



Red Hat Enterprise Linux 9

配置基本系统设置

设置系统的基本功能，并自定义您的系统环境

Red Hat Enterprise Linux 9 配置基本系统设置

设置系统的基本功能，并自定义您的系统环境

Legal Notice

Copyright © 2025 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

执行基本的系统管理任务、配置环境设置、注册您的系统以及配置网络访问和系统安全。管理用户、组和文件权限。使用系统角色来管理多个 RHEL 系统上的系统配置接口。使用 systemd 进行有效的服务管理。使用 chrony 配置网络时间协议(NTP)。使用 ReaR 备份和恢复您的系统。

Table of Contents

对红帽文档提供反馈	4
第 1 章 配置和管理基本网络访问	5
1.1. 在图形安装模式中配置网络和主机名	5
1.2. 使用 NMCLI 配置以太网连接	6
1.3. 使用 NMTUI 配置以太网连接	8
1.4. 使用 NETWORK RHEL 系统角色和接口名称，配置具有动态 IP 地址的以太网连接	12
1.5. 其他资源	14
第 2 章 注册系统并管理订阅	15
2.1. 使用命令行注册系统	15
2.2. 使用 WEB 控制台注册系统	16
2.3. 在 GNOME 桌面环境中注册系统	16
第 3 章 访问红帽支持	18
3.1. 使用 SOSREPORT 工具收集有关系统的 DAIGNOSTIC 信息，将其附加到支持问题单中	18
第 4 章 更改基本环境设置	19
4.1. 配置日期和时间	19
4.2. 使用 WEB 控制台配置时间设置	20
4.3. 配置系统区域设置	21
4.4. 配置键盘布局	21
4.5. 在文本控制台模式下更改字体大小	22
第 5 章 使用 OPENSSSH 的两个系统间使用安全通讯	24
5.1. 生成 SSH 密钥对	24
5.2. 将基于密钥的身份验证设置为 OPENSSSH 服务器的唯一方法	25
5.3. 使用 SSH-AGENT 缓存 SSH 凭证	26
5.4. 通过保存在智能卡中的 SSH 密钥进行身份验证	27
5.5. 其他资源	28
第 6 章 使用日志文件对问题进行故障排除	29
6.1. 处理 SYSLOG 信息的服务	29
6.2. 存储 SYSLOG 信息的日志文件	29
6.3. 使用命令行查看日志	29
6.4. 查看 WEB 控制台中的日志	30
6.5. 其他资源	35
第 7 章 管理用户和组	36
7.1. 管理用户和组群帐户简介	36
7.2. 管理用户帐户入门	37
7.3. 从命令行管理用户	38
7.4. 在 WEB 控制台中管理用户帐户	41
7.5. 使用命令行编辑用户组	43
7.6. 更改和重置根密码	46
第 8 章 管理 SUDO 访问	49
8.1. SUDOERS 中的用户授权	49
8.2. 添加 SUDO 规则以允许组的成员以 ROOT 用户身份执行命令	50
8.3. 使非特权用户运行某些命令	51
8.4. 使用 RHEL 系统角色应用自定义 SUDOERS 配置	53
第 9 章 管理文件系统权限	55
9.1. 管理文件权限	55

9.2. 管理访问控制列表	61
9.3. 管理 UMASK	62
第 10 章 管理 SYSTEMD	67
10.1. SYSTEMD 单元文件位置	67
10.2. 使用 SYSTEMCTL 管理系统服务	68
10.3. 引导至目标系统状态	74
10.4. 关闭、挂起和休眠系统	77
第 11 章 配置时间同步	85
11.1. CHRONY 套件介绍	85
11.2. 使用 CHRONYC 来控制 CHRONYD	86
11.3. 使用 CHRONY	87
11.4. 带有 HW 时间戳的 CHRONY	95
11.5. CHRONY 中的网络时间安全概述(NTS)	99
第 12 章 恢复系统	104
12.1. 设置 REAR 并手动创建备份	104
12.2. 在 64 位 IBM Z 构架上使用 REAR 救援镜像	106
12.3. REAR 排除	108

对红帽文档提供反馈

我们感谢您对我们文档的反馈。让我们了解如何改进它。

通过 Jira 提交反馈（需要帐户）

1. 登录到 [Jira](#) 网站。
2. 在顶部导航栏中点 **Create**
3. 在 **Summary** 字段中输入描述性标题。
4. 在 **Description** 字段中输入您的改进建议。包括文档相关部分的链接。
5. 点对话框底部的 **Create**。

第 1 章 配置和管理基本网络访问

NetworkManager 为主机上安装的每个以太网适配器创建一个连接配置文件。默认情况下，此配置文件将 DHCP 用于 IPv4 和 IPv6 连接。修改此自动创建的配置文件，或在以下情况下添加新配置文件：

- 网络需要自定义设置，如静态 IP 地址配置。
- 您需要多个配置文件，因为主机在不同的网络中漫游。

Red Hat Enterprise Linux 为管理员提供不同的选项来配置以太网连接。例如：

- 在命令行中使用 nmcli 配置连接。
- 使用 nmtui 在基于文本的用户界面中配置连接。
- 使用 GNOME Settings 菜单或 nm-connection-editor 应用程序在图形界面中配置连接。
- 使用 nmstatectl 通过 Nmstate API 配置连接。
- 使用 RHEL 系统角色在一个或多个主机上自动配置连接。

1.1. 在图形安装模式中配置网络和主机名

按照以下步骤来配置您的网络和主机名。

步骤

1. 在 **Installation Summary** 窗口中点击 **Network and Host Name**。
2. 在左侧窗格的列表选择一个接口。详情显示在右侧方框中。
3. 使用 **ON/OFF** 开关来启用或禁用所选接口。
您不能手动添加或删除接口。
4. 点击 **+** 添加虚拟网络接口，可以是：Team（已弃用）、Bonded、Bridge 或 VLAN。
5. 点 **-** 删除虚拟接口。
6. 点 **Configure** 更改设置，如 IP 地址、DNS 服务器或者现有接口的路由配置（虚拟和物理）。
7. 在 **Host Name** 字段中输入您系统的主机名。
主机名可以是完全限定域名(FQDN)，格式为 **hostname.domainname**，也可以是没有域的短主机名。许多网络具有动态主机配置协议(DHCP)服务，该服务自动为连接的系统提供域名。要允许 DHCP 服务为这个系统分配域名，请只指定短主机名。

主机名只能包含字母数字字符和 - 或 .。主机名应等于或小于 64 个字符。主机名不能以 - 和 . 开头或结尾要与 DNS 兼容，FQDN 的每个部分都应等于或小于 63 个字符，FQDN 总计长度，包括点，不应超过 255 个字符。

值 **localhost** 意味着没有为目标系统配置特定的静态主机名，安装的系统的实际主机名会在处理网络配置的过程中配置，例如，使用 DHCP 或 DNS 通过 NetworkManager 配置。

使用静态 IP 和主机名配置时，它根据计划的系统用例来决定使用短名称还是 FQDN。Red Hat Identity Management 在置备过程中配置 FQDN，但有些第三方软件产品可能需要短名称。在任何一种情况下，要确保在所有情况下两种形式都可用，请在 **/etc/hosts** 中为主机添加一个条目，

格式为 **IP FQDN 短别名**。

- 单击 **Apply**，将主机名应用到安装程序环境。
- 或者，在 **Network and Hostname** 窗口中，您可以选择 **Wireless** 选项。单击右侧窗格中的 **Select network** 以选择您的 wifi 连接，如需要请输入密码，然后单击 **Done**。

其他资源

- [自动安装 RHEL](#)
- 有关网络设备命名标准的更多信息，请参阅 [配置和管理网络](#)。

1.2. 使用 nmcli 配置以太网连接

如果您通过以太网将主机连接到网络，您可以使用 **nmcli** 工具在命令行上管理连接的设置。

先决条件

- 服务器配置中存在物理或虚拟以太网网络接口控制器(NIC)。

流程

1. 列出 NetworkManager 连接配置文件：

```
# nmcli connection show
NAME                UUID                                TYPE    DEVICE
Wired connection 1  a5eb6490-cc20-3668-81f8-0314a27f3f75  ethernet  enp1s0
```

默认情况下，NetworkManager 为主机中的每个 NIC 创建一个配置文件。如果您计划仅将这个 NIC 连接到特定的网络，请调整自动创建的配置文件。如果您计划将这个 NIC 连接到具有不同设置的网络，请为每个网络创建单独的配置文件。

2. 如果要创建额外的连接配置文件，请输入：

```
# nmcli connection add con-name <connection-name> ifname <device-name> type ethernet
```

跳过此步骤以修改现有配置文件。

3. 可选：重命名连接配置文件：

```
# nmcli connection modify "Wired connection 1" connection.id "Internal-LAN"
```

在具有多个配置文件的主机上，有意义的名称可以更容易识别配置文件的用途。

4. 显示连接配置文件的当前设置：

```
# nmcli connection show Internal-LAN
...
connection.interface-name:  enp1s0
connection.autoconnect:    yes
```

```

ipv4.method:      auto
ipv6.method:      auto
...

```

5. 配置 IPv4 设置：

- 要使用 DHCP，请输入：

```
# nmcli connection modify Internal-LAN ipv4.method auto
```

如果 **ipv4.method** 已设置为 **auto**（默认），请跳过这一步。

- 要设置静态 IPv4 地址、网络掩码、默认网关、DNS 服务器和搜索域，请输入：

```
# nmcli connection modify Internal-LAN ipv4.method manual ipv4.addresses
192.0.2.1/24 ipv4.gateway 192.0.2.254 ipv4.dns 192.0.2.200 ipv4.dns-search
example.com
```

6. 配置 IPv6 设置：

- 要使用无状态地址自动配置(SLAAC)，请输入：

```
# nmcli connection modify Internal-LAN ipv6.method auto
```

如果 **ipv6.method** 已设置为 **auto**（默认），请跳过这一步。

- 要设置静态 IPv6 地址、网络掩码、默认网关、DNS 服务器和搜索域，请输入：

```
# nmcli connection modify Internal-LAN ipv6.method manual ipv6.addresses
2001:db8:1::fffe/64 ipv6.gateway 2001:db8:1::fffe ipv6.dns 2001:db8:1::ffbb
ipv6.dns-search example.com
```

7. 要在配置文件中自定义其他设置，请使用以下命令：

```
# nmcli connection modify <connection-name> <setting> <value>
```

将值用空格或分号括在引号中。

8. 激活配置文件：

```
# nmcli connection up Internal-LAN
```

验证

1. 显示 NIC 的 IP 设置：

```
# ip address show enp1s0
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
    link/ether 52:54:00:17:b8:b6 brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.1/24 brd 192.0.2.255 scope global noprefixroute enp1s0
        valid_lft forever preferred_lft forever
    inet6 2001:db8:1::fffe/64 scope global noprefixroute
        valid_lft forever preferred_lft forever
```

2. 显示 IPv4 默认网关：

```
# ip route show default
default via 192.0.2.254 dev enp1s0 proto static metric 102
```

3. 显示 IPv6 默认网关：

```
# ip -6 route show default
default via 2001:db8:1::fffe dev enp1s0 proto static metric 102 pref medium
```

4. 显示 DNS 设置：

```
# cat /etc/resolv.conf
search example.com
nameserver 192.0.2.200
nameserver 2001:db8:1::ffbb
```

如果多个连接配置文件同时处于活跃状态，则 **nameserver** 条目的顺序取决于这些配置文件中的 DNS 优先级值和连接类型。

5. 使用 **ping** 工具验证这个主机是否可以向其他主机发送数据包：

```
# ping <host-name-or-IP-address>
```

故障排除

- 验证网线是否插入到主机和交换机。
- 检查链路失败是否只存在于此主机上，或者连接到同一交换机的其它主机上。
- 验证网络电缆和网络接口是否如预期工作。执行硬件诊断步骤并替换缺陷电缆和网络接口卡。
- 如果磁盘中的配置与设备中的配置不匹配，则启动或重启 NetworkManager 会创建一个代表该设备的配置的内存连接。有关详情以及如何避免此问题，请参阅红帽知识库解决方案 [NetworkManager 在重启 NetworkManager 服务后复制连接](#)。

其他资源

- 您系统上的 **nm-settings (5)** 手册页

1.3. 使用 NMTUI 配置以太网连接

如果通过以太网将主机连接到网络，您可以使用 **nmtui** 工具在基于文本的用户界面中管理连接的设置。使用 **nmtui** 创建新配置文件，并在没有图形界面的主机上更新现有配置文件。



注意

在 **nmtui** 中：

- 使用光标键导航。
- 选择按钮并按 **Enter** 键。
- 使用空格选择和清除 **复选框**。
- 要返回上一个屏幕，请使用 **ESC**。

先决条件

- 服务器配置中存在物理或虚拟以太网网络接口控制器(NIC)。

流程

1. 如果您不知道连接中使用的网络设备名称，显示可用的设备：

```
# nmcli device status
DEVICE  TYPE    STATE      CONNECTION
enp1s0  ethernet unavailable --
...
```

2. 启动 **nmtui**：

```
# nmtui
```

3. 选择 **Edit a connection**，然后按 **Enter**。
4. 选择是否添加一个新连接配置文件或修改现有连接配置文件：
 - 要创建一个新配置文件：
 - i. 按 **Add**。
 - ii. 从网络类型列表中选择 **Ethernet**，然后按 **Enter** 键。
 - 要修改现有的配置文件，请从列表中选择配置文件，然后按 **Enter**。
5. 可选：更新连接配置文件的名称。
在具有多个配置文件的主机上，有意义的名称可以更容易识别配置文件的用途。
6. 如果创建新连接配置文件，请在 **Device** 字段中输入网络设备名称。
7. 根据您的环境，相应地在 **IPv4 configuration** 和 **IPv6 configuration** 区中配置 IP 地址。为此，请按这些区域旁边的按钮，并选择：
 - **Disabled**，如果此连接不需要 IP 地址。
 - **Automatic**，如果 DHCP 服务器为这个 NIC 动态分配了一个 IP 地址。
 - **Manual**，如果网络需要静态 IP 地址设置。在这种情况下，您必须填写更多字段：
 - i. 在您要配置的协议旁边按 **Show** 以显示其他字段。

- ii. 按 **Addresses** 旁边的 **Add**，并以无类别域间路由(CIDR)格式输入 IP 地址和子网掩码。
如果没有指定子网掩码，NetworkManager 会为 IPv4 地址设置 **/32** 子网掩码，并为 IPv6 地址设置 **/64**。
- iii. 输入默认网关的地址。
- iv. 按 **DNS servers** 旁边的 **Add**，并输入 DNS 服务器地址。
- v. 按 **Search domains** 旁边的 **Add**，并输入 DNS 搜索域。

图 1.1. 使用静态 IP 地址设置的以太网连接示例

Edit Connection

Profile name: Example-Connection
Device: enp7s0

= ETHERNET <Show>

IPv4 CONFIGURATION <Manual> <Hide>

Addresses: 192.0.2.1/24 <Remove>
<Add...>

Gateway: 192.0.2.254

DNS servers: 192.0.2.200 <Remove>
<Add...>

Search domains: example.com <Remove>
<Add...>

Routing (No custom routes) <Edit...>

☐ Never use this network for default route

☐ Ignore automatically obtained routes

☐ Ignore automatically obtained DNS parameters

☐ Require IPv4 addressing for this connection

IPv6 CONFIGURATION <Manual> <Hide>

Addresses: 2001:db8:1::1/64 <Remove>
<Add...>

Gateway: 2001:db8:1::fffe

DNS servers: 2001:db8:1::ffbb <Remove>
<Add...>

Search domains: example.com <Remove>
<Add...>

Routing (No custom routes) <Edit...>

☐ Never use this network for default route

☐ Ignore automatically obtained routes

☐ Ignore automatically obtained DNS parameters

☐ Require IPv6 addressing for this connection

☒ Automatically connect

☒ Available to all users

<Cancel> **<OK>**

8. 按 **OK** 创建并自动激活新连接。
9. 按 **Back** 返回到主菜单。

10. 选择 **Quit**，然后按 **Enter** 关闭 **nmtui** 应用程序。

验证

1. 显示 NIC 的 IP 设置：

```
# ip address show enp1s0
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
    link/ether 52:54:00:17:b8:b6 brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.1/24 brd 192.0.2.255 scope global noprefixroute enp1s0
        valid_lft forever preferred_lft forever
    inet6 2001:db8:1::ffe/64 scope global noprefixroute
        valid_lft forever preferred_lft forever
```

2. 显示 IPv4 默认网关：

```
# ip route show default
default via 192.0.2.254 dev enp1s0 proto static metric 102
```

3. 显示 IPv6 默认网关：

```
# ip -6 route show default
default via 2001:db8:1::ffe dev enp1s0 proto static metric 102 pref medium
```

4. 显示 DNS 设置：

```
# cat /etc/resolv.conf
search example.com
nameserver 192.0.2.200
nameserver 2001:db8:1::ffbb
```

如果多个连接配置文件同时处于活跃状态，则 **nameserver** 条目的顺序取决于这些配置文件中的 DNS 优先级值和连接类型。

5. 使用 **ping** 工具验证这个主机是否可以向其他主机发送数据包：

```
# ping <host-name-or-IP-address>
```

故障排除

- 验证网线是否插入到主机和交换机。
- 检查链路失败是否只存在于此主机上，或者连接到同一交换机的其它主机上。
- 验证网络电缆和网络接口是否如预期工作。执行硬件诊断步骤并替换缺陷电缆和网络接口卡。
- 如果磁盘中的配置与设备中的配置不匹配，则启动或重启 **NetworkManager** 会创建一个代表该设备的配置的内存连接。有关详情以及如何避免此问题，请参阅红帽知识库解决方案 [NetworkManager 在重启 NetworkManager 服务后复制连接](#)。

其他资源

- [配置 NetworkManager 以避免使用特定配置集提供默认网关](#)
- [配置 DNS 服务器顺序](#)

1.4. 使用 NETWORK RHEL 系统角色和接口名称，配置具有动态 IP 地址的以太网连接

要将 Red Hat Enterprise Linux 主机连接到以太网网络，请为网络设备创建一个 NetworkManager 连接配置文件。通过使用 Ansible 和 **network** RHEL 系统角色，您可以自动化此过程，并在 playbook 中定义的主机上远程配置连接配置文件。

您可以使用 **network** RHEL 系统角色配置以太网连接，该连接从 DHCP 服务器检索其 IP 地址、网关和 DNS 设置，以及 IPv6 无状态地址自动配置(SLAAC)。使用此角色，您可以将连接配置文件分配给指定的接口名称。

先决条件

- [您已准备好控制节点和受管节点](#)
- 以可在受管主机上运行 playbook 的用户登录到控制节点。
- 用于连接到受管节点的帐户具有 **sudo** 权限。
- 物理或者虚拟以太网设备在服务器配置中存在。
- 网络中有 DHCP 服务器和 SLAAC。
- 受管节点使用 NetworkManager 服务来配置网络。

流程

1. 创建一个包含以下内容的 playbook 文件，如 **~/playbook.yml**：

```
---
- name: Configure the network
  hosts: managed-node-01.example.com
  tasks:
    - name: Ethernet connection profile with dynamic IP address settings
      ansible.builtin.include_role:
        name: redhat.rhel_system_roles.network
      vars:
        network_connections:
          - name: enp1s0
            interface_name: enp1s0
            type: ethernet
            autoconnect: yes
            ip:
              dhcp4: yes
              auto6: yes
            state: up
```

示例 playbook 中指定的设置包括以下内容：

dhcp4: yes

启用来自 DHCP、PPP 或类似服务的自动 IPv4 地址分配。

auto6: yes

启用 IPv6 自动配置。默认情况下，NetworkManager 使用路由器通告。如果路由器宣布 **managed** 标记，则 NetworkManager 会从 DHCPv6 服务器请求 IPv6 地址和前缀。

有关 playbook 中使用的所有变量的详情，请查看控制节点上的 `/usr/share/ansible/roles/rhel-system-roles.network/README.md` 文件。

2. 验证 playbook 语法：

```
$ ansible-playbook --syntax-check ~/playbook.yml
```

请注意，这个命令只验证语法，不会防止错误但有效的配置。

3. 运行 playbook：

```
$ ansible-playbook ~/playbook.yml
```

验证

- 查询受管节点的 Ansible 事实，并验证接口是否收到 IP 地址和 DNS 设置：

```
# ansible managed-node-01.example.com -m ansible.builtin.setup
...
  "ansible_default_ipv4": {
    "address": "192.0.2.1",
    "alias": "enp1s0",
    "broadcast": "192.0.2.255",
    "gateway": "192.0.2.254",
    "interface": "enp1s0",
    "macaddress": "52:54:00:17:b8:b6",
    "mtu": 1500,
    "netmask": "255.255.255.0",
    "network": "192.0.2.0",
    "prefix": "24",
    "type": "ether"
  },
  "ansible_default_ipv6": {
    "address": "2001:db8:1::1",
    "gateway": "2001:db8:1::fffe",
    "interface": "enp1s0",
    "macaddress": "52:54:00:17:b8:b6",
    "mtu": 1500,
    "prefix": "64",
    "scope": "global",
    "type": "ether"
  },
  ...
  "ansible_dns": {
    "nameservers": [
      "192.0.2.1",
      "2001:db8:1::ffbb"
    ],
    "search": [
```

```
    "example.com"
  ],
  ...
```

其他资源

- `/usr/share/ansible/roles/rhel-system-roles.network/README.md` 文件
- `/usr/share/doc/rhel-system-roles/network/` 目录

1.5. 其他资源

- [配置和管理网络](#)

第 2 章 注册系统并管理订阅

订阅覆盖了在 Red Hat Enterprise Linux 中安装的产品，包括操作系统本身。如果您还没有注册系统，则无法访问 RHEL 软件仓库。您无法安装软件更新，如安全、错误修复。即使您有自助支持订阅，它也授予对知识库的访问权限，而更多资源在缺少订阅时仍不可用。通过购买订阅并使用 Red Hat Content Delivery Network (CDN)，您可以跟踪：

- 注册的系统
- 在注册的系统安装产品
- 附加到安装产品的订阅

2.1. 使用命令行注册系统

订阅覆盖了在 Red Hat Enterprise Linux 中安装的产品，包括操作系统本身。如果您还没有注册系统，则无法访问 RHEL 软件仓库。您无法安装软件更新，如安全、错误修复。即使您有自助支持订阅，它也授予对知识库的访问权限，而更多资源在缺少订阅时仍不可用。您需要注册系统来激活和管理您的红帽帐户的 Red Hat Enterprise Linux 订阅。



注意

要将系统注册到 Red Hat Insights，您可以使用 **rhc connect** 工具。详情请参阅 [设置远程主机配置](#)。

先决条件

- 您有有效的 Red Hat Enterprise Linux 系统订阅。

流程

- 注册并订阅系统：

```
# subscription-manager register
Registering to:          subscription.rhsm.redhat.com:443/subscription
Username: <example_username>
Password: <example_password>
The system has been registered with ID: 37to907c-ece6-49ea-9174-20b87ajk9ee7
The registered system name is:      client1.example.com
```

该命令会提示您输入红帽客户门户网站帐户的用户名和密码。

如果注册过程失败，您可以使用特定池注册系统。详情请查看以下步骤：

- 确定订阅的池 ID：

```
# subscription-manager list --available --all
```

这个命令会显示您的红帽账户中的所有可用订阅。对于每个订阅，会显示各种相关信息，包括池 ID。

- 使用上一步中决定的池 ID 替换 `<example_pool_id>` 来为您的系统附加适当的订阅：

```
# subscription-manager attach --pool=<example_pool_id>
```

验证

- 验证混合云控制台中的 **Inventory** → **Systems** 下的系统。

其他资源

- [了解红帽订阅管理](#)
- [了解您购买红帽产品的工作流](#)
- [在 Hybrid Cloud 控制台中查看您的订阅清单](#)

2.2. 使用 WEB 控制台注册系统

订阅覆盖了在 Red Hat Enterprise Linux 中安装的产品，包括操作系统本身。如果您还没有注册系统，则无法访问 RHEL 软件仓库。您无法安装软件更新，如安全、错误修复。即使您有自助支持订阅，它也授予对知识库的访问权限，而更多资源在缺少订阅时仍不可用。您可以使用 Red Hat web 控制台中的帐户凭证注册新安装的 Red Hat Enterprise Linux。

先决条件

- 您有一个有效的 RHEL 系统订阅。
- 您已安装了 RHEL 9 web 控制台。
- 您已启用了 cockpit 服务。
- 您的用户帐户被允许登录到 web 控制台。
具体步骤请参阅[安装并启用 Web 控制台](#)。

流程

1. 在浏览器中打开 `https://<ip_address_or_hostname>:9090`，并登录到 web 控制台。
2. 在 **Overview** 页面的 **Health** 字段中，点击 **Not registered** 警告，或者点击主菜单中的 **Subscriptions** 来进入页面。
3. 在 **Overview** 字段中，点 **Register**。
4. 在 **注册系统** 对话框中，选择注册方法。
可选：输入您的机构名称或 ID。如果您的帐户属于红帽客户门户网站中的多个机构，您必须添加机构名称或 ID。要获得机构 ID，请在红帽与您的大客户经理联系。
5. 如果您不想将您的系统连接到 Red Hat Insights，请清除 **Insights** 复选框。
6. 点 **Register**。

验证

- [在混合云控制台中](#) 查看您的订阅详情。

2.3. 在 GNOME 桌面环境中注册系统

订阅覆盖了在 Red Hat Enterprise Linux 中安装的产品，包括操作系统本身。如果您还没有注册系统，则无法访问 RHEL 软件仓库。您无法安装软件更新，如安全性、错误修复。即使您有自助支持订阅，它也授

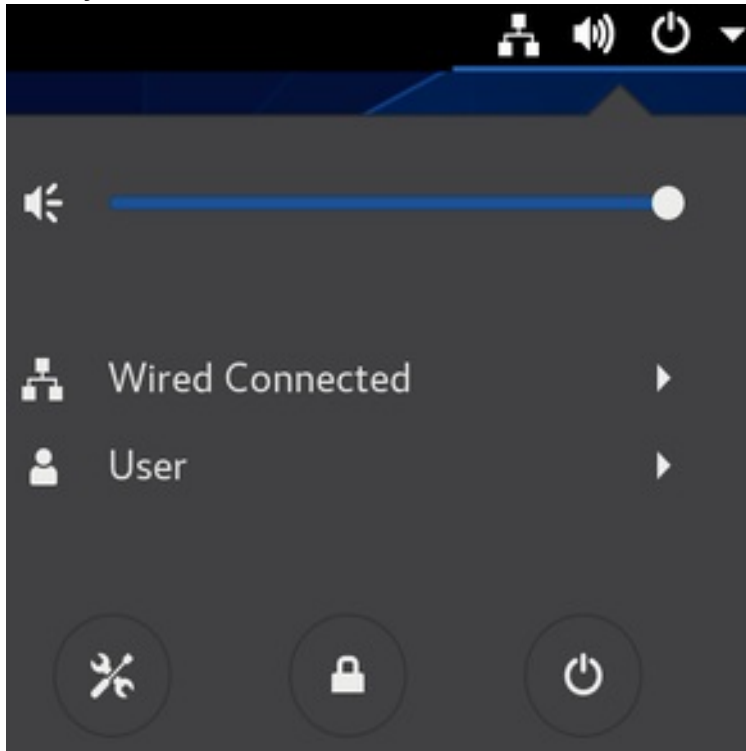
予对知识库的访问权限，而更多资源在缺少订阅时仍不可用。按照以下步骤将系统注册到您的红帽帐户中。

先决条件

- 您已创建了 [红帽帐户](#)。
- 您是一个 root 用户，并登录到 GNOME 桌面环境。详情请参阅在 [Red Hat Subscription Manager 中注册和订阅 RHEL 系统](#)。

流程

1. 打开 **system menu**，该菜单可从右上角访问，然后单击 **Settings**。



2. 转至 **About → Subscription**。
3. 如果要通过 Red Hat Satellite 注册系统：
 - a. 在 **Registration Server** 部分中，选择 **Custom Address**。
 - b. 在 **URL** 字段中输入服务器地址。
4. 在 **Registration Type** 部分中，选择您首选的注册方法。
5. 填写 **Registration Details** 部分。
6. 点 **Register**。

第 3 章 访问红帽支持

如果您需要对问题进行故障排除的帮助，请联系红帽支持。

流程

- 登录到 Red Hat Support 网站并选择以下选项之一：
 - 创建新的支持问题单。
 - 与红帽支持专家启动实时聊天。
 - 通过致电或发送电子邮件与红帽专家联系。

3.1. 使用 SOSREPORT 工具收集有关系统的 DAIGNOSTIC 信息，将其附加到支持问题单中

sosreport 命令从 Red Hat Enterprise Linux 系统收集配置详情、系统信息和诊断信息。

以下小节论述了如何使用 **sosreport** 命令为您的支持问题单生成报告。

先决条件

- 红帽客户门户网站中的有效用户帐户。请参阅 [创建红帽登录帐号](#)。
- RHEL 系统的有效订阅。
- 支持问题单号。

流程

1. 安装 **sos** 软件包：

```
# dnf install sos
```

2. 生成一个报告：

```
# sosreport
```

（可选）将 **--upload** 选项传给命令，以自动上传报告并将其附加到支持问题单中。这需要访问互联网和您的客户门户网站凭证。

3. 可选：手动将报告附加到您的支持问题单中。
如需更多信息，请参阅红帽知识库解决方案 [如何将文件附加到红帽支持问题单？](#)

其他资源

- [什么是 sosreport，以及如何在 Red Hat Enterprise Linux 中创建？](#)（红帽知识库）

第 4 章 更改基本环境设置

配置基本环境设置是安装过程的一部分。以下部分介绍了在稍后修改时的信息。环境的基本配置包括：

- 日期和时间
- 系统区域设置
- 键盘布局
- 语言

4.1. 配置日期和时间

因为以下原因，准确计时非常重要。在 Red Hat Enterprise Linux 中，**NTP** 协议保证计时的准确性，该协议由用户空间运行的守护进程实施。user-space 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 9 及更新的版本使用 **chronyd** 守护进程来实现 **NTP**。**chronyd** 可从 **chrony** 软件包得到。如需更多信息，请参阅 [使用 chrony 套件来配置 NTP](#)。

4.1.1. 手动配置日期、时间和时区设置

要显示当前日期和时间，请使用这些步骤之一。

流程

1. 可选：列出时区：

```
# timedatectl list-timezones

Europe/Berlin
```

2. 设置时区：

```
# timedatectl set-timezone <time_zone>
```

3. 设置日期和时间：

```
# timedatectl set-time <YYYY-mm-dd HH:MM:SS>
```

验证

1. 显示日期、时间和时区：

```
# date
Mon Mar 30 16:02:59 CEST 2020
```

2. 要查看更多详细信息，请使用 **timedatectl** 命令：

```
# timedatectl

Local time: Mon 2020-03-30 16:04:42 CEST
```

```
Universal time: Mon 2020-03-30 14:04:42 UTC
RTC time: Mon 2020-03-30 14:04:41
Time zone: Europe/Prague (CEST, +0200)
System clock synchronized: yes
NTP service: active
RTC in local TZ: no
```

其他资源

- **date (1)** 和 **timedatectl (1)** man page

4.2. 使用 WEB 控制台配置时间设置

您可以在 RHEL web 控制台中设置时区，并将系统时间与网络时间协议(NTP)服务器同步。

先决条件

- 您已安装了 RHEL 9 web 控制台。
- 您已启用了 cockpit 服务。
- 您的用户帐户被允许登录到 web 控制台。
具体步骤请参阅[安装并启用 Web 控制台](#)。

流程

1. 登录到 RHEL 9 web 控制台。
详情请参阅 [登录到 web 控制台](#)。
2. 点**概述**中的当前系统时间。
3. 点 **System time**。
4. 在 **更改系统时间** 对话框中，根据需要更改时区。
5. 在 **Set Time** 下拉菜单中选择以下之一：

手动

如果您需要手动设定时间，而不使用 NTP 服务器，则使用这个选项。

自动使用 NTP 服务器

这是一个默认选项，它会自动与预设置的 NTP 服务器同步。

自动使用特定的 NTP 服务器

只有在您需要将系统与特定 NTP 服务器同步时使用这个选项。指定服务器的 DNS 名称或 IP 地址。

6. 点击 **更改**。

验证

- 检查在 **System** 标签页中显示的系统时间。

其他资源

- 使用 Chrony 套件配置 NTP

4.3. 配置系统区域设置

系统范围的区域设置存储在 `/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在 `/etc/locale.conf` 中配置的 locale 设置，单独程序或个人用户可以单独覆盖它们。

流程

1. 可选：显示当前系统区域设置：

```
# localectl status
System Locale: LANG=en_US.UTF-8
VC Keymap: de-nodeadkeys
X11 Layout: de
X11 Variant: nodeadkeys
```

2. 列出可用系统区域设置：

```
$ localectl list-locales
C.UTF-8
...
en_US.UTF-8
en_ZA.UTF-8
en_ZW.UTF-8
...
```

3. 更新 system 区域设置：

例如：

+

```
# localectl set-locale LANG=en_US.UTF-8
```



注意

GNOME 终端不支持非 UTF8 系统区域设置。如需更多信息，请参阅红帽知识库解决方案 [gnome-terminal 应用程序在系统区域设置设置为非 UTF8 时无法启动](#)。

其他资源

- `man localectl(1)`、`man locale(7)` 和 `man locale.conf(5)`

4.4. 配置键盘布局

键盘布局设置控制文本控制台和图形用户界面中的布局。

流程

1. 要列出可用的键映射：

-

```
$ localectl list-keymaps
ANSI-dvorak
al
al-plisi
amiga-de
amiga-us
...
```

2. 显示 keymaps 设置的当前状态：

```
$ localectl status
...
VC Keymap: us
...
```

3. 要设置或更改默认系统 keymap：例如：

```
# localectl set-keymap us
```

其他资源

- [man localectl \(1\)](#), [man locale \(7\)](#), 和 [man locale.conf \(5\)](#) man pages

4.5. 在文本控制台模式下更改字体大小

您可以更改虚拟控制台中的字体大小。

流程

1. 显示当前使用的字体文件：

```
# cat /etc/vconsole.conf

FONT="eurlatgr"
```

2. 列出可用的字体文件：

```
# ls -l /usr/lib/kbd/consolefonts/*.psfu.gz

/usr/lib/kbd/consolefonts/eurlatgr.psfu.gz
/usr/lib/kbd/consolefonts/LatArCyrHeb-08.psfu.gz
/usr/lib/kbd/consolefonts/LatArCyrHeb-14.psfu.gz
/usr/lib/kbd/consolefonts/LatArCyrHeb-16.psfu.gz
/usr/lib/kbd/consolefonts/LatArCyrHeb-16+.psfu.gz
/usr/lib/kbd/consolefonts/LatArCyrHeb-19.psfu.gz
```

选择一个支持您的字符集和代码页的字体文件。

3. 可选：要测试字体文件，请临时加载它：

```
# setfont LatArCyrHeb-16.psfu.gz
```

setfont 工具会立即应用字体文件，终端使用新的和字体大小，直到您重启或应用不同的字体文件。

4. 要返回 **/etc/vconsole.conf** 中定义的字体文件，请输入 **setfont** without any parameters.
5. 编辑 **/etc/vconsole.conf** 文件，并将 **FONT** 变量设置为 RHEL 应该在引导时加载的字体文件，例如：

```
FONT=LatArCyrHeb-16
```

6. 重启主机

```
# reboot
```

第 5 章 使用 OPENSSH 的两个系统间使用安全通讯

SSH(Secure Shell)是一种协议，它使用客户端-服务器架构在两个系统之间提供安全通信，并允许用户远程登录到服务器主机系统。与其他远程通信协议（如 FTP 或 Telnet）不同，SSH 会加密登录会话，这样可防止入侵者从连接中收集未加密的密码。

5.1. 生成 SSH 密钥对

您可以通过在本地系统上生成 SSH 密钥对并将生成的公钥复制到 OpenSSH 服务器来在不输入密码的情况下登录到 OpenSSH 服务器。要创建密钥的每个用户都必须运行此流程。

要在重新安装系统后保留之前生成的密钥对，请在创建新密钥前备份 `~/.ssh/` 目录。重新安装后，将其复制到主目录中。您可以为系统中的所有用户（包括 **root** 用户）进行此操作。

先决条件

- 您以希望使用密钥连接到 OpenSSH 服务器的用户身份登录。
- OpenSSH 服务器配置为允许基于密钥的身份验证。

流程

1. 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/<username>/.ssh/id_ecdsa):
Enter passphrase (empty for no passphrase): <password>
Enter same passphrase again: <password>
Your identification has been saved in /home/<username>/.ssh/id_ecdsa.
Your public key has been saved in /home/<username>/.ssh/id_ecdsa.pub.
The key fingerprint is:
SHA256:Q/x+qms4j7PCQ0qFd09iZEFHA+SqwBKRNauU72oZfaCI
<username>@<localhost.example.com>
The key's randomart image is:
+---[ECDSA 256]---+
|.00..0=++      |
|.. 0 .00 .     |
|. .. 0. 0      |
|...0.+...      |
|0.00.0 +S .    |
|.=.+ .0        |
|E.*. . . .     |
|.=..+ +.. 0    |
| . 00*+0.      |
+----[SHA256]-----+
```

您还可以通过输入 **ssh-keygen -t ed25519** 命令，使用 **ssh-keygen -t ed25519** 命令生成 RSA 密钥对，而无需任何参数或 Ed25519 密钥对。请注意，Ed25519 算法不兼容 FIPS-140，OpenSSH 在 FIPS 模式下无法使用 Ed25519 密钥。

2. 将公钥复制到远程机器上：

```
$ ssh-copy-id <username>@<ssh-server-example.com>
```

```

/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are
already installed
<username>@<ssh-server-example.com>'s password:
...
Number of key(s) added: 1

Now try logging into the machine, with: "ssh '<username>@<ssh-server-example.com>'" and
check to make sure that only the key(s) you wanted were added.

```

将 **<username>@<ssh-server-example.com>** 替换为您的凭证。

如果您没有在会话中使用 **ssh-agent** 程序，上一个命令会复制最新修改的 **~/.ssh/id*.pub** 公钥。要指定另一个公钥文件，或在 **ssh-agent** 内存中缓存的密钥优先选择文件中的密钥，使用带有 **-i** 选项的 **ssh-copy-id** 命令。

验证

- 使用密钥文件登录到 OpenSSH 服务器：

```
$ ssh -o PreferredAuthentications=publickey <username>@<ssh-server-example.com>
```

其他资源

- **ssh-keygen (1)** 和 **ssh-copy-id (1)** man page

5.2. 将基于密钥的身份验证设置为 OPENSSH 服务器的唯一方法

要提高系统安全性，通过在 OpenSSH 服务器上禁用密码身份验证来强制进行基于密钥的身份验证。

先决条件

- 已安装 **openssh-server** 软件包。
- **sshd** 守护进程正在服务器中运行。
- 已使用密钥连接到 OpenSSH 服务器。
详情请参阅 [生成 SSH 密钥对](#) 部分。

流程

1. 在文本编辑器中打开 **/etc/ssh/sshd_config** 配置，例如：

```
# vi /etc/ssh/sshd_config
```

2. 将 **PasswordAuthentication** 选项改为 **no**:

```
PasswordAuthentication no
```

3. 在新默认安装以外的系统中，检查 **PubkeyAuthentication** 参数是否未设置或设置为 **yes**。
4. 将 **Kbd interactiveAuthentication** 指令设置为 **no**。
请注意，对应的条目在配置文件中被注释掉，默认值为 **yes**。

5. 要在 NFS 挂载的主目录中使用基于密钥的验证，启用 **use_nfs_home_dirs** SELinux 布尔值：

```
# setsebool -P use_nfs_home_dirs 1
```

6. 如果您要进行远程连接，而不使用控制台或带外访问，在禁用密码验证前测试基于密钥的登录过程。
7. 重新载入 **sshd** 守护进程以应用更改：

```
# systemctl reload sshd
```

其他资源

- **sshd_config(5)** and **setsebool(8)** man pages on your system

5.3. 使用 SSH-AGENT 缓存 SSH 凭证

为了避免在每次发起 SSH 连接时输入密码短语，您可以使用 **ssh-agent** 实用程序为登录会话缓存 SSH 私钥。如果代理正在运行并且您的密钥被解锁，您可以使用这些密钥登录到 SSH 服务器，但不必再次输入密钥的密码。确保私钥和密语安全。

先决条件

- 您有一个运行 SSH 守护进程的远程主机，并可通过网络访问。
- 您知道登录到远程主机的 IP 地址或者主机名以及凭证。
- 您已用密码生成了 SSH 密钥对，并将公钥传送到远程机器。
详情请参阅 [生成 SSH 密钥对](#) 部分。

流程

1. 添加命令，在您的会话中自动启动 **ssh-agent** 到 **~/.bashrc** 文件：

- a. 在您选择的文本编辑器中打开 **~/.bashrc**，例如：

```
$ vi ~/.bashrc
```

- b. 在文件中添加以下行：

```
eval $(ssh-agent)
```

- c. 保存更改，退出编辑器。

2. 在 **~/.ssh/config** 文件中添加以下行：

```
AddKeysToAgent yes
```

如果此选项和 **ssh-agent** 在会话中启动，代理仅在您第一次连接到主机时提示输入密码。

验证

- 登录到使用代理中缓存私钥对应的公钥的主机，例如：

```
$ ssh <example.user>@<ssh-server@example.com>
```

请注意您不必输入密码短语。

5.4. 通过保存在智能卡中的 SSH 密钥进行身份验证

您可以在智能卡中创建和存储 ECDSA 和 RSA 密钥，并由 OpenSSH 客户端中的智能卡进行身份验证。智能卡验证替换了默认密码验证。

先决条件

- 在客户端中安装了 **opensc** 软件包，**pcscd** 服务正在运行。

流程

1. 列出所有由 OpenSC PKCS #11 模块提供的密钥，包括其 PKCS #11 URIs，并将输出保存到 **key.pub** 文件：

```
$ ssh-keygen -D pkcs11: > keys.pub
```

2. 将公钥传送到远程服务器。使用带有上一步中创建的 **key.pub** 文件的 **ssh-copy-id** 命令：

```
$ ssh-copy-id -f -i keys.pub <username@ssh-server-example.com>
```

3. 使用 ECDSA 密钥连接到 **<ssh-server-example.com>**。您只能使用 URI 的子集，它唯一引用您的密钥，例如：

```
$ ssh -i "pkcs11:id=%01?module-path=/usr/lib64/pkcs11/opensc-pkcs11.so" <ssh-server-example.com>
Enter PIN for 'SSH key':
[ssh-server-example.com] $
```

因为 OpenSSH 使用 **p11-kit-proxy** 打包程序和 OpenSC PKCS modprobe 模块注册到 **p11-kit** 工具，所以您可以简化上一个命令：

```
$ ssh -i "pkcs11:id=%01" <ssh-server-example.com>
Enter PIN for 'SSH key':
[ssh-server-example.com] $
```

如果您跳过 PKCS #11 URI 的 **id=** 部分，则 OpenSSH 会加载代理模块中可用的所有密钥。这可减少输入所需的数量：

```
$ ssh -i pkcs11: <ssh-server-example.com>
Enter PIN for 'SSH key':
[ssh-server-example.com] $
```

4. 可选：您可以使用 **~/.ssh/config** 文件中的同一 URI 字符串使配置持久：

```
$ cat ~/.ssh/config
IdentityFile "pkcs11:id=%01?module-path=/usr/lib64/pkcs11/opensc-pkcs11.so"
```

```
$ ssh <ssh-server-example.com>  
Enter PIN for 'SSH key':  
[ssh-server-example.com] $
```

ssh 客户端工具现在自动使用来自智能卡中的这个 URI 和密钥。

其他资源

- 系统上的 **p11-kit(8)**, **opensc.conf(5)**, **pcscd(8)**, **ssh(1)**, 和 **ssh-keygen(1)** 手册页

5.5. 其他资源

- **sshd(8)**, **ssh(1)**, **scp(1)**, **sftp(1)**, **ssh-keygen(1)**, **ssh-copy-id(1)**, **ssh_config(5)**, **sshd_config(5)**, **update-crypto-policies(8)**, 和 **crypto-policies(7)** man page
- [为使用非标准配置的应用程序和服务配置 SELinux](#)
- [使用 firewalld 控制网络流量](#)

第 6 章 使用日志文件对问题进行故障排除

日志文件包含有关系统的消息，包括内核、服务及其上运行的应用。这些信息可帮助故障排除问题或监控系统功能。Red Hat Enterprise Linux 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。

6.1. 处理 SYSLOG 信息的服务

以下两个服务处理 **syslog** 信息：

- **systemd-journald** 守护进程

systemd-journald 守护进程收集来自各种来源的信息并将其转发到 **Rsyslog** 以便进一步处理。**systemd-journald** 守护进程从以下来源收集信息：

- 内核
- 引导过程的早期阶段
- 启动并运行守护进程的标准和错误输出
- **Syslog**
- **Rsyslog** 服务

Rsyslog 服务根据类型和优先权对 **syslog** 信息进行排序，并将其写入 **/var/log** 目录下的文件中。**/var/log** 目录会永久保存日志信息。

6.2. 存储 SYSLOG 信息的日志文件

/var/log 目录下的以下日志文件存储 **syslog** 信息。

- **/var/log/messages** - 除以下外的所有 **syslog** 信息
- **/var/log/secure** - 与安全和验证相关的信息和错误
- **/var/log/maillog** - 与邮件服务器相关的信息和错误
- **/var/log/cron** - 与定期执行的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



注意

上述列表仅包含一些文件，**/var/log/** 目录中的实际文件列表取决于哪些服务和应用程序登录到这个目录。

6.3. 使用命令行查看日志

Journal 是 systemd 的一个组件，可帮助查看和管理日志文件。它解决了与传统日志记录相关的问题，与系统的其余部分紧密集成，并支持各种日志记录技术以及日志条目的访问管理。

您可以通过命令行，使用 **journalctl** 命令查看系统日志中的消息。

表 6.1. 查看系统信息

命令	描述
<code>journalctl</code>	显示所有收集的日志条目。
<code>journalctl FILEPATH</code>	显示与特定文件相关的日志。例如： <code>journalctl /dev/sda</code> 命令显示与 <code>/dev/sda</code> 文件系统相关的日志。
<code>journalctl -b</code>	显示当前引导的日志。
<code>journalctl -k -b -1</code>	显示当前引导的内核日志。

表 6.2. 查看有关特定服务的信息

命令	描述
<code>journalctl -b _SYSTEMD_UNIT=<name.service></code>	过滤日志以显示与 systemd 服务匹配的条目。
<code>journalctl -b _SYSTEMD_UNIT=<name.service> _PID=<number></code>	合并匹配。例如，这个命令显示与 <name.service> 和 PID <number> 匹配的 systemd-units 的日志。
<code>journalctl -b _SYSTEMD_UNIT=<name.service> _PID=<number> + _SYSTEMD_UNIT=<name2.service></code>	加号(+)分隔符将两个表达式按逻辑 OR 组合在一起。例如，这个命令显示来自带有 PID 的 <name.service> 服务进程的所有消息，加上来自 <name2.service> 服务（来自其任何进程）的所有消息。
<code>journalctl -b _SYSTEMD_UNIT=<name.service> _SYSTEMD_UNIT=<name2.service></code>	此命令显示与引用同一字段的任一表达式匹配的所有条目。在这里，这个命令会显示与 <name.service> 或 systemd-unit <name2.service> 匹配的日志。

表 6.3. 查看与特定引导相关的日志

命令	描述
<code>journalctl --list-boots</code>	显示引导号、其 ID 以及与引导相关的第一条和最后一个消息的时间戳列表。您可以在下一个命令中使用 ID 来查看详细信息。
<code>journalctl --boot=ID _SYSTEMD_UNIT=<name.service></code>	显示有关指定的引导 ID 的信息。

6.4. 查看 WEB 控制台中的日志

了解如何在 RHEL web 控制台中访问、查看和过滤日志。

6.4.1. 查看 web 控制台中的日志

RHEL 9 web 控制台日志部分是 **journalctl** 实用程序的 UI。您可以在 web 控制台界面中访问系统日志。

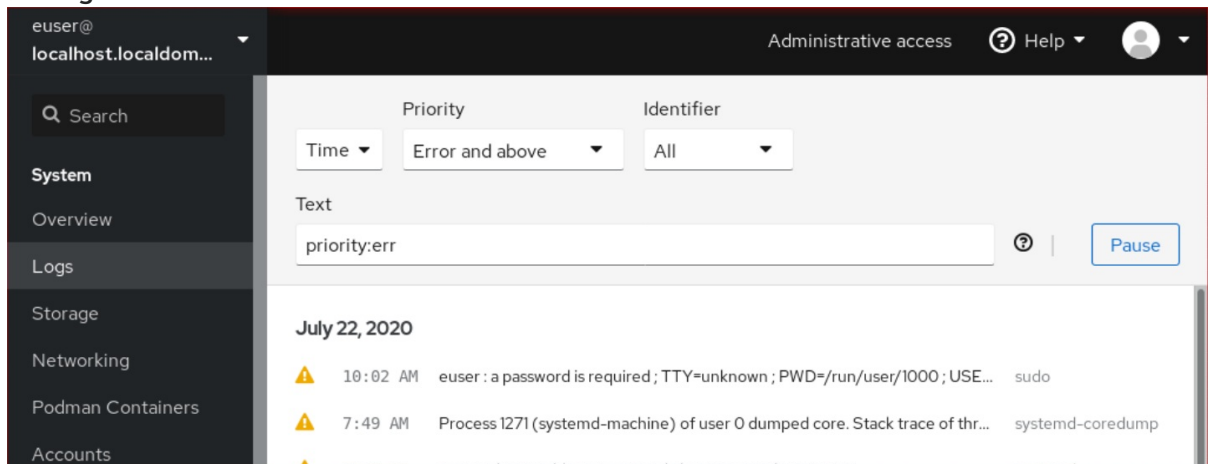
先决条件

- 您已安装了 RHEL 9 web 控制台。
- 您已启用了 cockpit 服务。
- 您的用户帐户被允许登录到 web 控制台。
具体步骤请参阅[安装并启用 Web 控制台](#)。

流程

1. 登录到 RHEL 9 web 控制台。
详情请参阅[登录到 web 控制台](#)。

2. 点 Logs。



3. 点击列表中的选定日志条目条目，打开日志条目详情。



注意

您可以使用 **暂停** 按钮在显示时暂停新日志条目。恢复新日志条目后，Web 控制台将加载您使用 **Pause** 按钮后报告的所有日志条目。

您可以根据时间、优先级或标识符过滤日志。如需更多信息，请参阅[web 控制台中的过滤日志](#)。

6.4.2. 在 web 控制台中过滤日志

您可以在 web 控制台中过滤日志条目。

先决条件

- 您已安装了 RHEL 9 web 控制台。
- 您已启用了 cockpit 服务。

- 您的用户帐户被允许登录到 web 控制台。
具体步骤请参阅[安装并启用 Web 控制台](#)。

流程

1. 登录到 RHEL 9 web 控制台。
详情请参阅[登录到 web 控制台](#)。
2. 点 **Logs**。
3. 默认情况下，Web 控制台显示最新的日志条目。要根据具体时间范围过滤，请点 **Time** 下拉菜单并选择一个首选的选项。
4. 默认情况下会显示 **Error 及更高级别**的日志列表。要根据不同的优先级过滤，请点击 **Error 及更高**下拉菜单并选择一个首选的优先级。
5. 默认情况下，Web 控制台会显示所有标识符的日志。要过滤特定标识符的日志，请点 **All** 下拉菜单并选择标识符。
6. 要打开日志条目，请点所选日志。

6.4.3. 在 web 控制台中过滤日志的文本搜索选项

文本搜索选项功能为过滤日志提供了大量选项。如果您决定使用文本搜索过滤日志，您可以使用三个下拉菜单中定义的预定义选项，或者您可以自己键入整个搜索。

下拉菜单

您可以使用三个下拉菜单来指定搜索的主参数：

- **Time**:此下拉菜单包含搜索的不同时间范围的预定义搜索。
- **Priority**:此下拉菜单提供了不同优先级级别的选项。它对应于 **journalctl --priority** 选项。默认优先级值为 **Error 及以上**。每次在不指定其它优先级时，会设置它。
- **Identifier**:在这个下拉菜单中，您可以选择要过滤的标识符。对应于 **journalctl --identifier** 选项。

限定符

您可以使用六个限定符来指定搜索。它们包含在用于过滤日志表的 Options 中。

日志字段

如果要搜索特定日志字段，可以用其内容指定字段。

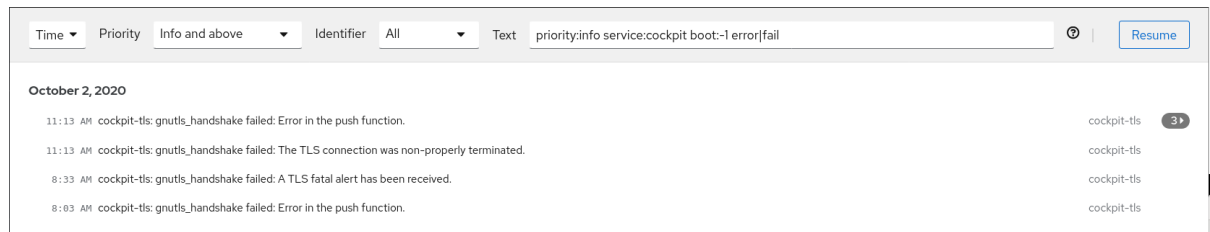
在日志信息中进行自由文本搜索

您可以在日志消息中过滤您选择的任何文本字符串。字符串也可以采用正则表达式的形式。

高级日志过滤 I

过滤 2020 年 10 月 22 日之后带有 'systemd' 识别的、日志字段 'JOB_TYPE' 是 'start' 或 'restart' 的所有日志信息。

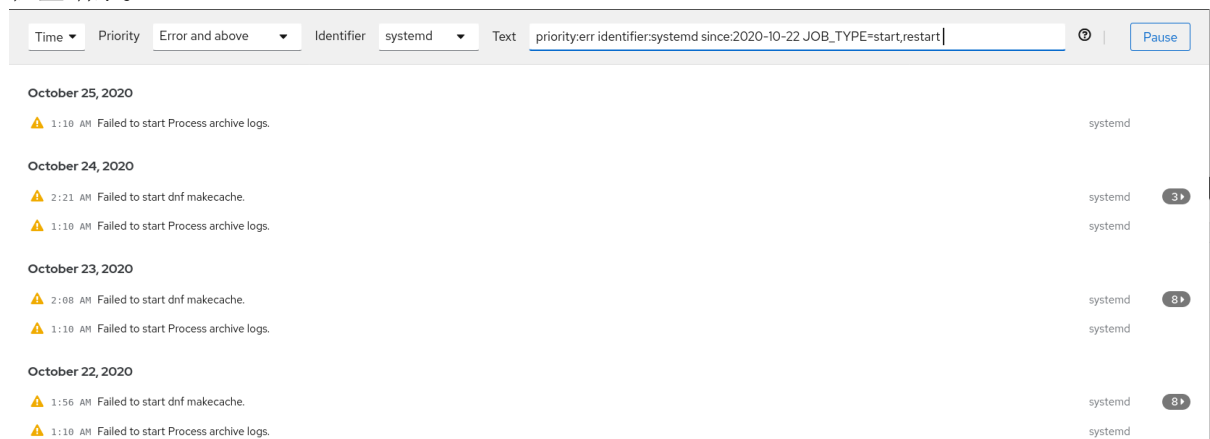
1. 在搜索字段中输入 **identifier:systemd since:2020-10-22 JOB_TYPE=start,restart**。
2. 检查结果。



高级日志过滤 II

过滤上一次启动前出现的所有来自 "cockpit.service" systemd 单元且邮件正文包含 "error" 或 "fail" 的所有日志消息。

1. 在搜索字段中输入 **service:cockpit boot:-1 error|fail**。
2. 检查结果。



6.4.4. 使用文本搜索框过滤 web 控制台中的日志

您可以在 web 控制台使用文本搜索框，根据不同的参数过滤日志。搜索结合了过滤下拉菜单、限定符、日志字段和自由格式字符串搜索的使用。

先决条件

- 您已安装了 RHEL 9 web 控制台。
- 您已启用了 cockpit 服务。
- 您的用户帐户被允许登录到 web 控制台。
具体步骤请参阅[安装并启用 Web 控制台](#)。

流程

1. 登录到 RHEL web 控制台。
详情请参阅 [Web 控制台的日志记录](#)。
2. 点 **Logs**。
3. 使用下拉菜单指定您想要过滤的三个主要的限定符 - 时间范围、优先级和标识符。
优先级 (Priority) 限定符总需要有一个值。如果没有指定，它会自动过滤 **Error 及以上** 优先级。请注意，您设置的选项反映了在文本搜索框中。
4. 指定您要过滤的日志字段。
您可以添加几个日志字段。

5. 您可以使用自由格式的字符串搜索任何其他内容。搜索框也接受正则表达式。

6.4.5. 日志过滤选项

有几个 **journalctl** 选项可用于在 web 控制台中过滤日志，这或许非常有用。其中一些已作为 web 控制台界面的下拉菜单的一部分进行介绍。

表 6.4. 表

选项名称	使用	注
priority	按消息优先级过滤输出。取单个数字或文本日志级别。日志级别是常见的 syslog 日志级别。如果指定了单一日志级别，则会显示具有此日志级别的所有消息或低（更重要）日志级别。	包括在优先级下拉菜单中。
identifier	显示被 syslog 标识为 SYSLOG_IDENTIFIER 的信息。可多次指定。	包括在 Identifier 下拉菜单中。
follow	仅显示最新的日志条目，并在新条目附加到日志中时持续打印新条目。	没有包含在下拉菜单中。
service	显示指定 systemd 单元的消息。可多次指定。	没有包含在下拉菜单中。对应于 journalctl --unit 参数。
boot	显示来自特定启动的消息。 正整数代表从日志开始查找启动，等于或小于零的整数代表将从日志末尾查找启动。因此，1 表示日志中的第一个引导（按时间顺序排列），2 为第 2 个，以此类推；-0 是最后一次引导，-1 是最后一次引导的前一个，以此类推。	在时间下拉菜单中作为 Current boot 或 Previous boot 。其他选项需要手动编写。
since	开始显示指定日期更新或分别位于指定日期或比指定日期旧的条目。日期规格应为 "2012-10-30 18:17:16"。如果省略了时间部分，使用 "00:00:00"。如果只省略了秒的组件，使用 ":00"。如果省略了日期的部分，使用当前日期。另外，还可以使用 "yesterday"、"today"、"tomorrow"（分别代表前一天、当天和明天的 00:00:00），以及 "now"（代表当前时间）。最后，可以指定相对时间，前缀为 "-" 或 "+"，分别引用当前时间前或之后的时间。	没有包含在下拉菜单中。

6.5. 其他资源

- 您系统上的 **journalctl (1)** 手册页
- [配置远程日志记录解决方案](#)

第 7 章 管理用户和组

防止未经授权访问文件和流程需要准确的用户和组管理。如果您没有集中管理帐户，或者您只需要特定系统上的用户帐户或组，您可以在主机上本地创建它们。

7.1. 管理用户和组群帐户简介

用户和组群的控制是 Red Hat Enterprise Linux (RHEL) 系统管理的核心元素。每个 RHEL 用户都有不同的登录凭证，并可分配给不同的组以自定义其系统权限。

7.1.1. 用户和组介绍

创建文件的用户是该文件的拥有者以及该文件的组所有者。这个文件会单独为拥有者、组和组以外的成员分配读、写和执行权限。文件所有者只能由 **root** 用户更改。**root** 用户和文件拥有者都可以更改对该文件的访问权限。常规用户可以将他们拥有的文件的组群所有权改为他们所属的组。

每个用户都与一个唯一数字身份号关联，称为 *user ID (UID)*。每个组都与一个 *group ID (GID)* 关联。组群中的用户共享相同的读取、写入和执行该组所拥有的文件的权限。

7.1.2. 配置保留的用户和组群 ID

默认情况下，RHEL 为系统用户和组保留在 1000 以下的用户和组群 ID。您可以在 **setup** 软件包中找到保留的用户和组群 ID。在更改 **UID_MIN** 和 **GID_MIN** 值前创建的用户和组的 UID 和 GID 不会改变。保留的用户和组群 ID 记录在：

```
/usr/share/doc/setup/uidgid
```

要将 ID 分配给从 5000 开始的新用户和组，因为保留范围将来可能会增加。

修改 **/etc/login.defs** 文件中的 **UID_MIN** 和 **GID_MIN** 参数，以定义默认值(1000)以外的启动 ID。



警告

不要通过更改 **SYS_UID_MAX** 来提高系统 1000 以上保留的 ID，以避免与保留 1000 限制的冲突。

流程

1. 在编辑器中打开 **/etc/login.defs** 文件。
2. 设置 **UID_MIN** 变量，例如：

```
# Min/max values for automatic uid selection in useradd
#
UID_MIN          5000
```

3. 设置 **GID_MIN** 变量，例如：


```
# Min/max values for automatic gid selection in groupadd
#
GID_MIN          5000
```

常规用户动态分配的 UID 和 GID 现在从 5000 开始。

7.1.3. 用户私人组群

RHEL 使用用户私有组(UPG)系统配置, 这有助于管理 Linux 组。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。用户私人组群的名称与为其创建的用户名称相同, 该用户是该用户私人组群中的唯一成员。

UPG 简化了多个用户之间在项目上的协作。此外, UPG 系统配置可以安全地为新创建的文件或目录设置默认权限, 因为它允许该用户和此用户所属的组对文件或目录进行修改。

所有本地组的列表都存储在 `/etc/group` 配置文件中。

7.2. 管理用户帐户入门

Red Hat Enterprise Linux 是多用户操作系统, 可让不同计算机中的多个用户访问在同一机器中安装的单一系统。每个用户都在其自身帐户下运行, 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。

以下是不同类型的用户帐户：

- **普通用户帐户：**
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。
- **系统用户帐户：**
系统用户帐户代表系统中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。如果这些帐户是远程配置和提供的, 如 LDAP 配置的实例中, 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。要为用户和组定义 min/max ID, 系统用户和组, 请查看 `/etc/login.defs` 文件。

- **组：**
组是出于共同目的将多个用户帐户连接在一起的实体, 例如对特定文件授予访问权限。

7.2.1. 使用命令行工具管理帐户和组

使用以下基本命令行工具管理用户帐户和组。

流程

- 创建新的用户帐户：

```
# useradd example.user
```

- 为 example.user 所属用户帐户分配新密码：

```
# passwd example.user
```

- 将用户添加到组中：

```
# usermod -a -G example.group example.user
```

其他资源

- **useradd (8)**, **passwd (1)**, 和 **usermod (8)** man pages

7.3. 从命令行管理用户

您可以使用命令行界面（CLI）来管理用户和组。这可让您在 Red Hat Enterprise Linux 环境中添加、删除和修改用户和组。

7.3.1. 使用命令行添加新用户

您可以使用 **useradd** 工具添加新用户。

先决条件

- 有 **Root** 访问权限

流程

- 添加新用户，使用：

```
# useradd <options> <username>
```

使用 **useradd** 命令的选项替换 options，并使用用户名称替换 username。

例 7.1. 添加新用户

添加用户 ID 为 **5000** 的用户 **sarah**，使用：

```
# useradd -u 5000 sarah
```

验证

- 要验证新用户是否已添加，使用 **id** 工具程序。

```
# id sarah
```

该命令返回：

```
uid=5000(sarah) gid=5000(sarah) groups=5000(sarah)
```

其他资源

- [useradd 手册页](#)

7.3.2. 使用命令行添加新组

您可以使用 **groupadd** 工具添加新组。

先决条件

- 有 **Root** 访问权限

流程

- 要添加新组，请使用：

```
# groupadd options group-name
```

使用 **groupadd** 命令的命令行选项替换 **options**，并使用 **group-name** 替换 **group-name**。

例 7.2. 添加新组

要添加组 ID 为 **5000** 的组 **sysadmins**，请使用：

```
# groupadd -g 5000 sysadmins
```

验证

- 要验证新组是否已添加，使用 **tail** 实用程序。

```
# getent group sysadmin
```

该命令返回：

```
sysadmins:x:5000:
```

其他资源

- [groupadd 手册页](#)

7.3.3. 从命令行将用户添加到补充组中

您可以将用户添加到补充组中，以管理权限或启用对特定文件或设备的访问权限。

先决条件

- 有 **root** 访问权限

流程

- 要在用户的附加组中添加一个组，请使用：

```
# usermod --append -G <group_name> <username>
```

验证

- 要验证新的组被添加到用户 **sysadmin** 的附加组中，请使用：

```
# groups <username>
```

7.3.4. 创建组目录

在 UPG 系统配置下，您可以将 **set-group** 身份权限（**setgid** 位）应用到目录。**setgid** 位使得管理共享目录的组项目变得更加简单。当您将 **setgid** 位应用到某个目录中时，在该目录中创建的文件会自动分配给拥有该目录的组群。在此组中具有写和执行权限的任何用户现在可以在目录中创建、修改和删除文件。

先决条件

- 有 **Root** 访问权限

流程

1. 创建目录：

```
# mkdir <directory-name>
```

2. 创建组：

```
# groupadd <group-name>
```

3. 向组中添加用户：

```
# usermod --append -G <group_name> <username>
```

4. 将目录的用户和组群所有权与 **group-name** 组关联：

```
# chgrp <group_name> <directory>
```

5. 设置写入权限以允许用户创建和修改文件和目录，并设置 **setgid** 位以在目录中应用此权限：

```
# chmod g+rwxs <directory>
```

验证

- 要验证设置权限的正确性，请使用：

```
# ls -ld <directory>
```

该命令返回：

```
*drwx__rws__r-x.* 2 root __group-name_ 6 Nov 25 08:45 __directory-name_
```

7.3.5. 在命令行上删除用户

您可以使用命令行删除用户帐户。此外，下面提到的命令是删除用户帐户的命令，也可选择性地删除用户数据和元数据，如其主目录和配置文件。

- 您有 **root** 访问权限。
- 用户当前存在。
- 确保用户已退出登录：

```
# loginctl terminate-user user-name
```

- 要仅删除用户帐户，而不删除用户数据：

```
# userdel user-name
```

- 要删除用户、数据和元数据：
 - a. 删除用户、其主目录、其邮件假脱机及其 SELinux 用户映射：

```
# userdel --remove --selinux-user user-name
```

- b. 删除其他用户元数据：

```
# rm -rf /var/lib/AccountsService/users/user-name
```

此目录存储系统在主目录可用之前所需的有关用户的信息。根据系统配置，主目录可能无法使用，直到用户在登录屏幕进行身份验证。



重要

如果您没有删除此目录，之后又的重新创建同一用户，重新创建的用户仍将使用从已删除用户继承来的某些设置。

其他资源

- [userdel \(8\) 手册页](#)

7.4. 在 WEB 控制台中管理用户帐户

RHEL web 控制台为添加、编辑和删除系统用户帐户提供了一个图形界面。

您还可以在 web 控制台中设置密码过期和终止用户会话。

7.4.1. 使用 Web 控制台添加新帐户

您可以将用户帐户添加到系统，并通过 RHEL web 控制台向帐户设置管理权限。

先决条件

- 您已安装了 RHEL 9 web 控制台。
- 您已启用了 cockpit 服务。
- 您的用户帐户被允许登录到 web 控制台。
具体步骤请参阅[安装并启用 Web 控制台](#)。

流程

1. 登录到 RHEL 9 web 控制台。
详情请参阅[登录到 web 控制台](#)。
2. 点 **Account**。
3. 点 **Create New Account**。
4. 在 **Full Name** 字段中输入用户全名。
RHEL web 控制台会自动在全名中推荐用户名并在 **User Name** 字段中填充该用户名。如果您不想使用原始命名规则（由名的第一个字母和完整的姓组成），对它进行更新。
5. 在 **Password/Confirm** 字段中输入密码并重新输入该密码以便验证您的密码是否正确。
下面字段的颜色栏显示您输入密码的安全级别，其不允许您创建具有弱密码的用户。
6. 点 **Create** 保存设置并关闭对话框。
7. 选择新创建的帐户。
8. 在 **Groups** 下拉菜单中，选择您要添加到新帐户的组。

The screenshot shows a 'New User' dialog box with the following fields and options:

- Full name:** New User
- User name:** nuser
- Groups:** nuser (selected from a dropdown menu)
- Last login:** Never
- Options:**
 - ☐ Disallow interactive password
 - ☒ Never expire account [edit](#)
- Password:**
 - [Set password](#)
 - [Force change](#)
 - Never expire password [edit](#)

At the top right of the dialog, there are two buttons: 'Terminate session' (grey) and 'Delete' (red).

验证

- 您可以在 **Accounts** 设置中看到新帐户，您可以使用其凭证连接到该系统。

7.4.2. 在 web 控制台中强制密码过期

默认情况下，用户帐户将密码设定为永远不会过期。您可以设置系统密码在指定的天数后过期。当密码过期时，用户必须在下次登录时更改密码，然后才能访问系统。

先决条件

- 您已安装了 RHEL 9 web 控制台。

- 您已启用了 cockpit 服务。
- 您的用户帐户被允许登录到 web 控制台。
具体步骤请参阅[安装并启用 Web 控制台](#)。

流程

1. 登录到 RHEL 9 web 控制台。
2. 点 **Account**。
3. 选择您要强制密码过期的用户帐户。
4. 点 **Password** 行上的 **edit**。

Password	Set password	Force change	Require password change on March 2, 2024	edit
-----------------	------------------------------	------------------------------	--	----------------------

5. 在 **Password expiration** 对话框中，选择 **Require password change every ... days** 并输入一个正数，代表密码过期的天数。
6. 点 **Change**。
Web 控制台会立即在 **Password** 行上显示将来密码更改请求的日期。

7.5. 使用命令行编辑用户组

用户属于某个组集合，允许用户的逻辑组集合对文件和文件夹具有类似的访问权限。您可以从命令行编辑主和补充用户组，以更改用户的权限。

7.5.1. 主和补充用户组

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

在 RHEL 中，用户组可以充当主或补充组。主和补充组具有以下属性：

主组

- 每个用户始终只有一个主组。
- 您可以更改用户的主组。

补充组

- 您可以将现有用户添加到现有的补充组中，以使用相同的安全和访问权限管理组中的用户。
- 用户可以是零、一个或多个补充组的成员。

7.5.2. 列出用户的主和补充组

您可以列出用户的组，以查看他们所属的主和补充组。

流程

- 显示用户的主组以及任何补充组的名称：

\$ groups user-name

如果不提供用户名，则命令将显示当前用户的组成员身份。第一个组是主组，后跟可选的补充组。

例 7.3. 列出用户 sarah 的组：

```
$ groups sarah
```

输出显示：

```
sarah : sarah wheel developer
```

用户 **sarah** 有一个主组 **sarah**，它是补充组 **wheel** 和 **developer** 的成员。

7.5.3. 更改用户的主组

您可以将现有用户的主组更改为一个新组。

先决条件

1. **root** 访问权限
2. 新组必须存在

流程

- 更改用户的主组：

```
# usermod -g <group-name> <user-name>
```



注意

更改用户的主组时，命令还会将用户主目录中所有文件的组所有权自动更改为新的主组。您必须手动修复用户主目录外文件的组所有权。

- 验证您是否更改了用户的主组：

```
$ groups <username>
```

7.5.4. 从命令行将用户添加到补充组中

您可以将用户添加到补充组中，以管理权限或启用对特定文件或设备的访问权限。

先决条件

- 有 **root** 访问权限

流程

- 要在用户的附加组中添加一个组，请使用：

```
# usermod --append -G <group_name> <username>
```

验证

- 要验证新的组被添加到用户 **sysadmin** 的附加组中，请使用：

```
# groups <username>
```

7.5.5. 从补充组中删除用户

您可以从补充组中删除现有的用户，以限制他们对文件和设备的权限或访问。

先决条件

- 有 **root** 访问权限

流程

- 从补充组中删除用户：

```
# gpasswd -d <user-name> <group-name>
```

验证

- 验证您是否从次要组 **developers** 中删除了用户 **sarah**：

```
$ groups <username>
```

7.5.6. 更改用户的所有补充组

您可以覆盖您希望用户保留其成员的补充组的列表。

先决条件

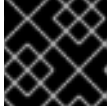
- 您有 **root** 访问权限。
- 补充组必须存在。

流程

- 覆盖用户的补充组的列表：

```
# usermod -G <group-names> <username>
```

要将用户一次添加到多个补充组中，请使用逗号分隔组名称，并且没有插入空格。例如：**wheel,developer**。



重要

如果用户是当前您未指定的组的成员，则该命令会从组中删除该用户。

验证

- 验证您是否正确设置了补充组列表：

```
# groups <username>
```

7.6. 更改和重置根密码

如果现有的 **root** 密码不再满意，您可以以 **root** 用户身份和非 **root** 用户更改它。

7.6.1. 作为 **root** 用户更改 **root** 密码

您可以使用 **passwd** 命令以 **root** 用户身份更改 **root** 密码。

先决条件

- 有 **Root** 访问权限

流程

- 要更改 **root** 密码，使用：

```
# passwd
```

在修改前，会提示您输入您当前的密码。

7.6.2. 以非 **root** 用户的身份更改或重置根密码

您可以使用 **passwd** 命令以非 **root** 用户身份更改或重置 **root** 密码。

先决条件

- 您可以以非 **root** 用户身份登录。
- 有以 **root** 身份使用 **sudo** 执行命令的权限。

流程

- 以 **wheel** 组中的非 **root** 用户身份修改或重置 **root** 密码，请使用：

```
$ sudo passwd root
```

此时会提示您输入当前的非 **root** 密码，然后才能更改 **root** 密码。

7.6.3. 重置 **root** 密码

如果您无法以 `root` 用户身份登录，且没有具有 `sudo` 权限的非 `root` 用户，您可以重置 `root` 密码或不属于管理 `wheel` 组，您可以通过将系统引导至特殊模式来重置 `root` 密码。在这个模式中，引导过程会在系统从 `initramfs` 接管控制权到实际系统前停止。

流程

1. 重启系统，在 GRUB 引导屏幕上按 **e** 键中断引导过程。

此时会出现内核引导参数。

```
load_video
set gfx_payload=keep
insmod gzio
linux ($root)/vmlinuz-5.14.0-70.22.1.el9_0.x86_64 root=/dev/mapper/rhel-root ro crash\
kernel=auto resume=/dev/mapper/rhel-swap rd.lvm.lv/swap rhgb quiet
initrd ($root)/initramfs-5.14.0-70.22.1.el9_0.x86_64.img $tuned_initrd
```

2. 将光标设置为以 `linux` 开头的行的末尾。
3. 将 `rd.break` 附加到以 `linux` 开头的行的末尾。
4. 按 **Ctrl+x** 使用更改的参数启动系统。
此时会出现 `switch_root` 提示符。
5. 将文件系统重新挂载为可写：

```
# mount -o remount,rw /sysroot
```

默认情况下，文件系统以只读形式挂载到 `/sysroot` 目录中。将文件系统重新挂载为可写才可以更改密码。

6. 进入 `chroot` 环境：

```
# chroot /sysroot
```

7. 重置 `root` 密码：

```
# passwd
```

按照命令行中的步骤完成 `root` 密码的更改。

8. 在下次系统引导时启用 SELinux 重新标记进程：

```
# touch /.autorelabel
```

9. 退出 `chroot` 环境：

```
# exit
```

10. 退出 `switch_root` 提示符，以重启系统：

```
exit
```

11. 等待 SELinux 重新标记过程完成。请注意，重新标记大型磁盘可能需要很长时间。系统会在这个过程完成后自动重启。

验证

1. 使用新的 **root** 密码，以 root 用户身份登录。
2. 可选：显示与当前有效用户 ID 关联的用户名：

```
# whoami
```

第 8 章 管理 SUDO 访问

系统管理员可以授予 **sudo** 访问权限，以允许非 root 用户执行通常为 root 用户保留的管理命令。因此，非 root 用户可以在不登录 root 用户帐户的情况下执行此类命令。

8.1. SUDOERS 中的用户授权

/etc/sudoers 文件，默认情况下，在 **/etc/sudoers.d/** 目录中指定哪些用户可以使用 **sudo** 命令以其他用户身份执行命令。规则可应用到单个用户和用户组。您还可以使用别名轻松地为主机组、命令甚至用户定义规则。

当用户输入带有 **sudo** 的命令时（因为用户没有授权），系统会将一条包含字符串 **<username> : user NOT in sudoers** 的消息记录到日志中。

默认的 **/etc/sudoers** 文件提供授权信息和示例。您可以通过取消相应行的注释来激活特定的示例规则。带有用户授权的部分标有以下介绍：

```
## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
```

您可以使用以下格式创建新的 **sudoers** 授权，并修改现有的授权：

```
<username> <hostname.example.com>=(<run_as_user>:<run_as_group>) <path/to/command>
```

其中：

- **<username>** 是输入命令的用户，如 **user1**。如果值以 **%** 开头，则它会定义一个组，例如 **%group1**。
- **<hostname.example.com>** 是应用该规则的主机的名称。
- 部分 **(<run_as_user>:<run_as_group>)** 定义执行命令的用户或组。如果省略这部分，**<username>** 可以以 root 用户身份执行命令。
- **<path/to/command>** 是该命令的完整的绝对路径。您还可以通过在命令路径后面添加这些选项，将用户限制为仅使用特定的选项和参数执行命令。如果没有指定任何选项，用户可以使用带有所有选项的命令。

您可以通过将任何这些变量替换为 **ALL**，来将规则应用到所有用户、主机或命令。



警告

通过在规则的某些或多个部分中使用 **ALL**，可能会导致严重的安全风险。

您可以使用 **!** 运算符对参数求反。例如，**!root** 指定除 root 以外的所有用户。请注意，允许特定用户、组和命令比禁止特定用户、组和命令更安全。这是因为允许规则也会阻止新的未授权的用户或组。

**警告**

避免对命令使用负规则，因为用户可以使用 **alias** 命令重命名命令来克服此类规则。

系统会从头到尾读取 **/etc/sudoers** 文件。因此，如果文件中包含用户的多个条目，则按顺序应用条目。如果值冲突，系统将使用最后匹配的项，即使它不是最具体的匹配。

要在系统更新期间保留规则，并轻松地修复错误，请在 **/etc/sudoers.d/** 目录中创建新文件，而不是直接在 **/etc/sudoers** 文件中输入规则。当系统在 **/etc/sudoers** 文件中达到以下行时，会读取 **/etc/sudoers.d** 目录中的文件：

```
#includedir /etc/sudoers.d
```

请注意，此行开头的数字符号(**#**)是语法的一部分，并不意味着该行是一个注释行。该目录中文件的名称不得包含句点，不得以波形符(**~**)结尾。

其他资源

- **sudoers (5) 手册页**

8.2. 添加 SUDO 规则以允许组的成员以 ROOT 用户身份执行命令

系统管理员可以通过授予它们 **sudo** 访问权限来允许非 root 用户执行管理命令。**sudo** 命令为用户提供了管理访问权限，而无需使用 root 用户的密码。

当用户需要执行管理命令时，您可以在使用 **sudo** 命令前执行该命令。如果用户具有命令的授权，则就像它们是 root 一样执行命令。

请注意以下限制：

- 只有 **sudoers** 配置文件中列出的用户才能使用 **sudo** 命令。
- 命令在用户的 shell 下执行，而不是在 root shell 下执行。但是，有一些例外情况，如向任何用户授予完整的 **sudo** 权限时。在这种情况下，用户可以切换到 root shell，并在 root shell 中运行命令。例如：
- **sudo -i**
- **sudo su -**

先决条件

- 有对系统的 root 访问权限。

流程

1. 以 root 身份，打开 **/etc/sudoers** 文件。

```
# visudo
```

`/etc/sudoers` 文件定义 **sudo** 命令应用的策略。

2. 在 `/etc/sudoers` 文件中，找到为 **wheel** 管理组中用户授予 **sudo** 访问权限的行。

```
## Allows people in group wheel to run all commands
%wheel    ALL=(ALL)    ALL
```

3. 确保以 **%wheel** 开头的行没有使用数字符号(**#**)注释掉。
4. 保存所有更改并退出编辑器。
5. 将您要授予 **sudo** 访问权限的用户添加到 **wheel** 管理组中。

```
# usermod --append -G wheel <username>
```

将 **<username>** 替换为用户名称。

验证

- 以 **wheel** 组的成员登录并运行：

```
# sudo whoami
root
```

其他资源

- **sudo (8)**, **sudoers (5)** 和 **visudo (8)** man pages

8.3. 使非特权用户运行某些命令

作为管理员，您可以通过在 `/etc/sudoers.d/` 目录中配置策略来允许非特权用户在特定工作站上输入某些命令。这比为用户授予完全 **sudo** 访问权限或授予 **root** 密码更加安全，理由如下：

- 对特权操作进行更精细的控制。您可以允许用户对特定主机执行某些操作，而不是向他们提供完整的管理访问权限。
- 更好的日志记录。当用户通过 **sudo** 执行操作时，该操作将使用其用户名而非 **root** 记录。
- 透明控制。当用户试图使用 **sudo** 权限时，您可以为用户设置电子邮件通知。

先决条件

- 有对系统的 **root** 访问权限。

流程

1. 在 `/etc/sudoers.d` 目录中创建一个新文件：

```
# visudo -f /etc/sudoers.d/<filename>
```

文件会在编辑器中自动打开。

2. 在 `/etc/sudoers.d/<filename>` 文件中添加以下行：

```
<username> <hostname.example.com> = (<run_as_user>:<run_as_group>)
<path/to/command>
```

- 将 **<username>** 替换为用户名称。
- 将 **<hostname.example.com>** 替换为主机的 URL。
- 将 **(<run_as_user> : <run_as_group>)** 替换为可执行命令的用户或组。如果省略这部分，**<username>** 可以以 root 用户身份执行命令。
- 将 **<path/to/command>** 替换为命令的完整的绝对路径。您还可以通过在命令路径后面添加这些选项，将用户限制为仅使用特定的选项和参数执行命令。如果没有指定任何选项，用户可以使用带有所有选项的命令。
- 要在同一主机上的一行上允许两个和多个命令，您可以使用逗号后跟空格进行分割，来列出它们。

例如，要允许 **user1** 在 **host1.example.com** 上执行 **dnf** 和 **reboot** 命令，请输入：

```
user1 host1.example.com = /bin/dnf, /sbin/reboot
```

1. 可选：要在每次尝试使用 **sudo** 权限时收到电子邮件通知，请在文件中添加以下行：

```
Defaults    mail_always
Defaults    mailto="<email@example.com>"
```

2. 保存更改，退出编辑器。

验证

1. 要验证用户是否可以使用 **sudo** 特权运行命令，请切换帐户：

```
# su <username> -
```

2. 以用户身份，使用 **sudo** 命令输入命令：

```
$ sudo whoami
[sudo] password for <username>:
```

输入用户的 **sudo** 密码。

3. 如果正确配置了特权，**sudo** 以配置的用户身份执行命令。例如，使用 **dnf** 命令，它显示以下输出：

```
...
usage: dnf [options] COMMAND
...
```

如果系统返回以下出错信息，用户不允许使用 **sudo** 运行命令。

```
<username> is not in the sudoers file. This incident will be reported.
```

- + 如果系统返回以下出错信息，其配置未正确完成。

`<username> is not allowed to run sudo on <host.example.com>.`

+ 如果系统返回以下出错信息，则在用户的规则中没有正确定义该命令。

``Sorry, user _<username>_ is not allowed to execute '_<path/to/command>_' as root on _<host.example.com>_.``

其他资源

- **visudo (8)**, 和 **sudoers (5)** man pages

8.4. 使用 RHEL 系统角色应用自定义 SUDOERS 配置

您可以使用 **sudo** RHEL 系统角色在受管节点上应用自定义 **sudoers** 配置。这样，您可以定义哪些用户可以在哪些主机上运行哪些命令，从而提高配置效率和更精细的控制。

先决条件

- [您已准备好控制节点和受管节点](#)
- 以可在受管主机上运行 `playbook` 的用户登录到控制节点。
- 用于连接到受管节点的帐户具有 **sudo** 权限。

流程

1. 创建一个包含以下内容的 `playbook` 文件，如 `~/playbook.yml`：

```
---
- name: "Configure sudo"
  hosts: managed-node-01.example.com
  tasks:
    - name: "Apply custom /etc/sudoers configuration"
      ansible.builtin.include_role:
        name: redhat.rhel_system_roles.sudo
      vars:
        sudo_sudoers_files:
          - path: "/etc/sudoers"
            user_specifications:
              - users:
                  - <user_name>
                hosts:
                  - <host_name>
                commands:
                  - <path_to_command_binary>
```

`playbook` 中指定的设置包括：

users

适用于该规则的用户列表。

主机

适用于该规则的主机列表。对于所有主机，您可以使用 **ALL**。

commands

适用于该规则的命令列表。对于所有命令，您可以使用 **ALL**。

有关 playbook 中使用的所有变量的详情，请查看控制节点上的 `/usr/share/ansible/roles/rhel-system-roles.sudo/README.md` 文件。

2. 验证 playbook 语法：

```
$ ansible-playbook --syntax-check ~/playbook.yml
```

请注意，这个命令只验证语法，不会防止错误但有效的配置。

3. 运行 playbook：

```
$ ansible-playbook ~/playbook.yml
```

验证

1. 在受管节点上，验证 playbook 是否应用了新规则。

```
# cat /etc/sudoers | tail -n1  
<user_name> <host_name>= <path_to_command_binary>
```

其他资源

- `/usr/share/ansible/roles/rhel-system-roles.sudo/README.md` file
- `/usr/share/doc/rhel-system-roles.sudo/sudo/` directory

第 9 章 管理文件系统权限

文件系统权限控制用户和组帐户读、修改和执行文件的内容以及进入目录的能力。仔细设置权限以保护您的数据免受未授权的访问。

9.1. 管理文件权限

每个文件或目录都有三个级别的所有权：

- 用户所有者 (u)。
- 组所有者 (g)。
- 其他 (o)。

可为每个级别的所有权分配以下权限：

- 读 (r)。
- 写 (w)。
- 执行 (x)。

请注意，文件的执行权限允许执行该文件。目录的执行权限允许访问目录中的内容，但不执行它。

创建新文件或目录时，会自动为其分配默认权限集。文件或目录的默认权限基于两个因素：

- 基本权限。
- user file-creation mode mask (umask)。

9.1.1. 基本文件权限

每当创建新文件或目录时，会自动为其分配基本权限。文件或目录的基本权限可以使用符号或者数值表示。

权限	符号	数值
无权限	---	0
执行	--x	1
写	-w-	2
写和执行	-wx	3
读	r--	4
读和执行	r-x	5
读写	rw-	6

读、写、执行	rwX	7
--------	-----	---

目录的基本权限是 **777 (drwxrwxrwxrwx)**，它为任何人都授予读、写和执行的权限。这意味着目录所有者、组和其它可以列出目录的内容，并可以在该目录下（以及其子目录）中创建、删除和编辑项。

请注意，一个目录中的单个文件可以有它们自己的权限，例如可以阻止用户您编辑它们，即使用户对该目录有非受限的访问权限。

文件的基本权限为 **666 (-rw-rw-rw-)**，它为所有人都授予读取和写入的权限。这意味着文件所有者、组和其它用户都可以读和编辑该文件。

例 9.1. 文件的权限

如果文件有以下权限：

```
$ ls -l
-rwxrw----. 1 sysadmins sysadmins 2 Mar 2 08:43 file
```

- **-** 表示它是文件。
- **rwX** 表示文件所有者有读、写和执行文件的权限。
- **rw-** 表示组有读写权限，但不能执行文件。
- **---** 表示其他用户没有读、写或执行文件的权限。
- **.** 表示为该文件设定了 SELinux 安全上下文。

例 9.2. 目录的权限

如果一个目录有以下权限：

```
$ ls -dl directory
drwxr----. 1 sysadmins sysadmins 2 Mar 2 08:43 directory
```

- **d** 表示它是一个目录。
- **rwX** 表示目录所有者有读、写和访问目录内容的权限。
作为目录所有者，您可以列出目录中的项目（文件、子目录），访问这些项目的内容并进行修改。
- **r-x** 表示组有读取目录内容的权限，但没有写权限 - 创建新条目或删除文件。**x** 权限意味着您也可以使用 **cd** 命令访问该目录。
- **---** 表示其他用户没有权限读取、写入或者访问该目录的内容。
作为不是用户拥有者或作为组的用户，您无法列出目录中的项目、访问这些项目的信息或修改它们。
- **.** 表示为该目录设定了 SELinux 安全性上下文。



注意

自动分配给某个文件或者目录的基本权限不是文件或目录最终的默认权限。当您创建文件或目录时，基本权限会被 `umask` 更改。基本权限和 `umask` 的组合会为文件和目录创建默认权限。

9.1.2. 用户文件创建模式掩码

用户文件创建模式掩码(`umask`)是一个变量，用于控制如何为新创建的文件和目录设置文件权限。`umask` 会自动从基本权限值中删除权限，以提高 Linux 系统的整体安全性。`umask` 可以用符号 或 八进制 值表示。

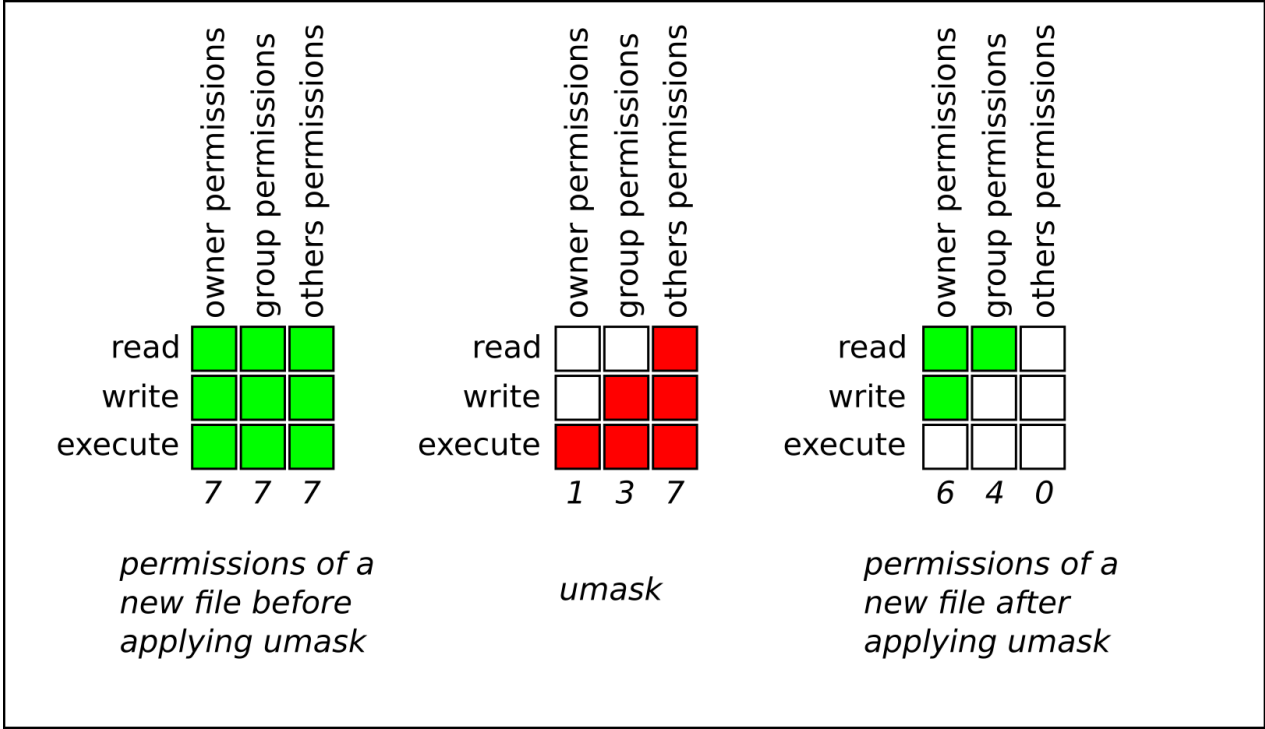
权限	符号	数值
读、写和执行	<code>rwX</code>	0
读写	<code>rw-</code>	1
读和执行	<code>r-X</code>	2
读	<code>r--</code>	3
写和执行	<code>-wX</code>	4
写	<code>-w-</code>	5
执行	<code>--X</code>	6
无权限	<code>---</code>	7

标准用户和 `root` 用户的默认 `umask` 为 `0022`。

`umask` 的第一个数字代表特殊权限 (sticky 位)。`umask` 的最后三位数字分别代表从用户拥有者 (`u`)、组群所有者 (`g`) 和其它 (`o`) 中删除的权限。

例 9.3. 在创建文件时应用 `umask`

下面的例子演示了， 对一个基本权限为 `777` 的文件应用值为 `0137` 的 `umask`， 使在创建该文件时其默认权限变为 `640`。



9.1.3. 默认的文件权限

为所有新创建的文件和目录自动设置默认权限。默认权限的值通过将 umask 应用到基本权限来确定。

例 9.4. 目录的默认权限

当 标准用户 或 root 用户 创建一个新 目录 时，umask 被设置为 022 (rwxr-xr-x)，目录的基本权限被设置为 777 (rwxrwxrwx)。这会使默认权限为 755 (rwxr-xr-x)。

	符号	数值
基本权限	rwxrwxrwx	777
Umask	rwxr-xr-x	022
默认权限	rwxr-xr-x	755

这意味着目录所有者可以列出目录的内容，并可以在该目录下（以及其子目录）中创建、删除和编辑项。这个组群和其它只能列出该目录的内容并将其下移。

例 9.5. 文件的默认权限

当 标准用户 或 root 用户 创建一个新 文件 时，umask 被设置为 022 (rwxr-xr-x)，文件的基本权限被设置为 666 (rw-rw-rw-)。这会使默认权限为 644 (-rw-r--r-)。

	符号	数值
--	----	----

基本权限	rw-rw-rw-	666
Umask	rw-r--r--	022
默认权限	rw-r--r--	644

这意味着，文件所有者可以读取和编辑文件，而组和其它用户只能读取该文件。



注意

出于安全考虑，常规文件默认没有执行权限，即使umask 设为 000 (rwxrwxrwx)。但是，创建的目录可以具有执行权限。

9.1.4. 使用符号值更改文件权限

您可以使用带有符号值（字母和符号的组合）的 **chmod** 工具来更改文件或目录的文件权限。

您可以分配以下 权限：

- 读(r)
- 写(w)
- 执行(x)

权限可分配给以下 所有权级别：

- 用户所有者(u)
- 组所有者(g)
- 其他(o)
- 所有(a)

要添加或删除权限，您可以使用以下 符号：

- + 在现有权限之上添加权限
- - 从现有权限中删除权限
- = 删除现有权限，并明确定义新权限

流程

- 要更改文件或目录的权限，请使用：

```
$ chmod <level><operation><permission> file-name
```

将 <level> 替换为您要为其设置权限的 所有权级别。将 <operation> 替换为其中一个 符号。将 <permission> 替换为您要分配的 权限。用文件或目录的名称替换 file-name。例如，要为每个人授予读、写和执行(rwx) my-script.sh 的权限，请使用 **chmod a=rwx my-script.sh** 命令。

如需了解更多详细信息，请参阅 [基本文件权限](#)。

验证

- 要查看特定文件的权限，请使用：

```
$ ls -l file-name
```

用文件名替换 file-name。

- 要查看特定目录的权限，请使用：

```
$ ls -dl directory-name
```

使用目录名替换 directory-name。

- 要查看特定目录中所有文件的权限，请使用：

```
$ ls -l directory-name
```

使用目录名替换 directory-name。

例 9.6. 更改文件和目录的权限

- 要将 **my-file.txt** 的文件权限从 **-rw-rw-r--** 改为 **-rw-----**，请使用：

- 显示 **my-file.txt** 的当前权限：

```
$ ls -l my-file.txt
-rw-rw-r--. 1 username username 0 Feb 24 17:56 my-file.txt
```

- 从组所有者(**g**)和其他用户(**o**)删除读、写和执行(**rw****x**)文件的权限：

```
$ chmod go= my-file.txt
```

请注意，任何在等号(=)之后没有被指定的权限都会被自动禁止。

- 验证 **my-file.txt** 的权限是否设置正确：

```
$ ls -l my-file.txt
-rw-----. 1 username username 0 Feb 24 17:56 my-file.txt
```

- 要将 **my-directory** 的文件权限从 **drwxrwx---** 改为 **drwxrwxr-x**，请使用：

- 显示 **my-directory** 的当前权限：

```
$ ls -dl my-directory
drwxrwx---. 2 username username 4096 Feb 24 18:12 my-directory
```

- 为所有用户(**a**)添加读和执行(**r-x**)权限：

```
$ chmod o+rx my-directory
```


3. 验证 **my-directory** 及其内容的权限是否设置正确：

```
$ ls -dl my-directory
drwxrwxr-x. 2 username username 4096 Feb 24 18:12 my-directory
```

9.1.5. 使用数值更改文件权限

您可以使用带有八进制（数字）的 **chmod** 工具来更改文件或目录的文件权限。

流程

- 要为现有文件或者目录更改文件权限，请使用：

```
$ chmod octal_value file-name
```

用文件或目录的名称替换 `file-name`。使用数值替换 `octal_value`。如需了解更多详细信息，请参阅 [基本文件权限](#)。

9.2. 管理访问控制列表

每个文件和目录同时只能有一个用户所有者和一个组所有者。如果您要授予用户权限访问属于不同用户或组的特定文件或目录，同时保持其他文件和目录私有，则您可以使用 Linux 访问控制列表(ACL)。

9.2.1. 设置访问控制列表

您可以使用 **setfacl** 工具为文件或目录设置 ACL。

先决条件

- 有 **root** 访问权限。

流程

- 要显示特定文件或目录的当前 ACL，请运行：

```
$ getfacl file-name
```

用文件或目录的名称替换 `file-name`。

- 要为文件或目录设置 ACL，请使用：

```
# setfacl -m u:username:symbolic_value file-name
```

使用用户名替换 `username`，使用符号值替换 `symbolic_value`，使用文件或目录的名称替换 `file-name`。如需更多信息，请参阅系统中的 **setfacl** man page。

例 9.7. 修改组项目的权限

以下示例描述了如何修改属于 **root** 组的 **root** 用户拥有的 **group-project** 文件的权限，以便使该文件：

- 不能被任何人执行。
- 用户 **andrew** 有 **rw-** 权限。
- 用户 **susan** 有 **---** 权限。
- 其他用户有 **r--** 权限。

流程

```
# setfacl -m u:andrew:rw- group-project
# setfacl -m u:susan:--- group-project
```

验证

- 要验证用户 **andrew** 有 **rw-** 权限，用户 **susan** 有 **---** 权限，其他用户有 **r--** 权限，使用：

```
$ getfacl group-project
```

输出会返回：

```
# file: group-project
# owner: root
# group: root
user:andrew:rw-
user:susan:---
group::r--
mask::rw-
other::r--
```

9.3. 管理 UMASK

您可以使用 **umask** 工具显示、设置或更改 **umask** 的当前或默认值。

9.3.1. 显示 **umask** 的当前值

您可以使用 **umask** 工具以符号或数值模式显示 **umask** 的当前值。

流程

- 要在符号模式下显示 **umask** 的当前值，请使用：

```
$ umask -S
```

- 要在八进制模式下显示 **umask** 的当前值，请使用：

```
$ umask
```



注意

以八进制模式显示 umask 时，您可以注意到它显示了四位数字（**0002** 或 **0022**）。umask 的第一个数字代表一个特殊的位（spicky 位、SGID 位或 SUID 位）。如果第一个数字设定为 **0**，则代表没有设置特殊位。

9.3.2. 使用符号值设置 umask

您可以使用 **umask** 工具及符号值（字母和符号组合）来为当前的 shell 会话设置 umask

您可以分配以下 权限：

- 读(r)
- 写(w)
- 执行(x)

权限可分配给以下 所有权级别：

- 用户所有者(u)
- 组所有者(g)
- 其他(o)
- 所有(a)

要添加或删除权限，您可以使用以下 符号：

- + 在现有权限之上添加权限
- - 从现有权限中删除权限
- = 删除现有权限，并明确定义新权限



注意

任何在等号(=)后未指定的权限都将被自动禁止。

流程

- 要为当前的 shell 会话设置 umask，请使用：

```
$ umask -S <level><operation><permission>
```

将 **<level>** 替换为您要为其设置 umask 的 [所有权级别](#)。将 **<operation>** 替换为其中一个 [符号](#)。将 **<permission>** 替换为您要分配的 [权限](#)。例如，要将 umask 设为 **u=rwx,g=rwx,o=rwx**，使用 **umask -S a=rwx**。

如需了解更多详细信息，请参阅 [用户文件创建模式](#)。



注意

umask 仅对当前 shell 会话有效。

9.3.3. 使用数值设置 `umask`

您可以使用 **umask** 工具和八进制值（数字）来为当前 shell 会话设置 `umask`。

流程

- 要为当前的 shell 会话设置 `umask`，请使用：

```
$ umask octal_value
```

使用数值替换 `octal_value`。如需了解更多详细信息，请参阅 [用户文件创建模式掩码](#)。



注意

`umask` 仅对当前 shell 会话有效。

9.3.4. 更改非登录 shell 的默认 `umask`

您可以通过修改 `/etc/bashrc` 文件来更改标准用户的默认 **bash** `umask`。

先决条件

- 有 **root** 访问权限。

流程

1. 在编辑器中打开 `/etc/bashrc` 文件。

将 `umask` 的默认值（**002**）替换为另一个数值。如需了解更多详细信息，请参阅 [用户文件创建模式掩码](#)。

1. 保存更改并退出编辑器。

9.3.5. 更改登录 shell 的默认 `umask`

您可以通过修改 `/etc/login.defs` 文件来更改 **root** 用户的默认 **bash** `umask`。

先决条件

- **root** 访问权限

流程

1. 以 **root** 用户身份，在编辑器中打开 `/etc/login.defs` 文件。
2. 修改以下部分以设置新的默认 **bash** `umask`：

```
# Default initial "umask" value used by login(1) on non-PAM enabled systems.
# Default "umask" value for pam_umask(8) on PAM enabled systems.
# UMASK is also used by useradd(8) and newusers(8) to set the mode for new
# home directories if HOME_MODE is not set.
# 022 is the default value, but 027, or even 077, could be considered
# for increased privacy. There is no One True Answer here: each sysadmin
```

```
# must make up their mind.
```

```
UMASK      022
```

将 `umask` 的数值 (**022**) 替换为另一个数值。如需了解更多详细信息, 请参阅 [用户文件创建模式掩码](#)。

3. 保存更改并退出编辑器。

9.3.6. 更改特定用户的默认 `umask`

您可以通过修改用户的 `.bashrc` 来更改特定用户的默认 `umask`。

流程

- 将指定 `umask` 的八进制值的行追加到特定用户的 `.bashrc` 文件中。

```
$ echo 'umask octal_value' >> /home/username/.bashrc
```

使用数值替换 `octal_value`, 并使用用户名替换 `username`。如需了解更多详细信息, 请参阅 [用户文件创建模式掩码](#)。

9.3.7. 为新创建的主目录设置默认权限

您可以通过修改 `/etc/login.defs` 文件来更改新创建的用户的主目录的权限模式。

流程

1. 以 **root** 用户身份, 在编辑器中打开 `/etc/login.defs` 文件。
2. 修改以下部分来设置新的默认 `HOME_MODE` :

```
# HOME_MODE is used by useradd(8) and newusers(8) to set the mode for new
# home directories.
# If HOME_MODE is not set, the value of UMASK is used to create the mode.
HOME_MODE      0700
```

将默认的八进制值(**0700**)替换为另一个八进制值。所选模式将用于为主目录创建权限。

3. 如果设置了 `HOME_MODE`, 请保存更改并退出编辑器。
4. 如果没有设置 `HOME_MODE`, 请修改 `UMASK` 来为新创建的主目录设置模式 :

```
# Default initial "umask" value used by login(1) on non-PAM enabled systems.
# Default "umask" value for pam_umask(8) on PAM enabled systems.
# UMASK is also used by useradd(8) and newusers(8) to set the mode for new
# home directories if HOME_MODE is not set.
# 022 is the default value, but 027, or even 077, could be considered
# for increased privacy. There is no One True Answer here: each sysadmin
# must make up their mind.
```

```
UMASK      022
```

← 默认值 = 掩码值 - 其他值 = 最终值 | 八进制值 | 八进制值 - 八进制值 = 八进制值 | 八进制值 - 八进制值 = 八进制值

将默认的八进制值(**022**)替换为另一个八进制值。如需了解更多信息，请参阅 [用户文件创建模式掩码](#)。

5. 保存更改并退出编辑器。

第 10 章 管理 SYSTEMD

作为系统管理员，您可以使用 **systemd** 管理系统的方面。充当 Linux 操作系统的系统和服务管理器的 **systemd** 软件套件，提供用于控制、报告和系统初始化的工具和服务。**systemd** 的主要功能包括：

- 在启动过程中并行启动系统服务
- 按需激活守护进程
- 基于依赖项的服务控制逻辑

systemd 管理的基本对象是 **systemd** 单元，表示系统资源和服务。**systemd** 单元由一个名称、类型和一个定义和管理特定任务的配置文件组成。您可以使用单元文件来配置系统行为。请参阅以下各种 **systemd** 单元类型示例：

服务

控制和管理单个系统服务。

目标

表示定义系统状态的一组单元。

设备

管理硬件设备及其可用性。

Mount

处理文件系统挂载。

计时器

规划任务以在特定间隔运行。

10.1. SYSTEMD 单元文件位置

您可以在以下目录中找到单元配置文件：

表 10.1. **systemd** 单元文件位置

目录	描述
/usr/lib/systemd/system/	与安装的 RPM 软件包一起分发的 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/	使用 systemctl enable 命令创建的 systemd 单元文件，以及添加的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。

systemd 的默认配置在编译过程中定义，您可以在 **/etc/systemd/system.conf** 文件中找到配置。通过编辑此文件，您可以通过全局覆盖 **systemd** 单元的值来修改默认配置。

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

10.2. 使用 SYSTEMCTL 管理系统服务

作为系统管理员，您可以使用 **systemctl** 工具管理系统服务。您可以执行各种任务，如启动、停止、重启运行的服务、启用和禁用服务以在引导时启动、列出可用的服务以及显示系统服务状态。

10.2.1. 列出系统服务

您可以列出所有当前载入的服务单元，并显示所有可用服务单元的状态。

流程

使用 **systemctl** 命令执行以下任务：

- 列出所有当前载入的服务单元：

```
$ 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-d.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, or a 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'
```

默认情况下，**systemctl list-units** 命令只显示活跃的单位。对于每个服务单元文件，命令提供以下参数的概述：

UNIT

服务单元的全名

LOAD

配置文件的负载状态

ACTIVE 或 SUB

当前高级别和低级单元文件激活状态

DESCRIPTION

单元目的和功能的简短描述

- 使用带有 **--all** 或 **-a** 命令行选项的命令，列出 **所有载入的单元，而不考虑其状态**：

```
$ systemctl list-units --type service --all
```

- 列出所有可用服务单元的状态(**enabled** 或 **disabled**)：

```
$ systemctl list-unit-files --type service
```



```
UNIT FILE                                STATE
abrt-ccpp.service                       enabled
abrt-oops.service                       enabled
abrt-d.service                           enabled
...
wpa_supplicant.service                  disabled
ypbind.service                          disabled

208 unit files listed.
```

对于每个服务单元，这个命令会显示：

UNIT FILE

服务单元的全名

STATE

服务单元是否已启用或禁用，以便在引导时自动启动的的信息

其他资源

- [显示系统服务状态](#)

10.2.2. 显示系统服务状态

您可以检查任何服务单元以获取详细信息，并验证服务的状态，其是否已启用为在引导时启动还是当前正在运行。您还可以查看在特定的服务单元之后或之前启动的服务。

流程

- 显示与系统服务相应的服务单元的详细信息：

```
$ systemctl status <name>.service
```

将 **<name>** 替换为您要检查的服务单元的名称（例如：**gdm**）。

这个命令显示以下信息：

- 所选服务单元的名称，后跟一个简短描述
- [可用服务单元信息](#) 中描述的一个或多个字段
- 服务单元的执行：如果单元由 **root** 用户执行
- 最新的日志条目

表 10.2. 可用的服务单元信息

项	描述
Loaded	是否服务单元已载入的信息，单元文件的绝对路径，以及是否在引导时启动该单元的说明。
Active	服务单元是否在运行的信息，后面有一个时间戳。

项	描述
Main PID	进程 ID 和相应系统服务的名称。
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	有关相关的控制组(cgroups)的额外信息。

- 验证特定的服务单元是否正在运行：

```
$ systemctl is-active <name>.service
```

- 确定是否在引导时启用了特定的服务单元：

```
$ systemctl is-enabled <name>.service
```



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** 命令都会返回 **0** 退出状态。

- 检查在指定服务单元前 **systemd** 启动哪些服务

```
# systemctl list-dependencies --after <name>.service
```

例如，要查看在 **gdm** 之前启动的服务的列表，请输入：

```
# 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]
```

- 检查在指定服务单元后 **systemd** 启动哪些服务：

```
# systemctl list-dependencies --before <name>.service
```

例如，要查看在 **gdm** 后 **systemd** 启动的服务列表，请输入：

```
# 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

```

其他资源

- [列出系统服务](#)

10.2.3. 启动和停止 systemd 单元

您可以使用 **systemctl start** 命令在当前会话中启动系统服务。

先决条件

- 您有 Root 访问权限。

流程

- 在当前会话中启动一个系统服务：

```
# *systemctl start <systemd_unit> *
```

将 **<systemd_unit>** 替换为您要启动的服务单元的名称（例如 **httpd.service**）。

注意

在 **systemd** 中，服务之间存在正和负的依赖项。启动一个特定的服务可能需要启动一个或多个其他服务（**正依赖项**）或停止一个或多个服务（**负依赖项**）。

当您尝试启动一个新服务时，**systemd** 会自动解析所有依赖项，而不明确通知用户。这意味着，如果您已运行了一个服务，并且您尝试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，如果您正在运行 **sendmail** 服务，并且您试图启动 **postfix** 服务，**systemd** 首先自动停止 **sendmail**，因为这些服务会冲突，且无法在同一端口上运行。

其他资源

- 系统中的 **systemctl (1)** 手册页
- [启用一个系统服务，以便在引导时启动](#)
- [显示系统服务状态](#)

10.2.4. 停止一个系统服务

如果要在当前会话中停止系统服务，请使用 **systemctl stop** 命令。

先决条件

- 根访问权限

流程

- 停止一个系统服务：

```
# systemctl stop <name>.service
```

将 **<name>** 替换为您要停止的服务单元的名称（例如：**bluetooth**）。

其他资源

- 系统中的 **systemctl (1)** 手册页
- [禁用一个系统服务在引导时启动](#)
- [显示系统服务状态](#)

10.2.5. 重启并重新加载一个系统服务

您可以使用 **restart** 命令在当前会话中重启系统服务，以执行以下操作：

- 在当前会话中停止所选服务单元，并立即再次启动它。
- 仅在相应的服务已在运行时重启服务单元。
- 重新加载系统服务的配置，而不中断其执行。

先决条件

- 您有 Root 访问权限。

流程

- 重启一个系统服务：

```
# systemctl restart <name>.service
```

使用您要重启的服务单元的名称替换 **<name>**（例如 **httpd**）。

如果所选服务单元没有运行，这个命令会启动它。

- 仅在相应的服务已在运行时重启服务单元：

```
# systemctl try-restart <name>.service
```

- 重新载入配置而不中断服务执行：

```
# systemctl reload <name>.service
```



注意

不支持此功能的系统服务忽略此命令。要重新启动这些服务，请改为使用 **reload-or-restart** 和 **reload-or-try-restart** 命令。

其他资源

- 您系统上的 **systemctl** 手册页
- [显示系统服务状态](#)

10.2.6. 使系统服务在引导时启动

您可以使服务在引导时自动启动，这些更改将在下次重启时应用。

先决条件

- 您有 Root 访问权限。

流程

- 验证单元是否已屏蔽：

```
# systemctl status <systemd_unit>
```

- 如果这个单元被屏蔽，请首先取消屏蔽它：

```
# systemctl unmask <systemd_unit>
```

- 在引导时启用服务：

```
# systemctl enable <systemd_unit>
```

将 `<systemd_unit>` 替换为您要启用的服务单元的名称（例如 `httpd`）。

（可选）将 `now` 选项传给命令，以便立即启动该单元。

其他资源

- 系统中 **systemctl (1)** 手册页
- [显示系统服务状态](#)
- [启动一个系统服务](#)

10.2.7. 禁止一个系统服务在引导时启动

您可以防止服务单元在引导时自动启动。如果您禁用某个服务，它不会在引导时启动，但可以手动启动。您还可以屏蔽服务，使其无法手动启动。屏蔽是一种禁用服务的方法，使该服务能够永久不可用，直到再次屏蔽该服务。

先决条件

- 您有 Root 访问权限。

流程

- 禁用要在引导时启动的服务：

```
# systemctl disable <name>.service
```

将 **<name>** 替换为您要禁用的服务单元的名称（例如：**bluetooth**）。（可选）传递 **now** 命令，以便在服务当前正在运行时同时停止该服务。

- 可选：要防止这个单元被管理员意外启动，或者作为其它单元的依赖项启动，请屏蔽该服务：

```
# systemctl mask <name>.service
```

其他资源

- 系统中 **systemctl** (1) 手册页
- [显示系统服务状态](#)
- [停止一个系统服务](#)

10.3. 引导至目标系统状态

作为系统管理员，您可以控制系统的引导过程，并定义您希望系统引导进入的状态。这称为 **systemd** 目标，它是您的系统启动以达到特定级别功能的一组 **systemd** 单元。在使用 **systemd** 目标时，您可以查看默认目标，选择运行时的一个目标，更改默认引导目标，引导到紧急或救援目标。

10.3.1. 目标单元文件

systemd 中的目标是一组相关的单元，在系统开始期间充当同步点。目标单元文件以 **.target** 文件扩展名结尾，代表 **systemd** 目标。目标单元的目的是通过一组依赖项将各种 **systemd** 单元分组到一起。

考虑以下示例：

- 同样，**multi-user.target** 单元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。

您可以将以下 **systemd** 目标设置为默认或当前目标：

表 10.3. 常见 **systemd** 目标

救援	在基本系统中拉取的单元目标，并生成一个救援 shell
多用户	用于设置多用户系统的单元目标
图形化	用于设置图形登录屏幕的单元目标
紧急	在主控制台上启动紧急 shell 的单元目标

其他资源

- 您系统上的 **systemd.special (7)** 和 **systemd.target (5)** 手册页

10.3.2. 更改要引导到的默认目标

default.target 符号链接指的是系统应引导到的 systemd 目标。当系统启动时，systemd 会解析此链接并引导至定义的目标。您可以在 **/etc/systemd/system/default.target** 文件中找到当前所选的默认目标单元。每个目标代表某一个功能级别，用于对其他单元进行分组。另外，目标单元会在引导过程中充当同步点。您可以更改系统引导到的默认目标。当您设置默认目标单元时，当前目标将保持不变，直到下次重启为止。

先决条件

- 您有 Root 访问权限。

流程

1. 确定使用当前的默认目标单元 **systemd** 来启动系统：

```
# systemctl get-default
graphical.target
```

2. 列出当前载入的目标：

```
# systemctl list-units --type target
```

3. 将系统配置为默认使用不同的目标单元：

```
# systemctl set-default <name>.target
```

将 **<name>** 替换为您要默认使用的目标单元的名称。

Example:

```
# systemctl set-default multi-user.target
Removed /etc/systemd/system/default.target
Created symlink /etc/systemd/system/default.target -> /usr/lib/systemd/system/multi-user.target
```

4. 验证默认目标单元：

```
# systemctl get-default
multi-user.target
```

5. 可选：切换到新的默认目标：

```
# systemctl isolate default.target
```

或者，重启系统。

其他资源

- **systemctl (1)**, **systemd.special (7)**, 和 **bootup (7)** man page

10.3.3. 更改当前目标

在运行的系统商，您可以在不重启的情况下更改当前启动中的目标单元。如果您切换到不同的目标，**systemd** 会启动所有服务及其这个目标需要的依赖项，并停止新目标没有启用的所有服务。手动切换到其他目标只是临时操作。重启主机时，**systemd** 会再次引导到默认目标。

流程

1. 可选：显示您可以选择的目标列表：

```
# systemctl list-units --type target
```



注意

您只能隔离单元文件中设置了 **AllowIsolate=yes** 选项的目标。

2. 在当前引导中切换到不同的目标单元：

```
# systemctl isolate <name>.target
```

将 <name> 替换为您要在当前引导中使用的目标单元的名称。

Example:

```
# systemctl isolate multi-user.target
```

这个命令启动名为 **multi-user** 的目标单元和所有依赖单元，并立即停止所有其他单元。

其他资源

- 系统中的 **systemctl (1)** 手册页

10.3.4. 引导至救援模式

您可以引导到救援模式，其提供单用户环境的以便在系统无法进入到后续目标，且常规引导过程失败时进行故障排除或修复。在救援模式下，系统会尝试挂载所有本地文件系统，并启动某些重要的系统服务，但不会激活网络接口。

先决条件

- 根访问权限

流程

- 要进入救援模式，在当前会话中更改当前目标：

```
# systemctl rescue
```

```
Broadcast message from root@localhost on pts/0 (Fri 2023-03-24 18:23:15 CEST):
```

```
The system is going down to rescue mode NOW!
```




注意

这个命令与 **systemctl isolate rescue.target** 类似，但它也会向当前登录到该系统的所有用户发送信息性消息。

要防止 **systemd** 发送信息，使用 **--no-wall** 命令行选项输入以下命令：

```
# systemctl --no-wall rescue
```

故障排除

如果您的系统无法进入救援模式，您可以引导至紧急模式，其提供尽可能小的环境。在紧急模式下，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。

10.3.5. 引导过程故障排除

作为系统管理员，您可以在引导时选择非默认目标来对引导过程进行故障排除。在引导时更改目标仅影响单个引导。您可以引导到紧急模式，它提供了尽可能小的环境。

流程

1. 重启系统，通过按 **Enter** 键之外的任意键中断引导装载程序菜单倒计时，这将发起一个正常的启动。
2. 将光标移至要启动的内核条目。
3. 按 **E** 键编辑当前条目。
4. 移动到以 **linux** 开头的行的末尾，然后按 **Ctrl+E** 跳到行尾：

```
linux ($root)/vmlinuz-5.14.0-70.22.1.el9_0.x86_64 root=/dev/mapper/rhel-root ro crashl
kernel=auto resume=/dev/mapper/rhel-swap rd.lvm.lv/swap rhgb quiet
```

5. 要选择备用引导目标，请将 **systemd.unit=** 参数附加到 **linux** 开头的行的末尾：

```
linux ($root)/vmlinuz-5.14.0-70.22.1.el9_0.x86_64 root=/dev/mapper/rhel-root ro crashl
kernel=auto resume=/dev/mapper/rhel-swap rd.lvm.lv/swap rhgb quiet
systemd.unit=<name>.target
```

将 **<name>** 替换为您要使用的目标单元的名称。例如：**systemd.unit=emergency.target**

6. 按 **Ctrl+X** 使用这些设置引导。

10.4. 关闭、挂起和休眠系统

作为系统管理员，您可以使用不同的电源管理选项来管理功耗，执行正确的关机以确保所有数据都被保存，或者重启系统以应用更改和更新。

10.4.1. 系统关闭

要关闭系统，您可以直接使用 **systemctl** 工具，或者通过 **shutdown** 命令来调用这个工具。

使用 **shutdown** 工具有以下优点：

- 在 RHEL 8 中，您可以使用时间参数调度关闭。这也为用户提供了系统已计划关机的警告。

10.4.2. 计划一个系统关机

作为系统管理员，您可以计划一个延迟关闭，为用户提供时间来保存其工作并注销系统。使用 **shutdown** 命令执行以下操作：

- 关闭系统，并在特定时间关闭机器：

```
# shutdown --poweroff hh:mm
```

其中 **hh:mm** 是 24 小时表示法时间。为防止新的登录，在系统关闭前 5 分钟会创建 **/run/nologin** 文件。

当使用时间参数时，您可以通过指定可选的 **wall message** 来通知登录到计划关机的系统，如 **shutdown --poweroff 13:59 "Attention. 系统将于 13:59" 关闭**。

- 在延迟后关闭和停止系统，而不断电：

```
# shutdown --halt +m
```

其中 **+m** 是延迟时间（以分钟为单位）。您可以使用 **now** 关键字作为 **+0** 的别名。

- 取消一个待处理的关机

```
# shutdown -c
```

其他资源

- [shutdown\(8\) 手册页](#)

-

使用 **systemctl** 命令来关闭系统

10.4.3. 使用 systemctl 命令来关闭系统

作为系统管理员，您可以关闭系统并关闭机器，或使用 **systemctl** 命令关闭和停止系统，而不断电。

先决条件

-

根访问权限

流程

使用 **systemctl** 命令执行以下任务：

-

关闭系统并给机器断电：

```
# systemctl poweroff
```

-

在断电的情况下关闭和停止系统：

```
# systemctl halt
```



注意

默认情况下，运行其中任何一个命令都可让 **systemd** 向当前登录到系统的所有用户发送一条通知性消息。要防止 **systemd** 发送这条消息，请使用 **--no-wall** 命令行选项运行所选命令。

10.4.4. 重启系统

当您重启系统时，**systemd** 会停止所有正在运行的程序和服务，系统关闭，然后立即启动。

先决条件

- 您有 **Root** 访问权限。

流程

- 重启系统：

```
# systemctl reboot
```



注意

默认情况下，当您使用此命令时，**systemd** 会向当前登录到系统的所有用户发送一条通知性消息。要防止 **systemd** 发送这条消息，请使用 **--no-wall** 选项运行这个命令。

10.4.5. 通过挂起和休眠系统来优化功耗

作为系统管理员，您可以管理功耗，在系统上节省能源，并保留系统的当前状态。要做到这一点，请应用以下模式之一：

- 挂起
- 休眠
- Hybrid Sleep
- 挂起-然后-休眠

先决条件

- 您有 **Root** 访问权限。

流程

选择适当的节能方法：

-

暂停 Suspend ing 将系统状态保存在 RAM 中，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，而不是保存在硬盘上，因此，从挂起模式恢复系统比从休眠模式恢复要快得多。但是，暂停的系统状态也会受到断电的影响。要挂起系统，请运行：

```
# systemctl suspend
```

-

Hibernate Hibernating 在硬盘中保存系统状态，并关闭机器。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上，而不是保存在 RAM 中，因此机器不必保持对 RAM 模块的供电。但是，因此，从休眠模式恢复系统要比将其恢复为挂起模式恢复要慢得多。要休眠系统，请运行：

```
# systemctl hibernate
```

-

混合睡眠状态 组合了休眠和暂停的元素。系统首先在硬盘上保存当前状态，并进入类似挂起的低电源状态，这使系统可以更快地恢复。混合睡眠的好处是，如果系统在睡眠状态下掉电，它仍然可以从硬盘上保存的镜像恢复之前的状态，类似于休眠。要休眠并挂起系统，请运行：

```
# systemctl hybrid-sleep
```

-

suspend -then-hibernate 此模式首先挂起系统，这会将当前系统状态保存到 RAM，并将系统置于低电源模式。如果系统保持挂起一段时间，则系统会休眠，您可以在 `HibernateDelaySec` 参数中定义。休眠将系统状态保存到硬盘上，并完全关闭系统。挂起-然后休眠模式提供保留电池电源的好处，而您仍能够快速恢复工作。另外，这个模式可确保在出现电源故障时保存您的数据。挂起，然后休眠系统：

```
# systemctl suspend-then-hibernate
```

10.4.6. 更改电源按钮行为

当按计算机上的 **power** 按钮时，它默认挂起或关闭系统。您可以根据您的偏好自定义此行为。

10.4.6.1. 在按按钮且 GNOME 未运行时更改电源按钮的行为

当您在非图形 **systemd** 目标中按 **power** 按钮时，它默认关闭系统。您可以根据您的偏好自定义此行为。

先决条件

- **管理访问权限。**

流程

1. **编辑 `/etc/systemd/logind.conf` 配置文件，并将 `HandlePowerKey=poweroff` 变量设置为以下选项之一：**

`poweroff`

关闭计算机。

`reboot`

重启系统。

`halt`

发起系统停止。

`kexec`

发起 `kexec` 重启。

`suspend`

挂起系统。

`hibernate`

发起系统休眠。

`ignore`

什么都不做。

例如，要在按下电源按钮时重启系统，请使用这个设置：

```
HandlePowerKey=reboot
```

10.4.6.2. 在按按钮和 GNOME 运行时更改电源按钮的行为

在图形登录屏幕或在图形用户会话中，按 power 按钮默认挂起机器。当用户物理按下 power 按钮或从远程控制台按下虚拟 power 按钮时，才会出现这种情况。您可以选择不同的 power 按钮行为。

流程

1.

在 `/etc/dconf/db/local.d/01-power` 文件中为系统范围的设置创建一个本地数据库，其内容如下：

```
[org/gnome/settings-daemon/plugins/power]
power-button-action=<value>
```

使用以下电源按钮操作之一替换 `< value >`：

nothing

什么都不做。

suspend

挂起系统。

hibernate

休眠系统。

interactive

显示一个弹出窗口查询，询问用户要做什么。

交互模式下，按下 power 按钮后，系统在 60 秒后自动关闭。但是，您可以从弹出查询中选择不同的行为。

2.

可选：覆盖用户设置，并阻止用户更改它。在 `/etc/dconf/db/local.d/locks/01-power` 文件中输入以下配置：

■

```
/org/gnome/settings-daemon/plugins/power/power-button-action
```

3.

更新系统数据库：

```
# dconf update
```

4.

注销并重新登录，以使系统范围的设置生效。

第 11 章 配置时间同步

在 IT 环境中保持准确的时间非常重要。所有网络设备间的一致时间可以提高日志文件的可追溯性，某些协议依赖于同步时钟。例如，Kerberos 使用时间戳来防止重放攻击。用户空间守护进程更新内核中运行的系统时钟。从 Red Hat Enterprise Linux 8 开始，NTP 协议由 `chronyd` 守护进程实现，它可从 `chrony` 软件包的软件仓库中获得。

11.1. CHRONY 套件介绍

网络时间协议(NTP)的实现是 `chrony`。您可以使用 `chrony`：

- 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 GPS 接收器
- 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

在各种条件中 `chrony` 执行良好：

- 包括网络连接间歇性
- 大量嵌套的网络
- 更改温度（普通计算机时钟对温度敏感）
- 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将两台计算机之间的准确性提高到子微秒级。

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程序）。

chronyd 守护进程可以通过命令行工具 **chronyc** 监控和控制。这个工具提供了一个命令提示，允许输入大量命令来查询 **chronyd** 的当前状态并修改其配置。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。

11.2. 使用 CHRONYC 来控制 CHRONYD

您可以使用 **chronyc** 命令行工具控制 **chronyd**。

流程

1. 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>
```

2. 要列出所有的命令，请输入 **help**。
3. 或者，如果与以下命令一同调用，该工具也可以在非交互命令模式下调用：

```
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**。

11.3. 使用 CHRONY

以下小节介绍了如何启动和停止 **chronyd**，以及如何检查 **chrony** 是否同步。这些章节还介绍了如何手动调整系统时钟。

11.3.1. 管理 chrony

您可以启动、停止并检查 **chronyd** 的状态。

1.

chrony 在 Red Hat Enterprise Linux 被默认安装。以 root 用户运行以下命令进行验证：

```
# dnf 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**。

2.

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
```

3.

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
```

4.

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
```

11.3.2. 检查是否同步 chrony

您可以检查 **chrony** 是否与跟踪、源和 **sourcestats** 命令的使用同步。

流程

1.

要检查 **chrony** 跟踪，请输入：

```
$ chronyc tracking
Reference ID   : CB00710F (ntp-server.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
```

2.

chronyc sources 命令显示 **chronyd** 正在访问的当前时间源的信息。

```
$ 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
```

您可以指定可选的 **-v** 参数来打印更详细的信息。在这种情况下，会输出额外的标头行显示字段含义的信息。

3.

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。要检查 **chrony** 源的统计信息，请运行以下命令：

```
$ 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
```

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。

其他资源

- [系统中的 chronyc \(1\) 手册页](#)

11.3.3. 手动调整系统时钟

您可以手动调整系统时钟。

流程

- 要立即调整系统时钟，绕过 **slewing** 的任何调整，请输入：

```
# chronyc makestep
```

重要

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** 测量实时时钟漂移速率的需要。

11.3.4. 禁用 chrony 分配程序脚本

chrony 分配程序脚本管理 **NTP** 服务器的在线和离线状态。作为系统管理员，您可以禁用分配程序脚

本，以使 **chronyd** 持续轮询服务器。

NetworkManager 在接口重新配置、停止或启动操作过程中执行 **chrony** 分配程序脚本。但是，如果您在 **NetworkManager** 之外配置某些接口或路由，则您可能会遇到以下情况：

1. 当没有到 **NTP** 服务器的路由时，分配程序脚本可能会运行，导致 **NTP** 服务器切换到离线状态。
2. 如果您稍后建立路由，脚本默认不会再次运行，**NTP** 服务器保持在离线状态。

要确保 **chronyd** 可以与您的 **NTP** 服务器同步（其有单独的受管接口），请禁用分配程序脚本。

流程

- 要禁用 **chrony** 分配程序脚本，请创建一个到 **/dev/null** 的符号链接：

```
# ln -f -s /dev/null /etc/NetworkManager/dispatcher.d/20-chrony-onoffline
```



注意

更改后，**NTP** 服务器始终保持在线状态。

11.3.5. 在隔离的网络中设置 **chrony**

对于从未连接到互联网的网络，可以将一台计算机选为主计时服务器。其他计算机要么是服务器的直接客户端，要么是客户端的客户端。在服务器上，必须使用系统时钟的平均偏移率手动设置 **drift** 文件。如果服务器重启了，它将从周围的系统获取时间，并计算一个平均值来设置系统时钟。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。

要在隔离的网络中为系统设置 **chrony**，请按照以下步骤操作：

流程

1.

在选择成为服务器的系统上，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 <subnet>
```

其中 `<subnet>` 是允许客户端连接的网络。使用无类别域间路由(CIDR)标记来指定子网。

2.

在选择为服务器直接客户端的系统上，编辑 `/etc/chrony.conf`，如下所示：

```
server <server_fqdn>
driftfile /var/lib/chrony/drift
logdir /var/log/chrony
log measurements statistics tracking
keyfile /etc/chrony.keys
commandkey 24
local stratum 10
initstepslew 20 ntp1.example.net
allow <server_ip_address>
```

其中 `<server_fqdn>` 是服务器的主机名，`<server_ip_address>` 是服务器的地址。具有此配置的客户端重启后，其将与服务器重新同步。

在不是服务器直接客户端的客户端系统上，`/etc/chrony.conf` 文件应该是一样的，除了应该省略 `local` 和 `allow` 指令外。

在隔离的网络中，您还可以使用 `local` 指令来启用本地参考模式。该模式可允许 `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。每一个服务器都需要配置为使用 `local` 轮询所有其他服务器。这样可确保只有最小参考 ID 的服务器具有本地参考活跃状态，其他服务器与之同步。当服务器出现故障时，另一台服务器将接管。

11.3.6. 配置远程监控访问

chronyc 工具可以使用以下方法访问 ***chronyd***：

- ***IPv4* 或 *IPv6*。**
- ***域套接字*，可由 *root* 和 *chrony* 用户本地访问。**

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，***chronyc*** 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***
- ***rtcddata***
- ***smoothing***
- ***sources***
- ***sourcestats***
- ***tracking***
- ***waitsync***

chronyd 接受这些命令的主机集合可以使用以下方法配置：

- 您可以在 **chronyd** 的配置文件中 **使用 cmdallow 指令**。
- 在 **chronyc** 中 **运行 cmdallow 命令**。

默认情况下，仅接受来自 localhost（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。

以下流程描述了如何使用 chronyc 远程访问 chronyd。

流程

1. 通过在 **/etc/chrony.conf** 文件中添加以下内容，将 **chrony** 配置为侦听本地接口：

```
bindcmdaddress 0.0.0.0
```

and

```
bindcmdaddress ::
```

2. 允许来自远程 IP 地址、网络和子网的命令：

在 /etc/chrony.conf 文件中添加以下内容：

```
cmdallow 192.168.1.0/24
```

```
cmdallow 2001:db8::/64
```

3.

在防火墙中打开端口 323 以允许来自远程系统的连接：

```
# firewall-cmd --permanent --add-port=323/udp
```

4.

重新载入防火墙配置：

```
# firewall-cmd --reload
```

其他资源

•

系统中 `chrony.conf` (5) 手册页

11.3.7. 使用 RHEL 系统角色管理时间同步

您可以使用 `timesync` 角色在多个目标机器上管理时间同步。`timesync` 角色安装并配置 NTP 或 PTP 实现，作为 NTP 或 PTP 客户端进行操作来同步系统时钟。



警告

`timesync` 角色替换了受管主机上给定或检测到的供应商服务的配置。之前的设置即使没有在角色变量中指定，也会丢失。如果没有定义 `timesync_ntp_provider` 变量，唯一保留的设置就是供应商选择。

以下示例演示了如何在只有一个服务器池的情况下应用 `timesync` 角色。

例 11.1. 为单一服务器池应用 `timesync` 角色的 playbook 示例

```
---
- hosts: timesync-test
  vars:
    timesync_ntp_servers:
      - hostname: 2.rhel.pool.ntp.org
        pool: yes
```

```

iburst: yes
roles:
  - rhel-system-roles.timesync

```

有关 `timesync` 角色变量的详细参考，请安装 `rhel-system-roles` 软件包，并参阅 `/usr/share/doc/rhel-system-roles/timesync` 目录中的 `README.md` 或 `README.html` 文件。

其他资源

- [准备一个控制节点和受管节点以使用 RHEL 系统角色](#)

11.3.8. 其他资源

- [系统中的 `chronyc` \(1\) 和 `chronyd` \(8\) man page](#)
- [常见问题解答](#)

11.4. 带有 HW 时间戳的 CHRONY

某些网络接口控制器(NIC)中的硬件时间戳(HW)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，当启用了 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，可达到次微秒级的准确性。

另一个用于使用硬件时间戳进行时间同步的协议是 PTP

与 NTP 不同，PTP 依赖于网络交换机和路由器。如果要实现同步的最佳准确性，请在带有 PTP 支持的交换机和路由器的网络中使用 PTP，并在没有这样的交换机和路由器的网络上首选 NTP。

11.4.1. 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。

流程

- 显示设备的时间戳功能和关联的 PTP 硬件时钟：

```
# ethtool -T enp1s0
```

11.4.2. 启用硬件时间戳

您可以使用 `/etc/chrony.conf` 文件中的 `hwtimestamp` 指令，在一个或多个接口上启用硬件时间戳。该指令可指定单一接口，也可以指定通配符字符来启用所有支持接口的硬件时间戳。

流程

1. 编辑 `/etc/chrony.conf` 文件并进行以下更改：
 - a. 为支持硬件时间戳的接口添加 `hwtimestamp` 设置。例如：

```
hwtimestamp enp1s0
hwtimestamp eno*
```

如果没有其他应用程序，您可以使用 `*` 通配符，如 `ptp4l` 使用硬件时间戳。

- b. 通过在服务器设置中附加 `minpoll` 和 `maxpoll` 选项来配置简短的客户端轮询间隔，例如：

```
server ntp.example.com local minpoll 0 maxpoll 0
```

对于硬件时间戳，您必须配置一个较短的轮询间隔，而不是默认范围(641024 秒)，以最大程度降低系统时钟的偏移量。

- c. 通过在服务器设置中附加 `xleave` 选项来启用 NTP 交错模式：

```
server ntp.example.com local minpoll 0 maxpoll 0 xleave
```

有了这个设置，`chrony` 仅在发送数据包后获取硬件传输时间戳。这个行为可防止在响

应的数据包中保存时间戳。使用 `xleave` 选项, `chrony` 可以接收传输后生成的传输时间戳。

d.

可选：增加为服务器上客户端访问的日志而分配的最大内存大小，例如：

```
clientloglimit 100000000
```

默认服务器配置允许数以千计的客户端同时使用交错模式。通过增加 `clientloglimit` 设置的值，您可以为大量客户端配置服务器。

2.

重启 `chronyd` 服务：

```
# systemctl restart chronyd
```

验证

1.

可选：在启用了硬件时间的 `/var/log/messages` 日志文件中进行验证：

```
chronyd[4081]: Enabled HW timestamping on enp1s0  
chronyd[4081]: Enabled HW timestamping on eno1
```

2.

如果 `chronyd` 配置为 `NTP` 客户端或 `peer`，显示传输和接收时间戳模式和交集模式：

```
# chronyc ntpdata
```

Output:

```
[literal,subs="+quotes,verbatim,normal"]
```

```
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

```

3.

报告 NTP 测量的稳定性：

```
# chronyc sourcestats
```

Output:

```
[literal,subs="+quotes,verbatim,normal"]
```

```
....
```

```
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
....

```

Std Dev 列中会报告这个稳定性。启用硬件时间戳后，在正常负载下，NTP 测量的稳定性应该以十秒或几百纳秒为单位。

11.4.3. 配置 PTP-NTP 桥接

如果网络中有一个高度准确的精确时间协议(PTP)主时间服务器，但没有 PTP 支持的交换机或路由器，则计算机可能被专用于 PTP 客户端和 stratum-1 NTP 服务器。此类计算机需要有两个或多个网络接口，并且接近主时间服务器或与它直接连接。这样可保证高度准确的网络同步。

流程

1.

从 linuxptp 软件包中配置 ptp4l 和 phc2sys 程序，以使用 PTP 来同步系统时钟。

1.

将 chronyd 配置为使用其他接口提供系统时间：

```
bindaddress 203.0.113.74
```

```
hwtimestamp enp1s0
local stratum 1
```

2.

重启 **chronyd** 服务：

```
# systemctl restart chronyd
```

11.5. CHRONY 中的网络时间安全概述(NTS)

Network Time Security(NTS)是用于**网络时间协议(NTP)**的身份验证机制，旨在扩展大量客户端。它将验证从服务器计算机接收的数据包在移到客户端机器时是否被取消处理。**Network Time Security(NTS)**包含 **Key Establishment(NTS-KE)**协议，该协议会自动创建在服务器及其客户端中使用的加密密钥。



警告

NTS 与 FIPS 和 OSPP 配置文件不兼容。当您启用 **FIPS** 和 **OSPP** 配置文件时，使用 **NTS** 配置的 **chronyd** 可能会中止，并显示一条致命消息。您可以通过将 **GNUTLS_FORCE_FIPS_MODE=0** 设置添加到 **/etc/sysconfig/chronyd** 文件来禁用 **chronyd** 服务的 **OSPP** 配置集和 **FIPS** 模式。

11.5.1. 在客户端上启用网络时间安全(NTS)

默认情况下不启用 **Network Time Security(NTS)**。您可以在 **/etc/chrony.conf** 中启用 **NTS**。为此，请执行以下步骤：

先决条件

•

时间服务器支持 **NTS**。

流程

编辑 **/etc/crony.conf** 文件并进行以下更改：

1.

除推荐的 **iburst** 选项外，使用 **nts** 选项指定服务器。

For example:

```
server time.example.com iburst nts
server nts.netnod.se iburst nts
server ptbtime1.ptb.de iburst nts
```

2.

添加以下设置以避免在系统引导时重复 **Network Time Security-Key Establishment (NTS-KE)** 会话：

```
ntsdumpdir /var/lib/chrony
```

3.

如果存在，注释掉或删除以下设置，以禁用与 **DHCP** 提供的网络时间协议(**NTP**)服务器的同步：

```
sourcedir /run/chrony-dhcp
```

4.

重启 **chronyd** 服务：

```
systemctl restart chronyd
```

验证

•

验证 **NTS** 密钥是否已成功建立：

```
# chronyc -N authdata
```

```
Name/IP address Mode KeyID Type KLen Last Atmp NAK Cook CLen
```

```
=====
```

```
time.example.com NTS 1 15 256 33m 0 0 8 100
```

```
nts.netnod.se NTS 1 15 256 33m 0 0 8 100
```

```
ptbtime1.ptb.de NTS 1 15 256 33m 0 0 8 100
```

KeyID、**Type** 和 **KLen** 应带有非零值。如果该值为零，请检查系统日志中来自 **chronyd** 的错误消息。

•

验证客户端是否正在进行 **NTP** 测量：


```
# chronyc -N sources
```

```
MS Name/IP address Stratum Poll Reach LastRx Last sample
```

```
=====
```

```
time.example.com 3      6 377 45 +355us[ +375us] +/- 11ms
```

```
nts.netnod.se 1      6 377 44 +237us[ +237us] +/- 23ms
```

```
ptbtime1.ptb.de 1      6 377 44 -170us[ -170us] +/- 22ms
```

Reach 列中应具有非零值；理想情况是 377。如果值很少为 377 或永远不是 377，这表示 NTP 请求或响应在网络中丢失。

其他资源

- [系统中 chrony.conf \(5\) 手册页](#)

11.5.2. 在时间服务器上启用网络时间安全(NTS)

如果您运行自己的网络时间协议(NTP)服务器，您可以启用服务器网络时间协议(NTS)支持来促进其客户端安全地同步。

如果 NTP 服务器是其它服务器的客户端，即它不是 Stratum 1 服务器，它应使用 NTS 或对称密钥进行同步。

先决条件

- 以 PEM 格式的服务器私钥
- 带有 PEM 格式的所需中间证书的服务器证书

流程

1. 编辑 `/etc/chrony.conf` 文件并进行以下更改：

```
ntsserverkey /etc/pki/tls/private/<ntp-server.example.net>.key
ntsservercert /etc/pki/tls/certs/<ntp-server.example.net>.crt
```

2.

对私钥和证书文件设置权限，允许 `chrony` 用户读取文件，例如

```
# chown root:chrony /etc/pki/tls/private/<ntp-server.example.net>.key
/etc/pki/tls/certs/<ntp-server.example.net>.crt

# chmod 644 /etc/pki/tls/private/<ntp-server.example.net>.key /etc/pki/tls/certs/<ntp-
server.example.net>.crt
```

3.

确保 `ntsdumpdir /var/lib/chrony` 设置存在。

4.

在 `firewalld` 中打开所需端口：

```
# firewall-cmd --permanent --add-port={323/udp,4460/tcp}
# firewall-cmd --reload
```

5.

重启 `chronyd` 服务：

```
# systemctl restart chronyd
```

验证

1.

从客户端机器执行测试：

```
$ chronyd -Q -t 3 'server

ntp-server.example.net iburst nts maxsamples 1'
2021-09-15T13:45:26Z chronyd version 4.1 starting (+CMDMON +NTP +REFCLOCK
+RTC +PRIVDROP +SCFILTER +SIGND +ASYNCDNS +NTS +SECHASH +IPV6
+DEBUG)
2021-09-15T13:45:26Z Disabled control of system clock
2021-09-15T13:45:28Z System clock wrong by 0.002205 seconds (ignored)
2021-09-15T13:45:28Z chronyd exiting
```

System clock wrong 消息指示 NTP 服务器接受 NTS-KE 连接并使用 NTS 保护的 NTP 消息进行响应。

2.

验证 NTS-KE 连接并验证服务器中观察的 NTP 数据包：

```
# chronyc serverstats
```

```
NTP packets received      : 7
NTP packets dropped       : 0
Command packets received  : 22
Command packets dropped   : 0
Client log records dropped : 0
NTS-KE connections accepted: 1
NTS-KE connections dropped : 0
Authenticated NTP packets: 7
```

如果 **NTS-KE connections accepted** 和 **Authenticated NTP packets** 项带有一个非零值, 这意味着至少有一个客户端能够连接到 **NTS-KE** 端口并发送经过身份验证的 **NTP** 请求。

第 12 章 恢复系统

要使用现有备份来恢复系统，Red Hat Enterprise Linux 提供了一个 **Relax-and-Recover (ReaR)** 程序。

您可以使用这个工具作为灾难恢复解决方案，也用于系统迁移。

该工具可让您执行以下任务：

- 生成可引导镜像，并使用镜像从现有备份中恢复系统。
- 复制原始存储布局。
- 恢复用户和系统文件。
- 将系统还原到不同的硬件中。

另外，对于灾难恢复，您还可以将某些备份软件与 **ReaR** 集成。

12.1. 设置 **REAR** 并手动创建备份

使用以下步骤，使用 **Relax-and-Recover(ReaR)** 工具安装软件包，来创建一个救援系统，配置并生成一个备份。

先决条件

- 根据备份恢复计划完成必要的配置。

请注意：您可以使用 **NETFS** 备份方法，该方法是 **ReaR** 完全整合的、内置的方法。

流程

1. 安装 **ReaR** 工具：

```
# dnf install rear
```

2. 在您选择的编辑器中修改 **ReaR** 配置文件，例如：

```
# vi /etc/rear/local.conf
```

3. 在 **/etc/rear/local.conf** 中添加备份设置详情。例如，在使用 **NETFS** 备份方法时添加以下行：

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

使用备份位置的 **URL** 替换 **backup.location**。

4. 要将 **ReaR** 配置为在创建新归档时保留之前的备份归档，还需要将以下行添加到配置文件中：

```
NETFS_KEEP_OLD_BACKUP_COPY=y
```

5. 要让递增形式进行备份，在每个运行中只备份修改了的文件，添加以下行：

```
BACKUP_TYPE=incremental
```

6. 创建一个救援系统：

```
# rear mkrescue
```

7. 根据恢复计划创建备份。例如，在使用 **NETFS** 备份方法时运行以下命令：

```
# rear mkbackuponly
```

另外，您可以通过运行以下命令来在一个步骤中创建救援系统和备份：

```
# rear mkbbackup
```

该命令将 `rear mkrescue` 和 `rear mkbbackuponly` 的功能组合在一起。

12.2. 在 64 位 IBM Z 架构上使用 REAR 救援镜像

基本 Relax 和 Recover (ReaR) 功能现在包括在 64 位 IBM Z 构架中，自 RHEL 9.2 开始被完全支持。您只能在 z/VM 环境中的 IBM Z 上创建 ReaR 救援镜像。备份和恢复逻辑分区(LPAR)还没有进行测试。

重要

64 位 IBM Z 架构上的 ReaR 仅支持 `rear` 软件包版本 2.6-17.el9 或更高版本。早期版本仅作为技术预览提供。有关红帽技术预览功能支持范围的更多信息，请参阅 <https://access.redhat.com/support/offerings/techpreview>。

当前唯一可用的输出方法是 Initial Program Load(IPL)。IPL 生成一个内核和一个初始 RAM 磁盘 (`initrd`)，可与 zIPL 引导装载程序一起使用。

先决条件

- **ReaR 已安装。**
 - 要安装 ReaR，请运行 `dnf install rear` 命令

流程

将以下变量添加到 `/etc/rear/local.conf` 中来配置 ReaR，以便在 64 位 IBM Z 架构上生成救援镜像：

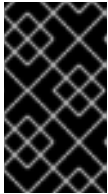
1. 要配置 IPL 输出方法，请添加 `OUTPUT=IPL`。

2.

要配置备份方法和目的地，请添加 **BACKUP** 和 **BACKUP_URL** 变量。例如：

```
BACKUP=NETFS
```

```
BACKUP_URL=nfs://<nfsserver name>/<share path>
```



重要

目前在 64 位 IBM Z 构架上不支持本地备份存储。

3.

可选：您还可以配置 **OUTPUT_URL** 变量来保存内核和 **initrd** 文件。默认情况下，**OUTPUT_URL** 与 **BACKUP_URL** 保持一致。

4.

要执行备份和救援镜像创建：

```
# rear mkbackup
```

5.

这会在 **BACKUP_URL** 或 **OUTPUT_URL**（如果设置了）变量指定的位置创建内核和 **initrd** 文件，并使用指定的备份方法进行备份。

6.

要恢复系统，请使用第 3 步中创建的 **ReaR** 内核和 **initrd** 文件，并从与 **zipl** 引导装载程序、内核和 **initrd** 一起准备的直接附加存储设备(DASD)或者附加光纤通道协议(FCP)的 **SCSI** 设备启动。如需更多信息，请参阅[使用准备的 DASD](#)。

7.

当救援内核和 **initrd** 引导时，它会启动 **ReaR** 救援环境。继续系统恢复。



警告

目前，救援过程会重新格式化连接到系统的所有 **DASD**（直接附加存储设备）。如果系统存储设备中存在任何有价值的数据，则不要尝试系统恢复。这还包括使用 **zipl** 引导装载程序、**ReaR** 内核和用来引导到救援环境的 **initrd** 准备的设备。确保保留一份副本。

其他资源

- [在 z/VM 下安装](#)
- [使用一个准备好的 DASD。](#)

12.3. REAR 排除

ReaR 工具会根据恢复过程中在 `/var/lib/rear/layout/disklayout.conf` 布局文件中的描述重新创建原始系统的存储布局。存储布局包括分区、卷组、逻辑卷、文件系统和其他存储组件。

ReaR 在创建救援镜像时创建布局文件，并在镜像中嵌入这个文件。您还可以使用 `rear savelayout` 命令创建布局文件。这可让您快速创建布局文件并检查它，而无需创建整个救援镜像。

布局文件描述了原始系统的整个存储布局，但有一些例外，因为 **ReaR** 从布局文件中排除一些存储组件，并在恢复过程中重新创建。从布局排除存储组件由以下配置变量控制：

- **AUTOEXCLUDE_DISKS**
- **AUTOEXCLUDE_MULTIPATH**
- **AUTOEXCLUDE_PATH**

EXCLUDE_RECREATE

您可以查看 `/usr/share/rear/conf/default.conf` 文件中的配置变量的默认值，并可以更改本地 `/etc/rear/local.conf` 配置文件中的这些值。

有关布局文件的语法以及可用于排除某些存储组件的配置变量的更多信息，请参阅 *ReaR 用户指南* 中的 *Layout 配置* 章节，该配置以 `/usr/share/doc/rear/relax-and-recover-user-guide.html` 的形式安装。

您还可以配置由内部 **NETFS** 和 **RSYNC** 备份方法备份哪些文件。默认情况下，如果布局文件中包含文件系统，则所有挂载的本地（基于磁盘）文件系统中的文件由 `rear mkbackup` 或 `rear mkbackuponly` 备份。从布局文件中排除某些文件系统，这些文件由

AUTOEXCLUDE_DISKS、**AUTOEXCLUDE_MULTIPATH**、**AUTOEXCLUDE_PATH** 和 **EXCLUDE_RECREATE** 等变量控制，也会从备份中排除其内容。您还可以使用

BACKUP_PROG_EXCLUDE 配置变量，从备份中排除某些文件或目录树，而不将文件系统从布局文件中排除。当以这种方式排除文件系统中的所有文件和目录时，文件系统会在恢复过程中重新创建，但会为空，因为备份不包含要恢复到其中的任何数据。这对包含临时数据且不需要保留的文件系统很有用，或者用于使用独立于 *ReaR* 的方法备份的数据。

BACKUP_PROG_EXCLUDE 变量是 `glob (3)` 式通配符模式的数组，传递给 `tar` 或 `rsync`。请注意，需要引用模式以防止 `shell` 在读取配置文件时被 `shell` 扩展。此变量的默认值在 `/usr/share/rear/conf/default.conf` 文件中设置。默认值包含 `/tmp configured` 模式，它排除 `/tmp` 目录下的所有文件和目录，但不排除 `/tmp` 目录本身。

如果您需要排除其他文件和目录，请将其他模式附加到变量，而不是覆盖它，以保留默认值。例如，要排除 `/data/temp` 目录下的所有文件和目录，请使用：

```
# BACKUP_PROG_EXCLUDE+=( '/data/temp/*' )
```

`rear mkbackup` 命令在日志中列出备份排除模式。您可以在 `/var/log/rear` 目录中找到日志文件。这可用于在执行完整系统恢复前验证排除的规则。例如，日志可以包含以下条目：

```
2025-04-29 10:17:41.312431050 Making backup (using backup method NETFS)
2025-04-29 10:17:41.314369109 Backup include list (backup-include.txt contents):
2025-04-29 10:17:41.316197323 /
2025-04-29 10:17:41.318052001 Backup exclude list (backup-exclude.txt contents):
2025-04-29 10:17:41.319857125 /tmp/*
2025-04-29 10:17:41.321644442 /dev/shm/*
2025-04-29 10:17:41.323436363 /var/lib/rear/output/*
```

在这里，整个 **root** 文件系统包含在备份中，除了 **/tmp**、**/dev/shm** 和 **/var/lib/rear/output** 目录下的所有文件和目录外。