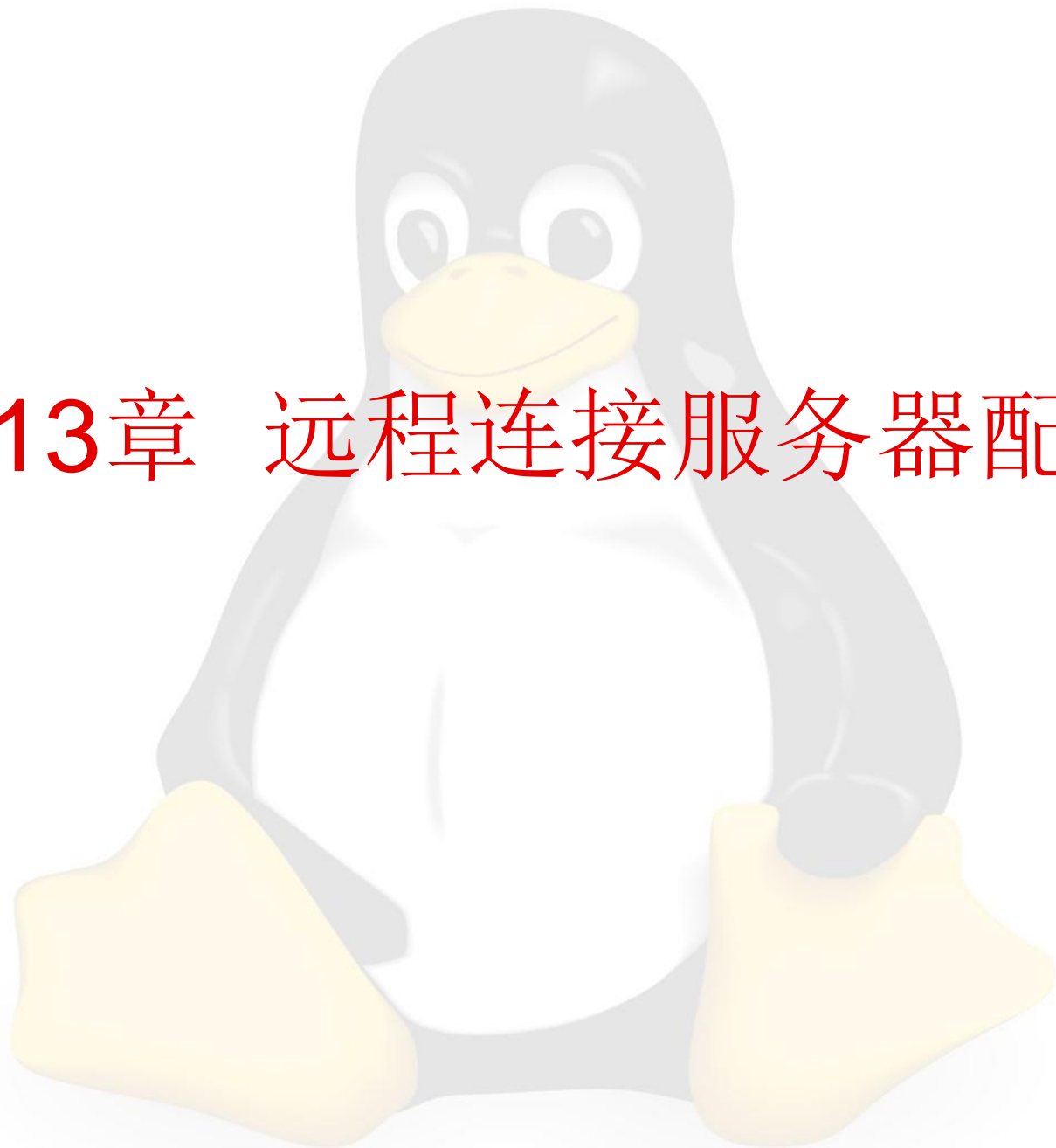


第13章 远程连接服务器配置



本章内容

- 13.1 SSH和OpenSSH简介
- 13.2 OpenSSH服务器安装和配置
- 13.3 配置OpenSSH客户端
- 13.4 VNC服务器配置
- 13.5 连接VNC服务器

13.1 SSH和OpenSSH简介

- 使用SSH可以在本地主机和远程服务器之间进行加密地传输数据，实现数据的安全传输。而OpenSSH是SSH协议的免费开源实现，它用安全、加密的网络连接工具代替了telnet，ftp，rlogin，rsh和rcp工具。

什么是SSH

- **ftp**和**telnet**在本质上是不安全的，因为它们在网上使用明文传输口令和数据，别有用心的人非常容易就可以截获这些口令和数据。而且这些程序的安全验证方式也是有弱点的，很容易受到“中间人”这种方式的攻击。
- **SSH**（**Secure Shell**，安全Shell）是由**IETF**的网络工作小组所制定，为建立在应用层和传输层基础上的安全协议。**SSH**是目前较可靠，专为远程登录会话和其它网络服务提供安全性的协议。利用**SSH**协议可以有效防止远程管理过程中的信息泄露问题。
- 通过使用**SSH**可以把所有传输的数据进行加密，这样“中间人”这种攻击方式就不可能实现了，而且也可以防止**DNS**和**IP**欺骗。还有一个额外的好处就是传输的数据是经过压缩的，所以可以加快传输的速度。**SSH**有很多功能，它既可以代替**telnet**，又可以为**ftp**、**pop**和**ppp**提供一个安全的通道。

什么是OpenSSH

- SSH因为受版权和加密算法的限制，现在很多人都转而使用OpenSSH。OpenSSH（Open Secure Shell，开放安全Shell）是SSH的替代软件，而且是免费的。默认使用RSA密钥，它采用安全、加密的网络连接工具代替telnet、ftp、rlogin、rsh和rcp工具。
- 使用OpenSSH工具将增进系统安全性，在使用OpenSSH软件进行通信时，登录验证口令将会被加密。OpenSSH提供了服务端后台程序和客户端工具，用来加密远程控件和文件传输过程中的数据，并由此来代替原来的类似服务。
- telnet和ftp使用纯文本口令，并以明文发送。这些信息可能会被截取，口令可能会被检索，未经授权的人员可能会使用截取的口令登录用户的系统，而对系统产生危害，所以应该尽可能使用OpenSSH工具来避免这些安全问题。
- 另一个使用OpenSSH的原因是，它自动把DISPLAY变量转发给客户主机。如果在本地主机上运行X窗口系统，并且使用ssh命令登录到远程主机上，当在远程主机上执行一个需要X的程序时，该程序会在本地主机上执行。

安装OpenSSH服务器软件包

- 安装openssh-server、openssh、openssh-clients和openssh-askpass软件包。

```
[root@rhel ~]# cd /run/media/root/RHEL-7.2\  
Server.x86_64/Packages
```

```
[root@rhel Packages]# rpm -ivh openssh-6.6.1p1-  
22.el7.x86_64.rpm
```

```
[root@rhel Packages]# rpm -ivh openssh-server-6.6.1p1-  
22.el7.x86_64.rpm
```

```
[root@rhel Packages]# rpm -ivh openssh-clients-6.6.1p1-  
22.el7.x86_64.rpm
```

```
[root@rhel Packages]# rpm -ivh openssh-askpass-  
6.6.1p1-22.el7.x86_64.rpm
```

/etc/ssh/sshd_config文件详解（1）

- OpenSSH服务器的主配置文件是 `/etc/ssh/sshd_config` 文件，这个文件的每一行都是“关键词值”的格式。一般情况下不需要配置该文件即可让用户在客户端计算上进行连接。
- 在 `/etc/ssh/sshd_config` 配置文件中，以“#”开头的行是注释行，它为用户配置参数起到解释作用，这样的语句默认不会被系统执行。

/etc/ssh/sshd_config文件详解（2）

Port 22

设置OpenSSH服务器监听的端口号，默认为22。

ListenAddress 0.0.0.0

设置OpenSSH服务器绑定的IP地址。

HostKey /etc/ssh/ssh_host_key

设置包含计算机私有主机密钥的文件。

ServerKeyBits 1024

设置服务器密钥的位数。最小值是512，默认为1024。

LoginGraceTime 2m

设置如果用户不能成功登录，在切断连接之前服务器需要等待的时间。

PermitRootLogin yes

设置root用户是否能够使用ssh登录。

/etc/ssh/sshd_config文件详解（3）

IgnoreRhosts yes

设置RhostsRSA验证和Hostbased验证的时候是否使用.rhosts和.shosts文件。

IgnoreUserKnownHosts no

设置sshd是否在进行RhostsRSAAuthentication安全验证的时候忽略用户的~/.ssh/known_hosts。

StrictModes yes

设置ssh在接收登录请求之前是否检查用户主目录和rhosts文件的权限和所有权。这通常是必要的，因为新手经常会把自己的目录和文件设成任何人都有写权限。

PrintMotd yes

设置sshd是否在用户登录的时候显示/etc/motd文件中的信息。

LogLevel INFO

设置记录sshd日志消息的级别。

/etc/ssh/sshd_config文件详解（4）

RhostsRSAAuthentication no

设置是否允许用rhosts或/etc/hosts.equiv加上RSA进行安全验证。

RSAAuthentication yes

设置是否允许只有RSA安全验证。

PasswordAuthentication yes

设置是否允许口令验证。

PermitEmptyPasswords no

设置是否允许用户口令为空字符串的账号登录，默认是no。

AllowGroups

设置允许连接的组群。

AllowUsers

设置允许连接的用户。

DenyGroups

设置拒绝连接的组群。

/etc/ssh/sshd_config文件详解（5）

DenyUsers

设置拒绝连接的用户。如果模式写成USER@HOST，则USER和HOST将同时被检查，限制特定用户在特定主机上连接OpenSSH服务器。比如zhangsan@192.168.0.5表示拒绝用户zhangsan在主机192.168.0.5上连接OpenSSH服务器。

MaxSessions 10

指定允许每个网络连接打开的最大会话数，默认为10。

ClientAliveCountMax 3

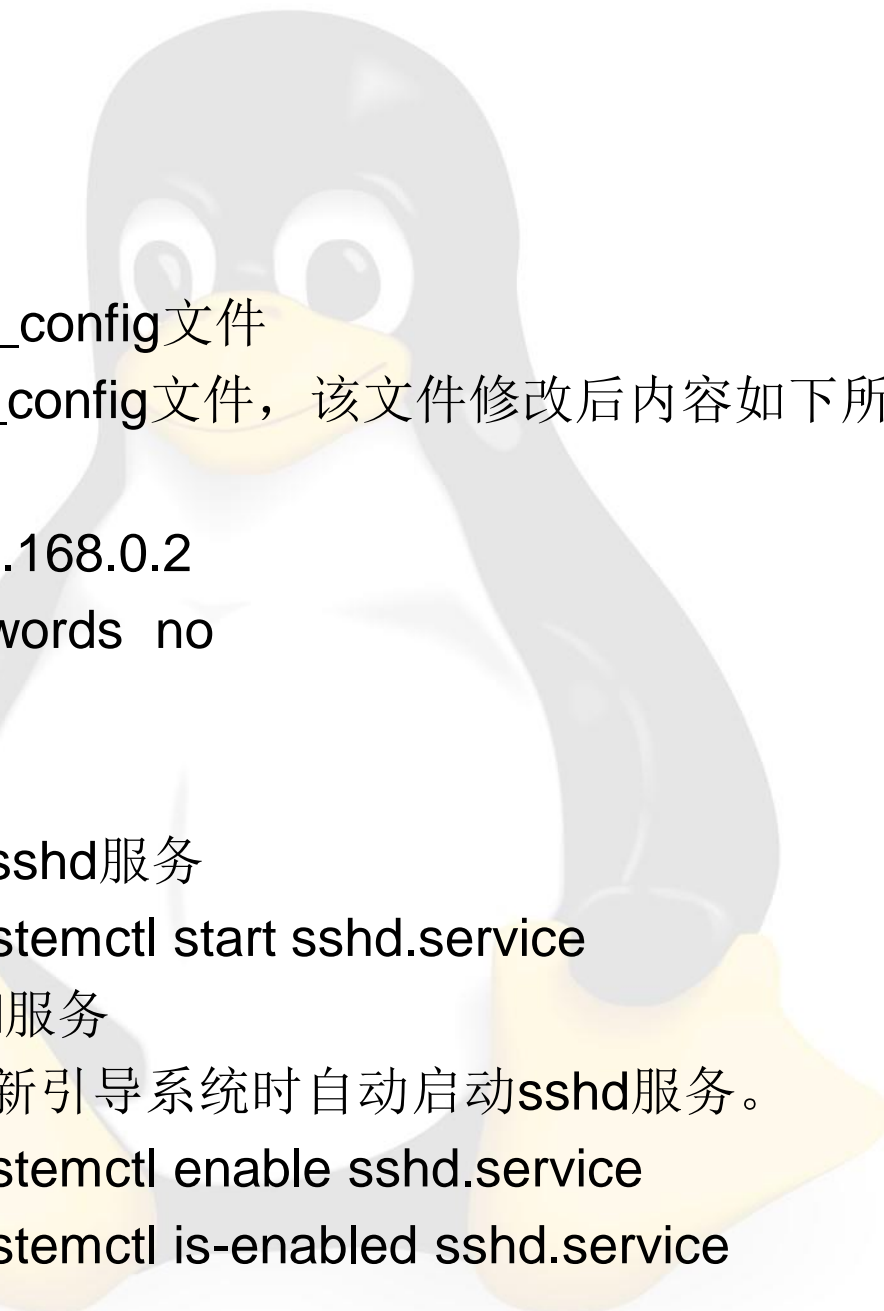
指定从客户端断开连接之前，在没有接收到响应时能够发送客户端活跃消息的次数。这个参数设置允许超时的次数。

MaxStartups 10:30:100

指定SSH守护进程未经身份验证的并发连接的最大数量，默认值是10:30:100。10:30:100表示的意思是，从第10个连接开始，以30%的概率（递增）拒绝新的连接，直到连接数达到100。

OpenSSH服务器配置实例

- 在公司内部配置一台OpenSSH服务器，为公司网络内的客户端计算机提供远程SSH登录服务，具体参数如下。
 - OpenSSH服务器IP地址：192.168.0.2。
 - OpenSSH服务器监听端口：22。
 - 不允许空口令用户登录。
 - 禁止用户lisi登录。

- 
1. 编辑/etc/ssh/sshd_config文件
修改/etc/ssh/sshd_config文件，该文件修改后内容如下所示。

Port 22

ListenAddress 192.168.0.2

PermitEmptyPasswords no

DenyUsers lisi

2. 启动sshd服务
使用以下命令启动sshd服务

```
[root@rhel ~]#systemctl start sshd.service
```

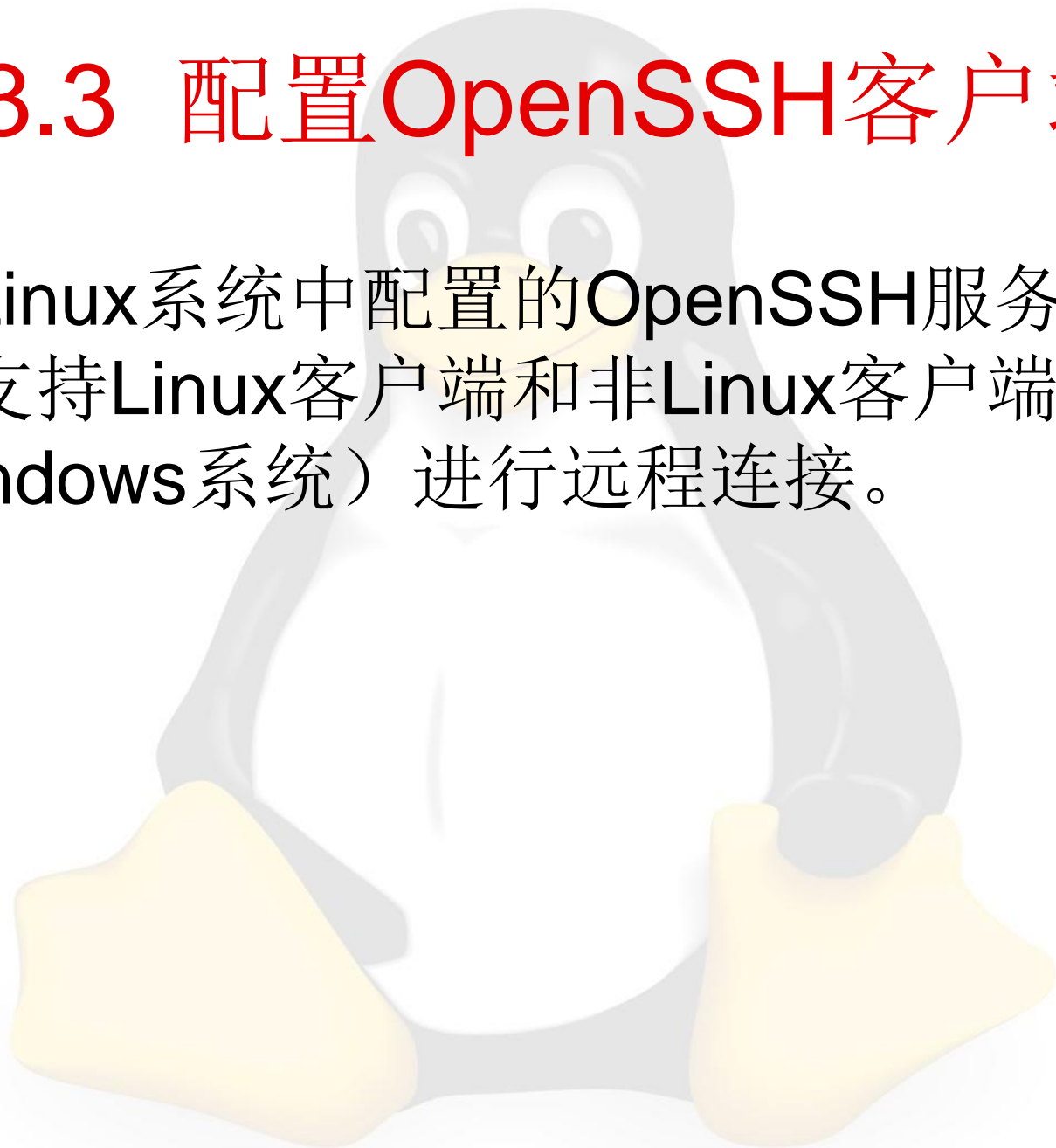
3. 开机自动启动sshd服务
使用以下命令在重新引导系统时自动启动sshd服务。

```
[root@rhel ~]#systemctl enable sshd.service
```

```
[root@rhel ~]#systemctl is-enabled sshd.service  
enabled
```

13.3 配置OpenSSH客户端

- 在Linux系统中配置的OpenSSH服务器可以支持Linux客户端和非Linux客户端（如Windows系统）进行远程连接。



安装软件包

- 安装openssh-clients和openssh软件包。

```
[root@linux ~]# cd /run/media/root/RHEL-7.2\  
Server.x86_64/Packages
```

```
[root@linux Packages]# rpm -ivh openssh-clients-6.6.1p1-  
22.el7.x86_64.rpm
```

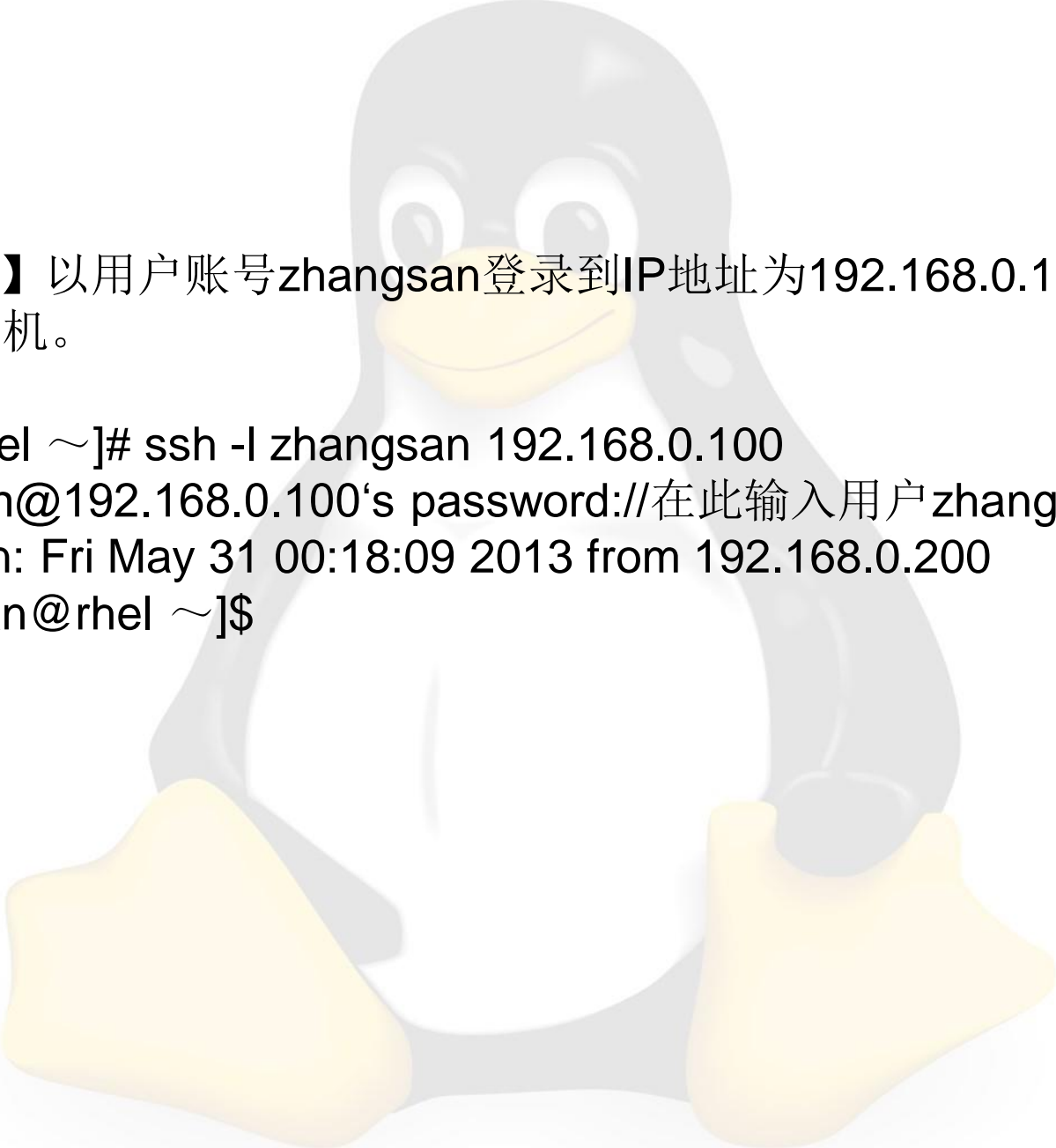
```
[root@linux Packages]# rpm -ivh openssh-6.6.1p1-  
22.el7.x86_64.rpm
```

ssh命令

- 使用**ssh**命令可以用于登录到远程计算机和在远程计算机上执行命令，它是为了取代**rlogin**和**rsh**。**ssh**连接并登录到指定的主机名（带有可选的用户名），用户必须证明身份以便能使用。

命令语法：

ssh [选项] [用户@]主机 [命令]



【例13.1】 以用户账号zhangsan登录到IP地址为192.168.0.100的远程SSH计算机。

```
[root@rhel ~]# ssh -l zhangsan 192.168.0.100
zhangsan@192.168.0.100's password://在此输入用户zhangsan的密码
Last login: Fri May 31 00:18:09 2013 from 192.168.0.200
[zhangsan@rhel ~]$
```

【例13.2】 以root账号连接远程主机192.168.0.100，并执行ls /boot命令。

```
[root@rhel ~]# ssh root@192.168.0.100 ls /boot
```

```
root@192.168.0.100's password:           //在此输入用户root的密码
```

```
config-3.10.0-327.el7.x86_64
```

```
extlinux
```

```
grub2
```

```
..... (省略)
```

```
[root@rhel ~]#
```

//执行该命令后会看到远程主机/boot目录下的内容，然后就会返回到本地Shell提示下

使用scp命令

- 使用**scp**命令可以用来通过安全、加密的连接在不同主机之间传输文件，它与**rcp**相似。

命令语法：

scp [选项] [[用户@]主机1:]文件1 [[用户@]主机2:]文件2



【例13.3】 用root账号把本地文件root/a传送到192.168.0.100远程主机下的/root下，并改名为b。

```
[root@rhel ~]# scp /root/a root@192.168.0.100:/root/b
root@192.168.0.100's password:          //在此输入用户root的口令
a                100% 1222      1.2KB/s   00:00
[root@rhel ~]# ssh root@192.168.0.100 ls /root/b
root@192.168.0.100's password:          //在此输入用户root的口令
/root/b
```

【例13.4】 用root账号把本地/ab目录下所有文件传送到192.168.0.100远程主机的/root目录。

```
[root@rhel ~]# scp /ab/* root@192.168.0.100:/root
root@192.168.0.100's password: //在此输入用户root的口令
abc                             100%    0    0.0KB/s   00:00
[root@rhel ~]# ssh root@192.168.0.100 ls /root
root@192.168.0.100's password: //在此输入用户root的口令
abc
anaconda-ks.cfg
install.log
install.log.syslog
..... (省略)
```

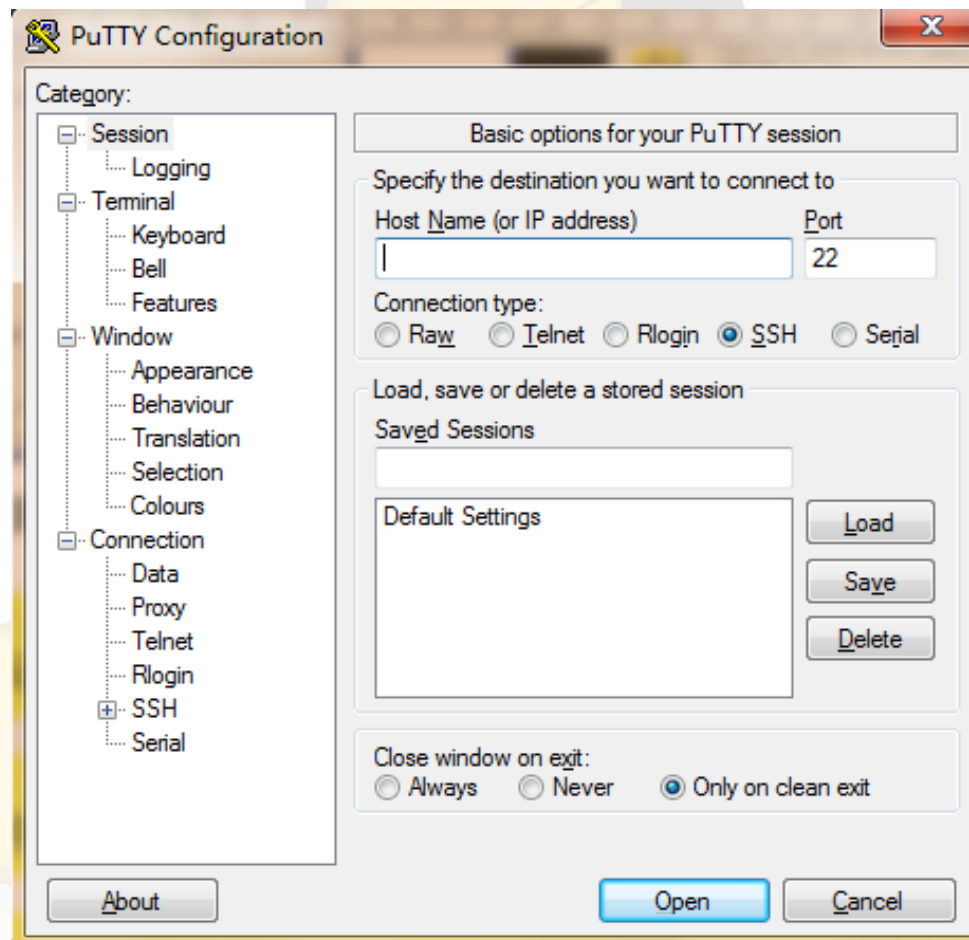
【例13.5】 用root账号把远程主机192.168.0.100上的文件/root/abc传送到本地主机/root目录下，并改名为a。

```
[root@rhel ~]# scp root@192.168.0.100:/root/abc /root/a
root@192.168.0.100's password:      //在此输入用户root的口令
abc                                100%    0    0.0KB/s   00:00
[root@rhel ~]# ls -l
总用量 100
-rw-r--r-- 1 root root      0 03-08 06:35 a
-rw----- 1 root root  3003 02-04 23:13 anaconda-ks.cfg
-rw-r--r-- 1 root root 58267 02-04 23:12 install.log
-rw-r--r-- 1 root root  8784 02-04 22:51 install.log.syslog
```

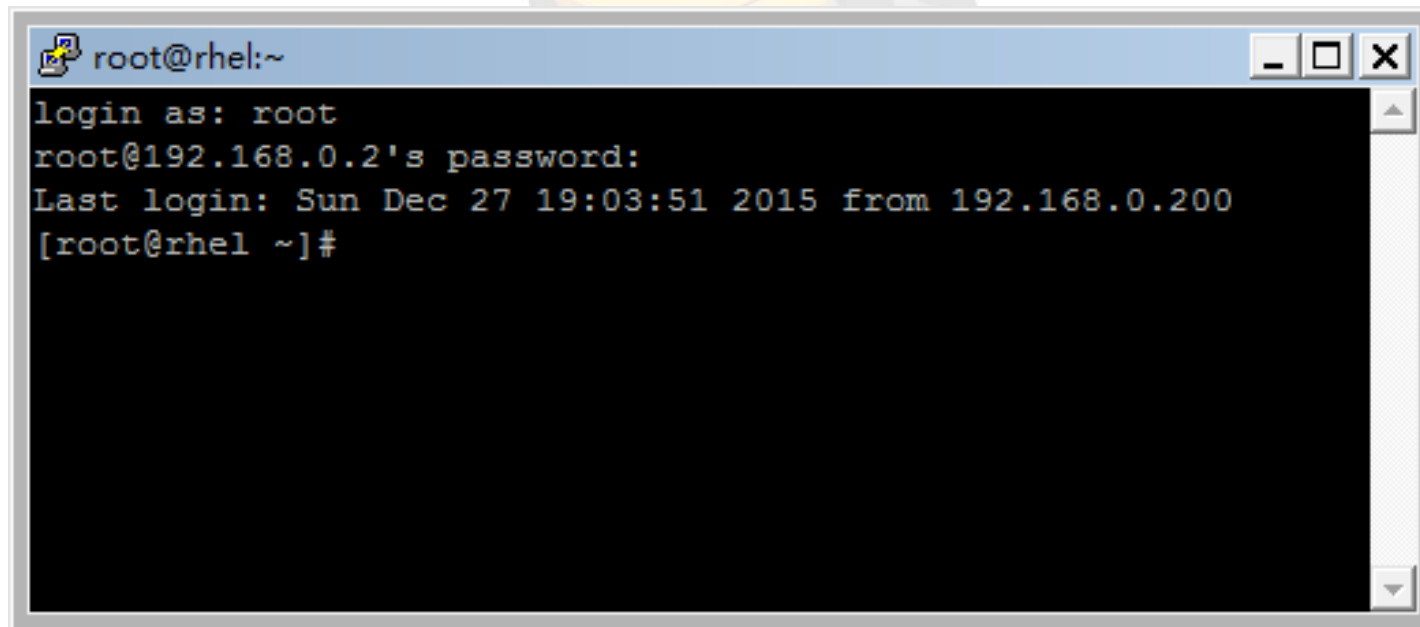
Windows客户端连接

- 在Windows操作系统下连接OpenSSH服务器可以通过PuTTY等软件实现。PuTTY是Windows平台下的一个免费的telnet, rlogin和SSH客户端，其功能丝毫不逊色于商业类的工具。

PuTTY软件



登录Linux主机



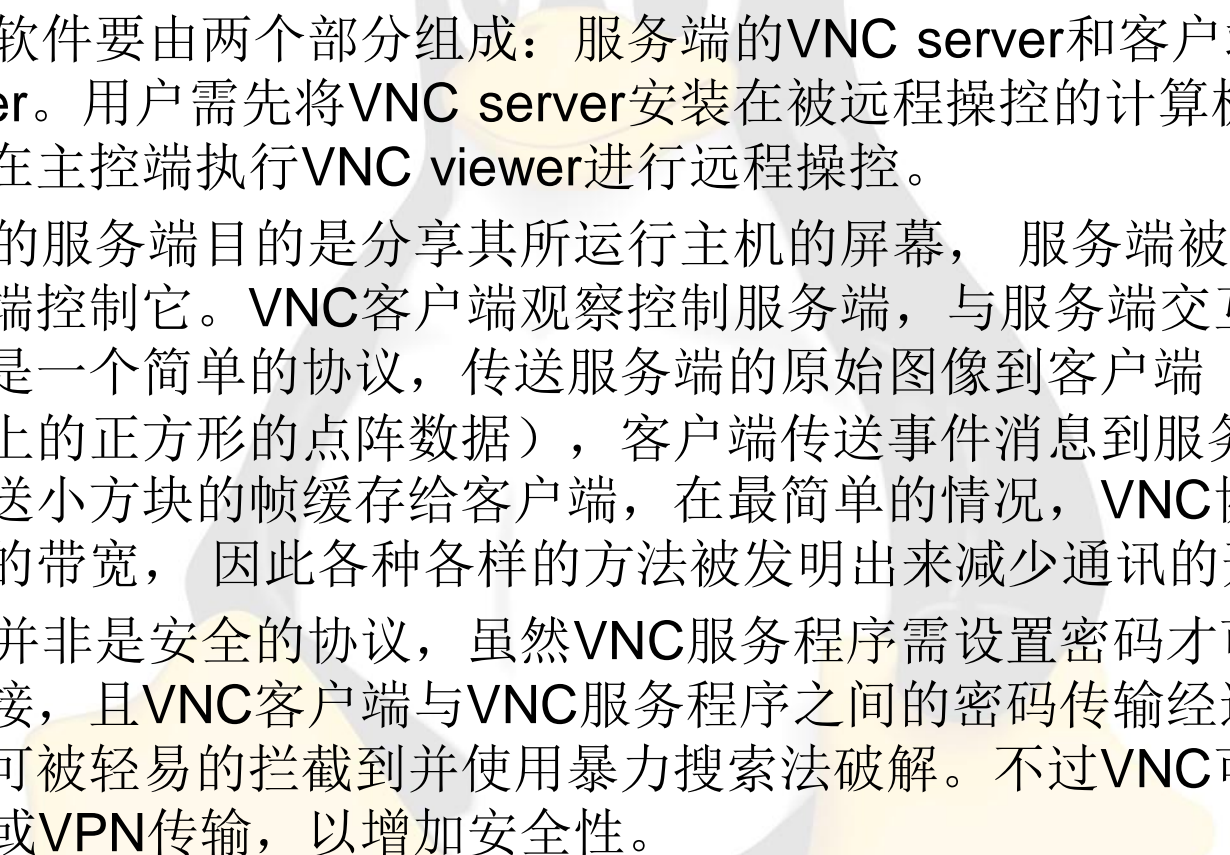
```
root@rhel:~  
login as: root  
root@192.168.0.2's password:  
Last login: Sun Dec 27 19:03:51 2015 from 192.168.0.200  
[root@rhel ~]#
```

13.4 VNC服务器配置

- 虚拟网络计算（Virtual Network Computing, VNC）是一款由AT&T欧洲研究实验室开发的远程控制软件，允许用户在网络的任何地方使用简单的程序来和一个特定的计算机（服务器）进行交互。

VNC简介

- VNC是基于Unix/Linux 操作系统的免费开源软件，远程控制能力强大，高效实用，其性能可以和Windows系统中的任何远程控制软件媲美。
- VNC基本上是属于一种显示系统，也就是说它能将完整的窗口界面通过网络，传输到另一台计算机的屏幕上。Windows系统上的Terminal Server和PCAnywhere都是属于这种原理的软件，同时这些软件又在VNC的原理基础上做了各自相应改进，提高了易用性、连通率和可穿透内网。
- 因为VNC是免费的，并且可以用于数量庞大的不同操作系统，它简单、可靠和向后兼容性，使之进化成为最为广泛使用的远程控制软件，多平台的支持对网络管理员是十分重要的，它使网络管理员可以使用一种工具管理几乎所有系统。

- 
- VNC软件要由两个部分组成：服务端的VNC server和客户端的VNC viewer。用户需先将VNC server安装在被远程操控的计算机上后，才能在主控端执行VNC viewer进行远程操控。
 - VNC的服务端目的是分享其所运行主机的屏幕， 服务端被动的允许客户端控制它。VNC客户端观察控制服务端，与服务端交互。VNC协议是一个简单的协议，传送服务端的原始图像到客户端（一个X,Y位置上的正方形的点阵数据），客户端传送事件消息到服务端。服务器发送小方块的帧缓存给客户端，在最简单的情况，VNC协议使用大量的带宽， 因此各种各样的方法被发明出来减少通讯的开支。
 - VNC并非是安全的协议，虽然VNC服务程序需设置密码才可接受外来连接，且VNC客户端与VNC服务程序之间的密码传输经过加密，但仍可被轻易的拦截到并使用暴力搜索法破解。不过VNC可设计以SSH或VPN传输，以增加安全性。

VNC服务器配置实例

1. 安装tigervnc-server软件包

```
[root@rhel ~]# yum -y install tigervnc-server
```



2. 启动VNC服务器

```
[root@rhel ~]# vncserver
```

```
Password:           //设置用户root的VNC登陆密码
```

```
Verify:
```

```
xauth: (stdin):1: bad display name "rhel:1" in "add" command
```

```
New 'rhel:1 (root)' desktop is rhel:1
```

```
Creating default startup script /root/.vnc/xstartup
```

```
Starting applications specified in /root/.vnc/xstartup
```

```
Log file is /root/.vnc/rhel:1.log
```

执行命令后，会要求为服务器设立一个保护密码，如果设置成功，会出现类似rhel:1 的提示，表示当前用户分配的是vnc的第一个虚拟桌面

3. 查看进程

```
[root@rhel ~]# ps -ef|grep Xvnc
```

```
root    2003    1  0 06:16 pts/1    00:00:00 /usr/bin/Xvnc :1 -desktop  
rhel:1 (root) -auth /root/.Xauthority -geometry 1024x768 -rfbwait  
30000 -rfbauth /root/.vnc/passwd -rfbport 5901 -fp  
catalogue:/etc/X11/fontpath.d -pn
```

//显示进程号为2003，使用的端口号为5901，虚拟桌面号是1

4. 查看端口号

```
[root@rhel ~]# netstat -ant|grep 5901
```

```
tcp      0      0 0.0.0.0:5901        0.0.0.0:*
```

LISTEN



创建或更改VNC登录密码

- 使用vncpasswd命令可以创建或更改一个VNC的登录密码，这将同时在用户的主目录下创建一个隐藏的目录“.vnc”，该目录内有一个文件passwd保存着VNC登录密码。

命令语法：

vncpasswd [密码文件]

vncpasswd [选项]

【例13.6】 创建或更改VNC登录密码。

```
[root@rhel ~]# vncpasswd
```

Password:

Verify:



管理VNC服务器

- 使用vncserver命令可以管理VNC服务器，比如启动和停止VNC服务器。

命令语法：

vncserver [:虚拟桌面号码] [选项] [Xvnc选项]

【例13.7】 列出当前用户的vnc虚拟桌面。

```
[root@rhel ~]#vncserver -list
```

TigerVNC server sessions:

X DISPLAY #	PROCESS ID
-------------	------------

:1	78363
----	-------

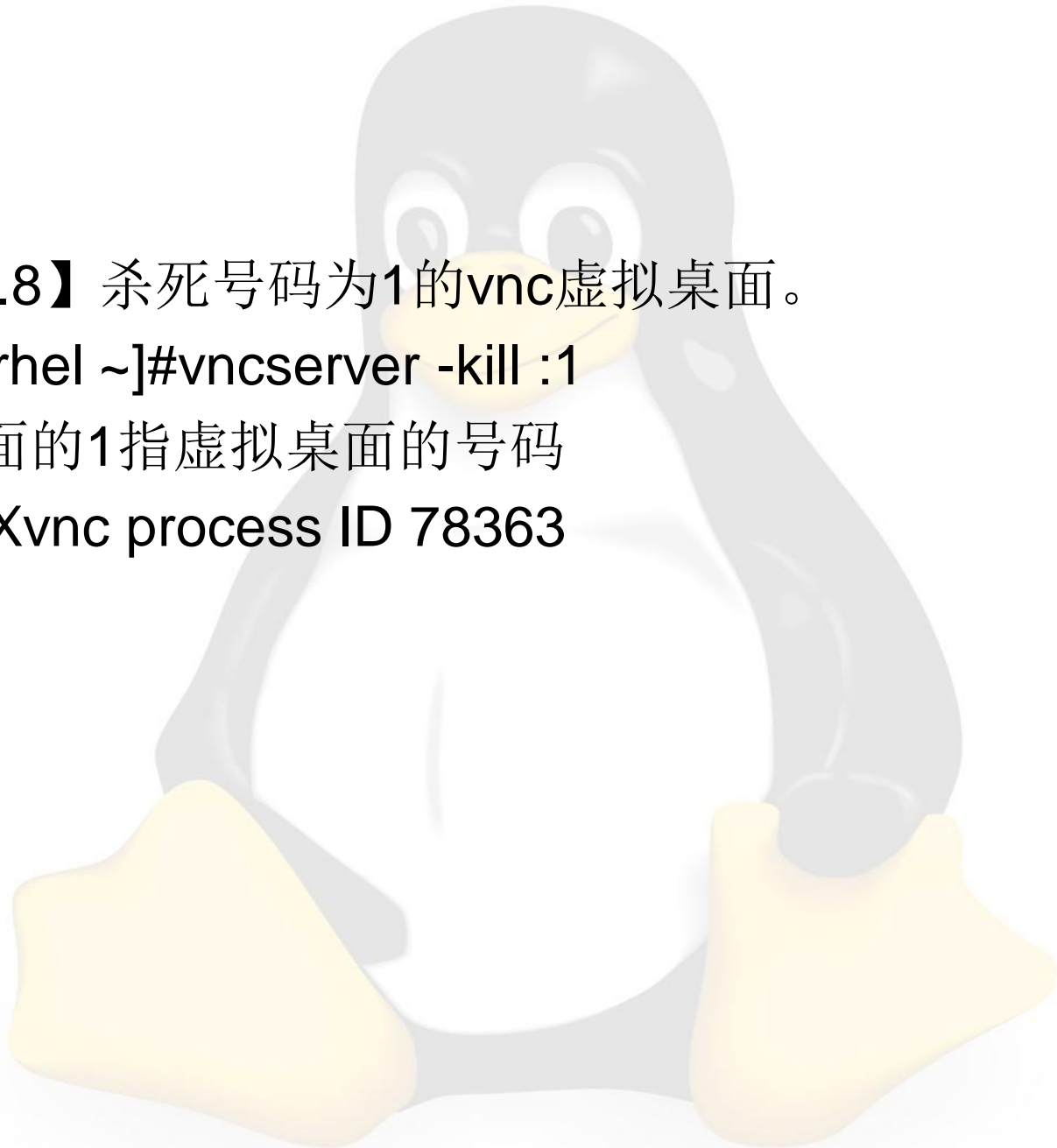



【例13.8】 杀死号码为1的vnc虚拟桌面。

```
[root@rhel ~]#vncserver -kill :1
```

//kill后面的1指虚拟桌面的号码

Killing Xvnc process ID 78363





【例13.9】 启动号码为5的vnc虚拟桌面。

```
[root@rhel ~]#vncserver :5
```

```
xauth: (stdin):1: bad display name "rhel:5" in "add" command
```

```
New 'rhel:5 (root)' desktop is rhel:5
```

```
Starting applications specified in /root/.vnc/xstartup
```

```
Log file is /root/.vnc/rhel:5.log
```

13.5 连接VNC服务器

- 在Linux系统中配置的VNC服务器可以支持Linux客户端和非Linux客户端（如Windows系统）以图形界面方式远程登录。



Linux客户端连接

- 安装tigervnc软件包

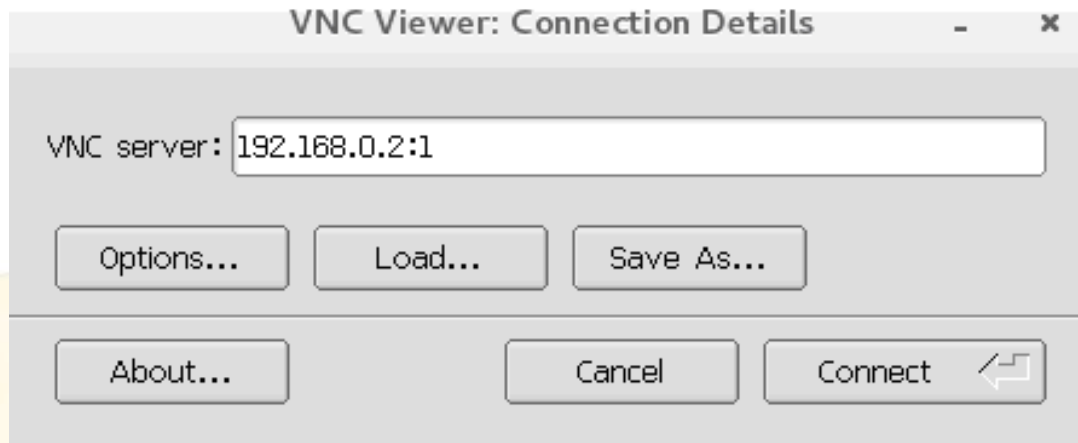
```
[root@rhel ~]# cd /run/media/root/RHEL-7.2\ Server.x86_64/Packages
```

```
[root@rhel Packages]# rpm -ivh tigervnc-1.3.1-3.el7.x86_64.rpm
```



- 连接VNC服务器

在图形界面用客户端软件连接VNC服务器，进行登录操作：选择图形界面上的【应用程序】->【互联网】->【TigerVNC Viewer】，打开软件界面。



- 使用vncviewer命令连接到VNC服务器。

命令语法:

vncviewer [选项] [主机][:虚拟桌面号码]

vncviewer [选项] [主机][:端口]

vncviewer [选项]

[root@rhel ~]#vncviewer 192.168.0.2:1

或

[root@rhel ~]#vncviewer 192.168.0.2:5901

Windows客户端连接

- Windows系统下的VNC客户端软件有很多，这里主要讲解VNC Viewer软件，该软件是一款优秀的远程控制工具软件，远程控制能力强大，高效实用。



- 下载vncviewer，具体操作与Linux下的类似。

