Name: Ralph Andrei M. Diocera	Date Performed: 8/29/2023
Course/Section: CPE232/CPE31S4	Date Submitted:
Instructor: Dr. Jonathan V. Taylar	Semester and SY: 1st/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.

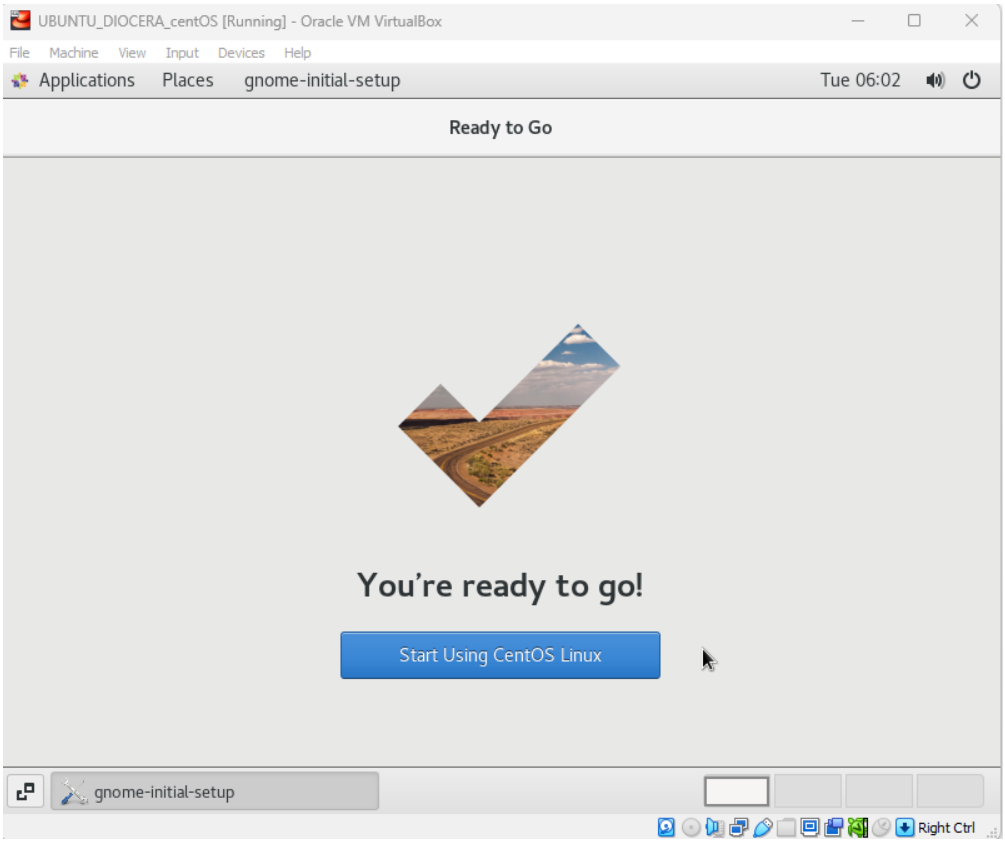
CentOS uses the RPM package format and YUM/DNF as the package manager.

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

```
[root@localhost ralph]# yum install openssh-server
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirror.aktkn.sg
 * extras: mirror.aktkn.sg
 * updates: mirror.aktkn.sg
Package openssh-server-7.4p1-23.el7_9.x86_64 already installed and latest version
Nothing to do
[root@localhost ralph]#
```

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

```
[root@localhost ralph]# systemctl start sshd
[root@localhost ralph]# systemctl enable sshd
[root@localhost ralph]#
```

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

```
[root@localhost ralph]# systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enable
d)
   Active: active (running) since Sat 2023-09-02 02:35:45 EDT; 2min 14s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 1138 (sshd)
    Tasks: 1
   CGroup: /system.slice/sshd.service
           └─1138 /usr/sbin/sshd -D

Sep 02 02:35:45 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Sep 02 02:35:45 localhost.localdomain sshd[1138]: Server listening on 0.0.0.0 port 22.
Sep 02 02:35:45 localhost.localdomain sshd[1138]: Server listening on :: port 22.
Sep 02 02:35:45 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
Hint: Some lines were ellipsized, use -l to show in full.
[root@localhost ralph]#
```

4. Open the SSH port 22 to allow incoming traffic:
   $ firewall-cmd --zone=public --permanent --add-service=ssh
   $ firewall-cmd --reload

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

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 ralph]# man /etc/ssh/sshd_config

[1]+  Stopped                 man /etc/ssh/sshd_config
[root@localhost ralph]# systenctl reload sshd
bash: systenctl: command not found...
[root@localhost ralph]# systemctl reload sshd
[root@localhost ralph]#
```

Task 3: Copy the Public Key to CentOS

1. Make sure that *ssh* is installed on the local machine.

```
ralph@manageNode:~$ ssh 'ralph@manageNode'
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-31-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Sat Sep  2 21:18:24 2023 from 10.0.2.15
ralph@manageNode:~$
```

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

```
ralph@manageNode:~$ ssh-copy-id ralph@192.168.56.104
The authenticity of host '192.168.56.104 (192.168.56.104)' can't be established.
ED25519 key fingerprint is SHA256:9IfOhcMyNeUcKCl8TVzI0EiF2sTQLwivXZ2MDvhB3c8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[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 prompt
ed now it is to install the new keys
ralph@192.168.56.104's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'ralph@192.168.56.104'"
and check to make sure that only the key(s) you wanted were added.

ralph@manageNode:~$ ssh 'ralph@192.168.56.104'
Last login: Sat Sep  2 10:10:04 2023
[ralph@localhost ~]$ logout
Connection to 192.168.56.104 closed.
ralph@manageNode:~$
```

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

```
[ralph@localhost ~]$ cat ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQCqIKjc/KlQs4F5xirq+v+oby4TvT0E3WcknR394iu5nTW0cKI
P06t0whETbdNBKChueBpjFGjjRl2mC21paNCO9Vqj7pRans5pYJ1ThLueImmbVR4vGwVYgb/2uyvaZlNvKMvdfU
h5lxF+swwIx9F15/W6xp3s/pEcpTwNqJmSn/0RYQrPLbh8KduK+BnZSMWOCCRUDhyqBfL095ZcVn5xtC0m2xbBP
nkXEWMROTMC0t88OrVm1eKR2VDFwiTwq4DluwHCZDSmEUR5pMiS3LoCFsD4iCxDwJU9uzhj+BwYGRIV1BIPDhNn
1LQl1S5k7/Y0IYGeaix3HzuqnAffn4530DOeEEP/Eh0A1MZIgTN7QRCwP6fj6D3Z4Qij6TvxQsASqRg5dPe6uDc
FvL9ICW6/BAM14VC4k23HAJIDSc16FbAvUu1s9lnWr+2Et5MfacAQ/E/eWpi08DPVs/orNoLru5+TKuC4FZusQx
NWX+ovpXkxP0eVvSNBaN9EBjU5YS+fv8DyZ7ElLmh5X2s26Jpj1qHOtDlgJiLcBgr7+OctpdPVhQ1TTh0JSEM+r
bl1JPi72VrDtBXX3lcbNN20gPn/RKQVMHRL2DiFH/Jj75D4H6bLrfHxThLd9wnfFQRvm2hB0ZEjVG/0xq6qVMi+
ns2hVnWmpYWEt7XgZsXlO3iKTw== ralph@manageNode
[ralph@localhost ~]$
```

Task 4: Verify ssh remote connection

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

```
ralph@manageNode:~$ ssh 'ralph@192.168.56.104'
Last login: Sat Sep  2 10:10:04 2023
[ralph@localhost ~]$ logout
Connection to 192.168.56.104 closed.
```

2. Show evidence that you are connected.

```
ralph@manageNode:~$ ping 192.168.56.104
PING 192.168.56.104 (192.168.56.104) 56(84) bytes of data.
64 bytes from 192.168.56.104: icmp_seq=1 ttl=64 time=1.05 ms
64 bytes from 192.168.56.104: icmp_seq=2 ttl=64 time=0.565 ms
64 bytes from 192.168.56.104: icmp_seq=3 ttl=64 time=0.542 ms
64 bytes from 192.168.56.104: icmp_seq=4 ttl=64 time=0.582 ms
64 bytes from 192.168.56.104: icmp_seq=5 ttl=64 time=0.593 ms
64 bytes from 192.168.56.104: icmp_seq=6 ttl=64 time=0.565 ms
64 bytes from 192.168.56.104: icmp_seq=7 ttl=64 time=0.586 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?
   - When it comes to choosing the best distribution between Debian and RedHat Linux, it is better to consider firstly the availability or Licensing of a distribution from a specific Linux platform whereas Debian offers a free-to-use distribution that provides accessible software products that lets a user access a specific licensed application without any restrictions. Unlike Red Hat, this platform offers open-source products that require users to buy, especially the additional features it offers within the distributions. The second thing we should look for based on the question is the packages it provides whereas Debian has a lot more packages than Red Hat Linux giving off better libraries we can use and manipulate for a specific situation. And lastly, we should consider how secure it is because based on these two platforms, Red Hat has better support than Debian for it gives robust security and confidentiality of specific data.
2. What are the main differences between Debian and Red Hat Linux distributions?

- When it comes to the distributions of Debian and Red Hat Linux, there's really a big difference between the two of them whereas the Debian distribution is easier to use because it's a community-driven project which makes it entirely free while the Red Hat one is a commercial product. The package management they both use is entirely different whereas Debian uses APT or Advanced Package Tool which is a free software interface that mainly installs, updates, deletes, and manages packages within the software while Red Hat uses RPM or Red Hat Package Manager is kind of the same with the APT but it can only be used on a Red Hat distribution.

Conclusion

- **In conclusion, this hands-on activity has given me a lot of knowledge about the entire purpose of SSH. The SSH or Secure Shell has the purpose of letting the user configure the connections between two or more servers which requires a public or private key for authentication that specifically has the main node manage different data of its sub-nodes just like how we have simulated on connecting the main node to the CentOS using its IP address. I have also learned that the SSH has proven its stability for how secure it is which makes it more convenient on being used to connect and access the CentOS server remotely and daily updates all of the system that revolves around the SSH configuration.**