

# Configuration vpn ssl/tls

1) `sudo apt update` : Updates the local package list to ensure you have the latest versions of available software.

2) `sudo apt install wireguard` : Installs the WireGuard VPN software on your system.

3) `wg --version` : Displays the installed version of WireGuard.

4) `wg genkey | tee privatekey.key | wg pubkey > publickey` :

generates a private key for  
the WireGuard VPN interface

saves the private key to  
file named `privatekey`

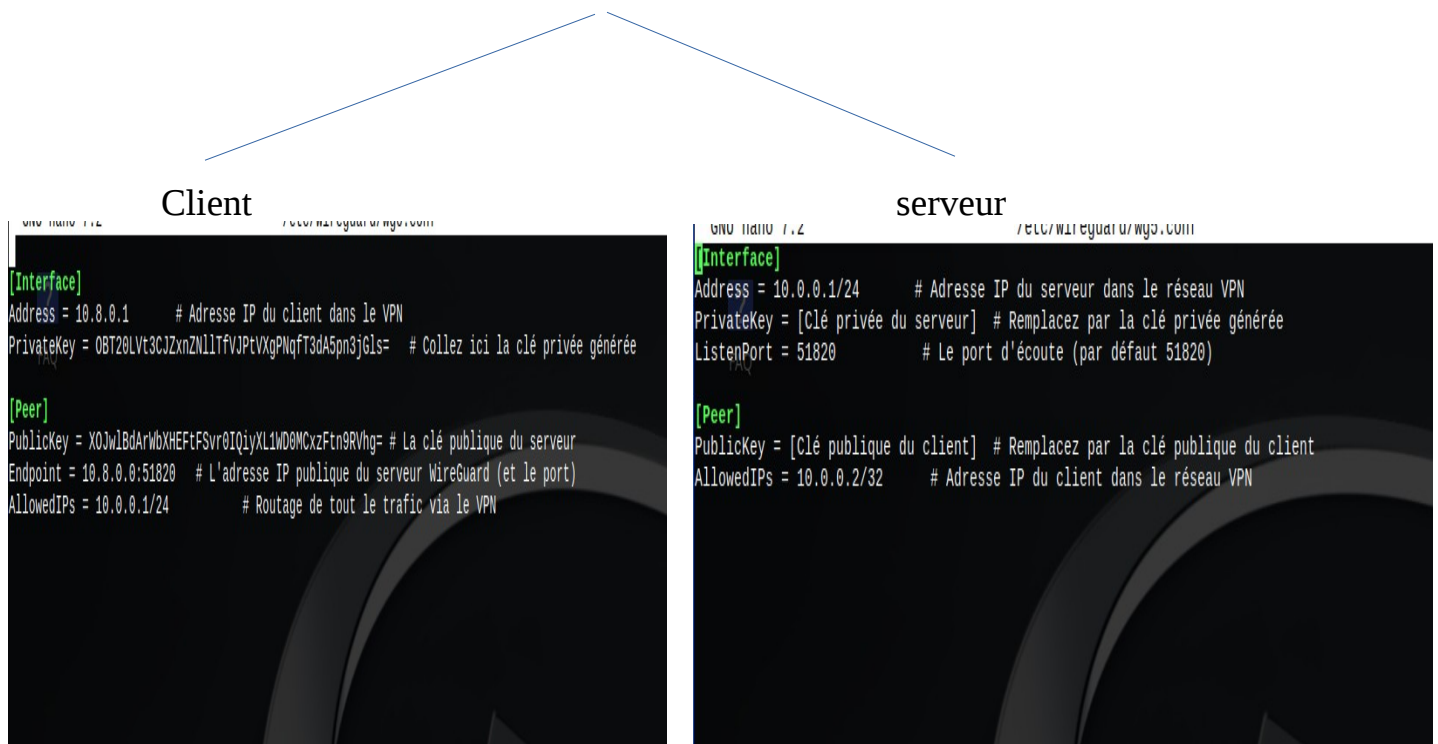
generates the corresponding  
public key from the private key.

writes the public key to  
the `publickey` file.

```
manar@mx:~$ sudo apt install wireguard
[sudo] Mot de passe de manar :
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
wireguard est déjà la version la plus récente (1.0.20210914-1).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 29 non mis à jour.
manar@mx:~$ wg genkey | tee privatekey.key | wg pubkey > publickey
manar@mx:~$
```



5) `sudo nano /etc/wireguard/wg0.conf` : Opens the WireGuard configuration file for editing using the nano text editor.



-----client-----

```
[Interface]
Address = 10.0.0.1/24      # Adresse IP du client dans le réseau VPN
PrivateKey = [Clé privée du client] # Remplacez par la clé privée générée
[Peer]
PublicKey = [Clé publique du server] # Remplacez par la clé publique du client
Endpoint=<ip_du_serveur>:51820
AllowedIPs = 0.0.0.0/0     # Adresse IP du client dans le réseau VPN
```

-----server-----

```
[Interface]
Address = 10.0.0.1/24      # Adresse IP du serveur dans le réseau VPN
PrivateKey = [Clé privée du serveur] # Remplacez par la clé privée générée
ListenPort = 51820        # Le port d'écoute (par défaut 51820)

[Peer]
PublicKey = [Clé publique du client] # Remplacez par la clé publique du client
AllowedIPs = 10.0.0.2/32    # Adresse IP du client dans le réseau VPN
```

6) `sudo wg-quick up wg0` : Brings up the WireGuard interface `wg0` and starts the VPN connection.

7) `sudo systemctl enable wg-quick@wg0` : Enables the WireGuard VPN interface to automatically start at boot.

8) `sudo wg` : Displays the current status of the WireGuard VPN, including the active peers.

```
manar@mx:~$ sudo wg-quick up wg0
[#] ip link add wg0 type wireguard
[#] wg setconf wg0 /dev/fd/63
Warning: AllowedIP has nonzero host part: 10.0.0.1/24
[#] ip -4 address add 10.8.0.1 dev wg0
[#] ip link set mtu 1420 up dev wg0
[#] ip -4 route add 10.0.0.0/24 dev wg0
manar@mx:~$ sudo systemctl enable wg-quick@wg0
manar@mx:~$ sudo wg
interface: wg0
  public key: J/SiYrJ+4BDN+lqcCAWiMhTvyDdGuwFwo/cSCTSPolw=
  private key: (hidden)
  listening port: 45475

peer: X0Jw1BdArWbXHEftFSvr0IQiyXL1WD0MCxzFtn9RVhg=
  endpoint: 10.8.0.0:51820
  allowed ips: 10.0.0.0/24
manar@mx:~$
```

## Verification :

```
manar@mx:~$ ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Boucle locale)
    RX packets 163 bytes 23260 (22.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 163 bytes 23260 (22.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
    inet 10.8.0.1 netmask 255.255.255.255 destination 10.8.0.2
    inet6 fe80::2813:d3a5:9fb0:9a87 prefixlen 64 scopeid 0x20<link>
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 500 (UNSPEC)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 9 bytes 432 (432.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wg0: flags=209<UP,POINTOPOINT,RUNNING,NOARP> mtu 1420
    inet 10.8.0.1 netmask 255.255.255.255 destination 10.8.0.1
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 1000 (UNSPEC)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

# SSL Encryption (OpenSSL)

1) `sudo apt install openssl` : Installs OpenSSL, a toolkit for Secure Sockets Layer (SSL) and public key cryptography

2) `openssl genpkey -algorithm RSA -out private_key.pem -aes256` :  
Generates an RSA private key and encrypts it with AES-256

3) `openssl rsa -pubout -in private_key.pem -out public_key.pem` :  
Generates the corresponding public key from the private key.

\*\*\* Now ,the server send his public key to client , by :

4) `scp /etc/wireguard/publickey.key user@client ip:/path/to/destination`

## Client :

1) `echo "ceci est un message sécurisé" > message.txt` :Creates a file named `message . txt` and writes the encrypted message into it.

2) `openssl pkeyutl -encrypt -inkey server_public.key -pubin -in message.txt -out message_encrypted.bin` : Encrypts the contents of `message . txt` using the server's public key

3) `scp %path/message_encrypted.bin user@server ip:chemin` : Uses SCP (Secure Copy Protocol) to transfer the encrypted file to the server

## Server :

1) `openssl rsautl -decrypt -inkey server-private.key -in /tmp/message-encrypted.bin -out message_decrypted.txt` : Decrypts the encrypted message file using the server's private key

2) `cat message_decrypted.txt` : Displays the contents of the decrypted message on the server.