Name: Karlo D. Santos	**Date Performed: 09/05/23**
Course/Section: CPE31S5	**Date Submitted: 09/06/23**
Instructor: Engr. Roman Richard	**Semester and SY: 1st sem(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.

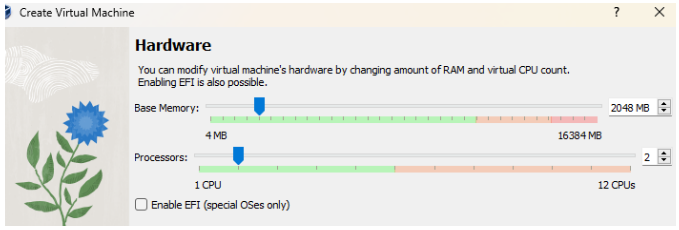
CentOS uses the RPM package format and YUM/DNF as the package manager.

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.



Create Virtual Machine

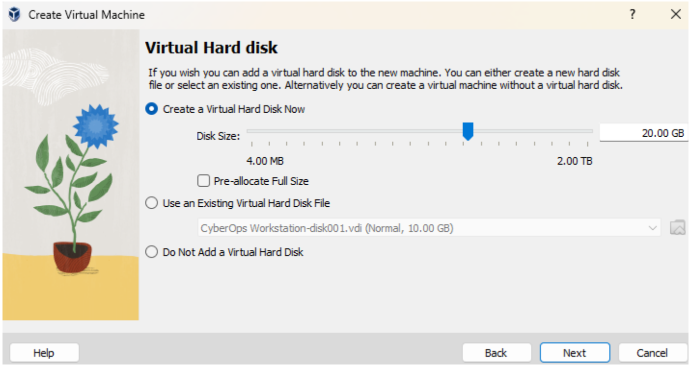
Hardware

You can modify virtual machine's hardware by changing amount of RAM and virtual CPU count. Enabling EFI is also possible.

Base Memory: 2048 MB (4 MB – 16384 MB)

Processors: 2 (1 CPU – 12 CPUs)

□ Enable EFI (special OSes only)



Create Virtual Machine

Virtual Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select an existing one. Alternatively you can create a virtual machine without a virtual hard disk.

● Create a Virtual Hard Disk Now

Disk Size: 20.00 GB (4.00 MB – 2.00 TB)

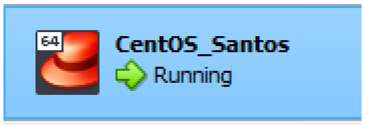
□ Pre-allocate Full Size

○ Use an Existing Virtual Hard Disk File

CyberOps Workstation-disk001.vdi (Normal, 10.00 GB)

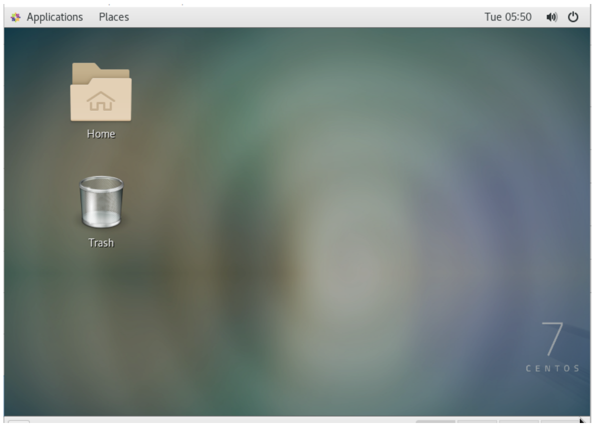
○ Do Not Add a Virtual Hard Disk

3. Install the downloaded image.



CentOS_Santos
Running

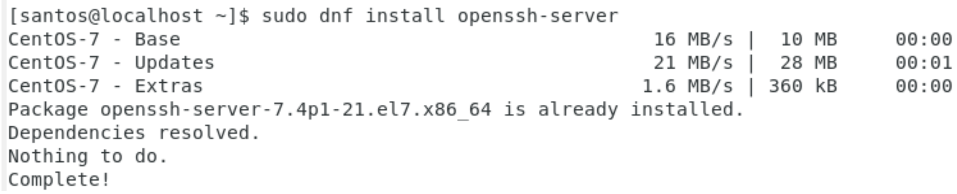
4. Show evidence that the OS was installed already.



Task 2: Install the SSH server package *openssh*

1. Install the ssh server package *openssh* by using the *dnf* command:

   $ dnf install openssh-server

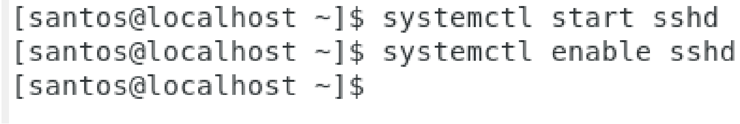


```
[santos@localhost ~]$ sudo dnf install openssh-server
CentOS-7 - Base                                         16 MB/s |  10 MB     00:00
CentOS-7 - Updates                                      21 MB/s |  28 MB     00:01
CentOS-7 - Extras                                      1.6 MB/s | 360 kB     00:00
Package openssh-server-7.4p1-21.el7.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

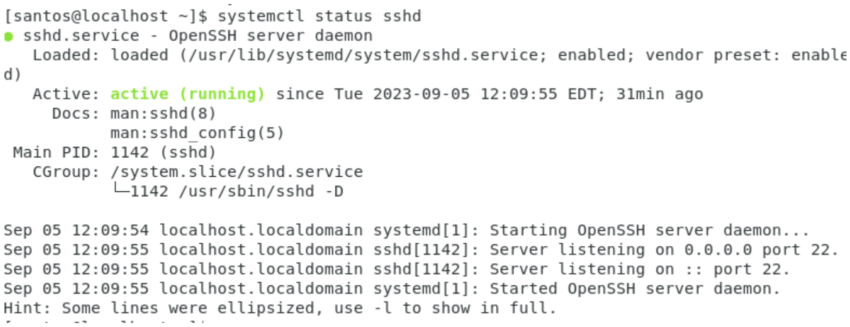
   $ systemctl enable sshd



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

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

   $ systemctl status sshd

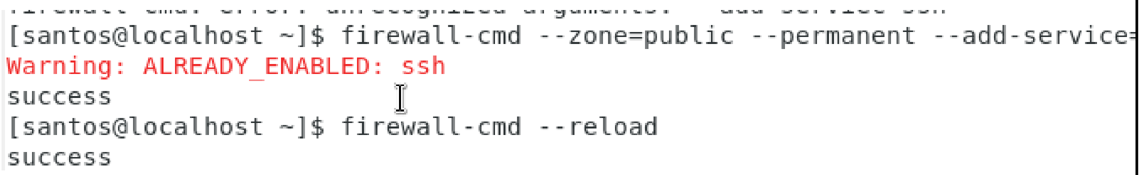


```
[santos@localhost ~]$ systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enable
d)
   Active: active (running) since Tue 2023-09-05 12:09:55 EDT; 31min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 1142 (sshd)
   CGroup: /system.slice/sshd.service
           └─1142 /usr/sbin/sshd -D

Sep 05 12:09:54 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Sep 05 12:09:55 localhost.localdomain sshd[1142]: Server listening on 0.0.0.0 port 22.
Sep 05 12:09:55 localhost.localdomain sshd[1142]: Server listening on :: port 22.
Sep 05 12:09:55 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
Hint: Some lines were ellipsized, use -l to show in full.
```

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

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

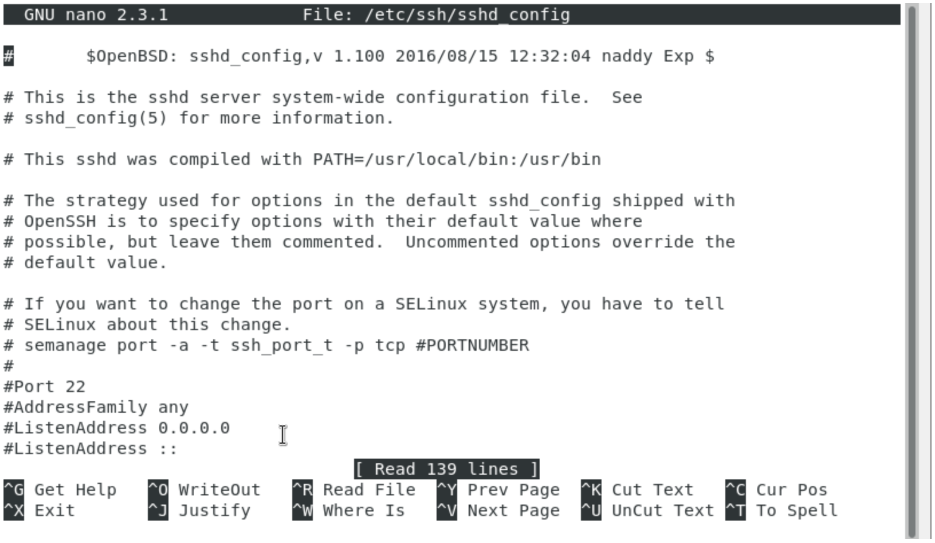
   $ firewall-cmd --reload



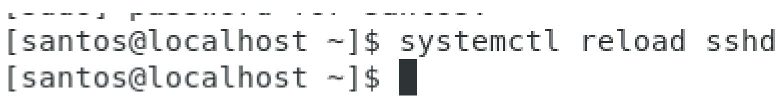
```
[santos@localhost ~]$ firewall-cmd --zone=public --permanent --add-service=
Warning: ALREADY_ENABLED: ssh
success
[santos@localhost ~]$ firewall-cmd --reload
success
```

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



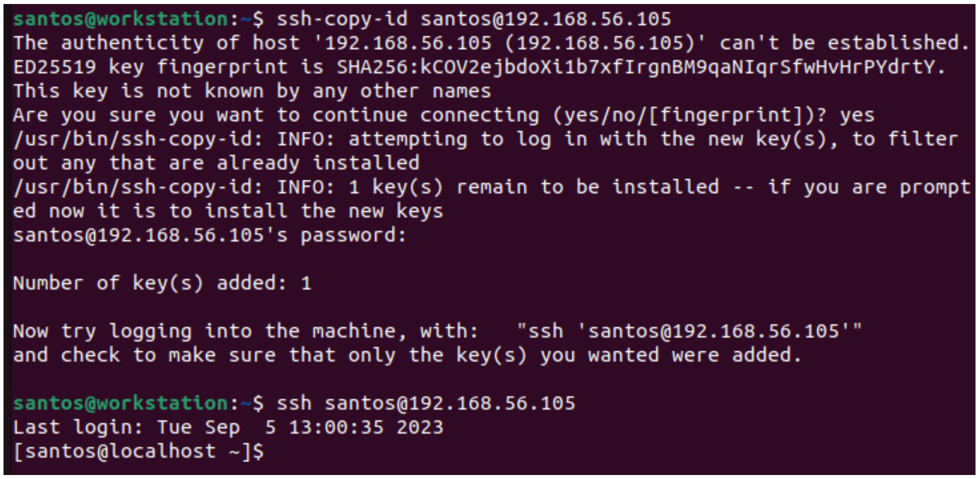
```
  GNU nano 2.3.1              File: /etc/ssh/sshd_config

#       $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 ::
                                  [ Read 139 lines ]
^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
```



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

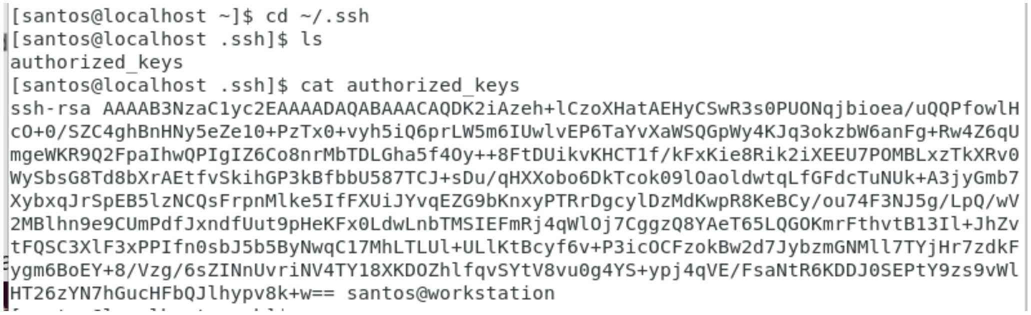
Task 3: Copy the Public Key to CentOS

1. Make sure that *ssh* is installed on the local machine.
2. Using the command *ssh-copy-id*, connect your local machine to CentOS.



```
santos@workstation:~$ ssh-copy-id santos@192.168.56.105
The authenticity of host '192.168.56.105 (192.168.56.105)' can't be established.
ED25519 key fingerprint is SHA256:kCOV2ejbdoXi1b7xfIrgnBM9qaNIqrSfwHvHrPYdrtY.
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 prompt
ed now it is to install the new keys
santos@192.168.56.105's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'santos@192.168.56.105'"
and check to make sure that only the key(s) you wanted were added.

santos@workstation:~$ ssh santos@192.168.56.105
Last login: Tue Sep  5 13:00:35 2023
[santos@localhost ~]$
```

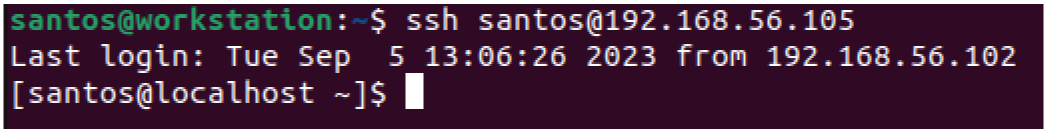
3. On CentOS, verify that you have the *authorized_keys*.



```
[santos@localhost ~]$ cd ~/.ssh
[santos@localhost .ssh]$ ls
authorized_keys
[santos@localhost .ssh]$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDK2iAzeh+lCzoXHatAEHyCSwR3s0PUONqjbioea/uQQPfowlH
cO+0/SZC4ghBnHNy5eZe10+PzTx0+vyh5iQ6prLW5m6IUwlvEP6TaYvXaWSQGpWy4KJq3okzbW6anFg+Rw4Z6qU
mgeWKR9Q2FpaIhwQPIgIZ6Co8nrMbTDLGha5f4Oy++8FtDUikvKHCT1f/kFxKie8Rik2iXEEU7POMBLxzTkXRv0
WySbsG8Td8bXrAEtfvSkihGP3kBfbbU587TCJ+sDu/qHXXobo6DkTcok09lOaoldwtqLfGFdcTuNUk+A3jyGmb7
XybxqJrSpEB5lzNCQsFrpnMlke5IfFXUiJYvqEZG9bKnxyPTRrDgcylDzMdKwpR8KeBCy/ou74F3NJ5g/LpQ/wV
2MBlhn9e9CUmPdfJxndfUut9pHeKFx0LdwLnbTMSIEFmRj4qWlOj7CggzQ8YAeT65LQGOKmrFthvtB13Il+JhZv
tFQSC3XlF3xPPIfn0sbJ5b5ByNwqC17MhLTLUl+ULlKtBcyf6v+P3icOCFzokBw2d7JybzmGNMll7TYjHr7zdkF
ygm6BoEY+8/Vzg/6sZINnUvriNV4TY18XKDOZhlfqvSYtV8vu0g4YS+ypj4qVE/FsaNtR6KDDJ0SEPtY9zs9vWl
HT26zYN7hGucHFbQJlhypv8k+w== santos@workstation
```

Task 4: Verify ssh remote connection

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



```
santos@workstation:~$ ssh santos@192.168.56.105
Last login: Tue Sep  5 13:06:26 2023 from 192.168.56.102
[santos@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?
   - For me there are some factors that we need to look for in choosing the best distribution between Debian and Red Hat Linux. First, is about the maintenance and support of that particular distribution. It's important to know if that certain distribution gives you updates, security patches, and also gives long-term support. Another one, is hardware architecture requirements that are needed and we need to see if that certain distribution supports your hardware since it's important to know the compatibility of it. Next, is the community support and documentation. Having a good amount and quality of community support will definitely help in finding answers and solutions when you encounter different problems on your system. Lastly, is the usage of your system, it is important to identify first the purpose of it to know that distribution fits your needs.

2. What are the main diffence between Debian and Red Hat Linux distributions?
   - One of the main differences between Debian and Red Hat Linux is its package management. Debian have Debian Package Management System(APT) and also the .deb package format, while the Red Hat have Red Hat Package Manager (RPM) and .rpm package format. Another difference is its target person that will use the distribution. Debian is used in general things and it makes it good in desktop, server, and also embedded system, while the Red Hat is focus more on enterprise marker and improving its security and stability.