Red Hat Enterprise Linux 7

系统管理员指南

RHEL 7 的部署、配置和管理

Last Updated: 2024-08-18

Red Hat Enterprise Linux 7 系统管理员指南

RHEL 7 的部署、配置和管理

法律通告

摘要

系统管理员指南介绍了 有关 Red Hat Enterprise Linux 7 部署、配置和管理相关信息。它面向对系统有基本了解的系统管理员。 为提高您的经验，您可能还对 Red Hat System Administration II (RH134)、Red Hat System Administration the、RH254） 或实验室 工程师 (RH199) 培训感兴趣。 如果要将 Red Hat Enterprise Linux 7 与 Linux 容器功能搭配使用，请参阅 Red Hat Enterprise Linux Atomic Host 产品文档。有关 Red Hat Enterprise Linux 7 的一般 Linux 容器概念及其当前功能的概述，请参阅 Red Hat Systems 中的容器概述。 Red Hat Enterprise Linux Atomic Host 7 Managing Containers 指南介绍了与容器管理和管理相关的主题。

目录

部分 I. 基本系统配置

这部分涵盖了基本的安装后任务和基本系统管理任务，如键盘配置、日期和时间配置、管理用户和组以及获取特权。

第 1 章 开始使用

本章论述了安装 Red Hat Enterprise Linux 7 后您可能需要执行的基本任务。

请注意，这些项目可能包括在安装过程中通常已经完成的任务，但不一定必须完成，比如注册系统。处理这些任务的子章节提供了如何在安装过程中实现这个问题的简要概述，以及在特殊章节中查看相关文档的链接。

有关红帽企业 Linux 7 安装的详细信息，请参阅红帽企业 Linux 7 安装指南 。

注意

本章提到要执行的一些命令。**root** 用户输入的命令在提示符中具有 **#**，而常规用户可以执行的命令在其提示符中具有 **$**。

有关常见安装后任务的更多信息，请参阅 Red Hat Enterprise Linux 7 安装指南 。

虽然可以通过命令行来完成所有安装后任务，也可以使用 **Web 控制台** 工具来执行其中一些任务。

Web 控制台以及它可以使用什么任务

Web 控制台是一个 系统管理工具，提供通过 Web 浏览器监控和管理服务器的用户界面。

Web 控制台 可用于执行这些任务：

- 监控基本系统特性，如硬件、互联网连接或性能特性
- 分析系统日志文件的内容
- 配置基本网络功能，如接口、网络日志、数据包大小
- 管理用户帐户
- 监控并配置系统服务
- 创建诊断报告
- 设置内核转储配置
- 配置 SELinux
- 管理系统订阅
- 访问终端

有关安装和使用 **Web 控制台** 的详情，请参阅使用 RHEL 7 web 控制台管理系统。

1.1. 环境的基本配置

环境的基本配置包括：

- 日期和时间
- 系统区域设置
- 键盘布局

这些项目的设置通常是安装过程的一部分。

如需更多信息，请参阅根据安装方法划分的源：

- 当使用 Anaconda 安装程序安装时，请参考：
  Red Hat Enterprise Linux 7 安装指南中的日期&Time、语言支持和 键盘配置
- 使用 Kickstart 文件安装时，请参考：
  红帽企业 Linux 7 安装指南中的 Kickstart 命令和选项 .

如果您需要在安装后重新配置环境的基本特性，请按照本节中的内容进行操作。

1.1.1. 配置日期和时间简介

因为许多原因，保持准确的时间非常重要。在 Red Hat Enterprise Linux 7 中，**NTP** 协议保证了时间保持，该协议由用户空间运行的守护进程实施。用户空间守护进程更新内核中运行的系统时钟。系统时钟可以通过使用不同的时钟源来维护系统的时间。

Red Hat Enterprise Linux 7 使用以下守护进程来实现 **NTP** ：

- **chronyd**
  chronyd 守护进程默认使用。它包括在 **chrony** 软件包中。有关使用 **chronyd** 配置和使用 **NTP** 的详情请参考 第 18 章 *使用 chrony 套件配置 NTP* 。
- **ntpd**
  ntpd 守护进程可从 **ntp** 软件包获得。有关使用 **ntpd 配置和使用 NTP** 的详情请参考 第 19 章 *使用 ntpd 配置 NTP*。

如果要使用 **ntpd** 而不是默认的 **chronyd，则需要禁用 chronyd 、安装、启用和配置 ntpd，如 第 19 章 *使用 ntpd 配置 NTP* 所示。**

显示当前日期和时间

要显示当前的日期和时间，请使用以下命令之一：

```
~]$ date
```

```
~]$ timedatectl
```

请注意，timedatectl 命令提供更为详细的输出，包括通用时间、当前使用的时区、网络时间协议 (NTP)配置的状态，以及一些附加信息。

有关配置日期和时间的详情请参考 第 3 章 *配置日期和时间*。

1.1.2. 配置系统区域介绍

系统范围的区域设置保存在 /etc/locale.conf 文件中，该文件在早期引导时由 systemd 守护进程读取。每个服务或用户都会继承 /etc/locale.conf 中配置的区域设置，单独程序或个人用户均覆盖它们。

处理系统区域的基本任务：

- 列出可用的系统区域设置：

```
~]$ localectl list-locales
```

- 显示系统区域设置的当前状态：

```
~]$ localectl status
```

- 设置或更改默认系统区域设置：

```
~]# localectl set-locale LANG=locale
```

有关配置系统区域设置的详情请参考 第 2 章 *系统位置和键盘配置*。

1.1.3. 配置键盘布局简介

键盘布局设置控制文本控制台和图形用户界面中的布局。

处理键盘布局的基本任务包括：

- 列出可用的键映射：

```
~]$ localectl list-keymaps
```

- 显示 keymap 设置的当前状态：

```
~]$ localectl status
```

- 设置或更改默认系统键映射：

```
~]# localectl set-keymap
```

有关配置键盘布局的详情请参考 第 2 章 *系统位置和键盘配置*。

1.2. 配置和检查网络访问

网络访问通常在安装过程中配置。但是，安装过程不会提示您在一些常见安装路径中配置网络接口。因此，安装后可能不会配置网络访问。如果发生这种情况，您可以在安装后配置网络访问。

有关在安装过程中配置网络访问的快速入门，请参阅 第 1.2.1 节 “在安装过程中配置网络访问”。要在安装后配置网络访问，您可以使用 nmcli 命令行实用程序，如 Red Hat Enterprise Linux 7 网络指南 或 nmtui 文本用户界面实用程序所述，如 Red Hat Enterprise Linux 7 网络指南 中所述。

nmcli 和 nmtui 实用程序还允许您添加一个或多个新网络连接，以及修改和检查现有连接。如果要使用 nmcli 创建和管理网络连接，请参阅 第 1.2.2 节 “使用 nmcli 在安装过程后管理网络连接”。如果要使用 nmtui 创建和管理网络连接，请参阅 第 1.2.3 节 “使用 nmtui 在安装过程后管理网络连接”。

1.2.1. 在安装过程中配置网络访问

在安装过程中配置网络访问的方法：

- Anaconda 安装程序图形用户界面中的安装概述 屏幕中的 Network & Hostname 菜单
- Anaconda 安装程序文本模式中的 网络设置 选项
- Kickstart 文件

当系统在安装完成后第一次引导时，您在安装过程中配置的所有网络接口都会被自动激活。

有关在安装过程中网络访问配置的详细信息，请参阅 Red Hat Enterprise Linux 7 安装指南。

1.2.2. 使用 nmcli 在安装过程后管理网络连接

以 root 用户身份运行以下命令，以使用 nmcli 实用程序管理网络连接。

创建新连接：

```
~]# nmcli con add type type of the connection "con-name" connection name ifname ifname interface-name the name of the interface ipv4 address ipv4 address gw4 address gateway address
```

修改现有连接：

```
~]# nmcli con mod "con-name"
```

显示所有连接：

```
~]# nmcli con show
```

显示活跃连接：

```
~]# nmcli con show --active
```

显示特定连接的所有配置设置：

```
~]# nmcli con show "con-name"
```

有关 nmcli 命令行实用程序的更多信息，请参阅 Red Hat Enterprise Linux 7 网络指南。

1.2.3. 使用 nmtui 在安装过程后管理网络连接

NetworkManager 文本用户界面(TUI)实用程序 nmtui 提供了通过控制 NetworkManager 配置网络的文本界面。

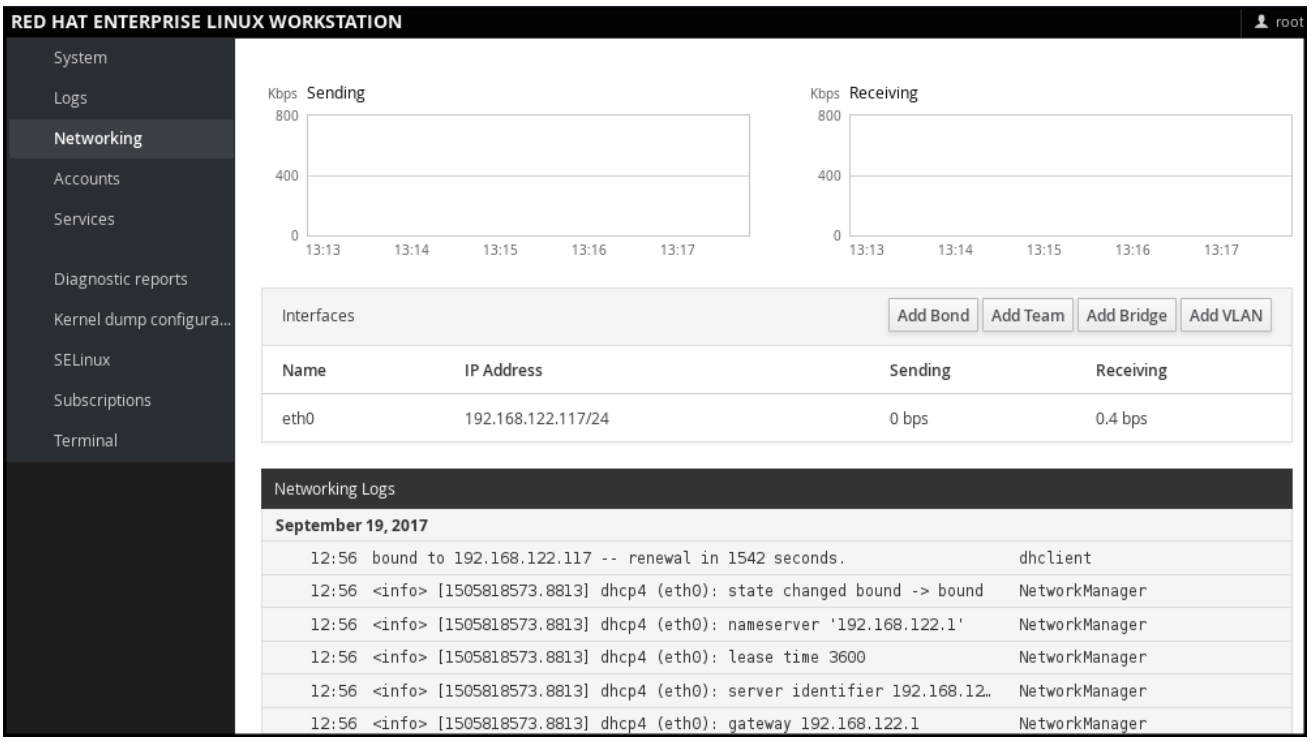
有关安装和使用 nmtui 文本界面工具的更多信息，请参阅 Red Hat Enterprise Linux 7 网络指南。

1.2.4. 在 Web 控制台中管理网络

在 Web 控制台中，您可以使用 Networking 菜单：

- 显示当前接收并发送的数据包
- 显示可用网络接口最重要的信息
- 显示网络日志的内容。
- 添加各种网络接口类型（bond、team、bridge、VLAN）

图 1.1. 在 Web 控制台中管理网络



1.3. 注册系统管理订阅的基础知识

1.3.1. 红帽订阅是什么以及哪些任务可供使用

订阅涵盖在红帽企业 Linux 7 上安装的产品（包括操作系统本身）。

Red Hat Content Delivery Network 订阅用来跟踪：

- 注册的系统
- 在这些系统中安装的产品
- 附加到这些产品的订阅

1.3.2. 安装过程中注册系统

这部分提供了在安装过程中注册 Red Hat Enterprise Linux 7 的简单概述。如果在安装后您的操作系统没有注册，您可以通过阅读本节来找出安装过程中可能会错过的内容。有关详细信息，请参阅《红帽企业 Linux 7 安装指南》。

基本上，在安装过程中注册该系统的方法有两种：

- 通常，注册是 Initial Setup 配置过程的一部分。如需更多信息，请参阅 Red Hat Enterprise Linux 7 安装指南。
- 另一个选项是将 Subscription Manager 作为安装后脚本运行，该脚本会在安装完成时并在第一次重启系统前执行自动注册。为确保这一点，修改 Kickstart 文件的 %post 部分。有关将订阅管理器作为安装后脚本运行的详情，请参考 Red Hat Enterprise Linux 7 安装指南。

1.3.3. 安装后注册系统

如果您在安装过程中没有注册您的系统，则可以应用以下步骤进行操作。请注意，这个过程中的所有命令都需要以 root 用户身份执行。

注册和订阅您的系统

1. 注册您的系统：

```
~]# subscription-manager register
```

该命令将提示您输入您的红帽客户门户网站用户名和密码。

2. 确定您需要的订阅池 ID：

```
~]# subscription-manager list --available
```

此命令显示您的红帽帐户的所有可用订阅。对于每个订阅，会显示各种相关信息，包括池 ID。

3. 通过使用上一步中决定的池 ID 替换 *pool_id* 来为您的系统附加适当的订阅：

```
~]# subscription-manager attach --pool=pool_id
```

有关注册您的系统以及附加 Red Hat Content Delivery Network 订阅的详情请参考 第 7 章 *注册系统管理并管理订阅*。

1.3.4. 将系统注册到 EUS 内容

要访问延长的更新支持(EUS)内容，请按以下方式注册您的系统：

1. 验证 EUS 权利是否可用：

```
~]# subscription-manager list --available --matches="*Extended Update Support"
```

```
+-------------------------------------------+
   Available Subscriptions
+-------------------------------------------+
Subscription Name:   Extended Update Support
Provides:            Red Hat Enterprise Linux High Availability for x86_64 - Extended
Update Support
                     Red Hat Enterprise Linux Resilient Storage for x86_64 - Extended
Update Support
                     Red Hat Enterprise Linux for x86_64 - Extended Update Support
                     Red Hat EUCJP Support (for RHEL Server) - Extended Update Support
                     RHEL for SAP - Extended Update Support
                     Red Hat Enterprise Linux Load Balancer (for RHEL Server) - Extended
Update Support
                     Red Hat Enterprise Linux Scalable File System (for RHEL Server) -
Extended Update Support
                     Red Hat CodeReady Linux Builder for x86_64 - Extended Update
Support
                     RHEL for SAP HANA - Extended Update Support
                     Red Hat Enterprise Linux High Performance Networking (for RHEL
Server) - Extended Update Support
                     Oracle Java (for RHEL Server) - Extended Update Support
```

```
                Red Hat S-JIS Support (for RHEL Server) - Extended Update Support
SKU:            RH00030
Contract:       12069074
Pool ID:        8a99f9ac7238188b01723d9c8a8a06a9
Provides Management: No
Available:      8
Suggested:      0
Service Level:  Layered
Service Type:   L1-L3
Subscription Type:  Instance Based
Starts:         05/22/2020
Ends:           05/21/2021
System Type:    Physical
```

2. 使用池标识符附加适用的订阅：

```
~]# subscription-manager attach --pool 8a99f9ac7238188b01723d9c8a8a06a9
```

3. 将为系统启用的默认软件仓库替换为 EUS 变体：

```
~]# subscription-manager repos --disable \*
```

4. 启用代表使用 RHEL 修订版本设置的 EUS 内容的软件仓库：

```
~]# subscription-manager repos --enable rhel-7-server-eus-rpms
```

5. 为最终系统选择所需的和支持发行版本：

```
~]# subscription-manager release --set 7.6
```

对于当前支持的 EUS 版本，请参阅延长更新支持附加组件。

1.3.5. 将系统注册到 E4S 内容

以下流程描述了如何注册系统和利用 E4S 内容。

1. 使用以下命令注册您的系统：

```
~]# subscription-manager register
```

2. 验证 **E4S** 权利是否可用：

```
~]# subscription-manager list --available --matches="*Update Services for SAP Solutions*"
```

```
+-------------------------------------------+
   Available Subscriptions
+-------------------------------------------+
Subscription Name:   Red Hat Enterprise Linux for SAP Solutions, Standard (Physical or Virtual Nodes)
Provides:            dotNET on RHEL Beta (for RHEL Server)
                     Red Hat CodeReady Linux Builder for x86_64
                     Red Hat Enterprise Linux for SAP HANA for x86_64
                     Red Hat Ansible Engine
                     RHEL for SAP HANA - Update Services for SAP Solutions
                     Red Hat Enterprise Linux Scalable File System (for RHEL Server) - Extended Update Support
                     RHEL for SAP HANA - Extended Update Support
                     Red Hat Enterprise Linux Atomic Host Beta
                     Red Hat Beta
                     Red Hat EUCJP Support (for RHEL Server) - Extended Update Support
                     Red Hat Enterprise Linux High Availability for x86_64
                     Red Hat Enterprise Linux Load Balancer (for RHEL Server) - Extended Update Support
                     dotNET on RHEL (for RHEL Server)
                     Red Hat CodeReady Linux Builder for x86_64 - Extended Update Support
                     Red Hat Enterprise Linux High Availability - Update Services for SAP Solutions
                     Red Hat Enterprise Linux Resilient Storage for x86_64 - Extended Update Support
                     Red Hat Enterprise Linux High Availability for x86_64 - Extended Update Support
                     Oracle Java (for RHEL Server)
                     Red Hat Enterprise Linux Server - Update Services for SAP Solutions
                     Red Hat Software Collections (for RHEL Server)
                     Red Hat Enterprise Linux Scalable File System (for RHEL Server)
                     Red Hat Enterprise Linux High Performance Networking (for RHEL Server) - Extended Update Support
                     RHEL for SAP - Update Services for SAP Solutions
                     Oracle Java (for RHEL Server) - Extended Update Support
                     Red Hat Enterprise Linux Atomic Host
                     Red Hat Developer Tools (for RHEL Server)
                     Red Hat Software Collections Beta (for RHEL Server)
                     Red Hat Enterprise Linux Server
                     Red Hat Enterprise Linux for SAP Applications for x86_64
                     Red Hat Developer Tools Beta (for RHEL Server)
                     Red Hat Enterprise Linux for x86_64
                     Red Hat Enterprise Linux for x86_64 - Extended Update Support
                     RHEL for SAP - Extended Update Support
                     Red Hat Developer Toolset (for RHEL Server)
                     Red Hat S-JIS Support (for RHEL Server) - Extended Update Support
SKU:                 RH00764
Contract:            11977725
```

```
Pool ID:            8a85f99c6c4825eb016c4a30d3493064
Provides Management: Yes
Available:          18
Suggested:          0
Service Level:      Standard
Service Type:       L1-L3
Subscription Type:  Instance Based
Starts:             03/29/2020
Ends:               12/31/2021
System Type:        Physical
```

3. 使用池标识符附加适用的订阅：

```
~]# subscription-manager attach --pool=##################
```

4. 将为系统启用的默认软件仓库替换为 EUS 变体：

```
~]# subscription-manager repos --disable="*"
```

5. 启用代表正在使用的 RHEL 修订的 E4S 内容的软件仓库：

```
~]# subscription-manager --enable=rhel-7-server-e4s-rpms
```

6. 清除存储库缓存并将系统锁定到支持您的 SAP 应用程序的 E4S 有效发行版本：

```
~]# yum clean all && subscription-manager release --set=7.7
```

1.4. 安装软件

这部分提供了在 Red Hat Enterprise Linux 7 系统中完成软件安装基础知识的信息。它提到了在 第 1.4.1 节 “软件安装的先决条件” 中安装软件所需的先决条件，提供 第 1.4.2 节 “软件打包和软件存储库系统简介” 中软件打包和软件存储库的基本信息，并参考 第 1.4.3 节 “使用 Subscription Manager 和 Yum 管理基本软件安装任务” 中与软件安装相关的基本任务。

1.4.1. 软件安装的先决条件

Red Hat Content Delivery Network 订阅服务提供处理红帽软件库存的机制，并可让您安装其他软件或更新已安装的软件包。您可以在注册系统并附加订阅后开始安装软件，如 第 1.3 节 “注册系统管理订阅的基础知识” 所述。

1.4.2. 软件打包和软件存储库系统简介

Red Hat Enterprise Linux 系统上的所有软件都被分成 RPM 包，这些软件包存储在特定的存储库中。当系统订阅 Red Hat Content Delivery Network 时，会在 /etc/yum.repos.d/ 目录中创建一个仓库文件。

使用 yum 实用程序管理软件包操作：

- 搜索软件包信息
- 安装软件包
- 更新软件包
- 删除软件包
- 检查当前可用的软件仓库列表
- 添加或删除软件仓库
- 启用或禁用软件仓库

有关与安装软件相关的基本任务的详情请参考 第 1.4.3 节 “使用 Subscription Manager 和 Yum 管理基本软件安装任务”。有关管理软件存储库的详情请参考 第 7.2 节 “管理软件存储库”。有关使用 yum 工具的详情请参考 第 9 章 *yum*。

1.4.3. 使用 Subscription Manager 和 Yum 管理基本软件安装任务

安装操作系统后可能需要的最基本软件安装任务包括：

- 列出所有可用存储库：

```
~]# subscription-manager repos --list
```

- 列出所有当前启用的软件仓库：

```
~]$ yum repolist
```

- 启用或禁用存储库：

```
~]# subscription-manager repos --enable repository
```

```
~]# subscription-manager repos --disable repository
```

- 搜索与特定字符串匹配的软件包：

```
~]$ yum search string
```

- 安装软件包：

```
~]# yum install package_name
```

- 更新所有软件包及其依赖项：

```
~]# yum update
```

- 更新软件包：

```
~]# yum update package_name
```

- 卸载软件包以及依赖于它的任何软件包：

```
~]# yum remove package_name
```

- 列出所有已安装和可用软件包的信息：

```
~]$ yum list all
```

- 列出所有安装的软件包信息：

```
~]$ yum list installed
```

1.5. 在引导时启动 SYSTEMD 服务

Systemd 是 Linux 操作系统的系统和服务管理器，它引进了 systemd 单元的概念。有关 systemd 的详情请参考 第 10.1 节 “systemd 简介”。

本节介绍如何在引导时启用或禁用服务。它还介绍了如何通过 Web 控制台 管理服务。

1.5.1. 启用或禁用服务

您可以在安装过程中在引导时启用或禁用的服务，或者您可以在安装的操作系统中启用或禁用服务。

要在安装过程中在引导时启用或禁用的服务列表，请使用 Kickstart 文件中的 services 选项：

```
services [--disabled=list] [--enabled=list]
```

注意

禁用的服务列表会在启用的服务列表前进行处理。因此，如果服务出现在这两个列表中，它将被启用。服务列表应以逗号分隔的格式指定。不要在服务列表中包含空格。有关详细信息，请参阅《Red Hat Enterprise Linux 7 安装指南》。

在已安装的操作系统中启用或禁用服务：

```
~]# systemctl enable service_name
```

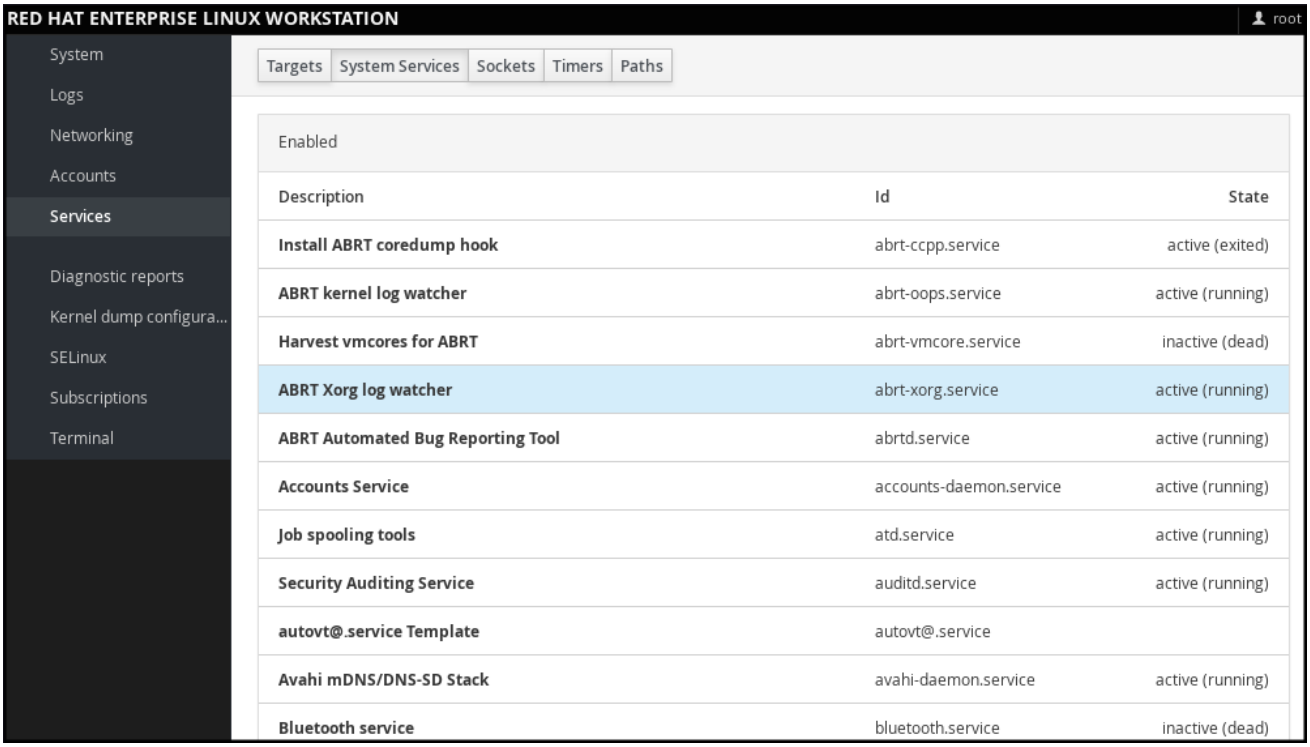
```
~]# systemctl disable service_name
```

详情请查看 第 10.2 节 “管理系统服务”。

1.5.2. 在 web 控制台中管理服务

在 Web 控制台中，选择 Services 来管理 systemd 目标、服务、套接字、计时器和路径。您可以检查其状态、启动或停止它们、启用或禁用它们、启用或禁用它们。

图 1.2. 在 web 控制台中管理服务



1.5.3. systemd 服务中的其他资源

有关 systemd 的详情请参考 第 10 章 *使用 systemd 管理服务*。

1.6. 使用防火墙、SELINUX 和 SSH 日志提高系统安全性

计算机安全性是保护计算机系统免受硬件、软件或信息损坏或损害其提供的服务的影响或错误。因此，确保计算机安全性是关键任务，不仅在企业中处理敏感数据或处理某些业务交易。

计算机安全性包括各种功能和工具。本节仅涵盖安装操作系统后您需要配置的基本安全功能。有关保护红帽企业 Linux 7 安全性的详细信息，请参阅红帽企业 Linux 7 安全指南。

1.6.1. 确保防火墙已启用并正在运行

1.6.1.1. 什么是防火墙问题及其如何增强系统安全性

防火墙是一种网络安全系统，它根据预先确定的安全规则监控和控制传入和传出的网络流量。防火墙通常在可信、安全的内部网络和其他外部网络之间建立一个障碍。

在 Red Hat Enterprise Linux 7 中，防火墙由 firewalld 服务提供，该服务会在安装 Red Hat Enterprise Linux 期间自动启用。但是，如果您明确禁用该服务，例如在 kickstart 配置中，您可以重新启用它，如 第 1.6.1.2 节 “重新启用 firewalld 服务” 所述。有关 Kickstart 文件中的防火墙设置选项概述，请参阅 Red Hat Enterprise Linux 7 安装指南。

1.6.1.2. 重新启用 firewalld 服务

如果 firewalld 服务在安装后被禁用，红帽建议红帽考虑重新启用该服务。

您可以以常规用户身份显示 firewalld 的当前状态：

```
~]$ systemctl status firewalld
```

如果没有启用并运行 firewalld，切换到 root 用户并更改其状态：

```
~]# systemctl start firewalld
```

```
~]# systemctl enable firewalld
```

有关与 firewalld 相关的安装后流程的更多信息，请参阅 Red Hat Enterprise Linux 7 安全指南。有关配置和使用防火墙的详情，请查看 Red Hat Enterprise Linux 7 安全指南

1.6.2. 确定适当的 SELinux 状态

1.6.2.1. SELinux 是什么及其如何增强系统安全性

Security Enhanced Linux(SELinux) 是一个额外的系统安全层，它决定哪个进程可以访问哪些文件、目录和端口。

SELinux 状态

SELinux 有两个可能的状态：

- Enabled
- Disabled

禁用 SELinux 时，仅使用自主访问控制(DAC)规则。

SELinux 模式

启用 SELinux 时，它可以以以下模式之一运行：

- Enforcing
- Permissive

强制模式意味着 SELinux 策略会被强制实施。SELinux 根据 SELinux 策略规则拒绝访问，并且只启用特别允许的交互。强制模式是安装后的默认模式，也是最安全的 SELinux 模式。

许可模式意味着 SELinux 策略不会被强制实施。SELinux 不会拒绝访问，但是对于在 enforcing 模式运行时会被拒绝的操作，则会记录拒绝信息。Permissive 模式是安装过程中的默认模式。在某些情况下，以 permissive 模式运行也很有用，例如，在进行故障排除时您需要访问 Access Vector Cache(AVC)拒绝。

有关 Red Hat Enterprise Linux 7 中 SELinux 的更多信息，请参阅 Red Hat Enterprise Linux 7 SELinux 用户和管理员指南。

1.6.2.2. 确保 SELinux 所需的状态

默认情况下，SELinux 在安装过程中以 permissive 模式运行，安装完成后处于强制模式。

然而，在某些特定情况下，SELinux 可能会明确设置为 permissive 模式，或者甚至可能在安装的操作系统中禁用。可以在 kickstart 配置中设置此设置，例如：有关 Kickstart 文件中的 SELinux 设置选项概述，请参阅 Red Hat Enterprise Linux 7 安装指南。

重要

红帽建议使您的系统保持在 enforcing 模式下。

显示当前的 SELinux 模式，并根据需要设置模式：

确保 SELinux 所需的状态

1. 显示当前生效的 SELinux 模式：

```
~]$ getenforce
```

2. 如果需要，请在 SELinux 模式之间切换。

切换可以是临时的，也可以是永久性的。临时切换不会在重新启动后保留，而永久切换为：

- 临时切换到 enforcing 或 permissive 模式：

```
~]# setenforce Enforcing
```

```
~]# setenforce Permissive
```

- 要永久设置 SELinux 模式，修改 /etc/selinux/config 配置文件中的 ***SELINUX*** 变量。

例如，将 SELinux 切换到 enforcing 模式：

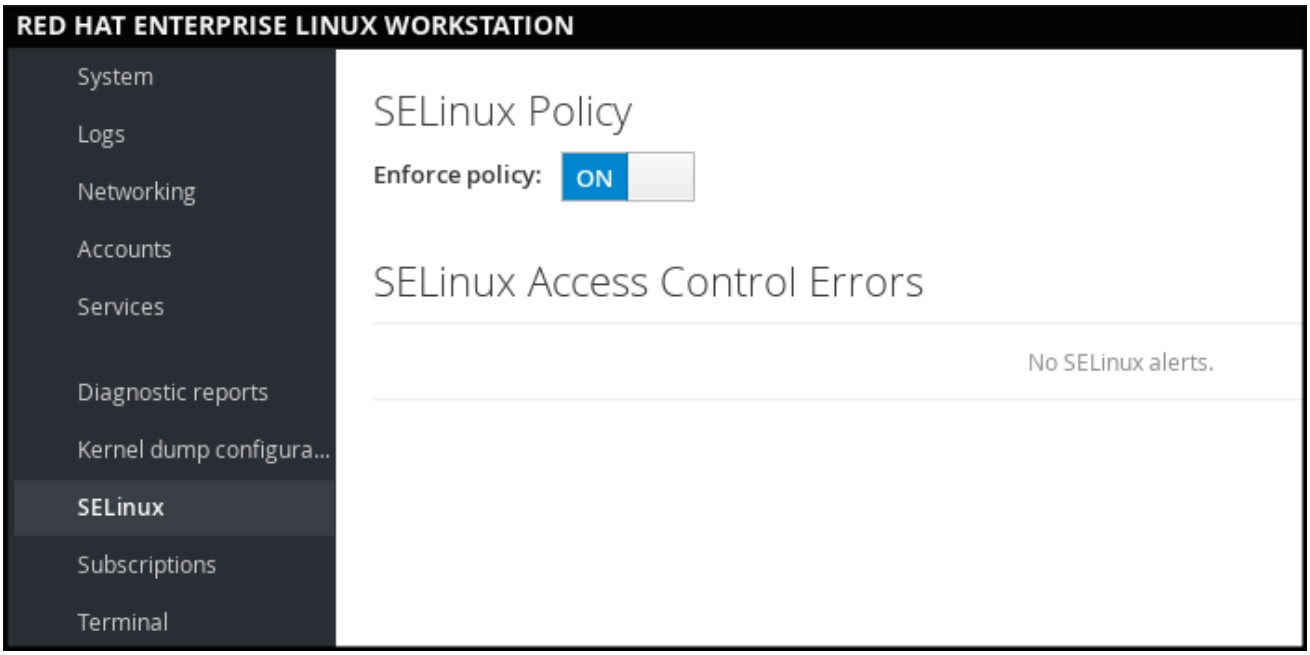
```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=enforcing
```

1.6.2.3. 在 web 控制台中管理 SELinux

在 Web 控制台中，使用 SELinux 选项打开或关闭 SELinux 强制策略。

默认情况下，web 控制台中的 SELinux enforcing 策略是 on，SELinux 在 enforcing 模式下运行。通过关闭 SELinux，您可以将 SELinux 切换为 permissive 模式。请注意，与 /etc/sysconfig/selinux 文件中的默认配置偏差会在下一次引导时自动恢复。

图 1.3. 在 web 控制台中管理 SELinux



1.6.3. 使用基于 SSH 的身份验证

1.6.3.1. 基于 SSH 的身份验证及其如何增强系统安全性

如果要保护与其他计算机的通信，您可以使用基于 SSH 的身份验证。

安全外壳(SSH)是一种协议，可促进客户端-服务器通信，并允许用户远程登录任何运行 SSH 的主机系统。SSH 加密连接。客户端使用加密将其身份验证信息传输到服务器，会话期间发送和接收的所有数据也在加密下传输。

SSH 使其用户无需输入密码即可进行身份验证。为此，SSH 使用私钥-公钥方案。

有关 SSH 保护的详情请参考 第 12.1.2 节 “主要功能”。

1.6.3.2. 建立 SSH 连接

为了能够使用 SSH 连接，创建由公钥和私钥组成的两对密钥。

创建密钥文件并将 Them 复制到服务器

1. 生成公钥和私钥：

```
~]$ ssh-keygen
```

这两个密钥都存储在 ~/.ssh/ 目录中：

- ~/.ssh/id_rsa.pub - public key
- ~/.ssh/id_rsa - private key

公钥不需要是保密的。它用于验证私钥。私钥是机密。您可以选择使用密钥生成过程中指定的密语来保护私钥。使用密码短语时，身份验证更安全，但不再是免密码操作。您可以使用 ssh-agent 命令避免这种情况。在这种情况下，您将仅在会话开始时输入一次密码短语。有关 ssh-agent 配置的详情请参考 第 12.2.4 节 “使用基于密钥的身份验证”。

2. 将最新修改的公钥复制到您要登录到的远程机器中：

```
~]# ssh-copy-id USER@hostname
```

现在，您可以安全地输入系统，但不会输入密码。

1.6.3.3. 禁用 SSH Root 登录

要提高系统安全性，您可以禁用 root 用户的 SSH 访问，这是默认启用的。

有关此主题的更多信息，请参阅 Red Hat Enterprise Linux 7 安全指南。

禁用 SSH Root 登录

1. 访问 /etc/ssh/sshd_config 文件：

```
~]# vi /etc/ssh/sshd_config
```

2. 将 #PermitRootLogin yes 的行改为：

```
PermitRootLogin no
```

3. 重启 sshd 服务：

```
~]# systemctl restart sshd
```

1.7. 管理用户帐户的基础知识

Red Hat Enterprise Linux 7 是一个多用户操作系统，可让不同计算机上的多个用户访问安装在同一台计算机上的单一系统。每个用户都在自己的帐户下运行，因此管理用户帐户代表 Red Hat Enterprise Linux 系统管理的一个核心元素。

普通帐户和系统帐户

为特定系统用户创建普通帐户。这些帐户可以在正常的系统管理过程中添加、删除和修改。

系统帐户代表系统上的特定应用程序标识符。此类帐户通常仅在软件安装时添加或操作，且不会在以后进行修改。

警告

系统帐户假定在一个系统中本地可用。如果远程配置和提供这些帐户，如 LDAP 配置实例中，则可能会出现系统中断和服务启动故障。

对于系统帐户，1000 以下的用户 ID 被保留。对于普通帐户，使用从 1000 开始的 ID。但推荐做法是使用从 5000 开始的 ID。如需更多信息，请参阅 第 4.1 节 “用户和组介绍”。分配 ID 的指南可以在 /etc/login.defs 文件中找到。

```
# Min/max values for automatic uid selection in useradd
#
UID_MIN         1000
UID_MAX        60000
# System accounts
SYS_UID_MIN       201
SYS_UID_MAX       999
```

对于哪些组和哪些组，它们可以用作哪些用途

组是出于共同目的将多个用户帐户连接在一起的实体，例如授予对特定文件的访问权限。

1.7.1. 管理用户帐户和组的最基本命令行工具

管理用户帐户和组群的最基本任务以及适当的命令行工具，包括：

- 显示用户和组群 ID:

```
~]$ id
```

- 创建新用户帐户：

```
~]# useradd [options] user_name
```

- 为属于用户名的用户帐户分配新密码：

```
~]# passwd user_name
```

- 将用户添加到组中：

```
~]# usermod -a -G group_name user_name
```

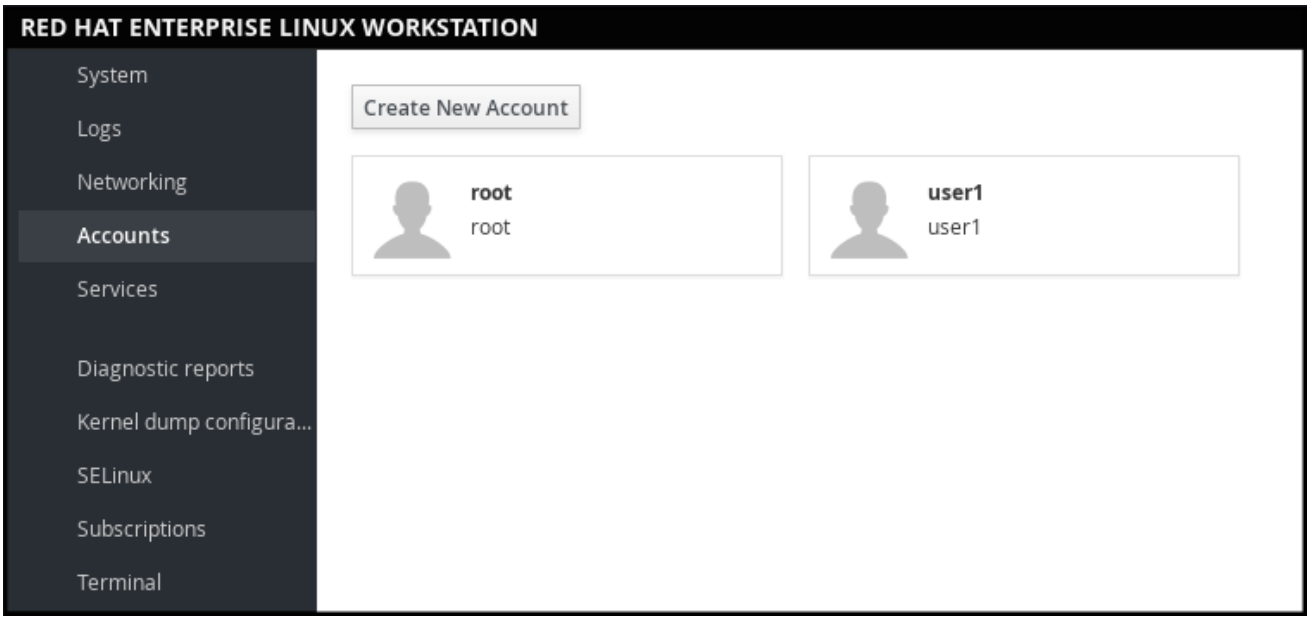
有关管理用户和组的详情请参考 第 4 章 管理用户和组。

如果要使用图形用户界面管理用户和组，请参阅 第 4.2 节 “在图形环境中管理用户”。

1.7.2. 在 web 控制台中管理用户帐户

要在 web 控制台中 管理帐户，请选择 Accounts 菜单。

图 1.4. 在 web 控制台中管理用户帐户



1.8. 使用 KDUMP 机制转储已清除内核

本节介绍了内核崩溃转储机制，也称为 kdump，并在 第 1.8.1 节 “kdump 是什么，它可以用于什么任务” 中简单解释了 kdump 的用途。

激活 kdump 服务是安装过程的一部分，默认情况下，k dump 在安装过程中启用。本节总结了如何在 第 1.8.2 节 “在安装过程中启用和激活 kdump” 安装过程中激活 kdump，并在安装 第 1.8.3 节 “确保安装过程后已安装并启用 kdump” 后禁用 kdump 服务时如何手动启用 kdump 服务。

您还可以使用 Web 控制台配置 kdump。如需更多信息，请参阅 第 1.8.4 节 “在 web 控制台中配置 kdump”。

1.8.1. kdump 是什么，它可以用于什么任务

如果系统崩溃，您可以使用名为 kdump 的内核崩溃转储机制，以便保存系统内存内容，以便稍后进行分析。kdump 机制依赖于 kexec 系统调用，该调用可用于从另一个内核上下文引导 Linux 内核，绕过 BIOS，并保留第一个内核内存内容，否则会丢失第一个内核的内存内容。

当发生内核崩溃时，kdump 使用 kexec 引导进入第二个内核（捕获内核），该内核位于第一个内核无法访问的系统内存的保留部分。第二个内核捕获崩溃内核的内存（崩溃转储）的内容并将其保存。

1.8.2. 在安装过程中启用和激活 kdump

在安装过程中，可以在 Anaconda 安装程序中或使用 Kickstart 文件中的 %addon

com_redhat_kdump 命令来启用和激活 kdump。

如需更多信息，请参阅根据安装方法划分的源：

- 当使用 Anaconda 安装程序安装时，请参考：

  在《红帽企业 Linux 7 安装指南》中安装 Anaconda.

- 当使用 Kickstart 文件安装时，请参考：

  红帽企业 Linux 7 安装指南中的 Kickstart 命令和选项.

1.8.3. 确保安装过程后已安装并启用 kdump

确保安装了 kdump 并进行配置：

检查 kdump 是否已安装并配置 kdump

1. 检查是否在您的系统中安装了 kdump:

   ```
   ~]$ rpm -q kexec-tools
   ```

2. 如果没有安装 kdump，请以 root 用户身份 输入：

   ```
   ~]# yum install kexec-tools
   ```

3. 配置 kdump ：

   使用命令行或图形用户界面。

   这两个选项均在 Red Hat Enterprise Linux 7 内核崩溃指南中进行了详细介绍。

   如果您需要安装图形配置工具：

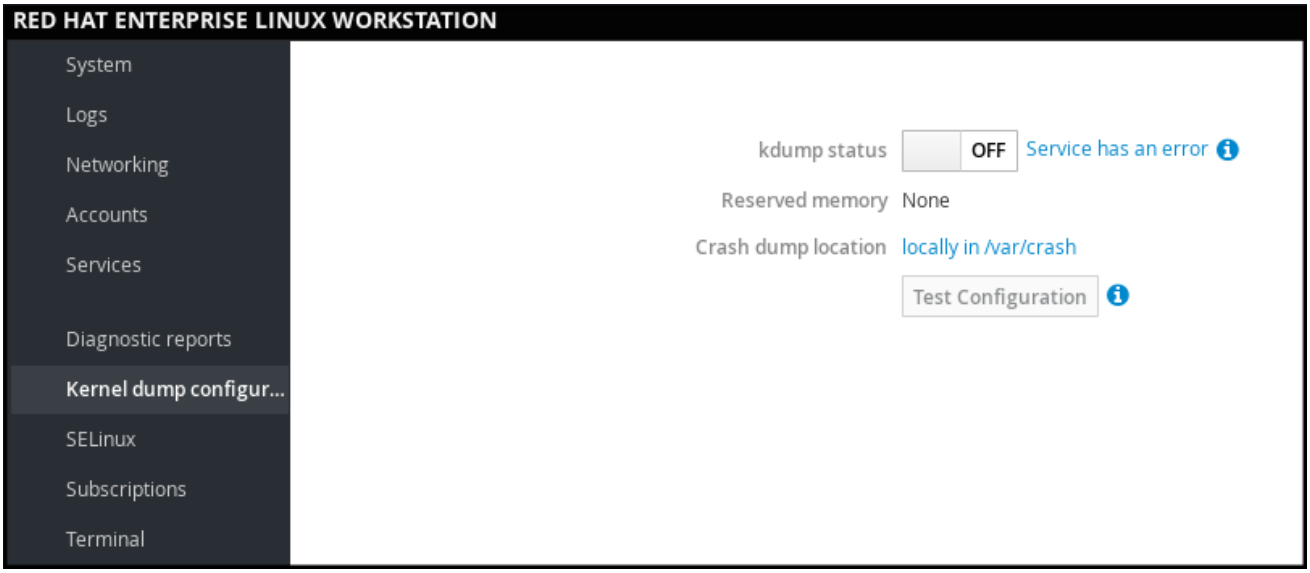
```
~]# yum install system-config-kdump
```

1.8.4. 在 web 控制台中配置 kdump

在 Web 控制台中，选择 Kernel dump 进行验证：

- kdump 状态
- 为 kdump保留的内存量
- 崩溃转储文件的位置

图 1.5. 在 web 控制台中配置 kdump



1.8.5. kdump 上的其他资源

有关 kdump 的更多信息，请参阅 Red Hat Enterprise Linux 7 内核崩溃指南。

1.9. 执行系统救援并使用 REAR 创建系统备份

当软件或硬件故障破坏操作系统时，您需要一种机制来救援系统。保存系统备份也很有用。红帽建议使用 Relax-and-Recover(ReaR)工具来满足这两个需求。

1.9.1. ReaR 是否和哪些任务可供使用

Rear 是一个灾难恢复和系统迁移实用程序，可让您创建完整的救援系统。默认情况下，这个救援系统只恢复存储布局和启动加载器，而不是实际的用户和系统文件。

此外，某些备份软件允许您集成 ReaR 以用于灾难恢复。

rear 启用执行以下任务：

- 在新硬件中引导救援系统
- 复制原始存储布局
- 恢复用户和系统文件

1.9.2. 安装和配置 ReaR 的快速入门

要安装 ReaR，以 root 用户身份输入：

```
~]# yum install rear
```

使用 /etc/rear/local.conf 文件中的设置来配置 ReaR。

如需更多信息，请参阅 第 27.1 节 “基本 ReaR 用法”。

1.9.3. 使用 ReaR 创建救援系统的快速入门

要创建救援系统，请以 root 用户身份 执行以下命令

```
~]# rear mkrescue
```

有关使用 ReaR 创建救援系统的详情请参考 第 27.1.3 节 “创建救援系统”。

1.9.4. 使用备份软件配置 ReaR 的快速入门

Rear 包含完全集成的内置或内部备份方法，称为 NETFS。

要使 ReaR 使用其内部备份方法，请将这些行添加到 /etc/rear/local.conf 文件中：

```
BACKUP=NETFS
BACKUP_URL=backup location
```

您还可以将 ReaR 配置为在创建新归档时保留之前的备份归档，方法是在 /etc/rear/local.conf 中添加以下行：

```
NETFS_KEEP_OLD_BACKUP_COPY=y
```

要让备份递增，意味着每次运行时只备份更改的文件，将这一行添加到 /etc/rear/local.conf 中：

```
BACKUP_TYPE=incremental
```

有关使用 ReaR NETFS 内部备份方法的详情请参考 第 27.2.1 节 “内置备份方法”。

有关支持的外部备份方法和不支持的备份方法的详情，请参考 第 27.2.2 节 “支持的备份方法” 和 第 27.2.3 节 “不支持的备份方法”。

1.10. 使用日志文件来故障排除问题

在对问题进行故障排除时，您可能会欣赏包含不同操作系统信息和消息的日志文件。Red Hat Enterprise Linux 7 中的日志记录系统基于内置的 syslog 协议。特定的程序使用这个系统记录事件并将其整理到日志文件中，这些文件在审核操作系统和故障排除各种问题时非常有用。

有关日志文件的详情请参考 第 23 章 *查看和管理日志文件*。

1.10.1. 服务处理 syslog 消息

系统日志消息由两个服务处理：

- systemd-journald 守护进程 - 收集来自内核的消息、启动过程的早期阶段、标准输出以及守护进程在启动和运行过程中的错误，以及 syslog，并将消息转发到 rsyslog 服务以便进一步处

理。

- rsyslog 服务 - 按类型和优先级清理 syslog 消息，并将其写入 /var/log 目录中的文件，以持久存储日志。

1.10.2. 保护 syslog 消息的子目录

系统日志消息根据包含的信息和日志类型保存在 /var/log 目录下的不同子目录中：

- var/log/messages - 除下面所述之外的所有 syslog 信息
- var/log/secure - 与安全性和身份验证相关的消息和错误
- var/log/maillog - 与邮件服务器相关的消息和错误
- var/log/cron - 与定期执行任务相关的日志文件
- var/log/boot.log - 与系统启动相关的日志文件

1.11. 访问红帽支持

要获得红帽支持，请使用红帽客户门户网站，它提供对您的订阅中所有可用的访问权限。

本节描述：

- 获取红帽支持，请参阅 第 1.11.1 节 “通过红帽客户门户网站获取红帽支持”
- 使用 SOS 报告 对问题进行故障排除，请参阅 第 1.11.2 节 “使用 SOS 报告故障排除问题”

1.11.1. 通过红帽客户门户网站获取红帽支持

通过使用红帽客户门户网站，您可以：

- 创建新的支持问题单
- 与红帽专家开启实时聊天
- 通过致电或发送电子邮件联系红帽专家

要访问红帽客户门户，请访问

要使用与红帽支持相关的红帽客户门户网站服务，您可以使用：

- Web 浏览器
- 红帽支持工具

1.11.1.1. Red Hat 支持工具及其可以用于哪些任务

红帽支持工具 是基于命令行的工具，可为基于订阅的红帽访问服务提供一个文本控制台界面。此工具包含在 redhat-support-tool 软件包中。

红帽支持工具 可让您执行与支持相关的任务，例如：

- 打开或更新支持问题单
- 搜索红帽知识库解决方案
- 分析 Python 和 Java 错误

以互动模式启动该工具：

```
~]$ redhat-support-tool
Welcome to the Red Hat Support Tool.
Command (? for help):
```

在互动模式中，输入？显示可用命令 ：

```
Command (? for help): ?
```

有关安装和使用红帽支持工具的更多信息，请参阅 第 8 章 *使用红帽支持工具访问支持* 和红帽知识库文章 Red Hat Access: Red Hat Support Tool。

1.11.2. 使用 SOS 报告故障排除问题

SOS 报告 从 Red Hat Enterprise Linux 系统收集配置详情、系统信息和诊断信息。当您创建一个支持问题单时附加报告。

请注意，SOS 报告 在 sos 软件包中提供，该软件包未安装 Red Hat Enterprise Linux 7 的默认最小安装。

安装 sos 软件包：

```
~]# yum install sos
```

生成 SOS 报告 ：

```
~]# sosreport
```

要将 sos 报告 附加到您的支持问题单中，请参阅红帽知识库文章 如何将文件附加到红帽支持问题单？请注意，在附加 sos 报告时，系统会提示您输入支持案例的数量。

有关 SOS 报告 的更多信息，请参阅红帽知识库文章 什么是 sosreport 以及如何在 Red Hat Enterprise Linux 4.6 及之后的版本中创建？

第 2 章 系统位置和键盘配置

系统区域设置指定系统服务和用户界面的语言设置。键盘布局设置控制文本控制台和图形用户界面中使用的布局。

可以通过修改 /etc/locale.conf 配置文件或使用 localectl 实用程序来设置这些设置。此外，您可以使用图形用户界面执行任务；有关此方法的说明，请参阅 Red Hat Enterprise Linux 7 安装指南。

2.1. 设置系统区域

系统范围的区域设置保存在 /etc/locale.conf 文件中，该文件在早期引导时由 systemd 守护进程读取。每个服务或用户都会继承 /etc/locale.conf 中配置的区域设置，单独程序或个人用户均覆盖它们。

/etc/locale.conf 的基本文件格式是一个以换行分隔的变量分配列表。例如：在 /etc/locale.conf 中带有英语信息的德语区域设置如下：

```
LANG=de_DE.UTF-8
LC_MESSAGES=C
```

此处，LC_MESSAGES 选项决定用于写入到标准错误输出的诊断消息的区域设置。要进一步指定 /etc/locale.conf 中的 区域设置，您可以使用几个其他选项，具体会在 表 2.1 “在 /etc/locale.conf 中可配置的选项” 中概述。有关这些选项的详情，请查看 locale(7) 手册页。请注意，不应在 /etc/locale.conf 中配置 LC_ALL 选项，它代表所有可能的选项。

表 2.1. 在 /etc/locale.conf 中可配置的选项

选项	描述
LANG	为系统区域设置提供默认值。
LC_COLLATE	更改比较本地字母中字符串的函数行为。
LC_CTYPE	更改字符处理和分类功能以及多字节字符函数的行为。
LC_NUMERIC	描述数字通常的打印方式，详情包括十进制点和十进制逗号。
LC_TIME	更改当前时间、24 小时与 12 小时的显示。
LC_MESSAGES	确定用于写入到标准错误输出的诊断消息的区域设置。

2.1.1. 显示当前状态

localectl 命令可用于查询和更改系统区域设置和键盘布局设置。要显示当前的设置，请使用 status 选项：

```
localectl status
```

例 2.1. 显示当前状态

上一命令的输出列出了当前设置的区域设置、为控制台和 X11 窗口系统配置的键盘布局。

```
~]$ localectl status
  System Locale: LANG=en_US.UTF-8
    VC Keymap: us
   X11 Layout: n/a
```

2.1.2. 列出可用的区域

要列出您的系统所有可用区域，请输入：

```
localectl list-locales
```

例 2.2. 列出区域

假设您想要选择特定的英语区域设置，但您不确定它是否在系统中可用。您可以使用以下命令列出所有英语区域来检查：

```
~]$ localectl list-locales | grep en_
en_AG
en_AG.utf8
en_AU
en_AU.iso88591
en_AU.utf8
en_BW
en_BW.iso88591
en_BW.utf8

output truncated
```

2.1.3. 设置区域

要设置默认系统区域设置，以 root 用户身份运行以下命令：

```
localectl set-locale LANG=locale
```

使用区域名称替换 *locale*，使用 localectl list-locales 命令找到。以上语法还可用于从 表 2.1 “在 /etc/locale.conf 中可配置的选项” 配置参数。

例 2.3. 更改默认区域

例如，如果要将 English 设为默认区域设置，请先使用 list-locales 来查找此区域设置的名称。然后，作为 root 用户 以以下格式输入命令：

```
~]# localectl set-locale LANG=en_GB.utf8
```

2.1.4. 在使用 Kickstart 安装时永久进行系统区域设置

使用 Red Hat Kickstart 安装方法安装 Red Hat Enterprise Linux 时，升级操作系统后系统区域设置可能不会保留。

当 Kickstart 文件的 %packages 部分包含 --instLang 选项时，_install_langs RPM 宏被设置为这个安装的特定值，并相应地调整已安装的区域集合。但是，这个调整只会影响此安装，而不是后续升级。如果升级重新安装 glibc 软件包，则会升级整个区域集合，而不是只升级您在安装过程中请求的区域设置。

为避免这种情况，请永久选择区域设置。您有以下选项：

- 如果您还没有启动 Kickstart 安装，请修改 Kickstart 文件使其包含应用这个步骤以全局设置 RPM 宏的说明： 在 Kickstart 安装过程中设置 RPM 宏
- 如果您已经安装了该系统，请应用这个步骤在系统中全局设置 RPM 宏： 全局设置 RPM 宏

在 Kickstart 安装过程中设置 RPM 宏

1. 修改 Kickstart 文件的 %post 部分：

```
LANG=en_US
```

```
echo "%_install_langs $LANG" > /etc/rpm/macros.language-conf

yum-config-manager --setopt=override_install_langs=$LANG --save
```

2. 修改 Kickstart 文件的 %packages 部分：

```
%packages
yum-utils*
%end
```

全局设置 RPM 宏

1. 在 /etc/rpm/macros.language-conf 中创建包含以下内容的 RPM 配置文件：

```
%_install_langs LANG
```

LANG 是 instLang 选项的值。

2. 使用以下内容更新 /etc/yum.conf 文件：

```
override_install_langs=LANG
```

2.2. 更改键盘布局

键盘布局设置让用户能够控制文本控制台和图形用户界面中使用的布局。

2.2.1. 显示当前设置

如前文所述，您可以使用以下命令检查当前的键盘布局配置：

```
localectl status
```

例 2.4. 显示键盘设置

在以下输出中，您可以看到为虚拟控制台和 X11 窗口系统配置的键盘布局。

```
~]$ localectl status
  System Locale: LANG=en_US.utf8
```

```
    VC Keymap: us
   X11 Layout: us
```

2.2.2. 列出可用的键映射

要列出系统中可以配置的所有可用键盘布局，请输入：

```
localectl list-keymaps
```

例 2.5. 搜索部分关键字图

您可以使用 grep 搜索上一命令的输出，以查找特定键映射名称。通常，有多个与当前设置的区域设置兼容的密钥映射。例如，要查找可用的层次键盘布局，请输入：

```
~]$ localectl list-keymaps | grep cz
cz
cz-cp1250
cz-lat2
cz-lat2-prog
cz-qwerty
cz-us-qwertz
sunt5-cz-us
sunt5-us-cz
```

2.2.3. 设置 Keymap

要为您的系统设置默认键盘布局，以 root 用户身份运行以下命令：

```
localectl set-keymap map
```

使用 localectl list-keymaps 命令输出中的 keymap 的名称替换 *map*。除非传递 --no-convert 选项，否则所选设置也会应用于 X11 窗口系统的默认键盘映射（在将其转换为最匹配的 X11 键盘映射）。这也适用于反向，您可以以 root 用户身份使用以下命令来指定键映射：

```
localectl set-x11-keymap map
```

如果您希望 X11 布局与控制台布局不同，请使用 --no-convert 选项。

```
localectl --no-convert set-x11-keymap map
```

使用这个选项时，可以在不更改之前的控制台布局设置的情况下指定 X11 密钥映射。

例 2.6. 设置 X11 Keymap 并行

假设您想要在图形界面中使用德语键盘布局，但对于要保留美国键盘映射的控制台操作。要做到这一点，以 root 用户身份输入：

```
~]# localectl --no-convert set-x11-keymap de
```

然后，您可以通过检查当前状态来验证您的设置是否成功：

```
~]$ localectl status
  System Locale: LANG=de_DE.UTF-8
    VC Keymap: us
   X11 Layout: de
```

除了键盘布局（映射）外，还可以指定其他三个选项：

```
localectl set-x11-keymap map model variant options
```

使用键盘型号名称、变体和 选项替换 model，用键盘变体和选项组件替换模型，可用于增强键盘行为。默认情况下不设置这些选项。有关 X11 Model、X11 Variant 和 X11 选项的更多信息，请参阅 kbd(4) man page。

2.3. 其它资源

有关如何在 Red Hat Enterprise Linux 中配置键盘布局的详情，请查看以下列出的资源：

安装的文档

- localectl(1)- localectl 命令行实用程序的 man page 文档如何使用此工具配置系统区域设置和键盘布局。
- loadkeys(1)- loadkeys 命令的 man page 提供了有关如何使用此工具更改虚拟控制台中的键盘布局的更多信息。

另请参阅

- *第 6 章 获取特权 文档如何使用 su 和 sudo 命令获得管理权限。*

- *第 10 章 使用 systemd 管理服务 提供有关 systemd 的更多信息，以及如何使用 systemctl 命令来管理系统服务。*

第 3 章 配置日期和时间

现代操作系统区分以下两种时钟：

- 实时时钟 (RTC)，通常称为硬件时钟（通常是系统板上的集成电路），完全独立于操作系统的当前状态并在计算机关机时运行。

- 系统时钟 （也称为软件时钟 ）由内核维护，其初始值基于实时时钟。引导系统且系统时钟初始化后，系统时钟就完全独立于实时时钟。

系统时间始终保持在统一世界时间 (UTTC)中，并根据需要在应用程序中转换为本地时间。本地时间是当前时区的实际时间，考虑到夏天节省时间 (DST)。实时时钟可以使用 UTC 或本地时间。建议 UTC。

Red Hat Enterprise Linux 7 提供了三个命令行工具，可用于配置和显示有关系统日期和时间的信息：

- timedatectl 工具，它是 Red Hat Enterprise Linux 7 中的新功能，是 systemd 的一部分。

- 传统的 date 命令.

- 用于访问硬件时钟的 The hwclock 实用程序.

3.1. 使用 TIMEDATECTL 命令

timedatectl 工具作为 systemd 系统和服务管理器的一部分发布，可让您查看和更改系统时钟的配置。您可以使用此工具更改当前的日期和时间，设置时区，或者启用与远程服务器自动同步系统时钟。

有关如何以自定义格式显示当前日期和时间的详情，请参考 第 3.2 节 “使用 date 命令”。

3.1.1. 显示当前日期和时间

要显示当前的日期和时间，以及有关系统和硬件时钟配置的详细信息，请在没有附加命令行选项的情况下运行 timedatectl 命令：

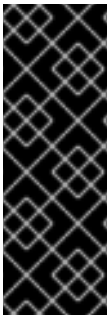
```
timedatectl
```

这会显示本地和通用时间、当前使用的时区、网络时间协议(NTP)配置的状态，以及与 DST 相关的其他信息。

例 3.1. 显示当前日期和时间

以下是一个不使用 NTP 将系统时钟与远程服务器同步的系统中 timedatectl 命令的输出示例：

```
~]$ timedatectl
  Local time: Mon 2016-09-16 19:30:24 CEST
Universal time: Mon 2016-09-16 17:30:24 UTC
   Timezone: Europe/Prague (CEST, +0200)
  NTP enabled: no
NTP synchronized: no
 RTC in local TZ: no
  DST active: yes
 Last DST change: DST began at
       Sun 2016-03-31 01:59:59 CET
       Sun 2016-03-31 03:00:00 CEST
 Next DST change: DST ends (the clock jumps one hour backwards) at
       Sun 2016-10-27 02:59:59 CEST
       Sun 2016-10-27 02:00:00 CET
```



重要

timedatectl 不会立即注意到对 chrony 或 ntpd 状态的更改。如果更改了这些工具的配置或状态，请输入以下命令：

```
~]# systemctl restart systemd-timedated.service
```

3.1.2. 更改当前时间

要更改当前时间，以 root 用户身份在 shell 提示符后输入以下内容：

```
timedatectl set-time HH:MM:SS
```

将 HH 替换为一小时，MM 替换为一分钟，SS 替换为秒，全部以两位数形式键入。

此命令同时更新系统时间和硬件时钟。结果类似于使用 date --set 和 hwclock --systohc 命令。

如果启用了 NTP 服务，该命令将失败。请参阅 第 3.1.5 节 “将系统时钟与远程服务器同步” 以临时禁用该服务。

例 3.2. 更改当前时间

要将当前时间更改为 11:26，以 root 用户身份运行以下命令：

```
~]# timedatectl set-time 23:26:00
```

默认情况下，系统配置为使用 UTC。要将您的系统配置为在本地时间维护时钟，以 root 用户身份使用 set-local-rtc 选项运行 timedatectl 命令：

```
timedatectl set-local-rtc boolean
```

要将您的系统配置为在本地时间维护时钟，请将 布尔值 替换为 yes （或者 y、y、true、t 或 1）。要将系统配置为使用 UTC，请将 布尔值 替换为 no （或者 n、n、false、f 或 0）。默认选项为 no。

3.1.3. 更改当前日期

要更改当前日期，以 root 用户身份在 shell 提示符后输入以下内容：

```
timedatectl set-time YYYY-MM-DD
```

将 YYY 替换为四位数年，MM 替换为两位数月，DD 替换为每月的两位数。

请注意，在不指定当前时间的情况下更改日期会导致将时间设置为 00:00:00。

例 3.3. 更改当前日期

要将当前日期更改为 2017 年 6 月 2 日并保留当前时间（下午 11:26），以 root 用户身份运行以下命令：

```
~]# timedatectl set-time "2017-06-02 23:26:00"
```

3.1.4. 更改时区

要列出所有可用时区，在 shell 提示符后输入以下内容：

```
timedatectl list-timezones
```

要更改当前使用的时区，以 root 用户身份输入 ：

```
timedatectl set-timezone time_zone
```

使用 time datectl list-timezones 命令列出的任何值替换 time _zone。

例 3.4. 更改时区

要确定哪一个时区与您当前位置最接近，请使用 timedatectl 命令和 list-timezones 命令行选项。例如，要列出欧洲所有可用时区，请输入：

```
~]# timedatectl list-timezones | grep Europe
Europe/Amsterdam
Europe/Andorra
Europe/Athens
Europe/Belgrade
Europe/Berlin
Europe/Bratislava
…
```

要将时区更改为 欧洲/地区， 以 root 用户身份输入 ：

```
~]# timedatectl set-timezone Europe/Prague
```

3.1.5. 将系统时钟与远程服务器同步

与前面部分所述的手动调整不同，timedatectl 命令还允许您使用 NTP 协议自动与一组远程服务器同步系统时钟。启用 NTP 可启用 chronyd 或 ntpd 服务，具体取决于安装了哪些服务。

使用以下命令可以启用或禁用 NTP 服务：

```
timedatectl set-ntp boolean
```

要让您的系统能够将系统时钟与远程 NTP 服务器同步，请将 布尔值 替换为 yes （默认选项）。要禁用此功能，请将 布尔值 替换为 no。

例 3.5. 将系统时钟与远程服务器同步

要启用与远程服务器自动同步系统时钟，请输入：

```
~]# timedatectl set-ntp yes
```

如果未安装 NTP 服务，命令将失败。如需更多信息，请参阅 第 18.3.1 节 “安装 chrony”。

3.2. 使用 DATE 命令

date 实用程序在所有 Linux 系统上可用，并允许您显示和配置当前日期和时间。它经常在脚本中使用，以自定义格式显示系统时钟的详细信息。

有关如何更改时区或启用与远程服务器自动同步系统时钟的详情请参考 第 3.1 节 “使用 timedatectl 命令”。

3.2.1. 显示当前日期和时间

要显示当前的日期和时间，请在没有附加命令行选项的情况下运行 date 命令：

```
date
```

这将显示星期几，后跟当前日期、本地时间、缩写时区和年份。

默认情况下，date 命令显示本地时间。要在 UTC 中显示时间，请使用 --utc 或 -u 命令行选项运行命令：

```
date --utc
```

您还可以在命令行中提供 +"格式" 选项来自定义显示信息的格式：

```
date +"format"
```

使用一个或多个支持的控制序列替换格式，如 例 3.6 “显示当前日期和时间” 所示。有关这些选项的完整列表，请参阅 表 3.1 “常用控制序列”，了解最常用格式化选项的列表，或 日期(1)手册页。

表 3.1. 常用控制序列

控制序列	描述
%H	*HH* 格式的小时（如 **17**）。
%M	*MM* 格式的分钟（如 **30**）。
%S	*SS* 格式的第二个版本（如 **24**）。
%d	*DD* 格式的月日（如 **16**）。
%m	*MM* 格式的月份（如 **09**）。
%Y	*YYYY* 格式的年份（例如： **2016 年**）。
%Z	时区缩写（如 **CEST**）。
%F	*YYYY-MM-DD* 格式的完整日期（例如 **2016-09-16**）。此选项等于 **%Y-%m-%d**。
%T	*HH:MM:SS* 格式的全职（例如 17:30:24）。这个选项等于 **%H:%M:%S**

例 3.6. 显示当前日期和时间

要显示当前日期和本地时间，在 shell 提示符后输入以下内容：

```
~]$ date
Mon Sep 16 17:30:24 CEST 2016
```

要在 UTC 中显示当前的日期和时间，在 shell 提示符后输入以下内容：

```
~]$ date --utc
Mon Sep 16 15:30:34 UTC 2016
```

要自定义 date 命令的输出，请输入：

```
~]$ date +"%Y-%m-%d %H:%M"
2016-09-16 17:30
```

3.2.2. 更改当前时间

要更改当前时间，以 root 用户身份使用 --set 或 -s 选项运行 date 命令：

```
date --set HH:MM:SS
```

将 HH 替换为一小时，MM 替换为一分钟，SS 替换为秒，全部以两位数形式键入。

默认情况下，date 命令会将系统时钟设置为本地时间。要在 UTC 中设置系统时钟，请使用 --utc 或 -u 命令行选项运行命令：

```
date --set HH:MM:SS --utc
```

例 3.7. 更改当前时间

要将当前时间更改为 11:26，以 root 用户身份运行以下命令：

```
~]# date --set 23:26:00
```

3.2.3. 更改当前日期

要更改当前日期，以 root 用户身份使用 --set 或 -s 选项运行 date 命令：

```
date --set YYYY-MM-DD
```

将 YYY 替换为四位数年，MM 替换为两位数月，DD 替换为每月的两位数。

请注意，在不指定当前时间的情况下更改日期会导致将时间设置为 00:00:00。

例 3.8. 更改当前日期

要将当前日期更改为 2017 年 6 月 2 日并保留当前时间（下午 11:26），以 root 用户身份运行以

下命令：

```
~]# date --set "2017-06-02 23:26:00"
```

3.3. 使用 HWCLOCK 命令

hwclock 是访问硬件时钟的实用程序，也称为实时时钟(RTC)。硬件时钟独立于您使用的操作系统，即使在机器关闭时也能正常工作。此实用程序用于显示硬件时钟的时间。Hwclock 还包含用于补偿硬件时钟中系统偏移的功能。

硬件时钟存储以下值：year、month、day、hour、minute 和 second。它无法存储时间标准、本地时间或协调的通用时间(UTC)，也无法设置夏天制(DST)。

The hwclock 实用程序将其设置保存在 /etc/adjtime 文件中，该文件会在您进行第一次更改时创建，例如，当您手动设置时间或将硬件时钟与系统时间同步时。

注意

有关 Red Hat Enterprise Linux 6 和 7 之间的 hwclock 行为更改，请参阅 *Red Hat Enterprise Linux 7 迁移规划指南*。

3.3.1. 显示当前日期和时间

由于 root 用户将日期和时间返回本地时间的日期和时间到标准输出，因此不带命令行选项的 Running hwclock。

```
hwclock
```

请注意，在 hwclock 命令中使用 --utc 或 --localtime 选项并不意味着您在 UTC 或本地时间显示硬件时钟时间。这些选项用于设置硬件时钟，以在其中任何一个中保持时间。时间始终在本地时间显示。另外，使用 hwclock --utc or hwclock --local 命令不会更改 /etc/adjtime 文件中的记录。如果您知道 /etc/adjtime 中保存的设置不正确，但您不想更改设置，此命令非常有用。另一方面，如果以错误的方式使用命令，您可能会收到误导的信息。详情请查看 hwclock(8)手册页。

例 3.9. 显示当前日期和时间

要显示当前日期以及硬件时钟的当前本地时间，以 root 用户身份运行：

```
~]# hwclock
Tue 15 Apr 2017 04:23:46 PM CEST  -0.329272 seconds
```

CEST 是时区缩写，代表中欧夏季时间。

有关如何更改时区的详情请参考 第 3.1.4 节 “更改时区”。

3.3.2. 设置日期和时间

除了显示日期和时间外，您还可以手动将硬件时钟设置为特定的时间。

当您需要更改硬件时钟日期和时间时，您可以根据您的规格附加 --set 和 --date 选项：

```
hwclock --set --date "dd mmm yyyy HH:MM"
```

将 dd 替换为一天（一个两位数），mmm 替换为一个月（一个三字母缩写），yyy 替换为一年（四位数字）、HH 替换为一小时（两位数），MM 替换为分钟（两位数）。

同时，您还可以通过添加 --utc 或 -- localtime 选项，将硬件时钟设置为在 UTC 或本地时间中保持时间。在这种情况下，UTC 或 LOCAL 会记录在 /etc/adjtime 文件中。

例 3.10. 将硬件时钟设置为特定日期和时间

如果要将日期和时间设置为特定值，例如："21:17, 2016 年 10 月 21 日"，并在 UTC 中保持硬件时钟，以 root 用户身份以以下格式运行命令：

```
~]# hwclock --set --date "21 Oct 2016 21:17" --utc
```

3.3.3. 同步日期和时间

您可以在两个方向上同步硬件时钟和当前系统时间。

- 您可以使用这个命令将硬件时钟设置为当前系统时间：

```
hwclock --systohc
```

请注意，如果您使用 NTP，硬件时钟每 11 分钟会自动同步到系统时钟中，此命令仅在启动时用于获得合理的初始系统时间。

- 或者，您可以使用以下命令从硬件时钟设置系统时间：

```
hwclock --hctosys
```

当您同步硬件时钟和系统时间时，您还可以通过添加 --utc 或 -- localtime 选项来指定是否要将硬件时钟保留在本地时间或 UTC 中。与使用 --set 类似，UTC 或 LOCAL 也会记录在 /etc/adjtime 文件中。

The hwclock --systohc --utc 命令的功能类似于 timedatectl set-local-rtc false，s hwclock --systohc --local 命令是 timedatectl set-local-rtc true 的替代选择。

例 3.11. 将硬件时钟与系统时间同步

要将硬件时钟设置为当前系统时间并保留硬件时钟在本地时间，以 root 用户身份运行以下命令：

```
~]# hwclock --systohc --localtime
```

为避免时区和 DST 切换出现问题，建议在 UTC 中保持硬件时钟。显示的 **例 3.11 “将硬件时钟与系统时间同步”** 很有用，例如，如果使用 Windows 系统进行多次引导，且假设硬件时钟默认在本地时间运行，所有其他系统还需要使用本地时间容纳它。虚拟机也可能需要它；如果主机提供的虚拟硬件时钟正在本地时间运行，则还需要将 guest 系统配置为使用本地时间。

3.4. 其它资源

有关如何在 Red Hat Enterprise Linux 7 中配置日期和时间的详情，请查看以下列出的资源。

安装的文档

- timedatectl(1)- timedatectl 命令行工具的 man page 记录了如何使用该工具查询和更改系统时钟及其设置。

- *日期(1)- date 命令的 man page 提供了受支持命令行选项的完整列表。*

- *hwclock(8)- the hwclock 命令的 man page 提供了所支持的命令行选项的完整列表。*

另请参阅

- *第 2 章 系统位置和键盘配置 文档如何配置键盘布局。*

- *第 6 章 获取特权 文档如何使用 su 和 sudo 命令获得管理权限。*

- *第 10 章 使用 systemd 管理服务 提供有关 systemd 的更多信息，以及如何使用 systemctl 命令来管理系统服务。*

第 4 章 管理用户和组

用户和组群的控制是 Red Hat Enterprise Linux 系统管理的核心元素。本章解释了如何在图形用户界面和命令行中添加、管理和删除用户和组，并介绍高级主题，如创建组目录。

4.1. 用户和组介绍

虽然用户可以是用户（这意味着与物理用户相关联的帐户），或者是特定应用使用的帐户，但组是组织的逻辑表达式，将用户连接在一起以实现共同目的。组中的用户共享相同的读取、写入或执行该组所拥有的文件的权限。

每个用户都与一个唯一数字标识号关联，称为用户 ID (UID)。类似地，每个组都与组 ID(GID)关联。创建文件的用户也是该文件的所有者和组所有者。文件会为所有者、组和其他任何人单独分配读取、写入和执行权限。文件所有者只能由 root 更改，并且 root 用户和文件所有者都可以更改访问权限。

此外，Red Hat Enterprise Linux 支持文件和目录的访问控制列表(ACL)，它们允许设置所有者之外的特定用户的权限。有关这个功能的详情请参考 第 5 章 访问控制列表。

保留的用户和组群 ID

Red Hat Enterprise Linux 为系统用户和组保留 1000 以下的用户和组群 ID。默认情况下，用户管理器 不显示系统用户。保留的用户和组 ID 记录在 setup 软件包中。要查看文档，请使用这个命令：

```
cat /usr/share/doc/setup*/uidgid
```

建议的做法是分配尚未保留的 5,000 个 ID，因为保留范围将来可能会增加。要使分配给新用户的 ID 默认从 5,000 开始，请更改 /etc/login.defs 文件中的 UID _MIN 和 GID_MIN 指令：

```
[file contents truncated]
UID_MIN         5000
[file contents truncated]
GID_MIN         5000
[file contents truncated]
```

注意

对于在更改 UID_MIN 和 GID_MIN 指令之前创建的用户，UID 仍会从默认的 1000 开始。

即使使用以 5,000 开头的新用户和组 ID，建议不要提高 1000 以上系统保留的 ID，以避免与保留 1000 限制的系统冲突。

4.1.1. 用户专用组

Red Hat Enterprise Linux 使用用户专用组 (UPG)方案，这使 UNIX 组更易于管理。无论何时在系统中添加新用户，都会创建一个用户私人组群。它的名称与为其创建的用户的名称相同，并且该用户是用户专用组的唯一成员。

用户专用组可以安全地为新创建的文件或目录设置默认权限，从而允许该用户的用户和组对文件或目录进行修改。

此设置决定了哪些权限应用到新创建的文件或目录，称为 umask，并在 /etc/bashrc 文件中配置。通常在基于 UNIX 的系统上，umask 设置为 022，它只允许创建文件或目录的用户进行修改。在此方案下，所有其他用户（包括创建者组的成员）不得进行任何修改。但是，根据 UPG 方案，这种"组保护"并不是必需的，因为每个用户都有自己的专用组。如需更多信息，请参阅 第 4.3.5 节 “使用 umask 为新文件设置默认权限”。

所有组的列表存储在 /etc/group 配置文件中。

4.1.2. shadow 密码

在有多个用户的环境中，使用 shadow -utils 软件包提供的shadow 密码 来增强系统身份验证文件的安全性非常重要。因此，安装程序默认启用 shadow 密码。

以下是影子密码相对于传统在基于 UNIX 的系统中存储密码的优势列表：

- 影子密码通过将加密的密码哈希从全局可读的 /etc/passwd 文件移动到 /etc/ shadow，这仅可由 root 用户读取，从而提高系统安全性。
- 影子密码存储有关密码有效期的信息.
- 影子密码允许实施 /etc/login.defs 文件中设置的一些安全策略。

shadow-utils 软件包提供的大多数实用程序可以正常工作，无论是否启用了 shadow 密码。但是，由于密码过期信息只存储在 /etc/shadow 文件中，一些实用程序和命令在未首先启用影子密码的情况下无

法正常工作：

- 用于设置密码期限参数的 chage 实用程序。详情请查看 Red Hat Enterprise Linux 7 安全指南中 的密码 安全部分。
- 用于管理 /etc/group 文件的 gpasswd 实用程序。
- usermod 命令带有 -e、--expiredate 或 -f, --inactive 选项。
- useradd 命令带有 -e、--expiredate 或 -f, --inactive 选项。

4.2. 在图形环境中管理用户

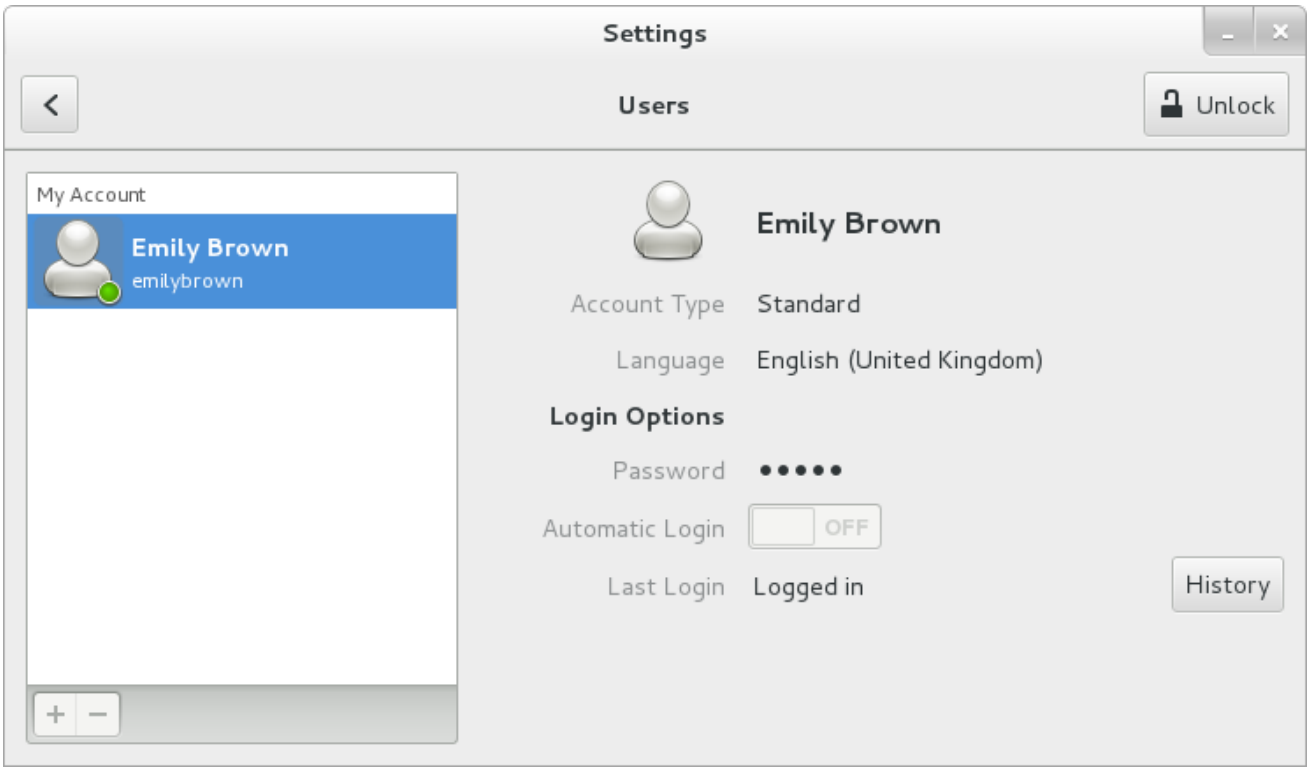
通过 Users 实用程序，您可以在图形用户界面中查看、修改、添加和删除本地用户。

4.2.1. 使用用户设置工具

按 Super 键以进入活动概览，键入 Users，然后按 Enter 键。此时 会出现用户 设置工具。Super 键显示在各种 guis 中，具体取决于键盘和其他硬件，但通常作为 Windows 或 Command 键，通常在空格的左侧。或者，也可以在点击屏幕右上角的用户名后，从 Settings 菜单打开 Users 实用程序。

要更改用户帐户，请先选择"解除锁定 "按钮，然后按照显示的对话框进行自己验证。请注意，除非您具有超级用户特权，否则应用将提示您以 root 身份进行身份验证。要添加和删除用户，可分别选择 + 和 - 按钮。要将用户添加到管理组 wheel 中，请将 Account Type 从 Standard 更改为 Administrator。要编辑用户的语言设置，请选择语言，然后显示一个下拉菜单。

图 4.1. 用户设置工具



创建新用户时，该帐户会在设置密码之前禁用。"密码"下拉菜单（如 图 4.2 “密码菜单” 所示）包含设置管理员密码、在第一次登录时立即选择密码的选项，或者创建不需要登录的客户机帐户。您还可以从此菜单中禁用或启用帐户。

图 4.2. 密码菜单

4.3. 使用命令行工具

除了 第 4.2 节 “在图形环境中管理用户” 中描述的 用户设置工具（用于基本管理用户 ）外，您还可以使用命令行工具管理 表 4.1 “管理用户和组的命令行工具” 中列出的用户和组。

表 4.1. 管理用户和组的命令行工具

工具	描述
id	显示用户和组 ID.
useradd, **usermod**, **userdel**	用于添加、修改和删除用户帐户的标准实用程序.
groupadd、**groupmod**、**groupdel**	用于添加、修改和删除组的标准实用程序.
gpasswd	实用程序主要用于修改由 **newgrp 命令使用的 /etc/g shadow** 文件中的组密码。
pwck,**grpck**	可用于验证密码、组和相关影子文件的实用程序.

工具	描述
pwconv, **pwunconv**	实用程序，可用于将密码转换为影子密码，或者从影子密码转换回标准密码。
grpconv, **grpunconv**	与前面的 类似，这些实用程序可用于转换组帐户的影子信息。

4.3.1. 添加新用户

要在系统中添加新用户，以 root 用户身份在 shell 提示符后输入以下内容：

```
useradd options username
```

...其中选项是命令行选项，如 表 4.2 “常用 useradd 命令行选项” 所述。

默认情况下，user add 命令会创建一个锁定的用户帐户。要解锁帐户，以 root 用户身份运行以下命令来分配密码：

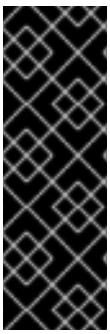
```
passwd username
```

（可选）您可以设置密码过期策略。请参阅 Red Hat Enterprise Linux 7 安全指南中 的密码安全部分。

表 4.2. 常用 useradd 命令行选项

选项	
-c *'评论'*	*注释可以被替换为任何字符串。*此选项通常用于指定用户的全名。
-d *home_directory*	要使用的主目录，而不是默认的 **/home/*username*/**。
-e *date*	以 YYYY-MM-DD 格式禁用的帐户的日期。
-f *days*	密码到期之后的天数，直到禁用帐户为止.如果指定了 **0**，则帐户会在密码过期后立即禁用。如果指定了 **-1**，则在密码过期后不会禁用帐户。
-g *group_name*	用户默认（主要）组的组名或组号。组必须在此处指定之前存在。

选项	
-G *group_list*	以逗号分开的额外（补充，非默认）组名或组号列表，用户是其中的成员。组必须在此处指定之前存在。
-m	如果主目录不存在，则创建该目录。
-M	不要创建主目录。
-N	不要为用户创建用户专用组。
-p *password*	通过 **crypt 加密的密码**。
-r	创建 UID 小于 1000 且没有主目录的系统帐户。
-s	用户的登录 shell，默认为 **/bin/bash**。
- *UUID*	用户的用户 ID，它必须唯一且大于 999。



重要

在 Red Hat Enterprise Linux 7 中，系统和普通用户的默认 ID 范围已从早期版本中更改。在以前的版本中，UID 1-499 用于系统用户和以上正常用户的值。系统用户的默认范围现在是 1-999。因为这个变化，当迁移到 Red Hat Enterprise Linux 7 时，如果有用户的 UID 和 GID 在 500 到 999 之间，则可能会造成问题。UID 和 GID 的默认范围可以在 /etc/login.defs 文件中更改。

解释进程

以下步骤演示了在启用了影子密码的系统上发出 useradd juan 命令时会发生什么情况：

1. ***在 /etc/passwd 中为 juan 创建一个新行：***

```
juan:x:1001:1001::/home/juan:/bin/bash
```

该行具有以下特征：

- ***它以用户名 juan 开始。***

- 密码字段有一个 x 表示系统正在使用 shadow 密码。
- 创建大于 999 的 UID。在 Red Hat Enterprise Linux 7 下，1000 以下的 UID 保留给系统使用，不应分配给用户。
- 创建大于 999 的 GID。在 Red Hat Enterprise Linux 7 下，1000 以下的 GID 保留给系统使用，不应分配给用户。
- 可选的 GECOS 信息留空。GECOS 字段可用于提供有关用户的其他信息，如用户的全名或电话号码。
- juan 的主目录设为 /home/juan/。
- 默认 shell 设置为 /bin/bash。

2. 在 /etc/shadow 中为 juan 创建一个新行：

```
juan:!!:14798:0:99999:7:::
```

该行具有以下特征：

- 它以用户名 juan 开始。
- 两个感叹号(!!)显示在 /etc/shadow 文件的密码字段中，该字段将锁定帐户。

注意

如果使用 -p 标志传递加密的密码，则会将其放置在用户新行中的 /etc/shadow 文件中。

- 密码设置为永不过期。

3. 在 /etc/group 中为名为 juan 的组创建一个新行：

```
juan:x:1001:
```

与用户同名的组称为用户专用组。有关用户私人组群的详情请参考 第 4.1.1 节 “用户专用组”。

在 /etc/group 中创建的行具有以下特征：

- 它以组名 juan 开头。
- 在密码字段中会出现 x，表示系统正在使用 shadow 组密码。
- GID 与 /etc/passwd 中为 juan 的主组列出的匹配。

4. 在 /etc/gshadow 中为名为 juan 的组创建一个新行：

```
juan:!::
```

该行具有以下特征：

- 它以组名 juan 开头。
- 感叹号(!)显示在 /etc/gshadow 文件的密码字段中，该文件将锁定组。
- 所有其他字段均为空。

5. 在 /home 目录中为用户 juan 创建一个目录：

```
~]# ls -ld /home/juan
drwx------. 4 juan juan 4096 Mar 3 18:23 /home/juan
```

该目录归用户 juan 和组 juan 所有。它仅对用户 juan 具有读取、写入和执行特权。所有其他权限都将被拒绝。

6. */etc/skel/ 目录中的文件（包含默认用户设置）复制到新的 /home/juan/ 目录中：*

```
~]# ls -la /home/juan
total 28
drwx------. 4 juan juan 4096 Mar 3 18:23 .
drwxr-xr-x. 5 root root 4096 Mar 3 18:23 ..
-rw-r--r--. 1 juan juan  18 Jun 22 2010 .bash_logout
-rw-r--r--. 1 juan juan 176 Jun 22 2010 .bash_profile
-rw-r--r--. 1 juan juan 124 Jun 22 2010 .bashrc
drwxr-xr-x. 4 juan juan 4096 Nov 23 15:09 .mozilla
```

此时系统上存在名为 juan 的锁定帐户。要激活它，管理员接下来必须使用 passwd 命令为帐户分配密码，也可以设置密码过期指南（详情请参阅 Red Hat Enterprise Linux 7 安全指南中的密码安全部分）。

4.3.2. 添加新组

要在系统中添加新组，以 root 用户身份在 shell 提示符后输入以下内容：

```
groupadd options group_name
```

...其中选项是命令行选项，如表 4.3 “常用 groupadd 命令行选项” 所述。

表 4.3. 常用 groupadd 命令行选项

选项	描述
-f,--force	与 **-g** *gid* 和 *gid* 搭配使用时，**groupadd** 将为 *组* 选择另一个唯一。
-g *gid*	组的组 ID，它必须唯一且大于 999。
-k,--key *key=value*	覆盖 **/etc/login.defs** defaults。
-o,--non-unique	允许使用重复的 GID 创建组.
-p,--password *password*	将此加密密码用于新组。
-r	创建 GID 小于 1000 的系统组。

4.3.3. 将现有用户添加到现有组中

使用 usermod 实用程序将现有用户添加到现有的组中。

usermod 的各种选项对用户的主组以及他/她的补充组有不同影响。

要覆盖用户的主组群，以 root 用户身份运行以下命令：

```
~]# usermod -g group_name user_name
```

要覆盖用户的补充组，以 root 用户身份运行以下命令：

```
~]# usermod -G group_name1,group_name2,... user_name
```

请注意，在这种情况下，用户的所有前一个补充组都会被新组或多个新组替代。

要在用户的附加组中添加一个或多个组，以 root 用户身份运行以下命令之一：

```
~]# usermod -aG group_name1,group_name2,... user_name
```

```
~]# usermod --append -G group_name1,group_name2,... user_name
```

请注意，在这种情况下，新组会添加到用户的当前补充组中。

4.3.4. 创建组目录

系统管理员通常喜欢为每个主要项目创建组，并在需要访问该项目的文件时将人员分配给组。采用这种传统方案时，文件管理比较困难；当有人创建文件时，它与其所属的主要组相关联。当单个人处理多个项目时，将正确的文件与正确的组关联变得困难。但是，使用 UPG 方案时，组会自动分配到设置了 setgid 位的目录中创建的文件。setgid 位使得管理共享一个通用目录的组项目非常简单，因为用户在该目录中创建的任何文件都归拥有该目录的组所有。

例如，一组人需要处理 /opt/myproject/ 目录中的文件。些人信任修改此目录的内容，但不是每个人。

1. 以 root 用户身份，在 shell 提示符下键入以下内容来创建 /opt/myproject/ 目录：

```
mkdir /opt/myproject
```

2. 在系统中添加 myproject 组：

```
groupadd myproject
```

3. 将 /opt/myproject/ 目录的内容与 myproject 组关联：

```
chown root:myproject /opt/myproject
```

4. 允许组中的用户在目录中创建文件并设置 setgid 位 ：

```
chmod 2775 /opt/myproject
```

此时，myproject 组 的所有成员都可以在 /opt/myproject/ 目录中创建和编辑文件，无需管理员每次用户写入新文件时更改文件权限。要验证权限是否已正确设置，请运行以下命令：

```
~]# ls -ld /opt/myproject
drwxrwsr-x. 3 root myproject 4096 Mar 3 18:31 /opt/myproject
```

5. 将用户添加到 myproject 组：

```
usermod -aG myproject username
```

4.3.5. 使用 umask为新文件设置默认权限

当进程创建文件时，该文件具有特定的默认权限，例如 -rw-rw-r--。这些初始权限部分由文件模式创建掩码定义，也称为文件权限掩码或 umask。每个进程都有自己的 umask，例如，默认情况下 bash 具有 umask 0022。可以更改进程 umask。

umask 包括什么

umask 由与标准文件权限对应的位组成。例如，对于 umask 0137，数字表示：

- 0 = 无含义，它始终为 0 （umask 不影响特殊位）
- 1 = 对于所有者权限，将设置执行位
- 3 = 对于组权限，会设置执行位和写入位
- 7 = 对于其他权限，将设置执行、写入和读取位

Umasks 可以用二进制、八进制或符号表示法表示。例如，八进制表示 0137 等于符号表示 u=rw-,g=r--,o=---。符号表示规格与八进制表示规范相反：它显示允许的权限，而不是禁止的权限。

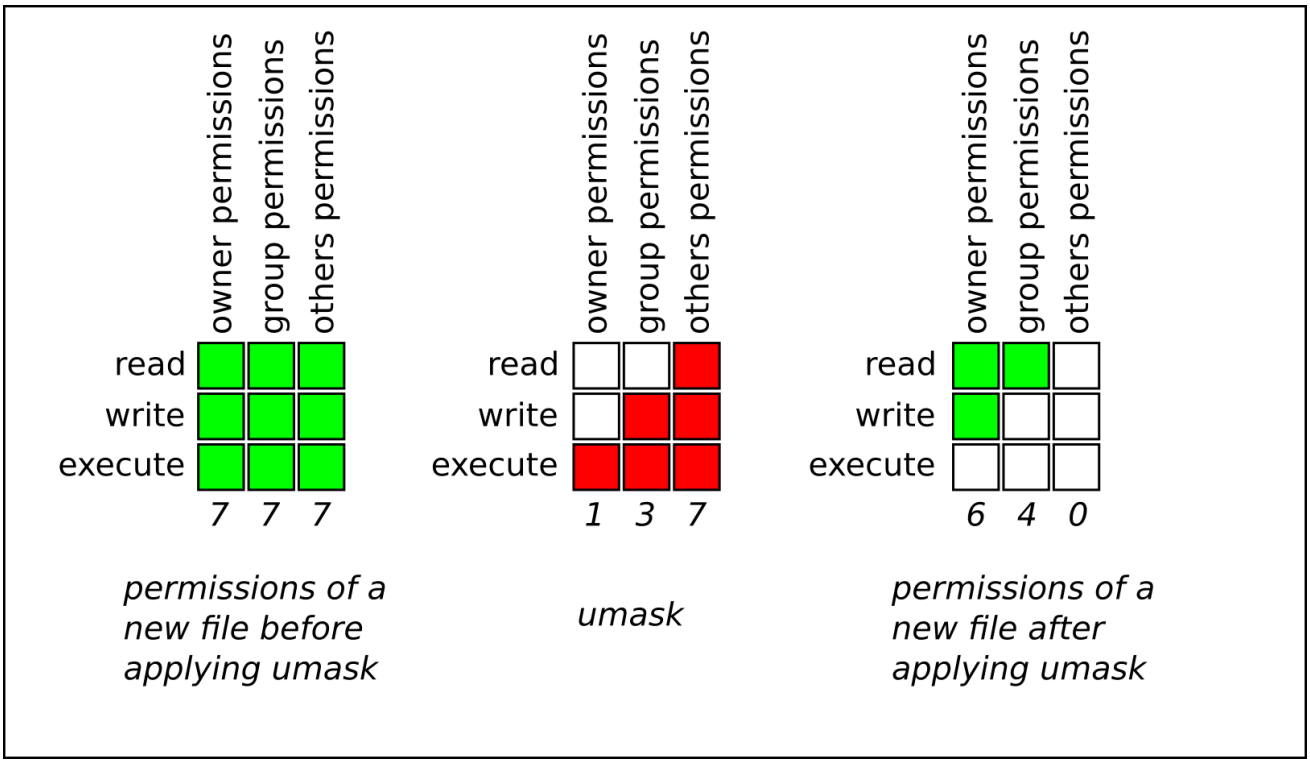
umask 的工作原理

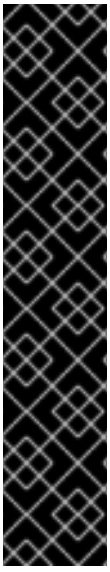
umask 禁止为文件设置权限 ：

- 在 umask 中设置了一个位时，该文件中将取消设置。
- 在 umask 中未设置位时，可以在 文件中设置它，具体取决于其他因素。

下图显示了 umask 0137 如何影响新文件的创建。

图 4.3. 在创建文件时应用 umask





重要

出于安全原因，在默认情况下，常规文件将不能拥有执行权限。因此，即使 umask 为 0000（不禁止任何权限），新的常规文件仍不具有执行权限。但是，可以创建具有执行权限的目录：

```
[john@server tmp]$ umask 0000
[john@server tmp]$ touch file
[john@server tmp]$ mkdir directory
[john@server tmp]$ ls -lh .
total 0
drwxrwxrwx. 2 john john 40 Nov 2 13:17 directory
-rw-rw-rw-. 1 john john 0 Nov 2 13:17 file
```

4.3.5.1. 在 Shell 中管理 umask

对于常用的 shell，如 bash、ksh、zsh 和 tcsh，umask 使用 umask shell 内置进行管理。从 shell 启动的进程继承其 umask。

显示当前掩码

使用数值表示法显示当前的 umask：

```
~]$ umask
0022
```

使用符号表示法显示当前的 umask:

```
~]$ umask -S
u=rwx,g=rx,o=rx
```

使用 umask 在 shell 中设置掩码

使用数值表示法运行为当前 shell 会话设置 umask:

```
~]$ umask octal_mask
```

用从 0 到 7 的四位或更少数字替换 octal_mask。提供三个或更少的数字时，权限会像命令包含前导零一样设置。例如，umask 7 转换为 0007。

例 4.1. 使用 Octal 表示设置 umask

要禁止新文件拥有所有者和组的写入和执行权限，并且不允许其他人拥有任何权限：

```
~]$ umask 0337
```

或只需：

```
~]$ umask 337
```

使用符号表示法为当前 shell 会话设置 umask ：

```
~]$ umask -S symbolic_mask
```

例 4.2. 使用符号链接表示设置 umask

使用符号表示法设置 umask 0337 ：

```
~]$ umask -S u=r,g=r,o=
```

使用默认的 shell umask

Shell 通常有一个配置文件，其中设置了其默认 umask。对于 bash，它是 /etc/bashrc。显示默认

bash umask:

```
~]$ grep -i -B 1 umask /etc/bashrc
```

输出显示是否设置了 umask，可使用 umask 命令或 UMASK 变量。在以下示例中，使用 umask 命令将 umask 设置为 022 ：

```
~]$ grep -i -B 1 umask /etc/bashrc
 # By default, we want umask to get set. This sets it for non-login shell.
--
 if [ $UID -gt 199 ] && [ "id -gn" = "id -un" ]; then
  umask 002
 else
  umask 022
```

要更改 bash 的默认 umask，请在 /etc/bashrc 中更改 umask 命令调用或 UMASK 变量分配。本例将默认 umask 更改为 0227 ：

```
 if [ $UID -gt 199 ] && [ "id -gn" = "id -un" ]; then
  umask 002
 else
  umask 227
```

使用特定用户的默认 shell umask

默认情况下，新用户的 bash umask 默认为 /etc/bashrc 中定义的 umask。

要为 特定用户更改 bash umask，请在用户的 $HOME/.bashrc 文件中添加对 umask 命令的调用。例如，将用户 john 的 bash umask 更改为 0227 ：

```
john@server ~]$ echo 'umask 227' >> /home/john/.bashrc
```

为新创建的主目录设置默认权限

要更改创建用户主目录的权限，请在 /etc/login.defs 文件中更改 UMASK 变量：

```
# The permission mask is initialized to this value. If not specified,
# the permission mask will be initialized to 022.
UMASK 077
```

4.4. 其它资源

有关如何在 Red Hat Enterprise Linux 中管理用户和组的详情，请查看以下列出的资源。

安装的文档

有关管理用户和组的各种工具的详情，请查看以下 man page：

- useradd(8)- useradd 命令的 man page 描述了如何使用它创建新用户。
- userdel(8)- userdel 命令的 man page 描述了如何使用它删除用户。
- usermod(8)- usermod 命令的 man page 描述了如何使用它修改用户。
- groupadd(8)- groupadd 命令的 man page 描述了如何使用它创建新组。
- groupdel(8)- groupdel 命令的 man page 描述了如何使用它删除组。
- groupmod(8)- groupmod 命令的 man page 包括了如何使用它修改组成员资格。
- gpasswd(1)- gpasswd 命令的 man page 描述了如何管理 /etc/group 文件。
- grpck(8)- grpck 命令的 man page 记录了如何使用它来验证 /etc/group 文件的完整性。
- pwck(8)- pwck 命令的 man page 描述了如何使用它来验证 /etc/passwd 和 /etc /shadow 文件的完整性。
- pwconv(8)- pwconv、pw unconv、grpconv 和 grpunconv 命令的手册页记录了如何转换密码和组的影子信息。
- ID(1)- id 命令的 man page 记录了如何显示用户和组 ID。

- umask(2)- umask 命令的 man page 记录了如何使用文件模式创建掩码。

有关相关配置文件的详情请参考：

- 组(5)- /etc/group 文件的 man page 文档如何使用此文件定义系统组。
- passwd(5)- /etc/passwd 文件的 man page 文档如何使用此文件定义用户信息。
- Shadow(5)- /etc/shadow 文件的 man page 文档如何使用此文件为系统设置密码和帐户过期信息。

在线文档

- 红帽企业 Linux 7 安全指南 - 红帽企业 Linux 7 的安全 指南 提供了其他信息，如何通过启用密码期限和用户帐户锁定来确保工作站的安全。

另请参阅

- 第 6 章 获取特权 文档如何使用 su 和 sudo 命令获得管理权限。

第 5 章 访问控制列表

文件和目录为文件所有者、与 文件关联的组以及系统的所有其他用户设置了权限集。但是，这些权限集存在限制。例如，无法为不同的用户配置不同的权限。因此，实施了访问控制列表 (ACL)。

Red Hat Enterprise Linux 内核为 ext3 文件系统和 NFS 导出的文件系统提供 ACL 支持。通过 Samba 访问的 ext3 文件系统也可识别 ACL。

除了内核中的支持外，还需要 acl 软件包来实现 ACL。它包含用于添加、修改、删除和检索 ACL 信息的实用程序。

cp 和 mv 命令复制或移动任何与文件和目录关联的 ACL。

5.1. 挂载文件系统

在将 ACL 用于文件或目录之前，文件或目录的分区必须使用 ACL 支持进行挂载。如果它是一个本地 ext3 文件系统，它可以使用以下命令挂载：

```
mount -t ext3 -o acl device-name 分区
```

例如：

```
mount -t ext3 -o acl /dev/VolGroup00/LogVol02 /work
```

或者，如果分区在 /etc/fstab 文件中列出，则分区的条目可以包含 acl 选项：

```
LABEL=/work   /work    ext3  acl    1 2
```

如果通过 Samba 访问 ext3 文件系统并且为其启用了 ACL，则识别 ACL，因为 Samba 已使用 --with-acl-support 选项编译。访问或挂载 Samba 共享时不需要特殊标志。

5.1.1. NFS

默认情况下，如果 NFS 服务器导出的文件系统支持 ACL，并且 NFS 客户端可以读取 ACL，客户端系统将使用 ACL。

要在配置服务器时禁用 NFS 共享上的 ACL，请在 /etc/exports 文件中包含 no_acl 选项。要在客户端上挂载 NFS 共享时禁用 ACL，请通过命令行或 /etc/fstab 文件使用 no_acl 选项进行挂载。

5.2. 设置访问权限 ACL

ACL 有两种类型：访问 ACL 和默认 ACL。访问 ACL 是特定文件或目录的访问控制列表。默认 ACL 只能与目录关联；如果目录中的文件没有访问权限 ACL，它将使用目录的默认 ACL 规则。默认 ACL 是可选的。

可以配置 ACL：

1. 每个用户
2. 每个组
3. 通过有效的权限掩码
4. 对于不在文件用户组中的用户

setfacl 实用程序为文件和目录设置 ACL。使用 -m 选项添加或修改文件或目录的 ACL：

```
# setfacl -m rules files
```

规则（规则）必须以以下格式指定：如果通过逗号分隔多个规则，则可以在同一命令中指定多个规则。

u:uid:perms

设置用户的访问权限 ACL。可以指定用户名或 UID。用户可以是系统上的任何有效用户。

g:gid:perms

设置组的访问权限 ACL。可以指定组名或 GID。组可以是系统上的任何有效组。

m:perms

设置有效的权利掩码。掩码是所属组以及所有用户和组条目的所有权限的并集。

O:perms

为 组中不属于 文件的用户设置访问权限 ACL。

权限(perms)必须是字符 r、w 和 x 的组合，用于读取、写入和执行。

如果文件或目录已具有 ACL，并且使用了 setfacl 命令，则其他规则将添加到现有 ACL 中或修改现有规则。

例 5.1. 授予读取和写入权限

例如，为用户和rius 授予读取和写入权限：

```
# setfacl -m u:andrius:rw /project/somefile
```

要删除用户、组群或其他用户的所有权限，请使用 **-x** 选项且不指定任何权限：

```
# setfacl -x rules files
```

例 5.2. 删除所有权限

例如，要从 UID 500 的用户中删除所有权限：

```
# setfacl -x u:500 /project/somefile
```

5.3. 设置默认 ACL

要设置默认 ACL，请在规则前面添加 d: 并指定目录而不是文件名。

例 5.3. 设置默认 ACL

例如，要将 /share/ 目录的默认 ACL 设置为对不属于用户组的用户读取和执行（单个文件的访问 ACL 可以覆盖）：

```
# setfacl -m d:o:rx /share
```

5.4. 检索 ACL

要确定文件或目录的现有 ACL，请使用 getfacl 命令。在以下示例中，getfacl 用于确定文件的现有 ACL。

例 5.4. 检索 ACL

```
# getfacl home/john/picture.png
```

以上命令返回以下输出：

```
# file: home/john/picture.png
# owner: john
# group: john
user::rw-
group::r--
other::r--
```

如果指定了具有默认 ACL 的目录，则默认 ACL 也会显示如下所述。例如，getfacl home/sales/ 将显示类似的输出：

```
# file: home/sales/
# owner: john
# group: john
user::rw-
user:barryg:r--
group::r--
mask::r--
other::r--
default:user::rwx
default:user:john:rwx
default:group::r-x
default:mask::rwx
default:other::r-x
```

5.5. 使用 ACL 归档文件系统

默认情况下，转储命令现在在备份操作期间保留 ACL。使用 tar 归档文件或文件系统时，请使用 --acls 选项来保留 ACL。类似地，使用 cp 通过 ACL 复制文件时，请包含 --preserve=mode 选项，以确保在之后复制 ACL。此外，cp 的 -a 选项（等同于 -dR --preserve=all）也会在备份期间保留 ACL 以及其他信息，如时间戳、SELinux 上下文等。有关转储、tar 或 cp 的更多信息，请参考其各自的 man page。

星级实用程序与 tar 实用程序类似，它可以用于生成文件存档；但是，其某些选项有所不同。有关更常用的选项列表，请参阅 表 5.1 “星号命令行选项”。有关所有可用选项，请参阅 man 星。需要星级软件包才能使用此实用程序。

表 5.1. 星号命令行选项

选项	描述
-c	创建存档文件.
-n	不要提取文件；将 与 **-x** 结合使用来显示提取文件的作用。
-r	替换存档中的文件.文件写入存档文件的末尾，用相同路径和文件名替换所有文件。
-t	显示存档文件的内容.
-u	更新存档文件。如果存档中不存在这些文件，或者文件比存档中相同名称的文件更新，则文件将写入存档的末尾。只有在归档是文件或者可能后退空间的未阻塞磁带时才起作用。
-x	从存档中提取文件。如果与 **-U** 和存档中的文件一起使用，则不会提取该文件。
-help	显示最重要的选项。
-xhelp	显示最小选项。
-/	从存档中提取文件时，请勿从文件名剥离前导斜杠。默认情况下，提取文件时会剥离它们。
-ACL	创建或提取时，归档或恢复与文件和目录关联的任何 ACL。

5.6. 与旧系统的兼容性

如果在给定文件系统上的任何文件上设置了 ACL，则该文件系统具有 ext_attr 属性。使用以下命令查看此属性：

```
# tune2fs -l filesystem-device
```

获取 ext_attr 属性的文件系统可以使用较旧的内核挂载，但这些内核不会强制实施任何已设置的 ACL。

1.22 版及 e2 fsprogs 软件包（包括红帽企业 Linux 2.1 和 4 中的版本）中包含的 e2fs ck 实用程序的版本可以检查具有 ext_attr 属性的文件系统。旧版本拒绝检查。

5.7. ACL 参考

详情请参考以下 man page。

- man acl - ACL 的说明
- man getfacl - 讨论如何获取文件访问控制列表
- man setfacl - 说明如何设置文件访问控制列表
- man 星 - 解释有关 星形 实用程序及其多个选项的更多信息

第 6 章 获取特权

系统管理员，在某些情况下，用户需要使用管理访问权限来执行某些任务。以 root 用户身份访问系统具有潜在的危险性，可能会导致系统和数据出现广泛损坏。本章涵盖了使用 setuid 程序（如 su 和 sudo）获取管理权限的方法。这些程序允许特定用户执行通常仅对 root 用户可用的任务，同时保持更高级别的控制和系统安全性。

有关管理控制、潜在威胁和防止不当使用特权访问造成的数据丢失的更多信息，请参阅 Red Hat Enterprise Linux 7 安全指南。

6.1. 使用 SU 实用程序配置管理访问权限

当用户执行 su 命令时，系统会提示他们输入 root 密码，并且在验证后，系统会获得 root shell 提示符。

使用 su 命令登录后，该用户 是 root 用户，对系统具有绝对管理访问权限。请注意，这种访问仍会受到 SELinux 实施的限制（如果已启用）。此外，用户一旦成为 root 用户，便可以使用 su 命令更改为系统上的任何其他用户，而无需提示输入密码。

由于该程序如此强大，组织内的管理员可能希望限制谁有权访问 命令。

执行此操作的一种最简单的方法是将用户添加到名为 wheel 的特殊管理组中。要做到这一点，以 root 用户身份输入以下命令：

```
~]# usermod -a -G wheel username
```

在上一命令中，将 username 替换为您要添加到 wheel 组 的用户名。

您还可以使用 Users 设置工具来 修改组成员资格，如下所示：请注意，您需要管理员特权来执行此步骤。

1. 按 Super 键以进入活动概览，键入 Users，然后按 Enter 键。此时 会出现用户 设置工具。Super 键显示在各种 guis 中，具体取决于键盘和其他硬件，但通常作为 Windows 或 Command 键（通常在 空格栏 的左侧）。

2. 要启用更改，请单击 Unlock 按钮并输入有效的管理员密码。

3. 单击左侧列中的用户图标，以显示用户在右侧窗格中的属性。

4. 将帐户类型从 Standard 更改为 Administrator。这会将用户添加到 wheel 组。

有关 用户 工具的详情，请查看 第 4.2 节 “在图形环境中管理用户”。

将所需的用户添加到 wheel 组后，建议仅允许这些特定用户使用 su 命令。为此，请编辑 su、/etc/pam.d/su 的可插拔验证模块 (PAM)配置文件。在文本编辑器中打开此文件，并通过删除 # 字符取消注释下面这一行：

```
#auth      required    pam_wheel.so use_uid
```

这一更改意味着只有管理组 wheel 的成员可以使用 su 命令切换到其他用户。

6.2. 使用 SUDO 实用程序配置管理访问权限

sudo 命令提供了另一种授予用户管理访问权限的方法。当受信任的用户在管理命令之前加上 sudo 时，系统会提示他们输入 自己的 密码。然后，当它们经过身份验证并假定允许 命令时，将像 root 用户一样执行管理命令。

sudo 命令的基本格式如下：

```
sudo command
```

在上例中，命令 将被替换为通常保留给 root 用户的命令，如 mount。

sudo 命令具有高度的灵活性。例如，只有 /etc/sudoers 配置文件中列出的用户才能使用 sudo 命令，命令则在 用户的 shell 中执行，而不是 root shell。这意味着可以完全禁用 root shell，如 Red Hat Enterprise Linux 7 安全指南 中所示。

使用 sudo 命令的每个成功身份验证都会记录到文件 /var/log/messages，而与签发者的用户名一同发布的命令将记录到 /var/log/secure 文件。如果需要额外的日志记录，请使用 pam_tty_audit 模块为指定

用户启用 TTY 审核，方法是在 /etc/pam.d/system-auth 文件中添加以下行：

```
session required pam_tty_audit.so disable=pattern enable=pattern
```

其中，pattern 代表通过可选方式使用 glob 的以逗号分隔的用户列表。例如，以下配置将为 root 用户启用 TTY 审核，并为所有其他用户禁用它：

```
session required pam_tty_audit.so disable=* enable=root
```

重要

为 TTY 审计记录配置 pam_tty_audit PAM 模块，仅用于 TTY 输入。这意味着，当被审计用户登录时，pam_tty_audit 会记录用户在 /var/log/audit/audit.log 文件中进行的确切击键操作。如需更多信息，请参阅 pam_tty_audit(8)手册页。

sudo 命令的另一个优点是，管理员可以允许不同的用户根据自己的需要访问特定的命令。

想要编辑 sudo 配置文件 /etc/sudoers 的管理员应使用 visudo 命令。

要授予某人完整的管理权限，请键入 visudo 并添加类似用户权限规范部分中的下面这一行：

```
juan ALL=(ALL) ALL
```

此示例表示用户 juan 可以从任何主机使用 sudo 并执行任何命令。

以下示例演示了配置 sudo 时可能的粒度：

```
%users localhost=/usr/sbin/shutdown -h now
```

这个示例表示，users 系统组 的任何成员都可以发出 /sbin/shutdown -h 命令，只要它 从控制台发布。

sudoers 的 man page 包含此文件的选项的详细列表。

您还可以使用 /etc/sudoers 文件中的 NOPASSWD 选项配置不需要提供任何密码的 sudo 用户：

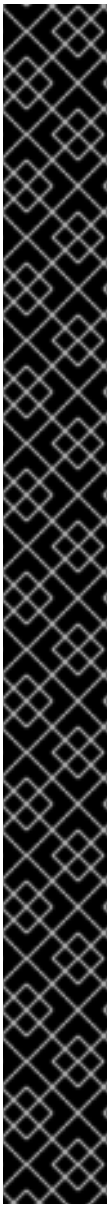
```
user_name ALL=(ALL) NOPASSWD: ALL
```

但是，即使对于此类用户，sudo 也运行可插拔验证模块(PAM)帐户管理模块，这可以检查 PAM 模块在身份验证阶段之外实施的限制。这样可确保 PAM 模块正常工作。例如，如果 pam_time 模块，基于时间的帐户限制不会失败。

警告

始终将 sudo 包含在所有基于 PAM 的访问控制规则中允许的服务列表中。否则，用户在尝试访问 sudo 时将收到"权限被拒绝"错误消息，但根据当前的访问控制规则禁止访问。

如需更多信息，请参阅在修补 Red Hat Enterprise Linux 7.6 后，Edo 给出了 permission denied 错误。



重要

使用 sudo 命令时，请记住几个潜在的风险：您可以按照上述所述，使用 visudo 编辑 /etc/sudoers 配置文件来避免它们。将 /etc/sudoers 文件保留为默认状态可授予 wheel 组中的每个用户 无限制 root 访问权限。

- 默认情况下，s udo 将密码存储为五分钟的超时时间。在此期间内，任何后续使用 命令都不会提示用户输入密码。如果用户在仍登录时无人值守和解锁其工作站，则攻击者可能会利用此功能。可以通过在 /etc/sudoers 文件中添加以下行来更改此行为：

```
Defaults  timestamp_timeout=value
```

其中 value 是所需的超时长度（以分钟为单位）。将 值设 为 0 会导致 sudo 每次都需要密码。

- 如果帐户泄露，攻击者可以使用 sudo 打开具有管理权限的新 shell：

```
sudo /bin/bash
```

以 root 用户身份以这种或类似的方式打开新 shell 可让攻击者在理论上无限制的时间内管理访问，绕过 /etc/sudoers 文件中指定的超时时间，并且无需攻击者再次为 sudo 输入密码，直到新打开的会话关闭为止。

6.3. 其它资源

虽然允许用户获得管理特权的程序存在潜在的安全风险，但安全性本身超出了本书的范围。因此，您应该参考以下列出的资源来了解有关安全和特权访问的更多信息。

安装的文档

- su(1)- su 的 man page 提供了关于此命令可用选项的信息。
- sudo(8)- sudo 的 man page 包含此命令的详细描述，列出了可用于自定义其行为的选项。
- PAM(8)- 描述在 Linux 中使用可插拔验证模块(PAM)的 man page。

在线文档

- *红帽企业 Linux 7 安全指南 - 红帽企业 Linux 7 安全指南 更详细地介绍了与 setuid 程序相关的潜在安全问题，以及用于缓解这些风险的技术。*

另请参阅

- *第 4 章 管理用户和组 记录了如何在图形用户界面和命令行中管理系统用户和组。*

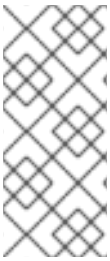
部分 II. 订阅和支持

要在 Red Hat Enterprise Linux 系统中接收软件更新，必须订阅 Red Hat Content Delivery Network (CDN)并启用适当的软件仓库。这部分论述了如何为 Red Hat Content Delivery Network 订阅系统。

红帽通过 客户门户 提供支持，您可以使用 红帽支持工具直接从命令行访问该支持。这部分论述了此命令行工具的使用。

第 7 章 注册系统管理并管理订阅

订阅服务提供处理红帽软件库存的机制，并允许您使用 yum 软件包管理器安装其他软件或将已安装的程序更新为更新的版本。在红帽企业 Linux 7 中，建议注册您的系统并附加订阅 是使用红帽订阅管理。



注意

也可以在初始设置过程中注册系统并附加订阅。有关初始设置的详情，请查看 Red Hat Enterprise Linux 7 的《安装指南》中 的初始设置章节。请注意，Initial Setup 应用程序仅在安装时在 X Window 系统安装的系统上可用。

7.1. 注册系统和附加订阅

完成以下步骤以注册您的系统并使用红帽订阅管理附加一个或多个订阅。请注意，所有 subscription-manager 命令都应以 root 用户身份运行。

1. 运行以下命令注册您的系统。此时会提示您输入您的用户名和密码。请注意：用户名和密码与您的红帽客户门户网站登录证书相同。

```
subscription-manager register
```

2. 确定您需要的订阅池 ID。要做到这一点，在 shell 提示符后输入以下内容以显示系统可用的所有订阅列表：

```
subscription-manager list --available
```

对于每个可用的订阅，这个命令会显示其名称、唯一标识符、到期日期以及与订阅相关的其他详情。要列出所有架构的订阅，请添加 --all 选项。池 ID 列在以 池 ID 开头的行上。

3. 输入以下命令为您的系统附加适当的订阅：

```
subscription-manager attach --pool=pool_id
```

将 pool_id 替换为您在上一步中确定的池 ID。

要验证您的系统当前已附加的订阅列表，请运行：

```
subscription-manager list --consumed
```

有关如何使用红帽订阅管理注册您的系统并将其与订阅关联的详情，请参考指定的解决方案文章。有关订阅的综合信息，请参阅红帽订阅管理指南集合。

7.2. 管理软件存储库

当系统订阅 Red Hat Content Delivery Network 时，会在 /etc/yum.repos.d/ 目录中创建一个仓库文件。要验证，请使用 yum 列出所有启用的软件仓库：

```
yum repolist
```

红帽订阅管理还允许您手动启用或禁用红帽提供的软件存储库。要列出所有可用的软件仓库，请使用以下命令：

```
subscription-manager repos --list
```

软件仓库名称取决于您使用的 Red Hat Enterprise Linux 的具体版本，其格式如下：

```
rhel-version-variant-rpms
rhel-version-variant-debug-rpms
rhel-version-variant-source-rpms
```

其中 version 是 Red Hat Enterprise Linux 系统版本（6 或 7），变体 是 Red Hat Enterprise Linux 系统变体（服务器 或 工作站），例如：

```
rhel-7-server-rpms
rhel-7-server-debug-rpms
rhel-7-server-source-rpms
```

要启用存储库，请输入以下命令：

```
subscription-manager repos --enable repository
```

使用要启用的存储库的名称替换 repository。

同样，要禁用存储库，请使用以下命令：

```
subscription-manager repos --disable repository
```

第 9.5 节 “配置 Yum 和 Yum 存储库” 提供关于使用 yum 管理软件存储库的详细信息。

如果要自动更新存储库，您可以使用 yum-cron 服务。如需更多信息，请参阅 第 9.7 节 “使用 Yum-cron 自动刷新软件包数据库和下载更新”。

7.3. 删除订阅

要删除特定的订阅，请完成以下步骤：

1. 通过列出已附加订阅的信息，确定您要删除的订阅的序列号：

```
subscription-manager list --consumed
```

序列号是列为 串行 号。例如，以下示例中的 744993814251016831：

```
SKU:       ES0113909
Contract:     01234567
Account:      1234567
Serial:      744993814251016831
Pool ID:      8a85f9894bba16dc014bccdd905a5e23
Active:      False
Quantity Used:   1
Service Level:   SELF-SUPPORT
Service Type:   L1-L3
Status Details:
Subscription Type: Standard
Starts:      02/27/2015
Ends:      02/27/2016
System Type:    Virtual
```

2. 按如下所示输入命令以删除所选订阅：

```
subscription-manager remove --serial=serial_number
```

使用您在上一步中确定的序列号替换 serial_number。

要删除附加到该系统的所有订阅，请运行以下命令：

```
subscription-manager remove --all
```

7.4. 其它资源

有关如何使用红帽订阅管理注册您的系统并将其与订阅关联的更多信息，请参阅以下列出的资源。

安装的文档

- subscription-manager(8)- Red Hat Subscription Management 的 man page 提供了支持的选项和命令的完整列表。

相关书

- 红帽订阅管理指南集合 - 这些指南包含如何使用红帽订阅管理的详细信息。
- 《安装指南》 - 有关如何在初始设置过程中注册的详细信息，请参阅初始设置章节。

另请参阅

- 第 6 章 获取特权 文档如何使用 su 和 sudo 命令获得管理权限。
- 第 9 章 yum 提供有关使用 yum 软件包管理器安装和更新软件的信息。

第 8 章 使用红帽支持工具访问支持

红帽支持工具 redhat-support-tool 软件包中可作为交互式 shell 和单一执行计划运行。它可以通过 SSH 或从任何终端运行。例如，它可从命令行搜索红帽知识库，直接在命令行中复制解决方案，打开和更新支持案例，并向红帽发送文件进行分析。

8.1. 安装红帽支持工具

默认情况下，红帽支持工具 在 Red Hat Enterprise Linux 中安装。如果需要，请以 root 用户身份 输入以下命令：

```
~]# yum install redhat-support-tool
```

8.2. 使用命令行注册红帽支持工具

要使用命令行将红帽支持工具注册到客户门户中，请运行以下命令：

```
~]# redhat-support-tool config user username
```

其中 username 是红帽客户门户网站帐户的用户名。

```
~]# redhat-support-tool config password
Please enter the password for username:
```

8.3. 在互动 SHELL 模式中使用红帽支持工具

要在互动模式下启动该工具，请输入以下命令：

```
~]$ redhat-support-tool
Welcome to the Red Hat Support Tool.
Command (? for help):
```

工具可以作为非特权用户运行，因此可以减少一组命令，或者以 root 用户身份运行。

可以通过输入 ? 字符列出这些命令。程序或菜单选择可以通过输入 q 或 e 字符退出。当您首次搜索知识库或支持案例时，系统会提示您输入您的红帽客户门户网站用户名和密码。另外，使用交互模式为您的红帽客户门户网站帐户设置用户名和密码，并选择性地将其保存到配置文件中。

8.4. 配置红帽支持工具

当处于互动模式中时，可以通过输入命令 config --help 来列出配置选项：

```
~]# redhat-support-tool
Welcome to the Red Hat Support Tool.
Command (? for help): config --help

Usage: config [options] config.option <new option value>

Use the 'config' command to set or get configuration file values.
Options:
 -h, --help  show this help message and exit
 -g, --global Save configuration option in /etc/redhat-support-tool.conf.
 -u, --unset  Unset configuration option.

The configuration file options which can be set are:
 user   : The Red Hat Customer Portal user.
 password : The Red Hat Customer Portal password.
 debug   : CRITICAL, ERROR, WARNING, INFO, or DEBUG
 url    : The support services URL. Default=https://api.access.redhat.com
 proxy_url : A proxy server URL.
 proxy_user: A proxy server user.
 proxy_password: A password for the proxy server user.
 ssl_ca  : Path to certificate authorities to trust during communication.
 kern_debug_dir: Path to the directory where kernel debug symbols should be downloaded
and cached. Default=/var/lib/redhat-support-tool/debugkernels

Examples:
- config user
- config user my-rhn-username
- config --unset user
```

使用互动模式注册红帽支持工具

要使用互动模式将红帽支持工具注册到客户门户中，请按如下操作：

1. 输入以下命令启动工具：

   ```
   ~]# redhat-support-tool
   ```

2. 输入您的红帽客户门户网站用户名：

   ```
   Command (? for help): config user username
   ```

要将您的用户名保存到全局配置文件中，请添加 -g 选项。

3. 输入您的红帽客户门户网站密码：

```
Command (? for help): config password
Please enter the password for username:
```

8.4.1. 将设置保存到配置文件中

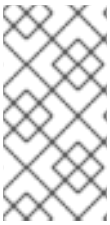
红帽支持工具（除非另有指示）使用 ~/.redhat-support-tool/redhat-support-tool/redhat-support-tool.conf 配置文件在本地存储值和选项。如果需要，建议将密码保存到此文件，因为它仅可由特定用户读取。当工具启动时，它将从全局配置文件 /etc/redhat-support-tool.conf 和本地配置文件中读取值。本地存储的值和选项优先于全局存储设置。

警告

建议 不要将 密码保存在全局 /etc/redhat-support-tool.conf 配置文件中，因为密码仅经过 base64 编码且易于解码。此外，文件完全可读。

要在全局配置文件中保存值或选项，请按如下所示添加 -g, --global 选项：

```
Command (? for help): config setting -g value
```



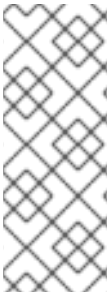
注意

为了能够在全局范围内保存设置，使用 -g、--global 选项，红帽支持工具 必须以 root 用户身份运行，因为普通用户没有写入 /etc/redhat-support-tool.conf 所需的权限。

要从本地配置文件中删除值或选项，请添加 -u, --unset 选项，如下所示：

```
Command (? for help): config setting -u value
```

这将清除（未设置）工具中的 参数，并回退到全局配置文件中的等效设置（如果可用）。



注意

以非特权用户身份运行时，无法使用 -u, --unset 选项删除存储在全局配置文件中的值，但是它们可以通过与 -u, --unset 选项同时使用 -u, --unset 选项从工具的当前运行实例中清除、 取消设置。如果以 root 身份运行，则可以使用 -g、--global 和 -u, -- unset 选项同时从全局配置文件中删除值和选项。

8.5. 使用互动模式打开和更新支持案例

使用互动模式打开新的支持案例

要使用互动模式打开一个新的支持问题单，请按如下所示：

1. 输入以下命令启动工具：

```
~]# redhat-support-tool
```

2. 输入 opencase 命令：

```
Command (? for help): opencase
```

3. 按照屏幕上的 提示选择产品和版本。

4. 输入案例摘要。

5. Ctrl+D。

6. 选择案例的严重性。

7. 在打开支持问题单前， 可以选择查看此问题是否有解决方案。

8. 确认您仍然希望打开支持案例。

```
Support case 0123456789 has successfully been opened
```

9. （可选）选择附加 SOS 报告。

10. （可选）选择附加文件。

使用交互模式查看和更新现有支持案例

要使用互动模式查看和更新现有支持问题单，请按如下操作：

1. 输入以下命令启动工具：

```
~]# redhat-support-tool
```

2. 输入 getcase 命令：

```
Command (? for help): getcase case-number
```

其中 case-number 是您要查看和更新的情况的编号。

3. 按照屏幕提示中的 以查看案例、修改或添加注释、获取或添加附件。

使用互动模式修改现有支持案例

要使用互动模式修改现有支持问题单的属性，请按如下所示：

1. 输入以下命令启动工具：

```
~]# redhat-support-tool
```

2. 输入 修改问题单 命令：

```
Command (? for help): modifycase case-number
```

其中 case-number 是您要查看和更新的情况的编号。

3. 修改选择列表会出现：

```
Type the number of the attribute to modify or 'e' to return to the previous menu.
 1 Modify Type
 2 Modify Severity
 3 Modify Status
 4 Modify Alternative-ID
 5 Modify Product
 6 Modify Version
End of options.
```

按照屏幕提示中的，修改一个或多个选项。

4. 例如，要修改状态，请输入 3 ：

```
Selection: 3
 1  Waiting on Customer
 2  Waiting on Red Hat
 3  Closed
Please select a status (or 'q' to exit):
```

8.6. 在命令行中查看支持案例

在命令行中查看案例的内容可快速轻松地从命令行应用解决方案。

要在命令行中查看现有支持问题单，请输入以下命令：

```
~]# redhat-support-tool getcase case-number
```

其中 case-number 是您要下载的情况的编号。

8.7. 其它资源

Red Hat 知识库文章 红帽支持工具 包含其他信息、示例和视频教程。

部分 III. 安装和管理软件

Red Hat Enterprise Linux 系统上的所有软件都被分成 RPM 包，这些软件包可以被安装、升级或删除。这部分论述了如何使用 Yum 在 Red Hat Enterprise Linux 中管理软件包。

第 9 章 YUM

yum 是红帽软件包管理器，可以查询有关可用软件包的信息，从存储库获取软件包，安装和卸载它们，并将整个系统更新至最新可用版本。yum 在更新、安装或删除软件包时执行自动依赖项解析，因此能够自动确定、获取和安装所有可用的依赖软件包。

yum 可使用新的、额外的存储库或软件包来源进行配置，也可提供许多插件来增强和扩展其功能。yum 可以执行 RPM 可以执行的许多相同任务；此外，许多命令行选项类似。yum 可以在一台机器或一组机器上轻松、简单的软件包管理。

以下部分假设您的系统在安装过程中注册了红帽订阅管理，如 Red Hat Enterprise Linux 7 安装指南 所述。如果您的系统没有订阅，请参阅 第 7 章 注册系统管理并管理订阅。

重要

yum 通过启用 GPG(Gnu Privacy Guard)对 GPG 签名包（软件包源）或单个存储库打开的签名验证来提供安全软件包管理。启用签名验证后，yum 将拒绝安装任何未使用该存储库的正确密钥签名的 GPG 软件包。这意味着，您可以信任在您的系统上下载和安装的 RPM 软件包来自红帽等可信源，并在传输过程中不会被修改。有关使用 yum 启用签名检查的详情，请查看 第 9.5 节 “配置 Yum 和 Yum 存储库”。

yum 还可让您轻松设置自己的 RPM 软件包存储库，以便在其他计算机上下载和安装。yum 尽可能使用 并行下载 多个软件包和元数据来加快下载。

学习 yum 是一项合理的投资，因为它通常是执行系统管理任务的最快方法，而且除了 PackageKit 图形包管理工具提供的功能之外。

注意

您必须具有超级用户权限，才能使用 yum 在您的系统上安装、更新或删除软件包。本章中的所有示例都假定您已使用 su 或 sudo 命令获取了超级用户权限。

9.1. 检查和更新软件包

yum 可让您检查您的系统是否有等待应用的更新。您可以列出需要更新的软件包，并整体更新它们，也可以更新选定的单个软件包。

9.1.1. 检查更新

要查看系统中安装的软件包有可用的更新，请使用以下命令：

```
yum check-update
```

例 9.1. yum check-update 命令的输出示例

yum check-update 的输出结果如下：

```
~]# yum check-update
Loaded plugins: product-id, search-disabled-repos, subscription-manager
dracut.x86_64            033-360.el7_2   rhel-7-server-rpms
dracut-config-rescue.x86_64      033-360.el7_2   rhel-7-server-rpms
kernel.x86_64            3.10.0-327.el7   rhel-7-server-rpms
rpm.x86_64             4.11.3-17.el7   rhel-7-server-rpms
rpm-libs.x86_64          4.11.3-17.el7   rhel-7-server-rpms
rpm-python.x86_64          4.11.3-17.el7   rhel-7-server-rpms
yum.noarch             3.4.3-132.el7   rhel-7-server-rpms
```

以上输出中的软件包列为具有可用的更新。列表中的第一个软件包是 dracut。示例输出中的每一行都由多个行组成，如果是 dracut ：

- dracut - 软件包的名称，
- x86_64 - 构建软件包的 CPU 架构，
- 033 - 要安装的更新软件包的版本，
- 360.el7 - 更新软件包的发布，
- _2 - 构建版本，作为 z-stream 更新的一部分添加.
- rhel-7-server-rpms - 更新软件包所在的存储库。

输出中还显示，我们可以更新内核（内核 包）、yum 和 RPM（yum 和 rpm 包），以及它们的依赖项（如 rpm -libs 和 rpm- python 软件包），所有这些都使用 yum 命令。

9.1.2. 更新软件包

您可以选择更新单个软件包、多个软件包或所有软件包。如果您更新的软件包或软件包的任何依赖项都有可用更新，则它们也会更新。

更新单个软件包

要更新单个软件包，以 root 用户身份运行以下命令：

```
yum update package_name
```

例 9.2. 更新 rpm 软件包

要更新 rpm 软件包，请输入：

```
~]# yum update rpm
Loaded plugins: langpacks, product-id, subscription-manager
Updating Red Hat repositories.
INFO:rhsm-app.repolib:repos updated: 0
Setting up Update Process
Resolving Dependencies
--> Running transaction check
---> Package rpm.x86_64 0:4.11.1-3.el7 will be updated
--> Processing Dependency: rpm = 4.11.1-3.el7 for package: rpm-libs-4.11.1-3.el7.x86_64
--> Processing Dependency: rpm = 4.11.1-3.el7 for package: rpm-python-4.11.1-3.el7.x86_64
--> Processing Dependency: rpm = 4.11.1-3.el7 for package: rpm-build-4.11.1-3.el7.x86_64
---> Package rpm.x86_64 0:4.11.2-2.el7 will be an update
--> Running transaction check
...
--> Finished Dependency Resolution

Dependencies Resolved
================================================================================

 Package        Arch    Version     Repository    Size
================================================================================

Updating:
 rpm            x86_64   4.11.2-2.el7  rhel      1.1 M
Updating for dependencies:
 rpm-build      x86_64   4.11.2-2.el7  rhel      139 k
 rpm-build-libs     x86_64   4.11.2-2.el7  rhel      98 k
 rpm-libs       x86_64   4.11.2-2.el7  rhel      261 k
 rpm-python     x86_64   4.11.2-2.el7  rhel      74 k

Transaction Summary
================================================================================
```

```
Upgrade 1 Package (+4 Dependent packages)

Total size: 1.7 M
Is this ok [y/d/N]:
```

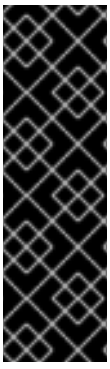
这个输出包含几个值得关注的项目：

1. 加载的插件：Langpacks、products-id、subscription-manager - Yum 始终通知您已安装和启用哪些 yum 插件。有关 yum 插件的常规信息，或 第 9.6.3 节 “使用 Yum 插件” 的常规信息，请参阅 第 9.6 节 “yum 插件” 以了解特定插件的描述。

2. rpm.x86_64 - 您可以下载并安装新的 rpm 软件包及其依赖项。对每个软件包执行事务检查。

3. yum 显示更新信息，然后提示您确认更新；yum 默认以交互方式运行。如果您知道 yum 命令计划执行 哪些事务，您可以使用 -y 选项自动 回答 yum 询问的任何问题（在这种情况下，它以非交互方式运行）。但是，您应该始终检查 yum 计划对系统做出哪些更改，以便您可以轻松地对可能出现的任何问题进行故障排除。您也可以选择下载软件包而不进行安装。为此，请在下载提示符处选择 d 选项。这将启动所选软件包的背景下载。

   如果事务失败，您可以使用 yum history 命令查看 yum 事务 历史记录，如 第 9.4 节 “使用事务历史记录” 所述。



重要

无论您使用 yum update 或 yum install 命令，yum 始终都会 安装 新内核。

另一方面，使用 RPM 时，务必使用 rpm -i 内核命令来 安装新内核，而不是 rpm -u 内核 （替换 当前内核）。

同样，也可以更新软件包组。以 root 用户身份键入：

```
yum group update group_name
```

在这里，使用您要更新的软件包组的名称替换 group_name。有关软件包组的详情请参考 第 9.3 节 “使用软件包组”。

yum 还提供与启用的过时配置选项相同的升级命令（请参阅第 9.5.1 节 “设置 [main] 选项”）。默认情况下，/etc/yum.conf 中打开了过时的功能，这使得这两个命令具有同等性。

更新所有软件包及其依赖项

要更新所有软件包及其依赖项，请使用不带任何参数的 yum update 命令：

```
yum update
```

更新与安全相关的软件包

如果软件包有可用的安全更新，则只能将这些软件包更新至其最新版本。以 root 用户身份键入：

```
yum update --security
```

您还可以仅将软件包更新为包含最新安全更新的版本。以 root 用户身份键入：

```
yum update-minimal --security
```

例如，假设：

- kernel-3.10.0-1 软件包安装在您的系统中；
- kernel-3.10.0-2 软件包已作为安全更新发布；
- kernel-3.10.0-3 软件包已作为程序错误修复更新发布。

然后 yum update-minimal --security 将软件包更新至 kernel-3.10.0-2，yum update --security 将软件包更新为 kernel-3.10.0-3。

自动更新软件包

要刷新软件包数据库并自动下载更新，您可以使用 yum-cron 服务。如需更多信息，请参阅第 9.7 节 “使用 Yum-cron 自动刷新软件包数据库和下载更新”。

9.1.3. 使用 ISO 和 Yum 升级系统离线

对于与互联网或红帽网络断开连接的系统，将 yum update 命令与 Red Hat Enterprise Linux 安装 ISO 映像配合使用是将系统升级到最新次要版本的一种简单快速方式。以下步骤演示了升级过程：

1. 创建目标目录以挂载您的 ISO 映像。挂载时不会自动创建该目录，因此请在继续下一步之前创建该目录。以 root 用户身份键入：

```
mkdir mount_dir
```

使用挂载目录的路径替换 mount_dir。通常，用户将其创建为 /media 目录中的子目录。

2. 将 Red Hat Enterprise Linux 7 安装 ISO 镜像挂载到之前创建的目标目录中。以 root 用户身份键入：

```
mount -o loop iso_name mount_dir
```

将 iso_name 替换为您的 ISO 镜像的路径，mount_dir 替换为目标目录的路径。此处需要 -o loop 选项，才能将文件挂载为块设备。

3. 将 media.repo 文件从 mount 目录复制到 /etc/yum.repos.d/ 目录。请注意，此目录中的配置文件必须具有 .repo 扩展名才能正常工作。

```
cp mount_dir/media.repo /etc/yum.repos.d/new.repo
```

这会为 yum 存储库创建一个配置文件。使用文件名替换 new.repo，如 rhel7.repo。

4. 编辑新配置文件，使其指向 Red Hat Enterprise Linux 安装 ISO。在 /etc/yum.repos.d/new.repo 文件中添加以下行：

```
baseurl=file:///mount_dir
```

使用挂载点的路径替换 mount_dir。

5. 更新所有 yum 存储库，包括 /etc/yum.repos.d/new.repo。以 root 用户身份键入：

```
yum update
```

这会将您的系统升级到挂载的 ISO 镜像提供的版本。

6. 升级成功后，您可以卸载 ISO 镜像。以 root 用户身份键入：

```
umount mount_dir
```

其中 mount_dir 是挂载目录的路径。此外，您可以删除在第一步中创建的挂载目录。以 root 用户身份键入：

```
rmdir mount_dir
```

7. 如果您不会将之前创建的配置文件用于另一个安装或更新，您可以将其删除。以 root 用户身份键入：

```
rm /etc/yum.repos.d/new.repo
```

例 9.3. 从 Red Hat Enterprise Linux 7.0 升级到 7.1

如果需要，需要使用带有较新版本的系统（如 rhel-server-7.1-x86_64-dvd.iso）的 ISO 映像来升级不能访问互联网的系统，请创建用于挂载的目标目录，如 /media/rhel7/。以 root 用户身份，使用您的 ISO 镜像更改到 目录并键入：

```
~]# mount -o loop
  rhel-server-7.1-x86_64-dvd.iso /media/rhel7/
```

然后通过从挂载目录中复制 media.repo 文件来为您的镜像设置 yum 存储库：

```
~]# cp /media/rhel7/media.repo /etc/yum.repos.d/rhel7.repo
```

要将 yum 识别挂载点为仓库，请在上一步中复制的 /etc/yum.repos.d/rhel7.repo 中添加以下行：

```
baseurl=file:///media/rhel7/
```

现在，更新 yum 存储库会将您的系统升级到 rhel-server-7.1-x86_64-dvd.iso 提供的版本。以 root 用户身份执行：

```
~]# yum update
```

成功升级您的系统时，您可以卸载镜像，删除目标目录和配置文件：

```
~]# umount /media/rhel7/
```

```
~]# rmdir /media/rhel7/
```

```
~]# rm
    /etc/yum.repos.d/rhel7.repo
```

9.2. 使用软件包

yum 允许您对软件包执行一整套操作，包括搜索软件包、查看有关软件包的信息、安装和删除软件包。

9.2.1. 搜索软件包

您可以使用以下命令搜索所有 RPM 软件包名称、描述和摘要：

```
yum search term&hellip;
```

使用您要搜索的软件包名称替换 term。

例 9.4. 搜索与特定字符串匹配的软件包

要列出与 "vim"、"gvim" 或 "emacs" 匹配的所有软件包，请输入：

```
~]$ yum search vim gvim emacs
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
============================== N/S matched: vim ==============================
vim-X11.x86_64 : The VIM version of the vi editor for the X Window System
vim-common.x86_64 : The common files needed by any version of the VIM editor
[output truncated]

============================== N/S matched: emacs
==============================
emacs.x86_64 : GNU Emacs text editor
emacs-auctex.noarch : Enhanced TeX modes for Emacs
```

```
[output truncated]

 Name and summary matches mostly, use "search all" for everything.
Warning: No matches found for: gvim
```

yum search 命令可用于搜索您不知道名称但了解相关术语的软件包。请注意，默认情况下，yum search 会在 包名称和摘要中返回匹配项，这样可加快搜索速度。使用 yum search all 命令进行更详细但较慢的搜索。

过滤结果

所有 yum 的 list 命令都允许您通过附加一个或多个 glob 表达式作为参数来过滤结果。glob 表达式是包含一个或多个通配符字符 *（扩展以匹配任何字符子集）和 ？（扩展以匹配任何单个字符）的普通字符字符串。

当将 glob 表达式作为参数传递给 yum 命令时，请小心转义 glob 表达式，否则 Bash shell 会将这些表达式解释为 路径名扩展，并可能 将当前目录中与全局表达式匹配的所有文件传递给 yum。要确定将 glob 表达式传递给 yum，请使用以下方法之一：

- 在通配符前面使用反斜杠字符转义
- 双引号或单引号整个 glob 表达式。

下一节中的示例演示了这两种方法的用法。

9.2.2. 列出软件包

要列出所有安装和可用软件包的信息，在 shell 提示符下键入以下内容：

```
yum list all
```

要列出与插入 glob 表达式匹配的已安装和可用软件包，请使用以下命令：

```
yum list glob_expression&hellip;
```

例 9.5. 列出与 ABRT 相关的软件包

带有各种 ABRT 附加组件和插件的软件包以"abrt-addon-"或"abrt-plugin-"开头。要列出这些软件包，请在 shell 提示符后键入以下命令：请注意如何使用反斜杠字符转义通配符字符：

```
~]$ yum list abrt-addon\* abrt-plugin\*
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
Installed Packages
abrt-addon-ccpp.x86_64          2.1.11-35.el7       @rhel-7-server-rpms
abrt-addon-kerneloops.x86_64       2.1.11-35.el7       @rhel-7-server-rpms
abrt-addon-pstoreoops.x86_64       2.1.11-35.el7       @rhel-7-server-rpms
abrt-addon-python.x86_64          2.1.11-35.el7       @rhel-7-server-rpms
abrt-addon-vmcore.x86_64          2.1.11-35.el7       @rhel-7-server-rpms
abrt-addon-xorg.x86_64          2.1.11-35.el7       @rhel-7-server-rpms
```

要列出系统上安装的所有软件包，请使用 已安装的 关键字。输出中最右侧的列列出了从中检索软件包的存储库。

```
yum list installed glob_expression&hellip;
```

例 9.6. 列出 krb 软件包的所有已安装版本

以下示例演示了如何列出所有以"krb"开头的软件包，后跟一个字符和连字符。这在您要列出特定组件的所有版本时很有用，因为它们按数字区分。整个 glob 表达式都用引号括起，以确保正确处理。

```
~]$ yum list installed "krb?-*"
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
Installed Packages
krb5-libs.x86_64          1.13.2-10.el7          @rhel-7-server-rpms
```

要列出所有启用的软件仓库中可用于安装的软件包，请使用以下命令：

```
yum list available glob_expression&hellip;
```

例 9.7. 列出可用的 gstreamer 插件

例如，要列出包含"gstreamer"和"plugin"的名称的所有可用软件包，请运行以下命令：

```
~]$ yum list available gstreamer*plugin\*
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
Available Packages
gstreamer-plugins-bad-free.i686       0.10.23-20.el7      rhel-7-server-rpms
gstreamer-plugins-base.i686          0.10.36-10.el7      rhel-7-server-rpms
gstreamer-plugins-good.i686          0.10.31-11.el7      rhel-7-server-rpms
```

```
gstreamer1-plugins-bad-free.i686      1.4.5-3.el7      rhel-7-server-rpms
gstreamer1-plugins-base.i686          1.4.5-2.el7      rhel-7-server-rpms
gstreamer1-plugins-base-devel.i686    1.4.5-2.el7      rhel-7-server-rpms
gstreamer1-plugins-base-devel.x86_64  1.4.5-2.el7      rhel-7-server-rpms
gstreamer1-plugins-good.i686          1.4.5-2.el7      rhel-7-server-rpms
```

列出存储库

要列出系统中每个启用的存储库的存储库 ID、名称和软件包数量，请使用以下命令：

```
yum repolist
```

要列出这些存储库的更多信息，请添加 -v 选项。启用此选项后，将显示每个列出的存储库的信息，包括文件名、总体大小、最后一次更新的日期和基本 URL。另外，您可以使用生成相同输出的 repoinfo 命令。

```
yum repolist -v
```

```
yum repoinfo
```

若要列出已启用和禁用的存储库，可使用以下命令：在输出列表中添加一个状态列，以显示启用了哪些存储库。

```
yum repolist all
```

通过将 disabled 作为第一个参数传递，您可以将命令输出减少至禁用的存储库。如需进一步规格，您可以将存储库的 ID 或名称或相关的 glob_expressions 作为参数传递。请注意，如果存储库 ID 或名称与插入的参数完全匹配，则即使未传递已启用或禁用 的过滤器，也会列出此存储库。

9.2.3. 显示软件包信息

要显示一个或多个软件包的信息，请使用以下命令（glob 表达式在这里也有效）：

```
yum info package_name&hellip;
```

使用软件包名称替换 package_name。

例 9.8. 显示 abrt 软件包中的信息

要显示 abrt 软件包的信息，请输入：

```
~]$ yum info abrt
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
Installed Packages
Name    : abrt
Arch    : x86_64
Version   : 2.1.11
Release   : 35.el7
Size    : 2.3 M
Repo    : installed
From repo  : rhel-7-server-rpms
Summary   : Automatic bug detection and reporting tool
URL     : https://fedorahosted.org/abrt/
License   : GPLv2+
Description : abrt is a tool to help users to detect defects in applications and
      : to create a bug report with all information needed by maintainer to fix
      : it. It uses plugin system to extend its functionality.
```

yum info package_name 命令类似于 rpm -q --info package_name 命令，但提供了从 安装 RPM 软件包的 yum 存储库的名称（查找输出中的 From repo: 行）。

使用 yumdb

您还可以使用以下命令查询 yum 数据库以获取有关软件包的替代和有用信息：

```
yumdb info package_name
```

此命令提供关于软件包的其他信息，包括软件包的校验和（以及用于生成软件包的算法，如 SHA-256）、命令行上为安装软件包所提供的命令（如果有），以及将软件包安装到系统上的原因（用户 指明其由用户安装，而 dep 表示它是作为依赖项引入的）。

例 9.9. 查询 yumdb 以了解有关 yum 软件包的信息

要显示 yum 软件包的附加信息，请输入：

```
~]$ yumdb info yum
Loaded plugins: langpacks, product-id
yum-3.4.3-132.el7.noarch
  changed_by = 1000
  checksum_data =
a9d0510e2ff0d04d04476c693c0313a11379053928efd29561f9a837b3d9eb02
  checksum_type = sha256
  command_line = upgrade
```

```
    from_repo = rhel-7-server-rpms
    from_repo_revision = 1449144806
    from_repo_timestamp = 1449144805
    installed_by = 4294967295
    origin_url =
https://cdn.redhat.com/content/dist/rhel/server/7/7Server/x86_64/os/Packages/yum-3.4.3-
132.el7.noarch.rpm
    reason = user
    releasever = 7Server
    var_uuid = 147a7d49-b60a-429f-8d8f-3edb6ce6f4a1
```

有关 yumdb 命令的更多信息，请参阅 yumdb(8)手册页。

9.2.4. 安装软件包

要安装单个软件包及其所有未安装的依赖项，以 root 用户身份输入以下命令：

```
yum install package_name
```

您还可以通过将多个软件包作为参数附加来同时安装多个软件包。要做到这一点，以 root 用户身份输入：

```
yum install package_name package_name&hellip;
```

如果要在 multilib 系统（如 AMD64 或 Intel 64 机器）上安装软件包，您可以通过在软件包名称中添加.arch 来指定软件包的构架（只要启用的软件仓库中可用）：

```
yum install package_name.arch
```

例 9.10. 在 multilib 系统上安装软件包

要为 i686 架构安装 sqlite 软件包，请输入：

```
~]# yum install sqlite.i686
```

您可以使用 glob 表达式快速安装多个名称相似的软件包。以 root 用户身份执行：

```
yum install glob_expression&hellip;
```

例 9.11. 安装所有udacious插件

如果要安装多个名称相似的软件包时，全局表达式很有用。要安装所有 audacious 插件，请使用以下格式的命令：

```
~]# yum install audacious-plugins-\*
```

除了软件包名称和 glob 表达式外，您还可以为 yum install 提供文件名。如果您知道要安装的二进制文件的名称，但不知道其软件包名称，您可以为 yum install 提供路径名称。以 root 用户身份键入：

```
yum install /usr/sbin/named
```

yum 随后搜索其包列表，找到提供 /usr/sbin/named 的软件包（如果有），并提示您是否安装它。

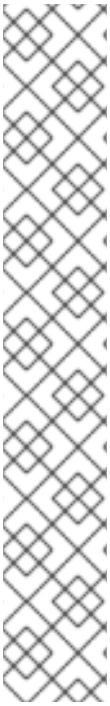
如上例中所示，yum install 命令不需要严格定义的参数。它可以处理各种软件包名称和 glob 表达式格式，从而方便用户安装。另一方面，需要花些时间才能正确 解析 输入，特别是您指定了大量软件包时。要优化软件包搜索，您可以使用以下命令来显式定义如何解析参数：

```
yum install-n name
```

```
yum install-na name.architecture
```

```
yum install-nevra name-epoch:version-release.architecture
```

使用 install-n 时，yum 会将 name 解释为软件包的确切名称。install-na 命令告知 yum，后续参数包含用点字符划分的软件包名称和架构。使用 install-nevra 时，yum 期望格式为 name-epoch:version-release.architecture 的参数。同样，搜索要删除的软件包时，您可以使用 yum remove-n、yum remove-na 和 yum remove-nevra。



注意

如果您知道要安装包含 命名 二进制的软件包，但您不知道在哪个 bin/ or sbin/ 目录中安装了该文件，请使用带有 glob 表达式的 yum 提供 命令：

```
~]# yum provides "*bin/named"
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-
      : manager
32:bind-9.9.4-14.el7.x86_64 : The Berkeley Internet Name Domain (BIND) DNS
            : (Domain Name System) server
Repo    : rhel-7-server-rpms
Matched from:
Filename  : /usr/sbin/named
```

yum 提供"*/file_name" 是查找包含 file_name 的软件包的有用方法。

例 9.12. 安装过程

以下示例提供了使用 yum 的安装概述。要下载并安装最新版本的 httpd 软件包，以 root 用户身份执行：

```
~]# yum install httpd
Loaded plugins: langpacks, product-id, subscription-manager
Resolving Dependencies
--> Running transaction check
---> Package httpd.x86_64 0:2.4.6-12.el7 will be updated
---> Package httpd.x86_64 0:2.4.6-13.el7 will be an update
--> Processing Dependency: 2.4.6-13.el7 for package: httpd-2.4.6-13.el7.x86_64
--> Running transaction check
---> Package httpd-tools.x86_64 0:2.4.6-12.el7 will be updated
---> Package httpd-tools.x86_64 0:2.4.6-13.el7 will be an update
--> Finished Dependency Resolution

Dependencies Resolved
```

执行上述命令后，yum 会加载必要的插件并运行事务检查。在本例中，httpd 已经安装。由于安装的软件包比最新可用版本旧，因此将会进行更新。这同样适用于 httpd 依赖的 httpd -tools 软件包。然后会显示一个事务概述：

```
================================================================================
 Package    Arch   Version        Repository       Size
================================================================================
Updating:
 httpd      x86_64  2.4.6-13.el7      rhel-x86_64-server-7  1.2 M
```

```
Updating for dependencies:
 httpd-tools  x86_64  2.4.6-13.el7      rhel-x86_64-server-7   77 k

Transaction Summary
================================================================================
=
Upgrade 1 Package (+1 Dependent package)

Total size: 1.2 M
Is this ok [y/d/N]:
```

在这一步中，yum 会提示您确认安装。除了 y (yes)和 N （否）选项外，您可以选择 d （仅下载）下载软件包，但不能直接安装它们。如果您选择 y，安装会继续包含以下信息，直到成功完成为止。

```
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
 Updating  : httpd-tools-2.4.6-13.el7.x86_64            1/4
 Updating  : httpd-2.4.6-13.el7.x86_64                  2/4
 Cleanup   : httpd-2.4.6-12.el7.x86_64                  3/4
 Cleanup   : httpd-tools-2.4.6-12.el7.x86_64            4/4
 Verifying : httpd-2.4.6-13.el7.x86_64                  1/4
 Verifying : httpd-tools-2.4.6-13.el7.x86_64            2/4
 Verifying : httpd-tools-2.4.6-12.el7.x86_64            3/4
 Verifying : httpd-2.4.6-12.el7.x86_64                  4/4

Updated:
 httpd.x86_64 0:2.4.6-13.el7

Dependency Updated:
 httpd-tools.x86_64 0:2.4.6-13.el7

Complete!
```

要从系统中的本地目录安装之前下载的软件包，请使用以下命令：

```
yum localinstall path
```

使用您要安装的软件包的路径替换 path。

9.2.5. 下载软件包

如 例 9.12 “安装过程” 所示，在某个安装过程点，系统会提示您使用以下信息确认安装：

```
...
Total size: 1.2 M
Is this ok [y/d/N]:
...
```

使用 d 选项时，yum 将下载软件包，而不立即安装它们。您稍后可以使用 yum localinstall 命令离线安装这些软件包，也可以与其他设备共享这些软件包。下载的软件包保存在缓存目录的一个子目录中，默认为 /var/cache/yum/$basearch/$releasever/packages/。下载操作以后台模式进行，以便您可以并行使用 yum 进行其他操作。

9.2.6. 删除软件包

与软件包安装类似，yum 可让您卸载它们。要卸载特定软件包以及依赖它的软件包，以 root 用户身份运行以下命令：

```
yum remove package_name&hellip;
```

与安装多个软件包时一样，您可以通过在 命令中添加更多软件包名称来一次性删除多个软件包。

例 9.13. 删除多个软件包

要删除 totem，在 shell 提示符后输入以下内容：

```
~]# yum remove totem
```

与 安装 类似，删除 可以使用这些参数：

- 软件包名称
- glob 表达式
- 文件列表

- 软件包提供

警告

yum 无法删除软件包，除非同时删除依赖于它的软件包。不建议这种类型的操作（只能由 RPM 执行），并可能会使您的系统处于非正常运行状态或导致应用程序无法正常工作或崩溃。

9.3. 使用软件包组

软件包组是满足共同用途的软件包集合，用于实例系统工具或 Sound 和 Video。安装软件包组会拉取一组依赖软件包，从而节省大量时间。yum groups 命令是顶级命令，涵盖在 yum 中对包组执行的所有操作。

9.3.1. 列出软件包组

Summary 选项用于查看已安装组、可用组、可用环境组的数量，以及已安装和可用的语言组：

```
yum groups summary
```

例 9.14. yum group summary 的输出示例

```
~]$ yum groups summary
Loaded plugins: langpacks, product-id, subscription-manager
Available Environment Groups: 12
Installed Groups: 10
Available Groups: 12
```

要列出 yum 存储库中的所有软件包组，请添加 list 选项。您可以根据组名称过滤命令输出。

```
yum group list glob_expression&hellip;
```

可以将几个可选参数传递给此命令，包括 隐藏 列出未标记为用户可见的组，以及用于列出组 ID 的 id。您可以添加 语言、环境、已安装 或 可用的 选项，将命令输出减少到特定组类型。

要列出特定组中的强制和可选软件包，请使用以下命令：

```
yum group info glob_expression&hellip;
```

例 9.15. 查看 LibreOffice 软件包组的信息

```
 ~]$ yum group info LibreOffice
Loaded plugins: langpacks, product-id, subscription-manager

Group: LibreOffice
 Group-Id: libreoffice
 Description: LibreOffice Productivity Suite
 Mandatory Packages:
 =libreoffice-calc
  libreoffice-draw
 -libreoffice-emailmerge
  libreoffice-graphicfilter
 =libreoffice-impress
 =libreoffice-math
 =libreoffice-writer
 +libreoffice-xsltfilter
 Optional Packages:
  libreoffice-base
  libreoffice-pyuno
```

如上例中所示，软件包组中的软件包可以具有使用以下符号标记的不同状态：

- " - " - 未安装包，不会将其作为包组的一部分安装。
- " + " - 包未安装，但将在下一次 yum 升级或 yum 组升级时安装。
- " = " - 包已安装并且作为包组的一部分安装.
- 无符号 - 软件包已安装，但安装在软件包组之外。这意味着 yum group remove 不会删除这些软件包。

这些区别仅在 group_command 配置参数设置为 对象 时才会进行，这是默认设置。如果您不希望 yum跟踪软件包是否作为组的一部分安装或单独安装，则将此参数设置为不同的值，这将"无符号"包，等同于 "=" 软件包。

您可以使用 yum group mark 命令更改上述软件包状态。例如，yum group 将任何给定安装的包标记为 指定组的成员。要避免在组更新时安装新软件包，请使用 yum group mark blacklist。有关 yum group mark 功能的更多信息，请参阅 yum (8)man page。

注意

您可以使用 @^ 前缀识别环境组，软件包组则可标记为 @。使用 yum group list、info、install 或 remove 时，传递 @group_name 以指定软件包组、@^group_name 指定环境组，或者包含它们的 group_name。

9.3.2. 安装软件包组

每个软件包组都有一个名称和组 ID(groupid)。要列出所有软件包组的名称，及其组 ID（以括号中显示），请输入：

```
yum group list ids
```

例 9.16. 查找软件包组的名称和 groupid

要查找软件包组的名称或 ID，例如与 KDE 桌面环境相关的组，请输入：

```
~]$ yum group list ids kde\*
Available environment groups:
  KDE Plasma Workspaces (kde-desktop-environment)
Done
```

某些组由配置的仓库中的设置隐藏。例如，在服务器上，使用 隐藏 命令选项同时列出隐藏组：

```
~]$ yum group list hidden ids kde\*
Loaded plugins: product-id, subscription-manager
Available Groups:
  KDE (kde-desktop)
Done
```

您可以将其完整组名（不带 groupid 部分）传递到 group install 命令，以此安装软件包组。以 root 用户身份键入：

```
yum group install "group name"
```

您还可以按 groupid 安装。以 root 用户身份执行以下命令：

```
yum group install groupid
```

如果您使用 @ 符号附加 groupid 或带引号的组名称，您可以将 groupid 或 quoted 组名传递给 install 命令，该符号告知 yum 想要执行 组安装。以 root 用户身份键入：

```
yum install @group
```

使用 groupid 或带引号的组名称替换 group。相同的逻辑适用于环境组：

```
yum install @^group
```

例 9.17. 安装 KDE 桌面组的四种等效方法

如前文所述，您可以使用四种替代方案，但使用等效的方式来安装软件包组。对于 KDE Desktop，命令如下所示：

```
~]# yum group install "KDE Desktop"
~]# yum group install kde-desktop
~]# yum install @"KDE Desktop"
~]# yum install @kde-desktop
```

9.3.3. 删除软件包组

您可以使用类似于 安装 语法的语法删除软件包组，并使用软件包组的名称或其 id。以 root 用户身份键入：

```
yum group remove group_name
```

```
yum group remove groupid
```

此外，如果您在 remove 命令前面添加 @-symbol（告知 yum 想要执行组删除），则您也可以将 groupid 或带引号的名称传递给 remove 命令。以 root 用户身份键入：

```
yum remove @group
```

使用 groupid 或带引号的组名称替换 group。同样，您可以替换环境组：

```
yum remove @^group
```

例 9.18. 删除 KDE 桌面 组的四种等效方法

与安装类似，您可以使用四种替代方案，但使用等效的方法删除软件包组：对于 KDE Desktop，命令如下所示：

```
~]# yum group remove "KDE Desktop"
~]# yum group remove kde-desktop
~]# yum remove @"KDE Desktop"
~]# yum remove @kde-desktop
```

9.4. 使用事务历史记录

yum history 命令允许用户查看有关 yum 事务时间表、发生日期和时间、受影响的软件包数量、这些事务是否成功还是被中止的信息，以及是否在不同事务之间更改了 RPM 数据库。此外，此命令可用于撤销或恢复某些事务。所有历史记录数据都存储在 /var/lib/yum/history/ 目录中的历史记录 DB 中。

9.4.1. 列出事务

以 root 用户身份显示二十项最新事务的列表，或者在没有额外参数的情况下运行 yum history，或者在 shell 提示符下键入以下内容：

```
yum history list
```

要显示所有事务，请添加 all 关键字：

```
yum history list all
```

要只显示给定范围内的事务，请使用以下格式的命令：

```
yum history list start_id..end_id
```

您也可以仅列出与特定软件包或软件包相关的事务。要做到这一点，使用带有软件包名称或 glob 表达式的命令：

> *yum history list glob_expression…*

> *例 9.19. 列出五个最旧的事务*
>
> *在 yum history 列表中，最新的事务会显示在列表的顶部。要显示历史记录数据库中存储的五个最旧的事务的信息，请输入：*
>
> ```
> ~]# yum history list 1..5
> Loaded plugins: langpacks, product-id, subscription-manager
> ID   | Login user       | Date and time  | Action(s)   | Altered
> -------------------------------------------------------------------------------
>   5 | User <user>      | 2013-07-29 15:33 | Install    |  1
>   4 | User <user>      | 2013-07-21 15:10 | Install    |  1
>   3 | User <user>      | 2013-07-16 15:27 | I, U       |  73
>   2 | System <unset>   | 2013-07-16 15:19 | Update     |  1
>   1 | System <unset>   | 2013-07-16 14:38 | Install    | 1106
> history list
> ```

所有形式的 yum history list 命令生成表格输出，每行由以下列组成：

- *id - 标识特定事务的整数值。*
- *登录用户 - 用于启动事务的登录会话的用户名称。此信息通常显示在 Full Name <username> 表单中。对于不是由用户发布的事务（如自动系统更新），改为使用 System <unset>。*
- *日期和时间 - 签发交易的日期和时间。*
- *action(s) - 事务期间执行的操作列表，如 表 9.1 “Action(s)字段的可能值” 所述。*
- *更改 - 受事务影响的软件包数量，后跟 表 9.2 “Altered 字段的可能值” 所述的附加信息。*

表 9.1. Action(s)字段的可能值

操作	缩写	描述
降级	D	至少一个软件包已降级到较旧版本。

操作	缩写	描述
擦除	**E**	至少已删除一个软件包。
安装	**I**	至少已安装了一个新软件包。
Obsoleting	**O**	至少一个软件包被标记为过时。
重新安装	**R**	至少已重新安装了一个软件包。
Update（更新）	**U**	至少一个软件包已更新为更新的版本。

表 9.2. Altered 字段的可能值

符号	描述
<	在事务完成前，rpm **db** 数据库在 yum 外部更改。
>	事务完成后，rpm **db 数据库** 在 yum 外部被更改。
*****	事务未能完成。
#	事务成功完成，但 yum 返回一个非零退出代码。
E	事务成功完成，但会显示错误或警告。
P	事务成功完成，但 **rpmdb** 数据库中已存在问题。
s	事务成功完成，但是使用了 **--skip-broken** 命令行选项并跳过某些软件包。

要将任何已安装软件包的 rpmdb 或 yumdb 数据库内容与当前使用的 rpmdb 或 yumdb 数据库 同步，请输入以下内容：

```
yum history sync
```

要显示有关当前使用历史记录数据库的一些总体统计信息，请使用以下命令：

```
yum history stats
```

例 9.20. yum history stats 输出示例

```
~]# yum history stats
Loaded plugins: langpacks, product-id, subscription-manager
File    : //var/lib/yum/history/history-2012-08-15.sqlite
Size    : 2,766,848
Transactions: 41
Begin time : Wed Aug 15 16:18:25 2012
End time  : Wed Feb 27 14:52:30 2013
Counts   :
 NEVRAC : 2,204
 NEVRA : 2,204
 NA   : 1,759
 NEVR  : 2,204
 rpm DB : 2,204
 yum DB : 2,204
history stats
```

yum 还允许您显示所有过去事务的摘要。要做到这一点，以 root 用户身份运行以下命令：

```
yum history summary
```

要只显示给定范围内的事务，请输入：

```
yum history summary start_id..end_id
```

与 yum history list 命令类似，您可以通过提供软件包名称或 glob 表达式来显示与特定软件包或软件包相关的事务摘要：

```
yum history summary glob_expression&hellip;
```

例 9.21. 五个最新事务摘要

```
~]# yum history summary 1..5
Loaded plugins: langpacks, product-id, subscription-manager
Login user         | Time         | Action(s)    | Altered
-------------------------------------------------------------------------------
Jaromir ... <jhradilek>  | Last day      | Install     |    1
Jaromir ... <jhradilek>  | Last week     | Install     |    1
Jaromir ... <jhradilek>  | Last 2 weeks    | I, U       |    73
System <unset>       | Last 2 weeks    | I, U       |   1107
history summary
```

所有形式的 yum history summary 命令都会 生成与 yum history list 输出类似的简化表格输出。

如上所示，yum history list 和 yum history summary 都面向事务，尽管它们允许您只显示与给定软件包或软件包相关的事务，但它们缺少重要的详情，如软件包版本。要从软件包的视角列出事务，以 root 用户身份运行以下命令：

```
yum history package-list glob_expression&hellip;
```

例 9.22. 跟踪软件包历史记录

例如，要跟踪 subscription-manager 和相关软件包的历史记录，在 shell 提示符后输入以下内容：

```
~]# yum history package-list subscription-manager\*
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
ID   | Action(s)   | Package
-------------------------------------------------------------------------------
   2 | Updated     | subscription-manager-1.13.22-1.el7.x86_64     EE
   2 | Update      |          1.15.9-15.el7.x86_64     EE
   2 | Obsoleted   | subscription-manager-firstboot-1.13.22-1.el7.x86_64 EE
   2 | Updated     | subscription-manager-gui-1.13.22-1.el7.x86_64   EE
   2 | Update      |           1.15.9-15.el7.x86_64   EE
   2 | Obsoleting  | subscription-manager-initial-setup-addon-1.15.9-15.el7.x86_64 EE
   1 | Install     | subscription-manager-1.13.22-1.el7.x86_64
   1 | Install     | subscription-manager-firstboot-1.13.22-1.el7.x86_64
   1 | Install     | subscription-manager-gui-1.13.22-1.el7.x86_64
history package-list
```

在本例中，初始系统安装过程中安装了三个软件包：subscription -manager、subscription-manager-firstboot 和 subscription-manager-gui。在第三个事务中，所有这些软件包已从 1.10.11 更新至 1.10.17 版本。

9.4.2. 检查事务

要以 root 用户身份显示单个事务的摘要，以以下格式使用 yum history summary 命令：

```
yum history summary id
```

在这里，id 代表事务的 ID。

要更详细地检查特定的事务或事务，以 root 用户身份运行以下命令：

```
yum history info id&hellip;
```

-

id 参数是可选的，当省略它时，yum 会自动使用最后一个事务。请注意，在指定多个事务时，您还可以使用范围：

```
yum history info start_id..end_id
```

例 9.23. yum history info 的输出示例

以下是两个事务的输出示例，每个事务都安装一个新的软件包：

```
~]# yum history info 4..5
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
Transaction ID : 4..5
Begin time   : Mon Dec 7 16:51:07 2015
Begin rpmdb  : 1252:d2b62b7b5768e855723954852fd7e55f641fbad9
End time     :      17:18:49 2015 (27 minutes)
End rpmdb    : 1253:cf8449dc4c53fc0cbc0a4c48e496a6c50f3d43c5
User         : Maxim Svistunov <msvistun>
Return-Code  : Success
Command Line  : install tigervnc-server.x86_64
Command Line  : reinstall tigervnc-server
Transaction performed with:
  Installed   rpm-4.11.3-17.el7.x86_64         @rhel-7-server-rpms
  Installed   subscription-manager-1.15.9-15.el7.x86_64 @rhel-7-server-rpms
  Installed   yum-3.4.3-132.el7.noarch         @rhel-7-server-rpms
Packages Altered:
  Reinstall tigervnc-server-1.3.1-3.el7.x86_64 @rhel-7-server-rpms
history info
```

您还可以查看其他信息，如事务时使用的配置选项，或者从哪个存储库以及安装某些软件包的原因。要确定某个事务有哪些可用的额外信息，以 root 用户身份在 shell 提示符后输入以下内容：

```
yum history addon-info id
```

与 yum history info 类似，如果没有提供 id，yum 会自动使用最新的事务。引用最新事务的另一种方法是使用 最后一个 关键字：

```
yum history addon-info last
```

例 9.24. yum history addon-info输出示例

对于历史记录中的第四个事务，yum history addon-info 命令提供以下输出：

```
~]# yum history addon-info 4
Loaded plugins: langpacks, product-id, subscription-manager
Transaction ID: 4
Available additional history information:
 config-main
 config-repos
 saved_tx

history addon-info
```

yum history addon-info 命令的输出中提供了三种类型的信息：

- config-main - 事务期间使用的全局 yum 选项。有关如何更改全局选项的详情，请查看第 9.5.1 节 “设置 [main] 选项”。
- config-repos - 单个 yum 软件仓库的选项。有关如何更改独立软件仓库选项的详情，请查看 第 9.5.2 节 “设置 [repository] 选项”。
- saved_tx - yum load-transaction 命令可以使用的数据在另一台计算机上重复事务（参见下方）。

要显示所选类型的附加信息，以 root 用户身份运行以下命令：

```
yum history addon-info id information
```

9.4.3. 恢复和重复事务

除了查看事务历史记录外，yum history 命令还提供了恢复或重复所选事务的方法。要恢复事务，以 root 用户身份在 shell 提示符后输入以下内容：

```
yum history undo id
```

要重复特定的事务，以 root 用户身份运行以下命令：

```
yum history redo id
```

两个命令也接受 最后一个关键字来 撤销或重复最新的事务。

请注意，yum history undo 和 yum history redo 命令只会恢复或重复事务期间执行的步骤。如果事务安装了新软件包，yum history undo 命令将将其卸载，如果事务卸载了软件包，命令将再次安装它。如果这些较旧的软件包仍然可用，这个命令还会尝试将所有更新的软件包降级到之前的版本。

管理多个相同的系统时，yum 还允许您对其中一个系统执行事务，将事务详细信息存储在文件中，并在经过一段时间测试后，在剩余系统上重复同样的事务。要将事务详情保存到文件中，以 root 用户身份在 shell 提示符后输入以下内容：

```
yum -q history addon-info id saved_tx > file_name
```

将此文件复制到目标系统后，您可以以 root 用户身份运行以下命令重复事务：

```
yum load-transaction file_name
```

您可以配置 load-transaction 来忽略缺少的软件包或 rpmdb 版本。有关这些配置选项的更多信息，请参阅 yum.conf(5)man page。

9.4.4. 启动新事务历史记录

yum 将事务历史记录存储在单个 SQLite 数据库文件中。要启动新的事务历史记录，以 root 用户身份运行以下命令：

```
yum history new
```

这将在 /var/lib/yum/history/ 目录中创建一个新的空数据库文件。将保留旧的事务历史记录，但只要目录中存在更新的数据库文件，就无法访问。

9.5. 配置 YUM 和 YUM 存储库

注意

为了扩展您的专业知识，您可能还对红帽系统管理三(RH254) 和 RHCSA 快速提升课程(RH199) 培训课程感兴趣。

yum 及相关实用程序的配置信息位于 /etc/yum.conf。此文件包含一个必填 [main] 部分，它允许您设置具有全局效果的 yum 选项，还可包含一个或多个 [repository] 部分，供您设置特定于存储库的选项。

但是，建议您在 /etc/yum.repos.d/ 目录中的新或现有 .repo 文件中定义单独的仓库。您在 /etc/yum.conf 文件的单独 [repository] 部分中 定义的值会覆盖 [main] 部分中设置的值。

本节演示了如何：

- 通过编辑 /etc/yum.conf 配置文件的 [main] 部分来设置全局 yum 选项；
- 通过编辑 /etc/yum .conf 和 /etc/yum. repos.d/ 目录中的 [repository] 部分 为单个仓库设置选项；
- 使用 /etc/yum.conf 中的 yum 变量以及 /etc/yum.repos.d/ 目录中的文件，以便正确处理动态版本和体系结构值；
- 在命令行中添加、启用和禁用 yum 存储库；和
- 设置您自己的自定义 yum 存储库。

9.5.1. 设置 [main] 选项

/etc/yum.conf 配置文件正好包含一个 [main] 部分，本节中的一些键值对会影响 yum 的运行方式，另一些则影响 yum 如何处理存储库。

您可以在 /etc/yum.conf 中的 [main] 部分标题下添加多个附加选项。

/etc/yum.conf 配置文件示例类似如下：

```
[main]
cachedir=/var/cache/yum/$basearch/$releasever
keepcache=0
debuglevel=2
logfile=/var/log/yum.log
exactarch=1
obsoletes=1
gpgcheck=1
plugins=1
installonly_limit=3
```

```
[comments abridged]

# PUT YOUR REPOS HERE OR IN separate files named file.repo
# in /etc/yum.repos.d
```

以下是 [main] 部分中最常用的选项：

assumeyes=value

assumeyes 选项确定 yum 是否提示确认关键操作。使用以下之一替换 value ：

0 （默认）- yum 提示确认其执行的关键操作。

1 - 不提示确认重要的 yum 操作.如果设置了 assumeyes=1，yum 的行为与命令行选项 -y 和 --assumeyes 相同。

cachedir=directory

使用此选项设置 yum 存储其缓存和数据库文件的目录。使用 目录的绝对路径替换 directory。默认情况下，yum 的缓存目录为 /var/cache/yum/$basearch/$releasever/。

有关 $basearch 和 $ releasever yum 变量的描述，请参阅 第 9.5.3 节 “使用 Yum 变量”。

debuglevel=value

这个选项指定 yum 生成的输出调试详情。在这里，值是 1 到 10 之间的整数。设置更高的 调试级别 值会导致 yum 显示更详细的调试输出。debuglevel=2 是默认值，而 debuglevel=0 禁用调试输出。

exactarch=value

使用这个选项时，您可以将 yum 设置为在更新已安装的软件包时考虑正确的架构。将 value 替换为：

0 - 在更新软件包时不考虑确切的架构。

1 （默认）- 在更新包时考虑确切的架构。使用这个设置时，yum 不会安装 32 位体系结构的软件包，以更新已在具有 64 位体系结构的系统中安装的软件包。

exclude=package_name more_package_names

exclude 选项允许您在安装或系统更新期间按关键字排除软件包。通过引用以空格分隔的软件包列表，可实现用于排除的多个软件包列表。允许使用通配符的 shell glob 表达式（如 * 和 ?）。

gpgcheck=value

使用 gpgcheck 选项指定 yum 是否应对包执行 GPG 签名检查。将 value 替换为：

0 - 禁止对所有存储库中的软件包进行 GPG 签名检查，包括本地软件包安装.

1 （默认）- 启用检查所有存储库中所有包的 GPG 签名，包括本地包安装.启用 gpgcheck 后，将检查所有包的签名。

如果在 /etc/yum.conf 文件的 [main] 部分中设置了这个选项，它会为所有存储库设置 GPG 检查规则。但是，您也可以为单个存储库设置 gpgcheck=值 ；即，您可以在一个存储库上启用 GPG 检查，同时禁用另一个存储库。如果 /etc/yum.conf 中存在单个存储库，则设置 gpgcheck=值 会覆盖默认值。

group_command=value

使用 group_command 选项指定 yum group install、yum group upgrade 和 yum group remove 命令如何处理软件包组。在以下位置替换 value ：

simple - 安装软件包组的所有成员.仅升级之前安装的软件包，但不要安装在此期间添加到组中的软件包。

compat - 类似于 simple，但 yum 升级 也会安装自上一次升级以来添加到组中的软件包。

对象 - （默认.）使用这个选项，yum 跟踪之前安装的组，并区分作为组一部分安装的软件包和单独安装的软件包。请查看 例 9.15 “查看 LibreOffice 软件包组的信息”

group_package_types=package_type more_package_types

您可以在调用 yum group install 命令时，指定安装哪些类型的包（可选、默认 或 必需）。默认选择默认 和强制软件包类型。

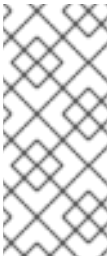
history_record=value

使用这个选项，您可以将 yum 设置为记录事务历史记录。使用以下之一替换 value ：

0 - yum 不应该记录事务的历史记录条目。

1（默认）- yum 应记录事务的历史记录条目.此操作需要一定数量的磁盘空间，并在事务中额外花费一些时间，但它提供了有关过去操作的许多信息，这些信息可通过 yum history 命令显示。history_record=1 是默认值。

有关 yum history 命令的详情请参考 第 9.4 节 “使用事务历史记录”。



注意

yum 使用历史记录来检测对 yum 之外已完成的 rpmdb 数据源的修改。在这种情况下，yum 显示警告并自动搜索更改 rpmdb 导致的问题。当 history_record 关闭后，yum 无法检测到这些更改并且不执行自动检查。

installonlypkgs=空格 分隔 的软件包 列表

您可以在此处提供一个以空格分隔的软件包列表，供 yum 安装，但不会更新。有关默认仅安装的软件包列表，请参阅 yum.conf(5)手册页。

如果将 installonlypkgs 指令添加到 /etc/yum.conf，请确保列出应仅安装 的所有 包，包括 yum.conf(5)的 installonlypkgs 部分下列出的任何包。特别是，请确保内核软件包始终在 installonlypkgs 中列出（默认情况下，当它们是一样），并且 installonly_limit 始终设置为大于 2 的值，以便在默认软件包无法引导时始终可用备份内核。

installonly_limit=value

此选项设置 installonlypkgs 指令中列出的软件包数量，可以同时安装。使用代表 installonlypkgs 中列出的任意单个软件包的最大版本数的整数替换 值。

installonlypkgs 指令的默认值包含多个不同的内核软件包，因此请注意，更改 installonly_limit 的值也会影响单个内核软件包安装版本的最大数量。/etc/yum.conf 中列出的默认值是 installonly_limit=3，最小可能的值为 installonly_limit=2。

您无法设置 installonly_limit=1，因为这会阻止 yum 删除正在运行的内核。如果使用 installonly_limit=1，yum 将失败。

使用 installonly_limit=2 可确保一个备份内核可用。但是，建议保留默认设置 installonly_limit=3，以便您有两个备份内核可用。

keepcache=value

***keepcache** 选项**决定 yum 在安装成功后是否保留**标头**和**软件包的**缓存。在**这里，**值**是 ：*

***0 （默**认）**- 在成功安装后，** 请**勿保留**标头**和**软件包的**缓存。*

***1 - 在成功安装后保留**缓存.*

logfile=file_name

要指定日志**输**出的位置，** 请**将 file_name 替**换为 **yum** 应**在其中写入其日志**输**出的文件的**绝对**路径。默**认**情况下，yum logs to /var/log/yum.log。

max_connenctions=number

此**处**的**值**代表并**发连**接的最大数量，默**认为 **5。

multilib_policy=value

如果有多个架**构**版本可用于**软件包安装，** 则 **multilib_policy** 选项**会**设**置安装行**为。**在**这**里，** 值**代表 ：

***最佳 -** 为这**个系**统**安装最佳**选择**架**构**.例如：在 AMD64 系**统**中**设**置 multilib_policy=best 会**导**致 yum 安装 64 位**软件包**版本。*

***all - 始**终为**每个**软件包安装所有可能的架**构**.例如：当 AMD64 系**统**中将 multilib_policy** 设**置**为 **all** 时，**yum 会安装**软**件包的 i686 和 AMD64 版本（如果**两者都可用）。*

obsoletes=value

obsoletes** 选项**在更新**过**程中**启用**过时**的**进程**逻辑。When 在其 spec 文件中声明它弃用 了** 另**一个**软**件包，在安装了前一个**软**件包**时，**前一个**软**件包会被替**换**掉。例如，当**软**件包被重命名**时，**会声明**过时**的**项。使用以下之一替**换 **value ：

***0 - 在**执**行更新**时**禁用 yum 的**过时处**理**逻辑。*

***1 （默**认）**- 在**执**行更新**时启用 **yum 的**过时处**理**逻辑。*

plugins=value

这是一个全局交换机来启用或禁用 yum 插件，value 是：

0 - 全局禁用所有 yum 插件.

重要

不建议禁用所有插件，因为某些插件提供重要的 yum 服务。特别是 product-id 和 subscription-manager 插件，它支持基于证书的内容交付网络 (CDN)。全局禁用插件作为方便选项，通常仅在诊断 yum 潜在问题时才建议使用。

1 （默认）- 全局启用所有 yum 插件.使用 plugins=1 时，您仍然可以通过在该插件配置文件中设置 enabled=0 来禁用特定的 yum 插件。

有关各种 yum 插件的详情请参考 第 9.6 节 “yum 插件”。有关控制插件的详情请参考 第 9.6.1 节 “启用、配置和禁用 Yum 插件”。

reposdir=directory

此处，目录是指向其中 .repo 文件的目录的绝对路径。all .repo 文件包含存储库信息（类似于 /etc/yum.conf 的 [repository] 部分 ）。yum 从 .repo 文件和 /etc/yum.conf 文件的 [repository] 部分 收集所有存储库信息，以创建要用于事务的存储库的主列表。如果没有设置 reposdir，yum 将使用默认目录 /etc/yum.repos.d/。

retries=value

此选项设置 yum 在返回错误之前应尝试检索文件的次数。值 是一个整数 0 或更高。将值设为 0 时，yum 会永久重试。默认值为 10。

有关可用 [main] 选项的完整列表，请查看 yum.conf(5)手册页中的 [main] OPTIONS 部分。

9.5.2. 设置 [repository] 选项

[repository] 部分，其中 repository 是唯一存储库 ID，如 my_personal_repo （不允许空间），允许您定义单独的 yum 存储库。为避免冲突，自定义存储库不应使用红帽存储库使用的名称。

以下是 [repository] 部分 采用的形式的最小示例：

```
[repository]
name=repository_name
baseurl=repository_url
```

每个 [repository] 部分必须包含以下指令：

name=repository_name

此处 repository_name 是描述存储库的人类可读字符串。

baseurl=repository_url

使用存储库数据目录所在目录的 URL 替换 repository_url ：

- 如果存储库通过 HTTP 提供，请使用： http://path/to/repo
- 如果仓库可以通过 FTP 获得，请使用： ftp://path/to/repo
- 如果存储库对机器是本地的，请使用： file:///path/to/local/repo
- 如果特定的在线存储库需要基本的 HTTP 身份验证，您可以通过将用户名和密码放在 URL 中作为 用户名 ：密码@链接 来指定您的用户名和密码。例如，如果 http://www.example.com/repo/ 上的存储库需要用户名"user"和密码"password"，则 baseurl 链接可以指定为 http://user:/repo/。

  这个 URL 通常是一个 HTTP 链接，例如：

  ```
  baseurl=http://path/to/repo/releases/$releasever/server/$basearch/os/
  ```

  请注意，yum 总是扩展 URL 中的 $releasever、$arch 和 $basearch 变量。有关 yum 变量的详情请参考 第 9.5.3 节 “使用 Yum 变量”。

其他有用的 [repository] 指令包括：

enabled=value

这是告诉 yum 使用或忽略特定库的简单方法，value 是：

0 - 在执行更新和安装时，不要将此存储库作为软件包源包含在内。这是快速打开和关闭存储库的一种简单方法，当您希望从不需要启用更新或安装的仓库中单个软件包时，这很有用。

1 - 将此仓库作为包源包含在内.

也可以通过将 --enablerepo=repo_name 或 --disablerepo=repo_name 选项传递到 yum，或通过 PackageKit 程序的 Add/Remove Software 窗口来执行打开和关闭存储库。

async=value

控制存储库软件包的并行下载。在这里，值是 ：

auto （默认）- 如果可能，将使用并行下载，这意味着 yum 会自动为插件创建的存储库禁用它，以避免故障。

on - 为存储库启用并行下载。

off - 禁止并行下载。

还有 [存储库] 选项，其中一部分具有与特定 [main] 选项 相同的形式和功能。有关完整列表，请查看 yum.conf(5)手册页中的 [repository] OPTIONS 部分。

> **例 9.25. /etc/yum.repos.d/redhat.repo 文件示例**
>
> 以下是 /etc/yum.repos.d/redhat.repo 文件示例：
>
> ```
> #
> # Red Hat Repositories
> # Managed by (rhsm) subscription-manager
> #
>
> [red-hat-enterprise-linux-scalable-file-system-for-rhel-6-entitlement-rpms]
> name = Red Hat Enterprise Linux Scalable File System (for RHEL 6 Entitlement) (RPMs)
> baseurl = https://cdn.redhat.com/content/dist/rhel/entitlement-
> 6/releases/$releasever/$basearch/scalablefilesystem/os
> enabled = 1
> ```

```
gpgcheck = 1
gpgkey = file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
sslverify = 1
sslcacert = /etc/rhsm/ca/redhat-uep.pem
sslclientkey = /etc/pki/entitlement/key.pem
sslclientcert = /etc/pki/entitlement/11300387955690106.pem

[red-hat-enterprise-linux-scalable-file-system-for-rhel-6-entitlement-source-rpms]
name = Red Hat Enterprise Linux Scalable File System (for RHEL 6 Entitlement) (Source
RPMs)
baseurl = https://cdn.redhat.com/content/dist/rhel/entitlement-
6/releases/$releasever/$basearch/scalablefilesystem/source/SRPMS
enabled = 0
gpgcheck = 1
gpgkey = file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
sslverify = 1
sslcacert = /etc/rhsm/ca/redhat-uep.pem
sslclientkey = /etc/pki/entitlement/key.pem
sslclientcert = /etc/pki/entitlement/11300387955690106.pem

[red-hat-enterprise-linux-scalable-file-system-for-rhel-6-entitlement-debug-rpms]
name = Red Hat Enterprise Linux Scalable File System (for RHEL 6 Entitlement) (Debug
RPMs)
baseurl = https://cdn.redhat.com/content/dist/rhel/entitlement-
6/releases/$releasever/$basearch/scalablefilesystem/debug
enabled = 0
gpgcheck = 1
gpgkey = file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
sslverify = 1
sslcacert = /etc/rhsm/ca/redhat-uep.pem
sslclientkey = /etc/pki/entitlement/key.pem
sslclientcert = /etc/pki/entitlement/11300387955690106.pem
```

9.5.3. 使用 Yum 变量

您可以在 yum 命令和所有 yum 配置文件（即 /etc/yum .conf 和 /etc/yum. repos. d/ 目录中的 all.repo 文件）中使用和引用以下内置变量：

$releasever

您可以使用这个变量来引用 Red Hat Enterprise Linux 的发行版本。yum 从 /etc/yum.conf 配置文件中的 distroverpkg=value 行获取 $releasever 的值。如果 /etc/yum.conf 中没有此类行，那么 yum 会从提供 redhat-release 文件的 redhat-release 产品 包中获得版本号来推断正确的值。

$arch

您可以在调用 Python 的s os.uname（） 功能时，使用此变量指代系统的 CPU 架构。$arch 的有效值包括： i586、i686 和 x86_64。

$basearch

您可以使用 $basearch 来引用系统基本架构。例如，i686 和 i586 机器都具有 i386 基本架构，AMD64 和 Intel 64 机器基本架构为 x86_64。

$YUM0-9

这些十个变量各自替换为同名任何 shell 环境变量的值。如果其中一个变量被引用（例如在 /etc/yum.conf 中）和一个具有相同名称的 shell 环境变量不存在，则不会替换配置文件变量。

要定义自定义变量或覆盖现有变量的值，请在 /etc/yum/vars/ 目录中创建一个名称与变量相同的文件（不含 "$" 符号），并在第一行中添加所需的值。

例如，存储库描述通常包含操作系统名称。要定义名为 $osname 的新变量，请在第一行中创建一个带有"Red Enterprise Linux"的新文件，并将它保存为 /etc/yum/vars/osname:

```
~]# echo "Red Hat Enterprise Linux 7" > /etc/yum/vars/osname
```

现在，您可以在 .repo 文件中使用以下内容而不是 " Red Hat Enterprise Linux 7"：

```
name=$osname $releasever
```

9.5.4. 查看当前配置

要显示全局 yum 选项（即 /etc/yum.conf 文件的 [main] 部分指定的选项）的当前值，请使用不带命令行选项的 yum-config-manager 命令执行 yum-config-manager 命令：

```
yum-config-manager
```

要列出不同配置部分或部分的内容，请使用以下格式的命令：

```
yum-config-manager section&hellip;
```

您还可以使用 glob 表达式显示所有匹配部分的配置：

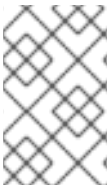
```
yum-config-manager glob_expression&hellip;
```

例 9.26. 查看主部分的配置

要列出所有配置选项及其主部分对应的值，在 shell 提示符后输入以下内容：

```
~]$ yum-config-manager main \*
Loaded plugins: langpacks, product-id, subscription-manager
================================== main ==================================
[main]
alwaysprompt = True
assumeyes = False
bandwith = 0
bugtracker_url = https://bugzilla.redhat.com/enter_bug.cgi?
product=Red%20Hat%20Enterprise%20Linux%206&component=yum
cache = 0
[output truncated]
```

9.5.5. 添加、启用和禁用 Yum 存储库



注意

为了扩展您的专业知识，您可能还对红帽系统管理三(RH254) 培训课程感兴趣。

第 9.5.2 节 “设置 [repository] 选项” 描述用来定义 yum 存储库的各种选项。本节介绍如何使用 yum-config-manager 命令添加、启用和禁用存储库。

重要

使用红帽订阅管理注册到基于证书的内容交付网络 (CDN)时，Red Hat Subscription Manager 工具用于管理 /etc/yum.repos.d/redhat.repo 文件中的存储库。

添加 Yum 存储库

要定义新存储库，您可以将 [repository] 部分添加到 /etc/yum.conf 文件，也可以添加到 /etc/yum.repos.d/ 目录中的 a. repo 文件。此目录中带有 .repo 文件扩展名的所有文件都由 yum 读取，建议您在此处定义您的程序库，而不是在 /etc/yum.conf 中定义。

警告

从红帽的内容交付网络 (CDN)之外的不受验证或不受信任的软件来源获取和安装软件包会带来潜在的安全风险，并可能导致安全性、稳定性、兼容性和可维护性问题。

Yum 存储库通常提供自己的 .repo 文件。要在您的系统中添加此类存储库并启用它，以 root 用户身份运行以下命令：

```
yum-config-manager --add-repo repository_url
```

... repository_url 是指向 .repo 文件的链接。

例 9.27. 添加 example.repo

要添加位于 http://www.example.com/example.repo 的存储库，在 shell 提示符下输入以下内容：

```
~]# yum-config-manager --add-repo http://www.example.com/example.repo
Loaded plugins: langpacks, product-id, subscription-manager
adding repo from: http://www.example.com/example.repo
grabbing file http://www.example.com/example.repo to /etc/yum.repos.d/example.repo
example.repo                                             | 413 B   00:00
repo saved to /etc/yum.repos.d/example.repo
```

启用 Yum 存储库

要启用特定的存储库或存储库，以 root 用户身份在 shell 提示符后输入以下内容：

```
yum-config-manager --enable repository&hellip;
```

...其中 repository 是唯一的存储库 ID（使用 yum repolist all 列出可用的存储库 ID）。或者，您可以使用 glob 表达式启用所有匹配的软件仓库：

```
yum-config-manager --enable glob_expression&hellip;
```

例 9.28. 启用 /etc/yum.conf 的自定义部分中定义的仓库.

要启用 [example]、[example -debuginfo] 和 [example-source]部分中定义的软件仓库，请输入：

```
~]# yum-config-manager --enable example\*
Loaded plugins: langpacks, product-id, subscription-manager
============================== repo: example ==============================
[example]
bandwidth = 0
base_persistdir = /var/lib/yum/repos/x86_64/7Server
baseurl = http://www.example.com/repo/7Server/x86_64/
cache = 0
cachedir = /var/cache/yum/x86_64/7Server/example
[output truncated]
```

例 9.29. 启用所有软件仓库

要启用 /etc/yum .conf 文件和 /etc/yum. repos.d/ 目录中定义的所有软件仓库，请输入：

```
~]# yum-config-manager --enable \*
Loaded plugins: langpacks, product-id, subscription-manager
============================== repo: example ==============================
[example]
bandwidth = 0
base_persistdir = /var/lib/yum/repos/x86_64/7Server
baseurl = http://www.example.com/repo/7Server/x86_64/
cache = 0
cachedir = /var/cache/yum/x86_64/7Server/example
[output truncated]
```

成功后，yum-config-manager --enable 命令显示当前的存储库配置。

禁用 Yum 存储库

要禁用 yum 存储库，以 root 用户身份运行以下命令：

```
yum-config-manager --disable repository&hellip;
```

…其中 repository 是唯一的存储库 ID（使用 yum repolist all 列出可用的存储库 ID）。与 yum-config-manager --enable 类似，您可以使用 glob 表达式同时禁用所有匹配的存储库：

```
yum-config-manager --disable glob_expression&hellip;
```

例 9.30. 禁用所有软件仓库

要禁用在 /etc/yum.conf 文件和 /etc/yum. repos.d/ 目录中定义的所有软件仓库，请输入：

```
~]# yum-config-manager --disable \*
Loaded plugins: langpacks, product-id, subscription-manager
============================== repo: example ==============================
[example]
bandwidth = 0
base_persistdir = /var/lib/yum/repos/x86_64/7Server
baseurl = http://www.example.com/repo/7Server/x86_64/
cache = 0
cachedir = /var/cache/yum/x86_64/7Server/example
[output truncated]
```

成功后，yum-config-manager --disable 命令显示当前配置。

9.5.6. 创建 Yum 存储库

设置 yum 存储库：

1. 安装 createrepo 软件包：

```
# yum install createrepo
```

2. 将新存储库的所有软件包复制到一个目录中，如 /tmp/local_repo/ ：

```
cp /your/packages/*.rpm /tmp/local_repo/
```

3. 创建存储库运行：

```
createrepo /tmp/local_repo/
```

这会为 yum 存储库创建必要的元数据，并将元数据放置在新创建的子目录 repodata 中。

现在，该程序库可以被 yum 使用。此存储库可以通过 HTTP 或 FTP 协议共享，或者直接从本地计算机引用。有关如何配置 yum 软件仓库的详情，请查看 第 9.5.2 节 “设置 [repository] 选项” 部分。

注意

在构建存储库的 URL 时，请参考 /mnt/local_repo，而不是 /mnt/local_repo/repodata，因为此目录仅包含元数据。实际的 yum 软件包位于 /mnt/local_repo 中。

9.5.6.1. 将软件包添加到已创建的 yum 软件仓库

将软件包添加到已创建的 yum 存储库：

1. 将新软件包复制到您的仓库目录中，如 /tmp/local_repo/:

```
cp /your/packages/*.rpm /tmp/local_repo/
```

2. 要反映元数据中新添加的软件包，请运行：

```
createrepo --update /tmp/local_repo/
```

3. 可选：如果您已经使用任何 yum 命令用于新更新的存储库，请运行：

```
yum clean expire-cache
```

9.5.7. 添加 Optional 和 Supplementary 存储库

可选和补充订阅频道为 Red Hat Enterprise Linux 提供了额外的软件包，包括开源许可软件（在可选频道中）和专有许可软件（在 Supplementary 频道中）。

在订阅 Optional 和 Supplementary 频道前，请查看覆盖范围详情。如果您决定从这些频道安装软件包，请按照红帽客户门户网站中名为 How to access Optional 和 Supplementary 频道以及 -devel 软件包(RHSM)中的步骤进行操作。

9.6. YUM 插件

yum 提供扩展和增强操作的插件。默认安装某些插件。每当您调用任何 yum 命令时，yum 始终会通知您加载和活动的插件（如果有）。例如：

```
~]# yum info yum
Loaded plugins: langpacks, product-id, subscription-manager
[output truncated]
```

请注意，Loaded 插件后面的插件 名称是您可以向 --disableplugin=plugin_name 选项提供的名称。

9.6.1. 启用、配置和禁用 Yum 插件

要启用 yum 插件，请确保 /etc/yum.conf 的 [main] 部分中存在以 plugins= 开头的行，并确保其值为 1 ：

```
plugins=1
```

您可以通过将此行更改为 plugins=0 来禁用所有插件。

重要

不建议禁用所有插件，因为某些插件提供重要的 yum 服务。特别是 product-id 和 subscription-manager 插件，它们为基于证书的 内容发布网络 (CDN)提供支持。全局禁用插件作为方便选项，通常仅在诊断 yum 潜在问题时才建议使用。

每个安装的插件在 /etc/yum/pluginconf.d/ 目录中都有自己的配置文件。您可以在这些文件中设置插件特定选项。例如，以下是 aliases 插件的 aliases .conf 配置文件：

```
[main]
enabled=1
```

与 /etc/yum.conf 文件类似，插件配置文件始终包含 [main] 部分，其中 enabled= 选项控制在运行 yum 命令时插件是否启用。如果缺少这个选项，您可以手动将其添加到该文件中。

如果您通过在 /etc/yum.conf 中设置 enabled=0 来禁用所有插件，则所有插件都会禁用，无论它们的各个配置文件中是否启用了它们。

如果您只想禁用单个 yum 命令的所有 yum 插件，请使用 --noplugins 选项。

如果要为单个 yum 命令禁用一个或多个 yum 插件，请在该命令中添加 --disableplugin=plugin_name 选项。例如，要在更新系统时禁用 aliases 插件，请输入：

```
~]# yum update --disableplugin=aliases
```

您提供给 --disableplugin= 选项的插件名称与任何 yum 命令输出中 Loaded plugins 行后列出的名称相同。您可以通过逗号分隔多个插件。另外，您可以使用 glob 表达式匹配多个插件名称或缩短长名称：

```
~]# yum update --disableplugin=aliases,lang*
```

9.6.2. 安装额外的 Yum 插件

yum 插件通常遵循 yum-plugin-plugin_name package-naming 规则，但并不总是如此：提供 kabi 插件的软件包名为 kabi-yum-plugins。您可以像安装其他软件包一样安装 yum 插件。例如，要安装 yum-aliases 插件，在 shell 提示符后输入以下内容：

```
~]# yum install yum-plugin-aliases
```

9.6.3. 使用 Yum 插件

以下列表提供了多个有用的 yum 插件的说明和用法说明。插件按照名称列出，方括号中含有软件包的名称。

search-disabled-repos (subscription-manager)

search-disabled-repos 插件允许您临时或永久启用禁用的软件仓库，以帮助解决依赖项。启用此插件后，当 Yum 由于依赖项解析失败而无法安装软件包时，它提供 暂时启用禁用的存储库并重试。如果安装成功，Yum 也提供可永久启用已使用存储库的存储库。请注意，该插件仅适用于由 subscription-manager 管理的存储库，而不是自定义存储库。

重要

如果使用 --assumeyes 或 -y 选项执行 yum，或者在 /etc/yum.conf 中启用了 assumeyes 指令，则插件可在不提示确认的情况下暂时和永久启用禁用的存储库。这可能会导致问题，例如启用您不想启用的软件仓库。

要配置 search-disabled-repos 插件，编辑位于 /etc/yum/pluginconf.d/search-disabled-repos.conf 中的配置文件。有关您可以在 [main] 部分中使用的指令列表，请参考下表。

表 9.3. 支持的 search-disabled-repos.conf 指令

指令	描述
enabled=*value*	允许您启用或禁用插件。该 *值* 必须是 **1**（启用）或 **0**（禁用）。插件默认启用。
notify_only=*value*	允许您将插件的行为限制为仅通知。该 *值* 必须是 **1**（仅不修改 Yum 的行为）或 **0**（修改 Yum 的行为）。默认情况下，插件仅通知用户。
ignored_repos=*repositories*	允许您指定插件不会启用的存储库。

kabi (kabi-yum-plugins)

kabi 插件检查驱动程序更新软件包是否符合官方 Red Hat kernel Application Binary Interface(kABI)。启用此插件后，当用户尝试安装使用不在白名单中的内核符号的软件包时，会将警告消息写入系统日志。另外，将插件配置为以强制模式运行可防止安装此类软件包。

要配置 kabi 插件，请编辑位于 /etc/yum/pluginconf.d/kabi.conf 中的配置文件。下表中显示了 [main] 部分中可以使用的指令列表。

表 9.4. 支持的 kabi.conf 指令

指令	描述
enabled=*value*	允许您启用或禁用插件。该 *值* 必须是 **1**（启用）或 **0**（禁用）。安装之后，插件会被默认启用。
whitelists=*directory*	*允许您指定包含内核符号的文件所在的目录*。默认情况下，kabi 插件使用 kernel-abi-whitelists 软件包（即 **/usr/lib/modules/kabi-rhel70/** 目录）提供的文件。
enforce=*value*	允许您启用或禁用强制模式。该 *值* 必须是 **1**（启用）或 **0**（禁用）。默认情况下，这个选项被注释掉，kabi 插件只会显示警告消息。

product-id (subscription-manager)

product-id 插件管理从 Content Delivery Network 安装的产品的产品身份证书。product-id 插件会被默认安装。

语言包(yum-langpacks)

langpacks 插件用于为安装的每个软件包搜索所选语言的区域化软件包。langpacks 插件默认安装。

别名 (yum-plugin-aliases)

aliases 插件 添加 alias 命令行选项， 该选项允许为 yum 命令配置和使用别名。

yum-changelog (yum-plugin-changelog)

yum-changelog 插件添加 --changelog 命令行选项，可启用在更新之前和之后查看软件包更改日志。

yum-tmprepo (yum-plugin-tmprepo)

yum-tmprepo 插件添加 --tmprepo 命令行选项，该选项采用存储库文件的 URL，下载并启用该存储库文件。此插件会尝试确保安全地临时使用存储库。默认情况下，它不允许禁用 gpg 检查。

yum-verify (yum-plugin-verify)

yum-verify 插件添加了 verify 、 valid-rpm 和 verify-all 命令行选项，用于查看系统上的验证数据。

yum-versionlock (yum-plugin-versionlock)

yum-versionlock 插件排除所选软件包的其他版本，这可防止软件包被新版本更新。使用 versionlock 命令行选项，您可以查看和编辑锁定的软件包列表。

9.7. 使用 YUM-CRON 自动刷新软件包数据库和下载更新

yum-cron 服务会自动检查并下载软件包更新。安装后 yum-cron 服务提供的 cron 作业将立即处于活动状态。yum-cron 服务也可以自动安装下载的更新。

使用默认设置，yum-cron 服务：

- 每小时更新 yum 缓存中的元数据。
- 将待处理软件包更新下载至 yum 缓存每天一次。如果存储库中提供了新软件包，则会发送电子邮件。如需更多信息，请参阅第 第 9.7.2 节 “设置可选电子邮件通知” 章。

yum-cron 服务有两个配置文件：

/etc/yum/yum-cron.conf

用于日常任务。

/etc/yum/yum-cron-hourly.conf

用于每小时任务。

9.7.1. 启用自动安装更新

要启用自动安装下载的更新，请通过设置 apply_updates 选项来编辑每日安装的配置文件或每小时安装的配置文件：

```
apply_updates = yes
```

9.7.2. 设置可选电子邮件通知

默认情况下，yum-cron 服务使用 cron 来发送包含已执行命令输出的电子邮件。此电子邮件按照 cron 配置发送，通常发送到本地超级用户，并存储在 /var/spool/mail/root 文件中。

您可以使用不同于影响所有 cron 作业 的设置的特定电子邮件配置。但是，此电子邮件配置不支持 TLS，整个电子邮件内置逻辑非常基本。

启用 yum-cron 内置电子邮件通知：

1. 打开所选 yum-cron 配置文件：

   /etc/yum/yum-cron.conf

   用于日常任务。

   /etc/yum/yum-cron-hourly.conf

   用于每小时任务。

2. 在 [emitters] 部分，设置以下选项：

   ```
   emit_via = email
   ```

3. 根据需要设置 email_from、email_to、email_host 选项

9.7.3. 启用或禁用特定存储库

yum-cron 不支持对存储库的特定配置。作为为 yum-cron 启用或禁用特定软件仓库的一个临时解决方案，但一般不要为 yum 启用或禁用以下步骤：

1. 创建系统任意位置的空存储库配置目录。

2. 将 /etc/yum.repos.d/ 目录中的所有配置文件复制到此新创建的目录中。

3. 在 /etc/yum.repos.d/ 中的对应 .repo 配置文件中，按如下所示设置 启用 的选项：

已启用 = 1

启用存储库：

已启用 = 0

以禁用存储库。

4. 在所选 yum-cron 配置文件的末尾添加以下选项，它指向新创建的存储库目录：

```
reposdir=/path/to/new/reposdir
```

9.7.4. 测试 Yum-cron 设置

在不等待下一次调度 yum-cron 任务的情况下测试 yum-cron 设置：

1. 打开所选 yum-cron 配置文件：

/etc/yum/yum-cron.conf

用于日常任务。

/etc/yum/yum-cron-hourly.conf

用于每小时任务。

2. 在所选配置文件中设置 random_sleep 选项，如下所示：

```
random_sleep = 0
```

3. 运行配置文件：

```
# yum-cron /etc/yum/yum-cron.conf
# yum-cron /etc/yum/yum-cron-hourly.conf
```

9.7.5. 禁用 Yum-cron 消息

yum-cron 消息无法完全禁用，但只能限制为具有关键优先级的消息。限制信息：

1. 打开所选 yum-cron 配置文件：

/etc/yum/yum-cron.conf

用于日常任务。

/etc/yum/yum-cron-hourly.conf

用于每小时任务。

2. 在配置文件的 [base] 部分中设置以下选项：

```
debuglevel = -4
```

9.7.6. 自动清除软件包

yum-cron 服务不支持删除与 yum clean all 命令类似的包的任何配置选项。要自动清理软件包，您可以将 cron 作业创建为可执行 shell 脚本：

1. 在 /etc/cron.daily/ 目录中创建一个 shell 脚本，其中包含：

```
#!/bin/sh
yum clean all
```

2. 使脚本可执行：

```
# chmod +x /etc/cron.daily/script-name.sh
```

9.8. 其它资源

有关如何在 Red Hat Enterprise Linux 中管理软件包的详情，请查看以下列出的资源。

安装的文档

- yum(8)- yum 命令行实用程序的 man page 提供了所支持选项和命令的完整列表。
- yumdb(8)- yumdb 命令行实用程序的 man page 文档如何使用此工具查询并根据需要更改 yum 数据库。
- yum.conf(5)- 名为 yum.conf 的 man page 包括了可用的 yum 配置选项。
- yum-utils(1)- 名为 yum-utils 列表的 man page，简略介绍用于管理 yum 配置、操作存储库和使用 yum 数据库的其他实用程序。

在线资源

- yum Guides - 项目主页上的 Yum 指南页面提供了进一步文档的链接。
- 红帽客户门户 Labs - 红帽客户门户网站 Labs 包括"Yum Repository Configuration Helper"。

另请参阅

- 第 6 章 获取特权 文档如何使用 su 和 sudo 命令获得管理权限。

部分 IV. 基础架构服务

这部分提供有关如何配置服务和守护进程以及启用对 Red Hat Enterprise Linux 计算机的远程访问的信息。

第 10 章 使用 SYSTEMD 管理服务

10.1. SYSTEMD 简介

Systemd 是 Linux 操作系统的系统和服务管理器。它设计为与 SysV init 脚本向后兼容，并提供许多功能，如在引导时并行启动系统服务、按需激活后台程序或基于依赖项的服务控制逻辑。在 Red Hat Enterprise Linux 7 中，systemd 替换 Upstart 作为默认的 init 系统。

systemd 引进了 systemd 单元的概念。这些单元由位于 表 10.2 “systemd 单元文件位置” 中列出的目录中的单元配置文件来表示，并封装有关系统服务、侦听套接字以及与 init 系统相关的其他对象的信息。有关可用 systemd 单元类型的完整列表，请参阅 表 10.1 “可用的 systemd 单元类型”。

表 10.1. 可用的 systemd 单元类型

单位类型	文件扩展	描述
服务单元	**.service**	系统服务。
目标单元	**.target**	一组 systemd 单元。
Automount 单元	**.automount**	文件系统自动挂载点。
设备单元	**.device**	内核可识别的设备文件。
挂载单位	**.mount**	文件系统挂载点。
路径单元	**.path**	文件系统中的一个文件或者目录。
Scope 单元	**.scope**	外部创建的进程。
Slice 单元	**.slice**	一组管理系统进程的分层组织单元。
快照单元	**.snapshot**	已保存的 systemd 管理器状态。
套接字单元	**.socket**	进程间的通信套接字。
Swap 单元	**.swap**	一个交换设备或者一个交换文件。
计时器单元	**.timer**	systemd 计时器。

表 10.2. systemd 单元文件位置

目录	描述
/usr/lib/systemd/system/	安装的 RPM 软件包中的 systemd 单元文件。
/run/systemd/system/	在运行时创建的 systemd 单元文件。该目录优先于安装了的服务单元文件的目录。
/etc/systemd/system/	Systemd 单元文件由 **systemctl enable** 命令创建，并添加用于扩展服务的单元文件。这个目录优先于带有运行时单元文件的目录。

使用 system.conf 覆盖默认 systemd 配置

默认 systemd 配置是在编译过程中定义的，可以在 /etc/systemd/system.conf 中的 systemd 配置文件中找到。如果您想与那些默认值分离，并全局覆盖所选的 systemd 单元默认值，请使用这个文件。

例如，若要覆盖设为 90 秒的超时限制的默认值，可使用 DefaultTimeoutStartSec 参数输入所需的值（以秒为单位）。

```
DefaultTimeoutStartSec=required value
```

*另请参阅 **例 10.21 “更改超时限制”**。*

10.1.1. 主要功能

在 Red Hat Enterprise Linux 7 中，systemd 系统和服务管理器提供以下主要功能：

- ***基于套接字的激活 - 在引导时，systemd 会为支持这类激活的所有系统服务创建侦听套接字，并在套接字启动后立即将套接字传递给这些服务。这不仅允许 systemd 并行启动服务，还使得可以在服务不可用时重新启动服务而不丢失发送到该服务的任何消息：对应的套接字可被访问，所有消息都已排队。***

  Systemd 使用套接字单元进行基于套接字的激活。

- ***基于总线激活 - 将 D-Bus 用于进程间通信的系统服务可以按需启动，当客户端应用第一次尝试与之通信时。Systemd 使用 D-Bus 服务文件进行基于总线的激活。***

- ***基于设备的激活 - 当特定类型的硬件插入或可用时，支持基于设备的激活的系统服务可以按需启动。Systemd 使用设备单元进行基于设备的激活。***

- 基于路径激活 - 当特定文件或目录更改其状态时，支持基于路径激活的系统服务可以按需启动。Systemd 使用路径单元作为基于路径的激活。

- 挂载和自动挂载点管理 - Systemd 监控和管理挂载和自动挂载点。Systemd 使用 mount 单元作为挂载点，自动挂载单元用于自动挂载点。

- 积极并行化 - 由于使用了基于套接字的激活，因此所有监听套接字都就 systemd 可以并行启动系统服务。和支持按需激活的系统服务相结合，并行激活可大大减少引导系统所需的时间。

- 事务性单元激活逻辑 - 在激活或停用单元之前，systemd 计算其依赖项，创建临时交易，并验证此事务是否一致。如果事务不一致，systemd 会自动尝试更正它并从中删除非必要作业，然后再报告错误。

- 与 SysV init 的后向兼容性 - Systemd 支持 SysV init 脚本，如 Linux 标准基础核心规范 中所述，简化了到 systemd 服务单元的升级路径。

10.1.2. 兼容性更改

systemd 系统和服务管理器的主要设计思想是与 SysV init 和 Upstart 兼容。以下是与之前的 Red Hat Enterprise Linux 系统主发行版本相关的最显著兼容性变化：

- Systemd 仅对运行级别提供有限支持。它提供了多个可直接映射到这些运行级别的目标单元，出于兼容性的原因，它也通过之前的 运行级别 命令进行分发。然而，并非所有 systemd 目标都可以直接映射到运行级别，因此此命令可能会返回 N 来指明未知运行级别。建议您尽可能避免使用 runlevel 命令。

  有关 systemd 目标及其与运行级别比较的详情请参考 第 10.3 节 “使用 systemd 目标”。

- systemctl 程序不支持自定义命令。除了 启动、停止 和 状态 等标准命令外，SysV init 脚本的作者还可以实施对任意命令的支持，以提供额外的功能。例如，可以使用 panic 命令执行 Red Hat Enterprise Linux 6 中 iptables 的 init 脚本，该命令会立即启用 panic 模式，再重新配置系统以开始丢弃所有传入和传出数据包。systemd 不支持，systemctl 只 接受记录的命令。

  有关 systemctl 工具及其与之前 的服务 工具比较的更多信息，请参阅 第 10.2 节 “管理系统服务”。

- systemctl 实用程序不会与 systemd 启动的服务通信。当 systemd 启动系统服务时，它会保存其主进程的 ID 以跟踪它。然后，systemctl 程序使用这个 PID 来查询和管理该服务。因此，如果用户直接在命令行启动某个特定的守护进程，systemctl 就无法决定其当前状态或停止它。

- systemd 只停止运行的服务。在以前的版本中，当启动关闭序列时，Red Hat Enterprise Linux 6 及更早版本的系统使用位于 /etc/rc0.d/ 目录中的符号链接来停止所有可用系统服务，而不考虑它们的状态。使用 systemd 时，只有运行的服务才会在关闭时停止。

- 系统服务无法从标准输入流读取。当 systemd 启动服务时，它会将其标准输入连接到 /dev/null，以防止与用户进行任何交互。

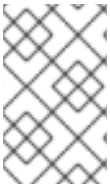
- 系统服务不继承调用用户及其会话的任何上下文（如 HOME 和 PATH 环境变量）。每个服务在干净的执行上下文中运行。

- 加载 SysV 初始化脚本时，systemd 会读取 Linux Standard Base(LSB)标头中编码的依赖信息，并在运行时对其进行解读。

- 服务单元中的所有操作都会有默认的 5 分钟超时，以防止出现故障的服务中断。这个值在从 initscripts 生成的且无法更改的服务中是被硬编码的。但是，单独的配置文件可用于指定每个服务更长的超时值，请参阅 例 10.21 “更改超时限制”

有关 systemd 引入的兼容性更改的详细列表，请参阅红帽企业 Linux 7 的迁移规划指南。

10.2. 管理系统服务



注意

为了扩展您的专业知识，您可能还对红帽系统管理二(RH134) 培训课程感兴趣。

之前的 Red Hat Enterprise Linux 版本使用 SysV init 或 Upstart 分发，使用位于 /etc/rc.d/init.d/ 目录的 init 脚本。这些初始化脚本通常使用 Bash 编写，并允许系统管理员控制其系统中的服务和守护进程状态。在 Red Hat Enterprise Linux 7 中，这些初始化脚本已被服务单元替代。

服务单元带有 .service 文件扩展名，其用途与初始化脚本类似。要查看、启动、停止、重启、启用或禁用系统服务，请使用 systemctl 命令，如 表 10.3 “服务实用程序与 systemctl 的比较”、表 10.4

"chkconfig 实用程序与 systemctl 的比较" 及其他部分所述。在系统中仍可使用 service 和 chkconfig 命令，并可按预期工作。这只用于兼容性，应该尽量不使用。

表 10.3. 服务实用程序与 systemctl 的比较

service	systemctl	描述
service *name* start	**systemctl start *name*.service**	启动一个服务。
service *name* stop	**systemctl stop *name*.service**	停止服务。
service *name* restart	**systemctl restart *name*.service**	重启服务。
service *name* condrestart	**systemctl try-restart *name*.service**	仅在运行时重启服务。
service *name* reload	**systemctl reload *name*.service**	重新加载配置。
service *name* status	**systemctl status *name*.service** **systemctl is-active *name*.service**	检查服务是否在运行。
service --status-all	**systemctl list-units --type service --all**	显示所有服务的状态。

表 10.4. chkconfig 实用程序与 systemctl 的比较

chkconfig	systemctl	描述
chkconfig *name* on	**systemctl enable *name*.service**	启用服务。
chkconfig *name* off	**systemctl disable *name*.service**	禁用服务。
chkconfig --list *name*	**systemctl status *name*.service** **systemctl is-enabled *name*.service**	检查是否启用了服务。
chkconfig --list	**systemctl list-unit-files --type service**	列出所有服务并检查是否启用它们。

chkconfig	systemctl	描述
chkconfig --list	**systemctl list-dependencies --after**	列出在指定单元前排序启动的服务。
chkconfig --list	**systemctl list-dependencies --before**	列出在指定单元之后排序启动的服务。

指定服务单元

为清楚起见，本节其余部分中的所有命令示例都使用带有 .service 文件扩展名的完整单元名称，例如：

```
~]# systemctl stop nfs-server.service
```

但是，可以省略文件扩展名，在这种情况下，system ctl 实用程序假定参数是服务单元。以下命令等同于以上命令：

```
~]# systemctl stop nfs-server
```

此外，某些单元具有别名名称。这些名称的名称比单元短，可以使用它们，而不是实际的单元名称。要查找可用于特定单元的所有别名，请使用：

```
~]# systemctl show nfs-server.service -p Names
```

systemctl 在 chroot 环境中的行为

如果您使用 chroot 命令更改根目录，大多数 systemctl 命令会拒绝执行任何操作。原因在于 systemd 进程和使用 chroot 命令的用户对 文件系统没有相同的视图。当从 kickstart 文件中调用 systemctl 时会出现这种情况。

一个例外是单元文件命令，如 systemctl enable 和 systemctl disable 命令。这些命令不需要运行中的系统，也不会影响正在运行的进程，但它们确实会影响单元文件。因此，即使在 chroot 环境中也可以运行这些命令。例如：要在 /srv/website1/ 目录下的系统中启用 httpd 服务：

```
~]# chroot /srv/website1
~]# systemctl enable httpd.service
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service, pointing to
/usr/lib/systemd/system/httpd.service.
```

10.2.1. 列出服务

要列出所有目前载入的服务单元，在 shell 提示下键入以下内容：

```
systemctl list-units --type service
```

对于每个服务单元文件，此命令会显示其全名(UNIT)，后跟一个备注，该单元文件是否已加载(LOAD)、其高级(ACTIVE)和低级(SUB)单元文件激活状态，以及简短描述(DESCRIPTION)。

默认情况下，systemctl list-units 命令只显示活跃的单位。如果您想列出所有载入的单元，无论它们的状态如何，请使用 --all 或 -a 选项。

```
systemctl list-units --type service --all
```

您还可以列出所有可用服务单元以查看是否启用了它们。要做到这一点，请键入：

```
systemctl list-unit-files --type service
```

对于每个服务单元，此命令会显示其全名(UNIT FILE)，后跟服务单元是否已启用(STATE)的信息。有关如何确定独立服务单元状态的详情请参考 第 10.2.2 节 “显示服务状态”。

例 10.1. 列出服务

要列出所有目前载入的服务单元，请运行以下命令：

```
~]$ systemctl list-units --type service
UNIT              LOAD  ACTIVE SUB    DESCRIPTION
abrt-ccpp.service       loaded active exited  Install ABRT coredump hook
abrt-oops.service       loaded active running ABRT kernel log watcher
abrt-vmcore.service     loaded active exited  Harvest vmcores for ABRT
abrt-xorg.service       loaded active running ABRT Xorg log watcher
abrtd.service           loaded active running ABRT Automated Bug Reporting Tool
...
systemd-vconsole-setup.service loaded active exited  Setup Virtual Console
tog-pegasus.service     loaded active running OpenPegasus CIM Server

LOAD  = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB  = The low-level unit activation state, values depend on unit type.

46 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'
```

要列出所有安装的服务单元文件以确定它们是否启用，请输入：

```
~]$ systemctl list-unit-files --type service
UNIT FILE                STATE
abrt-ccpp.service        enabled
abrt-oops.service        enabled
abrt-vmcore.service      enabled
abrt-xorg.service        enabled
abrtd.service            enabled
...
wpa_supplicant.service   disabled
ypbind.service           disabled

208 unit files listed.
```

10.2.2. 显示服务状态

要显示与系统服务对应的服务单元的详细信息，请在 shell 提示符后输入以下内容：

```
systemctl status name.service
```

使用您要检查的服务单元的名称替换 name（例如： gdm）。这个命令显示所选服务单元的名称，后跟其简短描述、表 10.5 “可用服务单元信息” 中描述的一个或多个字段，如果由 root 用户执行，也是最新的日志条目。

表 10.5. 可用服务单元信息

项	描述
Loaded	是否载入了服务单元、到这个单元文件的绝对路径，以及是否启用该单位的信息。
Active	服务单元是否在运行的信息，后面有一个时间戳。
Main PID	对应系统服务的 PID 及其名称。
Status	相关系统服务的额外信息。
Process	有关相关进程的附加信息。
CGroup	有关相关控制组群（cgroups）的附加信息。

要只验证某个服务单元是否正在运行，运行以下命令：

```
systemctl is-active name.service
```

要确定某个服务单元是否启用，运行：

```
systemctl is-enabled name.service
```

请注意，如果指定的服务单元正在运行或已启用，则 systemctl is-active 和 systemctl is-enabled 的返回退出状态为 0。有关如何列出所有当前载入的服务单元的详情请参考 第 10.2.1 节 “列出服务”。

例 10.2. 显示服务状态

GNOME 显示管理器的服务单元名为 gdm.service。要确定这个服务单元的当前状态，在 shell 提示下键入以下内容：

```
~]# systemctl status gdm.service
gdm.service - GNOME Display Manager
  Loaded: loaded (/usr/lib/systemd/system/gdm.service; enabled)
  Active: active (running) since Thu 2013-10-17 17:31:23 CEST; 5min ago
 Main PID: 1029 (gdm)
  CGroup: /system.slice/gdm.service
      ├─1029 /usr/sbin/gdm
      ├─1037 /usr/libexec/gdm-simple-slave --display-id /org/gno...
      └─1047 /usr/bin/Xorg :0 -background none -verbose -auth /r...

Oct 17 17:31:23 localhost systemd[1]: Started GNOME Display Manager.
```

例 10.3. 在服务前按顺序显示服务

要确定在指定服务前调度什么服务启动，在 shell 提示下键入以下内容：

```
~]# systemctl list-dependencies --after gdm.service
gdm.service
├─dbus.socket
├─getty@tty1.service
├─livesys.service
├─plymouth-quit.service
├─system.slice
├─systemd-journald.socket
├─systemd-user-sessions.service
└─basic.target
[output truncated]
```

例 10.4. 在服务后显示被启动的服务

要确定在指定服务后调度的服务启动，在 shell 提示符后输入以下内容：

```
~]# systemctl list-dependencies --before gdm.service
gdm.service
├─dracut-shutdown.service
├─graphical.target
│ ├─systemd-readahead-done.service
│ ├─systemd-readahead-done.timer
│ └─systemd-update-utmp-runlevel.service
└─shutdown.target
  ├─systemd-reboot.service
  └─final.target
    └─systemd-reboot.service
```

10.2.3. 启动服务

要启动与系统服务对应的服务单元，以 root 用户身份在 shell 提示符后输入以下内容：

```
systemctl start name.service
```

使用您要启动的服务单元的名称替换 name（例如： gdm）。这个命令会在当前会话中启动所选服务单元。有关如何在引导时启用服务单元的详情请参考 第 10.2.6 节 “启用服务”。有关如何确定特定服务单元状态的详情请参考 第 10.2.2 节 “显示服务状态”。

例 10.5. 启动服务

Apache HTTP 服务器的服务单元名为 httpd.service。要激活这个服务单元并在当前会话中启动 httpd 守护进程，以 root 用户身份运行以下命令：

```
~]# systemctl start httpd.service
```

10.2.4. 停止服务

要停止与系统服务对应的服务单元，以 root 用户身份在 shell 提示符后输入以下内容：

```
systemctl stop name.service
```

使用您要停止的服务单元的名称替换 name（例如： bluetooth）。该命令将在当前会话中停止所选服务单元。有关如何禁用服务单元并阻止它在引导时启动的详情请参考 第 10.2.7 节 “禁用服务”。有关如何确定特定服务单元状态的详情请参考 第 10.2.2 节 “显示服务状态”。

例 10.6. 停止服务

bluetoothd 守护进程的服务单元名为 bluetooth.service。要取消激活这个服务单元并在当前会话中停止 bluetoothd 守护进程，以 root 用户身份运行以下命令：

```
~]# systemctl stop bluetooth.service
```

10.2.5. 重启服务

要重启与系统服务对应的服务单元，以 root 用户身份在 shell 提示符后输入以下内容：

```
systemctl restart name.service
```

使用您要重启的服务单元的名称替换 name（例如 httpd）。这个命令可在当前会话中停止所选服务单元，并立即重新启动。最重要的是，如果所选服务单元没有运行，这个命令也会启动它。要让 systemd 仅在相应服务已在运行时重启服务单元，以 root 用户身份运行以下命令：

```
systemctl try-restart name.service
```

某些系统服务还允许您在不中断其执行的情况下重新载入其配置。要做到这一点，以 root 用户身份输入：

```
systemctl reload name.service
```

请注意，不支持这个功能的系统服务会忽略这个命令。为方便起见，systemctl 命令还支持 reload-or-restart 和 reload-or-try-restart 命令来替代重启这些服务。有关如何确定特定服务单元状态的详情请参考 第 10.2.2 节 “显示服务状态”。

例 10.7. 重启服务

为了防止用户遇到不必要的错误消息或部分呈现的 Web 页面，Apache HTTP 服务器允许您编辑和重新加载其配置，而无需重新启动它并中断主动处理的请求。要做到这一点，以 root 根用户身份在 shell 提示符后输入以下内容：

```
~]# systemctl reload httpd.service
```

10.2.6. 启用服务

要配置与系统服务对应的服务单元，在引导时自动启动，以 root 用户身份在 shell 提示符后输入以下内容：

```
systemctl enable name.service
```

使用您要启用的服务单元的名称替换 name（例如 httpd）。该命令读取所选服务单元的 [Install] 部分，并在 /etc/systemd/system/ 目录和其子目录中创建到 /usr/lib/systemd/system/name.service 的符号链接。但是，这个命令不会重写已经存在的链接。如果要确保重新创建符号链接，以 root 用户身份使用以下命令：

```
systemctl reenable name.service
```

该命令禁用所选服务单元，并立即再次启用。有关如何确定某个服务单元是否已启用在引导时启动的详情请参考 第 10.2.2 节 “显示服务状态”。有关如何在当前会话中启动服务的详情请参考 第 10.2.3 节 “启动服务”。

例 10.8. 启用服务

要将 Apache HTTP 服务器配置为在引导时自动启动，以 root 用户身份运行以下命令：

```
~]# systemctl enable httpd.service
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to
/usr/lib/systemd/system/httpd.service.
```

10.2.7. 禁用服务

要防止与系统服务对应的服务单元在引导时自动启动，以 root 用户身份在 shell 提示符后输入以下内容：

```
systemctl disable name.service
```

使用您要禁用的服务单元的名称替换 name（例如： bluetooth）。该命令读取所选服务单元的 [Install] 部分，并从 /etc/systemd/system/ 目录及其子目录中删除到

/usr/lib/systemd/system/name.service 文件的符号链接。另外，您可以屏蔽所有服务单元，以防止手动启动或者由其他服务启动。要做到这一点，以 root 运行以下命令：

```
systemctl mask name.service
```

这个命令将 /etc/systemd/system/name.service 文件替换为 /dev/null 的符号链接，从而导致 systemd 无法访问实际的单元文件。要恢复这个动作并取消掩码一个服务单元，以 root 用户身份输入：

```
systemctl unmask name.service
```

有关如何确定某个服务单元是否已启用在引导时启动的详情请参考 第 10.2.2 节 “显示服务状态”。有关如何在当前会话中停止服务的详情请参考 第 10.2.4 节 “停止服务”。

例 10.9. 禁用服务

例 10.6 “停止服务” 演示如何在当前会话中停止 bluetooth.service 单元。要防止这个服务单元在引导时启动，以 root 用户身份在 shell 提示符后输入以下内容：

```
~]# systemctl disable bluetooth.service
Removed symlink /etc/systemd/system/bluetooth.target.wants/bluetooth.service.
Removed symlink /etc/systemd/system/dbus-org.bluez.service.
```

10.2.8. 启动冲突服务

在 Systemd 中，不同服务间会存在正或负的依赖关系。启动特定的服务可能需要启动一个或多个其他服务（正向依赖项）或者停止一个或多个服务（负依赖项）。

当您试图启动新服务时，systemd 会自动解析所有依赖项。请注意，这是在没有向用户发出显式通知的情况下完成的。如果您已经运行了服务，且您试图使用负依赖项启动另一个服务，则第一个服务会自动停止。

例如，如果您运行 postfix 服务，并且尝试启动 sendmail 服务，systemd 首先会自动停止 postfix，因为这两个服务彼此冲突且无法在同一个端口上运行。

10.3. 使用 SYSTEMD 目标

之前版本的 Red Hat Enterprise Linux（使用 SysV init 或 Upstart 发布）实施了一组代表特定操作模式的预定义运行级别。这些运行级别从 0 到 6，由系统管理员启用特定运行级别时要运行的系统服务选

择定义。在 Red Hat Enterprise Linux 7 中，运行级别的概念已被 systemd 目标替代。

systemd 目标由目标单元表示。目标单元以 .target 文件扩展名结尾，它们的唯一用途是通过依赖项链将其他 systemd 单元分组在一起。例如，用于启动图形会话的 graphical.target 单元将启动系统服务，如 GNOME 显示管理器(gdm.service)或帐户服务(accounts-daemon.service)，还激活 multi-user.target 单元。同样，multi -user.target 单元会启动其他基本系统服务，如 NetworkManager(NetworkManager.service)或 D-Bus(dbus.service)，并激活另一个名为 basic.target 的目标单元。

Red Hat Enterprise Linux 7 发布有多个预定义目标，它们与之前系统版本中的标准运行级别集类似。出于兼容性的原因，它还为这些目标提供了别名，可将其直接映射到 SysV 运行级别。表 10.6 “SysV 运行级别与 systemd 目标比较” 提供 SysV 运行级别的完整列表及其相应的 systemd 目标。

表 10.6. SysV 运行级别与 systemd 目标比较

运行级别	目标单元	描述
0	runlevel0.target, poweroff.target	关闭系统。
1	runlevel1.target, rescue.target	设置救援 shell。
2	runlevel2.target, multi-user.target	设置一个非图形化的多用户系统。
3	runlevel3.target, multi-user.target	设置一个非图形化的多用户系统。
4	runlevel4.target, multi-user.target	设置一个非图形化的多用户系统。
5	runlevel5.target, graphical.target	设置图形化多用户系统。
6	runlevel6.target, reboot.target	关闭并重启系统。

要查看、更改或配置 systemd 目标，请使用 systemctl 工具，如 表 10.7 “SysV init 命令与 systemctl 的比较” 和以下部分所述。运行级别 和 telinit 命令仍可在系统中使用，并可按预期工作，但只出于兼容性原因被包括在内，因此应该尽量避免使用。

表 10.7. SysV init 命令与 systemctl 的比较

旧命令	新命令	描述
runlevel	**systemctl list-units --type target**	列出当前载入的目标单元。
telinit *runlevel*	**systemctl isolate *name*.target**	更改当前目标。

10.3.1. 查看默认目标

要确定默认使用哪个目标单元，运行以下命令：

```
systemctl get-default
```

这个命令解析位于 /etc/systemd/system/default.target 的符号链接并显示结果。有关如何更改默认目标的详情请参考 第 10.3.3 节 “更改默认目标”。有关如何列出所有当前载入的目标单元的详情请参考 第 10.3.2 节 “查看当前目标”。

例 10.10. 查看默认目标

要显示默认目标单元，键入：

```
~]$ systemctl get-default
graphical.target
```

10.3.2. 查看当前目标

要列出所有当前载入的目标单元，在 shell 提示下键入以下命令：

```
systemctl list-units --type target
```

对于每个目标单元，这个命令会显示其全名(UNIT)，后跟一个备注，该单元是否已加载(LOAD)、其高级(ACTIVE)和低级(SUB)单元激活状态，以及简短描述(DESCRIPTION)。

默认情况下，systemctl list-units 命令只显示活跃的单位。如果您想列出所有载入的单元，无论它们的状态如何，请使用 --all 或 -a 选项。

```
systemctl list-units --type target --all
```

有关如何显示默认目标的详情，请查看 第 10.3.1 节 “查看默认目标”。有关如何更改当前目标的详情请参考 第 10.3.4 节 “更改当前目标”。

例 10.11. 查看当前目标

要列出所有当前载入的目标单元，请运行以下命令：

```
~]$ systemctl list-units --type target
UNIT           LOAD  ACTIVE SUB  DESCRIPTION
basic.target     loaded active active Basic System
cryptsetup.target   loaded active active Encrypted Volumes
getty.target     loaded active active Login Prompts
graphical.target   loaded active active Graphical Interface
local-fs-pre.target  loaded active active Local File Systems (Pre)
local-fs.target    loaded active active Local File Systems
multi-user.target   loaded active active Multi-User System
network.target    loaded active active Network
paths.target     loaded active active Paths
remote-fs.target   loaded active active Remote File Systems
sockets.target    loaded active active Sockets
sound.target     loaded active active Sound Card
spice-vdagentd.target loaded active active Agent daemon for Spice guests
swap.target      loaded active active Swap
sysinit.target    loaded active active System Initialization
time-sync.target   loaded active active System Time Synchronized
timers.target     loaded active active Timers

LOAD  = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB  = The low-level unit activation state, values depend on unit type.

17 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
```

10.3.3. 更改默认目标

要将系统配置为默认使用不同的目标单元，以 root 用户身份在 shell 提示符后输入以下内容：

```
systemctl set-default name.target
```

将 name 替换为您要默认使用的目标单元的名称（例如： multi-user）。这个命令将 /etc/systemd/system/default.target 文件替换为指向 /usr/lib/systemd/system/name.target 的 符号链接，其中 name 是您要使用的目标单元的名称。有关如何更改当前目标的详情请参考 第 10.3.4 节 “更改当前目标”。有关如何列出所有当前载入的目标单元的详情请参考 第 10.3.2 节 “查看当前目标”。

例 10.12. 更改默认目标

要将系统配置为默认使用 multi-user.target 单元，以 root 用户身份运行以下命令：

```
~]# systemctl set-default multi-user.target
rm '/etc/systemd/system/default.target'
ln -s '/usr/lib/systemd/system/multi-user.target' '/etc/systemd/system/default.target'
```

10.3.4. 更改当前目标

要切换到当前会话中的不同目标单元，以 root 用户身份在 shell 提示符后输入以下内容：

```
systemctl isolate name.target
```

使用您要使用的目标单元的名称替换 name（例如： multi-user）。这个命令启动名为 name 的目标单元 以及所有依赖的单元，并立即停止所有其它单元。有关如何更改默认目标的详情请参考 第 10.3.3 节 “更改默认目标”。有关如何列出所有当前载入的目标单元的详情请参考 第 10.3.2 节 “查看当前目标”。

例 10.13. 更改当前目标

要关闭图形用户界面并改为当前会话中的 multi-user.target 单元，以 root 用户身份运行以下命令：

```
~]# systemctl isolate multi-user.target
```

10.3.5. 进入救援模式

救援模式提供了一个方便的单用户环境，它可让您在无法完成常规引导过程时修复您的系统。在救援模式中，系统会尝试挂载所有本地文件系统并启动一些重要的系统服务，但不激活网络接口或者同时允许更多的用户登录到该系统。在 Red Hat Enterprise Linux 7 中，救援模式等同于单用户模式，需要 root 密码。

要改变当前目标并在当前会话中进入救援模式，以 root 用户身份在 shell 提示符后输入以下内容：

```
systemctl rescue
```

这个命令和 systemctl isolate rescue.target 类似，但它也会向所有当前登录到系统的用户发送一个

信息信息。要防止 systemd 发送这个信息，使用 --no-wall 命令行选项运行这个命令：

```
systemctl --no-wall rescue
```

有关如何进入紧急模式的详情请参考 第 10.3.6 节 “进入紧急模式”。

例 10.14. 进入救援模式

要在当前会话中进入救援模式，以 root 以用户身份运行以下命令：

```
~]# systemctl rescue
Broadcast message from root@localhost on pts/0 (Fri 2013-10-25 18:23:15 CEST):
The system is going down to rescue mode NOW!
```

10.3.6. 进入紧急模式

紧急模式 提供最小的环境，并可在系统无法进入救援模式的情况下修复您的系统。在紧急模式中，系统仅挂载用于读取的 root 文件系统，不会尝试挂载任何其他本地文件系统，不激活网络接口，并且仅启动几个必要的服务。在 Red Hat Enterprise Linux 7 中，紧急模式需要 root 密码。

要改变当前目标并进入紧急模式，以 root 用户身份在 shell 提示符后输入以下内容：

```
systemctl emergency
```

这个命令和 systemctl isolate emergency.target 类似,但它也会向所有当前登录到系统的用户发送一个信息信息。要防止 systemd 发送这个信息，使用 --no-wall 命令行选项运行这个命令：

```
systemctl --no-wall emergency
```

有关如何进入救援模式的详情请参考 第 10.3.5 节 “进入救援模式”。

例 10.15. 进入紧急模式

要进入紧急模式而不向目前登录到该系统的所有用户发送信息，以 root 用户身份运行以下命令：

```
~]# systemctl --no-wall emergency
```

10.4. 关闭、托管和占用系统

在 Red Hat Enterprise Linux 7 中，systemctl 工具替换了之前版本的 Red Hat Enterprise Linux 系统中使用的很多电源管理命令。出于兼容性的原因，表 10.8 “Power Management 命令与 systemctl 的比较” 中列出的命令仍然可用，但建议您尽可能使用 systemctl。

表 10.8. Power Management 命令与 systemctl 的比较

旧命令	新命令	描述
halt	**systemctl halt**	关闭系统。
poweroff	**systemctl poweroff**	关闭系统。
reboot	**systemctl reboot**	重启该系统。
pm-suspend	**systemctl suspend**	挂起系统。
pm-hibernate	**systemctl hibernate**	休眠系统。
pm-suspend-hybrid	**systemctl hybrid-sleep**	休眠并挂起系统。

10.4.1. 关闭系统

systemctl 实用程序提供关闭系统的命令，但也支持传统的 **shutdown** 命令。虽然 **shutdown** 命令会调用 **systemctl** 实用程序执行关闭，但它支持使用一个时间参数。这对预定的维护特别有用，用户可以有足够的时间响应系统已经调度关闭的警告。取消关闭的选项也很优越。

使用 systemctl 命令

要关闭系统并关闭机器，以 root 用户身份在 shell 提示符后输入以下内容：

```
systemctl poweroff
```

要在不关闭机器的情况下关闭和停止系统，以 root 用户运行以下命令：

```
systemctl halt
```

默认情况下，运行其中任何一个命令会导致 systemd 向所有当前登录该系统的用户发送信息。要防止

systemd 发送这个信息，使用 --no-wall 命令行选项运行所选命令，例如：

```
systemctl --no-wall poweroff
```

使用 shutdown 命令

要关闭该系统并在一定时间关闭机器，以 root 用户身份使用以下格式的命令：

```
shutdown --poweroff hh:mm
```

这里的 hh:mm 是 24 小时时钟格式的时间。/run/nologin 文件会在系统关闭前 5 分钟创建，以防止新的登录。当使用时间参数时，可以选择将一个信息（wall message）附加至命令中。

要在一段延迟后关闭和停止系统，在不关闭机器的情况下，以 root 用户身份使用以下格式的命令：

```
shutdown --halt +m
```

其中 +m 是延迟时间（以分钟为单位）。now 等同于 +0。

等待被关闭的系统可由 root 用户取消，如下所示：

```
shutdown -c
```

更多命令选项请查看 shutdown(8) 手册页。

10.4.2. 重启系统

要重启该系统，以 root 运行以下命令：

```
systemctl reboot
```

默认情况下，这个命令可让 systemd 向所有当前登录该系统的用户发送信息。要防止 systemd 发送这个信息，使用 --no-wall 命令行选项运行这个命令：

```
systemctl --no-wall reboot
```

10.4.3. 挂起系统

要挂起系统，以 root 用户身份在 shell 提示符后输入以下内容：

```
systemctl suspend
```

该命令在 RAM 中保存系统状态，除了 RAM 模块外，关闭机器中的大多数设备。当您重新打开机器时，系统会从内存中恢复其状态，而无需再次引导。由于系统状态保存在 RAM 中而不是保存在硬盘中，将系统从挂起模式恢复比从休眠状态恢复要快得多，但因此，暂停的系统状态也容易出现电源中断。

有关如何休眠系统的详情请参考 第 10.4.4 节 “休眠系统”。

10.4.4. 休眠系统

要休眠系统，以 root 用户身份在 shell 提示符后输入以下内容：

```
systemctl hibernate
```

该命令在硬盘驱动器中保存系统状态，并断开机器电源。当您重新打开机器时，系统会从保存的数据中恢复其状态，而无需再次引导。由于系统状态保存在硬盘中，而不是保存在 RAM 中，因此计算机不必维护 RAM 模块的电力，但因此，将系统从休眠模式恢复比将其恢复为暂停模式要慢得多。

要让系统休眠并暂停系统，以 root 用户身份运行以下命令：

```
systemctl hybrid-sleep
```

有关如何挂起该系统的详情请参考 第 10.4.3 节 “挂起系统”。

10.5. 控制远程机器上的 SYSTEMD

除了在本地控制 systemd 系统和服务管理器外，system ctl 还允许您通过 SSH 协议与远程机器上运行的 systemd 交互。如果远程机器上的 sshd 服务正在运行，您可以使用 --host 或 -H 命令行选项运行 systemctl 命令来连接此机器：

```
systemctl --host user_name@host_name command
```

使用远程用户名称替换 user _name，host_name 替换为计算机的主机名，而 command 替换为上述任何 systemctl 命令。请注意，远程计算机必须配置为允许选定的用户通过 SSH 协议进行远程访问。有关如何配置 SSH 服务器的详情请参考 第 12 章 OpenSSH。

例 10.16. 远程管理

以 root 用户身份登录名为 server-01.example.com 的远程机器， 并确定 httpd.service 单元的当前状态，在 shell 提示符后输入以下内容：

```
~]$ systemctl -H root@server-01.example.com status httpd.service
>>>>>>> systemd unit files -- update
root@server-01.example.com's password:
httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled)
  Active: active (running) since Fri 2013-11-01 13:58:56 CET; 2h 48min ago
 Main PID: 649
  Status: "Total requests: 0; Current requests/sec: 0; Current traffic:  0 B/sec"
  CGroup: /system.slice/httpd.service
```

10.6. 创建和修改 SYSTEMD 单元文件

单元文件包含描述这个单元并定义其行为的配置指令。几个 systemctl 命令可在后台使用单元文件。要进行更细的调整，系统管理员必须手动编辑或创建单元文件。表 10.2 “systemd 单元文件位置” 列出存储在系统中单元文件的三个主要目录，/etc/systemd/system/ 目录为系统管理员创建或自定义的单元文件保留。

单元文件名的格式如下：

```
unit_name.type_extension
```

此处，unit_name 代表单元名称，type_extension 代表单元类型，请参阅 表 10.1 “可用的 systemd 单元类型” 了解单元类型的完整列表。例如，系统通常会有 sshd.service 和 sshd.socket 单元。

可通过一个目录来补充单元文件，以了解额外的配置文件。例如，要将自定义配置选项添加到 sshd.service，请创建 sshd.service.d/custom.conf 文件，并在其中插入其他指令：有关配置目录的详情请参考 第 10.6.4 节 “修改现有单元文件”。

另外，sshd.service.wants/ 和 sshd.service.requires/ 目录可以被创建。这些目录包含到 sshd 服务依赖的单元文件的符号链接。符号链接会在安装过程中根据 [Install] 单元文件选项（请参阅 表 10.11 “重

要 [Install] 部分选项”）或者在运行时根据 [Unit] 选项自动创建（请参阅 表 10.9 “重要 [Unit] 部分选项”）。也可以手动创建这些目录和符号链接。

可以使用名为 单元指定符 - 通配符字符串（在加载单元文件时动态替换为单元参数）来设置许多单元文件选项。这可创建通用单元文件，用作生成实例化单元的模板。详情请查看 第 10.6.5 节 “使用 Instantiation 单元”。

10.6.1. 了解单元文件结构

单元文件通常由三个部分组成：

- [Unit] - 包含不依赖于这个单元类型的通用选项。这些选项提供单元描述，指定单元的行为，并将依赖项设置为其他单元。有关最常用 [Unit] 选项的列表请参考 表 10.9 “重要 [Unit] 部分选项”。
- [单元类型] - 如果单元具有特定于类型的指令，则这些指令分组在以单元类型命名的部分中。例如，服务单元文件包含 [Service] 部分，有关最常用的 [Service] 选项，请参阅 表 10.10 “重要 [Service] 部分选项”。
- [install] - 包含 systemctl enable 和 disable 命令使用的单元安装信息，请参阅 表 10.11 “重要 [Install] 部分选项” 了解 [Install] 选项列表。

表 10.9. 重要 [Unit] 部分选项

选项[a] 部分，请参阅 systemd.unit(5) 手册页。]	描述
描述	对单元进行有意义的描述。这个文本显示在 **systemctl status** 命令的输出中。
Documentation	提供单元参考文档的 URI 列表。
后[b]	定义启动单位的顺序。这个单元仅在 **After** 中指定的单元处于活跃状态后才启动。与 **Requires** 不同，**After** 不会显式激活指定的单元。**Before** 选项与 **After** 的功能相反。
Requires	配置其它单元上的依赖关系。**Requires** 中列出的单元与单元一同被激活。如果任何需要的单元无法启动，则该单位就不会被激活。

选项[a] 部分，请参阅 systemd.unit(5) 手册页。]	描述
期望	配置比 **Requires** 更弱的依赖项。如果列出的单元没有成功启动，它对单元激活不会有影响。这是建立自定义单元依赖项的建议方法。
Conflicts	配置负的依赖关系，与 **Requires** 相反。

[a] 使用 [Unit] 配置选项的完整列表

[b] 在大多数情况下，只需要**After** 和 **Before** 单元文件选项设置顺序依赖关系就足够了。如果还使用 **Wants**（推荐）或 **Requires**设置了需要的依赖关系，仍需要指定依赖关系顺序。这是因为排序和要求依赖关系可以独立地工作。

表 10.10. 重要 [Service] 部分选项

选项[a] 部分，请参阅 systemd.service(5) 手册页。]	描述
Type	配置单元进程启动类型，它会影响 **ExecStart 的功能** 和相关选项。其中之一： * **simple** - 默认值。使用 **ExecStart** 启动的进程是该服务的主要进程。 * **forking** - 使用 **ExecStart** 启动的进程生成一个子进程，成为服务的主进程。父进程在启动完成后会退出。 * **oneshot** – 这个类型与 **simple** 类似，但在启动相应单位前会退出。 * **dbus** - 这个类型与 **simple** 类似，但仅在主进程获得 D-Bus 名称后启动。 * **notify** - 此类型与 **simple** 类似，但只有在通过 sd_notify（）函数发送通知消息后才启动相应单元。 * **idle** - 与 **simple** 类似，服务二进制文件的实际执行会延迟到所有作业完成前，这样可以避免将状态输出与服务的 shell 输出混合。
ExecStart	指定在启动该单元时要执行的命令或脚本。**ExecStartPre** 和 **ExecStartPost** 指定在 **ExecStartPtart** 之前和之后要执行的自定义命令。**Type=oneshot** 启用指定可按顺序执行的多个自定义命令。
ExecStop	指定在该单元停止时要执行的命令或脚本。
ExecReload	指定重新载入该单元时要执行的命令或脚本。

选项[a] 部分，请参阅 systemd.service(5) 手册页。]	描述
Restart	启用此选项后，服务会在进程退出后重新启动，但 **systemctl** 命令的干净停止除外。
RemainAfterExit	如果设置为 True，即使所有进程都退出，该服务也被视为活动状态。默认值为 False。这个选项在配置了 **Type=oneshot** 时特别有用。

[a] 在 [Service] 中可配置选项的完整列表

表 10.11. 重要 [Install] 部分选项

选项[a] 部分，请参阅 systemd.unit(5) 手册页。]	描述
Alias	为这个单元提供空格分开的额外名称列表。除 **systemctl enable** 以外，多数**systemctl** 命令可使用别名而不是实际的单元名称。
RequiredBy	依赖于这个单元的单元列表。当启用此单元时，在 **RequiredBy** 中列出的单元会获得对这个单元的一个 **Require** 依赖项。
WantedBy	依赖于这个单元的单位列表。当启用这个单元时，在 **WantedBy** 中列出的单元会得到一个 **Want** 依赖项。
Also	指定要随这个单元一起安装或卸载的单元列表。
DefaultInstance	仅限于实例化单元，这个选项指定启用单位的默认实例。请查看 第 10.6.5 节 “使用 Instantiation 单元”

[a] 在 [Install）中可配置的完整选项列表

可用来微调单元配置的完整选项，例 10.17 “postfix.service 单元文件” 显示系统中安装的服务单元示例。另外，可以定义单元文件选项以支持动态创建单元文件，如 第 10.6.5 节 “使用 Instantiation 单元” 所述。

> ***例 10.17. postfix.service 单元文件***
>
> ***以下是 /usr/lib/systemd/system/postfix.service 单元文件，当前由 postfix 软件包提供：***
>
> ```
> [Unit]
> ```

```
Description=Postfix Mail Transport Agent
After=syslog.target network.target
Conflicts=sendmail.service exim.service

[Service]
Type=forking
PIDFile=/var/spool/postfix/pid/master.pid
EnvironmentFile=-/etc/sysconfig/network
ExecStartPre=-/usr/libexec/postfix/aliasesdb
ExecStartPre=-/usr/libexec/postfix/chroot-update
ExecStart=/usr/sbin/postfix start
ExecReload=/usr/sbin/postfix reload
ExecStop=/usr/sbin/postfix stop

[Install]
WantedBy=multi-user.target
```

[Unit] 部分描述服务，指定排序依赖关系以及冲突的单元。在 [Service] 中，指定一系列自定义脚本，在单元激活、停止和重新加载期间执行。EnvironmentFile 指向服务环境变量的定义位置，PIDFile 为服务的主进程指定稳定的 PID。最后，[Install] 部分列出了依赖于该服务的单元。

10.6.2. 创建自定义单元文件

从头开始创建单元文件有几个用例：您可以运行自定义守护进程、创建一些现有服务的第二个实例（如 例 10.19 “创建 sshd 服务第二个实例”）或导入 SysV 初始化脚本（更多在 第 10.6.3 节 “将 SysV Init 脚本转换为单元文件”中）。另一方面，如果您只修改或扩展现有单元的行为，请使用 第 10.6.4 节 “修改现有单元文件” 中的说明。以下流程描述了创建自定义服务的一般过程：

1. *使用自定义服务准备可执行文件。这可以是自定义创建的脚本，也可以是软件供应商提供的可执行文件。如果需要，准备 PID 文件来保存自定义服务主要进程的恒定 PID。也可以包含环境文件来存储该服务的 shell 变量。确保源脚本可执行（通过执行 chmod a+x）且不进行交互。*

2. *在 /etc/systemd/system/ 目录中创建一个单元文件，并确定它有正确的文件权限。以 root 用户身份执行：*

   ```
   touch /etc/systemd/system/name.service
   chmod 664 /etc/systemd/system/name.service
   ```

   使用要创建的服务的名称替换 name。请注意，该文件不需要可执行。

3. *打开上一步中创建的 name.service 文件并添加服务配置选项。根据您要创建的服务类型，可以使用各种选项，请参阅 第 10.6.1 节 “了解单元文件结构”。以下是网络相关服务的单元配置*

示例：

```
[Unit]
Description=service_description
After=network.target

[Service]
ExecStart=path_to_executable
Type=forking
PIDFile=path_to_pidfile

[Install]
WantedBy=default.target
```

其中：

- service_description 是一个说明性描述，在 journal 日志文件和 systemctl status 命令的输出中显示。
- After 设置可确保仅在网络运行时启动该服务。添加以空格分隔的其他相关服务或目标列表。
- path_to_executable 代表到实际可执行服务的路径。
- type=forking 用于生成 fork 系统调用的守护进程。该服务的主要进程使用 path_to_pidfile 中指定的 PID 创建。在 表 10.10 “重要 [Service] 部分选项” 中查找其他启动类型。
- WantedBy 指出该服务应该启动的目标。将这些目标视为旧运行级别概念的替代，详情请查看 第 10.3 节 “使用 systemd 目标”。

4. 以 root 用户身份执行以下命令来通知 systemd 是否存在新 name.service 文件：

```
systemctl daemon-reload
systemctl start name.service
```

警告

在创建新的单元文件或修改现有单元文件后，始终运行 systemctl daemon-reload 命令。否则，systemctl start 或 systemctl enable 命令可能会因为 systemd 状态和磁盘上的实际服务单元文件不匹配而失败。

name.service 单元现在可以与任何其他系统服务一样使用 第 10.2 节 “管理系统服务” 所述的命令进行管理。

例 10.18. 创建 emacs.service 文件

当使用 Emacs 文本编辑器时，在后台运行它通常更迅速且方便，而不是在每次编辑文件时启动新实例。以下步骤演示了如何为 Emacs 创建单元文件，这样它就可以像服务一样被处理。

1. 在 /etc/systemd/system/ 目录中创建一个单元文件，并确定它具有正确的文件权限。以 root 用户身份执行：

```
~]# touch /etc/systemd/system/emacs.service
~]# chmod 664 /etc/systemd/system/emacs.service
```

2. 在文件中添加以下内容：

```
[Unit]
Description=Emacs: the extensible, self-documenting text editor

[Service]
Type=forking
ExecStart=/usr/bin/emacs --daemon
ExecStop=/usr/bin/emacsclient --eval "(kill-emacs)"
Environment=SSH_AUTH_SOCK=%t/keyring/ssh
Restart=always

[Install]
WantedBy=default.target
```

使用上述配置时，/usr/bin/emacs 可执行文件在服务启动时以守护进程模式启动。SSH_AUTH_SOCK 环境变量使用代表运行时目录的 "%t" 单元指定符进行设置。如果意外退出，服务还会重启 emacs 进程。

3. 执行以下命令重新载入配置并启动自定义服务：

```
~]# systemctl daemon-reload
~]# systemctl start emacs.service
```

因为编辑器现在注册为 systemd 服务，您可以使用所有标准 systemctl 命令。例如：运行 systemctl status emacs 来显示编辑器的状态，或者 systemctl enable emacs 以便在系统引导时自动启动编辑器。

例 10.19. 创建 sshd 服务第二个实例

系统管理员通常需要配置并运行多个服务实例。这可以通过创建原始服务配置文件的副本并修改某些参数来避免与服务的主实例冲突。以下流程演示了如何创建 sshd 服务第二个实例：

1. 创建第二个守护进程将使用的 sshd_config 文件副本：

```
~]# cp /etc/ssh/sshd{,-second}_config
```

2. 编辑上一步中创建的 sshd-second_config 文件，为第二个守护进程分配不同的端口号和 PID 文件：

```
Port 22220
PidFile /var/run/sshd-second.pid
```

有关 Port 和 PidFile 选项的详情，请查看 sshd_config(5)手册页。请确定您选择的端口没有被其他服务使用。在运行该服务前，PID 文件不一定存在，它会在服务启动时自动生成。

3. 为 sshd 服务创建 systemd 单元文件副本：

```
~]# cp /usr/lib/systemd/system/sshd.service /etc/systemd/system/sshd-second.service
```

4. 按如下方式更改上一步中创建的 sshd-second.service:

   a. 修改 Description 选项：

   -

```
Description=OpenSSH server second instance daemon
```

b. 将 sshd.service 添加到 After 选项中指定的服务，因此第二实例仅在第一个实例启动后启动：

```
After=syslog.target network.target auditd.service sshd.service
```

c. sshd 的第一个实例包括密钥生成，因此删除ExecStartPre=/usr/sbin/sshd-keygen 行。

d. 为 sshd 命令添加 -f /etc/ssh/sshd-second_config 参数，以便使用其它配置文件：

```
ExecStart=/usr/sbin/sshd -D -f /etc/ssh/sshd-second_config $OPTIONS
```

e. 在进行以上修改后，sshd-second.service 应该如下所示：

```
[Unit]
Description=OpenSSH server second instance daemon
After=syslog.target network.target auditd.service sshd.service

[Service]
EnvironmentFile=/etc/sysconfig/sshd
ExecStart=/usr/sbin/sshd -D -f /etc/ssh/sshd-second_config $OPTIONS
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartSec=42s

[Install]
WantedBy=multi-user.target
```

5. 如果使用 SELinux，请将第二个 sshd 实例的端口添加到 SSH 端口中，否则 sshd 的第二个实例将被拒绝绑定到端口：

```
~]# semanage port -a -t ssh_port_t -p tcp 22220
```

6. 启用 sshd-second.service，以便在引导时自动启动：

```
~]# systemctl enable sshd-second.service
```

使用 systemctl status 命令验证 sshd-second.service 是否在运行。另外，通过连接到该服务来验证是否正确启用了端口：

```
~]$ ssh -p 22220 user@server
```

如果使用防火墙，请确定正确配置了防火墙以便允许到第二个 sshd 实例的连接。

要了解如何正确地选择自定义单元文件排序和依赖项的目标，请查看以下文章。

- 如何编写强制必须启动特定服务的服务单元文件
- 如何决定 systemd 服务单元定义应具有哪些依赖项

如需包含单元文件中排序和依赖项所触发的一些真实示例的其他信息，请参见以下文章：是否存在有关编写单元文件的任何有用信息？

如果要为 systemd 启动的服务设置限制，请参阅红帽知识库文章 如何在 RHEL 7 和 systemd 中为服务设置限制。这些限制需要在服务的单元文件中设置。请注意，systemd 忽略 /etc/security/limits.conf 和 /etc/security/limits.d/*.conf 配置文件中设定的限制。这些文件中定义的限制在启动登录会话时由 PAM 设置，但 systemd 启动的守护进程不使用 PAM 登录会话。

10.6.3. 将 SysV Init 脚本转换为单元文件

在花费时间将 SysV 初始化脚本转换为单元文件之前，请确保在别处尚未执行转换。Red Hat Enterprise Linux 7 中安装的所有核心服务都有默认的单元文件，许多第三方软件包也是如此。

将初始化脚本转换成单元文件需要分析脚本并从中提取所需信息。根据这个数据，您可以创建一个单元文件，如 第 10.6.2 节 “创建自定义单元文件” 所述。因为初始化脚本可能会有很大差异，具体取决于服务类型，因此您可能需要使用比本章中介绍的更多配置选项进行转换。请注意，systemd 单元不再支持 init 脚本提供的某些级别的自定义，请参阅 第 10.1.2 节 “兼容性更改”。

脚本标题中提供了转换所需的大部分信息。以下示例显示了在 Red Hat Enterprise Linux 6 中启动 postfix 服务初始化脚本的打开部分：

```
#!/bin/bash
#
```

```
# postfix   Postfix Mail Transfer Agent
#
# chkconfig: 2345 80 30
# description: Postfix is a Mail Transport Agent, which is the program \
#       that moves mail from one machine to another.
# processname: master
# pidfile: /var/spool/postfix/pid/master.pid
# config: /etc/postfix/main.cf
# config: /etc/postfix/master.cf

### BEGIN INIT INFO
# Provides: postfix MTA
# Required-Start: $local_fs $network $remote_fs
# Required-Stop: $local_fs $network $remote_fs
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: start and stop postfix
# Description: Postfix is a Mail Transport Agent, which is the program that
#       moves mail from one machine to another.
### END INIT INFO
```

在上例中，只有以 # chkconfig 和 # description 开头的行是强制的，因此您可能不会在不同的 init 文件中找到其他行。# BEGIN INIT INFO 和# END INIT INFO 行间包含的文本称为 Linux Standard Base(LSB)标头。如果指定，LSB 标头包含定义服务描述、依赖项和默认运行级别的指令。下面是一个分析任务概述，旨在收集新单元文件所需的数据。postfix init 脚本被用作示例，请参阅 例 10.17 “postfix.service 单元文件” 中生成的 postfix 单元文件。

查找服务描述

在以 #description 开头的行中找到有关脚本的描述性信息。将此描述与单元文件的 [Unit] 部分中的 Description 选项中的服务名称一同使用。LSB 标头可能会在 #Short-Description 和 #Description 行中包含类似的数据。

查找服务依赖项

LSB 标头可能包含一些在服务间组成相依性指令。大多数可以转换到 systemd 单元选项，请参阅 表 10.12 “LSB 标头中的依赖项选项”

表 10.12. LSB 标头中的依赖项选项

LSB 选项	描述	单元文件的对等
Provides	指定服务的引导工具名称，可在其他初始化脚本中引用（使用"$"前缀）。因为单元文件根据文件名指向其他单元，所以不再需要这个操作。	-

LSB 选项	描述	单元文件的对等
Required-Start	包含所需服务的引导工具名称。这被转换为排序依赖关系，引导工具名称替换为相应服务或所属服务的单元文件名。例如，如果是 **postfix**，$network 上的 Required-Start 依赖关系被转换为 network.target 上的 After 依赖关系。	**After**, **Before**
Should-Start	比 Required-Start 更弱的依赖项。Should-Start 依赖项失败不会影响服务的启动。	**After**, **Before**
required-Stop, **Should-Stop**	组成负依赖关系。	**Conflicts**

查找服务的默认目标

以 #chkconfig 开头的行包含三个数字值。最重要的是第一个代表启动该服务的默认运行级别的数字。使用 表 10.6 “SysV 运行级别与 systemd 目标比较” 将这些运行级别映射到对等的 systemd 目标。然后，在单元文件的 [Install] 部分中列出这些目标。例如，postfix 以前 是在 运行级别 2、3、4 和 5 中启动的，它们转换为 Red Hat Enterprise Linux 7 中的 multi-user.target 和 graphical.target。请注意，graphical.target 依赖于 multiuser.target，因此不需要同时指定，如 例 10.17 “postfix.service 单元文件” 中所示。您可能会在 LSB 标头的 #Default-Start 和 #Default-Stop 行中找到默认和禁止运行级别的信息。

#chkconfig 行中指定的另外两个值代表初始化脚本的启动和关闭优先级。如果 systemd 加载初始化脚本，则对这些值进行解释，但没有等效的单元文件。

查找服务使用的文件

初始化脚本需要从专用目录中载入功能库，并允许导入配置、环境和 PID 文件。环境变量在初始化脚本标头中以 #config 开始的行上指定，该行转换为 EnvironmentFile 单元文件选项。在 #pidfile init 脚本行中指定的 PID 文件通过 PIDFile 选项导入到单元文件中。

未包含在初始化脚本标头中的关键信息是该服务可执行文件的路径，以及该服务可能需要的一些其他文件。在之前的 Red Hat Enterprise Linux 版本中，init 脚本使用 Bash case 语句来定义服务对默认操作的行为，如 start、stop 或 restart，以及自定义定义的操作。以下摘录自 postfix 初始化脚本显示了要在 service 启动时执行的代码块：

```
conf_check() {
  [ -x /usr/sbin/postfix ] || exit 5
  [ -d /etc/postfix ] || exit 6
  [ -d /var/spool/postfix ] || exit 5
}
```

```
make_aliasesdb() {
 if [ "$(/usr/sbin/postconf -h alias_database)" == "hash:/etc/aliases" ]
 then
  # /etc/aliases.db might be used by other MTA, make sure nothing
  # has touched it since our last newaliases call
  [ /etc/aliases -nt /etc/aliases.db ] ||
   [ "$ALIASESDB_STAMP" -nt /etc/aliases.db ] ||
   [ "$ALIASESDB_STAMP" -ot /etc/aliases.db ] || return
  /usr/bin/newaliases
  touch -r /etc/aliases.db "$ALIASESDB_STAMP"
 else
  /usr/bin/newaliases
 fi
}

start() {
 [ "$EUID" != "0" ] && exit 4
 # Check that networking is up.
 [ ${NETWORKING} = "no" ] && exit 1
 conf_check
 # Start daemons.
 echo -n $"Starting postfix: "
 make_aliasesdb >/dev/null 2>&1
 [ -x $CHROOT_UPDATE ] && $CHROOT_UPDATE
 /usr/sbin/postfix start 2>/dev/null 1>&2 && success || failure $"$prog start"
 RETVAL=$?
 [ $RETVAL -eq 0 ] && touch $lockfile
    echo
 return $RETVAL
}
```

init 脚本的可扩展性允许指定从 start （）函数块调用的两个自定义功能：conf _check （）和 make_aliasesdb ( )。然后,上面的代码中提到几个外部文件和目录：主服务可执行文件 /usr/sbin/postfix、/etc/postfix/ 和 /var/spool/postfix/ 配置目录，以及 /usr/sbin/postconf/ 目录。

systemd 仅支持预定义的操作，但启用了通过 ExecStart、ExecStartPre、ExecStart Post、ExecStartPost 和Exec Reload 选项执行自定义可执行文件。如果红帽企业 Linux 7 上有 postfix，则 service start 将执行 /usr/sbin/postfix 以及支持脚本。请参考 例 10.17 “postfix.service 单元文件” 上的 postfix 单元文件。

转换复杂的 init 脚本需要了解脚本中的每个语句的用途。其中一些语句特定于操作系统版本，因此您不需要转换它们。另一方面，新环境中可能需要进行一些调整，无论是在单元文件还是服务可执行文件中，还是在支持文件中。

10.6.4. 修改现有单元文件

在系统中安装的服务会附带保存在 /usr/lib/systemd/system/ 目录中的默认单元文件。系统管理员不应该直接修改这些文件，因此任何自定义都必须仅限于 /etc/systemd/system/ 目录中的配置文件。根据

所需更改的程度，选择以下方法之一：

- 在 /etc/systemd/system/unit.d/ 中创建一个附加配置文件的目录。我们推荐在大多数用例中使用这个方法。它启用了额外的功能来扩展默认配置，同时仍然引用原始的单元文件。因此，软件包升级引入的默认单元的更改会被自动应用。如需更多信息，请参阅 “扩展默认单元配置”一节。

- 在 /etc/systemd/system/ 中创建原始单元文件 /usr/lib/systemd/system/ 的副本并在此进行修改。这个副本会覆盖原始文件，因此不会应用软件包更新带来的更改。这个方法对无论软件包更新都应保留的重要单元更改都很有用。详情请查看 “覆盖默认单元配置”一节。

要返回这个单元的默认配置，请删除 /etc/systemd/system/ 中的自定义配置文件。要在不重启系统的情况下对单元文件应用更改，请执行：

```
systemctl daemon-reload
```

daemon-reload 选项会重新载入所有单元文件并重新创建依赖项树，这是立即将任何更改应用到单元文件所必需的。另外，您可以使用以下命令获得相同的结果：

```
init q
```

另外，如果修改后的单元文件属于正在运行的服务，则必须重启该服务以接受新设置：

```
systemctl restart name.service
```

重要

要修改由 SysV initscript 处理的服务的属性（如依赖项或超时），请不要修改 initscript 本身。相反，请为服务创建一个 systemd 置入配置文件，如 “扩展默认单元配置”一节 和 “覆盖默认单元配置”一节 所述。然后，以与普通 systemd 服务相同的方式管理此服务。

例如：要扩展 network 服务的配置，不要修改 /etc/rc.d/init.d/network initscript 文件。反之，创建新目录 /etc/systemd/system/network.service.d/ 和一个 systemd drop-in 文件 /etc/systemd/system/network.service.d/my_config.conf。然后将修改的值放到 drop-in 文件中。注：systemd 知道 network 服务为 network.service，这就是为什么创建的目录必须名为 network.service.d

扩展默认单元配置

要使用附加配置选项扩展默认单元文件，请首先在 /etc/systemd/system/ 中创建配置目录。如果扩展服务单元，以 root 用户身份执行以下命令：

```
mkdir /etc/systemd/system/name.service.d/
```

使用您要扩展的服务的名称替换 name。以上语法适用于所有单元类型。

在上一步中创建的目录中创建配置文件。请注意，文件名必须以 .conf 后缀结尾。类型：

```
touch /etc/systemd/system/name.service.d/config_name.conf
```

使用配置文件的名称替换 config_name。这个文件遵循一般的单元文件结构，因此所有指令都必须在适当的部分指定，请参阅 第 10.6.1 节 “了解单元文件结构”。

例如，要添加自定义依赖项，请使用以下内容创建配置文件：

```
[Unit]
Requires=new_dependency
After=new_dependency
```

这里的 new_dependency 代表这个单元被标记为依赖项。另一个例子是主进程退出后重新启动服务的配置文件，延迟 30 秒：

```
[Service]
Restart=always
RestartSec=30
```

建议您创建仅关注一项任务的小配置文件。这些文件可轻松地移动或者链接到其他服务的配置目录。

要应用对单位所做的更改，以 root 用户身份执行：

```
systemctl daemon-reload
systemctl restart name.service
```

例 10.20. 扩展 httpd.service 配置

要修改 httpd.service 单元，以便在启动 Apache 服务时自动执行自定义 shell 脚本，请执行以下

步骤。首先，创建目录和自定义配置文件：

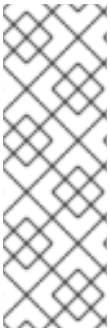
```
~]# mkdir /etc/systemd/system/httpd.service.d/
~]# touch /etc/systemd/system/httpd.service.d/custom_script.conf
```

如果想要用 Apache 自动启动的脚本位于 /usr/local/bin/custom.sh，在 custom_script.conf 文件中插入以下文本：

```
[Service]
ExecStartPost=/usr/local/bin/custom.sh
```

要应用单元更改，请执行：

```
~]# systemctl daemon-reload
~]# systemctl restart httpd.service
```



注意

/etc/systemd/system/ 配置文件中的配置文件优先于 /usr/lib/systemd/system/ 中的单元文件。因此，如果配置文件包含只能指定一次的选项，如 Description 或 ExecStart，则此选项的默认值将被覆盖。请注意，在 systemd-delta 命令的输出中，“监控覆盖单元”一节 中描述的这些单元总是被标记为 [EXTENDED]，即使在 sum 中，某些选项实际上会被覆盖。

覆盖默认单元配置

要在更新提供该单元文件的软件包后保留更改，首先要将该文件复制到 /etc/systemd/system/ 目录。要做到这一点，以 root 用户身份执行以下命令：

```
cp /usr/lib/systemd/system/name.service /etc/systemd/system/name.service
```

其中 name 代表您希望修改的服务单元的名称。以上语法适用于所有单元类型。

使用文本编辑器打开复制的文件，并进行必要的修改。要应用单元更改，以 root 用户身份执行：

```
systemctl daemon-reload
systemctl restart name.service
```

例 10.21. 更改超时限制

您可以为每个服务指定一个超时值，以防止出现故障的服务中断。否则，一般服务的超时时间会被默认设置为 90 秒，SysV 兼容的服务会被设置为 300 秒。

例如：要为 httpd 服务扩展超时限制：

1. 将 httpd 单元文件复制到 /etc/systemd/system/ 目录中：

```
cp /usr/lib/systemd/system/httpd.service /etc/systemd/system/httpd.service
```

2. 打开文件 /etc/systemd/system/httpd.service，并在 [Service] 部分指定 TimeoutStartSec 值：

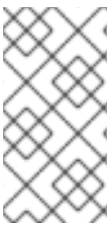
```
...
[Service]
...
PrivateTmp=true
TimeoutStartSec=10

[Install]
WantedBy=multi-user.target
...
```

3. 重新载入 systemd 守护进程：

```
systemctl daemon-reload
```

4. Optional.验证新的超时值：

```
systemctl show httpd -p TimeoutStartUSec
```



注意

要全局更改超时限制，在/etc/systemd/system.conf 中输入 DefaultTimeoutStartSec。请参阅 第 10.1 节 “systemd 简介”。

监控覆盖单元

要显示覆盖或修改的单元文件概述，请使用以下命令：

```
systemd-delta
```

例如，以上命令的输出结果如下：

```
[EQUIVALENT] /etc/systemd/system/default.target → /usr/lib/systemd/system/default.target
[OVERRIDDEN] /etc/systemd/system/autofs.service → /usr/lib/systemd/system/autofs.service

--- /usr/lib/systemd/system/autofs.service   2014-10-16 21:30:39.000000000 -0400
+++ /etc/systemd/system/autofs.service 2014-11-21 10:00:58.513568275 -0500
@@ -8,7 +8,8 @@
 EnvironmentFile=-/etc/sysconfig/autofs
 ExecStart=/usr/sbin/automount $OPTIONS --pid-file /run/autofs.pid
 ExecReload=/usr/bin/kill -HUP $MAINPID
-TimeoutSec=180
+TimeoutSec=240
+Restart=Always

 [Install]
 WantedBy=multi-user.target

[MASKED]   /etc/systemd/system/cups.service → /usr/lib/systemd/system/cups.service
[EXTENDED]  /usr/lib/systemd/system/sssd.service →
/etc/systemd/system/sssd.service.d/journal.conf

4 overridden configuration files found.
```

表 10.13 “systemd-delta 差别类型” 列出 systemd-delta 输出中可能出现的覆盖类型。请注意，如果文件被覆盖，systemd-delta 默认会显示与 diff 命令输出类似的总揽。

表 10.13. systemd-delta 差别类型

类型	描述
[MASKED]	屏蔽的单元文件，请参阅 第 10.2.7 节 “禁用服务” 以了解单元屏蔽的信息。
[EQUIVALENT]	未修改的副本会覆盖原始文件，但在内容上没有变化，通常是符号链接。
[REDIRECTED]	重定向到另一个文件的文件。
[OVERRIDEN]	覆盖和更改文件。
[EXTENDED]	使用 **/etc/systemd/system/ unit.d /** 目录中的 .conf 文件扩展的文件。

类型	描述
[UNCHANGED]	只有在使用 **--type=unchanged** 选项时才会显示未修改的文件。

最好在系统更新后运行 **systemd-delta**，以检查是否对当前由自定义配置覆盖的默认单元进行了更新。也有可能将输出限制在某种差别类型。例如，要仅查看覆盖的单元，请执行：

```
systemd-delta --type=overridden
```

10.6.5. 使用 Instantiation 单元

可以在运行时使用单一模板配置文件实例化多个单元。"@"字符用于标记模板并与其关联。实例化的单元可以从另一个单元文件（使用 **Requires** 或者 **Wants** 选项）或者 **systemctl start** 命令启动。以下列方式命名实例化服务单元：

```
template_name@instance_name.service
```

其中 **template_name** 代表模板配置文件的名称。将 **instance_name** 替换为单元实例的名称。多个实例可以指向带有通用于单元所有实例的配置选项的同一个模板文件。模板单元名称具有以下格式：

```
unit_name@.service
```

例如，单位文件中的以下 **Wants** 设置：

```
Wants=getty@ttyA.service,getty@ttyB.service
```

首先为给定服务单元进行 **systemd** 搜索。如果没有找到这样的单元，"@" 和类型后缀之间的部分将被忽略，**systemd** 搜索 **getty@.service** 文件，从中读取配置并启动服务。

通配符字符（称为单元指定符）可用于任何单元配置文件中。单元指定符替换某些单元参数，并在运行时解释。*表 10.14 “重要单元指定符”* 列出对模板单元特别有用的单元指定符。

表 10.14. 重要单元指定符

单元指定符	含义	描述

单元指定符	含义	描述
%n	完整单元名称	代表完整的单元名称，包括类型后缀。**%N** 具有相同的含义，但也将禁止的字符替换为 ASCII 代码。
%p	前缀名称	代表删除了类型后缀的单元名称。对于实例化单元 %p 代表"@"字符前面的单元名称的一部分。
%i	实例名称	是"@"字符和类型后缀之间的实例化单元名称的一部分。**%I** 具有相同的含义，但也会替换 ASCII 代码的禁止字符。
%H	主机名	代表在载入单元配置时的运行系统的主机名。
%t	运行时目录	代表运行时目录，对于 **root** 用户是 **/run**，对于非特权用户是 XDG_RUNTIME_DIR 变量的值。

有关单元指定符的完整列表，请参见 systemd.unit(5) 手册页。

例如, getty@.service 模板包含以下指令：

```
[Unit]
Description=Getty on %I
...
[Service]
ExecStart=-/sbin/agetty --noclear %I $TERM
...
```

当 getty@ttyA.service 和 getty@ttyB.service 实例化为上述模板时，Description= Getty on ttyA 和 Getty on ttyB 解析 为 Getty on ttyB。

10.7. 管理服务时的其他注意事项

在正常操作过程中，systemd 维护单元抽象和系统上活动的基础进程之间的关联。

From: man systemd

■

> Processes systemd spawns are placed in individual Linux control groups named after the unit which they belong to in the private systemd hierarchy. (see cgroups.txt[1] for more information about control groups, or short "cgroups"). systemd uses this to effectively keep track of processes. Control group information is maintained in the kernel, and is accessible via the file system hierarchy (beneath /sys/fs/cgroup/systemd/), or in tools such as ps(1) (ps xawf -eo pid,user,cgroup,args is particularly useful to list all processes and the systemd units they belong to).

cgroup 层次结构对于 systemd 的进程和服务健康状况视图至关重要。当进程分叉本身时，它将继承创建进程的 cgroup。在这种情况下，可以通过读取适用的 cgroup.procs 文件的内容来验证与给定单元关联的所有进程，例如：

```
~]# cat /sys/fs/cgroup/systemd/system.slice/httpd.service/cgroup.procs
11854
11855
11856
11857
11858
11859
```

输出与 systemctl status 单元 操作期间返回的 CGroup 信息匹配：

```
~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
  Active: active (running) since Wed 2019-05-29 12:08:16 EDT; 45s ago
    Docs: man:httpd(8)
          man:apachectl(8)
 Main PID: 11854 (httpd)
  Status: "Total requests: 0; Current requests/sec: 0; Current traffic:  0 B/sec"
  CGroup: /system.slice/httpd.service
          ├─11854 /usr/sbin/httpd -DFOREGROUND
          ├─11855 /usr/sbin/httpd -DFOREGROUND
          ├─11856 /usr/sbin/httpd -DFOREGROUND
          ├─11857 /usr/sbin/httpd -DFOREGROUND
          ├─11858 /usr/sbin/httpd -DFOREGROUND
          └─11859 /usr/sbin/httpd -DFOREGROUND

May 29 12:08:16 localhost systemd[1]: Starting The Apache HTTP Server...
May 29 12:08:16 localhost systemd[1]: Started The Apache HTTP Server.
```

要直接查看系统范围内进程的这些分组，可以使用 systemd-cgls 实用程序：

```
~]# systemd-cgls | head -17
├─1 /usr/lib/systemd/systemd --switched-root --system --deserialize 22
├─user.slice
│ └─user-0.slice
│   └─session-168.scope
│     ├─ 3049 login -- root
```

```
│   ├─11884 -bash
│   ├─11943 systemd-cgls
│   └─11944 head -17
└─system.slice
  ├─httpd.service
  │ ├─11854 /usr/sbin/httpd -DFOREGROUND
  │ ├─11855 /usr/sbin/httpd -DFOREGROUND
  │ ├─11856 /usr/sbin/httpd -DFOREGROUND
  │ ├─11857 /usr/sbin/httpd -DFOREGROUND
  │ ├─11858 /usr/sbin/httpd -DFOREGROUND
  │ └─11859 /usr/sbin/httpd -DFOREGROUND
  ├─rhnsd.service
```

要使 systemd 正常工作，必须通过 systemd 系统启动或停止服务，以维护正确的进程进行单元分组。任何执行外部操作的操作都会导致不创建所需的 cgroup 结构。这是因为 systemd 不知道正在启动的进程的特殊性质。

作为上述约束的示例，停止 httpd 服务，然后直接发布 /usr/sbin/httpd 会导致以下结果：

```
~]# systemctl stop httpd
~]# /usr/sbin/httpd
# systemd-cgls | head -17
├─1 /usr/lib/systemd/systemd --switched-root --system --deserialize 22
├─user.slice
│ └─user-0.slice
│   └─session-168.scope
│     ├─ 3049 login -- root
│     ├─11884 -bash
│     ├─11957 /usr/sbin/httpd
│     ├─11958 /usr/sbin/httpd
│     ├─11959 /usr/sbin/httpd
│     ├─11960 /usr/sbin/httpd
│     ├─11961 /usr/sbin/httpd
│     ├─11962 /usr/sbin/httpd
│     ├─11963 systemd-cgls
│     └─11964 head -17
└─system.slice
  ├─rhnsd.service
  │ └─3261 rhnsd
```

请注意，httpd 进程现在在 user-0.slice 和 session-168.scope 下可见。此服务被视为启动的进程，而非系统服务，后者应直接监控和管理。由于这种错误对齐可能会发生一些故障，包括但不限于：

- 在系统关闭或重启事件期间，服务无法正确关闭。
- 用户注销期间会发送意外信号，如 SIGHUP 和 SIGTERM。

- 带有 Restart= 指令时，不会自动重启失败的进程

注意

非大型应用程序关闭事件可能会导致大量后续应用程序故障，如客户端故障、数据丢失和磁盘损坏。

10.8. 其它资源

有关 systemd 及其在 Red Hat Enterprise Linux 7 中使用的详情，请查看以下列出的资源。

安装的文档

- systemctl(1)- systemctl 命令行工具的 man page 提供了支持的选项和命令的完整列表。
- systemd(1)- systemd 系统和服务管理器的手册页面提供有关其概念和记录可用命令行选项、环境变量、支持的配置文件和目录、可识别信号和可用内核选项的更多信息。
- systemd-delta(1)- systemd-delta 实用程序的 man page，可用于查找扩展和覆盖的配置文件。
- systemd.unit(5)- 名为 systemd.unit 的 man page 提供关于 systemd 单元文件和记录所有可用配置选项的详细信息。
- systemd.service(5)- 名为 systemd.service 的 man page 记录服务单元文件的格式。
- systemd.target(5)- 名为 systemd.target 的 man page 记录了目标单元文件的格式。
- systemd.kill(5)- 名为 systemd.kill 的 man page 记录了进程终止程序的配置。

在线文档

- 红帽企业 Linux 7 联网指南 - 红帽企业 Linux 7 的网络指南记录了有关在此系统中配置和管理网络接口、网络和网络服务的相关信息。它介绍了 hostnamectl 实用程序，说明如何使用 在

命令行中查看和设置本地和远程的主机名，并提供有关主机名称和域名选择的重要信息。

- 红帽企业 Linux 7 桌面迁移和管理指南 - 红帽企业 Linux 7 的桌面迁移和管理指南 记录了系统上 GNOME 3 桌面的迁移规划、部署、配置和管理。它引进了 logind 服务，列举其最重要的功能，并说明如何使用 loginctl 实用程序列出活动会话并启用多椅支持。

- Red Hat Enterprise Linux 7 SELinux 用户和管理员指南 - Red Hat Enterprise Linux 7 的 SELinux 用户和管理员指南 介绍了 SELinux 的基本原理，并详细介绍了如何配置和使用 SELinux 与 Apache HTTP 服务器、Postfix、PostgreSQL 或 OpenShift 等服务。它解释了如何为 systemd 管理的系统服务配置 SELinux 访问权限。

- Red Hat Enterprise Linux 7 安装指南 - Red Hat Enterprise Linux 7 安装 指南包括了如何在 AMD64 和 Intel 64 系统、64 位 IBM Power Systems 服务器和 IBM Z 中安装该系统。它还涵盖了高级安装方法，如 Kickstart 安装、PXE 安装和通过 VNC 协议安装。另外，它描述了常见的安装后任务，并解释了如何对安装问题进行故障排除，包括如何引导进入救援模式或恢复 root 密码的详细说明。

- 红帽企业 Linux 7 安全指南 - 红帽企业 Linux 7 安全指南 帮助用户和管理员学习保护工作站和服务器免受本地和远程入侵、攻击和恶意活动的流程和实践。它还说明了如何保护关键系统服务。

- systemd 主页 - 项目主页提供了更多关于 systemd 的信息。

另请参阅

- 第 2 章 系统位置和键盘配置 记录如何管理系统区域设置和键盘布局。它介绍了如何使用 localectl 实用程序查看当前的区域设置，列出可用的区域设置，并在命令行中设置系统区域设置，以及查看当前的键盘布局、列出可用的键盘映射，以及在命令行中启用特定的键盘布局。

- 第 3 章 配置日期和时间 记录如何管理系统日期和时间。它解释了实时时钟与系统时钟之间的区别，并描述了如何使用 timedatectl 实用程序显示系统时钟的当前设置、配置日期和时间、更改时区，以及将系统时钟与远程服务器同步。

- 第 6 章 获取特权 文档如何使用 su 和 sudo 命令获得管理权限。

- 第 12 章 OpenSSH 描述如何配置 SSH 服务器以及如何使用 ssh、scp 和 sftp 客户端实用程序进行访问。

- 第 23 章 查看和管理日志文件 介绍了 journald。它描述日志、引入 journald 服务，以及如何使用 journalctl 实用程序查看日志条目、进入实时查看模式和过滤日志条目。此外，本章介绍了如何为非 root 用户授予系统日志的访问权限，并为日志文件启用持久存储。

第 11 章 配置系统可访问性

Orca 屏幕阅读器确保了红帽企业 Linux 7 的可访问性，该屏幕读取器包含在操作系统的默认安装中。本章解释了系统管理员如何配置系统以支持具有视觉问题的用户。

orca 从屏幕读取信息，并使用以下方法与用户通信：

- 演讲组合器，提供演讲输出
- 程序错误显示，它提供了一个 tactile 输出

有关 Orca 设置的详情，请查看其 帮助页面。

为了使 Orca的通信输出正常工作，系统管理员需要：

- 配置 brltty 服务，如 所述 第 11.1 节 “配置 brltty 服务”
- 打开 Always Show Universal Access Menu，如 所述 第 11.2 节 “switch On Always Show Universal Access Menu”
- 启用 Festival 发音器，如 所述 第 11.3 节 “启用 Festival Speech Synthesis 系统”

11.1. 配置 BRLTTY 服务

Braille 显示使用 brltty 服务为视觉损害的用户提供 tactile 输出。

启用 brltty 服务

除非 brltty 正在运行，否则 braille 显示无法正常工作。默认情况下，brltty 被禁用。在引导时启用 brltty ：

```
~]# systemctl enable brltty.service
```

授权用户使用 Braille 显示

要设置有权使用损坏显示的用户，请选择以下流程之一，它们具有相同的效果。使用 /etc/brltty.conf 文件的步骤适用于无法将用户和组分配给文件的文件系统。使用 /etc/brlapi.key 文件的步骤仅适用于可以为其分配用户或组的文件系统。

使用 /etc/brltty.conf 设置对 Braille 显示的访问

1. 打开 /etc/brltty.conf 文件，并找到名为 应用程序编程接口参数 的部分。

2. 指定用户。

   a. 要指定一个或多个单独的用户，请在以下行中列出用户：

   ```
   api-parameters Auth=user:user_1, user_2, ...  # Allow some local user
   ```

   b. 要指定用户组，请在以下行中输入它的名称：

   ```
   api-parameters Auth=group:group  # Allow some local group
   ```

使用 /etc/brlapi.key 设置对 Braille 显示的访问

1. 创建 /etc/brlapi.key 文件。

   ```
   ~]# mcookie > /etc/brlapi.key
   ```

2. 将 /etc/brlapi.key 的所有权更改为特定用户或组。

   a. 指定单个用户：

   ```
   ~]# chown user_1 /etc/brlapi.key
   ```

   b. 指定组：

   ```
   ~]# chown group_1 /etc/brlapi.key
   ```

3. 调整 /etc/brltty.conf 的内容 使其包含以下内容：

```
api-parameters Auth=keyfile:/etc/brlapi.key
```

设置 Braille 驱动程序

/etc/brltty.conf 中的 braille-driver 指令指定用于 braille 显示的双字母驱动程序识别代码。

设置 Braille 驱动程序

1. 确定您是否要使用自动检测来查找相应的连字符驱动程序。

   a. 如果要使用自动检测，请将 braille 驱动程序 保留为 auto，这是默认选项。

   ```
   braille-driver auto  # autodetect
   ```

   > **警告**
   >
   > 自动检测尝试所有驱动程序.因此，可能需要很长时间甚至失败。因此，建议设置特定的损坏驱动程序。

   b. 如果您不想使用自动检测，请在 braille -driver 指令中指定所需 braille 驱动程序的识别代码。

   从 /etc/brltty.conf 提供的列表中选择所需的 braille 驱动程序识别代码，例如：

   ```
   braille-driver xw  # XWindow
   ```

   您还可以设置多个驱动程序，用逗号分开，然后在其中执行自动检测。

设置 Braille 设备

/etc/brltty.conf 中的 braille-device 指令指定 braille 显示连接到的设备。支持以下设备类型（请参阅 表 11.1 “Braille 设备类型和 Correspoing Syntax”）：

表 11.1. Braille 设备类型和 Correspoing Syntax

损坏设备类型	Type 的语法
串行设备	serial:path [a]
USB 设备	[serial-number] [b]
蓝牙设备	bluetooth:address

[a] 相对路径位于 **/dev**。

[b] 此处的括号表示可选性。

特定设备的设置示例：

```
braille-device serial:ttyS0        # First serial device
braille-device usb:             # First USB device matching braille driver
braille-device usb:nnnnn        # Specific USB device by serial number
braille-device bluetooth:xx:xx:xx:xx:xx:xx # Specific Bluetooth device by address
```

您还可以设置多个设备，用逗号分开，每个设备将被依次探测。

警告

如果设备由串行至USB 适配器连接，则将 braille-device 设置为 usb：无法正常工作。在这种情况下，请识别内核为适配器创建的虚拟串行设备。虚拟串行设备类似如下：

```
serial:ttyUSB0
```

```
You can find the actual device name in the kernel messages on the device plug with the following command:
```

```
~]# dmesg | fgrep ttyUSB0
```

为 particular Braille Displays 设置特定参数

如果您需要为特定 braille 显示设置特定参数，请在 /etc/brltty.conf 中使用 braille-parameters 指

令。braille-parameters 指令将非常规参数传递到 braille 驱动程序。从 /etc/brltty.conf 列表中的列表中选择所需的参数。

设置文本表

/etc/brltty.conf 中的 text-table 指令指定用于对符号进行编码的文本表。到文本表的相对路径位于 /etc/brltty/Text/ 目录中。

设置文本表

1. 决定是否使用自动选择来查找相应的文本表。

2.

a. 如果要使用自动选择，请将 text-table 保留 为 auto，这是默认选项。

```
text-table auto  # locale-based autoselection
```

这可确保执行基于本地的自动选择并回退到 en-nabcc。

b. 如果您不想使用 autoselection，请从 /etc/brltty.conf 中的列表中选择所需的 文本-table。

例如，使用美国英语的文本表：

```
text-table en_US  # English (United States)
```

设置合同表

/etc/brltty.conf 中的 contraction -table 指令指定用于编码缩写的表。到特定 合同表的相对路径位于 /etc/brltty/Contraction/ 目录中。

从 /etc/brltty.conf 中的列表中选择所需的conion -table。

例如，要将合同表用于美国英语，第 2 级：

```
contraction-table en-us-g2  # English (US, grade 2)
```

警告

如果没有指定，则不使用合同表。

11.2. SWITCH ON ALWAYS SHOW UNIVERSAL ACCESS MENU

要切换到 Orca 屏幕阅读器，请按 Super+Alt+S 组合键。因此，顶栏中会显示 Universal Access Menu 图标。

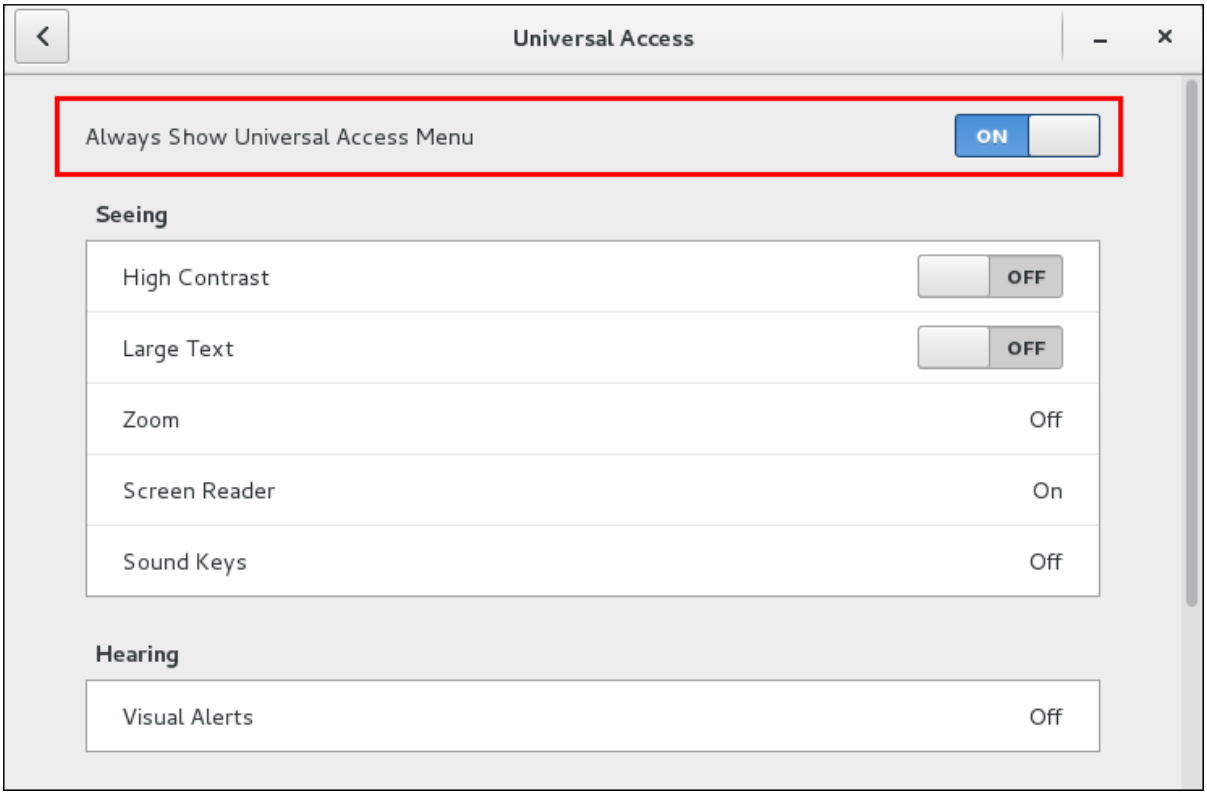
警告

如果用户从 Universal Access Menu 中关闭所有提供的选项，此图标将消失。缺少图标可能会导致具有视觉问题的用户遇到困难。系统管理员可以通过切换 Always Show Universal Access 菜单来防止图标不可访问。当打开 Always Show Universal Access Menu 时，即使此菜单中的所有选项都已关闭，顶栏中也会显示此图标。

切换 Always 显示通用访问菜单

1. *打开 Gnome settings 菜单，然后单击 Universal Access。*

2. *打开 Always Show Universal Access Menu.*



3. *可选：验证是否在顶栏中显示 Universal Access Menu 图标，即使此菜单中的所有选项都已关闭。*

11.3. 启用 FESTIVAL SPEECH SYNTHESIS 系统

默认情况下，Organ 使用 eSpeak 发音器，但也支持 Festival Speech Synthesis 系统。eSpeak 和 Festival Speech Synthesis System(Festival)结合了不同语音。有些用户可能更喜欢使用默认的 eSpeak 组合 器。要启用 Festival，请按照以下步骤执行：

安装 Festival 并使其在引导时运行

1. *安装 festival:*

```
~]# yum install festival festival-freebsoft-utils
```

2. *使 Festival 在引导时运行：*

   a. *创建新的 systemd 单元文件：*

在 /etc/systemd/system/ 目录中创建一个文件，并使其可执行。

```
~]# touch /etc/systemd/system/festival.service
~]# chmod 664 /etc/systemd/system/festival.service
```

b. 确保 /usr/bin/festival_server 文件中的 脚本用于运行 Festival。在 /etc/systemd/system/festival.service 文件中添加以下内容：

```
[Unit]
Description=Festival speech synthesis server
[Service]
ExecStart=/usr/bin/festival_server
Type=simple
```

c. 通知 systemd 存在新的 festival.service 文件：

```
~]# systemctl daemon-reload
~]# systemctl start festival.service
```

d. 启用 festival.service ：

```
~]# systemctl enable festival.service
```

为 Festival 选择选择

Festival 提供多个语音。

要获得可用的语音，请从以下列表中安装相关软件包：

- festvox-awb-arctic-hts
- festvox-bdl-arctic-hts
- festvox-clb-arctic-hts
- festvox-kal-diphone

- ***festvox-ked-diphone***
- ***festvox-rms-arctic-hts***
- ***festvox-slt-arctic-hts***
- ***hispavoces-pal-diphone***
- ***hispavoces-sfl-diphone***

查看有关特定语音的详细信息：

```
~]# yum info package_name
```

要使所需语音可用，请使用这个语音安装软件包，然后重启：

```
~]# yum install package_name
~]# reboot
```

第 12 章 OPENSSH

SSH (Secure Shell)是一种协议，它使用客户端-服务器架构促进两个系统之间的安全通信，并允许用户远程登录服务器主机系统。与其他远程通信协议（如 FTP 或 Telnet ）不同，SSH 对登录会话进行加密，使得入侵者很难收集未加密的密码。

ssh 程序旨在替换用于登录远程主机（如 telnet 或 rsh ）的旧、不太安全的终端应用。名为 scp 的相关程序 取代了设计用于在主机之间复制文件的旧程序，如 rcp。由于这些较旧的应用程序不会加密客户端和服务器之间传输的密码，因此尽可能避免这些密码。使用安全方法登录远程系统可降低客户端系统和远程主机的风险。

Red Hat Enterprise Linux 包括常规 OpenSSH 软件包 openssh，以及 OpenSSH 服务器、open ssh-server 和客户端、open ssh-clients、软件包。请注意，OpenSSH 软件包需要 OpenSSL package openssl-libs，它会安装几个重要的加密库，使 OpenSSH 提供加密通信。

12.1. SSH 协议

12.1.1. 为什么使用 SSH？

潜在入侵者拥有各种工具可供他们使用，使他们能够中断、拦截和重新路由网络流量，以努力访问系统。总体而言，这些威胁可分类如下：

拦截两个系统间通信

攻击者可以位于通信方之间的网络上，复制他们之间传递的任何信息。他可以截获和保存信息，或者更改信息并将其发送给预期收件人。

此攻击通常使用数据包嗅探器来执行，这是一种比较常见的网络实用程序，可捕获通过网络流的每个数据包，并分析其内容。

模拟特定主机

攻击者的系统被配置为作为传输的预期接收者。如果此策略正常工作，用户系统仍不知道它正在与错误的主机通信。

此攻击可以通过称为 DNS 投毒或所谓的 IP 欺骗进行。在第一种情形中，入侵者使用破解的 DNS 服务器将客户端系统指向恶意重复的主机。在第二种情况下，入侵者会发送似乎来自可信主机的虚假网络数据包。

这两种技术都会截获潜在的敏感信息，如果因为恶意原因而进行拦截，结果可能会令人沮丧。如果 SSH 用于远程 shell 登录和文件复制，这些安全威胁可能会大大降低。这是因为 SSH 客户端和服务器使用数字签名来验证其身份。另外，所有客户端和服务器系统之间的沟通都是加密的。尝试欺骗通信两侧的身份无效，因为每个数据包都使用仅由本地和远程系统识别的密钥进行加密。

12.1.2. 主要功能

SSH 协议提供以下保护：

没有人可以组成预期服务器

在初始连接后，客户端可以验证它是否连接到之前连接到的同一服务器。

没有人可以捕获验证信息

客户端使用强大的 128 位加密将其身份验证信息传输到服务器。

没有人可以拦截通信

会话期间发送和接收的所有数据都使用 128 位加密进行传输，因此被拦截的传输极难解密和读取。

另外，它还提供以下选项：

它为通过网络使用图形应用程序提供了安全方法

使用名为 X11 转发的技术，客户端可以从服务器转发X11（X Window 系统）应用程序。

它提供了一种保护其他不安全协议的方法

SSH 协议加密它发送和接收的所有内容。使用称为端口转发的技术，SSH 服务器可以成为保护其他不安全协议（如 POP）并提升整体系统和数据安全性的渠道。

它可以用来创建安全频道

OpenSSH 服务器和客户端可以配置为创建一个类似于虚拟专用网络的隧道，以用于服务器和客户端计算机之间的流量。

它支持 Kerberos 身份验证

可以配置 OpenSSH 服务器和客户端，以使用 Kerberos 网络身份验证协议的 GSSAPI （通用安全服务应用程序接口）实施进行身份验证。

12.1.3. 协议版本

目前存在两种 SSH 变体：版本 1 和更新版本 2。红帽企业 Linux 7 下的 OpenSSH 套件使用 SSH 版本 2，其增强的密钥交换算法不会受到版本 1 中已知漏洞的影响。在 Red Hat Enterprise Linux 7 中，OpenSSH 套件不支持版本 1 连接。

12.1.4. SSH 连接的事件序列

以下一系列事件有助于保护两个主机之间的 SSH 通信的完整性。

1. 制作加密握手，以便客户端能够验证它是否与正确的服务器通信。
2. 客户端和远程主机之间的连接传输层使用对称密码进行加密。
3. 客户端向服务器验证自身。
4. 客户端通过加密连接与远程主机交互。

12.1.4.1. 传输层

传输层的主要作用是促进两台主机之间在身份验证时以及后续通信期间的通信安全。传输层通过处理数据的加密和解密，以及在数据包发送和接收时提供完整性保护来实现此目的。传输层还提供压缩，加快信息的传输速度。

SSH 客户端联系服务器后，交换关键信息，以便两个系统能够正确构建传输层。在此交换中会执行以下步骤：

- 交换密钥
- 确定公钥加密算法
- 对称加密算法确定
- 确定消息验证算法

- 确定哈希算法

在密钥交换期间，服务器利用唯一主机密钥向客户端识别自身。如果之前客户端从未与此特定服务器通信，则服务器的主机密钥对客户端未知且不会连接。OpenSSH 通过接受服务器的主机密钥来解决这个问题。这是在用户收到通知并且已接受并验证新主机密钥之后完成的。在后续连接中，会根据客户端中保存的版本检查服务器的主机密钥，从而确保客户端确实与预期服务器通信。如果以后主机密钥不再匹配，用户必须删除客户端的保存版本，然后才能进行连接。

警告

在初始联系期间，攻击者有可能伪装为 SSH 服务器，因为本地系统不知道预期服务器和攻击者设置的假服务器之间的区别。为帮助防止这种情况，请在第一次连接或出现主机密钥不匹配之前联系服务器管理员来验证新 SSH 服务器的完整性。

SSH 设计为使用几乎任何类型的公钥算法或编码格式。在初始密钥交换创建了用于交换的哈希值和共享 secret 值后，两个系统会立即开始计算新的密钥和算法，以保护身份验证和将来通过连接发送的数据。

使用给定密钥和算法传输一定数量的数据（具体取决于 SSH 实施）后，再进行另一个密钥交换，生成另一组哈希值和新共享 secret 值。即使攻击者能够确定哈希值和共享 secret 值，此信息仅在有限时间内有用。

12.1.4.2. Authentication

传输层构建了一个安全隧道以在两个系统之间传递信息后，服务器会告知客户端支持的不同身份验证方法，例如使用私钥编码签名或输入密码。然后，客户端尝试使用以下任一支持的方法对服务器进行身份验证。

SSH 服务器和客户端可以配置为允许不同类型的身份验证，从而互相提供最佳的控制量。服务器可以根据其安全模型决定它支持的加密方法，客户端也可以选择从可用选项尝试的身份验证方法顺序。

12.1.4.3. Channels

在 SSH 传输层成功验证后，可以通过称为多路复用的技术打开多个通道[1]。这些通道各自处理不同终端会话和转发 X11 会话的通信。

客户端和服务器都可以创建新频道。然后，在连接的每个末尾为每个频道分配一个不同的数字。当客户端尝试打开新频道时，客户端会随请求一起发送频道号。此信息由服务器存储，用于直接与该通道通信。这样，不同类型的会话不会相互影响，因此当给定会话结束时，其通道可以在不中断主 SSH 连接的情况下关闭。

频道还支持 flow-control，允许它们以有序的方式发送和接收数据。这样，在客户端收到打开频道的消息前，不会通过通道发送数据。

客户端和服务器根据客户端请求的服务类型以及用户连接到网络的方式自动协商每个频道的特征。这为处理不同类型的远程连接提供了极大的灵活性，无需更改协议的基本基础架构。

12.2. 配置 OPENSSH

12.2.1. 配置文件

共有两组不同的配置文件：用于客户端程序（即 ssh、scp 和 s ftp）的配置文件，以及用于服务器（sshd 守护进程）的配置文件。

系统范围的 SSH 配置信息保存在 /etc/ssh/ 目录中，如 表 12.1 “系统范围的配置文件” 所述。用户特定的 SSH 配置信息保存在用户主目录中的 ~/.ssh/ 中，如 表 12.2 “用户特定配置文件” 所述。

表 12.1. 系统范围的配置文件

File	描述
/etc/ssh/moduli	包含 Diffie-Hellman 组，用于 Diffie-Hellman 密钥交换，这对构建安全传输层至关重要。当在 SSH 会话开始时交换密钥时，会创建一个共享的 secret 值，它不能由任一方单独决定。该值随后用于提供主机身份验证。
/etc/ssh/ssh_config	默认的 SSH 客户端配置文件。请注意，如果 ~/.ssh/config 存在，它将被 ~/.ssh/config 覆盖。
/etc/ssh/sshd_config	sshd 守护进程的配置文件。
/etc/ssh/ssh_host_ecdsa_key	sshd 守护进程使用的 ECDSA 私钥。
/etc/ssh/ssh_host_ecdsa_key.pub	sshd 守护进程使用的 ECDSA 公钥。

File	描述
/etc/ssh/ssh_host_rsa_key	**sshd** 守护进程用于 SSH 协议版本 2 的 RSA 私钥。
/etc/ssh/ssh_host_rsa_key.pub	**sshd** 守护进程用于 SSH 协议版本 2 的 RSA 公钥。
/etc/pam.d/sshd	**sshd** 守护进程的 PAM 配置文件。
/etc/sysconfig/sshd	**sshd** 服务的配置文件。

表 12.2. 用户特定配置文件

File	描述
~/.ssh/authorized_keys	保存服务器的授权公钥列表。当客户端连接到服务器时，服务器通过检查存储在此文件中的签名公钥来验证客户端的身份验证。
~/.ssh/id_ecdsa	包含用户的 ECDSA 私钥。
~/.ssh/id_ecdsa.pub	用户的 ECDSA 公钥.
~/.ssh/id_rsa	**ssh** 用于 SSH 协议版本 2 的 RSA 私钥。
~/.ssh/id_rsa.pub	**ssh** 用于 SSH 协议版本 2 的 RSA 公钥。
~/.ssh/known_hosts	包含用户访问的 SSH 服务器的主机密钥。此文件对于确保 SSH 客户端连接到正确的 SSH 服务器非常重要。

警告

如果设置 SSH 服务器，请不要使用 /etc/ssh/sshd_config 文件中的 UsePrivilegeSeparation no 指令来关闭特权隔离功能。关闭特权解析会禁用许多安全功能，并会使服务器暴露于潜在的安全漏洞和目标攻击。有关 UsePrivilegeSeparation 的更多信息，请参阅 sshd_config(5)手册页，或者 /etc/ssh/sshd_config 文件中 UsePrivilegeSeparation 指令的重要性以及如何对其进行测试？红帽知识库文章.

有关 SSH 配置文件中可以使用的各种指令的详情，请查看 ssh_config(5)和 sshd_config(5)手册页。

12.2.2. 启动 OpenSSH 服务器

为了运行 OpenSSH 服务器，您必须安装 openssh-server 软件包。有关如何安装新软件包的详情请参考 第 9.2.4 节 “安装软件包”。

要在当前会话中启动 sshd 守护进程，以 root 用户身份在 shell 提示符后输入以下内容：

```
~]# systemctl start sshd.service
```

要在当前会话中停止正在运行的 sshd 守护进程，以 root 用户身份运行以下命令：

```
~]# systemctl stop sshd.service
```

如果您希望守护进程在引导时自动启动，以 root 用户身份输入 ：

```
~]# systemctl enable sshd.service
Created symlink from /etc/systemd/system/multi-user.target.wants/sshd.service to
/usr/lib/systemd/system/sshd.service.
```

sshd 守护进程依赖于 network.target 目标 单元，这足以用于静态配置的网络接口和默认的 ListenAddress 0.0.0.0 选项。要在 ListenAddress 指令中指定不同的地址并使用较慢的动态网络配置，请将 network-online.target 目标 单元的依赖关系添加到 sshd.service 单元文件中。要做到这一点，使用以下选项创建 /etc/systemd/system/sshd.service.d/local.conf 文件：

```
[Unit]
Wants=network-online.target
After=network-online.target
```

之后，使用以下命令重新载入 systemd 管理器配置：

```
~]# systemctl daemon-reload
```

有关如何在 Red Hat Enterprise Linux 中管理系统服务的详情请参考 第 10 章 使用 systemd 管理服务。

请注意，如果您重新安装系统，则会创建新的识别密钥集合。因此，在重新安装前使用任何 OpenSSH 工具连接到该系统的客户端会看到以下信息：

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@: REMOTE HOST IDENTIFICATION HAS CHANGED!   @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the RSA host key has just been changed.
```

要防止这种情况，您可以从 /etc/ssh/ 目录中备份相关文件。有关完整的列表，请查看 表 12.1 “系统范围的配置文件”，每当您重新安装系统时都会恢复文件。

12.2.3. 远程连接需要 SSH

要使 SSH 真正有效，应禁止使用不安全的连接协议。否则，用户的密码可能通过 SSH 对一个会话进行保护，稍后仅在使用 Telnet 登录时捕获。要禁用的一些服务包括 telnet、rsh、rlogin 和 vsftpd。

有关如何配置 vsftpd 服务的详情请参考 第 16.2 节 “FTP”。要了解如何在 Red Hat Enterprise Linux 7 中管理系统服务，请参阅 第 10 章 使用 systemd 管理服务。

12.2.4. 使用基于密钥的身份验证

要进一步提高系统安全性，请生成 SSH 密钥对，然后通过禁用密码身份验证来强制进行基于密钥的身份验证。要做到这一点，在文本编辑器（如 vi 或 nano ）中打开 /etc/ssh/sshd_config 配置文件，并更改 PasswordAuthentication 选项，如下所示：

```
PasswordAuthentication no
```

如果您在使用新默认安装以外的系统中，请检查 PubkeyAuthentication no 没有设置。如果远程连接，不使用控制台或带外访问，建议在禁用密码身份验证前测试基于密钥的登录过程。

要能够使用 ssh、scp 或 sftp 从客户端计算机连接到服务器，请按照以下步骤生成授权密钥对：请注意，必须为每个用户单独生成密钥。

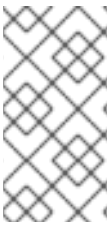
要在 NFS 挂载的主目录中使用基于密钥的身份验证，请首先启用 use_nfs_home_dirs SELinux 布尔值：

```
~]# setsebool -P use_nfs_home_dirs 1
```

Red Hat Enterprise Linux 7 默认使用 SSH 协议 2 和 RSA 密钥（详情请参阅 第 12.1.3 节 “协议版本”）。

重要

如果以 root 身份完成这些步骤，则只有 root 用户才能使用该密钥。



注意

如果您重新安装您的系统并希望保留先前生成的密钥对，请备份 ~/.ssh/ 目录。重新安装后，将其复制到主目录中。此过程可以针对您系统上的所有用户完成，包括 root 用户。

12.2.4.1. 生成密钥对

要为 SSH 协议的版本 2 生成 RSA 密钥对，请按照以下步骤执行：

1. 通过在 shell 提示符后输入以下内容来生成 RSA 密钥对：

```
~]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/USER/.ssh/id_rsa):
```

2. 按 Enter 键确认新创建的密钥的默认位置 ~/.ssh/id_rsa。

3. 输入密码短语，并在系统提示时再次输入密码进行确认。为安全起见，请避免使用与登录到您的帐户相同的密码。

之后，您将会看到类似如下的信息：

```
Your identification has been saved in /home/USER/.ssh/id_rsa.
Your public key has been saved in /home/USER/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:UNlgIT4wfhdQH/K7yqmjsbZnnyGDKiDviv492U5z78Y
USER@penguin.example.com
The key's randomart image is:
+---[RSA 2048]----+
|o ..==o+.    |
|.+ . .=oo    |
| .o. ..o     |
| ... ..    |
```

```
|   .S    |
|o .   .    |
|o+ o .o+ ..   |
|+.++=o*.o .E   |
|BBBo+Bo. oo   |
+----[SHA256]-----+
```

注意

要获得之前版本中的默认指纹 MD5 密钥指纹，请使用带有 -E md5 选项的 ssh-keygen 命令。

4. 默认情况下，~/.ssh/ 目录的权限被设置为 rwx------ 或 700 （以八进制表示法表示）。这是为了确保只有 USER 可以查看其内容。如果需要，可以使用以下命令确认：

```
~]$ ls -ld ~/.ssh
drwx------. 2 USER USER 54 Nov 25 16:56 /home/USER/.ssh/
```

5. 要将公钥复制到远程机器中，以以下格式发出命令：

```
ssh-copy-id user@hostname
```

这将复制最新修改的 ~/.ssh/id*.pub 公钥（如果尚未安装）。或者，指定公钥的文件名，如下所示：

```
ssh-copy-id -i ~/.ssh/id_rsa.pub user@hostname
```

这会将 ~/.ssh/id_rsa.pub 的内容复制到您要连接的计算机上的 ~/.ssh/authorized_keys 文件中。如果文件已存在，则密钥将附加到该文件末尾。

要为 SSH 协议的版本 2 生成 ECDSA 密钥对，请按照以下步骤执行：

1. 通过在 shell 提示符下键入以下内容来生成 ECDSA 密钥对：

```
~]$ ssh-keygen -t ecdsa
Generating public/private ecdsa key pair.
Enter file in which to save the key (/home/USER/.ssh/id_ecdsa):
```

2. 按 Enter 键确认新创建的密钥的默认位置 ~/.ssh/id_ecdsa。

3. 输入密码短语，并在系统提示时再次输入密码进行确认。为安全起见，请避免使用与登录到您的帐户相同的密码。

   之后，您将会看到类似如下的信息：

```
Your identification has been saved in /home/USER/.ssh/id_ecdsa.
Your public key has been saved in /home/USER/.ssh/id_ecdsa.pub.
The key fingerprint is:
SHA256:8BhZageKrLXM99z5f/AM9aPo/KAUd8ZZFPcPFWqK6+M
USER@penguin.example.com
The key's randomart image is:
+---[ECDSA 256]---+
|   . .   +=|
| . . . =   o.o|
| + . * .  o...|
| = . . * . + +..|
|. + . . So o * ..|
|  . o . .+ = ..|
|   o oo ..=. .|
|    ooo...+ |
|   .E++oo  |
+----[SHA256]-----+
```

4. 默认情况下，~/.ssh/ 目录的权限被设置为 rwx------ 或 700 （以八进制表示法表示）。这是为了确保只有 USER 可以查看其内容。如果需要，可以使用以下命令确认：

```
~]$ ls -ld ~/.ssh
      ~]$ ls -ld ~/.ssh/
drwx------. 2 USER USER 54 Nov 25 16:56 /home/USER/.ssh/
```

5. 要将公钥复制到远程机器中，以以下格式发出命令：

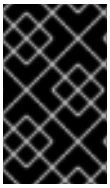
```
ssh-copy-id USER@hostname
```

   这将复制最新修改的 ~/.ssh/id*.pub 公钥（如果尚未安装）。或者，指定公钥的文件名，如下所示：

```
ssh-copy-id -i ~/.ssh/id_ecdsa.pub USER@hostname
```

   这会将 ~/.ssh/id_ecdsa.pub 的内容复制到您要连接的机器上的 ~/.ssh/authorized_keys 中。如果文件已存在，则密钥将附加到该文件末尾。

有关如何设置您的系统以记住密码，请参阅 第 12.2.4.2 节 “配置 ssh-agent”。



重要

私钥仅供您个人使用，因此务必不要将其提供给任何人。

12.2.4.2. 配置 ssh-agent

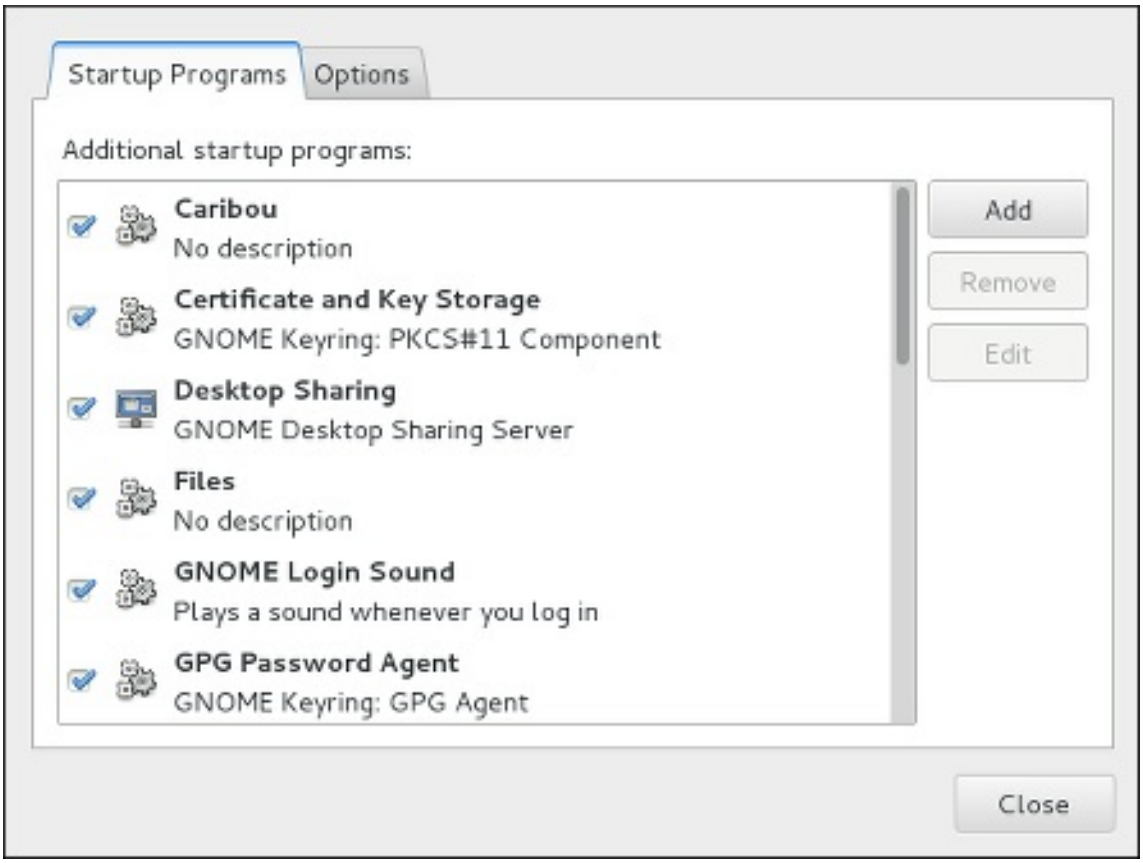
要存储密码短语，以便您不必在每次与远程计算机发起连接时输入密码短语，您可以使用 ssh-agent 身份验证代理。如果您正在运行 GNOME，您可以将其配置为在每次登录时提示您输入密码短语，并在整个会话期间记住密码短语。否则，您可以存储特定 shell 提示符的密码短语。

要在 GNOME 会话中保存密码短语，请按照以下步骤执行：

1. 确保已安装了 openssh-askpass 软件包。如果没有，请参阅 第 9.2.4 节 “安装软件包” 以了解有关如何在 Red Hat Enterprise Linux 中安装新软件包的更多信息。

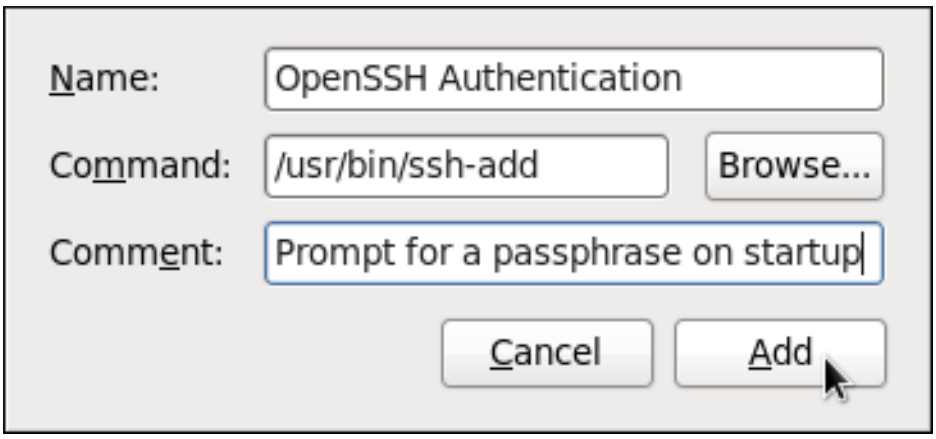
2. 按 Super 键进入 Activities Overview，键入 Startup Applications，然后按 Enter 键。此时会出现启动应用首选项工具。默认情况下，将显示包含可用启动程序列表的选项卡。Super 键显示在各种 guis 中，具体取决于键盘和其他硬件，但通常作为 Windows 或 Command 键，通常在空格的左侧。

图 **12.1.** 启动应用首选项



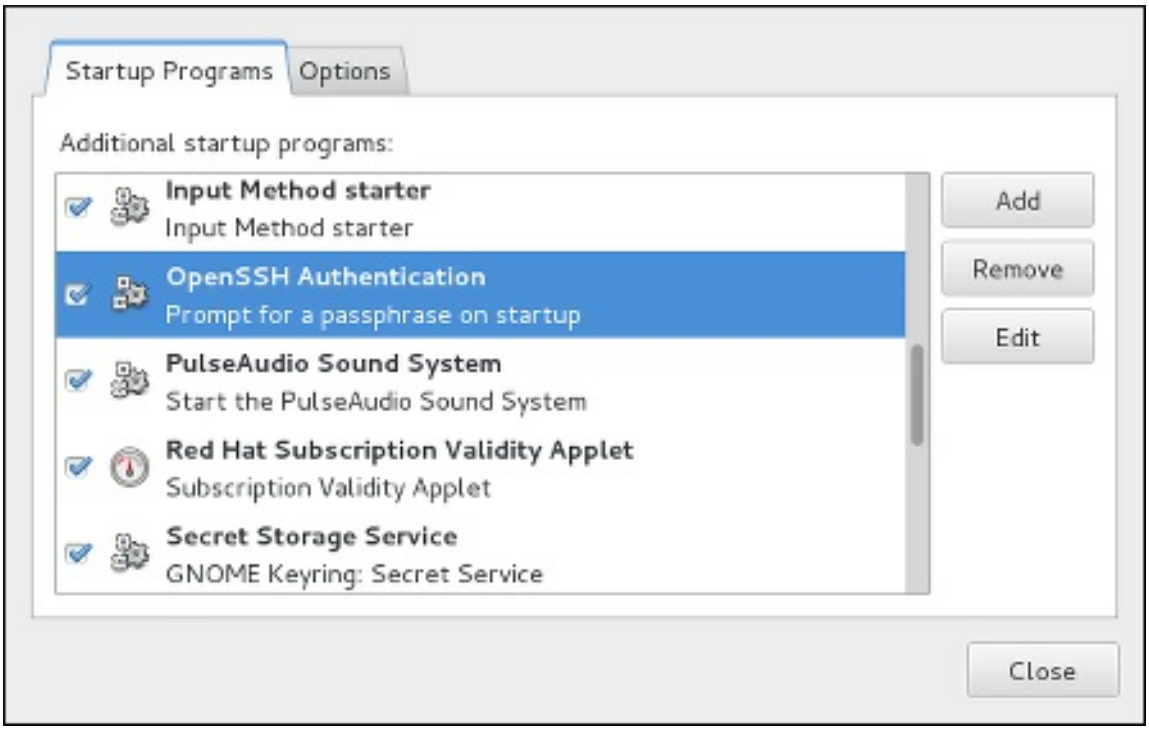
3. 单击右侧的 添加 按钮，然后在 Command 字段中输入 /usr/bin/ssh-add。

图 **12.2.** 添加新应用程序



4. 单击添加，并确保已选中新添加项目旁边的复选框。

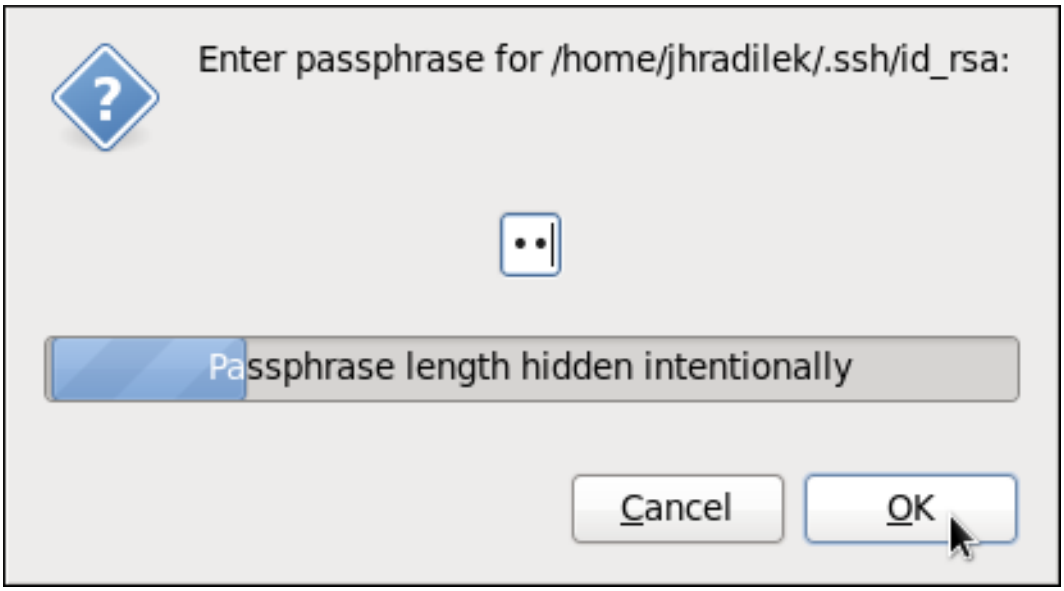
图 12.3. 启用应用程序



5.

注销，然后重新登录。系统将显示一个对话框，提示您输入密码短语。从此时开始，不会通过 ssh、scp 或 s ftp 提示您输入密码。

图 12.4. 输入密码



要为特定的 shell 提示符保存密码短语，请使用以下命令：

```
~]$ ssh-add
Enter passphrase for /home/USER/.ssh/id_rsa:
```

请注意，退出时会忘记您的密码短语。每次登录虚拟控制台或终端窗口时，都必须执行 命令。

12.3. OPENSSH 客户端

要从客户端机器连接到 OpenSSH 服务器，您必须安装 openssh-clients 软件包（有关如何在 Red Hat Enterprise Linux 中安装新软件包的更多信息，请参阅 第 9.2.4 节 “安装软件包”）。

12.3.1. 使用 ssh 实用程序

ssh 实用程序允许您登录远程计算机并在其中执行命令。它是 rlogin、rsh 和 telnet 程序的安全替代品。

与 telnet 命令类似，使用以下命令登录到远程机器：

```
ssh hostname
```

例如，要登录到名为 penguin .example.com 的远程机器，在 shell 提示符后输入以下内容：

```
~]$ ssh penguin.example.com
```

这将使用与本地计算机上使用的相同用户名登录。如果要指定不同的用户名，请使用以下格式的命令：

```
ssh username@hostname
```

例如，要以 USER 身份登录 penguin.example.com，请输入：

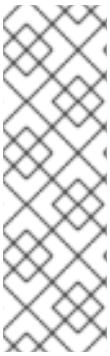
```
~]$ ssh USER@penguin.example.com
```

第一次启动连接时，您将会看到类似如下的消息：

```
The authenticity of host 'penguin.example.com' can't be established.
ECDSA key fingerprint is SHA256:vuGKK9dsW34zrZzwjl5g+vOE6EZQvHRQ8zObKYO2mW4.
ECDSA key fingerprint is MD5:7e:15:c3:03:4d:e1:dd:ee:99:dc:3e:f4:b9:67:6b:62.
Are you sure you want to continue connecting (yes/no)?
```

用户在回答此对话框中的问题之前，应始终检查指纹是否正确。用户可以要求服务器管理员确认密钥正确。这应该以安全的方式完成，并且先前已达成一致。如果用户可以访问服务器的主机密钥，可以使用 ssh-keygen 命令检查指纹，如下所示：

```
~]# ssh-keygen -l -f /etc/ssh/ssh_host_ecdsa_key.pub
SHA256:vuGKK9dsW34zrZzwjl5g+vOE6EZQvHRQ8zObKYO2mW4
```



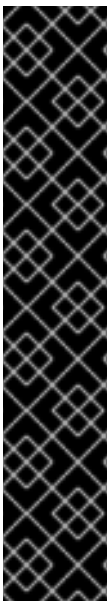
注意

要获得 MD5 密钥指纹（这是之前版本中的默认指纹），请使用带有 -E md5 选项的 ssh-keygen 命令，例如：

```
~]# ssh-keygen -l -f /etc/ssh/ssh_host_ecdsa_key.pub -EM md5
MD5:7e:15:c3:03:4d:e1:dd:ee:99:dc:3e:f4:b9:67:6b:62
```

键入 yes 以接受密钥并确认连接。您将看到一个通知，显示该服务器已添加到已知主机列表中，并看到一个提示您输入密码：

```
Warning: Permanently added 'penguin.example.com' (ECDSA) to the list of known hosts.
USER@penguin.example.com's password:
```



重要

如果 SSH 服务器的主机密钥发生更改，客户端会通知用户连接无法进行，直到从 ~/.ssh/known_hosts 文件中删除服务器的主机密钥为止。但是，在进行此操作前，请先联系 SSH 服务器的系统管理员来验证服务器是否未被破坏。

要从 ~/.ssh/known_hosts 文件中删除密钥，请使用以下命令：

```
~]# ssh-keygen -R penguin.example.com
# Host penguin.example.com found: line 15 type ECDSA
/home/USER/.ssh/known_hosts updated.
Original contents retained as /home/USER/.ssh/known_hosts.old
```

输入密码后，系统将为您提供用于远程计算机的 shell 提示符。

另外，ssh 程序 可用于在远程机器上执行命令，而无需登录到 shell 提示符：

```
ssh username@hostname command
```

例如，/etc/redhat-release 文件提供有关 Red Hat Enterprise Linux 版本的信息。要在 penguin.example.com 中查看此文件的内容，请输入：

```
~]$ ssh USER@penguin.example.com cat /etc/redhat-release
USER@penguin.example.com's password:
Red Hat Enterprise Linux Server release 7.0 (Maipo)
```

输入正确的密码后，将显示用户名，您将返回到本地 shell 提示符。

12.3.2. 使用 scp 实用程序

SCP 可用于通过安全、加密的连接在计算机之间传输文件。在设计中，它与 rcp 非常相似。

要将本地文件传输到远程系统，请使用以下格式的命令：

```
scp localfile username@hostname:remotefile
```

例如，如果要将 taglist.vim 传送到名为 penguin .example.com 的远程机器，在 shell 提示符后输入以下内容：

```
~]$ scp taglist.vim USER@penguin.example.com:.vim/plugin/taglist.vim
USER@penguin.example.com's password:
taglist.vim                   100% 144KB 144.5KB/s  00:00
```

可以同时指定多个文件.要将 .vim/plugin/ 的内容传输到远程机器 penguin .example.com 上的同一目录中，请输入以下命令：

```
~]$ scp .vim/plugin/* USER@penguin.example.com:.vim/plugin/
USER@penguin.example.com's password:
closetag.vim                  100%  13KB 12.6KB/s  00:00
snippetsEmu.vim               100%  33KB 33.1KB/s  00:00
taglist.vim                   100% 144KB 144.5KB/s  00:00
```

要将远程文件传输到本地系统，请使用以下语法：

```
scp username@hostname:remotefile localfile
```

例如，要从远程机器下载 .vimrc 配置文件，请输入：

```
~]$ scp USER@penguin.example.com:.vimrc .vimrc
USER@penguin.example.com's password:
.vimrc                    100% 2233   2.2KB/s   00:00
```

12.3.3. 使用 sftp 实用程序

sftp 实用程序可用于打开安全、交互式 FTP 会话。在其设计中，它类似于 ftp，除了使用安全加密的连接。

要连接到远程系统，请使用以下格式的命令：

```
sftp username@hostname
```

例如，使用用户名 USER 登录名为 penguin .example.com 的远程机器，请输入：

```
~]$ sftp USER@penguin.example.com
USER@penguin.example.com's password:
Connected to penguin.example.com.
sftp>
```

输入正确的密码后，您将会看到一个提示。sftp 实用程序接受一组与 ftp 所用命令类似的命令（请参阅 表 12.3 “系列可用的 sftp 命令”）。

表 12.3. 系列可用的 sftp 命令

命令	描述
ls [*目录*]	列出远程目录的内容.如果没有提供，则默认使用当前工作目录。
cd *directory*	将远程工作目录更改为 *目录*。
mkdir *directory*	创建远程目录.
rmdir *路径*	删除远程目录.
put *localfile* [*remotefile*]	将 *localfile* 传输到远程计算机。
get *remotefile* [*localfile*]	从远程计算机传输 *remotefile*.

有关可用命令的完整列表，请查看 sftp(1)手册页。

12.4. 更多安全 SHELL

安全命令行界面只是可以使用 SSH 的许多方式的开头。鉴于适当的带宽量，X11 会话可以通过 SSH 通道进行定向。或者，通过使用 TCP/IP 转发，系统之间的之前不安全的端口连接可以映射到特定的 SSH 频道。

12.4.1. X11 转发

要通过 SSH 连接打开 X11 会话，请使用以下格式的命令：

```
ssh -Y username@hostname
```

例如，使用用户名 USER 登录名为 penguin .example.com 的远程机器，请输入：

```
~]$ ssh -Y USER@penguin.example.com
USER@penguin.example.com's password:
```

当从安全 shell 提示符运行 X 程序时，SSH 客户端和服务器会创建一个新的安全频道，并且 X 程序数据通过该通道以透明方式发送到客户端计算机。

请注意，在进行 X11 转发前，必须在远程系统上安装 X Window 系统。以 root 用户身份输入以下命令安装 X11 软件包组：

```
~]# yum group install "X Window System"
```

有关软件包组的详情请参考 第 9.3 节 “使用软件包组”。

X11 转发非常有用。例如，可以使用 X11 转发来创建 Print Settings 实用程序的安全、交互式会话。要做到这一点，使用 ssh 连接到服务器并键入：

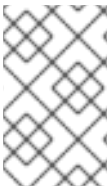
```
~]$ system-config-printer &
```

这时将显示 "打印设置" 工具，使远程用户能够安全地在远程系统上配置打印。

12.4.2. 端口转发

SSH 可以通过端口转发保护不安全的 TCP/IP 协议。使用这种技术时，SSH 服务器会变为 SSH 客户端的加密通道。

通过将客户端上的本地端口映射到服务器上的远程端口，端口转发发挥作用。SSH 可以将服务器中的任何端口映射到客户端上的任何端口。端口号不需要与此技术匹配。



注意

将端口转发设置为侦听低于 1024 的端口需要 root 级别访问权限。

要创建一个侦听 本地主机 上连接的 TCP/IP 端口转发频道，请使用以下格式的命令：

```
ssh -L local-port:remote-hostname:remote-port username@hostname
```

例如，要通过 加密连接检查名为 mail.example.com 的服务器上的电子邮件，请使用以下命令：

```
~]$ ssh -L 1100:mail.example.com:110 mail.example.com
```

在客户端机器和邮件服务器之间进行端口转发通道后，请指示 POP3 邮件客户端使用 本地主机 上的端口 1100 检查新电子邮件。发送到客户端系统上端口 1100 的任何请求将安全定向到 mail.example.com 服务器。

如果 mail.example.com 没有运行 SSH 服务器，但是同一网络上的另一台计算机为，那么仍可以使用 SSH 来保护连接的一部分。但是，需要一个略有不同的命令：

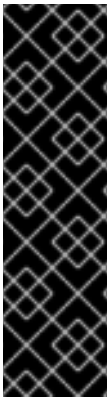
```
~]$ ssh -L 1100:mail.example.com:110 other.example.com
```

在本例中，来自客户端计算机上端口 1100 的 POP3 请求将通过端口 22 上的 SSH 连接转发到 SSH 服务器 other.example.com。然后，other.example.com 连接到 mail.example.com 上的端口 110 以检查新电子邮件。请注意，使用这种技术时，只有客户端系统与 other.example.com SSH 服务器之间的连接是安全的。

OpenSSH 套件还提供 UNIX 域套接字的本地和远程端口转发。要将 UNIX 域套接字通过网络转发到另一台机器，请使用 ssh -L local-socket:remote-socket username@hostname 命令，例如：

```
~]$ ssh -L /var/mysql/mysql.sock:/var/mysql/mysql.sock username@hostname
```

端口转发还可用于通过网络防火墙安全地获取信息。如果防火墙配置为允许 SSH 流量通过其标准端口（即端口 22）访问，但阻止对其他端口的访问，则使用受阻端口的两个主机之间的连接仍可通过通过已建立的 SSH 连接重定向来进行。



重要

以这种方式使用端口转发转发连接可让客户端系统上的任何用户连接到该服务。如果客户端系统被破坏，攻击者也有权访问转发的服务。

关注端口转发的系统管理员可以通过为 /etc/ssh/sshd_config 中的 AllowTcpForwarding 行指定 No 参数来禁用此功能，然后重新启动 sshd 服务。

12.5. 其它资源

有关如何在 Red Hat Enterprise Linux 中配置或连接到 OpenSSH 服务器的详情，请查看以下列出的资源。

安装的文档

- sshd(8)- sshd 守护进程的 man page 文档可用的命令行选项并提供支持的配置文件和目录的完整列表。
- ssh(1)- ssh 客户端应用的 man page 提供了可用命令行选项以及支持的配置文件和目录的完整列表。
- SCP(1)- scp 实用程序的 man page 提供了此实用程序及其用法的更详细描述。
- sftp(1)- sftp 实用程序的 man page。
- ssh-keygen(1)- ssh-keygen 实用程序文档的 man page 详细说明了如何使用它生成、管理和转换 ssh 使用的身份验证密钥。
- ssh_config(5)- 名为 ssh_config 的 man page 可记录可用的 SSH 客户端配置选项。

- sshd_config(5)- 名为 sshd_config 的 man page 提供了可用 SSH 守护进程配置选项的完整描述。

在线文档

- OpenSSH 主页 - OpenSSH 主页，包含更多文档、常见问题、邮件列表的链接、错误报告和其他有用资源。

- OpenSSL 主页 - OpenSSL 主页，包含更多文档、常见问题、邮件列表链接和其他有用资源。

另请参阅

- 第 6 章 获取特权 文档如何使用 su 和 sudo 命令获得管理权限。

- 第 10 章 使用 systemd 管理服务 提供有关 systemd 的更多信息，以及如何使用 systemctl 命令来管理系统服务。

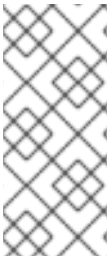
---

[1] 多路连接由通过共享、常见介质发送的多个信号组成。使用 SSH 时，不同的频道通过常见的安全连接发送。

第 13 章 TIGERVNC

tigervnc（网络虚拟网络计算）是一个图形桌面共享系统，可让您远程控制其他计算机。

tigervnc 遵循 client-server 原则：服务器共享其输出(vncserver)，客户端 (vncviewer)连接到服务器。



注意

与以前的 Red Hat Enterprise Linux 发行版不同，红帽企业 Linux 7 中的 TigerVNC 使用 systemd 系统管理守护进程进行配置。/etc/sysconfig/vncserver 配置文件已被 /etc/systemd/system/vncserver@.service 替代。

13.1. VNC 服务器

vncserver 是一个可启动 VNC（虚拟网络计算）桌面的实用程序。它使用适当的选项运行 Xvnc，并在 VNC 桌面上启动窗口管理器。vncserver 允许用户在一台计算机上并行运行独立的会话，然后任何数量的客户端可以从任何位置访问这些会话。

13.1.1. 安装 VNC 服务器

要安装 TigerVNC 服务器，以 root 用户身份运行以下命令：

```
~]# yum install tigervnc-server
```

13.1.2. 配置 VNC 服务器

VNC 服务器可以配置为为一个或多个用户启动显示，前提是系统中存在用户的帐户，以及显示设置、网络地址和端口以及安全设置等可选参数。

为单个用户配置 VNC 显示

1. 需要名为 /etc/systemd/system/vncserver@.service 的配置文件。要创建这个文件，以 root 用户身份复制 /usr/lib/systemd/system/vncserver@.service 文件：

```
~]# cp /usr/lib/systemd/system/vncserver@.service
/etc/systemd/system/vncserver@.service
```

不需要在文件名中包含显示号，因为 systemd 会自动根据需要 在内存中创建适当的命名实例，将服务文件中的 '%i' 替换为显示号。对于单个用户，无需重命名该文件。对于多个用户，每个用户都需要一个唯一命名的服务文件，例如，以某种方式将用户名添加到文件名中。详情请查看 第 13.1.2.1 节 “为两个用户配置 VNC 服务器”。

2. 编辑 /etc/systemd/system/vncserver@.service，将 USER 替换为实际用户名。不修改 文件的其余行。

```
ExecStart=/usr/bin/vncserver_wrapper <USER> %i
```

注意

VNC 桌面的默认大小为 1024x768。

可以使用 ~/.vnc/config 文件进一步配置用户的 VNC 会话。

例如，要更改 VNC 窗口大小，请添加以下行：

geometry= <WIDTH> x <HEIGHT>

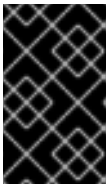
3. 保存更改。

4. 要使更改立即生效，请运行以下命令：

```
~]# systemctl daemon-reload
```

5. 设置 配置文件中定义的用户或用户的密码。请注意，您需要先从 root 切换到 USER。

```
~]# su - USER
~]$ vncpasswd
Password:
Verify:
```



重要

存储的密码未加密；有权访问密码文件的任何人都可以找到纯文本密码。

继续 第 13.1.3 节 “启动 VNC 服务器”。

13.1.2.1. 为两个用户配置 VNC 服务器

如果要在同一计算机上配置多个用户，请为每个用户创建不同的模板类型服务文件。

1. 创建两个服务文件，如 vncserver-USER_1@.service 和 vncserver-USER_2@.service。在这两个文件中，用正确的用户名替换 USER。

2. 为两个用户设置密码：

```
~]$ su - USER_1
~]$ vncpasswd
Password:
Verify:
~]$ su - USER_2
~]$ vncpasswd
Password:
Verify:
```

13.1.3. 启动 VNC 服务器

要启动并启用服务，请在 命令中直接指定显示号。以上 为单个用户配置 VNC 显示 中配置的文件作为模板运行，其中 %i 被 systemd 替换为显示号。使用有效的显示号，执行以下命令：

```
~]# systemctl start vncserver@:display_number.service
```

您还可以启用 服务在系统启动时自动启动。然后，当您登录时，vncserver 会自动启动。作为 root 用户，按以下方式发出命令：

```
~]# systemctl enable vncserver@:display_number.service
```

此时，其他用户可以使用 VNC viewer 程序使用定义的桌面号和密码连接到 VNC 服务器。如果安装了图形桌面，则将显示该桌面的实例。它与当前目标计算机上显示的实例不同。

13.1.3.1. 为两个用户配置 VNC 服务器，以及两个不同的显示器

对于两个配置的 VNC 服务器，vncserver-USER_1@.service 和 vncserver-USER_2@.service，您可以启用不同的显示号。例如，以下命令将导致 USER_1 的 VNC 服务器在桌面 3 中启动，以及 USER_2 的 VNC 服务器在桌面 5 中启动：

```
~]# systemctl start vncserver-USER_1@:3.service
~]# systemctl start vncserver-USER_2@:5.service
```

13.1.4. VNC 设置基于 GDM 的 XDMCP 的 xinetd

对于主要由瘦客户端组成的客户端系统，基于 X 显示管理器控制协议(XDMCP)的 xinetd 进行 VNC 设置是非常有用的设置。设置后，客户端可以访问 GDM 登录窗口并登录到任何系统帐户。设置的先决条件是安装 gdm、vnc、vnc -server & 和 xinetd 软件包。

```
~]# yum install gdm tigervnc tigervnc-server xinetd
```

必须启用服务 xinetd。

```
~]# systemctl enable xinetd.service
```

系统默认目标单元应为 graphical.target。要获得当前设置的默认目标单元，请使用：

```
~]# systemctl get-default
```

可以使用以下方法更改默认目标单元：

```
~]# systemctl set-default target_name
```

访问 GDM 登录窗口并登录

1. 通过编辑 /etc/gdm/custom.conf 配置文件将 GDM 设置为启用 XDMCP：

   ```
   [xdmcp]
   Enable=true
   ```

2. 创建名为 /etc/xinetd.d/xvncserver 的文件，其中包含以下内容：

```
service service_name
{
disable = no
protocol = tcp
socket_type = stream
wait = no
user = nobody
server = /usr/bin/Xvnc
server_args = -inetd -query localhost -once -geometry selected_geometry -depth
selected_depth securitytypes=none
}
```

在 server_args 部分中，-query localhost 选项将使每个 Xvnc 实例查询 localhost 以获取 xdmcp 会话。深度选项指定要创建的 VNC 桌面的像素深度（以位为单位）。可接受的值为 8、15、16 和 24 - 任何其它值都可能会导致应用程序的无法预计的行为。

3. 编辑文件 /etc/services 以定义服务。要做到这一点，请在 /etc/services 文件中附加以下片段：

```
# VNC xinetd GDM base
service_name 5950/tcp
```

4. 为确保配置更改生效，请重新启动 计算机。

或者，您可以执行以下操作：将 init 级别更改为 3 并返回到 5，以强制 gdm 重新加载。

```
# init 3
# init 5
```

验证 gdm 是否在侦听 UDP 端口 177。

```
# netstat -anu|grep 177
udp    0   0 0.0.0.0:177         0.0.0.0:*
```

重新启动 xinetd 服务。

```
~]# systemctl restart xinetd.service
```

验证 xinetd 服务是否已加载新服务。

```
# netstat -anpt|grep 595
tcp    0   0 :::5950           :::*            LISTEN   3119/xinetd
```

5. 使用 vncviewer 命令测试设置：

```
# vncviewer localhost:5950
```

命令将在未要求密码的本地主机上启动 VNC 会话。您将看到 GDM 登录屏幕，并且您可以使用有效的用户名和密码登录系统中的任何用户帐户。然后，您可以在远程连接上运行相同的测试。

为设置配置防火墙。运行防火墙配置工具并添加 TCP 端口 5950，以允许进入系统连接。

```
~]# firewall-cmd --permanent --zone=public --add-port=5950/tcp
~]# firewall-cmd --reload
```

13.1.5. 终止 VNC 会话

与启用 vncserver 服务类似，您可以禁用系统启动时自动启动该服务：

```
~]# systemctl disable vncserver@:display_number.service
```

或者，当您的系统运行时，您可以以 root 用户身份运行以下命令来停止该服务：

```
~]# systemctl stop vncserver@:display_number.service
```

13.2. 共享现有桌面

默认情况下，登录的用户在桌面 0 上提供 X Server 提供的桌面。用户可以使用 TigerVNC 服务器 x0vncserver 共享其桌面。

共享 X 桌面

要使用 x0vncserver 共享登录用户的桌面，请按如下操作：

1. 以 root用户身份输入以下命令

```
~]# yum install tigervnc-server
```

2. 为用户设置 VNC 密码：

```
~]$ vncpasswd
Password:
Verify:
```

3. 以该用户身份输入以下命令：

```
~]$ x0vncserver -PasswordFile=.vnc/passwd -AlwaysShared=1
```

如果防火墙配置为允许连接端口 5900，则远程查看器现在可以连接到显示 0，并查看已登录的用户桌面。有关如何配置防火墙的详情，请查看 第 13.3.2.1 节 “为 VNC 配置防火墙”。

13.3. VNC VIEWER

vncviewer 是一个显示图形用户界面并远程控制 vncserver 的程序。

对于运行 vncviewer，有一个弹出菜单，其中包含执行各种操作（例如切换至全屏模式或退出全屏模式）或退出查看器的条目。或者，您可以通过终端操作 vncviewer。在命令行中输入 vncviewer -h 以列出 vncviewer 的参数.

13.3.1. 安装 VNC Viewer

要安装 TigerVNC 客户端 vncviewer，以 root 用户身份运行以下命令：

```
~]# yum install tigervnc
```

13.3.2. 连接到 VNC 服务器

配置了 VNC 服务器后，您可以从任何 VNC viewer 连接到它。

使用 GUI 连接到 VNC 服务器

1. 输入不带参数的 vncviewer 命令，显示 VNC Viewer: Connection Details 实用程序。它将提示 VNC 服务器连接到。

2. 如果需要，要防止断开任何现有 VNC 连接到同一显示，请选择允许共享桌面的选项，如下所示：

   a. 选择"选项"按钮。

   b. 选择 Misc. 选项卡。

   c. 选择 共享 按钮。

   d. 按确定返回主菜单。

3. 输入要连接的地址和显示号：

   ```
   address:display_number
   ```

4. 按"连接 "以连接至 VNC 服务器显示。

5. 系统将提示您输入 VNC 密码。这将是与显示号对应的用户的 VNC 密码，除非设置了全局默认 VNC 密码。

   此时将显示显示 VNC 服务器桌面的窗口。请注意，这不是普通用户看到的桌面，而是 Xvnc 桌面。

使用 CLI 连接到 VNC 服务器

1. 输入带有地址并显示为参数的 viewer 命令：

   ```
   vncviewer address:display_number
   ```

   其中 address 是 IP 地址或主机名。

2. 通过输入 VNC 密码进行身份验证。这将是与显示号对应的用户的 VNC 密码，除非设置了全

局默认 VNC 密码。

3. 此时将显示显示 VNC 服务器桌面的窗口。请注意，这不是普通用户看到的桌面，而是 Xvnc 桌面。

13.3.2.1. 为 VNC 配置防火墙

使用非加密连接时，firewalld 可能会阻止连接。要允许 firewalld 传递 VNC 数据包，您可以打开 TCP 流量的特定端口。使用 -via 选项时，流量将通过 SSH 重定向，这在 firewalld 中默认启用。

注意

VNC 服务器的默认端口为 5900。要访问远程桌面的端口，请概述默认端口和用户分配的显示号。例如，第二个显示：2 + 5900 = 5902。

对于显示 0 到 3，请使用 service 选项使用 firewalld 对 VNC 服务的支持，如下所述。请注意，对于大于 3 的显示号，必须打开对应的端口，具体如 *在 firewalld 中打开端口* 中所述。

在 firewalld 中启用 VNC 服务

1. 运行以下命令，以查看有关 firewalld 设置的信息：

```
~]$ firewall-cmd --list-all
```

2. 要允许来自特定地址的所有 VNC 连接，请使用以下命令：

```
~]# firewall-cmd --add-rich-rule='rule family="ipv4" source address="192.168.122.116"
service name=vnc-server accept'
success
```

请注意，在系统下次启动后这些更改不会保留。要永久更改防火墙，请重复添加 --permanent 选项的命令。有关使用防火墙富语言命令的更多信息，请参阅红帽企业 Linux 7 安全指南。

3. 要验证上述设置，请使用以下命令：

```
~]# firewall-cmd --list-all
```

```
public (default, active)
 interfaces: bond0 bond0.192
 sources:
 services: dhcpv6-client ssh
 ports:
 masquerade: no
 forward-ports:
 icmp-blocks:
 rich rules:
 rule family="ipv4" source address="192.168.122.116" service name="vnc-server"
accept
```

要打开特定端口或端口范围，请使用 firewall -cmd 命令行工具的 --add- port 选项。例如，VNC 显示 4 要求为 TCP 流量打开端口 5904。

在 firewalld 中打开端口

1. 要在公共区中为 TCP 流量打开端口，以 root 身份发出命令，如下所示：

```
~]# firewall-cmd --zone=public --add-port=5904/tcp
success
```

2. 要查看公共区当前打开的端口，请使用以下命令：

```
~]# firewall-cmd --zone=public --list-ports
5904/tcp
```

可以使用 firewall-cmd --zone=zone --remove-port=number/protocol 命令删除端口。

请注意，在系统下次启动后这些更改不会保留。要永久更改防火墙，请重复添加 --permanent 选项的命令。有关在 firewalld 中打开和关闭端口的更多信息，请参阅 Red Hat Enterprise Linux 7 安全指南。

13.3.3. 使用 SSH 连接到 VNC 服务器

VNC 是明文网络协议，对通信中可能出现的攻击没有安全性。为确保通信安全，您可以使用 -via 选项加密您的 server-client 连接。这将在 VNC 服务器和客户端之间创建一个 SSH 隧道。

加密 VNC server-client 连接的命令格式如下：

```
vncviewer -via user@host:display_number
```

例 13.1. 使用 -via 选项

1. 要使用 SSH 连接到 VNC 服务器，请输入以下命令：

```
~]$ vncviewer -via USER_2@192.168.2.101:3
```

2. 系统提示时，键入密码并通过按 Enter 进行确认。

3. 在您的屏幕上会显示带有远程桌面的窗口。

限制 VNC 访问

如果您只想使用加密的连接，可以使用 systemd.service 文件中的 -localhost 选项完全防止未加密的连接，使用 ExecStart 行：

```
ExecStart=/usr/sbin/runuser -l user -c "/usr/bin/vncserver -localhost %i"
```

这将阻止 vncserver 接受来自本地主机之外的任何连接以及使用 SSH 发送的端口转发连接。

有关使用 SSH 的详情请参考 第 12 章 OpenSSH。

13.4. 其它资源

有关 TigerVNC 的更多信息，请参阅以下列出的资源。

安装的文档

- vncserver(1) - VNC 服务器实用程序的 man page。
- vncviewer(1) - VNC viewer 的 man page。
- vncpasswd(1) - VNC password 命令的 man page。

- ***Xvnc(1) - Xvnc 服务器配置选项的 man page。***

- ***x0vncserver(1) - TigerVNC 服务器的手册页用于共享现有 X 服务器。***

部分 V. 服务器

这部分讨论与服务器相关的各种主题，如如何设置 Web 服务器或通过网络共享文件和目录。

第 14 章 WEB 服务器

Web 服务器是一个通过 Web 向客户端提供内容的网络服务。这通常是网页，但也可以提供任何其他文档。Web 服务器也称为 HTTP 服务器，因为它们使用 超文本传输协议 (HTTP)。

Red Hat Enterprise Linux 7 中的 web 服务器包括：

- Apache HTTP 服务器
- nginx

重要

请注意，nginx web 服务器仅可用作 Red Hat Enterprise Linux 7 的软件集合。有关获取 nginx 访问权限、软件集合和其他信息，请参阅 Red Hat Software Collections 发行注记。

14.1. APACHE HTTP 服务器

本节重点介绍 Apache HTTP 服务器 2.4 httpd，这是由 Apache 软件基金会 开发的开源 Web 服务器。

如果您要从之前的 Red Hat Enterprise Linux 版本升级，则需要相应地更新 httpd 服务配置。本节回顾了一些新添加的功能，概述了 Apache HTTP 服务器 2.4 和版本 2.2 之间的重要更改，并指导您更新较旧的配置文件。

14.1.1. 显著变化

与 Red Hat Enterprise Linux 6 相比，Red Hat Enterprise Linux 7 中的 Apache HTTP 服务器有以下变化：

httpd 服务控制

在从 SysV 初始化脚本迁出后，服务器管理员应切换为使用 apachectl 和 systemctl 命令来控制服务（代替 service 命令）。以下示例专用于 httpd 服务。

该命令：

```
service httpd graceful
```

替换为

```
apachectl graceful
```

httpd 的 systemd 单元文件与初始化脚本有不同的行为，如下所示：

- 在重新加载服务时，默认使用正常重启。
- 服务停止时，默认使用正常停止。

  该命令：

  ```
  service httpd configtest
  ```

  替换为

  ```
  apachectl configtest
  ```

私有/tmp

为增强系统安全性，systemd 单元文件使用专用 /tmp 目录运行 httpd 守护进程，独立于系统的 /tmp 目录。

配置布局

加载模块的配置文件现在放在 /etc/httpd/conf.modules.d/ 目录中。为 httpd 提供其他可加载模块的软件包（如 php ）将文件放在此目录中。在 /etc/httpd/conf/httpd.conf 文件的主部分前面使用 Include 指令来包含 /etc/httpd/conf.modules.d/ 目录中的文件。这意味着在 httpd .conf 主体之前处理conf.modules. d/ 中的任何配置文件。httpd.conf 文件的末尾放置了针对 /etc/httpd/conf.d/ 目录中文件的 IncludeOptional 指令。这意味着 /etc/httpd/conf.d/ 中的文件现在在 httpd.conf 的主正文之后处理。

httpd 软件包本身提供一些额外的配置文件：

- /etc/httpd/conf.d/autoindex.conf - 这将配置 mod_autoindex 目录索引。

- /etc/httpd/conf.d/userdir.conf - 它配置对用户目录的访问，例如 http://example.com/~username/。由于安全原因，此类访问会被默认禁用。

- /etc/httpd/conf.d/welcome.conf - 如之前版本中，这会配置在没有内容时为 http://localhost/ 显示的欢迎页面。

默认配置

现在默认提供了一个最小的 httpd.conf 文件。默认配置中不再明确配置许多常用配置设置，如 Timeout 或 KeepAlive ；默认情况下，将使用硬编码的设置。所有配置指令的硬编码默认设置在 manual 中指定。如需更多信息，请参阅 “可安装文档”一节。

不兼容的语法变化

如果将现有配置从 httpd 2.2 迁移到 httpd 2.4，则需要对 httpd 配置语法进行向后兼容的一些更改。有关升级 http://httpd.apache.org/docs/2.4/upgrading.html 的详情，请查看以下 Apache 文档

处理模型

在以前的 Red Hat Enterprise Linux 版本中，提供了不同的 多处理模型 (MPM)作为不同的 httpd 二进制文件：分叉模型、"prefork"作为 /usr/sbin/httpd，以及基于线程的模式"worker"作为 /usr/sbin/httpd.worker。

在红帽企业 Linux 7 中，仅使用一个 httpd 二进制文件，三个 MPM 则作为可加载模块提供：worker、prefork（默认）和事件。根据需要编辑配置文件 /etc/httpd/conf.modules.d/00-mpm.conf，添加和删除注释字符 #，以便只加载三个 MPM 模块中的一个。

打包的更改

LDAP 身份验证和授权模块现在在单独的子软件包 mod_ldap 中提供。新模块 mod_session 和关联的帮助程序模块在新的子软件包 mod_session 中提供。新模块 mod_proxy_html 和 mod_xml2enc 在新的子软件包 mod_proxy_html 中提供。这些软件包都位于可选频道中。

注意

在订阅 Optional 和 Supplementary 频道前，请查看覆盖范围详情。如果您决定从这些频道安装软件包，请按照红帽客户门户网站中名为 How to access Optional 和 Supplementary 频道以及 -devel 软件包(RHSM)中的步骤进行操作。

打包文件系统布局

不再提供 /var/cache/mod_proxy/ 目录，而是使用 proxy 和 ssl 子目录打包 /var/cache/httpd/ 目录。

httpd 提供的打包内容已从 /var/www/ 移到 /usr/share/httpd/ ：

- */usr/share/httpd/icons/ - 包含一组用于目录索引的图标的目录（之前包含在 /var/wwww/icons/ 中） 已移到 /usr/share/httpd/icons/。在默认配置 http://localhost/icons/ 中可用；图标的位置和可用结构可在 /etc/httpd/conf.d/autoindex.conf 文件中进行配置。*

- */usr/share/httpd/manual/ - /var/www/manual/ 已移到 /usr/share/httpd/manual/。此目录包含在 httpd-manual 软件包中，包含 httpd 的 manual 的 HTML 版本。如果安装了该软件包，位于 http://localhost/manual/ 处可用，则在 /etc/httpd/conf.d/manual.conf 文件中可以配置手动的位置和可用性。*

- */usr/share/httpd/error/ - /var/www/error/ 已移至 /usr/share/httpd/error/。自定义多语言 HTTP 错误页面.默认情况下不配置，示例配置文件在 /usr/share/doc/httpd-VERSION/httpd-multilang-errordoc.conf 中提供。*

身份验证、授权和访问控制

用于控制身份验证、授权和访问控制的配置指令已显著变化。使用 Order、Deny 和 Allow 指令的现有配置文件应改编为使用新的 Require 语法。详情请查看以下 Apache 文档 http://httpd.apache.org/docs/2.4/howto/auth.html

suexec

为提高系统安全性，suexec 二进制文件不再安装，就像由 root 用户一样，而是设置文件系统功能位，从而允许一组更严格的权限。与此更改结合使用，suexec 二进制文件不再使用 /var/log/httpd/suexec.log 日志文件。相反，日志消息会发送到 syslog ；默认情况下，这些消息将显示在 /var/log/secure 日志文件中。

模块接口

由于 httpd 模块接口的更改，根据 httpd 2.2 构建的第三方二进制模块不兼容 httpd 2.4。对于 httpd 2.4 模块接口，需要根据需要调整这些模块，然后重新构建。有关版本 2.4 中 API 更改的详细列表，请参考 http://httpd.apache.org/docs/2.4/developer/new_api_2_4.html。

用于从源构建模块的 apxs 二进制文件已从 /usr/sbin/apxs 移到 /usr/bin/apxs。

删除的模块

在 Red Hat Enterprise Linux 7 中删除的 httpd 模块列表：

mod_auth_mysql, mod_auth_pgsql

httpd 2.4 在 mod_authn_dbd 模块内部提供 SQL 数据库身份验证支持。

mod_perl

上游 httpd 2.4 不支持 mod_perl。

mod_authz_ldap

httpd 2.4 使用 mod_ authnz_ldap 在子软件包 mod_ldap 中提供 LDAP 支持。

14.1.2. 更新配置

要从 Apache HTTP Server 版本 2.2 更新配置文件，请执行以下步骤：

1. 确保所有模块名称都正确，因为它们可能已更改。调整已重命名的每个模块的 LoadModule 指令。

2. 重新编译所有第三方模块，然后尝试载入它们。这通常意味着身份验证和授权模块。

3. 如果您使用 mod_userdir 模块，请确保提供了指明目录名称（通常是 public_html）的 UserDir 指令。

4. 如果您使用 Apache HTTP 安全服务器，请参阅 第 14.1.8 节 “启用 mod_ssl 模块” 来获得有关启用安全套接字层(SSL)协议的重要信息。

请注意，您可以使用以下命令检查配置中可能存在的错误：

```
~]# apachectl configtest
Syntax OK
```

有关将 Apache HTTP 服务器配置从 2.2 升级到 2.4 的更多信息，请参阅 http://httpd.apache.org/docs/2.4/upgrading.html。

14.1.3. 运行 httpd 服务

这部分论述了如何启动、停止、重新启动和检查 Apache HTTP 服务器的当前状态。若要能够使用 httpd 服务，请确保已安装了 httpd。您可以使用以下命令完成此操作：

```
~]# yum install httpd
```

有关目标概念以及如何在 Red Hat Enterprise Linux 中管理系统服务的详情请参考 第 10 章 使用 systemd 管理服务。

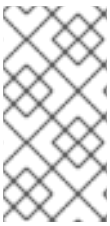
14.1.3.1. 启动服务

要运行 httpd 服务，以 root 用户身份在 shell 提示符后输入以下内容：

```
~]# systemctl start httpd.service
```

如果您希望该服务在引导时自动启动，请使用以下命令：

```
~]# systemctl enable httpd.service
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to
/usr/lib/systemd/system/httpd.service.
```



注意

如果将 Apache HTTP 服务器作为安全服务器运行，则计算机启动后可能需要密码（如果使用加密的私钥 SSL 密钥）。

14.1.3.2. 停止服务

要停止正在运行的 httpd 服务，以 root 用户身份在 shell 提示符后输入以下内容：

```
~]# systemctl stop httpd.service
```

要防止服务在引导时自动启动，请输入：

```
~]# systemctl disable httpd.service
Removed symlink /etc/systemd/system/multi-user.target.wants/httpd.service.
```

14.1.3.3. 重启服务

重启正在运行的 httpd 服务有三种不同的方法：

1. 要完全重启该服务，以 root 用户身份输入以下命令：

```
~]# systemctl restart httpd.service
```

这将停止正在运行的 httpd 服务，并立即重新启动它。安装或删除动态加载的模块（如 PHP）后，请使用此命令。

2. 要只重新载入配置，以 root 用户身份输入：

```
~]# systemctl reload httpd.service
```

这会导致正在运行的 httpd 服务重新加载其配置文件。当前处理的任何请求都将中断，这可能会导致客户端浏览器显示错误消息或呈现部分页面。

3. 要在不影响活跃请求的情况下重新载入配置，以 root 用户身份输入以下命令：

```
~]# apachectl graceful
```

这会导致正在运行的 httpd 服务重新加载其配置文件。当前处理的任何请求都将继续使用旧配置。

有关如何在 Red Hat Enterprise Linux 7 中管理系统服务的详情请参考 第 10 章 使用 systemd 管理服务。

14.1.3.4. 验证服务状态

要验证 httpd 服务是否正在运行，在 shell 提示符后输入以下内容：

```
~]# systemctl is-active httpd.service
active
```

14.1.4. 编辑配置文件

当 httpd 服务启动时，它默认从 表 14.1 “httpd 服务配置文件” 中列出的位置读取配置。

表 14.1. httpd 服务配置文件

路径	描述
/etc/httpd/conf/httpd.conf	主配置文件。
/etc/httpd/conf.d/	主配置文件中包含的配置文件的辅助目录。

虽然默认配置应该适合大多数情况，但最好至少熟悉一些更重要的配置选项。请注意，要使任何更改生效，必须首先重新启动 Web 服务器。有关如何重启 httpd 服务的更多信息，请参阅 第 14.1.3.3 节 “重启服务”。

要检查配置中的可能错误，在 shell 提示符后输入以下内容：

```
~]# apachectl configtest
Syntax OK
```

为了更容易从错误中恢复，建议您先复制原始文件的副本，然后再编辑该文件。

14.1.5. 使用模块

作为模块化应用，httpd 服务与多个 动态共享对象(DSO)一起分发，可以根据需要在运行时动态加载或卸载。在 Red Hat Enterprise Linux 7 中，这些模块位于 /usr/lib64/httpd/modules/ 中。

14.1.5.1. 加载模块

若要加载特定的 DSO 模块，可使用 LoadModule 指令。请注意，由单独的包提供的模块通常在 /etc/httpd/conf.d/ 目录中具有自己的配置文件。

例 14.1. 载入 mod_ssl DSO

```
LoadModule ssl_module modules/mod_ssl.so
```

完成后，重新启动 Web 服务器以重新加载配置。有关如何重启 httpd 服务的更多信息，请参阅 第 14.1.3.3 节 “重启服务”。

14.1.5.2. 编写模块

如果您打算创建新的 DSO 模块，请确保已安装了 httpd-devel 软件包。要做到这一点，以 root 用户身份输入以下命令：

```
~]# yum install httpd-devel
```

此软件包包含编译模块所需的 include 文件、标题文件和 APache eXtenSion (apxs)实用程序。

编写完成后，可以使用以下命令构建模块：

```
~]# apxs -i -a -c module_name.c
```

如果构建成功，您应该能够像通过 Apache HTTP 服务器分发的任何其他模块一样加载该模块。

14.1.6. 设置虚拟主机

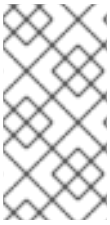
Apache HTTP 服务器内置的虚拟主机允许服务器根据请求的 IP 地址、主机名或端口提供不同的信息。

若要创建基于名称的虚拟主机，请将示例配置文件 /usr/share/doc/httpd-VERSION/httpd-vhosts.conf 复制到 / etc/httpd/conf.d/ 目录中，并替换 @@Port@@ 和 @@ ServerRoot@ 占位符值。根据 例 14.2 “虚拟主机配置示例” 中所示的要求自定义选项。

例 14.2. 虚拟主机配置示例

```
<VirtualHost *:80>
  ServerAdmin webmaster@penguin.example.com
  DocumentRoot "/www/docs/penguin.example.com"
  ServerName penguin.example.com
  ServerAlias www.penguin.example.com
  ErrorLog "/var/log/httpd/dummy-host.example.com-error_log"
  CustomLog "/var/log/httpd/dummy-host.example.com-access_log" common
</VirtualHost>
```

请注意，ServerName 必须是分配给机器的有效 DNS 名称。<VirtualHost> 容器可高度自定义，并接受主服务器配置中提供的大部分指令。此容器中 不支持 的指令包括由 SuexecUserGroup 替代的 用户和组 。



注意

如果您将虚拟主机配置为侦听非默认端口，请确保相应地更新 /etc/httpd/conf/httpd.conf 文件的全局 settings 部分中的 Listen 指令。

要激活新创建的虚拟主机，必须首先重新启动 Web 服务器。有关如何重启 httpd 服务的更多信息，请参阅 第 14.1.3.3 节 “重启服务”。

14.1.7. 设置 SSL 服务器

安全套接字层 (SSL)是一种加密协议，允许服务器和客户端安全地通信。除了称为传输层安全性(TLS)的扩展和改进版本外，它还确保了隐私和数据完整性。Apache HTTP 服务器与 mod_ssl 相结合，此模块使用 OpenSSL 工具包来提供 SSL/TLS 支持，通常称为 SSL 服务器。Red Hat Enterprise Linux 还支持将 Mozilla NSS 用作 TLS 实现。对 Mozilla NSS 的支持由 mod_nss 模块提供。

与能够拦截它的 HTTP 连接的 HTTP 连接不同，通过 HTTP（称为 HTTPS）使用 SSL/TLS 会阻止对传输的内容进行任何检查或修改。本节提供有关如何在 Apache HTTP 服务器配置中启用此模块的基本信息，并指导您完成生成私钥和自签名证书的过程。

14.1.7.1. 证书和安全概述

安全通信基于密钥的使用。在传统对称加密中，交易两端都有相同的密钥来解码彼此的传输。另一方面，在公钥或非对称加密中 ，两个密钥并存：一个保存机密的私钥，也是通常与公钥共享的公钥。虽然使用公钥编码的数据只能使用私钥进行解码，但使用私钥编码的数据反过来只能使用公钥解码。

若要使用 SSL 提供安全通信，SSL 服务器必须使用证书颁发机构(CA)签署的数字证书。证书列出了服务器的各种属性（即服务器主机名、公司名称、位置等），以及使用 CA 的私钥生成的签名。此签名可确保特定证书认证机构已签署证书，并且证书没有被以任何方式修改。

Web 浏览器建立新的 SSL 连接时，它将检查 Web 服务器提供的证书。如果证书没有来自可信 CA 的签名，或者证书中列出的主机名与用于建立连接的主机名不匹配，则它将拒绝与服务器通信，并且通常向用户提供相应的错误消息。

默认情况下，大多数 Web 浏览器都配置为信任一组广泛使用的证书颁发机构。因此，设置安全服务器时应选择适当的 CA，以便目标用户信任连接，否则会显示错误消息，并且必须手动接受证书。由于鼓励用户覆盖证书错误可能会允许攻击者拦截连接，所以您应该尽可能使用信任的 CA。详情请查看 表 14.2 “有关常用 Web 浏览器使用的 CA 列表的信息”。

表 14.2. 有关常用 Web 浏览器使用的 CA 列表的信息

Web 浏览器	link
Mozilla Firefox	Mozilla root CA 列表.
opera	有关 Opera 使用的 root 证书的信息。
Internet Explorer	Microsoft Windows 使用的 root 证书的信息。
chromium	有关 Chromium 项目使用的根证书的信息。

在设置 SSL 服务器时，您需要生成证书请求和私钥，然后发送证书请求、公司身份证明以及向证书颁发机构付款。当 CA 验证证书请求和您的身份后，它将向您发送签名的证书，您可以将其用于您的服务器。或者，您可以创建一个不包含 CA 签名的自签名证书，因此只用于测试目的。

14.1.8. 启用 mod_ssl 模块

如果您打算使用 mod_ssl 设置 SSL 或 HTTPS 服务器，则无法具有另一个应用或模块，如 mod_nsss 配置为使用同一端口。端口 443 是 HTTPS 的默认端口。

要使用 mod_ssl 模块和 OpenSSL 工具包设置 SSL 服务器，请安装 mod_ssl 和 openssl 软件包。以 root 用户身份输入以下命令：

```
~]# yum install mod_ssl openssl
```

这将在 /etc/httpd/conf.d/ssl.conf 中创建 mod_ ssl 配置文件，默认包含在 Apache HTTP 服务器主配置文件中。要加载模块，请重新启动 httpd 服务，如 第 14.1.3.3 节 “重启服务” 所述。

重要

由于 POODLE 中描述的漏洞：SSLv3 漏洞(CVE-2014-3566)，红帽建议禁用 SSL，仅使用 TLSv1.1 或 TLSv1.2。可以使用 TLSv1.0 实现向后兼容性。许多红帽支持的产品都能够使用 SSLv2 或 SSLv3 协议，或者默认启用它们。但是，现在强烈建议使用 SSLv2 或 SSLv3。

14.1.8.1. 在 mod_ssl 中启用和禁用 SSL 和 TLS

要禁用并启用 SSL 和 TLS 协议的特定版本，可在全局范围内添加配置文件的" # SSL 全局上下文"部分中的 SSLProtocol 指令，并在所有其他位置将其删除，或者在所有"VirtualHost"部分中编辑"SSL 协议支持"下的默认条目。如果您没有在每个域 VirtualHost 部分中指定，它将继承 global 部分中的设置。

为确保协议版本被禁用，管理员应仅在 "SSL Global Context"部分中指定 SSLProtocol，或者 在所有域 VirtualHost 部分中指定。

禁用 SSLv2 和 SSLv3

要禁用 SSL 版本 2 和 SSL 版本 3，这意味着在所有 VirtualHost 部分中启用 SSL 版本 2 和 SSL 版本 3 之外的所有内容，如下方所示：

1. 以 root 身份，打开 /etc/httpd/conf.d/ssl.conf 文件并搜索 SSLProtocol 指令 的所有 实例。默认情况下，配置文件包含一个如下所示的部分：

```
~]# vi /etc/httpd/conf.d/ssl.conf
#  SSL Protocol support:
# List the enable protocol levels with which clients will be able to
# connect. Disable SSLv2 access by default:
SSLProtocol all -SSLv2
```

本节位于 VirtualHost 部分中。

2. 编辑 SSLProtocol 行，如下所示：

```
#  SSL Protocol support:
# List the enable protocol levels with which clients will be able to
# connect. Disable SSLv2 access by default:
SSLProtocol all -SSLv2 -SSLv3
```

对所有 VirtualHost 部分重复此操作。保存并关闭该文件。

3. 验证所有出现 SSLProtocol 指令都已更改，如下所示：

```
~]# grep SSLProtocol /etc/httpd/conf.d/ssl.conf
SSLProtocol all -SSLv2 -SSLv3
```

如果您有一个以上的默认 VirtualHost 部分，则这一步尤为重要。

4. 重启 Apache 守护进程，如下所示：

```
~]# systemctl restart httpd
```

请注意，任何会话都将中断。

禁用所有 SSL 和 TLS 协议，接受 TLS 1 并启动

要禁用除 TLS 版本 1 及更高版本以外的所有 SSL 和 TLS 协议版本，请按如下操作：

1. 以 root 身份，打开 /etc/httpd/conf.d/ssl.conf 文件并搜索 所有 SSLProtocol 指令实例。默认情况下，该文件包含如下部分：

```
~]# vi /etc/httpd/conf.d/ssl.conf
#  SSL Protocol support:
# List the enable protocol levels with which clients will be able to
# connect. Disable SSLv2 access by default:
SSLProtocol all -SSLv2
```

2. 编辑 SSLProtocol 行，如下所示：

```
#  SSL Protocol support:
# List the enable protocol levels with which clients will be able to
# connect. Disable SSLv2 access by default:
SSLProtocol -all +TLSv1 +TLSv1.1 +TLSv1.2
```

保存并关闭该文件。

3. 按如下所示验证更改：

```
~]# grep SSLProtocol /etc/httpd/conf.d/ssl.conf
SSLProtocol -all +TLSv1 +TLSv1.1 +TLSv1.2
```

4. 重启 Apache 守护进程，如下所示：

```
~]# systemctl restart httpd
```

请注意，任何会话都将中断。

测试 SSL 和 TLS 协议的状态

要检查哪些版本的 SSL 和 TLS 已启用或禁用，请使用 openssl s_client -connect 命令。该命令具有以下格式：

```
openssl s_client -connect hostname:port -protocol
```

其中端口是要测试的端口，协议是要测试的协议版本。若要测试本地运行的 SSL 服务器，请使用 localhost 作为主机名。例如：要测试安全 HTTPS 连接的默认端口，端口 443 查看是否启用 SSLv3，请发出以下命令：

```
~]# openssl s_client -connect localhost:443 -ssl3
CONNECTED(00000003)
139809943877536:error:14094410:SSL routines:SSL3_READ_BYTES:sslv3 alert handshake
failure:s3_pkt.c:1257:SSL alert number 40
139809943877536:error:1409E0E5:SSL routines:SSL3_WRITE_BYTES:ssl handshake
failure:s3_pkt.c:596:
output omitted
New, (NONE), Cipher is (NONE)
Secure Renegotiation IS NOT supported
Compression: NONE
Expansion: NONE
SSL-Session:
  Protocol : SSLv3
output truncated
```

以上输出表明握手失败，因此不会协商任何密码。

```
~]$ openssl s_client -connect localhost:443 -tls1_2
CONNECTED(00000003)
depth=0 C = --, ST = SomeState, L = SomeCity, O = SomeOrganization, OU =
SomeOrganizationalUnit, CN = localhost.localdomain, emailAddress =
root@localhost.localdomain
output omitted
New, TLSv1/SSLv3, Cipher is ECDHE-RSA-AES256-GCM-SHA384
Server public key is 2048 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
SSL-Session:
  Protocol : TLSv1.2
output truncated
```

以上输出表明没有发生握手失败，并且协商一组密码。

openssl s_client 命令选项记录在 s_client(1) 手册页中。

有关 SSLv3 漏洞以及如何对其进行测试的更多信息，请参阅红帽知识库文章 POODLE：SSLv3 漏洞(CVE-2014-3566)。

14.1.9. 启用 mod_nss 模块

如果您打算使用 mod_nss 设置 HTTPS 服务器，则无法 安装 mod_ssl 软件包，其默认设置为 mod_ssl，默认情况下 将使用端口 443，但这是默认的 HTTPS 端口。如果可能，删除 软件包。

要删除 mod_ssl，以 root 用户身份输入以下命令：

```
~]# yum remove mod_ssl
```

注意

如果需要 mod_ssl 用于其他目的，请修改 /etc/httpd/conf.d/ssl.conf 文件，以使用 443 之外的端口来防止 mod_ssl 在端口被更改为 443 时与 mod_nss 冲突。

只有一个模块可以拥有一个端口，因此 mod_nss 和 mod_ssl 只能同时共存（如果它们使用唯一端口）。因此，默认情况下 mod_nss 使用 8443，但 HTTPS 的默认端口是端口 443。端口由 Listen 指令以及 VirtualHost 名称或地址指定。

NSS 中的所有内容都与"令牌"关联。软件令牌存在于 NSS 数据库中，但您也可以有一个包含证书的物理令牌。使用 OpenSSL 时，离散证书和私钥保存在 PEM 文件中。对于 NSS，这些内容存储在数据库中。每个证书和密钥都与令牌关联，每个令牌都可以有密码保护。此密码是可选的，但是如果使用密码，则 Apache HTTP 服务器需要该密码的副本才能在系统启动时无需用户干预即可打开数据库。

配置 mod_nss

1. 以 root 用户身份安装 mod_nss:

   ```
   ~]# yum install mod_nss
   ```

   这将在 /etc/httpd/conf.d/ nss.conf 中创建 mod_ nss 配置文件。默认情况下，/etc/httpd/conf.d/ 目录包含在 Apache HTTP 服务器主配置文件中。要加载模块，请重新启动 httpd 服务，如 第 14.1.3.3 节 “重启服务” 所述。

2. 以 root 身份，打开 /etc/httpd/conf.d/nss.conf 文件并搜索 Listen 指令 的所有 实例。

编辑 **Listen 8443 行，如下所示：**

```
Listen 443
```

端口 443 是 HTTPS 的默认端口。

3. 编辑**默认 VirtualHost default: 8443 行，如下所示：**

```
VirtualHost default:443
```

编辑**任何其他非默认虚拟主机部分（若存在）。保存并关闭该文件。**

4. **Mozilla NSS 将证书存储在 /etc/httpd/conf.d/nss.conf 文件中的 NSSCertificateDatabase 指令表示的服务器证书数据库中。默认情况下，路径设置为 /etc/httpd/alias，这是在安装过程中创建的 NSS 数据库。**

要查看默认 NSS 数据库，请使用以下命令：

```
~]# certutil -L -d /etc/httpd/alias

Certificate Nickname                 Trust Attributes
                                     SSL,S/MIME,JAR/XPI

cacert                               CTu,Cu,Cu
Server-Cert                          u,u,u
alpha                                u,pu,u
```

在以上命令输出中，Server-Cert 是默认的 NSSNickname。L 选项会在证书数据库中列出所有证书，或者显示有关指定证书的信息。d 选项指定包含证书和密钥数据库文件的数据库目录。有关更多命令行选项，请参阅 certutil(1) man page。

5. **要将 mod_nss 配置为使用其他数据库，请编辑 /etc/httpd/conf.d/nss.conf 文件中的 NSSCertificateDatabase 行。默认文件在 VirtualHost 部分中具有以下行：**

```
#  Server Certificate Database:
#  The NSS security database directory that holds the certificates and
#  keys. The database consists of 3 files: cert8.db, key3.db and secmod.db.
#  Provide the directory that these files exist.
NSSCertificateDatabase /etc/httpd/alias
```

在以上命令输出中，alia 是默认 NSS 数据库目录 /etc/httpd/alias/。

6. 要将密码应用到默认 NSS 证书数据库，以 root 用户身份运行以下命令：

```
~]# certutil -W -d /etc/httpd/alias
Enter Password or Pin for "NSS Certificate DB":
Enter a password which will be used to encrypt your keys.
The password should be at least 8 characters long,
and should contain at least one non-alphabetic character.

Enter new password:
Re-enter password:
Password changed successfully.
```

7. 在部署 HTTPS 服务器之前，请使用由证书颁发机构(CA)签名的证书创建新证书数据库。

例 14.3. 在 Mozilla NSS 数据库中添加证书

certutil 命令用于在 NSS 数据库文件中添加 CA 证书：

```
certutil -d /etc/httpd/nss-db-directory/ -A -n "CA_certificate" -t CT,, -a -i
certificate.pem
```

以上命令添加存储在名为 certificate.pem 的 PEM 格式文件中的 CA 证书。d 选项指定包含证书和密钥数据库文件的 NSS 数据库目录，-n 选项设置证书的名称 -t CT， 表示证书受信任可用于 TLS 客户端和服务器。A 选项将 现有证书添加到证书数据库。如果数据库不存在，它将被创建。a 选项 允许将 ASCII 格式用于输入或输出，而 -i 选项则将 certificate.pem 输入文件传递到 命令。

有关更多命令行选项，请参阅 certutil(1) man page。

8. NSS 数据库应设有密码保护，以保护私钥。

例 14.4. 为 Mozilla NSS 数据库设置密码

certutil 工具可以为 NSS 数据库设置密码，如下所示：

```
certutil -W -d /etc/httpd/nss-db-directory/
```

例如，对于默认数据库，以 root 身份发出命令，如下所示：

```
~]# certutil -W -d /etc/httpd/alias
Enter Password or Pin for "NSS Certificate DB":
Enter a password which will be used to encrypt your keys.
The password should be at least 8 characters long,
and should contain at least one non-alphabetic character.

Enter new password:
Re-enter password:
Password changed successfully.
```

9. 通过使用 NSS PassPhraseDialog 指令更改行，将 mod_nss 配置为使用 NSS 内部软件令牌：

```
~]# vi /etc/httpd/conf.d/nss.conf
NSSPassPhraseDialog file:/etc/httpd/password.conf
```

这是为了避免在系统启动时手动输入密码。软件令牌存在于 NSS 数据库中，但您也可以拥有包含您的证书的物理令牌。

10. 如果 NSS 数据库中包含的 SSL 服务器证书是 RSA 证书，请确保未注释 NSSNickname 参数并与上面第 4 步中显示的 nickname 匹配：

```
~]# vi /etc/httpd/conf.d/nss.conf
NSSNickname Server-Cert
```

如果 NSS 数据库中包含的 SSL 服务器证书是 ECC 证书，请确保 NSSECCNickname 参数未注释，并与上一步 4 中显示的 nickname 匹配：

```
~]# vi /etc/httpd/conf.d/nss.conf
NSSECCNickname Server-Cert
```

确保 NSSCertificateDatabase 参数未注释，并指向在第 4 步中显示的 NSS 数据库目录，或者在上面的步骤 5 中配置：

```
~]# vi /etc/httpd/conf.d/nss.conf
NSSCertificateDatabase /etc/httpd/alias
```

将 /etc/httpd/alias 替换为要使用的证书数据库的路径。

11. 以 root 用户身份创建 /etc/httpd/password.conf 文件：

```
~]# vi /etc/httpd/password.conf
```

使用以下格式添加一行：

```
internal:password
```

使用上面步骤 6 中应用到 NSS 安全数据库的密码替换 password。

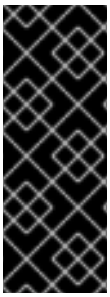
12. 将适当的所有权和权限应用到 /etc/httpd/password.conf 文件：

```
~]# chgrp apache /etc/httpd/password.conf
~]# chmod 640 /etc/httpd/password.conf
~]# ls -l /etc/httpd/password.conf
-rw-r-----. 1 root apache 10 Dec 4 17:13 /etc/httpd/password.conf
```

13. 要将 mod_nss 配置为使用 /etc/httpd/password.conf 中的软件令牌，请按如下方式编辑 /etc/httpd/conf.d/nss.conf ：

```
~]# vi /etc/httpd/conf.d/nss.conf
```

14. 重新启动 Apache 服务器以使更改生效，如 所述 第 14.1.3.3 节 “重启服务”



重要

由于 POODLE 中描述的漏洞：SSLv3 漏洞(CVE-2014-3566)，红帽建议禁用 SSL，仅使用 TLSv1.1 或 TLSv1.2。可以使用 TLSv1.0 实现向后兼容性。许多红帽支持的产品都能够使用 SSLv2 或 SSLv3 协议，或者默认启用它们。但是，现在强烈建议使用 SSLv2 或 SSLv3。

14.1.9.1. 在 mod_nss 中启用和禁用 SSL 和 TLS

要禁用并启用 SSL 和 TLS 协议的特定版本，或者通过在配置文件的"# SSL 全局上下文"部分中添加 NSSProtocol 指令并在其它位置将其删除，或者在所有"VirtualHost"部分中编辑"SSL 协议"下的默认条

目。如果您没有在每个域 VirtualHost 部分中指定，它将继承 global 部分中的设置。为确保协议版本被禁用，管理员应该 仅在 "SSL Global Context"部分中指定 NSSProtocol，或者 在所有 域 VirtualHost 部分中指定它。

在 mod_nss 中禁用所有 SSL 和 TLS 协议覆盖 TLS 1 和 Up

要禁用除 TLS 版本 1 及更高版本以外的所有 SSL 和 TLS 协议版本，请按如下操作：

1. 以 root 身份，打开 /etc/httpd/conf.d/nss.conf 文件并搜索 NSSProtocol 指令 的所有 实例。默认情况下，配置文件包含一个如下所示的部分：

```
~]# vi /etc/httpd/conf.d/nss.conf
#  SSL Protocol:
output omitted
#  Since all protocol ranges are completely inclusive, and no protocol in the
#  middle of a range may be excluded, the entry "NSSProtocol SSLv3,TLSv1.1"
#  is identical to the entry "NSSProtocol SSLv3,TLSv1.0,TLSv1.1".
NSSProtocol SSLv3,TLSv1.0,TLSv1.1
```

本节位于 VirtualHost 部分中。

2. 编辑 NSSProtocol 行，如下所示：

```
#  SSL Protocol:
NSSProtocol TLSv1.0,TLSv1.1
```

对所有 VirtualHost 部分重复此操作。

3. 编辑 Listen 8443 行，如下所示：

```
Listen 443
```

4. 编辑默认 VirtualHost default: 8443 行，如下所示：

```
VirtualHost default:443
```

编辑任何其他非默认虚拟主机部分（若存在）。保存并关闭该文件。

5. 验证所有 NSSProtocol 指令都已更改，如下所示：

```
~]# grep NSSProtocol /etc/httpd/conf.d/nss.conf
#  middle of a range may be excluded, the entry "NSSProtocol SSLv3,TLSv1.1"
#  is identical to the entry "NSSProtocol SSLv3,TLSv1.0,TLSv1.1".
NSSProtocol TLSv1.0,TLSv1.1
```

如果您有多个 VirtualHost 部分，这一步尤为重要。

6. 重启 Apache 守护进程，如下所示：

```
~]# service httpd restart
```

请注意，任何会话都将中断。

测试 mod_nss 中的 SSL 和 TLS 协议的状态

要检查在 mod_nss 中启用或禁用了 SSL 和 TLS 版本，请使用 openssl s_client -connect 命令。以 root 用户身份安装 openssl 软件包：

```
~]# yum install openssl
```

openssl s_client -connect 命令有以下格式：

```
openssl s_client -connect hostname:port -protocol
```

其中端口是要测试的端口，协议是要测试的协议版本。若要测试本地运行的 SSL 服务器，请使用 localhost 作为主机名。例如：要测试安全 HTTPS 连接的默认端口，端口 443 查看是否启用 SSLv3，请发出以下命令：

```
~]# openssl s_client -connect localhost:443 -ssl3
CONNECTED(00000003)
3077773036:error:1408F10B:SSL routines:SSL3_GET_RECORD:wrong version
number:s3_pkt.c:337:
output omitted
New, (NONE), Cipher is (NONE)
Secure Renegotiation IS NOT supported
Compression: NONE
Expansion: NONE
```

```
SSL-Session:
  Protocol : SSLv3
output truncated
```

以上输出表明握手失败，因此不会协商任何密码。

```
~]$ openssl s_client -connect localhost:443 -tls1
CONNECTED(00000003)
depth=1 C = US, O = example.com, CN = Certificate Shack
output omitted
New, TLSv1/SSLv3, Cipher is AES128-SHA
Server public key is 1024 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
SSL-Session:
  Protocol : TLSv1
output truncated
```

以上输出表明没有发生握手失败，并且协商一组密码。

openssl s_client 命令选项记录在 s_client(1) 手册页中。

有关 SSLv3 漏洞以及如何对其进行测试的更多信息，请参阅红帽知识库文章 POODLE：SSLv3 漏洞(CVE-2014-3566)。

14.1.10. 使用现有密钥和证书

如果您之前创建了密钥和证书，您可以将 SSL 服务器配置为使用这些文件，而不必生成新的密钥和证书。只有两种情况无法做到这一点：

1. 您正在更改 IP 地址或域名。

   为特定的 IP 地址和域名对签发证书。如果这些值之一改变，证书就会变得无效。

2. 您有来自 VeriSign 的证书，并且正在更改服务器软件。

   Verisign 是广泛使用的证书颁发机构，发布特定软件产品、IP 地址和域名的证书。更改软件产品会使证书无效。

在上述任一情形中，您将需要获取新的证书。有关此主题的详情请参考 第 14.1.11 节 “生成新密钥和证书”。

如果要使用现有密钥和证书，请将相关文件分别移到 /etc/pki/tls/private/ 和 /etc/pki/tls/certs/ 目录中。您可以以 root 用户身份运行以下命令：

```
~]# mv key_file.key /etc/pki/tls/private/hostname.key
~]# mv certificate.crt /etc/pki/tls/certs/hostname.crt
```

然后，在 /etc/httpd/conf.d/ssl.conf 配置文件中添加以下行：

```
SSLCertificateFile /etc/pki/tls/certs/hostname.crt
SSLCertificateKeyFile /etc/pki/tls/private/hostname.key
```

要加载更新的配置，请重新启动 httpd 服务，如 第 14.1.3.3 节 “重启服务” 所述。

例 14.5. 使用来自 Red Hat Secure Web 服务器的密钥和证书

```
~]# mv /etc/httpd/conf/httpsd.key /etc/pki/tls/private/penguin.example.com.key
~]# mv /etc/httpd/conf/httpsd.crt /etc/pki/tls/certs/penguin.example.com.crt
```

14.1.11. 生成新密钥和证书

要生成新密钥和证书对，必须在系统上安装 crypto-utils 软件包。要安装它，以 root 用户身份输入以下命令：

```
~]# yum install crypto-utils
```

此软件包提供一组工具来生成和管理 SSL 证书和私钥，并包含 genkey，即红帽密钥对生成实用程序，将引导您完成密钥生成过程。

重要

如果服务器已经拥有有效的证书，并且您要用一个新证书替换它，请指定不同的序列号。这样可确保客户端浏览器收到此更改的通知，如预期更新到这个新证书，且不会无法访问该页面。要使用自定义序列号创建新证书，以 root 用户身份使用以下命令而不是 genkey ：

```
~]# openssl req -x509 -new -set_serial number -key hostname.key -out hostname.crt
```

注意

如果系统中已有特定主机名的密钥文件，gen key 将拒绝启动。在这种情况下，以 root 用户身份使用以下命令删除现有文件：

```
~]# rm /etc/pki/tls/private/hostname.key
```

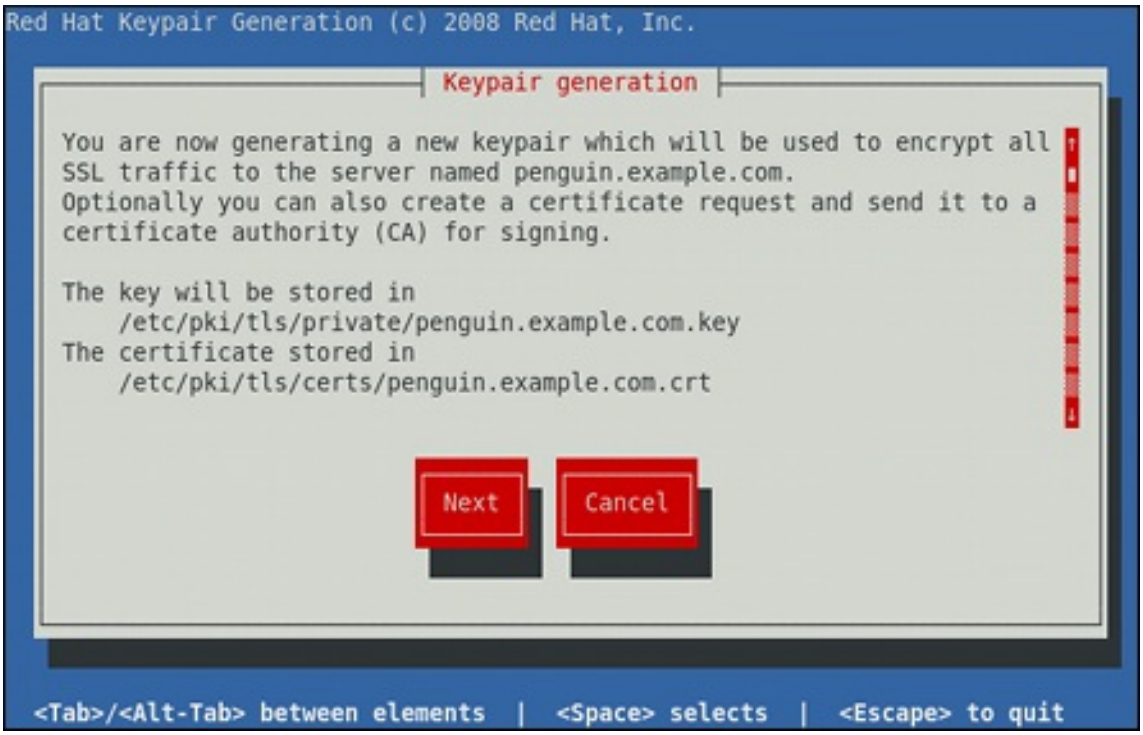
要运行实用程序，请以 root 身份输入 genkey 命令，再输入相应的主机名（如 penguin .example.com）：

```
~]# genkey hostname
```

要完成密钥和证书创建，请执行以下步骤：

1. 检查将在其中存储密钥和证书的目标位置。

图 14.1. 运行 genkey 工具

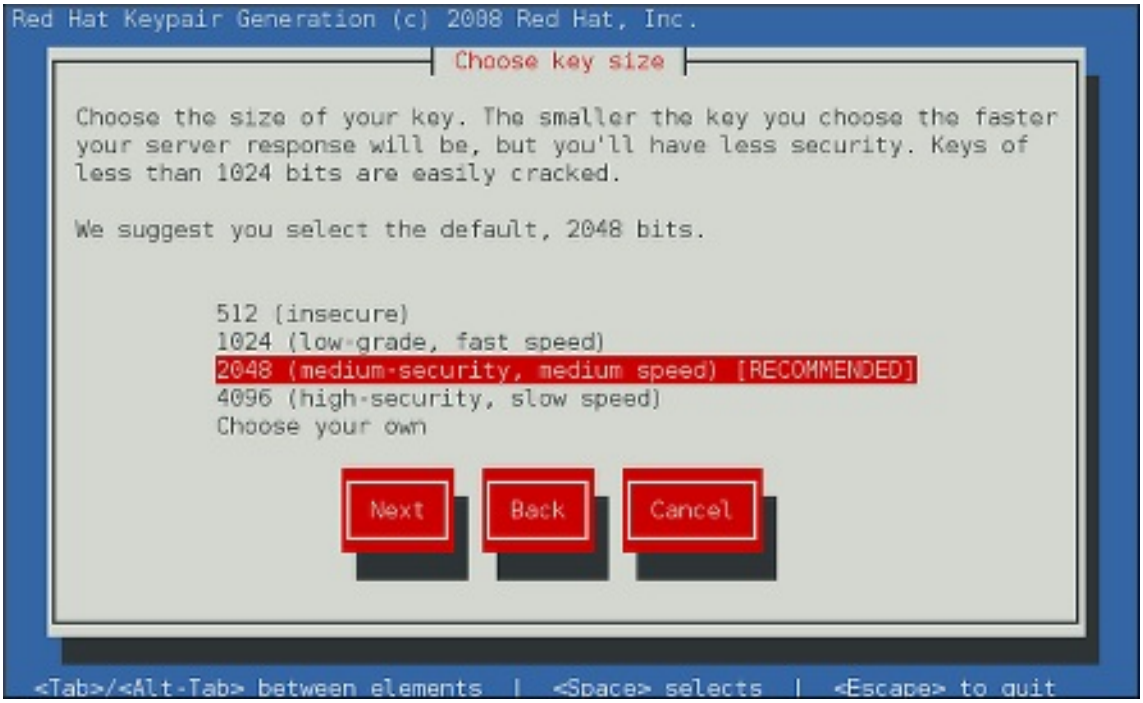


使用 Tab 键选择 下一步按钮，然后按 Enter 键继续下一屏幕。

2.

使用向上和 向下箭头键，选择合适的键大小。请注意，虽然较大的密钥会增加安全性，但也会增加服务器的响应时间。NIST 建议使用 2048 位。请参阅 NIST 特殊出版物 800-131A。

图 14.2. 选择密钥大小



完成后，使用 Tab 键选择 下一步按钮，然后按 Enter 以启动随机位生成过程。根据所选的密钥大小，这可能需要一些时间。

3.

决定是否要向证书颁发**机**构发**送**证书请**求。**

图 **14.3. 生成**证书请**求**

使用 Tab 键选择 **Yes 来**编**写**证书请**求，或者**选择 **No 来生成自**签**名**证书。**然后按 Enter** 键**确**认**您的**选择。

4.

使用 空格键，启**用([*])或禁用([])加密私**钥。

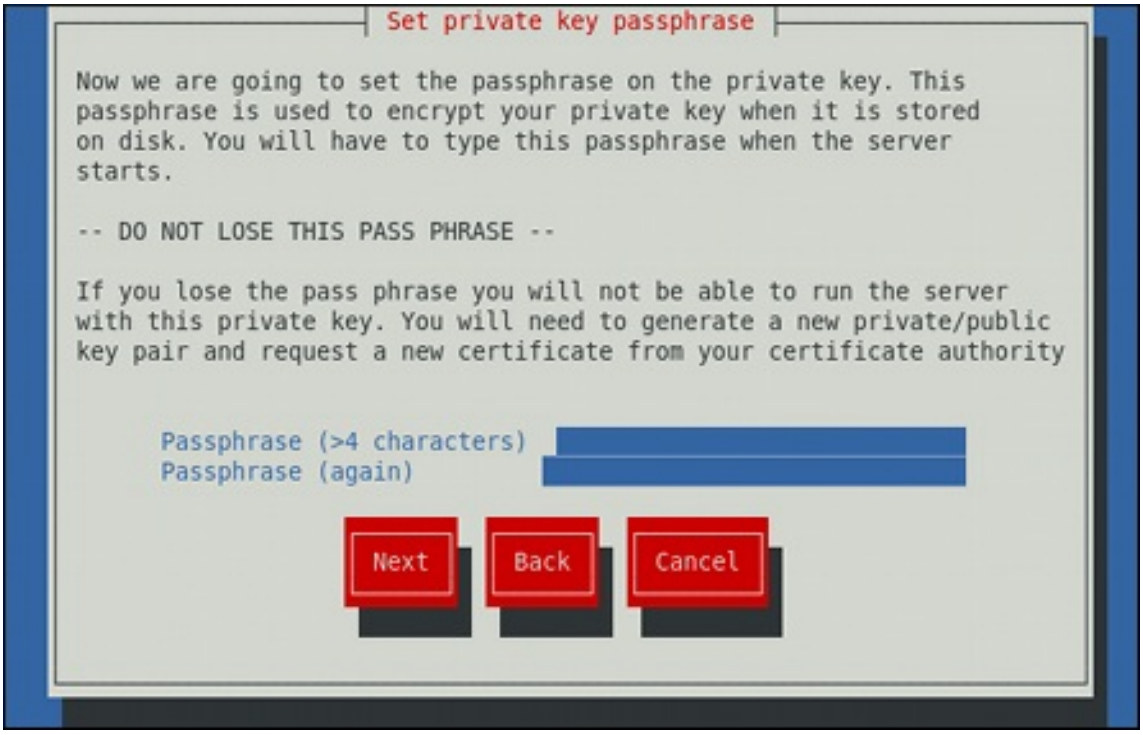
图 **14.4.** 加密私钥

使用 Tab 键选择 **下一步按钮，然后按 Enter** 键继续**下一屏幕。**

5.

如果您启**用了私**钥**加密，**请输**入适当的密**码**短**语。请**注意，出于安全原因，它不会**显**示在您**键**入**时，**且**长**度至少**为**五个字符。**

图 **14.5.** 输**入密**码



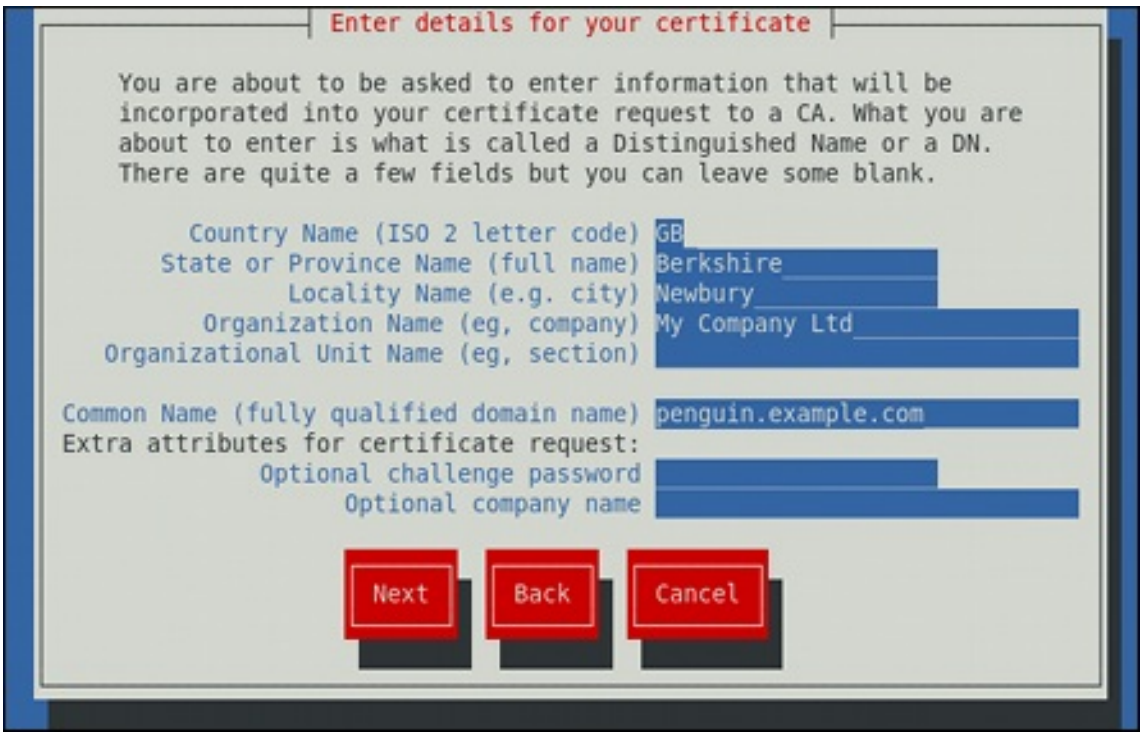
使用 Tab 键选择 下一步按钮，然后按 Enter 键继续下一屏幕。

重要

需要输入正确的密码短语才能启动服务器。如果丢失，则需要生成新的密钥和证书。

6. 自定义证书详细信息。

图 14.6. 指定证书信息



使用 Tab 键选择 下一步按钮，然后按 Enter 键完成密钥生成。

7. 如果您之前启用了证书请求生成，则会提示您将其发送到证书颁发机构。

图 14.7. 有关如何发送证书请求的说明

按 Enter 返回 shell 提示符。

生成后，将密钥和证书位置添加到 /etc/httpd/conf.d/ssl.conf 配置文件：

```
SSLCertificateFile /etc/pki/tls/certs/hostname.crt
SSLCertificateKeyFile /etc/pki/tls/private/hostname.key
```

最后，按照 第 14.1.3.3 节 “重启服务” 所述重启 httpd 服务，以便载入更新的配置。

14.1.12. 使用命令行配置 HTTP 和 HTTPS 的防火墙

默认情况下，Red Hat Enterprise Linux 不允许 HTTP 和 HTTPS 流量。要让系统成为 Web 服务器，请使用 firewalld 的支持服务启用 HTTP 和 HTTPS 流量根据需要通过防火墙。

要使用命令行启用 HTTP，以 root 用户身份运行以下命令：

```
~]# firewall-cmd --add-service http
 success
```

要使用命令行启用 HTTPS，以 root 用户身份运行以下命令：

```
~]# firewall-cmd --add-service https
 success
```

请注意，在系统下次启动后这些更改不会保留。要永久更改防火墙，请重复添加 --permanent 选项的命令。

14.1.12.1. 使用命令行检查网络访问传入 HTTPS 和 HTTPS

要检查防火墙配置为允许哪些服务，请以 root 用户身份运行以下命令：

```
~]# firewall-cmd --list-all
public (default, active)
 interfaces: em1
 sources:
 services: dhcpv6-client ssh
output truncated
```

在这个示例中，防火墙被启用，但 HTTP 和 HTTPS 不允许通过。

启用 HTTP 和 HTTP 防火墙服务后，service 行将显示类似如下：

```
services: dhcpv6-client http https ssh
```

有关启用防火墙服务或通过 firewalld 打开和关闭端口的更多信息，请参阅 Red Hat Enterprise Linux 7 安全指南。

14.1.13. 其它资源

要了解更多有关 Apache HTTP 服务器的信息，请参见以下资源：

安装的文档

- httpd(8) - httpd 服务的 man page，其中包含其命令行选项的完整列表。
- genkey(1) - genkey 实用程序的 man page，由 crypto-utils 软件包提供。

- apachectl(8) - Apache HTTP 服务器控制接口的 man page。

可安装文档

- http://localhost/manual/ - Apache HTTP 服务器的官方文档，其指令和可用模块的完整描述。请注意，若要访问此文档，您必须安装 httpd-manual 软件包，并且 Web 服务器必须正在运行。

  在访问文档前，以 root 用户身份运行以下命令：

```
~] yum install httpd-manual ~] apachectl graceful
```

在线文档

- http://httpd.apache.org/ - 包含所有指令和默认模块文档的 Apache HTTP 服务器的官方网站。

- http://www.openssl.org/ - OpenSSL 主页，其中包含更多文档、常见问题、邮件列表链接和其他有用的资源。

第 15 章 邮件服务器

红帽企业 Linux 提供许多高级应用程序来提供和访问电子邮件。本章描述了当今使用的现代电子邮件协议，以及一些旨在发送和接收电子邮件的程序。

15.1. 电子邮件协议

如今，电子邮件通过客户端/服务器架构发送。使用邮件客户端程序创建电子邮件消息。然后，该程序将消息发送到服务器。服务器随后将邮件转发到收件人的电子邮件服务器，然后邮件将发送到收件人的电子邮件客户端。

为启用此过程，各种标准网络协议允许不同的计算机，通常运行不同的操作系统并使用不同的电子邮件程序来发送和接收电子邮件。

以下讨论的协议是电子邮件传输中最常用的协议。

15.1.1. 邮件传输协议

从客户端应用程序向服务器发送的邮件以及从原始服务器发送到目标服务器的邮件发送由简单邮件传输协议 (SMTP) 处理。

15.1.1.1. SMTP

SMTP 的主要用途是在邮件服务器之间传输电子邮件。然而，这对电子邮件客户端也至关重要。若要发送电子邮件，客户端将邮件发送到传出邮件服务器，后者又联系目标邮件服务器以进行发送。但更中间的 SMTP 服务器可能包含在此链中。此概念称为邮件中继。因此，在配置电子邮件客户端时需要指定 SMTP 服务器。

在 Red Hat Enterprise Linux 下，用户可以在本地计算机上配置 SMTP 服务器来处理邮件发送。但是，也可以为传出邮件配置远程 SMTP 服务器。

关于 SMTP 协议的一个重要要点是它不需要身份验证。这样，互联网上的任何人都可以发送电子邮件给他人甚至大量人。SMTP 正是这种特征使垃圾电子邮件或垃圾邮件成为可能。实施中继限制限制互联网上的随机用户从通过 SMTP 服务器发送电子邮件到互联网上的其他服务器。不施加此类限制的服务器称为开放中继服务器。

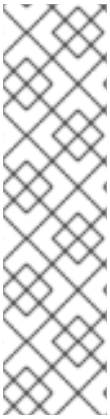
Red Hat Enterprise Linux 7 提供 Postfix 和 Sendmail SMTP 程序。

15.1.2. 邮件访问协议

电子邮件客户端应用程序使用两种主要协议从邮件服务器检索电子邮件：Post Office 协议 (POP)和 Internet 消息访问协议(IMAP)。

15.1.2.1. POP

Red Hat Enterprise Linux 下的默认 POP 服务器为 Dovecot，它由 dovecot 软件包提供。



注意

要安装 Dovecot，请运行以下命令：

```
~]# yum install dovecot
```

有关使用 Yum 安装软件包的详情请参考 第 9.2.4 节 “安装软件包”。

使用 POP 服务器时，电子邮件消息由电子邮件客户端应用程序下载。默认情况下，大多数 POP 电子邮件客户端配置为在电子邮件服务器上成功传输消息后将其删除，但此设置通常可以更改。

POP 与重要的互联网消息传递标准完全兼容，例如允许电子邮件附件的 多用途 Internet 邮件扩展(MIME)。

POP 最适合通过一个系统读取电子邮件。它还非常适合没有永久连接到互联网或包含邮件服务器的网络的用户。不幸的是，对于网络连接较慢的用户，POP 在验证时需要客户端程序来下载每条消息的所有内容。如果消息有很大的附件，可能需要很长时间。

标准 POP 协议的最新版本是 POP3。

但是，有不同的使用较少使用的 POP 协议变体：

-

APOP - 使用 MD5 验证的 POP3.用户密码的编码哈希从电子邮件客户端发送到服务器，而不是发送未加密的密码。

- KPOP - 带 Kerberos 验证的 POP3.
- RPOP - 使用 RPOP 身份验证的 POP3.这使用用户 ID（类似于密码）来验证 POP 请求。但是，此 ID 未加密，因此 RPOP 并不比标准 POP 更安全。

要提高安全性，您可以使用安全套接字层(SSL)加密进行客户端身份验证和数据传输会话。要启用 SSL 加密，请使用：

- pop3s 服务
- stunnel 应用程序
- starttls 命令

有关保护电子邮件通信的详情请参考 第 15.5.1 节 “安全通信”。

15.1.2.2. IMAP

Red Hat Enterprise Linux 下的默认 IMAP 服务器为 Dovecot，它由 dovecot 软件包提供。有关如何安装 Dovecot 的详情，请查看 第 15.1.2.1 节 “POP”。

使用 IMAP 邮件服务器时，电子邮件信息仍保留在用户可以读取或删除它们的服务器上。IMAP 还允许客户端应用程序在服务器上创建、重命名或删除邮件目录，以组织和存储电子邮件。

IMAP 对于使用多台计算机访问其电子邮件的用户特别有用。协议对于通过较慢的连接连接到邮件服务器的用户也很方便，因为仅为消息下载电子邮件标题信息，直到打开并节省带宽。用户也可以在不查看或下载消息的情况下删除消息。

为方便起见，IMAP 客户端应用程序可以在本地缓存消息副本，因此当用户没有直接连接到 IMAP 服务器时，可以浏览之前读取的消息。

IMAP 与 POP 完全兼容，与重要的 Internet 消息标准（如 MIME）完全兼容，后者允许电子邮件附件。

若要提高安全性，可以将 SSL 加密用于客户端身份验证和数据传输会话。这可以通过使用 imaps 服务或使用 stunnel 程序来启用。

- pop3s 服务
- stunnel 应用程序
- starttls 命令

有关保护电子邮件通信的详情请参考 第 15.5.1 节 “安全通信”。

其他免费的，以及商业、IMAP 客户端和服务器可用，其中很多扩展了 IMAP 协议并提供附加功能。

15.1.2.3. Dovecot

实施 IMAP 和 POP3 协议的 imap-login 和 pop 3 -login 进程由 dovecot 软件包中包含的主 dovecot 守护进程生成。IMAP 和 POP 的使用通过 /etc/dovecot/dovecot.conf 配置文件进行配置；默认情况下，dovecot 使用 SSL 运行 IMAP 和 POP3。要将 dovecot 配置为使用 POP，请完成以下步骤：

1. 编辑 /etc/dovecot/dovecot.conf 配置文件，以确保 protocol 变量被取消注释（删除行开头的 hash 符号(#)）并包含 pop3 参数。例如：

```
protocols = imap pop3 lmtp
```

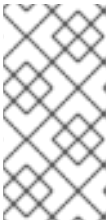
当 协议 变量被注释掉时，dovecot 将使用默认值，如上方所述。

2. 以 root 用户身份运行以下命令，为当前会话启用更改操作：

```
~]# systemctl restart dovecot
```

3. 运行以下命令，使更改在下一次重启后正常运行：

```
~]# systemctl enable dovecot
Created symlink from /etc/systemd/system/multi-user.target.wants/dovecot.service to
/usr/lib/systemd/system/dovecot.service.
```



注意

请注意，dvecot 仅报告其已启动 IMAP 服务器，而且会启动 POP3 服务器。

与 SMTP 不同，IMAP 和 POP3 都需要连接客户端以使用用户名和密码进行身份验证。默认情况下，两个协议的密码都通过网络未加密传递。

在 dovecot 上配置 SSL ：

- 编辑 /etc/dovecot/conf.d/10-ssl.conf 配置以确保 ssl_protocols 变量被取消注释并包含 !SSLv2 !SSLv3 参数：

```
ssl_protocols = !SSLv2 !SSLv3
```

这些值可确保 dovecot 避免 SSL 版本 2 和 3，因为它们都已知不安全。这是因为 POODLE 中描述的漏洞：SSLv3 漏洞(CVE-2014-3566)。有关详细信息，请参阅 Postfix 和 Dovecot 中的 POODLE SSL 3.0 漏洞(CVE-2014-3566) 解析。

确保 /etc/dovecot/conf.d/10-ssl.conf 包含以下选项：

```
ssl=required
```

- 按照您偏好编辑 /etc/pki/dovecot/dovecot-openssl.cnf 配置文件。但是，在典型的安装中，此文件不需要修改。

- 重命名、移动或删除文件 /etc/pki/dovecot/certs/dovecot.pem 和 /etc/pki/dovecot/private/dovecot.pem。

-

执行 /usr/libexec/dovecot/mkcert.sh 脚本，该脚本可创建 dovecot 自签名证书。这些证书复制到 /etc/pki/dovecot/certs 和 /etc/pki/dovecot/private 目录中。要实施更改，请以 root 用户身份运行以下命令来重启 dovecot ：

```
~]# systemctl restart dovecot
```

有关 dovecot 的更多详细信息，请访问

15.2. 电子邮件计划分类

一般来说，所有电子邮件应用程序至少分为三种分类之一。每个分类在移动和管理电子邮件消息过程中都起到特定的作用。尽管大多数用户只知道他们用于接收和发送邮件的具体电子邮件程序，但对于确保电子邮件到达正确的目的地来说，每个程序都很重要。

15.2.1. 邮件传输代理

邮件传输代理 (MTA)使用 SMTP 在主机之间传输电子邮件消息。当邮件移动到其预期目标时，该邮件可能会涉及多个 MTA。

尽管计算机之间的邮件传输似乎非常简单，但确定特定 MTA 是否或应接受邮件进行传输的整个过程非常复杂。此外，由于垃圾邮件的问题，特定 MTA 的使用通常受到 MTA 配置或 MTA 所在网络的访问配置的限制。

些电子邮件客户端程序可以在发送电子邮件时充当 MTA。但是，此类电子邮件客户端程序没有真正 MTA 的角色，因为它们只能向被授权使用的 MTA 发送出站邮件，但是它们无法直接将邮件传输到预期收件人的电子邮件服务器。如果运行应用的主机没有其自身的 MTA，则此功能非常有用。

由于 Red Hat Enterprise Linux 提供两个 MTA（Postfix 和 Sendmail ），因此电子邮件客户端程序通常不需要充当 MTA。Red Hat Enterprise Linux 还包括一个特殊用途的 MTA，称为 Fetchmail。

有关 Postfix、Sendmail 和 Fetchmail 的详情请参考 第 15.3 节 “邮件传输代理”。

15.2.2. 邮件发送代理

MTA 将邮件分发代理 (MDA)调用至相应用户的邮箱中传入电子邮件的文件。在很多情况下，MDA 实际上是 本地发送代理 (LDA)，如 mail 或 Procmail。

任何实际处理消息以传输到电子邮件客户端应用可读取消息的程序都可以视为 MDA。因此，某些 MTA（如 Sendmail 和 Postfix）可以在将新电子邮件邮件附加到本地用户的邮件池文件时填写 MDA 的角色。通常，MDA 不会在系统之间传输消息，也不提供用户界面；MDA 在本地计算机上分发和排序消息，供电子邮件客户端应用访问。

15.2.3. 邮件用户代理

邮件用户代理 (MUA)与电子邮件客户端应用同义词。MUA 是一种至少允许用户读取和编写电子邮件消息的程序。MUA 可以处理这些任务：

- 通过 POP 或 IMAP 协议检索消息
- 设置用于存储邮件的邮箱.
- 发送出站邮件到 MTA.

MUA 可能是图形化的，如 Thunderbird、Evolution 或具有简单的基于文本的界面，如 mail 或 Mutt。

15.3. 邮件传输代理

Red Hat Enterprise Linux 7 提供两个主要 MTA：Postfix 和 Sendmail。Postfix 配置为默认的 MTA，并且 Sendmail 被视为已弃用。如果需要将默认 MTA 切换到 Sendmail，您可以卸载 Postfix 或以 root 用户身份使用以下命令切换到 Sendmail：

```
~]# alternatives --config mta
```

您还可以使用以下命令启用所需的服务：

```
~]# systemctl enable service
```

同样，要禁用该服务，在 shell 提示符后输入以下内容：

```
~]# systemctl disable service
```

有关如何在 Red Hat Enterprise Linux 7 中管理系统服务的详情请参考 第 10 章 使用 systemd 管理服务。

15.3.1. postfix

Postfix 最初由安全专家和程序员 Wietse Venema 在 IBM 开发，它是一个兼容 Sendmail 的 MTA，旨在安全、快速且易于配置。

为提高安全性，Postfix 使用模块化设计，由 master 守护进程启动具有有限特权的小型进程。较少的特权较小的进程执行与邮件发送的各个阶段相关的非常具体的任务，并在更改的根环境中运行，以限制攻击的影响。

将 Postfix 配置为接受来自本地计算机以外的主机的网络连接，只需在其配置文件中进行一些细微更改。然而，对于具有更复杂需求的用户，Postfix 提供各种配置选项，以及使其成为功能非常多、功能齐全的第三方附加组件。

Postfix 的配置文件是人类可读的，并且支持超过 250 个指令。与 Sendmail 不同，更改生效不需要宏处理，并且大部分最常用的选项在大量注释的文件中予以说明。

15.3.1.1. 默认 Postfix 安装

Postfix 可执行文件是 postfix。此守护进程启动处理邮件发送所需的所有相关进程。

Postfix 将其配置文件存储在 /etc/postfix/ 目录中。以下是最常用的文件列表：

- 访问权限 - 用于访问控制，此文件指定允许哪些主机连接到 Postfix。
- main.cf - 全局 Postfix 配置文件。大多数配置选项在此文件中指定。
- master.cf - 指定 Postfix 如何与各种进程交互来完成邮件发送。
- transport - 映射用于中继主机的电子邮件地址。

别名 文件可以在 /etc 目录中找到。此文件在 Postfix 和 Sendmail 之间共享。它是邮件协议所需的

可配置列表，用于描述用户 ID 别名。

重要

默认的 /etc/postfix/main.cf 文件不允许 Postfix 接受来自本地计算机以外的主机的网络连接。有关将 Postfix 配置为其他客户端的服务器的详情请参考 第 15.3.1.3 节 “基本 Postfix 配置”。

在更改 /etc/postfix/ 目录下配置文件中的任何选项后，重新启动 postfix 服务，以便使这些更改生效。要做到这一点，以 root 运行以下命令：

```
~]# systemctl restart postfix
```

15.3.1.2. 从以前的版本升级

Red Hat Enterprise Linux 7 中的以下设置与之前的版本不同：

- disable_vrfy_command = no - 默认情况下是禁用的，这与 Sendmail 的默认值不同。如果更改为 yes，它可以阻止某些电子邮件地址回收方法。
- allow_percent_hack = yes - 默认情况下是启用的。它允许删除电子邮件地址中的 % 字符。百分点黑客是一个旧的临时解决方案，允许发件人控制电子邮件信息路由。DNS 和邮件路由现在更加可靠，但 Postfix 继续支持 hack。要关闭百分比重写，请将 allow_percent_hack 设置为 no。
- smtpd_helo_required = no - 默认情况下是禁用的，因为它在 Sendmail 中，因为它可以阻止某些应用程序发送邮件。可以将其更改为 yes，要求客户端在尝试发送 MAIL、FROM 或 ETRN 命令前发送 HELO 或 EHLO 命令。

15.3.1.3. 基本 Postfix 配置

默认情况下，Postfix 不接受来自本地主机以外的任何主机的网络连接。以 root 用户身份执行以下步骤，为网络中的其他主机启用邮件发送：

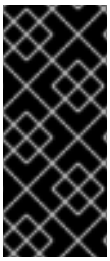
- 使用文本编辑器（如 vi ）编辑 /etc/postfix/main.cf 文件。
-

通过删除哈希符号(#)来取消注释 mydomain 行，并将 domain.tld 替换为邮件服务器服务的域，如 example.com。

- 取消注释 myorigin = $mydomain 行。
- 取消注释 myhostname 行，并将 host.domain.tld 替换为计算机的主机名。
- 取消注释 mydestination = $myhostname, localhost.$mydomain 行。
- 取消注释 mynetworks 行，并将 168.100.189.0/28 替换为可以连接到服务器的主机的有效网络设置。
- 取消注释 inet_interfaces = 所有 行.
- 为 inet_interfaces = localhost 行添加注释。
- 重新启动 postfix 服务。

完成这些步骤后，主机将接受外部电子邮件进行发送。

Postfix 对配置选项有较大的分类。学习如何配置 Postfix 的最佳方式之一是读取 /etc/postfix/main.cf 配置文件中的 注释。其他资源包括有关 Postfix 配置、SpamAssassin 集成或 /etc/postfix/main.cf 参数的详细信息，网址为



重要

由于 POODLE 中描述的漏洞：SSLv3 漏洞(CVE-2014-3566)， 红帽建议禁用 SSL，仅使用 TLSv1.1 或 TLSv1.2。有关详细信息，请参阅 Postfix 和 Dovecot 中的 POODLE SSL 3.0 漏洞(CVE-2014-3566) 解析。

15.3.1.4. 将 Postfix 与 LDAP 搭配使用

Postfix 可以将 LDAP 目录用作各种查找表的源（如 别名、虚拟、规范 等）。这允许 LDAP 存储分层用户信息和 Postfix 仅在需要时获得 LDAP 查询的结果。通过不在本地存储此信息，管理员可以轻松对

其进行维护。

15.3.1.4.1. /etc/aliases 查找示例

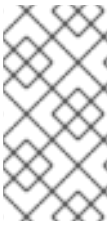
以下是使用 LDAP 查找 /etc/aliases 文件的基本示例：确保您的 /etc/postfix/main.cf 文件包含以下内容：

```
alias_maps = hash:/etc/aliases, ldap:/etc/postfix/ldap-aliases.cf
```

如果您还没有文件，请创建一个 /etc/postfix/ldap-aliases.cf 文件，并确保该文件包含以下内容：

```
server_host = ldap.example.com
search_base = dc=example, dc=com
```

其中 ldap.example.com 、 example .com 和 com 是需要替换为现有可用 LDAP 服务器的规格的参数。



注意

/etc/postfix/ldap-aliases.cf 文件可以指定各种参数，包括启用 LDAP SSL 和 STARTTLS 的参数。如需更多信息，请参阅 ldap_table(5) man page。

有关 LDAP 的更多信息，请参阅 系统级身份验证指南 中的 OpenLDAP。

15.3.2. sendmail

Sendmail 的核心用途，与其他 MTA 一样，是在主机之间安全地传输电子邮件，通常使用 SMTP 协议。请注意，Sendmail 被视为已弃用，建议管理员尽可能使用 Postfix。如需更多信息，请参阅 第 15.3.1 节 “postfix”。

15.3.2.1. 用途和限制

切记 Sendmail 是什么以及它可以做什么，而不是了解它的内容。在履行多个角色的单体式应用程序中，Sendmail 可能看似是组织内运行电子邮件服务器的唯一应用程序。从技术上讲，这是正确的，因为 Sendmail 可以假脱机发送邮件至每个用户的 目录，并为用户发送出站邮件。但是，大多数用户实际上需要的不仅仅是简单的电子邮件发送。用户通常希望使用使用 POP 或 IMAP 的 MUA 与电子邮件交互，以将其消息下载到其本地计算机。或者，他们可能希望使用 Web 界面来获取其邮箱的访问权限。这些其他应用程序可以与 Sendmail 配合使用，但实际上由于不同原因而存在，并可相互分离。

进入 Sendmail 应该或可配置为执行的所有内容超出了本节的范围。在字面上数百种不同的选项和规则集中，整个卷都致力于帮助解释所有可以执行的操作，以及如何修复出错的事情。有关 Sendmail 资源列表，请查看 第 15.7 节 “其它资源”。

本节默认回顾通过 Sendmail 安装的文件并回顾基本配置更改，包括如何停止不需要的电子邮件（垃圾邮件）以及如何使用轻量级目录访问协议 (LDAP) 扩展 Sendmail。

15.3.2.2. 默认 Sendmail 安装

要使用 Sendmail，首先确保以 root 用户身份运行 在您的系统中安装 sendmail 软件包：

```
~]# yum install sendmail
```

要配置 Sendmail，请以 root 用户身份 在您的系统中安装 sendmail-cf 软件包：

```
~]# yum install sendmail-cf
```

有关使用 Yum 安装软件包的详情请参考 第 9.2.4 节 “安装软件包”。

在使用 Sendmail 之前，必须先从 Postfix 切换默认 MTA。有关如何切换默认 MTA 的更多信息，请参阅 第 15.3 节 “邮件传输代理”

Sendmail 可执行文件是 sendmail。

Sendmail 配置文件位于 /etc/mail/sendmail.cf。避免直接编辑 sendmail.cf 文件。要对 Sendmail 进行配置更改，请编辑 /etc/mail/sendmail.mc 文件，备份原始 /etc/mail/sendmail.cf 文件，然后重新启动 sendmail 服务。作为重启的一部分，会重建 sendmail.cf 文件和所有数据库二进制表示法：

```
# systemctl restart sendmail
```

有关配置 Sendmail 的更多信息，请参阅 第 15.3.2.3 节 “常见 Sendmail 配置更改”。

各种 Sendmail 配置文件安装在 /etc/mail/ 目录中，其中包括：

- 访问 - 指定哪些系统可以使用 Sendmail 作为出站电子邮件.
- domaintable - 指定域名映射。
- local-host-names - 指定主机的别名。
- mailertable - 指定覆盖特定域路由的指令。
- virtusertable - 指定特定域的别名形式，允许在一台计算机上托管多个虚拟主机。

/etc/mail/ 目录中的多个配置文件（如 access、domain table、mailertable 和 virtusertable ）会在 Sendmail 可以使用任何配置更改之前将其信息存储在数据库文件中。

要将对这些配置所做的任何更改包含在其数据库文件中，请运行以下命令：

```
# systemctl restart sendmail
```

15.3.2.3. 常见 Sendmail 配置更改

更改 Sendmail 配置文件时，最好不要编辑现有文件，而是生成一个全新的 /etc/mail/sendmail.cf 文件。

警告

在替换或更改 sendmail.cf 文件之前，请先创建备份副本：

要向 Sendmail 添加所需的功能，请以 root 用户身份 编辑 /etc/mail/sendmail.mc 文件。完成后，重启 sendmail 服务，如果安装了 m4 软件包，m4 宏处理器将自动生成新的 sendmail.cf 配置文件：

```
~]# systemctl restart sendmail
```

重要

默认 sendmail.cf 文件不允许 Sendmail 接受来自本地计算机以外的任何主机的网络连接。要将 Sendmail 配置为其他客户端的服务器，请编辑 /etc/mail/sendmail.mc 文件，然后更改 DAEMON_OPTIONS 指令的 Addr= 选项中指定的地址，从 127.0.0.1 注释到活动网络设备的 IP 地址，或者在 行首 添加注释掉 DAEMON_OPTIONS 指令。完成后，重启服务来重新生成 /etc/mail/sendmail.cf:

```
~]# systemctl restart sendmail
```

Red Hat Enterprise Linux 中的默认配置适用于大多数 SMTP站点。

编辑 /usr/share/sendmail-cf/README 文件，然后编辑 /usr/share/sendmail-cf/ 目录下的任何文件，因为它们可能会影响 /etc/mail/sendmail.cf 文件的未来配置。

15.3.2.4. 伪装

种常见的 Sendmail 配置是让单个计算机充当网络上所有计算机的邮件网关。例如，公司可能希望一台名为 mail.example.com 的计算机来处理其所有电子邮件，并为所有传出邮件分配一致的返回地址。

在这种情况下，Sendmail 服务器必须伪装公司网络上的计算机名称，以便其返回地址为 user@example.com 而不是 user@host.example.com。

要做到这一点，请在 /etc/mail/sendmail.mc 中添加以下行：

```
FEATURE(always_add_domain)dnl
FEATURE(masquerade_entire_domain)dnl
FEATURE(masquerade_envelope)dnl
FEATURE(allmasquerade)dnl
MASQUERADE_DOMAIN(`example.com.')dnl
MASQUERADE_AS(`example.com')dnl
```

从 sendmail.mc 中更改的配置生成一个新的 sendmail.cf 文件后，使用以下命令重启 sendmail 服务：

```
# systemctl restart sendmail
```

请注意，邮件服务器、DNS 和 DHCP 服务器的管理员以及任何调配应用程序都应就组织中使用的主机名格式达成一致。有关推荐命名实践的更多信息，请参阅红帽企业 Linux 7 网络指南。

15.3.2.5. 停止 Spam

垃圾邮件可定义为无需任何请求通信的用户收到的不必要的电子邮件。它是一种破坏性、昂贵且广泛滥用互联网通信标准的行为。

Sendmail 使得阻止新垃圾邮件发送垃圾电子邮件相对容易。默认情况下，它甚至会阻止许多更常见的垃圾邮件方法。sendmail 中的主要反垃圾邮件功能是标题检查、转发拒绝 （默认为 8.9 版本）、访问数据库和发件人信息检查。

例如，自 Sendmail 版本 8.9 开始，默认禁用 SMTP 消息转发（也称为转发）。在发生此更改之前，Sendmail 会指示邮件主机(x.edu)接受来自某一方(y.com)的邮件并将其发送到其他方(z.net)。但是现在，Sendmail 必须配置为允许任何域通过服务器转发邮件。要配置中继域，请编辑 /etc/mail/relay-domains 文件并重新启动 Sendmail

```
~]# systemctl restart sendmail
```

但是，互联网上的服务器也可以发送垃圾邮件消息。在这些实例中，可以通过 /etc/mail/access 文件使用 Sendmail 的访问控制功能来防止来自不需要的主机的连接。以下示例说明了如何将此文件用于 块，并特别允许访问 Sendmail 服务器：

```
badspammer.com ERROR:550 "Go away and do not spam us anymore" tux.badspammer.com OK 10.0 RELAY
```

此示例显示从 badspammer.com 发送的任何电子邮件都会被阻止，并显示 550 个 RFC-821 兼容错误代码，并发回邮件。接受从 tux.badspammer.com 子域发送的电子邮件。最后一行显示从 10.0 发送的任何电子邮件。网络可以通过邮件服务器转发。

因为 /etc/mail/access.db 文件是一个数据库，请使用以下命令更新任何更改：

```
# systemctl restart sendmail
```

以上示例仅代表 Sendmail 允许或阻止访问的一小部分。有关详细信息和示例，请参见 /usr/share/sendmail-cf/README 文件。

由于 Sendmail 在发送邮件时调用 Procmail MDA，因此也可以使用垃圾邮件过滤程序（如 SpamAssassin）为用户识别和文件垃圾邮件。有关使用 SpamAssassin 的详情，请查看 第 15.4.2.6 节 “垃圾邮件过滤器”。

15.3.2.6. 将 Sendmail 与 LDAP 搭配使用

使用 LDAP 是一种非常快速、强大的方式，可以从大得多的组中查找特定用户的特定信息。例如，LDAP 服务器可用于按用户的姓氏从通用企业目录查找特定的电子邮件地址。在这种实施中，LDAP 很大程度上独立于 Sendmail，而 LDAP 存储分层用户信息和 Sendmail 仅获得预先寻址电子邮件消息中的 LDAP 查询的结果。

但是，Sendmail 支持与 LDAP 的更大集成，它使用 LDAP 在支持中级组织的不同邮件服务器上单独替换 /etc/aliases 和 /etc/mail/virtusertables 等单独维护的文件。简而言之，LDAP 将邮件路由级别从 Sendmail 抽象到可以被许多不同的应用程序利用的强大 LDAP 群集。

当前版本的 Sendmail 包含对 LDAP 的支持。要使用 LDAP 扩展 Sendmail 服务器，请首先获取一个 LDAP 服务器，如 OpenLDAP，并运行并正确配置。然后编辑 /etc/mail/sendmail.mc 以包括以下内容：

```
LDAPROUTE_DOMAIN('yourdomain.com')dnl
FEATURE('ldap_routing')dnl
```

注意

这仅适用于使用 LDAP 的 Sendmail 的非常基本配置。配置可能会有很大差异，具体取决于 LDAP 的实施，特别是将多个 Sendmail 计算机配置为使用通用 LDAP 服务器时。

有关详细的 LDAP 路由配置说明和示例，请参阅 /usr/share/sendmail-cf/README。

接下来，通过运行 m4 宏处理器并再次重新启动 Sendmail 来重新创建 /etc/mail/sendmail.cf 文件。具体步骤请查看 第 15.3.2.3 节 “常见 Sendmail 配置更改”。

有关 LDAP 的更多信息，请参阅 系统级身份验证指南 中的 OpenLDAP。

15.3.3. fetchmail

获取邮件是一个 MTA，它从远程服务器检索电子邮件并将其发送到本地 MTA。许多用户都感谢能够将下载位于远程服务器的消息与在 MUA 中读取和整理电子邮件的过程分开。根据拨号用户的需求设计，Fetchmail 使用包括 POP3 和 IMAP 在内的任意协议（包括 POP3 和 IMAP）连接并快速下载所有电子邮件到邮件池文件。如有必要，它甚至可以将电子邮件转发至 SMTP 服务器。

注意

要使用 Fetchmail，请首先确保以 root 用户身份在您的系统中安装 fetchmail 软件包：

```
~]# yum install fetchmail
```

有关使用 Yum 安装软件包的详情请参考 第 9.2.4 节 “安装软件包”。

通过在用户主目录中的 a .fetchmailrc 文件为每个用户配置 fetchmail。如果尚不存在，请在主目录中创建 .fetchmailrc 文件

使用 .fetchmailrc 文件中的首选项，Fetchmail 检查远程服务器上的电子邮件并下载。然后，它将该邮件传送到本地计算机上的端口 25，使用本地 MTA 将电子邮件放入正确的用户的假脱机文件中。如果 Procmail 可用，则会启动 以过滤电子邮件并将其放置在邮箱中，以便 MUA 可读取该电子邮件。

15.3.3.1. 获取邮件配置选项

虽然可以在命令行上传递所有必要的选项，以便在执行 Fetchmail 时检查远程服务器上的电子邮件，但使用 a .fetchmailrc 文件要简单得多。将任何所需的配置选项放在 .fetchmailrc 文件中，以便在每次发出 fetchmail 命令时使用这些选项。通过在命令行中指定该选项，可以在运行 Fetchmail 时覆盖这些设置。

用户的 .fetchmailrc 文件包含三类配置选项：

- 全局选项 - 提供控制程序操作的 Fetchmail 指令，或为每个检查电子邮件的连接提供设置。
- 服务器选项 - 指定有关被轮询服务器的必要信息，如主机名，以及特定电子邮件服务器的首选项，如要检查的端口或在超时前等待的秒数。这些选项会影响使用该服务器的每个用户。
- 用户选项 - 包含用户名和密码等信息，以便使用指定的电子邮件服务器进行身份验证和检查电子邮件。

全局选项将显示在 .fetchmailrc 文件的顶部，后跟一个或多个服务器选项，各自指定 Fetchmail 应检查的不同电子邮件服务器。用户选项遵循检查该电子邮件服务器的每个用户帐户的服务器选项。与服务器

选项一样，可以指定多个用户选项以用于特定的服务器，并检查同一服务器上的多个电子邮件帐户。

服务器选项在 .fetchmailrc 文件中使用特殊的 选项动词（ poll 或 skip ）调用到 service 中，该动词前面是任何服务器信息。poll 操作告知 Fetchmail 在运行时使用此 server 选项，该选项将使用指定的用户选项检查电子邮件。跳过 操作后的任何服务器选项都不会被检查，除非调用 Fetchmail 时指定此服务器的主机名。skip 选项可用于在 .fetchmailrc 文件中测试配置，因为它仅在特别调用时检查跳过的服务器，不影响任何当前正常工作的配置。

以下是 a .fetchmailrc 文件的示例：

```
set postmaster "user1"
set bouncemail

poll pop.domain.com proto pop3
  user 'user1' there with password 'secret' is user1 here

poll mail.domain2.com
  user 'user5' there with password 'secret2' is user1 here
  user 'user7' there with password 'secret3' is user1 here
```

在本例中，全局选项指定用户作为最后的手段（postmaster 选项）发送电子邮件，所有电子邮件错误都会发送到 postmaster 而不是发件人（退回邮件 选项）。set 操作告知 Fetchmail，此行包含全局选项。然后，指定了两台电子邮件服务器，一个设置为使用 POP3 进行检查，另一台用于尝试各种协议来查找有效的协议。使用第二个 server 选项检查两个用户，但为任何用户找到的所有电子邮件都将发送到 user1 的邮件假脱机。这允许在多个服务器上检查多个邮箱，同时显示在一个 MUA 收件箱中。每个用户的具体信息均以 用户 操作开头。

注意

用户无需将其密码放置在 .fetchmailrc 文件中。如果省略 密码为"" 部分，则会在 Fetchmail 启动时询问密码。

fetchmail 具有许多全局、服务器和本地选项。这些选项中很多很少被使用，或者仅适用于非常具体的情况。fetchmail man page 详细介绍了每个选项，但以下三节中列出了最常见的选项。

15.3.3.2. 全局选项

每个全局选项应在设置操作后放置到一行上 。

- 守护进程 秒 - 指定 daemon-mode，其中 Fetchmail 保留在后台。将 秒 替换为 Fetchmail 在轮询服务器前要等待的秒数。

- postmaster - 指定本地用户在发送问题时发送邮件。

- syslog - 指定日志文件以查找错误和状态消息。默认情况下，这是 /var/log/maillog。

15.3.3.3. 服务器选项

服务器选项必须在 轮询 或 跳过 操作后放在其自己的行 in .fetchmailrc 中。

- auth auth-type - 使用要使用的身份验证类型替换 auth-type。默认情况下 使用密码 身份验证，但某些协议支持其他类型的身份验证，包括 kerberos_v5、kerberos_v4 和 ssh。如果使用任何 身份验证类型，Fetchmail 首先会尝试不需要密码的方法，然后使用屏蔽密码的方法，最后尝试发送未加密的密码以向服务器发送身份验证。

- 间隔 号 - 按其在所有配置的服务器 中 检查电子邮件的次数对指定服务器轮询一次。此选项通常用于用户很少接收邮件的电子邮件服务器。

- 端口 port-number - 使用 端口号替换 port-number。这个值覆盖指定协议的默认端口号。

- Pro to 协议 - 将协议替换为协议，如 pop3 或 imap，以便在检查服务器上的消息时使用。

- 超时 秒 - 用 Fetchmail 在连接尝试中放弃的服务器 数 替换秒数。如果没有设置这个值，则使用默认值 300 秒。

15.3.3.4. 用户选项

用户选项可以放置在其服务器选项下的行上，或者放在与服务器选项相同的行上。在这两种情况下，定义的选项都必须遵循 user 选项（如下定义）。

- fetchall - Orders Fetchmail 下载队列中的所有消息，包括已经查看的邮件。默认情况下，Fetchmail 仅拉取新邮件。

- fetchlimit number - 将 编号 替换为停止前要检索的消息数。

- flush - 在检索新消息前删除之前在队列中查看的所有消息。

- 限制 max-number-bytes - 将 max-number-bytes 替换为 Fetchmail 检索时允许的最大字节大小。在网络链接速度较慢的情况下，当大量消息下载用时，这个选项非常有用。

- 密码 'password' - 将 password 替换为用户的密码.

- 预连接"命令 "" - 使用要执行 的命令替换命令，然后再检索用户的消息.

- 后连接"命令" - 使用检索用户消息后要执行的命令替换 命令.

- SSL - 激活 SSL 加密.编写本文时，默认操作是使用 SSL2、SSL3、SSL23、TLS1、TLS1.1 和 TLS 1.2 中提供的最佳功能。请注意，SSL2 被视为过时，且由于 POODLE：SSLv3 漏洞(CVE-2014-3566)，不应使用 SSLv3。但是，无法强制使用 TLS1 或更新版本，因此确保连接到的邮件服务器已 配置为使用 SSLv2 和 SSLv3。使用 stunnel，其中无法将服务器 配置为使用 SSLv2 和 SSLv3。

- sslproto - 定义允许 SSL 或 TLS 协议.可能的值有 SSL2、SSL3、SSL 23 和 TLS1。默认值 if sslproto 被 省略、未设置或设置为无效值，即 SSL23。默认操作是最佳使用 SSLv2、SSLv 3、TLSv 1、TLS1.1 和 TLS1.2。请注意，为 SSL 或 TLS 设置任何其他值将禁用所有其他协议。由于 POODLE: SSLv3 漏洞(CVE-2014-3566)，建议省略此选项，或者将其设置为 SSLv23，并将对应的邮件服务器 配置为使用 SSLv2 和 SSLv3。使用 stunnel，其中无法将服务器 配置为使用 SSLv2 和 SSLv3。

- 用户 "用户名 "" - 使用 Fetchmail 用于检索邮件 的用户名替换 username.这个选项必须在所有其他用户选项之前。

15.3.3.5. 获取邮件命令选项

执行 fetchmail 命令时，大多数命令行中使用的 Fetchmail 选项都会镜像 .fetchmailrc 配置选项。这样，Fetchmail 可以与或不带有配置文件一起使用。大多数用户不会在命令行中使用这些选项，因为将这些选项保存在 .fetchmailrc 文件中更为简单。

有时可能需要使用其他选项来运行 fetchmail 命令，以满足特定目的。可以发出命令选项来临时覆盖导致错误的 a .fetchmailrc 设置，因为命令行中指定的任何选项都会覆盖配置文件选项。

15.3.3.6. 信息或调试选项

fetchmail 命令后使用的某些选项可以提供重要信息。

- --configdump - 根据来自 .fetchmailrc 和 Fetchmail defaults 的信息显示每个可能的选项。使用此选项时，不会检索任何用户的电子邮件。
- -s - 以静默模式执行 Fetchmail，防止任何消息（除错误）在 fetchmail 命令之后显示。
- -v - 以详细模式执行 Fetchmail，显示 Fetchmail 和远程电子邮件服务器之间的每个通信.
- -v - 显示详细的版本信息，列出其全局选项，并显示要与每位用户一起使用的设置，包括电子邮件协议和身份验证方法。使用此选项时，不会检索任何用户的电子邮件。

15.3.3.7. 特殊选项

这些选项偶尔可用于覆盖在 .fetchmailrc 文件中经常找到的默认值。

- -a - Fetchmail 从远程电子邮件服务器下载所有邮件，无论新的还是之前查看过。默认情况下，Fetchmail 仅下载新邮件。
- -k - Fetchmail 下载邮件后将邮件保存在远程电子邮件服务器上。这个选项覆盖下载消息后删除消息的默认行为。
- -L max-number-bytes - Fetchmail 不会以特定大小下载任何邮件，而是将它们保存在远程电子邮件服务器上。
- --quit - 退出 Fetchmail 守护进程进程。

可以在 fetch mail man page 中找到更多命令和.fetchmail rc 选项。

15.3.4. 邮件传输代理(MTA)配置

邮件传输代理 (MTA)是发送电子邮件的基础。邮件用户代理 (MUA)（如 Evolution 或 Mutt）用于读取和编写电子邮件。当用户从 MUA 发送电子邮件时，该邮件将移交给 MTA，该 MTA 通过一系列 MTA 发送邮件，直至该邮件到达其目的地。

即使用户不计划从系统发送电子邮件，某些自动任务或系统程序也可能使用 mail 命令将包含日志消息的电子邮件发送给本地系统的 root 用户。

Red Hat Enterprise Linux 7 提供两个 MTA：Postfix 和 Sendmail。如果同时安装了这两者，Postfix 是默认的 MTA。

15.4. 邮件发送代理

红帽企业 Linux 包括 Procmail 作为主 MDA。两个应用程序都被视为 LDA，并将电子邮件从 MTA 的假脱机文件移动到用户的邮箱。但是，Procmail 提供强大的过滤系统。

本节仅详细介绍 Procmail。有关 mail 命令的信息，请查阅其 man page(man mail)。

Procmail 在放置在 localhost 的邮件假脱机文件中时发送和过滤电子邮件。它非常强大，系统资源很强大，使用广泛。Procmail 在发送电子邮件以供电子邮件客户端应用读取方面可起到关键作用。

Procmail 可以通过多种不同的方式调用：每当 MTA 将电子邮件放置到邮件池文件时，Procmail 将启动。然后，Procmail 过滤并记录 MUA 的电子邮件并退出。或者，可以将 MUA 配置为在收到邮件时执行 Procmail，以便邮件移至其正确的邮箱。默认情况下，当 MTA 收到新邮件时，用户的主目录中存在 /etc/procmailrc 或 ~/.proc mailrc 文件（也称为 arc 文件）会调用 Procmail。

默认情况下，/etc 目录中不存在系统范围的 rc 文件，任何用户的主目录中都不存在 no .procmailrc 文件。因此，要使用 Procmail，每个用户都必须使用特定的环境变量和规则构造 a .procmailrc 文件。

Procmail 是否对电子邮件消息作用取决于消息是否与 rc 文件中的一组指定条件或 配方 匹配。如果消息与方法匹配，则电子邮件将放置在指定的文件中，将被删除或处理。

当 Procmail 启动时，它会读取电子邮件并将正文与标题信息分隔开。接下来，Procmail 在 /etc/procmailrc s/ 目录中查找 / etc/procmailrc 文件和rc 文件，用于默认、系统范围、Procmail 环境变量和配方。然后，Procmail 搜索用户主目录中的 a .procmailrc 文件。许多用户还会为 Procmail 创建额外的 rc 文件，该文件在主目录中的 .procmailrc 文件中引用。

15.4.1. Procmail 配置

Procmail 配置文件包含重要的环境变量。这些变量指定了要排序的消息，以及如何处理与任何方法不匹配的消息。

这些环境变量通常以以下格式显示在 ~/.procmailrc 文件的开头：

```
env-variable="value"
```

在本例中，env-variable 是变量的名称，值 定义 变量。

大多数 Procmail 用户未使用许多环境变量，而且许多更重要的环境变量都已使用默认值定义。大多数情况下，使用以下变量：

- DEFAULT - 设置默认邮箱，其中放置与任何方法不匹配的消息。

  default DEFAULT 值与 $ORGMAIL 相同。

- INCLUDERC - 指定包含要对其检查消息的更多方法的额外 rc 文件。这会将 Procmail 方法列表分解为履行不同角色的单独文件，例如阻止垃圾邮件和管理电子邮件列表，然后关闭这些文件或使用用户的 ~/.procmailrc 文件中的注释字符。

  例如，用户的 ~/.procmailrc 文件中的行可能类似如下：

  ```
  MAILDIR=$HOME/Msgs
  INCLUDERC=$MAILDIR/lists.rc
  INCLUDERC=$MAILDIR/spam.rc
  ```

  要关闭电子邮件列表的 Procmail 过滤但保留垃圾邮件控制，请使用哈希符号(#)注释掉第一个 INCLUDERC 行。请注意，它使用相对于当前目录的路径。

- LOCKSLEEP - 设置 Procmail 尝试使用特定锁定文件之间的时间间隔（以秒为单位）。默认值为 8 秒。

- LOCKTIMEOUT - 设置在最后修改锁定文件后必须经过的时间（以秒为单位），然后 Procmail 假定 lockfile 旧且可以删除。默认值为 1024 秒。

- LOGFILE - 将任何 Procmail 信息或错误消息写入的文件。

- MAILDIR - 为 Procmail 设置当前工作目录.如果设置，所有其他 Procmail 路径均相对于此目录。

- ORGMAIL - 指定原始邮箱或其他位置，以便在邮件无法放置在默认或必需位置时放置邮件。

  默认情况下，使用 /var/spool/mail/$LOGNAME 的值。

- SUSPEND - 设置 Procmail 在必要的资源（如交换空间）不可用时暂停的时间（以秒为单位）。

- SWITCHRC - 允许用户指定包含其他 Procmail 配方的外部文件（类似于 INCLUDERC 选项），但实际停止了引用配置文件的引用配置文件，仅使用 SWITCHRC-specified 文件上的配方。

- VERBOSE - 导致 Procmail 记录更多信息。此选项对于调试非常有用。

其他重要环境变量从 shell 中拉取，如 LOGNAME、登录名称、HOME、主目录的位置；以及 SHELL （默认 shell）。

procmailrc man page 中提供了所有环境变量及其默认值的全面说明。

15.4.2. Procmail Recipes

新用户通常发现构建配方的方法是使用 Procmail 时最难学习的部分。这种困难通常导致了使用用于指定字符串匹配条件的正则表达式来配方匹配消息。但是，正则表达式在阅读时并不难以构造，甚至更难以理解。此外，Procmail 配方的编写方式的一致性，无论正则表达式如何，都让您能够轻松通过示例学习。要查看 Procmail 方法示例，请参阅 第 15.4.2.5 节 “配方示例”。

Procmail 方法采用以下形式：

```
:0 flags : lockfile-name
* condition_1_special-condition-character condition_1_regular_expression
* condition_2_special-condition-character condition-2_regular_expression
* condition_N_special-condition-character condition-N_regular_expression
  special-action-character
  action-to-perform
```

Procmail 方法中的前两个字符是冒号和一个零。可以在零的后面放置各种标志，以控制 Procmail 如何处理该方法。flags 部分后有一个冒号指定为此消息创建一个 lockfile。如果创建了 lockfile，可以通过替换 lockfile-name 来指定名称。

配方可以包含与消息匹配的多个条件。如果没有条件，则每个消息都与配方匹配。正则表达式在某些条件中放置，以便于消息匹配。如果使用多个条件，它们必须全部匹配才能执行该操作。根据配方第一行中设置的标志检查条件。在星号字符(*)后面放置的可选特殊字符可进一步控制条件。

action-to-perform 参数指定消息与其中一个条件匹配时执行的操作。每个方法只能有一个操作。许多情况下，此处使用邮箱名称将匹配的邮件定向到该文件中，从而有效地对电子邮件进行排序。也可以在指定操作之前使用特殊操作字符。如需更多信息，请参阅 第 15.4.2.4 节 “特殊条件和操作”。

15.4.2.1. 交付与.非交付 Recipes

如果方法与某一特定消息匹配，则使用的操作决定了它是否被视为传送方式还是非传送方法 。传输方法包含将消息写入文件、将消息发送到另一个程序的操作，或者将消息转发到另一个电子邮件地址。非传送方法涵盖任何其他操作，如嵌套块。嵌套块是一组操作，包含在大括号 {} 中，在与方法条件匹配的消息上执行。嵌套块可以互相嵌套，为识别和执行消息的操作提供更好的控制。

当消息与传输方法匹配时，Procmail 执行指定的操作，并停止将消息与任何其他方法进行比较。与非传送配方匹配的消息继续与其他配方进行比较。

15.4.2.2. 标记

标志对于确定方法或是否将方法与消息进行比较至关重要。The egrep 实用程序在内部用于匹配条件。常用的标记如下：

- a - 指定仅在之前没有 A 或标志的配方匹配这个消息时使用此方法。
- a - 指定仅在之前与 A 或 标志 匹配的方法 并且 成功完成时，才会使用此方法。

- b - 解析邮件正文并查找匹配条件。
- b - 使用任何结果操作中的正文，例如将消息写入文件或将其转发。这是默认的行为。
- c - 生成电子邮件的碳副本。这可用于传输方法，因为可以对消息执行必要的操作，并且可以在 rc 文件中继续处理消息的副本。
- D - 使 egrep 比较区分大小写.默认情况下，比较过程不区分大小写。
- e - 虽然与 A 标志类似，但方法中的条件仅与消息相比，当前方的配方没有 E 标志不匹配时。这等同于 其他操作。
- e - 只有在立即前面的配方中指定的操作失败时，才会将方法与消息进行比较。
- f - 将管道用作过滤器。
- h - 解析邮件的标题并查找匹配条件。这是默认的行为。
- h - 在生成的操作中使用标头。这是默认的行为。
- w - Tells Procmail 等待指定过滤器或程序完成，并报告在考虑过滤后邮件是否成功。
- w - 除"程序失败"消息被抑制外，与 w 相同。

有关其他标志的详细列表，请参阅 procmailrc man page。

15.4.2.3. 指定本地锁定文件

Procmail 提供 Lockfiles 非常有用，可确保多个进程不会尝试同时更改邮件。通过在方法的第一行上的任何标志后面加上冒号(:)来指定本地锁定文件。这会基于目标文件名加上 LOCKEXT 全局环境变量中设置的任何内容创建一个本地锁定文件。

或者，指定冒号后要用于此方法的本地锁定文件名称。

15.4.2.4. 特殊条件和操作

Procmail 方法条件和操作会改变它们的解释方式。

以下字符可在方法行开头的星号字符(*)后面使用：

- ! - 在条件行中，这个字符会颠倒该条件，只有在条件与邮件不匹配时才会导致匹配。
- < - 检查消息是否在指定数量的字节下。
- > - 检查消息是否超过指定数量的字节。

以下字符用于执行特殊操作：

- ! - 在操作行中，此字符告知 Procmail 将邮件转发至指定的电子邮件地址。
- $ - 引用之前在 rc 文件中设置的变量。这通常用于设置由不同方法引用的通用邮箱。
- | - 启动指定程序来处理消息.
- { 和 } - 构造一个嵌套块，用于包含应用到匹配消息的额外方法。

如果在操作行开头没有使用特殊字符，Procmail 假定操作行正在指定要在其中写入邮件的邮箱。

15.4.2.5. 配方示例

Procmail 是一个极其灵活的程序，但由于这种灵活性，新用户很难从头开始制作 Procmail 配方。

培养构建 Procmail 配方条件的技能的最佳方式源自对正则表达式的深入了解，以及查看他人构建的许多示例。正则表达式的详尽说明不在本节的范围之内。Procmail 配方和有用示例 Procmail 配方和有用示例 Procmail 配方可在 Internet 上的不同位置找到。要查看这些方法示例，可以生成对正则表达式的正确使用和改编。此外，可以在 grep(1) man page 中找到有关基本正则表达式规则的介绍信息。

以下简单示例演示了 Procmail 配方的基本结构，并且可为更加复杂的结构提供基础。

基本方法可能甚至不包含条件，如下例所示：

```
:0:
new-mail.spool
```

第一行指定将创建本地锁定文件，但不指定名称，因此 Procmail 使用目标文件名并附加 LOCKEXT 环境变量中指定的值。未指定任何条件，因此每个邮件均与此方法匹配，并放置在 MAILDIR 环境变量指定的目录中，名为 new-mail.spool 文件。然后 MUA 可以在该文件中查看消息。

这样的基本方法可以放在 all rc 文件的末尾，以将消息定向到默认位置。

以下示例匹配来自特定电子邮件地址的消息并将其丢弃。

```
:0
* ^From: spammer@domain.com
/dev/null
```

在这个示例中，spammer@domain.com 发送的所有信息都会发送到 / dev/null 设备，删除它们。

警告

请确保规则正在按预期运行，然后发送消息到 /dev/null 以永久删除。如果意外捕获意想不到的消息，并且这些消息消失，那么对规则进行故障排除就会变得困难。

更好的解决方案是将配方的操作指向一个特殊的邮箱，可以不时检查该邮箱查找假的正状态。且发现没有意外匹配任何消息，请删除邮箱并指示操作将消息发送到 /dev/null。

以下配方抓取从特定邮件列表中发送的电子邮件并将其放置在指定的文件夹中：

```
:0:
* ^(From|Cc|To).*tux-lug
tuxlug
```

从 tux-lug@domain.com 邮件列表发送的所有消息都自动放在 MUA 的 tuxlug 邮箱中。请注意，如果此示例中的条件在 From、Cc 或 To 行上有邮件列表的电子邮件地址，则此示例中的条件与邮件相匹配。

如需更详细、功能强大的配方，请参阅 第 15.7 节 “其它资源” 中的许多 Procmail 在线资源。

15.4.2.6. 垃圾邮件过滤器

由于接收新电子邮件后 Sendmail、Postfix 和 Fetchmail 调用它，Procmail 可用作防御垃圾邮件的强大工具。

当 Procmail 与 SpamAssassin 结合使用时，这一点尤为重要。当一起使用时，这两个应用可以快速识别垃圾邮件电子邮件，并对其进行排序或销毁。

SpamAssassin 使用标题分析、文本分析、黑名单、垃圾邮件跟踪数据库和自学 Bayesian 垃圾邮件分析来快速准确地识别和标记垃圾邮件。

注意

要使用 SpamAssassin，请首先确保以 root 用户身份在您的系统中安装 spamassassin 软件包：

```
~]# yum install spamassassin
```

有关使用 Yum 安装软件包的详情请参考 第 9.2.4 节 “安装软件包”。

本地用户使用 SpamAssassin 的最简单方法是将以下行放在 ~/.procmailrc 文件顶部附近：

```
INCLUDERC=/etc/mail/spamassassin/spamassassin-default.rc
```

/etc/mail/spamassassin/spamassassin-default.rc 包含一个简单的 Procmail 规则，用于激活所有传入电子邮件的 SpamAssassin。如果确定电子邮件为垃圾邮件，则在标题中将其标记，标题前面带有以下模式：

```
*****SPAM*****
```

电子邮件的正文也会紧随其后，确定是什么元素导致该邮件被诊断为垃圾邮件。

要提交标记为垃圾邮件的电子邮件，可以使用类似如下的规则：

```
:0 Hw
* ^X-Spam-Status: Yes
spam
```

该规则会将标题中标记的所有电子邮件作为垃圾邮件文件到名为垃圾邮件的邮箱中。

由于 SpamAssassin 是 Perl 脚本，因此在繁忙的服务器中可能需要使用二进制 SpamAssassin 守护进程（垃圾邮件）和客户端应用程序(spamc)。然而，以这种方式配置 SpamAssassin 需要对该主机的 root 访问权限。

要启动 spamd 守护进程，请输入以下命令：

```
~]# systemctl start spamassassin
```

要在系统引导时启动 SpamAssassin 守护进程，请运行：

```
systemctl enable spamassassin.service
```

有关启动和停止服务的更多信息，请参阅 第 10 章 使用 systemd 管理服务。

要将 Procmail 配置为使用 SpamAssassin 客户端应用程序而不是 Perl 脚本，请在 ~/.procmailrc 文件顶部附近放置下面这一行：对于系统范围的配置，将其放在 /etc/procmailrc 中：

```
INCLUDERC=/etc/mail/spamassassin/spamassassin-spamc.rc
```

15.5. 邮件用户代理

Red Hat Enterprise Linux 提供各种电子邮件程序，包括图形电子邮件客户端程序，如 Evolution，以及 mutt 等基于文本的电子邮件程序。

本节的其余部分侧重于保护客户端和服务器之间的通信。

15.5.1. 安全通信

MUA 包含在红帽企业 Linux 中，如 Thunderbird、Evolution 和 Mutt 提供 SSL 加密电子邮件会话。

与通过未加密网络流出的任何其他服务一样，重要的电子邮件信息（如用户名、密码和整个消息）可能会被网络上的用户截获和查看。此外，由于标准 POP 和 IMAP 协议以未加密的方式传递身份验证信息，攻击者可以通过收集通过网络传递的用户名和密码来获得用户帐户的访问权限。

15.5.1.1. 安全电子邮件客户端

大多数 Linux MUA 旨在检查远程服务器上的电子邮件，都支持 SSL 加密。要在检索电子邮件时使用 SSL，必须在电子邮件客户端和服务器中启用 SSL。

SSL 可在客户端上轻松启用，通常通过单击 MUA 配置窗口中的按钮或 MUA 配置文件中的 选项来完成。安全 IMAP 和 POP 具有已知端口号（分别为 993 和 995），MUA 用于验证和下载消息。

15.5.1.2. 保护电子邮件客户端通讯

向 IMAP 和 POP 用户在电子邮件服务器上提供 SSL 加密非常简单。

首先，创建 SSL 证书。这可以通过两种方式完成：应用到 SSL 证书的证书颁发机构(CA)或创建自签名证书。

警告

自签名证书应仅用于测试目的。在生产环境中使用的任何服务器都应使用由 CA 签名的 SSL 证书。

要为 IMAP 或 POP 创建自签名 SSL 证书，请更改为 /etc/pki/dovecot/ 目录，编辑 /etc/pki/dovecot/dovecot-openssl.cnf 配置文件中的证书参数，以 root 用户身份输入以下命令：

```
dovecot]# rm -f certs/dovecot.pem private/dovecot.pem
dovecot]# /usr/libexec/dovecot/mkcert.sh
```

完成后，请确保在 /etc/dovecot/conf.d/10-ssl.conf 文件中具有以下配置：

```
ssl_cert = </etc/pki/dovecot/certs/dovecot.pem
ssl_key = </etc/pki/dovecot/private/dovecot.pem
```

使用以下命令重启 dovecot 守护进程：

```
~]# systemctl restart dovecot
```

或者，可以将 stunnel 命令用作围绕 IMAP 或 POP 服务的标准非安全连接的加密打包程序。

stunnel 实用程序使用红帽企业 Linux 中包含的外部 OpenSSL 库来提供强大的加密和保护网络连接。建议应用到 CA 以获取 SSL 证书，但也可以创建自签名证书。

有关如何安装 stunnel 并创建其基本配置的说明，请参阅 Red Hat Enterprise Linux 7 安全指南中的

stunnel。要将 stunnel 配置为 IMAPS 和 POP3S 的打包程序，请在 /etc/stunnel/stunnel.conf 配置文件中添加以下行：

```
[pop3s]
accept = 995
connect = 110

[imaps]
accept = 993
connect = 143
```

《安全指南》还解释了如何启动和停止 stunnel。启动后，可以使用 IMAP 或 POP 电子邮件客户端并使用 SSL 加密连接到电子邮件服务器。

15.6. 使用防垃圾邮件和防病毒配置邮件服务器

当您的电子邮件发送正常工作后，传入电子邮件可能包含主动邮件（也称为垃圾邮件）。这些消息还可以包含恶意病毒和恶意软件，在您的系统上造成安全风险和潜在的生产损失。

为避免这些风险，您可以使用反垃圾邮件和防病毒来过滤传入的信息，并根据病毒进行检查。

15.6.1. 为邮件传输代理或邮件发送代理配置 Spam 过滤

您可以过滤邮件传输代理(MTA)、邮件发送代理(MDA)或邮件用户代理(MUA)中的垃圾邮件。本章论述了 MTA 和 MDA 中的垃圾邮件过滤。

15.6.1.1. 在邮件传输代理中配置 Spam 过滤

Red Hat Enterprise Linux 7 提供两个主要 MTA：Postfix 和 Sendmail。

有关如何安装和配置 MTA 的详情，请参考 第 15.3 节 “邮件传输代理”。

使用 Sendmail 可以在 MTA 端停止垃圾邮件，其具有多个反垃圾邮件功能：标头检查、转发拒绝、访问数据库和 发送者信息检查。如需更多信息，请参阅 第 15.3.2.5 节 “停止 Spam”。

此外，Postfix 和 Sendmail 都可以与第三方邮件过滤器（通配符）配合使用，以过滤邮件处理链中的垃圾邮件和病毒。如果是 Postfix，则 milters 的支持直接包含在 postfix 软件包中。如果是

Sendmail，您需要安装 sendmail-milter 包，以便能够使用 milters。

15.6.1.2. 在邮件发送代理中配置 Spam 过滤

Red Hat Enterprise Linux 包括两个主要 MDA，Procmail 和 mail 实用程序。如需更多信息，请参阅 第 15.2.2 节 “邮件发送代理”。

要在 MDA 中停止垃圾邮件，Procmail 用户可以安装 spamassin 包中名为 SpamAssassin 的第三方软件。SpamAssassin 是垃圾邮件检测系统，使用多种方法识别传入邮件中的垃圾邮件。有关 Spamassin 安装、配置和部署的更多信息，请参阅 第 15.4.2.6 节 “垃圾邮件过滤器” 或如何配置 Spamassassin 以过滤服务器上的所有传入邮件？红帽知识库文章.有关 SpamAssassin 的更多信息，请参阅 SpamAssassin 项目网站。

警告

请注意，SpamAssasin 是第三方软件，红帽不支持使用它。

spamassassin 软件包只能通过 Extra Packages for Enterprise Linux(EPEL) 存储库提供。要了解更多有关使用 EPEL 软件仓库的信息，请参阅 第 15.6.3 节 “使用 EPEL 存储库安装防垃圾邮件和防病毒软件”。

要了解有关红帽如何处理第三方软件以及红帽为其提供何种级别的支持的更多信息，请参阅红帽全球支持服务如何处理第三方软件、驱动程序和/或未经认证的硬件/管理程序或客户机操作系统？红帽知识库文章.

15.6.2. 配置防病毒保护

为保护您的系统免受病毒攻击，您可以安装 ClamAV，这是一款开源反病毒引擎，用于检测熵、病毒、恶意软件和其他恶意软件。有关 ClamAV 的更多信息，请参阅 ClamAV 项目网站。

警告

请注意，ClamAV 是第三方软件，红帽不支持使用它。

clam av、c lamav-data、clamav-server 和 clamav-update 软件包仅在企业 Linux 的 Extra Packages(EPEL)存储库中可用。要了解更多有关使用 EPEL 软件仓库的信息，请参阅 第 15.6.3 节 “使用 EPEL 存储库安装防垃圾邮件和防病毒软件”。

要了解有关红帽如何处理第三方软件以及红帽为其提供何种级别的支持的更多信息，请参阅红帽全球支持服务如何处理第三方软件、驱动程序和/或未经认证的硬件/管理程序或客户机操作系统？红帽知识库文章.

启用了 EPEL 存储库后，以 root 用户身份运行以下命令来安装 ClamAV：

```
~]# yum install clamav clamav-data clamav-server clamav-update
```

15.6.3. 使用 EPEL 存储库安装防垃圾邮件和防病毒软件

EPEL 是 Fedora 特殊兴趣组，可为红帽企业 Linux 创建、维护和管理一组高质量的额外软件包。如需更多信息，请参阅 Fedora EPEL 网站。

要使用 EPEL 存储库，请下载 红帽企业 Linux 7 的最新版本 epel-release 软件包。您还可以以 root 用户身份运行以下命令：

```
~]# yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpmzu
```

第一次使用 EPEL 存储库时，您需要使用 GPG 公钥进行身份验证。如需更多信息，请参阅 Fedora 软件包签名密钥。

15.7. 其它资源

以下是电子邮件应用程序的其他文档列表。

15.7.1. 安装的文档

- 有关配置 Sendmail 的信息包含在 sendmail 和 sendmail-cf 包中。
  - /usr/share/sendmail-cf/README - 包含有关 m4 宏处理器、Sendmail 的文件位置、支持的邮件发送程序、如何访问增强功能等信息。

    此外，sendmail 和 别名 man page 包含有关各种 Sendmail 选项和正确配置 Sendmail /etc/mail/aliases 文件的有用信息。
- /usr/share/doc/postfix- version-number/ - 包含有关如何配置 Postfix 的大量信息。使用 Postfix 的版本号替换 version-number。
- /usr/share/doc/fetchmail- version-number/ - 包含 FEATURES 文件中 Fetchmail 功能的完整列表和简介 常见问题 文档。使用 Fetchmail 的版本号替换 version-number。
- /usr/share/doc/procmail- version-number/ - 包含提供 Procmail 概述的 README 文件、探讨所有程序功能的 FEATURES 文件，以及含有许多常见配置问题的答案的 常见问题解答 文件。使用 Procmail 的版本号替换 version-number。

  在学习 Procmail 的工作原理和创建新配方时，以下 Procmail man page 非常宝贵：
  - Procmail - 概述 Procmail 的工作原理，以及与过滤电子邮件相关的步骤。
  - procmailrc - 说明用于构造配方的 rc 文件格式。
  - procmailex - 给出了许多实用的、真实的 Procmail 方法论示例。
  - procmailsc - 说明 Procmail 用于将特定方法与邮件相匹配的加权评分技术。
  - /usr/share/doc/spamassassin- version-number/ - 包含与 SpamAssassin 相关的大量信息。使用 spamassassin 软件包的版本号替换 version-number。

15.7.2. 在线文档

- 如何使用 TLS 配置 Postfix？- 描述将 Postfix 配置为使用 TLS 的红帽知识库文章。
- 如何配置 Sendmail 智能主机 - 描述配置 sendmail 智能主机的红帽知识库解决方案。
- http://www.sendmail.org/ - 提供对 Sendmail 功能、文档和配置示例的全面技术分类。
- http://www.sendmail.com/ - 包含有关 Sendmail 的新闻、访谈和文章，包括更多可用选项的扩展视图。
- http://www.postfix.org/ - Postfix 项目主页包含大量有关 Postfix 的信息。邮件列表是查找信息特别好的地方。
- http://www.fetchmail.info/fetchmail-FAQ.html - 有关 Fetchmail 的详细常见问题解答。
- http://www.spamassassin.org/ - SpamAssassin 项目的官方站点。

15.7.3. 相关书

- Sendmail Milters：由 Bryan Comles 和 Marcia Flynt 查找 Spam 的指南 ； Addison-Wesley - 帮助自定义邮件过滤器的良好 Sendmail 指南。
- 带 Eric Allmanet al. 的 Bryaniffles Sendmail；O'Reilly 和 Associates - 在原始 Delivermail 和 Sendmail 创建者的帮助下编写的良好 Sendmail 参考。
- 删除 Spam：Gegerff Mulligan 的电子邮件处理和过滤 ； Addison-Wesley Publishing Company - 此卷着眼于电子邮件管理员使用的各种方法（如 Sendmail 和 Procmail）来管理垃圾邮件问题。
- Internet 电子邮件协议：Kevin Johnson 的开发人员指南 ； Addison-Wesley 发布公司 - 提供对主要电子邮件协议及其提供的安全性的非常全面的回顾。
- Dianna Mullet 和 Kevin Mullet 管理 IMAP ；O'Reilly & Associates - 详细说明配置 IMAP 服务器所需的步骤。

第 16 章 文件和打印服务器

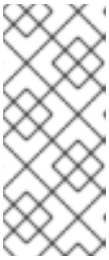
本章将引导您完成 Samba 的安装和配置，这是 服务器信息块(SMB)和 通用互联网文件系统 (CIFS)协议的开源实施，以及由 Red Hat Enterprise Linux 提供的主 FTP 服务器。此外，还介绍了如何使用 "打印设置" 工具来配置打印机。

16.1. SAMBA

Samba 在红帽企业 Linux 中实施服务器消息块(SMB)协议。SMB 协议用于访问服务器上的资源，如文件共享和共享打印机。此外，Samba 实施由 Microsoft Windows 使用的分布式计算环境远程过程调用 (DCE RPC)协议。

您可以以以下方式运行 Samba：

- Active Directory(AD)或 NT4 域成员
- 独立服务器
- NT4 主域控制器(PDC)或备份域控制器(BDC)



注意

红帽仅在支持 NT4 域的现有安装中支持这些模式。红帽建议不要设置新的 Samba NT4 域，因为 Microsoft 操作系统稍后于 Windows 7 和 Windows Server 2008 R2 不支持 NT4 域。

独立于安装模式，您可以选择共享目录和打印机。这可让 Samba 充当文件和打印服务器。

注意

红帽不支持将 Samba 作为 AD 域控制器（DC）运行。

16.1.1. Samba 服务

Samba 提供以下服务：

smbd

此服务使用 SMB 协议提供文件共享和打印服务。另外，该服务负责资源锁定和验证连接用户。smb systemd 服务启动并停止 smbd 守护进程。

要使用 smbd 服务，请安装 samba 软件包。

nmbd

此服务通过 IPv4 协议使用 NetBIOS 提供主机名和 IP 解析。除了名称解析之外，nmbd 服务还支持浏览 SMB 网络来查找域、工作组、主机、文件共享和打印机。为此，服务可将此信息直接报告给广播客户端，或者将其转发到本地或主浏览器。nmb systemd 服务启动并停止 nmbd 守护进程。

请注意，现代 SMB 网络使用 DNS 解析客户端和 IP 地址。

要使用 nmbd 服务，请安装 samba 软件包。

winbindd

winbindd 服务为名称服务交换机(NSS)提供接口，以使用 AD 或 NT4 域用户和组在本地系统上。例如，这使域用户能够向 Samba 服务器托管的服务或其他本地服务进行身份验证。winbind systemd 服务启动并停止 winbindd 守护进程。

如果将 Samba 设置为域成员，必须在 smb d 服务之前启动 winbind d。否则，本地系统无法使用域用户和组。

要使用 winbindd 服务，请安装 samba-winbind 软件包。

重要

红帽仅支持将 Samba 作为带有 winbindd 服务的服务器运行，以便为本地系统提供域用户和组。由于某些限制，如缺少 Windows 访问控制列表(ACL)支持和 NT LAN Manager(NTLM)回退，这些用例目前不支持使用带有 Samba 的系统安全服务守护进程(SSSD)。详情请查看 Red Hat 知识库文章在 IdM 客户端上运行的 Samba 文件服务器的支持状态是什么，或直接注册了 SSSD 用作客户端守护进程的 AD 客户端。

16.1.2. 使用 testparm 实用程序验证 smb.conf 文件

testparm 实用程序验证 /etc/samba/smb.conf 文件中的 Samba 配置是否正确。实用程序检测无效的参数和值，但也检测不正确的设置，如 ID 映射。如果 testparm 报告没有问题，Samba 服务将成功加载 /etc/samba/smb.conf 文件。请注意，testparm 无法验证配置的服务是否可用或按预期工作。

重要

红帽建议在每次修改此文件后使用 testparm 来验证 /etc/samba/smb.conf 文件。

要验证 /etc/samba/smb.conf 文件，请以 root 用户身份运行 testparm 实用程序。如果 testparm 报告了配置中不正确的参数、值或其他错误，请修复问题并再次运行 实用程序。

例 16.1. 使用 testparm

以下输出报告了一个不存在的参数和不正确的 ID 映射配置：

```
~]# testparm
Load smb config files from /etc/samba/smb.conf
rlimit_max: increasing rlimit_max (1024) to minimum Windows limit (16384)
Unknown parameter encountered: "log levell"
Processing section "[example_share]"
Loaded services file OK.
ERROR: The idmap range for the domain * (tdb) overlaps with the range of DOMAIN (ad)!

Server role: ROLE_DOMAIN_MEMBER

Press enter to see a dump of your service definitions

# Global parameters
[global]
 ...

[example_share]
 ...
```

16.1.3. 了解 Samba 安全模式

/etc/samba/smb.conf 文件中的 [global] 部分中的 security 参数管理 Samba 如何验证连接到该服务的用户的身份。根据您在其中安装 Samba 的模式，参数必须设为不同的值：

- 在 AD 域成员中，设置 security = ads。

  在这个模式中，Samba 使用 Kerberos 来验证 AD 用户。

  有关将 Samba 设置为域成员的详情，请参考 第 16.1.5 节 “将 Samba 设置为域成员”。

- 在单机服务器上，设置 security = user。

  在这个模式中，Samba 使用本地数据库验证连接用户。

  有关将 Samba 设置为单机服务器的详情，请参考 第 16.1.4 节 “将 Samba 设置为单机服务器”。

- 在 NT4 PDC 或 BDC 上，设置 security = user.

  在此模式中，Samba 将用户身份验证到本地或 LDAP 数据库。

- 在 NT4 域成员上，设置 security = domain。

  在此模式中，Samba 验证将用户连接到 NT4 PDC 或 BDC 的身份。您不能在 AD 域成员中使用这个模式。

  有关将 Samba 设置为域成员的详情，请参考 第 16.1.5 节 “将 Samba 设置为域成员”。

详情请查看 smb.conf(5)man page 中的 security 参数描述。

16.1.4. 将 Samba 设置为单机服务器

在某些情况下，管理员希望设置不属于某个域成员的 Samba 服务器。在此安装模式中，Samba 对用户进行本地数据库而非中央 DC 身份验证。另外，您可以启用客户机访问，允许用户在没有身份验证的情况下连接到一个或多个服务。

16.1.4.1. 为单机服务器设置服务器配置

将 Samba 设置为单机服务器：

将 Samba 设置为单机服务器

1. **安装 samba 软件包：**

```
~]# yum install samba
```

2. 编辑 **/etc/samba/smb.conf 文件并设置以下参数：**

```
[global]
 workgroup = Example-WG
 netbios name = Server
 security = user

 log file = /var/log/samba/%m.log
 log level = 1
```

此配置在 Example-learning 工作组中定义了一个名为 Server 的单机服务器。此外，此配置启用了最小级别(1)的日志记录，日志文件将存储在 /var/log/samba/ 目录中。Samba 会将 日志文件 参数中的 %m 宏扩展到连接客户端的 NetBIOS 名称。这可为每个客户端启用独立的日志文件。

详情请查看 **smb.conf(5)man page 中的参数描述。**

3. **配置文件或打印机共享.请参阅：**

- **第 16.1.6 节 “在 Samba 服务器上配置文件共享”**
- **第 16.1.7 节 “设置 Samba 打印服务器”**

4. 验证 **/etc/samba/smb.conf 文件：**

```
~]# testparm
```

详情请查看 第 16.1.2 节 “使用 testparm 实用程序验证 smb.conf 文件”。

5. 如果您设置了需要身份验证的共享，请创建用户帐户。详情请查看 第 16.1.4.2 节 “创建和启用本地用户帐户”。

6. 打开所需的端口并使用 firewall-cmd 工具重新载入防火墙配置：

```
~]# firewall-cmd --permanent --add-port={139/tcp,445/tcp}
~]# firewall-cmd --reload
```

7. 启动 smb 服务：

```
~]# systemctl start smb
```

8. 另外，还可在系统引导时启用 smb 服务自动启动：

```
~]# systemctl enable smb
```

16.1.4.2. 创建和启用本地用户帐户

要让用户在连接到共享时进行身份验证，您必须在操作系统和 Samba 数据库中在 Samba 主机上创建帐户。Samba 要求操作系统帐户验证文件系统对象上的访问控制列表(ACL)和 Samba 帐户，从而对连接用户进行身份验证。

如果您使用 passdb backend = tdbsam 默认设置，Samba 会将用户帐户存储在 /var/lib/samba/private/passdb.tdb 数据库中。

例如，要创建 Samba 用户示例：

创建 Samba 用户

1. 创建操作系统帐户：

```
~]# useradd -M -s /sbin/nologin example
```

上一命令在不创建主目录的情况下添加 example 帐户。如果帐户仅用于对 Samba 进行身

份验证，请将 /sbin/nologin 命令指定为 shell，以防止帐户在本地登录。

2. 为操作系统帐户设置密码以启用它：

```
~]# passwd example
Enter new UNIX password: password
Retype new UNIX password: password
passwd: password updated successfully
```

Samba 不会使用操作系统帐户中的密码集进行身份验证。然而，您需要设置密码才能启用帐户。如果一个帐户被禁用，当这个用户连接时，Samba 会拒绝访问。

3. 将用户添加到 Samba 数据库，并为帐户设置密码：

```
~]# smbpasswd -a example
New SMB password: password
Retype new SMB password: password
Added user example.
```

当使用此帐户连接到 Samba 共享时，使用此密码进行验证。

4. 启用 Samba 帐户：

```
~]# smbpasswd -e example
Enabled user example.
```

16.1.5. 将 Samba 设置为域成员

运行 AD 或 NT4 域的管理员通常希望使用 Samba 将其红帽企业 Linux 服务器作为该域的成员加入。这可让您：

- 访问其他域成员上的域资源
- 向本地服务（如 sshd）验证域用户
- 托管在服务器上的共享目录和打印机，以充当文件和打印服务器

16.1.5.1. 加入域

要将 Red Hat Enterprise Linux 系统加入到域中：

将 Red Hat Enterprise Linux 系统加入一个域

1. 安装以下软件包：

```
~]# yum install realmd oddjob-mkhomedir oddjob samba-winbind-clients \
  samba-winbind samba-common-tools
```

2. 要在域成员中共享目录或打印机，请安装 samba 软件包：

```
~]# yum install samba
```

3. 如果您加入 AD，还要安装 samba-winbind-krb5-locator 软件包：

```
~]# yum install samba-winbind-krb5-locator
```

此插件可让 Kerberos 根据使用 DNS 服务记录的 AD 站点查找密钥分发中心(KDC)。

4. 另外，还可重命名现有的 /etc/samba/smb.conf Samba 配置文件：

```
~]# mv /etc/samba/smb.conf /etc/samba/smb.conf.old
```

5. 加入域。例如，要加入名为 ad.example.com的域

```
~]# realm join --membership-software=samba --client-software=winbind
ad.example.com
```

使用上一命令，realm 工具会自动：

- 为 ad.example .com 域中的成员资格创建 /etc/samba/smb. conf 文件

- 将用户和组查找的 winbind 模块添加到 /etc/nsswitch.conf 文件中

- 更新 /etc/pam.d/ 目录中的可插拔验证模块(PAM)配置文件

- 启动 winbind 服务并使服务在系统引导时启动

有关 realm 实用程序的详情，请查看 域(8)man page 和 Red Hat Windows 集成指南 中 的对应章节。

6. （可选）在 /etc/samba/smb.conf 文件中设置替代 ID 映射后端或自定义 ID 映射设置。详情请查看 第 16.1.5.3 节 “了解 ID 映射”。

7. （可选）验证配置。请参阅 第 16.1.5.2 节 “验证 Samba 正确加入为域成员”。

8. 验证 winbindd 是否正在运行：

```
~]# systemctl status winbind
```

重要

要启用 Samba 查询域用户和组信息，必须在启动 smb d 之前运行 winbind d 服务。

9. 如果您安装了 samba 软件包来共享目录和打印机，请启动 smbd 服务：

```
~]# systemctl start smb
```

16.1.5.2. 验证 Samba 正确加入为域成员

加入红帽企业 Linux 作为域成员后，您可以运行不同的测试来验证该连接是否成功。请参阅：

- “验证操作系统是否可以恢复域用户帐户和组”一节

- “验证 AD 域用户是否可以获取 Kerberos 票据”一节
- “列出可用的域”一节

验证操作系统是否可以恢复域用户帐户和组

使用 getent 实用程序验证操作系统是否可以检索域用户和组。例如：

- 查询 AD 域中 的管理员 帐户：

```
~]# getent passwd AD\\administrator
AD\administrator:*:10000:10000::/home/administrator@AD:/bin/bash
```

- 查询 AD 域中 Domain Users 组的成员：

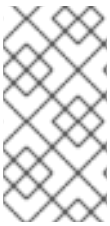
```
~]# getent group "AD\\Domain Users"
AD\domain users:x:10000:user
```

如果 命令正常工作，请验证您可以在设置文件和目录的权限时使用域用户和组。例如，将 /srv/samba/example.txt 文件的所有者设置为 AD\administrator，组设置为 AD\Domain Users ：

```
~]# chown "AD\administrator":"AD\Domain Users" /srv/samba/example.txt
```

验证 AD 域用户是否可以获取 Kerberos 票据

在 AD 环境中，用户可以从 DC 获取 Kerberos 票据。例如，验证 管理员用户 是否可以获取 Kerberos 票据：



注意

若要使用 kinit 和 klist 实用程序，可在 Samba 域成员 上安装 krb5-workstation 软件包。

获取 Kerberos 票据

1. 获取 administrator@AD.EXAMPLE.COM 主体的票据：

```
~]# kinit administrator@AD.EXAMPLE.COM
```

2. 显示缓存的 Kerberos ticket：

```
~]# klist
Ticket cache: KEYRING:persistent:0:0
Default principal: administrator@AD.EXAMPLE.COM

Valid starting     Expires       Service principal
11.09.2017 14:46:21 12.09.2017 00:46:21
krbtgt/AD.EXAMPLE.COM@AD.EXAMPLE.COM
 renew until 18.09.2017 14:46:19
```

列出可用的域

要通过 winbindd 服务列出所有可用的域，请输入：

```
~]# wbinfo --all-domains
```

如果 Samba 成功作为域成员加入，命令可显示内置和本地主机名，以及包含受信任域的成员。

例 16.2. 显示可用域

```
~]# wbinfo --all-domains
BUILTIN
SAMBA-SERVER
AD
```

16.1.5.3. 了解 ID 映射

Windows 域通过唯一的安全标识符(SID)区分用户和组。但是，Linux 需要为每个用户和组群有唯一的 UID 和 GID。如果您以域成员身份运行 Samba，winbindd 服务负责向操作系统提供域用户和组的信息。

要启用 winbindd 服务为用户和组向 Linux 提供唯一 ID，您必须在 /etc/samba/smb.conf 文件中为以下目的配置 ID 映射：

- 本地数据库（默认域）
- Samba 服务器所属的 AD 或 NT4 域

- 每个用户必须能够访问这个 Samba 服务器上的资源的可信域

16.1.5.3.1. 规划 ID 范围

无论您是在 AD 中存储 Linux UID 和 GID，还是将 Samba 配置为生成它们，每个域配置都需要一个唯一的 ID 范围，不得与任何其他域重叠。

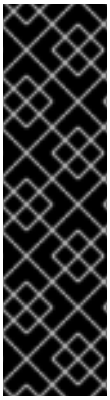
警告

如果您设置了重叠 ID 范围，Samba 无法正常工作。

例 16.3. 唯一的 ID 范围

以下显示了默认(*)、AD-DOM 和 TRUST-DOM 域的非覆盖 ID 映射范围。

```
[global]
...
idmap config * : backend = tdb
idmap config * : range = 10000-999999

idmap config AD-DOM:backend = rid
idmap config AD-DOM:range = 2000000-2999999

idmap config TRUST-DOM:backend = rid
idmap config TRUST-DOM:range = 4000000-4999999
```



重要

每个域只能分配一个范围。因此，在域范围之间有足够的空间。这可让您在域扩展后扩展范围。

如果您稍后将不同的范围分配给某个域，这些用户和组之前创建的文件和目录的所有权将会丢失。

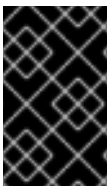
16.1.5.3.2. * 默认域

在域环境中，您可以为以下每个情况添加一个 ID 映射配置：

- Samba 服务器所属的域
- 每个可以访问 Samba 服务器的可信域

但是，对于所有其他对象，Samba 会从默认域分配 ID。这包括：

- 本地 Samba 用户和组
- Samba 内置帐户和组，如 BUILTIN\Administrators



重要

您必须配置默认域，如本节所述，才能使 Samba 正常运行。

默认域后端必须可写，才能永久存储分配的 ID。

对于默认域，您可以使用以下后端之一：

tdb

当您将默认域配置为使用 tdb 后端时，设置一个足够大的 ID 范围，使其包含将来将要创建并且不属于定义的域 ID 映射配置的对象。

例如，在 /etc/samba/smb.conf 文件中的 [global] 部分中设置以下内容：

```
idmap config * : backend = tdb
idmap config * : range = 10000-999999
```

详情请查看 第 16.1.5.4.1 节 “使用 tdb ID 映射后端”。

autorid

当您将默认域配置为使用 自动后端 时，可以为域添加额外的 ID 映射配置是可选的。

例如，在 /etc/samba/smb.conf 文件中的 [global] 部分中设置以下内容：

```
idmap config * : backend = autorid
idmap config * : range = 10000-999999
```

详情请查看 配置 自动 后端。

16.1.5.4. 不同的 ID 映射后端

Samba 为特定配置提供不同的 ID 映射后端。最常用的后端是：

表 16.1. 常用的 ID 映射后端

后端	使用案例
tdb	* 默认域
ad	仅限 AD 域
rid	AD 和 NT4 域
autorid	AD、NT4 和 * 默认域

以下小节介绍了后端的优势、使用后端的建议场景以及配置方法。

16.1.5.4.1. 使用 tdb ID 映射后端

winbindd 服务默认使用可写 tdb ID 映射后端来存储安全标识符(SID)、UID 和 GID 映射表。这包括本地用户、组和内置主体。

仅将此后端用于 * 默认域。例如：

```
idmap config * : backend = tdb
idmap config * : range = 10000-999999
```

有关 * 默认域的详情，请参考 第 16.1.5.3.2 节 “* 默认域”。

16.1.5.4.2. 使用 ad ID 映射后端

ad ID 映射后端实施只读 API，以从 AD 读取帐户和组信息。它具有以下优点：

- 所有用户和组群设置都集中存储在 AD 中。
- 使用这个后端的所有 Samba 服务器中的用户和组群 ID 是一致的。
- ID 不会存储在本地数据库中（本地数据库可能会被损坏），因此文件所有者不会丢失。

ad 后端从 AD 中读取以下属性：

表 16.2. 属性从用户和组对象中读取 ad 后端

AD Attribute Name	对象类型	映射到
sAMAccountName	用户和组群	用户和组名称，具体取决于对象
uidNumber	用户	用户 ID（UID）
gidNumber	组	组 ID（GID）
loginShell [a]	用户	用户 shell 的路径
unixHomeDirectory	用户	用户主目录的路径
primaryGroupID [b]	用户	主组群 ID

[a] 如果您设置了 **idmap config *DOMAIN*:unix_nss_info = yes**，则 Samba 仅读取此属性。

[b] 如果您设置了 **idmap config *DOMAIN*:unix_primary_group = yes**，则 Samba 仅读取此属性。

后端 的先决条件

使用 ad ID 映射后端：

- 用户和组必须在 AD 中设置唯一 ID，并且 ID 必须在 /etc/samba/smb.conf 文件中配置的范围范围内。ID 范围之外的对象在 Samba 服务器上不可用。

- 用户和组必须在 AD 中设置所有必需的属性。如果缺少所需的属性，该用户或组将无法在 Samba 服务器中可用。所需的属性取决于您的配置。请参阅 表 16.2 “属性从用户和组对象中读取 ad 后端”。

配置 ad Backendd

将 Samba AD 成员配置为使用 ad ID 映射后端：

在域成员中配置 ad Backd

1. 编辑 /etc/samba/smb.conf 文件中的 [global] 部分：

   a. 如果默认域(*)不存在，请为它添加 ID 映射配置。例如：

   ```
   idmap config * : backend = tdb
   idmap config * : range = 10000-999999
   ```

   有关默认域配置的详情，请参考 第 16.1.5.3.2 节 “* 默认域”。

   b. 为 AD 域启用 ad ID 映射后端：

   ```
   idmap config DOMAIN : backend = ad
   ```

   c. 设置分配给 AD 域中用户和组的 ID 范围。例如：

   ```
   idmap config DOMAIN : range = 2000000-2999999
   ```

   重要

   范围不得与这个服务器上的任何其他域配置重叠。此外，范围必须足够大，以便包含将来分配的所有 ID。详情请查看 第 16.1.5.3.1 节 “规划 ID 范围”。

d. 设置 Samba 在读取 AD 属性时使用 RFC 2307 模式：

```
idmap config DOMAIN : schema_mode = rfc2307
```

e. 要让 Samba 从对应的 AD 属性读取登录 shell 和用户主目录的路径，请设置：

```
idmap config DOMAIN : unix_nss_info = yes
```

或者，您可以设置应用于所有用户的统一的域范围的主目录路径和登录 shell。例如：

```
template shell = /bin/bash
template homedir = /home/%U
```

有关变量替换的详情，请查看 smb.conf(5)man page 中的 VARIABLE SUBSTITUTIONS 部分。

f. 默认情况下，Samba 使用用户对象的 primaryGroupID 属性作为 Linux 上用户的主组。或者，您可以将 Samba 配置为使用 the gidNumber 属性中设置的值：

```
idmap config DOMAIN : unix_primary_group = yes
```

2. 验证 /etc/samba/smb.conf 文件：

```
~]# testparm
```

详情请查看 第 16.1.2 节 “使用 testparm 实用程序验证 smb.conf 文件”。

3. 重新载入 Samba 配置：

```
~]# smbcontrol all reload-config
```

4. 验证设置是否按预期工作。请参阅 “验证操作系统是否可以恢复域用户帐户和组”一节。

详情请查看 smb.conf(5)和 idmap_ad(8)man page。

16.1.5.4.3. 使用丢弃 的 ID 映射后端

Samba 可以使用 Windows SID 的相对标识符(RID)在 Red Hat Enterprise Linux 上生成 ID。

注意

RID 是 SID 的最后部分。例如，如果用户的 SID 是 S-1-5-21-5421822485-1151247151-421485315-30014，则 30014 是对应的 RID。详情请查看 Samba 如何计算本地 ID，请参阅 idmap_rid(8)man page。

删除的 ID 映射后端实施只读 API，以根据 AD 和 NT4 域的算法映射方案计算帐户和组信息。配置后端时，您必须在 idmap config DOMAIN : range 参数中设置最低和最高的 RID。Samba 不会映射比这个参数中设置低或更高 RID 的用户或组。

重要

作为只读后端，删除 无法 分配新 ID，例如用于 BUILTIN 组。因此，请勿将此后端用于 * 默认域。

优点

- 所有在配置范围内具有 RID 的域用户和组都会自动在域成员中可用。
- 您不需要手动分配 ID、主目录和登录 shell。

缺陷

- 所有域用户可以获得相同的登录 shell 和主目录。但是，您可以使用变量。
- 如果所有删除 后端 都使用相同的 ID 范围设置，则用户和组 ID 仅在 Samba 域成员之间相同。
- 您不能阻止单独的用户或组在域成员中可用。只有超出配置范围以外的用户和组才会包括。
- 根据 winbindd 服务用于计算 ID 的公式，如果不同域中的对象具有相同的 RID，则重复 ID

可能会在多域环境中发生。

配置 删除 的后端

将 Samba 域成员配置为使用丢弃 ID 映射后端：

在 域 成员中配置删除的后端

1. 编辑 /etc/samba/smb.conf 文件中的 [global] 部分：

   a. 如果默认域(*)不存在，请为它添加 ID 映射配置。例如：

   ```
   idmap config * : backend = tdb
   idmap config * : range = 10000-999999
   ```

   有关默认域配置的详情，请参考 第 16.1.5.3.2 节 “* 默认域”。

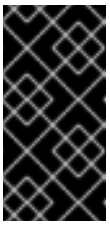
   b. 为域启用丢弃的 ID 映射后端：

   ```
   idmap config DOMAIN : backend = rid
   ```

   c. 设置一个足够大的范围，使其包含将来将分配的所有 RID。例如：

   ```
   idmap config DOMAIN : range = 2000000-2999999
   ```

   Samba 会忽略此域中 RID 不在 范围内的用户和组。

   重要

   范围不得与这个服务器上的任何其他域配置重叠。详情请查看 第 16.1.5.3.1 节 “规划 ID 范围”。

   d. 设置分配给所有映射用户的 shell 和主目录路径。例如：

   ```
   template shell = /bin/bash
   template homedir = /home/%U
   ```

-

有关变量替换的详情，请查看 smb.conf(5)man page 中的 VARIABLE SUBSTITUTIONS 部分。

2. 验证 /etc/samba/smb.conf 文件：

```
~]# testparm
```

详情请查看 第 16.1.2 节 “使用 testparm 实用程序验证 smb.conf 文件”。

3. 重新载入 Samba 配置：

```
~]# smbcontrol all reload-config
```

4. 验证设置是否按预期工作。请参阅 “验证操作系统是否可以恢复域用户帐户和组”一节。

16.1.5.4.4. 使用自动 ID 映射后端

自动后端 的工作方式类似于丢弃的 ID 映射后端，但可以自动为不同的域分配 ID。这可让您 在以下情况下使用自动 后端：

- 仅用于 * 默认域。
- 对于 * 默认域和其他域，无需为每个额外域创建 ID 映射配置。
- 仅供特定域使用.

优点

- 所有在配置范围内计算 UID 和 GID 的域用户和组都会在域成员中自动可用。
- 您不需要手动分配 ID、主目录和登录 shell。

- 没有重复的 ID，即使多域环境中的多个对象有相同的 RID。

缺陷

- 在 Samba 域成员中用户和组群 ID 不相同。

- 所有域用户可以获得相同的登录 shell 和主目录。但是，您可以使用变量。

- 您不能阻止单独的用户或组在域成员中可用。只有计算 UID 或 GID 不在配置范围内的用户和组才会包括。

配置 自动 后端

将 Samba 域成员配置为使用 * 默认域的 自动 ID 映射后端：

注意

如果您对默认域使用 autorid，可以为域添加额外的 ID 映射配置是可选的。

在域 成员中配置自动 后端

1. 编辑 /etc/samba/smb.conf 文件中的 [global] 部分：

   a. 为 * 默认域启用 自动 ID 映射后端：

   ```
   idmap config * : backend = autorid
   ```

   b. 设置一个足够大的范围来为所有现有和将来的对象分配 ID。例如：

   ```
   idmap config * : range = 10000-999999
   ```

   Samba 忽略在此域中计算 ID 不在范围范围内的用户和组。有关后端如何计算 ID 的详情，请查看 idmap_autorid(8)man page 中的 THE MAPPING FORMULAS 部分。

警告

设置范围并开始使用 Samba 后，您只能增加范围的上限。对范围的任何其他更改都可能导致新的 ID 分配，从而释放文件所有权。

c. 另外，还可设置范围大小。例如：

```
idmap config * : rangesize = 200000
```

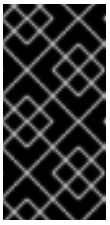
Samba 会为每个域的对象分配这个数量的连续 ID，直到获取 idmap config * : range 参数中设置范围中的所有 ID。详情请查看 idmap_autorid(8)man page 中的 rangesize 参数描述。

d. 设置分配给所有映射用户的 shell 和主目录路径。例如：

```
template shell = /bin/bash
template homedir = /home/%U
```

有关变量替换的详情，请查看 smb.conf(5)man page 中的 VARIABLE SUBSTITUTIONS 部分。

e. 另外，还可为域添加额外的 ID 映射配置。如果没有单个域的配置可用，Samba 使用之前配置的 * 默认域中的 自动后端 设置来计算 ID。



重要

如果您为单个域配置额外的后端，则所有 ID 映射配置的范围不得互相重叠。详情请查看 第 16.1.5.3.1 节 “规划 ID 范围”。

2. 验证 /etc/samba/smb.conf 文件：

```
~]# testparm
```

详情请查看 第 16.1.2 节 “使用 testparm 实用程序验证 smb.conf 文件”。

3. 重新载入 Samba 配置：

```
~]# smbcontrol all reload-config
```

4. 验证设置是否按预期工作。请参阅 “验证操作系统是否可以恢复域用户帐户和组”一节。

16.1.6. 在 Samba 服务器上配置文件共享

要将 Samba 用作文件服务器，请将共享添加到独立或域成员配置的 /etc/samba/smb.conf 文件中。

您可以使用以下任一方法添加共享：

- POSIX ACL.请参阅 第 16.1.6.1 节 “设置使用 POSIX ACL 的共享”。
- 细粒度 Windows ACL.请参阅 第 16.1.6.2 节 “设置使用 Windows ACL 的共享”。

16.1.6.1. 设置使用 POSIX ACL 的共享

作为 Linux 服务，Samba 支持与 POSIX ACL 的共享。它们允许您使用诸如 chmod 等实用程序在 Samba 服务器上本地管理权限。如果共享存储在支持扩展属性的文件系统中，您可以使用多个用户和组定义 ACL。

注意

如果您需要使用细粒度 Windows ACL，请参阅 第 16.1.6.2 节 “设置使用 Windows ACL 的共享”。

在添加共享之前，请先设置 Samba。请参阅：

- 第 16.1.4 节 “将 Samba 设置为单机服务器”

- 第 16.1.5 节 “将 Samba 设置为域成员”

16.1.6.1.1. 添加使用 POSIX ACL 的共享

创建名为 example 的共享，该共享提供 /srv/samba/example/ 目录的内容并使用 POSIX ACL：

添加使用 POSIX ACL 的共享

1. （可选）创建 文件夹（如果文件夹不存在）。例如：

```
~]# mkdir -p /srv/samba/example/
```

2. 如果您以 enforcing 模式运行 SELinux，请在目录中设置 samba_share_t 上下文：

```
~]# semanage fcontext -a -t samba_share_t "/srv/samba/example(/.*)?"
~]# restorecon -Rv /srv/samba/example/
```

3. 在目录中设置文件系统 ACL。详情请查看 第 16.1.6.1.2 节 “设置 ACL”。

4. 将 示例 共享添加到 /etc/samba/smb.conf 文件中。例如，添加启用了共享的写操作：

```
[example]
 path = /srv/samba/example/
 read only = no
```

注意

无论文件系统 ACL 是什么；如果您没有 只设置只读 = no，Samba 都会以只读模式共享该目录。

5. 验证 /etc/samba/smb.conf 文件：

```
~]# testparm
```

详情请查看 第 16.1.2 节 “使用 testparm 实用程序验证 smb.conf 文件”。

6. 打开所需的端口并使用 firewall-cmd 工具重新载入防火墙配置：

```
~]# firewall-cmd --permanent --add-service=samba
~]# firewall-cmd --reload
```

7. 重启 smb 服务：

```
~]# systemctl restart smb
```

8. 另外，还可启用 smb 服务在引导时自动启动：

```
~]# systemctl enable smb
```

16.1.6.1.2. 设置 ACL

使用 POSIX ACL 支持的共享：

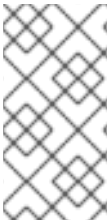
- 标准 Linux ACL.详情请查看 设置标准 Linux ACL。
- 扩展 ACL.详情请查看 设置扩展 ACL。

设置标准 Linux ACL

Linux 中的标准 ACL 支持为一个所有者、一个组和所有其他未定义用户设置权限。您可以使用 chown、chgrp 和 chmod 实用程序来更新 ACL。如果您需要精确控制，然后使用更复杂的 POSIX ACL，请参阅 设置扩展 ACL。

例如，要将 /srv/samba/example/ 目录的所有者设置为 root 用户，请为 Domain Users 组授予读写权限，并拒绝所有其他用户的访问权限：

```
~]# chown root:"Domain Users" /srv/samba/example/
~]# chmod 2770 /srv/samba/example/
```



注意

在目录上启用 set-group-ID(SGID)位会自动设置目录组的所有新文件和子目录的默认组，而不是将其设置为创建新目录条目的用户的主要组。

有关权限的详情，请查看 chown(1) 和chmod(1)man page。

设置扩展 ACL

如果文件系统中保存了共享目录的支持扩展 ACL，您可以使用它们设置复杂的权限。扩展 ACL 可以包含多个用户和组群的权限。

扩展 POSIX ACL 可让您使用多个用户和组配置复杂的 ACL。但是，您只能设置以下权限：

- 无权限
- 读权限
- 写权限
- 完整控制

如果您需要细粒度 Windows 权限，如 创建文件夹 / 附加数据，请将共享配置为使用 Windows ACL。请参阅 第 16.1.6.2 节 “设置使用 Windows ACL 的共享”。

在共享中使用扩展 POSIX ACL：

在共享上启用扩展 POSIX ACL

1. 在 /etc/samba/smb.conf 文件中的共享部分启用以下参数，以启用扩展 ACL 的 ACL 继承：

```
inherit acls = yes
```

详情请查看 smb.conf(5)man page 中的参数描述。

2. 重启 smb 服务：

```
~]# systemctl restart smb
```

3. 另外，还可启用 smb 服务在引导时自动启动：

```
~]# systemctl enable smb
```

4. 在目录中设置 ACL。有关使用扩展 ACL 的详情，请参考 第 5 章 访问控制列表。

例 16.4. 设置扩展 ACL

以下步骤为 Domain Admins 组设置读取、写入和执行权限，为 Domain Users 组设置读取和执行权限，并拒绝 /srv/samba/example/ 目录中其他人的访问权限：

设置扩展 ACL

a. 为主用户帐户组禁用自动授予权限：

```
~]# setfacl -m group::--- /srv/samba/example/
~]# setfacl -m default:group::--- /srv/samba/example/
```

目录的主要组还映射到动态 CREATOR GROUP 主体。当您在 Samba 共享中使用扩展 POSIX ACL 时，会自动添加这个主体，您无法将其删除。

b. 设置目录中的权限：

i. 向 Domain Admins 组授予读取、写入和执行权限：

```
~]# setfacl -m group:"DOMAIN\Domain Admins":rwx /srv/samba/example/
```

ii. 为 Domain Users 组授予读取和执行权限：

```
~]# setfacl -m group:"DOMAIN\Domain Users":r-x /srv/samba/example/
```

iii. 为其他 ACL 条目设置权限，以拒绝不匹配其他 ACL 条目的用户的访问：

```
~]# setfacl -R -m other::--- /srv/samba/example/
```

这些设置只适用于这个目录。在 Windows 中，这些 ACL 仅映射到此文件夹模式。

c. 启用上一步中设置的权限由此目录中创建的新文件系统对象继承：

```
~]# setfacl -m default:group:"DOMAIN\Domain Admins":rwx
/srv/samba/example/
~]# setfacl -m default:group:"DOMAIN\Domain Users":r-x /srv/samba/example/
~]# setfacl -m default:other::--- /srv/samba/example/
```

使用这些设置时，现在将主体的 此文件夹仅 设置为此 文件夹、子文件夹和文件。

Samba 将之前设置的权限映射到以下 Windows ACL：

主体	权限	适用于
DOMAIN\Domain Admins	完整控制	这个文件夹、子文件夹和文件
DOMAIN\Domain 用户	读和执行	这个文件夹、子文件夹和文件
每个人都 [a]	无	这个文件夹、子文件夹和文件
所有者（Unix Userpass:attributes[]*所有者*）[b]	完整控制	只限于这个文件夹
primary_group (Unix Userpass:attributes[]*primary_group*) [c]	无	只限于这个文件夹
创建者所有者 [d] [e]	完整控制	只适用于子文件夹和文件
创建组 [f]	无	只适用于子文件夹和文件

主体	权限	适用于

[a] Samba **从其他** ACL 条目映射此主体的权限。

[b] Samba 将目录的所有者映射到此条目。

[c] Samba 将目录的主组群映射到这个条目。

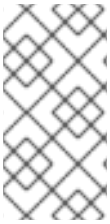
[d] 在新文件系统对象中，创建者会自动继承这个主体的权限。

[e] 在使用 POSIX ACL 的共享中不支持从 ACL 配置或删除这些主体。

[f] 在新文件系统对象中，创建器的主组群自动继承这个主体的权限。

16.1.6.1.3. 在共享中设置权限

若要选择限制或授予对 Samba 共享的访问权限，您可以在 /etc/samba/smb.conf 文件中的共享部分设置某些参数。



注意

如果用户、组或主机能够访问共享，则管理基于共享的权限。这些设置不会影响文件系统 ACL。

使用基于共享的设置来限制对共享的访问.例如，若要拒绝特定主机的访问：

配置基于用户和组的共享访问

借助基于用户和组的访问控制，您可以授予或拒绝特定用户和组对共享的访问权限。例如，要在 用户帐户 访问时允许 Domain Users 组的所有成员访问共享，请在共享的配置中添加以下参数：

```
valid users = +DOMAIN\"Domain Users"
invalid users = DOMAIN\user
```

无效的 users 参数具有高于 有效 users 参数的优先级。例如，如果 用户帐户 是 Domain Users 组的成员，则在使用上例时拒绝访问此帐户。

详情请查看 smb.conf(5)man page 中的参数描述。

配置基于主机的共享访问

通过基于主机的访问控制，您可以根据客户端的主机名、IP 地址或 IP 范围授予或拒绝对共享的访问。

例如，要启用 127.0.0.1 IP 地址、192.0.2.0/24 IP 范围和 client1.example.com 主机来访问共享，并拒绝 client2.example.com 主机的访问：

配置基于主机的共享访问

1. 在 /etc/samba/smb.conf 中的共享配置中添加以下参数：

```
hosts allow = 127.0.0.1 192.0.2.0/24 client1.example.com
hosts deny = client2.example.com
```

2. 重新载入 Samba 配置

```
~]# smbcontrol all reload-config
```

hosts deny 参数的优先级高于 hosts allow。例如，如果 client1.example.com 解析为 hosts allow 参数中列出的 IP 地址，则此主机的访问将被拒绝。

详情请查看 smb.conf(5)man page 中的参数描述。

16.1.6.2. 设置使用 Windows ACL 的共享

Samba 支持在共享和文件系统对象中设置 Windows ACL。这可让您：

- 使用精细 Windows ACL
- 使用 Windows 管理共享权限和文件系统 ACL

或者，您可以将共享配置为使用 POSIX ACL。详情请查看 第 16.1.6.1 节 “设置使用 POSIX ACL 的共享”。

16.1.6.2.1. 授予 SeDiskOperatorPrivilege 权限

只有具有授予 SeDiskOperatorPrivilege 特权的用户和组才能对使用 Windows ACL 的共享配置权限。例如，要将权限授予 DOMAIN\Domain Admins 组：

```
~]# net rpc rights grant "DOMAIN\Domain Admins" SeDiskOperatorPrivilege \
   -U "DOMAIN\administrator"
Enter DOMAIN\administrator's password:
Successfully granted rights.
```

注意

在域环境中，为域组授予 SeDiskOperatorPrivilege。这可让您通过更新用户的组成员资格来集中管理权限。

列出所有授予 SeDiskOperatorPrivilege 的 用户和组：

```
~]# net rpc rights list privileges SeDiskOperatorPrivilege \
   -U "DOMAIN\administrator"
Enter administrator's password:
SeDiskOperatorPrivilege:
 BUILTIN\Administrators
 DOMAIN\Domain Admins
```

16.1.6.2.2. 启用 Windows ACL 支持

要配置支持 Windows ACL 的共享，您必须在 Samba 中启用此功能。要全局启用所有共享，请在 /etc/samba/smb.conf 文件的 [global] 部分添加以下设置：

```
vfs objects = acl_xattr
map acl inherit = yes
store dos attributes = yes
```

或者，您可以通过将相同的参数添加到共享部分来启用对单个共享的 Windows ACL 支持。

16.1.6.2.3. 添加使用 Windows ACL 的共享

要创建一个名为 example 的共享，该共享 /srv/samba/example/ 目录的内容并使用 Windows ACL：

添加使用 Windows ACL 的共享

1. （可选）创建 文件夹（如果文件夹不存在）。例如：

```
~]# mkdir -p /srv/samba/example/
```

2. 如果您以 enforcing 模式运行 SELinux，请在目录中设置 samba_share_t 上下文：

```
~]# semanage fcontext -a -t samba_share_t "/srv/samba/example(/.*)?"
~]# restorecon -Rv /srv/samba/example/
```

3. 将 示例 共享添加到 /etc/samba/smb.conf 文件中。例如，添加启用了共享的写操作：

```
[example]
 path = /srv/samba/example/
 read only = no
```

注意

无论文件系统 ACL 是什么；如果您没有 只设置只读 = no，Samba 都会以只读模式共享该目录。

4. 如果您还没有在所有共享的 [global] 部分中启用 Windows ACL 支持，请在 [example] 部分中添加以下参数来为这个共享启用此功能：

```
vfs objects = acl_xattr
map acl inherit = yes
store dos attributes = yes
```

5. 验证 /etc/samba/smb.conf 文件：

```
~]# testparm
```

详情请查看 第 16.1.2 节 “使用 testparm 实用程序验证 smb.conf 文件”。

6. 打开所需的端口并使用 firewall-cmd 工具重新载入防火墙配置：

```
~]# firewall-cmd --permanent --add-service=samba
~]# firewall-cmd --reload
```

7. 重启 smb 服务：

```
~]# systemctl restart smb
```

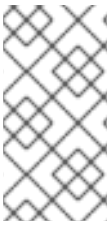
8. 另外，还可启用 smb 服务在引导时自动启动：

```
~]# systemctl enable smb
```

16.1.6.2.4. 管理使用 Windows ACL 共享的共享权限和文件系统 ACL

要在使用 Windows ACL 的 Samba 共享上管理共享和文件系统 ACL，请使用 Windows 应用程序，如 计算机管理。详情请查看您的 Windows 文档。

或者，也可使用 smbcacls 实用程序来管理 ACL。详情请查看 第 16.1.6.3 节 “使用 smbcacls管理 SMB 共享上的 ACL”。



注意

要从 Windows 修改文件系统权限，您必须使用授予 SeDiskOperatorPrivilege 权限的帐户。请参阅 第 16.1.6.2.1 节 “授予 SeDiskOperatorPrivilege 权限”。

16.1.6.3. 使用 smbcacls管理 SMB 共享上的 ACL

smbcacls 实用程序可以列出、设置和删除存储在 SMB 共享中的文件和目录的 ACL。您可以使用 smbcacls 来管理文件系统 ACL：

- 在本地或远程 Samba 服务器上使用高级 Windows ACL 或 POSIX ACL。
- 在 Red Hat Enterprise Linux 上，远程管理 Windows 上托管的共享上的 ACL。

16.1.6.3.1. 了解访问控制条目

文件系统对象的每个 ACL 条目均包含以下格式的访问控制条目(ACE)：

```
security_principal:access_right/inheritance_information/permissions
```

例 16.5. 访问控制条目

如果 AD\Domain Users 组修改了适用于 此 文件夹、子文件夹和 Windows 上的文件的权限，ACL 将包含以下 ACE：

```
AD\Domain Users:ALLOWED/OI|CI/CHANGE
```

下面描述了独立的 ACE：

安全主体

安全主体是 ACL 中权限的用户、组群或 SID。

访问权利

定义是否授予或拒绝对对象的访问权限。该值可以是 ALLOWED 或 DENIED。

继承信息

存在以下值：

表 16.3. 继承设置

值	描述	映射到
OI	对象实例	这个文件夹和文件
CI	容器继承	这个文件夹和子文件夹
IO	只继承	ACE 不适用于当前文件或目录。
ID	继承	ACE 从父目录继承。

另外，这些值可以合并如下：

表 16.4. 继承设置组合

值组合	映射到 Windows 应用到 设置
OI/CI	这个文件夹、子文件夹和文件
OI/CI/IO	只适用于子文件夹和文件

值组合	映射到 Windows 应用到 设置
CI/IO	只使用子文件夹
OI/IO	仅限文件

权限

这个值可以是代表一个或多个 Windows 权限的十六进制值，也可以是 smbcacls 别名：

- *代表一个或多个 Windows 权限的十六进制值。*

  下表以十六进制格式显示高级 Windows 权限及其对应的值：

  表 16.5. Windows Permissions 和 Their Correspo ing smbcacls 值（Hex 格式）

Windows 权限	十六进制值
完整控制	**0x001F01FF**
遍历文件夹 / 执行文件	**0x00100020**
列出文件夹 / 读数据	**0x00100001**
读取属性	**0x00100080**
读取扩展属性	**0x00100008**
创建文件 / 写数据	**0x00100002**
创建文件夹/附加数据	**0x00100004**
写入属性	**0x00100100**
写扩展属性	**0x00100010**
删除子文件夹和文件	**0x00100040**
删除	**0x00110000**
读取权限	**0x00120000**
更改权限	**0x00140000**

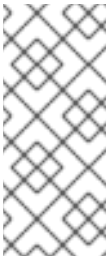
Windows 权限	十六进制值
获取所有权	**0x00180000**

可以使用 位范围 OR 操作将多个权限组合为一个十六进制值。详情请查看 第 16.1.6.3.3 节 “计算 ACE 掩码”。

- *smbcacls 别名。 下表显示了可用的别名：*

表 16.6. 现有 smbcacls Aliases 和 ir Correspoing Windows 权限

smbcacls Alias	映射到 Windows 权限
R	读
读取	读和执行
W	特殊 - 创建文件 / 写数据 - 创建文件夹/附加数据 - 写入属性 - 写扩展属性 - 读取权限
D	删除
P	更改权限
O	获取所有权
X	遍历 / 执行
更改	修改
FULL	完整控制



注意

设置权限时，您可以组合单例别名。例如，您可以设置 RD 以应用 Windows 权限 读取 和删除。但是，您既不能组合多个非字母别名，也无法组合别名和十六进制值。

16.1.6.3.2. 使用 smbcacls显示 ACL

如果您运行不带任何操作参数的 smbcacls，如 --add，则实用程序会显示文件系统对象的 ACL。

例如，列出 //server/example 共享的根目录的 ACL：

```
~]# smbcacls //server/example / -U "DOMAINpass:quotes[administrator]"
Enter DOMAINpass:quotes[administrator]'s password:
REVISION:1
CONTROL:SR|PD|DI|DP
OWNER:AD\Administrators
GROUP:AD\Domain Users
ACL:AD\Administrator:ALLOWED/OI|CI/FULL
ACL:AD\Domain Users:ALLOWED/OI|CI/CHANGE
ACL:AD\Domain Guests:ALLOWED/OI|CI/0x00100021
```

命令的输出会显示：

- REVISION ：安全描述符的内部 Windows NT ACL 版本
- CONTROL ：安全描述符控制
- OWNER ：安全描述符所有者的名称或 SID
- GROUP ：安全描述符组的名称或 SID
- ACL 条目.详情请查看 第 16.1.6.3.1 节 “了解访问控制条目”。

16.1.6.3.3. 计算 ACE 掩码

在大多数情况下，当添加或更新 ACE 时，您可以使用 表 16.6 “现有 smbcacls Aliases 和 ir Correspoing Windows 权限” 中列出的 smbcacls 别名。

但是，如果要设置 表 16.5 “Windows Permissions 和 Their Correspo ing smbcacls 值（Hex 格式）” 中列出的高级 Windows 权限，则必须使用位范围 OR 操作来计算正确的值。您可以使用以下 shell 命令计算值：

```
~]# echo $(printf '0x%X' $

    hex_value_1 | hex_value_2 | ...)
```

例 16.6. 计算 ACE 掩码

您需要设置以下权限：

- 遍历文件夹/执行文件 (0x00100020)
- 列出文件夹/读取数据 (0x00100001)
- 读取属性 (0x00100080)

要计算之前权限的十六进制值，请输入：

```
~]# echo $(printf '0x%X' $(( 0x00100020 | 0x00100001 | 0x00100080 )))
0x1000A1
```

设置或更新 ACE 时使用返回的值。

16.1.6.3.4. 使用 smbcacls添加、更新和删除 ACL

根据您传递给 smbcacls 实用程序的参数，您可以从文件或目录中添加、更新和删除 ACL。

添加 ACL

将 ACL 添加到 //server/example 共享的根目录，该共享将授予 此文件夹、子文件夹和文件的 CHANGE 权限到 AD\Domain Users 组：

```
~]# smbcacls //server/example / -U "DOMAIN\administrator \
   --add ACL:"AD\Domain Users":ALLOWED/OI|CI/CHANGE
```

更新 ACL

更新 ACL 与添加新的 ACL 类似。您可以使用 --modify 参数和现有的安全主体覆盖 ACL，以此更新 ACL。如果 smbcacls 在 ACL 列表中找到安全主体，则实用程序会更新这些权限。否则，命令会失败并显示错误：

```
ACL for SID principal_name not found
```

例如，要更新 AD\Domain Users 组的权限，并为此 文件夹、子文件夹和文件将其设置为 READ ：

```
~]# smbcacls //server/example / -U "DOMAIN\administrator \
   --modify ACL:"AD\Domain Users":ALLOWED/OI|CI/READ
```

删除 ACL

要删除 ACL，请将带有确切 ACL 的 --delete 传递到 smbcacls 实用程序。例如：

```
~]# smbcacls //server/example / -U "DOMAIN\administrator \
   --delete ACL:"AD\Domain Users":ALLOWED/OI|CI/READ
```

16.1.6.4. 启用用户在 Samba 服务器上共享目录

在 Samba 服务器上，您可以配置用户可以共享目录，而无需 root 权限。

16.1.6.4.1. 启用用户共享功能

在用户可以共享目录之前，管理员必须在 Samba 中启用用户共享。例如，只启用本地 示例 组的成员来创建用户共享：

启用用户共享

1. 如果本地 示例 组不存在，请创建它：

```
~]# groupadd example
```

2. 为 Samba 准备目录以存储用户共享定义并正确设置其权限。例如：

   a. 创建目录：

   ```
   ~]# mkdir -p /var/lib/samba/usershares/
   ```

   b. 为 示例 组设置写入权限：

   ```
   ~]# chgrp example /var/lib/samba/usershares/
   ~]# chmod 1770 /var/lib/samba/usershares/
   ```

   设置粘性位以防止用户重命名或删除此目录中其他用户存储的文件。

3. 编辑 /etc/samba/smb.conf 文件，并将以下内容添加到 [global] 部分：

   a. 设置您配置用来存储用户共享定义的目录的路径。例如：

   ```
   usershare path = /var/lib/samba/usershares/
   ```

   b. 设置允许在这个服务器上创建多少个用户共享 Samba。例如：

   ```
   usershare max shares = 100
   ```

   如果您对 usershare max shares 参数使用默认值 0，则用户共享将被禁用。

   c. 另外，还可设置绝对目录路径列表。例如，要配置 Samba 仅允许共享 /data 和 / srv 目录的子目录，请设置：

   ```
   usershare prefix allow list = /data /srv
   ```

   有关您可以设置的其他用户共享相关参数的列表，请参阅 smb.conf(5)man page 中的 USERSHARES 部分。

4. 验证 /etc/samba/smb.conf 文件：

```
~]# testparm
```

详情请查看 第 16.1.2 节 “使用 testparm 实用程序验证 smb.conf 文件”。

5. 重新载入 Samba 配置：

```
~]# smbcontrol all reload-config
```

用户现在可以创建用户共享。详情请查看 第 16.1.6.4.2 节 “添加用户共享”。

16.1.6.4.2. 添加用户共享

根据 第 16.1.6.4.1 节 “启用用户共享功能” 配置 Samba 后，用户可以通过运行 net usershare add 命令，在没有 root 权限的情况下共享 Samba 服务器上的目录。

net usershare add 命令的概要：net usershare add share_namepathcommentACLsguest_ok=y|n

重要

如果在创建用户共享时设置了 ACL，您必须在 ACL 之前指定注释参数。要设置空注释，请在双引号中使用空字符串。

请注意，如果管理员设置了 usershare，在 /etc/samba/smb.conf 文件的 [global] 部分中，用户只能启用用户共享的 guest 访问。

例 16.7. 添加用户共享

用户想要在 Samba 服务器上共享 /srv/samba/ 目录。该共享应命名为 example，未设置任何注释，并且 guest 用户应可以访问该共享。此外，共享权限应设置为 AD\Domain Users 组的完整访问权限，以及其他用户的读取权限。要添加此共享，请以用户身份运行：

```
~]$ net usershare add example /srv/samba/ "" \
   "AD\Domain Users":F,Everyone:R guest_ok=yes
```

16.1.6.4.3. 更新用户共享的设置

如果要更新用户共享的设置，请使用具有相同共享名称和新设置的 net usershare add 命令覆盖共享。请参阅 第 16.1.6.4.2 节 “添加用户共享”。

16.1.6.4.4. 显示关于现有用户共享的信息

用户可以在 Samba 服务器上输入 net usershare info 命令，以显示用户共享及其设置。

显示任意用户创建的所有用户共享：

```
~]$ net usershare info -l
[share_1]
path=/srv/samba/
comment=
usershare_acl=Everyone:R,host_name\user:F,
guest_ok=y
...
```

若要仅列出运行 命令的用户创建的共享，请省略 -l 参数。

要仅显示关于特定共享的信息，请将共享名称或通配符传递到 命令。例如，显示名称以 share_ 开头的共享的信息：

```
~]$ net usershare info -l share*_
```

16.1.6.4.5. 列出用户共享

如果您只列出可用的用户共享，且在 Samba 服务器上没有设置，请使用 net usershare list 命令。

列出任意用户创建的共享：

```
~]$ net usershare list -l
share_1
share_2
...
```

若要仅列出运行 命令的用户创建的共享，请省略 -l 参数。

要仅列出特定的共享，请将共享名称或通配符传递到 命令。例如，仅列出名称以 share_ 开头的共享：

```
~]$ net usershare list -l share_*
```

16.1.6.4.6. 删除用户共享

要删除用户共享，请以创建共享的用户身份输入，或者以 root 用户身份 输入：

```
~]$ net usershare delete share_name
```

16.1.6.5. 启用客户机访问共享

在某些情况下，您想要共享一个用户无需身份验证即可连接到的目录。若要配置此配置，请在共享中启用来宾访问权限。

警告

不需要身份验证的共享可能会造成安全隐患。

如果在共享中启用了 guest 访问权限，Samba 将映射客户机与 guest 帐户 参数中设置的操作系统帐户的连接。如果至少满足以下条件之一，客户机用户可以访问这些文件：

- 该帐户在文件系统 ACL 中列出
- 其他 用户的 POSIX 权限允许它

例 16.8. 客户机共享权限

如果您将 Samba 配置为将客户机帐户映射到 nobody （默认值），以下示例中的 ACL：

- 允许 guest 用户读取 file1.txt
- 允许 guest 用户读取和修改 file2.txt.
- 阻止 guest 用户读取或修改 file3.txt

```
-rw-r--r--. 1 root    root   1024 1. Sep 10:00 file1.txt
-rw-r-----. 1 nobody   root   1024 1. Sep 10:00 file2.txt
-rw-r-----. 1 root    root   1024 1. Sep 10:00 file3.txt
```

例如，要启用现有 [example] 共享的客户机访问：

设置客户机共享

1. 编辑 /etc/samba/smb.conf 文件：

   a. 如果这是您在这个服务器上设置的第一个客户机共享：

      i. 在 [global] 部分将 map 设置为 guest = Bad User ：

      ```
      [global]
       ...
       map to guest = Bad User
      ```

      使用这个设置时，Samba 将拒绝使用错误密码的登录尝试，除非用户名不存在。如果指定的用户名不存在，并且共享中启用了 guest 访问，Samba 会将连接视为客户机登录。

      ii. 默认情况下，Samba 将 guest 帐户映射到 Red Hat Enterprise Linux 中的 nobody 帐户。另外，您还可以设置不同的帐户。例如：

```
[global]
 ...
 guest account = user_name
```

此参数中设置的帐户必须在 Samba 服务器中本地存在。出于安全考虑，红帽建议您使用没有分配有效 shell 的帐户。

b. 在 [example] 部分添加 客户机 ok = yes 设置：

```
[example]
 ...
 guest ok = yes
```

2. 验证 /etc/samba/smb.conf 文件：

```
~]# testparm
```

详情请查看 第 16.1.2 节 “使用 testparm 实用程序验证 smb.conf 文件”。

3. 重新载入 Samba 配置：

```
~]# smbcontrol all reload-config
```

16.1.7. 设置 Samba 打印服务器

如果您将 Samba 设置为打印服务器，则网络中的客户端可以使用 Samba 进行打印。此外，如果配置了 Windows 客户端，可以从 Samba 服务器下载该驱动程序。

在共享打印机前，请设置 Samba:

- 第 16.1.4 节 “将 Samba 设置为单机服务器”
- 第 16.1.5 节 “将 Samba 设置为域成员”

16.1.7.1. Samba spoolsd 服务

Samba 假脱机是一种集成到 smbd 服务中的服务。在 Samba 配置中启用假脱机，以显著提高具有大量作业或打印机的打印服务器的性能。

如果没有 spoolsd，Samba 就会对 smbd 进程执行分叉，并为每个打印作业初始化 printcap 缓存。如果有大量打印机，smb d 服务可能会在初始化缓存时多秒钟内变得无响应。spoolssd 服务可让您启动处理打印作业的预派 smbd 进程。主 spoolsd smbd 进程使用较少的内存，分叉和终止子进程。

启用 spoolsd 服务：

启用 spoolsd 服务

1. 编辑 /etc/samba/smb.conf 文件中的 [global] 部分：

   a. 添加以下参数：

   ```
   rpc_server:spoolss = external
   rpc_daemon:spoolssd = fork
   ```

   b. 另外，您可以设置以下参数：

参数	Default（默认）	描述
spoolssd:prefork_min_children	5	最小子进程数量
spoolssd:prefork_max_children	25	子进程的最大数量
spoolssd:prefork_spawn_rate	5	Samba 在此参数中设置的新子进程数量中分叉，最多为 **spoolsd:prefork_max_children** 中设置的值（如果建立新连接）
spoolssd:prefork_max_allowed_clients	100	客户端数，子进程服务
spoolssd:prefork_child_min_life	60	子进程的最低生命周期（以秒为单位）。60 秒是最小的。

2. 验证 /etc/samba/smb.conf 文件：

```
~]# testparm
```

详情请查看 第 16.1.2 节 “使用 testparm 实用程序验证 smb.conf 文件”。

3. 重启 smb 服务：

```
~]# systemctl restart smb
```

重启该服务后，Samba 会自动启动 smbd 子进程：

```
~]# ps axf
...
30903 smbd
30912 \_ smbd
30913   \_ smbd
30914   \_ smbd
30915   \_ smbd
...
```

16.1.7.2. 在 Samba 中启用打印服务器支持

启用打印服务器支持：

在 Samba 中启用打印服务器支持

1. 在 Samba 服务器上，设置 CUPS 并将打印机添加到 CUPS 后端。详情请查看 第 16.3 节 “打印设置”。

注意

只有 Samba 打印服务器上本地安装了 CUPS 时，Samba 才能将打印作业转发到 CUPS。

2. 编辑 /etc/samba/smb.conf 文件：

a.

如果要启用 spoolsd 服务，请在 [global] 部分添加以下参数：

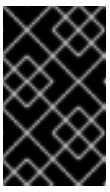
```
rpc_server:spoolss = external
rpc_daemon:spoolssd = fork
```

详情请查看 第 16.1.7.1 节 “Samba spoolsd 服务”。

b.

要配置打印后端，请添加 [printers] 部分：

```
[printers]
 comment = All Printers
 path = /var/tmp/
 printable = yes
 create mask = 0600
```



重要

打印机 共享名称是硬编码的，无法更改。

3.

验证 /etc/samba/smb.conf 文件：

```
~]# testparm
```

详情请查看 第 16.1.2 节 “使用 testparm 实用程序验证 smb.conf 文件”。

4.

打开所需的端口并使用 firewall-cmd 工具重新载入防火墙配置：

```
~]# firewall-cmd --permanent --add-service=samba
~]# firewall-cmd --reload
```

5.

重启 smb 服务：

```
~]# systemctl restart smb
```

重新启动 服务后，Samba 会自动共享在 CUPS 后端中配置的所有打印机。如果您只想手动共享特定打印机，请参阅 第 16.1.7.3 节 “手动共享特定打印机”。

16.1.7.3. 手动共享特定打印机

如果您将 Samba 配置为打印服务器，默认情况下，Samba 共享 CUPS 后端配置的所有打印机。仅共享特定的打印机：

手动共享特定打印机

1. 编辑 /etc/samba/smb.conf 文件：

   a. 在 [global] 部分中，通过设置禁用自动打印机共享：

   ```
   load printers = no
   ```

   b. 为您要共享的每个打印机添加 部分。例如，要在 Samba 中将 CUPS 后端中名为 example 的打印机共享为 Example-Printer，请添加以下部分：

   ```
   [Example-Printer]
    path = /var/tmp/
    printable = yes
    printer name = example
   ```

   您不需要为每个打印机单独设置 spool 目录。您可以在打印机 的路径 参数中设置与您在 [printers] 部分中设置相同的 spool 目录。

2. 验证 /etc/samba/smb.conf 文件：

   ```
   ~]# testparm
   ```

   详情请查看 第 16.1.2 节 “使用 testparm 实用程序验证 smb.conf 文件”。

3. 重新载入 Samba 配置：

   ```
   ~]# smbcontrol all reload-config
   ```

16.1.7.4. 为 Windows 客户端设置自动打印机驱动程序下载

如果您正在运行 Windows 客户端的 Samba 打印服务器，可以上传驱动程序和预配置打印机。如果

用户连接到打印机,Windows 会自动在客户端本地下载并安装驱动程序。用户不需要本地管理员权限进行安装。另外，Windows 应用预配置的驱动程序设置，如托盘的数量。

注意

在设置自动打印机驱动程序下载之前，必须将 Samba 配置为打印服务器并共享打印机。详情请查看 第 16.1.7 节 “设置 Samba 打印服务器”。

16.1.7.4.1. 有关打印机驱动程序的基本信息

本节提供有关打印机驱动程序的一般信息。

支持的驱动程序模型版本

Samba 仅支持 Windows 2000 及更高版本中支持的打印机驱动程序模型版本 3，以及 Windows Server 2000 及更高版本。Samba 不支持 Windows 8 和 Windows Server 2012 中引入的驱动程序模型版本 4。但是，这些及之后的 Windows 版本也支持版本 3 驱动程序。

可软件包的驱动程序

Samba 不支持可打包的驱动程序。

为进行上传准备打印机驱动程序

在您将驱动程序上传到 Samba 打印服务器之前：

- 如果驱动程序采用压缩格式提供，请解包它。
- 有些驱动程序需要启动一个设置应用程序，以便在 Windows 主机上在本地安装驱动程序。在某些情况下，安装程序会在设置运行期间将单个文件提取到操作系统的临时文件夹中。使用驱动程序文件上传：

  a. 启动安装程序。

  b. 将临时文件夹中的文件复制到新位置。

c.
取消安装。

请您的打印机厂商提供支持上传到打印服务器的驱动程序。

为客户端为打印机提供 32 位和 64 位驱动程序

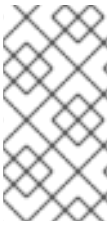
要为 32 位和 64 位 Windows 客户端提供打印机的驱动程序，您必须上传两个架构具有完全相同名称的驱动程序。例如，如果您上传名为 Example PostScript 的 32 位驱动程序和名为 Example PostScript (v1.0) 的 64 位驱动程序，则名称不匹配。因此，您只能为打印机分配其中一个驱动程序，且该驱动程序无法对这两个架构都适用。

16.1.7.4.2. 启用用户上传和预配置驱动程序

要上传和预配置打印机驱动程序，用户或组需要授予 SePrintOperatorPrivilege 权限。用户必须添加到 printadmin 组中。安装 samba 软件包时，Red Hat Enterprise Linux 会自动创建这个组。printadmin 组被分配了低于 1000 的可用最小动态系统 GID。

为 printadmin 组授予 SePrintOperatorPrivilege 权限：

```
~]# net rpc rights grant "printadmin" SePrintOperatorPrivilege \
   -U "DOMAIN\administrator"
Enter DOMAIN\administrator's password:
Successfully granted rights.
```



注意

在域环境中，将 SePrintOperatorPrivilege 授予给域组。这可让您通过更新用户的组成员资格来集中管理权限。

列出所有授予 SePrintOperatorPrivilege 的 用户和组：

```
~]# net rpc rights list privileges SePrintOperatorPrivilege \
   -U "DOMAIN\administrator"
Enter administrator's password:
SePrintOperatorPrivilege:
 BUILTIN\Administrators
 DOMAIN\printadmin
```

16.1.7.4.3. 设置 打印$ 共享

Windows 操作系统从名为 print$ 的共享中下载打印机驱动程序。这个共享名称在 Windows 中硬编码，无法更改。

以 print$ 用户身份共享 /var/lib/samba/drivers/ 目录，并启用本地 printadmin 组的成员上传打印机驱动程序：

设置 打印$ 共享

1. 在 /etc/samba/smb.conf 文件中添加 [print$] 部分：

```
[print$]
 path = /var/lib/samba/drivers/
 read only = no
 write list = @printadmin
 force group = @printadmin
 create mask = 0664
 directory mask = 2775
```

使用这些设置：

- 只有 printadmin 组的成员才能将打印机驱动程序上传到共享。
- 新创建文件和目录的组将设为 printadmin。
- 新文件的权限将设置为 664。
- 新目录的权限将设置为 2775。

2. 要只为打印机上传 64 位驱动程序，请在 /etc/samba/smb.conf 文件的 [global] 部分包含此设置：

```
spoolss: architecture = Windows x64
```

如果没有这个设置，Windows 只会显示您至少上传 32 位版本的驱动程序。

3. 验证 /etc/samba/smb.conf 文件：

```
~]# testparm
```

详情请查看 第 16.1.2 节 “使用 testparm 实用程序验证 smb.conf 文件”。

4. 重新载入 Samba 配置

```
~]# smbcontrol all reload-config
```

5. 如果 printadmin 组不存在，则创建它：

```
~]# groupadd printadmin
```

6. 为 printadmin 组授予 SePrintOperatorPrivilege 特权。

```
~]# net rpc rights grant "printadmin" SePrintOperatorPrivilege \
   -U "DOMAIN\administrator"
Enter DOMAIN\administrator's password:
Successfully granted rights.
```

详情请查看 第 16.1.7.4.2 节 “启用用户上传和预配置驱动程序”。

7. 如果您以 enforcing 模式运行 SELinux，请在目录中设置 samba_share_t 上下文：

```
~]# semanage fcontext -a -t samba_share_t "/var/lib/samba/drivers(/.*)?"
~]# restorecon -Rv /var/lib/samba/drivers/
```

8. 在 /var/lib/samba/drivers/ 目录中设置权限：

- 如果使用 POSIX ACL，请设置：

```
~]# chgrp -R "printadmin" /var/lib/samba/drivers/
~]# chmod -R 2775 /var/lib/samba/drivers/
```

- 如果使用 Windows ACL，请设置：

主体	权限	适用于
创建者所有者	完整控制	只适用于子文件夹和文件
经过身份验证的用户	读和执行、列出目录内容、读	此文件夹、子文件夹和文件
printadmin	完整控制	此文件夹、子文件夹和文件

有关在 Windows 上设置 ACL 的详情，请查看您的 Windows 文档。

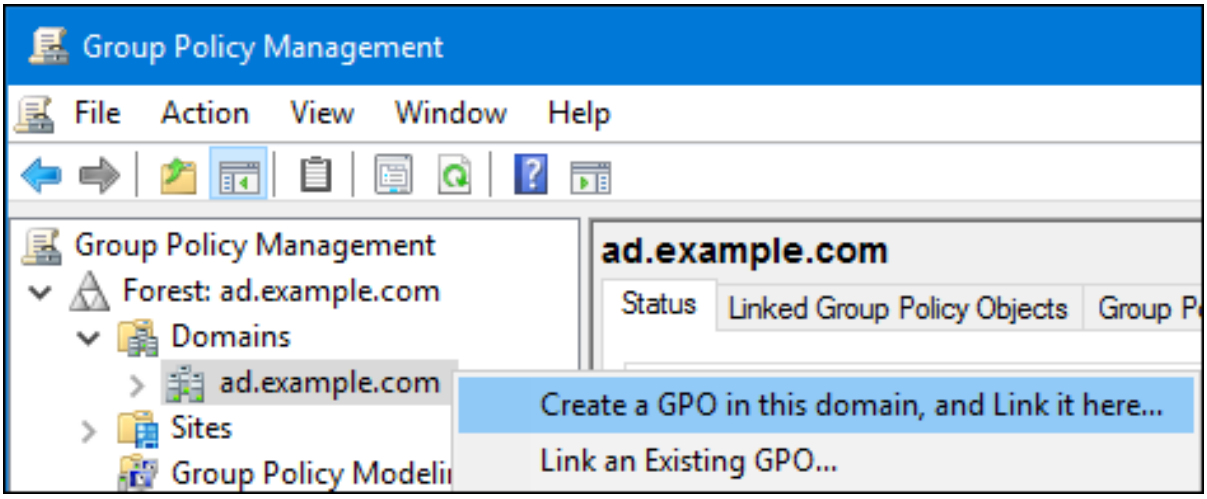
16.1.7.4.4. 创建 GPO 以启用客户端信任 Samba 打印服务器

出于安全考虑，最近的 Windows 操作系统会阻止客户端从不受信任的服务器下载非软件包的打印机驱动程序。如果您的打印服务器是 AD 中的成员，您可以在域中创建一个组策略对象(GPO)来信任 Samba 服务器。

要创建 GPO，您使用的 Windows 计算机必须安装有 Windows Remote Server 管理工具(RSAT)。详情请查看您的 Windows 文档。

创建 GPO 以启用客户端信任 Samba 打印服务器

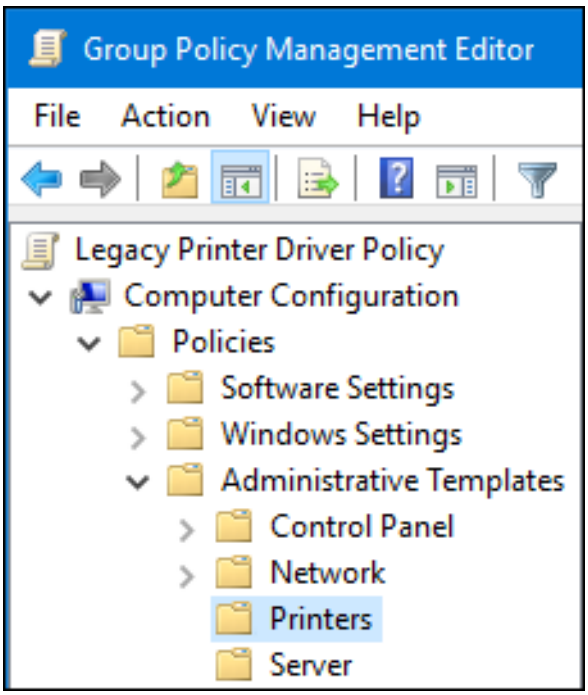
1. 使用允许编辑组策略的帐户（如 AD 域 管理员用户）登录到 Windows 计算机。
2. 打开 组策略管理控制台。
3. 右键单击您的 AD 域并选择 创建此域中的 GPO，然后链接此处



4. *输入 **GPO 的名称，如 Legacy 打印机**驱动**程序策略 并**单击**" 确定"。新的 GPO 将在域条目下**显示。*

5. ***右**键单击**新**创**建的 GPO，然后**选择 **Edit 以打开 Group Policy Management Editor。***

6. *进入 **Computer Configuration → Policies → Administrative Templates → Printers。***



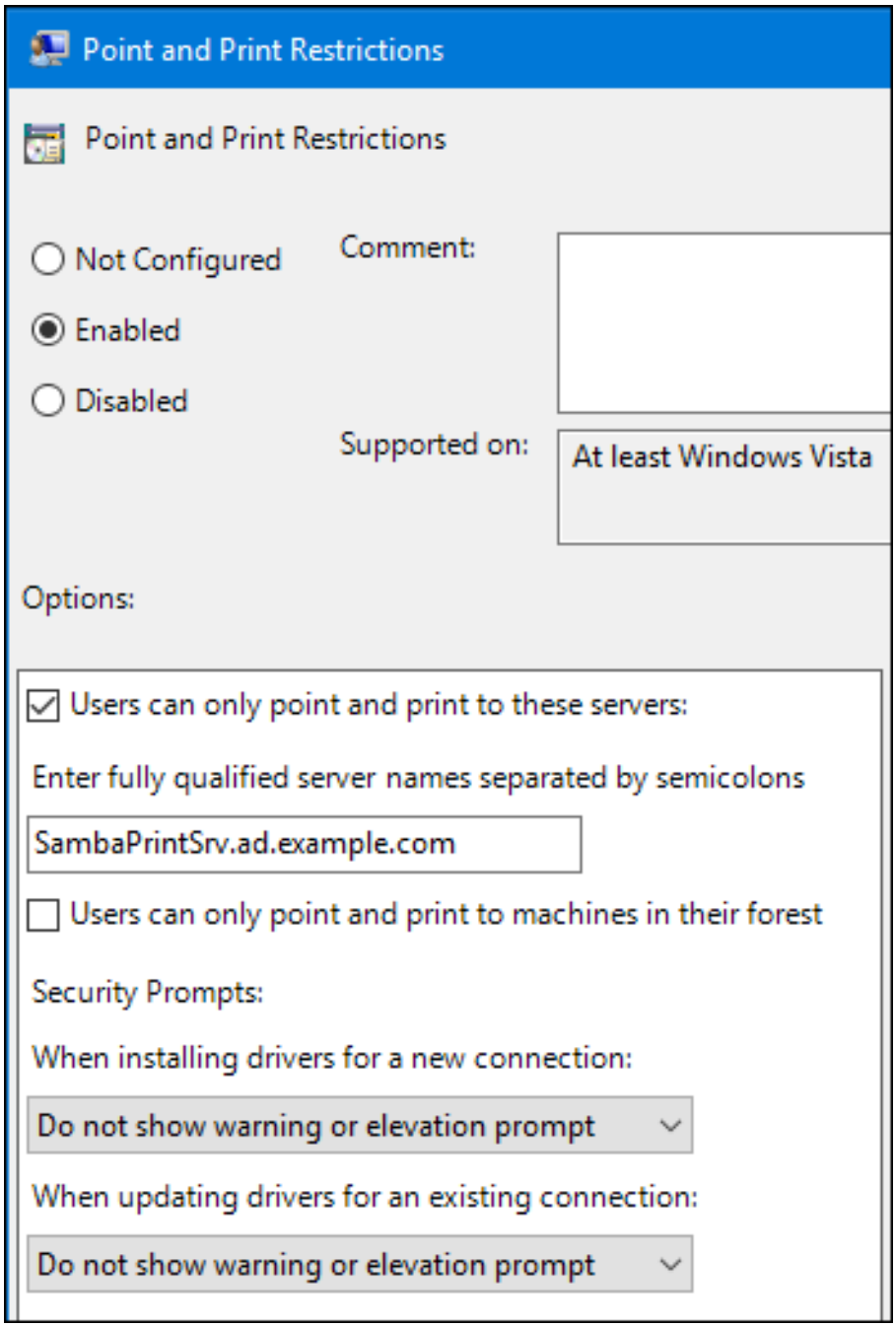
7. ***在窗口的右**侧，**双**击 **Point 和 Print Restriction 以**编辑**策略：***

   a. *启**用策略并**设**置以下**选项：*

      i. *选择 **Users 只能指向**这**些服**务**器并打印到**这**些服**务**器，再将 Samba 打印服**务**器*

的完全限定域名(FQDN)添加到此选项旁边的字段。

ii. *在 Security Prompts 下的两个复选框中，选择 Do not show warning 或 elevation 提示。*

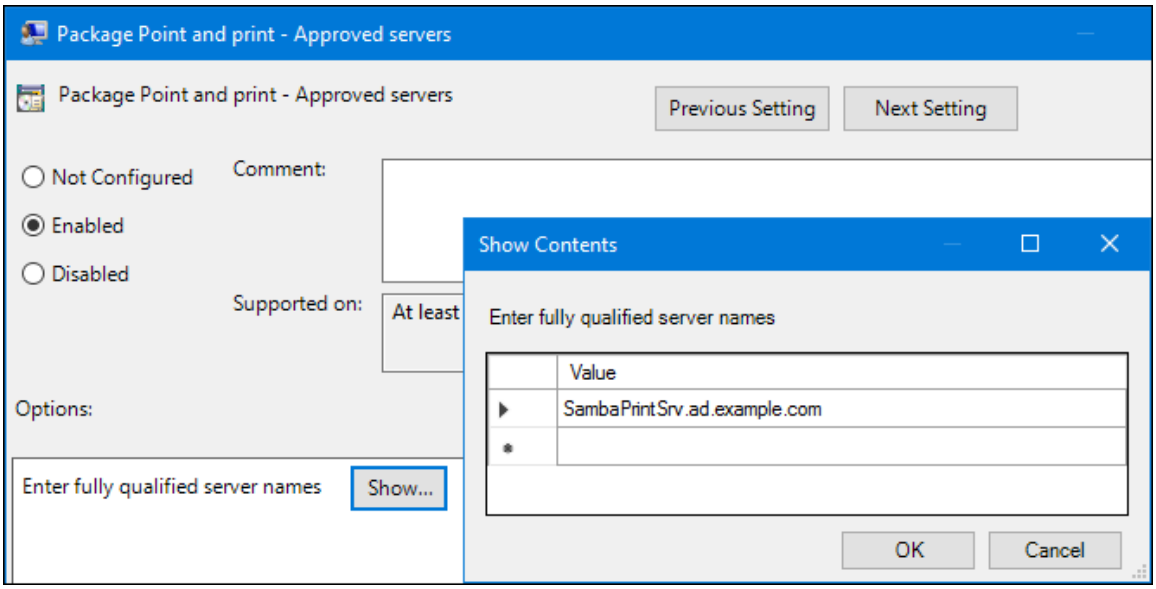


b. *点确定。*

8. *双击 Package Point 和 Print - Approved servers 以编辑策略：*

a. *启用策略并单击 Show 按钮。*

b. 输入 Samba 打印服务器的 FQDN。



c. 单击 OK，以关闭 Show Contents 和 policy 属性窗口。

9. 关闭 组策略管理编辑器。

10. 关闭 组策略管理控制台。

在 Windows 域成员应用组策略后，用户连接到打印机时会自动从 Samba 服务器下载打印机驱动程序。

有关使用组策略的详情，请查看您的 Windows 文档。

16.1.7.4.5. 上传驱动程序和预配置打印机

在 Windows 客户端上使用 Print Management 应用上传托管在 Samba 打印服务器上的驱动程序和预配置打印机。详情请查看您的 Windows 文档。

16.1.8. 调优 Samba 服务器的性能

本节介绍在某些情况下，哪些设置可以提高 Samba 的性能，以及哪些设置可能会对性能造成负面影响。

16.1.8.1. 设置 SMB 协议版本

每个新的 SMB 版本都会添加功能并提高协议的性能。最新的 Windows 和 Windows 服务器操作系统始终支持最新的协议版本。如果 Samba 还使用最新的协议版本，则连接 Samba 的 Windows 客户端可从性能改进中受益。在 Samba 中，服务器 max 协议 的默认值被设置为最新支持的 stable SMB 协议版本。

要始终启用最新的稳定 SMB 协议版本，请不要设置 server max protocol 参数。如果手动设置 参数，则需要修改 SMB 协议的每个新版本的设置，以启用最新的协议版本。

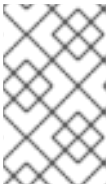
要取消设置，请从 /etc/samba/smb.conf 文件的 [global] 部分中删除 server max protocol 参数。

16.1.8.2. 使用包含大量文件的目录调优共享

提高包含超过 100.000 文件的目录的共享性能：

使用包含大量文件的目录调优共享

1. 将共享上的所有文件重命名为小写。



注意

使用这个过程中的设置，名称不为小写的文件将不再显示。

2. 在共享部分中设置以下参数：

```
case sensitive = true
default case = lower
preserve case = no
short preserve case = no
```

有关参数的详情，请查看 smb.conf(5)man page 中的描述。

3. 重新载入 Samba 配置：

```
~]# smbcontrol all reload-config
```

应用这些设置后，此共享上所有新建文件的名称都使用小写。由于这些设置，Samba 不再需要扫描大写和小写的 目录，这样可以提高性能。

16.1.8.3. 可具有潜在性能影响的设置

默认情况下，调整 Red Hat Enterprise Linux 中的内核以获得高网络性能。例如，内核对缓冲区大小使用自动轮询机制。在 /etc/samba/smb.conf 文件中设置 socket options 参数会覆盖这些内核设置。因此，设置此参数会在大多数情况下降低 Samba 网络性能。

要使用内核优化的设置，请从 /etc/samba/smb.conf 中的 [global] 部分删除 socket options 参数。

16.1.9. 常用 Samba 命令行实用程序

这部分论述了在使用 Samba 服务器时常用的命令。

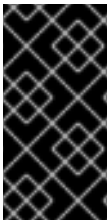
16.1.9.1. 使用 网络 实用程序

net 实用程序允许您在 Samba 服务器中执行多个管理任务。本节介绍 net 实用程序最常用的子命令。

详情请查看 net(8)man page。

16.1.9.1.1. 使用 net ads join 和 net rpc join 命令

使用 net 实用程序的 join 子命令，您可以将 Samba 加入到 AD 或 NT4 域。要加入该域，您必须手动创建 /etc/samba/smb.conf 文件，并选择性地更新其他配置，如 PAM。



重要

红帽建议使用 realm 实用程序加入域。realm 实用程序自动更新所有相关配置文件。详情请查看 第 16.1.5.1 节 “加入域”。

使用 net 命令加入域：

使用 net 命令加入域

1. 使用以下设置手动创建 /etc/samba/smb.conf 文件：

   - 对于 AD 域成员：

     ```
     [global]
     workgroup = domain_name
     security = ads
     passdb backend = tdbsam
     realm = AD_REALM
     ```

   - 对于 NT4 域成员：

     ```
     [global]
     workgroup = domain_name
     security = user
     passdb backend = tdbsam
     ```

2. 为 * 默认域和要加入到 /etc/samba/smb.conf 中的 [global] 部分的域添加 ID 映射配置。详情请查看 第 16.1.5.3 节 “了解 ID 映射”。

3. 验证 /etc/samba/smb.conf 文件：

   ```
   ~]# testparm
   ```

   详情请查看 第 16.1.2 节 “使用 testparm 实用程序验证 smb.conf 文件”。

4. 以域管理员身份加入域：

   - 加入 AD 域：

     ```
     ~]# net ads join -U "DOMAINpass:quotes[administrator]"
     ```

   - 要加入 NT4 域：

     ```
     ~]# net rpc join -U "DOMAINpass:quotes[administrator]"
     ```

5.

5. 将 winbind 源附加到 /etc/nsswitch.conf 文件中的 passwd 和 组 数据库条目中：

```
passwd:   files winbind
group:   files winbind
```

6. 启用并启动 winbind 服务：

```
~]# systemctl enable winbind
~]# systemctl start winbind
```

7. （可选）使用 authconf 实用程序配置 PAM。

详情请查看 红帽系统级 身份验证指南中的使用可插拔验证 模块(PAM) 部分。

8. 另外，对于 AD 环境，配置 Kerberos 客户端。

详情请查看 Red Hat System-Level Authentication Guide 中的 配置 Kerberos 客户端 部分。

16.1.9.1.2. 使用 net rpc 权限 命令

在 Windows 中，您可以为帐户和组分配执行特殊操作的特权，如设置共享上的 ACL 或上传打印机驱动程序。在 Samba 服务器上，您可以使用 net rpc permissions 命令来 管理特权。

列出特权

若要列出所有可用的特权及其所有者，可使用 net rpc permissions list 命令。例如：

```
net rpc rights list -U "DOMAINpass:attributes[{blank}]administrator"
Enter DOMAINpass:attributes[{blank}]administrator's password:
  SeMachineAccountPrivilege Add machines to domain
  SeTakeOwnershipPrivilege Take ownership of files or other objects
     SeBackupPrivilege Back up files and directories
    SeRestorePrivilege Restore files and directories
  SeRemoteShutdownPrivilege Force shutdown from a remote system
  SePrintOperatorPrivilege Manage printers
    SeAddUsersPrivilege Add users and groups to the domain
 SeDiskOperatorPrivilege Manage disk shares
    SeSecurityPrivilege System security
```

授予特权

若要为帐户或组授予特权，可使用 net rpc 权限 grant 命令。

例如，为 DOMAIN\printadmin 组授予 SePrintOperatorPrivilege 权限：

```
~]# net rpc rights grant "DOMAIN\printadmin" SePrintOperatorPrivilege \
   -U "DOMAIN\administrator"
Enter DOMAIN\administrator's password:
Successfully granted rights.
```

撤销特权

若要从帐户或组撤销特权，可使用 net rpc 权限撤销。

例如，要从 DOMAIN\printadmin 组撤销 SePrintOperatorPrivilege 权限：

```
~]# net rpc rights remoke "DOMAIN\printadmin" SePrintOperatorPrivilege \
   -U "DOMAIN\administrator"
Enter DOMAIN\administrator's password:
Successfully revoked rights.
```

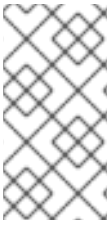
16.1.9.1.3. 使用 net rpc 共享 命令

net rpc share 命令提供在本地或远程 Samba 或 Windows 服务器上列出、添加和删除共享的功能。

列出共享

若要列出 SMB 服务器上的共享，请使用 net rpc share list 命令。（可选）将 -S server_name 参数传递到 命令，以列出远程服务器的共享。例如：

```
~]# net rpc share list -U "DOMAIN\administrator" -S example
Enter DOMAIN\administrator's password:
IPC$
share_1
share_2
...
```



注意

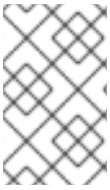
在 Samba 服务器上托管的共享（在 /etc/samba/smb.conf 文件中的 部分已设置 = no set）不会显示在 /etc/samba/smb.conf 文件中。

添加共享

net rpc 共享 add 命令允许您向 SMB 服务器添加共享。

例如，要在共享 C:\example\ 目录的远程 Windows 服务器中添加一个名为 example 的共享：

```
~]# net rpc share add example="C:\example" -U "DOMAIN\administrator" -S server
```



注意

在指定 Windows 目录名称时，您必须省略路径中的结尾反斜杠。

使用命令在 Samba 服务器中添加共享：

- 在 -U 参数中指定的用户必须授予 SeDiskOperatorPrivilege 权限。
- 您必须编写一个在 /etc/samba/smb.conf 文件中添加共享部分并重新加载 Samba 的脚本。该脚本必须在 /etc/samba/smb.conf 的 [global] 部分中的 add share 命令 参数中设置。详情请查看 smb.conf(5)man page 中的 add share 命令 描述。

删除共享

net rpc share delete 命令允许您从 SMB 服务器中删除共享。

例如，要从远程 Windows 服务器中删除名为 example 的共享：

```
~]# net rpc share delete example -U "DOMAIN\administrator" -S server
```

使用命令从 Samba 服务器中删除共享：

- 在 -U 参数中指定的用户必须授予 SeDiskOperatorPrivilege 权限。

- 您必须编写一个从 /etc/samba/smb.conf 文件中删除共享的 部分并重新加载 Samba 的脚本。该脚本必须在 /etc/samba/smb.conf 的 [global] 部分中的 delete share 命令 参数中设置。详情请查看 smb.conf(5)man page 中的 删除共享命令 描述。

16.1.9.1.4. 使用 net 用户 命令

net user 命令可让您在 AD DC 或 NT4 PDC 中执行以下操作：

- 列出所有用户帐户

- 添加用户

- 删除用户

注意

只有在列出域用户帐户时，才需要指定连接方法，如 AD 域或 rpc 的 NT4 域的 ad。其他用户相关的子命令可以自动探测连接方法。

将 -U user_name 参数传递给 命令，以指定允许执行请求操作的用户。

列出域用户帐户

列出 AD 域中的所有用户：

```
~]# net ads user -U "DOMAIN\administrator"
```

列出 NT4 域中的所有用户：

```
~]# net rpc user -U "DOMAIN\administrator"
```

将用户帐户添加到域

在 Samba 域成员中，您可以使用 net user add 命令将用户帐户添加到该域。

例如，将 用户帐户 添加到域中：

将用户帐户添加到域

1. 添加帐户：

```
~]# net user add user password -U "DOMAIN\administrator"
User user added
```

2. （可选）使用远程过程调用(RPC)shell 在 AD DC 或 NT4 PDC 中启用帐户。例如：

```
~]# net rpc shell -U DOMAIN\administrator -S DC_or_PDC_name
Talking to domain DOMAIN (S-1-5-21-1424831554-512457234-5642315751)

net rpc> user edit disabled user no
Set user's disabled flag from [yes] to [no]

net rpc> exit
```

从域中删除用户帐户

在 Samba 域成员中，您可以使用 net user delete 命令从域中删除用户帐户。

例如，要从域中删除 用户帐户 ：

```
~]# net user delete user -U "DOMAIN\administrator"
User user deleted
```

16.1.9.1.5. 使用 net usershare 命令

请参阅 第 16.1.6.4 节 “启用用户在 Samba 服务器上共享目录”。

16.1.9.2. 使用 rpcclient 实用程序

The rpcclient 实用程序可让您在本地或远程 SMB 服务器上手动执行客户端 Microsoft 远程过程调用 (MS-RPC)功能。但是，大部分功能都已集成到 Samba 提供的单独实用程序中。Use rpcclient 仅用于测

试 MS-PRC 功能。

例如，您可以使用该工具：

- 管理打印机池子系统(SPOOLSS)。

  例 16.9. 将驱动程序分配给打印机

  ```
  ~]# rpcclient server_name -U "DOMAINpass:quotes[administrator]" \
     -c 'setdriver "printer_name" "driver_name"'
  Enter DOMAINpass:quotes[administrator]s password:
  Successfully set printer_name to driver driver_name.
  ```

- 检索有关 SMB 服务器的信息。

  例 16.10. 列出所有文件共享和共享的打印机

  ```
  ~]# rpcclient server_name -U "DOMAINpass:quotes[administrator]" -c
  'netshareenum'
  Enter DOMAINpass:quotes[administrator]s password:
  netname: Example_Share
   remark:
   path:  C:\srv\samba\example_share\
   password:
  netname: Example_Printer
   remark:
   path:  C:\var\spool\samba\
   password:
  ```

- 使用安全帐户管理器远程(SAMR)协议执行操作。

  例 16.11. 在 SMB 服务器中列出用户

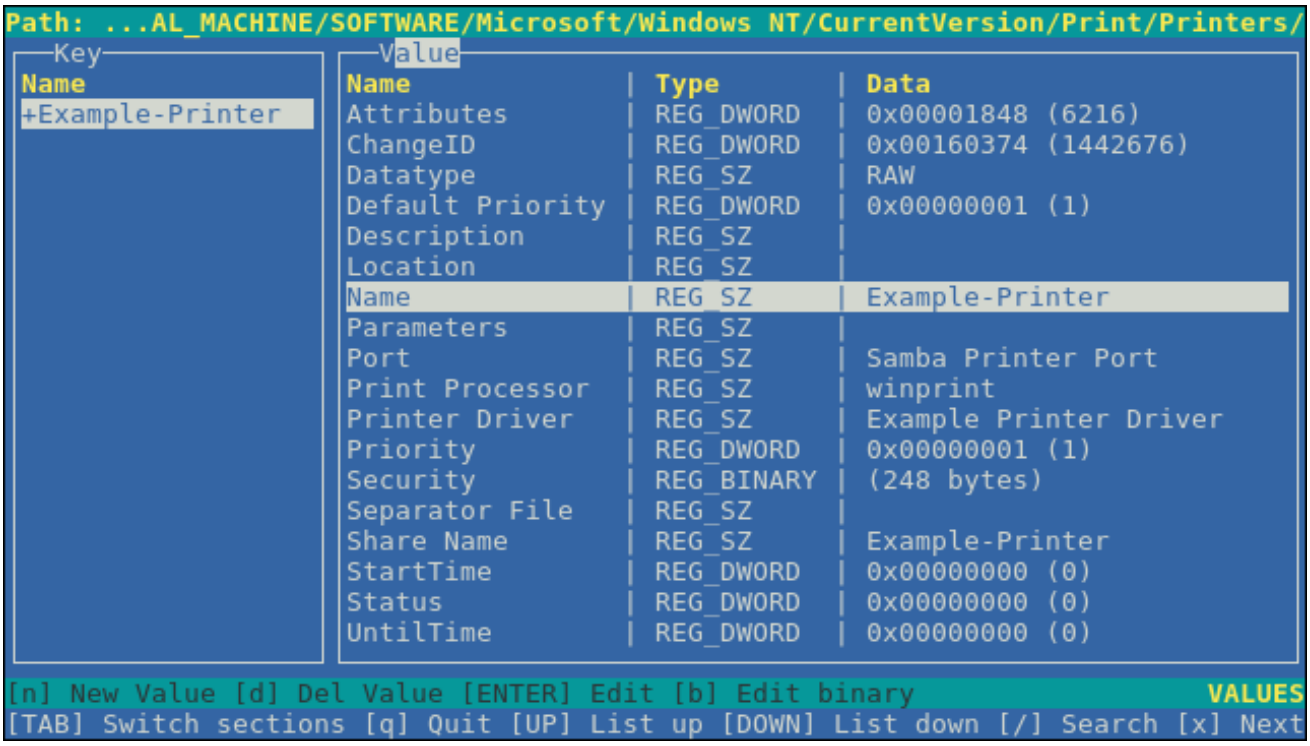
  ```
  ~]# rpcclient server_name -U "DOMAINpass:quotes[administrator]" -c
  'enumdomusers'
  Enter DOMAINpass:quotes[administrator]s password:
  user:[user1] rid:[0x3e8]
  user:[user2] rid:[0x3e9]
  ```

如果您针对单机服务器或域成员运行 命令，它会将用户列在本地数据库中。针对 AD DC 或 NT4 PDC 运行命令列出域用户。

有关支持子命令的完整列表，请参见 rpcclient(1)man page 中的 COMMANDS 部分。

16.1.9.3. 使用 samba-regedit 应用

某些设置（如打印机配置）存储在 Samba 服务器上的注册表中。您可以使用基于 ncurses 的 samba-regedit 应用来编辑 Samba 服务器的注册表。



要启动应用程序，请输入：

```
~]# samba-regedit
```

使用以下键：

- 上键和下键：在注册表树和值中进行导航。
- Enter：打开关键字或编辑值。
- 选项卡 ：在 Key 和 Value 窗格间切换。

- Ctrl+C：关闭应用程序。

16.1.9.4. 使用 smbcacls 实用程序

请参阅 第 16.1.6.3 节 “使用 smbcacls管理 SMB 共享上的 ACL”。

16.1.9.5. 使用 smbclient 实用程序

smbclient 实用程序允许您访问 SMB 服务器上的文件共享，这与命令行 FTP 客户端类似。例如，您可以使用它向共享上传和下载文件。

例如，使用 DOMAIN\user 帐户对 服务器上 托管的示例 共享 进行身份验证：

```
~]# smbclient -U "DOMAIN\user" //server/example
Enter domain\user's password:
Domain=[SERVER] OS=[Windows 6.1] Server=[Samba 4.6.2]
smb: \>
```

在 smbclient 连接到共享后，实用程序进入互动模式并显示以下提示：

```
smb: \>
```

要在互动 shell 中显示所有可用命令，请输入：

```
smb: \> help
```

要显示特定命令的帮助信息，请输入：

```
smb: \> help command_name
```

有关交互式 shell 中可用命令的详情和说明，请参阅 smbclient(1)man page。

16.1.9.5.1. 在交互模式中使用 smbclient

如果您使用不带 -c 参数的 smbclient，则实用程序将进入交互模式。

以下流程演示了如何连接到 SMB 共享并从子目录下载文件：

使用 smbclient从 SMB 共享下载文件

1. 连接到共享：

```
~]# smbclient -U "DOMAINpass:quotes[user_name]" //server_name/share_name
```

2. 进入 /example/ 目录：

```
smb: \> cd /example/
```

3. 列出目录中的文件：

```
smb: \example\> ls
.          D     0 Mon Sep 1 10:00:00 2017
..         D     0 Mon Sep 1 10:00:00 2017
example.txt     N  1048576 Mon Sep 1 10:00:00 2017

    9950208 blocks of size 1024. 8247144 blocks available
```

4. 下载 example.txt 文件：

```
smb: \example\> get example.txt
getting file \directory\subdirectory\example.txt of size 1048576 as example.txt
(511975,0 KiloBytes/sec) (average 170666,7 KiloBytes/sec)
```

5. 从共享断开：

```
smb: \example\> exit
```

16.1.9.5.2. 在脚本模式中使用 smbclient

如果将 -c 命令 参数传递给 smbclient，则可以自动对远程 SMB 共享执行命令。这可让您在脚本中使用 smbclient。

以下命令显示如何连接到 SMB 共享并从子目录下载文件：

-

```
~]# smbclient -U DOMAINpass:quotes[user_name] //server_name/share_name \
   -c "cd /example/ ; get example.txt ; exit"
```

16.1.9.6. 使用 smbcontrol 实用程序

smbcontrol 实用程序允许您向 smbd、nmbd、winbindd 或所有这些服务发送命令消息。这些控制消息指示服务重新载入其配置。

例 16.12. 重新加载 smbd、nmbd 和 winbindd 服务的配置

例如，要重新载入 smbd、nmbd、winbindd 的配置，将 reload-config message- type 发送到所有 目的地：

```
~]# smbcontrol all reload-config
```

详情以及可用命令消息类型的列表，请参阅 smbcontrol(1)man page。

16.1.9.7. 使用 smbpasswd 实用程序

smbpasswd 实用程序在本地 Samba 数据库中管理用户帐户和密码。

如果您以用户身份运行 命令，smb passwd 将更改用户的 Samba 密码。例如：

```
[user@server ~]$ smbpasswd
New SMB password:
Retype new SMB password:
```

如果以 root 用户身份运行 smbpasswd，您可以使用该实用程序，例如：

- 创建一个新用户：

```
[root@server ~]# smbpasswd -a user_name
New SMB password:
Retype new SMB password:
Added user user_name.
```

注意

在将用户添加到 Samba 数据库之前，您必须先在本地操作系统中创建帐户。请查看 第 4.3.1 节 “添加新用户”

- 启用 Samba 用户：

```
[root@server ~]# smbpasswd -e user_name
Enabled user user_name.
```

- 禁用 Samba 用户：

```
[root@server ~]# smbpasswd -x user_name
Disabled user user_name.
```

- 删除用户：

```
[root@server ~]# smbpasswd -x user_name
Deleted user user_name.
```

详情请查看 smbpasswd(8)man page。

16.1.9.8. 使用 smbstatus 实用程序

smbstatus 工具报告：

- 每个 smbd 守护进程的每个 PID 与 Samba 服务器的连接。此报告包括用户名、主组群、SMB 协议版本、加密和签名信息。
- 每个 Samba 共享的连接。此报告包含 smbd 守护进程的 PID、连接计算机的 IP、连接的时间戳、加密和签名信息。
- 锁定文件列表。报告条目包括更多详情，如 Opportunistic lock(oplock)类型

例 16.13. smbstatus 实用程序的输出

```
~]# smbstatus

Samba version 4.6.2
PID Username      Group       Machine            Protocol Version Encryption Signing
----------------------------------------------------------------------------------------------------------------
------
963 DOMAIN\administrator DOMAIN\domain users client-pc (ipv4:192.0.2.1:57786) SMB3_02
-      AES-128-CMAC

Service pid Machine  Connected at        Encryption Signing:
---------------------------------------------------------------------------
example 969 192.0.2.1 Mo Sep 1 10:00:00 2017 CEST -      AES-128-CMAC

Locked files:
Pid Uid  DenyMode  Access  R/W   Oplock   SharePath      Name   Time
---------------------------------------------------------------------------------------------------
969 10000 DENY_WRITE 0x120089 RDONLY LEASE(RWH) /srv/samba/example file.txt Mon
Sep 1 10:00:00 2017
```

详情请查看 **smbstatus(1)man page**。

16.1.9.9. 使用 smbtar 实用程序

smbtar 实用程序备份 SMB 共享的内容或其子目录，并将内容存储在 tar 存档中。或者，您可以将内容写入磁带设备。

例如，在 //server/example/ 共享中备份 demo 目录的内容，并将内容存储在 /root/example.tar 归档中：

```
~]# smbtar -s server -x example -u user_name -p password -t /root/example.tar
```

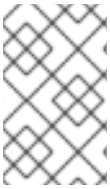
详情请查看 **smbtar(1)man page**。

16.1.9.10. 使用 testparm 实用程序

请参阅 第 16.1.2 节 “使用 testparm 实用程序验证 smb.conf 文件”。

16.1.9.11. 使用 wbinfo 实用程序

The wbinfo 实用程序查询并返回 winbindd 服务创建和使用的信息。



注意

winbindd 服务必须配置并运行才能使用 wbinfo。

您可以使用 wbinfo，例如：

- **列出域用户：**

```
~]# wbinfo -u
AD\administrator
AD\guest
...
```

- **列出域组：**

```
~]# wbinfo -g
AD\domain computers
AD\domain admins
AD\domain users
...
```

- **显示用户的 SID：**

```
~]# wbinfo --name-to-sid="AD\administrator"
S-1-5-21-1762709870-351891212-3141221786-500 SID_USER (1)
```

- **显示域和信任的信息：**

```
~]# wbinfo --trusted-domains --verbose
Domain Name  DNS Domain      Trust Type Transitive In  Out
BUILTIN              None    Yes     Yes Yes
server               None    Yes     Yes Yes
DOMAIN1    domain1.example.com  None    Yes     Yes Yes
DOMAIN2    domain2.example.com  External  No     Yes Yes
```

详情请查看wbinfo(1)man page。

16.1.10. 其它资源

-

Red Hat Samba 软件包包含所有 Samba 命令的 man page 以及软件包安装的配置文件。例如，显示 /etc/samba/smb.conf 文件的 man page，说明您可以在此文件中设置的所有配置参数：

```
~]# man 5 smb.conf
```

- /usr/share/docs/samba-版本/ ：包含 Samba 项目提供的常规文档、示例脚本和 LDAP 架构文件。
- 红帽集群存储管理指南 ：提供有关设置 Samba 和集群 Trivial 数据库(CDTB)以共享 GlusterFS 卷中存储的目录的信息。
- Red Hat Enterprise Linux High Availability Add-on 管理 指南中的红帽高可用性群集章节中的主动/主动 Samba 服务器 介绍了如何启动 Samba 高可用性安装。
- 有关在 Red Hat Enterprise Linux 中挂载 SMB 共享的详情，请查看《红帽存储管理指南》中的对应章节。

16.2. FTP

文件传输协议(FTP)是当今 Internet 上最旧且最常用的协议之一。其用途是在网络上的计算机主机之间可靠地传输文件，而无需用户直接登录远程主机或了解如何使用远程系统。它允许用户使用一组标准简单的命令来访问远程系统上的文件。

本节概述了 FTP 协议的基础知识并介绍了 vsftpd，这是 Red Hat Enterprise Linux 中的首选 FTP 服务器。

16.2.1. 文件传输协议

FTP 使用客户端-服务器架构来通过 TCP 网络协议传输文件。由于 FTP 是相当陈旧的协议，因此它使用未加密的用户名和密码身份验证。因此，它被视为不安全的协议，除非绝对必要，否则不应使用。但是，由于 FTP 在互联网上很普遍，因此通常需要将其共享给公众。因此，系统管理员应了解 FTP 的唯一特征。

这部分论述了如何将 vsftpd 配置为建立受 TLS 保护的连接，以及如何使用 SELinux 保护 FTP 服务器的安全。完美替代 FTP 来自 OpenSSH 工具套件的 sftp。有关配置 OpenSSH 和 SSH 协议的详情请参考 第 12 章 OpenSSH。

与 Internet 上使用的大多数协议不同，FTP 需要多个网络端口才能正常工作。当 FTP 客户端应用发起与 FTP 服务器的连接时，它会在服务器上打开端口 21 - 称为 命令端口。此端口用于向服务器发出所有命令。从服务器请求的任何数据将通过数据端口返回到客户端。数据连接的端口号和数据连接初始化方式也有所不同，具体取决于客户端是以主动模式还是被动 模式请求数据。

以下定义了这些模式：

活动模式

Active 模式是 FTP 协议用于将数据传输到客户端应用的原始方法。当 FTP 客户端发起主动模式数据传输时，服务器打开从服务器上端口 20 到 IP 地址的连接，以及客户端指定的随机非特权端口（大于 1024）。这种安排意味着必须允许客户端计算机接受任何 1024 以上端口的连接。随着不安全的网络（如互联网）的增长，现在普遍使用防火墙来保护客户端机器。由于这些客户端防火墙通常拒绝来自主动模式 FTP 服务器的传入连接，因此建立被动模式。

被动模式

被动模式（如主动模式）由 FTP 客户端应用发起。从服务器请求数据时，FTP 客户端表示它希望以被动模式访问数据，服务器在服务器上提供 IP 地址和随机的非特权端口（高于 1024）。然后，客户端连接到服务器上的该端口，以下载请求的信息。

被动模式确实解决了客户端防火墙对数据连接干扰的问题，但可能会使服务器端防火墙的管理变得复杂。您可以通过限制 FTP 服务器上的非特权端口范围来减少服务器上的开放端口数量。这也简化了为服务器配置防火墙规则的过程。

16.2.2. vsftpd 服务器

高安全性 FTP 后台程序(vsftpd)设计为快速、稳定且最重要的是安全。vsftpd 是唯一随红帽企业 Linux 分发的独立 FTP 服务器，因为它能够高效而安全地处理大量连接。

vsftpd 使用的安全模式有三个主要方面：

- 强隔离特权和无特权进程 - 独立进程处理不同的任务，其中每个进程使用任务所需的最小特权运行。
- 需要升级特权的任务由必要的最少权限的进程处理 - 利用 libcap 库中找到的兼容 功能，通常需要完全 root 特权的任务可以从更低的特权进程中安全地执行。
- 大多数进程在 chroot 存放位置中运行 - 只要可能，进程都会更改为正在共享的目录；然

后，该目录被视为 chroot 存放位置。例如，如果 /var/ftp/ 目录是主共享目录，则 vsftpd 将 /var/ftp/ 重新分配到新的根目录，称为 /。这不允许新根目录中未包含的任何目录的任何潜在的恶意黑客活动。

这些安全实践对 vsftpd 如何处理请求有以下影响：

- 父进程以所需权限最少运行 - 父进程动态计算所需的特权级别，以最大程度降低风险。子进程处理与 FTP 客户端的直接交互，并以尽可能接近或无特权运行。
- 所有需要升级特权的操作都由一个小的父进程来处理 - 与 Apache HTTP 服务器 一样，vsftpd 会 启动 非特权子进程来处理传入连接。这允许特权父进程尽可能小并处理相对较少的任务。
- 来自非特权子进程的所有请求都不受父进程信任 - 父进程通过套接字接收与子进程的通信，并在操作之前检查来自子进程的所有信息的有效性。
- 与 FTP 客户端的大多数交互都由 chroot 存放中的非特权子进程来处理 - 由于这些子进程具有非特权且只能访问正在共享的目录，因此任何崩溃的进程都只允许攻击者访问共享文件。

16.2.2.1. 启动和停止 vsftpd

要在当前会话中启动 vsftpd 服务，以 root 用户身份在 shell 提示符后输入以下内容：

```
~]# systemctl start vsftpd.service
```

要在当前会话中停止该服务，以 root 用户身份输入 ：

```
~]# systemctl stop vsftpd.service
```

要重启 vsftpd 服务，以 root 用户身份运行以下命令：

```
~]# systemctl restart vsftpd.service
```

此命令将停止并立即启动 vsftpd 服务，这是编辑此 FTP 服务器的配置文件后使配置更改生效的最有效方式。另外，您可以使用以下命令仅在 vsftpd 服务已在运行时重启它：

```
~]# systemctl try-restart vsftpd.service
```

默认情况下，vsftpd 服务在引导时不自动启动。要将 vsftpd 服务配置为在引导时启动，以 root 用户身份在 shell 提示符后输入以下内容：

```
~]# systemctl enable vsftpd.service
Created symlink from /etc/systemd/system/multi-user.target.wants/vsftpd.service to
/usr/lib/systemd/system/vsftpd.service.
```

有关如何在 Red Hat Enterprise Linux 7 中管理系统服务的详情请参考 第 10 章 使用 systemd 管理服务。

16.2.2.2. 启动多个 vsftpd 副本

有时，一台计算机用于为多个 FTP 域提供服务。这是一款叫做多选的技术。使用 vsftpd 进行多主页的一种方法是运行 守护进程的多个副本，每个副本都有自己的 配置文件。

要做到这一点，首先将所有相关 IP 地址 分配给系统上的网络设备或别名网络设备。有关配置网络设备、设备别名和有关网络配置脚本的其他信息的详情，请参考 Red Hat Enterprise Linux 7 网络指南。

接下来，必须将 FTP 域的 DNS 服务器配置为引用正确的计算机。有关 BIND、Red Hat Enterprise Linux 中使用的 DNS 协议实施及其配置文件的详情，请查看 Red Hat Enterprise Linux 7 网络指南。

要让 vsftpd 应答不同 IP 地址 上的请求，必须正在运行 守护进程的多个副本。为了协助启动 vsftpd 守护进程的多个实例，vsftp 软件包中提供了一个特殊的 systemd 服务单元(vsftpd @.service)，用于将 vsftpd 启动为实例化服务。

要使用此服务单元，必须为 FTP 服务器的每个必需实例创建单独的 vsftpd 配置文件，并将其放置在 /etc/vsftpd/ 目录中。请注意，每个配置文件都必须具有唯一名称（如 /etc/vsftpd/vsftpd-site-2.conf），并且必须仅可由 root 用户读取和写入。

在侦听 IPv4 网络的每个 FTP 服务器的每个配置文件中，以下指令必须是唯一的：

```
listen_address=N.N.N.N
```

使用所提供服务的 FTP 站点 的唯一 IP 地址 替换 N.N.N.N。如果站点使用 IPv6，则改为使用 listen_address6 指令。

旦 /etc/vsftpd/ 目录中存在多个配置文件，即可以 root 用户身份执行以下命令来启动 vsftpd 守护进程的各个实例：

```
~]# systemctl start vsftpd@configuration-file-name.service
```

在以上命令中，将 configuration-file-name 替换为所请求服务器配置文件的唯一名称，如 vsftpd-site-2。请注意，命令中不应包含配置文件的 .conf 扩展名。

如果要一次启动多个 vsftpd 守护进程实例，您可以使用 systemd 目标单元文件(vsftpd.target)，该文件在 vsftpd 软件包中提供。此 systemd 目标会使 /etc/vsftpd/ 目录中的每个可用的 vsftpd 配置文件启动独立的 vsftpd 守护进程。以 root 用户身份执行以下命令以启用目标：

```
~]# systemctl enable vsftpd.target
Created symlink from /etc/systemd/system/multi-user.target.wants/vsftpd.target to
/usr/lib/systemd/system/vsftpd.target.
```

以上命令将 systemd 服务管理器配置为在启动时启动 vsftpd 服务（以及配置的 vsftpd 服务器实例）。要在不重启系统的情况下立即启动该服务，以 root 用户身份执行以下命令：

```
~]# systemctl start vsftpd.target
```

有关如何使用 systemd 目标管理服务的更多信息，请参阅 第 10.3 节 “使用 systemd 目标”。

要考虑逐个服务器进行更改的其他指令有：

- *anon_root*
- *local_root*
- *vsftpd_log_file*
- *xferlog_file*

16.2.2.3. 使用 TLS 加密 vsftpd 连接

为了消除默认在不加密的情况下传输用户名、密码和数据的 FTP 本质上不安全的性质，可以将 vsftpd 守护进程配置为使用 TLS 协议来验证连接并加密所有传输。请注意，支持 TLS 的 FTP 客户端需要与启用 TLS 的 vsftpd 通信。

注意

SSL （安全套接字层）是较旧安全协议实施的名称。新版本称为 TLS (Transport Layer Security)。只有较新版本(TLS)应使用，因为 SSL 遭受严重安全漏洞。vsftpd 服务器附带的文档以及 vsftpd .conf 文件中使用的配置指令，在引用与安全相关的事务时使用 SSL 名称，但在 ssl_enable 指令设置为 YES 时，默认支持并使用 TLS。

将 vsftpd.conf 文件中的 ssl_enable 配置指令设置为 YES 以启用 TLS 支持。启用 ssl_enable 选项时自动激活的其他 TLS相关指令的默认设置提供了配置合理的、配置良好的 TLS 设置。此外，这包括要求仅将 TLS v1 协议用于所有连接（默认禁用使用不安全 SSL 协议版本）或强制所有非匿名登录以使用 TLS 发送密码和数据传输。

例 16.14. 配置 vsftpd 使用 TLS

在本例中，配置指令在 vsftpd.conf 文件中明确禁用了旧的安全协议的 SSL 版本：

```
ssl_enable=YES
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
```

在修改了 vsftpd 服务的配置后重启它：

```
~]# systemctl restart vsftpd.service
```

有关由 vsftpd 调整 TLS的使用，请参阅 vsftpd.conf(5)手册页以了解其他与 TLS 相关的配置指令 。

16.2.2.4. vsftpd 的 SELinux 策略

SELinux 策略管理 vsftpd 后台程序（以及其他 ftpd 进程 ），定义了强制访问控制，默认情况下，它基于最少所需的访问权限。要允许 FTP 守护进程访问特定文件或目录，需要为其分配适当的标签。

例如，为了能够以匿名方式共享文件，必须将 public_content_t 标签分配给要共享的文件和目录。您可以使用 chcon 命令作为 root 用户完成此操作：

```
~]# chcon -R -t public_content_t /path/to/directory
```

在以上命令中，将 /path/to/directory 替换为您要为其分配该标签的目录的路径。同样，如果要设置用于上传文件的目录，则需要为特定目录分配 public_content_rw_t 标签。此外，allow_ftpd_anon_write SELinux 布尔值选项必须设置为 1。以 root 用户身份使用 setsebool 命令进行此操作：

```
~]# setsebool -P allow_ftpd_anon_write=1
```

如果您希望本地用户能够通过 FTP 访问其主目录（这是 Red Hat Enterprise Linux 7 中的默认设置），则需要将 ftp_home_dir 布尔值选项设置为 1。如果允许 vsftpd 以单机模式运行（Red Hat Enterprise Linux 7 上默认启用），则还需要将 ftpd_is_daemon 选项设置为 1。

有关如何配置与 FTP 相关的 SELinux 策略的更多信息，请参阅 ftpd_selinux(8)手册页，包括其他有用的标签和布尔值选项示例。另外，请参阅 Red Hat Enterprise Linux 7 SELinux 用户和管理员指南中有关 SELinux 的详情。

16.2.3. 其它资源

有关 vsftpd 的更多信息，请参阅以下资源。

16.2.3.1. 安装的文档

- /usr/share/doc/vsftpd-version-number/ 目录 - 使用已安装的 vsftpd 软件包版本替换 version -number。此目录包含一个含有软件基本信息的 README 文件。The TUNING 文件包含基本的性能调优提示，SEC URITY/ 目录包含 vsftpd 使用的安全模型的信息。

- vsftpd- 相关的 man page - 守护进程和配置文件有多个 man page。下表列出了一些更重要的 man page。

  服务器应用程序

  {blank}

  - vsftpd(8)- 描述 vsftpd 的可用命令行选项。

  配置文件

> {blank}

- vsftpd.conf(5)- 包含适用于 vsftpd 的配置文件中可用选项的详细列表。

- hosts_access(5)- 描述 TCP 打包程序配置文件中可用的格式和选项：host.allow 和 hosts.deny。

与 SELinux 的交互

> {blank}

- ftpd_selinux(8)- 包含管理 ftpd 进程的 SELinux 策略的说明，以及 SELinux 标签的分配方式和布尔值设置的说明。

16.2.3.2. 在线文档

关于 vsftpd 和 FTP

> {blank}

- http://vsftpd.beasts.org/ - vsftpd 项目页面非常适合查找最新的文档并联系软件作者。

- http://slacksite.com/other/ftp.html - 这个网站简要解释了主动模式和被动模式 FTP 之间的区别。

Red Hat Enterprise Linux 文档

> {blank}

- 红帽企业 Linux 7 联网指南 - 红帽企业 Linux 7 的网络指南记录了有关在此系统中配置和管理网络接口、网络和网络服务的相关信息。它介绍了 hostnamectl 实用程序，并说明了如何使用它在本地和远程命令行上查看和设置主机名。

- Red Hat Enterprise Linux 7 SELinux 用户和管理员指南 - Red Hat Enterprise Linux 7 的 SELinux 用户和管理员指南介绍了 SELinux 的基本原理，并详细介绍了如何配置和使用 SELinux 与 Apache HTTP 服务器、Post fix、PostgreSQL 或 OpenShift 等服务。它解释了如何为 systemd 管理的系统服务配置 SELinux 访问权限。

- 红帽企业 Linux 7 安全指南 - 红帽企业 Linux 7 安全指南 帮助用户和管理员学习保护工作站和服务器免受本地和远程入侵、攻击和恶意活动的流程和实践。它还说明了如何保护关键系统服务。

相关的 RFC 文档

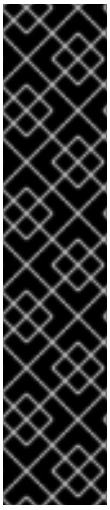
{blank}

- RFC 0959 - IETF FTP 协议的原始 注释请求(RFC)。
- RFC 1123 - 小型 FTP相关部分可扩展并阐明 RFC 0959。
- RFC 2228 - FTP 安全扩展.vsftpd 实施支持 TLS 和 SSL 连接所需的小子集。
- RFC 2389 - 建议 FEAT 和 OPTS 命令.
- RFC 2428 - IPv6 支持.

16.3. 打印设置

"打印设置" 工具可用于配置打印机、维护打印机配置文件、打印假脱机目录和打印过滤器，以及打印机类管理。

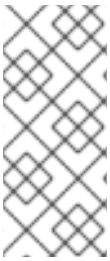
该工具基于通用 Unix 打印系统(CUPS)。如果您从之前使用 CUPS 的 Red Hat Enterprise Linux 版本升级系统，升级过程会保留配置的打印机。



重要

cupsd.conf man page 记录了 CUPS 服务器的配置。它包含用于启用 SSL 支持的指令。但是，CUPS 不允许控制所使用的协议版本。由于 POODLE SSLv3.0 漏洞(CVE-2014-3566)中描述的组件存在安全漏洞，红帽建议您不依赖这些组件通过配置设置禁用 SSLv3。建议您使用 stunnel 提供安全隧道并禁用 SSLv3。有关使用 stunnel 的更多信息，请参阅 Red Hat Enterprise Linux 7 安全指南。

对于到远程系统的打印设置工具的临时安全连接，请使用 X11 通过 SSH 进行转发，如 第 12.4.1 节 “X11 转发” 所述。



注意

您可以直接从 CUPS Web 应用或命令行对打印机执行相同的操作和其他操作。要在 Web 浏览器中访问应用程序，请访问 http://localhost:631/。有关 CUPS 手册，请参考网站的 Home 选项卡上的链接。

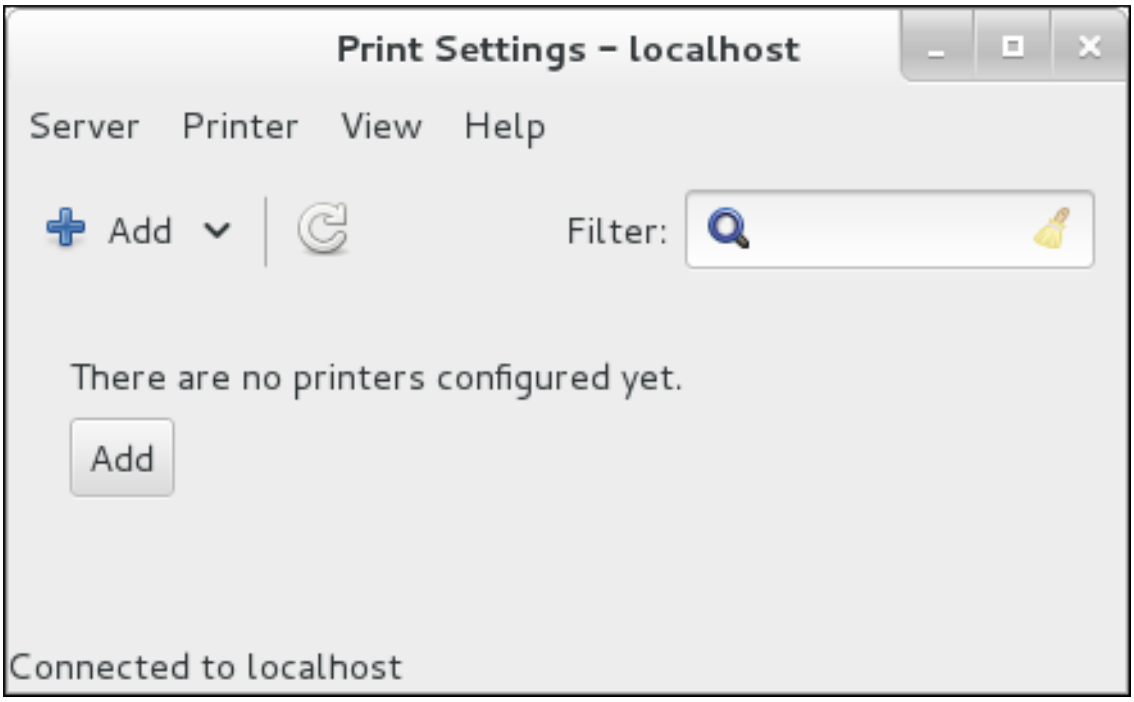
16.3.1. 启动打印设置配置工具

通过 "打印设置" 配置工具，您可以对现有打印机执行各种操作并设置新打印机。您还可以直接使用 CUPS （运行 http://localhost:631/ 访问 CUPS Web 应用程序）。

要从命令行启动 "打印设置" 工具，请在 shell 提示符下键入 system-config-printer。这时将显示 "打印设置" 工具。或者，如果使用 GNOME 桌面，按 Super 键进入 "活动概览"，键入 "打印设置"，然后按 Enter 键。这时将显示 "打印设置" 工具。Super 键显示在各种 guis 中，具体取决于键盘和其他硬件，但通常作为 Windows 或 Command 键（通常在 空格栏 的左侧）。

此时将显示 图 16.1 “打印设置窗口” 中描述的 Print Settings 窗口。

图 16.1. 打印设置窗口



16.3.2. 启动打印机设置

打印机设置流程因打印机队列类型而异。

如果您要设置与 USB 连接的本地打印机，则会自动发现并添加打印机。系统将提示您确认要安装的软件包并提供管理员或 root 用户密码。需要手动设置与其他端口类型和网络打印机连接的本地打印机。

按照以下步骤启动手动打印机设置：

1. 启动打印设置工具（请参阅 第 16.3.1 节 “启动打印设置配置工具”）。
2. 转至 "服务器"" → 新 → 打印机 "。
3. 在 Authenticate 对话框中，输入管理员或 root 用户密码。如果您首次配置远程打印机，系统将提示您授权调整防火墙。
4. 选择打印机连接类型，并在右侧区域中提供其详细信息。

16.3.3. 添加本地打印机

按照以下步骤添加与串行端口以外的本地打印机：

1. **打开 "添加 打印机"对话框（请参考 第 16.3.2 节 “启动打印机设置”）。**

2. **如果设备没有自动显示，请在左侧列表中选择连接到该打印机的端口（如 Serial Port #1 或 LPT #1）。**

3. **在右侧输入连接属性：**

   对于其他

   URI （例如 file:/dev/lp0）

   对于 串行端口

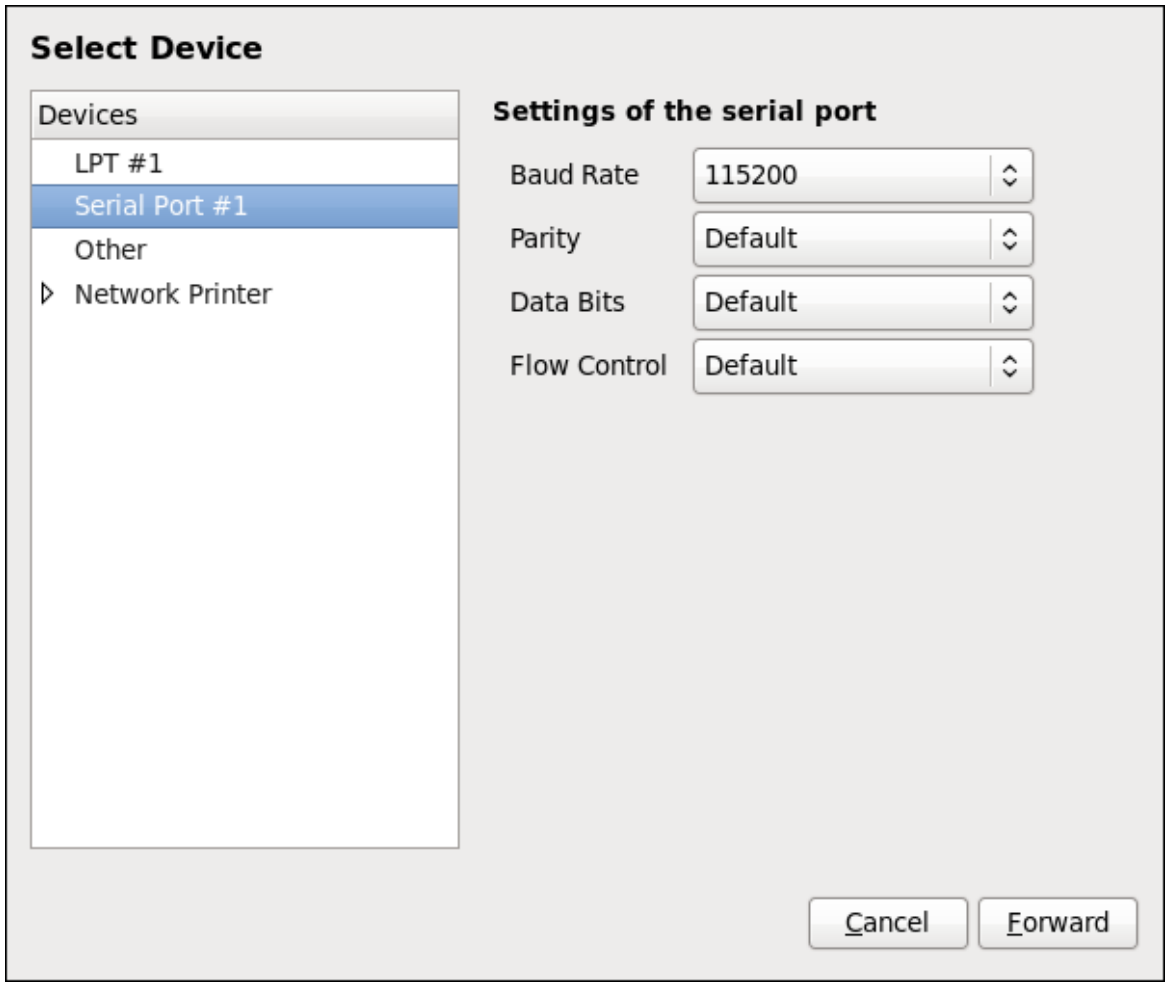
   Baud Rate

   奇偶校验

   数据范围

   流控制

图 16.2. 添加本地打印机



4. 点 Forward。

5. 选择打印机型号。详情请查看 第 16.3.8 节 “选择打印机模型和完成”。

16.3.4. 添加 AppSocket/HP JetDirect 打印机

按照以下步骤添加 AppSocket/HP JetDirect 打印机：

1. 打开 新打印机 对话框（请参阅 第 16.3.1 节 “启动打印设置配置工具”）。

2. 在左侧的列表中，选择 Network Printer → AppSocket/HP JetDirect。

3. 在右侧输入连接设置：

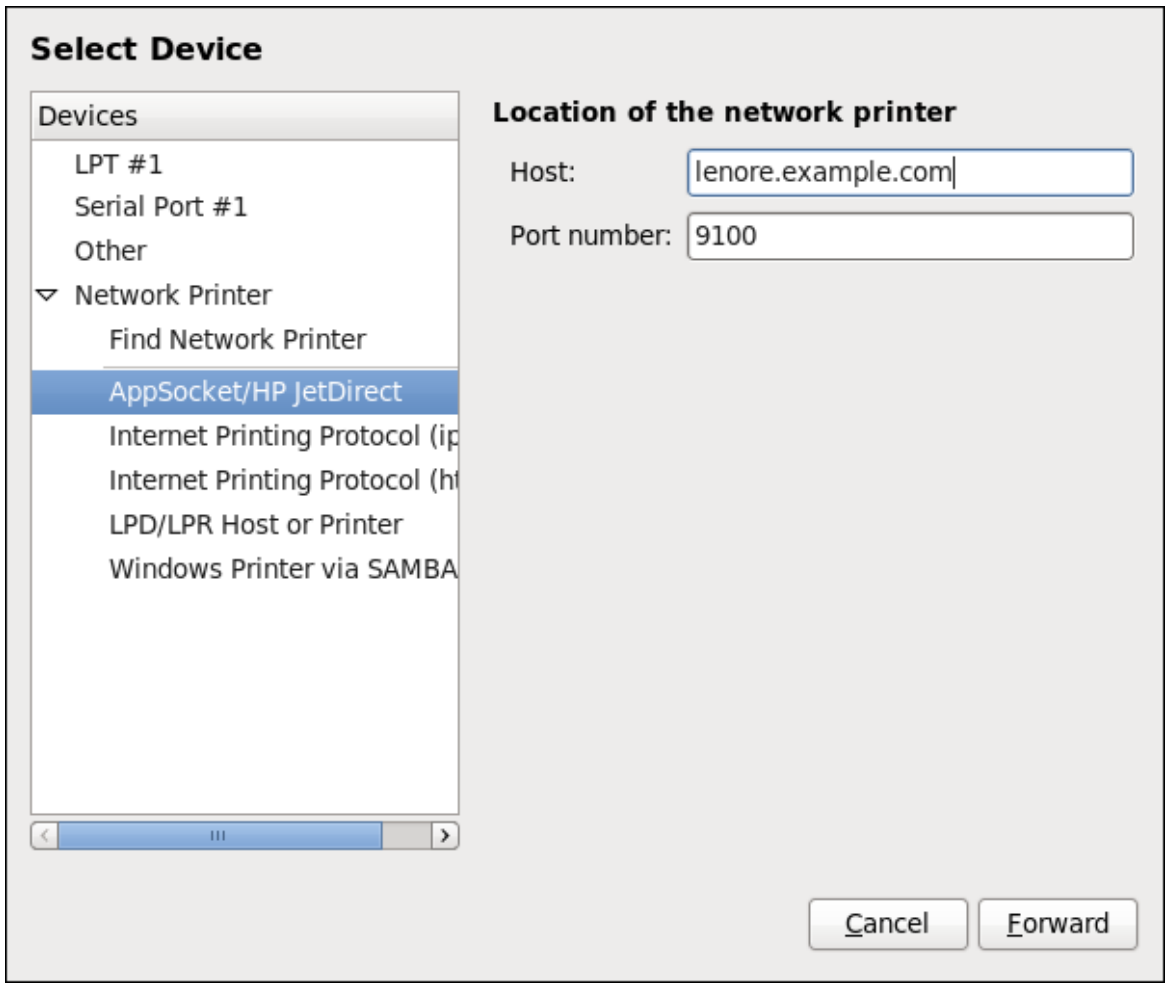
主机名

打印机主机名或 IP 地址.

端口号

打印作业侦听打印机端口（默认为 9100 ）。

图 **16.3. 添加 JetDirect 打印机**



4. **点 Forward。**

5. 选择**打印机型号。** 详情请查看 **第 16.3.8 节 “选择打印机模型和完成”。**

16.3.5. 添加 IPP 打印机

IPP 打印机是连接到同一 TCP/IP 网络上不同系统的打印机。系统连接此打印机可以运行 CUPS 或仅配置为使用 IPP。

如果在打印机服务器上启用了防火墙，则必须将防火墙配置为允许端口 631 上的传入 TCP 连接。请注意，CUPS 浏览协议允许客户端计算机自动发现共享的 CUPS 队列。要启用此功能，必须将客户端计算机上的防火墙配置为允许端口 631 上传入的 UDP 数据包。

按照以下步骤添加 IPP 打印机：

1. 打开 新打印机 对话框（请参阅 第 16.3.2 节 “启动打印机设置”）。

2. 在左侧的设备列表中，选择网络打印机和 Internet 打印协议(ipp)或 Internet 打印协议 (https)。

3. 在右侧输入连接设置：

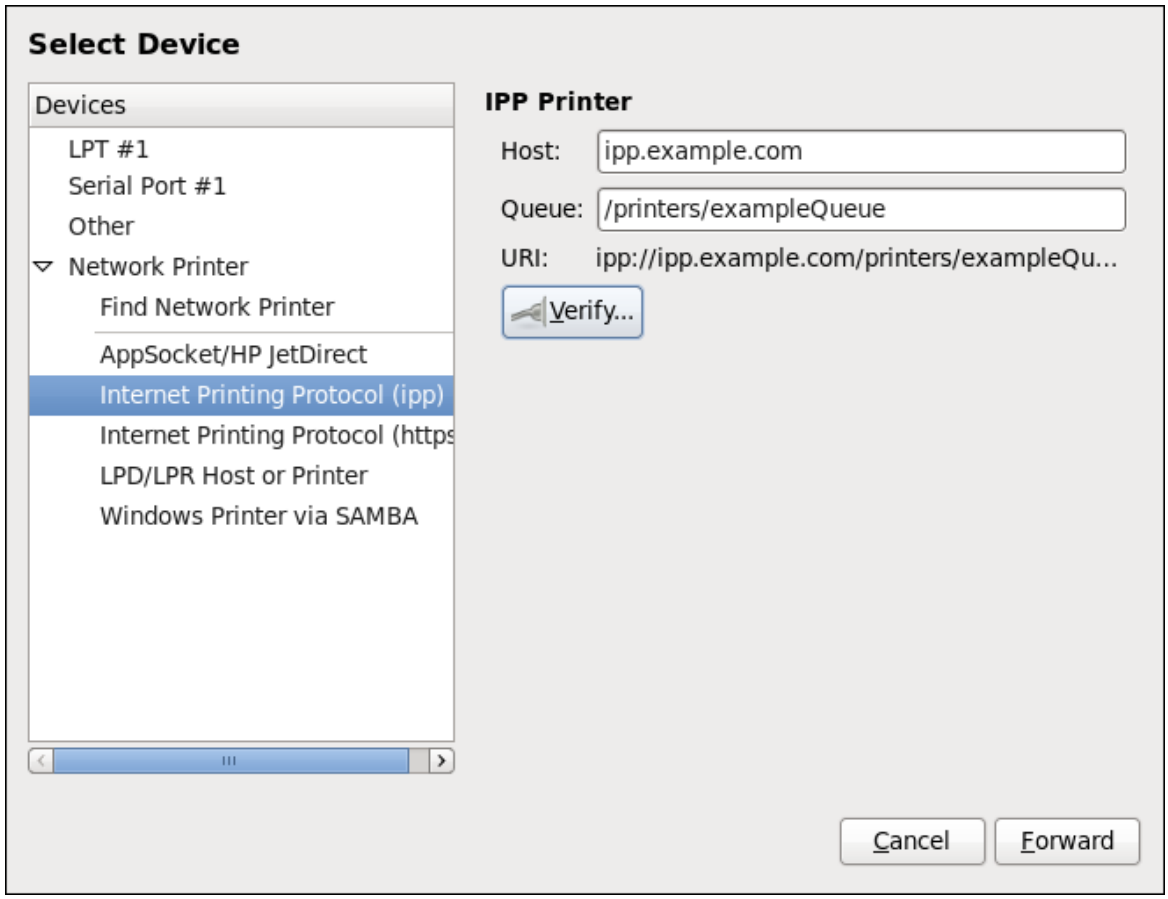
主机

IPP 打印机的主机名。

Queue

要提供给新队列的队列名称（如果该框留空，则将使用基于设备节点的名称）。

图 16.4. 添加 IPP 打印机



4. 单击"下一步"以继续。

5. 选择打印机型号。详情请查看 第 16.3.8 节 “选择打印机模型和完成”。

16.3.6. 添加 LPD/LPR 主机或打印机

按照以下步骤添加 LPD/LPR 主机或打印机：

1. 打开 新打印机 对话框（请参阅 第 16.3.2 节 “启动打印机设置”）。

2. 在左侧的设备列表中，选择 Network Printer → LPD/LPR Host 或 Printer。

3. 在右侧输入连接设置：

主机

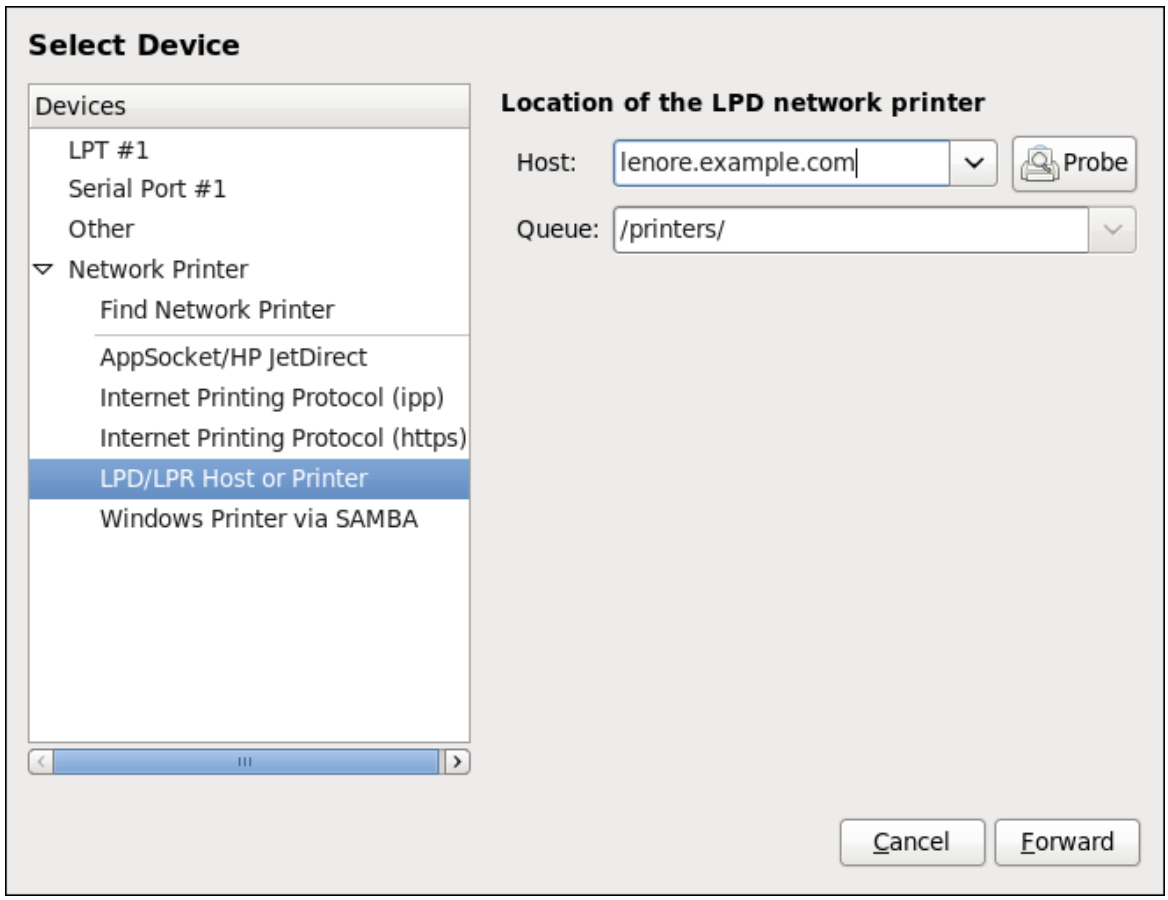
LPD/LPR 打印机或主机的主机名。

（可选）点击 Probe 来查找 LPD 主机上的队列。

Queue

要提供给新队列的队列名称（如果该框留空，则将使用基于设备节点的名称）。

图 16.5. 添加 LPD/LPR 打印机

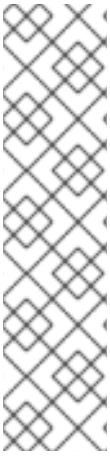


4. 单击"下一步"以继续。

5. 选择打印机型号。详情请查看 第 16.3.8 节 “选择打印机模型和完成”。

16.3.7. 添加 Samba(SMB)打印机

按照以下步骤添加 Samba 打印机：



注意

请注意，要添加 Samba 打印机，您需要安装 samba-client 软件包。您可以以 root 用户身份运行：

```
yum install samba-client
```

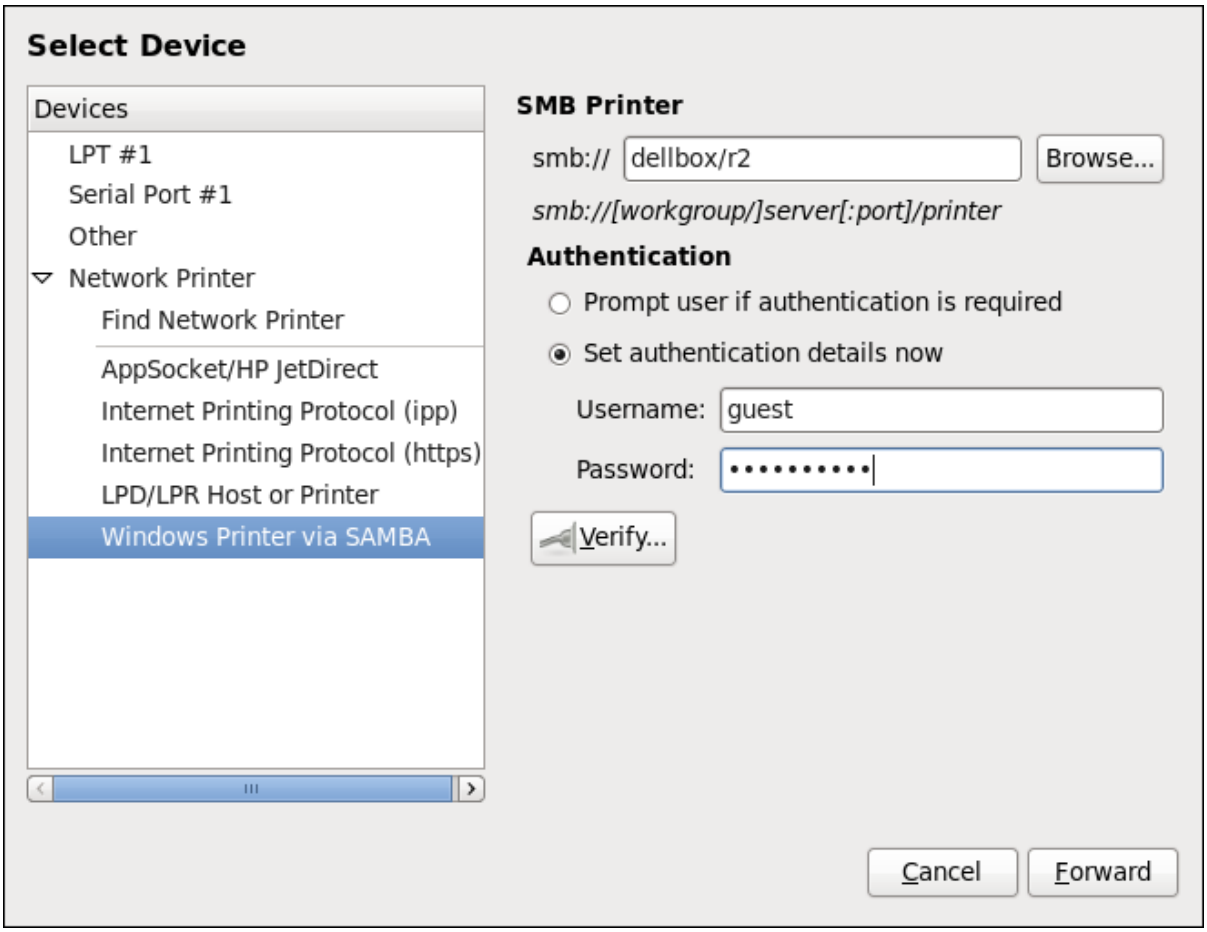
有关使用 Yum 安装软件包的更多信息，请参阅 第 9.2.4 节 “安装软件包”。

1. 打开 新打印机 对话框（请参阅 第 16.3.2 节 “启动打印机设置”）。

2. 在左侧的列表中，通过 SAMBA 选择网络打印机 Windows 打印机打印机。

3. 在 smb:// 字段中输入 SMB 地址。使用格式计算机名称/打印机共享。在 图 16.6 “添加 SMB 打印机” 中，计算机名称为 dellbox，打印机共享 是 r2。

图 16.6. 添加 SMB 打印机



4.

单击 **Browse 以查看可用的工作组/域。要仅显示特定主机的队列，请键入主机名（NetBios 名称）并单击 Browse。**

5. 选择**其中任何一个选项：**

   a. **如果需要身份验证，提示用户 ：打印文档时会从用户收集用户名和密码。**

   b. 现在 设置验证详情：现在 提供验证信息，以便在以后不需要。在 "用户名" 字段中，输入要访问打印机的用户名。此用户必须存在于 SMB 系统中，用户必须具有访问打印机的权限。默认用户名通常是 Windows 服务器的 guest，或 nobody （Samba 服务器）。

6. 为 **Username 字段中指定的用户 输入密码 （如果需要）。**

警告

Samba 打印机用户名和密码作为 root 和 LinuxPrinting Daemon lpd 可读的未加密文件存储在打印机服务器上。因此，对打印机服务器具有 root 访问权限的其他用户可以查看您用于访问 Samba 打印机的用户名和密码。

因此，当您选择要访问 Samba 打印机的用户名和密码时，建议您选择不同于用于访问本地 Red Hat Enterprise Linux 系统的密码。

如果 Samba 打印服务器上共享了文件，建议它们也使用不同于打印队列使用的密码。

7. **点 Verify 测试连接。验证成功后，会出现一个对话框，确认打印机共享可访问性。**

8. **点 Forward。**

9. 选择**打印机型号。详情请查看 第 16.3.8 节 “选择打印机模型和完成”。**

16.3.8. 选择打印机模型和完成

正确选择打印机连接类型后，系统会尝试获取驱动程序。如果进程失败，您可以手动查找或搜索驱动程序资源。

按照以下步骤提供打印机驱动程序并完成安装：

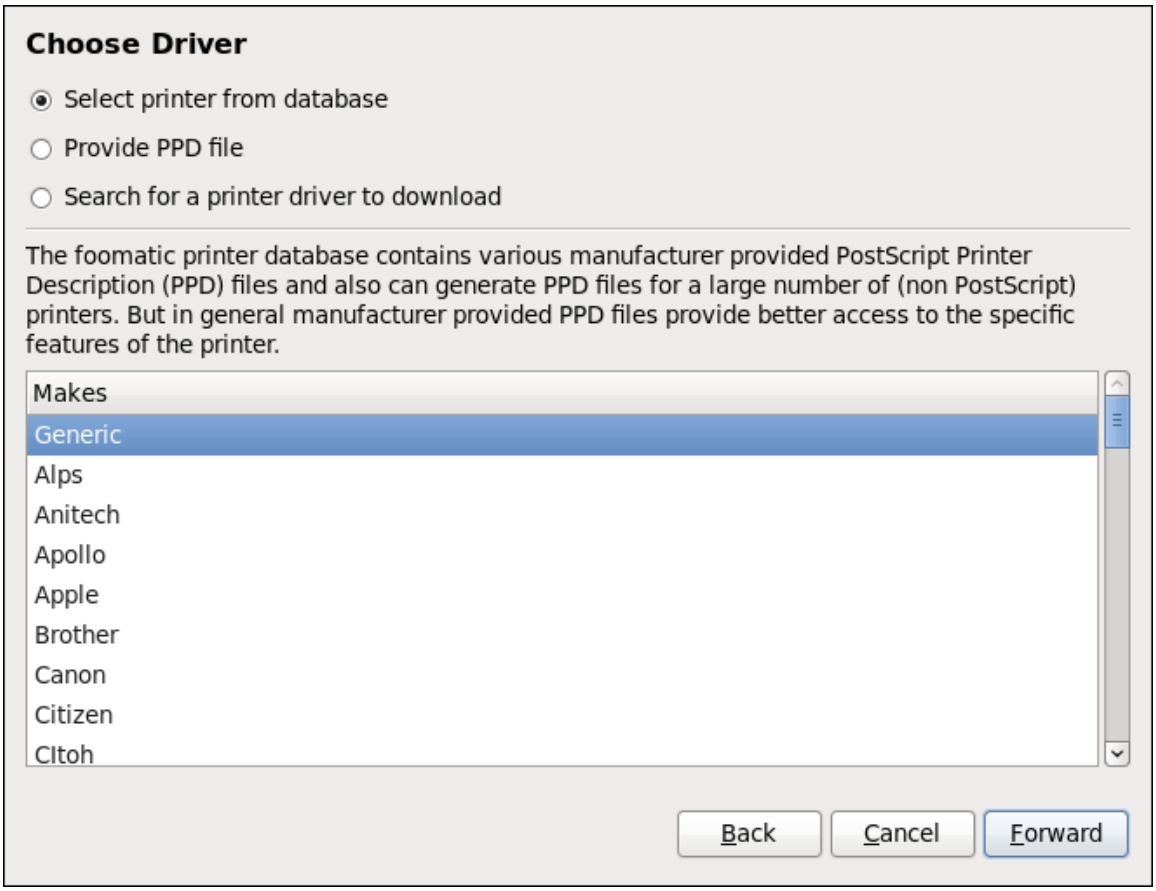
1. 在自动驱动程序检测失败后显示的窗口中，选择以下选项之一：

   a. 从数据库中选择打印机 - 系统会根据打印机的选择版本从 Makes 列表中选择驱动程序。如果没有列出打印机型号，请选择 Generic。

   b. 提供 PPD 文件 - 系统使用提供的 PostScript 打印机描述 (PPD)文件进行安装。PPD 文件也可与打印机一同提供，就像制造商通常提供的一样。如果 PPD 文件可用，您可以选择此选项并使用选项描述下的浏览器栏来选择 PPD 文件。

   c. 搜索打印机驱动程序以进行下载 - 在 Make 和 model 字段中输入打印机制作和型号，以便在 OpenPrinting.org 上搜索相应的软件包。

图 16.7. 选择打印机品牌



2. 根据您之前选择的选项，在显示的区域中提供详情：

   - Select printer from database 选项的打印机品牌。
   - Provide PPD 文件选项的 PPD 文件 位置。
   - 用于 搜索打印机驱动程序以进行打印机制作和型号以下载 选项.

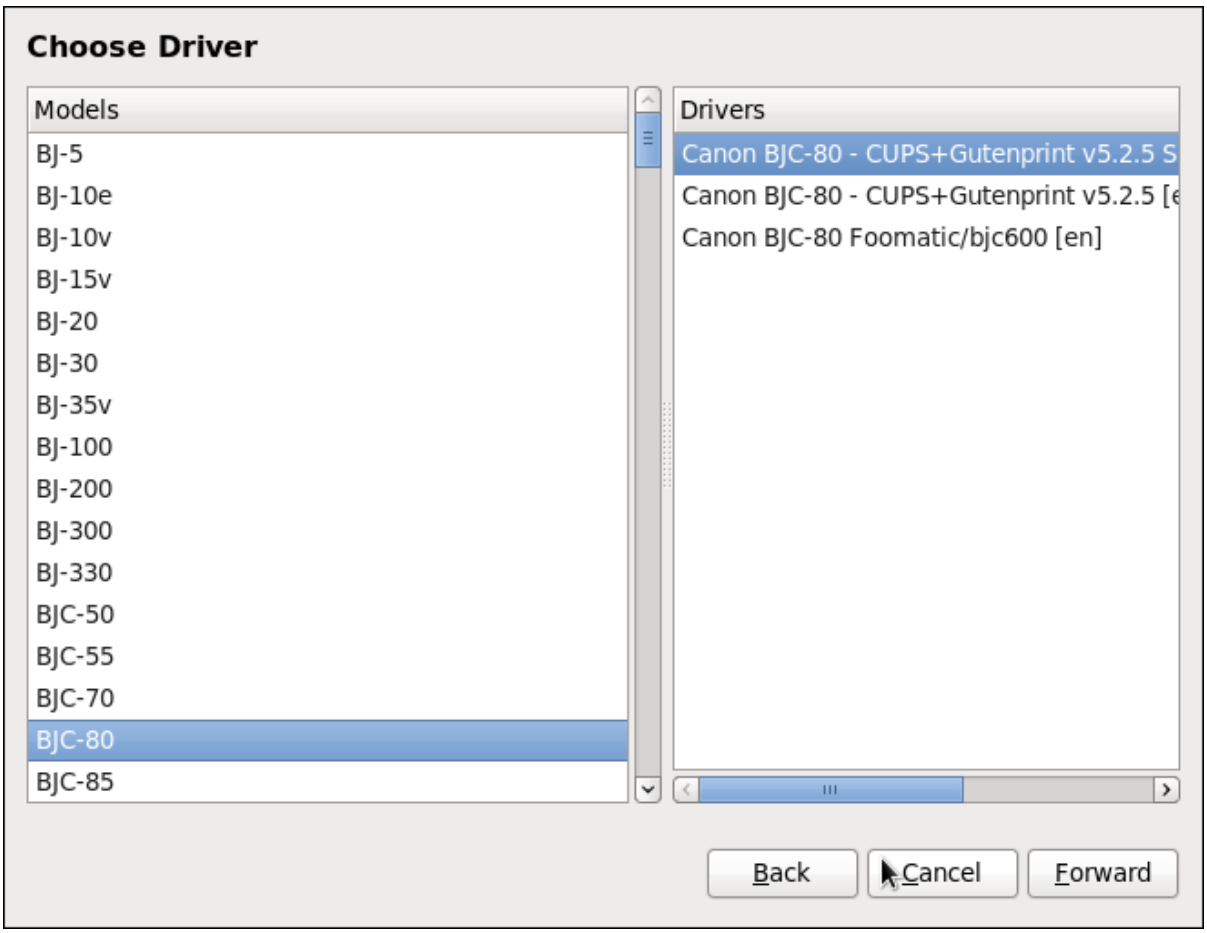
3. 单击"下一步"以继续。

4. 如果适用于您的选项，将显示 图 16.8 “选择打印机型号” 中显示的窗口。在左侧的 Models 列中选择对应的模型。

注意

在右侧，会自动选择推荐的打印机驱动程序；但是，您可以选择另一个可用的驱动程序。打印驱动程序处理您要打印到打印机能够理解的格式中的数据。由于本地打印机已直接连接到您的计算机，因此您需要一个打印机驱动程序来处理发送到打印机的数据。

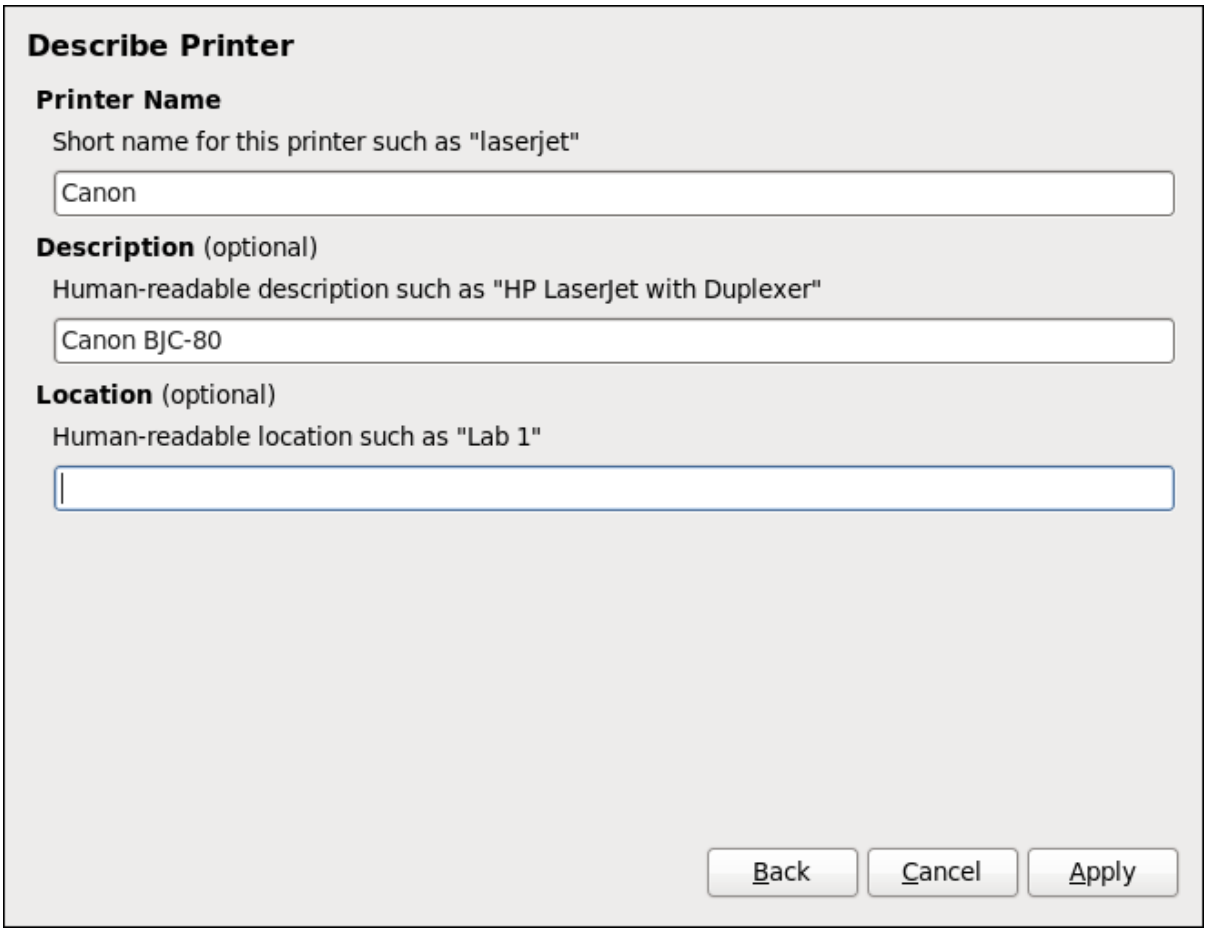
图 16.8. 选择打印机型号



5. 点 Forward。

6. 在 "描述打印机" 下，在"打印机名称 "字段中输入打印机 的唯一名称。打印机名称可以包含字母、数字、短划线(-)和下划线（）；它必须不包含任何空格。您还可以使用 Description 和 Location 字段添加更多打印机信息。这两个字段都是可选的，可以包含空格。

图 16.9. 打印机设置



7. 单击 **Apply** 以确认打印机配置并添加打印队列（如果设置正确无误）。单击 **Back** 以修改打印机配置。

8. 应用更改后，系统会显示一个对话框，供您打印测试页面。单击 **Yes** 以打印测试页面。另外，您可以稍后打印测试页面，如 第 16.3.9 节 “打印测试页面” 所述。

16.3.9. 打印测试页面

设置打印机或更改打印机配置后，请输出测试页面以确保打印机正常工作：

1. 在打印 窗口中右键单击打印机并单击 **属性**。

2. 在 **Properties** 窗口中，单击左侧的 **Settings**。

3. 在显示的 **Settings** 选项卡上，单击 **" 打印测试页面 "按钮**。

16.3.10. 修改现有打印机

要删除现有打印机，请在 "打印设置" 窗口中，选择打印机并转至 "打印机 → 删除 "。确认删除打印机。或者，按 Delete 键。

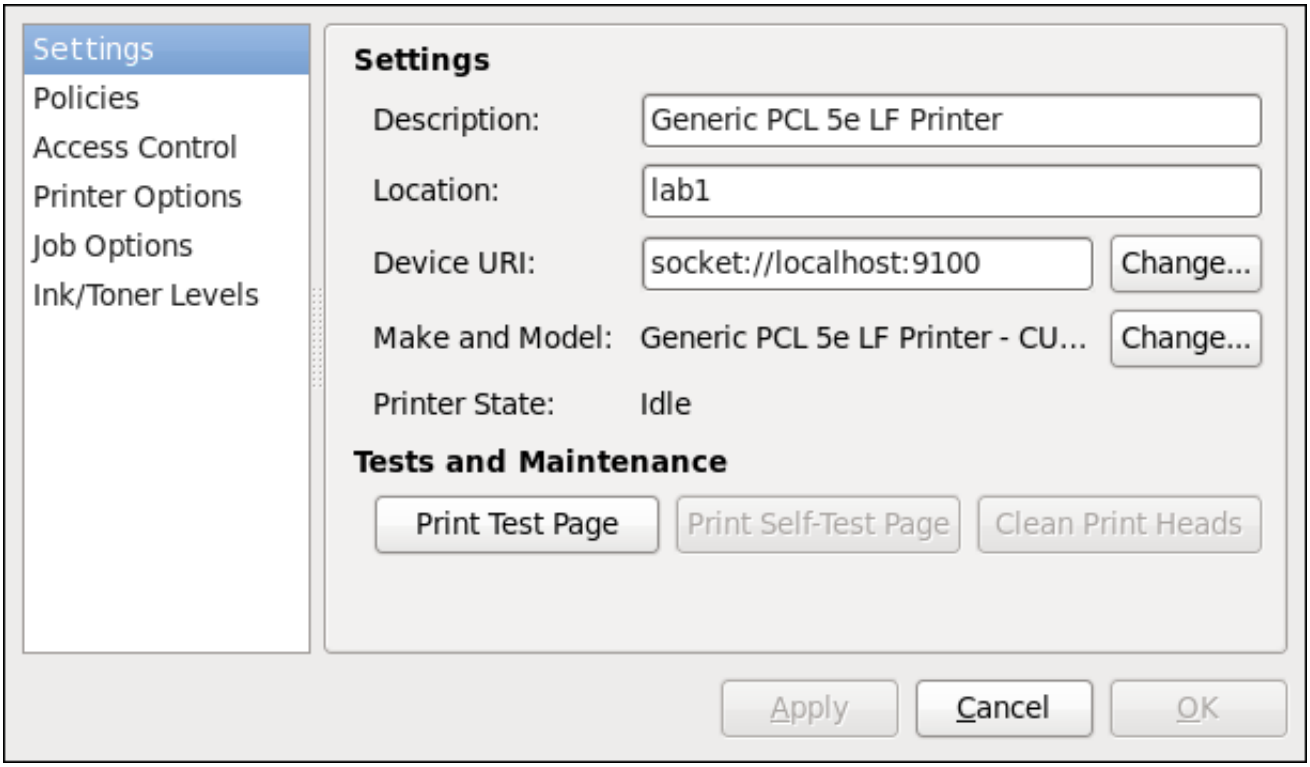
要设置默认打印机，请在打印机列表中右键单击打印机，然后单击上下文菜单中的" 设置为默认 "按钮。

16.3.10.1. 设置页面

要更改打印机驱动程序配置，请双击 打印机 列表中对应的名称，然后单击左侧的 Settings 标签以显示 Settings 页面。

您可以修改打印机设置，如 make 和 model、打印测试页面、更改设备位置(URI)等等。

图 16.10. 设置页面



16.3.10.2. 策略页面

单击左侧的" 策略 "按钮，以更改打印机状态和打印输出中的设置。

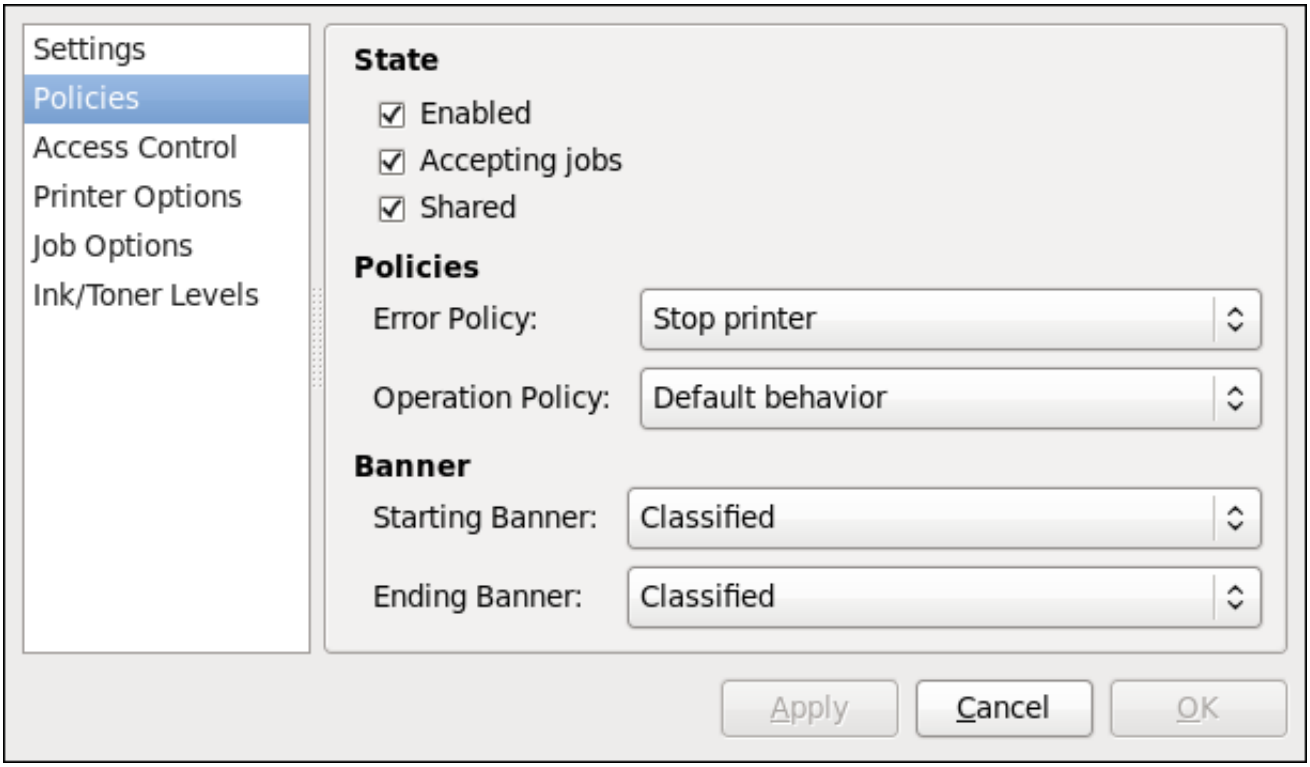
您可以选择打印机状态，配置打印机的错误策略（您可以在发生错误时决定中止打印作业、重试或停止）。

您还可以创建一个 横幅页面 （描述打印作业的各个方面，如原始打印机、来自作业的用户名和打印的文件的安全状态）：单击 Starting Banner 或 Ending Banner 下拉菜单并选择最能描述打印作业性质的选项（例如，机密）。

16.3.10.2.1. 共享打印机

在 Policies 页面中，您可以将打印机标记为共享：如果共享打印机，则网络上发布的用户可以使用它。要允许打印机的共享功能，请转至 "服务器"" → 设置 "，然后选择" 发布连接到此系统的共享打印机 "。

图 16.11. 策略页面



确保防火墙允许传入 TCP 连接的端口 631，即网络打印服务器(IPP)协议的端口。要允许 IPP 流量通过 Red Hat Enterprise Linux 7 上的防火墙，请使用 firewalld 的IPP 服务。要做到这一点，请按以下方法操作：

在 firewalld 中启用 IPP 服务

1. 要启动图形化 firewall-config 工具，请按 Super 键进入"活动概览"，键入 firewall，然后按 Enter 键。此时将打开 防火墙配置 窗口。系统将提示您输入管理员或 root 密码。

另外，要使用命令行启动图形防火墙配置工具，以 root 用户身份输入以下命令：

```
~]# firewall-config
```

此时将打开防火墙配置窗口。

查找左下角的"连接"。这表明 firewall-config 工具已连接到用户空间守护进程 firewalld。

要立即更改当前的防火墙设置，请确保标记为 Configuration 的下拉菜单已设置为 Runtime。或者，若要编辑要在下一次系统启动或重新加载时应用的设置，请从下拉列表中选择永久。

2. 选择" 区域 "选项卡，然后选择与要使用的网络接口对应的防火墙区域。默认值为 public 区域。Interfaces 选项卡显示已分配给区域的接口。

3. 选择" 服务 "选项卡，然后选择 ipp 服务以启用共享。访问网络打印机需要 ipp-client 服务。

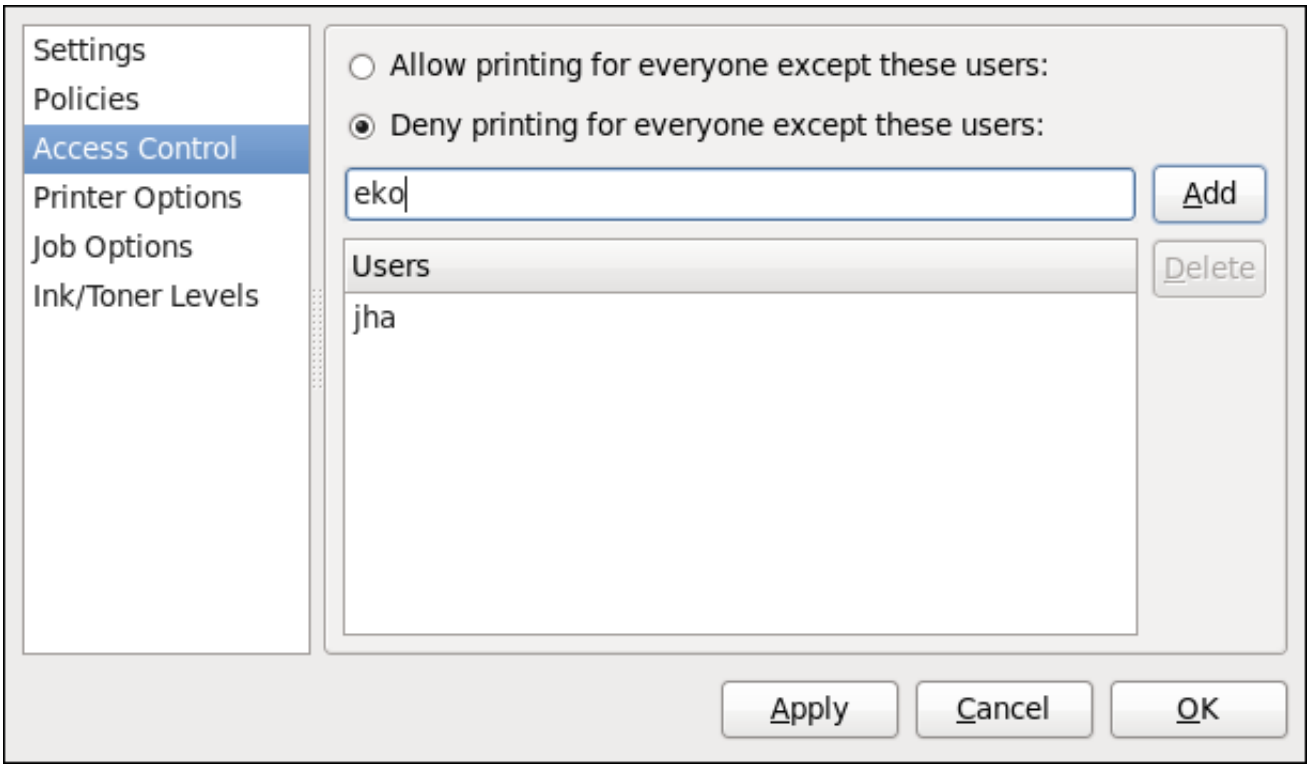
4. 关闭 firewall-config 工具。

有关在 firewalld 中打开和关闭端口的更多信息，请参阅 Red Hat Enterprise Linux 7 安全指南。

16.3.10.2.2. 访问控制页面

您可以更改用户级别对 Access Control 页面中配置的打印机的访问。单击左侧的 Access Control 标签，以显示该页面。为除这些用户以外的每个人选择"允许打印 "或"拒绝打印"，然后定义 下面设置的用户：在文本框中输入用户名，然后单击添加 按钮 将用户添加到用户集。

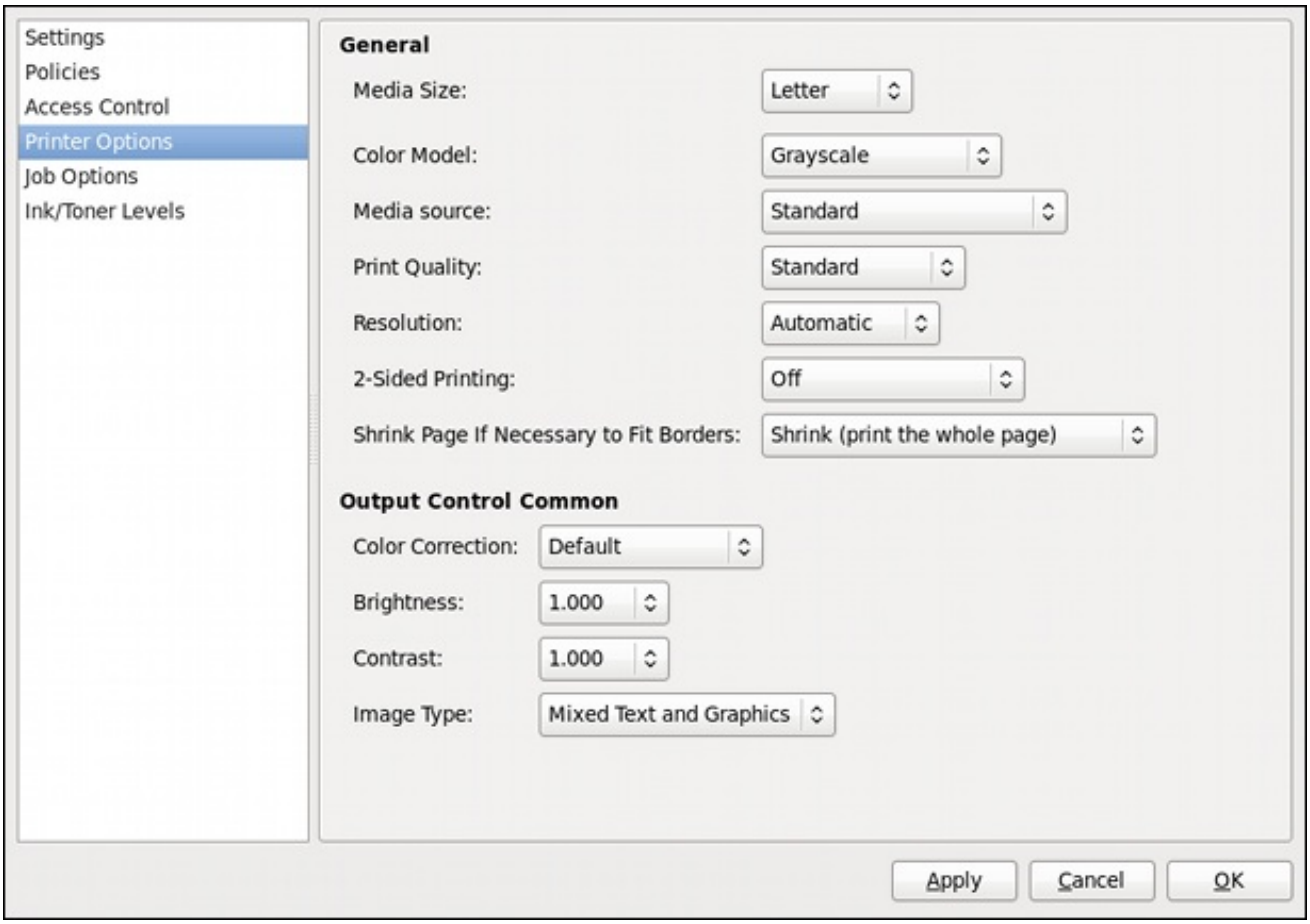
图 16.12. 访问控制页面



16.3.10.2.3. 打印机选项页面

打印机选项 页包含打印机媒体和输出的各种配置选项，其内容可能因打印机而异。它包含一般打印、纸张、质量和打印大小设置。

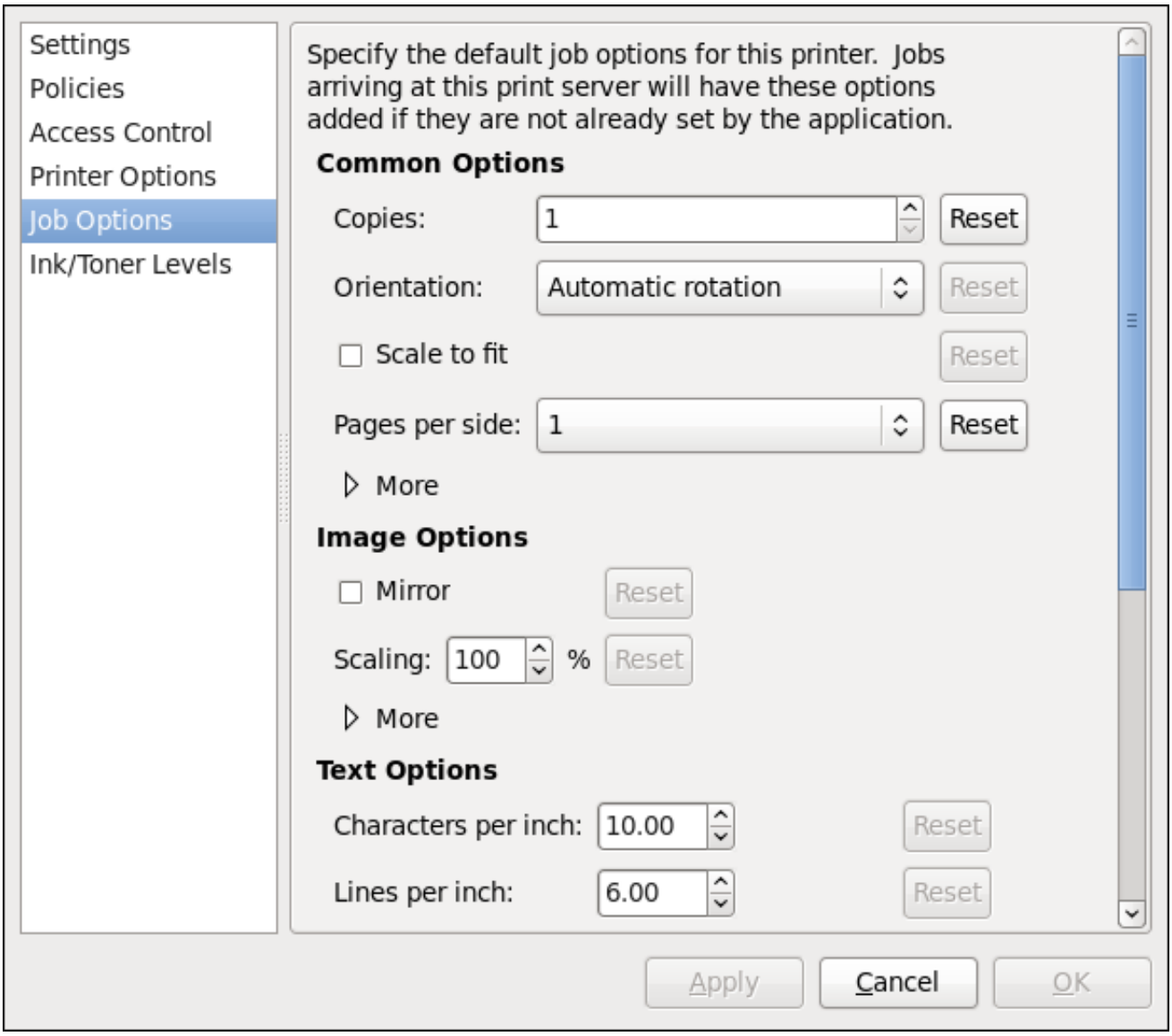
图 16.13. 打印机选项页面



16.3.10.2.4. 作业选项页面

在 Job Options 页面中，您可以详细描述打印机作业选项。单击左侧的 Job Options 标签，以显示该页面。编辑默认设置以应用自定义作业选项，如副本数、方向数、每侧页面数、扩展（增加或减少可打印区域的大小，可用于将超大小的打印区域融入到较小的打印介质物理表）、详细的文本选项和自定义作业选项。

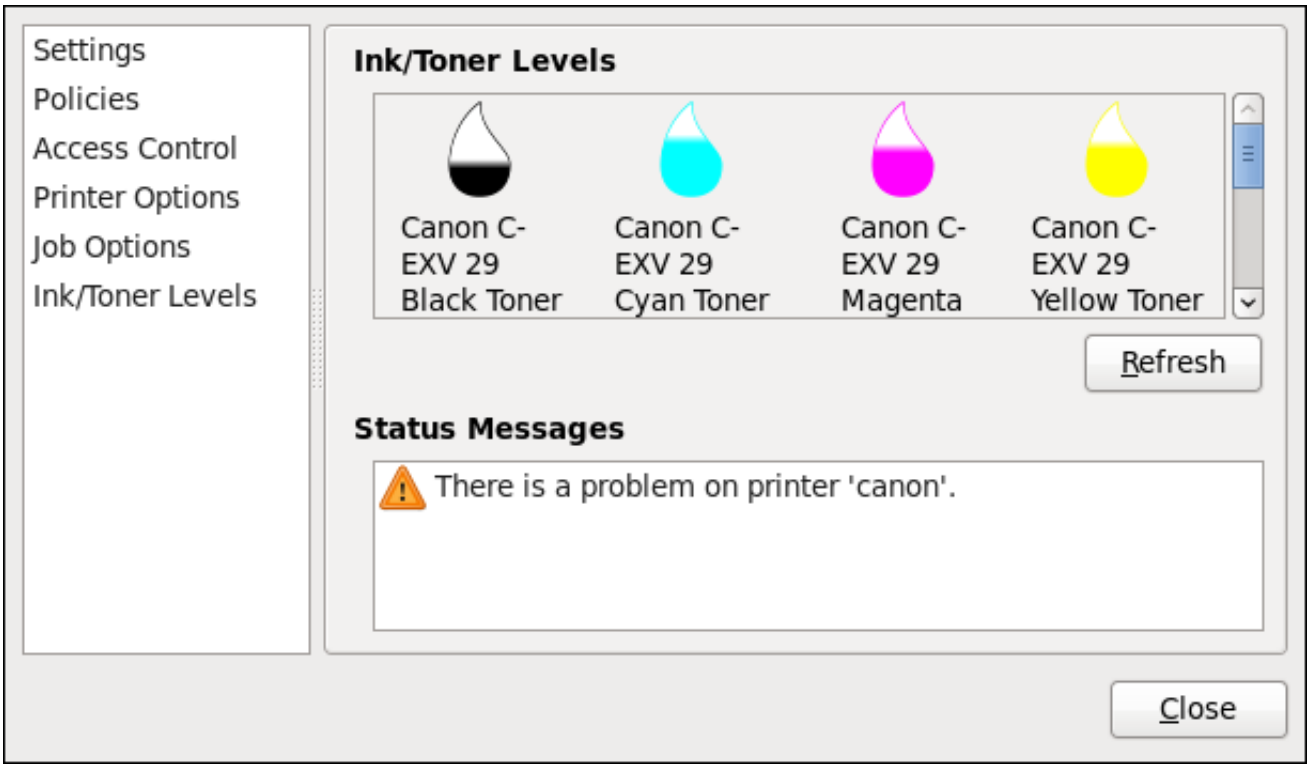
图 16.14. Job Options 页



16.3.10.2.5. ink/ner 级别页

Ink/Toner Levels 页面 包含有关 toner 状态（如果可用）和打印机状态的详细信息。单击左侧的 Ink/Toner Levels 标签，以显示该页面。

图 16.15. ink/ner Levels 页面

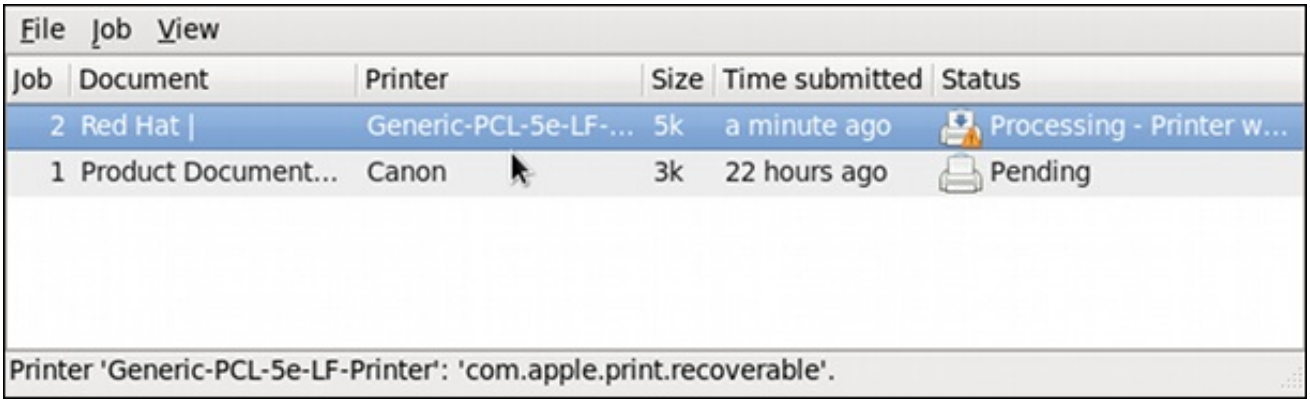


16.3.10.3. 管理打印作业

当您向打印机守护进程发送打印作业时，如从 Emacs 打印文本文件或从 GIMP 打印图像，打印作业将添加到打印假脱机队列中。打印假脱机队列是已发送到打印机的打印作业列表以及有关每个打印请求的信息，如请求的状态、作业编号等。

在打印过程中，打印机状态图标将显示在面板上的通知区域中。要检查打印作业的状态，请点击 Printer Status，它会显示一个类似 图 16.16 “GNOME 打印状态” 的窗口。

图 16.16. GNOME 打印状态



要取消、搁置、释放、打印或验证打印作业，请在 GNOME 打印状态中选择作业，并在 Job 菜单中单击相应的命令。

若要从 shell 提示符在打印假脱机中查看打印作业列表，请键入命令 lpstat -o。最后几行类似如下：

例 16.15. lpstat -o 输出示例

```
$ lpstat -o
Charlie-60       twaugh      1024  Tue 08 Feb 2011 16:42:11 GMT
Aaron-61         twaugh      1024  Tue 08 Feb 2011 16:42:44 GMT
Ben-62           root        1024  Tue 08 Feb 2011 16:45:42 GMT
```

如果要取消打印作业，请使用命令 lpstat -o 查找请求的作业编号，然后使用命令 取消 作业编号。例如： 取消 60 将取消 例 16.15 “lpstat -o 输出示例” 中的打印作业。您不能通过 cancel 命令取消其他用户启动的打印作业。但是，您可以通过发出 cancel -U root job_number 命令来强制删除此类作业。要防止此类取消，请将打印机操作策略更改为 Authenticated，以强制进行 root 身份验证。

您还可以直接从 shell 提示符打印文件。例如，命令 lp sample.txt 会打印文本文件 sample.txt。打印过滤器决定了该文件的类型，并将其转换为打印机可理解的格式。

16.3.11. 其它资源

要了解有关在 Red Hat Enterprise Linux 中打印的更多信息，请参见以下资源：

安装的文档

- lp(1) - lp 命令的 man page，允许您从命令行打印文件。
- lpr(1) - lpr 命令的 man page，允许您从命令行打印文件。
- cancel(1) - 命令行实用工具的 man page，用于从打印队列中删除打印作业。
- mpage(1) - 命令行实用程序的 man page，用于在一张纸上打印多个页面。
- cupsd(8) - CUPS 打印机守护进程的 man page。
- cupsd.conf(5) - CUPS 打印机守护进程配置文件的 man page。

- *class.conf(5) - CUPS 的类配置文件的 man page。*

- *lpstat(1) - lpstat 命令的 man page，显示有关类、作业和打印机的状态信息。*

在线文档

- *http://www.linuxprinting.org/ - Linux Foundation 网站上的 OpenPrinting 组包含有关在 Linux 中打印的大量信息。*

- *http://www.cups.org/ - CUPS 网站提供有关 CUPS 的文档、常见问题和新闻组。*

第 17 章 数据库服务器

本章将引导您完成 MariaDB 服务器的安装和配置，这是基于 MySQL 技术的开源快速而强大的数据库服务器。本章还介绍了如何备份 MariaDB 数据。

17.1. MARIADB

MariaDB 是一个关系数据库，将数据转换为结构化信息，并为访问数据提供 SQL 接口。它包括多种存储引擎和插件，以及地理信息系统(GIS)。

红帽企业 Linux 7 包含 MariaDB 5.5，作为来自 MySQL 数据库系列服务器的默认实施。MariaDB 数据库服务器的新版本可作为 Red Hat Enterprise Linux 6 和 Red Hat Enterprise Linux 7 的 Software Collections 提供。有关最新版本的更多信息，请参阅 Red Hat Software Collections 发行注记。

17.1.1. 安装 MariaDB 服务器

要安装 MariaDB 服务器，请按照以下步骤执行：

安装 MariaDB 服务器

1. 确保 mariadb 和 mariadb-server 软件包安装在所需的服务器上：

```
~]# yum install mariadb mariadb-server
```

2. 启动 mariadb 服务：

```
~]# systemctl start mariadb.service
```

3. 在引导时启用 mariadb 服务：

```
~]# systemctl enable mariadb.service
```

17.1.1.1. 提高 MariaDB 安装安全性

要提高安装 MariaDB 服务器时的安全性，您可以运行 mysql_secure_installation 命令：

```
~]# mysql_secure_installation
```

此命令将启动完全交互式脚本，该脚本会提示您输入流程中的每个步骤。这个脚本能够通过以下方式提高安全性：

- 为 root 帐户设置密码
- 删除匿名用户
- 不允许远程（本地主机以外的）root 登录
- 删除 test 数据库

17.1.2. 为网络配置 MariaDB 服务器

要为联网配置 MariaDB 服务器，请使用 /etc/my.cnf. d/server.cnf 文件的 [mysqld] 部分，您可以在其中设置以下配置指令：

- **bind-address**

  bind-address 是服务器要侦听的地址。

  可能的选项有：主机名、IPv4 地址或 IPv6 地址。

- **skip-networking**

  可能的值有：

  0 - 侦听所有客户端

  1 - 仅侦听本地客户端

- **port**

  MariaDB 侦听 TCP/IP 连接的端口。

17.1.3. 备份 MariaDB 数据

从 MariaDB 数据库备份数据的方法主要有两种：

- 逻辑备份
- 物理备份

17.1.3.1. 逻辑备份

逻辑备份由恢复数据所需的 SQL 语句组成。这种类型的备份在纯文本文件中导出信息和记录。

与物理备份相比，逻辑备份的主要优势在于可移植性和灵活性。数据可以在其他硬件配置、MariaDB 版本或数据库管理系统(DBMS)上恢复，这无法通过物理备份来实现。

警告

只有 mariadb.service 正在运行时才能执行逻辑备份。逻辑备份不包括日志和配置文件。

17.1.3.2. 物理备份

物理备份由存储内容的文件和目录的副本组成。

与逻辑备份相比，物理备份具有以下优点：

- 输出更为紧凑。
- 备份的大小会较小。
- 备份和恢复速度更快。
- 备份包括日志和配置文件。

警告

当 mariadb.service 没有运行或者数据库中的所有表被锁定以防止备份期间更改时，必须执行物理备份。

第 18 章 使用 CHRONY 套件配置 NTP

因为许多原因，保持准确的时间非常重要。例如在网络中，需要准确的数据包和日志的时间戳。在 Linux 系统中，NTP 协议是由在用户空间运行的守护进程实现的。

用户空间守护进程更新内核中运行的系统时钟。系统时钟可以通过使用不同的时钟源来维护系统的时间。通常，使用 时间戳计数器（TSC）。TSC 是一个 CPU 寄存器，它计算从上次重置的循环数。它非常快，分辨率很高，且不会被中断。

在守护进程 ntpd 和 chronyd 之间有一个选择，分别来自 ntp 和 chrony 软件包中的软件仓库。

本章论述了 chrony 套件的使用。

18.1. CHRONY 套件简介

Chrony 是网络时间协议(NTP)的一种实现。您可以使用 Chrony:

- 要将系统时钟与 NTP 服务器同步，
- 将系统时钟与参考时钟同步，如ASCII 接收器.
- 要将系统时钟与手动时间输入同步，
- 作为 NTPv4(RFC 5905) 服务器或对等服务器，为网络中的其他计算机提供时间服务。

Chrony 在各种条件下表现良好，包括网络间连接、高度拥塞的网络、温度变化（普通计算机时钟对温度敏感）以及不持续运行或在虚拟机上运行的系统。

通过互联网镜像同步的两天机器之间的准确性通常在几毫秒之内，而对于 LAN 中的机器则为几十微秒。硬件时间戳或硬件参考时钟可以提高同步到子微秒级别的两台计算机之间的准确性。

Chrony 包括 chronyd、一个在用户空间运行的守护进程和 chronyc （可用来监控 chronyd 性能并在运行时更改各种操作参数的命令行程序）。

18.1.1. ntpd 和 chronyd 之间的区别

chronyd 可以优于 ntpd:

- chronyd 可以正常工作，其中对时间参考的访问是间歇性的，而 ntpd 需要定期轮询时间引用才能正常工作。
- 即使网络在较长时间内拥塞，chronyd 也能表现良好。
- chronyd 通常可以更快准确地同步时钟。
- chronyd 能够快速适应时钟速率的突然变化，例如，由于碳粉电器温度的变化，ntpd 可能需要很长时间才能再次下降。
- 在默认配置中，chrony d 从不调整时钟在系统启动时同步的时间，以便不设置其他正在运行的程序。ntpd 也可以配置为从不调整时间，但是它必须使用不同的调整时钟的方法，这种方法存在一些缺点，包括对时钟准确性的负面影响。
- chronyd 可以调整较大范围内 Linux 系统上的时钟速率，即使在时钟中断或不稳定的机器上运行。例如，在某些虚拟机上：
- chronyd 比较小，使用较少的内存，仅在需要时才会唤醒 CPU，这更有利于节能。

chronyd 可以执行 ntpd 不能执行的操作：

- chronyd 支持隔离的网络，其中唯一的调整时间方法是手动输入。例如，管理员看一下时钟。chronyd 可以检查在不同更新中更正的错误，以估算计算机获得或丢失时间的速率，并随后使用此估算来调整计算机时钟。
- chronyd 支持降低实时时钟的增益或丢失率，例如维护关闭计算机时间的时钟。当系统引导时，它可以使用实时时钟调整时间值来设置系统时间。这些实时时钟设施目前仅在 Linux 系统上可用。

- chronyd 支持 Linux 上的硬件时间戳，允许在本地网络上进行非常准确的同步。

ntpd 可以执行 chronyd 不能执行的操作：

- ntpd 支持 NTP 版本 4(RFC 5905)的所有工作模式，包括广播、多播和多播客户端和服务器。请注意，广播和多播模式本质上也不如普通服务器和客户端模式准确且安全性较低，因此通常应避免使用。

- ntpd 支持自动密钥协议(RFC 5906)，用于通过公钥加密对服务器进行身份验证。请注意，该协议已被证明不安全，并可能替换为网络时间安全(NTS)规范的实施。

- ntpd 包含许多参考时钟的驱动程序，而 chronyd 依赖于其他程序（如 gpsd ）使用共享内存(SHM)或 Unix 域套接字(SOCK)从参考时钟访问数据。

18.1.2. 在 NTP 守护进程之间选择

除由不支持 chrony 的工具管理或监控的系统外，chrony 应该 首选所有系统，或者具有硬件参考时钟且无法与 chrony 一起使用的系统。

注意

执行使用 Autokey 协议验证数据包的系统只能与 ntpd 一起使用，因为 chronyd 不支持这个协议。Autokey 协议存在严重的安全问题，因此应避免使用此协议。使用对称密钥的身份验证（由 chronyd 和 ntpd 支持），而不是 Autokey。Chrony 支持 SHA256 和 SHA512 等更强大的哈希功能，而 ntpd 只能使用 MD5 和 SHA1。

18.2. 了解 CHRONY 及其配置

18.2.1. 了解 chronyd 和 chronyc

chrony 守护进程（chronyd） 可以由命令行工具 chronyc 监控和控制。这个工具提供了一个命令提示，它允许输入多个命令来查询 chronyd 的当前状态并更改其配置。在默认情况下，chronyd 只接受来自本地 chronyc 实例的命令，但它也可以被配置为接受来自远程主机的监控命令。应该限制远程访问。

18.2.2. 了解 chrony 配置命令

chronyd 的默认配置文件是 /etc/chrony.conf。-f 选项可以用来指定备选配置文件路径。更多选项请

查看 chronyd man page。

以下是 chronyd 配置选项选择：

注释

注释需要以 #、%、; 或 ! 开始

allow

（可选）指定一个主机、子网或网络来允许 NTP 连接到作为 NTP 服务器的机器。默认是不允许连接。

例 18.1. 使用 allow 选项授予访问权限：

- 使用这个命令授予对 IPv4 的访问权限：

```
allow 192.0.2.0/24
```

- 使用这个命令授予对 IPv6 的访问权限：

```
allow 2001:0db8:85a3::8a2e:0370:7334
```

注意

需要在防火墙中打开 UDP 端口号 123 以便允许客户端访问：

```
~]# firewall-cmd --zone=public --add-port=123/udp
```

如果您想永久打开端口 123，请使用 --permanent 选项：

```
~]# firewall-cmd --permanent --zone=public --add-port=123/udp
```

cmdallow

这与 allow 指令（请参阅 allow 类似，不同的地方是它允许到特定子网或主机的访问控制（而不是 NTP 客户端访问）。（访问控制表示 chronyc 可以在这些主机上运行并成功地连接到这个计算机上的 chronyd。）其语法是相同的。还有一个 cmddeny all 指令，它与 cmdallow all 指令类似。

dumpdir

在 chronyd 重启后保存测量历史记录的目录路径（假设系统时钟行为在未运行期间不会进行任何更改）。如果使用此功能（通过配置文件中的 dumponexit 命令，或者 chronyc中的 dump onexit 命令），应使用 dumpdir 命令来定义保存测量历史记录的目录。

dumponexit

如果存在此命令，则表示 chronyd 应在程序退出时保存其每个时间源的测量历史记录。（请参阅上面的 dumpdir 命令）。

hwtimestamp

hwtimestamp 指令启用硬件时间戳进行非常准确的同步。详情请查看 chrony.conf(5) 手册页。

local

local 关键字用于允许 chronyd 从客户端轮询它的角度实时显示同步，即使它没有当前的同步源。这个选项通常用于隔离网络中的"主（master）"计算机，其中需要几台计算机同步，同时使用手动输入将"master"与实时保持一致。

这个命令的示例如下：

```
local stratum 10
```

较大的值 10 表示时钟与参考时钟非常低，使其时间不可靠。如果计算机访问了另一个计算机并最终与参考时钟同步，则几乎可以肯定这个值会小于 10。因此，本地 命令选择高值（如 10）可防止计算机本身与实时混淆，这让它始终暴露给对真实服务器可见的客户端。

log

log 命令表示要记录某些信息。它接受以下选项：

measurements

这个选项将原始 NTP 测量以及相关信息记录到名为 measurements.log 的文件。

statistics

这个选项将有关递归处理的信息记录到名为 statistics.log 的文件。

tracking

这个选项将系统速率或丢失率估算的更改记录到名为 trace.log 的文件。

rtc

这个选项记录有关系统实时时钟的信息。

refclocks

这个选项将原始和过滤的参考时钟测量记录到名为 **refclocks.log** 的文件。

tempcomp

这个选项将温度测量和系统速率记录到名为 **tempcomp.log** 的文件。

日志文件被写入 **logdir** 命令指定的目录中。

这个命令的示例如下：

```
log measurements statistics tracking
```

logdir

该指令允许指定记录文件所在的目录。

使用这个指令的示例如下：

```
logdir /var/log/chrony
```

makestep

通常 **chronyd** 会根据需要降低或加快时钟速度，让系统逐渐修正任何时间误差。在某些情况下，系统时钟可能还没有改变，这个修正过程需要很长时间才能修正系统时钟。如果调整大于阈值，则这个指令会强制 **chronyd** 对系统时钟进行步骤，但前提是自 **chronyd** 启动以来没有更多时钟更新（一个负值可用来禁用限制）。这在使用参考时钟时特别有用，因为 **initstepslew** 指令只适用于 NTP 源。

使用这个指令的示例如下：

```
makestep 1000 10
```

如果调整大于 1000 秒，这会增加系统时钟，但仅在前十个时钟更新中。

maxchange

这个指令在时钟更新中设定了最大允许的误差。检查只能在指定数量的更新后执行，以便进行大量系统时钟的初始调整。当误差大于指定的最大值时，它将忽略指定的次数，然后 chronyd 将放弃并退出（一个负值可用来永不退出）。在这两种情况下都会向 syslog 发送一条信息。

使用这个指令的示例如下：

```
maxchange 1000 1 2
```

在第一次时钟更新后，chronyd 会在每次时钟更新时检查误差，它将忽略两个大于 1000 秒的调整，然后退出。

maxupdateskew

chronyd 的任务之一就是根据计算机的参考源确定运行时钟的速度或速度。此外，它计算了围绕估算值进行错误绑定的估算。

如果错误范围太大，这表明测量结果尚未下降，而且估计的增益或损失率不是非常可靠。

maxupdateskew 参数是确定一个估算是否不可靠而无法使用的阈值。默认情况下，阈值是 1000 ppm。

语法格式为：

```
maxupdateskew skew-in-ppm
```

skew-in-ppm 的典型值可能是 100（通过电话线拨号连接到服务器），5 或 10（LAN 中的计算机）。

需要注意的是，这不是唯一的防止使用不可靠估算的方法。在任何时候，chronyd 都会跟踪估计的增益或丢失率，以及估算时绑定的错误。当根据其中一个来源进行另一个计算后产生一个新的估算时，会使用加权组合算法更新主估算。因此，如果 chronyd 有一个高度可靠的 master 估算值，并且生成了一个有大错误约束的新估计值，那么在新的 master 估计值中会略过现有的 master 估算。

minsources

minsources 指令设置在更新本地时钟前需要在源选择算法中被视为可选择的最少源数量。

语法格式为：

```
minsources number-of-sources
```

默认情况下，number-of-sources 是 1。可将 minsources 设置为较大的值来提高可靠性，因为多个源会相互对应。

noclientlog

该指令不使用任何参数，它表明客户访问不会被记录。通常它们会被记录，允许在 chronyc 中使用 client 命令报告统计信息。

reselectdist

当 chronyd 从可用源中选择同步源时，它更喜欢使用最小同步距离的源。然而，为了避免当不同源的距离相似时频繁地重选源，为当前没有被选择的源增加了一个固定的距离。这可通过 reselectdist 选项设置。默认情况下，长度为 100 微秒。

语法格式为：

```
reselectdist dist-in-seconds
```

stratumweight

stratumweight 指令设置当 chronyd 从可用源中选择同步源时每个 stratum 应添加的距离到同步距离。

语法格式为：

```
stratumweight dist-in-seconds
```

默认情况下, dist-seconds 为 1 毫秒。这意味着，stratum 低的源会比 stratum 高的源优先考虑，即使它们的距离更差也是如此。把 stratumweight 设置为 0 时，会导致在选择源时 chronyd 忽略 stratum。

rtcfile

The rtcfile 指令定义文件的名称，其中 chronyd 可以保存与跟踪系统实时时钟(RTC)准确性相关的参数。

语法格式为：

```
rtcfile /var/lib/chrony/rtc
```

当文件退出以及 writetc 命令在 chrony c 中发出时，chrony d 会 保存此文件中的信息。保存的信息是 RTC 在某些时期出现的错误，该时期（从 1970 年 1 月 1 日起秒数）以及 RTC 赢得或丢失时间的速度。并非所有实时时钟都受到支持，因为它们的代码特定于系统。请注意，如果使用这个指令，则不应该手动调整实时时钟，因为这会影响 chrony如果实时时钟偏移是以随机间隔调整的速率。

rtcsync

默认情况下，rtcsync 指令会出现在 /etc/chrony.conf 文件中。这会通知保持系统时钟同步的内核，内核会每 11 分钟更新实时时钟。

18.2.3. 使用 chronyc 的安全性

chronyc 可以通过两种方式访问 chronyd ：

- 互联网协议(IPv4 或 IPv6)
- UNIX 域套接字，可由 root 用户 或 chrony 用户在本地访问。

默认情况下，chronyc 连接到 Unix 域套接字。默认路径为 /var/run/chrony/chronyd.sock。如果这个连接失败，比如 chronyc 在非 root 用户下运行时，chrony c 会尝试连接到 127.0.0.1，然后 ::1。

网络中只允许以下监控命令，它们不会影响 chronyd 的行为：

- activity
- manual list
- rtcdata

- *smoothing*
- *sources*
- *sourcestats*
- *tracking*
- *waitsync*

chronyd 接受这些命令的主机集合可以使用 chronyd 配置文件中的 cmdallow 指令，或者在 chronyc 中使用 cmdallow 命令配置。默认情况下，仅接受来自本地主机（127.0.0.1 或 ::1）的命令。

所有其他命令只能通过 Unix 域套接字进行。当通过网络发送时，chronyd 会返回 Notauthorized 错误，即使它来自 localhost。

使用 chronyc 远程访问 chronyd

1. *通过在 /etc/chrony.conf 文件中添加以下内容来允许 IPv4 和 IPv6 地址的访问：*

   ```
   bindcmdaddress 0.0.0.0
   ```

   或者

   ```
   bindcmdaddress :
   ```

2. *使用 cmdallow 指令允许来自远程 IP 地址、网络或者子网的命令。*

   在 /etc/chrony.conf 文件中添加以下内容：

   ```
   cmdallow 192.168.1.0/24
   ```

3. 在防火墙中打开端口 323 从远程系统连接。

```
~]# firewall-cmd --zone=public --add-port=323/udp
```

如果想永久打开端口 323，使用 --permanent。

```
~]# firewall-cmd --permanent --zone=public --add-port=323/udp
```

请注意, allow 指令是用于 NTP 访问，而 cmdallow 指令用于允许接收远程命令。可以使用本地运行的 chronyc 临时完成这些更改。编辑配置文件以做永久性更改。

18.3. 使用 CHRONY

18.3.1. 安装 chrony

chrony 套件默认安装在 Red Hat Enterprise Linux 7 的一些版本中。如果需要，请以 root 用户身份运行以下命令：

```
~]# yum install chrony
```

chrony 守护进程的默认位置为 /usr/sbin/chronyd。命令行工具将安装到 /usr/bin/chronyc。

18.3.2. 检查 chronyd 的状态

运行以下命令检查 chronyd 的状态：

```
~]$ systemctl status chronyd
chronyd.service - NTP client/server
  Loaded: loaded (/usr/lib/systemd/system/chronyd.service; enabled)
  Active: active (running) since Wed 2013-06-12 22:23:16 CEST; 11h ago
```

18.3.3. 启动 chronyd

要启动 chronyd，使用 root 用户身份运行以下命令：

```
~]# systemctl start chronyd
```

要确保 chronyd 在系统启动时自动启动，以 root 身份运行以下命令：

```
~]# systemctl enable chronyd
```

18.3.4. 停止 chronyd

要停止 chronyd，以 root 身份运行以下命令：

```
~]# systemctl stop chronyd
```

要防止 chronyd 在系统启动时自动启动，以 root 身份运行以下命令：

```
~]# systemctl disable chronyd
```

18.3.5. 检查 chrony 是否同步

要检查是否同步了 chrony，使用tracking、sources 和 sourcestats 命令。

18.3.5.1. 检查 chrony 跟踪

运行以下命令检查 chrony 跟踪：

```
~]$ chronyc tracking
Reference ID  : CB00710F (foo.example.net)
Stratum     : 3
Ref time (UTC) : Fri Jan 27 09:49:17 2017
System time   : 0.000006523 seconds slow of NTP time
Last offset   : -0.000006747 seconds
RMS offset   : 0.000035822 seconds
Frequency    : 3.225 ppm slow
Residual freq  : 0.000 ppm
Skew      : 0.129 ppm
Root delay   : 0.013639022 seconds
Root dispersion : 0.001100737 seconds
Update interval : 64.2 seconds
Leap status   : Normal
```

这些字段如下：

参考 ID

这是当前与计算机同步的服务器的参考 ID 和名称（或 IP）。参考 ID 是一个十六进制数字，用于避免与 IPv4 地址混淆。

Stratum

stratum 代表到带有一个附加的参考时钟的计算机的距离。此类计算机是 stratum-1 计算机，因此示例中的计算机是两个跃点（即 a.b.c 是 stratum-2 并且从 stratum-1 同步）。

ref time

(UTC)是处理参考源中最后一次测量的时间。

System time

在正常操作中，chrony d 从不停止系统时钟，因为时间范围的任何跳转都可能会给某些应用程序程序带来负面影响。相反，系统时钟中的任何错误都会通过稍微加快或减慢系统时钟直到错误被删除，然后返回到系统时钟正常速度来解决。这是因为，系统时钟（如使用 gettimeofday（）系统调用的其他程序读取，或 shell 中的 date 命令）将不同于 chronyd 对当前真实时间的估算值（这在服务器模式中运行时报告 NTP 客户端）。这一行中报告的数值是因此造成的不同。

Last offset

这是最后一次时钟更新中估计的本地误差。

RMS offset

这是位移值的一个长期平均值。

Frequency

"频率"是指如果 chronyd 没有更正系统时钟会出错的速率。它以 ppm（每百万的一部分）表示。例如，1 ppm 的值意味着，当系统的时钟认为它已提前 1 秒时，它实际上已比真实时间增长 1.000001 秒。

Residual freq

这显示了当前所选参考源的"有效频率"。这反映了来自参考源的测量结果表示频率和当前使用频率之间的差别。

原因并不总是为零，即对频率应用平滑过程。每次从参考源进行测量并计算出新的遗留频率时，都会将这一消耗的估计准确性与现有频率值的估算准确性（参见下一个 偏移 数）进行对比。为新频率计算加权平均值，权重取决于这些准确度。如果参考源中的测量结果采用一致的倾向，则持续时间会一直趋向于零。

skew

这是频率上估计的错误绑定。

Root delay

这是到 stratum-1 计算机的网络路径延迟总数，最终与计算机同步。根延迟值以纳秒分辨率打印。在某些极端情况下，这个值可以是负数。（这可能出现在对称对等排列中，计算机的频率不会互相跟踪，并且相对于每台计算机的往返时间而言，网络延迟非常短。）

Root dispersion

这是所有计算机回滚到 stratum-1 计算机的总分散积累，最终与计算机同步。分布是由系统时钟解析、统计测量差异导致的。Root 分散值以纳秒分辨率打印。

Leap status

这是闰秒状态，可以是 Normal、Insert second、Delete second 或 Not syncd。

18.3.5.2. 检查 chrony 源

sources 命令显示 chronyd 正在访问的当前时间源的信息。

可以使用可选参数 -v 来包括详细信息。在这种情况下，会输出额外的标头行显示字段含义的信息。

```
~]$ chronyc sources
 210 Number of sources = 3
MS Name/IP address     Stratum Poll Reach LastRx Last sample
===============================================================================

#* GPS0           0  4  377  11  -479ns[ -621ns] +/- 134ns
^? a.b.c         2  6  377  23  -923us[ -924us] +/-  43ms
^+ d.e.f         1  6  377  21 -2629us[-2619us] +/-  86ms
```

这些列如下：

M

这表示源的模式。^ 表示服务器，= 表示对等，# 代表本地连接的参考时钟。

S

此列指示源的状态。"*"表示当前同步的 chronyd 的源。"+"表示可接受的源与所选源结合使用。"-"表示合并算法排除的可接受的源。"?"表示丢失了哪个连接或者数据包没有通过所有测试的源。"x" " 表示 chronyd 认为是 假勾号（ 时间与大多数其他来源不一致）。"~"表示时间似乎有太多变化的来源。"?" 条件也会在启动时显示，直到从中收集了至少三个样本。

名称/IP 地址

此时会显示源的名称或 IP 地址，或者参考时钟的引用 ID。

Stratum

这显示了源的 stratum，如其最近收到的示例中所报告。Stratum 1 代表有本地附加参考时钟的计算机。与 stratum 1 计算机同步的计算机处于 stratum 2。与 stratum 2 计算机同步的计算机处于 stratum 3，以此类推。

Poll

显示源轮询率（以 2 为底数进行计算，以秒为单位）。因此，数值 6 表示每 64 秒就进行一次测量。

chronyd 会自动根据需要条件来改变轮询率。

Reach

这将显示源的 reach 寄存器，以一个十六进制数字代表。寄存器有 8 位，并在每次从源接收或错过的数据包时进行更新。377 表示在最后的 8 个数据传输中都收到有效回复。

LastRx

这一列显示了在多久前从源中获取了最后的样本。这通常以秒为单位。字母 m、h、d 和 y 代表分钟、小时、天和年。值为 10 年表示尚未从该源收到任何样本。

Last sample

这一列显示本地时钟和最后一个测量源之间的偏差。方括号中的数字显示了实际测量的误差。这可以使用 ns（代表 nanoseconds）、us（代表 microseconds）、ms（代表 milliseconds）或 s（代表秒）后缀。方括号左边的数字显示了原来的测量，经过调整以允许应用本地时钟。+/- 指示符后的数字显示了测量中的错误裕度。正偏差表示本地时钟在源前面。

18.3.5.3. 检查 chrony 源统计信息

sourcestats 命令显示目前被 chronyd 检查的每个源的偏移率和误差估算过程的信息。

可以使用可选参数 -v 来包括详细信息。在这种情况下，会输出额外的标头行显示字段含义的信息。

```
~]$ chronyc sourcestats
210 Number of sources = 1
Name/IP Address      NP NR Span Frequency Freq Skew Offset Std Dev
==============================================================================

abc.def.ghi        11  5  46m   -0.001   0.045   1us  25us
```

这些列如下：

名称/IP 地址

这是 NTP 服务器（或 peer）的名称或 IP 地址，或者引用其他行中其它部分相关参考时钟的 ID。

NP

这是服务器当前保留的样本点数。偏移率和当前偏移是通过在这些点上进行线性回退来估算的。

NR

这是在上一次回滚返回后运行具有相同符号的驻留的数量。如果这个数值与样本数量相比变得太小了，意味着直线不再适合于数据。如果数量太低，chronyd 会丢弃旧的样本并重新运行回滚直到运行次数变得可以接受。

Span

这是最旧和最新样本之间的间隔。如果没有单位显示，则该数值以秒为单位。在这个示例中，间隔为 46 分钟。

Frequency

这是服务器估计的遗留频率，以每百万个部分为单位。在这种情况下，相比服务器，计算机的时钟预计运行速度为 10^9 个部分。

freq Skew

这是 Freq 上估计的错误绑定（以每百万分页表示）。

Offset

这是源估计的误差。

Std Dev

这是估计的示例标准偏差。

18.3.6. 手动调整系统时钟

要立即调整系统时钟，绕过单机进行的任何调整，以 root 身份运行以下命令：

```
~]# chronyc makestep
```

如果使用了 rtcfile 指令，则不应该手动调整实时时钟。随机调整会影响 chrony 测量实时时钟偏移速度的需求。

18.4. 为不同的环境设置 CHRONY

18.4.1. 在隔离网络中为系统设置 chrony

对于从来不连接到互联网的网络来说，一台计算机被选为主计时服务器。其他计算机要么是主计算机的直接客户端，要么是客户端的客户端。在 master 上，必须使用系统时钟的平均偏移率手动设置 drift 文件。如果 master 被重启，它将从周围的系统获得时间并计算设定系统时钟的平均值。之后它会恢复基于 drift 文件的调整。当使用 settime 命令时会自动更新 drift 文件。

在选择成为 master 的系统上，以 root 用户身份运行一个文本编辑器来编辑 /etc/chrony.conf，如下所示：

```
driftfile /var/lib/chrony/drift
commandkey 1
keyfile /etc/chrony.keys
initstepslew 10 client1 client3 client6
local stratum 8
manual
allow 192.0.2.0
```

其中 192.0.2.0 是允许客户端连接的网络或者子网地址。

在选择作为 master 客户端的系统上，以 root 用户身份运行一个文本编辑器来编辑 /etc/chrony.conf，如下所示：

```
server master
driftfile /var/lib/chrony/drift
logdir /var/log/chrony
log measurements statistics tracking
keyfile /etc/chrony.keys
commandkey 24
local stratum 10
initstepslew 20 master
allow 192.0.2.123
```

其中 192.0.2.123 是 master 的地址， master 是 master 的主机名。带有此配置的客户端如果重启 master 将重新同步。

在不是 master 客户端的客户端系统中，/etc/chrony.conf 文件应当相同，除了应该省略 local 和 allow 指令。

在隔离网络中，您还可以使用 local 指令来启用本地参考模式，允许 chronyd 作为 NTP 服务器实时显示同步，即使它从未同步或者最后一次更新时钟早前发生。

要允许网络中的多个服务器使用相同的本地配置并相互同步，而不让客户端轮询多个服务器，请使用 local 指令的 orphan 选项启用孤立模式。每一服务器需要配置为通过 本地 轮询所有其他服务器。这样可确保只有具有最小参考 ID 的服务器具有本地参考活跃状态，并同步其他服务器。当服务器出现故障时，另一台服务器将接管。

18.5. 使用 CHRONYC

18.5.1. 使用 chronyc 控制 chronyd

要在互动模式中使用命令行工具 chronyc 来更改本地 chronyd 实例，以根用户身份输入以下命令：

```
~]# chronyc
```

如果要使用某些受限命令，chronyc 需要以 root 运行。

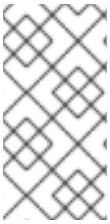
chronyc 命令提示符如下所示：

```
chronyc>
```

您可以键入 help 来列出所有命令。

如果与以下命令一同调用，工具也可以在非互动命令模式下调用：

```
chronyc command
```



注意

使用 chronyc 所做的更改不具有持久性，它们会在 chronyd 重启后丢失。要使更改有持久性，修改 /etc/chrony.conf。

18.6. 带有 HW 时间戳的 CHRONY

18.6.1. 了解硬件时间戳

硬件时间戳是一些网络接口控制器(NIC)支持的一项功能，它提供传入和传出数据包的准确时间戳。NTP 时间戳通常由内核及使用系统时钟的 chronyd 创建。但是，当启用 HW 时间戳时，NIC 使用自己的时钟在数据包进入或离开链路层或物理层时生成时间戳。与 NTP 一起使用时，硬件时间戳可以显著提高同步的准确性。为了获得最佳准确性，NTP 服务器和 NTP 客户端都需要使用硬件时间戳。在理想条件下，可能还会有次微秒的准确性。

另一个用于使用硬件时间戳进行时间同步的协议是 PTP有关 PTP 的详情请参考 第 20 章 使用 ptp4l 配置 PTP。与 NTP 不同，PTP 依赖于网络交换机和路由器。如果您想要达到同步的最佳准确性，请在带有 PTP 支持的网络中使用 PTP，在使用不支持这个协议的交换机和路由器的网络上选择 NTP。

18.6.2. 验证硬件时间戳支持

要验证接口是否支持使用 NTP 的硬件时间戳，请使用 ethtool -T 命令。如果 ethtool 列出了 SOF_TIMESTAMPING_TX_HARDWARE 和 SOF_TIMESTAMPING_TX_SOFTWARE 模式，以及 HWTSTAMP_FILTER_ ALL 过滤器模式，则可以使用硬件时间戳的 NTP。

例 18.2. 在特定界面上验证硬件时间戳支持

```
~]# ethtool -T eth0
```

输出：

```
Timestamping parameters for eth0:
Capabilities:
    hardware-transmit   (SOF_TIMESTAMPING_TX_HARDWARE)
    software-transmit   (SOF_TIMESTAMPING_TX_SOFTWARE)
    hardware-receive   (SOF_TIMESTAMPING_RX_HARDWARE)
    software-receive   (SOF_TIMESTAMPING_RX_SOFTWARE)
    software-system-clock (SOF_TIMESTAMPING_SOFTWARE)
    hardware-raw-clock  (SOF_TIMESTAMPING_RAW_HARDWARE)
PTP Hardware Clock: 0
Hardware Transmit Timestamp Modes:
    off         (HWTSTAMP_TX_OFF)
    on          (HWTSTAMP_TX_ON)
Hardware Receive Filter Modes:
    none        (HWTSTAMP_FILTER_NONE)
    all         (HWTSTAMP_FILTER_ALL)
    ptpv1-l4-sync     (HWTSTAMP_FILTER_PTP_V1_L4_SYNC)
    ptpv1-l4-delay-req  (HWTSTAMP_FILTER_PTP_V1_L4_DELAY_REQ)
    ptpv2-l4-sync     (HWTSTAMP_FILTER_PTP_V2_L4_SYNC)
    ptpv2-l4-delay-req  (HWTSTAMP_FILTER_PTP_V2_L4_DELAY_REQ)
    ptpv2-l2-sync     (HWTSTAMP_FILTER_PTP_V2_L2_SYNC)
```

```
ptpv2-l2-delay-req  (HWTSTAMP_FILTER_PTP_V2_L2_DELAY_REQ)
ptpv2-event     (HWTSTAMP_FILTER_PTP_V2_EVENT)
ptpv2-sync      (HWTSTAMP_FILTER_PTP_V2_SYNC)
ptpv2-delay-req    (HWTSTAMP_FILTER_PTP_V2_DELAY_REQ)
```

18.6.3. 启用硬件时间戳

要启用硬件时间戳，请使用 /etc/chrony.conf 文件中的 hwtimestamp 指令。该指令可以指定单个接口，也可以使用通配符()在支持它的所有接口上启用硬件时间戳。如果 linuxptp 软件包 中没有其它应用程序（如 [application]*ptp4l ）在接口中使用硬件时间戳，请使用通配符规格。在 chrony 配置文件中允许使用多个 hwtimestamp 指令。

例 18.3. 使用 hwtimestamp 指令启用硬件时间戳

```
hwtimestamp eth0
hwtimestamp eth1
hwtimestamp *
```

18.6.4. 配置客户端轮询间隔

建议为互联网中的服务器使用默认的轮询间隔范围（64-1024秒）。对于本地服务器和硬件时间戳，需要配置一个较短的轮询间隔，以便最小化系统时钟偏差。

/etc/chrony.conf 中的以下指令使用一秒轮询间隔指定本地 NTP 服务器：

```
server ntp.local minpoll 0 maxpoll 0
```

18.6.5. 启用交集模式

不是硬件 NTP 设备，但一般情况下运行软件 NTP 实施的计算机（如 chrony ）的 NTP 服务器只有在发送数据包后才会获得硬件传输时间戳。此行为可防止服务器在它对应的数据包中保存时间戳。要启用 NTP 客户端接收传输时间戳后生成的传输时间戳，请将客户端配置为使用 NTP 交集模式，方法是在 /etc/chrony.conf 的 server 指令中添加 xleave 选项：

```
server ntp.local minpoll 0 maxpoll 0 xleave
```

18.6.6. 为大量客户端配置服务器

默认服务器配置允许几千个客户端同时使用交集模式。要为更多的客户端配置服务器，增大 /etc/chrony.conf 中的 clientloglimit 指令。这个指令指定为服务器中客户端访问的日志分配的最大内存

大小：

```
clientloglimit 100000000
```

18.6.7. 验证硬件时间戳

要校验该接口是否已成功启用了硬件时间戳，请检查系统日志。日志应包含来自 chronyd 的每个接口的消息，并成功启用硬件时间戳。

例 18.4. 带有 Enabled Hardware Timestamping 的接口的日志消息

```
chronyd[4081]: Enabled HW timestamping on eth0
chronyd[4081]: Enabled HW timestamping on eth1
```

当 chronyd 配置为 NTP 客户端或对等时，您可以有传输和接收时间戳模式，以及 chronyc ntpdata 命令为每个 NTP 源报告交集模式：

例 18.5. 报告每个 NTP 源的转换、接收时间戳和交集模式

```
~]# chronyc ntpdata
```

输出：

```
Remote address : 203.0.113.15 (CB00710F)
Remote port   : 123
Local address  : 203.0.113.74 (CB00714A)
Leap status   : Normal
Version     : 4
Mode      : Server
Stratum     : 1
Poll interval  : 0 (1 seconds)
Precision    : -24 (0.000000060 seconds)
Root delay   : 0.000015 seconds
Root dispersion : 0.000015 seconds
Reference ID  : 47505300 (GPS)
Reference time : Wed May 03 13:47:45 2017
Offset     : -0.000000134 seconds
Peer delay   : 0.000005396 seconds
Peer dispersion : 0.000002329 seconds
Response time  : 0.000152073 seconds
Jitter asymmetry: +0.00
NTP tests    : 111 111 1111
Interleaved   : Yes
Authenticated  : No
TX timestamping : Hardware
```

```
RX timestamping : Hardware
Total TX    : 27
Total RX    : 27
Total valid RX : 27
```

例 18.6. 报告 NTP 测量的稳定性

```
# chronyc sourcestats
```

启用硬件时间戳后，NTP 测量的稳定性应该以十秒或数百纳秒为单位，处于正常负载下。此稳定性会在 chronyc sourcestats 命令的输出结果的 Std Dev 列中报告：

输出：

```
210 Number of sources = 1
Name/IP Address      NP NR Span Frequency Freq Skew Offset Std Dev
ntp.local        12  7  11   +0.000   0.019   +0ns  49ns
```

18.6.8. 配置 PTP-NTP 桥接

如果网络中存在一个高度准确的 Precision Time Protocol (PTP) grandmaster，但没有支持 PTP 支持的交换机或路由器，则一个计算机会可能会被指定专门用于作为 PTP slave 和一个 stratum-1 NTP 服务器。此类计算机需要两个或多个网络接口，并且与降电员接近或与之直接连接。这样可保证高度准确的网络同步。

配置 linuxptp 软件包中的 ptp4l 和 phc2sys 程序，以使用 PTP 同步系统时钟。该配置在 第 20 章 使用 ptp4l 配置 PTP 中描述。配置 chronyd 以使用其他接口提供系统时间：

例 18.7. 配置 chronyd 以使用其他接口提供系统时间

```
bindaddress 203.0.113.74
hwtimestamp eth1
local stratum 1
```

18.7. 其它资源

以下信息来源提供了有关 chrony 的其他资源。

18.7.1. 安装的文档

- chronyc(1) man page - 描述 chronyc 命令行界面工具，包括命令和命令选项。
- chronyd(8)man page - 描述 chronyd 守护进程，包括命令和命令选项。
- chrony.conf(5) man page - 描述 chrony 配置文件。

18.7.2. 在线文档

- http://chrony.tuxfamily.org/doc/3.1/chronyc.html
- http://chrony.tuxfamily.org/doc/3.1/chronyd.html
- http://chrony.tuxfamily.org/doc/3.1/chrony.conf.html

常见问题解答请参考 http://chrony.tuxfamily.org/faq.html

第 19 章 使用 NTPD 配置 NTP

19.1. NTP 简介

网络时间协议 (NTP)可以准确显示时间和日期信息，以便让联网计算机系统上的时间时钟与网络或互联网上的常见引用保持同步。世界各地的许多标准正文都有原子时钟，这些时钟可以作为参考提供。组成全球定位系统的卫星包含多个原子时钟，使其时间信号可能非常准确。出于军事原因，这些信号可以有意降级。理想的情况是，每个站点都有一个服务器（附加有自己的参考时钟）作为站点范围内的时间服务器。许多通过低频率放射或全局定位系统(GPS)获取时间和日期的设备都存在。然而，在大多数情况下，可以使用在地理分散位置连接到互联网的一系列公开访问时间服务器。这些 NTP 服务器提供"统一通用时间"(UTC)。有关这些时间服务器的信息，请访问

因为许多原因，保持准确的时间非常重要。例如在网络中，需要准确的数据包和日志的时间戳。日志用于调查服务和安全问题，因此不同系统上的时间戳必须由同步时钟变为实际值。随着系统和网络变得越来越快，对时钟的相应需求会更加准确和解决。在某些国家，有保持准确同步时钟的法律责任。如需更多信息，请参阅 www.ntp.org。在 Linux 系统中，NTP 由在用户空间运行的守护进程实施。Red Hat Enterprise Linux 7 中的默认 NTP 用户空间守护进程是 chronyd。如果要使用 ntpd 守护进程，则必须禁用它。有关 chrony 的详情请查看 第 18 章 使用 chrony 套件配置 NTP。

用户空间守护进程更新系统时钟，这是在内核中运行的软件时钟。Linux使用软件时钟作为系统时钟更好地解析，优于称为"实时时钟" (RTC) 的典型嵌入式硬件时钟。有关硬件时钟的详情，请查看 rtc(4) 和 hwclock(8) man page。系统时钟可以通过使用不同的时钟源来维护系统的时间。通常, 使用 时间戳计数器（TSC）。TSC 是一个 CPU 寄存器，它计算从上次重置的循环数。它非常快速，分辨率很高，而且没有任何中断。在系统启动时，系统时钟会读取 RTC 的时间和日期。由于温度变化，RTC 保持的时间将从实际时间下降至每月 5 分钟。因此，系统时钟需要与外部时间引用持续同步。当系统时钟由 ntpd 同步时，内核将自动每 11 分钟更新一次 RTC。

19.2. NTP STRATA

NTP 服务器按照其与作为时间信号来源的原子时钟的同步距离来分类。服务器被视为按层或层次排列，从上到 15 个层进行排列。因此，术语 stratum 在引用特定层时使用。Atomic 时钟被称为 Stratum 0，因为这是来源，但 Internet 上没有发送 Stratum 0 数据包，所有 stratum 0 原子时钟都附加到服务器（称为 stratum 1）。这些服务器发送标记为 Stratum 1 的数据包。通过标记为 stratum n 的数据包同步的服务器，它属于下一个低级和 stratum，并将其数据包标记为 stratum n+1。同一级别的服务器可以相互交换数据包，但仍指定为仅属于一个级别，级别位于它们同步的最佳参考下。Stratum 16 的名称用于表示该服务器目前未同步到可靠的时间源。

请注意，默认情况下 NTP 客户端充当下面的 stratum 中这些系统的服务器。

以下是 NTP Strata 概述：

Stratum 0

Atomic Clocks 及其信号通过 Radio 和 GPT 广播

- 北欧（绿色定位系统）
- 移动电话系统
- Low Frequency Radio 广播 WWVB(Colorado, USA.)、JJY-40 和 JJY-60（日本）、DCF77(Germany)和 MSF（英国）

  这些信号可以由专用设备接收，通常由 RS-232 连接到用作组织或站点时间服务器的系统。

Stratum 1

附加了无线时钟、LVM 时钟或 atomic 时钟的计算机

Stratum 2

从级别 1 读取；确定到较低级别

Stratum 3

从 stratum 2 中读取；已过渡到较低级别

Stratum n+1

从 stratum n 中读取；选择至较低级别

Stratum 15

从级别 14 中进行读取；这是最低级别。

这个过程会下降到 stratum 15，这是最低的有效级别。标签 Stratum 16 用于表示一个未同步状态。

19.3. 了解 NTP

Red Hat Enterprise Linux 使用的 NTP 版本如 RFC 1305 网络时间协议(Version 3)规范、实施和分析以及 RFC 5905 网络时间协议版本 4：协议和算法规范中所述

这种 NTP 的实施可实现子秒的准确性。在互联网上，10 毫秒的准确性是正常现象。在局域网(LAN)上，可在理想条件下使用 1 ms 准确度。这是因为，现在对时钟偏移进行了记帐和纠正，而早期更简单的时间协议系统没有这么做。使用 64 位时间戳提供 233 picoseconds 分辨率。时间戳的前 32 位用于秒，最后 32 位用于秒数。

NTP 表示时间为从 00:00(midnight)1 起的秒数，1900 GMT。由于 32 位用于计算秒数，这意味着 2036 年将"滚动"。但是 NTP 工作于时间戳之间的差别，因此这不会产生与其它时间协议实施相同的问题级别。如果引导时正确时间为 68 年的硬件时钟可用，NTP 将正确解释当前日期。NTP4 规范提供了"Era Number(Era Offset)"和"Era Offset"，可用于在处理超过 68 年的时间长度时使软件更加稳定。不要将其与 Unix Year 2038 问题混淆。

NTP 协议提供了其他信息以提高准确性。使用四个时间戳来计算往返时间和服务器响应时间：为了使其角色中的系统作为 NTP 客户端与参考时间服务器同步，需要使用"原始时间戳"发送数据包。当数据包到达时，时间服务器会添加一个"接收时间戳"。在处理时间和日期信息请求后，又刚刚在返回数据包之前，它将添加一个"传输时间戳"。当返回的数据包到达 NTP 客户端时，将生成"接收时间戳"。客户现在可以计算总往返用时，并通过缩短处理时间而缩短实际旅行时间。假设传出和返回行程用时相同，即可计算接收 NTP 数据时的单条延迟。完整的 NTP 算法比这里介绍的要复杂得多。

收到包含时间信息的数据包后，它不会立即响应，而是首先接受验证检查，然后与几个其他时间样本一起处理，以达到预计时间。然后，与系统时钟进行比较以确定时间偏移、系统时钟的时间与 ntpd 决定时间应该的差值。系统时钟调整缓慢，其中大多数速度为每秒 0.5 毫秒，以通过更改计数器使用的频率来减少这一偏差。使用此方法将时钟调整为 1 秒至少需要 2000 秒。这种缓慢的更改被称为 slewing 且无法向后移动。如果时钟的时间偏移超过 128毫秒（默认设置），ntpd 可以"步骤"时钟正向或后退。如果系统启动的时间偏移大于 1000 秒，则用户或安装脚本应手动调整。请参阅 第 3 章 配置日期和时间。使用 ntpd 命令的 -g 选项（默认为使用），系统启动时的任何偏移都将被修正，但在正常操作中，最多 1000 秒的偏移才会得到更正。

如果时间向后更改，某些软件可能会出现故障或生成错误。对于对时间步骤变化敏感的系统，可使用 -x 选项（与 - g 选项无关），将阈值更改为 600 s，而非 128 m。使用 -x 选项将步骤限值从 0.128 增加到 600 存在缺点，因为必须使用不同的方法来控制时钟。它禁用了内核时钟，并可能会对时钟准确性造成负面影响。x 选项 可以添加到 /etc/sysconfig/ntpd 配置文件中。

19.4. 了解偏移文件

偏移文件用于存储系统时钟以其频率运行的间隔和 UTC 同步所需的频率之间的频率偏移。如果存在，偏移文件中所包含的值会在系统启动时读取，用于更正时钟源。使用偏移文件可缩短实现稳定准确时间所需的时间。计算值，并且偏移文件每小时一次被 ntpd 取代。已替换 drift 文件，而不仅仅是更新，因此偏移文件必须位于 ntpd 具有 写入权限的目录中。

19.5. UTC、时区和 DST

由于 NTP 完全在 UTC（统一时间、协调时间）中，时区和 DST（节省时间）由系统本地应用。文件

/etc/localtime 是 /usr/share/zoneinfo 中的区域信息文件的副本或符号链接。RTC 可能处于本地时间或 UTC 中，如 /etc/adjtime 的第三行所指定，该行将是 LOCAL 或 UTC 之一，用于指示 RTC 时钟的设置方式。用户可以通过在 日期和时间 图形配置工具中的" 系统时钟使用 UTC"复选框来 轻松更改此设置。有关如何使用该工具的详情，请查看 第 3 章 配置日期和时间。建议在 UTC 中运行 RTC，以避免在夏时制更改时出现各种问题。

ntpd 的操作在 ntpd (8) 的 man page 中进行了更详细的说明。resources 部分列出了有用的信息来源。请参阅 第 19.20 节 “其它资源”。

19.6. NTP 验证选项

NTPv4 NTPv4 添加了对自动密钥安全架构的支持，该架构基于公共非对称加密，同时保留对称密钥加密的支持。 RFC 5906 网络时间协议版本 4 中描述了 Autokey 协议：自动密钥规范。不幸的是，之后发现该协议存在严重的安全问题，因此红帽强烈推荐使用对称密钥。ntp_auth(5) man page 描述了 ntpd 的身份验证选项和命令。

网络上的攻击者可以通过发送带有不正确时间信息的 NTP 数据包来尝试中断服务。在使用 NTP 服务器公共池的系统上，通过 /etc/ntp.conf 中的公共 NTP 服务器列表中有超过三个 NTP 服务器，可以降低风险。如果只有一个时间源被入侵或欺骗，ntp d 将忽略该源。您应该进行风险评估，并考虑错误时间对应用程序和组织的影响。如果您有内部时间源，您应该考虑保护 NTP 数据包所分发的网络的步骤。如果您进行风险评估并总结风险可以接受，并且对应用程序的影响降至最低，那么您可以选择不使用身份验证。

默认情况下，广播和多播模式需要进行身份验证。如果您决定信任网络，那么您可以通过在 ntp.conf 文件中 禁用 auth 指令来禁用 身份验证。或者，身份验证需要使用 SHA1 或 MD5 对称密钥进行配置，或者通过使用 Autokey 方案配置公钥（对称）密钥加密。ntp_auth(8)man page 中解释了非对称加密的自动密钥方案，并在 ntp-keygen(8)中解释了密钥的生成。要实施对称密钥加密，请参阅 第 19.17.12 节 “使用密钥配置对称身份验证” 了解 密钥 选项的说明。

19.7. 管理虚拟机上的时间

虚拟机无法访问实际的硬件时钟，虚拟时钟不稳定，因为稳定性依赖于主机系统的工作负载。因此，使用中的虚拟化应用程序应当提供半虚拟化时钟。在带有 KVM 的红帽企业 Linux 上，默认时钟源为 kvm-clock。请参阅《 红帽企业 Linux 7 虚拟化部署和管理指南 》中的 KVM 客户机计时管理 章节。

19.8. 了解 LEAP 秒

Greenwich Mean Time(GMT)是通过测量 Solaris 日来派生的，它依赖于地球的旋转。第一次生成原子时钟时，可能实现更精确的时间定义。1958 年，国际原子时间(TAI)是基于更加准确、非常稳定的原子时钟而引入的。同时还引进了更加准确的通用时间 1(UT1)，以取代 GMT。原子时钟实际上比地球旋转更加稳定，因此两次开始偏离。因此，UTC 作为一个实际措施被引入。它保持在 UT1 的一秒之内，但为了避免进行很多小的细微调整，决定引入一个跃点数第二的概念，以便以可管理的方式调节差异。UT1 和

UTC 之间的差别受到监控，直到它们偏离了一半以上的时间。然后，仅需要引入一秒钟调整，即正向或向后调整。由于人类旋转速度的异常性质，对调整的需求在未来无法预测。调整时间的决策由 国际地球转站和参考系统服务(IERS) 做出。但是，这些公告只对 Stratum 1 服务器的管理员很重要，因为 NTP 会传输有关待处理 leap 秒的信息并自动应用它们。

19.9. 了解 NTPD 配置文件

守护进程 ntpd 在系统启动时或 服务重启时读取配置文件。文件的默认位置为 /etc/ntp.conf，您可以输入以下命令来查看该文件：

```
~]$ less /etc/ntp.conf
```

本章稍后将简要阐述配置命令，请参阅 第 19.17 节 “配置 NTP” 并在 ntp.conf(5) man page 中更详细地阐述。

以下是默认配置文件内容的简短说明：

driftfile 条目

指定了到 drift 文件的路径，Red Hat Enterprise Linux 中的默认条目为：

```
driftfile /var/lib/ntp/drift
```

如果您对此进行更改，请确保目录可由 ntpd 写入。文件包含一个值，用于在每次系统或服务启动后调整系统时钟频率。如需更多信息，请参阅了解偏移文件。

访问控制条目

以下行设置默认访问控制限制：

```
restrict default nomodify notrap nopeer noquery
```

- nomodify 选项可防止对配置进行任何更改。
- notrap 选项可防止 ntpdc 控制 消息协议陷阱。
- nopeer 选项可防止形成同级关联。

- noquery 选项可防止回答 ntpq 和 ntpdc 查询，而不是时间查询。

重要

ntpq 和 ntpdc 查询可用于放大攻击，因此请勿在可公开访问的系统上从 restrict 默认 命令中删除 noquery 选项。

详情请查看 CVE-2013-5211。

不同进程或应用 有时需要范围 127.0.0.0/8 范围内的地址。由于上面的"限制默认"行阻止对未明确允许的所有内容的访问，因此允许通过以下行访问 IPv4 和 IPv6 的标准回环地址：

```
# the administrative functions.
restrict 127.0.0.1
restrict ::1
```

如果另一个应用程序有特别要求，可以在下面添加地址。

由于上面的"restrict default"行，不允许本地网络上的主机。要更改此值，例如，允许 192.0.2.0/24 网络中的主机查询时间和统计信息，但不需要再使用以下格式的行：

```
restrict 192.0.2.0 mask 255.255.255.0 nomodify notrap nopeer
```

要允许特定主机（如 192.0.2.250/32 ）进行不受限制的访问，需要以下格式的行：

```
restrict 192.0.2.250
```

如果未指定，则将应用 255.255.255.255 掩码。

ntp_acc(5) 手册页面中解释了 restrict 命令。

公共服务器条目

默认情况下，ntp .conf 文件包含四个公共服务器条目：

```
server 0.rhel.pool.ntp.org iburst
server 1.rhel.pool.ntp.org iburst
server 2.rhel.pool.ntp.org iburst
server 3.rhel.pool.ntp.org iburst
```

广播多播服务器条目

默认情况下，ntp .conf 文件包含一些注释掉的示例。这些都是自我解释的。有关特定命令的说明，请参阅 第 19.17 节 “配置 NTP”。如果需要，在示例中添加命令。

注意

当 DHCP 客户端程序, dhclient 收到来自 DHCP 服务器的 NTP 服务器列表时，它会将它们添加到 ntp.conf 并重新启动服务。要禁用此功能，请将 PEERNTP=no 添加到 /etc/sysconfig/network。

19.10. 了解 NTPD SYSCONFIG 文件

在 service start 上，该文件将由 ntpd init 脚本读取。默认内容如下：

```
# Command line options for ntpd
OPTIONS="-g"
```

g 选项可让 ntpd 忽略偏移限制 1000 s，并尝试同步时间（即使偏移大于 1000，但仅在系统启动时）。如果没有该选项 ntpd，如果时间偏移大于 1000，则退出。如果服务重启并且偏移大于 1000 s，即使使用 -g 选项，它也将在系统启动后退出。

19.11. 禁用 CHRONY

要使用 ntpd 默认用户空间守护进程，必须停止并禁用 chronyd。以 root 用户身份运行以下命令：

```
~]# systemctl stop chronyd
```

要防止它在系统启动时重启，以 root 用户身份运行以下命令：

```
~]# systemctl disable chronyd
```

运行以下命令检查 chronyd 的状态：

```
~]$ systemctl status chronyd
```

19.12. 检查 NTP 守护进程是否已安装

要检查是否安装了 ntpd，以 root 用户身份输入以下命令：

```
~]# yum install ntp
```

NTP 通过守护进程或服务 ntpd 来实施，后者包含在 ntp 软件包中。

19.13. 安装 NTP 守护进程(NTPD)

要安装 ntpd，以 root 用户身份输入以下命令：

```
~]# yum install ntp
```

要在系统启动时启用 ntpd，以 root 用户身份输入以下命令：

```
~]# systemctl enable ntpd
```

19.14. 检查 NTP 的状态

要检查 ntpd 是否在系统启动时运行并配置为在系统启动时运行，请运行以下命令：

```
~]$ systemctl status ntpd
```

要从 ntpd 获取简短状态报告，请运行以下命令：

```
~]$ ntpstat
unsynchronised
 time server re-starting
  polling server every 64 s
```

```
~]$ ntpstat
synchronised to NTP server (10.5.26.10) at stratum 2
  time correct to within 52 ms
  polling server every 1024 s
```

19.15. 将防火墙配置为允许传入的 NTP 数据包

NTP 流量由端口 123 上的 UDP 数据包组成，并且需要通过基于网络和主机的防火墙来允许 NTP 运行。

使用图形防火墙 配置 工具，检查防火墙是否已配置为允许客户端的传入 NTP 流量。

要启动图形化 firewall-config 工具，请按 Super 键进入"活动概览"，键入 firewall，然后按 Enter 键。此时将打开 防火墙配置 窗口。系统将提示您输入用户密码。

要使用命令行启动图形防火墙配置工具，以 root 用户身份输入以下命令：

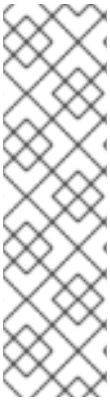
```
~]# firewall-config
```

此时将打开防火墙配置窗口。请注意，该命令可以以普通用户身份运行，但随后会提示您输入 root 密码。

查找左下角的"连接"。这表明 firewall-config 工具已连接到用户空间守护进程 firewalld。

19.15.1. 更改防火墙设置

要立即更改当前的防火墙设置，请确保标记为 Configuration 的下拉菜单已设置为 Runtime。或者，若要编辑要在下一次系统启动或重新加载时应用的设置，请从下拉列表中选择 永久。



注意

在 Runtime 模式中更改防火墙设置时，当设置或清除与服务关联的复选框时，您的选择将立即生效。在操作可能供其他用户使用的系统时，您应该牢记这一点。

以 永久 模式更改防火墙设置时，您的选择仅在重新加载防火墙或系统重新启动时生效。要重新加载防火墙，请选择 Options 菜单并选择 Reload Firewall。

19.15.2. 在 NTP Packet 的防火墙中打开端口

要允许通过防火墙到特定端口的流量，请启动 firewall-config 工具并选择要更改的网络区。选择 端口

选项卡，然后单击 添加 按钮。此时会打开 端口和协议 窗口。

输入端口号 123，然后从下拉列表中选择 udp。

19.16. 配置 NTPDATE 服务器

ntpdate 服务的目的是在系统引导期间设置时钟。之前，这用于确保 ntpdate 之后启动的服务具有正确的时间并且不观察时钟中的跳过。ntpdate 和 step-tickers 列表被视为已弃用，因此 Red Hat Enterprise Linux 7 在 ntpd 命令中使用 -g 选项，而不是默认 ntpdate。

如果在没有 ntpd 服务的情况下或为 ntp d 命令指定了 -x 选项，则 Red Hat Enterprise Linux 7 中的 ntp date 服务 很有用。如果 ntpd 与 -x 一起使用，但没有启用 ntpdate 服务，则只有在时间差大于 600 秒时才会通过步骤纠正时钟。如果误差小于 600 秒，时钟调整缓慢，每更正的秒约为 2000 秒。

要检查是否启用了 ntpdate 服务在系统启动时运行，请运行以下命令：

```
~]$ systemctl status ntpdate
```

要启用服务在系统启动时运行，以 root 用户身份运行以下命令：

```
~]# systemctl enable ntpdate
```

在 Red Hat Enterprise Linux 7 中，默认的 /etc/ntp/step-tickers 文件包含 0.rhel.pool.ntp.org。要配置额外的 ntpdate 服务器，请使用以 root 用户身份运行的文本编辑器来编辑 /etc/ntp/step-tickers。列出的服务器数量并不重要，因为 ntpdate 将仅使用它来获取系统启动时的日期信息。如果您有内部时间服务器，则第一行使用该主机名。第二行中作为备份的另一个主机是明智的。选择备份服务器以及第二主机是否为内部还是外部取决于您的风险评估。例如，任何问题影响第一台服务器也影响第二台服务器的几率如何？在网络中断对第一服务器的访问时，与外部服务器的连接是否比内部服务器连接更加可能？

19.17. 配置 NTP

要更改 NTP 服务的默认配置，请使用以 root 用户身份运行的文本编辑器来编辑 /etc/ntp.conf 文件。此文件与 ntpd 一同安装，默认配置为使用红帽池中的时间服务器。man page ntp.conf(5) 描述了除访问和速率限制命令之外的配置文件中可以使用的命令选项，这些选项在 ntp_acc(5) 手册页中进行了说明。

19.17.1. 配置 NTP 服务的访问权限控制

要限制或控制对系统上运行的 NTP 服务的访问，请使用 ntp.conf 文件中的 restrict 命令。请参阅注

释掉示例：

```
# Hosts on local network are less restricted.
#restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap
```

restrict 命令采用以下格式：

```
restrict address [mask mask] option
```

其中地址和 掩码指定您要应用限制的 IP 地址，选项是以下一个或多个 ：

- *忽略 - 所有封包都将被忽略，包括 ntpq 和 ntpdc 查询。*
- *Kod - 将发送"Kiss-o'-death"数据包，以减少不需要的查询。*
- *有限 - 如果数据包违反速率限制默认值或 discard 命令指定的值，则不要响应时间服务请求。ntpq 和 ntpdc 查询不受影响。有关 discard 命令和默认值的详情请参考 第 19.17.2 节 "配置对 NTP 服务的访问权限等级限制"。*
- *lowpriotrap - 匹配主机设置的陷阱低优先级。*
- *nomodify - 阻止对配置进行任何更改。*
- *noquery - 防止回答 ntpq 和 ntpdc 查询，但不提供时间查询。*
- *nopeer - 阻止形成同级关联.*
- *noserve - 拒绝 ntpq 和 ntp dc 查询以外的所有数据包。*
- *notrap - 防止 ntpdc 控制消息协议陷阱.*

- notrust - 拒绝未经加密身份验证的数据包。

- ntpport - 修改匹配算法，使其仅在源端口是标准 NTP UDP 端口 123 时应用限制。

- Version - 拒绝与当前 NTP 版本不匹配的数据包。

要将速率限制访问配置为完全不响应查询，相应的 restrict 命令必须具有 有限 选项。如果 ntpd 应该回复 KoD 数据包，则 restrict 命令需要同时具有 有限和 kod 选项。

ntpq 和 ntpdc 查询可用于放大攻击（如需了解更多详细信息，请参阅 CVE-2013-5211 ），不要从 restrict default 命令中删除公开可访问的系统上的 noquery 选项。

19.17.2. 配置对 NTP 服务的访问权限等级限制

要启用对系统中运行的 NTP 服务的速率限制访问，请在 restrict 命令中添加 restricted 选项，如 第 19.17.1 节 “配置 NTP 服务的访问权限控制” 中所述。如果您不想使用默认丢弃参数，则也使用 discard 命令，如下所述。

discard 命令采用以下格式：

```
discard average value minimum value monitor value
```

- 平均 - 指定允许的最低平均数据包间隔，它在 log2 秒中接受参数。默认值为3（2^3 等于 8 秒）。

- minimum - 指定允许的最小数据包间隔，它接受 log2 秒中的参数。默认值为1（2^1 等于 2 秒）。

- monitor - 指定数据包在超过允许的速率限值后丢弃的可能性。默认值为 3000 秒。这个选项适用于每秒接收 1000 个或更多请求的服务器。

discard 命令示例如下：

```
discard average 4
```

```
discard average 4 minimum 2
```

19.17.3. 添加 Peer 地址

要添加对等点的地址，就是说，运行同一级别 NTP 服务的服务器的地址，请使用 ntp.conf 文件中的 peer 命令。

peer 命令的格式如下：

```
peer address
```

其中 address 是 IP 单播地址或 DNS 可解析名称。地址必须是已知属于同一级别成员的系统的地址。同伴应至少有一个相互不同的时间源。同行通常处于相同的管理控制之下。

19.17.4. 添加服务器地址

要添加服务器的地址，也就是说，运行较高级别 NTP 服务的服务器的地址将使用 ntp.conf 文件中的 server 命令。

server 命令的格式如下：

```
server address
```

其中 address 是 IP 单播地址或 DNS 可解析名称。要接收数据包的远程参考服务器或本地参考时钟的地址。

19.17.5. 添加广播或多播服务器地址

要添加用于发送的广播或多播地址，就是说，广播或多播 NTP 数据包的地址将利用 ntp.conf 文件中的 broadcast 命令。

默认情况下，广播和多播模式需要进行身份验证。请参阅 第 19.6 节 “NTP 验证选项”。

广播 命令的格式如下：

```
broadcast address
```

其中 address 是将数据包发送到的 IP 广播或多播地址。

此命令将系统配置为充当 NTP 广播服务器。使用的地址必须是广播或多播地址。广播地址表示 IPv4 地址 255.255.255.255。默认情况下，路由器不会传递广播消息。多播地址可以是 IPv4 类 D 地址，也可以是 IPv6 地址。IANA 已将 IPv4 多播地址 224.0.1.1 和 IPv6 地址 FF05::101 （站点 local）分配给 NTP。也可在管理员范围内使用 IPv4 多播地址，如 RFC 2365 管理员跳过 IP 多播 所述。

19.17.6. 添加多播客户端地址

要添加多广播客户端地址，也就是说，若要配置用于 NTP 服务器发现的多播地址，请使用 ntp.conf 文件中的 manycastclient 命令。

manycastclient 命令采用以下格式：

```
manycastclient address
```

其中 address 是将要从中接收数据包的 IP 多播地址。客户端将向地址发送请求，然后从响应中选择最好的服务器并忽略其他服务器。然后 NTP 通信使用单播关联，就如在 ntp.conf 中列出了发现的 NTP 服务器一样。

此命令将系统配置为充当 NTP 客户端。系统可以同时是客户端和服务器。

19.17.7. 添加广播客户端地址

要添加广播客户端地址，也就是说，若要配置要监控的广播地址以获取广播 NTP 数据包，请使用 ntp.conf 文件中的 broadcastclient 命令。

broadcastclient 命令的格式如下：

```
broadcastclient
```

启用广播消息的接收。默认要求身份验证.请参阅 第 19.6 节 “NTP 验证选项”。

此命令将系统配置为充当 NTP 客户端。系统可以同时是客户端和服务器。

19.17.8. 添加多广播服务器地址

要添加多广播服务器地址，就是说，要配置地址以允许客户端通过多播 NTP 数据包发现服务器，请使用 ntp.conf 文件中的 manycastserver 命令。

manycastserver 命令的格式如下：

```
manycastserver address
```

支持发送多播消息。其中 address 是多播到.这应当与身份验证一起使用，以防止服务中断。

此命令将系统配置为充当 NTP 服务器。系统可以同时是客户端和服务器。

19.17.9. 添加多播客户端地址

要添加多播客户端地址，也就是说，要配置要监控多播 NTP 数据包的多播地址，请使用 ntp.conf 文件中的 多播client 命令。

multicast client 命令的格式如下：

```
multicastclient address
```

支持接收多播消息。其中 address 是要订阅的地址。这应当与身份验证一起使用，以防止服务中断。

此命令将系统配置为充当 NTP 客户端。系统可以同时是客户端和服务器。

19.17.10. 配置 Burst 选项

对公共服务器使用 突发 选项被视为滥用。不要将这个选项用于公共 NTP 服务器。仅将其用于您自己

的组织内的应用程序。

要提高平均时间偏移统计信息，请在服务器命令末尾添加以下选项：

```
burst
```

在每个轮询间隔中，当服务器响应时，系统将发送一个最多 8 个数据包的突发事件，而不是通常的一个数据包。用于 server 命令，以提高超时计算的平均质量。

19.17.11. 配置 iburst 选项

要延长初始同步所需的时间，请在服务器命令末尾添加以下选项：

```
iburst
```

当服务器无法访问时，发送包含 8 个数据包的突发事件，而不是通常的一个数据包。数据包间通常为 2s；但是，第一个和第二个数据包之间的间隔可以通过 calldelay 命令更改，以便有更多时间完成 modem 或 ISDN 调用。以便与 server 命令搭配使用，以减少初始同步所需的时间。现在，这是配置文件中的默认选项。

19.17.12. 使用密钥配置对称身份验证

要使用密钥配置对称验证，请在服务器或对等命令末尾添加以下选项：

```
key number
```

其中 数字 介于 1 到 65534 之间。此选项允许在数据包中使用消息身份验证代码 (MAC)。此选项用于 peer、server、cast 和many castclient 命令。

该选项可在 /etc/ntp.conf 文件中使用，如下所示：

```
server 192.168.1.1 key 10
broadcast 192.168.1.255 key 20
manycastclient 239.255.254.254 key 30
```

另请参阅 第 19.6 节 “NTP 验证选项”。

19.17.13. 配置 Poll Interval

要更改默认轮询间隔，请在服务器或对等命令末尾添加以下选项：

minpoll value and maxpoll value

用于更改默认轮询间隔的选项将通过增加 2 到值幂数来计算间隔（以秒为单位），换句话说，间隔以 log2 秒表示。默认的 minpoll 值为 6，2^6 等于 64 s。maxpoll 的默认值为 10，等于 1024。允许的值介于 3 到 17 范围中，分别等于 8 s 到 364 h。这些选项可用于 peer 或 server。设置较短的 maxpoll 可以提高时钟准确性。

19.17.14. 配置服务器首选项

要指定特定服务器应高于其他具有类似统计质量的服务器，请在服务器或对等命令的末尾添加以下选项：

prefer

使用此服务器进行同步，优先选择其他具有类似统计质量的服务器。此选项用于 peer 或 server 命令。

19.17.15. 配置 NTP 数据包的生存时间

要指定应该使用特定的生存时间(TTL)值代替默认值，请在服务器或 peer 命令末尾添加以下选项：

ttl value

指定要在广播服务器和多播 NTP 服务器发送的数据包中使用的数据包的生存时间值。指定多播客户端用于"展开环搜索"的最大时间到活动值。默认值为 127。

19.17.16. 配置 NTP 版本以使用

要指定应该使用特定版本的 NTP 来代替默认值，请在服务器或 peer 命令末尾添加以下选项：

version value

指定在创建的 NTP 数据包中设置的 NTP 版本。该值可以在 1 到 4 范围内。默认值为 4。

19.18. 配置硬件时钟更新

系统时钟可用于更新硬件时钟，也称为实时时钟(RTC)。本节演示了任务的三种方法：

即时一次性更新

要对硬件时钟进行即时一次性更新，以 root 用户身份运行这个命令：

```
~]# hwclock --systohc
```

在每次引导时更新

要在执行 ntpdate 同步程序后在每次引导时更新硬件时钟，请执行以下操作：

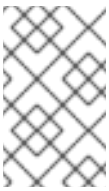
a. 在 /etc/sysconfig/ntpdate 文件中添加以下行：

```
SYNC_HWCLOCK=yes
```

b. 以 root 用户身份启用 ntpdate 服务：

```
~]# systemctl enable ntpdate.service
```

请注意，nt date 服务使用 /etc/ntp/step-tickers 文件中定义的 NTP 服务器。



注意

在虚拟机上，下次启动主机时将更新硬件时钟，而不是虚拟机。

通过 NTP 更新

每次由 ntpd 或 chronyd 服务更新系统时钟时，您可以更新硬件时钟：

以 root 用户身份启动 ntpd 服务：

```
~]# systemctl start ntpd.service
```

要使行为在引导后保留，请在引导时自动启动该服务：

```
~]# systemctl enable ntpd.service
```

或者

以 root 用户身份启动 chronyd 服务：

```
~]# systemctl start chronyd.service
```

要使行为在引导后保留，请在引导时自动启动该服务：

```
~]# systemctl enable chronyd.service
```

因此，每次由 ntpd 或 chronyd 同步系统时钟时，内核会在 11 分钟内自动更新硬件时钟。

警告

这种方法可能并不总是有效，因为上面提到的 11 分钟模式并不总是启用。因此，硬件时钟不一定会在系统时钟更新中更新。

要检查软件时钟与硬件时钟的同步，以 root 用户身份使用 ntpdc -c kerninfo 或 ntptime 命令：

```
~]# ntpdc -c kerninfo
```

结果可能类似如下：

```
pll offset:      0 s
pll frequency:    0.000 ppm
maximum error:    8.0185 s
estimated error:   0 s
status: 2001 pll nano
```

```
pll time constant:  6
precision:      1e-09 s
frequency tolerance: 500 ppm
```

或者

```
~]# ntptime
```

结果可能类似如下：

```
ntp_gettime() returns code 0 (OK)
 time dcba5798.c3dfe2e0 Mon, May 8 2017 11:34:00.765, (.765135199),
 maximum error 8010000 us, estimated error 0 us, TAI offset 0
ntp_adjtime() returns code 0 (OK)
 modes 0x0 (),
 offset 0.000 us, frequency 0.000 ppm, interval 1 s,
 maximum error 8010000 us, estimated error 0 us,
status 0x2001 (PLL,NANO),
 time constant 6, precision 0.001 us, tolerance 500 ppm,
```

要识别硬件时钟是否与系统时钟同步，请查看输出中的状态行。如果该行包含单词 unsync 或 UNSYNC，则硬件时钟不会与系统时钟同步。

硬件时钟与系统时钟同步。

```
status 0x2001 (PLL,NANO)
```

硬件时钟未与系统时钟同步。

```
status 0x41 (PLL,UNSYNC)
```

19.19. 配置时钟源

要列出系统中的可用时钟源，请运行以下命令：

```
~]$ cd /sys/devices/system/clocksource/clocksource0/
clocksource0]$ cat available_clocksource
kvm-clock tsc hpet acpi_pm
clocksource0]$ cat current_clocksource
kvm-clock
```

在上面的示例中，内核使用的是 kvm-clock。这是在启动时选择的，因为它是一个虚拟机。请注意，可用的时钟源取决于构架。

要覆盖默认时钟源，请在内核 GRUB 2 菜单条目的末尾附加 clocksource 指令。使用 grubby 工具进行更改。例如：要强制系统中的默认内核使用 tsc 时钟源，请输入以下命令：

```
~]# grubby --args=clocksource=tsc --update-kernel=DEFAULT
```

--update-kernel 参数还接受关键字 ALL，或用逗号分开的内核索引编号列表。

有关更改 GRUB 2 菜单的更多信息，请参阅 第 26 章 使用 GRUB 2。

19.20. 其它资源

以下信息来源提供有关 NTP 和 ntpd 的其他资源。

19.20.1. 安装的文档

- ntpd(8)手册页 - 详细介绍 ntpd，包括命令行选项。
- ntp.conf(5) 手册页 - 包含如何配置服务器和同级服务器关联的信息。
- ntpq(8)手册页 - 描述用于监控和查询 NTP 服务器的 NTP 查询实用程序。
- ntpdc(8)手册页 - 描述用于查询和更改 ntpd 状态的 ntpd 实用程序。
- ntp_auth(5) 手册页 - 描述 ntpd 的身份验证选项、命令和密钥管理。
- ntp_keygen(8)手册页 - 描述为 ntpd 生成公钥和私钥。
- ntp_acc(5) 手册页 - 使用 restrict 命令描述访问控制选项。

- ntp_mon(5) 手册页 - 描述用于收集统计数据的监控选项。
- ntp_clock(5) 手册页 - 描述用于配置参考时钟的命令.
- ntp_misc(5) 手册页 - 描述各种选项.
- ntp_decode(5) 手册页 - 列出用于 ntpd 报告和监控的状态词语、事件消息和错误代码。
- ntpstat(8)手册页 - 描述用于报告本地计算机上运行的 NTP 守护进程同步状态的实用程序。
- ntptime(8)手册页 - 描述用于读取和设置内核时间变量的实用程序。
- tickadj(8)手册页 - 描述用于读取和可选设置勾号长度的实用程序。

19.20.2. 有用的网站

http://doc.ntp.org/

NTP 文档归档

http://www.eecis.udel.edu/~mills/ntp.html

网络时间同步研究项目.

http://www.eecis.udel.edu/~mills/ntp/html/manyopt.html

有关 NTPv4 中自动服务器发现的信息。

第 20 章 使用 PTP4L 配置 PTP

20.1. PTP 简介

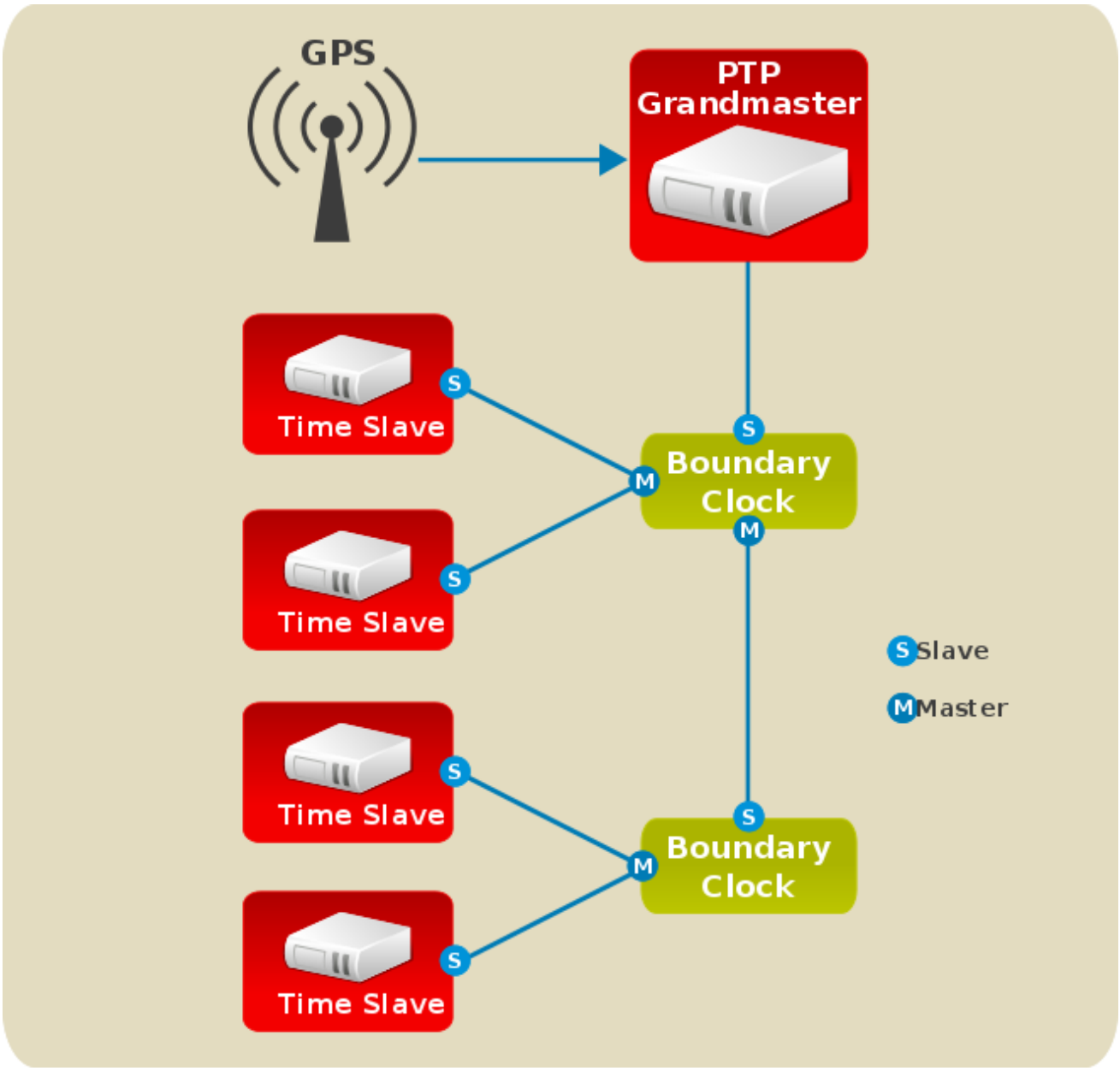
预告时间协议 (PTP)是一个用于同步网络中时钟的协议。与硬件支持配合使用时，P TP 可以采用次微秒精确度，这比 NTP 可获得的要好得多。PTP 支持在内核和用户空间间划分。Red Hat Enterprise Linux 中的内核包括对 PTP 时钟的支持，这些时钟由网络驱动程序提供。协议的实际实施称为 linuxptp，根据适用于 Linux 的 IEEE 标准 1588 实现 PTPv2。

linuxptp 软件包包括用于时钟同步的 ptp4l 和 phc2sys 程序。ptp4l 程序实现 PTP 边界时钟和普通时钟。使用硬件时间戳时，它用于将 PTP 硬件 时钟与主时钟同步，并将系统时钟与 master 时钟同步的软件时间戳。phc2sys 程序只需要硬件时间戳，才能将系统时钟与 网络接口卡 (NIC)上的 PTP 硬件时钟同步。

20.1.1. 了解 PTP

PTP 同步的时钟按主从层次结构组织。从接口同步到其主设备，主从设备可能是自有主设备的主接口。层次结构由最佳 master 时钟 (BMC)算法自动创建和更新，该算法在每个时钟上运行。当时钟只有一个端口时，它可以是 master 或 slave ，此类时钟被称为普通时钟(OC)。具有多个端口的时钟可以在一个端口和另一个端口上主，此类时钟称为边界时钟 (BC)。顶级主时钟称为协调主机时钟 ，可以通过使用全局定位系统 (NC)时间源同步。通过使用基于NC 的时间源，不同的网络可以与高度精确度同步。

图 20.1. PTPtenmaster、边界和从属时钟



20.1.2. PTP 的优点

PTP 与网络时间协议 (NTP)相比有一个主要优势，那就是各种网络接口控制器 (NIC)和网络交换机中存在硬件支持。这种特殊硬件允许 PTP 考虑消息传输的延迟，并极大地提高了时间同步的准确性。虽然可以在网络中使用非 PTP 启用的硬件组件，但这通常会导致 jitter 增加或引入非对称，从而导致同步不准确，这增加了通信路径中使用的多个非 PTP 识别组件。为了获得最佳准确性，建议启用 PTP 时钟间的所有网络组件。在大型网络中的时间同步，并非所有网络硬件都支持 PTP 可能更适合 NTP。

对于硬件 PTP 支持，NIC 中有自己的板载时钟，用于为接收和传输的 PTP 信息添加时间戳。这是与 PTP master 同步的板载时钟，计算机的系统时钟与 NIC 上的 PTP 硬件时钟同步。对于软件 PTP 支持，系统时钟用于为 PTP 信息加上 时间戳，并直接与 PTP master 同步。硬件 PTP 支持提供更高的准确性，因为 NIC 可以在准确发送和接收时对 PTP 数据包进行时间戳，而软件 PTP 支持需要操作系统对 PTP 数据包进行额外的处理。

20.2. 使用 PTP

要使用 PTP，预期接口的内核网络驱动程序必须支持软件或硬件时间戳功能。

20.2.1. 检查驱动程序和硬件支持

除了驱动程序中存在的硬件时间戳支持外，NIC 还必须能够支持物理硬件中的此功能。验证特定驱动程序和 NIC 的时间戳功能的最佳方法是，使用 ethtool 工具来查询接口。在本例中，eth3 是您要检查的接口：

```
~]# ethtool -T eth3
Time stamping parameters for eth3:
Capabilities:
   hardware-transmit   (SOF_TIMESTAMPING_TX_HARDWARE)
   software-transmit   (SOF_TIMESTAMPING_TX_SOFTWARE)
   hardware-receive   (SOF_TIMESTAMPING_RX_HARDWARE)
   software-receive   (SOF_TIMESTAMPING_RX_SOFTWARE)
   software-system-clock (SOF_TIMESTAMPING_SOFTWARE)
   hardware-raw-clock  (SOF_TIMESTAMPING_RAW_HARDWARE)
PTP Hardware Clock: 0
Hardware Transmit Timestamp Modes:
   off        (HWTSTAMP_TX_OFF)
   on         (HWTSTAMP_TX_ON)
Hardware Receive Filter Modes:
   none       (HWTSTAMP_FILTER_NONE)
   all        (HWTSTAMP_FILTER_ALL)
```

注意

ethtool 输出的 PTP 硬件 Clock 值是 PTP 硬件时钟的索引。它对应于 /dev/ptp* 设备的命名。第一个 PHC 的索引为 0。

对于软件时间戳支持，参数列表应包括：

- SOF_TIMESTAMPING_SOFTWARE
- SOF_TIMESTAMPING_TX_SOFTWARE
- SOF_TIMESTAMPING_RX_SOFTWARE

对于硬件时间戳支持，参数列表应包括：

- SOF_TIMESTAMPING_RAW_HARDWARE
- SOF_TIMESTAMPING_TX_HARDWARE
- SOF_TIMESTAMPING_RX_HARDWARE

20.2.2. 安装 PTP

Red Hat Enterprise Linux 中的内核包括对 PTP 的支持。用户空间支持由 linuxptp 软件包中的工具提供。要安装 linuxptp，以 root 用户身份运行以下命令：

```
~]# yum install linuxptp
```

这将安装 ptp4l 和 phc2sys。

不要运行多个服务来同时设置系统时钟的时间。如果要使用 NTP 为 PTP 时间提供服务，请参阅 第 20.8 节 “使用 NTP 提供 PTP 时间”。

20.2.3. 启动 ptp4l

ptp4l 程序可以从命令行启动，也可以作为服务启动。作为服务运行时，可在 /etc/sysconfig/ptp4l 文件中指定选项。应该在 /etc/ptp4l.conf 文件中指定服务和命令行使用时所需的选项。/etc/sysconfig/ptp4l 文件包含 -f /etc/ptp4l.conf 命令行选项，这会导致 ptp4l 程序读取 /etc/ptp4l.conf 文件并处理它所包含的选项。/etc/ptp4l.conf 的使用在 第 20.4 节 “指定配置文件” 中进行了说明。有关不同 ptp4l 选项和配置文件设置的更多信息，请参阅 ptp4l(8)手册页。

启动 ptp4l 作为服务

要启动 ptp4l 作为服务，以 root 用户身份运行以下命令：

```
~]# systemctl start ptp4l
```

有关在 Red Hat Enterprise Linux 7 中管理系统服务的详情请参考 第 10 章 使用 systemd 管理服务。

从命令行使用 ptp4l

ptp4l 程序默认尝试**使用硬件**时间**戳。要使用**带**有硬件**时间**戳功能**驱动**程序和 NIC 的 ptp4l，您必**须**提供要与 -i** 选项**一起使用的网**络**接口。以 root 用**户**身份**输入**以下命令：**

```
~]# ptp4l -i eth3 -m
```

其中 eth3 是您要配置的接口。以下是当 NIC 上的 PTP 时钟**与 master 同步**时**的 ptp4l** 输**出示例：**

```
~]# ptp4l -i eth3 -m
selected eth3 as PTP clock
port 1: INITIALIZING to LISTENING on INITIALIZE
port 0: INITIALIZING to LISTENING on INITIALIZE
port 1: new foreign master 00a069.fffe.0b552d-1
selected best master clock 00a069.fffe.0b552d
port 1: LISTENING to UNCALIBRATED on RS_SLAVE
master offset -23947 s0 freq +0 path delay    11350
master offset -28867 s0 freq +0 path delay    11236
master offset -32801 s0 freq +0 path delay    10841
master offset -37203 s1 freq +0 path delay    10583
master offset -7275 s2 freq -30575 path delay  10583
port 1: UNCALIBRATED to SLAVE on MASTER_CLOCK_SELECTED
master offset -4552 s2 freq -30035 path delay  10385
```

master 偏移值**是从主控机偏移（以**纳**秒**为单**位）**测**量的偏移量。s0、s1 和 s2 字符串表示不同的**时钟**交换状**态**：s0 已被解**锁，**s1** 为时钟**步**骤，s2 已**锁**定。当 servo** 处**于**锁**定状**态**(s2)后，**时钟**将不会受阻（只是**缓**慢**调**整），除非在配置文件中将 pi_offset_const** 选项**设**置**为**正**值（在 ptp4l(8)man page 中描述）。adj** 值**是每十**亿**(ppb)的**时钟频**率**调**整。路径延**迟值**是从主控机（以**纳**秒**为单**位）**发**送的同步消息的**预计**延**迟。端口 0 是用于本地 PTP 管理的 Unix 域套接字。端口 1 是 eth3 接口（基于上述示例）。INITIALIZING、LISTENING、UNCALIBRATED 和 SLAVE 是 INITIALIZE、RS_SLAVE、MASTER_CLOCK_SELECTED 事件的一些可能端口状**态**。在最后一个状**态**更改消息中，端口状**态**从 UNCALIBRATED 更改**为 SLAVE，表示与 PTP master** 时钟**成功同步。**

来自 ptp4l 的日志记录**信息**

默认**情况下，消息**发**送到 /var/log/messages。但是，指定 -m** 选项**可**记录**到**标**准**输**出，**这对**于**调试**非常有用。**

要启**用**软**件**时间**戳，需要按如下所示使用 -S** 选项**：**

```
~]# ptp4l -i eth3 -m -S
```

20.2.3.1. 选择**延**迟测**量机制**

有两种不同的延迟测量机制，它们可以通过添加到 ptp4l 命令的选项进行选择，如下所示：

-P

P 选择 peer-to-peer(P2P)延迟测量机制。

P2P 机制是首选的，因为它能更快地响应网络拓扑中的更改，并且可能比其他机制在测量延迟时更准确。P2P 机制只能在拓扑中使用，每个端口最多与另一个 P2P 端口交换 PTP 消息。它必须在通信路径上被所有硬件（包括透明时钟）提供支持和使用。

-E

E 选择 端到端 (E)延迟测量机制。这是默认值。

E2E 机制也称为延迟"请求响应"机制。

-A

A 支持 自动选择延迟测量机制。

自动选项在 E2E 模式下启动 ptp4l。如果收到对等延迟请求，它将更改为 P2P 模式。

注意

单一 PTP 通信路径中的所有时钟都必须使用相同的机制来测量延迟。在以下情况下会打印警告信息：

- 使用 E2E 机制在端口上收到对等延迟请求时。
- 使用 P2P 机制在端口上收到 E2E 延迟请求时。

20.3. 使用带有多个接口的 PTP

当将 PTP 与不同网络中的多个接口一起使用时，需要将反向路径转发模式改为松散模式。Red Hat Enterprise Linux 7 默认按照 RFC 3704(Ingress Filtering for Multihomed Networks)中的 Strict Reverse Path 推荐使用 Strict Reverse Path 转发。如需了解更多详细信息，请参阅 Red Hat Enterprise Linux 7 安全指南中的 Reverse Path Forwarding 部分。

sysctl 实用程序用于为内核中的可调项读取和写入值。可以使用命令行上的 sysctl 命令直接对正在运行的系统进行更改，并通过向 /etc/sysctl.conf 文件添加行来进行永久性更改。

- 要全局更改为松散模式过滤，以 root 用户身份输入以下命令：

```
~]# sysctl -w net.ipv4.conf.default.rp_filter=2
~]# sysctl -w net.ipv4.conf.all.rp_filter=2
```

- 要更改每个网络接口的反向路径过滤模式，请在所有 PTP 接口上使用 net.ipv4.interface.rp_filter 命令。例如，对于设备名称为 em1 的接口：

```
~]# sysctl -w net.ipv4.conf.em1.rp_filter=2
```

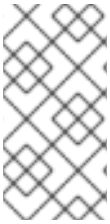
要使这些设置在重新引导后保留，请修改 /etc/sysctl.conf 文件。您可以更改所有接口或特定接口的模式。

要更改所有接口的模式，请使用以 root 用户身份运行的编辑器打开 /etc/sysctl.conf 文件，并添加一行，如下所示：

```
net.ipv4.conf.all.rp_filter=2
```

要只更改某些接口，请以以下格式添加多行：

```
net.ipv4.conf.interface.rp_filter=2
```



注意

使用所有接口和特定接口的设置时，将对每个接口进行源验证时使用来自 conf/{all,interface}/rp_filter 的最大值。

您还可以使用默认设置来更改模式，这意味着它仅适用于新创建的接口。

有关使用 sysctl 参数中的所有 、 默认 或特定设备设置的更多信息，请参阅红帽知识库文章 "所有"、"default"和 sysctl 参数中的特定设备有什么区别？

请注意，由于在引导过程中运行 sysctl 服务的时间，您可能遇到两种类型的问题：

1. 在 sysctl 服务运行前会载入驱动程序。

   在这种情况下，受影响的网络接口使用内核中预先设置的模式，并且忽略 sysctl 的默认值。

   有关此问题的解决方案，请查看红帽知识库文章"所有"、"默认"和 sysctl 参数中的特定设备有什么区别？

2. sysctl 服务运行后会加载或重新加载驱动程序。

   在这种情况下，重启后可能不会使用一些 sysctl.conf 参数。这些设置可能不可用，或者它们可能返回默认值。

   要解决这个问题，请参阅 Red Hat 知识库文章在重启后不会使用一些 sysctl.conf 参数，手动调整设置可以正常工作。

20.4. 指定配置文件

可以在可选配置文件中设置命令行选项和其他选项，这些选项不能在命令行中设置。

默认情况下不读取任何配置文件，因此需要使用 -f 选项在运行时指定该文件。例如：

```
~]# ptp4l -f /etc/ptp4l.conf
```

等同于以上显示的 -i eth3 -m -S 选项的配置文件如下：

```
~]# cat /etc/ptp4l.conf
[global]
verbose         1
time_stamping     software
[eth3]
```

20.5. 使用 PTP 管理客户端

PTP 管理客户端 pmc 可以用来从 ptp4l 获取更多信息，如下所示：

```
~]# pmc -u -b 0 'GET CURRENT_DATA_SET'
sending: GET CURRENT_DATA_SET
   90e2ba.fffe.20c7f8-0 seq 0 RESPONSE MANAGMENT CURRENT_DATA_SET
      stepsRemoved    1
      offsetFromMaster -142.0
      meanPathDelay   9310.0
```

```
~]# pmc -u -b 0 'GET TIME_STATUS_NP'
sending: GET TIME_STATUS_NP
   90e2ba.fffe.20c7f8-0 seq 0 RESPONSE MANAGMENT TIME_STATUS_NP
      master_offset       310
      ingress_time        1361545089345029441
      cumulativeScaledRateOffset  +1.000000000
      scaledLastGmPhaseChange  0
      gmTimeBaseIndicator    0
      lastGmPhaseChange     0x0000'0000000000000000.0000
      gmPresent         true
      gmIdentity        00a069.fffe.0b552d
```

将 -b 选项设置为 零，将边界限制为本地运行的 ptp4l 实例。更大的边界值将从本地时钟进一步检索 PTP 节点的信息。可检索的信息包括：

- stepsRemoved 是到movmaster 时钟的通信路径数。
- offsetFromMaster 和 master_offset 是 master 最后一次测量时钟偏移（以纳秒为单位）。
- meanPathDelay 是从主控机（以纳秒为单位）发送的同步消息的预计延迟。
- 如果 gmPresent 为 true，P TP 时钟与 master 同步，本地时钟不是默认时钟。
- gmIdentity 是宿主的身份。

要获得 pmc 命令的完整列表，以 root 用户身份输入以下内容：

```
~]# pmc help
```

pmc(8)man page 中提供了其他信息。

20.6. 同步时钟

phc2sys 程序用于将系统时钟与 NIC 上的 PTP 硬件时钟(PHC)同步。phc2sys 服务在 /etc/sysconfig/phc2sys 配置文件中配置。/etc/sysconfig/phc2sys 文件中的默认设置如下：

```
OPTIONS="-a -r"
```

-a 选项可使 phc2sys 从 ptp4l 应用读取时钟同步。它将紧随 PTP 端口状态的更改，相应地调整 NIC 硬件时钟之间的同步。系统时钟没有同步，除非还指定了 -r 选项。如果您希望系统时钟有资格成为时间源，请指定 -r 选项两次。

更改 /etc/sysconfig/phc2sys 后，以 root 用户身份从命令行重启 phc2sys 服务：

```
~]# systemctl restart phc2sys
```

在正常情况下，使用 systemctl 命令启动、停止和重新启动 phc2sys 服务。

如果您不想将 phc2sys 启动为服务，您可以从命令行启动它。例如，以 root 用户身份输入以下命令：

```
~]# phc2sys -a -r
```

-a 选项可使 phc2sys 从 ptp4l 应用读取时钟同步。如果您希望系统时钟有资格成为时间源，请指定 -r 选项两次。

或者，使用 -s 选项将系统时钟同步到特定接口的 PTP 硬件时钟。例如：

```
~]# phc2sys -s eth3 -w
```

-w 选项等待正在运行的 ptp4l 应用程序同步 PTP 时钟，然后从 ptp4l 检索TAI 到 UTC 偏移。

通常，PTP 在国际原子时间(TAI)时区运行，而系统时钟则保持在统一的通用时间 (UTTC)中。TAI 和 UTC 计时器之间的当前偏移是 36 秒。插入或删除跳秒时的偏移会改变，这通常每几年发生一次。如果不

使用 -w，则需要使用 -O 选项手动设置这个偏移，如下所示：

```
~]# phc2sys -s eth3 -O -36
```

当 phc2sys servo 处于锁定状态后，除非使用 -S 选项，否则时钟不会受到影响。这意味着 phc2sys 程序应在 ptp4l 程序同步了 PTP 硬件时钟后启动。但是，使用 -w 时，在 ptp4l 后不需要启动 phc2sys，因为它将等待它同步时钟。

phc2sys 程序也可以作为服务启动：

```
~]# systemctl start phc2sys
```

作为服务运行时，可在 /etc/sysconfig/phc2sys 文件中指定选项。有关不同 phc2sys 选项的更多信息，请参见 phc2sys(8)man page。

请注意，本节中的示例假定 命令在从系统或从端口上运行。

20.7. 验证时间同步

当 PTP 时间同步正常工作时，如果使用硬件时间戳，则定期将带有偏移和频率调整的新消息打印到 ptp4l 和 phc2sys 输出。输出值很快会聚合。您可以在 /var/log/messages 文件中看到这些消息。

以下是 ptp4l 和 phc2sys 输出示例：

- 偏移（以纳秒为单位）
- 频率偏移（以每十亿分之一为单位(ppb)）
- 路径延迟（以纳秒为单位）

ptp4l 输出示例：

```
ptp4l[352.359]: selected /dev/ptp0 as PTP clock
ptp4l[352.361]: port 1: INITIALIZING to LISTENING on INITIALIZE
```

```
ptp4l[352.361]: port 0: INITIALIZING to LISTENING on INITIALIZE
ptp4l[353.210]: port 1: new foreign master 00a069.fffe.0b552d-1
ptp4l[357.214]: selected best master clock 00a069.fffe.0b552d
ptp4l[357.214]: port 1: LISTENING to UNCALIBRATED on RS_SLAVE
ptp4l[359.224]: master offset    3304 s0 freq   +0 path delay   9202
ptp4l[360.224]: master offset    3708 s1 freq -29492 path delay   9202
ptp4l[361.224]: master offset   -3145 s2 freq -32637 path delay   9202
ptp4l[361.224]: port 1: UNCALIBRATED to SLAVE on MASTER_CLOCK_SELECTED
ptp4l[362.223]: master offset    -145 s2 freq -30580 path delay   9202
ptp4l[363.223]: master offset    1043 s2 freq -29436 path delay   8972
ptp4l[364.223]: master offset    266 s2 freq -29900 path delay   9153
ptp4l[365.223]: master offset    430 s2 freq -29656 path delay   9153
ptp4l[366.223]: master offset    615 s2 freq -29342 path delay   9169
ptp4l[367.222]: master offset    -191 s2 freq -29964 path delay   9169
ptp4l[368.223]: master offset    466 s2 freq -29364 path delay   9170
ptp4l[369.235]: master offset     24 s2 freq -29666 path delay   9196
ptp4l[370.235]: master offset    -375 s2 freq -30058 path delay   9238
ptp4l[371.235]: master offset    285 s2 freq -29511 path delay   9199
ptp4l[372.235]: master offset    -78 s2 freq -29788 path delay   9204
```

phc2sys 输出示例：

```
phc2sys[526.527]: Waiting for ptp4l...
phc2sys[527.528]: Waiting for ptp4l...
phc2sys[528.528]: phc offset   55341 s0 freq   +0 delay  2729
phc2sys[529.528]: phc offset   54658 s1 freq -37690 delay  2725
phc2sys[530.528]: phc offset    888 s2 freq -36802 delay  2756
phc2sys[531.528]: phc offset   1156 s2 freq -36268 delay  2766
phc2sys[532.528]: phc offset    411 s2 freq -36666 delay  2738
phc2sys[533.528]: phc offset    -73 s2 freq -37026 delay  2764
phc2sys[534.528]: phc offset    39 s2 freq -36936 delay  2746
phc2sys[535.529]: phc offset    95 s2 freq -36869 delay  2733
phc2sys[536.529]: phc offset   -359 s2 freq -37294 delay  2738
phc2sys[537.529]: phc offset   -257 s2 freq -37300 delay  2753
phc2sys[538.529]: phc offset    119 s2 freq -37001 delay  2745
phc2sys[539.529]: phc offset    288 s2 freq -36796 delay  2766
phc2sys[540.529]: phc offset   -149 s2 freq -37147 delay  2760
phc2sys[541.529]: phc offset   -352 s2 freq -37395 delay  2771
phc2sys[542.529]: phc offset    166 s2 freq -36982 delay  2748
phc2sys[543.529]: phc offset    50 s2 freq -37048 delay  2756
phc2sys[544.530]: phc offset    -31 s2 freq -37114 delay  2748
phc2sys[545.530]: phc offset   -333 s2 freq -37426 delay  2747
phc2sys[546.530]: phc offset    194 s2 freq -36999 delay  2749
```

要减少 ptp4l 输出并只打印值，请使用 summary_interval 指令。summary_interval 指令指定为 2，表示 n 的电源（以秒为单位）。例如，要将输出减少到每 1024 秒，请在 /etc/ptp4l.conf 文件中添加以下行：

```
summary_interval 10
```

ptp4l 输出示例，概述_interval 设置为 6：

```
ptp4l: [615.253] selected /dev/ptp0 as PTP clock
ptp4l: [615.255] port 1: INITIALIZING to LISTENING on INITIALIZE
ptp4l: [615.255] port 0: INITIALIZING to LISTENING on INITIALIZE
ptp4l: [615.564] port 1: new foreign master 00a069.fffe.0b552d-1
ptp4l: [619.574] selected best master clock 00a069.fffe.0b552d
ptp4l: [619.574] port 1: LISTENING to UNCALIBRATED on RS_SLAVE
ptp4l: [623.573] port 1: UNCALIBRATED to SLAVE on MASTER_CLOCK_SELECTED
ptp4l: [684.649] rms 669 max 3691 freq -29383 ± 3735 delay 9232 ± 122
ptp4l: [748.724] rms 253 max 588 freq -29787 ± 221 delay 9219 ± 158
ptp4l: [812.793] rms 287 max 673 freq -29802 ± 248 delay 9211 ± 183
ptp4l: [876.853] rms 226 max 534 freq -29795 ± 197 delay 9221 ± 138
ptp4l: [940.925] rms 250 max 562 freq -29801 ± 218 delay 9199 ± 148
ptp4l: [1004.988] rms 226 max 525 freq -29802 ± 196 delay 9228 ± 143
ptp4l: [1069.065] rms 300 max 646 freq -29802 ± 259 delay 9214 ± 176
ptp4l: [1133.125] rms 226 max 505 freq -29792 ± 197 delay 9225 ± 159
ptp4l: [1197.185] rms 244 max 688 freq -29790 ± 211 delay 9201 ± 162
```

默认情况下，summary_interval 设置为 0，因此每秒打印一次消息，这是最大频率。消息记录在 LOG_INFO 级别上。要禁用信息，请使用 -l 选项将最大日志级别设置为 5 或更低：

```
~]# phc2sys -l 5
```

您可以使用 -u 选项来减少 phc2sys 输出：

```
~]# phc2sys -u summary-updates
```

其中 Summary-updates 是要包括在汇总统计中的时钟更新数。下面是一个示例：

```
~]# phc2sys -s eth3 -w -m -u 60
phc2sys[700.948]: rms 1837 max 10123 freq -36474 ± 4752 delay 2752 ± 16
phc2sys[760.954]: rms 194 max 457 freq -37084 ± 174 delay 2753 ± 12
phc2sys[820.963]: rms 211 max 487 freq -37085 ± 185 delay 2750 ± 19
phc2sys[880.968]: rms 183 max 440 freq -37102 ± 164 delay 2734 ± 91
phc2sys[940.973]: rms 244 max 584 freq -37095 ± 216 delay 2748 ± 16
phc2sys[1000.979]: rms 220 max 573 freq -36666 ± 182 delay 2747 ± 43
phc2sys[1060.984]: rms 266 max 675 freq -36759 ± 234 delay 2753 ± 17
```

与这些选项一起使用时，更新统计数据的时间间隔设置为 60 秒(-u)，ph c2sys 等待 ptp4l 处于同步状态(-w)，并将消息打印到标准输出(-m)。有关 phc2sys 选项的详情，请查看 phc2sys(5) man page。

输出包括：

- 偏移根平均值方块(rms)
- 最大绝对误差（最大值）
- 频率偏移(freq)：其平均值和标准偏差
- 路径延迟(delay)：其含义和标准偏差

20.8. 使用 NTP 提供 PTP 时间

可以使用 LOCAL 参考时钟驱动程序将 ntpd 守护进程配置为从系统时钟同步（通过 ptp4l 或 phc2sys 同步的系统时钟）中分配时间。为防止 ntpd 调整系统时钟，ntp .conf 文件不得指定任何 NTP 服务器。以下是 ntp.conf 的一个最小示例：

```
~]# cat /etc/ntp.conf
server  127.127.1.0
fudge  127.127.1.0 stratum 0
```

注意

当 DHCP 客户端程序, dhclient 收到来自 DHCP 服务器的 NTP 服务器列表时，它会将它们添加到 ntp.conf 并重新启动服务。要禁用此功能，请将 PEERNTP=no 添加到 /etc/sysconfig/network。

20.9. 使用 PTP 提供 NTP 时间

NTP 到 PTP 同步方向也可以相反。当使用 ntpd 同步系统时钟时，pt p4l 可使用 priority1 选项（或其他最佳 master 时钟算法中包含的其他时钟选项）配置为维护主时钟并通过 PTP 从系统时钟分配时间：

```
~]# cat /etc/ptp4l.conf
[global]
priority1 127
[eth3]
# ptp4l -f /etc/ptp4l.conf
```

使用硬件时间戳时，phc2sys 需要用于将 PTP 硬件时钟与系统时钟同步。如果将 phc2sys 作为服务运行，请编辑 /etc/sysconfig/phc2sys 配置文件。/etc/sysconfig/phc2sys 文件中的默认设置如下：

```
OPTIONS="-a -r"
```

作为 root 用户编辑该行，如下所示：

```
~]# vi /etc/sysconfig/phc2sys
OPTIONS="-a -r -r"
```

r 选项在此处使用两次，以允许从系统时钟同步 NIC 上的 PTP 硬件时钟。重启 phc2sys 服务以使更改生效：

```
~]# systemctl restart phc2sys
```

为防止 PTP 时钟频率出现快速变化，可以使用较小的 P （转发）和 I （智能）常量来松散到系统时钟的同步：

```
~]# phc2sys -a -r -r -P 0.01 -I 0.0001
```

20.10. 使用 TIMEMASTER 同步到 PTP 或者 NTP TIME

如果网络上有多个 PTP 域，或者需要回退到 NTP，则 timemaster 程序可用于将系统时钟同步到所有可用的时间源。PTP 时间由 phc2sys 和 ptp4l 通过 共享内存驱动程序 提供（SHM 参考时钟到 chronyd 或 ntpd （取决于系统上配置的 NTP 守护进程）。然后 NTP 守护进程可以比较所有时间源，包括 PTP 和 NTP，并使用最佳源来同步系统时钟。

启动时，timemaster 会读取指定 NTP 和 PTP 时间源的配置文件，检查哪些网络接口具有自己的或共享 PTP 硬件时钟(PHC)，为 ptp4l 和 chronyd 或 ntpd 生成配置文件， 并根据需要启动 ptp4l、phc2sys 和 chronyd 或 ntpd 进程。它将在退出时删除生成的配置文件。它将 chronyd、ntpd 和 ptp4l 的配置文件写入 /var/run/timemaster/。

20.10.1. 启动 timemaster 即服务

要启动 timemaster 作为服务，以 root 身份运行以下命令：

```
~]# systemctl start timemaster
```

这将读取 /etc/timemaster.conf 中的选项。有关在 Red Hat Enterprise Linux 7 中管理系统服务的详

情请参考 第 10 章 使用 systemd 管理服务。

20.10.2. 了解时间 master 配置文件

Red Hat Enterprise Linux 提供了一个默认的 /etc/timemaster.conf 文件，其中包含包含默认选项的多个部分。部分标题括在方括号中。

要查看默认配置，请使用以下命令：

```
~]$ less /etc/timemaster.conf
# Configuration file for timemaster

#[ntp_server ntp-server.local]
#minpoll 4
#maxpoll 4

#[ptp_domain 0]
#interfaces eth0

[timemaster]
ntp_program chronyd

[chrony.conf]
include /etc/chrony.conf

[ntp.conf]
includefile /etc/ntp.conf

[ptp4l.conf]

[chronyd]
path /usr/sbin/chronyd
options -u chrony

[ntpd]
path /usr/sbin/ntpd
options -u ntp:ntp -g

[phc2sys]
path /usr/sbin/phc2sys

[ptp4l]
path /usr/sbin/ptp4l
```

请注意如下部分：

```
[ntp_server address]
```

这是 NTP 服务器部分"ntp-server.local"的示例，是本地 LAN 上 NTP 服务器的主机名示例。根据需要添加更多部分，将主机名或 IP 地址 用作小节名称的一部分。请注意，示例中的简短轮询值不适合公共服务器，请参阅 第 19 章 使用 ntpd 配置 NTP 了解合适的 minpoll 和 maxpoll 值的说明。

请注意如下部分：

> [ptp_domain number]

"PTP 域"是一个或多个 PTP 时钟的组，它们相互同步。它们不一定同步到另一个域中的时钟。使用相同域编号配置的时钟组成了域。这包括一个 PTP Pu master 时钟。每个"PTP 域"部分中的域编号需要与网络上配置的 PTP 域之一对应。

每个接口都启动 ptp4l 实例，该接口具有自己的 PTP 时钟，并且硬件时间戳会自动启用。支持硬件时间戳的接口附加了 PTP 时钟(PHC)，但 NIC 上的一组接口可共享 PHC。将为共享同一 PHC 的每个接口以及仅支持软件时间戳的每个接口启动单独的 ptp4l 实例。所有 ptp4l 实例都配置为作为从设备运行。如果在多个 PTP 域中指定了硬件时间戳的接口，则只有创建的第一个 ptp4l 实例才会启用硬件时间戳。

请注意如下部分：

> [timemaster]

默认 timemaster 配置包括系统 ntpd 和 chrony 配置（/etc/ntp.conf 或 /etc/chronyd.conf），以包含访问限制和身份验证密钥的配置。这意味着，指定的所有 NTP 服务器都将与 timemaster 一起使用。

部分标题如下：

- [ntp_server ntp-server.local] - 指定此服务器的轮询间隔。根据需要创建其他部分。在小节标题中包含主机名或 IP 地址。
- [ptp_domain 0] - 指定为此域配置了 PTP 时钟的接口。根据需要，使用相应的域号创建其他部分。
- [timemaster] - 指定要使用的 NTP 守护进程。可能的值有 chronyd 和 ntpd。

- [chrony.conf] - 指定要复制到为 chronyd 生成的配置文件的任何额外设置。
- [NTP.conf] - 指定要复制到为 ntpd 生成的配置文件的任何其他设置。
- [ptp4l.conf] - 指定要复制到为 ptp4l 生成的配置文件中的选项。
- [chronyd] - 指定命令行上要传递给 chronyd 的任何其他设置。
- [ntpd] - 指定命令行上要传递给 ntpd 的任何其他设置。
- [phc2sys] - 指定要在命令行中传递给 phc2sys 的任何其他设置。
- [ptp4l] - 指定命令行上要传递给所有 ptp4l 实例的任何其他设置。

部分标题和其内容在 timemaster(8)手册页 中详细讨论。

20.10.3. 配置 timemaster 选项

编辑 timemaster 配置文件

1. 要更改默认配置，打开 /etc/timemaster.conf 文件以 root 用户身份编辑：

```
~]# vi /etc/timemaster.conf
```

2. 对于您要使用 timemaster 控制的每个 NTP 服务器，创建 [ntp_server address] 部分。请注意，示例部分中的简短轮询值不适合公共服务器，请参阅 第 19 章 使用 ntpd 配置 NTP 了解合适的 minpoll 和 maxpo ll 值的说明。

3. 要添加域中应该使用的接口，编辑 #[ptp_domain 0] 部分并添加接口。根据需要创建其他域。例如：

```
[ptp_domain 0]
  interfaces eth0
```

```
[ptp_domain 1]
interfaces eth1
```

4. 如果需要将 ntpd 用作此系统上的 NTP 守护进程，请将 [timemaster] 部分中的默认条目从 chronyd 改为 ntpd。有关 ntpd 和 chronyd 之间的区别，请查看 第 18 章 使用 chrony 套件配置 NTP。

5. 如果在此系统上使用 chronyd 作为 NTP 服务器，请在默认选项下方添加任何附加选项，在 [chrony.conf] 部分中包含 /etc/ chrony.conf 条目。如果已知到 /etc/chrony.conf 的路径已更改，请编辑默认 include 条目。

6. 如果在此系统上使用 ntpd 作为 NTP 服务器，请在默认选项下方添加任何附加选项，在 [ntp.conf] 部分中包含 /etc/ ntp.conf 条目。如果已知到 /etc/ntp.conf 的路径已更改，请编辑默认 include 条目。

7. 在 [ptp4l.conf] 部分中，添加要复制到为 ptp4l 生成的配置文件的任何选项。本章记录了常用选项和更多信息，请参见 ptp4l(8)手册页 面。

8. 在 [chronyd] 部分中，添加 时间master 调用时要传递给 chronyd 的任何命令行选项。有关使用 chronyd 的详情，请查看 第 18 章 使用 chrony 套件配置 NTP。

9. 在 [ntpd] 部分中，添加 时间master 调用时要传递给 ntpd 的任何命令行选项。有关使用 ntpd 的详情，请查看 第 19 章 使用 ntpd 配置 NTP。

10. 在 [phc2sys] 部分中，添加任何要传递给 phc2sys 的命令行选项（如果 时间master 调用 ）。本章记录了常用选项和更多信息，请参见 phy2sys(8)手册页 面。

11. 在 [ptp4l] 部分中，添加任何要传递给 ptp4l 的命令行选项（如果被 timemaster 调用 ）。本章记录了常用选项和更多信息，请参见 ptp4l(8)手册页 面。

12. 以 root 用户身份运行以下命令保存配置文件并重启 timemaster ：

```
~]# systemctl restart timemaster
```

20.11. 改进准确度

在以前的版本中，测试结果显示禁用无空循环内核功能可显著提高系统时钟的稳定性，从而提高 PTP 同步 准确性（这将增加功耗）。通过在内核引导选项参数中添加 nohz=off，可以禁用内核无盘模式。但是，最近应用于 kernel-3.10.0-197.el7 的改进大幅提升了系统时钟的稳定性和时钟稳定性，并且没有 nohz=off 的改进现在应该更小得多。

ptp4l 和 phc2sys 应用可以配置为使用新的自适应服务。与 PI servo 相比的优势在于，它不需要配置 PI 常量才能正常工作。要将它用于 ptp4l，请在 /etc/ptp4l.conf 文件中添加以下行：

```
clock_servo linreg
```

对 /etc/ptp4l.conf 进行更改后，以 root 用户身份运行以下命令从命令行重启 ptp4l 服务：

```
~]# systemctl restart ptp4l
```

要将它用于 phc2sys，请在 /etc/sysconfig/phc2sys 文件中添加以下行：

```
-E linreg
```

更改 /etc/sysconfig/phc2sys 后，以 root 用户身份运行以下命令从命令行重启 phc2sys 服务：

```
~]# systemctl restart phc2sys
```

20.12. 其它资源

以下信息来源提供了有关 PTP 和 ptp4l 工具的其他资源。

20.12.1. 安装的文档

- ptp4l(8)man page - 描述 ptp4l 选项，包括配置文件的格式。
- pMC(8)man page - 描述 PTP 管理客户端及其命令选项。
- pc2sys(8)手册页 - 描述将系统时钟与 PTP 硬件时钟(PHC)同步的工具。
-

timemaster(8)手册页 - 描述使用 ptp4l 和 phc2sys 使用 chronyd 或 ntpd 同步系统时钟的程序。

20.12.2. 有用的网站

http://www.nist.gov/el/isd/ieee/ieee1588.cfm

IEEE 1588 标准.

部分 VI. 监控和自动化

这部分描述了允许系统管理员监控系统性能、自动执行系统任务和报告错误的各种工具。

第 21 章 系统监控工具

要配置系统，系统管理员通常需要确定可用内存量、可用磁盘空间量、硬盘驱动器的分区方式或正在运行的进程。

21.1. 查看系统进程

21.1.1. 使用 ps 命令

ps 命令允许您显示正在运行的进程的信息。它生成一个静态列表，即，用于执行 命令时正在运行的快照。如果您希望持续更新正在运行的进程列表，请使用 top 命令或 系统监控器 应用程序。

要列出系统中运行的所有进程，包括其他用户拥有的进程，在 shell 提示符后输入以下内容：

```
ps ax
```

对于列出的每个进程，ps ax 命令显示进程 ID(PID)、与其关联的终端(TTY)、当前状态(STAT)、模拟 CPU 时间(TIME)，以及可执行文件的名称(COMMAND)。例如：

```
~]$ ps ax
PID TTY   STAT  TIME COMMAND
 1 ?    Ss   0:01 /usr/lib/systemd/systemd --switched-root --system --deserialize 23
 2 ?    S   0:00 [kthreadd]
 3 ?    S   0:00 [ksoftirqd/0]
 5 ?    S>   0:00 [kworker/0:0H]
[output truncated]
```

要显示每个进程的所有者，请使用以下命令：

```
ps aux
```

除 ps ax 命令提供的信息外，ps aux 显示进程所有者的有效用户名(USER)、CPU(%MEM)使用率、以 KB 为单位的虚拟内存大小(VSZ)、非交换的物理内存大小（以 KSS）(RSS)以及进程启动的时间或日期。例如：

```
~]$ ps aux
USER PID %CPU %MEM  VSZ  RSS TTY  STAT START  TIME COMMAND
root  1 0.3 0.3 134776 6840 ?   Ss  09:28  0:01 /usr/lib/systemd/systemd --switched-root --
system --d
root  2 0.0 0.0   0   0 ?   S  09:28  0:00 [kthreadd]
```

```
root  3 0.0 0.0   0   0 ?   S  09:28  0:00 [ksoftirqd/0]
root  5 0.0 0.0   0   0 ?   S>  09:28  0:00 [kworker/0:0H]
[output truncated]
```

您还可以将 ps 命令与 grep 结合使用来查看特定进程是否正在运行。例如，要确定 Emacs 是否 在运行，请输入：

```
~]$ ps ax | grep emacs
12056 pts/3  S+   0:00 emacs
12060 pts/2  S+   0:00 grep --color=auto emacs
```

有关可用命令行选项的完整列表，请查看 ps(1)手册页。

21.1.2. 使用 top 命令

top 命令显示系统上正在运行的进程的实时列表。它还显示有关系统正常运行时间、当前 CPU 和内存使用情况或正在运行的进程总数的附加信息，并允许您执行诸如排序列表或终止进程等操作。

要运行 top 命令，在 shell 提示符后输入以下内容：

```
top
```

对于列出的每个进程，top 命令显示进程 ID(PID)、进程所有者的有效用户名(USER)、优先级(PR)、nice 值(NI)、进程使用的虚拟内存量(VIRT)、进程使用的非交换物理内存量(RES)。进程使用的共享内存量(SHR)、进程状态字段 S、CPU 和内存(%MEM)使用量的百分比、模拟 CPU 时间(TIME+)，以及可执行文件的名称(COMMAND)。例如：

```
~]$ top
top - 16:42:12 up 13 min, 2 users, load average: 0.67, 0.31, 0.19
Tasks: 165 total,  2 running, 163 sleeping,  0 stopped,  0 zombie
%Cpu(s): 37.5 us, 3.0 sy, 0.0 ni, 59.5 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 1016800 total,  77368 free,  728936 used,  210496 buff/cache
KiB Swap:  839676 total,  776796 free,  62880 used.  122628 avail Mem

 PID USER   PR NI  VIRT  RES  SHR S %CPU %MEM   TIME+ COMMAND
3168 sjw    20  0 1454628 143240 15016 S 20.3 14.1  0:22.53 gnome-shell
4006 sjw    20  0 1367832 298876 27856 S 13.0 29.4  0:15.58 firefox
1683 root   20  0 242204 50464  4268 S 6.0 5.0  0:07.76 Xorg
4125 sjw    20  0 555148 19820 12644 S 1.3 1.9  0:00.48 gnome-terminal-
  10 root   20  0    0   0   0 S 0.3 0.0  0:00.39 rcu_sched
3091 sjw    20  0  37000  1468  904 S 0.3 0.1  0:00.31 dbus-daemon
3096 sjw    20  0 129688  2164  1492 S 0.3 0.2  0:00.14 at-spi2-registr
3925 root   20  0    0   0   0 S 0.3 0.0  0:00.05 kworker/0:0
   1 root   20  0 126568  3884  1052 S 0.0 0.4  0:01.61 systemd
```

```
  2 root   20  0    0   0   0 S 0.0 0.0  0:00.00 kthreadd
  3 root   20  0    0   0   0 S 0.0 0.0  0:00.00 ksoftirqd/0
  6 root   20  0    0   0   0 S 0.0 0.0  0:00.07 kworker/u2:0
[output truncated]
```

表 21.1 “交互式顶部命令” 包含有用的交互式命令，可用于 top。如需更多信息，请参阅前(1)man page。

表 21.1. 交互式顶部命令

命令	描述
输入,Space	立即刷新显示。
h	显示交互式命令的帮助屏幕.
H，是吗？	显示窗口和字段组的帮助屏幕.
k	终止进程.系统将提示您输入进程 ID 以及向其发送的信号。
n	更改显示的进程数.系统将提示您输入数字。
u	按用户对列表排序.
M	根据内存使用情况对列表排序.
P	根据 CPU 使用情况对列表排序.
q	终止 实用程序并返回到 shell 提示符。

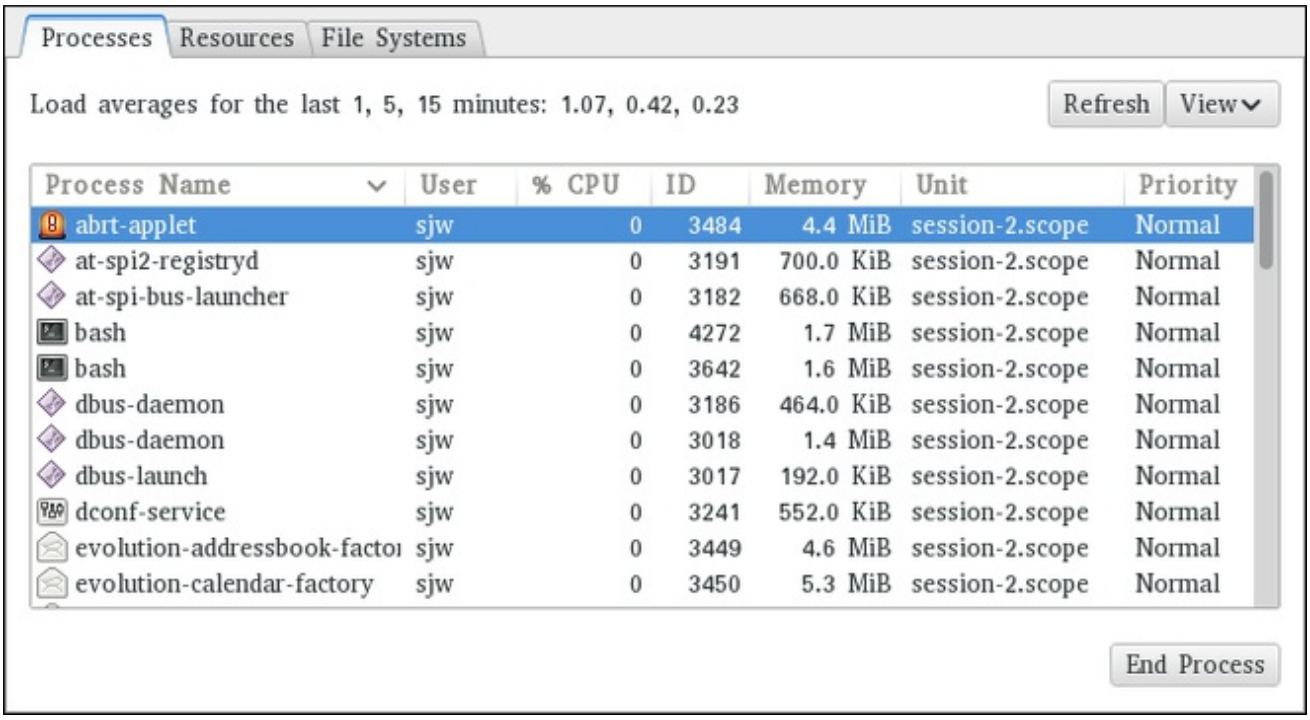
21.1.3. 使用系统监控器工具

系统监控器 工具的 Processes 选项卡允许您从图形用户界面查看、搜索、更改 的优先级和终止进程。

要从命令行启动 系统监控器 工具，请在 shell 提示符下键入 gnome-system-monitor。系统监控 器工具会出现。或者，如果使用 GNOME 桌面，按 Super 键进入"活动概览"，键入 "系统监控器"，然后按 Enter 键。系统监控 器工具会出现。Super 键显示在各种 guis 中，具体取决于键盘和其他硬件，但通常作为 Windows 或 Command 键（通常在 空格栏 的左侧）。

单击 Processes 选项卡，以查看正在运行的进程列表。

图 21.1. 系统监控器 - 进程



对于列出的每个进程，系统监控器 工具显示其名称(Process Name)、当前状态(Status)、CPU 使用率(% CPU)、nice 值(Nice)、进程ID(ID)、内存使用情况(Memory)、进程正在等待的频道（等待通道），以及会话的其他详细信息（会话）。若要按特定列按升序排列信息，可单击该列的名称。再次单击列的名称，将排序切换为升序和降序。

默认情况下，System Monitor 工具显示当前用户拥有的进程的列表。通过从 View 菜单中选择各种选项，您可以：

- 仅查看活动进程，
- 查看所有进程，
- 查看您的进程，
- 查看进程依赖关系，

另外，两个按钮可让您：

- 刷新进程列表，

- 从列表中选择进程，然后单击"结束进程 "按钮，以结束该进程。

21.2. 查看内存使用情况

21.2.1. 使用空闲命令

可用 命令允许您显示系统上的可用内存和已用内存量。要做到这一点，在 shell 提示符后输入以下内容：

```
free
```

可用 命令 提供有关物理内存(Mem)和交换空间(Swap)的信息。它显示内存总量（总内存量），以及正在使用的内存量（已用）、可用（可用）、共享（共享）、缓冲区和缓存(buff/cache)以及可用（可用）。例如：

```
~]$ free
      total    used    free   shared buff/cache  available
Mem:   1016800   727300    84684    3500   204816   124068
Swap:   839676    66920   772756
```

默认情况下，free 以 KB 为单位显示值。要以 MB 为单位显示值，请提供 -m 命令行选项：

```
free -m
```

例如：

```
~]$ free -m
      total    used    free   shared buff/cache  available
Mem:     992     711     81      3     200     120
Swap:     819     65     754
```

有关可用命令行选项的完整列表，请查看免费(1)手册页。

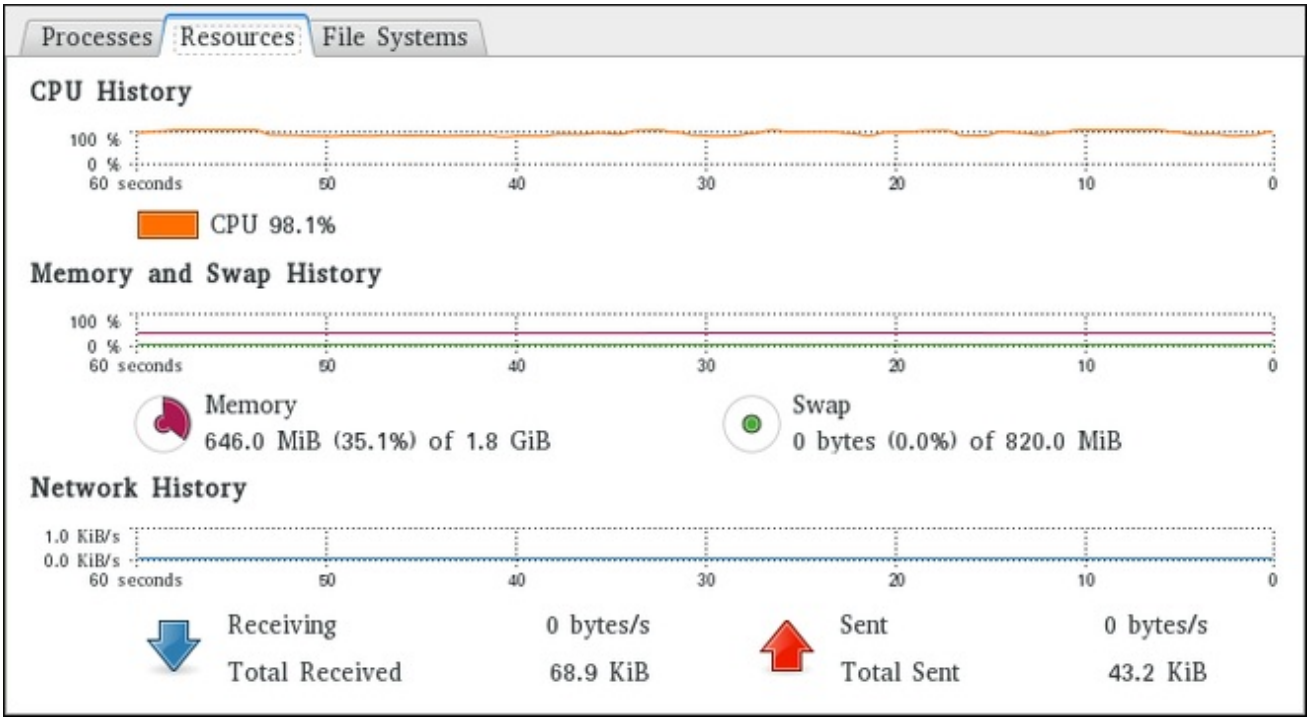
21.2.2. 使用系统监控器工具

通过系统监控器 工具的 Resources 选项卡，您可以查看系统上的可用和已用内存量。

要从命令行启动 系统监控器 工具，请在 shell 提示符下键入 gnome-system-monitor。系统监控 器工具会出现。或者，如果使用 GNOME 桌面，按 Super 键进入"活动概览"，键入 "系统监控器"，然后按 Enter 键。系统监控 器工具会出现。Super 键显示在各种 guis 中，具体取决于键盘和其他硬件，但通常作为 Windows 或 Command 键（通常在 空格栏 的左侧）。

单击 Resources 选项卡，以查看系统的内存使用情况。

图 21.2. 系统监控器 - 资源



在 Memory and Swap History 部分中，系统监控器 工具显示内存和交换使用历史记录的图形表示，以及物理内存（内存）和交换空间(Swap)的总容量，以及正在使用的内存量。

21.3. 查看 CPU 使用情况

21.3.1. 使用系统监控器工具

系统监控器 工具的 Resources 选项卡允许您查看系统上的当前 CPU 使用情况。

要从命令行启动 系统监控器 工具，请在 shell 提示符下键入 gnome-system-monitor。系统监控 器工具会出现。或者，如果使用 GNOME 桌面，按 Super 键进入"活动概览"，键入 "系统监控器"，然后按

Enter 键。系统监控器工具会出现。Super 键显示在各种 guis 中，具体取决于键盘和其他硬件，但通常作为 Windows 或 Command 键（通常在空格栏的左侧）。

单击 Resources 选项卡，以查看系统的 CPU 使用情况。

在 CPU History 部分中，系统监控器工具显示 CPU 使用历史记录的图形显示，并显示当前使用的 CPU 数量的百分比。

21.4. 查看块设备和文件系统

21.4.1. 使用 lsblk 命令

lsblk 命令允许您显示可用块设备的列表。它比 blkid 命令提供更多信息和更好的控制输出格式。它从 udev 读取信息，因此非root 用户可使用它。要显示块设备列表，在 shell 提示符后输入以下内容：

```
lsblk
```

对于每个列出的块设备，lsblk 命令显示设备名称(NAME)、主设备编号和次要设备号(MAJ:MIN)（如果设备是可移动的(RM)、其大小(SIZE)，如果设备是只读(RO)）、类型是什么(TYPE)，以及挂载该设备的位置(MOUNTPOINT)。例如：

```
~]$ lsblk
NAME          MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sr0            11:0  1 1024M 0 rom
vda            252:0  0  20G 0 rom
|-vda1         252:1  0  500M 0 part /boot
`-vda2         252:2  0 19.5G 0 part
 |-vg_kvm-lv_root (dm-0) 253:0  0  18G 0 lvm /
 `-vg_kvm-lv_swap (dm-1) 253:1  0  1.5G 0 lvm [SWAP]
```

默认情况下，lsblk 以类似于树的格式列出块设备。要将信息显示为普通列表，请添加 -l 命令行选项：

```
lsblk -l
```

例如：

```
~]$ lsblk -l
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sr0          11:0  1 1024M 0 rom
```

```
vda         252:0  0  20G 0 rom
vda1         252:1  0  500M 0 part /boot
vda2         252:2  0 19.5G 0 part
vg_kvm-lv_root (dm-0) 253:0  0  18G 0 lvm /
vg_kvm-lv_swap (dm-1) 253:1  0  1.5G 0 lvm [SWAP]
```

有关可用命令行选项的完整列表，请查看 lsblk(8)手册页。

21.4.2. 使用 blkid 命令

blkid 命令允许您显示有关可用块设备的低级别信息。它需要 root 特权，因此非root 用户应使用 lsblk 命令。要做到这一点，以 root 根用户身份在 shell 提示符后输入以下内容：

```
blkid
```

对于列出的每个块设备，blkid 命令显示可用属性，如其 通用唯一标识符(UUID)、文件系统类型 (TYPE)或卷标签(LABEL)。例如：

```
~]# blkid
/dev/vda1: UUID="7fa9c421-0054-4555-b0ca-b470a97a3d84" TYPE="ext4"
/dev/vda2: UUID="7IvYzk-TnnK-oPjf-ipdD-cofz-DXaJ-gPdgBW" TYPE="LVM2_member"
/dev/mapper/vg_kvm-lv_root: UUID="a07b967c-71a0-4925-ab02-aebcad2ae824" TYPE="ext4"
/dev/mapper/vg_kvm-lv_swap: UUID="d7ef54ca-9c41-4de4-ac1b-4193b0c1ddb6" TYPE="swap"
```

默认情况下，blkid 命令列出所有可用块设备。要只显示特定设备的信息，请在命令行中指定设备名称：

```
blkid device_name
```

例如，要显示 /dev/vda1 的信息，以 root 用户身份输入 ：

```
~]# blkid /dev/vda1
/dev/vda1: UUID="7fa9c421-0054-4555-b0ca-b470a97a3d84" TYPE="ext4"
```

您还可以将以上命令与 -p 和 -o udev 命令行选项一起使用来获取更多详细信息。请注意，运行这个命令需要 root 权限：

```
blkid -po udev device_name
```

例如：

```
~]# blkid -po udev /dev/vda1
ID_FS_UUID=7fa9c421-0054-4555-b0ca-b470a97a3d84
ID_FS_UUID_ENC=7fa9c421-0054-4555-b0ca-b470a97a3d84
ID_FS_VERSION=1.0
ID_FS_TYPE=ext4
ID_FS_USAGE=filesystem
```

有关可用命令行选项的完整列表，请查看 blkid(8)手册页。

21.4.3. 使用 findmnt 命令

findmnt 命令允许您显示当前挂载的文件系统列表。要做到这一点，在 shell 提示符后输入以下内容：

```
findmnt
```

对于每个列出的文件系统，findmnt 命令显示目标挂载点(TARGET)、源设备(SOURCE)、文件系统类型(FSTYPE)和相关挂载选项(OPTIONS)。例如：

```
~]$ findmnt
TARGET                SOURCE      FSTYPE     OPTIONS
/                     /dev/mapper/rhel-root
                                  xfs        rw,relatime,seclabel,attr2,inode64,noquota
├─/proc               proc        proc       rw,nosuid,nodev,noexec,relatime
│ ├─/proc/sys/fs/binfmt_misc  systemd-1  autofs
rw,relatime,fd=32,pgrp=1,timeout=300,minproto=5,maxproto=5,direct
│ └─/proc/fs/nfsd     sunrpc      nfsd       rw,relatime
├─/sys                sysfs       sysfs      rw,nosuid,nodev,noexec,relatime,seclabel
│ ├─/sys/kernel/security  securityfs  securityfs  rw,nosuid,nodev,noexec,relatime
│ ├─/sys/fs/cgroup      tmpfs       tmpfs      rw,nosuid,nodev,noexec,seclabel,mode=755
[output truncated]
```

默认情况下，findmnt 以类似于树形格式列出文件系统。要将信息显示为普通列表，请添加 -l 命令行选项：

```
findmnt -l
```

例如：

```
~]$ findmnt -l
```

```
TARGET          SOURCE       FSTYPE     OPTIONS
/proc           proc         proc       rw,nosuid,nodev,noexec,relatime
/sys            sysfs        sysfs      rw,nosuid,nodev,noexec,relatime,seclabel
/dev            devtmpfs     devtmpfs
rw,nosuid,seclabel,size=933372k,nr_inodes=233343,mode=755
/sys/kernel/security  securityfs  securityfs  rw,nosuid,nodev,noexec,relatime
/dev/shm        tmpfs        tmpfs      rw,nosuid,nodev,seclabel
/dev/pts        devpts       devpts
rw,nosuid,noexec,relatime,seclabel,gid=5,mode=620,ptmxmode=000
/run            tmpfs        tmpfs      rw,nosuid,nodev,seclabel,mode=755
/sys/fs/cgroup  tmpfs        tmpfs      rw,nosuid,nodev,noexec,seclabel,mode=755
[output truncated]
```

您也可以选择仅列出特定类型的文件系统。要做到这一点，请添加 -t 命令行选项后接一个文件系统类型：

```
findmnt -t type
```

例如：要列出所有 xfs 文件系统，请输入：

```
~]$ findmnt -t xfs
TARGET SOURCE      FSTYPE OPTIONS
/   /dev/mapper/rhel-root xfs  rw,relatime,seclabel,attr2,inode64,noquota
└─/boot /dev/vda1      xfs  rw,relatime,seclabel,attr2,inode64,noquota
```

有关可用命令行选项的完整列表，请查看 findmnt(8)手册页。

21.4.4. 使用 df 命令

The df 命令允许您显示系统磁盘空间使用情况的详细报告。要做到这一点，在 shell 提示符后输入以下内容：

```
df
```

对于每个列出的文件系统，df 命令显示其名称（文件系统）、大小（1K-blocks 或 Size）、使用空间量（已使用）、剩余空间（可用）、可用空间的百分比(Use%)，以及挂载的文件系统的位置（挂载）。例如：

```
~]$ df
Filesystem      1K-blocks   Used Available Use% Mounted on
/dev/mapper/vg_kvm-lv_root 18618236  4357360 13315112 25% /
tmpfs           380376    288  380088  1% /dev/shm
/dev/vda1       495844   77029  393215 17% /boot
```

默认情况下，df 命令以 1 KB 块为单位显示分区大小，以及已用和可用磁盘空间量（以 KB 为单位）。要查看以兆字节和千兆字节为单位的信息，请提供 -h 命令行选项，该选项可使 df 以人类可读格式显示这些值：

```
df -h
```

例如：

```
~]$ df -h
Filesystem        Size Used Avail Use% Mounted on
/dev/mapper/vg_kvm-lv_root  18G 4.2G  13G 25% /
tmpfs             372M 288K 372M  1% /dev/shm
/dev/vda1         485M  76M 384M 17% /boot
```

有关可用命令行选项的完整列表，请查看df(1)手册页。

21.4.5. 使用 du 命令

du 命令允许您显示目录中文件正在使用的空间量。要显示当前工作目录中每个子目录的磁盘使用情况，请在没有附加命令行选项的情况下运行命令：

```
du
```

例如：

```
~]$ du
14972  ./Downloads
4    ./.mozilla/extensions
4    ./.mozilla/plugins
12   ./.mozilla
15004  .
```

默认情况下，du 命令以 KB 为单位显示磁盘使用情况。要查看以兆字节和千兆字节为单位的信息，请提供 -h 命令行选项，该选项可使实用程序以人类可读格式显示这些值：

```
du -h
```

例如：

```
~]$ du -h
15M   ./Downloads
4.0K  ./.mozilla/extensions
4.0K  ./.mozilla/plugins
12K   ./.mozilla
15M   .
```

在列表的末尾，du 命令始终显示当前目录的最大总数。要只显示此信息，请提供 -s 命令行选项：

```
du -sh
```

例如：

```
~]$ du -sh
15M   .
```

有关可用命令行选项的完整列表，请查看 du(1)手册页。

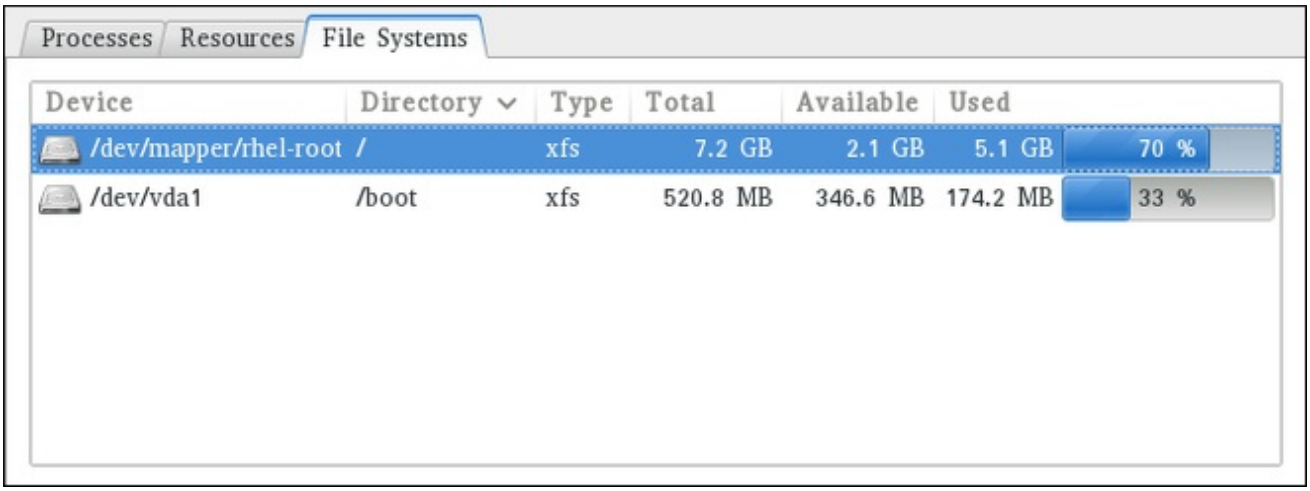
21.4.6. 使用系统监控器工具

系统监控器 工具 的文件系统选项卡 允许您在图形用户界面中查看文件系统和磁盘空间使用情况。

要从命令行启动 系统监控器 工具，请在 shell 提示符下键入 gnome-system-monitor。系统监控 器工具会出现。或者，如果使用 GNOME 桌面，按 Super 键进入"活动概览"，键入 "系统监控器"，然后按 Enter 键。系统监控 器工具会出现。Super 键显示在各种 guis 中，具体取决于键盘和其他硬件，但通常作为 Windows 或 Command 键（通常在 空格栏 的左侧）。

单击 File Systems 选项卡，以查看文件系统列表。

图 21.3. 系统监控器 - 文件系统



对于列出的每个文件系统，系统监控器工具会显示源设备（设备）、目标挂载点（目录）和文件系统类型（类型）及其大小（总大小）以及可用空间（可用）和使用的空间（已使用）。

21.5. 查看硬件信息

21.5.1. 使用 lspci 命令

lspci 命令允许您显示有关 PCI 总线及其关联的设备的信息。要列出系统中的所有 PCI 设备，在 shell 提示符后输入以下内容：

```
lspci
```

这会显示一个简单的设备列表，例如：

```
~]$ lspci
00:00.0 Host bridge: Intel Corporation 82X38/X48 Express DRAM Controller
00:01.0 PCI bridge: Intel Corporation 82X38/X48 Express Host-Primary PCI Express Bridge
00:1a.0 USB Controller: Intel Corporation 82801I (ICH9 Family) USB UHCI Controller #4 (rev
02)
00:1a.1 USB Controller: Intel Corporation 82801I (ICH9 Family) USB UHCI Controller #5 (rev
02)
00:1a.2 USB Controller: Intel Corporation 82801I (ICH9 Family) USB UHCI Controller #6 (rev
02)
[output truncated]
```

您还可以使用 -v 命令行选项显示更详细的输出，或使用 -vv 来获得非常详细的输出：

```
lspci -v|-vv
```

例如，要确定系统的显卡厂商、型号和内存大小，请输入：

```
~]$ lspci -v
[output truncated]

01:00.0 VGA compatible controller: nVidia Corporation G84 [Quadro FX 370] (rev a1) (prog-if
00 [VGA controller])
    Subsystem: nVidia Corporation Device 0491
    Physical Slot: 2
    Flags: bus master, fast devsel, latency 0, IRQ 16
    Memory at f2000000 (32-bit, non-prefetchable) [size=16M]
    Memory at e0000000 (64-bit, prefetchable) [size=256M]
    Memory at f0000000 (64-bit, non-prefetchable) [size=32M]
    I/O ports at 1100 [size=128]
    Expansion ROM at <unassigned> [disabled]
    Capabilities: <access denied>
    Kernel driver in use: nouveau
    Kernel modules: nouveau, nvidiafb

[output truncated]
```

有关可用命令行选项的完整列表，请查看 lspci(8)手册页。

21.5.2. 使用 lsusb 命令

lsusb 命令允许您显示有关 USB 总线和连接到它们的设备的相关信息。要列出系统中的所有 USB 设备，在 shell 提示符后输入以下内容：

```
lsusb
```

这会显示一个简单的设备列表，例如：

```
~]$ lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
[output truncated]
Bus 001 Device 002: ID 0bda:0151 Realtek Semiconductor Corp. Mass Storage Device
(Multicard Reader)
Bus 008 Device 002: ID 03f0:2c24 Hewlett-Packard Logitech M-UAL-96 Mouse
Bus 008 Device 003: ID 04b3:3025 IBM Corp.
```

您还可以使用 -v 命令行选项来显示更详细的输出：

```
lsusb -v
```

例如：

```
~]$ lsusb -v
[output truncated]

Bus 008 Device 002: ID 03f0:2c24 Hewlett-Packard Logitech M-UAL-96 Mouse
Device Descriptor:
 bLength         18
 bDescriptorType     1
 bcdUSB          2.00
 bDeviceClass      0 (Defined at Interface level)
 bDeviceSubClass     0
 bDeviceProtocol     0
 bMaxPacketSize0     8
 idVendor      0x03f0 Hewlett-Packard
 idProduct     0x2c24 Logitech M-UAL-96 Mouse
 bcdDevice       31.00
 iManufacturer       1
 iProduct        2
 iSerial         0
 bNumConfigurations    1
 Configuration Descriptor:
  bLength         9
  bDescriptorType     2
[output truncated]
```

有关可用命令行选项的完整列表，请查看 lsusb(8)手册页。

21.5.3. 使用 lscpu 命令

lscpu 命令允许您列出系统中存在的 CPU 的信息，包括 CPU 的数量、其架构、供应商、系列、型号、CPU 缓存等。要做到这一点，在 shell 提示符后输入以下内容：

```
lscpu
```

例如：

```
~]$ lscpu
Architecture:     x86_64
CPU op-mode(s):    32-bit, 64-bit
Byte Order:       Little Endian
CPU(s):         4
On-line CPU(s) list:  0-3
Thread(s) per core:  1
Core(s) per socket:  4
Socket(s):        1
```

```
NUMA node(s):     1
Vendor ID:        GenuineIntel
CPU family:       6
Model:            23
Stepping:         7
CPU MHz:          1998.000
BogoMIPS:         4999.98
Virtualization:   VT-x
L1d cache:        32K
L1i cache:        32K
L2 cache:         3072K
NUMA node0 CPU(s):   0-3
```

有关可用命令行选项的完整列表，请查看 lscpu(1)手册页。

21.6. 检查硬件错误

红帽企业 Linux 7 引入了新的硬件事件报告机制 (HERM.)这种机制收集系统报告的内存错误，以及错误检测和更正 (EDAC)机制报告的错误，用于双行内存模块(DIMM)，并将它们报告给用户空间。用户空间守护进程 rasdaemon 捕获和处理来自内核追踪机制的所有 可靠性、可用性和可维护性 (RAS)错误事件，并记录它们。以前由 edac-utils 提供的函数现在由 rasdaemon 替代。

要安装 install rasdaemon，以 root 用户身份输入以下命令：

```
~]# yum install rasdaemon
```

按如下所示启动服务：

```
~]# systemctl start rasdaemon
```

要使服务在系统启动时运行，请输入以下命令：

```
~]# systemctl enable rasdaemon
```

The ras-mc-ctl 实用程序提供了一种使用 EDAC 驱动程序的方法。输入以下命令查看命令选项列表：

```
~]$ ras-mc-ctl --help
Usage: ras-mc-ctl [OPTIONS...]
 --quiet      Quiet operation.
 --mainboard    Print mainboard vendor and model for this hardware.
 --status       Print status of EDAC drivers.
output truncated
```

要查看内存控制器事件摘要，以 root 用户身份运行：

```
~]# ras-mc-ctl --summary
Memory controller events summary:
   Corrected on DIMM Label(s): 'CPU_SrcID#0_Ha#0_Chan#0_DIMM#0' location: 0:0:0:-1
errors: 1

No PCIe AER errors.

No Extlog errors.
MCE records summary:
   1 MEMORY CONTROLLER RD_CHANNEL0_ERR Transaction: Memory read error errors
   2 No Error errors
```

要查看内存控制器报告的错误列表，以 root 用户身份运行：

```
~]# ras-mc-ctl --errors
Memory controller events:
1 3172-02-17 00:47:01 -0500 1 Corrected error(s): memory read error at
CPU_SrcID#0_Ha#0_Chan#0_DIMM#0 location: 0:0:0:-1, addr 65928, grain 7, syndrome 0
area:DRAM err_code:0001:0090 socket:0 ha:0 channel_mask:1 rank:0

No PCIe AER errors.

No Extlog errors.

MCE events:
1 3171-11-09 06:20:21 -0500 error: MEMORY CONTROLLER RD_CHANNEL0_ERR Transaction:
Memory read error, mcg mcgstatus=0, mci Corrected_error, n_errors=1, mcgcap=0x01000c16,
status=0x8c00004000010090, addr=0x1018893000, misc=0x15020a086, walltime=0x57e96780,
cpuid=0x00050663, bank=0x00000007
2 3205-06-22 00:13:41 -0400 error: No Error, mcg mcgstatus=0, mci Corrected_error
Error_enabled, mcgcap=0x01000c16, status=0x9400000000000000, addr=0x0000abcd,
walltime=0x57e967ea, cpuid=0x00050663, bank=0x00000001
3 3205-06-22 00:13:41 -0400 error: No Error, mcg mcgstatus=0, mci Corrected_error
Error_enabled, mcgcap=0x01000c16, status=0x9400000000000000, addr=0x00001234,
walltime=0x57e967ea, cpu=0x00000001, cpuid=0x00050663, apicid=0x00000002,
bank=0x00000002
```

这些命令在 ras-mc-ctl(8)man page 中进行了说明。

21.7. 使用 NET-SNMP 监控性能

红帽企业 Linux 7 包含 Net-SNMP 软件套件，其中包括灵活且可扩展的简单网络管理协议 (SNMP)代理。此代理及其相关实用程序可用于提供从大量系统到支持 SNMP 协议轮询的各种工具的性能数据。

本节介绍将 Net-SNMP 代理配置为通过网络安全地提供性能数据，使用 SNMP 协议检索数据，以及扩展 SNMP 代理以提供自定义性能指标。

21.7.1. 安装 Net-SNMP

Net-SNMP 软件套件可作为 Red Hat Enterprise Linux 软件分发包中的一组 RPM 软件包使用。表 21.2 “可用的 Net-SNMP 软件包” 总结了每个软件包及其内容。

表 21.2. 可用的 Net-SNMP 软件包

软件包	Provides
net-snmp	SNMP 代理后台程序和文档.导出性能数据时需要此软件包。
net-snmp-libs	nets **nmp** 库和捆绑 *的管理信息库(* MIBs)。导出性能数据时需要此软件包。
net-snmp-utils	SNMP 客户端，如 **snmpget** 和 **snmpwalk**.需要这个软件包才能通过 SNMP 查询系统性能数据。
net-snmp-perl	**mib2c** 程序和 **NetSNMP** Perl 模块。请注意，这个软件包由可选频道提供。有关红帽附加频道的详情，请查看 第 9.5.7 节 “添加 Optional 和 Supplementary 存储库”。
net-snmp-python	适用于 Python 的 SNMP 客户端库。请注意，这个软件包由可选频道提供。有关红帽附加频道的详情，请查看 第 9.5.7 节 “添加 Optional 和 Supplementary 存储库”。

要安装这些软件包，请使用以下格式的 yum 命令：

```
yum install package&hellip;
```

例如，要安装本节其余部分中使用的 SNMP 代理守护进程和 SNMP 客户端，以 root 用户身份在 shell 提示符后输入以下内容：

```
~]# yum install net-snmp net-snmp-libs net-snmp-utils
```

有关如何在 Red Hat Enterprise Linux 中安装新软件包的详情请参考 第 9.2.4 节 “安装软件包”。

21.7.2. 运行 Net-SNMP 守护进程

net-snmp 软件包包含 snmpd，即 SNMP 代理守护进程。本节介绍如何启动、停止和重新启动 snmpd 服务。有关在 Red Hat Enterprise Linux 7 中管理系统服务的详情请参考 第 10 章 使用 systemd 管理服务。

21.7.2.1. 启动服务

要在当前会话中运行 snmpd 服务，以 root 用户身份在 shell 提示符后输入以下内容：

```
systemctl start snmpd.service
```

要将服务配置为在引导时自动启动，请使用以下命令：

```
systemctl enable snmpd.service
```

21.7.2.2. 停止服务

要停止正在运行的 snmpd 服务，以 root 用户身份在 shell 提示符后输入以下内容：

```
systemctl stop snmpd.service
```

要禁用在引导时启动服务，请使用以下命令：

```
systemctl disable snmpd.service
```

21.7.2.3. 重启服务

要重启正在运行的 snmpd 服务，在 shell 提示符后输入以下内容：

```
systemctl restart snmpd.service
```

此命令将停止服务，并快速启动该服务。要在不停止服务的情况下重新载入配置，请运行以下命令：

```
systemctl reload snmpd.service
```

这会导致正在运行的 snmpd 服务重新加载其配置。

21.7.3. 配置 Net-SNMP

要更改 Net-SNMP 代理守护进程配置，请编辑 /etc/snmp/snmpd.conf 配置文件。Red Hat Enterprise Linux 7 中包含的默认 snmpd.conf 文件会受到大量注释，它可作为代理配置的良好起点。

本节重点介绍两种常见任务：设置系统信息和配置身份验证。有关可用配置指令的更多信息，请参阅 snmpd.conf(5)手册页。此外，net -snmp 软件包中也有一个名为 snmp conf 的实用程序，可用于以交互方式生成有效的代理配置。

请注意，必须安装 net-snmp-utils 软件包，才能使用本节中描述的 snmpwalk 实用程序。

注意

要将配置文件的任何更改生效，请以 root 用户身份运行以下命令来强制 snmpd 服务重新读取配置：

```
systemctl reload snmpd.service
```

21.7.3.1. 设置系统信息

net-SNMP 通过系统树提供一些基本 系统 信息。例如，以下 snmpwalk 命令显示具有默认代理配置的 系统 树：

```
~]# snmpwalk -v2c -c public localhost system
SNMPv2-MIB::sysDescr.0 = STRING: Linux localhost.localdomain 3.10.0-123.el7.x86_64 #1
SMP Mon May 5 11:16:57 EDT 2014 x86_64
SNMPv2-MIB::sysObjectID.0 = OID: NET-SNMP-MIB::netSnmpAgentOIDs.10
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (464) 0:00:04.64
SNMPv2-MIB::sysContact.0 = STRING: Root <root@localhost> (configure
/etc/snmp/snmp.local.conf)
[output truncated]
```

默认情况下，sysName 对象被设置为主机名。通过更改 sys location 和 sys contact 指令的值，可以在 /etc/snmp/snmpd.conf 文件中配置 sys Location 和 sys Contact 对象，例如：

```
syslocation Datacenter, Row 4, Rack 3
syscontact UNIX Admin <admin@example.com>
```

更改配置文件后，重新载入配置并再次运行 snmpwalk 命令测试它：

```
~]# systemctl reload snmp.service
~]# snmpwalk -v2c -c public localhost system
SNMPv2-MIB::sysDescr.0 = STRING: Linux localhost.localdomain 3.10.0-123.el7.x86_64 #1
SMP Mon May 5 11:16:57 EDT 2014 x86_64
SNMPv2-MIB::sysObjectID.0 = OID: NET-SNMP-MIB::netSnmpAgentOIDs.10
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (35424) 0:05:54.24
SNMPv2-MIB::sysContact.0 = STRING: UNIX Admin <admin@example.com>
SNMPv2-MIB::sysName.0 = STRING: localhost.localdomain
SNMPv2-MIB::sysLocation.0 = STRING: Datacenter, Row 4, Rack 3
[output truncated]
```

21.7.3.2. 配置身份验证

Net-SNMP 代理守护进程支持所有三种版本的 SNMP 协议。前两个版本（1 和 2c）提供使用社区字符串的简单身份验证。此字符串是代理和任何客户端实用程序之间的共享 secret。字符串通过网络以明文形式传递，但被视为不安全。SNMP 协议的版本 3 支持使用各种协议进行用户身份验证和消息加密。Net-SNMP 代理还支持通过 SSH 隧道以及 X.509 证书的 TLS 身份验证。

配置 SNMP 版本 2c 社区

要配置 SNMP 版本 2c 社区，请在 /etc/snmp/snmpd.conf 配置文件中使用 rocommunity 或 rwcommunity 指令。指令的格式如下：

```
directive community source OID
```

…其中社区是要使用的社区字符串，source 是 IP 地址或子网，OID 则是 SNMP 树，用于提供对.例如，以下指令提供了对系统树的只读访问，使用本地机器上的社区字符串"redhat"：

```
rocommunity redhat 127.0.0.1 .1.3.6.1.2.1.1
```

要测试配置，请使用 snmpwalk 命令及 -v 和 -c 选项。

```
~]# snmpwalk -v2c -c redhat localhost system
SNMPv2-MIB::sysDescr.0 = STRING: Linux localhost.localdomain 3.10.0-123.el7.x86_64 #1
SMP Mon May 5 11:16:57 EDT 2014 x86_64
SNMPv2-MIB::sysObjectID.0 = OID: NET-SNMP-MIB::netSnmpAgentOIDs.10
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (101376) 0:16:53.76
SNMPv2-MIB::sysContact.0 = STRING: UNIX Admin <admin@example.com>
SNMPv2-MIB::sysName.0 = STRING: localhost.localdomain
SNMPv2-MIB::sysLocation.0 = STRING: Datacenter, Row 4, Rack 3
[output truncated]
```

配置 SNMP 版本 3 用户

要配置 SNMP 版本 3 用户，请使用 net-snmp-create-v3-user 命令。此命令向 /var/lib/net-snmp/snmpd.conf 和 /etc/snmp/snmpd.conf 文件添加条目，这些文件创建用户并为用户授予访问权限。请注意，net-snmp-create-v3-user 命令只能在代理未运行时运行。以下示例创建"admin"用户，其密码为 "redhatsnmp:

```
~]# systemctl stop snmpd.service
~]# net-snmp-create-v3-user
Enter a SNMPv3 user name to create:
admin
Enter authentication pass-phrase:
redhatsnmp
Enter encryption pass-phrase:
 [press return to reuse the authentication pass-phrase]

adding the following line to /var/lib/net-snmp/snmpd.conf:
  createUser admin MD5 "redhatsnmp" DES
adding the following line to /etc/snmp/snmpd.conf:
  rwuser admin
~]# systemctl start snmpd.service
```

rwuser 指令（或者提供 -ro 命令行选项时 rouser），net -snmp-create-v3-user 添加到 /etc/snmp/snmpd.conf 具有与 rwcommunity 和 rocommunity 指令类似的格式：

```
directive user noauth|auth|priv OID
```

... 其中 user 是用户名，OID 是 SNMP 树，用于提供对.默认情况下，Net-SNMP 代理守护进程仅允许经过身份验证的请求（ auth 选项）。noauth 选项允许您允许未经身份验证的请求，而 priv 选项则强制使用加密。authpriv 选项指定请求必须经过身份验证并且应加密回复。

例如，以下行授予用户"admin"对整个树的读写访问权限：

```
rwuser admin authpriv .1
```

要测试配置，请在用户的主目录中创建一个 .snmp/ 目录，并在该目录中创建一个名为 snmp.conf 的配置文件(~/.snmp/snmp.conf)：

```
defVersion 3
defSecurityLevel authPriv
defSecurityName admin
defPassphrase redhatsnmp
```

现在，在查询代理时，s nmpwalk 命令将使用这些身份验证设置：

```
~]$ snmpwalk -v3 localhost system
SNMPv2-MIB::sysDescr.0 = STRING: Linux localhost.localdomain 3.10.0-123.el7.x86_64 #1
SMP Mon May 5 11:16:57 EDT 2014 x86_64
[output truncated]
```

21.7.4. 通过 SNMP 检索性能数据

红帽企业 Linux 中的 Net-SNMP 代理通过 SNMP 协议提供各种性能信息。此外，可以查询代理查找系统上已安装的 RPM 软件包列表、系统上当前运行的进程的列表或系统的网络配置。

本节概述 SNMP 上可用的与性能调优相关的 OID。它假定已安装 net-snmp-utils 软件包，并授予该用户 SNMP 树的访问权限，如 第 21.7.3.2 节 “配置身份验证” 所述。

21.7.4.1. 硬件配置

Net-SNMP 中包含 的主机资源 MIB 提供了有关主机当前硬件和软件配置到客户端实用程序的信息。表 21.3 “可用 OID” 总结了该 MIB 下提供的不同 OID。

表 21.3. 可用 OID

OID	描述
HOST-RESOURCES-MIB::hrSystem	包含常规系统信息，如正常运行时间、用户数和运行的进程数。
HOST-RESOURCES-MIB::hrStorage	包含有关内存和文件系统使用情况的数据。
HOST-RESOURCES-MIB::hrDevices	包含所有处理器、网络设备和文件系统的列表。
HOST-RESOURCES-MIB::hrSWRun	包含所有正在运行的进程的列表。
HOST-RESOURCES-MIB::hrSWRunPerf	包含 HOST-RESOURCES-MIB::hrSWRun 进程表的内存和 CPU 统计信息。
HOST-RESOURCES-MIB::hrSWInstalled	包含 RPM 数据库的列表。

主机资源 MIB 中也提供了多个 SNMP 表，可用于检索可用信息的摘要。以下示例显示 HOST-RESOURCES-MIB::hrFSTable:

```
~]$ snmptable -Cb localhost HOST-RESOURCES-MIB::hrFSTable
SNMP table: HOST-RESOURCES-MIB::hrFSTable

 Index MountPoint RemoteMountPoint              Type
```

```
 Access Bootable StorageIndex LastFullBackupDate LastPartialBackupDate
  1    "/"        "" HOST-RESOURCES-TYPES::hrFSLinuxExt2
readWrite   true      31   0-1-1,0:0:0.0     0-1-1,0:0:0.0
  5 "/dev/shm"        ""   HOST-RESOURCES-TYPES::hrFSOther
readWrite  false      35   0-1-1,0:0:0.0     0-1-1,0:0:0.0
  6  "/boot"        "" HOST-RESOURCES-TYPES::hrFSLinuxExt2
readWrite  false      36   0-1-1,0:0:0.0     0-1-1,0:0:0.0
```

有关 HOST-RESOURCES-MIB 的更多信息，请参阅 /usr/share/snmp/mibs/HOST-RESOURCES-MIB.txt 文件。

21.7.4.2. CPU 和内存信息

大多数系统性能数据在 UCD SNMP MIB 中可用。systemStats OID 围绕处理器使用情况提供多个计数器：

```
~]$ snmpwalk localhost UCD-SNMP-MIB::systemStats
UCD-SNMP-MIB::ssIndex.0 = INTEGER: 1
UCD-SNMP-MIB::ssErrorName.0 = STRING: systemStats
UCD-SNMP-MIB::ssSwapIn.0 = INTEGER: 0 kB
UCD-SNMP-MIB::ssSwapOut.0 = INTEGER: 0 kB
UCD-SNMP-MIB::ssIOSent.0 = INTEGER: 0 blocks/s
UCD-SNMP-MIB::ssIOReceive.0 = INTEGER: 0 blocks/s
UCD-SNMP-MIB::ssSysInterrupts.0 = INTEGER: 29 interrupts/s
UCD-SNMP-MIB::ssSysContext.0 = INTEGER: 18 switches/s
UCD-SNMP-MIB::ssCpuUser.0 = INTEGER: 0
UCD-SNMP-MIB::ssCpuSystem.0 = INTEGER: 0
UCD-SNMP-MIB::ssCpuIdle.0 = INTEGER: 99
UCD-SNMP-MIB::ssCpuRawUser.0 = Counter32: 2278
UCD-SNMP-MIB::ssCpuRawNice.0 = Counter32: 1395
UCD-SNMP-MIB::ssCpuRawSystem.0 = Counter32: 6826
UCD-SNMP-MIB::ssCpuRawIdle.0 = Counter32: 3383736
UCD-SNMP-MIB::ssCpuRawWait.0 = Counter32: 7629
UCD-SNMP-MIB::ssCpuRawKernel.0 = Counter32: 0
UCD-SNMP-MIB::ssCpuRawInterrupt.0 = Counter32: 434
UCD-SNMP-MIB::ssIORawSent.0 = Counter32: 266770
UCD-SNMP-MIB::ssIORawReceived.0 = Counter32: 427302
UCD-SNMP-MIB::ssRawInterrupts.0 = Counter32: 743442
UCD-SNMP-MIB::ssRawContexts.0 = Counter32: 718557
UCD-SNMP-MIB::ssCpuRawSoftIRQ.0 = Counter32: 128
UCD-SNMP-MIB::ssRawSwapIn.0 = Counter32: 0
UCD-SNMP-MIB::ssRawSwapOut.0 = Counter32: 0
```

特别是，ssCpuRawUser、ssCpuRawSystem、ssCpuRawWait 和 ssCpuRawIdle OID 提供计数器，它们有助于确定系统是否在内核空间、用户空间或 I/O 中消耗大部分处理器时间。ssRawSwapIn 和 ssRawSwapOut 在确定系统是否存在内存耗尽时非常有用。

UCD-SNMP-MIB::memory OID 下提供了更多内存信息，它提供与 可用 命令类似的数据：

```
~]$ snmpwalk localhost UCD-SNMP-MIB::memory
UCD-SNMP-MIB::memIndex.0 = INTEGER: 0
UCD-SNMP-MIB::memErrorName.0 = STRING: swap
UCD-SNMP-MIB::memTotalSwap.0 = INTEGER: 1023992 kB
UCD-SNMP-MIB::memAvailSwap.0 = INTEGER: 1023992 kB
UCD-SNMP-MIB::memTotalReal.0 = INTEGER: 1021588 kB
UCD-SNMP-MIB::memAvailReal.0 = INTEGER: 634260 kB
UCD-SNMP-MIB::memTotalFree.0 = INTEGER: 1658252 kB
UCD-SNMP-MIB::memMinimumSwap.0 = INTEGER: 16000 kB
UCD-SNMP-MIB::memBuffer.0 = INTEGER: 30760 kB
UCD-SNMP-MIB::memCached.0 = INTEGER: 216200 kB
UCD-SNMP-MIB::memSwapError.0 = INTEGER: noError(0)
UCD-SNMP-MIB::memSwapErrorMsg.0 = STRING:
```

UCD SNMP MIB 中 也提供负载平均值。SNMP 表 UCD-SNMP-MIB::laTable 包含 1、5 和 15 分钟负载平均值列表：

```
~]$ snmptable localhost UCD-SNMP-MIB::laTable
SNMP table: UCD-SNMP-MIB::laTable

 laIndex laNames laLoad laConfig laLoadInt laLoadFloat laErrorFlag laErrMessage
    1 Load-1  0.00  12.00     0  0.000000   noError
    2 Load-5  0.00  12.00     0  0.000000   noError
    3 Load-15  0.00  12.00     0  0.000000   noError
```

21.7.4.3. 文件系统和磁盘信息

主机资源 MIB 提供有关文件系统大小和使用情况的信息。每个文件系统（以及每个内存池）在 HOST-RESOURCES-MIB::hrStorageTable 表中都有一个条目：

```
~]$ snmptable -Cb localhost HOST-RESOURCES-MIB::hrStorageTable
SNMP table: HOST-RESOURCES-MIB::hrStorageTable

 Index                   Type      Descr
AllocationUnits  Size  Used AllocationFailures
   1      HOST-RESOURCES-TYPES::hrStorageRam Physical memory
1024 Bytes 1021588 388064         ?
   3 HOST-RESOURCES-TYPES::hrStorageVirtualMemory Virtual memory
1024 Bytes 2045580 388064         ?
   6     HOST-RESOURCES-TYPES::hrStorageOther Memory buffers
1024 Bytes 1021588 31048         ?
   7     HOST-RESOURCES-TYPES::hrStorageOther  Cached memory
1024 Bytes 216604 216604         ?
  10 HOST-RESOURCES-TYPES::hrStorageVirtualMemory   Swap space
1024 Bytes 1023992   0         ?
  31   HOST-RESOURCES-TYPES::hrStorageFixedDisk        /
4096 Bytes 2277614 250391         ?
```

```
 35   HOST-RESOURCES-TYPES::hrStorageFixedDisk    /dev/shm
4096 Bytes 127698   0        ?
 36   HOST-RESOURCES-TYPES::hrStorageFixedDisk     /boot
1024 Bytes 198337 26694        ?
```

HOST-RESOURCES-MIB::hrStorageSize 和 HOST-RESOURCES-MIB::hrStorageUsed 下的 OID 可以用来计算每个挂载的文件系统的剩余容量。

UCD-SNMP-MIB::systemStats（ss lORawSent.0 和 ssIORawRecieved.0 ）和 UCD-DISKIO-MIB::diskIOTable 中都可以使用 I/O 数据。后者提供了更精细的数据。在此表中是 diskIONReadX 和 diskIONWrittenX 的 OID，它们为系统引导后读取并写入块设备的字节数提供计数器：

```
~]$ snmptable -Cb localhost UCD-DISKIO-MIB::diskIOTable
SNMP table: UCD-DISKIO-MIB::diskIOTable

 Index Device   NRead NWritten Reads Writes LA1 LA5 LA15  NReadX NWrittenX
...
  25  sda 216886272 139109376 16409  4894  ?  ?  ? 216886272 139109376
  26  sda1  2455552   5120  613   2  ?  ?  ?  2455552   5120
  27  sda2  1486848     0  332   0  ?  ?  ?  1486848     0
  28  sda3 212321280 139104256 15312  4871  ?  ?  ? 212321280 139104256
```

21.7.4.4. 网络信息

Interfaces MIB 提供有关网络设备的信息。IF-MIB::ifTable 提供 SNMP 表，其中包含系统上每个接口的条目、接口的配置以及接口的各种数据包计数器。以下示例显示了在有两个物理网络接口的系统中 ifTable 的前几列：

```
~]$ snmptable -Cb localhost IF-MIB::ifTable
SNMP table: IF-MIB::ifTable

 Index Descr       Type  Mtu  Speed   PhysAddress AdminStatus
   1  lo softwareLoopback 16436 10000000              up
   2 eth0  ethernetCsmacd 1500    0 52:54:0:c7:69:58     up
   3 eth1  ethernetCsmacd 1500    0 52:54:0:a7:a3:24    down
```

网络流量位于 OIDs IF-MIB::ifOutOctets 和 IF-MIB::ifInOctets 下。以下 SNMP 查询将检索此系统中每个接口的网络流量：

```
~]$ snmpwalk localhost IF-MIB::ifDescr
IF-MIB::ifDescr.1 = STRING: lo
IF-MIB::ifDescr.2 = STRING: eth0
IF-MIB::ifDescr.3 = STRING: eth1
~]$ snmpwalk localhost IF-MIB::ifOutOctets
IF-MIB::ifOutOctets.1 = Counter32: 10060699
IF-MIB::ifOutOctets.2 = Counter32: 650
```

```
IF-MIB::ifOutOctets.3 = Counter32: 0
~]$ snmpwalk localhost IF-MIB::ifInOctets
IF-MIB::ifInOctets.1 = Counter32: 10060699
IF-MIB::ifInOctets.2 = Counter32: 78650
IF-MIB::ifInOctets.3 = Counter32: 0
```

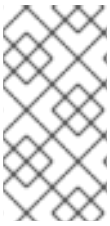
21.7.5. 扩展 Net-SNMP

可以扩展 Net-SNMP 代理，以提供除原始系统指标外的应用指标。这可实现容量规划以及性能问题故障排除。例如，了解电子邮件系统的负载平均值为 15 分钟，而测试期间的负载平均值为 15，但知道电子邮件系统的负载平均值为 15 次，处理第 15 条消息会很有帮助。当应用程序指标通过与系统指标相同的接口获得时，也可以视觉化了解不同负载场景对系统性能的影响（例如，每条额外的 10,000 个消息都会线性增大负载平均值到 100,000）。

红帽企业 Linux 中包含的许多应用程序扩展了 Net-SNMP 代理，以通过 SNMP 提供应用指标。另外，也可以为自定义应用程序扩展代理。本节论述了使用 shell 脚本和可选频道中的 Perl 插件扩展代理。它假设安装了 net-snmp-utils 和 net-snmp-perl 软件包，并授予该用户对 SNMP 树的访问权限，如第 21.7.3.2 节 “配置身份验证” 所述。

21.7.5.1. 使用 Shell 脚本扩展 Net-SNMP

Net-SNMP 代理提供一个扩展 MIB(NET-SNMP-EXTEND-MIB)，可用于查询任意 shell 脚本。要指定要运行的 shell 脚本，请使用 /etc/snmp/snmpd.conf 文件中的 extend 指令。定义后，代理将通过 SNMP 提供 命令的退出代码和任何输出。以下示例通过脚本演示此机制，此脚本决定了进程表中的 httpd 进程数。



注意

Net-SNMP 代理还提供内置机制，用于通过 proc 指令检查流程表。如需更多信息，请参阅 snmpd.conf(5)手册页。

以下 shell 脚本的退出代码是在给定时间点运行在系统上运行的 httpd 进程数：

```
#!/bin/sh

NUMPIDS=pgrep httpd | wc -l

exit $NUMPIDS
```

要使该脚本可通过 SNMP 使用，请将 脚本复制到系统路径上的位置，设置可执行位，并将 扩展 指令添加到 /etc/snmp/snmpd.conf 文件。扩展 指令的格式如下：

```
extend name prog args
```

… 其中 name 是扩展的标识字符串，prog 是要运行的程序，args 则是提供程序的参数。例如，如果将上述 shell 脚本复制到 /usr/local/bin/check_apache.sh 中，以下指令会将脚本添加到 SNMP 树中：

```
extend httpd_pids /bin/sh /usr/local/bin/check_apache.sh
```

然后，可以在 NET-SNMP-EXTEND-MIB::nsExtendObjects 中查询该脚本：

```
~]$ snmpwalk localhost NET-SNMP-EXTEND-MIB::nsExtendObjects
NET-SNMP-EXTEND-MIB::nsExtendNumEntries.0 = INTEGER: 1
NET-SNMP-EXTEND-MIB::nsExtendCommand."httpd_pids" = STRING: /bin/sh
NET-SNMP-EXTEND-MIB::nsExtendArgs."httpd_pids" = STRING:
/usr/local/bin/check_apache.sh
NET-SNMP-EXTEND-MIB::nsExtendInput."httpd_pids" = STRING:
NET-SNMP-EXTEND-MIB::nsExtendCacheTime."httpd_pids" = INTEGER: 5
NET-SNMP-EXTEND-MIB::nsExtendExecType."httpd_pids" = INTEGER: exec(1)
NET-SNMP-EXTEND-MIB::nsExtendRunType."httpd_pids" = INTEGER: run-on-read(1)
NET-SNMP-EXTEND-MIB::nsExtendStorage."httpd_pids" = INTEGER: permanent(4)
NET-SNMP-EXTEND-MIB::nsExtendStatus."httpd_pids" = INTEGER: active(1)
NET-SNMP-EXTEND-MIB::nsExtendOutput1Line."httpd_pids" = STRING:
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."httpd_pids" = STRING:
NET-SNMP-EXTEND-MIB::nsExtendOutNumLines."httpd_pids" = INTEGER: 1
NET-SNMP-EXTEND-MIB::nsExtendResult."httpd_pids" = INTEGER: 8
NET-SNMP-EXTEND-MIB::nsExtendOutLine."httpd_pids".1 = STRING:
```

请注意，退出代码（本示例中为"8"）作为 INTEGER 类型提供，任何输出都以 STRING 类型提供。若要以整数形式公开多个指标，请使用 extend 指令向 脚本提供不同的参数。例如，以下 shell 脚本可用于确定匹配任意字符串的进程数量，同时还会输出一个文本字符串，给出的进程数量：

```
#!/bin/sh

PATTERN=$1
NUMPIDS=pgrep $PATTERN | wc -l

echo "There are $NUMPIDS $PATTERN processes."
exit $NUMPIDS
```

当以上脚本复制到 / usr/local/bin/proc.sh 时，以下 /etc/ s nmpd.conf 指令将同时提供 httpd PID 的数量以及 snmpd PID 的数量：

```
extend httpd_pids /bin/sh /usr/local/bin/check_proc.sh httpd
extend snmpd_pids /bin/sh /usr/local/bin/check_proc.sh snmpd
```

以下示例显示了 then sExtendObjects OID 的 snmpwalk 的输出结果：

```
~]$ snmpwalk localhost NET-SNMP-EXTEND-MIB::nsExtendObjects
NET-SNMP-EXTEND-MIB::nsExtendNumEntries.0 = INTEGER: 2
NET-SNMP-EXTEND-MIB::nsExtendCommand."httpd_pids" = STRING: /bin/sh
NET-SNMP-EXTEND-MIB::nsExtendCommand."snmpd_pids" = STRING: /bin/sh
NET-SNMP-EXTEND-MIB::nsExtendArgs."httpd_pids" = STRING: /usr/local/bin/check_proc.sh
httpd
NET-SNMP-EXTEND-MIB::nsExtendArgs."snmpd_pids" = STRING:
/usr/local/bin/check_proc.sh snmpd
NET-SNMP-EXTEND-MIB::nsExtendInput."httpd_pids" = STRING:
NET-SNMP-EXTEND-MIB::nsExtendInput."snmpd_pids" = STRING:
...
NET-SNMP-EXTEND-MIB::nsExtendResult."httpd_pids" = INTEGER: 8
NET-SNMP-EXTEND-MIB::nsExtendResult."snmpd_pids" = INTEGER: 1
NET-SNMP-EXTEND-MIB::nsExtendOutLine."httpd_pids".1 = STRING: There are 8 httpd
processes.
NET-SNMP-EXTEND-MIB::nsExtendOutLine."snmpd_pids".1 = STRING: There are 1 snmpd
processes.
```

警告

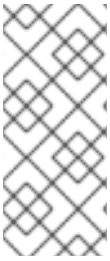
整数退出代码限制为 0-255 范围。对于可能超过 256 的值，请使用脚本的标准输出（将键入为字符串）或不同的扩展代理方法。

最后一个示例显示了对系统的可用内存和 httpd 进程数量的查询。此查询可用于性能测试，以确定进程数量对内存压力的影响：

```
~]$ snmpget localhost \
  'NET-SNMP-EXTEND-MIB::nsExtendResult."httpd_pids"' \
  UCD-SNMP-MIB::memAvailReal.0
NET-SNMP-EXTEND-MIB::nsExtendResult."httpd_pids" = INTEGER: 8
UCD-SNMP-MIB::memAvailReal.0 = INTEGER: 799664 kB
```

21.7.5.2. 使用 Perl 扩展 Net-SNMP

使用 extend 指令执行 shell 脚本是一种相当受限的方法，用于通过 SNMP 公开自定义应用指标。Net-SNMP 代理还提供嵌入式 Perl 接口来公开自定义对象。Optional 频道中的 net-snmp-perl 软件包提供 NetSNMP::agent Perl 模块，用于在 Red Hat Enterprise Linux 中编写嵌入式 Perl 插件。



注意

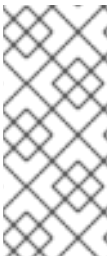
在订阅 Optional 和 Supplementary 频道前，请查看覆盖范围详情。如果您决定从这些频道安装软件包，请按照红帽客户门户网站中名为 How to access Optional 和 Supplementary 频道以及 -devel 软件包(RHSM)中的步骤进行操作。

NetSNMP::agent Perl 模块提供了一个 代理 对象，用于处理代理 OID 树的一部分的请求。agent 对象的构造器具有作为 snmpd 或独立代理的子代理运行代理的选项。创建嵌入式代理不需要任何参数：

```
use NetSNMP::agent (':all');

my $agent = new NetSNMP::agent();
```

代理 对象具有一个 寄存器 方法，用于使用特定的 OID 注册回调函数。register 函数将名称 OID 和指针用于回调函数。以下示例将使用 SNMP Agent 注册名为 hello_handler 的回调函数，该代理将处理 OID .1.3.6.1.4.1.8072.9999.9999 下的请求：

```
$agent->register("hello_world", ".1.3.6.1.4.1.8072.9999.9999",
        \&hello_handler);
```



注意

OID .1.3.6.1.4.1.8072.9999.9999 (NET-SNMP-MIB::netSnmpPlaypen)通常仅用于演示目的。如果您的组织还没有根 OID，可以通过联系美国 ISO 名称注册机构(ANSI)来获取一个。

该处理程序功能将通过四个参数调用：HANDLER、REGISTRATION_INFO、REQUEST_INFO 和 REQUESTS。REQUESTS 参数包含当前调用中的请求列表，应迭代并使用数据填充。列表中 的请求 对象具有 get 和 set 方法，允许操作请求的 OID 和 值。例如，以下调用会将请求对象的值设置为字符串 "hello world"：

```
$request->setValue(ASN_OCTET_STR, "hello world");
```

处理程序函数应响应两种类型的 SNMP 请求：GET 请求和 GETNEXT 请求。请求类型通过调用作为第三个参数传递给处理程序函数的 request_info 对象的 getMode 方法来确定。如果请求是 GET 请求，调用者将预期处理程序设置 请求 对象 的值，具体取决于请求的 OID。如果请求是 GETNEXT 请求，调用者还将请求的 OID 设置为树中下一个可用的 OID。下面的代码示例演示了这种情况：

```
my $request;
my $string_value = "hello world";
my $integer_value = "8675309";
```

```
for($request = $requests; $request; $request = $request->next()) {
 my $oid = $request->getOID();
 if ($request_info->getMode() == MODE_GET) {
  if ($oid == new NetSNMP::OID(".1.3.6.1.4.1.8072.9999.9999.1.0")) {
   $request->setValue(ASN_OCTET_STR, $string_value);
  }
  elsif ($oid == new NetSNMP::OID(".1.3.6.1.4.1.8072.9999.9999.1.1")) {
   $request->setValue(ASN_INTEGER, $integer_value);
  }
 } elsif ($request_info->getMode() == MODE_GETNEXT) {
  if ($oid == new NetSNMP::OID(".1.3.6.1.4.1.8072.9999.9999.1.0")) {
   $request->setOID(".1.3.6.1.4.1.8072.9999.9999.1.1");
   $request->setValue(ASN_INTEGER, $integer_value);
  }
  elsif ($oid < new NetSNMP::OID(".1.3.6.1.4.1.8072.9999.9999.1.0")) {
   $request->setOID(".1.3.6.1.4.1.8072.9999.9999.1.0");
   $request->setValue(ASN_OCTET_STR, $string_value);
  }
 }
}
```

当 getMode 返回 MODE_GET 时，处理程序会分析请求对象上 getOID 调用的值。如果 OID 以 ".1.0" 结束，请求的值将设为 string_value，如果 OID 以 ".1.1" 结束，则设置为 integer_value。如果 getMode 返回 MODE_GETNEXT，处理程序将决定请求的 OID 是否为 ".1.0"，然后设置 OID 和值 ".1.1"。如果树上的请求高于 ".1.0"，则设置了 ".1.0" 的 OID 和值。实际上，这将返回树中的"下一步"值，以便 snmpwalk 等程序可以遍历树，而不必事先了解结构。

变量的类型使用 NetSNMP::ASN 中的常量来设置。有关可用常数的完整列表，请参阅 NetSNMP::ASN 的 perldoc。

这个示例 Perl 插件的完整代码列表如下：

```
#!/usr/bin/perl

use NetSNMP::agent (':all');
use NetSNMP::ASN qw(ASN_OCTET_STR ASN_INTEGER);

sub hello_handler {
 my ($handler, $registration_info, $request_info, $requests) = @_;
 my $request;
 my $string_value = "hello world";
 my $integer_value = "8675309";

 for($request = $requests; $request; $request = $request->next()) {
  my $oid = $request->getOID();
  if ($request_info->getMode() == MODE_GET) {
   if ($oid == new NetSNMP::OID(".1.3.6.1.4.1.8072.9999.9999.1.0")) {
    $request->setValue(ASN_OCTET_STR, $string_value);
   }
```

```
  elsif ($oid == new NetSNMP::OID(".1.3.6.1.4.1.8072.9999.9999.1.1")) {
   $request->setValue(ASN_INTEGER, $integer_value);
  }
 } elsif ($request_info->getMode() == MODE_GETNEXT) {
  if ($oid == new NetSNMP::OID(".1.3.6.1.4.1.8072.9999.9999.1.0")) {
   $request->setOID(".1.3.6.1.4.1.8072.9999.9999.1.1");
   $request->setValue(ASN_INTEGER, $integer_value);
  }
  elsif ($oid < new NetSNMP::OID(".1.3.6.1.4.1.8072.9999.9999.1.0")) {
   $request->setOID(".1.3.6.1.4.1.8072.9999.9999.1.0");
   $request->setValue(ASN_OCTET_STR, $string_value);
  }
 }
}

my $agent = new NetSNMP::agent();
$agent->register("hello_world", ".1.3.6.1.4.1.8072.9999.9999",
        \&hello_handler);
```

要测试插件，请将上述程序复制到 /usr/share/snmp/hello_world.pl，并将以下行添加到 /etc/snmp/snmpd.conf 配置文件：

```
perl do "/usr/share/snmp/hello_world.pl"
```

需要重启 SNMP 代理守护进程来加载新的 Perl 插件。重启后，s nmpwalk 应该返回新数据：

```
~]$ snmpwalk localhost NET-SNMP-MIB::netSnmpPlaypen
NET-SNMP-MIB::netSnmpPlaypen.1.0 = STRING: "hello world"
NET-SNMP-MIB::netSnmpPlaypen.1.1 = INTEGER: 8675309
```

snmpget 也应用于操作处理程序的其他模式：

```
~]$ snmpget localhost \
  NET-SNMP-MIB::netSnmpPlaypen.1.0 \
  NET-SNMP-MIB::netSnmpPlaypen.1.1
NET-SNMP-MIB::netSnmpPlaypen.1.0 = STRING: "hello world"
NET-SNMP-MIB::netSnmpPlaypen.1.1 = INTEGER: 8675309
```

21.8. 其它资源

要了解更多有关收集系统信息的信息，请参阅以下资源。

21.8.1. 安装的文档

- *lscpu(1)- lscpu 命令的 man page。*
- *lsusb(8)- lsusb 命令的 man page。*
- *findmnt(8)- findmnt 命令的 man page。*
- *blkid(8)- blkid 命令的 man page。*
- *lsblk(8)- lsblk 命令的 man page。*
- *ps(1)- ps 命令的 man page。*
- *前(1)- 顶部 命令的 man page。*
- *免费(1)- 免费 命令的 man page。*
- *df(1)- df 命令的 man page。*
- *du(1)- du 命令的 man page。*
- *lspci(8)- lspci 命令的 man page。*
- *snmpd(8)- snmpd 服务的 man page。*
- *snmpd.conf(5)- 包含可用配置指令完整文档的 /etc/snmp/snmpd.conf 文件的 man page。*

第 22 章 OPENLMI

Open Linux 管理基础架构 通常简写为 OpenLMI，是用于管理 Linux 系统管理的通用基础架构。它建立在现有工具基础上，充当抽象层，以便向系统管理员隐藏基础系统的大部分复杂性。OpenLMI 分发有一组服务，这些服务可通过本地或远程访问并提供多种语言绑定、标准 API 和标准脚本接口，这些接口可用于管理和监控硬件、操作系统和系统服务。

22.1. 关于 OPENLMI

OpenLMI 旨在为在物理机和虚拟机上运行红帽企业 Linux 系统的生产服务器提供一个通用管理接口。它由以下三个组件组成：

1. 系统管理代理 - 这些代理安装在受管系统上，并实施提供给标准对象代理的对象模型。OpenLMI 中实施的初始代理包括存储配置和网络配置，但后续工作将解决系统管理的其他元素。系统管理代理通常称为通用信息模型提供商或 CIM 提供商。

2. 标准对象代理 - 对象代理管理系统管理代理并为其提供接口。标准对象代理也称为 CIM 对象监控器或 CIMOM。

3. 客户端应用程序和脚本 - 客户端应用程序和脚本通过标准对象代理调用系统管理代理。

OpenLMI 项目提供可由脚本或系统管理控制台使用的低级接口，补充现有的管理举措。随 OpenLMI 提供的接口包括 C、C++、Python、Java 和交互式命令行客户端，它们都提供对每个代理中实施的功能的完全访问权限。这样可确保无论您决定使用哪个编程接口，始终都能访问相同的功能。

22.1.1. 主要功能

以下是在您的系统上安装和使用 OpenLMI 的主要优点：

- OpenLMI 为本地和远程系统的配置、管理和监控提供了标准接口。
- 它允许您配置、管理和监控物理和虚拟机上运行的生产服务器。
- 它附带一系列 CIM 提供商，供您配置、管理和监控存储设备和复杂网络。

- 它允许您从 C、C++、Python 和 Java 程序调用系统管理功能，并且包含提供命令行界面的 LMIShell。

- 它是基于开放行业标准的免费软件。

22.1.2. 管理功能

OpenLMI 的主要功能包括管理存储设备、网络、系统服务、用户帐户、硬件和软件配置、电源管理和 Active Directory 的交互。有关使用 Red Hat Enterprise Linux 7 分发的 CIM 供应商的完整列表，请参考 表 22.1 “可用的 CIM 供应商”。

表 22.1. 可用的 CIM 供应商

软件包名称	描述
openlmi-account	用于管理用户帐户的 CIM 提供程序。
openlmi-logicalfile	用于读取文件和目录的 CIM 提供程序。
openlmi-networking	用于网络管理的 CIM 提供程序。
openlmi-powermanagement	电源管理的 CIM 提供程序.
openlmi-service	管理系统服务的 CIM 提供商。
openlmi-storage	存储管理的 CIM 供应商。
openlmi-fan	控制计算机粉丝的 CIM 提供商.
openlmi-hardware	用于检索硬件信息的 CIM 提供商。
openlmi-realmd	用于配置 realmd 的 CIM 提供程序。
openlmi-software[a]	用于软件管理的 CIM 提供商。

[a] 在红帽企业 Linux 7 中，OpenLMI Software 提供程序是作为技术预览包括的。这个提供程序可以完全正常工作，但存在一个已知的性能扩展问题，其中列出大量软件包可能会消耗过多内存和时间。要临时解决这个问题，请调整软件包搜索，以尽可能少地返回软件包。

22.2. INSTALLING OPENLMI

OpenLMI 作为一系列 RPM 软件包进行分发，其中包括 CIMOM、各个 CIM 提供程序和客户端应用。这可让您区分受管和客户端系统，并且只安装您需要的组件。

22.2.1. 在受管系统中安装 OpenLMI

受管系统是您想要使用 OpenLMI 客户端工具监控和管理的系统。要在受管系统中安装 OpenLMI，请完成以下步骤：

1. 以 root 用户身份在 shell 提示符后输入以下内容来安装 tog-pegasus 软件包：

```
yum install tog-pegasus
```

此命令安装 OpenPegasus CIMOM 及其系统的所有依赖项，并为 peg asus 用户创建一个用户帐户。

2. 作为 root 运行以下命令安装所需的 CIM 供应商：

```
yum install openlmi-{storage,networking,service,account,powermanagement}
```

此命令安装用于存储、网络、服务、帐户和电源管理的 CIM 提供程序。有关使用 Red Hat Enterprise Linux 7 分发的 CIM 供应商的完整列表，请参考 表 22.1 “可用的 CIM 供应商”。

3. 编辑 /etc/Pegasus/access.conf 配置文件，以自定义被允许连接到 OpenPegasus CIMOM 的用户列表。默认情况下，只有 pegasus 用户才能远程和本地访问 CIMOM。要激活此用户帐户，以 root 用户身份运行以下命令设定用户的密码：

```
passwd pegasus
```

4. 通过激活 tog-pegasus.service 单元来启动 OpenPegasus CIMOM。要在当前会话中激活 tog-pegasus.service 单元，以 root 用户身份在 shell 提示符后输入以下内容：

```
systemctl start tog-pegasus.service
```

要将 tog-pegasus.service 单元配置为在引导时自动启动，以 root 用户身份输入 ：

```
systemctl enable tog-pegasus.service
```

5. 如果您打算从远程机器与受管系统交互，请在端口 5989 (wbem-https)上启用 TCP 通信。

要在当前会话中打开此端口，以 root 用户身份运行以下命令：

```
firewall-cmd --add-port 5989/tcp
```

要永久打开端口 5989 用于 TCP 通信，以 root 用户身份输入 ：

```
firewall-cmd --permanent --add-port 5989/tcp
```

现在，您可以连接到受管系统并使用 OpenLMI 客户端工具与其交互，如 第 22.4 节 “使用 LMIShell” 所述。如果您打算直接在受管系统上执行 OpenLMI 操作，也请完成 第 22.2.2 节 “在客户端系统上安装 OpenLMI” 中描述的步骤。

22.2.2. 在客户端系统上安装 OpenLMI

客户端系统是您要与受管系统交互的系统。在典型的场景中，客户端系统和受管系统安装在两台单独的计算机上，但您也可以在托管系统上安装客户端工具并与之直接交互。

要在客户端系统中安装 OpenLMI，请完成以下步骤：

1. 以 root 用户身份在 shell 提示符后输入以下内容来安装 openlmi-tools 软件包：

```
yum install openlmi-tools
```

此命令安装 LMIShell，这是交互式客户端和解释器，用于访问 OpenPegasus 提供的 CIM 对象，以及它对系统的所有依赖项。

2. 按照 第 22.3 节 “为 OpenPegasus 配置 SSL 证书” 所述为 OpenPegasus 配置 SSL 证书。

现在您可以使用 LMIShell 客户端与受管系统交互，如 第 22.4 节 “使用 LMIShell” 所述。

22.3. 为 OPENPEGASUS 配置 SSL 证书

OpenLMI 使用基于 Web 的企业管理(WBEM)协议，该协议通过 HTTP 传输层运行。标准 HTTP 基本身份验证在此协议中执行，这意味着用户名和密码与请求一起传输。

需要将 OpenPegasus CIMOM 配置为使用 HTTPS 进行通信，以确保安全的身份验证。受管系统上需要一个安全套接字层(SSL)或传输层安全(TLS)证书来建立加密频道。

系统中可以通过两种方法管理 SSL/TLS 证书：

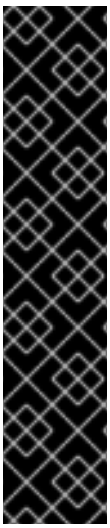
- 自签名证书使用的基础架构更少，但更难部署到客户端并难以安全管理。
- 授权签名的证书在设置后更易于向客户部署，但可能需要更大的初始投资。

在使用授权签名的证书时，需要在客户端系统上配置可信证书颁发机构。然后，该授权可用于签署所有托管系统的 CIMOM 证书。证书也可以是证书链的一部分，因此用于签署受管系统证书的证书可能由另一个更高权威机构（如 Verisign、CAcert 和 RSA 及其他）签名。

文件系统中的默认证书和信任存储位置列在 表 22.2 “证书和受信任存储位置” 中。

表 22.2. 证书和受信任存储位置

配置选项	位置	描述
sslCertificateFilePath	/etc/Pegasus/server.pem	CIMOM 的公共证书。
sslKeyFilePath	/etc/Pegasus/file.pem	仅对 CIMOM 已知的私钥.
sslTrustStore	/etc/Pegasus/client.pem	提供可信证书颁发机构列表的文件或目录。



重要

如果您修改了 表 22.2 “证书和受信任存储位置” 中提到的任何文件，请重启 tog-pegasus 服务以确保它识别新证书。要重启该服务，以 root 用户身份在 shell 提示符后输入以下内容：

```
systemctl restart tog-pegasus.service
```

有关如何在 Red Hat Enterprise Linux 7 中管理系统服务的详情请参考 第 10 章 使用 systemd 管理服务。

22.3.1. 管理自签名证书

自签名证书使用自己的私钥签署自身，并且它没有连接到任何信任链。在托管的系统上，如果在首次启动 tog-pegasus 服务之前管理员未提供证书，则将使用系统的主主机名作为证书主题自动生成一组自签名证书。

重要

自动生成的自签名证书默认为 10 年有效，但它们没有自动续订功能。对这些证书的任何修改都需要根据 OpenSSL 或 Mozilla NSS 文档中关于该主题的指南来手动创建新的证书。

要将客户端系统配置为信任自签名证书，请完成以下步骤：

1. 将 /etc/Pegasus/server.pem 证书从受管系统复制到客户端系统上的 /etc/pki/ca-trust/source/anchors/ 目录。要做到这一点，以 root 根用户身份在 shell 提示符后输入以下内容：

```
scp root@hostname:/etc/Pegasus/server.pem /etc/pki/ca-trust/source/anchors/pegasus-hostname.pem
```

使用受管系统的主机名替换 hostname。请注意，只有在 sshd 服务在受管系统上运行并且配置为允许 root 用户通过 SSH 协议登录系统时，此命令才起作用。有关如何安装和配置 sshd 服务并使用 scp 命令通过 SSH 协议传输文件的更多信息，请参阅 第 12 章 OpenSSH。

2. 通过将校验和与原始文件的检查总和进行比较，验证客户端系统上证书的完整性。要在受管系统中计算 /etc/Pegasus/server.pem 文件的检查总和，请在该系统上以 root 用户身份运行以下命令：

```
sha1sum /etc/Pegasus/server.pem
```

要在客户端系统中计算 /etc/pki/ca-trust/source/anchors/pegasus-hostname.pem 文件的检查总和，请在此系统中运行以下命令：

```
sha1sum /etc/pki/ca-trust/source/anchors/pegasus-hostname.pem
```

使用受管系统的主机名替换 hostname。

3. 以 root 用户身份运行以下命令来更新客户端系统中的信任存储：

```
update-ca-trust extract
```

22.3.2. 使用身份管理管理授权机构签名证书（推荐）

红帽企业 Linux 的身份管理功能提供了一个域控制器，简化了加入域的系统中 SSL 证书的管理。除了其他功能外，身份管理服务器提供嵌入式证书颁发机构。有关如何将客户端和受管系统加入到域中的信息，请参阅红帽企业 Linux 7 域身份、身份验证和政策指南或 FreeIPA 文档。

必须将受管系统注册到身份管理；对于客户端系统，注册是可选的。

受管系统中需要执行以下步骤：

1. 安装 ipa-client 软件包并将系统注册到身份管理，如 红帽企业 Linux 7 Linux 域身份、身份验证和策略指南 中所述。

2. 以 root 用户身份输入以下命令，将身份管理签名证书复制到可信存储中：

```
cp /etc/ipa/ca.crt /etc/pki/ca-trust/source/anchors/ipa.crt
```

3. 以 root 用户身份运行以下命令来更新信任存储：

```
update-ca-trust extract
```

4. 以特权域用户身份运行以下命令，将 Pegasus 注册到 Identity Management 域中的服务：

```
ipa service-add CIMOM/hostname
```

使用受管系统的主机名替换 hostname。

此命令可以从安装了 ipa-admintools 软件包的 Identity Management 域中的任何系统运行。它在身份管理中创建服务条目，可用于生成签名 SSL 证书。

5. 备份位于 /etc/Pegasus/ 目录中的 PEM 文件（推荐）。

6. 以 root 用户身份运行以下命令来检索签名的证书：

```
ipa-getcert request -f /etc/Pegasus/server.pem -k /etc/Pegasus/file.pem -N CN=hostname -K CIMOM/hostname
```

使用受管系统的主机名替换 hostname。

证书和密钥文件现在保存在正确的位置。ipa-client-install 脚本在受管系统中安装的 certmonger 守护进程 可确保证书保持最新，并根据需要续订。

如需更多信息，请参阅 Red Hat Enterprise Linux 7 Linux 域身份、身份验证和策略指南。

要注册客户端系统并更新信任存储，请按照以下步骤操作。

1. 安装 ipa-client 软件包并将系统注册到身份管理，如 红帽企业 Linux 7 Linux 域身份、身份验证和策略指南 中所述。

2. 以 root 用户身份输入以下命令，将身份管理签名证书复制到可信存储中：

```
cp /etc/ipa/ca.crt /etc/pki/ca-trust/source/anchors/ipa.crt
```

3. 以 root 用户身份运行以下命令来更新信任存储：

```
update-ca-trust extract
```

如果客户端系统不是在身份管理中注册，请完成以下步骤以更新信任存储。

1. 将 /etc/ipa/ca.crt 文件安全地从加入到同一身份管理域的任何其他系统复制到可信存储 /etc/pki/ca-trust/source/anchors/ 目录。

2. 以 root 用户身份运行以下命令来更新信任存储：

```
update-ca-trust extract
```

22.3.3. 手动管理授权签名证书

使用身份管理之外的其他机制管理授权签名证书需要更多手动配置。

需要确保所有客户端信任将要签署受管系统证书的颁发机构证书：

- 如果证书认证机构在默认情况下受信任，则不需要执行任何特定步骤来完成此操作。
- 如果默认情况下证书认证机构不被信任，则必须在客户端和受管系统上导入证书。

a. 以 root 用户身份输入以下命令将证书复制到可信存储中：

```
cp /path/to/ca.crt /etc/pki/ca-trust/source/anchors/ca.crt
```

b. 以 root 用户身份运行以下命令来更新信任存储：

```
update-ca-trust extract
```

在受管系统中，完成以下步骤：

1. 创建新的 SSL 配置文件 /etc/Pegasus/ssl.cnf，以存储有关证书的信息。这个文件的内容必须与以下示例类似：

```
[ req ]
distinguished_name   = req_distinguished_name
prompt         = no
[ req_distinguished_name ]
C          = US
ST           = Massachusetts
L          = Westford
O          = Fedora
OU           = Fedora OpenLMI
CN           = hostname
```

使用受管系统的完全限定域名替换 hostname。

2. 使用以下命令，以 root 用户身份在管理系统中生成私钥：

```
openssl genrsa -out /etc/Pegasus/file.pem 1024
```

3. 以 root 用户身份运行这个命令来生成证书签名请求(CSR)：

```
openssl req -config /etc/Pegasus/ssl.cnf -new -key /etc/Pegasus/file.pem -out
/etc/Pegasus/server.csr
```

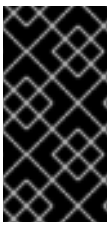
4. 将 /etc/Pegasus/server.csr 文件发送到认证机构进行签名。提交文件的详细步骤取决于特定的证书颁发机构。

5. 从证书颁发机构收到签名的证书时，将其保存为 /etc/Pegasus/server.pem。

6. 将可信颁发机构的证书复制到 Pegasus 信任存储中，以确保 Pegasus 能够通过以 root 用户身份运行来信任其自身证书：

```
cp /path/to/ca.crt /etc/Pegasus/client.pem
```

完成所有上述步骤后，信任签名授权的客户端能够与受管服务器的 CIMOM 成功通信。



重要

与 Identity Management 解决方案不同的是，如果证书过期且需要续订，则必须再次执行所有上述手动步骤。建议在证书过期前续订证书。

22.4. 使用 LMISHELL

LMIShell 是交互式客户端和非交互式解释器，可用于访问 OpenPegasus CIMOM 提供的 CIM 对象。它基于 Python 解释器，但也实施额外的功能和类来与 CIM 对象交互。

22.4.1. 启动、使用和退出 LMIShell

与 Python 解释器类似，您可以使用 LMIShell 作为交互式客户端，或者用作 LMIShell 脚本的非交互式解释器。

以交互模式启动 LMIShell

要在互动模式下启动 LMIShell 解释程序，请运行不带额外参数的 lmishell 命令：

```
lmishell
```

默认情况下，当 LMIShell 尝试建立与 CIMOM 的连接时，它将根据证书颁发机构信任存储验证服务器端证书。要禁用此验证，请使用 --noverify 或 -n 命令行选项运行 lmishell 命令：

```
lmishell --noverify
```

使用 Tab 完成

在交互模式下运行时，LMIShell 解释器允许您按 Tab 键完成基本编程结构和 CIM 对象，包括命名空间、类、方法和对象属性。

浏览历史记录

默认情况下，LMIShell 在 ~/.lmishell_history 文件的交互式提示符处存储您键入的所有命令。这样，您可以浏览命令历史记录并重复使用以交互模式重新输入的行，而无需在提示符下再次键入这些行。要在命令历史记录中向后移动，请按向上箭头键或 Ctrlp 组合键。要在命令历史记录中向前移动，请按向下箭头键或 Ctrln 组合键。

LMIShell 还支持增量反向搜索。Ctrl+r。例如：

```
> (reverse-i-search)`connect': c = connect("server.example.com", "pegasus")
```

要清除命令历史记录，请使用 clear_history（） 函数，如下所示：

```
clear_history()
```

您可以通过更改 ~/.lmishellrc 配置文件中的 history_length 选项的值来配置命令历史记录中存储的行数。另外，您可以通过更改此配置文件中 history_file 选项的值来更改历史记录文件的位置。例如，要将历史记录文件的位置设置为 ~/.lmishell_history 并配置 LMIShell 以存储其中的最大 1000 行，请将以下行添加到 ~/.lmishellrc 文件中：

```
history_file = "~/.lmishell_history"
history_length = 1000
```

处理例外

默认情况下，LMIShell 解释器处理所有异常并使用返回值。要禁用此行为来处理代码中的所有异常，请使用 use_exceptions（） 函数，如下所示：

```
use_exceptions()
```

要重新启用自动异常处理，请使用：

```
use_exception(False)
```

您可以通过将 ~/.lmishellrc 配置文件中的 use_exceptions 选项的值改为 True 来永久禁用异常处理：

```
use_exceptions = True
```

配置临时缓存

使用默认配置时，LMIShell 连接对象使用临时缓存来存储 CIM 类名称和 CIM 类，以减少网络通信。要清除这个临时缓存，请使用 clear_cache（） 方法，如下所示：

```
object_name.clear_cache()
```

使用连接对象的名称替换 object_name。

要禁用特定连接对象的临时缓存，请使用 use_cache（） 方法，如下所示：

```
object_name.use_cache(False)
```

要再次启用它，请使用：

```
object_name.use_cache(True)
```

您可以通过将 ~/.lmishellrc 配置文件中的 use_cache 选项的值更改为 False 来永久禁用连接对象的临时缓存：

```
use_cache = False
```

退出 LMIShell

要终止 LMIShell 解释器并返回到 shell 提示符，请按 Ctrl+d 组合键或发布 quit（） 功能，如下所示：

```
> quit()
~]$
```

运行 LMIShell 脚本

要运行 LMIShell 脚本，请运行 lmishell 命令，如下所示：

```
lmishell file_name
```

将 file_name 替换为脚本的名称。要在执行后检查 LMIShell 脚本，还要指定 --interact 或 -i 命令行选项：

```
lmishell --interact file_name
```

LMIShell 脚本的首选文件扩展名为 .lmi。

22.4.2. 连接到 CIMOM

LMIShell 允许您连接到在同一系统上本地运行的 CIMOM，或者连接到可通过网络访问的远程计算机上。

连接到远程 CIMOM

要访问远程 CIMOM 提供的 CIM 对象，请使用 connect（） 功能创建一个连接对象，如下所示：

```
connect(host_name, user_name, password)
```

使用受管系统的主机名替换 host_name，user_name 替换为被允许连接到该系统上运行的 OpenPegasus CIMOM 的用户名称，密码为 用户的密码。如果省略密码，LMIShell 会提示用户输入密码。函数返回 LMIConnection 对象。

例 22.1. 连接到远程 CIMOM

以用户 peg asus 的身份连接到在 server.example.com 上运行的 OpenPegasus CIMOM，在交互式提示下输入以下内容：

```
> c = connect("server.example.com", "pegasus")
password:
>
```

连接到本地 CIMOM

LMIShell 允许您使用 Unix 套接字连接到本地 CIMOM。对于这种连接，您必须以 root 用户身份运行 LMIShell 解释器，并且 /var/run/tog-peg-pegasus/cimxml.socket 套接字必须存在。

要访问本地 CIMOM 提供的 CIM 对象，请使用 connect（） 功能创建一个连接对象，如下所示：

```
connect(host_name)
```

将 host_name 替换为 localhost、127. 0.0.1 或 ::1。函数返回 LMIConnection 对象或 None。

例 22.2. 连接到本地 CIMOM

以 root 用户身份 连接到 localhost 上运行的 OpenPegasus CIMOM，在互动提示下输入以下内容：

```
> c = connect("localhost")
>
```

验证到 CIMOM 的连接

connect（） 函数返回 LMIConnection 对象，如果无法建立连接，则返回 None。另外，当 connect（） 函数无法建立连接时，它会向标准错误输出输出错误消息。

要验证与 CIMOM 的连接是否已成功建立，请使用 isinstance（） 功能，如下所示：

```
isinstance(object_name, LMIConnection)
```

使用连接对象的名称替换 object_name。如果 object_name 是 LMIConnection 对象，则此函数返回 True，否则返回 False。

例 22.3. 验证到 CIMOM 的连接

要验证 例 22.1 “连接到远程 CIMOM” 中创建的 c 变量是否包含 LMIConnection 对象，在互动提示符后输入以下内容：

```
> isinstance(c, LMIConnection)
True
>
```

另外，您可以验证 c 是否不是 None ：

```
> c is None
False
>
```

22.4.3. 使用命名空间

LMIShell 命名空间提供组织可用类的自然方法，并充当到其他命名空间和类的层次结构访问点。root 命名空间是连接对象的第一个入口点。

列出可用的命名空间

要列出所有可用的命名空间，请使用 print_namespaces（） 方法，如下所示：

```
object_name.print_namespaces()
```

将 object_name 替换为要检查的对象的名称。此方法将可用命名空间打印到标准输出。

要获取可用命名空间的列表，请访问对象属性 命名空间 ：

```
object_name.namespaces
```

这将返回字符串列表。

例 22.4. 列出可用的命名空间

要检查 例 22.1 “连接到远程 CIMOM” 中创建的 c 连接对象的 根 命名空间对象并列出所有可用的命名空间，在互动提示符下输入以下内容：

```
> c.root.print_namespaces()
cimv2
interop
```

```
PG_InterOp
PG_Internal
>
```

要将这些命名空间的列表分配给名为 root_namespaces 的变量，请输入：

```
> root_namespaces = c.root.namespaces
>
```

访问命名空间对象

要访问特定的命名空间对象，请使用以下语法：

```
object_name.namespace_name
```

将 object_name 替换为要检查的对象的名称，并将 namespace_name 替换为要访问的命名空间的名称。这会返回 LMINamespace 对象。

例 22.5. 访问命名空间对象

要访问在 例 22.1 “连接到远程 CIMOM” 中创建的 c connection 对象的 cimv2 命名空间并将其分配给变量 named ns，在互动提示中输入以下内容：

```
> ns = c.root.cimv2
>
```

22.4.4. 使用类

LMIShell 类表示 CIMOM 提供的类。您可以访问和列出其属性、方法、实例、实例名称和 ValueMap 属性，打印它们的文档字符串，然后创建新实例和实例名称。

列出可用的类

要列出特定命名空间中的所有可用类，请使用 print_classes（） 方法，如下所示：

```
namespace_object.print_classes()
```

将 namespace_object 替换为要检查的命名空间对象。此方法将可用的类打印到标准输出。

要获取可用类的列表，请使用 class（） 方法：

```
namespace_object.classes()
```

此方法返回字符串列表。

例 22.6. 列出可用的类

要检查 例 22.5 “访问命名空间对象” 中创建 的命名空间 对象并列出所有可用的类，在互动提示下输入以下内容：

```
> ns.print_classes()
CIM_CollectionInSystem
CIM_ConcreteIdentity
CIM_ControlledBy
CIM_DeviceSAPImplementation
CIM_MemberOfStatusCollection
...
>
```

要将这些类的列表分配给名为 cimv2_classes 的变量，请输入：

```
> cimv2_classes = ns.classes()
>
```

访问类对象

要访问 CIMOM 提供的特定类对象，请使用以下语法：

```
namespace_object.class_name
```

将 namespace_object 替换为要检查的命名空间对象的名称，并将 class_name 替换为要访问的类的名称。

例 22.7. 访问类对象

要访问 例 22.5 “访问命名空间对象” 中创建 的命名空间 对象 LMI_IPNetworkConnection 类并将其分配给名为 cls 的变量，请在交互式提示中输入以下内容：

```
> cls = ns.LMI_IPNetworkConnection
>
```

检查类对象

所有类对象存储其名称和所属命名空间的信息，以及详细的类文档。要获取特定类对象的名称，请使用以下语法：

```
class_object.classname
```

将 class_object 替换为要检查的类对象的名称。这会返回对象名称的字符串表示。

要获取有关类对象所属命名空间的信息，请使用：

```
class_object.namespace
```

这会返回命名空间的字符串表示。

要显示详细的类文档，请使用 doc（） 方法，如下所示：

```
class_object.doc()
```

例 22.8. 检查类对象

要检查 **例 22.7 “访问类对象” 中创建 的 cls** 类对象并显示其名称和对应命名空间，请在交互式提示符后输入以下内容：

```
> cls.classname
'LMI_IPNetworkConnection'
> cls.namespace
'root/cimv2'
>
```

要访问类文档，请输入：

```
> cls.doc()
Class: LMI_IPNetworkConnection
  SuperClass: CIM_IPNetworkConnection
  [qualifier] string UMLPackagePath: 'CIM::Network::IP'
```

```
    [qualifier] string Version: '0.1.0'
...
```

列出可用的方法

要列出特定类对象的所有可用方法，请使用 print_methods（） 方法，如下所示：

```
class_object.print_methods()
```

将 class_object 替换为要检查的类对象的名称。此方法将可用方法打印到标准输出。

要获取可用方法的列表，请使用 method （） 方法：

```
class_object.methods()
```

此方法返回字符串列表。

例 22.9. 列出可用的方法

要检查 **例 22.7 “访问类对象”** 中创建 的 cls 类对象并列出所有可用的方法，请在交互式提示符处输入以下内容：

```
> cls.print_methods()
RequestStateChange
>
```

要将这些方法的列表分配给名为 service_methods 的变量，请输入：

```
> service_methods = cls.methods()
>
```

列出可用属性

要列出特定类对象的所有可用属性，请使用 print_properties（） 方法，如下所示：

```
class_object.print_properties()
```

将 class_object 替换为要检查的类对象的名称。此方法打印标准输出的可用属性。

要获取可用属性的列表，请使用 properties（） 方法：

```
class_object.properties()
```

此方法返回字符串列表。

例 22.10. 列出可用属性

要检查 例 22.7 “访问类对象” 中创建 的 cls 类对象并列出所有可用属性，请在交互式提示符处输入以下内容：

```
> cls.print_properties()
RequestedState
HealthState
StatusDescriptions
TransitioningToState
Generation
...
>
```

要将这些类的列表分配给名为 service_properties 的变量，请输入：

```
> service_properties = cls.properties()
>
```

列出和查看值映射属性

CIM 类可以在其受管对象格式(MOF)定义中包含 ValueMap 属性。ValueMap 属性包含恒定值，在调用方法或检查返回的值时很有用。

要列出特定类对象的所有可用 ValueMap 属性，请使用 print_valuemap_properties（） 方法，如下所示：

```
class_object.print_valuemap_properties()
```

将 class_object 替换为要检查的类对象的名称。此方法将可用的 ValueMap 属性输出到标准输出：

要获取可用 ValueMap 属性列表，请使用 valuemap_properties（） 方法：

```
class_object.valuemap_properties()
```

此方法返回字符串列表。

例 22.11. 列出 ValueMap 属性

要检查 **例 22.7 “访问类对象”** 中创建 的 cls 类对象并列出所有可用的 ValueMap 属性，请在交互式提示符后输入以下内容：

```
> cls.print_valuemap_properties()
RequestedState
HealthState
TransitioningToState
DetailedStatus
OperationalStatus
...
>
```

要将这些 ValueMap 属性的列表分配给名为 service_valuemap_properties 的变量，请输入：

```
> service_valuemap_properties = cls.valuemap_properties()
>
```

要访问特定的 ValueMap 属性，请使用以下语法：

```
class_object.valuemap_propertyValues
```

将 valuemap_property 替换为要访问的 ValueMap 属性的名称。

要列出所有可用的常量值，请使用 print_values（） 方法，如下所示：

```
class_object.valuemap_propertyValues.print_values()
```

此方法将打印标准输出的命名常数值。您还可以使用 values( )方法获取可用常数值 的列表：

```
class_object.valuemap_propertyValues.values()
```

此方法返回字符串列表。

例 22.12. 访问 ValueMap 属性

例 22.11 “列出 ValueMap 属性” 提到名为 Requested State 的 ValueMap 属性。要检查此属性并列出可用的常数值，请在互动提示中输入以下内容：

```
> cls.RequestedStateValues.print_values()
Reset
NoChange
NotApplicable
Quiesce
Unknown
...
>
```

要将这些常量值的列表分配给名为 requests _state_values 的变量，请输入：

```
> requested_state_values = cls.RequestedStateValues.values()
>
```

要访问特定的常量值，请使用以下语法：

```
class_object.valuemap_propertyValues.constant_value_name
```

将 constant_value_name 替换为常量值的名称。或者，您可以使用 value（） 方法，如下所示：

```
class_object.valuemap_propertyValues.value("constant_value_name")
```

要确定特定常量值的名称，请使用 value_name（） 方法：

```
class_object.valuemap_propertyValues.value_name("constant_value")
```

此方法返回字符串。

例 22.13. 访问 Constant 值

例 22.12 “访问 ValueMap 属性” 显示 Requested State 属性提供了名为 Reset 的常量值。要访问名为常量值的这个值，在互动提示中输入以下内容：

```
> cls.RequestedStateValues.Reset
11
> cls.RequestedStateValues.value("Reset")
11
>
```

要确定这个恒定值的名称，请输入：

```
> cls.RequestedStateValues.value_name(11)
u'Reset'
>
```

获取 CIMClass 对象

许多类方法不需要访问 CIMClass 对象，这就是为什么 LMIShell 仅当调用方法真正需要时从 CIMOM 获取此对象。要手动获取 CIMClass 对象，请使用 fetch（） 方法，如下所示：

```
class_object.fetch()
```

将 class_object 替换为类对象的名称。请注意，需要访问 CIMClass 对象的方法会自动获取它。

22.4.5. 操作实例

LMIShell 实例表示 CIMOM 提供的实例。您可以获取和设置其属性、列表和调用方法，打印其文档字符串，获取关联或关联对象的列表，将修改的对象推送到 CIMOM，以及从 CIMOM 删除个别实例。

访问实例

要获取特定类对象的所有可用实例列表，请使用 instance （） 方法，如下所示：

```
class_object.instances()
```

将 class_object 替换为要检查的类对象的名称。此方法返回 LMIInstance 对象列表。

要访问类对象的第一个实例，请使用 first_instance（） 方法：

```
class_object.first_instance()
```

此方法返回 LMIInstance 对象。

除了列出所有实例或返回第一个实例外，instances （） 和 first_instance （） 都支持可选参数来允许您过滤结果：

```
class_object.instances(criteria)
```

```
class_object.first_instance(criteria)
```

使用由键值对组成的字典替换条件，其中键表示实例属性和值表示这些属性的必要值。

例 22.14. 访问实例

要查找在 **例 22.7 “访问类对象”** 中创建的 cls 类对象的第一个实例，其 ElementName 属性等于 eth0，并将它分配给名为 device 的变量，在交互式提示下输入以下内容：

```
> device = cls.first_instance({"ElementName": "eth0"})
>
```

检查实例

所有实例对象存储其类名称和所属命名空间的信息，以及其属性和值的详细文档。此外，实例对象允许您检索唯一身份识别对象。

要获取特定实例对象的类名称，请使用以下语法：

```
instance_object.classname
```

将 instance_object 替换为要检查的实例对象的名称。这将返回类名称的字符串表示。

要获取有关实例对象所属命名空间的信息，请使用：

```
instance_object.namespace
```

这会返回命名空间的字符串表示。

要检索实例对象的唯一标识对象，请使用：

```
instance_object.path
```

这会返回 LMIInstanceName 对象。

最后，要显示详细的文档，请使用 doc（） 方法，如下所示：

```
instance_object.doc()
```

例 22.15. 检查实例

要检查 **例 22.14 “访问实例” 中**创建 的设备 实例对象并显示其类名称和对应命名空间，请在互动提示符后输入以下内容：

```
> device.classname
u'LMI_IPNetworkConnection'
> device.namespace
'root/cimv2'
>
```

要访问实例对象文档，请输入：

```
> device.doc()
Instance of LMI_IPNetworkConnection
  [property] uint16 RequestedState = '12'

  [property] uint16 HealthState

  [property array] string [] StatusDescriptions
...
```

创建新实例

某些 CIM 提供程序允许您创建特定类对象的新实例。要创建类对象的新实例，请使用 create_instance（） 方法，如下所示：

```
class_object.create_instance(properties)
```

将 class_object 替换为类对象的名称和属性，该字典由键值对组成，其中键表示实例属性和值代表属性值。此方法返回 LMIInstance 对象。

例 22.16. 创建新实例

LMI_Group 类代表系统组，LMI_Account 类代表受管系统上的用户帐户。要使用 例 22.5 “访问命名空间对象” 中创建 的命名空间 对象，为名为 peg asus 的系统组和名为 lmishell-user 的用户创建这两个类的实例，并将其分配给名为 group 和 user 的变量，在交互式提示中输入以下内容：

```
> group = ns.LMI_Group.first_instance({"Name" : "pegasus"})
> user = ns.LMI_Account.first_instance({"Name" : "lmishell-user"})
>
```

要为 lmishell-user 用户获取 LMI_Identity 类的实例，请输入：

```
> identity = user.first_associator(ResultClass="LMI_Identity")
>
```

LMI_MemberOfGroup 类代表系统组成员资格。要使用 LMI_MemberOfGroup 类将 lmishell-user 添加到 pegasus 组，请按照如下所示创建新实例：

```
> ns.LMI_MemberOfGroup.create_instance({
...   "Member" : identity.path,
...   "Collection" : group.path})
LMIInstance(classname="LMI_MemberOfGroup", ...)
>
```

删除单个实例

要从 CIMOM 删除特定实例，请使用 delete（） 方法，如下所示：

```
instance_object.delete()
```

将 instance_object 替换为要删除的实例对象的名称。此方法返回布尔值：请注意，删除实例后，其属性和方法将无法访问。

例 22.17. 删除单个实例

LMI_Account 类代表受管系统上的用户帐户。要使用 例 22.5 “访问命名空间对象” 中创建 的命名空间 对象，请为名为 lmishell-user 的用户创建一个 LMI_Account 类实例，并将其分配给名为 user

的变量，在交互式提示下键入以下内容：

```
> user = ns.LMI_Account.first_instance({"Name" : "lmishell-user"})
>
```

要删除此实例并从系统中删除 lmishell-user，请输入：

```
> user.delete()
True
>
```

列出和访问可用属性

要列出特定实例对象的所有可用属性，请使用 print_properties（） 方法，如下所示：

```
instance_object.print_properties()
```

将 instance_object 替换为要检查的实例对象的名称。此方法打印标准输出的可用属性。

要获取可用属性的列表，请使用 properties（） 方法：

```
instance_object.properties()
```

此方法返回字符串列表。

例 22.18. 列出可用属性

要检查 **例 22.14 “访问实例”** 中创建 的设备 实例对象并列出所有可用的属性，请在互动提示下输入以下内容：

```
> device.print_properties()
RequestedState
HealthState
StatusDescriptions
TransitioningToState
Generation
...
>
```

要将这些属性的列表分配给名为 device_properties 的变量，请输入：

```
> device_properties = device.properties()
>
```

要获取特定属性的当前值，请使用以下语法：

```
instance_object.property_name
```

将 property_name 替换为要访问的属性的名称。

要修改特定属性的值，请按如下所示为其分配一个值：

```
instance_object.property_name = value
```

使用 属性的新值替换 value。请注意，要将更改传播到 CIMOM，还必须执行 push（） 方法：

```
instance_object.push()
```

此方法返回一个包含返回值、返回值参数和错误字符串的三项元组。

例 22.19. 访问单个属性

要检查 **例 22.14 “访问实例”** 中创建 的设备 实例对象并显示名为 SystemName 的属性值，请在交互式提示符后输入以下内容：

```
> device.SystemName
u'server.example.com'
>
```

列出和使用可用的方法

要列出特定实例对象的所有可用方法，请使用 print_methods（） 方法，如下所示：

```
instance_object.print_methods()
```

将 instance_object 替换为要检查的实例对象的名称。此方法将可用方法打印到标准输出。

要获取可用方法的列表，请使用 method（）方法 ：

```
instance_object.methods()
```

此方法返回字符串列表。

例 22.20. 列出可用的方法

要检查 **例 22.14 “访问实例”** 中创建 的设备 实例对象并列出所有可用的方法，请在交互式提示下输入以下内容：

```
> device.print_methods()
RequestStateChange
>
```

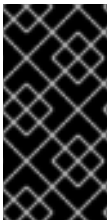
要将这些方法的列表分配给名为 network_device_methods 的变量，请输入：

```
> network_device_methods = device.methods()
>
```

要调用特定方法，请使用以下语法：

```
instance_object.method_name(
 parameter=value,
 ...)
```

将 instance_object 替换为要使用的实例对象名称，metric_name 替换为要调用的方法的名称，parameter 替换为要设置的参数的名称，value 替换为此参数的值。方法返回由返回值、返回值参数和错误字符串组成的三项元组。



重要

LMIInstance 对象 不会自动 刷新其内容（属性、方法、限定符等）。要做到这一点，请使用如下所示的 refresh（） 方法。

例 22.21. 使用方法

PG_ComputerSystem 类代表系统。要使用在 例 22.5 “访问命名空间对象” 中创建了 then s namespace 对象并把它分配给名为 sys 的变量来创建此类实例，请在互动提示中输入以下内容：

```
> sys = ns.PG_ComputerSystem.first_instance()
>
```

LMI_AccountManagementService 类实施的方法允许您管理系统中的用户和组。要创建此类实例并将其分配给名为 acc 的变量，请输入：

```
> acc = ns.LMI_AccountManagementService.first_instance()
>
```

要在系统中创建名为 lmishell-user 的新用户，请使用 CreateAccount（） 方法，如下所示：

```
> acc.CreateAccount(Name="lmishell-user", System=sys)
LMIReturnValue(rval=0, rparams=NocaseDict({u'Account':
LMIInstanceName(classname="LMI_Account"...), u'Identities':
[LMIInstanceName(classname="LMI_Identity"...),
LMIInstanceName(classname="LMI_Identity"...)]}), errorstr='')
```

LMIShell 支持同步方法调用：使用同步方法时，LMIShell 会等待对应的作业对象将其状态更改为"finished"，然后返回此作业的返回参数。如果给定方法返回以下类之一的对象，LMIShell 能够执行同步方法调用：

- LMI_StorageJob
- LMI_SoftwareInstallationJob
- LMI_NetworkJob

LMIShell 首先尝试将信号用作等待方法。如果失败，则改为使用轮询方法。

要执行同步方法调用，请使用以下语法：

```
instance_object.Syncmethod_name(
 parameter=value,
 ...)
```

将 instance_object 替换为要使用的实例对象名称，metric_name 替换为要调用的方法的名称，parameter 替换为要设置的参数的名称，value 替换为此参数的值。所有同步方法的名称中包含 Sync 前缀，并返回由作业返回值、作业返回值参数和作业的错误字符串组成的三项元组。

您还可以强制 LMIShell 仅使用轮询方法。要做到这一点，请指定 PreferPolling 参数，如下所示：

```
instance_object.Syncmethod_name(
 PreferPolling=True
 parameter=value,
 ...)
```

列出和查看 ValueMap 参数

CIM 方法可以在其受管对象格式(MOF)定义中包含 ValueMap 参数。ValueMap 参数包含恒定值。

要列出特定方法的所有可用 ValueMap 参数，请使用 print_valuemap_parameters（） 方法，如下所示：

```
instance_object.method_name.print_valuemap_parameters()
```

将 instance_object 替换为实例对象的名称，将 method_name 替换为要检查的方法的名称。此方法将可用的 ValueMap 参数打印到标准输出。

要获取可用 ValueMap 参数列表，请使用 valuemap_parameters（） 方法：

```
instance_object.method_name.valuemap_parameters()
```

此方法返回字符串列表。

例 22.22. 列出 ValueMap 参数

要检查 例 22.21 “使用方法” 中创建 的 acc 实例对象并列出 CreateAccount（） 方法的所有可用 ValueMap 参数，在交互式提示符处输入以下内容：

```
> acc.CreateAccount.print_valuemap_parameters()
CreateAccount
>
```

要将这些 ValueMap 参数的列表分配给名为 create_account_parameters 的变量，请输入：

```
> create_account_parameters = acc.CreateAccount.valuemap_parameters()
>
```

要访问特定的 ValueMap 参数，请使用以下语法：

```
instance_object.method_name.valuemap_parameterValues
```

将 valuemap_parameter 替换为要访问的 ValueMap 参数的名称。

要列出所有可用的常量值，请使用 print_values（） 方法，如下所示：

```
instance_object.method_name.valuemap_parameterValues.print_values()
```

此方法将打印标准输出的命名常数值。您还可以使用 values( )方法获取可用常数值 的列表：

```
instance_object.method_name.valuemap_parameterValues.values()
```

此方法返回字符串列表。

例 22.23. 访问 ValueMap 参数

例 22.22 “列出 ValueMap 参数” 提到名为 CreateAccount 的 ValueMap 参数。要检查这个参数并列出可用的常数值，在互动提示中输入以下内容：

```
> acc.CreateAccount.CreateAccountValues.print_values()
Operationunsupported
Failed
Unabletosetpasswordusercreated
Unabletocreatehomedirectoryusercreatedandpasswordset
Operationcompletedsuccessfully
>
```

要将这些常量值的列表分配给名为 create_account_values 的变量，请输入：

```
> create_account_values = acc.CreateAccount.CreateAccountValues.values()
>
```

要访问特定的常量值，请使用以下语法：

```
instance_object.method_name.valuemap_parameterValues.constant_value_name
```

将 constant_value_name 替换为常量值的名称。或者，您可以使用 value（） 方法，如下所示：

```
instance_object.method_name.valuemap_parameterValues.value("constant_value_name")
```

要确定特定常量值的名称，请使用 value_name（） 方法：

```
instance_object.method_name.valuemap_parameterValues.value_name("constant_value")
```

此方法返回字符串。

例 22.24. 访问 **Constant** 值

例 22.23 “访问 ValueMap 参数” 显示 CreateAccount ValueMap 参数提供了名为 Failed 的常量值。要访问名为常量值的这个值，在互动提示中输入以下内容：

```
> acc.CreateAccount.CreateAccountValues.Failed
2
> acc.CreateAccount.CreateAccountValues.value("Failed")
2
>
```

要确定这个恒定值的名称，请输入：

```
> acc.CreateAccount.CreateAccountValues.value_name(2)
u'Failed'
>
```

刷新实例对象

如果这样的对象在使用 LMIShell 时发生改变，则代表 CIMOM 端 CIM 对象的 LMIShell 使用的本地对象可能会过时。要更新特定实例对象的属性和方法，请使用 refresh（） 方法，如下所示：

```
instance_object.refresh()
```

将 instance_object 替换为要刷新的对象的名称。此方法返回一个包含返回值、返回值参数和错误字符串的三项元组。

例 22.25. 刷新实例对象

要更新在 例 22.14 “访问实例” 中创建 的设备 实例对象的属性和方法，在互动提示中输入以下内容：

```
> device.refresh()
LMIReturnValue(rval=True, rparams=NocaseDict({}), errorstr='')
>
```

显示 MOF 表述

要显示实例对象的受管对象格式(MOF)，请使用 tomof（） 方法，如下所示：

```
instance_object.tomof()
```

将 instance_object 替换为要检查的实例对象的名称。此方法将对象的 MOF 表示形式打印到标准输出。

例 22.26. 显示 MOF 表述

要显示在 例 22.14 “访问实例” 中创建 的设备 实例对象的 MOF 表述，在互动提示中输入以下内容：

```
> device.tomof()
instance of LMI_IPNetworkConnection {
    RequestedState = 12;
    HealthState = NULL;
    StatusDescriptions = NULL;
    TransitioningToState = 12;
...
```

22.4.6. 使用实例名称

LMIShell 实例名称是存放一组主键及其值的对象。这种类型的对象完全识别实例。

访问实例名称

CIMInstance 对象由 CIMInstanceName 对象标识。要获取所有可用实例名称对象的列表，请使用 instance_names（） 方法，如下所示：

```
class_object.instance_names()
```

将 class_object 替换为要检查的类对象的名称。此方法返回 LMIInstanceName 对象列表。

要访问类对象的第一个实例名称对象，请使用 first_instance_name（） 方法：

```
class_object.first_instance_name()
```

此方法返回 LMIInstanceName 对象。

除了列出所有实例名称对象或返回第一个对象外，instance _names（） 和 first_instance_name（） 支持可选参数以允许您过滤结果：

```
class_object.instance_names(criteria)
```

```
class_object.first_instance_name(criteria)
```

使用由键值对组成的字典替换条件，其中键表示键属性和值表示这些键属性的必要值。

例 22.27. 访问实例名称

要查找在 例 22.7 “访问类对象” 中创建的 cls 类对象的第一个实例名称，其 Name 键属性等于 eth0，并将它分配给名为 device_name 的变量，在交互式提示下输入以下内容：

```
> device_name = cls.first_instance_name({"Name": "eth0"})
>
```

检查实例名称

所有实例名称对象存储其类名称和所属命名空间的信息。

要获取特定实例名称对象的类名称，请使用以下语法：

```
instance_name_object.classname
```

将 instance_name_object 替换为要检查的实例名称对象的名称。这将返回类名称的字符串表示。

要获取有关实例名称对象所属命名空间的信息，请使用：

```
instance_name_object.namespace
```

这会返回命名空间的字符串表示。

例 22.28. 检查实例名称

要检查 **例 22.27 “访问实例名称”** 中创建的 device_name 实例名称对象并显示其类名称和对应命名空间，请在交互式提示符后输入以下内容：

```
> device_name.classname
u'LMI_IPNetworkConnection'
> device_name.namespace
'root/cimv2'
>
```

创建新实例名称

如果您知道远程对象的所有主键，则 LMIShell 允许您创建新的 wrapped CIMInstanceName 对象。然后，此实例名称对象可用于检索整个实例对象。

要为类对象创建新实例名称，请使用 new_instance_name（） 方法，如下所示：

```
class_object.new_instance_name(key_properties)
```

将 class_object 替换为类对象的名称，而 key_properties 替换为由键值对组成的字典，其中键代表键属性和值代表键属性值。此方法返回 LMIInstanceName 对象。

例 22.29. 创建新实例名称

LMI_Account 类代表受管系统上的用户帐户。要使用 例 22.5 “访问命名空间对象” 中创建 的命名空间 对象，并创建代表受管系统中 lmishell-user 用户的 LMI_Account 类的新实例名称，请在交互式提示符后输入以下内容：

```
> instance_name = ns.LMI_Account.new_instance_name({
...   "CreationClassName" : "LMI_Account",
...   "Name" : "lmishell-user",
...   "SystemCreationClassName" : "PG_ComputerSystem",
...   "SystemName" : "server"})
>
```

列出和访问密钥属性

要列出特定实例名称对象的所有可用键属性，请使用 print_key_properties（） 方法，如下所示：

```
instance_name_object.print_key_properties()
```

将 instance_name_object 替换为要检查的实例名称对象的名称。此方法打印标准输出的可用密钥属性。

要获取可用密钥属性列表，请使用 key_properties（） 方法：

```
instance_name_object.key_properties()
```

此方法返回字符串列表。

例 22.30. 列出可用密钥属性

要检查 例 22.27 “访问实例名称” 中创建的 device_name 实例名称对象并列出所有可用的密钥属性，请在互动提示符后输入以下内容：

```
> device_name.print_key_properties()
CreationClassName
SystemName
Name
SystemCreationClassName
>
```

要将这些密钥属性的列表分配给名为 device_name_properties 的变量，请输入：

```
> device_name_properties = device_name.key_properties()
>
```

要获取特定键属性的当前值，请使用以下语法：

```
instance_name_object.key_property_name
```

将 key_property_name 替换为要访问的键属性的名称。

例 22.31. 访问单个密钥属性

要检查 例 22.27 “访问实例名称” 中创建的 device_name 实例名称对象并显示名为 SystemName 的键属性值，在交互式提示符处输入以下内容：

```
> device_name.SystemName
u'server.example.com'
>
```

将实例名称转换为实例

每个实例名称可以转换为实例。要做到这一点，请使用 to_instance（） 方法，如下所示：

```
instance_name_object.to_instance()
```

将 instance_name_object 替换为要转换的实例名称对象的名称。此方法返回 LMIInstance 对象。

例 22.32. 将实例名称转换为实例

要将 例 22.27 “访问实例名称” 中创建的 device_name 实例名称对象转换为实例对象，并将其分配给名为 device 的变量，请在互动提示中输入以下内容：

```
> device = device_name.to_instance()
>
```

22.4.7. 使用关联的对象

通用信息模型定义受管对象之间的关联关系。

访问关联的实例

要获取与特定实例对象关联的所有对象的列表，请使用关联器（） 方法，如下所示：

```
instance_object.associators(
 AssocClass=class_name,
 ResultClass=class_name,
 ResultRole=role,
 IncludeQualifiers=include_qualifiers,
 IncludeClassOrigin=include_class_origin,
 PropertyList=property_list)
```

要访问与特定实例对象关联的第一个对象，请使用 first_associator（） 方法：

```
instance_object.first_associator(
 AssocClass=class_name,
 ResultClass=class_name,
 ResultRole=role,
 IncludeQualifiers=include_qualifiers,
 IncludeClassOrigin=include_class_origin,
 PropertyList=property_list)
```

将 instance_object 替换为要检查的实例对象的名称。您可以通过指定以下参数来过滤结果：

- AssocClass - 每个返回的对象必须通过本课程的实例或其子类与源对象关联。默认值为 None。
- ResultClass - 返回的每个对象必须是此类的实例或其子类，或者必须是此类或其子类之一。默认值为 None。
- 角色 - 每个返回的对象必须通过关联（源对象在其中充当指定角色）与源对象关联。引用源对象的关联类中的属性名称必须与此参数的值匹配。默认值为 None。
- ResultRole - 每个返回的对象必须通过关联与源对象关联，在其中返回的对象充当指定角色。引用返回对象的关联类中的属性名称必须与此参数的值匹配。默认值为 None。

剩余的参数指的是：

- IncludeQualifiers - 表示每个对象的所有限定符（包括对象上和任何返回的属性上的限定符）是否都应包含在响应中，作为 QUALIFIER 元素包含在响应中。默认值为 False。

- IncludeClassOrigin - 指明是否应将 CLASSORIGIN 属性显示在每个返回的对象中的所有相关元素的布尔值。默认值为 False。

- AtityList - 此列表的成员定义了一个或多个属性名称。返回的对象不包括在此列表中缺失的任何属性的元素。如果 properties List 是空列表，则返回的对象中不包含任何属性。如果是 None，则不定义额外的过滤。默认值为 None。

例 22.33. 访问关联的实例

LMI_StorageExtent 类代表系统中可用的块设备。要使用 **例 22.5 “访问命名空间对象”** 中创建 的命名空间 对象，请为名为 /dev/vda 的块设备创建一个 LMI_StorageExtent 类实例，并将其分配给名为 vda 的变量，在交互式提示下键入以下内容：

```
> vda = ns.LMI_StorageExtent.first_instance({
...   "DeviceID" : "/dev/vda"})
>
```

要获取此块设备上所有磁盘分区的列表并将其分配给名为 vda_partitions 的变量，请使用关联 器（） 方法，如下所示：

```
> vda_partitions = vda.associators(ResultClass="LMI_DiskPartition")
>
```

访问关联的实例名称

要获取特定实例对象的所有关联的实例名称列表，请使用 associator_names（） 方法，如下所示：

```
instance_object.associator_names(
  AssocClass=class_name,
  ResultClass=class_name,
  Role=role,
  ResultRole=role)
```

要访问特定实例对象的第一个关联的实例名称，请使用 first_associator_name（） 方法：

```
instance_object.first_associator_name(
  AssocClass=class_object,
```

```
ResultClass=class_object,
Role=role,
ResultRole=role)
```

将 instance_object 替换为要检查的实例对象的名称。您可以通过指定以下参数来过滤结果：

- AssocClass - 每个返回的名称都标识必须通过此类实例或其子类与源对象关联的对象。默认值为 None。
- ResultClass - 每个返回的名称标识一个对象，该对象必须是本课程的实例或其子类，或者它必须是此类或其子类之一。默认值为 None。
- 角色 - 每个返回的名称都标识必须通过源对象扮演指定角色的关联与源对象关联的对象。引用源对象的关联类中的属性名称必须与此参数的值匹配。默认值为 None。
- ResultRole - 每个返回的名称标识必须通过关联的关联与源对象关联的对象，在其中返回的指定对象 play 指定指定的角色。引用返回对象的关联类中的属性名称必须与此参数的值匹配。默认值为 None。

例 22.34. 访问关联的实例名称

要使用 **例 22.33 “访问关联的实例”** 中创建的 vda 实例对象，请获取其关联的实例名称列表并将其分配给名为 vda_partitions 的变量，请输入：

```
> vda_partitions = vda.associator_names(ResultClass="LMI_DiskPartition")
>
```

22.4.8. 使用关联对象

通用信息模型定义受管对象之间的关联关系。关联对象定义另外两个对象之间的关系。

访问关联实例

要获得引用特定目标对象的关联对象列表，请使用 reference （） 方法，如下所示：

```
instance_object.references(
 ResultClass=class_name,
 Role=role,
```

```
IncludeQualifiers=include_qualifiers,
IncludeClassOrigin=include_class_origin,
PropertyList=property_list)
```

要访问引用特定目标对象的第一个关联对象，请使用 first_reference（） 方法：

```
instance_object.first_reference(
... ResultClass=class_name,
... Role=role,
... IncludeQualifiers=include_qualifiers,
... IncludeClassOrigin=include_class_origin,
... PropertyList=property_list)
>
```

将 instance_object 替换为要检查的实例对象的名称。您可以通过指定以下参数来过滤结果：

- ResultClass - 返回的每个对象必须是此类的实例或其子类，或者必须是此类或其子类之一。默认值为 None。
- 角色 - 每个返回的对象必须通过具有与此参数值匹配的名称的属性引用目标对象。默认值为 None。

剩余的参数指的是：

- IncludeQualifiers - 指明每个对象（包括对象上的限定符以及任何返回的属性上的限定符）的布尔值应包含在响应中，作为 QUALIFIER 元素包含在响应中。默认值为 False。
- IncludeClassOrigin - 指明是否应将 CLASSORIGIN 属性显示在每个返回的对象中的所有相关元素的布尔值。默认值为 False。
- AtityList - 此列表的成员定义了一个或多个属性名称。返回的对象不包括在此列表中缺失的任何属性的元素。如果 properties List 是空列表，则返回的对象中不包含任何属性。如果是 None，则不定义额外的过滤。默认值为 None。

例 22.35. 访问关联实例

LMI_LANEndpoint 类代表与特定网络接口设备关联的通信端点。要使用 例 22.5 “访问命名空间对象” 中创建 的命名空间 对象，请为名为 eth0 的网络接口设备创建一个 LMI_LANEndpoint 类实

例，并将其分配给名为 lan_endpoint 的变量，在交互式提示符后输入以下内容：

```
> lan_endpoint = ns.LMI_LANEndpoint.first_instance({
...   "Name" : "eth0"})
>
```

要访问引用 LMI_BindsToLANEndpoint 对象的第一个关联对象，并将其分配给名为 bind 的变量，请输入：

```
> bind = lan_endpoint.first_reference(
...   ResultClass="LMI_BindsToLANEndpoint")
>
```

现在，您可以使用 Dependent 属性访问代表对应网络接口设备的 IP 地址的依赖 LMI_IPProtocolEndpoint 类：

```
> ip = bind.Dependent.to_instance()
> print ip.IPv4Address
192.168.122.1
>
```

访问关联实例名称

要获取特定实例对象的关联实例名称列表，请使用 reference_names（） 方法，如下所示：

```
instance_object.reference_names(
  ResultClass=class_name,
  Role=role)
```

要访问特定实例对象的第一个关联实例名称，请使用 first_reference_name（） 方法：

```
instance_object.first_reference_name(
  ResultClass=class_name,
  Role=role)
```

将 instance_object 替换为要检查的实例对象的名称。您可以通过指定以下参数来过滤结果：

- ResultClass - 返回的每个对象名称标识此类的实例或其子类，或标识此类或其子类之一。默认值为 None。

角色 - 每个返回的对象通过属性引用目标实例，该属性与此参数的值匹配。默认值为 None。

例 22.36. 访问关联实例名称

要使用 例 22.35 “访问关联实例” 中创建的 lan_endpoint 实例对象，访问引用 LMI_BindsToLANEndpoint 对象的第一个关联实例名称，并将其分配给名为 bind 的变量，类型：

```
> bind = lan_endpoint.first_reference_name(
...   ResultClass="LMI_BindsToLANEndpoint")
```

现在，您可以使用 Dependent 属性访问代表对应网络接口设备的 IP 地址的依赖 LMI_IPProtocolEndpoint 类：

```
> ip = bind.Dependent.to_instance()
> print ip.IPv4Address
192.168.122.1
>
```

22.4.9. 使用 Indications

表示是对响应特定数据变化的特定事件作出响应。LMIShell 可以订阅一个信号来接收此类事件响应。

订阅印度语

要订阅指示信息，请使用 subscribe_indication（） 方法，如下所示：

```
connection_object.subscribe_indication(
  QueryLanguage="WQL",
  Query='SELECT * FROM CIM_InstModification',
  Name="cpu",
  CreationNamespace="root/interop",
  SubscriptionCreationClassName="CIM_IndicationSubscription",
  FilterCreationClassName="CIM_IndicationFilter",
  FilterSystemCreationClassName="CIM_ComputerSystem",
  FilterSourceNamespace="root/cimv2",
  HandlerCreationClassName="CIM_IndicationHandlerCIMXML",
  HandlerSystemCreationClassName="CIM_ComputerSystem",
  Destination="http://host_name:5988")
```

或者，您可以使用方法调用的较短版本，如下所示：

```
connection_object.subscribe_indication(
```

```
Query='SELECT * FROM CIM_InstModification',
Name="cpu",
Destination="http://host_name:5988")
```

使用连接对象替换 connection_object，将 host_name 替换为您要向其发送声明的系统的主机名。

默认情况下，当解释器终止时，LMIShell 解释器创建的所有订阅都会被自动删除。要更改此行为，请将 persistent=True 关键字参数传递到 subscribe_indication（） 方法调用。这将阻止 LMIShell 删除订阅。

例 22.37. 订阅印度语

要使用在 例 22.1 “连接到远程 CIMOM” 中创建的 c 连接对象并订阅名为 cpu 的指示，在互动提示中输入以下内容：

```
> c.subscribe_indication(
...   QueryLanguage="WQL",
...   Query='SELECT * FROM CIM_InstModification',
...   Name="cpu",
...   CreationNamespace="root/interop",
...   SubscriptionCreationClassName="CIM_IndicationSubscription",
...   FilterCreationClassName="CIM_IndicationFilter",
...   FilterSystemCreationClassName="CIM_ComputerSystem",
...   FilterSourceNamespace="root/cimv2",
...   HandlerCreationClassName="CIM_IndicationHandlerCIMXML",
...   HandlerSystemCreationClassName="CIM_ComputerSystem",
...   Destination="http://server.example.com:5988")
LMIReturnValue(rval=True, rparams=NocaseDict({}), errorstr='')
>
```

列出订阅的引用

要列出所有订阅的指示，请使用 print_subscribed_indications（） 方法，如下所示：

```
connection_object.print_subscribed_indications()
```

将 connection_object 替换为要检查的连接对象的名称。此方法打印订阅了标准输出的指示。

要获得订阅的注解列表，请使用 subscription _indications（） 方法：

```
connection_object.subscribed_indications()
```

此方法返回字符串列表。

例 22.38. 列出订阅的引用

要检查 例 22.1 “连接到远程 CIMOM” 中创建的 c 连接对象并列出所有订阅的声明，请在互动提示下输入以下内容：

```
> c.print_subscribed_indications()
>
```

要将这些暗示的列表分配给名为投注的变量，请输入：

```
> indications = c.subscribed_indications()
>
```

取消订阅 Indications

默认情况下，当解释器终止时，LMIShell 解释器创建的所有订阅都会被自动删除。要更快地删除单个订阅，请使用 unsubscribe_indication（） 方法，如下所示：

```
connection_object.unsubscribe_indication(indication_name)
```

使用连接对象的名称替换 connection_object，将 indicating_name 替换为要删除的指示符的名称。

要删除所有订阅，请使用 unsubscribe_all_indications（） 方法：

```
connection_object.unsubscribe_all_indications()
```

例 22.39. 取消订阅 Indications

要使用在 例 22.1 “连接到远程 CIMOM” 中创建的 c 连接对象并取消订阅在 例 22.37 “订阅印度语” 中创建的指示，在互动提示符下输入以下内容：

```
> c.unsubscribe_indication('cpu')
LMIReturnValue(rval=True, rparams=NocaseDict({}), errorstr='')
>
```

实施 Indication Handler

subscribe_indication（） 方法允许您指定您要向其发送声明的系统的主机名。以下示例演示了如何实施指示处理器：

```
> def handler(ind, arg1, arg2, kwargs): ... exported_objects = ind.exported_objects() ...
do_something_with(exported_objects) > listener = LmiIndicationListener("0.0.0.0",
listening_port) > listener.add_handler("indication-name-XXXXXXXX", handler, arg1, arg2,
kwargs)
> listener.start()
>
```

处理程序的第一个参数是 LmiIndication 对象，其中包含由指示器导出的方法和对象的列表。其他参数特定于用户：向侦听器添加处理程序时需要指定这些参数。

在上例中，add _handler（） 方法调用使用具有八个"X"字符的特殊字符串。这些字符替换为由侦听器生成的随机字符串，以避免可能的处理程序名称冲突。要使用随机字符串，请首先启动指示监听器，然后订阅指示，使得处理程序对象的 Destination 属性包含以下值：
schema://host_name/random_string。

例 22.40. 实施 Indication Handler

以下脚本演示了如何编写监控位于 192.168.122.1 的受管系统的处理程序，并在创建新用户帐户时调用 indicates _callback（） 功能：

```
#!/usr/bin/lmishell

import sys
from time import sleep
from lmi.shell.LMIUtil import LMIPassByRef
from lmi.shell.LMIIndicationListener import LMIIndicationListener

# These are passed by reference to indication_callback
var1 = LMIPassByRef("some_value")
var2 = LMIPassByRef("some_other_value")

def indication_callback(ind, var1, var2):
  # Do something with ind, var1 and var2
  print ind.exported_objects()
  print var1.value
  print var2.value

c = connect("hostname", "username", "password")

listener = LMIIndicationListener("0.0.0.0", 65500)
unique_name = listener.add_handler(
  "demo-XXXXXXXX",  # Creates a unique name for me
  indication_callback, # Callback to be called
```

```
  var1,      # Variable passed by ref
  var2       # Variable passed by ref
)

listener.start()

print c.subscribe_indication(
  Name=unique_name,
  Query="SELECT * FROM LMI_AccountInstanceCreationIndication WHERE
SOURCEINSTANCE ISA LMI_Account",
  Destination="192.168.122.1:65500"
)

try:
  while True:
    sleep(60)
except KeyboardInterrupt:
  sys.exit(0)
```

22.4.10. 用法示例

本节为随 OpenLMI 软件包分发的各种 CIM 提供程序提供了多个示例。本节中的所有示例都使用以下两个变量定义：

```
c = connect("host_name", "user_name", "password")
ns = c.root.cimv2
```

使用受管系统的主机名替换 host _name，user_name 替换为被允许连接到该系统上运行的 OpenPegasus CIMOM 的用户名称，密码为用户的密码。

使用 OpenLMI Service Provider

openlmi-service 软件包安装 CIM 提供程序以管理系统服务。以下示例演示了如何使用 CIM 提供程序列出可用的系统服务，以及如何启动、停止、启用和禁用这些服务。

例 22.41. 列出可用的服务

要列出受管机器上的所有可用服务以及有关服务是否已启动(TRUE)还是停止(FALSE)和状态字符串的信息，请使用以下代码片段：

```
for service in ns.LMI_Service.instances():
  print "%s:\t%s" % (service.Name, service.Status)
```

要只列出默认启用的服务，请使用这个代码片段：

```
cls = ns.LMI_Service
for service in cls.instances():
  if service.EnabledDefault == cls.EnabledDefaultValues.Enabled:
    print service.Name
```

请注意，已启用的服务 EnabledDefault 属性的值等于 2，对于禁用的服务则等于 3。

要显示 cups 服务的信息，请使用：

```
cups = ns.LMI_Service.first_instance({"Name": "cups.service"})
cups.doc()
```

例 22.42. 启动和停止服务

要启动和停止 cups 服务并查看其当前状态，请使用以下代码片段：

```
cups = ns.LMI_Service.first_instance({"Name": "cups.service"})
cups.StartService()
print cups.Status
cups.StopService()
print cups.Status
```

例 22.43. 启用和禁用服务

要启用或禁用 cups 服务并显示其 EnabledDefault 属性，请使用以下代码片段：

```
cups = ns.LMI_Service.first_instance({"Name": "cups.service"})
cups.TurnServiceOff()
print cups.EnabledDefault
cups.TurnServiceOn()
print cups.EnabledDefault
```

使用 OpenLMI 网络提供程序

openlmi-networking 软件包会安装用于网络的 CIM 提供程序。以下示例演示了如何使用此 CIM 提供程序列出与特定端口号关联的 IP 地址、创建新连接、配置静态 IP 地址以及激活连接。

例 22.44. 列出与给定端口号关联的 IP 地址

要列出与 eth0 网络接口关联的所有 IP 地址，请使用以下代码片段：

```
device = ns.LMI_IPNetworkConnection.first_instance({'ElementName': 'eth0'})
for endpoint in device.associators(AssocClass="LMI_NetworkSAPSAPDependency",
ResultClass="LMI_IPProtocolEndpoint"):
  if endpoint.ProtocolIFType == ns.LMI_IPProtocolEndpoint.ProtocolIFTypeValues.IPv4:
    print "IPv4: %s/%s" % (endpoint.IPv4Address, endpoint.SubnetMask)
  elif endpoint.ProtocolIFType == ns.LMI_IPProtocolEndpoint.ProtocolIFTypeValues.IPv6:
    print "IPv6: %s/%d" % (endpoint.IPv6Address, endpoint.IPv6SubnetPrefixLength)
```

此代码片段使用与给定的 LMI_IPProtocolEndction 类关联的 LMI_IPProtocolEndpoint 类。

要显示默认网关，请使用这个代码片段：

```
for rsap in
device.associators(AssocClass="LMI_NetworkRemoteAccessAvailableToElement",
ResultClass="LMI_NetworkRemoteServiceAccessPoint"):
  if rsap.AccessContext ==
ns.LMI_NetworkRemoteServiceAccessPoint.AccessContextValues.DefaultGateway:
    print "Default Gateway: %s" % rsap.AccessInfo
```

默认网关由 LMI_NetworkRemoteServiceAccessPoint 实例表示，AccessContext 属性等于 DefaultGateway。

要获取 DNS 服务器列表，需要按照如下所示遍历对象模型：

1. 使用 LMI_Network SAPSAPDependency 获取与给定 LMI_IPNetworkConnection 关联的 LMI_ IPProtocolEndpoint 实例。

2. 对 LMI_DNSProtocolEndpoint 实例使用相同的关联。

具有 AccessContext 属性的 LMI_NetworkRemoteServiceAccessPoint 实例等于通过 LMI_NetworkRemoteAccessAvailableToElement 关联的 DNS 服务器地址，在 AccessInfo 属性中具有 DNS 服务器地址。

可以通过更多路径进入 RemoteServiceAccessPath，并且可以重复条目。以下代码片段使用 set（） 功能从 DNS 服务器列表中删除重复条目：

```
dnsservers = set()
for ipendpoint in device.associators(AssocClass="LMI_NetworkSAPSAPDependency",
ResultClass="LMI_IPProtocolEndpoint"):
  for dnsedpoint in
ipendpoint.associators(AssocClass="LMI_NetworkSAPSAPDependency",
ResultClass="LMI_DNSProtocolEndpoint"):
    for rsap in
dnsedpoint.associators(AssocClass="LMI_NetworkRemoteAccessAvailableToElement",
ResultClass="LMI_NetworkRemoteServiceAccessPoint"):
      if rsap.AccessContext ==
ns.LMI_NetworkRemoteServiceAccessPoint.AccessContextValues.DNSServer:
        dnsservers.add(rsap.AccessInfo)
print "DNS:", ", ".join(dnsservers)
```

例 22.45. 创建新连接和配置静态 IP 地址

要为网络接口 eth0 创建带有静态 IPv4 和无状态 IPv6 配置的新设置，请使用以下代码片段：

```
capability = ns.LMI_IPNetworkConnectionCapabilities.first_instance({ 'ElementName':
'eth0' })
result = capability.LMI_CreateIPSetting(Caption='eth0 Static',
    IPv4Type=capability.LMI_CreateIPSetting.IPv4TypeValues.Static,
    IPv6Type=capability.LMI_CreateIPSetting.IPv6TypeValues.Stateless)
setting = result.rparams["SettingData"].to_instance()
for settingData in
setting.associators(AssocClass="LMI_OrderedIPAssignmentComponent"):
  if setting.ProtocolIFType == ns.LMI_IPAssignmentSettingData.ProtocolIFTypeValues.IPv4:
    # Set static IPv4 address
    settingData.IPAddresses = ["192.168.1.100"]
    settingData.SubnetMasks = ["255.255.0.0"]
    settingData.GatewayAddresses = ["192.168.1.1"]
    settingData.push()
```

此代码片段通过在 LMI_ IPNetworkConnectionCapabilities 实例上调用 LMI_CreateIPSetting（） 方法来创建新设置，它通过 LMI_IPNetworkConnectionEnementCapabilities 与 LMI_IPNetwork ConnectionElementCapabilities 关联。它还使用 push（） 方法修改 设置。

例 22.46. 激活连接

要将设置应用到网络接口，请调用 LMI_ IPConfigurationService 类的 ApplySettingToIP NetworkConnection（） 方法。这个方法是异步的，并返回一个作业。以下代码片段演示了如何同步调用此方法：

```
setting = ns.LMI_IPAssignmentSettingData.first_instance({ "Caption": "eth0 Static" })
port = ns.LMI_IPNetworkConnection.first_instance({ 'ElementName': 'ens8' })
```

```
service = ns.LMI_IPConfigurationService.first_instance()
service.SyncApplySettingToIPNetworkConnection(SettingData=setting,
IPNetworkConnection=port, Mode=32768)
```

Mode 参数会影响设置的应用方式。此参数最常用的值如下：

- 1 - 立即应用 设置，并将其自动激活。
- 2 - 使设置自动激活且现在不应用。
- 4 - 断开和禁用自动激活.
- 5 - 不更改设置状态，仅禁用自动激活。
- 32768 - 应用 设置。
- 32769 - 断开连接.

使用 OpenLMI 存储提供程序

openlmi-storage 软件包安装 CIM 提供程序以进行存储管理。以下示例演示了如何使用此 CIM 提供程序创建卷组，创建逻辑卷，构建文件系统，挂载文件系统，以及列出系统已知的块设备。

除了 c 和 ns 变量外，这些示例还使用以下变量定义：

```
MEGABYTE = 1024*1024
storage_service = ns.LMI_StorageConfigurationService.first_instance()
filesystem_service = ns.LMI_FileSystemConfigurationService.first_instance()
```

例 22.47. 创建卷组

要创建位于 /dev/myGroup/ 中，且默认扩展大小为 4 MB 的新卷组，请使用以下代码片断：

```
# Find the devices to add to the volume group
# (filtering the CIM_StorageExtent.instances()
```

```
# call would be faster, but this is easier to read):
sda1 = ns.CIM_StorageExtent.first_instance({"Name": "/dev/sda1"})
sdb1 = ns.CIM_StorageExtent.first_instance({"Name": "/dev/sdb1"})
sdc1 = ns.CIM_StorageExtent.first_instance({"Name": "/dev/sdc1"})

# Create a new volume group:
(ret, outparams, err) = storage_service.SyncCreateOrModifyVG(
    ElementName="myGroup",
    InExtents=[sda1, sdb1, sdc1])
vg = outparams['Pool'].to_instance()
print "VG", vg.PoolID, \
    "with extent size", vg.ExtentSize, \
    "and", vg.RemainingExtents, "free extents created."
```

例 22.48. 创建逻辑卷

要创建两个大小为 100 MB 的逻辑卷，请使用这个代码片段：

```
# Find the volume group:
vg = ns.LMI_VGStoragePool.first_instance({"Name": "/dev/mapper/myGroup"})

# Create the first logical volume:
(ret, outparams, err) = storage_service.SyncCreateOrModifyLV(
    ElementName="Vol1",
    InPool=vg,
    Size=100 * MEGABYTE)
lv = outparams['TheElement'].to_instance()
print "LV", lv.DeviceID, \
    "with", lv.BlockSize * lv.NumberOfBlocks,\
    "bytes created."

# Create the second logical volume:
(ret, outparams, err) = storage_service.SyncCreateOrModifyLV(
    ElementName="Vol2",
    InPool=vg,
    Size=100 * MEGABYTE)
lv = outparams['TheElement'].to_instance()
print "LV", lv.DeviceID, \
    "with", lv.BlockSize * lv.NumberOfBlocks, \
    "bytes created."
```

例 22.49. 创建文件系统

要从 **例 22.48 “创建逻辑卷”** 在逻辑卷 lv 中创建 ext3 文件系统，请使用以下代码片段：

```
(ret, outparams, err) = filesystem_service.SyncLMI_CreateFileSystem(
    FileSystemType=filesystem_service.LMI_CreateFileSystem.FileSystemTypeValues.EXT3,
    InExtents=[lv])
```

例 22.50. 挂载文件系统

要挂载在 **例 22.49 “创建文件系统”** 中创建的文件系统，请使用以下代码片段：

```
# Find the file system on the logical volume:
fs = lv.first_associator(ResultClass="LMI_LocalFileSystem")

mount_service = ns.LMI_MountConfigurationService.first_instance()
(rc, out, err) = mount_service.SyncCreateMount(
  FileSystemType='ext3',
  Mode=32768, # just mount
  FileSystem=fs,
  MountPoint='/mnt/test',
  FileSystemSpec=lv.Name)
```

例 22.51. 列出块设备

要列出所有系统已知的块设备，请使用以下代码片段：

```
devices = ns.CIM_StorageExtent.instances()
for device in devices:
 if lmi_isinstance(device, ns.CIM_Memory):
  # Memory and CPU caches are StorageExtents too, do not print them
  continue
 print device.classname,
 print device.DeviceID,
 print device.Name,
 print device.BlockSize*device.NumberOfBlocks
```

使用 OpenLMI 硬件提供程序

openlmi-hardware 软件包安装用于监控硬件的 CIM 提供程序。以下示例演示了如何使用 CIM 提供程序检索有关机器的 CPU、内存模块、PCI 设备、制造商和型号的信息。

例 22.52. 查看 CPU 信息

要显示基本 CPU 信息，如 CPU 名称、处理器内核数和硬件线程数量，请使用以下代码片断：

```
cpu = ns.LMI_Processor.first_instance()
cpu_cap = cpu.associators(ResultClass="LMI_ProcessorCapabilities")[0]
print cpu.Name
print cpu_cap.NumberOfProcessorCores
print cpu_cap.NumberOfHardwareThreads
```

例 22.53. 查看内存信息

要显示有关内存模块的基本信息，比如它们的各个大小，请使用以下代码片段：

```
mem = ns.LMI_Memory.first_instance()
for i in mem.associators(ResultClass="LMI_PhysicalMemory"):
  print i.Name
```

例 22.54. 查看机箱信息

要显示有关机器的基本信息，如制造商或其型号，请使用以下代码片段：

```
chassis = ns.LMI_Chassis.first_instance()
print chassis.Manufacturer
print chassis.Model
```

例 22.55. 列出 PCI 设备

要列出所有系统已知的 PCI 设备，请使用以下代码片段：

```
for pci in ns.LMI_PCIDevice.instances():
  print pci.Name
```

22.5. 使用 OPENLMI 脚本

LMIShell 解释器基于 Python 模块构建，可用于开发自定义管理工具。OpenLMI 脚本项目提供了多个 Python 库，以便与 OpenLMI 提供程序交互。此外，它通过 lmi 分发，是一种可扩展的实用程序，可用于从命令行与这些库交互。

要在您的系统中安装 OpenLMI 脚本，在 shell 提示符后输入以下内容：

```
easy_install --user openlmi-scripts
```

此命令会在 ~/.local/ 目录中安装 Python 模块和 lmi 实用程序。要扩展 lmi 工具的功能，请使用以下命令安装额外的 OpenLMI 模块：

```
easy_install --user package_name
```

有关可用模块的完整列表，请查看 Python 网站。有关 OpenLMI 脚本的更多信息，请参阅官方 OpenLMI 脚本文档。

22.6. 其它资源

有关 OpenLMI 和系统管理的更多信息，请参阅以下列出的资源。

安装的文档

- lmishell(1)- lmishell 客户端和解释器的 man page 提供了有关其执行和使用的详细信息。

在线文档

- 红帽企业 Linux 7 联网指南 - 红帽企业 Linux 7 的网络指南记录了有关系统上网络接口和网络服务的配置和管理的相关信息。
- 红帽企业 Linux 7 存储管理指南 - 红帽企业 Linux 7 的存储管理指南 提供了有关如何管理系统上存储设备和文件系统的说明。
- 红帽企业 Linux 7 电源管理指南 - 红帽企业 Linux 7 的电源管理指南解释了如何有效管理系统的功耗。它讨论了降低服务器和笔记本电脑功耗的不同技术，并解释了每种技术如何影响系统整体性能。
- 红帽企业 Linux 7 域身份、身份验证和政策指南 - 红帽企业 Linux 7 的 Linux 域身份、身份验证和策略指南 涵盖了安装、配置和管理 IPA 域（包括服务器和客户端）的所有方面。本指南适用于 IT 和系统管理员。
- FreeIPA 文档 - FreeIPA 文档 充当使用 FreeIPA 身份管理项目的主要用户文档。
- OpenSSL 主页 - OpenSSL 主页提供 OpenSSL 项目概述。
- Mozilla NSS 文档 - Mozilla NSS 文档 充当使用 Mozilla NSS 项目的主要用户文档。

另请参阅

- 第 4 章 管理用户和组 记录了如何在图形用户界面和命令行中管理系统用户和组。

- 第 9 章 yum 描述如何使用 Yum 软件包管理器在命令行中搜索、安装、更新和卸载软件包。

- 第 10 章 使用 systemd 管理服务 介绍 systemd 和文档，以及如何使用 systemctl 命令来管理系统服务、配置 systemd 目标和执行电源管理命令。

- 第 12 章 OpenSSH 描述如何配置 SSH 服务器以及如何使用 ssh、scp 和 sftp 客户端实用程序进行访问。

第 23 章 查看和管理日志文件

日志文件是包含有关系统的消息的文件，包括内核、服务及其上运行的应用。不同信息有不同的日志文件。例如，有默认的系统日志文件、仅用于安全消息的日志文件，以及用于 cron 任务的日志文件。

当尝试对系统问题进行故障排除（如尝试加载内核驱动程序或查找未授权登录尝试系统时）时，日志文件非常有用。本章讨论查找日志文件的位置、如何查看日志文件以及在日志文件中查找什么内容。

有些日志文件由名为 rsyslogd 的守护进程控制。rsyslogd 守护进程是 sysklogd 的增强替代品，提供扩展过滤、加密保护消息转发、各种配置选项、输入和输出模块，支持通过 TCP 或 UDP 协议传输。请注意，r syslogd 与 sysklogd 兼容。

日志文件也可以由 journald 守护进程（ systemd 的一个组件）进行管理。journald 守护进程捕获 Syslog 消息、内核日志消息、初始 RAM 磁盘和早期启动消息，以及写入到所有服务的标准输出和标准错误输出的消息，对其进行索引，并使其可供用户使用。原生日志文件格式是结构化、索引化的二进制文件，改进了搜索速度和速度，还存储了时间戳或用户 ID 等元数据信息。默认情况下，journald 生成的日志文件不是永久的，日志文件仅存储在内存中或 /run/log/journal/ 目录中的小型 ring-buffer 中。记录的数据量取决于可用内存，当您达到容量限制时，会删除最旧的条目。但是，这个设置可以被更改 - 请参阅 第 23.10.5 节 “启用持久性存储”。有关日志的详情请参考 第 23.10 节 “使用日志”。

默认情况下，这两个日志记录工具在您的系统上共存。journald 守护进程是故障排除的主要工具。它还提供了创建结构化日志消息所需的其他数据。journald 获取的数据将转发到 /run/systemd/journal/syslog 套接字，供 rsyslogd 用于进一步处理数据。但是，r syslog 默认通过 imjournal 输入模块进行实际集成，从而避免上述套接字。您还可以使用 a omjournal 模块，以相反的方向将数据从 rsyslogd 传输到 journald。如需更多信息，请参阅 第 23.7 节 “Rsyslog 和日志的交互”。该集成支持以一致的格式维护基于文本的日志，以确保与依赖于 rsyslogd 的潜在应用程序或配置兼容。另外，您还可以以结构化格式维护 rsyslog 信息（请参阅 第 23.8 节 “使用 Rsyslog 的结构化日志记录”）。

23.1. 查找日志文件

由 rsyslogd 维护的日志文件列表可在 /etc/rsyslog.conf 配置文件中找到。大多数日志文件都位于 /var/log/ 目录中。httpd 和 samba 等一些应用在其日志文件中有一个 /var/log/ 中的目录。

您可以在 /var/log/ 目录中看到多个文件，其编号位于 /var/log/ 目录中（例如： cron-20100906）。这些数据表示已添加到轮转的日志文件中的时间戳。日志文件会被轮转，使其文件大小不会变得太大。logrotate 软件包包含一个 cron 任务，它根据 /etc/logrotate.conf 配置文件和 /etc/logrotate.d/ 目录中的配置文件自动轮转日志文件。

23.2. RSYSLOG 的基本配置

rsyslog 的主要配置文件是 /etc/rsyslog.conf。您可以在此处指定由 过滤 和操作部分组成的 全局指令、模块和 规则 。另外，您可以在哈希符号(#)后面以文本形式添加注释。

23.2.1. 过滤器

规则由过滤器部分指定，该过滤器选择 syslog 消息的子集和操作部分，后者指定如何处理选定的消息。要在 /etc/rsyslog.conf 配置文件中定义规则，请在一行上同时定义过滤器和操作，并使用一个或多个空格或标签页将它们分隔。

rsyslog 提供了根据所选属性过滤 syslog 消息的各种方式。可用的过滤方法可以划分为 基于 facility/Priority、基于 属性 和 表达式的过滤器。

基于工具/基于优先级的过滤器

过滤 syslog 消息的最常用方式是使用基于设备/优先级的过滤器，该过滤器根据两个条件过滤 syslog 消息：由点隔开的设备和 优先级。要创建选择器，请使用以下语法：

```
FACILITY.PRIORITY
```

其中：

- FACILITY 指定生成特定 syslog 消息的子系统。例如，mail 子系统 处理与邮件相关的所有 syslog 消息。FACILITY 可以通过以下关键字之一（或数字代码）表示： Kern(0)、用户 (1)、邮件 (2)、后台程序 (3)、auth (4)、syslog (5)、lpr (6)、新闻 (7)、cron (8)、authpriv (9)、ftp(10) 和 local0 通过 local7 (16 - 23)。

- PRIORITY 指定 syslog 消息的优先级。PRIORITY 可通过以下关键字之一表示（或按编号表示）： 调试 (7)、info(6)、 notice (5)、 warning （4)、err (3)、crit (2)、警报 (1)和 emerg(0)。

  上述语法选择优先级为已定义或更高的 syslog 消息。通过前面的任何优先级关键字及等号(=)，您仅会选择具有指定优先级的 syslog 消息。所有其他优先级将被忽略。相反，在优先级关键字前面带有感叹号(!)可选择除定义优先级的消息之外的所有 syslog 消息。

  除了上面指定的关键字外，您还可以使用星号(*)定义所有设施或优先级（具体取决于您将星号放置在逗号之前或之后）。指定 priority 关键字 none 适用于没有指定优先级的设备。设备和优先级条件都区分大小写。

要定义多个设备和优先级，请使用逗号分隔(,)。要在一行中定义多个选择器，请使用分号(;)分隔它们。请注意，选择器字段中的每个选择器都可以覆盖前面的选择器，这可从模式中排除一些优先级。

> **例 23.1. 工具/基于优先级的过滤器**
>
> 以下是可在 /etc/rsyslog.conf 中指定的简单工具/ 基于优先级的过滤器的几个示例。要选择所有具有任何优先级的内核 syslog 信息，请在配置文件中添加以下文本：
>
> ```
> kern.*
> ```
>
> 要选择优先级为 crit 或更高版本的所有邮件 syslog 信息，请使用以下格式：
>
> ```
> mail.crit
> ```
>
> 要选择除 info 或 debug 优先级以外的所有 cron syslog 信息，请使用以下格式设置配置：
>
> ```
> cron.!info,!debug
> ```

基于属性的过滤器

通过基于属性的过滤器，您可以按任何属性（如 时间生成 或 syslogtag）过滤 syslog 消息。有关属性的详情请参考 “属性”一节。您可以使用 表 23.1 “基于属性的 compare-operations” 中列出的 compare-operations 之一将每个指定属性与特定值进行比较。属性名称和比较操作都区分大小写。

基于属性的过滤器必须以冒号(:)开头。要定义过滤器，请使用以下语法：

```
:PROPERTY, [!]COMPARE_OPERATION, "STRING"
```

其中：

- PROPERTY 属性指定所需的属性。
- 可选感叹号(!)否定 compare-operation 的输出。基于属性的过滤器目前不支持其他布尔值操作器。

- COMPARE_OPERATION 属性指定 表 23.1 “基于属性的 compare-operations” 中列出的其中一个 compare-operations。

- STRING 属性指定属性提供的文本与以下值相比的值：这个值必须用引号括起。要转义字符串内的特定字符（如引号(")），请使用反斜杠字符(\)。

表 23.1. 基于属性的 compare-operations

compare-operation	描述
包含	检查提供的字符串是否与 属性提供的文本的任何部分匹配。
isequal	将提供的字符串与 属性提供的所有文本进行比较。这两个值必须完全等于匹配。
开头	检查提供的字符串是否完全在 属性提供的文本开头找到。
regex	将提供的 POSIX BRE（基本正则表达式）与 属性提供的文本进行比较。
ereregex	将提供的 POSIX ERE（扩展正则表达式）正则表达式与 属性提供的文本进行比较。
isempty	检查 属性是否为空。该值将被丢弃。这在使用规范化数据时尤其有用，因为这些数据可能会根据规范化结果进行填充。

例 23.2. 基于属性的过滤器

以下是可以在 /etc/rsyslog.conf 中指定的基于属性的过滤器的几个示例。要选择在消息文本中包含字符串 error 的 syslog 信息，请使用：

```
:msg, contains, "error"
```

以下过滤器选择从主机名 host1 接收的 syslog 信息：

```
:hostname, isequal, "host1"
```

要选择不包含任何字词严重和错误信息 （ 例如，致命 lib 错误），请输入：

```
:msg, !regex, "fatal .* error"
```

基于表达式的过滤器

基于表达式的过滤器根据定义的算术、布尔值或字符串操作选择 syslog 消息。基于表达式的过滤器使用 rsyslog自身的脚本语言（称为 RainerScript ）来构建复杂的过滤器。

基于表达式的过滤器的基本语法如下：

```
if EXPRESSION then ACTION else ACTION
```

其中：

- EXPRESSION 属性表示要评估的表达式，例如： $msg start with 'DEVNAME' 或 $syslogfacility-text == 'local0'。您可以使用 和 和 或 运算符 在单个过滤器中指定多个表达式。

  请注意， r syslog 支持在基于表达式的过滤器中进行不区分大小写的比较。您可以在 EXPRESSION 属性中使用 include_i 或 startwith_i compare-operations，例如：

  如果 $hostname start with_i "<HOST_NAME>"，则 ACTION.

- ACTION 属性表示在表达式返回值 true 时要执行的操作。这可以是单个操作，也可以是用大括号括起的任意复杂脚本。

- 如果是在新行开头，则关键字表示基于表达式的过滤器。then 关键字将 EXPRESSION 与 ACTION 分隔开。（可选）您可以使用 else 关键字指定在未满足条件时要执行的操作。

使用基于表达式的过滤器，您可以使用大括号中括起的脚本来嵌套条件，如 例 23.3 “基于表达式的过滤器” 中所示。脚本允许您在表达式内使用基于 facility/priority 的过滤器。另一方面，此处不建议基于属性的过滤器。RainerScript 支持带有特殊函数 re_match（） 和 re_extract（）的 正则表达式。

例 23.3. 基于表达式的过滤器

以下表达式包含两个嵌套条件：由名为prog1 的程序创建的日志文件根据消息中是否存在"test"字符串分成两个文件。

```
if $programname == 'prog1' then {
  action(type="omfile" file="/var/log/prog1.log")
  if $msg contains 'test' then
   action(type="omfile" file="/var/log/prog1test.log")
  else
   action(type="omfile" file="/var/log/prog1notest.log")
}
```

有关各种基于表达式的过滤器的更多示例，请参阅 “在线文档”一节。RainerScript 是 rsyslog新配置格式的基础，请参考 第 23.3 节 “使用新配置格式”

23.2.2. 操作

操作指定将使用由已定义的选择器过滤的消息执行的操作。以下是您可以在规则中定义的一些操作：

将 syslog 消息保存到日志文件中

大多数操作都指定将 syslog 消息保存到的日志文件中。这可以通过在您已定义的选择器后指定文件路径来实现：

```
FILTER PATH
```

其中 FILTER 代表用户指定的选择器，PATH 是目标文件的路径。

例如，以下规则由一个选择器组成，该选择器选择所有 cron syslog 信息，以及将其保存到 /var/log/cron.log 日志文件中的操作：

```
cron.* /var/log/cron.log
```

默认情况下，每次生成 syslog 消息时都会同步日志文件。使用短划线标记(-)作为您指定的文件路径前缀，以忽略同步：

```
FILTER -PATH
```

请注意，如果在写入尝试后系统终止了正确的信息，则可能会丢失信息。但是，这个设置可以提高性能，特别是您运行生成非常详细的日志消息的程序。

您指定的文件路径可以是静态的，也可以是动态的。静态文件由固定文件路径表示，如上例中所

示。动态文件路径可能会根据收到的消息而有所不同。动态文件路径由模板和一个问号(?)前缀表示：

```
FILTER ?DynamicFile
```

其中 DynamicFile 是预定义模板的名称，用于修改输出路径。您可以使用短划线前缀(-)禁用同步，也可以使用用冒号(:)分隔的多个模板。有关模板的更多信息，请参阅 “生成动态文件名”一节。

如果您指定的文件是现有 终端 或 /dev/console 设备，则 syslog 消息会在使用 X Window 系统时分别发送到标准输出（使用特殊 终端-handling）或控制台（使用特殊的 /dev/console-handling）。

通过网络发送 syslog 信息

rsyslog 允许您通过网络发送和接收 syslog 消息。此功能允许您管理一台计算机上多个主机的 syslog 消息。要将 syslog 信息转发到远程机器，请使用以下语法：

```
@(zNUMBER)HOST:PORT
```

其中：

- at 符号(@)表示 syslog 消息使用 UDP 协议转发到主机。若要使用 TCP 协议，请使用带符号的两个在它们之间没有空格(@@)。
- 可选的 zNUMBER 设置为 syslog 信息启用 zlib 压缩。NUMBER 属性指定压缩级别（从 1 降至最低到 9 倍）。rsyslogd 自动检查压缩增益，只有在存在压缩增益并且不会压缩 60 字节下面的消息时，才会压缩消息。
- HOST 属性指定接收所选 syslog 消息的主机。
- PORT 属性指定主机计算机的端口。

当将 IPv6 地址 指定为主机时，请将地址括在方括号([,]) 内。

例 23.4. 通过网络发送 syslog 消息

以下是通过网络转发 syslog 消息的一些操作示例（请注意，所有操作都以选择器开头，该选择器可选择具有任何优先级的所有消息）。要通过 UDP 协议转发消息到

192.168.0.1，请输入：

```
. @192.168.0.1
```

要使用端口 6514 和 TCP 协议将信息转发到"example.com"，请使用：

```
. @@example.com:6514
```

以下使用 zlib （级别 9 压缩）压缩消息，并使用 UDP 协议将它们转发到 2001:db8::1

```
. @(z9)[2001:db8::1]
```

输出频道

输出频道主要用于指定日志文件可扩展至的最大大小。这对于日志文件轮转非常有用（更多信息，请参阅 第 23.2.5 节 “Log Rotation”）。输出通道基本上是关于输出操作的信息的集合。输出频道由 $outchannel 指令定义。要在 /etc/rsyslog.conf 中定义输出频道，请使用以下语法：

```
$outchannel NAME, FILE_NAME, MAX_SIZE, ACTION
```

其中：

- NAME 属性指定输出频道的名称。
- FILE_NAME 属性指定输出文件的名称。输出通道只能写入文件，不能写入管道、终端或其他类型的输出。
- MAX_SIZE 属性表示指定文件的最大大小（在 FILE_NAME中）可增大到：这个值以字节为单位指定。
- ACTION 属性指定在MAX_SIZE 中定义的最大大小被敲击时执行的操作。

要使用定义的输出频道作为规则内的操作，请输入：

```
FILTER :omfile:$NAME
```

例 23.5. 输出频道日志轮转

以下输出显示了使用输出频道进行简单的日志轮转。首先，输出频道通过 $outchannel 指令定义：

```
$outchannel log_rotation, /var/log/test_log.log, 104857600,
/home/joe/log_rotation_script
```

然后，用于选择每个具有任何优先级的 syslog 消息并在获取的 syslog 消息上执行之前定义的输出频道的规则：

```
. :omfile:$log_rotation
```

达到限制（示例中为 100 MB）后，将执行 /home/joe/log_rotation_script。此脚本可以包含任何内容，包括将文件移动到其他文件夹、编辑特定内容或将其删除。

向特定用户发送 syslog 信息

rsyslog 可通过指定您要将信息发送到的用户的用户名（如 例 23.7 “指定多个操作”）向特定用户发送 syslog 信息。要指定多个用户，请使用逗号(,)来分隔每个用户名。要向当前登录的每个用户发送消息，请使用星号(*)。

执行程序

rsyslog 允许您对选定的 syslog 消息执行程序，并使用 system（） 调用在 shell 中执行程序。要指定要执行的程序，请为它加上脱字符号(^)前缀。因此，指定一个模板，它格式化收到的消息并将其作为一行参数传递给指定的可执行文件（有关模板的更多信息，请参阅 第 23.2.3 节 “模板”）。

```
FILTER ^EXECUTABLE; TEMPLATE
```

此处 FILTER 条件的输出由 EXECUTABLE 表示的程序处理。该程序可以是任何有效的可执行文件。将 TEMPLATE 替换为格式化模板的名称。

例 23.6. 执行程序

在以下示例中，所有带有任何优先级的 syslog 消息都会被选择，使用 模板 模板格式化并作为参数传递给 test-program 程序，然后使用提供的参数执行：

```
. ^test-program;template
```

警告

在接受来自任何主机的消息和使用 shell 执行操作时，您可能容易受到命令注入的影响。攻击者可能会尝试在您指定的程序中注入和执行命令。为避免出现任何可能的安全威胁，请仔细考虑使用 shell 执行操作。

在数据库中存储 syslog 消息

可使用数据库写入操作将选定的 syslog 消息直接写入到数据库表中。数据库写入器使用以下语法：

```
:PLUGIN:DB_HOST,DB_NAME,DB_USER,DB_PASSWORD;TEMPLATE
```

其中：

- PLUGIN 调用处理数据库写入的指定插件（例如，ommysql 插件）。
- DB_HOST 属性指定数据库主机名。
- DB_NAME 属性指定数据库的名称。
- DB_USER 属性指定数据库用户。
- DB_PASSWORD 属性指定与上述数据库用户使用的密码。
- TEMPLATE 属性指定对用于修改 syslog 消息的模板的可选用途。有关模板的更多信息，请参阅 第 23.2.3 节 “模板”。

重要

目前，r syslog 仅支持 MySQL 和 PostgreSQL 数据库。要使用 MySQL 和 PostgreSQL 数据库写入功能，请分别安装 rsyslog-mysql 和 rsyslog-pgsql 软件包。另外，请确保在 /etc/rsyslog.conf 配置文件中载入适当的模块：

```
module(load="ommysql") # Output module for MySQL support
module(load="ompgsql") # Output module for PostgreSQL support
```

有关 rsyslog 模块的详情请参考 第 23.6 节 “使用 Rsyslog 模块”。

或者，您可以使用 omlibdb 模块提供的通用数据库接口（支持：Firebird/Interbase、MS SQL、Sybase、SQLLite、Ingres、Oracle、mSQL）。

丢弃 syslog 信息

要丢弃您选择的消息，请使用 stop。

discard 操作主要用于在执行任何进一步处理前过滤出消息。如果要省略一些填写日志文件的重复消息，它可以有效。丢弃操作的结果取决于在其配置文件中指定的位置，最佳结果会将这些操作放在操作列表的顶部。请注意，一旦丢弃了一条消息，就无法在以后的配置文件行中检索该消息。

例如，以下规则丢弃所有与 local5.* 过滤器相关的信息：

```
local5.* stop
```

在以下示例中，会丢弃任何 cron syslog 信息：

```
cron.* stop
```

注意

对于 rsyslog 7 之前的版本，使用了波形符字符(~)，而不是 停止 丢弃 syslog 消息。

指定多个操作

对于每个选择器，您可以指定多个操作。要为一个选择器指定多个操作，请在单独的行中写每个操作，并在它前面添加一个 符号(&)字符：

```
FILTER ACTION
& ACTION
& ACTION
```

指定多个操作可提高所需结果的整体性能，因为指定的选择器只需要评估一次。

例 23.7. 指定多个操作

在以下示例中，所有具有关键优先级(crit)的内核 syslog 消息都发送到用户 user1，由模板 临时处理并传递 至 test-program 可执行文件，并通过 UDP 协议转发到 192.168.0.1。

```
kern.=crit user1
& ^test-program;temp
& @192.168.0.1
```

任何操作都可以跟随格式化消息的模板。要指定模板，请为带有分号(;)的操作后缀并指定模板的名称。有关模板的更多信息，请参阅 第 23.2.3 节 “模板”。

警告

模板必须在操作中使用之前定义，否则将被忽略。换句话说，模板定义应始终在 /etc/rsyslog.conf 中的规则定义之前。

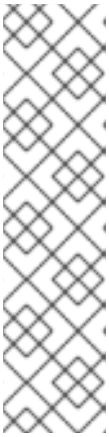
23.2.3. 模板

rsyslog 生成的任何输出都可以通过使用 模板 来根据您的需要修改和格式化。要在 /etc/rsyslog.conf 中创建模板，请使用以下语法：

```
template(name=”TEMPLATE_NAME” type=”string” string="text %PROPERTY% more text"
[option.OPTION="on"])
```

其中：

- template（） 是引入定义模板的块的指令。
- TEMPLATE_NAME 强制参数用于引用模板。请注意， TEMPLATE_NAME 应该是唯一的。
- type required 参数可以获取以下值之一： "list"、 "subtree"、 "string" 或 "plugin"。
- string 参数是实际的模板文本。在此文本中， 可以使用特殊字符， 如 \n 表示换行符， 或使用 \r 表示回车。如果要按字面意义使用这些字符， 则必须转义其他字符， 如%或 "。在此文本中， 可以使用特殊字符， 如 \n 表示新行， 或者使用 \r 表示回车。如果要按字面意义使用这些字符， 则必须转义其他字符， 如 % 或 "。
- 在两个百分比符号之间指定的文本(%)指定 允许您 访问 syslog 消息的特定内容的属性。有关属性的详情请参考 “属性”一节。
- OPTION 属性指定修改模板功能的任何选项。当前支持的模板选项为 sql 和 stdsql （用于将文本格式化为 SQL 查询）， 或 json 格式化适合 JSON 处理的文本， 以及设置属性名称 区分大小写 的区分大小写。



注意

请注意， 数据库作者将检查模板中是否指定了 sql 或 stdsql 选项。如果没有， 数据库作者不会执行任何操作。这是为了避免可能的任何安全威胁， 如 SQL 注入。

如需更多信息， 请参阅 第 23.2.2 节 “操作” 中的存储数据库中 syslog 信息部分。

生成动态文件名

模板可用于生成动态文件名。通过将属性指定为文件路径的一部分， 将为每个唯一属性创建一个新文件， 这是对 syslog 消息进行分类的一种便捷方式。

例如， 使用 timegenerated 属性（从消息中提取时间戳）来为每个 syslog 消息生成唯一的文件名：

```
template(name="DynamicFile" type="list") {
constant(value="/var/log/test_logs/")
property(name="timegenerated")
constant(value"-test.log")
}
```

请记住，$template 指令仅指定模板。您必须在规则内使用它才能生效。在 /etc/rsyslog.conf 中，在操作定义中使用问号(?)来标记动态文件名模板：

```
. ?DynamicFile
```

属性

模板中定义的属性（在 20% 符号之间）通过使用 属性替换器 访问 syslog 消息的不同内容。要在模板中定义属性（最好使用两个引号("...")），请使用以下语法：

```
%PROPERTY_NAME:FROM_CHAR:TO_CHAR:OPTION%
```

其中：

- PROPERTY_NAME 属性指定属性的名称。可以在 rsyslog.conf(5) 手册页的 Available Properties 部分下找到所有可用属性及其详细描述的列表。
- FROM_CHAR 和 TO_CHAR 属性表示指定属性将对其操作的一系列字符。或者，可以使用正则表达式来指定一系列字符。为此，请将字母 R 设置为 FROM_CHAR 属性，并将您所需的正则表达式指定为 TO_CHAR 属性。
- OPTION 属性指定任何属性选项，例如 小写 选项，可将输入转换为小写选项。有关所有可用属性选项及其详细描述的列表，请参阅Coreject Options 部分下的 rsyslog.conf(5) 手册页。

以下是简单属性的一些示例：

- 以下属性获取 syslog 消息的整个消息文本：

  ```
  %msg%
  ```

- 以下属性获取系统日志消息消息文本的前两个字符：

```
%msg:1:2%
```

- 以下属性获取 syslog 消息的完整消息文本，并丢弃其最后一行回馈字符：

```
%msg:::drop-last-lf%
```

- 以下属性获取收到 syslog 消息时生成的时间戳的前 10 个字符，并根据 RFC 3999 日期标准对其进行格式化。

```
%timegenerated:1:10:date-rfc3339%
```

模板示例

本节介绍 rsyslog 模板的几个示例。

例 23.8 “详细 syslog 消息模板” 显示一个模板，用于格式化 syslog 消息，以便其输出消息的严重性、设备、消息收件时间的时间戳、主机名、消息标记、消息文本，并且以新行结尾。

例 23.8. 详细 syslog 消息模板

```
template(name="verbose" type="list") {
property(name="syslogseverity")
property(name="syslogfacility")
property(name="timegenerated")
property(name="HOSTNAME")
property(name="syslogtag")
property(name="msg")
constant(value="\n")
}
```

例 23.9 “墙壁消息模板” 显示类似传统墙壁消息的模板（将发送给登录的每个用户并将其 mesg(1) 权限设置为 yes）。此模板在新行（使用 \r 和 \n）上输出消息文本，以及主机名、消息标记和时间戳（使用 \r 和 \n）并环环 bell（使用 \7）。

例 23.9. 墙壁消息模板

```
template(name="wallmsg" type="list") {
constant(value="\r\n\7Message from syslogd@")
property(name="HOSTNAME")
constant(value=" at ")
property(name="timegenerated")
constant(value=" ...\r\n ")
property(name="syslogtag")
```

```
  constant(value=" ")
  property(name="msg")
  constant(value="\r\n")
}
```

例 23.10 "数据库格式化的消息模板" 显示格式化 syslog 消息的模板，以便它可以用作数据库查询。注意在作为 template 选项指定的模板末尾使用sql 选项。它告知数据库作者将消息格式化为 MySQL SQL 查询。

例 23.10. 数据库格式化的消息模板

```
template(name="dbFormat" type="list" option.sql="on") {
constant(value="insert into SystemEvents (Message, Facility, FromHost, Priority,
DeviceReportedTime, ReceivedAt, InfoUnitID, SysLogTag)")
constant(value=" values ('")
property(name="msg")
constant(value="', ")
property(name="syslogfacility")
constant(value=", '")
property(name="hostname")
constant(value="', ")
property(name="syslogpriority")
constant(value=", '")
property(name="timereported" dateFormat="mysql")
constant(value="', '")
property(name="timegenerated" dateFormat="mysql")
constant(value="', ")
property(name="iut")
constant(value=", '")
property(name="syslogtag")
constant(value="')")
}
```

rsyslog 还包含由 RSYSLOG_ 前缀标识的一组预定义模板。它们为 syslog 的使用保留，建议不要使用此前缀创建模板以避免冲突。下表显示了这些预定义模板及其定义：

RSYSLOG_DebugFormat

用于对属性问题进行故障排除的特殊格式。

```
template(name="RSYSLOG_DebugFormat" type="string" string="Debug line with all
properties:\nFROMHOST: '%FROMHOST%', fromhost-ip: '%fromhost-ip%', HOSTNAME:
'%HOSTNAME%', PRI: %PRI%,\nsyslogtag '%syslogtag%', programname:
'%programname%', APP-NAME: '%APP-NAME%', PROCID: '%PROCID%', MSGID:
'%MSGID%',\nTIMESTAMP: '%TIMESTAMP%', STRUCTURED-DATA: '%STRUCTURED-
DATA%',\nmsg: '%msg%'\nescaped msg: '%msg:::drop-cc%'\nrawmsg: '%rawmsg%'\n\n")
```

RSYSLOG_SyslogProtocol23Format

IETF 的 internet-draft ietf-syslog-protocol-23 中指定的格式，被假定为成为新的 syslog 标准 RFC。

```
template(name="RSYSLOG_SyslogProtocol23Format" type="string" string="%PRI%1
%TIMESTAMP:::date-rfc3339% %HOSTNAME% %APP-NAME% %PROCID% %MSGID%
%STRUCTURED-DATA% %msg%\n ")
```

RSYSLOG_FileFormat

现代日志文件格式类似于 TraditionalFileFormat，但具有高精确度时间戳和时区信息。

```
template(name="RSYSLOG_FileFormat" type="list") {
property(name="timestamp" dateFormat="rfc3339")
constant(value=" ")
property(name="hostname")
constant(value=" ")
property(name="syslogtag")
property(name="msg" spifno1stsp="on" )
property(name="msg" droplastlf="on" )
constant(value="\n")
}
```

RSYSLOG_TraditionalFileFormat

较旧的默认日志文件格式，具有低精确度时间戳。

```
template(name="RSYSLOG_TraditionalFileFormat" type="list") {
property(name="timestamp")
constant(value=" ")
property(name="hostname")
constant(value=" ")
property(name="syslogtag")
property(name="msg" spifno1stsp="on" )
property(name="msg" droplastlf="on" )
constant(value="\n")
}
```

RSYSLOG_ForwardFormat

具有高精确度时间戳和时区信息的转发格式。

```
template(name="ForwardFormat" type="list") {
constant(value="<")
property(name="pri")
constant(value=">")
property(name="timestamp" dateFormat="rfc3339")
constant(value=" ")
```

```
property(name="hostname")
constant(value=" ")
property(name="syslogtag" position.from="1" position.to="32")
property(name="msg" spifno1stsp="on" )
property(name="msg")
}
```

RSYSLOG_TraditionalForwardFormat

具有低精确度时间戳的传统转发格式。

```
template(name="TraditionalForwardFormat" type="list") {
constant(value="<")
property(name="pri")
constant(value=">")
property(name="timestamp")
constant(value=" ")
property(name="hostname")
constant(value=" ")
property(name="syslogtag" position.from="1" position.to="32")
property(name="msg" spifno1stsp="on" )
property(name="msg")
}
```

23.2.4. 全局指令

全局指令是应用到 rsyslogd 守护进程的 配置选项。它们通常为影响 rsyslogd 守护进程行为或下列规则的特定预定义变量指定一个值。所有全局指令都包含在 全局 配置块中。以下是指定日志消息覆盖本地主机名的 global 指令示例：

```
global(localHostname="machineXY")
```

您可以在 /etc/rsyslog.conf 配置文件中定义多个指令。指令会影响所有配置选项的行为，直到检测到同一指令的另一次出现。全局指令可用于配置操作、队列和调试。您可以在 "在线文档"一节 中找到所有可用配置指令的完整列表。目前，开发了新的配置格式来取代基于 $ 的语法（请参阅 第 23.3 节 "使用新配置格式"）。但是，典型的全局指令仍作为传统格式提供支持。

23.2.5. Log Rotation

以下是 /etc/logrotate.conf 配置文件示例：

```
# rotate log files weekly
weekly
# keep 4 weeks worth of backlogs
```

```
rotate 4
# uncomment this if you want your log files compressed
compress
```

示例配置文件中的所有行都定义了应用到每个日志文件的全局选项。在我们的示例中，日志文件每周轮转，轮转日志文件保留四周，并且所有轮转的日志文件都会压缩 到 .gz 格式。任何以井号(#)开头的行都是注释，不会处理。

您可以为特定的日志文件定义配置选项，并将它放到全局选项下。不过，建议为 /etc/logrotate.d/ 目录中的任何特定日志文件创建单独的配置文件，并在其中定义任何配置选项。

以下是 /etc/logrotate.d/ 目录中的配置文件示例：

```
/var/log/messages {
  rotate 5
  weekly
  postrotate
  /usr/bin/killall -HUP syslogd
  endscript
}
```

此文件中的配置选项仅特定于 /var/log/messages 日志文件。此处指定的设置会尽可能覆盖全局设置。因此，轮转的 /var/log/messages 日志文件将保留五周，而不是全局选项中定义的四周。

以下是您可以在 logrotate 配置文件中指定的一些指令列表：

- weekly - 指定每周要执行的日志文件轮转。类似的指令包括：
  - daily
  - monthly
  - 每年
- 压缩 - 启用轮转日志文件的 压缩。类似的指令包括：

- nocompress
- presscmd - 指定用于压缩的命令。
- uncompresscmd
- pressext - 指定用于压缩的扩展名。
- pressoptions - 指定要传递给使用的压缩程序的任何选项。
- delaycompress - 将日志文件的压缩加载到下一次日志文件轮转。

- 轮转 INTEGER - 指定日志文件在删除或邮寄到特定地址之前进行的轮转数量。如果指定了 0，则删除旧的日志文件，而不是轮转。

- 邮件 ADDRESS - 此选项允许将 rotate 指令所定义多次的日志文件发送至指定地址。类似的指令包括：
  - nomail
  - 邮件优先 - 指定将发送刚才轮转的日志文件，而不是即将推出的日志文件。
  - maillast - 指定将要对扩展的日志文件进行邮件传递，而不是刚刚轮转的日志文件。这是启用 邮件 时的默认选项。

有关指令和各种配置选项的完整列表，请查看 logrotate(5) 手册页。

23.2.6. 增加 Open 文件的限制

在某些情况下，r syslog 超过打开文件的最大数量的限制。因此，r syslog 无法打开新文件。

在 rsyslog 中增加打开文件的限制：

使用以下内容创建 /etc/systemd/system/rsylog.service.d/increase_nofile_limit.conf 文件：

```
[Service]
LimitNOFILE=16384
```

23.3. 使用新配置格式

在 rsyslog 版本 7 中，默认为 Red Hat Enterprise Linux 7 安装在 rsyslog 软件包中，引入了新的配置语法。这种新的配置格式旨在通过不允许某些无效构造来更强大、更直观，并防止常见错误。语法增强由依赖于 RainerScript 的新配置处理器启用。旧格式仍被完全支持，它在 /etc/rsyslog.conf 配置文件中默认使用。

RainerScript 是脚本语言，设计用于处理网络事件和配置事件处理器，如 rsyslog。RainerScript 首先用于定义基于表达式的过滤器，请参阅 例 23.3 “基于表达式的过滤器”。rsyslog 版本 7 中的 RainerScript 版本实施了 input() 和 ruleset() 语句，允许使用新语法编写 /etc/rsyslog.conf 配置文件。新语法的主要区别在于，它的结构更强；参数作为参数传递到语句，如输入、操作、模板和模块加载。选项的范围受块限制。这增强了可读性，并减少了配置错误导致的错误数量。性能也显著提高。部分功能在两种语法中都公开，有些功能仅在新语法中公开。

将编写的配置与旧式参数进行比较：

```
$InputFileName /tmp/inputfile
$InputFileTag tag1:
$InputFileStateFile inputfile-state
$InputRunFileMonitor
```

以及使用新 format 语句的相同配置：

```
input(type="imfile" file="/tmp/inputfile" tag="tag1:" statefile="inputfile-state")
```

这可显著减少配置中使用的参数数量，提高可读性，还提供更高的执行速度。有关 RainerScript 语句和参数的详情请参考 “在线文档”一节。

23.3.1. ruleset

保留特殊指令时，r syslog 会按照 规则 的定义处理消息，该规则由过滤器条件组成，并在条件为 true 时要执行的操作。使用传统编写的 /etc/rsyslog.conf 文件，会按照每条输入消息的外观对所有规则

进行评估。这个过程以第一个规则开始，持续到所有规则都已处理或消息被其中一个规则丢弃为止。

但是，规则可以分组为多个序列，称为规则集。使用规则集时，您可以通过定义一组绑定到特定输入的不同操作集，将某些规则的效果限制为所选输入或提高 rsyslog 的性能。换而言之，可以跳过某些类型消息中无可避免评估为假的过滤条件。/etc/rsyslog.conf 中的旧规则集定义如下所示：

```
$RuleSet rulesetname
rule
rule2
```

当定义了另一个规则，或者调用默认规则集时，该规则会结束，如下所示：

```
$RuleSet RSYSLOG_DefaultRuleset
```

对于 rsyslog 7 中的新配置格式，因此为此操作保留了 input() 和 ruleset() 语句。/etc/rsyslog.conf 中的新格式规则集定义如下所示：

```
ruleset(name="rulesetname") {
  rule
  rule2
  call rulesetname2
  &hellip;
}
```

将 rulesetname 替换为规则集的标识符。ruleset 名称不能以 RSYSLOG_ 开始，因为此命名空间保留给 rsyslog 使用。RSYSLOG_DefaultRuleset 然后，如果消息没有分配其他规则集，则定义要执行的默认规则集。使用 规则和 rule2，您可以使用上述过滤器操作格式定义规则。使用 call 参数时，您可以通过从其他规则集块中调用它们来嵌套规则集。

创建规则集后，您需要指定它将应用到哪些输入：

```
input(type="input_type" port="port_num" ruleset="rulesetname");
```

此处您可以通过 input_type （即收集消息的输入模块）或 port_num - 端口号来识别输入消息。可以为 input() 指定 文件或 标签 等其他参数。将 rulesetname 替换为针对消息评估的规则集的名称。如果没有显式绑定到规则集，则会触发默认规则集。

您还可以使用旧格式来定义规则集，如需更多信息，请参阅 “在线文档”一节。

例 23.11. 使用 ruleset

以下规则集确保对来自不同端口的远程消息进行不同的处理。在 /etc/rsyslog.conf 中添加以下内容：

```
ruleset(name="remote-6514") {
  action(type="omfile" file="/var/log/remote-6514")
}

ruleset(name="remote-601") {
  cron.* action(type="omfile" file="/var/log/remote-601-cron")
  mail.* action(type="omfile" file="/var/log/remote-601-mail")
}

input(type="imtcp" port="6514" ruleset="remote-6514");
input(type="imtcp" port="601" ruleset="remote-601");
```

上例中所示的规则集从两个端口定义远程输入的日志目的地，如果是端口 601，则消息按照设备进行排序。然后，TCP 输入被启用并绑定到 ruleset。请注意，您必须加载所需的模块(imtcp)才能使此配置正常工作。

23.3.2. 与 sysklogd 的兼容性

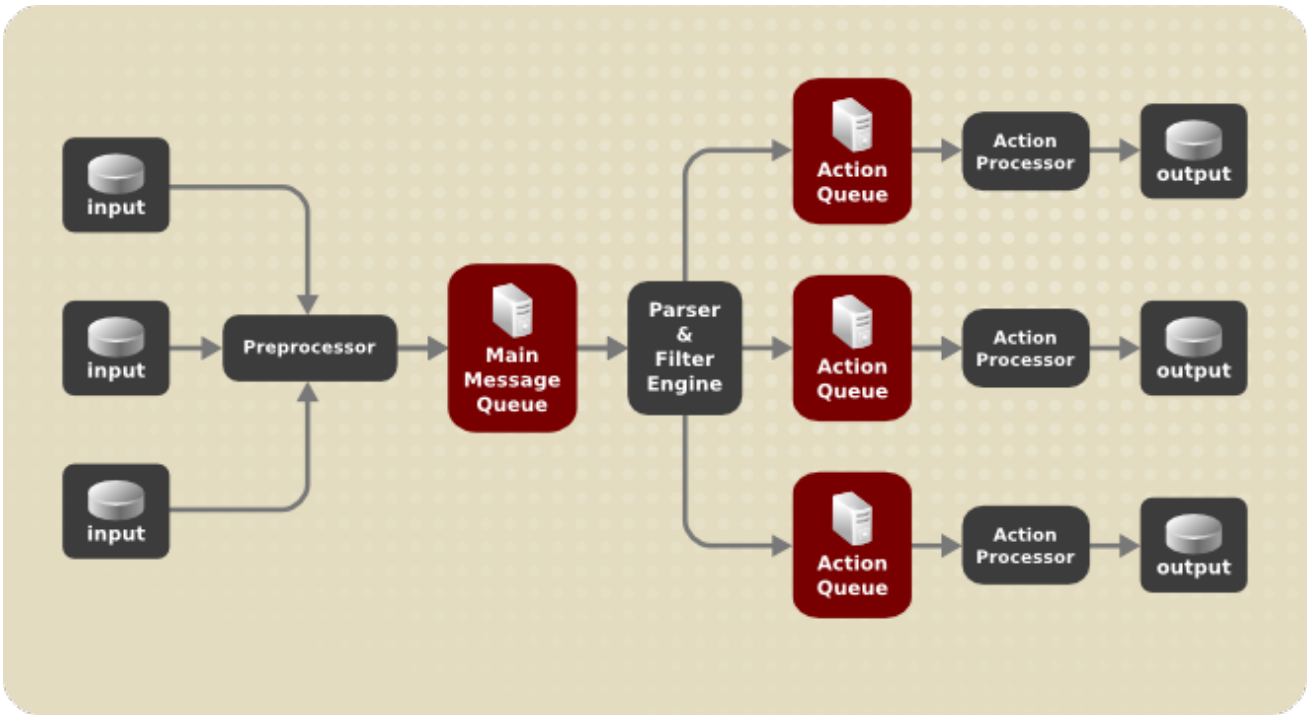
通过 -c 选项指定的兼容性模式存在于 rsyslog 版本 5 中，但不存在于版本 7 中。此外，sysklogd 风格的命令行选项已弃用，应避免使用这些命令行选项配置 rsyslog。但是，您可以使用多个模板和指令将 rsyslogd 配置为模拟类似于 sysklogd 的行为。

有关各种 rsyslogd 选项的更多信息，请参阅 rsyslogd(8)手册页。

23.4. 在 RSYSLOG 中使用队列

队列用于在 rsyslog 组件之间传递内容，大部分是 syslog 消息。利用队列，rsyslog 可以同时处理多个消息，并一次性对单个消息应用多个操作。rsyslog 中的数据流可以说明如下：

图 23.1. Rsyslog 中的消息流



每当 rsyslog 收到消息时，它会将此消息传递到预处理器，然后将它放入 主消息队列。等待那里的消息被降序并传递到 规则处理器。

规则处理器是分析和过滤引擎。此处应用 /etc/rsyslog.conf 中定义的规则。根据这些规则，规则处理器将评估要执行的操作。每个操作都有自己的操作队列。消息通过此队列传递到创建最终输出的对应操作处理器。请注意，此时可在一条消息上同时运行多个操作。为此，消息会重复并传递给多个操作处理器。

每个操作只能有一个队列。根据配置，可以在无需操作排队的情况下将消息发送到操作处理器。这是直接队列的行为 （请参见下文）。如果输出操作失败，操作处理器会通知操作队列，然后重新取一个未处理的元素，并在某一时间间隔后再次尝试该操作。

总之，队列在 rsyslog 中有两个位置：作为单个 主消息队列 在规则处理器前面，还是在不同类型的输出操作的前面作为 操作队列。队列提供两个主要优势，它们都会导致消息处理性能提高：

- 它们充当在 rsyslog结构中 分离 生产者和消费者的缓冲
- 它们允许并行化对消息执行的操作

此外，也可使用多个 指令配置队列，以便为您的系统提供最佳性能。以下小节中介绍了这些配置选项。

警告

如果输出插件无法发送消息，它将存储在前面的消息队列中。如果队列填满，则输入块直到其不再完整为止。这将阻止新消息通过被阻止的队列记录。如果没有单独的操作队列，这可能会产生严重后果，例如防止 SSH 记录，进而可以阻止 SSH 访问。因此，建议将专用的操作队列用于通过网络或数据库转发的输出。

23.4.1. 定义队列

根据消息的存储位置，有多种类型的队列：直接、内存中、磁盘 和磁盘辅助内存中队列使用最广。您可以为主消息队列选择这些类型之一，也可以选择用于操作队列。在 /etc/rsyslog.conf 中添加以下内容：

```
object(queue.type="queue_type")
```

通过添加此项，您可以为以下内容应用设置：

- 主消息队列：将 对象 替换为 main_queue
- 操作队列：将 object 替换为 操作
- ruleset：将 对象 替换为 ruleet

使用 直接、链接列表 或 固定阵列（ 内存中队列）或 磁盘 之一替换 queue_type。

主消息队列的默认设置是 FixedArray 队列，限制为 10,000 个消息。操作队列默认设置为直接队列。

直接队列

对于许多简单的操作，如将输出写入本地文件时，不需要在操作前构建队列。为了避免排队，请使用：

```
object(queue.type="Direct")
```

将 object 替换为 main_queue、操作 或 规则集，以将此选项分别用于主消息队列、操作队列或规则集。使用直接队列时，消息会直接直接从制作者传递至消费者。

磁盘队列

磁盘队列严格将消息存储在硬盘驱动器中，这使其高度可靠，同时也是所有可能的队列模式中最慢的消息。此模式可用于防止丢失高度重要的日志数据。但是，大多数用例中不建议使用磁盘队列。要设置磁盘队列，请在 /etc/rsyslog.conf 中输入以下内容：

```
object(queue.type="Disk")
```

将 object 替换为 main_queue、操作 或 规则集，以将此选项分别用于主消息队列、操作队列或规则集。可使用以下配置指令修改队列的默认大小：

```
object(queue.size="size")
```

其中 size 代表磁盘队列部分的指定大小。定义的大小限制不受限制，rsyslog 始终会编写一个完整的队列条目，即使它违反了大小限制。磁盘队列的每个部分都与单个文件匹配。这些文件的命名指令如下：

```
object(queue.filename="name")
```

这会为文件设置一个名称前缀，后跟一个从 1 开始并递增每个文件的 7 位数字。

```
object(queue.maxfilesize="size")
```

磁盘队列以部分形式编写，默认大小为 1 MB。指定要使用其他值的大小。

内存中队列

使用内存中队列时，requeued 消息保存在内存中，这使得进程变得非常快速。如果计算机重启或关机，则排队的数据将丢失。但是，您可以使用 操作(queue.saveonshutdown="on") 设置在关闭前保存数据。内存队列有两种类型：

- FixedArray queue - 主消息队列的默认模式，限制为 10,000 个元素。这种类型的队列使用固定的预分配的阵列，该阵列包含指向队列元素的指针。由于这些指示符，即使队列为空，也会消耗一定量的内存。但是，FixedArray 提供了最佳的运行时性能，并在您期望相对较少的排队消息和高性能时是最佳的。

- LinkedList 队列 - 在这里，所有结构都在链接列表中动态分配，因此内存仅在需要时分配。LinkedList 队列处理偶尔的消息突发性。

通常情况下，当有疑问时使用 LinkedList 队列。与 FixedArray 相比，它会消耗较少的内存并降低处理开销。

使用以下语法配置内存队列：

```
object(queue.type="LinkedList")
```

```
object(queue.type="FixedArray")
```

将 object 替换为 main_queue、操作 或 规则集，以将此选项分别用于主消息队列、操作队列或规则集。

磁盘辅助内存中队列

磁盘和内存中队列都有其优势，而 rsyslog 则允许您将其组合 到磁盘辅助内存中队列中。为此，请配置普通内存中队列，然后将 queue.filename="file_name" 指令添加到其块中，以定义磁盘协助的文件名。然后，这个队列变为 diskassisted，这意味着它将内存队列与磁盘队列相结合，从而协同工作。

如果内存中队列已满或者需要在关机后保留，则激活磁盘队列。使用磁盘辅助队列，您可以同时设置特定于磁盘和内存的特定配置参数。这种队列或许是最常用的，对于潜在的长时间运行且不可靠的操作特别有用。

要指定磁盘辅助内存队列的功能，请使用所谓的水位线：

```
object(queue.highwatermark="number")
```

```
object(queue.lowwatermark="number")
```

将 object 替换为 main_queue、操作 或 规则集，以将此选项分别用于主消息队列、操作队列或规则集。将 number 替换为一组 enqueued 消息。当内存中队列达到高水位线定义的数量时，它开始将消息写入磁盘并继续，直到内存中队列大小下降到使用低水位线定义的数字。正确设置水位线可最大程度减少不必要的磁盘写入，但也为消息突发保留内存空间，因为写入磁盘文件非常长。因此，高水位线必须小于通过 queue.size 设置的完整队列容量。高水位线与总体队列大小之间的区别在于为消息突发保留的备用内存缓冲区。另一方面，将高水位线设置得过低将会带来不必要的磁盘帮助。

例 23.12. 可靠将日志消息转发至服务器

rsyslog 通常用于维护集中式日志记录系统，其中日志消息通过网络转发到服务器。为了避免服务器不可用时消息丢失，建议为转发操作配置操作队列。这样，发送失败的消息会存储在本地，直到服

务器再次可访问为止。请注意，此类队列无法用于使用 UDP 协议的连接。

转发到单一服务器

假设任务是将日志消息从系统转发到主机名为 example.com 的服务器，并且配置操作队列以缓冲消息（以防服务器中断）。要做到这一点，请执行以下步骤：

1. 在 /etc/rsyslog.conf 中使用 以下配置，或在 /etc/rsyslog.d/ 目录中创建包含以下内容的文件：

```
. action(type=”omfwd”
queue.type=”LinkedList”
queue.filename=”example_fwd”
action.resumeRetryCount="-1"
queue.saveonshutdown="on"
Target="example.com" Port="6514" Protocol="tcp")
```

其中：

- queue.type 启用 LinkedList 内存中队列，
- queue.filename 定义磁盘存储，在这种情况下，备份文件在 /var/lib/rsyslog/ 目录中创建，前缀为 example_fwd。
- action.resumeRetryCount= "-1" 设置阻止 rsyslog 在重试连接时丢弃消息（如果服务器没有响应）
- 如果 rsyslog 关闭，启用的 queue.saveonshutdown 可保存内存中数据。
- 最后一行使用可靠的 TCP 发送将所有收到的消息转发到日志记录服务器，端口规格是可选的。

使用上述配置时，如果远程服务器无法访问，rsyslog 将消息保留在内存中。只有在 rsyslog 用尽配置的内存队列空间或需要关闭时，才会在磁盘上创建文件，这将提高系统性能。

转发到多个服务器

将日志信息转发到多个服务器的过程与前面的步骤类似：

1. 每个目标服务器都需要独立的转发规则、操作队列规格和磁盘上的备份文件。例如，在 /etc/rsyslog.conf 中使用 以下配置，或在 /etc/rsyslog.d/ 目录中创建包含以下内容的文件：

```
. action(type="omfwd"
   queue.type="LinkedList"
   queue.filename="example_fwd1"
   action.resumeRetryCount="-1"
   queue.saveonshutdown="on"
   Target="example1.com" Protocol="tcp")
. action(type="omfwd"
   queue.type="LinkedList"
   queue.filename="example_fwd2"
   action.resumeRetryCount="-1"
   queue.saveonshutdown="on"
   Target="example2.com" Protocol="tcp")
```

23.4.2. 为 rsyslog 日志文件创建新目录

rsyslog 作为 syslogd 守护进程运行，并由 SELinux 管理。因此，rsyslog 需要写入到的所有文件都必须具有相应的 SELinux 文件上下文。

创建新工作目录

1. 如果需要使用不同的目录存储工作文件，请按如下所示创建一个目录：

```
~]# mkdir /rsyslog
```

2. 安装工具来管理 SELinux 策略：

```
~]# yum install policycoreutils-python
```

3. 将 SELinux 目录上下文类型设置为与 /var/lib/rsyslog/ 目录相同：

```
~]# semanage fcontext -a -t syslogd_var_lib_t /rsyslog
```

4. 应用 SELinux 上下文：

```
~]# restorecon -R -v /rsyslog
restorecon reset /rsyslog context unconfined_u:object_r:default_t:s0-
>unconfined_u:object_r:syslogd_var_lib_t:s0
```

5. 如果需要，请按如下所示检查 SELinux 上下文：

```
~]# ls -Zd /rsyslog
drwxr-xr-x. root root system_u:object_r:syslogd_var_lib_t:s0  /rsyslog
```

6. 根据需要创建子目录。例如：

```
~]# mkdir /rsyslog/work/
```

将使用与父目录相同的 SELinux 上下文创建子目录。

7. 在 /etc/rsyslog.conf 立即添加以下行才能生效：

```
global(workDirectory="/rsyslog/work")
```

此设置将保持有效，直到在解析配置文件时遇到下一个 WorkDirectory 指令。

23.4.3. 管理队列

可以进一步配置所有类型的队列，以符合您的要求。您可以使用多个指令来修改操作队列和主消息队列。目前，可用的队列参数超过 20 个，请参阅 "在线文档"一节。其中一些设置通常使用，另一些（如 worker 线程管理）提供对队列行为的更紧密控制，并保留给高级用户。使用高级设置，您可以优化 rsyslog的性能、调度排队或修改系统关闭时队列的行为。

限制队列大小

您可以使用以下设置限制队列可以包含的信息数量：

```
object(queue.highwatermark="number")
```

将 object 替换为 main_queue、操作 或 规则集，以将此选项分别用于主消息队列、操作队列或规则集。将 number 替换为一组 enqueued 消息。您只能将队列大小设置为消息数，而不是其实际内存大小。默认队列大小为主消息队列和规则集队列的 10,000 个消息，而 1000 则表示操作队列。

默认情况下，磁盘支持的队列没有限制，且无法使用这个指令进行限制，但您可以使用以下设置来为它们保留物理磁盘空间（以字节为单位）：

```
object(queue.maxdiskspace="number")
```

使用 main_queue、action 或 rules et 替换 object。当按数字指定的大小限制被命中时，消息会被丢弃，直到 dequeued 消息释放足够数量的空间。

丢弃消息

当队列到达特定数量的消息时，您可以丢弃不太重要的消息，以便在队列中为优先级较高的条目节省空间。启动丢弃过程的阈值可使用所谓的 discard 标记来设置：

```
object(queue.discardmark="number")
```

使用 MainMsg 或 Action 替换 object，以分别将此选项用于主消息队列或操作队列。此处，number 代表开始丢弃过程时必须要处于队列中的多个消息。要定义要丢弃的信息，请使用：

```
object(queue.discardseverity="number")
```

将 number 替换为以下对应优先级之一：7 (debug)、6 (info)、5 （通知）、4（警告）、3 (err)、2(crit)、1 （证书）或 0 (emerg)。使用此设置时，在到达丢弃标记后，将立即从队列中删除优先级低于定义的新传入和排队消息。

使用时间戳

您可以将 rsyslog 配置为在特定时间段内处理队列。例如，您可以使用此选项将一些处理转移到非高峰时段。要定义时间段，请使用以下语法：

```
object(queue.dequeuetimebegin="hour")
```

```
object(queue.dequeuetimeend="hour")
```

通过小时，您可以指定绑定时间段的小时。使用 24 小时格式（无分钟）。

配置工作程序线程

worker 线程在 enqueued 消息上执行指定的操作。例如，在主消息队列中，worker 任务是将过滤器逻辑应用到每一传入消息，并将它们排队到相关的操作队列。当消息到达时，会自动启动 worker 线程。当消息数量达到特定数字时，将打开另一个 worker 线程。要指定这个号码，请使用：

```
object(queue.workerthreadminimummessages="number")
```

将 number 替换为将触发补充工作线程的消息。例如，当数字设置为 100 时，当到达 100 条消息时会启动一个新的 worker 线程。当到达 200 条信息时，第三个 worker 线程启动，以此类推。但是，并行运行的太多工作线程会变得无效，因此您可以使用以下方法限制它们的最大数量：

```
object(queue.workerthreads=”number”)
```

其中，数字表示可以并行运行的最大工作线程数。对于主消息队列，默认限制为 1 线程。启动工作线程后，它将持续运行，直到出现不活跃的超时为止。要设置超时长度，请输入：

```
object(queue.timeoutworkerthreadshutdown=”time”)
```

将 time 替换为以毫秒为单位设置的持续时间。指定没有新消息的时间，并在该消息后关闭 worker 线程。默认设置为一分钟。

批量取消

要提高性能，您可以将 rsyslog 配置为一次性取消排队多个消息。要为此类 dequeue 设置上限，请使用：

```
$object(queue.DequeueBatchSize= ”number”)
```

将 number 替换为一次可以取消排队的消息的最大数量。请注意，较高的设置结合了更多允许的工作线程数，可增加内存消耗。

终止队列

当终止仍包含信息的队列时，您可以通过为 worker 线程指定完成队列处理的时间间隔来尝试最小化数据丢失：

```
object(queue.timeoutshutdown=”time”)
```

以毫秒为单位指定时间。如果在该周期之后仍然有一些 enqueued 消息，worker 会完成当前数据元素，然后终止。因此，未经处理的消息会丢失。可以为 worker 设置另一个时间间隔以完成最终元素：

```
object(queue.timeoutactioncompletion=”time”)
```

如果这个超时过期，所有剩余的工作程序都会被关闭。要在关闭时保存数据，请使用：

```
object(queue.saveonshutdown=”on”)
```

如果设置，则所有队列元素都会保存到磁盘，然后 rsyslog 终止。

23.4.4. 将 New Syntax 用于 rsyslog 队列

在 rsyslog 7 中提供的新语法中，队列在 action（） 对象中定义，可以单独或在 /etc/rsyslog.conf 中的规则集内使用。操作队列的格式如下：

```
action(type="action_type "queue.size="queue_size" queue.type="queue_type"
queue.filename="file_name"
```

使用要执行该操作的模块的名称替换 action_type，并将 queue_size 替换为队列可以包含的最大消息数。对于 queue_type，请选择 disk，或从其中一个内存中队列中选择：直接、链接列表 或 固定阵列。对于 file_name 仅指定文件名，而不是路径。请注意，如果创建新目录以存放日志文件，则必须设置 SELinux 上下文。示例请查看 第 23.4.2 节 “为 rsyslog 日志文件创建新目录”。

例 23.13. 定义操作队列

要使用基于异步链接列表的操作队列配置输出操作，可以保存最多 10,000 条信息，请输入以下命令：

```
action(type="omfile" queue.size="10000" queue.type="linkedlist" queue.filename="logfile")
```

直接操作队列的 rsyslog 7 语法如下：

```
. action(type="omfile" file="/var/lib/rsyslog/log_file
  )
```

带有多个参数的操作队列的 rsyslog 7 语法可以编写如下：

```
. action(type="omfile"
      queue.filename="log_file"
      queue.type="linkedlist"
      queue.size="10000"
  )
```

将使用默认工作目录或要设置的最后一个工作目录。如果需要使用不同的工作目录，请在操作队列前添加以下行：

```
global(workDirectory="/directory")
```

例 23.14. 使用新语法转发到单一服务器

以下示例基于 转发到单一服务器 过程，以便显示传统 sysntax 和 rsyslog 7 语法之间的区别。The omfwd 插件用于通过 UDP 或 TCP 提供转发。默认值为 UDP。由于插件内置在其中，因此无需加载。

在 /etc/rsyslog.conf 中使用 以下配置，或在 /etc/rsyslog.d/ 目录中创建包含以下内容的文件：

```
. action(type="omfwd"
  queue.type="linkedlist"
  queue.filename="example_fwd"
  action.resumeRetryCount="-1"
  queue.saveOnShutdown="on"
  target="example.com" port="6514" protocol="tcp"
  )
```

其中：

- queue.type="linkedlist" 启用 LinkedList 内存队列，
- queue.filename 定义磁盘存储。在上一全局 workDirectory 指令指定的工作目录中，使用 example_fwd 前缀创建备份文件，
- action.resumeRetryCount -1 设置阻止 rsyslog 在重试连接时丢弃消息（如果服务器没有响应）
- 如果 rsyslog 关闭，启用 queue.saveOnShutdown="on" 将保存内存中数据，
- 最后一行将所有收到的消息转发到日志记录服务器，端口规格是可选的。

23.5. 在日志记录服务器上配置 RSYSLOG

rsyslog 服务提供运行日志记录服务器和将各个系统配置为将其日志文件发送到日志记录服务器的功能。有关客户端 rsyslog 配置的详情，请查看 例 23.12 “可靠将日志消息转发至服务器”。

rsyslog 服务必须安装到您要用作记录服务器的系统上，并且要配置为向其发送日志的所有系统。默认情况下，Red Hat Enterprise Linux 7 中会安装 rsyslog。如果需要，请以 root 用户身份 输入以下命令：

```
~]# yum install rsyslog
```

系统日志流量的默认协议和端口为 UDP 和 514，如 /etc/services 文件中列出的。但是，r syslog 默认为在端口 514 上使用 TCP。在 配置文件 /etc/rsyslog.conf 中，TCP 由 @@ 表示。

示例有时会使用其他端口，但 SELinux 被配置为默认允许在以下端口上发送和接收：

```
~]# semanage port -l | grep syslog
syslog_tls_port_t      tcp   6514, 10514
syslog_tls_port_t      udp   6514, 10514
syslogd_port_t         tcp   601, 20514
syslogd_port_t         udp   514, 601, 20514
```

semanage 实用程序作为 policycoreutils-python 软件包的一部分提供。如果需要，按以下方式安装软件包：

```
~]# yum install policycoreutils-python
```

此外，默认情况下，rsyslog d _t 的 SELinux 类型被配置为允许通过 SELinux typersh _port_t 发送和接收到远程 shell(rsh)端口，默认为 514 端口。因此，不需要使用 semanage 来显式允许端口 514 上的 TCP。例如，要在端口 514 中检查将哪个 SELinux 设置为允许，请按如下所示输入命令：

```
~]# semanage port -l | grep 514
output omitted
rsh_port_t          tcp   514
syslogd_port_t         tcp   6514, 601
syslogd_port_t         udp   514, 6514, 601
```

有关 SELinux 的更多信息，请参阅 Red Hat Enterprise Linux 7 SELinux 用户和管理员指南。

在您要用作记录服务器的系统上执行以下步骤。必须以 root 用户身份 执行这些步骤中的所有步骤。

将 SELinux 配置为允许端口上的 rsyslog 流量

如果需要将新端口用于 rsyslog 流量，请在日志记录服务器和客户端上按照以下步骤操作。例如，要在端口 10514 上发送和接收 TCP 流量，请执行以下命令顺序：

1. 使用以下参数运行 semanage port 命令：

```
~]# semanage port -a -t syslogd_port_t -p tcp 10514
```

2. 输入以下命令检查 SELinux 端口：

```
~]# semanage port -l | grep syslog
```

3. 如果在 /etc/rsyslog.conf 中配置了新端口，请重启 rsyslog 以使更改生效：

```
~]# service rsyslog restart
```

4. 验证 rsyslog 正在侦听哪些端口：

```
~]# netstat -tnlp | grep rsyslog
tcp    0   0 0.0.0.0:10514      0.0.0.0:*  LISTEN   2528/rsyslogd
tcp    0   0 :::10514           :::*       LISTEN   2528/rsyslogd
```

有关 semanage port 命令的详情，请查看 semanage -port(8)手册页。

配置 firewalld

配置 firewalld 以允许传入的 rsyslog 流量。例如，要允许端口 10514 上的 TCP 流量，请按如下操作：

```
~]# firewall-cmd --zone=zone --add-port=10514/tcp
success
```

其中 zone 是要使用的接口区。请注意，在系统下次启动后这些更改不会保留。要永久更改防火墙，请重复添加 --permanent 选项的命令。有关在 firewalld 中打开和关闭端口的更多信息，请参阅 Red Hat Enterprise Linux 7 安全指南。

1. 要验证上述设置，请使用以下命令：

```
~]# firewall-cmd --list-all
public (default, active)
```

```
interfaces: eth0
sources:
services: dhcpv6-client ssh
ports: 10514/tcp
masquerade: no
forward-ports:
icmp-blocks:
rich rules:
```

将 rsyslog 配置为接收和排序远程日志消息

1. 在文本编辑器中打开 /etc/rsyslog.conf 文件，如下所示：

   a. 在 modules 部分下方添加这些行，但在 Provides UDP syslog 接收 部分上方：

   ```
   # Define templates before the rules that use them

   # Per-Host Templates for Remote Systems #
   $template TmplAuthpriv,
   "/var/log/remote/auth/%HOSTNAME%/%PROGRAMNAME:::secpath-replace%.log"
   $template TmplMsg,
   "/var/log/remote/msg/%HOSTNAME%/%PROGRAMNAME:::secpath-replace%.log"
   ```

   b. 将默认的 Provides TCP syslog 接收 部分替换为以下内容：

   ```
   # Provides TCP syslog reception
   $ModLoad imtcp
   # Adding this ruleset to process remote messages
   $RuleSet remote1
   authpriv.*  ?TmplAuthpriv
   *.info;mail.none;authpriv.none;cron.none  ?TmplMsg
   $RuleSet RSYSLOG_DefaultRuleset  #End the rule set by switching back to the
   default rule set
   $InputTCPServerBindRuleset remote1 #Define a new input and bind it to the
   "remote1" rule set
   $InputTCPServerRun 10514
   ```

   保存对 /etc/rsyslog.conf 文件的更改。

2. rsyslog 服务必须在日志记录服务器和尝试登录到该服务器的系统上运行。

   a. 使用 systemctl 命令启动 rsyslog 服务。

```
~]# systemctl start rsyslog
```

b. 要确保 rsyslog 服务以后自动启动，以 root 用户身份输入以下命令：

```
~]# systemctl enable rsyslog
```

您的日志服务器现在已配置为从环境中的其他系统接收和存储日志文件。

23.5.1. 在日志记录服务器上使用新模板语法

rsyslog 7 具有多种不同的模板样式。字符串模板与传统格式最相似。使用字符串格式从上例中重现模板，如下所示：

```
template(name="TmplAuthpriv" type="string"
    string="/var/log/remote/auth/%HOSTNAME%/%PROGRAMNAME:::secpath-replace%.log"
   )

template(name="TmplMsg" type="string"
    string="/var/log/remote/msg/%HOSTNAME%/%PROGRAMNAME:::secpath-replace%.log"
   )
```

这些模板也可以采用列表格式编写，如下所示：

```
template(name="TmplAuthpriv" type="list") {
  constant(value="/var/log/remote/auth/")
  property(name="hostname")
  constant(value="/")
  property(name="programname" SecurePath="replace")
  constant(value=".log")
  }

template(name="TmplMsg" type="list") {
  constant(value="/var/log/remote/msg/")
  property(name="hostname")
  constant(value="/")
  property(name="programname" SecurePath="replace")
  constant(value=".log")
  }
```

此模板文本格式可能更易于阅读，以方便那些新的 rsyslog 读取，因此更容易根据要求的变化进行调整。

要完成对新语法的更改，我们需要重现模块 load 命令，添加规则集，然后将规则集绑定到协议、端口

和规则集：

```
module(load="imtcp")

ruleset(name="remote1"){
  authpriv.*  action(type="omfile" DynaFile="TmplAuthpriv")
  *.info;mail.none;authpriv.none;cron.none action(type="omfile" DynaFile="TmplMsg")
}

input(type="imtcp" port="10514" ruleset="remote1")
```

23.6. 使用 RSYSLOG 模块

由于其模块化设计，r syslog 提供了各种 模块，提供额外的功能。请注意，模块可以由第三方编写。大多数模块提供额外的输入（请参阅下面的 输入模块 ）或 输出（请参见下面的输出模块 ）。其他模块提供特定于各个模块的特殊功能。模块可能会在加载模块后提供额外的配置指令。要载入模块，请使用以下语法：

```
module(load="MODULE")
```

其中 MODULE 代表您所需的模块。例如：如果要加载可让 rsyslog 将任何标准文本文件转换为 syslog 信息的文本文件输入模块(imfile)，请在 /etc/rsyslog.conf 配置文件中指定以下行：

```
module(load="imfile")
```

rsyslog 提供了多个模块，它们被分为以下主要类别：

- 输入模块 - 输入模块从各种来源收集消息。输入模块的名称始终以 im 前缀开头，如 imfile 和 imjournal。
- 输出模块 - 输出模块提供相应的功能，可以发出消息到各种目标，如通过网络发送、存储在数据库或加密等目标。输出模块的名称始终以 om 前缀（如 omsnmp、omrelp 等）开头。
- 解析器模块 - 这些模块可用于创建自定义解析规则或解析错误的消息。熟悉 C 编程语言，您可以创建自己的消息解析器。parser 模块的名称始终以 pm 前缀开头，如 pmrfc5424、pmrfc3164 等。
- Message Modification Modules - 消息修改模块更改 syslog 消息的内容。这些模块的名称以 mm 前缀开头。消息修改模块（如 mmanon、MMnormalize 或 mmjsonparse ）用于消息的

匿名化或规范化。

- string Generator Modules - String 生成器模块基于消息内容生成字符串，并与 rsyslog 提供的模板功能高度协作。有关模板的更多信息，请参阅 第 23.2.3 节 “模板”。字符串生成器模块的名称始终以 sm 前缀开头，如 smfile 或 smtradfile。

- Library 模块 - Library 模块提供其他可加载模块的功能。在需要时，rsyslog 会自动加载这些模块，用户无法配置这些模块。

有关所有可用模块及其详细描述的完整列表，请访问

警告

请注意，当 rsyslog 加载任何模块时，它为它们提供了对其部分功能和数据的访问权限。这会带来潜在的安全威胁。要最大程度降低安全风险，仅使用值得信赖的模块。

23.6.1. 导入文本文件

Text File Input Module（简写为 imfile ）使 rsyslog 可以将任何文本文件转换为 syslog 消息流。您可以使用 imfile 从创建自己的文本文件日志的应用导入日志消息。要加载 imfile，请在 /etc/rsyslog.conf 中添加以下内容：

```
module(load="imfile"
  PollingInterval="int")
```

即使导入多个文件，也足以加载一次性的 imfile。PollingInterval 模块参数指定 rsyslog 检查连接文本文件中更改的频率。要更改它，默认间隔为 10 秒，以秒为单位指定的时间间隔替换 int。

要识别要导入的文本文件，在 /etc/rsyslog.conf 中使用以下语法：

```
# File 1
input(type="imfile"
  File="path_to_file"
  Tag="tag:"
```

```
  Severity="severity"
  Facility="facility")

# File 2
input(type="imfile"
  File="path_to_file2")
...
```

指定输入文本文件所需的设置：

- 使用到文本文件的路径替换 path_to_file。
- 将 tag: 替换为此消息的标签名称。

除了所需的指令外，还有一些其他设置可用于文本输入。通过将严重性替换为适当的关键字来设置导入消息的严重性。使用关键字替换 facility，以定义生成消息的子系统。严重性和设备的关键词与基于设备/优先级过滤器中使用的关键字相同，请参阅 第 23.2.1 节 “过滤器”。

例 23.15. 导入文本文件

Apache HTTP 服务器以文本格式创建日志文件。要将 rsyslog 的处理功能应用到 apache 错误消息，请先使用 imfile 模块导入消息。在 /etc/rsyslog.conf 中添加以下内容：

```
module(load="imfile")
input(type="imfile"
  File="/var/log/httpd/error_log"
  Tag="apache-error:")
```

23.6.2. 将消息导出到数据库

在数据库中执行日志数据时，可以更快、方便地处理日志数据，而不是使用文本文件。根据所用的 DBMS 类型，从各种输出模块（如 ommysql、ompgsql、omoracle 或 ommongodb）中进行选择。另外，还可使用依赖于 lib dbi 库的 genericomlib dbi 输出模块。The omlibdbi 模块支持数据库系统 Firebird/Interbase、MS SQL、Sybase、SQLite、Ingres、Oracle、mSQL、MySQL 和 PostgreSQL。

例 23.16. 将 Rsyslog 消息导出到数据库

要将 rsyslog 消息存储在 MySQL 数据库中，请在 /etc/rsyslog.conf 中添加以下内容：

```
module(load="ommysql")

. action(type="ommysql"
server="database-server"
db="database-name"
uid="database-userid"
pwd="database-password"
serverport="1234")
```

首先，加载输出模块，然后指定通信端口。在上例的最后一行中指定了其他信息，如服务器名称和数据库以及身份验证数据。

23.6.3. 启用加密传输

网络传输中的机密性和完整性可以由 TLS 或 GSSAPI 加密协议提供。

传输层安全性(TLS)是一种加密协议，旨在通过网络提供通信安全性。使用 TLS 时，rsyslog 消息 会在发送之前加密，并且发件人和接收器之间也存在互相身份验证。有关配置 TLS，请参阅 "使用 TLS 配置加密的消息传输"一节。

通用安全服务 API(GSSAPI)是一个用于访问安全服务的程序的应用编程接口。若要将其与 rsyslog 连接使用，您必须有一个正常运作的 Kerberos 环境。有关配置 GSSAPI，请参阅 "使用 GSSAPI 配置加密的消息传输"一节。

使用 TLS 配置加密的消息传输

若要通过 TLS 使用加密传输，您需要同时配置服务器和客户端。

1. 创建公钥、私钥和证书文件，请参阅 第 14.1.11 节 "生成新密钥和证书"。

2. 在 服务器 端，在 /etc/rsyslog.conf 配置文件中配置以下内容：

   a. 将 gtls netstream 驱动程序设置为默认驱动程序：

   ```
   global(defaultnetstreamdriver="gtls")
   ```

   b. 提供证书文件的路径：

   -

```
global(defaultnetstreamdrivercafile="path_ca.pem"
defaultnetstreamdrivercertfile="path_cert.pem"
defaultnetstreamdriverkeyfile="path_key.pem")
```

如果您希望使用不太灵活的配置文件，您可以将所有全局指令合并到单个块中。

替换：

- 带有指向公钥的路径的 path_ca.pem
- 带有证书文件路径的 path_cert.pem
- 路径为私钥的 path_key.pem

c. 加载 imtcp 模块并设置驱动程序选项：

```
module(load="imtcp"
StreamDriver.Mode="number"
StreamDriver.AuthMode="anon")
```

d. 启动服务器：

```
input(type="imtcp" port="port")
```

替换：

- 指定驱动程序模式的数字。要启用 TCP 模式，请使用 1
- 端口 号，用于启动侦听器，例如 10514

anon 设置表示客户端未经身份验证。

3. 在客户端，在 /etc/rsyslog.conf 配置文件中配置以下内容：

   a. 加载公钥：

   ```
   global(defaultnetstreamdrivercafile="path_ca.pem")
   ```

   将 path_ca.pem 替换为公钥的路径。

   b. 将 gtls netstream 驱动程序设置为默认驱动程序：

   ```
   global(defaultnetstreamdriver="gtls")
   ```

   c. 配置驱动程序并指定要执行的操作：

   ```
   module(load="imtcp"
     streamdrivermode="number"
     streamdriverauthmode="anon")
   input(type="imtcp"
     address="server.net"
     port="port")
   ```

   将 number、on 和 port 替换为与服务器上相同的值。

   在上面列表中的最后一行，示例操作将来自服务器的消息转发到指定的 TCP 端口。

使用 GSSAPI 配置加密的消息传输

在 rsyslog 中，与 GSSAPI 的交互由 imgssapi 模块提供。打开 GSSAPI 传输模式：

1. 在 /etc/rsyslog.conf 中放入以下配置：

   ```
   $ModLoad imgssapi
   ```

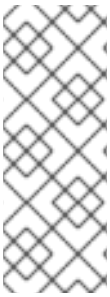
   此指令加载 imgssapi 模块。

2. 按如下所示指定输入：

```
$InputGSSServerServiceName name
$InputGSSServerPermitPlainTCP on
$InputGSSServerMaxSessions number
$InputGSSServerRun port
```

- 使用 GSS 服务器的名称替换 name。
- 替换数字，以设置支持的最大会话数。默认情况下不限制这个数字。
- 使用您要在其上启动 GSS 服务器的所选端口替换 port。

设置 上的 $InputGSSServerPermitPlainTCP 允许服务器在同一端口上同时接收纯 TCP 消息。默认情况下为 off。



注意

当配置文件读取器遇到 /etc/rsyslog.conf 配置文件中的 $InputGSSServerRun 指令时，会立即初始化 The imgsapi 模块。因此，$InputGSSServerRun 之后配置的附加选项将被忽略。要使配置生效，必须在 $InputGSSServerRun 之前放置所有 imgssapi 配置选项。

例 23.17. 使用 GSSAPI

以下配置启用端口 1514 上的 GSS 服务器，该服务器也允许在同一端口上接收普通的 tcp syslog 消息。

```
$ModLoad imgssapi
$InputGSSServerPermitPlainTCP on
$InputGSSServerRun 1514
```

23.6.4. 使用 RELP

可靠的事件记录协议 (RELP)是用于计算机网络中数据登录的网络协议。它旨在提供可靠的事件消息传送，这使其在消息丢失无法接受的环境中很有用。

配置 RELP

要配置 RELP，您需要使用 /etc/rsyslog.conf 文件配置服务器和客户端。

1. 配置客户端：

   a. 加载所需的模块：

   ```
   module(load="imuxsock")
   module(load="omrelp")
   module(load="imtcp")
   ```

   b. 配置 TCP 输入，如下所示：

   ```
   input(type="imtcp" port="port")
   ```

   替换 port，以在所需端口上启动侦听器。

   c. 配置传输设置：

   ```
   action(type="omrelp" target="target_IP" port="target_port")
   ```

   使用标识目标服务器的 IP 地址和端口替换 target_IP 和 target_port。

2. 配置服务器：

   a. 配置载入模块：

   ```
   module(load="imuxsock")
   module(load="imrelp" ruleset="relp")
   ```

   b. 配置与客户端配置类似的 TCP 输入：

   ```
   input(type="imrelp" port="target_port")
   ```

使用与客户端上相同的值替换 target_port。

c.

配置规则并选择要执行的操作。在以下示例中，log_path 指定存储信息的路径：

```
ruleset (name="relp") {
action(type="omfile" file="log_path")
}
```

使用 TLS 配置 RELP

要使用 TLS 配置 RELP，您需要配置身份验证。然后，您需要使用 /etc/rsyslog.conf 文件配置服务器和客户端。

1.

创建公钥、私钥和证书文件。具体步骤请查看 第 14.1.11 节 “生成新密钥和证书”。

2.

配置客户端：

a.

加载所需的模块：

```
module(load="imuxsock")
module(load="omrelp")
module(load="imtcp")
```

b.

配置 TCP 输入，如下所示：

```
input(type="imtcp" port="port")
```

替换 port，以在所需端口上启动侦听器。

c.

配置传输设置：

```
action(type="omrelp" target="target_IP" port="target_port" tls="on"
tls.caCert="path_ca.pem"
tls.myCert="path_cert.pem"
tls.myPrivKey="path_key.pem"
tls.authmode="mode"
tls.permittedpeer=["peer_name"]
)
```

替换：

- target_IP 和 target_port，其 IP 地址和端口标识目标服务器。
- path_ca.pem、path_cert.pem 和 path_key.pem，包含认证文件的路径
- 模式事务的身份验证模式。使用 "name" 或 "fingerprint"
- peer_name，具有允许的对等者的证书指纹。如果您指定此项，tls.permitpeer 将连接限制到所选的对等组。

  TLS="on" 设置启用 TLS 协议。

3. 配置服务器：

   a. 配置载入模块：

   ```
   module(load="imuxsock")
   module(load="imrelp" ruleset="relp")
   ```

   b. 配置与客户端配置类似的 TCP 输入：

   ```
   input(type="imrelp" port="target_port" tls="on"
   tls.caCert="path_ca.pem"
   tls.myCert="path_cert.pem"
   tls.myPrivKey="path_key.pem"
   tls.authmode="name"
   tls.permittedpeer=["peer_name","peer_name1","peer_name2"]
   )
   ```

   将突出显示的值替换为与客户端上相同的值。

   c. 配置规则并选择要执行的操作。在以下示例中，log_path 指定存储信息的路径：

```
ruleset (name="relp") {
action(type="omfile" file="log_path")
}
```

23.7. RSYSLOG 和日志的交互

如前文所述，系统上存在的两个日志记录应用程序 Rsyslog 和 Journal 具有几个独特的功能，使它们适合特定的用例。在很多情况下，组合其功能很有用，例如要创建结构化的信息并将其存储在文件数据库中（请参阅 第 23.8 节 “使用 Rsyslog 的结构化日志记录”）。Rsyslog 方的输入和输出模块以及 日志的通信套接字提供了开展协作所需的通信接口。

默认情况下，rsyslogd 使用 imjournal 模块作为日志文件的默认输入模式。使用此模块时，您不仅导入消息，还会导入 journald 提供的结构化数据。另外，可以从 journald 导入较旧的数据（除非使用 IgnorePreviousMessages 选项禁止）。有关 imjournal 的基本配置，请参阅 第 23.8.1 节 “从日志导入数据”。

另外，也可将 rsyslogd 配置为从 日志 提供的套接字读取，作为基于 syslog 的应用的输出。到套接字的路径为 /run/systemd/journal/syslog。当您要维护纯 rsyslog 消息时，请使用这个选项。与伪装套接字输入相比， 目前提供更多功能，如规则集绑定或过滤。要通过套接字导入 Journal 数据，请在 /etc/rsyslog.conf 中使用以下配置：

```
module(load="imuxsock"
   SysSock.Use="on"
   SysSock.Name="/run/systemd/journal/syslog")
```

您还可以使用 the omjournal 模块输出来自 Rsyslog 到 Journal 的消息。在 /etc/rsyslog.conf 中配置输出，如下所示：

```
module(load="omjournal")
action(type="omjournal")
```

例如，以下配置将 tcp 端口 10514 上的所有接收的信息转发到日志：

```
module(load="imtcp")
module(load="omjournal")

ruleset(name="remote") {
 action(type="omjournal")
}

input(type="imtcp" port="10514" ruleset="remote")
```

23.8. 使用 RSYSLOG 的结构化日志记录

在生成大量日志数据的系统上，以结构化格式维护日志消息非常方便。利用结构化消息，搜索特定信息、生成统计数据以及应对消息结构中的更改和不一致变得更容易。rsyslog 使用 JSON （JavaScript 对象表示法）格式为日志消息提供结构。

比较以下非结构化日志消息：

> Oct 25 10:20:37 localhost anacron[1395]: Jobs will be executed sequentially

凭借结构化功能：

> {"timestamp":"2013-10-25T10:20:37", "host":"localhost", "program":"anacron", "pid":"1395", "msg":"Jobs will be executed sequentially"}

使用键值对搜索结构化数据比使用正则表达式搜索文本文件更快、更准确。通过该结构，您可以在各种应用生成的消息中搜索相同的条目。此外，JSON 文件可以存储在文档数据库中，如 MongoDB，后者提供额外的性能和分析功能。另一方面，结构化消息需要的磁盘空间要多于非结构化空间。

在 rsyslog 中，使用 imjournal 模块从 Journal 中拉取包含元数据的日志消息。使用 mmjsonparse 模块时，您可以解析从 Journal 导入 的数据以及从其他来源导入的数据，并进一步处理它们，例如数据库输出。若要成功解析，mmjson 稀疏要求输入消息的结构由 Lumberjack 项目定义。

Lumberjack 项目旨在以向后兼容的方式将结构化日志记录添加到 rsyslog 中。若要识别结构化消息，Lumberjack 指定 @cee: 字符串，该字符串以前缀于实际 JSON 结构。此外，Lumberjack 定义了应当用于 JSON 字符串中实体的标准字段名称列表。有关 Lumberjack 的详情请参考 “在线文档”一节。

以下是 lumberjack 格式的消息示例：

> @cee: {"pid":17055, "uid":1000, "gid":1000, "appname":"logger", "msg":"Message text."}

要在 Rsyslog 中构建此结构，需要使用一个模板，请参阅 第 23.8.2 节 “过滤结构化消息”。应用程序和服务器可以使用 libumberlog 库生成符合 lumberjack 的格式的消息。有关 libumberlog 的详情请参考 “在线文档”一节。

23.8.1. 从日志导入数据

imjournal 模块是 Rsyslog的输入模块，用于原生读取日志文件（请参阅 第 23.7 节 “Rsyslog 和日志的交互”）。然后，日志消息作为其他 rsyslog 消息以文本格式记录。但是，通过进一步处理，可以将 日

志 提供的元数据转换为结构化消息。

要将数据从 Journal 导入到 Rsyslog，请在 /etc/rsyslog.conf 中使用以下配置：

```
module(load="imjournal"
  PersistStateInterval="number_of_messages"
  StateFile="path"
  ratelimit.interval="seconds"
  ratelimit.burst="burst_number"
  IgnorePreviousMessages="off/on")
```

- 使用 number_of_messages 时，您可以指定必须保存日志数据的频率。每次到达指定数量的消息时都会发生此情况。
- 使用到状态文件的路径替换 path。此文件跟踪最近处理的日志条目。
- 使用秒时，您可以设置速率限值间隔的长度。在此间隔内处理的消息数量不能超过 burst_number 中指定的值。默认设置是每 600 秒 20,000 条消息。rsyslog 会丢弃在指定时间范围内最大突发后的消息。
- 通过 IgnorePreviousMessages，您可以忽略当前位于日志中的消息，仅导入新消息，可在未指定状态文件时使用。默认设置为 off。请注意，如果此设置关闭且没有状态文件，则会处理 Journal 中的所有消息，即使已在以前的 rsyslog 会话中处理。

注意

您可以将 imjournal 与作为传统系统日志输入的 imuxsock 模块同时使用。但是，为了避免消息重复，您必须防止 异常 读取日志的系统套接字。要做到这一点，使用 SysSock.Use 指令：

```
module(load"imjournal")
module(load"imuxsock"
  SysSock.Use="off"
  Socket="/run/systemd/journal/syslog")
```

您可以将 Journal 存储的所有数据和元数据转换为结构化消息。其中一些元数据条目在 例 23.19 “详细 journalctl Output” 中列出，有关 journal 字段的完整列表请查看 systemd.journal-fields(7) 手册页。例如，可以专注于 内核日志字段，这些字段由内核中源自的消息使用。

23.8.2. 过滤结构化消息

要创建 rsyslog的解析模块所需的 lumberjack 格式的消息，请使用以下模板：

```
template(name="CEETemplate" type="string" string="%TIMESTAMP% %HOSTNAME%
%syslogtag% @cee: %$!all-json%\n")
```

此模板将 @cee: 字符串前缀放在 JSON 字符串中，例如，在创建带有 omfile 模块的输出文件时应用。若要访问 JSON 字段名称，请使用 $! 前缀。例如，以下过滤器条件使用特定主机名和 UID 搜索消息：

```
($!hostname == "hostname" && $!UID== "UID")
```

23.8.3. 解析 JSON

mmjson 稀 疏模块用于解析结构化消息。

这些消息可以来自 日志 或其他输入源，并且必须通过 Lumberjack 项目定义的方式格式化。这些消息通过存在 @cee: 字符串来标识。然后，mmjson 稀疏检查 JSON 结构是否有效，然后解析消息。

要使用 mmjsonparse 解析 lumberjack 格式的 JSON 消息，请在 /etc/rsyslog.conf 中使用以下配置：

```
module(load”mmjsonparse”)

. :mmjsonparse:
```

在本例中，mmjsonparse 模块加载到第一行，然后所有消息都转发到它。目前，没有可用于 mmjson 稀疏的配置参数。

23.8.4. 在 MongoDB 中存储消息

rsyslog 支持通过 the ommongodb 输出模块将 JSON 日志存储在 MongoDB 文档数据库中。

要将日志消息转发到 MongoDB，请在 /etc/rsyslog.conf 中使用以下语法（ ommongodb 的配置参数仅以新配置格式可用），请参阅 第 23.3 节 “使用新配置格式”：

```
module(load”ommongodb”)

. action(type="ommongodb" server="DB_server" serverport="port" db="DB_name"
collection="collection_name" uid="UID" pwd="password")
```

- 使用 MongoDB 服务器的名称或地址替换 DB_server。指定从 MongoDB 服务器中选择非标准端口的端口。默认 端口 值为 0，通常不需要更改此参数。

- 使用 DB_name，您可以识别您要将输出定向到 MongoDB 服务器上的数据库。将 collection_name 替换为此数据库中集合的名称。在 MongoDB 中，集合是一组文档，等同于 RDBMS 表。

- 您可以通过替换 UID 和密码来设置登录详细信息。

您可以使用模板来构建最终数据库输出的形式。默认情况下，r syslog 使用基于标准 lumber jack 字段名称的模板。

23.9. 调试 RSYSLOG

要在调试模式下运行 rsyslogd，请使用以下命令：

```
rsyslogd -dn
```

使用此命令，rsyslogd 生成调试信息并将其打印到标准输出中。n 代表 "no fork"。您可以使用环境变量修改调试，例如，您可以将调试输出存储在日志文件中。在启动 rsyslogd 前，在命令行中输入以下内容：

```
export RSYSLOG_DEBUGLOG="path"
export RSYSLOG_DEBUG="Debug"
```

使用记录调试信息的文件的所需位置替换 path。有关可用于 RSYSLOG_DEBUG 变量的完整选项列表，请参阅 rsyslogd(8)手册页 中的相关部分。

检查 /etc/rsyslog.conf 文件中使用的语法是否有效：

```
rsyslogd -N 1
```

其中 1 表示输出消息的详细程度。这是一个向前兼容性选项，因为目前仅提供一个级别。但是，您必须添加此参数才能运行验证。

23.10. 使用日志

Journal 是 systemd 的一个组件，负责查看和管理日志文件。它可以并行使用，也可以代替传统的 syslog 守护进程，如 rsyslogd。该杂志旨在解决与传统记录相关的问题。它与系统的其余部分紧密集成，支持各种日志记录技术和日志文件的访问管理。

日志记录数据由日志的 journald 服务收集、存储和处理。它基于从内核、用户进程、标准输出以及系统服务的标准错误输出或其原生 API 收到的日志信息来创建和维护名为日志的二进制文件。这些日志经过结构化和索引化，可提供相对较快的寻道时间。日志条目可以具有一个唯一标识符。journald 服务为每个日志消息收集大量元数据字段。实际日志文件是安全的，因此无法手动编辑。

23.10.1. 查看日志文件

若要访问日志，可使用 journalctl 工具。对于日志类型的基本视图，以 root 用户身份进行：

```
journalctl
```

此命令的输出是系统上生成的所有日志文件的列表，包括系统组件和用户生成的消息。这个输出的结构与 /var/log/messages/ 中使用的结构类似，但有一些改进：

- 条目的优先级有视觉上的标记。错误优先级和更高的行使用红色颜色突出显示，对于具有 notice 和 warning 优先级的行使用粗体字体
- 时间戳会转换为您系统的本地时区
- 显示所有记录的数据，包括轮转的日志
- 引导的开始带有特殊行的标记

例 23.18. journalctl 输出示例

以下是 journalctl 工具提供的示例输出：如果没有参数调用，列出的条目以时间戳开头，后面提到执行操作的主机名和应用，后面是实际消息。这个示例显示了日志日志中的前三个条目：

```
# journalctl
-- Logs begin at Thu 2013-08-01 15:42:12 CEST, end at Thu 2013-08-01 15:48:48 CEST. --
Aug 01 15:42:12 localhost systemd-journal[54]: Allowing runtime journal files to grow to
49.7M.
Aug 01 15:42:12 localhost kernel: Initializing cgroup subsys cpuset
Aug 01 15:42:12 localhost kernel: Initializing cgroup subsys cpu

[...]
```

在许多情形中，日志中只有最新的条目才相关。减少 journalctl 输出的最简单方法是使用 -n 选项，它只列出指定数量的最新日志条目：

```
journalctl -n Number
```

使用要显示的行数替换 Number。如果未指定数字，journalctl 将显示最新的十个条目。

journalctl 命令允许使用以下语法控制输出格式：

```
journalctl -o form
```

使用指定所需输出格式的关键字替换 form。有多个选项，如 verbose （返回包含所有字段的全结构条目项）、export（ 创建 适合备份和网络传输的二进制流）和 json （将条目格式化为 JSON 数据结构）。有关关键字的完整列表，请查看 journalctl(1) 手册页。

例 23.19. 详细 journalctl Output

要查看所有条目的完整元数据，请输入：

```
# journalctl -o verbose
[...]

Fri 2013-08-02 14:41:22 CEST
[s=e1021ca1b81e4fc688fad6a3ea21d35b;i=55c;b=78c81449c920439da57da7bd5c56a770;m=
27cc
    _BOOT_ID=78c81449c920439da57da7bd5c56a770
    PRIORITY=5
    SYSLOG_FACILITY=3
    _TRANSPORT=syslog
    _MACHINE_ID=69d27b356a94476da859461d3a3bc6fd
    _HOSTNAME=localhost.localdomain
    _PID=562
    _COMM=dbus-daemon
    _EXE=/usr/bin/dbus-daemon
```

```
    _CMDLINE=/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --
systemd-activation
    _SYSTEMD_CGROUP=/system/dbus.service
    _SYSTEMD_UNIT=dbus.service
    SYSLOG_IDENTIFIER=dbus
    SYSLOG_PID=562
    _UID=81
    _GID=81
    _SELINUX_CONTEXT=system_u:system_r:system_dbusd_t:s0-s0:c0.c1023
    MESSAGE=[system] Successfully activated service 'net.reactivated.Fprint'
    _SOURCE_REALTIME_TIMESTAMP=1375447282839181

[...]
```

本例列出了标识单个日志条目的字段。这些元数据可用于信息过滤，如 “高级过滤”一节 所示。有关所有可能字段的完整描述请查看 systemd.journal-fields(7) 手册页。

23.10.2. 访问控制

默认情况下，没有 root 特权的 Journal 用户只能查看由他们生成的日志文件。系统管理员可以将选定的用户添加到 adm 组，授予他们完成日志文件的访问权限。要做到这一点，以 root 用户身份输入：

```
usermod -a -G adm username
```

此处，将 username 替换为要添加到 adm 组的用户名称。然后，此用户收到与 root 用户相同的 journalctl 命令输出。请注意，只有在为 Journal 启用了持久性存储时访问控制才有效。

23.10.3. 使用实时视图

如果没有参数调用，journalctl 将显示条目的完整列表，从收集的最早条目开始。通过实时视图，您可以在新条目出现时实时管理日志消息。要使用 live view 模式启动 journalctl，请输入：

```
journalctl -f
```

此命令返回最新十条日志行的列表。然后 journalctl 实用程序会保持运行并等待新更改立即显示。

23.10.4. 过滤消息

在不使用参数的情况下执行的 journalctl 命令的输出通常很广泛，因此您可以使用各种过滤方法提取信息以满足您的需求。

按优先级过滤

日志消息通常用于跟踪系统上的错误行为。要只查看所选或更高优先级的条目，请使用以下语法：

```
journalctl -p priority
```

此处，将 priority 替换为以下关键字之一（或使用编号）：debug (7)、info(6)、notice (5)、warning (4)、err (3)、crit (2)、警报 (1)和 emerg(0)。

例 23.20. 按优先级过滤

要只查看优先级为 error 或更高的条目，请使用：

```
journalctl -p err
```

按时间过滤

要只从当前引导中查看日志条目，请键入：

```
journalctl -b
```

如果您偶尔重新启动系统，-b 不会显著减少 journalctl 的输出。在这种情况下，基于时间的过滤更有用：

```
journalctl --since=value --until=value
```

使用 --since 和 --until 时，您只能查看在指定时间范围内创建的日志消息。您可以将值以日期和时间形式传递给这些选项，如下例中所示。

例 23.21. 按时间和优先级过滤

可以组合过滤选项，根据特定请求减少结果集。例如，要从特定时间点查看警告或更高优先级的信息，请输入：

```
journalctl -p warning --since="2013-3-16 23:59:59"
```

高级过滤

例 23.19 “详细 journalctl Output” 列出指定日志条目的一组字段，它们都可用于过滤。有关

systemd 可存储的元数据的完整描述，请参阅 systemd.journal-fields(7) 手册页。为每个日志消息收集此元数据，无需用户干预。值通常基于文本，但可以采用二进制和大值；字段可以分配多个值，但并不很常见。

要查看指定字段中出现的唯一值列表，请使用以下语法：

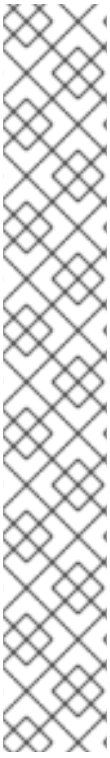
```
journalctl -F fieldname
```

使用您感兴趣的字段的名称替换 fieldname。

要只显示适合特定条件的日志条目，请使用以下语法：

```
journalctl fieldname=value
```

使用字段名称替换 fieldname，并将值替换为该字段中包含的特定值。因此，只会返回与这个条件匹配的行。



注意

由于 systemd 存储的元数据字段数量非常大，很容易忘记相关字段的确切名称。不确定时，键入：

```
journalctl
```

并按 Tab 键两次。这将显示可用字段名称的列表。基于上下文的 Tab 补全基于字段名称，因此您可以从字段名称中键入一组不同的字母，然后按 Tab 键自动填写名称。类似地，您可以从字段中列出唯一值。类型：

```
journalctl fieldname=
```

并按 Tab 键两次。这充当 journalctl -F fieldname 的替代选择。

您可以为一个字段指定多个值：

```
journalctl fieldname=value1 fieldname=value2 ...
```

为同一字段指定两个匹配项会导致逻辑 OR 匹配项的组合。显示与 value1 或 value2 匹配的条目。

另外，您可以指定多个字段值对来进一步减少输出集：

```
journalctl fieldname1=value fieldname2=value ...
```

如果指定了不同字段名称的两个匹配项，则将与逻辑 AND 组合。条目必须与这两个条件匹配。

通过使用 + 符号，您可以为多个字段设置逻辑 OR 匹配项的组合：

```
journalctl fieldname1=value + fieldname2=value ...
```

此命令将返回至少匹配其中一个条件的条目，而不仅仅是与这两个条件匹配的条目。

例 23.22. 高级过滤

要显示 user 下由 avahi-daemon. service 或 crond.service 创建的条目，请使用以下命令：

```
journalctl _UID=70 _SYSTEMD_UNIT=avahi-daemon.service
_SYSTEMD_UNIT=crond.service
```

由于为 _SYSTEMD_UNIT 字段设置了两个值，因此两个结果都将显示，但仅在匹配 _UID=70 条件时才显示。这可以简单地表达为：（UID=70 和（avahi 或 cron）。

您可以在 live-view 模式中应用上述过滤，以跟踪所选日志条目组中的最新更改：

```
journalctl -f fieldname=value ...
```

23.10.5. 启用持久性存储

默认情况下，日志仅将日志文件 存储在 内存中或 /run/log/journal/ 目录中的小型环缓冲器中。这足以显示带有 journalctl 的最近日志历史记录。此目录易失性，日志数据不会永久保存。使用默认配置时，syslog 会读取日志并将其存储在 /var/log/ 目录中。启用持久日志记录后，日志文件存储在 /var/log/journal 中，这意味着它们会在重启后保留。然后日志可以替换某些用户的 rsyslog （但请参见 章节简介）。

启用的持久性存储有以下优点

- 记录了更多数据用于在较长时间内进行故障排除
- 为了立即进行故障排除，重启后可以获得更多数据
- 服务器控制台当前从日志中读取数据，而不是日志文件

持久性存储也有一些缺点：

- 即使使用持久性存储，存储的数据量取决于可用内存量，也无法保证覆盖特定的时间范围
- 日志需要更多磁盘空间

要为 Journal 启用持久存储，请手动创建日志目录，如下例中所示：作为 root 类型：

```
mkdir -p /var/log/journal/
```

然后，重启 journald 以应用更改：

```
systemctl restart systemd-journald
```

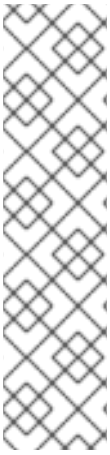
23.11. 在图形环境中管理日志文件

作为上述命令行实用程序的替代选择，红帽企业 Linux 7 提供了用于管理日志消息的可访问 GUI。

23.11.1. 查看日志文件

大多数日志文件都以纯文本格式存储。您可以使用任何文本编辑器（如 Vi 或 Emacs ）来查看它们。某些日志文件可供系统上的所有用户读取；但是，需要 root 特权才能读取大多数日志文件。

要在交互式实时应用中查看系统日志文件，请使用 系统日志。



注意

要使用 系统日志，首先确保以 root 用户身份 在您的系统中安装 gnome-system-log 软件包：

```
~]# yum install gnome-system-log
```

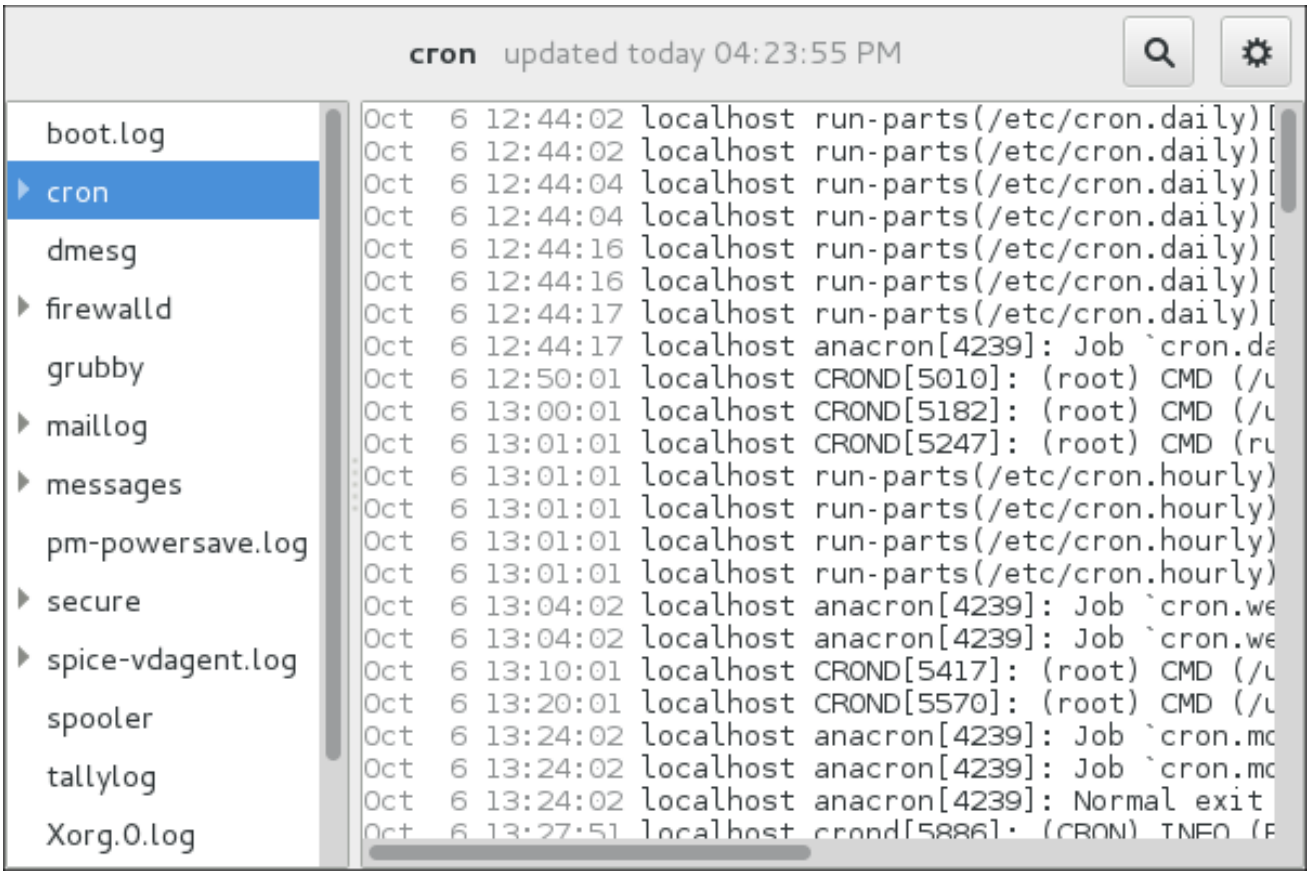
有关使用 Yum 安装软件包的详情请参考 第 9.2.4 节 “安装软件包”。

安装 gnome-system-log 软件包后，点 Applications Tools System Log 打开系统日志，或者在 shell 提示符后输入以下命令：

```
~]$ gnome-system-log
```

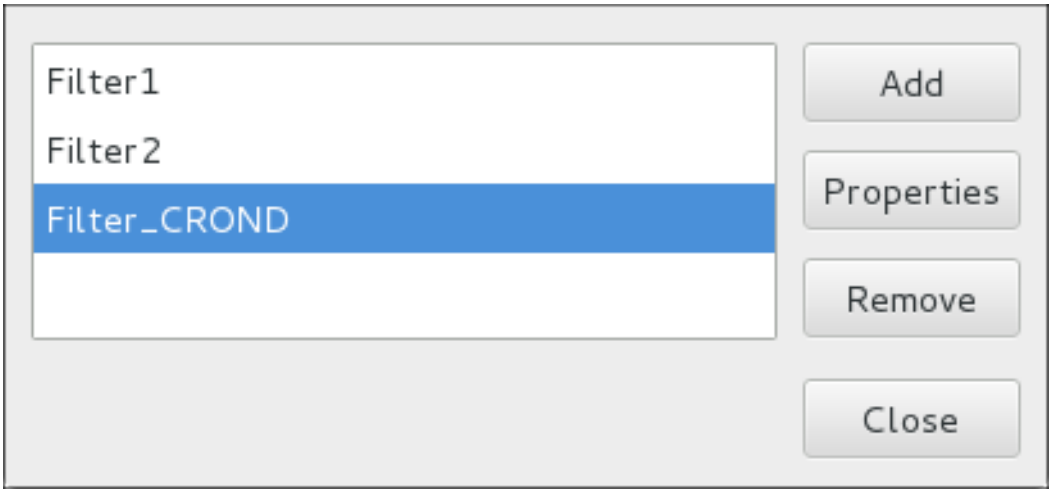
应用仅显示存在的日志文件，因此列表可能与 图 23.2 “系统日志” 中显示的不同。

图 23.2. 系统日志



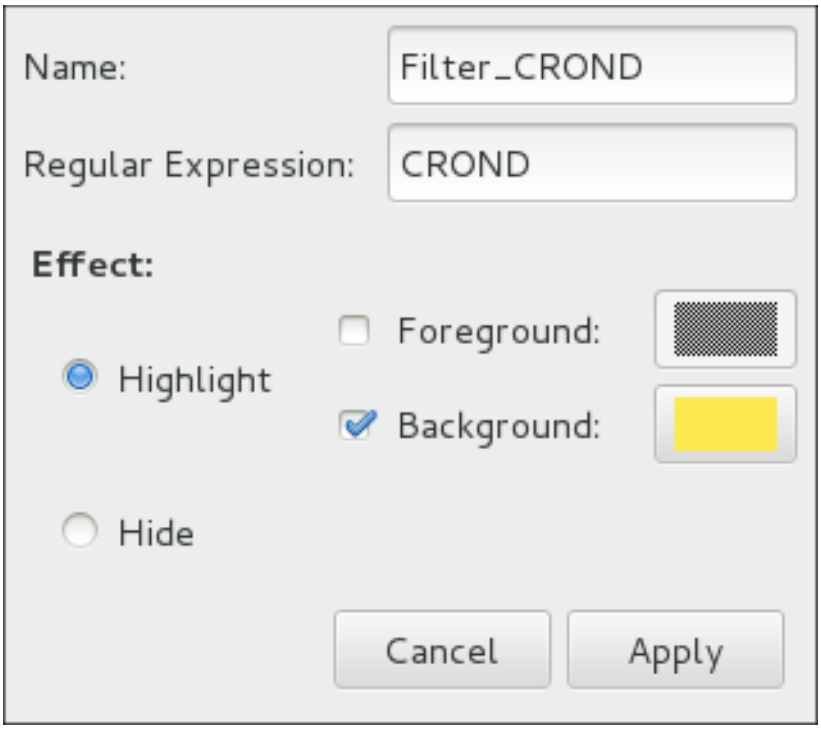
系统日志 应用允许您过滤任何现有的日志文件。点击标有齿轮符号的按钮来查看菜单，选择菜单：[Filters > Manage Filters]来定义或编辑所需的过滤器。

图 23.3. 系统日志 - 过滤器



添加或编辑过滤器可让您定义其参数，如 图 23.4 “系统日志 - 定义过滤器” 所示。

图 23.4. 系统日志 - 定义过滤器



在定义过滤器时，可以编辑以下参数：

- name - 指定过滤器的名称。
-

正则表达式 - 指定将应用到日志文件的正则表达式，并尝试匹配其中的任何可能文本字符串。

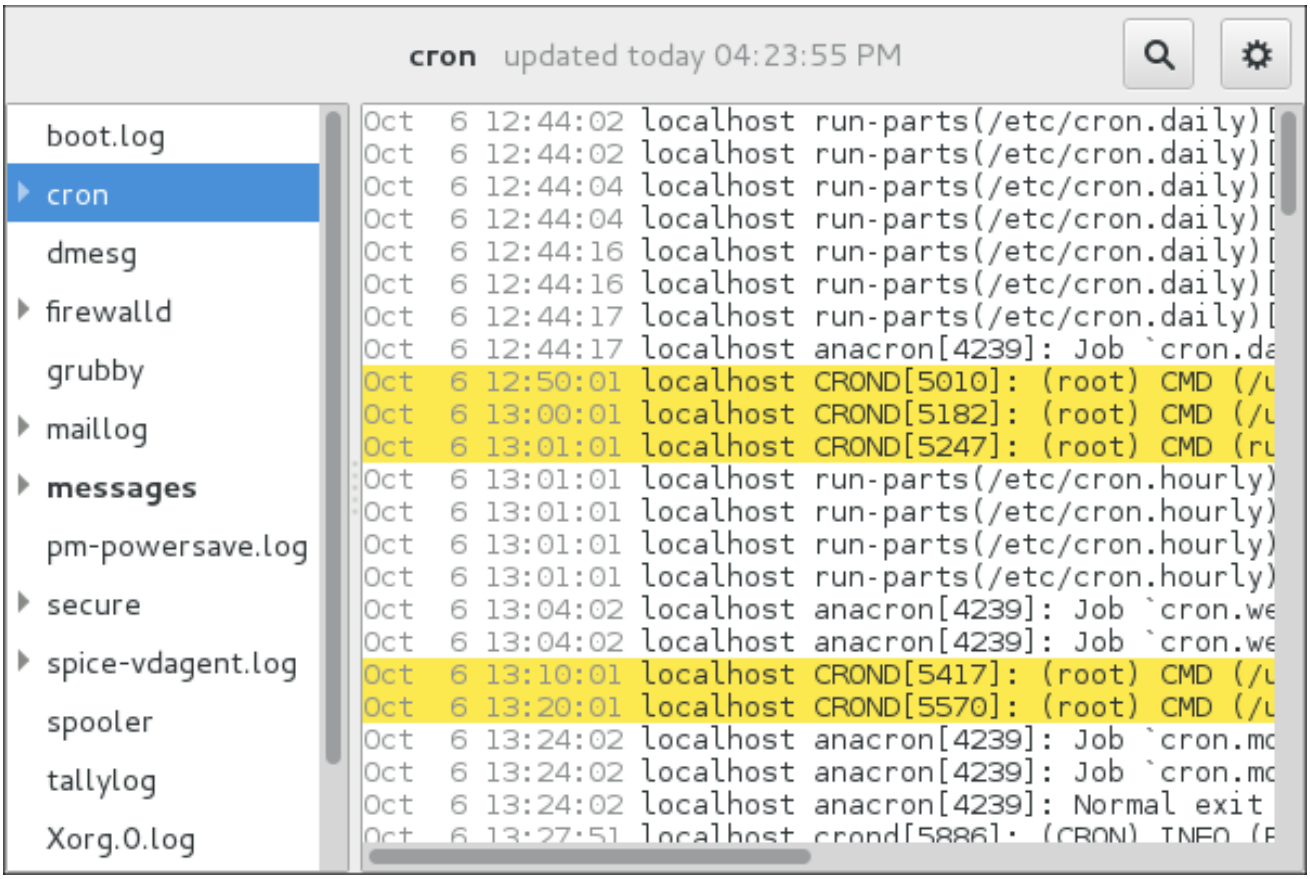
- 效果

  - 突出显示 - 如果选中，将使用选定的颜色突出显示找到的结果。您可以选择是突出显示背景还是文本的前台。

  - 隐藏 - 如果选中，发现的结果会在您正在查看的日志文件中隐藏。

当至少定义了一个过滤器后，可以从 过滤器菜单中选择该过滤器，它将自动搜索过滤器中定义的字符串，并在当前查看的日志文件中突出显示或隐藏每个成功匹配项。

图 23.5. 系统日志 - 启用过滤器

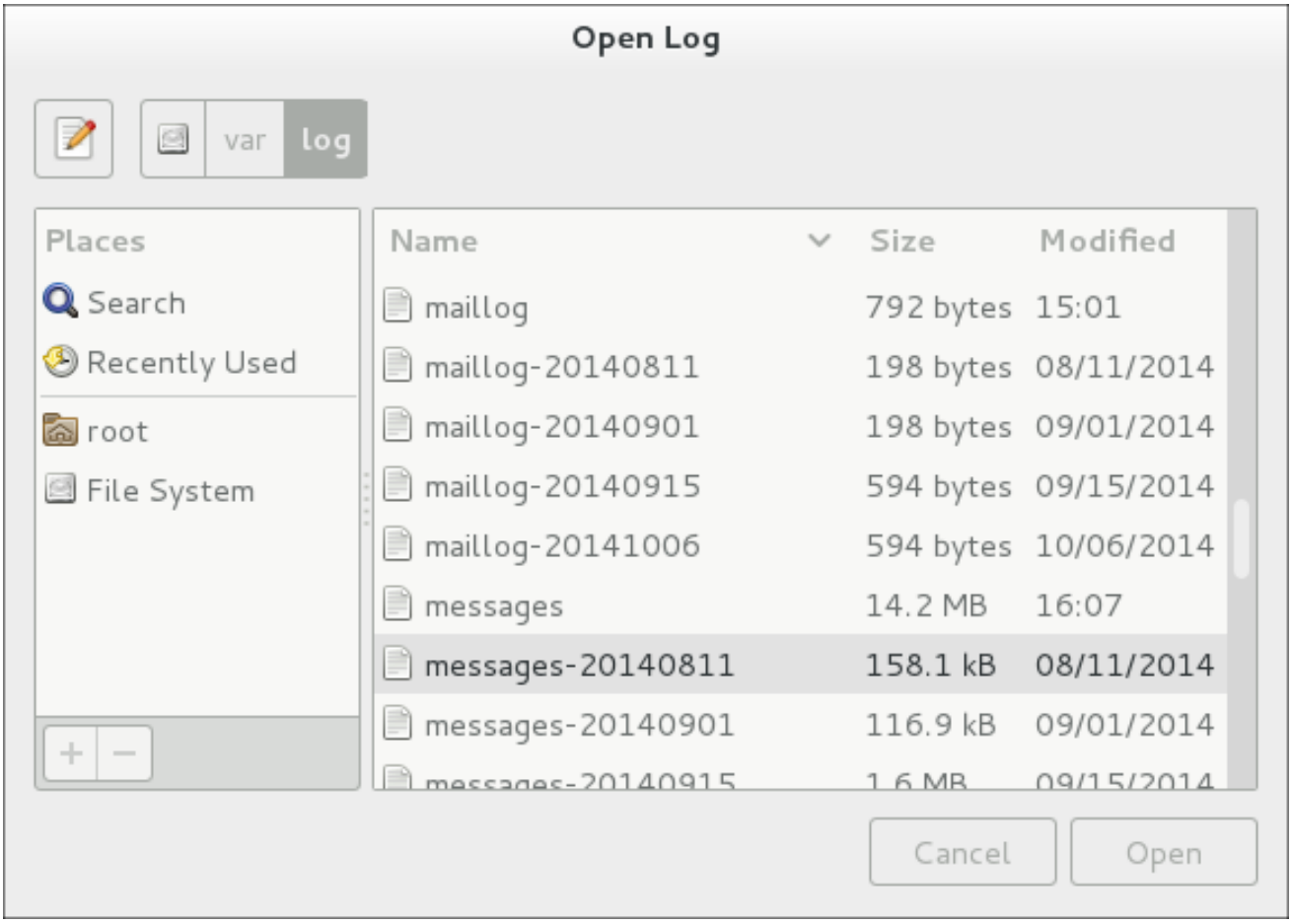


当您选择 **Show match only** 选项时，当前查看的日志文件中仅会显示匹配的字符串。

23.11.2. 添加日志文件

要在列表中添加您要查看的日志文件，请选择 File → → Open 。这将显示 Open Log 窗口，您可以在其中选择您要查看的日志文件的目录和文件名。图 23.6 “系统日志 - 添加日志文件” 演示打开日志窗口.

图 23.6. 系统日志 - 添加日志文件



单击"打开 "按钮以打开 文件。文件会立即添加到查看列表中，您可以在其中选择该文件并查看其内容。

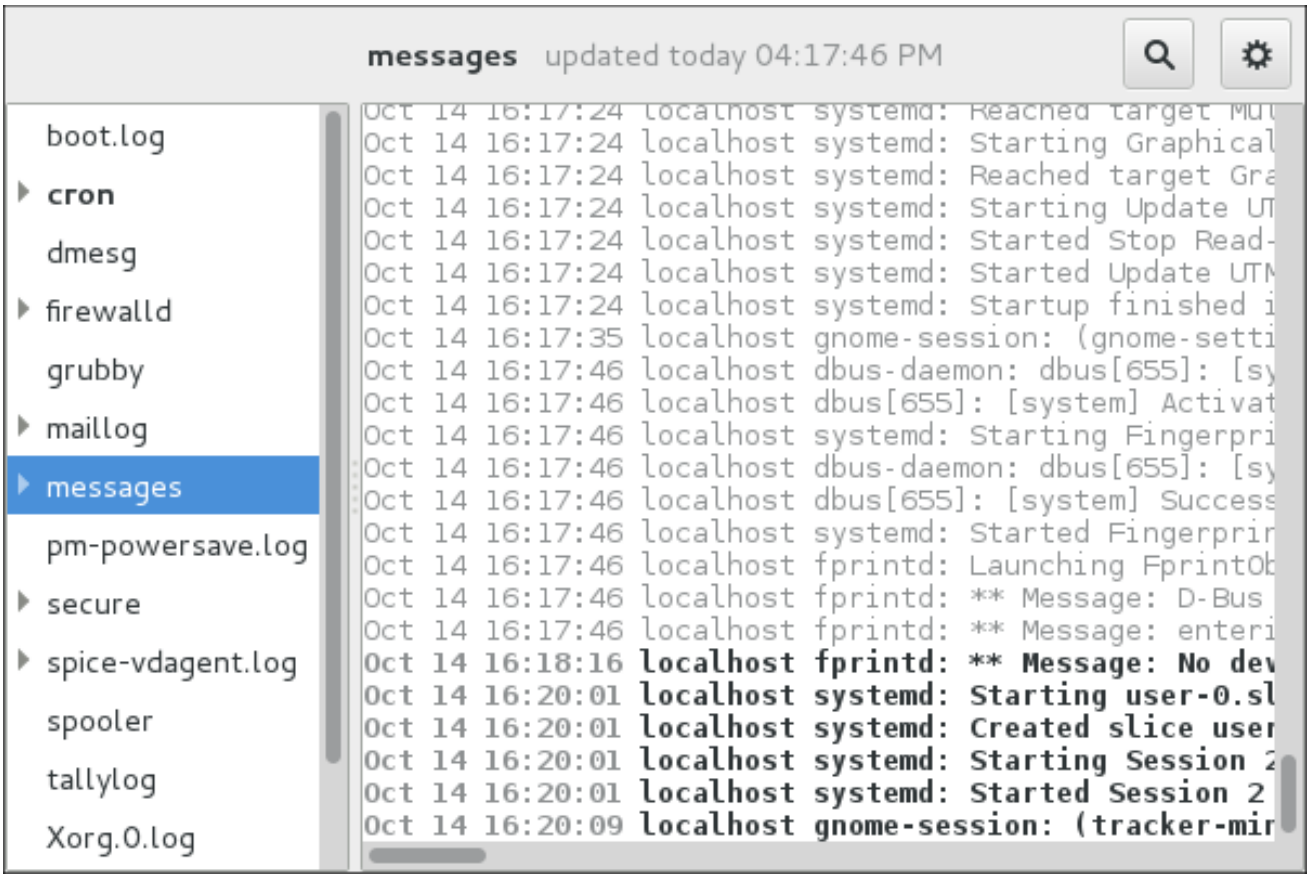
注意

系统日志 还允许您以 .gz 格式打开压缩的日志文件。

23.11.3. 监控日志文件

系统日志 默认监控所有打开的日志.如果向受监控的日志文件中添加了新的行，日志名称将显示在日志列表中粗体显示。如果选择了或显示日志文件，新的行将以粗体显示在日志文件的底部。图 23.7 “系统日志 - 新日志警报” 在 cron 日志文件和 消息 日志文件中演示了一个新警报。单击 消息 日志文件，将显示文件中带有粗体的新行的日志。

图 23.7. 系统日志 - 新日志警报



23.12. 其它资源

有关如何配置 rsyslog 守护进程以及如何查找、查看和监控日志文件的更多信息，请参见以下列出的资源。

安装的文档

- rsyslogd(8)- rsyslogd 守护进程的 man page 包括了其用法。
- rsyslog.conf(5)- 名为 rsyslog.conf 的 man page 包括了可用的配置选项。
- logrotate(8)- logrotate 实用程序的 man page 更详细地说明了如何配置和使用它。
- journalctl(1)- journalctl 守护进程的 man page 记录其用法。
- journald.conf(5)- 本手册页记录了可用的配置选项。

- systemd.journal-fields(7)- 此 man page 列出了特殊的 Journal 字段。

可安装文档

/usr/share/doc/rsyslog版本/html/index.html - 此文件由可选频道 中的 rsyslog-doc 软件包提供，包含有关 rsyslog 的信息。有关红帽附加频道的详情，请查看 第 9.5.7 节 “添加 Optional 和 Supplementary 存储库”。在访问文档前，您必须以 root 用户身份运行以下命令：

```
~]# yum install rsyslog-doc
```

在线文档

rsyslog 主页提供了其他文档、配置示例和视频教程。请确定查看与您使用的版本相关的文档：

- rsyslog 主页上的RainerScript 文档 - RainerScript 中可用的数据类型、表达式和功能的注释摘要.
- rsyslog 主页上的 rsyslog 版本 7 文档 - rsyslog 软件包中的红帽企业 Linux 7 提供了 rsyslog 版本 7 的版本 7。
- rsyslog 主页上的队列描述 - 有关各种类型消息队列及其用法的一般信息。

另请参阅

- 第 6 章 获取特权 文档如何使用 su 和 sudo 命令获得管理权限。
- 第 10 章 使用 systemd 管理服务 提供有关 systemd 的更多信息，以及如何使用 systemctl 命令来管理系统服务。

第 24 章 自动执行系统任务

您可以将 Red Hat Enterprise Linux 配置为自动运行任务，也称为作业：

- 使用 cron 定期使用，请参阅 第 24.1 节 “使用 Cron 调度周期性作业”
- 使用 anacron 在特定天数内异步显示，请参阅 第 24.2 节 “使用 Anacron 计划周期性作业”
- 在使用 时的特定时间，请查看 第 24.3 节 “使用 at 将作业计划在特定时间运行”
- 使用 批处理 系统负载平均值降至指定值后，请参阅 第 24.4 节 “使用批处理调度作业在系统负载 Drop 上运行”
- 下一次引导后，请参阅 第 24.5 节 “使用 systemd 单元文件调度作业在下一次引导时运行”

本章论述了如何执行这些任务。

24.1. 使用 CRON 调度周期性作业

Cron 是一个服务，可让您定期调度运行任务（通常称为作业）。只有在系统按计划的时间运行时，才会执行 cron 作业。有关可以在系统引导时将其执行延迟到调度作业，因此如果系统未运行，则作业不会"丢失"，请参阅 第 24.3 节 “使用 at 将作业计划在特定时间运行”。

用户在 cron 表文件中指定 cron 作业，也称为 crontab 文件。然后，crond 服务读取这些文件，该服务将执行作业。

24.1.1. Cron 作业的先决条件

在调度 cron 任务前：

1. 安装 cronie 软件包：

```
~]# yum install cronie
```

2. 安装时，crond 服务已启用 - 在引导时自动启动。如果禁用该服务，启用该服务：

```
~]# systemctl enable crond.service
```

3. 为当前会话启动 crond 服务：

```
~]# systemctl start crond.service
```

4. （可选）配置 cron.例如，您可以更改：

- 执行作业时要使用的 shell
- PATH 环境变量
- 如果作业发送电子邮件，请邮件.

  有关配置 cron 的详情，请查看 crontab(5)手册页。

24.1.2. 调度 Cron 作业

以 root 用户身份调度作业

root 用户使用 /etc/crontab 中的 cron 表，或者最好在 /etc/cron.d/ 中创建 cron 表文件。使用这个流程以 root 用户身份调度作业：

1. 选择：

- 在每小时的几分钟内执行作业。例如，使用 0,10,20,30,40,50 或 0/10 指定每小时 10 分钟。
- 作业执行一天的几小时。例如，使用 17-20 指定从 17:00 到 20:59 的时间。
- 执行作业的每月天数。例如，使用 15 指定一个月 15 天。

- 在一年内执行该作业。例如，使用 Jun、Jul、Aug 或 6,7,8 指定一年的暑期月。

- 在一周内执行该作业的天数。例如，将 * 用于作业以独立于星期几执行。

  将所选值合并到时间规格中。以上示例值导致这个规格：

  0,10,20,30,40,50 17-20 15 Jun,Jul,Aug *

2. 指定用户。该作业将像该用户运行一样执行。例如，使用 root。

3. 指定要执行的命令。例如：使用 /usr/local/bin/my-script.sh

4. 将以上规格放在一行中：

   ```
   0,10,20,30,40,50 17-20 15 Jun,Jul,Aug * root /usr/local/bin/my-script.sh
   ```

5. 将生成的行添加到 /etc/crontab，或者最好在 /etc/cron.d/ 中创建 cron 表文件，并在其中添加 行。

该作业现在将按计划运行。

有关如何指定作业的完整参考，请参阅 crontab(5)手册页。有关基本信息，请查看 /etc/crontab 文件的开头：

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# .---------------- minute (0 - 59)
# | .------------- hour (0 - 23)
# | | .---------- day of month (1 - 31)
# | | | .------- month (1 - 12) OR jan,feb,mar,apr ...
```

```
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name command to be executed
```

以非 root 用户身份调度作业

非 root 用户可以使用 crontab 实用程序配置 cron 作业。如果由该用户执行，作业将像一样运行。

以特定用户身份创建 cron 任务 ：

1. 在用户的 shell 中运行以下命令：

```
[bob@localhost ~]$ crontab -e
```

这将开始使用由 VISUAL 或 EDITOR 环境变量指定的编辑器编辑用户自己的 crontab 文件。

2. 以与 “以 root 用户身份调度作业”一节 中的相同方式指定作业，但将该字段保留为用户名。例如，而不是添加

```
0,10,20,30,40,50 17-20 15 Jun,Jul,Aug * bob /home/bob/bin/script.sh
```

添加：

```
0,10,20,30,40,50 17-20 15 Jun,Jul,Aug * /home/bob/bin/script.sh
```

3. 保存文件并退出编辑器。

4. （可选）要验证新作业，请运行以下命令列出当前用户 crontab 文件的内容：

```
[bob@localhost ~]$ crontab -l
@daily /home/bob/bin/script.sh
```

计划 Hourly、Daily、Weekly 和 Monthly 作业

调度一个每小时、每天、每周或每月的作业：

1. 将您希望作业执行的操作放入 shell 脚本中。

2. 将 shell 脚本放在以下目录中之一：

   - /etc/cron.hourly/
   - /etc/cron.daily/
   - /etc/cron.weekly/
   - /etc/cron.monthly/

从现在开始，将执行脚本 - crond 服务会自动执行 /etc/cron.hourly、/ etc/cron.daily、/etc/cron.weekly 和 /etc/cron.monthly 目录中的任何脚本。

24.2. 使用 ANACRON 计划周期性作业

anacron （如 cron ）是一项允许您定期调度运行任务（通常称为作业）的服务。但是，anacron 与 cron 有两种不同之处：

- 如果系统未在计划的时间运行，anacron 作业会延迟到系统运行；
- anacron 作业最多可以每天运行一次。

用户在 anacron 表文件中指定 anacron 作业，也称为 anacrontab 文件。然后，crond 服务读取这些文件，该服务将执行作业。

24.2.1. Anacrob 任务的先决条件

在调度 anacron 作业前：

1. 验证您已安装 cronie-anacron 软件包：

```
~]# rpm -q cronie-anacron
```

cronie-anacron 可能已经安装，因为它是 cronie 软件包的子软件包。如果没有安装，请使用这个命令：

```
~]# yum install cronie-anacron
```

2. 安装时，crond 服务已启用 - 在引导时自动启动。如果禁用该服务，启用该服务：

```
~]# systemctl enable crond.service
```

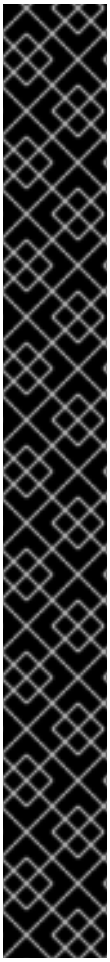
3. 为当前会话启动 crond 服务：

```
~]# systemctl start crond.service
```

4. （可选）配置 anacron.例如，您可以更改：

- 执行作业时要使用的 shell
- PATH 环境变量
- 如果作业发送电子邮件，请邮件.

有关配置 anacron 的相关信息，请参见 anacron tab(5)手册页。



重要

默认情况下，anacron 配置包含一个阻止其运行（如果计算机未插入）的条件。此设置可确保通过运行 anacron 作业，防止垃圾排空。

如果您想允许 anacron 在 battery 电源上运行，请打开 /etc/cron.hourly/0anacron 文件并注释掉以下部分：

```
# Do not run jobs when on battery power
online=1
for psupply in AC ADP0 ; do
 sysfile="/sys/class/power_supply/$psupply/online"

 if [ -f $sysfile ] ; then
  if [ `cat $sysfile 2>/dev/null`x = 1x ]; then
   online=1
   break
  else
   online=0
  fi
 fi
done
```

24.2.2. 调度 Anacron 作业

以 root 用户身份调度 anacron 作业

root 用户使用 /etc/anacrontab 中的 anacron 表。以 root 用户身份使用以下步骤调度作业。

以 root 用户身份调度 anacron 作业

1. 选择：

   - 执行作业的频率。例如，使用 1 指定每天或 3 在 3 天内指定一次。
   - 执行作业的延迟。例如，使用 0 指定不延迟，或者 60 代表 1 小时的延迟。
   - 作业标识符，它将用于日志记录。例如，使用 my.anacron.job 将作业记录为 my.anacron.job 字符串。

- 要执行的命令。例如：使用 /usr/local/bin/my-script.sh

  将所选值合并到作业规格中。以下是一个示例规格：

  ```
  3 60 cron.daily /usr/local/bin/my-script.sh
  ```

2. 将结果行添加到 /etc/anacrontab。

该作业现在将按计划运行。

有关简单作业示例，请查看 /etc/anacrontab 文件。有关如何指定作业的完整参考，请参见 anacrontab(5)手册页。

计划 Hourly、Daily、Weekly 和 Monthly 作业

您可以使用 anacron 计划每日、每周和每月的作业。请参阅 “计划 Hourly、Daily、Weekly 和 Monthly 作业”一节。

24.3. 使用 AT 将作业计划在特定时间运行

若要计划一个称为作业的一次性任务以在指定时间运行一次，请使用 at 实用程序。

用户使用 at 实用程序指定 at 作业。然后，作业由 atd 服务执行。

24.3.1. at Jobs 的先决条件

在调度 at 作业前 ：

1. 安装 at 软件包：

   ```
   ~]# yum install at
   ```

2. 安装时，已启用 atd 服务 - 设为在引导时自动启动。如果禁用该服务，启用该服务：

```
~]# systemctl enable atd.service
```

3. 为当前会话启动 atd 服务：

```
~]# systemctl start atd.service
```

24.3.2. 调度 at 作业

1. 作业始终由某些用户运行。以所需用户身份登录并运行：

```
~]# at time
```

使用时间规格替换 time。

有关指定时间的详情，请查看 at(1)手册页和 /usr/share/doc/at/timespec 文件。

例 24.1. 指定 At 的时间

要在 15:00 执行作业，请运行：

```
~]# at 15:00
```

如果已传递指定时间，则下一日将同时执行该作业。

要在 2017 年 8 月 20 日执行该作业，请运行：

```
~]# at August 20 2017
```

或者

```
~]# at 082017
```

要从现在开始 5 天执行作业，请运行：

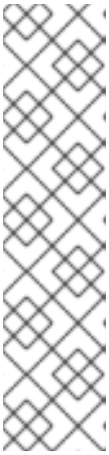
```
~]# now + 5 days
```

```
-
```

2. 在显示 at> 提示符处，输入要执行的命令并按 Enter：

```
~]# at 15:00
at> sh /usr/local/bin/my-script.sh
at>
```

对您要执行的每个命令重复此步骤。



注意

at> 提示符显示它将使用哪个 shell：

```
warning: commands will be executed using /bin/sh
```

at 实用程序使用用户的 SHELL 环境变量或用户的登录 shell 或 /bin/sh （以先到者为准）中设置的 shell。

3. 在空行中按 Ctrl+D 完成指定作业。

注意

如果一组命令或脚本尝试显示标准输出的信息，则输出会通过电子邮件发送给用户。

查看待处理作业

要查看待处理作业列表，请使用 atq 命令：

```
~]# atq
26   Thu Feb 23 15:00:00 2017 a root
28   Thu Feb 24 17:30:00 2017 a root
```

每个作业都以以下格式在单独的行中列出：

```
job_number scheduled_date scheduled_hour job_class user_name
```

job_queue 列指定作业是 at 还是 批处理 作业。a 代表 at, b 表示 批处理。

非 root 用户仅查看其自己的作业。root 用户查看所有用户的作业。

删除调度的作业

删除调度的作业：

1. 使用 atq 命令列出待处理的作业：

```
~]# atq
26    Thu Feb 23 15:00:00 2017 a root
28    Thu Feb 24 17:30:00 2017 a root
```

2. 查找您要删除的作业（按计划的时间和用户）。

3. 运行 atrm 命令，按作业编号指定作业：

```
~]# atrm 26
```

24.3.2.1. 控制对 At 和 Batch 的访问

您可以限制特定用户对 at 和 batch 命令的访问。要做到这一点，根据这些规则将用户名放在 /etc/at.allow 或 /etc/at.deny 中：

- 两种访问控制文件都使用相同的格式：每行一个用户名。
- 任一文件中均不允许使用空格。
- 如果 at.allow 文件存在，则仅该文件中列出的用户可以在 或 批处理 中使用，并且 at.deny 文件将被忽略。
- 如果 at.allow 不存在，则 at.deny 中列出的用户不得在 或 批处理 中使用。

- root 用户不受访问控制文件的影响，并且始终可以执行 at 和 batch 命令。

如果修改了访问控制文件，则无需重新启动 at 守护进程(atd)。每次用户尝试执行 at 或 batch 命令时，都会读取访问控制文件。

24.4. 使用批处理调度作业在系统负载 DROP 上运行

若要计划一次性任务（也称为作业），以在系统负载平均值低于指定值时运行，请使用 batch 实用程序。这对于执行资源需求任务或防止系统处于空闲状态非常有用。

用户使用 batch 实用程序指定批处理作业。然后，作业由 atd 服务执行。

24.4.1. 批处理作业的先决条件

batch 实用程序在 at 软件包中提供，批处理作业由 atd 服务管理。因此，批处理作业的先决条件与 at 作业的前提条件相同。请参阅 第 24.3.1 节 “at Jobs 的先决条件”。

24.4.2. 调度批处理作业

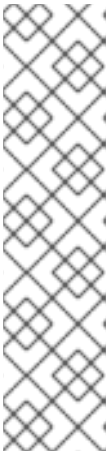
1. 作业始终由某些用户运行。以所需用户身份登录并运行：

```
~]# batch
```

2. 在显示 at> 提示符处，输入要执行的命令并按 Enter：

```
~]# batch
at> sh /usr/local/bin/my-script.sh
```

对您要执行的每个命令重复此步骤。



注意

at> 提示符显示它将使用哪个 shell：

```
warning: commands will be executed using /bin/sh
```

批处理 实用程序使用用户的 SHELL 环境变量或用户的登录 shell 或 /bin/sh（以先到者为准）中设置的 shell。

3. 在空行中按 Ctrl+D 完成指定作业。

注意

如果一组命令或脚本尝试显示标准输出的信息，则输出会通过电子邮件发送给用户。

更改默认系统负载平均值限制

默认情况下，当系统负载平均值低于 0.8 时，批处理 作业会启动。此设置保存在 atq 服务中。更改系统负载限制：

1. 在 /etc/sysconfig/atd 文件中添加以下行：

```
OPTS='-l x'
```

将 x 替换为新的负载平均值。例如：

```
OPTS='-l 0.5'
```

2. 重启 atq 服务：

```
# systemctl restart atq
```

查看待处理作业

要查看待处理作业的列表，请使用 atq 命令。请参阅 “查看待处理作业”一节。

删除调度的作业

若要删除计划的作业，可使用 atrm 命令。请参阅 “删除调度的作业”一节。

控制对批处理的访问

您还可以限制 批处理 实用程序的使用。这是针对 批处理 以及 实用程序 一起完成的。请参阅 第 24.3.2.1 节 “控制对 At 和 Batch 的访问”。

24.5. 使用 SYSTEMD 单元文件调度作业在下一次引导时运行

cron、anacron、at 和 batch 实用程序允许在特定时间或系统工作负载达到特定水平时调度作业。也可以创建将在下一次系统引导期间运行的作业。这可以通过创建一个 systemd 单元文件来实现，它指定要运行的脚本及其依赖项。

将脚本配置为在下次引导时运行：

1. 创建 systemd 单元文件，指定要在引导过程的哪个阶段运行 脚本。这个示例显示了一个单元文件，它带有合理的 Wants= 和 After= 依赖项集合：

```
~]# cat /etc/systemd/system/one-time.service
[Unit]
# The script needs to execute after:
# network interfaces are configured
Wants=network-online.target
After=network-online.target
# all remote filesystems (NFS/_netdev) are mounted
After=remote-fs.target
# name (DNS) and user resolution from remote databases (AD/LDAP) are available
After=nss-user-lookup.target nss-lookup.target
# the system clock has synchronized
After=time-sync.target

[Service]
Type=oneshot
ExecStart=/usr/local/bin/foobar.sh

[Install]
WantedBy=multi-user.target
```

如果您使用这个示例：

- 将 /usr/local/bin/foobar.sh 替换为脚本的名称

- 如果需要，修改 After= 条目集合

  有关指定引导阶段的详情请参考 第 10.6 节 “创建和修改 systemd 单元文件”。

2. 如果您在执行脚本后希望 systemd 服务保持活跃状态，请将 RemainAfterExit=yes 行添加到 [Service] 部分：

```
[Service]
Type=oneshot
RemainAfterExit=yes
ExecStart=/usr/local/bin/foobar.sh
```

3. 重新载入 systemd守护进程：

```
~]# systemctl daemon-reload
```

4. 启用 systemd 服务：

```
~]# systemctl enable one-time.service
```

5. 创建要执行的脚本：

```
~]# cat /usr/local/bin/foobar.sh
#!/bin/bash

touch /root/test_file
```

6. 如果您希望脚本只在下一次引导时运行，而不是在每次引导时都运行，请添加一行来禁用 systemd 单元：

```
#!/bin/bash

touch /root/test_file
systemctl disable one-time.service
```

7. 使脚本可执行：

```
~]# chmod +x /usr/local/bin/foobar.sh
```

24.6. 其它资源

有关在 Red Hat Enterprise Linux 上自动执行系统任务的详情，请参考以下列出的资源。

安装的文档

- cron - crond 守护进程的 man page 记录 crond 的工作原理以及如何更改其行为。
- crontab - crontab 工具的 man page 提供了所支持选项的完整列表。
- crontab(5)- crontab 实用程序 man page 的这一部分记录 crontab 文件的格式。

第 25 章 自动错误报告工具(ABRT)

25.1. ABRT 介绍

自动错误报告工具 通常简写为 ABRT，它旨在帮助用户检测和报告应用程序崩溃。主要目的是简化报告问题和查找解决方案的过程。在这种情况下，解决方案可以是 Bugzilla ticket、知识库文章或建议将软件包更新至包含修复的版本。

ABRT 由 Abrtd 守护进程和用于处理、分析和报告检测到的问题的一系列系统服务和实用程序组成。守护进程大部分时间都在后台运行，并在检测到应用程序崩溃或内核 oops 时进行操作。然后守护进程会收集相关问题数据，例如，如果存在核心文件，则崩溃应用命令行参数和其他取证工具数据。

ABRT 目前支持检测以 C、C++、Java、Python 和 Ruby 编程语言编写的应用程序，以及 X.Org 崩溃、内核oopses 和内核 panic。如需有关支持故障类型和崩溃类型的更多详细信息，以及检测到各种崩溃类型的详情，请参阅 第 25.4 节 “检测软件问题”。

可将识别的问题报告给远程问题跟踪器，报告可以在检测到问题时自动执行。问题数据也可以存储在本地或专用系统上，并由用户手动检查、报告和删除。报告工具可将问题数据发送到 Bugzilla 数据库或红帽技术支持(RHTSupport)网站。这些工具也可以使用 FTP 或 SCP 上传，以电子邮件形式发送或将其写入文件。

处理现有问题数据（例如，创建新问题数据）的 ABRT 组件是单独项目 libreport 的一部分。libreport 库提供了一种用于分析和报告问题的通用机制，它也供 ABRT 以外的应用程序使用。但是，ABRT 和 libreport 操作和配置紧密集成。因此，它们在本文档中被作为一个讨论。

注意

请注意，ABRT 报告仅在生成内核转储时才生成。核心转储仅为某些信号生成。例如，SIGKILL(-9)不生成内核转储，因此 ABRT 无法捕获此失败。有关信号和核心转储生成的更多信息，请参阅 man 7 信号。

25.2. 安装 ABRT 并启动其服务

要使用 ABRT，请确保您的系统上安装 了 abrt-desktop 或 abrt-cli 软件包。abrt-desktop 软件包为 ABRT 提供图形用户界面，而 abrt-cli 软件包则包含用于在命令行中使用 ABRT 的工具。您还可以同时安装.ABRT GUI 和命令行工具的一般工作流在过程上类似，并遵循相同的模式。

警告

请注意，安装 ABRT 软件包会覆盖 /proc/sys/kernel/core_pattern 文件，该文件可包含用于命名 core-dump 文件的模板。此文件的内容将被覆盖：

```
|/usr/libexec/abrt-hook-ccpp %s %c %p %u %g %t e
```

有关如何使用 Yum 软件包管理器安装软件包的常规信息，请参阅 第 9.2.4 节 “安装软件包”。

25.2.1. 安装 ABRT GUI

ABRT 图形用户界面为在桌面环境中工作提供了易于使用的前端。您可以以 root 用户身份运行以下命令安装所需的软件包：

```
~]# yum install abrt-desktop
```

安装时，ABRT 通知小程序配置为在图形桌面会话启动时自动启动。您可以通过在终端中运行以下命令来验证 ABRT 小程序是否正在运行：

```
~]$ ps -el | grep abrt-applet
0 S  500 2036 1824 0 80  0 - 61604 poll_s ?    00:00:00 abrt-applet
```

如果小程序没有运行，您可以通过运行 abrt-applet 程序在当前桌面会话中手动启动它：

```
~]$ abrt-applet &
[1] 2261
```

25.2.2. 为命令行安装 ABRT

命令行界面在无外设计算机、通过网络连接的远程系统或脚本中非常有用。您可以以 root 用户身份运行以下命令安装所需的软件包：

```
~]# yum install abrt-cli
```

25.2.3. 安装补充 ABRT 工具

要接收关于 ABRT 检测到的崩溃的电子邮件通知，您需要安装 libreport-plugin-mailx 软件包。您可以以 root 用户身份执行以下命令来安装它：

```
~]# yum install libreport-plugin-mailx
```

默认情况下，它会在本地计算机上向 root 用户发送通知。电子邮件目标可以在 /etc/libreport/plugins/mailx.conf 文件中配置。

要在登录时在控制台中显示通知，还要安装 abrt-console-notification 软件包。

ABRT 可以检测、分析和报告各种类型的软件故障。默认情况下，安装 ABRT 支持最常见的故障类型，如 C 和 C++ 应用程序的崩溃。对其他类型的故障的支持由独立的软件包提供。例如，要安装对使用 Java 语言编写的应用程序中检测异常的支持，以 root 用户身份运行以下命令：

```
~]# yum install abrt-java-connector
```

有关 ABRT 支持的语言和软件项目列表，请查看 **第 25.4 节 “检测软件问题”**。部分还包含启用检测各种故障类型的所有对应软件包的列表。

25.2.4. 启动 ABRT 服务

abrtd 守护进程需要存在 abrt 用户，以便在 /var/spool/abrt 目录中执行文件系统操作。安装 abrt 软件包时，如果此类用户尚不存在，它会自动创建 UID 和 GID 为 173 的 abrt 用户。否则，可以手动创建 abrt 用户。在这种情况下，可以选择任何 UID 和 GID，因为 brtd 不需要特定的 UID 和 GID。

abrtd 守护进程配置为在引导时启动。您可以使用以下命令来验证其当前状态：

```
~]$ systemctl is-active abrtd.service
active
```

如果 systemctl 返回 inactive 或 unknown，则守护进程没有运行。您可以以 root 用户身份输入以下命令来为当前会话启动它：

```
~]# systemctl start abrtd.service
```

您可以使用同样的命令来启动或检查相关错误检测服务的状态。例如，如果您希望 ABRT 检测到 C 或 C++ 崩溃，请确保 abrt-ccpp 服务正在运行。有关所有可用 ABRT 检测服务及其相应软件包的列表，请

参阅 第 25.4 节 “检测软件问题”。

除 abrt-vmcore 和 abrt-pstoreoops 服务除外，这些服务仅在内核 panic 或内核 oops 发生时启动，所有 ABRT 服务将在安装相应软件包时自动启用并启动。您可以使用 systemctl 工具禁用或启用任何 ABRT 服务，如 第 10 章 使用 systemd 管理服务 所述。

25.2.5. 测试 ABRT 崩溃检测

要测试 ABRT 正常工作，请使用 kill 命令发送 SEGV 信号以终止进程。例如，启动一个 睡眠 进程并使用 kill 命令终止它：

```
~]$ sleep 100 &
[1] 2823
~]$ kill -s SIGSEGV 2823
```

ABRT 执行 kill 命令后很快检测到崩溃，如果图形会话正在运行，GUI 通知小程序将通知用户检测到的问题。在命令行中，您可以通过运行 abrt-cli list 命令或检查 /var/spool/abrt/ 目录中创建的崩溃转储来检查是否检测到了崩溃。有关如何使用检测到的崩溃的更多信息，请参阅 第 25.5 节 “处理检测到的问题”。

25.3. 配置 ABRT

问题生命周期由 ABRT 中的事件 驱动。例如：

- 事件 #1 - 已创建一个问题数据目录。
- 事件 #2 - 问题数据被分析。
- 事件 #3 - 问题报告至 Bugzilla。

每当检测到问题时，ABRT 会将其与所有现有的问题数据进行比较，并确定是否已经记录了同样的问题。如果存在，则对现有问题数据进行更新，并且不再记录最新的（重复数据删除）问题。如果 ABRT 无法识别 该问题，则会创建一个问题数据目录。问题数据目录通常由以下文件组成： 分析器、架构、核心转储、 cmdline、可执行文件、内核、os_release、原因、时间和 uid。

根据使用了分析器方法及其配置设置，可以在分析问题时创建 回溯追踪 等其他文件。每个文件都包含有关系统和问题本身的特定信息。例如，内核 文件记录了崩溃内核的版本。

创建问题数据目录并收集问题后，您可以使用 ABRT GUI 或命令行 abrt-cli 实用程序来解决问题。有关处理记录问题的 ABRT 工具的更多信息，请参阅 第 25.5 节 “处理检测到的问题”。

25.3.1. 配置事件

ABRT 事件使用 插件 来执行实际的报告操作。插件是处理问题数据目录内容的事件调用的紧凑实用程序。通过使用插件，ABRT 能够向不同目的地报告问题，几乎每个报告目的地都需要进行一些配置。例如，Bugzilla 需要用户名、密码和指向 Bugzilla 服务的 URL。

有些配置详情可以具有默认值（如 Bugzilla URL），但其他配置详情不具有明智的默认值（如用户名）。ABRT 在 /etc/libreport/events/ 或 $HOME/ .cache/abrt/events/ 中分别查找配置文件（如 report_Bugzilla.conf ）或 $HOME/.cache/abrt/events/ 目录，分别用于系统范围或用户特定的设置。配置文件包含指令和值对。

这些文件是运行事件和处理问题数据目录所需的最低程度。The gnome-abrt 和 abrt-cli 工具从这些文件中读取配置数据，并将其传递到其运行的事件。

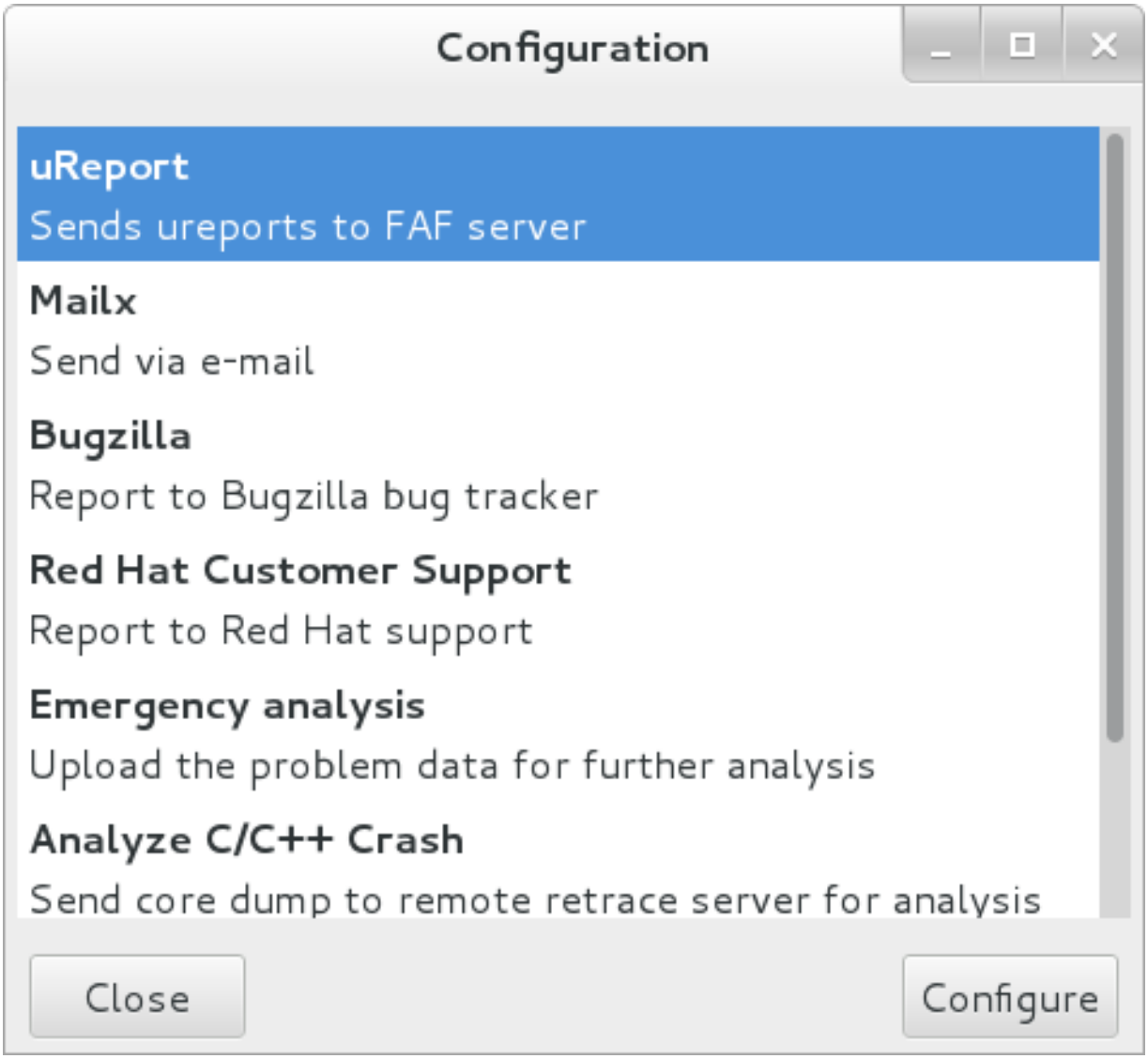
有关事件的其他信息（如描述、名称、可作为环境变量传递的参数类型和其他属性）存储在 /usr/share/libreport/events/ 目录中的 event_name.xml 文件中。这些文件由 gnome-abrt 和 abrt-cli 使用，使用户界面更易理解。除非要修改标准安装，否则请不要编辑这些文件。如果要执行此操作，请将要修改的文件复制到 /etc/libreport/events/ 目录并修改新文件。这些文件可包含以下信息：

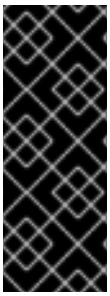
- 用户友好的事件名称和描述（Bugzilla，报告至 Bugzilla 错误跟踪器）
- 事件成功所需的问题数据目录中项目列表。
- 默认和强制选择要发送的项目，
- GUI 是否应该提示进行数据审核，
- 存在哪些配置选项，其类型（字符串、布尔值等）、默认值、提示字符串等；这允许 GUI 构建适当的配置对话框。

例如，report_Logger 事件接受输出文件名作为参数。使用对应的 event_name.xml 文件，ABRT GUI 确定可以为所选事件指定哪些参数，并允许用户为这些参数设置值。这些值由 ABRT GUI 保存，并在这些事件的后续调用中重复使用。请注意，ABRT GUI 使用 GNOME 密钥环 工具保存配置选项，并通过将它们传递给事件来覆盖文本文件中的数据。

要打开图形 配置 窗口，请在 gnome-abrt 应用程序的 正在运行的实例中选择 自动错误报告工具首 → → → 选项。此窗口显示在使用 GUI 时可在报告过程中选择的事件列表。当您选择一个可配置的事件时，您可以单击 Configure 按钮并修改该事件的设置。

图 25.1. 配置 ABRT 事件





重要

/etc/libreport/ 目录层次结构中的所有文件都是全局可读的，旨在用作全局设置。因此，不建议将用户名、密码或任何其他敏感数据存储在其中。每个用户设置（在 GUI 应用中设置并由 $HOME 的所有者唯一可读）安全存储在 GNOME 密钥环 中，或者可以存储在 $HOME/.abrt/ 中的文本文件中，以用于 abrt-cli。

下表显示了由标准安装 ABRT 提供的默认分析、收集和报告事件选择。表列出了 /etc/libreport/events.d/ 目录中的各个事件的名称、标识符、配置文件，以及一个简短描述。请注意，虽然配置文件使用事件标识符，但 ABRT GUI 使用它们的名称来指代各个事件。另请注意，并非所有事件都可使用 GUI 进行设置。有关如何定义自定义事件的详情请参考 第 25.3.2 节 "创建自定义事件"。

表 25.1. 标准 ABRT 事件

名称	标识符和配置文件	描述
uReport	report_uReport	将 ：Report 上传到 FAF 服务器。
mailx	report_Mailx **mailx_event.conf**	通过 Mailx 实用程序将问题报告发送到指定的电子邮件地址。
Bugzilla	report_Bugzilla **bugzilla_event.conf**	向 Bugzilla 错误跟踪程序指定的安装报告问题。
红帽客户支持	report_RHTSupport **rhtsupport_event.conf**	向红帽技术支持系统报告问题。
分析 C 或 C++ Crash	analyze_CCpp **ccpp_event.conf**	将核心转储发送到远程回溯服务器进行分析，或者在远程分析失败时执行本地分析。
报告上传程序	report_Uploader **uploader_event.conf**	使用 **FTP** 或 **SCP** 协议将含有问题数据的 tarball(**.tar.gz**)存档上传到所选目的地。
分析 VM 内核	analyze_VMcore **vmcore_event.conf**	针对内核oops 问题数据运行 **GDB**（GNU 调试器），并生成内核 **的回溯追踪**。
本地 GNU 调试器	analyze_LocalGDB **ccpp_event.conf**	对应用的问题数据运行 **GDB**（GNU 调试器），并为程序生成 **回溯追踪**。
收集 .xsession-errors	analyze_xsession_errors **ccpp_event.conf**	将 **~/.xsession-errors** 文件中的相关行保存到问题报告。
日志记录器	report_Logger **print_event.conf**	创建问题报告并将其保存到指定的本地文件中。
Kerneloops.org	report_Kerneloops **koops_event.conf**	将内核问题发送到 kerneloops.org 上的 oops 跟踪器。

25.3.2. 创建自定义事件

每个事件都通过对应配置文件中的一个规则结构来定义。配置文件通常存储在 /etc/libreport/events.d/ 目录中。这些配置文件由主配置文件 /etc/libreport/report_event.conf 加载。不需要编辑默认配置文件，因为 abrt 将运行 /etc/libreport/events.d/ 中包含的脚本。此文件接受 shell 元字符（如 *、$、 ？）并且相对路径相对路径相对其位置。

每个 规则 以非空格前字符的行开头，并以 空格 字符或 制表符 开头的所有后续行都被视为此规则的一部分。每个规则由两个部分组成， 一个条件部分和程序 部分。条件部分包含以下形式之一的条件：

- VAR=VAL
- VAR!=VAL
- VAL~=REGEX

其中：

- VAR 可以是 EVENT 关键字，也可以是问题数据目录元素的名称（如 可执行文件、软件包、主机名 等）
- VAL 是事件的名称或问题数据元素，以及
- REGEX 是正则表达式。

程序部分由程序名称和 shell 可解读代码组成。如果条件部分中的所有条件都有效，程序部分将在 shell 中运行。以下是事件示例 ：

```
EVENT=post-create date > /tmp/dt
    echo $HOSTNAME uname -r
```

此事件会用当前日期和时间覆盖 /tmp/dt 文件的内容，并在标准输出中打印计算机的主机名及其内核版本。

以下是更加复杂的事件的示例，实际上是预定义事件之一。它将 ~/.xsession-errors 文件中的相关行保存到使用 abrt-ccpp 服务的问题报告中，前提是崩溃的应用程序在崩溃时载入了任何 X11 库：

```
EVENT=analyze_xsession_errors analyzer=CCpp dso_list~=./libX11.
    test -f ~/.xsession-errors || { echo "No ~/.xsession-errors"; exit 1; }
    test -r ~/.xsession-errors || { echo "Can't read ~/.xsession-errors"; exit 1; }
    executable=cat executable &&
    base_executable=${executable##*/} &&
    grep -F -e "$base_executable" ~/.xsession-errors | tail -999 >xsession_errors &&
    echo "Element 'xsession_errors' saved"
```

组可能的事件并不确定。系统管理员可以根据自己需要在 /etc/libreport/events.d/ 目录中添加事件。

目前，标准 ABRT 和 libreport 安装会提供以下事件名称：

post-create

此事件由 abrtd 运行，以处理新创建的问题数据目录。当 创建后事件 运行时，abr td 会检查新问题数据是否与任何已存在的问题目录匹配。如果存在此类问题目录，则会更新，并丢弃新问题数据。请注意，如果创建后 事件中 的任何定义中的脚本都以非零值退出，则 abrtd 将终止进程并丢弃问题数据。

notify,notify-dup

notify 事件在创建后运行 。当事件运行时，用户可以确保问题吸引了他们的关注。notify-dup 类似，但用于同一问题的重复发生。

analyze_name_suffix

其中 name_suffix 是事件名称的可替换部分。此事件用于处理收集的数据。例如，s ights_LocalGDB 事件使用 GNU Debugger(GDB)实用程序来处理应用的核心转储并生成崩溃的回溯追踪。

collect_name_suffix

> {blank}

... 其中 name_suffix 是事件名称的可调整部分。此事件用于收集有关问题的其他信息。

report_name_suffix

> {blank}

... 其中 name_suffix 是事件名称的可调整部分。此事件用于报告问题。

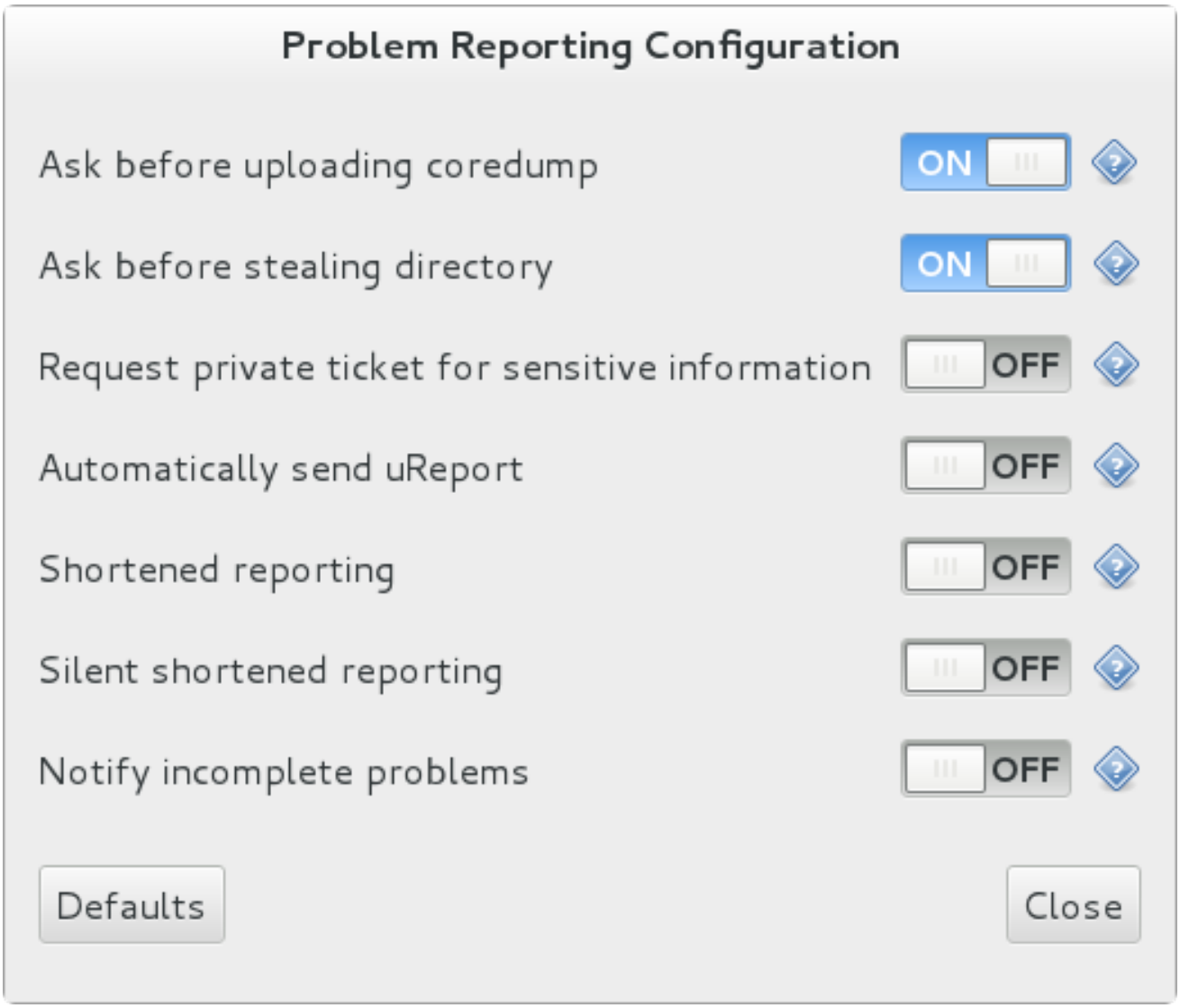
25.3.3. 设置自动报告

ABRT 可以配置为在不用户交互的情况下自动发送任何检测到的问题或自动崩溃的初始匿名报告或仅发送 一个匿名报告。当打开自动报告时，通常在检测到崩溃报告过程开始时发送名为 ¹Report 的名为 ¹Report。这可防止基于相同崩溃的重复支持案例。要启用自动报告功能，以 root 用户身份运行以下命令：

```
~]# abrt-auto-reporting enabled
```

以上命令将 /etc/abrt/abrt.conf 配置文件中的 AutoreportingEnabled 指令设置为 yes。此系统范围设置适用于系统的所有用户。请注意，通过启用此选项，图形桌面环境中也将启用自动报告功能。要在 ABRT GUI 中启用自动报告，请在 问题报告配置 窗口中将 Automatically send uReport 选项切换到 YES。要打开此窗口，请 从 gnome-abrt 应用程序的正在运行的实例中选择 自动错误报告工具 AB → → → RT 配置。要启动应用程序，请转至 Applications → → → Sundry → → Automatic Bug Reporting Tool 。

图 25.2. 配置 ABRT 问题报告



默认情况下，在检测到崩溃后，ABRT 会向红帽的 ABRT 服务器提交有关该问题的基本信息。服务器确定问题是否已知，提供问题的简短描述，并提供所报告案例的 URL （如果已知）或者要求用户报告问题（如果未知）。

注意

□Report(microreport)是一个代表问题的 JSON 对象，如二进制崩溃或内核 oops。报告设计为简略、计算机可读且完全匿名，这就是为什么它们可用于自动报告的原因。
⋮ Reports 可以跟踪错误的发生，但它们通常不会为工程师提供足够的信息来修复这个程序错误。需要一个完整的错误报告才可以打开支持问题单。

要更改自动报告功能的默认行为发送 ›Report，修改 /etc/abrt/abrt.conf 配置文件中的 AutoreportingEvent 指令的值，使其指向不同的 ABRT 事件。有关标准事件的概述请查看 表 25.1 “标准 ABRT 事件”。

25.4. 检测软件问题

ABRT 能够在以各种编程语言编写的应用中检测、分析和处理崩溃。当安装其中一个主要 ABRT 软件包(brt-desktop，abr t-cli)时，系统会自动安装许多包含检测各种崩溃类型的软件包。有关如何安装 ABRT 的说明，请参阅 第 25.2 节 “安装 ABRT 并启动其服务”。下表中列出了受支持的崩溃类型和相应的软件包。

表 25.2. 支持的编程语言和软件项目

Langauge/Project	软件包
c 或 C++	abrt-addon-ccpp
Python	abrt-addon-python
Ruby	rubygem-abrt
Java	abrt-java-connector
X.Org	abrt-addon-xorg
Linux（内核oops）	abrt-addon-kerneloops
Linux（内核 panic）	abrt-addon-vmcore
Linux（永久存储）	abrt-addon-pstoreoops

25.4.1. 检测 C 和 C++ Crashes

abrt-ccpp 服务安装自己的核心转储处理程序，当启动时，会覆盖内核的 core_pattern 变量的默认值，以便 C 和 C++ 崩溃由 abrtd 处理。如果您停止 abrt-ccpp 服务，则之前指定的 core_pattern 值会被恢复。

默认情况下，/proc/sys/kernel/core_pattern 文件包含字符串 core，这意味着内核会在崩溃进程的当前目录中生成 core. 前缀 的文件。abrt-ccpp 服务覆盖 core_pattern 文件，使其包含以下命令：

```
|/usr/libexec/abrt-hook-ccpp %s %c %p %u %g %t e
```

该命令指示内核将内核转储传送到 abrt-hook-ccpp 程序，该程序将其保存在 ABRT的转储位置，并通知新崩溃的缩写守护进程。它还存储 /proc/PID / 目录（其中 PID 是崩溃进程的 ID）中的以下文件以进行调试：map、限制、c groups、状态。有关这些文件的格式和含义的说明，请参见 proc(5)。

25.4.2. 检测 Python 例外

abrt-addon-python 软件包为 Python 应用安装自定义异常处理程序。然后，Py th 解释器会自动导入 /usr/lib64/python2.7/site-packages/ 中安装的 abrt.pth 文件，该文件又导入 abrt_exception_handler.py。这会使用自定义处理程序覆盖 Python 的默认 sys.excepthook，它将未处理的异常通过 Socket API 转发到 abrtd。

要禁用特定于站点的模块的自动导入，并防止在运行 Python 应用程序时使用 ABRT 自定义异常处理器，将 -S 选项传递给 Python 解释器：

```
~]$ python -S file.py
```

在以上命令中，将 file.py 替换为您要执行的 Python 脚本的名称，而不使用特定于站点的模块。

25.4.3. 检测 Ruby Exceptions

rubygem-abrt 软件包使用 at_exit 功能注册自定义处理程序，该功能在程序结束时执行。这样便可检查可能的未处理异常。每次捕获未处理异常时，ABRT 处理程序都会准备一个错误报告，可使用标准 ABRT 工具将其提交到红帽 Bugzilla。

25.4.4. 检测 Java 例外

ABRT Java Connector 是一个 JVM 代理，它报告 无法破碎的 Java 异常。代理注册了多个 JVMTI 事件回调，必须使用 -agentlib 命令行参数加载到 JVM 中。请注意，注册回调的处理会对应用程序的性能造成负面影响。使用以下命令，使 ABRT 捕获来自 Java 类的异常：

```
~]$ java -agentlib:abrt-java-connector=abrt=on $MyClass -platform.jvmtiSupported true
```

在上面的命令中，将 $MyClass 替换为您要测试的 Java 类的名称。通过将 abrt=on 选项传递给连接器，您可以确保由 abrtd 处理异常。如果您希望连接器将异常输出到标准输出，请省略这个选项。

25.4.5. 检测 X.Org Crashes

abrt-xorg 服务从 /var/log/Xorg.0.log 文件中收集并处理有关 X.Org 服务器 崩溃的信息。请注意，如果加载了黑名单 X.org 模块，则不会生成报告。相反，会在 问题数据目录中创建一个不可报告的文件，并给出相应的说明。您可以在 /etc/abrt/plugins/xorg.conf 文件中找到出错模块列表。默认情况下，只有专有图形驱动程序模块才会列入黑名单。

25.4.6. 检测内核 Oopses 和 Panics

通过检查内核日志的输出，ABRT 可以捕获和处理所谓的内核oopses - 非严重偏差于 Linux 内核的正确行为。此功能由 abrt-oops 服务提供。

ABRT 还可以使用 abrt-vmcore 服务检测和处理内核 panics - 需要重新启动的严重、不可恢复的错误。仅当 vmcore 文件（内核转储）显示在 /var/crash/ 目录中时，服务才会启动。找到 core-dump 文件时，abr t-vmcore 在 /var/spool/abrt/ 目录中创建一个新的问题数据 目录，并将 core-dump 文件复制到新创建的 issue-data 目录中。搜索 /var/crash/ 目录后，服务将停止。

要使 ABRT 能够检测到内核 panic，必须在系统中启用 kdump 服务。为 kdump 内核保留的内存量必须正确设置。您可以使用 system -config-kdump 图形工具进行设置，或在 GRUB 2 菜单的内核选项列表中指定 crashkernel 参数。有关如何启用和配置 kdump 的详情，请参考 Red Hat Enterprise Linux 7 内核崩溃转储指南。有关对 GRUB 2 菜单进行更改的详情请参考 第 26 章 使用 GRUB 2。

通过使用 abrt-pstoreoops 服务，ABRT 能够收集和处理有关内核 panic 的信息，这些信息存储在支持 pstore 的系统上，存储在自动挂载的 /sys/fs/pstore/ 目录中。依赖于平台的 pstore 接口（持久存储）提供了一种在系统重启后存储数据的机制，从而允许保留内核 panic 信息。当内核崩溃转储文件出现在 /sys/fs/pstore/ 目录中时，服务会自动启动。

25.5. 处理检测到的问题

可以使用命令行工具、abr t- cli 或图形工具 gnome-abrt 查看、报告和删除 abrt d 保存的问题数据。

注意

请注意，ABRT 通过将新问题与本地保存的所有问题进行比较来识别重复的问题。对于重复崩溃，ABRT 要求您仅对它执行一次操作。但是，如果您删除了该问题的崩溃转储，下一次发生这个问题时，ABRT 会将其视为新崩溃： ABRT 将提醒您，提示您填写描述并报告它。为避免 ABRT 向您发送重复问题通知，请不要删除其问题数据。

25.5.1. 使用命令行工具

在命令行中，如果用户安装了 abrt-console-notification 软件包，则用户会在登录时收到新崩溃通知。控制台通知类似于以下内容：

```
ABRT has detected 1 problem(s). For more info run: abrt-cli list --since 1398783164
```

要查看检测到的问题，请输入 abrt-cli list 命令：

```
~]$ abrt-cli list
```

```
id 6734c6f1a1ed169500a7bfc8bd62aabaf039f9aa
Directory:   /var/tmp/abrt/ccpp-2014-04-21-09:47:51-3430
count:     1
executable:   /usr/bin/sleep
package:    coreutils-8.22-11.el7
time:      Mon 21 Apr 2014 09:47:51 AM EDT
uid:      1000
Run 'abrt-cli report /var/tmp/abrt/ccpp-2014-04-21-09:47:51-3430' for creating a case in Red Hat Customer Portal
```

abrt-cli list 命令输出中列出的每个崩溃都有一个唯一标识符和一个目录，可用于使用 abrt-cli 进行进一步操作。

要查看只有一个问题的信息，请使用 abrt-cli info 命令：

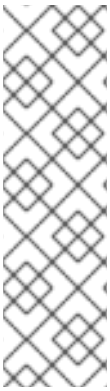
```
abrt-cli info -d directory_or_id
```

若要增加使用 list 和 info 子命令时显示的信息量，可将 -d (--tailed) 选项传递给它们，该选项可显示所列出问题的所有存储信息，包括相关的后端 文件（如果已生成）。

要分析和报告某个问题，请使用 abrt-cli report 命令：

```
abrt-cli report directory_or_id
```

在调用上述命令时，系统会要求您提供与红帽客户支持相关的凭证，以便提交问题单。接下来，abrt-cli 会打开包含报告内容的文本编辑器。您可以查看报告的内容，并填写有关如何重现崩溃和其他注释的说明。您还应检查回溯追踪，因为后端可能会发送到公共服务器并由任何人查看，具体取决于问题报告器事件设置。



注意

您可以选择使用哪个文本编辑器来检查报告。abrt-cli 使用 ABRT_EDITOR 环境变量中定义的编辑器。如果未定义 变量，它将检查 VISUAL 和 EDITOR 变量。如果未设置这些变量，则使用 vi 编辑器。您可以在您的 .bashrc 配置文件中设置首选编辑器。例如，如果您首选 GNU Emacs，请在文件中添加以下行：

```
export VISUAL=emacs
```

使用完报告后，保存您的更改并关闭编辑器。如果您已将问题报告至红帽客户支持数据库，则会在数据库中提交问题。从现在开始，您将获得有关问题的信息 - 报告过程中提供的电子邮件将为您解决问题。

您还可以使用在创建问题问题单或红帽支持收到的电子邮件时为您提供的 URL 来监控问题案例。

如果您确定您不想报告某个特定问题，您可以将其删除。要删除某个问题，以便 ABRT 不保留有关它的信息，使用以下命令：

```
abrt-cli rm directory_or_id
```

要显示特定 abrt-cli 命令的帮助信息，请使用 --help 选项：

```
abrt-cli command --help
```

25.5.2. 使用 GUI

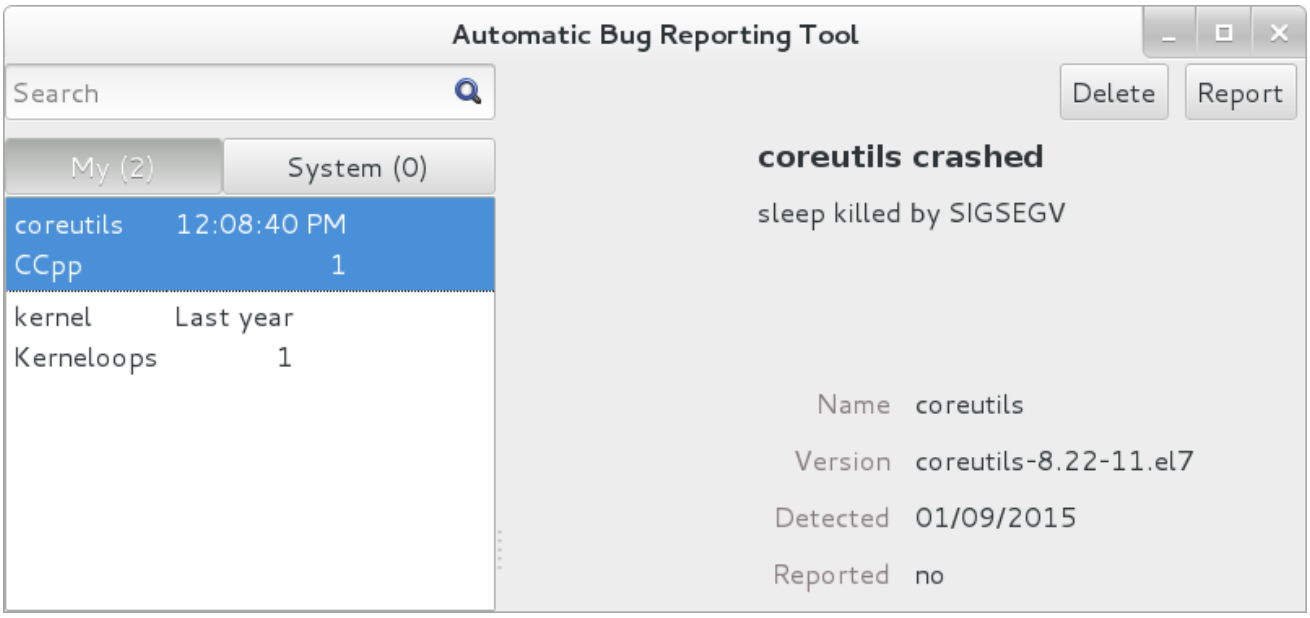
ABRT 守护进程会在创建问题报告时广播 D-Bus 消息。如果 ABRT 小程序在图形桌面环境中运行，它将捕获此消息并在桌面上显示通知对话框。您可以使用此对话框打开 ABRT GUI，方法是单击 报告 按钮。您还可以选择 Applications → → → Sundry → → Automatic Bug Reporting Tool 菜单项打开 ABRT GUI。

或者，您可以从命令行运行 ABRT GUI，如下所示：

```
~]$ gnome-abrt &
```

ABRT GUI 窗口显示检测到的问题列表。每个问题条目都由失败应用程序的名称、应用程序崩溃的原因以及上次发生问题的日期组成。

图 25.3. ABRT GUI



要访问详细的问题描述，请双击问题报告行，或者在选择相应问题行时单击报告按钮。然后，您可以按照说明继续进行描述问题、确定问题分析方式以及报告位置的过程。要丢弃问题，请单击 Delete 按钮。

25.6. 其它资源

有关 ABRT 和相关主题的更多信息，请参见以下列出的资源。

安装的文档

- abrtd(8)- abrtd 守护进程的 man page 提供了有关可与守护进程搭配使用的选项的信息。
- abrt_event.conf(5)- abrt_event.conf 配置文件的 man page 描述其指令和规则的格式，并提供 XML 文件中事件元数据配置的参考信息。

在线文档

- 红帽企业 Linux 7 联网指南 - 红帽企业 Linux 7 的网络指南记录了有关此系统上网络接口和网络服务的配置和管理的相关信息。
- Red Hat Enterprise Linux 7 内核崩溃转储指南 - 红帽企业 Linux 7 的内核 崩溃转储指南 记录了如何配置、测试和使用 kdump 崩溃恢复服务，并提供如何使用 崩溃 调试程序分析所生成的内核转储的简要概述。

另请参阅

- 第 23 章 查看和管理日志文件 描述 rsyslog 守护进程的配置和 systemd 日志，并说明如何查找、查看和监控系统日志。
- 第 9 章 yum 描述如何使用 Yum 软件包管理器在命令行中搜索、安装、更新和卸载软件包。
- 第 10 章 使用 systemd 管理服务 介绍 systemd 和文档，以及如何使用 systemctl 命令来管理系统服务、配置 systemd 目标和执行电源管理命令。

部分 VII. 使用 BOOTLOADER 自定义内核

这部分论述了如何使用 GRUB 2 引导装载程序来协助管理员进行内核自定义。

第 26 章 使用 GRUB 2

Red Hat Enterprise Linux 7 带有 GNU GRand Unified Bootloader(GRUB 2)版本 2，允许用户选择在系统启动时载入的操作系统或内核。GRUB 2 还允许用户向内核传递参数。

26.1. GRUB 2 简介

GRUB 2 从基于 BIOS 的传统机器上的 /boot/grub2/grub.cfg 文件中读取其配置，以及 UEFI 计算机上的 /boot/efi/EFI/redhat/grub.cfg 文件。此文件包含菜单信息。

GRUB 2 配置文件 grub.cfg 在安装 期间或通过调用 /usr/sbin/grub2-mkconfig 实用程序生成，每次安装新内核时，RUB by 会自动更新该文件。使用 grub2-mkconfig 手动 生成时，文件会根据 /etc/grub.d/ 中的模板文件生成，以及 /etc/default/grub 文件中的自定义设置。当使用 grub2-mkconfig 重新生成文件时，grub.cfg 的编辑都将丢失，因此还必须小心反映 /etc/default/grub 中的任何手动更改。

grub.cfg 上的常规操作（例如删除和添加新内核）应使用 grubby 工具执行；对于脚本，应使用 new-kernel-pkg 工具执行。如果您使用 grubby 修改默认内核，则安装新内核时将继承更改。有关 grubby 的更多信息，请参阅 第 26.4 节 “使用 grubby 工具对 GRUB 2 菜单进行持久更改”。

grub2-mkconfig 工具使用 /etc/default/grub 文件，该工具供 anaconda 在安装过程中创建 grub.cfg 使用；出现故障时，例如需要重新创建启动加载器配置时，可以使用该文件。通常，不建议手动运行 grub 2-mkconfig 来替换 grub.cfg 文件，除非作为最后的手段。请注意，对 /etc/default/grub 的任何手动更改都需要重建 grub.cfg 文件。

grub.cfg 中的菜单条目

在各种代码片段和指令中，grub .cfg 配置文件包含一个或多个 菜单输入 块，各自代表单个 GRUB 2 引导菜单条目。这些块始终以 menuentry 关键字开头，后跟标题、选项列表和打开大括号，最后是右花括号。打开和关闭括号之间的任何内容都应缩进。例如，以下是带有 Linux 内核 3.8.0-0.40.el7.x86_64 的 Red Hat Enterprise Linux 7 的菜单输入 块示例：

```
menuentry 'Red Hat Enterprise Linux Server' --class red --class gnu-linux --class gnu --class
os $menuentry_id_option 'gnulinux-simple-c60731dc-9046-4000-9182-64bdcce08616' {
    load_video
    set gfxpayload=keep
    insmod gzio
    insmod part_msdos
    insmod xfs
    set root='hd0,msdos1'
    if [ x$feature_platform_search_hint = xy ]; then
      search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-efi=hd0,msdos1 --
hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1' 19d9e294-65f8-4e37-8e73-d41d6daa6e58
    else
```

```
    search --no-floppy --fs-uuid --set=root 19d9e294-65f8-4e37-8e73-d41d6daa6e58
    fi
    echo  'Loading Linux 3.8.0-0.40.el7.x86_64 ...'
    linux16  /vmlinuz-3.8.0-0.40.el7.x86_64 root=/dev/mapper/rhel-root ro rd.md=0 rd.dm=0
rd.lvm.lv=rhel/swap crashkernel=auto rd.luks=0 vconsole.keymap=us rd.lvm.lv=rhel/root rhgb
quiet
    echo  'Loading initial ramdisk ...'
    initrd /initramfs-3.8.0-0.40.el7.x86_64.img
}
```

代表安装的 Linux 内核的每个 菜单输入 块包含 64 位 IBM POWER 系列 中的 linux、x86_64 基于 BIOS 的系统中 linux16，以及基于 UEFI 的系统上的 linuxefi。然后，init rd 指令加上到内核的路径和 initramfs 镜像。如果创建了单独的 /boot 分区，到内核和 initramfs 镜像 的路径相对于 /boot。在上例中，init rd /initramfs-3.8.0-0.40.el7.x86_64.img 行表示 initramfs 映像实际位于 /boot/initramfs-3.8.0-0.40.el7.x86_64.img （在 根 文件系统挂载时）。

linux16 /vmlinuz-kernel_version 行上提供的内核版本号必须与每个 菜单条目块 的 initrd /initramfs-kernel_version.img 行中给出的 initramfs 映像的版本号匹配。有关如何验证初始 RAM 磁盘镜像的更多信息，请参阅 Red Hat Enterprise 7 内核管理指南。

注意

在 菜单条目 块中，init rd 指令必须指向 initramfs 文件与同一内核版本对应的 initramfs 文件的位置（相对于 /boot/ 目录）。此指令称为 initrd，因为之前的工具创建了初始 RAM 磁盘镜像 mkinitrd，它创建了所谓的 initrd 文件。grub.cfg 指令保留 initrd，以保持与其他工具的兼容性。使用 dracut 实用程序创建初始 RAM 磁盘映像的系统文件算法是 initramfs-kernel_version.img。

有关使用 Dracut 的详情，请参考 Red Hat Enterprise 7 内核管理指南。

26.2. 配置 GRUB 2

可以在启动时临时更改 GRUB 2 菜单，在系统运行时对于单个系统永久保留，或者作为编写新的 GRUB 2 配置文件的一部分。

- 要对 GRUB 2 菜单进行非持久更改，请参阅 第 26.3 节 “对 GRUB 2 菜单进行临时更改”。
- 要对正在运行的系统进行持久更改，请参阅 第 26.4 节 “使用 grubby 工具对 GRUB 2 菜单进行持久更改”。

- 有关创建和自定义 GRUB 2 配置文件的详情请参考 第 26.5 节 “自定义 GRUB 2 配置文件”。

26.3. 对 GRUB 2 菜单进行临时更改

对内核菜单进行临时更改

要只在单个引导过程中更改内核参数，请按如下操作：

1. 启动系统并在 GRUB 2 引导屏幕上将光标移至要编辑的菜单条目，然后按 e 键进行编辑。

2. 向下移动光标以查找内核命令行。内核命令行以 64 位 IBM Power 系列上的 linux 开头、在基于 x86-64 BIOS 的系统上 linux 16，或者 UEFI 系统上的 linuxefi。

3. 将光标移至行末。

   Ctrl+a Ctrl+e 键，分别跳到行首和行尾。在某些系统上，Home 和 End 也起作用。

4. 根据需要编辑内核参数。例如，要以紧急模式运行系统，请在 linux16 行末尾添加 emergency 参数：

   ```
   linux16  /vmlinuz-3.10.0-0.rc4.59.el7.x86_64 root=/dev/mapper/rhel-root ro rd.md=0
   rd.dm=0 rd.lvm.lv=rhel/swap crashkernel=auto rd.luks=0 vconsole.keymap=us
   rd.lvm.lv=rhel/root rhgb quiet emergency
   ```

   可以删除 rhgb 和 quiet 参数，以启用系统消息。

   这些设置不是永久性的，仅适用于一次引导。若要永久更改系统上的菜单条目，请使用 grubby 工具。有关使用 grubby 的更多信息，请参阅 “从 GRUB 2 菜单条目中添加和删除参数”一节。

26.4. 使用 GRUBBY 工具对 GRUB 2 菜单进行持久更改

grubby 工具可用于从 grub.cfg 文件读取信息，并对该文件进行永久性更改。例如，它启用更改 GRUB 2 菜单条目以指定要在系统启动时传递给内核并更改默认内核的参数。

在 Red Hat Enterprise Linux 7 中，如果手动调用 grubby 而无需指定 GRUB 2 配置文件，则默认搜索 /etc/grub2.cfg，这是指向 grub.cfg 文件的符号链接，该文件的位置取决于架构。如果找不到该文件，它将搜索依赖于默认架构的架构。

列出默认内核

要找出默认内核的文件名，请按如下所示输入命令：

```
~]# grubby --default-kernel
/boot/vmlinuz-3.10.0-229.4.2.el7.x86_64
```

要找出默认内核的索引号，请按如下所示输入命令：

```
~]# grubby --default-index
0
```

更改默认引导条目

要在指定为默认内核的内核中进行持久更改，请使用 grubby 命令，如下所示：

```
~]# grubby --set-default /boot/vmlinuz-3.10.0-229.4.2.el7.x86_64
```

查看内核的 GRUB 2 菜单条目

要列出所有内核菜单条目，请按如下所示输入命令：

```
~]$ grubby --info=ALL
```

在 UEFI 系统中，所有 grubby 命令必须以 root 身份输入。

要查看特定内核的 GRUB 2 菜单条目，请输入以下命令：

```
~]$ grubby --info /boot/vmlinuz-3.10.0-229.4.2.el7.x86_64
index=0
kernel=/boot/vmlinuz-3.10.0-229.4.2.el7.x86_64
args="ro rd.lvm.lv=rhel/root crashkernel=auto rd.lvm.lv=rhel/swap vconsole.font=latarcyrheb-
sun16 vconsole.keymap=us rhgb quiet LANG=en_US.UTF-8"
root=/dev/mapper/rhel-root
initrd=/boot/initramfs-3.10.0-229.4.2.el7.x86_64.img
title=Red Hat Enterprise Linux Server (3.10.0-229.4.2.el7.x86_64) 7.0 (Maipo)
```

尝试 Tab 补全以查看 /boot/ 目录中的可用内核。

从 GRUB 2 菜单条目中添加和删除参数

当与 -- args 结合使用时，可以使用 -- update-kernel 选项更新菜单条目，以添加新参数和 --remove-arguments 以删除现有参数。这些选项接受带引号的空格分隔列表。从 GRUB 2 菜单条目同时添加和删除参数的命令具有以下格式：

```
grubby --remove-args="argX argY" --args="argA argB" --update-kernel /boot/kernel
```

要从内核的 GRUB 2 菜单条目中添加和删除参数，请使用以下命令：

```
~]# grubby --remove-args="rhgb quiet" --args=console=ttyS0,115200 --update-kernel
/boot/vmlinuz-3.10.0-229.4.2.el7.x86_64
```

该命令将删除 Red Hat 图形引导参数，启用看到引导消息，并添加串行控制台。由于控制台参数将在行末尾添加，因此新控制台的优先级将高于配置的任何其他控制台。

要查看更改，请使用 --info 命令选项，如下所示：

```
~]# grubby --info /boot/vmlinuz-3.10.0-229.4.2.el7.x86_64
index=0
kernel=/boot/vmlinuz-3.10.0-229.4.2.el7.x86_64
args="ro rd.lvm.lv=rhel/root crashkernel=auto rd.lvm.lv=rhel/swap vconsole.font=latarcyrheb-
sun16 vconsole.keymap=us LANG=en_US.UTF-8 ttyS0,115200"
root=/dev/mapper/rhel-root
initrd=/boot/initramfs-3.10.0-229.4.2.el7.x86_64.img
title=Red Hat Enterprise Linux Server (3.10.0-229.4.2.el7.x86_64) 7.0 (Maipo)
```

使用相同参数更新所有内核菜单

要在所有内核菜单条目中添加相同的内核引导参数，请输入以下命令：

```
~]# grubby --update-kernel=ALL --args=console=ttyS0,115200
```

--update-kernel 参数还接受 DEFAULT 或以逗号分隔的内核索引号列表。

更改内核参数

要更改现有内核参数中的值，请再次指定 参数，根据需要更改值。例如，要更改虚拟控制台字体大小，使用以下命令：

```
~]# grubby --args=vconsole.font=latarcyrheb-sun32 --update-kernel /boot/vmlinuz-3.10.0-
229.4.2.el7.x86_64
index=0
```

```
kernel=/boot/vmlinuz-3.10.0-229.4.2.el7.x86_64
args="ro rd.lvm.lv=rhel/root crashkernel=auto rd.lvm.lv=rhel/swap vconsole.font=latarcyrheb-
sun32 vconsole.keymap=us LANG=en_US.UTF-8"
root=/dev/mapper/rhel-root
initrd=/boot/initramfs-3.10.0-229.4.2.el7.x86_64.img
title=Red Hat Enterprise Linux Server (3.10.0-229.4.2.el7.x86_64) 7.0 (Maipo)
```

有关更多命令选项，请参阅 grubby(8)手册页。

26.5. 自定义 GRUB 2 配置文件

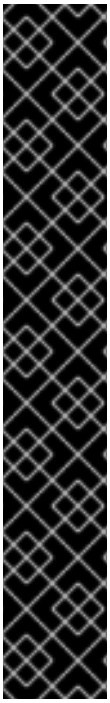
GRUB 2 脚本搜索用户的计算机并根据脚本找到的操作系统构建引导菜单。为了反映最新的系统引导选项，当内核更新或添加新内核时，会自动重新构建引导菜单。

但是，用户可能希望构建包含特定条目的菜单，或者按特定顺序拥有条目。GRUB 2 允许基本自定义引导菜单，使用户能够控制屏幕上实际显示的内容。

GRUB 2 使用一系列脚本来构建菜单；它们位于 /etc/grub.d/ 目录中。包括以下文件：

- 00_header，它将从 /etc/default/grub 文件中加载 GRUB 2 设置。
- 01_users，从 user.cfg 文件中读取超级用户密码。在 Red Hat Enterprise Linux 7.0 和 7.1 中，该文件仅在安装期间在 kickstart 文件中定义引导密码时创建，并且该文件以纯文本形式包含定义的密码。
- 10_Linux，在 Red Hat Enterprise Linux 的默认分区中找到内核。
- 30_OS-prober，为其他分区上的操作系统构建条目。
- 40_custom，模板，可用于创建额外的菜单条目。

/etc/grub.d/ 目录中的脚本按字母顺序读取，因此可重命名为更改特定菜单条目的引导顺序。



重要

要隐藏可引导内核列表，请不要在 /etc/default/grub 中将 GRUB_TIMEOUT 设置为 0。使用此类设置时，系统始终会在默认菜单条目中立即启动，如果默认内核无法引导，则无法引导较早的内核。

相反，为了避免 GRUB 2 在系统启动时显示可引导内核列表，请在 /etc/default/grub 文件中设置 GRUB_TIMEOUT_STYLE 选项，如下所示：

```
GRUB_TIMEOUT_STYLE=hidden
```

要在引导时显示列表，请在使用键盘或其他串行控制台显示 BIOS 信息时按 并按住任何字母数字键，并且 GRUB 2 将为您提供 GRUB 2 菜单。

26.5.1. 更改默认引导条目

默认情况下，/etc/default/grub 文件中的 GRUB_DEFAULT 指令的键是 保存 的词语。这指示 GRUB 2 将 saved_entry 指令指定的内核加载到位于 /boot/grub2/grubenv 的 GRUB 2 环境文件中。您可以使用 grub2-set-default 命令将另一个 GRUB 2 记录设置为默认值，该命令将更新 GRUB 2 环境文件。

默认情况下，saved_entry 值被设置为软件包类型为内核的最新安装的内核的名称 。这在 /etc/sysconfig/kernel 中由 UPDATEDAULT 和DEFAULT KERNEL 指令定义。该文件可由 root 用户查看，如下所示：

```
~]# cat /etc/sysconfig/kernel
# UPDATEDEFAULT specifies if new-kernel-pkg should make
# new kernels the default
UPDATEDEFAULT=yes

# DEFAULTKERNEL specifies the default kernel package type
DEFAULTKERNEL=kernel
```

The DEFAULTKERNEL 指令指定将用作默认软件包类型。安装类型为 kernel-debug 的软件包不会更改默认内核，而 DEFAULTKERNEL 设置为软件包类型 内核。

GRUB 2 支持使用数字值作为 saved_entry 指令的键，以更改载入操作系统的默认顺序。要指定应首先加载的操作系统，请将编号传递到 grub2-set-default 命令。例如：

```
~]# grub2-set-default 2
```

请注意，列表中菜单条目的位置由以零开头的数字表示；因此，在上面的示例中，将加载第三个条目。该值将被要安装的下一个内核的名称覆盖。

要强制系统始终使用特定的菜单条目，请使用菜单条目名称作为 /etc/default/grub 文件中的 GRUB_DEFAULT 指令的密钥。要列出可用的菜单条目，以 root 用户身份运行以下命令：

```
~]# awk -F\' '$1=="menuentry " {print $2}' /etc/grub2.cfg
```

文件名 /etc/grub2.cfg 是指向 grub.cfg 文件的符号链接，其位置取决于架构。出于可靠性的原因，本章中的其他示例不使用该符号链接。最好在写入文件时使用绝对路径，特别是在修复系统时。

对 /etc/default/grub 的更改需要重新构建 grub.cfg 文件，如下所示：

- 在基于 BIOS 的机器中，以 root 用户身份运行以下命令：

  ```
  ~]# grub2-mkconfig -o /boot/grub2/grub.cfg
  ```

- 在基于 UEFI 的机器中，以 root 用户身份运行以下命令：

  ```
  ~]# grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
  ```

26.5.2. 编辑菜单条目

如果需要，使用不同的参数准备新的 GRUB 2 文件，请在 /etc/default/grub 文件中编辑 GRUB_CMDLINE_LINUX 键的值。请注意，您可以为 GRUB_CMDLINE_LINUX 密钥指定多个参数，这与在 GRUB 2 引导菜单中添加参数类似。例如：

```
GRUB_CMDLINE_LINUX="console=tty0 console=ttyS0,9600n8"
```

console=tty0 是第一个虚拟终端，console=ttyS0 是要使用的串行终端。

对 /etc/default/grub 的更改需要重新构建 grub.cfg 文件，如下所示：

- 在基于 BIOS 的机器中，以 root 用户身份运行以下命令：

```
~]# grub2-mkconfig -o /boot/grub2/grub.cfg
```

- 在基于 UEFI 的机器中，以 root 用户身份运行以下命令：

```
~]# grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

26.5.3. 添加新条目

执行 grub2-mkconfig 命令时，GRUB 2 根据位于 /etc/grub.d/ 目录中的文件搜索 Linux 内核和其他操作系统。/etc/grub.d/10_linux 脚本搜索同一分区上已安装的 Linux 内核。/etc/grub.d/30_os-prober 脚本将搜索其他操作系统。更新内核时，菜单条目也会自动添加到引导菜单中。

位于 /etc/grub.d/ 目录中的 40_custom 文件是自定义条目的模板，如下所示：

```
#!/bin/sh
exec tail -n +3 $0
# This file provides an easy way to add custom menu entries. Simply type the
# menu entries you want to add after this comment. Be careful not to change
# the 'exec tail' line above.
```

可以编辑或复制此文件。请注意，一个有效的菜单条目必须至少包含以下项：

```
menuentry "<Title>"{
<Data>
}
```

26.5.4. 创建自定义菜单

如果您不希望菜单项自动更新，您可以创建自定义菜单。

重要

在继续之前，如果您需要稍后恢复更改，请备份 /etc/grub.d/ 目录的内容。

注意

请注意，修改 /etc/default/grub 文件不会影响创建自定义菜单。

1. 在基于 BIOS 的机器上，复制 /boot/grub2/grub.cfg 或 UEFI 计算机上的内容，复制 /boot/efi/EFI/redhat/grub.cfg 的内容。将 grub.cfg 的内容放在现有标题行的 /etc/grub.d/40_custom 文件中。必须保留 40_custom 脚本的可执行部分。

2. 从放入 /etc/grub.d/40_custom 文件中的内容，仅需要 菜单输入 块来创建自定义菜单。/boot/grub2/grub.cfg 和 /boot/efi/EFI/redhat/grub.cfg 文件可能会在 菜单输入 块的上方、下面包含功能规格和其他内容。如果您将这些不必要的行放在上一步中的 40_custom 文件中，请将其清除。

   这是自定义 40_custom 脚本的示例 ：

```
#!/bin/sh
exec tail -n +3 $0
# This file provides an easy way to add custom menu entries. Simply type the
# menu entries you want to add after this comment. Be careful not to change
# the 'exec tail' line above.

menuentry 'First custom entry' --class red --class gnu-linux --class gnu --class os
$menuentry_id_option 'gnulinux-3.10.0-67.el7.x86_64-advanced-32782dd0-4b47-4d56-
a740-2076ab5e5976' {
    load_video
    set gfxpayload=keep
    insmod gzio
    insmod part_msdos
    insmod xfs
    set root='hd0,msdos1'
    if [ x$feature_platform_search_hint = xy ]; then
     search --no-floppy --fs-uuid --set=root --hint='hd0,msdos1' 7885bba1-8aa7-4e5d-
a7ad-821f4f52170a
    else
     search --no-floppy --fs-uuid --set=root 7885bba1-8aa7-4e5d-a7ad-821f4f52170a
    fi
    linux16 /vmlinuz-3.10.0-67.el7.x86_64 root=/dev/mapper/rhel-root ro
rd.lvm.lv=rhel/root vconsole.font=latarcyrheb-sun16 rd.lvm.lv=rhel/swap
vconsole.keymap=us crashkernel=auto rhgb quiet LANG=en_US.UTF-8
    initrd16 /initramfs-3.10.0-67.el7.x86_64.img
}
menuentry 'Second custom entry' --class red --class gnu-linux --class gnu --class os
$menuentry_id_option 'gnulinux-0-rescue-07f43f20a54c4ce8ada8b70d33fd001c-
advanced-32782dd0-4b47-4d56-a740-2076ab5e5976' {
    load_video
    insmod gzio
    insmod part_msdos
    insmod xfs
    set root='hd0,msdos1'
    if [ x$feature_platform_search_hint = xy ]; then
     search --no-floppy --fs-uuid --set=root --hint='hd0,msdos1' 7885bba1-8aa7-4e5d-
a7ad-821f4f52170a
    else
     search --no-floppy --fs-uuid --set=root 7885bba1-8aa7-4e5d-a7ad-821f4f52170a
    fi
```

```
    linux16 /vmlinuz-0-rescue-07f43f20a54c4ce8ada8b70d33fd001c
root=/dev/mapper/rhel-root ro rd.lvm.lv=rhel/root vconsole.font=latarcyrheb-sun16
rd.lvm.lv=rhel/swap vconsole.keymap=us crashkernel=auto rhgb quiet
    initrd16 /initramfs-0-rescue-07f43f20a54c4ce8ada8b70d33fd001c.img
}
```

3. 从 /etc/grub.d/ 目录中除以下内容外删除所有文件：

   - 00_header,
   - 40_custom,
   - 01_users （如果存在）
   - 和 README.

   或者，如果您要将文件保存在 /etc/grub2.d/ 目录中，通过运行 chmod a-x <file_name> 命令使其不可执行。

4. 根据需要编辑、添加或删除 40_custom 文件中的菜单条目。

5. 运行 grub 2-mkconfig -o 命令重建 grub.cfg 文件，如下所示：

   - 在基于 BIOS 的机器中，以 root 用户身份运行以下命令：

     ```
     ~]# grub2-mkconfig -o /boot/grub2/grub.cfg
     ```

   - 在基于 UEFI 的机器中，以 root 用户身份运行以下命令：

     ```
     ~]# grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
     ```

26.6. 使用密码保护 GRUB 2

GRUB 2 提供两种类型的密码保护：

- 修改菜单条目时需要密码，但不需要引导现有菜单条目 ；
- 修改菜单条目和引导一个、多个或所有菜单条目需要密码。

将 GRUB 2 配置为仅为修改条目需要一个密码

要需要密码身份验证才能修改 GRUB 2 条目，请按照以下步骤操作：

1. 以 root 用户身份运行 grub2-setpassword 命令：

```
~]# grub2-setpassword
```

2. 输入并确认密码：

```
Enter password:
Confirm password:
```

按照此过程创建包含密码哈希的 /boot/grub2/user.cfg 文件。此密码的用户 root 在 /boot/grub2/grub.cfg 文件中定义。在这个版本中，在启动过程中修改引导条目需要指定 root 用户名和密码。

将 GRUB 2 配置为需要修改和引导条目的密码

使用 grub2-setpassword 设置密码 可防止菜单条目未经授权的修改，但不能未经授权的启动。要同时需要密码来引导条目，请使用 grub2-setpassword 设置密码后按照以下步骤执行：

警告

如果忘记 GRUB 2 密码，您将无法在以下过程中引导您重新配置的条目。

1. 打开 /boot/grub2/grub.cfg 文件。

2. 通过搜索 菜单项开头的行，找到您要使用密码保护的引导条目。

3. 从菜单条目块中删除 --unrestricted 参数，例如：

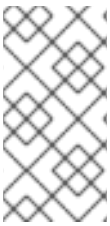
```
[file contents truncated]
menuentry 'Red Hat Enterprise Linux Server (3.10.0-327.18.2.rt56.223.el7_2.x86_64) 7.2
(Maipo)' --class red --class gnu-linux --class gnu --class os --unrestricted
$menuentry_id_option 'gnulinux-3.10.0-327.el7.x86_64-advanced-c109825c-de2f-4340-
a0ef-4f47d19fe4bf' {
    load_video
    set gfxpayload=keep
[file contents truncated]
```

4. 保存并关闭该文件。

现在，即使是引导该条目，也需要输入 root 用户名和密码。

注意

安装新内核版本时，对 /boot/grub2/grub.cfg 的手动更改会保留，但在使用 grub 2-mkconfig 命令重新生成 grub.cfg 时会丢失。因此，若要保留密码保护，请在每次使用 grub2-mkconfig 后使用上述程序。



注意

如果您从 /boot/grub2/grub.cfg 文件的每个菜单条目中删除 --unrestricted 参数，则所有新安装的内核都将在 未限制的情况下创建菜单条目， 因此自动继承密码保护。

在升级到 Red Hat Enterprise Linux 7.2 前设置密码

Red Hat Enterprise Linux 7.2 中添加了 grub2-setpassword 工具，现在是设置 GRUB 2 密码的标准方法。这与之前版本的 Red Hat Enterprise Linux 不同，后者的引导条目需要在 /etc/grub.d/40_custom 文件中手动指定，超级用户在 /etc/grub.d/01_users 文件中需要手动指定。

其他 GRUB 2 用户

在没有 --unrestricted 参数的情况下引导条目需要 root 密码。但是，GRUB 2 也支持创建其他非 root 用户，这些用户可以在不提供密码的情况下引导此类条目。修改条目仍然需要 root 密码。有关创建此类用户的详情请参考 GRUB 2 手册。

26.7. 重新安装 GRUB 2

重新安装 GRUB 2 是一种便捷的方式，可以修复通常由 GRUB 2 安装错误导致的问题，或者系统丢失。重新安装 GRUB 2 的其他原因包括：

- 从之前版本的 GRUB 升级.
- 用户需要 GRUB 2 引导装载程序来控制安装的操作系统。但是，一些操作系统安装有自己的启动加载器。重新安装 GRUB 2 会将控制权返回到所需的操作系统。
- 将启动信息添加到另一个驱动器.

26.7.1. 在基于 BIOS 的机器中重新安装 GRUB 2

使用 grub2-install 命令时，会更新启动信息并恢复缺少的文件。请注意，仅当文件未损坏时，才会恢复这些文件。

如果系统正常运行，则使用 grub2-install device 命令重新安装 GRUB 2。例如，如果 sda 是您 的设备：

```
~]# grub2-install /dev/sda
```

26.7.2. 在基于 UEFI 的机器上重新安装 GRUB 2

使用 yum reinstall grub2-efi shim 命令时，会更新引导信息并恢复缺少的文件。请注意，仅当文件未损坏时，才会恢复这些文件。

如果系统正常运行，请使用 yum reinstall grub2-efi shim 命令重新安装 GRUB 2。例如：

```
~]# yum reinstall grub2-efi shim
```

26.7.3. 重置和重新安装 GRUB 2

这个方法可完全删除所有 GRUB 2 配置文件和系统设置。应用此方法将所有配置设置重置为默认值。删除配置文件并随后重新安装 GRUB 2 修复故障，导致文件损坏和配置不正确。要做到这一点，以 root

用户身份执行以下步骤：

1. 运行 rm /etc/grub.d/* 命令；
2. 运行 rm /etc/sysconfig/grub 命令；
3. 对于 EFI 系统，运行以下命令：

```
~]# yum reinstall grub2-efi shim grub2-tools
```

4. 对于 BIOS 和 EFI 系统，运行这个命令：

```
~]# yum reinstall grub2-tools
```

5. 运行 grub 2-mkconfig -o 命令重建 grub.cfg 文件，如下所示：
   - 在基于 BIOS 的机器中，以 root 用户身份运行以下命令：

     ```
     ~]# grub2-mkconfig -o /boot/grub2/grub.cfg
     ```

   - 在基于 UEFI 的机器中，以 root 用户身份运行以下命令：

     ```
     ~]# grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
     ```

6. 现在按照 第 26.7 节 “重新安装 GRUB 2” 中的步骤在 /boot/ 分区中恢复 GRUB 2。

26.8. 从 GRUB LEGACY 升级到 GRUB 2

当您从版本 6 原位升级到 Red Hat Enterprise Linux(RHEL)到 7 时，从 GRUB Legacy 升级到 GRUB 2 不会自动进行，但应该手动进行。执行 GRUB 升级，原因如下：

- 在 RHEL 7 及更新的版本中，GRUB Legacy 不再被维护，也不会接收更新。

- GRUB Legacy 无法在没有 /boot/ 目录的系统上引导。
- GRUB 2 具有更多功能且更为可靠。
- GRUB 2 支持更多硬件配置、文件系统和驱动器布局。

在操作系统原位升级后，从 GRUB Legacy 升级到 GRUB 2

从 GRUB Legacy 升级到 GRUB 2

1. 确定 Red Hat Upgrade Tool 已卸载 GRUB Legacy 软件包：

```
~]# yum remove grub
```

注意

卸载 grub2 软件包不会影响已安装的 GRUB Legacy 引导加载程序。

2. 确保已安装 grub2 软件包。如果在升级到 RHEL 7 后 grub2 不在系统中，您可以运行以下命令手动安装它：

```
~]# yum install grub2
```

3. 如果系统使用 EFI 引导，如果缺少以下软件包：

```
~]# yum install grub2-efi-x64 shim-x64
```

生成 GRUB 2 配置文件

这部分论述了如何在不删除原始 GRUB 传统配置 的情况下添加 GRUB 2 配置。如果 GRUB 2 无法正常工作，这个过程会保留 GRUB Legacy 配置。

1. 使用以下选项之一 手动创建 /etc/default/grub 文件：
   - 创建 /etc/default/grub 文件。详情请查看 如何在 Red Hat Enterprise Linux 7? 文档中重新创建缺少的 /etc/default/grub 文件。

- 从类似 Red Hat Enterprise Linux 7 系统复制 /etc/default/grub 文件，并相应地调整文件。

2. 根据您的引导装载程序：

   a. 如果系统使用旧的 BIOS 引导，请安装 GRUB 2 指示安装设备：

   ```
   ~]# grub2-install /dev/<DEVICE_NAME> --grub-setup=/bin/true
   ```

   grub2-install 命令将 GRUB 镜像安装到 /boot/grub 目标目录中。

   GRUB -setup=/bin/true 选项可确保旧 GRUB Legacy 配置不会被删除。

   b. 如果系统使用 EFI 引导，请为 shim 引导装载程序创建一个引导条目，并更改 BootOrder 变量以使固件引导 GRUB 2 通过 shim ：

   ```
   ~]# efibootmgr -c -L 'Red Hat Enterprise Linux 7' -d /dev/device_name -p 1 -l
   '\EFI\redhat\shimx64.efi'
   ```

   使用可引导设备文件替换 /dev/device_name。

警告

请注意配置文件扩展的不同：

- .conf 用于 GRUB
- .cfg 用于 GRUB 2

下一步中请勿错误地覆盖旧的 GRUB 配置文件。

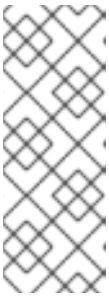
3. 生成 GRUB 2 配置文件：

   a. 如果系统使用旧的 BIOS：

   ```
   ~]# grub2-mkconfig -o /boot/grub2/grub.cfg
   ```

   b. 如果系统使用 EFI：

   ```
   ~]# grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
   ```

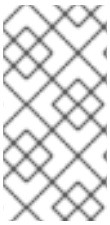


注意

有关自定义生成的 GRUB 2 配置文件，请参阅 第 26.5 节 “自定义 GRUB 2 配置文件”。您应该在 /etc/default/grub 中进行更改，而不是直接在 /boot/grub2/grub.cfg 中 进行更改。否则，每次重新生成文件时，/boot/grub2/grub.cfg 中的更改都会丢失。

使用 GRUB Legacy 引导装载程序测试 GRUB 2 仍然安装

这部分论述了如何 在不删除 GRUB Legacy 配置的情况下测试 GRUB 2。在验证 GRUB 2 配置之前，GRUB Legacy 配置需要保留；否则，系统可能无法引导。为了安全地测试 GRUB 2 配置，我们将从 GRUB Legacy 启动 GRUB 2。



注意

这部分只适用于旧的 BIOS 引导。如果是 EFI，则旧启动加载器和新启动加载器都有引导条目，您可以通过使用 EFI 固件设置选择引导条目来引导旧的旧 GRUB。

1. 将新部分添加到 /boot/grub/grub.conf ：

   对于具有独立 /boot 分区的系统，请使用：

   ```
   title GRUB 2 Test
   root (hd0,0)
   kernel /grub2/i386-pc/core.img
   boot
   ```

   将 (hd0,0)替换为 GRUB Legacy 可引导设备的名称。

   对于没有独立 /boot 分区的系统，请使用：

   ```
   title GRUB 2 Test
   root (hd0,0)
   kernel /boot/grub2/i386-pc/core.img
   boot
   ```

   将 (hd0,0)替换为 GRUB Legacy 可引导设备的名称。

2. 重启系统。

3. 出现 GRUB Legacy 菜单时，选择 GRUB 2 Test 条目。

4. 当显示 GRUB 2 菜单时，请选择要引导的内核。

5. 如果上述内容不起作用，则重新启动，且 不要 在下次引导时选择 GRUB 2 Test 条目。

在使用 BIOS 的系统中替换 GRUB Legacy 引导装载程序

如果 GRUB 2 成功：

1. **将 GRUB Legacy 引导装载程序替换为 GRUB 2 引导装载程序：**

```
~]# grub2-install /dev/sdX
```

2. **删除旧的 GRUB Legacy 配置文件：**

```
~]# rm /boot/grub/grub.conf
```

3. **重启系统：**

```
~]# reboot
```

在使用 EFI 的系统上删除 GRUB Legacy

如果 GRUB 2 成功：

1. **检查 /boot/efi/EFI/redhat/ 目录的内容，并删除只与传统 GRUB 相关的过时文件：**

```
~]# rm /boot/efi/EFI/redhat/grub.efi
~]# rm /boot/efi/EFI/redhat/grub.conf
```

2. **如果您使用 Preupgrade Assistant 和 Red Hat Upgrade Tool 工具程序执行 RHEL 6 到 RHEL 7 的原位升级，请删除以上以 .preupg 后缀结尾的文件的备份副本：**

```
~]# rm /boot/efi/EFI/redhat/*.preupg
```

3. **使用 efibootmgr 命令查找引用 \EFI\redhat\grub.efi 文件的旧引导条目：**

```
~]# efibootmgr -v | grep '\\EFI\\redhat\\grub.efi'
```

输出示例：

```
Boot0001* Linux    HD(1,GPT,542e410f-cbf2-4cce-9f5d-
61c4764a5d54,0x800,0x64000)/File(\EFI\redhat\grub.efi)
```

本例中的条目编号为 0001。

4. 删除标识的引导条目。以下命令从上例中删除引导条目：

```
~]# efibootmgr -Bb 0001
```

如果您有多个这样的引导条目，请删除所有确定的旧引导条目。

警告

从旧版本（如 RHEL6）升级到 RHEL7 后，在成功完成 GRUB Legacy bootloader 手动升级到 GRUB 2 之前，该操作系统不被支持。这是因为不支持在主版本间安装软件包。对于 RHEL 7，仅支持、开发和测试 GRUB 2；与 RHEL 6 的 GRUB Legacy 不同。

26.9. 通过串行控制台的 GRUB 2

这部分论述了如何在没有显示或键盘的机器上配置 GRUB 2 串行通信。

要通过串行连接访问 GRUB 2 终端，必须在内核定义中添加另一个选项，以使特定内核监控串行连接。

例如：

```
console=ttyS0,9600n8
```

其中 console=ttyS0 是要使用的串行终端，9600 是 baud 速率，n 表示没有奇偶校验，而 8 是以位为单位的字词长度。对于以下日志文件等任务来说，最好使用更高的 baud 速率（如 115200）。

有关串行控制台设置的详情，请参考 “可安装和外部文档”一节

26.9.1. 为单个引导配置 GRUB 2

要将系统设置为仅在单个引导过程中使用串行终端，显示 GRUB 2 引导菜单时，将光标移至要启动的内核，然后按 e 键编辑内核参数。删除 rhgb 和 quiet 参数，并在 linux16 行末尾添加控制台参数，如下所示：

```
linux16   /vmlinuz-3.10.0-0.rc4.59.el7.x86_64 root=/dev/mapper/rhel-root ro rd.md=0 rd.dm=0
rd.lvm.lv=rhel/swap crashkernel=auto rd.luks=0 vconsole.keymap=us rd.lvm.lv=rhel/root
console=ttyS0,9600
```

这些设置不是永久性的，仅适用于一次引导。

26.9.2. 为持久性更改配置 GRUB 2

若要永久更改系统上的菜单条目，请使用 grubby 工具。例如，要更新默认内核的条目，请按如下所示输入命令：

```
~]# grubby --remove-args="rhgb quiet" --args=console=ttyS0,9600 --update-kernel=DEFAULT
```

--update-kernel 参数还接受关键字 ALL 或以逗号分隔的内核索引编号列表。有关使用 grubby 的更多信息，请参阅 “从 GRUB 2 菜单条目中添加和删除参数”一节。

26.9.3. 配置新的 GRUB 2 文件

如果需要，请在 /etc/default/grub 文件中添加以下两行：

```
GRUB_TERMINAL="serial"
GRUB_SERIAL_COMMAND="serial --speed=9600 --unit=0 --word=8 --parity=no --stop=1"
```

第一行将禁用图形终端。请注意，指定 GRUB_TERMINAL 键会覆盖 GRUB_TERMINAL_INPUT 和 GRUB_TERMINAL_OUTPUT 的值。在第二行中，调整 Baud 速率、奇偶校验和其他值以适合您的环境和硬件。对于以下日志文件等任务来说，最好使用更高的 baud 速率（如 115200 ）。完成 /etc/default/grub 文件中的更改后，需要更新 GRUB 2 配置文件。

运行 grub2-mkconfig -o 命令重建 grub.cfg 文件，如下所示：

- 在基于 BIOS 的机器中，以 root 用户身份运行以下命令：

```
~]# grub2-mkconfig -o /boot/grub2/grub.cfg
```

- 在基于 UEFI 的机器中，以 root 用户身份运行以下命令：

```
~]# grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

26.9.4. 使用屏幕连接到串行控制台

屏幕 工具充当强大的串行终端。要安装它，以 root 用户身份运行：

```
~]# yum install screen
```

要使用串行控制台连接到您的机器，请使用以下格式命令：

```
screen /dev/console_port baud_rate
```

默认情况下，如果没有指定选项，屏幕 将使用标准 9600 升序。要设置更高的 Baud 速率，请输入：

```
~]$ screen /dev/console_port 115200
```

这里的 console_port is ttyS0, or ttyUSB0 等等。

要结束 屏幕中 的会话，请按 Ctrl+a，键入 :quit 并按 Enter。

更多选项和详细信息请查看 screen(1) 手册页。

26.10. 引导期间编辑终端

可以修改菜单条目，并在启动时传输到内核的参数。这通过菜单条目编辑器界面完成，该界面在启动加载器菜单中所选菜单条目上按 e 键时触发。Esc 键丢弃任何更改并重新加载标准菜单界面。c 键加载命令行界面。

命令行界面是最基本的 GRUB 2 界面，但它也是授予最多控制权的界面。通过命令行，可以键入任何相关的 GRUB 2 命令，后跟 Enter 键来执行它们。此界面具有与 shell 类似的一些高级功能，包括基于上

下文的 Tab 键 完成和 Ctrl+a 以移到行首，而 Ctrl+e 可移到行末。此外，箭头、Home、Ed 和 Delete 键的工作方式与 bash shell 中相同。

26.10.1. 引导至救援模式

救援模式提供了一个方便的单用户环境，并允许您在无法完成正常引导过程时修复您的系统。在救援模式中，系统会尝试挂载所有本地文件系统并启动一些重要的系统服务，但不激活网络接口或者同时允许更多的用户登录到该系统。在 Red Hat Enterprise Linux 7 中，救援模式等同于单用户模式，需要 root 密码。

1. 要在启动过程中进入救援模式，请在 GRUB 2 引导屏幕上按 e 键进行编辑。

2. 在 64 位 IBM Power 系列中的 linux 行的末尾添加以下参数，在基于 x86-64 BIOS 的系统中 linux16 行，或者 UEFI 系统上的 linuxefi 行：

   ```
   systemd.unit=rescue.target
   ```

   Ctrl+a Ctrl+e 键，分别跳到行首和行尾。在某些系统上，Home 和 End 也起作用。

   请注意，等同的参数（1、s 和 single ）也可以传输到内核。

3. Ctrl+x 使用 参数启动系统。

26.10.2. 引导至紧急模式

紧急模式提供最最小的环境，并允许您在系统无法进入救援模式的情况下修复您的系统。在紧急模式中，系统仅挂载用于读取的 root 文件系统，不会尝试挂载任何其他本地文件系统，不激活网络接口，并且仅启动很少的基本服务。在 Red Hat Enterprise Linux 7 中，紧急模式需要 root 密码。

1. 要进入紧急模式，请在 GRUB 2 引导屏幕上按 e 键进行编辑。

2. 在 64 位 IBM Power 系列中的 linux 行的末尾添加以下参数，在基于 x86-64 BIOS 的系统中 linux16 行，或者 UEFI 系统上的 linuxefi 行：

   ```
   systemd.unit=emergency.target
   ```

Ctrl+a Ctrl+e 键，分别跳到行首和行尾。在某些系统上，Home 和 End 也起作用。

请注意，等同的参数（emergency 和 -b ）也可以传输到内核。

3. Ctrl+x 使用参数启动系统。

26.10.3. 引导至 Debug Shell

systemd 调试 shell 在启动过程的早期阶段提供了一个 shell，可用于诊断 systemd 相关的启动问题。进入 debug shell 时，system ctl 命令（如 systemctl list-jobs 和 systemctl list-units ）可用于查找引导问题的原因。此外，可以将 debug 选项添加到内核命令行中，以增加日志消息的数量。对于 systemd，内核命令行选项 debug 现在是 systemd.log_level=debug 的快捷方式。

添加 Debug Shell 命令

要只为此会话激活 debug shell，请按照如下所示：

1. 在 GRUB 2 引导屏幕上，将光标移至要编辑的菜单条目，然后按 e 键进行编辑。

2. 在 64 位 IBM Power 系列中的 linux 行的末尾添加以下参数，在基于 x86-64 BIOS 的系统中 linux16 行，或者 UEFI 系统上的 linuxefi 行：

```
systemd.debug-shell
```

（可选）添加 debug 选项。

Ctrl+a Ctrl+e 键，分别跳到行首和行尾。在某些系统上，Home 和 End 也起作用。

3. Ctrl+x 使用参数启动系统。

如果需要，可以通过 systemctl enable debug-shell 命令启用 debug shell 来在每次引导时启动 debug shell。或者，grubby 工具可用于对 GRUB 2 菜单中的内核命令行进行永久性更改。有关使用 grubby 的更多信息，请参阅 第 26.4 节 “使用 grubby 工具对 GRUB 2 菜单进行持久更改”。

警告

永久启用 debug shell 是一种安全风险，因为不需要进行身份验证即可使用。调试会话结束后禁用它。

连接到 Debug Shell

在启动过程中，systemd-debug-generator 将在 TTY9 上配置 debug shell。

1. CtrlAltF9 连接到调试 shell。如果使用虚拟机，发送此组合键需要虚拟化应用程序的支持。例如，如果使用 虚拟机管理器，请从菜单中选择 Send Key → Ctrl+Alt+F9。

2. debug shell 不需要身份验证，因此应当在 TTY9 中看到类似如下的提示：
[root@localhost /]#

3. 如果需要，请按如下所示输入一个命令来验证您在 debug shell 中：

```
/]# systemctl status $$
● debug-shell.service - Early root shell on /dev/tty9 FOR DEBUGGING ONLY
  Loaded: loaded (/usr/lib/systemd/system/debug-shell.service; disabled; vendor
preset: disabled)
  Active: active (running) since Wed 2015-08-05 11:01:48 EDT; 2min ago
   Docs: man:sushell(8)
 Main PID: 450 (bash)
  CGroup: /system.slice/debug-shell.service
     ├─ 450 /bin/bash
     └─1791 systemctl status 450
```

4. 要返回到默认 Ctrl+Alt+F1。

要诊断启动问题，可以通过在内核命令行上添加 systemd.mask=unit_name 几次来屏蔽某些 systemd 单元。要在引导过程中启动其他进程，请将 systemd.wants=unit_name 添加到内核命令行。systemd-debug-generator(8)手册页 描述了这些选项。

26.10.4. 更改和重置根密码

设置 root 密码是 Red Hat Enterprise Linux 7 安装的强制部分。如果忘记或丢失 root 密码，可以重置它，但属于 wheel 组成员的用户可以更改 root 密码，如下所示：

```
~]$ sudo passwd root
```

请注意，在 GRUB 2 中，重置密码不再以单用户模式执行，因为它包含在 Red Hat Enterprise Linux 6 中的 GRUB 中。现在，需要 root 密码 才能在单用户模式 和紧急 模式下运行。

此处显示了两个重置 root 密码的步骤：

- 使用安装磁盘重置 Root 密码 带您进入 shell 提示符，而无需编辑 GRUB 2 菜单。它是两个程序的时间较短，也是推荐的方法。您可以使用引导磁盘或常规的 Red Hat Enterprise Linux 7 安装磁盘。
- 使用 rd.break 重置 Root 密码 在将控制权从 initramfs 传递给 systemd 之前，使用 rd.break 来中断引导过程。此方法的缺点在于它需要更多步骤，包括必须编辑 GRUB 2 菜单，涉及在消耗的 SELinux 文件重新标记或更改 SELinux 强制模式之间选择，然后在启动完成后恢复 /etc/shadow/ 的 SELinux 安全上下文。

使用安装磁盘重置 Root 密码

1. 启动系统并显示 BIOS 信息时，为引导菜单选择 选项并选择 从安装磁盘引导。
2. 选择 Troubleshooting。
3. 选择 Rescue a Red Hat Enterprise Linux System。
4. 选择 Continue，这是默认选项。此时，如果找到加密的文件系统，您将获得密码短语。
5. 按确定以确认显示的信息，直至显示 shell 提示符。
6. 按如下所示更改文件系统 root ：

```
sh-4.2# chroot /mnt/sysimage
```

7. 输入 passwd 命令并按照命令行中显示的说明更改 root 密码。

8. 删除 自动相关文件，以防止消耗 SELinux 重新标记磁盘：

```
sh-4.2# rm -f /.autorelabel
```

9. 输入 exit 命令退出 chroot 环境。

10. 再次输入 exit 命令以恢复初始化并完成系统引导。

使用 rd.break 重置 Root 密码

1. 启动系统，并在 GRUB 2 引导屏幕上按 e 键进行编辑。

2. 从 linux16 行的末尾或附近删除 rhgb 和 quiet 参数，或从 UEFI 系统上的 linuxefi 中删除。

   Ctrl+a Ctrl+e 键，分别跳到行首和行尾。在某些系统上，Home 和 End 也起作用。

   重要

   必须删除 rhgb 和 quiet 参数，才能启用系统消息。

3. 在 64 位 IBM Power 系列中的 linux 行的末尾添加以下参数，在基于 x86-64 BIOS 的系统中 linux16 行，或者 UEFI 系统上的 linuxefi 行：

```
rd.break enforcing=0
```

   添加 enforcing=0 选项可忽略 SELinux 重新标记过程所需的时间。

   在将控制权交给 Linux 内核 之前，init ramfs 将停止，使您能够使用 root 文件系统。

   请注意，init ramfs 提示将显示在 Linux 行中指定的最后一个控制台中。

4. Ctrl+x 使用更改的参数启动系统。

使用加密的文件系统时，此时需要输入密码。但是，密码提示符可能不会显示，因为日志消息会屏蔽密码提示。您可以按 Backspace 键来查看提示。释放密钥并输入加密文件系统的密码，同时忽略日志记录消息。

此时会显示 initramfs switch_root 提示符。

5. 在 /sysroot/ 中，文件系统以只读形式挂载。如果文件系统不可写入，则不允许更改密码。

将文件系统重新挂载为可写：

```
switch_root:/# mount -o remount,rw /sysroot
```

6. 在启用 write 时，文件系统会重新挂载。

按如下方式更改文件系统的 根 ：

```
switch_root:/# chroot /sysroot
```

提示更改为 sh-4.2#。

7. 输入 passwd 命令并按照命令行中显示的说明更改 root 密码。

请注意，如果系统不可写入，passwd 工具 会失败并显示以下错误：

```
Authentication token manipulation error
```

8. 更新密码文件会导致文件带有不正确的 SELinux 安全上下文。要在下一次系统引导时重新标记所有文件，请输入以下命令：

```
sh-4.2# touch /.autorelabel
```

另外，为节省重新标记大磁盘所需的时间，您可以省略此步骤，只要在第 3 步中包含 enforcing=0 选项。

9. 将文件系统重新挂载为只读：

```
sh-4.2# mount -o remount,ro /
```

10. 输入 exit 命令退出 chroot 环境。

11. 再次输入 exit 命令以恢复初始化并完成系统引导。

使用加密的文件系统时，此时需要一个通过词语或短语。但是，密码提示符可能不会显示，因为日志消息会屏蔽密码提示。您可以按住 Backspace 键来查看提示。释放密钥并输入加密文件系统的密码，同时忽略日志记录消息。

注意

请注意，SELinux 重新标记过程可能需要很长时间。系统重新启动将在过程完成后自动进行。

12. 如果您在第 3 步中添加了 enforcing=0 选项，并在第 8 步中省略 touch /.autorelabel 命令，请输入以下命令来恢复 /etc/shadow 文件的 SELinux 安全上下文：

```
~]# restorecon /etc/shadow
```

输入以下命令重新打开 SELinux 策略强制并验证它是否启动：

```
~]# setenforce 1
~]# getenforce
Enforcing
```

26.11. 统一可扩展固件接口(UEFI)安全引导

统一可扩展固件接口 (UEFI)安全引导技术可确保系统固件检查系统引导装载程序是否使用由固件中包含的公钥数据库授权的加密密钥签名。通过在下一阶段引导装载程序和内核中进行签名验证，可以防止执行尚未由可信密钥签名的内核空间代码。

信任链从固件到签名的驱动程序和内核模块，如下方所示：第一阶段引导装载程序(shim.efi) 由 UEFI

私钥签名，并由存储在固件数据库中的证书颁发机构(CA)签名的公钥进行身份验证。The shim.efi 包含红帽公钥"Red Hat 安全引导（CA 密钥 1）"，用于验证 GRUB 2 启动加载器、grubx64.efi 和红帽内核。反过来，内核包含用于验证驱动程序和模块的公钥。

安全引导(Secure Boot)是 Unified Extensible Firmware Interface(UEFI)规范中的引导路径验证组件。该规范定义了：

- 用于非易失性存储中加密保护 UEFI 变量的编程接口，
- 可信 X.509 根证书如何存储在 UEFI 变量中，
- UEFI 应用程序（如引导装载程序和驱动程序）的验证，
- 撤销已知错误的证书和应用程序哈希的流程。

UEFI 安全引导不会阻止安装或删除第二阶段引导装载程序，也不要求用户明确确认此类更改。签名在引导过程中被验证，而不是在安装或更新引导装载程序时验证。因此，UEFI 安全引导不会停止引导路径操作，这有助于检测未授权的更改。新的启动加载器或内核将正常工作，只要它由系统信任的密钥签名。

26.11.1. Red Hat Enterprise Linux 7 中的 UEFI 安全引导支持

Red Hat Enterprise Linux 7 包括对 UEFI 安全引导功能的支持，这意味着可以在启用了 UEFI 安全引导的系统上安装并运行 Red Hat Enterprise Linux 7。在启用了安全引导技术的基于 UEFI 的系统上，载入的所有驱动程序必须使用可信密钥签名，否则系统将不会接受它们。红帽提供的所有驱动程序都由红帽的私钥签名，并由内核中对应的红帽公钥进行身份验证。

如果要加载外部构建的驱动程序（不是在 Red Hat Enterprise Linux DVD 上提供的驱动程序），您必须确保对这些驱动程序进行了签名。

有关签署自定义驱动程序的信息，请参见《红帽企业 Linux 7 内核管理指南》。

UEFI 安全引导的限制

由于 Red Hat Enterprise Linux 7 中的 UEFI 安全引导支持旨在确保系统仅在正确验证签名后运行内核模式代码，因此存在某些限制。

GRUB 2 模块加载已被禁用，因为没有用于签署和验证 GRUB 2 模块的基础架构，这意味着允许加载它们将构成在安全引导定义的安全边界内执行不受信任的代码。相反，红帽提供了一个经过签名的 GRUB 2 二进制文件，其中包含 Red Hat Enterprise Linux 7 支持的所有模块。

有关详细信息，请参阅 **UEFI 安全引导所带来的红帽知识库文章**。

26.12. 其它资源

有关 GRUB 2 引导装载程序的详情，请查看以下资源：

安装的文档

- /usr/share/doc/grub2-tools-version-number/ - 此目录包含有关使用和配置 GRUB 2 的信息。version-number 对应于安装的 GRUB 2 软件包的版本。
- info grub2 - GRUB 2 info page 包含教程、用户参考手册、程序员参考手册，以及有关 GRUB 2 及其用法的常见问题文档。
- grubby(8) - 用于配置 GRUB 和 GRUB 2 的命令行工具的 man page。
- new-kernel-pkg(8) - 工具的 man page，用于脚本内核安装。

可安装和外部文档

- /usr/share/doc/kernel-doc-kernel_version/Documentation/serial-console.txt - 此文件（由 kernel-doc 软件包提供）包含有关串行控制台的信息。在访问内核文档前，您必须以 root 用户身份运行以下命令：

```
~]# yum install kernel-doc
```

- **Red Hat 安装指南** - 安装指南提供了有关 GRUB 2 的基本信息，例如安装、术语、界面和命令。

部分 VIII. 系统备份和恢复

这部分论述了如何使用 Relax-and-Recover(ReaR)灾难恢复和系统迁移实用程序。

第 27 章 RELAX-AND-RECOVER (REAR)

当软件或硬件故障破坏系统时，系统管理员需要三项任务来将其恢复为新硬件环境中的完全正常工作状态：

1. 在新硬件中引导救援系统
2. 复制原始存储布局
3. 恢复用户和系统文件

大多数备份软件仅解决第三个问题。要解决第一个和第二个问题，请使用 Relax-and-Recover(ReaR)，这是灾难恢复和系统迁移实用程序。

备份软件创建备份.通过创建救援系统为备份软件提供补充.在新硬件上启动救援系统可让您发出 rear restore 命令，从而启动恢复过程。在此过程中，ReaR 复制分区布局和文件系统，提示从备份软件创建的备份中恢复用户和系统文件，最后安装启动加载器。默认情况下，由 ReaR 创建的救援系统仅恢复存储布局和启动加载器，而不是实际的用户和系统文件。

本章论述了如何使用 ReaR。

27.1. 基本 REAR 用法

27.1.1. 安装 ReaR

作为 root 运行以下命令安装 rear 软件包：

```
~]# yum install rear
```

27.1.2. 配置 ReaR

rear 在 /etc/rear/local.conf 文件中配置。通过添加以下行来指定救援系统配置：

```
OUTPUT=output format
OUTPUT_URL=output location
```

使用救援系统格式替换输出格式，例如：ISO 磁盘镜像的 ISO 或 USB 用于可引导 USB。

使用将输出位置替换为本地文件系统目录的 file:///mnt/rescue_system/，或 SFTP 目录的 sftp://backup:.0.0/。

> 例 27.1. 配置救援系统格式和位置
>
> 要将 ReaR 配置为将救援系统作为 ISO 镜像输出到 /mnt/rescue_system/ 目录中，请将这些行添加到 /etc/rear/local.conf 文件中：
>
> ```
> OUTPUT=ISO
> OUTPUT_URL=file:///mnt/rescue_system/
> ```

有关所有选项的列表，请参阅 rear(8)man page 的"救援映像配置"一节。

特定于 ISO 的配置

使用 例 27.1 “配置救援系统格式和位置” 中的配置会在两个位置生成两个等同的输出文件：

- /var/lib/rear/output/ - rear 的默认输出位置
- /mnt/rescue_system/HOSTNAME/rear-localhost.iso - 在 OUTPUT_URL 中指定的输出位置

但是，通常只需要一个 ISO 镜像。要使 ReaR 只在用户指定的目录中创建 ISO 镜像，请将这些行添加到 /etc/rear/local.conf 中：

```
OUTPUT=ISO
BACKUP=NETFS
OUTPUT_URL=null
BACKUP_URL="iso:///backup"
ISO_DIR="output location"
```

将输出位置替换为输出的所需位置。

27.1.3. 创建救援系统

以下示例演示了如何创建带有详细输出的救援系统：

```
~]# rear -v mkrescue
Relax-and-Recover 1.17.2 / Git
Using log file: /var/log/rear/rear-rhel7.log
mkdir: created directory '/var/lib/rear/output'
Creating disk layout
Creating root filesystem layout
TIP: To login as root via ssh you need to set up /root/.ssh/authorized_keys or
SSH_ROOT_PASSWORD in your configuration file
Copying files and directories
Copying binaries and libraries
Copying kernel modules
Creating initramfs
Making ISO image
Wrote ISO image: /var/lib/rear/output/rear-rhel7.iso (124M)
Copying resulting files to file location
```

使用 例 27.1 “配置救援系统格式和位置” 中的配置，ReaR 会打印上述输出。最后两行确认救援系统已成功创建并复制到配置的备份位置 /mnt/rescue_system/。因为系统的主机名是 rhel7，备份位置现在包含与救援系统的目录 rhel7/ 以及一个辅助文件：

```
~]# ls -lh /mnt/rescue_system/rhel7/
total 124M
-rw-------. 1 root root 202 Jun 10 15:27 README
-rw-------. 1 root root 166K Jun 10 15:27 rear.log
-rw-------. 1 root root 124M Jun 10 15:27 rear-rhel7.iso
-rw-------. 1 root root 274 Jun 10 15:27 VERSION
```

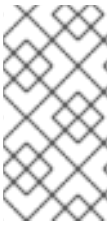
将救援系统传输到外部介质，以便在出现灾难时不会丢失它。

27.1.4. 调度 ReaR

rear 软件包提供的 /etc/cron.d/ rear crontab 文件会在 1:30 AM 每天自动运行 rear mkrescue 命令，以计划定期创建救援系统的 Relax-and-Recover (ReaR)工具。该命令只创建一个救援系统，而不是数据备份。您仍需要自行计划定期备份数据。例如：

- 您可以添加将调度 rear mkbackuponly 命令的另一个 crontab。
- 您还可以更改现有的 crontab 来运行 rear mkbackup 命令，而不是默认的 /usr/sbin/rear checklayout || /usr/sbin/rear mkrescure 命令。
-

如果使用外部备份方法，您可以调度外部备份。详情取决于您在 ReaR 中使用的备份方法。如需了解更多详细信息，请参阅将 ReaR 与备份软件集成。



注意

rear 软件包中提供的 /etc/cron.d/rear crontab 文件已弃用，因为默认情况下，无法执行备份。详情请查看对应的 已弃用的功能 shell 和命令行工具。

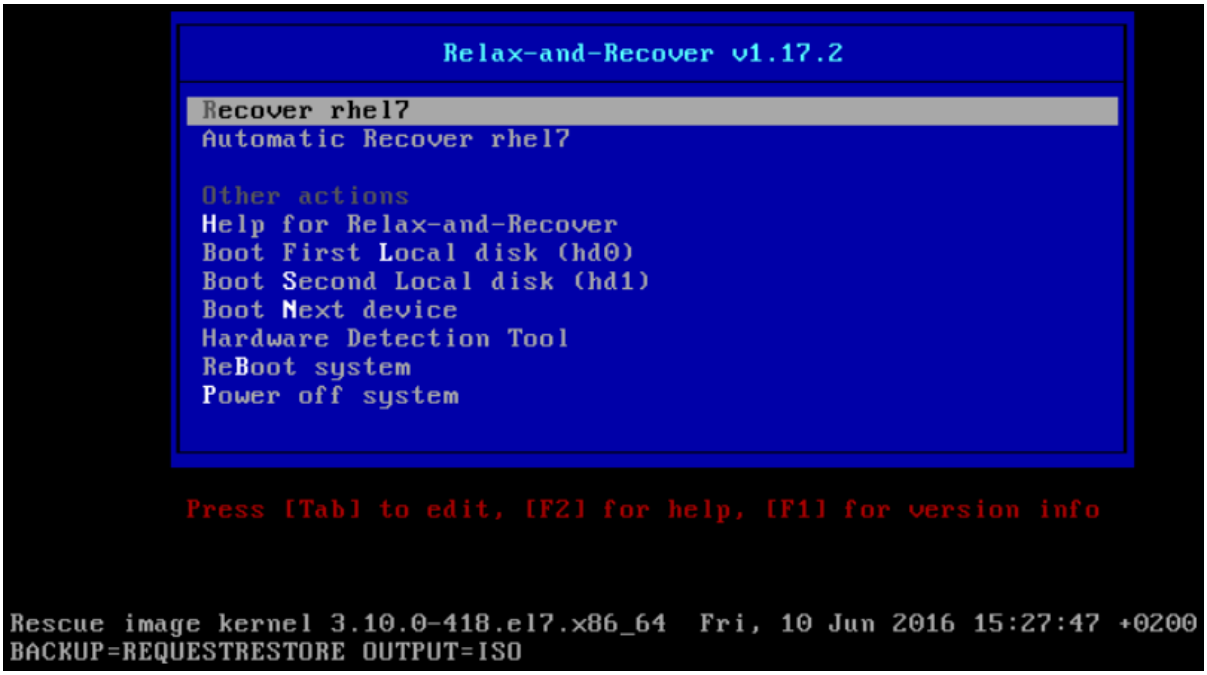
27.1.5. 执行系统救援

执行恢复或迁移：

1. 在新硬件上启动救援系统。例如：将 ISO 镜像刻录到 DVD 并从 DVD 引导。

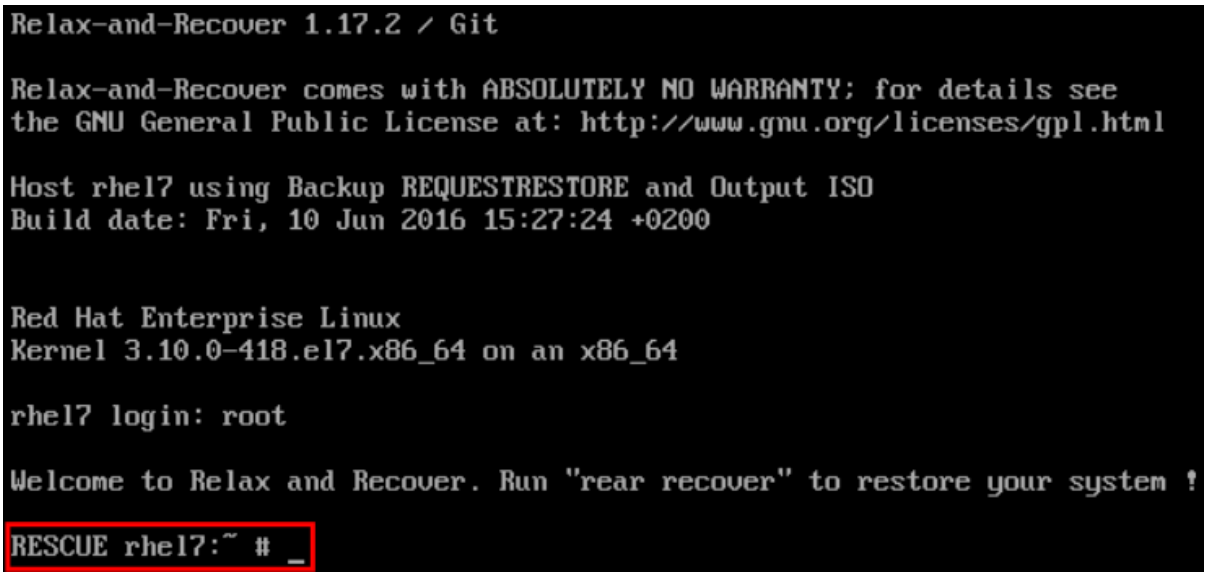
2. 在控制台界面中，选择 "Recover" 选项：

图 27.1. rescue system: 菜单



3. 您将进入提示：

图 27.2. rescue system: prompt

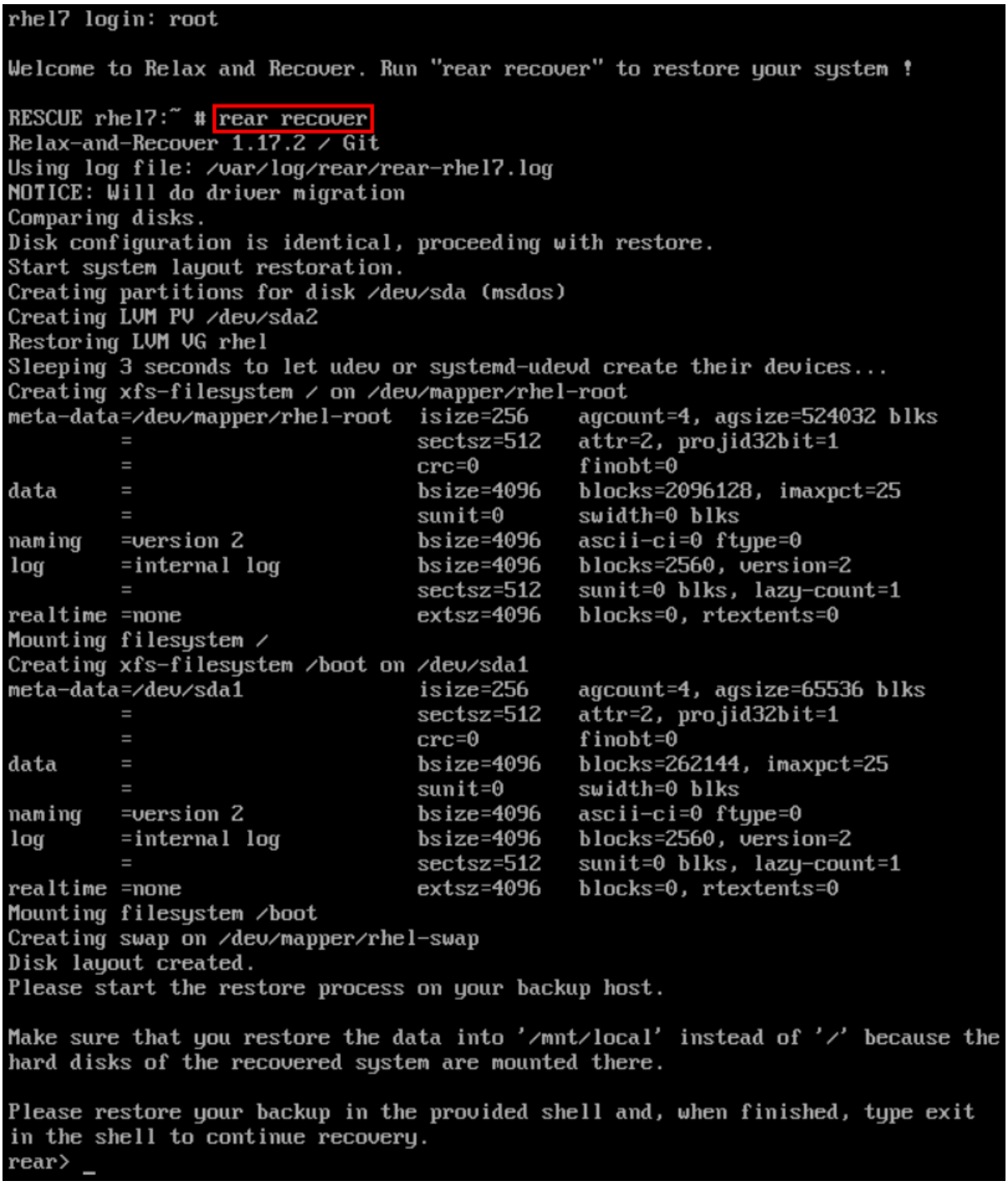


警告

当您在下一步中开始恢复后，它可能无法被撤销，您可能丢失系统物理磁盘上的任何存储内容。

4. 运行 **rear restore** 命令，以执行恢复或迁移。然后，救援系统会重新创建分区布局和文件系统：

图 27.3. 救援系统：运行"rear restore"



5. **将备份中的用户和系统文件恢复到 /mnt/local/ 目录。**

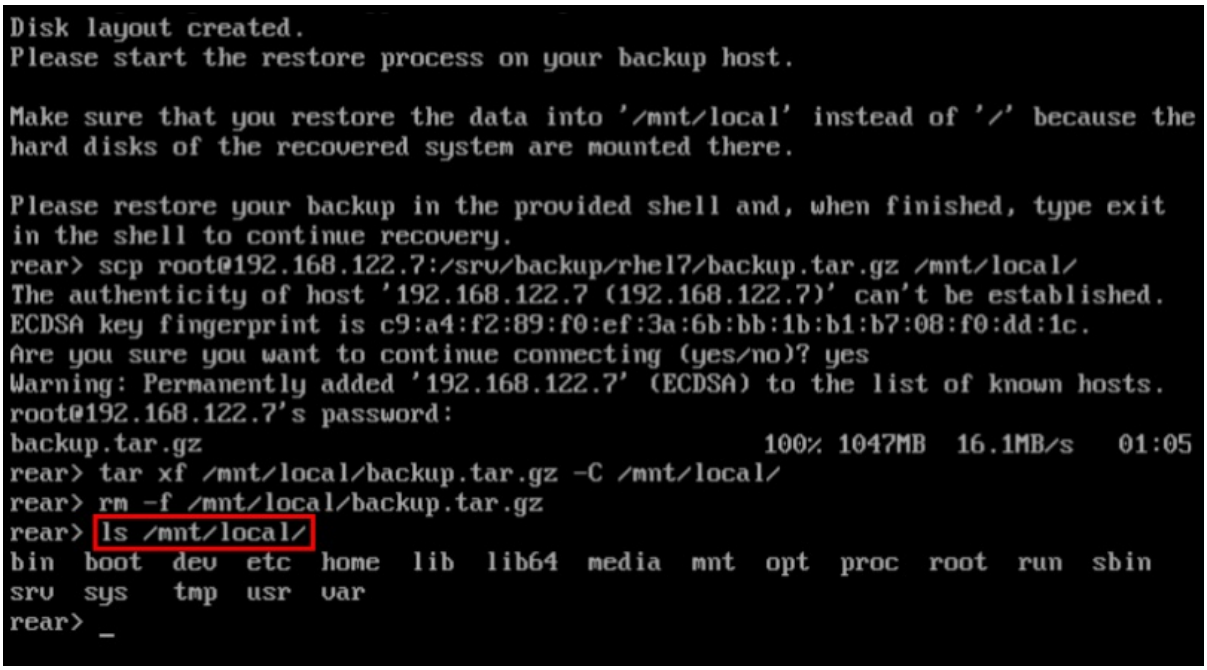
> ***例 27.2. 恢复用户和系统文件***
>
> ***在这个示例中，备份文件是按照 第 27.2.1.1 节 "配置内部备份方法" 中的说明创建的 tar 归档。首先，从存储中复制存档，然后将文件解压缩到 /mnt/local/ 中，然后删除归档：***
>
> ```
> ~]# scp root@192.168.122.7:/srv/backup/rhel7/backup.tar.gz /mnt/local/
> ~]# tar xf /mnt/local/backup.tar.gz -C /mnt/local/
> ~]# rm -f /mnt/local/backup.tar.gz
> ```

> **新存储必须有足够的空间用于归档和提取的文件。**

6. 验证这些文件是否已恢复：

```
~]# ls /mnt/local/
```

图 27.4. 救援系统：从备份中恢复用户和系统文件



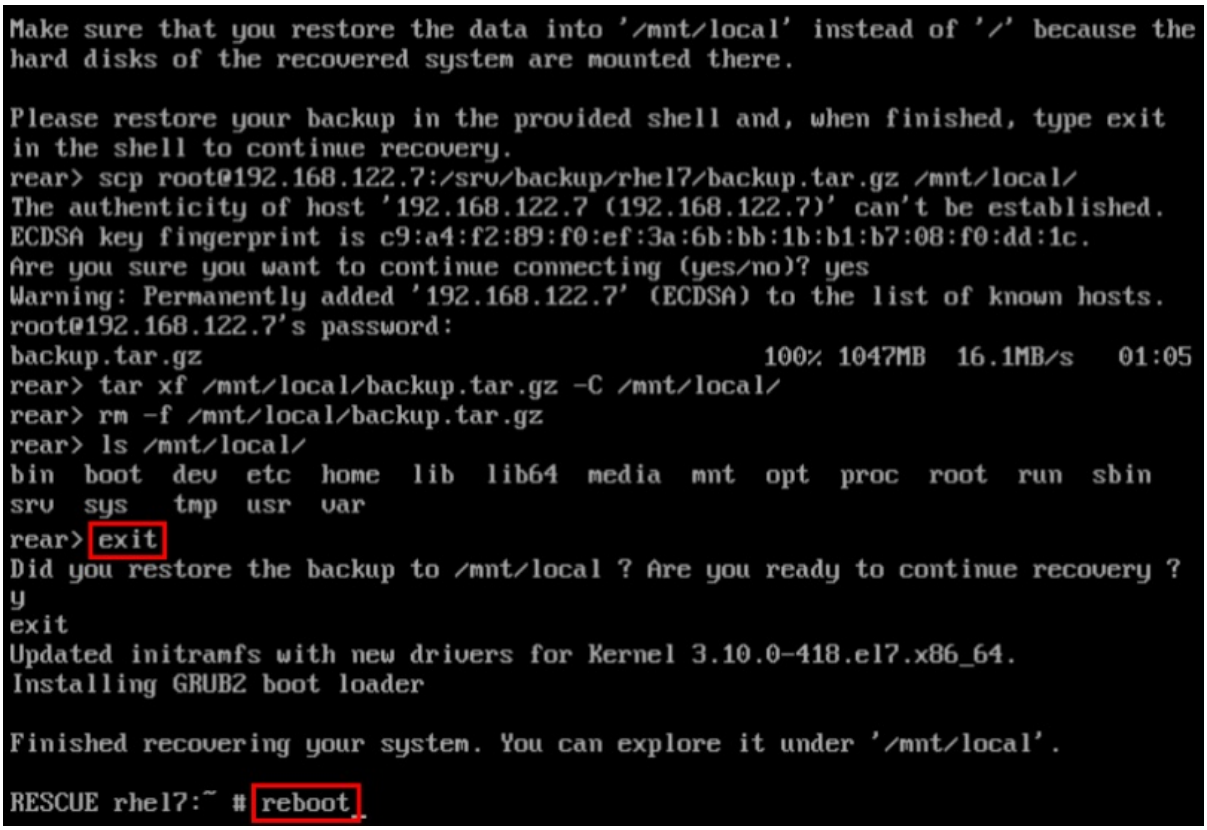
7. 确保 SELinux 在下次引导时重新标记文件：

```
~]# touch /mnt/local/.autorelabel
```

> 否则，您可能无法登录系统，因为 /etc/passwd 文件的 SELinux 上下文可能不正确。

8. 通过输入 exit 来完成恢复。rear 将重新安装启动加载器。之后，重启系统：

图 27.5. 救援系统：完成恢复



重新引导后，SELinux 将重新标记整个文件系统。然后，您将能够登录恢复的系统。

27.2. 将 REAR 与备份软件集成

ReaR 的主要目的是生成救援系统，但它也可与备份软件集成。集成意味着内置、受支持和不支持的备份方法有所不同。

27.2.1. 内置备份方法

Rear 包括内置或内部备份方法。这个方法完全集成了 ReaR，它有以下优点：

- 可以使用一个 rear mkbackup 命令创建救援系统和完整系统备份
- 救援系统自动从备份中恢复文件

因此，ReaR 可以覆盖创建救援系统和完整系统备份的整个过程。

27.2.1.1. 配置内部备份方法

要使 ReaR 使用其内部备份方法，请将这些行添加到 /etc/rear/local.conf 中：

```
BACKUP=NETFS
BACKUP_URL=backup location
```

这些行将 ReaR 配置为使用 tar 命令创建包含完整系统备份的存档。使用 rear(8)man page 的"Backup Software Integration"一节中的其中一个选项替换备份位置。确保备份位置有足够的空间。

例 27.3. 添加 tar 备份

要扩展 第 27.1 节 “基本 ReaR 用法” 中的示例，将 ReaR 配置为同时将 tar 完整系统备份输出到 /srv/backup/ 目录中：

```
OUTPUT=ISO
OUTPUT_URL=file:///mnt/rescue_system/
BACKUP=NETFS
BACKUP_URL=file:///srv/backup/
```

内部备份方法允许进一步配置。

- 要在创建新存档时保留旧的备份存档，请添加以下行：

  ```
  NETFS_KEEP_OLD_BACKUP_COPY=y
  ```

- 默认情况下，ReaR 在每次运行时都会创建一个完整备份。要逐步备份，意味着每次运行时只备份更改的文件，添加以下行：

  ```
  BACKUP_TYPE=incremental
  ```

  这会将 NETFS_KEEP_OLD_BACKUP_COPY 设置为 y。

- 要确保除增量备份外定期执行完整备份，请添加以下行：

  ```
  FULLBACKUPDAY="Day"
  ```

使用"Mon"、"Tue"、"Wed"、"Thu"之一替换"Day"。"Fri", "Sat", "Sun".

- Rear 也可以在 ISO 镜像中包含救援系统和备份。要做到这一点，将 BACKUP_URL 指令设置为 iso:///backup/ ：

```
BACKUP_URL=iso:///backup/
```

这是系统备份的最简单方法，因为救援系统不需要用户在恢复期间获取备份。然而，它需要更多存储。此外，单ISO 备份不能是增量的。

例 27.4. 配置单ISO 救援系统和备份

此配置会创建一个救援系统和备份文件作为单个 ISO 镜像，并将其放在 /srv/backup/ 目录中：

```
OUTPUT=ISO
OUTPUT_URL=file:///srv/backup/
BACKUP=NETFS
BACKUP_URL=iso:///backup/
```

注意

在这种情况下，ISO 镜像可能较大。因此，红帽建议您仅创建一个 ISO 镜像，而不是两个 ISO 镜像。详情请查看 “特定于 ISO 的配置”一节。

- 要使用 rsync 而不是 tar，请添加以下行：

```
BACKUP_PROG=rsync
```

请注意，只有使用 tar 时支持增量备份。

27.2.1.2. 使用内部备份方法创建备份

使用 BACKUP=NETFS 设置，ReaR 可以创建救援系统、备份文件或两者。

- 要只创建救援系统，请运行：

```
rear mkrescue
```

- 要只创建备份，请运行：

```
rear mkbackuponly
```

- 要创建救援系统和备份，请运行：

```
rear mkbackup
```

请注意，只有使用 NETFS 方法，才能使用 ReaR 触发备份。Rear 无法触发其他备份方法。

注意

在恢复时，使用 BACKUP=NETFS 设置创建的救援系统需要在执行 重新恢复 前存在备份。因此，救援系统启动后，将备份文件复制到 BACKUP_URL 中指定的目录中，除非使用单个 ISO 镜像。只有之后才运行re ar恢复。

为了避免不必要的重新创建救援系统，您可以使用以下命令检查存储布局是否已自上次救援系统创建以来更改：

```
~]# rear checklayout
~]# echo $?
```

非零状态表示磁盘布局出现更改。如果 ReaR 配置已更改，则返回非零状态。

重要

rear checklayout 命令不会检查救援系统目前是否存在在输出位置，即使不存在也返回 0。因此，它不能保证救援系统可用，仅确保布局自上次创建救援系统以来没有改变。

例 27.5. 使用 rear checklayout

要创建一个救援系统，但前提是布局已改变，请使用这个命令：

```
~]# rear checklayout || rear mkrescue
```

27.2.2. 支持的备份方法

除了 NETFS 内部备份方法外，ReaR 还支持多种外部备份方法。这意味着救援系统会自动从备份中恢复文件，但无法使用 ReaR 触发备份创建。

有关支持的外部备份方法的列表和配置选项，请参阅 rear(8)man page 的"Backup Software Integration"部分。

27.2.3. 不支持的备份方法

使用不支持的备份方法，有两个选项：

1. 救援系统会提示用户手动恢复文件。这是"基本 ReaR 使用"中所述的场景，但备份文件格式除外，其格式可能与 tar 存档不同。

2. rear 执行用户提供的自定义命令。要配置此功能，将 BACKUP 指令设置为 EXTERNAL。然后，使用 EXTERNAL_BACKUP 和 EXTERNAL_ RESTORE 指令指定要在备份和恢复期间运行的命令。另外，还可指定 EXTERNAL_IGNORE_ERRORS 和 EXTERNAL_CHECK 指令。有关示例配置，请参阅 /usr/share/rear/conf/default.conf。

27.2.4. 创建多个备份

使用 2.00 版本时，ReaR 支持创建多个备份。支持此功能的备份方法包括：

- BACKUP=NETFS （内部方法）
- BACKUP=BORG （外部方法）

您可以使用 rear 命令的 -C 选项指定单个备份 。参数是 /etc/rear/ 目录中额外备份配置文件的基名。每个特定备份的方法、目的地和选项都在特定的配置文件中定义，而不是在主配置文件中定义。

执行系统的基本恢复：

系统的基本恢复

1. 创建 ReaR 恢复系统 ISO 镜像以及基本系统文件的备份：

```
~]# rear -C basic_system mkbackup
```

2. 在 /home 目录中备份文件：

```
~]# rear -C home_backup mkbackuponly
```

请注意，指定的配置文件应包含系统基本恢复所需的目录，如 /boot、/root 和 /usr。

在 rear 恢复 shell 中恢复系统

要在 rear 恢复 shell 中恢复系统，请使用以下命令顺序：

```
~]# rear -C basic_system recover
```

```
~]# rear -C home_backup restoreonly
```

第 28 章 选择有效的红帽产品

通过红帽云基础架构或红帽云解决方案套件订阅，您可以访问多个带有补充功能集的红帽产品。

要确定适合您的组织和用例的产品，您可以使用 Cloud Deployment Planner(CDP)。CDP 是一个交互式工具，总结了不同产品版本的特定互操作性和功能兼容性注意事项。

要根据 Red Hat Enterprise Linux 版本比较特定功能的支持性和各种产品的兼容性，请参阅 综合兼容性列表。

第 29 章 红帽客户门户网站 LABS 与系统管理相关

红帽客户门户 Labs 是旨在帮助您提高性能、解决问题、识别安全问题和优化配置的工具。本附录提供与系统管理相关的红帽客户门户概述。所有红帽客户门户 Labs 均可通过客户门户 Labs 获得。

iSCSI Helper

iSCSI 帮助程序通过 Internet 协议(IP)网络提供块级存储，并允许在服务器虚拟化中使用存储池。

使用 iSCSI 帮助程序 生成一个脚本，该脚本可根据您提供的设置为系统准备 iSCSI 目标（服务器）或 iSCSI 启动器（客户端）的角色。

NTP 配置

使用 NTP（网络时间协议）配置来设置 ：

- 运行 NTP 服务的服务器
- 与 NTP 服务器同步的客户端

Samba 配置帮助程序

Samba 配置帮助程序创建一个配置，通过 Samba 提供基本文件和打印机共享：

- 单击 Server 以指定基本服务器设置。
- 单击 Shares 以添加您要共享的目录
- 单击 Server 以分别添加附加的打印机。

VNC 配置器

VNC 配置器设计为在 Red Hat Enterprise Linux 服务器上安装并配置 VNC（虚拟网络计算）服务器。

使用 VNC 配置器 生成经过优化的一体化脚本，在 Red Hat Enterprise Linux 服务器上安装和配置

VNC 服务。

网桥配置

网桥配置旨在为使用 Red Hat Enterprise Linux 5.4 或更高版本的应用程序（如 KVM）配置桥接网络接口。

网络绑定帮助程序

网络绑定帮助器允许管理员使用绑定内核模块和绑定网络接口将多个网络接口控制器绑定到单个频道。

使用 网络绑定帮助程序 启用两个或多个网络接口，以充当一个绑定接口。

LVM RAID 计算器

在指定存储选项后，LVM RAID 计算器决定了在给定 RAID 存储中创建逻辑卷(LVM)的最佳参数。

使用 LVM RAID 计算器 生成在给定 RAID 存储中创建 LVM 的一系列命令。

NFS 帮助程序

NFS 帮助程序简化了配置新的 NFS 服务器或客户端的过程。按照以下步骤指定导出和挂载选项。然后，生成可下载的 NFS 配置脚本。

Load Balancer Configuration Tool

负载平衡器配置工具可在基于 Apache 的负载平衡器和 JBoss/Tomcat 应用服务器之间实现最佳平衡。

使用 负载平衡器配置工具 生成配置文件以及有关如何提高环境性能的建议。

Yum Repository Configuration Helper

Yum 存储库配置帮助程序旨在设置一个简单的 Yum 存储库。

使用 Yum Repository Configuration Helper 设置：

- 本地 Yum 存储库

- 基于 HTTP/FTP 的 Yum 存储库

文件系统布局计算器

文件系统布局计算器在提供描述当前或计划存储的存储选项后，确定创建 ext3、ext4 和 xfs 文件系统的最佳参数。

使用 文件系统布局计算器 生成一条命令，该命令使用指定 RAID 存储上提供的参数创建文件系统。

RHEL Backup and Restore Assistant

RHEL Backup 和 Restore Assistant 提供有关备份和恢复工具以及 Linux 使用情况的常见情况的信息。

描述的工具：

- 转储和恢复 ：备份 ext2、ext3 和 ext4 文件系统。
- tar 和 cpio ：用于归档或恢复文件和文件夹，尤其是备份磁带驱动器时。
- rsync ：用于执行备份操作，并在不同位置间同步文件和目录。
- dd ：通过块独立于相关文件系统或操作系统，将文件从源复制到目标块。

描述的场景：

- 灾难恢复
- 硬件迁移
- 分区表备份

- 重要文件夹备份
- 增量备份
- 差异备份

DNS 帮助程序

DNS 帮助程序为配置不同类型的 DNS 服务器提供帮助。

使用 DNS 帮助程序 生成 bash 脚本，以自动创建和配置 DNS 服务器。

AD Integration Helper(Samba FS - winbind)

AD Integration Helperr 用于将 Samba 文件系统服务器连接到 Active Directory(AD)服务器。

使用 AD Integration Helper 根据用户提供的基本 AD 服务器信息生成脚本。生成脚本将配置 Samba、名称服务切换(NSS)和可插拔身份验证模块(PAM)。

Red Hat Enterprise Linux Upgrade Helper

Red Hat Enterprise Linux Upgrade Helper 旨在帮助您将 Red Hat Enterprise Linux 从 6.5/6.6/6.7/6.6.6.8 升级到 7.x。

Registration Assistant

Registration Assistant 旨在帮助您为 Red Hat Enterprise Linux 环境选择最合适的注册选项。

救援模式 Assistant

Rescue Mode Assistant 旨在帮助您解决 Red Hat Enterprise Linux 的救援模式中的以下问题：

- 重置 root 密码
- 生成 SOS 报告

- 执行文件系统检查(fsck)
- 重新安装 GRUB
- 重新构建 Initial Ramdisk 镜像
- 缩小根文件系统的大小

Kernel Oops Analyzer

Kernel Oops 分析器旨在帮助您解决内核崩溃。

使用 Kernel Oops 分析器输入包含一个或多个内核oops 消息的文本或文件，并找到适合您的情况的解决方案。

kdump Helper

Kdump Helper 旨在设置 Kdump 机制。

使用 Kdump Helper 生成一个脚本，以设置 Kdump 将内存中的数据转储到名为 vmcore 的转储文件中。

SCSI 解码器

SCSI 解码器 旨在解码 /log/* 文件或日志文件片段中的 SCSI 错误消息，因为这些错误消息可能很难被用户理解。

使用 SCSI 解码器 单独诊断每个 SCSI 错误消息，并获得有效解决问题的解决方案。

Red Hat Memory Analyzer

红帽内存分析器根据 SAR 实用程序获得的信息视觉化您的系统的内存使用情况。

多路径帮助程序

多路径帮助程序在 Red Hat Enterprise Linux 5、6 和 7 中为多路径设备创建最佳配置。

使用 多路径帮助程序 创建高级多路径配置，如自定义别名或设备黑名单。

多路径帮助程序还提供 multipath.conf 文件供检查。完成所需的配置后，下载安装脚本以在您的服务器上运行。

多路径配置可视化工具

多路径配置可视化工具分析 SOS 报告中的文件，并提供了一个视觉化多路径配置图。使用 多路径配置可视化工具 显示：

- 主机组件，包括主机总线适配器(HBA)、本地设备和 iSCSI 设备
- 存储组件
- 服务器和存储之间的光纤或以太网组件
- 到所有上述组件的路径

您可以上传以 .xz、.gz 或 .bz2 格式压缩的 SOS 报告，或者在您选择作为客户端分析来源的目录中提取 SOS 报告。

Red Hat I/O usage Visualizer

红帽 I/O 使用情况可视化工具显示 SAR 实用程序所捕获的 I/O 设备使用情况统计数据的视觉化。

存储/LVM 配置查看器

Storage / LVM 配置查看器分析 SOS 报告中所含的文件，并创建一个图来视觉化存储/LVM 配置。

第 30 章 修订历史记录

0.14-26

Mon Aug 05 2019, Marie Doleželová (mdolezel@redhat.com)

- 为 7.7 GA 发布准备文档.

0.14-23

Mon Aug 13 2018, Marie Doleželová (mdolezel@redhat.com)

- 为 7.6 Beta 版出版物准备文档.

0.14-19

Tue Mar 20 2018, Marie Doleželová (mdolezel@redhat.com)

- 为 7.5 GA 发布准备文档.

0.14-17

Tue Dec 5 2017, Marie Doleželová (mdolezel@redhat.com)

- 更新了 Samba 部分.添加有关使用 TLS 配置 RELP 的部分。更新了有关从 GRUB Legacy 升级到 GRUB 2 的章节。

0.14-16

Mon Aug 8 2017, Marie Doleželová (mdolezel@redhat.com)

- 本指南中添加了有关为自定义单元文件排序和依赖项排序和依赖项的文章链接，并添加链接到"创建自定义单元文件"一章。

0.14-14

Thu Jul 27 2017, Marie Doleželová (mdolezel@redhat.com)

- *发布 7.4 GA 的文件版本.*

0.14-8

2016 年 11 月 3 日， Thim Svistunov(msvistun@redhat.com)

- *7.3 GA 发布版本.*

0.14-7

2016 年 6 月 20 日， Thim Svistunov(msvistun@redhat.com)

- *添加了 Relax-and-Recover(ReaR) ；略有改进.*

0.14-6

2016 年 3 月 10 日， Thim Svistunov(msvistun@redhat.com)

- *次要修复和更新.*

0.14-5

Thu Jan 21 2016, Lenka Špačková (lspackova@redhat.com)

- *次要更新.*

0.14-3

2015 年 11 月 11 日， Jana Heves(jheves@redhat.com)

- *7.2 GA 版本.*

0.14-1

2015 年 11 月 9 日， Jana Heves(jheves@redhat.com)

- 小修复，添加至 RH 培训课程的链接.

0.14-0.3

Fri Apr 3 2015 年，StephenWadeley(swadeley@redhat.com)

- 添加了 注册系统管理订阅、使用红帽支持工具访问支持、更新 查看和管理日志文件.

0.13-2

2015 年 2 月 24 日，Stephen Wadeley(swadeley@redhat.com)

- 7.1 GA 版本.

0.12-0.6

2014 年 11 月 18 日，Stephen Wadeley(swadeley@redhat.com)

- 改进了 TigerVNC.

0.12-0.4

2014 年 11 月 10 日，StephenWadeley(swadeley@redhat.com)

- 改进了 Yum、使用 systemd、OpenLDAP 管理服务、查看和管理日志文件、OProfile 和使用 GRUB 2 引导加载器。

0.12-0

2014 年 8 月 19 日，Stephen Wadeley(swadeley@redhat.com)

- 红帽企业 Linux 7.0 GA 版本系统管理员指南.

30.1. 致谢

该文本的某些部分首先出现在《红帽企业 Linux 6 部署指南》 中，版权所有 © 2010-2018 Red Hat, Inc.，地址为 https://access.redhat.com/documentation/zh-CN/Red_Hat_Enterprise_Linux/6/html/Deployment_Guide/index.html。

第 21.7 节 “使用 Net-SNMP 监控性能” 基于 Michael Solberg 撰写的文章。