Setting up a wireguard VPN

Followed these directions as a general guide:
https://technofaq.org/posts/2017/10/how-to-setup-wireguard-vpn-on-your-debian-gnulinux-server-with-ipv6-support/

Install wireguard software
Installation on the server and client is the same. You might not need to install linux-headers. dpkg --get-selections|grep linux-headers-$(uname -r) to see if you already have it - look for 'install' in the output.

```
  sudo add-apt-repository ppa:wireguard/wireguard
  sudo apt update
  sudo apt install wireguard-dkms wireguard-tools linux-headers-$(uname -r)
```

Generate keys:

```
  sudo umask 077
  sudo wg genkey | tee server_private_key | wg pubkey > server_public_key
  sudo wg genkey | tee client_private_key | wg pubkey > client_public_key
```

Repeat the last command for each new client, using different key file names. The single-line contents of these key files will be used for configuration.

Configure the server
Create /etc/wireguard/wg0.conf (in the PostUp/PostDown rules, eth0 is the internet-facing interface. Yours might be different, like enp3s0, e.g.)

```
$ cat wg0.conf
[Interface]
Address = 10.200.200.1/24
SaveConfig = true
PostUp = iptables -A FORWARD -i wg0 -j ACCEPT; iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
PostDown = iptables -D FORWARD -i wg0 -j ACCEPT; iptables -t nat -D POSTROUTING -o eth0 -j MASQUERADE
ListenPort = 51820
PrivateKey = <server_private_key>
```

Did not add a [Peer] section to the server's wg0.conf as shown in the Debian guide. The server does that when we add clients to the server's configuration using the command line (see below).

On the server (may have to 'apt install ufw' and 'ufw enable'):

```
  sudo ufw allow 51820/udp
```

Bring up the wg0 interface on the server and check its status:

```
  sudo wg-quick up wg0 && ifconfig wg0
```

Should see something like this:

```
wg0       Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:10.200.200.1  P-t-P:10.200.200.1  Mask:255.255.255.0
          UP POINTOPOINT RUNNING NOARP  MTU:1420  Metric:1
          RX packets:216588 errors:78 dropped:0 overruns:0 frame:78
          TX packets:363965 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:49769456 (49.7 MB)  TX bytes:505690184 (505.6 MB)
```

Add a client to the server's configuration

```
sudo wg set wg0 peer <client_public_key> allowed-ips 10.200.200.2
```

Configure the client

The IP address (10.200.200.2) in the command above must be entered on the 'Address =' line in the client's /etc/wireguard/wg0-client.conf file (named mine wg0-client.conf, not client.conf as shown in the guide):

```
$ cat wg0-client.conf
[Interface]
Address = 10.200.200.2/24
PrivateKey = <client_private_key>

[Peer]
PublicKey = <server_public_key>
Endpoint = <public IP address of server>:51820
AllowedIPs = 0.0.0.0/0
PersistentKeepalive = 21
```

Bring up the client's wg0-client interface and check its status:

```
sudo wg-quick up wg0-client && ifconfig wg0-client
```

Should see:

```
wg0-client Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:10.200.200.2  P-t-P:10.200.200.2  Mask:255.255.255.0
          UP POINTOPOINT RUNNING NOARP  MTU:1420  Metric:1
          RX packets:38711 errors:0 dropped:0 overruns:0 frame:0
          TX packets:25016 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:54378188 (54.3 MB)  TX bytes:4810872 (4.8 MB)
```

Test connectivity (might need to install lynx → sudo apt install lynx):

```
ping 1.1.1.1
ping www.google.com
lynx https://www.wireguard.com
```

Run this command to ensure that your public facing IP address is correct:

```
curl ipv4.icanhazip.com
```

Enable autostart of the wireguard VPN

On the server:

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

On the client(s):

```
sudo systemctl enable wg-quick@wg0-client.service
```

** The configuration files in /etc/wireguard should be mode 0600. **

Use ‘sudo wg show’ to get configuration info (server or client).

Wireguard website: https://www.wireguard.com/
Wireguard git repository: https://git.zx2c4.com/

Android App:
https://bit.ly/2PDCVRL

Copy a file containing a wg0-client.conf (like the one above) configuration to your phone, start the app, click the big ‘+’ to create a tunnel and import the file. Switch the slider to the ‘On’ position. It should connect immediately.

Apparently there’s also an iOS app:
https://git.zx2c4.com/wireguard-ios/about/
but have no idea how to install this.

There is currently no wireguard client for Windows, but who cares. Rumor has it that one will be available soon, however.

Setting up Linux *clients* to use wireguard could be automated with an Ansible playbook or a shell script.

The script that be cloned from:
   https://github.com/benrb3/wireguard-script.git
was tested on an ubuntu-18.04 box and worked well. One of the advantages of using this script is that you don’t have to bother with generating key files for clients and copying their contents to configuration files. The keys are created and dumped into the client configuration files inside the script itself.