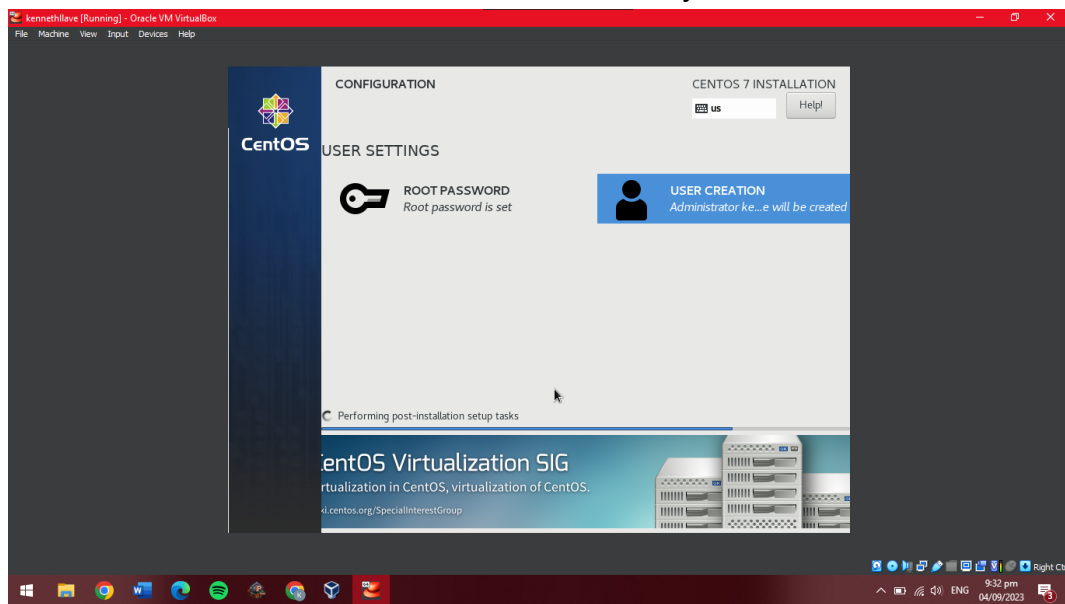


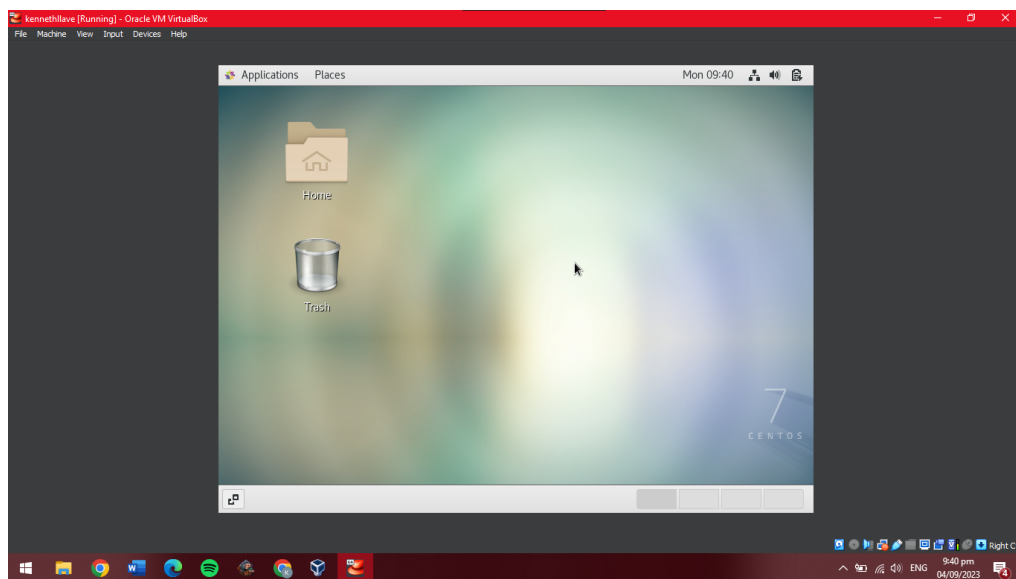
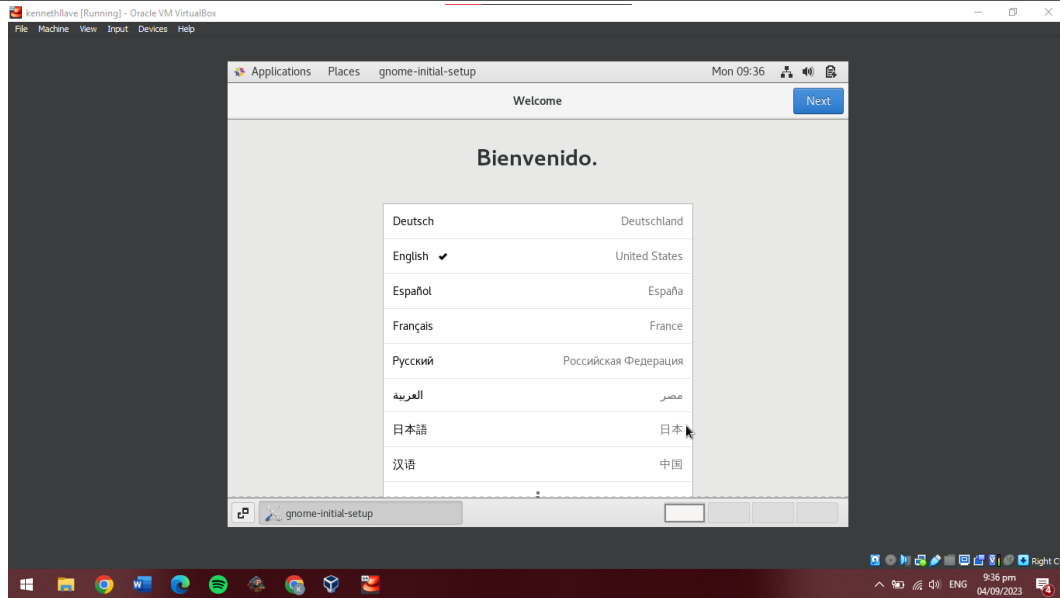
| | |
|--|---|
| Name: Kenneth Gabriel A. Llave | Date Performed: 08/23/2023 |
| Course/Section: CPE 232/CPE31S4 | Date Submitted: 09/04/2023 |
| Instructor: Prof. Johnathan Taylar | Semester and SY: 1 st sem 2023-2024 |
| Activity 3: Install SSH server on CentOS or RHEL 8 | |
| 1. Objectives: 1.1 Install 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/
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.





Successfully 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

```
kennllave@localhost:~  
File Edit View Search Terminal Help  
[kennllave@localhost ~]$ dnf install openssh-server  
Error: This command has to be run under the root user.  
[kennllave@localhost ~]$ sudo dnf install openssh-server  
CentOS-7 - Base 2.8 MB/s | 10 MB 00:03  
CentOS-7 - Updates 4.5 MB/s | 28 MB 00:06  
CentOS-7 - Extras 171 kB/s | 360 kB 00:02  
Package openssh-server-7.4p1-23.el7_9.x86_64 is already installed.  
Dependencies resolved.  
Nothing to do.  
Complete!  
[kennllave@localhost ~]$
```

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

\$ systemctl start sshd

\$ systemctl enable sshd

```
Complete:  
[kennllave@localhost ~]$ systemctl start sshd  
[kennllave@localhost ~]$ systemctl enable sshd  
[kennllave@localhost ~]$
```

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

\$ systemctl status sshd

```
[kennllave@localhost ~]$ 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 Mon 2023-09-04 09:53:26 EDT; 10min ago  
     Docs: man:sshd(8)  
           man:sshd_config(5)  
   Main PID: 19170 (sshd)  
    CGroup: /system.slice/ssh.service  
            └─19170 /usr/sbin/sshd -D  
  
Sep 04 09:53:26 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...  
Sep 04 09:53:26 localhost.localdomain sshd[19170]: Server listening on 0.0.0.0 port 22.  
Sep 04 09:53:26 localhost.localdomain sshd[19170]: Server listening on :: port 22.  
Sep 04 09:53:26 localhost.localdomain systemd[1]: Started OpenSSH server daemon.  
Hint: Some lines were ellipsized, use -l to show in full.  
[kennllave@localhost ~]$
```

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

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

\$ firewall-cmd --reload

```
kennllave@localhost:~  
File Edit View Search Terminal Help  
[kennllave@localhost ~]$ firewall-cmd --zone=public --permanent --add-service=ssh  
Warning: ALREADY_ENABLED: ssh  
success  
[kennllave@localhost ~]$ firewall-cmd --reload  
success  
[kennllave@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

```
File Edit View Search Terminal Help
GNU nano 2.3.1 File: /etc/ssh/sshd_config Modified

# $OpenBSD: sshd_config,v 1.100 2016/08/15 12:32:04 naddy Exp $

# This is an edit of the sshd_config file

# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

-----
[kennllave@localhost ~]$ sudo nano /etc/ssh/sshd_config
[sudo] password for kennllave:
[kennllave@localhost ~]$ systemctl reload sshd
[kennllave@localhost ~]$
```

```
GNU nano 2.3.1 File: /etc/ssh/sshd config

# $OpenBSD: sshd_config,v 1.100 2016/08/15 12:32:04 naddy Exp $

# This is an edit of the sshd_config file

# 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
```

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.

```
[kennethllave@centos ~]$ ssh-copy-id -i ~/.ssh/id_rsa kennethllave@centos
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/kennethllave/.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
kennethllave@centos's password:

Number of key(s) added: 1

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

[kennethllave@centos ~]$
```

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

```
[kennethllave@centos ~]$ ls -la .ssh
total 24
drwx-----. 2 kennethllave kennethllave 103 Sep  5 17:21 .
drwx-----. 15 kennethllave kennethllave 4096 Sep  5 17:16 ..
-rw-----. 1 kennethllave kennethllave 745 Sep  5 17:21 authorized_keys
-rw-----. 1 kennethllave kennethllave 3389 Sep  5 17:16 id_rsa
-rw-r--r--. 1 kennethllave kennethllave 745 Sep  5 17:16 id_rsa.pub
-rw-----. 1 kennethllave kennethllave 816 Sep  5 17:21 known_hosts
-rw-r--r--. 1 kennethllave kennethllave  88 Sep  5 17:19 known_hosts.old
[kennethllave@centos ~]$ |
```

Task 4: Verify ssh remote connection

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

```
kennllave@workstation:~$ ssh kennethllave@centos
kennethllave@centos's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Tue Sep  5 17:14:56 2023 from 192.168.56.103
[kennethllave@centos ~]$
```

2. Show evidence that you are connected.

```
Last login: Tue Sep  5 17:14:56 2023 from 192.168.56.103
[kennethllave@centos ~]$ sudo nano /etc/hostname
```

```
GNU nano 5.6.1 /etc/hostname
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?

We should see if the distribution that we got is appropriate for the use case of the OS. We can see the features of the OS in its downloads page or in the page where the developer shows the uses of the OS.

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

The main difference between Debian and Red Hat linux are the package management for Red Hat uses RPM while Debian uses DEB format. We also can see a difference in their target audience. Red Hat targets enterprise level clients while Debian is more of a personal or educational audience. Red hat also supports distributions for longer and gives security support as well and is patched to keep up with breaches or system risks. Debian on the other hand may produce more versions more frequently, but it may have bugs that you won't see or aren't likely to see on Red Hat products.