# Wireguard on Debian LXC - Proxmox

In Proxmox, create a new Linux container using an Debian template (version 13 was used here). The following configuration is perfectly adequate for home use. Depending on your needs, you can of course increase the number of cores and the amount of memory.

**unprivileged container** (yes)
**0 Cores** (let the Host Kernel decide on CPUs)
**512 MB RAM**
**0 MB Swap**
**8 GB Storage**

# Change LXC Properties

Shutdown the LXC Container, then on the HOST

```
vim /etc/pve/lxc/CONTAINER_NUMBER.conf
```

Paste **the last 2 lines**

```
arch: amd64
features: nesting=1
hostname: Wireguard
memory: 512
net0:
name=eth0,bridge=vmbr0,firewall=1,gw=192.168.1.1,hwaddr=BC:76:55:26:78:BE,ip=192.168.1.101/24,
type=veth
ostype: debian
rootfs: local-lvm:vm-101-disk-0,size=8G
swap: 0
unprivileged: 1
lxc.cgroup.devices.allow: c 10:200 rwm
lxc.mount.entry: /dev/net dev/net none bind,create=dir
```

Save the config file and chown the tun device to give the LXC Container permissions

```
chown 100000:100000 /dev/net/tun
```

# Install Wireguard

Start and enter the LXC Container

```
pct start 101 # (assuming your LXC Container is 101)
pct enter 101 # (assuming your LXC Container is 101)
```

```
apt update
apt dist-upgrade
```

Then you can install wireguard via install script

```
wget git.io/wireguard -O wireguard-install.sh
chmod+x wireguard-install.sh
./wireguard-install.sh
```

It's going to ask for information like below

```
Welcome to this WireGuard road warrior installer!

This server is behind NAT. What is the public IPv4 address or hostname?
Public IPv4 address / hostname [xxx.xxx.xxx.xxx.]: xxxxxxxxx

What port should WireGuard listen to?
Port [51820]:

Enter a name for the first client:
Name [client]: Testclient

Select a DNS server for the client:
   1) Current system resolvers
   2) Google
   3) 1.1.1.1
   4) OpenDNS
```

```
   5) Quad9
   6) AdGuard
DNS server [1]: 1

BoringTun will be installed to set up WireGuard in the system.
Should automatic updates be enabled for it? [Y/n]:

WireGuard installation is ready to begin.
Press any key to continue...
```

Note: If you have a dynamic IP address instead of a static one, you must use a DynDNS service to access your home network from outside your network. Enter your DynDNS address in the “hostname” field and forward port 51820 (UDP) on your router to the WireGuard server.

Inside your LXC, edit the configfile `wg0.conf`

```
vim /etc/wireguard/wg0.conf
```

And add under `interfaces` the following:

```
PostUp = iptables -A FORWARD -i %i -j ACCEPT; iptables -A FORWARD -o %i -j ACCEPT; iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
PostDown = iptables -D FORWARD -i %i -j ACCEPT; iptables -D FORWARD -o %i -j ACCEPT; iptables -t nat -D POSTROUTING -o eth0 -j MASQUERADE
```

to check if the wireguard service is running, exec `systemctl status wg-quick@wg0`

```
root@WireGuard:~# systemctl status wg-quick@wg0
* wg-quick@wg0.service - WireGuard via wg-quick(8) for wg0
     Loaded: loaded (/lib/systemd/system/wg-quick@.service; enabled; vendor preset: enabled)
    Drop-In: /etc/systemd/system/wg-quick@wg0.service.d
             `-boringtun.conf
     Active: active (exited) since Wed 2021-12-08 03:58:02 UTC; 9h ago
       Docs: man:wg-quick(8)
             man:wg(8)
             https://www.wireguard.com/
             https://www.wireguard.com/quickstart/
             https://git.zx2c4.com/wireguard-tools/about/src/man/wg-quick.8
             https://git.zx2c4.com/wireguard-tools/about/src/man/wg.8
    Process: 2880 ExecStart=/usr/bin/wg-quick up wg0 (code=exited, status=0/SUCCESS)
   Main PID: 2880 (code=exited, status=0/SUCCESS)
```

```
        CPU: 41ms

Dec 08 03:58:02 WireGuard systemd[1]: Starting WireGuard via wg-quick(8) for wg0...
Dec 08 03:58:02 WireGuard wg-quick[2880]: [#] ip link add wg0 type wireguard
Dec 08 03:58:02 WireGuard wg-quick[2880]: [#] wg setconf wg0 /dev/fd/63
Dec 08 03:58:02 WireGuard wg-quick[2880]: [#] ip -4 address add 10.7.0.1/24 dev wg0
Dec 08 03:58:02 WireGuard wg-quick[2880]: [#] ip -6 address add fdfd:6c4:2d4:2E4::1/64 dev wg0
Dec 08 03:58:02 WireGuard wg-quick[2880]: [#] ip link set mtu 1420 up dev wg0
Dec 08 03:58:02 WireGuard wg-quick[2880]: [#] iptables -A FORWARD -i wg0 -j ACCEPT; iptables -
A FORWARD -o wg0 -j ACCEPT; iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
Dec 08 03:58:02 WireGuard systemd[1]: Finished WireGuard via wg-quick(8) for wg0.
```

If the service isn't running, you can check it via

```
systemctl enable wg-quick@wg0.service
systemctl daemon-reload
systemctl start wg-quick@wg0
```

# Manage Clients

WireGuard comes with a minimalist management tool: the installation script you downloaded earlier. Run the script as follows:

```
./wireguard-install.sh
```

then a Menu appears:

```
WireGuard is already installed.

Select an option:
   1) Add a new client
   2) Remove an existing client
   3) Remove WireGuard
   4) Exit
Option:
```

Chosse „1“ to create a new Client, enter a name and select a DNS-Resolver

```
Provide a name for the client:
Name: Testclient

Select a DNS server for the client:
   1) Current system resolvers
   2) Google
   3) 1.1.1.1
   4) OpenDNS
   5) Quad9
   6) AdGuard
DNS server [1]: 1
```

Additional there is a QR-Code displayed. You can make a picture of it using a Phone. The config file is usually stored in `$HOME` of the user you runned the script with. In this case root

```
Testclient added. Configuration available in: /root/Testclient.conf
```

the config file looks something similat to this:

```
[Interface]
Address = 10.0.0.0/24, fddd:123:345:678::8/64
DNS = 192.168.1.1
PrivateKey = supersecretprivatekey

[Peer]
PublicKey = supersecretpublickey
PresharedKey = supersecretpresharedkey
AllowedIPs = 0.0.0.0/0, ::/0
Endpoint = xxxxxxxxxx.de:51820
PersistentKeepalive = 25
```

You can copy your Wireguard config using linux commands like scp

`scp root@ip-address:/root/Testclient.conf ~/Downloads`

Or Windows Tools like WinSCP.

# Connect to your Wireguard Server

WireGuard clients are available for all operating systems and platforms. Download the appropriate client, import a previously created configuration file, and enable the VPN connection.

You can view existing peers and their connection status on the server and clients using the following command:

```
wg show
```

on a Linuxclient you'd copy your config file to `/etc/wireguard/wg0.conf`

```
sudo cp ~/Downloads/Testclient.conf /etc/wireguard/wg0.conf
```

And start it via:

```
sudo wg-quick up wg0
```

to disconnect use:

```
sudo wg-quick down wg0
```

# Server Update / Migration – Configuration File

In the event that you need to reset or migrate the server, backing up and restoring the configurations is very easy. Simply back up the file

```
/etc/wireguard/wg0.conf
```

This file contains all the relevant configurations. Simply restore this file on the new server and, if necessary, restart the WireGuard service.

---

Revision #2
Created 2026-03-28 00:14:46 UTC by tinfoil-hat

Updated 2026-03-28 00:15:17 UTC by tinfoil-hat