|  |  |
| --- | --- |
| **Name: Maravilla, Keith Dominic T.** | **Date Performed: August 18, 2022** |
| **Course/Section: CPE232/CPE31S23** | **Date Submitted: August 22, 2022** |
| **Instructor: Engr. Jonathan V. Taylar** | **Semester and SY: 1st Sem - 3rd Year** |
| **Activity 1: Configure Network using Virtual Machines** | |
| 1. **Objectives:**    1. Create and configure Virtual Machines in Microsoft Azure or VirtualBox    2. Set-up a Virtual Network and Test Connectivity of VMs | |
| 1. **Discussion:**   **Network Topology:**  Assume that you have created the following network topology in Virtual Machines, *provide screenshots for each task*. (Note: *it is assumed that you have the prior knowledge of cloning and creating snapshots in a virtual machine*). | |
| **Task 1**: Do the following on Server 1, Server 2, and Local Machine. In editing the file using nano command, press control + O to write out (save the file). Press enter when asked for the name of the file. Press control + X to end.   * 1. Change the hostname using the command *sudo nano /etc/hostname*   2. Use server1 for Server 1      * 1. Use server2 for Server 2      * 1. Use workstation for the Local Machine  1. Edit the hosts using the command *sudo nano /etc/hosts.* Edit the second line.    1. Type 127.0.0.1 server 1 for Server 1      * 1. Type 127.0.0.1 server 2 for Server 2      * 1. Type 127.0.0.1 workstation for the Local Machine | |
| **Task 2**: Configure SSH on Server 1, Server 2, and Local Machine. Do the following:   * 1. Upgrade the packages by issuing the command *sudo apt update* and *sudo apt upgrade* respectively.       ***Initializing update and upgrade of packages in Server 1***  ***(NOTE: This will be the same output for Server 2)***   * 1. Install the SSH server using the command *sudo apt install openssh-server*.     ***Installing the SSH server in Server 1***  ***(NOTE: This will be the same output for Server 2)***   * 1. Verify if the SSH service has started by issuing the following commands:   2. *sudo service ssh start*   3. *sudo systemctl status ssh*     ***Verifying the SSH service in Server 1***  ***(NOTE: This will be the same output for Server 2)***   * 1. Configure the firewall to all port 22 by issuing the following commands:   2. *sudo ufw allow ssh*   3. *sudo ufw enable*   4. *sudo ufw status*     ***Activating or enabling firewall in port 22 in Server 1***  ***(NOTE: This will be the same output for Server 2)*** | |
| **Task 3:** Verify network settings on Server 1, Server 2, and Local Machine. On each device, do the following:   * 1. Record the ip address of Server 1, Server 2, and Local Machine. Issue the command *ifconfig* and check network settings. Note that the ip addresses of all the machines are in this network 192.168.56.XX.   2. Server 1 IP address: 192.168.56.103      * 1. Server 2 IP address: 192.168.56.104      * 1. Server 3 IP address: 192.168.56.\_\_\_   2. Make sure that they can ping each other.   3. Connectivity test for Local Machine 1 to Server 1:  Successful  Not Successful      * 1. Connectivity test for Local Machine 1 to Server 2:  Successful  Not Successful      * 1. Connectivity test for Server 1 to Server 2:  Successful  Not Successful | |
| **Task 4:** Verify SSH connectivity on Server 1, Server 2, and Local Machine.   1. On the Local Machine, issue the following commands:    1. ssh username@ip\_address\_server1 for example, *ssh [jvtaylar@192.168.56.120](mailto:jvtaylar@192.168.56.120)*     ***Accessing SSH server of Server 1 through local machine***   * 1. Enter the password for server 1 when prompted     ***Entering the password for server 1 using local machine***   * 1. Verify that you are in server 1. The user should be in this format user@server1. For example, *jvtaylar@server1*     ***Verification that local machine is connected to server 1***   1. Logout of Server 1 by issuing the command *control + D.*     ***Logging out to server 1 in a local machine***   1. Do the same for Server 2.     ***Verification that local machine is connected to server 2***   1. Edit the hosts of the Local Machine by issuing the command *sudo nano /etc/hosts.* Below all texts type the following:    1. IP\_address server 1 (provide the ip address of server 1 followed by the hostname)      * 1. IP\_address server 2 (provide the ip address of server 2 followed by the hostname)      * 1. Save the file and exit.  1. On the local machine, verify that you can do the SSH command but this time, use the hostname instead of typing the IP address of the servers. For example, try to do *ssh jvtaylar@server1*. Enter the password when prompted. Verify that you have entered Server 1. Do the same for Server 2.       ***Issuing and verifying SSH command using hostname*** | |
| **Reflections:**  Answer the following:   * 1. How are we able to use the hostname instead of IP address in SSH commands?   We were able to use the hostname in SSH commands instead of using IP address because there is an addition of hostname associated to the IP address of a certain server in /etc/hosts. In this way, we could access and issue SSH commands of a server by simply typing the hostname since it was registered or added in the registry or file containing the IP addresses along with its server or unit.   * 1. How secured is SSH?   Secure Shell or SSH is highly secured as it uses encryption and authentication to all available connections. Another reason why SSH is highly secured is due to remote managing of SSH clients. Using secure or SSH keys/credentials for SSH connections basically tells us that accessing SSH connection between devices is not easy. However, the moment SSH will not be secured if it is not properly managed which causes a typical SSH brute force attacks. | |

**Conclusion/Learning:**

**After doing this activity, I learned how to clone Virtual Machines for different servers to work on. I also learned and observed how to test the connection between a local machine and server/s. In addition to that, I also learned how to use and issue SSH commands in a local machine and I was able to verify connections between those systems. Overall, this activity gave me a brief idea on what our next topic will be and this serves as a preparation to manage SSH connections.**

***I affirm that I will not give or receive any unauthorized help on this activity/exam and that all work will be my own.***

***-Keith Maravilla***