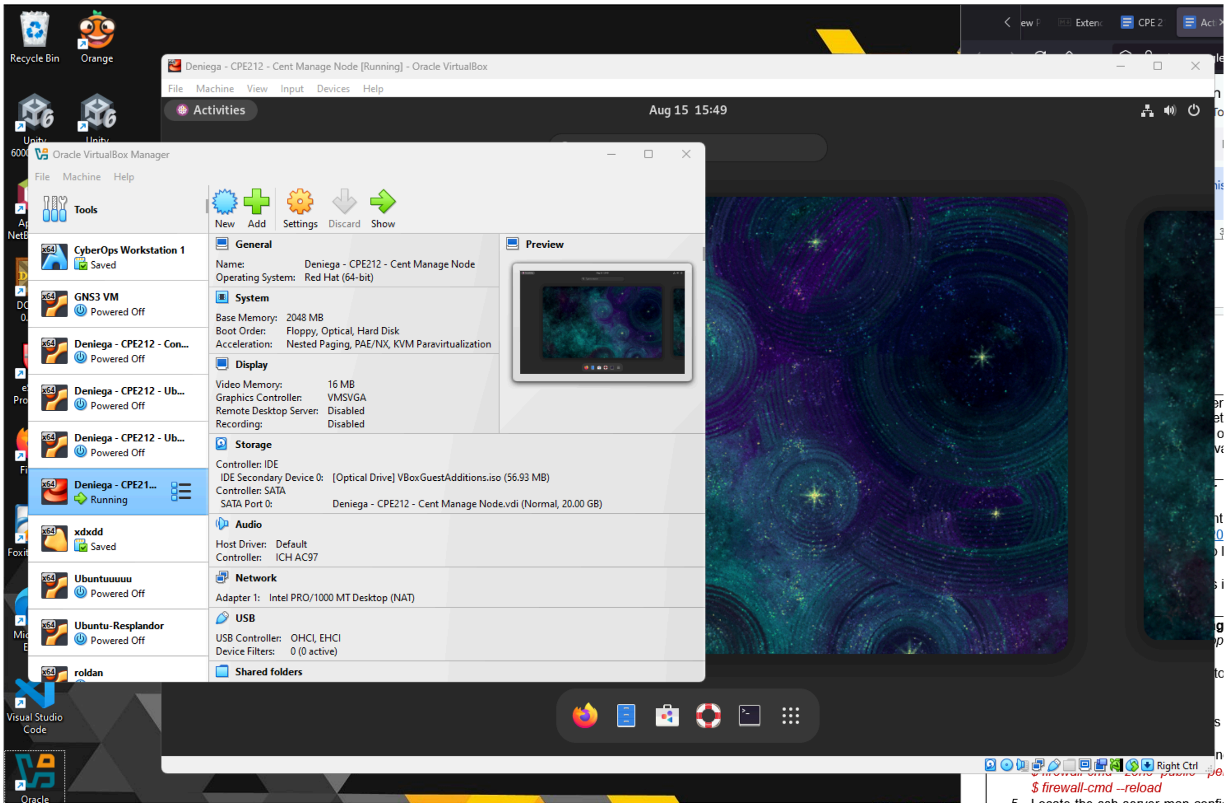
CentOS uses the RPM package format and YUM/DNF as the package manager.

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.



CyberOps Workstation 1 — Saved
GNS3 VM — Powered Off
Deniega - CPE212 - Con... — Powered Off
Deniega - CPE212 - Ub... — Powered Off
Deniega - CPE212 - Ub... — Powered Off
Deniega - CPE21... — Running
xdxdd — Saved
Ubuntuuuuu — Powered Off
Ubuntu-Resplandor — Powered Off
roldan

General
Name: Deniega - CPE212 - Cent Manage Node
Operating System: Red Hat (64-bit)

System
Base Memory: 2048 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: Nested Paging, PAE/NX, KVM Paravirtualization

Display
Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage
Controller: IDE
IDE Secondary Device 0: [Optical Drive] VBoxGuestAdditions.iso (56.93 MB)
Controller: SATA
SATA Port 0: Deniega - CPE212 - Cent Manage Node.vdi (Normal, 20.00 GB)

Audio
Host Driver: Default
Controller: ICH AC97

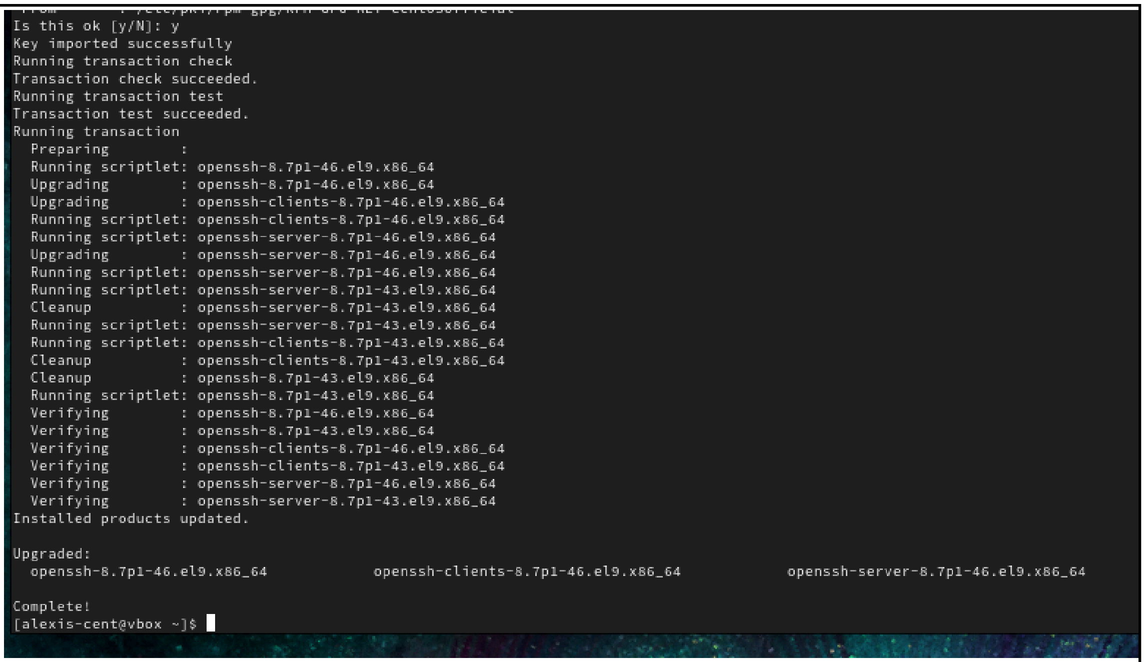
Network
Adapter 1: Intel PRO/1000 MT Desktop (NAT)

USB
USB Controller: OHCI, EHCI
Device Filters: 0 (0 active)

Shared folders

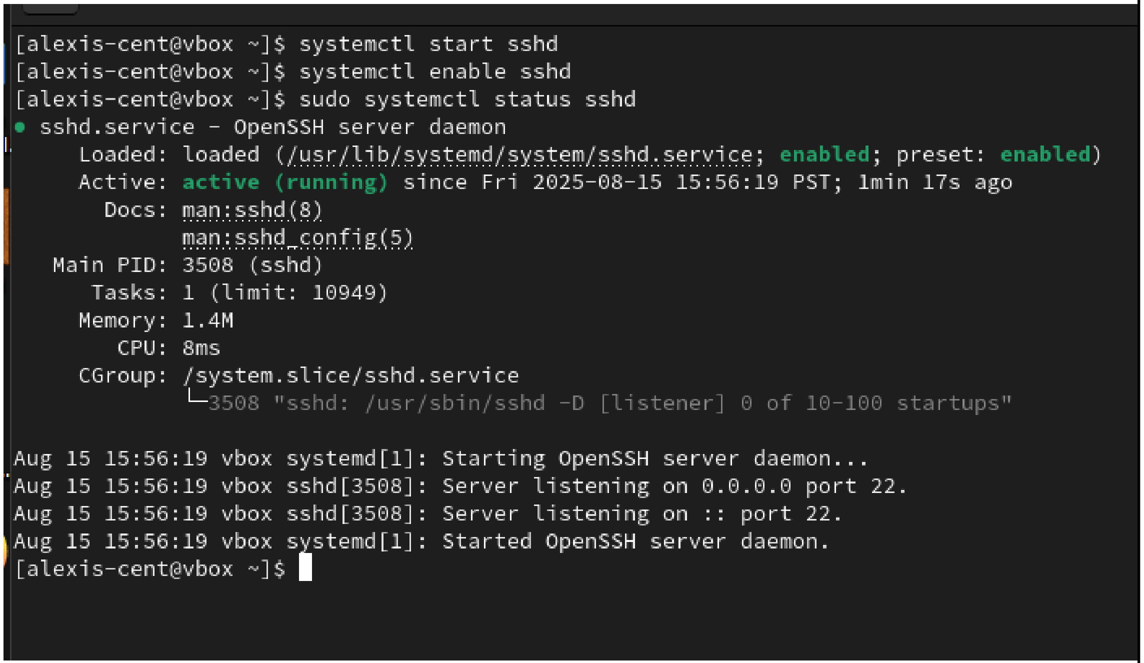
Task 2: Install the SSH server package *openssh*

1. Install the ssh server package *openssh* by using the *dnf* command:
   $ dnf install openssh-server



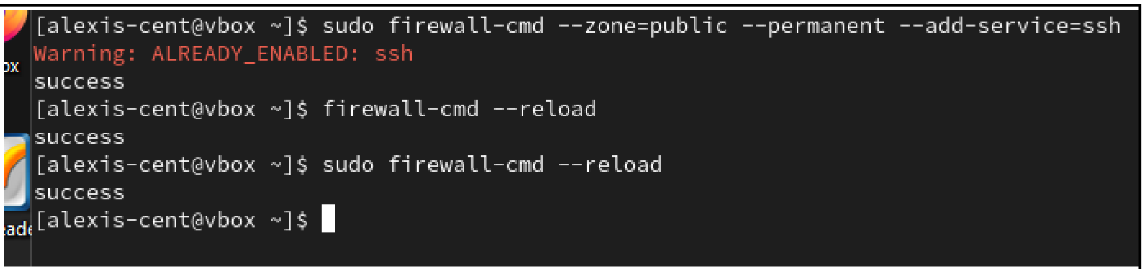
```
  From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-centosofficial
Is this ok [y/N]: y
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :
  Running scriptlet: openssh-8.7p1-46.el9.x86_64
  Upgrading        : openssh-8.7p1-46.el9.x86_64
  Upgrading        : openssh-clients-8.7p1-46.el9.x86_64
  Running scriptlet: openssh-clients-8.7p1-46.el9.x86_64
  Running scriptlet: openssh-server-8.7p1-46.el9.x86_64
  Upgrading        : openssh-server-8.7p1-46.el9.x86_64
  Running scriptlet: openssh-server-8.7p1-46.el9.x86_64
  Running scriptlet: openssh-server-8.7p1-43.el9.x86_64
  Cleanup          : openssh-server-8.7p1-43.el9.x86_64
  Running scriptlet: openssh-server-8.7p1-43.el9.x86_64
  Running scriptlet: openssh-clients-8.7p1-43.el9.x86_64
  Cleanup          : openssh-clients-8.7p1-43.el9.x86_64
  Cleanup          : openssh-8.7p1-43.el9.x86_64
  Running scriptlet: openssh-8.7p1-43.el9.x86_64
  Verifying        : openssh-8.7p1-46.el9.x86_64
  Verifying        : openssh-8.7p1-43.el9.x86_64
  Verifying        : openssh-clients-8.7p1-46.el9.x86_64
  Verifying        : openssh-clients-8.7p1-43.el9.x86_64
  Verifying        : openssh-server-8.7p1-46.el9.x86_64
  Verifying        : openssh-server-8.7p1-43.el9.x86_64
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!
[alexis-cent@vbox ~]$ 
```

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



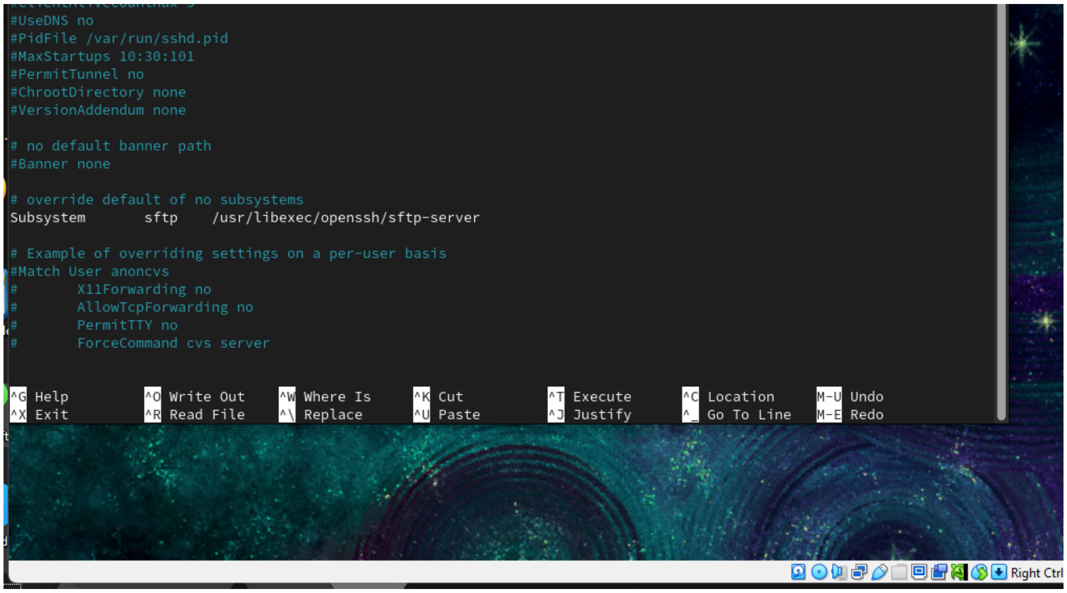
```
[alexis-cent@vbox ~]$ systemctl start sshd
[alexis-cent@vbox ~]$ systemctl enable sshd
[alexis-cent@vbox ~]$ sudo systemctl status sshd
● sshd.service - OpenSSH server daemon
     Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
     Active: active (running) since Fri 2025-08-15 15:56:19 PST; 1min 17s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 3508 (sshd)
      Tasks: 1 (limit: 10949)
     Memory: 1.4M
        CPU: 8ms
     CGroup: /system.slice/sshd.service
             └─3508 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 15 15:56:19 vbox systemd[1]: Starting OpenSSH server daemon...
Aug 15 15:56:19 vbox sshd[3508]: Server listening on 0.0.0.0 port 22.
Aug 15 15:56:19 vbox sshd[3508]: Server listening on :: port 22.
Aug 15 15:56:19 vbox systemd[1]: Started OpenSSH server daemon.
[alexis-cent@vbox ~]$ 
```

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

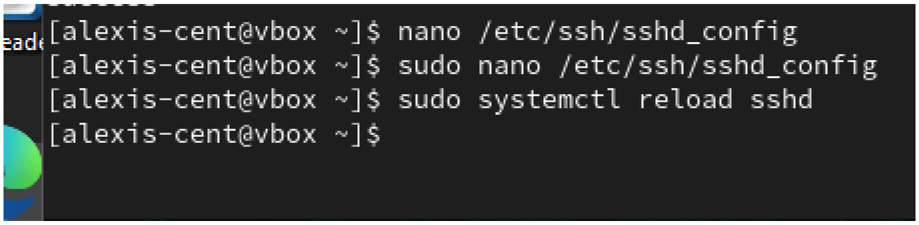


```
[alexis-cent@vbox ~]$ sudo firewall-cmd --zone=public --permanent --add-service=ssh
Warning: ALREADY_ENABLED: ssh
success
[alexis-cent@vbox ~]$ firewall-cmd --reload
success
[alexis-cent@vbox ~]$ sudo firewall-cmd --reload
success
[alexis-cent@vbox ~]$
```

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



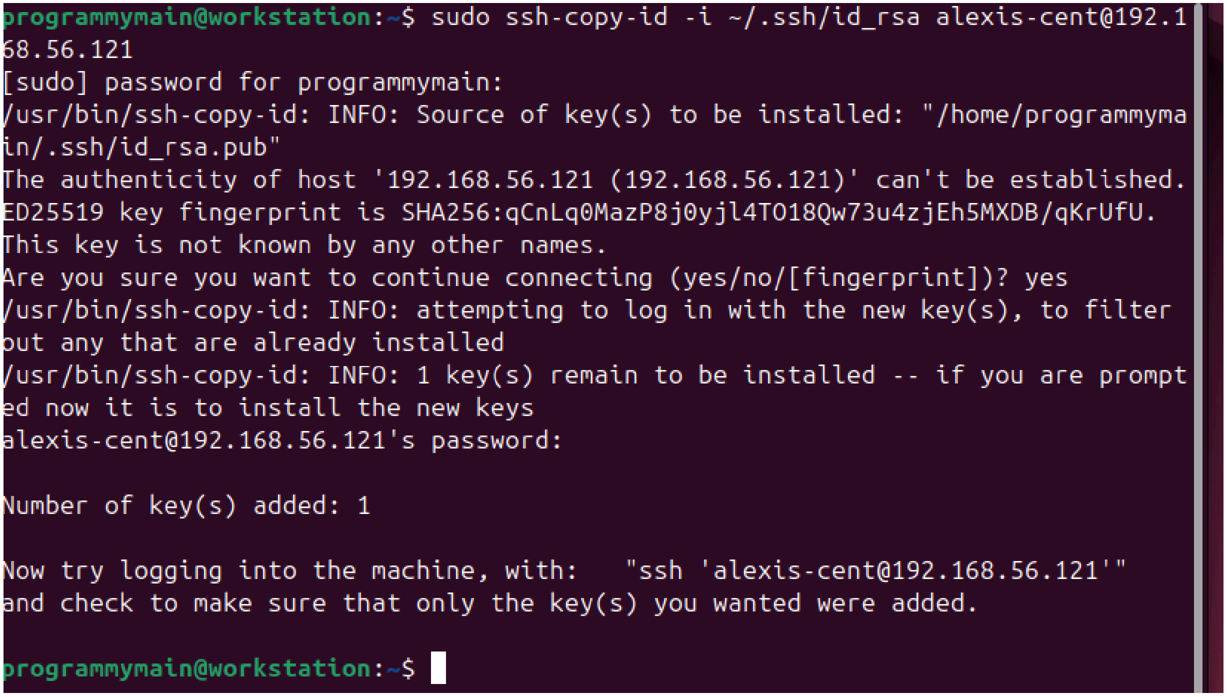
```
#UseDNS no
#PidFile /var/run/sshd.pid
#MaxStartups 10:30:101
#PermitTunnel no
#ChrootDirectory none
#VersionAddendum none

# no default banner path
#Banner none

# override default of no subsystems
Subsystem       sftp    /usr/libexec/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#       X11Forwarding no
#       AllowTcpForwarding no
#       PermitTTY no
#       ForceCommand cvs server


^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^_ Go To Line  M-E Redo
```



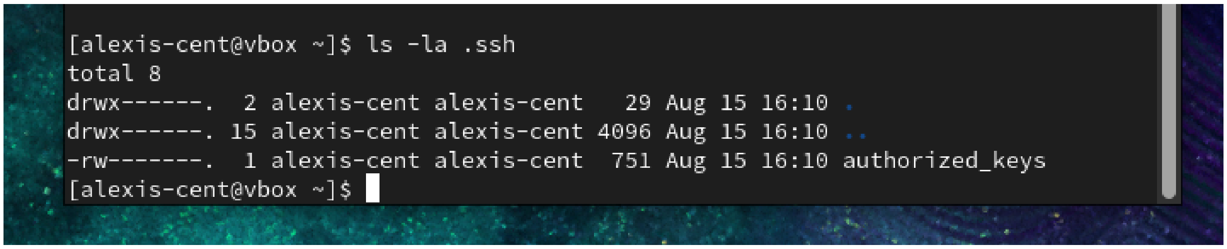
```
[alexis-cent@vbox ~]$ nano /etc/ssh/sshd_config
[alexis-cent@vbox ~]$ sudo nano /etc/ssh/sshd_config
[alexis-cent@vbox ~]$ sudo systemctl reload sshd
[alexis-cent@vbox ~]$
```

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*.



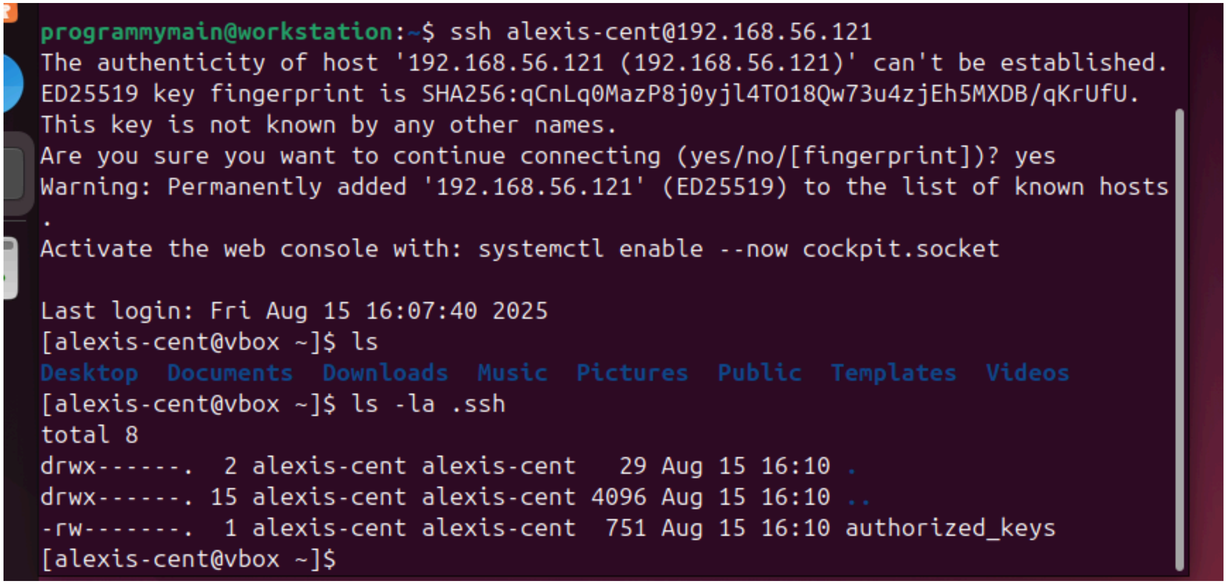
```
programmymain@workstation:~$ sudo ssh-copy-id -i ~/.ssh/id_rsa alexis-cent@192.1
68.56.121
[sudo] password for programmymain:
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/programmyma
in/.ssh/id_rsa.pub"
The authenticity of host '192.168.56.121 (192.168.56.121)' can't be established.
ED25519 key fingerprint is SHA256:qCnLq0MazP8j0yjl4TO18Qw73u4zjEh5MXDB/qKrUfU.
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
alexis-cent@192.168.56.121's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'alexis-cent@192.168.56.121'"
and check to make sure that only the key(s) you wanted were added.

programmymain@workstation:~$
```



```
[alexis-cent@vbox ~]$ ls -la .ssh
total 8
drwx------.  2 alexis-cent alexis-cent   29 Aug 15 16:10 .
drwx------. 15 alexis-cent alexis-cent 4096 Aug 15 16:10 ..
-rw-------.  1 alexis-cent alexis-cent  751 Aug 15 16:10 authorized_keys
[alexis-cent@vbox ~]$
```

Task 4: Verify ssh remote connection

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



```
programmymain@workstation:~$ ssh alexis-cent@192.168.56.121
The authenticity of host '192.168.56.121 (192.168.56.121)' can't be established.
ED25519 key fingerprint is SHA256:qCnLq0MazP8j0yjl4TO18Qw73u4zjEh5MXDB/qKrUfU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.121' (ED25519) to the list of known hosts
.
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Fri Aug 15 16:07:40 2025
[alexis-cent@vbox ~]$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
[alexis-cent@vbox ~]$ ls -la .ssh
total 8
drwx------.  2 alexis-cent alexis-cent   29 Aug 15 16:10 .
drwx------. 15 alexis-cent alexis-cent 4096 Aug 15 16:10 ..
-rw-------.  1 alexis-cent alexis-cent  751 Aug 15 16:10 authorized_keys
[alexis-cent@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?

It really depends on the use case of the end user. If someone wants to get the most convenience and the best long time support, then Debian systems are the best for that. If automation is key for the end user(s), then Red Hat distros are best.

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

The main difference between the distributions is how they are packaged. Debian uses Advanced Package Tool (apt (by Bruno Mars and Rose)) while Red Hat uses Yellowdog Updater (YUM), which have different methods of packaging. That greatly affects a few use cases, and screws up compatibility between the two distributions.