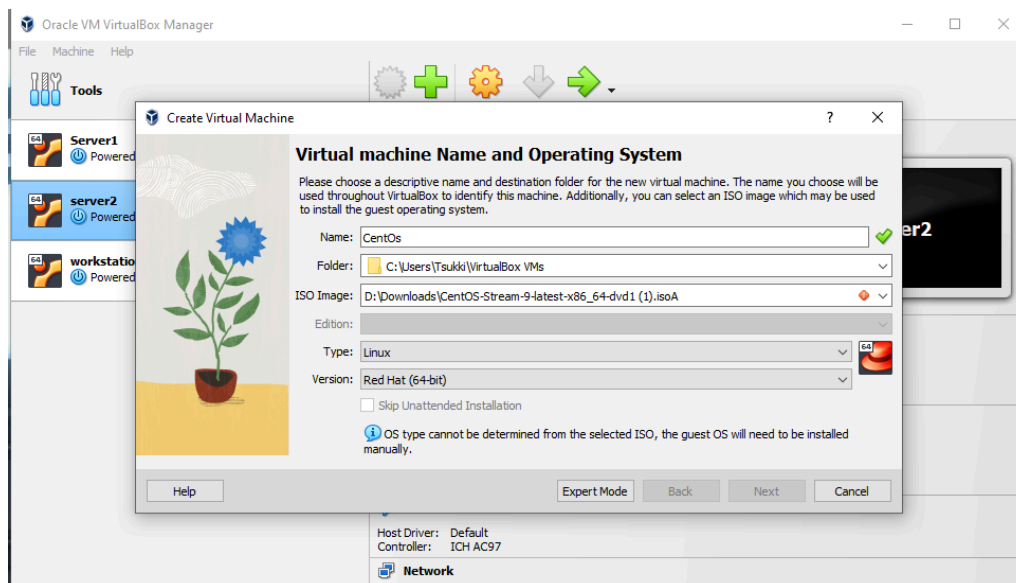


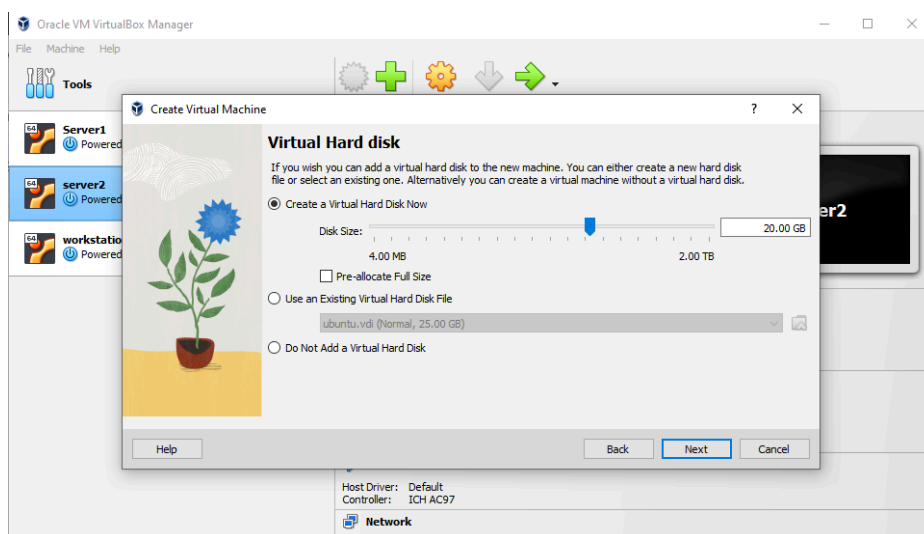
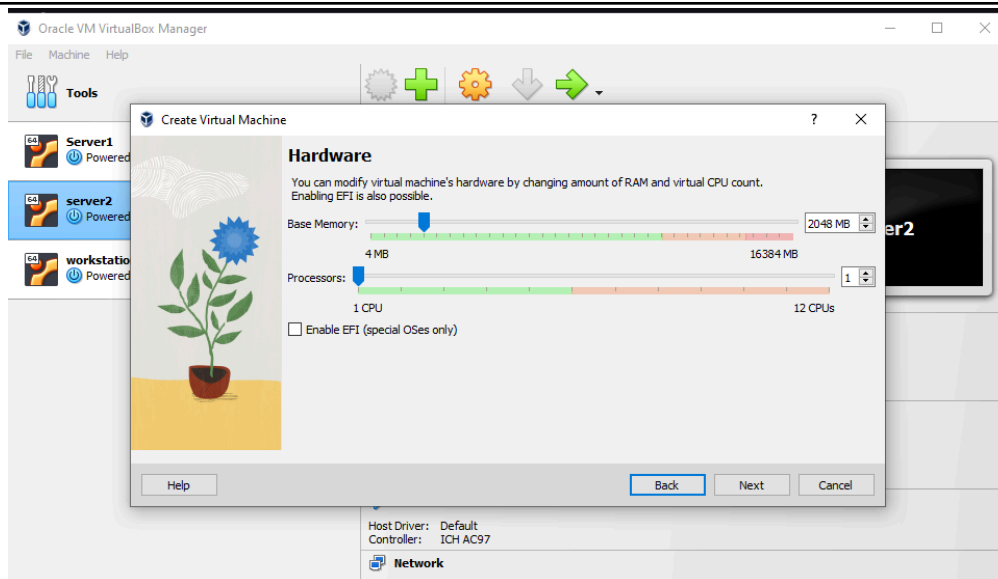
Name:	Date Performed:
Course/Section:	Date Submitted:
Instructor:	Semester and SY:
Activity 3: Install SSH server on CentOS or RHEL 8	
1. Objectives: 1.1 Install Community Enterprise OS or Red Hat Linux OS 1.2 Configure remote SSH connection from remote computer to CentOS/RHEL-8	
2. Discussion: CentOS 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.	

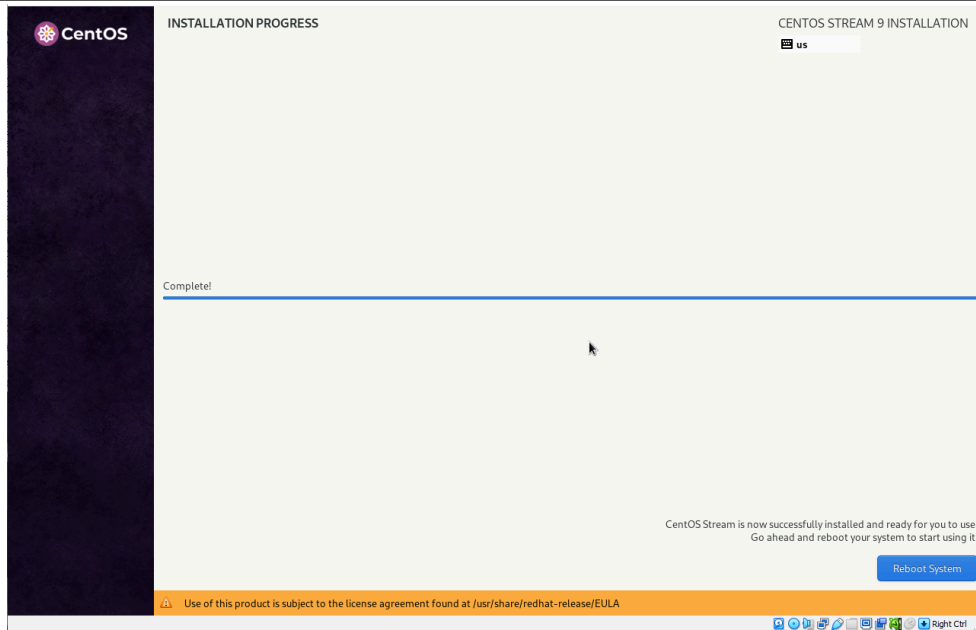
Both offer full-feature package management with network-based repository support, dependency checking and resolution, etc.. If you're familiar with one but not the other, you may have a little trouble switching over, but they're not overwhelmingly different. They both have similar features, just available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

```
[tdee@localhost ~]$ sudo dnf install openssh-server
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use "rhc
subscription-manager" to register.

Last metadata expiration check: 0:00:30 ago on Fri 13 Sep 2024 08:29:28 AM
Package openssh-server-8.7p1-43.el9.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

```
[tdee@localhost ~]$ systemctl start sshd
[tdee@localhost ~]$ systemctl enable sshd
```

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

\$ systemctl status sshd

```
[tdee@localhost ~]$ systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset
   Active: active (running) since Fri 2024-09-13 07:53:57 EDT; 36min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
    Main PID: 857 (sshd)
      Tasks: 1 (limit: 10951)
     Memory: 2.3M
        CPU: 17ms
    CGroup: /system.slice/sshd.service
            └─857 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startup
```

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

```
$ firewall-cmd --zone=public --permanent --add-service=ssh
$ firewall-cmd --reload
```

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

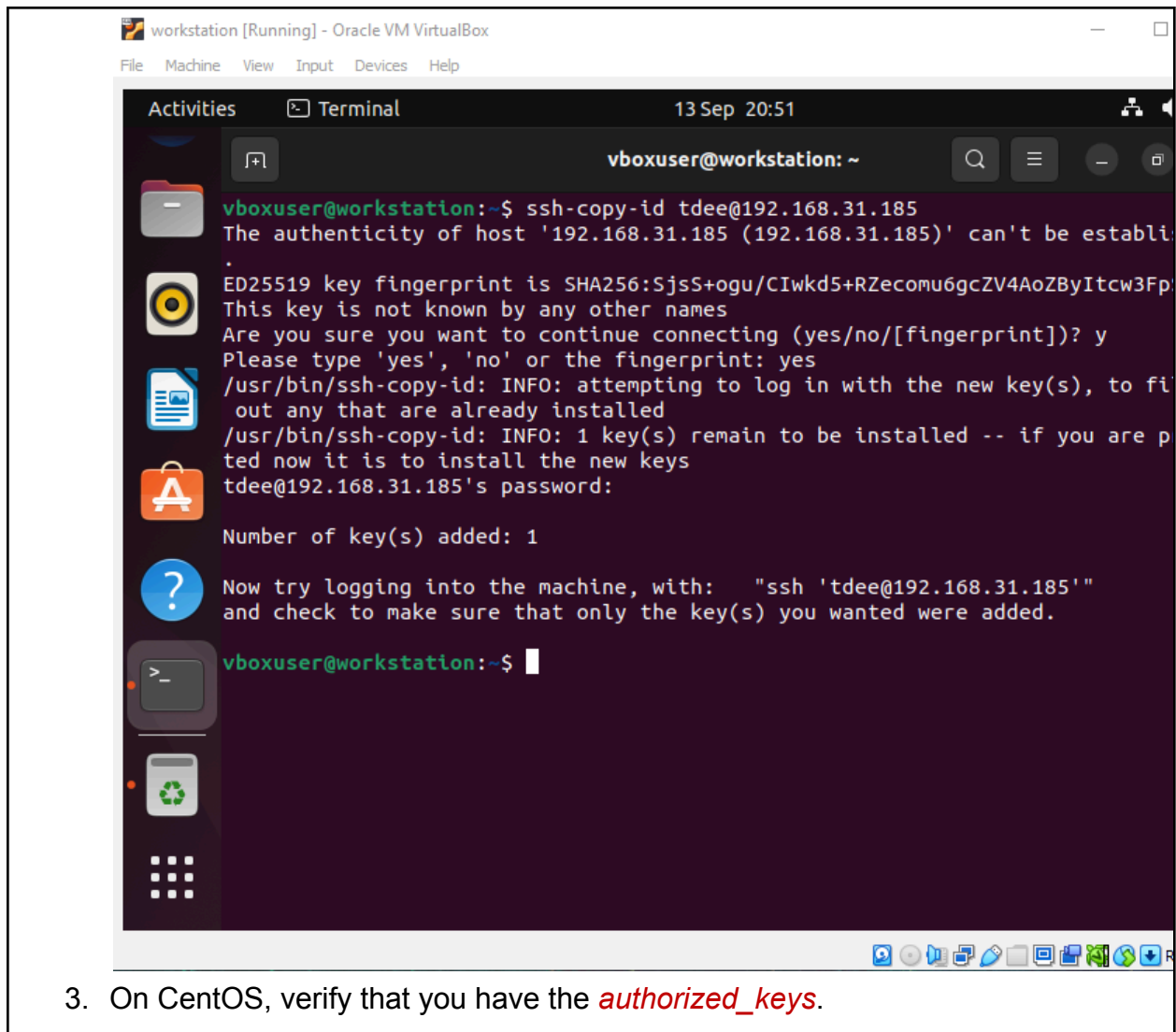
5. Locate the ssh server man config file */etc/ssh/sshd_config* and perform custom configuration. Every time you make any change to the */etc/ssh/sshd-config* configuration file reload the *sshd* service to apply changes:

```
$ systemctl reload sshd
```

```
[tdee@localhost ~]$ systemctl reload sshd
[tdee@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.



```
Terminal Sep 13 08:52

tdee@MiWiFi-R4A-srv:~

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 18 bytes 2112 (2.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 18 bytes 2112 (2.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[tdee@MiWiFi-R4A-srv ~]$ cat ~/.ssh/authorized_key
cat: /home/tdee/.ssh/authorized_key: No such file or directory
[tdee@MiWiFi-R4A-srv ~]$ cat ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCSPI4zokVEo1NfQTb1HhHuJgiQ4ZLIpr+isPAhEA2P
1d04HGceucsYUQaKAokLa6WN+c7W/Ht4wQEInT4mkK9j7Wo0K9Z7Cw1WgoXMBG+R0aUHX7GAYdLjx1It
2zNSlAfAvrsutyKWecatGtHWhTJf25RmiEFSiXYmSwdHl0M0ibbRBK5zzIgOptcuriifvh0w4pDwVhMe
aUwt9ChZQxgXN+94/qTfmYMyPSCLiipX0ZkNBxw9kq3InyekuFglBqmsvPjsALy7zu7n4mmArRm07XK0
9u9fLZPBH2AFPOQM694AELG6/9FHFaf7kha/2wMuSoPq2NqX8a16uwNHzsBckpS36NSc9G50x1lyWLhHZ
qMRqvyyS8ASrQILwpR5NXJeTL7DChYvsqmMe4lrEKYorGuwx09eWjnrb2GLwtjhSaGkEnDMREEffhFE/
pXzIOu7vIrHvTA1PHu3hiIUj4p6jl2oXNerPYLD+llpYV/RTbZeUo18WvhPa8ozIB7yw8PuPRXzd2n0d
xQlBoJQcz7IEKImmORmQq7IQF8YejJQPlvL8eJ1ZKgbbaoBstTrI+LQqu0Ne38smLBQBkNLCU1wc59lK
A8nFoLQFhudUrmPy/DvAM8xdNoJQ$gfd2ofRY76DNMC06rX7bQu61ZEG+gPSuboj6sIhLGo/rzInJt1x
bQ== vboxuser@workstation
[tdee@MiWiFi-R4A-srv ~]$
```

Task 4: Verify ssh remote connection

1. Using your local machine, connect to CentOS using ssh.
2. Show evidence that you are connected.

The screenshot shows a terminal window titled "workstation [Running] - Oracle VM VirtualBox". The terminal is running a command to copy an SSH key to a remote host. The output shows the key fingerprint, a confirmation prompt, and the successful addition of the key. The user then attempts to log in to the remote host.

```
workstation [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Activities Terminal 13 Sep 20:52

tdee@MiWiFi-R4A-srv:~

vboxuser@workstation:~$ ssh-copy-id tdee@192.168.31.185
The authenticity of host '192.168.31.185 (192.168.31.185)' can't be established
ED25519 key fingerprint is SHA256:SjsS+ogu/CIwkd5+RZecomu6gcZV4AoZByItcw3FpSI.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the 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 promp
ted now it is to install the new keys
tdee@192.168.31.185's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'tdee@192.168.31.185'"
and check to make sure that only the key(s) you wanted were added.

vboxuser@workstation:~$ ssh tdee@192.168.31.185
Last login: Fri Sep 13 08:48:40 2024
[tdee@MiWiFi-R4A-srv ~]$
```

The screenshot shows the output of the 'whoami' and 'hostname' commands on a remote server. The user is logged in as 'tdee' and the hostname is 'MiWiFi-R4A-srv'.

```
vboxuser@workstation:~$ ssh tdee@192.168.31.185
Last login: Fri Sep 13 08:48:40 2024
[tdee@MiWiFi-R4A-srv ~]$ whoami
tdee
[tdee@MiWiFi-R4A-srv ~]$ hostname
MiWiFi-R4A-srv
[tdee@MiWiFi-R4A-srv ~]$
```

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, consider the package management system, which affects ease of use and software availability. Evaluate the support option Red Hat offers enterprise-level paid support, while Debian is community-driven with free support. Lastly, think about the target environment

Debian is often chosen for flexibility and stability in diverse environments, while Red Hat excels in commercial, and enterprise setups.

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

Debian uses the APT package manager and .deb packages, while Red Hat uses YUM/DNF and .rpm packages. Red Hat is known for its enterprise focus, offering commercial support and certifications, while Debian is more community-driven and non-commercial. System administration differs too, with Red Hat-based systems using systemd extensively, whereas Debian offers more flexibility in choosing init systems.