

第 3 章

Nginx 的安装

学习目标

- 熟悉 Linux 服务器环境的搭建；
- 掌握 Nginx 的安装和基本使用；
- 掌握 Nginx 运行环境的基本部署。

在使用 Nginx 之前,首先需要搭建服务器环境,并安装一些必备的软件。本章将在 Windows 系统中通过 VMware 虚拟机来部署网络环境并搭建 Linux 服务器,同时考虑到 Linux 系统的复杂多样,下面基于 CentOS 最小化安装的方式进行讲解,以便读者更好地学习 Nginx 的安装与使用。

3.1 Linux 服务器搭建

Linux 系统以免费、可靠、安全、稳定等优点,在服务器领域广泛应用。在互联网环境下,服务器要求具有很强的稳定性,能够长时间持续稳定工作,因此搭建一个稳定、可靠、安全的服务器非常重要。本节将针对 Linux 服务器的搭建进行详细讲解。

3.1.1 最小化安装 CentOS

前面已经讲解如何利用 VMware 虚拟机快捷安装 Linux 系统。在实际应用时,对于大部分中小企业而言,服务器的采购和维护成本非常昂贵,因此会由专业的运维人员对服务器进行优化,从而最大化地利用硬件资源。所以 Linux 系统的安装和配置对于搭建服务器而言非常重要。

在安装 CentOS 作为服务器来使用时,推荐读者使用最小化安装的方式进行学习。最小化安装是指在安装系统时只预装最基本的组件,然后由用户自行安装其他软件。通过这种方式,可以尽可能地节省磁盘和内存空间,防止不必要的资源浪费。

1. 创建虚拟机

使用 VMware 虚拟机可以很方便地部署实验环境,用户只需要一台计算机,就能够运行多台虚拟的服务器,并支持多种网络环境。

由于采用最小安装的方式,读者既可以使用前面讲过的 CentOS-6.8-x86_64-bin-DVD1.iso 镜像进行安装,也可以下载迷你版镜像 CentOS-6.8-x86_64-minimal.iso 进行安装。

在使用 VMware 创建虚拟机时,可以对硬件参数进行配置。本书配置的硬件参数如表 3-1 所示。

表 3-1 虚拟机的硬件参数

硬 件	说 明
处理器	请根据物理机的核心数设置虚拟机的总核心数(数量越多处理速度越快)
内存	最小支持 512MB(但不支持图形安装界面),推荐 1024MB
网络连接	使用 NAT(网络地址转换)
磁盘空间	推荐 10GB 或以上
USB 控制器	不需要此虚拟设备
声卡	不需要此虚拟设备
打印机	不需要此虚拟设备

在表 3-1 中,当内存设置为 512MB 时,CentOS 将不会启动图形化安装界面,并且只能进行最小化安装。接下来将具体的操作步骤进行图文演示,具体步骤如下。

- (1) 启动 VMware,执行“文件”→“新建虚拟机”命令。
- (2) 在弹出的“新建虚拟机向导”页面中选择“典型(推荐)”,然后单击“下一步”按钮。
- (3) 在“安装客户机操作系统”页面,选择“稍后安装操作系统”单选按钮,如图 3-1 所示。选择后单击“下一步”按钮。



图 3-1 安装客户机操作系统

需要注意的是,如果此处选择“安装程序光盘映像文件(iso)”单选按钮并浏览到 CentOS 的光盘镜像文件,VMware 会自动检测镜像中的操作系统版本并启动快捷安装功能。快捷安装方式虽然可以很轻松地完成系统的安装,但是并不利于学习更深度手动定

制安装,因此推荐读者选择“稍后安装操作系统”单选按钮。

(4) 在选择客户机操作系统时,选择 Linux 系统,版本选择 CentOS,然后单击“下一步”按钮,如图 3-2 所示。



图 3-2 选择操作系统版本

值得一提的是,由于不同的操作系统对于硬件的配置要求不同,因此 VMware 虚拟机对于支持的操作系统提供了推荐的默认配置。当我们正确选择操作系统版本后,VMware 就会根据系统版本来决定硬件的推荐配置。

(5) 配置虚拟机的名称和保存位置,读者可根据物理机的实际情况进行选择,然后单击“下一步”按钮,如图 3-3 所示。

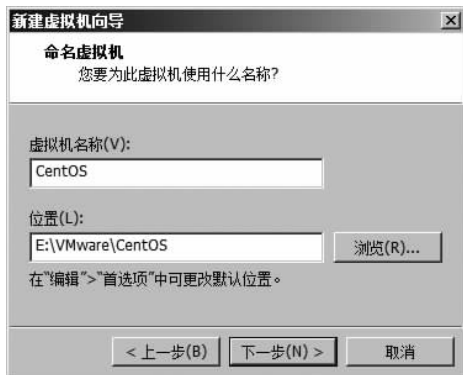


图 3-3 配置虚拟机的名称和保存位置

(6) 指定磁盘容量大小,此处配置 10GB 的空间即可,如果物理机支持大于 4GB 以上的单文件,则选择“将虚拟磁盘存储为单个文件”。在完成此处的配置后单击“下一步”按钮。

(7) 在“已准备好创建虚拟机”页面进行硬件定制,如图 3-4 所示。

(8) 单击“自定义硬件”按钮,弹出如图 3-5 所示的“硬件”对话框。

在图 3-5 中,由于服务器用不到声卡、USB 控制器、打印机这些设备,因此可以将这些设备移除。对于虚拟机的内存和处理器,根据物理机的硬件能力来分配即可。

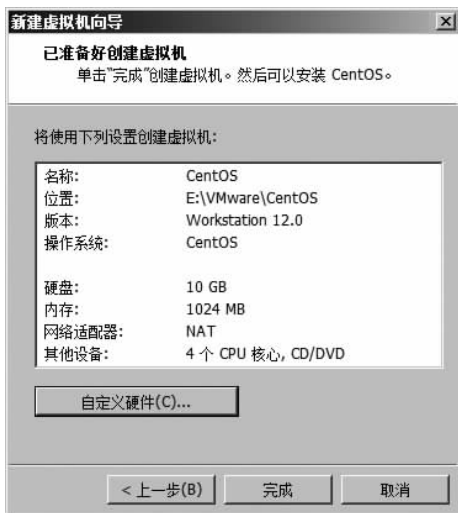


图 3-4 已准备好创建虚拟机

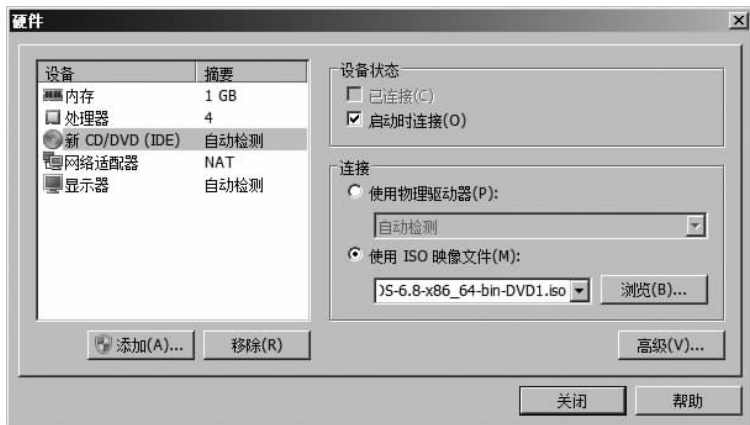


图 3-5 自定义硬件

接着,在左侧设备栏目中选择“新 CD/DVD(IDE)”设备,然后在右侧“连接”分组框中选择“使用 ISO 映像文件”单选按钮,单击“浏览”按钮找到 CentOS 镜像文件,将 CentOS 的光盘镜像装载到虚拟机的光驱设备中,从而在开启虚拟机之后安装系统。

(9) 在硬件定制完成之后,单击“关闭”按钮,然后在“已准备好创建虚拟机”页面单击“完成”按钮即可完成虚拟机的创建。

2. 安装系统

关于 CentOS 的安装步骤,前面是通过 VMware 的快捷安装功能来完成的。而本次在创建虚拟机时并没有开启快捷安装功能,因此在启动虚拟机后会读取光盘镜像进入安装向导。关于最小化安装方式在安装过程中的必要配置如表 3-2 所示。

表 3-2 CentOS 安装说明

安装步骤	配置说明
启动菜单	选择 Install or upgrade on existing system(安装或升级系统)
发现光盘(Disc Found)	选择 Skip 跳过光盘检查
安装语言选择	选择简体中文或 English(根据读者的偏好)
键盘类型选择	选择“美国英语式”键盘
您的安装将使用哪种设备	选择“基本存储设备”
以下设备中可能包含数据	维持默认值并单击“是,忽略所有数据”按钮
请为这台计算机命名	维持默认值并单击“下一步”按钮
请选择离本地时区最近的城市	城市选择“亚洲/上海”,并取消选中“系统时钟使用 UTC 时间”。然后单击“下一步”按钮
请为根用户输入一个密码	即配置 root 用户的密码,可以输入 123456 并单击“下一步”按钮,然后选择“无论如何都使用”。
您要进行哪种类型的安装	选择“使用所有空间”并单击“下一步”按钮
将存储配置写入磁盘	选择“将修改写入磁盘”
选择 CentOS 安装的软件	选择 Minimal(最小安装)并单击“下一步”按钮

在按照表 3-2 中的说明进行操作以后,系统就会开始进行安装,等待安装完成即可。在安装完成后,单击“重新引导”按钮重新启动系统。

在完成安装并重启之后,就会出现 CentOS 的登录界面,如图 3-6 所示。在登录界面中输入用户名 root 和密码 123456 之后就可以登录系统。



图 3-6 CentOS 登录界面



多学一招：查看服务器内存和磁盘空间

在使用最小安装方式安装完成系统后,可以查看当前服务器的内存和磁盘空间,具体方法如下。

(1) 查看内存空间。

使用 free 命令可以查看服务器的内存空间,选项 -m 表示以 MB(兆字节)的数据存储单位进行显示。执行结果如下所示:

```
[root@localhost ~]#free -m
              total        used         free       shared    buffers     cached
Mem:           995          168          827           0           6           41
-/+buffers/cache:    121          874
Swap:         1023           0         1023
```

在以上输出结果中,Mem 表示系统的物理内存,total 表示内存的总大小(995MB),used 表示已经使用的空间(168MB),free 表示可用空间(827MB)。关于其他内容读者可参考 Linux 帮助手册进行学习,这里就不再详细解释。

(2) 查看磁盘空间。

使用 df 命令可以查看服务器的磁盘空间,选项 -lh 表示利用方便阅读的数据存储单位显示本地文件系统。执行结果如下所示:

```
[root@localhost ~]#df -lh
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/VolGroup-lv_root
                8.3G  646M  7.2G   9%  /
tmpfs           498M    0  498M   0%  /dev/shm
/dev/sda1       477M   28M  425M   7%  /boot
```

在以上输出结果中,Filesystem 是文件系统,Size 表示该分区的总大小,Used 表示已经使用的空间,Avail 表示可用空间,Use% 表示已经使用的百分比,Mounted on 表示挂载路径。可以看出,系统共分为 3 个文件系统,其中 /dev/mapper/VolGroup-lv_root 是挂载到根目录的文件系统,总大小为 8.3GB,已经使用的空间为 646MB。

通过上述命令可以看出,最小安装后的 CentOS 占用的内存和磁盘空间都非常小。用户可以根据自己的需要来手动安装软件和配置系统,避免因为大量预装软件而导致内存、磁盘等硬件资源的浪费。

3.1.2 网络配置

Linux 服务器可以提供的服务包括 Web、FTP、DNS、DHCP、数据库和邮箱等多种类型,但是这些服务都离不开网络环境。下面将针对 VMware 虚拟机和 Linux 服务器的网络配置进行详细讲解。

1. VMware 网络配置

VMware 虚拟机提供了虚拟网络功能,可以很方便地进行网络环境部署。在程序的菜单栏中执行“编辑”→“虚拟网络配置”命令,打开如图 3-7 所示的对话框,查看网络配置。

从图 3-7 中可以看出,VMware 提供了桥接、NAT(网络地址转换)和仅主机模式,分别对应的名称为 VMnet0、VMnet8 和 VMnet1。关于这 3 种模式的具体介绍如下。

1) 桥接模式

当虚拟机的网络处于桥接模式时,相当于这台虚拟机与物理机同时连接到一个局域网,这两台机器的 IP 地址将处于同一个网段中。以目前家庭普遍使用的宽带上网环境为例,其网络结构如图 3-8 所示。

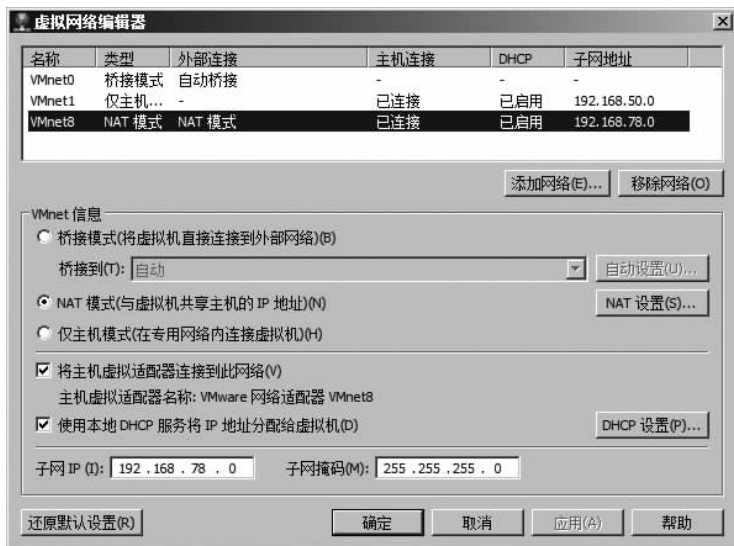


图 3-7 VMware 虚拟网络编辑器

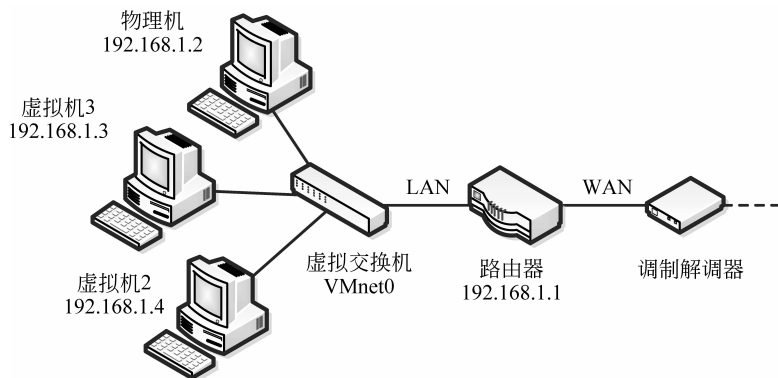


图 3-8 VMnet0 虚拟网络

从图 3-8 可以看出,两台虚拟机和一台物理机同时处于一个局域网内(VMware 支持同时运行多个虚拟机),如果路由器已经接入网络,则图 3-8 中的 3 台计算机都可以访问外部网络。

2) NAT 模式

NAT 是 VMware 虚拟机中默认使用的模式,其最大的优势是虚拟机接入网络非常简单,只要物理机可以访问网络,虚拟机就可以访问网络。其网络结构如图 3-9 所示。

从图 3-9 可以看出,物理机网卡和 VMnet8 虚拟网络中的 NAT(网络地址转换)网关共享了同一个 IP 地址 192.168.1.2,因此只要物理机连上网,虚拟机就能上网。为了让物理机和虚拟机能够直接互访,需要在物理机中增加一个虚拟网卡接入 VMnet8 虚拟交换机中。

3) 仅主机模式

仅主机模式与 NAT 模式相似,但是在该网络中没有虚拟 NAT,因此只有物理机能上网而虚拟机无法上网,只能在 VMnet1 虚拟网内相互访问。其网络结构如图 3-10 所示。

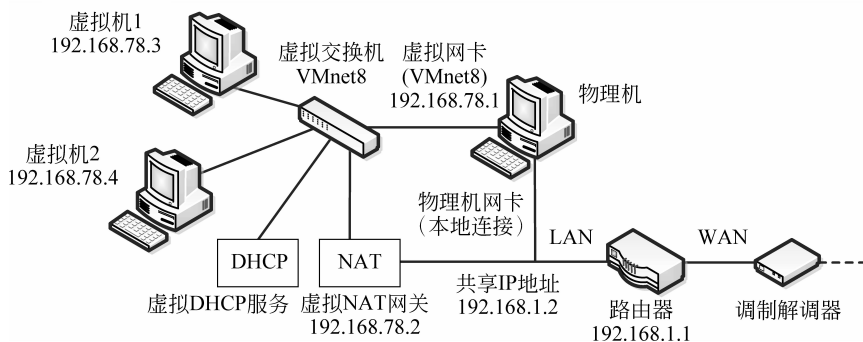


图 3-9 VMnet8 虚拟网络

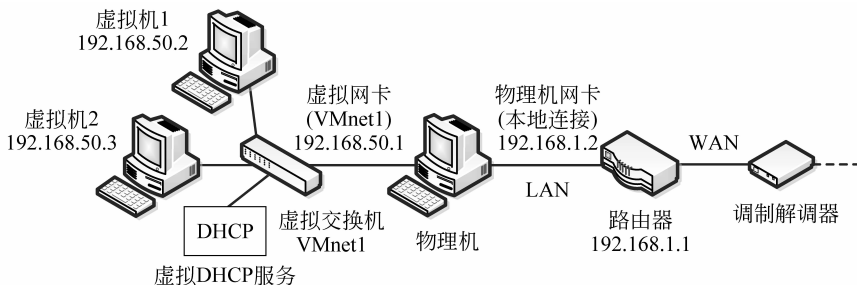


图 3-10 VMnet1 虚拟网络

由于 VMnet8 和 VMnet1 两种虚拟网络都需要虚拟网卡实现物理机和虚拟机的互访，因此在 VMware 安装时就自动安装了这两个虚拟网卡。在物理机(Windows 系统)中打开命令提示符，然后输入命令 ipconfig 查看网卡信息，从这些信息中可以找到 VMnet8 和 VMnet1 虚拟网卡，如图 3-11 所示。

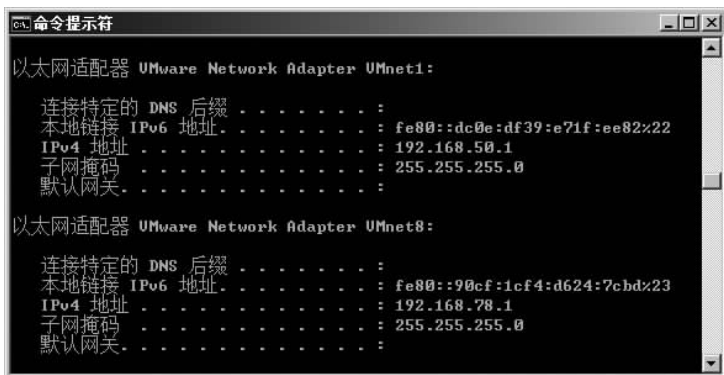


图 3-11 查看 VMware 虚拟网卡

从图 3-11 中可以看出，VMnet1 的 IP 地址为 192.168.50.1，VMnet8 的 IP 地址为 192.168.78.1。这两个 IP 地址是根据 VMware 虚拟网络编辑器中的子网 IP 来自动设置的，如果更改了子网 IP，则这两个网卡的 IP 地址会由 VMware 自动更新。

2. 更改网络模式

在 VMware 中,桥接、NAT 和仅主机 3 种模式是共存的,但是一个虚拟机只能选择一种模式来使用。在 VMware 的菜单栏中执行“虚拟机”→“设置”命令,在弹出的“虚拟机设置”对话框中选择“网络适配器”,可以查看或更改虚拟机的网络模式,如图 3-12 所示。



图 3-12 “虚拟机设置”对话框

在图 3-12 所示的窗口中,右下方有一个“高级”按钮,单击后可以打开“网络适配器高级设置”对话框,如果需要查看或更改虚拟机网卡的 MAC 地址可以在此处完成。

3. Linux 网络配置

在了解 VMware 虚拟机的网络环境以后,接下来对 Linux 服务器进行网络配置。在 Linux 系统中,通过 `ifconfig -a` 命令可以查看所有的网卡,如图 3-13 所示。

```
[root@localhost ~]# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 08:0C:29:48:2A:8A
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

图 3-13 查看网卡

从图 3-13 中可以看出,目前系统中共有 2 个网卡,第 1 个是 eth0(即编号为 0 的以太网卡),第 2 个是 lo(即本地回环网卡)。其中 eth0 网卡用于访问外部网络,默认情况下是关闭

的;lo 网卡用于在本机内部访问,IP 地址为 127.0.0.1(即 Loopback Address,本机回送地址)。

如果使用 VMware 的 NAT 模式或仅主机模式,那么网络中的虚拟机可以通过 DHCP(动态主机配置协议)自动获取 IP 地址。但是在真实环境中,应该为所有的服务器配置静态 IP 地址,从而确保通过一个 IP 地址就能找到一台服务器。下面分别介绍如何配置动态和静态 IP 地址。

1) 动态 IP

为了使 eth0 网卡工作,可以通过 `ifup eth0` 命令临时启动网卡,也可以修改 eth0 网卡的配置文件,使该网卡自动启动。接下来切换到网卡配置文件 `ifcfg-eth0` 所在的目录:

```
[root@localhost ~]#cd /etc/sysconfig/network-scripts
```

在修改配置文件之前,为了防止配置出错,建议提前备份该配置文件 `ifcfg-eth0`:

```
[root@localhost network-scripts]#cp ifcfg-eth0 ifcfg-eth0.bak
```

然后通过 vi 编辑器修改网卡配置文件:

```
[root@localhost network-scripts]#vi ifcfg-eth0
```

在打开配置文件后,具体内容如下所示。

```
DEVICE=eth0
HWADDR=00:0C:29:48:2A:8A
TYPE=Ethernet
UUID=de5dcf98-9e30-4d0e-a578-4ddcea528ae6
ONBOOT=no
NM_CONTROLLED=yes
BOOTPROTO=dhcp
```

在上述配置中,需要重点关注的是 `ONBOOT` 和 `BOOTPROTO` 两个配置,其他配置保持默认值即可。其中 `BOOTPROTO` 用于设置获取 IP 的方式是动态还是静态的,此处默认值为 `dhcp` 表示动态获取 IP;`ONBOOT` 用于设置网卡是否自动启动,默认值为 `no`,更改为 `yes` 即可实现自动启动。

修改完成后,保存并退出编辑,然后执行重新加载网络服务的命令 `service network reload` 使配置生效。在配置生效之后,通过 `ifconfig` 命令查看 eth0 网卡的状态,如图 3-14 所示。

从图 3-14 中可以看出,eth0 网卡已经获取到 IP 地址 192.168.78.128,说明服务器已经成功连接到 NAT 网络中。如果在重新加载网络服务时报错,则可能是网卡配置文件更改有误,或 VMware 虚拟网络配置有误,按照前面讲解的内容检查并更正即可。

2) 静态 IP

静态 IP 是用户手动设置的 IP,设置后固定不变。只要将 `ifcfg-eth0` 配置文件中 `BOOTPROTO` 的值设置为 `static`,将 `IPADDR`(IP 地址)的值设置为其所在子网中正确的、无冲突的 IP 地址即可。

```
[root@localhost network-scripts]# service network reload
Shutting down interface eth0:                [ OK ]
Shutting down loopback interface:            [ OK ]
Bringing up loopback interface:              [ OK ]
Bringing up interface eth0:
Determining IP information for eth0... done.   [ OK ]

[root@localhost network-scripts]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:48:2A:8A
          inet addr:192.168.78.128  Bcast:192.168.78.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe48:2a8a/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:182 errors:0 dropped:0 overruns:0 frame:0
          TX packets:35 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:17692 (17.2 KiB)  TX bytes:3942 (3.8 KiB)
```

图 3-14 查看网卡状态

假设在 VMware 的 NAT 模式中,子网 IP 为 192.168.78.0、VMnet8 虚拟网卡 IP 为 192.168.78.1、NAT 网关 IP 为 192.168.78.2、DHCP 地址池为 192.168.78.128~192.168.78.254,则 192.168.78.3~192.168.78.127 范围内的 IP 都可以作为静态 IP 使用。

接下来打开 ifcfg-eth0 配置文件进行修改,修改后的示例如下。

```
... (此处省略了前面几行)
BOOTPROTO=static
IPADDR=192.168.78.3
NETMASK=255.255.255.0
GATEWAY=192.168.78.2
DNS1=192.168.78.2
```

上述配置将 BOOTPROTO 的值由 dhcp 修改为 static,然后增加了 IPADDR(IP 地址)、NETMASK(子网掩码)和 GATEWAY(网关)和 DNS1(首选域名服务器)。其中,若网关不设置,虚拟机只能在局域网内访问,无法访问外部网络;若 DNS 不设置,则无法解析域名。

修改配置文件后执行 service network reload 命令使配置生效即可。在配置生效后,可以通过如下操作查看当前使用的默认网关、DNS 服务器。

```
[root@localhost ~]# route | grep default
default          192.168.78.2    0.0.0.0         UG  0      0      0 eth0
[root@localhost ~]# cat /etc/resolv.conf
nameserver 192.168.78.2
```

4. 访问测试

在完成 Linux 服务器的网络配置以后,就可以进行访问测试。无论是 Windows 还是 Linux 系统,都提供了 ping 命令用于检测网络是否连通。在物理机(Windows 系统)中打开命令提示符,执行“ping 虚拟机 IP 地址”,运行结果如图 3-15 所示。

从图 3-15 中可以看出,物理机共向 IP 地址 192.168.78.3 发送了 4 次 ping 请求,4 次都是成功的,发送的数据包为 32 字节,响应时间小于 1 毫秒,TTL(生存时间值)值为 64。其中 TTL 在发送时的默认值为 64,每经过一个路由则减 1,此处显示的最终结果为 64,说明中间没有经过路由。



图 3-15 物理机 ping 虚拟机

当使用虚拟机 ping 物理机时,若物理机(Windows)的防火墙为开启的状态,是无法 ping 通的。可以临时关闭 Windows 防火墙,或者将防火墙入站规则中的“文件和打印机共享(回显请求-ICMPv4-In)”设置为“允许连接”。在解决防火墙问题后,虚拟机 ping 物理机的执行结果如图 3-16 所示。

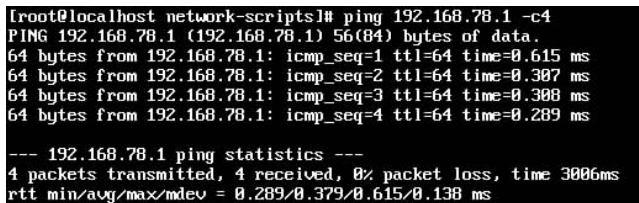


图 3-16 虚拟机 ping 物理机

从图 3-16 中可以看出,在 Linux 中使用 ping 命令时加上了参数-c4,表示 ping 的次数为 4。如果省略该参数,则会一直执行,直到按 Ctrl+C 键停止程序。

在测试局域网内的访问后,还需要测试虚拟机能否访问外网。在物理机正确接入外网的前提下,用虚拟机 ping 外部主机(如 ping baidu.com)是可以 ping 通的。如果在正确配置后虚拟机仍然无法访问网络,则有可能是物理机中安装了多个网卡,VMware 会自动使用优先级较高的网卡(无论该网卡是否接入外网),更改网卡优先级或者禁用这些网卡可以解决问题。在默认情况下,新安装的网卡优先级高于原有网卡,但 VMnet1、VMnet8 两个虚拟网卡的优先级低于本地连接网卡,这样可以避免影响用户正常使用网络。

3.1.3 远程终端访问

当服务器部署好以后,除了直接在服务器上操作,还可以通过网络进行远程连接访问。CentOS 6.8 默认支持 SSH(Secure Shell,安全 Shell 协议),该协议通过高强度的加密算法提高了数据在网络传输中的安全性,防止中间人攻击(Man-in-the-Middle Attack,一种黑客常用的攻击手段)。本节将针对如何通过 SSH 远程访问服务器进行详细讲解。

1. SSH 客户端

目前支持 SSH 的客户端有很多,在 Windows 中可以使用 Xshell、SecureCRT 等软件,通过这类软件可以在 Windows 系统上远程控制 Linux 系统。

本书以 Xshell 为例,该软件提供了家庭、学校授权版本,可以免费使用。在 Xshell 的官方网站 <http://www.netsarang.com> 可以找到软件的下载地址。Xshell 的安装非常简单,按照提示进行操作即可,下面开始分步骤讲解 Xshell 如何使用。

(1) 安装完成以后,打开 Xshell,会自动弹出一个“会话”对话框,如图 3-17 所示。如果关闭了此对话框,也可通过在菜单栏中执行“文件”→“打开”命令再次打开此对话框。

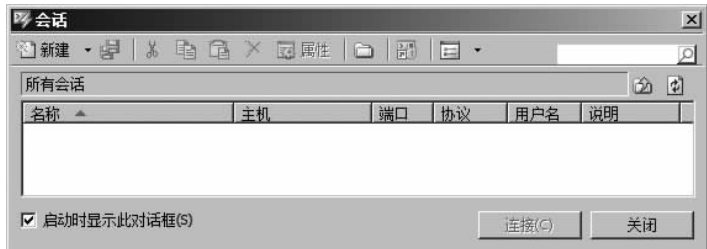


图 3-17 新建会话

(2) 在图 3-17 所示的对话框中,单击工具栏中的“新建”按钮,或在 Xshell 窗口的菜单栏执行“文件”→“新建”命令,会弹出一个“新建会话属性”的对话框,如图 3-18 所示。



图 3-18 新建会话属性

(3) 在“常规”分组框中输入“名称”和“主机”,其中名称可以随意填写,主机填写服务器的 IP 地址。“协议”选择默认的 SSH 即可,“端口号”保持默认值 22。

(4) 在左侧的“类别”列表中选择“用户身份验证”,然后输入 Linux 服务器的用户名 (root)和密码(123456),如图 3-19 所示。此处输入的用户名和密码会保存到客户端,用于快捷登录。如果考虑安全性,此处可以留空,在每次登录时输入用户名和密码。

(5) 在“类别”列表中选择“终端”,将“终端类型”修改为 linux,如图 3-20 所示。需要注意的是,此处也可以使用默认值 xterm,但键盘中的 Num Lock 数字小键盘区的映射会出现问题。

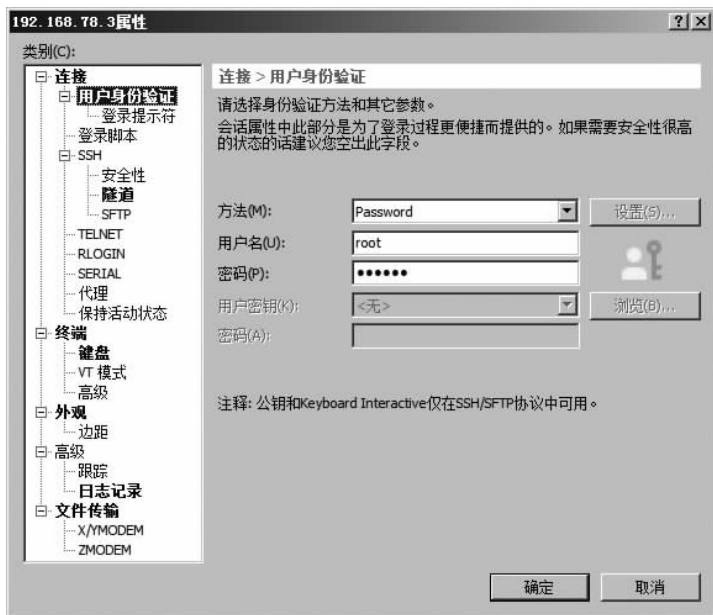


图 3-19 输入用户名和密码



图 3-20 选择终端类型

(6) 设置完成后,单击“确定”按钮保存会话并返回原来的“会话”对话框,如图 3-21 所示。

(7) 选中刚才保存的 192.168.78.3 会话并单击“连接”按钮,即可远程连接到服务器。在连接并登录成功后效果如图 3-22 所示。

值得一提的是,在图 3-22 所示的窗口中,工具栏中有一个“新建文件传输”按钮,通过该

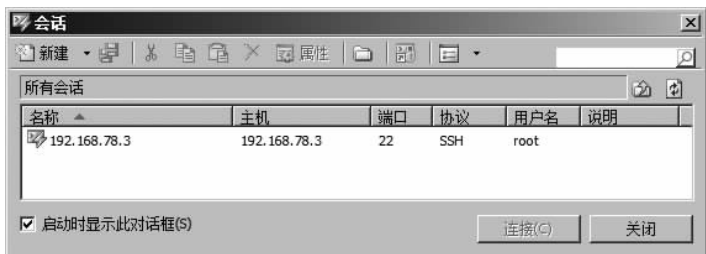


图 3-21 查看保存的会话



图 3-22 远程登录

按钮可以打开 Xftp 远程文件管理工具。由于 Xftp 需要额外安装,如果没有安装,程序会提示到官方网站中进行下载。安装 Xftp 以后可以用图形化的方式远程管理服务器中的文件。

2. SFTP 远程文件管理

SFTP(Secure File Transfer Protocol,安全文件传送协议)是一种安全的远程文件传输协议,和 SSH 协议类似,在传输过程中会进行加密。前面提到的 Xftp 就是一种 SFTP 的客户端,可以与 Xshell 配合一起使用。接下来,本书以 Xftp 的使用为例进行讲解。

(1) 安装 Xftp 后,在 Xshell 远程服务器登录成功的状态下单击工具栏中的“新建文件传输”按钮可以自动打开 Xftp 并登录服务器,如图 3-23 所示。



图 3-23 Xftp 远程文件管理

在图 3-23 所示的窗口中,左侧为客户端 Windows 系统的文件列表,右侧为 Linux 系统的文件列表。通过这个窗口,可以实现文件的上传、下载、复制、剪切、删除、修改文件权限和

属性等操作,支持文件拖曳功能,使用非常方便。

(2) Xftp 支持为远程服务器中的文件关联文本编辑器,默认关联的是 Windows 记事本。本书以开源软件 Notepad++ 编辑器为例,在 Xftp 窗口中执行菜单栏中的“工具”→“选项”操作,切换到“高级”选项卡,将可执行文件 notepad++.exe 的路径添加到“编辑器路径”中,如图 3-24 所示。

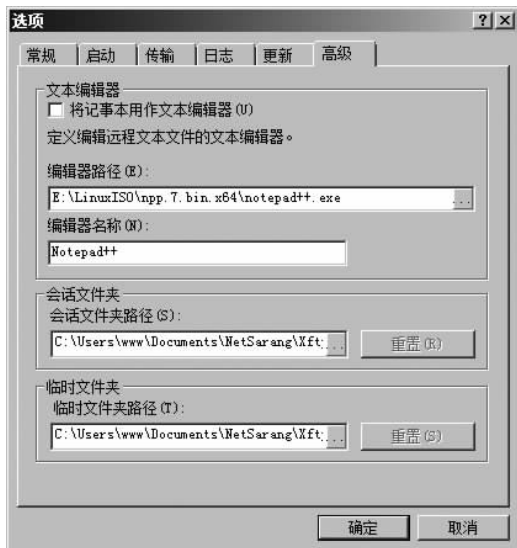


图 3-24 关联文本编辑器

(3) 关联之后,在远程服务器的文件列表中选中一个文件右击,就会出现“以 Notepad++ 编辑”的一个菜单项,单击后即可调用 Notepad++ 编辑器自动打开文件。

值得一提的是,在使用 Notepad++ 创建 Linux 系统中的文件时,推荐将文件的编码格式设置为“UTF8 无 BOM 格式编码”,并且将换行符设置为 UNIX 格式,这样做的目的是保证该文件能够被 Linux 系统中的程序正确识别。

3.1.4 安装必备软件

在通过最小化方式安装的 CentOS 6.8 系统中,并没有安装 gcc、gcc-c++ 编译器,而后续的课程中用到的大部分软件和模块都需要 gcc 和 gcc-c++。因此在学习 Nginx 之前有必要先安装这些必备的软件。

另外在学习的过程中,推荐读者安装 wget 工具,wget 可以从网络上自动下载文件,支持 HTTP、HTTPS 和 FTP 3 种常见协议的下载,并支持 HTTP 代理。表 3-3 列举了这些必备的软件和说明。

表 3-3 安装必备软件

软件名称	说 明
gcc	用来编译 C 程序
gcc-c++	用来编译 C++ 程序
wget	用来从网络上下载文件

在 CentOS 系统中,可以通过 yum 方式联网自动下载和安装这些软件,具体命令如下:

```
[root@localhost ~]#yum -y install gcc gcc-c++wget
```

在上述命令中,install 表示安装软件;-y 表示在整个操作过程中全部执行默认的 yes 操作,如果省略该选项,则每一步都会提示用户选择 yes 或 no。

安装完成后,效果如图 3-25 所示。

```
Installed:
gcc.x86_64 0:4.4.7-17.el6          gcc-c++.x86_64 0:4.4.7-17.el6          wget.x86_64 0:1.12-8.el6

Dependency Installed:
clog-ppl.x86_64 0:0.15.7-1.2.el6          cpp.x86_64 0:4.4.7-17.el6
glibc-devel.x86_64 0:2.12-1.192.el6       glibc-headers.x86_64 0:2.12-1.192.el6
kernel-headers.x86_64 0:2.6.32-642.4.2.el6  libgomp.x86_64 0:4.4.7-17.el6
libstdc++.devel.x86_64 0:4.4.7-17.el6     mpfr.x86_64 0:2.4.1-6.el6
ppl.x86_64 0:0.10.2-11.el6

Dependency Updated:
glibc.x86_64 0:2.12-1.192.el6          glibc-common.x86_64 0:2.12-1.192.el6  libgcc.x86_64 0:4.4.7-17.el6
libstdc++.x86_64 0:4.4.7-17.el6       tzdata.noarch 0:2016f-1.el6

Complete!
```

图 3-25 yum 方式安装软件

wget 工具的使用非常简单,通过“wget 下载地址”命令即可将文件下载到当前目录下,并以下载地址中包含的文件名来保存。示例命令如下。

```
[root@localhost ~]#wget http://nginx.org/download/nginx-1.10.1.tar.gz
```

当命令执行后,文件就会被下载到当前目录下,并以 nginx-1.10.1.tar.gz 文件名保存。

脚下留心：时间和时区问题

在 Linux 系统安装完成后,可以通过 date 命令查看当前服务器的时间。在默认情况下,操作系统的时间是读取的计算机主板 BIOS 中的时间,而 VMware 虚拟机能够自动设置 BIOS 时间为物理机时间,因此在安装后,CentOS 的系统时间和物理机 Windows 中显示的时间是相同的。

需要注意的是,在安装系统时,有一个“系统时钟使用 UTC 时间”复选框,在前面讲解的步骤中是取消选中的,这是因为物理机 Windows 系统并没有使用 UTC 时间,如果选中此项,则会因为时区不同导致虚拟机显示的系统时间与物理机不一致。

VMware 虚拟机提供了方便的快照功能,但是在还原到以前的快照时,系统时间并不会自动更新成当前时间,因此推荐读者在还原快照后通过 NTP 服务器联网同步最新时间,具体步骤如下。

(1) 通过 yum 安装 ntpdate 工具。

在最小安装的 CentOS 系统中没有 ntpdate,可以使用 yum 进行安装,具体命令如下。

```
[root@localhost ~]#yum -y install ntpdate
```

(2) 更新系统时间。

目前网络中有许多 NTP 服务器可以使用,读者可以通过百度等搜索引擎查找。下面以 ntp.org 提供的中国节点的服务器为例,具体命令和执行结果如下。

```
[root@localhost ~]#ntpdate cn.pool.ntp.org
28 Sep 15:57:58 ntpdate[1215]: step time server 115.28.122.198 offset 1306.825 sec
[root@localhost ~]#date
Wed Sep 28 15:58:15 CST 2016
```

从上述结果可以看出,已经成功连接到 cn.pool.ntp.org 服务器获取到最新的系统时间。然后通过 date 命令查询当前时间,检查时间是否正确。



多学一招: Linux 系统软件安装方式

Linux 系统中常见的软件安装方式有两种,一种是源码方式安装,另一种是利用 RPM 包和软件包管理器进行安装。

- 所谓源码方式安装,就是将“源码文件”编译成“二进制文件”,并复制到“系统指定目录”的过程。该安装方式的优点是软件运行速度和效率非常高,并可以灵活进行配置。缺点是安装过程相对软件包管理器方式烦琐一些。
- 软件包管理器是一些发行版 Linux 中提供的辅助工具。例如,在 Red Hat、CentOS、Fedora 系统中可以使用 yum(全称为 Yellow dog Updater, Modified)管理软件;在 Debain、Ubuntu 系统中可以使用 APT(Advanced Packaging Tools)管理软件。
- CentOS 系统支持 RPM 包,利用 yum 工具可以从指定的服务器自动下载 RPM 包,并且自动处理依赖关系,实现“一键安装”的效果。该安装方式的优点是软件安装非常方便、快速、智能。缺点是软件版本可能会低于官网提供的版本,并且 yum 服务器并不是囊括了所有软件,对于没有的软件只能手动下载安装。

3.2 Linux 环境下安装 Nginx

在上一节中,完成了 Linux 服务器环境的搭建,并安装了必备的软件。本节将开始讲解如何在 Linux 环境中安装 Nginx,认识 Nginx 的目录结构,以及启动、停止或重启 Nginx 服务。完成 Nginx 的部署后,Web 服务器就搭建完成,在 Windows 系统中可以通过浏览器访问 Web 服务器中的网页。

3.2.1 获取 Nginx

Nginx 在官方网站提供了软件下载。目前 Nginx 发布了 3 种类型的版本,分别为 Mainline version(开发版)、Stable version(稳定版)和 Legacy versions(早期版本),每种类型的版本中又提供了 Linux 版本和 Windows 版本,如图 3-26 所示。

本书基于稳定版 1.10.1 进行讲解,单击图 3-26 所示的 Stable version 标题下第 2 列的 nginx-1.10.1 可以下载 Nginx 的源码包,下载后上传到 Linux 服务器上即可。

在操作时,读者可以在物理机 Windows 系统上进行下载,然后通过 Xftp 工具将文件上传到 Linux 服务器,也可以从浏览器上复制下载文件的 URL 地址后在服务器中使用 wget 命令进行下载。

下载后的 Nginx 文件是一个 .tar.gz 格式的压缩包,在 Linux 系统中使用 tar -zxvf 命令进行解压缩,其中选项 z 表示压缩格式为 gzip,x 表示解压缩,v 表示显示解压过程,f 表示对

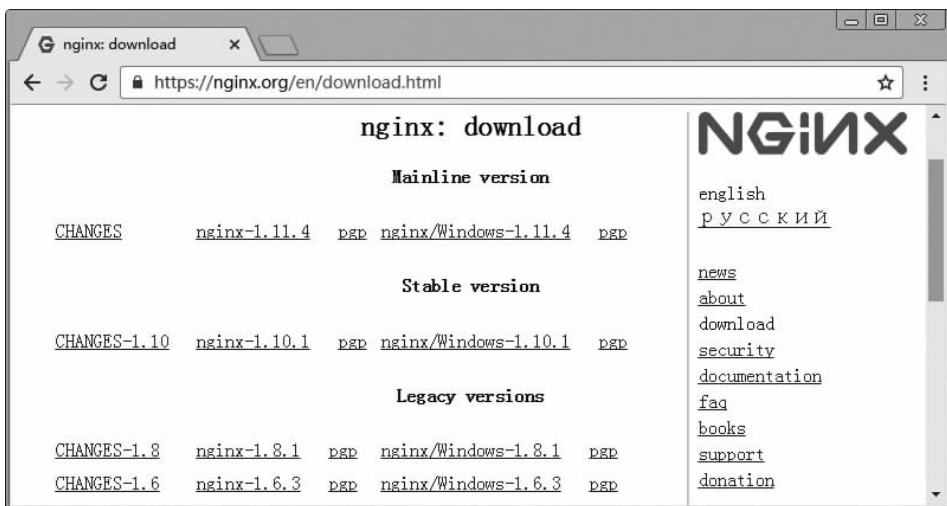


图 3-26 获取 Nginx

文件进行操作。

```
[root@localhost ~]#tar -zxvf nginx-1.10.1.tar.gz
```

上述命令执行后,文件就被解压缩到当前目录下的 nginx-1.10.1 目录中。使用 cd 命令切换到该目录,然后使用 ls 命令查看该目录下的文件,具体如下。

```
[root@localhost ~]#cd nginx-1.10.1
[root@localhost nginx-1.10.1]#ls
auto      CHANGES.ru  configure  html        man        src
CHANGES  conf         contrib    LICENSE     README
```

关于上述目录结构的具体介绍如下。

- src 目录: 存放 Nginx 的源代码。
- man 目录: 存放 Nginx 的帮助文档。
- html 目录: 存放默认网站文件。
- contrib 目录: 存放其他机构或组织贡献的文档资料。
- conf 目录: 存放 Nginx 服务器的配置文件。
- auto 目录: 存放大量的脚本文件,和 configure 脚本程序相关。
- configure 文件: Nginx 自动安装脚本,用于检查环境,生成编译代码需要的 makefile 文件。
- CHANGES、CHANGES.ru、LICENSE 和 README 都是 Nginx 服务器的相关文档资料。

3.2.2 编译安装 Nginx

1. 安装依赖包

由于 Nginx 中的功能是模块化的,而模块又依赖于一些软件包(如 pcre 库、zlib 库和

openssl 库)才能使用。因此,在安装 Nginx 前,需要完成 Nginx 模块依赖的软件包安装。关于这些软件包的说明如表 3-4 所示。

表 3-4 Nginx 依赖包的说明

软件包	说明
pcre-devel	为 Nginx 模块(如 rewrite)提供正则表达式库
zlib-devel	为 Nginx 模块(如 gzip)提供数据压缩用的函数库
openssl-devel	为 Nginx 模块(如 ssl)提供密码算法、证书以及 SSL 协议等功能

接下来,通过 yum 方式安装 Nginx 相关依赖包,具体命令如下。

```
[root@localhost ~]#yum -y install pcre-devel openssl-devel
```

由于 openssl-devel 依赖于 zlib-devel,在通过 yum 进行安装时会自动解决依赖,因此上述命令中省略了 zlib-devel。安装完成后,如图 3-27 所示。

```
Installed:
  openssl-devel.x86_64 0:1.0.1e-48.el6_8.3                pcre-devel.x86_64 0:7.8-7.el6

Dependency Installed:
  keyutils-libs-devel.x86_64 0:1.4-5.el6                  krb5-devel.x86_64 0:1.10.3-57.el6
  libcom_err-devel.x86_64 0:1.41.12-22.el6                libsasl-devel.x86_64 0:2.0.94-7.el6
  libsepol-devel.x86_64 0:2.0.41-4.el6                    zlib-devel.x86_64 0:1.2.3-29.el6

Dependency Updated:
  openssl.x86_64 0:1.0.1e-48.el6_8.3

Complete!
[root@localhost ~]#
```

图 3-27 安装依赖包

2. Nginx 的编译安装

Nginx 安装的准备工作完成后,接下来使用源码方式完成 Nginx 的安装,具体步骤如下。

(1) 切换到 nginx 解压目录。

```
[root@localhost ~]#cd nginx-1.10.1
```

(2) 配置 Nginx 的编译选项,指定 Nginx 的安装目录。

```
[root@localhost nginx-1.10.1]#./configure \
--prefix=/usr/local/nginx \
--with-http_ssl_module
```

在上述命令中,源码安装的第一步./configure 用于对即将安装的软件进行配置,检查当前的环境是否满足安装软件(Nginx)的依赖关系。其中,--prefix 选项用于设置 Nginx 的安装目录,默认值是/usr/local/nginx,因此也可以省略此选项或指定到其他位置;--with-http_ssl_module 选项用于设置在 Nginx 中允许使用 http_ssl_module 模块的相关功能。

值得一提的是,第 1、2 行末尾的“\”表示当前命令没有结束需要换到下一行书写,直到