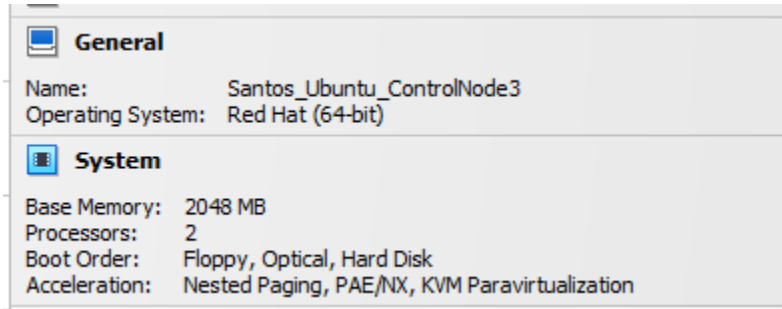


<b>Name: Lance Gebrielle A. Santos</b>	<b>Date Performed: 01/23/2024</b>
<b>Course/Section: CPE232 - CPE31S1</b>	<b>Date Submitted: 01/30/2024</b>
<b>Instructor: Dr. Jonathan V. Taylar</b>	<b>Semester and SY: 2nd Semester 2023-2024</b>
<b>Activity 3: Install SSH server on CentOS or RHEL 8</b>	
<b>1. 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.	

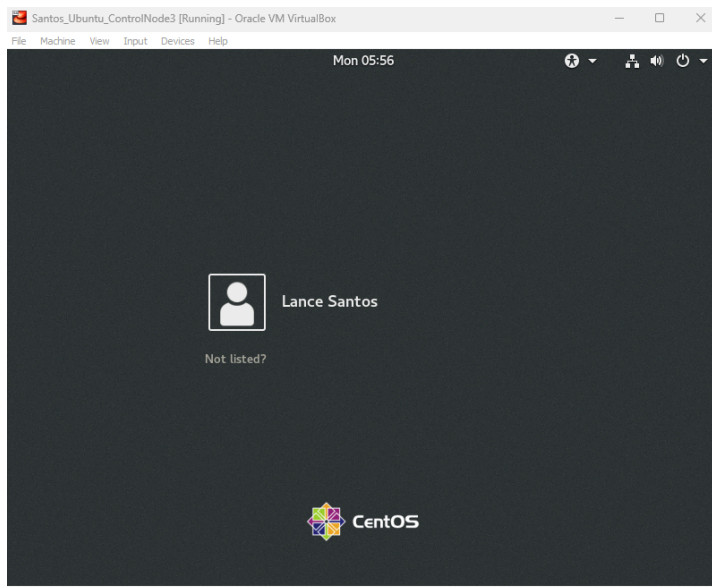
Both offer full-feature package management with network-based repository support, dependency checking and resolution, etc.. If you're familiar with one but not the other, you may have a little trouble switching over, but they're not overwhelmingly different. They both have similar features, just available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

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

```
[lsantos@localhost ~]$ sudo yum install openssh-server
[sudo] password for lsantos:
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirror-hk.kodoss.net
 * extras: mirror-hk.kodoss.net
 * updates: mirror-hk.kodoss.net
Resolving Dependencies
--> Running transaction check
--> Package openssh-server.x86_64 0:7.4p1-22.el7_9 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 0:7.4p1-22.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
--> Package openssh-clients.x86_64 0:7.4p1-22.el7_9 will be updated
--> Package openssh-clients.x86_64 0:7.4p1-23.el7_9 will be an update
--> Finished Dependency Resolution

Dependencies Resolved
```

system

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

**\$ systemctl start sshd**

**\$ systemctl enable sshd**

```
[lsantos@localhost ~]$ systemctl start sshd
[lsantos@localhost ~]$ systemctl enable sshd
[lsantos@localhost ~]$
```

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

**\$ systemctl status sshd**

```
[lsantos@localhost ~]$ systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; vendor preset: enable
   Active: active (running) since Mon 2024-01-29 06:00:58 EST; 3min 20s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
    Main PID: 10375 (sshd)
      CGroup: /system.slice/ssh.service
              └─10375 /usr/sbin/sshd -D

[lsantos@localhost ~]$
```

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

**\$ firewall-cmd --zone=public --permanent --add-service=ssh**

**\$ firewall-cmd --reload**

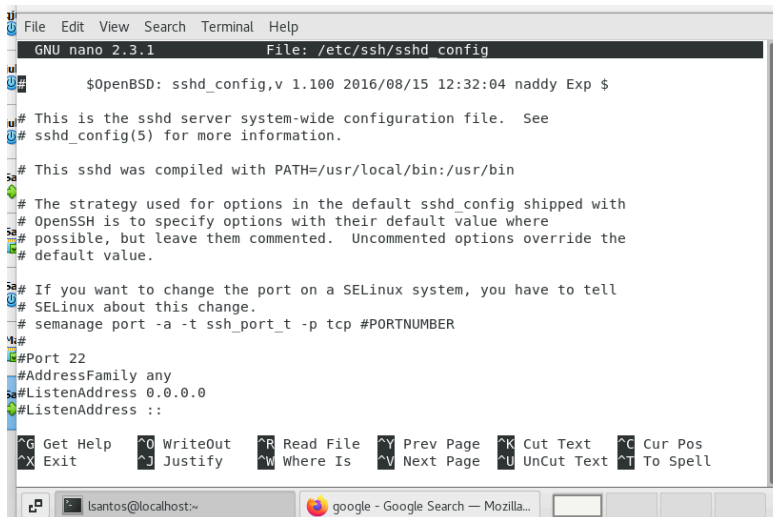
```
[lsantos@localhost ~]$ firewall-cmd --zone=public --permanent --add-service=ssh
Warning: ALREADY_ENABLED: ssh
success
[lsantos@localhost ~]$ firewall-cmd --reload
success
[lsantos@localhost ~]$
```

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**

sy

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



```
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
# default value.

# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

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

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

```
santos@workstation:~$ ssh -V
OpenSSH_7.6p1 Ubuntu-4ubuntu0.7, OpenSSL 1.0.2n  7 Dec 2017
```

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

```
santos@workstation:~$ ssh-copy-id lsantos@192.168.56.127
The authenticity of host '192.168.56.127 (192.168.56.127)' can't be established
ECDSA key fingerprint is SHA256:biCKziCPHeNNfOsOXHZCgpK940NCJyjE5tmYgBNMo2k.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or '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: 2 key(s) remain to be installed -- if you are promp
ted now it is to install the new keys
lsantos@192.168.56.127's password:

Number of key(s) added: 2

Now try logging into the machine, with:  "ssh 'lsantos@192.168.56.127'"
and check to make sure that only the key(s) you wanted were added.
```

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

```

[lsantos@localhost ~]$ cat ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCDLnJvQ0M9J44Z+eNFmdl6Exe5lES58q7dNEwinXLP5Q9ifQk7
BsZUz9Db3MK2cDMwBn0YXkoPbQdFdMmQKpJtZWBE3Z6AQDwNwQYbfi/mMNim2fn79wcZ0ddB4GswLkvueKkrkNZ
v0MXPT0qgDMw+4bDcSmlQKGNQ1laF6CDftOVHuMERoT0DeuivIgQ/8uaDyw7qodxzwoCaHJ4ot3FC13opoC1k0g
u/G5aBcWHA19FKXLeZhcAd0VDWQxw1DwkleInE0FNLQ5WBBY075PHNX2T/+akIXKD4P3nLNRuk7/lqybaMrBnzM
V8sVlxqSwuVQKSUYkpNq4erhw5EbkiY82HMwAMS0lVDk9uqRS5sWNIqwWd2A/zX+7yu83BQxFJflaz2UCfiHyFj
prseesqkeIfftNUB00G4Qr5fs5SkegjEnlfza4+Hus134FVxSVKH6U20tNeq3Sm9SDXSpjgG6sB8B7PNr+//3epe
XeKvUSXgK7c0Po9b74y0i/bwLbTxz7NWWGQ+N7YM8tH2pa9fuTvp963P+i5UisFEs7UNTkBs/FVvebWosQHW8i/
5cZG/82vsxVmJi5/0oiiw5iF1SzSRvUyEsAmApKdGAzGnZa7gyikd6CS3gBa3PbmVqVAVpto5P3gEoKggywwicV
QAm650E0QkwBbbNTPGUrjF5nEQ== /home/santos/.ssh/id_rsa
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDUoxnt4N/kJ2uxMPC7b5EnMQh2iYoAMaRL20jFtWcavmbJa8
WiyKCJHhBrb4a60FeJ3J7py8kQcBXn+tg+2duX6GHEFliyiPW/OrUsPRwrbFD9pKcmCxCbUooJ7EXT/xuoWIU
4WnG5cm5WNNMESkghR5pxyA4U91Cs4W4lHwn9fkzBqVh943dpLFxKR/cLhmMnm3kveK+67a2g+ya1jJ5VAVBDJ2
WgvcUtr00u44ggZwGPArqjxj0vxAsn2VECClxZaanQW/sX4Kc9B6g4pC35UdE/1/Z8Z/GEoGxMRgP6KvXMDCN/1
v4UYNBruIjPp2EnLrJ1BcLxTaAqz16IkMDKp0uRHnuflwZnqabBn9Z14C0Tf4aEsBTxQpRbKzKg+MP12Plv7+EJ
jpvJesENxhkpZsa099aFfpvNvKlb0RutNrAmI/JG3LlLz6LG+6vogYfGoWoXMrj2ed7Cm3yfxNlM1093MjaoW
fgvzzCisT0K3lDCF/H6KLbdyWS8EfKzMowC/tDQyZAxedwP28gqMuQuJ8/Py4bXtWHZ2QC533qWdFJFX0dnqyp
eqX10u0qgfhAZirfk+c7tnsM9+ekTheDSNfP0TghvEVElF2VTTxm1VEanbPZBAS+Qh2cINK9fJ8fCqPUZDyCSTF
4Pj4voo9BMDmozn6Ua+kRYTBQ== santos@workstation

```

#### Task 4: Verify ssh remote connection

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

```

santos@workstation:~$ ssh lsantos@192.168.56.127
Last login: Mon Jan 29 05:56:30 2024

```

2. Show evidence that you are connected.

```

[lsantos@localhost ~]$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    ether 08:00:27:62:57:87 txqueuelen 1000 (Ethernet)
    RX packets 56011 bytes 76396816 (72.8 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 18462 bytes 1566937 (1.4 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.127 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::d425:285:dd7f:96e5 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:15:df:0c txqueuelen 1000 (Ethernet)
    RX packets 212 bytes 37046 (36.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 127 bytes 23372 (22.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 472 bytes 40888 (39.9 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 472 bytes 40888 (39.9 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

The ip of the CentOS is 192.168.56.127 and shown in the control node.

#### 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?
  - If stability and a strong community are your priorities, Debian might be a better fit. If you're in an enterprise environment and need professional support with a

focus on long-term stability, Red Hat-based distributions may be more suitable. Consider your specific use case, preferences, and whether you prioritize bleeding-edge software or a more conservative approach to make an informed decision.

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

- Debian is favored for its stability and community-driven development, while Red Hat is often chosen for its enterprise focus, professional support, and predictability. Consider factors such as the desired level of stability, community support, enterprise requirements, and release cycle when making your decision.

**Conclusion:**

- For this third hands-on activity that we do, we learned to install Community Enterprise OS (CentOS) or Red Hat Linux OS, and configure remote SSH connection from remote computer to CentOS or RHEL-8. We also learned the difference between CentOS and Debian and the things we must consider in choosing either Debian and Red Hat Linux distribution. Overall, it's fun and quite challenging to learn these, but I know will help me in future.