

LXC container as a 44Net VPN Router/Firewall

Mark Phillips, NI2O 07/2024 V1.3
Copyright © 2024 All Rights Reserved

Contents

Abstract.....	2
Assumptions.....	2
Requirements.....	2
Obtain a POP account.....	2
Create an LXC container.....	3
Perform system updates.....	8
Configure the VPN.....	8
Test the connection.....	9
Survive a reboot.....	9
Ensure both IPv4 and IPv6 connectivity.....	11
Force IPv4 default rather than IPv6 (optional).....	12
Firewall incoming connections (optional but highly recommended!!!!!!).....	13
NAT outgoing connections.....	13
Route incoming connections.....	14

Abstract

This document aims to walk the reader through the setting up of an LXC container for use as a 44Net VPN Router and Firewall. While this document speaks to ProxMox and LXC systems it may be useful for other Linux/Mac implementations as the steps and commands are almost identical.

Assumptions

Familiarity with ProxMox Virtual Environment
Familiarity with Linux (Debian) networking
Networking basics

Requirements

Functioning Proxmox LXC environment
Functioning LAN with Internet connectivity

Obtain a POP account

Create a POP access account at <https://pop.44net.cloud>. Then request a tunnel. Ensure to click the refresh button at the end of the “Preshared Key (optional)” box. A Preshared Key will ensure even greater security (Wireguard is very secure in itself).

Request a tunnel

Server Required

NODE_Apollo Eastern-Atlanta Vultr Atlanta	zeus Europe Vultr - Frankfurt	VPN1_Fremont_CA Western-California Fremont_Hurricane Electric
<input type="radio"/> Interface_1 44.33.1.0/24, 2a0a:bb06:1::/48	<input type="radio"/> user 44.33.3.0/26, 2a0a:bb06:2:1::/64	<input checked="" type="radio"/> Interface_1 44.31.197.0/24

Name (Optional)

Give your tunnel a name to help you identify it.

Public Key (Optional)

If you do not provide a public key, we will generate you a private key for you.

Preshared Key (Optional)

A preshared key offers an added layer of security.

Dynamic Routing (Optional)

Send me the configuration by email
The email will contain a sample configuration file and, if we generate a private key for you, a QRCode you can use with the WireGuard mobile application.

LXC container as a 44Net Subnet Router V1.3

Copyright © Mark Phillips NI2O 2024 All Rights Reserved

Tunnel details

Your tunnel has been successfully created! Please find the details below.

Your Configuration

Private key—Keep this in a secure place, as it cannot be shown to you again.

UL2w9J94MmentAp0NNIXLdgR6o/LktozHyImq5XTQ2U=

Public key

AcazL4JfynIjXBQ7p+ssQwVZEXhhBxTmxQZ9B1yXRWA=

Allocations

44.31.197.62/32

Server Configuration

Public key

Eq2CoxEu9ekfB+DkxCAJyjjRjYzR38xNAdvR1rzk9Fc=

Preshared key

vKX+NWOLbIjVbnzT7iqrG3cnoLDZqw3Copkpw21hQ9A=

Endpoint

107.161.208.53:12346

Addresses

44.31.197.1

Configuration

Some example configurations to help get you started!

QRCode

wg-quick

[Interface]

PrivateKey = UL2w9J94MmentAp0NNIXLdgR6o/LktozHyImq5XTQ2U=

Address = 44.31.197.62/32

DNS = 1.1.1.1, 1.0.0.1

[Peer]

PublicKey = Eq2CoxEu9ekfB+DkxCAJyjjRjYzR38xNAdvR1rzk9Fc=

PresharedKey = vKX+NWOLbIjVbnzT7iqrG3cnoLDZqw3Copkpw21hQ9A=

Endpoint = 107.161.208.53:12346

PersistentKeepalive = 10

AllowedIPs = 0.0.0.0/0, ::/0

Ensure to copy the “wg-quick” information to a text file somewhere for safe keeping. It will NEVER be shown again. Keep this file for later use.

Create an LXC container

On your ProxMox host create a container with the following specifications

512MB memory

512MB swap

1 CPU

1 network adapter

LXC container as a 44Net Subnet Router V1.3

Copyright © Mark Phillips NI2O 2024 All Rights Reserved

4GB disk space (routers don't need much)
Unprivileged
Nesting

Create the container. Label it in a useful manner. Add both a root password as well as SSH keys to ensure secure remote access if required.

Create: LXC Container

General | Template | Disks | CPU | Memory | Network | DNS | Confirm

Node: pve-radio | Resource Pool: | Password: | Confirm password: | SSH public key(s):
+foCdDIPs1pQVo2Xnl0AIHj79R+
QJuRyC4RIUFkiOzgpQgTMEbC4/
kbQE1N+olpepk0A2R6pRrYVeQ=
= g7lft@deephought.g7lft.com

CT ID: 110

Hostname: 44net_POP

Unprivileged container:

Nesting:

Load SSH Key File

Help | Advanced | Back | Next

Use your favorite linux distribution as the basis for the container (we use Debian)

Create: LXC Container

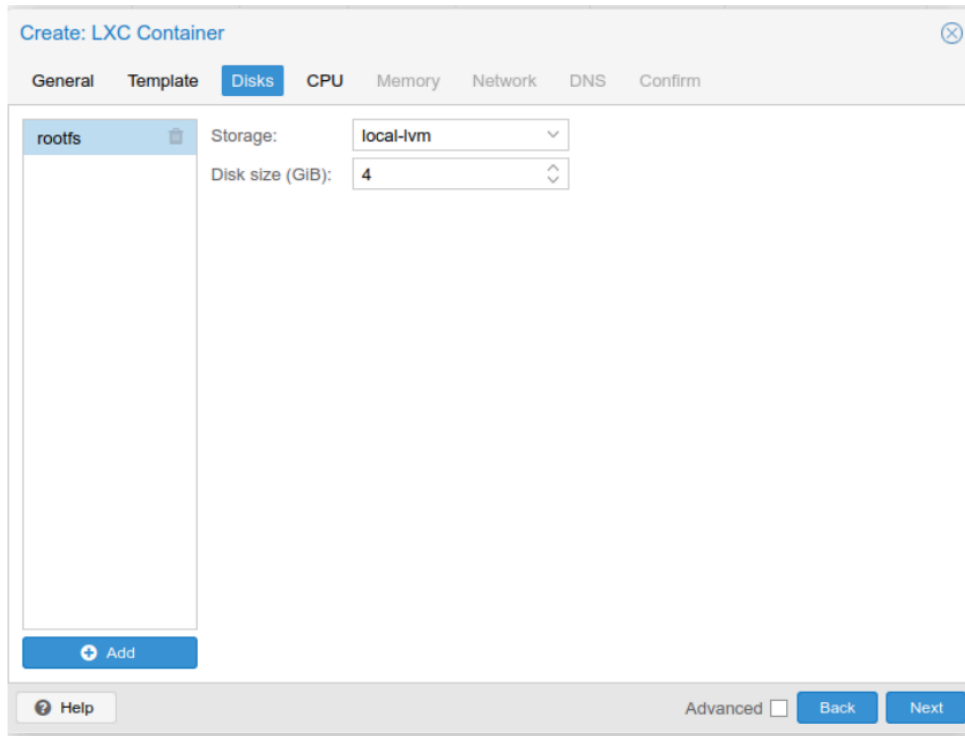
General | **Template** | Disks | CPU | Memory | Network | DNS | Confirm

Storage: local

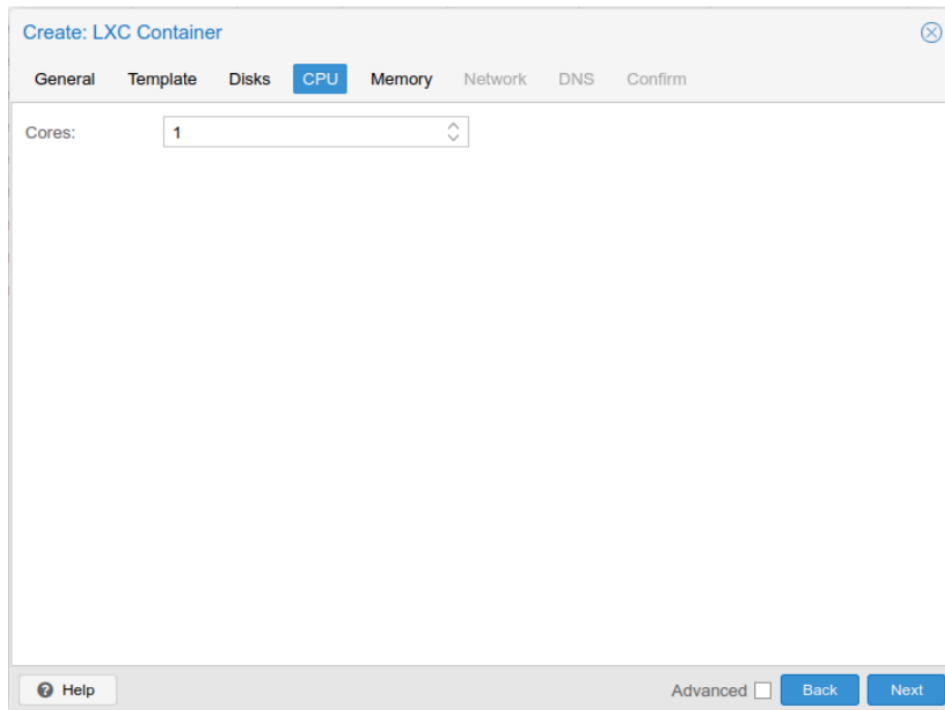
Template: debian-12-standard_12.2-1_amd64

Help | Advanced | Back | Next

Create a disk with 4GB of space for the operating system



Add just 1 CPU core



512MB of RAM and SWAP

Create: LXC Container

General Template Disks CPU **Memory** Network DNS Confirm

Memory (MiB): 512

Swap (MiB): 512

Help Advanced Back Next

Add a static internal IPv4 address to the network interface

Create: LXC Container

General Template Disks CPU Memory **Network** DNS Confirm

Name: eth0 IPv4: Static DHCP

MAC address: auto IPv4/CIDR: 192.168.161.93/24

Bridge: vubr0 Gateway (IPv4): 192.168.161.1

VLAN Tag: no VLAN IPv6: Static DHCP SLAAC

Firewall: IPv6/CIDR:

Disconnect: Rate limit (MB/s): unlimited

MTU: Same as bridge Gateway (IPv6):

Help Advanced Back Next

LXC container as a 44Net Subnet Router V1.3

Copyright © Mark Phillips NI2O 2024 All Rights Reserved

Keep the host DNS settings

Create: LXC Container ⓧ

General Template Disks CPU Memory Network **DNS** Confirm

DNS domain:

DNS servers:

Advanced Back Next

Confirm everything is correct

Create: LXC Container ⓧ

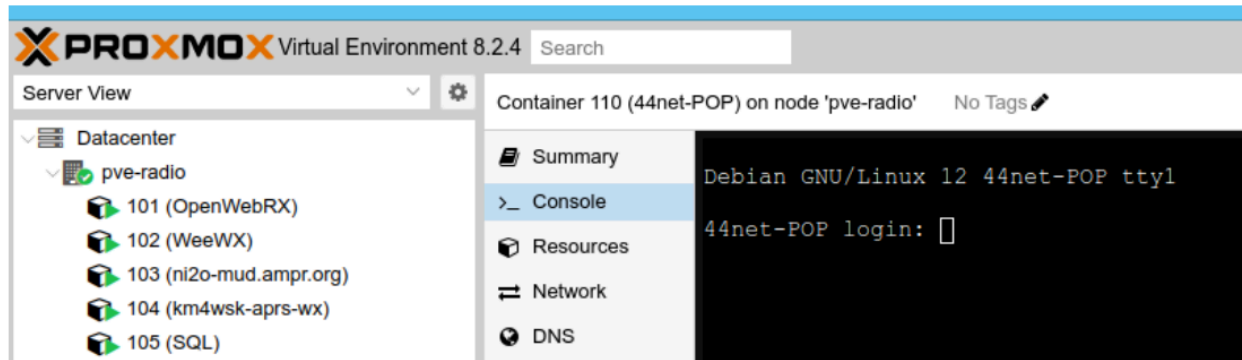
General Template Disks CPU Memory Network DNS **Confirm**

Key ↑	Value
cores	1
features	nesting=1
hostname	44net-POP
memory	512
net0	name=eth0,bridge=vbr0,firewall=1,gw=192.168.161.119,ip6=auto
nodename	pve-radio
ostemplate	local:vztmpl/debian-12-standard_12.2-1_amd64.tar.zst
pool	
rootfs	local-lvm:4
ssh-public-keys	ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA... ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA...
swap	512
unprivileged	1
vmid	110

Start after created

Advanced Back Finish

Start the container!



Perform system updates

Log in as root to the container via the console and perform the system updates and also install some applications and tools(Debian assumed)

```
apt update && apt upgrade -y  
apt install net-tools wireguard ufw resolvconf mtr
```

Configure the VPN

Now that we have the Wireguard software installed create a VPN config file

```
nano /etc/wireguard/44net-pop.conf
```

Then paste in your wg-quick config file saved from earlier

```
[Interface]  
PrivateKey = 8H8ey27QlQq68/H5NOydICzDQbx1E2fWo6t1mo4nfnU=  
Address = 44.33.1.32/32, 2a0a:bb06:1::d/128  
DNS = 1.1.1.1, 1.0.0.1  
  
[Peer]  
PublicKey = CCK21hIIo1BpAzqfQVUjPhVjFes1Z/q9Vh0AU1LJ218=  
PresharedKey = WY8OUQRiErfenglOHzwon8oXGj56vSFj1J9wt0Zfdpo=  
Endpoint = 45.32.220.92:12345  
PersistentKeepalive = 10  
AllowedIPs = 0.0.0.0/0, ::/0
```

LXC container as a 44Net Subnet Router V1.3

Copyright © Mark Phillips NI2O 2024 All Rights Reserved

Test the connection

```
wg-quick up 44net-pop
```

```
root@44net-POP:~# wg-quick up 44net-pop
[#] ip link add 44net-pop type wireguard
[#] wg setconf 44net-pop /dev/fd/63
[#] ip -4 address add 44.33.1.32/32 dev 44net-pop
[#] ip -6 address add 2a0a:bb06:1::d/128 dev 44net-pop
[#] ip link set mtu 1420 up dev 44net-pop
[#] resolvconf -a tun.44net-pop -m 0 -x
[#] wg set 44net-pop fwmark 51820
[#] ip -6 route add ::/0 dev 44net-pop table 51820
[#] ip -6 rule add not fwmark 51820 table 51820
[#] ip -6 rule add table main suppress_prefixlength 0
[#] nft -f /dev/fd/63
[#] ip -4 route add 0.0.0.0/0 dev 44net-pop table 51820
[#] ip -4 rule add not fwmark 51820 table 51820
[#] ip -4 rule add table main suppress_prefixlength 0
[#] sysctl -q net.ipv4.conf.all.src_valid_mark=1
[#] nft -f /dev/fd/63
root@44net-POP:~#
```

Try a traceroute to the ARDC portal

```
mtr portal.ampr.org
```

```
44net-POP (2a0a:bb06:1::d) -> portal.ampr.org (2a0a:bb00:0:44::11) 2024-06-26T23:30:55+0000
[?]sys: Help Display mode Restart statistics Order of fields quit
```

Host	Packets			Pings			
	Loss%	Snt	Last	Avg	Best	Worst	StDev
1. 2a0a:bb06:1::1	0.0%	15	26.0	25.4	21.3	29.4	2.2
2. 2001:19f0:9401:1712:5400:4ff:fe82:2a41	0.0%	15	22.5	24.2	20.4	36.5	3.9
3. (waiting for reply)							
4. v1198-dsl-j2-c35r106-b.sea3.constant.com	0.0%	15	25.0	24.5	20.9	28.0	2.2
5. ethernetet-2-0-15-sr1.atl2.constant.com	0.0%	15	33.1	25.8	21.5	33.1	2.9
6. ethernetatl-er1.atl2.constant.com	0.0%	15	26.6	30.8	21.9	60.1	11.6
7. ethernetee0-er1.atl2.constant.com	0.0%	15	22.6	25.2	21.0	42.6	5.4
8. et-2-0-19-cr2-atl2.ip6.gtt.net	0.0%	15	27.1	27.3	22.9	35.1	3.2
9. 2001:668:0:3:ffff:0:addd:333e	0.0%	14	23.9	26.5	23.0	32.0	2.7
10. atl-bb2-v6.ip.twelve99.net	0.0%	14	21.4	26.0	21.4	34.7	3.6
11. aoh-bb2-v6.ip.twelve99.net	0.0%	14	37.1	36.3	32.6	39.2	2.2
12. nyk-bb2-v6.ip.twelve99.net	92.3%	14	39.1	39.1	39.1	39.1	0.0
13. ldn-bb1-v6.ip.twelve99.net	0.0%	14	113.9	112.9	108.8	117.4	2.9
14. slou-b2-v6.ip.twelve99.net	53.8%	14	114.7	114.2	110.0	119.1	3.1
15. everestdata-ic321418-slou-b1.ip.twelve99-cust.net	0.0%	14	112.3	122.3	109.0	228.0	31.1
16. 2a04:4047:ffff:49::1	0.0%	14	198.3	130.5	110.0	198.3	26.0
17. 2a04:4047:ffff:11::2	0.0%	14	113.7	116.6	110.2	135.4	7.1
18. web1.rdnf.ardc.net	0.0%	14	110.6	114.0	109.7	126.6	4.3

Survive a reboot

We must make the VPN setup survive a reboot so that it all comes back up again when restarted. Use the below commands to create a system “service” that will restart the VPN every time the container is rebooted. The final command reboots the container.

```
systemctl enable wg-quick@44net-pop.service
systemctl daemon-reload
```

LXC container as a 44Net Subnet Router V1.3

Copyright © Mark Phillips NI2O 2024 All Rights Reserved

```
systemctl status wg-quick@44net-pop
```

Finally, reboot the system

```
reboot
```

When the container returns log in and then check for the VPN interface

```
ifconfig
```

```
root
Password:
Linux 44net-POP 6.5.13-1-pve #1 SMP PREEMPT_DYNAMIC PMX 6.5.13-1 (2024-02-05T13:50Z) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jun 26 20:53:47 UTC 2024 on tty1
root@44net-POP:~# ifconfig
44net-pop: flags=209<UP,POINTOPOINT,RUNNING,NOARP> mtu 1420
    inet 44.33.1.32 netmask 255.255.255.255 destination 44.33.1.32
    inet6 2a0a:bb06:1::d prefixlen 128 scopeid 0x0<global>
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 1000 (UNSPEC)
    RX packets 942 bytes 143764 (140.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1004 bytes 166728 (162.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.161.119 netmask 255.255.255.0 broadcast 192.168.161.255
    inet6 fe80::be24:11ff:fe36:a48e prefixlen 64 scopeid 0x20<link>
    inet6 2600:4040:7c7a:c738:be24:11ff:fe36:a48e prefixlen 64 scopeid 0x0<global>
    ether bc:24:11:36:a4:8e txqueuelen 1000 (Ethernet)
    RX packets 100227 bytes 12650038 (12.0 MiB)
    RX errors 0 dropped 55 overruns 0 frame 0
    TX packets 9781 bytes 1615911 (1.5 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 44.33.2.105 netmask 255.255.255.248 broadcast 44.33.2.111
    inet6 fe80::be24:11ff:fe4f:5e91 prefixlen 64 scopeid 0x20<link>
    ether bc:24:11:4f:5e:91 txqueuelen 1000 (Ethernet)
    RX packets 3 bytes 210 (210.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10421 bytes 438254 (427.9 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@44net-POP:~#
```

LXC container as a 44Net Subnet Router V1.3

Copyright © Mark Phillips NI2O 2024 All Rights Reserved

Ensure both IPv4 and IPv6 connectivity

```
ping portal.ampr.org  
ping6 portal.ampr.org
```

```
root@44net-POP:~# ping portal.ampr.org  
PING portal.ampr.org (44.1.1.17) 56(84) bytes of data.  
64 bytes from webl.rdnng.ardc.net (44.1.1.17): icmp_seq=1 ttl=48 time=119 ms  
64 bytes from webl.rdnng.ardc.net (44.1.1.17): icmp_seq=2 ttl=48 time=118 ms  
64 bytes from webl.rdnng.ardc.net (44.1.1.17): icmp_seq=3 ttl=48 time=117 ms  
64 bytes from webl.rdnng.ardc.net (44.1.1.17): icmp_seq=4 ttl=48 time=118 ms  
^C  
--- portal.ampr.org ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3003ms  
rtt min/avg/max/mdev = 116.597/118.022/118.963/0.881 ms  
root@44net-POP:~# ping6 portal.ampr.org  
PING portal.ampr.org(webl.rdnng.ardc.net (2a0a:bb00:0:44::11)) 56 data bytes  
64 bytes from webl.rdnng.ardc.net (2a0a:bb00:0:44::11): icmp_seq=1 ttl=49 time=112 ms  
64 bytes from webl.rdnng.ardc.net (2a0a:bb00:0:44::11): icmp_seq=2 ttl=49 time=114 ms  
64 bytes from webl.rdnng.ardc.net (2a0a:bb00:0:44::11): icmp_seq=3 ttl=49 time=113 ms  
64 bytes from webl.rdnng.ardc.net (2a0a:bb00:0:44::11): icmp_seq=4 ttl=49 time=118 ms  
^C  
--- portal.ampr.org ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3004ms  
rtt min/avg/max/mdev = 111.913/114.268/117.753/2.168 ms  
root@44net-POP:~#
```

Force IPv4 default rather than IPv6 (optional)

By default Linux will prefer IPv6 routes rather than the traditional IPv4. This is not an issue as both IPv4 and IPv6 data will pass over the VPN. For your own personal reasons you may prefer to use IPv4 as the default. Edit the `/etc/gai.conf` file by removing the hash mark from the beginning of the line shown in the example below;

```
nano /etc/gai.conf
```

```
#precedence 2002::/16 30  
#precedence ::/96 20  
#precedence ::ffff:0:0/96 10  
#  
# For sites which prefer IPv4 connections change the last line to  
#  
precedence ::ffff:0:0/96 100  
#  
# scopev4 <mask> <value>  
# Add another rule to the RFC 6724 scope table for IPv4 addresses.  
# By default the scope IDs described in section 3.2 in RFC 6724 are
```

Restart the VPN tunnel

LXC container as a 44Net Subnet Router V1.3

Copyright © Mark Phillips NI2O 2024 All Rights Reserved

```
wg-quick down 44net-pop
```

```
wg-quick up 44net-pop
```

Then test the tunnel again noting the IPv4 addresses used this time

```
mtr portal.ampr.org
```

```
44net-POP (44.33.1.32) -> portal.ampr.org (44.1.1.17) 2024-06-26T23:26:31+0000
[?]: Help Display mode Restart statistics Order of fields quit
Host      Loss%  Snt   Last   Avg    Best  Wrst StDev
 1. 44.33.1.1      0.0%  18    21.7  24.3  21.7  28.0  2.0
 2. 169.254.44.0   0.0%  18    26.4  24.1  20.8  27.3  2.2
 3. (waiting for reply)
 4. 100.100.100.1  0.0%  17    24.9  25.6  21.9  31.7  2.7
 5. 10.77.2.41     0.0%  17    22.5  26.3  22.0  39.0  4.0
 6. 10.77.0.137   0.0%  17    32.0  31.6  23.1  68.9  11.6
 7. 213.248.96.150 0.0%  17    27.4  28.0  24.0  32.5  2.5
 8. atl-bb2-link.ip.twelve99.net 23.5%  17    22.3  25.6  22.3  29.0  2.4
 9. ash-bb2-link.ip.twelve99.net 12.5%  17    40.5  36.4  32.8  40.5  2.3
10. nyk-bb2-link.ip.twelve99.net  0.0%  17    42.2  42.0  37.6  45.3  2.2
11. ldn-bb1-link.ip.twelve99.net  0.0%  17   113.4 116.5 113.0 124.3  3.2
12. slou-b2-link.ip.twelve99.net  0.0%  17   120.0 117.5 114.1 120.2  2.2
13. pulsant-ic-321418.ip.twelve99-cust.net 0.0%  17   209.2 127.5 114.2 209.2  8.6
14. 189-28-167-90.as60610.net  0.0%  17   113.5 121.7 113.5 202.7  21.0
15. e2-g10.rdnq.coreservern.uk  0.0%  17   120.8 118.6 114.8 121.7  2.5
16. web1.rdnq.ordc.net  0.0%  17   116.6 117.5 113.5 120.9  2.2
```

Firewall incoming connections (optional but highly recommended!!!!)

In this document we have created a VPN tunnel from your home to the Great Unwashed Internet and have allowed all that is nasty there to traverse your link. If Freedom of Speech is your thing, you are finished. If you want to protect yourself, others and your equipment I would recommend you install a firewall to stop the miscreants on the Net from hacking your systems.

Earlier in this document we installed a few applications including a firewall tool called Uncomplicated Firewall (UFW). To turn on the firewall simply type

```
ufw enable
```

This will turn on the firewall and protect your system from incoming threats. By default UFW will allow your outgoing connections but disallow any incoming connections just like your home router/firewall. It can be configured to allow port forwarding and other firewall services as you see fit. It is not within the scope of this document to explain the workings of the Uncomplicated Firewall. Further information can be found at

<https://manpages.ubuntu.com/manpages/trusty/man8/ufw.8.html>

NAT outgoing connections

If you are using your new VPN container as a home style router you will need to NAT the outgoing data so that it appears to emanate from the router itself rather than your desktop machine.

To enable this feature it is necessary to further edit the VPN config file we created earlier; LXC container as a 44Net Subnet Router V1.3

Copyright © Mark Phillips NI2O 2024 All Rights Reserved

```
nano /etc/wireguard/44net-pop.conf
```

Add the following lines below the last line in the [Interface] section;

```
PostUp = iptables -A FORWARD -i eth0 -j ACCEPT; iptables -t nat  
-A POSTROUTING -o 44net-pop -j MASQUERADE
```

```
PostDown = iptables -D FORWARD -i eth0 -j ACCEPT; iptables -t  
nat -D POSTROUTING -o 44net-pop -j MASQUERADE
```

As shown in the example below;



```
GNU nano 7.2 /etc/wireguard/44net-pop.conf  
[Interface]  
PrivateKey = 8H8ey27QlQq68/H5NOyfiCzDMbx1E2fWo6t1mo4nfnU=  
Address = 44.33.1.32/32, 2a0a:bb06:1::d/128  
DNS = 1.1.1.1, 1.0.0.1  
PostUp = iptables -A FORWARD -i %i -j ACCEPT; iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE  
PostDown = iptables -D FORWARD -i %i -j ACCEPT; iptables -t nat -D POSTROUTING -o eth0 -j MASQUERADE  
  
[Peer]  
PublicKey = CCK2lhIIolBpAzqfQVUjPhVjFyslZ/R9Vh0AU1LJ218=  
PresharedKey = WY8OUQRiErfeng1OHzwon8oXGj5EvSFj1J9wt0Zfdpo=  
Endpoint = 45.32.220.92:12345  
PersistentKeepalive = 10  
AllowedIPs = 0.0.0.0/0, ::/0
```

Then restart the VPN

```
wg-quick down 44net-pop  
wg-quick up 44net-pop
```

Now we need to tell linux that it should forward traffic. By default it is not a router. To enable forwarding type

```
sysctl -w net.ipv4.ip_forward=1
```

All your outgoing traffic will not only be sent to the Internet over the VPN but will be NAT'd or Masqueraded as if coming from the VPN container itself just like your home router does. Do not forget to add a route into your home router to send all 44.0.0.0/9 and 44.128.0.0/10 data to this container!

Route incoming connections

By default, all incoming connections will be stopped by the UFW firewall application. To allow incoming connections for any subnet you may have please read the appropriate documentation.