

<b>Name:</b> Kazuki A. Ogata	<b>Date Performed:</b> September 05, 2023
<b>Course/Section:</b> CPE 232 - CPE31S5	<b>Date Submitted:</b> September 06, 2023
<b>Instructor:</b> Engr. Roman Richard	<b>Semester and SY:</b> 1st semester S.Y 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.

1 of 6



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 arches 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 arches 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/](http://mirror.rise.ph/centos/7.9.2009/isos/x86_64/)

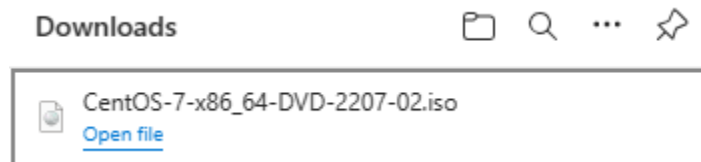
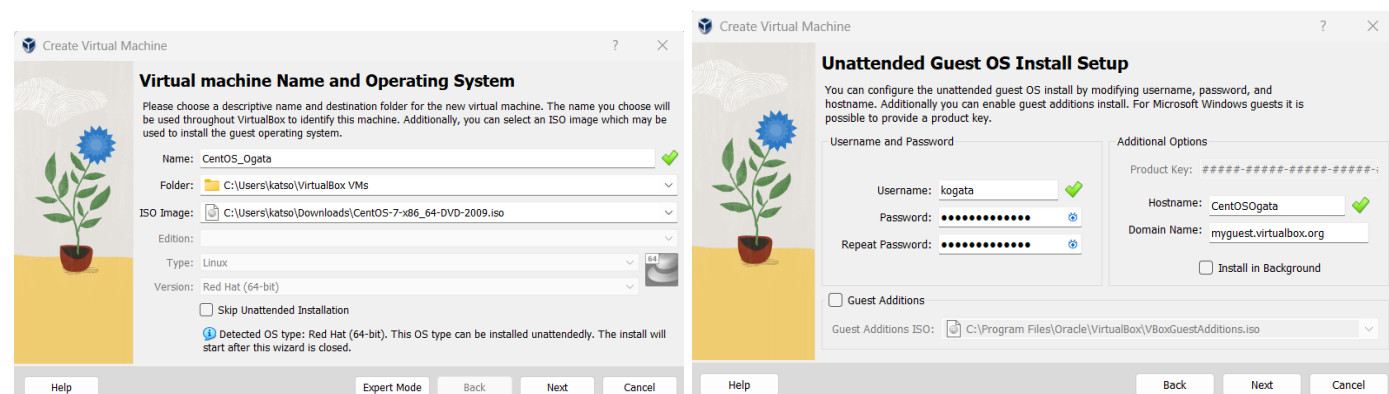


Figure 1.1: Download CentOS

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



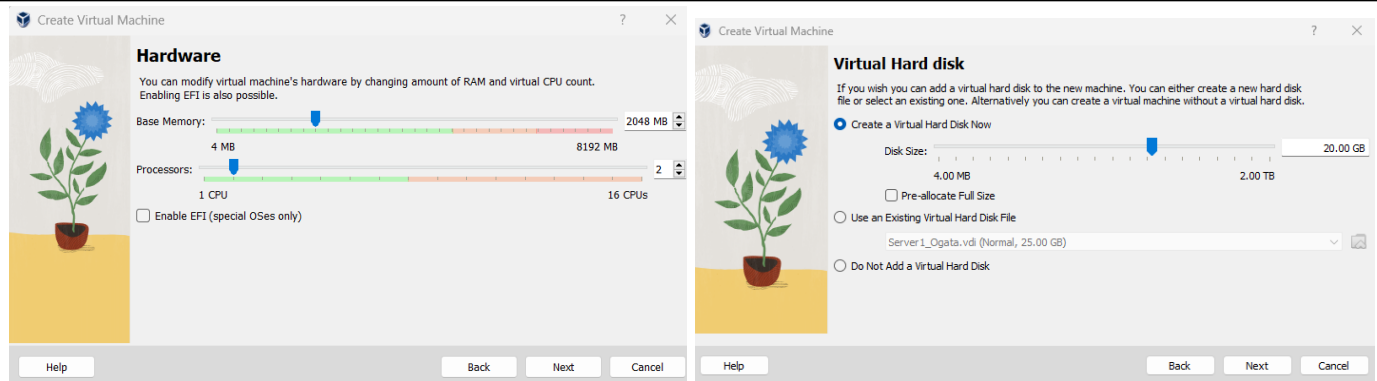
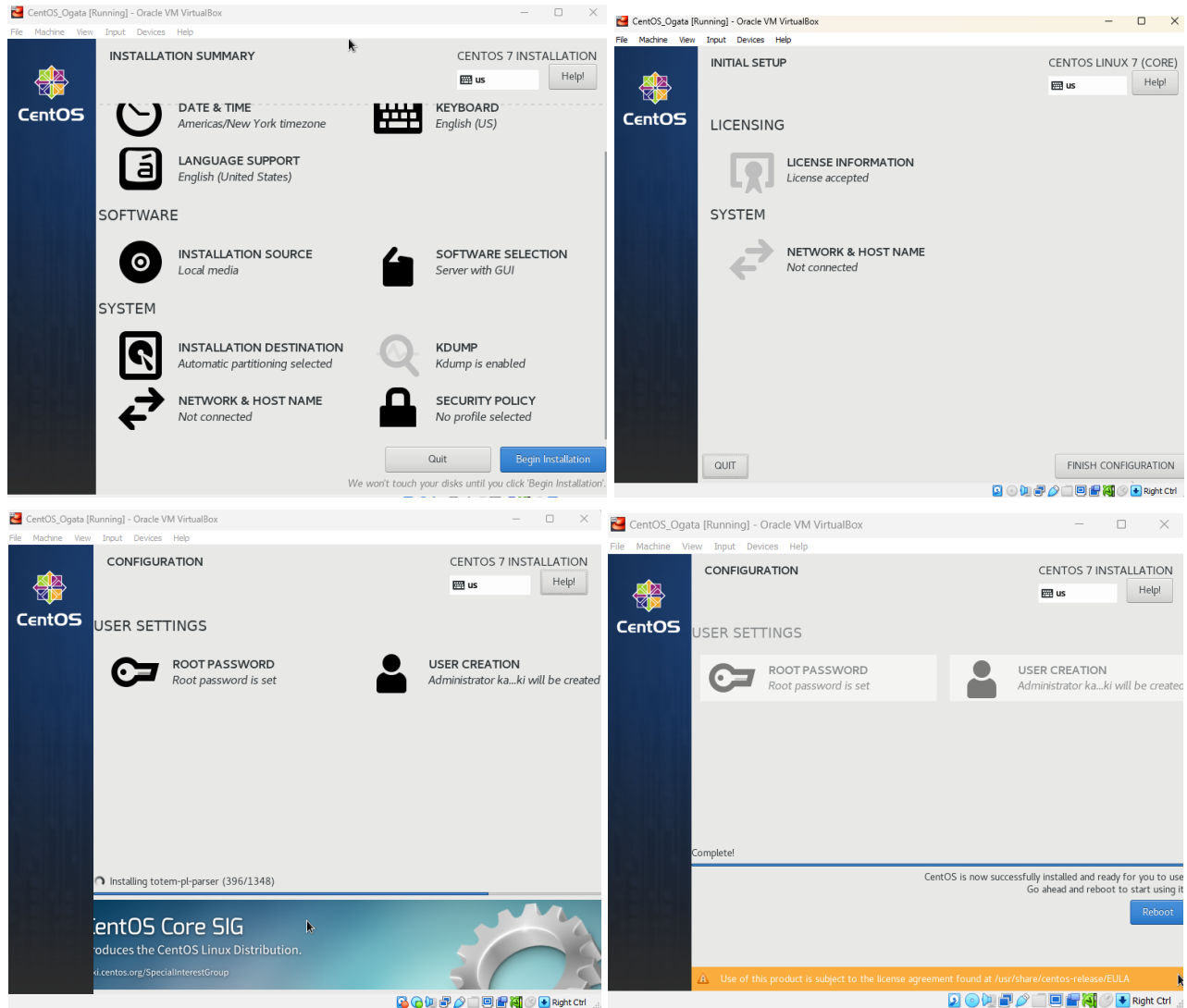


Figure 1.2-5: Step-by-Step Creation of VM Machine

### 3. Install the downloaded image.



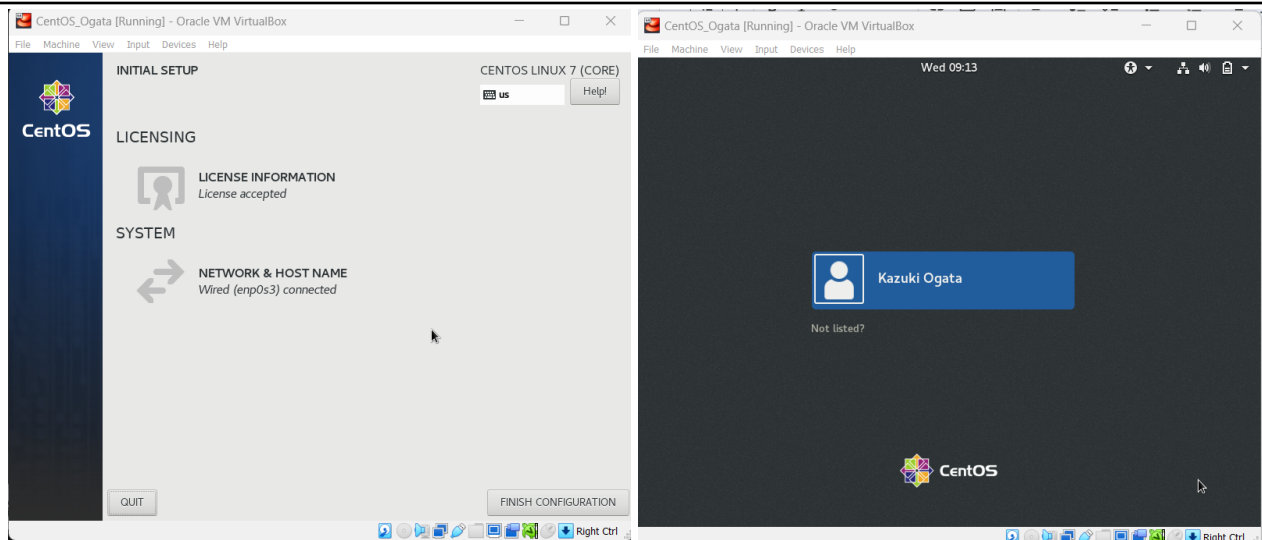


Figure 1.6-11: Installation of CentOS

#### 4. Show evidence that the OS was installed already.

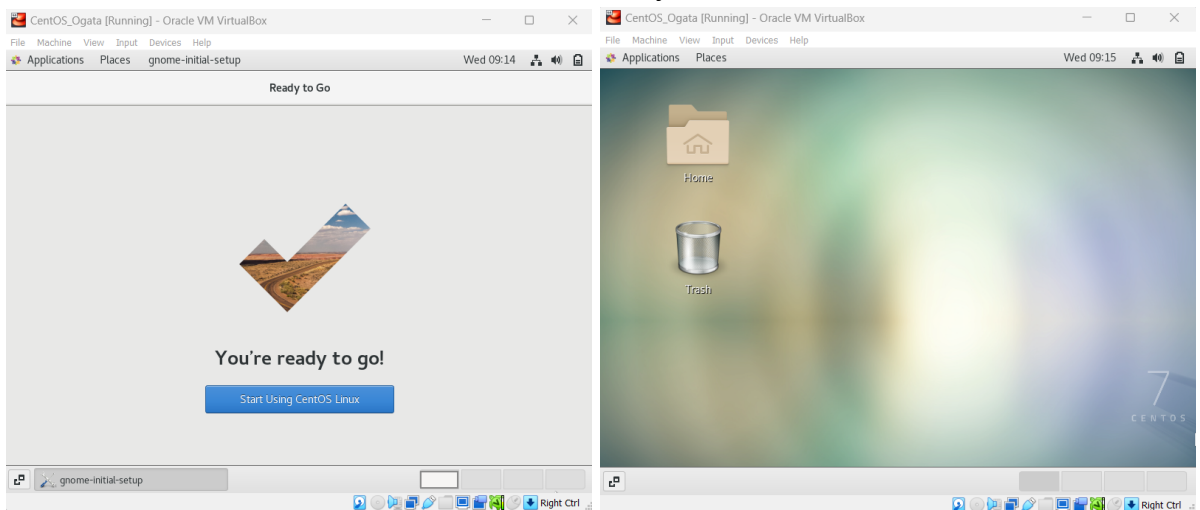


Figure 1.12: Installed CentOS

### Task 2: Install the SSH server package *openssh*

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

**\$ dnf install openssh-server**

```
[kogata@centos ~]$ dnf install openssh-server
Error: This command has to be run under the root user.
[kogata@centos ~]$ su
Password:
[root@centos kogata]# dnf install openssh-server
CentOS-7 - Base                               4.3 MB/s | 10 MB      00:02
CentOS-7 - Updates                             2.6 MB/s | 28 MB      00:10
CentOS-7 - Extras                             1.3 MB/s | 360 kB      00:00
Package openssh-server-7.4p1-22.el7_9.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**

```
[root@centos kogata]# systemctl start sshd  
[root@centos kogata]# systemctl enable sshd
```

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

**\$ systemctl status sshd**

```
[root@centos kogata]# systemctl status sshd  
● sshd.service - OpenSSH server daemon  
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; vendor preset: enable  
d)  
   Active: active (running) since Sat 2023-09-09 20:23:39 PST; 5min ago  
     Docs: man:sshd(8)  
           man:sshd_config(5)  
  Main PID: 1178 (sshd)  
    CGroup: /system.slice/ssh.service  
            └─1178 /usr/sbin/sshd -D  
  
Sep 09 20:23:39 centos systemd[1]: Starting OpenSSH server daemon...  
Sep 09 20:23:39 centos sshd[1178]: Server listening on 0.0.0.0 port 22.  
Sep 09 20:23:39 centos sshd[1178]: Server listening on :: port 22.  
Sep 09 20:23:39 centos systemd[1]: Started OpenSSH server daemon.
```

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

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

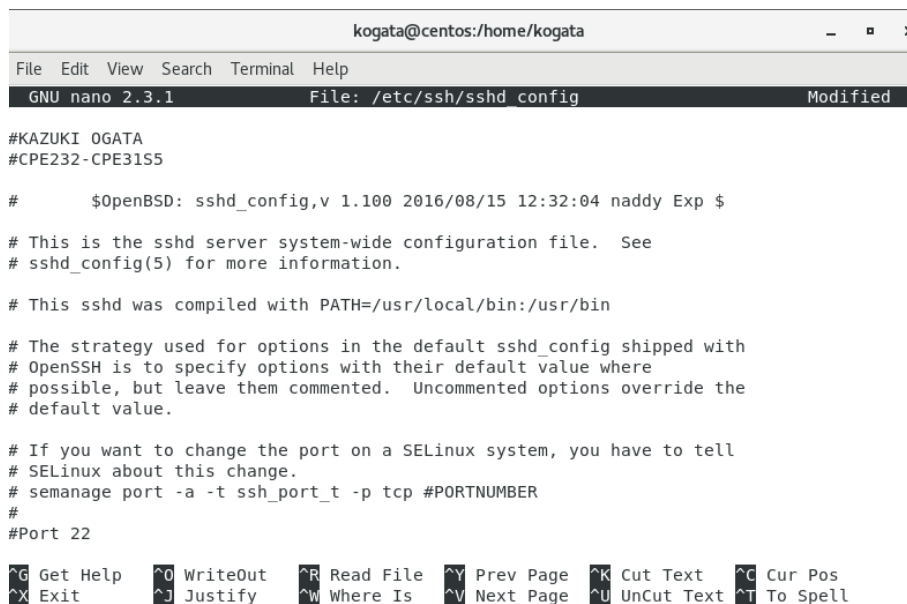
**\$ firewall-cmd --reload**

```
[root@centos kogata]# firewall-cmd --zone=public --permanent --add-service=ssh  
Warning: ALREADY_ENABLED: ssh  
success  
[root@centos kogata]# firewall-cmd --reload  
success
```

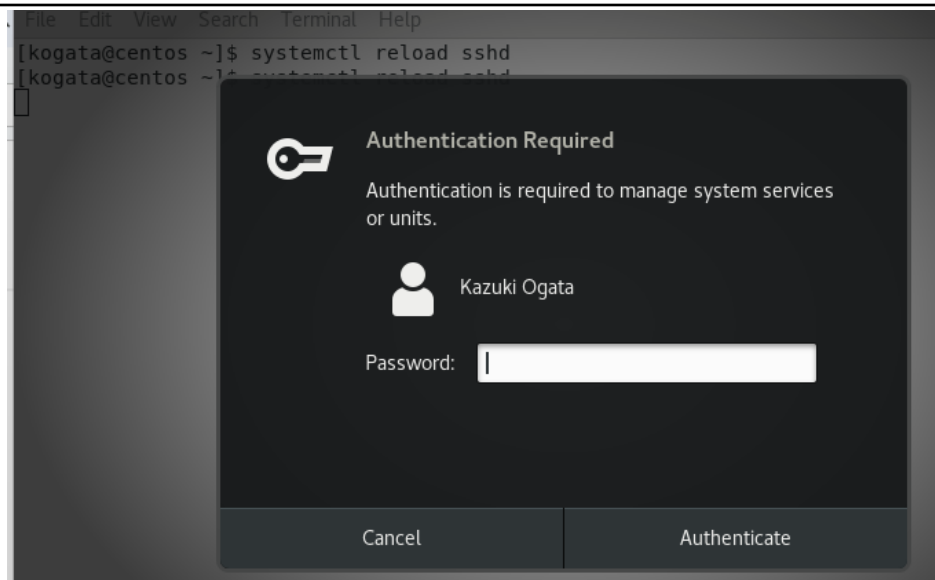
5. Locate the ssh server man config file **/etc/ssh/sshd\_config** and perform custom configuration. Every time you make any change to the **/etc/ssh/sshd-config** configuration file reload the **sshd** service to apply changes:

**\$ systemctl reload sshd**

```
[root@centos kogata]# sudo nano /etc/ssh/sshd config
```



```
#KAZUKI OGATA  
#CPE232-CPE31S5  
  
#      $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  
  
^G Get Help      ^O WriteOut     ^R Read File    ^V Prev Page    ^K Cut Text     ^C Cur Pos  
^X Exit         ^J Justify      ^W Where Is    ^N 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.

```
kazuki@workstation:~$ sudo systemctl status ssh
[sudo] password for kazuki:
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-09-14 05:39:26 +08; 22min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 739 (sshd)
    Tasks: 1 (limit: 4599)
   Memory: 3.6M
      CPU: 49ms
   CGroup: /system.slice/ssh.service
           └─739 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

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

```
kazuki@workstation:~$ ssh-copy-id -i ~/.ssh/id_rsa kogata@192.168.56.116
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/kazuki/.ssh/id_rsa.pub"
/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 prompted now it is
to install the new keys
kogata@192.168.56.116's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'kogata@192.168.56.116'"
and check to make sure that only the key(s) you wanted were added.
```

3. On CentOS, verify that you have the **authorized\_keys**.

```
[kogata@centos ~]$ ls ~/.ssh
authorized_keys
```

#### Task 4: Verify ssh remote connection

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

```
kazuki@workstation:~$ ssh kogata@192.168.56.116
Last login: Thu Sep 14 06:48:16 2023 from 192.168.56.114
[kogata@centos ~]$
```

2. Show evidence that you are connected.

```
kazuki@workstation:~$ ssh kogata@192.168.56.116
Last login: Thu Sep 14 06:54:56 2023 from 192.168.56.114
[kogata@centos ~]$ logout
Connection to 192.168.56.116 closed.
kazuki@workstation:~$ ssh kogata@centos
Last login: Thu Sep 14 06:55:05 2023 from 192.168.56.114
[kogata@centos ~]$
```

#### 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?
  - In choosing the best distribution between Debian and Red Hat Linux distribution, we should know the specific needs of our system. We can consider the factors like stability, release cycle, package management, and its community. I have already experienced using CentOS and I am not liking its GUI... and based on my research, Debian has a more user-friendly GUI. So, for me, I'll pick Debian.
2. What are the main differences between Debian and Red Hat Linux distributions?
  - CentOS is a Linux distribution renowned for its stability for enterprise environments. It is supported by the Red Hat community. It also has a large market since CentOS has a user-friendly nature. But it has a complicated GUI and has limited packages. In terms of package management, CentOS uses YUM package manager. Debian is a Linux distribution, to be specific, a community-driven Linux distribution where it is maintained and developed by a community. It lacks a market because of its terminal end usage. In terms of architecture, Debian has multiple architecture support compared to CentOS. Also, Debian has a user-friendly GUI and has a lot of packages. Additionally, Debian uses APT package manager.