



Red Hat Enterprise Linux 9

执行高级 RHEL 9 安装

使用 Kickstart 安装 RHEL

Red Hat Enterprise Linux 9 执行高级 RHEL 9 安装

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

摘要

您可以使用 Kickstart 执行高级 RHEL 安装。Kickstart 可帮助您使用简单的文本文件自动化安装。

目录

对红帽文档提供反馈	4
第 1 章 简介	5
1.1. 支持的构架	5
1.2. 安装术语	5
第 2 章 安装方法	6
2.1. 可用安装方法	6
部分 I. 使用 KICKSTART 执行自动安装	7
第 3 章 KICKSTART 安装基础	8
3.1. KICKSTART 安装是什么	8
3.2. 自动化安装工作流	8
第 4 章 创建 KICKSTART 文件	9
4.1. 使用 KICKSTART 配置工具创建 KICKSTART 文件	9
4.2. 执行手动安装来创建 KICKSTART 文件	9
4.3. 从以前的 RHEL 安装转换为一个 KICKSTART 文件	10
4.4. 使用 IMAGE BUILDER 创建自定义镜像	10
第 5 章 为安装程序提供 KICKSTART 文件	11
5.1. 基于网络安装的端口	11
5.2. 在 NFS 服务器中提供 KICKSTART 文件	11
5.3. 在 HTTP 或 HTTPS 服务器中提供 KICKSTART 文件	12
5.4. 在 FTP 服务器中提供 KICKSTART 文件	13
5.5. 在本地卷中提供 KICKSTART 文件	15
5.6. 在本地卷中生成 KICKSTART 文件以便自动载入	16
第 6 章 为 KICKSTART 安装创建安装源	17
6.1. 安装源的类型	17
6.2. 基于网络安装的端口	17
6.3. 在 NFS 服务器中创建安装源	18
6.4. 使用 HTTP 或 HTTPS 创建安装源	19
6.5. 使用 FTP 创建安装源	20
第 7 章 在具有 KERNEL-64K 的 ARM 上安装 RHEL	23
7.1. 使用 KICKSTART 在 ARM 上安装 KERNEL-64K	23
7.2. 使用命令行在 ARM 上安装 KERNEL-64K	23
第 8 章 启动 KICKSTART 安装	25
8.1. 手动启动 KICKSTART 安装	25
8.2. 使用 PXE 自动启动 KICKSTART 安装	25
8.3. 使用本地卷自动启动 KICKSTART 安装	26
第 9 章 安装过程中的控制台和日志记录	28
第 10 章 维护 KICKSTART 文件	29
10.1. 安装 KICKSTART 维护工具	29
10.2. 验证 KICKSTART 文件	29
部分 II. 从 CONTENT DELIVERY NETWORK 和 SATELLITE 注册并安装 RHEL	30
第 11 章 使用 KICKSTART 从 CDN 注册并安装 RHEL	31
11.1. 从 CDN 注册并安装 RHEL	31

11.2. 从 CDN 验证您的系统注册	33
11.3. 从 CDN 取消注册您的系统	34
第 12 章 使用 KICKSTART 在 SATELLITE 中注册并安装 RHEL	35
12.1. 从 SATELLITE 注册并安装 RHEL	35
12.2. 从 SATELLITE 取消注册您的系统	37
部分 III. 高级配置选项	39
第 13 章 配置系统目的	40
13.1. 概述	40
13.2. 在 KICKSTART 文件中配置系统目的	41
13.3. 其他资源	42
第 14 章 在安装过程中更新驱动程序	43
14.1. 概述	43
14.2. 驱动程序更新的类型	43
14.3. 准备驱动程序更新	44
14.4. 执行自动驱动程序更新	44
14.5. 执行驱动程序更新	45
14.6. 执行手动驱动程序更新	45
14.7. 禁用驱动程序	46
第 15 章 引导带有 UEFI 安全引导的 BETA 系统	47
15.1. UEFI 安全引导和 RHEL BETA 版本	47
15.2. 为 UEFI 安全引导添加 BETA 公钥	47
15.3. 删除 BETA 公钥	47
部分 IV. KICKSTART 参考	49
附录 A. KICKSTART 脚本文件格式参考	50
A.1. KICKSTART 文件格式	50
A.2. KICKSTART 中的软件包选择	50
A.3. KICKSTART 文件中的脚本	54
A.4. KICKSTART 错误处理部分	58
A.5. KICKSTART 附加组件部分	59
附录 B. KICKSTART 命令和选项参考	60
B.1. KICKSTART 的修改	60
B.2. 用于安装程序配置和流量控制的 KICKSTART 命令	61
B.3. KICKSTART 命令进行系统配置	71
B.4. KICKSTART 命令用于网络配置	82
B.5. 用于处理存储的 KICKSTART 命令	87
B.6. RHEL 安装程序提供的附加组件的 KICKSTART 命令	111
B.7. ANACONDA 中使用的命令	114
B.8. KICKSTART 命令进行系统恢复	115

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. 点顶部导航栏中的 **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**。

第 1 章 简介

Red Hat Enterprise Linux 9 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。它可部署为虚拟机监控程序及云供应商环境，也可以部署到物理构架中，以便您的应用程序可在主要硬件构架平台中利用创新。

1.1. 支持的构架

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



注意

有关 IBM Power Servers 的安装说明，请参阅 [IBM 安装文档](#)。要确定您的系统支持安装 RHEL，请参阅 <https://catalog.redhat.com> 和 <https://access.redhat.com/articles/rhel-limits>。

1.2. 安装术语

这部分论述了 Red Hat Enterprise Linux 安装术语。同一概念可能会使用不同的术语，具体要看上游或下游来源。

Anaconda: Fedora、Red Hat Enterprise Linux 以及其变体使用的操作系统安装程序。Anaconda 是一组 Python 模块和脚本，它带有其他文件，比如 Gtk widgets（使用 C）、systemd 单元以及 dracut 程序库。它们一起形成允许用户设置结果（target）系统参数的工具。在这个文档中，**安装程序** 指的是 Anaconda 的安装方面。

第 2 章 安装方法

根据具体要求，您可以使用几种方法安装 Red Hat Enterprise Linux。查看以下部分以确定符合您的要求的最佳安装方法。

2.1. 可用安装方法

您可以使用以下任一方法安装 Red Hat Enterprise Linux：

- 基于 GUI 的安装
- 系统或云镜像安装
- 高级安装

基于 GUI 的安装

您可以从以下基于 GUI 的安装方法中选择：

- **使用客户门户网站中的 ISO 镜像安装 RHEL**：通过从客户门户网站下载 **DVD ISO** 镜像文件安装 Red Hat Enterprise Linux。系统注册在 GUI 安装完成后进行。GUI 和 Kickstart 支持这个安装方法。
- **从 Content Delivery Network 注册并安装 RHEL**：注册您的系统、附加订阅并从 Content Delivery Network (CDN) 安装 Red Hat Enterprise Linux。这个安装方法支持 **Boot ISO** 和 **DVD ISO** 镜像文件，但推荐使用 **Boot ISO** 镜像文件，因为引导 ISO 镜像文件的安装源默认为 CDN。注册系统后，安装程序会从 CDN 下载并安装软件包。Kickstart 支持这个安装方法。
- **使用 VNC 执行远程 RHEL 安装**：RHEL 安装程序提供两种虚拟网络计算(VNC)安装模式：Direct 和 Connect。连接建立后，这两种模式没有区别。您选择的模式取决于您的环境。
- **使用 PXE 从网络安装 RHEL**：使用预启动执行环境(PXE)进行网络安装，您可以将 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。

系统或云镜像安装

您只能在虚拟和云环境中使用系统或云镜像安装方法。要执行系统或基于云镜像的安装，请使用 Red Hat Image Builder（红帽镜像构建器）。镜像构建器创建 Red Hat Enterprise Linux 的自定义系统镜像，包括用于云部署的系统镜像。

有关使用镜像构建器安装 RHEL 的更多信息，请参阅 [制作自定义的 RHEL 系统镜像](#)。

高级安装

您可以从以下高级安装方法中选择：

- **使用 Kickstart 执行自动 RHEL 安装**：Kickstart 是一个自动化过程，它可以帮助您通过在文件中指定所有要求和配置来安装操作系统。Kickstart 文件包含 RHEL 安装选项，例如时区、磁盘分区或要安装的软件包。提供准备好的 Kickstart 文件来完成安装，而无需用户干预。这在一次在大量系统上部署 Red Hat Enterprise Linux 非常有用。
- **从 Content Delivery Network 注册并安装并注册 RHEL**：通过 Content Delivery Network (CDN) 在不同架构中安装 Red Hat Enterprise Linux。注册是在从 CDN 下载并安装软件包前执行的。图形用户界面和 Kickstart 支持这个安装方法。

部分 I. 使用 KICKSTART 执行自动安装

第 3 章 KICKSTART 安装基础

以下提供了有关 Kickstart 的基本信息，以及如何使用它自动安装 Red Hat Enterprise Linux。

3.1. KICKSTART 安装是什么

Kickstart 提供了一种部分自动化 RHEL 安装过程的方法。

Kickstart 文件包含一些或所有 RHEL 安装选项。例如，设置时区，如何对驱动器进行分区，或者应该安装哪些软件包。提供准备好的 Kickstart 文件可进行安装而无需用户干预。这在同时大量部署 Red Hat Enterprise Linux 时特别有用。

Kickstart 文件还提供有关软件选择的更多选项。当使用图形安装界面手动安装 Red Hat Enterprise Linux 时，软件选择仅限于预定义环境和附加组件。Kickstart 文件也允许您安装或删除独立软件包。

Kickstart 文件可保存在单一服务器系统中，并在安装过程中由独立计算机读取。这个安装方法支持使用单一 Kickstart 文件在多台机器中安装 Red Hat Enterprise Linux，使其成为网络和系统管理员的理想选择。

所有执行它们的 Kickstart 脚本和日志文件都存储在新安装系统的 `/tmp` 目录中，以帮助调试安装问题。用于安装的 kickstart 以及 Anaconda 生成的输出 kickstart 存储在目标系统的 `/root` 中，kickstart scriptlet 执行的日志存储在 `/var/log/anaconda` 中。



注意

在以前的 Red Hat Enterprise Linux 版本里，可以使用 Kickstart 来升级系统。从 Red Hat Enterprise Linux 7 开始，删除了这个功能，系统升级是由特殊工具处理的。有关升级到 Red Hat Enterprise Linux 9 的详情，请参阅 [从 RHEL 8 升级到 RHEL 9](#) 和 [采用 RHEL 的注意事项](#)。

3.2. 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 [使 Kickstart 文件可供安装程序使用](#)。
3. 创建用来开始安装的引导介质。请参阅 [创建安装介质](#) 和 [使用 PXE 准备从网络安装](#)。
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 4 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 9 安装转换 Red Hat Enterprise Linux 8 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。

请注意，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。

4.1. 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. 在下拉菜单中选择 **Red Hat Enterprise Linux 9** 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。

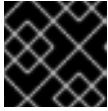
4.2. 执行手动安装来创建 KICKSTART 文件

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 [执行标准的 RHEL 9 安装](#)。
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。



重要

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。

其他资源

- [执行标准的 RHEL 9 安装](#)

4.3. 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转换 RHEL Kickstart 文件的详情，请参考 <https://access.redhat.com/labs/kickstartconvert/>。

4.4. 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 5 章 为安装程序提供 KICKSTART 文件

以下提供了有关在目标系统中让安装程序使用 Kickstart 文件的信息。

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

其他资源

- [安全网络](#)

5.2. 在 NFS 服务器中提供 KICKSTART 文件

这个步骤描述了如何在 NFS 服务器中保存 Kickstart 脚本。这个方法可让您使用单一源安装多个系统而无需为 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 9 服务器具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。

步骤

1. 作为 root 运行以下命令安装 **nfs-utils** 软件包：

```
# dnf 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
```

4. 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 (*)。有关这一字段的格式的详细信息，请参见 `exports(5)` 手册页。
使 `/rhel9-install/` 目录对所有客户端只读的一个基本配置是：

```
/rhel9-install *
```

5. 保存 `/etc/exports` 文件并退出文本编辑器。
6. 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是 `myserver.example.com`，并且已在 `/rhel9-install/my-ks.cfg` 中将保存了文件，请指定 `inst.ks=nfs:myserver.example.com:/rhel9-install/my-ks.cfg` 作为安装源引导选项。

其他资源

- [准备使用 PXE 从网络安装](#)

5.3. 在 HTTP 或 HTTPS 服务器中提供 KICKSTART 文件

这个步骤描述了如何在 HTTP 或 HTTPS 服务器中存储 Kickstart 脚本文件。这个方法可让您使用单一源安装多个系统而无需为 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 9 服务器具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。

步骤

1. 要在 HTTP 中存储 Kickstart 文件，请安装 `httpd` 软件包：

```
# dnf install httpd
```

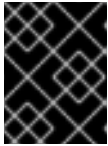
要将 Kickstart 文件存储在 HTTPS 中，请安装 `httpd` 和 `mod_ssl` 软件包：


```
# dnf 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>。



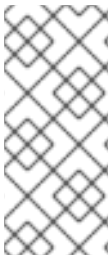
重要

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。



注意

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，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/rhel9-install/my-ks.cfg`，请将 `http://myserver.example.com/rhel9-install/my-ks.cfg` 指定为文件位置。

其他资源

- [部署 Web 服务器和代理](#)
- [配置和使用数据库服务器](#)

5.4. 在 FTP 服务器中提供 KICKSTART 文件

这个步骤描述了如何在 FTP 服务器中保存 Kickstart 脚本。这个方法可让您使用单一源安装多个系统而无需为 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 9 服务器具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。

步骤

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

```
# dnf 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. 另外，还可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
```

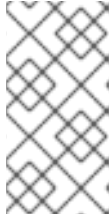
如果在更改 `/etc/vsftpd/vsftpd.conf` 文件前运行该服务，重启该服务以载入经过编辑的文件：

```
# 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。

5.5. 在本地卷中提供 KICKSTART 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。

5.6. 在本地卷中生成 KICKSTART 文件以便自动载入

在要安装的系统中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。

步骤

1. 列出要复制 Kickstart 文件的卷信息。

```
# 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 6 章 为 KICKSTART 安装创建安装源

这部分论述了如何使用包含所需软件仓库和软件包的 DVD ISO 镜像为引导 ISO 镜像创建安装源。

6.1. 安装源的类型

您可选择以下安装源之一用于最小引导镜像：

- **DVD**：将 DVD ISO 镜像刻录到 DVD。DVD 将自动用作安装源（软件包源）。
- **磁盘或者 USB 驱动器**：将 DVD ISO 镜像复制到磁盘，并将安装程序配置为从驱动器安装软件包。如果您使用 USB 驱动器，请在安装开始前确定连接到该系统。安装过程开始后，安装程序无法检测到介质。
 - **磁盘限制**：磁盘上的 DVD ISO 镜像必须在安装程序可挂载的文件系统的分区上。支持的文件系统有：**xf**s、**ext2**、**ext3**、**ext4** 和 **vf**at(FAT32)。



警告

在 Microsoft Windows 系统上，格式化磁盘时使用的默认文件系统是 NTFS。exFAT 文件系统也可用。但是，这些文件系统无法在安装过程中被挂载。如果您要在 Microsoft Windows 上创建磁盘或者 USB 驱动器作为安装源，请确定您将驱动器格式化为 FAT32。请注意：FAT32 文件系统无法存储大于 4GiB 的文件。

在 Red Hat Enterprise Linux 9 中，您可以从本地磁盘上的一个目录启用安装。为此，您需要将 DVD ISO 镜像的内容复制到磁盘上的一个目录中，然后将目录指定为安装源而不是 ISO 镜像。例如：**inst.repo=hd:<device>:<path to the directory>**

- **网络位置**：将 DVD ISO 镜像或安装树（DVD ISO 镜像的提取内容）复制到网络位置，并使用以下协议通过网络执行安装：
 - **NFS**: DVD ISO 镜像位于网络文件系统(NFS)共享中。
 - **HTTPS、HTTP 或者 FTP**: 安装树位于可通过 HTTP、HTTPS 或者 FTP 访问的网络位置。

6.2. 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 6.1. 基于网络安装的端口

使用的协议	打开端口
HTTP	80
HTTPS	443

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

其他资源

- [安全网络](#)

6.3. 在 NFS 服务器中创建安装源

使用这个安装方法从单一源安装多个系统，而无需连接到物理介质。

先决条件

- 您对 Red Hat Enterprise Linux 9 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网络中。
- 您已下载了 Binary DVD 镜像。如需更多信息，请参阅 [下载安装 ISO 镜像](#)。
- 您已使用镜像文件创建了可引导 CD、DVD 或者 USB 设备。如需更多信息，请参阅 [创建安装介质](#)。
- 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 [基于网络安装的端口](#)。

流程

1. 安装 **nfs-utils** 软件包：

```
# dnf install nfs-utils
```

2. 将 DVD ISO 镜像复制到 NFS 服务器上的目录。
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
 - 所有目标系统可用的子网都可以访问 ISO 镜像
 - 要允许任何可通过网络访问 NFS 服务器的系统使用 ISO 镜像，使用星号符号(*)

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 `/rhel9-install/` 目录对所有客户端只读的基本配置是：

```
/rhel9-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` 文件前该服务在运行，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。



注意

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，如果服务器主机名是 **myserver.example.com**，并且已将 ISO 镜像保存在 `/rhel9-install/` 中，请指定 **nfs:myserver.example.com:/rhel9-install/** 作为安装源。

6.4. 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安装源，该安装树是一个包含提取的 DVD ISO 镜像内容和有效 `.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 9 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网络中。
- 您已下载了 Binary DVD 镜像。如需更多信息，请参阅 [下载安装 ISO 镜像](#)。
- 您已使用镜像文件创建了可引导 CD、DVD 或者 USB 设备。如需更多信息，请参阅 [创建安装介质](#)。
- 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 [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，首选启用 TLSv1.3 协议。默认情况下启用 TLSv1.2 (LEGACY)。



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. 创建一个用于挂载 DVD ISO 镜像的合适的目录，例如：

```
# mkdir /mnt/rhel9-install/
```

3. 将 DVD ISO 镜像挂载到目录：

```
# mount -o loop,ro -t iso9660 /image_directory/image.iso /mnt/rhel9-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/rhel9-install/ /var/www/html/
```

这个命令创建包含镜像内容的 `/var/www/html/rhel9-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，如此流程所示，将正确复制 `.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，以及包含 ISO 镜像中文件的目录（相对于 HTTP 服务器的根）。例如，如果您使用 HTTP，服务器主机名为 `myserver.example.com`，并且已将镜像中的文件复制到 `/var/www/html/rhel9-install/`，请指定 `http://myserver.example.com/rhel9-install/` 作为安装源。

其他资源

- [部署不同类型的服务器](#)

6.5. 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安装源，该安装树是一个包含提取的 DVD ISO 镜像内容和有效 `.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 9 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网络中。

- 您已下载了 Binary DVD 镜像。如需更多信息，请参阅 [创建可引导的安装介质](#)。
- 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 [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：为您的配置添加自定义更改。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. 创建一个用于挂载 DVD ISO 镜像的合适的目录，例如：

```
# mkdir /mnt/rhel9-install
```

5. 将 DVD ISO 镜像挂载到目录：

```
# mount -o loop,ro -t iso9660 /image-directory/image.iso /mnt/rhel9-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/rhel9-install
# cp -r /mnt/rhel9-install/ /var/ftp/
```

这个命令创建包含镜像内容的 ***/var/ftp/rhel9-install/*** 目录。请注意，有些复制方法可以跳过有效安装源所需的 **.treeinfo** 文件。为整个目录输入 **cp** 命令，如此流程所示，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/rhel9-install
# find /var/ftp/rhel9-install -type f -exec chmod 444 {} \;
# find /var/ftp/rhel9-install -type d -exec chmod 755 {} \;
```

8. 启动 **vsftpd** 服务：

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

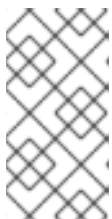
如果在更改 **/etc/vsftpd/vsftpd.conf** 文件前运行该服务，重启该服务以载入经过编辑的文件：

```
# 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。



注意

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 服务器的根）。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/rhel9-install/**，请指定 **ftp://myserver.example.com/rhel9-install/** 作为安装源。

第 7 章 在具有 KERNEL-64K 的 ARM 上安装 RHEL

默认情况下，RHEL 9 与支持 4k 页面大小的内核一起分发。这个 4k 内核对于较小环境中或小的云实例中的有效内存使用足够了，其中由于空间、电源和成本约束，使用 64k 页内核是不现实的。



重要

不建议在不重新安装操作系统的初始引导后在 4k 和 64k 页大小内核之间移动。

7.1. 使用 KICKSTART 在 ARM 上安装 KERNEL-64K

RHEL 提供 ARM64 硬件架构来支持需要大型物理内存配置以获得最佳性能的工作负载。这种大型内存配置需要使用大型 MMU 页大小(64k)。

安装 RHEL 9 时，您可以选择 **kernel-64k** 软件包来安装具有支持 64k 页大小内核的 RHEL。

流程

- 在 kickstart 文件的 **%packages** 部分中添加以下软件包列表：

```
%packages
kernel-64k
-kmod-kvdo
-vdo
-kernel
%end
```

验证步骤

- 要验证页大小，在安装完成并重新引导系统后，打开终端并运行：

```
$ getconf PAGESIZE
65536
```

输出 **65536** 表示 64k 内核正在使用。

- 要验证是否启用了交换分区，请输入：

```
$ free
              total        used        free      shared  buff/cache   available
Mem:      35756352    3677184    34774848     25792     237120    32079168
Swap:      6504384           0     6504384
```

total 和 free 列非零，这表示交换被成功启用。

7.2. 使用命令行在 ARM 上安装 KERNEL-64K

如果您已经安装了带有默认内核的 RHEL（支持 4k 页大小），您可以在安装后使用命令行安装 **kernel-64k**。

流程

- 以 root 用户身份打开终端，并输入：

```
# dnf -y install kernel-64k
```

- 要将 **kernel-64k** 设置为默认，请输入：

```
# k=$(echo /boot/vmlinuz*64k)
# grubby --set-default=$k \
  --update-kernel=$k \
  --args="crashkernel=2G-:640M"
```

- 将系统引导顺序设置为使用 RHEL 作为默认选项。

- 获取当前的引导顺序。例如：

```
# efibootmgr
BootCurrent: 0000
Timeout: 5 seconds
BootOrder: 0003,0004,0001,0000,0002,0005
Boot0000\* Red Hat Enterprise Linux
```

- 将引导顺序设置为优先 RHEL。例如，对于上一步中的输出，请使用以下命令：

```
# efibootmgr -o 0000,0001,0002,0003,0004,0005
```

- 重启系统：

```
# reboot
```

- 可选：重启后，删除 4k 内核：

```
# dnf erase kernel
```

当您使用 **yum update** 命令在以后更新内核时，意外保留这两个版本可能会使 4k 内核成为默认。

验证

- 要验证页大小，请打开终端并以任何用户身份运行以下命令：

```
$ getconf PAGESIZE
65536
```

输出 **65536** 表示 64k 内核正在使用。

- 要验证是否启用了 swap，请输入：

```
$ free
total    used    free   shared  buff/cache  available
Mem:    35756352  3677184  34774848   25792    237120   32079168
Swap:   6504384      0   6504384
```

total 和 free 列非零，这表示交换被成功启用。

第 8 章 启动 KICKSTART 安装

您可以以多种方式启动 Kickstart 安装：

- 输入安装程序引导菜单并指定选项，包括 Kickstart 文件。
- 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。

如何在以下部分中了解如何执行这些方法。

8.1. 手动启动 KICKSTART 安装

这部分论述了如何手动启动 Kickstart 安装，这意味着需要一些用户互动（在 **boot:** 提示中添加引导选项）。引导安装系统时，使用引导选项 **inst.ks=location**，使用 Kickstart 文件的位置替换 location。指定引导选项和引导提示格式的具体方法取决于系统的架构。详情请查看 [RHEL 安装程序引导选项 指南](#)。

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。

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 Kickstart 文件或者所需库位于网络中，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. 添加 **inst.ks=** 引导选项以及 Kickstart 文件的位置。
 - c. 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。
3. 通过确认您添加的引导选项启动安装。

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



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。

8.2. 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。

这个步骤只是一般参考，具体步骤根据您的系统架构有所不同，且不是所有选项都可用于所有架构（例如：您无法在 64 位 IBM Z 上使用 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。

步骤

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：

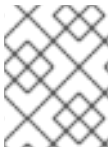
```
append initrd=initrd.img inst.ks=http://10.32.5.1/mnt/archive/RHEL-9/9.x/x86_64/kickstarts/ks.cfg
```

- 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-9/9.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。

8.3. 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。

第 9 章 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。



注意

一般来说，除非需要诊断安装问题，不需要离开默认的图形安装环境。

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**。



注意

如果选择文本模式安装，会在虚拟控制台 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 9.1. 可用的 **tmux** 窗口

快捷键	内容
Ctrl+b 1	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。
Ctrl+b 2	有 root 权限的互动 shell 提示符。
Ctrl+b 3	安装日志；显示信息保存在 /tmp/anaconda.log 中。
Ctrl+b 4	存储日志；显示与存储设备和配置相关的消息，保存在 /tmp/storage.log 中。
Ctrl+b 5	程序日志；显示安装过程中执行的实用程序的信息，保存在 /tmp/program.log 中。

第 10 章 维护 KICKSTART 文件

您可以在 Kickstart 文件中自动运行检查。通常，您要验证一个新的或已存在的 Kickstart 文件是否有效。

10.1. 安装 KICKSTART 维护工具

要使用 Kickstart 维护工具，您必须安装包含它们的软件包。

流程

- 安装 `pykickstart` 软件包：

```
# dnf install pykickstart
```

10.2. 验证 KICKSTART 文件

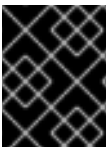
使用 `ksvalidator` 命令行工具验证 Kickstart 文件是否有效。当您对 Kickstart 文件进行广泛更改时，这非常有用。在 `ksvalidator` 命令中使用 `-v RHEL9` 选项来认可 RHEL9 类的新命令。

流程

- 在 Kickstart 文件中运行 `ksvalidator`：

```
$ ksvalidator -v RHEL9 /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。

其他资源

- `ksvalidator(1)` man page

部分 II. 从 CONTENT DELIVERY NETWORK 和 SATELLITE 注册 并安装 RHEL

第 11 章 使用 KICKSTART 从 CDN 注册并安装 RHEL

本节介绍如何使用 Kickstart 注册您的系统、附加 RHEL 订阅并从 Red Hat Content Delivery Network(CDN)安装。

11.1. 从 CDN 注册并安装 RHEL

使用 **rhsm** Kickstart 命令（这个命令支持 **syspurpose** 命令和 Red Hat Insights）注册您的系统、附加 RHEL 订阅并从 Red Hat Content Delivery Network(CDN)安装。在注册系统时，**rhsm** Kickstart 命令不需要使用自定义 **%post** 脚本。

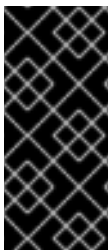


重要

Boot ISO 和 **DVD ISO** 镜像文件支持 CDN 功能。但建议您使用 **引导 ISO** 镜像文件作为引导 ISO 镜像文件的默认 CDN 安装源。

先决条件

- 系统连接到可以访问 CDN 的网络。
- 您已创建了一个 Kickstart 文件，并使其可用于可移动介质、磁盘或使用 HTTP (S)、FTP 或者 NFS 服务器的网络位置上的安装程序。
- Kickstart 文件位于要安装的系统可访问的位置。
- 您已创建了用来开始安装并让安装源对安装程序可用的引导介质。



重要

- 系统注册后的安装源存储库依赖于系统的引导方式。如需更多信息，请参阅 [执行标准的 RHEL 9 安装](#) 文档中的 [系统注册后的安装源存储库](#) 部分。
- Kickstart 文件中不需要仓库配置，因为您的订阅管理系统可访问的 CDN 子集和库。

流程

1. 打开 Kickstart 文件。
2. 编辑该文件，将 **rhsm** Kickstart 命令及其选项添加到该文件中：

机构（必需）

输入机构 id。例如：

```
--organization=1234567
```



注意

出于安全考虑，在从 CDN 注册和安装时 Kickstart 不支持使用红帽用户名和密码帐户详情。

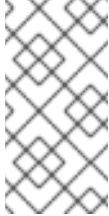
激活码（必需）

输入激活码。只要在您的订阅中进行了注册，您可以输入多个激活码。例如：

```
--activation-key="Test_key_1" --activation-key="Test_key_2"
```

Red Hat Insights (可选)

将目标系统连接到 Red Hat Insights。



注意

Red Hat Insights 是一个软件即服务(SaaS)服务(SaaS),它提供对注册的红帽系统的持续分析,以主动识别跨物理、虚拟和云环境以及容器部署的安全性、性能和稳定性。与使用安装程序 GUI 的手动安装不同,在使用 Kickstart 时,默认情况下不启用到 Red Hat Insights 的连接。

例如：

```
--connect-to-insights
```

HTTP 代理 (可选)

设置 HTTP 代理。例如：

```
--proxy="user:password@hostname:9000"
```



注意

只有主机名是必须的。如果需要代理在没有身份验证的默认端口中运行,则选项为：**--proxy="hostname"**

系统目的 (可选)

使用以下命令设置系统目的的角色、SLA 和用法：

```
subscription-manager syspurpose role --set="Red Hat Enterprise Linux Server" --
sla="Premium" --usage="Production"
```

示例

以下示例显示了包含所有 **rhsm** Kickstart 命令选项的最小 Kickstart 文件。

```
graphical
lang en_US.UTF-8
keyboard us
rootpw 12345
timezone America/New_York
zerombr
clearpart --all --initlabel
autopart
syspurpose --role="Red Hat Enterprise Linux Server" --sla="Premium" --
usage="Production"
rhsm --organization="12345" --activation-key="test_key" --connect-to-insights --
proxy="user:password@hostname:9000"
reboot
```

```
%packages
vim
%end
```

3. 保存 Kickstart 文件并启动安装过程。

其他资源

- [配置系统目的](#)
- [启动 Kickstart 安装](#)
- [Red Hat Insights 产品文档](#)
- [了解激活码](#)
- 有关为订阅管理器设置 HTTP 代理的详情，请参考 **subscription-manager** 手册页中的 **PROXY CONFIGURATION** 部分。

11.2. 从 CDN 验证您的系统注册

使用这个步骤验证您的系统是否在 CDN 中注册。

先决条件

- 您已完成了注册和安装过程，如 [使用 CDN 注册和安装](#) 中所述。
- 您已启动 Kickstart 安装，如 [启动 Kickstart 安装](#) 所述。
- 安装的系统已重启，并打开终端窗口。

步骤

1. 在终端窗口中以 **root** 用户身份登录并验证注册：

```
# subscription-manager list
```

输出会显示附加的订阅详情，例如：

```
Installed Product Status

Product Name: Red Hat Enterprise Linux for x86_64
Product ID: 486
Version: X
Arch: x86_64
Status: Subscribed
Status Details
Starts: 11/4/2019
Ends: 11/4/2020
```

2. 要查看详细报告，请运行以下命令：

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

11.3. 从 CDN 取消注册您的系统

使用这个流程从 Red Hat CDN 取消注册您的系统。

先决条件

- 您已完成了注册和安装过程，如 [从 CDN 注册和安装 RHEL 所述](#)。
- 您已启动 Kickstart 安装，如 [启动 Kickstart 安装](#) 所述。
- 安装的系统已重启，并打开终端窗口。

步骤

- 在终端窗口中以 **root** 用户身份登录并取消注册：

```
# subscription-manager unregister
```

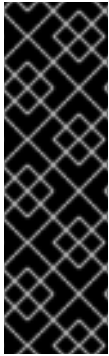
从系统中取消注册附加的订阅，并删除到 CDN 的连接。

第 12 章 使用 KICKSTART 在 SATELLITE 中注册并安装 RHEL

本节介绍如何使用 Kickstart 注册您的系统、附加 RHEL 订阅并从 Red Hat Satellite 安装。

12.1. 从 SATELLITE 注册并安装 RHEL

此流程描述了如何使用 **rhsm** Kickstart 命令从 Satellite 实例注册您的系统、附加 RHEL 订阅并从 Satellite 实例安装。它还介绍了如何配置系统目的并将系统连接到 Red Hat Insights。在注册系统时，**rhsm** Kickstart 命令不需要使用自定义 **%post** 脚本。



重要

- 引导 ISO 和 DVD ISO 镜像文件支持 Satellite 安装。但是，建议您使用引导 ISO 镜像文件作为引导 ISO 镜像文件的默认是 Satellite 的安装源。
- 系统注册后的安装源存储库依赖于系统的引导方式。如需更多信息，请参阅 [系统注册后安装源存储库](#)。
- Kickstart 文件中不需要存储库配置，因为您的订阅管理系统可访问的 satellite 托管的软件仓库。

先决条件

- 系统连接到可以访问 Satellite 实例的网络。
- Red Hat Satellite 服务器版本为 6.11 或更高
- 您已创建了一个 Kickstart 文件，并使其可用于可移动介质、磁盘或使用 HTTP (S)、FTP 或者 NFS 服务器的网络位置上的安装程序。
- Kickstart 文件位于要安装的系统可访问的位置。
- 您有一个机构 ID、激活码和您要使用的 Satellite 6.11 实例的 URL。
- 您已启用、同步并将所需的 BaseOS 和 AppStream RPM 存储库添加到 content-view 中。
- 激活码将发行版本设置为 9.x，并在其中选择了相关的 content-view。

步骤

1. 打开 Kickstart 文件。
2. 编辑该文件，将 **rhsm** Kickstart 命令及其选项添加到该文件中：

机构（必需）

输入机构 id。例如：

```
--organization=1234567
```



注意

出于安全考虑，在从 satellite 注册和安装时 Kickstart 不支持红帽用户名和密码帐户详情。

激活码（必需）

输入激活码。只要在您的订阅中进行了注册，您可以输入多个激活码。例如：

```
--activation-key="Test_key_1" --activation-key="Test_key_2"
```

Red Hat Insights（可选）

将目标系统连接到 Red Hat Insights。



注意

Red Hat Insights 是一个软件即服务(SaaS)服务(SaaS),它提供对注册的红帽系统的持续分析,以主动识别跨物理、虚拟和云环境以及容器部署的安全性、性能和稳定性。与使用安装程序 GUI 的手动安装不同,在使用 Kickstart 时,默认情况下不启用到 Red Hat Insights 的连接。

例如：

```
--connect-to-insights
```

HTTP 代理（可选）

设置 HTTP 代理。例如：

```
--proxy="user:password@hostname:9000"
```



注意

只有主机名是必须的。如果需要代理在没有身份验证的默认端口中运行,则选项为: **--proxy="hostname"**

服务器主机名



注意

服务器主机名不需要 HTTP 协议,如 **nameofhost.com**。

如果要注册到 Satellite 实例,请设置服务器主机名。例如：

```
--server-hostname="nameofhost.com"
```

系统目的（可选）

使用以下命令设置系统目的角色、SLA 和用法：

```
syspurpose --role="Red Hat Enterprise Linux Server" --sla="Premium" --usage="Production"
```

示例

以下示例显示了包含所有 **rhsm** Kickstart 命令选项的最小 Kickstart 文件。


```
graphical
lang en_US.UTF-8
keyboard us
rootpw 12345
timezone America/New_York
zerombr
clearpart --all --initlabel
autopart
syspurpose --role="Red Hat Enterprise Linux Server" --sla="Premium" --
usage="Production"
rhm --organization="12345" --activation-key="test_key" --connect-to-insights --server-
hostname="nameofhost.com" --proxy="user:password@hostname:9000"
reboot
%packages
vim
%end
```

3. 保存 Kickstart 文件并启动安装过程。

验证步骤

安装该系统后，重新引导并打开终端窗口，您可以确认您的系统是否已注册到 satellite：

1. 在终端窗口中以 root 用户身份输入以下命令：

```
# subscription-manager list
Installed Product Status
Product Name: Red Hat Enterprise Linux for x86_64
Product ID: 486
Version: 9
Arch: x86_64
Status: Subscribed
Status Details
Starts: 11/4/2019
Ends: 11/4/2020
```

2. 查看详细报告：

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

其他资源

- [配置系统目的](#)
- [启动 Kickstart 安装](#)
- [Red Hat Insights 产品文档](#)
- [了解使用红帽订阅管理的激活码](#)，要了解有关为订阅管理器设置 HTTP 代理的信息，请参阅 **subscription-manager** 手册页中的 **PROXY CONFIGURATION** 部分。

12.2. 从 SATELLITE 取消注册您的系统

这个步骤描述了如何从 satellite 取消注册您的系统。

先决条件

- 您已完成注册和安装过程。
- 您已启动 Kickstart 安装。
- 安装的系统已重启，并打开终端窗口。

步骤

- 在终端窗口中以 root 用户身份输入以下命令：

```
# subscription-manager unregister
```

从系统中取消注册附加的订阅，并删除到 satellite 的连接。

部分 III. 高级配置选项

第 13 章 配置系统目的

您可以使用系统目的记录 Red Hat Enterprise Linux 9 系统的预期使用。设置系统目的可让授权服务器自动附加最合适的订阅。这部分论述了如何使用 Kickstart 配置系统目的。

优点包括：

- 为系统管理员和商业操作提供深入系统级的信息。
- 降低管理成本以决定系统为什么及其预期目的。
- 改进了 Subscription Manager 自动附加的客户体验，并自动发现并协调系统使用。

13.1. 概述

可以用以下方法之一输入系统目的：

- 创建镜像期间
- 在 GUI 安装过程中，使用 **Connect to Red Hat** 屏幕注册您的系统，并附加您的红帽订阅
- 在使用 **syspurpose Kickstart** 命令时的 Kickstart 安装过程中
- 使用 **subscription-manager** 命令行(CLI)工具安装后

要记录系统的预期目的，您可以配置以下系统目的组件。注册时授权服务器会使用所选值为您的系统附加最合适的订阅。

Role

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

服务等级协议

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

使用

- 生产环境
- 开发/测试
- 灾难恢复

其他资源

- [创建自定义 RHEL 系统镜像](#)

- [执行高级 RHEL 9 安装](#)
- [使用并配置 Red Hat Subscription Manager](#)

13.2. 在 KICKSTART 文件中配置系统目的

按照以下步骤在安装过程中配置系统目的。为此，请在 Kickstart 配置文件中使用 **syspurpose** Kickstart 命令。

尽管系统目的是 Red Hat Enterprise Linux 安装程序的可选功能，但我们强烈建议您将系统目的配置为自动附加最合适的订阅。



注意

您还可以在安装完成后启用系统目的。为此，请使用 **subscription-manager** 命令行工具。**subscription-manager** 工具命令与 **syspurpose** Kickstart 命令不同。

以下操作可用于 **syspurpose** Kickstart 命令：

role

设置系统预期的角色。这个操作使用以下格式：

```
syspurpose --role=
```

分配的角色可以是：

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

SLA

设置系统的预期 SLA。这个操作使用以下格式：

```
syspurpose --sla=
```

分配的 sla 可以是：

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

usage

设置系统的预期用法。这个操作使用以下格式：

```
syspurpose --usage=
```

分配的使用量可以是：

- **Production**

- **Development/Test**
- **Disaster Recovery**

addon

任何额外的层次产品或功能。要添加多个项，重复使用 **--addon** 指定，每次一个分层产品/功能。这个操作使用以下格式：

```
syspurpose --addon=
```

13.3. 其他资源

- 使用 [subscription-manager](#) 命令行工具配置系统目的

第 14 章 在安装过程中更新驱动程序

这部分论述了如何在 Red Hat Enterprise Linux 安装过程中完成驱动程序更新。



注意

这是安装过程的一个可选步骤。红帽建议不要执行驱动程序更新，除非有此需要。

先决条件

- 红帽、您的硬件零售商或者可信的第三方零售商会通知您在 Red Hat Enterprise Linux 安装过程中需要进行驱动程序更新。

14.1. 概述

Red Hat Enterprise Linux 支持很多硬件设备的驱动程序，但可能不支持一些新发布驱动程序。只有在不支持的驱动程序无法完成安装时才应执行驱动程序更新。通常在安装过程中需要更新驱动程序来支持特定的配置。例如：为存储适配器卡安装驱动程序，提供对您的系统存储设备的访问。



警告

驱动程序更新磁盘可能会禁用冲突内核驱动程序。在个别情况下，卸载内核模块可能会导致安装错误。

14.2. 驱动程序更新的类型

红帽、您的硬件零售商或可信第三方以 ISO 镜像文件的形式提供驱动程序更新。您收到 ISO 镜像文件后，请选择驱动程序更新的类型。

驱动程序更新的类型

自动

推荐的驱动程序更新方法；标有 **OEMDRV** 的存储设备（包括 CD、DVD 或者 USB 闪存驱动器）是物理连接到该系统的。如果在安装启动时存在 **OEMDRV** 存储设备，它将被视为驱动程序更新磁盘，且安装程序会自动载入其驱动程序。

Assisted

安装程序会提示您定位驱动程序更新。您可以使用任何带 **OEMDRV** 以外的标签的本地存储设备。在开始安装时会指定 **inst.dd** 引导选项。如果您在没有参数的情况下使用这个选项，安装程序会显示所有连接到该系统的存储设备，并提示您选择包含驱动程序更新的设备。

Manual (手动)

手动指定到驱动程序更新镜像或 RPM 软件包的路径。您可以使用任何带 **OEMDRV** 标签的本地存储设备，或者可从安装系统访问的网络位置。在开始安装时会指定 **inst.dd=location** 引导选项，其中 *location* 是驱动程序更新磁盘或 ISO 镜像的路径。当您指定这个选项时，安装程序会尝试载入在指定位置找到的所有驱动程序更新。使用手动驱动程序更新，您可以指定本地存储设备或者网络位置（HTTP、HTTPS 或者 FTP 服务器）。



注意

- 您可以同时使用 **inst.dd=location** 和 **inst.dd**，其中 *location* 是驱动程序更新磁盘或 ISO 镜像的路径。在这种情况下，安装程序会尝试从位置载入所有可用的驱动程序更新，并提示您选择包含驱动程序更新的设备。

限制

在启用了安全引导技术的 UEFI 系统中，必须使用有效证书为所有驱动程序签名。红帽驱动程序使用红帽的私钥签署，并由内核中对应的公钥验证。如果您载入了额外的独立驱动程序，请确认它们已进行了签名。

14.3. 准备驱动程序更新

这个步骤描述了如何在 CD 和 DVD 中准备驱动程序更新。

先决条件

- 您已收到来自红帽、您的硬件厂商或可信的第三方供应商提供的驱动程序更新 ISO 镜像。
- 您已将驱动程序更新 ISO 镜像刻录到 CD 或者 DVD。



警告

如果 CD 或者 DVD 中只有一个以 **.iso** 结尾的 ISO 镜像文件，则刻录过程将无法成功。有关如何将 ISO 镜像刻录到 CD 或者 DVD 的说明，请查看您的系统刻录软件文档。

流程

- 将驱动程序更新 CD 或者 DVD 插入系统的 CD/DVD 驱动器中，并使用系统的文件管理器工具浏览。
- 验证单个文件 **rhdd3** 是否可用。**rhdd3** 是一个签名文件，其中包含驱动程序说明以及名为 **rpms** 的目录，其中包含各种架构的实际驱动程序 RPM 软件包。

14.4. 执行自动驱动程序更新

这个步骤描述了如何在安装过程中执行自动驱动程序更新。

先决条件

- 您已将驱动程序更新镜像放在带有 **OEMDRV** 标签的标准磁盘分区中，或将 **OEMDRV** 驱动程序更新镜像刻录到 CD 或者 DVD 中。在驱动程序更新过程中可能无法访问高级存储，比如 RAID 或者 LVM 卷。
- 您已将块设备与 **OEMDRV** 卷标签连接到您的系统，或者在开始安装过程前将准备的 CD 或 DVD 插入系统的 CD/DVD 驱动器中。

步骤

- 完成先决条件步骤后，安装程序在系统安装过程中启动并安装时会自动载入驱动程序。

14.5. 执行驱动程序更新

这个步骤描述了如何在安装过程中执行驱动程序更新。

先决条件

- 您已将没有 **OEMDRV** 卷标签的块设备连接到您的系统，并将驱动程序磁盘镜像复制到这个设备中，或者您已准备了一个驱动程序更新 CD 或者 DVD，并在开始安装进程前将其插入到系统的 CD 或者 DVD 驱动器中。



注意

如果您将 ISO 镜像文件刻录到 CD 或者 DVD，但没有 **OEMDRV** 卷标签，则可以使用没有参数的 **inst.dd** 选项。安装程序提供从 CD 或者 DVD 中扫描和选择驱动程序的选项。在这种情况下，安装程序不会提示您选择驱动程序更新 ISO 镜像。另一个情况是使用 CD 或者 DVD 与 **inst.dd=location** 引导选项；这允许安装程序自动扫描 CD 或者 DVD 进行驱动程序更新。如需更多信息，请参阅[执行手动驱动程序更新](#)。

步骤

1. 在引导菜单窗口中按键盘中的 **Tab** 键显示引导命令行。
2. 在命令行中附加 **inst.dd** 引导选项，然后按 **Enter** 键以执行引导过程。
3. 在菜单中选择本地磁盘分区或者 CD 或者 DVD 设备。安装程序扫描 ISO 文件或驱动程序更新 RPM 软件包。
4. 可选：选择驱动程序更新 ISO 文件。



注意

如果选择的设备或者分区包含驱动程序更新 RPM 软件包而不是 ISO 镜像文件（例如：包含驱动程序更新 CD 或者 DVD 的光驱），则不需要这一步。

5. 选择所需驱动程序。
 - a. 使用键盘上的数字键切换驱动器选择。
 - b. 按 **c** 键安装所选驱动程序。载入所选驱动程序并开始安装过程。

14.6. 执行手动驱动程序更新

这个步骤描述了如何在安装过程中手动驱动程序更新。

先决条件

- 您已将驱动程序更新 ISO 镜像文件放在 USB 闪存驱动器或 web 服务器中，并将其连接到您的计算机。

流程

1. 在引导菜单窗口中按键盘中的 **Tab** 键显示引导命令行。
2. 在命令行中附加 **inst.dd=location** 引导选项，其中 *location* 是驱动程序更新的路径。通常，镜像文件位于 web 服务器上，例如 `http://server.example.com/dd.iso` 或在 USB 闪存驱动器上，例如 `/dev/sdb1`。您还可以指定包含驱动程序更新的 RPM 软件包，例如 `http://server.example.com/dd.rpm`。
3. 按 **Enter** 执行引导过程。指定位置中的可用驱动程序会自动载入并启动安装过程。

其他资源

- [inst.dd 引导选项](#)

14.7. 禁用驱动程序

这个步骤描述了如何禁用发生故障的驱动程序。

先决条件

- 您已引导安装程序引导菜单。

流程

1. 在引导菜单中，按键盘中的 **Tab** 键显示引导命令行。
2. 将 **modprobe.blacklist=driver_name** 引导选项附加到命令行。
3. 使用您要禁用的驱动程序或驱动程序名称替换 *driver_name*，例如：

```
modprobe.blacklist=ahci
```

使用 **modprobe.blacklist=** 引导选项禁用的驱动程序在安装的系统中保持禁用状态，并出现在 `/etc/modprobe.d/anaconda-blacklist.conf` 文件中。

4. 按 **Enter** 执行引导过程。

第 15 章 引导带有 UEFI 安全引导的 BETA 系统

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。

15.1. 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，要将 UEFI 安全引导与 Beta 版本一起使用，请使用机器所有者密钥(MOK)设备将红帽 Beta 公钥添加到您的系统中。

15.2. 为 UEFI 安全引导添加 BETA 公钥

本节包含关于如何为 UEFI 安全引导添加 Red Hat Enterprise Linux Beta 公钥的信息。

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，系统重启后仍然禁用了安全引导。
- 您已登录到系统，**Initial Setup** 窗口中的任务已完成。

步骤

1. 开始在系统的机器所有者密钥(MOK)列表中注册红帽 Beta 公钥：

```
# 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 密钥管理工具在系统启动期间启动。
4. 选择 **Enroll MOK**。
5. 选择 **Continue**。
6. 选择 **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。

15.3. 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(GA)版本或不同的操作系统，请删除 Beta 公钥。

这流程骤描述了如何删除 Beta 公钥。

步骤

1. 开始从系统的机器所有者密钥(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 密钥管理工具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。密钥从系统的固件中删除。
7. 选择 **Reboot**。

部分 IV. KICKSTART 参考

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 kickstart 安装过程中需要用户互动，只在位完成安装过程必须进行更新的窗口中输入。输入 spoke 可能会导致重置 kickstart 配置。重置配置将特别适用于输入 Installation Destination 窗口后与存储相关的 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

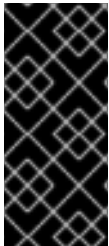
使用 **%packages** 命令启动一个 Kickstart 部分，该部分描述了要安装的软件包。**%packages** 部分必须以 **%end** 命令结尾。

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。如需环境和组群列表，请参阅 Red Hat Enterprise Linux 9 安装 DVD 中的软件仓库 **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 9 所需的软件包。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](#)。请注意：**Initial Setup** 无法在使用 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。Red Hat Enterprise Linux 9 安装 DVD 中的 **repository/repodata/*-comps-repository.architecture.xml** 文件包括了所有可用的环境。

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。

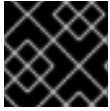
在 Red Hat Enterprise Linux 9 中，模块仅存在于 AppStream 软件仓库中。要列出可用的模块，请在安装的 Red Hat Enterprise Linux 9 系统上使用 **dnf module list** 命令。

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。

其他资源

- [使用 DNF 工具管理软件](#)

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。

--inst-langs

指定要安装的语言列表。请注意，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。

--exclude-weakdeps

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 DNF 尝试下载软件包的次数（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 DNF 配置。

--timeout=

以秒为单位设置 DNF 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 DNF 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
```

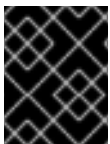
--nodefaults

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。

请注意，某些软件包组（如 **Scientific Support**）没有任何强制或默认软件包 - 仅包括可选软件包。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。在使用 **--nodefaults** 安装时只会安装必需的软件包，并在安装的系统后安装过程中安装可选软件包。

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。

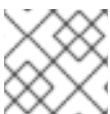


重要

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 中的 Python 的详情，请查看 [安装和使用动态编程语言](#) 中的 [Python 简介](#) 部分。

--erroronfail

显示错误并在脚本失败时暂停安装。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启动的状态。您可以使用 **inst.nokill** 选项来调试脚本。

--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到 `/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。

注意：**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
```

请注意，您可以有多个 **%pre-install** 部分，它们具有相同或不同的解释器。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`。

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 中的 Python 的详情，请查看 [安装和使用动态编程语言](#) 中的 [Python 简介](#) 部分。

--erroronfail

显示错误并在脚本失败时暂停安装。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启动的状态。您可以使用 **inst.nokill** 选项来调试脚本。

--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。

请注意，在执行 **%post** 部分的过程中，仍然必须插入安装介质。

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**。

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 中的 Python 的详情，请查看 [安装和使用动态编程语言](#) 中的 *Python 简介* 部分。

--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

显示错误并在脚本失败时暂停安装。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启动的状态。您可以使用 **inst.nokill** 选项来调试脚本。

--log=

将脚本的输出记录到指定的日志文件中。请注意，无论您是否使用 **--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。请注意，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.4. KICKSTART 错误处理部分

从 Red Hat Enterprise Linux 7 开始，Kickstart 安装可以包含安装程序遇到致命错误时运行的自定义脚本。例如：已经为安装请求的软件包中有一个错误，指定时无法启动 VNC，或者在扫描存储设备时出错。发生此类错误后安装无法继续。安装程序会按照在 Kickstart 文件中提供的顺序运行所有 **%onerror** 脚本。此外，如果出现回溯，将运行 **%onerror** 脚本。

每个 **%onerror** 脚本都需要以 **%end** 结尾。

错误处理部分接受以下选项：

--erroronfail

显示错误并在脚本失败时暂停安装。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启动的状态。您可以使用 **inst.nokill** 选项来调试脚本。

--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`。

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 中的 Python 的详情，请查看 [安装和使用动态编程语言](#) 中的 *Python 简介* 部分。

--log=

将脚本的输出记录到指定的日志文件中。

A.5. 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

这个参考是 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

以下小节描述了 Red Hat Enterprise Linux 9 中的 Kickstart 命令和选项的更改。

B.1.1. 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` 软件包以提供兼容性层。

B.1.2. 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.1.3. 弃用的 Kickstart 命令和选项

以下 Kickstart 命令和选项已在 9 中弃用。

- `timezone --ntpservers` - 使用 `timesource` 命令
- `timezone --nntp`
- `logging --level`
- `%packages --excludeWeakdeps` - 使用 `--exclude-weakdeps`
- `%packages --instLangs` - 使用 `--inst-langs` 替代
- `%anaconda`
- `pwpolicy` - 改为使用 Anaconda 配置文件
- `syspurpose` - 改为使用 `subscription-manager syspurpose`

如果只列出具体的选项，则基础命令及其它选项仍可用且没有弃用。在 Kickstart 文件中使用已弃用的命令会在日志中显示警告信息。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。

B.1.4. 删除的 Kickstart 命令和选项

以下 Kickstart 命令和选项已在 9 中完全删除。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**
- **autostep**

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。

语法

```
cdrom
```

备注

- 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,ostreesetup,rhsm**, 或 **url**，除非内核命令行上指定了 **inst.repo** 选项。

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。

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 系统中很有用。

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。

Syntax

```
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。请注意，该分区必须指定为完整路径（例如 **/dev/sdb1**），而不只是分区名称（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。

备注

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*，并使用特定名称替换 *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。

Syntax

```
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。

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。这个模式除默认模式外还启用了 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。

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。

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。必须使用安装程序可挂载的文件系统格式化驱动器：**ext2**、**ext3**、**ext4**、**vfat**，或 **xfs**。

Syntax

```
harddrive OPTIONS
```

选项

- **--partition=** - 要从中安装的分區（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**，除非内核命令行上指定了 **inst.repo** 选项。

B.2.9. 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。

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**、**ostreosetup**、**rhsm**，或 **url**，除非内核命令行上指定了 **inst.repo** 选项。

B.2.10. 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。



注意

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。

B.2.11. 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。

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.2.12. nfs

nfs Kickstart 命令是可选的。它从指定的 NFS 服务器执行安装。

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

- 要实际运行安装，您必须指定 **cdrom,harddrive,hmc,nfs,liveimg,ostreesetup,rhsm**，或 **url**，除非内核命令行上指定了 **inst.repo** 选项。

B.2.13. ostreesetup

ostreesetup Kickstart 命令是可选的。它被用来设置基于 OSTree 的安装。

Syntax

```
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.2.14. 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。

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，请参阅您的硬件文档。
- 这个命令没有选项。

B.2.15. reboot

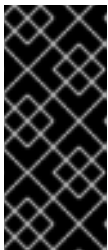
reboot Kickstart 命令是可选的。它指示安装程序在安装成功（没有参数）后重启。通常，Kickstart 会显示信息并等待用户按任意键来重新引导系统。

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。
- 指定 **reboot**，以便在 64 位 IBM Z 中使用命令行模式安装时完全自动安装。

- 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.2.16. rhsm

rhsm Kickstart 命令是可选的。它指示安装程序从 CDN 注册并安装 RHEL。



注意

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。只要在您的订阅中进行了注册，即可多次使用选项（每次使用一个激活码）。
- **--connect-to-insights** - 将目标系统连接到 Red Hat Insights。
- **--proxy=** - 设置 HTTP 代理。
- **--server-hostname=** - 设置要注册的 Satellite 实例主机名。
- 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，您没有使用 **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.2.17. shutdown

shutdown Kickstart 命令是可选的。它会在安装成功完成后关闭系统。

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.2.18. 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。

Syntax

```
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.2.19. 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。

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.2.20. 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。

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
```

备注

- 要实际运行安装，您必须指定 **cdrom**、**harddrive**、**hmc**、**nfs**、**liveimg**、**ostreesetup**、**rhsm**，或 **url**，除非内核命令行上指定了 **inst.repo** 选项。

B.2.21. 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。

Syntax

```
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。

其他资源

- [准备使用 PXE 从网络安装](#)

B.2.22. %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。

Syntax

```
%include path/to/file
```

B.2.23. %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。

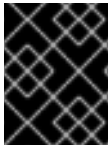
Syntax

```
%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。

Syntax

```
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。

Syntax

```
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。

Syntax

```
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。

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。

Syntax

```
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。格式为使用逗号分开的列表，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。

Syntax

```
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 9 中，模块仅存在于 AppStream 软件仓库中。要列出可用的模块，请在已安装的 Red Hat Enterprise Linux 9 系统上使用带有有效订阅的 **dnf module list** 命令。

其他资源

- [使用 DNF 工具管理软件](#)

B.3.8. repo

repo Kickstart 命令是可选的。它配置额外的 dnf 存储库，可用作软件包安装的来源。您可以添加多个 **repo** 行。

Syntax

```
repo --name=repoid [ --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。此处不支持 dnf 存储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。

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。
- **--allow-ssh** - 如果存在这个选项，root 用户可以使用带密码的 SSH 登录到系统。这个选项只在 RHEL 9.1 及更新的版本中可用。

在 kickstart 安装方法期间向 kickstart 文件中添加以下行，以启用 **基于密码的 SSH root 登录**。选项 **--allow-ssh** 在 RHEL 9.0 中不可用。

```
%post
echo "PermitRootLogin yes" > /etc/ssh/sshd_config.d/01-permitrootlogin.conf
%end
```

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**。

Syntax

```
selinux [--disabled|--enforcing|--permissive]
```

选项

--enforcing

使用默认目标策略启用 SELinux **enforcing**。

--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。

其他资源

- [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。

Syntax

```
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。

备注

- 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。
- 多个服务应该用逗号分开，没有任何空格。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 只启用或禁用服务最多使用第一个空格。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。

Syntax

```
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。

Syntax

```
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。



注意

Red Hat Enterprise Linux 9.0 及更新的版本使您能够通过使 **role**、**service-level**、**usage**，以及 **addons** 子命令在一个 **subscription-manager syspurpose** 模块下可用来管理并显示系统目的的属性。之前，系统管理员使用四个独立的 **syspurpose** 命令之一来管理每个属性。从 RHEL 9.0 开始，这个独立的 **syspurpose** 命令已弃用，并计划在 RHEL 9 后删除。红帽将在当前发行生命周期中将提供对这个功能的 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。

Syntax

```
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。

B.3.15. timezone (必需)

timezone Kickstart 命令是必需的。它设置系统时区。

Syntax

```
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。
- **--nntp** - 禁用 NTP 服务自动启动。这个选项已弃用。
- **--ntpserver=** - 指定用作没有空格的逗号分隔列表的 NTP 服务器列表。这个选项已弃用，改为使用 **timesource** 命令。

备注

在 Red Hat Enterprise Linux 9 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. timesource (可选)

timesource kickstart 命令是可选的。使用它来设置提供时间数据的 NTP、NTS 服务器和池，以及控制系统上是否启用或禁用 NTP 服务。

Syntax

```
timesource [--ntp-server NTP_SERVER | --ntp-pool NTP_POOL | --ntp-disable] [--nts]
```

必填选项

在使用 **timesource** 命令时，必须指定以下选项之一：

- **--ntp-server** - 添加一个 NTP 服务器作为时间源.这个选项只能添加到单个命令中，以添加一个 NTP 时间源服务器。要添加多个源，每次都使用单个 **--ntp-server** 或 **--ntp-pool** 选项添加多个 **timesource** 命令。例如，为 *Europe* 时区添加多个源

```
timezone Europe
timesource --ntp-server 0.rhel.pool.ntp.org
timesource --ntp-server 1.rhel.pool.ntp.org
timesource --ntp-server 2.rhel.pool.ntp.org
```

- **--ntp-pool** - 添加 NTP 服务器池作为时间源.这个选项只能添加一次，以添加单个 NTP 时间源池。重复 **timesource** 命令，以添加多个源。
- **--ntp-disable** - 禁用已安装系统的 NTP 时间源。

可选选项

- **--ntp-disable** 此命令添加的服务器或池使用 NTS 协议。请注意，即使使用 **--ntp-disable**，也可以添加这个选项，但它无效。

备注

- **timezone** 命令中的 **--ntpservers** 选项已弃用。红帽建议使用这个新选项来表达 **timesource** 命令的功能。
- 只有 **timesource** 命令可以将服务器和池标记为使用 **NTS**，而不是普通的 **NTP** 协议。

B.3.17. user

user Kickstart 命令是可选的。它在系统上创建新用户。

Syntax

```
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。
- 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。

B.3.18. xconfig

xconfig Kickstart 命令是可选的。它配置 X 窗口系统。

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 9 不包含 KDE 桌面环境，请不要使用上游 **--defaultdesktop=** 文档。

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。

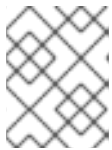
Syntax

```
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，会使用 **inst.ks.device=** Anaconda 引导选项。请注意，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 **--device=**。



重要

网络团队在 Red Hat Enterprise Linux 9 中已弃用。考虑使用网络绑定驱动程序作为替代方案。详情请参阅 [配置网络绑定](#)。

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 *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。Red Hat 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**。

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。

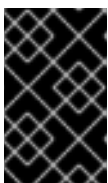
- **--ethtool=** - 指定将传递给 ethtool 程序的网络设备的其他低级别设置。
- **--onboot=** - 是否在引导时启用该设备。
- **--dhcpclass=** - DHCP 类。
- **--mtu=** - 设备的 MTU。
- **--noipv4** - 在这个设备上禁用 IPv4。
- **--noipv6** - 在这个设备上禁用 IPv6。
- **--bondslaves=** - 使用这个选项时，由 **--device=** 选项指定的绑定设备会使用 **--bondslaves=** 选项中定义的辅助设备创建。例如：

```
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**。[有关可用模式和 supported 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
```

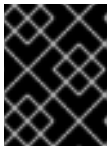
以上命令在 **em1** 设备上创建一个名为 **vlan171** 的虚拟 LAN 接口，其 ID 为 **171**。

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。



重要

网络团队在 Red Hat Enterprise Linux 9 中已弃用。考虑使用网络绑定驱动程序作为替代方案。详情请参阅 [配置网络绑定](#)。

- **--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'}}"
```



重要

网络团队在 Red Hat Enterprise Linux 9 中已弃用。考虑使用网络绑定驱动程序作为替代方案。详情请参阅 [配置网络绑定](#)。

- **--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)** man page 中的 [网桥设置表](#)，或者查看 [网络配置设置规格](#)。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**)。请注意，这个选项独立于 **--device=** 选项 - 即使相同的 **network** 命令也指定了设备名称、**link** 或 **bootif**，也将应用 **--bindto=mac**。

备注

- 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 *执行标准的 RHEL 9 安装* 文档中的 [配置网络和主机名选项](#) 部分。

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。

Syntax

```
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**）格式不保证重启后一致的设备名称，这可能会使一些 Kickstart 命令的使用复杂化。当命令需要设备节点名称时，您可以使用 **/dev/disk** 中的任何项目来作为替代。例如，使用以下设备名称：

```
part / --fstype=xfst --onpart=sda1
```

您可以使用类似以下条目之一：

```
part / --fstype=xfst --onpart=/dev/disk/by-path/pci-0000:00:05.0-scsi-0:0:0:0-part1
```

```
part / --fstype=xfst --onpart=/dev/disk/by-id/ata-ST3160815AS_6RA0C882-part1
```

通过使用此方法，命令始终以相同的存储设备为目标。这在大型存储环境中特别有用。要探索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。有关一致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](#)。

B.5.1. autopart

autopart Kickstart 命令是可选的。它自动创建分区。

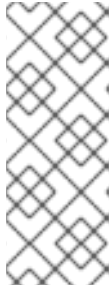
自动创建的分区是：根(/)分区（1 GiB 或更大）、**swap** 分区，以及适合架构的 **/boot** 分区。在足够大的驱动器（50 GiB 及更大）上，这还会创建一个 **/home** 分区。

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**、**xfst** 和 **vfat**。默认的文件系统是 **xfst**。
- **--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，请确保磁盘扇区大小一致。

B.5.2. 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。

Syntax

```
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，使用 **zipl** 引导装载程序的 64 位 IBM Z 系统中的 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** - 安装程序将 Red Hat Enterprise Linux 9 添加到 UEFI 中安装的系统的列表中。它不会将安装的系统添加到引导顺序中。所有现有的引导条目及其顺序都会被保留。



重要

这个选项适用于 Power 和 UEFI 系统。

- **--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 *执行标准的 RHEL 9 安装* 文档中的 [配置引导装载程序](#) 部分。

- **sdX**（或 **/dev/sdX**）格式不保证重启后一致的设备名称，这可能会使一些 Kickstart 命令的使用复杂化。当命令需要设备节点名称时，您可以使用 **/dev/disk** 中的任何项目来作为替代。例如，使用以下设备名称：

```
part / --fstype=xfst --onpart=sda1
```

您可以使用类似以下条目之一：

```
part / --fstype=xfst --onpart=/dev/disk/by-path/pci-0000:00:05.0-scsi-0:0:0:0-part1
```

```
part / --fstype=xfst --onpart=/dev/disk/by-id/ata-ST3160815AS_6RA0C882-part1
```

通过使用此方法，命令始终以相同的存储设备为目标。这在大型存储环境中特别有用。要探索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。有关一致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.5.3. zipl

zipl Kickstart 命令是可选的。它为 64 位 IBM Z 指定 ZIPL 配置。

选项

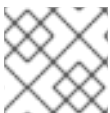
- **--secure-boot** - 如果安装系统支持，则启用安全引导。



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。

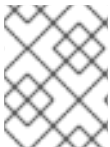
- **--force-secure-boot** - 无条件启用安全引导。



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.5.4. 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。

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。

备注

- **sdX**（或 **/dev/sdX**）格式不保证重启后一致的设备名称，这可能会使一些 Kickstart 命令的使用复杂化。当命令需要设备节点名称时，您可以使用 **/dev/disk** 中的任何项目来作为替代。例如，使用以下设备名称：

```
part / --fstype=xfst --onpart=sda1
```

您可以使用类似以下条目之一：

```
part / --fstype=xfst --onpart=/dev/disk/by-path/pci-0000:00:05.0-scsi-0:0:0:0-part1
```

```
part / --fstype=xfst --onpart=/dev/disk/by-id/ata-ST3160815AS_6RA0C882-part1
```

通过使用此方法，命令始终以相同的存储设备为目标。这在大型存储环境中特别有用。要探索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。有关一致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](#)。

- 如果使用 **clearpart** 命令，则无法在逻辑分区中使用 **part --onpart** 命令。

B.5.5. 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。

Syntax

```
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.5.6. 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。

Syntax

```
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

- 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**）格式不保证重启后一致的设备名称，这可能会使一些 Kickstart 命令的使用复杂化。当命令需要设备节点名称时，您可以使用 **/dev/disk** 中的任何项目来作为替代。例如，使用以下设备名称：

```
part / --fstype=xfst --onpart=sda1
```

您可以使用类似以下条目之一：

```
part / --fstype=xfst --onpart=/dev/disk/by-path/pci-0000:00:05.0-scsi-0:0:0:0-part1
```

```
part / --fstype=xfst --onpart=/dev/disk/by-id/ata-ST3160815AS_6RA0C882-part1
```

通过使用此方法，命令始终以相同的存储设备为目标。这在大型存储环境中特别有用。要探索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。有关一致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.5.7. 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。

Syntax

```
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.5.8. iscsiname

iscsiname Kickstart 命令是可选的。它为 **iscsi** 命令指定的 iSCSI 节点分配名称。

Syntax

```
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.5.9. logvol

logvol Kickstart 命令是可选的。它为逻辑卷管理 (LVM) 创建一个逻辑卷。

Syntax

```
logvol mntpoint --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。

--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 硬编码值，并忽略用户指定的 **--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

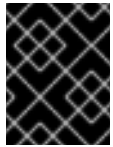
扩展逻辑卷以占据可用空间（若有），或使用指定的最大值（若有）。只有在磁盘镜像中预分配了最小存储空间时，必须使用该选项，并希望该卷扩展并占据所有可用空间。在物理环境中，这是一次性的操作。但是在虚拟环境中，当虚拟机将任何数据写入虚拟磁盘时，卷大小会增加。

--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

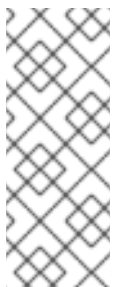
创建逻辑卷时使用这个选项，根据您的系统硬件自动决定这个卷的大小。

--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

指定该逻辑卷应该通过 Linux Unified Key Setup(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，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=

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

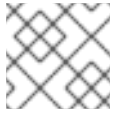
指定与精简逻辑卷搭配使用的配置配置集的名称。如果使用，该名称也会包含在给定逻辑卷的元数据中。默认情况下，可用的配置集为 **default** 和 **thin-performance**，在 **/etc/lvm/profile/** 目录中定义。有关其他信息，请参见 **lvm(8)** 手册页。

--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 **--backuppassphrase** 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
```

其他资源

- [配置和管理逻辑卷](#)

B.5.10. 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.5.11. nvdimmm

nvdimmm Kickstart 命令是可选的。它对非线性内存模块(NVDIMM)设备执行操作。

Syntax

```
nvdimmm action [OPTIONS]
```

Actions

- **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 **nvdimmm use** 命令就冗余了。这个操作使用以下格式：

```
nvdimmm reconfigure [--namespace=NAMESPACE] [--mode=MODE] [--sectorsize=SECTORSIZE]
```

- **--namespace=** - 按命名空间的设备规格。例如：

```
nvdimmm 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：

```
nvdimmm 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*
```

备注

- 默认情况下，安装程序忽略所有 NVDIMM 设备。您必须使用 **nvdimm** 命令在这些设备中启用安装。

B.5.12. part 或 partition

需要 **part** 或 **partition** Kickstart 命令。它在系统上创建一个分区。

Syntax

```
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：

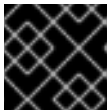
```
swap --hibernation
```

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。

- **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）。



重要

如果 **--size** 值太小，安装会失败。将 **--size** 值设置为您需要的最小空间量。

- **--grow** - 告诉分区使用所有可用空间（若有），或使用设置的最大值（如果指定了最大值）。



注意

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。但请注意，使用这种方式格式化的设备中不支持 GRUB2 安装，且必须将其放在有分区表的驱动器中。

```
partition pv.1 --onpart=hdb
```

- **--ondisk=** 或 **--ondrive=** - 在现有磁盘中创建分区（由 **part** 命令指定）。这个命令总是创建一个分区。强制在特定磁盘中创建分区。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，在分区表名称前加上 **lv**。使用格式 **lv pv.1**。...

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=xfs --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=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**。

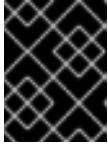
- **--fstype=** - 为分区设置文件系统类型。有效值为 **xfs**、**ext2**、**ext3**、**ext4**、**swap**、**vfat**、**efi** 和 **biosboot**。
- **--fsoptions** - 指定在挂载文件系统时要使用的自由格式选项字符串。该字符串将复制到安装的系统 **/etc/fstab** 文件中，并且应用引号括起来。



注意

在 EFI 系统分区 (**/boot/efi**) 中，**anaconda** 硬编码值，并忽略用户指定的 **--fsoptions** 值。

- **--label=** - 为单个分区分配标签。
- **--recommended** - 自动确定分区的大小。



重要

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。
- **--backuppssphrase** - 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。

- **part** 命令不是强制的，但您必须在 Kickstart 脚本中包含 **part**、**autopart** 或 **mount**。
- 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都会格式化为安装过程的一部分。
- **sdX**（或 **/dev/sdX**）格式不保证重启后一致的设备名称，这可能会使一些 Kickstart 命令的使用复杂化。当命令需要设备节点名称时，您可以使用 **/dev/disk** 中的任何项目来作为替代。例如，使用以下设备名称：

```
part / --fstype=xfstyp --onpart=sda1
```

您可以使用类似以下条目之一：

```
part / --fstype=xfstyp --onpart=/dev/disk/by-path/pci-0000:00:05.0-scsi-0:0:0:0-part1
```

```
part / --fstype=xfstyp --onpart=/dev/disk/by-id/ata-ST3160815AS_6RA0C882-part1
```

通过使用此方法，命令始终以相同的存储设备为目标。这在大型存储环境中特别有用。要探索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。有关一致引用存储设备的不同方法的更多信息，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 **--backupphrase** 选项创建加密密码短语备份。

B.5.13. raid

raid Kickstart 命令是可选的。它组成一个软件 RAID 设备。

Syntax

```
raid mntpoint --level=level --device=device-name partitions*
```

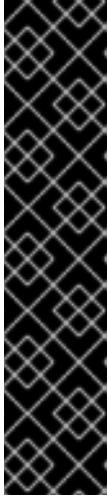
选项

- **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。



注意

在 EFI 系统分区 (**/boot/efi**) 中，anaconda 硬编码值，并忽略用户指定的 **--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。
- **--backuppssphrase** - 向这个设备添加随机生成的密码短语。将该密语存储在 **/root** 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 **--backupp passphrase** 选项创建加密密码短语备份。

B.5.14. 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。

Syntax

```
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.5.15. 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。

Syntax

```
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.5.16. volgroup

volgroup Kickstart 命令是可选的。它创建一个逻辑卷管理 (LVM) 组。

Syntax

```
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.5.17. 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。

Syntax

```
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.5.18. 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。

Syntax

```
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

如果自动 LUN 扫描可用且安装 9 或更高版本，则指定 FCP 设备总线 ID 足够了。否则，所有这三个参数都是必需的。如果没有通过 **zfcplun.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

```
zfcplun --devnum=0.0.4000 --wwpn=0x5005076300C213e9 --fcplun=0x5022000000000000
zfcplun --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。

Syntax

```
%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，请参阅 [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。

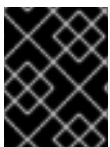
如果启用 kdump 且没有指定 **--reserve-mb=** 选项，则会使用 **auto** 值。

- **--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 com_redhat_oscapp

%addon com_redhat_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。

Syntax

```
%addon com_redhat_oscap
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 **%addon com_redhat_oscap** 部分的示例，它使用安装介质的 **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 com_redhat_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 com_redhat_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
```

其他资源

- [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（已弃用）

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。

Syntax

```
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。

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. nvdim
- e. zfc

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