Name: LEE IVAN TITONG	**Date Performed: 9-5-2025**
Course/Section: CPE 212-CPE31S2	**Date Submitted: 9-5-2025**
Instructor: Engr. Robin Valenzuela	**Semester and SY: 2025-2026**

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.

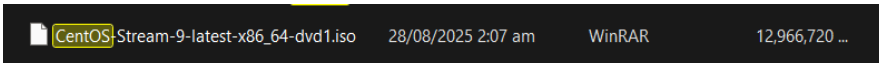
CentOS uses the RPM package format and YUM/DNF as the package manager.

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.

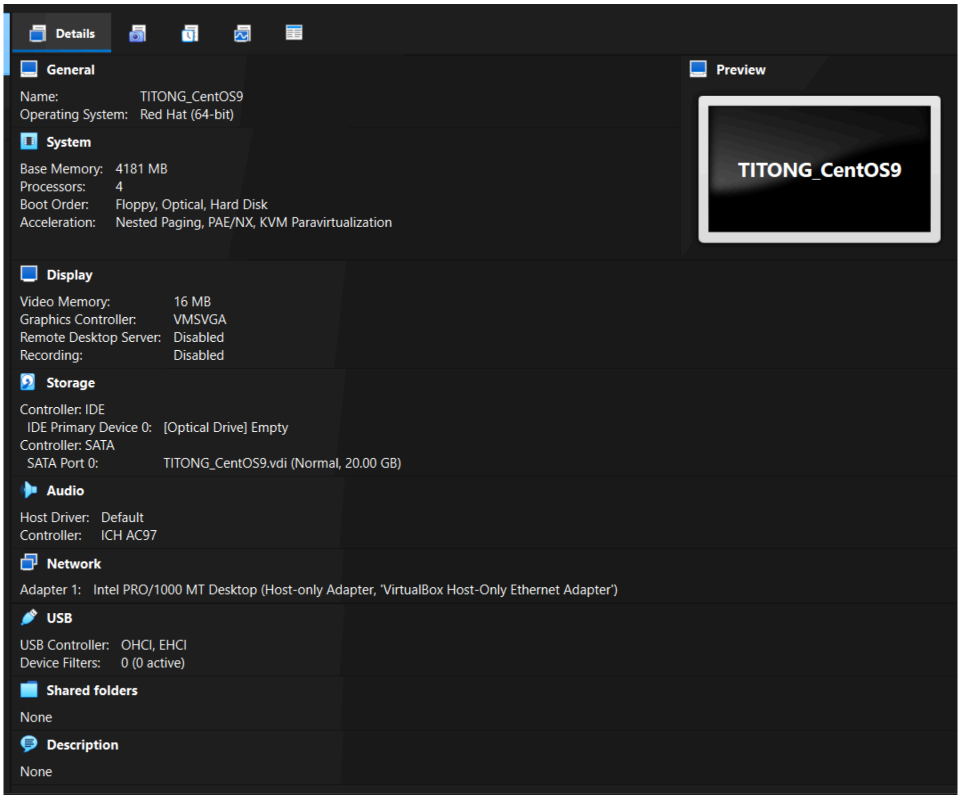
Task 1: Download the CentOS or RHEL-8 image (Create screenshots of the following)

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



CentOS-Stream-9-latest-x86_64-dvd1.iso   28/08/2025 2:07 am   WinRAR   12,966,720 ...

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



General
Name: TITONG_CentOS9
Operating System: Red Hat (64-bit)

System
Base Memory: 4181 MB
Processors: 4
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 Primary Device 0: [Optical Drive] Empty
Controller: SATA
SATA Port 0: TITONG_CentOS9.vdi (Normal, 20.00 GB)

Audio
Host Driver: Default
Controller: ICH AC97

Network
Adapter 1: Intel PRO/1000 MT Desktop (Host-only Adapter, 'VirtualBox Host-Only Ethernet Adapter')

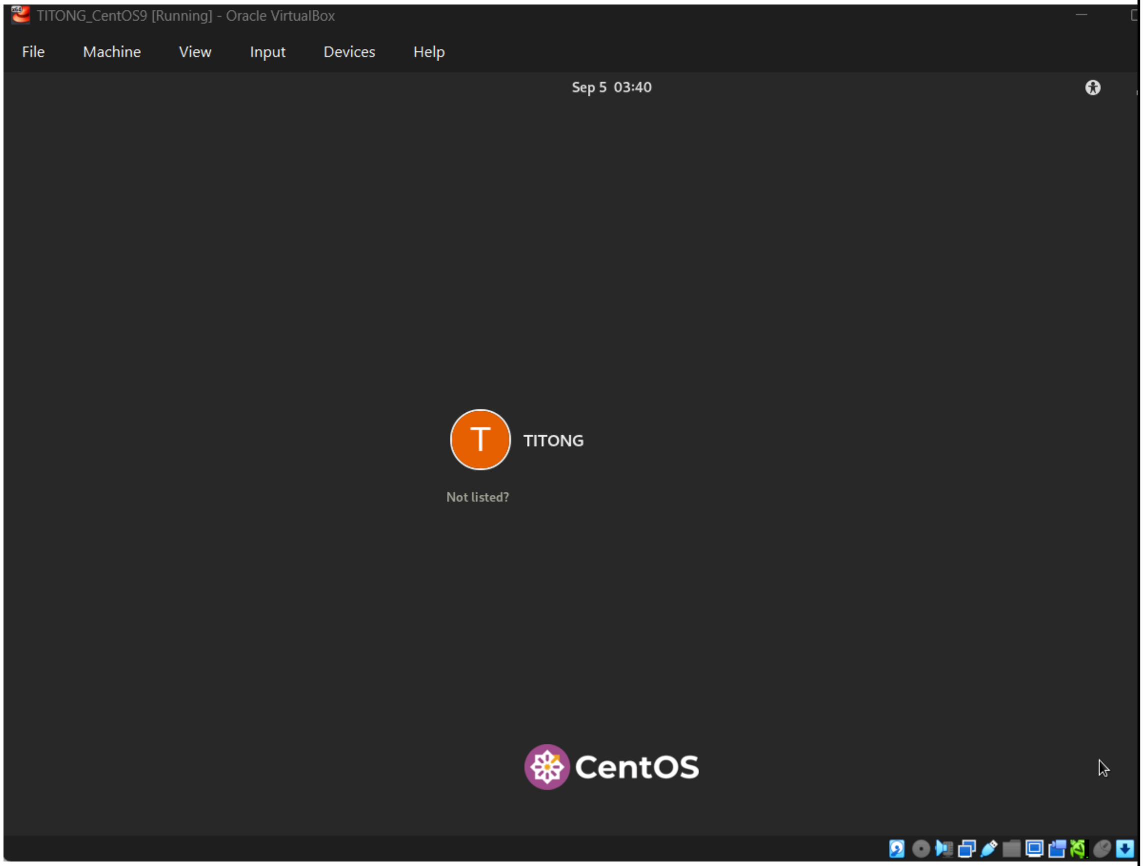
USB
USB Controller: OHCI, EHCI
Device Filters: 0 (0 active)

Shared folders
None

Description
None

3. Install the downloaded image.

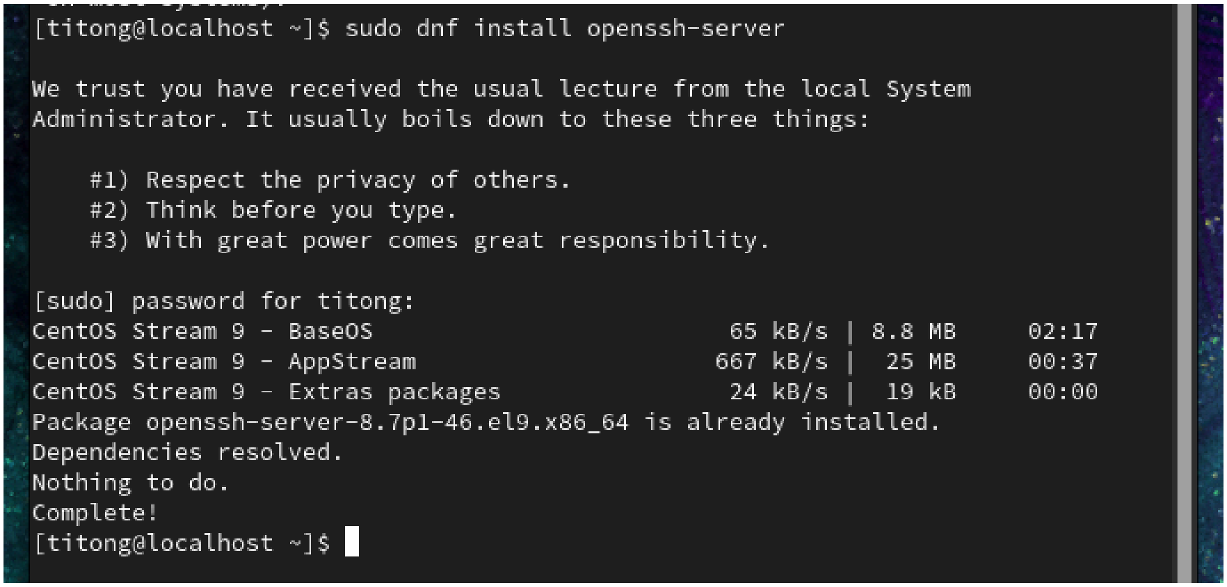
4. Show evidence that the OS was installed alread



.

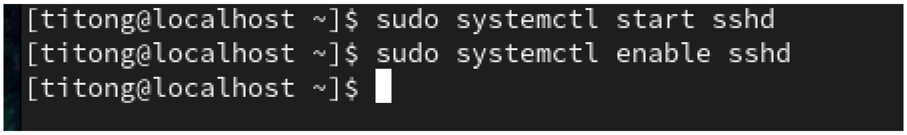
Task 2: Install the SSH server package *openssh*

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



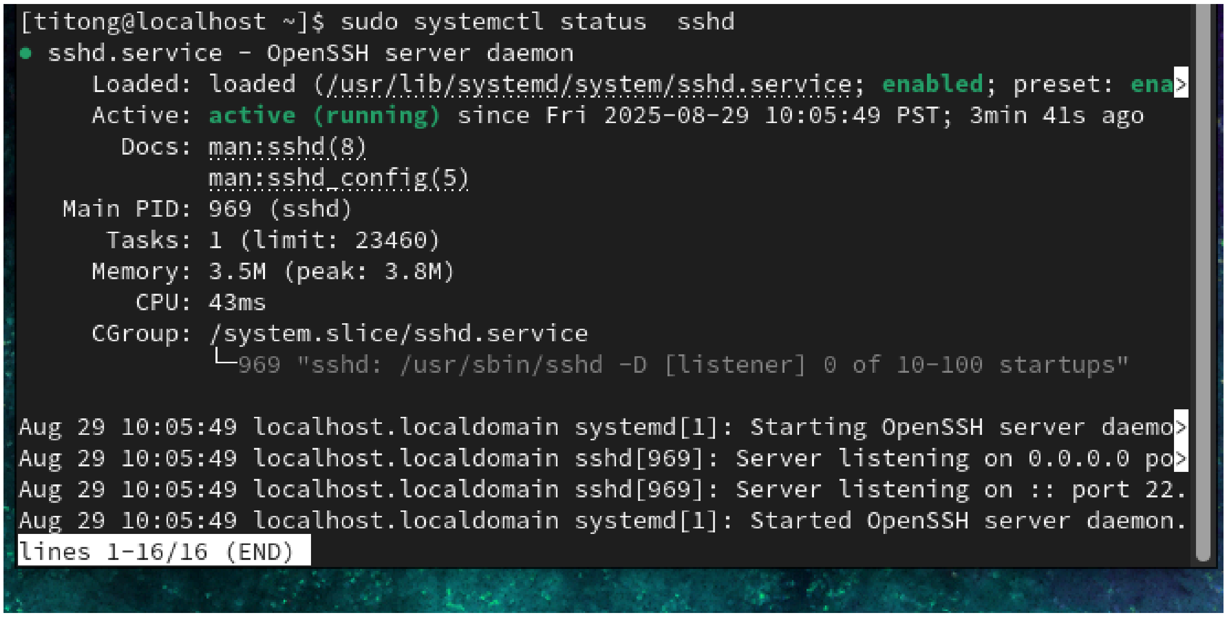
```
[titong@localhost ~]$ sudo dnf install openssh-server

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for titong:
CentOS Stream 9 - BaseOS                         65 kB/s | 8.8 MB     02:17
CentOS Stream 9 - AppStream                     667 kB/s |  25 MB     00:37
CentOS Stream 9 - Extras packages                24 kB/s |  19 kB     00:00
Package openssh-server-8.7p1-46.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[titong@localhost ~]$ 
```

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



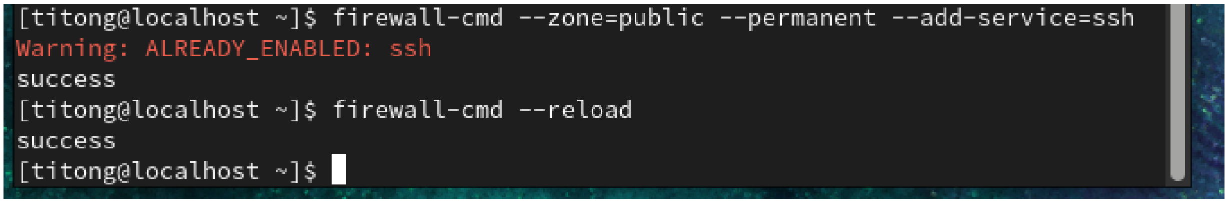
```
[titong@localhost ~]$ sudo systemctl start sshd
[titong@localhost ~]$ sudo systemctl enable sshd
[titong@localhost ~]$ 
```

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



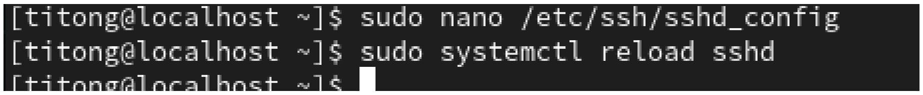
```
[titong@localhost ~]$ sudo systemctl status  sshd
● sshd.service - OpenSSH server daemon
     Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: ena>
     Active: active (running) since Fri 2025-08-29 10:05:49 PST; 3min 41s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 969 (sshd)
      Tasks: 1 (limit: 23460)
     Memory: 3.5M (peak: 3.8M)
        CPU: 43ms
     CGroup: /system.slice/sshd.service
             └─969 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Aug 29 10:05:49 localhost.localdomain systemd[1]: Starting OpenSSH server daemo>
Aug 29 10:05:49 localhost.localdomain sshd[969]: Server listening on 0.0.0.0 po>
Aug 29 10:05:49 localhost.localdomain sshd[969]: Server listening on :: port 22.
Aug 29 10:05:49 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
lines 1-16/16 (END)
```

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



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

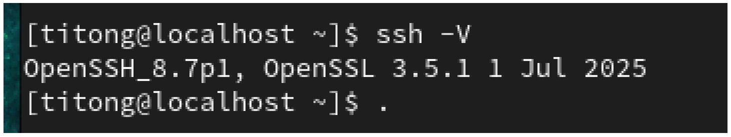
5. Locate the ssh server man config file */etc/ssh/sshd_config* and perform custom configuration. Every time you make any change to the */etc/ssh/sshd-config* configuration file reload the *sshd* service to apply changes:
   $ systemctl reload sshd



```
[titong@localhost ~]$ sudo nano /etc/ssh/sshd_config
[titong@localhost ~]$ sudo systemctl reload sshd
[titong@localhost ~]$ 
```

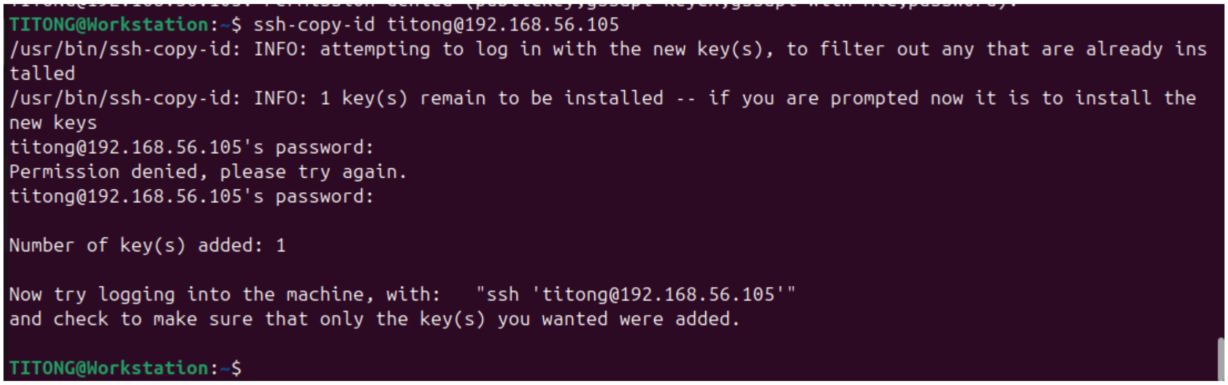
Task 3: Copy the Public Key to CentOS

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



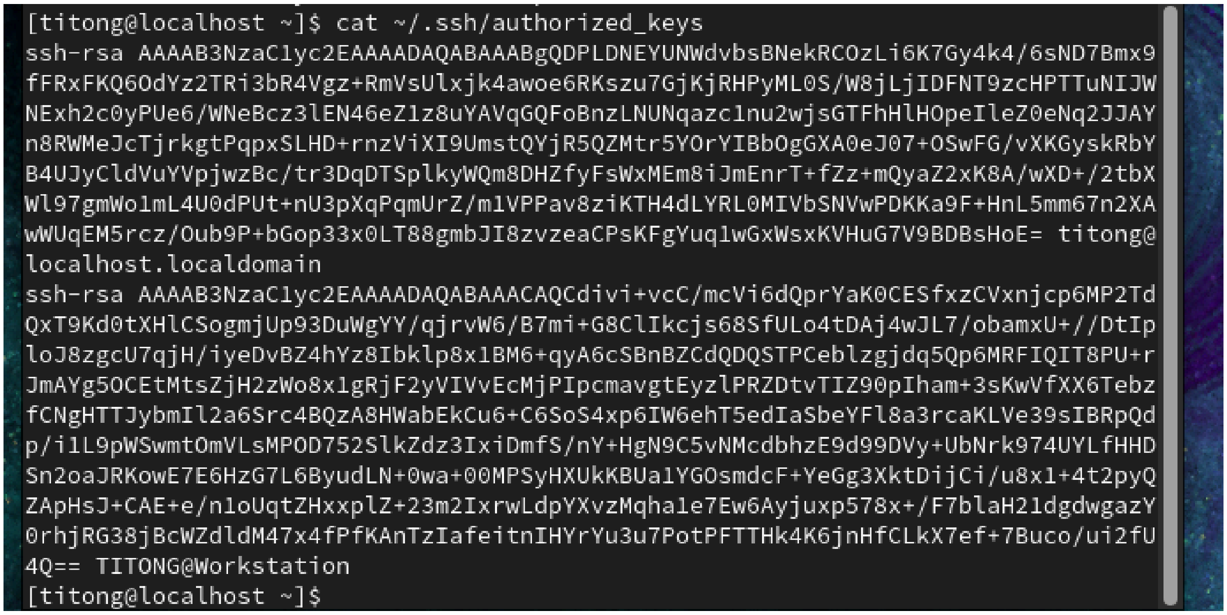
```
[titong@localhost ~]$ ssh -V
OpenSSH_8.7p1, OpenSSL 3.5.1 1 Jul 2025
[titong@localhost ~]$ .
```

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



```
TITONG@Workstation:~$ ssh-copy-id titong@192.168.56.105
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already ins
talled
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the
new keys
titong@192.168.56.105's password:
Permission denied, please try again.
titong@192.168.56.105's password:

Number of key(s) added: 1

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

TITONG@Workstation:~$
```

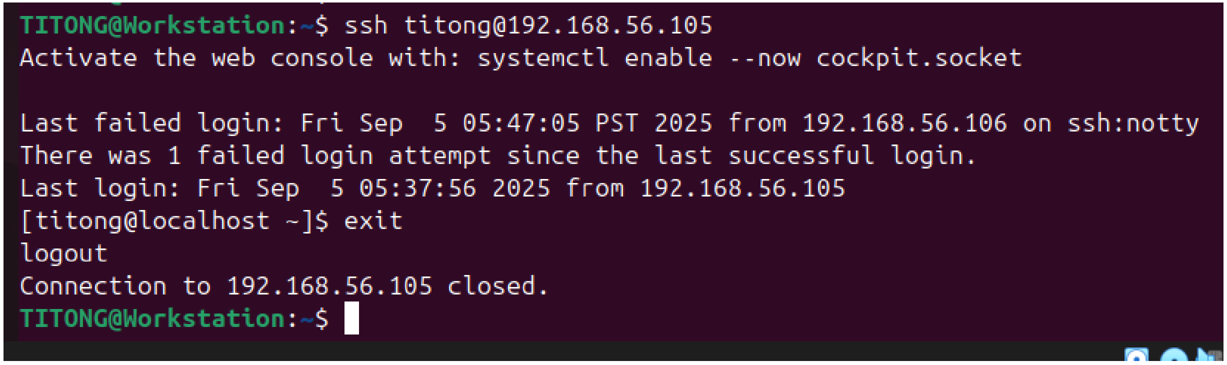
3. On CentOS, verify that you have the *authorized_keys*.



```
[titong@localhost ~]$ cat ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDPLDNEYUNWdvbsBNekRCOzLi6K7Gy4k4/6sND7Bmx9
fFRxFKQ6OdYz2TRi3bR4Vgz+RmVsUlxjk4awoe6RKszu7GjKjRHPyML0S/W8jLjIDFNT9zcHPTTuNIJW
NExh2c0yPUe6/WNeBcz3lEN46eZ1z8uYAVqGQFoBnzLNUNqazc1nu2wjsGTFhHlHOpeIleZ0eNq2JJAY
n8RWMeJcTjrkgtPqpxSLHD+rnzViXI9UmstQYjR5QZMtr5YOrYIBbOgGXA0eJ07+OSwFG/vXKGyskRbY
B4UJyCldVuYVpjwzBc/tr3DqDTSplkyWQm8DHZfyFsWxMEm8iJmEnrT+fZz+mQyaZ2xK8A/wXD+/2tbX
Wl97gmWo1mL4U0dPUt+nU3pXqPqmUrZ/m1VPPav8ziKTH4dLYRL0MIVbSNVwPDKKa9F+HnL5mm67n2XA
wWUqEM5rcz/Oub9P+bGop33x0LT88gmbJI8zvzeaCPsKFgYuq1wGxWsxKVHuG7V9BDBsHoE= titong@
localhost.localdomain
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQCdivi+vcC/mcVi6dQprYaK0CESfxzCVxnjcp6MP2Td
QxT9Kd0tXHlCSogmjUp93DuWgYY/qjrvW6/B7mi+G8ClIkcjs68SfULo4tDAj4wJL7/obamxU+//DtIp
loJ8zgcU7qjH/iyeDvBZ4hYz8Ibklp8x1BM6+qyA6cSBnBZCdQDQSTPCeblzgjdq5Qp6MRFIQIT8PU+r
JmAYg5OCEtMtsZjH2zWo8x1gRjF2yVIVvEcMjPIpcmavgtEyzlPRZDtvTIZ90pIham+3sKwVfXX6Tebz
fCNgHTTJybmIl2a6Src4BQzA8HWabEkCu6+C6SoS4xp6IW6ehT5edIaSbeYFl8a3rcaKLVe39sIBRpQd
p/i1L9pWSwmtOmVLsMPOD752SlkZdz3IxiDmfS/nY+HgN9C5vNMcdbhzE9d99DVy+UbNrk974UYLfHHD
Sn2oaJRKowE7E6HzG7L6ByudLN+0wa+00MPSyHXUkKBUa1YGOsmdcF+YeGg3XktDijCi/u8x1+4t2pyQ
ZApHsJ+CAE+e/n1oUqtZHxxplZ+23m2IxrwLdpYXvzMqha1e7Ew6Ayjuxp578x+/F7blaH21dgdwgazY
0rhjRG38jBcWZdldM47x4fPfKAnTzIafeitnIHYrYu3u7PotPFTTHk4K6jnHfCLkX7ef+7Buco/ui2fU
4Q== TITONG@Workstation
[titong@localhost ~]$
```

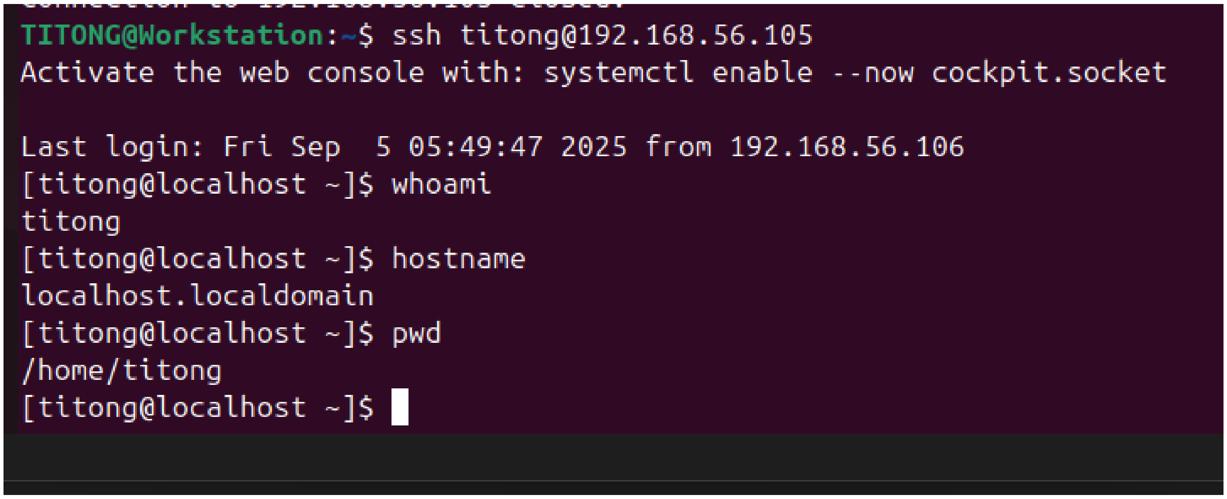
Task 4: Verify ssh remote connection

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



```
TITONG@Workstation:~$ ssh titong@192.168.56.105
Activate the web console with: systemctl enable --now cockpit.socket

Last failed login: Fri Sep  5 05:47:05 PST 2025 from 192.168.56.106 on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Fri Sep  5 05:37:56 2025 from 192.168.56.105
[titong@localhost ~]$ exit
logout
Connection to 192.168.56.105 closed.
TITONG@Workstation:~$ 
```

2. Show evidence that you are connected.



```
TITONG@Workstation:~$ ssh titong@192.168.56.105
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Fri Sep  5 05:49:47 2025 from 192.168.56.106
[titong@localhost ~]$ whoami
titong
[titong@localhost ~]$ hostname
localhost.localdomain
[titong@localhost ~]$ pwd
/home/titong
[titong@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?
   - When choosing between Debian and Red Hat Linux distributions, the decision should be based on your specific needs and experience level. For beginners and personal use, Debian is often the better choice because it is free, user-friendly, and has a large community for support. If you are setting up servers for a company or need a very stable system with professional support, Red Hat is more suitable since it offers long-term updates and paid customer service. You should also consider which package system you prefer—Debian uses .deb files while Red Hat uses .rpm files—and whether you value free software versus corporate-backed reliability.

2. What are the main diffence between Debian and Red Hat Linux distributions?
   - The main differences between Debian and Red Hat Linux are their target users, support models, and package systems. Debian is free and

community-supported, making it ideal for learning and personal projects. It uses .deb packages and the apt command for installing software. Red Hat, on the other hand, is designed for businesses and offers paid support with very stable long-term releases. It uses .rpm packages and the yum or dnf commands. In short, Debian is great for individuals and experimentation, while Red Hat is stronger for professional and enterprise environments where paid support and reliability are important.