

# 编辑网络配置文件

## 培训目标

学完本节后，您应能够通过编辑配置文件来修改网络配置。

## 描述连接配置文件

默认情况下，通过 `nmcli con mod name` 进行的更改会自动保存到 `/etc/sysconfig/network-scripts/ifcfg-name`。还可以使用文本编辑器手动编辑此文件。执行此操作后，运行 `nmcli con reload` 以便 NetworkManager 读取配置更改。

出于向后兼容性的原因，此文件中保存的指令具有不同于 `nm-settings(5)` 名称的名称和语法。下表将部分关键设置名称映射到 `ifcfg-*` 指令。

### `nm-settings` 与 `ifcfg-*` 指令的比较

NMCLI CON MOD	IFCFG-* FILE	影响
<code>ipv4.method manual</code>	<code>BOOTPROTO=none</code>	IPv4 以静态方式配置。
<code>ipv4.method auto</code>	<code>BOOTPROTO=dhcp</code>	从 DHCPv4 服务器中查找配置设置。如果还设置了静态地址，则在我们从 DHCPv4 中获取信息之前，将不会激活这些静态地址。
<code>ipv4.addresses "192.0.2.1/24 192.0.2.254"</code>	<code>IPADDR0=192.0.2.1 PREFIX0=24 GATEWAY0=192.0.2.254</code>	设置静态 IPv4 地址、网络前缀和默认网关。如果为连接设置了多个，则 <code>ifcfg-*</code> 指令将以 1、2、3 等结尾，而不是以 0 结尾。
<code>ipv4.dns 8.8.8.8</code>	<code>DNS0=8.8.8.8</code>	修改 <code>/etc/resolv.conf</code> 以使用此名称服务器。
<code>ipv4.dns-search example.com</code>	<code>DOMAIN=example.com</code>	修改 <code>/etc/resolv.conf</code> ，以在 <code>search</code> 指令中使用这个域。
<code>ipv4.ignore-auto-dns true</code>	<code>PEERDNS=no</code>	忽略来自 DHCP 服务器的 DNS 服务器信息。
<code>ipv6.method manual</code>	<code>IPV6_AUTOCONF=no</code>	IPv6 地址以静态方式配置。
<code>ipv6.method auto</code>	<code>IPV6_AUTOCONF=yes</code>	使用路由器播发中的 SLAAC 来配置网络设置。

NMCLI CON MOD	IFCFG-* FILE	影响
<code>ipv6.method dhcp</code>	<code>IPV6_AUTOCONF=no</code> <code>DHCPV6C=yes</code>	使用 DHCPv6 (而不使用 SLAAC) 来配置网络设置。
<code>ipv6.addresses "2001:db8::a/64 2001:db8::1"</code>	<code>IPV6ADDR=2001:db8::a/64</code> <code>IPV6_DEFAULTGW=2001:db8</code>	设置静态 IPv6 地址、网络前缀和默认网关。如果为连接设置了多个地址, <code>IPV6_SECONDARIES</code> 将采用空格分隔的地址/前缀定义的双引号列表。
<code>ipv6.dns ...</code>	<code>DNS0= ...</code>	修改 <code>/etc/resolv.conf</code> 以使用此名称服务器。与 IPv4 完全相同。
<code>ipv6.dns-search example.com</code>	<code>DOMAIN=example.com</code>	修改 <code>/etc/resolv.conf</code> , 以在 <code>search</code> 指令中使用这个域。与 IPv4 完全相同。
<code>ipv6.ignore-auto-dns true</code>	<code>IPV6_PEERDNS=no</code>	忽略来自 DHCP 服务器的 DNS 服务器信息。
<code>connection.autoconnect yes</code>	<code>ONBOOT=yes</code>	在系统引导时自动激活此连接。
<code>connection.id ens3</code>	<code>NAME=ens3</code>	此连接的名称。
<code>connection.interface-name ens3</code>	<code>DEVICE=ens3</code>	连接与具有此名称的网络接口绑定。
<code>802-3-ethernet.mac-address ...</code>	<code>HWADDR= ...</code>	连接与具有此 MAC 地址的网络接口绑定。

## 修改网络配置

也可以通过直接编辑连接配置文件来配置网络。连接配置文件控制单个网络设备的软件接口。这些文件通常命名为 `/etc/sysconfig/network-scripts/ifcfg-name`, 其中 name 是指配置文件控制的设备或连接的名称。以下是在用于静态或动态 IPv4 配置的文件中找到的标准变量。

**ifcfg 文件的 IPv4 配置选项**

静态	动态灵活	任意
<b>BOOTPROTO=none</b> <b>IPADDR0=172.25.250.10</b> <b>PREFIX0=24</b> <b>GATEWAY0=172.25.250.254</b> <b>DEFROUTE=yes</b> <b>DNS1=172.25.254.254</b>	<b>BOOTPROTO=dhcp</b>	<b>DEVICE=ens3</b> <b>NAME="static-ens3"</b> <b>ONBOOT=yes</b> <b>UUID=f3e8( . . . )ad3e</b> <b>USERCTL=yes</b>

在静态设置中，IP 地址、前缀和网关等变量的末尾都是数字。这允许将多组值指定到该接口。DNS 变量也有一个数字，用于在指定了多个服务器时指定查询的顺序。

在修改了配置文件后，请运行 **nmcli con reload** 使 NetworkManager 读取配置更改。接口依然需要重新启动，以便更改生效。

```
[root@host ~]# nmcli con reload
[root@host ~]# nmcli con down "static-ens3"
[root@host ~]# nmcli con up "static-ens3"
```

**参考文献**

**nmcli(1) man page**

如需更多信息，请参阅红帽企业 Linux 8.0中的配置和管理网络，网址为：  
[https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/8/html-single/configuring\\_and\\_managing\\_networking/](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/configuring_and_managing_networking/)

## ▶ 指导练习

# 编辑网络配置文件

在本练习中，您将手动修改网络配置文件并确保新设置生效。

## 成果

您能够向各个系统添加一个额外的网络地址。

## 在你开始之前

在 workstation 上，以 student 用户身份并使用密码 student 进行登录。

从 workstation，运行 **lab net-edit start** 命令。该命令将运行一个起始脚本，它将确定主机 servera 和 serverb 是否可从网络访问。

```
[student@workstation ~]$ lab net-edit start
```

- ▶ 1. 使用 **ssh** 命令，以 student 用户身份登录 servera。系统已配置为使用 SSH 密钥来进行身份验证，因此不需要输入密码就能登录 servera。

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- ▶ 2. 找到网络接口名称。



### 重要

网络接口名称由其总线类型和启动期间设备的检测顺序确定。具体的网络接口名称会因所用的课程平台和硬件而异。

在您的系统上，找到与以太网地址 **52:54:00:00:fa:0a** 相关联的接口名称（如 **ens06** 或 **en1p2**）。使用此接口名称替换本练习中所用的 **enX** 占位符。

找到与以太网地址 **52:54:00:00:fa:0a** 相关联的网络接口名称。记下或记住此名称，并使用它来替换后续命令中的 **enX** 占位符。活动连接也被命名为 **Wired connection 1** [因由文件 **/etc/sysconfig/network-scripts/ifcfg-Wired\_connection\_1** 管理]。

如果您在本章中已经完成了以前的练习，并且这对您的系统来说也是如此，那么此练习也应如此。

```
[student@servera ~]$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    qlen 1000
        link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
```

```
2: enX: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
    default qlen 1000
        link/ether 52:54:00:00:fa:0a brd ff:ff:ff:ff:ff:ff
[student@servera ~]$ nmcli con show --active
NAME           UUID                                  TYPE      DEVICE
Wired connection 1  03da038a-3257-4722-a478-53055cc90128  ethernet  enX
[student@servera ~]$ ls /etc/sysconfig/network-scripts/ifcfg-Wired_connection_1
/etc/sysconfig/network-scripts/ifcfg-Wired_connection_1
```

► 3. 编辑 servera 上的 **/etc/sysconfig/network-scripts/ifcfg-Wired\_connection\_1** 文件，以便增加一个地址 **10.0.1.1/24**。

3.1. 在文件中附加一个条目，以指定 IPv4 地址。

```
[student@servera ~]$ echo "IPADDR1=10.0.1.1" | \
sudo tee -a /etc/sysconfig/network-scripts/ifcfg-Wired_connection_1
[sudo] password for student: student
IPADDR1=10.0.1.1
```

3.2. 在文件中附加一个条目，以指定网络前缀。

```
[student@servera ~]$ echo "PREFIX1=24" | \
sudo tee -a /etc/sysconfig/network-scripts/ifcfg-Wired_connection_1
PREFIX1=24
```

► 4. 激活新的地址。

4.1. 重新加载配置更改。

```
[student@servera ~]$ sudo nmcli con reload
```

4.2. 使用新设置重新启动连接。

```
[student@servera ~]$ sudo nmcli con up "Wired connection 1"
Connection successfully activated (D-Bus active path: /org/freedesktop/
NetworkManager/ActiveConnection/3)
```

4.3. 验证新 IP 地址。

```
[student@servera ~]$ ip addr show enX
2: enX: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
    default qlen 1000
        link/ether 52:54:00:00:fa:0a brd ff:ff:ff:ff:ff:ff
        inet 172.25.250.10/24 brd 172.25.250.255 scope global noprefixroute enX
            valid_lft forever preferred_lft forever
        inet 10.0.1.1/24 brd 10.0.1.255 scope global noprefixroute enX
            valid_lft forever preferred_lft forever
        inet6 fe80::4bf3:e1d9:3076:f8d7/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
```

4.4. 从 servera 退出，并以 student 用户身份返回到 workstation。

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

- 5. 在 `serverb` 上, 以 `student` 用户身份编辑 `/etc/sysconfig/network-scripts/ifcfg-Wired_connection_1` 文件, 以便增加一个地址 `10.0.1.2/24`, 然后加载新配置。

5.1. 从 `workstation`, 使用 `ssh` 命令以 `student` 用户身份登录 `serverb`。

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

5.2. 修改 `ifcfg-Wired_connection_1` 文件, 以添加第二个 IPv4 地址和网络前缀。

```
[student@serverb ~]$ echo "IPADDR2=10.0.1.2" | \
sudo tee -a /etc/sysconfig/network-scripts/ifcfg-Wired_connection_1
[sudo] password for student: student
IPADDR2=10.0.1.2
[student@serverb ~]$ echo "PREFIX2=24" | \
sudo tee -a /etc/sysconfig/network-scripts/ifcfg-Wired_connection_1
PREFIX2=24
```

5.3. 重新加载配置更改。

```
[student@serverb ~]$ sudo nmcli con reload
```

5.4. 使用新设置开启连接。

```
[student@serverb ~]$ sudo nmcli con up "Wired connection 1"
Connection successfully activated (D-Bus active path: /org/freedesktop/
NetworkManager/ActiveConnection/4)
```

5.5. 验证新 IP 地址。

```
[student@serverb ~]$ ip addr show enX
2: enX: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
    default qlen 1000
        link/ether 52:54:00:00:fa:0b brd ff:ff:ff:ff:ff:ff
        inet 172.25.250.11/24 brd 172.25.250.255 scope global noprefixroute enX
            valid_lft forever preferred_lft forever
        inet 10.0.1.2/24 brd 10.0.1.255 scope global noprefixroute enX
            valid_lft forever preferred_lft forever
        inet6 fe80::74c:3476:4113:463f/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
```

- 6. 使用新的网络地址测试连接。

6.1. 从 serverb, ping servera 的新地址。

```
[student@serverb ~]$ ping -c3 10.0.1.1
PING 10.0.1.1 (10.0.1.1) 56(84) bytes of data.
64 bytes from 10.0.1.1: icmp_seq=1 ttl=64 time=0.342 ms
64 bytes from 10.0.1.1: icmp_seq=2 ttl=64 time=0.188 ms
64 bytes from 10.0.1.1: icmp_seq=3 ttl=64 time=0.317 ms

--- 10.0.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 35ms
rtt min/avg/max/mdev = 0.188/0.282/0.342/0.068 ms
```

6.2. 从 serverb 退出，并返回到 workstation。

```
[student@serverb ~]$ exit
logout
Connection to serverb closed.
[student@workstation ~]$
```

6.3. 从 workstation，以 student 用户身份使用 ssh 命令访问 servera，从而 ping serverb 的新地址。

```
[student@workstation ~]$ ssh student@servera ping -c3 10.0.1.2
PING 10.0.1.2 (10.0.1.2) 56(84) bytes of data.
64 bytes from 10.0.1.2: icmp_seq=1 ttl=64 time=0.269 ms
64 bytes from 10.0.1.2: icmp_seq=2 ttl=64 time=0.338 ms
64 bytes from 10.0.1.2: icmp_seq=3 ttl=64 time=0.361 ms

--- 10.0.1.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 48ms
rtt min/avg/max/mdev = 0.269/0.322/0.361/0.044 ms
```

## 完成

在 workstation 上，运行 lab net-edit finish 脚本来结束本练习。

```
[student@workstation ~]$ lab net-edit finish
```

本引导式练习到此结束。

# 配置主机名和名称解析

## 培训目标

学完本节后，您应能够配置服务器的静态主机名及名称解析，并测试其结果。

## 更改系统主机名

**hostname** 命令显示或临时修改系统的完全限定主机名。

```
[root@host ~]# hostname  
host@example.com
```

可以在 **/etc/hostname** 文件中指定静态主机名。**hostnamectl** 命令用于修改此文件，也可用于查看系统的完全限定主机名的状态。如果此文件不存在，则主机名在接口被分配了 IP 地址时由反向 DNS 查询设定。

```
[root@host ~]# hostnamectl set-hostname host@example.com  
[root@host ~]# hostnamectl status  
  Static hostname: host.example.com  
            Icon name: computer-vm  
      Chassis: vm  
   Machine ID: 73ab164e278e48be9bf80e80714a8cd5  
    Boot ID: 6b1cbc4177164ef58c0e9ed4adb2904f  
Virtualization: kvm  
Operating System: Red Hat Enterprise Linux 8.0 beta (0otpa)  
      CPE OS Name: cpe:/o:redhat:enterprise_linux:8.0:beta  
        Kernel: Linux 4.18.0-60.el8.x86_64  
  Architecture: x86-64  
[root@host ~]# cat /etc/hostname  
host@example.com
```



### 重要

在红帽企业 Linux 7 及更高版本中，静态主机名存储在 **/etc/hostname** 中。红帽企业 Linux 6 和更早版本将主机名存储为 **/etc/sysconfig/network** 文件中的变量。

## 配置名称解析

根解析器用于将主机名称转换为 IP 地址，反之亦可。它将根据 **/etc/nsswitch.conf** 文件的配置来确定查找位置。默认情况下，先检查 **/etc/hosts** 文件的内容。

```
[root@host ~]# cat /etc/hosts
127.0.0.1      localhost localhost.localdomain localhost4 localhost4.localdomain4
::1            localhost localhost.localdomain localhost6 localhost6.localdomain6

172.25.254.254 classroom.example.com
172.25.254.254 content.example.com
```

可以通过 **getent hosts hostname** 命令，利用 **/etc/hosts** 文件测试主机名解析。

如果在 **/etc/hosts** 文件中未找到条目，默认情况下，根解析器会尝试使用 DNS 名称服务器来查询主机名。**/etc/resolv.conf** 文件控制如何执行这一查询：

- **search**: 对于较短主机名尝试搜索的域名列表。不应在同一文件中设置此参数和 **domain**，如果在同一文件中设置它们，将使用最后一个实例。有关详细信息，请参见 **resolv.conf(5)**。
- **nameserver**: 要查询的名称服务器的 IP 地址。可以指定最多三个名称服务器指令，以在其中一个名称服务器停机时提供备用名称服务器。

```
[root@host ~]# cat /etc/resolv.conf
# Generated by NetworkManager
domain example.com
search example.com
nameserver 172.25.254.254
```

NetworkManager 将使用连接配置文件中的 DNS 设置更新 **/etc/resolv.conf** 文件。使用 **nmcli** 修改连接。

```
[root@host ~]# nmcli con mod ID ipv4.dns IP
[root@host ~]# nmcli con down ID
[root@host ~]# nmcli con up ID
[root@host ~]# cat /etc/sysconfig/network-scripts/ifcfg-ID
...output omitted...
DNS1=8.8.8.8
...output omitted...
```

**nmcli con mod ID ipv4.dns IP** 的默认行为是将任何旧的 DNS 设置替换为提供的新 IP 列表。**ipv4.dns** 参数前面的 + 或 - 符号可添加或删除个别条目。

```
[root@host ~]# nmcli con mod ID +ipv4.dns IP
```

要将 IPv6 地址为 **2001:4860:4860::8888** 的 DNS 服务器添加到要与 **static-ens3** 连接一起使用的名称服务器的列表：

```
[root@host ~]# nmcli con mod static-ens3 +ipv6.dns 2001:4860:4860::8888
```



### 注意

静态 IPv4 和 IPv6 DNS 最后都成为 **/etc/resolv.conf** 中的 **nameserver** 指令。您应确保至少列出一个可访问的 IPv4 名称服务器（假设是双栈系统）。最好至少有一个使用 IPv4 的名称服务器以及使用 IPv6 的第二服务器，以免您的 IPv4 或 IPv6 网络发生网络问题。

## 测试 DNS 名称解析

可以使用 **host HOSTNAME** 命令测试 DNS 服务器连接。

```
[root@host ~]# host classroom.example.com
classroom.example.com has address 172.25.254.254
[root@host ~]# host 172.25.254.254
254.25.25.172.in-addr.arpa domain name pointer classroom.example.com.
```



### 重要

DHCP 会在接口启动时自动重写 **/etc/resolv.conf** 文件，除非您在相关的接口配置文件中指定了 **PEERDNS=no**。使用 **nmcli** 命令设置此项。

```
[root@host ~]# nmcli con mod "static-ens3" ipv4.ignore-auto-dns yes
```



### 参考文献

**nmcli(1)**、**hostnamectl(1)**、**hosts(5)**、**getent(1)**、**host(1)** 和 **resolv.conf(5)** man page

如需更多信息，请参阅红帽企业 Linux 8.0中的配置和管理网络，网址为：

[https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/8/html-single/configuring\\_and\\_managing\\_networking/](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/configuring_and_managing_networking/)

## ▶ 指导练习

# 配置主机名和名称解析

在本练习中，您将手动配置系统的静态主机名、**/etc/hosts** 文件和 DNS 名称解析器。

## 成果

您应能够设置自定义主机名并配置名称解析设置。

## 在你开始之前

在 workstation 上，以 student 用户身份并使用密码 student 进行登录。

从 workstation，运行 **lab net-hostnames start** 命令。该命令将运行一个起始脚本，它将确定主机 servera 是否可从网络访问。

```
[student@workstation ~]$ lab net-hostnames start
```

- 1. 使用 **ssh** 命令，以 student 用户身份登录 servera。系统已配置为使用 SSH 密钥来进行身份验证，因此不需要输入密码就能登录 servera。

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

- 2. 查看当前的主机名设置。

2.1. 显示当前的主机名。

```
[student@servera ~]$ hostname  
servera.lab.example.com
```

2.2. 显示主机名状态。

```
[student@servera ~]$ hostnamectl status  
Static hostname: n/a  
Transient hostname: servera.lab.example.com  
Icon name: computer-vm  
Chassis: vm  
Machine ID: 73ab164e278e48be9bf80e80714a8cd5  
Boot ID: 76b13a300c944ab49445af778cb8f749  
Virtualization: kvm  
Operating System: Red Hat Enterprise Linux 8.0 (Ootpa)  
CPE OS Name: cpe:/o:redhat:enterprise_linux:8.0:GA  
Kernel: Linux 4.18.0-80.el8.x86_64  
Architecture: x86-64
```

► 3. 设置静态主机名，以便和当前的临时主机名匹配。

3.1. 更改主机名和主机名配置文件。

```
[student@servera ~]$ sudo hostnamectl set-hostname servera.lab.example.com  
[sudo] password for student: student  
[student@servera ~]$
```

3.2. 查看在网络启动时提供主机名的配置文件。

```
[student@servera ~]$ cat /etc/hostname  
servera.lab.example.com
```

3.3. 显示主机名状态。

```
[student@servera ~]$ hostnamectl status  
Static hostname: servera.lab.example.com  
        Icon name: computer-vm  
        Chassis: vm  
      Machine ID: 73ab164e278e48be9bf80e80714a8cd5  
        Boot ID: 76b13a300c944ab49445af778cb8f749  
Virtualization: kvm  
Operating System: Red Hat Enterprise Linux 8.0 (Ootpa)  
      CPE OS Name: cpe:/o:redhat:enterprise_linux:8.0:GA  
        Kernel: Linux 4.18.0-80.el8.x86_64  
      Architecture: x86-64
```

► 4. 临时更改主机名。

4.1. 更改主机名。

```
[student@servera ~]$ sudo hostname testname
```

4.2. 显示当前的主机名。

```
[student@servera ~]$ hostname  
testname
```

4.3. 查看在网络启动时提供主机名的配置文件。

```
[student@servera ~]$ cat /etc/hostname  
servera.lab.example.com
```

4.4. 重新启动系统。

```
[student@servera ~]$ sudo systemctl reboot  
Connection to servera closed by remote host.  
Connection to servera closed.  
[student@workstation ~]$
```

4.5. 从 workstation，以 student 用户身份登录 servera

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

4.6. 显示当前的主机名。

```
[student@servera ~]$ hostname  
servera.lab.example.com
```

► 5. 为课堂服务器添加一个本地别名。

5.1. 查询 classroom.example.com 的 IP 地址。

```
[student@servera ~]$ host classroom.example.com  
classroom.example.com has address 172.25.254.254
```

5.2. 修改 **/etc/hosts**, 以便附加名称 **class** 可用于访问 IP 地址 172.25.254.254。

```
[student@servera ~]$ sudo vim /etc/hosts  
[student@servera ~]$ cat /etc/hosts  
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4  
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6  
  
172.25.254.254 classroom.example.com classroom class  
172.25.254.254 content.example.com content  
...content omitted...
```

5.3. 查找 **class** 的 IP 地址。

```
[student@servera ~]$ host class  
Host class not found: 2(SERVFAIL)  
[student@servera ~]$ getent hosts class  
172.25.254.254 classroom.example.com class
```

5.4. ping **class**。

```
[student@servera ~]$ ping -c3 class  
PING classroom.example.com (172.25.254.254) 56(84) bytes of data.  
64 bytes from classroom.example.com (172.25.254.254): icmp_seq=1 ttl=64 time=0.397 ms  
64 bytes from classroom.example.com (172.25.254.254): icmp_seq=2 ttl=64 time=0.447 ms  
64 bytes from classroom.example.com (172.25.254.254): icmp_seq=3 ttl=64 time=0.470 ms  
  
--- classroom.example.com ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2000ms  
rtt min/avg/max/mdev = 0.397/0.438/0.470/0.030 ms
```

5.5. 从 **servera** 退出。

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

## 完成

在 workstation 上，运行 **lab net-hostnames finish** 脚本来结束本练习。

```
[student@workstation ~]$ lab net-hostnames finish
```

本引导式练习到此结束。

## ▶ 开放研究实验

# 管理网络

## 任务执行清单

在本实验中，您将在红帽企业 Linux 服务器上配置网络设置。

## 成果

您应能够为主网络接口配置两个静态 IPv4 地址。

## 在你开始之前

在 **workstation** 上，以 **student** 用户身份并使用密码 **student** 进行登录。

从 **workstation**，运行 **lab net-review start** 命令。该命令将运行一个起始脚本，它将确定主机 **serverb** 是否可从网络访问。

```
[student@workstation ~]$ lab net-review start
```

1. 使用 **ssh** 命令，以 **student** 用户身份登录 **serverb**。系统已配置为使用 SSH 密钥来进行身份验证，因此不需要输入密码就能登录 **serverb**。
2. 使用 **sudo -i** 命令，切换为 **root** 用户。收到提示时，使用 **student** 作为密码。
3. 利用表格中的设置，创建一个使用静态网络连接的新连接。

参数	设置
连接名称	lab
接口名称	enX (可能会有所不同，请使用 MAC 地址为 52:54:00:00:fa:0b 的接口)
IP 地址	172.25.250.11/24
网关地址	172.25.250.254
DNS 地址	172.25.250.254

4. 将新连接配置为自动启动。其他连接则不应自动启动。
5. 修改新连接，使其也使用地址 10.0.1.1/24。
6. 配置 **hosts** 文件，使 10.0.1.1 可被引用为 **private**。
7. 重新启动系统。
8. 从 **workstation**，使用 **ping** 命令验证 **serverb** 是否已初始化。

## 评估

在 **workstation** 上，运行 **lab net-review grade** 脚本来确认是否成功完成本实验。

```
[student@workstation ~]$ lab net-review grade
```

## 完成

在 workstation 上，运行 **lab net-review finish** 脚本来结束本实验。

```
[student@workstation ~]$ lab net-review finish
```

本实验到此结束。

## ▶ 解决方案

# 管理网络

### 任务执行清单

在本实验中，您将在红帽企业 Linux 服务器上配置网络设置。

### 成果

您应能够为主网络接口配置两个静态 IPv4 地址。

### 在你开始之前

在 workstation 上，以 student 用户身份并使用密码 student 进行登录。

从 workstation，运行 **lab net-review start** 命令。该命令将运行一个起始脚本，它将确定主机 serverb 是否可从网络访问。

```
[student@workstation ~]$ lab net-review start
```

1. 使用 **ssh** 命令，以 student 用户身份登录 serverb。系统已配置为使用 SSH 密钥来进行身份验证，因此不需要输入密码就能登录 serverb。

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

2. 使用 **sudo -i** 命令，切换为 root 用户。收到提示时，使用 student 作为密码。

```
[student@serverb ~]$ sudo -i
[sudo] password for student: student
[root@serverb ~]#
```

3. 利用表格中的设置，创建一个使用静态网络连接的新连接。

参数	设置
连接名称	lab
接口名称	enX (可能会有所不同，请使用 MAC 地址 52:54:00:00:fa:0b 的接口)
IP 地址	172.25.250.11/24
网关地址	172.25.250.254
DNS 地址	172.25.250.254

确定接口名称及当前活动连接的名称。本解决方案假定接口名称为 `enX`, 而连接名称为 `Wired connection 1`。

```
[root@serverb ~]# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    qlen 1000
        link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enX: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
    default qlen 1000
        link/ether 52:54:00:00:fa:0b brd ff:ff:ff:ff:ff:ff
[root@serverb ~]# nmcli con show --active
NAME           UUID                                  TYPE      DEVICE
Wired connection 1  03da038a-3257-4722-a478-53055cc90128  ethernet  enX
```

根据说明中所述的表中的信息，创建新的 `lab` 连接配置文件。将配置文件与上一个 `ip link` 命令的输出中所列的网络接口名称相关联。

```
[root@serverb ~]# nmcli con add con-name lab ifname enX type ethernet \
    ipv4.method manual \
    ipv4.address 172.25.250.11/24 ipv4.gateway 172.25.250.254
[root@serverb ~]# nmcli con mod "lab" ipv4.dns 172.25.250.254
```

4. 将新连接配置为自动启动。其他连接则不应自动启动。

```
[root@serverb ~]# nmcli con mod "lab" connection.autoconnect yes
[root@serverb ~]# nmcli con mod "Wired connection 1" connection.autoconnect no
```

5. 修改新连接，使其也使用地址 10.0.1.1/24。

```
[root@serverb ~]# nmcli con mod "lab" +ipv4.addresses 10.0.1.1/24
```

或者也可：

```
[root@serverb ~]# echo "IPADDR1=10.0.1.1" \
>> /etc/sysconfig/network-scripts/ifcfg-lab
[root@serverb ~]# echo "PREFIX1=24" >> /etc/sysconfig/network-scripts/ifcfg-lab
```

6. 配置 `hosts` 文件，使 10.0.1.1 可被引用为 `private`。

```
[root@serverb ~]# echo "10.0.1.1 private" >> /etc/hosts
```

7. 重新启动系统。

```
[root@serverb ~]# systemctl reboot
Connection to serverb closed by remote host.
Connection to serverb closed.
[student@workstation ~]$
```

8. 从 **workstation**, 使用 **ping** 命令验证 **serverb** 是否已初始化。

```
[student@workstation ~]$ ping -c3 serverb
PING serverb.lab.example.com (172.25.250.11) 56(84) bytes of data.
64 bytes from serverb.lab.example.com (172.25.250.11): icmp_seq=1 ttl=64
time=0.478 ms
64 bytes from serverb.lab.example.com (172.25.250.11): icmp_seq=2 ttl=64
time=0.504 ms
64 bytes from serverb.lab.example.com (172.25.250.11): icmp_seq=3 ttl=64
time=0.513 ms

--- serverb.lab.example.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 78ms
rtt min/avg/max/mdev = 0.478/0.498/0.513/0.023 ms
[student@workstation ~]$
```

## 评估

在 **workstation** 上, 运行 **lab net-review grade** 脚本来确认是否成功完成本实验。

```
[student@workstation ~]$ lab net-review grade
```

## 完成

在 **workstation** 上, 运行 **lab net-review finish** 脚本来结束本实验。

```
[student@workstation ~]$ lab net-review finish
```

本实验到此结束。

## 总结

---

在本章中，您学到了：

- TCP/IP 网络模型是一种简化的四层抽象集合，用于描述不同的协议如何进行互操作，以便计算机通过互联网将流量从一台计算机发送到另一台计算机。
- IPv4 是当今互联网上使用的主要网络协议。IPv6 旨在最终取代 IPv4 网络协议。默认情况下，红帽企业 Linux 以双栈模式运行，并行使用两种协议。
- NetworkManager 是监控和管理网络配置的守护进程。
- **nmcli** 命令是一个用于通过 NetworkManager 来配置网络设置的命令行工具。
- 系统的静态主机名存储在 **/etc/hostname** 文件中。**hostnamectl** 命令用于修改或查看系统的主机名和相关设置的状态。**hostname** 命令显示或临时修改系统的主机名。

海量视频题库 myitpub.com QQ:5565462  
www.52myit.com

## 章 13

# 归档和传输文件

### 目标

将文件归档，并从一个系统复制文件到另一系统。

### 培训目标

- 使用 **tar** 将文件和目录归档到压缩文件中，以及提取现有 **tar** 存档的内容。
- 通过 SSH，与远程系统安全地来回传输文件。
- 将本地文件或目录的内容与远程服务器上的副本同步。

### 章节

- 管理压缩的 **tar** 存档（及引导式练习）
- 在系统之间安全地传输文件（及引导式练习）
- 在系统之间安全地同步文件（及引导式练习）

### 实验

归档和传输文件

# 管理压缩的 TAR 存档

---

## 培训目标

学完本节后，您应能够使用 **tar** 将文件和目录归档到压缩文件中，还能提取现有 **tar** 存档的内容。

### tar 命令

创建备份和通过网络传输数据时，归档和压缩文件非常有用。用来创建和使用备份存档的其中一个最早也是最常见的命令是 **tar** 命令。

通过 **tar** 命令，用户可以将大型文件集汇集为一个文件（存档）。**tar** 存档是一个结构化的文件数据序列，其中包含有关每个文件和索引的元数据，以便可以提取单个文件。该存档可以使用 **gzip**、**bzip2** 或 **xz** 压缩方式进行压缩。

**tar** 命令能够列出存档内容，或者将其文件提取到当前系统。

### 所选的 tar 选项

**tar** 命令选项划分成不同的操作（您想要进行的操作），其中包括一般选项和压缩选项。下表显示了常用选项、选项的长版本及其说明：

#### tar 操作概述

选项	描述
<b>-c</b> 、 <b>--create</b>	创建一个新存档。
<b>-x</b> 、 <b>--extract</b>	从现有存档提取。
<b>-t</b> 、 <b>--list</b>	列出存档的目录。

#### 所选的 tar 一般选项

选项	描述
<b>-v</b> 、 <b>--verbose</b>	详细信息。显示存档或提取的文件有哪些。
<b>-f</b> 、 <b>--file=</b>	文件名。此选项必须后接要使用或创建的存档的文件名。
<b>-p</b> 、 <b>--preserve-permissions</b>	在提取存档时保留文件和目录的权限，而不去除 umask。

### tar 压缩选项概述

选项	描述
<b>-z、--gzip</b>	使用 gzip 压缩方式 ( <b>.tar.gz</b> )。
<b>-j、--bzip2</b>	使用 bzip2 压缩方式 ( <b>.tar.bz2</b> )。bzip2 的压缩率通常比 gzip 高。
<b>-J、--xz</b>	使用 xz 压缩方式 ( <b>.tar.xz</b> )。xz 的压缩率通常比 bzip2 更高。

## 列出 tar 命令的选项

**tar** 命令需要以下三个选项之一：

- 使用 **-c** 或 **--create** 选项来创建存档。
- 使用 **-t** 或 **--list** 选项来列出存档的内容。
- 使用 **-x** 或 **--extract** 选项来提取存档。

其他常用选项包括：

- 使用 **-f** 或 **--file=** 选项加上文件名，作为要运行的存档的参数。
- 使用 **-v** 或 **--verbose** 选项以获取详细信息；用于查看添加到存档中或从中提取的文件有哪些。



#### 注意

**tar** 命令实际上支持第三种旧的选项样式，它使用标准单字母选项，不带前导 **-**。这种选项仍然较为常见，在使用其他人的指令或命令时，可能会遇到该语法。借助 **info tar 'old options'** 命令可以详细了解它与普通短选项的区别。

您可以暂时忽略旧的选项，并专注于标准的短选项和长选项语法。

## 归档文件和目录

创建新存档时要使用的第一项为 **c** 选项，后跟 **f** 选项，接着是一个空格，然后是要创建的存档的文件名，最后是应当添加到该存档中的文件和目录列表。存档会创建在当前目录中，除非另外指定。



#### 警告

在创建 **tar** 存档之前，请先验证目录中没有其他存档与要创建的新存档名称相同。**tar** 命令将覆盖现有的存档，而不提供警告。

以下命令创建名为 **archive.tar** 的存档，其内容为用户主目录中的 **file1**、**file2** 和 **file3**。

```
[user@host ~]$ tar -cf archive.tar file1 file2 file3
[user@host ~]$ ls archive.tar
archive.tar
```

以上 **tar** 命令也可以使用长版本选项执行。

```
[user@host ~]$ tar --file=archive.tar --create file1 file2 file3
```

**注意**

在使用绝对路径名归档文件时，将默认从文件名中删除该路径中的前面的 / 符号。删除路径中的前导 / 可帮助用户在提取存档时避免覆盖重要文件。**tar** 命令相对于当前的工作目录提取文件。

要使 **tar** 能够归档选定的文件，执行 **tar** 命令的用户必须要可以读取这些文件。例如，为 **/etc** 文件夹及其所有内容创建新存档需要 **root** 特权，因为只有 **root** 用户才可读取 **/etc** 目录的所有文件。非特权用户可以创建 **/etc** 目录的存档，但是该存档将忽略用户没有读取权限的文件，并且将忽略用户没有读取和执行权限的目录。

要以 **root** 身份创建名为 **/root/etc.tar** 且内容为 **/etc** 目录的 **tar** 存档：

```
[root@host ~]# tar -cf /root/etc.tar /etc
tar: Removing leading `/' from member names
[root@host ~]#
```

**重要**

在本课程中未涉及的部分高级权限（如 ACL 和 SELinux 上下文）不会自动存储到 **tar** 存档中。在创建存档时使用 **--xattrs** 选项，以将这些扩展属性存储到 **tar** 存档中。

## 列出存档的内容

**t** 选项指示 **tar** 列出存档的内容（目录，所以是 **t**）。使用 **f** 选项，加上要查询的存档的名称。例如：

```
[root@host ~]# tar -tf /root/etc.tar
etc/
etc/fstab
etc/crypttab
etc/mtab
...output omitted...
```

## 从存档中提取文件

**tar** 存档通常应当提取到空目录中，以确保它不会覆盖任何现有的文件。当 **root** 提取存档时，**tar** 命令会保留文件的原始用户和组所有权。如果普通用户使用 **tar** 提取文件，文件所有权将属于从存档中提取文件的用户。

要将 **/root/etc.tar** 存档中的文件恢复到 **/root/etcbackup** 目录，可运行：

```
[root@host ~]# mkdir /root/etcbackup
[root@host ~]# cd /root/etcbackup
[root@host etcbackup]# tar -tf /root/etc.tar
etc/
etc/fstab
etc/crypttab
```