

Solutions — Lesson 03

Worked answers for the homework tasks

Try the homework yourself first. PBR makes much more sense when you've seen it not work and figured out why.

Task 1 — Source-based routing

```
# 1. Namespaces
for ns in client-a client-b shared gw-a gw-b; do sudo ip netns add $ns; done

# 2. Veth pairs (4 pairs: each client↔shared, shared↔each gw)
sudo ip link add veth-ca type veth peer name veth-sa
sudo ip link add veth-cb type veth peer name veth-sb
sudo ip link add veth-sga type veth peer name veth-gua
sudo ip link add veth-sgb type veth peer name veth-gub

# Move ends into namespaces
sudo ip link set veth-ca netns client-a
sudo ip link set veth-cb netns client-b
sudo ip link set veth-sa netns shared
sudo ip link set veth-sb netns shared
sudo ip link set veth-sga netns shared
sudo ip link set veth-sgb netns shared
sudo ip link set veth-gua netns gw-a
sudo ip link set veth-gub netns gw-b

# 3. IPs and link up
sudo ip -n client-a addr add 10.10.1.5/24 dev veth-ca
sudo ip -n shared addr add 10.10.1.1/24 dev veth-sa
sudo ip -n client-b addr add 10.10.2.5/24 dev veth-cb
sudo ip -n shared addr add 10.10.2.1/24 dev veth-sb
sudo ip -n shared addr add 10.99.0.2/24 dev veth-sga
sudo ip -n gw-a addr add 10.99.0.1/24 dev veth-gua
```

Task 2 — fwmark-based routing

```
sudo iptables -t mangle -A OUTPUT -p tcp --dport 22 -j MARK --set-mark 0x42
sudo ip rule add pref 90 fwmark 0x42 lookup 300
sudo ip route add default via 10.99.1.1 dev eth0 table 300

ip rule show          # pref 90 fwmark 0x42 lookup 300 present
ip route show table 300
ip route get 8.8.8.8 mark 0x42  # → via 10.99.1.1
```

Task 3 — CONNMARK persistence

```
# Order matters: restore BEFORE the conditional --set-mark.
# Otherwise the first packet works (NEW state matches), but you've
# already overwritten the mark before save can capture it.
sudo iptables -t mangle -A PREROUTING -j CONNMARK --restore-mark
sudo iptables -t mangle -A PREROUTING -m conntrack --ctstate NEW -j MARK --set-mark
sudo iptables -t mangle -A PREROUTING -j CONNMARK --save-mark

# Long-lived flow (run in two terminals or two namespaces):
#   server: nc -lk 9999
#   client: nc 10.x.x.x 9999

# Inspect:
sudo conntrack -L | grep mark
# Expected output line includes "mark=66" (0x42 = 66 decimal)
```

Why the ordering matters

- **restore first:** if a non-NEW packet arrives with no mark, this pulls the mark from conntrack so routing sees it.
- **set-mark second (NEW only):** NEW packets (no conntrack entry yet) get the initial mark.
- **save last:** stamps the now-set mark onto the conntrack entry so future packets can restore from it.

Common mistake: putting `--save-mark` before `--set-mark` saves a zero mark to conntrack, then every "restore" pulls back zero. Symptom: only the first packet

of every new flow takes the policy path.

Common mistakes across all three tasks

- **Priority collisions.** Two `ip rule` entries with the same `pref` are valid but undefined-ordered. Use distinct priorities.
- **Forgot the return route.** Building three-namespace topologies, both directions need routes. Symptom: tcpdump shows only one direction.
- **Forgot forwarding on the router namespace.** `net.ipv4.ip_forward=1` must be set inside the router namespace, not on the host.
- **rp_filter dropping replies.** If you use PBR with non-default interfaces, set `rp_filter=2` (loose mode) on the relevant interfaces.
- **Flushing the wrong table.** `ip route flush table main` is catastrophic. Always specify the non-main table number explicitly.