

Bridged Setup

Mainly used for Virtualization (e.g. Proxmox/Virtualizor/Virtfusion) to put Virtual Machines and Containers into the bridge so they can use the IPs!

Requirements

Follow the Docs [General](#) at first!

Reminder

To ensure you use the variables correctly, please note that they are written in "<>". You will retrieve the information from our API, which you should have opened beforehand. Replace the placeholders with the related content.

Debian Networking

GRETAP

Install required packages using: `apt update -y; apt install bridge-utils -y`

Add this to the end of the file `/etc/network/interfaces`, e.g. with `nano /etc/network/interfaces`.

Variables to Replace:

<ONE-of-ALLOWED_ENDPOINT(s)> = Literally just the IPv4 of your Server, but it needs to be allowed by us, you enter this on Order!

<ROUTER_IP> = The IPv4 or IPv6 of our Router, the Variable is exactly named like that on the API Page! Depends if your Endpoint is Type v4 or v6

<MTU> = IPv4 Endpoint: 1462, IPv6 Endpoint: 1434 <TYPE> = IPv4 Endpoint: gretap, IPv6 Endpoint: ip6gretap

```

auto vmbr6378
iface vmbr6378 inet manual
    bridge_ports gretap6378
    bridge_stp off
    bridge_fd 0
    mtu <MTU>
    pre-up ip link add name gretap6378 type <TYPE> local <ONE-of-ALLOWED_ENDPOINT(s)> remote
<ROUTER_IP>
    pre-up ip link set up gretap6378
    post-up ip link set up vmbr6378
    down ip link del gretap6378

```

Wireguard

Install required packages using: `apt update -y; apt install bridge-utils wireguard -y`

Add this to the end of the file `/etc/network/interfaces`, e.g. with `nano /etc/network/interfaces`

Variables to Replace:

<ROUTED_IP(s)> = The IPv4 Addresses we assigned to you! Make a 1 to 1 copy and paste from the variable`s content of our API.

```

auto vmbr6378
iface vmbr6378 inet manual
    address 100.96.255.1/32
    bridge_ports none
    bridge_stp off
    bridge_fd 0
    post-up echo 1 > /proc/sys/net/ipv4/conf/all/forwarding
    post-up wg-quick up wg6378
    post-up ip link set up vmbr6378
    post-up bash -c 'export IP_BLOCKS=(<ROUTED_IP(s)>); for ip in "${IP_BLOCKS[@]}"; do ip rule add from "$ip"
table 6378 prio 1; ip route add "$ip" dev vmbr6378; ip addr add "$(echo "$ip" | cut -d"/" -f1 | awk -F"." '{print
$1"."$2"."$3".1"}')"/$(echo "$ip" | cut -d"/" -f2)" dev vmbr6378 2>/dev/null || true; done'
    down bash -c 'export IP_BLOCKS=(<ROUTED_IP(s)>); for ip in "${IP_BLOCKS[@]}"; do ip rule del from "$ip"
table 6378 prio 1; done'
    down ip link del wg6378

```

Create a new file in `/etc/wireguard/`, e.g. with `nano /etc/wireguard/wg6378.conf`

Variables to Replace:

<ROUTER_IPv4> = The IPv4 of our Router, the Variable is exactly named like that on the API Page!
<ROUTER_PORT> = The Tunnel Port of our Router, the Variable is exactly named like that on the API Page!
<CLIENT_PRIVATEKEY>, <ROUTER_PUBLICKEY> = Encryption-Keys required for the Tunnel. Make a 1 to 1 copy and paste from the variable`s content of our API.

[Interface]

PrivateKey = <CLIENT_PRIVATEKEY>

Address = 100.97.255.2/24

Table = 6378

MTU = 1420

[Peer]

PublicKey = <ROUTER_PUBLICKEY>

Endpoint = <ROUTER_IPv4>:<ROUTER_PORT>

PersistentKeepalive = 25

AllowedIPs = 0.0.0.0/0

Last Steps

Bring the bridge up using: `ifup vmbr6378`

or if this command **does not exist** you may restart your networking, which **will interrupt** any other Traffic using `systemctl restart networking`

Revision #11

Created 18 June 2024 21:43:40 by Admin

Updated 22 August 2024 13:53:06 by Admin