### Configure the [Interface] section in the **server's** configuration `wg0.conf` file:

* Address: The IP address and subnet mask to be assigned to the WireGuard server, in CIDR notation.
* ListenPort: The port that the server should listen on for incoming traffic.
* PrivateKey: The server's private key.
* PostUp: The terminal commands that should run when the server is brought up. This section should contain the commands to temporarily enable NAT on the machine running the server.
* PostDown: The terminal commands that should run when the server is brought down. This should remove the temporary NAT configuration that was set up for use while the server was running.

The server's configuration should look like this:

[Interface]

PrivateKey = [your server's private key]

Address = [Wireguard-internal IPs of the server, e.g. 10.6.0.1/24, fd08:4711::1/64]

ListenPort = 51820

PostUp = nft add table ip wireguard; nft add chain ip wireguard wireguard\_chain {type nat hook postrouting priority srcnat\; policy accept\;}; nft add rule ip wireguard wireguard\_chain ip saddr 10.6.0.0/24 masquerade; nft add table ip6 wireguard; nft add chain ip6 wireguard wireguard\_chain {type nat hook postrouting priority srcnat\; policy accept\;}; nft add rule ip6 wireguard wireguard\_chain ip6 saddr fd08:4711::/64 masquerade

PostDown = nft delete table ip wireguard; nft delete table ip6 wireguard

### Add peers to the server's configuration:

For each peer, add:

* PublicKey: The peer's public key.
* AllowedIPs: The source IP address/es from which to accept incoming packets from this client.

The section being added should look like this:

[Peer]

PublicKey = [your client's public key]  
AllowedIPs = 10.6.0.2/32, fd08:4711::2/128

### Enable IP forwarding on the device running the server:

Uncomment the following lines in the `/etc/sysctl.conf` file:

#net.ipv4.ip\_forward=1

#net.ipv6.conf.all.forwarding=1

Run the following command to apply the configuration settings from `/etc/sysctl.conf` without requiring a system reboot (normally this would require rebooting the system).

sudo sysctl -p

### Configure the [Interface] section in the **client's** configuration:

PrivateKey: Your client's private key.

Address: The static IP address you want the client to use in the VPN network. It should match the entry in this client's peer configuration within the server configuration file and be within the subnet specified in the server's configuration.

DNS: The default gateway for DNS requests. This should be the IP address of the VPN server (within the VPN subnet, so in this case `10.6.0.1`), to ensure that DNS requests are tunneled through the server.

*\*Currently, when the DNS server is set to the VPN server's IP address, the client is not able to access the internet. As a temporary workaround, you can set the DNS server to another option such as 1.1.1.1 (Cloudflare), as the DNS requests are still routed through the VPN, however the VPN server does not make the requests itself. The most important thing is to use a service like dnsleaktest.com first to ensure that your DNS requests are not being made from your client's IP address.*

This section should look like this:

[Interface]

PrivateKey = [your client's private key]

Address = 10.6.0.2/32, fd08:4711::2/128

DNS = 10.6.0.1

### Add the server as a peer in the client's configuration:

AllowedIPs: This should be set to `0.0.0.0/0` and `::/0` to if you want to tunnel all internet traffic through the VPN, to ensure your client will accept packets (via the tunnel) with any source IP address.

Endpoint: The socket (public IP address and port of your server machine).

PersistentKeepalive: The time interval (in seconds) between transmission of keepalive packets.

This section should look like this:

[Peer]

PublicKey = [your server's public key]

AllowedIPs = 0.0.0.0/0, ::/0

Endpoint: 110.20.180.153:51820

PersistentKeepalive = 25