Wireguard VPN Notes - Ron Swift

This is an overview of setting up a wireguard vpn server and client (peer) using a Raspberry Pi as the server on your home network and clients running Android, IOS, Windows or a Linux device.
The assumptions are:
Pi server is running Debian OS like Raspbian Buster or Ubuntu 20.04 and it has a static ip address configured.
Internal home network is using 192.168.1.0/24 private network.
Home network has a dynamic external ip address so you will need a dynamic host naming application like no-ip.com to provide a name for your home network like myhome.ddns.net
Home router can be used to port forward to your Pi wireguard server.
Add ip packet forwarding on the wireguard Pi server: sudo nano /etc/sysctl.conf #uncomment net.ipv4.ip_forward=1
Wireguard server port is 51820 udp and it's allowed in firewall:
sudo ufw allow 51820/udp #if using ufw front-end to iptables

Install
First verify if wireguard is already installed by running this command "which wg" (without quotes). If you get a response it is already installed so you can skip to Generate Keys.

Ubuntu <= 20.04	Add to repository: sudo add-apt-repository ppa:wireguard/wireguard sudo apt update sudo apt install wireguard
Raspbian/Debian:	sudo sh -c "echo 'deb http://deb.debian.org/debian/ unstable main' >> /etc/apt/sources.list.d/unstable.list" sudo sh -c "printf 'Package: *\nPin: release a=unstable\nPin-Priority: 90\n' >> /etc/apt/preferences.d/limit-unstable" sudo apt install wireguard

Generate Keys
Umask 077
wg genkey | sudo tee /etc/wireguard/privatekey | wg publickey | sudo tee /etc/wireguatd/publickey
Verify keys: sudo ls /etc/wireguard #Should see both privatekey and publickey
sudo cat /etc/wireguard/privatekey #Copy result to text fille

sudo cat /et/wireguard/publickey #Copy result to text file

Create wg0.conf file
ip -o -4 route show to default | awk '{print $5}' ##shows network interface device i.e eth0
sudo nano /etc/wireguard/wg0.conf ##Sample shown for server

[Interface]
Address = 10.0.0.1/24 #wireguard network which must be different from Lan addresses
Privatekey = aBcDeEfFo0ikxxx=
ListenPort = 51820
PostUP = iptables -A forward -i %: -j ACCEPT; iptables -t nat -A POSTROUTING eth0 -j MASQUERADE
PostDOWN = iptables -D forward -i %: -j ACCEPT; iptables -t nat -D POSTROUTING eth0 -j MASQUERADE

[Peer] #need one for each connecting client
PublicKey = xxxxxxxxxx245= #For the connecting client
AllowedIPs = 10.0.0.2/32, 192.168.1.0/24 #Next address in server range and private LAN access

On the remote client (Peer) that will access the server wireguard needs to be installed either via a GUI app (i.e. Android, IOS, Windows) or for Linux on the command line as shown above for the server.
Private and Public Keys need to be generated for the client(Peer)
The sample client config file would look like this:
[Interface]
PrivateKey = zxde63452rg= #Generated on the client
Address = 10.0.0.2/24 #second address in server's private range

[Peer] #To the server
PublicKey = xxxxxxxxxx666= #Server publickey
Endpoint = myhome.ddns.net:51820 #External address and port to server
AllowedIPs = 10.0.0.1/32, 192.168.1.0/24 #server ip address and LAN range when connected

On the wireguard Pi server set the client's ip address in the private VPN ip range
sudo wg set wg0 peer xxxxxxxxxx245=(publickey) allowed-ips 10.0.0.2 #first peer client address
sudo wg set wg0 peer xxxxxxx777xxx=(publickey) allowed-ips 10.0.0.3 #second peer client
sudo wg set wg0 peer xxxxxx98uixx=(publickey) allowed-ips 10.0.0.3 remove #remove peer client when needed
sudo chmod 600 /etc/wireguard/wg0.conf
sudo wg-quick up wg0 #check that it works and loads the wg0 device
sudo systemctl enable wg-quick wg-quick@wg0 #Set Wireguard up to load at startup