**openvpn的几种组网方式**

服务器和客户端均为电脑主机，若要下挂私网，则主机需要有两块网卡，一块连外网，一块连私网。主机均安装Ubuntu操作系统。如果客户端/服务器本身就是路由器，可以省略SNAT的路由配置和路由转发功能步骤。路由模式-TUN是3层路由模式，网桥模式-TAP是2层连接模式。

# 路由模式-TUN

## 独立单客户端

### 拓扑图

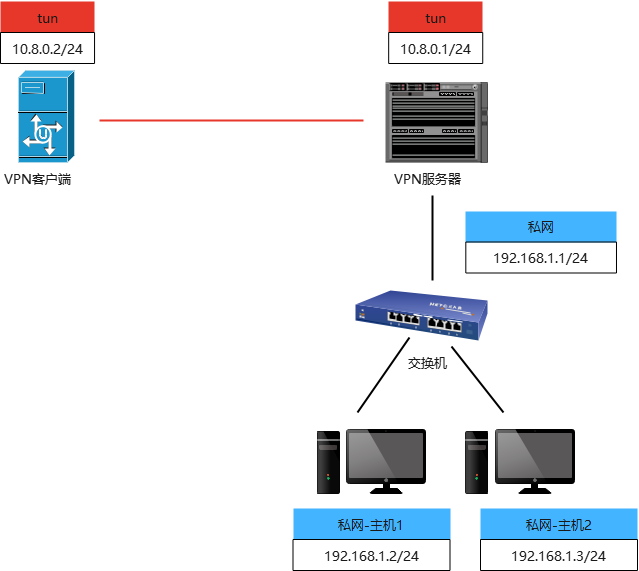


图 1 独立单客户端

### 目标

1. 客户端需要远程访问服务器；
2. 客户端不能访问服务器后面的私网；

即客户端可以访问服务器192.168.1.1，但是不能访问主机1和主机2；

服务端的配置无需特殊配置，连接服务器后，可以直接连接成功。

### 服务端路由推送

push "route 192.168.1.0 255.255.255.0"

## 单客户端互联互通

### 拓扑图

跟图 1一致；

### 目标

1. 客户端需要远程访问服务器；
2. 客户端能访问服务器后面的私网，即能访问主机1和主机2；

### 配置文件

配置在独立单客户端的基础上，完善服务端的SANT和路由转发功能，如下所述。

### 服务端增加SNAT的路由

将来自10.8.0.0/24网段的数据包进行源地址封装伪装当前主机与内容的通信，使用iptables添加SNAT路由，如下图所示。

iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -j MASQUERADE

此命令的配置为一次性配置，设备重启后配置消失，如果想永久保存此配置，需要安装iptables-persistent，每次修改完iptables以后用如下命令保存。

iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -j MASQUERADE

apt-get install iptables-persistent

netfilter-persistent save

### 服务端开启路由转发功能

echo 1 > /proc/sys/net/ipv4/ip\_forward

此命令的配置为一次性配置，设备重启后配置消失，如果想永久保存此配置，需要将net.ipv4.ip\_forward = 1导入到内核里去，使系统永久生效开启路由转发功能，命令如下：

echo "net.ipv4.ip\_forward = 1" >> /etc/sysctl.conf

## 单客户端全联通

### 拓扑图

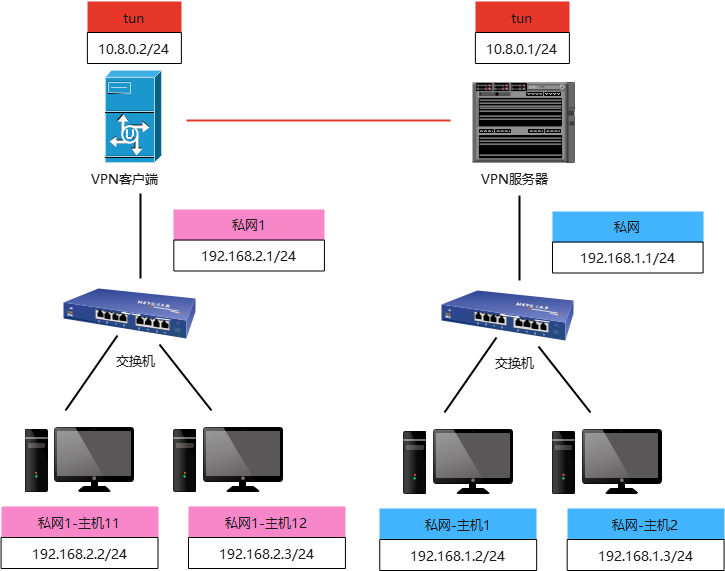


图 2 单客户端全联通

### 目标

1. 客户端需要访问服务器及私网；
2. 服务器需要访问客户端及私网1；

### 配置文件

在单客户端互联互通配置的基础上，增加如下配置。

### 服务端路由配置

route 192.168.2.0 255.255.255.0

### 服务端ccd配置

ccd文件夹中存放客户端的配置，比如客户端名称为client1，对应192.168.158.0/24网段。则在ccd需要创建文件：client1，然后存放以下内容：

iroute 192.168.2.0 255.255.255.0

此配置有2个作用：

(1)为服务端添加路由表，在VPN服务端进行数据转发时，将192.168.2.0 255.255.255.0转发给client1

(2)在进行路由推送时，忽略192.168.2.0 255.255.255.0对client1的推送。

### 客户端1增加SNAT的路由

跟服务器添加方法一致，如1.2.4。

### 客户端1开启路由转发功能

跟服务器添加方法一致，如1.2.5。

## 双独立客户端

### 拓扑图

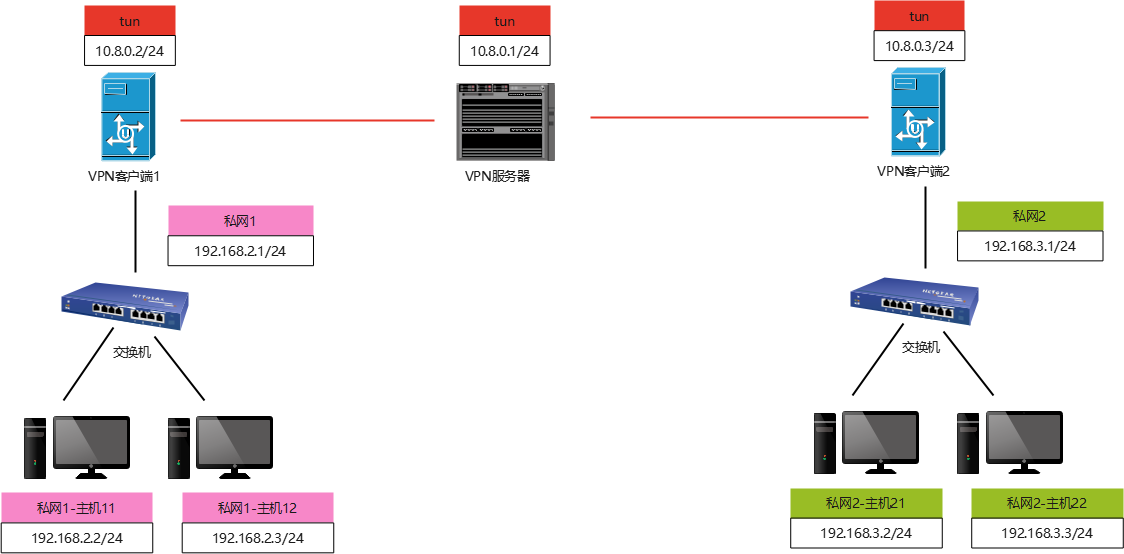


图 3 双独立客户端

### 目标

1. VPN服务端能够与两个路由器均连通；
2. VPN服务端能够与主机11，主机12连通；
3. VPN服务端能够与主机21，主机22连通；
4. 主机11与主机21，VPN客户端1与VPN客户端2，均不连通。

### 服务端路由配置

route 192.168.2.0 255.255.255.0

route 192.168.3.0 255.255.255.0

### 客户端1与客户端2增加SNAT的路由

跟服务器添加方法一致，如1.2.4。

### 客户端1与客户端2开启路由转发功能

跟服务器添加方法一致，如1.2.5。

## 双客户端互联互通

### 拓扑图

跟双独立客户端一致，如图 3所示。

### 目标

1. VPN服务端能够与两个客户端均连通；
2. VPN服务端能够与主机11，主机12连通；
3. VPN服务端能够与主机21，主机22连通；
4. 主机11与主机21，VPN客户端1与VPN客户端2，均连通。

### 配置文件

在双独立客户端配置的基础上增加下述配置。

### 开启client-to-client

默认情况下client-to-client配置项并未启用，所以客户端间是不能够进行连接的，要开启该功能，需要先在配置文件中加入：

client-to-client

### 服务端路由推送

两个网段互通的前提是：当向对向网段发起请求时，将数据包转发给tun。所以要在服务端配置下发给客户端的路由：

push "route 192.168.2.0 255.255.255.0"

push "route 192.168.3.0 255.255.255.0"

此时客户端在连接服务端后，则会在本地添加两条路由：

192.168.0.0/24 -> tun

192.168.1.0/24 -> tun

该路中的下发会引发客户端的路由地址冲突，因为客户端本身就存在192.168.0.0/24或192.168.1.0/24的路由段，该段的路由转发目的地为网关。会导致局域网访问不通的问题。

所以在配置推送（下发）的路由时，必须与ccd配置相结合。

### 服务端ccd配置

ccd文件夹中存放客户端的配置，比如两个客户端名称分别为client1, client2，对应2.0以3.0网段。

则在ccd需要创建文件：client1，然后存放以下内容：

iroute 192.168.2.0 255.255.255.0

此配置有2个作用：

1. 为服务端添加路由表，在 VPN 服务端进行数据转发时，将192.168.2.0 255.255.255.0转发给 client1；
2. 所以此时client1对应的路由器将仅仅得到下发的路由：192.168.2.0 255.255.255.0

同时，还需要在ccd中创建client2：

iroute 192.168.3.0 255.255.255.0

最后：

客户端1得到的路由(示意，实际路由条件会多，但效果相同）如下：

192.168.2.0/24 -> tun10.8.0.1 -> tun

客户端2得到的路由(示意，实际路由条件会多，但效果相同）如下：

192.168.3.0/24 -> tun10.8.0.1 -> tun

服务端路由：

192.168.2.0/24 -> client1

192.168.3.0/24 -> client2

## 全联通

我们还需要连通服务端所在的局域网。

### 拓扑图

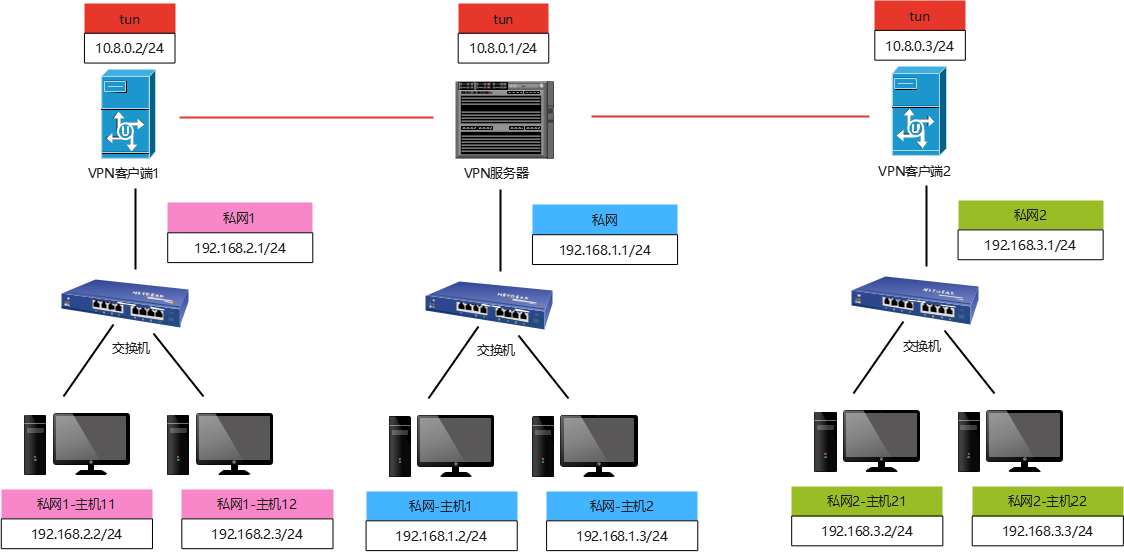


图 4 全联通

### 目标

1. VPN服务端能够与两个客户端均连通；
2. VPN服务端能够与主机11，主机12连通；
3. VPN服务端能够与主机21，主机22连通；
4. 机11与主机21，VPN客户端1与VPN客户端2，均连通
5. 主机1，主机11与主机21均连通，即私网、私网1和私网2全联通。

### 配置文件

在双客户端互联互通的基础上，完善服务器端的路由推送：

push "route 192.168.1.0 255.255.255.0"

push "route 192.168.2.0 255.255.255.0"

push "route 192.168.3.0 255.255.255.0"

# 网桥模式-TAP

服务器和客户端均以网桥的模式连接在一起，组成一个超大的虚拟局域网，客户端和服务器的子网地址不能冲突；客户端若要扩展子网就增加一张网卡，并将网卡加入网桥；若不想暴露子网，则不将网卡加入网桥。服务器端设有网桥，连接在服务器后面内网上的子网主机，只要IP在同一个局域网内，可以自动组成局域网。客户端连上服务器以后，服务器会自动分配一个IP给客户端的tap。

## 独立客户端

### 拓扑图

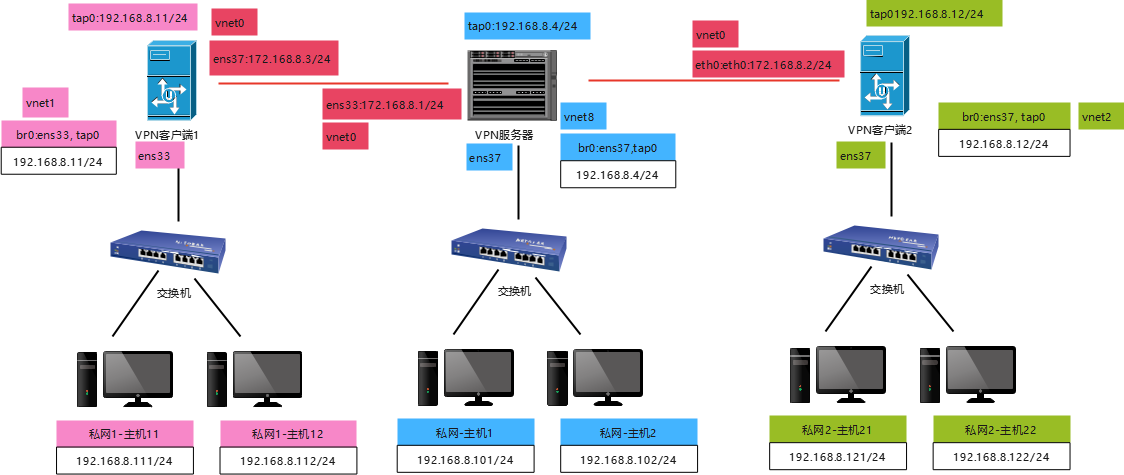


图 5 独立客户端

### 目标

1. VPN服务器可以访问VPN客户端1和VPN客户端2，以及下挂的子网；
2. VPN客户端1和VPN客户端2可以访问服务器以及子网；
3. VPN客户端1与VPN客户端2之间互不相能访问。

### 创建或停止网桥

在服务端创建网桥，脚本位置一般在/usr/share/doc/openvpn-xxx/sample-scripts目录下，xxx是版本号，根据实际业务需求设置参数。

创建网桥的脚本，sample-scripts/bridge-start：

#!/bin/bash

#################################

# Set up Ethernet bridge on Linux

# Requires: bridge-utils

#################################

# Define Bridge Interface

br="br0"

# Define list of TAP interfaces to be bridged,

# for example tap="tap0 tap1 tap2".

tap="tap0"

# Define physical ethernet interface to be bridged

# with TAP interface(s) above.

eth="eth0"

eth\_ip="192.168.8.4"

eth\_netmask="255.255.255.0"

eth\_broadcast="192.168.8.255"

for t in $tap; do

openvpn --mktun --dev $t

done

brctl addbr $br

brctl addif $br $eth

for t in $tap; do

brctl addif $br $t

done

for t in $tap; do

ifconfig $t 0.0.0.0 promisc up

done

ifconfig $eth 0.0.0.0 promisc up

ifconfig $br $eth\_ip netmask $eth\_netmask broadcast $eth\_broadcast

停止网桥的脚本sample-scripts/bridge-start

#!/bin/bash

####################################

# Tear Down Ethernet bridge on Linux

####################################

# Define Bridge Interface

br="br0"

# Define list of TAP interfaces to be bridged together

tap="tap0"

ifconfig $br down

brctl delbr $br

for t in $tap; do

openvpn --rmtun --dev $t

done

脚本执行顺序：

* run bridge-start
* run openvpn
* stop openvpn
* run bridge-stop

### 修改服务器端配置

dev tap0

server-bridge 192.168.8.4 255.255.255.0 192.168.8.128 192.168.8.254

### 修改客户端配置

dev tap

## 全联通

### 拓扑图

跟独立客户端一致，如图 5所示。

### 目标

1. VPN服务器可以访问VPN客户端1和VPN客户端2，以及下挂的子网；
2. VPN客户端1和VPN客户端2可以访问服务器以及子网；
3. VPN客户端1与VPN客户端2之间也可以相互访问。

### 配置文件

服务端配置在独立客户端的配置的基础上，增加如下配置。

### 开启client-to-client

默认情况下client-to-client配置项并未启用，所以客户端间是不能够进行连接的，要开启该功能，需要先在配置文件中加入：

client-to-client

## 客户端互联互通

### 拓扑图

跟独立客户端一致，如图 5所示。

### 目标

1. VPN服务器不能访问VPN客户端1和VPN客户端2，以及下挂的子网；
2. VPN客户端1和VPN客户端2可以访问服务器以及子网；
3. VPN客户端1与VPN客户端2之间也可以相互访问。

### 配置文件

有两种实现方案：

服务端配置在全联通配置的基础上，将服务器的配置文件改为

dev tap