

VPN

virtual private network VPN

VPN

VPN

Internet VPN (WAN)

- [n2n VPN](#)
- [FreeLAN](#)
- [Outline](#)
- [WireGuard](#)
- [Cloudflare Tunnel](#)
- [ngrok](#)
- [More Solutions](#)
- [Tailscale](#)

n2n VPN

Introduction

n2n ?????????????? EasyTier?????????

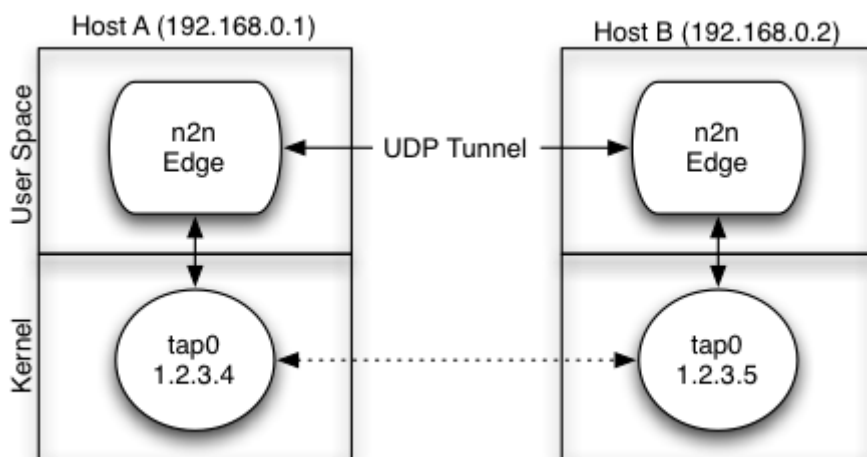
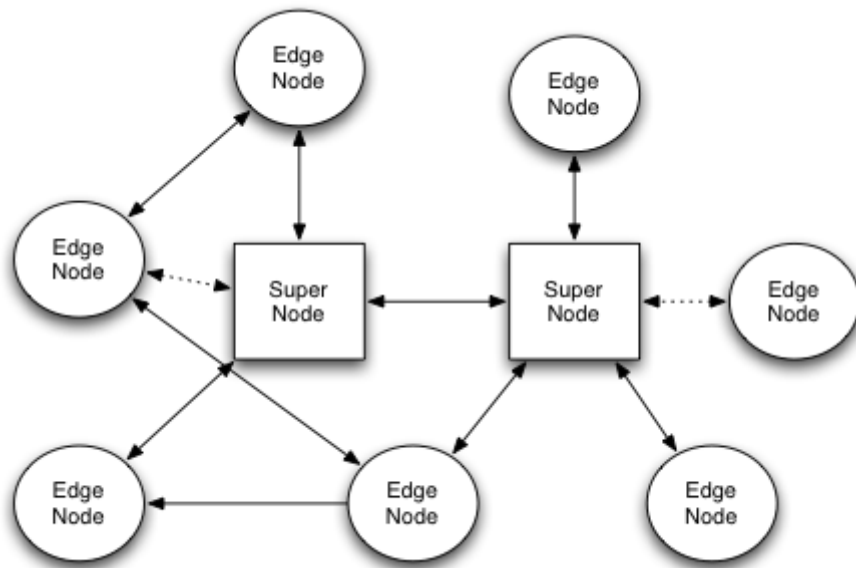
[n2n](#) is a layer-two **peer-to-peer** virtual private network (VPN) which allows users to exploit features typical of P2P applications at network instead of application level. This means that users can gain native IP visibility (e.g. two PCs belonging to the same n2n network can ping each other) and be reachable with the same network IP address regardless of the network where they currently belong. In a nutshell, as OpenVPN moved SSL from application (e.g. used to implement the https protocol) to network protocol, n2n moves P2P from application to network level.

In order to start using n2n, two elements are required:

- A *supernode*: it allows edge nodes to announce and discover other nodes. It must have a port publicly accessible on internet.
- *edge nodes*: the nodes which will be a part of the virtual networks

A virtual network shared between multiple edge nodes in n2n is called a *community*. A single supernode can relay multiple communities and a single computer can be part of multiple communities at the same time. An encryption key can be used by the edge nodes to encrypt the packets within their community.

n2n tries to establish a direct peer-to-peer connection via udp between the edge nodes when possible. When this is not possible (usually due to special NAT devices), the supernode is also used to relay the packets.



Installation

Download: <https://github.com/ntop/n2n>

Ubuntu/Debian

```
sudo dpkg -i n2n_3.0.0-1038_amd64.deb
sudo apt install -f
```

Usage

One-liner command in foreground.

```
# On Linux, change the community, encrypt key and tun IP to your own
# edge -c <community> -k <encrypt key> -a <tun IP address> -l <supernode host:port> -f
sudo edge -c my-community -k my-secret -a 10.9.9.10 -l n2n.lucktu.com:10090 -f
```

Run as service in background.

```
# Generate the config file
sudo cp /etc/n2n/edge.conf.sample /etc/n2n/edge.conf

# Start the edge
sudo systemctl start edge
sudo systemctl enable edge
```

Supernode

```
sudo supernode -p 10090
```

????

- <https://github.com/ntop/n2n>
- [n2n for Windows](#)
- [n2n GUI for Win](#)
- [OpenVPN Download](#)
- [Public Supernodes](#)
- [ZeroTier](#) - P2P VPN Service
- [FRP](#) - Fast Reverse Proxy
 - [Video] [?????FRP?????????????????](#)

FreeLAN

Introduction

[Freelan](#) is a free, open-source, multi-platform, peer-to-peer VPN software that abstracts a LAN over the Internet. It works on Windows, Linux and Mac OSX.

Whether you want to connect the computers of your family, play an old LAN-only game with your friends, or give a privileged access to your private network to your collaborators, freelan will do the job perfectly.

[FreeLAN](#) is free all-around VPN open-source software for Windows, Linux, and macOS that can be used to create three types of VPN:

- Client-server
- Peer-to-peer
- Hybrid that includes the two types mentioned above.

Installation

Download: <https://www.freelan.org/download.html>

Ubuntu/Debian

```
sudo apt update
sudo apt install freelan
```

Windows

1. [?? Visual C++?](#)
2. [?? freelan-2.2.0-amd64-install.exe?](#)

Usage

One-liner command in foreground.

```
# On PC#1, which is correctly configured to allow access to the 12000/UDP from the internet.
# By default, the listen port is 12000/UDP and the tun_tap.ip_address is 9.0.0.1
freelan --security.passphrase "my_secret"
```

```
# On PC#2
```

```
freelan --security.passphrase "my_secret" --tap_adapter.ipv4_address_prefix_length 9.0.0.2/24  
--fscp.contact <IP-to-PC1>:12000
```

Outline

??

[Outline](#) ?????????????????? VPN????????????????????????????? Outline ????? VPN
??

????

- <https://getoutline.org/zh-TW>
- <https://github.com/Jigsaw-Code>
- [??OUTLINE](#)
- [Outline VPN????? VPN ????](#)
- <https://outline.community/>

For ARM CPU

- [Outline Server with ARM64 support](#)

WireGuard

WireGuard Server

- [wireguard-install](#)

WireGuard Client

- YT: [WireGuard - How to Install and Configure WireGuard VPN Client on Ubuntu | Debian | LinuxMint - YouTube](#)
- [wireguard](#) - wireguard gtk gui for linux
- [How to configure WireGuard VPN client with NetworkManager GUI](#)
- [How to set up Wireguard VPN under Linux - Tutorial - YouTube](#)

wg-quick

Installation

```
# Ubuntu/Debian
sudo apt install wireguard

# Fedora
sudo dnf -y install wireguard-tools
```

Generate the key pairs

```
sudo -i
cd /etc/wireguard
wg genkey | tee privatekey | wg pubkey > publickey
```

Configure the WireGuard interface on Peer A

`/etc/wireguard/wg0.conf` :

```
cat << EOF > /etc/wireguard/wg0.conf
[Interface]
Address = 10.0.0.2/32
```

```
PrivateKey = XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
DNS = 8.8.8.8, 4.4.4.4

[Peer]
PublicKey = XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
AllowedIPs = 0.0.0.0/0
Endpoint = the.wireguard.server:51820
EOF
```

Up & Down the wg link

```
sudo wg-quick up wg0
sudo wg
sudo journalctl -fu wg-quick@wg0

sudo wg-quick down wg0
```

nmcli

```
# Import the config file
CONF_FILE="wg0.conf"
nmcli connection import type wireguard file "$CONF_FILE"

# Show the profiles
nmcli
nmcli conn show # List all profiles
nmcli conn show <name> # Display the details for specified profile

# Delete the profile
nmcli connection delete wg0

# Modify the profile my-wg0
nmcli connection modify my-wg0 \
    autoconnect yes \
    ipv4.method manual \
    ipv4.addresses 192.168.7.5/24 \
    wireguard.listen-port 50000 \
    ...

# Active/Inactive the interface
```

```
nmcli connection up my-wg0
nmcli connection down my-wg0
```

Algo VPN

- [Algo VPN](#) is a set of Ansible scripts that simplify the setup of a personal WireGuard and IPsec VPN.

NetBird

- [NetBird](#) combines a configuration-free peer-to-peer private network and a centralized access control system in a single open-source platform
- [Video] [Netbird - an Open Source, Self Hosted Wireguard based VPN system. Server GUI and client setup ease](#)
- [Proxmox VE for Beginners Guide with NetBird LXC](#)

PiVPN

[PiVPN](#) is a lightweight, open-source project designed to simplify setting up a VPN server on a Raspberry Pi or any Debian-based system.

It supports WireGuard and OpenVPN, allowing you to create a secure, private tunnel to your home network or VPS.

- [Self-host Your Own VPN Using PiVPN](#)

wg-easy

[wg-easy](#) is the easiest way to run WireGuard VPN + Web-based Admin UI.

- GitHub: <https://github.com/wg-easy/wg-easy>

WGDashboard

[WGDashboard](#) is a simple and easy-to-use dashboard to manage your WireGuard & AmneziaWG VPN Server(s).

WireGuard-UI

A web user interface to manage your WireGuard setup.

- GitHub: <https://github.com/ngoduykhanh/wireguard-ui>
- [How to Install WireGuard VPN + Web UI on Ubuntu](#)

ngrok

ngrok - put localhost on the internet

?????

- ????????????????
- ??? dev/test ?????????? webhook????????????????????????????????

Pyngrok

ngrok SDK for Python

- [pyngrok - a Python wrapper for ngrok — pyngrok 7.1.6 documentation](#)
- GitHub: <https://github.com/alexdlaird/pyngrok>

URLs

- [ngrok | Unified Application Delivery Platform for Developers](#)
- [?? ngrok ?????????? localhost ??? - MyApollo](#)

?????

- [Localtunnel ~ Expose yourself to the world](#)
- [localhost.run | localhost.run](#)
- [pgrok](#) - Personal ngrok alternative.

Tailscale

[Tailscale](#) ??? Mesh VPN ??????? WireGuard ? end-to-end ?? VPN ?????? Peer-to-peer VPN
???? NAT???????????????? 3 ??? 100 ????

???

- Peer-to-peer VPN ???IoT ????
- Tunnel VPN ???Netflix Sharing
- ????????? Client-server ??
- ?????????
- ????????????????????????????????????? NAT ?????????????
- ???
 - ????????????? Google, Microsoft AD, GitHub, Okta ??
 - Access Controls Lists (ACLs)
 - ?????????

???

- [Tailscale quickstart](#)
- [How you can connect two home labs with a site-to-site VPN \(and why you should\)](#)
- [How to Set Up Remote Access to Your Local Network Using Tailscale VPN](#)
- NetBird: [I switched from Tailscale to this fully self-hosted alternative, and I'm loving it so far](#)
- [Headscale: Awesome Self-Hosted Tailscale Control Server - Virtualization Howto](#)

Alternatives

- [Twingate](#) - ?????????5 users for free
 - [Docs: Twingate](#)
 - [Video] [the END of VPNs?](#)

Peer-to-peer VPN

- ????????????? tailnet ?????????????????????????????????
- ?? IoT ???????
- Tailnet ??????

Tunnet VPN


```
tailscale set --accept-dns=false
```

Exit Node

Advertise a device as an exit node

1. ???? tailscale client
2. ???? IP forwarding
3. ?? tailscale ??
4. ?? Exit node

?? IP forwarding

```
echo 'net.ipv4.ip_forward = 1' | sudo tee -a /etc/sysctl.d/99-tailscale.conf
echo 'net.ipv6.conf.all.forwarding = 1' | sudo tee -a /etc/sysctl.d/99-tailscale.conf
sudo sysctl -p /etc/sysctl.d/99-tailscale.conf
```

?? tailscale ??

```
sudo tailscale up
```

?? Exit node

```
sudo tailscale set --advertise-exit-node
```

?????????

1. Go to the **Machines** page of the admin console.
2. Locate the device in the list. It should display the **Exit Node** badge.
3. Select the ?? menu, then select **Edit route settings**.
4. Check the **Use as exit node** box, then select **Save**.

Use exit node

```
sudo tailscale up
tailscale status
sudo tailscale set --exit-node=<exit-node-ip>

# To stop using an exit node
sudo tailscale set --exit-node=
```

Tailscale Serve

?? Tailnet ?? Reverse Proxy ?????????????????? localhost ????

???? https???? Tailscale ?????? HTTPS Certificates????????????? https://your-device.tailnet-domain?

```
sudo tailscale serve --bg --https=443 127.0.0.1:18789
```

Tools

- [tsui](#) - An (experimental) elegant TUI for configuring Tailscale.

Tailscale on Proxmox

- [Tailscale-LXC/Guide at main · mossc001/Tailscale-LXC · GitHub](#)
- [Tailscale on a Proxmox host · Tailscale Docs](#)
- [Tailscale in LXC containers · Tailscale Docs](#)
- [Installing Tailscale within an LXC container – techwithbrad.com](#)

ScaleTail

? Docker ????? Tailscale ???????????

- [GitHub - tailscale-dev/ScaleTail: Tailscale Sidecar Configurations for Docker · GitHub](#)