Name: Paz, John Glen L.	Date Performed: 9/7/2023
Course/Section: CPE31S6	Date Submitted: 9/7/2023
Instructor: Dr. Jonathan V. Taylar	Semester and SY: 1st 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.

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 ~]# dnf install openssh-server
CentOS-7 - Base                                  1.0 MB/s |  10 MB     00:09
CentOS-7 - Updates                               1.9 MB/s |  28 MB     00:14
CentOS-7 - Extras                                678 kB/s | 360 kB     00:00
Package openssh-server-7.4p1-21.el7.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@localhost ~]#
```

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

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

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

```
[root@localhost ~]# 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 Thu 2023-09-07 05:34:14 EDT; 3min 33s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 1138 (sshd)
   CGroup: /system.slice/sshd.service
           └─1138 /usr/sbin/sshd -D

Sep 07 05:34:14 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Sep 07 05:34:14 localhost.localdomain sshd[1138]: Server listening on 0.0.0.0 port 22.
Sep 07 05:34:14 localhost.localdomain sshd[1138]: Server listening on :: port 22.
Sep 07 05:34:14 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
Hint: Some lines were ellipsized, use -l to show in full.
```

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

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

$ firewall-cmd --reload

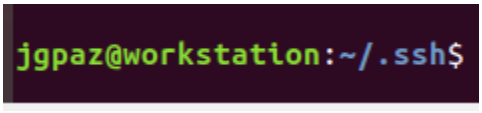
```
[root@localhost ~]# firewall-cmd --reload
success
```

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

```
[root@localhost ~]# systemctl reload sshd
```

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.

```
jgpaz@workstation:~/.ssh$ ssh-copy-id -i ~/.ssh/id_rsa jpaz@192.168.56.104
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/jgpaz/.ssh
/id_rsa.pub"
The authenticity of host '192.168.56.104 (192.168.56.104)' can't be established
.
ECDSA key fingerprint is SHA256:O8IaWi6rPVB/l9uh0suKIfJBy3F3NyRZrkTdVtZEYNQ.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': 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
```

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

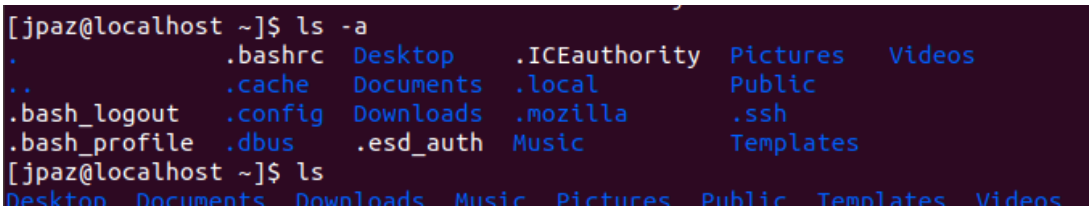
```
[jpaz@localhost ~]$ cat ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDesVGFW7i1aAOLVccIicFN/IHyprc9ptUyy4rjA+xZVEgq7Wp
u0BNaSdjtwu+4D7DAVXO5hj+h4Ag1TfXVP6CzxIn2R7wrp/AoIrVlxxxqDIrLt/zdeP4jzussdxE3tqnY9yoWA7
rPpYln5hrG1lFOjGoRib/Cb+XJMnrzGqMxAt8yGq0RqR819qEi7jLmjFK1bDGs0T1zUXNo2CrECrOOOnkGkyLEK
QMoRrrylhwCuqctkRPnAz5z85TyzINdIw+ObtJuzvOGrF3GHJyuuqXLFr1yJqZKQNEZawg5omH/0VS0JLsh9AKq
BKVlTtPyGzvDQr7CNQW6mLSTQEsPy1xfW6rBXggAhBR5kqiN7zJoiO+Pi6PWxyyzLI5hIRVUMDwF5IL+DYQ2DNQ
tes93K/sJuTTD3gjWaCml5qcqUpJwfYKaaRGg4J7olMhyV0m10jqXCjexyH+5CLPJQ8a3JwQJLWi3/Bi0qsfyac
CQL9mo/cajQlJEPDi1gv1Z4iH1JdeqQSl6NVmKpKYBejnHFGE4jq7LSunmMeOlpEWYFkUGuJzMA2nuA8W9wd9iN
4ZQgCHqJq3bcYWr5vnoCmVtkYVlxZI88PvOhjBnskK8P3iAnjVa8A8nX2ixH20TKtZfhZySYjWsHkQwbewH08CA
caG13OAu/0jHD+Dc4GpwEVX+Ow== jgpaz@workstation
```

Task 4: Verify ssh remote connection

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

```
jgpaz@workstation:~$ ssh jpaz@192.168.56.104
Last login: Thu Sep  7 05:35:20 2023
[jpaz@localhost ~]$ ls la
```

2. Show evidence that you are connected.



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?
   - The choice between Debian and Red Hat Linux distributions depends on your specific use case, budget, and preference for free software or enterprise-grade solutions. If you prioritize stability and reliability, Debian is a great choice, while RHEL is a popular choice for enterprise environments that require robust security and excellent enterprise support.

2. What are the main difference between Debian and Red Hat Linux distributions?
   Both of them have different target audiences, support structures, release cycles, and philosophical perspectives. Your individual use case, financial situation, and taste for free software or enterprise-level solutions will determine which one you should employ.

Conclusion:

In this activity, I learned that hen it comes to configuring remote SSH connection from a remote computer to CentOS/RHEL-8, both distributions offer SSH as a secure method to remotely manage and administer a system. To configure remote SSH connection, you need to install the openssh-server package on the server and openssh-client package on the client. Once installed, you can access the server with most terminal applications that support the SSH protocol. Red Hat Enterprise Linux also offers the RHEL web console, which provides a graphical interface for managing remote systems over SSH.