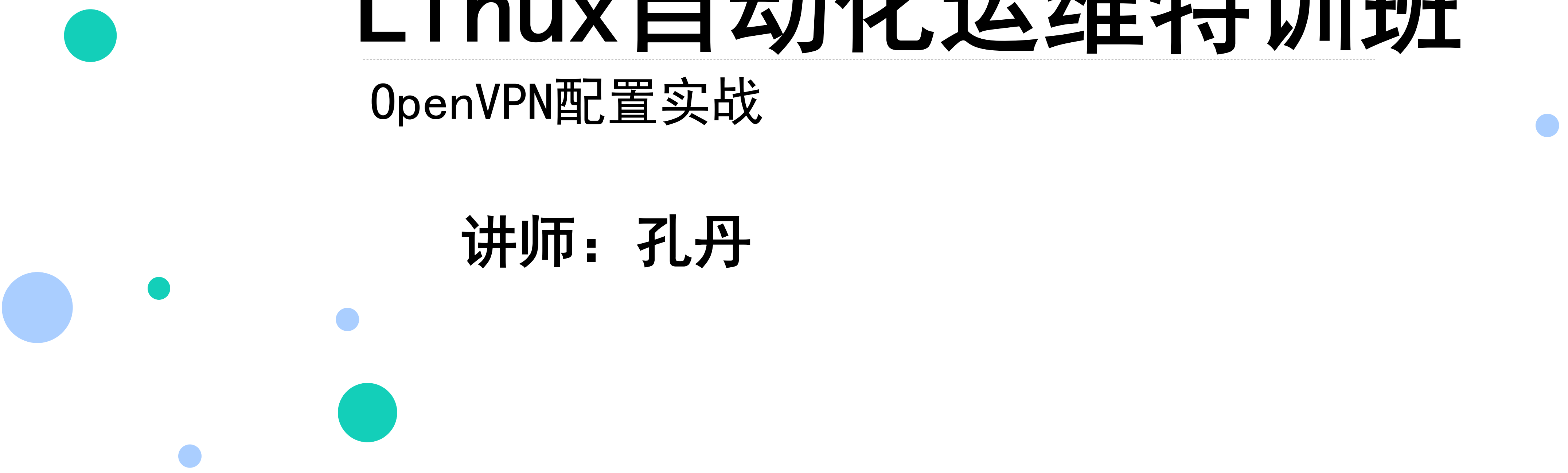


# Linux 自动化运维特训班

OpenVPN配置实战

讲师：孔丹



# 大纲

- OpenVPN简介
- OpenVpn通信原理
- OpenVpn应用场景
- OpenVpn实战案例

版权所有，侵权必究

# OpenVPN简介

- ❑ OpenVPN是一个用于创建虚拟专用网络加密通道的软件包，最早由James Yonan编写。OpenVPN允许创建的VPN使用公开密钥、数字证书、或者用户名 / 密码来进行身份验证。它大量使用了OpenSSL加密库中的SSLv3/TLSv1协议函数库。OpenVPN能在Solaris、Linux、OpenBSD、FreeBSD、NetBSD、Mac OS X与Windows 2000/XP/Vista/7以及Android和iOS上运行，并包含了许多安全性的功能。
- ❑ OpenVPN使用TLS加密是通过使用公开密钥（非对称密钥，加密解密使用不同的key，一个称为Public key，另一个数Private key）对数据进行加密。
- ❑ OpenVPN提供了多种身份验证方式，用以确认参与连接双方的身份，包括：预享私钥，第三方证书以及用户名/密码组合。预享密钥最为简单，但同时它只能用于建立点对点的VPN；基于PKI的第三方证书提供了最完善的功能，但是需要额外的精力去维护一个PKI证书体系。OpenVPN2.0后引入了用户名/口令组合的身份验证方式，它可以省略客户端证书，但是仍有一份服务器证书需要被用作加密。

# OpenVpn通信原理

- ❑ OpenVpn的技术核心是虚拟网卡，其次是SSL协议实现。
- ❑ 虚拟网卡是使用网络底层编程技术实现的一个驱动软件，安装后在主机上多出现一个网卡，可以像其它网卡一样进行配置。
- ❑ 在OpenVpn中，如果用户访问一个远程的虚拟地址（属于虚拟网卡配用的地址系列，区别于真实地址），则操作系统会通过路由机制将数据包（TUN模式）或数据帧（TAP模式）发送到虚拟网卡上，服务程序接收该数据并进行相应的处理后，通过SOCKET从外网上发送出去，远程服务程序通过SOCKET从外网上接收数据，并进行相应的处理后，发送给虚拟网卡，则应用软件可以接收到，完成了一个单向传输的过程，反之亦然。

# OpenVpn通信原理

## □ 发送数据

- 1、 应用程序发送网络数据。
- 2、 网络数据根据修改后的路由表把数据路由到虚拟网卡。
- 3、 虚拟网卡把数据放到数据队列中。
- 4、 字符设备从数据队列中取数据，然后送给应用层。
- 5、 应用层把数据转发给物理网卡。
- 6、 物理网卡发送数据

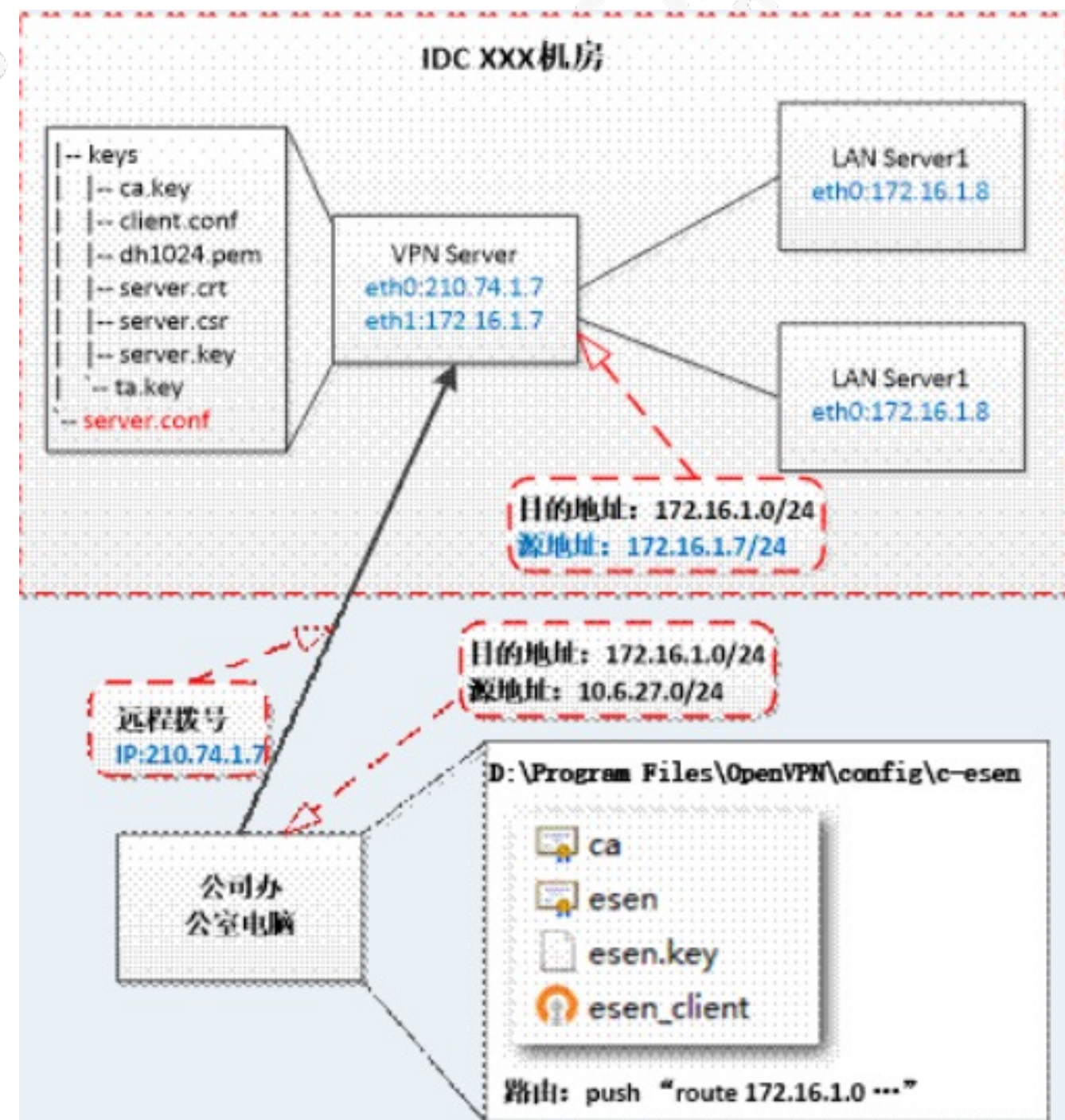
## □ 接收数据

- 1、 物理网卡接收到数据，并传到应用空间。
- 2、 应用守护程序通过字符设备，把数据传给驱动网卡。
- 3、 数据通过虚拟网卡重新进入网络堆栈。
- 4、 网络堆栈把数据传给上层真实的应用程序



# OpenVpn应用场景

- ❑ 1、个人电脑远程拨号到企业办公网络远程办公
- ❑ 2、个人电脑远程拨号到企业网站IDC机房，远程维护服务器。
- ❑ 3、企业办公网之间，或者一个企业的不同地点多个办公网之间，或者是多个IDC机房之间。



版权所有，侵权必究

# OpenVpn实战部署

- ❑ 项目1：配置远程访问VPN
- ❑ 环境规划

主机名	网卡	默认网关	用途
VPN Server	192.168.150.11（ens33，内网卡） 200.1.1.1（ens37，外网卡）	192.168.150.2	VPN Server
FTP Server	192.168.150.12	192.168.150.2	内网FTP Server
Remote Client	200.1.1.10		模拟外网测试客户机

说明：

VPN Server采用双网卡，网关指向192.168.150.2。

Remote Client 远程拨入VPN Server后要能访问内网FTP Server资源。

Remote Client拨号地址为200.1.1.1（模拟外网IP）

# OpenVpn配置

## □ 软件版本

Centos - 7.x

easy-rsa - 3.0.6

OpenVPN - 2.4.8

### 1、安装软件

配置epel

```
[root@node1 ~]# wget -O /etc/yum.repos.d/epel.repo  
http://mirrors.aliyun.com/repo/epel-7.repo
```

```
[root@node1 ~]# yum install openvpn easy-rsa -y
```

### 2、配置证书密钥

```
[root@node1 ~]# cp -r /usr/share/easy-rsa/3.0.6/  
/etc/openvpn/server/easy-rsa
```



# OpenVpn配置

## □ 配置步骤

### 2、配置证书密钥

```
[root@node1 ~]# cp -r /usr/share/easy-rsa/3.0.6/
/etc/openvpn/server/easy-rsa
[root@node1 ~]# cd /etc/openvpn/server/easy-rsa
[root@node1 easy-rsa]# cat vars
set_var EASYRSA_REQ_COUNTRY    "CN"        #国家
set_var EASYRSA_REQ_PROVINCE   "BJ"        #省
set_var EASYRSA_REQ_CITY       "Beijing"    #城市
set_var EASYRSA_REQ_ORG        "MyORG"      #组织
set_var EASYRSA_REQ_EMAIL      "vpn@kongd.com" #邮箱
set_var EASYRSA_REQ_OU         "it"         #公司、组织
```

#初始化pki，生成目录文件结构

```
[root@node1 easy-rsa]# ./easyrsa init-pki
```

...

```
Your newly created PKI dir is: /etc/openvpn/server/easy-rsa/pki
```

版权所有，侵权必究

# OpenVpn配置

## □ 配置步骤

### 2、配置证书密钥

#创建ca证书

```
[root@node1 easy-rsa]# ./easyrsa build-ca
```

#使用vars文件里面配置的信息

Note: using Easy-RSA configuration from: ./vars

服务端证书server.crt

```
[root@node1 easy-rsa]# ./easyrsa gen-req server nopass
```

#nopass设置免证书密码，如果要设置密码可以取消此参数选项

证书签名

```
[root@node1 easy-rsa]# ./easyrsa sign server server
```

#第二个server是只上面服务端证书的CN名字，我们用的默认server

dh证书

```
[root@node1 easy-rsa]# ./easyrsa gen-dh
```

版权所有，侵权必究

# OpenVpn配置

## □ 配置步骤

### 2、配置证书密钥

ta密钥

```
[root@node1 easy-rsa]# cd /etc/openvpn/
```

```
[root@node1 openvpn]# openvpn --genkey --secret ta.key
```

生成客户端证书

为了便于区别，我们把客户端使用的证书存放在新的路径。

```
[root@node1 openvpn]# mkdir -p /etc/openvpn/client
```

```
[root@node1 openvpn]# cd /etc/openvpn/client
```

```
[root@node1 client]# cp -r /usr/share/easy-rsa/3.0.6/* .
```

```
[root@node1 client]# cp /etc/openvpn/server/easy-rsa/vars .
```

```
[root@node1 client]# ./easyrsa init-pki
```

```
[root@node1 client]# ./easyrsa gen-req client nopass
```

# OpenVpn配置

## □ 配置步骤

### 2、配置证书密钥

对客户端证书签名

切换到服务端easy-rsa目录

```
[root@node1 client]# cd /etc/openvpn/server/easy-rsa/
```

#导入req

```
[root@node1 easy-rsa]# ./easyrsa import-req  
/etc/openvpn/client/pki/reqs/client.req client
```

#签名, 第一个client是固定的参数表示客户端, 第二个client指上面导入的客户端证书名

```
[root@node1 easy-rsa]# ./easyrsa sign client client
```

版权所有，侵权必究



# OpenVpn配置

## □ 配置步骤

### 3、配置 Server 端

创建Server配置文件

服务器端证书和密钥统一放到和server.conf一个目录下，便于配置

```
# cp /etc/openvpn/server/easy-rsa/pki/ca.crt /etc/openvpn/  
# cp /etc/openvpn/server/easy-rsa/pki/private/server.key  
    /etc/openvpn/  
# cp /etc/openvpn/server/easy-rsa/pki/issued/server.crt  
    /etc/openvpn/  
# cp /etc/openvpn/server/easy-rsa/pki/dh.pem /etc/openvpn/  
  
/etc/openvpn/server.conf
```

启动服务：

注意：开启路由转发 net.ipv4.ip\_forward = 1

```
# systemctl -f enable openvpn@server.service  
# systemctl start openvpn@server.service
```

版权所有，侵权必究

# OpenVpn配置

## 配置步骤

### 4、配置 client 端

将client.key、client.crt、ca.crt和ta.key下载到windows

windows客户端安装openvpn-install-2.4.4-1601

客户端配置文件：



客户端连接，连接后查看IP



版权所有，侵权必究

# OpenVpn配置

## □ 配置步骤

### 4、测试

访问内网ftp服务器

```
C:\Users\Administrator>ftp 192.168.150.12
连接到 192.168.150.12。
220 (vsFTPd 3.0.2)
用户(192.168.150.12:(none)): anonymous
331 Please specify the password.
密码:
230 Login successful.
ftp> bye
221 Goodbye.
```

服务器端查看:

```
[root@node1 ~]# more /etc/openvpn/ipp.txt
```

```
client, 10.8.0.4
```

查看状态日志:

```
[root@node1 ~]# more /etc/openvpn/openvpn-status.log
```

OpenVPN CLIENT LIST

Updated, Mon Dec 2 11:02:16 2019

Common Name, Real Address, Bytes Received, Bytes Sent, Connected

Since

```
client, 200.1.1.2:49165, 12281, 6594, Mon Dec 2 10:56:18 2019
```

ROUTING TABLE

Virtual Address, Common Name, Real Address, Last Ref

```
10.8.0.6, client, 200.1.1.2:49165, Mon Dec 2 11:02:07 2019
```

GLOBAL STATS

Max bcast/mcast queue length, 0

END

版权所有，侵权必究

# OpenVpn配置

## □ 配置OpenVPN支持用户名密码验证

### 1、修改服务器配置文件

增加以下几行

```
script-security 3 #允许通过环境变量将密码传递给脚本
```

```
auth-user-pass-verify /etc/openvpn/checkpsw.sh via-env #提供一个用户名密码对
```

```
client-cert-not-required #不使用客户端证书，使用密码对
```

```
username-as-common-name #使用认证用户名，不使用证书的  
common name
```

### 2、配置用户登录脚本

```
# vim /etc/openvpn/checkpsw.sh
```

注意：脚本要有执行权限

```
# chmod +x /etc/openvpn/checkpsw.sh
```



# OpenVpn配置

## ❑ 配置OpenVPN支持用户名密码验证

3、新建用户名和密码认证文件psw-file，用户名和密码用空格隔开，同时确保openvpn启动用户可读取该文件

```
echo "test1 test1" > /etc/openvpn/psw-file
```

```
chmod 400 /etc/openvpn/psw-file
```

```
chown nobody.nobody /etc/openvpn/psw-file
```

4、重启OpenVpn服务

```
systemctl restart openvpn@server.service
```

5、客户端配置文件修改

1>注释掉

```
;cert yangliangwei.crt
```

```
;key yangliangwei.key
```

#增加密码验证后，客户端只需包含ca.crt的配置文件

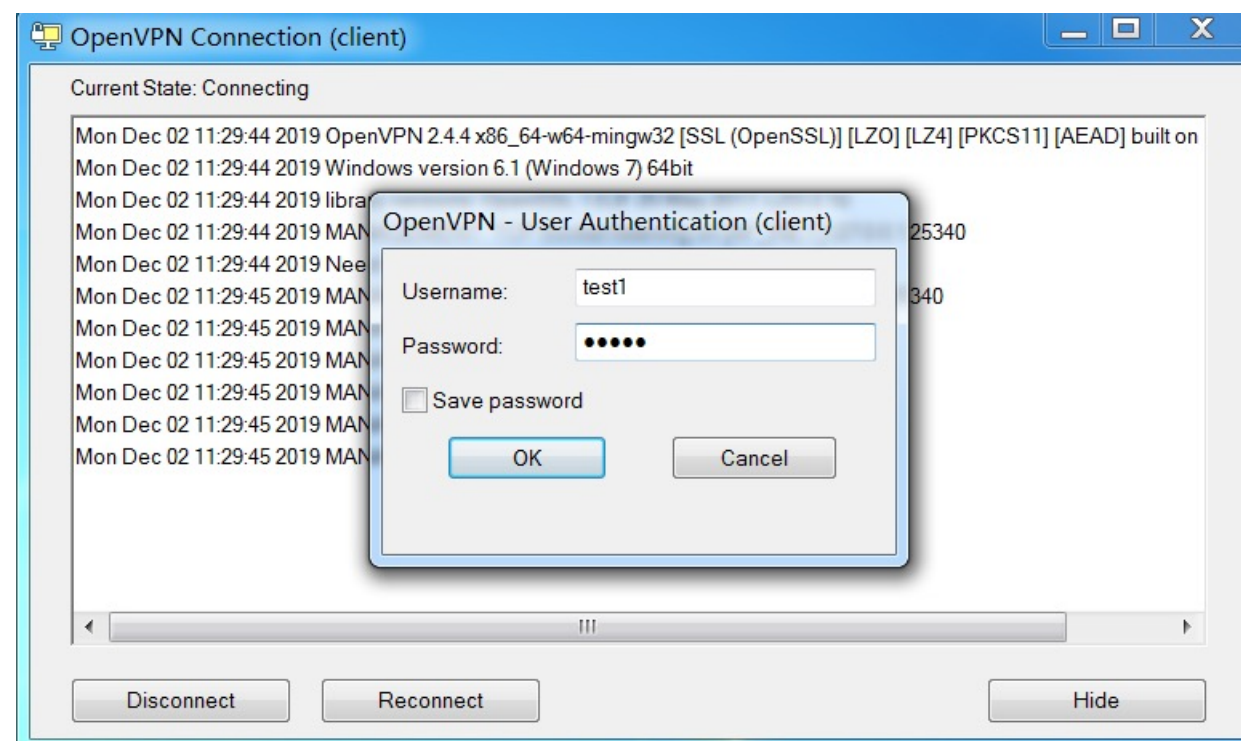
2>增加询问用户名和密码

```
auth-user-pass
```

版权所有，侵权必究

# OpenVpn配置

## □ 客户端拨号测试



```
C:\Users\Administrator>ftp 192.168.150.12
连接到 192.168.150.12。
220 (vsFTPd 3.0.2)
用户(192.168.150.12:(none)): anonymous
331 Please specify the password.
密码:
230 Login successful.
```

## □ 服务器查看

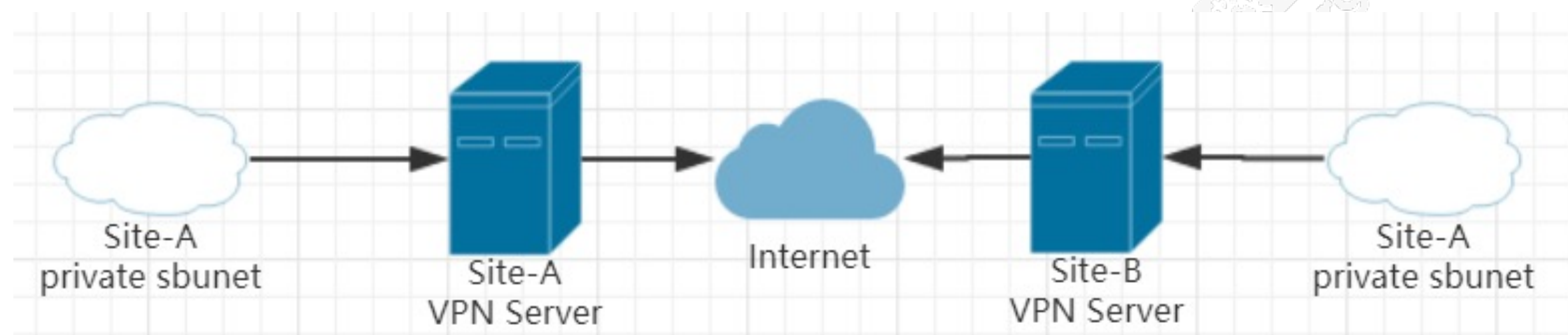
```
[root@node1 ~]# more /etc/openvpn/ipp.txt
client, 10.8.0.4
test1, 10.8.0.8
```

版权所有，侵权必究

# OpenVpn实战部署

## □ 项目2：配置站点到站点的VPN

### □ 环境规划



说明：

1、Site-A 为北京公司， Site-B 为上海公司

2、Vpnserver为client定义不同的路由

Vpn-server需要设置到上海内网的路由 "route 172.16.20.0 255.255.255.0"

sh需要推送到北京路由 push "route 172.16.10.0 255.255.255.0"

具体实现方法：为不同的client创建自定义的子配置文件，包含到主配置文件

在内网主机上指定网关：VPN Server内网地址

版权所有，侵权必究



# OpenVpn实战部署

- 项目2：配置站点到站点的VPN
- 环境规划

主机名	网卡	默认网关	用途
bj-vpnserver	192.168.150.11（ens33，外网卡）	192. 168. 150. 2	VPN Server
	172.16.10.11（ens37，内网卡）		
sh-vpnclient	192.168.150.12（ens33，外网卡）	192. 168. 150. 2	VPN Client
	172.16.20.12（ens37，内网卡）		
bj-client	172. 16. 10. 100	172. 16. 10. 11	模拟北京公司客户机
sh-client	172. 16. 20. 100	172. 16. 20. 12	模拟上海公司客户机

说明：

VPN Server采用双网卡，网关指向192. 168. 150. 2。

VPN Client采用双网卡，网关指向192. 168. 150. 2

VPN Server和VPN Client开启路由转发

模拟公司内网客户机网关执行VPN的内网卡地址。



# OpenVpn配置

## □ bj-vpnserver配置

安装openvpn软件

CA配置

自签名证书

为 bj-vpnserver 签发证书

为 sh-vpnclient 签发证书

1、使用之前生成的服务端证书代表bj-vpnserver

2、为sh-vpnclient签发证书

```
[root@bj-vpnserver client]# ./easyrsa gen-req sh nopass
```

```
[root@bj-vpnserver client]# cd /etc/openvpn/server/easy-rsa/
```

```
[root@bj-vpnserver easy-rsa]# ./easyrsa import-req /etc/openvpn/client/pki/reqs/sh.req sh
```

```
[root@bj-vpnserver easy-rsa]# ./easyrsa sign client sh
```

3、将之前测试密码验证的删除

版权所有，侵权必究

# OpenVpn配置

## □ bj-vpnserver配置

3、将之前测试密码验证的删除  
添加及修改：

```
client-config-dir ccd  
route 172.16.20.0 255.255.255.0  
push "route 172.16.10.0 255.255.255.0"  
;script-security 3  
;auth-user-pass-verify /etc/openvpn/checkpsw.sh via-env  
;client-cert-not-required  
;username-as-common-name
```

为sh-vpnclient创建自定义的子配置文件

```
[root@bj-vpnserver openvpn]# mkdir /etc/openvpn/ccd  
[root@bj-vpnserver openvpn]# vim /etc/openvpn/ccd/sh  
route 172.16.20.0 255.255.255.0
```

开启路由转发，重启服务：

```
# systemctl restart openvpn@server.service
```

# OpenVpn配置

## □ sh-vpnclient配置

1、安装openvpn软件

2、从vpnsrver端复制文件：

```
[root@bj-vpnserver openvpn]# scp /etc/openvpn/ca.crt  
192.168.150.12:/etc/openvpn/
```

```
[root@bj-vpnserver openvpn]# scp  
/etc/openvpn/client/pki/private/sh.key  
192.168.150.12:/etc/openvpn/
```

```
[root@bj-vpnserver openvpn]# scp  
/etc/openvpn/server/easy-rsa/pki/issued/sh.crt  
192.168.150.12:/etc/openvpn/
```

```
[root@bj-vpnserver openvpn]# scp /etc/openvpn/ta.key  
192.168.150.12:/etc/openvpn/
```

# OpenVpn配置

## □ sh-vpnclient配置

### 3、配置文件

```
[root@sh-vpnclient ~]# cat /etc/openvpn/server.conf
client
user nobody
group nobody
dev tun
proto tcp
remote 192.168.150.11 1194
ca ca.crt
cert sh.crt
key sh.key
remote-cert-tls server
tls-auth ta.key 1
cipher AES-256-CBC
comp-lzo
persist-key
persist-tun
verb 3
```

版权所有，侵权必究



# OpenVpn配置

## ❑ sh-vpnclient配置

### 4、开启路由转发，启动服务

```
[root@sh-vpnclient ~]# grep "net.ipv4.ip_forward"  
/etc/sysctl.conf
```

```
net.ipv4.ip_forward = 1
```

```
[root@sh-vpnclient ~]# systemctl -f enable
```

```
openvpn@server.service
```

```
[root@sh-vpnclient ~]# systemctl start
```

```
openvpn@server.service
```

### 5、客户端测试

win7模拟北京公司内部机器

node3模拟上海公司内部办公机器

```
C:\Users\Administrator>ipconfig  
  
Windows IP 配置  
  
以太网适配器 本地连接 2:  
  
媒体状态 . . . . . : 媒体已断开  
连接特定的 DNS 后缀 . . . . . :  
  
以太网适配器 本地连接:  
  
连接特定的 DNS 后缀 . . . . . :  
本地连接 IPv6 地址. . . . . : fe80::505f:7380:e20c:58e0%11  
IPv4 地址 . . . . . : 172.16.10.100  
子网掩码 . . . . . : 255.255.255.0  
默认网关 . . . . . : 172.16.10.11
```

```
C:\Users\Administrator>ping 172.16.20.100  
  
正在 Ping 172.16.20.100 具有 32 字节的数据:  
来自 172.16.20.100 的回复: 字节=32 时间=1ms TTL=62  
来自 172.16.20.100 的回复: 字节=32 时间=1ms TTL=62  
来自 172.16.20.100 的回复: 字节=32 时间=1ms TTL=62  
来自 172.16.20.100 的回复: 字节=32 时间=1ms TTL=62  
  
172.16.20.100 的 Ping 统计信息:  
数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),  
往返行程的估计时间(以毫秒为单位):  
最短 = 1ms, 最长 = 1ms, 平均 = 1ms
```

# OpenVpn配置

## □ sh-vpnclient配置

### 5、客户端测试

win7模拟北京公司内部机器

node3模拟上海公司内部办公机器

```
[root@node3 ~]# ifconfig ens33
ens33: flags=4163<UP, BROADCAST, RUNNING, MULTICAST> mtu 1500
        inet 172.16.20.100 netmask 255.255.255.0 broadcast
        172.16.20.255
[root@node3 ~]# ping -c2 172.16.10.100
PING 172.16.10.100 (172.16.10.100) 56(84) bytes of data.
64 bytes from 172.16.10.100: icmp_seq=1 ttl=126 time=1.12 ms
64 bytes from 172.16.10.100: icmp_seq=2 ttl=126 time=1.10 ms

--- 172.16.10.100 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 1.103/1.113/1.124/0.034 ms
```

版权所有，侵权必究

# 总结

- OpenVPN简介
- OpenVpn通信原理
- OpenVpn应用场景
- OpenVpn实战案例

版权所有，侵权必究



# 谢谢观看

更多好课，请关注万门大学APP

