

2022

Linux企业运维与实战

教师：丁敏



项目3：配置网络和使用SSH服务

- 项目导入

作为Linux系统的网络管理员，学习Linux服务器的网络配置是至关重要的，同时管理远程主机也是管理员必须熟练掌握的。这些是后续网络服务配置的基础，必须要学好。

本项目讲解了如何使用nmtui命令配置网络参数，以及通过nmcli命令查看网络信息并管理网络会话服务，从而让您能够在不同工作场景中快速地切换网络运行参数的方法；还讲解了如何手工绑定mode6模式双网卡，实现网络的负载均衡的方法。本项目还深入介绍了SSH协议与sshd服务程序的理论知识、Linux系统的远程管理方法以及在系统中配置服务程序的方法。

- 职业能力目标和要求

- ① 掌握常见网络服务的配置方法。
- ② 掌握远程控制服务。
- ③ 掌握不间断会话服务。



项目3：配置网络和使用SSH服务

检查并设置有线处于连接状态

任务3-1-设置主机名

任务3-2-配置网络

任务3-3 创建网络会话实例

任务3-4绑定两块网卡

任务3-5 配置远程控制服务

综合实训-配置Linux下的TCP/IP和远程管理

使用系统菜单配置网络

通过网卡配置文件配置网络

使用图形界面配置网络

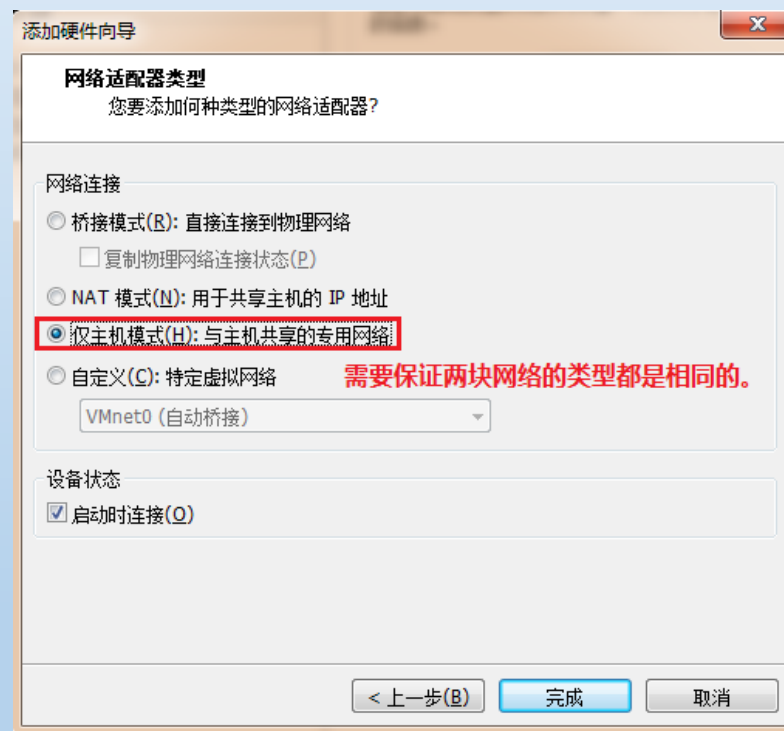
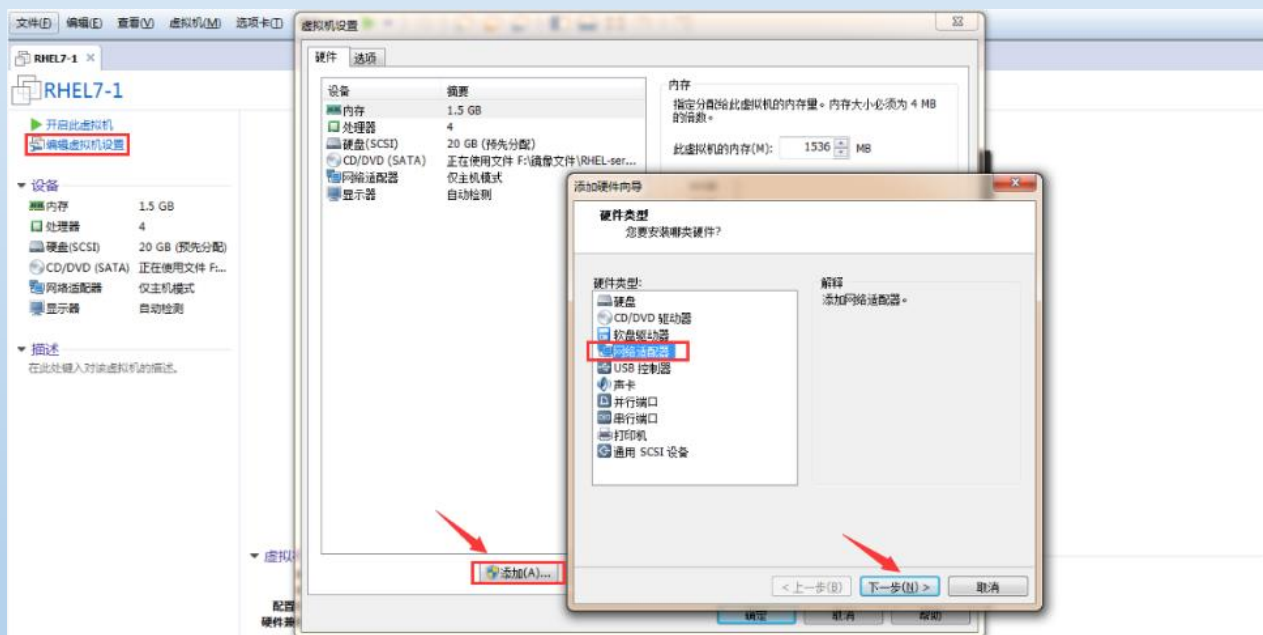
使用nmcli命令配置网络



项目3：配置网络和使用SSH服务

任务3-4 绑定两块网卡

▶ 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。





项目3: vim 配置网络和使用SSH服务

▶ [root@RHEL7-1 ~]# vim /etc/sysconfig/network-scripts/ifcfg-ens33

▶ TYPE=Ethernet

▶ BOOTPROTO=none

▶ ONBOOT=yes

▶ USERCTL=no

▶ DEVICE=ens33

▶ MASTER=bond0

▶ SLAVE=yes

▶ [root@RHEL7-1 ~]# vim /etc/sysconfig/network-scripts/ifcfg-ens38

▶ TYPE=Ethernet

▶ BOOTPROTO=none

▶ ONBOOT=yes

▶ USERCTL=no

▶ DEVICE=ens38

▶ MASTER=bond0

▶ SLAVE=yes



项目3：配置网络和使用SSH服务

▶ 第2步：使用Vim文本编辑器来配置网卡设备的绑定参数。

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 ~]# vim /etc/sysconfig/network-scripts/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
```



项目3：配置网络和使用SSH服务

- ▶ 第3步：让Linux内核支持网卡绑定驱动。

```
[root@RHEL7-1 ~]# vim /etc/modprobe.d/bond.conf
```

```
alias bond0 bonding  
options bond0 miimon=100 mode=6
```



项目3：配置网络和使用SSH服务

- ▶ 第4步：重启网络服务后网卡绑定操作即可成功。

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

```
[root@RHEL7-1 ~]# ifconfig
```

```
bond0: flags=5123<UP,BROADCAST,MASTER,MULTICAST> mtu 1500  
  inet 192.168.10.1 netmask 255.255.255.0 broadcast 192.168.10.255  
  ether 86:08:25:89:b4:6d txqueuelen 1000 (Ethernet)  
  RX packets 0 bytes 0 (0.0 B)  
  RX errors 0 dropped 0 overruns 0 frame 0  
  TX packets 0 bytes 0 (0.0 B)  
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```




项目3：配置网络和使用SSH服务

- ▶ 第4步：重启网络服务后网卡绑定操作即可成功。

```
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
ether 00:0c:29:66:42:8d txqueuelen 1000 (Ethernet)
RX packets 119 bytes 12615 (12.3 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
ens38: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
ether 00:0c:29:66:42:97 txqueuelen 1000 (Ethernet)
RX packets 48 bytes 6681 (6.5 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```



项目3：配置网络和使用SSH服务

- ▶ **第4步：**重启网络服务后网卡绑定操作即可成功。
 - ▶ 可以在本地主机执行 `ping 192.168.10.1` 命令检查网络的连通性。
 - ▶ 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，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）。然后另外一块网卡会继续为用户提供服务。



项目3：配置网络和使用SSH服务

- ▶ 第4步：重启网络服务后网卡绑定操作即可成功。
 - ▶ [root@RHEL7-1 ~]# **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=0.171 ms
 - ▶ 64 bytes from 192.168.10.1: icmp_seq=2 ttl=64 time=0.048 ms
 - ▶ 64 bytes from 192.168.10.1: icmp_seq=3 ttl=64 time=0.059 ms
 - ▶ 64 bytes from 192.168.10.1: icmp_seq=4 ttl=64 time=0.049 ms
 - ▶ ping: sendmsg: Network is unreachable
 - ▶ --- 192.168.10.1 ping statistics ---
 - ▶ 8 packets transmitted, 7 received, 12% packet loss, time 7006ms
 - ▶ rtt min/avg/max/mdev = 0.042/0.073/0.109/0.023 ms



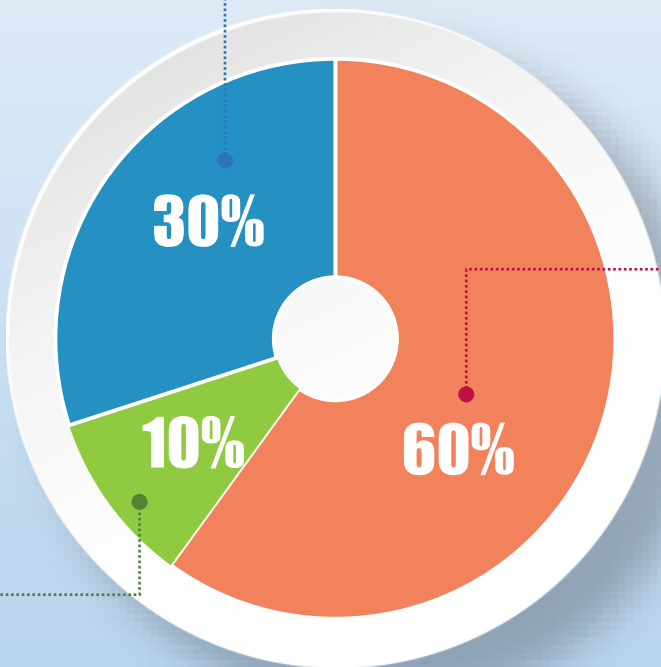
考核方式

过程性评价

- 学习态度
- 课堂作业
- 单元测试
- 子项目考核

综合素质评价

- 考勤
- 团队协作能力
- 创新能力



总结性评价

- 期末测试
- 项目答辩