# WireGuard 服务端部署及客户端设置手册

匿名版 · 适用于 WireGuard-UI + wg-quick + ESP32/手机/电脑客户端

整理日期：2026-07-03

**网关端访问配置页面进行客户端配置；伴侣端需要在设置中点击“WGweb 配置”选项中的“启动”，当切换到运行中时，可以访问伴侣的 IP 地址进行 web 访问，由于硬件资源原因，web 配置每次开启默认 5 分钟后自动关闭，节约资源**

# 目录

# 1. 目标架构与示例参数

本手册目标是部署一个 WireGuard 服务端，并通过 WireGuard-UI 管理客户端，使多个客户端可以通过 VPN 内网互联，同时客户端自己的普通上网流量不走服务器。

**核心目标：**客户端访问 VPN 内部地址时走 WireGuard；客户端访问普通互联网时仍走本地网络。

| 项目 | 示例值 / 建议值 |
| --- | --- |
| 示例服务器域名 | vpn.example.com |
| 示例服务器公网 IP | 203.0.113.10 |
| WireGuard UDP 端口 | 51820 |
| WireGuard 网段 | 10.88.10.0/24 |
| 服务端 VPN IP | 10.88.10.1/24 |
| 客户端 A 示例 IP | 10.88.10.2/32 |
| ESP32 客户端示例 IP | 10.88.10.3/32 |
| Gateway 客户端示例 IP | 10.88.10.4/32 |

后续命令和配置均使用上述匿名示例值。实际部署时请将 vpn.example.com 或 203.0.113.10 替换为你的服务器公网 IP 或域名。

# 2. 服务端安装 WireGuard

以下命令适用于 Ubuntu / Debian 系统。

**安装 WireGuard：**

```
apt update
apt install wireguard wireguard-tools wget -y
```

**检查版本：**

```
wg --version
```

# 3. 安装与启动 WireGuard-UI

建议将 WireGuard-UI 安装在 /opt/wireguard-ui。

**下载并解压：**

```
cd /opt
```

```
mkdir -p wireguard-ui
wget https://github.com/ngoduykhanh/wireguard-ui/releases/download/v0.6.2/wireguard-ui-v0.6.2-linux-amd64.tar.gz
tar -zxvf wireguard-ui-v0.6.2-linux-amd64.tar.gz -C /opt/wireguard-ui
```

## 3.1 配置 .env

```
BIND_ADDRESS=0.0.0.0:5000


EMAIL_FROM_ADDRESS=admin@example.com
EMAIL_FROM_NAME=admin
SMTP_HOSTNAME=smtp.example.com
SMTP_PORT=465
SMTP_USERNAME=admin@example.com
SMTP_PASSWORD=replace-with-mail-app-password
SMTP_AUTH_TYPE=LOGIN
SMTP_ENCRYPTION=SSL
```

**设置 .env 权限：**

```
chmod 600 /opt/wireguard-ui/.env
```

## 3.2 配置 systemd 服务

```
[Unit]
Description=WireGuard UI Daemon
Wants=network-online.target
After=network-online.target


[Service]
User=root
Group=root
Type=simple
WorkingDirectory=/opt/wireguard-ui
EnvironmentFile=/opt/wireguard-ui/.env
ExecStart=/opt/wireguard-ui/wireguard-ui


[Install]
WantedBy=multi-user.target
```

**启动 WireGuard-UI：**

```
systemctl daemon-reload
systemctl enable wireguard-ui.service
systemctl start wireguard-ui.service
systemctl status wireguard-ui.service --no-pager
```

浏览器访问：

```
http://203.0.113.10:5000
# 或
http://vpn.example.com:5000
```

**默认账号：**WireGuard-UI 默认账号通常为 admin / admin。首次登录后应立即修改密码。

# 4. WireGuard-UI 全局设置

进入 WireGuard-UI 的 Global Settings 页面，建议按下表设置。

| 项目 | 示例值 / 建议值 |
| --- | --- |
| Endpoint Address | vpn.example.com 或 203.0.113.10 |
| DNS Servers | 建议留空；分流模式下通常不需要强制下发 DNS |
| MTU | 1450 |
| Persistent Keepalive | 15 |
| Firewall Mark | 0xca6c，可保持默认 |
| Table | auto |
| WireGuard Config File Path | /etc/wireguard/wg0.conf |

**分流提示：**是否让客户端普通上网走服务器，不由服务端公网 IP 决定，而主要由客户端配置中的 AllowedIPs 决定。

# 5. 服务端 wg0 配置原则

进入 Wireguard Server 页面，服务端接口地址建议使用网关地址，不要使用网段地址。

| 项目 | 示例值 / 建议值 |
| --- | --- |
| Server Interface Addresses | 10.88.10.1/24 |
| Listen Port | 51820 |
| Private Key | 由 UI 自动生成，勿公开 |

**不要这样填：**不要把服务端地址填成 10.88.10.0/24，因为 .0 是网段地址，不适合作为接口地址。

最终服务端配置大致如下：

```
[Interface]
Address = 10.88.10.1/24
ListenPort = 51820
PrivateKey = <server-private-key>
MTU = 1450
PostUp = <见下一节>
PostDown = <见下一节>
```

# 6. PostUp / PostDown 推荐脚本

如果目标是“客户端之间互访走 VPN，客户端普通上网不走服务器”，推荐使用最小转发规则，只允许 wg0 到 wg0。

## 6.1 推荐 PostUp

```
sysctl -w net.ipv4.ip_forward=1; iptables -C FORWARD -i %i -o %i -j ACCEPT 2>/dev/null || iptables -
```

```
I FORWARD 1 -i %i -o %i -j ACCEPT
```

### 6.2 推荐 PostDown

```
iptables -D FORWARD -i %i -o %i -j ACCEPT 2>/dev/null || true
```

### 6.3 持久化开启 IPv4 转发

**执行：**

```
cat > /etc/sysctl.d/99-wireguard-forward.conf <<'EOF'
net.ipv4.ip_forward=1
EOF


sysctl --system
sysctl net.ipv4.ip_forward
```

### 6.4 关于 NAT / MASQUERADE

如果加入下面的 NAT 规则，服务端就具备为客户端提供公网出口的能力：

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

分流模式下不建议长期保留该规则。即使客户端 AllowedIPs 写错为 0.0.0.0/0，也不应让它意外通过服务器上网。

**如需删除 NAT：**

```
iptables -t nat -D POSTROUTING -o eth0 -j MASQUERADE 2>/dev/null || true
```

## 7. 创建客户端与分流设置

进入 Wireguard Clients 页面，为每个客户端分配唯一的 /32 地址。服务端 Peer 中的 AllowedIPs 保持每个客户端自己的 /32；客户端配置中的 AllowedIPs 则写 VPN 网段。

| 项目 | 示例值 / 建议值 |
| --- | --- |
| 客户端 A | IP Allocation: 10.88.10.2/32；Allowed IPs: 10.88.10.0/24 |
| ESP32 客户端 | IP Allocation: 10.88.10.3/32；Allowed IPs: 10.88.10.0/24 |
| Gateway 客户端 | IP Allocation: 10.88.10.4/32；Allowed IPs: 10.88.10.0/24 |
| Use server DNS | 建议不勾选 |
| Endpoint | 普通客户端留空 |
| Preshared Key | ESP32 建议不使用；手机/电脑按需 |

导出的普通客户端配置示例：

```
[Interface]
PrivateKey = <client-private-key>
Address = 10.88.10.2/32
MTU = 1450
```

```
[Peer]
PublicKey = <server-public-key>
AllowedIPs = 10.88.10.0/24
Endpoint = vpn.example.com:51820
PersistentKeepalive = 15
```

**AllowedIPs 说明：** AllowedIPs = 10.88.10.0/24 表示只有访问 VPN 内网走 WireGuard；AllowedIPs = 0.0.0.0/0 表示全部 IPv4 流量走 WireGuard。

## 8. ESP32 客户端特殊设置

ESP32 客户端需要特别注意本地掩码。调试中发现，如果 ESP32 VPN 掩码填成 255.255.255.255，它只能认为自己的单 IP 可达，客户端互访会异常。

| 项目 | 示例值 / 建议值 |
| --- | --- |
| WireGuard enabled | 开启 |
| ESP32 VPN IP | 10.88.10.3 |
| ESP32 VPN 掩码 | 255.255.255.0 |
| 服务器公钥 | <server-public-key> |
| 服务器地址 | vpn.example.com 或 203.0.113.10 |
| 服务器端口 | 51820 |
| 本地监听端口 | 51820 或随机端口 |
| Persistent Keepalive | 15 |
| PresharedKey | 不填 |

**ESP32 关键点：** ESP32 VPN IP = 10.88.10.3 且掩码 = 255.255.255.0，等价于 10.88.10.3/24。这样它可以访问 10.88.10.1、10.88.10.2、10.88.10.4 等 VPN 内部地址。

ESP32 对应服务端 Peer 仍然是：

```
[Peer]
PublicKey = <esp32-client-public-key>
AllowedIPs = 10.88.10.3/32
PersistentKeepalive = 15
```

服务端这里不要写 10.88.10.0/24，否则多个客户端会冲突。

## 9. 应用配置与状态检查

在 WireGuard-UI 中新增或修改客户端后，点击 Save，然后点击 Apply Config。之后可以在服务器上检查状态。

**重启与查看：**

```
systemctl restart wg-quick@wg0.service
wg show wg0
```

正常状态应看到每个客户端的 latest handshake：

```
peer: <client-public-key>
  endpoint: <client-public-ip>:<port>
  allowed ips: 10.88.10.x/32
  latest handshake: 5 seconds ago
  transfer: ...
```

**检查配置文件：**

```
cat /etc/wireguard/wg0.conf
```

**查看服务状态：**

```
systemctl status wg-quick@wg0.service --no-pager
systemctl status wireguard-ui.service --no-pager
systemctl status wgui.path --no-pager
```

**查看防火墙规则：**

```
iptables -S FORWARD
iptables -t nat -S POSTROUTING
```

# 10. 常见问题排查

## 10.1 wg show 没有 peer

说明服务端配置没有加载客户端，或者 UI 没有 Apply Config。检查 /etc/wireguard/wg0.conf 中是否存在 [Peer] 段。

**检查：**

```
grep -nE '^\[Interface\]|^\[Peer\]|Address|ListenPort|AllowedIPs|PublicKey' /etc/wireguard/wg0.conf
```

## 10.2 有 endpoint 但没有 latest handshake

说明服务器可能收到握手包但握手未完成。常见原因包括客户端私钥和服务端 Peer 公钥不匹配，或者 PresharedKey 不一致。ESP32 建议不使用 PresharedKey。

## 10.3 ESP32 握手失败

典型日志包括 Decrypt failed 或 WG handshake response invalid。优先检查服务端公钥、客户端私钥对应的客户端公钥、PresharedKey 是否清空或一致。

## 10.4 客户端互访不通

重点检查三个位置：服务端开启 ip_forward；PostUp 允许 wg0 -> wg0；客户端 AllowedIPs 包含 VPN 网段。

```
客户端 AllowedIPs = 10.88.10.0/24
ESP32 掩码 = 255.255.255.0
```

### 10.5 Web 可以访问但 WebSocket 不通

这通常不是 WireGuard 隧道问题，而是 WebSocket 地址、监听地址、浏览器混合内容或应用防火墙问题。

**在服务器抓包：**

```
tcpdump -ni wg0 'tcp port 80 and host 10.88.10.4'
```

检查服务是否监听 0.0.0.0 或 VPN IP，而不是只监听 127.0.0.1。

## 11. 多用户与多子网隔离

同一个 WireGuard 服务端可以承载多个用户组和多个子网。是否互相隔离主要由服务端防火墙规则决定。

| 项目 | 示例值 / 建议值 |
| --- | --- |
| 用户组 A | 10.88.10.0/24 |
| 用户组 B | 10.88.20.0/24 |
| 用户组 C | 10.88.30.0/24 |

服务端可以配置多个接口地址：

```
[Interface]
Address = 10.88.10.1/24, 10.88.20.1/24, 10.88.30.1/24
ListenPort = 51820
PrivateKey = <server-private-key>
```

每个客户端仍然使用自己的 /32：

```
[Peer]
PublicKey = <group-a-client-public-key>
AllowedIPs = 10.88.10.2/32


[Peer]
PublicKey = <group-b-client-public-key>
AllowedIPs = 10.88.20.2/32
```

如果需要不同子网之间互不访问，应添加精确防火墙规则，不要使用全放行规则。示例：

```
iptables -I FORWARD 1 -i wg0 -o wg0 -s 10.88.10.0/24 -d 10.88.10.0/24 -j ACCEPT
iptables -I FORWARD 1 -i wg0 -o wg0 -s 10.88.20.0/24 -d 10.88.20.0/24 -j ACCEPT
iptables -I FORWARD 3 -i wg0 -o wg0 -s 10.88.0.0/16 -d 10.88.0.0/16 -j DROP
```

## 12. 安全建议与最终配置摘要

### 12.1 安全建议

- 不要公开服务端 PrivateKey、客户端 PrivateKey 或 PresharedKey。
- 如果调试过程中密钥曾经被复制到不可信环境，正式使用前建议重新生成服务端和客户端密钥。
- WireGuard-UI 的 .env 包含 SMTP 密码，应设置 chmod 600。

- UI 默认 admin/admin 登录后应立即修改。
- 不需要客户端上网走服务器时，建议不要保留 MASQUERADE NAT 规则。

## 12.2 最终推荐配置摘要

服务端 Wireguard Server：

```
Server Interface Addresses: 10.88.10.1/24
Listen Port: 51820


PostUp:
sysctl -w net.ipv4.ip_forward=1; iptables -C FORWARD -i %i -o %i -j ACCEPT 2>/dev/null || iptables -
I FORWARD 1 -i %i -o %i -j ACCEPT


PostDown:
iptables -D FORWARD -i %i -o %i -j ACCEPT 2>/dev/null || true
```

普通客户端：

```
[Interface]
Address = 10.88.10.x/32
PrivateKey = <client-private-key>
MTU = 1450


[Peer]
PublicKey = <server-public-key>
AllowedIPs = 10.88.10.0/24
Endpoint = vpn.example.com:51820
PersistentKeepalive = 15
```

ESP32 客户端：

```
VPN IP: 10.88.10.3
VPN Mask: 255.255.255.0
Server: vpn.example.com
Server Port: 51820
Server PublicKey: <server-public-key>
PresharedKey: empty
Keepalive: 15
```

**最终效果：**客户端之间访问走 WireGuard 服务器；客户端访问普通互联网不走 WireGuard 服务器。