# OpenVPN Community VPN daemon

## CONCEPTS

> An option on the command line is the same as in a config file, just with the leading -- dropped: --verb 4 becomes verb 4.

**Client** tun0 device
↓ TLS handshake
**Control channel** certs, tls-crypt
↓ keys
**Data channel** data-ciphers
↓ UDP 1194
**Server** tun0, routes

> dev tun carries IP packets (routed VPN); dev tap carries Ethernet frames (bridging). Both ends must match.

## RUN

**openvpn --config** *<file>*
Start with a config file

**openvpn --config** *<file>* **--daemon**
Detach and run in background

**openvpn --config** *<file>* **--verb 4**
Set log verbosity (0-11, default 1)

**openvpn --config** *<file>* **--log-append** *<file>*
Log to a file (appends)

**openvpn --config** *<file>* **--status** *<file>* *<sec>*
Write a status file every N seconds

**openvpn --version**
Show version and build options

## CLIENT OPTIONS

**client**
Pull options from the server, act as TLS client

**remote** *<host>* *<port>* *<proto>*
Server to connect to

**dev tun**
Routed IP tunnel device

**proto udp**
Transport: udp, tcp-client, tcp-server

**remote** *<host>* **443 tcp-client**
Server reachable over TCP 443

## CLIENT OPTIONS (cont.)

**nobind**
Do not bind a fixed local port

**persist-key**
Keep keys across restarts

**persist-tun**
Keep the tun device across restarts

**auth-user-pass** *<file>*
Username and password from a file

## KEYS AND TLS

**ca** *<file>*
CA certificate to trust

**cert** *<file>*
Your certificate

**key** *<file>*
Your private key

**tls-crypt** *<file>*
Encrypt and authenticate control channel

**tls-auth** *<file>* *<0|1>*
Authenticate control channel with HMAC

**remote-cert-tls server**
Require a server-purpose certificate

**data-ciphers** *<list>*
Colon-separated data channel ciphers

**openvpn --genkey secret** *<file>*
Create a static key file

**openvpn --genkey tls-crypt** *<file>*
Create a tls-crypt key

**openvpn --show-ciphers**
List available ciphers

## SERVER OPTIONS

**server** *<network>* *<netmask>*
Server mode with a client address pool

**dh none**
Use ECDH only, no DH file

**push "***<option>***"**
Send an option to every client

**client-config-dir** *<dir>*
Per-client files named by common name

## SERVER OPTIONS (cont.)

**ifconfig-push** *<ip>* *<netmask>*
Fixed address, set in a client file

**keepalive** *<interval>* *<timeout>*
Ping and restart on silence

## ROUTING AND DNS

**push "redirect-gateway def1"**
Send all client traffic via the VPN

**redirect-gateway def1**
Same, set on the client

**push "route** *<network>* *<netmask>***"**
Give clients a route to a subnet

**sysctl -w net.ipv4.ip_forward=1**
Let the server forward packets

## OPERATE

**systemctl start openvpn-client@***<name>*
Start /etc/openvpn/client/<name>.conf

**systemctl start openvpn-server@***<name>*
Start /etc/openvpn/server/<name>.conf

**journalctl -u openvpn-client@***<name>* **-f**
Follow the service log

## TROUBLESHOOT

**ip addr show tun0**
Check the tunnel got an address

**ip route**
Check the VPN routes exist

**openvpn --config** *<file>* **--verb 6**
Debug handshake problems

### Common log messages

| Message | Usually means |
|---|---|
| `TLS key negotiation failed` | Port, firewall or key |
| `certificate verify failed` | Bad CA, expiry, clock |
| `AUTH_FAILED` | Wrong login |
| `Initialization Sequence Completed` | Connect |