Name:Calderon, Ricardo B.	Date Performed:1/28/24
Course/Section:CPE31S1	Date Submitted:1/30/24
Instructor: Dr. Jonathan Taylar	Semester and SY: 2nd sem SY 23-24

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.

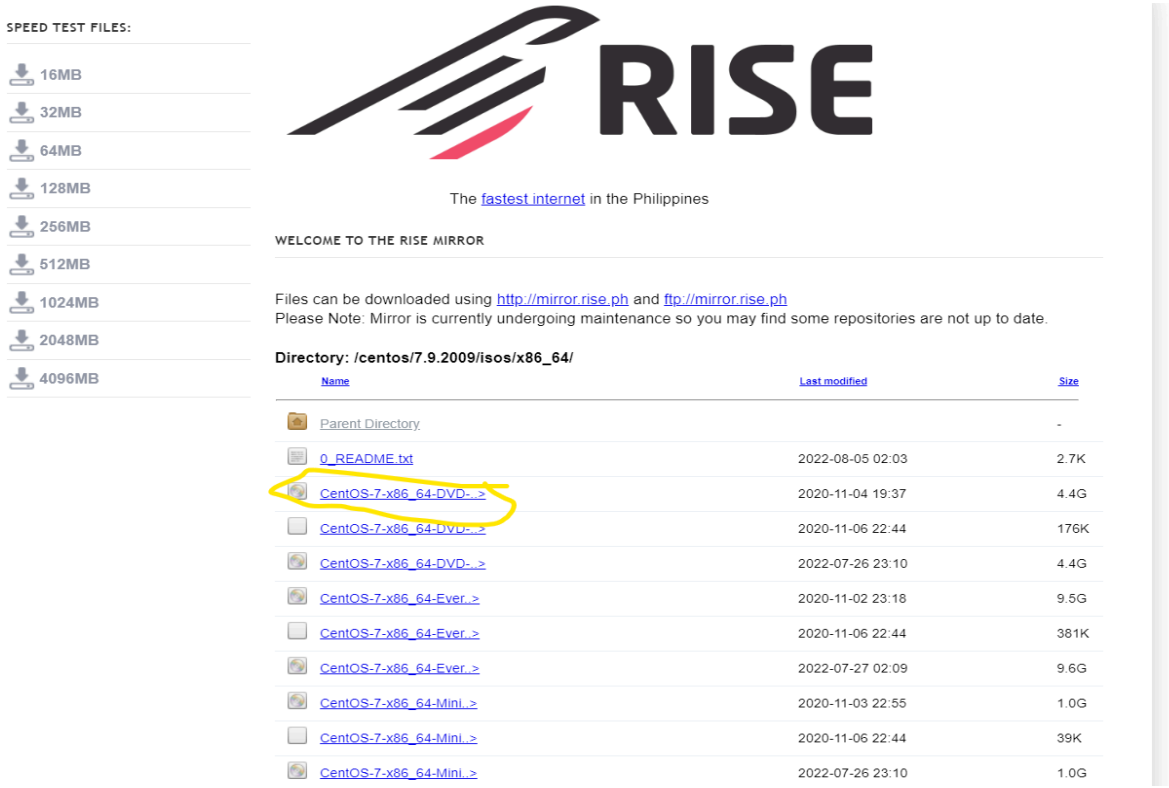
CentOS uses the RPM package format and YUM/DNF as the package manager.

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.

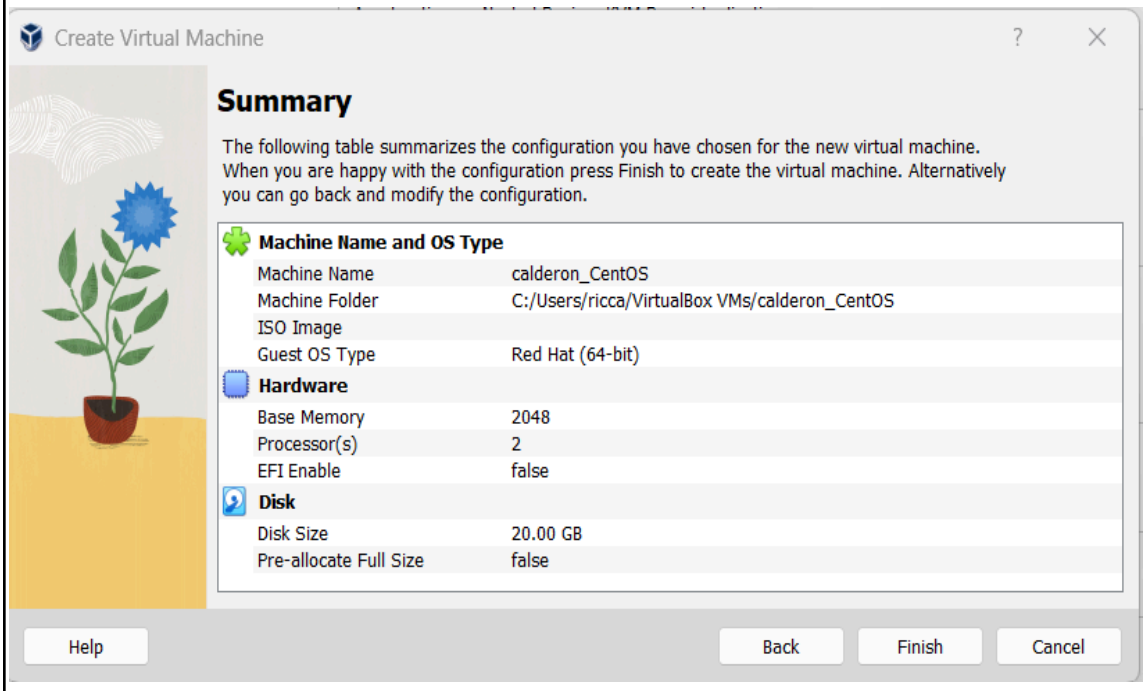
Task 1: Download the CentOS or RHEL-8 image (Create screenshots of the following)

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

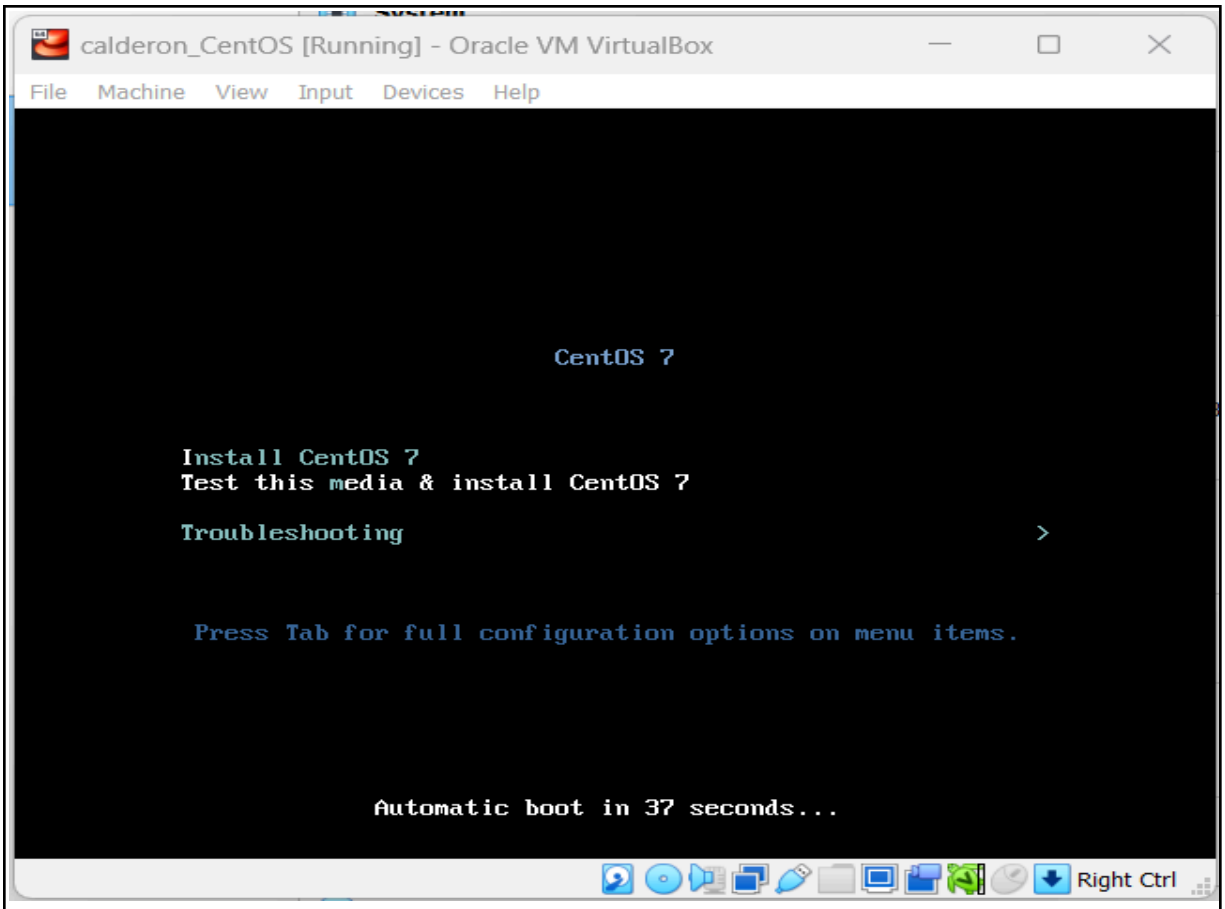


DVD Drive (D:) CentOS 7 x86_64
(14 items)

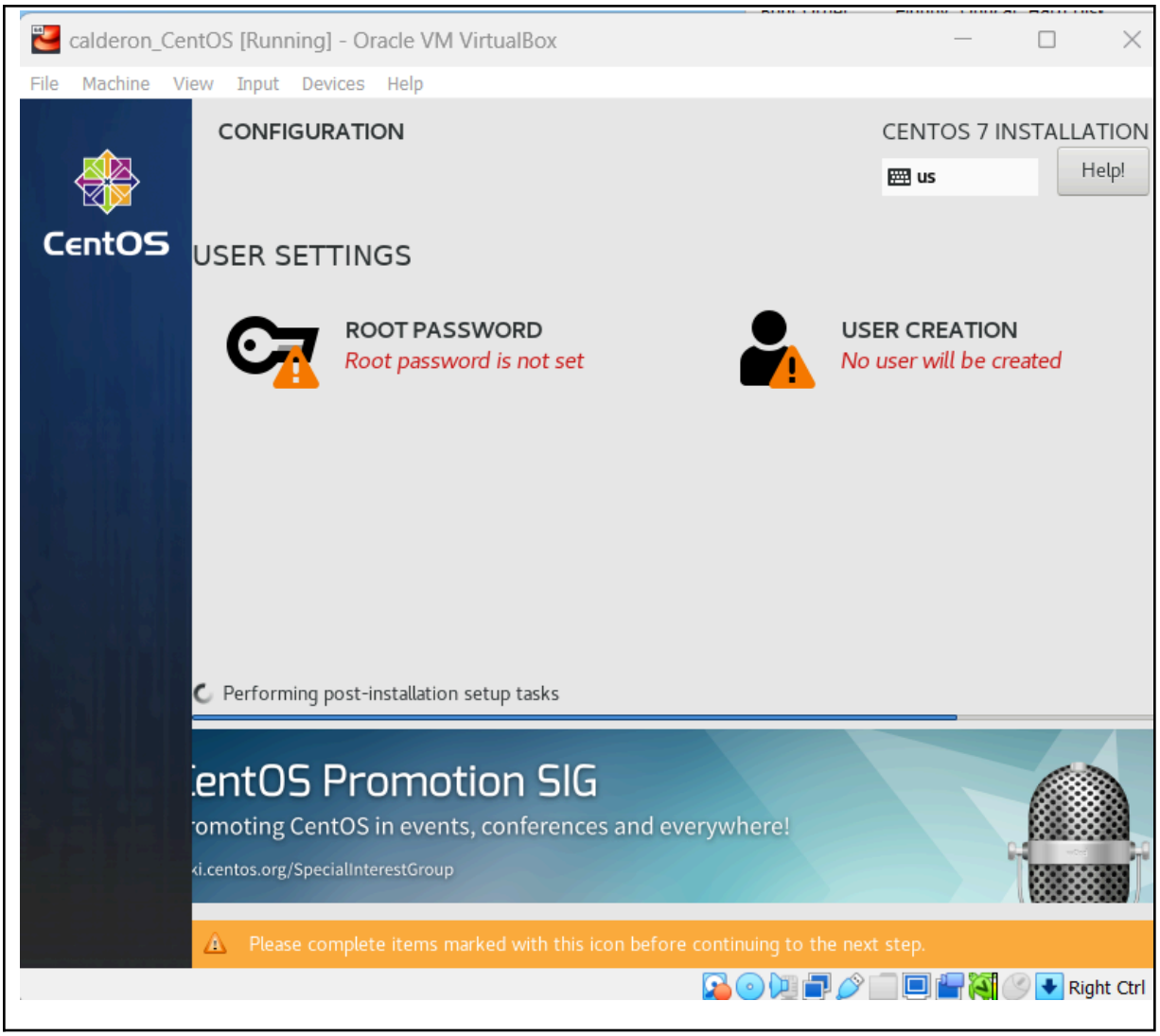
2. Create a VM machine with 2 Gb RAM and 20 Gb HD.

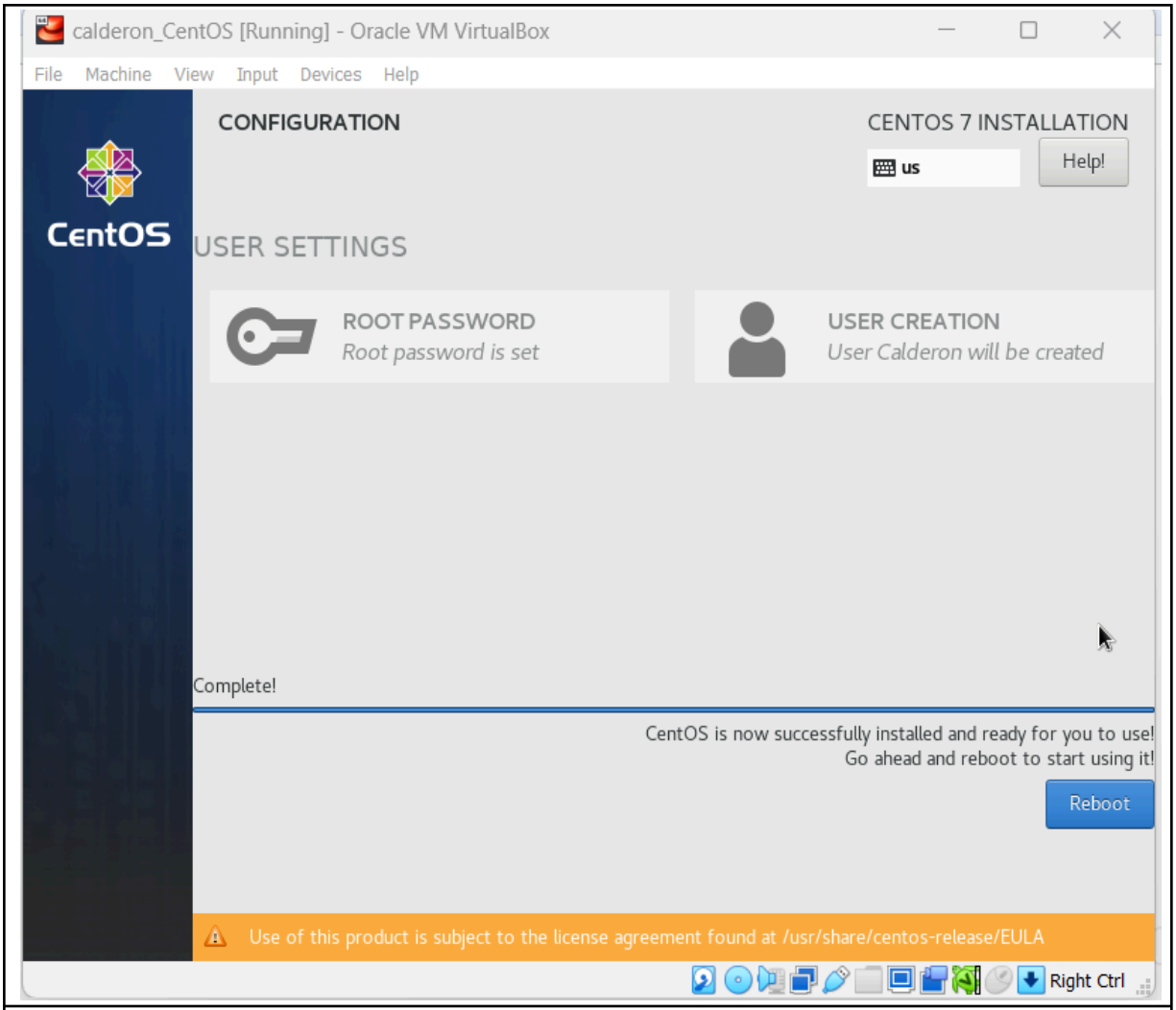


3. Install the downloaded image.



4. Show evidence that the OS was installed already.


calderon_CentOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
CONFIGURATION
CENTOS 7 INSTALLATION
us
Help!
CentOS
USER SETTINGS
ROOT PASSWORD
Root password is not set
USER CREATION
No user will be created
Performing post-installation setup tasks
entOS Promotion SIG
omoting CentOS in events, conferences and everywhere!
i.centos.org/SpecialInterestGroup
Please complete items marked with this icon before continuing to the next step.
Right Ctrl



Task 2: Install the SSH server package *openssh*

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

```
[calderon@localhost ~]$ sudo yum install openssh-server
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirror.rise.ph
 * extras: mirror.rise.ph
 * updates: mirror.rise.ph
Package openssh-server-7.4p1-23.el7_9.x86_64 already installed and latest version
Nothing to do
```

2. Start the *sshd* daemon and set to start after reboot:
   $ systemctl start sshd
   $ systemctl enable sshd

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

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

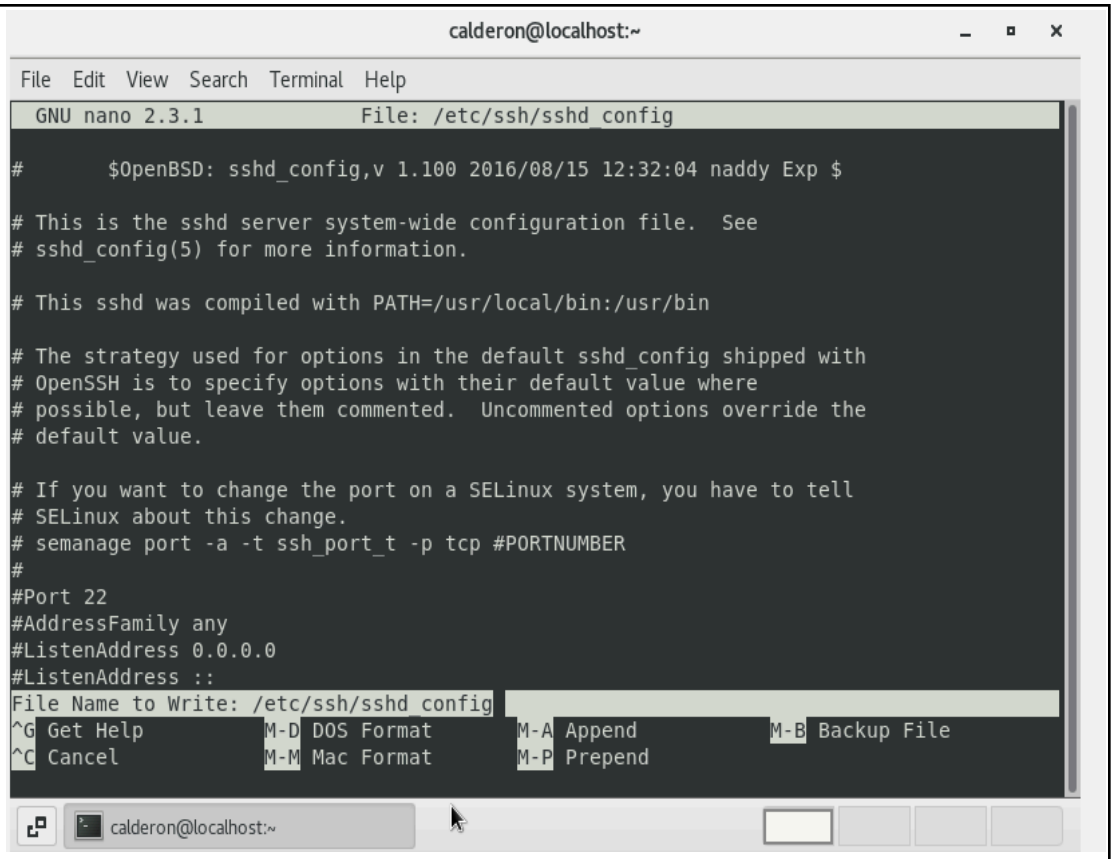
```
[calderon@localhost ~]$ sudo systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enable
d)
   Active: active (running) since Tue 2024-01-30 19:00:25 PST; 18min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 1088 (sshd)
   CGroup: /system.slice/sshd.service
           └─1088 /usr/sbin/sshd -D

Jan 30 19:00:24 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Jan 30 19:00:25 localhost.localdomain sshd[1088]: Server listening on 0.0.0.0 port 22.
Jan 30 19:00:25 localhost.localdomain sshd[1088]: Server listening on :: port 22.
Jan 30 19:00:25 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
Hint: Some lines were ellipsized, use -l to show in full.
```

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

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

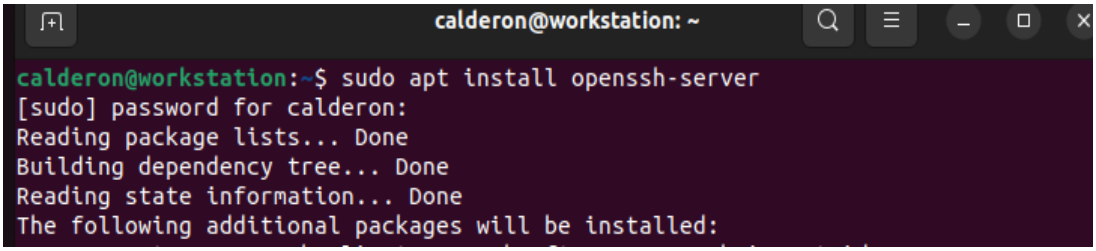
5. Locate the ssh server man config file */etc/ssh/sshd_config* and perform custom configuration. Every time you make any change to the */etc/ssh/sshd-config* configuration file reload the *sshd* service to apply changes:
   $ systemctl reload sshd

```
[calderon@localhost ~]$ sudo nano /etc/ssh/sshd_config
[calderon@localhost ~]$
```



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.

```
calderon@workstation:~$ ssh-copy-id -i ~/.ssh/id_rsa calderon@localhost
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/calderon/.s
sh/id_rsa.pub"
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ED25519 key fingerprint is SHA256:LcQfzcxLK6V2rjXX+zqNhfRdlWaiOTOSHlQaKThCKVY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the 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
calderon@localhost's password:

Number of key(s) added: 1

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

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

```
calderon@workstation:~$ ssh calderon@localhost
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.5.0-15-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
```

Task 4: Verify ssh remote connection

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

```
calderon@workstation:~$ ssh-copy-id -i ~/.ssh/id_rsa calderon@localhost
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/calderon/.s
sh/id_rsa.pub"
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ED25519 key fingerprint is SHA256:LcQfzcxLK6V2rjXX+zqNhfRdlWaiOTOSHlQaKThCKVY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the 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
calderon@localhost's password:

Number of key(s) added: 1

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

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 best distribution for me between Debian and CentOS depends on personal preferences and intended use case. Consider architecture support, package management, release cycle and support, user base and community, and purpose when we choose.

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

One of the main differences between Debian and Red Hat-based distributions is their release cycle. Debian follows a rolling release model with no set release schedule, while RedHat-based distributions (such as CentOS) have less frequent major releases and more frequent minor updates.

Conclusion

For me, CentOS and Debian have similarities and differences in architecture and package management. Choosing the best option depends on personal preferences and needs. Both have full-featured package management and support for various architectures, but some differences exist. All in all both are great options for Linux users.