

Lab #2

Class: CECS 303 – Networks and Network Security

Instructor: Chris Samayoa

Due Date: February 16, 2022 by 9pm PST

Objective: Prepare and test telnet and web services

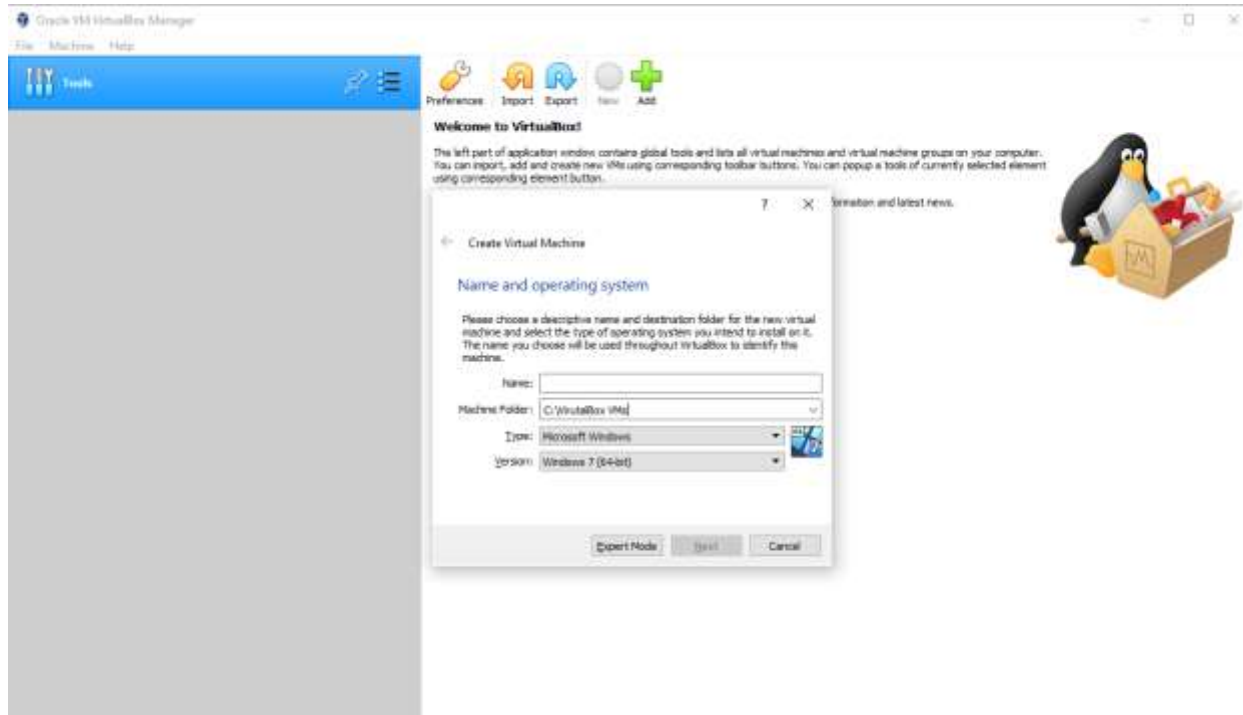
Links:

- VirtualBox: <https://www.virtualbox.org/wiki/Downloads>
- Ubuntu: <https://ubuntu.com/download/server> or <https://ubuntu.com/download/desktop>

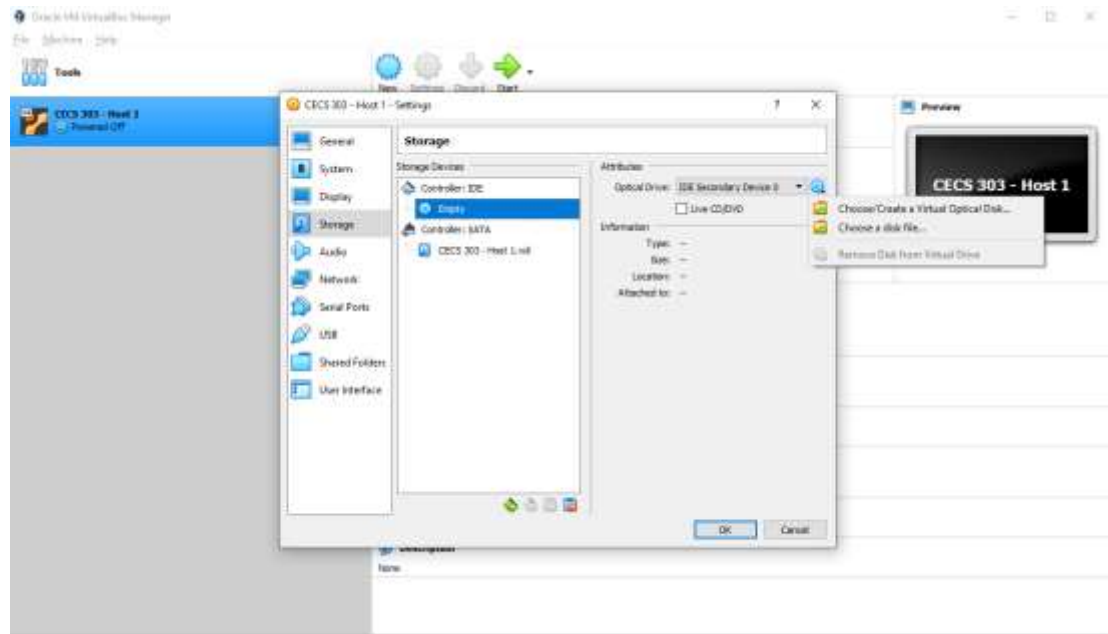
Prepare one (1) Ubuntu Workstation Instance:

Needs to be able to communicate with previously configured server instances

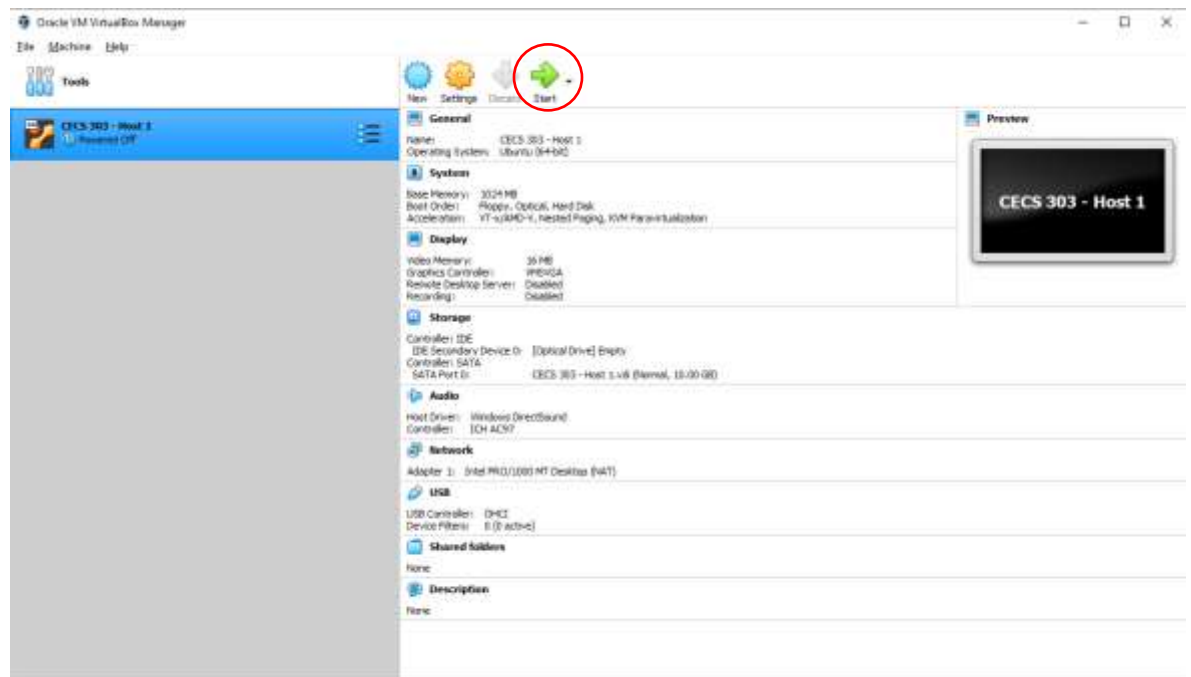
1. Download Ubuntu Desktop ISO (version 20.04.3 LTS)
2. Open VirtualBox application
 - a. Select “Machine -> New” from top level menu
 - i. Name: Provide unique name
 - ii. Machine Folder: Leave default or choose a different location
 - iii. Type: Linux
 - iv. Version: Ubuntu (64-bit)



- b.
- c. Memory Size: Default value is 1024
- d. Create Virtual Hard Disk: Default size of 10gb is fine (recommend 20gb if hard disk space is available on host)
- e. Leave 'Hard Disk File Type' and 'Storage on Physical Hard Disk' at default values
- f. Complete Setup
3. Change host network settings
 - a. Select host
 - b. Click 'Machine -> Settings'
 - c. Select 'Network' in left menu
 - d. Change 'Attached to:' drop down menu to 'Bridged Adapter'
 - i. Ensure that your active network device is selected under 'Name'
 - e. Click 'OK' on bottom to close
4. Load ISO (Ubuntu) Image
 - a. Go back to host settings
 - b. Select 'Storage'
 - i. Click on the 'Empty' device and select the optical disk icon on the right
 - ii. Select 'Choose a disk file' (see screenshot below)



- c.
 - d. Navigate to and select Ubuntu ISO installation file downloaded earlier
5. Start the host



- a.
6. Proceed with installing the Ubuntu operating system
- a. For the purposes of this lab the defaults work
 - b. You'll need to configure a host name, username, and password
 - i. BE SURE TO TAKE NOTE OF USERNAME and PASSWORD USED

- c. Once the installation is complete you will need to shut down the virtual machine in order for the Ubuntu ISO to unmount
 - i. Click 'File -> Close' and select 'Power off the machine'
7. Start Host again and login with username and password
8. Open terminal and Run command "sudo apt install net-tools"
 - a. Once this is complete, you should be able to reach the internet
 - b. Test by using ping
 - i. e.g. "ping -4 -c 4 google.com" and ensure you receive a response

```

user1@cecshost1:~$ ping -c 4 google.com
PING google.com (142.250.68.14) 56(84) bytes of data:
64 bytes from lax17s44-in-f14.1e100.net (142.250.68.14): icmp_seq=1 ttl=115 time=13.1 ms
64 bytes from lax17s44-in-f14.1e100.net (142.250.68.14): icmp_seq=2 ttl=115 time=12.8 ms
64 bytes from lax17s44-in-f14.1e100.net (142.250.68.14): icmp_seq=3 ttl=115 time=10.6 ms
64 bytes from lax17s44-in-f14.1e100.net (142.250.68.14): icmp_seq=4 ttl=115 time=11.1 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 10.584/11.919/13.121/1.085 ms
user1@cecshost1:~$ _

```

- ii.
9. Run command 'ifconfig' and take note of your host's IP address

```

user1@cecshost1:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.4.46 netmask 255.255.252.0 broadcast 192.168.7.255
    inet6 fe80::a00:27ff:fe12:cbcd prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:12:cb:cd txqueuelen 1000 (Ethernet)
    RX packets 50 bytes 13795 (13.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 50 bytes 5557 (5.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

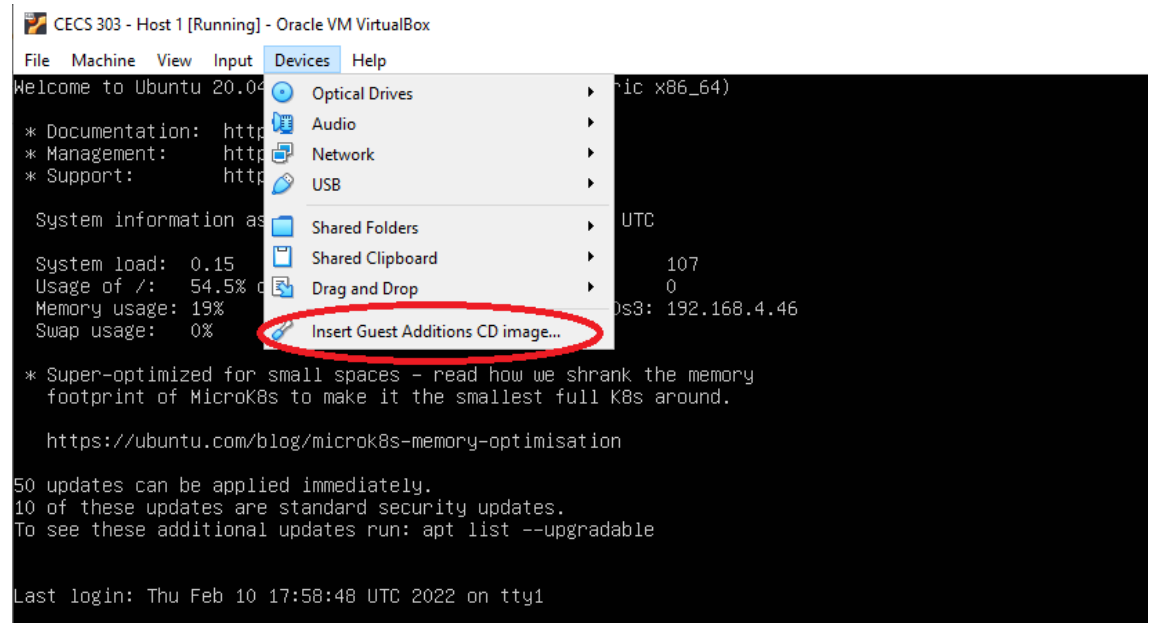
- a.
 - b. e.g. This host has an IP address of 192.168.4.46

Ubuntu Desktop – Install VirtualBox guest additions (OPTIONAL):

Installing the VirtualBox guest additions on Ubuntu Desktop will allow for the screen (and resolution) to scale based on your preference. I recommend installing these additions in order to make the Ubuntu Desktop more useable

1. Open a terminal (command prompt console)
 - a. Keyboard shortcut for this is Ctrl+Alt+T
2. Prepare for installation:
 - a. Run command 'sudo apt update'

- b. Run command 'sudo apt install -y build-essential linux-headers-\$(uname -r)' (note there is no space between "headers-" and "\$" in the command)
3. On VM window, select 'Devices -> Insert Guest Additions CD Image'



- a.
4. Mount ISO image in terminal
 - a. Command 'sudo mount '/dev/cdrom /media'' (note the empty space after "cdrom ")
5. Execute the installer from the mounted media
 - a. Command 'sudo ./VBoxLinuxAdditions.run'
6. Reboot to complete installation: 'sudo reboot'

Install Telnet Server package

Install the Telnet Server package on both Ubuntu server VMs created during Lab #1

1. Install the package
 - a. 'sudo apt update'
 - b. 'sudo apt install telnetd -y'
2. Check the status of the Telnet service after installation completes
 - a. 'sudo systemctl status inetd'
3. Repeat for other VM

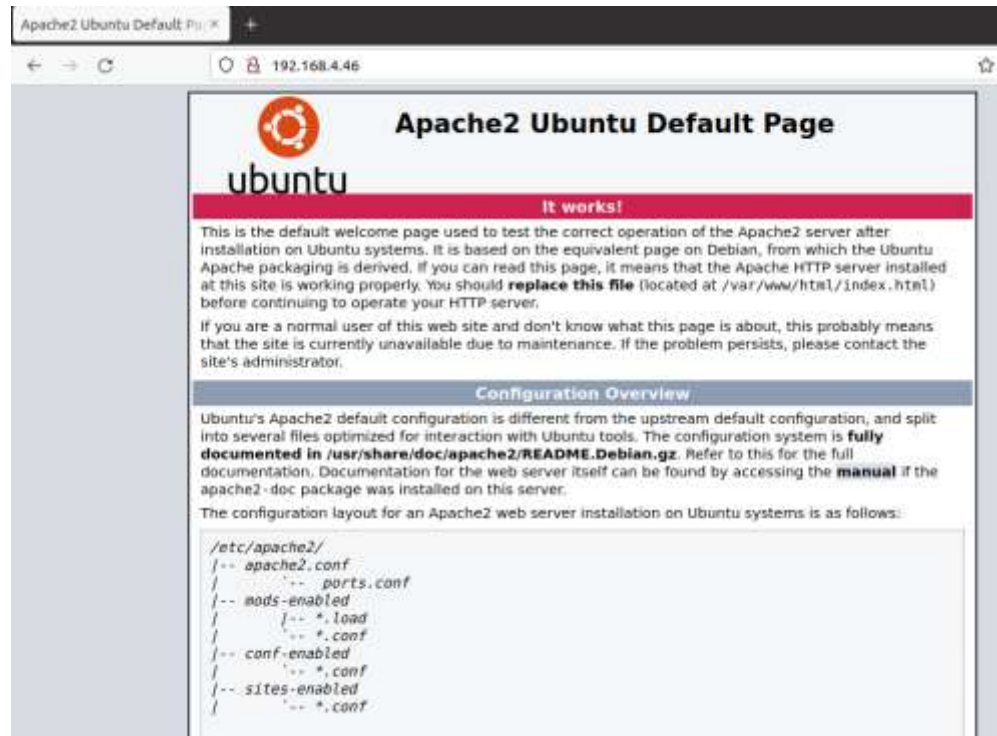
Install and Test Apache on one server

Install Apache on one of the two virtual machines that were previously created in Lab #1

1. Install Apache software/service
 - a. Run command 'sudo apt update'
 - i. This ensures that the current package information from configured sources are correct
 - b. Run command 'sudo apt install apache2'
2. Check that apache server is running now:
 - a. Command: 'sudo systemctl status apache2'

```
user1@cecs-host1:~$ sudo systemctl status apache2
• apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-02-10 18:39:06 UTC; 5min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2079 (apache2)
    Tasks: 55 (limit: 1066)
   Memory: 5.6M
   CGroup: /system.slice/apache2.service
           └─2079 /usr/sbin/apache2 -k start
             └─2082 /usr/sbin/apache2 -k start
               └─2083 /usr/sbin/apache2 -k start
```

- b.
 - c. Ensure that the service is "active (running)"
3. Test by visiting the Apache2 default page
 - a. Using a web browser in your newly configured Ubuntu Desktop instance or directly from your laptop browse to the IP address of the virtual machine where Apache was just installed
 - b. The "Apache2 Ubuntu Default Page" should show:



- i.
- 4. Test using telnet client
 - a. Ensure that both of your VMs created during Lab #1 are on and that you have both IP addresses available
 - i. Note: One of these VMs should now have Apache installed
 - b. Ping both VMs from newly created Ubuntu Desktop VM to ensure connectivity

```
wsuser1@cecs303-ws1:~$ ping 192.168.4.47
PING 192.168.4.47 (192.168.4.47) 56(84) bytes of data.
64 bytes from 192.168.4.47: icmp_seq=1 ttl=64 time=0.446 ms
64 bytes from 192.168.4.47: icmp_seq=2 ttl=64 time=0.289 ms
^C
--- 192.168.4.47 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1018ms
rtt min/avg/max/mdev = 0.289/0.367/0.446/0.078 ms
wsuser1@cecs303-ws1:~$ ping 192.168.4.46
PING 192.168.4.46 (192.168.4.46) 56(84) bytes of data.
64 bytes from 192.168.4.46: icmp_seq=1 ttl=64 time=0.332 ms
64 bytes from 192.168.4.46: icmp_seq=2 ttl=64 time=0.311 ms
^C
--- 192.168.4.46 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1032ms
rtt min/avg/max/mdev = 0.311/0.321/0.332/0.010 ms
```

- i.
- c. From new VM, test telnet connection to the Ubuntu server without Apache installed

- i. Note that Port 80 is specified after the basic telnet command in order to instruct telnet to use the common TCP/HTTP port.
- ii. 'telnet <server ip> 80'
- iii. Connection should fail

```
wsuser1@cecs303-ws1:~$ telnet 192.168.4.47 80
Trying 192.168.4.47...
telnet: Unable to connect to remote host: Connection refused
```

- iv.
- d. Now test telnet connection to Ubuntu server with Apache installed:

- i. 'telnet <server ip> 80'
- ii. Connection should succeed

```
wsuser1@cecs303-ws1:~$ telnet 192.168.4.46 80
Trying 192.168.4.46...
Connected to 192.168.4.46.
Escape character is '^['.
```

- iii.

5. Change default web page

- a. Switch back to your Apache server
- b. Navigate to default directory: 'cd /var/www/html'
- c. Backup default index.html file: 'sudo cp index.html index.bak'
- d. Command 'ls' should now show two files in the directory

```
user1@cecshost1:/var/www/html$ ls
index.bak  index.html
user1@cecshost1:/var/www/html$
```

- i.
- e. Delete current index.html file: 'sudo rm index.html'
- i. 'ls' command should now only show index.bak in the directory
- f. Create and edit new index.html page using nano or vim editors (below example shows use of vim)
- i. 'sudo vi index.html'
- 1. Be sure to use sudo here or the permissions will not allow you to save the file after you have modified it
- ii. Type 'i' to switch into insert mode so you can edit the file
- iii. Enter the following text (but replace my name with yours)


```
<!DOCTYPE html>
<html>
<body>

<h1>CECS 303 - Chris Samayoa</h1>

<p>Lab #2 assignment - Chris Samayoa</p>

</body>
</html>
```

1.
 - iv. Press escape ('Esc') key on keyboard to get out of insert mode
 - v. Type ':wq' to save and exit the file
 1. If this fails, then type ':wq!' to force quit
 - vi. Command 'cat index.html' should show you an output similar to the previous screenshot
- g. Switch back to new Ubuntu Desktop VM. Go back to browser previously used and either refresh the page or reenter the Apache server's IP address in the url bar
 - i. Your new page should now show:



- ii.
6. Let's test telnet again and manually retrieve the new index.html webpage:
 - a. From the new Ubuntu Desktop VM, telnet to the new apache server again using Port 80
 - b. After a connection is established, enter command: 'GET /index.html'
 - i. Your new index.html file contents should be displayed

```
wsuser1@cecs303-ws1:~$ telnet 192.168.4.46 80
Trying 192.168.4.46...
Connected to 192.168.4.46.
Escape character is '^]'.
GET /index.html
<!DOCTYPE html>
<html>
<body>

<h1>CECS 303 - Chris Samayoa</h1>

<p>Lab #2 assignment - Chris Samayoa</p>

</body>
</html>
Connection closed by foreign host.
wsuser1@cecs303-ws1:~$
```

ii.

iii. *Screenshot this output for your submission*

7. The webpage details can also be pulled manually using curl

a. Install curl using 'sudo apt install curl'

b. View webpage

i. 'curl <server IP address>'

c. View other server details:

i. 'curl -I <server IP address>'

```
wsuser1@cecs303-ws1:~$ curl -I 192.168.4.46
HTTP/1.1 200 OK
Date: Thu, 10 Feb 2022 20:53:41 GMT
Server: Apache/2.4.41 (Ubuntu)
Last-Modified: Thu, 10 Feb 2022 19:19:12 GMT
ETag: "7c-5d7aed34b3c02"
Accept-Ranges: bytes
Content-Length: 124
Vary: Accept-Encoding
Content-Type: text/html
```

ii.

iii. *Screenshot this output for your submission*

Modify local Hosts file:

By modifying the local hosts file on the new Ubuntu Desktop VM, you can reach your new Apache web server with a common name instead of the IPv4 IP address:

1. View your current hosts file configuration

- a. 'cat /etc/hosts'

```
wsuser1@cecs303-ws1:~$ cat /etc/hosts
127.0.0.1    localhost
127.0.1.1    cecs303-ws1

# 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
```

- b. wsuser1@cecs303-ws1:~\$

2. Modify the hosts file to give a common name to your Apache web server when accessing from your new Ubuntu Desktop VM

- a. 'sudo vi /etc/hosts'

- i. Modify the file to add a record for your new name server. IP address goes on left hand side and the common name (domain name) goes on the right hand side. Use any unique domain name you would like. Save the file when you are done (':wq')

```
wsuser1@cecs303-ws1:~$ cat /etc/hosts
127.0.0.1    localhost
127.0.1.1    cecs303-ws1
192.168.4.46 web.samayoa.local

# 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
wsuser1@cecs303-ws1:~$
```

- ii.

3. Ping your Apache web server using the new domain name you assigned it (note: this domain name will only work on the local machine where you modified the hosts file)

```
wsuser1@cecs303-ws1:~$ ping web.samayoa.local
PING web.samayoa.local (192.168.4.46) 56(84) bytes of data.
64 bytes from web.samayoa.local (192.168.4.46): icmp_seq=1 ttl=64 time=0.232 ms
64 bytes from web.samayoa.local (192.168.4.46): icmp_seq=2 ttl=64 time=0.355 ms
64 bytes from web.samayoa.local (192.168.4.46): icmp_seq=3 ttl=64 time=0.536 ms
64 bytes from web.samayoa.local (192.168.4.46): icmp_seq=4 ttl=64 time=0.362 ms
^C
--- web.samayoa.local ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3069ms
rtt min/avg/max/mdev = 0.232/0.371/0.536/0.108 ms
```

- a.
- b. Screenshot the successful ping using a domain name for your submission

You can now perform any of the above tests using the common name you created instead of the IPv4 address from this specific VM.

Deliverables (submit via BeachBoard)

1. Submit the three screenshots requested above to Dropbox (preferably in a single .doc, .docx, or .pdf file)
2. Why would the information learned from the 'curl -I <server ip address>' command be useful to a potential attacker? Provide a brief response, preferably in the same document as the requested screenshots.

Note: Command "shutdown now" will cleanly shut down virtual machines when you are done working with them

Additional systemctl commands to be familiar with for troubleshooting Apache:

- To stop your web server, type:
 - `sudo systemctl stop apache2`
- To start the web server when it is stopped, type:
 - `sudo systemctl start apache2`
- To stop and then start the service again, type:
 - `sudo systemctl restart apache2`
- If you are simply making configuration changes, Apache can often reload without dropping connections. To do this, use this command:
 - `sudo systemctl reload apache2`
- By default, Apache is configured to start automatically when the server boots. If this is not what you want, disable this behavior by typing:
 - `sudo systemctl disable apache2`
- To re-enable the service to start up at boot, type:
 - `sudo systemctl enable apache2`