

RHCE 系列（一）：如何设置和测试静态网络路由

2015-10-22 12:37 评论: 4 收藏: 7

参考原文：<http://www.tecmint.com/how-to...>

作者：Gabriel Cánepa

编译文章：LCTT <https://linux.cn/article-6451-1.html>

译者：ictlyh

RHCE（Red Hat Certified Engineer，红帽认证工程师）是红帽公司的一个认证，红帽向企业社区贡献开源操作系统和软件，同时它还给公司提供训练、支持和咨询服务。



RHCE 考试准备指南

这个 RHCE 是一个绩效考试（代号 EX300），面向那些拥有更多的技能、知识和能力的红帽企业版 Linux（RHEL）系统高级系统管理员。

重要：获得RHCE 认证前需要先

有 Red Hat Certified System Administrator, RHCSA 红帽认证系统管理员认证 <<https://linux.cn/article-6133-1.html>> 。

以下是基于红帽企业版 Linux 7 考试的考试目标，我们会在该 RHCE 系列中分别介绍：

- 第一部分：如何在 RHEL 7 中设置和测试静态路由 <<https://linux.cn/article-6451-1.html>>
- 第二部分：如何进行包过滤、网络地址转换和设置内核运行时参数 <<https://linux.cn/article-6458-1.html>>
- 第三部分：如何使用 Linux 工具集产生和发送系统活动报告

- 第四部分：使用 Shell 脚本进行自动化系统维护
- 第五部分：如何在 RHEL 7 中管理系统日志（配置、轮换和导入到数据库）
- 第六部分：设置 Samba 服务器并配置 FirewallD 和 SELinux 支持客户端文件共享
- 第七部分：设置 NFS 服务器及基于 Kerberos 认证的客户端
- 第八部分：在 Apache 上使用网络安全服务（NSS）通过 TLS 提供 HTTPS 服务
- 第九部分：如何使用无客户端配置来设置 Postfix 邮件服务器（SMTP）
- 第十部分：在 RHEL/CentOS 7 中设置网络时间协议（NTP）服务器
- 第十一部分：如何配置一个只缓存的 DNS 服务器

在你的国家查看考试费用和注册考试，可以到 [RHCE 认证 <https://www.redhat.com/en/services/certification/rhce>](https://www.redhat.com/en/services/certification/rhce) 网页。

在 RHCE 的第一和第二部分，我们会介绍一些基本的但典型的情形，也就是静态路由原理、包过滤和网络地址转换。



RHCE 系列第一部分：设置和测试网络静态路由

请注意我们不会作深入的介绍，但以这种方式组织内容能帮助你开始第一步并继续后面的内容。

红帽企业版 Linux 7 中的静态路由

现代网络的一个奇迹就是有很多可用设备能将一组计算机连接起来，不管是在一个房间里少量的机器还是在一栋建筑物、城市、国家或者大洲之间的多台机器。

然而，为了能在任意情形下有效的实现这些，需要对网络包进行路由，或者换句话说，它们从源到目的地的路径需要按照某种规则。

静态路由是为网络包指定一个路由的过程，而不是使用网络设备提供的默认网关。除非另

有指定静态路由，网络包会被导向默认网关；而静态路由则基于预定义标准所定义的其它路径，例如数据包目的地。

我们在该篇指南中会考虑以下场景。我们有一台 RHEL 7，连接到 1号路由器 [192.168.0.1] 以访问因特网以及 192.168.0.0/24 中的其它机器。

第二个路由器（2号路由器）有两个网卡：enp0s3 同样连接到路由器1号以访问互联网，及与 RHEL 7 和同一网络中的其它机器通讯，另外一个网卡（enp0s8）用于授权访问内部服务所在的 10.0.0.0/24 网络，例如 web 或数据库服务器。

该场景可以用下面的示意图表示：



静态路由网络示意图

在这篇文章中我们会集中介绍在 RHEL 7 中设置路由表，确保它能通过1号路由器访问因特网以及通过2号路由器访问内部网络。

在 RHEL 7 中，你可以通过命令行用 `ip` 命令 [<http://www.tecmint.com/ip-command-examples/>](http://www.tecmint.com/ip-command-examples/) 配置和显示设备和路由。这些更改能在运行的系统中及时生效，但由于重启后不会保存，我们会使用 `/etc/sysconfig/network-scripts` 目录下的 `ifcfg-enp0sX` 和 `route-enp0sX` 文件永久保存我们的配置。

首先，让我们打印出当前的路由表：

```
# ip route show
```

```
[root@rhel7 ~]# ip route show
default via 192.168.0.1 dev enp0s3
169.254.0.0/16 dev enp0s3 scope link metric 1002
192.168.0.0/24 dev enp0s3 proto kernel scope link src 192.168.0.18
[root@rhel7 ~]#
```

检查当前路由表

从上面的输出中，我们可以得出以下结论：

- 默认网关的 IP 是 192.168.0.1，可以通过网卡 enp0s3 访问。
- 系统启动的时候，它启用了到 169.254.0.0/16 的 zeroconf 路由（只是在本例

中)。也就是说，如果机器设置通过 DHCP 获取 IP 地址，但是由于某些原因失败了，它就会在上述网段中自动分配到一个地址。这一行的意思是，该路由会允许我们通过 enp0s3 和其它没有从 DHCP 服务器中成功获得 IP 地址的机器机器相连接。

- 最后，但同样重要的是，我们也可以通过 IP 地址是 192.168.0.18 的 enp0s3 与 192.168.0.0/24 网络中的其它机器连接。

下面是这样的配置中你需要做的一些典型任务。除非另有说明，下面的任务都在2号路由器上进行。

确保正确安装了所有网卡：

```
# ip link show
```

如果有某块网卡停用了，启动它：

```
# ip link set dev enp0s8 up
```

分配 10.0.0.0/24 网络中的一个 IP 地址给它：

```
# ip addr add 10.0.0.17 dev enp0s8
```

噢！我们分配了一个错误的 IP 地址。我们需要删除之前分配的那个并添加正确的地址（10.0.0.18）：

```
# ip addr del 10.0.0.17 dev enp0s8  
# ip addr add 10.0.0.18 dev enp0s8
```

现在，请注意你只能添加一个通过网关到目标网络的路由，网关需要可以访问到。因为这个原因，我们需要在 192.168.0.0/24 范围中给 enp0s3 分配一个 IP 地址，这样我们的 RHEL 7 才能连接到它：

```
# ip addr add 192.168.0.19 dev enp0s3
```

最后，我们需要启用包转发：

```
# echo "1" > /proc/sys/net/ipv4/ip_forward
```

并停用/取消防火墙（从现在开始，直到下一篇文章中我们介绍了包过滤）：

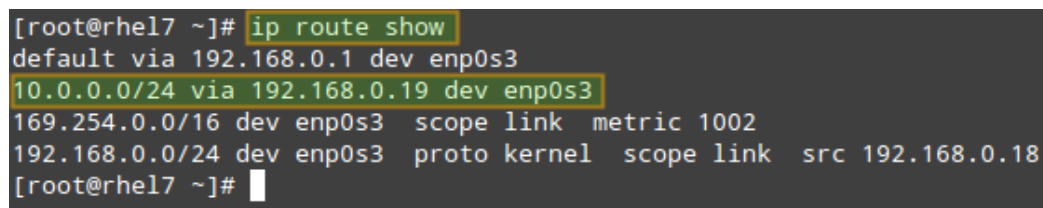
```
# systemctl stop firewalld
# systemctl disable firewalld
```

回到我们的 RHEL 7（192.168.0.18），让我们配置一个通过 192.168.0.19（2号路由器的 enp0s3）到 10.0.0.0/24 的路由：

```
# ip route add 10.0.0.0/24 via 192.168.0.19
```

之后，路由表看起来像下面这样：

```
# ip route show
```



```
[root@rhel7 ~]# ip route show
default via 192.168.0.1 dev enp0s3
10.0.0.0/24 via 192.168.0.19 dev enp0s3
169.254.0.0/16 dev enp0s3 scope link metric 1002
192.168.0.0/24 dev enp0s3 proto kernel scope link src 192.168.0.18
[root@rhel7 ~]#
```

确认网络路由表

同样，在你尝试连接的 10.0.0.0/24 网络的机器中添加对应的路由：

```
# ip route add 192.168.0.0/24 via 10.0.0.18
```

你可以使用 ping 测试基本连接：

在 RHEL 7 中运行：

```
# ping -c 4 10.0.0.20
```

10.0.0.20 是 10.0.0.0/24 网络中一个 web 服务器的 IP 地址。

在 web 服务器（10.0.0.20）中运行

```
# ping -c 192.168.0.18
```

192.168.0.18 也就是我们的 RHEL 7 机器的 IP 地址。

另外，我们还可以使用 `tcpdump` <<http://www.tecmint.com/12-tcpdump-commands-a-network-sniffer-tool/>>（需要通过 `yum install tcpdump` 安装）来检查我们 RHEL 7 和 10.0.0.20 中 web 服务器之间的 TCP 双向通信。

首先在第一台机器中启用日志：

```
# tcpdump -qnnvvv -i enp0s3 host 10.0.0.20
```

在同一个系统上的另一个终端，让我们通过 telnet 连接到 web 服务器的 80 号端口（假设 Apache 正在监听该端口；否则应在下面命令中使用正确的监听端口）：

```
# telnet 10.0.0.20 80
```

tcpdump 日志看起来像下面这样：

```
[root@rhel7 ~]# tcpdump -qnnvvv -i enp0s3 host 10.0.0.20
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture size 65535 bytes
23:12:22.682531 IP (tos 0x10, ttl 64, id 61663, offset 0, flags [DF], proto TCP (6), length 60)
    192.168.0.18.36676 > 10.0.0.20.80: tcp 0
23:12:22.683875 IP (tos 0x0, ttl 63, id 0, offset 0, flags [DF], proto TCP (6), length 60)
    10.0.0.20.80 > 192.168.0.18.36676: tcp 0
23:12:22.683965 IP (tos 0x10, ttl 64, id 61664, offset 0, flags [DF], proto TCP (6), length 52)
```

检查服务器之间的网络连接

通过查看我们 RHEL 7（192.168.0.18）和 web 服务器（10.0.0.20）之间的双向通信，可以看出已经正确地初始化了连接。

请注意你重启系统后会丢失这些更改。如果你想把它们永久保存下来，你需要在我们运行上面的命令的相同系统中编辑（如果不存在的话就创建）以下的文件。

尽管对于我们的测试例子不是严格要求，你需要知道 `/etc/sysconfig/network` 包含了一些系统范围的网络参数。一个典型的 `/etc/sysconfig/network` 看起来类似下面这样：

```
# Enable networking on this system?
NETWORKING=yes
# Hostname. Should match the value in /etc/hostname
HOSTNAME=yourhostnamehere
```

```
# Default gateway
GATEWAY=XXX.XXX.XXX.XXX

# Device used to connect to default gateway. Replace X with the appropriate
number.
GATEWAYDEV=enp0sX
```

当需要为每个网卡设置特定的变量和值时（正如我们在2号路由器上面做的），你需要编辑 `/etc/sysconfig/network-scripts/ifcfg-enp0s3` 和 `/etc/sysconfig/network-scripts/ifcfg-enp0s8` 文件。

下面是我们的例子，

```
TYPE=Ethernet
BOOTPROTO=static
IPADDR=192.168.0.19
NETMASK=255.255.255.0
GATEWAY=192.168.0.1
NAME=enp0s3
ONBOOT=yes
```

以及

```
TYPE=Ethernet
BOOTPROTO=static
IPADDR=10.0.0.18
NETMASK=255.255.255.0
GATEWAY=10.0.0.1
NAME=enp0s8
ONBOOT=yes
```

其分别对应 `enp0s3` 和 `enp0s8`。

由于要为我们的客户端机器(192.168.0.18)进行路由，我们需要编辑 `/etc/sysconfig/network-scripts/route-enp0s3`：

```
10.0.0.0/24 via 192.168.0.19 dev enp0s3
```

现在 **reboot** 你的系统，就可以在路由表中看到该路由规则。

总结

在这篇文章中我们介绍了红帽企业版 Linux 7 的静态路由。尽管场景可能不同，这里介绍的例子说明了所需的原理以及进行该任务的步骤。结束之前，我还建议你看一下

The Linux Documentation Project
Linux 文档项目 <<http://www.tldp.org/>> 网站上的

《
Securing Linux and Optimizing
安全加固和优化 Linux <<http://www.tldp.org/LDP/solrhe/Securing-Optimizing-Linux-RH-Edition-v1.3/net-manage.html>>，以了解这里介绍主题的更详细内容。

在下篇文章中我们会介绍数据包过滤和网络地址转换，结束 RHCE 验证需要的网络基本技巧。

如往常一样，我们期望听到你的回复，用下面的表格留下你的疑问、评论和建议吧。

via: <http://www.tecmint.com/how-to-setup-and-configure-static-network-routing-in-rhel/> <<http://www.tecmint.com/how-to-setup-and-configure-static-network-routing-in-rhel/>>

作者：Gabriel Cánepa <<http://www.tecmint.com/author/gacanepa/>> 译者：ictlyh
<<https://github.com/ictlyh>> 校对：wxy <<https://github.com/wxy>>

本文由 LCTT <<https://github.com/LCTT/TranslateProject>> 原创翻译，Linux中国
<<https://linux.cn/>> 荣誉推出