

Advanced Linux Network Lab – Step-by-Step Manual

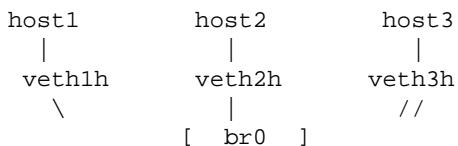
This document converts all lab steps into a structured, printable PDF. It is suitable for classroom labs, self-study, and evaluations.

Part 1: L2 Switching Using Linux Bridge

Objective

Create a Layer-2 network using a Linux bridge and connect three hosts using network namespaces and veth pairs.

Topology (ASCII)



Step 1: Create Network Namespaces

```
sudo ip netns add host1
sudo ip netns add host2
sudo ip netns add host3

sudo ip netns exec host1 ip link set lo up
sudo ip netns exec host2 ip link set lo up
sudo ip netns exec host3 ip link set lo up
```

Step 2: Create Bridge

```
sudo ip link add br0 type bridge
sudo ip link set br0 up
```

Step 3: Connect Hosts to Bridge

```
sudo ip link add veth1 type veth peer name veth1h
sudo ip link set veth1h netns host1
sudo ip link set veth1 master br0
sudo ip link set veth1 up
sudo ip netns exec host1 ip link set veth1h up

sudo ip link add veth2 type veth peer name veth2h
sudo ip link set veth2h netns host2
sudo ip link set veth2 master br0
sudo ip link set veth2 up
sudo ip netns exec host2 ip link set veth2h up

sudo ip link add veth3 type veth peer name veth3h
sudo ip link set veth3h netns host3
sudo ip link set veth3 master br0
sudo ip link set veth3 up
sudo ip netns exec host3 ip link set veth3h up
```

Step 4: Assign IP Addresses

```
sudo ip netns exec host1 ip addr add 192.168.10.11/24 dev veth1h
```

```
sudo ip netns exec host2 ip addr add 192.168.10.12/24 dev veth2h  
sudo ip netns exec host3 ip addr add 192.168.10.13/24 dev veth3h
```

Step 5: Test Connectivity

```
sudo ip netns exec host1 ping -c 3 192.168.10.12  
sudo ip netns exec host2 ping -c 3 192.168.10.13
```

Part 2: Router with NAT

Objective

Insert a router namespace between LAN and WAN and enable NAT for outbound traffic.

Topology (ASCII)

```
LAN (192.168.10.0/24)
  host1 host2 host3
    |
    [br0]
    |
192.168.10.1
  [ router ]
  203.0.113.1
  |
  [ WAN ]
  203.0.113.2
```

Step 1: Create Router and WAN Namespaces

```
sudo ip netns add router
sudo ip netns add wan

sudo ip netns exec router ip link set lo up
sudo ip netns exec wan ip link set lo up
```

Step 2: Connect Router to LAN

```
sudo ip link add vethr-br type veth peer name vethr
sudo ip link set vethr netns router
sudo ip link set vethr-br master br0
sudo ip link set vethr-br up

sudo ip netns exec router ip link set vethr up
sudo ip netns exec router ip addr add 192.168.10.1/24 dev vethr
```

Step 3: Connect Router to WAN

```
sudo ip link add veth-wan type veth peer name veth-wanh
sudo ip link set veth-wan netns router
sudo ip link set veth-wanh netns wan

sudo ip netns exec router ip link set veth-wan up
sudo ip netns exec wan ip link set veth-wanh up

sudo ip netns exec router ip addr add 203.0.113.1/24 dev veth-wan
sudo ip netns exec wan ip addr add 203.0.113.2/24 dev veth-wanh
```

Step 4: Enable Routing and NAT

```
sudo ip netns exec router sysctl -w net.ipv4.ip_forward=1
sudo ip netns exec router iptables -t nat -A POSTROUTING -o veth-wan -j MASQUERADE
```

Step 5: Verify Connectivity

```
sudo ip netns exec host1 ping -c 3 203.0.113.2
```


Part 3: Policy-Based Routing

Topology (ASCII)

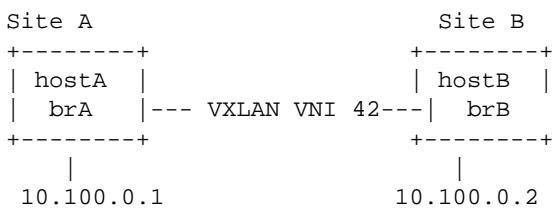
```
host1 ---> WAN1 (table 100)
host2 ---> WAN2 (table 200)
```

Steps

```
sudo ip netns exec router ip rule add from 192.168.10.11 table 100
sudo ip netns exec router ip rule add from 192.168.10.12 table 200

sudo ip netns exec router ip route add default via 203.0.113.2 table 100
sudo ip netns exec router ip route add default via 198.51.100.2 table 200
```

Part 4: VXLAN Overlay (Conceptual Steps)



Create VXLAN interfaces on both sites with the same VNI and attach them to their respective bridges to extend Layer-2 connectivity over an IP underlay.

Cleanup

```
sudo ip netns del host1
sudo ip netns del host2
sudo ip netns del host3
sudo ip netns del router
sudo ip netns del wan
sudo ip link del br0
```