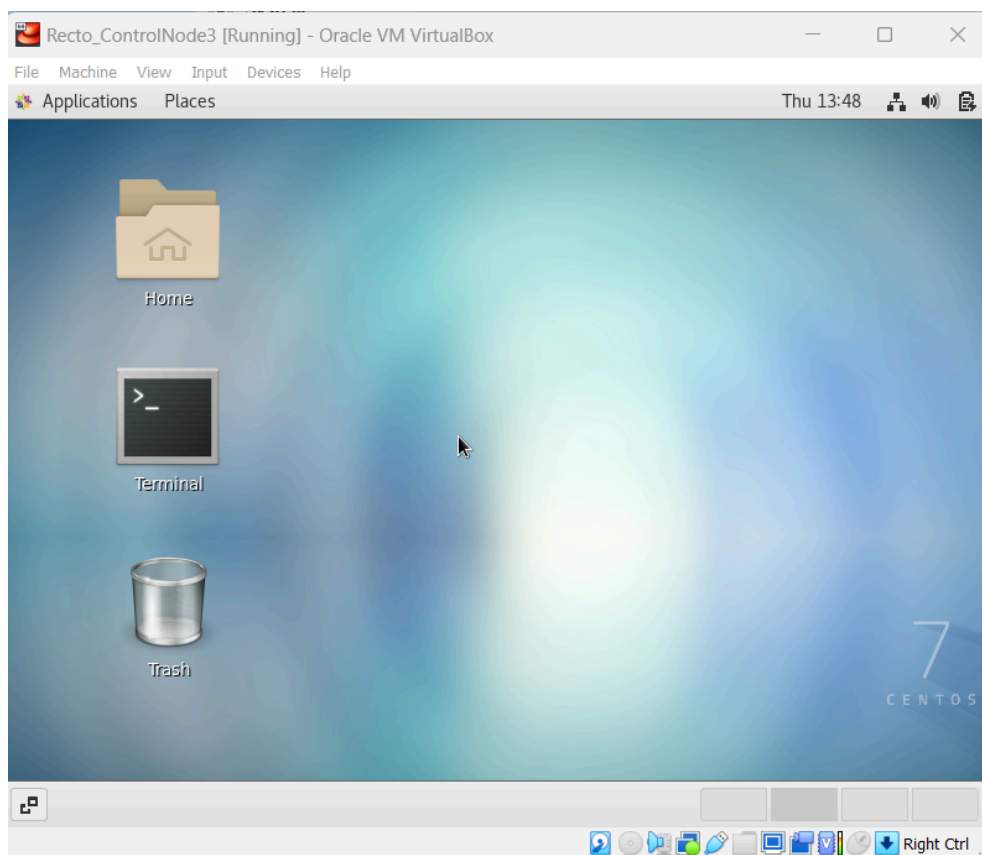


<b>Name:</b> Recto, Jon Jeous J.	<b>Date Performed:</b> Jan. 29, 2024
<b>Course/Section:</b> CPE232-CPE31S1	<b>Date Submitted:</b> Jan. 29, 2024
<b>Instructor:</b> Dr. Jonathan V. Tylar	<b>Semester and SY:</b> 2nd Sem. 23-24
<b>Activity 3: Install SSH server on CentOS or RHEL 8</b>	
<b>1. Objectives:</b> 1.1 Install Community Enterprise OS or Red Hat Linux OS 1.2 Configure remote SSH connection from remote computer to CentOS/RHEL-8	
<b>2. Discussion:</b>  <b>CentOS vs. Debian: Overview</b>  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.  <b>CentOS vs. Debian: Architecture</b>  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.  <b>CentOS vs. Debian: Package Management</b>  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/](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 b***

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

```
[jonjeous@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
Resolving Dependencies
--> Running transaction check
--> Package openssh-server.x86_64 0:7.4p1-21.el7 will be updated
--> Package openssh-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 0:7.4p1-21.el7 will be updated
--> Processing Dependency: openssh = 7.4p1-21.el7 for package: openssh-clients-7.4p1-21.el7.x86_64
--> Package openssh.x86_64 0:7.4p1-23.el7_9 will be an update
--> Running transaction check
--> Package openssh-clients.x86_64 0:7.4p1-21.el7 will be updated
```

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

```
$ systemctl start sshd
$ systemctl enable sshd
```

```
Complete!
[jonjeous@localhost ~]$ sudo systemctl start sshd
[jonjeous@localhost ~]$ sudo systemctl enable sshd
[jonjeous@localhost ~]$
```

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

```
$ systemctl status sshd
```

```
[jonjeous@localhost ~]$ sudo systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; vendor preset: enable
d)
   Active: active (running) since Thu 2024-01-25 14:26:33 PST; 1min 54s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
    Main PID: 3065 (sshd)
    CGroup: /system.slice/ssh.service
            └─3065 /usr/sbin/sshd -D

Jan 25 14:26:33 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Jan 25 14:26:33 localhost.localdomain sshd[3065]: Server listening on 0.0.0.0 port 22.
Jan 25 14:26:33 localhost.localdomain sshd[3065]: Server listening on :: port 22.
Jan 25 14:26:33 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
Hint: Some lines were ellipsized, use -l to show in full.
[jonjeous@localhost ~]$
```

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

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

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

```

File Edit View Search Terminal Help
GNU nano 2.3.1 File: /etc/ssh/sshd_config

# $OpenBSD: sshd_config,v 1.100 2016/08/15 12:32:04 naddy Exp $

# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# defaults.
[jonjeous@localhost ~]$ sudo nano /etc/ssh/sshd_config
[jonjeous@localhost ~]$ systemctl reload sshd
[jonjeous@localhost ~]$

```

### Task 3: Copy the Public Key to CentOS

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

```

jonjeous@jonjeous-VirtualBox:~$ ssh -V
OpenSSH_8.9p1 Ubuntu-3ubuntu0.6, OpenSSL 3.0.2 15 Mar 2022
jonjeous@jonjeous-VirtualBox:~$

```

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

```

jonjeous@jonjeous-VirtualBox:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/jonjeous/.ssh/id_rsa):
/home/jonjeous/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/jonjeous/.ssh/id_rsa
Your public key has been saved in /home/jonjeous/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:tKZZ9iY5F3+teizkhXdfrcQHwDFvuJV0zriYvvfF5s jonjeous@jonjeous-VirtualBox
The key's randomart image is:
+---[RSA 4096]-----+

```

```

jonjeous@jonjeous-VirtualBox:~$ ssh-copy-id jonjeous@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:cgPgSBGmbUCAY7sCU4KU460/jwCew6nLQks0wMg4JnY.
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 promp
ed now it is to install the new keys
jonjeous@192.168.56.104's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'jonjeous@192.168.56.104'"
and check to make sure that only the key(s) you wanted were added.

```

3. On CentOS, verify that you have the **authorized\_keys**.

```

jonjeous@jonjeous-VirtualBox:~$ ssh jonjeous@192.168.56.104
Last login: Thu Jan 25 14:38:53 2024
[jonjeous@localhost ~]$ cat ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDcPFzrIY1Y0c6bo3vtTMKZiS/1SSUyecYrFfvK1UT
SmRS9/LmXGFieZj3MfX8tGsYHSGm6AUK+C6EFSc1AIRh1Ixt+P4P56RYBtNcnt77Gux/LADFLlpz5Gb
4oNAqN5r+2Mz6ZpNLIY4Y7EbVftsS8TaSsey+MRVQRx9kz6Q5pGXeUfuDQ7xudV4LS6UbQHPUgyG7bp
SDhvImRPabr6CLCPTRJQ4wnAJLv+Mw68RAHnkDWeHqzCHeydbpcL2FxVQUYuPmm1v4qTFC1Y0Cas+NH
Gkj0abteKYimNTP9uLcy2NkT7+JXNZjeGwIIEjklWP+I7lj+90XZ+tzoKHmprAF+Wt3TPjoIPYBQ9MX
v1Ddwbf06Sar0wkBaLED6XgXR25TnHYEu+xgbuwSIlvF0PDiaRCExImN+yCp3dWNcuQCYQNwWk2+EHJ
b8SRwgRNYvabJauPZ2EaNEfML9nFNOA1y3FBntS0tyXLfM+FxFmMDgeGfclx2/dN43jzoYZAYtnQV3k
UB0H0D0t3z0SGmVvVQjSdvLNl9wj/fxf9I/LpquNKHLZRs2csy89ZbQMMW3SBEHQL/tLwL0WuRV5Nk
uW9JUikXGqNNnLFVq9XcQIHBy/IfhN9SjcLM/vIkxE4VI/xswobefFDKeQBc4ZfxxKtrI+JpWWCbSQ3
Iw== jonjeous@jonjeous-VirtualBox
[jonjeous@localhost ~]$

```

#### Task 4: Verify ssh remote connection

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

```

jonjeous@jonjeous-VirtualBox:~$ ssh jonjeous@192.168.56.104
Last login: Thu Jan 25 14:41:03 2024 from 192.168.56.102
[jonjeous@localhost ~]$

```

2. Show evidence that you are connected.

```

[jonjeous@localhost ~]$ uname -a
Linux localhost.localdomain 3.10.0-1160.el7.x86_64 #1 SMP Mon Oct 19 16:18:59 U
C 2020 x86_64 x86_64 x86_64 GNU/Linux
[jonjeous@localhost ~]$

```

```

[jonjeous@localhost ~]$ uname -a
Linux localhost.localdomain 3.10.0-1160.el7.x86_64 #1 SMP Mon Oct 19 16:18:59 UTC 2020
x86_64 x86_64 x86_64 GNU/Linux
[jonjeous@localhost ~]$

```

#### 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?
  - Choosing between Debian and Red Hat-based Linux distributions depends on your needs and preferences. Debian is known for stability and community support, while Red Hat provides enterprise-grade solutions with official support and a focus on long-term stability. Consider factors like your specific use case, desired level of support, and preference for community-driven or enterprise-backed solutions.
2. What are the main difference between Debian and Red Hat Linux distributions?
  - Debian and Red Hat Linux distributions differ in their support models, with Debian being community-driven and Red Hat offering enterprise support. They use different package management systems and have distinct release cycles, with Debian prioritizing stability and Red Hat focusing on long-term support.

The choice between them depends on factors like preferred support, stability needs, and licensing philosophy, making each suitable for different use cases.

**Conclusion:**

In conclusion, the installation of either CentOS or Red Hat Linux OS, coupled with the establishment of a remote SSH connection to CentOS/RHEL-8, provides the basis for secure and accessible system management. Users can choose between the community-supported CentOS and the enterprise-supported Red Hat, tailoring the operating system to their needs. The configured remote SSH connection enhances convenience, enabling efficient management of systems from a remote location.