

KVM网络管理

虚拟机是需要连网的,否则就失去了意义。虚拟机连网有很多方法,常见的有 NAT 方式和桥接方式两种,每一种的意义又不一样,因此,本章重点介绍这两种方式的原理以 及相应的配置。

▶ 学习目标:

- 掌握通过 NAT 方式与外界通信。
- 掌握通过桥接方式与外界通信。



5.1 NAT 网络

5.1.1 NAT 网络的原理

NAT 即为网络地址转换,在网络里主要是解决内网访问外网的通信问题的,通常 设置了 NAT 之后,内网可以访问外网,但是外网无法访问内网。如图 5-1 所示,虚拟机 连接在宿主机上,进行 NAT 转换之后,才能访问物理网络,而物理网络是不能访问虚 拟机的。



KVM 默认的连网方式是 NAT。当安装好 KVM 虚拟机后,就会在宿主机上安装一个网桥 virbr0,如图 5-2 所示,此网桥会把虚拟机都连接起来,处在同一个网段,并且 KVM 会修改 iptables 规则,让连接到此网桥的虚拟机访问外网时做一个网络地址转换。



图 5-2 NAT 连网的宿主机内部结构

5.1.2 NAT 网络的图形配置方法

NAT 网络的配置有两种方法,一是通过图形的方式来进行配置。二是通过配置文件的方式来进行配置。图形配置方法直观,下面介绍一下其具体的操作步骤:

(1) 打开配置网络的界面,如图 5-3 所示,选择虚拟机管理器 Edit 菜单下的 Connection Details 选项,将弹出如图 5-4 所示的对话框,选择里面的 Virtual Networks 选项卡来对虚拟网络进行配置。

📾 Virtual Machine Manager			-		×
File Edit View Help					
Connection Details Uirtual Machine Details	~				
Nam Delete		*	CPU u	isage	
▼ QE Preferences					_
Shutoff			~		_

图 5-3 选择 Connection Details 选项

(2) 在 Virtual Networks 选项卡中, 先删除 KVM 自己创建的 NAT 网络 default, 结果如图 5-5 所示, 然后再创建自己的网络。

(3) 在图 5-5 中,单击左下角的"+"按钮,新建一个虚拟网络,如图 5-6 所示,在 Network Name 的文本框中输入 WYLNAT。



图 5-4 选择 Virtual Networks 选项卡

Kā.	QEMU/KVM Conr	nection Details			-		×
Fi	le						
	Overview	Virtual Networks	Storage	Network Interfaces			
	+ ▷ ● €	2	No vi	tual network selected.		A	pply

图 5-5 删除默认的网络 default

(4) 在图 5-6 中,单击 Forward 按钮,将弹出如图 5-7 所示的对话框,在图中输入虚拟 网络的网段地址 192.168.200.0/24,并且启动 DHCP,指定 DHCP 地址池的范围,单击 Forward 按钮,弹出如图 5-8 所示的对话框,不启用 IPv6 的网络。



📾 Create a new virtual network	х
Create virtual network	
Choose a name for your virtual network:	
Network Name: WYLNAT	
SExample: network1	
Cancel Back Forward	

图 5-6 输入网络名字

Ka Create	a new virtual network	Х
	Create virtual network Step 2 of 4	
Choose I	Pv4 address space for the virtual network:	
🗹 Enab	e IPv4 network address space definition	
Networ	k: 192.168.200.0/24	
୍ଷ Hir IPv 192	at: The network should be chosen from one of the 4 private address ranges. eg 10.0.0.0/8 or 2.168.0.0/16	
Gat	eway: 192.168.200.1	
	Type: Private	
🕑 Ena	able DHCPv4	
Start:	192.168.200.128	
End:	192.168.200.254	
🔲 Ena	able Static Route Definition	
	Cancel Back Forward	

图 5-7 配置 IPv4 网络

(5) 在图 5-8 中,单击 Forward 按钮,弹出如图 5-9 所示的对话框,选择 Forwarding to physical network,目标网卡为 ens33,模式为 NAT,也就意味着虚拟机如果选择 WYLNAT 网络后,将通过 ens33 网卡进行地址转换后再转发给外网。单击 Finish 按钮, 完成网络的配置。

(6) 在如图 5-9 所示的对话框中,单击 Finish 按钮,可以看到一个新生成的虚拟网络 WYLNAT,如图 5-10 所示。



图 5-8 配置 IPv6 网络

📾 Create a new virtual network	×
Create virtual network Step 4 of 4	
Connected to a physical network :	
 Isolated virtual network 	
\odot Forwarding to physical network	
Destination: Physical device ens33 🔹	
Mode: NAT 🕶	
Enable IPv6 internal routing/networking If an IPv6 network address is not specified, this will enable IPv6 internal routing between virtual machines. By default, IPv4 internal routing is enabled.	
DNS Domain Name: WYLNAT	
Cancel Back Finish]

图 5-9 选择转换的物理网络

(7) 虚拟机如果要使用 WYLNAT 网络,可以打开虚拟机的详情页面,如图 5-11 所示,选择网卡,在虚拟网络接口页面中,选择网络源 WYLNAT。

(8) 启动虚拟机 vm01-clone,查看网卡的情况,发现 eth0 获取到 WYLNAT 网络提供的 IP 地址 192.168.200.195/24,如图 5-12 所示。

(9) 从宿主机上可以通过虚拟网络 WYLNAT 访问到虚拟机 vm01-clone,如图 5-13 所示。

(10) 查看在虚拟机 vm01-clone 是否可以访问外网中的其他主机,先查看一下外网

M QEMU/KVM Connection Deta	ils			-		×
File						
Overview Virtua	al Networks	Storage	Network Interfaces			
WYLNAT	Name:	WYLNAT				
	Device:	virbr0				
	State:	💵 Active				
	Autostart:	🗹 On Boot				
	Domain:	WYLNAT				
	 ✓ IPv4 cl Network: DHCP range Forwarding ↓ QoS co 	onfiguration 192.168.200.0/24 ge: 192.168.200.128 - g:	192.168.200.254			
					A	pply

图 5-10 新生成的 WYLNAT 网络

Will vr	n01-clone on QEMU/KVM		-		Х
File	Virtual Machine View	Send Key			
					4 0 0
	Overview	Virtual Network Interface			
44	Performance	Network source: Virtual network 'WYLNAT' : NAT to ens33 🔻			
$\overline{\Box}$	CPUs	Device model: virtio			
-	Memory	MAC address: 52:54:00:0f:03:a7			
60	Boot Options				
	VirtIO Disk 1				
÷	NIC :0f:03:a7				
	Tablet				
ð	Mouse				
	Keyboard				
<u> </u>	Display Spice				
	Sound: ich6				
a	Serial 1				
a	Channel qemu-ga				
â	Channel spice				
	Video QXL				
m.	Controller USB				
	Controller PCI				
	Add Hardware	Remove Cance		App	у

图 5-11 虚拟机选择使用 WYLNAT 网络

			vm01-cl	one on QEMU	/KVM			-	
File Virtual	Machine Viev	v Send Key							
	⊳ 00	•	Đ						
Red Hat Ente Kernel 3.10.	erprise Linux 0-514.el7.xf	< Server 7.3 36_64 on an p	(Maipo) ×86_64						
localhost lo	gin: root								
Password: Last login:	934 Fab 15 4	19.39.42 from	w asteriou						
front@localh	unst ~1tt in a	10.30.42 110 1ddr	ո ցուշտոց						
1: 1o: <loop< td=""><th>BACK, UP, LOW</th><th>R_UP> mtu 65</th><td>5536 gdisc m</td><th>noqueue stat</th><th>e UNKNOWN g</th><td>len 1</td><td></td><th></th><td></td></loop<>	BACK, UP, LOW	R_UP> mtu 65	5536 gdisc m	noqueue stat	e UNKNOWN g	len 1			
link/loo	pback 00:00	00:00:00:00	brd 00:00:0	30:00:00:00					
inet 127	'.0.0.1∕8 sco	pe host lo							
valid	l_lft foreven	preferred_	lft forever						
ineto ::	1/128 scope	host	lft foneien						
2: eth0: <bb< td=""><th>LIIC TOPEVEL MADCAST.MUL'</th><th>TICAST.UP.LO</th><td>JER HP> mtu</td><th>1500 adisc</th><th>nfifo fast</th><td>state IIP</td><td>ulen</td><th>1000</th><td></td></bb<>	LIIC TOPEVEL MADCAST.MUL'	TICAST.UP.LO	JER HP> mtu	1500 adisc	nfifo fast	state IIP	ulen	1000	
link/eth	er 52:54:00	0f:03:a7 bro	1:11:11:11:11	11:11:11:11:11:11:11:11:11:11:11:11:11:	pr 110_1 000		41011		
[root@localh	iost ~]# dhc	lient							
[root@localh	nost ~]# ip a	idd show ethe	3						
2: eth0: <br< td=""><th>IDADCAST, MUL</th><th>FICAST, UP, LOU</th><td>JER_UP> mtu</td><th>1500 qdisc</th><th>pfifo_fast</th><td>state UP</td><td>qlen</td><th>1000</th><td></td></br<>	IDADCAST, MUL	FICAST, UP, LOU	JER_UP> mtu	1500 qdisc	pfifo_fast	state UP	qlen	1000	
link/eth	er 52:54:00	Uf :U3:a7 bro	1 ff:ff:ff:ff:f	fiffiff Socono glob	al dunamio	ot10			
ualid	100.200.15 1ft 3591ser	neferred	.100.200.23. lft 3591sec	i scope giou	ar uynamic	C (110)			
inet6 fe	80::5054:ff	fe0f:3a7/64	scope link						
valid	l_lft forever	preferred_	lft forever						
[root@localh	iost~]#								

图 5-12 使用 WYLNAT 网络启动后的网卡情况

[root@node1 ~]# ping -c4 192.168.200.195 PING 192.168.200.195 (192.168.200.195) 56(84) bytes of data. 64 bytes from 192.168.200.195: icmp_seq=1 ttl=64 time=0.889 ms 64 bytes from 192.168.200.195: icmp_seq=2 ttl=64 time=0.900 ms 64 bytes from 192.168.200.195: icmp_seq=3 ttl=64 time=0.844 ms 64 bytes from 192.168.200.195: icmp_seq=4 ttl=64 time=0.950 ms --- 192.168.200.195 ping statistics ---4 packets transmitted, 4 received, 0% packet loss, time 3004ms rtt min/avg/max/mdev = 0.844/0.895/0.950/0.052 ms [root@node1 ~]#

图 5-13 宿主机可以访问虚拟机

中的一台 Windows 主机的 IP 地址,如图 5-14 所示,其 IP 地址为 192.168.100.1,然后在 虚拟机 vm01-clone 中去 ping 这台 Windows 主机,发现是通的,如图 5-15 所示。

C:\Windows\system32>ipconfig	
₩indows IP 配置	
以太网适配器 以太网:	
媒体状态 连接特定的 DNS 后缀	媒体已断开连接
以太网适配器 VMware Network Adapter VM	net8:
连接特定的 DNS 后缀 : 本地链接 IP∀6 地址 : IP∀4 地址	fe80::847a:12c6:523e:5a2c%9 192.168.100.1 255.255.255.0

图 5-14 外网中的一台 Windows 主机

vmO1-clone on QEMU/KVM
File Virtual Machine View Send Key
<pre>Lroot@localhost ~]# Lroot@localhost ~]# Lroot@localhost ~]# ping -c4 192.168.100.1 PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data. 64 bytes from 192.168.100.1: icmp_seq=1 ttl=63 time=1.97 ms 64 bytes from 192.168.100.1: icmp_seq=2 ttl=63 time=1.82 ms 64 bytes from 192.168.100.1: icmp_seq=4 ttl=63 time=1.42 ms 192.168.100.1 ping statistics 4 packets transmitted, 4 received, 0% packet loss, time 3006ms rtt min/avg/max/mdev = 1.247/1.733/1.972/0.290 ms Lroot@localhost ~]#</pre>

图 5-15 虚拟机可以访问外网

(11) 在宿主机上查看网络的情况,发现新建 WYLNAT 网络后,多了一个 virbr0 的 网桥,并且 virbr0 中内置了一个网络接口 virbr0-nic,以及网桥上的另一个接口 vnet0,此 接口是连接虚拟机 vm01-clone 的,如图 5-16 所示。

```
[root@node1 ~]# ip address
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 ::\overline{1}/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:1d:d8:c5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.145/24 brd 192.168.100.255 scope global dynamic ens33
       valid_lft 1706sec preferred_lft 1706sec
    inet6 fe80::20c:29ff:fe1d:d8c5/64 scope link
valid lft forever preferred lft forever
25: virbr0: <br/>
BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP qlen 1000
    link/ether 52:54:00:ef:f0:2d brd ff:ff:ff:ff:ff:ff
    inet 192.168.200.1/24 brd 192.168.200.255 scope global virbr0
       valid lft forever preferred lft forever
26: virbr0-nīc: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo fast master virbr0 state DOWN qlen 1000
    link/ether 52:54:00:ef:f0:2d brd ff:ff:ff:ff:ff:ff
27: vnet0: <BROADCAST,MULTICAST,UP,LOWER UP> mtu 1500 gdisc pfifo fast master virbr0 state UNKNOWN glen 100
    link/ether fe:54:00:0f:03:a7 brd ff:ff:ff:ff:ff
    inet6 fe80::fc54:ff:fe0f:3a7/64 scope link
       valid_lft forever preferred_lft forever
[root@node1 ~]#
```

图 5-16 宿主机网卡信息



5.1.3 NAT 网络的字符配置方法

如果没有图形的情况下或者在进行自动化部署的时候,就需要了解字符界面下的配置,具体操作如下:

(1) 查看当前的虚拟网络,如图 5-17 所示。图中显示了网络的名字(Name),状态(State) 是激活的,宿主机启动时是自动(Autostart) 开启网络的,此网络是永久(Persistent)有效的。

(2) 查看 WYLNAT 网络中的网桥情况,如图 5-18 所示,网桥名字为 virbr0,此网桥 有两个接口,一个是网桥内置的接口 virbr0-nic,另一个是连接虚拟机 vm01-clone 的接口 vnet0。

[root@node1 ~]# virsh net-list Name State Autostart Persistent WYI NAT active ves yes [root@node1 ~]# 图 5-17 查看虚拟网络状态 [root@node1 ~]# brctl show virbr0 bridge name bridge id STP enabled interfaces 8000.525400eff02d virbr0 ves virbr0-nic vnet0 [root@node1 ~]#

图 5-18 查看网桥 virbr0

(3) 查看路由情况,如图 5-19 所示,所有前往网段 192.168.200.0/24 的数据包,都从 virbr0 发出去。

[root@node1 ~]#	route -n						
Kernel IP routir	ng table						
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	192.168.100.2	0.0.0.0	UG	100	0	0	ens33
192.168.100.0	0.0.0.0	255.255.255.0	U	0	0	0	ens33
192.168.200.0	0.0.0.0	255.255.255.0	U	0	0	0	virbr0
[root@node1 ~]#							

图 5-19 查看宿主机路由表

(4)通过配置文件新建网络,要用到配置文件,配置文件有一个模板,在/usr/share/ libvirt/network/目录下,名字为 default,可以复制并修改此配置文件,来创建新的网络。 如图 5-20 所示,生成一个新的网络配置文件 nat. xml。

```
[root@node1 ~]# cd /usr/share/libvirt/networks/
[root@node1 networks]#
[root@node1 networks]# ls
default.xml
[root@node1 networks]#
[root@node1 networks]# cp default.xml nat.xml
[root@node1 networks]#
[root@node1 networks]# vim nat.xml
[root@node1 networks]#
```

图 5-20 生成新的配置文件 nat. xml

(5) 修改网络配置文件 nat. xml,如图 5-21 所示,一是修改网络名字为 WJHNAT,二 是修改网桥的名字为 virbr1,三是修改网桥的 IP 地址为 192.168.201.1,四是修改此网络 提供的 IP 地址池,范围为 192.168.201.101~192.168.201.200。

(6) 配置文件生成之后, 就可以通过 virsh net-define 命令定义此网络, 如图 5-22 所示, 定义完成后, 并没有发现此网络, 通过--all 选项才能看到, 是因为此网络没有激活。

(7) 如图 5-23 所示,通过 net-start 来激活此网络,发现此时状态为 active 了。

(8)图 5-23 中显示网络激活了,但是并不会随着宿主机启动而启动此网络,因此,需

68

图 5-21 修改网络配置文件 nat. xml

[root@nodel networks]# virsh net-define /usr/share/libvirt/networks/nat.xml
Network WJHNAT defined from /usr/share/libvirt/networks/nat.xml

[root@node1 [root@node1	networks]# networks]# virsh ne	t-list	
Name	State	Autostart	Persistent
WYLNAT	active	yes	yes
[root@node1	networks]# virsh ne	t-listall	
Name	State	Autostart	Persistent
WJHNAT	inactive	no	yes

[root@node1 networks]#

图 5-22 定义网络 WJHNAT

[root@node1 networks]# virsh net-start WJHNAT
Network WJHNAT started

[root@node1 netwo	rks]# virsh ne	et-listall	
Name	State	Autostart	Persistent
WJHNAT	active	no	yes
WYLNAT	active	yes	yes

[root@node1 networks]#

图 5-23 激活网络 WJHNAT

要通过 net-autostart 命令来实现自动启动 WJHNAT 网络,如图 5-24 所示,至此 WJHNAT 网络配置完成了。

[root@node1 networks]# virsh net-autostart WJHNAT
Network WJHNAT marked as autostarted

[root@node1 networks]#

图 5-24 自动启动 WJHNAT 网络

(9) 网络 WJHNAT 配置完成后,如果虚拟机要使用此网络,需要在虚拟机的详情页面进行设置,如图 5-25 所示,让虚拟机使用 WJHNAT 网络。

(10) 启动虚拟机后,发现虚拟机获取了 WJHNAT 网络的相关信息,如图 5-26 所示。

Kai vr	n01-clone on QEMU/KVM		-		×
File	Virtual Machine View	Send Key			
					4 0 0 0
	Overview	Virtual Network Interface			
-1+1	Performance	Network source: Virtual network 'WJHNAT' : NAT 🔹			
	CPUs	Device model: virtio			
	Memory	MAC address: 52:54:00:0f:03:a7			
80	Boot Options				
	VirtIO Disk 1				
P	NIC :0f:03:a7				
	Tablet				
ð	Mouse				
<u> </u>	Keyboard				
	Display Spice				
U	Sound: ich6				
2	Serial 1				
a	Channel qemu-ga				
2	Channel spice				
<u> </u>	Video QXL				
	Controller USB				
m.	Controller PCI				
	Add Hardware	Remove		App	у

图 5-25 配置虚拟机网络源





5.2 桥接



5.2.1 桥接网络的原理

网桥方式连接网络就是虚拟机与宿主机接在同一个交换机上,如图 5-27 所示,此时 虚拟机与宿主机以及其他的主机都是在同一个网段,相互间可以直接通信。虚拟机与宿 主机之前通过网桥通信,而虚拟机与外部主机之间则通过外部的物理交换机通信。



图 5-27 网桥连网方式

默认情况下 KVM 是没有网桥的,因此,需要管理员对其进行配置,生成一个网桥 br0,如图 5-28 所示,然后将宿主机网卡 eth0、连接虚拟机的网卡 vnet0、vnet1 连接到网桥 br0,此时,虚拟机配置的 IP 地址,需要与物理机的 IP 地址以及外部主机的 IP 地址在同 一个网段,这样虚拟机就可以与宿主机以及外部主机之间进行通信了。



图 5-28 网桥连网的宿主机内部结构

5.2.2 桥接网络的字符配置方法

(1) 在网卡的配置文件目录中,生成一个新的配置文件 ifcfg-br0,如图 5-29 所示,并 在其上配置好 IP 地址,每个选项的具体含义如表 5-1 所示。

```
[root@node1 ~]# cd /etc/sysconfig/network-scripts/
[root@nodel network-scripts]# ls
ifcfg-ens33 ifdown-ipv6
                            ifdown-TeamPort ifup-ippp
                                                         ifup-routes
                                                                            network-functions
ifcfg-lo
             ifdown-isdn
                            ifdown-tunnel
                                             ifup-ipv6
                                                          ifup-sit
                                                                            network-functions-ipv6
                                                          .
ifup-Team
             ifdown-post
                            ifup
                                             ifup-isdn
ifdown
ifdown-bnep ifdown-ppp
                            ifup-aliases
                                                          ifup-TeamPort
                                             ifup-plip
ifdown-eth
             ifdown-routes
                            ifup-bnep
                                             ifup-plusb
                                                         ifup-tunnel
ifdown-ib
            ifdown-sit
                            ifup-eth
                                             ifup-post
                                                         ifup-wireless
ifdown-ippp ifdown-Team
                            ifup-ib
                                             ifup-ppp
                                                          init.ipv6-global
[root@nodel network-scripts]# vim ifcfg-br0
[root@nodel network-scripts]# cat ifcfg-br0
DEVICE=br0
NAME=br0
0NB00T=yes
TYPE=Bridge
BOOTPROTO=static
IPADDR=192.168.100.145
PRFFTX=24
GATEWAY=192.168.100.2
DNS1=192.168.100.2
[root@nodel network-scripts]# vim ifcfg-ens33
[root@nodel network-scripts]# cat ifcfg-ens33
NAME=ens33
DEVICE=ens33
0NB00T=ves
BRTDGE=br0
[root@node1 network-scripts]# systemctl restart network
```

图 5-29 网桥的配置方法

表 5-1 网桥 br0 配置文件选项含义

选项	含 义
DEVICE	设备名称, br0 为网桥的名字
NAME	连接名
ONBOOT	系统启动时启动此网桥
TYPE	设备类型,此处为网桥
BOOTPROTO	获得 IP 的方式, static 为手工配置
IPADDR	设备 IP 地址
PREFIX	设备掩码
GATEWAY	设置网关地址
DNS1	设置 DNS 服务器 1 的 IP 地址

br0 配置完成后,需要修改物理网卡的配置文件,其中 BRIDGE=br0 选项比较特殊, 其含义是将此物理网卡桥接到 br0 网桥上,使得物理网卡也连接到网桥上,实现了虚拟机 与宿主机之间的连接。

(2) 网桥 br0 与网卡 ens33 配置完成后,必须重启网络使其生效,如图 5-30 所示,查 看网桥的信息,确认 ens33 是否已经连接到网桥 br0 上了。

[root@nodel network-scripts]# systemctl restart network
[root@nodel network-scripts]#
[root@nodel network-scripts]# brctl show br0
bridge name bridge id STP enabled interfaces
br0 8000.000c291dd8c5 no ens33
[root@nodel network-scripts]#

图 5-30 重启网络

(3) 查看网卡 ens33 与网桥 br0,发现现在的 IP 地址等信息配置在网桥 br0 上了,如 图 5-31 所示。

```
[root@nodel ~]# ip addr show ens33
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master br0 state UP qlen 1000
    link/ether 00:0c:29:1d:d8:c5 brd ff:ff:ff:ff:ff
[root@nodel ~]#
[root@nodel ~]# ip addr show br0
34: br0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP qlen 1000
    link/ether 00:0c:29:1d:d8:c5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.145/24 brd 192.168.100.255 scope global br0
    valid_lft forever preferred_lft forever
[root@nodel ~]#
```

图 5-31 查看 IP 地址

(4) 如果虚拟机 vm01 要使用网桥 br0 进行连网,必须在虚拟机 vm01 的详情页面中 设置网络源为 Bridge br0,如图 5-32 所示。

www.vm01 on QEMU/KVM	-		×
File Virtual Machine View Send Key			
			0 0 0 0
Virtual Network Interface			
Performance Network source: Bridge br0: Host device ens33	-		
CPUs Device model: virtio 🗸			
Memory MAC address: 52:54:00:da:92:28			
Boot Options			
Svirtlo Disk 1			
📴 NIC :da:92:28			
Tablet			
👌 Mouse			
🚔 Keyboard			
Display Spice			
Sound: ich6			
Serial 1			
🚵 Channel qemu-ga			
Channel spice			
Controller USB			
Controller PCI			
Add Hardware Re	emove Cancel	Арр	ly

图 5-32 设置虚拟机的网络源

(5) 启动虚拟机 vm01,查看网桥 br0 的信息,发现虚拟机 vm01 是采用网卡 vnet0 来 连接网桥的,如图 5-33 所示。

(6) 虚拟机 vm01 启动后,可以将虚拟机 vm01 的 IP 地址设置为与宿主机同一个网段 192.168.100.0/24,并测试是否可以访问到宿主机,如图 5-34 所示,发现虚拟机 vm01 与宿主机之间可以正常通信了。

[root@node1 ~]#	brctl show br0		
bridge name	bridge id	STP enabled	interfaces
br0	8000.000c291dd8c5	no	ens33
			vnet0
[root@node1 ~]#			

图 5-33 查看网桥

vm01 on QEMU/K	vм
File Virtual Machine View Send Key	
S S S S S S S S S S S S S S S S S S S	
[root@localhost network-scripts]# cat ifcfg-eth0 NAME="eth0"	
DEVICE=eth0	
ONBOOT=yes	
BOOTPROTO=none	
IPADDR=192.168.100.101	
PREF IX=24	
GHILWHY=192.168.100.2 DNS1-192.168.100.2	
[root@localhost_network-scripts]#	
[root@localhost network-scripts]# sustemet1 restart network	
[root@localhost network-scripts]#	
[root@localhost network-scripts]# ip addr	
1: lo: <loopback,up,lower_up> mtu 65536 qdisc noqueue state UNKNOW</loopback,up,lower_up>	Nglen 1
link/loopback 00:00:00:00:00 brd 00:00:00:00:00	
Inet 127.0.0.178 scope nost to	
inet6 ::1/128 score host	
valid lft forever preferred lft forever	
2: eth0: <broadcast,multicast,up,lower_up> mtu 1500 qdisc pfifo_fa</broadcast,multicast,up,lower_up>	st state UP glen 1000
link/ether 52:54:00:da:92:28 brd ff:ff:ff:ff:ff:ff	
inet 192.168.100.101/24 brd 192.168.100.255 scope global eth0	
valid_lft_forever_preferred_lft_forever inst{_fs90::E8E4:ff:fs1a:0228.64_scene_link	
INETS 1660::5054:HI:16da:5228/64 Scope IINK	
[root@localhost_network-scripts]#	
[root@localhost_network-scripts]# ping -c4 192.168.100.145	
PING 192.168.100.145 (192.168.100.145) 56(84) bytes of data.	
64 bytes from 192.168.100.145: icmp_seq=1 ttl=64 time=1.94 ms	
64 bytes from 192.168.100.145: icmp_seq=2 ttl=64 time=0.646 ms	
64 bytes from 192.168.100.145; icmp_seq=3 ttl=64 time=0.614 ms	
of bytes from 152.168.100.145; icmp_seq=4 tt1=64 time=0.543 ms	

图 5-34 测试是否可以通过桥接与宿主机通信

5.3 本章实验

74

5.3.1 实验目的

- ▶ 了解使用图形界面的方式来创建 NAT 网络。
- ▶掌握使用字符界面的方式来创建 NAT 网络。
- > 掌握使用字符界面的方式来创建桥接网络。

5.3.2 实验环境

在安装 KVM 的宿主机 nodel 上安装虚拟机 vm01 与 vm02。

5.3.3 实验拓扑

实验拓扑图如图 5-35 所示。



图 5-35 实验拓扑图

5.3.4 实验内容

如图 5-35 所示,创建 NAT 网络与桥接网络。

(1) 在宿主机上创建一个 NAT 的网络 NAT1, 网段地址为 192.168.200.0/24, 此时 会生成一个新的网桥 virbr1。虚拟机 vm01 选择此网络,将 vm01 重启后, 查看虚拟机 vm01 的 IP 地址,并且测试是否可以访问外网, 以及外网中的 PC 是否可以访问此虚拟 机 vm01。

(2) 在宿主机上创建一个网桥 br0,将物理网卡 ens33 连接至 br0。虚拟机 vm02 选择此网桥,将 vm02 重启后,查看虚拟机 vm02 的 IP 地址,并且测试是否可以访问外网以及外网是否可以访问此虚拟机 vm02。

说一说:

NAT 网络与桥接网络的原理,它们分别适应什么场合?

想一想:

实验中 vm02 会自动获取到 IP 地址吗?如果能获取到,它的 IP 地址是谁提供的呢?