DETAILED FULL-TUNNEL OPENVPN CONFIGURATION GUIDE

Objective: Configure a full-tunnel OpenVPN deployment where remote clients connect to the public IP (y.y.y.2) and route all internet traffic securely through the VPN.

1. NETWORK TOPOLOGY

```
Modem:
  LAN IP: y.y.y.1/24
  Gateway: ISP

OPNsense Firewall:
  WAN: y.y.y.2/24
  Gateway: y.y.y.1
  LAN: 192.168.10.1/24

OpenVPN Server:
  LAN IP: 192.168.10.10/24
  Default Gateway: 192.168.10.1

VPN Tunnel Network:
  10.8.0.0/24
```

2. MODEM CONFIGURATION

Set modem to router mode (not bridge). Ensure WAN connectivity is functional before proceeding.

2.1 Port Forwarding

```
Protocol: UDP
External Port: 1194
Internal IP: y.y.y.2
Internal Port: 1194
```

Ensure inbound UDP 1194 is permitted in modem firewall settings.

3. OPNSENSE CONFIGURATION

3.1 Interface Configuration

```
WAN Interface:
  IPv4: y.y.y.2/24
  Gateway: y.y.y.1

LAN Interface:
  IPv4: 192.168.10.1/24
```

3.2 Outbound NAT Configuration

Navigate to Firewall > NAT > Outbound and select Hybrid Outbound NAT mode.

```
Add Rule:
  Interface: WAN
  Source: 10.8.0.0/24
  Translation / Target: Interface Address
```

3.3 Firewall Rules

```
WAN Rule:
  Action: Pass
  Protocol: UDP
  Source: any
  Destination: WAN Address
  Destination Port: 1194

OpenVPN Interface Rule:
  Action: Pass
  Source: 10.8.0.0/24
  Destination: any

LAN Rule:
  Allow LAN net to any (default rule)
```

No static routes are required in this topology.

4. OPENVPN SERVER CONFIGURATION

4.1 Enable IP Forwarding (Linux)

```
Temporary:
echo 1 > /proc/sys/net/ipv4/ip_forward

Permanent:
Add to /etc/sysctl.conf:
net.ipv4.ip_forward=1
```

4.2 OpenVPN Server Configuration File (server.conf)

```
port 1194
proto udp
dev tun

server 10.8.0.0 255.255.255.0
topology subnet

push "redirect-gateway def1"
push "dhcp-option DNS 1.1.1.1"
push "dhcp-option DNS 8.8.8.8"

keepalive 10 120
persist-key
persist-tun

cipher AES-256-GCM
```

```
auth SHA256

user nobody
group nogroup

tls-server
ca ca.crt
cert server.crt
key server.key
dh dh.pem
```

4.3 Firewall Rules on OpenVPN Server (iptables example)

```
iptables -A FORWARD -i tun0 -o eth0 -j ACCEPT
iptables -A FORWARD -i eth0 -o tun0 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

5. TRAFFIC FLOW VERIFICATION

```
Client → VPN Tunnel (10.8.0.x)
       → OpenVPN Server (192.168.10.10)
       → OPNsense LAN (192.168.10.1)
       → OPNsense WAN (y.y.y.2)
       → Internet
```

Verification steps:

```
1. Connect client to VPN.
2. Run: curl ifconfig.me
3. Confirm public IP = y.y.y.2
4. Verify DNS resolution uses pushed DNS servers.
5. Confirm default route points to VPN interface.
```

Result: All client web traffic exits via OPNsense WAN interface (y.y.y.2). Split tunneling is disabled.