Name: Bernardo, Christian Emmanuel	Date Performed: 30/01/2024
Course/Section: CPE 232-CPE31S1	Date Submitted: 30/01/2024
Instructor: Dr. Jonathan Taylar	Semester and SY: 2nd 2023-2024

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.

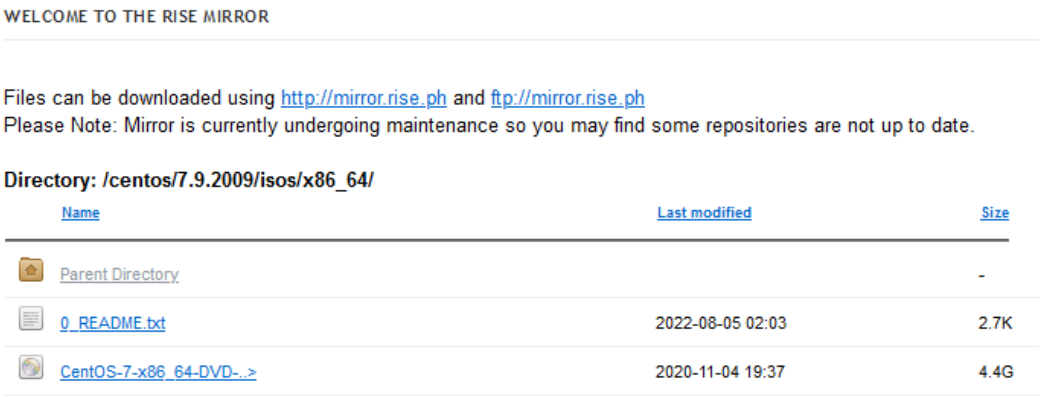
CentOS uses the RPM package format and YUM/DNF as the package manager.

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/

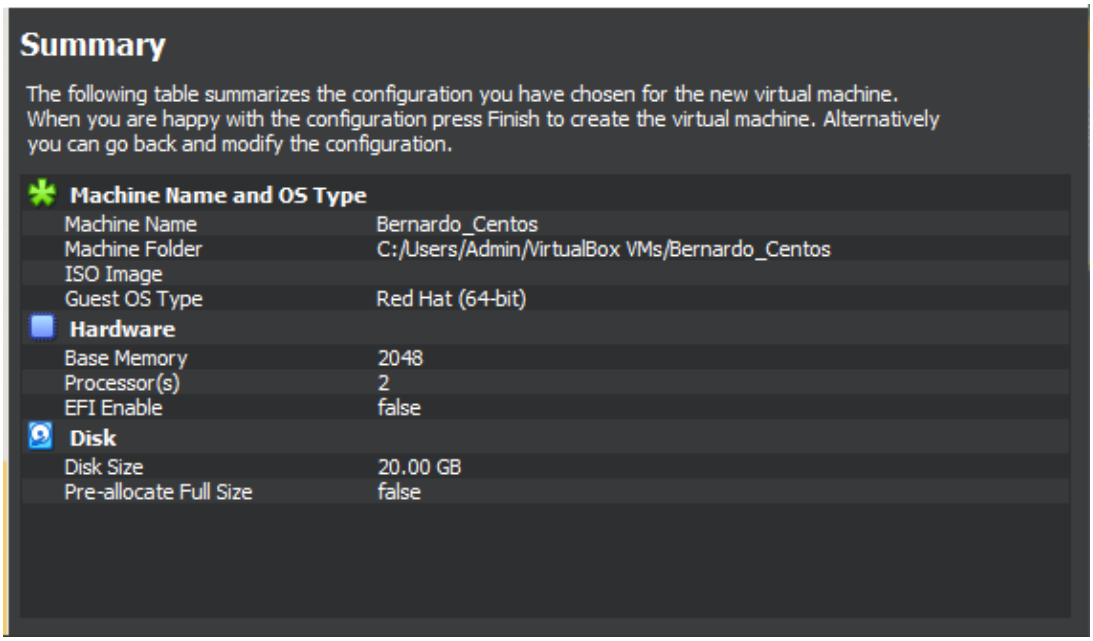

WELCOME TO THE RISE MIRROR

Files can be downloaded using http://mirror.rise.ph and ftp://mirror.rise.ph
Please Note: Mirror is currently undergoing maintenance so you may find some repositories are not up to date.

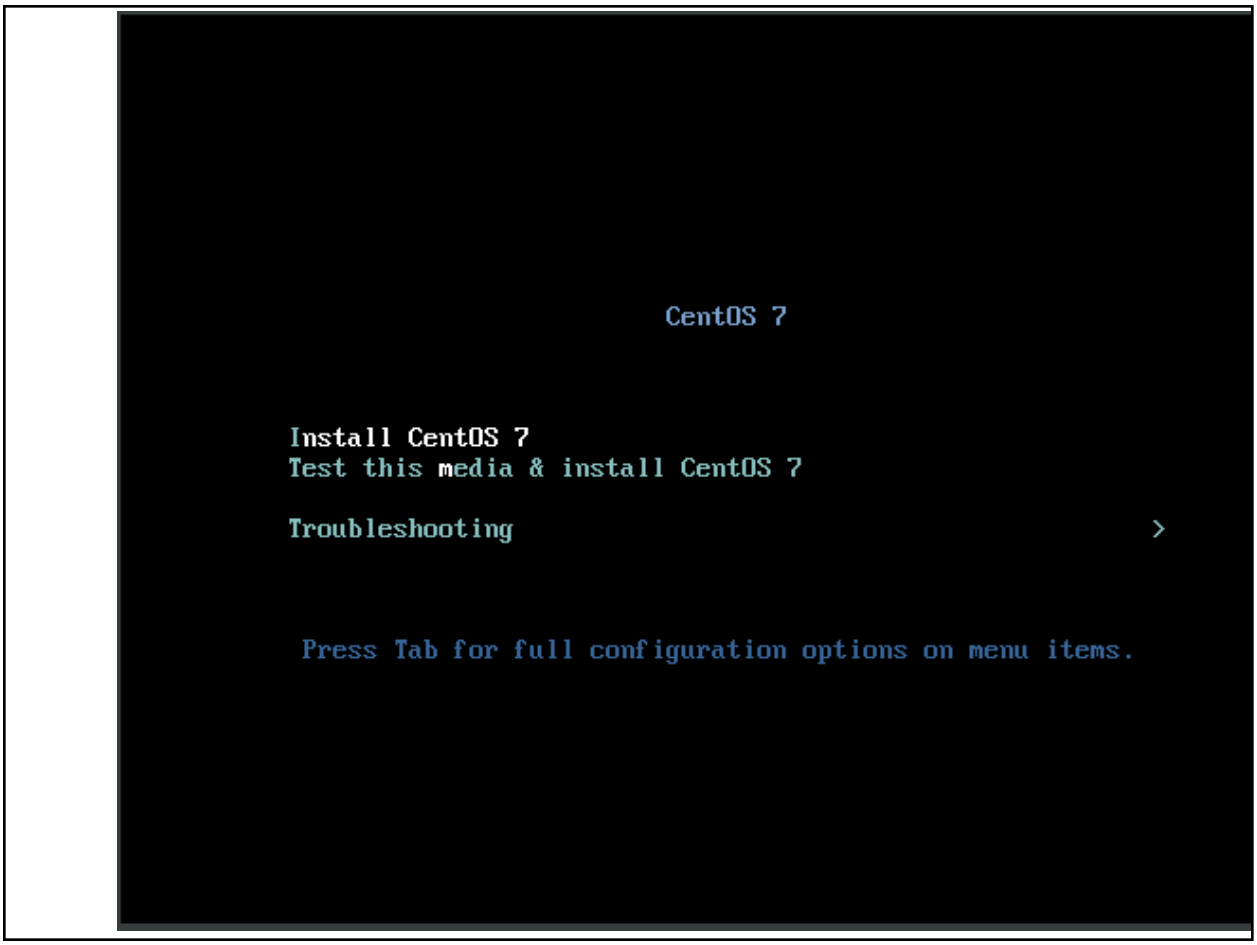
Directory: /centos/7.9.2009/isos/x86_64/

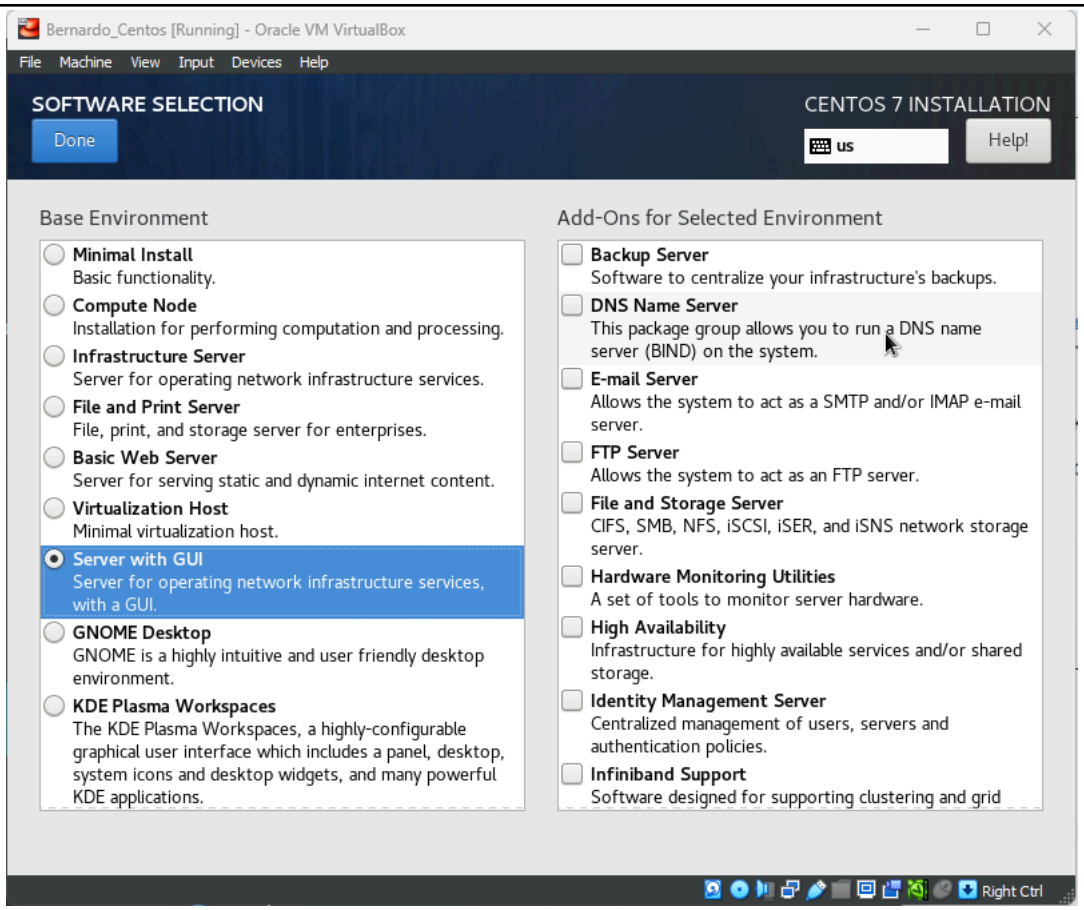
Name	Last modified	Size
Parent Directory		-
0_README.txt	2022-08-05 02:03	2.7K
CentOS-7-x86_64-DVD-..>	2020-11-04 19:37	4.4G

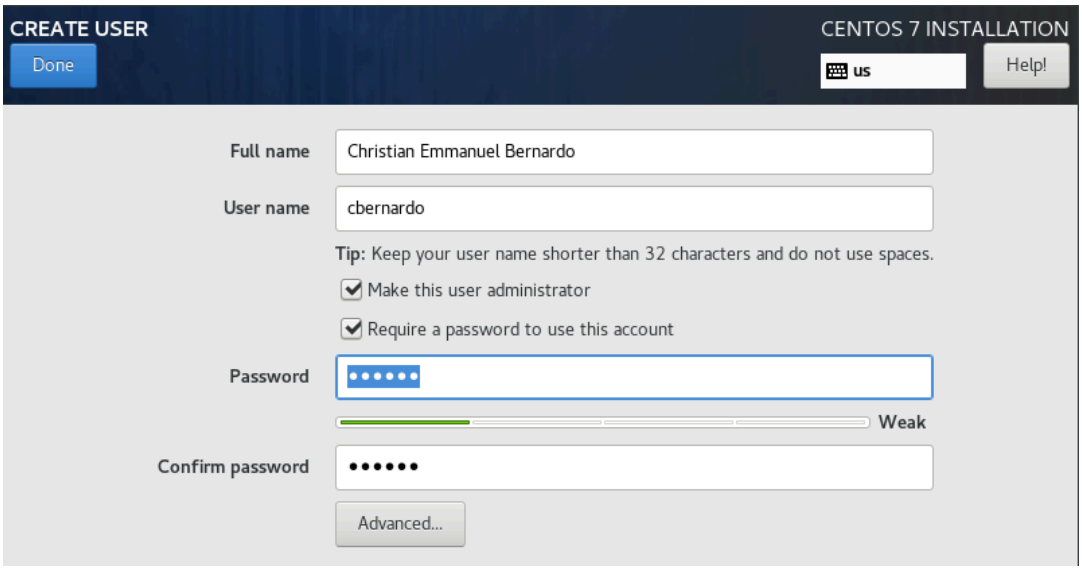
2. Create a VM machine with 2 Gb RAM and 20 Gb HD.



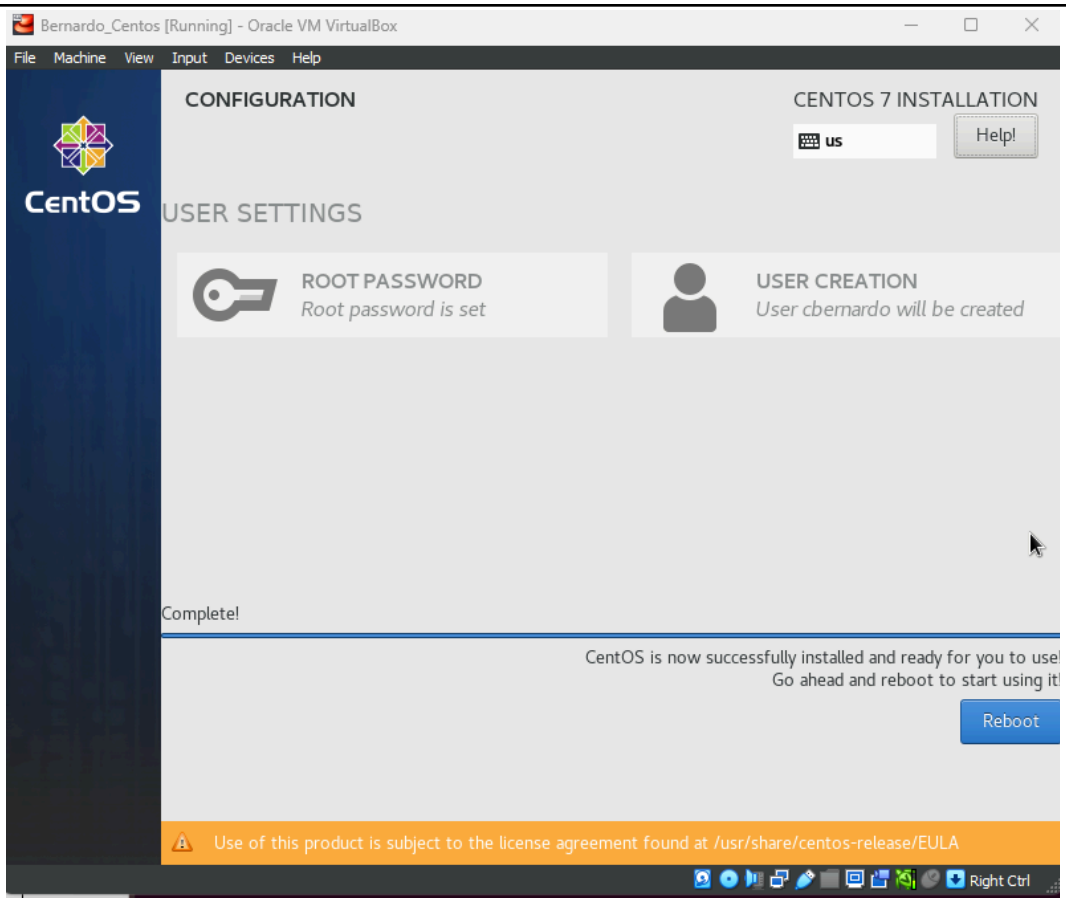
3. Install the downloaded image.


CentOS 7
Install CentOS 7
Test this media & install CentOS 7
Troubleshooting
Press Tab for full configuration options on menu items.





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

```
[cbernardo@localhost ~]$ sudo dhclient
[cbernardo@localhost ~]$ sudo yum update
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirror-hk.koddos.net
 * extras: mirror-hk.koddos.net
 * updates: mirror-hk.koddos.net
base                                                     | 3.6 kB  00:00:00
extras                                                   | 2.9 kB  00:00:00
updates                                                  | 2.9 kB  00:00:00
(1/4): base/7/x86_64/group_gz                            | 153 kB  00:00:02
(2/4): extras/7/x86_64/primary_db                        | 250 kB  00:00:02
(3/4): base/7/x86_64/primary_db                          | 6.1 MB  00:00:05
(4/4): updates/7/x86_64/primary_db                       |  25 MB  00:00:15
Resolving Dependencies
--> Running transaction check
---> Package NetworkManager.x86_64 1:1.18.8-1.el7 will be updated
---> Package NetworkManager.x86_64 1:1.18.8-2.el7_9 will be an update
---> Package NetworkManager-adsl.x86_64 1:1.18.8-1.el7 will be updated
---> Package NetworkManager-adsl.x86_64 1:1.18.8-2.el7_9 will be an update
---> Package NetworkManager-glib.x86_64 1:1.18.8-1.el7 will be updated
---> Package NetworkManager-glib.x86_64 1:1.18.8-2.el7_9 will be an update
---> Package NetworkManager-libnm.x86_64 1:1.18.8-1.el7 will be updated
```

```
[cbernardo@localhost ~]$ sudo yum 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 cbernardo:
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
Could not retrieve mirrorlist http://mirrorlist.centos.org/?release=7&arch=x86_64&repo=
os&infra=stock error was
14: curl#6 - "Could not resolve host: mirrorlist.centos.org; Unknown error"


 One of the configured repositories failed (Unknown),
 and yum doesn't have enough cached data to continue. At this point the only
 safe thing yum can do is fail. There are a few ways to work "fix" this:

     1. Contact the upstream for the repository and get them to fix the problem.

     2. Reconfigure the baseurl/etc. for the repository, to point to a working
        upstream. This is most often useful if you are using a newer
```

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

```
                                                    cbernardo@localhost:~
File  Edit  View  Search  Terminal  Help
[cbernardo@localhost ~]$ systemctl start sshd
[cbernardo@localhost ~]$ systemctl enable sshd
[cbernardo@localhost ~]$
```

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

```
[cbernardo@localhost ~]$ 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 19:23:49 PST; 21min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 1112 (sshd)
   CGroup: /system.slice/sshd.service
           └─1112 /usr/sbin/sshd -D

Jan 30 19:23:48 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Jan 30 19:23:49 localhost.localdomain sshd[1112]: Server listening on 0.0.0.0 port 22.
Jan 30 19:23:49 localhost.localdomain sshd[1112]: Server listening on :: port 22.
Jan 30 19:23:49 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

```
cbernardo@localhost:~
File  Edit  View  Search  Terminal  Help
[cbernardo@localhost ~]$ firewall-cmd --zone=public --permanent --add-service=ssh
Warning: ALREADY_ENABLED: ssh
success
[cbernardo@localhost ~]$ firewall-cmd --reload
success
[cbernardo@localhost ~]$
```

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

```
cbernardo@localhost:~
File  Edit  View  Search  Terminal  Help
[cbernardo@localhost ~]$ sudo nano /etc/ssh/sshd-config
[sudo] password for cbernardo:
[cbernardo@localhost ~]$
```

Task 3: Copy the Public Key to CentOS

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

```
christian@christian-VirtualBox:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeysphere ssh-askpass
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
0 upgraded, 4 newly installed, 0 to remove and 19 not upgraded.
Need to get 752 kB of archives.
After this operation, 6,050 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-sftp-
server amd64 1:8.9p1-3ubuntu0.6 [38.7 kB]
Get:2 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-serve
r amd64 1:8.9p1-3ubuntu0.6 [435 kB]
Get:3 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ncurses-term
all 6.3-2ubuntu0.1 [267 kB]
Get:4 http://ph.archive.ubuntu.com/ubuntu jammy/main amd64 ssh-import-id all 5.1
1-0ubuntu1 [10.1 kB]
```

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

```
christian@workstation:~$ ssh-copy-id -i ~/.ssh/id_rsa christian@localhost
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/christian/.
ssh/id_rsa.pub"
/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
christian@localhost's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'christian@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*.

```
christian@workstation:~$ ssh christian@localhost
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.5.0-14-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

22 updates can be applied immediately.
11 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
```

Task 4: Verify ssh remote connection

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

```
christian@workstation:~$ ssh-copy-id -i ~/.ssh/id_rsa christian@localhost
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/christian/.
ssh/id_rsa.pub"
/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
christian@localhost's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'christian@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?

Debian for community-driven development and stability, for Red-hat it's for people that want certifications, latest features, and enterprise-level support.

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

   Debian is often favored for its stability, community-driven development, and commitment to free software, while Red Hat-based distributions are widely used in enterprise environments focused on security, predictability, and commercial support.

Conclusion:

In this activity I learned how to add a server with a different OS to another. I also learned that the centos OS is a debian OS that is easy and free to use. I don’t know Red Hat but I think it would be a good OS to use too. I now understand how to make sure that the server passwords are kept very safe and use it privately so that other people can’t use it.