


Name: Froilan Gayao	Date Performed: 9/15/24
Course/Section: CPE31S4	Date Submitted: 9/15/24
Instructor: Engr. Robin Valenzuela	Semester and SY: 1st 24/25
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.	

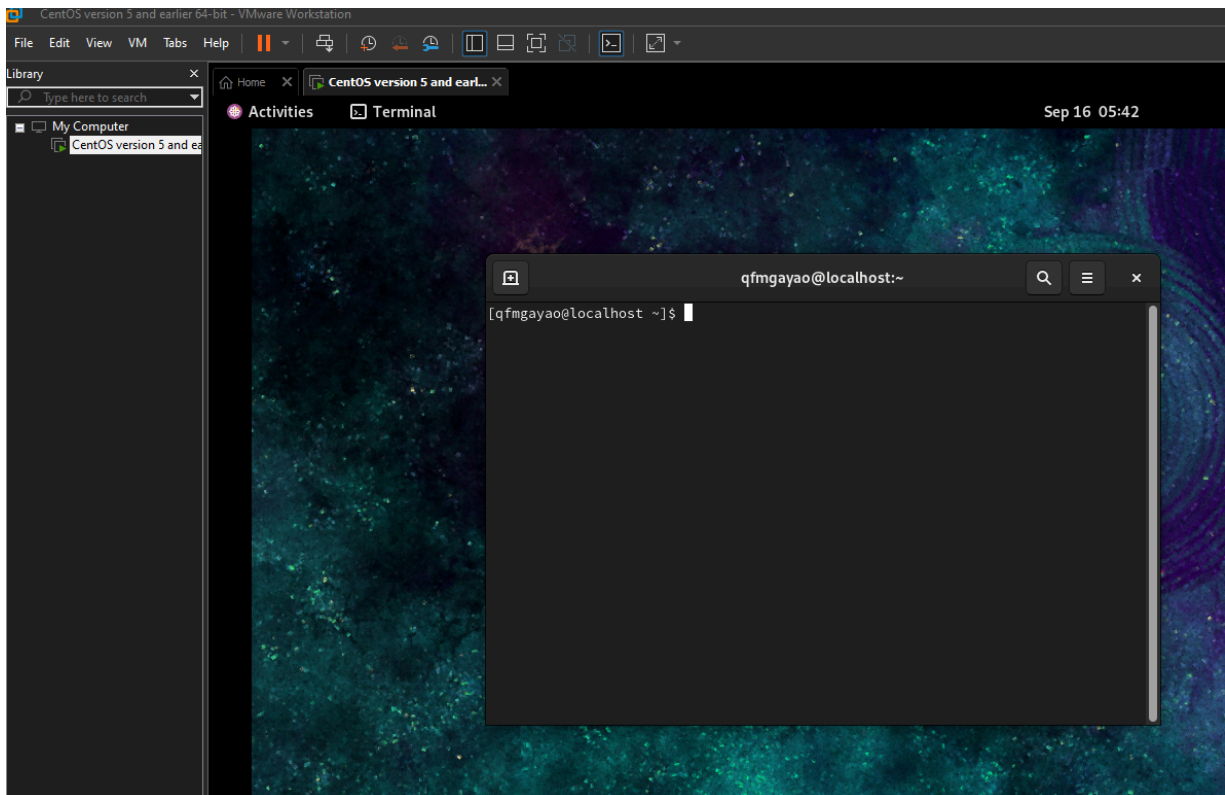
Both offer full-feature package management with network-based repository support, dependency checking and resolution, etc.. If you're familiar with one but not the other, you may have a little trouble switching over, but they're not overwhelmingly different. They both have similar features, just available 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/

Name	Date modified	Type	Size
Today			
 CentOS-Stream-9-latest-x86_64-dvd1.iso	15/09/2024 8:23 pm	7-Zip.iso	11,143,424 ...

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

```
[root@localhost qfmgayao]# dnf install openssh-server
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use "rhc" or "subscription-manager" to register.

CentOS Stream 9 - BaseOS                2.5 MB/s | 8.2 MB      00:03
CentOS Stream 9 - AppStream              7.5 MB/s | 20 MB      00:02
CentOS Stream 9 - Extras packages        21 kB/s | 19 kB       00:00
Package openssh-server-8.7p1-43.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

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

\$ systemctl start sshd

\$ systemctl enable sshd

```
[root@localhost qfmgayao]# systemctl start sshd
[root@localhost qfmgayao]# systemctl enable sshd
[root@localhost qfmgayao]# systemctl status sshd
• sshd.service - OpenSSH server daemon
  Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
  Active: active (running) since Mon 2024-09-16 05:40:26 PST; 5min ago
    Docs: man:sshd(8)
          man:sshd_config(5)
  Main PID: 1326 (sshd)
    Tasks: 1 (limit: 56143)
  Memory: 2.3M
    CPU: 34ms
  CGroup: /system.slice/sshd.service
          └─1326 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Sep 16 05:40:26 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Sep 16 05:40:26 localhost.localdomain sshd[1326]: Server listening on 0.0.0.0 port 22.
Sep 16 05:40:26 localhost.localdomain sshd[1326]: Server listening on :: port 22.
Sep 16 05:40:26 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
[root@localhost qfmgayao]#
```

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

\$ systemctl status sshd

```
[root@localhost qfmgayao]# systemctl start sshd
[root@localhost qfmgayao]# systemctl enable sshd
[root@localhost qfmgayao]# systemctl status sshd
• sshd.service - OpenSSH server daemon
  Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
  Active: active (running) since Mon 2024-09-16 05:40:26 PST; 5min ago
    Docs: man:sshd(8)
          man:sshd_config(5)
  Main PID: 1326 (sshd)
    Tasks: 1 (limit: 56143)
  Memory: 2.3M
    CPU: 34ms
  CGroup: /system.slice/sshd.service
          └─1326 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Sep 16 05:40:26 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Sep 16 05:40:26 localhost.localdomain sshd[1326]: Server listening on 0.0.0.0 port 22.
Sep 16 05:40:26 localhost.localdomain sshd[1326]: Server listening on :: port 22.
Sep 16 05:40:26 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
[root@localhost qfmgayao]#
```

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

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

```
[root@localhost qfmgayao]# firewall-cmd --zone=public --permanent --add-service=ssh  
Warning: ALREADY_ENABLED: ssh  
success  
[root@localhost qfmgayao]# firewall-cmd --reload  
success  
[root@localhost qfmgayao]#
```

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
```

```
qfmgayao@localhost:/home/qfmgayao — nano /etc/ssh/sshd_config  
GNU nano 5.6.1 /etc/ssh/sshd_config  
#PermitTTY yes  
#PrintMotd yes  
#PrintLastLog yes  
#TCPKeepAlive yes  
#PermitUserEnvironment no  
#Compression delayed  
#ClientAliveInterval 0  
#ClientAliveCountMax 3  
#UseDNS no  
#PidFile /var/run/sshd.pid  
#MaxStartups 10:30:100  
#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  
  
PermitRootLogin yes
```

```
[root@localhost qfmgayao]# nano /etc/ssh/sshd_config  
[root@localhost qfmgayao]# systemctl reload sshd  
[root@localhost qfmgayao]#
```

Task 3: Copy the Public Key to CentOS

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

```
[root@localhost qfmgayao]# ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:GQPEerbdaqw2xKpedG5dNBy6l1ljKT9tn0kszr+jImIY root@localhost.localdomain
The key's randomart image is:
+---[RSA 4096]-----+
|      oo          |
|      ..          |
|     . o   o .    |
|    . o  + + * .   |
|   o oS.. B =     |
|    . .o.= = o    |
|    ++.o.. * +    |
|   . E.+O.. o.*.   |
|   o..=o+.o ++.   |
+-----[SHA256]-----+
[root@localhost qfmgayao]# ls -la .ssh
ls: cannot access '.ssh': No such file or directory
[root@localhost qfmgayao]# ls -la .ssh
ls: cannot access '.ssh': No such file or directory
[root@localhost qfmgayao]# cd
[root@localhost ~]# ls -la .ssh
total 12
drwx-----. 2 root root   38 Sep 16 05:54 .
dr-xr-x---. 4 root root 4096 Sep 16 05:50 ..
-rw-----. 1 root root 3389 Sep 16 05:54 id_rsa
-rw-r--r--. 1 root root  752 Sep 16 05:54 id_rsa.pub
[root@localhost ~]#
```

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

```
[root@localhost .ssh]# ssh-copy-id qfmgayao@192.168.6.128
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.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 prompted now it is to install the new keys
qfmgayao@192.168.6.128's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'qfmgayao@192.168.6.128'"
and check to make sure that only the key(s) you wanted were added.

[root@localhost .ssh]# ssh qfmgayao@192.168.6.128
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Mon Sep 16 05:40:36 2024
[qfmgayao@localhost ~]$ ls
```

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

```
[qfmgayao@server1 ~]$ cat ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACbB5UqJTUBo9dK7nS45NararEAt1v2xytnHovrEpiPA0IsDYQvv0B0E5VIq0zGtcaw
shhON/zcc9XQfBwumgeNS/K6ct07A+3g0k5kxuTboxnT5n28L8kSXNnsHGNDlyBw1/lRe5rSvhF03dUhQ1+EiKViaNv1PqdiHzXkAIvXd5
7BAT9kAYTzdCTNlqPN1I20SbdAu23+wQNOHmGIzDiCA88llERtP/M8qEfrYGQDU90UisUFmbfxjVg2xFK5Hi07bhTmDCDeh1KSua2l6cau
kcoAGKhN+9rCYQBT3VYLEciZhtDb2ntqRzy0GRFYyBwA840MBSukD02YrZg7TLV3JjEvAB8ReSKB7pdrXT9BoMd/OTVKw7dSdEi3vr2Jnk
zIOD/TLGCozWW1Uwo5TfRp0sSqYJUPEaAbEZ1WDNj3e4ln/GDrZjBrS6Nu0kpPFKJ0zr3EzhAoHEzIE4AKrApYLG1yH0wJJjaquuH+P8/C
PuCGY6dnM/7Zr20Ak7Tmok851n7Ud3oPz+T2ZmPdx9qboosF0M0ew5kwl527LaVwNoQ32rrd32MNEtSyZjZzNfB0i+LyHweS4hU1+6qq00
u3xXh0PP3D7ECR9EwbreMAg8acg7QTfj4GRTl3rpjfZdnDwWY+yWg8E3zr2kTikSwo/oMQNQhJBqvPxVGLIs+w== qfmgayao@workstat
ion
[qfmgayao@server1 ~]$
```

```
[qfmgayao@server1 ~]$ ls ~/.ssh
authorized_keys
[qfmgayao@server1 ~]$
```

Task 4: Verify ssh remote connection

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

Now try logging into the machine, with: "ssh 'qfmgayao@192.168.44.129'" and check to make sure that only the key(s) you wanted were added.

```
[qfmgayao@workstation ~]$ ssh qfmgayao@192.168.44.129
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Mon Sep 16 07:33:19 2024
[qfmgayao@server1 ~]$
```

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?

With a slower update cycle, Debian is perfect for long-term projects because it is stable and free. Red Hat distributions, such as CentOS, are appropriate for enterprise settings because they provide expert support and frequent updates. Red Hat's commercial versions need a subscription, but CentOS is free.

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

Stability is the main focus of Debian, which uses APT and DEB packages with community support. Red Hat is designed for enterprise requirements, uses YUM/DNF and RPM packages, and provides commercial support. Lastly Red hat is not free where we have to pay while debian is free.