NAT- Commands

# Create namespaces for client, router, and public internet

sudo ip netns add client

sudo ip netns add router

sudo ip netns add public

# Create veth pair between client and router

sudo ip link add veth-c type veth peer name veth-r1

# Create veth pair between router and public with shorter names

sudo ip link add veth-r2 type veth peer name veth-pub

# Assign veth-c to client namespace

sudo ip link set veth-c netns client

# Assign veth-r1 to router namespace

sudo ip link set veth-r1 netns router

# Assign veth-r2 to router namespace

sudo ip link set veth-r2 netns router

# Assign veth-pub to public namespace

sudo ip link set veth-pub netns public

# In client namespace

sudo ip netns exec client ip addr add 192.168.10.2/24 dev veth-c

sudo ip netns exec client ip link set dev veth-c up

# In router namespace (LAN side)

sudo ip netns exec router ip addr add 192.168.10.1/24 dev veth-r1

sudo ip netns exec router ip link set dev veth-r1 up

# In router namespace (Public side)

sudo ip netns exec router ip addr add 10.0.0.1/24 dev veth-r2

sudo ip netns exec router ip link set dev veth-r2 up

# In public namespace

sudo ip netns exec public ip addr add 10.0.0.2/24 dev veth-pub

sudo ip netns exec public ip link set dev veth-pub up

# Enabling IP Forwarding on Router

sudo ip netns exec router sysctl -w net.ipv4.ip\_forward=1



# Apply masquerading for packets going out to the public network

sudo ip netns exec router iptables -t nat -A POSTROUTING -o veth-r2 -j MASQUERADE

# In client namespace, set default route via the router's LAN interface

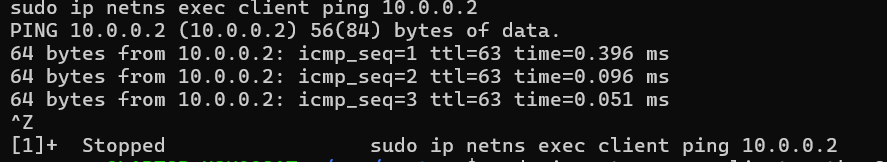
sudo ip netns exec client ip route add default via 192.168.10.1

# In public namespace, set default route via the router's public interface

sudo ip netns exec public ip route add default via 10.0.0.1

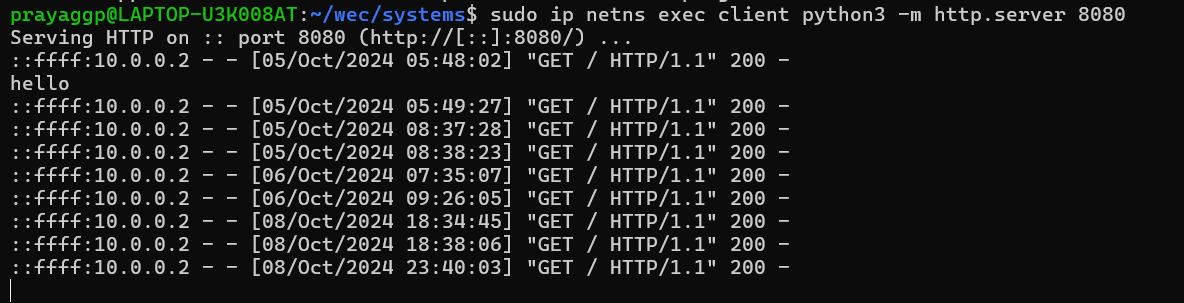
# Test ping from client to public network

sudo ip netns exec client ping 10.0.0.2



# Starting a basic HTTP server in client namespace

sudo ip netns exec client python3 -m http.server 8080



# Forward requests to the internal web server running on client (port 8080)

sudo ip netns exec router iptables -t nat -A PREROUTING -i veth-r2 -p tcp --dport 8080 -j DNAT --to-destination 192.168.10.2:8080

# Accept the forwarded packets

sudo ip netns exec router iptables -A FORWARD -i veth-r2 -o veth-r1 -p tcp --dport 8080 -j ACCEPT

# Access the web server from public namespace

sudo ip netns exec public curl http://10.0.0.1:8080

