第 五 次 实 验 报 告

学生姓名	陈曦	学号	2020302181081	指导老师	曹越
专业	网安	班级	3 班	实验时间	2023.5.1
成绩					

一、课程名称：网络安全实验

二、实验名称：VPN 实验

三、实验目的：

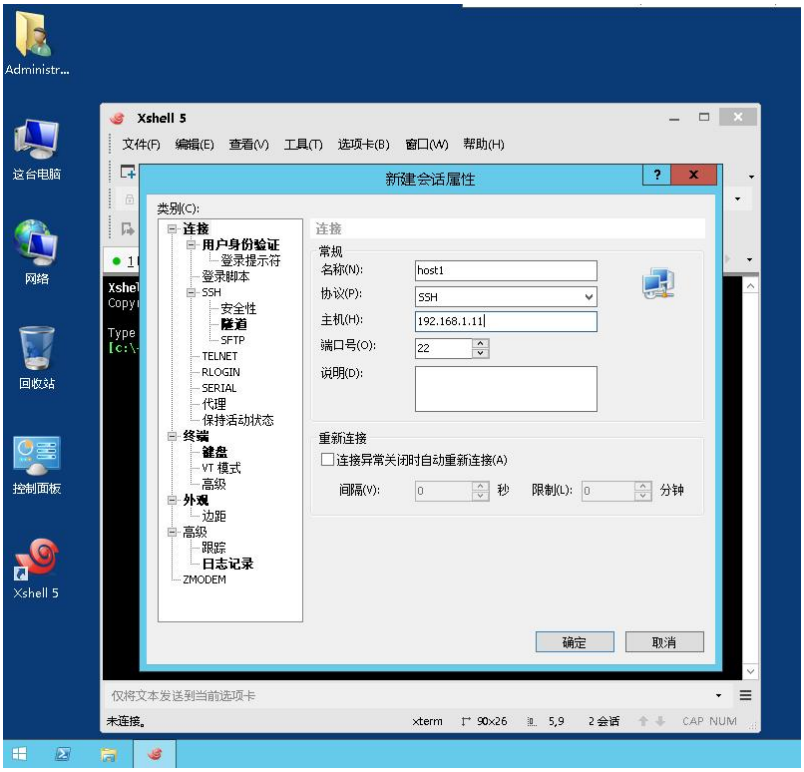
1. 掌握如何搭建基于隧道的虚拟专有网络
2. 掌握加密算法了解及其应用
3. 掌握如何安装部署配置 openvpn 服务端与客户端
4. 掌握 IPsecVPN 原理及安装部署
5. 了解公有云中 overlay 的实现

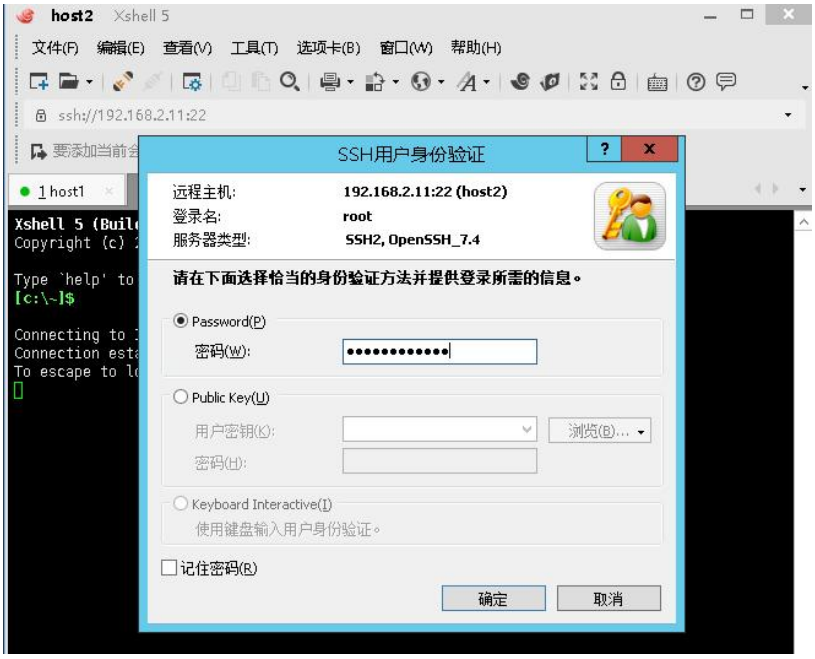
四、实验步骤

任务一：使用 IP 命令搭建基于隧道的虚拟专有网络

- 使用 xshell 登录远程主机并修改主机名

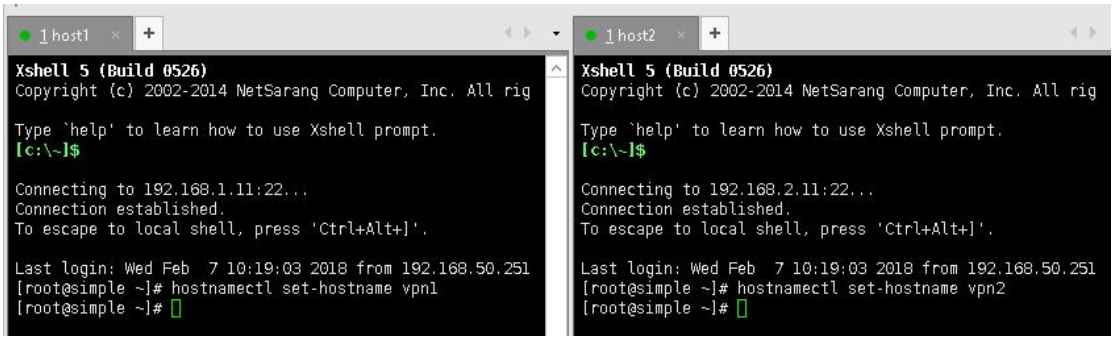
  启动本地机，通过 Xshell5 软件，在弹出的界面登陆主机 192.168.1.11 和 192.168.2.11 这两台主机.密码为 Simplexue123：



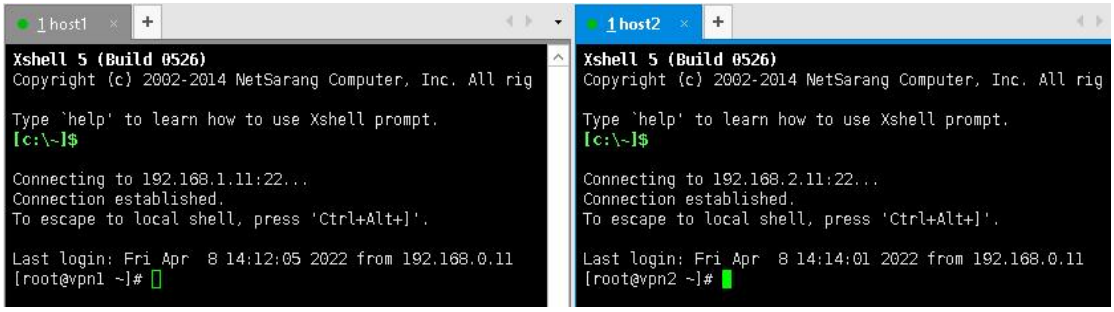


输入正确的账号和密码后便可通过 xshell 登录到目标机上

然后修改两台主机名分别为 vpn1 和 vpn2



退出后重新登录可以发现主机名已经更改



- 加载 gre 内核模块并检查

  分别使用如下命令加载内核模块并查询内核模块是否已经加载

  modprobe ip_gre

  lsmod | grep ip_gre

- 配置 tunnel，使它们互通

  vpn1 创建一个 GRE 类型隧道设备 gre1，并设置对端 IP 为 192.168.2.11。隧道数据包将被从 192.168.1.11 也就是本地 IP 地址发起，其 TTL 字段被设置为 255。隧道设备分配的 IP 地址为 10.10.10.1，掩码为 255.255.255.0

  - 创建 GRE 类型隧道设备 gre1，并验证是否添加成功

```
[root@vpn1 ~]# ip tunnel add gre1 mode gre remote 192.16
8.2.11 local 192.168.1.11 ttl 255
[root@vpn1 ~]# ip a | grep gre1
5: gre1@NONE: <POINTOPOINT,NOARP> mtu 1426 qdisc noop st
ate DOWN qlen 1
[root@vpn1 ~]#
```

- 启动 gre1 并分配 ip 地址 10.10.10.1，检测是否添加并启动

  使用如下命令启动 gre1 并分配 ip 地址 10.10.10.1，检测是否添加并启动

  ip link set gre1 up

  ip addr add 10.10.10.1/24 dev gre1

  ip a | grep gre1

```
[root@vpn1 ~]# ip link set gre1 up
[root@vpn1 ~]# ip addr add 10.10.10.1/24 dev gre1
[root@vpn1 ~]# ip a | grep gre1
5: gre1@NONE: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1426 q
disc noqueue state UNKNOWN qlen 1
    inet 10.10.10.1/24 scope global gre1
[root@vpn1 ~]#
```

- 查看隧道状态

  使用如下命令查看隧道状态

  ip -d link show

```
[root@vpn1 ~]# ip -d link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue st
ate UNKNOWN mode DEFAULT qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:0
0 promiscuity 0 addrgenmode eui64
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdis
c pfifo_fast state UP mode DEFAULT qlen 1000
    link/ether fa:16:3e:3c:06:ad brd ff:ff:ff:ff:ff:ff p
romiscuity 0 addrgenmode eui64
3: gre0@NONE: <NOARP> mtu 1476 qdisc noop state DOWN mod
e DEFAULT qlen 1
    link/gre 0.0.0.0 brd 0.0.0.0 promiscuity 0
    gre remote any local any ttl inherit nopmtudisc addr
genmode eui64
4: gretap0@NONE: <BROADCAST,MULTICAST> mtu 1462 qdisc no
op state DOWN mode DEFAULT qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff p
romiscuity 0
    gretap remote any local any ttl inherit nopmtudisc a
ddrgenmode eui64
5: gre1@NONE: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1426 q
disc noqueue state UNKNOWN mode DEFAULT qlen 1
```

- 同理对 vpn2 创建一个 GRE 类型隧道设备 gre1

  vpn2 创建一个 GRE 类型隧道设备 gre1，并设置对端 IP 为 192.168.1.11。隧道数据包将被从 192.168.2.11 也就是本地 IP 地址发起，其 TTL 字段被设置为 255。隧道设备分配的 IP 地址为 10.10.10.2，掩码为 255.255.255.0。

  ip tunnel add gre1 mode gre remote 192.168.1.11 local 192.168.2.11

ttl 255

ip link set gre1 up

ip addr add 10.10.10.2/24 dev gre1

ip a | grep gre1

```
[root@vpn2 ~]# ip tunnel add gre1 mode gre remote 192.168.1.11 local 192.168.2.11 ttl 255
[root@vpn2 ~]# ip link set gre1 up
[root@vpn2 ~]# ip addr add 10.0.0.2/24 dev gre1
[root@vpn2 ~]# ip a | grep gre1
5: gre1@NONE: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1426 qdisc noqueue state UNKNOWN qlen 1
    inet 10.0.0.2/24 scope global gre1
[root@vpn2 ~]#
```

- 测试隧道是否联通

  在 vpn1 主机上使用如下 ping 命令检测隧道是否连通

  ping 10.10.10.2

  发现可以 ping 通，隧道连通

```
[root@vpn1 ~]# ping 10.10.10.2
PING 10.10.10.2 (10.10.10.2) 56(84) bytes of data.
64 bytes from 10.10.10.2: icmp_seq=1 ttl=64 time=0.523 ms
64 bytes from 10.10.10.2: icmp_seq=2 ttl=64 time=0.647 ms
64 bytes from 10.10.10.2: icmp_seq=3 ttl=64 time=0.556 ms
64 bytes from 10.10.10.2: icmp_seq=4 ttl=64 time=0.631 ms
64 bytes from 10.10.10.2: icmp_seq=5 ttl=64 time=0.636 ms
64 bytes from 10.10.10.2: icmp_seq=6 ttl=64 time=0.542 ms
64 bytes from 10.10.10.2: icmp_seq=7 ttl=64 time=0.649 ms
64 bytes from 10.10.10.2: icmp_seq=8 ttl=64 time=0.611 ms
64 bytes from 10.10.10.2: icmp_seq=9 ttl=64 time=0.617 ms
64 bytes from 10.10.10.2: icmp_seq=10 ttl=64 time=0.636 ms
64 bytes from 10.10.10.2: icmp_seq=11 ttl=64 time=0.777 ms
64 bytes from 10.10.10.2: icmp_seq=12 ttl=64 time=0.569 ms
64 bytes from 10.10.10.2: icmp_seq=13 ttl=64 time=0.596 ms
64 bytes from 10.10.10.2: icmp_seq=14 ttl=64 time=0.724 ms
^C
--- 10.10.10.2 ping statistics ---
14 packets transmitted, 14 received, 0% packet loss, time 13008ms
rtt min/avg/max/mdev = 0.523/0.622/0.777/0.069 ms
```

- 最后卸载 gre 模块

  使用如下命令卸载 GRE 模块

  rmmod ip_gre

```
[root@vpn1 ~]# rmmod ip_gre
[root@vpn1 ~]# ls | grep gre
[root@vpn1 ~]#
```

```
[root@vpn2 ~]# rmmod ip_gre
[root@vpn2 ~]# ls | grep ger
[root@vpn2 ~]#
```

任务二 使用加密工具 OpenSSL 创建加密密钥

- 查看 openssl 命令的基本帮助

  使用如下命令查看 openssl 帮助

  openssl genrsa -

```
[root@vpn1 ~]# openssl genrsa -
usage: genrsa [args] [numbits]
 -des            encrypt the generated key with DES in cbc mode
 -des3           encrypt the generated key with DES in ede cbc mode (168 bit key)
 -idea           encrypt the generated key with IDEA in cbc mode
 -seed
                 encrypt PEM output with cbc seed
 -aes128, -aes192, -aes256
                 encrypt PEM output with cbc aes
 -camellia128, -camellia192, -camellia256
                 encrypt PEM output with cbc camellia
 -out file       output the key to 'file
 -passout arg    output file pass phrase source
 -f4             use F4 (0x10001) for the E value
 -3              use 3 for the E value
 -engine e       use engine e, possibly a hardware device.
 -rand file:file:...
                 load the file (or the files in the directory) into
                 the random number generator
[root@vpn1 ~]#
```

- 生成 RSA 密钥对
  - 生产 RSA 私钥

    使用如下命令生成长度为 2048 比特的私钥文件

    openssl genrsa -out rsa_private.key 2048

```
[root@vpn1 ~]# openssl genrsa -out rsa_private.key 2048
Generating RSA private key, 2048 bit long modulus
........+++
...........................+++
e is 65537 (0x10001)
[root@vpn1 ~]#
```

  - 生成私钥对应的公钥

    使用如下命令生成私钥对应的公钥

    openssl rsa -in rsa_private.key -pubout -out rsa_public.key

```
[root@vpn1 ~]# openssl rsa -in rsa_private.key -pubout -out rsa_public.key
writing RSA key
[root@vpn1 ~]# ls -al | grep key
-rw-r--r--   1 root root 1675 4月   8 14:44 rsa_private.key
-rw-r--r--   1 root root  451 4月   8 14:47 rsa_public.key
[root@vpn1 ~]#
```

- 生成 AES 加密的 RSA 密钥对

  生成 AES 加密的 RSA 私钥

  使用如下命令生成 AES 加密的 RSA 私钥，并设置密码为 simple

openssl genrsa -aes256 -passout pass:simple -out rsa_aes_private.key 2048

```
[root@vpn1 ~]# openssl genrsa -aes256 -passout pass:simple -out rsa_aes_private.key 2048
Generating RSA private key, 2048 bit long modulus
.......+++
...........................+++
e is 65537 (0x10001)
```

生成对应的公钥

```
[root@vpn1 ~]# openssl rsa -in rsa_aes_private.key -passin pass:simple -pubout -out rsa_aes_public.key
writing RSA key
[root@vpn1 ~]# ll | grep rsa
-rw-r--r-- 1 root root 1766 4月   8 14:51 rsa_aes_private.key
-rw-r--r-- 1 root root  451 4月   8 14:53 rsa_aes_public.key
-rw-r--r-- 1 root root 1675 4月   8 14:44 rsa_private.key
-rw-r--r-- 1 root root  451 4月   8 14:47 rsa_public.key
```

- 加密与非加密之间的转换

  可以使用如下命令进行加密和非加密之间的转换

  openssl rsa -in rsa_aes_private.key -passin pass:simple -out rsa_private.key

  openssl rsa -in rsa_private.key -aes256 -passout pass:simple -out rsa_aes_private.key

```
[root@vpn1 ~]# openssl rsa -in rsa_aes_private.key -passin pass:simple -out rsa_private.key
writing RSA key
[root@vpn1 ~]# openssl rsa -in rsa_private.key -aes256 -passout pass:simple -out rsa_aes_private.key
writing RSA key
[root@vpn1 ~]# ll | grep rsa
-rw-r--r-- 1 root root 1766 4月   8 15:04 rsa_aes_private.key
-rw-r--r-- 1 root root  451 4月   8 14:53 rsa_aes_public.key
-rw-r--r-- 1 root root 1675 4月   8 15:03 rsa_private.key
-rw-r--r-- 1 root root  451 4月   8 14:47 rsa_public.key
[root@vpn1 ~]#
```

- 生成自签名证书

  使用如下命令生成私钥和自签名证书

```
[root@vpn1 ~]# openssl req -newkey rsa:2048 -nodes -keyout rsa_private.key -x509 -days 365 -out cert.crt -subj "/C=CN/
ST=BJ/O=simpleedu/OU=edu/CN=simple/emailAddress=simple@simpleedu.com"
Generating a 2048 bit RSA private key
..+++
.........+++
writing new private key to 'rsa_private.key'
-----
[root@vpn1 ~]#
```

可以使用如下命令查看证书信息

openssl x509 -noout -text -in cert.crt

```
[root@vpn1 ~]# openssl x509 -noout -text -in cert.crt
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            fc:29:aa:d5:6b:d4:be:b6
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=CN, ST=BJ, O=simpleedu, OU=edu, CN=simple/emailAddress=simple@simpleedu.com
        Validity
            Not Before: Apr  8 07:15:40 2022 GMT
            Not After : Apr  8 07:15:40 2023 GMT
        Subject: C=CN, ST=BJ, O=simpleedu, OU=edu, CN=simple/emailAddress=simple@simpleedu.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:a8:9f:66:c1:f8:2e:70:90:5b:17:5e:e6:89:a4:
                    91:8b:2a:b5:3e:bb:0b:21:73:08:c9:8b:86:24:4d:
                    d8:69:d0:af:5e:1a:50:42:d7:be:c3:2d:a5:b8:50:
                    b8:bc:72:69:c6:9c:3b:8d:34:62:35:7d:80:c7:33:
```

- 生成签名请求及 CA 签名

  使用如下命令生成签名请求及 CA 签名

```
[root@vpn1 ~]# openssl genrsa -aes256 -passout pass:simpleedu -out server.key 2048 openssl req -new -key server.key -o
ut server.csr
Generating RSA private key, 2048 bit long modulus
.......................................................................................................................
.................................+++
.....................................+++
e is 65537 (0x10001)
[root@vpn1 ~]#
```

任务三 SSL VPN 之 OpenVPN 的安装配置

- 在 vpn1 机器安装 OpenVPN 并验证

  使用如下命令安装 OpenVpn

  yum clean all

  yum install openvpn -y

```
[root@vpn1 ~]# yum clean all
已加载插件：fastestmirror
正在清理软件源： simple
Cleaning up everything
Maybe you want: rm -rf /var/cache/yum, to also free up space taken by orphaned data from disabled or removed repos
[root@vpn1 ~]# yum install openvpn -y
已加载插件：fastestmirror
simple                                                                    | 2.9 kB  00:00:00
simple/primary_db                                                         |  14 kB  00:00:00
Determining fastest mirrors
正在解决依赖关系
--> 正在检查事务
---> 软件包 openvpn.x86_64.0.2.4.4-1.el7 将被 安装
--> 正在处理依赖关系 libpkcs11-helper.so.1()(64bit)，它被软件包 openvpn-2.4.4-1.el7.x86_64 需要
--> 正在处理依赖关系 liblz4.so.1()(64bit)，它被软件包 openvpn-2.4.4-1.el7.x86_64 需要
--> 正在检查事务
---> 软件包 lz4.x86_64.0.1.7.3-1.el7 将被 安装
---> 软件包 pkcs11-helper.x86_64.0.1.11-3.el7 将被 安装
--> 解决依赖关系完成
```

  使用如下命令验证安装是否成功

  rpm -qa | grep openvpn

```
[root@vpn1 ~]# rpm -qa | grep openvpn
openvpn-2.4.4-1.el7.x86_64
[root@vpn1 ~]#
```

- 修改 OpenVPN 配置文件
  - 拷贝模板文件到配置文件目录

    使用如下命令将配置模板拷贝到配置文件目录/etc/openvpn/下

```
[root@vpn1 ~]# cp /usr/share/doc/openvpn-2.4.4/sample/sample-config-files/server.conf /etc/openvpn
[root@vpn1 ~]# ls /etc/openvpn
client  server  server.conf
[root@vpn1 ~]#
```

  - 修改 OpenVPN 服务端的配置文件
  - 通过 vim 来修改配置文件

```
1 host1    2 host2
#################################################
# Sample OpenVPN 2.0 config file for            #
# multi-client server.                          #
#                                               #
# This file is for the server side              #
# of a many-clients <-> one-server              #
# OpenVPN configuration.                        #
#                                               #
# OpenVPN also supports                         #
# single-machine <-> single-machine             #
# configurations (See the Examples page         #
# on the web site for more info).               #
#                                               #
# This config should work on Windows            #
# or Linux/BSD systems.  Remember on            #
# Windows to quote pathnames and use            #
# double backslashes, e.g.:                     #
# "C:\\Program Files\\OpenVPN\\config\\foo.key" #
#                                               #
# Comments are preceded with '#' or ';'         #
#################################################

# Which local IP address should OpenVPN
# listen on? (optional)
;local a.b.c.d

"/etc/openvpn/server.conf" [noeol] 315L, 10784C
```

  - 指定使用 TCP 协议

```
# TCP or UDP server?
proto tcp
#proto udp
```

  - 配置 DNS

```
;push "redirect-gateway def1 bypass-dhcp"

# Certain Windows-specific network settings
# can be pushed to clients, such as DNS
# or WINS server addresses.  CAVEAT:
# http://openvpn.net/faq.html#dhcpcaveats
# The addresses below refer to the public
# DNS servers provided by opendns.com.
;push "dhcp-option DNS 208.67.222.222"
;push "dhcp-option DNS 208.67.220.220"
-- 插入 --
```

  - 设置启动用户

```
# You can uncomment this out on
# non-Windows systems.
;user nobody
;group nobody
```

- 注释 explicit-exit-notify 1

```
# Notify the client that when the server restarts so it
# can automatically reconnect.
#explicit-exit-notify 1
-- 插入 --
```

- 安装密钥生成软件

  使用如下命令安装密钥生成软件

  yum install easy-rsa -y

```
[root@vpn1 ~]# vim /etc/openvpn/server.conf
[root@vpn1 ~]# yum install easy-rsa -y
已加载插件: fastestmirror
Loading mirror speeds from cached hostfile
正在解决依赖关系
--> 正在检查事务
---> 软件包 easy-rsa.noarch.0.2.2.2-1.el5 将被 安装
--> 解决依赖关系完成

依赖关系解决
```

- 准备配置证书文件
  - 拷贝文件到/etc/openvpn

    使用如下命令拷贝文件到/etc/openvpn 目录下

    cp -r /usr/share/easy-rsa/ /etc/openvpn/

```
[root@vpn1 ~]# cp -r /usr/share/easy-rsa/ /etc/openvpn/
[root@vpn1 ~]# ls /etc/openvpn/
client  easy-rsa  server  server.conf
[root@vpn1 ~]#
```

  - 配置生成证书的环境变量，并使之生效

    使用命令 vim /etc/openvpn/easy-rsa/2.0/vars

    修改环境变量如下

```
# These are the default values for fields
# which will be placed in the certificate.
# Don't leave any of these fields blank.
export KEY_COUNTRY="CN"
export KEY_PROVINCE="BJ"
export KEY_CITY="BEIJING"
export KEY_ORG="SimpleEdu"
export KEY_EMAIL="simpleedu@simple.com"
export KEY_OU="MyOrganizationalUnit"
```

    使用如下命令激活环境变量

    cd /etc/openvpn/easy-rsa/2.0/

source vars

```
[root@vpn1 ~]# cd /etc/openvpn/easy-rsa/2.0
[root@vpn1 2.0]# source vars
NOTE: If you run ./clean-all, I will be doing a rm -rf on /etc/openvpn/easy-rsa/2.0/keys
[root@vpn1 2.0]#
```

➢ 生成证书

使用如下命令生成证书

./clean-all

./build-ca

```
[root@vpn1 2.0]# ./clean-all
[root@vpn1 2.0]# ./build.ca
-bash: ./build.ca: 没有那个文件或目录
[root@vpn1 2.0]# ./build-ca
Generating a 2048 bit RSA private key
.............................+++
.......................................................+++
writing new private key to 'ca.key'
-----
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.
-----
```

- 建立服务端的证书

生成服务器端的证书

使用如下命令生成服务器端的证书并设置密码为 simple123

./build-key-server server

```
[root@vpn1 2.0]# ./build-key-server server
Generating a 2048 bit RSA private key
...............+++
............+++
writing new private key to 'server.key'
-----
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.
-----
Country Name (2 letter code) [CN]:
State or Province Name (full name) [BJ]:
Locality Name (eg, city) [BEIJING]:
Organization Name (eg, company) [SimpleEdu]:
Organizational Unit Name (eg, section) [MyOrganizationalUnit]:
Common Name (eg, your name or your server's hostname) [server]:
Name [EasyRSA]:
Email Address [simpleedu@simple.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:simple123
```

- 生成防攻击的 key 文件

使用如下命令生成防止攻击的 key 文件

openvpn --genkey --secret keys/ta.key

```
[root@vpn1 2.0]# openvpn --genkey --secret keys/ta.key
[root@vpn1 2.0]# ls keys/
01.pem  ca.key     index.txt.attr  serial      server.crt  server.key
ca.crt  index.txt  index.txt.old   serial.old  server.csr  ta.key
[root@vpn1 2.0]#
```

- 建立客户端证书

  创建密钥文件

  ./build-dh

```
[root@vpn1 2.0]# ./build-dh
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
.....................................................+.............................+...................+.................
.....................................................+.......................+...........+...................+...........
.....................................................................................................................
...............................+.....................................................................................+...........
....................+....................................+.................................+.........................
.................................................+...................................................................
..............+..............+.......+.............................................................................
.....................................................................................................................
.........................................+...........................................................................
.....................................................................................................................
.....................................................................................................................
.........+...........................................................................................+.........................
```

  使用如下命令查看生成的密钥文件

  ll keys/dh2048.pem

```
[root@vpn1 2.0]# ll keys/dh2048.pem
-rw-r--r-- 1 root root 424 4月   8 15:59 keys/dh2048.pem
[root@vpn1 2.0]#
```

  拷贝密钥认证文件到配置文件目录下

  使用如下命令拷贝密钥认证文件到配置目录下

  cd /etc/openvpn/easy-rsa/2.0/keys/

  cp dh2048.pem ca.crt server.crt server.key ta.key /etc/openvpn

```
[root@vpn1 2.0]# cd /etc/openvpn/easy-rsa/2.0/keys
[root@vpn1 keys]# cp 2048.pem ca.crt server.crt server ta.key /etc/openvpn
cp: 无法获取"2048.pem" 的文件状态(stat): 没有那个文件或目录
cp: 无法获取"server" 的文件状态(stat): 没有那个文件或目录
[root@vpn1 keys]# cp dh2048.pem ca.crt server.crt server ta.key /etc/openvpn
cp: 是否覆盖"/etc/openvpn/ca.crt"?  u^Hy
cp: 是否覆盖"/etc/openvpn/server.crt"?  y
cp: 无法获取"server" 的文件状态(stat): 没有那个文件或目录
cp: 是否覆盖"/etc/openvpn/ta.key"?  y
[root@vpn1 keys]# ls /etc/openvpn
ca.crt  client  dh2048.pem  easy-rsa  server  server.conf  server.crt  ta.key
[root@vpn1 keys]#
```

- 创建一个通用名为 client 的客户端证书

  使用如下命令进行创建

  cd ..

./build-key client

```
[root@vpn1 keys]# cd ..
[root@vpn1 2.0]# ./build-key client
Generating a 2048 bit RSA private key
.....................................+++
...............................+++
writing new private key to 'client.key'
-----
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.
-----
Country Name (2 letter code) [CN]:
State or Province Name (full name) [BJ]:
Locality Name (eg, city) [BEIJING]:
Organization Name (eg, company) [SimpleEdu]:
```

```
[root@vpn1 2.0]# ll keys/client*
-rw-r--r-- 1 root root    0 4月   8 16:09 keys/client.crt
-rw-r--r-- 1 root root 1090 4月   8 16:09 keys/client.csr
-rw------- 1 root root 1704 4月   8 16:09 keys/client.key
[root@vpn1 2.0]#
```

- 启动并检查

  启动 OpenVPN 服务

  使用如下命令启动 OpenVPN 服务，并设置该服务开机自启动

  systemctl start openvpn@server.service

  systemctl enable openvpn@server.service

```
[root@vpn1 2.0]# systemctl start openvpn@server.service
[root@vpn1 2.0]# systemctl enable openvpn@server.service
[root@vpn1 2.0]# systemctl status openvpn@server.service
● openvpn@server.service - OpenVPN Robust And Highly Flexible Tunneling Application On server
   Loaded: loaded (/usr/lib/systemd/system/openvpn@.service; enabled; vendor preset: disabled)
   Active: active (running) since 五 2022-04-08 16:33:10 CST; 17s ago
 Main PID: 7573 (openvpn)
   Status: "Initialization Sequence Completed"
   CGroup: /system.slice/system-openvpn.slice/openvpn@server.service
           └─7573 /usr/sbin/openvpn --cd /etc/openvpn/ --config server.conf

4月 08 16:33:10 vpn1 openvpn[7573]: Fri Apr  8 16:33:10 2022 Could not determine IPv4/IPv6 prot
4月 08 16:33:10 vpn1 openvpn[7573]: Fri Apr  8 16:33:10 2022 Socket Buffers: R=[87380->87380] S
4月 08 16:33:10 vpn1 openvpn[7573]: Fri Apr  8 16:33:10 2022 Listening for incoming TCP connect
4月 08 16:33:10 vpn1 openvpn[7573]: Fri Apr  8 16:33:10 2022 TCPv4_SERVER link local (bound):
4月 08 16:33:10 vpn1 openvpn[7573]: Fri Apr  8 16:33:10 2022 TCPv4_SERVER link remote: [AF_UNSP
4月 08 16:33:10 vpn1 openvpn[7573]: Fri Apr  8 16:33:10 2022 MULTI: multi_init called, r=256 v=
4月 08 16:33:10 vpn1 openvpn[7573]: Fri Apr  8 16:33:10 2022 IFCONFIG POOL: base=10.8.0.4 size=
```

  检查是否正常启动

  使用如下命令检查是否正常启动

  netstat -lntup | grep openvpn

```
[root@vpn1 2.0]# netstat -lntup | grep openvpn
tcp        0      0 0.0.0.0:1194            0.0.0.0:*               LISTEN      7573/openvpn
[root@vpn1 2.0]#
```

- 客户端（vpn2）登录测试
  - 在客户端安装 OpenVPN

使用如下命令在 vpn2 安装 OpenVPN

yum install openvpn -y

```
[root@vpn2 ~]# yum install openvpn -y
已加载插件: fastestmirror
simple
simple/primary_db
Determining fastest mirrors
正在解决依赖关系
--> 正在检查事务
---> 软件包 openvpn.x86_64.0.2.4.4-1.el7 将被 安装
--> 正在处理依赖关系 libpkcs11-helper.so.1()(64bit), 它被软件包 openvpn-2.4.4-1.el7.x86_64 需要
--> 正在处理依赖关系 liblz4.so.1()(64bit), 它被软件包 openvpn-2.4.4-1.el7.x86_64 需要
--> 正在检查事务
---> 软件包 lz4.x86_64.0.1.7.3-1.el7 将被 安装
---> 软件包 pkcs11-helper.x86_64.0.1.11-3.el7 将被 安装
--> 解决依赖关系完成
```

- 在 vpn1 端把生产文件拷贝到客户端

  使用如下命令在 vpn1 端将生产文件通过 scp 服务拷贝到客户端

  cd /etc/openvpn/easy-rsa/2.0/keys/

```
[root@vpn1 2.0]# cd /etc/openvpn/easy-rsa/2.0/keys
[root@vpn1 keys]# scp ca.crt client.crt client.key ta.key 192.168.2.11:/etc/openvpn/client
The authenticity of host '192.168.2.11 (192.168.2.11)' can't be established.
ECDSA key fingerprint is SHA256:bseXee0cWWx0qD+4lRA/flPmfpKSd1FXok0pIsF52nU.
ECDSA key fingerprint is MD5:a3:d9:34:fd:1d:b6:38:65:21:8d:ba:1f:94:c3:d2:ad.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.2.11' (ECDSA) to the list of known hosts.
root@192.168.2.11's password:
Permission denied, please try again.
root@192.168.2.11's password:
ca.crt                                                    100% 1781     2.8MB/s   00:00
client.crt                                                100% 5452     7.3MB/s   00:00
client.key                                                100% 1708     3.1MB/s   00:00
ta.key                                                    100%  636     1.4MB/s   00:00
[root@vpn1 keys]#
```

- 编辑客户端配置文件

  编辑客户端配置文件/etc/openvpn/client/client.conf

```
[root@vpn2 ~]# vim /etc/openvpn/client/client.conf
[root@vpn2 ~]# cat /etc/openvpn/client/client.conf
client
dev tun
proto tcp
remote 192.168.1.11 1194
resolv-retry infinite
nobind
persist-key
persist-tun
ca /etc/openvpn/client/ca.crt
cert /etc/openvpn/client/client.crt
key /etc/openvpn/client/client.key
tls-auth /etc/openvpn/client/ta.key 1
cipher AES-256-CBC
verb 3
mute 20
[root@vpn2 ~]#
```

- 启动 openvpn 客户端并挂后台运行

  使用如下命令启动 OpenVPN 客户端并挂后台运行，且实时查看日志 cd /etc/openvpn/client/

openvpn /etc/openvpn/client/client.conf &

```
[root@vpn2 ~]# cd /etc/openvpn/client
[root@vpn2 client]# openvpn /etc/openvpn/client/client.conf &
[1] 7267
[root@vpn2 client]# Fri Apr  8 16:47:17 2022 OpenVPN 2.4.4 x86_64-redhat-linux-gnu [Fedora EPEL patched] [SSL (OpenSSL)]
O] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] built on Sep 26 2017
Fri Apr  8 16:47:17 2022 library versions: OpenSSL 1.0.2k-fips  26 Jan 2017, LZO 2.06
Fri Apr  8 16:47:17 2022 WARNING: No server certificate verification method has been enabled.  See http://openvpn.net/how
html#mitm for more info.
Fri Apr  8 16:47:17 2022 Outgoing Control Channel Authentication: Using 160 bit message hash 'SHA1' for HMAC authenticati
Fri Apr  8 16:47:17 2022 Incoming Control Channel Authentication: Using 160 bit message hash 'SHA1' for HMAC authenticati
Fri Apr  8 16:47:17 2022 TCP/UDP: Preserving recently used remote address: [AF_INET]192.168.1.11:1194
Fri Apr  8 16:47:17 2022 Socket Buffers: R=[87380->87380] S=[16384->16384]
Fri Apr  8 16:47:17 2022 Attempting to establish TCP connection with [AF_INET]192.168.1.11:1194 [nonblock]
Fri Apr  8 16:47:18 2022 TCP connection established with [AF_INET]192.168.1.11:1194
Fri Apr  8 16:47:18 2022 TCP_CLIENT link local: (not bound)
Fri Apr  8 16:47:18 2022 TCP_CLIENT link remote: [AF_INET]192.168.1.11:1194
Fri Apr  8 16:47:18 2022 TLS: Initial packet from [AF_INET]192.168.1.11:1194, sid=517bc36c 5a9e4197
Fri Apr  8 16:47:18 2022 VERIFY OK: depth=1, C=CN, ST=BJ, L=BEIJING, O=SimpleEdu, OU=MyOrganizationalUnit, CN=SimpleEdu C
name=EasyRSA, emailAddress=simpleedu@simple.com
Fri Apr  8 16:47:18 2022 VERIFY OK: depth=0, C=CN, ST=BJ, L=BEIJING, O=SimpleEdu, OU=MyOrganizationalUnit, CN=server, nam
asyRSA, emailAddress=simpleedu@simple.com
```

- 查看网卡信息，得知已获取到 ip

  使用如下命令查看网卡信息

  ip addr show tun0

```
ip addr show tun0
3: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN qlen 100
    link/none
    inet 10.8.0.6 peer 10.8.0.5/32 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::643d:73b6:4532:df59/64 scope link flags 800
       valid_lft forever preferred_lft forever
```

- 测试是否可以 ping 通

  使用如下命令进行 ping 测试

  ping 10.8.0.1

```
[root@vpn2 client]# ping 10.8.0.1
PING 10.8.0.1 (10.8.0.1) 56(84) bytes of data.
64 bytes from 10.8.0.1: icmp_seq=1 ttl=64 time=0.780 ms
64 bytes from 10.8.0.1: icmp_seq=2 ttl=64 time=0.830 ms
64 bytes from 10.8.0.1: icmp_seq=3 ttl=64 time=0.821 ms
64 bytes from 10.8.0.1: icmp_seq=4 ttl=64 time=0.778 ms
64 bytes from 10.8.0.1: icmp_seq=5 ttl=64 time=0.746 ms
64 bytes from 10.8.0.1: icmp_seq=6 ttl=64 time=0.837 ms
64 bytes from 10.8.0.1: icmp_seq=7 ttl=64 time=0.818 ms
64 bytes from 10.8.0.1: icmp_seq=8 ttl=64 time=0.945 ms
64 bytes from 10.8.0.1: icmp_seq=9 ttl=64 time=1.31 ms
64 bytes from 10.8.0.1: icmp_seq=10 ttl=64 time=0.772 ms
64 bytes from 10.8.0.1: icmp_seq=11 ttl=64 time=1.00 ms
```

- OpenVPN NAT 配置

  使用如下命令配置 vpn1 上 OpenVPN 的 NAT

  iptables -t nat -A POSTROUTING -s 10.8.0.1/24 -j MASQUERADE

```
[root@vpn1 keys]# iptables -t nat -A POSTROUTING -s 10.8.0.1/24 -j MASQUERADE
[root@vpn1 keys]# iptables -t nat -nvL
Chain PREROUTING (policy ACCEPT 1 packets, 94 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain INPUT (policy ACCEPT 1 packets, 94 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 1 packets, 156 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 1 packets, 156 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 MASQUERADE  all  --  *      *       10.8.0.0/24          0.0.0.0/0
[root@vpn1 keys]#
```

在 vpn2 上使用如下命令验证 vpn1 上的验证策略

```
[root@vpn2 client]# ping -c 1 www.baidu.com
ping: www.baidu.com: 域名解析暂时失败
[root@vpn2 client]#
```

- 两台主机上均关闭 OpenVPN 服务

  使用如下命令关闭服务

  pkill openvpn

```
[root@vpn1 keys]# pkill openvpn
[root@vpn1 keys]#
```

```
[root@vpn2 client]# pkill openvpn
Fri Apr  8 16:55:15 2022 event_wait : Interrupted system call (code=4)
Fri Apr  8 16:55:15 2022 /sbin/ip route del 10.8.0.1/32
[root@vpn2 client]# Fri Apr  8 16:55:15 2022 Closing TUN/TAP interface
Fri Apr  8 16:55:15 2022 /sbin/ip addr del dev tun0 local 10.8.0.6 peer 10.8.0.5
Fri Apr  8 16:55:15 2022 SIGTERM[hard,] received, process exiting
```

任务四 IPsecVPN 原理及安装配置

- 调整内核参数

  设置两台主机/etc/sysctl.conf 文件的内容

  vim /etc/sysctl.conf

```
1 host1    2 host2
# sysctl settings are defined through files in
# /usr/lib/sysctl.d/, /run/sysctl.d/, and /etc/sysctl.d/.
#
# Vendors settings live in /usr/lib/sysctl.d/.
# To override a whole file, create a new file with the same in
# /etc/sysctl.d/ and put new settings there. To override
# only specific settings, add a file with a lexically later
# name in /etc/sysctl.d/ and put new settings there.
#
# For more information, see sysctl.conf(5) and sysctl.d(5).
#

net.ipv4.ip_forward = 1
net.ipv4.conf.default.rp_filter = 0
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.default.send_redirects = 0
net.ipv4.conf.eth0.accept_redirects = 0
net.ipv4.conf.eth0.send_redirects = 0
net.ipv4.conf.eth1.accept_redirects = 0
net.ipv4.conf.eth1.send_redirects = 0
net.ipv4.conf.lo.accept_redirects = 0
net.ipv4.conf.lo.send_redirects = 0
```

```
1 host1    2 host2
# sysctl settings are defined through files in
# /usr/lib/sysctl.d/, /run/sysctl.d/, and /etc/sysctl.d/.
#
# Vendors settings live in /usr/lib/sysctl.d/.
# To override a whole file, create a new file with the same in
# /etc/sysctl.d/ and put new settings there. To override
# only specific settings, add a file with a lexically later
# name in /etc/sysctl.d/ and put new settings there.
#
# For more information, see sysctl.conf(5) and sysctl.d(5).

net.ipv4.ip_forward = 1
net.ipv4.conf.default.rp_filter = 0
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.default.send_redirects = 0
net.ipv4.conf.eth0.accept_redirects = 0
net.ipv4.conf.eth0.send_redirects = 0
net.ipv4.conf.eth1.accept_redirects = 0
net.ipv4.conf.eth1.send_redirects = 0
net.ipv4.conf.lo.accept_redirects = 0
net.ipv4.conf.lo.send_redirects = 0
~
~
~
"/etc/sysctl.conf" 23L, 900C
```

使配置生效

输入以下命令使配置生效

sysctl -p

```
[root@vpn1 keys]# pkill openvpn
[root@vpn1 keys]# vim /etc/sysctl.conf
[root@vpn1 keys]# vim /etc/sysctl.conf
[root@vpn1 keys]# sysctl -p
net.ipv4.ip_forward = 1
net.ipv4.conf.default.rp_filter = 0
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.default.send_redirects = 0
net.ipv4.conf.eth0.accept_redirects = 0
net.ipv4.conf.eth0.send_redirects = 0
sysctl: cannot stat /proc/sys/net/ipv4/conf/eth1/accept_red
irects: 没有那个文件或目录
sysctl: cannot stat /proc/sys/net/ipv4/conf/eth1/send_redir
ects: 没有那个文件或目录
net.ipv4.conf.lo.accept_redirects = 0
net.ipv4.conf.lo.send_redirects = 0
[root@vpn1 keys]#
```

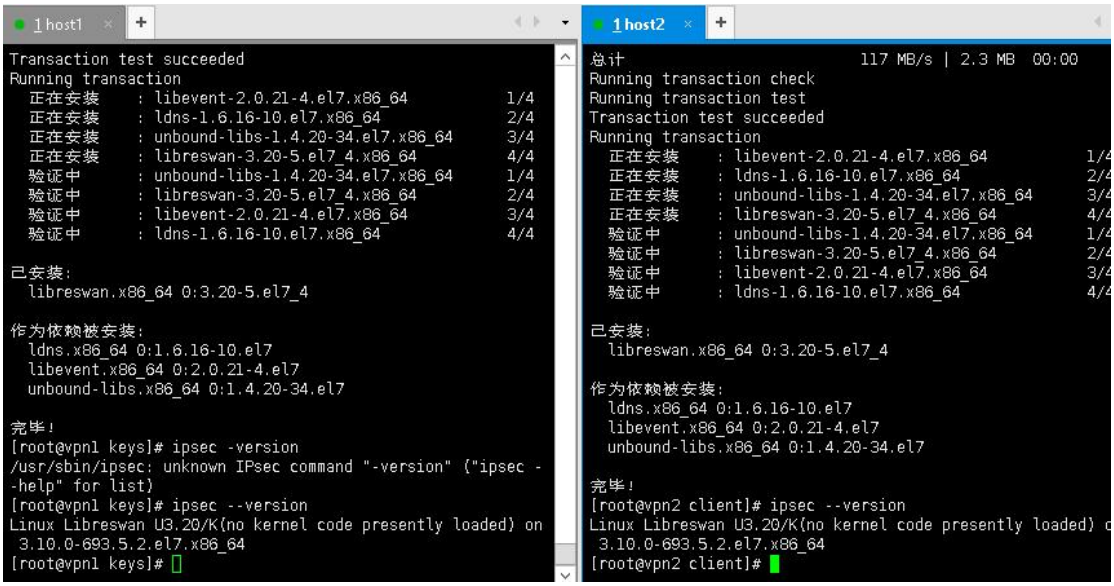
```
[root@vpn2 client]# vim /etc/systcl.conf
[root@vpn2 client]# vim /etc/sysctl.conf
[root@vpn2 client]# vim /etc/sysctl.conf
[root@vpn2 client]# sysctl -p
net.ipv4.ip_forward = 1
net.ipv4.conf.default.rp_filter = 0
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.default.send_redirects = 0
net.ipv4.conf.eth0.accept_redirects = 0
net.ipv4.conf.eth0.send_redirects = 0
sysctl: cannot stat /proc/sys/net/ipv4/conf/eth1/accept_red
irects: 没有那个文件或目录
sysctl: cannot stat /proc/sys/net/ipv4/conf/eth1/send_redir
ects: 没有那个文件或目录
net.ipv4.conf.lo.accept_redirects = 0
net.ipv4.conf.lo.send_redirects = 0
[root@vpn2 client]#
```

- 安装 openswan、libreswan 并验证安装

  使用如下命令进行安装和验证

  yum install openswan libreswan -y

  ipsec －version

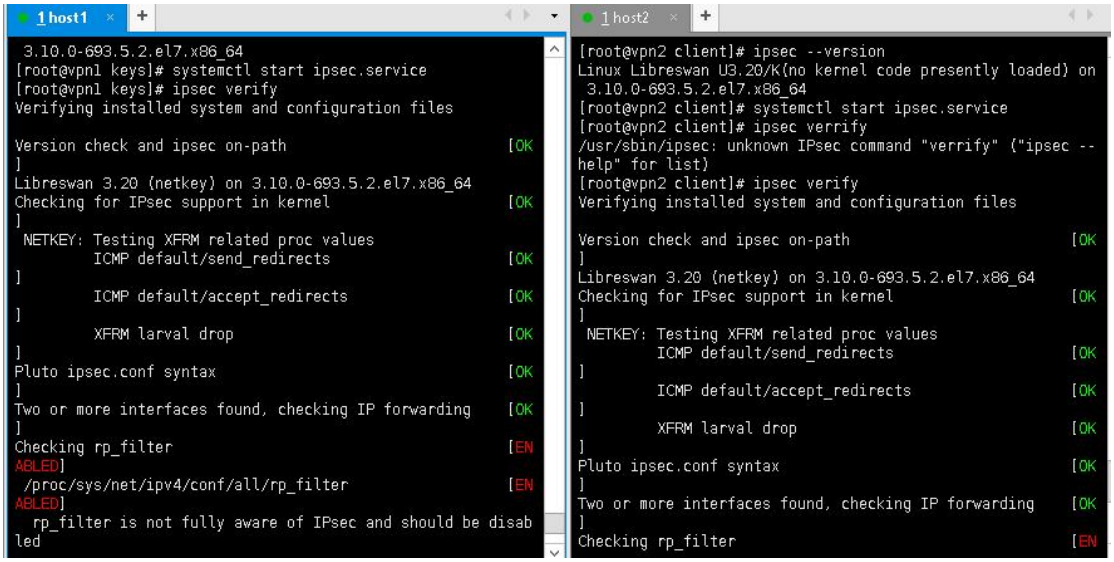


- 启动服务看是否正常

  使用如下命令启动对应的服务，并检查是否正常

  systemctl start ipsec.service

  ipsec verify



- 查看端口是否开启

使用如下命令进行检查

netstat -lntup | grep pluto

```
[root@vpn1 keys]# netstat -lntup | grep pluto
udp        0      0 127.0.0.1:4500          0.0.0.0:*
                     8213/pluto
udp        0      0 192.168.1.11:4500       0.0.0.0:*
                     8213/pluto
udp        0      0 127.0.0.1:500           0.0.0.0:*
                     8213/pluto
udp        0      0 192.168.1.11:500        0.0.0.0:*
                     8213/pluto
udp6       0      0 ::1:500                 :::*
                     8213/pluto
[root@vpn1 keys]#
```

```
[root@vpn2 client]# netstat -lntup | grep pluto
udp        0      0 127.0.0.1:4500          0.0.0.0:*
                     7867/pluto
udp        0      0 192.168.2.11:4500       0.0.0.0:*
                     7867/pluto
udp        0      0 127.0.0.1:500           0.0.0.0:*
                     7867/pluto
udp        0      0 192.168.2.11:500        0.0.0.0:*
                     7867/pluto
udp6       0      0 ::1:500                 :::*
                     7867/pluto
[root@vpn2 client]#
```

- 基于 pre-shared keys 认证方式

  修改配置文件参数

  在/etc/ipsec.conf 配置文件末尾增加如下参数

```
# For example connections, see your distribution's document
ation directory,
# or https://libreswan.org/wiki/
#
# There is also a lot of information in the manual page, "m
an ipsec.conf"
#
# It is best to add your IPsec connections as separate file
s in /etc/ipsec.d/
include /etc/ipsec.d/*.conf

conn net-to-net
        ike=aes256-sha2_256;modp2048
        phase2alg=aes256-sha2_256;modp2048
        authby=secret
        type=tunnel
        left=192.168.1.11
        leftsubnet=10.0.0.0/24
        leftid=@vpn1
        leftnexthop=%defaultroute
        right=192.168.2.11
        rightsubnet=10.0.1.0/24
        rightid=@vpn2
        rightnexthop=%defaultroute
        auto=add
"/etc/ipsec.conf" 67L, 2343C              67,2-9        底端
```

```
# For example connections, see your distribution's document
ation directory,
# or https://libreswan.org/wiki/
#
# There is also a lot of information in the manual page, "m
an ipsec.conf"
#
# It is best to add your IPsec connections as separate file
s in /etc/ipsec.d/
include /etc/ipsec.d/*.conf

conn net-to-net
        ike=aes256-sha2_256;modp2048
        phase2alg=aes256-sha2_256;modp2048
        authby=secret
        type=tunnel
        left=192.168.1.11
        leftsubnet=10.0.0.0/24
        leftid=@vpn1
        leftnexthop=%defaultroute
        right=192.168.2.11
        rightsubnet=10.0.1.0/24
        rightid=@vpn2
        rightnexthop=%defaultroute
        auto=add
"/etc/ipsec.conf" 68L, 2435C              68,9          底端
```

- 修改两台主机的密码配置文件

  修改 vpn1 的密码配置文件如下

  修改 vpn2 的密码配置文件如下

```
include /etc/ipsec.d/*.secrets

192.168.1.11 %any 0.0.0.0 : PSK "123"
~
~
```

```
include /etc/ipsec.d/*.secrets

192.168.2.11 %any 0.0.0.0 : PSK "123"
~
~
```

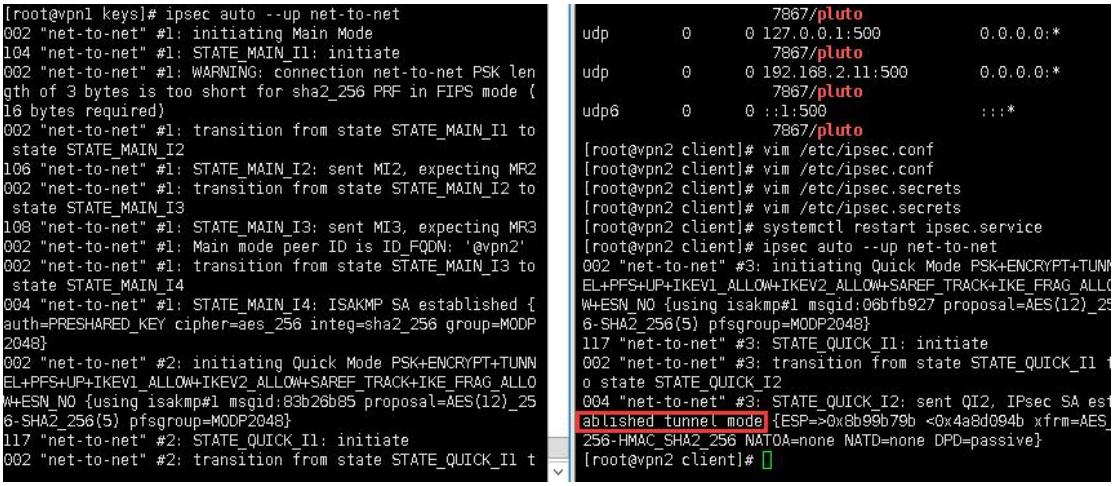
- 两端重新启动服务

  使用如下命令重启服务

systemctl restart ipsec.service

使用如下命令进行查看连接是否成功

ipsec auto --up net-to-net



- 测试是否可用

  首先使用如下命令在 VPN1 上搭建虚拟网络 10.0.0.1/24

  ip link add left1 type veth peer name left2

  ip netns add left

  ip link set left1 netns left

  ip link set left2 up

  ip addr add dev left2 10.0.0.1/24

  ip netns exec left ip link set lo up

  ip netns exec left ip link set left1 up

  ip netns exec left ip addr add dev left1 10.0.0.2/24

  ip netns exec left ip route add default via 10.0.0.1

使用如下命令查看虚拟网络

ip netns exec left ip

```
[root@vpn1 keys]# ip link set left1 netns left
Cannot find device "left1"
[root@vpn1 keys]# ip link add left1 type veth peer name left2
[root@vpn1 keys]# ip link set left1 netns left
[root@vpn1 keys]#
[root@vpn1 keys]# ip link set left2 up
[root@vpn1 keys]# ip addr add dev left2 10.0.0.1/24
[root@vpn1 keys]# ip netns exec left ip link set lo up
[root@vpn1 keys]# ip netns exec left ip link set left1 up
[root@vpn1 keys]# ip netns exec left ip addr add dev left1 10.0.0.2/24
[root@vpn1 keys]# ip netns exec left ip route add default via 10.0.0.1
[root@vpn1 keys]# ip netns exec left ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    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: ip_vti0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN qlen 1
    link/ipip 0.0.0.0 brd 0.0.0.0
6: left1@if5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP qlen 1000
    link/ether 42:21:1a:b1:fb:60 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.0.0.2/24 scope global left1
       valid_lft forever preferred_lft forever
    inet6 fe80::4021:1aff:feb1:fb60/64 scope link
       valid_lft forever preferred_lft forever
[root@vpn1 keys]#
```

类似的，在 vpn2 上使用如下命令搭建虚拟网络 10.0.1.1/24

```
ip link add left1 type veth peer name left2
ip netns add left
ip link set left1 netns left
ip link set left2 up
ip addr add dev left2 10.0.1.1/24
ip netns exec left ip link set lo up
ip netns exec left ip link set left1 up
ip netns exec left ip addr add dev left1 10.0.1.2/24
ip netns exec left ip route add default via 10.0.1.1
```

使用如下命令查看虚拟网络

```
ip netns exec left ip a
```

```
[root@vpn2 client]# ip link add left1 type veth peer name left2
[root@vpn2 client]# ip netns add left
[root@vpn2 client]# ip link set left1 netns left
[root@vpn2 client]# ip link set left2 up
[root@vpn2 client]# ip addr add dev left2 10.0.1.1/24
[root@vpn2 client]# ip netns exec left ip link set lo up
[root@vpn2 client]# ip netns exec left ip link set left1 up
[root@vpn2 client]# ip netns exec left ip addr add dev left1 10.0.1.2/24
[root@vpn2 client]# ip netns exec left ip addr route add default via 10.0.1.1
Command "route" is unknown, try "ip address help".
[root@vpn2 client]# ip netns exec left ip route add default via 10.0.1.1
[root@vpn2 client]# ip netns exec left ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    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: ip_vti0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN qlen 1
    link/ipip 0.0.0.0 brd 0.0.0.0
6: left1@if5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP qlen 1000
    link/ether 66:0c:96:b7:9c:78 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.0.1.2/24 scope global left1
       valid_lft forever preferred_lft forever
    inet6 fe80::640c:96ff:feb7:9c78/64 scope link
       valid_lft forever preferred_lft forever
[root@vpn2 client]#
```

最后在 vpn1 上进行 ping 测试

```
[root@vpn1 keys]# ip netns exec left ping 10.0.1.2
PING 10.0.1.2 (10.0.1.2) 56(84) bytes of data.
64 bytes from 10.0.1.2: icmp_seq=1 ttl=62 time=1.43 ms
64 bytes from 10.0.1.2: icmp_seq=2 ttl=62 time=0.772 ms
64 bytes from 10.0.1.2: icmp_seq=3 ttl=62 time=0.732 ms
64 bytes from 10.0.1.2: icmp_seq=4 ttl=62 time=0.790 ms
64 bytes from 10.0.1.2: icmp_seq=5 ttl=62 time=0.795 ms
64 bytes from 10.0.1.2: icmp_seq=6 ttl=62 time=0.692 ms
64 bytes from 10.0.1.2: icmp_seq=7 ttl=62 time=1.01 ms
^Z
[2]+  已停止               ip netns exec left ping 10.0.1.2
```

- 基于 RSA Signature 认证方式

  在 VPN1 和 VPN2 上分别生成一个新的 RSA 密钥对

  在 vpn1 和 vpn2 上分别执行下述命令

  rm -f /dev/random

  ln -s /dev/urandom /dev/random

  ipsec newhostkey --output /etc/ipsec.secrets

  ipsec showhostkey --left --ckaid {上一步生成的 CKAID}

```
[root@vpn1 keys]# rm -f /dev/random
[root@vpn1 keys]# ln -s /dev/urandom /dev/random
[root@vpn1 keys]# ipsec newhostkey --output /etc/ipsec.secrets
/usr/libexec/ipsec/newhostkey: WARNING: file "/etc/ipsec.secrets" exists, appending to it
Generated RSA key pair with CKAID 2ad5e0035b7121f309ba397cd7e78383f776c56d was stored in the NSS database
[root@vpn1 keys]# ipsec showhostkey --left --ckaid 2ad5e0035b7121f309ba397cd7e78383f776c56d
        # rsakey AwEAAci63
        leftrsasigkey=0sAwEAAci634kG079V4fBfiaYj5+JRHodiirJUKLqay6H/Y62JTe3UPfZobzuEwJa6SGLeOlEmTtl+O7MH1wzPI2AMo7P1JWQO6oBT4gryc1MmWcuRqYOctSpJTh+vH8roFGTI
YmE/28sBNKTJZv2g6iqwuSX7riaECcukxH+tPEIBl6ScKn5AGZKFHwHRITyNfLY2KeFsUkGwWUYF5TUOb/oUBw143PhAhUji/ieC5eRniKi30158jmly+8xZk42POcPeCLlGoMkXcgQdrOFgW9sM7Qqy50fr
0/rLxOLya1Wqs8X69NUBfeUCyQFxpDKTZw7J8kYeits/KstCsOGR7RnngaPjztm7V7BH7iQuA3gUJNfRcOER4ePKZJLviXXQVkQfnK9l6iPQ+m7LKqJicUjPSxlfDVrgVDJnIAkyJhJPD0JKrsifi/a3+YV7
pOWvZJ2rePWmbOWFQEfXm6WMFdAf+mtl3FUzLGw4rO8QcozBDy/P5NkppUhOSybrbqF5JyJuvaGbel57lwfnjUDeoyKuG/28mlzkVJwMfLrQlGRMi0P7oOLKAmC/X89gsiMcw/8d9CQurWUtlRvhUSmD6OHG
GDtCRa5f350ptfkZ3DSvNFV9pEU=
[root@vpn1 keys]#
```

```
[root@vpn2 client]# rm -f /dev/random
[root@vpn2 client]# ln -s /dev/urandom /dev/random
[root@vpn2 client]# ipsec newhostkey --output /etc/ipsec.secrets
/usr/libexec/ipsec/newhostkey: WARNING: file "/etc/ipsec.secrets" exists, appending to it
Generated RSA key pair with CKAID 805c55718cfc074d76dd42c41da2c9d51ec02df2 was stored in the NSS database
[root@vpn2 client]# ipsec newhostkey --left --ckaid 805c55718cfc074d76dd42c41da2c9d51ec02df2
ipsec newhostkey: unknown option `--left'
[root@vpn2 client]# ipsec newhostkey --left --ckaid 805c55718cfc074d76dd42c41da2c9d51ec02df2
ipsec newhostkey: unknown option `--left'
[root@vpn2 client]# ipsec showhostkey --left --ckaid 805c55718cfc074d76dd42c41da2c9d51ec02df2
        # rsakey AwEAAaDHn
        leftrsasigkey=0sAwEAAaDHnG2HZ/ADA6uacK75fbeEPPnzatEV7xgNwo7HhIzNsvr2zTkL8oh/lQnGZzmr07G9jVG6ZA4rFSyD6PjiAwhQTXHUgAoO4uHrZG
qz9nyniN+/Qwgy6KQjkMolp2QEyZ7XWpF7eAOgMtXcHtXYuLDw8tGiY7N8SeKjJFKCsDUPP65rqn4yYOo5/e3uNRYgIsEe/HFiMfQrCqRmFxUiMLSE4i0I6H+ABzGvaIQ5
9ac6UbpEWd7tlmGA3S8EvJAq0ygYYFmbqYnojHFRZTGMK+pEvdVrn4HtbhUbeeTdlNqoENlFc8nzyl6TLF4OsRUsUTIQuVALFCdHpy5GhYTlS6QjwtGQcBwIBNptzdWV4q
DyIaxyJMwWXVel+ygNGki5Zm9GAeHsIvvr9B8v48leUG4Oau8ElyQusJnC8SCnp0WyL4CAtJ6zxIyLOq4BPeLOslFccUQkMF59GTAnLk6iLTfo4REsZPtKBE/cQn35sMmP
pjCHj6c6yaVqB8urwtAUJ+lYIkOW5KlcLKpjXat0j4eHu7sSuplLF/GHTi0yy8UaVgcE0b5Pflfyw6xlh7tXCQI8APYBUltl+izaVWmSAla8gw==
[root@vpn2 client]#
```

- 修改两主机的配置文件

  修改两主机的配置文件/etc/ipsec.conf 如下

```
1 host1    2 host2
#
# It is best to add your IPsec connections as separate files in /etc/ipsec.d/
include /etc/ipsec.d/*.conf

conn net-to-net
        ike=aes256-sha2_256;modp2048
        phase2alg=aes256-sha2_256;modp2048
        authby=secret
        type=tunnel
        left=192.168.1.11
        leftsubnet=10.0.0.0/24
        leftid=@vpn1
        leftnexthop=0sAwEAAci634kG079V4fBfiaYj+JRHodiirJUKLqay6H/Y62JTe3UPfZobzuEwJa6SGLeOlEmTtl+O7MH1wzPI2AMo7P1JWQ06oBT4gryc1Mm
WcuRqY0ctSpJTh+vH8roYmE/28sBNKTJZv2g6iqwuSX7riaECcukxH+tPEIBl6ScKn+AGZKFHwHRITyNfLY2KeFsUkGwWUYF+TU0b/oUBw143PhAhUji/ieC+eRniKi301
+8jmly+8xZk42POcPeCL0/rLxOLya1Wqs8X69NUBfeUCyQFxpDKTZw7J8kYeits/KstCsOGR7RnngaPjztm7V7BH7iQuA3gUJNfRc0ER4ePKZJLviXXQVkQfnK9l6iPQ+m
7LKqJicUjPSx1fDVrgVDpCWvZJ2rePWmbOWFQEfXm6WMFdAf+mtl3FUzLGw4rO8QcozBDy/P+NkppUh0SybrbqF+JyJuvaGbel+7lwfnjUDeoyKuG/28mlzkVJwMfLrQlG
RMi0P7oOLKAmC/X89gsiGDtCRa+f3+OptfkZ3DSvNFV9pEU=
        right=192.168.2.11
        rightsubnet=10.0.1.0/24
        rightid=@vpn2
        rightnexthop=0sAwEAAaDHnG2HZ/ADA6uacK7+fbeEPPnzatEV7xgNwo7HhIzNsvr2zTkL8oh/lQnGZzmr07G9jVG6ZA4rFSyD6PjiAwhQTXHUgAoO4uHrZGq
z9nyniN+/Qwgy6KQjkMolp2QEyZ7XWpF7eAOgMtXcHtXYuLDw8tGiY7N8SeKjJFKCsDUPP6+rqn4yYOo+/e3uNRYgIsEe/HFiMfQrCqRmFxUiMLSE4i0I6H+ABzGvaIQ+9
ac6UbpEWd7tlmGA3S8EvJAq0ygYYFmbqYnojHFRZTGMK+pEvdVrn4HtbhUbeeTdlNqoENlFc8nzyl6TLF4OsRUsUTIQuVALFCdHpy+GhYTlS6QjwtGQcBwIBNptzdWV4qD
yIaxyJMwWXVel+ygNGki+Zm9GAeHsIvvr9B8v48leUG4Oau8ElyQusJnC8SCnp0WyL4CAtJ6zxIyLOq4BPeLOslFccUQkMF+9GTAnLk6iLTfo4REsZPtKBE/cQn3+sMmPp
jCHj6c6yaVqB8urwtAUJ+lYIkOW+KlcLKpjXat0j4eHu7sSuplLF/GHTi0yy8UaVgcE0b+Pflfyw6xlh7tXCQI8APYBUltl+izaVWmSAla8gw==
        auto=add
```

```
1 host1    2 host2
# It is best to add your IPsec connections as separate files in /etc/ipsec.d/
include /etc/ipsec.d/*.conf


conn net-to-net
        ike=aes256-sha2_256;modp2048
        phase2alg=aes256-sha2_256;modp2048
        authby=secret
        type=tunnel
        left=192.168.1.11
        leftsubnet=10.0.0.0/24
        leftid=@vpn1
        leftnexthop=0sAwEAAci634kG079V4fBfiaYj+JRHodiirJUKLqay6H/Y62JTe3UPfZobzuEwJa6SGLeOlEmTtl+O7MH1wzPI2AMo7P1JWQ06oBT4gryc1Mm
WcuRqY0ctSpJTh+vH8roYmE/28sBNKTJZv2g6iqwuSX7riaECcukxH+tPEIBl6ScKn+AGZKFHwHRITyNfLY2KeFsUkGwWUYF+TU0b/oUBw143PhAhUji/ieC+eRniKi301
+8jmly+8xZk42POcPeCL0/rLxOLya1Wqs8X69NUBfeUCyQFxpDKTZw7J8kYeits/KstCsOGR7RnngaPjztm7V7BH7iQuA3gUJNfRc0ER4ePKZJLviXXQVkQfnK9l6iPQ+m
7LKqJicUjPSx1fDVrgVDpCWvZJ2rePWmbOWFQEfXm6WMFdAf+mtl3FUzLGw4rO8QcozBDy/P+NkppUh0SybrbqF+JyJuvaGbel+7lwfnjUDeoyKuG/28mlzkVJwMfLrQlG
RMi0P7oOLKAmC/X89gsiGDtCRa+f3+OptfkZ3DSvNFV9pEU=
        right=192.168.2.11
        rightsubnet=10.0.1.0/24
        rightid=@vpn2
        rightnexthop=0sAwEAAaDHnG2HZ/ADA6uacK7+fbeEPPnzatEV7xgNwo7HhIzNsvr2zTkL8oh/lQnGZzmr07G9jVG6ZA4rFSyD6PjiAwhQTXHUgAoO4uHrZGq
z9nyniN+/Qwgy6KQjkMolp2QEyZ7XWpF7eAOgMtXcHtXYuLDw8tGiY7N8SeKjJFKCsDUPP6+rqn4yYOo+/e3uNRYgIsEe/HFiMfQrCqRmFxUiMLSE4i0I6H+ABzGvaIQ+9
ac6UbpEWd7tlmGA3S8EvJAq0ygYYFmbqYnojHFRZTGMK+pEvdVrn4HtbhUbeeTdlNqoENlFc8nzyl6TLF4OsRUsUTIQuVALFCdHpy+GhYTlS6QjwtGQcBwIBNptzdWV4qD
yIaxyJMwWXVel+ygNGki+Zm9GAeHsIvvr9B8v48leUG4Oau8ElyQusJnC8SCnp0WyL4CAtJ6zxIyLOq4BPeLOslFccUQkMF+9GTAnLk6iLTfo4REsZPtKBE/cQn3+sMmPp
jCHj6c6yaVqB8urwtAUJ+lYIkOW+KlcLKpjXat0j4eHu7sSuplLF/GHTi0yy8UaVgcE0b+Pflfyw6xlh7tXCQI8APYBUltl+izaVWmSAla8gw==
        auto=add
```

- 重新启动服务

  使用如下命令重新启动服务

systemctl restart ipsec.service

两台主机均执行如下命令

ipsec auto --up net-to-net

```
1 host1
[root@vpn1 keys]# vim /etc/ipsec.conf
[root@vpn1 keys]# vim /etc/ipsec.conf
[root@vpn1 keys]# vim /etc/ipsec.conf
[root@vpn1 keys]# systemctl restart ipsec.service
[root@vpn1 keys]# ipsec auto --up net-to-net
002 "net-to-net" #1: initiating Main Mode
104 "net-to-net" #1: STATE_MAIN_I1: initiate
003 "net-to-net" #1: ignoring informational payload NO_PROPOSAL
_CHOSEN, msgid=00000000, length=12
003 "net-to-net" #1: received and ignored informational message
010 "net-to-net" #1: STATE_MAIN_I1: retransmission; will wait 5
00ms for response
003 "net-to-net" #1: ignoring informational payload NO_PROPOSAL
_CHOSEN, msgid=00000000, length=12
003 "net-to-net" #1: received and ignored informational message
010 "net-to-net" #1: STATE_MAIN_I1: retransmission; will wait 1
000ms for response
003 "net-to-net" #1: ignoring informational payload NO_PROPOSAL
_CHOSEN, msgid=00000000, length=12
003 "net-to-net" #1: received and ignored informational message
010 "net-to-net" #1: STATE_MAIN_I1: retransmission; will wait 2
000ms for response
003 "net-to-net" #1: ignoring informational payload NO_PROPOSAL
_CHOSEN, msgid=00000000, length=12
003 "net-to-net" #1: received and ignored informational message
010 "net-to-net" #1: STATE_MAIN_I1: retransmission; will wait 4
000ms for response
```

```
1 host2
[root@vpn2 client]# vim /etc/ipsec.conf
[root@vpn2 client]# systemctl restart ipsec.service
[root@vpn2 client]# ipsec auto --up net-to-net
002 "net-to-net" #8: initiating Main Mode
104 "net-to-net" #8: STATE_MAIN_I1: initiate
003 "net-to-net" #8: ignoring informational payload NO_PROPOSA
L_CHOSEN, msgid=00000000, length=12
003 "net-to-net" #8: received and ignored informational messag
e
010 "net-to-net" #8: STATE_MAIN_I1: retransmission; will wait
500ms for response
003 "net-to-net" #8: ignoring informational payload NO_PROPOSA
L_CHOSEN, msgid=00000000, length=12
003 "net-to-net" #8: received and ignored informational messag
e
010 "net-to-net" #8: STATE_MAIN_I1: retransmission; will wait
1000ms for response
003 "net-to-net" #8: ignoring informational payload NO_PROPOSA
L_CHOSEN, msgid=00000000, length=12
003 "net-to-net" #8: received and ignored informational messag
e
010 "net-to-net" #8: STATE_MAIN_I1: retransmission; will wait
2000ms for response
003 "net-to-net" #8: ignoring informational payload NO_PROPOSA
L_CHOSEN, msgid=00000000, length=12
003 "net-to-net" #8: received and ignored informational messag
e
```

```
1 host1
003 "net-to-net" #1: received and ignored informational message
010 "net-to-net" #1: STATE_MAIN_I1: retransmission; will wait 2
000ms for response
003 "net-to-net" #1: ignoring informational payload NO_PROPOSAL
_CHOSEN, msgid=00000000, length=12
003 "net-to-net" #1: received and ignored informational message
010 "net-to-net" #1: STATE_MAIN_I1: retransmission; will wait 4
000ms for response
003 "net-to-net" #1: ignoring informational payload NO_PROPOSAL
_CHOSEN, msgid=00000000, length=12
003 "net-to-net" #1: received and ignored informational message
010 "net-to-net" #1: STATE_MAIN_I1: retransmission; will wait 8
000ms for response
003 "net-to-net" #1: ignoring informational payload NO_PROPOSAL
_CHOSEN, msgid=00000000, length=12
003 "net-to-net" #1: received and ignored informational message
010 "net-to-net" #1: STATE_MAIN_I1: retransmission; will wait 1
6000ms for response
003 "net-to-net" #1: ignoring informational payload NO_PROPOSAL
_CHOSEN, msgid=00000000, length=12
003 "net-to-net" #1: received and ignored informational message
010 "net-to-net" #1: STATE_MAIN_I1: retransmission; will wait 3
2000ms for response
003 "net-to-net" #1: ignoring informational payload NO_PROPOSAL
_CHOSEN, msgid=00000000, length=12
003 "net-to-net" #1: received and ignored informational message
```

```
1 host2
003 "net-to-net" #8: received and ignored informational messag
e
010 "net-to-net" #8: STATE_MAIN_I1: retransmission; will wait
2000ms for response
003 "net-to-net" #8: ignoring informational payload NO_PROPOSA
L_CHOSEN, msgid=00000000, length=12
003 "net-to-net" #8: received and ignored informational messag
e
010 "net-to-net" #8: STATE_MAIN_I1: retransmission; will wait
4000ms for response
003 "net-to-net" #8: ignoring informational payload NO_PROPOSA
L_CHOSEN, msgid=00000000, length=12
003 "net-to-net" #8: received and ignored informational messag
e
010 "net-to-net" #8: STATE_MAIN_I1: retransmission; will wait
8000ms for response
003 "net-to-net" #8: ignoring informational payload NO_PROPOSA
L_CHOSEN, msgid=00000000, length=12
003 "net-to-net" #8: received and ignored informational messag
e
010 "net-to-net" #8: STATE_MAIN_I1: retransmission; will wait
16000ms for response
003 "net-to-net" #8: ignoring informational payload NO_PROPOSA
L_CHOSEN, msgid=00000000, length=12
003 "net-to-net" #8: received and ignored informational messag
e
```

- ping 测试

在 vpn1 上使用如下命令进行 ping 测试

ip netns exec left ping 10.0.1.2

```
[root@vpn1 keys]# ip netns exec left ping 10.0.1.2
PING 10.0.1.2 (10.0.1.2) 56(84) bytes of data.
64 bytes from 10.0.1.2: icmp_seq=1 ttl=62 time=1.57 ms
64 bytes from 10.0.1.2: icmp_seq=2 ttl=62 time=1.06 ms
64 bytes from 10.0.1.2: icmp_seq=3 ttl=62 time=0.982 ms
64 bytes from 10.0.1.2: icmp_seq=4 ttl=62 time=0.834 ms
64 bytes from 10.0.1.2: icmp_seq=5 ttl=62 time=2.02 ms
64 bytes from 10.0.1.2: icmp_seq=6 ttl=62 time=0.888 ms
^C
--- 10.0.1.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5005ms
rtt min/avg/max/mdev = 0.834/1.227/2.023/0.430 ms
```

- 清除虚拟内网，停止服务

  在两台主机上均执行下列命令

  ip netns del left

  systemctl stop ipsec

```
[root@vpn1 keys]# ip netns del left
[root@vpn1 keys]# systemctl stop ipsec
[root@vpn1 keys]#
```

```
[root@vpn2 client]# ip netns del left
[root@vpn2 client]# systemctl stop ipsec
[root@vpn2 client]#
```

任务五 云计算中基于 Overlay 技术的隧道网络实现

- 在 VPN1 和 VPN2 分别安装 Open vSwitch 并启动服务

  安装 Open vSwitch

  使用如下命令安装 Open vSwitch

  yum install openvswitch -y

```
[root@vpn1 keys]# yum install openvswitch -y
已加载插件：fastestmirror
simple                                   | 2.9 kB     00:00
Loading mirror speeds from cached hostfile
正在解决依赖关系
--> 正在检查事务
---> 软件包 openvswitch.x86_64.0.2.5.0-2.el7 将被 安装
--> 解决依赖关系完成

依赖关系解决

=================================================================
 Package        架构       版本              源          大小
=================================================================
正在安装:
 openvswitch    x86_64     2.5.0-2.el7       simple      2.3 M

事务概要
=================================================================
安装  1 软件包
```

```
[root@vpn2 client]# systemctl stop ipsec
[root@vpn2 client]# yum install openvswitch -y
已加载插件：fastestmirror
Loading mirror speeds from cached hostfile
正在解决依赖关系
--> 正在检查事务
---> 软件包 openvswitch.x86_64.0.2.5.0-2.el7 将被 安装
--> 解决依赖关系完成

依赖关系解决

=================================================================
 Package        架构       版本              源          大小
=================================================================
正在安装:
 openvswitch    x86_64     2.5.0-2.el7       simple      2.3 M

事务概要
=================================================================
安装  1 软件包
```

- 启动服务并查看服务状态

  使用如下命令启动服务

  systemctl start openvswitch.service

  使用如下命令查看服务状态

  systemctl status openvswitch.service

```
[root@vpn1 keys]# systemctl start openvswitch.service
[root@vpn1 keys]# systemctl status openvswitch.service
● openvswitch.service - Open vSwitch
   Loaded: loaded (/usr/lib/systemd/system/openvswitch.service;
 disabled; vendor preset: disabled)
   Active: active (exited) since 五 2022-04-08 21:48:14 CST; 8s
 ago
  Process: 9318 ExecStart=/bin/true (code=exited, status=0/SUCC
ESS)
 Main PID: 9318 (code=exited, status=0/SUCCESS)

4月 08 21:48:14 vpn1 systemd[1]: Starting Open vSwitch...
4月 08 21:48:14 vpn1 systemd[1]: Started Open vSwitch.
```

```
[root@vpn2 client]# systemctl start openvswitch.service
[root@vpn2 client]# systemctl status openvswitch.service
● openvswitch.service - Open vSwitch
   Loaded: loaded (/usr/lib/systemd/system/openvswitch.service
; disabled; vendor preset: disabled)
   Active: active (exited) since 五 2022-04-08 21:48:32 CST; 7
s ago
  Process: 8971 ExecStart=/bin/true (code=exited, status=0/SUC
CESS)
 Main PID: 8971 (code=exited, status=0/SUCCESS)

4月 08 21:48:32 vpn2 systemd[1]: Starting Open vSwitch...
4月 08 21:48:32 vpn2 systemd[1]: Started Open vSwitch.
```

- 配置 VPN1

  在 VPN1 上添加名为 br0 的网桥

  使用如下命令在 VPN1 上添加名为 br0 的网桥

  ovs-vsctl add-br br0

  给 br0 网桥分配一个 ip

  使用如下命令给 br0 网桥分配一个 ip

  ifconfig br0 10.1.0.1/24 up

  查看网桥

  ifconfig br0

```
[root@vpn1 keys]# ovs-vsctl add-br br0
[root@vpn1 keys]# ifconfig br0 10.1.0.1/24 up
[root@vpn1 keys]# ifconfig br0
br0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.1.0.1  netmask 255.255.255.0  broadcast 10.1.0.
255
        inet6 fe80::dc87:7eff:fe0d:694a  prefixlen 64  scopeid
0x20<link>
        ether de:87:7e:0d:69:4a  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 648 (648.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collision
s 0
```

- 配置 VPN2

  在 VPN2 上添加名为 br0 的网桥

  同 vpn1，执行如下命令

  ovs-vsctl add-br br0

  给 br0 网桥分配一个 ip

  使用如下命令给 br0 网桥分配一个 ip

  ifconfig br0 10.1.0.1/24 up

  查看网桥

  ifconfig br0

```
[root@vpn2 client]# ovs-vsctl add-br br0
[root@vpn2 client]# ifconfig br0 10.1.0.2/24 up
[root@vpn2 client]# ifconfig br0
br0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.1.0.2  netmask 255.255.255.0  broadcast 10.1.0.255
        inet6 fe80::78a5:49ff:fea7:654f  prefixlen 64  scopeid 0x20<link>
        ether 7a:a5:49:a7:65:4f  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6  bytes 508 (508.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- 搭建 VXLAN 隧道

  在 VPN1 上设置 VXLAN

  执行下列命令,在 VPN1 上设置 VXLAN，远端 ip 设置为 VPN2 能对外通信的 br0 的 ip

```
[root@vpn1 keys]# ovs-vsctl add-port br0 vxl -- set interface vxl type=vxlan options:remote_ip=192.168.2.11
[root@vpn1 keys]# ovs-vsctl show
bef02934-ca27-46cd-8d09-1c6b0a888612
    Bridge "br0"
        Port vxl
            Interface vxl
                type: vxlan
                options: {remote_ip="192.168.2.11"}
        Port "br0"
            Interface "br0"
                type: internal
    ovs_version: "2.5.0"
[root@vpn1 keys]#
```

- 在 VPN2 上设置 VXLAN

  执行下列命令,在 VPN2 上设置 VXLAN，远端 ip 设置为 VPN1 能对外通信的 br0 的 ip

```
[root@vpn2 client]# ovs-vsctl add-port br0 vxl -- set interface vxl type=vxlan options:remote_ip=192.168.1.11
[root@vpn2 client]# ovs-vsctl show
d2013500-2804-44a7-8fe5-cb061471edd1
    Bridge "br0"
        Port "br0"
            Interface "br0"
                type: internal
        Port vxl
            Interface vxl
                type: vxlan
                options: {remote_ip="192.168.1.11"}
    ovs_version: "2.5.0"
[root@vpn2 client]#
```

- 验证 VXLAN 隧道

  在 vpn1 上进行 ping 测试

```
[root@vpn1 keys]# ping 10.1.0.2
PING 10.1.0.2 (10.1.0.2) 56(84) bytes of data.
64 bytes from 10.1.0.2: icmp_seq=1 ttl=64 time=2.38 ms
64 bytes from 10.1.0.2: icmp_seq=2 ttl=64 time=0.599 ms
64 bytes from 10.1.0.2: icmp_seq=3 ttl=64 time=0.644 ms
64 bytes from 10.1.0.2: icmp_seq=4 ttl=64 time=0.669 ms
^C
--- 10.1.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 0.599/1.073/2.382/0.756 ms
[root@vpn1 keys]#
```