



# Red Hat Enterprise Linux 8

## 自动安装 RHEL

通过预定义配置在一个或多个系统上部署 RHEL



# Red Hat Enterprise Linux 8 自动安装 RHEL

---

通过预定义配置在一个或多个系统上部署 RHEL

## 法律通告

Copyright © 2024 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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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.

## 摘要

您可以使用 Kickstart 自动化 RHEL 安装。使用此方法在多个系统上部署相同的 RHEL 配置。Kickstart 根据您在配置文件中指定的参数安装 RHEL。安装源可以是安装介质、ISO 文件、红帽内容交付网络(CDN)或本地网络中的服务器。

# 目录

对红帽文档提供反馈 .....	5
部分 I. 准备 RHEL 安装 .....	6
第 1 章 系统要求和支持的构架 .....	7
1.1. 支持的安装目标 .....	7
1.2. 系统规格 .....	7
1.3. 磁盘和内存要求 .....	8
1.4. 图形显示器分辨率要求 .....	9
1.5. UEFI 安全引导和 BETA 版本要求 .....	9
第 2 章 将 RHEL 系统注册到红帽的值 .....	10
第 3 章 自定义安装介质 .....	11
第 4 章 为 RHEL 创建一个可引导的安装介质 .....	12
4.1. 安装引导介质选项 .....	12
4.2. 创建可引导 DVD .....	12
4.3. 在 LINUX 中创建可引导 USB 设备 .....	13
4.4. 在 WINDOWS 中创建可引导 USB 设备 .....	14
4.5. 在 MACOS 上创建一个可引导 USB 设备 .....	15
第 5 章 准备基于网络的存储库 .....	17
5.1. 基于网络安装的端口 .....	17
5.2. 在 NFS 服务器中创建安装源 .....	17
5.3. 使用 HTTP 或 HTTPS 创建安装源 .....	18
5.4. 使用 FTP 创建安装源 .....	20
第 6 章 准备一个 UEFI HTTP 安装源 .....	23
6.1. 网络安装概述 .....	23
6.2. 为网络引导配置 DHCPV4 服务器 .....	23
6.3. 为网络引导配置 DHCPV6 服务器 .....	24
6.4. 为 HTTP 引导配置 HTTP 服务器 .....	26
第 7 章 准备一个 PXE 安装源 .....	29
7.1. 网络安装概述 .....	29
7.2. 为网络引导配置 DHCPV4 服务器 .....	29
7.3. 为网络引导配置 DHCPV6 服务器 .....	30
7.4. 为基于 BIOS 的客户端配置 TFTP 服务器 .....	32
7.5. 为基于 UEFI 的客户端配置 TFTP 服务器 .....	33
7.6. 为 IBM POWER 系统配置网络服务器 .....	35
第 8 章 准备一个启用了 UEFI 安全引导的系统，来安装和引导 RHEL BETA 版本 .....	38
8.1. UEFI 安全引导和 RHEL BETA 版本 .....	38
8.2. 为 UEFI 安全引导添加 BETA 公钥 .....	38
8.3. 删除 BETA 公钥 .....	38
第 9 章 在 64 位 IBM Z 上准备 RHEL 安装 .....	40
9.1. 规划在 64 位 IBM Z 上进行安装 .....	40
9.2. 64 位 IBM Z 服务器中的安装过程概述 .....	41
9.3. 在 64 位 IBM Z 服务器中安装 RHEL 的引导介质 .....	41
9.4. 自定义引导参数 .....	42
9.5. 64 位 IBM Z 中的参数和配置文件 .....	44
9.6. 在 Z/VM 客户虚拟机中准备安装 .....	49

<b>部分 II. 完全安装 RHEL 并半自动化</b> .....	<b>51</b>
<b>第 10 章 自动化安装 workflow</b> .....	<b>52</b>
<b>第 11 章 创建 KICKSTART 文件</b> .....	<b>53</b>
11.1. 使用 KICKSTART 配置工具创建 KICKSTART 文件	53
11.2. 执行手动安装来创建 KICKSTART 文件	53
11.3. 从以前的 RHEL 安装转换为一个 KICKSTART 文件	54
11.4. 使用 IMAGE BUILDER 创建自定义镜像	55
<b>第 12 章 将 KICKSTART 文件添加到 UEFI HTTP 或 PXE 安装源中</b> .....	<b>56</b>
12.1. 基于网络安装的端口	56
12.2. 在 NFS 服务器上共享安装文件	56
12.3. 在 HTTP 或 HTTPS 服务器上共享安装文件	57
12.4. 在 FTP 服务器上共享安装文件	58
<b>第 13 章 半自动化安装：使 KICKSTART 文件可供 RHEL 安装程序使用</b> .....	<b>61</b>
13.1. 在本地卷上共享安装文件	61
13.2. 在本地卷上共享安装文件，以便自动载入	61
<b>第 14 章 启动 KICKSTART 安装</b> .....	<b>63</b>
14.1. 使用 PXE 自动启动 KICKSTART 安装	63
14.2. 使用本地卷自动启动 KICKSTART 安装	64
14.3. 在 IBM Z 中引导安装以便在 LPAR 中安装 RHEL	64
14.4. 在 IBM Z 中引导安装以便在 Z/VM 中安装 RHEL	67
14.5. 安装过程中的控制台和日志记录	69
<b>部分 III. 安装后的任务</b> .....	<b>71</b>
<b>第 15 章 使用订阅管理器注册 RHEL</b> .....	<b>72</b>
15.1. 使用安装程序 GUI 注册 RHEL 8	72
15.2. REGISTRATION ASSISTANT	72
15.3. 使用命令行注册您的系统	72
<b>第 16 章 使用 SUBSCRIPTION-MANAGER 命令行工具配置系统目的</b> .....	<b>74</b>
<b>第 17 章 在 64 位 IBM Z 中配置 LINUX 实例</b> .....	<b>77</b>
17.1. 添加 DASD	77
17.2. 在线动态设定 DASD	77
17.3. 准备使用低级格式化的新 DASD	78
17.4. 在线永久设定 DASD	79
17.5. DASD 是 ROOT 文件系统一部分	79
17.6. DASD 不是 ROOT 文件系统的一部分	81
17.7. FCP LUN 是 ROOT 文件系统的一部分	82
17.8. FCP LUN 不是 ROOT 文件系统的一部分	84
17.9. 添加 QETH 设备	85
17.10. 动态添加 QETH 设备	85
17.11. 永久添加 QETH 设备	87
17.12. 为网络 ROOT 文件系统配置 64 位 IBM Z 网络设备	89
<b>第 18 章 保护您的系统</b> .....	<b>91</b>
<b>第 19 章 更改订阅服务</b> .....	<b>92</b>
19.1. 从 SUBSCRIPTION MANAGEMENT SERVER 中取消注册。	92
19.2. 在 SATELLITE 服务器中取消注册	93
<b>部分 IV. 附录</b> .....	<b>94</b>

---

<b>附录 A. KICKSTART 脚本文件格式参考</b> .....	<b>95</b>
A.1. KICKSTART 文件格式	95
A.2. KICKSTART 中的软件包选择	95
A.3. KICKSTART 文件中的脚本	99
A.4. ANACONDA 配置部分	104
A.5. KICKSTART 错误处理部分	104
A.6. KICKSTART 附加组件部分	105
<b>附录 B. KICKSTART 命令和选项参考</b> .....	<b>106</b>
B.1. KICKSTART 的修改	106
B.2. 用于安装程序配置和流量控制的 KICKSTART 命令	107
B.3. KICKSTART 命令进行系统配置	119
B.4. KICKSTART 命令用于网络配置	129
B.5. 用于处理存储的 KICKSTART 命令	134
B.6. RHEL 安装程序提供的附加组件的 KICKSTART 命令	158
B.7. ANACONDA 中使用的命令	160
B.8. KICKSTART 命令进行系统恢复	161
<b>附录 C. 引导选项参考</b> .....	<b>164</b>
C.1. 安装源引导选项	164
C.2. 网络引导选项	168
C.3. 控制台引导选项	171
C.4. 调试引导选项	173
C.5. 存储启动选项	174
C.6. 弃用的引导选项	175
C.7. 删除引导选项	176



---

## 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. 单击顶部导航栏中的 **Create**。
3. 在 **Summary** 字段中输入描述性标题。
4. 在 **Description** 字段中输入您对改进的建议。包括文档相关部分的链接。
5. 点对话框底部的 **Create**。

## 部分 I. 准备 RHEL 安装

# 第 1 章 系统要求和支持的构架

Red Hat Enterprise Linux 8 在混合云部署环境中提供稳定、安全、持续的基础，并提供更迅速地提供工作负载所需的工具。您可以将 RHEL 部署为受支持的 hypervisor 和云提供商环境以及物理基础架构上的客户机，因此您的应用程序可以利用领先的硬件架构平台中的创新。

在安装前，请查看为系统、硬件、安全性、内存和 RAID 提供的指南。

如果要将您的系统用作虚拟化主机，请查看 [虚拟化所需的硬件要求](#)。

Red Hat Enterprise Linux 支持以下构架：

- AMD 和 Intel 64 位构架
- 64 位 ARM 架构
- IBM Power Systems, Little Endian
- 64 位 IBM Z 架构

## 1.1. 支持的安装目标

安装目标是保存 Red Hat Enterprise Linux 并引导系统的存储设备。Red Hat Enterprise Linux 支持 AMD64、Intel 64 和 64 位 ARM 系统的以下安装目标：

- 通过标准内部接口（如 SCSI、SATA 或 SAS）连接的存储
- BIOS/固件 RAID 设备
- Intel64 和 AMD64 架构上扇区模式下的 NVDIMM 设备，由 `nd_pmem` 驱动程序支持。
- 光纤通道主机总线适配器和多路径设备。有些操作需要厂商提供的驱动程序。
- Xen 虚拟机上 Intel 处理器上的 Xen 块设备。
- KVM 虚拟机上 Intel 处理器上的 VirtIO 块设备。

红帽不支持安装到 USB 盘或 SD 内存卡。有关支持第三方虚拟化技术的详情，请查看 [红帽硬件兼容性列表](#)。

## 1.2. 系统规格

Red Hat Enterprise Linux 安装程序自动侦测并安装您的系统硬件，因此您不必提供具体系统信息。然而，对于某些 Red Hat Enterprise Linux 安装场景，建议您记录系统规格以备将来参考。这些情况包括：

### 使用自定义分区布局安装 RHEL

**Record:** 附加到系统的磁盘的型号、大小、类型和接口。例如：SATA0 上的 Seagate ST3320613AS 320 GB、SATA1 上的 Western Digital WD7500AAKS 750 GB。

### 在现有系统中将 RHEL 作为附加操作系统安装

**Record:** 系统上使用的分区。此信息可以包括文件系统类型、设备节点名称、文件系统标签和大小，并允许您在分区过程中识别特定的分区。如果某个操作系统是 Unix 操作系统，Red Hat Enterprise Linux 可能会报告不同的设备名称。通过执行与 `mount` 命令和 `blkid` 命令等效的命令，以及在 `/etc/fstab` 文件中可以找到其他信息。

如果安装了多个操作系统，Red Hat Enterprise Linux 安装程序会尝试自动探测它们，并配置引导装载程序来引导它们。如果没有自动探测到它们，您可以手动配置附加操作系统。

### 从本地磁盘上的镜像安装 RHEL

**Record:** 保存镜像的磁盘和目录。

### 从网络位置安装 RHEL

如果需要手动配置网络，即不使用 DHCP。

**Record:**

- IP 地址
- 子网掩码
- 网关的 IP 地址
- 服务器 IP 地址（如果需要）

如果您需要网络支持，请联系您的网络管理员。

### 在 iSCSI 目标上安装 RHEL

**Record:** iSCSI 目标的位置。根据您的网络，可能需要 CHAP 用户名和密码，以及反向 CHAP 用户名和密码。

### 如果系统是某个域的一部分，请安装 RHEL

验证域名是否由 DHCP 服务器提供。如果没有，在安装过程中输入域名。

## 1.3. 磁盘和内存要求

如果安装了一些操作系统，必须确定分配的磁盘空间与 Red Hat Enterprise Linux 要求的磁盘空间分离。在某些情况下，务必要将特定分区专用于 Red Hat Enterprise Linux，例如 AMD64、Intel 64 和 64 位 ARM，至少两个分区(/ 和 **swap**)必须专用于 RHEL，对于 IBM Power 系统服务器，至少有三个分区(/、**swap** 和 **PRReP** 引导分区)需要专用于 RHEL。

另外，必须至少有 10 GiB 可用磁盘空间。要安装 Red Hat Enterprise Linux，必须至少有 10GiB 未分区磁盘空间或可删除分区。

如需更多信息，请参阅 [分区参考](#)。

**表 1.1. 最低 RAM 要求**

安装类型	推荐的最小内存
本地介质安装（USB、DVD）	<ul style="list-style-type: none"> <li>• 1.5 GiB 用于 aarch64、s390x 和 x86_64 架构</li> <li>• 3 GiB 用于 ppc64le 架构</li> </ul>

安装类型	推荐的最小内存
NFS 网络安装	<ul style="list-style-type: none"> <li>● 1.5 GiB 用于 aarch64、s390x 和 x86_64 架构</li> <li>● 3 GiB 用于 ppc64le 架构</li> </ul>
HTTP、HTTPS 或者 FTP 网络安装	<ul style="list-style-type: none"> <li>● 3 GiB 用于 s390x 和 x86_64 架构</li> <li>● 4 GiB 用于 aarch64 和 ppc64le 架构</li> </ul>

使用比推荐的最低要求小的内存可完成安装。具体的要求取决于您的环境和安装路径。建议您测试各种配置，以确定环境所需的最小内存。使用 Kickstart 文件安装 Red Hat Enterprise Linux 与标准安装有相同的推荐最小 RAM 要求。然而，如果您的 Kickstart 文件包含需要额外内存的命令，或者将数据写入 RAM 磁盘，则可能需要额外的 RAM。如需更多信息，请参阅 [自动安装 RHEL](#) 文档。

## 1.4. 图形显示器分辨率要求

您的系统必须具有以下最低分辨率，以确保 Red Hat Enterprise Linux 的顺利和无错安装。

表 1.2. 显示器分辨率

产品版本	解决方案
Red Hat Enterprise Linux 8	<p><b>最小:</b> 800 x 600</p> <p><b>推荐:</b> 1026 x 768</p>

## 1.5. UEFI 安全引导和 BETA 版本要求

如果您计划在启用了 UEFI 安全引导的系统中安装 Red Hat Enterprise Linux Beta 版本，请首先禁用 UEFI 安全引导选项，然后开始安装。

UEFI 安全引导要求操作系统内核使用可识别的私钥签名，系统的 firmware 使用对应的公钥进行验证。对于 Red Hat Enterprise Linux Beta 版本，内核使用特定于红帽 Beta 的公钥进行签名，系统默认无法识别该公钥。因此，系统甚至无法引导安装介质。

### 其它资源

- 有关在 IBM 上安装 RHEL 的详情，请参考 [IBM 安装文档](#)
- [安全强化](#)
- [创建自定义 RHEL 系统镜像](#)
- [红帽生态系统目录](#)
- [RHEL 技术能力和限制](#)

## 第 2 章 将 RHEL 系统注册到红帽的值

注册在您的系统与红帽之间建立一个授权连接。红帽向注册的系统（无论是物理或虚拟机）发布一个标识和验证系统的证书，以便它能够接收来自红帽的保护的内容、软件更新、安全补丁、支持以及管理的服务。

通过有效的订阅，您可以使用以下方法注册 Red Hat Enterprise Linux (RHEL) 系统：

- 在安装过程中，使用安装程序图形用户界面(GUI)或文本用户界面(TUI)
- 安装后，使用命令行界面(CLI)
- 在安装过程中或之后，使用 kickstart 脚本或激活码自动执行。

注册您的系统的具体步骤取决于您使用的 RHEL 版本，以及您选择的注册方法。

将您的系统注册到红帽可启用您用来管理系统和报告数据的特性和功能。例如，注册的系统被授权通过 Red Hat Content Delivery Network (CDN) 或 Red Hat Satellite Server 访问订阅产品的受保护的内容存储库。这些内容存储库包括仅适用于具有有效订阅的客户的红帽软件包和更新。这些软件包和更新包括 RHEL 和其他红帽产品的安全补丁、错误修复和新功能。



### 重要

基于权利的订阅模式已弃用，并将在以后的版本中停用。简单内容访问现在是默认的订阅模式。它提供了一个改进的订阅体验，可以在能够访问该系统上的红帽订阅内容时，消除将订阅附加到系统的需要。如果您的红帽账户使用基于权利的订阅模式，请联络您的红帽客户团队，例如：技术客户经理(TAM)或解决方案架构师(SA)，以准备迁移到简单内容访问。如需更多信息，请参阅 [将订阅服务转换到混合云](#)。

## 第 3 章 自定义安装介质

详情请参阅 [制作自定义的 RHEL 系统镜像](#)。

## 第 4 章 为 RHEL 创建一个可引导的安装介质

您可以从 [客户门户网站](#) 下载 ISO 文件，以准备可引导的物理安装介质，如 USB 或者 DVD。从 RHEL 8 开始，红帽不再为 **Server** 和 **Workstation** 提供单独的变体。**Red Hat Enterprise Linux for x86\_64** 包括 **Server** 和 **Workstation** 功能。**Server** 和 **Workstation** 之间的区别是在安装或配置过程中通过系统目的进行管理。

从客户门户网站下载 ISO 文件后，创建一个可引导的物理安装介质，如 USB 或者 DVD，来继续安装过程。

对于禁止 USB 驱动器的安全环境，请考虑使用镜像构建器创建和部署引用镜像。此方法可确保遵守安全策略，同时保持系统的完整性。如需了解更多详细信息，请参阅 [镜像构建器文档](#)。



### 注意

默认情况下,安装介质中使用 `inst.stage2=` 引导选项,并被设置为特定标签,例如：`inst.stage2=hd:LABEL=RHEL8\x86_64`。如果您修改了包含运行时镜像的文件系统的默认标签，或者使用自定义流程引导安装系统，请验证标签是否已设置为正确的值。

### 4.1. 安装引导介质选项

引导 Red Hat Enterprise Linux 安装程序有多个选项。

#### 完整的安装 DVD 或者 USB 闪存驱动器

使用 **DVD ISO** 镜像创建完整的安装 DVD 或者 USB 闪存驱动器。DVD 或者 USB 闪存驱动器可用作引导设备，并作为安装软件包的安装源。

#### 最小安装 DVD、CD 或者 USB 闪存驱动器

使用 **引导 ISO** 镜像生成最小安装 CD、DVD 或者 USB 闪存驱动器，该镜像只包含引导系统并启动安装程序所需的最小文件。



### 重要

如果您不使用 Content Delivery Network(CDN)下载所需的软件包，则 **Boot ISO** 镜像需要一个包含所需软件包的安装源。

#### PXE 服务器

*预启动执行环境* (PXE) 服务器允许安装程序通过网络引导。系统引导后，您必须从不同的安装源（如本地磁盘或网络位置）完成安装。

#### 镜像构建器

使用镜像构建器，您可以创建自定义的系统和云镜像，以便在虚拟和云环境中安装 Red Hat Enterprise Linux。

### 4.2. 创建可引导 DVD

您可以使用 DVD 刻录软件和 DVD 刻录程序创建可引导安装 DVD。从 ISO 镜像文件生成 DVD 的具体步骤会很大不同，具体取决于操作系统以及安装的刻录软件。有关从 ISO 镜像文件刻录 DVD 的具体步骤，请参考您的系统刻录软件文档。



### 警告

您可以使用 DVD ISO 镜像（完整安装）或 Boot ISO 镜像（最小安装）创建可引导的 DVD。但是，DVD ISO 镜像大于 4.7 GB，因此它可能不适合单个或双层 DVD。继续操作前，请检查 DVD ISO 镜像文件的大小。当使用 DVD ISO 镜像创建可引导安装介质时，建议使用 USB 闪存。有关禁止 USB 驱动器的环境情况，请参阅 [镜像构建器文档](#)。

## 4.3. 在 LINUX 中创建可引导 USB 设备

您可以创建一个可引导 USB 设备，然后您可以用来在其他机器上安装 Red Hat Enterprise Linux。这个过程覆盖 USB 驱动器中的现有数据，而无需任何警告。备份任何数据或使用一个空闪存。可引导的 USB 驱动器不能用于存储数据。

### 先决条件

- 您已从 [Product Downloads](#) 页面下载了完整的安装 DVD ISO 或最小安装 Boot ISO 镜像。
- 您有一个有足够容量的 USB 闪存驱动器 ISO 镜像。所需的大小会有所不同，但推荐的 USB 大小为 8 GB。

### 流程

1. 将 USB 闪存驱动器连接到该系统。
2. 打开一个终端窗口，并显示最近事件的日志。

```
$ dmesg|tail
```

附加的 USB 闪存驱动器的消息会在日志的底部显示。记录连接设备的名称。

3. 以 root 用户身份登录：

```
$ su -
```

提示时请输入您的根密码。

4. 查找分配给该驱动器的设备节点。在这个示例中，驱动器名称是 **sdd**。

```
# dmesg|tail
[288954.686557] usb 2-1.8: New USB device strings: Mfr=0, Product=1, SerialNumber=2
[288954.686559] usb 2-1.8: Product: USB Storage
[288954.686562] usb 2-1.8: SerialNumber: 000000009225
[288954.712590] usb-storage 2-1.8:1.0: USB Mass Storage device detected
[288954.712687] scsi host6: usb-storage 2-1.8:1.0
[288954.712809] usbcore: registered new interface driver usb-storage
[288954.716682] usbcore: registered new interface driver uas
[288955.717140] scsi 6:0:0:0: Direct-Access   Generic STORAGE DEVICE  9228 PQ: 0
```

```
ANSI: 0
```

```
[288955.717745] sd 6:0:0:0: Attached scsi generic sg4 type 0
```

```
[288961.876382] sd 6:0:0:0: sdd Attached SCSI removable disk
```

5. 如果插入的 USB 设备自动挂载，请在继续执行后续步骤前将其卸载。要卸载，请使用 **umount** 命令。如需更多信息，请参阅 [使用 umount 卸载文件系统](#)。
6. 将 ISO 镜像直接写到 USB 设备：

```
# dd if=/image_directory/image.iso of=/dev/device
```

- 使用您下载的 ISO 镜像文件的完整路径替换 `/image_directory/image.iso`,
- 使用您通过 **dmesg** 命令得到的设备名称替换 `device`。  
在这个示例中，ISO 镜像的完整路径为 `/home/testuser/Downloads/rhel-8-x86_64-boot.iso`，设备名称为 **sdd**：

```
# dd if=/home/testuser/Downloads/rhel-8-x86_64-boot.iso of=/dev/sdd
```

分区名称通常是设备名称附带一个数字后缀。例如：**sdd** 是设备名称，**sdd1** 是设备 **sdd** 中的分区名称。

7. 等待 **dd** 命令完成将镜像写入该设备。运行 **sync** 命令，来将缓存的写同步到设备。当 **#** 提示符出现时，代表数据传输已完成。当看到提示符时，从 **root** 帐户注销，并拔出 USB 驱动器。USB 驱动器现在准备用作引导设备。

## 4.4. 在 WINDOWS 中创建可引导 USB 设备

您可以使用各种工具在 Windows 系统上创建一个可引导的 USB 设备。红帽建议您使用 Fedora Media Writer，下载地址为 <https://github.com/FedoraQt/MediaWriter/releases>。Fedora Media Writer 是一个社区产品，它不被红帽支持。您可以在 <https://github.com/FedoraQt/MediaWriter/issues> 中报告任何问题。

这个过程覆盖 USB 驱动器中的现有数据，而无需任何警告。备份任何数据或使用一个空闪存。可引导的 USB 驱动器不能用于存储数据。

### 先决条件

- 您已从 [Product Downloads](#) 页面下载了完整的安装 DVD ISO 或最小安装 Boot ISO 镜像。
- 您有一个有足够容量的 USB 闪存驱动器 ISO 镜像。所需的大小会有所不同，但推荐的 USB 大小为 8 GB。

### 流程

1. 从 <https://github.com/FedoraQt/MediaWriter/releases> 下载并安装 Fedora Media Writer。
2. 将 USB 闪存驱动器连接到该系统。
3. 打开 Fedora Media Writer。
4. 在主窗口中点击 **Custom Image** 并选择之前下载的 Red Hat Enterprise Linux ISO 镜像。
5. 在 **Write Custom Image** 窗口中，选择要使用的驱动器。

- 点 **Write to disk**。引导介质创建过程开始。操作完成后不要拔出驱动器。这个操作可能需要几分钟，具体要看 ISO 镜像的大小以及 USB 驱动器的写入速度。
- 当操作完成后，卸载 USB 驱动器。USB 驱动器现在可作为引导设备使用。

## 4.5. 在 MACOS 上创建一个可引导 USB 设备

您可以创建一个可引导 USB 设备，然后您可以用来在其他机器上安装 Red Hat Enterprise Linux。



### 重要

按照此流程会覆盖之前保存在 USB 驱动器中的任何数据，而没有任何警告。备份任何数据或使用一个空闪存。可引导的 USB 驱动器不能用于存储数据。

### 先决条件

- 您已从 [Product Downloads](#) 页面下载了完整的安装 DVD ISO 或最小安装 Boot ISO 镜像。
- 您有一个有足够容量的 USB 闪存驱动器 ISO 镜像。所需的大小会有所不同，但推荐的 USB 大小为 8 GB。

### 流程

- 将 USB 闪存驱动器连接到该系统。
- 使用 `diskutil list` 命令识别设备路径。设备路径的格式为 `/dev/disknumber`，其中 `number` 是磁盘号。该磁盘从零(0)开始编号。通常，`disk0` 是 OS X 恢复磁盘，`disk1` 是主 OS X 安装。在以下示例中，USB 设备为 `disk2`：

```
$ diskutil list
/dev/disk0
#:              TYPE NAME              SIZE  IDENTIFIER
0:      GUID_partition_scheme          *500.3 GB  disk0
1:              EFI EFI                209.7 MB  disk0s1
2:      Apple_CoreStorage              400.0 GB  disk0s2
3:      Apple_Boot Recovery HD         650.0 MB  disk0s3
4:      Apple_CoreStorage              98.8 GB  disk0s4
5:      Apple_Boot Recovery HD         650.0 MB  disk0s5
/dev/disk1
#:              TYPE NAME              SIZE  IDENTIFIER
0:      Apple_HFS YosemiteHD          *399.6 GB  disk1
Logical Volume on disk0s1
8A142795-8036-48DF-9FC5-84506DFBB7B2
Unlocked Encrypted
/dev/disk2
#:              TYPE NAME              SIZE  IDENTIFIER
0:      FDisk_partition_scheme         *8.1 GB   disk2
1:      Windows_NTFS SanDisk USB       8.1 GB   disk2s1
```

- 通过将 NAME、TYPE 和 SIZE 列与您的闪存驱动器进行比较来识别您的 USB 闪存驱动器。例如，NAME 应为 **Finder** 工具中闪存驱动器图标标题。您还可以将这些值与闪存驱动器信息面板中的值进行比较。
- 卸载该闪存驱动器的文件系统卷：

```
$ diskutil unmountDisk /dev/disknumber
Unmount of all volumes on disknumber was successful
```

该命令完成后，该闪存驱动器图标会从桌面消失。如果图标没有消失，您可能选择了错误的磁盘。尝试卸载系统磁盘会意外返回 **failed to unmount** 错误。

5. 将 ISO 镜像写入闪存驱动器：

```
# sudo dd if=/path/to/image.iso of=/dev/rdisknumber
```



### 注意

macOS 为每个存储设备提供了块(**/dev/disk\***)和字符设备(**/dev/rdisk\***)文件。将镜像写入 **/dev/rdisknumber** 字符设备比写入 **/dev/disknumber** 块设备要快。

例如，要将 **/Users/user\_name/Downloads/rhel-8-x86\_64-boot.iso** 文件写入 **/dev/rdisk2** 设备，请输入以下命令：

```
# sudo dd if=/Users/user_name/Downloads/rhel-8-x86_64-boot.iso of=/dev/rdisk2
```

6. 等待 **dd** 命令完成将镜像写入该设备。当 **#** 提示符出现时，代表数据传输已完成。当出现提示符后，退出 **root** 帐户并拔出 USB 驱动器。USB 驱动器现在可作为引导设备使用。

## 其它资源

- [配置系统目的](#)
- [用于 RHEL 8/9 服务器或工作站的 ISO](#)

## 第 5 章 准备基于网络的存储库

您必须准备存储库，以便从网络系统安装 RHEL。

### 5.1. 基于网络安装的端口

下表列出了必须在服务器上打开的端口，以便为每种基于网络的安装提供文件。

表 5.1. 基于网络安装的端口

使用的协议	打开端口
HTTP	80
HTTPS	443
FTP	21
NFS	2049, 111, 20048
TFTP	69

#### 其它资源

- [安全网络](#)

### 5.2. 在 NFS 服务器中创建安装源

您可以使用这个安装方法从单个源安装多个系统，而无需连接到物理介质。

#### 先决条件

- 您对 Red Hat Enterprise Linux 8 服务器有管理员级别的访问权限，且这个服务器与要安装的系统位于同一个网络上。
- 您已从 [Product Downloads](#) 页面下载了完整的安装 DVD ISO。
- 您已使用镜像文件创建了可引导 CD、DVD 或者 USB 设备。
- 您已确认防火墙允许您在安装的系统访问远程安装源。如需更多信息，请参阅 [用于基于网络安装的端口](#)。



#### 重要

确保您在 `inst.ks` 和 `inst.repo` 中使用不同的路径。当使用 NFS 托管安装源时，您不能使用相同的 nfs 共享来托管 Kickstart。

#### 流程

1. 安装 `nfs-utils` 软件包：

```
# yum install nfs-utils
```

- 将 DVD ISO 镜像复制到 NFS 服务器的目录中。
- 使用文本编辑器打开 `/etc/exports` 文件，并使用以下语法添加一行：

```
/exported_directory/ clients
```

- 使用到包括 ISO 镜像的目录的完全路径替换 `/exported_directory/`。
- 使用以下方法之一替换 `clients`：
  - 目标系统的主机名或 IP 地址
  - 所有目标系统可用于访问 ISO 镜像的子网
  - 要允许任何可通过网络访问 NFS 服务器的系统使用 ISO 镜像，请使用星号(\*)

有关这一字段的格式的详细信息，请参见 **exports(5)** 手册页。

例如，一个使 `/rhel8-install/` 目录对所有客户端只读的基本配置：

```
/rhel8-install *
```

- 保存 `/etc/exports` 文件并退出文本编辑器。
- 启动 nfs 服务：

```
# systemctl start nfs-server.service
```

如果在更改 `/etc/exports` 文件前服务在运行中，请重新载入 NFS 服务器配置：

```
# systemctl reload nfs-server.service
```

现在可通过 NFS 访问该 ISO 镜像并可作为安装源使用。

在配置安装源时，使用 **nfs:** 协议、服务器主机名或 IP 地址、冒号 (:) 以及该 ISO 镜像所在目录。例如：如果服务器主机名为 **myserver.example.com**，且在 `/rhel8-install/` 中保存了 ISO 镜像，则指定 **nfs:myserver.example.com:/rhel8-install/** 作为安装源。

### 5.3. 使用 HTTP 或 HTTPS 创建安装源

您可以使用安装树为基于网络的安装创建一个安装源，该树是包含提取的 DVD ISO 镜像内容和一个有效的 `.treeinfo` 文件的目录。可通过 HTTP 或 HTTPS 访问该安装源。

#### 先决条件

- 您对 Red Hat Enterprise Linux 8 服务器有管理员级别的访问权限，且这个服务器与要安装的系统位于同一个网络上。
- 您已从 [Product Downloads](#) 页面下载了完整的安装 DVD ISO。
- 您已使用镜像文件创建了可引导 CD、DVD 或者 USB 设备。

- 您已确认防火墙允许您在安装的系统访问远程安装源。如需更多信息，请参阅 [用于基于网络安装的端口](#)。
- **httpd** 软件包已安装。
- 如果您使用 **https** 安装源，则 **mod\_ssl** 软件包已安装。



### 警告

如果您的 Apache Web 服务器配置启用了 SSL 安全性，则首选启用 TLSv1.3 协议。默认情况下，TLSv1.2 被启用，您可以使用 TLSv1 (LEGACY) 协议。



### 重要

如果您使用带有自签名证书的 HTTPS 服务器，则必须使用 **noverifyssl** 选项引导安装程序。

## 流程

1. 将 DVD ISO 镜像复制到 HTTP(S) 服务器。
2. 创建一个挂载 DVD ISO 镜像的合适的目录，例如：

```
# mkdir /mnt/rhel8-install/
```

3. 将 DVD ISO 镜像挂载到目录：

```
# mount -o loop,ro -t iso9660 /image_directory/image.iso /mnt/rhel8-install/
```

使用 DVD ISO 镜像的路径替换 `/image_directory/image.iso`。

4. 将挂载镜像中的文件复制到 HTTP(S) 服务器 root 中。

```
# cp -r /mnt/rhel8-install/ /var/www/html/
```

这个命令使用镜像的内容创建 `/var/www/html/rhel8-install/` 目录。请注意，有些其他复制方法可能会跳过有效安装源所需的 `.treeinfo` 文件。对整个目录输入 `cp` 命令，如此流程中所示，正确地复制 `.treeinfo`。

5. 启动 **httpd** 服务：

```
# systemctl start httpd.service
```

安装树现在可以访问并可作为安装源使用。



## 注意

在配置安装源时，使用 **http://** 或 **https://** 作为协议、服务器主机名或 IP 地址，以及 ISO 镜像中包含文件的目录（相对于 HTTP 服务器 root）。例如，如果您使用 HTTP，则服务器主机名为 **myserver.example.com**，且已将镜像中的文件复制到 **/var/www/html/rhel8-install/**，指定 **http://myserver.example.com/rhel8-install/** 作为安装源。

## 其它资源

- [部署不同类型的服务器](#)

## 5.4. 使用 FTP 创建安装源

您可以使用安装树为基于网络的安装创建一个安装源，该树是包含提取的 DVD ISO 镜像内容和一个有效的 **.treeinfo** 文件的目录。通过 FTP 访问该安装源。

### 先决条件

- 您对 Red Hat Enterprise Linux 8 服务器有管理员级别的访问权限，且这个服务器与要安装的系统位于同一个网络上。
- 您已从 [Product Downloads](#) 页面下载了完整的安装 DVD ISO。
- 您已使用镜像文件创建了可引导 CD、DVD 或者 USB 设备。
- 您已确认防火墙允许您在安装的系统访问远程安装源。如需更多信息，请参阅 [用于基于网络安装的端口](#)。
- **vsftpd** 软件包已安装。

### 流程

1. 在文本编辑器中打开并编辑 **/etc/vsftpd/vsftpd.conf** 配置文件。
  - a. 将 **anonymous\_enable=NO** 行改为 **anonymous\_enable=YES**
  - b. 将 **write\_enable=YES** 行改为 **write\_enable=NO**。
  - c. 添加行 **pasv\_min\_port=<min\_port>** 和 **pasv\_max\_port=<max\_port>**。用被动模式下 FTP 服务器使用的端口范围替换 **<min\_port>** 和 **<max\_port>**，例如 **10021** 和 **10031**。  
在具有各种防火墙/NAT 设置的网络环境中，这个步骤可能是必要的。
  - d. 可选：向您的配置中添加自定义更改。可用选项请查看 **vsftpd.conf(5)** 手册页。这个步骤假设使用了默认选项。



### 警告

如果在 **vsftpd.conf** 文件中配置了 SSL/TLS 安全性，请确保只启用 TLSv1 协议，并禁用 SSLv2 和 SSLv3。这是因为 POODLE SSL 漏洞 (CVE-2014-3566)。详情请参考

<https://access.redhat.com/solutions/1234773>。

## 2. 配置服务器防火墙。

### a. 启用防火墙：

```
# systemctl enable firewalld
```

### b. 启动防火墙：

```
# systemctl start firewalld
```

### c. 配置防火墙以允许上一步中的 FTP 端口和端口范围：

```
# firewall-cmd --add-port min_port-max_port/tcp --permanent
# firewall-cmd --add-service ftp --permanent
```

将 `<min_port>` 和 `<max_port>` 替换为您输入到 `/etc/vsftpd/vsftpd.conf` 配置文件中的端口号。

### d. 重新载入防火墙以应用新规则：

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

## 3. 将 DVD ISO 镜像复制到 FTP 服务器。

## 4. 创建一个挂载 DVD ISO 镜像的合适的目录，例如：

```
# mkdir /mnt/rhel8-install
```

## 5. 将 DVD ISO 镜像挂载到目录：

```
# mount -o loop,ro -t iso9660 /image-directory/image.iso /mnt/rhel8-install
```

使用 DVD ISO 镜像的路径替换 `/image-directory/image.iso`。

## 6. 将挂载镜像中的文件复制到 FTP 服务器 root 目录中：

```
# mkdir /var/ftp/rhel8-install
# cp -r /mnt/rhel8-install/ /var/ftp/
```

这个命令使用镜像内容创建 `/var/ftp/rhel8-install/` 目录。某些复制方法可以跳过 `.treeinfo` 文件（一个有效的安装源需要这个文件）。对整个目录输入 `cp` 命令，如此流程中所示，将正确复制 `.treeinfo`。

7. 请确定在复制的内容中设置了正确的 SELinux 上下文和访问模式：

```
# restorecon -r /var/ftp/rhel8-install
# find /var/ftp/rhel8-install -type f -exec chmod 444 {} \;
# find /var/ftp/rhel8-install -type d -exec chmod 755 {} \;
```

8. 启动 **vsftpd** 服务：

```
# systemctl start vsftpd.service
```

如果在更改 **/etc/vsftpd/vsftpd.conf** 文件前运行该服务，重启该服务以载入经过编辑的文件：

```
# systemctl restart vsftpd.service
```

在引导过程中启用 **vsftpd** 服务：

```
# systemctl enable vsftpd
```

安装树现在可以访问并可作为安装源使用。

在配置安装源时，使用 **ftp://** 作为协议、服务器主机名或 IP 地址，以及保存 ISO 镜像中文件的目录（相对于 FTP 服务器 root）。例如：如果服务器主机名是 **myserver.example.com**，并且已将镜像中的文件复制到 **/var/ftp/rhel8-install/**，指定 **ftp://myserver.example.com/rhel8-install/** 作为安装源。

## 第 6 章 准备一个 UEFI HTTP 安装源

作为本地网络上服务器的管理员，您可以配置 HTTP 服务器，以便为网络中的其他系统启用 HTTP 引导和网络安装。

### 6.1. 网络安装概述

网络安装允许您将 Red Hat Enterprise Linux 安装到可访问安装服务器的系统。网络安装至少需要两个系统：

#### 服务器

运行 DHCP 服务器、HTTP、HTTPS、FTP 或 NFS 的服务器以及 PXE 引导情况下的 TFTP 服务器。虽然每个服务器可以运行在不同的物理系统上，但本节中的流程假设单个系统正在运行所有服务器。

#### 客户端

要安装 Red Hat Enterprise Linux 的系统。安装开始后，客户端会查询 DHCP 服务器，从 HTTP 或 TFTP 服务器接收引导文件，并从 HTTP、HTTPS、FTP 或者 NFS 服务器下载安装镜像。与其它安装方法不同，客户端不需要任何物理引导介质来启动安装。

要从网络引导客户端，请在固件中或客户端上的快速引导菜单中启用网络引导。在某些硬件上，从网络引导的选项可能被禁用了，或者不可用。

准备使用 HTTP 或 PXE 从网络安装 Red Hat Enterprise Linux 的工作流步骤如下：

#### 流程

1. 将安装 ISO 镜像或者安装树导出到 NFS、HTTPS、HTTP 或者 FTP 服务器中。
2. 配置 HTTP 或 TFTP 服务器以及 DHCP 服务器，并在服务器上启动 HTTP 或 TFTP 服务。
3. 引导客户端并开始安装。

您可以在以下网络引导协议之间进行选择：

#### HTTP

如果您的客户端 UEFI 支持，红帽建议使用 HTTP 引导。HTTP 引导通常更加可靠。

#### PXE (TFTP)

客户端系统更广泛地支持 PXE 引导，但通过此协议发送引导文件可能会很慢，并导致超时失败。

#### 其他资源

- [准备基于网络的存储库](#)
- [Red Hat Satellite 产品文档](#)

### 6.2. 为网络引导配置 DHCPV4 服务器

在您的服务器上启用 DHCP 版本 4 (DHCPv4) 服务，以便它可以提供网络引导功能。

#### 先决条件

- 您通过 IPv4 协议准备网络安装。  
对于 IPv6，请参阅 [为网络引导配置 DHCPv6 服务器](#)。

- 查找服务器的网络地址。  
在以下示例中，服务器有一个具有此配置的网卡：

**IPv4 地址**

192.168.124.2/24

**IPv4 网关**

192.168.124.1

**流程**

1. 安装 DHCP 服务器：

```
yum install dhcp-server
```

2. 设置 DHCPv4 服务器。在 `/etc/dhcp/dhcpd.conf` 文件中输入以下配置。替换地址以匹配您的网卡。

```
option architecture-type code 93 = unsigned integer 16;

subnet 192.168.124.0 netmask 255.255.255.0 {
  option routers 192.168.124.1;
  option domain-name-servers 192.168.124.1;
  range 192.168.124.100 192.168.124.200;
  class "pxeclients" {
    match if substring (option vendor-class-identifier, 0, 9) = "PXEClient";
    next-server 192.168.124.2;
    if option architecture-type = 00:07 {
      filename "redhat/EFI/BOOT/BOOTX64.EFI";
    }
    else {
      filename "pxelinux/pxelinux.0";
    }
  }
  class "httpclients" {
    match if substring (option vendor-class-identifier, 0, 10) = "HTTPClient";
    option vendor-class-identifier "HTTPClient";
    filename "http://192.168.124.2/redhat/EFI/BOOT/BOOTX64.EFI";
  }
}
```

3. 启动 DHCPv4 服务：

```
# systemctl enable --now dhcpd
```

## 6.3. 为网络引导配置 DHCPV6 服务器

在您的服务器上启用 DHCP 版本 6 (DHCPv4) 服务，以便它可以提供网络引导功能。

**先决条件**

- 您通过 IPv6 协议准备网络安装。  
对于 IPv4，请参阅 [为网络引导配置 DHCPv4 服务器](#)。

- 查找服务器的网络地址。  
在以下示例中，服务器有一个具有此配置的网卡：

#### IPv6 地址

```
fd33:eb1b:9b36::2/64
```

#### IPv6 网关

```
fd33:eb1b:9b36::1
```

### 流程

1. 安装 DHCP 服务器：

```
yum install dhcp-server
```

2. 设置 DHCPv6 服务器。在 `/etc/dhcp/dhcpd6.conf` 文件中输入以下配置。替换地址以匹配您的网卡。

```
option dhcp6.bootfile-url code 59 = string;
option dhcp6.vendor-class code 16 = {integer 32, integer 16, string};

subnet6 fd33:eb1b:9b36::/64 {
    range6 fd33:eb1b:9b36::64 fd33:eb1b:9b36::c8;

    class "PXECient" {
        match substring (option dhcp6.vendor-class, 6, 9);
    }

    subclass "PXECient" "PXECient" {
        option dhcp6.bootfile-url
        "tftp://[fd33:eb1b:9b36::2]/redhat/EFI/BOOT/BOOTX64.EFI";
    }

    class "HTTPClient" {
        match substring (option dhcp6.vendor-class, 6, 10);
    }

    subclass "HTTPClient" "HTTPClient" {
        option dhcp6.bootfile-url
        "http://[fd33:eb1b:9b36::2]/redhat/EFI/BOOT/BOOTX64.EFI";
        option dhcp6.vendor-class 0 10 "HTTPClient";
    }
}
```

3. 启动 DHCPv6 服务：

```
# systemctl enable --now dhcpd6
```

4. 如果 DHCPv6 数据包被防火墙中的 RP 过滤器丢弃了，请检查其日志。如果日志包含 `rpfilter_DROP` 条目，请使用 `/etc/firewalld/firewalld.conf` 文件中的以下配置禁用过滤器：

```
IPv6_rpfilter=no
```

## 6.4. 为 HTTP 引导配置 HTTP 服务器

您必须在服务器上安装并启用 **httpd** 服务，以便服务器可以在网络上提供 HTTP 引导资源。

### 先决条件

- 查找服务器的网络地址。  
在以下示例中，服务器有一个 IPv4 地址为 **192.168.124.2** 的网卡。

### 流程

1. 安装 HTTP 服务器：

```
# yum install httpd
```

2. 创建 **/var/www/html/redhat/** 目录：

```
# mkdir -p /var/www/html/redhat/
```

3. 下载 RHEL DVD ISO 文件。请参阅 [所有 Red Hat Enterprise Linux 下载](#)。

4. 为 ISO 文件创建一个挂载点：

```
# mkdir -p /var/www/html/redhat/iso/
```

5. 挂载 ISO 文件：

```
# mount -o loop,ro -t iso9660 path-to-RHEL-DVD.iso /var/www/html/redhat/iso
```

6. 将引导装载程序、内核和 **initramfs** 从挂载的 ISO 文件复制到 HTML 目录中：

```
# cp -r /var/www/html/redhat/iso/images /var/www/html/redhat/  
# cp -r /var/www/html/redhat/iso/EFI /var/www/html/redhat/
```

7. 使引导装载程序配置可编辑：

```
# chmod 644 /var/www/html/redhat/EFI/BOOT/grub.cfg
```

8. 编辑 **/var/www/html/redhat/EFI/BOOT/grub.cfg** 文件，并将其内容替换为以下内容：

```
set default="1"  
  
function load_video {  
    insmod efi_gop  
    insmod efi_uga  
    insmod video_bochs  
    insmod video_cirrus  
    insmod all_video  
}  
  
load_video  
set gfxpayload=keep  
insmod gzio
```

```

insmod part_gpt
insmod ext2

set timeout=60
# END /etc/grub.d/00_header #

search --no-floppy --set=root -l 'RHEL-9-3-0-BaseOS-x86_64'

# BEGIN /etc/grub.d/10_linux #
menuentry 'Install Red Hat Enterprise Linux 9.3' --class fedora --class gnu-linux --class gnu --
class os {
    linuxefi ../../images/pxeboot/vmlinuz inst.repo=http://192.168.124.2/redhat/iso quiet
    initrdefi ../../images/pxeboot/initrd.img
}
menuentry 'Test this media & install Red Hat Enterprise Linux 9.3' --class fedora --class gnu-
linux --class gnu --class os {
    linuxefi ../../images/pxeboot/vmlinuz inst.repo=http://192.168.124.2/redhat/iso quiet
    initrdefi ../../images/pxeboot/initrd.img
}
submenu 'Troubleshooting -->' {
    menuentry 'Install Red Hat Enterprise Linux 9.3 in text mode' --class fedora --class gnu-
linux --class gnu --class os {
        linuxefi ../../images/pxeboot/vmlinuz inst.repo=http://192.168.124.2/redhat/iso inst.text
quiet
        initrdefi ../../images/pxeboot/initrd.img
    }
    menuentry 'Rescue a Red Hat Enterprise Linux system' --class fedora --class gnu-linux --
class gnu --class os {
        linuxefi ../../images/pxeboot/vmlinuz inst.repo=http://192.168.124.2/redhat/iso inst.rescue
quiet
        initrdefi ../../images/pxeboot/initrd.img
    }
}
}

```

在这个文件中，替换以下字符串：

### ***rhel-9-3-0-BaseOS-x86\_64*** 和 ***Red Hat Enterprise Linux 9.3***

编辑版本号，以使其与您下载的 RHEL 版本匹配。

### ***192.168.124.2***

使用服务器的 IP 地址替换。

9. 使 EFI 引导文件可执行：

```
# chmod 755 /var/www/html/redhat/EFI/BOOT/BOOTX64.EFI
```

10. 在防火墙中打开端口，以允许 HTTP (80)、DHCP (67, 68)和 DHCPv6 (546, 547)流量：

```
# firewall-cmd --zone public \
--add-port={80/tcp,67/udp,68/udp,546/udp,547/udp}
```

这个命令启用了临时访问，直到下次服务器重启为止。

11. 可选：要启用永久访问，请在命令中添加-- **permanent** 选项。
12. 重新载入防火墙规则：

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

13. 启动 HTTP 服务器：

```
# systemctl enable --now httpd
```

14. 创建 **html** 目录，并使其内容可读和可执行：

```
# chmod -cR u=rwX,g=rX,o=rX /var/www/html
```

15. 恢复 **html** 目录的 SELinux 上下文：

```
# restorecon -FvR /var/www/html
```

## 第 7 章 准备一个 PXE 安装源

您必须在 PXE 服务器上配置 TFTP 和 DHCP，以启用 PXE 引导和网络安装。

### 7.1. 网络安装概述

网络安装允许您将 Red Hat Enterprise Linux 安装到可访问安装服务器的系统。网络安装至少需要两个系统：

#### 服务器

运行 DHCP 服务器、HTTP、HTTPS、FTP 或 NFS 的服务器以及 PXE 引导情况下的 TFTP 服务器。虽然每个服务器可以运行在不同的物理系统上，但本节中的流程假设单个系统正在运行所有服务器。

#### 客户端

要安装 Red Hat Enterprise Linux 的系统。安装开始后，客户端会查询 DHCP 服务器，从 HTTP 或 TFTP 服务器接收引导文件，并从 HTTP、HTTPS、FTP 或者 NFS 服务器下载安装镜像。与其它安装方法不同，客户端不需要任何物理引导介质来启动安装。

要从网络引导客户端，请在固件中或客户端上的快速引导菜单中启用网络引导。在某些硬件上，从网络引导的选项可能被禁用了，或者不可用。

准备使用 HTTP 或 PXE 从网络安装 Red Hat Enterprise Linux 的工作流步骤如下：

#### 流程

1. 将安装 ISO 镜像或者安装树导出到 NFS、HTTPS、HTTP 或者 FTP 服务器中。
2. 配置 HTTP 或 TFTP 服务器以及 DHCP 服务器，并在服务器上启动 HTTP 或 TFTP 服务。
3. 引导客户端并开始安装。

您可以在以下网络引导协议之间进行选择：

#### HTTP

如果您的客户端 UEFI 支持，红帽建议使用 HTTP 引导。HTTP 引导通常更加可靠。

#### PXE (TFTP)

客户端系统更广泛地支持 PXE 引导，但通过此协议发送引导文件可能会很慢，并导致超时失败。

#### 其他资源

- [准备基于网络的存储库](#)
- [Red Hat Satellite 产品文档](#)

### 7.2. 为网络引导配置 DHCPV4 服务器

在您的服务器上启用 DHCP 版本 4 (DHCPv4) 服务，以便它可以提供网络引导功能。

#### 先决条件

- 您通过 IPv4 协议准备网络安装。  
对于 IPv6，请参阅 [为网络引导配置 DHCPv6 服务器](#)。

- 查找服务器的网络地址。  
在以下示例中，服务器有一个具有此配置的网卡：

**IPv4 地址**

192.168.124.2/24

**IPv4 网关**

192.168.124.1

**流程**

1. 安装 DHCP 服务器：

```
yum install dhcp-server
```

2. 设置 DHCPv4 服务器。在 `/etc/dhcp/dhcpd.conf` 文件中输入以下配置。替换地址以匹配您的网卡。

```
option architecture-type code 93 = unsigned integer 16;

subnet 192.168.124.0 netmask 255.255.255.0 {
  option routers 192.168.124.1;
  option domain-name-servers 192.168.124.1;
  range 192.168.124.100 192.168.124.200;
  class "pxeclients" {
    match if substring (option vendor-class-identifier, 0, 9) = "PXEClient";
    next-server 192.168.124.2;
    if option architecture-type = 00:07 {
      filename "redhat/EFI/BOOT/BOOTX64.EFI";
    }
    else {
      filename "pxelinux/pxelinux.0";
    }
  }
  class "httpclients" {
    match if substring (option vendor-class-identifier, 0, 10) = "HTTPClient";
    option vendor-class-identifier "HTTPClient";
    filename "http://192.168.124.2/redhat/EFI/BOOT/BOOTX64.EFI";
  }
}
```

3. 启动 DHCPv4 服务：

```
# systemctl enable --now dhcpd
```

## 7.3. 为网络引导配置 DHCPV6 服务器

在您的服务器上启用 DHCP 版本 6 (DHCPv4) 服务，以便它可以提供网络引导功能。

**先决条件**

- 您通过 IPv6 协议准备网络安装。  
对于 IPv4，请参阅 [为网络引导配置 DHCPv4 服务器](#)。

- 查找服务器的网络地址。  
在以下示例中，服务器有一个具有此配置的网卡：

#### IPv6 地址

```
fd33:eb1b:9b36::2/64
```

#### IPv6 网关

```
fd33:eb1b:9b36::1
```

### 流程

1. 安装 DHCP 服务器：

```
yum install dhcp-server
```

2. 设置 DHCPv6 服务器。在 `/etc/dhcp/dhcpd6.conf` 文件中输入以下配置。替换地址以匹配您的网卡。

```
option dhcp6.bootfile-url code 59 = string;
option dhcp6.vendor-class code 16 = {integer 32, integer 16, string};

subnet6 fd33:eb1b:9b36::/64 {
    range6 fd33:eb1b:9b36::64 fd33:eb1b:9b36::c8;

    class "PXECClient" {
        match substring (option dhcp6.vendor-class, 6, 9);
    }

    subclass "PXECClient" "PXECClient" {
        option dhcp6.bootfile-url
        "tftp://[fd33:eb1b:9b36::2]/redhat/EFI/BOOT/BOOTX64.EFI";
    }

    class "HTTPClient" {
        match substring (option dhcp6.vendor-class, 6, 10);
    }

    subclass "HTTPClient" "HTTPClient" {
        option dhcp6.bootfile-url
        "http://[fd33:eb1b:9b36::2]/redhat/EFI/BOOT/BOOTX64.EFI";
        option dhcp6.vendor-class 0 10 "HTTPClient";
    }
}
```

3. 启动 DHCPv6 服务：

```
# systemctl enable --now dhcpd6
```

4. 如果 DHCPv6 数据包被防火墙中的 RP 过滤器丢弃了，请检查其日志。如果日志包含 `rpfilter_DROP` 条目，请使用 `/etc/firewalld/firewalld.conf` 文件中的以下配置禁用过滤器：

```
IPv6_rpfilter=no
```

## 7.4. 为基于 BIOS 的客户端配置 TFTP 服务器

您必须配置一个 TFTP 服务器和 DHCP 服务器，并在 PXE 服务器上为基于 BIOS 的 AMD 和 Intel 64 位系统启动 TFTP 服务。

### 流程

1. 以 root 用户身份，安装以下软件包：

```
# yum install tftp-server
```

2. 允许到防火墙中的 **tftp** 服务的传入连接：

```
# firewall-cmd --add-service=tftp
```

这个命令启用了临时访问，直到下次服务器重启为止。

3. 可选：要启用永久访问，请在命令中添加 **--permanent** 选项。  
根据安装 ISO 文件的位置，您可能需要允许 HTTP 或者其他服务的进入连接。
4. 从 DVD ISO 镜像文件中的 **SYSLINUX** 软件包访问 **pxelinux.0** 文件，其中 *my\_local\_directory* 是您创建的目录的名称：

```
# mount -t iso9660 /path_to_image/name_of_image.iso /mount_point -o loop,ro
```

```
# cp -pr /mount_point/BaseOS/Packages/syslinux-tftpboot-version-architecture.rpm  
/my_local_directory
```

```
# umount /mount_point
```

5. 解压软件包：

```
# rpm2cpio syslinux-tftpboot-version-architecture.rpm | cpio -dimv
```

6. 在 **tftpboot/** 中创建 **pxelinux/** 目录，并将该目录中的所有文件复制到 **pxelinux/** 目录中：

```
# mkdir /var/lib/tftpboot/pxelinux
```

```
# cp /my_local_directory/tftpboot/* /var/lib/tftpboot/pxelinux
```

7. 在 **pxelinux/** 目录中创建 **pxelinux.cfg/** 目录：

```
# mkdir /var/lib/tftpboot/pxelinux/pxelinux.cfg
```

8. 创建名为 **default** 的配置文件，并将其添加到 **pxelinux.cfg/** 目录中，如下例所示：

```
default vesamenu.c32  
prompt 1  
timeout 600  
  
display boot.msg
```

```

label linux
  menu label ^Install system
  menu default
  kernel images/RHEL-8/vmlinuz
  append initrd=images/RHEL-8/initrd.img ip=dhcp inst.repo=http://192.168.124.2/RHEL-
8/x86_64/iso-contents-root/
label vesa
  menu label Install system with ^basic video driver
  kernel images/RHEL-8/vmlinuz
  append initrd=images/RHEL-8/initrd.img ip=dhcp inst.xdriver=vesa nomodeset
inst.repo=http://192.168.124.2/RHEL-8/x86_64/iso-contents-root/
label rescue
  menu label ^Rescue installed system
  kernel images/RHEL-8/vmlinuz
  append initrd=images/RHEL-8/initrd.img inst.rescue
inst.repo=http://192.168.124.2/RHEL-8/x86_64/iso-contents-root/
label local
  menu label Boot from ^local drive
  localboot 0xffff

```



### 注意

- 安装程序在没有运行时镜像的情况下无法引导。使用 **inst.stage2** 引导选项指定镜像的位置。另外，您可以使用 **inst.repo=** 选项指定镜像和安装源。
- 与 **inst.repo** 一起使用的安装源位置必须包含有效的 **.treeinfo** 文件。
- 当您选择 RHEL8 安装 DVD 作为安装源时，**.treeinfo** 文件会指向 BaseOS 和 AppStream 软件仓库。您可以使用单个 **inst.repo** 选项来加载这两个软件仓库。

9. 创建一个子目录，以将引导镜像文件存储在 **/var/lib/tftpboot/** 目录中，并将引导镜像文件复制到目录。在本例中，目录为 **/var/lib/tftpboot/pxelinux/images/RHEL-8/**：

```

# mkdir -p /var/lib/tftpboot/pxelinux/images/RHEL-8/
# cp /path_to_x86_64_images/pxeboot/{vmlinuz,initrd.img}
/var/lib/tftpboot/pxelinux/images/RHEL-8/

```

10. 启动并启用 **tftp.socket** 服务：

```

# systemctl enable --now tftp.socket

```

PXE 引导服务器现在可以提供 PXE 客户端。您可以启动客户端，即您要安装 Red Hat Enterprise Linux 的系统，在提示指定引导源时选择 **PXE 引导**，并启动网络安装。

## 7.5. 为基于 UEFI 的客户端配置 TFTP 服务器

您必须配置一个 TFTP 服务器和 DHCP 服务器，并为基于 UEFI 的 AMD64、Intel 64 和 64 位 ARM 系统的 PXE 服务器启动 TFTP 服务。



## 重要

- Red Hat Enterprise Linux 8 UEFI PXE 引导支持基于 MAC 的 grub 菜单文件的小写文件格式。例如，grub2 的 MAC 地址文件格式为 **grub.cfg-01-aa-bb-cc-dd-ee-ff**

## 流程

- 以 root 用户身份，安装以下软件包：

```
# yum install tftp-server
```

- 允许到防火墙中的 **tftp** 服务的传入连接：

```
# firewall-cmd --add-service=tftp
```

这个命令启用了临时访问，直到下次服务器重启为止。

- 可选：要启用永久访问，请在命令中添加 **--permanent** 选项。  
根据安装 ISO 文件的位置，您可能需要允许 HTTP 或者其他服务的进入连接。

- 访问 DVD ISO 镜像中的 EFI 引导镜像文件：

```
# mount -t iso9660 /path_to_image/name_of_image.iso /mount_point -o loop,ro
```

- 复制 DVD ISO 镜像中的 EFI 引导镜像：

```
# mkdir /var/lib/tftpboot/redhat  
# cp -r /mount_point/EFI /var/lib/tftpboot/redhat/  
# umount /mount_point
```

- 修复复制的文件的权限：

```
# chmod -R 755 /var/lib/tftpboot/redhat/
```

- 使用以下示例替换 **/var/lib/tftpboot/redhat/EFI/BOOT/grub.cfg** 的内容：

```
set timeout=60  
menuentry 'RHEL 8' {  
    linux images/RHEL-8/vmlinuz ip=dhcp inst.repo=http://192.168.124.2/RHEL-8/x86_64/iso-  
    contents-root/  
    initrd images/RHEL-8/initrd.img  
}
```



## 注意

- 安装程序在没有运行时镜像的情况下无法引导。使用 **inst.stage2** 引导选项指定镜像的位置。另外，您可以使用 **inst.repo=** 选项指定镜像和安装源。
- 与 **inst.repo** 一起使用的安装源位置必须包含有效的 **.treeinfo** 文件。
- 当您选择 RHEL8 安装 DVD 作为安装源时，**.treeinfo** 文件会指向 BaseOS 和 AppStream 软件仓库。您可以使用单个 **inst.repo** 选项来加载这两个软件仓库。

8. 创建一个子目录，以将引导镜像文件存储在 `/var/lib/tftpboot/` 目录中，并将引导镜像文件复制到目录。在这个示例中，目录为 `/var/lib/tftpboot/images/RHEL-8/`：

```
# mkdir -p /var/lib/tftpboot/images/RHEL-8/
# cp /path_to_x86_64_images/pxeboot/{vmlinuz,initrd.img}/var/lib/tftpboot/images/RHEL-8/
```

9. 启动并启用 **tftp.socket** 服务：

```
# systemctl enable --now tftp.socket
```

PXE 引导服务器现在可以提供 PXE 客户端。您可以启动客户端，即您要安装 Red Hat Enterprise Linux 的系统，在提示指定引导源时选择 **PXE 引导**，并启动网络安装。

## 其它资源

- [使用 Shim 程序](#)

## 7.6. 为 IBM POWER 系统配置网络服务器

您可以使用 GRUB2 为 IBM Power 系统配置网络引导服务器。

### 流程

1. 以 root 用户身份，安装以下软件包：

```
# yum install tftp-server dhcp-server
```

2. 在防火墙中，允许到 **tftp** 服务的进入连接：

```
# firewall-cmd --add-service=tftp
```

这个命令启用了临时访问，直到下次服务器重启为止。

3. 可选：要启用永久访问，请在命令中添加 **--permanent** 选项。  
根据安装 ISO 文件的位置，您可能需要允许 HTTP 或者其他服务的进入连接。

4. 在 TFTP root 中创建一个 GRUB2 网络引导目录：

```
# grub2-mknetdir --net-directory=/var/lib/tftpboot
Netboot directory for powerpc-ieee1275 created. Configure your DHCP server to point to
/boot/grub2/powerpc-ieee1275/core.elf
```

命令输出会告诉您 DHCP 配置中需要配置的文件名，如下所述。

- a. 如果 PXE 服务器运行在 x86 机器上，则必须在 tftp root 中创建 **GRUB2** 网络引导目录前安装 **grub2-ppc64-modules** :

```
# yum install grub2-ppc64-modules
```

5. 创建一个 GRUB2 配置文件：`/var/lib/tftpboot/boot/grub2/grub.cfg`，如下例中所示：

```
set default=0
set timeout=5

echo -e "\nWelcome to the Red Hat Enterprise Linux 8 installer!\n\n"

menuentry 'Red Hat Enterprise Linux 8' {
  linux grub2-ppc64/vmlinuz ro ip=dhcp inst.repo=http://192.168.124.2/RHEL-8/x86_64/iso-
  contents-root/
  initrd grub2-ppc64/initrd.img
}
```



### 注意

- 安装程序在没有运行时镜像的情况下无法引导。使用 **inst.stage2** 引导选项指定镜像的位置。另外，您可以使用 **inst.repo=** 选项指定镜像和安装源。
- 与 **inst.repo** 一起使用的安装源位置必须包含有效的 **.treeinfo** 文件。
- 当您选择 RHEL8 安装 DVD 作为安装源时，**.treeinfo** 文件会指向 BaseOS 和 AppStream 软件仓库。您可以使用单个 **inst.repo** 选项来加载这两个软件仓库。

6. 使用以下命令挂载 DVD ISO 镜像：

```
# mount -t iso9660 /path_to_image/name_of_iso/ /mount_point -o loop,ro
```

7. 创建一个目录，并将 **initrd.img** 和 **vmlinuz** 文件从 DVD ISO 镜像复制到其中，例如：

```
# cp /mount_point/ppc/ppc64/{initrd.img,vmlinuz} /var/lib/tftpboot/grub2-ppc64/
```

8. 将您的 DHCP 服务器配置为使用与 **GRUB2** 打包的引导镜像，如下例所示。如果您已经配置了 DHCP 服务器，请在 DHCP 服务器上执行此步骤。

```
subnet 192.168.0.1 netmask 255.255.255.0 {
  allow bootp;
  option routers 192.168.0.5;
  group { #BOOTP POWER clients
    filename "boot/grub2/powerpc-ieee1275/core.elf";
    host client1 {
      hardware ethernet 01:23:45:67:89:ab;
      fixed-address 192.168.0.112;
    }
  }
}
```

- 调整示例参数 **subnet**、**netmask**、**routers**、**fixed-address** 和 **hardware ethernet** 以适合您的网络配置。 **file name** 参数；这是 **grub2-mknetdir** 命令在此流程前面输出的文件名。
- 在 DHCP 服务器上，启动并启用 **dhcpd** 服务。如果您已在 localhost 上配置了 DHCP 服务器，请在 localhost 上启动并启用 **dhcpd** 服务。

```
# systemctl enable --now dhcpd
```

- 启动并启用 **tftp.socket** 服务：

```
# systemctl enable --now tftp.socket
```

PXE 引导服务器现在可以提供 PXE 客户端。您可以启动客户端，即您要安装 Red Hat Enterprise Linux 的系统，在提示指定引导源时选择 **PXE 引导**，并启动网络安装。

## 第 8 章 准备一个启用了 UEFI 安全引导的系统，来安装和引导 RHEL BETA 版本

要增强操作系统的安全性，在启用了 UEFI 安全引导的系统上引导 Red Hat Enterprise Linux Beta 版本时，对签名验证使用 UEFI 安全引导特性。

### 8.1. UEFI 安全引导和 RHEL BETA 版本

UEFI 安全引导要求操作系统内核使用可识别的私钥进行签名。UEFI 安全引导然后使用对应的公钥验证签名。

对于 Red Hat Enterprise Linux Beta 版，内核使用特定于红帽 Beta 的私钥进行签名。UEFI 安全引导尝试使用对应的公钥验证签名，但由于硬件无法识别 Beta 私钥，因此 Red Hat Enterprise Linux Beta 版本系统无法引导。因此，要在 Beta 版本中使用 UEFI 安全引导，请使用 Machine Owner Key (MOK) 功能将红帽 Beta 公钥添加到您的系统中。

### 8.2. 为 UEFI 安全引导添加 BETA 公钥

本节介绍如何为 UEFI 安全引导添加 Red Hat Enterprise Linux Beta 公钥。

#### 先决条件

- UEFI 安全引导已在系统中禁用。
- 已安装 Red Hat Enterprise Linux Beta 版本，即使系统重启，安全引导也会被禁用。
- 您已登录到系统，**Initial Setup** 窗口中的任务已完成。

#### 流程

1. 开始将红帽 Beta 公钥注册到系统机器所有者密钥 (MOK) 列表中：

```
# mokutil --import /usr/share/doc/kernel-keys/$(uname -r)/kernel-signing-ca.cer
```

**\$(uname -r)** 替换为内核版本 - 例如：**4.18.0-80.el8.x86\_64**。

2. 出现提示时输入密码。
3. 重新启动系统并按任意键继续启动。Shim UEFI 密钥管理实用程序在系统启动期间启动。
4. 选择 **Enroll MOK**。
5. 选择 **Continue**。
6. 选 **Yes** 并输入密码。密钥导入到系统的固件中。
7. 选择 **Reboot**。
8. 在系统上启用安全引导。

### 8.3. 删除 BETA 公钥

如果您计划删除 Red Hat Enterprise Linux Beta 版本，并安装 Red Hat Enterprise Linux 正式发行版本 (GA) 或一个不同的操作系统，请删除 Beta 公钥。

这个步骤描述了如何删除 Beta 公钥。

## 流程

1. 开始从系统的 Machine Owner Key (MOK) 列表中删除 Red Hat Beta 公钥：

```
# mokutil --reset
```

2. 出现提示时输入密码。
3. 重新启动系统并按任意键继续启动。Shim UEFI 密钥管理实用程序在系统启动期间启动。
4. 选择 **Reset MOK**。
5. 选择 **Continue**。
6. 选择 **Yes**，并输入在第 2 步中指定的密码。密钥已从系统的固件中删除。
7. 选择 **Reboot**。

## 第 9 章 在 64 位 IBM Z 上准备 RHEL 安装

这部分论述了如何在 64 位 IBM Z 构架中安装 Red Hat Enterprise Linux。

### 9.1. 规划在 64 位 IBM Z 上进行安装

Red Hat Enterprise Linux 8 运行在 z13 或更新的 IBM 大型机系统上。

安装过程假设您熟悉 64 位 IBM Z，并可设置 *逻辑分区* (LPAR)和 z/VM 客户机虚拟机。

对于在 64 位 IBM Z 上安装 Red Hat Enterprise Linux，红帽支持 Direct Access Storage Device(DASD)、Fiber Channel Protocol(FCP)存储设备以及 **virtio-blk** 和 **virtio-scsi** 设备。当使用 FCP 设备时，红帽建议在多路径配置中使用它们，以获得更好的可靠性。



#### 重要

DASD 是允许每个设备最多三个分区的磁盘。例如，**dasda** 可以有分区 **dasda1**、**dasda2** 和 **dasda3**。

#### 预安装决策

- 操作系统要在 LPAR、KVM 上运行还是作为一个 z/VM 客户机操作系统。
- 是否需要交换空间，及其大小。虽然建议为 z/VM 虚拟机分配足够的内存，并让 z/VM 执行必要的交换操作，但有些情况下，需要的 RAM 量难以预测。此类实例应逐一检查。
- 网络配置。用于 64 位 IBM Z 的 Red Hat Enterprise Linux 8 支持以下网络设备：
  - 真实的和虚拟的 *Open Systems Adapter* (OSA)
  - 真实的和虚拟的 HiperSocket
  - 真实 OSA 的 LAN 通道站 (LCS)
  - **virtio-net** 设备
- 确保为 z/VM 虚拟机将机器类型选择为 **ESA**，因为选择任何其他机器类型可能会阻止 RHEL 安装。请参阅 [IBM 文档](#)。

#### 磁盘空间

您需要在 DASD 或者 SCSI 磁盘中分配足够的磁盘空间。

- 服务器安装至少需要 10 GiB，如果要安装所有软件包，则需要 20 GiB。
- 任何应用程序数据都需要磁盘空间。安装后，您可以添加或者删除多个 DASD 或者 SCSI 磁盘分区。
- 新安装的 Red Hat Enterprise Linux 系统（Linux 实例）所使用的磁盘空间必须与系统上安装的其他操作系统所使用的磁盘空间分开。

#### RAM

确定您的系统有足够的可用 RAM：

- 从 NFS 安装时最小 1.5 GiB。

- 从 HTTP 或 FTP 安装源安装时，最小 3 GiB。
- 在文本模式下安装时，只有在您使用 NFS 安装源时，1GiB 才足够。
- 对于安装的 Linux 实例，红帽建议使用 2 GiB。但是，在适当调优的系统上，1GiB 就足够了。



### 注意

- 在使用 **SWAPGEN** 工具程序在一个 Fixed Block Architecture (FBA) DASD 上初始化交换空间时，需要使用 **FBAPART** 选项。
- 有关 IBM Power 服务器上的安装说明，请参阅 [IBM 安装文档](#)。要确保您的系统支持安装 RHEL，请参阅 <https://catalog.redhat.com> 和 <https://access.redhat.com/articles/rhel-limits>。

### 其它资源

- 有关 64 位 IBM Z 的详情，请参考 <https://www.ibm.com/it-infrastructure/z>。
- 有关在 IBM Z 上使用 Linux 安全引导的详情，请参考 [IBM Z 上的 Linux 安全引导](#)。

## 9.2. 64 位 IBM Z 服务器中的安装过程概述

您可以在 64 位 IBM Z 上以互动方式安装 Red Hat Enterprise Linux，也可以以无人值守模式安装 Red Hat Enterprise Linux。64 位 IBM Z 上的安装与其他架构不同，因为它通常通过网络执行，而不是从本地介质执行。该安装由三个阶段组成：

### 1. 引导安装

- 连接到大型机
- 自定义引导参数
- 执行初始程序加载(IPL)，或者从包含安装程序的介质引导

### 2. 连接到安装系统

- 从本地机器使用 SSH 连接到远程 64 位 IBM Z 系统，并使用虚拟网络计算(VNC)启动安装程序。

### 3. 使用 RHEL 安装程序完成安装

## 9.3. 在 64 位 IBM Z 服务器中安装 RHEL 的引导介质

与大型机建立连接后，您需要从包含安装程序的介质执行初始程序加载(IPL)或引导。本文档描述了在 64 位 IBM Z 上安装 Red Hat Enterprise Linux 的最常用方法。通常，任何方法都可用来引导 Linux 安装系统，它由一个内核(**kernel.img**)和初始 RAM 磁盘(**initrd.img**)以及 **generic.prm** 文件中的参数补充。另外，还会载入 **generic.ins** 文件，用来决定文件名，initrd 的内存地址、内核和 **generic.prm**。

在本文档中，Linux 安装系统也称为 *安装程序*。

您可以从何处启动 IPL 进程的控制点取决于 Linux 的运行环境。如果您的 Linux 作为 z/VM 客户机操作系统运行，则控制点为托管 z/VM 的 *控制程序* (CP)。如果您的 Linux 是在 LPAR 模式下运行，则控制点是大型机的 *Support Element* (SE) 或附加的 64 位 IBM Z *硬件管理控制台*(HMC)。

只有在 Linux 是作为客户机操作系统在 z/VM 下运行时才能使用以下引导介质：

- Z/VM 读取器

只有在 Linux 是以 LPAR 模式运行时，您才可以使用以下引导介质：

- 通过远程 FTP 服务器的 SE 或 HMC
- SE 或 HMC DVD

对于 z/VM 和 LPAR，您可以使用以下引导介质：

- DASD
- 通过 FCP 频道连接的 SCSI 磁盘设备
- 连接 FCP 的 SCSI DVD

如果您使用 DASD 或者 FCP 附加的 SCSI 磁盘设备作为引导介质，您必须有一个配置的 **zipl** 引导装载程序。

## 9.4. 自定义引导参数

在开始安装前，您必须配置一些必需的引导参数。通过 z/VM 安装时，在引导到 **generic.prm** 文件之前，必须先配置这些参数。在 LPAR 上安装时，**rd.cmdline** 参数默认设为 **ask**，这意味着会给您一个提示，让您输入这些引导参数。在这两种情况下，所需的参数都是相同的。



### 注意

现在，所有网络配置都必须使用参数文件，或者在提示符下指定。

### 安装源

必须配置一个安装源。使用 **inst.repo** 选项来指定安装的软件包源。

### 网络设备

如果在安装过程中需要访问网络，则必须提供网络配置。如果您计划只使用本地介质（如磁盘）执行无人值守（基于 Kickstart 的）安装，则可以省略网络配置。

#### ip=

对基本网络配置使用 **ip=** 选项，根据需要其他选项。

#### rd.znet=

还可使用 **rd.znet=** 内核选项，它采用网络协议类型、以逗号分隔的子通道列表，以及可选的逗号分隔的 **sysfs** 参数和值对。可多次指定这个参数激活多个网络设备。

例如：

```
rd.znet=qeth,0.0.0600,0.0.0601,0.0.0602,layer2=1,portname=<name>
```



### 注意

当指定多个 **rd.znet** 引导选项时，只会将最后一个选项传递给安装的系统的内核命令行。这不会影响系统的网络，因为在安装过程中配置的所有网络设备都会在引导时正确激活和配置。

qeth 设备驱动器为 Ethernet 和 Hipersockets 设备分配相同的接口名：**enc<device number>**。总线 ID 由通道子系统 ID、子通道集 ID 和设备号组成，由点分开；设备编号是总线 ID 的最后一部分，没有开头的零和点。例如，带有总线 ID **0.0.0a00** 的设备的接口名将会是 **enca00**。

## 存储设备

对于文本模式的安装，至少需要配置一个存储设备。

**rd.dasd=** 选项使用 Direct Access Storage Device (DASD) 适配器设备总线 ID。如果需要多个 DASD，可以多次指定参数，或使用逗号分开的总线 ID 列表。要指定一个 DASD 范围，指定第一个和最后一个总线 ID。

例如：

```
rd.dasd=0.0.0200 rd.dasd=0.0.0202(ro),0.0.0203(ro:failfast),0.0.0205-0.0.0207
```

**rd.zfcp=** 选项使用一个 SCSI over FCP (zFCP) 适配器设备总线标识符、一个全局端口名称 (WWPN)，以及一个 FCP LUN，然后激活该设备。可多次指定这个参数激活多个 zFCP 设备。

例如：从 8 开始，只有当 NPIV 模式中没有配置 **zFCP** 设备，或当 **auto LUN** 扫描被 **zfcp.allow\_lun\_scan=0** 内核模块参数禁用时，才必须提供一个目标全球端口名(WWPN)和 FCP LUN。它对使用指定的总线 ID 附加到 FCP 设备上的存储区域网络中发现的所有 SCSI 设备提供访问权限。此参数需要至少指定两次，才能激活指向同一磁盘的多个路径。

```
rd.zfcp=0.0.4000,0x5005076300C213e9,0x5022000000000000
rd.zfcp=0.0.4000
```

## Kickstart 选项

如果您使用 Kickstart 文件执行自动安装，则必须使用 **inst.ks=** 选项指定 Kickstart 文件的位置。对于无人参与的全自动 Kickstart 安装，**inst.cmdline** 选项也很有用。

包含所有强制参数的自定义 **generic.prm** 示例文件类似如下：

### 例 9.1. 自定义 generic.prm 文件

```
ro ramdisk_size=40000 cio_ignore=all,!lcondev
inst.repo=http://example.com/path/to/repository
rd.znet=qeth,0.0.0600,0.0.0601,0.0.0602,layer2=1,portno=0,portname=foo
ip=192.168.17.115::192.168.17.254:24:foobar.systemz.example.com:enc600:none
nameserver=192.168.17.1
rd.dasd=0.0.0200 rd.dasd=0.0.0202
rd.zfcp=0.0.4000,0x5005076300c213e9,0x5022000000000000
rd.zfcp=0.0.5000,0x5005076300dab3e9,0x5022000000000000
inst.ks=http://example.com/path/to/kickstart
```

有些安装方法还需要一个文件，其中包含 DVD 或者 FTP 服务器文件系统中安装数据位置的映射以及要复制数据的内存位置。

该文件通常命名为 **generic.ins**，其中包含初始 RAM 磁盘、内核镜像和参数文件(**generic.prm**)的文件名，以及每个文件的内存位置。一个 **generic.ins** 示例类似如下：

### 例 9.2. generic.ins 文件示例

```
images/kernel.img 0x00000000
images/initrd.img 0x02000000
images/genericdvd.prm 0x00010480
images/initrd.addrsize 0x00010408
```

红帽提供了一个有效的 **generic.ins** 文件，以及其他引导安装程序所需文件。例如，仅当您想要加载与默认内核版本不同的内核版本时，才修改此文件。

## 其他资源

- [安装源引导选项](#)。

## 9.5. 64 位 IBM Z 中的参数和配置文件

这部分包含有关 64 位 IBM Z 中的参数和配置文件的信息。

### 9.5.1. 64 位 IBM Z 所需的配置文件参数

需要几个参数，且必须包含在参数文件中。这些参数也包括在安装 DVD 的 **images/** 目录的 **generic.prm** 文件中。

- **ro**  
挂载 root 文件系统，它是一个 RAM 磁盘且只读。
- **ramdisk\_size=size**  
修改为 RAM 磁盘保留的内存大小，以确保其适合 Red Hat Enterprise Linux 安装程序。例如：  
**ramdisk\_size=40000**。

**generic.prm** 文件还包括额外参数 **cio\_ignore=all,!icondev**。对于多种设备，这个设置可以加快引导和设备探测的速度。安装程序以透明的方式处理被忽略的设备的激活。

### 9.5.2. 64 位 IBM Z/VM 配置文件

在 z/VM 中，您可以使用 CMS 格式化磁盘中的配置文件。CMS 配置文件的目的是将配置初始网络设置、DASD 和 FCP 规格的参数从参数文件中移出，从而在参数文件中节省空间。

CMS 配置文件的每一行包含一个变量及其关联的值，采用以下 shell 样式语法：**variable=value**。

您还必须在参数文件中添加 **CMSDASD** 和 **CMSCONFFILE** 参数。这些参数将安装程序指向配置文件：

#### **CMSDASD=cmsdasd\_address**

其中 *cmsdasd\_address* 是包含配置文件 CMS 格式化磁盘的设备号。它通常是 CMS 用户的 **A** 磁盘。  
例如：**CMSDASD=191**

#### **CMSCONFFILE=configuration\_file**

其中 *configuration\_file* 是配置文件的名称。这个值必须使用小写字母指定。它使用 Linux 文件名格式指定：**CMS\_file\_name.CMS\_file\_type**。

CMS 文件 **REDHAT CONF** 被指定为 **redhat.conf**。CMS 文件名和文件类型可以是 1-8 个字符，使用 CMS 规则。

例如：**CMSCONFFILE=redhat.conf**

### 9.5.3. 在 64 位 IBM Z 中安装网络、DASD 和 FCP 参数

这些参数可用于自动设置初始网络，可以在 CMS 配置文件中定义。这些参数是唯一可用于 CMS 配置文件的参数。其它章节中的参数必须在参数文件中指定。

#### **NETTYPE="type"**

其中 *type* 必须是以下之一：**qeth**、**lcs** 或者 **ctc**。默认为 **qeth**。  
为以下选择 **lcs**:

- OSA-Express 功能

为以下选择 **qeth**:

- OSA-Express 功能
- HiperSockets
- z/VM 上的虚拟连接，包括 VSWTICH 和 Guest LAN

#### **SUBCHANNELS="device\_bus\_IDs"**

其中 *device\_bus\_IDs* 是一个用逗号分开的两个或者三个设备总线 ID 的列表。ID 必须用小写来指定。为各类网络接口提供所需的设备总线 ID:

```
qeth: SUBCHANNELS="read_device_bus_id,write_device_bus_id,data_device_bus_id"
lcs or ctc: SUBCHANNELS="read_device_bus_id,write_device_bus_id"
```

例如（一个 qeth SUBCHANNEL 声明示例）：

```
SUBCHANNELS="0.0.f5f0,0.0.f5f1,0.0.f5f2"
```

#### **PORTNAME="osa\_portname" PORTNAME="lcs\_portnumber"**

该变量支持在 qdio 模式或非 qdio 模式中操作的 OSA 设备。

使用 qdio 模式(**NETTYPE="qeth"**)时，*osa\_portname* 是 OSA 设备在 qeth 模式下运行时指定的端口名称。

使用非 qdio 模式(**NETTYPE="lcs"**)时，*lcs\_portnumber* 用于将相对端口号作为 0 到 15 范围内的十进制整数来传递。

#### **PORTNO="portnumber"**

您可以将 **PORTNO="0"**（使用端口 0）或 **PORTNO="1"**（使用 OSA 功能的端口 1，每个 CHPID 有两个端口）添加到 CMS 配置文件，以避免提示模式。

#### **LAYER2="value"**

其中 *value* 可以是 **0** 或 **1**。

使用 **LAYER2="0"**，在第 3 层模式(**NETTYPE="qeth"**)中操作 **OSA** 或 **HiperSockets** 设备。对于 2 层模式，使用 **LAYER2="1"**。对于 z/VM 下的虚拟网络设备，此设置必须与设备耦合成的 GuestLAN 或 VSWITCH 的定义匹配。

要使用在第 2 层（数据链路层或其 MAC 子层）上运行的网络服务，如 DHCP，第 2 层模式是一个不错的选择。

现在 OSA 设备的默认 qeth 设备驱动程序为 2 层模式。要继续使用之前的默认层 3 模式，请分别设定 **LAYER2="0"**。

**VSWITCH="value"**

其中 *value* 可以是 **0** 或 **1**。

当连接到 z/VM VSWITCH 或 GuestLAN，指定 **VSWITCH="1"**；当使用直接附加的实际 OSA 或直接附加的实际 HiperSockets 时，指定 **VSWITCH="0"**（或完全不指定）。

**MACADDR="MAC\_address"**

如果指定了 **LAYER2="1"** 和 **VSWITCH="0"**，您可以可选地使用此参数来指定 MAC 地址。Linux 需要六个以冒号分隔的八位字节，作为小写的十六进制数对 - 例如 **MACADDR=62:a3:18:e7:bc:5f**。这与 z/VM 使用的表示法不同。

如果指定了 **LAYER2="1"** 和 **VSWITCH="1"**，必须指定 **MACADDR**，因为 z/VM 为层 2 模式中的虚拟网络设备分配了一个唯一 MAC 地址。

**CTCProt="value"**

其中 *value* 可以是 **0**、**1** 或者 **3**。

为 **NETTYPE="ctc"** 指定 CTC 协议。默认值为 **0**。

**HOSTNAME="string"**

其中 *string* 是新安装的 Linux 实例的主机名。

**IPADDR="IP"**

其中 *IP* 是新的 Linux 实例的 IP 地址。

**NETMASK="netmask"**

其中 *netmask* 是子网掩码。

子网掩码支持 IPv4 CIDR (*classless interdomain routing*) 中指定的前缀整数（从 1 到 32）语法。例如：您可以使用 **24** 来代表 **255.255.255.0**，或者使用 **20** 代表 **255.255.240.0**。

**GATEWAY="gw"**

其中 *gw* 是这个网络设备的网关 IP 地址。

**MTU="mtu"**

其中 *mtu* 是这个网络设备的 *最大传输单元* (MTU)。

**DNS="server1:server2:additional\_server\_terms:serverN"**

这里的 "server1:server2:additional\_server\_terms:serverN" 是用分号间隔的 DNS 服务器列表。例如：

```
DNS="10.1.2.3:10.3.2.1"
```

**SEARCHDNS="domain1:domain2:additional\_dns\_terms:domainN"**

这里的 "domain1:domain2:additional\_dns\_terms:domainN" 是用分号间隔的搜索域列表。例如：

```
SEARCHDNS="subdomain.domain:domain"
```

如果您指定了 **DNS=** 参数，则只需要指定 **SEARCHDNS=**。

**DASD=**

为安装定义 DASD 或者 DASD 范围。

安装程序支持以逗号分隔的设备总线 ID 列表，或者带有可选属性 **ro**、**diag**、**erplog** 和 **failfast** 的设备总线 ID 的范围。（可选）您可以将设备总线 ID 缩写为去掉前面的零的设备号。所有可选属性都应用冒号隔开，并使用括号括起来。可选属性跟随设备总线 ID 或设备总线 ID 范围。

唯一支持的全局选项是 **autodetect**。这不支持对不存在的 DASD 的规范，来为以后添加 DASD 保留内核设备。使用持久的 DASD 设备名称，如 **/dev/disk/by-path/name**，来启用透明磁盘添加。安装程序不支持其他全局选项，如 **probeonly**、**nopav** 或 **nofcx**。

仅指定需要在您的系统上安装的 DASD。在此指定的未格式化的 DASD 必须在安装程序确认后进行格式化。

在安装后添加 root 文件系统或 **/boot** 分区不需要的任何数据 DASD。

例如：

```
DASD="eb1c,0.0.a000-0.0.a003,eb10-eb14(diag),0.0.ab1c(ro:diag)"
```

#### **FCP\_n="device\_bus\_ID [WWPN FCP\_LUN]"**

对于只有 FCP 的环境,请从 CMS 配置文件中删除 **DASD=** 选项以指示没有 DASD。

```
FCP_n="device_bus_ID [WWPN FCP_LUN]"
```

其中：

- *n* 通常是一个整数值（如 **FCP\_1** 或 **FCP\_2**），但可以是包含字母或数字字符或下划线的任意字符串。
- *device\_bus\_ID* 指定 FCP 设备的设备总线 ID,它代表 **主机总线适配器** (HBA)（例如：**0.0.fc00** 代表设备 fc00）。
- *WWPN* 是用于路由的全局端口名称（通常和多个路径一同使用），它使用 16 位的十六进制数值（例如：**0x50050763050b073d**）。
- *FCP\_LUN* 是指存储逻辑单元标识符，指定为 16 位十六进制值，右侧填充零（例如 **0x4020400100000000**）。



#### **注意**

当自动 LUN 扫描没有被 **zfcplib.allow\_lun\_scan=0** 内核模块参数禁用或当安装 RHEL-8.6 或旧版本时，如果 **zFCP** 设备没有在 NPIV 模式下配置，则必须提供目标全局端口名称 (WWPN) 和 FCP\_LUN。否则，仅有 **device\_bus\_ID** 值是必需的。

- 这些变量可用于使用 FCP 设备激活 FCP LUN 的系统，比如 SCSI 磁盘。额外的 FCP LUN 可在互动安装过程中激活，也可以使用 Kickstart 文件激活。数值示例类似如下：

```
FCP_1="0.0.fc00 0x50050763050b073d 0x4020400100000000"
FCP_2="0.0.4000"
```



#### **重要**

FCP 参数中使用的每个值（如 **FCP\_1** 或 **FCP\_2**）都特定于站点，通常由 FCP 存储管理员提供。

### 9.5.4. 64 位 IBM Z 上的 kickstart 安装的参数

以下参数可在参数文件中定义，但无法用于 CMS 配置文件。

**inst.ks=URL**

引用一个 Kickstart 文件，该文件通常位于用于 64 位 IBM Z 上安装 Linux 的网络中。使用包括 Kickstart 文件的文件名的完整路径替换 URL。这个参数使用 Kickstart 自动激活安装。

**inst.cmdline**

这需要使用回答所有问题的 Kickstart 文件进行安装，因为安装程序不支持在命令行模式下进行交互式用户输入。在使用 **inst.cmdline** 选项前，请确定您的 Kickstart 文件包含所有必需的参数。如果缺少所需的命令，安装将会失败。

**9.5.5. 64 位 IBM Z 中的其它参数**

以下参数可在参数文件中定义，但无法用于 CMS 配置文件。

**rd.live.check**

开启基于 ISO 的安装源的测试；例如，当将 **inst.repo=** 与本地磁盘上或使用 NFS 挂载的 ISO 一起使用时。

**inst.nompath**

禁用对多路径设备的支持。

**proxy=[protocol://][username[:password]@]host[:port]**

指定通过 HTTP、HTTPS 或者 FTP 进行安装的代理。

**inst.rescue**

引导进入到从 RAM 磁盘运行的救援系统，可用于修复和恢复已安装的系统。

**inst.stage2=URL**

指定包含 **install.img** 的树路径，而不是直接指定 **install.img**。否则，请遵循与 **inst.repo=** 相同的语法。如果指定了 **inst.stage2**，它通常优先于其他查找 **install.img** 的方法。但是，如果 Anaconda 在本地介质中找到 **install.img**，**inst.stage2** URL 将被忽略。

如果没有指定 **inst.stage2**，且无法在本地找到 **install.img**，Anaconda 会查看由 **inst.repo=** 或者 **method=** 指定的位置。

如果只给出 **inst.stage2=** 而没有 **inst.repo=** 或者 **method=**，Anaconda 会默认使用安装的系统所启用的任何资料库进行安装。

可以多次使用这个选项指定多个 HTTP、HTTPS 或者 FTP 源。HTTP、HTTPS 或者 FTP 路径会按顺序尝试，直到成功为止：

```
inst.stage2=http://hostname/path_to_install_tree/
inst.stage2=http://hostname/path_to_install_tree/
inst.stage2=http://hostname/path_to_install_tree/
```

**inst.syslog=IP/hostname[:port]**

将日志信息发送到远程 syslog 服务器。

此处描述的引导参数对于在 64 位 IBM Z 中安装和出现问题时很有用，但只有影响安装程序的子集。

**9.5.6. 64 位 IBM Z 中的参数文件和 CMS 配置文件示例**

要更改参数文件，可以从附带的 **generic.prm** 文件开始。

**generic.prm** 文件示例：

```
ro ramdisk_size=40000 cio_ignore=all,!condev
```

```
CMSDASD="191" CMSCONFFILE="redhat.conf"
inst.vnc
inst.repo=http://example.com/path/to/dvd-contents
```

配置 QETH 网络设备的 `redhat.conf` 文件示例（使用 `generic.prm` 中的 `CMSCONFFILE` 指定）：

```
NETTYPE="qeth"
SUBCHANNELS="0.0.0600,0.0.0601,0.0.0602"
PORTNAME="FOOBAR"
PORTNO="0"
LAYER2="1"
MACADDR="02:00:be:3a:01:f3"
HOSTNAME="foobar.systemz.example.com"
IPADDR="192.168.17.115"
NETMASK="255.255.255.0"
GATEWAY="192.168.17.254"
DNS="192.168.17.1"
SEARCHDNS="systemz.example.com:example.com"
DASD="200-203"
```

### 9.5.7. 在 64 位 IBM Z 中使用参数和配置文件

64 位 IBM Z 架构可以使用自定义参数文件将引导参数传递给内核和安装程序。

以下操作需要更改参数文件：

- 使用 Kickstart 进行无人操作。
- 选择安装程序的互动用户界面无法访问的非默认安装设置，比如救援模式。

在安装程序（Anaconda）启动前，可使用参数文件以非互动方式设置网络。

内核参数最多为 895 个字符再加上一个行尾字符。参数文件可以是变量或者固定的记录格式。固定记录格式会将每一行调整到固定的记录长度，从而会增加文件大小。如果安装程序无法识别 LPAR 环境中所有指定参数的问题，可以尝试将所有参数放在一行，或者每行都用一个空格开头和结束。

参数文件包含内核参数，比如 `ro`，以及安装进程的参数，比如 `vncpassword=test` 或 `vnc`。

## 9.6. 在 Z/VM 客户虚拟机中准备安装

使用 `x3270` 或 `c3270` 终端模拟器从其他 Linux 系统登录到 z/VM，或者在 64 位 IBM Z 硬件管理控制台 (HMC) 上使用 IBM 3270 终端模拟器。如果您正在运行微软 Windows 操作系统，则可以通过互联网搜索找到一些可用选项。还存在一个称为 `wc3270` 的 `c3270` 的空闲原生 Windows 端口。

确保为 z/VM 虚拟机将机器类型选择为 **ESA**，因为选择任何其他机器类型可能会阻止安装 RHEL。请参阅 [IBM 文档](#)。

### 流程

1. 登录到为 Linux 安装选择的 z/VM 虚拟机。
2. 可选：如果您的 3270 连接中断，且您无法再次登录，因为之前的会话仍然处于活跃状态，您可以通过在 z/VM 日志屏幕上输入以下命令来将旧会话替换为新会话：

```
logon user here
```

■  
+ 使用 z/VM 虚拟机的名称替换 *user*。根据是否使用外部安全管理程序（如 RACF），登录命令可能会有所不同。

1. 如果您还没有在客户端中运行 **CMS**（与 z/VM 一起提供的单用户操作系统），请输入以下命令来引导它：

```
cp ipl cms
```

2. 确保不要使用 CMS 磁盘，如将您的 A 磁盘（通常设备号为 0191）作为安装目标。要找出 CMS 在使用哪些磁盘，请使用以下查询：

```
query disk
```

3. 您可以使用以下 CP（z/VM 控制程序，即 z/VM hypervisor）查询命令来查找 z/VM 客户机虚拟机的设备配置：
  - a. 查询可用的主内存，这在 64 位 IBM Z 的术语中称为 *存储*。您的客户机应至少具有 1 GiB 主内存。

```
cp query virtual storage
```

- b. 根据类型查询可用的网络设备：

**osa**

OSA - CHPID 类型 OSD，实或虚（VSWITCH 或 GuestLAN），两者均在 QDIO 模式中

**hsi**

HiperSockets - CHPID 类型 IQD、实或虚（GuestLAN 类型 Hipers）

**lcs**

LCS - CHPID 类型 OSE

例如，要查询上述所有网络设备类型，请运行：

```
cp query virtual osa
```

- c. 查询可用的 DASD。只有那些标记为 **RW** 的读写模式才能用作安装目标：

```
cp query virtual dasd
```

- d. 查询可用的 FCP 设备(vHBA)：

```
cp query virtual fcp
```

## 部分 II. 完全安装 RHEL 并半自动化

## 第 10 章 自动化安装 workflow

Kickstart 安装可以使用本地 DVD、本地磁盘或 NFS、FTP、HTTP 或者 HTTPS 服务器来执行。这部分提供了 Kickstart 使用的高级概述。

1. 创建一个 Kickstart 文件。您可以手动写入该文件，复制手动安装后保存的 Kickstart 文件，或者使用在线生成器工具创建该文件，之后再编辑该文件。请参阅 [创建 Kickstart 文件](#)。
2. 使 Kickstart 文件对可移动介质、磁盘上的安装程序或者使用 HTTP (S)、FTP 或者 NFS 服务器的网络位置可用。请参阅 [将 Kickstart 文件添加到 UEFI HTTP 或 PXE 安装源](#) 或 [使 Kickstart 文件可供 RHEL 安装程序使用](#)。
3. 创建用来开始安装的引导介质。
4. 使安装源可供安装程序使用。请参阅 [为 Kickstart 安装创建安装源](#)。
5. 使用引导介质和 Kickstart 文件开始安装。请参阅 [启动 Kickstart 安装](#)。

如果 Kickstart 文件包含所有强制命令和部分，则会自动完成安装。如果缺少一个或多个这些强制部分，或者出现错误，则安装需要手动干预才能完成。

## 第 11 章 创建 KICKSTART 文件

您可以使用以下方法创建 Kickstart 文件：

- 使用在线 Kickstart 配置工具。
- 复制手动安装后创建的 Kickstart 文件。
- 手动编写整个 Kickstart 文件。
- 为 Red Hat Enterprise Linux 8 安装转换 Red Hat Enterprise Linux 7 Kickstart 文件。有关转换工具的更多信息，请参阅 [Kickstart 生成器实验室](#)。
- 如果是虚拟和云环境，请使用 Image Builder 创建自定义系统镜像。

某些高度特定的安装选项只能通过手动编辑 Kickstart 文件来配置。

### 11.1. 使用 KICKSTART 配置工具创建 KICKSTART 文件

有红帽客户门户网站帐户的用户可以使用客户门户网站 Labs 中的 Kickstart Generator 工具在线生成 Kickstart 文件。这个工具可帮助您完成基本配置，并下载得到的 Kickstart 文件。

#### 先决条件

- 您有一个红帽客户门户网站帐户，以及一个有效的红帽订阅。

#### 流程

1. 在 <https://access.redhat.com/labsinfo/kickstartconfig> 打开 Kickstart 生成器实验室信息页面。
2. 点击标题左侧的 **Go to Application** 按钮并等待下一个页面加载。
3. 从下拉菜单中选择 **Red Hat Enterprise Linux 8**，并等待页面更新。
4. 使用表单中的字段描述要安装的系统。  
您可以使用表单左侧的链接来快速导航表表单的部分内容。
5. 要下载生成的 Kickstart 文件，请点击本页顶部的红色 **下载** 按钮。  
浏览器会保存该文件。
6. 安装 **pykickstart** 软件包。

```
# yum install pykickstart
```

7. 对您的 Kickstart 文件运行 **ksvalidator**。

```
$ ksvalidator -v RHEL8 /path/to/kickstart.ks
```

使用要验证的 Kickstart 文件的路径替换 `/path/to/kickstart.ks`。

验证工具并不能保证安装过程可以成功。它只检查语法是否正确，且该文件不包含已弃用的选项。它不会验证 Kickstart 文件的 **%pre**、**%post** 和 **%packages** 部分。

### 11.2. 执行手动安装来创建 KICKSTART 文件

推荐的生成 Kickstart 文件的方法是使用手动安装 Red Hat Enterprise Linux 生成的文件。安装完成后，安装过程中进行的所有选择都会保存到名为 **anaconda-ks.cfg** 的 Kickstart 文件中，该文件位于安装的系统上的 **/root/** 目录中。您可以使用这个文件来重现安装。另外，复制此文件，根据需要进行修改，并使用得到的配置文件进行进一步安装。

## 流程

1. 安装 RHEL。详情请参阅 [从安装介质交互式安装 RHEL](#)。在安装过程中，创建一个具有管理员权限的用户。
2. 完成安装并重启系统。
3. 使用管理员帐户登录该系统。
4. 将文件 **/root/anaconda-ks.cfg** 复制到您选择的位置。该文件包含用户和密码的信息。

- 在终端显示文件内容：

```
# cat /root/anaconda-ks.cfg
```

您可以复制输出并保存到您选择的其他文件中。

- 要将文件复制到另一个位置，请使用文件管理器。请记住需要修改副本的权限，以便非 root 用户可读取该文件。

5. 安装 **pykickstart** 软件包。

```
# yum install pykickstart
```

6. 对您的 Kickstart 文件运行 **ksvalidator**。

```
$ ksvalidator -v RHEL8 /path/to/kickstart.ks
```

使用要验证的 Kickstart 文件的路径替换 */path/to/kickstart.ks*。



### 重要

验证工具并不能保证安装过程可以成功。它只检查语法是否正确，且该文件不包含已弃用的选项。它不会验证 Kickstart 文件的 **%pre**、**%post** 和 **%packages** 部分。

## 11.3. 从以前的 RHEL 安装转换为一个 KICKSTART 文件

您可以使用 Kickstart 转换器工具转换成一个 RHEL 7 Kickstart 文件，用于 RHEL 8 或 9 安装，或者转换成一个 RHEL 8 Kickstart 文件，用于 RHEL 9。有关工具以及如何使用它来转换 RHEL Kickstart 文件的详情，请参考 <https://access.redhat.com/labs/kickstartconvert/>。

## 流程

- 准备 kickstart 文件后，安装 **pykickstart** 软件包。

```
# yum install pykickstart
```

- 对您的 Kickstart 文件运行 **ksvalidator**。

```
$ ksvalidator -v RHEL8 /path/to/kickstart.ks
```

使用要验证的 Kickstart 文件的路径替换 `/path/to/kickstart.ks`。



### 重要

验证工具并不能保证安装过程可以成功。它只检查语法是否正确，且该文件不包含已弃用的选项。它不会验证 Kickstart 文件的 `%pre`、`%post` 和 `%packages` 部分。

## 11.4. 使用 IMAGE BUILDER 创建自定义镜像

您可以使用 Red Hat Image Builder 为虚拟和云部署创建自定义系统镜像。

有关使用镜像构建器创建自定义镜像的更多信息，请参阅 [制作自定义的 RHEL 系统镜像](#) 文档。

## 第 12 章 将 KICKSTART 文件添加到 UEFI HTTP 或 PXE 安装源中

在 Kickstart 文件就绪后，您可以使其用于目标系统上的安装。

### 12.1. 基于网络安装的端口

下表列出了必须在服务器上打开的端口，以便为每种基于网络的安装提供文件。

表 12.1. 基于网络安装的端口

使用的协议	打开端口
HTTP	80
HTTPS	443
FTP	21
NFS	2049, 111, 20048
TFTP	69

#### 其它资源

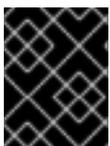
- [安全网络](#)

### 12.2. 在 NFS 服务器上共享安装文件

您可以在 NFS 服务器上存储 Kickstart 脚本文件。将其存储在 NFS 服务器上使您能够从单个源安装多个系统，而无需对 Kickstart 文件使用物理介质。

#### 先决条件

- 您对本地网络上的 Red Hat Enterprise Linux 8 服务器有管理员级别的访问权限。
- 要安装的系统可以连接到服务器。
- 服务器上的防火墙允许来自您要安装的系统连接。如需更多信息,请参阅 [用于基于网络的安装的端口](#)。



#### 重要

确保您在 `inst.ks` 和 `inst.repo` 中使用不同的路径。当使用 NFS 托管 Kickstart 时，您不能使用相同的 nfs 共享来托管安装源。

#### 流程

1. 作为 root 运行以下命令安装 `nfs-utils` 软件包：

```
# yum install nfs-utils
```

2. 将 Kickstart 文件复制到 NFS 服务器的目录中。
3. 使用文本编辑器打开 `/etc/exports` 文件，并使用以下语法添加一行：

```
/exported_directory/ clients
```

使用到包含 Kickstart 文件的目录的完整路径替换 `/exported_directory/`。要从这个 NFS 服务器安装计算机的主机名或 IP 地址替换 `clients`，如果您想允许任何可访问该 NFS 服务器的计算机都能使用该 ISO 镜像，则使用该系统所在的子网络访问该 ISO 镜像，或者使用星号 (\*)。有关此格式的信息，请参见 `exports(5)` 手册页。以下是一个基本配置，它使 `/rhel8-install/` 目录以只读方式对所有客户端可用：

```
/rhel8-install *
```

4. 保存 `/etc/exports` 文件并退出文本编辑器。
5. 启动 nfs 服务：

```
# systemctl start nfs-server.service
```

如果在更改 `/etc/exports` 文件前运行该服务，请输入以下命令以便运行的 NFS 服务器重新载入其配置：

```
# systemctl reload nfs-server.service
```

Kickstart 文件现在可以通过 NFS 访问并可用于安装。



### 注意

在指定 Kickstart 源时，使用 `nfs:` 协议、服务器的主机名或 IP 地址、冒号 (:) 以及包含该文件的目录内的路径。例如：如果服务器的主机名为 `myserver.example.com`，且您已将文件保存在 `/rhel8-install/my-ks.cfg` 中，指定 `inst.ks=nfs:myserver.example.com:/rhel8-install/my-ks.cfg` 作为安装源引导选项。

### 其他资源

- [使用 VNC 准备远程安装](#)

## 12.3. 在 HTTP 或 HTTPS 服务器上共享安装文件

您可以将 Kickstart 脚本文件存储在 HTTP 或 HTTPS 服务器上。在 HTTP 或 HTTPS 服务器中存储 Kickstart 文件可让您从一个源安装多个系统，而无需对 Kickstart 文件使用物理介质。

### 先决条件

- 您对本地网络上的 Red Hat Enterprise Linux 8 服务器有管理员级别的访问权限。
- 要安装的系统可以连接到服务器。
- 服务器上的防火墙允许来自您要安装的系统连接。如需更多信息，请参阅 [用于基于网络的安装的端口](#)。

### 流程

1. 要在 HTTP 上存储 Kickstart 文件，请安装 **httpd** 软件包：

```
# yum install httpd
```

要在 HTTPS 上存储 Kickstart 文件，请安装 **httpd** 和 **mod\_ssl** 软件包：

```
# yum install httpd mod_ssl
```



### 警告

如果您的 Apache 网页服务器配置启用了 SSL 安全性，请确定您只启用 TLSv1 协议，并禁用 SSLv2 和 SSLv3。这是因为 POODLE SSL 漏洞(CVE-2014-3566)。详情请查看 <https://access.redhat.com/solutions/1232413>。



### 重要

如果您使用带有自签名证书的 HTTPS 服务器，则必须使用 **inst.noverifyssl** 选项引导安装程序。

2. 将 Kickstart 文件复制到 HTTP(S) 服务器的 **/var/www/html/** 目录的子目录中。
3. 启动 httpd 服务：

```
# systemctl start httpd.service
```

Kickstart 文件现在可以访问并可用于安装。

当指定 Kickstart 文件的位置时，请使用 **http://** 或 **https://** 作为协议、服务器的主机名或 IP 地址，以及 Kickstart 文件的路径（相对于 HTTP 服务器 root）。例如：如果您使用 HTTP，则服务器的主机名为 **myserver.example.com**，并且您已将 Kickstart 文件复制为 **/var/www/html/rhel8-install/my-ks.cfg**，将 **http://myserver.example.com/rhel8-install/my-ks.cfg** 指定为文件的位置。

## 其它资源

- [部署不同类型的服务器](#)

## 12.4. 在 FTP 服务器上共享安装文件

您可以将 Kickstart 脚本文件存储在 FTP 服务器上。将脚本存储在 FTP 服务器上使您能够从单个源安装多个系统，而无需将物理介质用于 Kickstart 文件。

### 先决条件

- 您对本地网络上的 Red Hat Enterprise Linux 8 服务器有管理员级别的访问权限。
- 要安装的系统可以连接到服务器。

- 服务器上的防火墙允许来自您要安装的系统连接。如需更多信息，请参阅 [用于基于网络安装的端口](#)。

## 流程

1. 安装 **vsftpd** 软件包。请作为 root 运行以下命令完成此步骤：

```
# yum install vsftpd
```

2. 在文本编辑器中打开并编辑 `/etc/vsftpd/vsftpd.conf` 配置文件。
  - a. 将 **anonymous\_enable=NO** 行改为 **anonymous\_enable=YES**
  - b. 将 **write\_enable=YES** 行改为 **write\_enable=NO**。
  - c. 添加行 **pasv\_min\_port=*min\_port*** 和 **pasv\_max\_port=*max\_port***。使用被动模式下 FTP 服务器使用的端口范围替换 *min\_port* 和 *max\_port*，例如 **10021** 和 **10031**。在带有各种防火墙/NAT 设置的网络环境中，这个步骤可能是必要的。
  - d. 可选：在您的配置中添加自定义更改。可用选项请查看 **vsftpd.conf(5)** 手册页。这个步骤假设使用了默认选项。



### 警告

如果在 **vsftpd.conf** 文件中配置了 SSL/TLS 安全性，请确保只启用 TLSv1 协议，并禁用 SSLv2 和 SSLv3。这是因为 POODLE SSL 漏洞 (CVE-2014-3566)。详情请参考 <https://access.redhat.com/solutions/1234773>。

3. 配置服务器防火墙。

- a. 启用防火墙：

```
# systemctl enable firewalld
# systemctl start firewalld
```

- b. 在您的防火墙中启用前一步中的 FTP 端口和端口范围：

```
# firewall-cmd --add-port min_port-max_port/tcp --permanent
# firewall-cmd --add-service ftp --permanent
# firewall-cmd --reload
```

将 *min\_port-max\_port* 替换为您输入到 `/etc/vsftpd/vsftpd.conf` 配置文件的端口号。

4. 将 Kickstart 文件复制到 FTP 服务器的 `/var/ftp/` 目录或其子目录中。
5. 请确定在文件中正确设置了正确的 SELinux 上下文和访问模式：

```
# restorecon -r /var/ftp/your-kickstart-file.ks
# chmod 444 /var/ftp/your-kickstart-file.ks
```

## 6. 启动 **vsftpd** 服务：

```
# systemctl start vsftpd.service
```

如果在更改 `/etc/vsftpd/vsftpd.conf` 文件前运行该服务，重启该服务以载入经过编辑的文件：

```
# systemctl restart vsftpd.service
```

在引导过程中启用 **vsftpd** 服务：

```
# systemctl enable vsftpd
```

Kickstart 文件现在可以访问，并可用于同一网络中的系统安装。



### 注意

在配置安装源时，使用 **ftp://** 作为协议、服务器的主机名或 IP 地址以及 Kickstart 文件的路径（相对于 FTP 服务器 root）。例如：如果服务器的主机名是 **myserver.example.com**，且您已将文件复制到 `/var/ftp/my-ks.cfg` 中，请指定 **ftp://myserver.example.com/my-ks.cfg** 作为安装源。

## 第 13 章 半自动化安装：使 KICKSTART 文件可供 RHEL 安装程序使用

Kickstart 文件就绪后，就可以将其用于目标系统上的安装。

### 13.1. 在本地卷上共享安装文件

这个步骤描述了如何在要安装的系统中的卷中保存 Kickstart 脚本文件。这个方法可让您绕过其他系统的要求。

#### 先决条件

- 您有一个可移动到要安装的机器上的驱动器，如 USB 盘。
- 驱动器包含可由安装程序读取的分区。支持的类型有 **ext2**、**ext3**、**ext4**、**xfs** 和 **fat**。
- 该驱动器连接到系统，其卷已挂载。

#### 流程

1. 列出卷信息并记录您想要复制 Kickstart 文件的卷 UUID。

```
# lsblk -l -p -o name,rm,ro,hotplug,size,type,mountpoint,uuid
```

2. 导航到卷中的文件系统。
3. 将 Kickstart 文件复制到这个文件系统中。
4. 记录以后要与 **inst.ks=** 选项一起使用的字符串。这个字符串采用 **hd:UUID=volume-UUID:path/to/kickstart-file.cfg** 的形式。请注意，该路径相对于文件系统根目录，而不是文件系统层次结构的 `/root`。使用之前记录的 UUID 替换 `volume-UUID`。
5. 卸载所有驱动器卷：

```
# umount /dev/xyz ...
```

在命令中添加所有卷，用空格分开。

### 13.2. 在本地卷上共享安装文件，以便自动载入

在要安装的系统特殊命名的卷的根目录中会出现一个名为 Kickstart 文件。这可让您避开另一个系统的需要，让安装程序自动载入该文件。

#### 先决条件

- 您有一个可移动到要安装的机器上的驱动器，如 USB 盘。
- 驱动器包含可由安装程序读取的分区。支持的类型有 **ext2**、**ext3**、**ext4**、**xfs** 和 **fat**。
- 该驱动器连接到系统，其卷已挂载。

#### 流程

1. 列出您要将 Kickstart 文件复制到的卷信息。

```
# lsblk -l -p
```

2. 导航到卷中的文件系统。
3. 将 Kickstart 文件复制到文件系统的根目录下。
4. 将 Kickstart 文件重命名为 **ks.cfg**。
5. 将卷重命名为 **OEMDRV**:

- 对于 **ext2**、**ext3** 和 **ext4** 文件系统：

```
# e2label /dev/xyz OEMDRV
```

- 对于 XFS 文件系统：

```
# xfs_admin -L OEMDRV /dev/xyz
```

使用卷块设备的路径替换 `/dev/xyz`。

6. 卸载所有驱动器卷：

```
# umount /dev/xyz ...
```

在命令中添加所有卷，用空格分开。

## 第 14 章 启动 KICKSTART 安装

您可以以多种方式启动 Kickstart 安装：

- 通过在 PXE 引导中自动编辑引导选项。
- 通过使用特定名称在卷中自动提供文件。

您可以使用 Red Hat Content Delivery Network (CDN)注册 RHEL。CDN 是地理上分布的一系列 Web 服务器。例如，这些服务器向具有有效订阅的 RHEL 主机提供软件包和更新。

在安装过程中，从 CDN 注册和安装 RHEL 提供以下好处：

- 在安装后立即为最新系统使用最新的软件包，
- 对连接到 Red Hat Insights 和启用系统目提供集成的支持。

### 14.1. 使用 PXE 自动启动 KICKSTART 安装

AMD64、Intel 64 和 64 位 ARM 系统以及 IBM Power Systems 服务器能够使用 PXE 服务器引导。当您配置 PXE 服务器时，可以在引导装载程序配置文件中添加引导选项，以便您可以自动启动安装。使用这个方法，可以完全自动化安装，包括引导过程。

这个过程旨在作为一般参考。具体步骤因您的系统架构有所不同，且不是所有架构上都提供所有的选项（例如，在 64 位 IBM Z 上无法使用 PXE 引导）。

#### 先决条件

- 您有一个准备好的 Kickstart 文件，位于要安装的系统可访问的位置。
- 您有一个可用于引导系统并开始安装的 PXE 服务器。

#### 流程

1. 打开 PXE 服务器中的引导装载程序配置文件，并在适当的行中添加 **inst.ks=** 引导选项。该文件的名称及其句法取决于您系统的架构和硬件：

- 在使用 BIOS 的 AMD64 和 Intel 64 系统中，该文件名称可以是 `default`，或取决于您的系统 IP 地址。在这种情况下，在安装条目的附加行中添加 **inst.ks=** 选项。在配置文件中添加的行示例类似如下：

```
append initrd=initrd.img inst.ks=http://10.32.5.1/mnt/archive/RHEL-8/8.x/x86_64/kickstarts/ks.cfg
```

- 在使用 GRUB2 引导装载程序（带有 UEFI 固件和 IBM Power Systems 服务器的 AMD64、Intel 64 和 64 位 ARM 系统系统）中，该文件的名称将是 **grub.cfg**。在这个文件中，在安装条目的 `kernel` 行中添加 **inst.ks=** 选项。该配置文件 `kernel` 行示例类似如下：

```
kernel vmlinuz inst.ks=http://10.32.5.1/mnt/archive/RHEL-8/8.x/x86_64/kickstarts/ks.cfg
```

2. 从网络服务器引导安装。

现在开始使用在 Kickstart 文件中指定的安装选项安装。如果 Kickstart 文件有效并包含全部所需命令，则安装将是全自动的。



## 注意

如果您已在启用了 UEFI 安全引导的系统中安装了 Red Hat Enterprise Linux Beta 版本，然后将 Beta 公钥添加到系统的 Machine Owner Key (MOK) 列表中。

### 其他资源

- 有关设置 PXE 服务器的详情，请参考 [准备 PXE 安装源](#)

## 14.2. 使用本地卷自动启动 KICKSTART 安装

您可以通过在指定标记的存储卷上加上特定名称的 Kickstart 文件启动 Kickstart 安装。

### 先决条件

- 您有一个带有标签 **OEMDRV** 的卷，以及 root 中以 **ks.cfg** 形式存在的一个 Kickstart 文件。
- 安装程序启动时，包含此卷的驱动器在系统上可用。

### 流程

1. 使用本地介质（CD、DVD 或者 USB 闪存驱动器）引导系统。
2. 在 boot 提示符后指定所需的引导选项。
  - a. 如果所需的存储库位于网络位置，您可能需要使用 **ip=** 选项配置网络。在没有这个选项时，安装程序会默认尝试使用 DHCP 协议配置所有网络设备。
  - b. 要访问安装所需软件包的软件源，您可能需要添加 **inst.repo=** 选项。如果您没有指定这个选项，则必须在 Kickstart 文件中指定安装源。  
有关安装源的更多信息，请参阅 [用于安装程序配置和流控制的 Kickstart 命令](#)。
3. 通过确认您添加的引导选项启动安装。  
现在开始安装，Kickstart 文件会被自动探测，并用来自动启动 Kickstart 安装。



## 注意

如果您已在启用了 UEFI 安全引导的系统中安装了 Red Hat Enterprise Linux Beta 版本，然后将 Beta 公钥添加到系统的 Machine Owner Key (MOK) 列表中。有关 UEFI 安全引导和 Red Hat Enterprise Linux Beta 版本的更多信息，请参阅 [UEFI 安全引导和 Beta 版本要求](#)。

## 14.3. 在 IBM Z 中引导安装以便在 LPAR 中安装 RHEL

### 14.3.1. 从 FTP 服务器引导 RHEL 安装，以便在 IBM Z LPAR 中安装

使用 FTP 服务器将 Red Hat Enterprise Linux 安装到 LPAR 时使用这个步骤。

### 流程

1. 以足够特权的用户身份登录 IBM Z 硬件管理控制台(HMC)或支持元素(SE)，以便将新的操作系统安装到 LPAR。建议使用 **SYSPROG** 用户。

2. 在 **Systems** 选项卡中，选择您要使用的大型机，然后在 **Partitions** 选项卡中选择您要安装的 LPAR。
3. 在 **Daily** 屏幕底部，找到 **Operating System Messages**。双击 **Operating System Messages** 会显示将出现 Linux 引导信息的文本控制台。
4. 双击 **Load from Removable Media or Server**。
5. 在下面的对话框中，选择 **FTP 服务器** 并输入以下信息：
  - **Host Computer** - 要从其安装的 FTP 服务器的主机名或 IP 地址，例如 `ftp.redhat.com`
  - **用户 ID** - FTP 服务器上的用户名。或者，指定 `anonymous`。
  - **密码** - 您的密码。如果以匿名身份登录，使用您的电子邮件地址。
  - **文件位置（可选）** - FTP 服务器中为 IBM Z 保存 Red Hat Enterprise Linux 的目录，如 `/rhel/s390x/`。
6. 点 **Continue**。
7. 在随后的对话框中，保留默认选择的 `generic.ins`，然后点 **Continue**。

### 14.3.2. 从准备好的 DASD 引导 RHEL 安装，以便在 IBM Z LPAR 中安装

当使用已准备好的 DASD 将 Red Hat Enterprise Linux 安装到 LPAR 时执行这个步骤。

#### 流程

1. 以足够特权的用户身份登录 IBM Z 硬件管理控制台(HMC)或支持元素(SE)，以便将新的操作系统安装到 LPAR。建议使用 **SYSPROG** 用户。
2. 在 **Systems** 选项卡中，选择您要使用的大型机，然后在 **Partitions** 选项卡中选择您要安装的 LPAR。
3. 在 **Daily** 屏幕底部，找到 **Operating System Messages**。双击 **Operating System Messages** 会显示将出现 Linux 引导信息的文本控制台。
4. 双击 **Load**。
5. 在随后的对话框中，选择 **Normal** 作为 **Load type**。
6. 在 **Load address**，使用 DASD 的设备号。
7. 在 **Load parameter** 中，填写与您准备引导 Red Hat Enterprise Linux 安装程序的 `zipl` 引导菜单条目对应的数字。
8. 点**确定按钮**。

### 14.3.3. 从 FCP 附加 SCSI 磁盘引导 RHEL 安装，以便在 IBM Z LPAR 中安装

当使用准备好的连接 FCP 的 SCSI 磁盘将 Red Hat Enterprise Linux 安装到 LPAR 时，使用此流程。

#### 流程

1. 以足够特权的用户身份登录 IBM Z 硬件管理控制台(HMC)或支持元素(SE)，以便将新的操作系统安装到 LPAR。建议使用 **SYSPROG** 用户。
2. 在 **Systems** 选项卡中，选择您要使用的大型机，然后在 **Partitions** 选项卡中选择您要安装的 LPAR。
3. 在 **Daily** 屏幕底部，找到 **Operating System Messages**。双击 **Operating System Messages** 会显示将出现 Linux 引导信息的文本控制台。
4. 双击 **Load**。
5. 在随后的对话框中，选择 **SCSI** 作为 **Load type**。
6. 输入与 SCSI 磁盘连接的 FCP 通道的设备号来作为 **Load address**。
7. 以 16 位十六进制数填写包含磁盘的存储系统的 WWPN，来作为 **World wide port name**。
8. 在 **Logical unit number** 中输入 16 位十六进制数字到磁盘的 LUN。
9. 在 **Boot program selector** 中输入与您准备引导 Red Hat Enterprise Linux 安装程序的 **zipl** 引导菜单条目对应的数字。
10. 将 **Boot record logical block address** 设置为 **0**，**Operating system specific load parameters** 为空。
11. 点**确定按钮**。

#### 14.3.4. 从连接 FCP 的 SCSI DVD 驱动器引导 RHEL 安装，以便在 IBM Z LPAR 中安装

这需要一个连接到 FCP-to-SCSI 桥的 SCSI DVD 驱动器，该桥又连接到您系统 Z 机器中的 FCP 适配器。必须配置这个 FCP 适配器并使其可用。

##### 流程

1. 以有足够特权的用户身份登录 IBM System Z 硬件管理控制台(HMC)或支持元素(SE)，来将新的操作系统安装到 LPAR。建议使用 **SYSPROG** 用户。
2. 在 **System** 标签页中，选择您要使用的主框架，然后在 **Partitions** 标签中选择您要安装的 LPAR。
3. 在 **Daily** 屏幕底部，找到操作系统信息。双击 **Operating System Messages** 会显示将出现 Linux 引导信息的文本控制台。
4. 将用于 64 位 IBM Z 的 Red Hat Enterprise Linux DVD 插入到 DVD 驱动器中。
5. 双击 **Load**。
6. 在随后的对话框中，选择 **SCSI** 作为 **Load type**。
7. 在 **Load address** 中输入连接到 FCP-to-SCSI 桥接的 FCP 频道的设备号。
8. 在 **World wide port name** 中输入 16 位十六进制数字作为 FCP-to-SCSI 桥接的 WWPN。
9. 在 **Logical unit number** 中输入 16 位十六进制数字作为 DVD 驱动器的 LUN。
10. 作为 **Boot program selector**，输入数字 **1**，来在用于 64 位 IBM Z 的 Red Hat Enterprise Linux DVD 中选择引导条目。

11. 将 `Boot record logical block address` 设置为 `0`, `Operating system specific load parameters` 为空。
12. 点**确定**按钮。

## 14.4. 在 IBM Z 中引导安装以便在 Z/VM 中安装 RHEL

当在 z/VM 中安装时，您可以使用以下设备引导：

- z/VM 虚拟读取器
- DASD 或者 FCP 附加 SCSI 磁盘使用 `zipl` 引导装载程序准备
- 附加 FCP 的 SCSI DVD 驱动器

### 14.4.1. 使用 z/VM 读取器引导 RHEL 安装

执行以下步骤来从 z/VM 读取器引导：

#### 流程

1. 如有必要，将包含 z/VM TCP/IP 工具的设备添加到 CMS 磁盘列表中。例如：

```
cp link tcpmaint 592 592  
acc 592 fm
```

用任何 `FILEMODE` 字母替换 `fm`。

2. 执行该命令：

```
ftp host
```

其中 `host` 是提供引导镜像 (`kernel.img` 和 `initrd.img`) 的 FTP 服务器的主机名或 IP 地址。

3. 登录并执行以下命令。如果要覆盖现有的 `kernel.img`、`initrd.img`、`generic.prm` 或者 `redhat.exec` 文件，使用 `(repl` 选项：

```
cd //location/of/install-tree/images/  
ascii  
get generic.prm (repl  
get redhat.exec (repl  
locsite fix 80  
binary  
get kernel.img (repl  
get initrd.img (repl  
quit
```

4. 可选：使用 CMS 命令 `filelist` 来显示收到的文件及其格式，以检查文件是否已正确传输。`kernel.img` 和 `initrd.img` 有在 `Format` 列中用 `F` 表示，在 `Lrecl` 列中记录长度为 80 的固定记录长度格式很重要。例如：

```
VMUSER FILELIST A0 V 169 Trunc=169 Size=6 Line=1 Col=1 Alt=0  
Cmd Filename Filetype Fm Format Lrecl Records Blocks Date Time  
REDHAT EXEC B1 V 22 1 1 4/15/10 9:30:40
```

```

GENERIC PRM B1 V 44 1 1 4/15/10 9:30:32
INITRD IMG B1 F 80 118545 2316 4/15/10 9:30:25
KERNEL IMG B1 F 80 74541 912 4/15/10 9:30:17

```

按 **PF3** 键退出 filelist 并返回到 CMS 提示符。

5. 根据需要自定义 **generic.prm** 中的引导参数。详情请参阅 [自定义引导参数](#)。  
配置存储和网络设备的另外一种方法是使用 CMS 配置文件。在这种情况下，把 **CMSDASD=** 和 **CMSCONFFILE=** 参数添加到 **generic.prm**。如需了解更多详细信息，请参阅 [IBM Z/VM 配置文件](#)。
6. 最后，执行 REXX 脚本 redhat.exec 来引导安装程序：

```
redhat
```

#### 14.4.2. 使用准备的 DASD 引导 RHEL 安装

执行以下步骤使用准备的 DASD:

##### 流程

- 从准备的 DASD 引导,并选择指向 Red Hat Enterprise Linux 安装程序的 **zipl** 引导菜单条目。使用以下命令格式：

```
cp ipl DASD_device_number loadparm boot_entry_number
```

使用引导设备的设备号替换 *DASD\_device\_number*，使用这个设备的 **zipl** 配置菜单替换 *boot\_entry\_number*。例如：

```
cp ipl eb1c loadparm 0
```

#### 14.4.3. 使用准备好的连接 FCP 的 SCSI 磁盘引导 RHEL 安装

执行以下步骤来从准备好的连接 FCP 的 SCSI 磁盘引导：

##### 流程

1. 配置 z/VM 的 SCSI 引导加载程序，来访问 FCP 存储区域网络中准备好的 SCSI 磁盘。选择准备的 **zipl** 引导菜单中指向 Red Hat Enterprise Linux 安装程序的条目。使用以下命令格式：

```
cp set loaddev portname WWPN lun LUN bootprog boot_entry_number
```

使用存储系统的全球端口名称和带有磁盘逻辑单元号的 *LUN* 来替换 *WWPN*。16 位十六进制数必须分成两对，每对八位数。例如：

```
cp set loaddev portname 50050763 050b073d lun 40204011 00000000 bootprog 0
```

2. 可选：使用以下命令确认您的设置：

```
query loaddev
```

3. 使用以下命令引导与包含磁盘的存储系统连接的 FCP 设备：

```
cp ipl FCP_device
```

例如：

```
cp ipl fc00
```

#### 14.4.4. 使用附加 FCP 的 SCSI DVD 驱动器引导 RHEL 安装

执行以下步骤使用附加的 SCSI DVD 驱动器：

##### 先决条件

1. 这需要一个附加到 FCP-to-SCSI 桥接的 SCSI DVD 驱动器，该桥接又连接到您 64 位 IBM Z 中的 FCP 适配器。FCP 适配器必须在 z/VM 下配置并可用。

##### 流程

1. 将用于 64 位 IBM Z 的 Red Hat Enterprise Linux DVD 插入到 DVD 驱动器中。
2. 配置 z/VM 的 SCSI 引导装载程序以访问 FCP 存储区域网络中的 DVD 驱动器，并为用于 64 位 IBM Z 的 Red Hat Enterprise Linux DVD 上的引导条目指定 **1**。使用以下命令格式：

```
cp set loaddev portname WWPN lun FCP_LUN bootprog 1
```

使用 FCP-to-SCSI 桥接的 WWPN 替换 *WWPN*，使用 DVD 驱动器的 LUN 替换 *FCP\_LUN*。16 位十六进制数必须分成两对，每对八个字符。例如：

```
cp set loaddev portname 20010060 eb1c0103 lun 00010000 00000000 bootprog 1
```

3. 可选：使用以下命令确认您的设置：

```
cp query loaddev
```

4. 使用 FCP-to-SCSI 桥接连接的 FCP 设备中 IPL。

```
cp ipl FCP_device
```

例如：

```
cp ipl fc00
```

## 14.5. 安装过程中的控制台和日志记录

在主界面外，Red Hat Enterprise Linux 安装程序使用 **tmux** 终端显示和控制几个窗口。每个窗口都有不同的目的，它们会显示几个不同的日志，可用于在安装过程中排除问题。其中一个窗口提供带有 **root** 权限的互动 shell 提示符，除非使用一个引导选项或一个 Kickstart 命令特别禁用了这个提示符。

终端多路器在虚拟控制台 1 中运行。要从实际安装环境切换到 **tmux**，按 **Ctrl+Alt+F1**。要回到在虚拟控制台 6 中运行的主安装界面，按 **Ctrl+Alt+F6**。在文本模式安装过程中，在虚拟控制台 1 (**tmux**) 中启动，切换到控制台 6 将打开 shell 提示符而不是图形界面。

运行 **tmux** 的控制台有五个可用的窗口; 下表中描述了它们的内容以及键盘快捷键。请注意, 键盘快捷键有两个部分: 首先按 **Ctrl+b** 键, 然后释放这两个键, 再按您想要使用的窗口的数字键。

您还可以使用 **Ctrl+b n**、**Alt+ Tab** 和 **Ctrl+b p** 切换到下一个或前一个 **tmux** 窗口。

表 14.1. 可用的 **tmux** 窗口

快捷键	内容
<b>Ctrl+b 1</b>	安装程序主窗口。包含基于文本的提示（在文本模式安装或者使用 VNC 直接模式时），以及一些调试信息。
<b>Ctrl+b 2</b>	有 <b>root</b> 权限的互动 shell 提示符。
<b>Ctrl+b 3</b>	安装日志; 显示信息保存在 <b>/tmp/anaconda.log</b> 中。
<b>Ctrl+b 4</b>	存储日志; 显示与存储设备和配置相关的消息, 保存在 <b>/tmp/storage.log</b> 中。
<b>Ctrl+b 5</b>	程序日志; 显示安装过程中执行的实用程序的信息, 保存在 <b>/tmp/program.log</b> 中。

## 部分 III. 安装后的任务

## 第 15 章 使用订阅管理器注册 RHEL

安装后，您必须注册系统，以获得持续更新。

### 15.1. 使用安装程序 GUI 注册 RHEL 8

您可以使用 RHEL 安装程序 GUI 注册 Red Hat Enterprise Linux 8。

#### 先决条件

- 您在红帽客户门户网站中有一个有效的用户帐户。请参阅 [创建红帽登录页面](#)。
- 您有一个有效的激活码和机构 ID。

#### 流程

1. 从 **Installation Summary** 屏幕，在 **Software** 下，点击 **Connect to Red Hat**。
2. 使用 **Account** 或 **Activation Key** 选项激活您的红帽帐户。
3. 可选：在 **Set System Purpose** 字段中，选择您要从下拉菜单中设置的 **Role**、**SLA** 和 **Usage** 属性。  
此时您的 Red Hat Enterprise Linux 8 系统已被成功注册。

### 15.2. REGISTRATION ASSISTANT

Registration Assistant 可帮助为您的 Red Hat Enterprise Linux 环境选择最合适的注册选项。

#### 其他资源

- 有关使用用户名和密码注册 RHEL 订阅管理器客户端的帮助，请参阅客户门户网站中的 [RHEL 注册助手](#)。
- 有关将 RHEL 系统注册到 Red Hat Insights 的帮助，请参阅混合云控制台上的 [Insights 注册助手](#)。

### 15.3. 使用命令行注册您的系统

您可以使用命令行注册 Red Hat Enterprise Linux 8 订阅。



#### 注意

有关将主机注册到红帽的改进和简化的体验，请使用远程主机配置(RHC)。RHC 客户端将您的系统注册到 Red Hat 使您的系统准备好进行 Insights 数据收集，并从 Insights 启用对 Red Hat Enterprise Linux 的直接问题修复。如需更多信息，请参阅 [RHC 注册](#)。

#### 先决条件

- 您有一个有效的、非试用的 Red Hat Enterprise Linux 订阅。
- 验证您的红帽订阅状态。
- 您之前还没有收到 Red Hat Enterprise Linux 8 订阅。

- 您已成功安装了 Red Hat Enterprise Linux 8，并以 root 用户身份登录到该系统。

## 流程

1. 以 root 用户身份打开终端窗口。
2. 使用激活码注册 Red Hat Enterprise Linux 系统：

```
# subscription-manager register --activationkey=<activation_key_name> --  
org=<organization_ID>
```

当成功注册系统时，会显示类似如下的输出：

```
The system has been registered with id:  
62edc0f8-855b-4184-b1b8-72a9dc793b96
```

## 其他资源

- [使用激活码使用红帽订阅管理器注册系统](#)
- [RHEL 系统注册入门](#)

## 第 16 章 使用 SUBSCRIPTION-MANAGER 命令行工具配置系统目的

系统目的是 Red Hat Enterprise Linux 安装的一个功能，它可以帮助 RHEL 客户获得红帽混合云控制台中提供的订阅体验和服务的好处，该控制台是一个基于仪表盘的软件即服务(SaaS)应用程序，可让您在您的红帽帐户中查看订阅的使用情况。

您可以在激活码上或使用订阅管理器工具配置系统目的属性。虽然建议在激活码上配置系统目的，但您也可以在安装后使用 **subscription-manager syspurpose** 命令行工具来设置所需的属性来配置它。

### 先决条件

- 已安装并注册了 Red Hat Enterprise Linux 8 系统，但没有配置系统用途。
- 以 **root** 用户身份登录。



### 注意

在授权模式下，如果您的系统已注册，但没有满足所需目的的订阅，您可以运行 **subscription-manager remove --all** 命令来删除附加的订阅。然后，您可以使用命令行 **subscription-manager syspurpose {role, use, service-level}** 工具来设置所需的目的属性，最后运行 **subscription-manager attach --auto** 在考虑更新的属性的情况下重新赋予系统权限。在启用了 SCA 的帐户中，您可以在注册后直接更新系统目的详情，而无需对系统中的订阅进行更新。

### 流程

1. 在终端窗口中运行以下命令设定系统预期的角色：

```
# subscription-manager syspurpose role --set "VALUE"
```

用您要分配的角色替换 **VALUE**:

- **Red Hat Enterprise Linux Server**
- **Red Hat Enterprise Linux Workstation**
- **Red Hat Enterprise Linux Compute 节点**

例如：

```
# subscription-manager syspurpose role --set "Red Hat Enterprise Linux Server"
```

- a. 可选：在设置值前，请查看您机构的订阅所支持的可用角色：

```
# subscription-manager syspurpose role --list
```

- b. 可选：运行以下命令以取消设置角色：

```
# subscription-manager syspurpose role --unset
```

2. 运行以下命令来设定系统的预期服务水平协议(SLA)：

```
# subscription-manager syspurpose service-level --set "VALUE"
```

使用您要分配的 SLA 替换 **VALUE**:

- **Premium** (高级)
- **Standard** (标准)
- **Self-Support** (自助)

例如 :

```
# subscription-manager syspurpose service-level --set "Standard"
```

- a. 可选 : 在设置值前, 请查看您机构的订阅所支持的可用的服务级别 :

```
# subscription-manager syspurpose service-level --list
```

- b. 可选 : 运行以下命令以取消设置 SLA:

```
# subscription-manager syspurpose service-level --unset
```

3. 运行以下命令设定系统预定用法 :

```
# subscription-manager syspurpose usage --set "VALUE"
```

使用您要分配的用途来替换 **VALUE** :

- **生产环境**
- **灾难恢复**
- **开发/测试**

例如 :

```
# subscription-manager syspurpose usage --set "Production"
```

- a. 可选 : 在设置值前, 请查看您机构的订阅所支持的可用用法 :

```
# subscription-manager syspurpose usage --list
```

- b. 可选 : 运行以下命令以取消设置用法 :

```
# subscription-manager syspurpose usage --unset
```

4. 运行以下命令来显示当前系统目的属性 :

```
# subscription-manager syspurpose --show
```

- a. 可选 : 要获得更详细的语法信息, 请运行以下命令访问 **subscription-manager** 手册页, 并浏览 SYSPURPOSE OPTIONS:

```
# man subscription-manager
```

## 验证

- 要在使用启用了授权模式的帐户注册的系统验证系统的订阅状态：

```
# subscription-manager status
+-----+
System Status Details
+-----+
Overall Status: Current

System Purpose Status: Matched
```

- 整体状态 **Current** 表示附加的订阅和授权涵盖所有安装的产品，以访问其内容集的存储库。
  - 系统用途状态**匹配**意味着附加的订阅满足系统上设置的所有系统用途属性（角色、使用量、服务水平）。
  - 当状态信息不理想时，会显示附加信息来帮助系统管理员决定对附加订阅进行何种更正，以覆盖安装的产品和预期系统目的。
- 要在使用启用了 SCA 模式的帐户注册的系统验证系统的订阅状态：

```
# subscription-manager status
+-----+
System Status Details
+-----+
Overall Status: Disabled
Content Access Mode is set to Simple Content Access. This host has access to content,
regardless of subscription status.
System Purpose Status: Disabled
```

- 在 SCA 模式下，订阅不再需要附加到各个系统。因此，总体状态和系统目的状态都显示为 Disabled。但是，系统目的属性提供的技术、业务和操作用例对于订阅服务非常重要。如果没有这些属性，订阅服务数据就不太准确。

## 其他资源

- 要了解有关订阅服务的更多信息，请参阅 [订阅服务入门指南](#)。

## 第 17 章 在 64 位 IBM Z 中配置 LINUX 实例

这部分论述了在 64 位 IBM Z 上安装 Red Hat Enterprise Linux 的大多数常见任务。

### 17.1. 添加 DASD

直接访问存储设备(DASD)是 64 位 IBM Z 常用的存储类型。如需更多信息，请参阅 IBM 知识库中的 [使用 DASD](#)。以下示例是关于如何在线设置 DASD，对其进行格式化，并使更改持久的。

如果在 z/VM 中运行，请确认设备已经被附加或者连接到 Linux 系统。

```
CP ATTACH EB1C TO *
```

要连接一个您可以访问的最小磁盘，请运行以下命令：

```
CP LINK RHEL7X 4B2E 4B2E MR
DASD 4B2E LINKED R/W
```

### 17.2. 在线动态设定 DASD

本节包含有关在线设置 DASD 的信息。

#### 流程

1. 使用 **cio\_ignore** 程序从忽略的设备列表中删除 DASD，并使其在 Linux 中可见：

```
# cio_ignore -r device_number
```

使用 DASD 的设备号替换 *device\_number*。例如：

```
# cio_ignore -r 4b2e
```

2. 设置设备在线。使用以下命令格式：

```
# chccwdev -e device_number
```

使用 DASD 的设备号替换 *device\_number*。例如：

```
# chccwdev -e 4b2e
```

另外，也可以使用 **sysfs** 属性在线设定该设备：

- a. 使用 **cd** 命令将 **/sys/** 的目录改为代表那个卷的目录：

```
# cd /sys/bus/ccw/drivers/dasd-eckd/0.0.4b2e/
# ls -l
total 0
-r--r--r-- 1 root root 4096 Aug 25 17:04 availability
-rw-r--r-- 1 root root 4096 Aug 25 17:04 cmb_enable
-r--r--r-- 1 root root 4096 Aug 25 17:04 cutype
-rw-r--r-- 1 root root 4096 Aug 25 17:04 detach_state
-r--r--r-- 1 root root 4096 Aug 25 17:04 devtype
```

```
-r--r--r-- 1 root root 4096 Aug 25 17:04 discipline
-rw-r--r-- 1 root root 4096 Aug 25 17:04 online
-rw-r--r-- 1 root root 4096 Aug 25 17:04 readonly
-rw-r--r-- 1 root root 4096 Aug 25 17:04 use_diag
```

- b. 检查这个设备是否已经在线：

```
# cat online
0
```

- c. 如果不在线，请输入以下命令使它在线：

```
# echo 1 > online
# cat online
1
```

3. 请确认哪个块正在被访问：

```
# ls -l
total 0
-r--r--r-- 1 root root 4096 Aug 25 17:04 availability
lrwxrwxrwx 1 root root  0 Aug 25 17:07 block -> ../../../../block/dasdb
-rw-r--r-- 1 root root 4096 Aug 25 17:04 cmb_enable
-r--r--r-- 1 root root 4096 Aug 25 17:04 cutype
-rw-r--r-- 1 root root 4096 Aug 25 17:04 detach_state
-r--r--r-- 1 root root 4096 Aug 25 17:04 devtype
-r--r--r-- 1 root root 4096 Aug 25 17:04 discipline
-rw-r--r-- 1 root root  0 Aug 25 17:04 online
-rw-r--r-- 1 root root 4096 Aug 25 17:04 readonly
-rw-r--r-- 1 root root 4096 Aug 25 17:04 use_diag
```

如示例所示，`/dev/dasdb` 正在访问设备 4B2E。

这些指令为当前会话在线设置 DASD，但在重启后不会保留。

有关如何永久在线设置 DASD 的详情，请参阅 [持久在线设置 DASD](#)。当使用 DASD 时，请使用 `/dev/disk/by-path/` 中的持久设备符号链接。

## 17.3. 准备使用低级格式化的新 DASD

磁盘在线后，返回 `/root` 目录并低级格式化该设备。这在 DASD 的整个生命周期中只需要一次：

```
# cd /root
# dasdfmt -b 4096 -d cdl -p /dev/disk/by-path/ccw-0.0.4b2e
Drive Geometry: 10017 Cylinders * 15 Heads = 150255 Tracks
```

I am going to format the device `/dev/disk/by-path/ccw-0.0.4b2e` in the following way:

```
Device number of device : 0x4b2e
Labelling device       : yes
Disk label             : VOL1
Disk identifier        : 0X4B2E
Extent start (trk no) : 0
Extent end (trk no)   : 150254
Compatible Disk Layout : yes
```

```

Blocksize          : 4096

--->> ATTENTION! <<---
All data of that device will be lost.
Type "yes" to continue, no will leave the disk untouched: yes
cyl  97 of 3338 |#-----| 2%

```

当进度条达到结尾格式化完成时，**dasdfmt** 会打印以下输出：

```

Rereading the partition table...
Exiting...

```

现在，使用 **fdasd** 对 DASD 进行分区。您最多可在 DASD 中创建三个分区。在我们的示例中，我们创建一个覆盖整个磁盘的分区：

```

# fdasd -a /dev/disk/by-path/ccw-0.0.4b2e
reading volume label ..: VOL1
reading vtoc .....: ok

auto-creating one partition for the whole disk...
writing volume label...
writing VTOC...
rereading partition table...

```

DASD 在线后（低级格式化后），它可以和 Linux 中的其它磁盘一样使用。例如：您可以在其分区（例如 **/dev/disk/by-path/ccw-0.0.4b2e-part1**）上创建文件系统、LVM 物理卷或交换空间。除了 **dasdfmt** 和 **fdasd** 命令，不要使用完整的 DASD 设备（**dev/dasdb**）。如果您想要使用整个 DASD，创建一个覆盖整个驱动器的分区，如上例中的 **fdasd**。

要在以后添加附加磁盘而不破坏现有磁盘条目，例如 **/etc/fstab**，请使用 **/dev/disk/by-path/** 中的永久设备符号链接。

## 17.4. 在线永久设定 DASD

以上说明描述了如何在运行的系统中动态激活 DASD。但是这种更改不具有持久性，重启后无法保留。在您的 Linux 系统中使对 DASD 配置的修改具有持久性取决于 DASD 是否属于 root 文件系统。在启动过程中需要 **initramfs** 非常早地激活 root 文件系统所需的 DASD 才能挂载根文件系统。

对于持久性设备配置，**cio\_ignore** 命令会被透明处理，您不需要从忽略列表中手动释放设备。

## 17.5. DASD 是 ROOT 文件系统一部分

您必须进行修改来添加 DASD 的文件是 Red Hat Enterprise Linux 8 中已更改的文件系统的一部分。运行以下命令可以找到要编辑的新文件，而不编辑 **/etc/zipl.conf** 文件：

```

# machine_id=$(cat /etc/machine-id)
# kernel_version=$(uname -r)
# ls /boot/loader/entries/$machine_id-$kernel_version.conf

```

有一个引导选项可在引导过程早期激活 DASD: **rd.dasd=**。这个选项使用直接访问存储设备(DASD)适配器设备总线标识符。如果需要多个 DASD，可以多次指定参数，或使用逗号分开的总线 ID 列表。要指定一个 DASD 范围，指定第一个和最后一个总线 ID。以下是一个系统的

`/boot/loader/entries/4ab74e52867b4f998e73e06cf23fd761-4.18.0-80.el8.s390x.conf` 文件示例。它使用两个 DASD 分区中的物理卷用于 LVM 卷组 `vg_devel1`，其中包括一个逻辑卷 `lv_root` 用于 root 文件系统。

```
title Red Hat Enterprise Linux (4.18.0-80.el8.s390x) 8.0 (Ootpa)
version 4.18.0-80.el8.s390x
linux /boot/vmlinuz-4.18.0-80.el8.s390x
initrd /boot/initramfs-4.18.0-80.el8.s390x.img
options root=/dev/mapper/vg_devel1-lv_root crashkernel=auto rd.dasd=0.0.0200 rd.dasd=0.0.0207
rd.lvm.lv=vg_devel1/lv_root rd.lvm.lv=vg_devel1/lv_swap cio_ignore=all,!condev
rd.znet=qeth,0.0.0a00,0.0.0a01,0.0.0a02,layer2=1,portno=0
id rhel-20181027190514-4.18.0-80.el8.s390x
grub_users $grub_users
grub_arg --unrestricted
grub_class kernel
```

在带有设备总线 ID 为 `0.0.202b` 的第三个 DASD 分区上添加另一个物理卷。为此，请将 `rd.dasd=0.0.202b` 添加到 `/boot/loader/entries/4ab74e52867b4f998e73e06cf23fd761-4.18.0-32.el8.s390x.conf` 中引导内核的参数行：

```
title Red Hat Enterprise Linux (4.18.0-80.el8.s390x) 8.0 (Ootpa)
version 4.18.0-80.el8.s390x
linux /boot/vmlinuz-4.18.0-80.el8.s390x
initrd /boot/initramfs-4.18.0-80.el8.s390x.img
options root=/dev/mapper/vg_devel1-lv_root crashkernel=auto rd.dasd=0.0.0200 rd.dasd=0.0.0207
rd.dasd=0.0.202b rd.lvm.lv=vg_devel1/lv_root rd.lvm.lv=vg_devel1/lv_swap cio_ignore=all,!condev
rd.znet=qeth,0.0.0a00,0.0.0a01,0.0.0a02,layer2=1,portno=0
id rhel-20181027190514-4.18.0-80.el8.s390x
grub_users $grub_users
grub_arg --unrestricted
grub_class kernel
```



### 警告

请确定配置文件中的内核命令行长度不超过 896 字节。否则引导装载程序无法被保存，安装将失败。

运行 `zipl` 来对下一个 IPL 应用配置文件的更改：

```
# zipl -V
Using config file '/etc/zipl.conf'
Using BLS config file '/boot/loader/entries/4ab74e52867b4f998e73e06cf23fd761-4.18.0-80.el8.s390x.conf'
Target device information
Device.....: 5e:00
Partition.....: 5e:01
Device name.....: dasda
Device driver name.....: dasd
DASD device number.....: 0201
```

```

Type.....: disk partition
Disk layout.....: ECKD/compatible disk layout
Geometry - heads.....: 15
Geometry - sectors.....: 12
Geometry - cylinders.....: 13356
Geometry - start.....: 24
File system block size.....: 4096
Physical block size.....: 4096
Device size in physical blocks...: 262152
Building bootmap in '/boot'
Building menu 'zipl-automatic-menu'
Adding #1: IPL section '4.18.0-80.el8.s390x' (default)
  initial ramdisk...: /boot/initramfs-4.18.0-80.el8.s390x.img
  kernel image.....: /boot/vmlinuz-4.18.0-80.el8.s390x
  kernel parmline...: 'root=/dev/mapper/vg_devel1-lv_root crashkernel=auto rd.dasd=0.0.0200
rd.dasd=0.0.0207 rd.dasd=0.0.020b rd.lvm.lv=vg_devel1/lv_root rd.lvm.lv=vg_devel1/lv_swap
cio_ignore=all,!condev rd.znet=qeth,0.0.0a00,0.0.0a01,0.0.0a02,layer2=1,portno=0'
  component address:
    kernel image....: 0x00010000-0x0049afff
    parmline.....: 0x0049b000-0x0049bfff
    initial ramdisk.: 0x004a0000-0x01a26fff
    internal loader.: 0x0000a000-0x0000cfff
Preparing boot menu
  Interactive prompt.....: enabled
  Menu timeout.....: 5 seconds
  Default configuration...: '4.18.0-80.el8.s390x'
Preparing boot device: dasda (0201).
Syncing disks...
Done.

```

## 17.6. DASD 不是 ROOT 文件系统的一部分

不是 root 文件系统（即 *数据磁盘*）一部分的直接访问存储设备(DASD)会在 `/etc/dasd.conf` 文件中永久配置。这个文件每行包含一个 DASD，其中每行都以 DASD 的总线 ID 开头。

在向 `/etc/dasd.conf` 文件添加 DASD 时，请使用键值对来指定每个条目的选项。使用等号(=)分隔键及其值。添加多个选项时，请使用空格或 tab 来分隔各个选项。

### `/etc/dasd.conf` 文件示例

```

0.0.0207
0.0.0200 use_diag=1 readonly=1

```

对 `/etc/dasd.conf` 文件的更改在系统重启后或通过更改系统的 I/O 配置（即，DASD 被连接在 z/VM 下）来动态添加新 DASD 后生效。

另外，要激活添加到 `/etc/dasd.conf` 文件中的 DASD，请完成以下步骤：

1. 从忽略的设备列表中删除 DASD，并使用 `cio_ignore` 工具使其可见：

```
# cio_ignore -r device_number
```

其中 *device\_number* 是 DASD 设备号。

例如，如果设备号是 **021a**，请运行：

```
# cio_ignore -r 021a
```

- 通过写入设备的 **uevent** 属性来激活 DASD :

```
# echo add > /sys/bus/ccw/devices/dasd-bus-ID/uevent
```

其中 ***dasd-bus-ID*** 是 DASD 的总线 ID。

例如，如果总线 ID 是 **0.0.021a**，请运行：

```
# echo add > /sys/bus/ccw/devices/0.0.021a/uevent
```

## 17.7. FCP LUN 是 ROOT 文件系统的一部分

您为添加 FCP LUN 所必须修改的唯一文件是 Red Hat Enterprise Linux 8 中已更改的 root 文件系统的一部分。运行以下命令可以找到要编辑的新文件，而不编辑 **/etc/zipl.conf** 文件：

```
# machine_id=$(cat /etc/machine-id)
# kernel_version=$(uname -r)
# ls /boot/loader/entries/$machine_id-$kernel_version.conf
```

Red Hat Enterprise Linux 提供在引导过程早期激活 FCP LUN 的参数：**rd.zfcp=**。该值是一个包含 FCP 设备总线 ID 的用逗号分隔的列表，目标 WWPN 为前缀为 0x 的 16 位十六进制数，以及前缀为 **0x**，右边填充为 0，有 16 位十六进制数的 FCP LUN。

当自动 LUN 扫描没有被 **zfcp.allow\_lun\_scan=0** 内核模块参数禁用时或当安装 RHEL-8.6 或旧版本时，如果没有在 NPIV 模式下配置 **zFCP** 设备，WWPN 和 FCP LUN 值才是必需的。否则，可以省略它们，例如 **rd.zfcp=0.0.4000**。以下是一个系统的 **/boot/loader/entries/4ab74e52867b4f998e73e06cf23fd761-4.18.0-80.el8.s390x.conf** 文件示例。它使用两个 FCP LUN 分区中的物理卷用于一个 LVM 卷组 **vg\_devel1**，其中包括一个逻辑卷 **lv\_root** 用于 root 文件系统。为方便起见，该示例显示没有多路径的配置。

```
title Red Hat Enterprise Linux (4.18.0-32.el8.s390x) 8.0 (Ootpa)
version 4.18.0-32.el8.s390x
linux /boot/vmlinuz-4.18.0-32.el8.s390x
initrd /boot/initramfs-4.18.0-32.el8.s390x.img
options root=/dev/mapper/vg_devel1-lv_root crashkernel=auto
rd.zfcp=0.0.fc00,0x5105074308c212e9,0x401040a000000000
rd.zfcp=0.0.fc00,0x5105074308c212e9,0x401040a100000000 rd.lvm.lv=vg_devel1/lv_root
rd.lvm.lv=vg_devel1/lv_swap cio_ignore=all,!condev
rd.znet=qeth,0.0.0a00,0.0.0a01,0.0.0a02,layer2=1,portno=0
id rhel-20181027190514-4.18.0-32.el8.s390x
grub_users $grub_users
grub_arg --unrestricted
grub_class kernel
```

- 要在设备总线为 ID 0.0.fc00，WWPN 为 0x5105074308c212e9 和 FCP LUN 为 0x401040a300000000 的第三个 FCP LUN 分区上添加另一个物理卷，请将 **rd.zfcp=0.0.fc00,0x5105074308c212e9, 0x401040a300000000** 添加到 **/boot/loader/entries/4ab74e52867b4f998e73e06cf23fd761-4.18.0-32.el8.s390x.conf** 中引导内核的参数行。例如：

```
title Red Hat Enterprise Linux (4.18.0-32.el8.s390x) 8.0 (Ootpa)
```

```

version 4.18.0-32.el8.s390x
linux /boot/vmlinuz-4.18.0-32.el8.s390x
initrd /boot/initramfs-4.18.0-32.el8.s390x.img
options root=/dev/mapper/vg_devel1-lv_root crashkernel=auto
rd.zfcp=0.0.fc00,0x5105074308c212e9,0x401040a000000000
rd.zfcp=0.0.fc00,0x5105074308c212e9,0x401040a100000000
rd.zfcp=0.0.fc00,0x5105074308c212e9,0x401040a300000000 rd.lvm.lv=vg_devel1/lv_root
rd.lvm.lv=vg_devel1/lv_swap cio_ignore=all,!condev
rd.znet=qeth,0.0.0a00,0.0.0a01,0.0.0a02,layer2=1,portno=0
id rhel-20181027190514-4.18.0-32.el8.s390x
grub_users $grub_users
grub_arg --unrestricted
grub_class kernel

```



### 警告

请确定配置文件中的内核命令行长度不超过 896 字节。否则引导装载程序无法被保存，安装将失败。

- 运行 **dracut -f** 以更新目标内核的初始 RAM 磁盘。
- 运行 **zipl** 来对下一个 IPL 应用配置文件的更改：

```

# zipl -V
Using config file '/etc/zipl.conf'
Using BLS config file '/boot/loader/entries/4ab74e52867b4f998e73e06cf23fd761-4.18.0-32.el8.s390x.conf'
Target device information
Device.....: 08:00
Partition.....: 08:01
Device name.....: sda
Device driver name.....: sd
Type.....: disk partition
Disk layout.....: SCSI disk layout
Geometry - start.....: 2048
File system block size.....: 4096
Physical block size.....: 512
Device size in physical blocks..: 10074112
Building bootmap in '/boot/'
Building menu 'rh-automatic-menu'
Adding #1: IPL section '4.18.0-32.el8.s390x' (default)
kernel image.....: /boot/vmlinuz-4.18.0-32.el8.s390x
kernel parmline...: 'root=/dev/mapper/vg_devel1-lv_root crashkernel=auto
rd.zfcp=0.0.fc00,0x5105074308c212e9,0x401040a000000000
rd.zfcp=0.0.fc00,0x5105074308c212e9,0x401040a100000000
rd.zfcp=0.0.fc00,0x5105074308c212e9,0x401040a300000000 rd.lvm.lv=vg_devel1/lv_root
rd.lvm.lv=vg_devel1/lv_swap cio_ignore=all,!condev
rd.znet=qeth,0.0.0a00,0.0.0a01,0.0.0a02,layer2=1,portno=0'
initial ramdisk...: /boot/initramfs-4.18.0-32.el8.s390x.img
component address:

```

```
kernel image.....: 0x00010000-0x007a21ff
parmline.....: 0x00001000-0x000011ff
initial ramdisk.: 0x02000000-0x028f63ff
internal loader.: 0x0000a000-0x0000a3ff
Preparing boot device: sda.
Detected SCSI PCBIOS disk layout.
Writing SCSI master boot record.
Syncing disks...
Done.
```

## 17.8. FCP LUN 不是 ROOT 文件系统的一部分

不是 root 文件系统一部分的 FCP LUN，比如数据磁盘，会在 `/etc/zfcp.conf` 中永久配置。每行包含一个 FCP LUN。每行都包含 FCP 适配器的设备总线 ID、目标 WWPN 为前缀为 **0x** 的 16 位十六进制数，以及前缀为 **0x**，右边填充为 0，有 16 位十六进制数的 FCP LUN，由空格或标签页分开。

当 **auto LUN** 扫描被 `zfcp.allow_lun_scan=0` 内核模块参数禁用时或当安装 RHEL-8.6 或旧版本时，如果在 NPIV 下没有配置 **zFCP**，WWPN 和 FCP LUN 值才是必需的。否则，可以省略它们，并且只强制使用设备总线 ID。

在系统中添加 FCP 适配器时，`/etc/zfcp.conf` 中的条目会被激活并由 `udev` 配置。在引导时，会添加系统可见的所有 FCP 适配器并触发 `udev`。

`/etc/zfcp.conf` 内容示例：

```
0.0.fc00 0x5105074308c212e9 0x401040a000000000
0.0.fc00 0x5105074308c212e9 0x401040a100000000
0.0.fc00 0x5105074308c212e9 0x401040a300000000
0.0.fcd0 0x5105074308c2aee9 0x401040a000000000
0.0.fcd0 0x5105074308c2aee9 0x401040a100000000
0.0.fcd0 0x5105074308c2aee9 0x401040a300000000
0.0.4000
0.0.5000
```

只有重启系统后或通过更改系统的 I/O 配置（例如，z/VM 下连接的通道）来动态添加新 FCP 通道后，对 `/etc/zfcp.conf` 的修改才会生效。另外，可以执行以下命令为之前没有激活的 FCP 适配器激活 `/etc/zfcp.conf` 中的新条目：

1. 使用 `cio_ignore` 程序从忽略的设备列表中删除 FCP 适配器，并使其出现在 Linux 中：

```
# cio_ignore -r device_number
```

使用 FCP 适配器的设备号替换 `device_number`。例如：

```
# cio_ignore -r fcfc
```

2. 要触发激活更改的 `uevent`，请执行：

```
# echo add > /sys/bus/ccw/devices/device-bus-ID/uevent
```

例如：

```
# echo add > /sys/bus/ccw/devices/0.0.fcfc/uevent
```

## 17.9. 添加 QETH 设备

**qeth** 网络设备驱动程序支持 QDIO 模式、HiperSockets、z/VM 客户机 LAN 和 z/VM VSWITCH 中的 64 位 IBM Z OSA-Express 功能。

有关 **qeth** 设备驱动程序命名方案的更多信息，请参阅 [自定义引导参数](#)。

## 17.10. 动态添加 QETH 设备

本节包含如何动态添加 **qeth** 设备的信息。

### 流程

1. 决定是否载入 **qeth** 设备驱动程序模块。以下示例显示了载入的 **qeth** 模块：

```
# lsmod | grep qeth
qeth_l3          69632 0
qeth_l2          49152 1
qeth             131072 2 qeth_l3,qeth_l2
qdio             65536 3 qeth,qeth_l3,qeth_l2
ccwgroup        20480 1 qeth
```

如果 **lsmod** 命令的输出显示 **qeth** 模块还没有被加载，请运行 **modprobe** 命令去加载它们：

```
# modprobe qeth
```

2. 使用 **cio\_ignore** 程序从忽略的设备列表中删除网络通道，并使其出现在 Linux 中：

```
# cio_ignore -r read_device_bus_id,write_device_bus_id,data_device_bus_id
```

使用代表网络设备的三个设备总线 ID 替换 *read\_device\_bus\_id*、*write\_device\_bus\_id* 和 *data\_device\_bus\_id* 例如，如果 *read\_device\_bus\_id* 是 **0.0.f500**，*write\_device\_bus\_id* 是 **0.0.f501**，*data\_device\_bus\_id* 是 **0.0.f502**：

```
# cio_ignore -r 0.0.f500,0.0.f501,0.0.f502
```

3. 使用 **znetconf** 工具来检测并列出网络设备的候选配置：

```
# znetconf -u
Scanning for network devices...
Device IDs          Type   Card Type   CHPID Drv.
-----
0.0.f500,0.0.f501,0.0.f502 1731/01 OSA (QDIO)    00 qeth
0.0.f503,0.0.f504,0.0.f505 1731/01 OSA (QDIO)    01 qeth
0.0.0400,0.0.0401,0.0.0402 1731/05 HiperSockets 02 qeth
```

4. 选择您要使用的配置，并使用 **znetconf** 啦应用配置，并将配置的组设备上线来作为网络设备。

```
# znetconf -a f500
Scanning for network devices...
Successfully configured device 0.0.f500 (encf500)
```

5. 可选：您还可以在组设备设置在线前传递在组设备上配置的参数：

```
# znetconf -a f500 -o portname=myname
Scanning for network devices...
Successfully configured device 0.0.f500 (encf500)
```

现在可以继续配置 **encf500** 网络接口。

另外，您可以使用 **sysfs** 属性设定设备在线，如下：

1. 创建 **qeth** 组设备：

```
# echo read_device_bus_id,write_device_bus_id,data_device_bus_id >
/sys/bus/ccwgroup/drivers/qeth/group
```

例如：

```
# echo 0.0.f500,0.0.f501,0.0.f502 > /sys/bus/ccwgroup/drivers/qeth/group
```

2. 下一步，通过查找读取频道来验证 **qeth** 组设备是否已被正确创建：

```
# ls /sys/bus/ccwgroup/drivers/qeth/0.0.f500
```

根据您的系统设置情况以及您所需要的功能设置其他参数和特性，例如：

- **portno**
- **layer2**
- **portname**

3. 将在线 **sysfs** 属性写入 **1** 将设备设置为在线：

```
# echo 1 > /sys/bus/ccwgroup/drivers/qeth/0.0.f500/online
```

4. 然后确认该设备状态：

```
# cat /sys/bus/ccwgroup/drivers/qeth/0.0.f500/online
1
```

返回值为 **1** 表示设备在线，返回值 **0** 表示设备离线。

5. 查找分配给该设备的接口名称：

```
# cat /sys/bus/ccwgroup/drivers/qeth/0.0.f500/if_name
encf500
```

现在可以继续配置 **encf500** 网络接口。

**s390utils** 软件包提供的以下命令可以显示 **qeth** 设备的重要设置：

```
# lsqeth encf500
Device name           : encf500
-----
card_type             : OSD_1000
cdev0                 : 0.0.f500
```

```

cdev1          : 0.0.f501
cdev2          : 0.0.f502
chpid          : 76
online         : 1
portname       : OSAPORT
portno         : 0
state          : UP (LAN ONLINE)
priority_queueing : always queue 0
buffer_count   : 16
layer2         : 1
isolation      : none

```

## 17.11. 永久添加 QETH 设备

要让您的新 **qeth** 设备基于持久性，需要为您的新接口生成配置文件。网络接口配置文件位于 **/etc/sysconfig/network-scripts/** 目录下。

这个网络配置文件的命名规则为 **ifcfg-device**，其中 *device* 是之前创建的 **qeth** 组设备的 **if\_name** 文件中的值。例如：**enc9a0**。对于持久性设备配置，**cio\_ignore** 命令会被透明处理，您不需要从忽略列表中手动释放设备。

如果同一类型的另一个设备的配置文件已经存在，添加配置文件的最简单方法是将其复制到新名称中，然后编辑它：

```

# cd /etc/sysconfig/network-scripts
# cp ifcfg-enc9a0 ifcfg-enc600

```

要了解网络设备的 ID，使用 **lsqeth** 工具：

```

# lsqeth -p
devices          CHPID interface   cardtype   port checksum prio-q'ing rtr4 rtr6 lay'2 cnt
-----
0.0.09a0/0.0.09a1/0.0.09a2 x00  enc9a0  Virt.NIC QDIO 0  sw  always_q_2 n/a n/a 1  64
0.0.0600/0.0.0601/0.0.0602 x00  enc600  Virt.NIC QDIO 0  sw  always_q_2 n/a n/a 1  64

```

如果还没有定义类似的设备，必须生成一个新文件。使用 **/etc/sysconfig/network-scripts/ifcfg-0.0.09a0** 示例作为模板：

```

# IBM QETH
DEVICE=enc9a0
BOOTPROTO=static
IPADDR=10.12.20.136
NETMASK=255.255.255.0
ONBOOT=yes
NETTYPE=qeth
SUBCHANNELS=0.0.09a0,0.0.09a1,0.0.09a2
PORTNAME=OSAPORT
OPTIONS='layer2=1 portno=0'
MACADDR=02:00:00:23:65:1a
TYPE=Ethernet

```

编辑新的 **ifcfg-0.0.0600** 文件，如下：

1. 修改 **DEVICE** 声明以反映您的 **ccw** 组中 **if\_name** 文件的内容。

2. 修改 **IPADDR** 声明以反映您的新接口的 IP 地址。
3. 根据需要修改 **NETMASK** 声明。
4. 如果要在引导时激活新接口，请确定将 **ONBOOT** 设为 **yes**。
5. 请确定 **SUBCHANNELS** 的值和您的 qeth 设备的硬件地址一致。
6. 修改 **PORTNAME** 声明，如果您的环境不需要该参数，则无需修改。
7. 您可以将任何有效的 **sysfs** 属性及其值添加到 **OPTIONS** 参数。Red Hat Enterprise Linux 安装程序目前使用它来配置层模式(**layer2**)，以及 **qeth** 设备的相对端口号(**portno**)。现在 OSA 设备的默认 **qeth** 设备驱动程序为 2 层模式。要继续使用之前需要依赖 3 层模式的旧的 **ifcfg** 定义，在 **OPTIONS** 参数中添加 **layer2=0**。

#### **/etc/sysconfig/network-scripts/ifcfg-0.0.0600**

```
# IBM QETH
DEVICE=enc600
BOOTPROTO=static
IPADDR=192.168.70.87
NETMASK=255.255.255.0
ONBOOT=yes
NETTYPE=qeth
SUBCHANNELS=0.0.0600,0.0.0601,0.0.0602
PORTNAME=OSAPORT
OPTIONS='layer2=1 portno=0'
MACADDR=02:00:00:b3:84:ef
TYPE=Ethernet
```

只有重启系统后或通过更改系统的 I/O 配置（例如在 z/VM 下连接）来动态添加新的网络设备后，对 **ifcfg** 文件的更改才会生效。另外，您可以执行以下命令来为之前未激活的网络通道触发 **ifcfg** 文件的激活：

1. 使用 **cio\_ignore** 程序从忽略的设备列表中删除网络通道，并使其出现在 Linux 中：

```
# cio_ignore -r read_device_bus_id,write_device_bus_id,data_device_bus_id
```

使用代表网络设备的三个设备总线 ID 替换 *read\_device\_bus\_id*、*write\_device\_bus\_id* 和 *data\_device\_bus\_id* 例如，如果 *read\_device\_bus\_id* 是 **0.0.0600**，则 *write\_device\_bus\_id* 为 **0.0.0601**，*data\_device\_bus\_id* 是 **0.0.0602**

```
# cio_ignore -r 0.0.0600,0.0.0601,0.0.0602
```

2. 要触发激活更改的 uevent，请执行：

```
# echo add > /sys/bus/ccw/devices/read-channel/uevent
```

例如：

```
# echo add > /sys/bus/ccw/devices/0.0.0600/uevent
```

3. 检查网络设备状态：

```
# lsqeth
```

4. 现在启动新的接口：

```
# ifup enc600
```

5. 检查接口的状态：

```
# ip addr show enc600
3: enc600: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP group default qlen 1000
link/ether 3c:97:0e:51:38:17 brd ff:ff:ff:ff:ff:ff
inet 10.85.1.245/24 brd 10.34.3.255 scope global dynamic enc600
valid_lft 81487sec preferred_lft 81487sec
inet6 1574:12:5:1185:3e97:eff:fe51:3817/64 scope global noprefixroute dynamic
valid_lft 2591994sec preferred_lft 604794sec
inet6 fe45::a455:eff:d078:3847/64 scope link
valid_lft forever preferred_lft forever
```

6. 检查新接口的路由：

```
# ip route
default via 10.85.1.245 dev enc600 proto static metric 1024
12.34.4.95/24 dev enp0s25 proto kernel scope link src 12.34.4.201
12.38.4.128 via 12.38.19.254 dev enp0s25 proto dhcp metric 1
192.168.122.0/24 dev virbr0 proto kernel scope link src 192.168.122.1
```

7. 使用 **ping** 程序 ping 网关或者新设备子网中的另一台主机确认您的更改：

```
# ping -c 1 192.168.70.8
PING 192.168.70.8 (192.168.70.8) 56(84) bytes of data.
64 bytes from 192.168.70.8: icmp_seq=0 ttl=63 time=8.07 ms
```

8. 如果默认路由信息被改变了，需要更新相应的 **/etc/sysconfig/network**。

## 其它资源

- **nm-settings-keyfile** 手册页

## 17.12. 为网络 ROOT 文件系统配置 64 位 IBM Z 网络设备

要添加访问 root 文件系统所需的网络设备，您只需要修改引导选项。引导选项可在参数文件中，但 **/etc/zipl.conf** 文件不再包含引导记录说明。使用以下命令可以定位需要修改的文件：

```
# machine_id=$(cat /etc/machine-id)
# kernel_version=$(uname -r)
# ls /boot/loader/entries/$machine_id-$kernel_version.conf
```

**Dracut** 是提供 **initramfs** 中用于替换 **initrd** 的功能的 **mkinitrd** 成功程序，提供了一个引导参数以在引导过程早期在 64 位 IBM Z 中激活网络设备：**rd.znet=**。

作为输入，此参数采用以逗号分隔的 **NETTYPE** (qeth、lcs、ctc)、两个 (lcs、ctc) 或三个 (qeth) 设备总线 ID 的列表，以及由网络设备 **sysfs** 属性对应的键值对组成的可选的其它参数。这个参数配置和激活 64 位 IBM Z 网络硬件。对 IP 地址和其他具体网络的配置与其他平台一样。详情请查看 **dracut** 文档。

在 boot 中明确处理网络通道的 `cio_ignore` 命令。

通过 NFS 通过网络访问的 root 文件系统引导选项示例：

```
root=10.16.105.196:/nfs/nfs_root cio_ignore=all,!condev  
rd.znet=qeth,0.0.0a00,0.0.0a01,0.0.0a02,layer2=1,portno=0,portname=OSAPORT  
ip=10.16.105.197:10.16.105.196:10.16.111.254:255.255.248.0:nfs-server.subdomain.domain:enc9a0:n  
one rd_NO_LUKS rd_NO_LVM rd_NO_MD rd_NO_DM LANG=en_US.UTF-8  
SYSFONT=latacyrheb-sun16 KEYTABLE=us
```

## 第 18 章 保护您的系统

在完成安装过程后，您必须保护您的 Red Hat Enterprise Linux 系统。

### 先决条件

- 您已完成图形安装。

### 流程

1. 要更新您的系统，请作为 root 运行以下命令：

```
# yum update
```

2. 即使在安装 Red Hat Enterprise Linux 时默认自动启用了防火墙服务( **firewalld** )，但在一些情况下可能会明确禁用它，例如在 Kickstart 配置中。在那种情况下，建议您重新启用防火墙。要启动 **firewalld**，请作为 root 运行以下命令：

```
# systemctl start firewalld  
# systemctl enable firewalld
```

3. 要提高安全性，禁用您不需要的服务。例如，如果您的系统没有安装打印机，使用以下命令禁用 cups 服务：

```
# systemctl mask cups
```

要查看活跃的服务，运行以下命令：

```
$ systemctl list-units | grep service
```

## 第 19 章 更改订阅服务

要管理订阅，您可以在 Red Hat Subscription Management Server 或者 Red Hat Satellite Server 中注册 RHEL 系统。如果需要，可以稍后更改订阅服务。要更改您注册的订阅服务，请从当前服务中取消注册该系统，然后使用新服务进行注册。

要接收系统更新，请向任一管理服务器注册您的系统。

本节介绍了如何从 Red Hat Subscription Management Server 和 Red Hat Satellite Server 中取消注册 RHEL 系统。

### 先决条件

使用以下方法注册了您的系统：它包括以下之一：

- Red Hat Subscription Management Server
- Red Hat Satellite Server version 6.11

要接收系统更新，请向任一管理服务器注册您的系统。

### 19.1. 从 SUBSCRIPTION MANAGEMENT SERVER 中取消注册。

本节介绍如何使用命令行和 Subscription Manager 用户界面从 Red Hat Subscription Management Server 取消注册 RHEL 系统。

#### 19.1.1. 使用命令行取消注册

使用 **unregister** 命令从 Red Hat Subscription Management Server 取消注册 RHEL 系统。

#### 流程

1. 作为 root 用户运行 unregister 命令，无需任何附加参数。

```
# subscription-manager unregister
```

2. 提示时请提供 root 密码。

在 Subscription Management Server 中取消注册该系统，状态会显示“系统当前没有注册”信息，且 **Register** 按钮可用。

要继续不间断的服务，请使用任一管理服务重新注册系统。如果您没有使用管理服务注册系统，您可能无法收到系统更新。有关注册系统的更多信息，请参阅 [使用命令行注册您的系统](#)。

#### 其他资源

- [使用和配置红帽订阅管理器](#)

#### 19.1.2. 使用 Subscription Manager 用户界面取消注册

本节介绍如何使用 Subscription Manager 用户界面从 Red Hat Subscription Management Server 中取消注册 RHEL 系统。

#### 流程

1. 登录到您的系统。
2. 从窗口的左上方点击 **Activities**。
3. 在菜单选项中点显示**应用程序**图标。
4. 点 **Red Hat Subscription Manager** 图标，或使用 **Red Hat Subscription Manager** 进行搜索。
5. 在 **Authentication Required** 对话框中输入管理员密码。**Subscriptions** 窗口会出现并显示订阅、系统目的和已安装产品的当前状态。未注册的产品会显示红色 X。需要进行身份验证才能在系统上执行特权任务。
6. 点 **Unregister** 按钮。

在 Subscription Management Server 中取消注册该系统，状态会显示"系统当前没有注册"信息，且**Register** 按钮可用。

要继续不间断的服务，请使用任一管理服务重新注册系统。如果您没有使用管理服务注册系统，您可能无法收到系统更新。有关注册系统的更多信息，请参阅 [使用订阅管理器用户界面注册您的系统](#)。

#### 其它资源

- [使用和配置红帽订阅管理器](#)

## 19.2. 在 SATELLITE 服务器中取消注册

要从 Satellite 服务器中取消 Red Hat Enterprise Linux 系统注册，从 Satellite 服务器中删除该系统。

如需更多信息，请参阅[从 Red Hat Satellite 中删除主机](#)。

## 部分 IV. 附录

## 附录 A. KICKSTART 脚本文件格式参考

这个参考详细描述了 kickstart 文件格式。

### A.1. KICKSTART 文件格式

Kickstart 脚本是包含安装程序识别的关键字的纯文本文件，它们是安装指令。任何可将文件保存为 ASCII 文本的文本编辑，比如 **Gedit** 或 **vim**（Linux 系统），或 **Notepad**（Windows 系统），都可以用来创建和编辑 Kickstart 文件。Kickstart 配置的文件名无关紧要，但建议使用简单名称，因为您需要在其他配置文件或对话框中指定这个名称。

#### 命令

命令是作为安装指令的关键字。每个命令都必须位于一行。命令可以选择。指定命令和选项和在 shell 中使用 Linux 命令类似。

#### 部分

某些以 % 字符开头的特殊命令可启动某个部分。部分中的命令解释与其它部分的命令不同。每个部分都必须使用 **%end** 命令结束。

#### 部分类型

可用的部分有：

- **Add-on 部分**。这些部分使用 **%addon *addon\_name*** 命令。
- **软件包选择部分**。以 **%packages** 开始。使用它列出安装的软件包，包括间接方法，比如软件包组或者模块。
- **Script 部分**。它们以 **%pre**、**%pre-install**、**%post** 和 **%onerror** 开头。这些部分不是必需的。

#### command 部分

command 部分是那些不属于任何 script 部分或 **%packages** 部分的 Kickstart 文件中的命令的术语。

#### 脚本部分计数和排序

除 command 部分外的所有部分都是可选的，并可以多次出现。当要评估特定类型的脚本部分时，Kickstart 中存在的所有部分都会按照外观顺序进行评估：两个 **%post** 部分会按照出现的顺序进行评估。但是，您不必按任何顺序指定各种脚本部分类型：**%pre** 部分前是否有 **%post** 部分无关紧要。

#### 注释

Kickstart 注释是以 hash # 字符开头的行。安装程序会忽略这些行。

不必需的项目可以被省略。省略安装程序中所有需要的项目，这样用户就可以象常规互动安装一样，提供对相关项目的回答。也可以将 kickstart 脚本声明为非交互式，通过 **cmdline** 命令。在非互动模式中，任何缺少的回答都会中止安装过程。



#### 注意

如果在文本或图形模式下进行 kickstart 安装过程中需要用户交互，则仅进入要强制更新的窗口以完成安装。输入 spoke 可能会导致重置 kickstart 配置。在进入 Installation Destination 窗口后，重置配置专门应用于与存储相关的 kickstart 命令。

### A.2. KICKSTART 中的软件包选择

Kickstart 使用 **%packages** 命令启动的部分来选择要安装的软件包。您可以以这种方式安装软件包、组、环境、模块流和模块配置集。

### A.2.1. 软件包选择部分

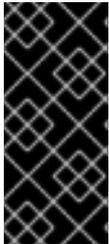
使用 **%packages** 命令启动一个 Kickstart 部分，该部分描述了要安装的软件包。**%packages** 部分必须以 **%end** 命令结尾。

您可以根据环境、组、模块流、模块配置集或者它们的软件包名称指定软件包。定义了包含相关软件包的几个环境和组。请参考有关环境和组列表的 Red Hat Enterprise Linux 8 安装 DVD 上的 **repository/repodata/\*-comps-repository.architecture.xml** 文件。

**\*-comps-repository.architecture.xml** 文件包含描述可用环境（使用 **<environment>** 标签标记）和组（**<group>** 标签）的结构。每个条目都有一个 ID、用户可见值、名称、描述和软件包列表。如果为安装选择了组，则在软件包列表中标记了 **mandatory** 的软件包总会被安装；如果其它位置没有特别排除，标记了 **default** 的软件包也会被安装，标记为 **optional** 的软件包需要在其他地方被指定包括时才会安装，即使已经选择该组也是如此。

您可以使用其 ID（标签）或名称（**<id>** 标签）指定软件包组或环境 **<name>**。

如果您不确定应该安装什么软件包，红帽建议您选择 **Minimal Install** 环境。**Minimal Install** 只提供运行 Red Hat Enterprise Linux 8 所必需的软件包。这将显著降低系统受某个漏洞影响的机会。如果需要，可以在安装后再添加附加软件包。有关 **Minimal Install** 的详情，请参阅 [安全强化文档中的安装最小软件包挂载部分](#)。**初始设置** 无法在从 Kickstart 文件安装系统后运行，除非安装中包含桌面环境和 X Window 系统，并且启用了图形登录。



#### 重要

要在 64 位系统中安装 32 位软件包：

- 为 **%packages** 部分指定 **--multilib** 选项
- 使用构建该软件包的 32 位架构附加软件包名称；例如：**glibc.i686**

### A.2.2. 软件包选择命令

这些命令可以在 Kickstart 文件的 **%packages** 部分中使用。

#### 指定一个环境

以 **@^** 符号开头的行形式指定要安装的整个环境：

```
%packages
@^Infrastructure Server
%end
```

这将安装属于 **Infrastructure Server** 环境一部分的所有软件包。所有可用环境都在 Red Hat Enterprise Linux 8 安装 DVD 上的 **repository/repodata/\*-comps-repository.architecture.xml** 文件中进行了描述。

在 Kickstart 文件中只能指定一个环境。如果指定了多个环境，则只使用最后指定的环境。

#### 指定组

指定组，每个条目一行，以 **@** 符号开头，然后是 **\*-comps-repository.architecture.xml** 文件中给出的完整组群名称或者组群 ID。例如：

■

```
%packages
@X Window System
@Desktop
@Sound and Video
%end
```

**Core** 组总是被选择 - 不需要在 **%packages** 部分指定它。

### 指定单独的软件包

根据名称指定单个软件包，每个条目对应一行。您可以在软件包名称中使用星号字符 (\*) 作为通配符。例如：

```
%packages
sqlite
curl
aspell
docbook*
%end
```

**docbook\*** 条目包含软件包 **docbook-dtds** 和 **docbook-style**，它们与通配符表示的模式匹配。

### 指定模块流的配置集

使用配置集语法为模块流指定配置集（一个条目为一行）：

```
%packages
@module:stream/profile
%end
```

这会安装模块流指定配置集中列出的所有软件包。

- 当模块指定了默认流时，您可以将其退出。如果没有指定默认流，您必须指定它。
- 当模块流指定默认配置集时，您可以将其退出。如果没有指定默认配置集，您必须指定它。
- 无法多次使用不同流安装模块。
- 有可能安装同一模块和流的多个配置集。

模块和组使用以 @ 符号开头的相同语法。当某个模块和软件包组具有相同名称时，该模块会优先使用。

在 Red Hat Enterprise Linux 8 中，模块只存在于 AppStream 存储库中。要列出提供的模块，请在安装的 Red Hat Enterprise Linux 8 系统上使用 **yum module list** 命令。

也可以使用模块 **Kickstart** 命令启用模块流，然后通过直接命名模块流中包含的软件包来安装它们。

### 排除环境、组群或者软件包

使用前导短划线 (-) 指定安装中排除的软件包或组。例如：

```
%packages
-@Graphical Administration Tools
-autofs
-ipa*compat
%end
```



## 重要

不支持在 Kickstart 文件中只使用 \* 安装所有可用软件包。

您可以使用多个选项更改 **%packages** 部分的默认行为。有些选项可以用于整个软件包选择，其它选项只与特定的组一起使用。

### 其它资源

- [安装软件](#)
- [安装、管理和删除用户空间组件](#)

### A.2.3. 通用软件包选择选项

以下选项可用于 **%packages** 部分。要使用某个选项，请将其附加到软件包选择部分的开头。例如：

```
%packages --multilib --ignoremissing
```

#### **--default**

安装默认软件包组。这与在互动安装过程中的软件包选择页面中没有其他选择时要安装的软件包组对应。

#### **--excludedocs**

不要安装软件包中的任何文档。在大多数情况下，这不包括通常安装在 **/usr/share/doc** 目录中的任何文件，但要排除的特定文件取决于各个软件包。

#### **--ignoremissing**

忽略安装源中缺少的软件包、组、模块流、模块配置集和环境,而不是停止安装来询问安装是中止还是继续。

#### **--instLangs=**

指定要安装的语言列表。这与软件包组级别选择不同。这个选项没有描述应该安装哪些软件包组，而是设定 RPM 宏控制应该安装单个软件包的转换文件。

#### **--multilib**

为多 lib 软件包配置安装的系统，允许在 64 位系统中安装 32 位软件包，并安装在这部分中指定的软件包。

通常在 AMD64 和 Intel 64 系统中，您只能安装 x86\_64 和 noarch 软件包。但是，使用 **--multilib** 选项，您可以自动安装 32 位 AMD 和 i686 Intel 系统软件包（若有）可用。

这只适用于 **%packages** 部分明确指定的软件包。那些只作为相依性安装而没有在 Kickstart 文件中指定的软件包只能安装到需要它们的架构版本中，即使它们可用于更多构架。

用户可将 Anaconda 配置为在安装系统期间以 **multilib** 模式安装软件包。使用以下选项之一启用 **multilib** 模式：

1. 使用以下行配置 Kickstart 文件：

```
%packages --multilib --default
%end
```

2. 在引导安装镜像时添加 **inst.multilib** 引导选项。

**--nocore**

禁用 **@Core** 软件包组的安装，否则该安装默认为始终安装。使用 **--nocore** 禁用 **@Core** 软件包组应当仅用于创建轻量级容器；使用 **--nocore** 安装桌面或服务器系统将导致系统不可用。

**备注**

- 使用 **-@Core** 来排除 **@Core** 软件包组中的软件包不起作用。排除 **@Core** 软件包组的唯一方式是使用 **--nocore** 选项。
- **@Core** 软件包组定义为安装工作系统所需的一组最小软件包。它与 [软件包清单](#) 和 [覆盖范围](#) 中定义的核心软件包无关。

**--excludeWeakdeps**

禁用从弱依赖项安装软件包。这些软件包都链接到由 Recommends 和 supplements 标记组成的所选软件包。默认情况下会安装弱依赖项。

**--retries=**

设置 YUM 试图下载软件包的次数（尝试）。默认值为 10。这个选项只适用于安装过程中，不会影响安装的系统上的 YUM 配置。

**--timeout=**

以秒为单位设置 YUM 超时。默认值为 30。这个选项只适用于安装过程中，不会影响安装的系统上的 YUM 配置。

**A.2.4. 特定软件包组的选项**

这个列表中的选项仅适用于单个软件包组。不要在 Kickstart 文件中的 **%packages** 命令中使用它们，而是将它们附加到组名称中。例如：

```
%packages
@Graphical Administration Tools --optional
%end
```

**--nodefaults**

只安装组的强制软件包，而不是默认选择。

**--optional**

除了安装默认选择外，还要安装在 **\*-comps-repository.architecture.xml** 文件中的组定义中标记为可选的软件包。

有些软件包组（如 **Scientific Support**）没有指定强制或默认软件包 - 仅可选软件包。在这种情况下，必须始终使用 **--optional** 选项，否则不会安装该组中的软件包。

**重要**

**--nodefaults** 和 **--optional** 选项不能一起使用。在安装过程中，您可以使用 **--nodefaults** 只安装必需的软件包，并在安装后在安装的系统上安装可选软件包。

**A.3. KICKSTART 文件中的脚本**

kickstart 文件可以包括以下脚本：

- **%pre**

- **%pre-install**
- **%post**

本节提供有关脚本的以下详情：

- 执行时间
- 可以包含在脚本中的命令类型
- 脚本的目的
- 脚本选项

### A.3.1. %pre 脚本

**%pre** 脚本在加载 Kickstart 文件后立即在系统中运行，但在完全解析并开始安装之前。每个部分必须以 **%pre** 开头并以 **%end** 结尾。

**%pre** 脚本可用于激活和配置联网和存储设备。还可以使用安装环境中可用的脚本来运行脚本。如果您在继续安装之前有需要特殊配置的联网和存储，或者具有设置其他日志参数或环境变量的脚本，则添加 **%pre** 脚本非常有用。

使用 **%pre** 脚本调试问题可能比较困难，因此建议仅在需要时使用 **%pre** 脚本。



#### 重要

Kickstart 的 **%pre** 部分会在安装阶段执行，该阶段发生在安装程序镜像(**inst.stage2**)被提取后：这意味着在 **root** 切换到安装程序环境（安装程序镜像）后，以及 **Anaconda** 安装程序本身启动后。然后，应用 **%pre** 中的配置被应用，可用于从配置的安装存储库中获取软件包，例如，通过 Kickstart 中的 URL。但是，它 **不能** 用于配置网络，以从网络获取镜像 (**inst.stage2**)。

除了安装环境 **/sbin** 和 **/bin** 目录中的大多数实用程序外，还可在 **%pre** 脚本中使用与网络、存储和文件系统相关的命令。

您可以在 **%pre** 部分中访问网络。然而，命名服务还没有被配置，所以只能使用 IP 地址，而不能使用 URL。



#### 注意

**pre** 脚本不会在 **chroot** 环境中运行。

#### A.3.1.1. %pre 脚本部分选项

以下选项可以用来改变预安装脚本的行为。要使用某个选项，请将其附加到脚本开头的 **%pre** 行中。例如：

```
%pre --interpreter=/usr/libexec/platform-python
-- Python script omitted --
%end
```

**--interpreter=**

允许指定不同的脚本语言，如 Python。可以使用系统中可用的脚本语言；在大多数情况下，它们是 `/usr/bin/sh`、`/usr/bin/bash` 和 `/usr/libexec/platform-python`。

请注意，`platform-python` 解释器使用 Python 版本 3.6。对于新路径和版本，您必须将您的针对以前的 RHEL 版本的 Python 脚本进行更改，以适用于新的路径和版本。另外，`platform-python` 用于系统工具：使用安装环境之外的 `python36` 软件包。有关 Red Hat Enterprise Linux 中 Python 的详情，请参考 [配置基本系统设置](#) 中的 [Python 简介](#)。

### **--erroronfail**

如果脚本失败，显示错误并停止安装。错误消息会指示您记录故障原因的位置。安装的系统可能会处于不稳定且无法引导的状态。您可以使用 `inst.nokill` 选项调试脚本。

### **--log=**

将脚本的输出记录到指定的日志文件中。例如：

```
%pre --log=/tmp/ks-pre.log
```

## A.3.2. %pre-install 脚本

`pre-install` 脚本中的命令会在以下任务完成后运行：

- 系统已被分区
- 文件系统创建并挂载在 `/mnt/sysroot` 下
- 网络已根据任何引导选项和 `kickstart` 命令进行配置

每个 `%pre-install` 部分必须以 `%pre-install` 开头并以 `%end` 结尾。

`%pre-install` 脚本可用于修改安装，并在软件包安装之前添加带有保证 ID 的用户和组。

建议您在安装所需的任何修改中使用 `%post` 脚本。只有在 `%post` 脚本对所需修改不够时才使用 `%pre-install` 脚本。

`pre-install` 脚本不会在 `chroot` 环境中运行。

### A.3.2.1. %pre-install script 部分选项

以下选项可用于更改 `pre-install` 脚本的行为。要使用某个选项，请将其附加到脚本开头的 `%pre-install` 行中。例如：

```
%pre-install --interpreter=/usr/libexec/platform-python
-- Python script omitted --
%end
```

您可以有多个 `%pre-install` 部分，它们具有相同的或不同的解释器。它们按照它们在 Kickstart 文件中的顺序进行评估。

### **--interpreter=**

允许指定不同的脚本语言，如 Python。可以使用系统中可用的脚本语言；在大多数情况下，它们是 `/usr/bin/sh`、`/usr/bin/bash` 和 `/usr/libexec/platform-python`。

`platform-python` 解释器使用 Python 版本 3.6。对于新路径和版本，您必须将您的针对以前的 RHEL 版本的 Python 脚本进行更改，以适用于新的路径和版本。另外，`platform-python` 用于系统工具：使用安装环境之外的 `python36` 软件包。有关 Red Hat Enterprise Linux 中 Python 的详情，请参考 [配置基本系统设置](#) 中的 [Python 简介](#)。

**--erroronfail**

如果脚本失败，显示错误并停止安装。错误消息会指示您记录故障原因的位置。安装的系统可能会处于不稳定且无法引导的状态。您可以使用 **inst.nokill** 选项调试脚本。

**--log=**

将脚本的输出记录到指定的日志文件中。例如：

```
%pre-install --log=/mnt/sysroot/root/ks-pre.log
```

**A.3.3. %post 脚本**

**%post** 脚本是安装后脚本，可在安装完成后运行，但在第一次重启系统前运行。您可以使用这部分来运行任务，比如系统订阅。

您可以添加系统在安装结束后但在第一次重启该系统之前要运行的命令。此部分必须以 **%post** 开头并以 **%end** 结尾。

**%post** 部分可用于安装其他软件或配置其他名称服务器等功能。**post-install** 脚本是一个 **chroot** 环境中运行的，因此，从安装介质中复制脚本或 RPM 软件包等任务在默认情况下不起作用。您可以使用 **--nochroot** 选项更改此行为，如下所述。然后 **%post** 脚本将在安装环境中运行，而不是在安装的目标系统中的 **chroot** 中运行。

由于安装后脚本在 **chroot** 环境中运行，因此大多数 **systemctl** 命令将拒绝执行任何操作。

在执行 **%post** 部分的过程中，必须仍然插入安装介质。

**A.3.3.1. %post 脚本部分选项**

以下选项可以用来改变安装后脚本的行为。要使用某个选项，请将其附加到脚本开头的 **%post** 行中。例如：

```
%post --interpreter=/usr/libexec/platform-python
-- Python script omitted --
%end
```

**--interpreter=**

允许指定不同的脚本语言，如 Python。例如：

```
%post --interpreter=/usr/libexec/platform-python
```

可以使用系统中可用的脚本语言；在大多数情况下，它们是 **/usr/bin/sh**、**/usr/bin/bash** 和 **/usr/libexec/platform-python**。

**platform-python** 解释器使用 Python 版本 3.6。对于新路径和版本，您必须将您的针对以前的 RHEL 版本的 Python 脚本进行更改，以适用于新的路径和版本。另外，**platform-python** 用于系统工具：使用安装环境之外的 **python36** 软件包。有关 Red Hat Enterprise Linux 中 Python 的详情，请参考 [配置基本系统设置](#) 中的 [Python 简介](#)。

**--nochroot**

允许您指定在 **chroot** 环境之外运行的命令。

以下示例将 **/etc/resolv.conf** 文件复制到刚安装的文件系统中。

```
%post --nochroot
cp /etc/resolv.conf /mnt/sysroot/etc/resolv.conf
%end
```

### --erroronfail

如果脚本失败，显示错误并停止安装。错误消息会指示您记录故障原因的位置。安装的系统可能会处于不稳定且无法引导的状态。您可以使用 **inst.nokill** 选项调试脚本。

### --log=

将脚本的输出记录到指定的日志文件中。无论是否使用 the- **nochroot** 选项，日志文件的路径都必须考虑。例如，没有 **--nochroot**:

```
%post --log=/root/ks-post.log
```

使用 **--nochroot**:

```
%post --nochroot --log=/mnt/sysroot/root/ks-post.log
```

### A.3.3.2. 示例：在安装后脚本中挂载 NFS

这个 **%post** 部分的示例挂载 NFS 共享并执行位于 **/usr/new-machines/** 上的名为 **runme** 的脚本。在 Kickstart 模式中不支持 NFS 文件锁定，因此需要使用 **-o nolock** 选项。

```
# Start of the %post section with logging into /root/ks-post.log
%post --log=/root/ks-post.log

# Mount an NFS share
mkdir /mnt/temp
mount -o nolock 10.10.0.2:/usr/new-machines /mnt/temp
openvt -s -w -- /mnt/temp/runme
umount /mnt/temp

# End of the %post section
%end
```

### A.3.3.3. 示例：将 subscription-manager 作为安装后脚本运行

Kickstart 安装中最常用的安装脚本之一就是使用 Red Hat Subscription Manager 自动注册安装的系统。以下是 **%post** 脚本中的自动订阅示例：

```
%post --log=/root/ks-post.log
subscription-manager register --username=admin@example.com --password=secret --auto-attach
%end
```

**subscription-manager** 命令行脚本在红帽订阅管理服务器（客户门户网站订阅管理、Satellite 6 或者 CloudForms System Engine）中注册系统。这个脚本还可用来自动在该系统中分配或者附加与该系统最匹配的订阅。在客户门户网站中注册时，请使用红帽网络登录证书。当在 Satellite 6 或者 CloudForms System Engine 中注册时，您可能需要指定更多 **subscription-manager** 选项 **--serverurl**、**--org**、**--environment** 以及您的本地管理员提供的凭证。请注意，以 **--org --activationkey** 组合格式的凭证是避免在共享的 kickstart 文件中公开 **--username --password** 值的好方法。

注册命令中可以使用附加选项为系统设置首选服务等级，并为那些需要在旧流中继续修复的延长更新支持订阅客户限制对特定 RHEL 次要版本的更新和勘误。

另请参阅红帽客户门户网站中的[如何在 kickstart 文件中使用 subscription-manager?](#) 文章来了解有关在 Kickstart `%post` 部分中使用 `subscription-manager` 的更多信息。

## A.4. ANACONDA 配置部分

可以在 Kickstart 文件的 `%anaconda` 部分配置其他安装选项。这部分控制安装系统的用户界面行为。

这部分必须放在 Kickstart 文件的末尾（在 Kickstart 命令后面），且必须以 `%anaconda` 开头并以 `%end` 结尾。

目前，`%anaconda` 部分中唯一可以使用的命令是 `pwpolicy`。

### 例 A.1. %anaconda 脚本示例

以下是 `%anaconda` 部分示例：

```
%anaconda
pwpolicy root --minlen=10 --strict
%end
```

这个示例 `%anaconda` 部分设置了一个密码策略，它要求 `root` 密码长度至少为 10 个字符，并严格禁止不满足此要求的密码。

## A.5. KICKSTART 错误处理部分

从 Red Hat Enterprise Linux 7 开始，当安装程序中遇到严重错误时，Kickstart 安装会运行自定义脚本。示例场景在为安装请求的软件包中有一个错误，如果配置中指定，VNC 无法启动，或者在扫描存储设备时出现错误。如果是此类事件，安装将中止。要分析这些事件，安装程序会按照 Kickstart 文件中提供的所有 `%onerror` 脚本运行。如果出现回溯，您可以运行 `%onerror` 脚本。

每个 `%onerror` 脚本都需要以 `%end` 结尾。

您可以使用 `inst.cmdline` 使每个错误出现严重错误来强制实施错误处理程序。

错误处理部分接受以下选项：

### --erroronfail

如果脚本失败，显示错误并停止安装。错误消息会指示您记录故障原因的位置。安装的系统可能会处于不稳定且无法引导的状态。您可以使用 `inst.nokill` 选项调试脚本。

### --interpreter=

允许指定不同的脚本语言，如 Python。例如：

```
%onerror --interpreter=/usr/libexec/platform-python
```

可以使用系统中可用的脚本语言；在大多数情况下，它们是 `/usr/bin/sh`、`/usr/bin/bash` 和 `/usr/libexec/platform-python`。

`platform-python` 解释器使用 Python 版本 3.6。对于新路径和版本，您必须将您的针对以前的 RHEL 版本的 Python 脚本进行更改，以适用于新的路径和版本。另外，`platform-python` 用于系统工具：使用安装环境之外的 `python36` 软件包。有关 Red Hat Enterprise Linux 中 Python 的详情，请参考 [配置](#)

基本系统设置中的 [Python 简介](#)。

## --log=

将脚本的输出记录到指定的日志文件中。

## A.6. KICKSTART 附加组件部分

从 Red Hat Enterprise Linux 7 开始，Kickstart 安装支持附加组件。这些附加组件可以在很多方面扩展基本 Kickstart(Anaconda)功能。

要在 Kickstart 文件中使用附加组件，请使用 **%addon *addon\_name options*** 命令，并使用 **%end** 语句结束命令，这和预安装和安装后脚本部分类似。例如：如果要使用默认由 Anaconda 分配的 Kdump 附加组件，请使用以下命令：

```
%addon com_redhat_kdump --enable --reserve-mb=auto
%end
```

**%addon** 命令不包含任何自己的选项 - 所有选项都依赖于实际附加组件。

## 附录 B. KICKSTART 命令和选项参考

此参考是 Red Hat Enterprise Linux 安装程序支持的所有 Kickstart 命令的完整列表。这些命令按字母顺序排序为几个广泛类别。如果某个命令可位于多个类别下，它将列在所有这些类别中。

### B.1. KICKSTART 的修改

以下章节描述了 Red Hat Enterprise Linux 8 中 Kickstart 命令和选项的更改。

#### 在 RHEL 8 中弃用了 `auth` 或 `authconfig`

因为已经删除了 `authconfig` 工具和软件包，Red Hat Enterprise Linux 8 中弃用了 `auth` 或 `authconfig` Kickstart 命令。

与命令行中的 `authconfig` 命令类似，Kickstart 脚本中的 `authconfig` 命令现在使用 `authselect-compat` 工具运行新的 `authselect` 工具。有关此兼容性层及其已知问题的描述，请参阅 [authselect-migration\(7\)](#) 手册页。安装程序将自动检测弃用命令的使用并在系统上安装 `authselect-compat` 软件包来提供兼容性层。

#### Kickstart 不再支持 Btrfs

从 Red Hat Enterprise Linux 8 起不再支持 Btrfs 文件系统。因此，图形用户界面(GUI)和 Kickstart 命令不再支持 Btrfs。

#### 使用之前 RHEL 发行本中的 Kickstart 文件

如果您正在使用之前的 RHEL 版本中的 Kickstart 文件,请参阅 [RHEL 8 文档中](#) 的 *Repositories* 部分，以了解有关 Red Hat Enterprise Linux 8 BaseOS 和 AppStream 软件仓库的详情。

### B.1.1. 弃用的 Kickstart 命令和选项

在 Red Hat Enterprise Linux 8 中弃用了以下 Kickstart 命令和选项。

如果只列出具体选项，则基础命令及其它选项仍可用且没有弃用。

- `auth` 或 `authconfig` - 使用 `authselect` 替代
- `device`
- `deviceprobe`
- `dmraid`
- `install` - 使用子命令或者方法作为命令
- `multipath`
- `bootloader --upgrade`
- `ignoredisk --interactive`
- `partition --active`
- `reboot --kexec`
- `syspurpose` - 使用 `subscription-manager syspurpose` 替代

除 **auth** 或 **authconfig** 命令外，使用 Kickstart 文件中的命令会在日志中打印警告信息。

您可以使用 **inst.ksstrict** 引导选项将已弃用的命令警告转换为错误，但 **auth** 或 **authconfig** 命令除外。

## B.1.2. 删除的 Kickstart 命令和选项

在 Red Hat Enterprise Linux 8 中完全删除了以下 Kickstart 命令和选项。在 Kickstart 文件中使用它们将导致错误。

- **device**
- **deviceprobe**
- **dmraid**
- **install** - 使用子命令或者方法作为命令
- **multipath**
- **bootloader --upgrade**
- **ignoredisk --interactive**
- **partition --active**
- **harddrive --biospart**
- **upgrade** (该命令之前已经弃用。)
- **btrfs**
- **part/partition btrfs**
- **part --fstype btrfs** 或 **partition --fstype btrfs**
- **logvol --fstype btrfs**
- **raid --fstype btrfs**
- **unsupported\_hardware**

如果只列出具体选项和值，则基础命令及其它选项仍可用且没有被删除。

## B.2. 用于安装程序配置和流量控制的 KICKSTART 命令

这个列表中的 Kickstart 命令可控制安装模式和安装过程，以及最后发生什么。

### B.2.1. cdrom

**cdrom** Kickstart 命令是可选的。它使用系统上的第一个光驱执行安装。仅使用命令一次。

#### 语法

```
cdrom
```

## 备注

- 在以前的版本中，**cdrom** 命令必须与 **install** 命令一同使用。**install** 命令已弃用，**cdrom** 可以自行使用，因为它表示 **install**。
- 这个命令没有选项。
- 要实际运行安装，您必须指定 **cdrom,harddrive,hmc,nfs,liveimg,ostreosetup,rhsm**, 或 **url** 其中之一，除非 **inst.repo** 选项在内核命令行上指定了。

## B.2.2. cmdline

**cmdline** Kickstart 命令是可选的。它以完全非互动的命令行模式执行安装。任何互动提示都会终止安装。仅使用命令一次。

## 语法

```
cmdline
```

## 备注

- 对于完全自动安装，您必须在 Kickstart 文件中指定可用模式之一（**图形**、**文本** 或 **命令行**），或者必须使用 **console=** 引导选项。如果没有指定模式，系统会尽可能使用图形模式，或者提示您从 VNC 和文本模式中选择。
- 这个命令没有选项。
- 这个模式在带有 x3270 终端的 64 位 IBM Z 系统上很有用。

## B.2.3. driverdisk

**driverdisk** Kickstart 命令是可选的。使用它为安装程序提供额外的驱动程序。

可在 Kickstart 安装过程中使用驱动程序磁盘提供默认不包括的额外驱动程序。您必须将驱动程序磁盘内容复制到系统的磁盘上分区的根目录中。然后，您必须使用 **driverdisk** 命令指定安装程序是否应该查找驱动程序磁盘及其位置。仅使用命令一次。

## 语法

```
driverdisk [partition]--source=url]--biospart=biospart]
```

## 选项

您必须以以下一种方式指定驱动程序磁盘的位置：

- **partition** - 包含驱动程序磁盘的分区。该分区必须指定为完整路径（例如 **/dev/sdb1**），而不仅仅是分区名称（例如 **sdb1**）。
- **--source=** - 驱动程序磁盘的 URL。示例包括：

```
driverdisk --source=ftp://path/to/dd.img
driverdisk --source=http://path/to/dd.img
driverdisk --source=nfs:host:/path/to/dd.img
```

- **--biospart=** - 包含驱动程序磁盘（如 **82p2**）的 BIOS 分区。

## 注

也可以从本地磁盘或类似的设备加载驱动程序磁盘，而不是通过网络或从 **initrd** 加载。按照以下步骤操作：

1. 在磁盘驱动器、USB 或者任何类似的设备上载入驱动程序磁盘。
2. 将标签（如 *DD*）设置为这个设备。
3. 在您的 Kickstart 文件中添加以下行：

```
driverdisk LABEL=DD:/e1000.rpm
```

使用特定标签替换 *DD*，使用特定名称替换 *e1000.rpm*。使用 **inst.repo** 命令支持的任何内容，而不是 *LABEL* 来指定您的磁盘驱动器。

### B.2.4. EULA

**eula** Kickstart 命令是可选的。使用这个选项在没有用户互动的情况下接受最终用户许可证协议(End User License Agreement,EULA)。指定这个选项可防止 Initial Setup 在完成安装并第一次重启系统后提示您接受该许可证。仅使用命令一次。

#### 语法

```
eula [--agreed]
```

#### 选项

- **--agreed**（必需） - 接受 EULA。必须始终使用这个选项，否则 **eula** 命令就无意义。

### B.2.5. firstboot

**firstboot** Kickstart 命令是可选的。它决定了系统首次启动时 **Initial Setup** 应用程序是否启动。如果启用，则必须安装 **initial-setup** 软件包。如果没有指定，这个选项默认是禁用的。仅使用命令一次。

#### 语法

```
firstboot OPTIONS
```

#### 选项

- **--enable** 或 **--enabled** - 系统第一次启动时启动 Initial Setup。
- **--disable** 或 **--disabled** - 系统第一次引导时不会启动 Initial Setup。
- **--reconfig** - 以重新配置模式在引导时启用 Initial Setup。这个模式除了默认选项外，还启用了 root 密码、时间和日期以及网络和主机名配置选项。

### B.2.6. 图形化

**graphical** Kickstart 命令是可选的。它在图形模式下执行安装。这是默认值。仅使用命令一次。

#### 语法

```
■
```

```
graphical [--non-interactive]
```

### 选项

- **--non-interactive** - 以完全非互动模式执行安装。这个模式将在用户交互需要时终止安装。

### 备注

- 对于完全自动安装，您必须在 Kickstart 文件中指定可用模式之一（图形、文本 或 命令行），或者必须使用 **console=** 引导选项。如果没有指定模式，系统会尽可能使用图形模式，或者提示您从 VNC 和文本模式中选择。

## B.2.7. halt

**halt** Kickstart 命令是可选的。

在成功完成安装后停止系统。这和手动安装相似，Anaconda 会显示一条信息并等待用户按任意键来重启系统。在 Kickstart 安装过程中，如果没有指定完成方法，将使用这个选项作为默认选项。仅使用命令一次。

### 语法

```
halt
```

### 备注

- **halt** 命令等同于 **shutdown -H** 命令。详情请查看 *shutdown(8)* man page。
- 有关其他完成方法，请查看 **poweroff**、**reboot** 和 **shutdown** 命令。
- 这个命令没有选项。

## B.2.8. harddrive

**harddrive** Kickstart 命令是可选的。它使用红帽安装树或者本地驱动器中的完整安装 ISO 镜像执行安装。驱动器必须使用安装程序可挂载的文件系统格式化：**ext2**、**ext3**、**ext4**、**vfat** 或 **xfs**。仅使用命令一次。

### 语法

```
harddrive OPTIONS
```

### 选项

- **--partition=** - 要从中安装的分区的名称（如 **sdb2**）。
- **--dir=** - 包含安装树 *variant* 目录或完整安装 DVD 的 ISO 镜像的目录。

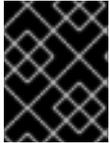
### 示例

```
harddrive --partition=hdb2 --dir=/tmp/install-tree
```

## 备注

- 在以前的版本中，**harddrive** 命令必须与 **install** 命令一同使用。**install** 命令已弃用，并且可以自行使用 **harddrive**，因为它表示 **安装**。
- 要实际运行安装，您必须指定 **cdrom,harddrive,hmc,nfs,liveimg,ostreesetup,rhsm**，或 **url** 其中之一，除非 **inst.repo** 选项在内核命令行上指定了。

## B.2.9. install（已弃用）



### 重要

**install** Kickstart 命令在 Red Hat Enterprise Linux 8 中已弃用。使用它的方法作为单独的命令。

**install** Kickstart 命令是可选的。它指定了默认的安装模式。

## 语法

```
install
installation_method
```

## 备注

- **install** 命令后必须跟随安装方法命令。安装方法命令必须位于单独的行中。
- 方法包括：
  - **cdrom**
  - **harddrive**
  - **hmc**
  - **nfs**
  - **liveimg**
  - **url**

有关方法的详情，请查看其独立参考页面。

## B.2.10. liveimg

**liveimg** Kickstart 命令是可选的。它从磁盘镜像而不是软件包执行安装。仅使用命令一次。

## 语法

```
liveimg --url=SOURCE [OPTIONS]
```

## 必填选项

- **--url=** - 从其中安装的位置。支持的协议包括 **HTTP**、**HTTPS**、**FTP** 和 **file**。

## 可选选项

- **--url=** - 从其中安装的位置。支持的协议包括 **HTTP**、**HTTPS**、**FTP** 和 **file**。
- **--proxy=** - 指定在执行安装时要使用的 **HTTP**、**HTTPS** 或者 **FTP** 代理。
- **--checksum=** - 包含镜像文件的 **SHA256** 校验和的可选参数，用于验证。
- **--noverifyssl** - 连接到 **HTTPS** 服务器时禁用 SSL 验证。

## 示例

```
liveimg --url=file:///images/install/squashfs.img --
checksum=03825f567f17705100de3308a20354b4d81ac9d8bed4bb4692b2381045e56197 --
noverifyssl
```

## 备注

- 镜像可以是来自实时 ISO 镜像的 **squashfs.img** 文件、压缩的 tar 文件 (**.tar**、**.tbz**、**.tgz**、**.txz**、**.tar.bz2**、**.tar.gz** 或 **.tar.xz**)，或者安装介质可以挂载的任何文件系统。支持的文件系统有 **ext2**、**ext3**、**ext4**、**vfat** 和 **xfs**。
- 将 **liveimg** 安装模式与驱动程序磁盘一起使用时，磁盘中的驱动程序不会自动包含在安装的系统中。如有必要，应手动安装这些驱动程序，或在 kickstart 脚本的 **%post** 部分中安装这些驱动程序。
- 要实际运行安装，您必须指定 **cdrom**、**harddrive**、**hmc**、**nfs**、**liveimg**、**ostreeresetup**、**rhsm**，或 **url** 其中之一，除非 **inst.repo** 选项在内核命令行上指定了。
- 在以前的版本中，**liveimg** 命令必须与 **install** 命令一同使用。**install** 命令已弃用，**liveimg** 可以自行使用，因为它代表 **install**。

## B.2.11. logging

**logging** Kickstart 命令是可选的。它控制在安装过程中 Anaconda 的错误日志。它对安装的系统没有影响。仅使用命令一次。

只支持使用 TCP 记录日志。对于远程日志记录，请确保在远程服务器上打开您在 **--port=** 选项中指定的端口号。默认端口为 514。

## 语法

```
logging OPTIONS
```

## 可选选项

- **--host=** - 向给定的远程主机发送日志信息，该主机必须配置有一个 **syslogd** 进程，以接受远程记录。
- **--port=** - 如果远程 **syslogd** 进程使用默认端口以外的端口，请使用这个选项进行设置。
- **--level=** - 指定 tty3 上出现的最小信息级别。无论这个级别是什么，仍会将所有的信息发送到日志文件。可能的值有 **debug**、**info**、**warning**、**error** 或 **critical**。

### B.2.12. mediacheck

**mediacheck** Kickstart 命令是可选的。该命令强制安装程序在开始安装前执行介质检查。因为这个命令需要在执行安装时有人工参与，因此它默认被禁用。仅使用命令一次。

#### 语法

```
mediacheck
```

#### 备注

- 这个 Kickstart 命令等同于 **rd.live.check** 引导选项。
- 这个命令没有选项。

### B.2.13. nfs

**nfs** Kickstart 命令是可选的。它从指定的 NFS 服务器执行安装。仅使用命令一次。

#### 语法

```
nfs OPTIONS
```

#### 选项

- **--server=** - 要从中安装的服务器（主机名或 IP）。
- **--dir=** - 包含安装树 *variant* 目录的目录。
- **--opts=** - 用于挂载 NFS 导出的挂载选项（可选）。

#### 示例

```
nfs --server=nfsserver.example.com --dir=/tmp/install-tree
```

#### 备注

- 在以前的版本中，**nfs** 命令必须与 **install** 命令一同使用。**install** 命令已弃用，**NFS** 可以自行使用，因为它代表 **install**。
- 要实际运行安装，您必须指定 **cdrom,harddrive,hmc,nfs,liveimg,ostreesetup,rhsm**，或 **url** 其中之一，除非 **inst.repo** 选项在内核命令行上指定了。

### B.2.14. ostreesetup

**ostreesetup** Kickstart 命令是可选的。它被用来设置基于 OSTree 的安装。仅使用命令一次。

#### 语法

```
ostreesetup --osname=OSNAME [--remote=REMOTE] --url=URL --ref=REF [--nogpg]
```

#### 必须的选项

- **--osname=*OSNAME*** - 用于操作系统安装的管理根。
- **--url=*URL*** - 要从中安装的存储库的 URL。
- **--ref=*REF*** - 用于安装的软件仓库中的分支名称。

可选选项：

- **--remote=*REMOTE*** - 远程存储库位置。
- **--nogpg** - 禁用 GPG 密钥验证。

备注

- 有关 OSTree 工具的更多信息，请参阅上游文档：<https://ostreedev.github.io/ostree/>

## B.2.15. poweroff

**poweroff** Kickstart 命令是可选的。它会在安装成功后关闭系统并关闭电源。通常，在手动安装过程中，Anaconda 会显示一条信息并等待用户按任意键来重新引导系统。仅使用命令一次。

语法

```
poweroff
```

备注

- **poweroff** 选项等同于 **shutdown -P** 命令。详情请查看 *shutdown(8)* man page。
- 有关其他完成方法，请查看 **halt**、**reboot** 和 **shutdown** Kickstart 命令。如果没有在 Kickstart 文件中明确指定其他方法，则 **halt** 选项是默认的完成方法。
- **poweroff** 命令高度依赖于所使用的系统硬件。特别是，某些硬件部件如 BIOS、APM（高级电源管理）和 ACPI（高级配置和电源接口）必须能和系统内核交互。有关系统 APM/ACPI 功能的更多信息，请参阅硬件文档。
- 这个命令没有选项。

## B.2.16. reboot

**reboot** Kickstart 命令是可选的。它指示安装程序在安装成功（没有参数）后重启。通常，Kickstart 会显示信息并等待用户按任意键来重新引导系统。仅使用命令一次。

语法

```
reboot OPTIONS
```

选项

- **--eject** - 在重新启动前尝试弹出可引导介质（DVD、USB 或其他介质）。
- **--kexec** - 使用 **kexec** 系统调用而不是执行完全重启，这样可立即将安装的系统加载到内存中，绕过通常由 BIOS 或固件执行的硬件初始化。



## 重要

这个选项已弃用，仅作为技术预览使用。有关红帽对技术预览功能支持范围的详情，请查看 [技术预览功能支持范围](#) 文档。

使用 **kexec** 时，设备寄存器（通常会在系统完全重启后清除）可能会继续填写数据，这可能会给某些设备驱动程序造成问题。

## 备注

- 使用 **reboot** 选项 *可能会导致安装无限循环*，具体取决于安装介质和方法。
- **reboot** 选项等同于 **shutdown -r** 命令。详情请查看 `shutdown(8)` man page。
- 在 64 位 IBM Z 中使用命令行模式安装时，指定 **reboot** 以完全自动安装。
- 有关其他完成方法，请查看 **halt**、**poweroff** 和 **shutdown** Kickstart 选项。如果没有在 Kickstart 文件中明确指定其他方法，则 **halt** 选项是默认的完成方法。

## B.2.17. rhsm

**rhsm** Kickstart 命令是可选的。它指示安装程序从 CDN 注册并安装 RHEL。仅使用命令一次。

在注册系统时，**rhsm** Kickstart 命令不需要使用自定义 **%post** 脚本。

## 选项

- **--organization=** - 使用组织 ID 从 CDN 注册和安装 RHEL。
- **--activation-key=** - 使用激活码从 CDN 注册和安装 RHEL。选项可以多次使用，每个激活密钥使用一次，只要使用的激活密钥已注册到订阅中。
- **--connect-to-insights** - 将目标系统连接到 Red Hat Insights。
- **--proxy=** - 设置 HTTP 代理。
- 要使用 **rhsm** Kickstart 命令将安装源存储库切换到 CDN，您必须满足以下条件：
  - 在内核命令行上，您已使用 **inst.stage2=<URL>** 来获取安装镜像，但没有使用 **inst.repo=** 指定安装源。
  - 在 Kickstart 文件中，您尚未使用 **url**、**cdrom**、**harddrive**、**liveimg**、**nfs** 和 **ostree** 设置命令指定安装源。
- 使用引导选项指定或者包含在 Kickstart 文件中的安装源 URL 优先于 CDN，即使 Kickstart 文件包含带有有效凭证的 **rhsm** 命令。已注册该系统，但会通过 URL 安装源进行安装。这样可保证早期安装进程正常运行。

## B.2.18. shutdown

**shutdown** Kickstart 命令是可选的。它会在安装成功完成后关闭系统。仅使用命令一次。

## 语法

```
shutdown
```

## 备注

- **shutdown** Kickstart 选项等同于 **shutdown** 命令。详情请查看 *shutdown(8)* man page。
- 有关其他完成方法，请查看 **halt**、**poweroff** 和 **reboot** Kickstart 选项。如果没有在 Kickstart 文件中明确指定其他方法，则 **halt** 选项是默认的完成方法。
- 这个命令没有选项。

## B.2.19. sshpw

**sshpw** Kickstart 命令是可选的。

在安装过程中，您可以与安装程序交互并通过 **SSH** 连接监控其进度。使用 **sshpw** 命令创建登录的临时帐户。该命令的每个实例都会创建一个只存在于安装环境中的单独帐户。这些不会转移到系统里。

## 语法

```
sshpw --username=name [OPTIONS] password
```

## 必填选项

- **--username=*name*** - 提供用户名称。这个选项是必需的。
- ***password*** - 用户要使用的密码。这个选项是必需的。

## 可选选项

- **--iscrypted** - 如果给出这个选项，则假设 *password* 参数已被加密。这个选项与 **--plaintext** 相互排斥。要生成加密的密码，可以使用 Python:

```
$ python3 -c 'import crypt,getpass;pw=getpass.getpass();print(crypt.crypt(pw) if (pw==getpass.getpass("Confirm: ")) else exit())'
```

这会使用随机 salt 为密码生成 sha512 兼容哈希。

- **--plaintext** - 如果给出这个选项，则假设 *password* 参数为纯文本。这个选项与 **--iscrypted** 相互排斥
- **--lock** - 如果给出这个选项，则默认锁定这个帐户。这意味着用户无法从控制台登录。
- **--sshKey** - 如果给出这个选项，则 *<password>* 字符串被解释为 ssh 密钥值。

## 备注

- 默认情况下，**ssh** 服务器不会在安装过程中启动。要使 **ssh** 在安装过程中可用，使用内核引导选项 **inst.sshd** 引导系统。
- 如果要禁用 root **ssh** 访问，同时允许其他用户 **ssh** 访问，请使用：

```
sshpw --username=example_username example_password --plaintext
sshpw --username=root example_password --lock
```

- 要简单地禁用 root **ssh** 访问，请使用：

■

```
sshpw --username=root example_password --lock
```

## B.2.20. text

**text** Kickstart 命令是可选的。它在文本模式下执行 Kickstart 安装。Kickstart 安装默认是以图形模式执行的。仅使用命令一次。

### 语法

```
text [--non-interactive]
```

### 选项

- **--non-interactive** - 以完全非互动模式执行安装。这个模式将在用户交互需要时终止安装。

### 备注

- 对于完全自动安装，您必须在 Kickstart 文件中指定可用模式之一（**图形**、**文本** 或 **命令行**），或者必须使用 **console=** 引导选项。如果没有指定模式，系统会尽可能使用图形模式，或者提示您从 VNC 和文本模式中选择。

## B.2.21. url

**url** Kickstart 命令是可选的。它用来使用 FTP、HTTP 或者 HTTPS 协议从远程服务器上的安装树镜像进行安装。您只能指定一个 URL。仅使用命令一次。

您必须指定 **--url**、**--metalink** 或 **--mirrorlist** 选项之一。

### 语法

```
url --url=FROM[OPTIONS]
```

### 选项

- **--url=*FROM*** - 指定要从中安装的 **HTTP**、**HTTPS**、**FTP** 或文件位置。
- **--mirrorlist=** - 指定要从中安装的镜像 URL。
- **--proxy=** - 指定在安装过程中要使用的 **HTTP**、**HTTPS** 或者 **FTP** 代理。
- **--noverifyssl** - 连接到 **HTTPS** 服务器时禁用 SSL 验证。
- **--metalink=*URL*** - 指定要从中安装的 metalink URL。变量替换用于 *URL* 中的 **\$releasever** 和 **\$basearch**。

### 示例

- 从 HTTP 服务器安装：

```
url --url=http://server/path
```

- 从 FTP 服务器安装：

```
url --url=ftp://username:password@server/path
```

## 注

- 在以前的版本中，**url** 命令必须与 **install** 命令一同使用。**install** 命令已弃用，并且 **url** 可以自行使用，因为它代表 **install**。
- 要实际运行安装，您必须指定 **cdrom**, **harddrive**, **hmc**, **nfs**, **liveimg**, **ostreosetup**, **rhsm**, 或 **url** 其中之一，除非 **inst.repo** 选项在内核命令行上指定了。

## B.2.22. vnc

**vnc** Kickstart 命令是可选的。它允许通过 VNC 远程查看图形安装。

与文本模式相比，这个模式通常是首选模式。因为在文本模式中有某些大小和语言的限制。如果没有附加选项，这个命令将在不需要密码的系统中启动 VNC 服务器，并显示连接它所需要的详情。仅使用命令一次。

## 语法

```
vnc [--host=host_name] [--port=port] [--password=password]
```

## 选项

### --host=

连接在给定主机名中侦听的 VNC viewer 进程。

### --port=

提供远程 VNC viewer 进程侦听的端口。如果没有提供，Anaconda 将使用 VNC 默认端口 5900。

### --password=

设定必须提供用来连接到 VNC 会话的密码。这是可选的，但推荐使用。

## 其它资源

- [准备使用 PXE 从网络安装](#)

## B.2.23. %include

**%include** Kickstart 命令是可选的。

使用 **%include** 命令，将另一文件的内容包含在 Kickstart 文件中，就好像其内容在 Kickstart 文件中的 **%include** 命令的位置一样。

它的内容只在 **%pre** 脚本部分之后评估，因此可用于将脚本生成的文件包含在 **%pre** 部分中。要在评估 **%pre** 部分之前包含文件，请使用 **%ksappend** 命令。

## 语法

```
%include path/to/file
```

## B.2.24. %ksappend

**%ksappend** Kickstart 命令是可选的。

使用 **%ksappend** 命令将另一个文件的内容包含在 Kickstart 文件中，就好像其内容在 Kickstart 文件中 **%ksappend** 命令的位置一样。

这个内容在 **%pre** 脚本部分之前评估，这与 **%include** 命令包括的内容不同。

## 语法

```
%ksappend path/to/file
```

## B.3. KICKSTART 命令进行系统配置

这个列表中的 Kickstart 命令配置结果系统的更多详情，比如用户、库或服务。

### B.3.1. auth 或 authconfig（已弃用）



#### 重要

使用新的 **authselect** 命令而不是已弃用的 **auth** 或 **authconfig** Kickstart 命令。**auth** 和 **authconfig** 仅适用于有限的向后兼容性。

**auth** 或 **authconfig** Kickstart 命令是可选的。它使用 **authconfig** 工具为系统设置身份验证选项，也可以在安装完成后在命令行中运行该工具。仅使用命令一次。

## 语法

```
authconfig [OPTIONS]
```

## 备注

- 在以前的版本中，**auth** 或 **authconfig** Kickstart 命令称为 **authconfig** 工具。在 Red Hat Enterprise Linux 8 中已弃用这个工具。这些 Kickstart 命令现在使用 **authselect-compat** 工具调用新的 **authselect** 工具。有关兼容性层及其已知问题的描述，请参阅 *authselect-migration(7)* 手册页。安装程序将自动检测弃用命令的使用并在系统上安装 **authselect-compat** 软件包以提供兼容性层。
- 默认使用影子密码。
- 使用带有 **SSL** 协议的 OpenLDAP 时，请确保在服务器配置中禁用了 **SSLv2** 和 **SSLv3** 协议。这是因为 POODLE SSL 漏洞(CVE-2014-3566)。详情请查看 <https://access.redhat.com/solutions/1234843>。

### B.3.2. authselect

**authselect** Kickstart 命令是可选的。它使用 **authselect** 命令为系统设置身份验证选项，也可以在安装完成后在命令行中运行该命令。仅使用命令一次。

## 语法

```
authselect [OPTIONS]
```

## 备注

- 这个命令会将所有选项传递给 **authselect** 命令。详情请查看 *authselect(8)* 手册页和 **authselect --help** 命令。
- 这个命令替换了 Red Hat Enterprise Linux 8 中已弃用的 **auth** 或 **authconfig** 命令以及 **authconfig** 工具。
- 默认使用影子密码。
- 使用带有 **SSL** 协议的 OpenLDAP 时，请确保在服务器配置中禁用了 **SSLv2** 和 **SSLv3** 协议。这是因为 POODLE SSL 漏洞(CVE-2014-3566)。详情请查看 <https://access.redhat.com/solutions/1234843>。

## B.3.3. firewall

**firewall** Kickstart 命令是可选的。它为安装的系统指定防火墙配置。

### 语法

```
firewall --enabled|--disabled [incoming] [OPTIONS]
```

### 必填选项

- **--enabled** 或 **--enable** - 拒绝那些不是响应出站请求（如 DNS 回复或 DHCP 请求）的传入连接。如果需要访问在这个机器中运行的服务，您可以选择允许指定的服务通过防火墙。
- **--disabled** 或 **--disable** - 不配置任何 iptables 规则。

### 可选选项

- **--trust** - 在此处列出设备，如 **em1**，允许进出该设备的所有流量通过防火墙。要列出多个设备，请多次使用这个选项，如 **--trust em1 --trust em2**。不要使用逗号分隔的格式，如 **--trust em1、em2**。
- **--remove-service** - 不允许服务穿过防火墙。
- *incoming* - 使用以下服务中的一个或多个来替换，从而允许特定的服务穿过防火墙。
  - **--ssh**
  - **--smtp**
  - **--http**
  - **--ftp**
- **--port=** - 您可以使用 `port:protocol` 格式指定允许通过防火墙的端口。例如，要允许 IMAP 通过您的防火墙，可指定 **imap:tcp**。数字端口也可以明确指定；例如，要允许 UDP 数据包在端口 1234 到，请指定 **1234:udp**。要指定多个端口，用逗号将它们隔开。
- **--service=** - 此选项提供允许服务穿过防火墙的更高级别方法。有些服务（如 **cups**、**vahi** 等）需要打开多个端口或其他特殊配置才能使服务正常工作。您可以使用 **--port** 选项指定各个端口，或者指定 **--service=** 并一次性全部打开它们。  
有效选项是 **firewalld** 软件包中 **firewall-offline-cmd** 程序可识别的任何内容。如果 **firewalld** 服务正在运行，**firewall-cmd --get-services** 会提供已知服务名称的列表。

- **--use-system-defaults** - 完全不配置防火墙。这个选项告诉 anaconda 不做任何工作，并允许系统依赖软件包或者 ostree 提供的默认值。如果将这个选项与其它选项一同使用，则将忽略所有其他选项。

### B.3.4. group

**group** Kickstart 命令是可选的。它在系统中创建新用户组。

```
group --name=name [--gid=gid]
```

#### 必填选项

- **--name=** - 提供组的名称。

#### 可选选项

- **--gid=** - 组的 GID。如果没有提供，则默认使用下一个可用的非系统 GID。

#### 备注

- 如果具有指定名称或 GID 的组群已经存在，这个命令会失败。
- **user** 命令可用于为新创建的用户创建新组。

### B.3.5. keyboard (必需)

**keyboard** Kickstart 命令是必需的。它为系统设置一个或多个可用的键盘布局。仅使用命令一次。

#### 语法

```
keyboard --vckeymap|--xlayouts OPTIONS
```

#### 选项

- **--vckeymap=** - 指定应使用的 **VConsole** 键映射。有效名称与 `/usr/lib/kbd/keymaps/xkb/` 目录中的文件列表对应，没有 `.map.gz` 扩展名。
- **--xlayouts=** - 指定 X 布局列表，该列表应当用作逗号分隔的列表，没有空格。接受与 **setxkbmap(1)** 相同格式的值，可以是 **布局** 格式（如 `as cz`），也可以是 **布局 (变体)** 格式（如 `cz (qwerty)`）。可以在下方的 **xkeyboard-config(7)** man page 中查看所有可用 **布局**。
- **--switch=** - 指定布局切换选项列表（在多个键盘布局之间切换的快捷方式）。必须使用逗号分开多个选项，没有空格。接受与 **setxkbmap(1)** 相同格式的值。您可以在 **xkeyboard-config(7)** man page 上的 **Options** 下查看可用的切换选项。

#### 备注

- 必须使用 **--vckeymap=** 或 **--xlayouts=** 选项。

#### 示例

以下示例使用 **--xlayouts=** 选项设置了两种键盘布局（**English (US)** 和 **Czech (qwerty)**），并允许使用 **Alt+Shift** 在它们之间进行切换：

```
keyboard --xlayouts=us,'cz (qwerty)' --switch=grp:alt_shift_toggle
```

### B.3.6. lang (必需)

**lang** Kickstart 命令是必需的。它设置了在安装过程中使用的语言以及系统的默认语言。仅使用命令一次。

#### 语法

```
lang language [--addsupport=language,...]
```

#### 必填选项

- **language** - 安装对此语言的支持并将其设置为系统默认。

#### 可选选项

- **--addsupport=** - 添加对其他语言的支持。格式为使用逗号分隔的列表，无空格。例如：

```
lang en_US --addsupport=cs_CZ,de_DE,en_UK
```

#### 备注

- **locale -a | grep \_** 或 **localectl list-locales | grep \_** 命令返回支持的区域列表。
- 文本模式安装中不支持某些语言（比如中文、日语、韩文和印度的语言）。如果您使用 **lang** 命令指定这些语言中的一种，安装过程将继续使用英语，但安装的系统会使用您选择的语言作为其默认语言。

#### 示例

要将语言设置为英语，Kickstart 文件应包含以下行：

```
lang en_US
```

### B.3.7. module

**module** Kickstart 命令是可选的。使用这个命令在 kickstart 脚本中启用软件包模块流。

#### 语法

```
module --name=NAME [--stream=STREAM]
```

#### 必填选项

##### **--name=**

指定要启用的模块名称。使用实际名称替换 *NAME*。

#### 可选选项

##### **--stream=**

指定要启用的模块流的名称。将 *STREAM* 替换为实际名称。

您不需要为定义的默认流的模块指定此选项。对于没有默认流的模块，这个选项是强制的，省略它将导致错误。无法多次启用带有不同流的模块。

## 备注

- 通过这个命令和 **%packages** 部分的组合，您可以安装由启用的模块和流组合提供的软件包，而无需明确指定模块和流。安装软件包前必须启用模块。使用 **module** 命令启用模块后，您可以通过在 **%packages** 部分列出此模块启用的软件包来安装它们。
- 单个 **module** 命令只能启用单个模块和流组合。要启用多个模块，请使用多个 **module** 命令。无法多次启用带有不同流的模块。
- 在 Red Hat Enterprise Linux 8 中，模块只存在于 AppStream 存储库中。要列出提供的模块，请在安装了具有有效订阅的 Red Hat Enterprise Linux 8 系统上使用 **yum module list** 命令。

## 其它资源

- [安装、管理和删除用户空间组件](#)

## B.3.8. repo

**repo** Kickstart 命令是可选的。它配置了作为软件包安装来源的额外的 yum 仓库。您可以添加多个 **repo** 行。

## 语法

```
repo --name=repo_id [--baseurl=url|--mirrorlist=url|--metalink=url] [OPTIONS]
```

## 必填选项

- **--name=** - 存储库 ID。这个选项是必需的。如果库的名称与另一个之前添加的库冲突，则会忽略它。因为安装程序使用预设置程序库列表，这意味着您无法添加名称与预先设置的库的名称相同的库。

## URL options

这些选项是互斥的，也是可选的。这里不支持 yum 库配置文件中可以使用的变量。您可以使用字符串 **\$releasever** 和 **\$basearch**，它们由 URL 中的对应值替换。

- **--baseurl=** - 存储库的 URL。
- **--mirrorlist=** - 指向存储库镜像列表的 URL。
- **--metalink=** - 存储库的 metalink 的 URL。

## 可选选项

- **--install** - 将已安装系统上的仓库配置保存在 **/etc/yum.repos.d/** 目录中。如果不使用这个选项，在 Kickstart 文件中配置的程序库将只在安装过程中使用，而无法在安装的系统中使用。
- **--cost=** - 为这个存储库分配成本的整数值。如果多个库提供同样的软件包，这个数字就会被用来决定优先使用哪个库。成本低的软件仓库优先于成本高的软件仓库。

- **--excludepkgs=** - 不能从此存储库拉取的软件包名称，是一个以逗号分隔的列表。如果多个存储库提供同样的软件包，您希望这个软件包来自特定的仓库，可以使用它。可接受完整软件包名称（如 **publican**）和 globs（如 **gnome-\***）。
- **--includepkgs=** - 允许从此存储库拉取的软件包名称和 glob 的逗号分隔列表。该程序仓库提供的其他软件包将被忽略。如果您只想从库中安装单个软件包或软件包组而不包括该程序库提供的所有其他软件包，这个选项就很有用了。
- **--proxy=[protocol://][username[:password]@]host[:port]** - 指定仅用于此存储库的 HTTP/HTTPS/FTP 代理。此设置不会影响任何其他存储库，也不会影响在 HTTP 安装中如何获取 **install.img**。
- **--noverifyssl** - 连接到 **HTTPS** 服务器时禁用 SSL 验证。

#### 备注

- 用于安装的软件仓库必须是稳定版本。如果在安装完成前修改库，则安装会失败。

### B.3.9. rootpw（必需）

需要 **rootpw** Kickstart 命令。它将系统的根密码设置为 *password* 参数的值。仅使用命令一次。

#### 语法

```
rootpw [--iscrypted|--plaintext] [--lock] password
```

#### 必填选项

- *password* - 密码规格。纯文本或者加密字符串。请参阅以下 **--iscrypted** 和 **--plaintext**。

#### 选项

- **--iscrypted** - 如果给出这个选项，则假设 *password* 参数已被加密。这个选项与 **--plaintext** 相互排斥。要创建一个加密的密码，您可以使用 python:

```
$ python -c 'import crypt,getpass;pw=getpass.getpass();print(crypt.crypt(pw) if (pw==getpass.getpass("Confirm: ")) else exit()'
```

这会使用随机 salt 为密码生成 sha512 兼容哈希。

- **--plaintext** - 如果给出这个选项，则假设 *password* 参数为纯文本。这个选项与 **--iscrypted** 相互排斥。
- **--lock** - 如果给出这个选项，则默认锁定 root 帐户。这意味着 root 用户无法从控制台登录。这个选项还在图形和文本手动安装中禁用 **Root 密码** 页面。

### B.3.10. selinux

**selinux** Kickstart 命令是可选的。它在安装的系统中设定 SELinux 状态。默认 SELinux 策略为 **enforcing**。仅使用命令一次。

#### 语法

```
selinux [--disabled|--enforcing|--permissive]
```

## 选项

### **--enforcing**

使用默认目标策略 **enforcing** 启用 SELinux。

### **--permissive**

根据 SELinux 策略输出警告，但并不强制执行该策略。

### **--disabled**

在系统上完全禁用 SELinux。

## 其它资源

- [使用 SELinux](#)

## B.3.11. services

**services** Kickstart 命令是可选的。它修改在默认 **systemd** 目标下运行的默认服务集合。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。

## 语法

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

## 选项

- **--disabled=** - 禁用在逗号分隔列表中给出的服务。
- **--enabled=** - 启用逗号分隔列表中给出的服务。

## 注

- 当使用 **services** 元素启用 **systemd** 服务时，请确保在 **%packages** 部分中包含指定服务文件的软件包。
- 使用逗号分开多个服务，没有空格。例如，要禁用四个服务，请输入：

```
services --disabled=auditd,cups,smartd,nfslock
```

如果您包含任何空格，Kickstart 只启用或禁用第一个空格前的服务。例如：

```
services --disabled=auditd, cups, smartd, nfslock
```

这仅禁用 **auditd** 服务。要禁用所有四个服务，此条目不得包含空格。

## B.3.12. skipx

**skipx** Kickstart 命令是可选的。如果存在，安装的系统上就不会配置 X。

如果您在软件包选择选项中安装 **display manager**，这个软件包会创建一个 X 配置，安装的系统会默认使用 **graphical.target**。这会覆盖 **skipx** 选项的影响。仅使用命令一次。

## 语法

skipx

#### 备注

- 这个命令没有选项。

### B.3.13. sshkey

**sshkey** Kickstart 命令是可选的。它将 SSH 密钥添加到已安装系统上指定用户的 **authorized\_keys** 文件中。

#### 语法

```
sshkey --username=user "ssh_key"
```

#### 必填选项

- **--username=** - 要安装密钥的用户。
- *ssh\_key* - 完整的 SSH 密钥指纹。它必须用引号括起。

### B.3.14. syspurpose

**syspurpose** Kickstart 命令是可选的。使用它来设置系统在安装后的系统目的。这些信息有助于在系统中应用正确的订阅授权。仅使用命令一次。



#### 注意

Red Hat Enterprise Linux 8.6 及更新的版本允许您使用 **subscription-manager** **syspurpose** 模块下提供的 **role**、**service-level**、**usage** 和 **addons** 子命令管理和显示具有单个模块的系统目的的属性。在以前的版本中，系统管理员使用四个独立的 **syspurpose** 命令来管理每个属性。从 RHEL 8.6 开始，这个独立的 **syspurpose** 命令被弃用，并计划在 RHEL 9 中删除。红帽将在当前发行生命周期中提供对这个功能的 bug 修复和支持，但此功能将不再获得改进。从 RHEL 9 开始，单个 **subscription-manager syspurpose** 命令及其相关的子命令是使用系统用途的唯一方法。

#### 语法

```
syspurpose [OPTIONS]
```

#### 选项

- **--role=** - 设置预期的系统角色。可用值有：
  - Red Hat Enterprise Linux Server
  - Red Hat Enterprise Linux Workstation
  - Red Hat Enterprise Linux Compute 节点
- **--SLA=** - 设置服务级别协议。可用值有：
  - Premium（高级）

- Standard (标准)
- Self-Support (自助)
- **--usage=** - 系统预定用途。可用值有：
  - Production
  - Disaster Recovery
  - Development/Test
- **--Addon=** - 指定额外的层次产品或功能。您可以多次使用这个选项。

#### 备注

- 输入有空格的值，并使用双引号包括它们：

```
syspurpose --role="Red Hat Enterprise Linux Server"
```

- 虽然强烈建议您配置系统目的，但它是 Red Hat Enterprise Linux 安装程序的可选功能。如果要在安装完成后启用系统目的，您可以使用 **syspurpose** 命令行工具完成此操作。



#### 注意

Red Hat Enterprise Linux 8.6 及更新的版本允许您使用 **subscription-manager syspurpose** 模块下提供的 **role**、**service-level**、**usage** 和 **addons** 子命令管理和显示具有单个模块的系统目的的属性。在以前的版本中，系统管理员使用四个独立的 **syspurpose** 命令来管理每个属性。从 RHEL 8.6 开始，这个独立的 **syspurpose** 命令被弃用，并计划在 RHEL 9 中删除。红帽将在当前发行生命周期中提供对这个功能的 bug 修复和支持，但此功能将不再获得改进。从 RHEL 9 开始，单个 **subscription-manager syspurpose** 命令及其相关的子命令是使用系统用途的唯一方法。

### B.3.15. timezone (必需)

**timezone** Kickstart 命令是必需的。它设置系统时区。仅使用命令一次。

#### 语法

```
timezone timezone [OPTIONS]
```

#### 必填选项

- *timezone* - 为系统设定的时区。

#### 可选选项

- **--UTC** - 如果存在，系统假定硬件时钟被设置为 UTC（格林威治 Mean）时间。
- **--nontp** - 禁用 NTP 服务自动启动。
- **--ntpserver=** - 指定用作没有空格的逗号分隔列表的 NTP 服务器列表。

#### 备注

在 Red Hat Enterprise Linux 8 中，时区名称使用 `pytz` 软件包提供的 `pytz.common_timezones` 列表进行验证。

### B.3.16. user

`user` Kickstart 命令是可选的。它在系统上创建新用户。

#### 语法

```
user --name=username [OPTIONS]
```

#### 必填选项

- **--name=** - 提供用户名称。这个选项是必需的。

#### 可选选项

- **--GECOS=** - 为用户提供 GECOS 信息。这个字符串包括使用逗号分开的各种具体系统字段。它通常用来指定用户全名、办公室号码等等。详情请查看 `passwd(5)` 手册页。
- **--groups=** - 除默认组外，还有以逗号分隔的用户应属于的组名列表。组群必须在创建该用户帐户前就已经存在。请参阅 `group` 命令。
- **--homedir=** - 用户的主目录。如果没有提供，则默认为 `/home/username`。
- **--lock** - 如果给出这个选项，则默认锁定这个帐户。这意味着用户无法从控制台登录。这个选项还在图形和文本手动安装中禁用 [创建用户](#) 页面。
- **--password=** - 新用户的密码。如果没有提供，则默认锁定该帐户。
- **--iscrypted** - 如果给出这个选项，则假设 `password` 参数已被加密。这个选项与 **--plaintext** 相互排斥。要创建一个加密的密码，您可以使用 python:

```
$ python -c 'import crypt,getpass;pw=getpass.getpass();print(crypt.crypt(pw) if
(pw==getpass.getpass("Confirm: ")) else exit()'
```

这会使用随机 salt 为密码生成 sha512 兼容哈希。

- **--plaintext** - 如果给出这个选项，则假设 `password` 参数为纯文本。这个选项与 **--iscrypted** 相互排斥
- **--shell=** - 用户的登录 shell。如果没有提供，则使用系统默认。
- **--uid=** - 用户的 UID（用户 ID）。如果没有提供，则默认使用下一个可用的非系统 UID。
- **--gid=** - 用于用户组的 GID（组 ID）。如果没有提供，则默认使用下一个可用的非系统组群 ID。

#### 备注

- 请考虑使用 **--uid** 和 **--gid** 选项设置常规用户及其默认组的 ID，范围从 **5000** 开始，而不是 **1000**。这是因为系统用户和组保留的范围（**0**到**999**）将来可能会增加，因此与常规用户的 ID 重叠。  
有关在安装后更改最小 UID 和 GID 限制，以确保您在创建用户时可以自动应用您选择的 UID 和 GID 范围，请参阅 [配置基本系统设置](#) 文档中的 [使用 umask 为新文件设置默认权限](#) 部分。

- 使用不同权限创建的文件和目录，由用来创建文件或目录的应用程序指定。例如，`mkdir` 命令创建启用了所有权限的目录。但是，应用无法为新创建的文件授予某些权限，如 `user file-creation mask` 设置所指定。  
`user file-creation mask` 可通过 `umask` 命令控制。新用户的 `user file-creation mask` 默认设置由安装系统上的 `/etc/login.defs` 配置文件中的 `UMASK` 变量定义。如果未设置，则默认为 `022`。这意味着，默认情况下，当应用程序创建一个文件时，会防止为该文件所有者以外的用户授予写入权限。不过，这可以被其他设置或脚本覆盖。

如需了解更多信息，请参阅 [配置基本系统设置](#) 文档中的 [使用 umask 配置基本系统设置](#)。

### B.3.17. xconfig

`xconfig` Kickstart 命令是可选的。它配置 X 窗口系统。仅使用命令一次。

#### 语法

```
xconfig [--startxonboot]
```

#### 选项

- `--startxonboot` - 在安装的系统上使用图形登录。

#### 备注

- 因为 Red Hat Enterprise Linux 8 没有包括 KDE 桌面环境，请不要使用上游中记录的 `--defaultdesktop=`。

## B.4. KICKSTART 命令用于网络配置

在这个列表中的 Kickstart 命令可让您在系统中配置联网。

### B.4.1. network（可选）

可选的 `network` Kickstart 命令配置目标系统的网络信息，并在安装环境中激活网络设备。第一个 `network` 命令中指定的设备会自动激活。您还可以使用 `--activate` 选项明确要求激活设备。

#### 语法

```
network OPTIONS
```

#### 选项

- `--activate` - 在安装环境中激活这个设备。  
如果您在已经激活的设备中使用 `--activate` 选项（例如，使用引导选项配置的界面以便系统可以检索 Kickstart 文件），则会重新激活该设备以使用 Kickstart 文件中指定的详情。

使用 `--nodefroute` 选项可防止设备使用默认路由。

- `--no-activate` - 不要在安装环境中激活这个设备。  
默认情况下，无论 `--activate` 选项是什么，Anaconda 都会激活 Kickstart 文件中的第一个网络设备。您可以使用 `--no-activate` 选项禁用默认设置。

- **--bootproto=** - **dhcp**、**bootp**、**ibft** 或 **static** 之一。默认选项为 **dhcp**；**dhcp** 和 **bootp** 选项的处理方式相同。要禁用设备的 **ipv4** 配置，可使用 **--noipv4** 选项。



### 注意

这个选项配置设备的 **ipv4** 配置。对于 **ipv6** 配置，请使用 **--ipv6** 和 **--ipv6gateway** 选项。

DHCP 方法使用 DHCP 服务器系统来获得它的网络配置。BOOTP 方法类似，需要 BOOTP 服务器来提供网络配置。要指示系统使用 DHCP：

```
network --bootproto=dhcp
```

要指示机器使用 BOOTP 获取其网络配置，在 Kickstart 文件中使用以下行：

```
network --bootproto=bootp
```

要指示机器使用 iBFT 中指定的配置，使用：

```
network --bootproto=ibft
```

**static** 方法要求您在 Kickstart 文件中至少指定 IP 地址和子网掩码。这个信息是静态的，并在安装过程中和安装后使用。

所有静态网络配置信息必须在 **一行**中指定；您不能象在命令行中换行一样使用反斜杠(\)来换行。

```
network --bootproto=static --ip=10.0.2.15 --netmask=255.255.255.0 --gateway=10.0.2.254 --nameserver=10.0.2.1
```

您还可以同时配置多个名称服务器。要做到这一点，使用 **--nameserver=** 选项一次，并指定每个 IP 地址，用逗号分开：

```
network --bootproto=static --ip=10.0.2.15 --netmask=255.255.255.0 --gateway=10.0.2.254 --nameserver=192.168.2.1,192.168.3.1
```

- **--device=** - 使用 **network** 命令指定要配置的设备（最终在 Anaconda 中激活）。如果在第一次使用 **network** 命令时缺少 **--device=** 选项，则使用 **inst.ks.device=** Anaconda 引导选项的值（如果提供的话）。这被视为已弃用的行为；在大多数情况下，您应该始终为每个 **network** 命令指定 **a- device=**。

如果缺少其 **--device=** 选项，则同一 Kickstart 文件中后续 **network** 命令的行为都会被取消指定。验证您是否为第一个以外的任何一个 **network** 命令指定这个选项。

您可以使用以下任一方法指定要激活的设备：

- 接口的设备名称，如 **em1**
- 接口的 MAC 地址，例如 **01:23:45:67:89:ab**
- 关键字 **link**，它指定链接为 **up** 状态的第一个接口
- 关键字 **bootif**，它使用 **pxelinux** 在 **BOOTIF** 变量中设置的 MAC 地址。在 **pxelinux.cfg** 文件中设置 **IPAPPEND 2**，使 **pxelinux** 设置 **BOOTIF** 变量。

例如：

```
network --bootproto=dhcp --device=em1
```

- **--ipv4-dns-search/--ipv6-dns-search** - 手动设置 DNS 搜索域。您必须将这些选项与 **--device** 选项一起使用,并镜像其相应的 NetworkManager 属性, 例如：

```
network --device ens3 --ipv4-dns-search domain1.example.com,domain2.example.com
```

- **--ipv4-ignore-auto-dns/--ipv6-ignore-auto-dns** - 设置其来忽略 DHCP 中的 DNS 设置。您必须将这些选项与 **--device** 选项一起使用, 这些选项不需要任何参数。
- **--ip=** - 设备的 IP 地址。
- **--ipv6=** - 设备的 IPv6 地址, 格式为 *address[/prefix length]* - 例如, **3ffe:ffff:0:1::1/128**。如果省略了 *prefix*, 则使用 **64**。您还可以使用 **auto** 进行自动配置, 或使用 **dhcp** 仅进行 DHCPv6 配置 (无路由器广告)。
- **--gateway=** - 作为单一 IPv4 地址的默认网关。
- **--ipv6gateway=** - 作为单一 IPv6 地址的默认网关。
- **--nodefroute** - 防止被设置为默认路由的接口。当您激活使用 **--activate=** 选项的其他设备时, 请使用这个选项, 例如: iSCSI 目标的单独子网中的 NIC。
- **--nameserver=** - DNS 名称服务器作为 IP 地址。要指定一个以上名称服务器, 使用这个选项, 并使用逗号分隔每个 IP 地址。
- **--netmask=** - 安装系统的网络掩码。
- **--hostname=** - 用于配置目标系统的主机名。主机名可以是完全限定域名 (FQDN), 格式为 **hostname.domainname**, 也可以是不包括域的短主机名。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 服务为这台机器分配域名, 请只指定简短主机名。  
使用静态 IP 和主机名配置时, 它取决于计划的系统用例是否使用短名称或 FQDN。红帽身份管理在置备过程中配置 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**。

**localhost** 值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在处理网络配置的过程中配置, 例如, 通过使用 DHCP 或 DNS 的 NetworkManager。

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, 并且 FQDN 总长度 (包括点) 不应超过 255 个字符。

如果您只想配置目标系统的主机名, 请在 **network** 命令中使用 **--hostname** 选项, 且不包含任何其他选项。

如果您在配置主机名时提供附加选项, **network** 命令将使用指定的选项来配置设备。如果您没有使用 **--device** 选项指定要配置的设备, 则使用默认的 **--device link** 值。另外, 如果您不使用 **--bootproto** 选项指定协议, 则该设备会被配置为默认使用 DHCP。

- **--ethtool=** - 指定将传递给 ethtool 程序的网络设备的其他低级别设置。
- **--onboot=** - 是否在引导时启用该设备。

- **--dhcpclass=** - DHCP 类。
- **--mtu=** - 设备的 MTU。
- **--noipv4** - 在这个设备上禁用 IPv4。
- **--noipv6** - 在这个设备上禁用 IPv6。
- **--bondslaves=** - 使用这个选项时，由 **--device=** 选项指定的绑定设备会使用 **--bondslaves=** 选项中定义的辅助设备创建。例如：

```
network --device=bond0 --bondslaves=em1,em2
```

以上命令创建了一个名为 **bond0** 的绑定设备，将 **em1** 和 **em2** 接口用作其辅助设备。

- **--bondopts=** - 绑定接口的可选参数列表，使用 **--bondslaves=** 和 **--device=** 选项指定。这个列表中的选项必须以逗号(",")或分号(";")分开。如果某个选项本身包含一个逗号，请使用分号来分隔选项。例如：

```
network --bondopts=mode=active-backup,balance-rr;primary=eth1
```



### 重要

**--bondopts=mode=** 参数只支持完整的模式名称，如 **balance-rr** 或 **broadcast**，而不是其数字表示，如 **0** 或 **3**。有关可用和支持的模式列表，请参阅 [配置和管理网络指南](#)。

- **--vlanid=** - 指定使用 **--device=** 中指定的设备作为父级创建的设备的虚拟 LAN (VLAN) ID 号 (802.1q 标签)。例如，**network --device=em1 --vlanid=171** 创建虚拟 LAN 设备 **em1.171**。
- **--interfaceName=** - 为虚拟 LAN 设备指定自定义接口名称。当 **--vlanid=** 选项所生成的默认名称并不是您所需要的名称时，应使用此选项。此选项必须与 **--vlanid=** 一起使用。例如：

```
network --device=em1 --vlanid=171 --interfacename=vlan171
```

以上命令在 **em1** 设备上创建一个名为 **vlan171** 的虚拟 LAN 接口，其 ID 为 **171**。

接口名称可以是任意名称（如 **my-vlan**），但在某些情况下，必须遵循以下约定：

- 如果名称包含句点(.)，则必须采用 **NAME.ID** 的形式。**NAME** 是任意的，但 **ID** 必须是 VLAN ID。例如：**em1.171** 或 **my-vlan.171**。
- 以 **vlan** 开头的名称必须使用 **vlanID** 的形式，如 **vlan171**。
- **--teamslaves=** - 由 **--device=** 选项指定的团队设备将使用这个选项中指定的辅助设备创建。辅助设备用逗号分开。辅助设备可以跟随其配置，这是单引号括起的 JSON 字符串，其中双引号用 **\** 字符转义。例如：

```
network --teamslaves="p3p1{'prio': -10, 'sticky': true},p3p2{'prio': 100}"
```

另请参阅 **--teamconfig=** 选项。

- **--teamconfig=** - 由双引号括起的组设备配置，这是一个 JSON 字符串，其中双引号由 **\** 字符转义。设备名称由 **--device=** 选项指定，其辅助设备及其配置由 **--teamslaves=** 选项指定。例如：

```
network --device team0 --activate --bootproto static --ip=10.34.102.222 --
netmask=255.255.255.0 --gateway=10.34.102.254 --nameserver=10.34.39.2 --
teamslaves="p3p1{'prio': -10, 'sticky': true},p3p2{'prio': 100}" --teamconfig="
{'runner': {'name': 'activebackup'}}"
```

- **--bridgeslaves=** - 使用此选项时，将创建 **--device=** 选项指定的设备网桥，并将 **--bridgeslaves=** 选项中定义的设备添加到网桥中。例如：

```
network --device=bridge0 --bridgeslaves=em1
```

- **--bridgeopts=** - 一个可选的、由逗号分隔的桥接接口参数列表。可用值包括 **stp**、**priority**、**forward-delay**、**hello-time**、**max-age** 和 **RunAsAny-time**。有关这些参数的详情，请查看 **nm-settings(5)** 手册页中的 *bridge setting* 表，或者查看 [网络配置设置规范](#)。有关网络桥接的常规信息，请参阅 [配置和管理网络](#) 文档。
- **--bindto=mac** - 在安装的系统上将设备配置文件绑定到设备 MAC 地址(**HWADDR**)，而不是默认绑定到接口名称(**DEVICE**)。这个选项独立于 the- **device=** 选项 **--bindto=mac**，即使同一 **network** 命令也指定了设备名称、[链接](#) 或 **bootif**。

## 注

- 由于命名方案的变化，**ethN** 设备名称（如 **eth0**）在 Red Hat Enterprise Linux 中不再可用。有关设备命名方案的更多信息，请参阅上游文档 [可扩展网络接口名称](#)。
- 如果您使用 Kickstart 选项或者引导选项指定网络中的安装程序库，但安装过程开始时无法使用网络，安装程序会在显示 [安装概述](#) 窗口前，显示 [网络配置](#) 窗口以用于设置网络连接。如需了解更多信息，请参阅 [配置网络和主机名选项](#)。

## B.4.2. realm

**realm** Kickstart 命令是可选的。使用它加入 Active Directory 或 IPA 域。有关此命令的更多信息，请参阅 **realm(8)man** page 的 **join** 部分。

## 语法

```
realm join [OPTIONS] domain
```

## 必填选项

- **domain** - 要加入的域。

## 选项

- **--computer-ou=OU=** - 提供可分辨的机构单元名称以便创建计算机帐户。可识别名称的具体格式取决于客户端软件和成员软件。可省略可识别名称的根 DSE 部分。
- **--no-password** - 无需密码自动加入。
- **--one-time-password=** - 使用一次性密码加入。不是所有域都支持它。
- **--client-software=** - 仅加入能够运行此客户端软件的域。有效值包括 **sssd** 和 **winbind**。不是所有域都支持所有值。默认情况下自动选择客户端软件。

- **--server-software=** - 仅加入能够运行此服务器软件的域。可能的值包括 **active-directory** 或 **freeipa**。
- **--membership-software=** - 加入域时使用此软件。有效值包括 **samba** 和 **adcli**。不是所有域都支持所有值。默认情况下自动选择成员软件。

## B.5. 用于处理存储的 KICKSTART 命令

这部分中的 Kickstart 命令配置存储的各个方面，比如设备、磁盘、分区、LVM 和文件系统。

### 重要

**sdX**（或 **/dev/sdX**）格式不能保证重启后一致的设备名称，这可能会使某些 Kickstart 命令的使用变得复杂。当命令需要设备节点名称时，您可以使用 **/dev/disk** 中的任何项目作为替代。例如，改为使用以下设备名称：

```
part / --fstype=trfs --onpart=sda1
```

您可以使用类似以下条目之一：

```
part / --fstype=trfs --onpart=/dev/disk/by-path/pci-0000:00:05.0-scsi-0:0:0:0-part1
```

```
part / --fstype=trfs --onpart=/dev/disk/by-id/ata-ST3160815AS_6RA0C882-part1
```

通过使用这种方法，命令总是以相同的存储设备为目标。这在大型存储环境中特别有用。要利用系统上的可用设备名称，您可以在交互式安装过程中使用 **ls -lR /dev/disk** 命令。有关统一引用存储设备的方法的更多信息，请参阅 [持久性命名属性的概述](#)。

### B.5.1. device（已弃用）

**device** Kickstart 命令是可选的。使用它载入其他内核模块。

在大多数 PCI 系统中，安装程序会自动探测以太网卡和 SCSI 卡。然而，在老的系统和某些 PCI 系统中，Kickstart 需要提示才能找到正确的设备。**device** 命令告知安装程序安装额外模块，使用以下格式：

#### 语法

```
device moduleName --opts=options
```

#### 选项

- **moduleName** - 使用应该安装的内核模块的名称替换。
- **--opts=** - 传递给内核模块的选项。例如：

```
device --opts="aic152x=0x340 io=11"
```

### B.5.2. autopart

**autopart** Kickstart 命令是可选的。它自动创建分区。

自动创建的分区是：根(/)分区（1 GiB 或更大）、**swap** 分区，以及适合架构的 **/boot** 分区。在足够大的驱动器（50 GiB 及更高版本）上，这还会创建一个 **/home** 分区。仅使用命令一次。

## 语法

```
autopart OPTIONS
```

## 选项

- **--type=** - 选择您要使用的预定义自动分区方案之一。可接受以下值：
  - **lvm** : LVM 分区方案。
  - **plain** : 没有 LVM 的常规分区。
  - **thinp** : LVM Thin Provisioning 分区方案。
- **--fstype=** - 选择其中一个可用文件系统类型。可用值包括 **ext2**、**ext3**、**ext4**、**xfs** 和 **vfat**。默认的文件系统是 **xfs**。
- **--nohome** - 禁用自动创建 **/home** 分区。
- **--nolvm** - 不使用 LVM 进行自动分区。这个选项等同于 **--type=plain**。
- **--noboot** - 不创建 **/boot** 分区。
- **--noswap** - 不创建交换分区。
- **--encrypted** - 使用 Linux 统一密钥设置 (LUKS) 加密所有分区。这等同于在手动图形安装的初始分区界面中选择**加密分区**选项。

**注意**

在加密一个或多个分区时，Anaconda 会尝试收集 256 字节熵，以保证安全加密分区。收集熵可能需要一些时间 - 无论是否有收集到足够的熵，该过程将在最多 10 分钟后停止。

与安装系统互动（通过键盘输入或移动鼠标）可加速此进程。如果要在虚拟机中安装，您还可以将 **virtio-rng** 设备（虚拟随机数生成器）附加到客户机。

- **--LUKS-version=LUKS\_VERSION** - 指定应该使用哪个版本的 LUKS 格式来加密文件系统。只有在指定了 **--encrypted** 时这个选项才有意义。
- **--passphrase=** - 为所有加密设备提供默认的系统范围密码短语。
- **--escrowcert=URL\_of\_X.509\_certificate** - 将所有加密卷的数据加密密钥保存在 **/root** 的文件中，使用来自 **URL\_of\_X.509\_certificate** 指定的 URL 的 X.509 证书进行加密。每个加密卷的密钥都作为单独的文件保存。只有在指定了 **--encrypted** 时这个选项才有意义。
- **--backuppssphrase** - 为每个加密卷添加随机生成的密码短语。将这些密码短语存储在 **/root** 中的单独文件中，使用通过 **--escrowcert** 指定的 X.509 证书进行加密。只有在指定了 **--escrowcert** 时这个选项才有意义。
- **--cipher=** - 指定在 Anaconda 默认 **aes-xts-plain64** 时要使用的加密类型。这个选项必须与 **--encrypted** 选项一同使用；其本身无效。[安全强化](#) 文档中列出了可用的加密类型，但红帽强烈建议您使用 **aes-xts-plain64** 或 **aes-cbc-essiv:sha256**。
- **--pbkdf=PBKDF** - 为 LUKS keylot 设置 PBKDF（Password-Based Key Derivation Function）算法。另请参阅 man page *cryptsetup(8)*。只有在指定了 **--encrypted** 时这个选项才有意义。

- **--PBKDF-memory=PBKDF\_MEMORY** - 设置 PBKDF 的内存成本。另请参阅 man page `cryptsetup(8)`。只有在指定了 **--encrypted** 时这个选项才有意义。
- **--PBKDF-time=PBKDF\_TIME** - 设置 PBKDF 密码处理所花费的毫秒数。另请参阅 man page `cryptsetup(8)` 中的 **--iter-time**。只有在指定了 **--encrypted** 时这个选项才有意义，并且与 **--pbkdf-iterations** 相互排斥。
- **--PBKDF-iterations=PBKDF\_ITERATIONS** - 设定直接迭代数量并避免 PBKDF 基准测试。另请参阅 man page `cryptsetup(8)` 中的 **--pbkdf-force-iterations**。只有在指定了 **--encrypted** 时这个选项才有意义，并且与 **--pbkdf-time** 相互排斥。

## 备注

- **autopart** 选项不能与同一 Kickstart 文件中的 **part/partition**、**raid**、**logvol** 或 **volgroup** 选项一同使用。
- **autopart** 命令不是强制的，但如果 Kickstart 脚本中没有 **part** 或 **mount** 命令，则必须包含该命令。
- 在 CMS 类型的单个 FBA DASD 中安装时，建议使用 **autopart --nohome** Kickstart 选项。这样可以保证安装程序不会创建单独的 **/home** 分区。安装过程可以成功进行。
- 如果您丢失了 LUKS 密码短语，那么就完全无法访问所有加密的分区及其数据。丢失的密码短语是无法找回的。但是，您可以使用 **--escrowcert** 保存加密密码短语，并使用 **--backuppssphrase** 选项创建加密密码短语备份。
- 在使用 **autopart**、**autopart --type=lvm** 或 **autopart=thinp** 时，确保磁盘扇区大小一致。

## B.5.3. bootloader (必需)

**bootloader** Kickstart 命令是必需的。它指定引导装载程序的安装方式。仅使用命令一次。

### 语法

```
bootloader [OPTIONS]
```

### 选项

- **--append=** - 指定附加内核参数。要指定多个参数，使用空格分隔它们。例如：

```
bootloader --location=mbr --append="hdd=ide-scsi ide=nodma"
```

安装 **plymouth** 软件包时会自动添加 **rhgb** 和 **quiet** 参数，即使您在此未指定参数或根本不使用 **--append=** 命令。要禁用此行为，请明确禁止安装 **plymouth**：

```
%packages
-plymouth
%end
```

这个选项可用于禁用在大多数现代处理器中存在的 Meltdown 和 Spectre speculative 安全漏洞 (CVE-2017-5754、CVE-2017-5753 和 CVE-2017-5715)。在某些情况下，这些机制可能并不是必需的，启用它们可能会导致性能降低而不会提高安全性。要禁用这些系统，请在您的 Kickstart 文件中添加相关选项，例如：**bootloader --append="nopti noibrs noibpb"** (在 AMD64/Intel 64 系统中)。



### 警告

在禁用任何漏洞缓解机制前，请确定您的系统不会受到安全攻击。有关 Meltdown 和 Spectre 漏洞的详情，请查看[红帽漏洞响应文章](#)。

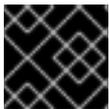
- **--boot-drive=** - 指定引导装载程序应写入的驱动器，因此要从哪个驱动器引导计算机。如果您使用多路径设备作为引导驱动器，使用它的 `disk/by-id/dm-uuid-mpath-WWID` 名称指定该设备。



### 重要

目前，在使用 **zipl** 引导装载程序的 64 位 IBM Z 系统上的 Red Hat Enterprise Linux 安装中会忽略 **--boot-drive=** 选项。安装 **zipl** 后，它会自行确定引导驱动器。

- **--leavebootorder** - 安装程序会将 Red Hat Enterprise Linux 8 添加到引导装载程序中安装的系统列表的顶部，并保留所有现有的条目及其顺序。



### 重要

这个选项只适用于 Power 系统，UEFI 系统不应该使用这个选项。

- **--driveorder=** - 指定哪个驱动器最先在 BIOS 引导顺序中。例如：

```
bootloader --driveorder=sda,hda
```

- **--location=** - 指定引导记录的写入位置。有效值如下：
  - **mbr** - 默认选项。具体要看驱动器是使用主引导记录（MBR）还是 GUID 分区表（GPT）方案：
    - 在 GPT 格式的磁盘中，这个选项会在 BIOS 引导分区中安装 stage 1.5 引导装载程序。
    - 在使用 MBR 格式化的磁盘中，会在 MBR 和第一个分区之间的空白空间中安装 stage 1.5。
  - **partition** - 在包含内核的分区的第一个扇区安装引导装载程序。
  - **none** - 不安装引导装载程序。

在大多数情况下，不需要指定这个选项。

- **--nombr** - 不在 MBR 中安装引导加载器。
- **--password=** - 如果使用 GRUB2，则将引导装载程序密码设置为使用这个选项指定的密码。这应该被用来限制对可传入任意内核选项的 GRUB2 shell 的访问。如果指定密码，GRUB2 还会询问用户名。用户名始终为 **root**。
- **--iscrypted** - 通常当您使用 **--password=** 选项指定引导装载程序密码时，会以明文形式将其保存在 Kickstart 文件中。如果要加密密码，使用这个选项和一个加密的密码。要生成加密的密码，请使用 **grub2-mkpasswd-pbkdf2** 命令，输入要使用的密码，并将命令的输出（以 **grub.pbkdf2** 开头的哈希值）复制到 Kickstart 文件中。带有加密密码的 **bootloader** Kickstart 条目示例类似如下：

```
bootloader --iscrypted --
password=grub.pbkdf2.sha512.10000.5520C6C9832F3AC3D149AC0B24BE69E2D4FB0DBE
EDBD29CA1D30A044DE2645C4C7A291E585D4DC43F8A4D82479F8B95CA4BA4381F8550
510B75E8E0BB2938990.C688B6F0EF935701FF9BD1A8EC7FE5BD2333799C98F28420C5
CC8F1A2A233DE22C83705BB614EA17F3FDFDF4AC2161CEA3384E56EB38A2E39102F53
34C47405E
```

- **--timeout=** - 指定引导装载程序在引导默认选项前等待的时间（以秒为单位）。
- **--default=** - 在引导装载程序配置中设置默认引导镜像。
- **--extlinux** - 使用 extlinux 引导装载程序而不是 GRUB2。这个选项只适用于支持 extlinux 的系统。
- **--disabled** - 这个选项是更强大的 **--location=none** 版本。虽然 **--location=none** 只是禁用引导装载程序安装，但 **--disabled** 禁用引导装载程序安装，同时禁用包含引导装载程序的软件包安装，从而节省了空间。

## 备注

- 红帽建议在每个系统中设置引导装载程序密码。一个没有保护的引导装载程序可以让潜在的攻击者修改系统的引导选项，并获得对系统的未授权访问。
- 在某些情况下，需要一个特殊的分区来在 AMD64、Intel 64 和 64 位 ARM 系统上安装引导装载程序。这个分区的类型和大小取决于您要安装引导装载程序的磁盘是否使用主引导记录（MBR）还是 GUID 分区表（GPT）模式。如需更多信息，请参阅 [配置引导装载程序](#) 部分。
- **sdX**（或 **/dev/sdX**）格式不能保证重启后一致的设备名称，这可能会使某些 Kickstart 命令的使用变得复杂。当命令需要设备节点名称时，您可以使用 **/dev/disk** 中的任何项目作为替代。例如，改为使用以下设备名称：

```
part / --fstype=xfs --onpart=sda1
```

您可以使用类似以下条目之一：

```
part / --fstype=xfs --onpart=/dev/disk/by-path/pci-0000:00:05.0-scsi-0:0:0:0-part1
```

```
part / --fstype=xfs --onpart=/dev/disk/by-id/ata-ST3160815AS_6RA0C882-part1
```

通过使用这种方法，命令总是以相同的存储设备为目标。这在大型存储环境中特别有用。要利用系统上的可用设备名称，您可以在交互式安装过程中使用 **ls -lR /dev/disk** 命令。有关统一引用存储设备的方法的更多信息，请参阅 [持久性命名属性的概述](#)。

- 在 Red Hat Enterprise Linux 8 中已弃用 **--upgrade** 选项。

## B.5.4. zipl

**zipl** Kickstart 命令是可选的。它为 64 位 IBM Z 指定 ZIPL 配置。只使用这个选项。

### 选项

- **--secure-boot** - 如果安装系统支持，则启用安全引导。

**注意**

当在 IBM z14 之后的系统中安装时，无法从 IBM z14 或更早的型号引导安装的系统。

- **--force-secure-boot** - 无条件启用安全引导。

**注意**

IBM z14 及更早的型号不支持安装。

- **--no-secure-boot** - 禁用安全引导。

**注意**

IBM z14 及更早的型号不支持安全引导。如果要在 IBM z14 及更早的型号中引导安装的系统，请使用 **--no-secure-boot**。

### B.5.5. clearpart

**clearpart** Kickstart 命令是可选的。在创建新分区之前，它会从系统中删除分区。默认情况下不会删除任何分区。仅使用命令一次。

#### 语法

```
clearpart OPTIONS
```

#### 选项

- **--all** - 断掉系统中的所有分区。  
这个选项将擦除安装程序可以访问的所有磁盘，包括任何附加的网络存储。请小心使用这个选项。

为了防止您需要保留的存储会被 **clearpart** 清除，您可以使用 **--drives=** 选项指定需要删除的存储，或在以后附加网络存储（例如：在 Kickstart 文件的 **%post** 部分），或将用来访问网络存储的内核模块列入阻塞名单。

- **--drives=** - 指定从中清除分区的驱动器。例如，下面的命令清除了主 IDE 控制器上前两个驱动器上所有分区：

```
clearpart --drives=hda,hdb --all
```

要清除多路径设备，请使用格式 **disk/by-id/scsi-*WWID***，其中 *WWID* 是该设备的通用识别符。例如：要清除 WWID **58095BEC5510947BE8C0360F604351918** 的磁盘，请使用：

```
clearpart --drives=disk/by-id/scsi-58095BEC5510947BE8C0360F604351918
```

这个格式是所有多路径设备的首选，但如果出现错误，没有使用逻辑卷管理（LVM）的多路径设备也可以使用 **disk/by-id/dm-uuid-mpath-*WWID*** 格式进行清除，其中 *WWID* 是该设备的通用识别符。例如，若要清除 WWID 为 **2416CD96995134CA5D787F00A5AA11017** 的磁盘，请使用：

```
clearpart --drives=disk/by-id/dm-uuid-mpath-2416CD96995134CA5D787F00A5AA11017
```

切勿按设备名称（如 `mpatha`）指定多路径设备。这样的设备名称并不是特定磁盘特有的。在安装过程中名为 `/dev/mpatha` 的磁盘可能不是您期望的磁盘。因此，`clearpart` 命令可能会以错误的磁盘为目标。

- `--initlabel` - 通过为所有磁盘在其对于的架构中创建一个默认的磁盘标签来格式化磁盘（例如：对于 x86，使用 `msdos`）。因为 `--initlabel` 可以查看所有磁盘，因此请确保只连接要格式化的驱动器。`clearpart` 清理的磁盘将创建标签，即使 `--initlabel` 没有使用。

```
clearpart --initlabel --drives=names_of_disks
```

例如：

```
clearpart --initlabel --drives=dasda,dasdb,dasdc
```

- `--list=` - 指定要清除哪些分区。如果使用此选项，这个选项将覆盖 `--all` 和 `--linux` 选项。可在不同的驱动器间使用。例如：

```
clearpart --list=sda2,sda3,sdb1
```

- `--disklabel=LABEL` - 设置要使用的默认 `disklabel`。只有支持该平台的磁盘标签才会被接受。例如，在 64 位 Intel 和 AMD 构架中，接受 `msdos` 和 `gpt` `disklabels`，但不接受 `dasd`。
- `--linux` - 删除所有 Linux 分区。
- `--none`（默认）- 不删除任何分区。
- `--cdl` - 将所有 LDL DASD 重新格式化为 CDL 格式。

## 注

- `sdX`（或 `/dev/sdX`）格式不能保证重启后一致的设备名称，这可能会使某些 Kickstart 命令的使用变得复杂。当命令需要设备节点名称时，您可以使用 `/dev/disk` 中的任何项目作为替代。例如，改为使用以下设备名称：

```
part / --fstype=xfs --onpart=sda1
```

您可以使用类似以下条目之一：

```
part / --fstype=xfs --onpart=/dev/disk/by-path/pci-0000:00:05.0-scsi-0:0:0:0-part1
```

```
part / --fstype=xfs --onpart=/dev/disk/by-id/ata-ST3160815AS_6RA0C882-part1
```

通过使用这种方法，命令总是以相同的存储设备为目标。这在大型存储环境中特别有用。要利用系统上的可用设备名称，您可以在交互式安装过程中使用 `ls -lR /dev/disk` 命令。有关统一引用存储设备不同方法的更多信息，请参阅 [持久性命名属性的概述](#)。

- 如果使用 `clearpart` 命令，则无法在逻辑分区中使用 `part --onpart` 命令。

## B.5.6. fcoe

`fcoe` Kickstart 命令是可选的。它指定除了由 EDD（Enhanced Disk Drive Services）发现的设备外，还要自动激活哪些 FCoE 设备。

## 语法

```
fcoe --nic=name [OPTIONS]
```

## 选项

- **--nic=** (必需) - 要激活的设备的名称。
- **--dcb=** - 建立数据中心桥接 (DCB) 设置。
- **--autovlan** - 自动发现 VLAN。默认启用这个选项。

## B.5.7. ignoredisk

**ignoredisk** Kickstart 命令是可选的。这会导致安装程序忽略指定的磁盘。

如果您使用自动分区并希望忽略某些磁盘，这就很有用。例如，如果没有 **ignoredisk**，尝试在 SAN-cluster 中部署，Kickstart 将失败，因为安装程序检测到到 SAN 的被动路径没有分区表。仅使用命令一次。

## 语法

```
ignoredisk --drives=drive1,drive2,... | --only-use=drive
```

## 选项

- **--drives=*driveN,...*** - 使用 **sda**、**sdb**、...、**hda**、... 等之一替换 *driveN*。
- **--only-use=*driveN,...*** - 指定安装程序要使用的磁盘列表。其它磁盘将被忽略。例如：要在安装过程中使用磁盘 **da** 并忽略所有其他磁盘：

```
ignoredisk --only-use=sda
```

要包括不使用 LVM 的多路径设备：

```
ignoredisk --only-use=disk/by-id/dm-uuid-mpath-2416CD96995134CA5D787F00A5AA11017
```

要包括使用 LVM 的多路径设备：

```
ignoredisk --only-use==/dev/disk/by-id/dm-uuid-mpath-
```

```
bootloader --location=mbr
```

您必须仅指定 **--drives** 或 **--only-use** 中的一个。

## 备注

- 在 Red Hat Enterprise Linux 8 中已弃用 **--interactive** 选项。这个选项允许用户手动在高级存储界面中进行操作。
- 要忽略不使用逻辑卷管理 (LVM) 的多路径设备，使用格式 **disk/by-id/dm-uuid-mpath-*WWID***，其中 *WWID* 是该设备的通用识别符。例如，要忽略 WWID 为 **2416CD96995134CA5D787F00A5AA11017** 的磁盘，请使用：

```
ignoredisk --drives=disk/by-id/dm-uuid-mpath-2416CD96995134CA5D787F00A5AA11017
```

- 切勿按设备名称（如 **mpatha**）指定多路径设备。这样的设备名称并不是特定磁盘特有的。在安装过程中名为 **/dev/mpatha** 的磁盘可能不是您期望的磁盘。因此，**clearpart** 命令可能会以错误的磁盘为目标。
- **sdX**（或 **/dev/sdX**）格式不能保证重启后一致的设备名称，这可能会使某些 Kickstart 命令的使用变得复杂。当命令需要设备节点名称时，您可以使用 **/dev/disk** 中的任何项目作为替代。例如，改为使用以下设备名称：

```
part / --fstype=xfs --onpart=sda1
```

您可以使用类似以下条目之一：

```
part / --fstype=xfs --onpart=/dev/disk/by-path/pci-0000:00:05.0-scsi-0:0:0-part1
```

```
part / --fstype=xfs --onpart=/dev/disk/by-id/ata-ST3160815AS_6RA0C882-part1
```

通过使用这种方法，命令总是以相同的存储设备为目标。这在大型存储环境中特别有用。要利用系统上的可用设备名称，您可以在交互式安装过程中使用 **ls -lR /dev/disk** 命令。有关统一引用存储设备的方法的更多信息，请参阅 [持久性命名属性的概述](#)。

### B.5.8. iscsi

**iscsi** Kickstart 命令是可选的。它指定了在安装过程中要添加的附加 iSCSI 存储。

#### 语法

```
iscsi --ipaddr=address [OPTIONS]
```

#### 必填选项

- **--ipaddr=**（必需）- 要连接的目标的 IP 地址。

#### 可选选项

- **--port=**（必需）- 端口号。如果没有，则默认自动使用 **--port=3260**。
- **--target=** - 目标 IQN（iSCSI 限定名称）。
- **--iface=** - 将连接绑定到特定网络接口，而不使用由网络层决定的默认接口。一旦使用，必须在整个 Kickstart 文件的 **iscsi** 命令的所有实例中指定它。
- **--user=** - 与目标进行身份验证所需的用户名
- **--password=** - 与为目标指定的用户名对应的密码
- **--reverse-user=** - 从使用反向 CHAP 身份验证的目标向启动器进行身份验证所需的用户名
- **--reverse-password=** - 与为启动器指定的用户名对应的密码

#### 备注

- 如果使用 **iscsi** 命令，还必须使用 **iscsiname** 命令为 iSCSI 节点分配名称。在 Kickstart 文件中，**iscsiname** 命令需要在 **iscsi** 命令的前面。
- 尽可能在系统 BIOS 或固件（Intel 系统的 iBFT）中配置 iSCSI 存储，而不是使用 **iscsi** 命令。Anaconda 自动检测并使用在 BIOS 或固件中配置的磁盘，且在 Kickstart 文件中不需要特殊配置。
- 如果您必须使用 **iscsi** 命令，请确保在安装开始时激活联网，并且在使用 **clearpart** 或 **ignoredisk** 等命令引用 iSCSI 磁盘 *前*，**iscsi** 命令会出现在 Kickstart 文件中。

### B.5.9. iscsiname

**iscsiname** Kickstart 命令是可选的。它为 **iscsi** 命令指定的 iSCSI 节点分配名称。仅使用命令一次。

#### 语法

```
iscsiname iqname
```

#### 选项

- **iqname** - 分配给 iSCSI 节点的名称。

#### 备注

- 如果在 Kickstart 文件中使用 **iscsi** 命令，则必须在 Kickstart 文件 *前面* 指定 **iscsiname**。

### B.5.10. logvol

**logvol** Kickstart 命令是可选的。它为逻辑卷管理（LVM）创建一个逻辑卷。

#### 语法

```
logvol mountpoint --vgname=name --name=name [OPTIONS]
```

#### 必填选项

##### **mountpoint**

挂载分区的挂载点。必须是以下格式之一：

- **/path**  
例如：/ 或 /home
- **swap**  
该分区被用作交换空间。

要自动决定 swap 分区的大小，请使用 **--recommended** 选项：

```
swap --recommended
```

要自动决定 swap 分区的大小，并允许系统的额外空间可以休眠，请使用 **--hibernation** 选项：

```
swap --hibernation
```

分配的大小将相当于由 **--recommended** 分配的交换空间加上您系统上的 RAM 量。有关这些命令分配的 swap 大小，请参阅 AMD64、Intel 64 和 64 位 ARM 系统 [推荐的分区方案](#)。

### **--vgname=*name***

卷组名称。

### **--name=*name***

逻辑卷名称。

## 可选选项

### **--noformat**

使用现有逻辑卷且不要对其进行格式化。

### **--useexisting**

使用现有逻辑卷并重新格式化它。

### **--fstype=**

为逻辑卷设置文件系统类型。有效值为 **xfs**、**ext2**、**ext3**、**ext4**、**swap** 和 **vfat**。

### **--fsoptions=**

指定挂载文件系统时要使用的选项的自由格式字符串。该字符串将复制到安装的系统的 **/etc/fstab** 文件中，并且应用引号括起来。



### 注意

在 EFI 系统分区(**/boot/efi**)中，anaconda 硬编码了值，并忽略了用户指定的 **--fsoptions** 值。

### **--mkfsoptions=**

指定要传递给在这个分区上创建文件系统的程序的其他参数。没有对参数列表进行任何操作，因此必须以可直接传递给 **mkfs** 程序的格式提供。这意味着，根据具体文件系统，多个选项应该用逗号分开，或使用双引号分开。例如，

```
part /opt/foo1 --size=512 --fstype=ext4 --mkfsoptions="-O
^has_journal,^flex_bg,^metadata_csum"

part /opt/foo2 --size=512 --fstype=xfs --mkfsoptions="-m bigtime=0,finobt=0"
```

详情请查看您要创建的文件系统的手册页。例如：**mkfs.ext4** 或 **mkfs.xfs**。

### **--fsprofile=**

指定传递给在这个分区上创建文件系统的程序的使用类型。使用类型定义了创建文件系统时使用的各种微调参数。要使用这个选项，文件系统必须支持使用类型，且必须有一个配置文件来列出有效类型。对于 **ext2**、**ext3** 和 **ext4**，此配置文件为 **/etc/mke2fs.conf**。

### **--label=**

为逻辑卷设置标签。

### **--grow**

扩展逻辑卷以占据可用空间（若有），或使用指定的最大值（若有）。只有在磁盘镜像中预分配了最小存储空间时，必须使用该选项，并希望该卷扩展并占据所有可用空间。在物理环境中，这是一次性的操作。但是在虚拟环境中，当虚拟机将任何数据写入虚拟磁盘时，卷大小会增加。

**--size=**

MiB 中的逻辑卷的大小。此选项不能与 **--percent=** 选项一同使用。

**--percent=**

考虑任何静态大小逻辑卷时的逻辑卷大小，作为卷组中剩余空间的百分比。这个选项不能与 **--size=** 选项一同使用。

**重要**

在创建新逻辑卷时，您必须使用 **--size=** 选项静态指定其大小，或使用 **--percent=** 选项指定剩余空间的百分比。您不能在同一逻辑卷中同时使用这些选项。

**--maxsize=**

当将逻辑卷设置为可扩充时 MiB 的最大值。在这里指定一个整数值，如 **500**（不要包含单位）。

**--recommended**

创建逻辑卷时使用这个选项，根据您的系统硬件自动决定这个卷的大小。有关推荐方案的详情，请参考 AMD64、Intel 64 和 64 位 ARM 系统的[建议分区方案](#)。

**--resize**

重新调整逻辑卷大小。如果使用这个选项，还必须指定 **--useexisting** 和 **--size**。

**--encrypted**

指定该逻辑卷应该使用 Linux Unified Key Setup(LUKS)进行加密，使用 **--passphrase=** 选项中提供的密码短语。如果您没有指定密码短语，安装程序将使用 **autopart --passphrase** 命令设定的默认系统级密码，或者在未设定默认密码时暂停安装并提示您输入密码短语。

**注意**

在加密一个或多个分区时，Anaconda 会尝试收集 256 字节熵，以保证安全加密分区。收集熵可能需要一些时间 - 无论是否有收集到足够的熵，该过程将在最多 10 分钟后停止。

与安装系统互动（通过键盘输入或移动鼠标）可加速此进程。如果要在虚拟机中安装，您还可以将 **virtio-rng** 设备（虚拟随机数生成器）附加到客户机。

**--passphrase=**

指定在加密这个逻辑卷时要使用的密码短语。这个选项必须与 **--encrypted** 选项一同使用，单独使用无效。

**--cipher=**

如果 Anaconda 默认的 **aes-xts-plain64** 不符合要求，则指定要使用的加密类型。这个选项必须与 **--encrypted** 选项一同使用；其本身无效。[安全强化](#) 文档中列出了可用的加密类型，但红帽强烈建议您使用 **aes-xts-plain64** 或 **aes-cbc-essiv:sha256**。

**--escrowcert=URL\_of\_X.509\_certificate**

将所有加密卷的数据加密密钥保存为 **/root** 中的文件，使用来自 **URL\_of\_X.509\_certificate** 指定的 URL 的 X.509 证书进行加密。每个加密卷的密钥都作为单独的文件保存。只有在指定了 **--encrypted** 时这个选项才有意义。

**--luks-version=LUKS\_VERSION**

指定应用来加密文件系统的 LUKS 格式版本。只有在指定了 **--encrypted** 时这个选项才有意义。

**--backuppssphrase**

为每个加密卷添加随机生成的密码短语。将这些密码短语存储在 `/root` 中的单独文件中，使用通过 `--escrowcert` 指定的 X.509 证书进行加密。只有在指定了 `--escrowcert` 时这个选项才有意义。

#### `--pbkdf=PBKDF`

为 LUKS keylot 设置基于密码的身份验证(PBKDF)算法。另请参阅 man page `cryptsetup(8)`。只有在指定了 `--encrypted` 时这个选项才有意义。

#### `--pbkdf-memory=PBKDF_MEMORY`

为 PBKDF 设置内存成本。另请参阅 man page `cryptsetup(8)`。只有在指定了 `--encrypted` 时这个选项才有意义。

#### `--pbkdf-time=PBKDF_TIME`

设置 PBKDF 密码处理所消耗的毫秒数。另请参阅 man page `cryptsetup(8)` 中的 `--iter-time`。只有在指定了 `--encrypted` 时这个选项才有意义，并且与 `--pbkdf-iterations` 相互排斥。

#### `--pbkdf-iterations=PBKDF_ITERATIONS`

设定直接迭代的次数，并避免 PBKDF 基准出现。另请参阅 man page `cryptsetup(8)` 中的 `--pbkdf-force-iterations`。只有在指定了 `--encrypted` 时这个选项才有意义，并且与 `--pbkdf-time` 相互排斥。

#### `--thinpool`

创建精简池逻辑卷。（使用 `none` 挂载点）

#### `--metadatasize=size`

指定新精简池设备的元数据大小（单位 MiB）。

#### `--chunksize=size`

为新的精简池设备指定块大小（单位 KiB）。

#### `--thin`

创建精简逻辑卷。（需要使用 `--poolname`）

#### `--poolname=name`

指定要在其中创建精简逻辑卷的精简池名称。需要 `--thin` 选项。

#### `--profile=name`

指定与精简逻辑卷搭配使用的配置配置集的名称。如果使用，该名称也会包含在给定逻辑卷的元数据中。默认情况下，可用的配置集为 `default` 和 `thin-performance`，在 `/etc/lvm/profile/` 目录中定义。有关其他信息，请参见 `lvm(8)` 手册页。

#### `--cachepvs=`

用逗号分开的物理卷列表，它应用作这个卷的缓存。

#### `--cachemode=`

指定应该使用哪种模式缓存这个逻辑卷 - `writeback` 或 `writethrough`。



#### 注意

有关缓存的逻辑卷及其模式的详情，请参考 `lvmcache(7)` 手册页。

#### `--cachesize=`

附加到逻辑卷的缓存大小，以 MiB 为单位。此选项需要 `--cachepvs=` 选项。

#### 备注

- 使用 Kickstart 安装 Red Hat Enterprise Linux 时，请勿在逻辑卷和卷组名称中使用短划线 (-) 字符。如果使用这个字符，安装会完成，但 `/dev/mapper/` 目录会列出这些卷和卷组的每个横线都会加倍。例如，名为 `volgrp-01` 的卷组，其包含名为 `logvol-01` 的逻辑卷将被列为 `/dev/mapper/volgrp-01-logvol-01`。

这个限制只适用于新创建的逻辑卷和卷组名称。如果您使用 **--noformat** 选项重复使用现有的名称，则不会更改它们的名称。

- 如果您丢失了 LUKS 密码短语，那么就完全无法访问所有加密的分区及其数据。丢失的密码短语是无法找回的。但是，您可以使用 **--escrowcert** 保存加密密码短语，并使用 **--backupphrase** 选项创建加密密码短语备份。

## 示例

- 首先创建分区，然后创建逻辑卷组，然后创建逻辑卷：

```
part pv.01 --size 3000
volgroup myvg pv.01
logvol / --vgname=myvg --size=2000 --name=rootvol
```

- 首先创建分区，然后创建逻辑卷组，再创建逻辑卷以占据卷组中剩余的 90% 空间：

```
part pv.01 --size 1 --grow
volgroup myvg pv.01
logvol / --vgname=myvg --name=rootvol --percent=90
```

## 其它资源

- [配置和管理逻辑卷](#)

## B.5.11. mount

**mount** Kickstart 命令是可选的。它为现有块设备分配挂载点，并选择性地将其重新格式化为给定格式。

### 语法

```
mount [OPTIONS] device mountpoint
```

### 必须的选项

- **device** - 要挂载的块设备。
- **mountpoint** - 挂载 **device** 的位置。它必须是有效的挂载点，如 **/** 或 **/usr**，如果设备不可卸载，则为 **none**（如 **swap**）。

### 可选选项：

- **--reformat=** - 指定应重新格式化该设备的新格式（如 **ext4**）。
- **--mkfsoptions=** - 指定要传递给 **mkfs** 命令的附加选项，该命令将创建 **--reformat=** 中指定的新文件系统。这里提供的选项列表没有被处理，因此必须使用可直接传递给 **mkfs** 程序的格式指定。根据具体文件系统，选项列表应该用逗号分开，或使用双引号分开。有关具体详情，请参阅 **mkfs** man page（例如 **mkfs.ext4(8)** 或 **mkfs.xfs(8)**）。
- **--mountoptions=** - 指定自由格式字符串，其中包含挂载文件系统时要使用的选项。字符串将复制到安装的系统上的 **/etc/fstab** 文件，并使用双引号括起来。有关挂载选项的完整列表，请参见 **mount(8)** 手册页，有关基础知识，请参见 **fstab(5)**。

## 备注

- 与 Kickstart 中的大多数其他存储配置命令不同，**mount** 不需要您在 Kickstart 文件中描述整个存储配置。您只需要确定系统中存在描述的块设备。但是，如果要使用挂载的所有设备 *创建* 存储堆栈，则必须使用其他命令，比如 **part** 要这样做。
- 在同一个 Kickstart 文件中，您不能将 **mount** 与其他存储相关的命令，如 **part**、**logvol** 或 **autopart** 一起使用。

## B.5.12. nvdimm

**nvdimm** Kickstart 命令是可选的。它对非线性内存模块(NVDIMM)设备执行操作。默认情况下，安装程序会忽略 NVDIMM 设备。您必须使用 **nvdimm** 命令在这些设备中启用安装。

### 语法

```
nvdimm action [OPTIONS]
```

### 操作

- **reconfigure** - 将特定 NVDIMM 设备重新配置为给定模式。另外，指定的设备被隐式标记为要使用的，因此后续的另一设备 **nvdimm use** 命令就冗余了。这个操作使用以下格式：

```
nvdimm reconfigure [--namespace=NAMESPACE] [--mode=MODE] [--sectorsize=SECTORSIZE]
```

- **--namespace=** - 按命名空间的设备规格。例如：

```
nvdimm reconfigure --namespace=namespace0.0 --mode=sector --sectorsize=512
```

- **--mode=** - 模式规格。目前，只有值 **sector** 可用。

- **--sectorsize=** - 扇区模式的扇区大小。例如：

```
nvdimm reconfigure --namespace=namespace0.0 --mode=sector --sectorsize=512
```

支持的扇区大小为 512 和 4096 字节。

- **use** - 将 NVDIMM 设备指定为安装目标。该设备必须已通过 **nvdimm reconfigure** 命令配置为扇区模式。这个操作使用以下格式：

```
nvdimm use [--namespace=NAMESPACE]--blockdevs=DEVICES]
```

- **--namespace=** - 按命名空间指定设备。例如：

```
nvdimm use --namespace=namespace0.0
```

- **--blockdevs=** - 指定与要使用的 NVDIMM 设备对应的块设备列表。支持星号 \* 通配符。例如：

```
nvdimm use --blockdevs=pmem0s,pmem1s
nvdimm use --blockdevs=pmem*
```

### B.5.13. part 或 partition

需要 **part** 或 **partition** Kickstart 命令。它在系统上创建一个分区。

#### 语法

```
part|partition mntpoint [OPTIONS]
```

#### 选项

- **mntpoint** - 挂载分区的位置。该值必须是以下格式之一：

- **/path**

例如：**/**、**/usr**、**/home**

- **swap**

该分区被用作交换空间。

要自动决定 swap 分区的大小，请使用 **--recommended** 选项：

```
swap --recommended
```

分配的大小将生效，但不会根据您的系统进行精确校准。

要自动确定 swap 分区的大小，同时允许系统的额外空间可以休眠，请使用 **--hibernation** 选项：

```
swap --hibernation
```

分配的大小将相当于由 **--recommended** 分配的交换空间加上您系统上的 RAM 量。有关这些命令分配的 swap 大小，请参阅 AMD64、Intel 64 和 64 位 ARM 系统 [推荐的分区方案](#)。

- **raid.id**

该分区用于软件 RAID（请参阅 **raid**）。

- **pv.id**

该分区用于 LVM（请参阅 **logvol**）。

- **biosboot**

该分区将用在 BIOS 引导分区中。在使用 BIOS 的 AMD64 和 Intel 64 系统中使用 GUID 分区表(GPT)中需要 1 MiB BIOS 引导分区；引导装载程序将被安装到其中。UEFI 系统中不需要此功能。另请参阅 **bootloader** 命令。

- **/boot/efi**

一个 EFI 系统分区。基于 UEFI 的 AMD64、Intel 64 和 64 位 ARM 上需要 50 MiB EFI 分区。推荐的大小为 200 MiB。它在 BIOS 系统上并不需要。另请参阅 **bootloader** 命令。

- **--size=** - 最小分区大小，以 MiB 为单位。在这里指定一个整数值，如 **500**（不要包含单位）。如果指定大小太小，安装会失败。将 **--size** 值设置为您需要的最小空间量。有关大小的建议，请参阅 [推荐的分区方案](#)。
- **--grow** - 指定分区增大以填充所有可用空间（若有），或使用设置的最大值（如果指定了最大值）。如果您在交换分区上使用 **--grow=** 但没有设置 **--maxsize=**，Anaconda 会限制 swap 分区的最大大小。对于物理内存小于 2 GiB 的系统，强制的限制为物理内存的两倍。对于拥有超过 2 GiB 的系统，强制限制为物理内存大小加上 2GiB。

- **--maxsize=** - 分区设置为增长时的最大分区大小（以 MiB 为单位）。在这里指定一个整数值，如 500（不要包含单位）。
- **--noformat** - 指定不要被格式化的分区，以便与 **--onpart** 命令一起使用。
- **--onpart=** 或 **--usepart=** - 指定要放置分区的设备。使用现有的空白设备并将其格式化到新指定类型。例如：

```
partition /home --onpart=hda1
```

将 **/home** 放置到 **/dev/hda1**。

这些选项还可以在逻辑卷中添加分区。例如：

```
partition pv.1 --onpart=hda2
```

设备必须已在系统上存在；**--onpart** 选项不会创建它。

在这种情况下，Anaconda 会在不创建分区表的情况下格式化并使用驱动器。但是，以这种方式格式化的设备上不支持 GRUB2 安装，且必须将其放置在带有分区表的驱动器中。

```
partition pv.1 --onpart=hdb
```

- **--ondisk=** 或 **--ondrive=** - 在现有磁盘创建分区（由 **part** 命令指定）。这个命令总是创建一个分区。强制在特定磁盘创建分区。例如：**--ondisk=sdb** 将分区放在系统的第二个 SCSI 磁盘中。要指定不使用逻辑卷管理（LVM）的多路径设备，使用格式 **disk/by-id/dm-uuid-mpath-WWID**，其中 **WWID** 是该设备的通用识别符。例如，要指定 WWID 为 **2416CD96995134CA5D787F00A5AA11017** 的磁盘，请使用：

```
part / --fstype=xfst --grow --asprimary --size=8192 --ondisk=disk/by-id/dm-uuid-mpath-2416CD96995134CA5D787F00A5AA11017
```



### 警告

切勿按设备名称（如 **mpatha**）指定多路径设备。这样的设备名称并不是特定磁盘特有的。在安装过程中名为 **/dev/mpatha** 的磁盘可能不是您期望的磁盘。因此，**part** 命令可能会以错误的磁盘为目标。

- **--asprimary** - 强制将该分区分配为 *primary* 分区。如果无法将该分区作为主分区分配（通常是因为已经分配了太多的主分区），则该分区进程会失败。只有使用主引导记录（MBR）时这个选项才有意义。
- **--fsprofile=** - 指定要传递给在这个分区上创建文件系统的程序的使用类型。使用类型定义了创建文件系统时使用的各种微调参数。要使用这个选项，文件系统必须支持使用类型，且必须有一个配置文件来列出有效类型。对于 **ext2**、**ext3**、**ext4**，此配置文件为 **/etc/mke2fs.conf**。
- **--mkfsoptions=** - 指定要传递给在此分区上创建文件系统的程序的其他参数。这与 **--fsprofile** 类似，但适用于所有文件系统，而不仅仅是支持配置集概念的文件系统。没有对参数列表进行任何

操作，因此必须以可直接传递给 `mkfs` 程序的格式提供。这意味着，根据具体文件系统，多个选项应该用逗号分开，或使用双引号分开。例如，

```
part /opt/foo1 --size=512 --fstype=ext4 --mkfsoptions="-O
^has_journal,^flex_bg,^metadata_csum"

part /opt/foo2 --size=512 --fstype=xfst --mkfsoptions="-m bigtime=0,finobt=0"
```

详情请查看您要创建的文件系统的手册页。例如：`mkfs.ext4` 或 `mkfs.xfst`。

- **--fstype=** - 为分区设置文件系统类型。有效值为 `xfst`、`ext2`、`ext3`、`ext4`、`swap`、`vfst`、`efi` 和 `biosboot`。
- **--fsoptions** - 指定在挂载文件系统时要使用的自由格式选项字符串。该字符串将复制到安装的系统 `/etc/fstab` 文件中，并且应用引号括起来。



### 注意

在 EFI 系统分区 (`/boot/efi`) 中，`anaconda` 硬编码了值，并忽略了用户指定的 **--fsoptions** 值。

- **--label=** - 为单个分区分配标签。
- **--recommended** - 自动确定分区的大小。有关推荐方案的详情，请参考 AMD64、Intel 64 和 64 位 ARM [推荐的分区方案](#)。这个选项只能用于生成文件系统（如 `/boot` 分区和 `交换空间`）的分区。它不能被用来创建 LVM 物理卷或 RAID 成员。
- **--onbiosdisk** - 强制 BIOS 发现的特定磁盘上创建的分区。
- **--encrypted** - 指定该分区应该用 LUKS 加密，使用 **--passphrase=** 选项中提供的密码短语。如果您没有指定密码短语，`Anaconda` 将使用 `autopart --passphrase` 命令设置的默认系统级密码，或者停止安装并在未设定默认密码短语时提示您输入密码短语。



### 注意

在加密一个或多个分区时，`Anaconda` 会尝试收集 256 字节熵，以保证安全加密分区。收集熵可能需要一些时间 - 无论是否有收集到足够的熵，该过程将在最多 10 分钟后停止。

与安装系统互动（通过键盘输入或移动鼠标）可加速此进程。如果要在虚拟机中安装，您还可以将 `virtio-rng` 设备（虚拟随机数生成器）附加到客户机。

- **--LUKS-version=LUKS\_VERSION** - 指定应该使用哪个版本的 LUKS 格式来加密文件系统。只有在指定了 **--encrypted** 时这个选项才有意义。
- **--passphrase=** - 指定在加密此分区时要使用的密码短语。这个选项必须与 **--encrypted** 选项一同使用；其本身无效。
- **--cipher=** - 指定在 `Anaconda` 默认 `aes-xts-plain64` 时要使用的加密类型。这个选项必须与 **--encrypted** 选项一同使用；其本身无效。[安全强化](#) 文档中列出了可用的加密类型，但红帽强烈建议您使用 `aes-xts-plain64` 或 `aes-cbc-essiv:sha256`。
- **--escrowcert=URL\_of\_X.509\_certificate** - 将所有加密分区的数据加密密钥保存在 `/root` 的文件中，使用来自 `URL_of_X.509_certificate` 指定的 URL 的 X.509 证书进行加密。每个加密分区的密钥都作为单独的文件保存。只有在指定了 **--encrypted** 时这个选项才有意义。

- **--backupperphrase** - 为每个加密分区添加随机生成的密码短语。将这些密码短语存储在 `/root` 中的单独文件中，使用通过 **--escrowcert** 指定的 X.509 证书进行加密。只有在指定了 **--escrowcert** 时这个选项才有意义。
- **--pbkdf=PBKDF** - 为 LUKS keylot 设置 PBKDF (Password-Based Key Derivation Function) 算法。另请参阅 man page `cryptsetup(8)`。只有在指定了 **--encrypted** 时这个选项才有意义。
- **--PBKDF-memory=PBKDF\_MEMORY** - 设置 PBKDF 的内存成本。另请参阅 man page `cryptsetup(8)`。只有在指定了 **--encrypted** 时这个选项才有意义。
- **--PBKDF-time=PBKDF\_TIME** - 设置 PBKDF 密码处理所花费的毫秒数。另请参阅 man page `cryptsetup(8)` 中的 **--iter-time**。只有在指定了 **--encrypted** 时这个选项才有意义，并且与 **--pbkdf-iterations** 相互排斥。
- **--PBKDF-iterations=PBKDF\_ITERATIONS** - 设定直接迭代数量并避免 PBKDF 基准测试。另请参阅 man page `cryptsetup(8)` 中的 **--pbkdf-force-iterations**。只有在指定了 **--encrypted** 时这个选项才有意义，并且与 **--pbkdf-time** 相互排斥。
- **--resize=** - 调整现有分区的大小。使用这个选项时，使用 **--size=** 选项指定目标的大小（单位 MiB），使用 **--onpart=** 指定目标分区。

## 备注

- **part** 命令不是强制的，但您必须在 Kickstart 脚本中包含 **part**、**autopart** 或 **mount**。
- 在 Red Hat Enterprise Linux 8 中已弃用 **--active** 选项。
- 如果因为某种原因分区失败，虚拟控制台 3 中会显示诊断信息。
- 除非使用 **--noformat** 和 **--onpart**，否则所有创建的分区的都会格式化为安装过程的一部分。
- **sdX**（或 `/dev/sdX`）格式不能保证重启后一致的设备名称，这可能会使某些 Kickstart 命令的使用变得复杂。当命令需要设备节点名称时，您可以使用 `/dev/disk` 中的任何项目作为替代。例如，改为使用以下设备名称：

```
part / --fstype=xfs --onpart=sda1
```

您可以使用类似以下条目之一：

```
part / --fstype=xfs --onpart=/dev/disk/by-path/pci-0000:00:05.0-scsi-0:0:0:0-part1
```

```
part / --fstype=xfs --onpart=/dev/disk/by-id/ata-ST3160815AS_6RA0C882-part1
```

通过使用这种方法，命令总是以相同的存储设备为目标。这在大型存储环境中特别有用。要利用系统上的可用设备名称，您可以在交互式安装过程中使用 **ls -lR /dev/disk** 命令。有关统一引用存储设备的方法的更多信息，请参阅 [持久性命名属性的概述](#)。

- 如果您丢失了 LUKS 密码短语，那么就完全无法访问所有加密的分区及其数据。丢失的密码短语是无法找回的。但是，您可以使用 **--escrowcert** 保存加密密码短语，并使用 **--backupperphrase** 选项创建加密密码短语备份。

## B.5.14. raid

**raid** Kickstart 命令是可选的。它组成一个软件 RAID 设备。

### 语法

```
raid mntpoint --level=level --device=device-name partitions*
```

## 选项

- `mntpoint` - 挂载 RAID 文件系统的位置。如果是 `/`，RAID 级别必须是 1，除非引导分区 (`/boot`) 存在。如果引导分区存在，`/boot` 分区必须是级别 1，`root (/)` 分区可以是任意可用的类型。`partitions*`（代表多个分区可以被列举）列出了要添加到 RAID 阵列的 RAID 标记。



### 重要

- 在 IBM Power 系统上，如果 RAID 设备已经准备好，且在安装过程中没有重新格式化，如果您打算将 `/boot` 和 PReP 分区放在 RAID 设备上，请确保 RAID 元数据版本为 **0.90** 或 **1.0**。对于 `/boot` 和 PReP 分区 `mdadm` 元数据版本 **1.1** 和 **1.2** 不支持。
- PowerNV 系统上不需要 **PReP** 引导分区。

- `--level=` - 使用的 RAID 级别（0、1、4、5、6 或 10）。
- `--device=` - 要使用的 RAID 设备的名称 - 例如 `--device=root`。



### 重要

不要使用 `md0` 格式的 `mdraid` 名称 - 无法保证这些名称具有持久性。相反，应使用有意义的名称，如 `root` 或 `swap`。使用有意义的名称可生成一个从 `/dev/md/name` 到 `/dev/mdX` 节点分配到的阵列的符号链接。

如果您有一个无法为其分配名称的旧阵列（v0.90 元数据），您可以使用文件系统标签或 UUID 指定该阵列。例如，`--device=LABEL=root` 或 `--device=UUID=93348e56-4631-d0f0-6f5b-45c47f570b88`。

您可以使用 RAID 设备上文件系统的 UUID 或者 RAID 设备本身的 UUID。RAID 设备的 UUID 应为 **8-4-4-4-12** 格式。mdadm 报告的 UUID 为 **8:8:8:8** 格式，需要更改。例如 `93348e56:4631d0f0:6f5b45c4:7f570b88` 应改为 `93348e56-4631-d0f0-6f5b-45c47f570b88`。

- `--CHUNKSIZE=` - 以 KiB 为单位设置 RAID 存储的块大小。在某些情况下，使用与默认块不同的块大小 (**512 Kib**) 可以提高 RAID 的性能。
- `--spares=` - 指定分配给 RAID 阵列的备用驱动器数量。可使用备用驱动器在驱动器失败时重建阵列。
- `--fsprofile=` - 指定要传递给在这个分区上创建文件系统的程序的使用类型。使用类型定义了创建文件系统时使用的各种微调参数。要使用这个选项，文件系统必须支持使用类型，且必须有一个配置文件来列出有效类型。对于 `ext2`、`ext3` 和 `ext4`，此配置文件为 `/etc/mke2fs.conf`。
- `--fstype=` - 为 RAID 阵列设置文件系统类型。有效值为 `xfs`、`ext2`、`ext3`、`ext4`、`swap` 和 `vfat`。
- `--fsoptions=` - 指定在挂载文件系统时要使用的自由格式选项字符串。该字符串将复制到安装的系统 `/etc/fstab` 文件中，并且应用引号括起来。在 EFI 系统分区 (`/boot/efi`) 中，anaconda 硬编码了值，并忽略了用户指定的 `--fsoptions` 值。

- **--mkfsoptions=** - 指定要传递给在此分区上创建文件系统的程序的其他参数。没有对参数列表进行任何操作，因此必须以可直接传递给 mkfs 程序的格式提供。这意味着，根据具体文件系统，多个选项应该用逗号分开，或使用双引号分开。例如，

```
part /opt/foo1 --size=512 --fstype=ext4 --mkfsoptions="-O
^has_journal,^flex_bg,^metadata_csum"

part /opt/foo2 --size=512 --fstype=xfs --mkfsoptions="-m bigtime=0,finobt=0"
```

详情请查看您要创建的文件系统的手册页。例如：**mkfs.ext4** 或 **mkfs.xfs**。

- **--label=** - 指定要生成的文件系统的标签。如果给定标签已被另一个文件系统使用，则会创建一个新标签。
- **--noformat** - 使用现有的 RAID 设备，且不要格式化 RAID 阵列。
- **--use existing** - 使用现有的 RAID 设备并重新格式化它。
- **--encrypted** - 指定该 RAID 设备应该用 LUKS 加密，使用 **--passphrase=** 选项中提供的密码短语。如果您没有指定密码短语，Anaconda 将使用 **autopart --passphrase** 命令设定的默认系统级密码，或者停止安装并在未设定默认密码短语时提示您输入密码短语。



### 注意

在加密一个或多个分区时，Anaconda 会尝试收集 256 字节熵，以保证安全加密分区。收集熵可能需要一些时间 - 无论是否有收集到足够的熵，该过程将在最多 10 分钟后停止。

与安装系统互动（通过键盘输入或移动鼠标）可加速此进程。如果要在虚拟机中安装，您还可以将 **virtio-rng** 设备（虚拟随机数生成器）附加到客户机。

- **--LUKS-version=LUKS\_VERSION** - 指定应该使用哪个版本的 LUKS 格式来加密文件系统。只有在指定了 **--encrypted** 时这个选项才有意义。
- **--cipher=** - 指定在 Anaconda 默认 **aes-xts-plain64** 时要使用的加密类型。这个选项必须与 **--encrypted** 选项一同使用；其本身无效。[安全强化](#) 文档中列出了可用的加密类型，但红帽强烈建议您使用 **aes-xts-plain64** 或 **aes-cbc-essiv:sha256**。
- **--passphrase=** - 指定加密此 RAID 设备时使用的密码短语。这个选项必须与 **--encrypted** 选项一同使用；其本身无效。
- **--escrowcert=URL\_of\_X.509\_certificate** - 将这个设备的数据加密密钥保存在 **/root** 中的一个文件中，使用来自 **URL\_of\_X.509\_certificate** 指定的 URL 的 X.509 证书加密。只有在指定了 **--encrypted** 时这个选项才有意义。
- **--backuppassphrase** - 向这个设备添加随机生成的密码短语。将该密码存储在 **/root** 中的文件中，使用通过 **--escrowcert** 指定的 X.509 证书进行加密。只有在指定了 **--escrowcert** 时这个选项才有意义。
- **--pbkdf=PBKDF** - 为 LUKS keylot 设置 PBKDF（Password-Based Key Derivation Function）算法。另请参阅 man page *cryptsetup(8)*。只有在指定了 **--encrypted** 时这个选项才有意义。
- **--PBKDF-memory=PBKDF\_MEMORY** - 设置 PBKDF 的内存成本。另请参阅 man page *cryptsetup(8)*。只有在指定了 **--encrypted** 时这个选项才有意义。

- **--PBKDF-time=PBKDF\_TIME** - 设置 PBKDF 密码处理所花费的毫秒数。另请参阅 man page *cryptsetup(8)* 中的 **--iter-time**。只有在指定了 **--encrypted** 时这个选项才有意义，并且与 **--pbkdf-iterations** 相互排斥。
- **--PBKDF-iterations=PBKDF\_ITERATIONS** - 设定直接迭代数量并避免 PBKDF 基准测试。另请参阅 man page *cryptsetup(8)* 中的 **--pbkdf-force-iterations**。只有在指定了 **--encrypted** 时这个选项才有意义，并且与 **--pbkdf-time** 相互排斥。

## 示例

下面的例子展示，假定系统里有三个 SCSI 磁，如何为 `/` 创建 RAID 1 分区，为 `/home` 创建 RAID 5。它还创建三个交换分区，每个驱动器都有一个。

```
part raid.01 --size=6000 --ondisk=sda
part raid.02 --size=6000 --ondisk=sdb
part raid.03 --size=6000 --ondisk=sdc
part swap --size=512 --ondisk=sda
part swap --size=512 --ondisk=sdb
part swap --size=512 --ondisk=sdc
part raid.11 --size=1 --grow --ondisk=sda
part raid.12 --size=1 --grow --ondisk=sdb
part raid.13 --size=1 --grow --ondisk=sdc
raid / --level=1 --device=rhel8-root --label=rhel8-root raid.01 raid.02 raid.03
raid /home --level=5 --device=rhel8-home --label=rhel8-home raid.11 raid.12 raid.13
```

## 备注

- 如果您丢失了 LUKS 密码短语，那么就完全无法访问所有加密的分区及其数据。丢失的密码短语是无法找回的。但是，您可以使用 **--escrowcert** 保存加密密码短语，并使用 **--backupphrase** 选项创建加密密码短语备份。

## B.5.15. reqpart

**reqpart** Kickstart 命令是可选的。它自动创建您的硬件平台所需的分区。这为带有 UEFI 固件的系统包括一个 `/boot/efi` 分区，带有 BIOS 固件和 GPT 的系统包括 **biosboot** 分区，为 IBM Power 系统包括 **PRePBoot** 分区。仅使用命令一次。

## 语法

```
reqpart [--add-boot]
```

## 选项

- **--add-boot** - 除基础命令创建的特定于平台的分区之外，还创建一个单独的 `/boot` 分区。

## 备注

- 这个命令不能与 **autopart** 命令一同使用，因为 **autopart** 会做所有 **reqpart** 命令要做的工作，另外，还创建其他分区或者逻辑卷，比如 `/` 和 **swap**。与 **autopart** 不同，这个命令只创建特定于平台的分区，并将驱动器的其余部分留空，允许您创建自定义布局。

## B.5.16. snapshot

**snapshot** Kickstart 命令是可选的。在安装过程中使用 LVM 精简卷快照来创建 LVM 精简卷快照。这让您在安装前或安装后备份逻辑卷。

要创建多个快照，请多次添加 **snaphost** Kickstart 命令。

## 语法

```
snapshot vg_name/lv_name --name=snapshot_name --when=pre-install/post-install
```

## 选项

- **vg\_name/lv\_name** - 设置卷组和逻辑卷的名称，以便从中创建快照。
- **--name=snapshot\_name** - 设置快照的名称。这个名称在卷组中必须是唯一的。
- **--when=pre-install/post-install** - 如果在安装开始前或安装完成后创建快照，则设置。

## B.5.17. volgroup

**volgroup** Kickstart 命令是可选的。它创建一个逻辑卷管理 (LVM) 组。

## 语法

```
volgroup name [OPTIONS] [partition*]
```

## 必填选项

- **name** - 新卷组的名称。

## 选项

- **partition** - 用作卷组存储支持的物理卷分区。
- **--noformat** - 使用现有卷组而不对其进行格式化。
- **--useexisting** - 使用现有卷组并重新格式化它。如果使用这个选项，不要指定分区。例如：

```
volgroup rhel00 --useexisting --noformat
```

- **--pesize=** - 以 KiB 为单位设置卷组物理扩展的大小。默认值为 4096 (4 MiB)，最小值为 1024 (1 MiB)。
- **--reserved-space=** - 指定在 MiB 的卷组中保留未使用的空间量。只适用于新创建的卷组。
- **--reserved-percent=** - 指定卷组空间占未使用空间的百分比。只适用于新创建的卷组。

## 备注

- 首先创建分区，然后创建逻辑卷组，然后创建逻辑卷。例如：

```
part pv.01 --size 10000
volgroup my_volgrp pv.01
logvol / --vgname=my_volgrp --size=2000 --name=root
```

- 使用 Kickstart 安装 Red Hat Enterprise Linux 时，请勿在逻辑卷和卷组名称中使用短划线 (-) 字符。如果使用这个字符，安装会完成，但 `/dev/mapper/` 目录会列出这些卷和卷组的每个横线都会加倍。例如，名为 `volgrp-01` 的卷组，其包含名为 `logvol-01` 的逻辑卷将被列为 `/dev/mapper/volgrp--01-logvol--01`。  
这个限制只适用于新创建的逻辑卷和卷组名称。如果您使用 `--noformat` 选项重复使用现有的名称，则不会更改它们的名称。

### B.5.18. zerombr

**zerombr** Kickstart 命令是可选的。**zerombr** 会初始化磁盘上找到的所有无效分区表，并销毁具有无效分区表的磁盘的所有内容。当在带有未格式化的 Direct Access Storage Device(DASD)磁盘的 64 位 IBM Z 系统中执行安装时，需要这个命令，否则未格式化的磁盘不会被格式化并在安装过程中使用。仅使用命令一次。

#### 语法

```
zerombr
```

#### 备注

- 在 64 位 IBM Z 上，如果指定了 **zerombr**，安装程序可以看到的直接访问存储设备 (DASD) 都会使用 `dasdfmt` 自动低级格式化。这个命令还可防止用户在互动安装过程中进行选择。
- 如果没有指定 **zerombr**，且安装程序至少可以看到一个未格式化的 DASD，非互动的 Kickstart 安装将无法成功退出。
- 如果没有指定 **zerombr**，且安装程序至少可以看到一个未格式化的 DASD，如果用户同意格式化所有可见和未格式化的 DASD，则会退出交互式安装。要绕过这个过程，请只激活那些您要在安装过程中使用的 DASD。您总是可在安装完成后添加更多的 DASD。
- 这个命令没有选项。

### B.5.19. zfcplun

**zfcplun** Kickstart 命令是可选的。它定义了光纤通道设备。

这个选项只适用于 64 位 IBM Z。必须指定以下所有选项。

#### 语法

```
zfcplun --devnum=devnum [--wwpn=wwpn --fcplun=lun]
```

#### 选项

- **--devnum=** - 设备号 (zFCP 适配器设备总线 ID)。
- **--WWPN=** - 设备的全球端口名称 (WWPN)。形式为 16 位数字，前面带有 **0x**。
- **--fcplun=** - 该设备的逻辑单元号 (LUN)。形式为 16 位数字，前面带有 **0x**。



### 注意

如果提供了自动 LUN 扫描且安装了 8 或更高版本，则可以指定 FCP 设备总线 ID。否则，所有这三个参数都是必需的。如果没有通过 `zfcplib.allow_lun_scan` 模块参数禁用，自动 LUN 扫描可用于在 NPIV 模式下对 FCP 设备的操作。（默认启用）它对使用指定的总线 ID 附加到 FCP 设备上的存储区域网络中发现的所有 SCSI 设备提供访问权限。

### 示例

```
zfcplib --devnum=0.0.4000 --wwpn=0x5005076300C213e9 --fcplun=0x5022000000000000
zfcplib --devnum=0.0.4000
```

## B.6. RHEL 安装程序提供的附加组件的 KICKSTART 命令

这部分中的 Kickstart 命令与 Red Hat Enterprise Linux 安装程序默认提供的附加组件相关：Kdump 和 OpenSCAP。

### B.6.1. %addon com\_redhat\_kdump

`%addon com_redhat_kdump` Kickstart 命令是可选的。这个命令配置 kdump 内核崩溃转储机制。

#### 语法

```
%addon com_redhat_kdump [OPTIONS]
%end
```



### 注意

这个命令的语法比较特殊，因为它是一个附加组件，不是内置的 Kickstart 命令。

#### 备注

Kdump 是内核崩溃转储机制，可让您保存系统内存内容，以便稍后进行分析。它依赖于 `kexec`，可用于在不重新启动系统的情况下从另一个内核上下文引导 Linux 内核，并保留第一个内核内存的内容，否则会丢失第一个内核的内存。

如果系统崩溃，`kexec` 引导进入第二个内核（捕获内核）。这个捕获内核位于系统内存的保留部分。然后 Kdump 会捕获崩溃内核的内存（崩溃转储）并将其保存到指定位置。无法使用这个 Kickstart 命令配置该位置；必须在安装后通过编辑 `/etc/kdump.conf` 配置文件来配置该位置。

有关 Kdump 的详情，请参考 [安装 kdump](#)。

#### 选项

- `--enable` - 在安装的系统中启用 kdump。
- `--disable` - 在安装的系统中禁用 kdump。
- `--reserve-mb=` - 要为 kdump 保留的内存量，单位为 MiB。例如：

```
%addon com_redhat_kdump --enable --reserve-mb=128
%end
```

您还可以指定 **auto** 而不是数字值。在这种情况下，安装程序将根据 *管理、监控和更新内核* 文档中的 **kdump 内存要求** 部分中描述的标准自动决定内存量。

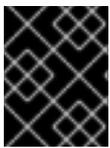
如果启用 **kdump** 且没有指定 **--reserve-mb=** 选项，则会使用 **auto** 值。

- **--enablefadump** - 在允许它的系统中（特别是 IBM Power Systems 服务器）启用固件辅助转储。

## B.6.2. %addon org\_fedora\_oscapp

**%addon org\_fedora\_oscapp** Kickstart 命令是可选的。

OpenSCAP 安装程序附加组件是用来在安装的系统中应用 SCAP（安全内容自动化协议）内容 - 安全策略。从 Red Hat Enterprise Linux 7.2 开始默认启用此附加组件。启用后，会自动安装提供这个功能的软件包。但默认情况下不会强制任何策略。这代表，除非特别指定，在安装过程中或安装后不会执行任何检查。



### 重要

不需要在所有系统中应用安全策略。只有在您的机构规则或政府法规强制某种策略时，才使用这个命令。

与大多数命令不同，此附加组件不接受常规选项，而是使用 **%addon** 定义正文中的键值对。这些键值对无法验证空白。值可以选择使用单引号 (') 或者双引号 (") 括起来。

### 语法

```
%addon org_fedora_oscapp
key = value
%end
```

### Keys

以下键可以被附加组件识别：

#### content-type

安全内容的类型。可能的值有 **datastream**、**archive**、**rpm** 和 **scap-security-guide**。

如果 **content-type** 是 **scap-security-guide**，则附加组件将使用 **scap-security-guide** 软件包提供的内容，该内容存在于引导介质中。这意味着，除了 **profile** 外所有其他键都无效。

#### content-url

安全内容的位置。必须通过 HTTP、HTTPS 或者 FTP 访问该内容。必须有可用的网络连接方可访问远程位置中的内容定义。

#### datastream-id

**content-url** 值中引用的数据流的 ID。仅在 **content-type** 是 **datastream** 时使用。

#### xccdf-id

要使用的基准 ID。

#### content-path

应使用到 **datastream** 或 **XCCDF** 文件的路径，在归档中使用相对路径。

### 配置集

要应用的配置集 ID。使用 **default** 应用 default 配置文件。

## fingerprint

由 **content-url** 引用的内容的 MD5、SHA1 或 SHA2 的校验和。

## tailoring-path

应使用的定制文件的路径，在归档中作为相对路径指定。

## 示例

- 以下是一个 **%addon org\_fedora\_oscap** 部分的示例，它使用安装介质中的 **scap-security-guide** 中的内容：

### 例 B.1. 使用 SCAP 安全指南的 OpenSCAP 附加组件定义示例

```
%addon org_fedora_oscap
content-type = scap-security-guide
profile = xccdf_org.ssgproject.content_profile_pci-dss
%end
```

- 以下是从 web 服务器加载自定义配置集的复杂示例：

### 例 B.2. 使用 Datastream 的 OpenSCAP 附加组件定义示例

```
%addon org_fedora_oscap
content-type = datastream
content-url = http://www.example.com/scap/testing_ds.xml
datastream-id = scap_example.com_datastream_testing
xccdf-id = scap_example.com_cref_xccdf.xml
profile = xccdf_example.com_profile_my_profile
fingerprint = 240f2f18222faa98856c3b4fc50c4195
%end
```

## 其它资源

- [安全强化](#)
- [OpenSCAP 安装程序附加组件](#)
- [OpenSCAP Portal](#)

## B.7. ANACONDA 中使用的命令

**pwpolicy** 命令是 Anaconda UI 特定的命令，只能在 kickstart 文件的 **%anaconda** 部分中使用。

### B.7.1. pwpolicy

**pwpolicy** Kickstart 命令是可选的。在安装过程中使用这个命令强制使用自定义密码策略。策略要求您为 root、用户或 luks 用户帐户创建密码。密码长度和强度等因素决定了密码的有效性。

## 语法

```
pwpolicy name [--minlen=length] [--minquality=quality] [--strict|--notstrict] [--emptyok|--notempty] [--changesok|--nochanges]
```

## 必填选项

- *name* - 使用 **root**、**user** 或 **luks** 替换，以分别强制执行与 **root** 密码、用户密码或 LUKS 密码短语相关的策略。

## 可选选项

- **--minlen=** - 以字符为单位设置允许密码长度的最小值。默认值为 **6**。
- **--minquality=** - 设置 **libpwquality** 库定义的最小允许密码质量。默认值为 **1**。
- **--strict** - 启用严格的密码强制。不接受没有满足 **--minquality=** 和 **--minlen=** 中指定的要求的密码。默认禁用这个选项。
- **--notstrict** - 在 GUI 中点 **Done** 两次后，可以接受不符合 **--minquality=** 和 **--minlen=** 选项中指定的最低质量要求的密码。文本模式界面使用类似的机制。
- **--emptyok** - 允许使用空密码。对于用户密码，默认启用。
- **--notempty** - 不允许使用空密码。对于 root 密码和 LUKS 密码，默认启用。
- **--changesok** - 允许在用户界面中更改密码，即使 Kickstart 文件已指定了密码。默认禁用此选项。
- **--nochanges** - 不允许更改在 Kickstart 文件中已设置的密码。默认启用此选项。

## 备注

- **pwpolicy** 命令是 Anaconda UI 特定的命令，只能在 kickstart 文件的 **%anaconda** 部分中使用。
- **libpwquality** 程序库用于检查最低密码要求（长度和服务）。您可以使用 **libpwquality** 软件包提供的 **pwscore** 和 **pwmake** 命令检查密码质量得分，或使用给定分数创建随机密码。有关这些命令的详情，请查看 **pwscore(1)** 和 **pwmake(1)** man page。

## B.8. KICKSTART 命令进行系统恢复

这部分中的 Kickstart 命令修复安装了的系统。

### B.8.1. rescue

**rescue** Kickstart 命令是可选的。它提供有 root 特权的 shell 环境，以及一组系统管理工具修复安装以及排除问题，例如：

- 以只读方式挂载文件系统
- 阻塞列表或添加驱动程序磁盘中提供的驱动程序
- 安装或升级系统软件包
- 管理分区



### 注意

Kickstart 的救援模式与系统的救援模式和紧急模式不同，后者是 systemd 和服务管理器的一部分。

**rescue** 命令不自行修改系统。它只能通过通过 `/mnt/sysimage` 下挂载系统的读写模式来设置救援环境。您可以选择不挂载系统，或者将其挂载为只读模式。仅使用命令一次。

## 语法

```
rescue [--nomount|--romount]
```

## 选项

- **--nomount** 或 **--romount** - 控制如何在救援环境中挂载安装的系统。默认情况下，安装程序会找到您的系统并以读写模式挂载它，同时告知它在什么位置进行挂载。您可以选择不挂载任何内容（**--nomount** 选项）或以只读模式挂载（**--romount** 选项）。只能使用这两个选项中的一个。

## 备注

要运行救援模式，请制作 Kickstart 文件的副本，并在该文件中包含 **rescue** 命令。

使用 **rescue** 命令可使安装程序执行以下步骤：

1. 运行 **%pre** 脚本。
2. 为救援模式设置环境。  
下面的 kickstart 命令生效：
  - a. updates
  - b. sshpw
  - c. logging
  - d. lang
  - e. network
3. 设置高级存储环境。  
下面的 kickstart 命令生效：
  - a. fcoe
  - b. iscsi
  - c. iscsiname
  - d. nvdimmm
  - e. zfcpl

4. 挂载系统

```
rescue [--nomount|--romount]
```

5. 运行 **%post** 脚本  
只有在安装的系统被挂载为读写模式时才运行这一步。
6. 启动 shell

## 7. 重启系统

## 附录 C. 引导选项参考

本节介绍可以用来修改安装程序默认行为的一些引导选项。有关 Kickstart 和高级引导选项，请查看 [RHEL 安装程序的引导选项](#) 文档。

### C.1. 安装源引导选项

这部分论述了各种安装源引导选项。

#### inst.repo=

**inst.repo=** 引导选项用来指定安装源，即提供软件包存储库的位置以及描述它们的有效 **.treeinfo** 文件。例如：**inst.repo=cdrom**。**inst.repo=** 选项的目标必须是以下安装介质之一：

- 一个可安装的树，它是一个目录结构，包含了安装程序镜像、软件包和存储库数据的目录结构，以及一个有效的 **.treeinfo** 文件。
- 一个 DVD（系统 DVD 驱动器中有物理磁盘）
- 完整的 Red Hat Enterprise Linux 安装 DVD 的 ISO 镜像放在磁盘或者系统可访问的网络位置上。

使用 **inst.repo=** 引导选项使用不同的格式配置不同的安装方法。下表包含 **inst.repo=** 引导选项语法的详情：

表 C.1. inst.repo= 引导选项和安装源的类型和格式

源类型	引导选项格式	源格式
CD/DVD 驱动器	<b>inst.repo=cdrom:&lt;device&gt;</b>	将 DVD 作为物理磁盘安装。 [a]
可挂载设备 (HDD 和 USB 盘)	<b>inst.repo=hd:&lt;device&gt;:/&lt;path&gt;</b>	安装 DVD 的镜像文件。
NFS 服务器	<b>inst.repo=nfs: [options:]&lt;server&gt;:/&lt;path&gt;</b>	DVD 或安装树的镜像文件， 这是安装 DVD 中的目录和文件的完整副本。 [b]
HTTP 服务器	<b>inst.repo=http://&lt;host&gt;/&lt;path&gt;</b>	安装树是安装 DVD 上目录和文件的完整副本。
HTTPS 服务器	<b>inst.repo=https://&lt;host&gt;/&lt;path&gt;</b>	
FTP 服务器	<b>inst.repo=ftp://&lt;username&gt;:&lt;password&gt;@&lt;host&gt;/&lt;path&gt;</b>	
HMC	<b>inst.repo=hmc</b>	

源类型	引导选项格式	源格式
[a]	如果禁用了设备，安装程序会自动搜索包含安装 DVD 的驱动器。	
[b]	NFS 服务器选项默认使用 NFS 协议版本 3。要使用不同的版本，请将 <b>nfsvers=X</b> 添加到 <i>options</i> 中，将 X 替换为您要使用的版本号。	

使用以下格式设定磁盘设备名称：

- 内核设备名称，例如 **/dev/sda1** 或者 **sdb2**
- 文件系统标签，如 **LABEL=Flash** 或 **LABEL=RHEL8**
- 文件系统 UUID，如 **UUID=8176c7bf-04ff-403a-a832-9557f94e61db**

非字母数字字符必须使用 **\xNN** 表示，其中 *NN* 是字符对应的十六进制代码。例如：**\x20** 是一个空格 (" ")。

**inst.addrepo=**

使用 **inst.addrepo=** 引导选项添加一个额外的存储库，该存储库可用作另一个安装源和主存储库 (**inst.repo=**)。您可以在一次引导过程中多次使用 **inst.addrepo=** 引导选项。下表包含 **inst.addrepo=** 引导选项语法的详情。



### 注意

**REPO\_NAME** 是该程序库的名称，它是安装过程中必需的。这些软件仓库仅在安装过程中使用，它们不会在安装的系统中安装。

有关统一的 ISO 的更多信息，请参阅 [统一的 ISO](#)。

表 C.2. 安装源和引导选项格式

安装源	引导选项格式	附加信息
在 URL 中可安装的树	<b>inst.addrepo=REPO_NAME, [http,https,ftp]://&lt;host&gt;/&lt;path&gt;</b>	在给定的 URL 中查找可安装的树。
NFS 路径中的可安装树	<b>inst.addrepo=REPO_NAME,nfs://&lt;server&gt;/&lt;path&gt;</b>	在给定的 NFS 路径中查找可安装的树。主机后需要一个冒号。安装程序将 <b>nfs://</b> 后的所有内容直接传递给 <b>mount</b> 命令，而不是根据 RFC 2224 解析 URL。

安装源	引导选项格式	附加信息
安装环境中的可安装树	<b>inst.addrepo=REPO_NAME, file://&lt;path&gt;</b>	在安装环境中的给定位置查找可安装的树。要使用这个选项，必须在安装程序尝试载入可用软件组前挂载该程序库。这个选项的好处是，您可以在一个可引导 ISO 上有多个库，您还可以从 ISO 安装主存储库和其他软件仓库。到额外软件仓库的路径为 <b>/run/install/source/REPO_ISO_PATH</b> 。另外，您可以在 Kickstart 文件的 <b>%pre</b> 部分中挂载存储库目录。该路径必须是绝对的，并以 <b>/</b> 开头，例如 <b>inst.addrepo=REPO_NAME, file:/// &lt;path&gt;</b>
磁盘	<b>inst.addrepo=REPO_NAME, hdd:&lt;device&gt;:&lt;path&gt;</b>	挂载给定的 <b>&lt;device&gt;</b> 分区并从由 <b>&lt;path&gt;</b> 指定的 ISO 中安装。如果没有指定 <b>&lt;path&gt;</b> ，安装程序会在 <b>&lt;device&gt;</b> 中查找有效安装 ISO。这个安装方法需要一个含有有效可安装树的 ISO。

### inst.stage2=

**inst.stage2=** 引导选项指定安装程序运行时镜像的位置。这个选项需要包含有效的 **.treeinfo** 文件的路径，并从 **.treeinfo** 文件中读取运行时镜像位置。如果 **.treeinfo** 文件不可用，安装程序会尝试从 **images/install.img** 中载入该镜像。

当您没有指定 **inst.stage2** 选项时，安装程序会尝试使用 **inst.repo** 选项指定的位置。

当您稍后在安装程序中手动指定安装源时，请使用这个选项。例如，当您要选择 Content Delivery Network (CDN) 作为安装源时。安装 DVD 和引导 ISO 已经包含一个合适的 **inst.stage2** 选项，来从相应的 ISO 引导安装程序。

如果要指定安装源，请使用 **inst.repo=** 选项。



### 注意

默认情况下，在安装介质中使用 **inst.stage2=** 引导选项，并被设置特定的标签；例如：**inst.stage2=hd:LABEL=RHEL-x-0-0-BaseOS-x86\_64**。如果您修改了包含运行时镜像的文件系统的默认标签，或者使用自定义流程引导安装系统，请验证 **inst.stage2=** 引导选项是否已被设置为正确的值。

### inst.noverifyssl

使用 **inst.noverifyssl** 引导选项防止安装程序为所有 HTTPS 连接验证 SSL 证书，但附加 Kickstart 软件仓库除外，其中 **--noverifyssl** 可为每个存储库设置。

例如：如果您的远程安装源使用自签名 SSL 证书，**inst.noverifyssl** 引导选项可让安装程序在不验证 SSL 证书的情况下完成安装。

### 使用 inst.stage2= 指定源时的示例

-

```
inst.stage2=https://hostname/path_to_install_image/ inst.noverifyssl
```

### 使用 `inst.repo=` 指定源时的示例

```
inst.repo=https://hostname/path_to_install_repository/ inst.noverifyssl
```

### `inst.stage2.all`

使用 `inst.stage2.all` 引导选项来指定几个 HTTP、HTTPS 或者 FTP 源。您可以多次使用 `inst.stage2=` 引导选项,同时使用 `inst.stage2.all` 选项按顺序从源获取镜像,直到成功为止。例如:

```
inst.stage2.all
inst.stage2=http://hostname1/path_to_install_tree/
inst.stage2=http://hostname2/path_to_install_tree/
inst.stage2=http://hostname3/path_to_install_tree/
```

### `inst.dd=`

`inst.dd=` 引导选项在安装过程中用来执行驱动程序更新。有关如何在安装过程中更新驱动程序的更多信息,请参阅 [自动安装 RHEL](#)。

### `inst.repo=hmc`

这个选项消除了外部网络设置的要求并扩展安装选项。当使用二进制 DVD 引导时,安装程序会提示您输入附加内核参数。要将 DVD 设置为安装源,请在内核参数中附加 `inst.repo=hmc` 选项。然后安装程序启用支持元素(SE)和硬件管理控制台(HMC)文件访问,从 DVD 中获取 stage2 的镜像,并提供对 DVD 上软件包的访问以供软件选择。

### `proxy=`

从 HTTP、HTTPS 和 FTP 协议执行安装时使用这个引导选项。例如:

```
[PROTOCOL://][USERNAME[:PASSWORD]@]HOST[:PORT]
```

### `inst.nosave=`

使用 `inst.nosave=` 引导选项控制安装日志以及没有保存到安装的系统中的相关文件,如 `input_ks`、`output_ks`、`all_ks`、`logs` 和 `all`。您可以以逗号分隔的方式组合多个值。例如,

```
inst.nosave=input_ks,logs
```



#### 注意

`inst.nosave` 引导选项用于排除安装系统中无法被 Kickstart `%post` 脚本删除的文件,如日志和输入/输出 Kickstart 结果。

### `input_ks`

禁用保存输入 Kickstart 结果的功能。

### `output_ks`

禁用保存安装程序生成的 Kickstart 结果的功能。

### `all_ks`

禁用保存输入和输出 Kickstart 结果的功能。

### 日志

禁用保存所有安装日志的功能。

**all**

禁用保存所有 Kickstart 结果以及所有日志的功能。

**inst.multilib**

使用 **inst.multilib** 引导选项将 DNF 的 **multilib\_policy** 设置为 **all**，而不是 **best**。

**inst.memcheck**

**inst.memcheck** 引导选项执行检查来验证系统是否有足够 RAM 来完成安装。如果没有足够的 RAM，安装过程将停止。系统检查不是完全准确的，在安装过程中的内存用量要根据软件包的选择、用户界面（如图形或文本）和其它参数而定。

**inst.nomemcheck**

**inst.nomemcheck** 引导选项不会执行检查来验证系统是否有足够 RAM 完成安装。在低于推荐最小内存量的系统中进行安装不被支持，并可能导致安装过程失败。

## C.2. 网络引导选项

如果您的场景需要通过网络从镜像引导，而不是从本地镜像引导，您可以使用以下选项来自定义网络引导。



### 注意

使用 **dracut** 工具初始化网络。有关 **dracut** 选项的完整列表，请查看 **dracut.cmdline(7)** 手册页。

**ip=**

使用 **ip=** 引导选项配置一个或多个网络接口。要配置多个接口，请使用以下方法之一；

- 多次使用 **ip** 选项，每个接口一次；为此，请使用 **rd.neednet=1** 选项，并使用 **bootdev** 选项指定一个主引导接口。
- 使用 **ip** 选项一次，然后使用 Kickstart 设置其他接口。这个选项接受几种不同的格式。下表包含大多数常用选项的信息。

在以下表格中：

- **ip** 参数指定客户端 IP 地址，IPv6 需要方括号，如 192.0.2.1 或 [2001:db8::99]。
- **gateway** 参数是默认网关。IPv6 需要方括号。
- **netmask** 参数是要使用的子网掩码。这可以是完整子网掩码（例如：255.255.255.0），或是一个前缀（例如 64）。
- **hostname** 参数是客户端系统的主机名。这个参数是可选的。

表 C.3. 配置网络接口的引导选项格式

引导选项格式	配置方法
<b>ip=method</b>	自动配置任意接口
<b>ip=interface:method</b>	自动配置特定的接口

引导选项格式	配置方法
<b>ip=ip::gateway:netmask:hostname:interface:none</b>	静态配置，例如 IPv4 : <b>ip=192.0.2.1::192.0.2.254:255.255.255.5.0:server.example.com:enp1s0:none</b>  IPv6: <b>ip=[2001:db8::1]::[2001:db8::ffe]:64:server.example.com:enp1s0:none</b>
<b>ip=ip::gateway:netmask:hostname:interface:method:mtu</b>	自动配置特定接口并进行覆盖

### 自动接口的配置方法

带有覆盖的特定接口的自动配置 方法使用自动配置的指定方法（如 **dhcp**）打开接口，但会覆盖自动获得的 IP 地址、网关、子网掩码、主机名或其他指定的参数。所有参数都是可选的，因此仅指定您要覆盖的参数。

**method** 参数可以是以下任意一个：

#### DHCP

**dhcp**

#### IPv6 DHCP

**dhcp6**

#### IPv6 自动配置

**auto6**

#### iSCSI Boot Firmware Table (iBFT)

**ibft**



### 注意

- 如果您使用需要网络访问的引导选项，如 **inst.ks=http://host/path**，但没有指定 **ip** 选项，则 **ip** 选项的默认值是 **ip=dhcp**。
- 要自动连接到 iSCSI 目标，请使用 **ip=ibft** 引导选项激活访问目标的网络设备。

#### nameserver=

**nameserver=** 选项指定名称服务器的地址。您可以多次使用这个选项。



### 注意

**ip=** 参数需要方括号。但是 IPv6 地址不能使用方括号。IPv6 地址使用的正确语法示例为 **nameserver= 2001:db8::1**。

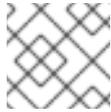
#### bootdev=

**bootdev=** 选项指定引导接口。如果您使用多个 **ip** 选项，则这个选项是必须的。

**ifname=**

**ifname=** 选项为具有给定 MAC 地址的网络设备分配接口名称。您可以多次使用这个选项。语法是 **ifname=interface:MAC**。例如：

```
ifname=eth0:01:23:45:67:89:ab
```

**注意**

**ifname=** 选项是在安装过程中设置自定义网络接口的唯一方法。

**inst.dhcpclass=**

**inst.dhcpclass=** 选项指定 DHCP 厂商级识别符。**dhcpd** 服务将此值识别为 **厂商级识别符**。默认值为 **anaconda-\$(uname -srm)**。要确保正确应用了 **inst.dhcpclass=** 选项，请在安装过程的早期阶段通过添加 **ip** 选项来请求网络激活。

**inst.waitfornet=**

使用 **inst.waitfornet=SECONDS** 引导选项可导致安装系统在安装前等待网络连接。**SECONDS** 参数中给出的值指定在超时和继续安装过程（即使网络连接不存在）前等待网络连接的最长时间。

**vlan=**

使用 **vlan=** 选项来在具有给定名称的指定接口上配置虚拟 LAN(VLAN)设备。语法为 **vlan=name:interface**。例如：

```
vlan=vlan5:enp0s1
```

这会在 **enp0s1** 接口上配置一个名为 **vlan5** 的 VLAN 设备。名称可采用以下格式：

- VLAN\_PLUS\_VID: **vlan0005**
- VLAN\_PLUS\_VID\_NO\_PAD: **vlan5**
- DEV\_PLUS\_VID: **enp0s1.0005**
- DEV\_PLUS\_VID\_NO\_PAD: **enp0s1.5**

**bond=**

使用 **bond=** 选项来配置绑定设备，语法如下：**bond=name[:interfaces][:options]**。使用绑定设备名称替换 *name*，使用以逗号分隔的物理（以太网）接口列表替换 *interfaces*，并使用以逗号分隔的绑定选项列表替换 *options*。例如：

```
bond=bond0:enp0s1,enp0s2:mode=active-backup,tx_queues=32,downdelay=5000
```

有关可用选项的列表，请执行 **modinfo** 绑定命令。

**team=**

使用 **team=** 选项来配置团队设备，语法如下：**team=name:interfaces**。使用所需团队设备替换 *name*，使用要用作团队设备中底层接口的、以逗号分隔的物理(以太网)设备列表替换 *interfaces*。例如：

```
team=team0:enp0s1,enp0s2
```

**bridge=**

使用 **bridge=** 选项来配置网桥设备，语法如下：**bridge=name:interfaces**。使用所需的网桥设备名称替换 *name*，使用要用作网桥设备中底层接口的、以逗号分隔的物理(以太网)设备列表替换 *interfaces*。例如：

```
bridge=bridge0:enp0s1,enp0s2
```

**其它资源**

- [配置和管理网络](#)

**C.3. 控制台引导选项**

这部分论述了如何为控制台、显示器显示和键盘配置引导选项。

**console=**

使用 **console=** 选项指定您要用作主控制台的设备。例如，若要在第一个串行端口上使用控制台，可使用 **console=ttyS0**。在使用 **console=** 参数时，安装以文本 UI 开始。如果必须多次使用 **console=** 选项，则引导信息会在所有指定的控制台上显示。但是，安装程序只使用最后指定的控制台。例如：如果您指定了 **console=ttyS0 console=ttyS1**，安装程序将使用 **ttyS1**。

**inst.lang=**

使用 **inst.lang=** 选项设定要在安装中使用的语言。要查看区域设置的列表，请输入命令 **locale -a | grep \_** 或 **localectl list-locales | grep \_** 命令。

**inst.singlelang**

使用 **inst.singlelang** 选项以单一语言模式安装，这会导致安装语言和语言支持配置没有可用的互动选项。如果使用 **inst.lang** 引导选项或 **lang** Kickstart 命令指定语言，则会使用它。如果没有指定语言，安装程序默认为 **en\_US.UTF-8**。

**inst.geoloc=**

使用 **inst.geoloc=** 选项在安装程序中配置地理位置的使用。地理位置用于预先设置语言和时区，并使用以下语法：**inst.geoloc=value**。值可以是以下参数：

- 禁用地理位置：**inst.geoloc=0**
- 使用 Fedora GeolP API: **inst.geoloc=provider\_fedora\_geolp**。
- 使用 Hostip.info GeolP API: **inst.geoloc=provider\_hostip**。  
如果您没有指定 **inst.geoloc=** 选项，则默认选项是 **provider\_fedora\_geolp**。

**inst.keymap=**

使用 **inst.keymap=** 选项指定用于安装的键盘布局。

**inst.cmdline**

使用 **inst.cmdline** 选项强制安装程序在命令行模式下运行。这个模式不允许任何互动，且您必须在 Kickstart 文件或者命令行中指定所有选项。

**inst.graphical**

使用 **inst.graphical** 选项强制安装程序在图形模式下运行。图形模式是默认设置。

**inst.text**

使用 **inst.text** 选项强制安装程序在文本模式而不是图形模式中运行。

**inst.noninteractive**

使用 **inst.noninteractive** 引导选项以非互动模式运行安装程序。在非互动模式下不允许用户互动 **inst.noninteractive**，您可以在图形或文本安装中使用 **inst.noninteractive** 选项。当您在文本模式下使用 **inst.noninteractive** 选项时，它的行为与 **inst.cmdline** 选项一样。

#### inst.resolution=

使用 **inst.resolution=** 选项指定图形模式中的页面分辨率。格式为 **NxM**，其中 *N* 是屏幕宽度，*M* 是屏幕高度（单位为像素）。推荐的分辨率为 1024x768。

#### inst.vnc

使用 **inst.vnc** 选项，使用虚拟网络计算(VNC)来运行图形安装。您必须使用 VNC 客户端应用程序与安装程序交互。启用 VNC 共享时，可以连接多个客户端。在文本模式中安装使用 VNC 启动的系统。

#### inst.vncpassword=

使用 **inst.vncpassword=** 选项在 VNC 服务器中设定安装程序使用的密码。

#### inst.vncconnect=

使用 **inst.vncconnect=** 选项连接到给定主机位置处的侦听 VNC 客户端，例如：**inst.vncconnect=<host>[:<port>]**，默认端口为 5900。您可以通过输入 **vncviewer -listen** 命令来使用此选项。

#### inst.xdriver=

使用 **inst.xdriver=** 选项指定在安装过程中和在安装的系统上使用的 X 驱动程序名称。

#### inst.usefbx

使用 **inst.usefbx** 选项提示安装程序使用帧缓冲 X 驱动程序，而不是具体硬件驱动程序。这个选项等同于 **inst.xdriver=fbdev** 选项。

#### modprobe.blacklist=

使用 **modprobe.blacklist=** 选项阻止列表或完全禁用一个或多个驱动程序。安装启动时，使用这个选项禁用的驱动程序（模式）无法加载。安装完成后，安装的系统会保持这些设置。您可以在 **/etc/modprobe.d/** 目录中找到 blocklisted 驱动程序的列表。使用以逗号分隔的列表禁用多个驱动程序。例如：

```
modprobe.blacklist=ahci,firewire_ohci
```



#### 注意

您可以将 **modprobe.blacklist** 与不同的命令行选项一起使用。例如，将其与 **inst.dd** 选项一起使用，以确保现有驱动程序的更新版本是从驱动程序更新磁盘加载的：

```
modprobe.blacklist=virtio_blk
```

#### inst.xtimeout=

使用 **inst.xtimeout=** 选项指定启动 X 服务器的时间（以秒为单位）。

#### inst.sshd

使用 **inst.sshd** 选项在安装过程中启动 **sshd** 服务，以便您可以在安装过程中使用 SSH 连接到该系统，并监控安装进程。有关 SSH 的详情，请参考 **ssh(1)** man 页。默认情况下，**sshd** 选项只在 64 位 IBM Z 构架上自动启动。在其它构架中，除非您使用 **inst.sshd** 选项，否则不会启动 **sshd**。



#### 注意

在安装过程中，**root** 帐户默认没有密码。您可在安装过程中使用 **sshpw** Kickstart 命令设定根密码。

#### inst.kdump\_addon=

使用 `inst.kdump_addon=` 选项启用或者禁用安装程序中的 Kdump 配置页面（附加组件）。这个屏幕默认是启用的；使用 `inst.kdump_addon=off` 禁用它。禁用该附加组件可在图形和文本界面以及 `%addon com_redhat_kdump` Kickstart 命令中禁用 Kdump 页面。

## C.4. 调试引导选项

本节论述了在调试问题时可以使用的选项。

### inst.rescue

使用 `inst.rescue` 选项运行救援环境，以诊断和修复系统。例如：您可以 [在救援模式中修复文件系统](#)。

### inst.updates=

使用 `inst.updates=` 选项指定您要在安装过程中应用的 `updates.img` 文件的位置。`updates.img` 文件可以从多个源中派生出来。

表 C.4. `updates.img` 文件源

源	描述	示例
从网络更新	指定 <code>updates.img</code> 的网络位置。这不需要对安装树进行任何修改。要使用这个方法，请编辑内核命令行使其包含 <code>inst.updates</code> 。	<code>inst.updates=http://website.com/path/to/updates.img</code>
从磁盘镜像更新	在软盘或者 USB 密钥上保存 <code>updates.img</code> 。这只适用于 <code>updates.img</code> 的 <code>ext2</code> 文件系统类型。要在软盘驱动器中保存镜像内容,请插入软盘并运行命令。	<code>dd if=updates.img of=/dev/fd0 bs=72k count=20</code> 。要使用 USB 密钥或闪存介质，请使用 USB 闪存驱动器的设备名称替换 <code>/dev/fd0</code> 。
安装树中的更新	如果您使用 CD、磁盘、HTTP 或者 FTP 安装，请在安装树中保存 <code>updates.img</code> ，以便所有安装都可以检测到 <code>.img</code> 文件。文件名必须是 <code>updates.img</code> 。	对于 NFS 安装，将文件保存在 <code>images/</code> 目录中，或保存在 <code>RHupdates/</code> 目录中。

### inst.loglevel=

使用 `inst.loglevel=` 选项指定终端日志信息的最低级别。这个选项只适用于终端日志记录；日志文件总是包含所有级别的信息。这个选项的可能值从最低级到最高级为：

- `debug`
- `info`
- `warning`
- 错误
- `critical`

默认值为 **info**，就是说默认情况下，日志记录终端会显示从 **info** 级别到 **critical** 级别范围的信息。

#### inst.syslog=

安装开始时，将日志消息发送给指定主机上的 **syslog** 进程。只有在远程 **syslog** 进程配置为接受进入的连接时，才可以使用 **inst.syslog=**。

#### inst.virtio=

使用 **inst.virtio=** 选项指定使用哪个 virtio 端口（一个位于 `/dev/virtio-ports/name` 的字符设备）来转发日志。默认值为 **org.fedoraproject.anaconda.log.0**。

#### inst.zram=

控制在安装过程中 zRAM swap 的使用。选项在系统 RAM 中创建一个压缩的块设备，并将其用于 swap 空间，而不是使用磁盘。这个设置可让安装程序使用较少的可用内存运行，并改进安装速度。您可以使用以下值配置 **inst.zram=** 选项：

- **inst.zram=1** 启用 zRAM swap，无论系统内存大小如何。默认情况下，在内存为 2 GiB 或更少的系统上启用 zRAM 上的 swap。
- **inst.zram=0** 禁用 zRAM swap，无论系统内存大小如何。默认情况下，在超过 2 GiB 内存的系统上禁用 zRAM 上的 swap。

#### rd.live.ram

将 **images/install.img** 中的 **stage 2** 镜像复制到 RAM 中。请注意，这会增加安装镜像所需的内存，镜像的大小通常介于 400 到 800MB 之间。

#### inst.nokill

当发生致命错误或安装过程结束时，防止安装程序重新启动。使用它捕获重启后可能丢失的安装日志。

#### inst.noshell

在安装过程中防止终端会话 2(tty2)上的 shell。

#### inst.notmux

防止在安装过程中使用 tmux。输出在没有终端控制字符的情况下生成，用于非互动性。

#### inst.remotelog=

使用 TCP 连接将所有日志发送到远程 **host:port**。如果没有监听程序而安装正常进行，则该连接将会停用。

## C.5. 存储启动选项

这部分论述了您可以指定用来定制从存储设备引导的选项。

#### inst.nodmraid

禁用 **dmraid** 支持。



#### 警告

请小心使用这个选项。如果您有一个被错误地识别为固件 RAID 阵列一部分的磁盘，则它上面可能有一些过时的 RAID 元数据，这些数据必须使用合适的工具删除，如 **dmraid** 或 **wipefs**。

## inst.nompath

禁用对多路径设备的支持。只有当您的系统出现误报，错误地将普通块设备识别为多路径设备时，才使用这个选项。



### 警告

请小心使用这个选项。不要在多路径硬件中使用这个选项。不支持使用这个选项安装到多路径设备的一个路径。

## inst.gpt

强制安装程序将分区信息安装到 GUID 分区表(GPT)，而不是主引导记录(MBR)。这个选项在基于 UEFI 的系统上无效，除非采用与 BIOS 兼容的模式。通常，基于 BIOS 的系统和基于 UEFI 的系统在兼容模式下尝试使用 MBR 模式来存储分区信息，除非磁盘大小为  $2^{32}$  个扇区或大于 2 个扇区。磁盘扇区大小通常是 512 字节，通常相当于 2 TiB。**inst.gpt** 引导选项允许将 GPT 写到较小的磁盘。

## inst.wait\_for\_disks=

使用 **inst.wait\_for\_disks=** 选项指定安装程序在安装开始时等待磁盘设备出现的秒数。当您使用 **OEMDRV-labeled** 设备自动加载 Kickstart 文件或内核驱动程序时使用这个选项，但设备在引导过程中需要很长时间才出现。默认情况下，安装程序等待 5 秒。使用 0 秒来最小化延迟。

## C.6. 弃用的引导选项

本节包含已弃用的引导选项的信息。安装程序仍然可以接受这些选项，但它们已被弃用，并计划在以后的 Red Hat Enterprise Linux 版本中删除。

### method

**method** 选项是 **inst.repo** 的别名。

### dns

使用 **nameserver** 替代 **dns**。请注意，名称服务器不接受用逗号分开的列表；使用多个名称服务器选项。

### netmask, gateway, hostname

**netmask**、**gateway** 和 **hostname** 选项作为 **ip** 选项的一部分提供。

### ip=bootif

自动使用 PXE 提供的 **BOOTIF** 选项，因此无需使用 **ip=bootif**。

### ksdevice

表 C.5. ksdevice 引导选项的值

Value	信息
不存在	不适用
<b>ksdevice=link</b>	忽略，这个选项与默认行为相同
<b>ksdevice=bootif</b>	如果存在 <b>BOOTIF=</b> ，则忽略此选项。

Value	信息
<b>ksdevice=ibft</b>	替换为 <b>ip=ibft</b> 。详情请查看 <b>ip</b> 。
<b>ksdevice=&lt;MAC&gt;</b>	使用 <b>BOOTIF=\${MAC}/-}</b> 替代
<b>ksdevice=&lt;DEV&gt;</b>	用 <b>bootdev</b> 替换

## C.7. 删除引导选项

这部分包含从 Red Hat Enterprise Linux 中删除的引导选项。



### 注意

**dracut** 提供高级引导选项。有关 **dracut** 的更多信息，请参阅 **dracut.cmdline(7)** man page。

#### askmethod, asknetwork

**initramfs** 是完全非互动的，因此删除了 **askmethod** 和 **asknetwork** 选项。使用 **inst.repo** 或者指定合适的网络选项。

#### blacklist, nofirewire

**modprobe** 选项现在可以处理阻止列表中的内核模块。使用 **modprobe.blacklist=<mod1>**, **<mod2>**。您可以使用 **modprobe.blacklist=firewire\_ohci** 阻止 **firewire** 模块列表。

#### inst.headless=

**headless=** 选项指定安装的系统没有任何显示硬件，且安装程序不需要查找任何显示硬件。

#### inst.decorated

**inst.decorated** 选项用于在解码窗口中指定图形安装。默认情况下，窗口未经过修饰，因此它没有标题栏，调整控制等。这个选项不再需要。

#### repo=nfsiso

使用 **inst.repo=nfs:** 选项。

#### serial

使用 **console=ttyS0** 选项。

#### updates

使用 **inst.updates** 选项。

#### essid, wepkey, wpakey

Dracut 不支持无线网络。

#### ethtool

这个选项不再需要。

#### gdb

这个选项已被删除，因为很多选项可用于调试基于 **dracut** 的 **initramfs**。

#### inst.mediacheck

使用 **dracut option rd.live.check** 选项。

#### ks=floppy

---

使用 **inst.ks=hd:<device>** 选项。

### 显示

对于 UI 的远程显示，使用 **inst.vnc** 选项。

### utf8

因为默认 TERM 设置的行为与预期一致，所以不再需要这个选项。

### noipv6

IPv6 是内置在内核中，不能被安装程序删除。您可以使用 **ipv6.disable=1** 禁用 ipv6。安装的系统将使用这个设置。

### upgradeany

因为安装程序不再处理升级，所以不再需要这个选项。