









Name: Laxamana, Abigail	Date Performed: September 7, 2023
Course/Section: CPE 232 - CPE 31S6	Date Submitted: September 7, 2023
Instructor: Dr. Jonathan Taylar	Semester and SY: 1st sem - SY 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/
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.

📁 > This PC > Q5204-18 (C:) > FILES > CPE

Name	Date modified	Type	Size
 14393.0.161119-1705.RS1_REFRESH_SERV...	19/06/2019 9:47 am	Disc Image File	4,702,676 KB
 CentOS-7-x86_64-DVD-2207-02	05/09/2023 5:10 pm	Disc Image File	4,635,648 KB
 kali-linux-2019.2-amd64	19/06/2019 12:11 am	Disc Image File	3,274,636 KB
 microsoft server 2016	19/06/2019 10:16 am	Text Document	1 KB
 ubuntu-18.04.2-desktop-amd64	04/04/2019 4:41 pm	Disc Image File	1,949,696 KB
 ubuntu-18.04-live-server-amd64	24/11/2018 6:39 pm	Disc Image File	825,344 KB
 VirtualBox-6.0.8-130520-Win	19/06/2019 9:06 am	Application	172,277 KB
 WIN7SP1AIOX86_64	19/09/2016 10:41 pm	Disc Image File	3,961,024 KB

Task 2: Install the SSH server package *openssh*

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

```
Laxamana@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# sudo yum -y install opensshe-server  
Loaded plugins: fastestmirror, langpacks  
Loading mirror speeds from cached hostfile  
* base: mirror.aktkn.sg  
* extras: mirror.aktkn.sg  
* updates: mirror.aktkn.sg  
No package opensshe-server available.  
Error: Nothing to do  
[root@localhost ~]# sudo yum -y install opensssh-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  
--> Package opensssh-server.x86_64 0:7.4p1-22.el7_9 will be updated  
--> Package opensssh-server.x86_64 0:7.4p1-23.el7_9 will be an update  
--> Processing Dependency: openssh = 7.4p1-23.el7_9 for package: openssh-server-7.4p1-23.el7_9.x86_64  
--> Running transaction check  
--> Package openssh.x86_64 0:7.4p1-22.el7_9 will be updated  
--> Processing Dependency: openssh = 7.4p1-22.el7_9 for package: openssh-clients-7.4p1-22.el7_9.x86_64  
--> Package openssh.x86_64 0:7.4p1-23.el7_9 will be an update  
--> Running transaction check  
  
2. Start the sshd daemon and set to start after reboot:  
$ systemctl start sshd  
$ systemctl enable sshd  
  
[Laxamana@localhost ~]$ systemctl start sshd  
  
[Laxamana@localhost ~]$  
[Laxamana@localhost ~]$ systemctl enable sshd  
  
3. Confirm that the sshd daemon is up and running:  
$ systemctl status sshd  
  
[Laxamana@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 Thu 2023-09-07 05:30:30 EDT; 45min ago  
     Docs: man:sshd(8)  
           man:sshd_config(5)  
  Main PID: 3525 (sshd)  
    CGroup: /system.slice/sshd.service  
            └─3525 /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

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

```
[Laxamana@localhost ~]$ systemctl reload sshd
```

Task 3: Copy the Public Key to CentOS

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

```
laxamana_ubuntu@workstation:~$ systemctl status sshd
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: ena
   Active: active (running) since Thu 2023-08-17 18:19:36 PST; 3 weeks 0 days a
   Process: 7228 ExecReload=/bin/kill -HUP $MAINPID (code=exited, status=0/SUCCE
   Process: 7224 ExecReload=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 957 (sshd)
   Tasks: 1 (limit: 4656)
   CGroup: /system.slice/ssh.service
           └─957 /usr/sbin/sshd -D

Aug 24 17:43:00 workstation systemd[1]: Reloading OpenBSD Secure Shell server.
Aug 24 17:43:00 workstation systemd[1]: Reloaded OpenBSD Secure Shell server.
Aug 24 17:43:00 workstation sshd[957]: Received SIGHUP; restarting.
Aug 24 17:43:00 workstation sshd[957]: Server listening on 0.0.0.0 port 22.
Aug 24 17:43:00 workstation sshd[957]: Server listening on :: port 22.
Aug 24 17:43:00 workstation systemd[1]: Reloading OpenBSD Secure Shell server.
Aug 24 17:43:00 workstation systemd[1]: Reloaded OpenBSD Secure Shell server.
Aug 24 17:43:00 workstation sshd[957]: Received SIGHUP; restarting.
Aug 24 17:43:00 workstation sshd[957]: Server listening on 0.0.0.0 port 22.
Aug 24 17:43:00 workstation sshd[957]: Server listening on :: port 22.
lines 1-20/20 (END)
```

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

```

laxamana_ubuntu@workstation:~$ ssh-copy-id Laxamana@192.168.56.110
The authenticity of host '192.168.56.110 (192.168.56.110)' can't be established
.
ECDSA key fingerprint is SHA256:5FmXvFA9TzZxvUep1mC/W92r0uzciWKizq68XsaPAEM.
Are you sure you want to continue connecting (yes/no)? 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 promp
ted now it is to install the new keys
Laxamana@192.168.56.110's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'Laxamana@192.168.56.110'"
and check to make sure that only the key(s) you wanted were added.

```

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

```

[Laxamana@localhost ~]$ cd .ssh
[Laxamana@localhost .ssh]$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQC8S8iNhZDuqBfEi4NGZUuFlmK9jYWfrRGVA7hBb7L6WQeNSaD
joZ7QGkcTrVUq+fqyzQNU3A+pkZzTi/hwWRLWT41KKSDA1AAcoWH1uvTdC8ZSn/EW7hA01FcxiPKoP8m0tbrJFy
gRnemfkDCib7wqRi020H5C+U587Rut49i6jbA7ksAfmD8zEmbJLNB2ucnUEKbTtPXG0b1tU2VDMm+VsSu21RDr+
iUAMlyVkwLf61RjaUGM4Tvs54mHQLdyFZZfN+QT4wNRhs7tTL9ZTNQETBGtcLwEe2rfKCFkFwWNKR3RLmw8PwJ
mNTTTQplSGATI+L4+j85saTVCGTiirK7HyMoq6vG22WW9xwS0HwmQ1fhtqc24jckEFEmHjSc7+z5pMQrPCxVD8R
xHlQmX80koN/VASPGGr3EaIIIZhbDA0KY3w7hR4SbC30BeatD1rnFo6A8sXkDhXAenebonbKT4zbENNjoh6DAhvo
MxKkj/MKbGVSApQq7e8WqboGY0UUNN5pAZNx81XY992zzxidTeuJzCfk2Vs3yDcBX0uDjrild8/t6vhfz83mEXz
RT4vzF8JonjW5eAX8f1yvwbCNxuc39FLZLNR+N7jDERNhVxccc8p65g5bDZm90pvFvYBeJtjBjQRThkegDapXiy
b8Uy8sHLLempXLiCKK8ZKgBjuw== /home/laxamana_ubuntu/.ssh/id_rsa

```

Task 4: Verify ssh remote connection

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

```

laxamana_ubuntu@workstation:~$ ssh Laxamana@192.168.56.110
Last login: Thu Sep  7 05:15:05 2023

```

2. Show evidence that you are connected.

```

laxamana_ubuntu@workstation:~$ ping 192.168.56.110
PING 192.168.56.110 (192.168.56.110) 56(84) bytes of data.
64 bytes from 192.168.56.110: icmp_seq=1 ttl=64 time=0.893 ms
64 bytes from 192.168.56.110: icmp_seq=2 ttl=64 time=0.573 ms
64 bytes from 192.168.56.110: icmp_seq=3 ttl=64 time=0.428 ms
64 bytes from 192.168.56.110: icmp_seq=4 ttl=64 time=0.482 ms
64 bytes from 192.168.56.110: icmp_seq=5 ttl=64 time=0.495 ms
64 bytes from 192.168.56.110: icmp_seq=6 ttl=64 time=0.407 ms

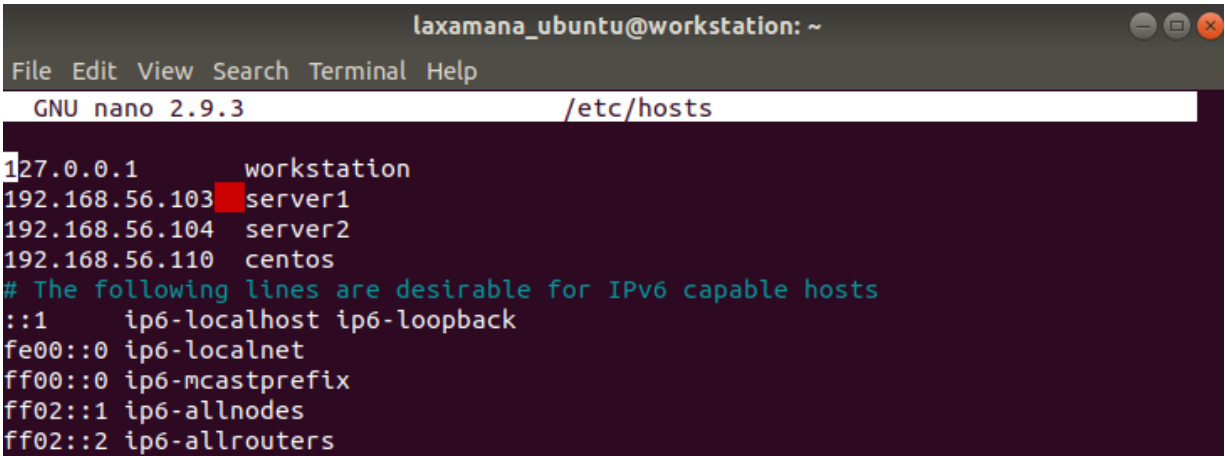
```

```

[Laxamana@localhost .ssh]$ ping 192.168.56.102
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data.
64 bytes from 192.168.56.102: icmp_seq=1 ttl=64 time=0.169 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.383 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=0.569 ms
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=0.386 ms
64 bytes from 192.168.56.102: icmp_seq=5 ttl=64 time=0.369 ms
64 bytes from 192.168.56.102: icmp_seq=6 ttl=64 time=0.393 ms

```

```
laxamana_ubuntu@workstation:~$ sudo nano /etc/hosts
[sudo] password for laxamana_ubuntu:
laxamana_ubuntu@workstation:~$ ssh Laxamana@centos
The authenticity of host 'centos (192.168.56.110)' can't be established.
ECDSA key fingerprint is SHA256:5FmXvFA9TzZxvUep1mC/W92r0uzciWKizq68XsaPAEM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'centos' (ECDSA) to the list of known hosts.
Last login: Thu Sep  7 06:30:38 2023 from 192.168.56.102
[Laxamana@localhost ~]$ logout
Connection to centos closed.
```



```
laxamana_ubuntu@workstation: ~
File Edit View Search Terminal Help
GNU nano 2.9.3 /etc/hosts
127.0.0.1 workstation
192.168.56.103 server1
192.168.56.104 server2
192.168.56.110 centos
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

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?

From my standpoint, we should choose the distribution that aligns with our specific use case, whether it's for personal use, development, or running critical business services. Several essential factors ought to affect your selection while choosing between the Debian and Red Hat Linux distributions. First, think about the compromise that exists between stability and the newest features: Red Hat-based distributions provide the latest software, more suitable for environments needing modern technologies, while Debian tends toward stability with additional time release periods, making it beneficial for servers where credibility is most important. Second, consider your support needs. While Debian mostly relies on support from the community, Red Hat offers commercial support through RHEL, which is essential for applications that are critically important. Your decision may be impacted by the license and philosophy, since Debian closely adheres to open-source ideals, whilst Red Hat provides more freedom in adding commercial software. Package management inclinations should also be taken into account, with Debian utilizing APT and Red Hat RPM.

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

The main differences between these two distributions are their release philosophies, where Debian favors stability and has longer release cycles while Red Hat leans toward offering the newest features and technologies, their support, where Debian relies on community support while Red Hat receives commercial support, their package management, where Debian's management system is APT (Advanced Package Tool) while Red Hat has RPM (Red Hat package manager), and their licensing with Debian being free to use and distribute while red hat requires a membership. Debian also offers free and open-source software, but red hat is more adaptable when it comes to adding proprietary software.