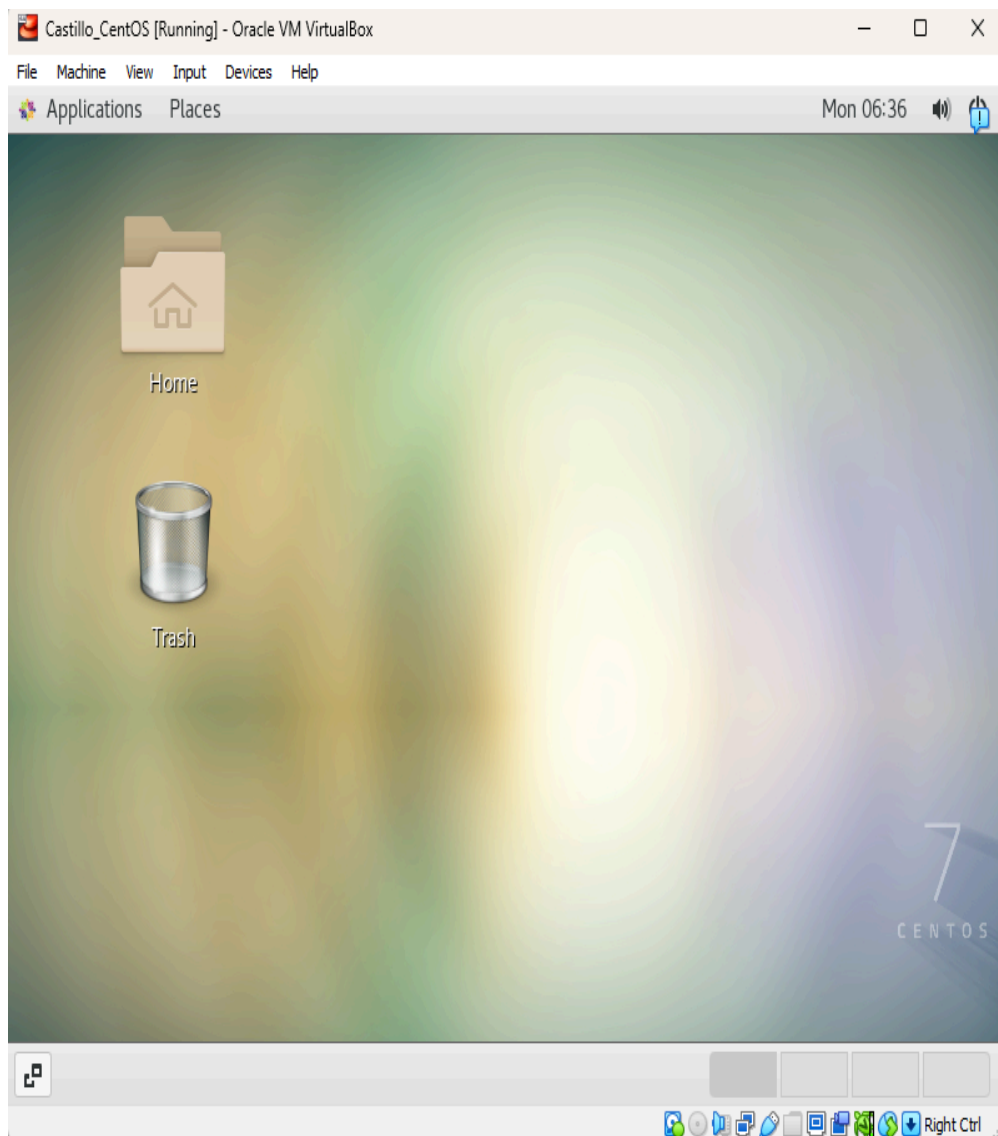


Name: Castillo, Joshua L.	Date Performed: Jan 29, 2024
Course/Section: CPE31S1	Date Submitted: Jan 29, 2024
Instructor: Dr. Jonathan Tylar	Semester and SY: 2023-2024
Activity 3: Install SSH server on CentOS or RHEL 8	
1. Objectives: 1.1 Install Community Enterprise OS or Red Hat Linux OS 1.2 Configure 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)

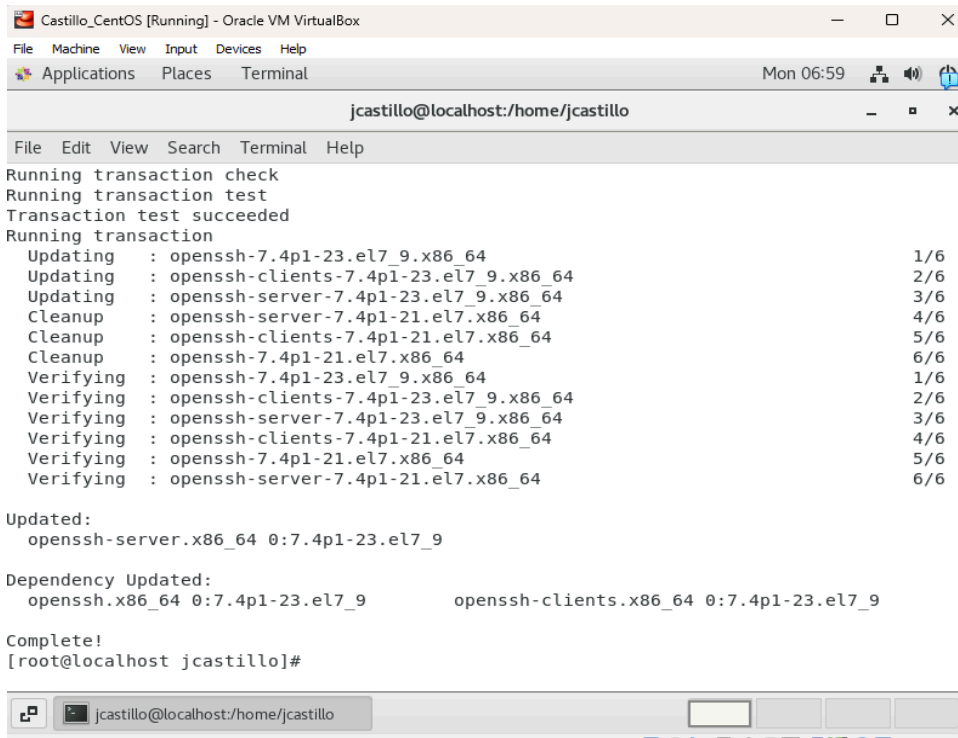
1. Download the image of the CentOS here:
http://mirror.rise.ph/centos/7.9.2009/isos/x86_64/
2. Create a VM machine with 2 Gb RAM and 20 Gb HD.
3. Install the downloaded image.
4. Show evidence that the OS was installed already.



Task 2: Install the SSH server package *openssh*

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

\$ dnf install openssh-server



```
Castillo_CentOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal Mon 06:59
jcastillo@localhost:/home/jcastillo
File Edit View Search Terminal Help
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Updating      : openssh-7.4p1-23.el7_9.x86_64                1/6
  Updating      : openssh-clients-7.4p1-23.el7_9.x86_64        2/6
  Updating      : openssh-server-7.4p1-23.el7_9.x86_64         3/6
  Cleanup       : openssh-server-7.4p1-21.el7.x86_64          4/6
  Cleanup       : openssh-clients-7.4p1-21.el7.x86_64          5/6
  Cleanup       : openssh-7.4p1-21.el7.x86_64                  6/6
  Verifying     : openssh-7.4p1-23.el7_9.x86_64                1/6
  Verifying     : openssh-clients-7.4p1-23.el7_9.x86_64        2/6
  Verifying     : openssh-server-7.4p1-23.el7_9.x86_64         3/6
  Verifying     : openssh-clients-7.4p1-21.el7.x86_64          4/6
  Verifying     : openssh-7.4p1-21.el7.x86_64                  5/6
  Verifying     : openssh-server-7.4p1-21.el7.x86_64           6/6

Updated:
  openssh-server.x86_64 0:7.4p1-23.el7_9

Dependency Updated:
  openssh.x86_64 0:7.4p1-23.el7_9      openssh-clients.x86_64 0:7.4p1-23.el7_9

Complete!
[root@localhost jcastillo]#
```

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

\$ systemctl start sshd

\$ systemctl enable sshd

```
[root@localhost jcastillo]# sudo systemctl start sshd
[root@localhost jcastillo]# sudo systemctl enable sshd
[root@localhost jcastillo]# █
```

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

\$ systemctl status sshd

```
[root@localhost jcastillo]# sudo 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 2024-01-29 06:58:50 EST; 2min 42s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
    Main PID: 3016 (sshd)
      CGroup: /system.slice/ssh.service
              └─3016 /usr/sbin/sshd -D

Jan 29 06:58:50 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Jan 29 06:58:50 localhost.localdomain sshd[3016]: Server listening on 0.0.0.0 port 22.
Jan 29 06:58:50 localhost.localdomain sshd[3016]: Server listening on :: port 22.
Jan 29 06:58:50 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
Hint: Some lines were ellipsized, use -l to show in full.
[root@localhost jcastillo]#
```

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

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

```
$ firewall-cmd --reload
```

```
[root@localhost jcastillo]# sudo firewall-cmd --zone=public --permanent --add-service=ssh
sh
Warning: ALREADY_ENABLED: ssh
success
[root@localhost jcastillo]# sudo firewall-cmd --reload
success
[root@localhost jcastillo]#
```

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@localhost jcastillo]# sudo systemctl reload sshd
[root@localhost jcastillo]# █
```

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

```
joshua@ManagedNode:~$ ssh-copy-id -i ~/.ssh/id_rsa jcastillo@192.168.56.126
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/joshua/.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
jcastillo@192.168.56.126's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'jcastillo@192.168.56.126'"
and check to make sure that only the key(s) you wanted were added.

joshua@ManagedNode:~$ █
```

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

```
[jcastillo@localhost ~]$ cd ~/.ssh
[jcastillo@localhost .ssh]$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQC83IanuqZ1J0rjXx/Zs70BHgZfUN2gVCT9WMom/bIgH
h1oaHcMEdexpHmYhfhsaEDli0k16zHgTY6nY2+8W3M56/j7tAnMFtaY5GFtFwgwJoZQSVm8vf8tcpjZqsg
Hmw1R+AFmDNPet0FX2e3YKSDDbD3I1bIIJEPWrZ+4nNTy/q3plsNCIMTpppDVenTrkn673PtgultXZ22L
CQzdWvEv/fvY65MwCwr4X4KAF9CD1tUXaHiipnrzBRwF4yTs/bQMZYw/Nejk/eqa0iVssfc7r2aeh09ru
dcLDLGz7n3aWrCjKGckGkdFA4tN0wYf2m0vh0Zwgr46SQtK9Nt0AgfQy21eiXsjG78yKyXPdKcCwz90d2
0Hs/mV3F6p5L/LpGhQxbrkNNYwreC50kqmk/km7i/JVNJ4dwplMnENL/YhPVfSQSEvNSTvAPYMsoahB6W
PcU9RmheC4ehy/C6XUjFcyRIgeao4/CkAr+GKPqWx8jpSLMVBMYKyG1C7tMQYqdWY6Hy7MgQGaePunk3U
X3oCA0xl9ej0yBg/0vFEETSG0fw0z5bQNEiPYLC1LA22ur+YUa0EPB+6LY5DL3jURhcqJ9E/A8kIGfa23
6+bKJNUDCoBLr5tFBgDq6ZFJkQ== joshua@ManagedNode
[jcastillo@localhost .ssh]$
```

Task 4: Verify ssh remote connection

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

```
joshua@ManagedNode:~$ ssh jcastillo@192.168.56.126
Last login: Mon Jan 29 07:23:32 2024 from 192.168.56.123
[jcastillo@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? Depending on the purpose, if you want stability you have to use debian which offers reliability to such different environments, on the other hand Red Hat offers complete support which gives long term support and regularity of updates.
2. What are the main differences between Debian and Red Hat Linux distributions?

They have different package management systems, Debian uses APT while Red hat uses yellow dog updater. Commercial support, Debian is maintained by Volunteer community of developers while Red hat is maintained by its company.