1-OpenVPN 简介部署

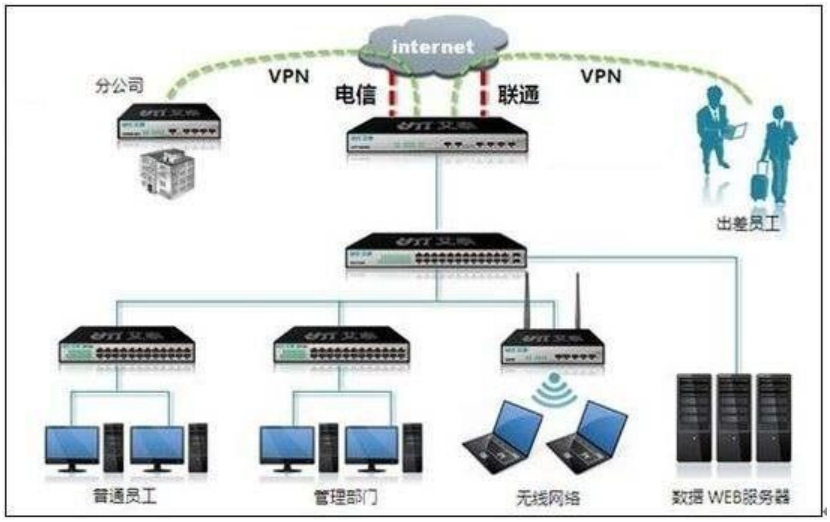
作者：Carey
原文链接：https://ld246.com/article/1611666519622
来源网站：链滴

1 OpenVPN简介

1.1 VPN 介绍

专用网：专用网就是在两个网络（例如，北京和广州）之间架设一条专用线路，但是它并不需要真正去铺设光缆之类的物理线路。虽然没有亲自去铺设，但是需要向电信运营商申请租用专线，在这条专的线路上只传输自己的信息,所以安全稳定,同时也费用高昂

VPN：Virtual Private Network，虚拟私有网络，又称为虚拟专用网络，用于在不安全的线路上安全传输数据。



1.2 OpenVPN

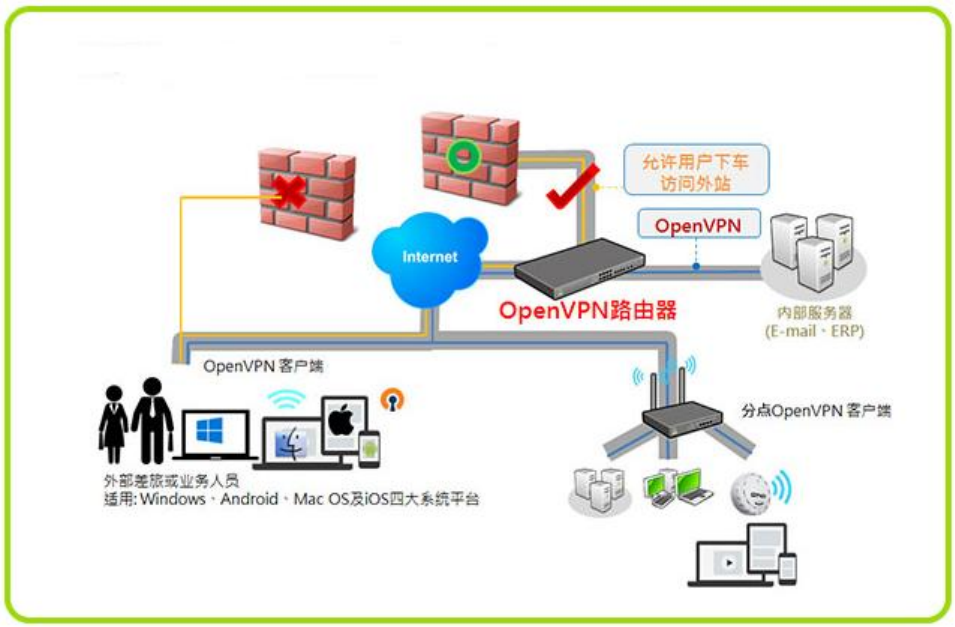
OpenVPN：一个实现VPN的开源软件，OpenVPN 是一个健壮的、高度灵活的 VPN 守护进程。它持SSL/TLS 安全、Ethernet bridging、经由代理的 TCP 或 UDP 隧道和 NAT。另外，它也支持动态 I

地址以及DHCP，可伸缩性足以支持数百或数千用户的使用场景，同时可移植至大多数主流操作系统台上。

官网：https://openvpn.net

GitHub地址：https://github.com/OpenVPN/openvpn

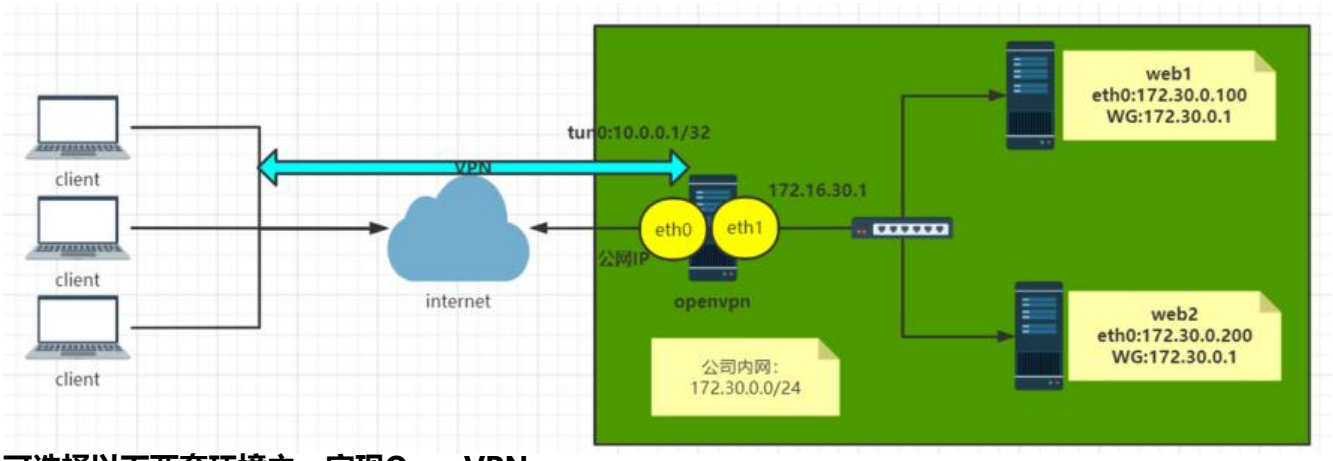
OpenVPN 示意图



2 OpenVPN 部署

2.1 准备 OpenVPN 部署环境

官文文档: https://openvpn.net/community-resources/how-to/



可选择以下两套环境之一实现OpenVPN

2.1.1 环境1: 阿里云OpenVPN 实战环境

准备阿里云网络实验环境

1 阿里云创建专有网络
指定城市和可用区
网段名称zhang-net1和地址段172.16.0.0/12
交换机名称zhang-net1-sw1 可用区A IPv4地址段172.30.0.0/24
安全组开放22端口
2 创建openvpn服务器有公网IP的实例1个
3 创建局域网的服务器无公网IP的实例2个
4 重设所有实例密码
5 修改安全组打开 1194/tcp/udp端口

准备完成的实例环境

俩台内网web服务器可以不设置网关

防火墙规则配置

2.1.2 环境2: 局域网 OpenVPN 实战环境

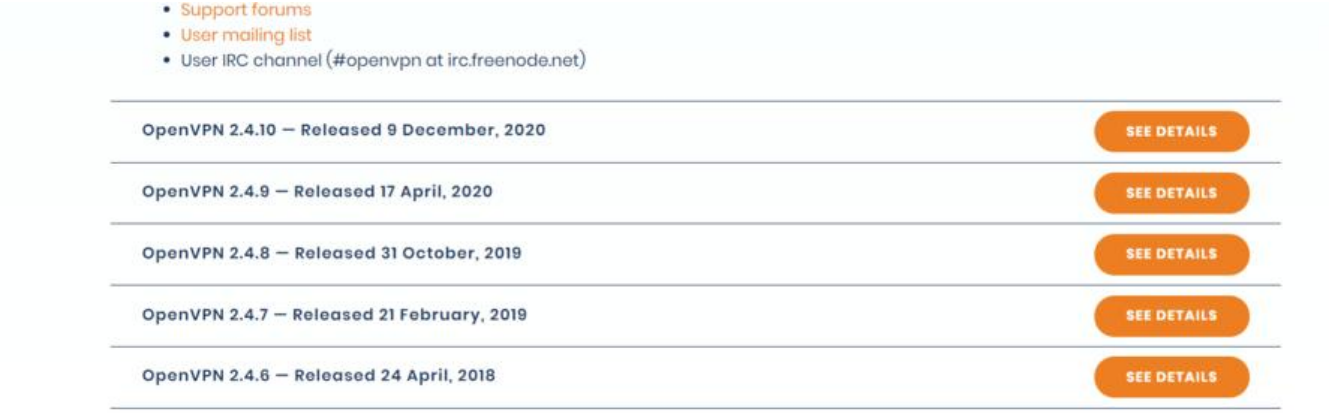
共四台主机
1 openvpn server：
CentOS 8.2
eth0:10.0.0.8/24 NAT模式,模拟公网IP
eth1:172.30.0.1/24 仅主机模式,私网IP
2 内网主机两台
第一台主机
eth0:172.30.0.100/24 仅主机模式,私网IP，无需网关
第二台主机
eth0:172.30.0.200/24 仅主机模式,私网IP，无需网关
3 Windows 客户端
Windows 10

2.2 安装OpenVPN软件包

2.2.1 查看版本

2.2.1.1 查看官网的OpenVPN的版本

访问官网：https://openvpn.net



2.2.1.2 在不同OS上查看OpenVPN版本

CentOS系统上的EPEL源OpenVPN版本比Ubuntu的仓库中版本更新,以下选择在CentOS8上部署OenVPN

范例: CentOS 查看OpenVPN版本

```
[11:58:01 root@openvpn-server ~]#yum list openvpn
Repository epel is listed more than once in the configuration
Extra Packages for Enterprise Linux Modular 8 - x86_64          153 kB/s | 537 kB     00:03
Available Packages
openvpn.x86_64                          2.4.10-1.el8                        epel
[12:03:21 root@openvpn-server ~]#yum list easy-rsa
Repository epel is listed more than once in the configuration
Last metadata expiration check: 0:00:32 ago on Tue 26 Jan 2021 12:03:20 PM CST.
Available Packages
easy-rsa.noarch                         3.0.8-1.el8                         epel
```

2.2.2 安装OpenVPN

2.2.2.1 安装OpenVPN和证书工具

```
#OpenVPN服务器端
[12:03:52 root@openvpn-server ~]#yum install -y openvpn
#证书管理工具
[12:05:19 root@openvpn-server ~]#yum install -y easy-rsa
```

2.2.2.2 查看包中相关文件

```
[12:06:00 root@openvpn-server ~]#rpm -ql openvpn
[12:06:08 root@openvpn-server ~]#rpm -ql easy-rsa
```

2.2.2.3 准备相关配置文件

```
#生成服务器配置文件
[12:06:38 root@openvpn-server ~]#cp /usr/share/doc/openvpn/sample/sample-config-files/server.conf /etc/openvpn/
#准备证书签发相关文件
[12:08:04 root@openvpn-server ~]#cp -r /usr/share/easy-rsa/ /etc/openvpn/easy-rsa-server
#准备签发证书相关变量的配置文件
[12:09:41 root@openvpn-server ~]#cp /usr/share/doc/easy-rsa/vars.example /etc/openvpn/easy-rsa-server/3/vars
#建议修改给CA和OpenVPN服务器颁发的证书的有效期,可适当加长
[12:10:40 root@openvpn-server ~]#vim /etc/openvpn/easy-rsa-server/3/vars
#CA的证书有效期默为为10年,可以适当延长,比如:36500天
#set_var EASYRSA_CA_EXPIRE  3650
set_var EASYRSA_CA_EXPIRE   36500
#服务器证书默为为825天,可适当加长,比如:3650天
#set_var EASYRSA_CERT_EXPIRE    825
set_var EASYRSA_CERT_EXPIRE 3650

[12:13:37 root@openvpn-server ~]#tree /etc/openvpn/
/etc/openvpn/
├── client
├── easy-rsa-server
│   ├── 3 -> 3.0.8
│   ├── 3.0 -> 3.0.8
│   └── 3.0.8
│       ├── easyrsa
│       ├── openssl-easyrsa.cnf
│       ├── vars
│       └── x509-types
│           ├── ca
│           ├── client
│           ├── code-signing
│           ├── COMMON
│           ├── email
│           ├── kdc
│           ├── server
│           └── serverClient
├── server
└── server.conf

7 directories, 12 files
```

2.3 准备证书相关文件

2.3.1 初始化PKI和CA签发机构环境

2.3.1.1 脚本easyrsa帮助用法

```
[12:13:44 root@openvpn-server ~]#cd /etc/openvpn/easy-rsa-server/3
[12:15:01 root@openvpn-server 3]#pwd
/etc/openvpn/easy-rsa-server/3
```

```
[12:15:13 root@openvpn-server 3]#file ./easyrsa
./easyrsa: POSIX shell script, ASCII text executable
[12:15:20 root@openvpn-server 3]#./easyrsa
```

2.3.1.2 初始化PKI生成PKI相关目录和文件

```
[12:15:24 root@openvpn-server 3]#cd /etc/openvpn/easy-rsa-server/3
[12:16:02 root@openvpn-server 3]#pwd
/etc/openvpn/easy-rsa-server/3
[12:16:06 root@openvpn-server 3]#ls
easyrsa  openssl-easyrsa.cnf  vars  x509-types
#初始化数据,在当前目录下生成pki目录及相关文件
[12:16:07 root@openvpn-server 3]#./easyrsa init-pki

Note: using Easy-RSA configuration from: /etc/openvpn/easy-rsa-server/3.0.8/vars

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /etc/openvpn/easy-rsa-server/3/pki

[12:16:32 root@openvpn-server 3]#tree
.
├── easyrsa
├── openssl-easyrsa.cnf
├── pki
│   ├── openssl-easyrsa.cnf
│   ├── private
│   ├── reqs
│   └── safessl-easyrsa.cnf
├── vars
└── x509-types
├── ca
├── client
├── code-signing
├── COMMON
├── email
├── kdc
├── server
└── serverClient

4 directories, 13 files
```

2.3.2 创建CA机构

```
[12:17:24 root@openvpn-server 3]#cd /etc/openvpn/easy-rsa-server/3
[12:17:26 root@openvpn-server 3]#tree pki/
pki/
├── openssl-easyrsa.cnf
├── private
├── reqs
└── safessl-easyrsa.cnf

2 directories, 2 files
[12:17:30 root@openvpn-server 3]#./easyrsa build-ca nopass
```

```
Note: using Easy-RSA configuration from: /etc/openvpn/easy-rsa-server/3.0.8/vars
Using SSL: openssl OpenSSL 1.1.1g FIPS  21 Apr 2020
Generating RSA private key, 2048 bit long modulus (2 primes)
.....................................................................................................................................................
.....+++++
.....................................................+++++
e is 65537 (0x010001)
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----------------------------------------------

Common Name (eg: your user, host, or server name) [Easy-RSA CA]:  #接受默认值，直接回车

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/etc/openvpn/easy-rsa-server/3/pki/ca.crt    #生成自签名的证书文件

[12:17:51 root@openvpn-server 3]#tree pki/
pki/
├── ca.crt              #生成自签名的证书文件
├── certs_by_serial
├── index.txt
├── index.txt.attr
├── issued
├── openssl-easyrsa.cnf
├── private
│   └── ca.key          #生成私钥文件
├── renewed
│   ├── certs_by_serial
│   ├── private_by_serial
│   └── reqs_by_serial
├── reqs
├── revoked
│   ├── certs_by_serial
│   ├── private_by_serial
│   └── reqs_by_serial
├── safessl-easyrsa.cnf
└── serial

12 directories, 7 files

#生成CA相关的文件
[12:18:56 root@openvpn-server 3]#cat pki/serial
01
[12:19:56 root@openvpn-server 3]#ll pki/index.txt
-rw------- 1 root root 0 Jan 26 12:17 pki/index.txt
[12:20:04 root@openvpn-server 3]#ll pki/ca.crt pki/private/ca.key
-rw------- 1 root root 1204 Jan 26 12:17 pki/ca.crt
-rw------- 1 root root 1679 Jan 26 12:17 pki/private/ca.key
```

```
[12:20:27 root@openvpn-server 3]#cat pki/ca.crt
-----BEGIN CERTIFICATE-----
MIIDTTCCAjWgAwIBAgIUEH1WvM8tOv4gvppxJPKExDdhpzgwDQYJKoZIhvcNAQEL
BQAwFjEUMBIGA1UEAwwLRWFzeS1SU0EgQ0EwIBcNMjEwMTI2MDQxNzUxWhgPMjEy
MTAxMDIwNDE3NTFaMBYxFDASBgNVBAMMC0Vhc3ktUlNBIENBMIIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxZ0WEH0wzonl/AyMjAmZLCUy/lnGF2x1kWf1
ALLkyHke11YcLcc4jE3mAmSd4B8/PG6mTdtQNye3C0QBLJ4ts+g8mIFe3rGhA55O
qXea0vkKsztDxxoByTV+oGXPOAscTMApOiUP4l1Zk4Zxp8WJtlxkHfcKl3nVpKuH
QM3RDURxjp485dyc1I2K3IBoM2KwghKRburfeN2DqxCUZyM6WZQ8k9O25p4DEM9M
dSuizmiffA1Uoj5lVeH1wm1nGJjblXe/VuojYpUMlWVvIypyEy+52z6Ocd4QV8Mj
GD34BU0KT2xL9OG+MEknqMq/YJW4fMQFuj6ktZanIhPlvMjV8wIDAQABo4GQMIGN
MB0GA1UdDgQWBBQoyNfaufa2jaECRtJ+9ve4X5qANDBRBgNVHSMESjBIgBQoyNfa
ufa2jaECRtJ+9ve4X5qANKEapBgwFjEUMBIGA1UEAwwLRWFzeS1SU0EgQ0GCFBB9
VrzPLTr+IL6acSTyhMQ3Yac4MAwGA1UdEwQFMAMBAf8wCwYDVR0PBAQDAgEGMA0G
CSqGSIb3DQEBCwUAA4IBAQBnRWrY/2cYbtwdlKBupDmquWZwNFTOQJkrs/yDSwYi
mpmqrGQTMTn9nsyRm0kIEP2gNQHdlSUF2IP0axuRn6Cg3i44WQaLpRc/pf4kmoOX
AFJLNVgBv5xuB3CZYwLWTTHCo+r/ubwAQULWNLYbQ1HX5cpZ71W3Zy4MWJLGS3g+
xhdAs/vgYxMaoCQ/M9DE7EDe05IIFq3TEo9ganYM0VikRpfPWnnQTScXARQL8R2T
z7PfPDZfrDJSv7fA33T6nDi2xwdqiZUxtaiD15rCp56FprpvxSQvuPt/0TEm8T0R
zz//1nTdDTl/Let4RYT0VuVQm3xC4p1UThatACewiDEg
-----END CERTIFICATE-----
[12:20:42 root@openvpn-server 3]#openssl x509 -in pki/ca.crt -noout -text
Certificate:
Data:
Version: 3 (0x2)
Serial Number:
10:7d:56:bc:cf:2d:3a:fe:20:be:9a:71:24:f2:84:c4:37:61:a7:38
Signature Algorithm: sha256WithRSAEncryption
Issuer: CN = Easy-RSA CA
Validity
Not Before: Jan 26 04:17:51 2021 GMT
Not After : Jan  2 04:17:51 2121 GMT
Subject: CN = Easy-RSA CA
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public-Key: (2048 bit)
Modulus:
00:c5:9d:16:10:7d:30:ce:89:e5:fc:0c:8c:8c:09:
99:2c:25:32:fe:59:c6:17:6c:75:91:67:f5:00:b2:
e4:c8:79:1e:d7:56:1c:2d:c7:38:8c:4d:e6:02:64:
9d:e0:1f:3f:3c:6e:a6:4d:db:50:37:27:b7:0b:44:
01:2c:9e:2d:b3:e8:3c:98:81:5e:de:b1:a1:03:9e:
4e:a9:77:9a:d2:f9:0a:b3:3b:43:c7:1a:01:c9:35:
7e:a0:65:cf:38:0b:1c:4c:c0:29:3a:25:0f:e2:5d:
59:93:86:71:a7:c5:89:b6:5c:64:1d:f7:0a:97:79:
d5:a4:ab:87:40:cd:d1:0d:44:71:8e:9e:3c:e5:dc:
9c:d4:8d:8a:dc:80:68:33:62:b0:82:12:91:6e:ea:
df:78:dd:83:ab:10:94:67:23:3a:59:94:3c:93:d3:
b6:e6:9e:03:10:cf:4c:75:2b:a2:ce:68:9f:7c:0d:
54:a2:3e:65:55:e1:f5:c2:6d:67:18:98:db:95:77:
bf:56:ea:23:62:95:0c:95:65:6f:23:2a:72:13:2f:
b9:db:3e:8e:71:de:10:57:c3:23:18:3d:f8:05:4d:
0a:4f:6c:4b:f4:e1:be:30:49:27:a8:ca:bf:60:95:
b8:7c:c4:05:ba:3e:a4:b5:96:a7:22:13:c8:bc:c8:
```

d5:f3
Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Subject Key Identifier:
28:C8:D7:DA:B9:F6:B6:8D:A1:02:46:D2:7E:F6:F7:B8:5F:9A:80:34
X509v3 Authority Key Identifier:
keyid:28:C8:D7:DA:B9:F6:B6:8D:A1:02:46:D2:7E:F6:F7:B8:5F:9A:80:34
DirName:/CN=Easy-RSA CA
serial:10:7D:56:BC:CF:2D:3A:FE:20:BE:9A:71:24:F2:84:C4:37:61:A7:38

X509v3 Basic Constraints:
CA:TRUE
X509v3 Key Usage:
Certificate Sign, CRL Sign
Signature Algorithm: sha256WithRSAEncryption
67:45:6a:d8:ff:67:18:6e:dc:1d:94:a0:6e:a4:39:aa:b9:66:
70:34:54:ce:40:99:2b:b3:fc:83:4b:06:22:9a:99:aa:ac:64:
13:31:39:fd:9e:cc:91:9b:49:08:10:fd:a0:35:01:dd:95:25:
05:d8:83:f4:6b:1b:91:9f:a0:a0:de:2e:38:59:06:8b:a5:17:
3f:a5:fe:24:9a:83:97:00:52:4b:35:58:01:bf:9c:6e:07:70:
99:63:02:d6:4d:31:c2:a3:ea:ff:b9:bc:00:41:42:d6:34:b6:
1b:43:51:d7:e5:ca:59:ef:55:b7:67:2e:0c:58:92:c6:4b:78:
3e:c6:17:40:b3:fb:e0:63:13:1a:a0:24:3f:33:d0:c4:ec:40:
de:d3:92:08:16:ad:d3:12:8f:60:6a:76:0c:d1:58:a4:46:97:
cf:5a:79:d0:4d:27:17:01:14:0b:f1:1d:93:cf:b3:df:3c:36:
5f:ac:32:52:bf:b7:c0:df:74:fa:9c:38:b6:c7:07:6a:89:95:
31:b5:a8:83:d7:9a:c2:a7:9e:85:a6:ba:6f:c5:24:2f:b8:fb:
7f:d1:31:26:f1:3d:11:cf:3f:ff:d6:74:dd:0d:39:7f:2d:eb:
78:45:84:f4:56:e5:50:9b:7c:42:e2:9d:54:4e:16:ad:00:27:
b0:88:31:20

2.3.3 创建服务端证书申请

[12:21:19 root@openvpn-server 3]#cd /etc/openvpn/easy-rsa-server/3
[12:22:15 root@openvpn-server 3]#pwd
/etc/openvpn/easy-rsa-server/3
#创建服务器证书申请文件，其中server是文件前缀
[12:21:19 root@openvpn-server 3]#cd /etc/openvpn/easy-rsa-server/3
[12:22:15 root@openvpn-server 3]#pwd
/etc/openvpn/easy-rsa-server/3
[12:28:59 root@openvpn-server 3]#./easyrsa gen-req server nopass

Note: using Easy-RSA configuration from: /etc/openvpn/easy-rsa-server/3.0.8/vars
Using SSL: openssl OpenSSL 1.1.1g FIPS 21 Apr 2020
Generating a RSA private key
.....+++++
....+++++
writing new private key to '/etc/openvpn/easy-rsa-server/3/pki/easy-rsa-11496.syBWUN/tmpKB9HvC'

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.

```
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----------------------------------------------

Common Name (eg: your user, host, or server name) [server]:   接受Common Name默认

Keypair and certificate request completed. Your files are:
req: /etc/openvpn/easy-rsa-server/3/pki/reqs/server.req     #生成的申请文件
key: /etc/openvpn/easy-rsa-server/3/pki/private/server.key  #生成的私钥文件

[12:29:17 root@openvpn-server 3]#tree pki/
pki/
├── ca.crt
├── certs_by_serial
├── index.txt
├── index.txt.attr
├── issued
├── openssl-easyrsa.cnf
├── private
│   ├── ca.key
│   └── server.key   #私钥文件
├── renewed
│   ├── certs_by_serial
│   ├── private_by_serial
│   └── reqs_by_serial
├── reqs
│   └── server.req  #申请文件
├── revoked
│   ├── certs_by_serial
│   ├── private_by_serial
│   └── reqs_by_serial
├── safessl-easyrsa.cnf
└── serial

12 directories, 9 files
```

2.3.4 签发服务端证书

2.3.4.1 查看颁发证书命令用法

```
[12:23:55 root@openvpn-server 3]#cd /etc/openvpn/easy-rsa-server/3
[12:25:06 root@openvpn-server 3]#./easyrsa help sign

Note: using Easy-RSA configuration from: /etc/openvpn/easy-rsa-server/3.0.8/vars

sign-req
Sign a certificate request of the defined type.  must be a known
type such as 'client', 'server', 'serverClient', or 'ca' (or a user-added type.)

This request file must exist in the reqs/ dir and have a .req file
extension. See import-req below for importing reqs from other sources.
```

2.3.4.2 颁发服务端证书

```
#将上面server.req的申请,颁发server类型的证书
[12:30:25 root@openvpn-server 3]#./easyrsa sign server server

Note: using Easy-RSA configuration from: /etc/openvpn/easy-rsa-server/3.0.8/vars
Using SSL: openssl OpenSSL 1.1.1g FIPS  21 Apr 2020

You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a server certificate for 3650 days:
vars文件指定的有效期
subject=
commonName                = server

Type the word 'yes' to continue, or any other input to abort.
Confirm request details: yes  #输入yes回车
Using configuration from /etc/openvpn/easy-rsa-server/3/pki/easy-rsa-11533.UjZ9m9/tmp.x
m4NL
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'server'
Certificate is to be certified until Jan 24 04:31:32 2031 GMT (3650 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /etc/openvpn/easy-rsa-server/3/pki/issued/server.crt
#生成服务器证书文件
```

2.3.4.3 验证结果

```
[12:31:32 root@openvpn-server 3]#cd /etc/openvpn/easy-rsa-server/3
[12:32:54 root@openvpn-server 3]#tree pki/
pki/
├── ca.crt
├── certs_by_serial
│   └── 5D3B930AA9D6B0AF69E65FA76C6251C4.pem   #服务器证书文件
├── index.txt
├── index.txt.attr
├── index.txt.attr.old
├── index.txt.old
├── issued
│   └── server.crt          #服务器证书文件
├── openssl-easyrsa.cnf
├── private
│   ├── ca.key
│   └── server.key
├── renewed
```

```
│   ├── certs_by_serial
│   ├── private_by_serial
│   └── reqs_by_serial
├── reqs
│   └── server.req
├── revoked
│   ├── certs_by_serial
│   ├── private_by_serial
│   └── reqs_by_serial
├── safessl-easyrsa.cnf
├── serial
└── serial.old

12 directories, 14 files
[12:32:57 root@openvpn-server 3]#diff pki/certs_by_serial/5D3B930AA9D6B0AF69E65FA76C6
51C4.pem pki/issued/server.crt
[12:34:24 root@openvpn-server 3]#ll !*
ll pki/certs_by_serial/5D3B930AA9D6B0AF69E65FA76C6251C4.pem pki/issued/server.crt
-rw------- 1 root root 4608 Jan 26 12:31 pki/certs_by_serial/5D3B930AA9D6B0AF69E65FA76C
251C4.pem
-rw------- 1 root root 4608 Jan 26 12:31 pki/issued/server.crt
#证书相关文件
[12:34:35 root@openvpn-server 3]#cat pki/serial
5D3B930AA9D6B0AF69E65FA76C6251C5
[12:35:09 root@openvpn-server 3]#cat pki/index.txt
V    310124043132Z        5D3B930AA9D6B0AF69E65FA76C6251C4    unknown    /CN=server
[12:35:29 root@openvpn-server 3]#cat pki/serial.old
5d3b930aa9d6b0af69e65fa76c6251c4
```

2.3.5 创建Diifie-Hellman密钥

2.3.5.1 Diffie-Hellman 算法

Diffie-Hellman 密钥交换方法，由惠特菲尔德·迪菲（Bailey Whitfield Diffie）、马丁·赫尔曼（Martin Edward Hellman）于1976年发表。它是一种安全协议，让双方在完全没有对方任何预先息的
条件下通过不安全信道建立起一个密钥，这个密钥一般作为“对称加密”的密钥而被双方在后续数据输中使
用。DH数学原理是base离散对数问题。做类似功能的还有非对称加密类算法，如：RSA。其应用非广泛，在
SSH、VPN、Https等都有应用。
wiki参考链接: https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange

2.3.5.2 创建Diffie-Hellman密钥

```
[12:35:32 root@openvpn-server 3]#cd /etc/openvpn/easy-rsa-server/3
[12:37:16 root@openvpn-server 3]#pwd
/etc/openvpn/easy-rsa-server/3
#方法1
[12:37:18 root@openvpn-server 3]#./easyrsa gen-dh

Note: using Easy-RSA configuration from: /etc/openvpn/easy-rsa-server/3.0.8/vars
```

```
Using SSL: openssl OpenSSL 1.1.1g FIPS  21 Apr 2020
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
.....................+  #需要等一会
DH parameters of size 2048 created at /etc/openvpn/easy-rsa-server/3/pki/dh.pem
#查看生成的文件
[12:37:54 root@openvpn-server 3]#ll pki/dh.pem
-rw------- 1 root root 424 Jan 26 12:37 pki/dh.pem
[12:38:53 root@openvpn-server 3]#cat pki/dh.pem
-----BEGIN DH PARAMETERS-----
MIIBCAKCAQEAypLcHbOieMk67cANDM+IBDD0w6SP3vJ9vY4Bz58SX017qLI9qLSD
CCRWIF7Y57zVHkqrXHsVJpSXZPBGTWPKg6LsMYrSrQctxajikAzkA2xqlezJquFz
oGkhR9P1xkA7Kbj0+w0/0lOxkPuVq6WbqSa2JBNaYmOzXRz1I4BZnR0CCKoI/WMB
WZ2cTeQcVI1AYqN9prOwWZwXZks420RUmnDXAL7BtvfElyKtgiZXPzQpiF4Psjhb
gNAwBnHJiV1vj1dTLg6CtU9e+yuk7nuz+74OhF3y2jfF3odg+7ZGWNlkoMP1wq6Z
eONjJO9n3cxLInPXDhJ4NfbwTh6LOKQ6YwIBAg==
-----END DH PARAMETERS-----
#方法2
[12:39:01 root@openvpn-server 3]#openssl dhparam -out /root/dh2048.pem 2048
```

2.3.6 准备客户端证书环境

上面服务端证书配置完成，下面是配置客户端证书

```
[13:48:50 root@openvpn-server ~]#cp -r /usr/share/easy-rsa/ /etc/openvpn/easy-rsa-client
#可选
[13:49:30 root@openvpn-server ~]#cp /usr/share/doc/easy-rsa/vars.example /etc/openvpn/e
sy-rsa-client/3/vars
[13:50:40 root@openvpn-server ~]#cd /etc/openvpn/easy-rsa-client/3
[13:50:53 root@openvpn-server 3]#pwd
/etc/openvpn/easy-rsa-client/3
[13:50:54 root@openvpn-server 3]#ls
easyrsa  openssl-easyrsa.cnf  vars  x509-types
[13:50:57 root@openvpn-server 3]#tree
.
├── easyrsa
├── openssl-easyrsa.cnf
├── vars
└── x509-types
├── ca
├── client
├── code-signing
├── COMMON
├── email
├── kdc
├── server
└── serverClient

1 directory, 11 files
#生成证书申请所需目录pki和文件
[13:52:04 root@openvpn-server 3]#tree
.
├── easyrsa
├── openssl-easyrsa.cnf
```

```
├── pki
│   ├── openssl-easyrsa.cnf
│   ├── private
│   ├── reqs
│   └── safessl-easyrsa.cnf
├── vars
└── x509-types
├── ca
├── client
├── code-signing
├── COMMON
├── email
├── kdc
├── server
└── serverClient

4 directories, 13 files
```

2.3.7 创建客户端证书申请

```
[13:52:08 root@openvpn-server 3]#cd /etc/openvpn/easy-rsa-client/3
[13:52:41 root@openvpn-server 3]#pwd
/etc/openvpn/easy-rsa-client/3
[13:52:45 root@openvpn-server 3]#./easyrsa gen-req zhangzhuo nopass

Note: using Easy-RSA configuration from: /etc/openvpn/easy-rsa-client/3.0.8/vars
Using SSL: openssl OpenSSL 1.1.1g FIPS  21 Apr 2020
Generating a RSA private key
....+++++
.....+++++
writing new private key to '/etc/openvpn/easy-rsa-client/3/pki/easy-rsa-11853.FRpJUj/tmp.Jr
3E6'
-----

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----

Common Name (eg: your user, host, or server name) [zhangzhuo]:  #接受默认回车

Keypair and certificate request completed. Your files are:
req: /etc/openvpn/easy-rsa-client/3/pki/reqs/zhangzhuo.req       #申请证书文件
key: /etc/openvpn/easy-rsa-client/3/pki/private/zhangzhuo.key    #私钥文件

#生成两个新文件
[13:53:11 root@openvpn-server 3]#tree
.
├── easyrsa
├── openssl-easyrsa.cnf
├── pki
```

```
│   ├── openssl-easyrsa.cnf
│   ├── private
│   │   └── zhangzhuo.key
│   ├── reqs
│   │   └── zhangzhuo.req
│   └── safessl-easyrsa.cnf
├── vars
└── x509-types
├── ca
├── client
├── code-signing
├── COMMON
├── email
├── kdc
├── server
└── serverClient

4 directories, 15 files
```

2.3.8 签发客户端证书

```
[13:54:22 root@openvpn-server 3]#cd /etc/openvpn/easy-rsa-server/3
[13:54:54 root@openvpn-server 3]#pwd
/etc/openvpn/easy-rsa-server/3
#将客户端证书请求文件复制到CA的工作目录
[13:55:38 root@openvpn-server 3]#./easyrsa import-req /etc/openvpn/easy-rsa-client/3/pki/r
qs/zhangzhuo.req zhangzhuo

Note: using Easy-RSA configuration from: /etc/openvpn/easy-rsa-server/3.0.8/vars
Using SSL: openssl OpenSSL 1.1.1g FIPS  21 Apr 2020

The request has been successfully imported with a short name of: zhangzhuo
You may now use this name to perform signing operations on this request.

[13:55:45 root@openvpn-server 3]#tree pki/
pki/
├── ca.crt
├── certs_by_serial
│   └── 5D3B930AA9D6B0AF69E65FA76C6251C4.pem
├── dh.pem
├── index.txt
├── index.txt.attr
├── index.txt.attr.old
├── index.txt.old
├── issued
│   └── server.crt
├── openssl-easyrsa.cnf
├── private
│   ├── ca.key
│   └── server.key
├── renewed
│   ├── certs_by_serial
│   ├── private_by_serial
│   └── reqs_by_serial
```

```
├── reqs
│   ├── server.req
│   └── zhangzhuo.req    #导入文件
├── revoked
│   ├── certs_by_serial
│   ├── private_by_serial
│   └── reqs_by_serial
├── safessl-easyrsa.cnf
├── serial
└── serial.old

12 directories, 16 files
[13:56:13 root@openvpn-server 3]#ll pki/reqs/zhangzhuo.req /etc/openvpn/easy-rsa-client/3
pki/reqs/zhangzhuo.req
-rw------- 1 root root 891 Jan 26 13:53 /etc/openvpn/easy-rsa-client/3/pki/reqs/zhangzhuo.r
q
-rw------- 1 root root 891 Jan 26 13:55 pki/reqs/zhangzhuo.req

#修改给客户端颁发的证书的有效期
[13:57:19 root@openvpn-server 3]#vim vars
set_var EASYRSA_CERT_EXPIRE 180    #修改之前的3650为180

#签发客户端证书
[13:58:41 root@openvpn-server 3]#./easyrsa sign client zhangzhuo

Note: using Easy-RSA configuration from: /etc/openvpn/easy-rsa-server/3.0.8/vars
Using SSL: openssl OpenSSL 1.1.1g FIPS  21 Apr 2020

You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a client certificate for 180 days:

subject=
commonName                = zhangzhuo

Type the word 'yes' to continue, or any other input to abort.
Confirm request details: yes    #输入yes后回车
Using configuration from /etc/openvpn/easy-rsa-server/3/pki/easy-rsa-11993.LhxZXn/tmp.Fl
0WC
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'zhangzhuo'
Certificate is to be certified until Jul 25 05:59:46 2021 GMT (180 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /etc/openvpn/easy-rsa-server/3/pki/issued/zhangzhuo.crt
#证书文件
```

```
[13:59:46 root@openvpn-server 3]#tree pki/
pki/
├── ca.crt
├── certs_by_serial
│   ├── 5D3B930AA9D6B0AF69E65FA76C6251C4.pem
│   └── 8EB7E418B1FE1715BCBB73A513498893.pem
├── dh.pem
├── index.txt
├── index.txt.attr
├── index.txt.attr.old
├── index.txt.old
├── issued
│   ├── server.crt
│   └── zhangzhuo.crt     #生成客户端证书
├── openssl-easyrsa.cnf
├── private
│   ├── ca.key
│   └── server.key
├── renewed
│   ├── certs_by_serial
│   ├── private_by_serial
│   └── reqs_by_serial
├── reqs
│   ├── server.req
│   └── zhangzhuo.req
├── revoked
│   ├── certs_by_serial
│   ├── private_by_serial
│   └── reqs_by_serial
├── safessl-easyrsa.cnf
├── serial
└── serial.old

12 directories, 18 files
[14:01:01 root@openvpn-server 3]#cat pki/index.txt
V    310124043132Z        5D3B930AA9D6B0AF69E65FA76C6251C4    unknown    /CN=server
V    210725055946Z        8EB7E418B1FE1715BCBB73A513498893    unknown    /CN=zhangzh
o
[14:01:47 root@openvpn-server 3]#ll pki/issued/
total 16
-rw------- 1 root root 4608 Jan 26 12:31 server.crt
-rw------- 1 root root 4499 Jan 26 13:59 zhangzhuo.crt
[14:01:57 root@openvpn-server 3]#ll pki/certs_by_serial/
total 16
-rw------- 1 root root 4608 Jan 26 12:31 5D3B930AA9D6B0AF69E65FA76C6251C4.pem
-rw------- 1 root root 4499 Jan 26 13:59 8EB7E418B1FE1715BCBB73A513498893.pem
```

2.3.9 将CA和服务器证书相关文件复制到服务器相应的目录

```
[14:03:01 root@openvpn-server ~]#mkdir /etc/openvpn/certs
[14:03:11 root@openvpn-server ~]#cp /etc/openvpn/easy-rsa-server/3/pki/ca.crt /etc/openv
n/certs/
[14:03:43 root@openvpn-server ~]#cp /etc/openvpn/easy-rsa-server/3/pki/issued/server.crt /
tc/openvpn/certs/
```

```
[14:04:05 root@openvpn-server ~]#cp /etc/openvpn/easy-rsa-server/3/pki/private/server.key
/etc/openvpn/certs/
[14:04:24 root@openvpn-server ~]#cp /etc/openvpn/easy-rsa-server/3/pki/dh.pem /etc/ope
vpn/certs/
[14:04:41 root@openvpn-server ~]#ll /etc/openvpn/certs/
total 20
-rw------- 1 root root 1204 Jan 26 14:03 ca.crt
-rw------- 1 root root  424 Jan 26 14:04 dh.pem
-rw------- 1 root root 4608 Jan 26 14:04 server.crt
-rw------- 1 root root 1704 Jan 26 14:04 server.key
```

2.3.10 将客户端私钥与证书相关文件复制到服务器相关的目录

```
[14:04:59 root@openvpn-server ~]#mkdir /etc/openvpn/client/zhangzhuo
[14:06:59 root@openvpn-server ~]#find /etc/openvpn/ -name "zhangzhuo.key" -o -name "zh
ngzhuo.crt" -o -name "ca.crt"
/etc/openvpn/easy-rsa-server/3.0.8/pki/issued/zhangzhuo.crt
/etc/openvpn/easy-rsa-server/3.0.8/pki/ca.crt
/etc/openvpn/easy-rsa-client/3.0.8/pki/private/zhangzhuo.key
/etc/openvpn/certs/ca.crt
[14:08:50 root@openvpn-server ~]#find /etc/openvpn/ \( -name "zhangzhuo.key" -o -name "
hangzhuo.crt" -o -name "ca.crt" \) -exec cp {} /etc/openvpn/client/zhangzhuo \;
[14:09:08 root@openvpn-server ~]#ll /etc/openvpn/client/zhangzhuo/
total 16
-rw------- 1 root root 1204 Jan 26 14:09 ca.crt
-rw------- 1 root root 4499 Jan 26 14:09 zhangzhuo.crt
-rw------- 1 root root 1704 Jan 26 14:09 zhangzhuo.key
```

2.4 准备 OpenVPN 服务器配置文件

2.4.1 服务器端配置文件说明

```
#server.conf文件中以#或;开头的行都为注释
[root@centos8 ~]#grep -Ev "^#|^$" /etc/openvpn/server.conf
;local a.b.c.d  #本机监听IP,默认为本机所有IP
port 1194       #端口
;proto tcp      #协议,生产推荐使用TCP
proto udp #默认协议
;dev tap   #创建一个以太网隧道，以太网使用tap,一个tap设备允许完整的以太网帧通过Openvpn隧
，可提供非ip协议的支持，比如IPX协议和AppleTalk协议,tap等同于一个以太网设备，它操作第二层
据包如以太网数据帧。
dev tun    #创建一个路由IP隧道，生产推存使用tun.互联网使用tun,一个tun设备大多时候，被用于
于IP协议的通讯。tun模拟了网络层设备，操作第三层数据包比如IP数据封包。
;dev-node MyTap  #TAP-Win32适配器。非windows不需要配置
ca ca.crt       #ca证书文件
cert server.crt  #服务器证书文件
key server.key   #服务器私钥文件
dh dh2048.pem    #dh参数文件
;topology subnet
server 10.8.0.0 255.255.255.0  #客户端连接后分配IP的地址池，服务器默认会占用第一个IP10.8.0.
将做为客户端的网关
ifconfig-pool-persist ipp.txt  #为客户端分配固定IP，不需要配置,建议注释
```

;server-bridge 10.8.0.4 255.255.255.0 10.8.0.50 10.8.0.100 #配置网桥模式，不需要配置,建议注释
;server-bridge
;push "route 192.168.10.0 255.255.255.0" #给客户端生成的到达服务器后面网段的静态路由，下跳为openvpn服务器的10.8.0.1
;push "route 192.168.20.0 255.255.255.0" #推送路由信息到客户端，以允许客户端能够连接到服器背后的其它私有子网
;client-config-dir ccd #为指定的客户端添加路由，此路由通常是客户端后面的内网网段而不是服务的，也不需要设置
;route 192.168.40.128 255.255.255.248
;client-config-dir ccd
;route 10.9.0.0 255.255.255.252
;learn-address ./script #运行外部脚本，创建不同组的iptables规则，无需配置
;push "redirect-gateway def1 bypass-dhcp" #启用后，客户端所有流量都将通过VPN服务器，因生产一般无需配置此项
;push "dhcp-option DNS 208.67.222.222" #推送DNS服务器，不需要配置
;push "dhcp-option DNS 208.67.220.220"
;client-to-client #允许不同的client直接通信,不安全,生产环境一般无需要配置
;duplicate-cn #多个用户共用一个证书，一般用于测试环境，生产环境都是一个用一个证书,无需开启
keepalive 10 120 #设置服务端检测的间隔和超时时间，默认为每10秒ping一次，如果 120秒有回应则认为对方已经down
tls-auth ta.key 0 #访止DoS等攻击的安全增强配置,可以使用以下命令来生成：openvpn --genkey --secret ta.key #服务器和每个客户端都需要拥有该密钥的一个拷贝。第二个参数在服务器应该为’0’，在客户端应该为’1’
cipher AES-256-CBC #加密算法
;compress lz4-v2 #启用Openvpn2.4.X新版压缩算法
;push "compress lz4-v2" #推送客户端使用新版压缩算法,和下面的comp-lzo不要同时使用
;comp-lzo #旧户端兼容的压缩配置，需要客户端配置开启压缩,openvpn2.4.X等新版可以不用启
;max-clients 100 #最大客户端数
;user nobody #运行openvpn服务的用户和组
;group nobody
persist-key #重启VPN服务时默认会重新读取key文件，开启此配置后保留使用第一次的key文,生产环境无需开启
persist-tun #启用此配置后,当重启vpn服务时，一直保持tun或者tap设备是up的，否则会先don然后再up,生产环境无需开启
status openvpn-status.log #openVPN状态记录文件，每分钟会记录一次
;log openvpn.log #第一种日志记录方式,并指定日志路径，log会在openvpn启动的时候清空志文件,不建议使用
;log-append openvpn.log #第二种日志记录方式,并指定日志路径，重启openvpn后在之前的日志面追加新的日志,生产环境建议使用
verb 3 #设置日志级别，0-9，级别越高记录的内容越详细,0 表示静默运行，只记录致命误,4 表示合理的常规用法,5 和 6 可以帮助调试连接错误。9 表示极度冗余，输出非常详细的日志信息
;mute 20 #相同类别的信息只有前20条会输出到日志文件中
explicit-exit-notify 1 #通知客户端，在服务端重启后自动重新连接，仅能用于udp模式，tcp模式需要配置即可实现断开重新连接,且开启此项后tcp配置后将导致openvpn服务无法启动,所以tcp时必不能开启此项

2.4.2 修改服务器端配置文件

[14:09:29 root@openvpn-server ~]#vim /etc/openvpn/server.conf
port 1194 #开启端口
proto tcp #使用的协议还可以使用udp
dev tun #创建一个路由IP隧道

```
ca /etc/openvpn/certs/ca.crt               #ca证书文件位置
cert /etc/openvpn/certs/server.crt         #服务证书文件位置
key /etc/openvpn/certs/server.key          #服务私钥文件位置
dh /etc/openvpn/certs/dh.pem               #dh参数文件，也就是密钥交换算法文件
server 10.0.0.0 255.255.255.0              #客户端连接后分配IP的地址池
push "route 172.30.0.0 255.255.255.0"      #给客户端生成的到达服务器后面网段的静态路由
keepalive 10 120                           #设置服务端检测的间隔和超时时间
cipher AES-256-CBC                         #加密算法
compress lz4-v2                            #启用Openvpn2.4.X新版压缩算法
push "compress lz4-v2"                     #推送客户端使用新版压缩算法
max-clients 2048                           #最大客户端数
user openvpn                               #运行openvpn服务的用户和组
group openvpn
status /var/log/openvpn/openvpn-status.log  #openVPN状态记录文件，每分钟会记录一次
log-append /var/log/openvpn/openvpn.log     #第二种日志记录方式
verb 3                                     #设置日志级别
mute 20                                    #相同类别的信息只有前20条会输出到日志文件中
#准备目志相关目录
[14:22:29 root@openvpn-server ~]#getent passwd openvpn
openvpn:x:988:985:OpenVPN:/etc/openvpn:/sbin/nologin
[14:23:02 root@openvpn-server ~]#mkdir /var/log/openvpn
[14:23:31 root@openvpn-server ~]#chown openvpn: /var/log/openvpn
[14:23:41 root@openvpn-server ~]#ll -d /var/log/openvpn/
drwxr-xr-x 2 openvpn openvpn 6 Jan 26 14:23 /var/log/openvpn/
```

2.5 准备iptables规则和内核参数

```
#在服务器开启ip_forward转发功能
[14:23:59 root@openvpn-server ~]#echo net.ipv4.ip_forward = 1 >>/etc/sysctl.conf
[14:25:03 root@openvpn-server ~]#sysctl -p
#添加SNAT规则
[14:26:28 root@openvpn-server ~]#echo "iptables -t nat -APOSTROUTING -s 10.0.0.0/24 -j M
SQUERADE" >>/etc/rc.d/rc.local
[14:27:11 root@openvpn-server ~]#chmod +x /etc/rc.d/rc.local
[14:29:03 root@openvpn-server ~]#/etc/rc.d/rc.local
[14:29:20 root@openvpn-server ~]#iptables -t nat -vnL
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target     prot opt in     out     source               destination

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target     prot opt in     out     source               destination
0     0 MASQUERADE  all  --  *      *       10.0.0.0/24          0.0.0.0/0

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target     prot opt in     out     source               destination
```

2.6 启动OpenVPN服务

2.6.1 启动 OpenVPN 服务

```
[14:33:01 root@centos7 ~]#rpm -ql openvpn | grep systemd
/usr/lib/systemd/system/openvpn-client@.service
/usr/lib/systemd/system/openvpn-server@.service
/usr/lib/systemd/system/openvpn@.service
/usr/share/doc/openvpn-2.4.10/README.systemd
#centos8缺失unit文件，从Centos7复制文件
[14:29:21 root@openvpn-server ~]#rpm -ql openvpn | grep systemd
/usr/lib/systemd/system/openvpn-client@.service
/usr/lib/systemd/system/openvpn-server@.service
/usr/share/doc/openvpn/README.systemd
[14:33:18 root@centos7 ~]#cat /usr/lib/systemd/system/openvpn@.service
[Unit]
Description=OpenVPN Robust And Highly Flexible Tunneling Application On %I
After=network.target

[Service]
Type=notify
PrivateTmp=true
ExecStart=/usr/sbin/openvpn --cd /etc/openvpn/ --config %i.conf

[Install]
WantedBy=multi-user.target
[14:34:46 root@centos7 ~]#scp /lib/systemd/system/openvpn@.service 39.98.146.209:/lib/sys
emd/system/
#启动OpenVPN服务,注意service名称和文件名不一致
[14:54:27 root@openvpn-server ~]#systemctl daemon-reload
[14:54:33 root@openvpn-server ~]#systemctl enable --now openvpn@server
Created symlink /etc/systemd/system/multi-user.target.wants/openvpn@server.service → /us
/lib/systemd/system/openvpn@.service.
```

2.6.2 查看服务状态

```
[14:54:42 root@openvpn-server ~]#systemctl status openvpn@server
● openvpn@server.service - OpenVPN Robust And Highly Flexible Tunneling Application On
erver
Loaded: loaded (/usr/lib/systemd/system/openvpn@.service; enabled; vendor preset: disabled

Active: active (running) since Tue 2021-01-26 14:54:42 CST; 29s ago
Main PID: 13647 (openvpn)
Status: "Initialization Sequence Completed"
Tasks: 1 (limit: 22788)
Memory: 1.3M
CGroup: /system.slice/system-openvpn.slice/openvpn@server.service
└─13647 /usr/sbin/openvpn --cd /etc/openvpn/ --config server.conf

Jan 26 14:54:42 openvpn-server systemd[1]: Starting OpenVPN Robust And Highly Flexible Tu
neling Applicati>
Jan 26 14:54:42 openvpn-server systemd[1]: Started OpenVPN Robust And Highly Flexible Tu
neling Applicatio>

#注意端口号
[14:55:33 root@openvpn-server ~]#ss -ntlp
State   Recv-Q  Send-Q   Local Address:Port   Peer Address:Port
LISTEN  0       32             0.0.0.0:1194        0.0.0.0:*     users:(("openvpn",pid=13647,fd=7))
```

```
LISTEN  0      128          0.0.0.0:5355       0.0.0.0:*     users:(("systemd-resolve",pid=963,fd=13)

LISTEN  0      128          0.0.0.0:111        0.0.0.0:*     users:(("systemd",pid=1,fd=59))
LISTEN  0      128          0.0.0.0:22         0.0.0.0:*     users:(("sshd",pid=1009,fd=5))
LISTEN  0      128             [::]:5355          [::]:*     users:(("systemd-resolve",pid=963,fd=15))
LISTEN  0      128             [::]:111           [::]:*     users:(("systemd",pid=1,fd=66))

[14:55:52 root@openvpn-server ~]#cat /var/log/openvpn/openvpn.log

[14:56:20 root@openvpn-server ~]#ip a
1: lo:  mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host
valid_lft forever preferred_lft forever
2: eth0:  mtu 1500 qdisc fq_codel state UP group default qlen 1000
link/ether 00:16:3e:14:71:66 brd ff:ff:ff:ff:ff:ff
inet 172.30.0.1/24 brd 172.30.0.255 scope global dynamic noprefixroute eth0
valid_lft 315348625sec preferred_lft 315348625sec
inet6 fe80::216:3eff:fe14:7166/64 scope link
valid_lft forever preferred_lft forever
4: tun0:  mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 100
link/none
inet 10.8.0.1 peer 10.8.0.2/32 scope global tun0
valid_lft forever preferred_lft forever
inet6 fe80::6f54:d0a2:9870:cc6b/64 scope link stable-privacy
valid_lft forever preferred_lft forever
[14:56:36 root@openvpn-server ~]#route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         172.30.0.253    0.0.0.0         UG    100    0        0 eth0
10.8.0.0        10.8.0.2        255.255.255.0   UG    0      0        0 tun0
10.8.0.2        0.0.0.0         255.255.255.255 UH    0      0        0 tun0
172.30.0.0      0.0.0.0         255.255.255.0   U     100    0        0 eth0
```

验证tun网卡设备

```
[14:56:50 root@openvpn-server ~]#ifconfig tun0
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::6f54:d0a2:9870:cc6b  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 100  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 17  bytes 1016 (1016.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

2.7 准备OpenVPN客户端配置文件

2.7.1 客户端默认范例配置文件说明

```
[14:59:50 root@openvpn-server ~]#grep -Ev "^(#|;)|^$" /usr/share/doc/openvpn/sample/sa
ple-config-files/client.conf
client                  #声明自己是个客户端
```

```
dev tun                 #接口类型，必须和服务端保持一致
proto udp               #协议类型，必须和服务端保持一致
remote my-server-1 1194 #server端的ip和端口，可以写域名但是需要可以解析成IP
resolv-retry infinite   #如果是写的server端的域名，那么就始终解析，如果域名发生变化，会重新接到新的域名对应的IP
nobind                  #本机不绑定监听端口，客户端是随机打开端口连接到服务端的1194
persist-key
persist-tun
ca ca.crt
cert client.crt
key client.key
remote-cert-tls server   #指定采用服务器证书校验方式
tls-auth ta.key 1
cipher AES-256-CBC
verb 3
```

2.7.2 生成客户端用户的配置文件

```
#生成客户端文件,文件后缀必须为.ovpn
[15:00:09 root@openvpn-server ~]#grep -Ev "^(#|;)|^$" /usr/share/doc/openvpn/sample/sample-config-files/client.conf >/etc/openvpn/client/zhangzhuo/client.ovpn
#修改配置文件,内容如下
[15:02:26 root@openvpn-server ~]#vim /etc/openvpn/client/zhangzhuo/client.ovpn
[15:04:30 root@openvpn-server ~]#cat /etc/openvpn/client/zhangzhuo/client.ovpn
client
dev tun
proto tcp
remote 39.98.146.209 1194    #生产中为OpenVPN公网IP
resolv-retry infinite
nobind
#persist-key
#persist-tun
ca ca.crt
cert zhangzhuo.crt
key zhangzhuo.key
remote-cert-tls server
#tls-auth ta.key 1
cipher AES-256-CBC
verb 3                  #此值不能随意指定，否则无法通信
compress lz4-v2         #此项在OpenVPN2.4.X版本使用，需要和服务器端保持一致，不指定默认comp-lz压缩
```

2.8 Windows 配置部署 OpenVPN 客户端

2.8.1 Windows 安装 OpenVPN 客户端

官方客户端下载地址：

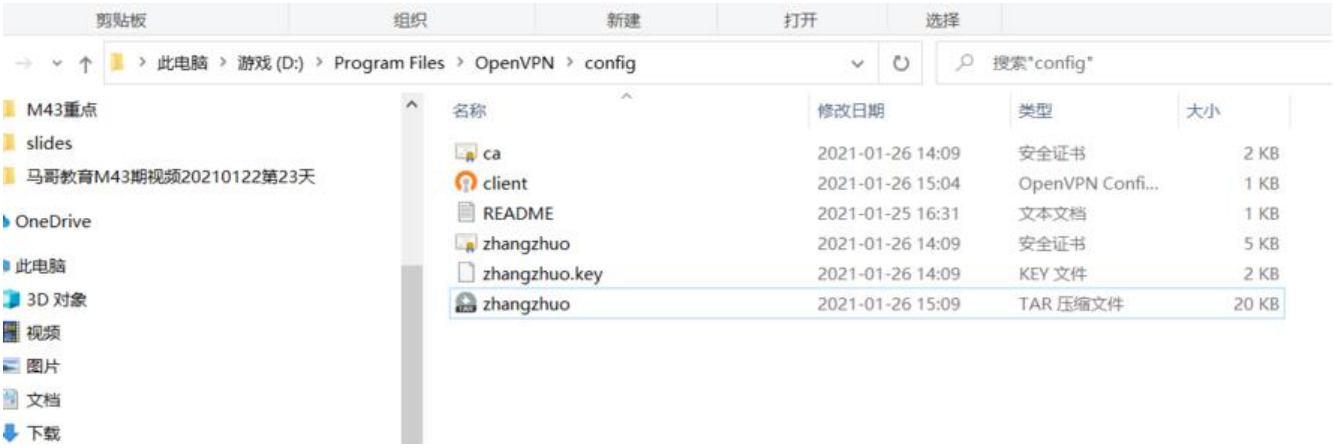
https://openvpn.net/community-downloads/

下载安装就可以了

2.8.2 Windows客户端配置准备

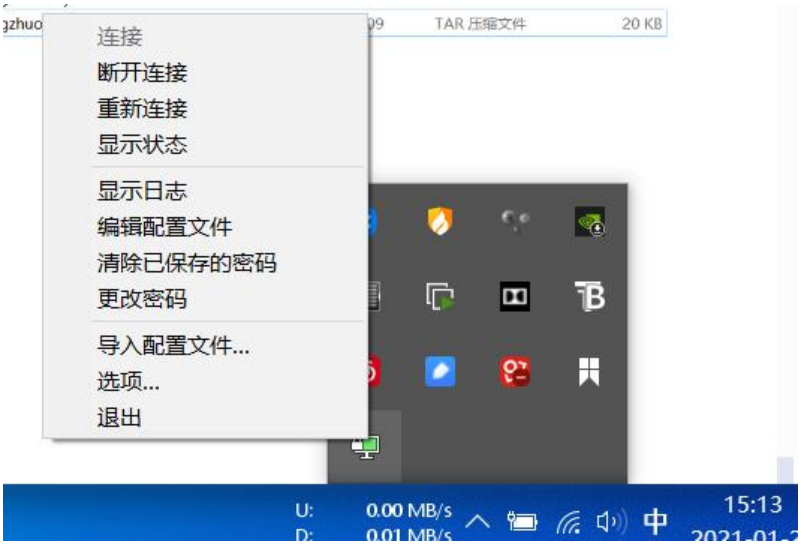
```
#在服务器打包证书并下载发送给windows客户端
[15:04:58 root@openvpn-server ~]#cd /etc/openvpn/client/zhangzhuo/
[15:08:34 root@openvpn-server zhangzhuo]#pwd
/etc/openvpn/client/zhangzhuo
[15:08:38 root@openvpn-server zhangzhuo]#tar cf zhangzhuo.tar ./
tar: ./zhangzhuo.tar: file is the archive; not dumped
[15:09:12 root@openvpn-server zhangzhuo]#ll
total 40
-rw------- 1 root root  1204 Jan 26 14:09 ca.crt
-rw-r--r-- 1 root root   237 Jan 26 15:04 client.ovpn
-rw------- 1 root root  4499 Jan 26 14:09 zhangzhuo.crt
-rw------- 1 root root  1704 Jan 26 14:09 zhangzhuo.key
-rw-r--r-- 1 root root 20480 Jan 26 15:09 zhangzhuo.tar

[15:09:13 root@openvpn-server zhangzhuo]#tar -tf zhangzhuo.tar
./
./zhangzhuo.crt
./ca.crt
./zhangzhuo.key
./client.ovpn
```

放置到windows客户端的 C:\Program Files\OpenVPN\config 目录下



开打OpenVPN GUI

进行连接



绿色之后就表示连接正常

2.8.4 Windows客户端验证通信

2.8.4.1 在Windows 客户端测试访问OpenVPN后端服务器

后端服务器显示是来自于OpenVPN服务器的连接

2.8.4.2 观察OpenVPN服务器日志

[15:18:52 root@openvpn-server zhangzhuo]#tail /var/log/openvpn/openvpn.log -f -n0
Tue Jan 26 15:18:57 2021 TCP connection established with [AF_INET]110.17.5.83:20328
Tue Jan 26 15:18:58 2021 110.17.5.83:20328 TLS: Initial packet from [AF_INET]110.17.5.83:203
8, sid=0f61dc6f b6fc7583
Tue Jan 26 15:18:58 2021 110.17.5.83:20328 VERIFY OK: depth=1, CN=Easy-RSA CA
Tue Jan 26 15:18:58 2021 110.17.5.83:20328 VERIFY OK: depth=0, CN=zhangzhuo

```
Tue Jan 26 15:18:58 2021 110.17.5.83:20328 peer info: IV_VER=2.4.10
Tue Jan 26 15:18:58 2021 110.17.5.83:20328 peer info: IV_PLAT=win
Tue Jan 26 15:18:58 2021 110.17.5.83:20328 peer info: IV_PROTO=2
Tue Jan 26 15:18:58 2021 110.17.5.83:20328 peer info: IV_NCP=2
Tue Jan 26 15:18:58 2021 110.17.5.83:20328 peer info: IV_CIPHERS=AES-256-GCM:AES-128-GCM:AES-256-CBC
Tue Jan 26 15:18:58 2021 110.17.5.83:20328 peer info: IV_LZ4=1
Tue Jan 26 15:18:58 2021 110.17.5.83:20328 peer info: IV_LZ4v2=1
Tue Jan 26 15:18:58 2021 110.17.5.83:20328 peer info: IV_LZO=1
Tue Jan 26 15:18:58 2021 110.17.5.83:20328 peer info: IV_COMP_STUB=1
Tue Jan 26 15:18:58 2021 110.17.5.83:20328 peer info: IV_COMP_STUBv2=1
Tue Jan 26 15:18:58 2021 110.17.5.83:20328 peer info: IV_TCPNL=1
Tue Jan 26 15:18:58 2021 110.17.5.83:20328 peer info: IV_GUI_VER=OpenVPN_GUI_11
Tue Jan 26 15:18:58 2021 110.17.5.83:20328 Control Channel: TLSv1.3, cipher TLSv1.3 TLS_AES_256_GCM_SHA384, 2048 bit RSA
Tue Jan 26 15:18:58 2021 110.17.5.83:20328 [zhangzhuo] Peer Connection Initiated with [AF_INET]110.17.5.83:20328
Tue Jan 26 15:18:58 2021 zhangzhuo/110.17.5.83:20328 MULTI_sva: pool returned IPv4=10.8.0.6, IPv6=(Not enabled)
Tue Jan 26 15:18:58 2021 zhangzhuo/110.17.5.83:20328 MULTI: Learn: 10.8.0.6 -> zhangzhuo 110.17.5.83:20328
Tue Jan 26 15:18:58 2021 zhangzhuo/110.17.5.83:20328 MULTI: primary virtual IP for zhangzho/110.17.5.83:20328: 10.8.0.6
Tue Jan 26 15:18:59 2021 zhangzhuo/110.17.5.83:20328 PUSH: Received control message: 'PUH_REQUEST'
Tue Jan 26 15:18:59 2021 zhangzhuo/110.17.5.83:20328 SENT CONTROL [zhangzhuo]: 'PUSH_EPLY,route 172.30.0.0 255.255.255.0,compress lz4-v2,route 10.8.0.1,topology net30,ping 10,ping-restart 120,ifconfig 10.8.0.6 10.8.0.5,peer-id 0,cipher AES-256-GCM' (status=1)
Tue Jan 26 15:18:59 2021 zhangzhuo/110.17.5.83:20328 Data Channel: using negotiated ciphe 'AES-256-GCM'
Tue Jan 26 15:18:59 2021 zhangzhuo/110.17.5.83:20328 Outgoing Data Channel: Cipher 'AES-56-GCM' initialized with 256 bit key
Tue Jan 26 15:18:59 2021 zhangzhuo/110.17.5.83:20328 Incoming Data Channel: Cipher 'AES-56-GCM' initialized with 256 bit key
```

2.8.4.3 验证OpenVPN服务器连接状态

```
[15:19:10 root@openvpn-server zhangzhuo]#ss -nt
State       Recv-Q      Send-Q            Local Address:Port             Peer Address:Port
ESTAB       0           36                 172.30.0.1:22              110.17.5.83:20645
ESTAB       0           0                 172.30.0.1:1194              110.17.5.83:20328
ESTAB       0           0                 172.30.0.1:47816           100.100.30.26:80
```

2.8.4.4 验证 Windows 客户端的 IP地址

```
C:\Users\zhangzhuo>ipconfig

Windows IP 配置


以太网适配器 net:

   媒体状态  . . . . . . . . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . . . :

未知适配器 本地连接:

   连接特定的 DNS 后缀 . . . . . . . :
   本地链接 IPv6 地址. . . . . . . . : fe80::50ca:1577:3352:1c78%7
   IPv4 地址 . . . . . . . . . . . . : 10.0.0.6
   子网掩码  . . . . . . . . . . . . : 255.255.255.252
   默认网关. . . . . . . . . . . . . :
```

2.8.4.5 验证Windows 客户端的路由表

```
IPv4 路由表
===========================================================================
活动路由:
网络目标        网络掩码          网关       接口   跃点数
          0.0.0.0          0.0.0.0      192.168.3.1     192.168.3.10     35
         10.0.0.0    255.255.255.0         在链路上          10.0.0.1    291
         10.0.0.1  255.255.255.255         在链路上          10.0.0.1    291
         10.0.0.1  255.255.255.255       10.0.0.5        10.0.0.6    281
         10.0.0.4  255.255.255.252         在链路上          10.0.0.6    281
         10.0.0.6  255.255.255.255         在链路上          10.0.0.6    281
         10.0.0.7  255.255.255.255         在链路上          10.0.0.6    281
       10.0.0.255  255.255.255.255         在链路上          10.0.0.1    291
        127.0.0.0        255.0.0.0         在链路上         127.0.0.1    331
        127.0.0.1  255.255.255.255         在链路上         127.0.0.1    331
  127.255.255.255  255.255.255.255         在链路上         127.0.0.1    331
      169.254.0.0      255.255.0.0         在链路上     169.254.247.90    291
   169.254.247.90  255.255.255.255         在链路上     169.254.247.90    291
  169.254.255.255  255.255.255.255         在链路上     169.254.247.90    291
       172.30.0.0    255.255.255.0       10.0.0.5        10.0.0.6    281
      192.168.3.0    255.255.255.0         在链路上       192.168.3.10    291
     192.168.3.10  255.255.255.255         在链路上       192.168.3.10    291
    192.168.3.255  255.255.255.255         在链路上       192.168.3.10    291
     192.168.10.0    255.255.255.0         在链路上       192.168.10.1    291
     192.168.10.1  255.255.255.255         在链路上       192.168.10.1    291
   192.168.10.255  255.255.255.255         在链路上       192.168.10.1    291
        224.0.0.0        240.0.0.0         在链路上         127.0.0.1    331
        224.0.0.0        240.0.0.0         在链路上          10.0.0.6    281
        224.0.0.0        240.0.0.0         在链路上       192.168.3.10    291
        224.0.0.0        240.0.0.0         在链路上     169.254.247.90    291
        224.0.0.0        240.0.0.0         在链路上          10.0.0.1    291
        224.0.0.0        240.0.0.0         在链路上       192.168.10.1    291
  255.255.255.255  255.255.255.255         在链路上         127.0.0.1    331
  255.255.255.255  255.255.255.255         在链路上          10.0.0.6    281
  255.255.255.255  255.255.255.255         在链路上       192.168.3.10    291
  255.255.255.255  255.255.255.255         在链路上     169.254.247.90    291
  255.255.255.255  255.255.255.255         在链路上          10.0.0.1    291
  255.255.255.255  255.255.255.255         在链路上       192.168.10.1    291
===========================================================================
```