

IP Tunnel

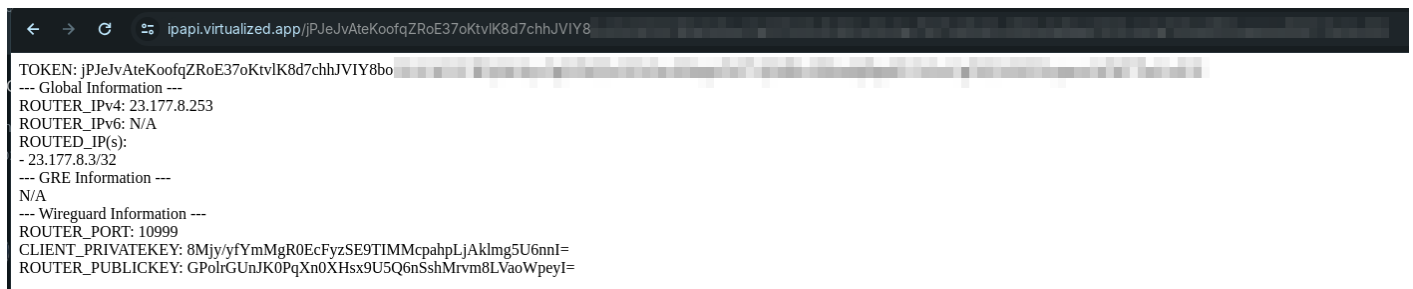
- [General](#)
- [Configuration](#)
 - [Bridged Setup](#)
- [Usage](#)
 - [Proxmox](#)

General

Before you start setting up anything, you should open the API page of your tunnel to get important information at a glance.

Open a Browser or run a curl on the following url like in the example:

```
https://ipapi.virtualized.app/<TOKEN>
```

A screenshot of a web browser window. The address bar shows the URL 'ipapi.virtualized.app/jPJeJvAteKoofqZRoE37oKtvIK8d7chhJVIY8'. The page content displays a JSON response from the API. The response includes a 'TOKEN' field with a long alphanumeric string, 'Global Information' with 'ROUTER_IPv4' (23.177.8.253) and 'ROUTER_IPv6' (N/A), 'ROUTED_IP(s)' with '- 23.177.8.3/32', 'GRE Information' (N/A), and 'Wireguard Information' with 'ROUTER_PORT' (10999), 'CLIENT_PRIVATEKEY' (8Mjy/yfYmMgR0EcFyzSE9TIMMcpahpLjAkimg5U6nnI=), and 'ROUTER_PUBLICKEY' (GP0lrGUnJK0PqXn0XHsx9U5Q6nSshMrvm8LVaoWpeyI=).

```
TOKEN: jPJeJvAteKoofqZRoE37oKtvIK8d7chhJVIY8bo
--- Global Information ---
ROUTER_IPv4: 23.177.8.253
ROUTER_IPv6: N/A
ROUTED_IP(s):
- 23.177.8.3/32
--- GRE Information ---
N/A
--- Wireguard Information ---
ROUTER_PORT: 10999
CLIENT_PRIVATEKEY: 8Mjy/yfYmMgR0EcFyzSE9TIMMcpahpLjAkimg5U6nnI=
ROUTER_PUBLICKEY: GP0lrGUnJK0PqXn0XHsx9U5Q6nSshMrvm8LVaoWpeyI=
```

You will find the `<TOKEN>` if you go to your tunnel product and then to "Product info". Replace it and open the site, if all is right you will get a response like in the example.

Now you have all the information you need to set up the tunnel, go on!

<https://docs.virtualized.app/books/ip-tun>

Configuration

How to configure the Tunnel?

Bridged Setup

Mainly used for Virtualization (e.g. Proxmox/Virtualizer/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`

Usage

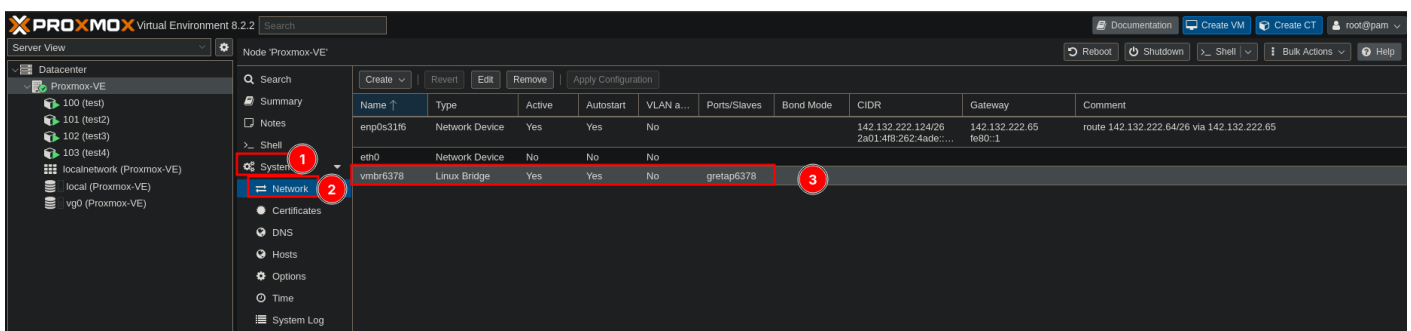
How to use the IPs now?

Proxmox

Requires a Tunnel Type of either GRETAP or Wireguard

Verify



To verify if the Bridge exist select the related Node then go to "System -> Network".




Network Configuration

the Image shows a Container Network Configuration, just as Example

CIDR: 32 | NETMASK: 255.255.255.255
GATEWAY: 100.97.255.1

Edit: Network Device (veth)  

Name:	<input type="text" value="eth0"/>	IPv4: <input checked="" type="radio"/> Static <input type="radio"/> DHCP
MAC address:	<input type="text" value="BC:24:11:8F:E2:D7"/>	IPv4/CIDR: <input type="text" value="77.90.52.253/32"/>
Bridge:	<input type="text" value="vmbr6378"/>	Gateway (IPv4): <input type="text" value="100.97.255.1"/>
VLAN Tag:	<input type="text" value="no VLAN"/>	IPv6: <input checked="" type="radio"/> Static <input type="radio"/> DHCP <input type="radio"/> SLAAC
Firewall:	<input type="checkbox"/>	IPv6/CIDR: <input type="text" value="None"/>
		Gateway (IPv6): <input type="text"/>

 Help Advanced ☐

Test Network

```
root@test:/# curl ipinfo.io
{
  "ip": "23.177.8.4",
  "city": "Tucson",
  "region": "Arizona",
  "country": "US",
  "loc": "32.2691,-110.9845",
  "postal": "85705",
  "timezone": "America/Phoenix",
  "readme": "https://ipinfo.io/missingauth"
}root@test:/# ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=53 time=7.57 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=53 time=7.70 ms
□
```