

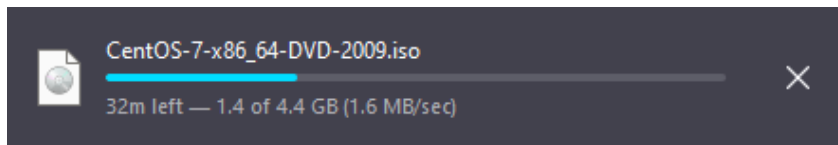
Name: Pacinos, Angela Monique A.	Date Performed: 08-29-23
Course/Section: CPE232 - CPE31S4	Date Submitted: 09-04-23
Instructor: Dr. Jonathan V. Tylar	Semester and SY: 1st Sem '23 - '24
Activity 3: Install SSH server on CentOS or RHEL 8	
1. Objectives: 1.1 Install Community Enterprise OS or Red Hat Linux 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.

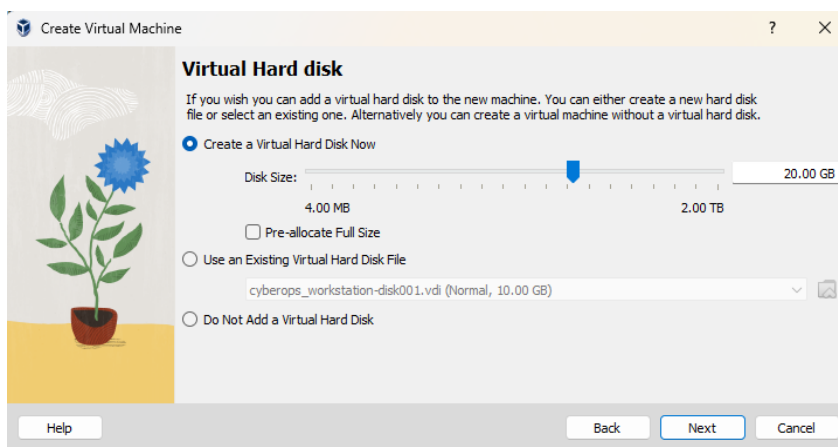
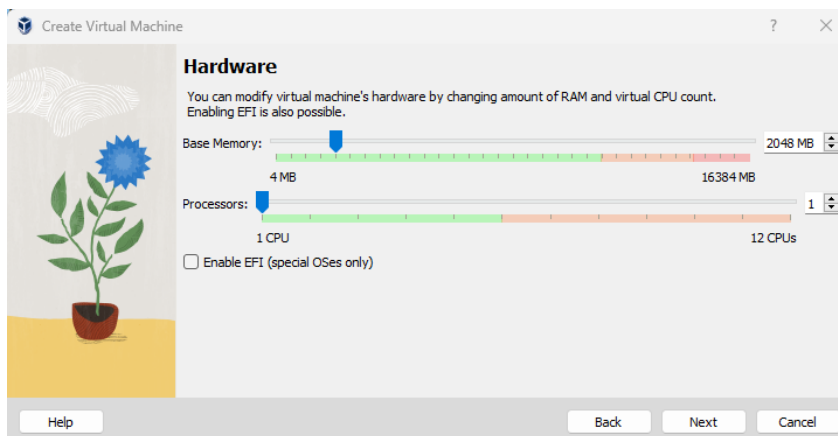
Task 1: Download the CentOS or RHEL-8 image

1. Download the image of the CentOS here:

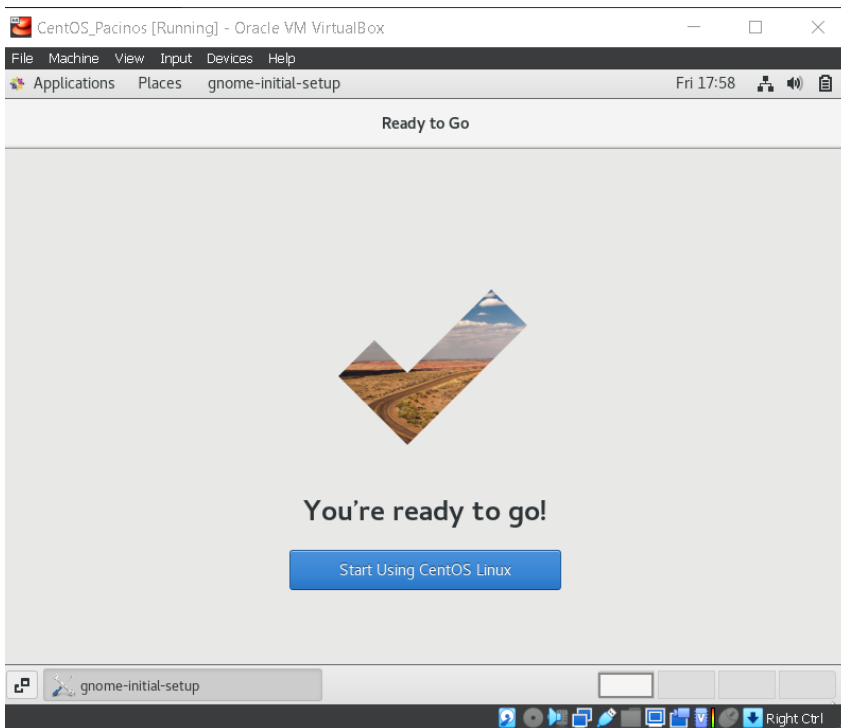
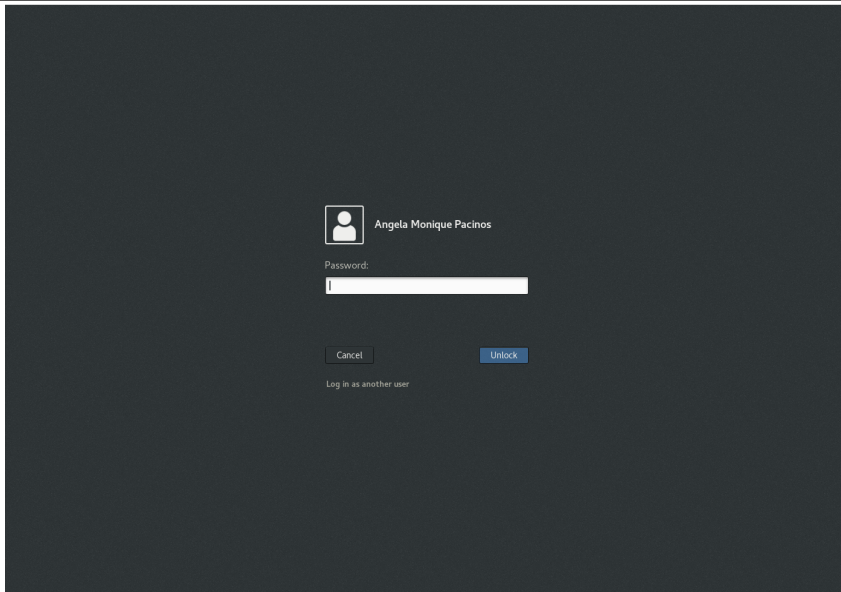
http://mirror.rise.ph/centos/7.9.2009/isos/x86_64/



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:

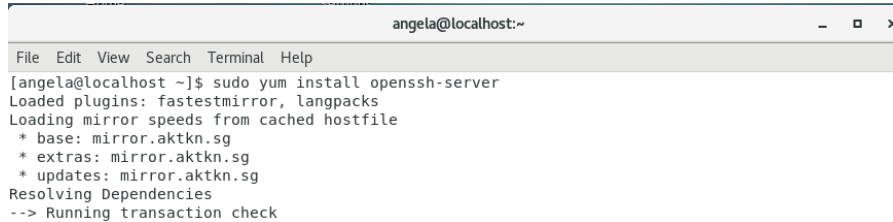
```
$ dnf install openssh-server
```

```
sudo yum install openssh-server
```

```
Updated:
  openssh-server.x86_64 0:7.4p1-23.el7_9

Dependency Updated:
  openssh.x86_64 0:7.4p1-23.el7_9    openssh-clients.x86_64 0:7.4p1-23.el7_9

Complete!
```

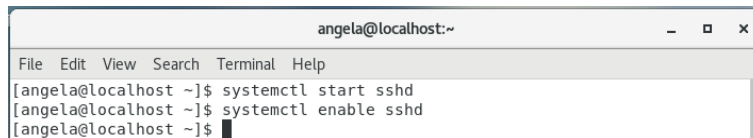


```
angela@localhost:~
File Edit View Search Terminal Help
[angela@localhost ~]$ sudo yum install openssh-server
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirror.aktkn.sg
 * extras: mirror.aktkn.sg
 * updates: mirror.aktkn.sg
Resolving Dependencies
--> Running transaction check
```

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

```
$ systemctl start sshd
```

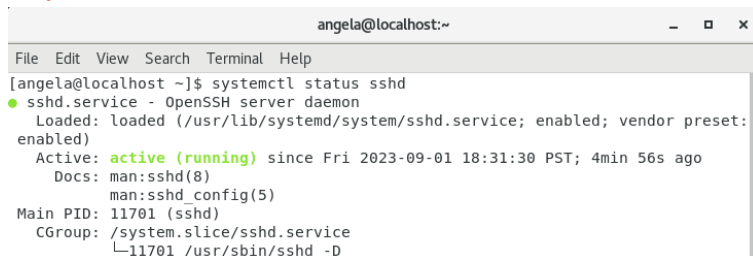
```
$ systemctl enable sshd
```



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

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

```
$ systemctl status sshd
```

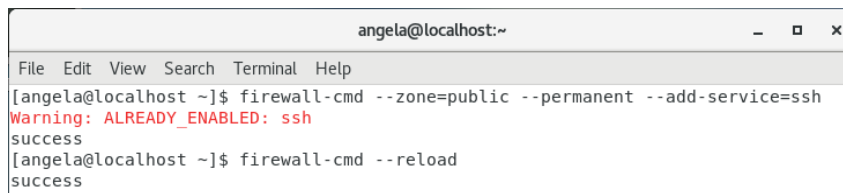


```
angela@localhost:~
File Edit View Search Terminal Help
[angela@localhost ~]$ systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset:
   enabled)
   Active: active (running) since Fri 2023-09-01 18:31:30 PST; 4min 56s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Main PID: 11701 (sshd)
    CGroup: /system.slice/sshd.service
            └─11701 /usr/sbin/sshd -D
```

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

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

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



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

Task 3: Copy the Public Key to CentOS

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

```
angela@localhost:~  
File Edit View Search Terminal Help  
[angela@localhost ~]$ ps aux | grep sshd  
root      11701  0.0  0.2 112900  4304 ?        Ss   18:31   0:00 /usr/sbin/sshd -D  
root      12320  0.7  0.2 156776  5584 ?        Ss   18:43   0:00 sshd: angela [priv]  
angela    12324  0.0  0.1 156776  2440 ?        S    18:44   0:00 sshd: angela@pts/1  
angela    12379  0.0  0.0 112808   960 pts/1    S+   18:44   0:00 grep --color=auto sshd
```

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

```
angela@localhost:~  
File Edit View Search Terminal Help  
[angela@localhost ~]$ ssh-keygen  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/angela/.ssh/id_rsa):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/angela/.ssh/id_rsa.  
Your public key has been saved in /home/angela/.ssh/id_rsa.pub.  
The key fingerprint is:  
SHA256:p6V6cUC8VGHika1gbcP40x5JVLRLZDgvak4wGtYmzk0 angela@localhost.localdomain  
The key's randomart image is:  
+---[RSA 2048]-----+  
|  +  =B0. |  
|  + * %+ . |  
|  o + B +. |  
|  + E = +. |  
|  + B = +Soo |  
|  + . O .=. |  
|  = ooo |  
|  o.. |  
|  .. |  
+-----[SHA256]-----+
```

```
angela@localhost:~  
File Edit View Search Terminal Help  
[angela@localhost ~]$ ssh-copy-id angela@192.168.56.103  
/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  
angela@192.168.56.103's password:  
  
Number of key(s) added: 1  
  
Now try logging into the machine, with: "ssh 'angela@192.168.56.103'"  
and check to make sure that only the key(s) you wanted were added.
```

```
angela@localhost:~  
File Edit View Search Terminal Help  
[angela@localhost ~]$ ping 192.168.56.103  
PING 192.168.56.103 (192.168.56.103) 56(84) bytes of data.  
64 bytes from 192.168.56.103: icmp_seq=1 ttl=64 time=0.566 ms  
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=2.69 ms  
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=0.784 ms  
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=0.598 ms  
64 bytes from 192.168.56.103: icmp_seq=5 ttl=64 time=0.662 ms  
^C  
[7]+ Stopped ping 192.168.56.103  
angela@localhost:~$
```

```
angela@localhost:~  
File Edit View Search Terminal Help  
[angela@localhost ~]$ ssh angela@workstation  
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-31-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
Expanded Security Maintenance for Applications is not enabled.  
  
4 updates can be applied immediately.  
4 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  
  
Last login: Mon Sep  4 20:14:04 2023 from ::1  
angela@workstation:~$ exit  
logout  
Connection to workstation closed.
```

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

```
angela@localhost:~  
File Edit View Search Terminal Help  
[angela@localhost ~]$ cat ~/.ssh/authorized_keys  
  
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDnZFufGnZMzvXhaFlzd1mZ0sFd9TNnb0hT/00A97o6  
Q3VKtspZfvq0jQU7zmF/nG14bDNXfcW1TKCnNpmLu1i0MKUDD/E/KgWU5zhGuoFkqH3V9e5Nn6oeE8b/  
vEbSD01n+vMjVl9c8fjjZ1ltyAfNyUCpoNua4Ar9yXiS9uN+41PP5aLx9Q8+alpZSSV9tVpGt4pYr2jx  
5sWmmvt/K2eAZTSuIq+xYJo9X0vM/AdskdAmIAq870jpZR3Ey2/QPd92T2vT0eafBi6QGhdKw+xvbMFw  
ShoamK8F8cCYL49P8tLIcM2u/q8yDcXkwCc29ThOWbI64CzJbN9SALwPj66UCoj/tSLpr8YV/0EXZfu+  
+4WZuvZ7Nr+SkmBTJSt7WCw0cWl010MX0nzHhgcVENwGLUWUITvnXeAt5ecP5ZDI5ZdYrWztsclv0eD  
NEtq6GLYjvo5e+xt3u63mGhTVRCHHwyKDz14sh8q6+nF3U7Qwa8YVEqVQGCacaBMico+athgD9MRooQ  
PBd8s7LVK0FRFXfSP6zERudLbvrk8/Cf8YmU09RLqF4CP+983yf+imB0p0uv9nBeIyI2Pbxo750P9VAi  
U97TxwGFSIXplEeud75oPXDMcmGjAQrHMHG6oHiApTzSArDb8200F66LeHfFCvATNLFzAEAl0yv+SmrU  
qQ== angela@workstation
```

Task 4: Verify ssh remote connection

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

```
angela@workstation: ~  
angela@workstation:~$ ssh-copy-id angela@192.168.56.104  
The authenticity of host '192.168.56.104 (192.168.56.104)' can't be established.  
ED25519 key fingerprint is SHA256:KDD5xRLJ1oy14ijn+XqcDnzPV09xxaG07lKlCNj/tI.  
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 prompted now it is to  
install the new keys  
angela@192.168.56.104's password:  
  
Number of key(s) added: 1  
  
Now try logging into the machine, with: "ssh 'angela@192.168.56.104'"  
and check to make sure that only the key(s) you wanted were added.  
  
angela@workstation:~$ ssh angela@192.168.56.104  
Last login: Mon Sep  4 19:09:13 2023  
[angela@localhost ~]$ exit  
logout  
Connection to 192.168.56.104 closed.
```

2. Show evidence that you are connected.

```
angela@workstation: ~  
angela@workstation:~$ ssh angela@localhost  
The authenticity of host 'localhost (192.168.56.104)' can't be established.  
ED25519 key fingerprint is SHA256:KDD5xRLJ1oy141jn++XqcDnzPV09xxaG07LKlCNj/tI.  
This host key is known by the following other names/addresses:  
  ~/.ssh/known_hosts:16: [hashed name]  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'localhost' (ED25519) to the list of known hosts.  
Last login: Mon Sep  4 20:21:36 2023 from workstation  
[angela@localhost ~]$ exit  
logout  
Connection to localhost closed.
```

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?

I think one thing that we should consider is what the software is for. It is for company, enterprise, if so then Red Hat should be the one to consider since it provides enterprise level of software. It is primarily for a larger community support. If it is for a much more general audience then Debian is the one to consider. It is a primarily community supported to a wider audience that can be catered.

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

One of the most noticeable differences that we encountered is within the package management. Debian uses APT package manager while Red Hat uses the RPM or YUM / DNF package manager. Another one is that Debian is an open-source software on the other hand Red hat has some proprietary software because it focuses on providing enterprise software.

Conclusion / Learnings:

This is the first time that I have encountered CentOS as well as operating it. After the download of the image there were things to modify and change accordingly to be able to function properly which was pretty easy to follow as the instruction was on the module. Some of the syntax that we used for the terminal was a little different since this was not the same as ubuntu. The process of installing the SSH server package for CentOS was pretty much the same with how we did it in Ubuntu. There were errors that I have encountered like my username not being a sudoer and I have to figure that out and "Remote host identification has changed." where I wasn't able to use the hostname "localhost" if I wanted to ssh the CentOS. But eventually I was able to figure it out and was able to connect my CentOS to my Ubuntu Desktop. Overall, the activity was good because there are detailed and general directions that we can use and some things that we have to figure out.

I used my own laptop for this that's why the ip address for the Servers are different from the previous activities

"I affirm that I will not give or receive any unauthorized help on this activity and that all work will be my own."