

Erstellung Wireguard Server und Client Configs

Auf dem Server:

Erstellung der Server Config, Muss nur einmal gemacht werden.

Schlüsselpaar erstellen.

Sollen mehre IPs erstellt werden empfehle ich für jeden Server + client ein Verzeichnis zu erstellen

Beispiel:

Dazu ein Verzeichnis keys anlegen mit folgenden Unterverzeichnissen je IP (Diese IP Adressen abhe ich nicht, Beispiel

```
.
├── ip_72.44.34.21
│   ├── client
│   └── server
├── ip_72.44.34.22
│   ├── client
│   └── server
```

Nun in das Verzeichnis Server der ersten IP gehen.

```
wg genkey | tee privatekey | wg pubkey > publickey
```

Nun haben wir zwei Schlüsseldateien.

Einmal private Key und publickey

Jetzt sieht das ganze so aus

```
.
├── ip_72.44.34.21
│   ├── client
│   ├── server
│   ├── privatekey
│   └── publickey
├── ip_72.44.34.22
│   ├── client
│   └── server
```

Nun die Config Datei erstellen mit den Schlüsseldateien.

Eine Wireguard Config ist in zwei Teile geteilt.

Einmal Interface das der Lokale Teil für den Computer

(Auch ein Client hat einen lokalen Teil, der dient dazu das Interface bereitzustellen.

Der zweite Teil sind die Peers, es können auch mehrere Peers eingetragen sein. Aber wir erstellen hier für jede IP eine eigene config. Den Publickey für den Peer haben wir noch nicht.

```
nano ~/keys/ip_27.44.34.21/server/ip27443421server.conf
```

```
[Interface]
```

```
Address = 172.16.0.1 # oder irgendein anderes privates Netz from RFC1918 (privates Netz) 10.10.0.x oder 192.168.0.100 oder oder egal. Wichtig, auf einem Computer dürfen die entze nicht gleich sein
```

```
PrivateKey = <private_key_vom_server>
```

```
ListenPort = 51820 # or jeder andere port der beliebt
```

```
[Peer]
```

```
PublicKey = <public_key_vom_client_also_der_derdie_ip_bekommen_soll>
```

```
AllowedIPs = publicip_v4/32,ipv6 mit subnet # Here tragen wir unsere PUBLIC IPS ein, die auf der PEER Seite zur Verfügung stehen sollen
```

Ein Beispiel, diese Keys wurden NIE verwendet, also viel Spaß damit

Inhalt Privatekey Server

```
mJn1IWnRFTze3wojk3a+d5TsPSOolRqt3dfN7ekgukU=
```

Inhalt Publickey Server

```
/amPOUXRsKhzW1QsLvQ/7UjchU1oFfkWYZbZj/mWtxA=
```

```
[Interface]
```

```
Address = 192.168.100.1
```

```
PrivateKey = mJn1IWnRFTze3wojk3a+d5TsPSOolRqt3dfN7ekgukU=
```

```
ListenPort = 51820
```

```
[Peer]
```

```
PublicKey = <haben_wir_noch_nicht_leer_lassen>
AllowedIPs = 27.44.34.21/32, 3a01:4d8:xxxx:8106::/64
```

Nun die Client Schlüssel erstellen konfig erstellen
Dazu ins client Verzeichnis gehen und wieder

```
wg genkey | tee privatekey | wg pubkey > publickey
```

Nun haben wir auch dort im Verzeichnis Client auch zwei Schlüssel

```
├── ip_72.44.34.21
│   ├── client
│   │   ├── privatekey
│   │   └── publickey
│   └── server
│       ├── privatekey
│       └── publickey
├── ip_72.44.34.22
│   ├── client
│   └── server
```

Unserer Privatekey
0ls0rjpsSpzBPD0T+Quv/3GWGe1nPKrNfXJa80wUY0E=

Unserer Public Key
2gWEqeQxoGa9hI94UvV4trVVdiG1G7sq0iLr6W5ymUE=

Nun in der Serverconfig den Public key nachtragen unter Peers

```
nano ~/keys/ip_27.44.34.21/server/ip27443421server.conf
```

Nun im peer bereich public key "haben wir noch nicht erstezen" Platzhalter mit dem öffentlichem Schlüssel vom Client ersetzen

```
....
[Peer]
PublicKey = 2gWEqeQxoGa9hI94UvV4trVVdiG1G7sq0iLr6W5ymUE=
AllowedIPs = 27.44.34.21/32, 3a01:4d8:xxxx:8106::/64
```

Nun die Client Config erstellen

```
nano ~/keys/ip_27.44.34.21/client/ip27443421client.conf
```

[Interface]

Address = publicip_v4/32,ipv6 OHNE subnet

PrivateKey = <private_key_vom_client>

ListenPort = 51820 # irgendeiner darf nur auf dem client nicht doppelt sein. Denn dieser Port wird ja nie benutzt. Denn wir verbinden uns ja zum Server und nicht umgekehrt

[Peer]

PublicKey = <public_key_vom_server_also_der_derdie_ip_bekommen_soll>

Endpoint: ip_adresse_vom_server(Die die der Server fest hat) und den Listen Port aus der Server conf

AllowedIPs = 0.0.0.0/0,::/0 Der erste Wert heißt als standard Gateway für ipv4, der zweite standard gateway für ipv6. Spricht route alles

Beispiel:

[Interface]

Address = 27.44.34.21

PrivateKey = 0ls0rjpsSpzBPDoT+Quv/3GWGe1nPKrNfXJa80wUY0E=

ListenPort = 51820

[Peer]

PublicKey = /amP0UXRsKhzW1QsLvQ/7UjchU1oFfkWYZbZj/mWtxA=

Endpoint: 27.44.34.20:51820

AllowedIPs = 0.0.0.0/0,::/0

Nun sind die Configs fertig und können ausgerollt werden.

Dazu die Server config nach /etc/wireguard kopieren

```
cp ~/keys/ip_27.44.34.21/server/ip27443421server.conf /etc/wireguard
```

Nun den Tunnel beim start aktiv schalten

```
systemctl enable wg-quick@ip27443421server
```

```
systemctl start wg-quick@ip27443421server
```

Wenn ein Loadbalancer im einsatz ist, die Schlüssel und configs auch auf dem Slaveloadbalancer kopieren.

Dort im /etc/wireguard die config auch reinkopieren und auch dort mit systemctl enable den Dienst in die Autostart packen und danach starten

Nun die Client config auf den Client kopieren, entweder den text per Copy und paste übermitteln in eine neue Datei auf den client oder per scp.

Diese muss dann auch in

```
/etc/wireguard/ip27443421client.conf
```

Nun diese auch beim starten ausführen

```
systemctl enable wg-quick@ip27443421client  
systemctl start wg-quick@ip27443421client
```

Fertig.

Version #5

Erstellt: 15 November 2022 06:55:07 von Admin

Zuletzt aktualisiert: 1 Oktober 2023 08:11:36 von Admin