Routing Basics

1 Overview

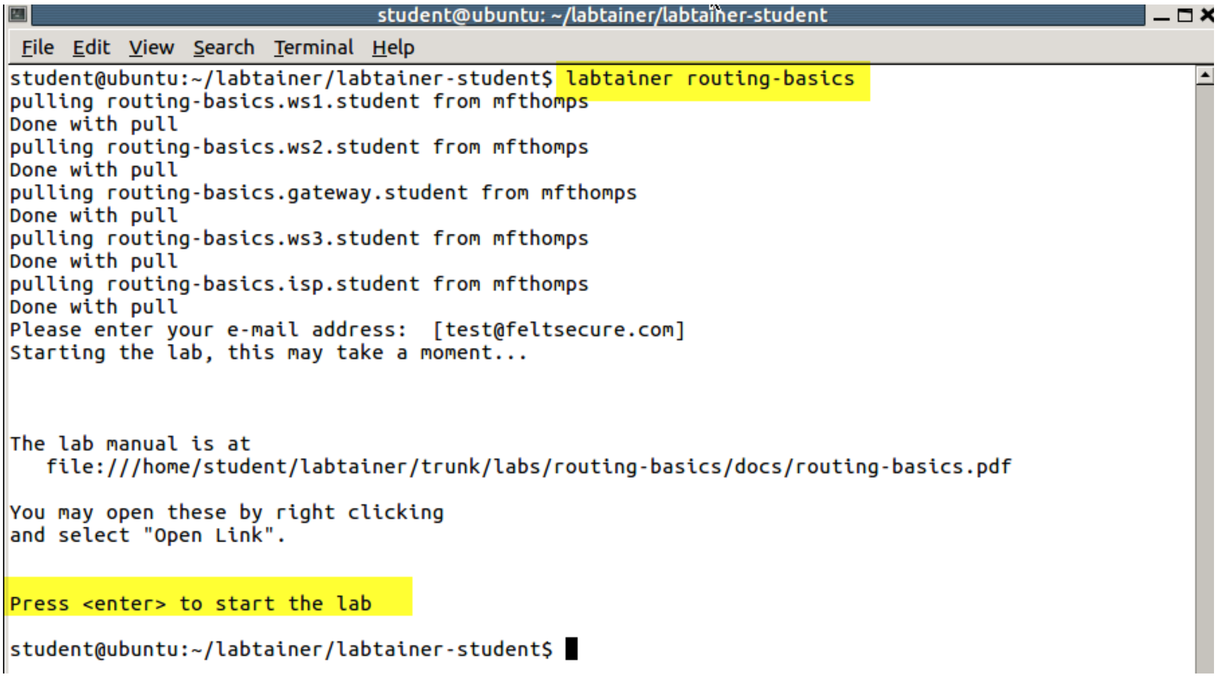
This exercise explores basic network routing concepts in a Linux environment. These include use of the route command, defining a DNS server in the /etc/resolv.conf file, and using Network Address Translation
(NAT).

This exercise, (and manual), is not intended to replace instruction or independent reading on the topic of network routing and routing in Linux systems. The exercise is intended to provide students with an environment with which they can experiment with the mechanics of routing network traffic.

2 Lab Environment

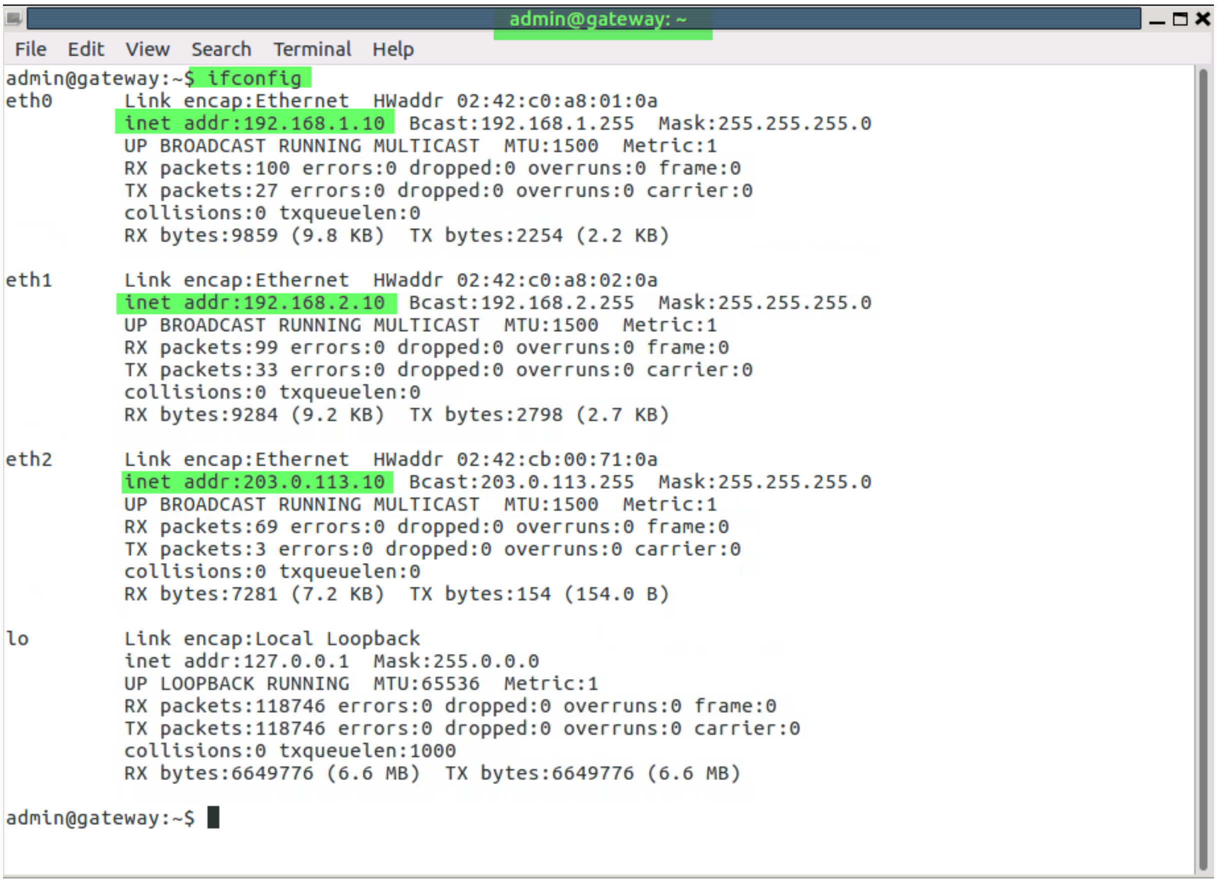
This lab runs in the Labtainer framework, available at http://my.nps.edu/web/c3o/labtainers. That site includes links to a pre-built virtual machine that has Labtainers installed, however Labtainers can be run on any Linux host that supports Docker containers.
From your labtainer-student directory start the lab using:

```
labtainer routing-basics
```


student@ubuntu: ~/labtainer/labtainer-student
File Edit View Search Terminal Help

```
student@ubuntu:~/labtainer/labtainer-student$ labtainer routing-basics
pulling routing-basics.ws1.student from mfthomps
Done with pull
pulling routing-basics.ws2.student from mfthomps
Done with pull
pulling routing-basics.gateway.student from mfthomps
Done with pull
pulling routing-basics.ws3.student from mfthomps
Done with pull
pulling routing-basics.isp.student from mfthomps
Done with pull
Please enter your e-mail address:  [test@feltsecure.com]
Starting the lab, this may take a moment...


The lab manual is at
   file:///home/student/labtainer/trunk/labs/routing-basics/docs/routing-basics.pdf

You may open these by right clicking
and select "Open Link".

Press <enter> to start the lab

student@ubuntu:~/labtainer/labtainer-student$
```

A link to this lab manual will be displayed.



```
admin@gateway:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:c0:a8:01:0a
          inet addr:192.168.1.10  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:100 errors:0 dropped:0 overruns:0 frame:0
          TX packets:27 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:9859 (9.8 KB)  TX bytes:2254 (2.2 KB)

eth1      Link encap:Ethernet  HWaddr 02:42:c0:a8:02:0a
          inet addr:192.168.2.10  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:99 errors:0 dropped:0 overruns:0 frame:0
          TX packets:33 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:9284 (9.2 KB)  TX bytes:2798 (2.7 KB)

eth2      Link encap:Ethernet  HWaddr 02:42:cb:00:71:0a
          inet addr:203.0.113.10  Bcast:203.0.113.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:69 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:7281 (7.2 KB)  TX bytes:154 (154.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:118746 errors:0 dropped:0 overruns:0 frame:0
          TX packets:118746 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6649776 (6.6 MB)  TX bytes:6649776 (6.6 MB)

admin@gateway:~$ 
```

3 Network Configuration

This lab includes four networked computers as shown in Figure 1. When the lab starts, you will get four virtual terminals, one connected to each component. The gateway is configured to perform routing between LAN1 and LAN2, and to route external addresses to an external gateway, e.g., to reach the Internet. The ws1 and ws2 workstations are pre-configured to route traffic to the gateway component. The ws3 workstation is not yet configured for routing.

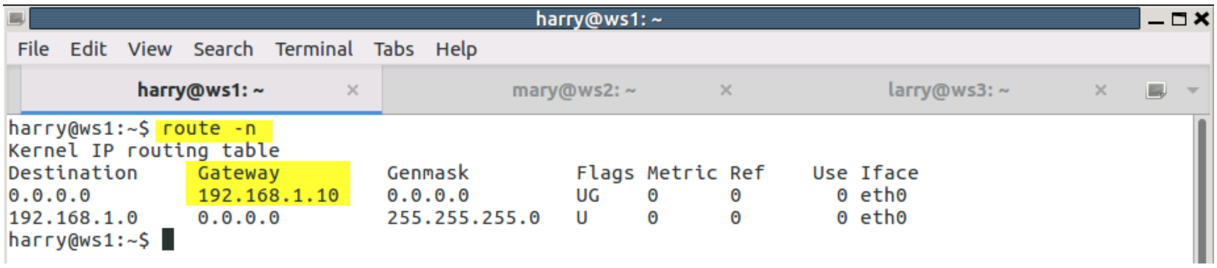
The gateway is configured to use NAT to translate sources addresses of traffic from internal IP addresses, e.g., 192.168.1.1, to our external address, i.e., 203.0.113.10.

4 Lab Tasks

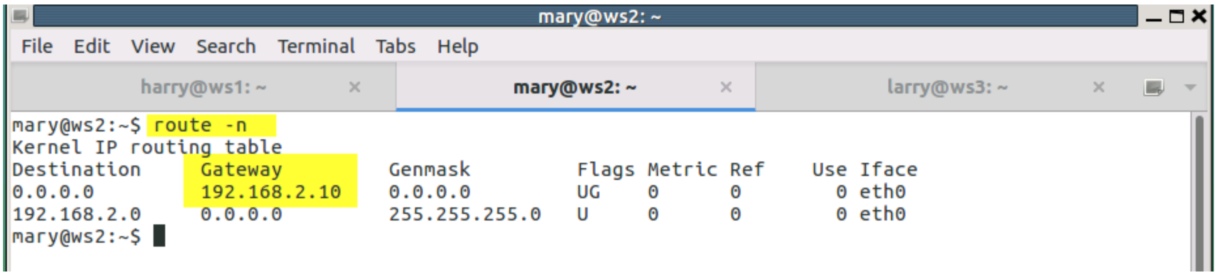
4.1 Internal Routing

From each of the three workstations, enter the following command:

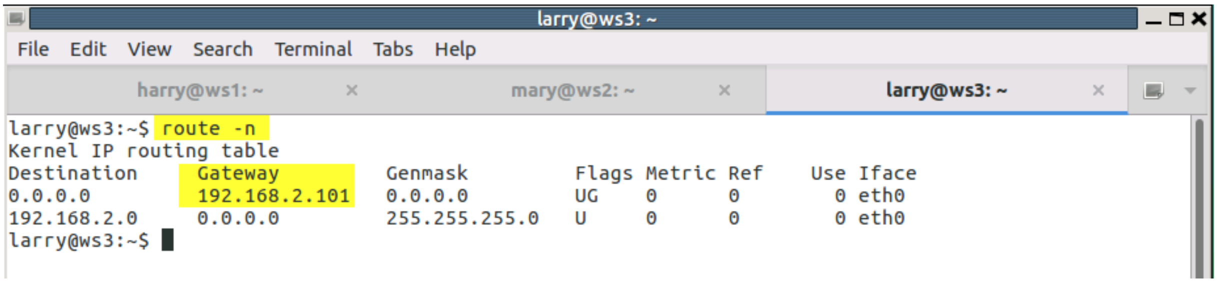
```
route −n
```



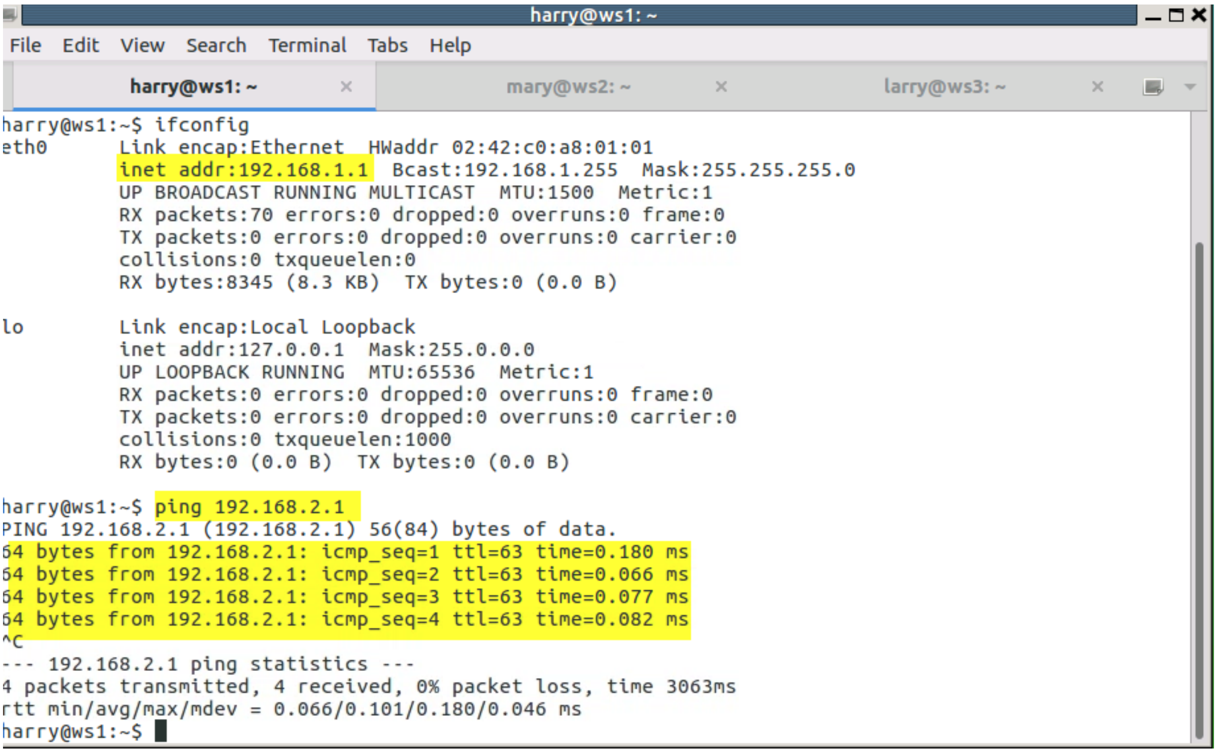
```
harry@ws1:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.10    0.0.0.0         UG    0      0        0 eth0
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
harry@ws1:~$
```



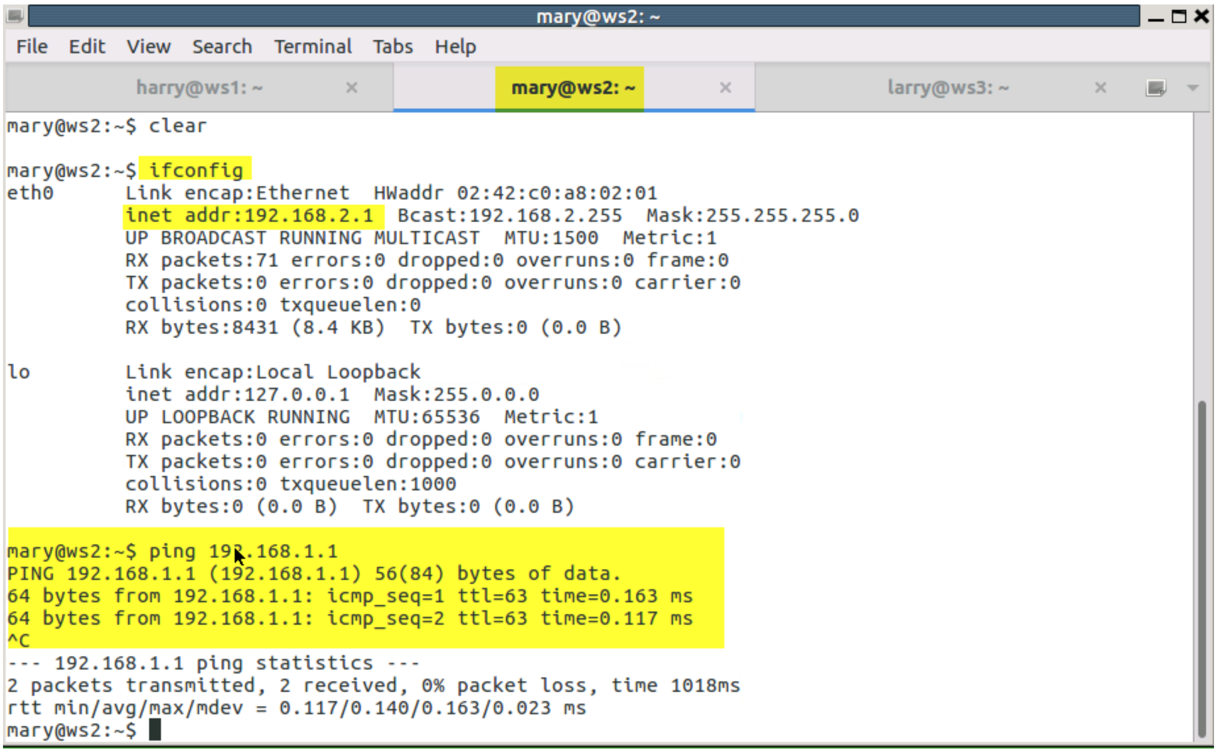
```
mary@ws2:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.2.10    0.0.0.0         UG    0      0        0 eth0
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
mary@ws2:~$
```



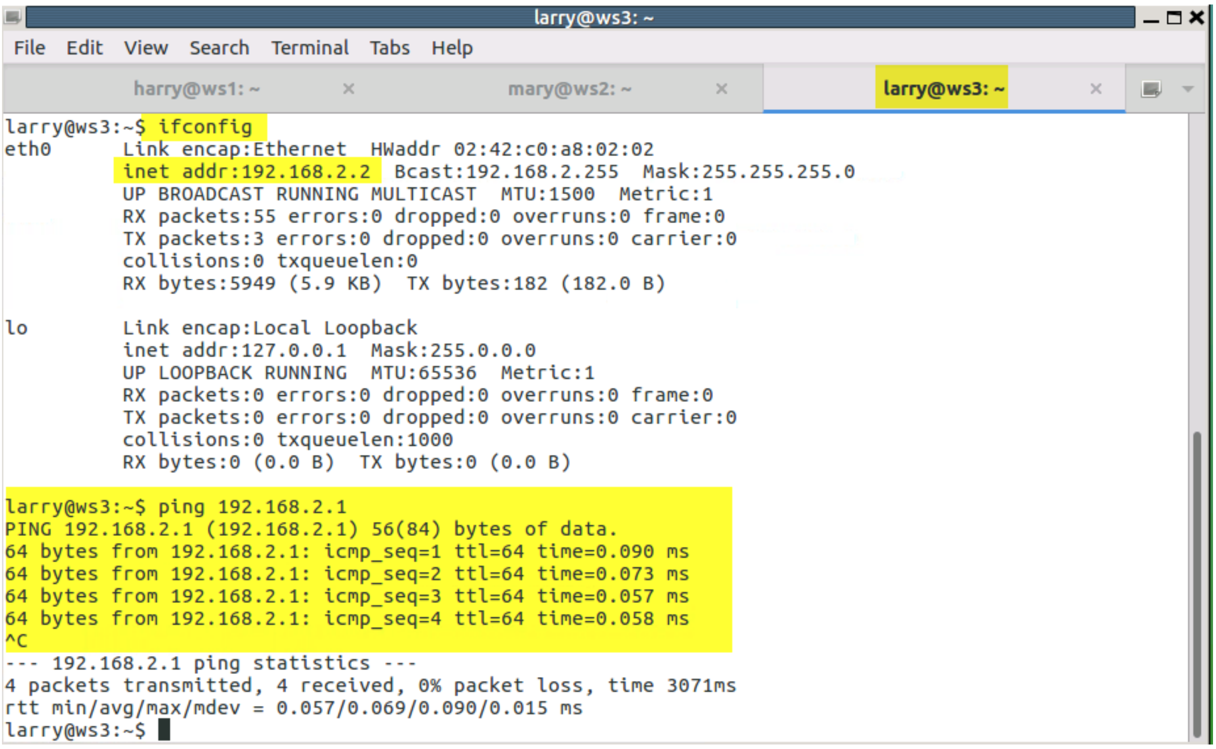
```
larry@ws3:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.2.101   0.0.0.0         UG    0      0        0 eth0
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
larry@ws3:~$
```



```
harry@ws1:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:c0:a8:01:01
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:70 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:8345 (8.3 KB)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

harry@ws1:~$ ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=63 time=0.180 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=63 time=0.066 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=63 time=0.077 ms
64 bytes from 192.168.2.1: icmp_seq=4 ttl=63 time=0.082 ms
^C
--- 192.168.2.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3063ms
rtt min/avg/max/mdev = 0.066/0.101/0.180/0.046 ms
harry@ws1:~$
```



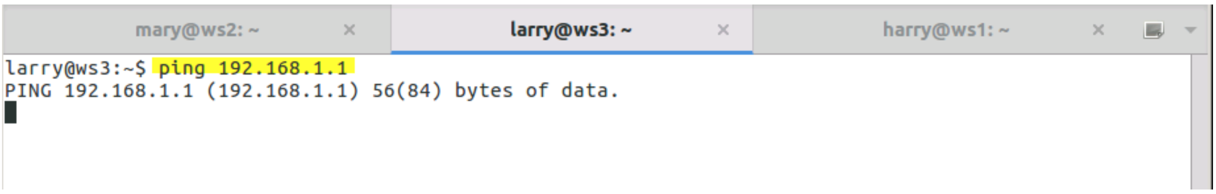
```
mary@ws2:~$ clear

mary@ws2:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:c0:a8:02:01
          inet addr:192.168.2.1  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:71 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:8431 (8.4 KB)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

mary@ws2:~$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=63 time=0.163 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=63 time=0.117 ms
^C
--- 192.168.1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1018ms
rtt min/avg/max/mdev = 0.117/0.140/0.163/0.023 ms
mary@ws2:~$
```



```
larry@ws3:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:c0:a8:02:02
          inet addr:192.168.2.2  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:55 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:5949 (5.9 KB)  TX bytes:182 (182.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

larry@ws3:~$ ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=64 time=0.090 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=64 time=0.073 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=64 time=0.057 ms
64 bytes from 192.168.2.1: icmp_seq=4 ttl=64 time=0.058 ms
^C
--- 192.168.2.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3071ms
rtt min/avg/max/mdev = 0.057/0.069/0.090/0.015 ms
larry@ws3:~$
```

Note how ws1 and ws2 include routing table entries that name the gateway as the default gateway. This allows ws1 and ws2 to address each other, which can be demonstrated by using ping from ws1 to reach ws2:

```
ping [ws2 IP]
```

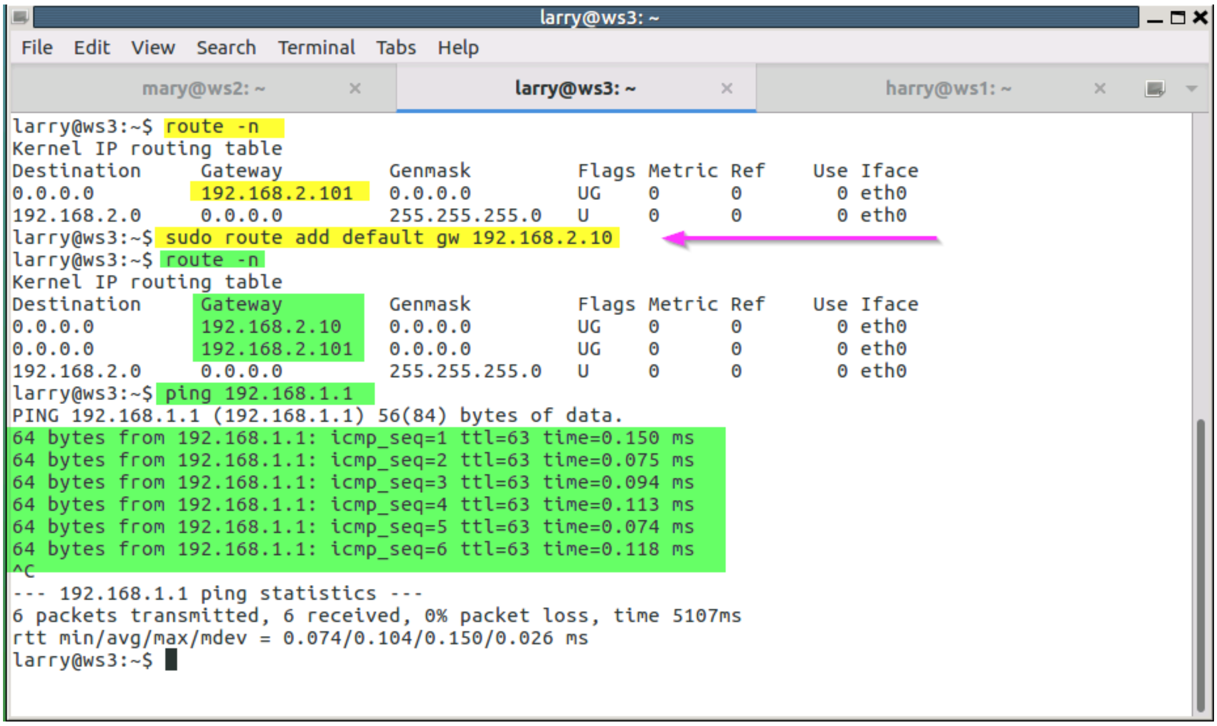


```
larry@ws3:~$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
```

Now consider ws2 and ws3. Since they are both on the same LAN, they can ping each other. Try that for yourself. Then try to ping ws1 from ws3. That will fail because ws3 has no routing table entry defining what to do with traffic that is not destined for a LAN directly connected to ws3.

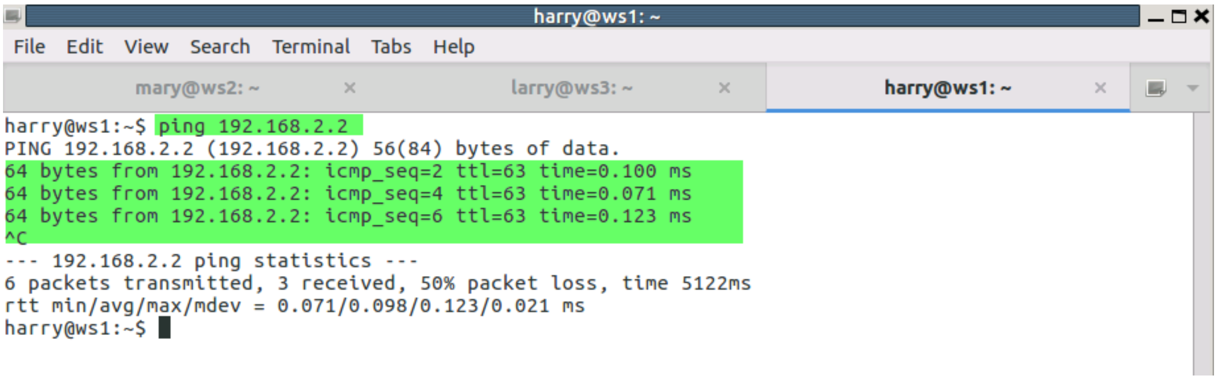
On ws3, define the gateway component as the default gateway using the route command, but this time using sudo because we are altering the routing:

```
sudo route add default gw [gateway IP]
```



```
larry@ws3:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.2.101   0.0.0.0         UG    0      0        0 eth0
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
larry@ws3:~$ sudo route add default gw 192.168.2.10
larry@ws3:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.2.10    0.0.0.0         UG    0      0        0 eth0
0.0.0.0         192.168.2.101   0.0.0.0         UG    0      0        0 eth0
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
larry@ws3:~$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=63 time=0.150 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=63 time=0.075 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=63 time=0.094 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=63 time=0.113 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=63 time=0.074 ms
64 bytes from 192.168.1.1: icmp_seq=6 ttl=63 time=0.118 ms
^C
--- 192.168.1.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5107ms
rtt min/avg/max/mdev = 0.074/0.104/0.150/0.026 ms
larry@ws3:~$
```

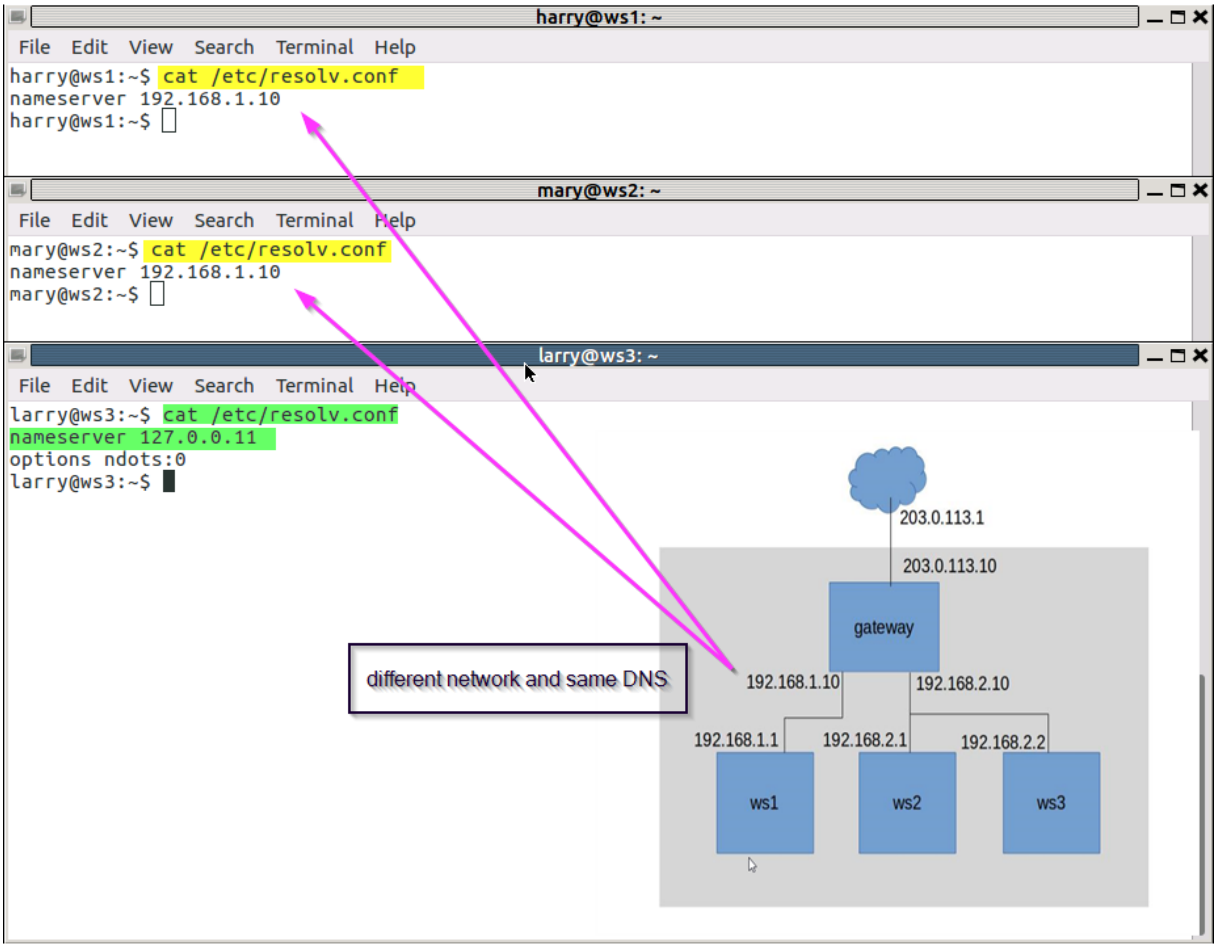
Then try to ping between ws1 and ws3.



```
harry@ws1:~$ ping 192.168.2.2
PING 192.168.2.2 (192.168.2.2) 56(84) bytes of data.
64 bytes from 192.168.2.2: icmp_seq=2 ttl=63 time=0.100 ms
64 bytes from 192.168.2.2: icmp_seq=4 ttl=63 time=0.071 ms
64 bytes from 192.168.2.2: icmp_seq=6 ttl=63 time=0.123 ms
^C
--- 192.168.2.2 ping statistics ---
6 packets transmitted, 3 received, 50% packet loss, time 5122ms
rtt min/avg/max/mdev = 0.071/0.098/0.123/0.021 ms
harry@ws1:~$
```

4.2 Routing to the Internet

The gateway component is configured to route to a simulated ISP at 203.0.113.1, which is a hidden component that provides routing to the Internet for this lab. From ws2, try to ping www.google.com. Then do the same from ws3. The problem with ws3 is that it has no domain name service (DNS) definition. Note, routing from ws3 to the Internet works fine, which you can confirm by pinging the IP address of www.google.com (as displayed when you pinged from ws2). The ws3 component simply lacks a DNS definition. On ws2, the DNS is defined to be the gateway component, and this is achieved in the /etc/resolv.conf file [1]. If you modify that file on ws3 to match that of ws2, that will tell ws3 to use the gateway as its DNS.

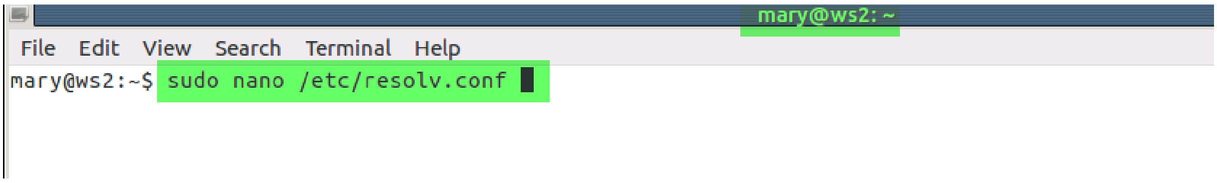


```
harry@ws1:~$ cat /etc/resolv.conf
nameserver 192.168.1.10
harry@ws1:~$
```

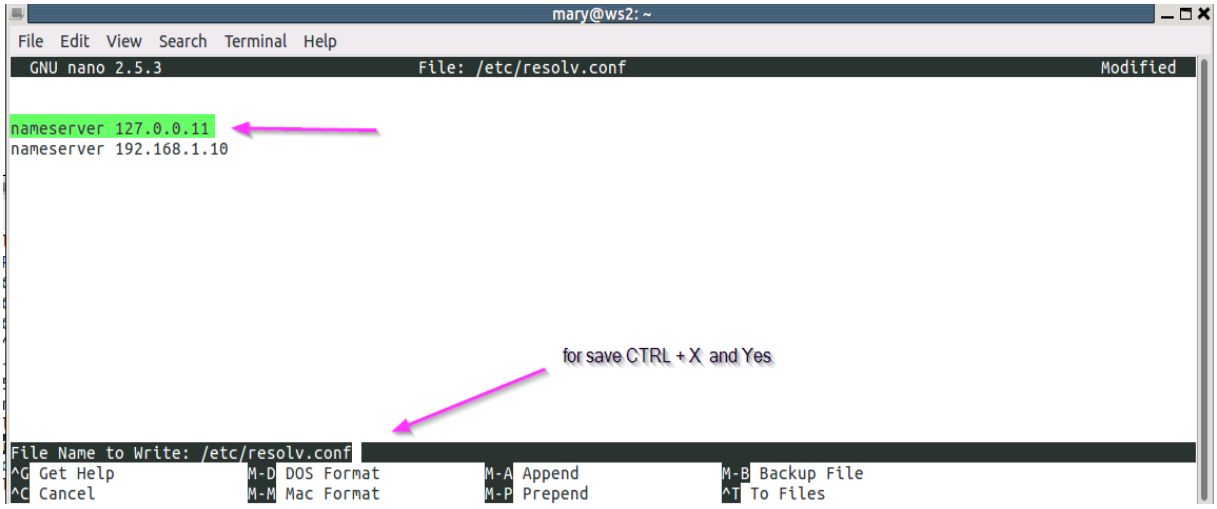
```
mary@ws2:~$ cat /etc/resolv.conf
nameserver 192.168.1.10
mary@ws2:~$
```

```
larry@ws3:~$ cat /etc/resolv.conf
nameserver 127.0.0.11
options ndots:0
larry@ws3:~$
```

different network and same DNS



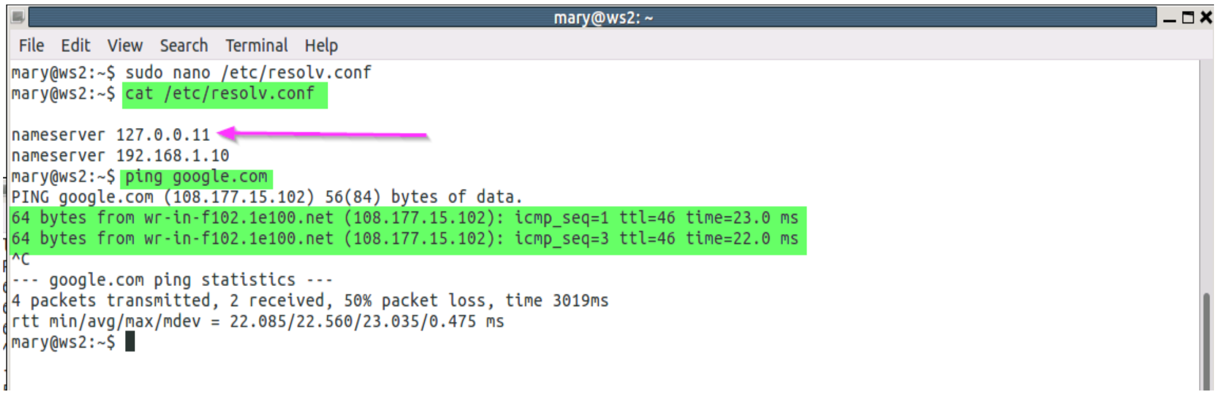
```
mary@ws2:~$ sudo nano /etc/resolv.conf
```



```
  GNU nano 2.5.3                    File: /etc/resolv.conf                        Modified


nameserver 127.0.0.11
nameserver 192.168.1.10
```

for save CTRL + X and Yes

```
File Name to Write: /etc/resolv.conf
^G Get Help          M-D DOS Format       M-A Append           M-B Backup File
^C Cancel            M-M Mac Format       M-P Prepend          ^T To Files
```

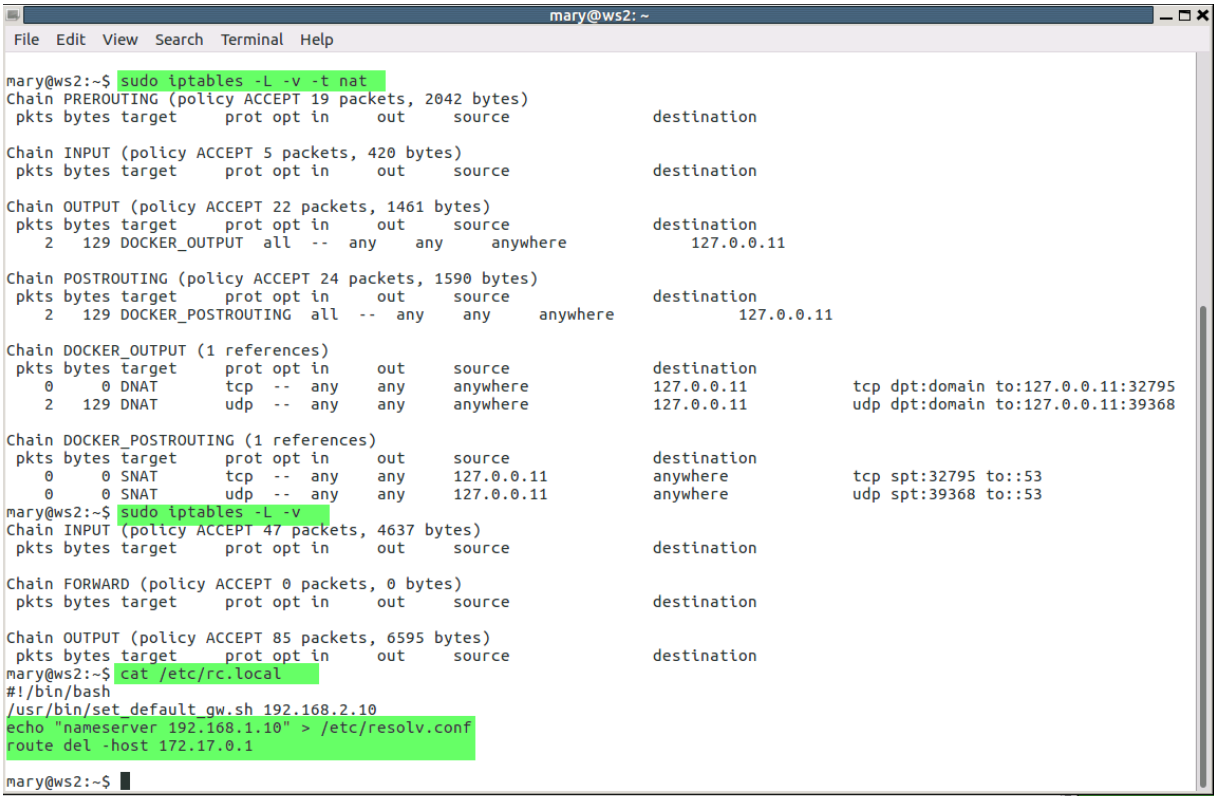


```
mary@ws2:~$ sudo nano /etc/resolv.conf
mary@ws2:~$ cat /etc/resolv.conf

nameserver 127.0.0.11
nameserver 192.168.1.10
mary@ws2:~$ ping google.com
PING google.com (108.177.15.102) 56(84) bytes of data.
64 bytes from wr-in-f102.1e100.net (108.177.15.102): icmp_seq=1 ttl=46 time=23.0 ms
64 bytes from wr-in-f102.1e100.net (108.177.15.102): icmp_seq=3 ttl=46 time=22.0 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 2 received, 50% packet loss, time 3019ms
rtt min/avg/max/mdev = 22.085/22.560/23.035/0.475 ms
mary@ws2:~$ 
```

4.3 Use of Network Address Translation (NAT)

Finally, review how the gateway component implements NAT using the iptables utility. Consider traffic from ws1 destined for www.google.com. The source IP address on those packets is 192.168.1.1. The ws1 component sends the packets to its default gateway, i.e., our gateway component. The gateway routing table is configured to send external traffic to 203.0.113.1. However, before that traffic is sent, we need to translate the source IP address to our exernal 203.0.113.10 address so that google knows where to send replies. Use this command:

```
sudo iptables -L -v -t nat
```



```
mary@ws2:~$ sudo iptables -L -v -t nat
Chain PREROUTING (policy ACCEPT 19 packets, 2042 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain INPUT (policy ACCEPT 5 packets, 420 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 22 packets, 1461 bytes)
 pkts bytes target     prot opt in     out     source               destination
    2   129 DOCKER_OUTPUT  all  --  any    any     anywhere             127.0.0.11

Chain POSTROUTING (policy ACCEPT 24 packets, 1590 bytes)
 pkts bytes target     prot opt in     out     source               destination
    2   129 DOCKER_POSTROUTING  all  --  any    any     anywhere             127.0.0.11

Chain DOCKER_OUTPUT (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DNAT       tcp  --  any    any     anywhere             127.0.0.11           tcp dpt:domain to:127.0.0.11:32795
    2   129 DNAT       udp  --  any    any     anywhere             127.0.0.11           udp dpt:domain to:127.0.0.11:39368

Chain DOCKER_POSTROUTING (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 SNAT       tcp  --  any    any     127.0.0.11           anywhere             tcp spt:32795 to::53
    0     0 SNAT       udp  --  any    any     127.0.0.11           anywhere             udp spt:39368 to::53
mary@ws2:~$ sudo iptables -L -v
Chain INPUT (policy ACCEPT 47 packets, 4637 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 85 packets, 6595 bytes)
 pkts bytes target     prot opt in     out     source               destination
mary@ws2:~$ cat /etc/rc.local
#!/bin/bash
/usr/bin/set_default_gw.sh 192.168.2.10
echo "nameserver 192.168.1.10" > /etc/resolv.conf
route del -host 172.17.0.1

mary@ws2:~$ █
```

to view our NAT rule, having a target of MASQUERADE, which will translate source addresses for all traffic destined for our external network interface. Then use this command:

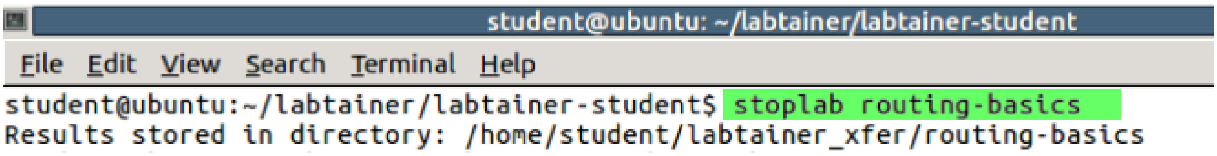
```
sudo iptables -L -v
```

to see that we are forwarding traffic received from the two LANs.
Our iptables NAT rules are defined in the /etc/rc.local file on the gateway component.

5 Submission

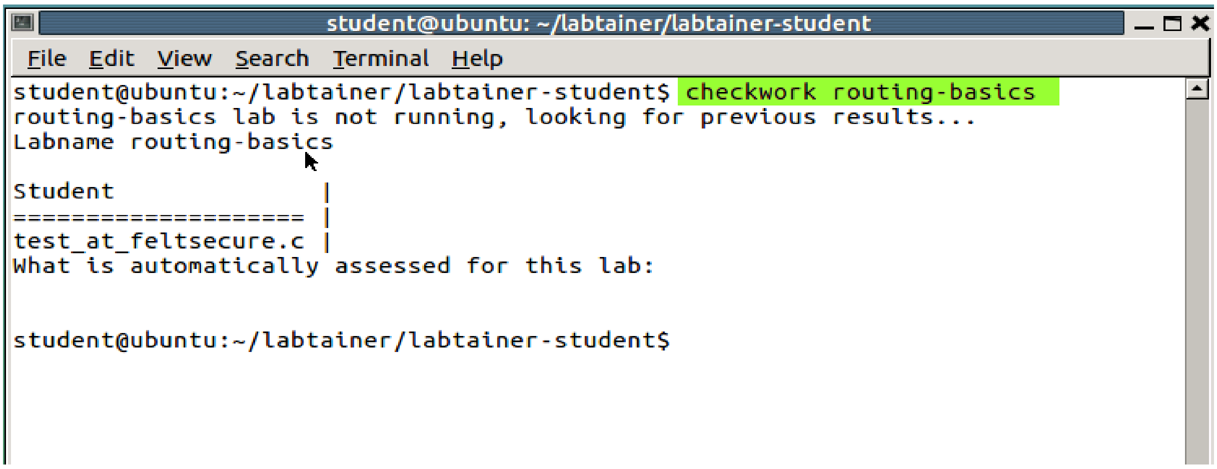
After finishing the lab, go to the terminal on your Linux system that was used to start the lab and type:

```
stoplab routing-basics
```



student@ubuntu: ~/labtainer/labtainer-student
File Edit View Search Terminal Help

```
student@ubuntu:~/labtainer/labtainer-student$ stoplab routing-basics
Results stored in directory: /home/student/labtainer_xfer/routing-basics
```



student@ubuntu: ~/labtainer/labtainer-student
File Edit View Search Terminal Help

```
student@ubuntu:~/labtainer/labtainer-student$ checkwork routing-basics
routing-basics lab is not running, looking for previous results...
Labname routing-basics

Student              |
==================== |
test_at_feltsecure.c |
What is automatically assessed for this lab:


student@ubuntu:~/labtainer/labtainer-student$
```

When you stop the lab, the system will display a path to the zipped lab results on your Linux system. Provide that file to your instructor, e.g., via the Sakai site.

[1] Many Linux systems include tools for defining your DNS, and these tools will overwrite the resolv.conf file. That is not anissue in these labs

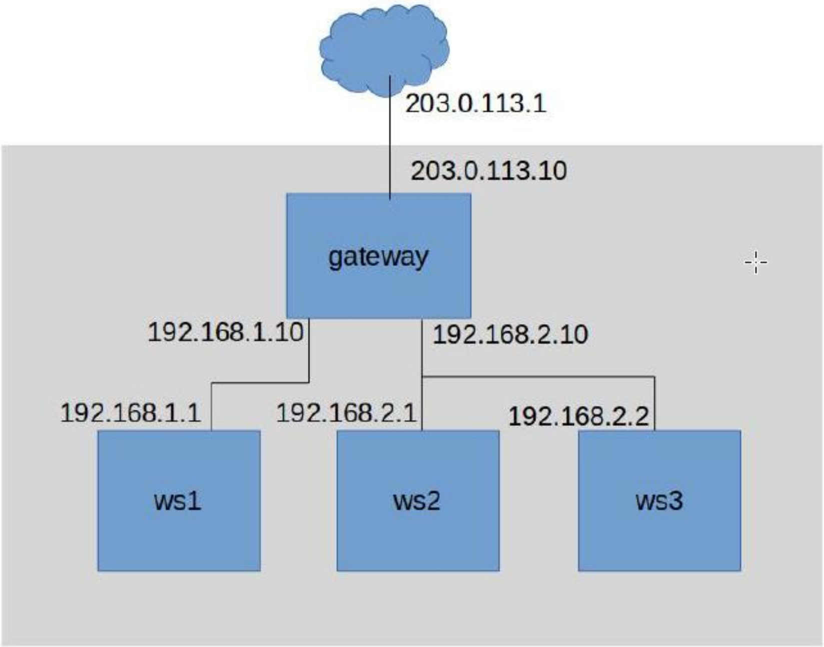


Figure 1: Network topology for routing-basics lab