Name: MANGUNDAYAO, PAULL ANDREI A.	**Date Performed:** 08/29/23
Course/Section: CPE31S4	**Date Submitted:** 09/04/23
Instructor: Dr. Jonathan Taylar	**Semester and SY:** First Semester - 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.

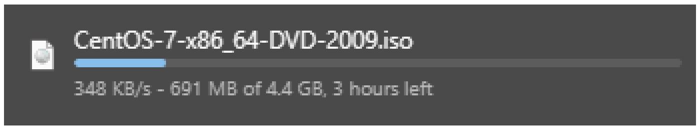
CentOS uses the RPM package format and YUM/DNF as the package manager.

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.

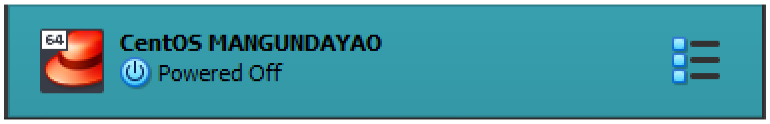
Task 1: Download the CentOS or RHEL-8 image (Create screenshots of the following)

1. Download the image of the CentOS here: http://mirror.rise.ph/centos/7.9.2009/isos/x86_64/

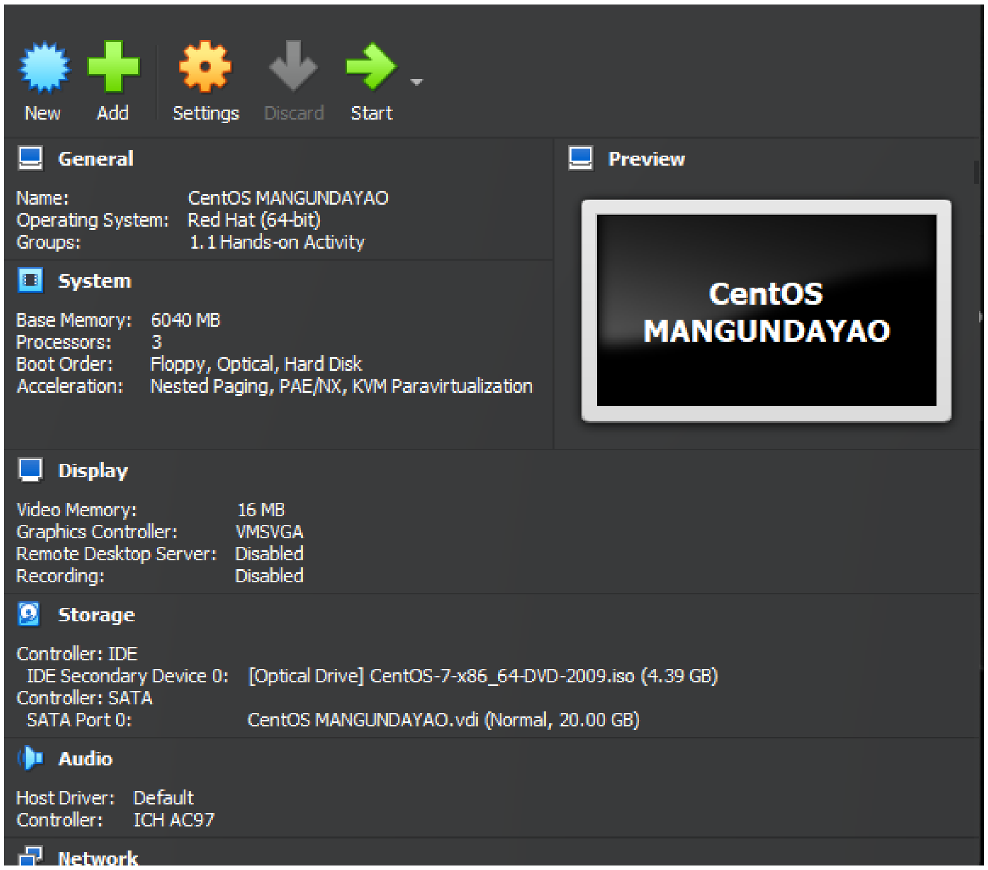


CentOS-7-x86_64-DVD-2009.iso
348 KB/s - 691 MB of 4.4 GB, 3 hours left

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.



CentOS MANGUNDAYAO
Powered Off



General

Name: CentOS MANGUNDAYAO
Operating System: Red Hat (64-bit)
Groups: 1.1 Hands-on Activity

System

Base Memory: 6040 MB
Processors: 3
Boot Order: Floppy, Optical, Hard Disk
Acceleration: Nested Paging, PAE/NX, KVM Paravirtualization

Display

Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

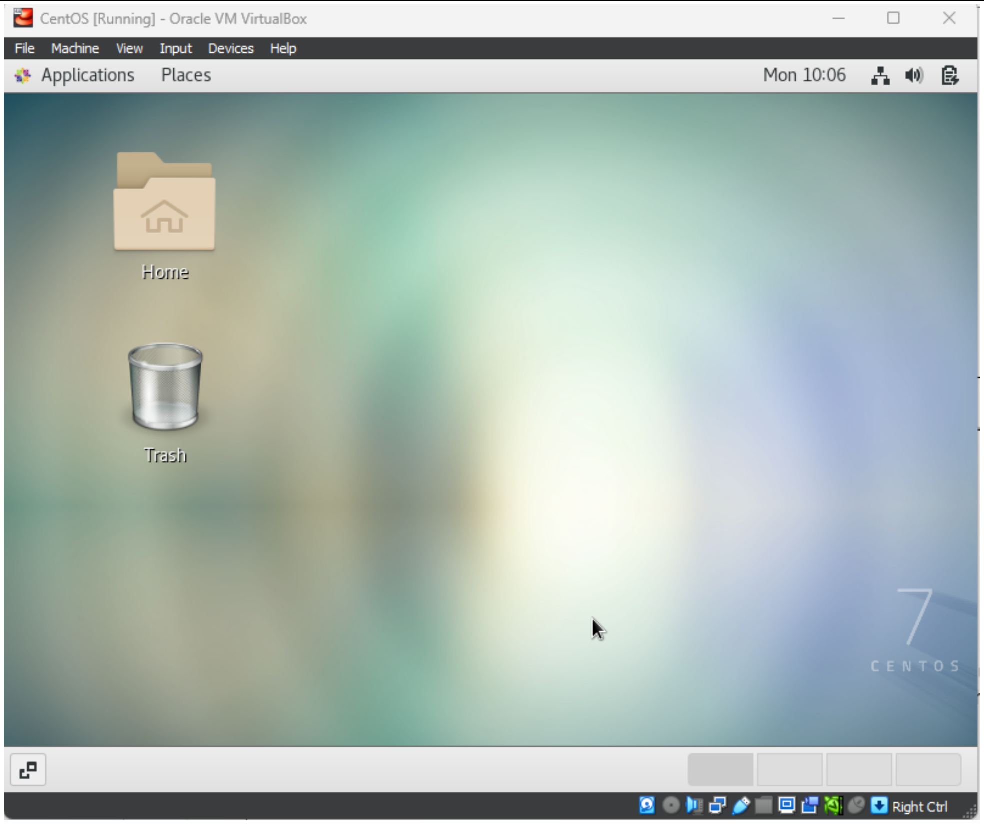
Storage

Controller: IDE
IDE Secondary Device 0: [Optical Drive] CentOS-7-x86_64-DVD-2009.iso (4.39 GB)
Controller: SATA
SATA Port 0: CentOS MANGUNDAYAO.vdi (Normal, 20.00 GB)

Audio

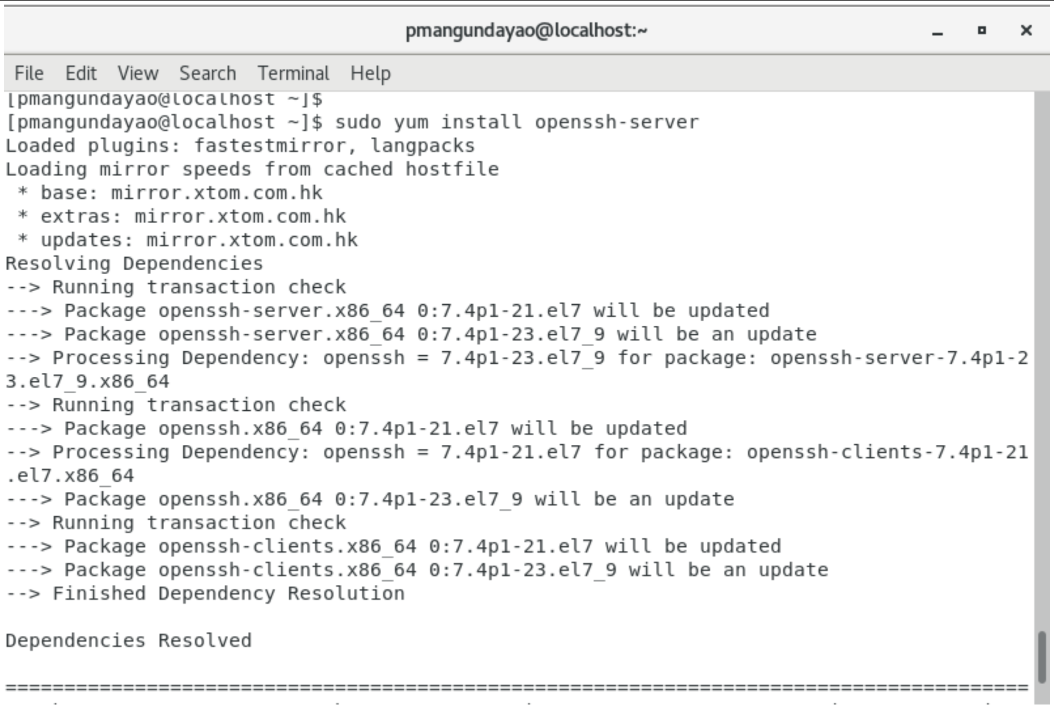
Host Driver: Default
Controller: ICH AC97

Network


CentOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places
Mon 10:06
Home
Trash
7 CENTOS
Right Ctrl

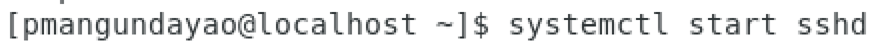
Task 2: Install the SSH server package *openssh*

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



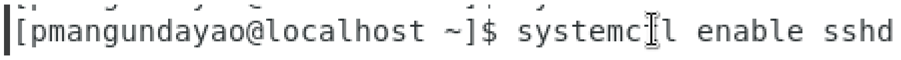
```
[pmangundayao@localhost ~]$
[pmangundayao@localhost ~]$ sudo yum install openssh-server
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirror.xtom.com.hk
 * extras: mirror.xtom.com.hk
 * updates: mirror.xtom.com.hk
Resolving Dependencies
--> Running transaction 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-7.4p1-2
3.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
---> Package openssh-clients.x86_64 0:7.4p1-23.el7_9 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

=========================================================================================
```

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



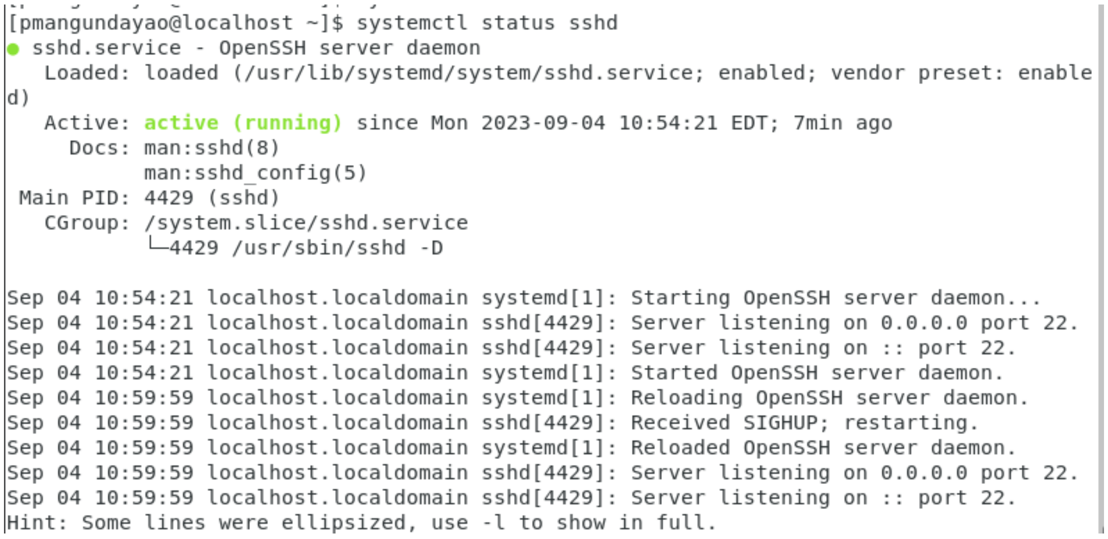
```
[pmangundayao@localhost ~]$ systemctl start sshd
```

$ systemctl enable sshd



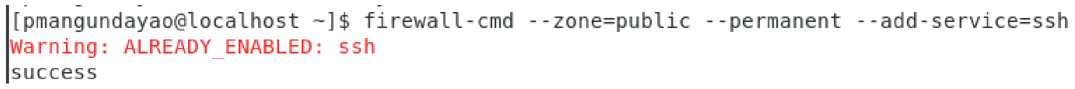
```
[pmangundayao@localhost ~]$ systemctl enable sshd
```

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



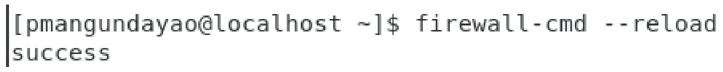
```
[pmangundayao@localhost ~]$ systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enable
d)
   Active: active (running) since Mon 2023-09-04 10:54:21 EDT; 7min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 4429 (sshd)
   CGroup: /system.slice/sshd.service
           └─4429 /usr/sbin/sshd -D

Sep 04 10:54:21 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Sep 04 10:54:21 localhost.localdomain sshd[4429]: Server listening on 0.0.0.0 port 22.
Sep 04 10:54:21 localhost.localdomain sshd[4429]: Server listening on :: port 22.
Sep 04 10:54:21 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
Sep 04 10:59:59 localhost.localdomain systemd[1]: Reloading OpenSSH server daemon.
Sep 04 10:59:59 localhost.localdomain sshd[4429]: Received SIGHUP; restarting.
Sep 04 10:59:59 localhost.localdomain systemd[1]: Reloaded OpenSSH server daemon.
Sep 04 10:59:59 localhost.localdomain sshd[4429]: Server listening on 0.0.0.0 port 22.
Sep 04 10:59:59 localhost.localdomain sshd[4429]: Server listening on :: port 22.
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



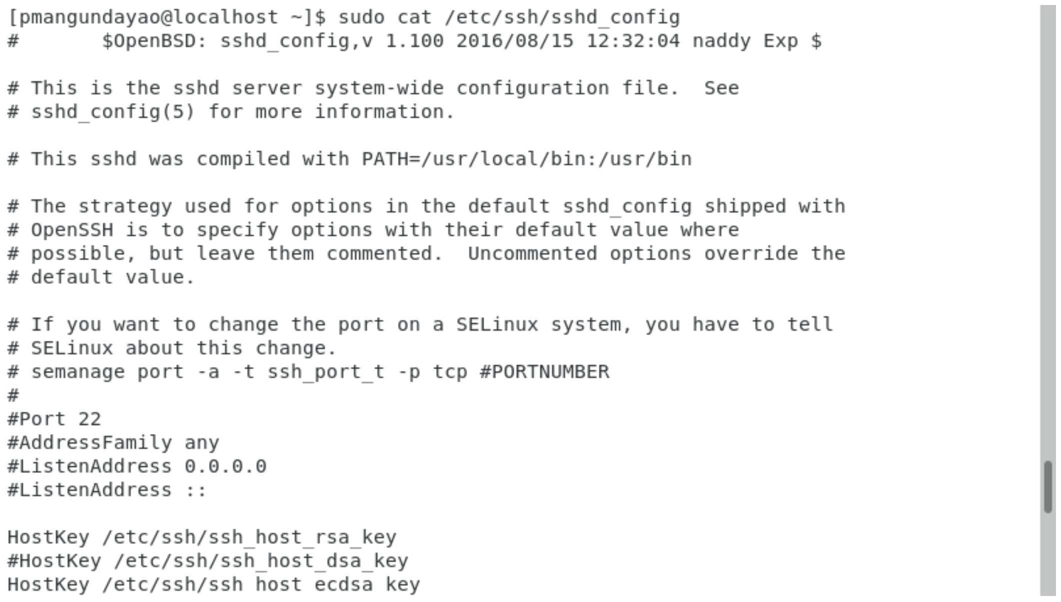
```
[pmangundayao@localhost ~]$ firewall-cmd --zone=public --permanent --add-service=ssh
Warning: ALREADY_ENABLED: ssh
success
```

   $ firewall-cmd --reload

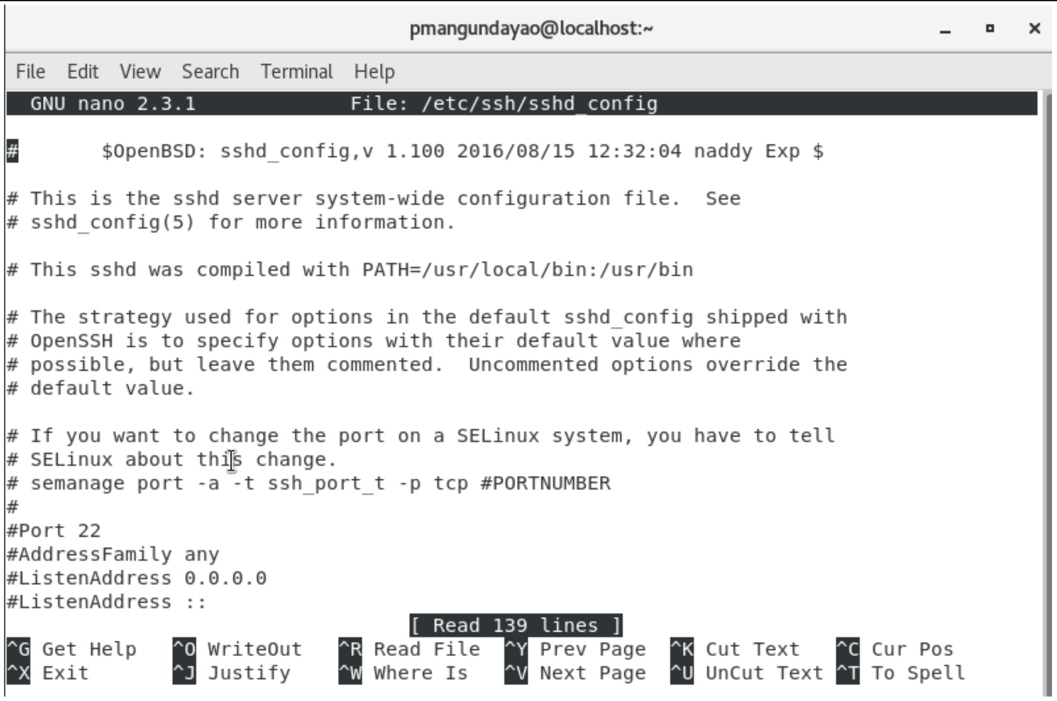


```
[pmangundayao@localhost ~]$ firewall-cmd --reload
success
```

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



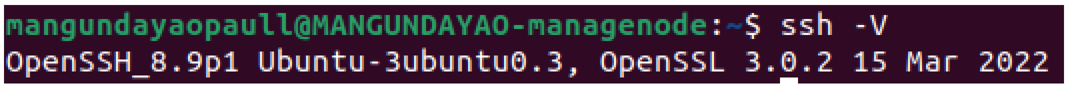
```
[pmangundayao@localhost ~]$ sudo cat /etc/ssh/sshd_config
#       $OpenBSD: sshd_config,v 1.100 2016/08/15 12:32:04 naddy Exp $

# 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 ::

HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_dsa_key
HostKey /etc/ssh/ssh host ecdsa key
```



```
pmangundayao@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 ::
                                   [ 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
```

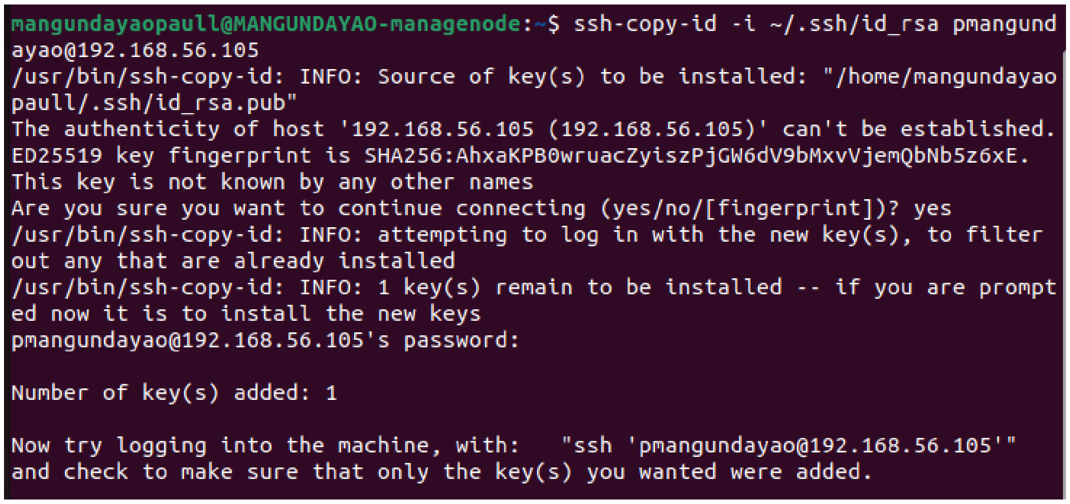
Task 3: Copy the Public Key to CentOS

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



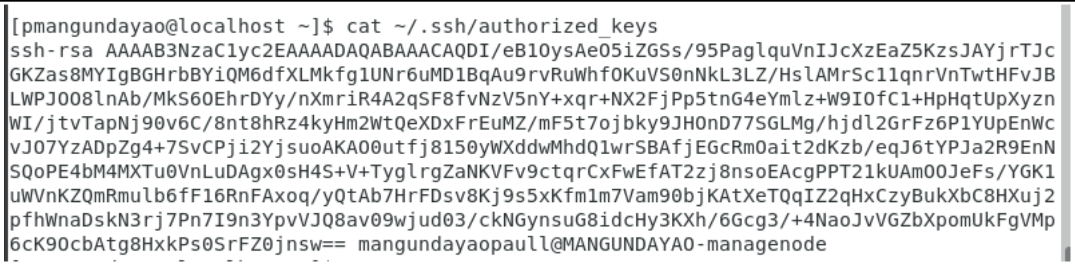
```
mangundayaopaull@MANGUNDAYAO-managenode:~$ ssh -V
OpenSSH_8.9p1 Ubuntu-3ubuntu0.3, OpenSSL 3.0.2 15 Mar 2022
```

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



```
mangundayaopaull@MANGUNDAYAO-managenode:~$ ssh-copy-id -i ~/.ssh/id_rsa pmangund
ayao@192.168.56.105
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/mangundayao
paull/.ssh/id_rsa.pub"
The authenticity of host '192.168.56.105 (192.168.56.105)' can't be established.
ED25519 key fingerprint is SHA256:AhxaKPB0wruacZyiszPjGW6dV9bMxvVjemQbNb5z6xE.
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
pmangundayao@192.168.56.105's password: 

Number of key(s) added: 1

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

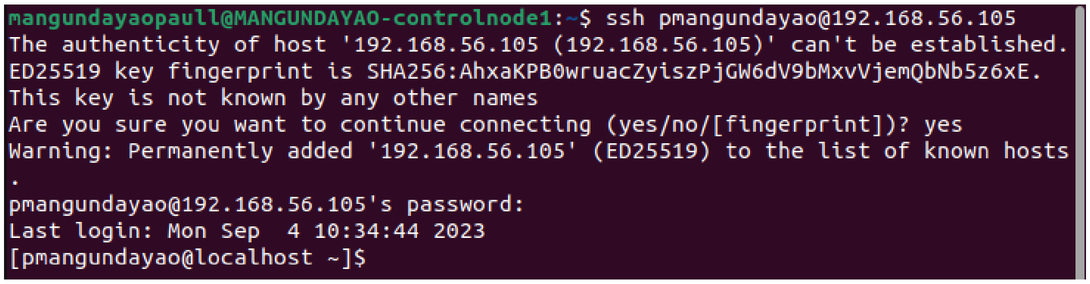
3. On CentOS, verify that you have the *authorized_keys*.



```
[pmangundayao@localhost ~]$ cat ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDI/eB1OysAeO5iZGSs/95PaglquVnIJcXzEaZ5KzsJAYjrTJc
GKZas8MYIgBGHrbBYiQM6dfXLMkfg1UNr6uMD1BqAu9rvRuWhfOKuVS0nNkL3LZ/HslAMrSc11qnrVnTwtHFvJB
LWPJOO8lnAb/MkS6OEhrDYy/nXmriR4A2qSF8fvNzV5nY+xqr+NX2FjPp5tnG4eYmlz+W9IOfC1+HpHqtUpXyzn
WI/jtvTapNj90v6C/8nt8hRz4kyHm2WtQeXDxFrEuMZ/mF5t7ojbky9JHOnD77SGLMg/hjdl2GrFz6P1YUpEnWc
vJO7YzADpZg4+7SvCPji2YjsuoAKAO0utfj8150yWXddwMhdQ1wrSBAfjEGcRmOait2dKzb/eqJ6tYPJa2R9EnN
SQoPE4bM4MXTu0VnLuDAgx0sH4S+V+TyglrgZaNKVFv9ctqrCxFwEfAT2zj8nsoEAcgPPT21kUAmOOJeFs/YGK1
uWVnKZQmRmulb6fF16RnFAxoq/yQtAb7HrFDsv8Kj9s5xKfm1m7Vam90bjKAtXeTQqIZ2qHxCzyBukXbC8HXuj2
pfhWnaDskN3rj7Pn7I9n3YpvVJQ8av09wjud03/ckNGynsuG8idcHy3KXh/6Gcg3/+4NaoJvVGZbXpomUkFgVMp
6cK9OcbAtg8HxkPs0SrFZ0jnsw== mangundayaopaull@MANGUNDAYAO-managenode
```

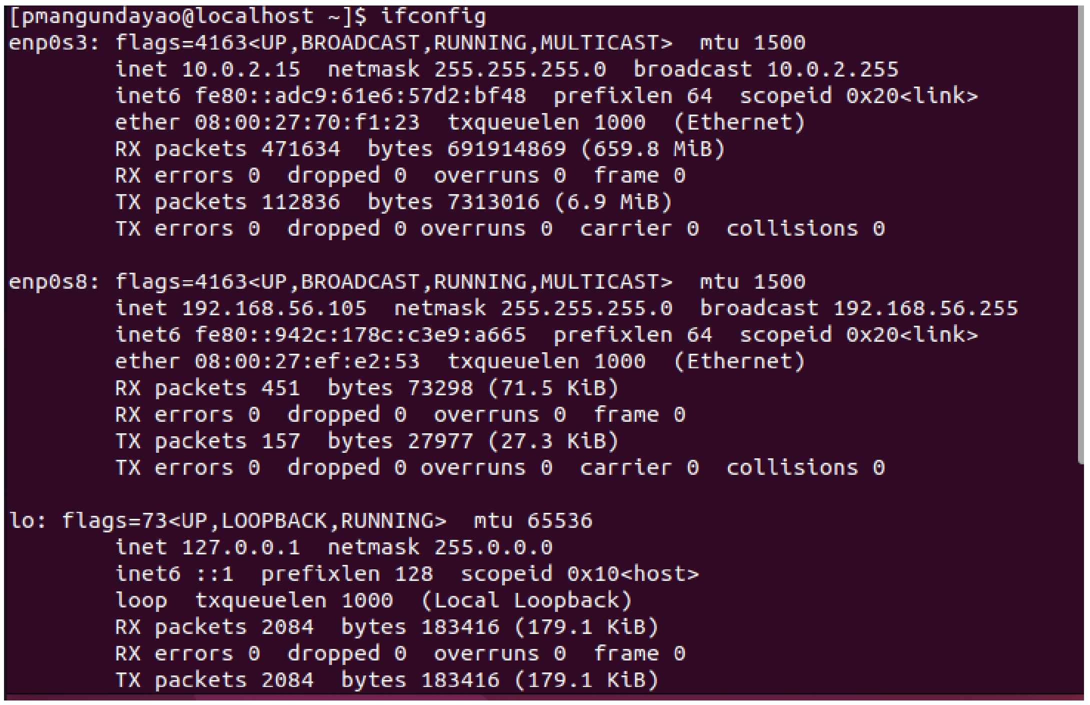
Task 4: Verify ssh remote connection

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



```
mangundayaopaull@MANGUNDAYAO-controlnode1:~$ ssh pmangundayao@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:AhxaKPB0wruacZyiszPjGW6dV9bMxvVjemQbNb5z6xE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.105' (ED25519) to the list of known hosts
.
pmangundayao@192.168.56.105's password:
Last login: Mon Sep  4 10:34:44 2023
[pmangundayao@localhost ~]$
```

2. Show evidence that you are connected.



```
[pmangundayao@localhost ~]$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::adc9:61e6:57d2:bf48  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:70:f1:23  txqueuelen 1000  (Ethernet)
        RX packets 471634  bytes 691914869 (659.8 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 112836  bytes 7313016 (6.9 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.105  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::942c:178c:c3e9:a665  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:ef:e2:53  txqueuelen 1000  (Ethernet)
        RX packets 451  bytes 73298 (71.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 157  bytes 27977 (27.3 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 2084  bytes 183416 (179.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2084  bytes 183416 (179.1 KiB)
```

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?

You should think about your priorities while choosing between Linux distributions based on Debian and Red Hat. Debian is a fantastic option for servers and production situations where you need a rock-solid system because of its well-known stability and dependability. It relies on a strong community, with no official paid support, but third-party companies offer support options. Debian uses the APT package manager, known for its user-friendliness.

On the other side, Red Hat distributions like CentOS and Fedora frequently offer more recent software and capabilities and are more cutting edge. Although it may not be the best option for mission-critical environments because to the frequent changes, this is perfect for users looking for the newest innovations. Red Hat Enterprise Linux (RHEL) comes with official paid support from Red Hat Inc., making it suitable for enterprises. CentOS offers community support, while Fedora is community-driven. Red Hat systems use YUM/DNF package managers, which are also user-friendly. Additionally, Red Hat has a strong enterprise focus, whereas Debian is community-driven, with a broader range of hardware support. Your choice should align with your specific needs, whether it's stability, support, licensing, or community involvement. Testing both distributions in your environment is a practical approach to determine which one suits you best.

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

   Debian prioritizes stability, making it ideal for reliable servers, while Red Hat-based distributions offer the latest features but may require careful updates. Debian lacks official paid support but has a robust community and uses APT for package management. Red Hat options, including RHEL with paid support and CentOS with community support, provide newer software via YUM/DNF package managers. Your choice should align with your stability, support, and software needs.

Conclusion:

In this activity, I have successfully installed a Red Hat Linux distribution, specifically CentOS. I also integrated the knowledge I had in the previous activities to connect the said OS to a server and integrate it to the said server. Once installed, configuring SSH for remote access allows you to manage your system efficiently from remote computers, enhancing accessibility and control. This secure connection method is essential for system administrators, developers, and users who need to remotely manage and maintain their CentOS or RHEL 8 systems.

By following proper installation and SSH configuration procedures, you can establish a reliable and secure remote connection to your Red Hat-based Linux system, ensuring seamless remote management and administration.