

# WireGuard

- [Basic concepts WireGuard](#)
- [Technical requirements and installation](#)
- [Creating a WireGuard Configuration](#)
- [Changing WireGuard Configuration](#)
- [Diagnostic Information](#)
- [Port Forwarding](#)

# Basic concepts WireGuard

[Order now](#) | [Download](#) | [FAQ](#)

Since version 1.3 **PUQVPNCP** supports VPN protocol **WireGuard**

**WireGuard** is the main **VPN** protocol of the **PUQVPNCP** system.

This means that the WireGuard protocol must be installed and configured correctly. If you have carefully carried out the panel installation process according to our instructions, then all packages are ready to work and you do not need to do anything else.

The primary place where configuration changes are made is the WireGuard interface, the internal network address space is configured on the interface, the public IP address for NAT implementation, as well as DNS server settings, and VPN clients are connected to the interface and much more.

WireGuard interface cannot be disabled (only removed)

## WireGuard protocol available to clients

- **Android** (Official application from **WireGuard**)
- **iOS** (Official application from **WireGuard**)
- **macOS** (Official application from **WireGuard**)
- **Linux** (Official application from **WireGuard** wireguard-dkms wireguard-tools)
- **Windows** (Official application from **WireGuard**)

## Usage features WireGuard

- User must install **WireGuard** software (<https://www.wireguard.com/install/>)
- Import VPN configuration to VPN client, from file or QR code.

# Technical requirements and installation

[Order now](#) | [Download](#) | [FAQ](#)

## Technical requirements

- Operating systems: Debian 11+ (amd64), Ubuntu 20+ (amd64)
- Real, public IP address on server interface
- Domain name for the server
- **PUQVPNCP**
- Installed packages **wireguard wireguard-dkms wireguard-tools** (*Included in the installation process*)

## Installation

We issue all commands after logging into the SSH terminal window as the root user.

Linux kernels less than 5.6 ( $\leq 5.5$ ) did not include Wireguard as a feature in the upstream kernel code. Adding Wireguard support to these (older) kernels is possible via additional modules

### Check kernel version

```
uname -sr
```

```
Linux 5.10.0-10-amd64
```

```
apt-get update
```

```
apt-get upgrade
```

```
reboot
```

For Debian 10: **WireGuard** is in Debian backported repo. Hence, enable backports as follows, run:

```
sudo sh -c "echo 'deb http://deb.debian.org/debian buster-backports main contrib non-free' >
/etc/apt/sources.list.d/buster-backports.list"
sudo apt update
```

```
apt-get install wireguard wireguard-dkms wireguard-tools -y
```

# Checking installed packages

## Checking the **wireguard** status

```
dpkg -s wireguard
```

Output should look similar to this:

```
Package: wireguard
Status: install ok installed
Priority: optional
Section: net
Installed-Size: 17
Maintainer: Daniel Kahn Gillmor <dkg@fifthhorseman.net>
Architecture: all
Version: 1.0.20210223-1
Depends: wireguard-modules (>= 0.0.20191219) | wireguard-dkms (>= 0.0.20200121-2), wireguard-
tools (>= 1.0.20210223-1)
Description: fast, modern, secure kernel VPN tunnel (metapackage)
WireGuard is a novel VPN that runs inside the Linux Kernel and uses
state-of-the-art cryptography (the "Noise" protocol). It aims to be
faster, simpler, leaner, and more useful than IPSec, while avoiding
the massive headache. It intends to be considerably more performant
than OpenVPN. WireGuard is designed as a general purpose VPN for
running on embedded interfaces and super computers alike, fit for
many different circumstances. It runs over UDP.
```

This metapackage explicitly depends on both the kernel module and the userspace tooling.

Homepage: <https://www.wireguard.com>

# Checking installed packages

## Checking the **wireguard-dkms**

```
dpkg -s wireguard-dkms
```

Output should look similar to this:

```
Package: wireguard-dkms
Status: install ok installed
Priority: optional
Section: kernel
Installed-Size: 1724
Maintainer: Daniel Kahn Gillmor <dkg@fifthorseman.net>
Architecture: all
Source: wireguard-linux-compat
Version: 1.0.20210219-1
Depends: dkms (>= 2.1.0.0), perl:any
Recommends: wireguard (>= 0.0.20191219), wireguard-tools (>= 0.0.20191219)
Description: fast, modern, secure kernel VPN tunnel (DKMS version)
 WireGuard is a novel VPN that runs inside the Linux Kernel and uses
 state-of-the-art cryptography (the "Noise" protocol). It aims to be
 faster, simpler, leaner, and more useful than IPSec, while avoiding
 the massive headache. It intends to be considerably more performant
 than OpenVPN. WireGuard is designed as a general purpose VPN for
 running on embedded interfaces and super computers alike, fit for
 many different circumstances. It runs over UDP.
.
 This package uses DKMS to automatically build the wireguard kernel
 module.
Homepage: https://www.wireguard.com
```

# Checking installed packages

## Checking the **wireguard-tools**

```
dpkg -s wireguard-tools
```

Output should look similar to this:

```
Package: wireguard-tools
Status: install ok installed
Priority: optional
Section: net
Installed-Size: 319
Maintainer: Daniel Kahn Gillmor <dkg@fifthhorseman.net>
Architecture: amd64
Source: wireguard
Version: 1.0.20210223-1
Depends: libc6 (>= 2.14)
Recommends: nftables | iptables, wireguard-modules (>= 0.0.20171001) | wireguard-dkms (>= 0.0.20191219)
Suggests: openresolv | resolvconf
Description: fast, modern, secure kernel VPN tunnel (userland utilities)
WireGuard is a novel VPN that runs inside the Linux Kernel and uses state-of-the-art cryptography (the "Noise" protocol). It aims to be faster, simpler, leaner, and more useful than IPSec, while avoiding the massive headache. It intends to be considerably more performant than OpenVPN. WireGuard is designed as a general purpose VPN for running on embedded interfaces and super computers alike, fit for many different circumstances. It runs over UDP.
.
This package contains command-line tools to interact with the WireGuard kernel module. Currently, it provides only a single tool:
.
wg: set and retrieve configuration of WireGuard interfaces
Homepage: https://www.wireguard.com
```



# Creating a WireGuard Configuration

[Order now](#) | [Download](#) | [FAQ](#)

In order for the WireGuard solution to work properly, it is necessary to create, among others: interface for Wireguard and configure other settings

**WireGuard's** configuration is available in the menu item **VPN servers->WireGuard**

The screenshot shows a dashboard for 'PUQVPNCP' with a user 'ruslan (2a02:a311:4041:200:49:6742:113a:949b)'. The 'WireGuard' section is active, displaying a table of configurations and a 'Create' button.

Find By Name	Peers	IKEv2	Find By Interface	Find By Network	Internal Traffic	IP:Port	Bandwidth	Keepalive		
<a href="#">1-4847</a>	0	YES	wg0	10.0.0.1/24	NO	77.87.125.200:51820 DNS: 8.8.8.8 1.1.1.1	100M / 100M	25	<a href="#">Port forwarding</a>	<a href="#">Delete</a>
<a href="#">77-87-125-209</a>	14	YES	wg110	10.0.111.1/24 2a11:ff00::101/120	YES	77.87.125.200:51930 DNS: 10.0.111.1	5M / 2M	25	<a href="#">Port forwarding</a>	<a href="#">Delete</a>
<a href="#">8-4986</a>	1	YES	wg2	10.0.2.1/24	YES	77.87.125.200:51822 DNS: 8.8.8.8 1.1.1.1	100M / 100M	25	<a href="#">Port forwarding</a>	<a href="#">Delete</a>
<a href="#">Default</a>	2	NO	wg1	10.0.1.1/24	NO	77.87.125.200:51821 DNS: 10.0.1.1 77.87.125.200	Unlimited / Unlimited	0	<a href="#">Port forwarding</a>	<a href="#">Delete</a>

To create a new **WireGuard** server, click the Create button.

PUQVPNCP ? HELP ? ruslan (213.134.190.109) Logout

Dashboard  
VPN servers  
VPN accounts  
Settings  
About us

## WireGuard / Add WireGuard

<b>Name</b> <input type="text" value="Default_2"/> <b>Private key</b> <input type="text" value="CFLkYZGWgJ/DmlJ4ycHKNJuQkhfe1"/> <b>Public key</b> <input type="text" value="6DnTiOck280zaNC7pzn94k6csZIBkF"/>	<b>Interface name</b> <input type="text" value="wg4"/> <b>MTU</b> <input type="text" value="0"/> <i>0 - disabled</i> <b>Port</b> <input type="text" value="51824"/> <b>Internal Traffic</b> <input type="text" value="DROP"/> <b>Disable NAT</b> <input type="text" value="NO"/> <b>IP/MASK</b> <input type="text" value="10.0.3.1/24"/> <b>External IP</b> <input type="text" value="77.87.125.200"/> <b>DNS 1</b> <input type="text" value="10.0.3.1"/> <b>DNS 2</b> <input type="text" value="77.87.125.200"/> <b>IPv6</b> <input type="text" value="NO"/> <b>IPv6/MASK</b> <input type="text" value="::/0"/> <b>DNS 1 IPv6</b> <input type="text" value="::"/> <b>DNS 2 IPv6</b> <input type="text" value="::"/>	<b>Peers configuration</b> <b>Bandwidth download (in M) per peer</b> <input type="text" value="0"/> <b>Bandwidth upload (in M) per peer</b> <input type="text" value="0"/> <b>Persistent Keepalive</b> <input type="text" value="0"/> <i>0 - disabled</i> <b>AllowedIPs</b> <input type="text" value="0.0.0.0/0, ::/0"/> <i>Empty will mean "0.0.0.0/0"</i> <i>If enabled IPv6 "0.0.0.0/0, ::/0"</i> <b>Endpoint</b> <input type="text"/> <i>Fill without port</i> <i>Empty then filled automatically</i> <b>IKEv2 Enabled</b> <input type="text" value="NO"/>
---	---	---

The system will automatically fill in the form for creating a new server with unique data.

You can change the data if necessary.

- **Name** - This is a unique configuration name, this name appears in the system as the main configuration model of the **WireGuard** interface, this parameter cannot be changed later
- **Private key/Public key** - Keys for encrypting the traffic of the WireGuard interface, the system generated new keys, but you can set them yourself when creating the **WireGuard** interface
- **Interface name** - Name of the **WireGuard** network interface in the system, this parameter cannot be changed
- **IP/MASK** -The parameters of the internal network of clients of this **WireGuard** interface, the address that is specified will be assigned to the interface and for all clients of this interface it will be the default gateway.
- **Internal Traffic** - Allow or deny traffic exchange between the client of this interface
- **Disable NAT**- If set to YES, then NAT rules will not be added to the firewall, which is necessary for public IP for the client or restricting access to the Internet.
- **Port** - Port on which the interface will listen for incoming connections
- **External IP** - The public IP address that will be used in the interface configuration, NAT will be organized through this address for all clients of this interface. **The address must be public and configured on the server.**
- **DNS 1/DNS 2** - DNS servers that will be issued to the client of this interface
- **Bandwidth download/Bandwidth upload** - conditional value for the throughput of each peer connected to this **WireGuard** interface. This data will be automatically applied when creating a VPN client for this WireGuard interface.
- **Persistent Keepalive** - A sensible interval that works with a wide variety of firewalls is

25 seconds. Setting it to 0 turns the feature off, which is the default, since most users will not need this, and it makes **WireGuard** slightly more chatty

- **MTU** - Ability to set **MTU** on the **WireGuard** interface. This parameter is involved in generating the client settings configuration.
- **AllowedIPs** - This parameter is involved in generating the client settings configuration.
- **IKEv2 Enabled** - Enables **IKEv2** protocol support for this interface. If set to **YES** then users of this interface will connect to the server using the **IKEv2** protocol
- **IPv6** - Enable or disable IPV6
- **IPv6/MASK** - IPv6 subnet to be distributed among peers
- **DNS 1 IPv6/DNS 2 IPv6** - IPv6 DNS servers

# Changing WireGuard Configuration

[Order now](#) | [Download](#) | [FAQ](#)

**WireGuard** configuration is available in the menu item **VPN servers->WireGuard**

Select the **WireGuard** interface you want to change and click on the **Edit** button

You must understand that changing any interface parameters will completely remove all old configuration and create an interface with new parameters.

In case of changing critical parameters, each client must reconfigure the connection taking into account the new configuration.

Find By Name	Peers	IKEv2	Find By Interface	Find By Network	Internal Traffic	IP:Port	Bandwidth	Keepalive		
<a href="#">1-4847</a>	0	YES	wg0	10.0.0.1/24	NO	77.87.125.200:51820 DNS: 8.8.8.8 1.1.1.1	100M / 100M	25	<a href="#">Port forwarding</a>	<a href="#">Delete</a>
<a href="#">77-87-125-209</a>	14	YES	wg110	10.0.111.1/24 2a11:ff00::101/120	YES	77.87.125.200:51930 DNS: 10.0.111.1	5M / 2M	25	<a href="#">Port forwarding</a>	<a href="#">Delete</a>
<a href="#">8-4986</a>	1	YES	wg2	10.0.2.1/24	YES	77.87.125.200:51822 DNS: 8.8.8.8 1.1.1.1	100M / 100M	25	<a href="#">Port forwarding</a>	<a href="#">Delete</a>
<a href="#">Default</a>	2	NO	wg1	10.0.1.1/24	NO	77.87.125.200:51821 DNS: 10.0.1.1 77.87.125.200	Unlimited / Unlimited	0	<a href="#">Port forwarding</a>	<a href="#">Delete</a>

PUQVPNCP ? HELP ? ruslan (2a02:a311:4041:200:49:67:42:113a:949b) Logout

**WireGuard / Edit WireGuard**

Save Set Bandwidth Port forwarding

Name:  Port: 51930

Interface name:

Private key:

Public key:  Public key

Firewall Nat:  10.0.111.0/24 -> 0.0.0.0/0 SNAT to:77.87.125.209 Pkt:0 Bytes:0

Firewall Filter:  10.0.111.0/24 -> 10.0.111.0/24 ACCEPT Pkt:0 Bytes:0

Traffic Control:  iptables -t nat -F root -r 28 10 default 0 direct\_packets\_stat 0 direct\_qlen 1000 Sent 0 bytes 0 pkt (dropped 0, overlimits 0, requests 0) backlog 0b 0p requests 0

Name	Status	IP	Download	Upload	Mangle	
dimon_pc	Enable	10.0.111.12 2a11:f00c:10b	5M	2M	324	<a href="#">Edit</a>
dimon_telefon	Enable	10.0.111.6 2a11:f00c:106	5M	2M	110	<a href="#">Edit</a>
dino_pc	Enable	10.0.111.13 2a11:f00c:108	5M	2M	538	<a href="#">Edit</a>
dino_telefon	Enable	10.0.111.7 2a11:f00c:10c	5M	2M	209	<a href="#">Edit</a>
galia_pc	Enable	10.0.111.14 2a11:f00c:10d	5M	2M	211	<a href="#">Edit</a>
mikrotik	Enable	10.0.111.9 2a11:f00c:102	5M	2M	100	<a href="#">Edit</a>
mikrotik2	Enable	10.0.111.10 2a11:f00c:103	5M	2M	214	<a href="#">Edit</a>
mikrotik3	Enable	10.0.111.8 2a11:f00c:104	5M	2M	216	<a href="#">Edit</a>
ruslan_dom_pc	Enable	10.0.111.2 2a11:f00c:109	5M	2M	322	<a href="#">Edit</a>
ruslan_mint	Enable	10.0.111.15 2a11:f00c:10a	5M	2M	101	<a href="#">Edit</a>
ruslan_pc	Enable	10.0.111.3 2a11:f00c:105	5M	2M	286	<a href="#">Edit</a>
ruslan_telefon	Enable	10.0.111.11 2a11:f00c:107	5M	2M	131	<a href="#">Edit</a>
ruslan_windows	Enable	10.0.111.4 2a11:f00c:10e	5M	2M	210	<a href="#">Edit</a>
test_mac	Enable	10.0.111.5 2a11:f00c:10f	5M	2M	212	<a href="#">Edit</a>

## You can change the following parameters of the WireGuard interface

- **Private key/Public key** - Keys for encrypting the traffic of the WireGuard interface, the system generated new keys, but you can set them yourself when creating the **WireGuard** interface
- **IP/MASK** - The parameters of the internal network of clients of this **WireGuard** interface, the address that is specified will be assigned to the interface and for all clients of this interface it will be the default gateway.
- **Internal Traffic** - Allow or deny traffic exchange between the client of this interface
- **Port** - Port on which the interface will listen for incoming connections
- **External IP** - The public IP address that will be used in the interface configuration, NAT will be organized through this address for all clients of this interface. **The address must be public and configured on the server.**
- **DNS 1/DNS 2** - DNS servers that will be issued to the client of this interface
- **Bandwidth download/Bandwidth upload** - conditional value for the throughput of each peer connected to this **WireGuard** interface. This data will be automatically applied when creating a VPN client for this WireGuard interface.
- **Persistent Keepalive** - A sensible interval that works with a wide variety of firewalls is 25 seconds. Setting it to 0 turns the feature off, which is the default, since most users will not need this, and it makes **WireGuard** slightly more chatty
- **MTU** - Ability to set **MTU** on the **WireGuard** interface. This parameter is involved in generating the client settings configuration.
- **AllowedIPs** - This parameter is involved in generating the client settings configuration.
- **IKEv2 Enabled** - Enables **IKEv2** protocol support for this interface. If set to **YES** then users of this interface will connect to the server using the **IKEv2** protocol
- **IPv6** - Enable or disable IPV6
- **IPv6/MASK** - IPv6 subnet to be distributed among peers
- **DNS 1 IPv6/DNS 2 IPv6** - IPv6 DNS servers

**"Set Bandwidth"** button, which automatically sets the bandwidth of all clients of the external interface/server Set Bandwidth for the parameters that are entered in the section Peer configuration

# Diagnostic Information

[Order now](#) | [Download](#) | [FAQ](#)

## WireGuard diagnostic Information is available in the menu item VPN servers->WireGuard

Select the **WireGuard** interface for which you want to display diagnostic information and click the button "Edit" in the corresponding row.

The screenshot shows the 'WireGuard' management page. On the left is a sidebar with navigation options: Dashboard, VPN servers (selected), IKEv2, VPN accounts, Settings, and About us. The main area has a 'Create' button and a table of existing VPN servers. The table columns are: Find By Name, Peers, IKEv2, Find By Interface, Find By Network, Internal Traffic, Port, External IP, DNS, Peer Download, Peer Upload, and Keepalive. Each row includes 'Edit' and 'Delete' buttons.

Find By Name	Peers	IKEv2	Find By Interface	Find By Network	Internal Traffic	Port	External IP	DNS	Peer Download	Peer Upload	Keepalive		
1-4819	0	YES	wg7	10.0.6.1/24	YES	51827	77.87.125.209	8.8.8.8 1.1.1.1	3M 4M		5	Edit	Delete
1-4820	1	YES	wg8	10.0.7.1/24	YES	51828	77.87.125.200	8.8.8.8 1.1.1.1	3M 4M		0	Edit	Delete
77_87_125_209	12	YES	wg110	10.0.110.1/24	YES	51930	77.87.125.200	10.0.110.1 77.87.125.200	30M 30M		5	Edit	Delete
Default	34	YES	wg0	13.11.11.10/24	NO	51820	77.87.125.200	8.8.8.8 1.1.1.1	10M 5M		0	Edit	Delete
Default_1	29	YES	wg1	10.0.0.1/24	YES	51821	77.87.125.200	8.8.8.8 1.1.1.1	10M 10M		0	Edit	Delete
Default_10	38	YES	wg10	10.0.200.3/24	YES	51830	77.87.125.200	8.8.8.8 1.1.1.1	20M 30M		0	Edit	Delete
Default_100	26	YES	wg100	10.0.99.1/24	NO	51920	77.87.125.200	8.8.8.8 1.1.1.1	10M 10M		0	Edit	Delete

The screenshot shows the 'WireGuard / Edit WireGuard' configuration page. The left sidebar is the same as the previous screenshot. The main area has 'Save' and 'Set Bandwidth' buttons. Configuration fields include: Name (77\_87\_125\_209), Interface name (wg110), Private key, Public key, IP/MASK (10.0.110.1/24), Internal Traffic (ACCEPT), Port (51930), External IP (77.87.125.200), DNS 1 (10.0.110.1), DNS 2 (77.87.125.200), Peers configuration (Bandwidth download: 30, Bandwidth upload: 30, Persistent Keepalive: 5, IKEv2 Enabled: YES). Below these are sections for Public key, Firewall Nat, Firewall Filter, and Traffic Control. A table lists peers with columns: Name, Status, IP, Download Upload, and Mangle. Each peer has an 'Edit' button.

Name	Status	IP	Download Upload	Mangle	
dimon_pc	Enable	10.0.110.5	30M 30M	324	Edit
dimon_telefon	Enable	10.0.110.7	30M 30M	110	Edit
dino_pc	Enable	10.0.110.6	30M 30M	538	Edit
dino_telefon	Enable	10.0.110.8	30M 30M	209	Edit
galia_pc	Enable	10.0.110.11	30M 30M	211	Edit
peer_101	Enable	10.0.110.9	30M 30M	207	Edit
peer_215	Enable	10.0.110.2	30M 30M	323	Edit
ruslan_dom_pc	Enable	10.0.110.4	30M 29M	322	Edit
ruslan_pc	Enable	10.0.110.3	30M 30M	286	Edit

- **Public key/Port** - The actual data that is installed in the system on this interface
- **Firewall Nat** - The actual data is taken from the system firewall, this is a rule that implements nat, with statistics on packet counters and traffic passing through this rule.
- **Firewall Filter** - The actual data is taken from the system firewall, these are rules allowing internal traffic of interface clients, with statistics on packet counters and traffic passing through this rules.
- **Traffic Control** - The actual data is taken from the system with the **Traffic control** configuration, it shows that the interface is involved in filtering traffic in order to limit the speed to the clients of this interface.

Further, there is a table in which the list of all clients which are assigned to this interface.

# Port Forwarding

[Order now](#) | [Download](#) | [FAQ](#)

**Port forwarding** is a networking technique used to allow external devices to access services running on a local network. Essentially, it involves redirecting incoming network traffic from a specific port on a router or firewall to a specific device or port on the internal network. This allows devices outside the local network to access resources such as web servers, FTP servers, or game servers hosted on a local network. Port forwarding is often used for remote access to devices, for example, accessing a security camera or a home automation system from a remote location.

To access port forwarding settings, select the Wireguard server for which you wish to configure port forwarding and click on the **port forwarding** button.

The screenshot shows the PUVPNCP web interface. The top navigation bar includes a logo, a help button, a user profile 'ruslan (77.87.125.4)', and a 'Logout' button. A sidebar on the left contains menu items: 'Dashboard', 'VPN servers', 'VPN accounts', 'Settings', and 'About us'. The main content area is titled 'WireGuard / Edit WireGuard' and features three buttons: 'Save', 'Set Bandwidth', and 'Port forwarding' (highlighted with a red box). Below these buttons are several form fields for configuring a WireGuard interface: Name (1-4847), Interface name (wg6), Private key (jGcmAi1IURH2tdaYkJ211KeyTYpMiDy), Public key (4gLjoeEEh5JmHC8d5N1kz8Hvl5ukCIE), IP/MASK (10.0.5.1/24), Internal Traffic (ACCEPT), Port (65535), and External IP (77.87.125.200). DNS 1 is 8.8.8.8 and DNS 2 is 1.1.1.1. To the right, there are sections for 'Public key' (4gLjoeEEh5JmHC8d5N1kz8Hvl5ukCIEroCM6a5GREBM=) with Port 65535, 'Firewall Nat' (10.0.5.0/24 -> 0.0.0.0/0 SNAT to:77.87.125.200 Pkt:0 Bytes:0), 'Firewall Filter' (10.0.5.0/24 -> 10.0.5.0/24 ACCEPT Pkt:0 Bytes:0), and 'Traffic Control' (qdisc htb 1: root refcnt 2 r2q 10 default 0 direct\_packets\_stat 0 direct\_qlen 1000 Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0) backlog 0b 0p requeues 0). At the bottom, a table lists current port forwarding rules:

Name	Status	IP	Download Upload	Mangle	
1-4847-2	Enable	10.0.5.2	100M 100M	217	<a href="#">Edit</a>
1-4847-3	Enable	10.0.5.3	100M 100M	221	<a href="#">Edit</a>

When you access the port forwarding settings, a list of all currently forwarded ports from the external IP address to the internal account will be displayed. If you wish to add a new port forwarding rule, simply fill out the necessary information and click on the **"ADD"** button. Conversely, if you need to remove an existing port forwarding rule, click on the **"DELETE"** button associated with the relevant entry. These options provide a great deal of flexibility in managing your port forwarding settings to ensure that external devices can access the resources on your VPN network that you want to make available.

- Dashboard
- VPN servers
- VPN accounts
- Settings
- About us

## WireGuard / 1-4847 / Port Forwarding

Successfully

**Name**  
1-4847

**Interface name**  
wg6

**IP/MASK**  
10.0.5.1/24

**Port**  
65535

**External IP**  
77.87.125.200

**Add port forwarding**

**Protocol**  
tcp

**DST port**  
Enter port

**To peer**  
1-4847-2 (10.0.5.2)

**To port**  
Enter port

Protocol	DST Port	To Peer (IP)	To port	
tcp	11	1-4847-4 (10.0.5.4)	22	<input type="button" value="Delete"/>
udp	444	1-4847-4 (10.0.5.4)	66	<input type="button" value="Delete"/>
tcp	44	1-4847-5 (10.0.5.5)	55	<input type="button" value="Delete"/>