Name: Nanquil, Nealian Beth B.	**Date Performed:** September 02, 2022
Course/Section: CPE232-CPE31S23	**Date Submitted:** September 02, 2022
Instructor: Engr, Jonathan Taylar	**Semester and SY:** 1st Sem – 2022-2023

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.

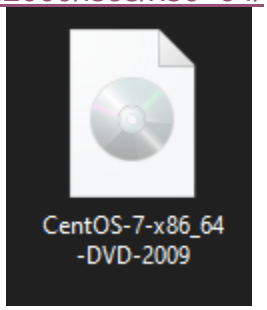
CentOS uses the RPM package format and YUM/DNF as the package manager.

Debian uses the DEB package format and dpkg/APT as the package manager.

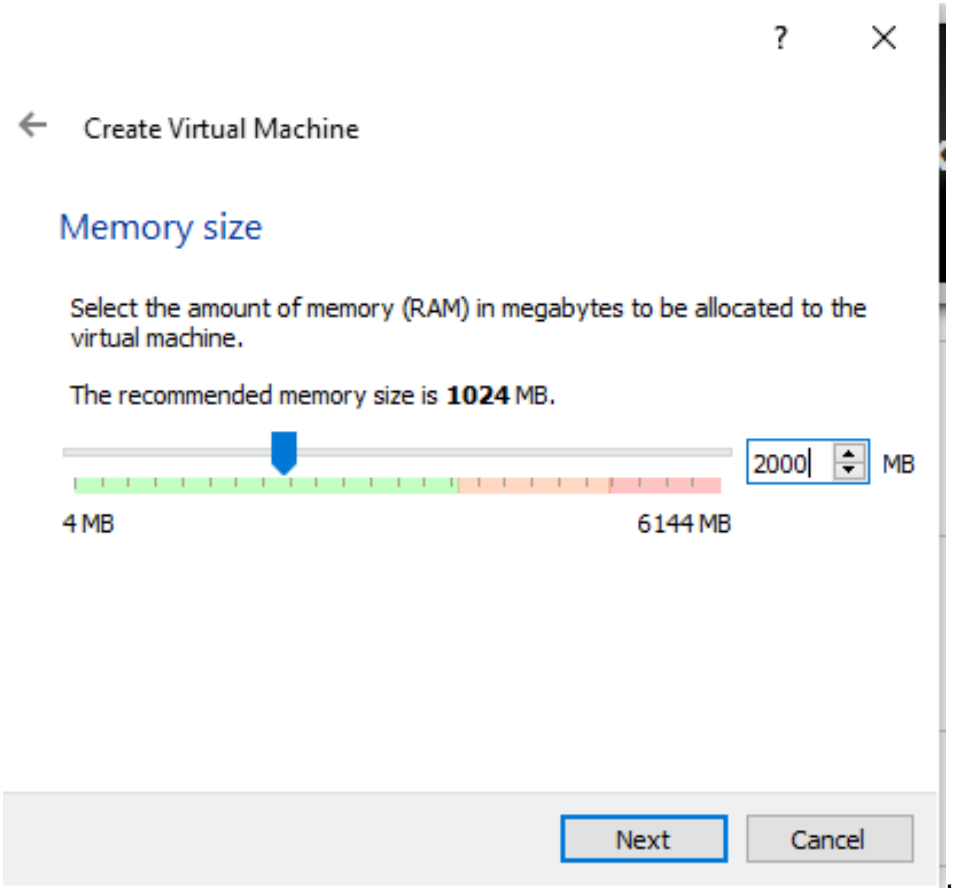
Both offer full-feature package management with network-based repository support, dependency checking and resolution, etc.. If you're familiar with one but not the other, you may have a little trouble switching over, but they're not overwhelmingly different. They both have similar features, just available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

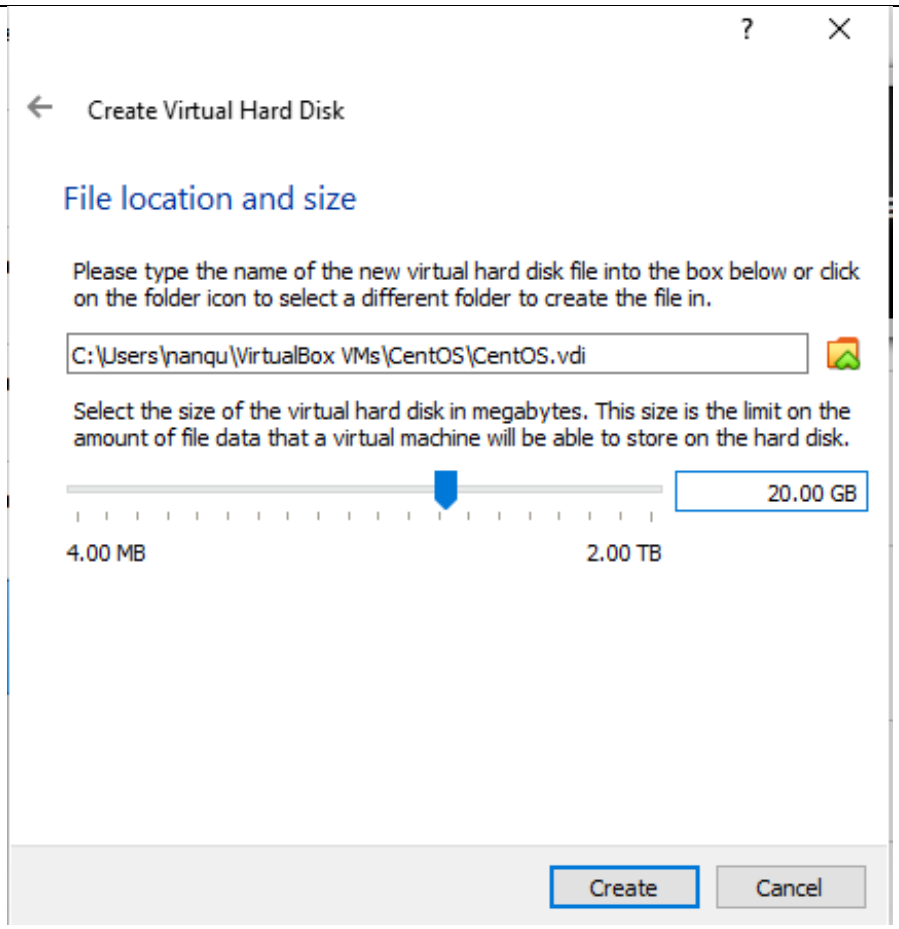
Task 1: Download the CentOS or RHEL-8 image (Create screenshots of the following)

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

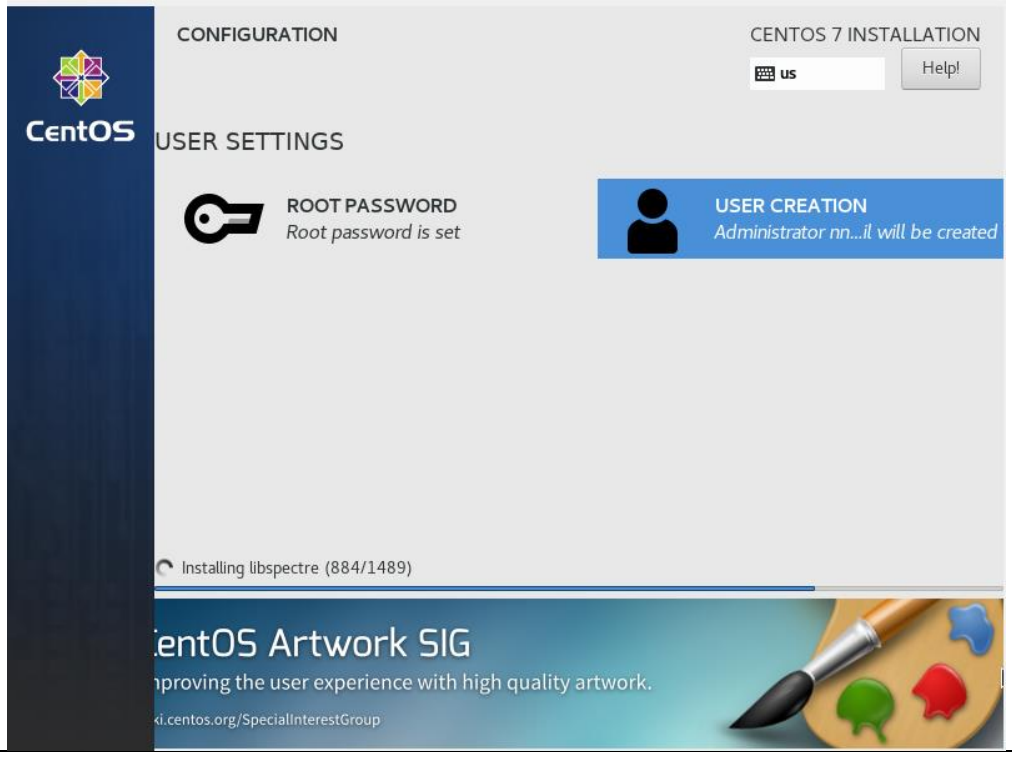


2. Create a VM machine with 2 Gb RAM and 20 Gb HD.

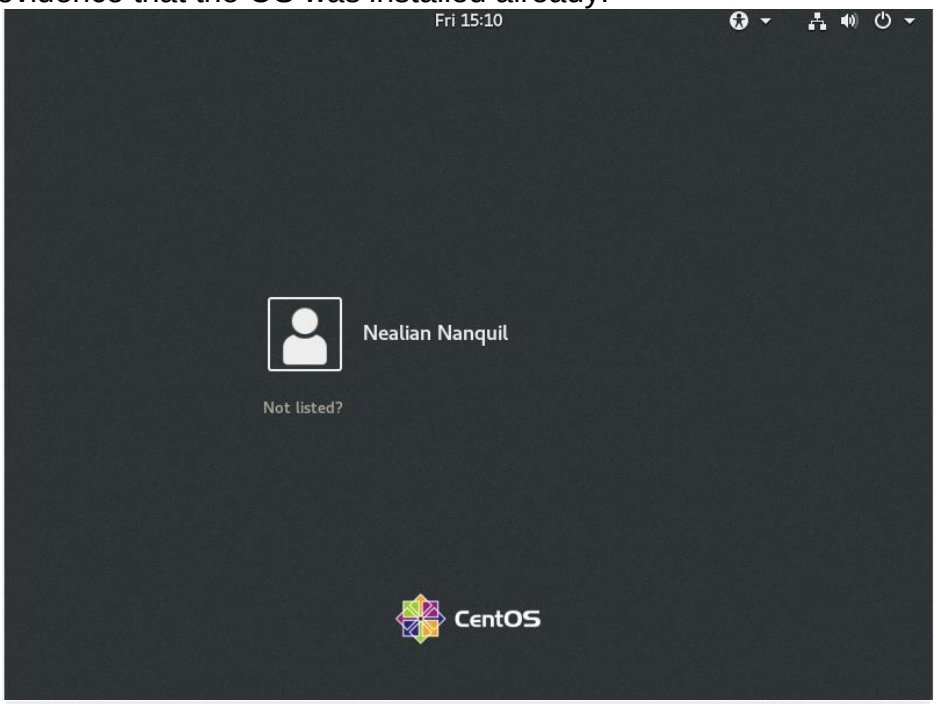


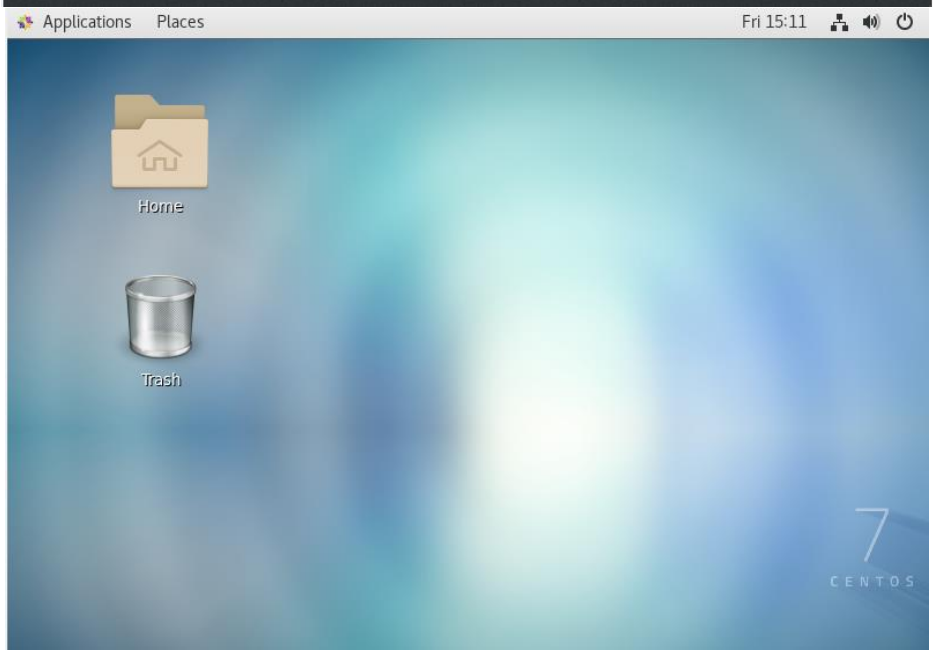


3. Install the downloaded image.



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

```
[nnanquil@localhost ~]$ dnf install openssh-server
Error: This command has to be run under the root user.
[nnanquil@localhost ~]$ sudo dnf install openssh-server
Extra Packages for Enterprise Linux 7 - x86_64          10 MB/s |  17 MB     00:01
CentOS-7 - Base                                        4.3 MB/s |  10 MB     00:02
CentOS-7 - Updates                                      15 MB/s |  21 MB     00:01
CentOS-7 - Extras                                      2.2 MB/s | 332 kB     00:00
Last metadata expiration check: 0:00:01 ago on Fri 02 Sep 2022 03:36:46 PM PST.
Package openssh-server-7.4p1-22.el7_9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[nnanquil@localhost ~]$
```

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

```
[nnanquil@localhost ~]$ sudo systemctl start sshd
```

   $ systemctl enable sshd

```
[nnanquil@localhost ~]$ sudo systemctl enable sshd
```

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

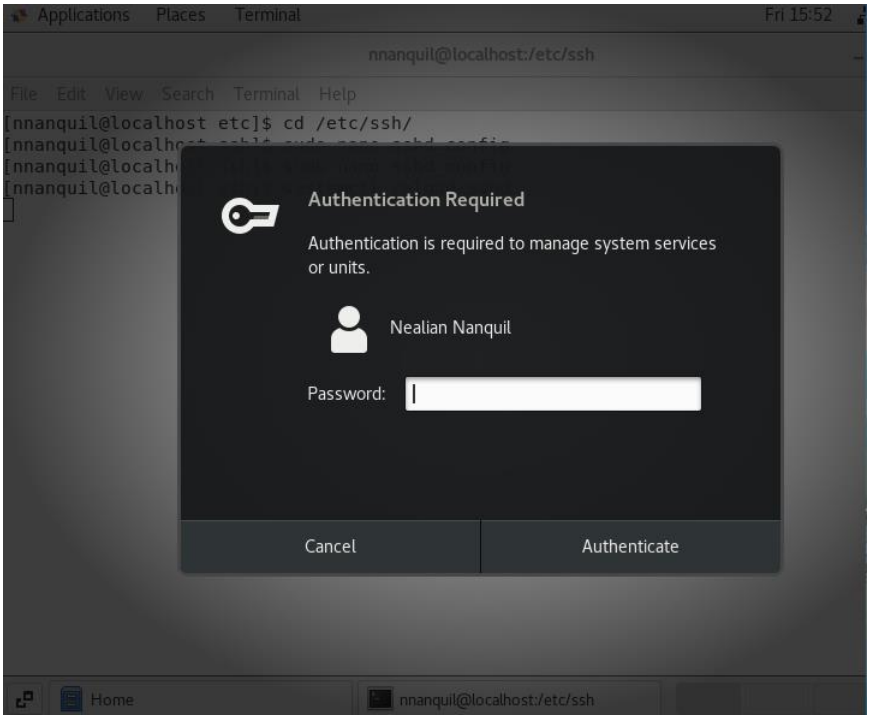
```
[nnanquil@localhost ~]$ sudo systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enable
d)
   Active: active (running) since Fri 2022-09-02 15:25:27 PST; 13min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 14950 (sshd)
   CGroup: /system.slice/sshd.service
           └─14950 /usr/sbin/sshd -D

Sep 02 15:25:26 localhost.localdomain systemd[1]: Stopped OpenSSH server daemon.
Sep 02 15:25:26 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Sep 02 15:25:27 localhost.localdomain sshd[14950]: Server listening on 0.0.0.0 port 22.
Sep 02 15:25:27 localhost.localdomain sshd[14950]: Server listening on :: port 22.
Sep 02 15:25:27 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
Hint: Some lines were ellipsized, use -l to show in full.
[nnanquil@localhost ~]$
```

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

```
[nnanquil@localhost ~]$ sudo firewall-cmd --zone=public --permanent --add-service=ssh
[sudo] password for nnanquil:
Warning: ALREADY_ENABLED: ssh
success
[nnanquil@localhost ~]$ sudo 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: 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 Po
^X Exit        ^J Justify     ^W Where Is    ^V Next Page   ^U UnCut Text  ^T To Spe
```




```
[nnanquil@localhost etc]$ cd /etc/ssh/
[nnanquil@localhost ssh]$ sudo nano sshd_config
[nnanquil@localhost ssh]$ sudo nano sshd_config
[nnanquil@localhost ssh]$ systemctl reload sshd
[nnanquil@localhost ssh]$
```

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.

```
nanqu@DESKTOP-1EOI7SG MINGW64 ~/.ssh (master)
$ ssh-copy-id -i ~/.ssh/id_rsa nnanquil@192.168.56.103
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/c/Users/nanqu/.ssh/id_rsa.p
ub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any tha
t are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it i
s to install the new keys
nnanquil@192.168.56.103's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'nnanquil@192.168.56.103'"
and check to make sure that only the key(s) you wanted were added.
```

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

```
[nnanquil@localhost ~]$ cd .ssh
[nnanquil@localhost .ssh]$ ls
authorized_keys
[nnanquil@localhost .ssh]$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDB+E4BKysOw7MR07L7hiRIsoSwKSU8btzdQMolxrVbWl89uRV
Az/K0lQjdtCCH8Gql+Ac7/ZEZaVlvD8giU5T6d5qGFRHSGo9sSbZBNcGvKhIhV2xGOGbYdvNIKObI/NqvvurRP+
raD+yBUvr58BLGlQ7anFfG7xT8ysWp7IUs+arOUc2rn6H258tQxMxoaRd8KQ9qifrvP6giQHIOUvbY3RSiGeMd1
yFaBW7DYemwDYK6ZtHm4sl9p9YoxcfSQXlYPjA1WBCE80Q3DTYjpUh12QqlnmYtbg0ToLoF24Y2iWWRJbW2iOfu
twJRGlNIUjE2LZd9pECl2ACmezLxxNszGIfZhD+oPrPuvOtIHHvNUidC2bAkCXXiw4H+MJy3dh0Ypu8hwFSTLgI
yV8zTvPeDbJktumhVheARcPvJ2F4aRbHbuVaPCAnUH5mq86Xo4hnrQrnFJ/yz8ulm9+H8onQY59RbbdLU83UcKD
iHGtY0sbXUTRMvTMEjFuNTei+R7c361AB94VyD5tJHmnIdTc++QRoID+EAVtKLgtc3M880lKTt+RWeEM2n05Z46
kAzbTjIOVgCotpVZg+Gy/OBkL4WgQYXCpeXVHcYhq8oR39N0tC5FFSRtXw1i5CQZ/4Bh/fiSIB/a0DEFl2a2D62
eN369jsSzNzloAPKDp4nN/QxAQ== nanqu@DESKTOP-1EOI7SG
[nnanquil@localhost .ssh]$
```

Task 4: Verify ssh remote connection

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

```
enp0s8: flags=4163<UP,BROADCA
        inet 192.168.56.103
```

```
nanqu@DESKTOP-1EOI7SG MINGW64 ~ (master)
$ ping 192.168.56.103

Pinging 192.168.56.103 with 32 bytes of data:
Request timed out.
Reply from 192.168.56.103: bytes=32 time<1ms TTL=64
Reply from 192.168.56.103: bytes=32 time<1ms TTL=64
Reply from 192.168.56.103: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.56.103:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

2. Show evidence that you are connected.

```
nanqu@DESKTOP-1EOI7SG MINGW64 ~/.ssh (master)
$ ssh nnanquil@192.168.56.103
Last login: Fri Sep  2 19:50:16 2022 from 192.168.56.1
```

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?

   I think we should look for security and accessibility. As what I have researched, Debian offers 59, 000 packages that includes developing of software and debugging that surely supports a wide range of capabilities. Also, each of the package is associated with registered Debian maintainer that is responsible for tracking bugs. This is the fact that Debian is more progressed than red hat.

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

   The main difference between Debian and Red Hat Linux is, Debian has 59, 000 packages that supports variety of function such as developing software, administering systems, connecting networks and debugging packages. Also, Debian is intelligent. While, Redhat has 3000 packages that fixes bugs that takes considerable time. Since it is controlled by a small group of people.