Todd Wenker

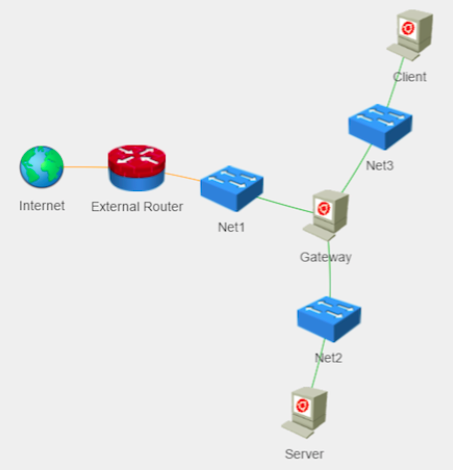
ASU ID: 1206233882

Lab Assignment 1

Computer Services Setup

Summary: The goal of the lab is based around setting up network services, like allowing the Client and Server machines access to the Internet, setting up an Apache2 server, and creating a DNS Server.

Network Setup:



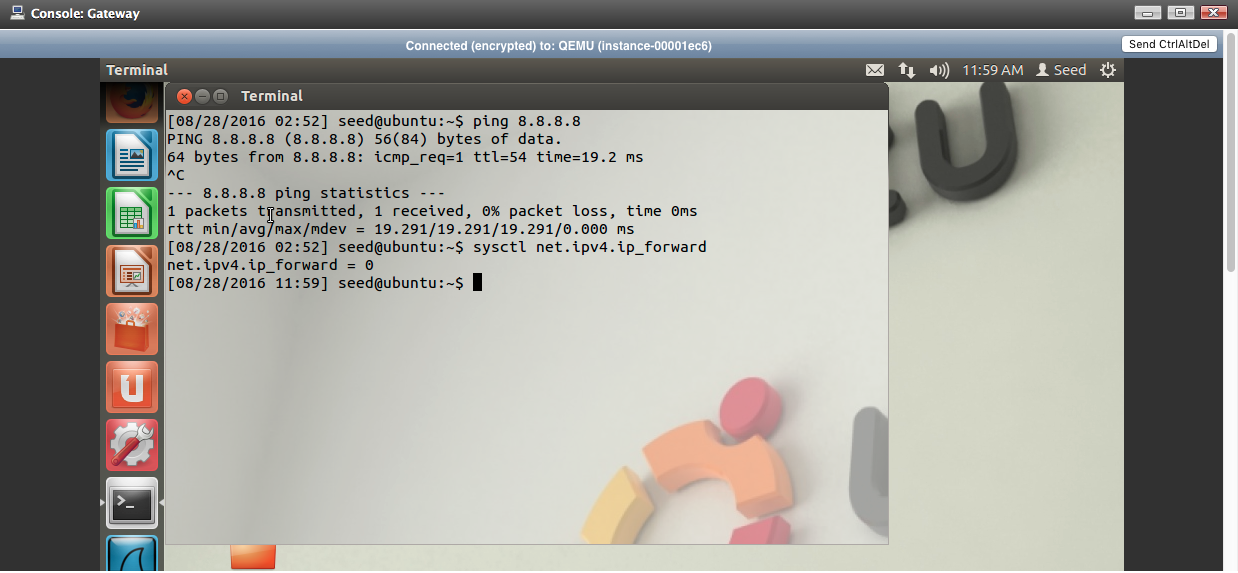
The three virtual machines used are Gateway, Client, and Server.

Software Packages Used:

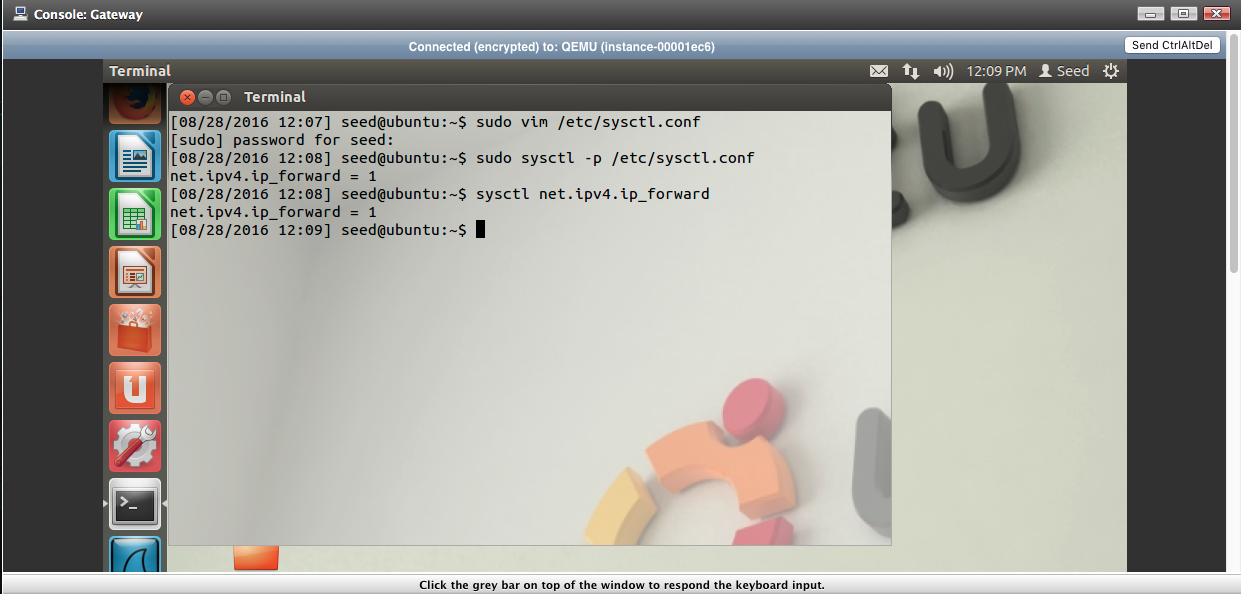
* Apache2
* Bind9

**Task 1 --- Enable IP Forwarding on Gateway:**

First, the command *sysctl net.ipv4.ip\_forward* is used to query the value of *net.ipv4.ip\_forward*. The results are below, showing that *net.ipv4.ip\_forward = 0* and thus, IP Forwarding is not activated.



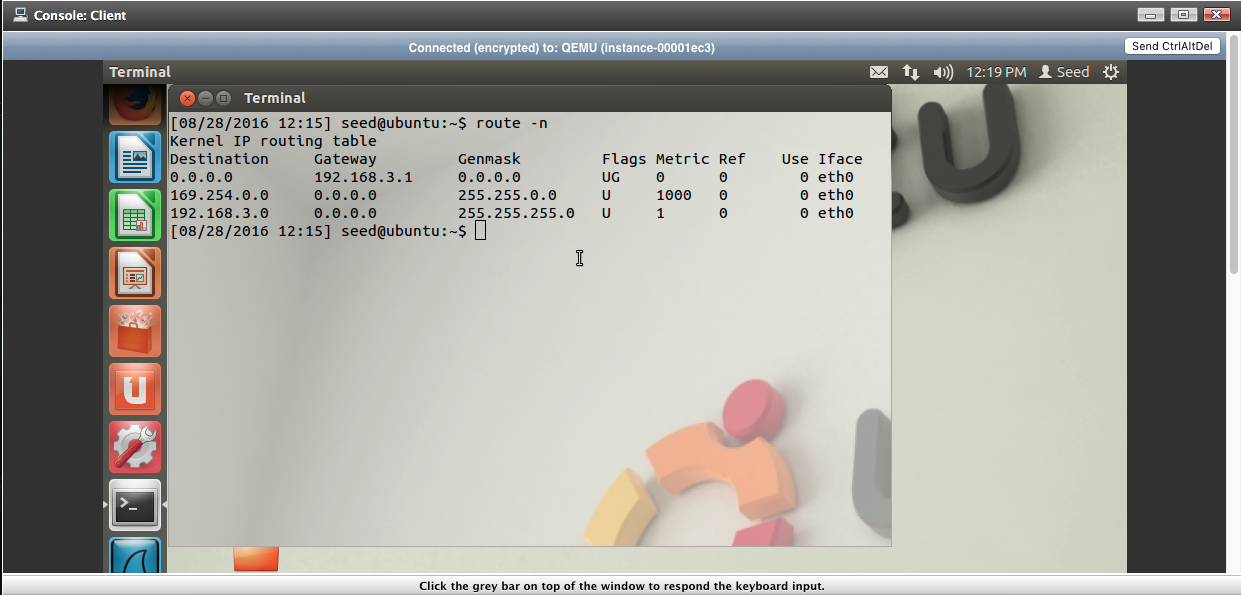
The next step involves changing the value in the */etc/sysctl.conf.* The command *sudo vim /etc/sysctl.conf*  is used to edit the *.conf* file using root privileges. The line number 27 has the code: *#net.ipv4.ip\_forwarding = 1*. By deleting the *#*, the line is no longer a comment and will change the value of this variable. The file is saved and then the command *sudo sysctl –p /etc/sysctl.conf* is used to enable the change. By using the command *sysctl net.ipv4.ip\_forward* again, we can see the change.



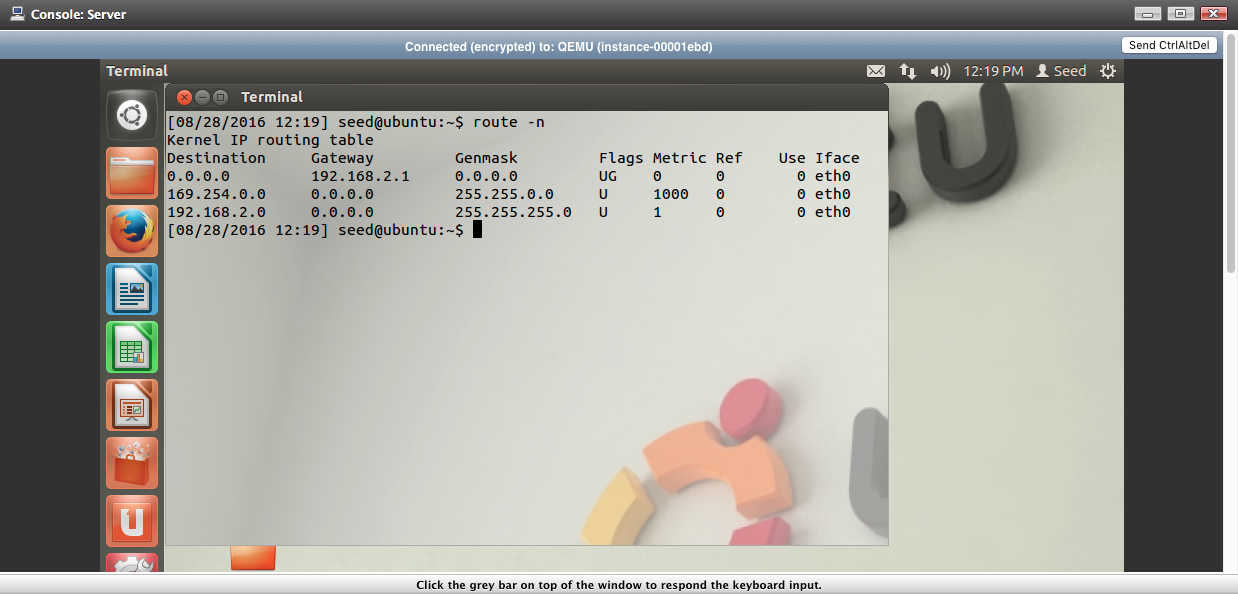
The line *net.ipv4.ip\_forward = 1* shows that IP Forwarding is now enabled on Gateway.

**Task 2 --- Edit Routing Tables on Client and Server:**

Using the command *route –n*, we can wee the routing table on both Client and Server:



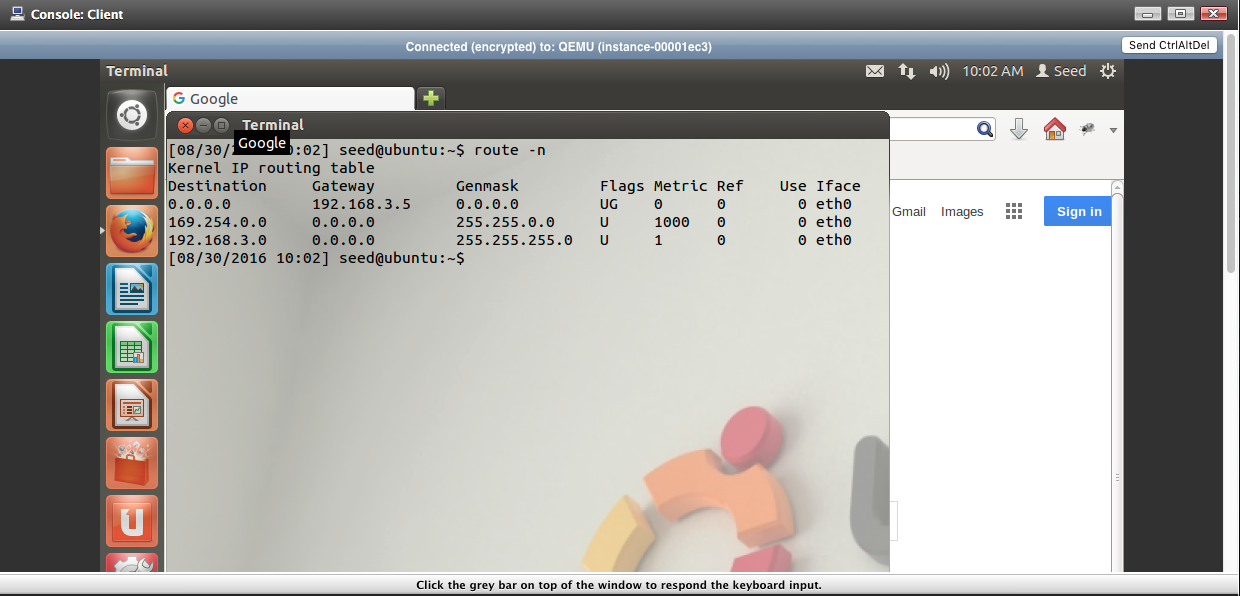
Client’s routing table. This shows that for a default destination, designated by the *0.0.0.0* entry under Destination, the gateway to be used is 192.168.3.1.



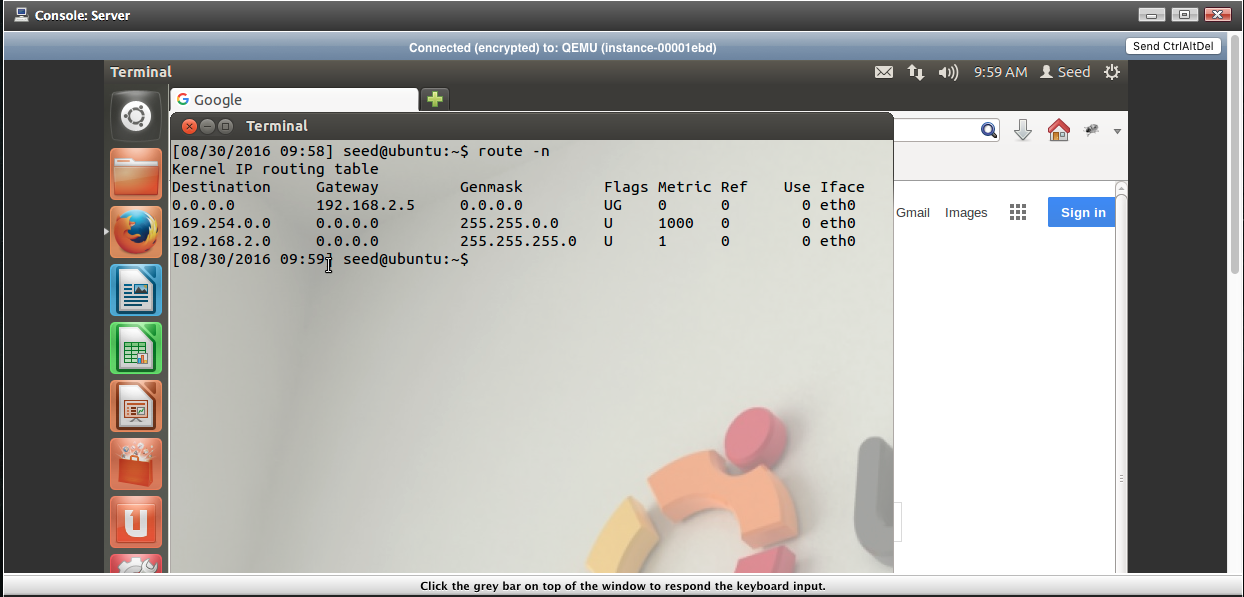
Server’s routing table. This shows that for a default destination, the gateway used will be 192.168.2.1.

To delete the default gateway, the command *sudo route del default.* This will delete the *0.0.0.0* Destination entry on both the Client and the Server machine.

Next, on the Client machine, the command *sudo route add default gw 192.168.3.5* is used to add the Gateway machine as the default gateway on Client. This is because, on the subnet that Gateway shares with Client, its IP address is 192.168.3.5.



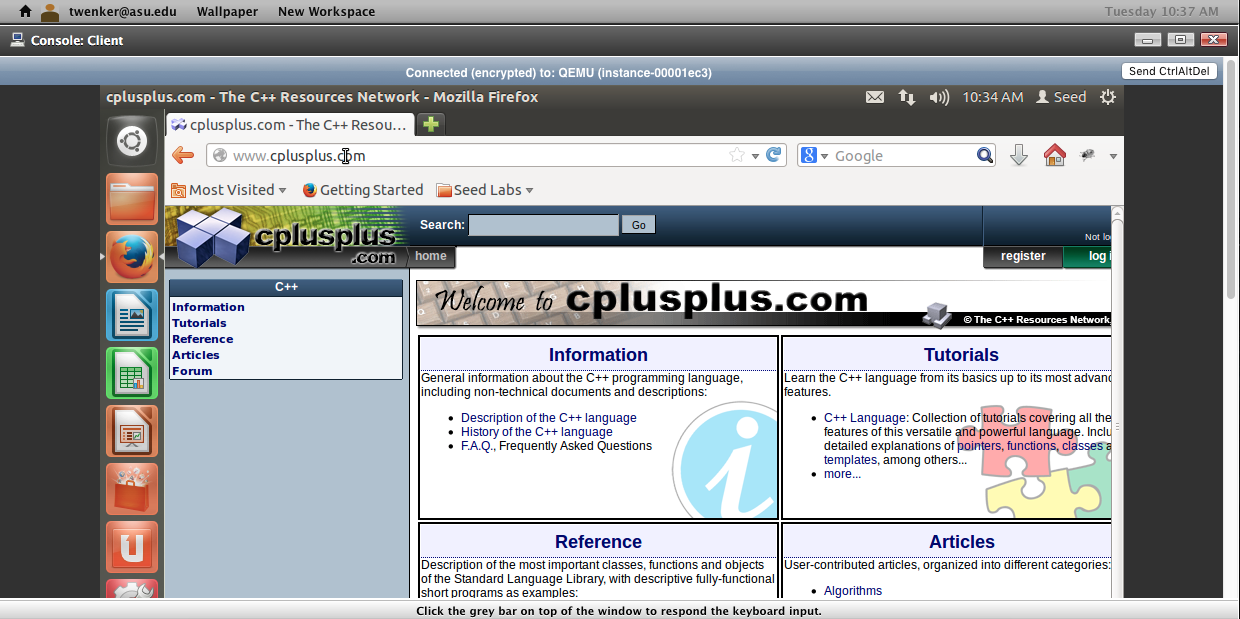
On the Server machine, the command *sudo route add default gw 192.168.2.5* is used to add the Gateway machine as the default gateway on Server. This is because *192.168.2.5* is the address of Gateway on that subnet.



**Task 3 --- Enable Network Address Translation on Gateway**

On the Gateway machine, the command *sudo iptables --table nat POSTROUTING --out-interface eth0 –j MASQUERADE* is used, followed by the commands *sudo iptables --append FORWARD --in-interface eth1 –j ACCEPT* and *sudo iptables --append FORWARD --in-interface eth2 –j ACCEPT.* This allows the Client and Server machines to access the Internet. This is done by network address translation (NAT), where Server and Client route their Internet access through Gateway. Outwardly, it looks like the requests originate from Gateway but once Gateway receives the packets, it routes them to either Client or Server, depending on which one made the request. The first command is what sets Gateway up to route requesting, with the *eth0* part denoting which way to route requests (this is the connection that links Gateway with the Internet). The second two commands add Client and Server, or more specifically *eth2* and *eth1*, the routes they are connected to respectively, to the forwarding table, allowing them to access the Internet through Gateway via NAT.

Both Client and Server are now capable of accessing the Internet by entering in IP address to the address bar in Firefox. By entering 167.114.170.15, Client and Server could access [www.cplusplus.com](http://www.cplusplus.com), shown below.

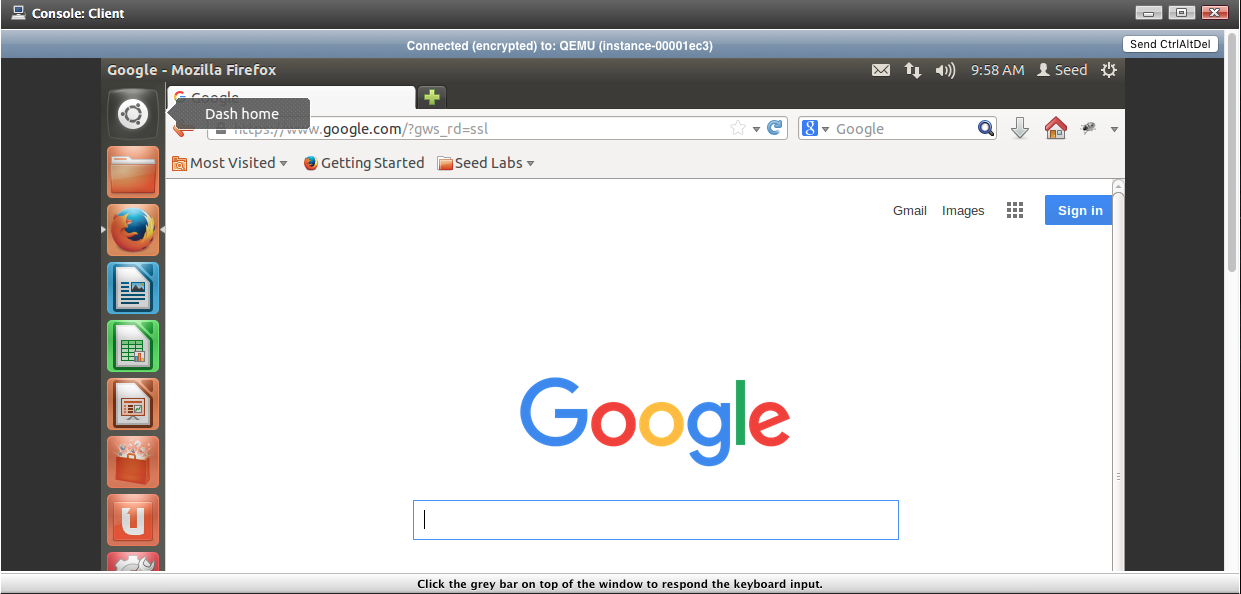


Client accessing IP address 167.114.170.15.



Server accessing IP address 167.114.170.15.

Both Client and Server can also access the Internet by entering in domain names to the address bar, as shown below.



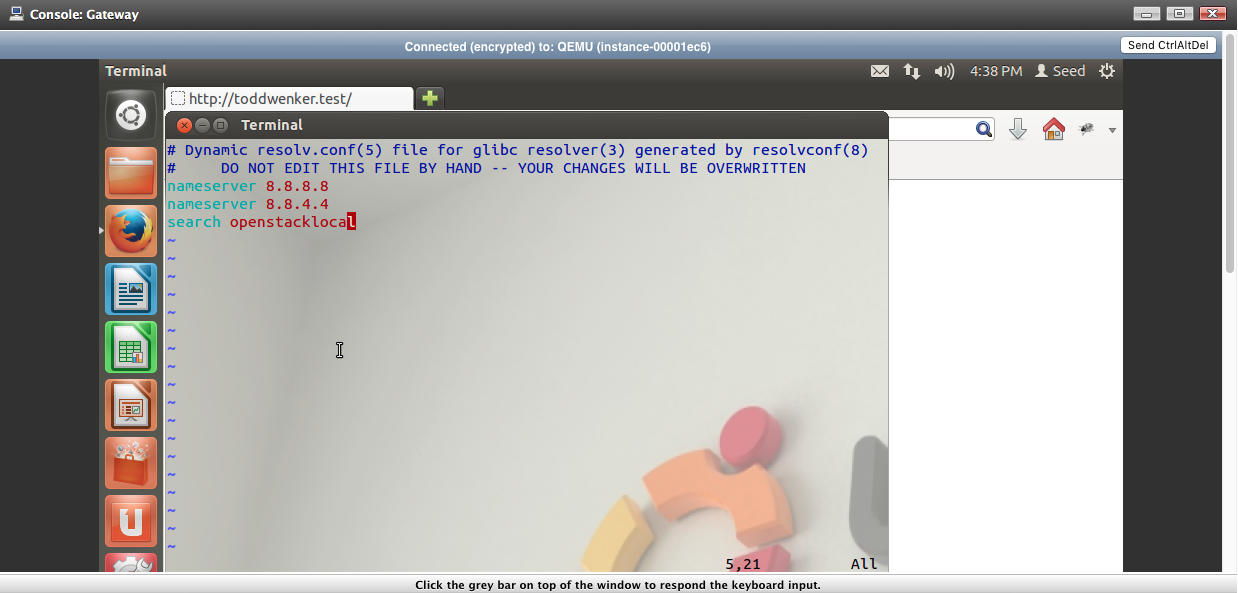
Client accessing google.com.

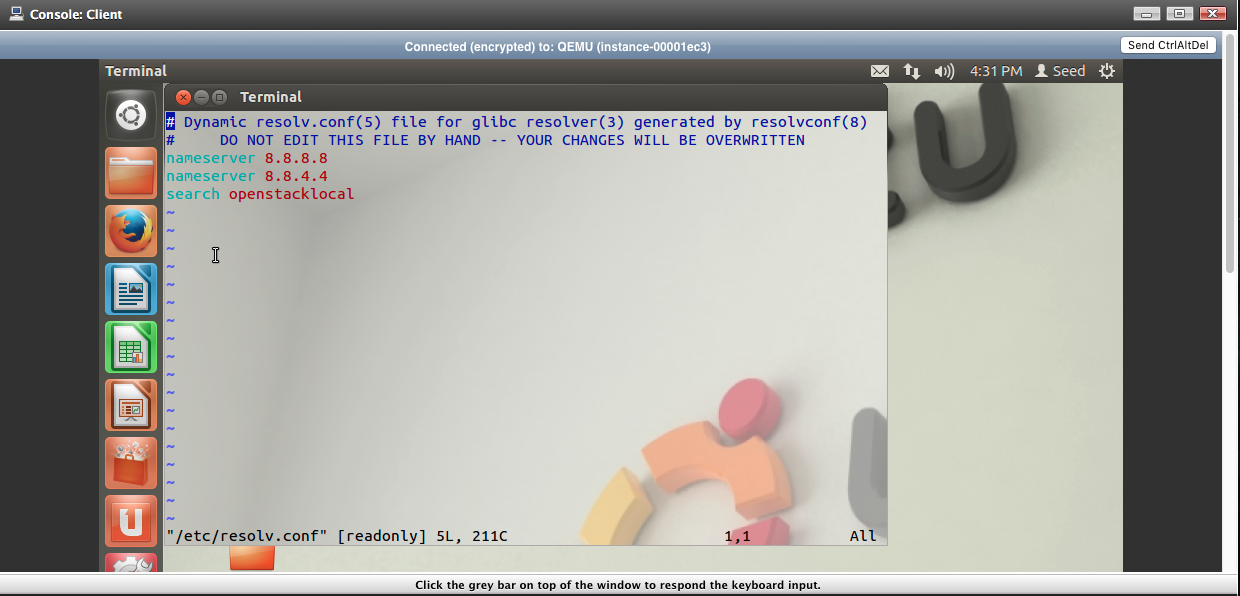


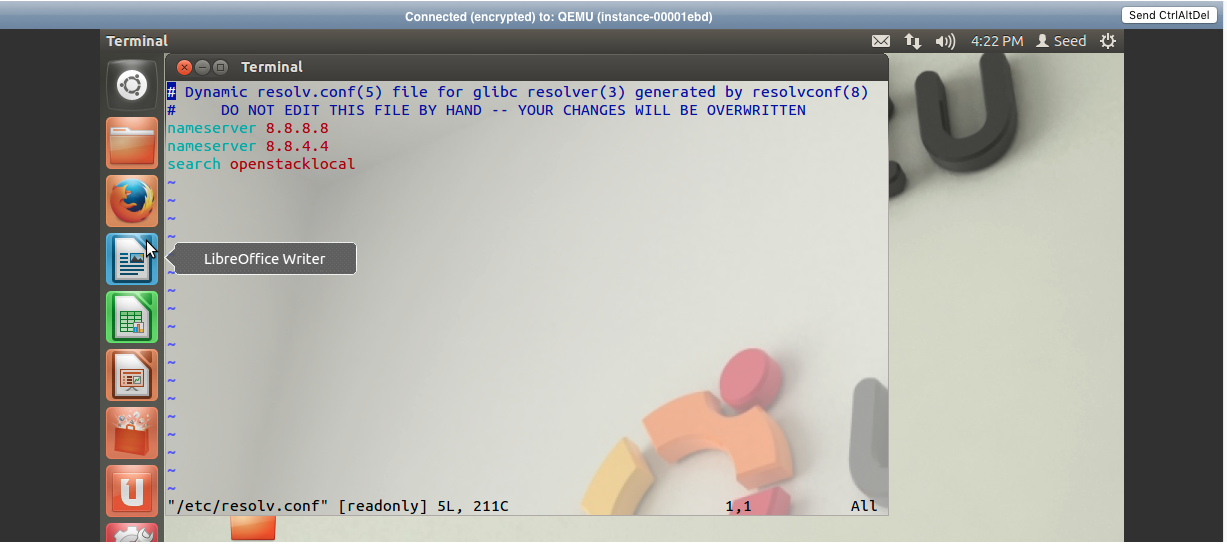
Server accessing google.com.

**Task 4 --- Add a Public DNS on Client, Server, and Gateway:**

Using the command *sudo vim /etc/resolv.conf* on all three machines, the file is edited to remove the line: *nameserver 127.0.1.1* and the lines: *nameserver 8.8.8.8 nameserver 8.8.4.4* are added to the file (with a newline separating them). This sets the Domain Name System (DNS) to Google’s public DNS instead of the DNS of the ISP. The change is shown below.







The changed /etc/resolv.conf file on Gateway, Client, and Server respectively.

**Task 5 --- Setup an Apache Server on Server:**

First, the command *sudo apt-get install update* is used to ensure that all software is up to date ( I noticed that without this set, Apache2 would fail to install successfully). Then the command *sudo apt-get install apache2* is used to actually download and install Apache2. To edit the default webpage, entitled index.html, the command *sudo vim /etc/www/index.html* is used to edit index.html and my ASU ID is added.

**Task 6 --- Setup a DNS Server on Gateway:**

First, the command *sudo apt-get install bind9* is used on Gateway to install Bind9 so a DNS server can be made.

The /etc/bind/named.conf.local needs to be appended with forward zones. The command *sudo vim /etc/bind/named.conf.local* is used and the following code is added:

*zone “toddwenker.test” {*

*type master;*

*file “/etc/bind/myzones/db.toddwenker.test”;*

*};*

*zone “2.168.192.in-addr.arpa” {*

*type master;*

*file “/etc/bind/myzones/db.192”;*

*};*

The next step is to actually create the zone files specified under the ‘file’ portions of the two zones. After using the command *mkdir /etc/bind/myzones*  to create our zone directory, the command *cp /etc/bind/db.local /etc/bind/myzones/db.toddwenker.test*  is used to create a copy of the default zone file db.local so we can use it as a template for our db.toddwenker.test. The command *vim /etc/bind/myzones/db.toddwenker.test*  is used so we can edit that file. After editing, the file looks as follows:

*$TTL 604800*

*@ IN SOA ubuntu-gateway.toddwenker.test. root.toddwenker.test.. (*

*3 ;serial*

*604800 ;Refresh*

*86400 ;Retry*

*2419200 ;expire*

*604800 ) ;Negative Cache TTL*

*;*

*IN NS ubuntu-server.toddwenker.test.*

*ubunut-server.toddwenker.test. IN A 192.168.2.3*

*www.toddwenekr.test. IN A 192.168.2.4*

After this zone is create, the reverse zone file needs to be created using the command *cp /etc/bind/db.127 /etc/bind/myzones/db.192* is used to create a copy of the default file db.127 so we can use it for a template. The file is edited to look as follows:

*$TTL 604800*

*@ IN SOA ubuntu-gateway.toddwenker.test. root.toddwenker.test.. (*

*3 ;serial*

*604800 ;Refresh*

*86400 ;Retry*

*2419200 ;expire*

*604800 ) ;Negative Cache TTL*

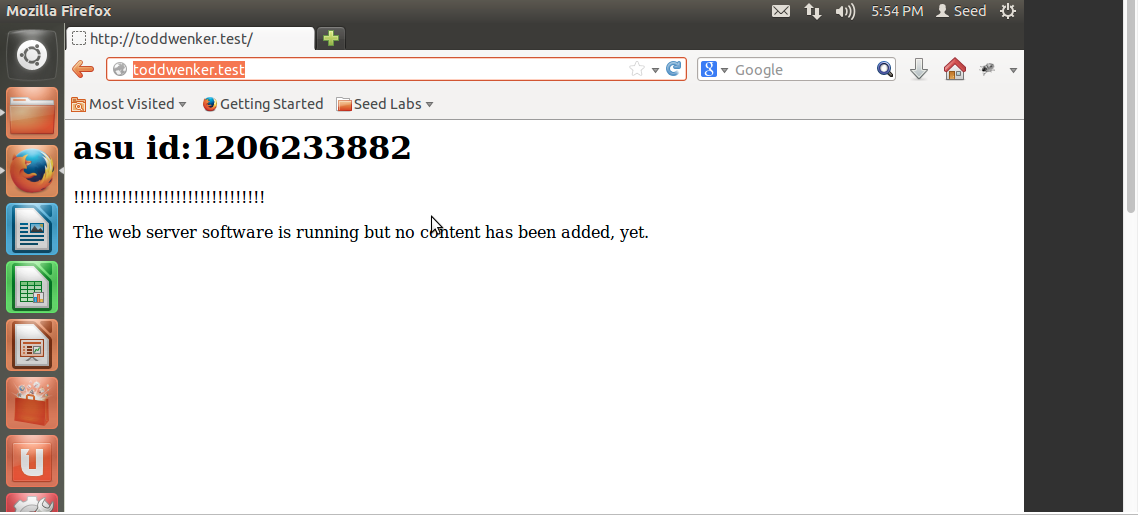
*IN NS ubuntu-gateway.toddwenker.test.*

*4 IN PTR ubuntu-gateway.toddwenker.test.*

*5 IN PTR www.toddwenker.test.*

Now, we can run the command *sudo service bind9 restart* to restart the DNS server so it adopts the changes made. Next, we reconfigure the /etc/resolv.conf file again and add the line *nameserver 192.168.2.4* by using the command *sudo vim /etc/resolv.conf*.

From the Client machine, we can now enter the toddwenker.test domain name and have the right html file open.



**Conclusion:**

This lab showed that a single machine with access to the Internet can be used as a gateway for other networks, allowing other machines Internet access. This lab also showed that a DNS server set up on this Gateway network can be used to display the Apache2 server of a different machine.