Full mesh secure & scalable VPN network with WireGuard v1.0.202 & Ubuntu

On STC OpenStack Cloud

Taher A. Bahashwan

Cloud Advisory Expert

STC Solution

@taher9990

Oct 2020

https://github.com/taher9990

Contents

Disclaimer

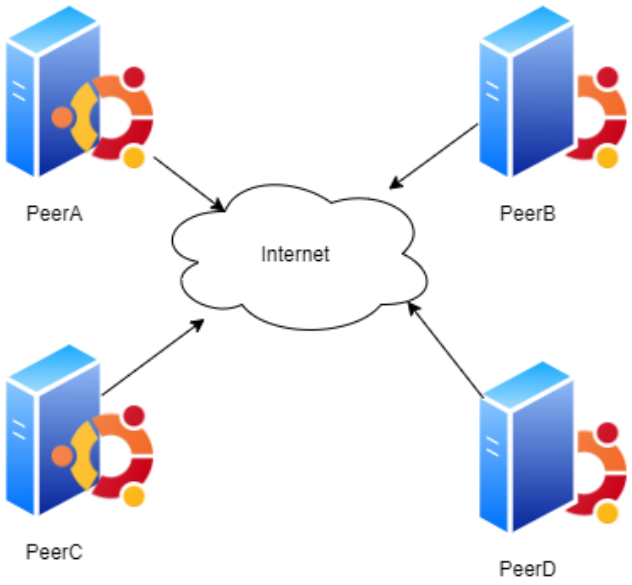
This article is made for educational and testing purposes, and you might find few settings that are not made for production, please do your full testing and follow your organizations best practices and standards along with the steps and guides in this document to get a full complete working solution.

WARNING --- IPs Misuse

All Public IPs that we use in this article are randomly selected from STC Public Cloud, and they will be deleted from our Cloud tenant after we complete the test, so you are not allowed to use or conduct any activity in to these IPs, if activity identified it will be considered as criminal activity, STC Cloud personnel have the right to take legal actions against you or your organization.

Network Topology

Public IP: 95.177.218.228
Tunnel IP: 10.200.200.1/32
Wireguard Port: 44444



Public IP: 95.177.218.232
Tunnel IP: 10.200.200.2/32
Wireguard Port: 44444

Public IP: 95.177.218.233
Tunnel IP: 10.200.200.3/32
Wireguard Port: 44444

Public IP: 95.177.218.234
Tunnel IP: 10.200.200.4/32
Wireguard Port: 44444

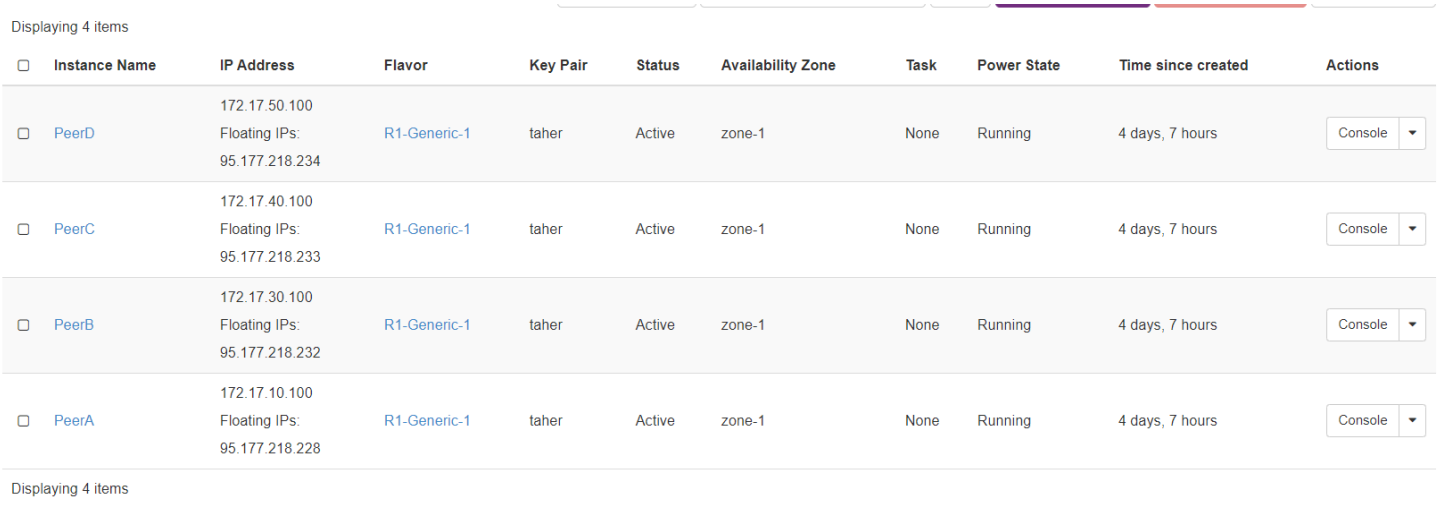
Successful Tunnel Status

Setup and Prepare Ubuntu Servers

Server Specs

```
CPU: 2 Memory: 4 GB, HDD: 30 GB
          Icon name: computer-vm
            Chassis: vm
    Virtualization: kvm
  Operating System: Ubuntu 18.04.3 LTS
            Kernel: Linux 5.0.0-31-generic
      Architecture: x86-64
```

Below is a screenshot for the servers hosted in STC Public Cloud



Displaying 4 items

Instance Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
PeerD	172.17.50.100 Floating IPs: 95.177.218.234	R1-Generic-1	taher	Active	zone-1	None	Running	4 days, 7 hours	Console
PeerC	172.17.40.100 Floating IPs: 95.177.218.233	R1-Generic-1	taher	Active	zone-1	None	Running	4 days, 7 hours	Console
PeerB	172.17.30.100 Floating IPs: 95.177.218.232	R1-Generic-1	taher	Active	zone-1	None	Running	4 days, 7 hours	Console
PeerA	172.17.10.100 Floating IPs: 95.177.218.228	R1-Generic-1	taher	Active	zone-1	None	Running	4 days, 7 hours	Console

Displaying 4 items

Servers & Tunnel IPs

PeerANetwork IPs:

Public IP: 95.177.218.228

Tunnel IP: 10.200.200.1/32

Wireguard Port: 44444

PeerB-Network IPs:

Public IP: 95.177.218.232

Tunnel IP: 10.200.200.2/32

Wireguard Port: 44444

PeerC-Network IPs:

Public IP: 95.177.218.233

Tunnel IP: 10.200.200.3/32

Wireguard Port: 44444

PeerD-Network IPs:

Public IP: 95.177.218.234

Tunnel IP: 10.200.200.4/32

Wireguard Port: 44444

Common System configurations applied on all servers

```
sudo su –
timedatectl set-timezone Asia/Riyadh
sudo add-apt-repository ppa:wireguard/wireguard
sudo apt-get update
sudo apt-get install wireguard –y

sudo su -
hostnamectl set-hostname PeerA
systemctl restart systemd-hostnamed
 su ubuntu
sudo hostnamectl set-hostname PeerA
sudo systemctl restart systemd-hostnamed
sudo su -
```

Note: Change PeerA with the name of each server e.g PeerB

First Method: Manual WireGuard setup

Generate Pre-shared key to use it on all servers:

```
wg genpsk > psk
```

Take this key and save it into each server

Common WireGuard configurations applied on all servers

1- Generate Private & public Key on all servers

```
umask 077; wg genkey | tee privatekey | wg pubkey > publickey
```

2- Add generated Pre shared key to all server

```
touch psk
cat >psk<<EOF
RrDNbpkGT6//9eU4eMa1cJmRaVNdnBcVc6I2oQBFvBY=
EOF
```

3- Add the generated public key only (publickey) in step 1 and save it into each server

```
touch PeerA-PublicKey
cat>PeerA-PublicKey<<EOF
oEGg3q2bx+vPgCoEFqkLzLKZC/oZdwG116/gIOZ4gxU=
EOF
touch PeerB-PublicKey
cat>PeerB-PublicKey<<EOF
zszrQrD5R1HtHkLY1ubF3dVlk7vMUP0Jud6GlaT84G4=
EOF
touch PeerC-PublicKey
cat>PeerC-PublicKey<<EOF
pmLUBZE0m0CMstOn0zea+kIv+E4EvrQcS6khYScbc0Q=
EOF
touch PeerD-PublicKey
cat>PeerD-PublicKey<<EOF
2SSJ0ADvzxhnY5Afrdij+6XJA3/SFCsBT2OgVgO0UzM=
EOF
```

4- Enable WireGuard on all servers so that it can automatically run when the server reboots

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

5- Enable routing on all servers

```
cat >> /etc/sysctl.conf << EOF
net.ipv4.ip_forward = 1
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
EOF
sysctl -p
echo 1 > /proc/sys/net/ipv4/ip_forward
```

WireGuard Configurations for each server

Server: PeerA Configurations

```
cat >>/etc/wireguard/wg0.conf<<EOF
[Interface]
Address = 10.200.200.1/24
SaveConfig = true
ListenPort = 44444
PrivateKey = /root/privatekey

[Peer]
PublicKey = /root/PeerB-PublicKey
PresharedKey = /root/psk
AllowedIPs = 10.200.200.2/32
Endpoint = 95.177.218.232:44444

[Peer]
PublicKey = /root/PeerC-PublicKey
PresharedKey = /root/psk
AllowedIPs = 10.200.200.3/32
Endpoint = 95.177.218.233:44444

[Peer]
PublicKey = /root/PeerD-PublicKey
PresharedKey = /root/psk
AllowedIPs = 10.200.200.4/32
Endpoint = 95.177.218.234:44444
```

```
EOF

chown -v root:root /etc/wireguard/wg0.conf
chmod -v 600 /etc/wireguard/wg0.conf

wg-quick up wg0
systemctl enable wg-quick@wg0.service
wg-quick save wg0
cp /etc/wireguard/wg0.conf /tmp/$(uname -n)-wireguard-$(date
+%Y-%m-%d_%H:%M:%S).conf
```

Server: PeerB Configurations

```
touch /etc/wireguard/wg0.conf

cat >>/etc/wireguard/wg0.conf<<EOF

[Interface]
Address = 10.200.200.2/24
SaveConfig = true
ListenPort = 44444
PrivateKey = /root/privatekey

[Peer]
PublicKey = /root/PeerA-PublicKey
PresharedKey = /root/psk
AllowedIPs = 10.200.200.1/32
Endpoint = 95.177.218.228:44444

[Peer]
PublicKey = /root/PeerC-PublicKey
PresharedKey = /root/psk
AllowedIPs = 10.200.200.3/32
Endpoint = 95.177.218.233:44444

[Peer]
PublicKey = /root/PeerD-PublicKey
```

```
PresharedKey = /root/psk
AllowedIPs = 10.200.200.4/32
Endpoint = 95.177.218.234:44444


EOF

chown -v root:root /etc/wireguard/wg0.conf
chmod -v 600 /etc/wireguard/wg0.conf

wg-quick up wg0
systemctl enable wg-quick@wg0.service
wg-quick save wg0
cp /etc/wireguard/wg0.conf /tmp/$(uname -n)-wireguard-$(date
+%Y-%m-%d_%H:%M:%S).conf
```

Server: PeerC Configurations

```
touch etc/wireguard/wg0.conf
cat >>/etc/wireguard/wg0.conf<<EOF
[Interface]
Address = 10.200.200.3/24
SaveConfig = true
ListenPort = 44444
PrivateKey = /root/privatekey

[Peer]
PublicKey = /root/PeerA-PublicKey
PresharedKey = /root/psk
AllowedIPs = 10.200.200.1/32
Endpoint = 95.177.218.228:44444

[Peer]
PublicKey = /root/PeerB-PublicKey
PresharedKey = /root/psk
AllowedIPs = 10.200.200.2/32
Endpoint = 95.177.218.232:44444
```

```
[Peer]
PublicKey = /root/PeerD-PublicKey
PresharedKey = /root/psk
AllowedIPs = 10.200.200.4/32
Endpoint = 95.177.218.234:44444


EOF
chown -v root:root /etc/wireguard/wg0.conf
chmod -v 600 /etc/wireguard/wg0.conf
wg-quick up wg0
systemctl enable wg-quick@wg0.service
wg-quick save wg0
cp /etc/wireguard/wg0.conf /tmp/$(uname -n)-wireguard-$(date
+%Y-%m-%d_%H:%M:%S).conf
```

Server: PeerD Configurations

```
touch etc/wireguard/wg0.conf
cat >>/etc/wireguard/wg0.conf<<EOF
[Interface]
Address = 10.200.200.4/24
SaveConfig = true
ListenPort = 44444
PrivateKey = /root/privatekey

[Peer]
PublicKey = /root/PeerA-PublicKey
PresharedKey = /root/psk
AllowedIPs = 10.200.200.1/32
Endpoint = 95.177.218.228:44444

[Peer]
PublicKey = /root/PeerB-PublicKey
PresharedKey = /root/psk
AllowedIPs = 10.200.200.2/32
Endpoint = 95.177.218.232:44444
```

```
[Peer]
PublicKey = /root/PeerC-PublicKey
PresharedKey = /root/psk
AllowedIPs = 10.200.200.3/32
Endpoint = 95.177.218.233:44444

EOF
chown -v root:root /etc/wireguard/wg0.conf
chmod -v 600 /etc/wireguard/wg0.conf
wg-quick up wg0
systemctl enable wg-quick@wg0.service
wg-quick save wg0
cp /etc/wireguard/wg0.conf /tmp/$(uname -n)-wireguard-$(date
+%Y-%m-%d_%H:%M:%S).conf
```

Configurations Completed

Till here all the configurations completed, in the next section will explain how to add addition Peer or server to the existing network.

Adding additional Server in the future

To add additional server in the future you need only to apply below few steps:

1- Apply the common system configurations as explained in under common system config section in this doc.

2- Generate new private and public Key for this server

```
umask 077; wg genkey | tee privatekey | wg pubkey > publickey
```

3- Copy the Pre shared key and Public Key to the new server.

```
touch psk
cat >psk<<EOF
RrDNbpkGT6//9eU4eMa1cJmRaVNdnBcVc6I2oQBFvBY=
EOF
```

To copy the public keys of all other servers to this new server please refer to section “Common WireGuard configurations applied on all servers” step number 3.

4- Now add the new configurations for the new server which will include all servers.

```
touch etc/wireguard/wg0.conf
cat >>/etc/wireguard/wg0.conf<<EOF
[Interface]
Address = 10.200.200.5/24
SaveConfig = true
ListenPort = 44444
PrivateKey = /root/privatekey
```

```
[Peer]
PublicKey = /root/PeerA-PublicKey
PresharedKey = /root/psk
AllowedIPs = 10.200.200.1/32
Endpoint = 95.177.218.228:44444

[Peer]
PublicKey = /root/PeerB-PublicKey
PresharedKey = /root/psk
AllowedIPs = 10.200.200.2/32
Endpoint = 95.177.218.232:44444

[Peer]
PublicKey = /root/PeerC-PublicKey
PresharedKey = /root/psk
AllowedIPs = 10.200.200.3/32
Endpoint = 95.177.218.233:44444

[Peer]
PublicKey = /root/PeerD-PublicKey
PresharedKey = /root/psk
AllowedIPs = 10.200.200.4/32
Endpoint = 95.177.218.234:44444

EOF
chown -v root:root /etc/wireguard/wg0.conf
chmod -v 600 /etc/wireguard/wg0.conf
wg-quick up wg0
systemctl enable wg-quick@wg0.service
wg-quick save wg0
cp /etc/wireguard/wg0.conf /tmp/$(uname -n)-wireguard-
$(date +%Y-%m-%d_%H:%M:%S).conf
```

Note: All below steps to be done on the other existing server

5- Copy the new server public Key to all other servers.

```
cat publickey
```

Copy the content of this file and go to each server and add it, you can do that by below commands

```
touch PeerF-PublicKey
cat>PeerF-PublicKey<<EOF
```

```
<Here put the new key that generated in step 2 above
publickey>
EOF
```

6- Now go to each server and execute below command

```
wg set wg0 peer PeerF-PublicKey preshared-key psk
allowed-ips 10.200.200.5/32 endpoint 95.177.218.235:44444
wg-quick save wg0
```

Second Method: Auto Generation for WireGuard

You can use this configurator method in any machine not restricted to peers or servers, we just need to generate the configurations and copy them to peers, so it does not matter which server you use as long as it is Ubuntu

Installing Python3 & Downloading the configurator script

```
sudo apt install python3-pip
sudo pip3 install -r requirements.txt
git clone https://github.com/taher9990/WireGuard-full-
mesh-networking.git
cd wireguard-mesh-configurator/
```

Accessing the configurator

```
python3 wireguard_mesh_configurator.py interactive
or
python3 wireguard_mesh_configurator.py int
```

Adding Peers

```
NewProfile

AddPeer
```

Now add the details for PeerA, then add the second peer and 3rd and so on so fourth

```
root@PeerA:~/wireguard-mesh-configurator# python3 wireguard_mesh_configurator.py int
WireGuard Mesh Configurator 1.2.0
(C) 2018-2019 K4YT3X
Licensed under GNU GPL v3
[WGC]> AddPeer
[?] USER: Address (leave empty if client only) [IP/CIDR]: 10.200.200.1/32
95.177.218.228
[?] USER: Listen port (leave empty for client) [1-65535]: 44444
[?] USER: Private key (leave empty for auto generation):
[?] USER: Keep alive? [y/N]: y
[?] USER: Alias (optional): PeerA
[?] USER: Description (optional): PeerA
[+] INFO: PeerA information summary:
Description: PeerA
Address: 10.200.200.1/32
Public Address: 95.177.218.228
Listen Port: 44444
Private Key: YG1l/KNPmrt0pvNq2wGLWlBp2hg9k7tX/CV+IVzaelo=
Keep Alive: True
```

Second Peer “PeerB”

```
[WGC]> AddPeer
[?] USER: Address (leave empty if client only) [IP/CIDR]: 10.200.200.2/32
[?] USER: Public address (leave empty if client only) [IP|FQDN]: 95.177.218.232
[?] USER: Listen port (leave empty for client) [1-65535]: 44444
[?] USER: Private key (leave empty for auto generation):
[?] USER: Keep alive? [y/N]:

PeerB
[?] USER: Description (optional): PeerB
[+] INFO: PeerB information summary:
Description: PeerB
Address: 10.200.200.2/32
Public Address: 95.177.218.232
Listen Port: 44444
Private Key: eL2JH5Z2zkWrOThAqcJ7TI25aQ6b7G67ZY+iIanZmFc=
[WGC]>
```

After completing all peers, you need now to export the configurations for all peers

Configurations export

```
GenerateConfigs /tmp/
```

```
[WGC]> GenerateConfigs /tmp/
[+] INFO: Generating configuration files
2020-03-20 05:52:16.338575 [+] INFO: Generating configuration file for 10.200.200.1/32
eL2JH5Z2zkWrOThAqcJ7TI25aQ6b7G67ZY+iIanZmFc=
2020-03-20 05:52:16.347049 [+] INFO: Generating configuration file for 10.200.200.2/32
YG1l/KNPmrt0pvNq2wGLWlBp2hg9k7tX/CV+IVzaelo=
[WGC]> Quit
```

You can now quit the configurator by typing “Quit”

Let us now list and view each file

```
ls -l /tmp/10.200.200.*
```

-rw-r--r-- 1 root root 338 Mar 20 05:52 /tmp/10.200.200.1.conf

-rw-r--r-- 1 root root 313 Mar 20 05:52 /tmp/10.200.200.2.conf

```
root@PeerA:~/wireguard-mesh-configurator# cat /tmp/10.200.200.1.conf
[Interface]
# Alias: PeerA
# Description: PeerA
PrivateKey = YG1l/KNPmrt0pvNq2wGLWlBp2hg9k7tX/CV+IVzaelo=
Address = 10.200.200.1/32
ListenPort = 44444

[Peer]
# Alias: PeerB
# Description: PeerB
PublicKey = NlJ/8sukAs8wBo6ASh3Wl8zSqUzULo4nO9j6CKkwN0w=
AllowedIPs = 10.200.200.2/32
Endpoint = 95.177.218.232:44444
PersistentKeepalive = 25
root@PeerA:~/wireguard-mesh-configurator#
```

```
root@PeerA:~/wireguard-mesh-configurator# cat /tmp/10.200.200.2.conf
[Interface]
# Alias: PeerB
# Description: PeerB
PrivateKey = eL2JH5Z2zkWrOThAqcJ7TI25aQ6b7G67ZY+iIanZmFc=
Address = 10.200.200.2/32
ListenPort = 44444

[Peer]
# Alias: PeerA
# Description: PeerA
PublicKey = 9rFF9tvmVWMxXMGJ0B/yqMJNStImdDCQ2Y6GrNgdB3I=
AllowedIPs = 10.200.200.1/32
Endpoint = 95.177.218.228:44444
root@PeerA:~/wireguard-mesh-configurator#
```

Now copy these files each into its related server

You can use scp to do that

```
scp /tmp/10.200.200.1.conf
root@95.177.218.232:/etc/wireguard/wg0.conf

scp /tmp/10.200.200.2.conf
root@95.177.218.232:/etc/wireguard/wg0.conf
```

Go to each server and run WireGuard

1- Enable routing on all servers

```
cat >> /etc/sysctl.conf << EOF
net.ipv4.ip_forward = 1
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
EOF
sysctl -p
echo 1 > /proc/sys/net/ipv4/ip_forward
```

2- Grant permissions on WireGuard files

```
chown -v root:root /etc/wireguard/wg0.conf
chmod -v 600 /etc/wireguard/wg0.conf
```

3- Enable & Turn on WireGuard

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

4- Save running configurations and take a backup

```
wg-quick save wg0
cp /etc/wireguard/wg0.conf /tmp/$(uname -n)-wireguard-$(date
+%Y-%m-%d_%H:%M:%S).conf
```

Testing WireGuard

Below commands will show you the WireGuard Configurations

```
wg
wg show
wg showconf wg0
```

Configure firewall rules on the server

We will need to set up a few firewall rules to manage our VPN and DNS traffic.

Track VPN connection

```
iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED
-j ACCEPT
iptables -A FORWARD -m conntrack --ctstate
RELATED,ESTABLISHED -j ACCEPT
```

Allowing incoming VPN traffic on the listening port

```
iptables -A INPUT -p udp -m udp --dport 44444 -m conntrack -
-ctstate NEW -j ACCEPT
```

Allow both TCP and UDP recursive DNS traffic

```
iptables -A INPUT -s 10.200.200.0/24 -p tcp -m tcp --dport
53 -m conntrack --ctstate NEW -j ACCEPT
iptables -A INPUT -s 10.200.200.0/24 -p udp -m udp --dport
53 -m conntrack --ctstate NEW -j ACCEPT
```

Allow forwarding of packets that stay in the VPN tunnel

```
iptables -A FORWARD -i wg0 -o wg0 -m conntrack --ctstate NEW
-j ACCEPT
```

Set up nat

```
iptables -t nat -A POSTROUTING -s 10.200.200.0/24 -o eth0 -j
MASQUERADE
```

We also want to ensure that the rules remain persistent across reboots.

```
apt-get install iptables-persistent
systemctl enable netfilter-persistent
netfilter-persistent save
```

Troubleshooting WireGuard

In case the tunnel is not up or you face any issue please follow below steps:

1- Make sure that you have latest configurations of /etc/wireguard/wg0.conf file
You can take a copy of it with below command

```
cp /etc/wireguard/wg0.conf /tmp/$(uname -n)-wireguard-$(date +%Y-%m-%d_%H:%M:%S).conf
```

2- Re-establish the connectivity again by first bring the tunnel down on the impacted server

```
wg-quick down wg0
```

3- Now clear the content of /etc/wireguard/wg0.conf file

```
> /etc/wireguard/wg0.conf
```

4- Make sure that the backed up configurations are correct, you need to double check below points before you copy it back again to the > /etc/wireguard/wg0.conf :
   a. Make sure the private keys of each server is correct.
   b. Make sure the public keys of each server is correct.
   c. Make sure the preshared key on each server is correct.
   d. Make sure the public IPs and endpoint IPs are correct.

5- Now copy back the backup copy to of your configurations to /etc/wireguard/wg0.conf

```
cp <type the path of backup file> /etc/wireguard/wg0.conf
```

6- Bring the tunnel up again and save the configurations:

```
wg-quick up wg0
wg-quick save wg0
```

Debugging & Logging

To enable debugging and logging execute below commands

```
modprobe wireguard
echo module wireguard +p > /sys/kernel/debug/dynamic_debug/control
```

Now view the log with below command:

```
journalctl -flu wg-quick@wg0
```

Or

```
tail -f /var/log/syslog
```