Installing and Configuring WireGuard® on Linux as a VPN server

WireGuard Overview

WireGuard (https://www.wireguard.com/) is a modern VPN (Virtual Private Network) (https://en.wikipedia.org/wiki/Virtual_private_network) software. It is designed to be run almost anywhere and to be cross-platform. Compared to other similar software, it is faster, more secure and simpler.

Requirements:

- You have an account and are logged into console.scaleway.com (https://console.scaleway.com)
- You have configured your SSH Key (https://www.scaleway.com/docs/configure-new-ssh-key/)
- You have a cloud Instance (https://www.scaleway.com/en/virtual-instances/development/) configured with local boot (https://www.scaleway.com/en/docs/enable-disable-localboot-virtual-cloud-servers/) and running on a Linux kernel ≥ 3.10.

Important: WireGuard is currently under development.

Installing and Configuring WireGuard on the server

Note: WireGuard needs kernel modules that are not yet implemented in the kernel. The installation process will install new kernel modules via DKMS.

The installation process is based on Ubuntu. Documentation regarding other platforms is available on the WireGuard website (https://www.wireguard.com/install/).

1 . Connect to your server via SSH (https://www.scaleway.com/docs/create-and-connect-to-your-server/#-Logging-into-the-Instance).

2 . Install Linux kernel headers and WireGuard.

```
$ sudo apt update && aut upgrade -y
$ sudo apt install linux-headers-$(uname --kernel-release) # installs the right kernel headers for your version
$ sudo apt install wireguard
```

Important: To install the Linux kernel headers, your instance must be configured to boot using local boot (https://www.scaleway.com/en/docs/enable-disable-localboot-virtual-cloud-servers/) and running on a Linux kernel ≥ 3.10.

Once WireGuard is installed, you can check that the installation succeeded by running: `wg`, if you get no output it's all good. In order to check that the WireGuard kernel module has loaded you can run `sudo modprobe wireguard`.

Generating Public and Private Keys on the Server

WireGuard relies on a public/private key authentication (asymmetric cryptography;; thus you need to create those keys. They are easily created with the `wg genkey` and `wg pubkey` subcommands.

1 . First let's create a directory to store these keys.

```
# mkdir -p /etc/wireguard/keys
```

2 . Now you can create the public and private key. The creation of the private key is done with `wg genkey` and the public key is generated by piping it into `wg pubkey`. `umask` tells the system to set the permissions of the new files to `600`.

```
# cd /etc/wireguard/keys
# umask 077
# wg genkey | tee privatekey | wg pubkey > publickey
```

Configuring WireGuard Server

The first step is to choose an IP range which will be used by the server. The private IP ranges defined by the RFC 19198 (https://tools.ietf.org/html/rfc1918) are the following:

- `10.0.0.0/8`
- `172.16.0.0/12`
- `192.168.0.0/16`

For this tutorial we will use `192.168.66.0/24` which is inside the `192.168.0.0/16` range. The

server will have the following IP address: `192.168.66.1`. It is also required to choose a port, which will be exposed publicly, for the server to listen on. Here it will be `8999`. Note that the standard documentation port is usually 51820.

Create the file `/etc/wireguard/wg0.conf` with the following content:

```
[Interface]
PrivateKey = <private key of the server>
Address = 192.168.66.1/32
ListenPort = 8999
```

Configuring the Linux, MacOS or Windows WireGuard Client

1 . On Linux you can install WireGuard the same way you did for the server. To install WireGuard on MacOS just run: `brew install wireguard-tools`. You can also use the Mac App Store application (https://apps.apple.com/us/app/wireguard/id1451685025?ls=1&mt=12). To install WireGuard on Windows you can find the executable on the WireGuard installation page (https://www.wireguard.com/install/#installation) but this guide will not cover the Windows use case.

2 . Once installed you will need to create the key pair as well:

```
# mkdir -p /etc/wireguard/keys
# cd /etc/wireguard/keys
# umask 077
# wg genkey | tee privatekey | wg pubkey > publickey
```

3 . After the keys are created, it is time to write the configuration file `/etc/wireguard/wg0.conf`:

```
[Interface]
PrivateKey = <private key of the client>
Address = 192.168.66.2/32
DNS = 1.1.1.1

[Peer]
PublicKey = <public key of the server>
Endpoint = <public ip of the server>:8999
AllowedIPs = 0.0.0.0/0
PersistentKeepalive = 25
```

It is quite similar to the server configuration. The `DNS` line specifies the DNS resolver for the client. The `Endpoint` tells WireGuard where to connect. `AllowedIPs` configures which IP range will be forwarded to the VPN server. In this case, `0.0.0.0/0` means that all the traffic from the client will go through the VPN. If you only want to communicate within the VPN network, you can set `192.168.66.0/24`. `PersistentKeepalive` tells WireGuard to send a UDP packet every 25 seconds, this is useful if you are behind a NAT and you want to keep the connection alive.

Important: If you decide to route all your traffic to the VPN server be sure to do the following on the server:

- Add the following lines in the `[Interface]` section of the server (Replace `ens2` by your main network interface if it is not `ens2`):
- - `PostUp = sysctl -w net.ipv4.ip_forward=1; iptables -A FORWARD -i %i -j ACCEPT; iptables -t nat -A POSTROUTING -o ens2 -j MASQUERADE`
- - `PostDown = iptables -D FORWARD -i %i -j ACCEPT; iptables -t nat -D POSTROUTING -o ens2 -j MASQUERADE`

4 . Now that the client is configured, you need to add the peer configuration to the server. Just add the following to your `/etc/wireguard/wg0.conf` on the server:

```
[Peer]
PublicKey = <public key of the client>
AllowedIPs = 192.168.66.2/32 # the ip address in the VPN network of the cl
ient you just created
```

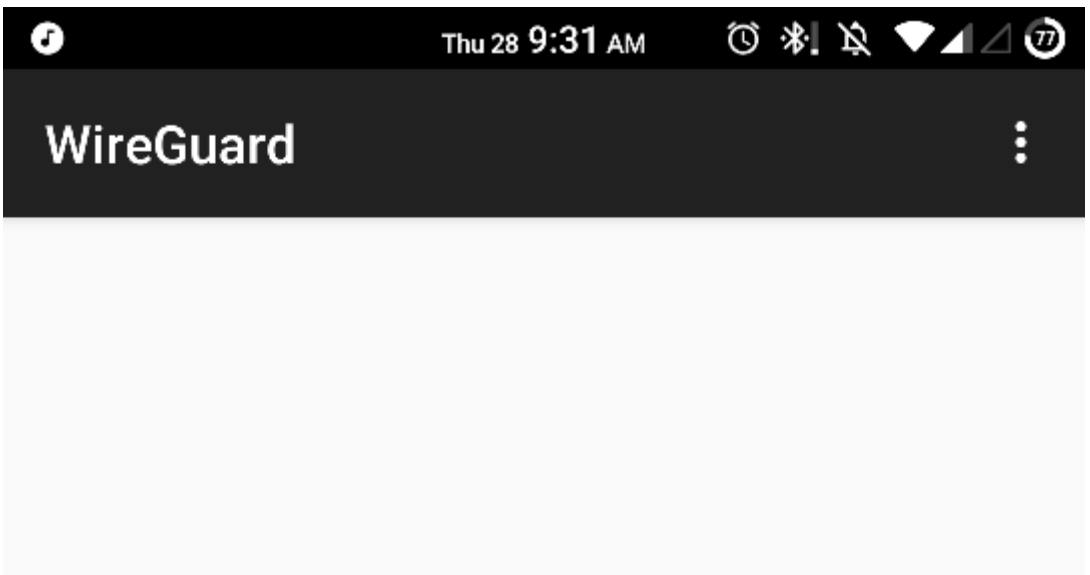
Configuring the Android or iOS WireGuard Client

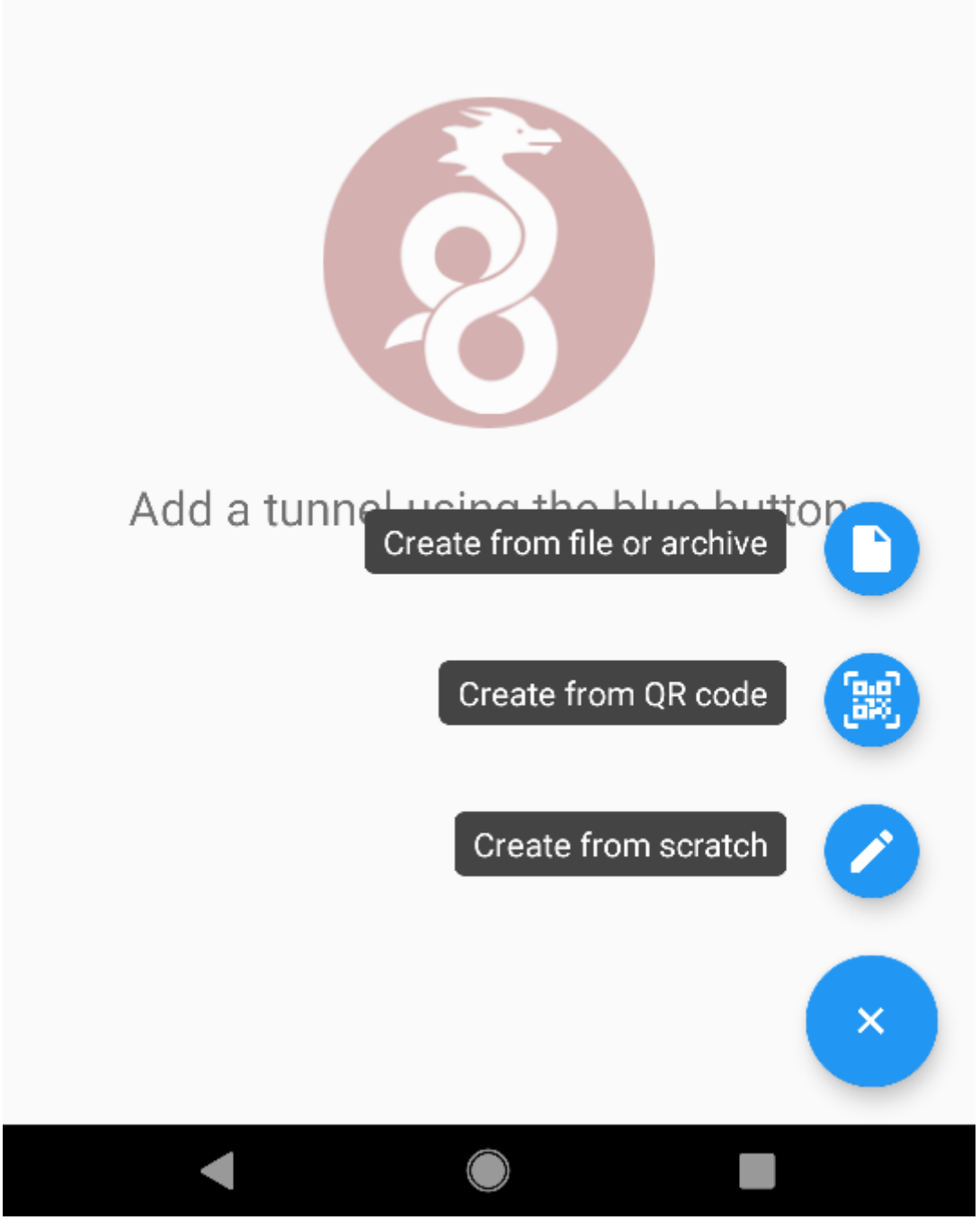
You can download the official WireGuard Android client (https://play.google.com/store/apps/details?id=com.wireguard.android&hl=fr) from the PlayStore and the official WireGuard iOS Client (https://apps.apple.com/us/app/wireguard/id1441195209?ls=1) from the iOS App store (this guide will only cover Android but the steps are the same).

There are two ways to configure the Android or iOS client. The easiest one is to follow the previous part and once the configuration file is done, export it with qrencode (https://linux.die.net/man/1/qrencode) like this: `qrencode -t ansiutf8 < path/to/phone.conf` . Finally scan the generated QR code with the WireGuard application.

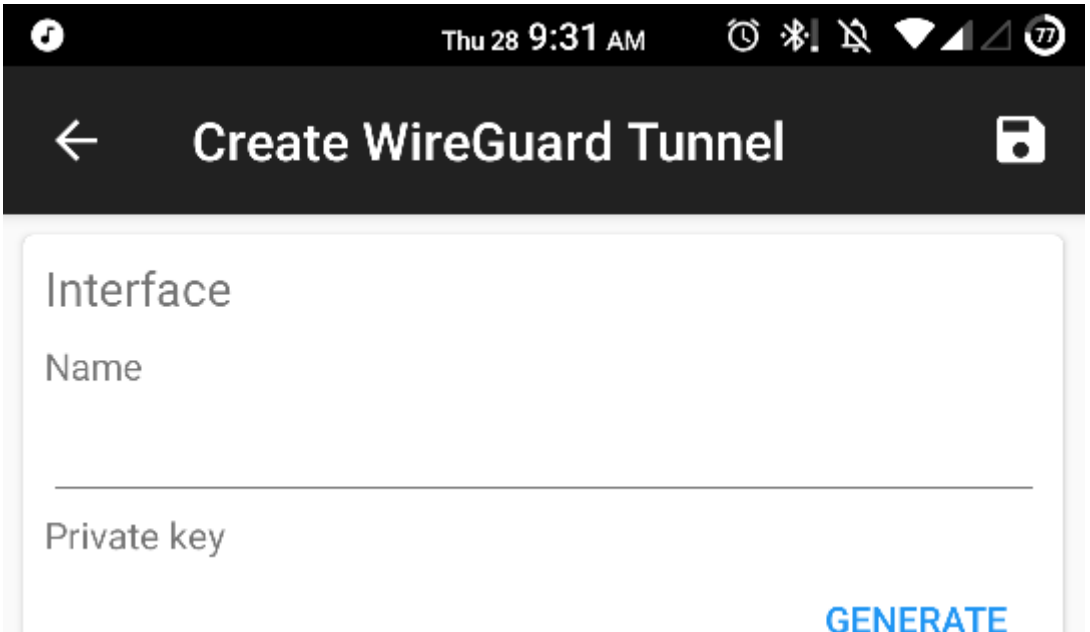
For the second way, follow these steps:

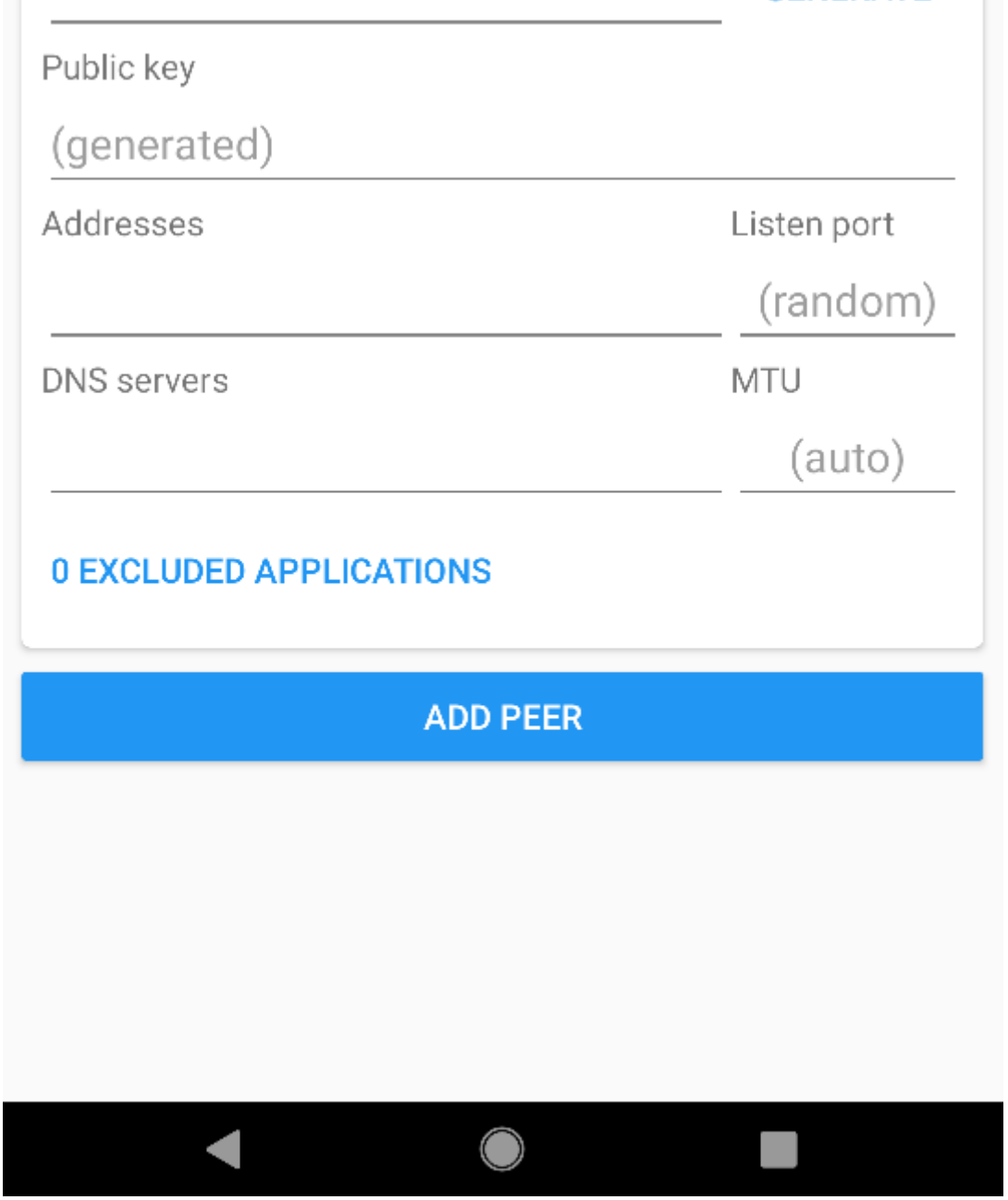
1 . Once the application is downloaded, open the application and click on the + icon and select *Create from scratch*.



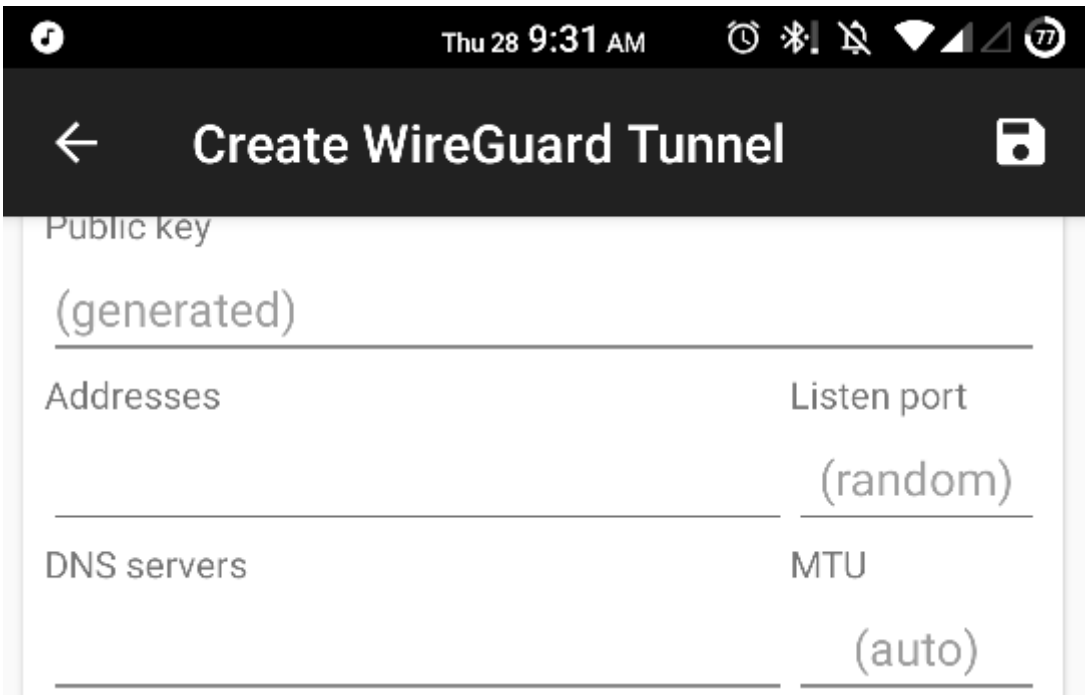


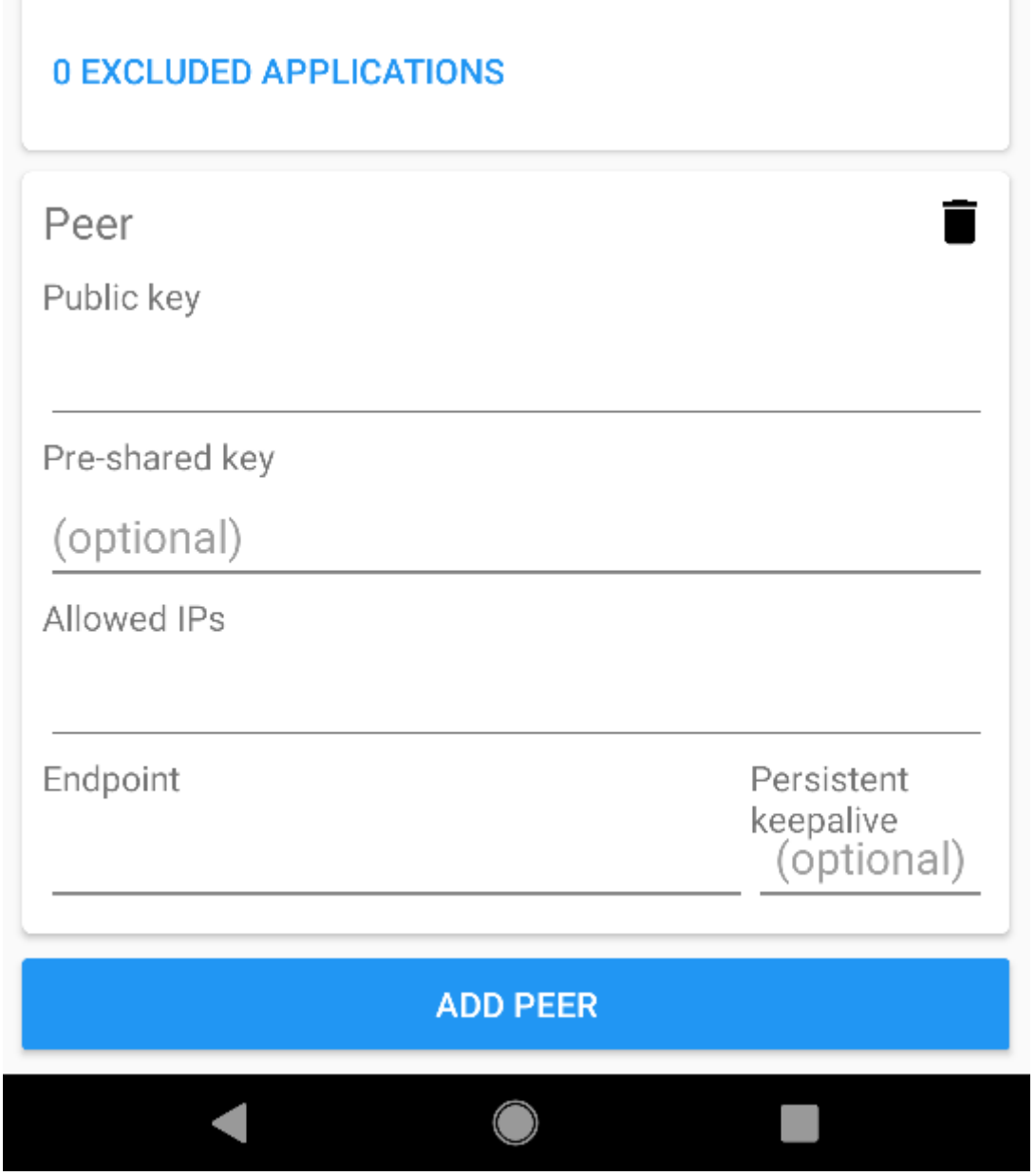
2 . You will need to click on *GENERATE* to generate the key pair (copy the public key in order to use in on the server). The rest is like the Linux client configuration, fill in the addresses, DNS servers and name. Now you will need to add the server as a peer.





3 . Click on *ADD PEER* and add the public key of the server, the public IP of the server and the port on which it is listening. If you decide to route all the traffic through the VPN, please read the *Important* section above.





4 . Finally, add the following to the server's `/etc/wireguard/wg0.conf` :

```
[Peer]
PublicKey = <public key of the android client>
AllowedIPs = 192.168.66.3/32 # the ip address in the VPN network of the cl
ient you just created
```

Launching WireGuard Server

Now that everything is configured, you can launch the WireGuard server with:

```
# wg-quick up wg0
```

And start the client with the same command:

```
# wg-quick up wg0
```

You can also enable the start of WireGuard at boot time with the following command:

```
# systemctl enable wg-quick@wg0.service
```

You can check the connection with the `wg` command (client or server):

```
# wg # on the client
interface: wg0
  public key: <public key of the client>
  private key: (hidden)
  listening port: 57576
  fwmark: 0xca6c

peer: <public key of the server>
  endpoint: <public IP of the server>:8999
  allowed ips: 0.0.0.0/0
  latest handshake: 50 seconds ago
  transfer: 8.35 KiB received, 18.00 KiB sent
  persistent keepalive: every 25 seconds

# ping 192.168.66.1
PING 192.168.66.1 (192.168.66.1) 56(84) bytes of data.
64 bytes from 192.168.66.1: icmp_seq=1 ttl=64 time=3.50 ms
64 bytes from 192.168.66.1: icmp_seq=2 ttl=64 time=4.53 ms
--- 192.168.66.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 3ms
rtt min/avg/max/mdev = 3.499/4.015/4.532/0.520 ms

# curl ifconfig.co
<public IP of the server>
```

As you can see, you can ping the VPN server through the VPN and all your traffic is being routed through the VPN server.

For more information you can check the WireGuard website (https://www.wireguard.com/).

The next tutorial with WireGuard will be about setting up a virtual private network between servers with WireGuard acting as a mesh VPN.

“WireGuard” is a registered trademark of Jason A. Donenfeld (https://www.zx2c4.com/).

Get started now (https://console.scaleway.com/register/)

Company

About (/en/about-us/)

Blog (https://blog.scaleway.com)

Careers (https://careers.scaleway.com/)

Scaleway Dedibox (/en/dedibox/)

Scaleway Datacenter (/en/datacenter/)

(https://careers.scaleway.com/)

Products

Development (/en/virtual-instances/development/)

General Purpose (/en/virtual-instances/general-purpose/)

GPU (/en/gpu-instances/)

Bare Metal Servers (/en/bare-metal-servers/)

Object Storage (/en/object-storage/)

Additional Volumes (/en/bare-metal-instances/#flexible_volumes)

Load Balancer (/en/load-balancer/)

Database (/en/database/)

Container Registry (/en/container-registry/)

Betas & Previews (/en/betas/)

Services

ImageHub (/en/imagehub/)

Resources

Pricing (/en/pricing/)

API (https://developers.scaleway.com/en/)

Developer (/en/developer-tools/)

Terms (/en/terms/)

Privacy policy (/en/privacy-policy/)

Legal notice (/en/legal-notice/)

Our security measures (/en/pdf/PSSI_en.pdf)

Assistance

Chat Room (https://slack.scaleway.com/)

FAQ (/en/faq/)

Help (/en/docs/)

Report abuse (https://console.online.net/en/account/abuses/search)

Status (https://status.scaleway.com)

Technical Assistance (/en/assistance/)

Changelog (/en/changelog)

Customer Testimonials (/en/customer-testimonials/)

(https://slack.scaleway.com/) (https://twitter.com/scaleway/)
(https://github.com/scaleway/) (https://instagram.com/scaleway/)
(https://facebook.com/scaleway/)
(https://www.meetup.com/Meetup-Cloud-Computing-Paris/)

+33 1 84 13 00 50(tel:+33184130050)