IP-Transit

- Wiregaurd proxmox setup (ip-transit)
- vxlan proxmox setup (ip-transit)
- gre proxmox setup (ip-transit)
- GRE tunnel on MikroTik router

Wiregaurd proxmox setup (ip-transit)

? Step 1: Install WireGuard

```
apt update && apt install wireguard -y
```

? Step 2: Create the WireGuard config

Paste the config that we sent you into `/etc/wireguard/wg0.conf`

? Step 3: Enable and start WireGuard

```
systemctl enable --now wg-quick@wg0
```

? Step 4: Add routing rules for public IPs

You can run this directly or save as `/root/wg-routes.sh`:

```
#!/bin/bash

for ip in {113..117}; do
  ip rule add from 5.231.32.$ip/32 table 11 prio 1
  ip route add 5.231.32.$ip/32 dev vmbr0 table 11
  ip route add 5.231.32.$ip/32 dev vmbr0
done
```

Make it executable:

```
chmod +x /root/wg-routes.sh
```

Then run it:

```
bash /root/wg-routes.sh
```

? Step 5: Enable IP forwarding and proxy ARP/NDP

You can paste this all at once:

```
echo 1 > /proc/sys/net/ipv4/conf/all/proxy_arp
echo 1 > /proc/sys/net/ipv6/conf/default/proxy_ndp
echo "net.ipv4.ip_forward=1" >> /etc/sysctl.conf
echo "net.ipv6.conf.all.forwarding=1" >> /etc/sysctl.conf
sysctl -p
```

Or add it into the same script above so it runs after reboot.

? Done!

Your public IPs should now route through the WireGuard tunnel.

vxlan proxmox setup (ip-transit)

Proxmox VXLAN Tunnel Setup

Requirements

Before starting, make sure you have:

- A working Proxmox installation
- Root access to your node
- The remote endpoint IP
- Basic networking knowledge

Misconfiguration may result in **loss of network connectivity**.

VXLAN allows you to create a **Layer-2 overlay network over Layer-3 infrastructure**, commonly used for connecting multiple hosts or data centers.

Tunnel Configuration

The configuration is done inside:

```
/etc/network/interfaces
```

Insert the VXLAN configuration template below and adjust the variables.

Variables appear inside `< >` and must be replaced with your actual values.

Variables

Variable	Description
<VXLAN_ID>	VXLAN Network Identifier (VNI), must match on both endpoints
<Local Endpoint>	Local server public IP

Variable	Description
<Remote Endpoint>	Remote server public IP
<VXLAN Interface>	VXLAN device name (example: vxlan10)
<Bridge Name>	Proxmox bridge name (example: vmbr10)
<VXLAN Port>	UDP port used by VXLAN (default: 4789)
<Physical Interface>	Network interface used for transport (example: eno1)

VXLAN Configuration Template

```
auto vxlan10
iface vxlan10 inet manual
    mtu 1450
    pre-up ip link add vxlan10 type vxlan \
        id <VXLAN_ID> \
        dev <Physical Interface> \
        remote <Remote Endpoint> \
        local <Local Endpoint> \
        dstport <VXLAN Port>
    post-down ip link del vxlan10
```

Bridge Configuration

Create a bridge so VMs and containers can use the VXLAN network.

```
auto vmbr10
iface vmbr10 inet manual
    mtu 1450
    bridge_ports vxlan10
    bridge_stp off
    bridge_fd 0
```

Example Configuration

Example setup:

- Local node: `192.0.2.10`
- Remote node: `192.0.2.20`
- VXLAN ID: `100`
- Interface: `eno1`

```
auto vxlan10
iface vxlan10 inet manual
    mtu 1450
    pre-up ip link add vxlan10 type vxlan id 100 dev eno1 \
        remote 192.0.2.20 local 192.0.2.10 dstport 4789
    post-down ip link del vxlan10

auto vmbr10
iface vmbr10 inet manual
    mtu 1450
    bridge_ports vxlan10
    bridge_stp off
    bridge_fd 0
```

Apply Configuration

Restart networking or bring interfaces up manually.

```
ifdown vmbr10 2>/dev/null
ifdown vxlan10 2>/dev/null
ifup vxlan10
ifup vmbr10
```

Using VXLAN in Proxmox VMs

1. Open VM settings
2. Select **Network Device**
3. Choose Bridge: `vmbr10`

Your VM will now be connected to the **VXLAN overlay network**.

Troubleshooting

Check VXLAN interface

```
ip link show vxlan10
```

Check bridge

```
bridge link
```

Verify connectivity

```
ping <remote-vxlan-ip>
```

Capture VXLAN packets

```
tcpdump -i <physical-interface> udp port 4789
```

Notes

- Default VXLAN port is **4789**
- VXLAN adds **~50 bytes overhead**, so MTU should typically be **1450**
- Firewalls must allow **UDP 4789**

gre proxmox setup (ip-transit)

```
sudo ip link add gre1 type gre local Server-B-IPv4 remote Server-A-IPv4 key yourgrekey ttl 255
sudo ip addr add 10.10.0.2/16 dev gre1
sudo ip link set gre1 up
```

```
#!/bin/bash

for ip in {113..117}; do
  ip rule add from 5.231.32.$ip/32 table 10 prio 1
  ip route add 5.231.32.$ip/32 dev vmbr0 table 10
  ip route add 5.231.32.$ip/32 dev vmbr0
done
```

```
echo 1 > /proc/sys/net/ipv4/conf/all/proxy_arp
echo 1 > /proc/sys/net/ipv6/conf/default/proxy_ndp
echo "net.ipv4.ip_forward=1" >> /etc/sysctl.conf
echo "net.ipv6.conf.all.forwarding=1" >> /etc/sysctl.conf
sysctl -p
```

GRE tunnel on MikroTik router

Note: In this setup, the MikroTik router sits behind your primary router.
GRE (protocol 47) must be allowed on the main router to ensure the tunnel can be established correctly.

IP addresses provided by Luxxy.cloud (example values):

Luxxy GRE server: 5.231.32.3
Luxxy local GRE server address: 100.65.0.1
Your local address: 100.65.0.2
Your public IPs: 5.231.32.101, 5.231.32.102, 5.231.32.103, 5.231.32.104
Note: These are example IP addresses. You will receive the correct IP addresses from Luxxy.cloud.

Local IP address of your MikroTik router (WAN interface): 192.168.1.201
Note: Your MikroTik WAN IP may be different.

1. Rename interfaces

```
/interface ethernet
set [find default-name=ether2] disable-running-check=no name=LAN
set [find default-name=ether1] disable-running-check=no name=WAN
```

2. Create GRE interface

```
/interface gre
add local-address=192.168.1.201 mtu=1476 name=GRE-luxxy remote-address=5.231.32.3
```

3. Add IP addresses

```
/ip address
add address=100.65.0.2/30 interface=GRE-luxxy network=100.65.0.0
add address=5.231.32.101 interface=LAN network=5.231.32.101
```

**Public IP 5.231.32.101 is your “main address” in the allocated range.
Remaining 5.231.32.102–104 will be assigned by DHCP to the servers.**

These rules apply only to the public IP — the rest of the router remains accessible normally.

4. Create a separate routing table

```
/routing table
add disabled=no fib name=luxxy
```

5. Routing: GRE as the default path for public IPs

```
/ip route
add disabled=no distance=1 dst-address=0.0.0.0/0 gateway=100.65.0.1 \
    routing-table=luxxy scope=30 suppress-hw-offload=no target-scope=10

add disabled=no dst-address=5.231.32.102/32 gateway=LAN routing-table=main
add disabled=no dst-address=5.231.32.103/32 gateway=LAN routing-table=main
add disabled=no dst-address=5.231.32.104/32 gateway=LAN routing-table=main
```

Servers on your LAN receiving public IPs will be ARP-visible on the LAN, therefore gateway=LAN is correct.

6. Routing rules — send traffic from public IPs via GRE

```
/routing rule
add action=lookup disabled=no src-address=5.231.32.102/32 table=luxxy
add action=lookup disabled=no src-address=5.231.32.103/32 table=luxxy
add action=lookup disabled=no src-address=5.231.32.104/32 table=luxxy
```

This ensures that servers with public IPs route their outbound traffic through the tunnel.

7. Firewall (access to the public IP)

```
/ip firewall filter
add action=accept chain=input dst-address=5.231.32.101 protocol=icmp
add action=drop chain=input dst-address=5.231.32.101
```

8. DHCP pool for public IPs

```
/ip pool
add name=POOL-luxxy ranges=5.231.32.102,5.231.32.103,5.231.32.104
```

9. Enable DHCP server

```
/ip dhcp-server
add address-pool=POOL-luxxy interface=LAN lease-time=2d name=DHCP-luxxy
```

10. DHCP network parameters

```
/ip dhcp-server network
add address=5.231.32.0/24 dns-server=1.1.1.1,8.8.8.8 gateway=5.231.32.101 netmask=24
```

? Done!

Your public IPs should now route through the GRE tunnel.