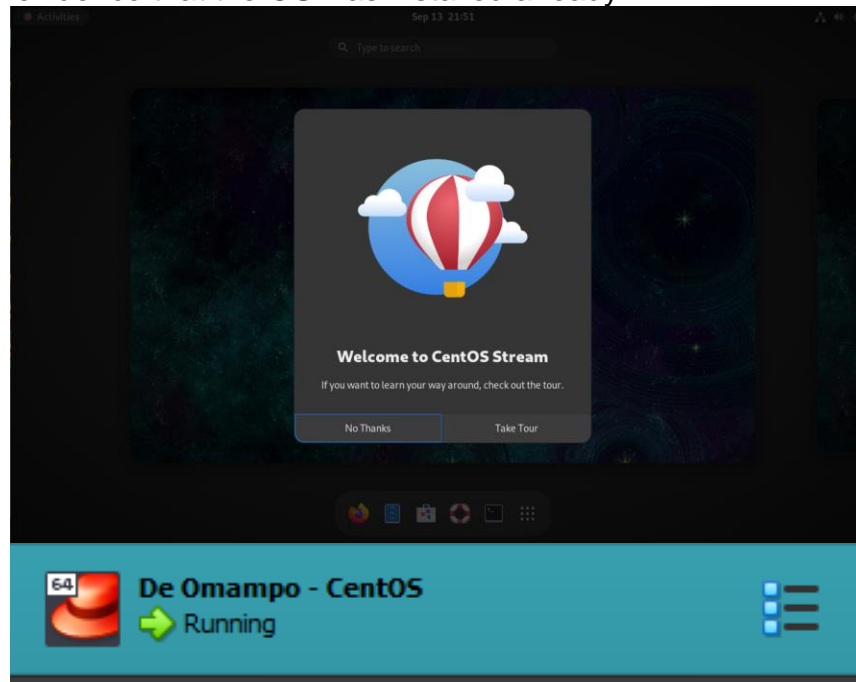


| | |
|--|--|
| Name: Julius Mark A. De Omampo | Date Performed: 09/13/24 |
| Course/Section: CPE212 – CPE31S2 | Date Submitted: 09/13/24 |
| Instructor: Engr. Robin Valenzuela | Semester and SY: 1 st Sem. (2024-2025) |
| Activity 3: Install SSH server on CentOS or RHEL 8 | |
| 1. Objectives: 1.1 Install 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

```
[julius-de-omampo@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" or "subscription-manager" to register.

CentOS Stream 9 - BaseOS                                0.0 B/s | 0 B   00:00
Errors during downloading metadata for repository 'baseos':
 - Curl error (6): Couldn't resolve host name for https://mirrors.centos.org/metalink?repo=centos-baseos-9-stream&arch=x86_64&protocol=https,http [Could not resolve host: mirrors.centos.org]
Error: Failed to download metadata for repo 'baseos': Cannot prepare internal mirrorlist: Curl error (6): Couldn't resolve host name for https://mirrors.centos.org/metalink?repo=centos-baseos-9-stream&arch=x86_64&protocol=https,http [Could not resolve host: mirrors.centos.org]
[julius-de-omampo@localhost ~]$
```

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

```
$ systemctl start sshd  
$ systemctl enable sshd
```

```
[julius-de-omampo@localhost ~]$ systemctl start sshd  
[julius-de-omampo@localhost ~]$ systemctl enable sshd  
[julius-de-omampo@localhost ~]$
```

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

```
$ systemctl status sshd
```

```
[julius-de-omampo@localhost ~]$ systemctl status sshd  
● sshd.service - OpenSSH server daemon  
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)  
   Active: active (running) since Fri 2024-09-13 21:56:14 PST; 13min ago  
     Docs: man:sshd(8)  
           man:sshd_config(5)  
    Main PID: 884 (sshd)  
      Tasks: 1 (limit: 22393)  
     Memory: 2.8M  
        CPU: 11ms  
    CGroup: /system.slice/sshd.service  
            └─884 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"  
  
Sep 13 21:56:14 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...  
Sep 13 21:56:14 localhost.localdomain sshd[884]: Server listening on 0.0.0.0 port 22.  
Sep 13 21:56:14 localhost.localdomain sshd[884]: Server listening on :: port 22.  
Sep 13 21:56:14 localhost.localdomain systemd[1]: Started OpenSSH server daemon.  
[julius-de-omampo@localhost ~]$
```

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

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

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

```

GNU nano 5.6.1                                     sshd_config
# $OpenBSD: sshd_config,v 1.104 2021/07/02 05:11:21 dtucker 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:/usr/local/sbin:/usr/sbin

# 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.

# To modify the system-wide sshd configuration, create a *.conf file under
# /etc/ssh/sshd_config.d/ which will be automatically included below
Include /etc/ssh/sshd_config.d/*.conf

# 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
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

```

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

```

julius-de-omampo@workstation:~$ ssh-copy-id 192.168.56.108
/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: WARNING: All keys were skipped because they already exist
on the remote system.
(if you think this is a mistake, you may want to use -f option)

julius-de-omampo@workstation:~$ S

```

```
GNU nano 7.2 /etc/hosts *
127.0.0.1 localhost
127.0.0.1 juliuslocal

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
192.168.56.104 server1
192.168.56.106 server2
192.168.56.108 servercentos

File Name to Write: /etc/hosts
^C Help M-D DOS Format M-A Append M-B Backup File
^C Cancel M-M Mac Format M-P Prepend ^T Browse

[julius-de-omampo@localhost ~]$ cd .ssh
[julius-de-omampo@localhost .ssh]$ ls
authorized_keys
[julius-de-omampo@localhost .ssh]$ cat authorized_keys
ssh-ed25519 AAAAC3NzaC1lZDIIE5AAAAIM4D9uleXnJ3Lp7xh18KKod0bN7Kg6J9sUx9I18n03qX julius-de-omampo@workstation
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACz8G36SyTWek+n19pmF9u05oxccDNy/rnBXPZUuz4b5T28kIIdN7kxalK3k4HTObXLK7uB19AGWWhD3ihtVrbkmihZonf290xccqyk+RHut7mrja0RM9ey
deul/fbuS/O9wtKenMWPxzZdyXXrvVpUbe/LgR+eMdqLTmT2B2HWPj02zuIQI/rKoXeWXYeNUKAEGUYMw7YE9futX6BEI28QB8dIQ8U2nRFVERnOwQe7ePtww07EvTKln7iC4BYDCvPy360sUGZvm0Y+MPJs
TTyIdevezAb/RfAgbxjR47ZmxZdQYAEAHUMR1Ab9cI7ZcfwShZum9jY/19MN1ShdSALrqlLUL3zcAR2lf21KR95Ezz0BdnTwpQlSvuGrjxtP+OCPJZ/W8FbosGLhhsuImVpSdqVdF9w4VZHM8I31H+J3AZHkO
3v19R+YD8ewctN3EUSw4A3U7YDgm+1R6uvMaENRSY48cLcdQrq2pe2YCeXLSyQ5AG329PH7yP+a/gk7B/W8FRS+btIcKR170H2SANKS11CwnJf2r2nnuRNOgkwKLNPCRR0bG6QePos8PGK8v3wJ43oF6WmckNpN
VE+5bf3s+5IA1oMQUZe388mwa3G9I/rr76rdulVtPNips6Xxawy+P+q7p053Gwr8k5GCK+VYBA8ceASUZKdm5CyKQ== julius-de-omampo@workstation
[julius-de-omampo@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.

```
julius-de-omampo@workstation:~$ ssh julius-de-omampo@servercentos
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Fri Sep 13 22:23:06 2024 from 192.168.56.105
[julius-de-omampo@localhost ~]$

julius-de-omampo@workstation:~$ ping servercentos
PING servercentos (192.168.56.108) 56(84) bytes of data.
64 bytes from servercentos (192.168.56.108): icmp_seq=1 ttl=64 time=0.401 ms
64 bytes from servercentos (192.168.56.108): icmp_seq=2 ttl=64 time=1.03 ms
64 bytes from servercentos (192.168.56.108): icmp_seq=3 ttl=64 time=0.344 ms
64 bytes from servercentos (192.168.56.108): icmp_seq=4 ttl=64 time=0.995 ms
```

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?

Base on my findings, when choosing between the said distributors, we ought to always consider of the following factors: Debian is ideal for users seeking a community-driven, flexible system with extensive packages and a focus on stability, suitable for both personal and small-scale projects. On the other hand, Red Hat is most suitable for enterprises requiring commercial support, long-term stability, and certification, which offers robust management tools, regular security checks, and professional support. In the end, it really depends on what particular purpose you intend to do with the said distributor products.

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

These distributions differ primarily in their package management, release cycles, and target audiences. Debian uses APT with .deb packages and offers a broad repository with a flexible release cycle across stable, testing, and unstable branches. Moreover, it is community-driven and suited for general use. However, Red Hat uses YUM or dnf with .rpm packages, which focus on long-term stability and enterprise environments through RHEL with commercial support and certification. Red Hat provides robust enterprise tools and professional support, making it ideal for mission-critical applications.