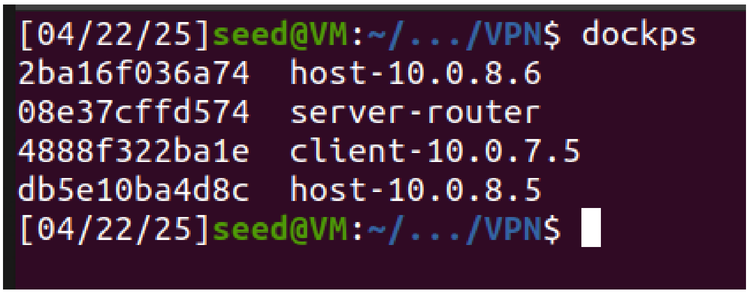
VPN Tunneling Lab

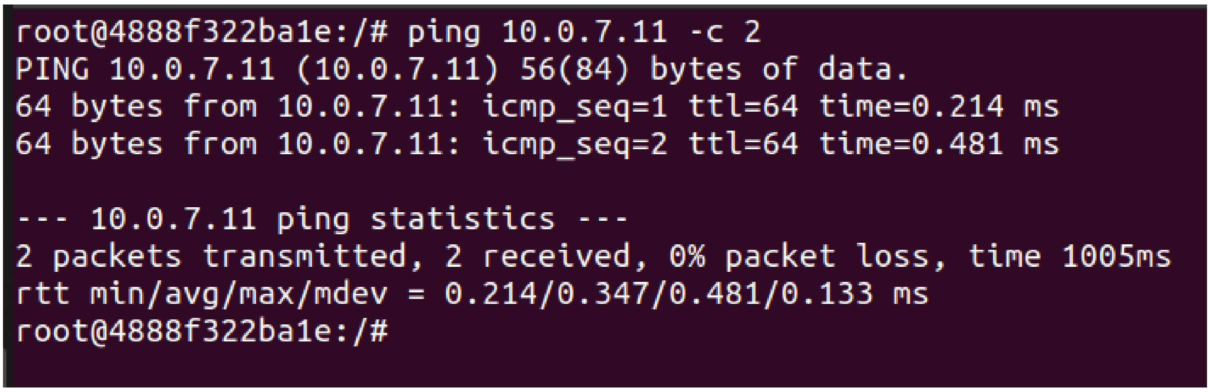
Task 1: Network Setup

Container ที่ใช้ docker-composes.yml



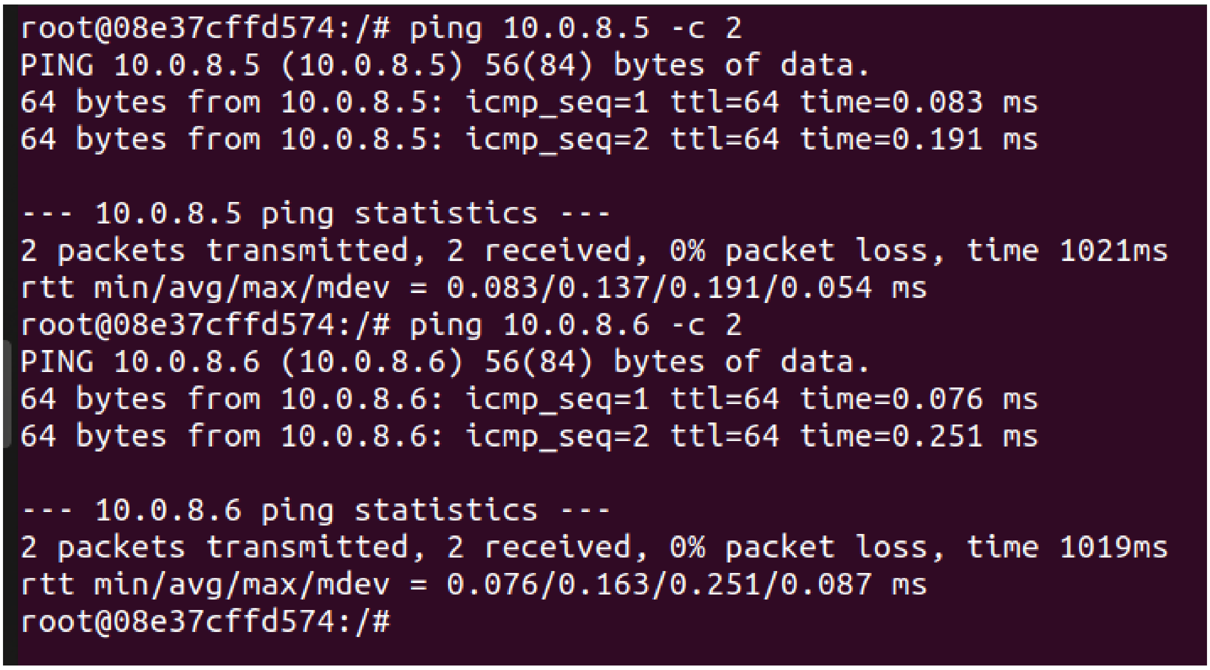
```
[04/22/25]seed@VM:~/.../VPN$ dockps
2ba16f036a74  host-10.0.8.6
08e37cffd574  server-router
4888f322ba1e  client-10.0.7.5
db5e10ba4d8c  host-10.0.8.5
[04/22/25]seed@VM:~/.../VPN$ 
```

ทดลองว่า client จะสามารถ ping หา router ได้



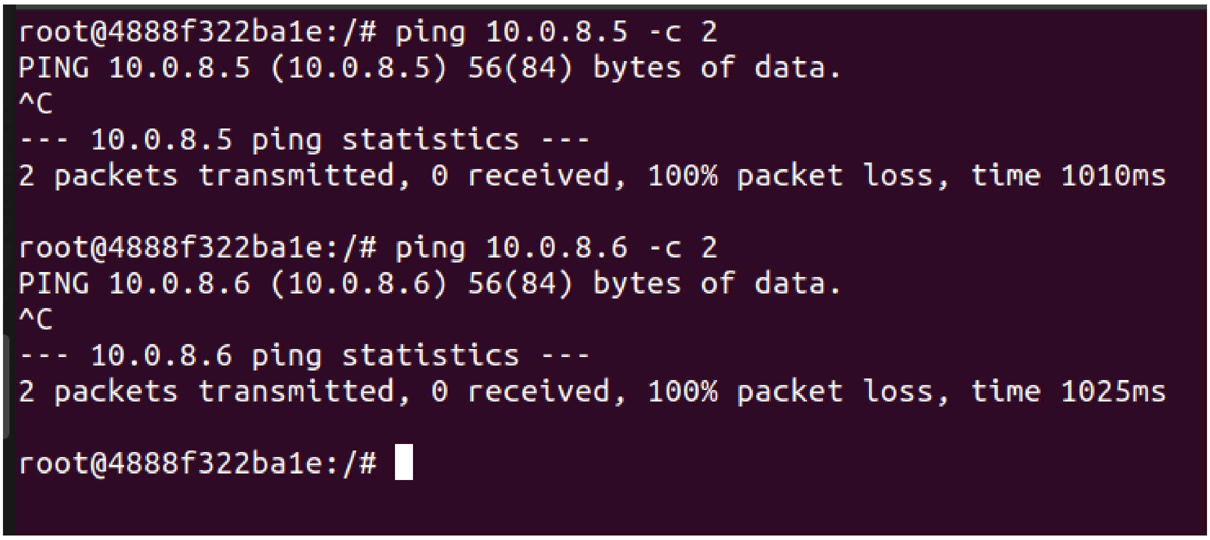
```
root@4888f322ba1e:/# ping 10.0.7.11 -c 2
PING 10.0.7.11 (10.0.7.11) 56(84) bytes of data.
64 bytes from 10.0.7.11: icmp_seq=1 ttl=64 time=0.214 ms
64 bytes from 10.0.7.11: icmp_seq=2 ttl=64 time=0.481 ms

--- 10.0.7.11 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1005ms
rtt min/avg/max/mdev = 0.214/0.347/0.481/0.133 ms
root@4888f322ba1e:/#
```

ทดลองว่า router สามารถ ping หา host ภายในทั้งสองเครื่องได้



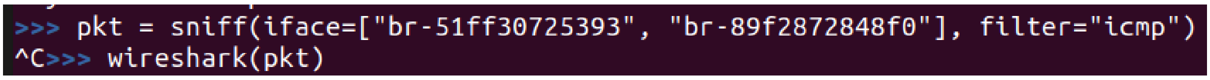
```
root@08e37cffd574:/# ping 10.0.8.5 -c 2
PING 10.0.8.5 (10.0.8.5) 56(84) bytes of data.
64 bytes from 10.0.8.5: icmp_seq=1 ttl=64 time=0.083 ms
64 bytes from 10.0.8.5: icmp_seq=2 ttl=64 time=0.191 ms

--- 10.0.8.5 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1021ms
rtt min/avg/max/mdev = 0.083/0.137/0.191/0.054 ms
root@08e37cffd574:/# ping 10.0.8.6 -c 2
PING 10.0.8.6 (10.0.8.6) 56(84) bytes of data.
64 bytes from 10.0.8.6: icmp_seq=1 ttl=64 time=0.076 ms
64 bytes from 10.0.8.6: icmp_seq=2 ttl=64 time=0.251 ms

--- 10.0.8.6 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1019ms
rtt min/avg/max/mdev = 0.076/0.163/0.251/0.087 ms
root@08e37cffd574:/#
```

ทดลองว่า client จะไม่สามารถ ping หา host ภายในได้

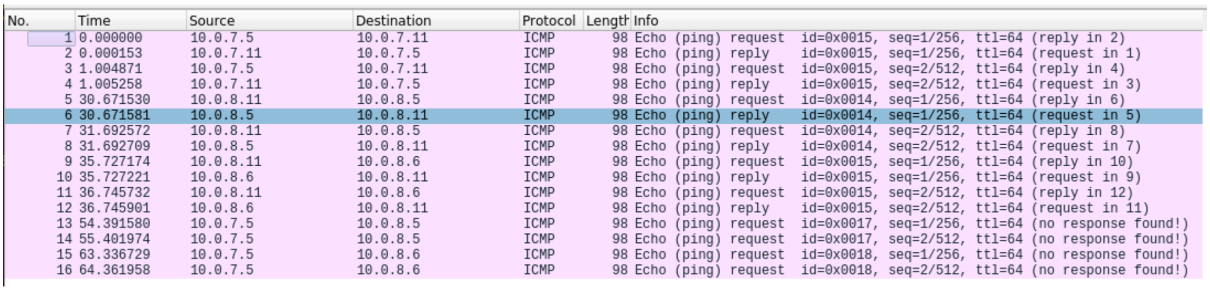


```
root@4888f322ba1e:/# ping 10.0.8.5 -c 2
PING 10.0.8.5 (10.0.8.5) 56(84) bytes of data.
^C
--- 10.0.8.5 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1010ms

root@4888f322ba1e:/# ping 10.0.8.6 -c 2
PING 10.0.8.6 (10.0.8.6) 56(84) bytes of data.
^C
--- 10.0.8.6 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1025ms

root@4888f322ba1e:/# 
```

ก่อนหน้าที่ทำการ ping ได้ใช้ scapy ตรวจจับ packet ไว้



```
>>> pkt = sniff(iface=["br-51ff30725393", "br-89f2872848f0"], filter="icmp")
^C>>> wireshark(pkt)
```

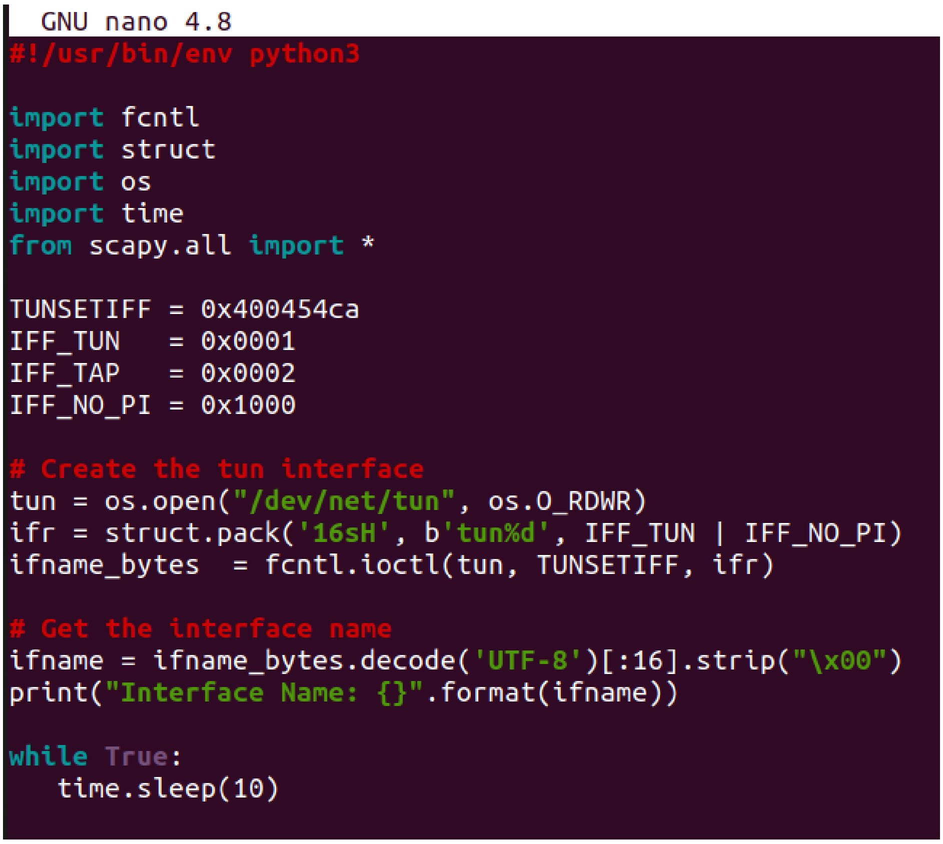


No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.7.5	10.0.7.11	ICMP	98	Echo (ping) request id=0x0015, seq=1/256, ttl=64 (reply in 2)
2	0.000153	10.0.7.11	10.0.7.5	ICMP	98	Echo (ping) reply id=0x0015, seq=1/256, ttl=64 (request in 1)
3	1.004871	10.0.7.5	10.0.7.11	ICMP	98	Echo (ping) request id=0x0015, seq=2/512, ttl=64 (reply in 4)
4	1.005258	10.0.7.11	10.0.7.5	ICMP	98	Echo (ping) reply id=0x0015, seq=2/512, ttl=64 (request in 3)
5	30.671530	10.0.8.11	10.0.8.5	ICMP	98	Echo (ping) request id=0x0014, seq=1/256, ttl=64 (reply in 6)
6	30.671581	10.0.8.5	10.0.8.11	ICMP	98	Echo (ping) reply id=0x0014, seq=1/256, ttl=64 (request in 5)
7	31.692572	10.0.8.11	10.0.8.5	ICMP	98	Echo (ping) request id=0x0014, seq=2/512, ttl=64 (reply in 8)
8	31.692709	10.0.8.5	10.0.8.11	ICMP	98	Echo (ping) reply id=0x0014, seq=2/512, ttl=64 (request in 7)
9	35.727174	10.0.8.11	10.0.8.6	ICMP	98	Echo (ping) request id=0x0015, seq=1/256, ttl=64 (reply in 10)
10	35.727221	10.0.8.6	10.0.8.11	ICMP	98	Echo (ping) reply id=0x0015, seq=1/256, ttl=64 (request in 9)
11	36.745732	10.0.8.11	10.0.8.6	ICMP	98	Echo (ping) request id=0x0015, seq=2/512, ttl=64 (reply in 12)
12	36.745901	10.0.8.6	10.0.8.11	ICMP	98	Echo (ping) reply id=0x0015, seq=2/512, ttl=64 (request in 11)
13	54.391580	10.0.7.5	10.0.8.5	ICMP	98	Echo (ping) request id=0x0017, seq=1/256, ttl=64 (no response found!)
14	55.401974	10.0.7.5	10.0.8.5	ICMP	98	Echo (ping) request id=0x0017, seq=2/512, ttl=64 (no response found!)
15	63.336729	10.0.7.5	10.0.8.6	ICMP	98	Echo (ping) request id=0x0018, seq=1/256, ttl=64 (no response found!)
16	64.361958	10.0.7.5	10.0.8.6	ICMP	98	Echo (ping) request id=0x0018, seq=2/512, ttl=64 (no response found!)

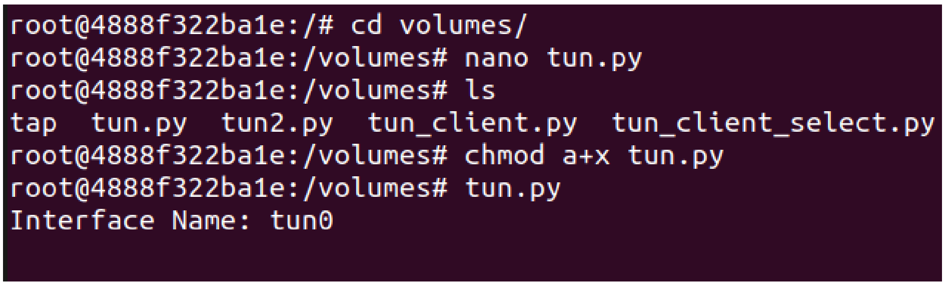
Task 2: Create and configure TUN interface

Task 2.a: Name of the interface

สร้าง TUN interface ที่ client 10.0.7.5

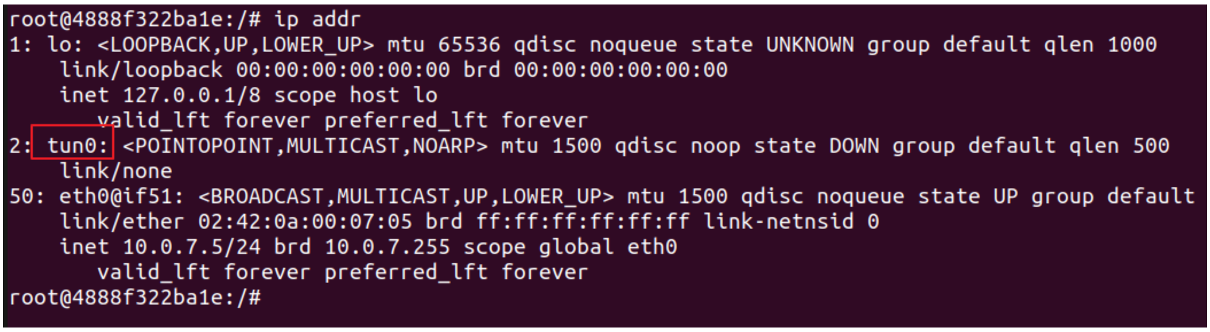


```
  GNU nano 4.8
#!/usr/bin/env python3

import fcntl
import struct
import os
import time
from scapy.all import *

TUNSETIFF = 0x400454ca
IFF_TUN   = 0x0001
IFF_TAP   = 0x0002
IFF_NO_PI = 0x1000

# Create the tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'tun%d', IFF_TUN | IFF_NO_PI)
ifname_bytes  = fcntl.ioctl(tun, TUNSETIFF, ifr)

# Get the interface name
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))

while True:
   time.sleep(10)
```



```
root@4888f322ba1e:/# cd volumes/
root@4888f322ba1e:/volumes# nano tun.py
root@4888f322ba1e:/volumes# ls
tap  tun.py  tun2.py  tun_client.py  tun_client_select.py
root@4888f322ba1e:/volumes# chmod a+x tun.py
root@4888f322ba1e:/volumes# tun.py
Interface Name: tun0
```

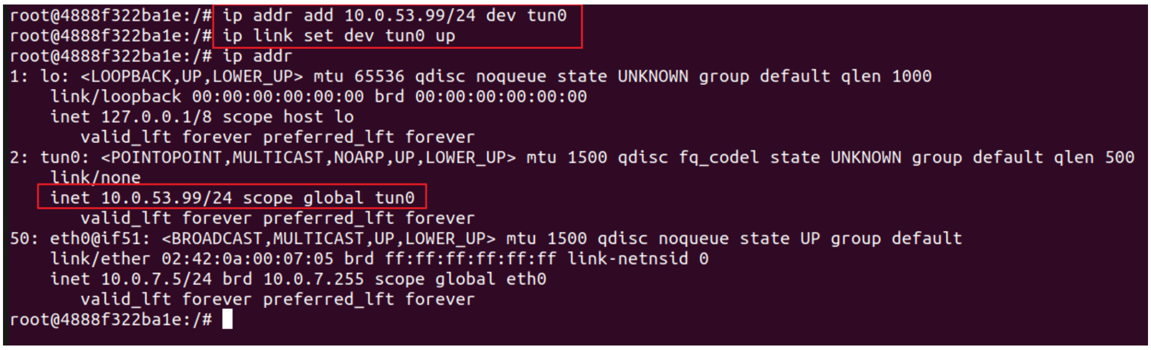
หลังจาก run code แล้วจะพบว่าที่ ip address จะมี interface tun0



```
root@4888f322ba1e:/# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: tun0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 500
    link/none
50: eth0@if51: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:00:07:05 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.0.7.5/24 brd 10.0.7.255 scope global eth0
       valid_lft forever preferred_lft forever
root@4888f322ba1e:/#
```

Task 2.b: Set up the TUN interface

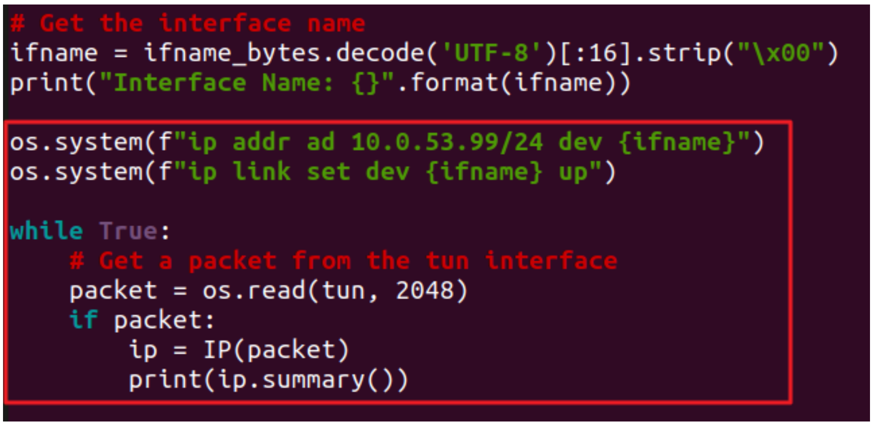
เพิ่ม IP ไปที่ tun0 และสั่ง up interface



```
root@4888f322ba1e:/# ip addr add 10.0.53.99/24 dev tun0
root@4888f322ba1e:/# ip link set dev tun0 up
root@4888f322ba1e:/# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 10.0.53.99/24 scope global tun0
       valid_lft forever preferred_lft forever
50: eth0@if51: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:00:07:05 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.0.7.5/24 brd 10.0.7.255 scope global eth0
       valid_lft forever preferred_lft forever
root@4888f322ba1e:/#
```

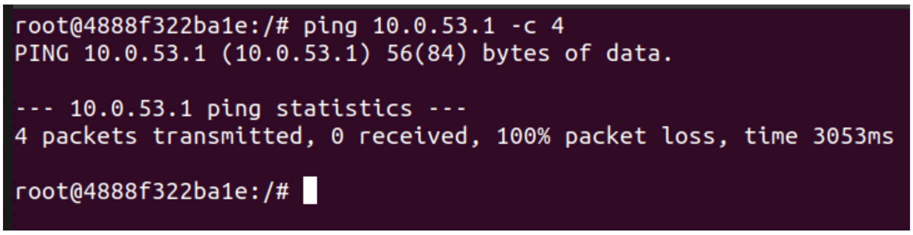
Task 2.c: Read from the TUN interface

ปรับโค้ดจากไฟล์เดิม เพื่อเพิ่ม IP และ interface up ผ่านโค้ด และให้ loop อ่าน packet ที่ tun0



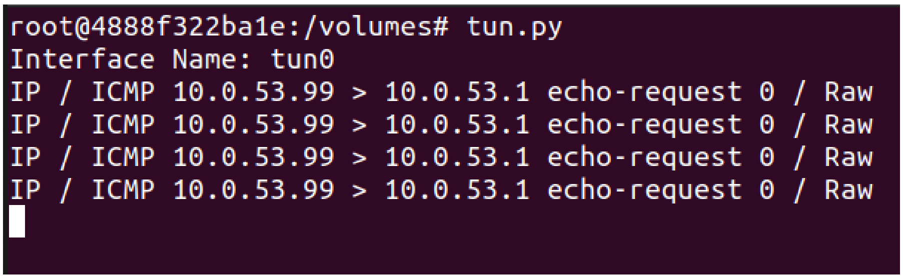
```python
# Get the interface name
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))

os.system(f"ip addr ad 10.0.53.99/24 dev {ifname}")
os.system(f"ip link set dev {ifname} up")

while True:
    # Get a packet from the tun interface
    packet = os.read(tun, 2048)
    if packet:
        ip = IP(packet)
        print(ip.summary())
```

ping 10.0.53.1 จาก client จะพบว่าไม่มีอะไรตอบกลับมา เนื่องจาก code ที่ใช้ยังไม่ได้รองรับการตอบกลับ



```
root@4888f322ba1e:/# ping 10.0.53.1 -c 4
PING 10.0.53.1 (10.0.53.1) 56(84) bytes of data.

--- 10.0.53.1 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3053ms

root@4888f322ba1e:/#
```

ในหน้าจอที่ run code tun.py จะพบว่าได้รับข้อมูล icmp ที่ส่งมา



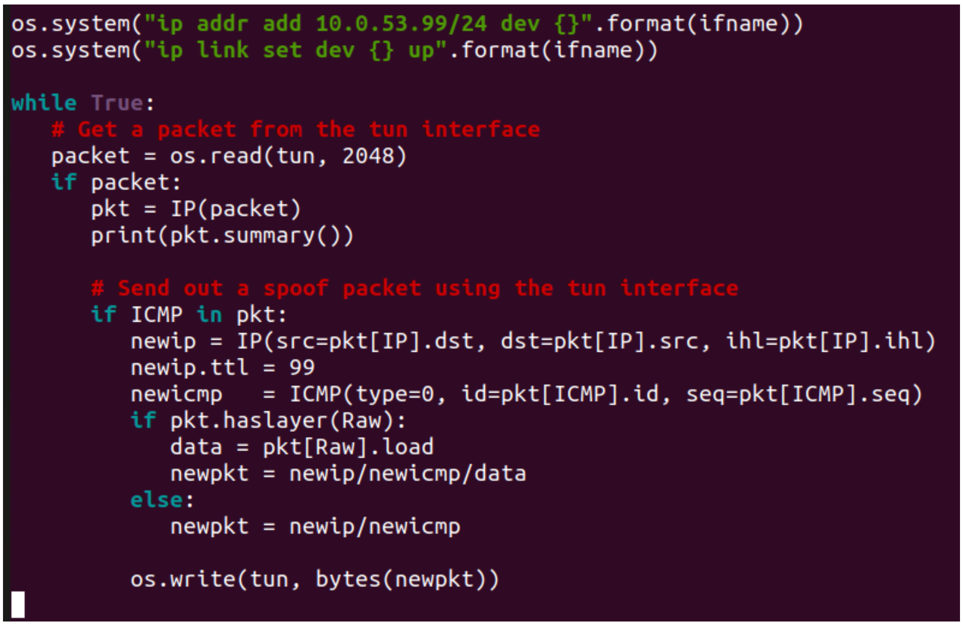
```
root@4888f322ba1e:/volumes# tun.py
Interface Name: tun0
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
```

และหากทดลอง ping 10.0.8.5 และ 10.0.8.6 จะพบว่า tun0 ไม่ได้รับค่าอะไรเนื่องจาก tun0 ยังมีแค่ฝั่งเดียวคือฝั่งที่ต่อกับ client

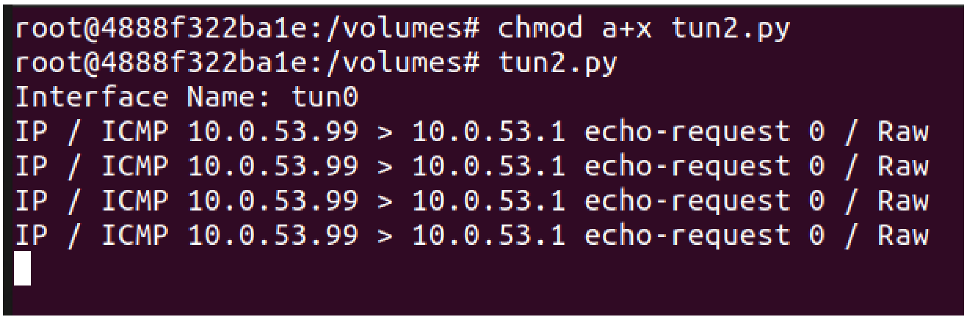
Task 2.d: Write to the TUN interface

จะใช้ code tun2.py ซึ่งเมื่อได้ packet และเป็น icmp packet request

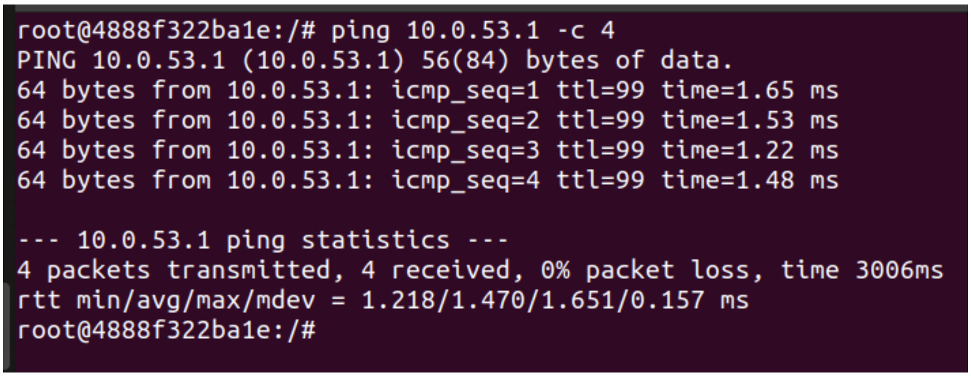
จะทำการสร้าง icmp echo reply กลับไปผ่านทาง tun0



```
os.system("ip addr add 10.0.53.99/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

while True:
   # Get a packet from the tun interface
   packet = os.read(tun, 2048)
   if packet:
      pkt = IP(packet)
      print(pkt.summary())

      # Send out a spoof packet using the tun interface
      if ICMP in pkt:
         newip = IP(src=pkt[IP].dst, dst=pkt[IP].src, ihl=pkt[IP].ihl)
         newip.ttl = 99
         newicmp   = ICMP(type=0, id=pkt[ICMP].id, seq=pkt[ICMP].seq)
         if pkt.haslayer(Raw):
            data = pkt[Raw].load
            newpkt = newip/newicmp/data
         else:
            newpkt = newip/newicmp

         os.write(tun, bytes(newpkt))
```



```
root@4888f322ba1e:/volumes# chmod a+x tun2.py
root@4888f322ba1e:/volumes# tun2.py
Interface Name: tun0
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
```



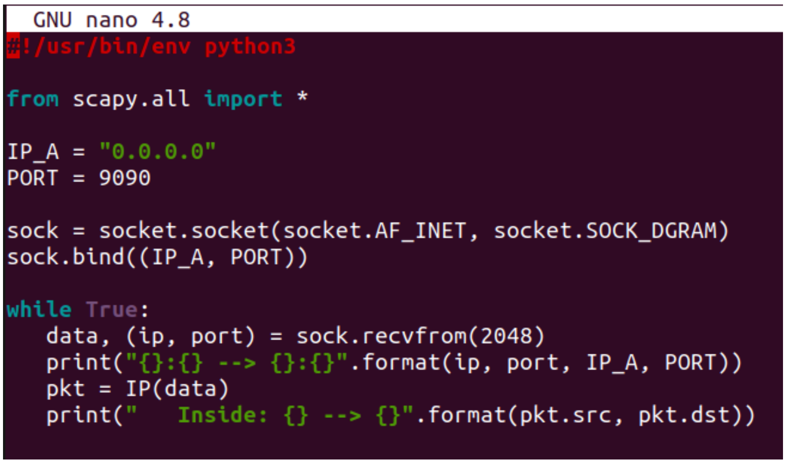
```
root@4888f322ba1e:/# ping 10.0.53.1 -c 4
PING 10.0.53.1 (10.0.53.1) 56(84) bytes of data.
64 bytes from 10.0.53.1: icmp_seq=1 ttl=99 time=1.65 ms
64 bytes from 10.0.53.1: icmp_seq=2 ttl=99 time=1.53 ms
64 bytes from 10.0.53.1: icmp_seq=3 ttl=99 time=1.22 ms
64 bytes from 10.0.53.1: icmp_seq=4 ttl=99 time=1.48 ms

--- 10.0.53.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 1.218/1.470/1.651/0.157 ms
root@4888f322ba1e:/#
```

จะพบว่า client ได้รับ icmp echo reply กลับมา

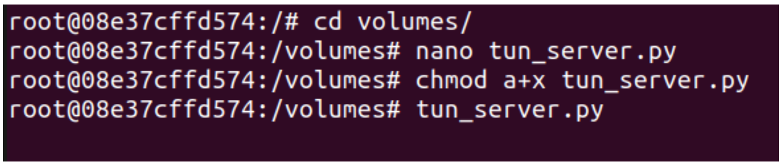
Task 3: Send the IP packet to VPN server through a tunnel

รับ ip packet ที่เข้า tun0 มาเป็น payload ของ UDP packet เพื่อทำ IP tunneling

Code ฝั่ง server-router

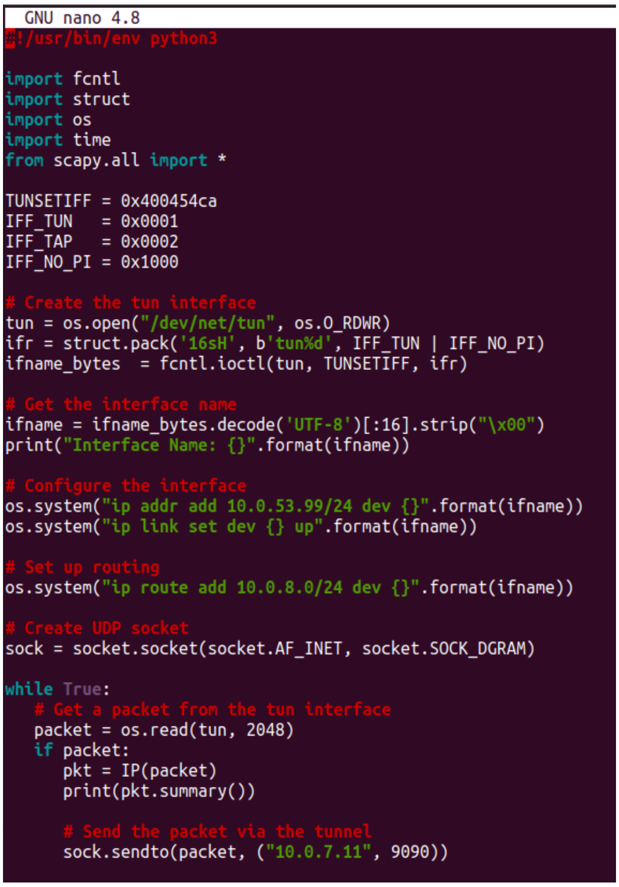


```
 GNU nano 4.8
#!/usr/bin/env python3

from scapy.all import *

IP_A = "0.0.0.0"
PORT = 9090

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((IP_A, PORT))

while True:
   data, (ip, port) = sock.recvfrom(2048)
   print("{}:{} --> {}:{}".format(ip, port, IP_A, PORT))
   pkt = IP(data)
   print("   Inside: {} --> {}".format(pkt.src, pkt.dst))
```

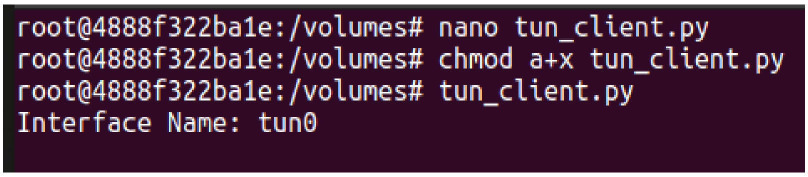


```
root@08e37cffd574:/# cd volumes/
root@08e37cffd574:/volumes# nano tun_server.py
root@08e37cffd574:/volumes# chmod a+x tun_server.py
root@08e37cffd574:/volumes# tun_server.py
```

Code ฝั่ง client

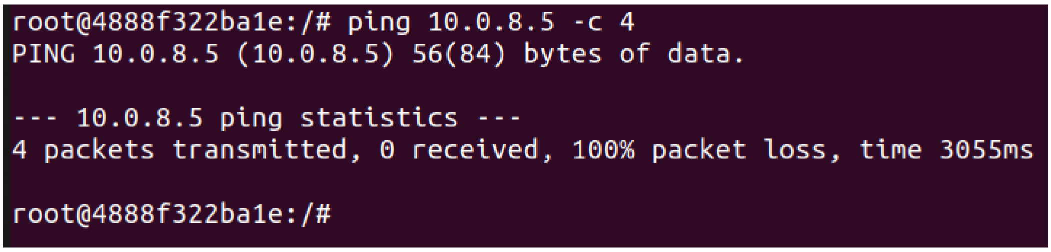


```
 GNU nano 4.8
#!/usr/bin/env python3

import fcntl
import struct
import os
import time
from scapy.all import *

TUNSETIFF = 0x400454ca
IFF_TUN   = 0x0001
IFF_TAP   = 0x0002
IFF_NO_PI = 0x1000

# Create the tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'tun%d', IFF_TUN | IFF_NO_PI)
ifname_bytes  = fcntl.ioctl(tun, TUNSETIFF, ifr)

# Get the interface name
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))

# Configure the interface
os.system("ip addr add 10.0.53.99/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

# Set up routing
os.system("ip route add 10.0.8.0/24 dev {}".format(ifname))

# Create UDP socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

while True:
   # Get a packet from the tun interface
   packet = os.read(tun, 2048)
   if packet:
      pkt = IP(packet)
      print(pkt.summary())

      # Send the packet via the tunnel
      sock.sendto(packet, ("10.0.7.11", 9090))
```

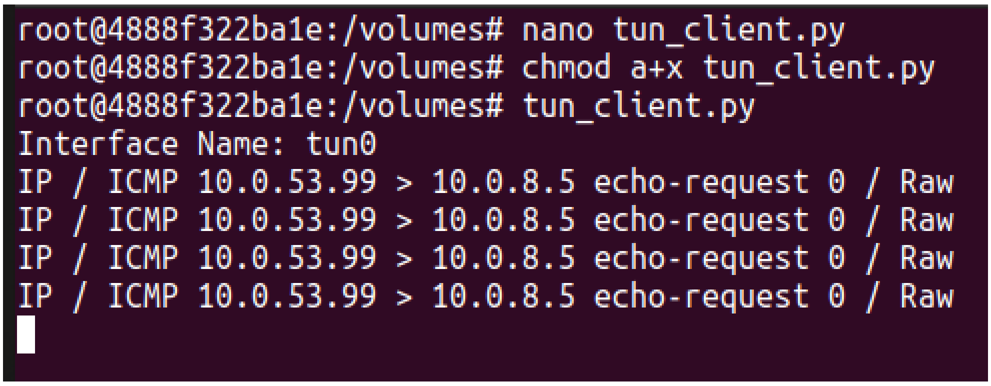


```
root@4888f322ba1e:/volumes# nano tun_client.py
root@4888f322ba1e:/volumes# chmod a+x tun_client.py
root@4888f322ba1e:/volumes# tun_client.py
Interface Name: tun0
```

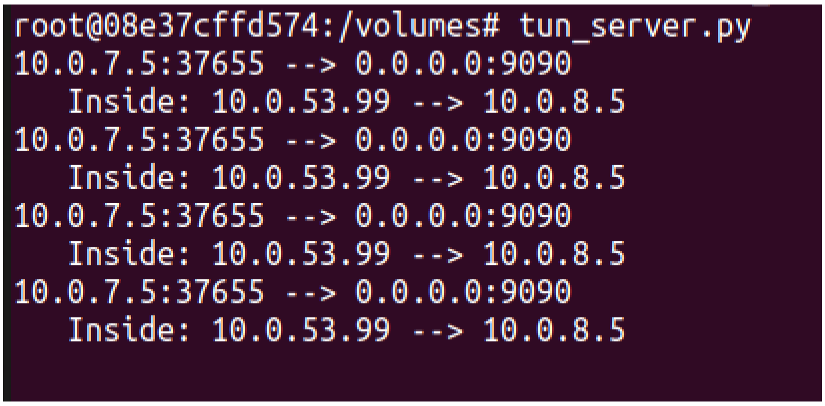
ลอง ping client -> host 10.0.8.5 จะพบว่ามี packet ผ่านทาง tun0 แล้วแต่ยังไม่มี icmp echo reply เนื่องจาก tunnel ยังไม่ได้ตั้งค่าให้ forwarding



```
root@4888f322ba1e:/# ping 10.0.8.5 -c 4
PING 10.0.8.5 (10.0.8.5) 56(84) bytes of data.

--- 10.0.8.5 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3055ms

root@4888f322ba1e:/#
```



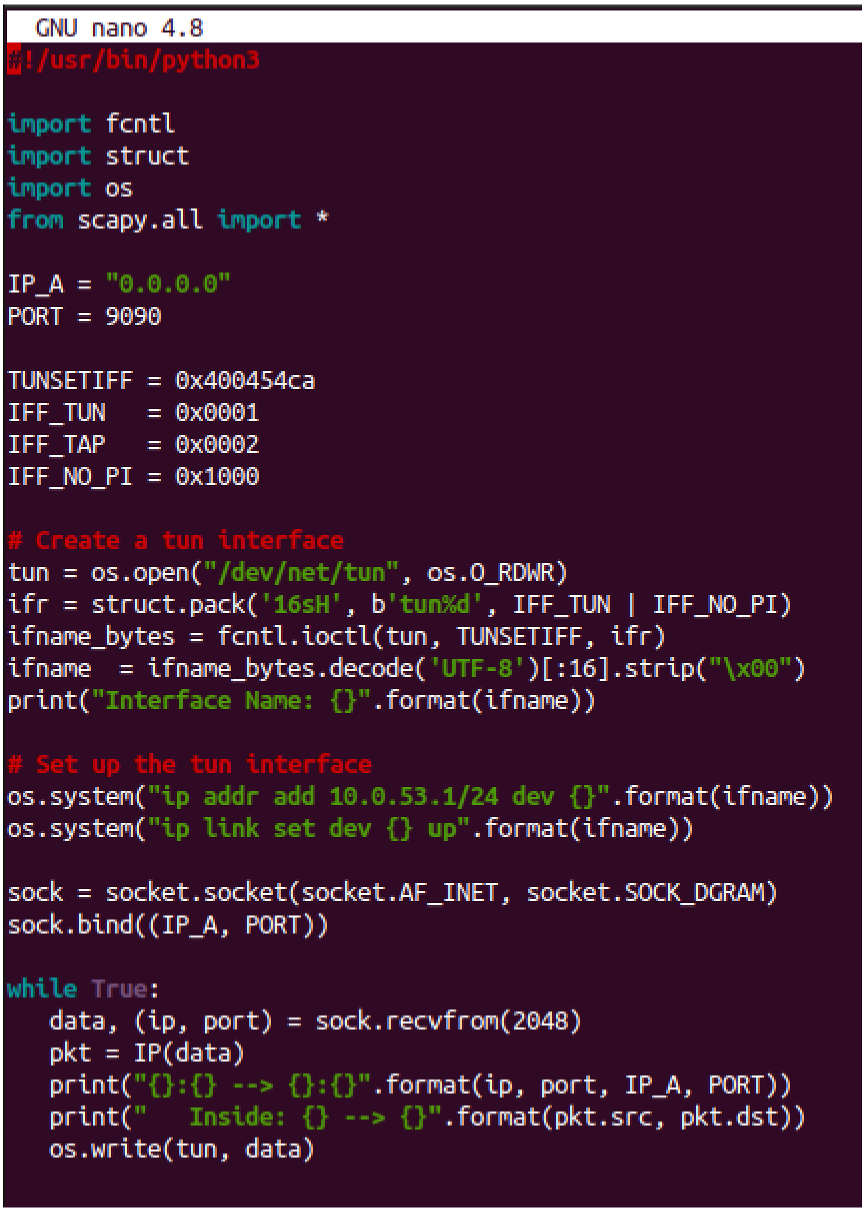
```
root@4888f322ba1e:/volumes# nano tun_client.py
root@4888f322ba1e:/volumes# chmod a+x tun_client.py
root@4888f322ba1e:/volumes# tun_client.py
Interface Name: tun0
IP / ICMP 10.0.53.99 > 10.0.8.5 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.8.5 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.8.5 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.8.5 echo-request 0 / Raw
```



```
root@08e37cffd574:/volumes# tun_server.py
10.0.7.5:37655 --> 0.0.0.0:9090
   Inside: 10.0.53.99 --> 10.0.8.5
10.0.7.5:37655 --> 0.0.0.0:9090
   Inside: 10.0.53.99 --> 10.0.8.5
10.0.7.5:37655 --> 0.0.0.0:9090
   Inside: 10.0.53.99 --> 10.0.8.5
10.0.7.5:37655 --> 0.0.0.0:9090
   Inside: 10.0.53.99 --> 10.0.8.5
```

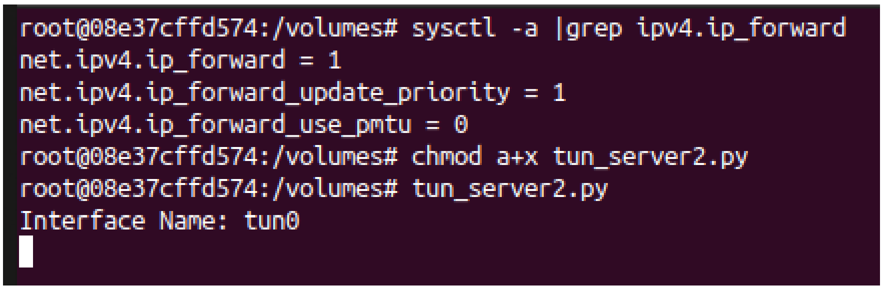
Task 4: Set up the VPN server

เปลี่ยนมาใช้ code tun_server2.py



```
 GNU nano 4.8
#!/usr/bin/python3

import fcntl
import struct
import os
from scapy.all import *

IP_A = "0.0.0.0"
PORT = 9090

TUNSETIFF = 0x400454ca
IFF_TUN   = 0x0001
IFF_TAP   = 0x0002
IFF_NO_PI = 0x1000

# Create a tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'tun%d', IFF_TUN | IFF_NO_PI)
ifname_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)
ifname  = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))

# Set up the tun interface
os.system("ip addr add 10.0.53.1/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((IP_A, PORT))

while True:
   data, (ip, port) = sock.recvfrom(2048)
   pkt = IP(data)
   print("{}:{} --> {}:{}".format(ip, port, IP_A, PORT))
   print("   Inside: {} --> {}".format(pkt.src, pkt.dst))
   os.write(tun, data)
```

เช็คว่ามีการเปิด ipv4.ip_forward ที่ server-router แล้ว และทำการ run code tun_server2.py



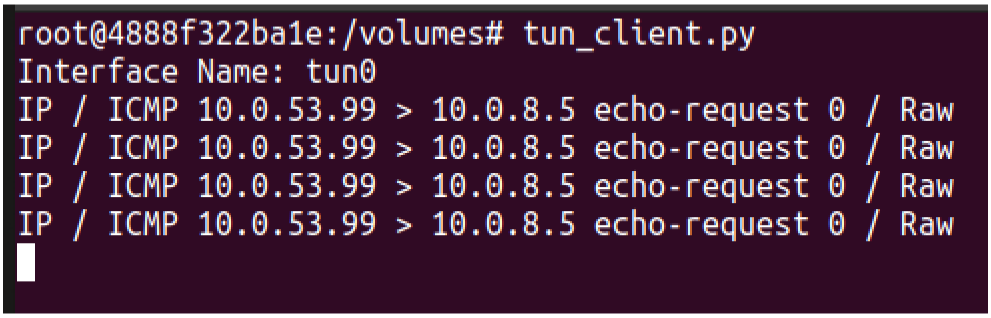
```
root@08e37cffd574:/volumes# sysctl -a |grep ipv4.ip_forward
net.ipv4.ip_forward = 1
net.ipv4.ip_forward_update_priority = 1
net.ipv4.ip_forward_use_pmtu = 0
root@08e37cffd574:/volumes# chmod a+x tun_server2.py
root@08e37cffd574:/volumes# tun_server2.py
Interface Name: tun0
```

หลังจากทำการ run tun_client.py ที่ client แล้วทำการ ping ไปที่ host 10.8.0.5

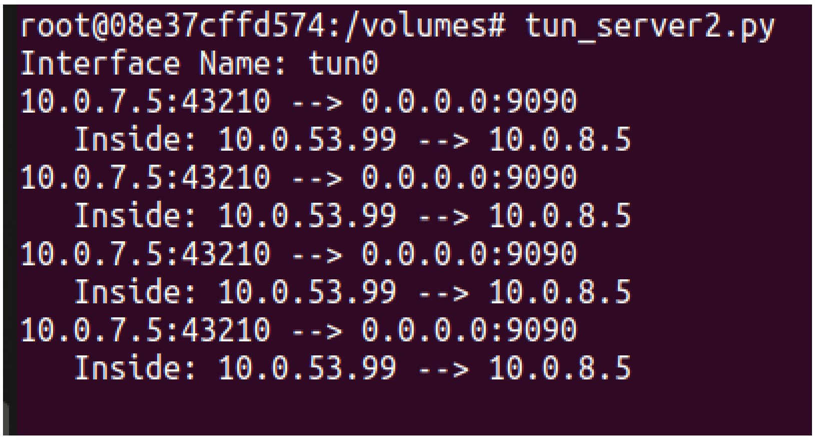
โดยที่ host 10.8.0.5 ได้ทำการ run tcpdump เพื่อรอดู packet

จะพบว่า host ได้รับ icmp echo request และมีการตอบ icmp echo reply กลับไปยัง tun0 ด้วย

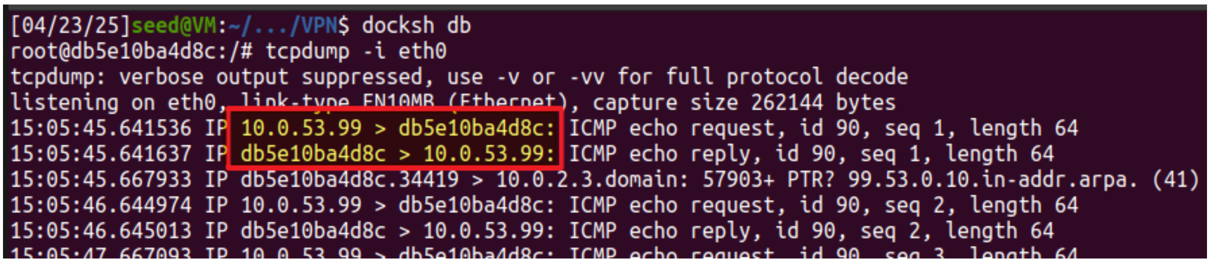
แต่ packet ยังไปไม่ถึง client เนื่องจาก tunnel ยังเป็นการส่งข้อมูลแบบทิศทางเดียวอยู่



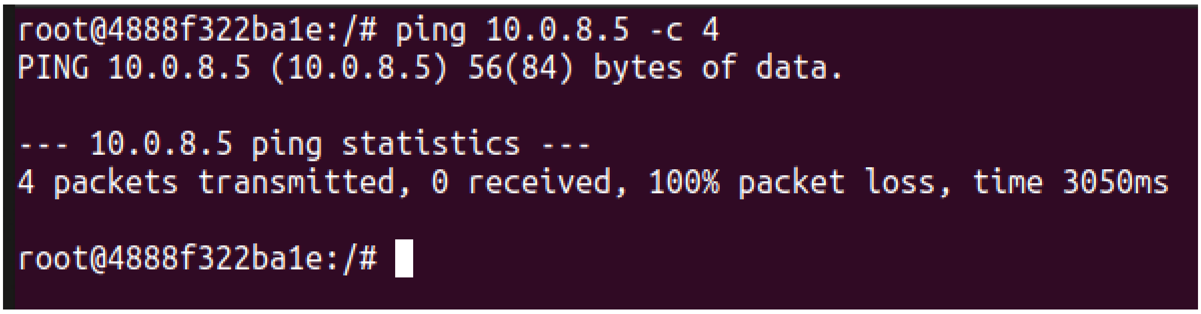
```
root@4888f322ba1e:/volumes# tun_client.py
Interface Name: tun0
IP / ICMP 10.0.53.99 > 10.0.8.5 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.8.5 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.8.5 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.8.5 echo-request 0 / Raw
```



```
root@08e37cffd574:/volumes# tun_server2.py
Interface Name: tun0
10.0.7.5:43210 --> 0.0.0.0:9090
   Inside: 10.0.53.99 --> 10.0.8.5
10.0.7.5:43210 --> 0.0.0.0:9090
   Inside: 10.0.53.99 --> 10.0.8.5
10.0.7.5:43210 --> 0.0.0.0:9090
   Inside: 10.0.53.99 --> 10.0.8.5
10.0.7.5:43210 --> 0.0.0.0:9090
   Inside: 10.0.53.99 --> 10.0.8.5
```



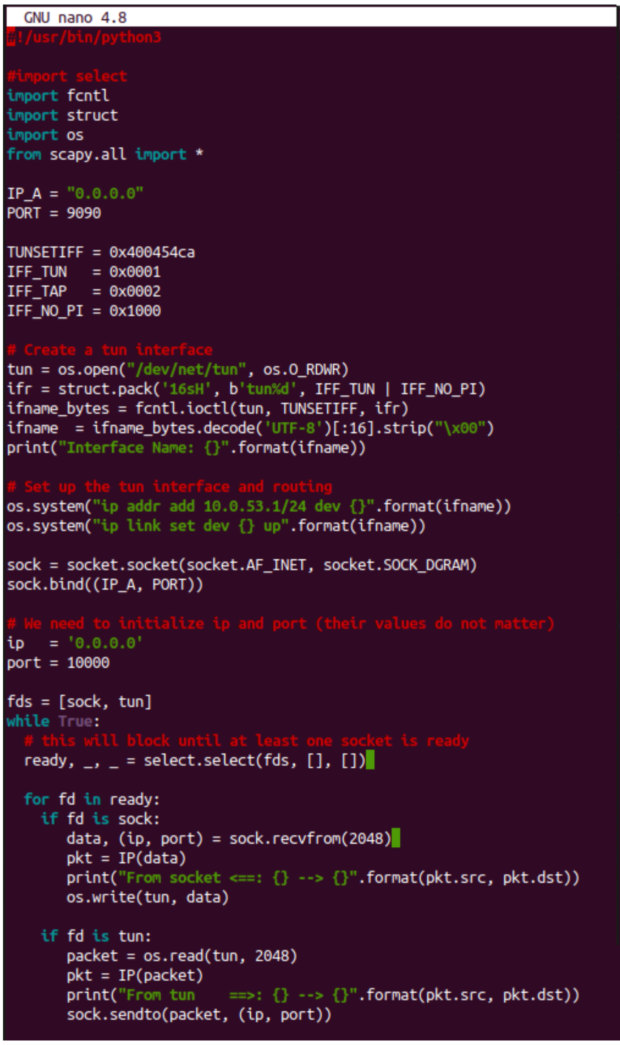
```
[04/23/25]seed@VM:~/.../VPN$ docksh db
root@db5e10ba4d8c:/# tcpdump -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
15:05:45.641536 IP 10.0.53.99 > db5e10ba4d8c: ICMP echo request, id 90, seq 1, length 64
15:05:45.641637 IP db5e10ba4d8c > 10.0.53.99: ICMP echo reply, id 90, seq 1, length 64
15:05:45.667933 IP db5e10ba4d8c.34419 > 10.0.2.3.domain: 57903+ PTR? 99.53.0.10.in-addr.arpa. (41)
15:05:46.644974 IP 10.0.53.99 > db5e10ba4d8c: ICMP echo request, id 90, seq 2, length 64
15:05:46.645013 IP db5e10ba4d8c > 10.0.53.99: ICMP echo reply, id 90, seq 2, length 64
15:05:47.667093 IP 10.0.53.99 > db5e10ba4d8c: ICMP echo request, id 90, seq 3, length 64
```



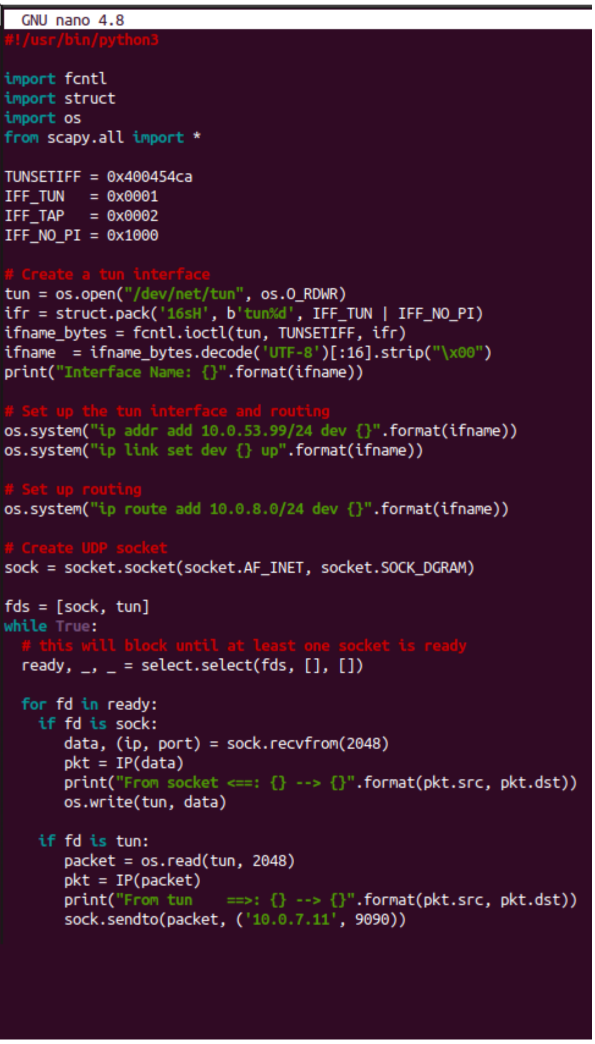
```
root@4888f322ba1e:/# ping 10.0.8.5 -c 4
PING 10.0.8.5 (10.0.8.5) 56(84) bytes of data.

--- 10.0.8.5 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3050ms

root@4888f322ba1e:/# 
```

Task 5: Handling traffic in both directions

รอบนี้จะใช้ tun_server_select.py และ tun_client_select.py

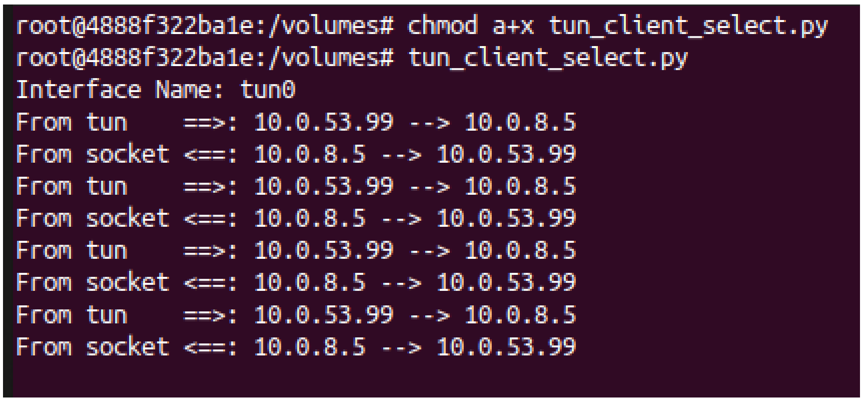


```
  GNU nano 4.8
#!/usr/bin/python3

#import select
import fcntl
import struct
import os
from scapy.all import *

IP_A = "0.0.0.0"
PORT = 9090

TUNSETIFF = 0x400454ca
IFF_TUN   = 0x0001
IFF_TAP   = 0x0002
IFF_NO_PI = 0x1000

# Create a tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'tun%d', IFF_TUN | IFF_NO_PI)
ifname_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)
ifname  = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))

# Set up the tun interface and routing
os.system("ip addr add 10.0.53.1/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((IP_A, PORT))

# We need to initialize ip and port (their values do not matter)
ip   = '0.0.0.0'
port = 10000

fds = [sock, tun]
while True:
  # this will block until at least one socket is ready
  ready, _, _ = select.select(fds, [], [])

  for fd in ready:
    if fd is sock:
       data, (ip, port) = sock.recvfrom(2048)
       pkt = IP(data)
       print("From socket <==: {} --> {}".format(pkt.src, pkt.dst))
       os.write(tun, data)

    if fd is tun:
       packet = os.read(tun, 2048)
       pkt = IP(packet)
       print("From tun    ==>: {} --> {}".format(pkt.src, pkt.dst))
       sock.sendto(packet, (ip, port))
```

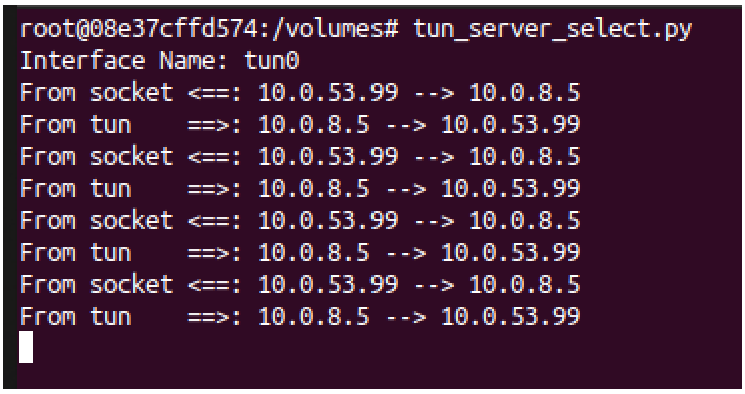


```
  GNU nano 4.8
#!/usr/bin/python3

import fcntl
import struct
import os
from scapy.all import *

TUNSETIFF = 0x400454ca
IFF_TUN   = 0x0001
IFF_TAP   = 0x0002
IFF_NO_PI = 0x1000

# Create a tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'tun%d', IFF_TUN | IFF_NO_PI)
ifname_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)
ifname  = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))

# Set up the tun interface and routing
os.system("ip addr add 10.0.53.99/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

# Set up routing
os.system("ip route add 10.0.8.0/24 dev {}".format(ifname))

# Create UDP socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

fds = [sock, tun]
while True:
  # this will block until at least one socket is ready
  ready, _, _ = select.select(fds, [], [])

  for fd in ready:
    if fd is sock:
       data, (ip, port) = sock.recvfrom(2048)
       pkt = IP(data)
       print("From socket <==: {} --> {}".format(pkt.src, pkt.dst))
       os.write(tun, data)

    if fd is tun:
       packet = os.read(tun, 2048)
       pkt = IP(packet)
       print("From tun    ==>: {} --> {}".format(pkt.src, pkt.dst))
       sock.sendto(packet, ('10.0.7.11', 9090))
```

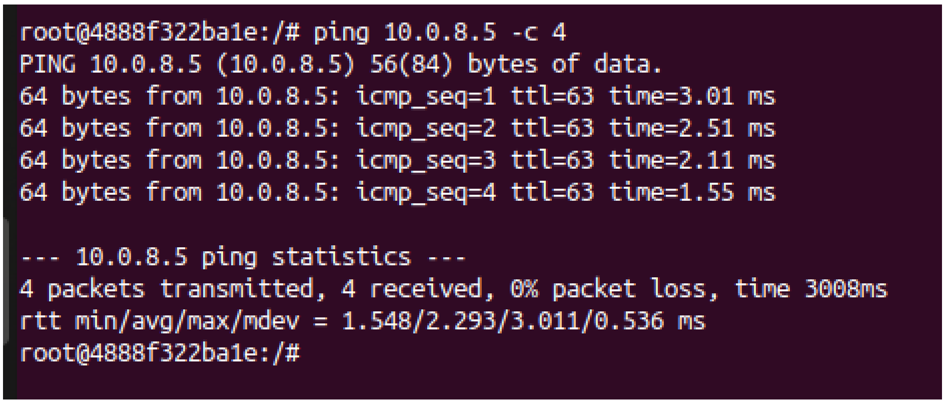
ทดลอง ping จาก client -> host 10.0.8.5



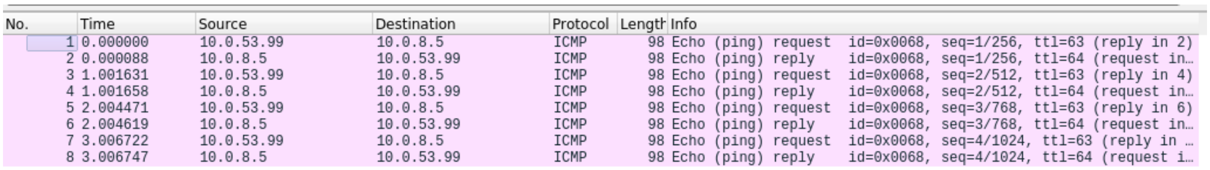
```
root@4888f322ba1e:/volumes# chmod a+x tun_client_select.py
root@4888f322ba1e:/volumes# tun_client_select.py
Interface Name: tun0
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.53.99
```



```
root@08e37cffd574:/volumes# tun_server_select.py
Interface Name: tun0
From socket <==: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.53.99
```



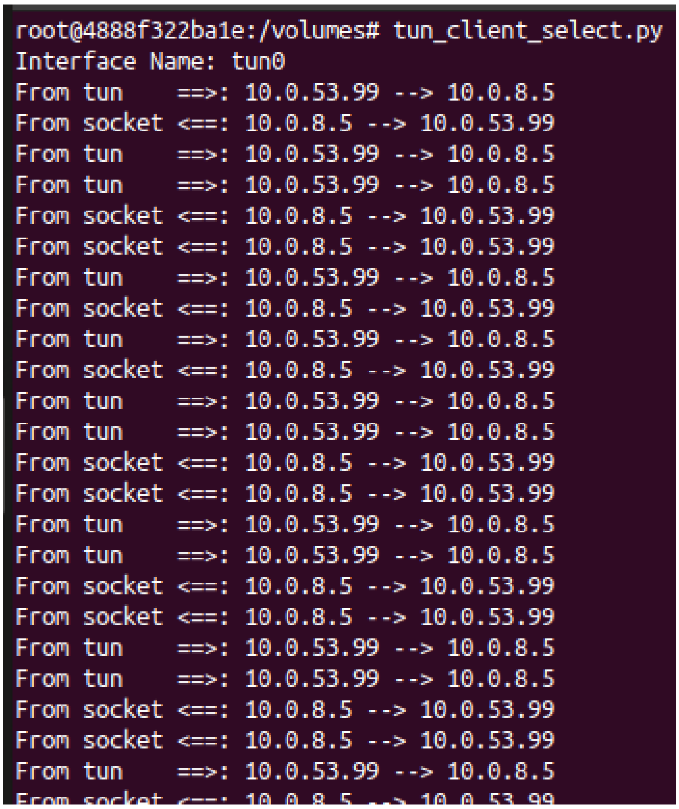
```
root@4888f322ba1e:/# ping 10.0.8.5 -c 4
PING 10.0.8.5 (10.0.8.5) 56(84) bytes of data.
64 bytes from 10.0.8.5: icmp_seq=1 ttl=63 time=3.01 ms
64 bytes from 10.0.8.5: icmp_seq=2 ttl=63 time=2.51 ms
64 bytes from 10.0.8.5: icmp_seq=3 ttl=63 time=2.11 ms
64 bytes from 10.0.8.5: icmp_seq=4 ttl=63 time=1.55 ms

--- 10.0.8.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3008ms
rtt min/avg/max/mdev = 1.548/2.293/3.011/0.536 ms
root@4888f322ba1e:/#
```



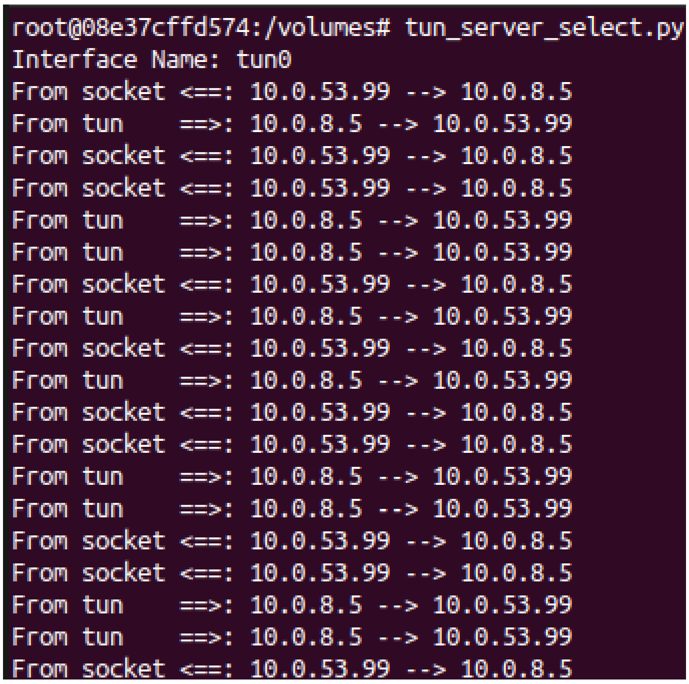
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.53.99	10.0.8.5	ICMP	98	Echo (ping) request id=0x0068, seq=1/256, ttl=63 (reply in 2)
2	0.000088	10.0.8.5	10.0.53.99	ICMP	98	Echo (ping) reply id=0x0068, seq=1/256, ttl=64 (request in…
3	1.001631	10.0.53.99	10.0.8.5	ICMP	98	Echo (ping) request id=0x0068, seq=2/512, ttl=63 (reply in 4)
4	1.001658	10.0.8.5	10.0.53.99	ICMP	98	Echo (ping) reply id=0x0068, seq=2/512, ttl=64 (request in…
5	2.004471	10.0.53.99	10.0.8.5	ICMP	98	Echo (ping) request id=0x0068, seq=3/768, ttl=63 (reply in 6)
6	2.004619	10.0.8.5	10.0.53.99	ICMP	98	Echo (ping) reply id=0x0068, seq=3/768, ttl=64 (request in…
7	3.006722	10.0.53.99	10.0.8.5	ICMP	98	Echo (ping) request id=0x0068, seq=4/1024, ttl=63 (reply in …
8	3.006747	10.0.8.5	10.0.53.99	ICMP	98	Echo (ping) reply id=0x0068, seq=4/1024, ttl=64 (request i…

จะพบว่ามี client ได้รับ icmp echo reply แล้ว เนื่องจาก tunnel สามารถสื่อสารสองทิศทางได้

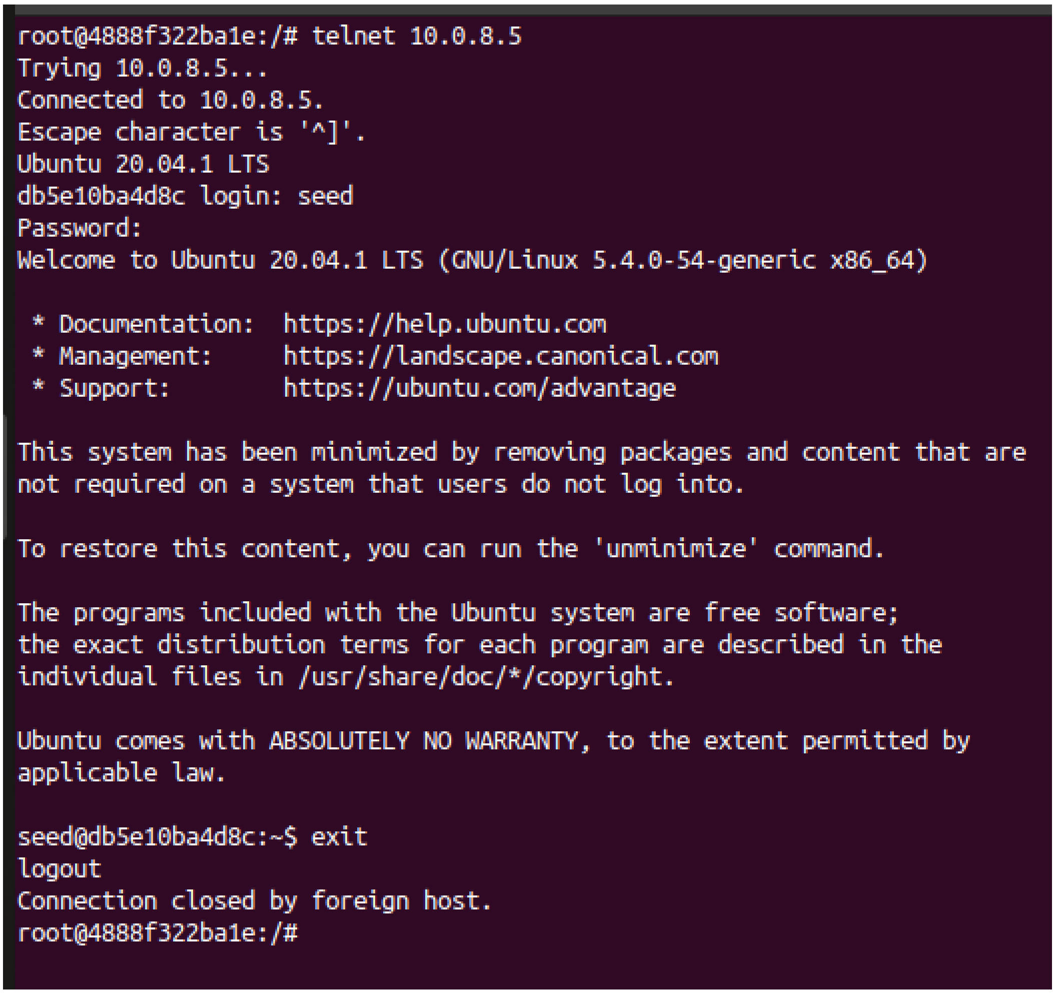
ทดลอง telnet จาก client -> host 10.0.8.5



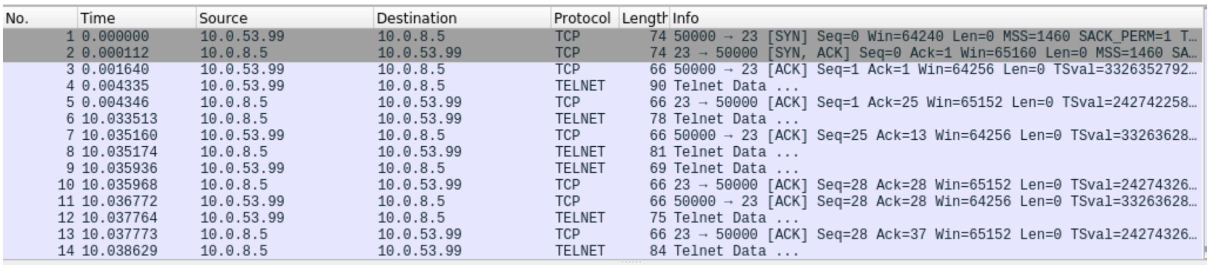
```
root@4888f322ba1e:/volumes# tun_client_select.py
Interface Name: tun0
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.53.99
```



```
root@08e37cffd574:/volumes# tun_server_select.py
Interface Name: tun0
From socket <==: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.53.99 --> 10.0.8.5
```



```
root@4888f322ba1e:/# telnet 10.0.8.5
Trying 10.0.8.5...
Connected to 10.0.8.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
db5e10ba4d8c login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

seed@db5e10ba4d8c:~$ exit
logout
Connection closed by foreign host.
root@4888f322ba1e:/#
```

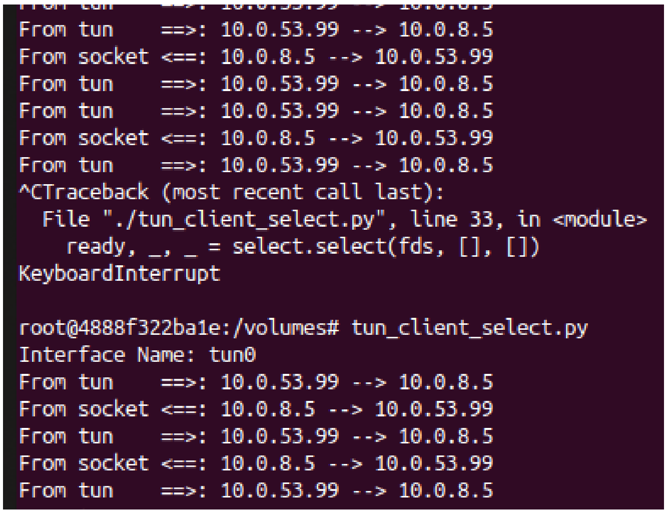


No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.53.99	10.0.8.5	TCP	74	50000 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 T…
2	0.000112	10.0.8.5	10.0.53.99	TCP	74	23 → 50000 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SA…
3	0.001640	10.0.53.99	10.0.8.5	TCP	66	50000 → 23 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3326352792…
4	0.004335	10.0.53.99	10.0.8.5	TELNET	90	Telnet Data ...
5	0.004346	10.0.8.5	10.0.53.99	TCP	66	23 → 50000 [ACK] Seq=1 Ack=25 Win=65152 Len=0 TSval=242742258…
6	10.033513	10.0.8.5	10.0.53.99	TELNET	78	Telnet Data ...
7	10.035160	10.0.53.99	10.0.8.5	TCP	66	50000 → 23 [ACK] Seq=25 Ack=13 Win=64256 Len=0 TSval=33263628…
8	10.035174	10.0.8.5	10.0.53.99	TELNET	81	Telnet Data ...
9	10.035936	10.0.53.99	10.0.8.5	TELNET	69	Telnet Data ...
10	10.035968	10.0.8.5	10.0.53.99	TCP	66	23 → 50000 [ACK] Seq=28 Ack=28 Win=65152 Len=0 TSval=24274326…
11	10.036772	10.0.53.99	10.0.8.5	TCP	66	50000 → 23 [ACK] Seq=28 Ack=28 Win=64256 Len=0 TSval=33263628…
12	10.037764	10.0.53.99	10.0.8.5	TELNET	75	Telnet Data ...
13	10.037773	10.0.8.5	10.0.53.99	TCP	66	23 → 50000 [ACK] Seq=28 Ack=37 Win=65152 Len=0 TSval=24274326…
14	10.038629	10.0.8.5	10.0.53.99	TELNET	84	Telnet Data ...

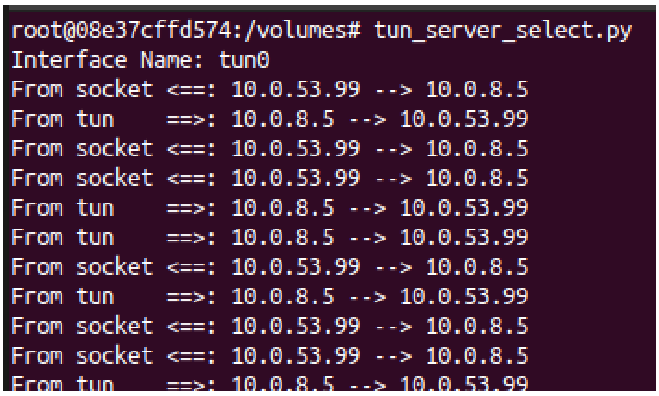
ผลที่ได้คือสามารถ telnet ได้สำเร็จ

Task 6: Tunneling-Breaking experiment

ทำการทดลองตัดการเชื่อมต่อของ tunnel ฝั่งหนึ่งออกระหว่างที่ telnet จะพบว่า client ไม่สามารถพิมพ์ต่อได้ แต่ถ้าหากเชื่อม tunnel ใหม่อีกครั้ง session telnet จะกลับมาพิมพ์ต่อจากเดิมได้

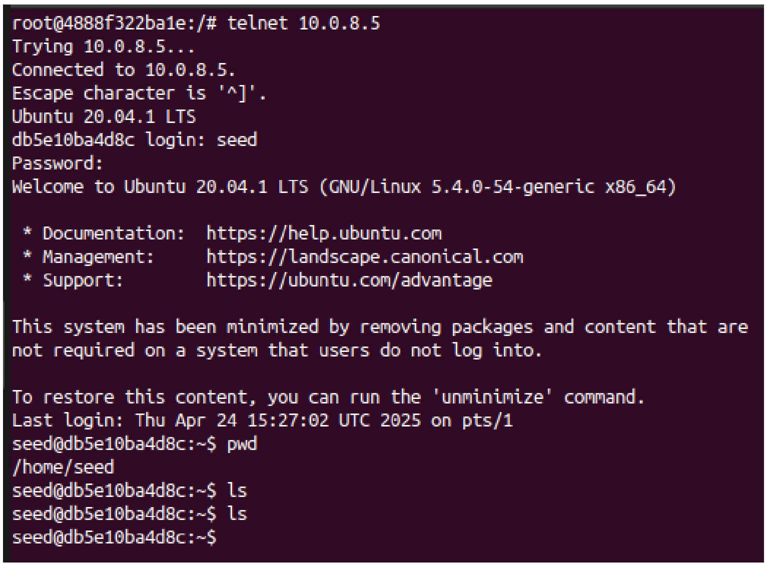


```
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.53.99 --> 10.0.8.5
^CTraceback (most recent call last):
  File "./tun_client_select.py", line 33, in <module>
    ready, _, _ = select.select(fds, [], [])
KeyboardInterrupt

root@4888f322ba1e:/volumes# tun_client_select.py
Interface Name: tun0
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.53.99 --> 10.0.8.5
```



```
root@08e37cffd574:/volumes# tun_server_select.py
Interface Name: tun0
From socket <==: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.53.99
```

จังหวะที่พิมพ์ ls ถูกตัดการเชื่อมต่อไปแล้ว และมีการพิมพ์ซ้ำอีกครั้ง หลังเชื่อมต่อใหม่จึงมีการเบิ้ลสองครั้ง

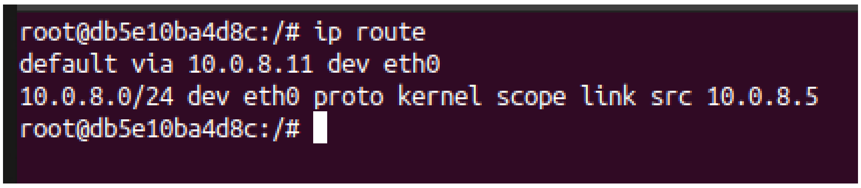


```
root@4888f322ba1e:/# telnet 10.0.8.5
Trying 10.0.8.5...
Connected to 10.0.8.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
db5e10ba4d8c login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Thu Apr 24 15:27:02 UTC 2025 on pts/1
seed@db5e10ba4d8c:~$ pwd
/home/seed
seed@db5e10ba4d8c:~$ ls
seed@db5e10ba4d8c:~$ ls
seed@db5e10ba4d8c:~$
```

Task 7: Routing experiment on host V (10.0.8.5)

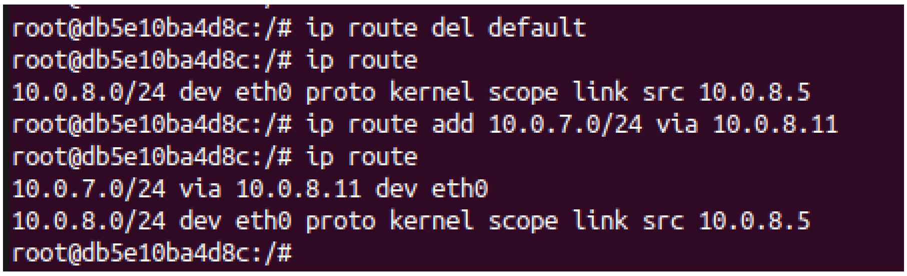
ลบ default route ที่ host 10.0.8.5 แล้วเพิ่ม route ให้ผ่าน router (vpn server)

Ip route ก่อนลบ

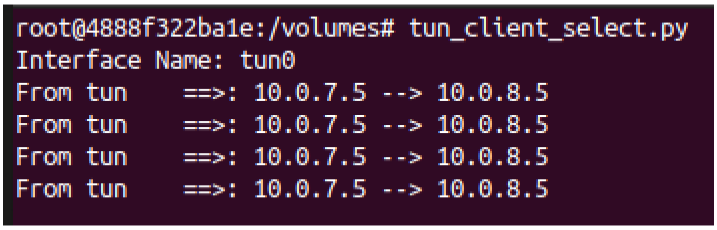


```
root@db5e10ba4d8c:/# ip route
default via 10.0.8.11 dev eth0
10.0.8.0/24 dev eth0 proto kernel scope link src 10.0.8.5
root@db5e10ba4d8c:/# 
```

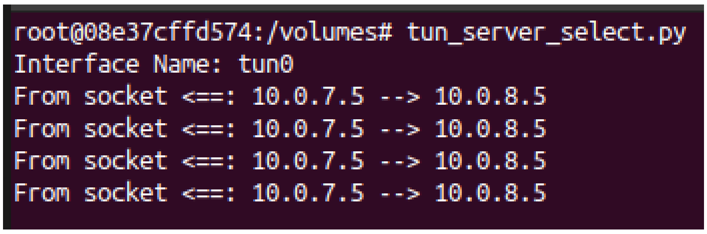
Ip route หลังลบ



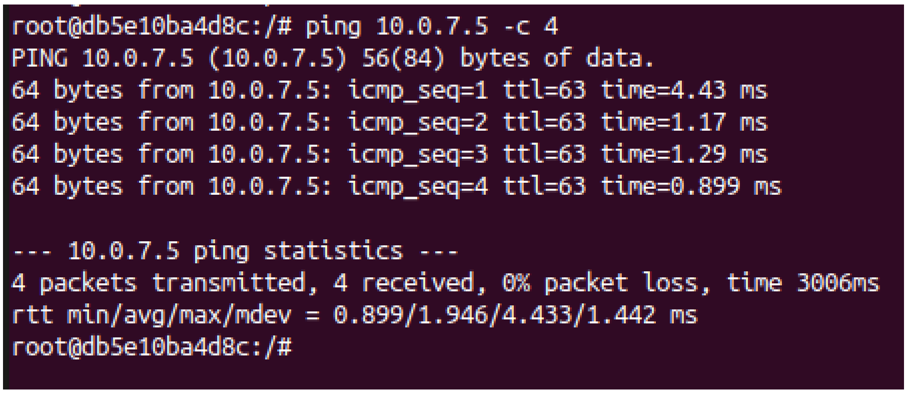
```
root@db5e10ba4d8c:/# ip route del default
root@db5e10ba4d8c:/# ip route
10.0.8.0/24 dev eth0 proto kernel scope link src 10.0.8.5
root@db5e10ba4d8c:/# ip route add 10.0.7.0/24 via 10.0.8.11
root@db5e10ba4d8c:/# ip route
10.0.7.0/24 via 10.0.8.11 dev eth0
10.0.8.0/24 dev eth0 proto kernel scope link src 10.0.8.5
root@db5e10ba4d8c:/#
```



```
root@4888f322ba1e:/volumes# tun_client_select.py
Interface Name: tun0
From tun    ==>: 10.0.7.5 --> 10.0.8.5
From tun    ==>: 10.0.7.5 --> 10.0.8.5
From tun    ==>: 10.0.7.5 --> 10.0.8.5
From tun    ==>: 10.0.7.5 --> 10.0.8.5
```



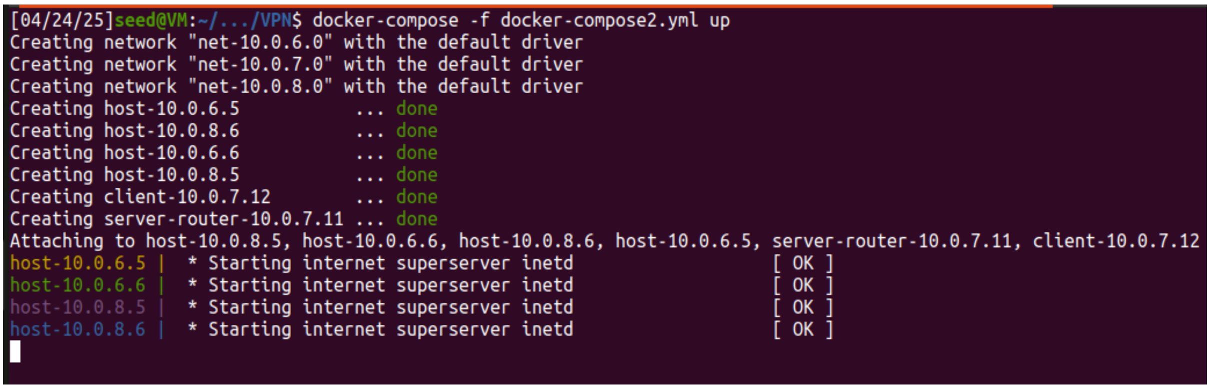
```
root@08e37cffd574:/volumes# tun_server_select.py
Interface Name: tun0
From socket <==: 10.0.7.5 --> 10.0.8.5
From socket <==: 10.0.7.5 --> 10.0.8.5
From socket <==: 10.0.7.5 --> 10.0.8.5
From socket <==: 10.0.7.5 --> 10.0.8.5
```



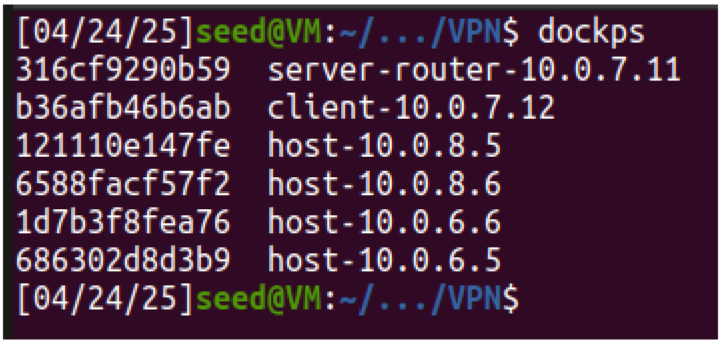
```
root@db5e10ba4d8c:/# ping 10.0.7.5 -c 4
PING 10.0.7.5 (10.0.7.5) 56(84) bytes of data.
64 bytes from 10.0.7.5: icmp_seq=1 ttl=63 time=4.43 ms
64 bytes from 10.0.7.5: icmp_seq=2 ttl=63 time=1.17 ms
64 bytes from 10.0.7.5: icmp_seq=3 ttl=63 time=1.29 ms
64 bytes from 10.0.7.5: icmp_seq=4 ttl=63 time=0.899 ms

--- 10.0.7.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 0.899/1.946/4.433/1.442 ms
root@db5e10ba4d8c:/#
```

Task 8:

ใช้ container docker-compose2.yml

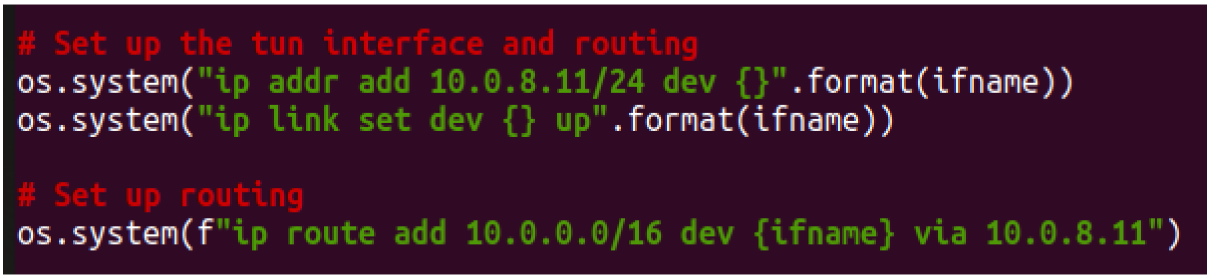


```
[04/24/25]seed@VM:~/.../VPN$ docker-compose -f docker-compose2.yml up
Creating network "net-10.0.6.0" with the default driver
Creating network "net-10.0.7.0" with the default driver
Creating network "net-10.0.8.0" with the default driver
Creating host-10.0.6.5           ... done
Creating host-10.0.8.6           ... done
Creating host-10.0.6.6           ... done
Creating host-10.0.8.5           ... done
Creating client-10.0.7.12        ... done
Creating server-router-10.0.7.11 ... done
Attaching to host-10.0.8.5, host-10.0.6.6, host-10.0.8.6, host-10.0.6.5, server-router-10.0.7.11, client-10.0.7.12
host-10.0.6.5 |  * Starting internet superserver inetd                  [ OK ]
host-10.0.6.6 |  * Starting internet superserver inetd                  [ OK ]
host-10.0.8.5 |  * Starting internet superserver inetd                  [ OK ]
host-10.0.8.6 |  * Starting internet superserver inetd                  [ OK ]
```



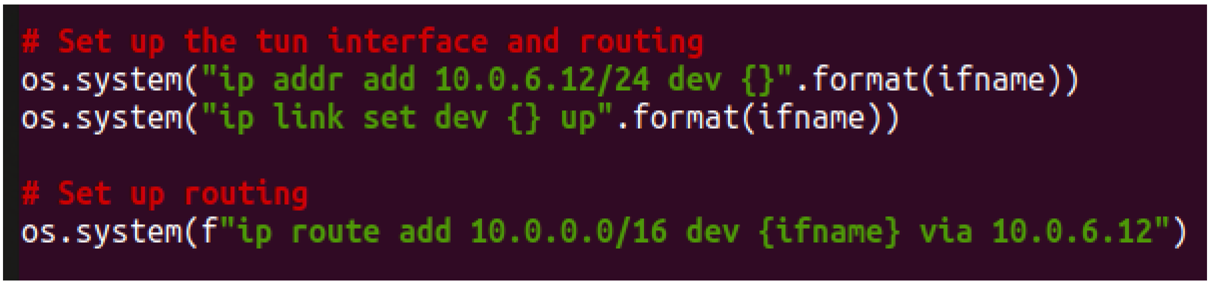
```
[04/24/25]seed@VM:~/.../VPN$ dockps
316cf9290b59  server-router-10.0.7.11
b36afb46b6ab  client-10.0.7.12
121110e147fe  host-10.0.8.5
6588facf57f2  host-10.0.8.6
1d7b3f8fea76  host-10.0.6.6
686302d8d3b9  host-10.0.6.5
[04/24/25]seed@VM:~/.../VPN$
```

ปรับแก้ tun_server_select.py ให้รันที่ server-router 10.0.7.11



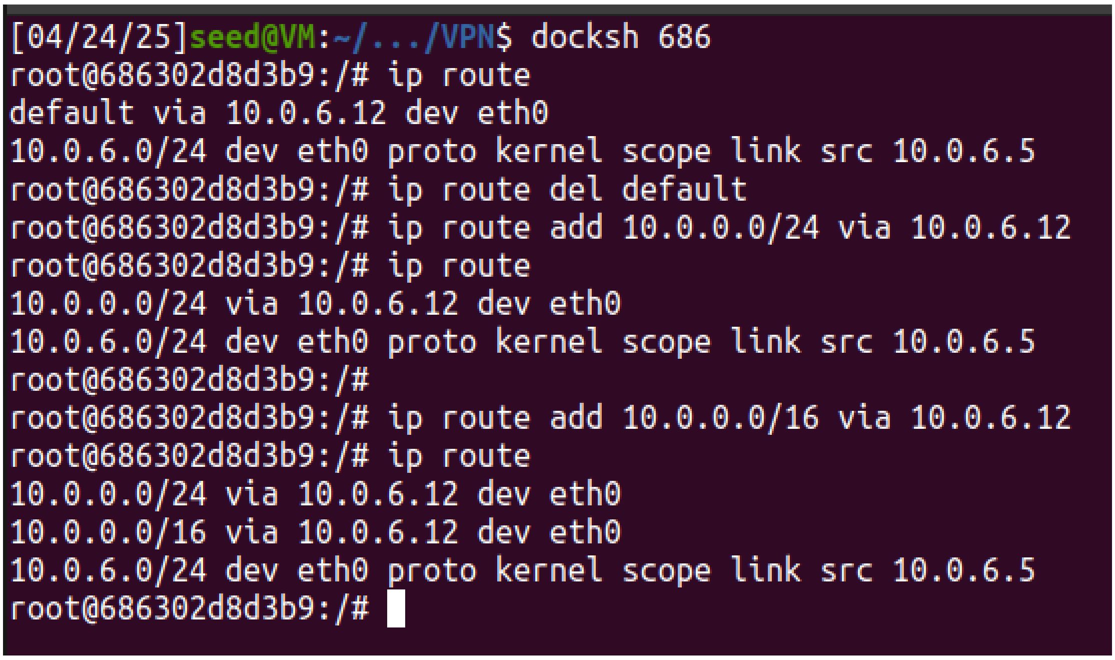
```
# Set up the tun interface and routing
os.system("ip addr add 10.0.8.11/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

# Set up routing
os.system(f"ip route add 10.0.0.0/16 dev {ifname} via 10.0.8.11")
```

และ tun_client_select.py ให้รันที่ client 10.0.7.12



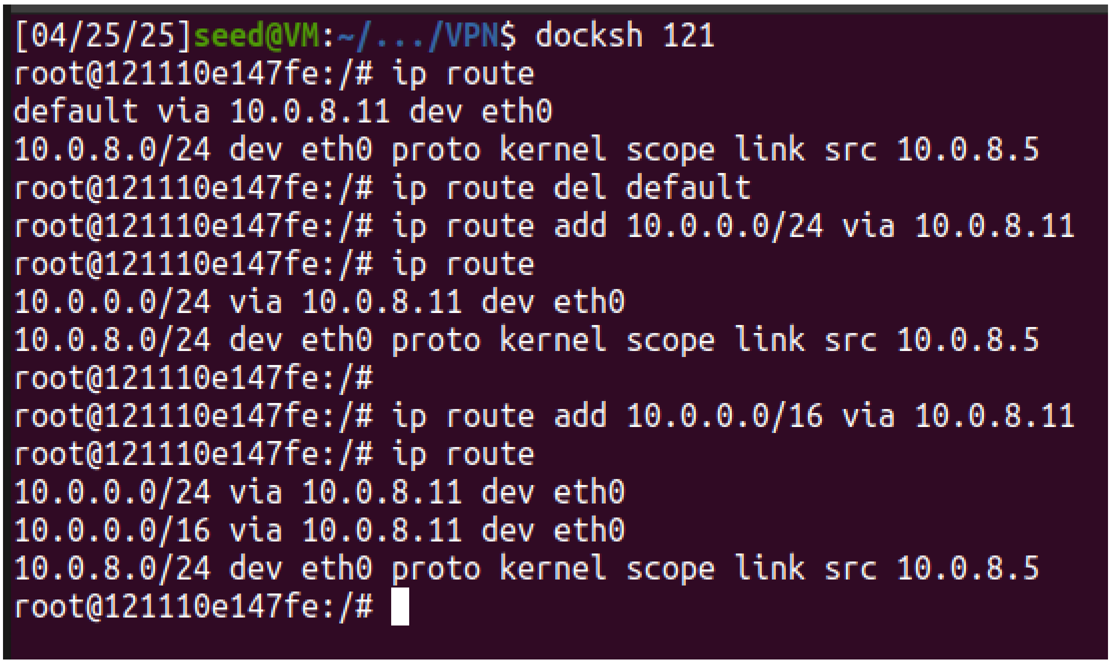
```
# Set up the tun interface and routing
os.system("ip addr add 10.0.6.12/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

# Set up routing
os.system(f"ip route add 10.0.0.0/16 dev {ifname} via 10.0.6.12")
```

Host 10.0.6.5 แก้ไข ip route



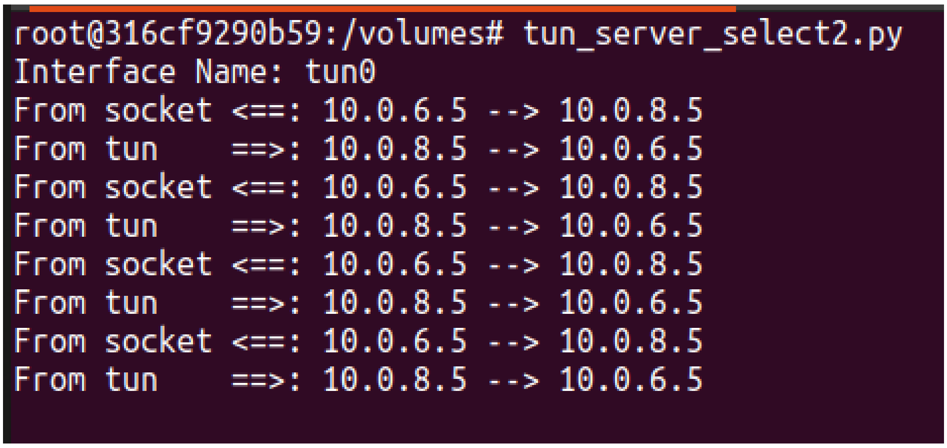
```
[04/24/25]seed@VM:~/.../VPN$ docksh 686
root@686302d8d3b9:/# ip route
default via 10.0.6.12 dev eth0
10.0.6.0/24 dev eth0 proto kernel scope link src 10.0.6.5
root@686302d8d3b9:/# ip route del default
root@686302d8d3b9:/# ip route add 10.0.0.0/24 via 10.0.6.12
root@686302d8d3b9:/# ip route
10.0.0.0/24 via 10.0.6.12 dev eth0
10.0.6.0/24 dev eth0 proto kernel scope link src 10.0.6.5
root@686302d8d3b9:/#
root@686302d8d3b9:/# ip route add 10.0.0.0/16 via 10.0.6.12
root@686302d8d3b9:/# ip route
10.0.0.0/24 via 10.0.6.12 dev eth0
10.0.0.0/16 via 10.0.6.12 dev eth0
10.0.6.0/24 dev eth0 proto kernel scope link src 10.0.6.5
root@686302d8d3b9:/#
```

Host 10.0.8.5 แก้ไข ip route

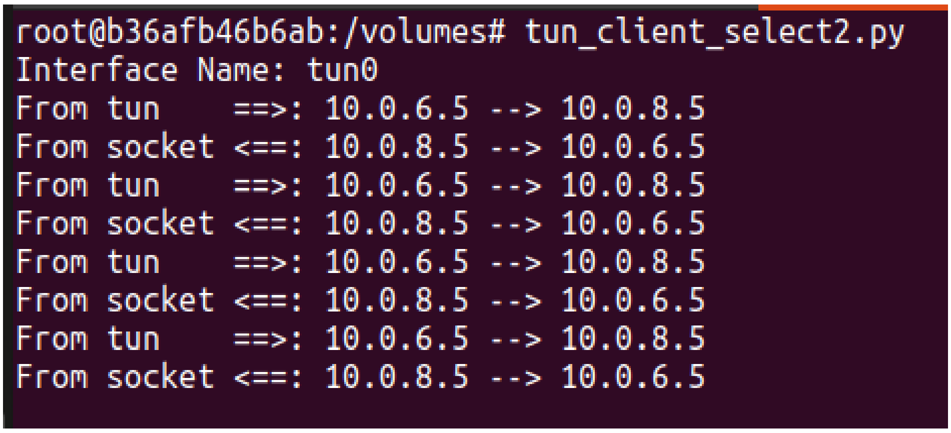


```
[04/25/25]seed@VM:~/.../VPN$ docksh 121
root@121110e147fe:/# ip route
default via 10.0.8.11 dev eth0
10.0.8.0/24 dev eth0 proto kernel scope link src 10.0.8.5
root@121110e147fe:/# ip route del default
root@121110e147fe:/# ip route add 10.0.0.0/24 via 10.0.8.11
root@121110e147fe:/# ip route
10.0.0.0/24 via 10.0.8.11 dev eth0
10.0.8.0/24 dev eth0 proto kernel scope link src 10.0.8.5
root@121110e147fe:/#
root@121110e147fe:/# ip route add 10.0.0.0/16 via 10.0.8.11
root@121110e147fe:/# ip route
10.0.0.0/24 via 10.0.8.11 dev eth0
10.0.0.0/16 via 10.0.8.11 dev eth0
10.0.8.0/24 dev eth0 proto kernel scope link src 10.0.8.5
root@121110e147fe:/#
```

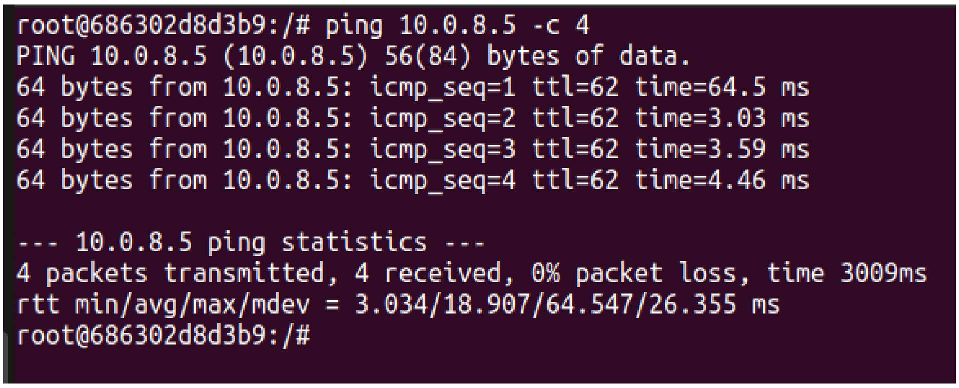
ทดลอง ping 10.0.6.5 -> 10.0.8.5



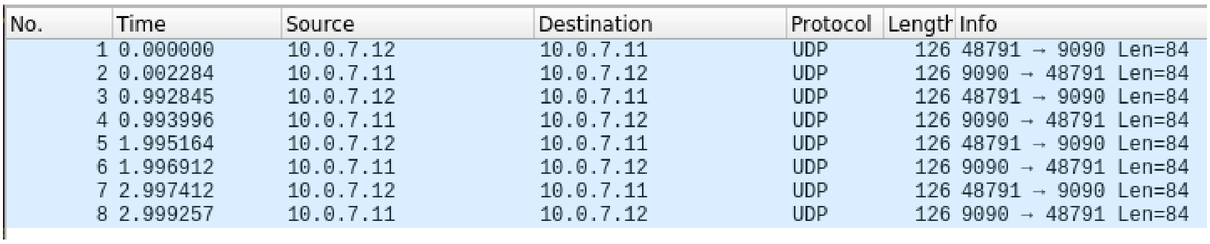
```
root@316cf9290b59:/volumes# tun_server_select2.py
Interface Name: tun0
From socket <==: 10.0.6.5 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.6.5
From socket <==: 10.0.6.5 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.6.5
From socket <==: 10.0.6.5 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.6.5
From socket <==: 10.0.6.5 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.6.5
```



```
root@b36afb46b6ab:/volumes# tun_client_select2.py
Interface Name: tun0
From tun    ==>: 10.0.6.5 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.6.5
From tun    ==>: 10.0.6.5 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.6.5
From tun    ==>: 10.0.6.5 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.6.5
From tun    ==>: 10.0.6.5 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.6.5
```

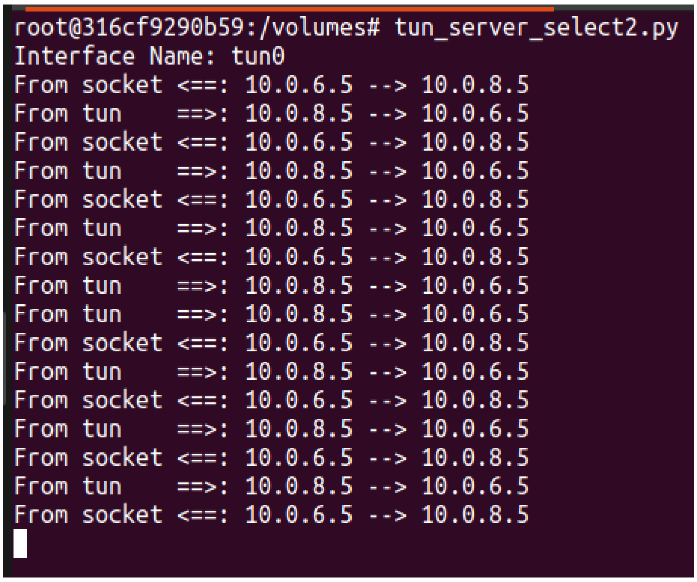


```
root@686302d8d3b9:/# ping 10.0.8.5 -c 4
PING 10.0.8.5 (10.0.8.5) 56(84) bytes of data.
64 bytes from 10.0.8.5: icmp_seq=1 ttl=62 time=64.5 ms
64 bytes from 10.0.8.5: icmp_seq=2 ttl=62 time=3.03 ms
64 bytes from 10.0.8.5: icmp_seq=3 ttl=62 time=3.59 ms
64 bytes from 10.0.8.5: icmp_seq=4 ttl=62 time=4.46 ms

--- 10.0.8.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3009ms
rtt min/avg/max/mdev = 3.034/18.907/64.547/26.355 ms
root@686302d8d3b9:/#
```

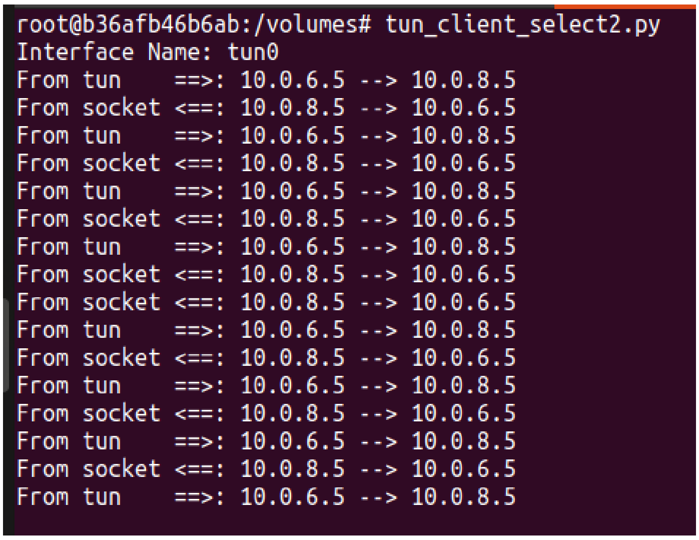


No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.7.12	10.0.7.11	UDP	126	48791 → 9090 Len=84
2	0.002284	10.0.7.11	10.0.7.12	UDP	126	9090 → 48791 Len=84
3	0.992845	10.0.7.12	10.0.7.11	UDP	126	48791 → 9090 Len=84
4	0.993996	10.0.7.11	10.0.7.12	UDP	126	9090 → 48791 Len=84
5	1.995164	10.0.7.12	10.0.7.11	UDP	126	48791 → 9090 Len=84
6	1.996912	10.0.7.11	10.0.7.12	UDP	126	9090 → 48791 Len=84
7	2.997412	10.0.7.12	10.0.7.11	UDP	126	48791 → 9090 Len=84
8	2.999257	10.0.7.11	10.0.7.12	UDP	126	9090 → 48791 Len=84

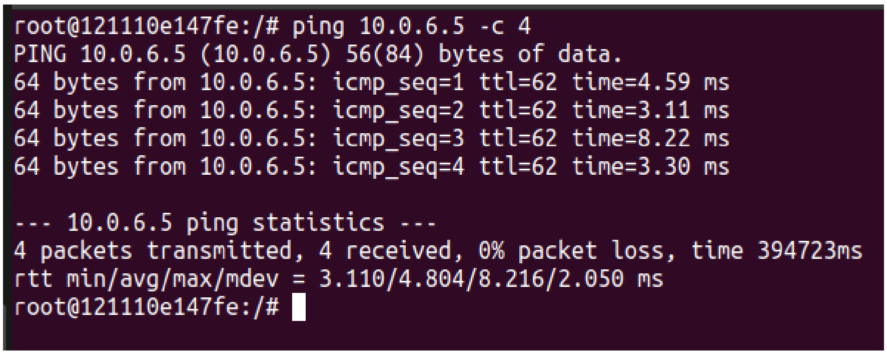
ทดลอง ping 10.0.8.5 -> 10.0.6.5



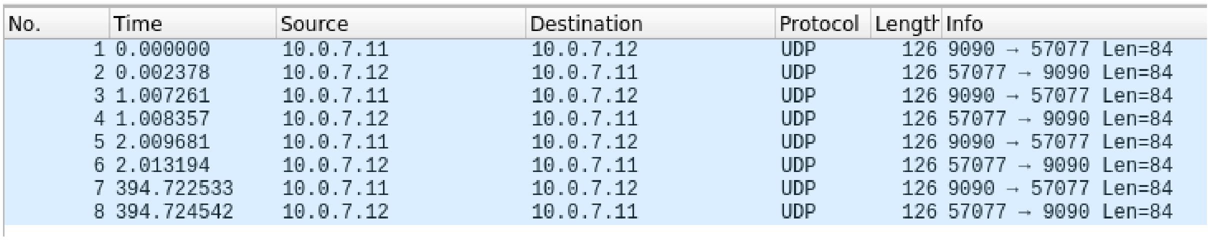
```
root@316cf9290b59:/volumes# tun_server_select2.py
Interface Name: tun0
From socket <==: 10.0.6.5 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.6.5
From socket <==: 10.0.6.5 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.6.5
From socket <==: 10.0.6.5 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.6.5
From socket <==: 10.0.6.5 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.6.5
From tun    ==>: 10.0.8.5 --> 10.0.6.5
From socket <==: 10.0.6.5 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.6.5
From socket <==: 10.0.6.5 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.6.5
From socket <==: 10.0.6.5 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.6.5
From socket <==: 10.0.6.5 --> 10.0.8.5
```



```
root@b36afb46b6ab:/volumes# tun_client_select2.py
Interface Name: tun0
From tun    ==>: 10.0.6.5 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.6.5
From tun    ==>: 10.0.6.5 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.6.5
From tun    ==>: 10.0.6.5 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.6.5
From tun    ==>: 10.0.6.5 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.6.5
From socket <==: 10.0.8.5 --> 10.0.6.5
From tun    ==>: 10.0.6.5 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.6.5
From tun    ==>: 10.0.6.5 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.6.5
From tun    ==>: 10.0.6.5 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.6.5
From tun    ==>: 10.0.6.5 --> 10.0.8.5
```



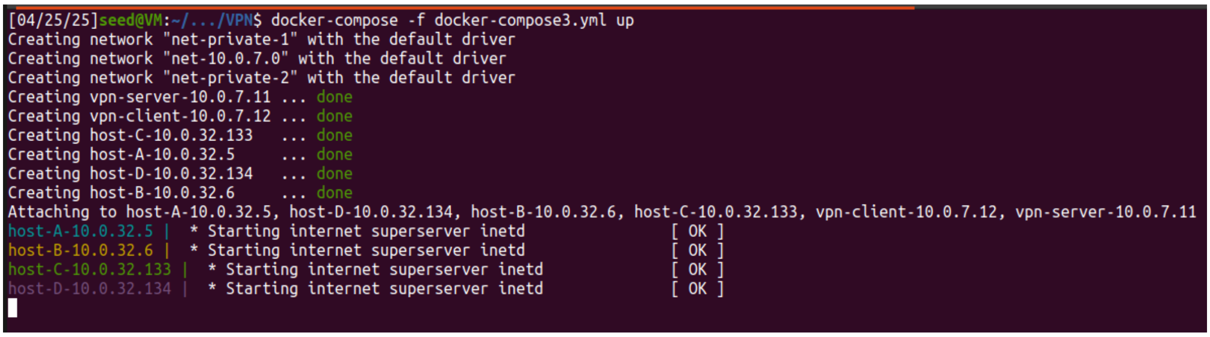
```
root@121110e147fe:/# ping 10.0.6.5 -c 4
PING 10.0.6.5 (10.0.6.5) 56(84) bytes of data.
64 bytes from 10.0.6.5: icmp_seq=1 ttl=62 time=4.59 ms
64 bytes from 10.0.6.5: icmp_seq=2 ttl=62 time=3.11 ms
64 bytes from 10.0.6.5: icmp_seq=3 ttl=62 time=8.22 ms
64 bytes from 10.0.6.5: icmp_seq=4 ttl=62 time=3.30 ms

--- 10.0.6.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 394723ms
rtt min/avg/max/mdev = 3.110/4.804/8.216/2.050 ms
root@121110e147fe:/#
```



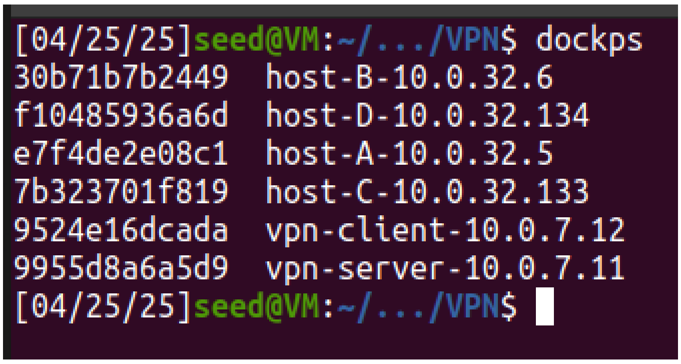
No.	Time	Source	Destination	Protocol	Lengtł	Info
1	0.000000	10.0.7.11	10.0.7.12	UDP	126	9090 → 57077 Len=84
2	0.002378	10.0.7.12	10.0.7.11	UDP	126	57077 → 9090 Len=84
3	1.007261	10.0.7.11	10.0.7.12	UDP	126	9090 → 57077 Len=84
4	1.008357	10.0.7.12	10.0.7.11	UDP	126	57077 → 9090 Len=84
5	2.009681	10.0.7.11	10.0.7.12	UDP	126	9090 → 57077 Len=84
6	2.013194	10.0.7.12	10.0.7.11	UDP	126	57077 → 9090 Len=84
7	394.722533	10.0.7.11	10.0.7.12	UDP	126	9090 → 57077 Len=84
8	394.724542	10.0.7.12	10.0.7.11	UDP	126	57077 → 9090 Len=84

Task 9: Experiment with the TAP interface

ใช้ container เป็น docker-compose3.yml



```
[04/25/25]seed@VM:~/.../VPN$ docker-compose -f docker-compose3.yml up
Creating network "net-private-1" with the default driver
Creating network "net-10.0.7.0" with the default driver
Creating network "net-private-2" with the default driver
Creating vpn-server-10.0.7.11 ... done
Creating vpn-client-10.0.7.12 ... done
Creating host-C-10.0.32.133   ... done
Creating host-A-10.0.32.5     ... done
Creating host-D-10.0.32.134   ... done
Creating host-B-10.0.32.6     ... done
Attaching to host-A-10.0.32.5, host-D-10.0.32.134, host-B-10.0.32.6, host-C-10.0.32.133, vpn-client-10.0.7.12, vpn-server-10.0.7.11
host-A-10.0.32.5 |  * Starting internet superserver inetd               [ OK ]
host-B-10.0.32.6 |  * Starting internet superserver inetd               [ OK ]
host-C-10.0.32.133 |  * Starting internet superserver inetd             [ OK ]
host-D-10.0.32.134 |  * Starting internet superserver inetd             [ OK ]
```



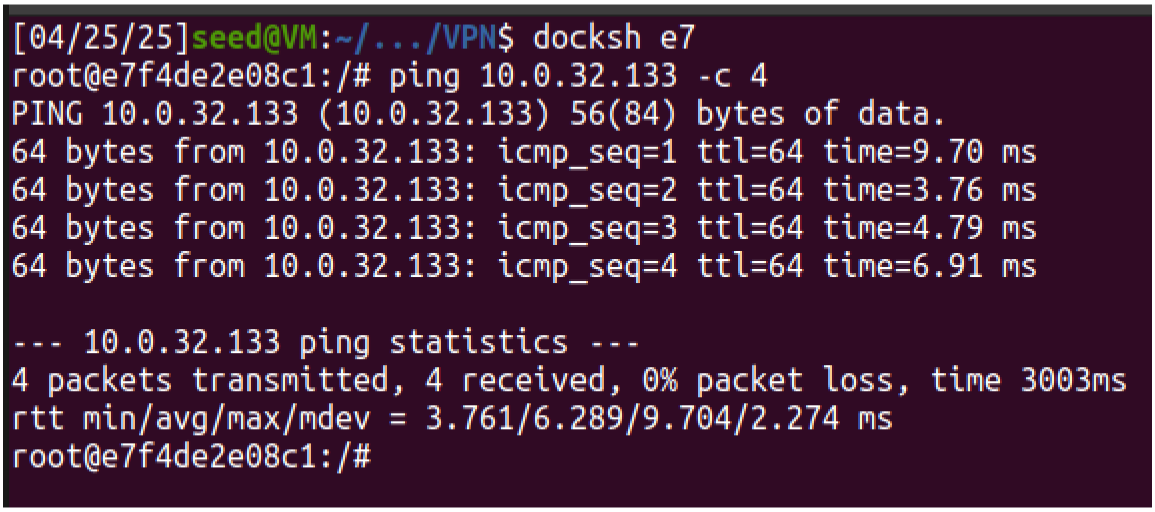
```
[04/25/25]seed@VM:~/.../VPN$ dockps
30b71b7b2449  host-B-10.0.32.6
f10485936a6d  host-D-10.0.32.134
e7f4de2e08c1  host-A-10.0.32.5
7b323701f819  host-C-10.0.32.133
9524e16dcada  vpn-client-10.0.7.12
9955d8a6a5d9  vpn-server-10.0.7.11
[04/25/25]seed@VM:~/.../VPN$
```

ใช้ TAP interface (MAC layer)

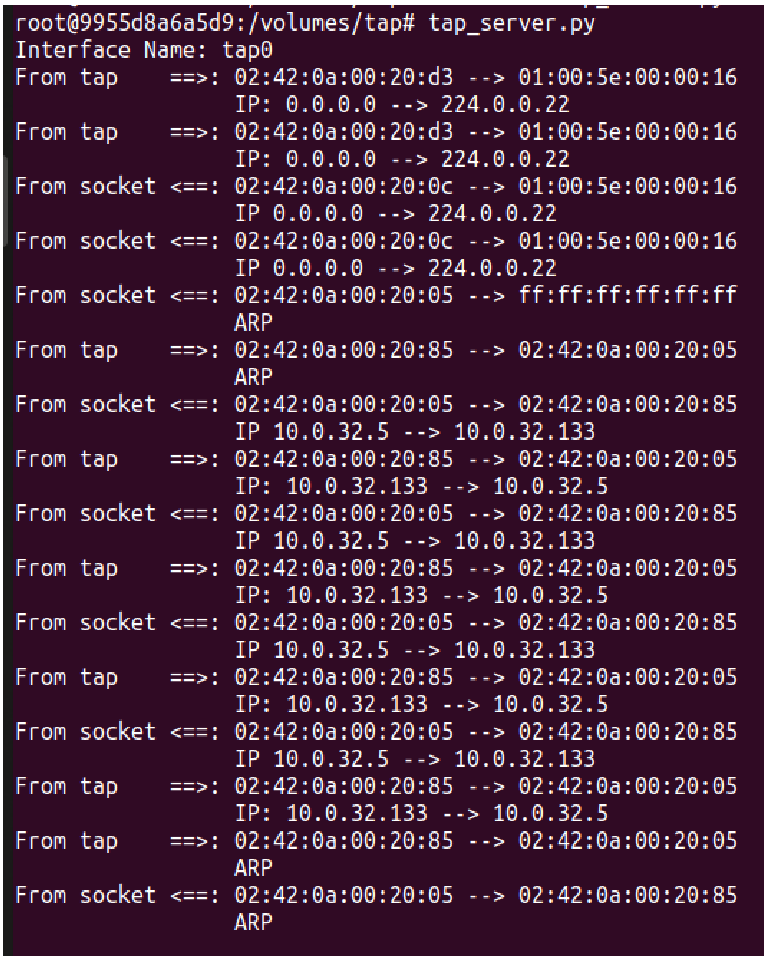
vpn-server 10.0.7.11 ใช้ code tap_server.py

vpn-client 10.0.7.12 ใช้ code tap_client.py

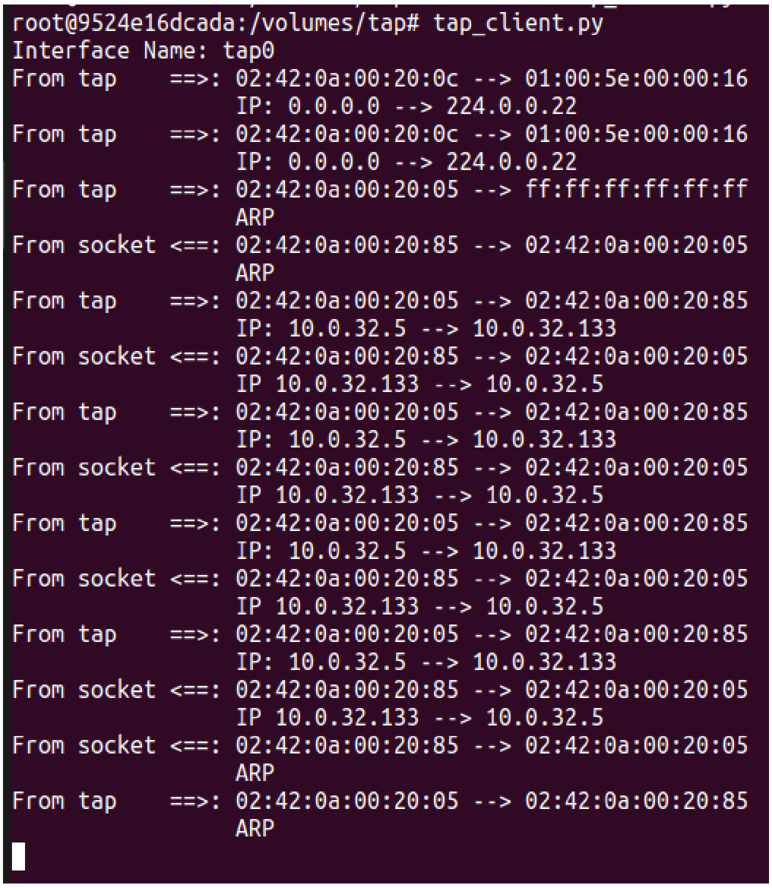
ทดลอง ping 10.0.32.5 -> 10.0.32.133



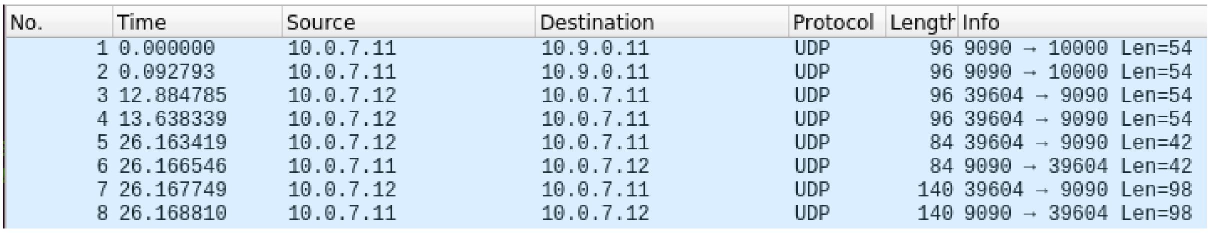
```
[04/25/25]seed@VM:~/.../VPN$ docksh e7
root@e7f4de2e08c1:/# ping 10.0.32.133 -c 4
PING 10.0.32.133 (10.0.32.133) 56(84) bytes of data.
64 bytes from 10.0.32.133: icmp_seq=1 ttl=64 time=9.70 ms
64 bytes from 10.0.32.133: icmp_seq=2 ttl=64 time=3.76 ms
64 bytes from 10.0.32.133: icmp_seq=3 ttl=64 time=4.79 ms
64 bytes from 10.0.32.133: icmp_seq=4 ttl=64 time=6.91 ms

--- 10.0.32.133 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 3.761/6.289/9.704/2.274 ms
root@e7f4de2e08c1:/#
```



```
root@9955d8a6a5d9:/volumes/tap# tap_server.py
Interface Name: tap0
From tap    ==>: 02:42:0a:00:20:d3 --> 01:00:5e:00:00:16
                 IP: 0.0.0.0 --> 224.0.0.22
From tap    ==>: 02:42:0a:00:20:d3 --> 01:00:5e:00:00:16
                 IP: 0.0.0.0 --> 224.0.0.22
From socket <==: 02:42:0a:00:20:0c --> 01:00:5e:00:00:16
                 IP 0.0.0.0 --> 224.0.0.22
From socket <==: 02:42:0a:00:20:0c --> 01:00:5e:00:00:16
                 IP 0.0.0.0 --> 224.0.0.22
From socket <==: 02:42:0a:00:20:05 --> ff:ff:ff:ff:ff:ff
                 ARP
From tap    ==>: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 ARP
From socket <==: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP 10.0.32.5 --> 10.0.32.133
From tap    ==>: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP: 10.0.32.133 --> 10.0.32.5
From socket <==: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP 10.0.32.5 --> 10.0.32.133
From tap    ==>: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP: 10.0.32.133 --> 10.0.32.5
From socket <==: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP 10.0.32.5 --> 10.0.32.133
From tap    ==>: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP: 10.0.32.133 --> 10.0.32.5
From socket <==: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP 10.0.32.5 --> 10.0.32.133
From tap    ==>: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP: 10.0.32.133 --> 10.0.32.5
From tap    ==>: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 ARP
From socket <==: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 ARP
```

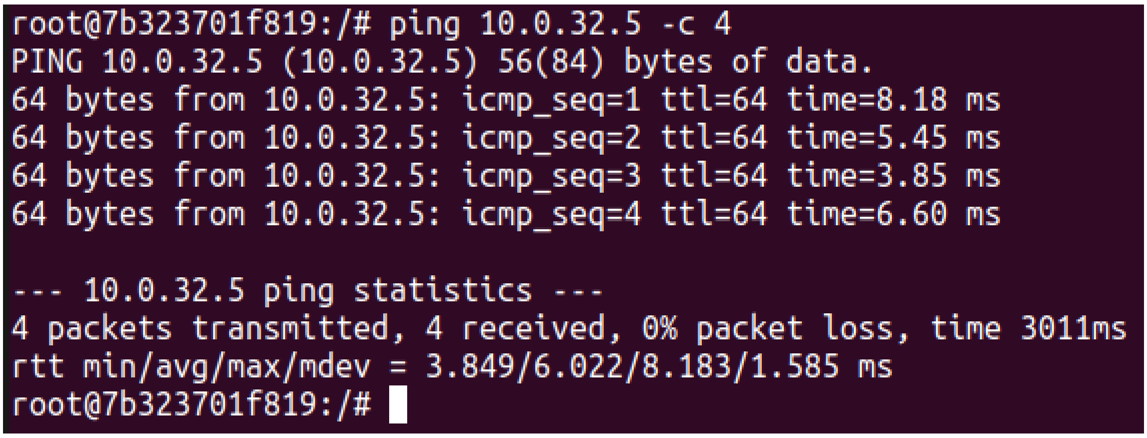


```
root@9524e16dcada:/volumes/tap# tap_client.py
Interface Name: tap0
From tap    ==>: 02:42:0a:00:20:0c --> 01:00:5e:00:00:16
                 IP: 0.0.0.0 --> 224.0.0.22
From tap    ==>: 02:42:0a:00:20:0c --> 01:00:5e:00:00:16
                 IP: 0.0.0.0 --> 224.0.0.22
From tap    ==>: 02:42:0a:00:20:05 --> ff:ff:ff:ff:ff:ff
                 ARP
From socket <==: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 ARP
From tap    ==>: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP: 10.0.32.5 --> 10.0.32.133
From socket <==: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP 10.0.32.133 --> 10.0.32.5
From tap    ==>: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP: 10.0.32.5 --> 10.0.32.133
From socket <==: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP 10.0.32.133 --> 10.0.32.5
From tap    ==>: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP: 10.0.32.5 --> 10.0.32.133
From socket <==: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP 10.0.32.133 --> 10.0.32.5
From tap    ==>: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP: 10.0.32.5 --> 10.0.32.133
From socket <==: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP 10.0.32.133 --> 10.0.32.5
From socket <==: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 ARP
From tap    ==>: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 ARP
```

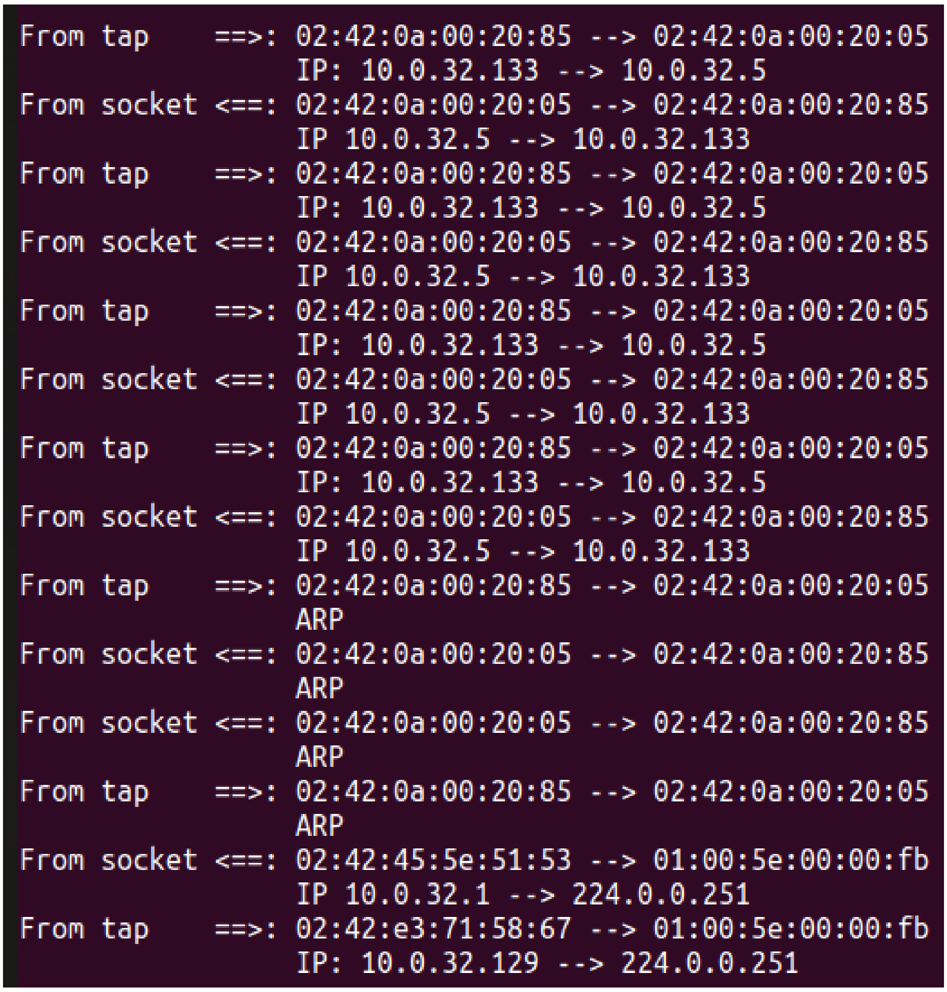


No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.7.11	10.9.0.11	UDP	96	9090 → 10000 Len=54
2	0.092793	10.0.7.11	10.9.0.11	UDP	96	9090 → 10000 Len=54
3	12.884785	10.0.7.12	10.0.7.11	UDP	96	39604 → 9090 Len=54
4	13.638339	10.0.7.12	10.0.7.11	UDP	96	39604 → 9090 Len=54
5	26.163419	10.0.7.12	10.0.7.11	UDP	84	39604 → 9090 Len=42
6	26.166546	10.0.7.11	10.0.7.12	UDP	84	9090 → 39604 Len=42
7	26.167749	10.0.7.12	10.0.7.11	UDP	140	39604 → 9090 Len=98
8	26.168810	10.0.7.11	10.0.7.12	UDP	140	9090 → 39604 Len=98

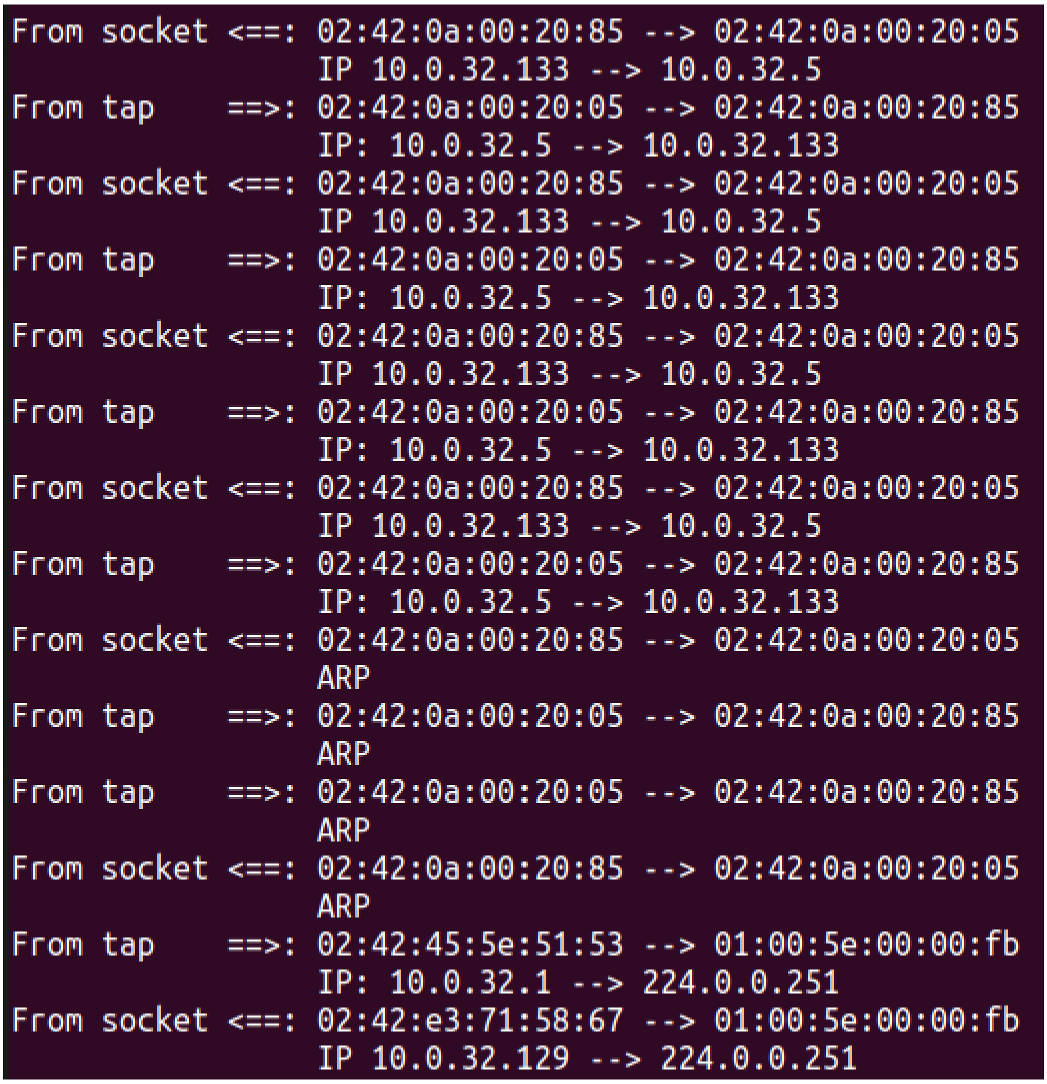
ลอง ping 10.0.32.133 -> 10.0.32.5



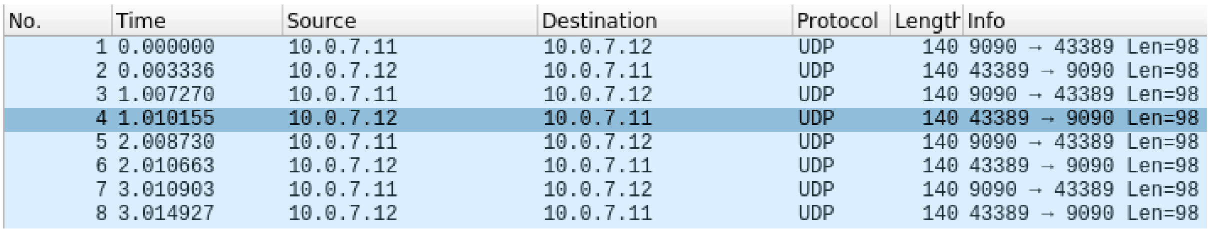
```
root@7b323701f819:/# ping 10.0.32.5 -c 4
PING 10.0.32.5 (10.0.32.5) 56(84) bytes of data.
64 bytes from 10.0.32.5: icmp_seq=1 ttl=64 time=8.18 ms
64 bytes from 10.0.32.5: icmp_seq=2 ttl=64 time=5.45 ms
64 bytes from 10.0.32.5: icmp_seq=3 ttl=64 time=3.85 ms
64 bytes from 10.0.32.5: icmp_seq=4 ttl=64 time=6.60 ms

--- 10.0.32.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3011ms
rtt min/avg/max/mdev = 3.849/6.022/8.183/1.585 ms
root@7b323701f819:/# 
```



```
From tap    ==>: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP: 10.0.32.133 --> 10.0.32.5
From socket <==: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP 10.0.32.5 --> 10.0.32.133
From tap    ==>: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP: 10.0.32.133 --> 10.0.32.5
From socket <==: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP 10.0.32.5 --> 10.0.32.133
From tap    ==>: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP: 10.0.32.133 --> 10.0.32.5
From socket <==: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP 10.0.32.5 --> 10.0.32.133
From tap    ==>: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP: 10.0.32.133 --> 10.0.32.5
From socket <==: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP 10.0.32.5 --> 10.0.32.133
From tap    ==>: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 ARP
From socket <==: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 ARP
From socket <==: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 ARP
From tap    ==>: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 ARP
From socket <==: 02:42:45:5e:51:53 --> 01:00:5e:00:00:fb
                 IP 10.0.32.1 --> 224.0.0.251
From tap    ==>: 02:42:e3:71:58:67 --> 01:00:5e:00:00:fb
                 IP: 10.0.32.129 --> 224.0.0.251
```



```
From socket <==: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP 10.0.32.133 --> 10.0.32.5
From tap    ==>: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP: 10.0.32.5 --> 10.0.32.133
From socket <==: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP 10.0.32.133 --> 10.0.32.5
From tap    ==>: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP: 10.0.32.5 --> 10.0.32.133
From socket <==: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP 10.0.32.133 --> 10.0.32.5
From tap    ==>: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP: 10.0.32.5 --> 10.0.32.133
From socket <==: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP 10.0.32.133 --> 10.0.32.5
From tap    ==>: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP: 10.0.32.5 --> 10.0.32.133
From socket <==: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 ARP
From tap    ==>: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 ARP
From tap    ==>: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 ARP
From socket <==: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 ARP
From tap    ==>: 02:42:45:5e:51:53 --> 01:00:5e:00:00:fb
                 IP: 10.0.32.1 --> 224.0.0.251
From socket <==: 02:42:e3:71:58:67 --> 01:00:5e:00:00:fb
                 IP 10.0.32.129 --> 224.0.0.251
```



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.7.11	10.0.7.12	UDP	140	9090 → 43389 Len=98
2	0.003336	10.0.7.12	10.0.7.11	UDP	140	43389 → 9090 Len=98
3	1.007270	10.0.7.11	10.0.7.12	UDP	140	9090 → 43389 Len=98
4	1.010155	10.0.7.12	10.0.7.11	UDP	140	43389 → 9090 Len=98
5	2.008730	10.0.7.11	10.0.7.12	UDP	140	9090 → 43389 Len=98
6	2.010663	10.0.7.12	10.0.7.11	UDP	140	43389 → 9090 Len=98
7	3.010903	10.0.7.11	10.0.7.12	UDP	140	9090 → 43389 Len=98
8	3.014927	10.0.7.12	10.0.7.11	UDP	140	43389 → 9090 Len=98