

## 项目 3 配置网络和使用 ssh 服务-实训任务指导书 ( 3-2 )

### 任务 3-4 绑定两块网卡

一般来讲，生产环境必须提供 7×24 小时的网络传输服务。借助于网卡绑定技术，不仅可以提高网络传输速度，更重要的是，还可以确保在其中一块网卡出现故障时，依然可以正常提供网络服务。假设我们对两块网卡实施了绑定技术，这样在正常工作中它们会共同传输数据，使得网络传输的速度变得更快；而且即使有一块网卡突然出现了故障，另外一块网卡便会立即自动顶替上去，保证数据传输不会中断。

#### 准备工作：

1. 可恢复到快照 1 状态。



具体实训操作步骤：

- 第 1 步：**在虚拟机系统中再添加一块网卡设备，请确保两块网卡都处在同一个网络连接中（即网卡模式相同），如图 2-18 和图 2-19 所示。处于**相同模式**的**网卡设备**才可以进行网卡绑定，否则这两块网卡无法互相传送数据。

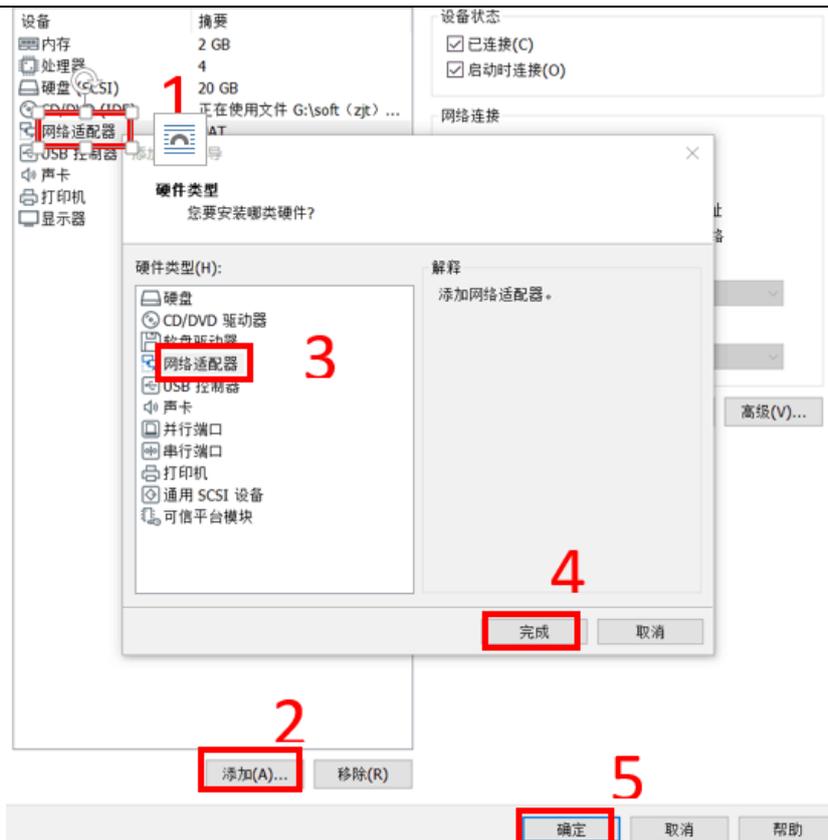


图 2-18 在虚拟机中再添加一块网卡设备

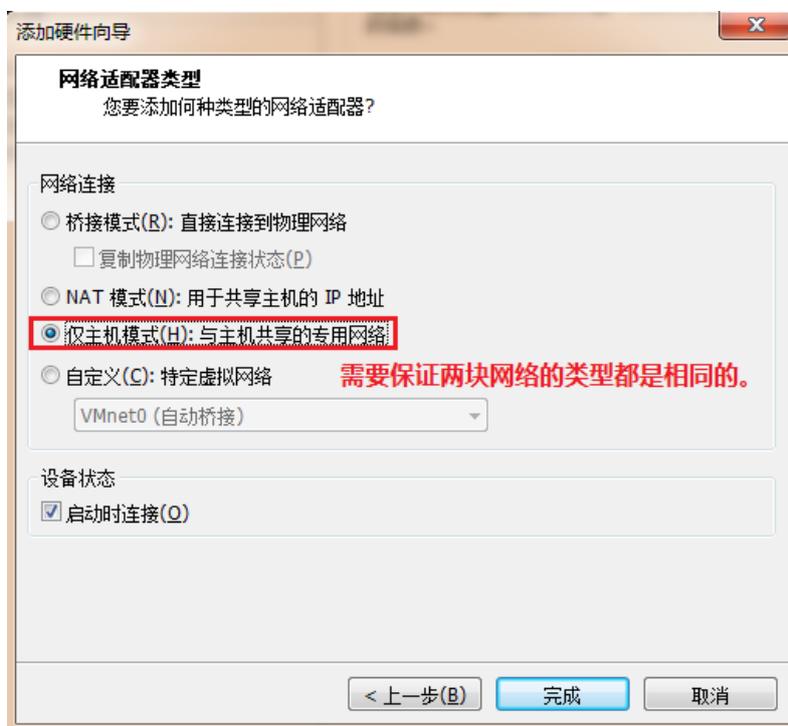


图 2-19 确保两块网卡处在同一个网络连接中 (即网卡模式相同)

虚拟机操作，网卡可采用 NAT 模式

第 2 步：使用 Vim 文本编辑器来配置网卡设备的绑定参数，逐个进行“初始设置”。这俩原本独立的网卡设备此时需要被配置成为一块“从属”网卡，服务于“主”网卡，不应该再有自己的 IP 地址等信息。在进行了初始设置之后，它们就可以支持网卡绑定。先使用 ifconfig 命令查询两块网卡的名称为：ens33 和 ens37。

```
[root@192 ~]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.58.165 netmask 255.255.255.0 broadcast 192.168.58.255
    inet6 fe80::b9bc:46be:e31d:dd38 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:32:c1:32 txqueuelen 1000 (Ethernet)
    RX packets 109633 bytes 161545569 (154.0 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 21809 bytes 1321666 (1.2 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens37: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.58.167 netmask 255.255.255.0 broadcast 192.168.58.255
    inet6 fe80::506e:dd3d:4f66:5db1 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:32:c1:3c txqueuelen 1000 (Ethernet)
    RX packets 6 bytes 926 (926.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 28 bytes 4725 (4.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
[root@RHEL7-1 ~]# cd /etc/sysconfig/network-scripts/
[root@RHEL7-1 network-scripts]# ll
[root@RHEL7-1 network-scripts]# vim ifcfg-ens33
TYPE=Ethernet
BOOTPROTO=none
ONBOOT=yes
USERCTL=no
DEVICE=ens33
MASTER=bond0
SLAVE=yes
编辑完成后，保存退出
[root@RHEL7-1 network-scripts]# vim ifcfg-ens37
TYPE=Ethernet
BOOTPROTO=none
ONBOOT=yes
USERCTL=no
DEVICE=ens37
MASTER=bond0
SLAVE=yes
编辑完成后，保存退出
```

是否允许非 root 用户控制该设备

操作技巧：可以把原来 ens33 网卡的内容首行输入#注释掉，如 1-15 行行首替换成#，vim ens33 命令

行模式输入：**1,15 s/^/#/g**，一次完成替换。

第 3 步还需要将**绑定后的设备命名为 bond0** 并把 IP 地址等信息填写进去，这样当用户访问相应服务的时候，实际上就是由这两块网卡设备在共同提供服务。

```
[root@RHEL7-1 ~]# cd /etc/sysconfig/network-scripts
[root@RHEL7-1 network-scripts]# vim ifcfg-bond0
TYPE=Ethernet
BOOTPROTO=none
ONBOOT=yes
USERCTL=no
DEVICE=bond0
IPADDR=192.168.10.1
PREFIX=24
DNS=192.168.10.1
NM_CONTROLLED=no
```

network manager 的参数，实时生效，不需要重启

第 4 步：让 Linux 内核支持网卡绑定驱动，修改/etc/modprobe.d/bonding.conf 配置文件。

常见的网卡绑定驱动有三种模式—mode0、mode1 和 mode6。

- mode0（平衡负载模式）：平时两块网卡均工作，且自动备援，但需要在与服务器本地网卡相连的交换机设备上端口聚合来支持绑定技术。
- mode1（自动备援模式）：平时只有一块网卡工作，在它故障后自动替换为另外的网卡。
- **mode6（平衡负载模式）**：平时两块网卡均工作，且自动备援，无须交换机设备提供辅助支持。

访问服务器的用户数量特别多时，为保障网络的可靠性，可选择 mode6 网卡绑定驱动模式。让两块网卡同时一起工作，当其中一块网卡出现故障后能自动备援，无需交换机设备支援，提供了可靠的网络传输保障。

下面使用 Vim 文本编辑器**创建一个用于网卡绑定的驱动文件**，使得绑定后的

**bond0 网卡设备能够支持绑定技术 (bonding); 同时定义网卡以 mode6 模式进行绑定, 且出现故障时自动切换的时间为 100 毫秒。**

```
[root@RHEL7-1 ~]# vim /etc/modprobe.d/bonding.conf
alias bond0 bonding
options bond0 miimon=100 mode=6
```

**第 5 步: 先停止网络管理服务再重启网络服务后网卡绑定操作即可成功。正常情况下只有 bond0 网卡设备才会有 IP 地址等信息:**

```
[root@RHEL7-1 ~]# systemctl stop NetworkManager
[root@RHEL7-1 ~]# systemctl restart network
[root@RHEL7-1 ~]# ifconfig
```

```
root@192 network-scripts]# vim /etc/modprobe.d/bonding.conf
root@192 network-scripts]# systemctl stop NetworkManager
root@192 network-scripts]# systemctl restart network
root@192 network-scripts]# ifconfig
bond0: flags=5187<UP,BROADCAST,RUNNING,MASTER,MULTICAST> mtu 1500
    inet 192.168.10.1 netmask 255.255.255.0 broadcast 192.168.10.255
    inet6 fe80::20c:29ff:fe32:c132 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:32:c1:32 txqueuelen 1000 (Ethernet)
    RX packets 8 bytes 522 (522.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 70 bytes 6633 (6.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens33: flags=6211<UP,BROADCAST,RUNNING,SLAVE,MULTICAST> mtu 1500
    ether 00:0c:29:32:c1:32 txqueuelen 1000 (Ethernet)
    RX packets 109901 bytes 161570975 (154.0 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 22023 bytes 1341004 (1.2 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens37: flags=6211<UP,BROADCAST,RUNNING,SLAVE,MULTICAST> mtu 1500
    ether 00:0c:29:32:c1:3c txqueuelen 1000 (Ethernet)
    RX packets 122 bytes 11526 (11.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 22023 bytes 1341004 (1.2 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

**可以在本地主机执行 ping 192.168.10.1 命令检查网络的连通性。**

```
[root@192 network-scripts]# cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.7.1 (April 27, 2011)

Bonding Mode: adaptive load balancing
Primary Slave: None
Currently Active Slave: ens33
MII Status: up
MII Polling Interval (ms): 100
Up Delay (ms): 0
Down Delay (ms): 0

Slave Interface: ens33
MII Status: up
Speed: 1000 Mbps
Duplex: full
Link Failure Count: 0
Permanent HW addr: 00:0c:29:32:c1:32
Slave queue ID: 0

Slave Interface: ens37
MII Status: up
Speed: 1000 Mbps
Duplex: full
Link Failure Count: 0
Permanent HW addr: 00:0c:29:32:c1:3c
Slave queue ID: 0
```

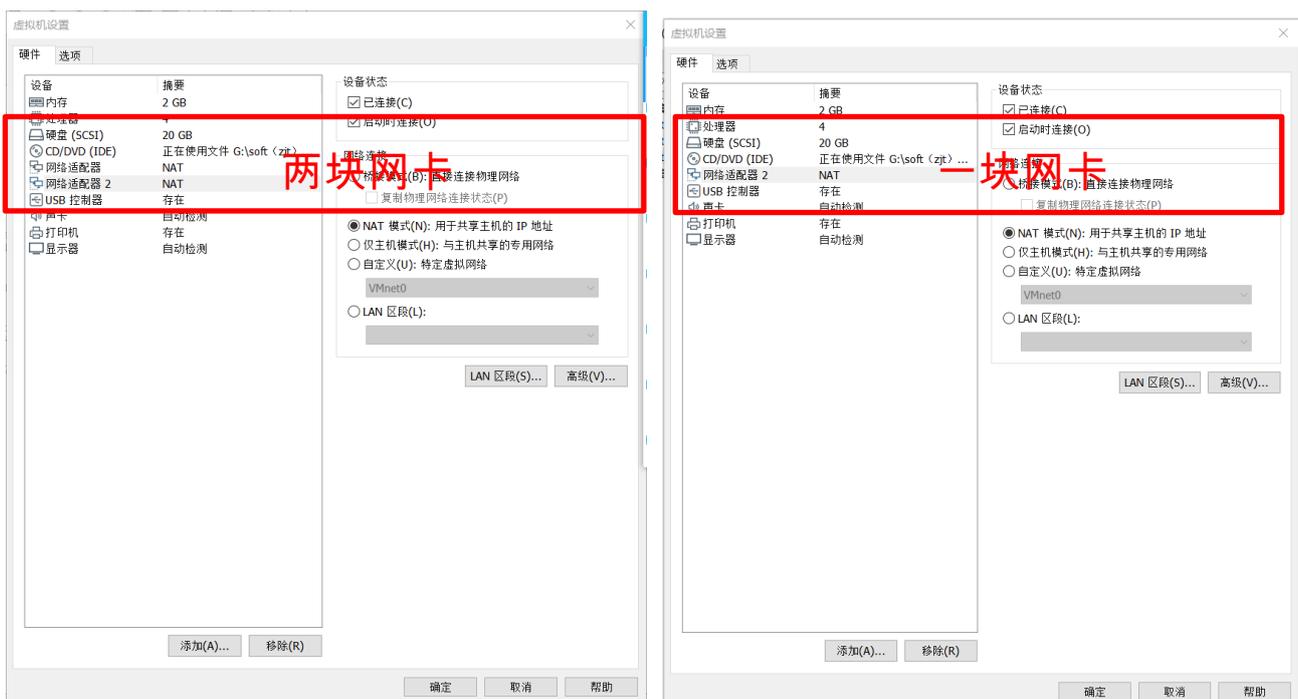
**•linux 中查看 bond 状态时, cat /proc/net/bonding/bond0**

```
网络管理 (NetworkManager) 未运行。
[root@192 network-scripts]# ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data:
64 bytes from 192.168.10.1: icmp_seq=1 ttl=64 time=0.062 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=64 time=0.107 ms
64 bytes from 192.168.10.1: icmp_seq=3 ttl=64 time=0.106 ms
64 bytes from 192.168.10.1: icmp_seq=4 ttl=64 time=0.136 ms
64 bytes from 192.168.10.1: icmp_seq=5 ttl=64 time=0.133 ms
64 bytes from 192.168.10.1: icmp_seq=6 ttl=64 time=0.107 ms
64 bytes from 192.168.10.1: icmp_seq=7 ttl=64 time=0.097 ms
64 bytes from 192.168.10.1: icmp_seq=8 ttl=64 time=0.122 ms
^C
192.168.10.1 ping statistics:
8 packets transmitted, 8 received, 0% packet loss, time 7004ms
rtt min/avg/max/mdev = 0.062/0.108/0.136/0.025 ms
```

并且使用 firefox 浏览器，能正常访问 apache 服务器。



第 6 步：为了检验网卡绑定技术的自动备援功能，我们突然在虚拟机硬件配置中随机移除一块网卡设备，然后另外一块网卡会继续为用户提供服务。如下图，硬件删除 ens33 网卡，还剩 ens37 一块网卡。



使用 ping 命令可以非常清晰地看到网卡切换的过程（一般只有 1 个数据丢包或不丢包）。

```
[root@192 network-scripts]# ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_seq=1 ttl=64 time=0.087 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=64 time=0.066 ms
64 bytes from 192.168.10.1: icmp_seq=3 ttl=64 time=0.086 ms
64 bytes from 192.168.10.1: icmp_seq=4 ttl=64 time=0.152 ms
64 bytes from 192.168.10.1: icmp_seq=5 ttl=64 time=0.123 ms
^C
--- 192.168.10.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4002ms
rtt min/avg/max/mdev = 0.066/0.102/0.152/0.033 ms
[root@192 network-scripts]#
```

### •linux 中查看 bond 状态时, cat /proc/net/bonding/bond0

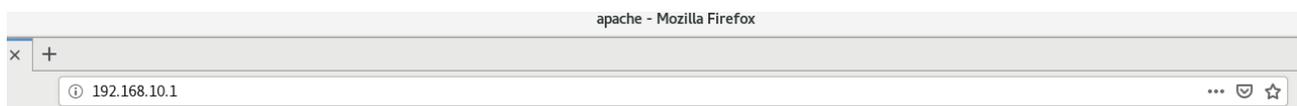
```
[root@192 network-scripts]# cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.7.1 (April 27, 2011)

Bonding Mode: adaptive load balancing
Primary Slave: None
Currently Active Slaves: ens37
MII Status: up
MII Polling Interval (ms): 100
Up Delay (ms): 0
Down Delay (ms): 0

Slave Interface: ens33
MII Status: up
Speed: 1000 Mbps
Duplex: full
Link Failure Count: 0
Permanent HW addr: 00:0c:29:32:c1:32
Slave queue ID: 0

Slave Interface: ens37
MII Status: up
Speed: 1000 Mbps
Duplex: full
Link Failure Count: 0
Permanent HW addr: 00:0c:29:32:c1:3c
Slave queue ID: 0
[root@192 network-scripts]#
```

使用 firefox 浏览器，仍能正常访问 apache 服务器。



注意：做完绑定网卡的实验后，为了不影响其他实训，请利用 VM 快照恢复到系统初始状态。或者删除掉绑定：删除网卡绑定的配置文件，利用系统菜单重新配置网络，然后重启系统。