Name: Denzel A. Dinglasan	**Date Performed: 07/09/2023**
Course/Section:CPE232-CPE31S6	**Date Submitted: 07/09/2023**
Instructor: Dr. Jonathan Vidal Taylar	**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.

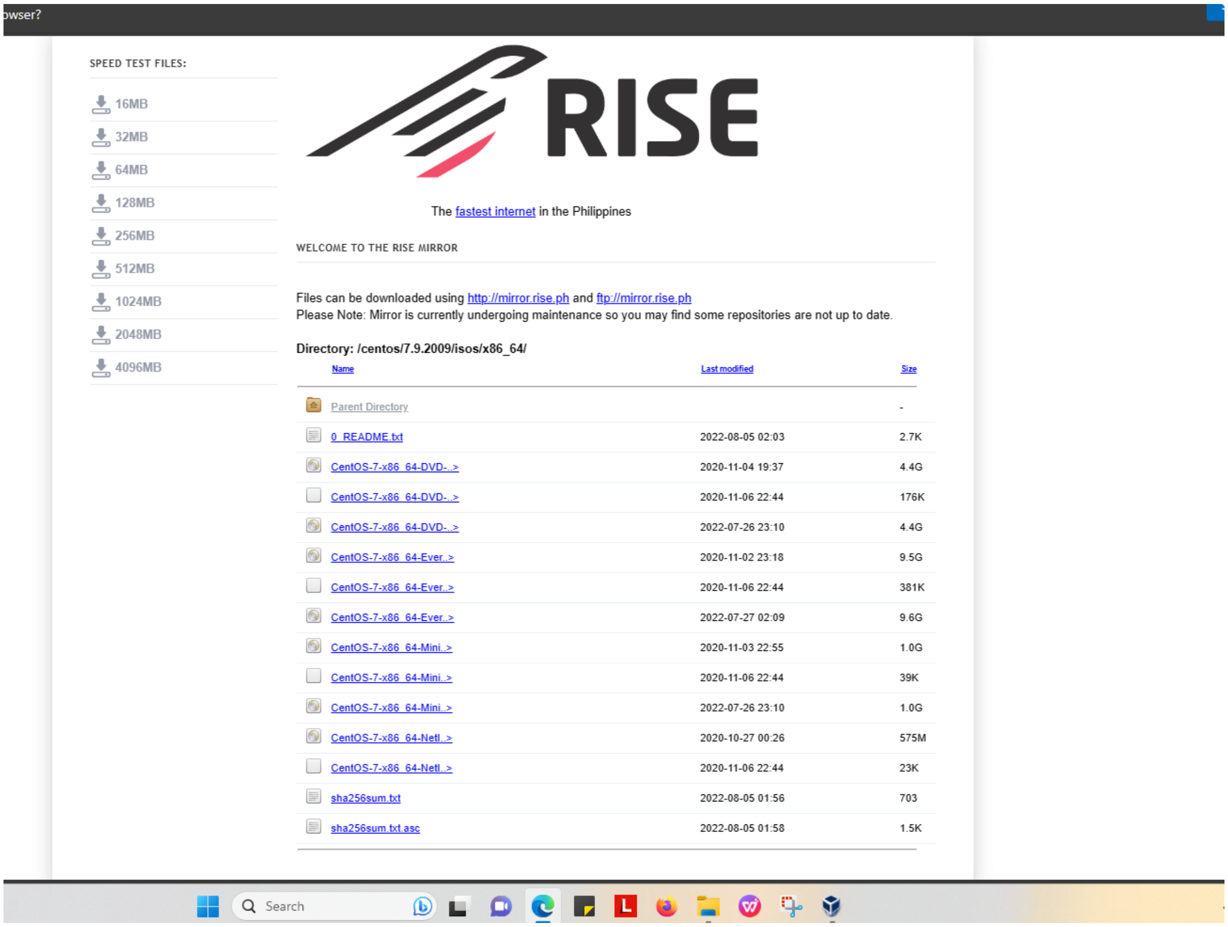
CentOS uses the RPM package format and YUM/DNF as the package manager.

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/



The fastest internet in the Philippines

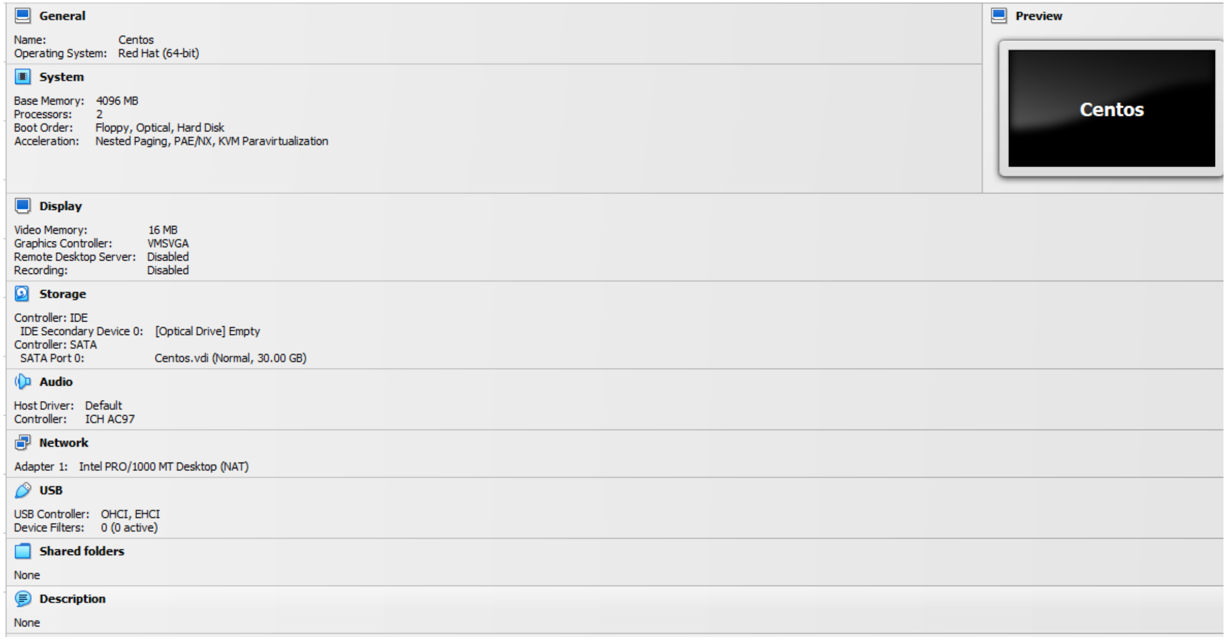
WELCOME TO THE RISE MIRROR

Files can be downloaded using http://mirror.rise.ph and ftp://mirror.rise.ph
Please Note: Mirror is currently undergoing maintenance so you may find some repositories are not up to date.

Directory: /centos/7.9.2009/isos/x86_64/

Name	Last modified	Size
Parent Directory		-
0_README.txt	2022-08-05 02:03	2.7K
CentOS-7-x86_64-DVD-..>	2020-11-04 19:37	4.4G
CentOS-7-x86_64-DVD-..>	2020-11-06 22:44	176K
CentOS-7-x86_64-DVD-..>	2022-07-26 23:10	4.4G
CentOS-7-x86_64-Ever..>	2020-11-02 23:18	9.5G
CentOS-7-x86_64-Ever..>	2020-11-06 22:44	381K
CentOS-7-x86_64-Ever..>	2022-07-27 02:09	9.6G
CentOS-7-x86_64-Mini..>	2020-11-03 22:55	1.0G
CentOS-7-x86_64-Mini..>	2020-11-06 22:44	39K
CentOS-7-x86_64-Mini..>	2022-07-26 23:10	1.0G
CentOS-7-x86_64-Netl..>	2020-10-27 00:26	575M
CentOS-7-x86_64-Netl..>	2020-11-06 22:44	23K
sha256sum.txt	2022-08-05 01:56	703
sha256sum.txt.asc	2022-08-05 01:58	1.5K

2. Create a VM machine with 2 Gb RAM and 20 Gb HD.



Name: Centos
Operating System: Red Hat (64-bit)

Base Memory: 4096 MB
Processors: 2
Boot Order: Floppy, Optical, Hard Disk
Acceleration: Nested Paging, PAE/NX, KVM Paravirtualization

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

Controller: IDE
IDE Secondary Device 0: [Optical Drive] Empty
Controller: SATA
SATA Port 0: Centos.vdi (Normal, 30.00 GB)

Host Driver: Default
Controller: ICH AC97

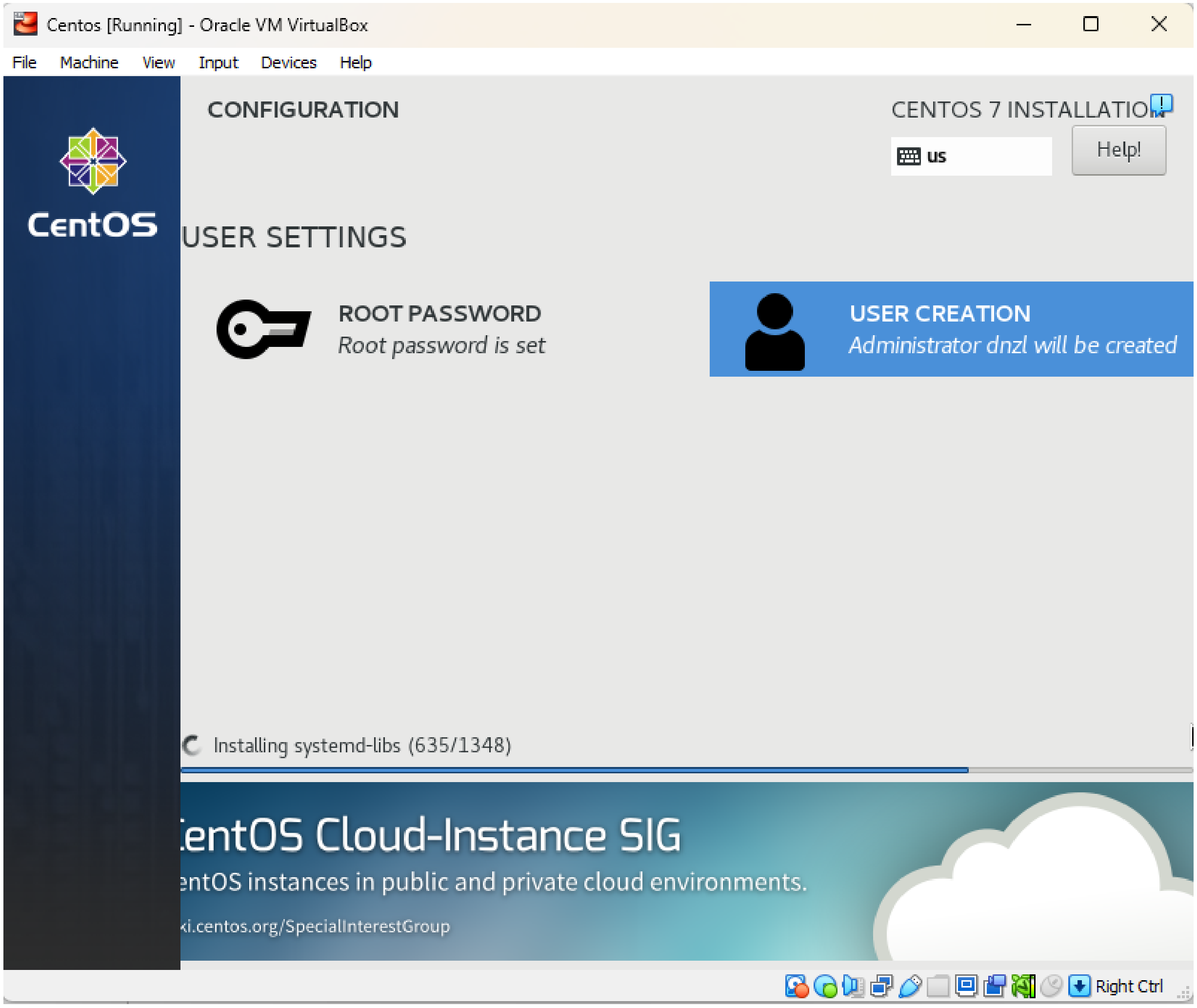
Adapter 1: Intel PRO/1000 MT Desktop (NAT)

USB Controller: OHCI, EHCI
Device Filters: 0 (0 active)

None

None

3. Install the downloaded image.



Centos [Running] - Oracle VM VirtualBox
File
Machine
View
Input
Devices
Help
CentOS
us
Help!
Right Ctrl

CONFIGURATION

CENTOS 7 INSTALLATION

USER SETTINGS

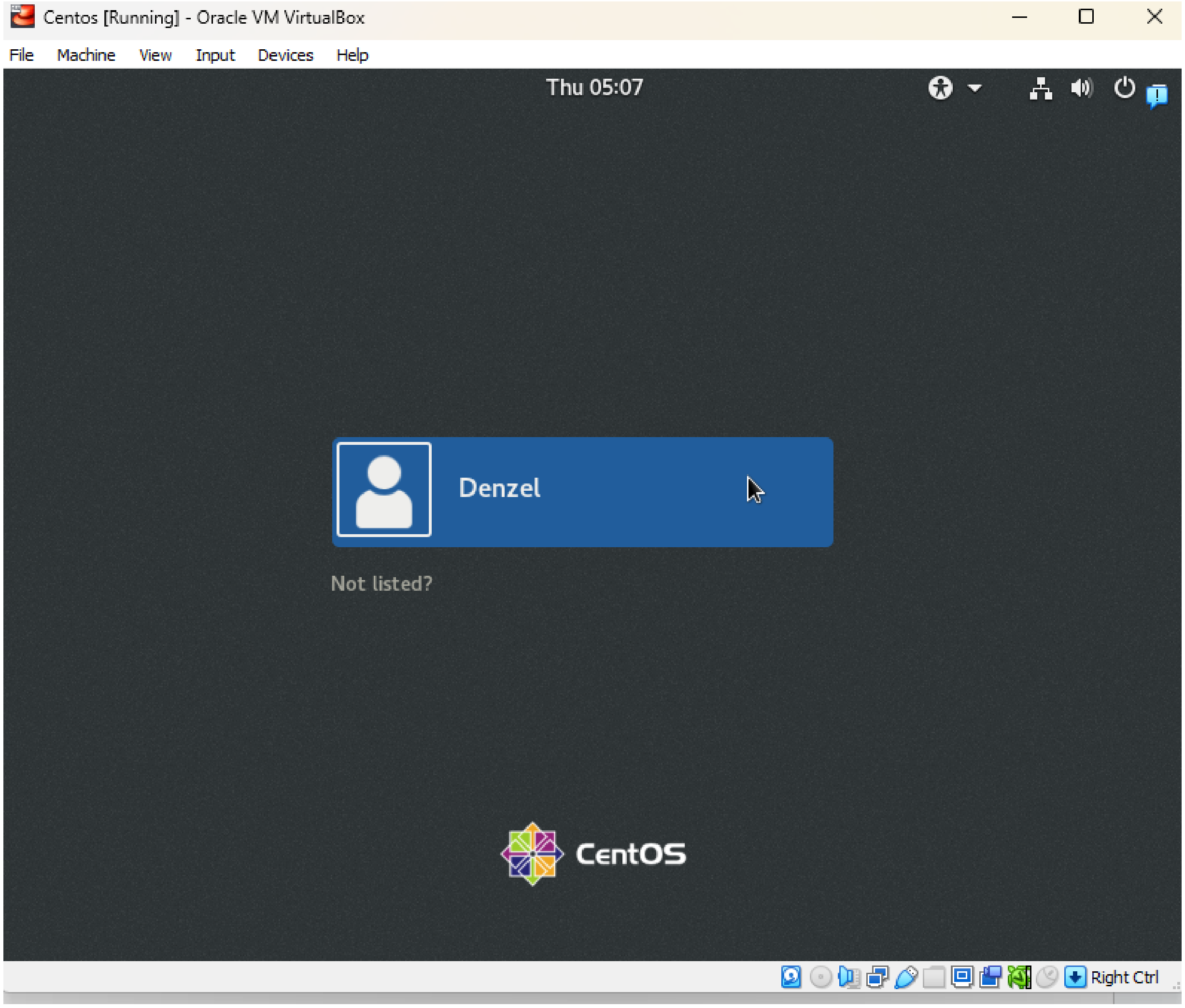
ROOT PASSWORD
Root password is set

USER CREATION
Administrator dnzl will be created

Installing systemd-libs (635/1348)

entOS Cloud-Instance SIG
entOS instances in public and private cloud environments.
ki.centos.org/SpecialInterestGroup

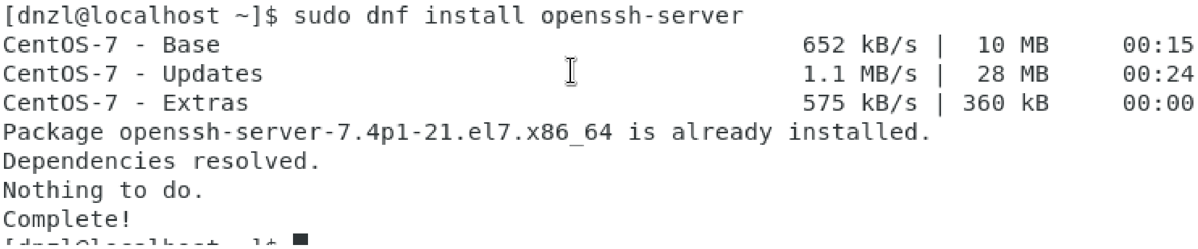
4. Show evidence that the OS was installed already.



Task 2: Install the SSH server package *openssh*

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

   $ dnf install openssh-server

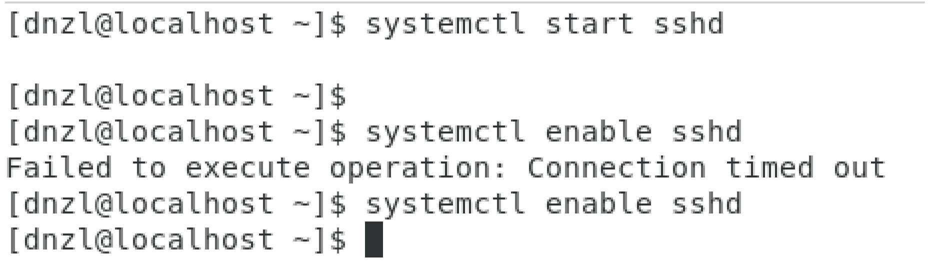


```
[dnzl@localhost ~]$ sudo dnf install openssh-server
CentOS-7 - Base                                    652 kB/s |  10 MB     00:15
CentOS-7 - Updates                                 1.1 MB/s |  28 MB     00:24
CentOS-7 - Extras                                  575 kB/s | 360 kB     00:00
Package openssh-server-7.4p1-21.el7.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[dnzl@localhost ~]$ █
```

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

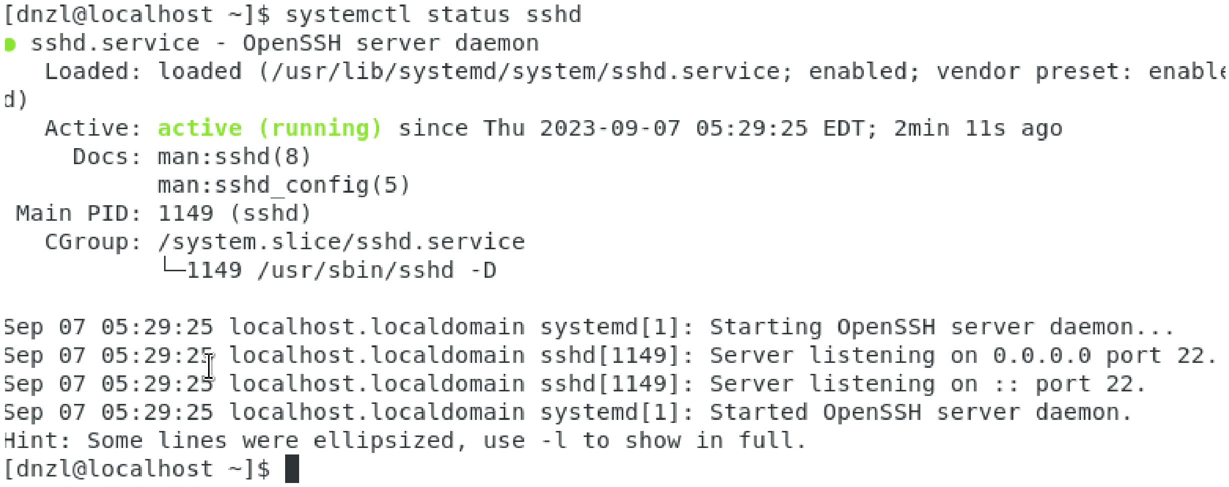
   $ systemctl start sshd

   $ systemctl enable sshd



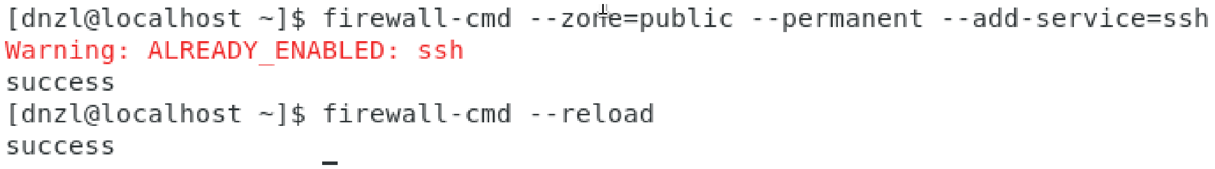
```
[dnzl@localhost ~]$ systemctl start sshd

[dnzl@localhost ~]$
[dnzl@localhost ~]$ systemctl enable sshd
Failed to execute operation: Connection timed out
[dnzl@localhost ~]$ systemctl enable sshd
[dnzl@localhost ~]$ 
```

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



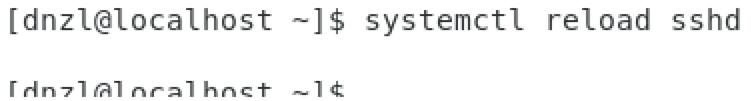
```
[dnzl@localhost ~]$ systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enable
d)
   Active: active (running) since Thu 2023-09-07 05:29:25 EDT; 2min 11s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 1149 (sshd)
   CGroup: /system.slice/sshd.service
           └─1149 /usr/sbin/sshd -D

Sep 07 05:29:25 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Sep 07 05:29:25 localhost.localdomain sshd[1149]: Server listening on 0.0.0.0 port 22.
Sep 07 05:29:25 localhost.localdomain sshd[1149]: Server listening on :: port 22.
Sep 07 05:29:25 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
Hint: Some lines were ellipsized, use -l to show in full.
[dnzl@localhost ~]$ 
```

4. Open the SSH port 22 to allow incoming traffic:
   $ firewall-cmd --zone=public --permanent --add-service=ssh
   $ firewall-cmd --reload



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

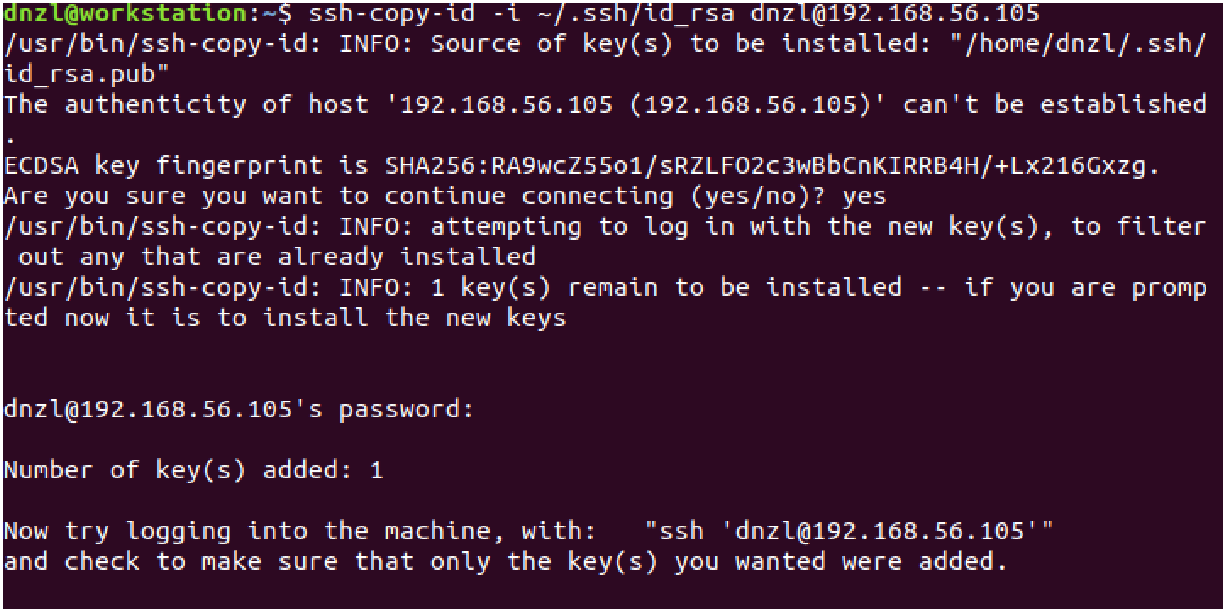
5. Locate the ssh server man config file /*etc/ssh/sshd_config* and perform custom configuration. Every time you make any change to the /*etc/ssh/sshd-config* configuration file reload the *sshd* service to apply changes:
   $ systemctl reload sshd



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

[dnzl@localhost ~]$
```

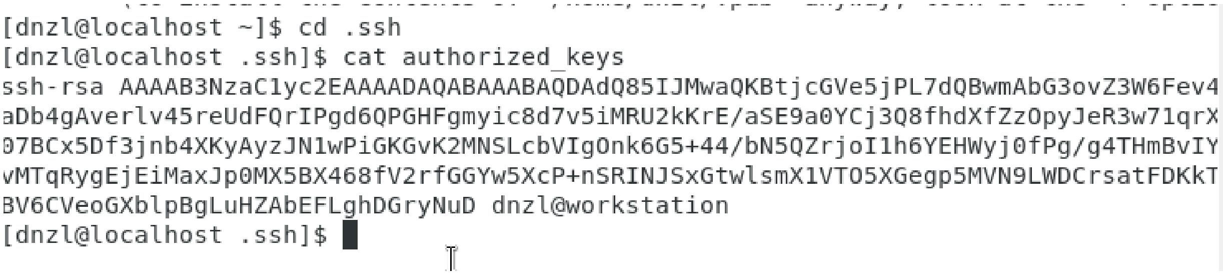
Task 3: Copy the Public Key to CentOS

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



```
dnzl@workstation:~$ ssh-copy-id -i ~/.ssh/id_rsa dnzl@192.168.56.105
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/dnzl/.ssh/
id_rsa.pub"
The authenticity of host '192.168.56.105 (192.168.56.105)' can't be established
.
ECDSA key fingerprint is SHA256:RA9wcZ55o1/sRZLFO2c3wBbCnKIRRB4H/+Lx216Gxzg.
Are you sure you want to continue connecting (yes/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: 1 key(s) remain to be installed -- if you are promp
ted now it is to install the new keys


dnzl@192.168.56.105's password: 

Number of key(s) added: 1

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

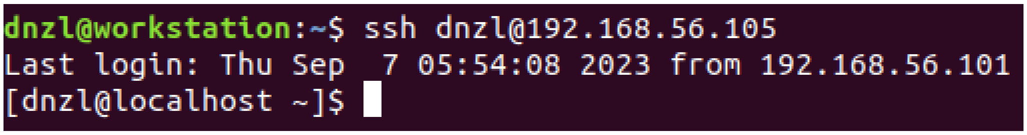
3. On CentOS, verify that you have the *authorized_keys*.



```
[dnzl@localhost ~]$ cd .ssh
[dnzl@localhost .ssh]$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDAdQ85IJMwaQKBtjcGVe5jPL7dQBwmAbG3ovZ3W6Fev4
aDb4gAverlv45reUdFQrIPgd6QPGHFgmyic8d7v5iMRU2kKrE/aSE9a0YCj3Q8fhdXfZzOpyJeR3w71qrX
07BCx5Df3jnb4XKyAyzJN1wPiGKGvK2MNSLcbVIgOnk6G5+44/bN5QZrjoI1h6YEHWyj0fPg/g4THmBvIY
vMTqRygEjEiMaxJp0MX5BX468fV2rfGGYw5XcP+nSRINJSxGtwlsmX1VTO5XGegp5MVN9LWDCrsatFDKkT
BV6CVeoGXblpBgLuHZAbEFLghDGryNuD dnzl@workstation
[dnzl@localhost .ssh]$ 
```

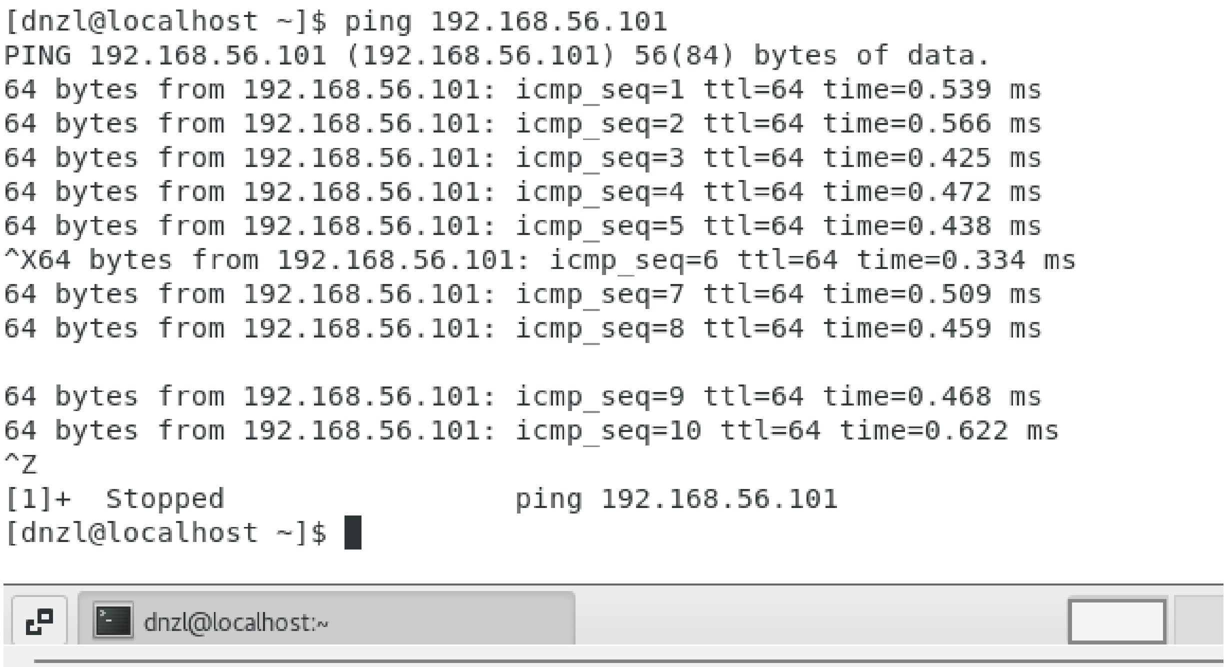
Task 4: Verify ssh remote connection

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

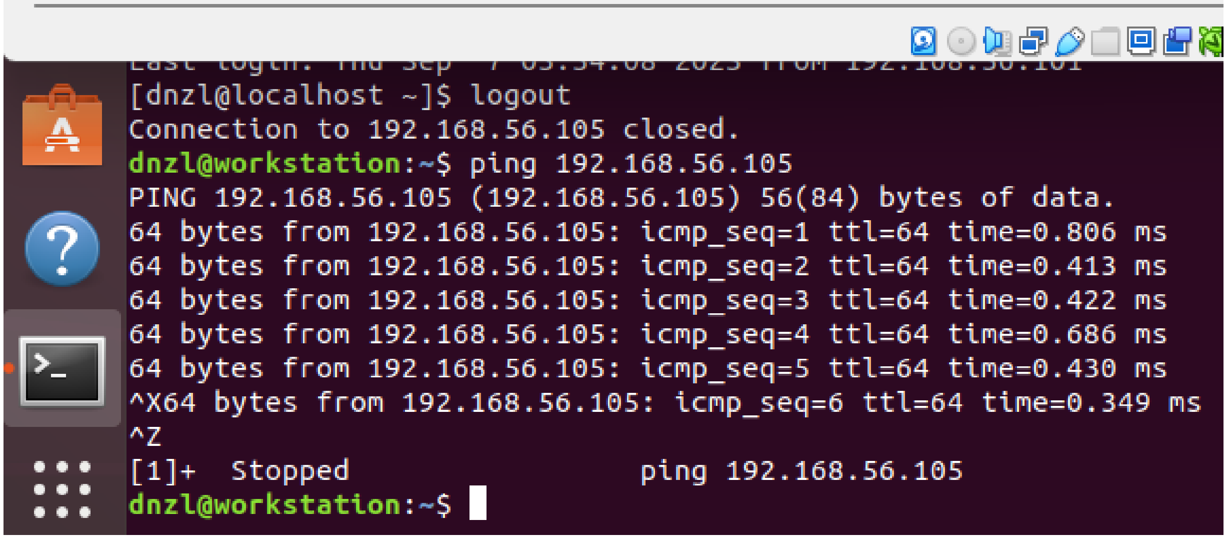


```
dnzl@workstation:~$ ssh dnzl@192.168.56.105
Last login: Thu Sep  7 05:54:08 2023 from 192.168.56.101
[dnzl@localhost ~]$ 
```

2. Show evidence that you are connected.



```
[dnzl@localhost ~]$ ping 192.168.56.101
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data.
64 bytes from 192.168.56.101: icmp_seq=1 ttl=64 time=0.539 ms
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=0.566 ms
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=0.425 ms
64 bytes from 192.168.56.101: icmp_seq=4 ttl=64 time=0.472 ms
64 bytes from 192.168.56.101: icmp_seq=5 ttl=64 time=0.438 ms
^X64 bytes from 192.168.56.101: icmp_seq=6 ttl=64 time=0.334 ms
64 bytes from 192.168.56.101: icmp_seq=7 ttl=64 time=0.509 ms
64 bytes from 192.168.56.101: icmp_seq=8 ttl=64 time=0.459 ms

64 bytes from 192.168.56.101: icmp_seq=9 ttl=64 time=0.468 ms
64 bytes from 192.168.56.101: icmp_seq=10 ttl=64 time=0.622 ms
^Z
[1]+  Stopped                 ping 192.168.56.101
[dnzl@localhost ~]$
```



```
Last login: Thu Sep  7 05:34:08 2023 from 192.168.56.101
[dnzl@localhost ~]$ logout
Connection to 192.168.56.105 closed.
dnzl@workstation:~$ ping 192.168.56.105
PING 192.168.56.105 (192.168.56.105) 56(84) bytes of data.
64 bytes from 192.168.56.105: icmp_seq=1 ttl=64 time=0.806 ms
64 bytes from 192.168.56.105: icmp_seq=2 ttl=64 time=0.413 ms
64 bytes from 192.168.56.105: icmp_seq=3 ttl=64 time=0.422 ms
64 bytes from 192.168.56.105: icmp_seq=4 ttl=64 time=0.686 ms
64 bytes from 192.168.56.105: icmp_seq=5 ttl=64 time=0.430 ms
^X64 bytes from 192.168.56.105: icmp_seq=6 ttl=64 time=0.349 ms
^Z
[1]+  Stopped                 ping 192.168.56.105
dnzl@workstation:~$
```

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?

   Choose Debian for stability, strong community support, and a focus on free and open-source software. Opt for Red Hat if you prioritize commercial support, security, and a balance between stability and newer features, but be mindful of potential licensing costs with Red Hat Enterprise Linux (RHEL).

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

   Debian is a community-driven distribution known for its emphasis on free and open-source software, versatile package management, and long-term stability. Red Hat, on the other hand, offers a mix of open-source and commercial options, prioritizes enterprise support, security, and predictable release cycles, often with associated licensing costs.

Conclusion:

In this activity, I’ve installed CentOS and set its configuration with SSH. I’ve also connected CentOS and my workstation in Ubuntu with the help of SSH.