



# Red Hat Enterprise Linux 9

## 9.2 发行注记

Red Hat Enterprise Linux 9.2 发行注记



# Red Hat Enterprise Linux 9 9.2 发行注记

---

Red Hat Enterprise Linux 9.2 发行注记

## 法律通告

Copyright © 2024 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

## 摘要

本发行注记提供了在 Red Hat Enterprise Linux 9.2 中已实现的改进和附加组件的高级信息，并记录了本版本中已知的问题，以及重要的 bug 修复、技术预览、已弃用的功能和其他详情。有关安装 Red Hat Enterprise Linux 的详情，请参考 Installation。

# 目录

|                                     |           |
|-------------------------------------|-----------|
| 对红帽文档提供反馈 .....                     | 5         |
| <b>第 1 章 概述</b> .....               | <b>6</b>  |
| 1.1. RHEL 9.2 中的主要变化                | 6         |
| 1.2. 原位升级                           | 8         |
| 1.3. 红帽客户门户网站 LABS                  | 9         |
| 1.4. 其他资源                           | 10        |
| <b>第 2 章 构架</b> .....               | <b>11</b> |
| <b>第 3 章 RHEL 9 发布的内容</b> .....     | <b>12</b> |
| 3.1. 安装                             | 12        |
| 3.2. 软件仓库                           | 12        |
| 3.3. 应用程序流                          | 12        |
| 3.4. 使用 YUM/DNF 的软件包管理              | 13        |
| <b>第 4 章 新功能</b> .....              | <b>14</b> |
| 4.1. 安装程序和镜像创建                      | 14        |
| 4.2. RHEL FOR EDGE                  | 15        |
| 4.3. 软件管理                           | 16        |
| 4.4. SHELL 和命令行工具                   | 17        |
| 4.5. 基础架构服务                         | 18        |
| 4.6. 安全性                            | 20        |
| 4.7. 网络                             | 25        |
| 4.8. 内核                             | 30        |
| 4.9. 文件系统和存储                        | 35        |
| 4.10. 高可用性和集群                       | 36        |
| 4.11. 动态编程语言、网页和数据库服务器              | 37        |
| 4.12. 编译器和开发工具                      | 41        |
| 4.13. 身份管理                          | 46        |
| 4.14. 桌面                            | 53        |
| 4.15. WEB 控制台                       | 54        |
| 4.16. RED HAT ENTERPRISE LINUX 系统角色 | 54        |
| 4.17. 虚拟化                           | 58        |
| 4.18. 支持性                           | 59        |
| 4.19. 容器                            | 60        |
| <b>第 5 章 对外部内核参数的重要更改</b> .....     | <b>63</b> |
| 新内核参数                               | 63        |
| 更新的内核参数                             | 64        |
| 新的 sysctl 参数                        | 71        |
| 更改了 sysctl 参数                       | 73        |
| <b>第 6 章 设备驱动程序</b> .....           | <b>74</b> |
| 6.1. 新驱动程序                          | 74        |
| 6.2. 更新的驱动程序                        | 75        |
| <b>第 7 章 可用的 BPF 功能</b> .....       | <b>76</b> |
| <b>第 8 章 程序错误修复</b> .....           | <b>93</b> |
| 8.1. 安装程序和镜像创建                      | 93        |
| 8.2. 订阅管理                           | 94        |
| 8.3. 软件管理                           | 94        |

|                                      |            |
|--------------------------------------|------------|
| 8.4. SHELL 和命令行工具                    | 95         |
| 8.5. 安全性                             | 95         |
| 8.6. 网络                              | 98         |
| 8.7. 内核                              | 99         |
| 8.8. 引导加载程序                          | 99         |
| 8.9. 文件系统和存储                         | 99         |
| 8.10. 高可用性和集群                        | 100        |
| 8.11. 编译器和开发工具                       | 101        |
| 8.12. 身份管理                           | 102        |
| 8.13. 图形基础结构                         | 103        |
| 8.14. WEB 控制台                        | 103        |
| 8.15. RED HAT ENTERPRISE LINUX 系统角色  | 103        |
| 8.16. 虚拟化                            | 105        |
| <b>第 9 章 技术预览</b> .....              | <b>106</b> |
| 9.1. 安装程序和镜像创建                       | 106        |
| 9.2. SHELL 和命令行工具                    | 106        |
| 9.3. 基础架构服务                          | 106        |
| 9.4. 安全性                             | 106        |
| 9.5. 网络                              | 107        |
| 9.6. 内核                              | 107        |
| 9.7. 文件系统和存储                         | 108        |
| 9.8. 编译器和开发工具                        | 109        |
| 9.9. 身份管理                            | 109        |
| 9.10. 桌面                             | 111        |
| 9.11. 图形基础结构                         | 112        |
| 9.12. WEB 控制台                        | 112        |
| 9.13. 虚拟化                            | 112        |
| 9.14. 云环境中的 RHEL                     | 114        |
| 9.15. 容器                             | 114        |
| <b>第 10 章 过时的功能</b> .....            | <b>115</b> |
| 10.1. 安装程序和镜像创建                      | 115        |
| 10.2. 订阅管理                           | 115        |
| 10.3. SHELL 和命令行工具                   | 116        |
| 10.4. 安全性                            | 116        |
| 10.5. 网络                             | 118        |
| 10.6. 内核                             | 118        |
| 10.7. 文件系统和存储                        | 119        |
| 10.8. 动态编程语言、网页和数据库服务器               | 119        |
| 10.9. 编译器和开发工具                       | 119        |
| 10.10. 身份管理                          | 120        |
| 10.11. 桌面                            | 121        |
| 10.12. 图形基础结构                        | 121        |
| 10.13. RED HAT ENTERPRISE LINUX 系统角色 | 122        |
| 10.14. 虚拟化                           | 122        |
| 10.15. 容器                            | 123        |
| 10.16. 已弃用的软件包                       | 124        |
| <b>第 11 章 已知问题</b> .....             | <b>125</b> |
| 11.1. 安装程序和镜像创建                      | 125        |
| 11.2. 软件管理                           | 128        |
| 11.3. SHELL 和命令行工具                   | 128        |
| 11.4. 基础架构服务                         | 129        |

---

|                                      |            |
|--------------------------------------|------------|
| 11.5. 安全性                            | 130        |
| 11.6. 网络                             | 134        |
| 11.7. 内核                             | 136        |
| 11.8. 文件系统和存储                        | 140        |
| 11.9. 动态编程语言、网页和数据库服务器               | 141        |
| 11.10. 编译器和开发工具                      | 141        |
| 11.11. 身份管理                          | 142        |
| 11.12. 桌面                            | 147        |
| 11.13. 图形基础结构                        | 148        |
| 11.14. WEB 控制台                       | 148        |
| 11.15. RED HAT ENTERPRISE LINUX 系统角色 | 148        |
| 11.16. 虚拟化                           | 149        |
| 11.17. 云环境中的 RHEL                    | 153        |
| 11.18. 支持性                           | 154        |
| 11.19. 容器                            | 154        |
| <b>附录 A. 按组件划分的问题单列表</b> .....       | <b>156</b> |
| <b>附录 B. 修订历史</b> .....              | <b>164</b> |





---

## 对红帽文档提供反馈

我们感谢您对我们文档的反馈。让我们了解如何改进它。

### 通过 JIRA 提交反馈（需要帐户）

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

# 第 1 章 概述

## 1.1. RHEL 9.2 中的主要变化

### 安装程序和镜像创建

镜像构建器的主要亮点：

- 内部镜像构建器现在提供一种新的改进方法，来在镜像构建器 Web 控制台中创建蓝图和镜像。
- 现在支持在 `/etc` 目录中创建自定义文件和目录。
- RHEL for Edge Simplified Installer 镜像类型现在在镜像构建器 web 控制台中提供。

如需更多信息，请参阅 [新功能 - 安装程序和镜像创建](#)。

### RHEL for Edge

RHEL for Edge 的主要亮点：

- 现在支持在蓝图中为 **simplified-installer** 镜像指定用户。
- RHEL for Edge Simplified 镜像现在支持 Ignition 置备工具。
- 现在，可在蓝图中制作没有 FDO 自定义部分的简化的安装程序镜像。

如需更多信息，请参阅 [新功能 - RHEL for Edge](#)。

### 安全性

与安全相关的主要亮点：

- OpenSSL 安全通讯库已 rebase 到版本 3.0.7。
- SELinux user-space 软件包已更新至版本 3.5。
- Keylime 已 rebase 到版本 6.5.2
- OpenSCAP 已 rebase 到版本 1.3.7。
- SCAP 安全指南 已 rebase 到版本 0.1.66。
- SCAP 安全指南中添加了用于闲置会话终止的新规则。
- Clevis 现在接受外部令牌。
- Rsyslog TLS 加密的日志现在支持多个 CA 文件。
- Rsyslog 特权被限制，以减少安全风险。
- fapolicyd 框架现在提供 RPM 数据库的过滤。

如需更多信息，请参阅 [新功能 - 安全性](#)。

### 动态编程语言、网页和数据库服务器

以下应用程序流的后续版本现在可用：

- Python 3.11

- nginx 1.22
- PostgreSQL 15

以下组件已升级：

- Git 升级到版本 2.39.1
- Git LFS 升级到版本 3.2.0

如需更多信息，请参阅 [新功能 - 动态编程语言、Web 和数据库服务器](#)。

## 编译器和开发工具

### 更新了系统工具链

RHEL 9.2 中已更新了以下系统工具链组件：

- GCC 11.3.1
- glibc 2.34
- binutils 2.35.2

### 更新了性能工具和调试器

RHEL 9.2 中已更新了以下性能工具和调试器：

- GDB 10.2
- Valgrind 3.19
- SystemTap 4.8
- Dyninst 12.1.0
- elfutils 0.188

### 更新了性能监控工具

RHEL 9.2 中已更新了以下性能监控工具：

- PCP 6.0.1
- Grafana 9.0.9

### 更新了编译器工具集

RHEL 9.2 中已更新了以下编译器工具集：

- GCC Toolset 12
- LLVM Toolset 15.0.7
- Rust Toolset 1.66
- Go Toolset 1.19.6

有关详细更改，请参阅 [新功能 - 编译器和开发工具](#)。

## RHEL 9 中的 Java 实现

RHEL 9 AppStream 软件仓库包括：

- **java-17-openjdk** 软件包，提供 OpenJDK 17 Java 运行时环境和 OpenJDK 17 Java 软件开发组件。
- **java-11-openjdk** 软件包，提供 OpenJDK 11 Java 运行时环境和 OpenJDK 11 Java 软件开发组件。
- **java-1.8.0-openjdk** 软件包，提供 OpenJDK 8 Java 运行时环境和开源 JDK 8 Java 软件开发组件。

红帽 OpenJDK 软件包的构建在其可移植 Linux 版本和 RHEL 9.2 及更新版本之间共享一组二进制文件。有了这个更新，在 RHEL 上从源 RPM 重建 OpenJDK 软件包的过程有一个变化。有关新重建过程的更多信息，请参阅 README.md 文件，该文件包含在 Red Hat build of OpenJDK 的 SRPM 软件包中，并由 `/usr/share/doc` 树下的 **javaawa-openjdk-headless** 软件包安装。

如需更多信息，请参阅 [OpenJDK 文档](#)。

## Web 控制台

RHEL web 控制台现在执行额外的步骤将 LUKS 加密的根卷绑定到 NBDE 部署。

您现在还可以通过图形界面应用以下 **加密子策略**：**DEFAULT:SHA1**、**LEGACY:AD-SUPPORT** 和 **FIPS:OSPP**。

如需更多信息，请参阅 [新功能 - Web 控制台](#)。

## 容器

主要变更包括：

- **podman** RHEL 系统角色现在可用。
- 现在提供了带有 Fulcio 和 Rekor 的 sigstore 签名的客户端。
- Skopeo 现在支持生成 sigstore 密钥对。
- Podman 现在支持事件的审核。
- Container Tools 软件包已更新。
- Aardvark 和 Netavark 网络堆栈现在支持自定义 DNS 服务器选择。
- Toolbox 现已正式提供。
- Podman Quadlet 现在作为技术预览提供。
- CNI 网络堆栈已弃用。

如需更多信息，请参阅 [新功能 - 容器](#)。

## 1.2. 原位升级

### 从 RHEL 8 原位升级到 RHEL 9

目前支持的原位升级路径包括：

- 在以下构架上从 RHEL 8.6 升级到 RHEL 9.0 以及从 RHEL 8.8 升级到 RHEL 9.2：
  - 64 位 Intel
  - 64 位 AMD

- 64-bit ARM
  - IBM POWER 9(little endian)
  - IBM Z 架构，不包括 z13
- 在带有 SAP HANA 的系统上，从 RHEL 8.6 升级到 RHEL 9.0 和从 RHEL 8.8 升级到 RHEL 9.2

如需更多信息，请参阅[支持的 Red Hat Enterprise Linux 原位升级路径](#)。

有关执行原位升级的步骤，请参阅[从 RHEL 8 升级到 RHEL 9](#)。

如果您要升级到带有 SAP HANA 的 RHEL 9.2，请确保在升级前系统已经过 SAP 认证。有关在具有 SAP 环境的系统上执行原位升级的说明，请参阅[如何将 SAP 环境从 RHEL 8 原位升级到 RHEL 9](#)。

主要改进包括：

- RHEL 原位升级路径策略已更改。如需更多信息，请参阅[支持的 Red Hat Enterprise Linux 原位升级路径](#)。
- 随着 RHEL 9.2 的发布，现在提供了从 RHEL 8 原位升级到 RHEL 9 的多个升级路径。对于当前版本，可以执行从 RHEL 8.8 原位升级到 RHEL 9.2 或从 RHEL 8.6 原位升级到 RHEL 9.0。请注意，可用的升级路径因标准的 RHEL 系统和具有 SAP HANA 的 RHEL 系统而异。
- **leapp-upgrade-el8toel9** 软件包的最新版本现在包含所有必需的 leapp 数据文件。客户不再需要手动下载这些数据文件。
- 现在支持 FIPS 模式下 RHEL 8.8 系统的原位升级。
- 现在，可以使用包含目标版本的 ISO 镜像进行原位升级。
- 现在，在原位升级过程中会自动检查 RPM 签名。要禁用自动检查，请在执行升级时使用 **--nogpgcheck** 选项。
- 订阅 RHSM 的系统现在会自动注册 Red Hat Insights。要禁用自动注册，请将 **LEAPP\_NO\_INSIGHTS\_REGISTER** 环境变量设为 1。
- 红帽现在收集与升级相关的数据，如升级启动和结束时间，以及升级是否成功，以用于工具使用情况分析。要禁用数据收集，请将 **LEAPP\_NO\_RHSM\_FACTS** 环境变量设为 1。

### 从 RHEL 7 原位升级到 RHEL 9

无法执行从 RHEL 7 直接升级到 RHEL 9 的原位升级。但是，您可以执行从 RHEL 7 原位升级到 RHEL 8，然后再执行到 RHEL 9 的第二个原位升级。如需更多信息，请参阅[从 RHEL 7 升级到 RHEL 8](#)。

## 1.3. 红帽客户门户网站 LABS

红帽客户门户网站 Labs 是客户门户网站的一个部分中的一组工具，地址为 <https://access.redhat.com/labs/>。红帽客户门户网站 Labs 中的应用程序可帮助您提高性能、快速解决问题、发现安全问题以及快速部署和配置复杂应用程序。一些最常用的应用程序有：

- [Registration Assistant](#)
- [Kickstart Generator](#)
- [Red Hat Product Certificates](#)
- [Red Hat CVE Checker](#)

- [Kernel Oops Analyzer](#)
- [Red Hat Code Browser](#)
- [VNC 配置器](#)
- [Red Hat OpenShift Container Platform Update Graph](#)
- [Red Hat Satellite Upgrade Helper](#)
- [JVM Options Configuration Tool](#)
- [负载均衡配置工具](#)
- [Red Hat OpenShift Data Foundation Supportability and Interoperability Checker](#)
- [Ansible Automation Platform Upgrade Assistant](#)
- [每个池计算器的 Ceph 放置组 \(PG\)](#)

## 1.4. 其他资源

与其他版本系统相比，Red Hat Enterprise Linux 9 的**能力和限制**可在知识库文章[Red Hat Enterprise Linux 技术能力和限制](#)中获得。

有关 Red Hat Enterprise Linux **生命周期** 的详情请查看 [Red Hat Enterprise Linux 生命周期文档](#)。

[软件包清单](#)文档为 RHEL 9 提供 **软件包列表**，包括许可证和应用程序兼容性等级。

[Red Hat Enterprise Linux 9: Application Compatibility Guide](#) 文档中的**解释应用程序兼容性等级**。

**RHEL 8 和 RHEL 9 的主要区别**（包括删除的功能）包括在[使用 RHEL 9 时的注意事项](#)。

有关如何执行从 **RHEL 8 到 RHEL 9 的原位升级**的说明，请参考[从 RHEL 8 升级到 RHEL 9](#) 的文档。

**Red Hat Insights**服务可让您主动发现、检查并解决已知的技术问题，所有 RHEL 订阅都可以使用它。有关如何安装 Red Hat Insights 客户端并将您的系统注册到该服务的说明，请查看[Red Hat Insights 入门](#)页面。

## 第 2 章 构架

Red Hat Enterprise Linux 9.2 与内核版本 5.14.0-284.11.1 一起分发，它支持以下构架的最低要求版本：

- AMD 和 Intel 64 位体系架构 (x86-64-v2)
- 64 位 ARM 架构(ARMv8.0-A)
- IBM Power Systems, Little Endian(POWER9)
- 64 位 IBM Z (z14)

请确定为每个构架购买正确的订阅。如需更多信息,请参阅 [Red Hat Enterprise Linux 入门 - 附加构架](#)。

## 第 3 章 RHEL 9 发布的内容

### 3.1. 安装

Red Hat Enterprise Linux 9 使用 ISO 镜像安装。AMD64、Intel 64 位、64 位 ARM、IBM Power Systems 和 IBM Z 架构有两种类型的 ISO 镜像：

- 安装 ISO：包含 BaseOS 和 AppStream 软件仓库的完整安装镜像,并允许您在没有附加软件仓库的情况下完成安装。在[产品下载页面](#)中，安装 ISO 被称为 **Binary DVD**。



#### 注意

安装 ISO 镜像的大小为几个 GB，因此可能不适用于光盘介质格式。当使用安装 ISO 镜像时，建议使用 USB 盘或 USB 硬盘驱动器创建可引导安装介质。您还可以使用 Image Builder 工具创建自定义的 RHEL 镜像。有关镜像构建器的更多信息，请参阅[编写自定义的 RHEL 系统镜像](#)文档。

- 引导 ISO：用来引导到安装程序的最小引导 ISO 镜像。这个选项需要访问 BaseOS 和 AppStream 软件仓库来安装软件包。软件仓库是安装 ISO 镜像的一部分。您还可以在安装过程中注册红帽 CDN 或 Satellite，以使用来自红帽 CDN 或 Satellite 的最新 BaseOS 和 AppStream 内容。

有关下载 ISO 镜像、创建安装介质以及完成 RHEL 安装的说明，请参阅[执行标准的 RHEL 9 安装](#)文档。有关自动 Kickstart 安装和其他高级主题，请参阅[执行高级 RHEL 9 安装](#)文档。

### 3.2. 软件仓库

Red Hat Enterprise Linux 9 由两个主要软件仓库发布：

- BaseOS
- AppStream

两个软件仓库都需要一个基本的 RHEL 安装，所有 RHEL 订阅都包括它们。

BaseOS 仓库的内容旨在提供底层操作系统功能的核心组件，为所有安装提供基础操作系统的基础。这部分内容采用 RPM 格式，它的支持条款与之前的 RHEL 版本相似。如需更多信息，请参阅[覆盖范围](#)详情文档。

AppStream 仓库的内容包括额外的用户空间应用程序、运行时语言和数据库来支持各种工作负载和使用案例。

另外，所有 RHEL 订阅都可以使用 CodeReady Linux Builder 软件仓库。它为开发人员提供了额外的软件包。不支持包括在 CodeReady Linux Builder 存储库中的软件包。

有关 RHEL 9 软件仓库及其提供的软件包的更多信息，请参阅[软件包清单](#)。

### 3.3. 应用程序流

用户空间组件的多个版本会以 Application Streams（应用程序流）的形式提供，其更新频率会比核心操作系统软件包的更新频率更快。这为自定义 RHEL 提供了更大的灵活性，而不影响平台或特定部署的基本稳定性。

应用程序流以 RPM 格式提供，可以是一个模块（RPM 格式的一个扩展），软件集合（Software Collections），或 Flatpaks。



每个 Application Stream 组件都有其特定的生命周期，可能和 RHEL 9 的生命周期相同或更短。有关 RHEL 生命周期信息，请查看 [Red Hat Enterprise Linux 生命周期](#)。

RHEL 9 改进了应用程序流的使用体验，它提供了初始的应用程序流版本，可以使用传统的 `dnf install` 命令作为 RPM 软件包进行安装。



### 注意

某些 RPM 格式的初始应用程序流的生命周期比 Red Hat Enterprise Linux 9 要短。

一些额外的 Application Stream 版本将作为模块发布，并在以后的 RHEL 9 次要发行本中带有较短的生命周期。模块是代表逻辑单元的软件包集合：应用程序、语言堆栈、数据库或一组工具。这些软件包被一同构建、测试并发布。

始终决定要安装哪个版本的应用程序流，并确保首先查看 [Red Hat Enterprise Linux 应用程序流生命周期](#)。

需要快速更新的内容（例如备用编译器和容器工具）会在滚动流中提供，且不会并行提供替代版本。滚动流可以打包为 RPM 或模块。

有关 RHEL 9 中可用的 Application Streams 及其应用程序兼容性级别的详情，请查看 [软件包清单](#)。Red Hat Enterprise Linux 9: [Application Compatibility Guide](#) 文档中的解释应用程序兼容性等级。

## 3.4. 使用 YUM/DNF 的软件包管理

在 Red Hat Enterprise Linux 9 中，使用 **DNF** 确保软件安装。红帽继续支持使用 **yum** 术语，以便与以前的 RHEL 主版本保持一致。如果您键入 **dnf** 而不是 **yum**，则命令按预期运行，因为它们都是兼容性的别名。

虽然 RHEL 8 和 RHEL 9 基于 **DNF**，但它们与 RHEL 7 中使用的 **YUM** 兼容。

如需更多信息，请参阅使用 [DNF 工具管理软件](#)。

## 第 4 章 新功能

这部分论述了 Red Hat Enterprise Linux 9.2 中引入的新功能和主要改进。

### 4.1. 安装程序和镜像创建

#### 在镜像构建器 Web 控制台中创建蓝图和镜像的一种新的改进方法

有了这个增强，您可以访问镜像构建器工具的统一版本，并大大改进用户体验。

镜像构建器仪表盘 GUI 中的显著改进包括：

- 现在，您可以使用之前只在 CLI 中支持的所有自定义（如内核、文件系统、防火墙、区域设置和其他自定义）自定义您的蓝图。
- 您可以通过上传或拖动 **.JSON** 或 **.TOML** 格式的蓝图来导入蓝图，并从导入的蓝图创建镜像。
- 您还可以以 **.JSON** 或 **.TOML** 格式导出或保存您的蓝图。
- 访问一个您可以排序、过滤和区分大小写的蓝图列表。
- 使用镜像构建器仪表盘，您现在可以通过以下标签页访问蓝图、镜像和源：
  - 蓝图 - 在 Blueprint 选项卡下，您现在可以导入、导出或删除您的蓝图。
  - 镜像 - 在 Images 选项卡下，您可以：
    - 下载镜像。
    - 下载镜像日志。
    - 删除镜像。
  - 资源 - 在 Sources 选项卡下，您可以：
    - 下载镜像。
    - 下载镜像日志。
    - 为镜像创建源。
    - 删除镜像。

Jira:RHELPLAN-139448

#### 能够在 `/etc` 目录中创建自定义文件和目录

有了这个增强，可以提供两个新的蓝图自定义。[[`customizations.files`]] 和 [[`customizations.directories`]] 蓝图自定义使您能够在镜像的 `/etc` 目录中创建自定义文件和目录。目前，您只能在 `/etc` 目录中使用这些自定义。

[[`customizations.directories`]] 使您能够：

- 创建新目录
- 为目录设置用户和组所有权
- 以八进制格式设置模式权限

使用 `[[customizations.files]]` 蓝图自定义，您可以：

- 在父目录 / 下创建新文件
- 修改现有文件 - 这会覆盖现有内容
- 为您要创建的文件设置用户和组所有权
- 以八进制格式设置模式权限



### 注意

所有镜像类型都支持新的蓝图自定义，如 `edge-container`、`edge-commit` 等。蓝图中用于创建安装程序镜像的不支持的自定义，如 `edge-raw-image`、`edge-installer` 和 `edge-simplified-installer`。

Jira:RHELPLAN-147428

### 能够在蓝图中为 `simplified-installer` 镜像指定用户

在以前的版本中，当为简化的安装程序镜像创建蓝图时，您无法在蓝图自定义中指定用户，因为自定义没有被使用，且被丢弃了。有了这个更新，当您从蓝图创建镜像时，此蓝图会在安装过程中在 `/usr/lib/passwd` 目录下创建一个用户，并在 `/usr/etc/shadow` 目录下创建一个密码。您可以使用您为蓝图创建的用户名和密码登录到设备。请注意，在访问系统后，您需要使用 `useradd` 命令创建用户。

Jira:RHELPLAN-149091

### 支持镜像构建器构建的 `.vhd` 镜像的 64 位 ARM

在以前的版本中，使用镜像构建器工具创建的 Microsoft Azure `.vhd` 镜像在 64 位 ARM 架构上不支持。这个更新增加了对 64 位 ARM Microsoft Azure `.vhd` 镜像的支持，现在您可以使用镜像构建器构建 `.vhd` 镜像，并将其上传到 Microsoft Azure 云。

Jira:RHELPLAN-139424

### 最小 RHEL 安装现在只安装 `s390utils-core` 软件包

在 RHEL 8.4 及之后的版本中，`s390utils-base` 软件包被分成 `s390utils-core` 软件包，以及一个辅助 `s390utils-base` 软件包。因此，将 RHEL 安装设置为 `minimal-environment` 只安装必要的 `s390utils-core` 软件包，而不是辅助 `s390utils-base` 软件包。如果要在最小 RHEL 安装中使用 `s390utils-base` 软件包，则必须在完成 RHEL 安装或使用 `kickstart` 文件显式安装 `s390utils-base` 后手动安装软件包。

Bugzilla:1932480

## 4.2. RHEL FOR EDGE

### RHEL for Edge Simplified 镜像中的 Ignition 支持

有了这个增强，您可以通过自定义蓝图在简化的安装程序镜像中添加 Ignition 文件。GUI 和 CLI 都支持 Ignition 自定义。RHEL for Edge 使用 Ignition 置备工具在引导过程的早期阶段将用户配置注入镜像。首次启动时，Ignition 从远程 URL 或简化的安装程序镜像中嵌入的文件读取其配置，并将该配置应用到镜像中。

Jira:RHELPLAN-139659

现在，可在蓝图中制作没有 FDO 自定义部分的简化的安装程序镜像

在以前的版本中，要构建 RHEL for Edge Simplified Installer 镜像，您必须在 FIDO 设备载入(FDO)自定义部分中添加详情。否则，镜像构建将失败。有了这个更新，蓝图中的 FDO 自定义现在是可选的，您可以构建 RHEL for Edge Simplified Installer 镜像，且无错误。

Jira:RHELPLAN-139655

### 对 RHEL for Edge 镜像的启用 MicroShift 的红帽构建

有了这个增强，您可以在 RHEL for Edge 系统中启用 MicroShift 服务的红帽构建。通过使用 `[[customizations.firewalld.zones]]` 蓝图自定义，您可以为蓝图自定义中的 `firewalld` 源添加支持。为此，为区域指定名称以及该特定区域的源列表。源可以是 `source[/mask]|MAC|ipset:ipset` 的形式。

以下是如何配置和自定义对 RHEL for Edge 系统中 Red Hat build of MicroShift 服务的红帽构建的蓝图示例。

```
[[packages]]
name = "microshift"
version = "*"
[customizations.services]
enabled = ["microshift"]
[[customizations.firewall.zones]]
name = "trusted"
sources = ["10.42.0.0/16", "169.254.169.1"]
```

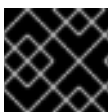
红帽 MicroShift 安装的构建要求，如防火墙策略、MicroShift RPM、`systemd` 服务，使您能够为生产环境创建一个部署，以实现将工作负载移植到最小的现场部署边缘设备，并在默认情况下启用 LVM 设备映射器。

Jira:RHELPLAN-136489

## 4.3. 软件管理

### 新的 `dnf offline-upgrade` 命令，用于在 RHEL 上离线更新

有了这个增强，您可以使用 DNF `system-upgrade` 插件中的新的 `dnf offline-upgrade` 命令对 RHEL 应用离线更新。



#### 重要

`system-upgrade` 插件中包含的 `dnf system-upgrade` 命令在 RHEL 上不支持。

[Bugzilla:2131288](#)

### 现在支持将公告安全过滤器应用到 `dnf offline-upgrade`

使用这个增强，添加了公告过滤的新功能。现在，您只能使用带有公告安全过滤器(`--advisory`、`--security`、`--bugfix` 及其他过滤器)的 `dnf offline-upgrade` 从指定的公告下载软件包及其依赖项。

[Bugzilla:2139326](#)

### `unmount_plugins` 函数现在可用于 DNF API

有了这个增强，在 DNF API 中已添加了新的 `unload_plugins` 函数，以允许插件卸载。



## 重要

请注意，您必须首先运行 `init_plugins` 函数，然后运行 `unload_plugins` 函数。

[Bugzilla:2121662](#)

### rpm2archive 的新的 `--nocompression` 选项

有了这个增强，`--nocompression` 选项已添加到 `rpm2archive` 工具中。您可以使用这个选项来避免在直接解包 RPM 软件包时压缩。

[Bugzilla:2150804](#)

## 4.4. SHELL 和命令行工具

### ReaR 现在在 64 位 IBM Z 架构上也被完全支持

在以前的版本中，基本的 Relax 和 Recover (ReaR) 功能在 64 位 IBM Z 架构上作为技术预览提供，并完全支持 `rear` 软件包版本 2.6-17.el9 或更高版本。您只能在 z/VM 环境中的 IBM Z 架构上创建 ReaR 救援镜像。目前不支持备份和恢复逻辑分区 (LPAR)。ReaR 仅在扩展计数密钥数据 (ECKD) 直接访问存储设备 (DASD) 上保存和恢复磁盘布局。不支持通过光纤通道协议 (FCP) 连接的固定的块访问 (FBA) DASD 和 SCSI 磁盘用于此目的。当前唯一可用的输出方法是 Initial Program Load (IPL)，它会生成一个内核和一个与 `zipl` 引导装载程序兼容的初始 `ramdisk` (`initrd`)。

如需更多信息，请参阅 [在 64 位 IBM Z 架构中使用 ReaR 救援镜像](#)。

[Bugzilla:2046653](#)

### systemd rebase 到版本 252

`systemd` 软件包已升级到版本 252。主要变更包括：

- 您可以使用 `system.conf` 和 `user.conf` 文件中的 `DefaultDeviceTimeoutSec=` 选项指定等待设备单元激活的默认超时时间。
- 现在，`systemd` 会记录阻止卸载文件系统的进程。
- 现在，您还可以对临时单元使用 `drop-ins`。
- 您可以在 `ConditionMemory=` 选项中使用大小后缀，如 K、M、G 和 T 等。
- 您可以使用 `systemctl list-automounts` 命令列出自动挂载点。
- 您可以使用 `systemd-logind` 工具，使用 `StopIdleSessionSec=` 选项在预配置的超时后停止闲置会话。
- `systemd-udev` 工具现在为 Infiniband 动词设备创建 `infiniband by-path` 和 `infiniband by-ibdev` 链接。
- `systemd-tmpfiles` 工具现在安全处理缺少的 `C` 复制源。
- `systemd-repart` 工具现在生成 `dm-verity` 分区，包括签名。

[Bugzilla:2217931](#)

更新的 `systemd-udev` 将一致的网络设备名称分配给 InfiniBand 接口

RHEL 9 中引入的 **systemd** 软件包的新版本包含更新的 **systemd-udev** 设备管理器。设备管理器将 InfiniBand 接口的默认名称更改为 **systemd-udev** 选择的一致性名称。

您可以按照 [重命名 IPoIB 设备](#) 流程为命名 InfiniBand 接口定义自定义命名规则。

有关命名方案的详情，请查看 [systemd.net-naming-scheme \(7\)](#) 手册页。

[Bugzilla:2136937](#)

## 4.5. 基础架构服务

### chrony rebase 到版本 4.3

**chrony** 套件已更新至版本 4.3。与版本 4.2 相比，主要改进包括：

- 添加了对网络时间协议(NTP)测量的长期数量过滤。您可以通过在 **pool**、**server** 或 **peer** 指令中添加 **maxdelayquant** 选项来启用此功能。
- 添加了选择日志，以提供有关 **chronyd** 选择源的更多信息。您可以通过在 **log** 指令中添加 **selection** 选项来启用选择日志。
- 改进了使用硬件时间戳和 Pulse-Per-Second Hardware Clock (PHC)参考时钟时的同步稳定性。
- 添加了对使用空闲运行稳定时钟（如 Temperature Compensated Crystal Oscillator (TCXO)、Oven-Controlled Crystal Oscillator）或原子时钟的支持。
- 将最大轮询率增加到每秒 128 条消息。

[Bugzilla:2133754](#)

### frr rebase 到版本 8.3.1

用于管理动态路由堆栈的 **frr** 软件包已更新至版本 8.3.1。8.2.2 版本的显著变化包括：

- 添加了一组新命令，以与边框网关协议(BGP)进行交互：
  - **set as-path replace** 命令，将 BGP 路由的 Autonomous System (AS) 路径属性替换为新值。
  - 在配置 BGP 路由映射时，与特定的 BGP peer 或 group 匹配的 **match peer** 命令。
  - **ead-es-frag evi-limit** 命令，用于设置对可在给定时间段内发送的每个 EVI 片段的以太网 A-D 数量的限制。
  - **match evpn route-type** 命令对某些类型的 EVPN 路由（如 route-target、route-distinguisher 或 MAC/IP 路由）执行特定的操作。
- 在 VTYSH 命令行界面中添加了 **show thread timers** 命令，以便与 FRR 守护进程进行交互。
- 添加了 **show ip ospf reachable-routers** 命令，以显示目前可通过 OSPF 协议访问的路由器列表。
- 添加了新命令，以与协议独立多播(PIM)守护进程进行交互：
  - **debug igmp trace detail** 命令，为具有详细追踪的互联网组管理协议(IGMP)消息启用调试。
  - **ip pim passive** 命令，以将接口配置为被动，而不是发送 PIM 消息。

- 为 **show zebra** 命令添加了新输出，如 ECMP、EVPN、MPLS 状态。
- 向 ZEBRA 组件中添加了 **show ip nht mrib** 命令，以显示内核中 **mroute** 表中与多播相关的信息。

[Bugzilla:2129731](#)

### vsftpd rebase 到版本 3.0.5

Very Secure FTP 守护进程(**vsftpd**)提供了一种在主机之间传输文件的安全方法。**vsftpd** 软件包已更新至版本 3.0.5。主要变化和增强包括以下 SSL 现代化：

- 默认情况下，**vsftpd** 工具现在需要使用 TLS 版本 1.2 或更高版本进行安全连接。
- **vsftpd** 工具现在与最新的 FileZilla 客户端兼容。

[Bugzilla:2018284](#)

### frr 软件包现在包含目标 SELinux 策略

由于管理动态路由堆栈的 **frr** 软件包快速发展，新的功能和访问向量缓存(AVC)会频繁地出现问题。有了这个增强，SELinux 规则与 FRR 一起打包，以更快地解决任何问题。SELinux 通过实行强制访问控制策略，为软件包添加了额外的保护级别。

[Bugzilla:2129743](#)

### powertop rebase 到版本 2.15

用于提高能源效率的 **powertop** 软件包已更新至版本 2.15。主要变化和增强包括：

- 修复了几个 Valgrind 错误和可能的缓冲区溢出，以提高 **powertop** 工具的稳定性。
- 改进了与 Ryzen 处理器和 Kaby Lake 平台的兼容性。
- 启用了 Lake Field、Aandder Lake N 和 Raptor Lake 平台支持。
- 启用了 Ice Lake NNPI 和 Meteor Lake 移动和桌面支持。

[Bugzilla:2044132](#)

### systemd-sysusers 工具在 chrony、dhcp、radvd 和 squid 软件包中提供

**systemd-sysusers** 工具在软件包安装过程中创建系统用户和组，并在删除软件包时删除它们。有了这个增强，以下软件包在其 scriptlets 中包含 **systemd-sysusers** 工具：

- **chrony**,
- **dhcp**,
- **radvd**,
- **squid**.

[Jira:RHELPLAN-136485](#)

### 现在提供了用于频率同步的新 synce4l 软件包

SyncE (Synchronous Ethernet)是一个硬件功能，它使 PTP 时钟能够在物理层上实现频率的精确同步。在某些网络接口卡(NIC)和网络交换机中支持 SyncE。

有了这个增强，现在新的 **synce4l** 软件包可用，它支持 SyncE。因此，由于更准确的时间同步，Telco Radio 访问网络(RAN)应用程序现在可以实现效率更高的通信。

Bugzilla:2143264

### tuned rebase 到版本 2.20.0

优化应用程序和工作负载性能的 TuneD 工具已更新至版本 2.20.0。2.19.0 的主要变化和增强包括：

- 扩展 API 使您能够在运行时在插件实例之间移动设备。
- **plugin\_cpu** 模块，其提供对 CPU 相关性能设置的微调，引进了以下改进：
  - **pm\_qos\_resume\_latency\_us** 功能使您能够将每个 CPU 允许的最大时间从闲置状态转换为活跃状态。
  - TuneD 添加了对 **intel\_pstate** 扩展驱动程序的支持，它提供缩放算法来根据不同的使用情况场景调整系统的电源管理。
- 通过 Unix 域套接字控制 TuneD 的套接字 API 现在作为技术预览提供。如需更多信息，请参阅 [TuneD 的套接字 API 作为技术预览提供](#)。

[Bugzilla:2133815](#), [Bugzilla:2113925](#), [Bugzilla:2118786](#), [Bugzilla:2095829](#)

## 4.6. 安全性

### libreswan rebase 到 4.9

**libreswan** 软件包已升级到版本 4.9。与以前版本相比的显著变化包括：

- 支持 **addconn** 和 **whack** 工具的 **{left,right}pubkey=** 选项
- KDF 自检
- 显示主机的身份验证密钥(**showhostkey**)：
  - 支持 ECDSA 公钥
  - 新的 **--pem** 选项,以打印 PEM 编码的公钥
- 互联网密钥交换协议版本 2 (IKEv2)：
  - 可扩展的身份验证协议 - 传输层安全(EAP-TLS)支持
  - 仅支持 EAP 身份验证
- **pluto** IKE 守护进程：
  - 支持 **maxbytes** 和 **maxpacket** 计数器

Bugzilla:2128669

### OpenSSL rebase 到 3.0.7

OpenSSL 软件包已更新至版本 3.0.7，其中包含各种 bug 修复和增强。最值得注意的是，默认供应商现在包含 **RIPEMD160** 哈希函数。

[Bugzilla:2129063](#)



## libssh 现在支持智能卡

现在，您可以通过公钥加密标准(PKCS)#11 统一资源标识符(URI)使用智能卡。因此，您可以将智能卡与 **libssh** SSH 库以及使用 **libssh** 的应用程序一起使用。

[Bugzilla:2026449](#)

## libssh rebase 到 0.10.4

用于实现安全远程访问和机器间文件传输的 SSH 协议的 **libssh** 库已更新至版本 0.10.4。

新特性：

- 添加了对 OpenSSL 3.0 的支持。
- 添加了对智能卡的支持。
- 添加了两个新的配置选项 **IdentityAgent** 和 **ModuliFile**。

其他显著变化包括：

- 不再支持比 1.0.1 早的 OpenSSL 版本
- 默认情况下，数字签名算法(DSA)支持已构建时禁用。
- SCP API 已被弃用。
- **pubkey** 和 **privatekey** API 已被弃用。

[Bugzilla:2068475](#)

## SELinux user-space 软件包已更新至 3.5

SELinux user-space 软件包 **libselinux**、**libsepol**、**libsemanage**、**checkpolicy**、**mcstrans** 和 **policycoreutils**，其包含 **sepolicy** 工具，已更新至版本 3.5。主要改进和程序错误修复包括：

- **sepolicy** 工具：
  - 在手册页中添加了缺失的布尔值
  - 多个 Python 和 GTK 更新
- 添加了对 **libselinux** 的临时解决方案，它减少了 **PCRE2** 库的堆内存使用率
- **libsepol** 软件包：
  - 拒绝了内核策略的类型 AV 规则中的属性
  - 不再写入空类定义，其允许更简单的往返测试
  - 更严格的策略验证
- **fixfiles** 脚本卸载 **SIGINT** 信号上的临时绑定挂载
- 修复了很多代码和拼写错误
- 删除了对已弃用的 Python 模块 **distutils** 和使用 PIP 的安装的依赖项
- **semodule** 选项 **--rebuild-if-modules-changed** 重命名为 **--refresh**

- 为生成的描述更新了翻译，并改进了对不支持的语言的处理
- 修复了很多静态代码分析 bug、模糊问题和编译器警告

[Bugzilla:2145224](#), [Bugzilla:2145228](#), [Bugzilla:2145229](#), [Bugzilla:2145226](#), [Bugzilla:2145230](#),  
[Bugzilla:2145231](#)

### OpenSCAP rebase 到 1.3.7

OpenSCAP 软件包已 rebase 到上游版本 1.3.7。此版本提供各种程序错误修复和增强，最重要的是：

- 修复了处理 OVAL 过滤器时的错误([RHBZ#2126882](#))
- 如果 XPath 不匹配，OpenSCAP 不再发出无效的空 `xmlfilecontent` 项([RHBZ#2139060](#))
- 防止 **Failed to check available memory** 错误([RHBZ#2111040](#))

[Bugzilla:2159286](#)

### SCAP 安全指南 rebase 到 0.1.66

SCAP 安全指南(SSG)软件包已 rebase 到上游版本 0.1.66。此版本提供各种改进和程序错误修复，最重要的是：

- 新的 CIS RHEL9 配置文件
- 弃用规则 `account_passwords_pam_faillock_audit`，而使用 `accounts_passwords_pam_faillock_audit`

[Bugzilla:2158405](#)

### 新的用于闲置会话终止的 SCAP 规则

新的 SCAP 规则 `logind_session_timeout` 已添加到用于增强和高级别的 ANSSI-BP-028 配置文件中的 `scap-security-guide` 软件包中。此规则使用 `systemd` 服务管理器的新功能，并在一定时间后终止闲置用户会话。此规则提供多个安全策略所需的健壮的空闲会话终止机制的自动配置。因此，OpenSCAP 可以自动检查与终止闲置用户会话相关的安全要求，如有必要，修复它。

[Bugzilla:2122325](#)

### Rsyslog 日志文件的 `scap-security-guide` 规则与 RainerScript 日志兼容

`scap-security-guide` 中用于检查和修复 Rsyslog 日志文件的所有权、组所有权和权限的规则现在也与 RainerScript 语法兼容。现代系统已在 Rsyslog 配置文件中使用了 RainerScript 语法，相应的规则无法识别此语法。因此，`scap-security-guide` 规则现在在两种可用语法中可以检查并修复 Rsyslog 日志文件的所有权、组所有权和权限。

[Bugzilla:2169414](#)

### Keylime rebase 到 6.5.2

`keylime` 软件包已 rebase 至上游版本 - `keylime-6.5.2-5.el9`。这个版本包含各种改进和 bug 修复，最重要的是：

- 解决了漏洞 [CVE-2022-3500](#)
- 当一个脚本在另一个之后快速执行时，Keylime 代理不再会无法通过 IMA 认证 [RHBZ#2138167](#)。
- 修复了 `/usr/share/keylime/create_mb_refstate` 脚本 [RHBZ#2140670](#) 中的分段错误

- 当启用了 `require_ek_cert` 选项时，注册中心不再会在 EK 验证过程中崩溃 [RHBZ#2142009](#)

[Bugzilla:2150830](#)

### Clevis 接受外部令牌

随着新的 `-e` 选项引入到 Clevis 自动加密工具，您可以提供一个外部令牌 ID，以避免在 `cryptsetup` 过程中输入密码。此功能使配置过程更加自动化、方便，特别是对于使用 Clevis 的软件包（如 `stratis`）等非常有用。

[Bugzilla:2126533](#)

### Rsyslog TLS 加密的日志现在支持多个 CA 文件

使用新的 `NetstreamDriverCaExtraFiles` 指令，您可以为 TLS 加密的远程日志记录指定额外的证书颁发机构(CA)文件的列表。请注意，新指令仅适用于 `ossl` (OpenSSL) Rsyslog 网络流驱动程序。

[Bugzilla:2124849](#)

### Rsyslog 权限是有限制的

Rsyslog 日志处理系统的权限现在仅限于 Rsyslog 明确所需的权限。这可最小化在输入资源（如网络插件）中出现潜在的错误时，最大限度地减少安全风险。因此，Rsyslog 具有相同的功能，但没有不必要的特权。

[Bugzilla:2127404](#)

### SELinux 策略允许 Rsyslog 在启动时丢弃特权

因为 Rsyslog 日志处理系统的特权现在被限制为最大程度减少安全风险([RHBZ#2127404](#))，所以 SELinux 策略已被更新，以允许 `rsyslog` 服务在启动时丢弃特权。

[Bugzilla:2151841](#)

### Tang 现在使用 `systemd-sysusers`

Tang 网络呈现服务器现在通过 `systemd-sysusers` 服务，而不是包含 `useradd` 命令的 shell 脚本来添加系统用户和组。这简化了系统用户列表的检查，您还可以通过提供具有更高优先级的 `sysuser.d` 文件来覆盖系统用户的定义。

[Bugzilla:2095474](#)

### opencryptoki rebase 到 3.19.0

`opencryptoki` 软件包已 rebase 至版本 3.19.0，它提供很多改进和 bug 修复。最值得注意的是，`opencryptoki` 现在支持以下功能：

- 特定于 IBM 的 Dilithium 密钥
- 双功能加密函数
- 使用新的 `C_SessionCancel` 函数取消基于活跃会话的操作，如 PKCS#11 Cryptographic Token Interface Base Specification v3.0 中所述
- 通过 `CKM_IBM_ECDSA_OTHER` 机制的 Schnorr 签名
- 通过 `CKM_IBM_BTC_DERIVE` 机制的比特币密钥派生
- IBM z16 系统中的 EP11 令牌

Bugzilla:2110314

## SELinux 现在限制 mptcpd 和 udftools

有了这个 **selinux-policy** 软件包的更新，SELinux 会限制以下服务：

- **mptcpd**
- **udftools**

Bugzilla:1972222

## fapolicyd 现在提供对 RPM 数据库的过滤

使用新的配置文件 `/etc/fapolicyd/rpm-filter.conf`，您可以自定义 **fapolicyd** 软件框架存储在信任数据库中的 RPM-database 文件的列表。这样，您可以阻止 RPM 安装的某些应用程序，或者允许被默认的配置过滤器拒绝的应用程序。

Jira:RHEL-192

## GnuTLS 可以在解密和加密过程中添加和删除填充

在解密和加密过程中，某些协议的实现需要 PKCS#7 填充。**gnutls\_cipher\_encrypt3** 和 **gnutls\_cipher\_decrypt3** 块密码函数已添加到 GnuTLS 中，以透明地处理填充。因此，如果原始纯文本的长度不是块大小的倍数，则您现在可以将这些函数与 **GNUTLS\_CIPHER\_PADDING\_PKCS7** 标志结合使用，以自动添加或删除填充。

Bugzilla:2084161

## NSS 不再支持少于 1023 位的 RSA 密钥

网络安全服务(NSS)库的更新将所有 RSA 操作的最小密钥大小从 128 位改为 1023 位。这意味着 NSS 不再执行以下功能：

- 生成大于 1023 位的 RSA 密钥。
- 使用 RSA 密钥签名或验证 RSA 签名少于 1023 位。
- 使用 RSA 密钥加密或解密值少于 1023 位。

Bugzilla:2091905

## 现在，在启用了 FIPS 的系统上强制 Extended Master Secret TLS 扩展

随着 [RHSA-2023:3722](#) 公告的发布，在启用了 FIPS 的 RHEL 9 系统上，对 TLS 1.2 连接强制 **Extended Master Secret (EMS)** 扩展 (RFC 7627)。这符合 FIPS-140-3 要求。TLS 1.3 不受影响。

不支持 EMS 或 TLS 1.3 的旧客户端现在无法连接到运行在 RHEL 9 上的 FIPS 服务器。同样，FIPS 模式下的 RHEL 9 客户端无法连接到只支持没有 EMS 的 TLS 1.2 服务器。在实践中意味着这些客户端无法连接到 RHEL 6、RHEL 7 和非 RHEL 传统操作系统上的服务器。这是因为传统的 OpenSSL 1.0.x 版本不支持 EMS 或 TLS 1.3。

另外，如果 hypervisor 使用没有 EMS 的 TLS 1.2，则从启用了 FIPS 的 RHEL 客户端连接到 hypervisor（如 VMWare ESX）现在会失败，并显示 **Provider routines::ems not enabled** 错误。要临时解决这个问题，请更新 hypervisor 以支持带有 EMS 扩展的 TLS 1.3 或 TLS 1.2。对于 VMWare vSphere，这意味着版本 8.0 或更高版本。

如需更多信息，请参阅 [使用 Red Hat Enterprise Linux 9.2 强制执行的 TLS 扩展 "Extended Master Secret"](#)。

[Bugzilla:2188046](#),[Bugzilla:2218721](#)

## 4.7. 网络

### NetworkManager rebase 到版本 1.42.2

**NetworkManager** 软件包已升级到上游版本 1.42.2, 与以前的版本相比, 它提供很多改进和 bug 修复 :

- 以太网绑定支持源负载均衡。
- NetworkManager 可以管理 **loopback** 设备上的连接。
- 添加了对 IPv4 equal-cost multi-path(ECMP)路由的支持。
- 添加了对 Virtual Local Area Networks (VLAN)连接中 **802.1ad** 标记的支持。
- **nmtui** 应用程序支持 Wi-Fi WPA-Enterprise、具有 802.1X 身份验证的以太网, 以及 MACsec 连接配置文件。
- 如果所有地址的 IPv6 重复地址检测(DAD)都失败, 则 NetworkManager 会拒绝 DHCPv6 租期。

有关显著变化的更多信息, 请参阅 [上游发行注记](#)。

[Bugzilla:2134897](#)

### 使用 NetworkManager, 在 ECMP 路由中引入了 **weight** 属性

有了这个更新, 在定义 IPv4 Equal-Cost Multi-Path (ECMP)路由时, RHEL 9 支持一个新的属性 **weight**。您可以使用 NetworkManager 配置多路径路由来负载均衡和稳定网络流量。这允许对两个节点之间的数据传输使用多路径, 这提高了网络效率, 并在链接失败时提供冗余。使用 **weight** 属性的条件包括 :

- 有效值为 1-256。
- 使用 **weight** 属性将多个下一跳路由定义为单跳路由。
- 如果没有设置 **weight**, NetworkManager 无法将路由合并到 ECMP 路由中。

[Bugzilla:2081302](#)

### NetworkManager 更新为跨多个网络的 DNS 配置提高了灵活性

有了这个更新, 您可以使用 `/etc/Networkmanager/NetworkManager.conf` 文件中现有的 `[global-dns]` 部分来配置 DNS 选项, 而无需在 `[global-dns-domain-*]` 部分中指定 `nameserver` 值。这使您能够在 `/etc/resolv.conf` 文件中配置 DNS 选项, 同时仍依赖于网络连接为实际 DNS 解析提供的 DNS 服务器。因此, 这个功能可在使用不同的 DNS 服务器连接到不同的网络时更轻松更灵活地管理 DNS 设置。特别是当您使用 `/etc/resolv.conf` 文件来配置 DNS 选项时。

[Bugzilla:2019306](#)

### NetworkManager 现在支持一个新的 `vlan.protocol` 属性

有了这个更新, `vlan` 接口类型接受一个新的 `protocol` 属性。属性类型是字符串。接受的值是 **802.1Q** (默认) 或 **802.1ad**。新属性指定哪个 VLAN 协议控制封装的标签标识符。

[Bugzilla:2128809](#)

### NetworkManager 现在允许通过非受管接口的 VLAN 配置

有了这个增强，在使用 NetworkManager 配置虚拟 LAN (VLAN) 时，您可以使用非受管网络接口作为基本接口。因此，VLAN 基础接口保持不变，除非通过 `nmcli device set enp1s0 managed true` 命令明确更改，或者通过 NetworkManager 的其他 API。

[Bugzilla:2110307](#)

## 现在完全支持使用 NetworkManager 配置多路径 TCP

在这个版本中，NetworkManager 工具为您提供了多路径 TCP (MPTCP) 功能。您可以使用 `nmcli` 命令控制 MPTCP，并使其设置持久。

如需更多信息，请参阅：

- [了解多路径 TCP：端点和高可用性以及未来的网络](#)
- [RFC 8684：具有多个地址的多路径操作的 TCP 扩展](#)
- [为 MPTCP 应用程序永久配置多路径](#)

[Bugzilla:2029636](#)

## NetworkManager 工具现在支持激活 loopback 接口上的连接

管理员可以管理 **loopback**，来：

- 向 **loopback** 接口添加额外的 IP 地址
- 定义 DNS 配置
- 定义一个不绑定到接口的特殊路由
- 定义一条路由规则，其不与接口相关
- 更改 **loopback** 接口的最大传输单元(MTU)大小

[Bugzilla:2073512](#)

## 现在支持 **balance-slb** 绑定模式

新的 **balance-slb** 绑定模式源负载均衡不需要交换机配置。**balance-slb** 使用 `xmit_hash_policy=vlan+srcmac` 划分源以太网地址上的流量，NetworkManager 为流量过滤添加必要的 **nftables** 规则。现在，您可以使用 NetworkManager 启用的 **balance-slb** 选项创建绑定配置文件。

[Bugzilla:2128216](#)

## **firewalld** rebase 到版本 1.2

**firewalld** 软件包已升级到版本 1.2，该版本提供了多个改进。主要变更包括：

- 对新服务（如 netdata、IPFS）的支持
- fail-safe 模式，以确保系统保持保护状态，如果 **firewalld** 服务在启动过程中遇到错误，则不会破坏网络通信
- 一些 **firewalld** 策略命令的命令行(CLI)中的 tab 补全

[Bugzilla:2125371](#)

## **firewalld** 现在支持启动 failsafe 机制



有了这个增强，**firewalld** 会在启动失败时回到 **failsafe** 默认值。此功能在出现无效配置或其他启动问题时保护主机。因此，即使用户配置无效，运行 **firewalld** 的主机也会启动 **failsafe**。

[Bugzilla:2077512](#)

### contrack-tools rebase 到版本 1.4.7

**contrack-tools** 软件包已升级至版本 1.4.7，它提供多个 bug 修复和增强。主要变更包括：

- 添加 **IPS\_HW\_OFFLOAD** 标志，其将 **contrack** 条目的卸载指定到硬件
- 添加了 **clash\_resolve** 和 **chaintoolong** 统计计数器
- 支持按 IP 地址系列过滤事件
- 在 **contrackd.conf** 文件中接受 **yes** 或 **no** 作为 **on** 或 **off** 的同义词
- 支持用户空间帮助程序在守护进程启动时自动加载。用户不必手动运行 **nfct add helper** 命令
- 删除了 **-o userspace** 命令选项，并且始终对用户空间触发的事件进行标记
- 仅将外部注入问题作为记录为警告
- 在查找缓存条目时忽略 **contrack** ID，以允许替换卡住的旧条目
- 修复了 **ssdp cthelper** 模块中的 IPv6 **M-SEARCH** 的解析中断问题
- 消除了 **nfct** 库中对延迟绑定技术的需求
- 清理协议值解析，捕获无效的值

[Bugzilla:2132398](#)

### nmstate API 现在支持 IPv6 本地链接地址作为 DNS 服务器

有了这个增强，您可以使用 **nmstate** API 将 IPv6 本地链接地址设置为 DNS 服务器。使用 **<link-local\_address>%<interface>** 格式，例如：

```
dns-resolver:
  config:
    server:
      - fe80::deef:1%enp1s0
```

[Bugzilla:2095207](#)

### nmstate API 现在支持 MPTCP 标记

这个更新增强了支持 MultiPath TCP (MPTCP) 标记的 **nmstate** API。因此，您可以使用 **nmstate** 来对具有静态或动态 IP 地址的接口设置 MPTCP 地址标记。

[Bugzilla:2120473](#)

### min-mtu 和 max-mtu 属性已添加到所有接口上的 MTU

在以前的版本中，异常信息不够清晰，不足以了解支持的 MTU 范围。此更新为所有接口引入了 **min-mtu** 和 **max-mtu** 属性。因此，当所需的 MTU 超出范围时，**nmstate** 将指示支持的 MTU 范围。

[Bugzilla:2044150](#)

## NetworkManager 现在允许通过非受管接口的 VLAN 配置

有了这个增强，在使用 NetworkManager 配置虚拟 LAN (VLAN) 时，您可以使用非受管网络接口作为基本接口。因此，VLAN 基础接口保持不变，除非通过 `nmcli device set enp1s0 managed true` 命令明确更改，或者通过 NetworkManager 的其他 API。

[Bugzilla:2058292](#)

## 现在支持 `balance-slb` 绑定模式

新的 `balance-slb` 绑定模式源负载均衡不需要交换机配置。`balance-slb` 使用 `xmit_hash_policy=vlan+srcmac` 划分源以太网地址上的流量，NetworkManager 为流量过滤添加必要的 `nftables` 规则。现在，您可以使用 NetworkManager 启用的 `balance-slb` 选项创建绑定配置文件。

[Bugzilla:2130240](#)

## Nmstate 中的新 `weight` 属性

此更新在 Nmstate API 和工具套件中引入了 `weight` 属性。您可以使用 `weight` 来指定 Equal Cost Multi-Path 路由 (ECMP) 组中每个路径的相对权重。权重是 1 到 256 之间的数字。因此，Nmstate 中的 `weight` 属性对 ECMP 组中流量分布提供了更大的灵活性和控制。

[Bugzilla:2162401](#)

## xdp-tools rebase 到版本 1.3.1

`xdp-tools` 软件包已升级到上游版本 1.3.1，与之前的版本相比，其提供了很多改进和 bug 修复：

- 添加了以下工具：
  - `xdp-bench`：在接收端执行 XDP 基准。
  - `xdp-monitor`：使用内核跟踪点监控 XDP 错误和统计信息。
  - `xdp-trafficgen`：通过 XDP 驱动程序钩子生成并发送流量。
- 在 `libxdp` 库中添加了以下功能：
  - 添加了 `xdp_multiprog__xdp_frags_support ()`、`xdp_program__set_xdp_frags_support ()` 和 `xdp_program__xdp_frags_support ()` 函数，以支持具有 XDP `frags` 支持的加载程序，这是一个称为 `multibuffer XDP` 的功能。
  - 在将程序附加到 `AF_XDP` 套接字时，库会执行合适的引用计数。因此，在使用套接字时，应用程序不再需要手动与 XDP 程序分离。现在，当程序不再使用时，`libxdp` 库会自动与程序分离。
  - 在库中添加了以下函数：
    - 用于创建 `xdp_program` 对象的 `xdp_program__create()`
    - 用于克隆 `xdp_program` 引用的 `xdp_program__clone()`
    - 用于通过 `BPF_PROG_TEST_RUN` 内核 API 运行 XDP 程序的 `xdp_program__test_run ()`
  - 当设置了 `LIBXDP_BPF_FS_AUTOMOUNT` 环境变量时，如果什么也没有找到，`libxdp` 库现在支持自动挂载 `bpf fs` 虚拟文件系统。现在，当没有挂载 `bpf fs` 时，库功能的子集也可以正常工作。



请注意，此版本还会更改正在网络设备上加载的 XDP 分配程序的版本号。这意味着您不能同时使用之前和新版本的 **libxdp** 和 **xdp-tools**。**libxdp** 1.3 库将显示分配程序的旧版本，但不会自动升级它们。另外，在使用 **libxdp** 1.3 加载程序后，旧版本不会与新版本进行交互。

[Bugzilla:2160066](#)

## iproute rebase 到版本 6.1.0

**iproute** 软件包已升级至版本 6.1.0，其提供多个 bug 修复和增强。主要变更包括：

- 支持读取 **vdpa** 设备统计信息
  - 显示在索引 1 处读取 **virtqueue** 数据结构的统计信息：

```
# vdma dev vstats show vdma-a qidx 1
vdma-a:
vdma-a: queue_type tx received_desc 321812 completed_desc 321812
```

- 演示在索引 16 处读取 **virtqueue 数据** 结构的统计信息：

```
# vdma dev vstats show vdma-a qidx 16
vdma-a: queue_type control_vq received_desc 17 completed_desc 17
```

- 更新对应的手册页

[Bugzilla:2155604](#)

## 现在，内核会在 SYN 洪水信息中记录侦听地址

此增强将侦听的 IP 地址添加到 SYN 洪水信息中：

```
Possible SYN flooding on port <ip_address>:<port>.
```

因此，如果很多进程绑定到不同 IP 地址上的同一端口，则管理员现在可以明确地识别受影响的套接字。

[Bugzilla:2143850](#)

## 为 VLAN 接口引进新的 nmstate 属性

使用此 **nmstate** 框架的更新，引进了以下 VLAN 属性：

- **registration-protocol**: VLAN 注册协议。有效值为 **gvrp** (GARP VLAN Registration Protocol), **mvrp** (Multiple VLAN Registration Protocol)和 **none**。
- **reorder-headers** : 重新排序输出数据包标头。有效值为 **true** 和 **false**。
- **loose-binding** : 放松接口到其主设备的操作状态的绑定。有效值为 **true** 和 **false**。

您的 YAML 配置文件类似以下示例：

```
---
interfaces:
- name: eth1.101
  type: vlan
  state: up
  vlan:
```

```
base-iface: eth1
id: 101
registration-protocol: mvrp
loose-binding: true
reorder-headers: true
```

Jira:RHEL-19142

## 4.8. 内核

### RHEL 9.2 中的内核版本

Red Hat Enterprise Linux 9.2 与内核版本 5.14.0-284.11.1 一起分发。

[Bugzilla:2177782](#)

### 64k 页大小内核现在可用

除了支持 4k 页的 ARM 内核的 RHEL 9 外，红帽现在还提供了支持 64k 页的可选内核软件包：**kernel-64k**。

64k 页大小内核是 ARM 平台上大型数据集的一个有用的选项。它为某些类型内存和 CPU 密集型操作启用了更好的性能。

您必须在安装时在 64 位 ARM 架构系统上选择页大小。您只能通过将 **kernel-64k** 软件包添加到 **Kickstart** 文件中的软件包列表来通过 Kickstart 安装 **kernel-64k**。

有关安装 **kernel-64k** 的更多信息，请参阅 [执行高级 RHEL 9 安装](#)。

[Bugzilla:2153073](#)

### 启用了 **kexec-tools** 的 **virtiofs** 支持

此增强通过引入新选项为 **kexec-tools** 添加了 **virtiofs** 功能，**virtiofs myfs**，其中 **myfs** 是 **qemu** 命令行中设置的变量标签名称，例如 **-device vhost-user-fs-pci,tag=myfs**

**virtiofs** 文件系统实现一个驱动程序，其允许客户机挂载已在主机上导出的目录。通过使用此增强，您可以将虚拟机的 **vmcore** 转储文件保存到：

- **virtiofs** 共享目录。
- 子目录（如 **/var/crash**），当根文件系统是 **virtiofs** 共享目录时。
- 不同的 **virtiofs** 共享目录，当虚拟机的根文件系统是 **virtiofs** 共享目录时。

[Bugzilla:2085347](#)

### **kexec-tools** 软件包现在对远程 **kdump** 目标添加了改进

有了此增强，**kexec-tools** 软件包添加了重要的 bug 修复和增强。最显著的变化包括：

- 通过只启用所需的网络接口来为 **kdump** 优化内存消耗。
- 在连接超时失败时，改进了 **kdump** 的网络效率。网络建立的默认等待时间为最多 10 分钟。这消除了传递 **dracut** 参数的需要，如 **rd.net.timeout.carrier** 或 **rd.net.timeout.dhcp**，来作为识别载体的一种临时解决方案。

[Bugzilla:2076416](#)

## BPF rebase 到版本 6.0

Berkeley Packet Filter (BPF)工具已 rebase 至具有多个增强的 Linux 内核版本 6.0。此更新使启用了所有依赖于 BPF 类型格式(BTF)的 BPF 功能。这些功能包括使用 BPF trampolines 进行追踪、Compile Once - Run Everywhere (CO-RE)机制的可用性以及几个与网络相关的功能。另外，内核模块现在包含调试信息，这意味着您不再需要安装 **debuginfo** 软件包来检查正在运行的模块。

有关在运行的内核中可用的 BPF 功能的完整列表，请使用 **bpftool feature** 命令。

Jira:RHELPLAN-133650

## rtla meta-tool 添加了 osnoise 和 timerlat 追踪程序，以提高追踪能力

Real-Time Linux Analysis (**rtla**)是一个 meta-tool，其中包含一组用于分析 Linux 实时属性的命令。**rtla** 利用内核追踪功能来提供有关意外系统结果的属性和根本原因的确切信息。**rtla** 目前添加了对 **osnoise** 和 **timerlat** 追踪程序命令的支持：

- **osnoise** 追踪程序报告有关操作系统噪声的信息。
- **timerlat** 追踪程序会在计时器 IRQ 处理程序和线程处理程序中定期打印计时器延迟。

请注意，要使用 **rtla** 的 **timerlat** 功能，您必须使用 **sysctl -w kernel.sched\_rt\_runtime\_us=-1** 脚本禁用准入控制。

Bugzilla:2075216

## Tuna 的 argparse 模块现在支持配置 CPU 套接字

有了这个增强，当有多个 CPU 套接字时，您可以指定特定的 CPU 套接字。您可以在子命令上使用 **-h** 来查看帮助用法，例如 **tuna show\_threads -h**。

要配置特定的 CPU 套接字，请对每个需要使用 CPU 套接字的 **tuna** 命令指定 **-S** 选项：

```
tuna <command> [-S CPU_SOCKET_LIST]
```

例如，使用 **tuna show\_threads -S 2,3** 查看线程或使用 **tuna show\_irqs -S 2,3** 查看附加的中断请求 (IRQ)。

因此，这个增强有助于基于 CPU 套接字的 CPU 使用率，而无需单独指定每个 CPU。

Bugzilla:2122781

## Tuna 中 cgroups 和 irq 的输出格式已被改进，以提供更好的可读性

有了这个增强，**cgroup** 工具的 **tuna show\_threads** 命令输出现在根据终端大小进行构建。您还可以通过向 **show\_threads** 命令添加新的 **-z** 或 **--spaced** 选项，将额外的空格配置到 **cgroups** 输出。

因此，**cgroups** 输出现在有一个改进的适合您终端大小的可读格式。

Bugzilla:2121517

## 向 tuna 工具中实时添加了一个新的命令行界面

此增强向 **tuna** 工具添加了一个新的命令行界面，该工具基于 **argparse** 解析模块。有了这个更新，您可以执行以下任务：

- 更改应用程序和内核线程的属性。

- 根据名称或号对中断请求(IRQ)进行操作。
- 使用进程标识符对任务或线程进行操作。
- 使用 CPU 或套接字号指定 CPU 和 CPU 集。

通过使用 **tuna -h** 命令，您可以打印命令行参数及其相应的选项。对于每个命令，您可以使用 **tuna <command> -h** 命令查看可选的参数。

因此，**tuna** 现在提供了一个界面，它有一个比命令行界面更容易使用和维持的命令和选项的标准化菜单。

[Bugzilla:2062865](#)

### **rteval** 命令输出现在包括程序加载和测量线程信息

**rteval** 命令现在显示带有程序负载、测量线程以及运行这些线程的相应 CPU 的报告概述。这些信息有助于评估特定硬件平台上负载下实时内核的性能。

**rteval** 报告被写入 XML 文件，以及系统的引导日志，并保存到 **rteval-<date>-N-tar.bz2** 压缩文件中。**date** 指定报告生成的日期，**N** 是第 **N** 次运行的计数器。

要生成 **rteval** 报告，请输入以下命令：

```
# rteval --summarize rteval-<date>-N.tar.bz2
```

[Bugzilla:2081325](#)

### 在 **oslat** 程序中添加了 **-W** 和 **--bucket-width** 选项来测量延迟

有了这个增强，您可以以纳秒的精度为单个存储桶指定延迟范围。不是 1000 纳秒倍数的 Widths 表示纳秒的精度。通过使用新选项 **-W** 或 **--bucket-width**，您可以修改存储桶之间的延迟间隔，以便在微秒延迟时间内测量延迟。

例如，要在 10 秒内为 32 个存储桶设置宽度为 100 纳秒的延迟存储桶，以便在 1-4 范围的 CPU 上运行，并省略零存储桶大小，请运行以下命令：

```
# oslat -b 32 -D 10s -W 100 -z -c 1-4
```

请注意，在使用选项前，您必须确定与错误测量相关的哪个级别的精度非常重要。

[Bugzilla:2041637](#)

### **NVMe/FC** 传输协议作为 **kdump** 存储目标启用

**kdump** 机制现在提供对通过 Fibre Channel (NVMe/FC) 协议作为转储目标的 Nonvolatile Memory Express (NVMe) 协议的支持。有了这个更新，您可以配置 **kdump** 来将内核崩溃转储文件保存在 NVMe/FC 存储目标上。

因此，**kdump** 可以在内核崩溃而没有 **timeout** 或 **reconnect** 错误的情况下捕获并在 **NVMe/FC** 上保存 **vmcore** 文件。

有关 NVMe/FC 配置的更多信息，请参阅 [管理存储设备](#)

[Bugzilla:2080110](#)

### **crash-utility** 工具已 rebase 至版本 8.0.2

分析活跃系统状态或在内核崩溃后分析活跃系统状态的 **crash-utility** 已 rebase 到版本 8.0.2。主要更改包括添加了对 **multiqueue (blk-mq)** 设备的支持。通过使用 **dev -d** 或 **dev -D** 命令，您可以显示 **multiqueue (blk-mq)** 设备的磁盘 I/O 统计信息。

[Bugzilla:2119685](#)

### openssl-ibmca rebase 到版本 2.3.1

64 位 IBM Z 架构上的 IBMCA 的动态 OpenSSL 引擎和提供程序已 rebase 到上游版本 2.3.1。建议 RHEL 9 用户使用 OpenSSL 提供程序来确保与 OpenSSL 的未来更新兼容。*engine* 功能已在 OpenSSL 版本 3 中弃用。

[Bugzilla:2110378](#)

### 使用客户密钥的安全客户机转储加密

这个新功能允许安全执行客户机的 hypervisor 发起的转储在 **kdump** 工具无法正常工作的情况下从 KVM 收集内核崩溃信息。请注意，安全执行的 hypervisor 发起的转储是为 IBM Z 系列 z16 和 LinuxONE Emperor 4 硬件设计的。

[Bugzilla:2044204](#)

### ADL-S 平台上实时的 TSN 协议已启用

有了这个增强，IEEE Time Sensitive Networking (TSN) 规范允许在 Intel Alder Lake S (ADL-S) 平台上通过网络启用时间同步和实时工作负载的确定性处理。它支持以下网络设备：

- 支持 TSN 的离散 2.5GbE MAC-PHY 组合：Intel® i225/i226
- 在具有来自 Marvell、Maxlinear 和涵盖 1GbE 和 2.5Gbe 的 TI 的第三方 PHY 芯片的 SOC 中的集成 2.5GbE MAC，可在选择 **skus** 和 SOC 时提供。

使用 TSN 协议，您可以在嵌入式实现中管理确定性应用程序调度、抢占和准确的时间同步类型工作负载。这些实现需要专用、专门和专有的网络，而工作负载运行在标准以太网、Wi-Fi 和 5G 网络上。

因此，TSN 为以下提供改进的功能：

- 硬件：用于在 IoT 中实现实时工作负载的基于 Intel 的系统
- 确定性和时间敏感的应用程序

[Bugzilla:2100606](#)

### Intel ice 驱动程序 rebase 到版本 6.0.0

Intel **ice** 驱动程序已升级至上游版本 6.0.0，与之前的版本相比，它提供了大量的改进和 bug 修复。主要改进包括：

- 通过以太网的点到点协议 (**PPPoE**) 协议硬件卸载
- Inter-Integrated Circuit (**I2C**) 协议写命令
- 以太网交换机设备驱动程序模型 (**switchdev**) 中的 VLAN 标签协议标识符 (**TPID**)
- **switchdev** 中的双 VLAN 标记

[Bugzilla:2104468](#)

### gnss 模块的写数据的选项现在可用

这个更新提供了将数据写到 **gnss** 接收器的选项。在以前的版本中，**gnss** 不能完全配置。有了这个增强，所有 **gnss** 功能现在都可用。

Bugzilla:2111048

## 为 IBM zSystems 托管安全引导证书

从 IBM z16 A02/AGZ 和 LinuxONE Rockhopper 4 LA2/AGL 开始，您可以在硬件管理控制台(HMC)上启动启用了安全引导的系统时管理用于验证 Linux 内核的证书。值得注意的是：

- 您可以使用 DPM 中的 HMC 和经典模式从 HMC 可以访问的 FTP 服务器加载系统证书存储中的证书。也可以从附加到 HMC 的 USB 设备加载证书。
- 您可以将存储在证书存储中的证书与 LPAR 分区相关联。多个证书可以与一个分区关联，一个证书可以与多个分区关联。
- 您可以使用 HMC 接口解除证书存储中的证书与分区的关联。
- 您可以从证书存储中删除证书。
- 您可以最多将 20 个证书与一个分区关联。

内置的固件证书仍然可用。特别是，当您使用用户管理的证书存储时，内置证书将不再可用。

载入到证书存储中的证书文件必须满足以下要求：

- 它们具有 **PEM** 或 **DER** 编码的 **X.509v3** 格式，并具有以下文件扩展名之一：**.pem**、**.cer**、**.crt** 或 **.der**。
- 它们不会过期。
- 密钥用法属性必须是 **数字签名**。
- 扩展的密钥用法属性必须包含 **代码签名**。

固件接口允许 Linux 内核在逻辑分区中运行，以加载与此分区关联的证书。IBM Z 上的 Linux 将这些证书存储在 **.platform** 密钥环中，允许 Linux 内核使用与该分区关联的证书验证 **kexec** 内核和第三方内核模块。

操作员只负责上传验证的证书，并删除已撤销的证书。



### 注意

[Product Signing Keys](#) 中提供了您需要加载到 HMC 的 **Red Hat Secureboot 302** 证书。

Bugzilla:2190123

## zipl 支持 64 位 IBM Z 上的安全引导 IPL 和转储

使用此更新，**zipl** 工具支持 64 位 IBM Z 架构上 Extended Count Key Data (ECKD) Direct Access Storage Devices (DASD) 中的 List-Directed IPL 和 List-Directed 转储。因此，IBM Z 上 RHEL 的安全引导也适用于 DASD 的 ECKD 类型。

Bugzilla:2044200

## rtla rebase 到上游 kernel 源代码的 6.6 版本

**rtla** 工具已升级到最新的上游版本，其提供多个 bug 修复和增强。主要变更包括：

- 添加了 **-C** 选项，来为除了主 **rtla** 线程外的 **rtla** 线程指定要在其中运行的其他控制组。
- 添加了 **--house-keeping** 选项，来将 **rtla** 线程放在内务清理 CPU 上，并将测量线程放在不同的 CPU 上。
- 添加了对 **timerlat** 追踪器的支持，以便您可以在用户空间中运行 **timerlat hist** 和 **timerlat top** 线程。

Jira:RHEL-18359

## 4.9. 文件系统和存储

### **nvme-cli rebase 到版本 2.2.1**

**nvme-cli** 软件包已升级到版本 2.2.1，其提供多个 bug 修复和增强。主要变更包括：

- 添加了新的 **nvme show-topology** 命令，该命令显示所有 NVMe 子系统的拓扑。
- 丢弃了 **libuuid** 依赖项。
- **uint128** 数据字段显示正确。
- 将 **libnvme** 依赖项更新至版本 1.2。

Bugzilla:2139753

### **libnvme rebase 到版本 1.2**

**libnvme** 软件包已升级到版本 1.2，其提供多个 bug 修复和增强。最显著的变化是丢弃了 **libuuid** 库的依赖项。

[Bugzilla:2139752](#)

### **Stratis 在池中强制实施一致的块大小**

Stratis 现在在池中强制实施一致的块大小，以解决池中存在混合块大小设备时可能会出现潜在边缘情况问题。有了这个增强，用户无法再创建池，或者添加与池中现有设备有不同块大小的设备。因此，池失败的风险降低了。

[Bugzilla:2039957](#)

### **支持 Stratis 池中现有磁盘增长**

在以前的版本中，当用户向 RAID 阵列中添加新磁盘时，RAID 阵列的大小通常会增加。然而，在所有情况下，Stratis 会忽略大小的增加，并继续使用 RAID 阵列中第一个添加到池中的空间。因此，Stratis 无法识别新设备，用户无法增加池的大小。

有了这个增强，Stratis 现在识别任何大小扩展了的池设备成员。现在，用户可以发出一条命令来根据要求扩展池。

除了通过添加新磁盘来增加池的现有功能外，Stratis 还支持池中现有磁盘的增长。

[Bugzilla:2039955](#)

### **改进了 Ivreduce 命令的功能**



有了这个增强，当逻辑卷(LV)处于活跃状态时，**lvreduce** 命令会检查是否缩小 LV 大小是否会破坏其上存在的任何文件系统。如果 LV 上的文件系统需要减少，并且没有启用 **lvreduce resizefs** 选项，则不会缩小 LV。

另外，新的选项现在可用于在减少 LV 的时候控制文件系统的处理。在使用 **lvreduce** 命令时，这些选项为用户提供了更大的灵活性和控制。

[Bugzilla:1878893](#)

### 添加了 **statx** 的直接 I/O 对齐信息

此更新向 **statx(2)** 调用引入了一个新的掩码值 "**STATX\_DIOALIGN**"。当在 **stx\_mask** 字段中设置这个值时，它会请求 **stx\_dio\_mem\_align** 和 **stx\_dio\_offset\_align** 值，这两个值分别表示用户内存缓冲区所需的对齐（以字节为单位）以及该文件上直接 I/O（**O\_DIRECT**）的文件偏移和 I/O 段长度。如果不支持文件上的直接 I/O，则两个值都是 0。现在，这个接口已为块设备以及 RHEL9 中的 **xfs** 和 **ext4** 文件系统上的文件实现。

[Bugzilla:2150284](#)

### NFSv4.1 会话中继发现

有了这个更新，客户端可以使用到同一服务器的多个连接和会话，从而加快数据传输。当 NFS 客户端挂载具有不同 IP 地址的多主目录 NFS 服务器时，默认只使用一个连接，忽略其余的连接。为提高性能，这个更新添加了对 **trunkdiscovery** 和 **max\_connect** 挂载选项的支持，该选项可让客户端测试每个连接，并将多个连接与同一个 NFSv4.1+ 服务器和会话关联。

[Bugzilla:2066372](#)

### NFS IO 大小现在可以设为 TCP 和 RDMA 的 **PAGE\_SIZE** 的倍数

此更新允许用户将 NFS IO 大小设为 TCP 和 RDMA 连接的 **PAGE\_SIZE** 的倍数。这在优化某些构架的 NFS 性能方面提供了更大的灵活性。

[Bugzilla:2107347](#)

### **nfsrahead** 已添加到 RHEL 9 中

随着 **nfsrahead** 工具的引入，您可以使用它来修改 NFS 挂载的 **readahead** 值，因此会影响 NFS 的读性能。

[Bugzilla:2143747](#)

## 4.10. 高可用性和集群

### 新的 **enable-authfile** Booth 配置选项

当您在集群配置中创建 Booth 配置以使用 Booth 票据管理器时，**pcs booth setup** 命令现在默认启用新的 **enable-authfile** Booth 配置选项。您可以使用 **pcs booth enable-authfile** 命令在现有集群上启用这个选项。另外，**pcs status** 和 **pcs booth status** 命令现在会在检测到可能的 **enable-authfile** 错误配置时显示警告。

[Bugzilla:2116295](#)

### **pcs** 现在可以运行资源和 stonith 代理的 **validate-all** 操作

当创建或更新资源或 STONITH 设备时，您现在可以指定 **--agent-validation** 选项。有了此选项，**pcs** 使用代理的 **validate-all** 操作（当它可用时），除了 **pcs** 根据代理元数据的所完成的验证外。



[Bugzilla:2112270](#),[Bugzilla:2159454](#)

## 4.11. 动态编程语言、网页和数据库服务器

### RHEL 9 中 Python 3.11 可用

RHEL 9.2 引入了 Python 3.11，由新软件包 **python3.11** 提供，以及为其构建的一套软件包，以及 **ubi9/python-311** 容器镜像。

与之前发布的 Python 3.9 相比，显著改进包括：

- 显著提高了性能。
- 使用新的 **match** 关键字（类似于其它语言中的 **switch**）的结构化模式匹配。
- 改进了错误消息，例如，表示未闭合的括号或方括号。
- 用于调试和其他用例的确切行号。
- 通过将定义放在括号中来支持跨多行定义上下文管理器。
- 与类型提示和 **typing** 模块相关的各种新功能，如新的 **X | Y** 类型联合操作符、可变泛型和新的 **Self** 类型。
- 追踪中指向导致错误的表达式的精确错误位置。
- 新的 **tomllib** 标准库模块，它支持解析 TOML。
- 可以使用例外组和新 **except\*** 语法，同时提高和处理多个不相关的异常。

Python 3.11 及为其构建的软件包可与 Python 3.9 并行安装在同一系统上。

要从 **python3.11** 堆栈安装软件包，请使用：

```
# dnf install python3.11
# dnf install python3.11-pip
```

要运行解释器，例如：

```
$ python3.11
$ python3.11 -m pip --help
```

如需更多信息，请参阅 [安装和使用 Python](#)。

请注意，Python 3.11 的生命周期会短于 Python 3.9，这是 RHEL 9 中的默认 Python 实现；请参阅 [Red Hat Enterprise Linux 应用程序流生命周期](#)。

[Bugzilla:2127923](#)

### nodejs:18 rebase 到版本 18.14，npm rebase 到版本 9

更新的 **Node.js 18.14** 包括 **npm** 从版本 8 到 9 的 SemVer 主升级。由于维护原因，这个更新是必需的，可能需要调整 **npm** 配置。

值得注意的是，不再支持不限定到特定注册中心的与身份验证相关的设置。这一变化是出于安全原因而做的。如果您使用了无范围的身份验证配置，则提供的令牌将被发送到 `.npmrc` 文件中列出的每个注册中心。

如果您使用无范围的身份验证令牌，请在 `.npmrc` 文件中生成并提供注册中心范围的令牌。

如果您有使用 `_auth` 的配置行，如 `.npmrc` 文件中的 `//registry.npmjs.org/:_auth`，请将它们替换为 `//registry.npmjs.org/:_authToken=${NPM_TOKEN}`，并提供您生成的有范围的令牌。

有关更改的完整列表，请参阅 [上游更改日志](#)。

[Bugzilla:2178088](#)

## git rebase 到版本 2.39.1

Git 版本控制系统已更新至版本 2.39.1，与之前发布的版本 2.31 相比，它提供了 bug 修复、功能增强和性能改进。

主要改进包括：

- `git log` 命令现在支持 `git describe` 输出的格式占位符：`git log --format=%(describe)`
- `git commit` 命令现在支持 `--fixup<commit>` 选项，该选项可让您修复提交的内容，而不更改日志消息。有了这个更新，您还可以使用：
  - `--fixup=amend:<commit>` 选项，来更改消息和内容。
  - `--fixup=reword:<commit>` 选项，来只更新提交消息。
- 您可以将新的 `--reject-shallow` 选项与 `git clone` 命令一起使用，以禁用来自 shallow 存储库的克隆。
- `git branch` 命令现在支持 `--recurse-submodules` 选项。
- 现在，您可以使用 `git merge-tree` 命令来：
  - 测试两个分支是否可以合并。
  - 如果分支被合并了，计算在合并提交中生成的树。
- 您可以使用新的 `secure.bareRepository` 配置变量来过滤 bare 存储库。

[Bugzilla:2139379](#)

## git-lfs rebase 到版本 3.2.0

Git 大文件存储(LFS) 扩展已更新至版本 3.2.0，与之前发布的版本 2.13 相比，其提供了 bug 修复、增强和性能改进。

主要变更包括：

- Git LFS 引入了一个纯基于 SSH 的传输协议。
- Git LFS 现在提供了一个合并驱动程序。
- `git lfs fsck` 工具现在还会检查指针是否规范，并且是否预期的 LFS 文件具有正确的格式。
- 对 NT LAN Manager (NTLM)身份验证协议的支持已删除。改为使用 Kerberos 或基本身份验证。

[Bugzilla:2139383](#)

## 新模块流：nginx:1.22

**nginx 1.22** web 和代理服务器现在作为 **nginx:1.22** 模块流提供。与之前发布的版本 1.20 相比，这个更新提供了很多 bug 修复、安全修复、新功能和增强。

新特性：

- **nginx** 现在支持：
  - 使用 OpenSSL 3.0 时的 OpenSSL 3.0 和 **SSL\_sendfile()** 函数。
  - PCRE2 库。
  - **mail** 代理模块中的 POP3 和 IMAP 管道。
- **nginx** 现在将 **Auth-SSL-Protocol** 和 **Auth-SSL-Cipher** 标头行传递给邮件代理服务器。

增强的指令：

- 现在提供了多个新指令，如 **ssl\_conf\_command** 和 **ssl\_reject\_handshake**。
- **proxy\_cookie\_flags** 指令现在支持变量。
- **nginx** 现在支持以下指令中的变量：**proxy\_ssl\_certificate**, **proxy\_ssl\_certificate\_key**, **grpc\_ssl\_certificate**, **grpc\_ssl\_certificate\_key**, **uwsgi\_ssl\_certificate**, 和 **uwsgi\_ssl\_certificate\_key**。
- 流模块中的 **listen** 指令现在支持一个新的 **fastopen** 参数，该参数为侦听套接字启用 **TCP Fast Open** 模式。
- 在 **mail** 代理模块中添加了一个新的 **max\_errors** 指令。

其他更改：

- **nginx** 现在总是返回一个错误，如果：
  - 使用了 **CONNECT** 方法。
  - **Content-Length** 和 **Transfer-Encoding** 头都在请求中指定。
  - 请求头名称包含空格或控制字符。
  - **Host** 请求头行包含空格或控制字符。
- **nginx** 现在阻止包含 **Transfer-Encoding** 头的所有 HTTP/1.0 请求。
- **nginx** 现在使用应用程序层协议协商(ALPN)建立了 HTTP/2 连接，且不再支持 Next Protocol Negotiation (NPN)协议。

要安装 **nginx:1.22** 流，请使用：

```
# dnf module install nginx:1.22
```

如需更多信息，请参阅[设置和配置 NGINX](#)。

有关支持的 **nginx** 模块流长度的详情，请查看 [Red Hat Enterprise Linux 应用程序流生命周期](#)。

Bugzilla:2096174

## mod\_security rebase 到版本 2.9.6

Apache HTTP 服务器的 **mod\_security** 模块已更新至版本 2.9.6，与之前可用的版本 2.9.3 相比，其提供了新功能、bug 修复和安全修复。

主要改进包括：

- 调整了 **modsecurity.conf-recommended** 文件中的解析器激活规则。
- 对 **mod\_security** 解析 HTTP 多部分请求的方式的改进。
- 添加了一个新的 **MULTIPART\_PART\_HEADERS** 集合。
- 向格式化的日志时间戳添加了微秒时间戳解析。
- 添加了缺少的地理国家。

Bugzilla:2143211

## 新软件包：tomcat

RHEL 9.2 引入了 Apache Tomcat 服务器版本 9。Tomcat 是 Java Servlet 和 JavaServer Pages 技术官方参考实现中使用的 servlet 容器。Java Servlet 和 JavaServer Pages 规范是由 Sun 在 Java 社区流程下开发的。Tomcat 是在开放和参与的环境中开发的，并在 Apache 软件许可证版本 2.0 下发布。

Bugzilla:2160511

## 新模块流：postgresql:15

RHEL 9.2 引入了 **PostgreSQL 15** 作为 **postgresql:15** 模块流。与版本 13 相比，**PostgreSQL 15** 提供了很多新功能和增强。主要变更包括：

- 现在，您可以使用子脚本访问 **PostgreSQL** JSON 数据。查询示例：

```
SELECT ({ "postgres": { "release": 15 } }::jsonb)['postgres']['release'];
```

- **PostgreSQL** 现在支持多范围数据类型，并扩展了 **range\_agg** 函数来聚合多范围数据类型。
- **PostgreSQL** 改进了监控和可观察性：
  - 现在，您可以跟踪 **COPY** 命令和 Write-ahead-log (WAL) 活动的进度。
  - **PostgreSQL** 现在提供对复制槽的统计信息。
  - 通过启用 **compute\_query\_id** 参数，您现在可以通过几个 **PostgreSQL** 功能（包括 **pg\_stat\_activity** 或 **EXPLAIN VERBOSE**）唯一跟踪查询。
- **PostgreSQL** 通过以下方法改进了对查询并行的支持：
  - 提高了并行顺序扫描的性能。
  - 在使用 **RETURN QUERY** 命令时，SQL Procedural Language (**PL/pgSQL**) 执行并行查询的能力。
  - 在 **REFRESH MATERIALIZED VIEW** 命令中启用的并行。

- **PostgreSQL** 现在包含 SQL 标准 **MERGE** 命令。您可以使用 **MERGE** 编写可在单个语句中包含 **INSERT**、**UPDATE** 和 **DELETE** 操作的条件 SQL 语句。
- **PostgreSQL** 提供以下新功能以使用正则表达式来检查字符串：**regexp\_count ()**，**regexp\_instr ()**，**regexp\_like ()** 和 **regexp\_substr ()**。
- **PostgreSQL** 添加了 **security\_invoker** 参数，您可以使用它来查询具有视图调用者，而不是视图创建者权限的数据。这有助于您确保视图调用者具有使用底层数据的正确权限。
- **PostgreSQL** 提高了性能，即在其归档和备份方面。
- **PostgreSQL** 添加了对 **LZ4** 和 **Zstandard (zstd)** 丢失压缩算法的支持。
- **PostgreSQL** 提高了其内存中和磁盘上排序算法。
- 更新的 **postgresql.service** systemd 单元文件现在可确保 **postgresql** 服务在网络启动后启动。

以下更改向后不兼容：

- 公共模式的默认权限已被修改。新创建的用户需要使用 **GRANT ALL ON SCHEMA public TO myuser;** 命令明确授予权限。例如：

```
postgres=# CREATE USER mydbuser;
postgres=# GRANT ALL ON SCHEMA public TO mydbuser;
postgres=# \c postgres mydbuser
postgres=# CREATE TABLE mytable (id int);
```

- 在管道模式下不再支持 **libpq PQsendQuery ()** 函数。修改受影响的应用程序，以使用 **PQsendQueryParams ()** 函数。

另请参阅 [使用 PostgreSQL](#)。

要安装 **postgresql:15** 流，请使用：

```
# dnf module install postgresql:15
```

如果要从 RHEL 9 中早期的 **postgresql** 流升级，请迁移 **PostgreSQL** 数据，如 [迁移到 PostgreSQL 的 RHEL 9 版本](#) 中所述。

有关支持的 **postgresql** 模块流长度的详情，请查看 [Red Hat Enterprise Linux 应用程序流生命周期](#)。

[Bugzilla:2128410](#)

## 4.12. 编译器和开发工具

### openblas rebase 到版本 0.3.21

OpenBLAS 库已更新至版本 0.3.21。这个更新包括 IBM POWER10 平台的性能优化补丁。

[Bugzilla:2112099](#)

### 新模块流：**swig:4.1**

RHEL 9.2 引入了 Simplified Wrapper 和 Interface Generator (SWIG) 版本 4.1 作为 CodeReady Linux Builder (CRB) 存储库中提供的 **swig:4.1** 模块流。请注意，不支持 CodeReady Linux Builder 存储库中包含的软件包。

与 RHEL 9.0 中发布的 **SWIG 4.0** 相比，**SWIG 4.1**：

- 添加了对 **Node.js** 版本 12 到 18 的支持，并删除对早于 6 的 **Node.js** 版本的支持。
- 添加了对 **PHP 8** 的支持。
- 通过 **PHP C API** 完全处理 **PHP** 包装，默认情况下不再生成一个 **.php** 包装器。
- 仅支持 **Perl 5.8.0** 及更新的版本。
- 向 3.11 添加了对 **Python** 版本 3.9 的支持。
- 仅支持 **Python 3.3** 及更新的 **Python 3** 版本以及 **Python 2.7**。
- 为 **Python** 生成的代码中的各种内存泄漏提供修复。
- 改进了对 C99、C++11、C++14 和 C++17 标准的支持，并开始实现 C++20 标准。
- 添加对 C++ **std::unique\_ptr** pointer 类的支持。
- 包括 C++ 模板处理中的几个小改进。
- 修复了各种情况下 C++ 声明用法。

要安装 **swig:4.1** 模块流：

1. 启用 [CodeReady Linux Builder \(CRB\) 存储库](#)。
2. 安装模块流：

```
# dnf module install swig:4.1
```

[Bugzilla:2139101](#)

### 新软件包：CRB 存储库中的 **jmc**

RHEL 9.2 引入了 HotSpot JVMs 版本 8.2.0 的 JDK Mission Control (JMC) 配置集，作为 AMD 和 Intel 64 位架构的 CodeReady Linux Builder (CRB) 存储库中的 **jmc** 软件包提供。

要安装 JMC，您必须首先启用 [CodeReady Linux Builder \(CRB\) 存储库](#)。

请注意，不支持 CRB 存储库中包含的软件包。

[Bugzilla:2122401](#)

### OpenJDK 服务属性现在在 FIPS 模式下可用

在以前的版本中，OpenJDK 在 FIPS 模式下提供的加密服务和算法被严格过滤，导致服务属性不可用。有了这个增强，这些服务属性在 FIPS 模式下可用。

[Bugzilla:2186803](#)

### Performance Co-Pilot rebase 到版本 6.0

Performance Co-Pilot (PCP) 已更新至版本 6.0。主要改进包括：

1. 版本 3 PCP 归档支持：

这包括对实例域 `change-deltas`、Y2038-safe 时间戳、纳秒精度时间戳、任意时区支持以及用于大型（超过 2 GB）单个卷的 64 位文件偏移的支持。

此功能目前通过 `/etc/pcp.conf` 文件中的 `PCP_ARCHIVE_VERSION` 设置进行选择。

版本 2 存档保留默认值。

2. 在整个 PCP 中只使用 OpenSSL。Mozilla NSS/NSPR 使用已被丢弃：这会影 响 `libpcp`、`pmAPI` 客户端和 `PMCD` 加密。现在，这些元素已配置，并与 `pmproxy` HTTPS 支持和 `redis-server` 一致，它们都使用 OpenSSL。
3. 新的纳秒精度时间戳 `PMAPI` 调用使用时间戳的 `PCP` 库接口。这些都是可选的，为现有工具保留完全向后兼容。
4. 以下工具和服务已更新：

#### **pcp2elasticsearch**

实现了身份验证支持。

#### **pcp-dstat**

实现了对 `top-alike` 插件的支持。

#### **pcp-htop**

更新至最新的稳定上游版本。

#### **pmseries**

添加了 `sum,avg,stdev,nth_percentile,max_inst,max_sample,min_inst` 和 `min_sample` 函数。

#### **pmdabpf**

添加了 CO-RE (Compile Once - Run Everywhere) 模块，并支持 AMD64、Intel 64 位、64 位 ARM 和 IBM Power Systems。

#### **pmdabpftrace**

将示例自动启动脚本移到 `/usr/share` 目录中。

#### **pmdadenki**

添加了对多个活跃电池的支持。

#### **pmdalinux**

对最新的 `/proc/net/netstat` 更改的更新。

#### **pmdaopenvswitch**

添加了额外的接口和覆盖统计信息。

#### **pmproxy**

请求参数现在可以在请求正文中发送。

#### **pmieconf**

为 Open vSwitch 指标添加了几个 `pmie` 规则。

#### **pmlogger\_farm**

添加了场日志记录器的默认配置文件。

#### **pmlogger\_daily\_report**

一些主要效率改进。

[Bugzilla:2117074](#)

**grafana rebase 到版本 9.0.9**

**grafana** 软件包已 rebase 至版本 9.0.9。主要变更包括：

- 现在，时间序列面板是默认的视觉化选项，替换了图形面板
- 新的 heatmap 面板
- 新的 Prometheus 和 Loki 查询构建器
- 更新了 Grafana Alerting
- 改进了多个 UI/UX 和性能
- 许可证从 Apache 2.0 变为 GNU Affero General Public License (AGPL)

以下内容作为 opt-in 实验性功能提供：

- 新条形图面板
- 新状态时间线面板
- 新的状态历史记录面板
- 新的直方图面板

如需更多信息，请参阅：[Grafana v9.0 中的新内容](#) 以及 [Grafana v8.0 中的新内容](#)。

Bugzilla:2116847

### **grafana-pcp rebase 到版本 5.1.1**

**grafana-pcp** 软件包已 rebase 到版本 5.1.1。主要变更包括：

#### **查询编辑器**

添加了禁用速率转换和时间利用率转换的按钮。

#### **Redis**

删除了已弃用的 **label\_values (metric, label)** 函数。

#### **Redis**

修复了带有许多系列的指标的网络错误（需要 Performance Co-Pilot v6+）。

#### **Redis**

将 **pmproxy** API 超时设为 1 分钟。

Bugzilla:2116848

### **更新了 GCC Toolset 12**

GCC Toolset 12 是一个编译器工具集，提供了开发工具的最新版本。它在 **AppStream** 存储库中以软件集合的形式作为应用程序流提供。

RHEL 9.2 中引入的显著变化包括：

- GCC 编译器已更新至版本 12.2.1，它提供了上游 GCC 中提供的很多 bug 修复和增强。
- **annobin** 已更新至版本 11.08。

GCC Toolset 12 提供了以下工具和版本：



| 工具       | 版本     |
|----------|--------|
| GCC      | 12.2.1 |
| GDB      | 11.2   |
| binutils | 2.38   |
| dwz      | 0.14   |
| annobin  | 11.08  |

要安装 GCC Toolset 12，以 root 用户身份运行以下命令：

```
# dnf install gcc-toolset-12
```

要从 GCC Toolset 12 运行工具：

```
$ scl enable gcc-toolset-12 tool
```

要运行一个 shell 会话，其中 GCC Toolset 12 中的工具版本会覆盖这些工具的系统版本：

```
$ scl enable gcc-toolset-12 bash
```

如需更多信息，请参阅 [GCC Toolset 12](#)。

[Bugzilla:2110583](#)

## 现在，更新的 GCC 编译器可用于 RHEL 9.2

系统 GCC 编译器版本 11.3.1 已更新，以包含上游 GCC 中的很多 bug 修复和增强。

GNU Compiler Collection (GCC) 提供用于使用 C、C++ 和 Fortran 编程语言开发应用程序的工具。

有关使用信息，请参阅 [RHEL 9 中开发 C 和 C++ 应用程序](#)。

[Bugzilla:2117632](#)

## LLVM Toolset rebase 到版本 15.0.7

LLVM Toolset 已更新至版本 15.0.7。主要变更包括：

- **-Wimplicit-function-declaration** 和 **-Wimplicit-int** 警告在 C99 及更新版本中默认启用。在 Clang 16 及更高版本中，这些警告将默认变为错误。

[Bugzilla:2118567](#)

## Rust Toolset rebase 到版本 1.66.1

Rust Toolset 已更新到版本 1.66.1。主要变更包括：

- **thread::scope** API 创建一个字典范围，其中本地变量可以被新生成的线程安全地借用，且这些线程都保证在范围结束前退出。

- **hint::black\_box** API 向编译器优化添加了一个障碍，这对于保留基准中的行为很有用，否则这些行为可能会被优化掉。
- **.await** 关键字现在使用 **IntoFuture** 特征进行转换，类似于 **for** 和 **Iterator** 之间的关系。
- 通用关联类型(GAT)允许特征包含具有通用参数的类型别名，对类型和生命周期启用新抽象。
- 新的 **let-else** 语句允许本地变量与条件模式匹配绑定，在模式不匹配时执行分支 **else** 块。
- 标记的块允许 **break** 语句跳到块的末尾，可以选择包含一个表达式值。
- **rust-analyzer** 是一种语言服务器协议的新实现，在很多编辑器中启用 Rust 支持。这取代了以前的 **rls** 软件包，但您可能需要调整编辑器配置以迁移到 **rust-analyzer**。
- **cargo** 有一个新的 **cargo remove** 子命令，用于从 **Cargo.toml** 中删除依赖项。

[Bugzilla:2123900](#)

## Go Toolset rebase 到版本 1.19.6

Go Toolset 已更新到版本 1.19.6。主要变更包括：

- 对以下软件包的安全修复：
  - **crypto/tls**
  - **mime/multipart**
  - **net/http**
  - **path/filepath**
- 对以下的 bug 修复：
  - **go** 命令
  - 链接器
  - 运行时
  - **crypto/x509** 软件包
  - **net/http** 软件包
  - **time** 软件包

[Bugzilla:2175173](#)

## **tzdata** 软件包现在包含 **/usr/share/zoneinfo/leap-seconds.list** 文件

在以前的版本中，**tzdata** 软件包只提供 **/usr/share/zoneinfo/leapseconds** 文件。有些应用程序依赖于 **/usr/share/zoneinfo/leap-seconds.list** 文件提供的备用格式，因此会出现错误。

有了这个更新，**tzdata** 软件包现在包含这两个文件，支持依赖任一格式的应用程序。

[Bugzilla:2157982](#)

## 4.13. 身份管理

## SSSD 支持将主目录转换为小写

有了这个增强，您现在可以配置 SSSD，来将用户主目录转换为小写。这有助于与 RHEL 环境的区分大小写的性质更好地集成。`/etc/sss/sss.conf` 文件的 `[nss]` 部分中的 `override_homedir` 选项现在可识别 `%h` 模板值。如果您使用 `%h` 作为 `override_homedir` 定义的一部分，SSSD 会用小写的用户主目录替换 `%h`。

Jira:RHELPLAN-139430

## SSSD 现在支持使用 shadow 密码策略更改 LDAP 用户密码

有了这个增强，如果您在 `/etc/sss/sss.conf` 文件中将 `ldap_pwd_policy` 设为 `shadow`，则 LDAP 用户现在可以更改存储在 LDAP 中的密码。在以前的版本中，如果 `ldap_pwd_policy` 设为 `shadow`，则密码更改将被拒绝，因为不清楚是否正在更新相应的 `shadow` LDAP 属性。

另外，如果 LDAP 服务器无法自动更新 `shadow` 属性，请在 `/etc/sss/sss.conf` 文件中将 `ldap_chpass_update_last_change` 选项设为 `True`，以表明 SSSD 要更新属性。

Bugzilla:1507035

## IdM 现在支持 min\_lifetime 参数

有了此增强，`min_lifetime` 参数已添加到 `/etc/gssproxy114.conf` 文件中。如果剩余生命周期低于这个值，则 `min_lifetime` 参数会触发续订服务票据。

默认情况下，其值为 15 秒。对于 NFS 等网络卷客户端，若要降低 KDC 暂时不可用时丢失访问权限的风险，请将此值设为 60 秒。

Bugzilla:2184333

## ipapwpolicy ansible-freeipa 模块现在支持新的密码策略选项

有了这个更新，`ansible-freeipa` 软件包中包含的 `ipapwpolicy` 模块支持额外的 `libpwquality` 库选项：

### maxrepeat

指定序列中相同字符的最大数量。

### maxsequence

指定单例字符序列的最大长度 (abcd)。

### dictcheck

检查密码是否为字典单词。

### usercheck

检查密码是否包含用户名。

如果设置了任何新密码策略选项，则密码的最小长度为 6 个字符。新密码策略设置仅应用到新密码。

在使用 RHEL 7 和 RHEL 8 服务器的混合环境中，新的密码策略设置仅在在 RHEL 8.4 及更新版本上运行的服务器中强制实施。如果用户登录到 IdM 客户端，并且 IdM 客户端与运行在 RHEL 8.3 或更早版本上的 IdM 服务器进行通信，则系统管理员设置的新密码策略要求不适用。为确保一致的行为，请将所有服务器升级到 RHEL 8.4 及更新的版本。

Jira:RHELPLAN-137416

## IdM 现在支持 ipanetgroup Ansible 管理模块

作为身份管理(IdM)系统管理员，您可以将 IdM 与 NIS 域和 `netgroup` 集成。使用 `ipanetgroup ansible-freeipa` 模块，您可以实现以下内容：

- 您可以确保现有的 IdM netgroup 包含特定的 IdM 用户、组、主机和主机组以及嵌套的 IdM netgroup。
- 您可以确保现有 IdM netgroup 中没有特定的 IdM 用户、组、主机和主机组以及嵌套的 IdM netgroup。
- 您可以确保 IdM 中存在或不存在特定的 netgroup。

Jira:RHELPLAN-137411

## 新的 `ipaclient_configure_dns_resolver` 和 `ipaclient_dns_servers` Ansible `ipaclient` 角色变量指定客户端的 DNS 解析器

在以前的版本中，当使用 `ansible-freeipa ipaclient` 角色安装身份管理(IdM)客户端时，无法在安装过程中指定 DNS 解析器。您必须在安装前配置 DNS 解析器。

有了这个增强，在使用 `ipaclient` 角色使用 `ipaclient_configure_dns_resolver` 和 `ipaclient_dns_servers` 变量安装 IdM 客户端时，您可以指定 DNS 解析器。因此，`ipaclient` 角色会修改 `resolv.conf` 文件和 `NetworkManager` 和 `systemd-resolved` 工具，以便以类似于 `ansible-freeipa ipaserver` 角色在 IdM 服务器上配置 DNS 解析器的方式，在客户端上配置 DNS 解析器。因此，在使用 `ipaclient` 角色安装 IdM 客户端时配置 DNS 现在更为高效。



### 注意

使用 `ipa-client-install` 命令行安装程序安装 IdM 客户端仍然需要在安装前配置 DNS 解析器。

Jira:RHELPLAN-137406

## 使用 `ipaclient` 角色安装带有 OTP 的 IdM 客户端不需要之前修改 Ansible 控制器

在以前的版本中，Ansible 控制器上的 `kinit` 命令是获取身份管理(IdM)客户端部署的一次性密码(OTP)的先决条件。在 Red Hat Ansible Automation Platform (AAP)上获取 OTP 是一个问题，因为默认情况下没有安装 `krb5-workstation` 软件包。

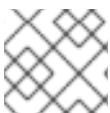
有了这个更新，对管理员的 TGT 的请求被委派给第一个指定或发现的 IdM 服务器。现在，您可以使用 OTP 授权安装 IdM 客户端，而无需额外修改 Ansible 控制器。这简化了使用带有 AAP 的 `ipaclient` 角色。

Jira:RHELPLAN-137403

## IdM 现在在 Kerberos 票据中强制存在 MS-PAC 结构

从 RHEL 9.2 开始，为了提高安全性，身份管理(IdM)和 MIT Kerberos 现在强制在 RHEL IdM Kerberos 分发中心(KDC)发布的 Kerberos 票据存在特权属性证书(MS-PAC)结构。

在 2022 年 11 月，为了响应 CVE-2022-37967，Microsoft 引入了一个扩展签名，它是通过整个 MS-PAC 结构而不是服务器校验和来计算的。从 RHEL 9.2 开始，IdM KDC 发布的 Kerberos 票据现在也包含扩展的签名。



### 注意

IdM 中还没有强制扩展签名的存在。

Jira:RHELPLAN-159146

## KDC 的新领域配置模板启用 FIPS 140-3 兼容密钥加密

此更新在 `/var/kerberos/krb5kdc/kdc.conf` 文件中提供了一个新的 **EXAMPLE.COM** 示例领域配置。它会带来两个变化：

- FIPS 140-3 兼容 **AES HMAC SHA-2** 系列被添加到密钥加密的支持类型的列表中。
- KDC 主密钥的加密类型从 **AES 256 HMAC SHA-1** 切换到 **AES 256 HMAC SHA-384**。



### 警告

这个更新是独立的 MIT 领域。不要更改 RHEL 身份管理中的 Kerberos 分发中心 (KDC) 配置。

建议为新领域使用此配置模板。模板不会影响任何已部署的领域。如果您计划根据模板升级领域的配置，请考虑以下几点：

对于升级主密钥，更改 KDC 配置中的设置不够充分。按照 MIT Kerberos 文档中描述的流程操作：  
<https://web.mit.edu/kerberos/krb5-1.20/doc/admin/database.html#updating-the-master-key>

将 **AES HMAC SHA-2** 系列添加到密钥加密的支持类型中在任何时候都安全，因为它不会影响 KDC 中的现有条目。只有在创建新主体或续订凭证时，才会生成密钥。请注意，无法根据现有密钥生成此新类型的密钥。要使这些新加密类型对某个主体可用，必须续订其凭证，这意味着也续订服务主体的 keytab。

主体不具有 **AES HMAC SHA-2** 密钥的唯一情况是活动目录(AD)跨域票据授予票据(TGT)。由于 AD 不实现 RFC8009，所以不使用 **AES HMAC SHA-2** 加密类型系列。因此，使用 **AES HMAC SHA-2** 加密的跨域 TGT 的跨域 TGS-REQ 将失败。防止 MIT Kerberos 客户端使用针对 AD 的 **AES HMAC SHA-2** 的最佳方法是不为 AD 跨域主体提供 **AES HMAC SHA-2** 密钥。要做到这一点，请确保使用 AD 支持的密钥加密类型的明确列表创建跨域 TGT 条目：

```
kadmin.local <<EOF
add_principal +requires_preauth -e aes256-cts-hmac-sha1-96,aes128-cts-hmac-sha1-96 -pw
[password] krbtgt/[MIT realm]@[AD realm]
add_principal +requires_preauth -e aes256-cts-hmac-sha1-96,aes128-cts-hmac-sha1-96 -pw
[password] krbtgt/[AD realm]@[MIT realm]
EOF
```

要确保 MIT Kerberos 客户端使用 **AES HMAC SHA-2** 加密类型，您还必须在客户端和 KDC 配置中将这些加密类型设为 **permitted**。在 RHEL 上，此设置由加密策略系统管理。例如，在使用 **DEFAULT** 加密策略的 RHEL 9 主机上允许 **AES HMAC SHA-2** 和 **AES HMAC SHA-1** 加密票据，而使用 **FIPS** 加密策略的主机只接受 **AES HMAC SHA-2** 票据。

[Bugzilla:2068535](#)

## 使用配置文件配置 pam\_pwhistory

有了这个更新，您可以在 `/etc/security/pwhistory.conf` 配置文件中配置 **pam\_pwhistory** 模块。**pam\_pwhistory** 模块保存了每个用户的最后一个密码，以管理密码更改历史记录。**authselect** 中还添加了支持，允许您向 PAM 堆栈中添加 **pam\_pwhistory** 模块。

[Bugzilla:2126640](#),[Bugzilla:2142805](#)

## IdM 现在支持新的活动目录证书映射模板

活动(AD)域管理员可以使用 **altSecurityIdentities** 属性手动将证书映射到 AD 中的用户。此属性有六个值，但三个映射现在被视为不安全。作为 [2022 年 5 月 10 日安全更新](#) 的一部分，当这个更新安装在域控制器上后，所有设备都处于兼容性模式。如果证书被弱映射到用户，则身份验证会如预期发生，但会记录一条警告信息，标识与完整执行模式不兼容的证书。从 2023 年 11 月 14 日起，所有设备都将更新为全强制模式，如果证书不符合强映射标准，则身份验证将被拒绝。

IdM 现在支持新的映射模板，使 AD 管理员更容易地使用新规则，而不维护这两个模板。IdM 现在支持以下新映射模板：

- 序列号：**LDAPU1:(altSecurityIdentities=X509:<l>{issuer\_dn!ad\_x500}<SR>{serial\_number!hex\_ur})**
- 主题键 Id: **LDAPU1: (altSecurityIdentities=X509:<SKI>{subject\_key\_id!hex\_u})**
- 用户 SID：**LDAPU1: (objectsid={sid})**

如果您不想使用新的 SID 扩展重新发布证书，您可以通过将适当的映射字符串添加到 AD 中的 **altSecurityIdentities** 属性来创建手动映射。

[Bugzilla:2087247](#)

## samba rebase 到版本 4.17.5

**samba** 软件包已升级到上游版本 4.17.5，与之前的版本相比，它提供了 bug 修复和增强。最显著的更改：

- 之前版本中的安全性改进会影响服务器消息块(SMB)服务器对高元数据数据工作负载的性能。这个更新在这种情况下提高了性能。
- **--json** 选项添加到 **smbstatus** 工具中，以显示 JSON 格式的详细的状态信息。
- **samba.smb.conf** 和 **samba.samba3.smb.conf** 模块已添加到 **smbconf** Python API 中。您可以在 Python 程序中使用它们来原生读和（可选）写 Samba 配置。

请注意，从 Samba 4.11 开始，服务器消息块版本 1 (SMB1) 协议已被弃用，并将在以后的版本中删除。

在启动 Samba 前备份数据库文件。当 **smbd**、**nmbd** 或 **winbind** 服务启动时，Samba 会自动更新其 **tdb** 数据库文件。红帽不支持降级 **tdb** 数据库文件。

更新 Samba 后，使用 **testparm** 工具来验证 **/etc/samba/smb.conf** 文件。

有关显著变化的更多信息，请在更新前阅读 [上游发行注记](#)。

[Bugzilla:2131993](#)

## ipa-client-install 现在支持使用 PKINIT 进行身份验证

在以前的版本中，**ipa-client-install** 仅支持基于密码的身份验证。这个更新支持使用 PKINIT 进行身份验证的 **ipa-client-install**。

例如：

```
ipa-client-install --pkinit-identity=FILE:/path/to/cert.pem,/path/to/key.pem --pkinit-anchor=FILE:/path/to/cacerts.pem
```

要使用 PKINIT 身份验证，您必须在 IdM 和 PKINIT 证书的 CA 链之间建立信任。如需更多信息，请参阅 **ipa-cacert-manage (1)** 手册页。此外，证书身份映射规则必须将主机的 PKINIT 证书映射到有权限添加或修改主机记录的主体。如需更多信息，请参阅 **ipa certmaprule-add** 手册页。

[Bugzilla:2143224](#)

### Red Hat IdM 和证书系统现在支持 EST 协议

通过安全传输(EST)注册是一个新的证书系统子系统功能，它在 RFC 7030 中指定，用于提供证书颁发机构(CA)的证书。EST 实现服务器端的操作，如 `/getcacerts`、`/simpleenroll` 和 `/simplereenroll`。

请注意，红帽支持证书系统中的 EST 和原始简单证书注册协议(SCEP)。

[Bugzilla:1849834](#)

### 增强负缓存使用率

这个更新提高了按安全标识符(SID)查找的 SSSD 性能。现在，它将在单个域的负缓存中存储不存在的 SID，并请求 SID 所属的域。

[Bugzilla:1766490](#)

### 目录服务器现在支持 TLS 的 ECDSA 私钥

在以前的版本中，您无法使用比 RSA 更强大的加密算法来保护目录服务器连接。有了这个增强，目录服务器现在同时支持 ECDSA 和 RSA 密钥。

[Bugzilla:2096795](#)

### 目录服务器现在支持搜索操作的扩展日志记录

在以前的版本中，访问日志中的记录不显示为何一些搜索操作有非常大的 **etime** 值。有了这个版本，您可以启用统计信息的日志，如多个索引查找（数据库读取操作）以及每个搜索操作的索引查找的整个持续时间。这些统计信息记录有助于分析为什么 **etime** 值会如此耗费资源。

[Bugzilla:1859271](#)

### NUNC\_STANS 错误日志记录级别被新的 1048576 日志记录级别替代

在以前的版本中，您无法轻松地调试密码策略问题。对错误日志使用新的 **1048576** 日志记录级别，您现在可以检查以下密码策略信息：

- 哪个本地策略拒绝或允许密码更新。
- 完全违反了语法。

[Bugzilla:2057070](#)

### 目录服务器引入了安全日志

为了为了随着时间的推移正确地跟踪问题，目录服务器现在有一个专门维护安全数据的日志。与具有所有信息的访问日志相比，安全日志不会快速轮转，且消耗较少的磁盘资源，但需要昂贵的解析来获取安全数据。

新的服务器日志记录安全事件，如身份验证事件、授权问题、DoS/TCP 攻击以及其他事件。

目录服务器将安全日志与其他日志文件一起存储在 `/var/log/dirsrv/slaped-instance_name/` 目录中。

[Bugzilla:2093981](#)



## 目录服务器现在可以压缩归档的日志文件

在以前的版本中，不压缩归档的日志文件。有了这个版本，您可以启用访问、错误、审计、审计失败日志、安全日志文件压缩来保存磁盘空间。请注意，默认只启用安全日志文件压缩。

使用 **cn=config** 条目中的以下新配置属性来管理压缩：

- 用于访问日志的 **nsslapd-accesslog-compress**
- 用于错误日志的 **nsslapd-errorlog-compress**
- 用于审计日志的 **nsslapd-auditlog-compress**
- 用于审计失败日志的 **nsslapd-auditfaillog-compress**
- 用于安全日志的 **nsslapd-securelog-compress**

Bugzilla:1132524

## 新的 **pamModuleIsThreadSafe** 配置选项现在可用

当 PAM 模块是线程安全的时，您可以通过将新的 **pamModuleIsThreadSafe** 配置选项设置为 **yes** 来提高 PAM 身份验证吞吐量和特定模块的响应时间：

```
pamModuleIsThreadSafe: yes
```

此配置应用于 PAM 模块配置条目( **cn=PAM Pass Through Auth,cn=plugins,cn=config** 的子级)。

在 **dse.ldif** 配置文件或 **ldapmodify** 命令中使用 **pamModuleIsThreadSafe** 选项。请注意，**ldapmodify** 命令要求您重启服务器。

Bugzilla:2142639

## 目录服务器现在可以导入证书包

在以前的版本中，当尝试使用 **dsconf** 或 **dsctl** 工具添加证书包时，流程会失败，并显示错误，证书包没有被导入。此行为是由 **certutil** 工具造成的，其一次只能导入一个证书。有了这个更新，目录服务器可以使用 **certutil** 临时解决问题，证书包可被成功添加。

Bugzilla:1878808

## 默认行为更改：目录服务器现在返回一个与添加到数据库时拼写完全相同的 DN

使用 **cn=config** 条目下的新的 **nsslapd-return-original-entrydn** 参数，您可以管理目录服务器如何在搜索操作期间返回条目的可分辨名称(DN)。

默认情况下，**nsslapd-return-original-entrydn** 参数被设置为 **on**，目录服务器会返回最初添加到数据库中时完全相同的 DN。例如，您添加或修改了条目 **uid=User,ou=PEople,dc=ExaMPIE,DC=COM**，并打开了设置，目录服务器会为条目返回相同拼写的 DN：**uid=User,ou=PEople,dc=ExaMPIE,DC=COM**。

当 **nsslapd-return-original-entrydn** 参数设置为 **off** 时，目录服务器通过将条目的相对 DN (RDN)与存储在 **cn=userroot,cn=ldb database,cn=plugins,cn=config** 下的数据库后缀配置中的基本 DN 放在一起，来生成条目 DN。如果将基础 DN 设置为 **ou=people,dc=example,dc=com**，将 **nsslapd-return-original-entrydn** 设置为 **off**，则目录服务器会在搜索过程中返回 **uid=User,ou=people,dc=example,dc=com**，而不是在将条目添加到数据库时 DN 的拼写。

Bugzilla:2075017



## MIT Kerberos 支持 Ticket 和 Extended KDC MS-PAC 签名

使用此更新，红帽使用的 MIT Kerberos 实现对 Microsoft 引入的两种 Privilege Attribute 证书(PAC)签名类型的支持，以响应最新的 CVE。具体来说，支持以下签名：

- 票据签名
  - 已在 [KB4598347](#) 中发布
  - 解决 [CVE-2020-17049](#)，也称为 "Bronze-Bit" 攻击
- 扩展的 KDC 签名
  - 已在 [KB5020805](#) 中发布
  - 解决 [CVE-2022-37967](#)

另请参阅 [RHSA-2023:2570](#) 和 [krb5-1.20.1-6.el9](#)。

[Bugzilla:2165827](#)

## 用于目录服务器审计日志的新的 `nsslapd-auditlog-display-attrs` 配置参数

在以前的版本中，区分名称(DN)是在审计日志事件中识别目标条目的唯一方法。使用新的 `nsslapd-auditlog-display-attrs` 参数，您可以将目录服务器配置为在审计日志中显示额外的属性，提供有关修改的条目的更多详情。

例如，如果您将 `nsslapd-auditlog-display-attrs` 参数设置为 `cn`，则审计日志会在输出中显示条目 `cn` 属性。要包含修改的条目的所有属性，请使用星号(\*)作为参数值。

如需更多信息，请参阅 [nsslapd-auditlog-display-attrs](#)。

[Bugzilla:2136610](#)

## 4.14. 桌面

### 禁用滑动切换工作区

在以前的版本中，用三个手指向上或向下滑动总是能在触摸屏上切换工作区。有了这个版本，您可以禁用工作区切换。

详情请参阅 [禁用滑动切换工作区](#)。

[Bugzilla:2154358](#)

### Wayland 现在在 Aspeed GPU 上启用

在以前的版本中，Aspeed GPU 驱动程序表现不够好，不能运行 Wayland 会话。要临时解决这个问题，对 Aspeed GPU 禁用 Wayland 会话。

有了这个版本，驱动程序性能显著提高，Wayland 会话现在有响应。因此，现在默认在 Aspeed GPU 上启用 Wayland 会话。

[Bugzilla:2131203](#)

### 在桌面上自定义右键菜单

现在，您可以自定义在右键点击桌面背景时打开的菜单。您可以在运行任意命令的菜单中创建自定义条目。

要自定义菜单，请参阅 [自定义桌面上的右键菜单](#)。

[Bugzilla:2160553](#)

## 4.15. WEB 控制台

现在，Web 控制台中提供了某些加密子策略

这个 RHEL web 控制台的更新扩展了 **Change crypto policy** 对话框中的选项。除了四个系统范围的加密策略外，您现在还可以通过图形界面应用以下子策略：

- **DEFAULT:SHA1** 是启用了 **SHA-1** 算法的 **DEFAULT** 策略。
- **LEGACY:AD-SUPPORT** 是 **LEGACY** 策略，其具有较少的安全设置，提高了活动目录服务的互操作性。
- **FIPS:OSPP** 是 **FIPS** 策略，具有由常见标准处理信息技术安全评估标准启发的进一步的限制。

Jira:RHELPLAN-137505

**Web 控制台现在执行额外的步骤，将 LUKS 加密根卷绑定到 NBDE**

有了这个更新，RHEL web 控制台执行将 LUKS 加密的根卷绑定到网络绑定磁盘加密(NBDE)部署所需的额外步骤。选择加密的根文件系统和 Tang 服务器后，您可以跳过将 **rd.neednet=1** 参数添加到内核命令行中，安装 **clevis-dracut** 软件包，以及重新生成初始的 ramdisk (**initrd**)。对于非 root 文件系统，web 控制台现在启用 **remote-cryptsetup.target** 和 **clevis-luks-akspass.path systemd** 单元，安装 **clevis-systemd** 软件包，并将 **\_netdev** 参数添加到 **fstab** 和 **crypttab** 配置文件中。现在，在为自动解锁 LUKS 加密的根卷创建 NBDE 部署时，您可以对所有 Clevis-client 配置步骤使用图形界面。

Jira:RHELPLAN-139125

## 4.16. RED HAT ENTERPRISE LINUX 系统角色

**路由规则可根据其名称查找路由表**

有了此更新，**rhel-system-roles.network** RHEL 系统角色支持在定义路由规则时按名称查找路由表。此功能为复杂网络配置提供快速导航，其中您需要对不同的网络段有不同的路由规则。

[Bugzilla:2131293](#)

**network 系统角色支持设置 DNS 优先级值**

此增强向 RHEL **network** 系统角色中添加了 **dns\_priority** 参数。您可以将此参数设为 **-2147483648** 到 **2147483647** 的值。默认值为 **0**。较低的值具有较高的优先级。请注意，负值导致系统角色排除具有更大数字优先级值的其他配置。因此，如果存在至少一个负优先级值，系统角色只使用来自连接配置文件的具有最低优先级值的 DNS 服务器。

因此，您可以使用 **network** 系统角色在不同的连接配置文件中定义 DNS 服务器的顺序。

[Bugzilla:2133858](#)

**vpn RHEL 系统角色的新 IPsec 自定义参数**

因为某些网络设备需要 IPsec 自定义才能正常工作，因此以下参数已添加到 **vpn** RHEL 系统角色中：



## 重要

不要在没有较深理解的情况下更改以下参数。大多数场景不需要自定义。

此外，出于安全原因，使用 Ansible Vault 加密 **shared\_key\_content** 参数的值。

- 隧道参数：
  - **shared\_key\_content**
  - **ike**
  - **esp**
  - **ikelifetime**
  - **salifetime**
  - **retransmit\_timeout**
  - **dpddelay**
  - **dpdtimeout**
  - **dpdaction**
  - **leftupdown**
- 每个主机参数：
  - **leftid**
  - **rightid**

因此，您可以使用 **vpn** 角色配置到大量网络设备的 IPsec 连接。

[Bugzilla:2119102](#)

## selinux RHEL 系统角色现在支持 local 参数

这个 **selinux** RHEL 系统角色的更新引进了对 **local** 参数的支持。通过使用此参数，您只能删除本地策略修改，并保留内置的 SELinux 策略。

[Bugzilla:2128843](#)

## ha\_cluster 系统角色现在支持自动化执行 firewall, selinux, 和 certificate 系统角色

ha\_cluster RHEL 系统角色现在支持以下功能：

### 使用 firewall 和 selinux 系统角色管理端口访问

要将集群的端口配置为运行 **firewalld** 和 **selinux** 服务，您可以将新的角色变量 **ha\_cluster\_manage\_firewall** 和 **ha\_cluster\_manage\_selinux** 设为 **true**。这将集群配置为使用 **firewall** 和 **selinux** 系统角色，并在 **ha\_cluster** 系统角色中自动化并执行这些操作。如果将这些变量设为默认值 **false**，则不会执行角色。有了这个版本，防火墙不再会被默认配置，因为它仅在 **ha\_cluster\_manage\_firewall** 设为 **true** 时进行配置。

### 使用 certificate 系统角色创建 pcsd 私钥和证书对

**ha\_cluster** 系统角色现在支持 **ha\_cluster\_pcsd\_certificates** 角色变量。设置此变量，将其值传给 **certificate** 系统角色的 **certificate\_requests** 变量。这提供了为 **pcsd** 创建私钥和证书对的替代方法。

[Bugzilla:2130010](#)

**postfix** RHEL 系统角色现在可以使用 **firewall** 和 **selinux** RHEL 系统角色来管理端口访问

有了这个增强，您可以使用新角色变量 **postfix\_manage\_firewall** 和 **postfix\_manage\_selinux** 自动管理端口访问：

- 如果它们设为 **true**，则每个角色都可用来管理端口访问。
- 如果它们设为 **false**（默认），则角色不参与。

[Bugzilla:2130329](#)

**vpn** RHEL 系统角色现在可以使用 **firewall** 和 **selinux** 角色来管理端口访问

有了这个增强，您可以通过 **firewall** 和 **selinux** 角色，在 **vpn** RHEL 系统角色中自动管理端口访问。如果将新角色变量 **vpn\_manage\_firewall** 和 **vpn\_manage\_selinux** 设为 **true**，则角色将管理端口访问。

[Bugzilla:2130344](#)

**logging** RHEL 系统角色现在支持端口访问和证书的生成

有了这个增强，您可以使用 **logging** 角色来管理端口访问，并使用新角色变量生成证书。如果将新角色变量 **logging\_manage\_firewall** 和 **logging\_manage\_selinux** 设为 **true**，则角色将管理端口访问。用于生成证书的新角色变量是 **logging\_certificates**。类型和用法与 **certificate** 角色 **certificate\_requests** 相同。现在，您可以使用 **logging** 角色直接自动化执行这些操作。

[Bugzilla:2130357](#)

**metrics** RHEL 系统角色现在可以使用 **firewall** 角色和 **selinux** 角色来管理端口访问

有了这个增强，您可以控制对端口的访问。如果将新角色变量 **metrics\_manage\_firewall** 和 **metrics\_manage\_selinux** 设为 **true**，则角色将管理端口访问。现在，您可以使用 **metrics** 角色直接自动化并执行这些操作。

[Bugzilla:2133528](#)

**nbde\_server** RHEL 系统角色现在可以使用 **firewall** 和 **selinux** 角色来管理端口访问

有了这个增强，您可以使用 **firewall** 和 **selinux** 角色管理端口访问。如果将新角色变量 **nbde\_server\_manage\_firewall** 和 **nbde\_server\_manage\_selinux** 设为 **true**，则角色将管理端口访问。现在，您可以使用 **nbde\_server** 角色直接自动化这些操作。

[Bugzilla:2133930](#)

**initscripts** 网络供应商支持默认网关的路由指标配置

有了此更新，您可以在 **rhel-system-roles.network** RHEL 系统角色中使用 **initscripts** 网络提供者来配置默认网关的路由指标。

此类配置的原因可能是：

- 在不同路径中分发流量负载
- 指定主路由和备份路由

- 利用路由策略通过特定路径将流量发送到特定的目的地

[Bugzilla:2134202](#)

### cockpit RHEL 系统角色与 firewall、selinux 和 certificate 角色集成

此功能增强使您能够将 **cockpit** 角色与 **firewall** 角色和 **selinux** 角色集成，来管理端口访问和 **certificate** 角色集成来生成证书。

要控制端口访问，请使用新的 **cockpit\_manage\_firewall** 和 **cockpit\_manage\_selinux** 变量。默认情况下，这两个变量都默认设为 **false**，并且不会被执行。将它们设为 **true**，以允许 **firewall** 和 **selinux** 角色管理 RHEL web 控制台服务端口访问。然后，操作将在 **cockpit** 角色内执行。

请注意，您负责管理防火墙和 SELinux 的端口访问。

要生成证书，请使用新的 **cockpit\_certificates** 变量。变量默认设为 **false**，并且不会被执行。您可以与在 **certificate** 角色中使用 **certificate\_request** 变量的相同方式使用此变量。然后 **cockpit** 角色将使用 **certificate** 角色来管理 RHEL web 控制台证书。

[Bugzilla:2137663](#)

### 用于直接与活动目录集成的新的 RHEL 系统角色

新的 **rhel-system-roles.ad\_integration** RHEL 系统角色已添加到 **rhel-system-roles** 软件包中。因此，现在管理员可以自动将 RHEL 系统直接与活动目录域集成。

[Bugzilla:2140795](#)

### 新的 Red Hat Insights 和订阅管理的 Ansible 角色

**rhel-system-roles** 软件包现在包含远程主机配置(**rhc**)系统角色。此角色使管理员能够轻松地将 RHEL 系统注册到 Red Hat Subscription Management (RHSM)和 Satellite 服务器。默认情况下，当使用 **rhc** 系统角色注册系统时，系统会连接到 Red Hat Insights。有了新的 **rhc** 系统角色，管理员现在可以在受管节点上自动执行以下任务：

- 配置到 Red Hat Insights 的连接，包括系统的自动更新、补救和标签。
- 启用和禁用存储库。
- 配置用于连接的代理。
- 设置系统的发行版本。

有关如何自动化这些任务的更多信息，请参阅 [使用 RHC 系统角色注册系统](#)。

[Bugzilla:2141330](#)

### 添加了对克隆的 MAC 地址的支持

克隆的 MAC 地址是设备 WAN 端口的 MAC 地址，它与机器的 MAC 地址相同。有了这个更新，用户可以使用 MAC 地址或策略（如 **random** 或 **preserve**）指定绑定或网桥接口，以获取绑定或网桥接口的默认 MAC 地址。

[Bugzilla:2143768](#)

### Microsoft SQL Server Ansible 角色支持异步高可用性副本

在以前的版本中，Microsoft SQL Server Ansible 角色只支持主、同步和见证高可用性副本。现在，您可以将 **mssql\_ha\_replica\_type** 变量设为 **asynchronous**，以使用新的或现有副本的异步副本类型对其进行配置。

[Bugzilla:2151282](#)

### Microsoft SQL Server Ansible 角色支持 read-scale 集群类型

在以前的版本中，Microsoft SQL Ansible 角色只支持外部集群类型。现在，您可以使用新的变量 **mssql\_ha\_ag\_cluster\_type** 配置角色。默认值为 **external**，使用它来配置具有 Pacemaker 的集群。要配置没有 Pacemaker 的集群，请对该变量使用值 **none**。

[Bugzilla:2151283](#)

### Microsoft SQL Server Ansible 角色可以生成 TLS 证书

在以前的版本中，您需要在配置 Microsoft SQL Ansible 角色前，手动在节点上生成 TLS 证书和私钥。有了这个更新，Microsoft SQL Server Ansible 角色可以使用 **redhat.rhel\_system\_roles.certificate** 角色来实现这一目的。现在，您可以使用 **certificate** 角色的 **certificate\_requests** 变量格式设置 **mssql\_tls\_certificates** 变量，以便在节点上生成 TLS 证书和私钥。

[Bugzilla:2151284](#)

### Microsoft SQL Server Ansible 角色支持配置 SQL Server 版本 2022

在以前的版本中，Microsoft SQL Ansible 角色只支持配置 SQL Server 版本 2017 和版本 2019。这个更新为您提供对 Microsoft SQL Ansible 角色的 SQL Server 版本 2022 的支持。现在，您可以将 **mssql\_version** 值设为 **2022**，以配置新的 SQL Server 2022 或将 SQL Server 从版本 2019 升级到版本 2022。请注意，不提供从 SQL Server 版本 2017 到版本 2022 的升级。

[Bugzilla:2153428](#)

### Microsoft SQL Server Ansible 角色支持活动目录身份验证的配置

有了这个更新，Microsoft SQL Ansible 角色支持 SQL Server 的活动目录身份验证的配置。现在，您可以使用 **mssql\_ad\_** 前缀设置变量来配置活动目录身份验证。

[Bugzilla:2163709](#)

### journald RHEL 系统角色现在可用

**journald** 服务收集日志数据并将其存储在集中式数据库中。有了此增强，您可以使用 **journald** 系统角色变量自动化 **systemd** 日志的配置，并使用 Red Hat Ansible Automation Platform 配置持久性日志记录。

[Bugzilla:2165175](#)

### ha\_cluster 系统角色现在支持仲裁设备配置

仲裁设备充当集群的第三方仲裁设备。对于偶数节点的集群，建议使用仲裁设备。对于双节点集群，使用仲裁设备可以更好地决定在脑裂情况下保留哪些节点。您现在可以使用 **ha\_cluster** 系统角色配置仲裁设备，**qdevice** 用于集群，**qnetd** 用于仲裁节点。

[Bugzilla:2140804](#)

## 4.17. 虚拟化

### 硬件加密设备现在可以自动热插拔

在以前的版本中，如果在介质设备启动前它们在主机上存在，则只能为 **passthrough** 定义加密设备。现



在，您可以定义一个介质设备阵列，列出您要传递给虚拟机(VM)的所有加密设备。因此，如果指定的加密设备在以后可用，则它们会自动传递给正在运行的虚拟机。另外，如果设备不可用，它们会从虚拟机中删除，但客户机操作系统会保持正常运行。

Bugzilla:1871126

### 提高了 IBM Z 上 PCI 透传设备的性能

有了这个更新，IBM Z 硬件上的 PCI passthrough 实现已通过对 I/O 处理的多个改进得到了增强。因此，传递到 IBM Z 主机上的 KVM 虚拟机(VM)的 PCI 设备现在性能显著提高。

另外，ISM 设备现在可以分配给 IBM Z 主机上的虚拟机。

Bugzilla:1871143

### 新软件包：passt

此更新添加了 **passt** 软件包，这样就可以为虚拟机使用 **passt** 用户模式网络后端。

有关使用 **passt** 的更多信息，请参阅 [配置 passt 用户空间连接](#)。

Bugzilla:2131015

### zPCI 设备分配

现在，可以将 zPCI 设备作为 passthrough 设备附加到在 IBM Z 硬件上运行 RHEL 的虚拟机(VM)。例如，这允许在虚拟机中使用 NVMe 闪存驱动器。

Jira:RHELPLAN-59528

### 新软件包：python-virt-firmware

此更新添加了 **python-virt-firmware** 软件包，其包含处理 Open Virtual Machine Firmware (OVMF)固件镜像的工具。您可以对以下情况使用这些工具：

- 打印固件镜像的内容
- 更新 **edk2** 变量存储
- 处理安全引导密钥保护，而不在 QEMU 中启动虚拟机

因此，这使构建 OVMF 镜像更容易

Bugzilla:2089785

## 4.18. 支持性

### sos 工具移到 4 周更新节奏

不是与 RHEL 次版本一起发布 **sos** 更新，**sos** 工具发行节奏从 6 个月改为 4 周。您可以在每 4 周的 RPM 更改日志中找到 **sos** 软件包的更新详情，也可以每 6 个月在 RHEL 发行注记中阅读 **sos** 更新概述。

[Bugzilla:2164987](#)

### sos clean 命令现在模糊处理 IPv6 地址

在以前的版本中，**sos clean** 命令不会模糊处理 IPv6 地址，将一些客户敏感数据留在收集的 **sos** 报告中。有了这个更新，**sos clean** 会预期检测并模糊处理 IPv6 地址。

[Bugzilla:2134906](#)

## 4.19. 容器

### 新的 podman RHEL 系统角色现在可用

从 Podman 4.2 开始，您可以使用 **podman** 系统角色来管理 Podman 配置、容器以及运行 Podman 容器的 **systemd** 服务。

Jira:RHELPLAN-118705

### Podman 现在支持审计事件

从 Podman v4.4 开始，您可以直接从单个事件和 **journald** 条目中收集关于容器的所有相关信息。要启用 Podman 审计，请修改 **container.conf** 配置文件，并将 **events\_container\_create\_inspect\_data=true** 选项添加到 **[engine]** 部分中。数据采用 JSON 格式，与 **podman container inspect** 命令的格式相同。如需更多信息，请参阅 [如何在 Podman 4.4 中使用新的容器事件和审计功能](#)。

Jira:RHELPLAN-136602

### container-tools meta-package 已更新

**container-tools** RPM 元软件包现已正式发布，其中包括 Podman、Buildah、Skopeo，crun 和 runc 工具现在可用。与之前的版本相比，这个版本应用了一系列 bug 修复和增强。

Podman v4.4 中的显著变化包括：

- 引入了 Quadlet，这是一个新的 systemd 生成器，它可使用 Podman 轻松创建和维护 systemd 服务。
- 添加了一个新命令 **podman network update**，它会更新容器和 pod 的网络。
- 添加了一个新命令 **podman buildx version**，它会显示 buildah 版本。
- 容器现在可以启动健康检查，允许运行命令来确保在激活常规健康检查前完全启动容器。
- 使用 **podman --dns** 命令支持自定义 DNS 服务器选择。
- 现在，提供使用 Fulcio 和 Rekor 创建并验证 sigstore 签名。
- 改进了与 Docker 的兼容性（新选项和别名）。
- 改进了 Podman 的 Kubernetes 集成 - 命令 **podman kube generate** 和 **podman kube play** 现在可用，替换了 **podman generate kube** 和 **podman play kube** 命令。**podman generate kube** 和 **podman play kube** 命令仍可用，但建议使用新的 **podman kube** 命令。
- systemd 管理的由 **podman kube play** 命令创建的 pod 现在与 sd-notify 集成，使用 **io.containers.sdnotify** 注解（或对特定容器使用 **io.containers.sdnotify/\$name**）。
- 现在，由 **podman kube play** 创建的 systemd 管理的 pod 可以使用 **io.containers.auto-update** 注解（或使用特定容器的 **io.containers.auto-update/\$name**）自动更新。

Podman 已升级至版本 4.4，要了解有关显著变化的更多信息，请参阅 [上游发行注记](#)。

Jira:RHELPLAN-136607

### Aardvark 和 Netavark 现在支持自定义 DNS 服务器选择



Aardvark 和 Netavark 网络堆栈现在支持容器的自定义 DNS 服务器选择，而不是主机上的默认 DNS 服务器。您有两个用于指定自定义 DNS 服务器的选项：

- 在 **containers.conf** 配置文件中添加 **dns\_servers** 字段。
- 使用新的 **--dns** Podman 选项指定 DNS 服务器的 IP 地址。

**--dns** 选项会覆盖 **container.conf** 文件中的值。

Jira:RHELPLAN-138024

### Skopeo 现在支持生成 sigstore 密钥对

您可以使用 **skopeo generate-sigstore-key** 命令生成一个 sigstore 公钥/私钥对。如需更多信息，请参阅 **skopeo-generate-sigstore-key** 手册页。

Jira:RHELPLAN-151481

### Toolbox 现在可用

有了 **toolbox** 工具，您可以在系统上使用容器化的命令行环境，而无需直接安装故障排除工具。Toolbox 构建在 Podman 和其他来自 OCI 的标准容器技术之上。如需更多信息，请参阅 [toolbox](#)。

Jira:RHELPLAN-150266

### 容器镜像现在有一个两位标签

在 RHEL 9.0 和 RHEL 9.1 中，容器镜像有一个三位标签。从 RHEL 9.2 开始，容器镜像现在有一个两位标签。

Jira:RHELPLAN-147982

### 提供签名镜像的多个可信的 GPG 密钥功能

**/etc/containers/policy.json** 文件支持一个新的 **keyPaths** 字段，该字段接受一个包含可信密钥的文件的列表。因此，在默认配置中现在接受使用红帽正式发行(GA)和 Beta GPG 密钥签名的容器镜像。

例如：

```
"registry.redhat.io": [
  {
    "type": "signedBy",
    "keyType": "GPGKeys",
    "keyPaths": ["/etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release", "/etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-beta"]
  }
]
```

Jira:RHELPLAN-129327

### Podman 现在支持预执行钩子

位于 **/usr/libexec/podman/pre-exec-hooks** 和 **/etc/containers/pre-exec-hooks** 目录中的 root 拥有的插件脚本定义了对容器操作的精细控制，特别是阻止了未授权的操作。

**/etc/containers/podman\_preexec\_hooks.txt** 文件必须由管理员创建，并可为空。如果 **/etc/containers/podman\_preexec\_hooks.txt** 不存在，则不会执行插件脚本。如果所有插件脚本都返回零值，则执行 **podman** 命令，否则 **podman** 命令将退出，并具有继承的退出码。

红帽建议使用以下命名约定以便以正确的顺序执行脚本：**DDD-plugin\_name.lang**，如 **010-check-group.py**。请注意，插件脚本在创建时是有效的。在插件脚本之前创建的容器不会受到影响。

Bugzilla:2119200

### sigstore 签名现在可用

从 Podman 4.2 开始，您可以使用容器镜像签名的 sigstore 格式。sigstore 签名会与容器镜像一起存储在容器注册中心，而无需有单独的签名服务器来存储镜像签名。

Jira:RHELPLAN-74672

### Toolbox 可以创建 RHEL 9 容器

在以前的版本中，Toolbox 工具只支持 RHEL UBI 8 镜像。有了这个版本，Toolbox 现在还支持 RHEL UBI 9。因此，您可以创建基于 RHEL 8 或 9 的 Toolbox 容器。

以下命令基于与主机系统相同的 RHEL 版本创建了一个 RHEL 容器：

```
$ toolbox create
```

另外，您可以使用特定的 RHEL 版本创建容器。例如，要创建基于 RHEL 9.2 的容器，请使用以下命令：

```
$ toolbox create --distro rhel --release 9.2
```

Bugzilla:2163752

### 新软件包：passt

此更新添加了 **passt** 软件包，这样就可以为容器使用 **pasta** 无根网络后端。

与 **Slirp** 连接（当前被 Podman 用作非特权网络的默认连接）相比，**pasta** 提供了以下改进：

- 改进了 IPv6 吞吐量及更好的支持，其中包括对邻居发现协议(NDP)和 DHCPv6 的支持
- 能够配置 IPv6 上 TCP 和 UDP 端口的端口转发

要使用 **pasta** 连接到 Podman 容器，请使用 **--network pasta** 命令行选项。

Bugzilla:2209419

## 第 5 章 对外部内核参数的重要更改

本章为系统管理员提供了与 Red Hat Enterprise Linux 9.2 一起发布的内核中的显著变化的总结。这些更改包括添加或更新的 **proc** 条目、**sysctl** 和 **sysfs** 默认值、引导参数、内核配置选项或者任何可见的行为更改。

### 新内核参数

#### nomodeset

使用这个内核参数，您可以禁用内核模式设置。DRM 驱动程序将不执行显示模式更改或加速渲染。如果这是通过固件或引导装载程序进行了设置，则只有系统帧缓冲才可用。

**nomodeset** 可用作回退，或测试和调试。

#### printk.console\_no\_auto\_verbose

使用这个内核参数，您可以禁用对 oops、panic 或 lockdep-detected 问题的控制台日志级别（仅在 lock debug 是 on 时）。除了在串行控制台上设置低波特率的例外，将此参数设为 **0** 以提供更多调试信息。

- 格式：**<bool>**
- 默认为 **0** (**auto\_verbose** 被启用)

#### rcupdate.rcu\_exp\_cpu\_stall\_timeout=[KNL]

使用这个内核参数，您可以为加速的 RCU CPU 失速警告信息设置超时。该值以毫秒为单位，允许的最大值为 21000 毫秒。

请注意，这个值被调整为拱形计时器刻度分辨率。把它设为零会导致使用 **rcupdate.rcu\_cpu\_stall\_timeout** 的值（在从秒转换为毫秒后）。

#### rcupdate.rcu\_task\_stall\_info=[KNL]

使用这个参数，您可以为 RCU 任务停滞信息性消息设置初始超时(以jiffies为单位)，对于那些没有足够耐心等待 10 分钟的情况，它会给出一些问题的指示。信息性消息仅在给定宽限期的失速警告消息前打印。使用值小于或等于零的值禁用。

- 默认值为 **10 秒**。
- 值的更改不会生效，直到下一个宽限期开始。

#### rcupdate.rcu\_task\_stall\_info\_mult=[KNL]

这个参数是给定 RCU 任务宽限期内连续的 RCU 任务停滞信息之间时间间隔的倍数。这个值被限制在 1 到 10（包括 1 和 10）。

它默认为 3，因此第一条信息性消息在进入宽限期 10 秒后打印，第二条消息在 40 秒时打印，第三条消息在 160 秒时打印，然后 600 秒时的停滞警告将防止 640 秒时的第四条消息。

#### smp.csd\_lock\_timeout=[KNL]

使用这个参数，您可以指定 **smp\_call\_function()** 和 friends 将等待 CPU 释放 CSD 锁的时间周期（以毫秒为单位）。这在诊断涉及 CPU 长时间禁用中断的 bug 时很有用。

- 默认值为 **5,000 毫秒**。
- 设置 **0** 代表禁用此功能。
- 使用 **csdlock\_debug**- 内核参数可以更有效地禁用这个功能。

**srcutree.big\_cpu\_lim=[KNL]**

使用这个参数，您可以指定构成大型系统的 CPU 数量，以便 **srcu\_struct** 结构应该立即分配一个 **srcu\_node** 数组。

- 默认值为 **128**。
- 仅在低顺序四位 **srcutree.convert\_to\_big** 等于 **3**（在引导时决定）时生效。

**srcutree.convert\_to\_big=[KNL]**

使用这个参数，您可以指定在什么情况下 SRCU 树 **srcu\_struct** 结构将转换为大的形式，即具有 **rcu\_node** 树：

- 0: Never。
- 1: 在 **init\_srcu\_struct()** 时间。
- 2: 当 **rcutorture** 决定时。
- 3: 在引导时（默认）决定。
- 0x1X: 如果竞争激烈，则加上以上。  
任何一种情况，**srcu\_node** 树都将根据实际的 CPU 数量(**nr\_cpu\_ids**)，而不是编译时间 **CONFIG\_NR\_CPUS** 来决定。

**srcutree.srcu\_max\_nodelay=[KNL]**

使用这个参数，您可以指定每个 jiffy 的无延迟实例数量，SRCU 宽限期 worker 线程将使用零延迟重新调度。除了这个限制外，worker 线程将使用睡眠延迟为一个 jiffy 重新调度。

**srcutree.srcu\_max\_nodelay\_phase=[KNL]**

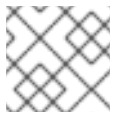
使用这个参数，您可以指定每个宽限期阶段，读取器的非休眠轮询的数量。除这个限制外，对于宽限期阶段，在读取器的每个重新扫描之间，宽限期 worker 线程将使用一个 jiffy 的睡眠延迟重新调度。

**srcutree.srcu\_retry\_check\_delay=[KNL]**

使用这个参数，您可以指定每个读取器的非休眠轮询之间非睡眠延迟的毫秒数。

**srcutree.small\_contention\_lim=[KNL]**

使用这个参数，您可以在启动 **srcu\_struct** 结构到大格式的转换前容许每个 jiffy 的更新侧争用事件的数量。

**注意**

**srcutree.convert\_to\_big** 的值必须设置 0x10 位，以便发生基于竞争的转换。

**更新的内核参数****crashkernel=size[KMG][@offset[KMG]]**

[KNL] 使用 **kexec**，Linux 可以在 panic 时切换到崩溃内核。此参数为那个内核镜像保留物理内存区域 [offset, offset + size]。如果省略 **@offset**，则会自动选择合适的偏移量。

[KNL, X86-64, ARM64] 首先选择 4G 下的区域，并在未指定 **@offset** 时回退到 4G 以上的保留区域。

如需了解更多详细信息，请参阅 **Documentation/admin-guide/kdump/kdump.rst**。

**crashkernel=size[KMG],low**

- [KNL, X86-64, ARM64] 使用此参数，您可以为第二个内核指定 4G 以下的低范围。当 **crashkernel=X,high** 被传递时，这需要一些低内存，例如 **swiotlb** 需要至少 64M+32K 低内存，还需要足够的额外低内存以确保 32 位设备的 DMA 缓冲区不会耗尽。内核会尝试自动分配 4G 以下的默认内存大小。默认大小取决于平台。
  - x86:  $\max(\text{swiotlb\_size\_or\_default}() + 8\text{MiB}, 256\text{MiB})$
  - arm64: 128MiB
  - **0** : 禁用低分配。

当 **crashkernel=X,high** 未使用或保留的内存低于 4G 时，将忽略此参数。

- [KNL, ARM64] 使用此参数，您可以为崩溃转储内核指定 DMA 区域中的低范围。当不使用 **crashkernel=X,high** 时，将忽略此参数。

### deferred\_probe\_timeout=[KNL]

使用这个参数，您可以设置延迟探测的超时时间（以秒为单位），以放弃等待依赖项探测。只有已选择的特定依赖项（子系统或驱动程序）才会被忽略。

超时为 **0** 将在 `initcall` 结束时超时。如果超时没有过期，则选项将被每个成功的驱动程序注册重启。这个选项还会在重试后转储仍然在延迟探测列表中的设备。

### driver\_async\_probe=[KNL]

使用这个参数，您可以列出异步探测到的驱动程序名称。\*（星号）与所有驱动程序名称匹配。

- 如果指定了 \*，则列出的驱动程序名称的其余部分是那些不匹配 \* 的名称。  
格式：`<driver_name1>,<driver_name2>...`

### hugetlb\_cma=[HW,CMA]

使用这个参数，您可以指定用于分配 gigantic 巨页的 CMA 区域的大小。或使用节点格式，每个节点的 CMA 区域的大小。

格式：`nn[KMGTPe]` 或(节点格式)`<node>:nn[KMGTPe][,<node>:nn[KMGTPe]]`

保留给定大小的 CMA 区域，并使用 CMA 分配器分配 gigantic 巨页。如果启用了，则会跳过 gigantic 巨页的引导时分配。

### hugepages=[HW]

使用这个参数，您可以指定要在启动时分配的 HugeTLB 页数。

- 如果这跟在 `hugepagesz` 后面，则它指定要分配的 `hugepagesz` 的页数。
- 如果这是命令行中的第一个 HugeTLB 参数，它会指定要为默认的巨页大小分配的页数。
- 如果使用节点格式，可以指定每个节点要分配的页数。  
另请参阅 [Documentation/admin-guide/mm/hugetlbpage.rst](#)。

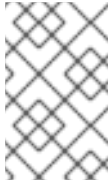
格式：`<integer>` 或（节点格式）`<node>:<integer>[,<node>:<integer>]`

### hugetlb\_free\_vmemmap=[KNL]

此参数需要启用 `CONFIG_HUGETLB_PAGE_OPTIMIZE_VMEMMAP`。允许大量 hugetlb 用户释放更多内存（每个 2MB hugetlb 页  $7 * \text{PAGE\_SIZE}$ ）。

- 格式：`{ [oO][Nn]/Y/y/1 | [oO][Ff]/N/n/0 (default) }`
- `[OO][Nn]/Y/y/1`: 启用该功能

- [OO][Ff]/N/n/0: 禁用该功能  
使用 `CONFIG_HUGETLB_PAGE_OPTIMIZE_VMEMMAP_DEFAULT_ON=y` 构建,  
默认为 `on`。



### 注意

这个参数与 `memory_hotplug.memmap_on_memory` 不兼容。如果启用了这两个参数，则 `hugetlb_free_vmemmap` 优先于 `memory_hotplug.memmap_on_memory`。

### `ivrs_ioapic=[HW,X86-64]`

此参数提供对 IVRS ACPI 表中提供的 IOAPIC-ID <-> DEVICE-ID 映射的覆盖。  
默认情况下，PCI 段为 `0`，可以省略。例如，

- 要将 IOAPIC-ID 十进制 10 映射到 PCI 设备 00:14.0，请将参数写为：

```
ivrs_ioapic[10]=00:14.0
```

- 要将 IOAPIC-ID 十进制 10 映射到 PCI 段 0x1 和 PCI 设备 00:14.0，请将参数写为：

```
ivrs_ioapic[10]=0001:00:14.0
```

### `ivrs_hpet=[HW,X86-64]`

此参数提供对 IVRS ACPI 表中提供的 HPET-ID <-> DEVICE-ID 映射的覆盖。  
默认情况下，PCI 段为 `0`，可以省略。例如：

- 要将 HPET-ID 十进制 0 映射到 PCI 设备 00:14.0，请将参数写为：

```
ivrs_hpet[0]=00:14.0
```

- 要将 HPET-ID 十进制 10 映射到 PCI 段 0x1 和 PCI 设备 00:14.0，请将参数写为：

```
ivrs_ioapic[10]=0001:00:14.0
```

### `ivrs_acpihid=[HW,X86-64]`

此参数提供对 IVRS ACPI 表中提供的 ACPI-HID:UID <-> DEVICE-ID 映射的覆盖。

例如，要将 `UART-HID:UID AMD0020:0` 映射到 PCI 段 0x1 和 PCI 设备 ID 00:14.5，请将参数写为：

```
ivrs_acpihid[0001:00:14.5]=AMD0020:0
```

默认情况下，PCI 段为 `0`，可以省略。例如，对于 PCI 设备 00:14.5，将参数写为：

```
ivrs_acpihid[00:14.5]=AMD0020:0
```

### `kvm.eager_page_split=[KVM,X86]`

使用这个参数，您可以控制 KVM 是否尝试在脏日志记录期间主动分割所有巨页。

`eager` 页分割通过消除写保护错误和 MMU 锁争用来减少对 vCPU 执行的中断，否则需要延迟分割巨页。很少执行写或只写入虚拟机内存小区域的虚拟机工作负载可能会受益于禁用 `eager` 页分割，从而使巨页仍被用于读。

eager 页分割的行为取决于是否启用或禁用了 **KVM\_DIRTY\_LOG\_INITIALLY\_SET**。

- 如果禁用了，则 memslot 中的所有巨页在对那个 memslot 启用了脏日志记录时被急切地分割。
- 如果启用了，则在 **KVM\_CLEAR\_DIRTY** ioctl 期间将执行 eager 页分割，并且仅对正在清除的页执行 eager 页分割。  
只有 **kvm.tdp\_mmu=Y** 才支持 eager 页分割。

默认为 **Y** (on)。

### kvm-arm.mode=[KVM,ARM]

使用这个参数，您可以选择 KVM/arm64 操作模式之一。

- none : 强制禁用 KVM。
- nvhe : 基于 nVHE 的标准模式，不支持受保护的客户机。
- protected : 基于 nVHE 的模式，对主机保持私有状态的客户机。  
默认为基于硬件支持的 **VHE/nVHE**。

### nosmep=[X86,PPC64s]

使用这个参数，您可以禁用 SMEP (Supervisor Mode Execution Prevention)，即使处理器支持它。

格式：**pci=option[,option...] [PCI] various\_PCI\_subsystem\_options**

此处的某些选项针对特定的设备或一组设备(<pci\_dev>)而操作。它们使用以下格式之一指定：

```
[<domain>:]<bus>:<dev>.<func>[/<dev>.<func>]*
pci:<vendor>:<device>[:<subvendor>:<subdevice>]
```

#### 注意

- 第一个格式指定 PCI 总线/设备/函数地址，如果插入了新硬件，如果主板固件更改了，或者因为其他内核参数导致的更改，该地址可能会改变。如果未指定域，它将被视为零。（可选），在基础地址（这对于重新编号问题有弹性）后，可以指定通过多个设备和函数地址的设备的路径。
- 第二种格式使用配置空间中的 ID 选择设备，这可能与系统中的多个设备匹配。

- earlydump : 在内核更改任何内容之前转储 PCI 配置空间
- off: [X86] 不探测 PCI 总线
- bios: [X86-32] 强制使用 PCI BIOS，不直接访问硬件。如果您的机器有一个非标准 PCI 主机网桥，则使用此选项。
- nobios: [X86-32] 不允许使用 PCI BIOS，只允许直接硬件访问方法。如果您在引导时遇到崩溃，并且怀疑是由 BIOS 引起的，请使用此选项。
- conf1: [X86] Force use PCI Configuration Access Mechanism 1 (IO 端口 0xCF8 中的配置地址，IO 端口 0xCFC 中的数据，两者都是 32 位)。
- conf2: [X86] 强制使用 PCI 配置访问机制 2 (IO 端口 0xCF8 是函数的一个 8 位端口，IO 端口 0xCFA，也是 8 位，设置总线号)。然后，通过端口 0xC000-0xCFFF 访问配置空间。

- 有关配置访问机制的更多信息，请参阅 <http://wiki.osdev.org/PCI>。
- noaer: [PCIE] 如果启用了 PCIEAER 内核配置参数，则可以使用这个内核引导选项来禁用 PCIE 高级错误报告的使用。
- nodomains: [PCI] 禁用对多个 PCI root 域的支持(也就是 ACPI-speak 中的 PCI 段)。
- nommconf: [X86] 禁用对 PCI 配置使用 MMCONFIG
- check\_enable\_amd\_mmconf [X86]: 检查并启用对 AMD 系列 10h CPU 上 PCI 配置空间的正确配置的 MMIO 访问
- nomsi: [MSI] 如果启用了 **PCI\_MSI** 内核配置参数，则可以使用这个内核引导选项来禁用系统范围内 MSI 中断的使用。
- noioapicquirk: [APIC] 禁用所有引导中断怪癖。保持引导 IRQ 启用的安全选项。这不一定是必需的。
- ioapicreroute: [APIC] 将引导 IRQs 重新路由到无法禁用引导 IRQs 的桥接的主 IO-APIC。当系统屏蔽 IRQ 时，这修复了伪 IRQ 的源。
- noioapicreroute [APIC] 禁用临时解决方案，该方案使用连接到无法禁用引导 IRQs 的芯片的 IRQ 的等效的 IRQ。与 ioapicreroute 相反。
- biosirq: [X86-32] 使用 PCI BIOS 调用来获取中断路由表。众所周知，这些调用在一些机器上是有 bug 的，在使用时它们会挂起机器，但在其他计算机上，它是获取中断路由表的唯一方法。如果内核无法在主板上分配 IRQ 或发现辅助的 PCI 总线，请尝试这个选项。
- ROM : [X86]将地址空间分配给扩展 ROM。请小心使用，因为某些设备在 ROM 和其他资源之间共享地址解码器。
- norom: [X86] 不会将地址空间分配给没有 BIOS 分配地址范围的 ROM。
- nobar: [X86] 不会将地址空间分配给未由 BIOS 分配的 BAR。
- irqmask=0xMMMM: [X86] 设置允许自动分配给 PCI 设备的 IRQ 的位掩码。您可以以这种方式使内核排除 ISA 卡的 IRQ。
- pirqaddr=0xAAAAA: [X86] 指定 PIRQ 表的物理地址（通常由 BIOS 生成）如果它在 **F0000h-10000h** 范围之外。
- lastbus=N: [X86] 扫描到 bus #N 的所有总线：如果内核无法找到二级总线，并且您想要明确告知它们是什么，则这很有用。
- assign-busses: [X86]总是自己分配所有的 PCI 总线号，覆盖固件可能已经完成的任何事情。
- usepirqmask: [X86] 尊重存储在 BIOS \$PIR 表中的可能的 IRQ 掩码。这在某些带有有问题的 BIOS 的系统上需要的，特别是一些 HP Pavilion N5400 和 Omnibook XE3 笔记本。如果启用了 ACPI IRQ 路由，则这将无效。
- noacpi: [X86]对 IRQ 路由或 PCI 扫描不使用 ACPI。
- use\_crs: [X86] 使用 ACPI 中的 PCI 主机网桥窗口信息。在 2008 或更高版本的 BIOS 上，这默认启用。如果您需要使用它，请报告一个 bug。
- nocrs: [X86]忽略来自 ACPI 的 PCI 主机网桥窗口。如果您需要使用它，请报告一个 bug。



- use\_e820: [X86] 使用 E820 保留来排除 PCI 主机网桥窗口的部分。这是主机网桥 \_CRS 方法中 BIOS 缺陷的一个临时解决方案。如果您需要使用它，请向 [linux-pci@vger.kernel.org](mailto:linux-pci@vger.kernel.org) 报告一个 bug。
- no\_e820: [X86] 忽略 PCI 主机网桥窗口的 E820 保留。这是现代硬件上的默认设置。如果您需要使用它，请向 [linux-pci@vger.kernel.org](mailto:linux-pci@vger.kernel.org) 报告一个 bug。
- routeirq : 为所有 PCI 设备执行 IRQ 路由。这通常在 `pci_enable_device ()` 中完成，因此此选项对于不调用它的有问题的驱动程序是一个临时解决方案。
- skip\_isa\_align: [X86] 不对齐 io 起始地址，因此可以处理更多 pci 卡
- 注意 : [X86] 不做任何早期类型 1 扫描。当读取某些设备的配置空间时，这可能会帮助机器检查某些坏掉的板子。但是，各种临时解决方案被禁用，一些 IOMMU 驱动程序将无法正常工作。
- bfsort : 将 PCI 设备按广度优先顺序排序。完成这个排序以获得与旧(= 2.4)内核兼容的设备顺序。
- nobfsort : 不要将 PCI 设备按广度优先顺序排序。
- pcie\_bus\_tune\_off : 禁用 PCIe MPS (Max Payload Size) 调整，并使用 BIOS 配置的 MPS 默认值。
- pcie\_bus\_safe : 将每个设备的 MPS 设置为根复合体以下所有设备支持的最大值。
- pcie\_bus\_perf : 将设备 MPS 设置为基于其父总线的最大允许的 MPS。另外，将 MRRS (最大读请求大小) 设置为最大支持的值 (不大于设备或总线支持的 MPS) 以获得最佳性能。
- pcie\_bus\_peer2peer : 将每个设备的 MPS 设置为 128B，保证每个设备都支持。此配置允许在任意对设备间点对点的 DMA，可能会以降低性能为代价。这也保证热添加的设备可以正常工作。
- cbiosize=nn[KMG] : 为 CardBus 网桥的 IO 窗口保留的固定总线空间的固定量。默认值为 256 字节。
- cbmemsize=nn[KMG] : 为 CardBus 网桥的内存窗口保留的总线空间的固定量。默认值为 64MB。
- resource\_alignment=
  - 格式 : [`<order of align>@<pci_dev>`]; ...]
  - 指定对齐和设备，以重新分配对应的内存资源。上面描述了如何指定设备。如果没有指定 `<order of align>`，则 `PAGE_SIZE` 将用作对齐。如果需要扩展资源窗口，则可以指定 PCI-PCI 网桥。要指定设备的多个实例的对齐，可以指定 PCI 供应商、设备、子供应商和子设备，例如 `12@pci:8086:9c22:103c:198f` 用于 4096 字节对齐。
- ecrc= : 启用/禁用 PCIe ECRC (事务层端到端 CRC 检查)。
  - bios : 使用 BIOS/固件设置。这是默认值。
  - off : 关闭 ECRC
  - on: 打开 ECRC。
- hpiosize=nn[KMG] : 为热插拔网桥的 IO 窗口保留的总线空间的固定量。默认大小为 256 字节。

- `hpmmiosize=nn[KMG]`：为热插拔网桥的 MMIO 窗口保留的总线空间的固定量。默认大小为 2 MB。
- `hpmmioprefsize=nn[KMG]`：为热插拔网桥的 MMIO\_PREF 窗口保留的总线空间的固定量。默认大小为 2 MB。
- `hpmemsize=nn[KMG]`：为热插拔网桥的 MMIO 和 MMIO\_PREF 窗口保留的总线空间的固定量。默认大小为 2 MB。
- `hpbussize=nn`：为热插拔网桥下总线保留的额外总线号的最小数。默认值为 1。
- `realloc=`：如果 BIOS 所做的分配太小，无法容纳所有子设备所需的资源，则启用/禁用重新分配 PCI 网桥资源。
  - `off`：关掉 `realloc`
  - `on`：打开 `realloc`
- `realloc`：与 `realloc=on` 相同
- `noari`：不要使用 PCIe ARI。
- `noats`：[PCIe, Intel-IOMMU, AMD-IOMMU] 不使用 PCIe ATS（和 IOMMU 设备 IOTLB）。
- `pcie_scan_all`：扫描所有可能的 PCIe 设备。否则，我们只查找 PCIe 下游端口下的一个设备。
- `big_root_window`：尝试向 AMD CPU 上的 PCIe 根联合体添加一个大型 64 位内存窗口。有些 GFX 硬件可以调整 BAR 的大小，以允许访问所有 VRAM。添加窗口有一点风险（可能与未报告的设备冲突），因此这会影响内核。
- `disable_acs_redir=<pci_dev>[; ...]`：指定一个或多个以分号分割的 PCI 设备（按上面指定的格式）。指定的每个设备都将强制关闭 PCI ACS 重定向功能，它允许通过网桥在设备之间进行 P2P 通信，而无需强制上游。注：这会删除设备间的隔离，并可能会将更多设备放在 IOMMU 组中。
- `force_floating`：[S390] 强制使用浮动中断。
- `nomio`：[S390] 不使用 MIO 指令。
- `norid`：[S390] 忽略 RID 字段，并强制每个 PCI 功能使用一个 PCI 域

#### `rcupdate.rcu_cpu_stall_timeout=[KNL]`

为 RCU CPU stall 失速警告信息设置超时。该值以秒为单位，允许的最大值为 300 秒。

#### `rcupdate.rcu_task_stall_timeout=[KNL]`

使用这个参数，您可以为 RCU 任务失速警告信息设置超时（以 jiffies 为单位）。使用值小于或等于零的值禁用。

默认值为 10 分钟。

值的更改不会生效，直到下一个宽限期开始。

#### `retbleed=[X86]`

有了这个参数，您可以控制 RETbleed（使用返回说明）漏洞的缓解（具有 Return 指令的任意规格代码执行）。

仅使用基于 AMD 的 UNRET 和 IBPB 缓解不会同级线程影响其他同级线程的预测。因此，STIBP 用于支持它的处理器，并缓解不支持它的处理器上的 SMT。

- off - 不缓解
  - auto - 自动选择一个缓解
  - auto,nosmt - 自动选择一个缓解，如果需要整个缓解则禁用 SMT（只在 Zen1 和没有 STIBP 的旧 Zen1 上）。
  - IBPB - 在 AMD 上，也缓解基本块边界上的短投机窗口。安全、最高的 perf 影响。它还启用 STIBP（如果存在的话）。不适用于 Intel。
  - IBPB,nosmt - 上面的 **ibpb** 类似，但将在 STIBP 不可用时禁用 SMT。这是没有 STIBP 的系统的替代选择。
  - unret - 强制启用未经训练的返回 thanks，仅对基于 AMD f15h-f17h 的系统有效。
  - unret,nosmt - 与 unret 类似，但将在 STIBP 不可用时禁用 SMT。这是没有 STIBP 的系统的替代选择。
- 根据 CPU，选择 **auto** 将在运行时选择缓解方法。

不指定这个选项等同于 **retbleed=auto**。

**swiotlb=[ARM,IA-64,PPC,MIPS,X86]**

格式：{ <int> [,<int>] | **force** | **noforce** }

- <int> - I/O TLB slabs 数
- <int> - 逗号后的第二个整数。自身带有锁的 **swiotlb** 区域号。将向上舍入为 2 的幂。
- **force** - 强制使用 bounce 缓冲区，即使它们没有被内核自动使用
- **noforce** - 永不使用 bounce 缓冲（用于调试）

## 新的 sysctl 参数

**kernel.nmi\_wd\_lpm\_factor** (只适用于 PPC)

这个因素代表了在 LPM 过程中计算 NMI watchdog 超时时添加到 **watchdog\_thresh** 的百分比。软锁定超时不会受到影响。使用此因素应用到 NMI watchdog 超时（仅在 **nmi\_watchdog** 设为 1 时）。

- 值 **0** 表示没有更改。
- 默认值为 **200**，这意味着 NMI watchdog 被设为 30s（根据 **watchdog\_thresh** 等于 10）。

**net.core.txrehash**

使用这个参数，当 **SO\_TXREHASH** 选项设为 **SOCK\_TXREHASH\_DEFAULT** 时，您可以控制侦听套接字的默认哈希 rethink 行为（这没有被 **setsockopt** 覆盖）。

- 如果设为 **1** (默认)，则在侦听套接字上执行哈希 rethink。
- 如果设为 **0**，则不会执行哈希 rethink。

**net.sctp.reconf\_enable** - BOOLEAN

借助此扩展，您可以启用或禁用 RFC6525 中指定的流重新配置功能的扩展。此扩展提供了"reset"流的功能，并包括 **Outgoing/Incoming SSN Reset**、**SSN/TSN Reset** 和 **Add Outgoing Streams** 的参数。

- 1：启用扩展。
- 0：禁用扩展。
- 默认为 0。

#### net.sctp.intl\_enable - BOOLEAN

有了这个扩展，您可以启用或禁用 RFC8260 中指定的用户消息交错功能的扩展。此扩展允许在不同流上发送的用户消息的交错。启用此功能后，I-DATA 块将替换 DATA 块，以便在对等点也支持时携带用户消息。请注意，要使用这个功能，必须将这个选项设为 1，并将套接字选项 **SCTP\_FRAGMENT\_INTERLEAVE** 设为 2，将 **SCTP\_INTERLEAVING\_SUPPORTED** 设为 1。

- 1：启用扩展。
- 0：禁用扩展。
- 默认为 0。

#### net.sctp.ecn\_enable - BOOLEAN

借助此扩展，您可以通过 SCTP 控制显式协调通知(ECN)的使用。与 TCP 一样，只有在 SCTP 连接的两端表示支持它时，才会使用 ECN。该功能在允许支持路由器在必须丢弃数据包之前发出拥塞信号，来避免由于拥塞而造成的损失方面很有用。

- 1：启用 ecn。
- 0：禁用 ecn。
- 默认为 1。

#### vm.hugetlb\_optimize\_vmemmap

当配置 **memory\_hotplug.memmap\_on\_memory** 内核参数或 *struct* 页( `include/linux/mm_types.h` 中定义的一个结构)的大小不是 2 的幂（不寻常的系统配置可能会导致这种情况）时，这个 knob 不可用。

您可以启用（设为 1）或禁用（设为 0）优化与每个 HugeTLB 页关联的 **vmemmap** 页的功能。

- 如果启用了，则后续从 buddy 分配器分配的 HugeTLB 页的 **vmemmap** 页会被优化（每个 2MB HugeTLB 页 7 个页，每个 1GB HugeTLB 页 4095 个页），而已经分配的 HugeTLB 页面不会被优化。当这些优化的 HugeTLB 页面从 HugeTLB 池中释放到伙伴分配器时，表示那个范围的 **vmemmap** 页需要再次重新映射，之前丢弃的 **vmemmap** 页需要再次重新定位。
- 如果您的用例是 HugeTLB 页面被分配 imbangu（例如，不要使用 **nr\_hugepages** 明确分配 HugeTLB 页面，但只设置 **nr\_overcommit\_hugepages**，这些过度承诺的 HugeTLB 页面会被分配 `impromptu`）而不是从 HugeTLB 池中拉取，您应该权衡节省内存的好处和在 HugeTLB 池和伙伴分配器之间分配或释放 HugeTLB 页的更多开销(比以前慢了约 2 倍)。请注意的另一个行为是，如果系统面临内存压力很大，则可能会阻止用户将 HugeTLB 页从 HugeTLB 池释放到伙伴分配器，因为 **vmemmap** 页的分配可能会失败，如果系统遇到这种情况，您不得不稍后重试。
- 如果禁用了，则后续来自伙伴分配器的 HugeTLB 页分配的 **vmemmap** 页不会被优化，这意味着来自 buddy 分配器分配时间的额外开销会消失，而已优化的 HugeTLB 页不受影响。如果要确保没有优化的 HugeTLB 页，您可以首先将 **nr\_hugepages** 设为 0，然后禁用它。请注

意，将 **0** 写到 **nr\_hugepages** 将使任何 在使用的 HugeTLB 页变为多余页。因此，这些多余页仍然会被优化，直到它们不再使用。在系统中没有优化的页之前，您需要等待这些多余页面被释放。

### net.core.rps\_default\_mask

新创建的网络设备上使用的默认 RPS CPU 掩码。空掩码表示默认禁用 RPS。

## 更改了 sysctl 参数

### kernel.numa\_balancing

使用这个参数，您可以启用、禁用和配置基于 NUMA 内存平衡的自动页错误。内存自动移到经常访问它的节点。设置的值可以是 ORing 的结果：

```

=====
0 NUMA_BALANCING_DISABLED
1 NUMA_BALANCING_NORMAL
2 NUMA_BALANCING_MEMORY_TIERING
=====

```

或 **NUMA\_BALANCING\_**，以在不同 NUMA 节点间优化页放置，以减少远程访问。在 NUMA 机器上，如果远程内存被 CPU 访问，则存在性能损失。当启用此功能时，内核通过定期取消映射页并稍后捕获页错误来采样哪个任务线程正在访问内存。在页错误时，它会确定正在被访问的数据是否应迁移到本地内存节点。

或 **NUMA\_BALANCING\_MEMORY\_TIERING**，以优化不同类型的内存（代表不同的 NUMA 节点）之间的页放置，以便将热页放在快速内存中。这也是根据未映射和页错误来实现的。

### net.ipv6.route.max\_size

现在，这已被 ipv6 弃用，因为垃圾回收管理缓存的路由条目。

### net.sctp.sctp\_wmem

之前，这个调整已被记录为没有任何效果。现在，只使用第一个值(**min**)，**默认值** 和 **max** 被忽略。

- **min**：可被 SCTP 套接字使用的最小发送缓冲区大小。即使面临内存压力，也保证每个 SCTP 套接字（但不关联）。
- 默认值为 **4K**。

## 第 6 章 设备驱动程序

### 6.1. 新驱动程序

- ACPI 视频驱动程序 (**video**)，仅适用于 64 位 ARM 架构
- CXL 内存端点设备的 CXL 驱动程序，以及内存扩展的切换(**cxl\_mem**)
- GNSS 接收器核(**gnss**)
- GPIO 模拟器模块(**gpio-sim**)，仅适用于 64 位 ARM 架构
- VirtIO GPIO 驱动程序(**gpio-virtio**)，仅适用于 64 位 ARM 架构
- NVIDIA Tegra HTE (Hardware Timestamping Engine) 驱动程序(**hte-tegra194**)，仅适用于 64 位 ARM 架构
- LPI2C 总线(**i2c-imx-lpi2c**)的 I2C 适配器驱动程序，仅适用于 64 位 ARM 架构
- VirtIO i2c 总线驱动程序(**i2c-virtio**)，仅适用于 64 位 ARM 架构
- 输入子系统的用户级驱动程序支持(**uinput**)，仅适用于 64 位 ARM 架构
- 模块实现可供 nvme 主机或目标驱动程序使用的通用功能(**nvme-common**)
- AMD PMC 驱动程序(**amd-pmc**)，仅适用于 AMD 和 Intel 64 位构架
- NVIDIA sn2201 平台驱动程序(**nvsw-sn2201**)，仅适用于 AMD 和 Intel 64 位构架
- 串行多实例化伪设备驱动程序(**serial-multi-instantiate**)，仅适用于 AMD 和 Intel 64 位构架
- Micro Crystal RV8803 RTC 驱动程序(**rtc-rv8803**)，仅适用于 64 位 ARM 架构、AMD 和 Intel 64 位构架
- NVIDIA Tegra QSPI 控制器驱动程序(**spi-tegra210-quad**)，仅适用于 64 位 ARM 架构
- Cypress CCGx Type-C 控制器的 UCSI 驱动程序(**ucsi\_ccg**)，仅适用于 64 位 ARM 架构
- 机密计算 EFI secret 区域访问(**efi\_secret**)，仅适用于 AMD 和 Intel 64 位构架
- TDX 客户机驱动程序(**tdx-guest**)，仅适用于 AMD 和 Intel 64 位构架
- HPE watchdog 驱动程序(**hpwdt**)，仅适用于 64 位 ARM 架构
- POWER Architecture Platform Watchdog 驱动程序(**pseries-wdt**)，仅适用于 IBM Power Systems Little Endian

### 网络驱动程序

- VXLAN 封装流量的驱动程序(**vxlan**)
- Marvell OcteonTX2 RVU Admin Function 驱动程序 (**rvu\_af**)，仅适用于 64 位 ARM 架构
- Marvell RVU NIC physical Function 驱动程序 (**rvu\_nicpf**)，仅适用于 64 位 ARM 架构
- Marvell RVU NIC PTP 驱动程序 (**otx2\_ptp**)，仅适用于 64 位 ARM 架构

- Marvell RVU NIC Virtual Function 驱动程序 (**rvu\_nicvf**), 仅适用于 64 位 ARM 架构
- NVIDIA Tegra MGBE 驱动程序(**dwmac-tegra**), 仅适用于 64 位 ARM 架构
- 串行线 CAN 接口(**slcan**), 仅适用于 64 位 ARM 架构
- Solarflare Siena 网络驱动程序(**sfc-siena**), 仅适用于 IBM Power Systems、Little Endian 和 AMD 和 Intel 64 位构架

## 图形驱动程序和各种驱动程序

- DRM Buddy Allocator (**drm\_buddy**), 仅适用于 64 位 ARM 架构和 IBM Power Systems Little Endian
- DRM 显示适配器帮助程序(**drm\_display\_helper**), 仅适用于 64 位 ARM 架构、IBM Power Systems、Little Endian 和 AMD 和 Intel 64 位构架
- DRM DisplayPort AUX 总线(**drm\_dp\_aux\_bus**), 仅适用于 64 位 ARM 架构
- Host1x driver for Tegra 产品(**host1x**), 仅适用于 64 位 ARM 架构
- NVIDIA Tegra DRM 驱动程序(**tegra-drm**), 仅适用于 64 位 ARM 架构
- Intel® GVT-g for KVM (**kvmgt**), 仅适用于 AMD 和 Intel 64 位构架
- HP® iLO/iLO2 管理处理器(**hpilo**), 仅适用于 64 位 ARM 架构
- GSC 设备的 Intel® 辅助驱动程序 (**mei-gsc**), 仅适用于 AMD 和 Intel 64 位构架

## 6.2. 更新的驱动程序

### 存储驱动程序更新

- Microchip Smart Family Controller 的驱动程序(**smartpqi**)已更新至版本 2.1.20-035 (仅适用于 64 位 ARM 架构、IBM Power Systems、Little Endian 和 AMD 和 Intel 64 位构架)。
- Emulex LightPulse Fibre Channel SCSI 驱动程序(**lpfc**)已更新至版本 14.2.0.8 (仅适用于 64 位 ARM 架构、IBM Power Systems、Little Endian 和 AMD 和 Intel 64 位构架)。
- MPI3 Storage Controller Device 驱动程序(**mpi3mr**)已更新至版本 8.2.0.3.0。
- CSI debug adapter 驱动程序 (**scsi\_debug**)已更新至版本 0191。
- LSI MPT Fusion SAS 3.0 Device 驱动程序 (**mpt3sas**)已更新至版本 43.100.00.00 (仅适用于 64 位 ARM 架构、IBM Power Systems、Little Endian 和 AMD 和 Intel 64 位构架)。

## 第 7 章 可用的 BPF 功能

本章提供了这个 Red Hat Enterprise Linux 9 次版本的 kernel 中 **Berkeley Packet Filter (BPF)** 功能的完整列表。表包括：

- [系统配置和其他选项](#)
- [可用的程序类型和支持的帮助程序](#)
- [可用的映射类型](#)

本章包含 **bpf tool feature** 命令自动生成的输出。

表 7.1. 系统配置和其他选项

| 选项                            | 值                              |
|-------------------------------|--------------------------------|
| unprivileged_bpf_disabled     | 2 (bpf() 系统调用限制为特权用户，管理员可以改变。) |
| JIT 编译器                       | 1 (启用)                         |
| JIT 编译器强化                     | 1 (为非特权用户启用)                   |
| JIT 编译器 kallsyms 导出           | 1 (为 root 用户启用)                |
| 非特权用户的 JIT 的内存限制              | 264241152                      |
| CONFIG_BPF                    | y                              |
| CONFIG_BPF_SYSCALL            | y                              |
| CONFIG_HAVE_EBPF_JIT          | y                              |
| CONFIG_BPF_JIT                | y                              |
| CONFIG_BPF_JIT_ALWAYS_ON      | y                              |
| CONFIG_DEBUG_INFO_BTF         | y                              |
| CONFIG_DEBUG_INFO_BTF_MODULES | y                              |
| CONFIG_CGROUPS                | y                              |
| CONFIG_CGROUP_BPF             | y                              |
| CONFIG_CGROUP_NET_CLASSID     | y                              |
| CONFIG_SOCK_CGROUP_DATA       | y                              |



| 选项                              | 值 |
|---------------------------------|---|
| CONFIG_BPF_EVENTS               | y |
| CONFIG_KPROBE_EVENTS            | y |
| CONFIG_UPROBE_EVENTS            | y |
| CONFIG_TRACING                  | y |
| CONFIG_FTRACE_SYSCALLS          | y |
| CONFIG_FUNCTION_ERROR_INJECTION | y |
| CONFIG_BPF_KPROBE_OVERRIDE      | n |
| CONFIG_NET                      | y |
| CONFIG_XDP_SOCKETS              | y |
| CONFIG_LWTUNNEL_BPF             | y |
| CONFIG_NET_ACT_BPF              | m |
| CONFIG_NET_CLS_BPF              | m |
| CONFIG_NET_CLS_ACT              | y |
| CONFIG_NET_SCH_INGRESS          | m |
| CONFIG_XFRM                     | y |
| CONFIG_IP_ROUTE_CLASSID         | y |
| CONFIG_IPV6_SEG6_BPF            | n |
| CONFIG_BPF_LIRC_MODE2           | n |
| CONFIG_BPF_STREAM_PARSER        | y |
| CONFIG_NETFILTER_XT_MATCH_BPF   | m |
| CONFIG_BPFILTER                 | n |
| CONFIG_BPFILTER_UMH             | n |

| 选项              | 值    |
|-----------------|------|
| CONFIG_TEST_BPF | m    |
| CONFIG_HZ       | 1000 |
| bpf() syscall   | 可用   |
| 大型程序大小限制        | 可用   |
| 绑定的循环支持         | 可用   |
| ISA 扩展 v2       | 可用   |
| ISA 扩展 v3       | 可用   |

表 7.2. 可用的程序类型和支持的帮助程序

| 程序类型          | 可用的帮助程序  |
|---------------|--|
| socket_filter | bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem,<br>bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call,<br>bpf_perf_event_output, bpf_skb_load_bytes, bpf_get_current_task,<br>bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_get_socket_uid,<br>bpf_skb_load_bytes_relative, bpf_map_push_elem, bpf_map_pop_elem,<br>bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_probe_read_user,<br>bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str,<br>bpf_jiffies64, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve,<br>bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock,<br>bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock,<br>bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr,<br>bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem,<br>bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel,<br>bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg,<br>bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem,<br>bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr,<br>bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data |

| 程序类型      | 可用的帮助程序  |
|-----------|--|
| kprobe    | bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task, bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str, bpf_perf_event_read_value, bpf_get_stack, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_send_signal, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_send_signal_thread, bpf_jiffies64, bpf_get_ns_current_pid_tgid, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_get_task_stack, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_get_attach_cookie, bpf_task_pt_regs, bpf_get_branch_snapshot, bpf_find_vma, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data  |
| sched_cls | bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_skb_store_bytes, bpf_l3_csum_replace, bpf_l4_csum_replace, bpf_tail_call, bpf_clone_redirect, bpf_get_cgroup_classid, bpf_skb_vlan_push, bpf_skb_vlan_pop, bpf_skb_get_tunnel_key, bpf_skb_set_tunnel_key, bpf_redirect, bpf_get_route_realm, bpf_perf_event_output, bpf_skb_load_bytes, bpf_csum_diff, bpf_skb_get_tunnel_opt, bpf_skb_set_tunnel_opt, bpf_skb_change_proto, bpf_skb_change_type, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_skb_change_tail, bpf_skb_pull_data, bpf_csum_update, bpf_set_hash_invalid, bpf_get_numa_node_id, bpf_skb_change_head, bpf_get_socket_cookie, bpf_get_socket_uid, bpf_set_hash, bpf_skb_adjust_room, bpf_skb_get_xfrm_state, bpf_skb_load_bytes_relative, bpf_fib_lookup, bpf_skb_cgroup_id, bpf_skb_ancestor_cgroup_id, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_sk_fullsock, bpf_tcp_sock, bpf_skb_ecn_set_ce, bpf_get_listener_sock, bpf_skc_lookup_tcp, bpf_tcp_check_syncookie, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_tcp_gen_syncookie, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_sk_assign, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_csum_level, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_skb_cgroup_classid, bpf_redirect_neigh, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_redirect_peer, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_check_mtu, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_skb_set_tstamp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_tcp_raw_gen_syncookie_ipv4, bpf_tcp_raw_gen_syncookie_ipv6, bpf_tcp_raw_check_syncookie_ipv4, bpf_tcp_raw_check_syncookie_ipv6 |

| 程序类型       | 可用的帮助程序  |
|------------|--|
| sched_act  | bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_skb_store_bytes, bpf_l3_csum_replace, bpf_l4_csum_replace, bpf_tail_call, bpf_clone_redirect, bpf_get_cgroup_classid, bpf_skb_vlan_push, bpf_skb_vlan_pop, bpf_skb_get_tunnel_key, bpf_skb_set_tunnel_key, bpf_redirect, bpf_get_route_realm, bpf_perf_event_output, bpf_skb_load_bytes, bpf_csum_diff, bpf_skb_get_tunnel_opt, bpf_skb_set_tunnel_opt, bpf_skb_change_proto, bpf_skb_change_type, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_skb_change_tail, bpf_skb_pull_data, bpf_csum_update, bpf_set_hash_invalid, bpf_get_numa_node_id, bpf_skb_change_head, bpf_get_socket_cookie, bpf_get_socket_uid, bpf_set_hash, bpf_skb_adjust_room, bpf_skb_get_xfrm_state, bpf_skb_load_bytes_relative, bpf_fib_lookup, bpf_skb_cgroup_id, bpf_skb_ancestor_cgroup_id, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_sk_fullsock, bpf_tcp_sock, bpf_skb_ecn_set_ce, bpf_get_listener_sock, bpf_skc_lookup_tcp, bpf_tcp_check_syncookie, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_tcp_gen_syncookie, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_sk_assign, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_csum_level, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_skb_cgroup_classid, bpf_redirect_neigh, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_redirect_peer, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_check_mtu, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_skb_set_tstamp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_tcp_raw_gen_syncookie_ipv4, bpf_tcp_raw_gen_syncookie_ipv6, bpf_tcp_raw_check_syncookie_ipv4, bpf_tcp_raw_check_syncookie_ipv6 |
| tracepoint | bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task, bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str, bpf_perf_event_read_value, bpf_get_stack, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_send_signal, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_send_signal_thread, bpf_jiffies64, bpf_get_ns_current_pid_tgid, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_get_task_stack, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_get_attach_cookie, bpf_task_pt_regs, bpf_get_branch_snapshot, bpf_find_vma, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data  |

| 程序类型       | 可用的帮助程序   |
|------------|---|
| xdp        | bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem,<br>bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call,<br>bpf_redirect, bpf_perf_event_output, bpf_csum_diff, bpf_get_current_task,<br>bpf_get_numa_node_id, bpf_xdp_adjust_head, bpf_redirect_map, bpf_xdp_adjust_meta,<br>bpf_xdp_adjust_tail, bpf_fib_lookup, bpf_sk_lookup_tcp, bpf_sk_lookup_udp,<br>bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem,<br>bpf_spin_lock, bpf_spin_unlock, bpf_skc_lookup_tcp, bpf_tcp_check_syncookie,<br>bpf_tcp_gen_syncookie, bpf_probe_read_user, bpf_probe_read_kernel,<br>bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64,<br>bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit,<br>bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock,<br>bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock,<br>bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf,<br>bpf_ktime_get_coarse_ns, bpf_check_mtu, bpf_for_each_map_elem, bpf_snprintf,<br>bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel,<br>bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_xdp_get_buff_len,<br>bpf_xdp_load_bytes, bpf_xdp_store_bytes, bpf_kptr_xchg,<br>bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem,<br>bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr,<br>bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_tcp_raw_gen_syncookie_ipv4,<br>bpf_tcp_raw_gen_syncookie_ipv6, bpf_tcp_raw_check_syncookie_ipv4,<br>bpf_tcp_raw_check_syncookie_ipv6 |
| perf_event | bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read,<br>bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call,<br>bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm,<br>bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task,<br>bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str,<br>bpf_perf_event_read_value, bpf_perf_prog_read_value, bpf_get_stack,<br>bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem,<br>bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_send_signal,<br>bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str,<br>bpf_probe_read_kernel_str, bpf_send_signal_thread, bpf_jiffies64,<br>bpf_read_branch_records, bpf_get_ns_current_pid_tgid,<br>bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output,<br>bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query,<br>bpf_get_task_stack, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr,<br>bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf,<br>bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback,<br>bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_get_attach_cookie,<br>bpf_task_pt_regs, bpf_get_branch_snapshot, bpf_find_vma, bpf_loop, bpf_strncmp,<br>bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem,<br>bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr,<br>bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data  |

| 程序类型        | 可用的帮助程序   |
|-------------|---|
| cgroup_skb  | bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem,<br>bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call,<br>bpf_perf_event_output, bpf_skb_load_bytes, bpf_get_current_task,<br>bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_get_socket_uid,<br>bpf_skb_load_bytes_relative, bpf_skb_cgroup_id, bpf_get_local_storage,<br>bpf_skb_ancestor_cgroup_id, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release,<br>bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock,<br>bpf_spin_unlock, bpf_sk_fullsock, bpf_tcp_sock, bpf_skb_ecn_set_ce,<br>bpf_get_listener_sock, bpf_skc_lookup_tcp, bpf_sk_storage_get,<br>bpf_sk_storage_delete, bpf_probe_read_user, bpf_probe_read_kernel,<br>bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64,<br>bpf_ktime_get_boot_ns, bpf_sk_cgroup_id, bpf_sk_ancestor_cgroup_id,<br>bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard,<br>bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock,<br>bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock,<br>bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf,<br>bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init,<br>bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs,<br>bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg,<br>bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem,<br>bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr,<br>bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data |
| cgroup_sock | bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem,<br>bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call,<br>bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm,<br>bpf_get_cgroup_classid, bpf_perf_event_output, bpf_get_current_task,<br>bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_get_current_cgroup_id,<br>bpf_get_local_storage, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem,<br>bpf_spin_lock, bpf_spin_unlock, bpf_sk_storage_get, bpf_probe_read_user,<br>bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str,<br>bpf_jiffies64, bpf_get_netns_cookie, bpf_get_current_ancestor_cgroup_id,<br>bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit,<br>bpf_ringbuf_discard, bpf_ringbuf_query, bpf_snprintf_btf, bpf_per_cpu_ptr,<br>bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns,<br>bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback,<br>bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_loop, bpf_strncmp,<br>bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem,<br>bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr,<br>bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data   |

| 程序类型    | 可用的帮助程序   |
|---------|---|
| lwt_in  | bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem,<br>bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call,<br>bpf_get_cgroup_classid, bpf_get_route_realm, bpf_perf_event_output,<br>bpf_skb_load_bytes, bpf_csum_diff, bpf_skb_under_cgroup, bpf_get_hash_recalc,<br>bpf_get_current_task, bpf_skb_pull_data, bpf_get_numa_node_id, bpf_lwt_push_encap,<br>bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock,<br>bpf_spin_unlock, bpf_probe_read_user, bpf_probe_read_kernel,<br>bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64,<br>bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit,<br>bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock,<br>bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock,<br>bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf,<br>bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init,<br>bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs,<br>bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg,<br>bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem,<br>bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr,<br>bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data |
| lwt_out | bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem,<br>bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call,<br>bpf_get_cgroup_classid, bpf_get_route_realm, bpf_perf_event_output,<br>bpf_skb_load_bytes, bpf_csum_diff, bpf_skb_under_cgroup, bpf_get_hash_recalc,<br>bpf_get_current_task, bpf_skb_pull_data, bpf_get_numa_node_id, bpf_map_push_elem,<br>bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock,<br>bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str,<br>bpf_probe_read_kernel_str, bpf_jiffies64, bpf_ktime_get_boot_ns, bpf_ringbuf_output,<br>bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query,<br>bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock,<br>bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf,<br>bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf,<br>bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init,<br>bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs,<br>bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg,<br>bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem,<br>bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr,<br>bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data                     |

| 程序类型     | 可用的帮助程序   |
|----------|---|
| lwt_xmit | bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem,<br>bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id,<br>bpf_skb_store_bytes, bpf_l3_csum_replace, bpf_l4_csum_replace, bpf_tail_call,<br>bpf_clone_redirect, bpf_get_cgroup_classid, bpf_skb_get_tunnel_key,<br>bpf_skb_set_tunnel_key, bpf_redirect, bpf_get_route_realm, bpf_perf_event_output,<br>bpf_skb_load_bytes, bpf_csum_diff, bpf_skb_get_tunnel_opt, bpf_skb_set_tunnel_opt,<br>bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task,<br>bpf_skb_change_tail, bpf_skb_pull_data, bpf_csum_update, bpf_set_hash_invalid,<br>bpf_get_numa_node_id, bpf_skb_change_head, bpf_lwt_push_encap,<br>bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock,<br>bpf_spin_unlock, bpf_probe_read_user, bpf_probe_read_kernel,<br>bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64,<br>bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit,<br>bpf_ringbuf_discard, bpf_ringbuf_query, bpf_csum_level, bpf_skc_to_tcp6_sock,<br>bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock,<br>bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr,<br>bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem,<br>bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel,<br>bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg,<br>bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem,<br>bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr,<br>bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data |
| sock_ops | bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem,<br>bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call,<br>bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id,<br>bpf_get_socket_cookie, bpf_setsockopt, bpf_sock_map_update, bpf_getsockopt,<br>bpf_sock_ops_cb_flags_set, bpf_sock_hash_update, bpf_get_local_storage,<br>bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock,<br>bpf_spin_unlock, bpf_tcp_sock, bpf_sk_storage_get, bpf_sk_storage_delete,<br>bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str,<br>bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_netns_cookie,<br>bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit,<br>bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock,<br>bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock,<br>bpf_load_hdr_opt, bpf_store_hdr_opt, bpf_reserve_hdr_opt, bpf_snprintf_btf,<br>bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf,<br>bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init,<br>bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs,<br>bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg,<br>bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem,<br>bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr,<br>bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data  |



| 程序类型          | 可用的帮助程序  |
|---------------|--|
| sk_skb        | bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem,<br>bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id,<br>bpf_skb_store_bytes, bpf_tail_call, bpf_perf_event_output, bpf_skb_load_bytes,<br>bpf_get_current_task, bpf_skb_change_tail, bpf_skb_pull_data, bpf_get_numa_node_id,<br>bpf_skb_change_head, bpf_get_socket_cookie, bpf_get_socket_uid,<br>bpf_skb_adjust_room, bpf_sk_redirect_map, bpf_sk_redirect_hash, bpf_sk_lookup_tcp,<br>bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem,<br>bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_skc_lookup_tcp,<br>bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str,<br>bpf_probe_read_kernel_str, bpf_jiffies64, bpf_ktime_get_boot_ns, bpf_ringbuf_output,<br>bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query,<br>bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock,<br>bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf,<br>bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf,<br>bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init,<br>bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs,<br>bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg,<br>bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem,<br>bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr,<br>bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data |
| cgroup_device | bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem,<br>bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call,<br>bpf_get_current_uid_gid, bpf_perf_event_output, bpf_get_current_task,<br>bpf_get_numa_node_id, bpf_get_current_cgroup_id, bpf_get_local_storage,<br>bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock,<br>bpf_spin_unlock, bpf_probe_read_user, bpf_probe_read_kernel,<br>bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64,<br>bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit,<br>bpf_ringbuf_discard, bpf_ringbuf_query, bpf_snprintf_btf, bpf_per_cpu_ptr,<br>bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf,<br>bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel,<br>bpf_task_pt_regs, bpf_loop, bpf_strncmp, bpf_get_retval, bpf_set_retval, bpf_kptr_xchg,<br>bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr,<br>bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read,<br>bpf_dynptr_write, bpf_dynptr_data  |

| 程序类型           | 可用的帮助程序   |
|----------------|---|
| sk_msg         | bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem,<br>bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call,<br>bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_cgroup_classid,<br>bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id,<br>bpf_msg_redirect_map, bpf_msg_apply_bytes, bpf_msg_cork_bytes,<br>bpf_msg_pull_data, bpf_msg_redirect_hash, bpf_get_current_cgroup_id,<br>bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_msg_push_data,<br>bpf_msg_pop_data, bpf_spin_lock, bpf_spin_unlock, bpf_sk_storage_get,<br>bpf_sk_storage_delete, bpf_probe_read_user, bpf_probe_read_kernel,<br>bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64,<br>bpf_get_netns_cookie, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns,<br>bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard,<br>bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock,<br>bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock,<br>bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf,<br>bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init,<br>bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs,<br>bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg,<br>bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem,<br>bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr,<br>bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data |
| raw_tracepoint | bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read,<br>bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call,<br>bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm,<br>bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task,<br>bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str,<br>bpf_perf_event_read_value, bpf_get_stack, bpf_get_current_cgroup_id,<br>bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock,<br>bpf_spin_unlock, bpf_send_signal, bpf_probe_read_user, bpf_probe_read_kernel,<br>bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_send_signal_thread,<br>bpf_jiffies64, bpf_get_ns_current_pid_tgid, bpf_get_current_ancestor_cgroup_id,<br>bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit,<br>bpf_ringbuf_discard, bpf_ringbuf_query, bpf_get_task_stack, bpf_snprintf_btf,<br>bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete,<br>bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init,<br>bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip,<br>bpf_task_pt_regs, bpf_get_branch_snapshot, bpf_find_vma, bpf_loop, bpf_strncmp,<br>bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem,<br>bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr,<br>bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data  |

| 程序类型             | 可用的帮助程序   |
|------------------|---|
| cgroup_sock_addr | bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem,<br>bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call,<br>bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm,<br>bpf_get_cgroup_classid, bpf_perf_event_output, bpf_get_current_task,<br>bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_setsockopt, bpf_getsockopt,<br>bpf_bind, bpf_get_current_cgroup_id, bpf_get_local_storage, bpf_sk_lookup_tcp,<br>bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem,<br>bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_skc_lookup_tcp,<br>bpf_sk_storage_get, bpf_sk_storage_delete, bpf_probe_read_user,<br>bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str,<br>bpf_jiffies64, bpf_get_netns_cookie, bpf_get_current_ancestor_cgroup_id,<br>bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit,<br>bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock,<br>bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock,<br>bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf,<br>bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init,<br>bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs,<br>bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg,<br>bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem,<br>bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr,<br>bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data |
| lwt_seg6local    | bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem,<br>bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call,<br>bpf_get_cgroup_classid, bpf_get_route_realm, bpf_perf_event_output,<br>bpf_skb_load_bytes, bpf_csum_diff, bpf_skb_under_cgroup, bpf_get_hash_recalc,<br>bpf_get_current_task, bpf_skb_pull_data, bpf_get_numa_node_id, bpf_map_push_elem,<br>bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock,<br>bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str,<br>bpf_probe_read_kernel_str, bpf_jiffies64, bpf_ktime_get_boot_ns, bpf_ringbuf_output,<br>bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query,<br>bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock,<br>bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf,<br>bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf,<br>bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init,<br>bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs,<br>bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg,<br>bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem,<br>bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr,<br>bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data   |
| lirc_mode2       | 不支持   |

| 程序类型           | 可用的帮助程序  |
|----------------|--|
| sk_reuseport   | bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem,<br>bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call,<br>bpf_skb_load_bytes, bpf_get_current_task, bpf_get_numa_node_id,<br>bpf_get_socket_cookie, bpf_skb_load_bytes_relative, bpf_sk_select_reuseport,<br>bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock,<br>bpf_spin_unlock, bpf_probe_read_user, bpf_probe_read_kernel,<br>bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64,<br>bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit,<br>bpf_ringbuf_discard, bpf_ringbuf_query, bpf_snprintf_btf, bpf_per_cpu_ptr,<br>bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns,<br>bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback,<br>bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_loop, bpf_strncmp,<br>bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem,<br>bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr,<br>bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data  |
| flow_dissector | bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem,<br>bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call,<br>bpf_skb_load_bytes, bpf_get_current_task, bpf_get_numa_node_id,<br>bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock,<br>bpf_spin_unlock, bpf_probe_read_user, bpf_probe_read_kernel,<br>bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64,<br>bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit,<br>bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock,<br>bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock,<br>bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf,<br>bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init,<br>bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs,<br>bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg,<br>bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem,<br>bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr,<br>bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data   |
| cgroup_sysctl  | bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem,<br>bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call,<br>bpf_get_current_uid_gid, bpf_perf_event_output, bpf_get_current_task,<br>bpf_get_numa_node_id, bpf_get_current_cgroup_id, bpf_get_local_storage,<br>bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock,<br>bpf_spin_unlock, bpf_sysctl_get_name, bpf_sysctl_get_current_value,<br>bpf_sysctl_get_new_value, bpf_sysctl_set_new_value, bpf_strtol, bpf_strtoul,<br>bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str,<br>bpf_probe_read_kernel_str, bpf_jiffies64, bpf_ktime_get_boot_ns, bpf_ringbuf_output,<br>bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query,<br>bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf,<br>bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init,<br>bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_loop,<br>bpf_strncmp, bpf_get_retval, bpf_set_retval, bpf_kptr_xchg,<br>bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr,<br>bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read,<br>bpf_dynptr_write, bpf_dynptr_data |

| 程序类型                        | 可用的帮助程序  |
|-----------------------------|--|
| raw_tracepoint_wri<br>table | bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read,<br>bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call,<br>bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm,<br>bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task,<br>bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str,<br>bpf_perf_event_read_value, bpf_get_stack, bpf_get_current_cgroup_id,<br>bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock,<br>bpf_spin_unlock, bpf_send_signal, bpf_probe_read_user, bpf_probe_read_kernel,<br>bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_send_signal_thread,<br>bpf_jiffies64, bpf_get_ns_current_pid_tgid, bpf_get_current_ancestor_cgroup_id,<br>bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit,<br>bpf_ringbuf_discard, bpf_ringbuf_query, bpf_get_task_stack, bpf_snprintf_btf,<br>bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete,<br>bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init,<br>bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip,<br>bpf_task_pt_regs, bpf_get_branch_snapshot, bpf_find_vma, bpf_loop, bpf_strncmp,<br>bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem,<br>bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr,<br>bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data |
| cgroup_sockopt              | bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem,<br>bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call,<br>bpf_get_current_uid_gid, bpf_perf_event_output, bpf_get_current_task,<br>bpf_get_numa_node_id, bpf_get_current_cgroup_id, bpf_get_local_storage,<br>bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock,<br>bpf_spin_unlock, bpf_tcp_sock, bpf_sk_storage_get, bpf_sk_storage_delete,<br>bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str,<br>bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_netns_cookie,<br>bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit,<br>bpf_ringbuf_discard, bpf_ringbuf_query, bpf_snprintf_btf, bpf_per_cpu_ptr,<br>bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf,<br>bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel,<br>bpf_task_pt_regs, bpf_loop, bpf_strncmp, bpf_get_retval, bpf_set_retval, bpf_kptr_xchg,<br>bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr,<br>bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read,<br>bpf_dynptr_write, bpf_dynptr_data  |
| tracing                     | 不支持  |
| struct_ops                  | 不支持  |
| ext                         | 不支持  |
| lsm                         | 不支持  |

| 程序类型      | 可用的帮助程序  |
|-----------|--|
| sk_lookup | bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem,<br>bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call,<br>bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_sk_release,<br>bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock,<br>bpf_spin_unlock, bpf_probe_read_user, bpf_probe_read_kernel,<br>bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_sk_assign,<br>bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit,<br>bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock,<br>bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock,<br>bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf,<br>bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init,<br>bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs,<br>bpf_skc_to_unix_sock, bpf_loop, bpf_strcmp, bpf_kptr_xchg,<br>bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem,<br>bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr,<br>bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data  |
| syscall   | bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read,<br>bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call,<br>bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm,<br>bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task,<br>bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str,<br>bpf_get_socket_cookie, bpf_perf_event_read_value, bpf_get_stack,<br>bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem,<br>bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_sk_storage_get,<br>bpf_sk_storage_delete, bpf_send_signal, bpf_skb_output, bpf_probe_read_user,<br>bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str,<br>bpf_send_signal_thread, bpf_jiffies64, bpf_get_ns_current_pid_tgid, bpf_xdp_output,<br>bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output,<br>bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query,<br>bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock,<br>bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_get_task_stack, bpf_d_path,<br>bpf_copy_from_user, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr,<br>bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf,<br>bpf_sock_from_file, bpf_for_each_map_elem, bpf_snprintf, bpf_sys_bpf,<br>bpf_btf_find_by_name_kind, bpf_sys_close, bpf_timer_init, bpf_timer_set_callback,<br>bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_task_pt_regs,<br>bpf_get_branch_snapshot, bpf_skc_to_unix_sock, bpf_kallsyms_lookup_name,<br>bpf_find_vma, bpf_loop, bpf_strcmp, bpf_xdp_get_buff_len, bpf_copy_from_user_task,<br>bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock,<br>bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr,<br>bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data |

表 7.3. 可用的映射类型

| 映射类型 | 可用 |
|------|----|
| hash | 是  |
| 数组   | 是  |

| 映射类型                  | 可用 |
|-----------------------|----|
| prog_array            | 是  |
| perf_event_array      | 是  |
| percpu_hash           | 是  |
| percpu_array          | 是  |
| stack_trace           | 是  |
| cgroup_array          | 是  |
| lru_hash              | 是  |
| lru_percpu_hash       | 是  |
| lpm_trie              | 是  |
| array_of_maps         | 是  |
| hash_of_maps          | 是  |
| devmap                | 是  |
| sockmap               | 是  |
| cpumap                | 是  |
| xskmap                | 是  |
| sockhash              | 是  |
| cgroup_storage        | 是  |
| reuseport_sockarray   | 是  |
| percpu_cgroup_storage | 是  |
| queue                 | 是  |
| queue                 | 是  |
| sk_storage            | 是  |
| devmap_hash           | 是  |

| 映射类型          | 可用 |
|---------------|----|
| struct_ops    | 是  |
| ringbuf       | 是  |
| inode_storage | 是  |
| task_storage  | 是  |
| bloom_filter  | 是  |



## 第 8 章 程序错误修复

这部分论述了 Red Hat Enterprise Linux 9.2 中修复的对用户有严重影响的 bug。

### 8.1. 安装程序和镜像创建

#### 安装程序现在显示具有多路径或 DDF RAID 设备的自定义分区中正确的磁盘空间总量

在以前的版本中，当在具有多路径或 DDF RAID 设备的系统上的安装程序中选择了自定义分区时，不能正确报告总磁盘空间，成员磁盘设备被列为可用于分区。

有了这个更新，安装程序中的自定义分区报告了总磁盘空间的正确值，只允许将 DDF RAID 或多路径设备作为一个整体使用。

[Bugzilla:2052938](#)

#### 安装程序现在将配置选项正确地添加到 yum repo 文件中

在以前的版本中，安装程序在从附加安装存储库中包含和排除软件包时，不能正确地将配置选项添加到 yum repo 文件中。有了这个更新，yum repo 文件会被正确创建。因此，在 **repo** kickstart 命令中使用 **--excludepkgs=** 或 **--includepkgs=** 选项现在在安装过程中按预期排除或包括指定的软件包。

[Bugzilla:2158210](#)

#### 使用 filename DHCP 选项不再阻止下载 kickstart 文件进行安装

在以前的版本中，当构建从 NFS 服务器获取 kickstart 文件的路径时，安装程序不会考虑 **filename** DHCP 选项。因此，安装程序不下载 kickstart 文件，并会阻止安装过程。有了这个更新，**filename** DHCP 选项可以正确地构造到 kickstart 文件的路径。因此，Kickstart 文件会被正确下载，安装过程会正确启动。

[Bugzilla:1991843](#)

#### 安装程序现在在自定义分区过程中会创建一个新的 GPT 磁盘布局

在以前的版本中，当在内核命令行上指定 **inst.gpt** 时，安装程序不会将磁盘布局改为 GPT，用户在自定义分区 spoke 上删除具有 MBR 磁盘布局的磁盘中的所有分区。因此，MBR 磁盘布局保留在磁盘上。

有了这个更新，如果在内核命令行上指定了 **inst.gpt**，安装程序会在磁盘上创建一个新的 GPT 磁盘布局，且所有分区都会从自定义分区 spoke 上的磁盘中删除。

[Bugzilla:2127100](#)

#### 安装程序现在在自定义分区过程中列出所有 PPC PreP Boot 或 BIOS Boot 分区

在以前的版本中，当在自定义分区中添加多个 **PPC PreP Boot** 或 **BIOS Boot** 分区时，自定义分区屏幕只显示相关类型的一个分区。因此，自定义分区屏幕没有反映预期分区布局的实际状态，从而导致分区过程变得困难和不透明。

有了这个更新，自定义分区屏幕可以正确地显示分区表中的所有 **PPC PreP Boot** 或 **BIOS Boot** 分区。现在，用户可以更好地了解和管理预期的分区布局。

[Bugzilla:2093793](#)

#### Anaconda 现在验证 FIPS 要求的 LUKS 密码短语

在以前的版本中，Anaconda 不会检查 LUKS 密码短语的长度是否满足 FIPS 要求，底层工具会执行这个检查。因此，使用小于 8 个字符的密码短语在 FIPS 模式下安装会导致安装程序提前终止。

有了这个更新，安装程序已被改进来验证并强制实施密码短语的最小长度。因此，如果 LUKS 密码短语在 FIPS 模式下太短，安装程序会告知，并防止意外终止。

[Bugzilla:2163497](#)

## 8.2. 订阅管理

### 订阅管理器不再拒绝红帽内容的注册和获取

之前，由于 RHEL 9 中的容器检测逻辑有所改进，当在 OpenShift Container Platform (OCP) 下运行时，**subscription-manager** 在容器模式下进行操作。因此，该系统无法使用提供的订阅凭证，因此不会获取红帽内容。

这个更新修复了容器检测逻辑，因此，在 OCP 下运行的 **subscription-manager** 不会将系统（即正在运行的 pod）检测为容器。现在，您可以使用提供的订阅凭证，或使用您自己的凭证注册，来从 OpenShift 容器获取红帽内容。

[Bugzilla:2108549](#)

### subscription-manager 不再在终端中保留非必要的文本

从 RHEL 9.1 开始，**subscription-manager** 在处理任何操作时会显示进度信息。在以前的版本中，对于某些语言（通常为非拉丁语言），在操作完成后不会清除进度消息。有了这个更新，在操作完成后，所有信息都会被正确清除。

如果您之前禁用了进度信息，您可以输入以下命令重新启用它们：

```
# subscription-manager config --rhsm.progress_messages=1
```

[Bugzilla:2136694](#)

## 8.3. 软件管理

### 在涉及 **fapolicyd** 服务重启的事务过程中，RPM 不再挂起

在以前的版本中，如果您尝试更新导致 **fapolicyd** 服务重启的软件包，例如 **systemd**，RPM 事务会停止响应，因为 **fapolicyd** 插件无法与 **fapolicyd** 守护进程进行通信。

有了这个更新，**fapolicyd** 插件现在可以正确地与 **fapolicyd** 守护进程进行通信。因此，在涉及 **fapolicyd** 服务重启的事务过程中，RPM 不再挂起。

[Bugzilla:2111251](#)

### 现在，可以为软件包组或环境还原 DNF 升级事务

在以前的版本中，当尝试为软件包组或环境还原升级事务时，**dnf history rollback** 命令会失败。

有了这个更新，这个问题已被解决，您现在可以为软件包组或环境还原 DNF 升级事务。

[Bugzilla:2122626](#)

### 现在，对于通过升级更改其架构的软件包，可以进行安全 DNF 升级

与 [RHBA-2022:8295](#) 一起引入的 [BZ#2108969](#) 引入的补丁会导致回归，其中 DNF 升级会使用安全过滤器跳过的软件包，这些软件包会通过升级将架构改为 **noarch**。因此，缺少这些软件包的安全升级可能会使系统处于易受攻击的状态。

有了这个更新，这个问题已解决，安全 DNF 升级不再跳过将架构更改为 **noarch** 的软件包。

[Bugzilla:2124480](#)

现在，当 RPM 软件包被构建或重新构建时，会打包带有 3 个字母名称的 qt 消息 QM 文件

在以前的版本中，**find-lang.sh** 脚本无法找到名称由 3 个字符组成的 Qt 消息 QM 文件(.qm)。因此，这些文件没有添加到 RPM 软件包中。

有了这个更新，这个问题已被解决，现在在构建或重建 RPM 时可打包 3 字母的 Qt 消息 QM 文件。

[Bugzilla:2144005](#)

## 8.4. SHELL 和命令行工具

**ReaR 在 IBM Z 构架上可正确处理排除的 DASD**

在以前的版本中，在 IBM Z 构架上，ReaR 在恢复过程中重新格式化所有直接连接的存储设备(DASD)，包括用户从保存的布局排除的 DASD，且不会恢复其内容。因此，如果您从保存的布局中排除了一些 DASD，则其数据会在系统恢复过程中丢失。有了这个更新，ReaR 不再在系统恢复过程中格式排除的 DASD，包括从中引导 ReaR 救援系统的设备（使用 zipl 引导装载程序）。在 ReaR 重新格式化 DASD 前，系统会提示您确认 DASD 格式化脚本。这样可确保排除的 DASD 中的数据在系统恢复后依然存在。

[Bugzilla:2172589](#)

**ReaR 不再无法恢复非 LVM XFS 文件系统**

在以前的版本中，当您使用 ReaR 恢复带有某些设置和磁盘映射的非 LVM XFS 文件系统时，ReaR 会使用默认设置而不是指定的设置创建文件系统。例如，如果您有一个 **sunit** 和 **swidth** 参数设为非零值的文件系统，并且您使用带有磁盘映射的 ReaR 恢复了文件系统，则使用忽略指定的值的默认 **sunit** 和 **swidth** 参数创建文件系统。因此，ReaR 会在使用特定 XFS 选项挂载文件系统过程中失败。有了这个更新，ReaR 可以使用指定的设置正确地恢复文件系统。

[Bugzilla:2160748](#)

**wsmancli 正确地处理 HTTP 401 Unauthorized 状态**

使用 Web 服务管理协议的管理系统的 **wsmancli** 工具现在处理身份验证，以更好地符合 RFC 2616。

在以前的版本中，当连接到需要身份验证的服务时，**wsmancli** 命令会在收到 HTTP 401 Unauthorized 响应后立即返回错误消息 **Authentication failed, please retry**，例如因为凭证不完整。要继续操作，**wsmancli** 会提示您提供用户名和密码，即使您已提供了凭证的一部分。

有了这个更新，**wsmancli** 只需要之前未提供的凭证。因此，第一次身份验证尝试不会显示任何错误消息。只有在提供了完整的凭证并且身份验证失败后，才会显示错误消息。

[Bugzilla:2127416](#)

## 8.5. 安全性

**即使未定义 RuleFile，USBGuard 也会保存规则**

在以前的版本中，如果在 USBGuard 中设置了 **RuleFolder** 配置指令，但没有设置 **RuleFile**，则规则集无法更改。有了这个更新，即使设置了 RulePRESS，但没有设置 RuleFile，您也可以更改规则集。因此，您可以修改 USBGuard 中的永久策略，以永久保存新添加的规则。

[Bugzilla:2155910](#)

## python-sqlalchemy rebase 到 1.4.45

**python-sqlalchemy** 软件包已 rebase 至版本 1.4.45，与版本 1.4.37 相比，它提供了很多 bug 修复。最值得注意的是，这个版本包含了对缓存密钥生成中关键内存 bug 的一个修复。

[Bugzilla:2152649](#)

## crypto-policies 现在为 BIND 禁用了 NSEC3DSA

在以前的版本中，系统范围的加密策略不会控制 BIND 配置中的 NSEC3DSA 算法。因此，在 DNS 服务器上没有禁用不满足当前安全要求的 NSEC3DSA。有了这个更新，所有加密策略默认在 BIND 配置中禁用 NSEC3DSA。

[Bugzilla:2152635](#)

## SECLEVEL=3 中的 openssl 现在可以与 PSK 密码套件一起使用

在以前的版本中，预共享密钥(PSK)密码套件不被识别为执行完美转发保密(PFS)密钥交换方法。因此，**ECDHE-PSK** 和 **DHE-PSK** 密码套件无法与 OpenSSL 一起使用来配置为 **SECLEVEL=3**，例如当系统范围的加密策略被设为 **FUTURE** 时。**openssl** 软件包的新版本解决了这个问题。

[Bugzilla:2060044](#)

## Clevis 现在可以正确跳过 crypttab 中注释的设备

在以前的版本中，Clevis 会尝试解锁 **crypttab** 文件中注释的设备，从而导致即使设备无效，**clevis-luks-askpass** 服务也会运行。这会导致不必要的服务运行，并很难进行故障排除。

有了这个修复，Clevis 会忽略注释的设备。现在，如果无效的设备被注释掉了，Clevis 不会尝试解锁它，**clevis-luks-askpass** 会正确完成。这样可以更轻松地排除故障并减少不必要的服务运行。

[Bugzilla:2159728](#)

## Clevis 不再从 pwmake 请求太多的熵

在以前的版本中，当 Clevis 使用 **pwmake** 创建密码，以在 **LUKS** 元数据中存储数据时，**pwmake** 密码生成工具会显示不需要的警告，这会导致 Clevis 使用较低的熵。有了这个更新，Clevis 仅限于为 **pwmake** 提供的 256 熵位，这消除了不需要的警告，并使用正确的熵数。

[Bugzilla:2159735](#)

## USBGuard 不再造成混淆的警告

在以前的版本中，当父进程比第一个子进程早完成时，USBGuard 中可能会发生竞争条件。因此，**systemd** 会报告存在一个具有错误标识的父 PID (PPID) 的进程。有了这个更新，父进程会在工作模式下等待第一个子进程完成。因此，**systemd** 不再报告这样的警告。

[Bugzilla:2042345](#)

## OOM 终止程序不再过早地终止 usbguard

在以前的版本中，**usbguard.service** 文件不包含 **systemd** 服务的 **OOMScoreAdjust** 选项的定义。因此，当系统资源不足时，**usbguard-daemon** 进程可能会在其他非特权进程之前被终止。有了这个更新，**usbguard.service** 文件现在包含 **OOMScoreAdjust** 设置，这可防止内存不足(OOM)终止程序过早终止 **usbguard-daemon** 进程。

[Bugzilla:2097419](#)

## logrotate 不再错误地在日志轮转中发送 Rsyslog 信号

在以前的版本中，**logrotate** 脚本中参数顺序被错误地设置，这会导致语法错误。这会导致 **logrotate** 在日志轮转过程中无法正确发送 Rsyslog 信号。

有了这个更新，**logrotate** 中的参数顺序被修复，即使设置了 **POSIXLY\_CORRECT** 环境变量，在日志轮转后 **logrotate** 可以正确地发送 Rsyslog 信号。

[Bugzilla:2124488](#)

### **imklog** 不再对缺失的对象调用 **free ()**

在以前的版本中，**imklog** 模块对已经释放的对象调用称为 **free ()** 的函数。因此，**imklog** 可能会导致分段错误。有了这个更新，对象不再释放两次。

[Bugzilla:2157659](#)

### **fagenrules --load** 现在可以正常工作

在以前的版本中，**fapolicyd** 服务无法正确处理信号挂起 (SIGHUP)。因此，在接收 SIGHUP 后 **fapolicyd** 终止，**fagenrules --load** 命令无法正常工作。此更新包含针对此问题的修复。因此，**fagenrules --load** 现在可以正常工作，规则更新不再需要手动重启 **fapolicyd**。

[Bugzilla:2070655](#)

### 扫描和补救可以正确地忽略 SCAP 审计规则审计密钥

在以前的版本中，在没有审计密钥 (**-k** 或 **-F** 密钥) 的情况下定义的审计监控规则可能会遇到以下问题：

- 即使规则的其他部分都正确，该规则也被标记为不合规。
- Bash 补救修复了监视规则的路径和权限，但它没有正确地添加审计密钥。
- 补救有时不能修复缺少的密钥，返回一个 **error**，而不是 **固定** 的值。

这会影响以下规则：

- **audit\_rules\_login\_events**
- **audit\_rules\_login\_events\_faillock**
- **audit\_rules\_login\_events\_lastlog**
- **audit\_rules\_login\_events\_tallylog**
- **audit\_rules\_usergroup\_modification**
- **audit\_rules\_usergroup\_modification\_group**
- **audit\_rules\_usergroup\_modification\_gshadow**
- **audit\_rules\_usergroup\_modification\_opasswd**
- **audit\_rules\_usergroup\_modification\_passwd**
- **audit\_rules\_usergroup\_modification\_shadow**
- **audit\_rules\_time\_watch\_localtime**
- **audit\_rules\_mac\_modification**



- `audit_rules_networkconfig_modification`
- `audit_rules_sysadmin_actions`
- `audit_rules_session_events`
- `audit_rules_sudoers`
- `audit_rules_sudoers_d`

有了这个更新，审计密钥已从检查和 Bash 以及 Ansible 补救中删除。因此，在检查和补救过程中由 key 字段导致的不一致问题不再发生，审核员可以任意选择这些密钥，以便更轻松地搜索审计日志。

[Bugzilla:2120978](#)

### 在测试访问多个 IMA 测量文件的系统时，Keylime 不再失败

在以前的版本中，如果运行 Keylime 代理的系统快速连续访问由完整性测量架构(IMA)测量的多个文件时，Keylime 验证器会错误地处理 IMA 日志添加。因此，运行的哈希值与正确的平台配置寄存器(PCR)状态不匹配，系统会在测试时失败。这个更新解决了这个问题，快速访问多个测量文件的系统不会在测试时失败。

[Bugzilla:2138167](#)

### Keylime 策略生成脚本不再导致分段错误和内核转储

`create_mb_refstate` 脚本为 Keylime 中测量的引导测试生成策略。在以前的版本中，`create_mb_refstate` 错误地计算 `DevicePath` 字段中的数据长度。因此，脚本会尝试使用错误计算的长度访问无效的内存，这会导致分段错误和内核转储。

此更新已在 [RHBA-2023:0309](#) 公告中发布，在处理测量的引导事件日志时防止分段错误。因此，您可以生成一个测量的引导策略。

[Bugzilla:2140670](#)

### TPM 证书不再导致 Keylime 注册中心崩溃

在以前的版本中，Keylime TPM 证书存储中的一些证书是格式错误的 x509 证书，并导致 Keylime 注册中心崩溃。这个更新解决了这个问题，Keylime 注册中心不再会因为错误的证书而崩溃。

[Bugzilla:2142009](#)

## 8.6. 网络

### NetworkManager 现在会在获取新的 DHCP 租期前在重新应用期间保留 IP 地址

在以前的版本中，在更改连接设置并使用 `nmcli device reapply` 命令后，NetworkManager 不会保留 DHCP 租期。因此，IP 地址会被临时删除。有了这个修复，NetworkManager 保留 DHCP 租期，并使用它直到租期过期或客户端请求了新的租期。因此，当 `nmcli device reapply` 命令重启 DHCP 客户端时，它不会临时删除 IP 地址。

[Bugzilla:2117352](#)

### firewalld 服务现在只在使用直接规则时触发 ipset 弃用警告

在以前的版本中，`firewalld` 服务在不需要时使用已弃用的 `ipset` 内核模块。因此，RHEL 会记录模块的弃用警告，这可能是误导，因为 `firewalld` 的 `ipset` 功能未弃用。有了这个更新，`firewalld` 只使用已弃用的 `ipset` 模块，并在用户明确使用带有 `--direct` 选项的 `ipsets` 时记录警告。

[Bugzilla:2122678](#)

## HNV 接口现在在重启后显示选项

在以前的版本中，**nmcli** 工具使用 NetworkManager API 创建混合网络虚拟化(HNV)绑定。因此，在重启后，HNV 绑定会丢失主端口设置。有了这个修复，**nmcli** 现在使用 **hcnmgr** 来为主端口设置绑定选项。**hcnmgr** 工具支持使用混合网络的单根输入/输出虚拟化(SR-IOV)的实时分区的迁移。因此，HNV 绑定接口在重启后会显示 **active slave/primary\_reselect** 选项。

[Bugzilla:2125152](#)

## 8.7. 内核

### 启用了安全引导的 fadump 可以正常工作

在以前的版本中，当在安全引导环境中启用了固件辅助转储(FADump)，且任何引导组件超过了分配的内存区域，系统重启会导致 GRUB 内存不足(OOM)状态。这个更新在 **kexec-tools** 中提供了一个修复，这样安全引导和 FA 转储可以一起工作。

[Bugzilla:2139000](#)

## 8.8. 引导加载程序

### grubby 现在可以正确地将参数传给新内核

当您使用 **grubby** 工具添加新内核且不指定任何参数，或者将参数留空时，**grubby** 不会将任何参数传给新内核，也不会设置 **root**。使用 **--args** 和 **--copy-default** 选项可确保新参数被追加到默认参数后面。

[Bugzilla:2127453](#)

### 现在，即使 PReP 大小不是 4 或 8 MiB，RHEL 安装也会成功

在以前的版本中，如果 PowerPC Reference Platform (PReP)分区的大小与使用 4 kiB 扇区的磁盘上的 4 MiB 或 8 MiB 不同，RHEL 安装程序无法安装引导装载程序。因此，您无法在磁盘上安装 RHEL。

有了这个版本，这个问题已被解决。因此，安装程序现在可以按预期在磁盘上安装 RHEL。

[Bugzilla:2026579](#)

## 8.9. 文件系统和存储

### 安装程序创建扇区大小为 512 字节的 LUKSv2 设备

在以前的版本中，如果磁盘有 4096 字节的物理扇区，RHEL 安装程序会创建 4096 字节扇区的 LUKSv2 设备。有了这个更新，安装程序会创建扇区大小为 512 字节的 LUKSv2 设备，以提供更好的磁盘兼容性，与不同物理扇区大小一起在一个 LVM 物理卷组中使用，即使 LVM 物理卷是加密的。

[Bugzilla:2103800](#)

### supported\_speeds sysfs 属性报告正确的速度值

在以前的版本中，由于 **qla2xxx** 驱动程序中不正确的定义，HBA 的 **supported\_speeds sysfs** 属性会报告 20 Gb/s 速度，而不是预期的 64 Gb/s 速度。因此，如果 HBA 支持 64 Gb/s 连接速度，则 **supported\_speeds sysfs** 值不正确，这会影响报告的速度值。

有了这个更新，HBA 的 **supported\_speeds sysfs** 属性会报告正确的速度值，值为 16 Gb/s、32 Gb/s 和 64 Gb/s。您可以通过执行 **cat /sys/class/fc\_host/host the/supported\_speeds** 命令来查看速度值。

Bugzilla:2069758

### lpfc 驱动程序在 D\_ID 端口交换过程中处于有效状态

在以前的版本中，在发出 NetApp giveback 操作后 SAN Boot 主机可能会导致 LVM 挂起任务警告和停滞的 I/O。由于光纤通道 D\_ID 端口交换，也会在 DM-Multipath 环境中提供备用路径时发生此问题。由于竞争条件，D\_ID 端口交换会导致 lpfc 驱动程序中的状态不一致，这会阻止 I/O 被发出。

有了这个修复，lpfc 驱动程序在 D\_ID 端口交换发生时确保有效状态。因此，光纤频道 D\_ID 端口交换不会挂起 I/O。

Bugzilla:2173947

## 8.10. 高可用性和集群

### pcs 不再允许您修改不应更改的集群属性

在以前的版本中，pcs 命令行界面允许您修改不应更改或更改不起作用的集群属性。有了这个修复，pcs 不再允许您修改这些集群属性：**cluster-infrastructure**、**cluster-name**、**dc-version**、**have-watchdog**、**last-lrm-refresh**。

Bugzilla:1620043

### pcs 现在显示没有明确配置的集群属性

在以前的版本中，要显示特定集群属性值的 pcs 命令不会列出 CIB 中没有明确配置的值。有了这个修复，如果集群属性没有设置，pcs 会显示属性的默认值。

Bugzilla:1796827

### 调用 crm\_mon 的集群资源现在在关闭时完全停止

在以前的版本中，crm\_mon 工具会在 Pacemaker 关闭过程中返回一个非零退出状态。在监控操作中调用 crm\_mon 的资源代理（如 **ocf:heartbeat:pqsql**）可能会在集群关闭时错误地返回一个错误。有了这个修复，即使集群处于关闭过程中，crm\_mon 也会返回成功。现在，调用 crm\_mon 的资源会在集群关闭时完全停止。

Bugzilla:2133546

### OCF 资源代理元数据操作现在可以调用 crm\_node，而不会导致意外隔离

从 RHEL 8.5 开始，OCF 资源代理元数据操作会阻止控制器，crm\_node 查询会执行控制器请求。因此，如果代理的元数据操作调用 crm\_node，则它将阻止控制器 30 秒，直到操作超时为止。这可能导致其他操作失败，并使节点被隔离。

有了这个修复，控制器会异步执行元数据操作。OCF 资源代理元数据操作现在可以调用 crm\_node 而没有任何问题。

Bugzilla:2125344

### Pacemaker 现在会在资源顺序更改时立即重新检查资源分配

从 RHEL 8.7 开始，当 CIB 中的资源顺序发生改变而没有更改资源定义时，Pacemaker 不会重新检查资源分配。如果配置重新排序会导致资源移动，则这不会在下次自然转换前发生，直到达到值 **cluster-recheck-interval-property** 为止。如果没有为资源配置粘性，这可能会导致问题。

有了这个变化，当 CIB 中的资源顺序更改时，Pacemaker 会重新检查资源分配，就像之前的 Pacemaker 版本所做的那样。如果需要，现在集群会立即响应这些更改。



[Bugzilla:2125337](#)

### 启用单个资源并监控操作不再为资源组中的所有资源启用监控操作

在以前的版本中，在不管理资源组中所有资源和监控操作后，管理该组中的一个资源及其监控操作，会为资源组中的所有资源重新启用监控操作。这可能会出发意外的集群行为。

有了这个修复，管理资源和重新启用其监控操作只会重新启用对该资源的监控操作，而不会启用对资源组中其他资源的监控操作。

[Bugzilla:2092950](#)

## 8.11. 编译器和开发工具

### 现在，即使某些 CNAME 记录无效，DNS 查找也会成功

在以前的版本中，**glibc** DNS stub 解析器使用不是主机名的所有者名称作为 DNS 数据包错误来处理 CNAME 记录。因此，由于 DNS 数据包错误，DNS 查询失败。有了这个更新，**glibc** stub 解析器会跳过无效的 CNAME 记录，且不会提取相应的别名信息。因此，即使服务器响应包含不是主机名的域名的 CNAME 链，DNS 查找也可以成功。

[Bugzilla:2129005](#)

### 在 x509 FIPS 模式下，golang 现在支持 4096 位密钥

在以前的版本中，在 x509 FIPS 模式下，**golang** 不支持 4096 位密钥。因此，当用户使用 4096 位密钥时，程序会崩溃。有了这个更新，在 x509 FIPS 模式下，**golang** 现在支持 4096 位密钥。

[Bugzilla:2133019](#)

### 您可以使用 pip 在所有构架上安装 SciPy

在以前的版本中，**openblas-devel** 软件包不包含 OpenBLAS 库的 pkg-config 文件。因此，在某些情况下，在使用 OpenBLAS 编译时，无法使用 **pkgconf** 工具确定编译器和链接器标志。例如，这会导致 64 位 IBM Z 和 IBM Power Systems Little Endian 架构上 **pip install scipy** 命令失败。

这个更新向所有支持的构架上的 **openblas-devel** 软件包中添加了 **openblas.pc** 文件。因此，您可以使用 **pip** 软件包安装程序安装 SciPy 库。

请注意，在 RHEL 9 中，建议针对 **flexiblas-devel** 软件包构建应用程序，并将项目链接到 FlexiBLAS 包装程序库。

[Bugzilla:2115737](#)

### 如果 TZ 数据中存在任何 DST 规则，则 glibc 中的 tzset 函数现在将 daylight 变量设为非零值

在以前的版本中，如果时区数据文件中最后的 DST 转换没有因为标准时间偏移的同时变化而导致时钟变化，则 **glibc** 中的 **tzset** 函数会将 daylight 变量设为 0。因此，当应用程序使用 daylight 变量检查 DST 是否活跃时，它们不会得到正确的结果，会根据此信息执行错误的操作。要解决这个问题，如果时区数据中存在任何 DST 规则，则 **tzset** 功能现在将 daylight 变量设为非零值，而不考虑偏移量。现在，无论偏移如何变化，应用程序都会观察 DST 规则是否存在。

[Bugzilla:2155352](#)

### OpenJDK RSAPSS 签名实现现在在使用 RSA 密钥前会先验证它们

在以前的版本中，OpenJDK 中的 RSAPSS 签名实现不会完全检查是否在尝试使用 RSA 密钥前，它们可以被 SunRSASign 使用，这会在使用自定义安全提供商时产生错误。这个 bug 现已修复，因此 RSAPSS 签名实现现在会验证 RSA 密钥，并允许其他提供商在无法处理这些密钥时处理这些密钥。

[Bugzilla:2188023](#)

### OpenJDK XML 签名提供者现在可以在 FIPS 模式下正常工作

在以前的版本中，OpenJDK XML 签名提供者无法在 FIPS 模式下操作。由于对 FIPS 模式支持的改进，OpenJDK XML 签名提供者现在在 FIPS 模式下启用。

[Bugzilla:2186810](#)

### FIPS 模式下的 OpenJDK 不再遇到某些 PKCS#11 令牌的意外错误

在以前的版本中，在 FIPS 模式下被 OpenJDK 使用前，一些 PKCS#11 令牌没有被完全初始化，从而导致意外错误。有了这个升级，这些错误是预期的，可由 FIPS 支持的代码处理。

[Bugzilla:2186806](#)

## 8.12. 身份管理

### 现在，对外部 IdP 进行身份验证都需要客户端 secret

在以前的版本中，SSSD 无法正确将客户端 secret 传给外部身份提供者(IdP)。因此，针对之前使用 **ipa idp-add --secret** 命令配置的需要一个客户端 secret 的外部 IdP 的身份验证会失败。有了这个更新，SSSD 将客户端 secret 传给 IdP，用户可以进行身份验证。

[Jira:RHELPLAN-148303](#)

### IdM 现在支持使用 Ansible 为 sudo 规则设置 hostmask

在以前的版本中，**ipa sudorule-add-host** 命令允许设置 **sudo** 规则使用的 hostmask，但此选项没有出现在 **ansible-freeipa** 软件包中。有了这个更新，您现在可以使用 **ansible-freeipa hostmask** 变量定义一个 hostmask 的列表，以便身份管理中定义的一个特定的 **sudo** 规则应用于此表。

现在，您可以使用 Ansible 自动为 IdM **sudo** 规则设置主机掩码。

[Bugzilla:2127913](#)

### 现在，当使用带有 db\_dir 参数的自定义路径时，dscreate 工具现在可以正常工作

在以前的版本中，使用自定义目录路径的实例启动失败，因为自定义目录有错误的 SELinux 标签。因此，SELinux 拒绝访问这些目录，实例没有被创建。有了这个版本，**dscreate** 工具为自定义实例目录设置正确的 SELinux 标签。

[Bugzilla:1924569](#)

### 目录服务器复制管理器帐户的密码更改现在可以正常工作

在以前的版本中，更改密码后，目录服务器无法正确更新复制协议的密码缓存。因此，当您更改复制管理器帐户的密码时，复制会失败。有了这个更新，目录服务器可以正确地更新缓存，因此复制可以按预期工作。

[Bugzilla:1956987](#)

### IdM 客户端安装程序不再在 ldap.conf 文件中指定 TLS CA 配置

在以前的版本中，IdM 客户端安装程序在 **ldap.conf** 文件中指定 TLS CA 配置。有了这个更新，OpenSSH 使用默认的信任存储，IdM 客户端安装程序不会在 **ldap.conf** 文件中设置 TLS CA 配置。

[Bugzilla:2094673](#)

### 当可信 AD 用户的名称包含混合问题单字符时，IdM 客户端可以正确地检索它们的信息

在以前的版本中，如果您尝试用户查找或用户的身份验证，并且可信活动目录(AD)用户在其名称中包含混合大小写字符，且在 IdM 中使用覆盖进行了配置，则会返回一个错误，阻止用户访问 IdM 资源。

随着 [RHBA-2023:4359](#) 的发布，区分大小写的比较被忽略字符大小写的不区分大小写的比较替代。因此，IdM 客户端现在可以查找 AD 可信域的用户，即使其用户名包含混合大小写字符，且它们在 IdM 中使用覆盖进行了配置。

JIRA:SSSD-6096

## 8.13. 图形基础结构

### Matrox G200e 现在在 VGA 显示器上显示输出

在以前的版本中，如果您使用以下系统配置，您的显示器可能不会显示图形输出：

- Matrox G200e GPU
- 通过 VGA 控制器连接的显示器

因此，您不能在此配置上使用或安装 RHEL。

有了这个版本，这个问题已被解决。因此，RHEL 会按预期正常引导并显示图形输出。

[Bugzilla:1960467](#)

## 8.14. WEB 控制台

### Web 控制台 NBDE 绑定步骤现在也适用于带有根文件系统的卷组

在 RHEL 9.2.0 中，由于用来确定用户是否向 root 文件系统添加了 Tang 密钥的代码中有一个错误，当 LUKS 容器上没有文件系统时，web 控制台中的绑定过程会崩溃。因为在点 **Verify key** 对话框中的 **Trust key** 按钮后 web 控制台会显示错误信息 **TypeError: Qe (...) is undefined**，所以在上述场景中，您必须在命令行界面中执行所有必要的步骤。

随着 [RHBA-2023:4346](#) 公告的发布，Web 控制台可以正确地处理向根文件系统添加 Tang 密钥。因此，在不同场景中，web 控制台完成了使用 Network-Bound Disk Encryption (NBDE)自动解锁 LUKS 加密卷所需的所有绑定步骤。

[Bugzilla:2207498](#)

## 8.15. RED HAT ENTERPRISE LINUX 系统角色

### nbde\_client 系统角色现在可以正确处理 clevis-luks-askpass 的不同名称

**nbde\_client** 系统角色已更新，来处理 **clevis-luks-askpass systemd** 单元在其上具有不同名称的系统。该角色现在可以与受管节点上的 **clevis-luks-askpass** 不同的名称一起正常工作，这需要解锁在引导过程后期挂载的 LUKS 加密的卷。

[Bugzilla:2126959](#)

## ha\_cluster 系统角色日志不再显示未加密的密码和 secret

**ha\_cluster** 系统角色接受可以是密码或其他 secret 的参数。在以前的版本中，一些任务会记录其输入和输出。因此，角色日志可能包含未加密的密码和其他 secret。

有了这个更新，任务已被修改为使用 Ansible **no\_log: true** 指令，任务输出不再在角色日志中显示。**ha\_cluster** 系统角色日志不再包含密码和其他 secret。虽然此次更新保护了安全信息，但在调试配置时，角色日志现在提供可以使用的较少的信息。

[Bugzilla:2143816](#)

## 使用 ha\_cluster 系统角色配置的使用 SBD 且未在引导时启动的集群现在可以正常工作

在以前的版本中，如果用户使用 **ha\_cluster** 系统角色将集群配置为使用 SBD 且没有在引导时启动，则 SBD 服务会被禁用，并且 SBD 不会启动。有了这个修复，如果集群被设置为使用 SBD，而不管集群是否配置为在引导时启动，SBD 服务总是被启用。

[Bugzilla:2153030](#)

## 启用隐式文件提供程序来修复 cockpit-session-recording SSSD 配置

禁用的 SSSD 隐式文件提供程序会导致 **cockpit-session-recording** 模块创建无效的系统安全服务守护进程(SSSD)配置。此更新无条件地启用文件提供程序，因此 **cockpit-session-recording** 创建的 SSSD 配置现在可以按预期工作。

[Bugzilla:2153043](#)

## nbde\_client\_clevis 角色不再向用户报告回溯

在以前的版本中，**nbde\_client\_clevis** 角色有时会失败，从而导致回溯并报告敏感数据返回给用户，如 **encryption\_password** 字段。有了这个更新，该角色不再报告敏感数据，只报告合适的错误消息。

[Bugzilla:2162782](#)

## 使用 ha\_cluster 系统角色设置 stonith-watchdog-timeout 属性现在可以在停止的集群中正常工作

在以前的版本中，当您使用 **ha\_cluster** 系统角色在停止的集群中设置 **stonith-watchdog-timeout** 属性时，属性会恢复回其之前的值，角色失败。有了此修复，使用 **ha\_cluster** 系统角色配置 **stonith-watchdog-timeout** 属性可以正常工作。

[Bugzilla:2167528](#)

## 现在，当使用带有 networking RHEL 系统角色的 initscripts 时，网络流量现在被定向到预期的网络接口

在以前的版本中，当使用 **initscripts** 提供程序时，网络连接的路由配置没有指定流量应该通过的输出设备。因此，内核可以使用与用户预期的输出设备不同的输出设备。现在，如果在 playbook 中为连接指定了网络接口名称，则它将用作路由配置文件中的输出设备。这与 NetworkManager 的行为一致后者会在设备上激活配置集时在路由中配置输出设备。因此，用户可以确保流量被定向到预期的网络接口。

[Bugzilla:2168735](#)

## selinux 角色现在以幂等方式管理策略模块

在以前的版本中，**selinux** 角色每次都将在现有模块复制到受管节点，即使模块已存在也会报告更改。有了此更新，**selinux** 角色会检查该模块是否已安装到受管节点上，如果已安装，则不会尝试复制并安装该模块。

[Bugzilla:2160152](#)

## 当 `rhc_auth` 包含激活码时，`rhc` 系统角色不会在注册的系统上失败

在以前的版本中，当使用 `rhc_auth` 参数中指定的激活码在注册的系统上执行 playbook 文件时，会出现失败。这个问题已解决。现在，可以在已经注册的系统上执行 playbook 文件，即使在 `rhc_auth` 参数中提供了激活码。

[Bugzilla:2186218](#)

## 8.16. 虚拟化

### 嵌套虚拟机上的系统时间现在工作正常

在以前的版本中，在某些情况下，嵌套虚拟机(VM)上的系统时间与级别 0 和级别 1 主机不同步。这有时会导致嵌套虚拟机变得无响应或意外终止。

有了这个更新，KVM 主机内核代码中的时间处理代码已被修复，这可以防止上面描述的错误发生。

[Bugzilla:2140899](#)

### 在使用 `memfd` 内存支持时，IBM Z 上的虚拟机不再启动失败

在以前的版本中，在 IBM Z 主机上，如果虚拟机(VM)被配置为使用 `memfd` 的巨页内存支持，则其无法启动，如下所示：

```
<memoryBacking>
  <hugepages/>
  <source type='memfd'/>
</memoryBacking>
```

有了这个更新，底层原因已被修复，受影响的虚拟机现在可以正常启动。

[Bugzilla:2116496](#)

### VNC 现在可在迁移后可靠地连接到 UEFI 虚拟机

在以前的版本中，如果您在迁移虚拟机(VM)过程中启用或禁用了消息队列，则虚拟网络计算(VNC)客户端在迁移完成后无法连接到虚拟机。

这个问题只会影响使用 Open Virtual Machine Firmware (OVMF)的基于 UEFI 的虚拟机。

这个问题已被解决，VNC 客户端现在在迁移完成后可以可靠地连接到 UEFI 虚拟机。

[Jira:RHELPLAN-135600](#)

### 安装程序显示要在虚拟机上安装 RHEL 的预期的系统磁盘

在以前的版本中，当使用 `virtio-scsi` 设备在虚拟机上安装 RHEL 时，这些设备可能会因为 `device-mapper-multipath` bug 而不在安装程序中出现。因此，在安装过程中，如果某些设备设置了串口，而有些设备没有，则 `multipath` 命令会声明所有具有串口的设备。因此，安装程序无法在虚拟机中找到要安装 RHEL 的预期的系统磁盘。

有了这个更新，`multipath` 可以正确地将没有串口的设备设置为没有全局识别符(WWID)，并忽略它们。在安装时，`multipath` 只声明 `multipathd` 用来绑定多路径设备的设备，安装程序会在虚拟机中显示要安装 RHEL 的预期的系统磁盘。

[Bugzilla:1926147](#)



## 第 9 章 技术预览

这部分列出了 Red Hat Enterprise Linux 9 中的所有技术预览。

如需有关红帽对技术预览功能支持范围的信息，请参阅 [技术预览功能支持范围](#)。

### 9.1. 安装程序和镜像创建

现在在 RHEL 安装程序中，**光纤通道设备上的 NVMe 作为技术预览提供**

现在，您可以将光纤通道设备上的 NVMe 作为技术预览添加到 RHEL 安装中。在 RHEL 安装程序中，您可以在 Installation Destination 屏幕上添加磁盘时，在 NVMe Fabrics Devices 部分中选择这些设备。

[Bugzilla:2107346](#)

### 9.2. SHELL 和命令行工具

**GIMP 在 RHEL 9 中作为技术预览提供**

GNU Image Manipulation Program(GIMP)2.99.8 现在作为技术预览在 RHEL 9 中提供。**gimp** 软件包版本 2.99.8 是一个预发行版本，它有一组改进，但只能保证稳定性。发布官方 GIMP 3 后，将作为此预发布版本的更新，在 RHEL 9 中引入。

在 RHEL 9 中，您可以作为 RPM 软件包轻松安装 **gimp**。

[Bugzilla:2047161](#)

### 9.3. 基础架构服务

**TuneD 的套接字 API 作为技术预览提供**

通过 Unix 域套接字控制 TuneD 的套接字 API 现在作为技术预览提供。套接字 API 将一对一与 D-Bus API 映射，并为 D-Bus 不可用的情况提供替代通信方法。通过使用套接字 API，您可以控制 TuneD 守护进程来优化性能，并更改各种调优参数的值。套接字 API 默认被禁用，您可以在 **tuned-main.conf** 文件中启用它。

[Bugzilla:2113900](#)

### 9.4. 安全性

**gnutls 现在使用 KTLS 作为技术预览**

更新的 **gnutls** 软件包可以使用 Kernel TLS (KTLS) 在加密频道上加速数据传输作为技术预览。要启用 KTLS，请使用 **modprobe** 命令添加 **tls.ko** 内核模块，并创建一个新的配置文件 **/etc/crypto-policies/local.d/gnutls-ktls.txt**，其中包含以下内容：

```
[global]
ktls = true
```

请注意，当前版本不支持通过 TLS **KeyUpdate** 消息更新流量密钥，这会影响到 AES-GCM passwordsuites 的安全性。如需更多信息，请参阅 [RFC 7841 - TLS 1.3](#) 文档。

[Bugzilla:2042009](#)

## 9.5. 网络

### WireGuard VPN 作为技术预览提供

WireGuard (红帽作为技术预览提供) 是一个在 Linux 内核中运行的高性能 VPN 解决方案。它使用现代加密, 比其他 VPN 解决方案更容易配置。此外, 因为 WireGuard 较小的代码基础, 减少了受攻击的风险, 因此提高了安全性。

详情请查看[设置 WireGuard VPN](#)。

Bugzilla:1613522

### KTLS 作为技术预览提供

RHEL 作为技术预览提供内核传输层(KTLS)。KTLS 使用内核中的对称加密或者解密算法为 AES-GCM 密码处理 TLS 记录。KTLS 还包括将 TLS 记录加密卸载到提供此功能的网络接口控制器(NIC)的接口。

Bugzilla:1570255

### systemd-resolved 服务作为技术预览提供

**systemd-resolved** 为本地应用程序提供名字解析。该服务实现了缓存和验证 DNS stub 解析器、链接本地多播名称解析(LLMNR)和多播 DNS 解析器和响应程序。

请注意, **systemd-resolved** 是一个不受支持的技术预览。

[Bugzilla:2020529](#)

## 9.6. 内核

### SGX 作为技术预览

软件扩展 (SGX) 是一个 Intel® 技术, 用于保护软件代码和数据不受公开和修改的影响。RHEL 内核部分提供 SGX v1 和 v1.5 功能。版本 1 启用使用 **Flexible Launch Control** 机制的平台使用 SGX 技术。

Bugzilla:1874182

### 用于内核的 Intel 数据流加速器驱动程序作为技术预览提供

内核的 Intel 数据流加速器驱动程序(IDXD)目前作为技术预览提供。它是一个 Intel CPU 集成的加速器, 包括共享工作队列 ID(pasid)提交和共享虚拟内存(SVM)。

[Bugzilla:2030412](#)

### Soft-iWARP 驱动程序作为技术预览提供

软硬件硬件(siw)是一种软件, 互联网是 RDMA 协议(iWARP), 适用于 Linux 的内核驱动程序。soft-iWARP 通过 TCP/IP 网络堆栈实施 iWARP 协议套件。这个协议套件在软件中完全实现, 不需要特定的远程直接内存访问(RDMA)硬件。Soft-iWARP 使具有标准以太网适配器的系统连接到 iWARP 适配器或安装了 Soft-iWARP 的其他系统。

Bugzilla:2023416

### SGX 作为技术预览

软件扩展 (SGX) 是一个 Intel® 技术, 用于保护软件代码和数据不受公开和修改的影响。RHEL 内核部分提供 SGX v1 和 v1.5 功能。版本 1 使用 **Flexible Launch Control** 机制启用平台, 以使用 SGX 技术。版本 2 添加了 **Enclave Dynamic Memory Management(EDMM)**。主要特性包括:

- 修改属于初始化 enclave 的常规 enclave 页的 EPCM 权限。
- 动态将常规 enclave 页添加到初始化的 enclave。
- 扩展初始化的 enclave ， 以容纳更多线程。
- 从初始化的 enclave 中删除常规的 enclave 页和 TCS 页。

Bugzilla:1660337

### rvu\_af,rvu\_nicpf 和 rvu\_nicvf 作为技术预览提供

对于 Marvell OCTEON TX2 Infrastructure Processor 系列， 以下内核模块作为技术预览提供：

- **rvu\_nicpf** - Marvell OcteonTX2 NIC 物理功能驱动程序
- **rvu\_nicvf** - Marvell OcteonTX2 NIC 虚拟功能驱动程序
- **rvu\_nicvf** - Marvell OcteonTX2 RVU Admin 功能驱动程序

Bugzilla:2040643

## 9.7. 文件系统和存储

### DAX 现在作为技术预览供 ext4 和 XFS 使用

在 RHEL 9 中， DAX 文件系统作为技术预览提供。 DAX 提供了将持久内存直接映射到其地址空间的方法。 要使用 DAX， 系统必须有某种可用的持久性内存， 通常使用一个或多个非线性内存模块(NVDIMM)， 必须在 NVDIMM 上创建 DAX 兼容文件系统。 另外， 该文件系统必须使用 **dax** 挂载选项挂载。 然后， 在 **dax** 挂载的文件系统中的文件 **mmap** 会导致存储直接映射到应用程序的地址空间中。

Bugzilla:1995338

### Stratis 作为技术预览提供

Stratis 是一个本地存储管理器。 它在存储池的上面为用户提供额外的功能：

- 管理快照和精简配置
- 根据需要自动增大文件系统大小
- 维护文件系统

要管理 Stratis 存储， 使用 **stratis** 工具来与 **stratisd** 后台服务进行通信。

Stratis 作为技术预览提供。

如需更多信息， 请参阅 Stratis 文档：[设置 Stratis 文件系统](#)。

Bugzilla:2041558

### NVMe-oF Discovery Service 功能作为技术预览

NVMe-oF Discovery Service 功能（在 NVMexpress.org 技术 Proposals(TP)8013 和 8014 中）作为技术预览提供。 要预览这些功能， 请使用 **nvme-cli 2.0** 软件包， 并将主机附加到实现 TP-8013 或 TP-8014 的 NVMe-oF 目标设备。 有关 TP-8013 和 TP-8014 的更多信息， 请参阅 <https://nvmexpress.org/specifications/> 网站中的 NVM Express 2.0 Ratified TPs。



Bugzilla:2021672

### NVMe-stas 软件包作为技术预览

**nvme-stas** 软件包，它是 Linux 的中央 Discovery Controller (CDC) 客户端，现在作为技术预览提供。它处理异步事件通知 (AEN)、自动化的 NVMe 子系统连接控制、错误处理和报告以及自动 (**zeroconf**) 和手动配置。

这个软件包由两个守护进程组成，分别是 Storage Appliance Finder (**stafd**) 和存储设备连接器 (**stacd**)。

Bugzilla:1893841

### NVMe TP 8006 in-band 身份验证作为技术预览提供

实现 Non-Volatile Memory Express (NVMe) TP 8006，它是一种针对 NVMe over Fabrics (NVMe-oF) 的带内验证，现在作为不支持的技术预览提供。NVMe Technical Proposal 8006 为 NVMe-oF 定义了 **DH-HMAC-CHAP** 带内验证协议，该协议由这个增强提供。

如需更多信息，请参阅 **nvme-connect (1)** 手册页中的 **dhchap-secret** 和 **dhchap-ctrl-secret** 选项描述。

Bugzilla:2027304

## 9.8. 编译器和开发工具

### jmc-core 和 owasp-java-encoder 作为技术预览

RHEL 9 与 **jmc-core** 和 **owasp-java-encoder** 软件包一起分发，作为 AMD 和 Intel 64 位架构的技术预览功能提供。

**JMC-core** 是一个为 Java Development Kit(JDK)Mission Control 提供核心 API 的库，包括用于解析和编写 JDK Flight Recording 文件的库，以及通过 Java 发现协议(JDP)发现的 Java 虚拟机(JVM)发现库。

**owasp-java-encoder** 软件包提供了 Java 的高性能低后台上下文组。

请注意，自 RHEL 9.2 开始，**jmc-core** 和 **owasp-java-encoder** 在 CodeReady Linux Builder (CRB)存储库中提供，您必须明确启用。如需更多信息，请参阅 [如何在 CodeReady Linux Builder 中启用和使用内容](#)。

[Bugzilla:1980981](#)

## 9.9. 身份管理

### DNSSEC 在 IdM 中作为技术预览提供

带有集成 DNS 的身份管理(IdM)服务器现在实现了 DNS 安全扩展(DNSSEC)，这是一组增强 DNS 协议安全的 DNS 扩展。托管在 IdM 服务器上的 DNS 区可以使用 DNSSEC 自动签名。加密密钥是自动生成和轮换的。

建议那些决定使用 DNSSEC 保护 DNS 区的用户读取并遵循这些文档：

- [DNSSEC 操作实践, 版本 2](#)
- [安全域名系统\(DNS\)部署指南](#)
- [DNSSEC 键翻滚时间注意事项](#)

请注意，集成了 DNSSEC 的 IdM 服务器验证从其他 DNS 服务器获取的 DNS 答案。这可能会影响未按照推荐的命名方法配置的 DNS 区域可用性。

[Bugzilla:2084180](#)

## 身份管理 JSON-RPC API 作为技术预览

一个 API 可用于 Identity Management(IdM)。要查看 API，IdM 还提供了一个 API 浏览器作为技术预览。

在以前的版本中，IdM API 被改进来启用多个 API 命令版本。这些增强可能会以不兼容的方式改变命令的行为。用户现在可以继续使用已有的工具和脚本，即使 IdM API 发生了变化。这可启用：

- 管理员要在服务器中使用之前或更高版本的 IdM，而不是在管理客户端中使用。
- 开发人员可以使用 IdM 调用的特定版本，即使 IdM 版本在服务器上发生了变化。

在所有情况下，与服务器进行通信是可能的，无论是否一方使用，例如，一个新的版本会为这个功能引进新的选项。

有关使用 API 的详细信息，请参阅[使用身份管理 API 与 IdM 服务器通信\(TECHNOLOGY PREVIEW\)](#)。

[Bugzilla:2084166](#)

## sssd-idp 子软件包作为技术预览提供

SSSD 的 **sssd-idp** 子软件包包含 **oidc\_child** 和 **krb5 idp** 插件，它们是对身份管理(IdM)服务器执行 OAuth2 身份验证的客户端组件。此功能仅适用于 RHEL 9.1 及更高版本上的 IdM 服务器。

[Bugzilla:2065693](#)

## SSSD 内部 krb5 idp 插件作为技术预览提供

SSSD **krb5 idp** 插件允许您使用 OAuth2 协议对外部身份提供者(IdP)进行身份验证。此功能仅适用于 RHEL 9.1 及更高版本上的 IdM 服务器。

[Bugzilla:2056482](#)

## RHEL IdM 允许将用户身份验证委派给外部身份提供程序作为技术预览

在 RHEL IdM 中，您可以把用户与支持 OAuth 2 设备授权流的外部身份提供程序(IdP)关联。当这些用户使用 RHEL 9.1 或更高版本中的 SSSD 版本进行身份验证时，它们会在执行身份验证和在外部 IdP 授权后接收到带有 Kerberos 票据的 RHEL IdM 单点登录功能。

主要特性包括：

- 使用 **ipa idp-\*** 命令为外部 IdP 添加、修改和删除引用
- 使用 **ipa user-mod --user-auth-type=idp** 命令为用户启用 IdP 验证

如需更多信息，请参阅[使用外部身份提供程序向 IdM 进行身份验证](#)。

[Bugzilla:2069202](#)

## ACME 支持将自动删除过期的证书作为一个技术预览

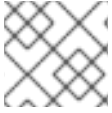
身份管理(IdM)中的自动证书管理环境(ACME)服务添加了一个自动机制，来作为一个技术预览清除证书颁发机构(CA)中的过期证书。现在，ACAC 可以在指定间隔自动删除过期的证书。默认情况下，删除过期的证书被禁用。要启用它，请输入：

有了这个增强，ACAC 可以在指定的间隔自动删除过期的证书。

默认情况下，删除过期的证书被禁用。要启用它，请输入：

```
# ipa-acme-manage pruning --enable --cron "0 0 1 * **"
```

这会在每月第一天的物业删除过期的证书。



### 注意

过期的证书会在保留周期后被删除。默认情况下，这是过期后 30 天。

详情请查看 [ipa-acme-manage \(1\)](#) 手册页。

[Bugzilla:2162677](#)

## 9.10. 桌面

### GNOME 用于 64 位 ARM 架构，作为一个技术预览

GNOME 桌面环境可用于 64 位 ARM 架构，作为技术预览。

现在，您可以使用 VNC 连接到 64 位 ARM 服务器上的桌面会话。因此，您可以使用图形应用程序管理服务。

64 位 ARM 提供了有限的图形应用程序集合。例如：

- Firefox Web 浏览器
- Red Hat 订阅管理器 (**subscription-manager-cockpit**)
- 防火墙配置(**firewall-config**)
- 磁盘用量分析器(**baobab**)

使用 Firefox，您可以连接到服务器上的 Cockpit 服务。

某些应用程序，如 LibreOffice，只提供命令行界面，其图形界面被禁用。

[Jira:RHELPLAN-27394](#)

### 用于 IBM Z 架构的 GNOME 作为技术预览提供

对于 IBM Z 架构，GNOME 桌面环境作为技术预览。

现在，您可以使用 VNC 连接到 IBM Z 服务器上的桌面会话。因此，您可以使用图形应用程序管理服务。

IBM Z 上提供了一组有限的图形应用程序。例如：

- Firefox Web 浏览器
- Red Hat 订阅管理器 (**subscription-manager-cockpit**)
- 防火墙配置(**firewall-config**)

- 磁盘用量分析器(**baobab**)

使用 Firefox，您可以连接到服务器上的 Cockpit 服务。

某些应用程序，如 LibreOffice，只提供命令行界面，其图形界面被禁用。

Jira:RHELPLAN-27737

## 9.11. 图形基础结构

### Intel Arc A 系列图形作为技术预览提供

Intel Arc A 系列图形（也称为 Alchemist 或 DG2）现在作为技术预览提供。

要启用带有 Intel Arc A 系列图形的硬件加速，请在内核命令行上添加以下选项：

```
i915.force_probe=pci-id
```

在这个选项中，将 ***pci-id*** 替换为以下之一：

- Intel GPU 的 PCI ID。
- \* 字符，以启用具有所有 alpha-quality 硬件的 i915 驱动程序

Bugzilla:2041690

## 9.12. WEB 控制台

### Stratis 在 RHEL web 控制台中作为技术预览提供

有了这个更新，Red Hat Enterprise Linux web 控制台提供了将 Stratis 存储作为技术预览管理的能力。

要了解更多有关 Stratis 的信息，请参阅 [什么是 Stratis](#)。

Jira:RHELPLAN-122345

## 9.13. 虚拟化

### 创建嵌套虚拟机

对于运行在 Intel、AMD64 和 IBM Z 主机上的 RHEL 9 KVM 虚拟机，嵌套的 KVM 虚拟化作为技术预览提供。有了这个功能，运行在物理 RHEL 9 主机上的 RHEL 7、RHEL 8 或 RHEL 9 虚拟机可以充当 hypervisor，并托管自己的虚拟机。

Jira:RHELDPCS-17040

### 虚拟机的 Intel SGX 作为技术预览提供

作为技术预览，现在可以为 RHEL 9 上托管的虚拟机(VM)配置 Intel Software Guard Extensions (SGX)。SGX 有助于保护 Intel 硬件上特定进程的数据完整性和保密性。在主机上设置 SGX 后，该功能被传递给其虚拟机，以便客户机操作系统(OS)可以使用它。

请注意，对于要使用 SGX 的客户机操作系统，您必须首先为特定的操作系统安装 SGX 驱动程序。另外，主机上的 SGX 无法对虚拟机进行内存加密。

Jira:RHELPLAN-69761

## 用于 KVM 虚拟机的 AMD SEV 和 SEV-ES

作为技术预览，RHEL 9 为使用 KVM 管理程序的 AMD EPYC 主机提供安全加密虚拟化(SEV)功能。如果在虚拟机(VM)上启用，SEV 会加密虚拟机的内存来保护虚拟机被主机访问。这提高了虚拟机的安全性。

另外，增强的 Encrypted State 版本 SEV-ES) 也作为技术预览提供。SEV-ES 在虚拟机停止运行时加密所有 CPU 注册内容。这可防止主机修改虚拟机的 CPU 注册或读取它们中的任何信息。

请注意，SEV 和 SEV-ES 仅适用于第 2 代 AMD EPYC CPU (代号 Rome) 或更新版本。另请注意，RHEL 9 包括 SEV 和 SEV-ES 加密，但不包括 SEV 和 SEV-ES 安全测试。

Jira:RHELPLAN-65217

## 虚拟化现在在 ARM 64 上可用

作为技术预览，现在可以使用 ARM 64 CPU 在系统中创建 KVM 虚拟机。

Jira:RHELPLAN-103993

## virtio-mem 现在包括在 AMD64、Intel 64 和 ARM 64 中

作为技术预览，RHEL 9 在 AMD64、Intel 64 和 ARM 64 系统中引入了 **virtio-mem** 功能。使用 **virtio-mem** 可让虚拟机(VM)动态添加或删除主机内存。

要使用 **virtio-mem**，请在虚拟机 XML 配置中定义 **virtio-mem** 内存设备，并使用 **virsh update-memory-device** 命令请求 VM 运行期间内存设备大小更改。要查看此类内存设备向正在运行的虚拟机公开的当前内存大小，请查看虚拟机的 XML 配置。

[Bugzilla:2014487](#), [Bugzilla:2044172](#), [Bugzilla:2044162](#)

## RHEL 客户机中的 Intel TDX

作为技术预览，Intel Trust Domain Extension (TDX)功能现在可以在 RHEL 9.2 客户机操作系统中使用。如果主机系统支持 TDX，您可以部署硬件隔离的 RHEL 9 虚拟机(VM)，称为信任域(TD)。但请注意，TDX 目前无法与 **kdump** 一起工作，启用 TDX 会导致 **kdump** 在虚拟机上失败。

Bugzilla:1955275

## RHEL 的统一内核镜像现在作为技术预览提供

作为技术预览，您现在可以获取 RHEL 内核作为虚拟机(VM)的统一内核镜像(UKI)。统一内核镜像将 kernel、initramfs 和内核命令行合并成一个签名的二进制文件。

UKI 可用于虚拟和云环境中，特别是在需要强大的 SecureBoot 功能的机密虚拟机中。UKI 作为 RHEL 9 存储库中的 **kernel-uki-virt** 软件包提供。

目前，RHEL UKI 只能在 UEFI 引导配置中使用。

Bugzilla:2142102

## Intel vGPU 作为技术预览提供

作为技术预览，可以将物理 Intel GPU 设备划分为多个虚拟设备，称为 **介质设备**。然后可将这些介质设备分配给多个虚拟机(VM)作为虚拟 GPU。因此，这些虚拟机共享单个物理 Intel GPU 的性能。

请注意，这个功能已弃用，并将在以后的 RHEL 发行版本中完全删除。

Jira:RHELDPCS-17050

## 9.14. 云环境中的 RHEL

### RHEL 现在在 Azure 机密虚拟机上作为技术预览提供

使用更新的 RHEL 内核，您现在可以在 Microsoft Azure 上作为技术预览创建并运行 RHEL 机密虚拟机 (VM)。新添加的统一内核镜像(UKI)现在在 Azure 上可以引导加密的机密虚拟机镜像。UKI 作为 RHEL 9 存储库中的 **kernel-uki-virt** 软件包提供。

目前，RHEL UKI 只能在 UEFI 引导配置中使用。

Jira:RHELPLAN-139800

## 9.15. 容器

### Podman 中的 Quadlet 现在作为技术预览提供

从 Podman v4.4 开始，您可以作为技术预览，使用 Quadlet 从容器描述中自动生成 **systemd** 服务文件。容器描述采用 **systemd** 单元文件格式。该描述侧重于相关的容器详情，隐藏了在 **systemd** 下运行容器的技术复杂性。与 **systemd** 单元文件相比，Qdlet 更易于编写和维护。

如需了解更多详细信息，请参阅 [上游文档](#)，以及 [使 systemd 更好地用于带有 Quadlet 的 Podman](#)。

Jira:RHELPLAN-148394

### 具有 Fulcio 和 Rekor 的 sigstore 签名的客户端现在作为技术预览提供

有了 Fulcio 和 Rekor 服务器，您现在可以根据 OpenID Connect (OIDC) 服务器身份验证使用短期证书来创建签名，而不是手动管理私钥。具有 Fulcio 和 Rekor 的 sigstore 签名的客户端现在作为技术预览提供。这个添加的功能只是客户端侧支持，不包括 Fulcio 或 Rekor 服务器。

在 **policy.json** 文件中添加 **fulcio** 部分。要签署容器镜像，请使用 **podman push --sign-by-sigstore=file.yml** 或 **skopeo copy --sign-by-sigstore=file.yml** 命令，其中 **file.yml** 是 sigstore 签名参数文件。

要验证签名，请在 **policy.json** 文件中添加 **fulcio** 部分和 **rekorPublicKeyPath** 或 **rekorPublicKeyData** 字段。如需更多信息，请参阅 **containers-policy.json** 手册页。

Jira:RHELPLAN-136611

### podman-machine 命令不受支持

用于管理虚拟机的 **podman-machine** 命令仅作为技术预览提供。相反，请直接从命令行运行 Podman。

Jira:RHELDPCS-16861

## 第 10 章 过时的功能

这部分提供在 Red Hat Enterprise Linux 9 中弃用的功能概述。

弃用的功能可能在以后的主要发行本中不被支持，因此不建议在新的部署中使用。有关特定主要发行本中已弃用功能的最新列表，请参考最新版本的发行文档。

在 Red Hat Enterprise Linux 9 中，已弃用的功能的支持状态不会改变。有关支持长度的详情，请查看 [Red Hat Enterprise Linux 生命周期](#) 和 [Red Hat Enterprise Linux 应用程序流生命周期](#)。

对于当前或将来的主发行版本中的新部署，我们不推荐使用已弃用的硬件组件。硬件驱动程序更新仅限于安全和关键修复。红帽建议尽快替换这个硬件。

一个软件包可以被弃用，我们不推荐在以后使用。在某些情况下，软件包可从产品中删除。然后，产品文档可识别提供类似、完全相同或者更高级功能的最新软件包，并提供进一步建议。

有关 RHEL 8 中存在但已在 RHEL 9 中删除的功能的详情，请参阅 [采用 RHEL 9 时的注意事项](#)。

### 10.1. 安装程序和镜像创建

#### 弃用的 Kickstart 命令

以下 Kickstart 命令已弃用：

- **timezone --ntpservers**
- **timezone --nntp**
- **logging --level**
- **%packages --excludeWeakdeps**
- **%packages --instLangs**
- **%anaconda**
- **pwpolicy**

请注意，当只列出具体选项时，基础命令及其其它选项仍可用且未被弃用。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。

Bugzilla:1899167

#### edge-commit 和 edge-container 蓝图中的用户和组自定义已弃用

在蓝图中指定用户或组自定义对于 **edge-commit** 和 **edge-container** 镜像类型已弃用，因为在升级镜像时用户自定义会消失，且不能在蓝图中再次指定用户。因此，您应该在蓝图中为用来部署现有 OSTree 提交的边缘镜像类型直接指定用户和组，如 **edge-raw-image**、**edge-installer**、**edge-simplified-installer**。

请注意，在蓝图中指定用户或组自定义仍被支持，但支持最终会移除。

Bugzilla:2173928

### 10.2. 订阅管理



## subscription-manager 命令的 --token 选项已弃用

**subscription-manager register** 命令的 **--token=<TOKEN>** 选项是一种身份验证方法，可帮助将您的系统注册到红帽。这个选项取决于授权服务器提供的功能。默认授权服务器

**subscription.rhsm.redhat.com** 计划关闭此功能。因此，尝试使用 **subscription-manager register --token=<TOKEN>** 可能会失败，并显示以下错误消息：

```
Token authentication not supported by the entitlement server
```

您可以使用其他授权方法继续注册您的系统，例如包括 **subscription-manager register** 命令的成对的选项 **--username / --password** 和 **--org / --activationkey**。

[Bugzilla:2163716](#)

## 10.3. SHELL 和命令行工具

### 在 ReaR 配置文件中设置 TMPDIR 变量已弃用

使用诸如 **export TMPDIR=...** 的语句在 **/etc/rear/local.conf** 或 **/etc/rear/site.conf** ReaR 配置文件中设置 **TMPDIR** 环境变量无法工作，且已弃用。

要为 ReaR 临时文件指定一个自定义目录，请在执行 ReaR 前在 shell 环境中导出变量。例如，执行 **export TMPDIR=...** 语句，然后在同一 shell 会话或脚本中执行 **rear** 命令。

[Jira:RHELDPCS-18049](#)

### dump 软件包中的 dump 工具已弃用

用于文件系统备份的 **dump** 工具已弃用，在 RHEL 9 中将不再提供。

在 RHEL 9 中，红帽建议根据使用情况使用 **tar**、**dd** 或 **bacula**、备份工具，对 ext2、ext3 和 ext4 文件系统提供了完整和安全的备份。

请注意，**dump** 软件包中的 **restore** 工具仍可用，在 RHEL 9 中也被支持，并作为 **restore** 软件包提供。

[Bugzilla:1997366](#)

### Bacula 中的 SQLite 数据库后端已被弃用

Bacula 备份系统支持多个数据库后端：PostgreSQL、MySQL 和 SQLite。SQLite 后端已被弃用，并将在以后的 RHEL 版本中不被支持。作为一种替代，迁移到其他一种后端(PostgreSQL 或 MySQL)，且在新部署中不使用 SQLite 后端。

[Bugzilla:2089395](#)

## 10.4. 安全性

### 对于加密目的，SHA-1 已被弃用

使用 SHA-1 消息摘要用于加密目的在 RHEL 9 中已被弃用。SHA-1 生成的摘要不被视为是安全的，因为已发现多个基于哈希进行的安全攻击。RHEL 核心加密组件不再默认使用 SHA-1 创建签名。RHEL 9 中的应用程序已更新，以避免在与安全相关的用例中使用 SHA-1。

其中一个例外是，仍然可以使用 SHA-1 创建 HMAC-SHA1 消息验证代码和 Universal Unique Identifier(UUID)值，因为这些用例目前不存在安全风险。另外，为了保持一些重要的互操作性和兼容性，SHA-1 还会在一些有限的情况下使用，例如 Kerberos 和 WPA-2。详情请查看 [RHEL 9 安全强化文档中的使用与 FIPS 140-3 不兼容的加密系统的 RHEL 应用程序列表](#)。



如果您需要使用 SHA-1 来验证现有或第三方加密签名，您可以输入以下命令启用它：

```
# update-crypto-policies --set DEFAULT:SHA1
```

或者，您可以将系统范围的加密策略切换到 **LEGACY** 策略。请注意，**LEGACY** 也启用了其他一些不安全的算法。

Jira:RHELPLAN-110763

### fapolicyd.rules 已被弃用

包含允许和拒绝执行规则的文件的 `/etc/fapolicyd/rules.d/` 目录替代了 `/etc/fapolicyd/fapolicyd.rules` 文件。`fagenrules` 脚本现在将此目录中的所有组件规则文件合并到 `/etc/fapolicyd/compiled.rules` 文件。`/etc/fapolicyd/fapolicyd` 中的规则仍由 `fapolicyd` 框架处理，但只是为了保证向后兼容。

[Bugzilla:2054740](#)

### 在 RHEL 9 中弃用 SCP

安全复制协议(SCP)已弃用，因为它有已知的安全漏洞。SCP API 仍可用于 RHEL 9 生命周期，但使用它可以降低系统安全性。

- 在 `scp` 实用程序中，默认情况下，SCP 被 SSH 文件传输协议(SFTP)替代。
- OpenSSH 套件在 RHEL 9 中不使用 SCP。
- SCP 在 `libssh` 库中已弃用。

Jira:RHELPLAN-99136

### SASL 中的 digest-MD5 已被弃用

Simple Authentication Security Layer(SASL)框架中的 Digest-MD5 身份验证机制已弃用，并可能在以后的主发行版本中从 `cyrus-sasl` 软件包中删除。

[Bugzilla:1995600](#)

### OpenSSL 弃用了 MD2, MD4, MDC2, Whirlpool, Blowfish, CAST, DES, IDEA, RC2, RC4, RC5, SEED, 和 PBKDF1

OpenSSL 项目已弃用了一组加密算法，因为它们不安全，不常用，或两者都不安全。红帽还不建议使用这些算法，RHEL 9 则为其提供迁移加密数据以使用新的算法。对于系统的安全性，用户不得依赖于这些算法。

以下算法的实现已移到 OpenSSL 中的旧提供者：MD2、MD4、MD4、MDC2、Mlpool、Blowfish、CAST、DES、IDEA、RC2、RC4、RC5、SEED 和 PBKDF1。

有关如何载入旧供应商的说明，请参阅 `/etc/pki/tls/openssl.cnf` 配置文件，并启用对已弃用算法的支持。

[Bugzilla:1975836](#)

### /etc/system-fips 现已弃用

支持通过 `/etc/system-fips` 文件指定 FIPS 模式，该文件将不会包含在将来的 RHEL 版本中。要在 FIPS 模式中安装 RHEL，请在系统安装过程中将 `fips=1` 参数添加到内核命令行。您可以使用 `fips-mode-setup --check` 命令检查 RHEL 是否以 FIPS 模式运行。

Jira:RHELPLAN-103232

## libcrypt.so.1 现已弃用

**libcrypt.so.1** 库现已弃用，它可能会在以后的 RHEL 版本中删除。

[Bugzilla:2034569](#)

## OpenSSL 需要在 FIPS 模式下对 RSA 加密进行填充

OpenSSL 在 FIPS 模式下不再支持没有填充的 RSA 加密。没有填充的 RSA 加密不常见，很少使用。请注意，带有 RSA (RSASVE) 的密钥封装不使用填充，但仍支持。

[Bugzilla:2168665](#)

## 10.5. 网络

### RHEL 9 中已弃用网络团队 (Network teams)

**teamd** 服务和 **libteam** 库在 Red Hat Enterprise Linux 9 中已弃用，并将在下一个主发行版本中删除。作为替换，配置绑定而不是网络组。

红帽注重于基于内核的绑定操作，以避免维护具有类似功能的两个功能：绑定和团队 (team)。绑定代码具有较高的客户采用率，非常可靠，具有活跃的社区开发。因此，绑定代码会收到功能增强和更新。

有关如何将团队迁移到绑定的详情，请参阅[将网络组配置迁移到网络绑定](#)。

[Bugzilla:1935544](#)

### ifcfg 格式的 NetworkManager 连接配置文件已弃用

在 RHEL 9.0 及更高版本中，**ifcfg** 格式的连接配置文件已弃用。下一个主要 RHEL 发行版本将删除对这个格式的支持。但是，在 RHEL 9 中，如果修改了配置文件，NetworkManager 仍然会使用这个格式处理和更新现有的配置文件。

默认情况下，NetworkManager 现在在 **/etc/NetworkManager/system-connections/** 目录中以 keyfile 格式存储连接配置文件。与 **ifcfg** 格式不同，keyfile 格式支持 NetworkManager 提供的所有连接设置。有关 keyfile 格式以及如何迁移配置集的详情，请参考[keyfile 格式的 NetworkManager 连接配置文件](#)。

[Bugzilla:1894877](#)

### firewalld 中的 iptables 后端已弃用

在 RHEL 9 中，**iptables** 框架已弃用。因此，**firewalld** 中的 **iptables** 后端和 **direct interface** 也被弃用。您可以使用 **firewalld** 中的原生功能，而不是 **direct interface** 来配置所需的规则。

[Bugzilla:2089200](#)

## 10.6. 内核

### 在 RHEL 9 中弃用 ATM 封装

异步传输模式(ATM)封装为 ATM Adaptation Layer 5(AAL-5)提供第 2 层 (Point-to-Point 协议、以太网) 或第 3 层 (IP) 连接。从 RHEL 7 开始，红帽尚未为 ATM NIC 驱动程序提供支持。RHEL 9 中丢弃对 ATM 实施的支持。这些协议目前仅在芯片组中使用，该协议支持 ADSL 技术，并由制造商逐步淘汰。因此，Red Hat Enterprise Linux 9 中已弃用 ATM 封装。

如需更多信息，请参阅 [PPP Over AAL5, Multiprotocol Encapsulation over ATM Adaptation Layer 5](#)，和 [Classical IP and ARP over ATM](#)。

[Bugzilla:2058153](#)

### kexec-tools 的 kexec\_load 系统调用已弃用

在以后的 RHEL 版本中将不支持 **kexec\_load** 系统调用（其载入第二个内核）。**kexec\_file\_load** 系统调用替换了 **kexec\_load**，它现在是在所有架构上的默认系统调用。

[Bugzilla:2113873](#)

### RHEL 9 中已弃用网络团队（Network teams）

**teamd** 服务和 **libteam** 库在 Red Hat Enterprise Linux 9 中已弃用，并将在下一个主发行版本中删除。作为替换，配置绑定而不是网络组。

红帽注重于基于内核的绑定操作，以避免维护具有类似功能的两个功能：绑定和团队（team）。绑定代码具有较高的客户采用率，非常可靠，具有活跃的社区开发。因此，绑定代码会收到功能增强和更新。

有关如何将团队迁移到绑定的详情，请参阅将[网络组配置迁移到网络绑定](#)。

[Bugzilla:2013884](#)

## 10.7. 文件系统和存储

### lvm2-activation-generator 及其生成的服务在 RHEL 9.0 中删除

**lvm2-activation-generator** 程序及其生成的服务 **lvm2-activation**、**lvm2-activation-early**、**lvm2-activation-net** 已在 RHEL 9.0 中删除。**lvm.conf event\_activation** 设置用于激活服务将不再起作用。自动激活卷组的唯一方法是基于事件激活。

[Bugzilla:2038183](#)

## 10.8. 动态编程语言、网页和数据库服务器

### libdb 已被弃用

RHEL 8 和 RHEL 9 目前提供 Berkeley DB(**libdb**)版本 5.3.28，该版本根据 LGPLv2 许可证发布。上游 Berkeley DB 版本 6 在 AGPLv3 许可证下提供，该许可证更严格。

从 RHEL 9 开始，**libdb** 软件包已弃用，可能不会在以后的 RHEL 版本中可用。

另外，在 RHEL 9 中，加密算法已从 **libdb** 中删除，从 RHEL 9 中删除了多个 **libdb** 依赖项。

建议 **libdb** 用户迁移到其他键值数据库。如需更多信息，请参阅 [RHEL 中已弃用的 Berkeley DB\(libdb\)](#) 的[知识库文章](#)。

[Bugzilla:1927780](#), [Jira:RHELPLAN-80695](#), [Bugzilla:1974657](#)

## 10.9. 编译器和开发工具

### openssl 3.0 弃用了比 2048 小的密钥

**openssl** 3.0 弃用了小于 2048 位的密钥，在 Go 的 FIPS 模式中无法正常工作。

[Bugzilla:2111072](#)

有些 PKCS1 v1.5 模式现已弃用

一些 **PKCS1** v1.5 模式在 **FIPS-140-3** 中未被批准用于加密，并被禁用。它们将不再在 Go 的 FIPS 模式下工作。

Bugzilla:2092016

## 10.10. 身份管理

### OpenDNSSec 中的 SHA-1 现已弃用

OpenDNSSEC 支持使用 **SHA-1** 算法导出数字签名和身份验证记录。不再支持使用 **SHA-1** 算法。在 RHEL 9 发行版本中，OpenDNSSec 中的 **SHA-1** 已被弃用，并可能在以后的次版本中删除。另外，OpenDNSSec 支持仅限于与红帽身份管理的集成。OpenDNSSEC 不支持独立。

Bugzilla:1979521

### SSSD 隐式文件供应商域默认禁用

SSSD 隐式 **文件** 供应商域，从 **/etc/shadow** 和 **/etc/** groups 等本地文件检索用户信息，现已默认禁用。

使用 SSSD 从本地文件检索用户和组信息：

1. 配置 SSSD.选择以下选项之一：
  - a. 使用 **sssd.conf** 配置文件中的 **id\_provider=files** 选项明确配置本地域。

```
[domain/local]
id_provider=files
...
```

- b. 通过在 **sssd.conf** 配置文件中设置 **enable\_files\_domain=true** 来启用 **文件** 供应商。

```
[sssd]
enable_files_domain = true
```

2. 配置名称服务切换。

```
# authselect enable-feature with-files-provider
```

Jira:RHELPLAN-100639

### -h 和 -p 选项在 OpenLDAP 客户端工具中被弃用。

上游 OpenLDAP 项目已在其工具中弃用了 **-h** 和 **-p** 选项，建议使用 **-H** 选项来指定 LDAP URI。因此，RHEL 9 已在所有 OpenLDAP 客户端工具中都弃用了这两个选项。**-h** 和 **-p** 选项将在以后的版本中从 RHEL 产品中删除。

Jira:RHELPLAN-137660

### SSSD 文件提供者已弃用

SSSD **文件** 提供者已在 Red Hat Enterprise Linux (RHEL) 9 中弃用。**文件** 提供者可能会从以后的版本中删除。

Jira:RHELPLAN-139805

### nsslapd-idlistscanlimit 参数已弃用，其默认值已更改

使用新的过滤器重新排序优化时，`nsslapd-idlistscanlimit` 属性对搜索性能的影响弊大于利。因此，属性已弃用。另外，默认值已改为 **2147483646**（无限）。

[Bugzilla:1952241](#)

### SMB1 协议在 Samba 中已弃用

从 Samba 4.11 开始，不安全的服务器消息块版本 1 (SMB1) 协议已弃用，并将在以后的发行版本中删除。

为提高安全性，在 Samba 服务器和客户端工具中默认禁用 SMB1。

[Jira:RHELDPCS-16612](#)

## 10.11. 桌面

### GTK 2 现已弃用

旧的 GTK 2 工具包及以下相关软件包已弃用：

- **adwaita-gtk2-theme**
- **gnome-common**
- **gtk2**
- **gtk2-immodules**
- **hexchat**

其它几个软件包目前依赖于 GTK 2。这些已被修改，以便它们不再依赖于未来的主 RHEL 发行版本中已弃用的软件包。

如果您维护使用 GTK 2 的应用程序，红帽建议您将应用移植到 GTK 4。

[Jira:RHELPLAN-131882](#)

### libreoffice 已被弃用

LibreOffice RPM 软件包现已弃用，并将在以后的主 RHEL 发行版本中删除。LibreOffice 在 RHEL 7、8 和 9 的整个生命周期中仍然被完全支持。

作为 RPM 软件包的替代，红帽建议您从 Document Foundation 提供的以下源中安装 LibreOffice：

- Flathub 存储库中的官方 Flatpak 软件包：<https://flathub.org/apps/org.libreoffice.LibreOffice>。
- 官方 RPM 软件包：<https://www.libreoffice.org/download/download-libreoffice/>

[Jira:RHELDPCS-16300](#)

## 10.12. 图形基础结构

### Motif 已被弃用

Motif 小部件工具包已在 RHEL 中被弃用，因为上游 Motif 社区的开发不活跃。

以下 Motif 软件包已被弃用，包括其开发和调试变体：

- **motif**
- **openmotif**
- **openmotif21**
- **openmotif22**

另外，**motif-static** 软件包已删除。

红帽建议使用 GTK 工具包作为替代品。与 Motif 相比，GTK 更易于维护，并提供了新功能。

Jira:RHELPLAN-98983

## 10.13. RED HAT ENTERPRISE LINUX 系统角色

在 RHEL 9 节点上配置 **team** 时，**network** 系统角色显示一条弃用警告

网络 **team** 功能在 RHEL 9 中已弃用。因此，在 RHEL 8 控制节点上使用 **network** RHEL 系统角色在 RHEL 9 节点上配置网络 **team**，显示一条有关弃用的警告。

[Bugzilla:1999770](#)

## 10.14. 虚拟化

使用基于 SHA1 的签名进行 SecureBoot 镜像验证已弃用

在 UEFI (PE/COFF) 可执行文件中使用基于 SHA1 的签名执行 SecureBoot 镜像验证已过时。反之，红帽建议使用基于 SHA2 算法或更新版本的签名。

[Bugzilla:1935497](#)

对虚拟机快照的支持有限

目前只对使用 UEFI 固件的虚拟机支持创建虚拟机(VM)的快照。另外，在快照操作过程中，QEMU 监控可能会被阻止，这会影响某些工作负载的 hypervisor 性能。

另请注意，创建虚拟机快照的当前机制已被弃用，红帽不推荐在生产环境中使用虚拟机快照。但是，一个新的虚拟机快照机制正在开发中，计划在以后的 RHEL 9 次要发行本中完全实施。

Jira:RHELPLAN-15509, [Bugzilla:1621944](#)

虚拟软盘驱动程序已弃用

用于控制虚拟软盘设备的 **isa-fdc** 驱动程序现已弃用，并将在以后的 RHEL 发行版本中不被支持。因此，为了确保与迁移的虚拟机(VM)兼容，红帽不建议在 RHEL 9 上托管的虚拟机中使用软盘磁盘设备。

[Bugzilla:1965079](#)

qcow2-v2 镜像格式已弃用

在 RHEL 9 中，虚拟磁盘镜像的 qcow2-v2 格式已弃用，并将在以后的 RHEL 主发行版本中不被支持。另外，RHEL 9 Image Builder 无法以 qcow2-v2 格式创建磁盘镜像。

红帽强烈建议您使用 qcow2-v3，而不是 qcow2-v2。要将 qcow2-v2 镜像转换为更新的格式版本，请使用 **qemu-img amend** 命令。

[Bugzilla:1951814](#)

## virt-manager 已被弃用

虚拟机管理器（也称 **virt-manager**）已弃用。RHEL web 控制台（也称为 **Cockpit**）旨在在以后的版本中成为它的替代。因此，建议您使用 web 控制台使用 GUI 管理虚拟化。但请注意，在 RHEL web 控制台中，**virt-manager** 中的一些功能可能还不可用。

Jira:RHELPLAN-10304

## libvirtd 已被弃用

单体 **libvirt** 守护进程 **libvirtd** 已在 RHEL 9 中弃用，并将在以后的 RHEL 主发行版本中删除。请注意，您仍然可以使用 **libvirtd** 在虚拟机监控程序上管理虚拟化，但红帽建议您切换到新引入的模块化 **libvirt** 守护进程。具体说明和详情，请参阅 [RHEL 9 配置和管理虚拟化](#) 文档。

Jira:RHELPLAN-113995

## 旧的 CPU 型号现已弃用

大量 CPU 模型已被弃用，并将在以后的 RHEL 主发行版本中的虚拟机 (VM) 不被支持。弃用的模型如下：

- 对于 Intel：Intel Xeon 55xx 和 75xx Processor 系列前的模型（也称为 Nehalem）
- AMD：AMD Opteron G4 之前的型号
- 对于 IBM Z：IBM z14 之前的型号

要检查您的虚拟机是否使用已弃用的 CPU 模型，请使用 **virsh dominfo** 工具，并在 **Messages** 部分查找类似如下的行：

```
tainted: use of deprecated configuration settings
deprecated configuration: CPU model 'i486'
```

[Bugzilla:2060839](#)

## 基于 RDMA 的实时迁移已弃用

有了这个更新，使用 Remote Direct Memory Access (RDMA) 迁移正在运行的虚拟机已被弃用。因此，仍可以使用 **rdma://** 迁移 URI 来通过 RDMA 请求迁移，但这个功能将在以后的 RHEL 主发行版本中不被支持。

Jira:RHELPLAN-153267

## 10.15. 容器

### 不支持在 RHEL 7 主机上运行 RHEL 9 容器

不支持在 RHEL 7 主机上运行 RHEL 9 容器。它可能可以正常工作，但却没有保证。

如需更多信息，请参阅 [Red Hat Enterprise Linux Container Compatibility Matrix](#)。

Jira:RHELPLAN-100087

### Podman 中的 SHA1 哈希算法已弃用

Podman 不再支持用来生成无根网络命名空间的文件名的 SHA1 算法。因此，如果在使用 Podman 4.1.1 或更高版本之前启动无根容器，则必须重启它们（而不只是使用 **slirp4netns**），以确保它们可以在升级后启动容器。



Bugzilla:2069279

## rhel9/pause 已被弃用

rhel9/pause 容器镜像已被弃用。

[Bugzilla:2106816](#)

## CNI 网络堆栈已弃用

Container Network Interface (CNI)网络堆栈将在以后的次版本中被弃用。在以前的版本中，容器只能通过 DNS 连接到单个 Container Network Interface (CNI)插件。podman v.4.0 引入了一个新的 Netavark 网络堆栈。您可以将 Netavark 网络堆栈与 Podman 和其他 Open Container Initiative(OCI)容器管理应用程序一起使用。Podman 的 Netavark 网络堆栈也与高级 Docker 功能兼容。多个网络中的容器可以访问任何这些网络上的容器。

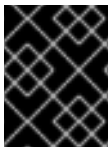
如需更多信息，请参阅 [将网络堆栈从 CNI 切换到 Netavark](#)。

Jira:RHELPLAN-147725

## 10.16. 已弃用的软件包

本节列出了已弃用的软件包，可能不会包括在 Red Hat Enterprise Linux 未来的主发行版本中。

有关 RHEL 8 和 RHEL 9 之间软件包的更改，请参阅 [使用 RHEL 9 文档中的软件包的更改](#)。



### 重要

在 RHEL 9 中，已弃用的软件包的支持状态保持不变。有关支持长度的更多信息，请参阅 [Red Hat Enterprise Linux 生命周期](#) 和 [Red Hat Enterprise Linux 应用程序流生命周期](#)。

RHEL 9 中已弃用了以下软件包：

- iptables-devel
- iptables-libs
- iptables-nft
- iptables-nft-services
- iptables-utils
- libdb
- mcpp
- mod\_auth\_mellon
- motif
- motif-devel
- python3-pytz
- xorg-x11-server-Xorg



## 第 11 章 已知问题

这部分论述了 Red Hat Enterprise Linux 9.2 中的已知问题。

### 11.1. 安装程序和镜像创建

#### auth 和 authconfig Kickstart 命令需要 AppStream 软件仓库

**auth** 和 **authconfig** Kickstart 命令在安装过程中需要 **authselect-compat** 软件包。如果没有这个软件包，如果使用了 **auth** 或 **authconfig**，则安装会失败。但根据设计，**authselect-compat** 软件包只包括在 AppStream 仓库中。

要临时解决这个问题，请确定安装程序可使用 BaseOS 和 AppStream 软件仓库，或者在安装过程中使用 **authselect** Kickstart 命令。

Bugzilla:1640697

#### reboot --kexec 和 inst.kexec 命令不提供可预测的系统状态

使用 **reboot --kexec** Kickstart 命令或 **inst.kexec** 内核引导参数执行 RHEL 安装不会提供与完全重启相同的可预期系统状态。因此，在不重启的情况下切换安装的系统可能会导致无法预计的结果。

请注意，**kexec** 功能已弃用，并将在以后的 Red Hat Enterprise Linux 版本中删除。

Bugzilla:1697896

#### 在 Anaconda 作为应用程序运行的系统中意外 SELinux 策略

当 Anaconda 作为应用程序运行在已安装的系统上（例如，使用 **-image anaconda** 选项对镜像文件执行另一次安装）时，不禁止系统在安装过程中修改 SELinux 类型和属性。因此，某些 SELinux 策略的元素可能会在运行 Anaconda 的系统上发生更改。要临时解决这个问题，请不要在生产环境系统上运行 Anaconda，而在临时虚拟机中执行它。因此，生产系统上的 SELinux 策略没有被修改。作为系统安装过程的一部分运行 anaconda，如从 **boot.iso** 或 **dvd.iso** 安装不会受此问题的影响。

Bugzilla:2050140

#### 当使用使用第三方工具创建的 USB 引导安装时，不会检测本地介质安装源

当从使用第三方工具创建的 USB 引导 RHEL 安装时，安装程序无法检测 **Local Media** 安装源（只检测到 *Red Hat CDN*）。

出现这个问题的原因是，默认的引导选项 **inst.stage2=** 会尝试搜索 **iso9660** 镜像格式。但是，第三方工具可能会创建具有不同格式的 ISO 镜像。

作为临时解决方案，请使用以下解决方案之一：

- 当引导安装时，点击 **Tab** 键来编辑内核命令行，并将引导选项 **inst.stage2=** 改为 **inst.repo=**。
- 要在 Windows 中创建可引导 USB 设备，使用 Fedora Media Writer。
- 使用 Rufus 等第三方工具创建可引导 USB 设备时，首先在 Linux 系统上重新生成 RHEL ISO 镜像，然后使用第三方工具创建可引导 USB 设备。

有关执行任何指定临时解决方案的步骤的更多信息，请参阅[安装 RHEL 8.3 过程中不会自动探测到安装介质](#)

Bugzilla:1877697

## USB CD-ROM 驱动器作为 Anaconda 中的安装源不可用

当源为 USB CD-ROM 驱动器，并且指定了 Kickstart `ignoredisk --only-use=` 命令时，安装会失败。在这种情况下，Anaconda 无法找到并使用这个源磁盘。

要临时解决这个问题，请使用 `harddrive --partition=sdX --dir=/` 命令从 USB CD-ROM 驱动器安装。因此，安装不会失败。

[Bugzilla:1914955](#)

## 驱动程序磁盘菜单无法在控制台上显示用户输入

当您在内核命令行上将 `inst.dd` 选项与驱动程序磁盘一起使用来开始 RHEL 安装时，控制台无法显示用户输入。因此，看上去应用程序不会响应用户输入和冻结，但会显示使用户混淆的输出。但是，此行为不会影响功能，用户输入会在按 **Enter** 后被注册。

作为临时解决方案，要查看预期结果，请忽略控制台中缺少用户输入，并在完成添加输入后按 **Enter** 键。

[Bugzilla:2109231](#)

## 使用 iso9660 文件系统的硬盘分区安装失败

您不能在使用 `iso9660` 文件系统进行分区的系统中安装 RHEL。这是因为将设置为忽略包含 `iso9660` 文件系统分区的硬盘的更新安装代码。即使在没有使用 DVD 的情况下安装 RHEL，也会发生这种情况。

要解决这个问题，请在 kickstart 文件中添加以下脚本，以在安装开始前格式化磁盘。

注：在执行临时解决方案前，请备份磁盘上的数据。`erafs` 命令对磁盘中的所有现有数据进行格式化。

```
%pre
wipefs -a /dev/sda
%end
```

因此，安装可以正常工作，且没有任何错误。

[Bugzilla:1929105](#)

## Anaconda 无法验证管理员用户帐户是否存在

在使用图形用户界面安装 RHEL 时，Anaconda 无法验证管理员帐户是否已创建。因此，用户可以在没有管理员用户帐户的情况下安装系统。

要临时解决这个问题，请确保配置管理员用户帐户或 root 密码已设置，且 root 帐户被解锁。因此，用户可以在安装的系统中执行管理任务。

[Bugzilla:2047713](#)

## 新的 XFS 功能可防止使用比版本 5.10 更早的固件引导 PowerNV IBM POWER 系统

PowerNV IBM POWER 系统使用 Linux 内核进行固件，并使用 Petitboot 作为 GRUB 的替代。这会导致固件内核挂载 `/boot`，Petitboot 读取 GRUB 配置和引导 RHEL。

RHEL 9 内核为 XFS 文件系统引入了 `bigtime=1` 和 `inobtcount=1` 功能，而使用比版本 5.10 旧固件的内核不理解。

要临时解决这个问题，您可以为 `/boot` 使用另一个文件系统，例如 `ext4`。

[Bugzilla:1997832](#)

## RHEL for Edge 安装程序镜像在安装 rpm-ostree 有效负载时无法创建挂载点

当部署 **rpm-ostree** 有效负载时，例如在 RHEL for Edge 安装程序镜像中，安装程序不会为自定义分区正确创建一些挂载点。因此，安装会中止，并报以下错误：

```
The command 'mount --bind /mnt/sysimage/data /mnt/sysroot/data' exited with the code 32.
```

要临时解决这个问题：

- 使用自动分区方案，且不手动添加任何挂载点。
- 只在 **/var** 目录中手动分配挂载点。例如：**/var/my-mount-point** 和以下标准目录：**/**、**/boot**、**/var**。

因此，安装过程成功完成。

[Bugzilla:2125542](#)

## 当连接到网络但没有配置 DHCP 或静态 IP 地址时，NetworkManager 无法在安装后启动

从 RHEL 9.0 开始，当没有设置特定的 **ip=** 或 **kickstart** 网络配置时，Anaconda 会自动激活网络设备。Anaconda 为每个以太网设备创建默认的持久配置文件。连接配置文件的 **ONBOOT** 和 **autoconnect** 值设为 **true**。因此，在启动安装的系统的过程中，RHEL 会激活网络设备，**networkManager-wait-online** 服务会失败。

作为临时解决方案，请执行以下操作之一：

- 使用 **nmcli** 工具删除所有连接，但您要使用的一个连接除外。例如：
  - a. 列出所有连接配置文件：
 

```
# nmcli connection show
```
  - b. 删除您不需要的连接配置文件：
 

```
# nmcli connection delete <connection_name>
```

 将 **<connection\_name>** 替换为您要删除的连接的名称。
- 如果没有设置特定的 **ip=** 或 **kickstart** 网络配置，请在 Anaconda 中禁用自动连接网络功能。
  - a. 在 Anaconda GUI 中，导航到 **Network & Host Name**。
  - b. 选择要禁用的网络设备。
  - c. 单击 **Configure**。
  - d. 在 **General** 选项卡中，取消选择 **Connect automatically with priority**。
  - e. 点 **Save**。

[Bugzilla:2115783](#)

## 无法从安装环境中的驱动程序更新磁盘载入更新的驱动程序

如果已加载了安装初始 ramdisk 中同样的驱动程序，则驱动程序更新磁盘中的驱动程序的新版本可能无法加载。因此，驱动程序的更新版本无法应用到安装环境。

作为临时解决方案，请将 `modprobe.blacklist=` 内核命令行选项与 `inst.dd` 选项一起使用。例如，要确保是否加载了驱动程序更新磁盘中的 `virtio_blk` 驱动程序的更新版本，请使用 `modprobe.blacklist=virtio_blk`，然后继续通常的流程来应用驱动程序更新磁盘中的驱动程序。因此，系统会加载驱动程序的更新版本，并在安装环境中使用它。

[Bugzilla:2164216](#)

## Kickstart 安装无法配置网络连接

Anaconda 只能通过 NetworkManager API 执行 kickstart 网络配置。Anaconda 在 `%pre` kickstart 部分后处理网络配置。因此，Kickstart `%pre` 部分中的一些任务会被阻止。例如，因为网络配置不可用，从 `%pre` 部分中下载软件包会失败。

要临时解决这个问题：

- 配置网络，例如使用 `nmcli` 工具作为 `%pre` 脚本的一部分。
- 使用安装程序引导选项为 `%pre` 脚本配置网络。

因此，可以将网络用于 `%pre` 部分中的任务，Kickstart 安装过程完成。

[Bugzilla:2173992](#)

## 11.2. 软件管理

### 安装过程有时将变为无响应

安装 RHEL 时，安装过程有时会变得无响应。`/tmp/packaging.log` 文件在末尾显示以下消息：

```
10:20:56,416 DDEBUG dnf: RPM transaction over.
```

要解决这个问题，重启安装过程。

[Bugzilla:2073510](#)

## 11.3. SHELL 和命令行工具

### 如果在配置文件中设置了 TMPDIR 变量，则 ReaR 在恢复过程中失败

在 `/etc/rear/local.conf` 或 `/etc/rear/site.conf` ReaR 配置文件中设置和导出 `TMPDIR` 无法工作，且已弃用。

ReaR 默认配置文件 `/usr/share/rear/conf/default.conf` 包含以下说明：

```
# To have a specific working area directory prefix for Relax-and-Recover
# specify in /etc/rear/local.conf something like
#
# export TMPDIR="/prefix/for/rear/working/directory"
#
# where /prefix/for/rear/working/directory must already exist.
# This is useful for example when there is not sufficient free space
# in /tmp or $TMPDIR for the ISO image or even the backup archive.
```

上述说明无法正常工作，因为 `TMPDIR` 变量在救援环境中具有相同的值，如果 `TMPDIR` 变量中指定的目录在救援镜像中不存在，则这是不正确的。

因此，在 `/etc/rear/local.conf` 文件中设置和导出 `TMPDIR` 在救援镜像引导时导致以下错误：

```
mktemp: failed to create file via template '/prefix/for/rear/working/directory/tmp.XXXXXXXXXX': No
such file or directory
cp: missing destination file operand after '/etc/rear/mappings/mac'
Try 'cp --help' for more information.
No network interface mapping is specified in /etc/rear/mappings/mac
```

或者，在运行 `rear recover` 时出现以下错误，并稍后中止：

```
ERROR: Could not create build area
```

要临时解决这个问题，如果您想有一个自定义临时目录，请在执行 ReaR 之前在 shell 环境中导出变量，来为 ReaR 临时文件指定一个自定义目录。例如，执行 `export TMPDIR=...` 语句，然后在同一 shell 会话或脚本中执行 `rear` 命令。因此，在上述配置中，恢复成功。

[Jira:RHEL-24847](#)

### 使用 `ifcfg` 文件重命名网络接口失败

在 RHEL 9 中，默认情况下不会安装 `initscripts` 软件包。因此，使用 `ifcfg` 文件重命名网络接口会失败。要解决这个问题，红帽建议您使用 `udev` 规则或链接文件来重命名接口。详情请查看 [Consistent 网络接口设备命名](#) 和 `systemd.link(5)` man page。

如果您无法使用推荐的解决方案之一，请安装 `initscripts` 软件包。

Bugzilla:2018112

### RHEL 9 中不默认安装 `chkconfig` 软件包

RHEL 9 中不默认安装 `chkconfig` 软件包（更新和查询系统服务运行级别信息）。

要管理服务，请使用 `systemctl` 命令或手动安装 `chkconfig` 软件包。

有关 `systemd` 的更多信息，请参阅[管理 systemd](#)。有关如何使用 `systemctl` 实用程序的步骤，请参阅[使用 systemctl 管理系统服务](#)。

Bugzilla:2053598

### 服务位置协议(SLP)易受到通过 UDP 的攻击

OpenSLP 为本地区域网络中的应用程序提供动态配置机制，如打印机和文件服务器。但是，SLP 会受到通过连接到互联网的系统上的 UDP 的反射性拒绝服务放大攻击。SLP 允许未经身份验证的攻击者注册新服务，而不受由 SLP 实现设置的限制。通过使用 UDP 和欺骗源地址，攻击者可以请求服务列表，在欺骗地址上创建拒绝服务。

要防止外部攻击者访问 SLP 服务，请在不受信任的网络上运行的所有系统上禁用 SLP，比如那些直接连接到互联网的系统。另外，要解决这个问题，请配置防火墙以阻止或过滤 UDP 和 TCP 端口 427 上的流量。

Bugzilla:2184570

## 11.4. 基础架构服务

`bind` 和 `unbound` 都禁用基于 SHA-1- 的签名验证

**bind** 和 **unbound** 组件禁用所有 RSA/SHA1（算法 5）和 RSASHA1-NSEC3-SHA1（算法号 7）签名，且签名的 SHA-1 用法在 DEFAULT 系统范围的加密策略中受到限制。

因此，某些 DNSSEC 记录使用 SHA-1、RSA/SHA1 和 RSASHA1-NSEC3-SHA1 摘要算法无法验证在 Red Hat Enterprise Linux 9 中，受影响的域名会存在安全漏洞。

要临时解决这个问题，升级到不同的签名算法，如 RSA/SHA-256 或 elliptic curve 键。

有关受影响和存在安全漏洞的顶级域的信息和列表，请参阅[使用 RSASHA1 签名的 DNSSEC 记录失败来验证](#) 解决方案。

[Bugzilla:2070495](#)

### 如果在多个区域中使用相同的可写区域文件，**named** 无法启动

BIND 不允许在多个区域中具有相同的可写区域文件。因此，如果配置包含多个区域，它们共享到可由 **named** 服务修改的文件的完整路径，则 **named** 无法启动。要临时解决这个问题，请使用 **in-view** 子句在多个视图间共享一个区域，并确保为不同的区使用不同的完整路径。例如，在完整路径中包含视图名称。

请注意，可写区域文件通常用于允许由 DNSSEC 维护的动态更新、从属区域或区域的区域。

[Bugzilla:1984982](#)

### **libotr** 与 FIPS 不兼容

**libotr** 库和非记录(OTR)消息的工具包为即时消息会话提供了端到端加密。但是，由于其使用了 **gcry\_pk\_sign()** 和 **gcry\_pk\_verify()** 函数，**libotr** 库不符合联邦信息处理标准(FIPS)。因此，您无法在 FIPS 模式下使用 **libotr** 库。

[Bugzilla:2086562](#)

### 设置控制台 **keymap** 在最小安装上需要 **libxkbcommon** 库

在 RHEL 9 中，某些 **systemd** 库依赖项已从动态链接转换为动态加载，以便您的系统在运行时打开并使用库（当它们可用时）。有了这个更改，除非您安装必要的库，否则无法使用依赖于此类库的功能。这也会影响在最小安装的系统上设置键盘布局。因此，**localectl --no-convert set-x11-keymap gb** 命令会失败。

要临时解决这个问题，请安装 **libxkbcommon** 库：

```
# dnf install libxkbcommon
```

[Bugzilla:2214130](#)

### **sysstat** 软件包中的 **%vmeff** 指标显示不正确的值

**sysstat** 软件包提供 **%vmeff** 指标来测量页面回收效率。**sar -B** 命令返回的 **%vmeff** 列的值不正确，因为 **sysstat** 不会解析后续内核版本提供的所有相关的 **/proc/vmstat** 值。要临时解决这个问题，您可以从 **/proc/vmstat** 文件中手动计算 **%vmeff** 值。详情请查看[为什么在 RHEL 8 和 RHEL 9 中 sar \(1\) 工具报告 %vmeff 值超过 100 % ?](#)

[Bugzilla:2230431](#)

## 11.5. 安全性

### **tangd-keygen** 无法正确处理非默认的 **umask**



**tangd-keygen** 脚本不会更改生成的密钥文件的文件权限。因此，在具有阻止向其他用户读取密钥的默认用户文件创建模式掩码(**umask**)的系统上，**tang-show-keys** 命令会返回错误消息 **Internal Error 500**，而不是显示密钥。

要临时解决这个问题，请使用 **chmod o+r \*.jwk** 命令更改 **/var/db/tang** 目录中的文件的权限。

[Bugzilla:2188743](#)

### OpenSSL 不会检测 PKCS#11 令牌是否支持创建原始 RSA 或 RSA-PSS 签名

TLS 1.3 协议需要支持 RSA-PSS 签名。如果 PKCS #11 令牌不支持原始 RSA 或 RSA-PSS 签名，如果密钥由 PKCS#11 令牌保存，则使用 OpenSSL 库的服务器应用程序将无法使用 RSA 密钥。因此，在上述场景中 TLS 通信会失败。

要临时解决这个问题，请配置服务器和客户端以使用 TLS 版本 1.2 作为可用最高 TLS 协议版本。

Bugzilla:1681178

### OpenSSL 错误处理 PKCS #11 tokens 不支持原始 RSA 或 RSA-PSS 签名

OpenSSL 库不会检测到 PKCS #11 令牌的与键相关的功能。因此，当使用不支持原始 RSA 或 RSA-PSS 签名的令牌创建签名时，建立 TLS 连接会失败。

要临时解决这个问题，请在 **/etc/pki/tls/openssl.cnf** 文件的 **crypto\_policy** 部分的 **.include** 行后面添加以下行：

```
SignatureAlgorithms =
RSA+SHA256:RSA+SHA512:RSA+SHA384:ECDSA+SHA256:ECDSA+SHA512:ECDSA+SHA384
MaxProtocol = TLSv1.2
```

因此，可以在描述的场景中建立 TLS 连接。

Bugzilla:1685470

### 当使用特定语法时，scp 会让将文件复制到自己的文件

**scp** 实用程序从安全复制协议 (SCP) 改为更安全的 SSH 文件传输协议 (SFTP)。因此，将文件从位置复制到同一位置，从而擦除文件内容。此问题会产生以下语法：

```
scp localhost:/myfile localhost:/myfile
```

要临时解决这个问题，请不要使用这个语法将文件复制到与源位置相同的目标。

这个问题已针对以下语法解决：

- **scp /myfile localhost:/myfile**
- **scp localhost:~/myfile ~/myfile**

[Bugzilla:2056884](#)

### OSCAP Anaconda 附加组件不会在图形安装中获取定制的配置文件的

OSCAP Anaconda 附加组件不提供一个选项，来在 RHEL 图形安装中选择或取消选择安全配置文件的定制。从 RHEL 8.8 开始，当从存档或 RPM 软件包安装时，附加组件不会考虑定制。因此，安装会显示以下出错信息，而不是获取 OSCP 定制的配置文件的：

```
There was an unexpected problem with the supplied content.
```

要临时解决这个问题，您必须在 Kickstart 文件的 `%addon org_fedora_oscap` 部分中指定路径，例如：

```
xccdf-path = /usr/share/xml/scap/sc_tailoring/ds-combined.xml
tailoring-path = /usr/share/xml/scap/sc_tailoring/tailoring-xccdf.xml
```

因此，您只能将用于 SCAP 定制的配置文件的图形安装与相应的 Kickstart 规格一起使用。

[Bugzilla:2165920](#)

## Ansible 补救需要额外的集合

用 **ansible-core** 软件包替换 Ansible Engine 时，RHEL 订阅提供的 Ansible 模块的列表会减少。因此，运行使用包含在 **scap-security-guide** 软件包中的 Ansible 内容的补救需要来自 **rhc-worker-playbook** 软件包的集合。

对于 Ansible 补救，请执行以下步骤：

1. 安装所需的软件包：

```
# dnf install -y ansible-core scap-security-guide rhc-worker-playbook
```

2. 进到 `/usr/share/scap-security-guide/ansible` 目录：

```
# cd /usr/share/scap-security-guide/ansible
```

3. 运行使用环境变量的相关 Ansible playbook，这些变量定义了到额外 Ansible 集合的路径：

```
# ANSIBLE_COLLECTIONS_PATH=/usr/share/rhc-worker-
playbook/ansible/collections/ansible_collections/ ansible-playbook -c local -i localhost, rhel9-
playbook-cis_server_11.yml
```

将 `cis_server_11` 替换为您要修复系统的配置文件的 ID。

因此，Ansible 内容会被正确处理。



### 注意

对 **rhc-worker-playbook** 中提供的集合的支持仅限于启用 **scap-security-guide** 中提供的 Ansible 内容。

[Bugzilla:2105162](#)

## oscap-anaconda-addon 不允许带有 Network Servers 软件包组的系统的 CIS 强化

当在选择了 Network Servers 软件包组的系统上使用 CIS 安全配置文件 (`cis`, `cis_server_11`, `cis_workstation_11`, 或 `cis_workstation_12`) 安装 RHEL 网络服务器时，**oscap-anaconda-addon** 会发送错误消息 **package tftp has been added to the list of excluded packages, but it can't be removed from the current software selection without breaking the install**。要继续安装，请回到 Software Selection，并取消选择 **Network Servers** 附加软件，以允许安装和强化完成。然后，安装所需的软件包。

[Bugzilla:2172264](#)

## Keylime 不接受串联的 PEM 证书



当 Keylime 将证书链作为 PEM 格式的、串联在一个文件中的多个证书接收时，**keylime-agent-rust** Keylime 组件在签名验证过程中不能正确地使用所有提供的证书，导致 TLS 握手失败。因此，客户端组件 (**keylime\_verifier** 和 **keylime\_tenant**) 无法连接到 Keylime 代理。要临时解决这个问题，请只使用一个证书而不是多个证书。

Jira:RHELPLAN-157225

### Keylime 需要 `tls_dir = default` 的特定文件

当在 Keylime verifier 或 registrar 配置中将 `tls_dir` 变量设为 **default** 时，Keylime 会检查 `/var/lib/keylime/cv_ca` 目录中是否存在 `cacert.crt` 文件。如果文件不存在，则 **keylime\_verifier** 或 **keylime\_registrar** 服务无法启动，会在日志中记录以下信息：**Exception: It appears that the verifier has not yet created a CA and certificates, please run the verifier first.** 因此，Keylime 会拒绝具有不同文件名的自定义证书颁发机构(CA)证书，即使它们被在 `/var/lib/keylime/ca_cv` 目录中。

要临时解决这个问题并使用自定义 CA 证书，请手动指定 `tls_dir =/var/lib/keylime/ca_cv`，而不是使用 `tls_dir = default`。

Jira:RHELPLAN-157337

### 默认 SELinux 策略允许无限制的可执行文件使其堆栈可执行

SELinux 策略中的 **selinuxuser\_execstack** 布尔值的默认状态是 on，这意味着无限制的可执行文件可以使其堆栈为可执行。可执行文件不应该使用这个选项，这通常代表开发的可执行代码的质量较差，或可能存在安全攻击的风险。但是，由于需要与其他工具、软件包和第三方产品保持兼容，红帽无法更改默认策略中的这个布尔值。如果您的环境没有此类兼容性问题，请使用 **setsebool -P selinuxuser\_execstack off** 命令在您的本地策略中将这个布尔值设置为 off。

Bugzilla:2064274

### STIG 配置文件中的 SSH 超时规则配置了不正确的选项

对 OpenSSH 的更新会影响以下 Defense Information Systems Agency Security Technical Implementation Guide (DISA STIG) 配置集中的规则：

- DISA STIG for RHEL 9 (**xccdf\_org.ssgproject.content\_profile\_stig**)
- DISA STIG with GUI for RHEL 9 (**xccdf\_org.ssgproject.content\_profile\_stig\_gui**)

在每个配置集中，以下两条规则会受到影响：

Title: Set SSH Client Alive Count Max to zero  
 CCE Identifier: CCE-90271-8  
 Rule ID: xccdf\_org.ssgproject.content\_rule\_sshd\_set\_keepalive\_0

Title: Set SSH Idle Timeout Interval  
 CCE Identifier: CCE-90811-1  
 Rule ID: xccdf\_org.ssgproject.content\_rule\_sshd\_set\_idle\_timeout

当应用到 SSH 服务器时，每个规则都会配置一个选项 (**ClientAliveCountMax** 和 **ClientAliveInterval**)，其行为不再像之前一样。因此，当 OpenSSH 达到这些规则配置的超时时间时，OpenSSH 不再断开空闲的 SSH 用户。作为临时解决方案，这些规则已从 DISA STIG for RHEL 9 和 DISA STIG with GUI for RHEL 9 配置集中临时删除，直到开发出解决方案为止。

Bugzilla:2038978

### GnuPG 错误地允许使用 SHA-1 签名，即使通过 `crypto-policies` 禁止使用 SHA-1 签名

无论系统范围的加密策略中定义的设置如何，GNU Privacy Guard(GnuPG)加密软件可以创建和验证使用 SHA-1 算法的签名。因此，您可以在 **DEFAULT** 加密策略中将 SHA-1 用于加密目的，这与这个不安全算法的系统范围弃用没有一致的。

要临时解决这个问题，请不要使用涉及 SHA-1 的 GnuPG 选项。因此，您将使用非安全 SHA-1 签名来防止 GnuPG 降低默认系统安全性。

[Bugzilla:2070722](#)

### gpg-agent 在 FIPS 模式下不作为 SSH 代理工作

在将密钥添加到 **ssh-agent** 程序时，**gpg-agent** 工具会创建 MD5 指纹，即使 FIPS 模式禁用了 MD5 摘要。因此，**ssh-add** 程序无法将密钥添加到身份验证代理中。

要临时解决这个问题，请在不使用 **gpg-agent --daemon --enable-ssh-support** 命令的情况下创建 `~/.gnupg/sshcontrol` 文件。例如，您可以将 `<FINGERPRINT> 0` 格式的 **gpg --list-keys** 命令的输出粘贴到 `~/.gnupg/sshcontrol`。因此，**gpg-agent** 作为 SSH 验证代理工作。

[Bugzilla:2073567](#)

### OpenSCAP 内存消耗问题

在具有有限内存的系统上，OpenSCAP 扫描程序可能会预先终止，或者可能无法产生结果文件。要临时解决这个问题，您可以自定义扫描配置文件，以取消选择涉及递归整个 / 文件系统的规则：

- **rpm\_verify\_hashes**
- **rpm\_verify\_permissions**
- **rpm\_verify\_ownership**
- **file\_permissions\_unauthorized\_world\_writable**
- **no\_files\_unowned\_by\_user**
- **dir\_perms\_world\_writable\_system\_owned**
- **file\_permissions\_unauthorized\_suid**
- **file\_permissions\_unauthorized\_sgid**
- **file\_permissions\_ungroupowned**
- **dir\_perms\_world\_writable\_sticky\_bits**

如需了解更多详细信息和临时解决方案，请参阅相关的 [知识库文章](#)。

[Bugzilla:2161499](#)

### 在 kickstart 安装过程中修复与服务相关的规则可能会失败

在 kickstart 安装过程中，OpenSCAP 工具有时会错误地显示服务的 **enable** 或 **disable** 状态补救不需要。因此，OpenSCAP 可能会将安装的系统上的服务设置为不合规的状态。作为临时解决方案，您可以在 kickstart 安装后扫描并修复该系统。这可以解决与服务相关的问题。

[BZ#1834716](#)

## 11.6. 网络

## nm-cloud-setup 服务从接口中删除了手动配置的辅助 IP 地址

根据从云环境收到的信息，**nm-cloud-setup** 服务会配置网络接口。禁用 **nm-cloud-setup** 以手动配置接口。然而，在某些情况下，主机上的其他服务也可以配置接口。例如，这些服务可以添加辅助 IP 地址。为了避免 **nm-cloud-setup** 删除辅助 IP 地址：

1. 停止并禁用 **nm-cloud-setup** 服务和计时器：

```
# systemctl disable --now nm-cloud-setup.service nm-cloud-setup.timer
```

2. 显示可用的连接配置文件：

```
# nmcli connection show
```

3. 激活受影响的连接配置文件：

```
# nmcli connection up "<profile_name>"
```

因此，该服务不再从接口中删除手动配置的辅助 IP 地址。

[Bugzilla:2151040](#)

## 更新会话密钥失败会导致连接中断

内核传输层安全(kTLS)协议不支持更新会话密钥，这些密钥由对称密码使用。因此，用户无法更新密钥，从而导致连接中断。要临时解决这个问题，请禁用 kTLS。因此，解决这一问题，可以成功更新会话密钥。

[Bugzilla:2013650](#)

## 默认情况下不安装 initscripts 软件包

默认情况下，不会安装 **initscripts** 软件包。因此，**ifup** 和 **ifdown** 工具不可用。一个替代的方法是，可以使用 **nmcli connection up** 和 **nmcli connection down** 命令来启用和禁用连接。如果这个替代方法无法正常工作，请报告这个问题并安装 **NetworkManager-initscripts-updown** 软件包，该软件包为 **ifup** 和 **ifdown** 工具提供了一个 NetworkManager 解决方案。

[Bugzilla:2082303](#)

## 使用带有 mlx5 驱动程序的 XDP 多缓冲模式，大于 3498 字节的 MTU 需要禁用 RX Striding RQ

在与以下所有条件匹配的主机上运行具有多缓冲区模式的 eXpress Data Path (XDP)脚本失败：

- 主机使用 **mlx5** 驱动程序。
- 最大传输单元(MTU)值大于 3498 字节。
- 接收跨步接收队列(RX Striding RQ)功能在 Mellanox 接口上已启用。

如果所有条件都适用，脚本会失败，并显示 **link set xdp fd failed** 错误。要在具有较高 MTU 的主机上运行 XDP 脚本，请禁用 Mellanox 接口上的 RX Striding RQ：

```
# ethtool --set-priv-flags <interface_name> rx_striding_rq off
```

因此，您可以在使用 **mlx5** 驱动程序，并有大于 3498 字节的 MTU 值的接口上使用 XDP 多缓冲区模式。

Jira:RHEL-6496

## 11.7. 内核

### 内核中的 `kdump` 机制会引起 64K 内核的 OOM 错误

64 位 ARM 架构上的 64K 内核页大小使用的内存比 4KB 内核使用的多。因此，`kdump` 会导致内核 panic，内存分配失败，并报内存不足(OOM)错误。作为临时解决方案，手动将 `crashkernel` 的值配为 640 MB。例如，将 `crashkernel=` 参数设为 `crashkernel=2G-:640M`。

因此，`kdump` 机制对上述场景的 64K 内核不会失败。

Bugzilla:2160676

### 当从 4k 变为 64k 页大小内核时，依赖内核页大小的客户应用程序可能需要更新

RHEL 与 4k 和 64k 页大小内核都兼容。当从 4k 变为 64k 页大小内核时，依赖 4k 内核页大小的客户应用程序可能需要更新。已知的实例包括 `jemalloc` 和依赖的应用程序。

`jemalloc` 内存分配器库对系统运行时环境中使用的页大小敏感。库可以构建成为与 4k 和 64k 页大小内核兼容，例如，当使用 `--with-lg-page=16` 或 `env JEMALLOC_SYS_WITH_LG_PAGE=16` 配置时（用于 `jemallocator` Rust crate）。因此，运行时环境的页大小与编译依赖于 `jemalloc` 的二进制文件时出现的页大小之间可能会出现不匹配。因此，使用基于 `jemalloc` 的应用程序会触发以下错误：

```
<jemalloc>: Unsupported system page size
```

要避免这个问题，请使用以下方法之一：

- 使用合适的构建配置或环境选项来创建 4k 和 64k 页大小兼容二进制文件。
- 在引导到最终的 64k 内核和运行时环境后，构建任何使用 `jemalloc` 的用户空间软件包。

例如，您可以构建 `fd-find` 工具，该工具也通过 `cargo` Rust 软件包管理器使用 `jemalloc`。在最后的 64k 环境中，输入 `cargo` 命令触发所有依赖项的新构建，以解决页大小中的不匹配：

```
# cargo install fd-find --force
```

Bugzilla:2167783

### `kdump` 服务无法在 IBM Z 系统中构建 `initrd` 文件

在 64 位 IBM Z 系统中，当 `znet` 相关配置信息（如 `s390-subchannels`）位于不活跃 `NetworkManager` 连接配置集时，`kdump` 服务无法加载初始 RAM 磁盘 (`initrd`)。因此，`kdump` 机制会失败并显示以下错误：

```
dracut: Failed to set up znet
kdump: mkdumprd: failed to make kdump initrd
```

作为临时解决方案，请使用以下解决方案之一：

- 通过重新使用具有 `znet` 配置信息的连接配置集来配置网络绑定或桥接：

```
$ nmcli connection modify enc600 master bond0 slave-type bond
```

- 将 `znet` 配置信息从不活跃连接配置集复制到活跃连接配置集中：

- a. 运行 **nmcli** 命令查询 **NetworkManager** 连接配置集：

```
# nmcli connection show

NAME                UUID                TYPE Device
bridge-br0          ed391a43-bdea-4170-b8a2 bridge br0
bridge-slave-enc600 caf7f770-1e55-4126-a2f4 ethernet enc600
enc600              bc293b8d-ef1e-45f6-bad1 ethernet --
```

- b. 使用不活跃连接中的配置信息更新活跃的配置集：

```
#!/bin/bash
inactive_connection=enc600
active_connection=bridge-slave-enc600
for name in nettype subchannels options; do
field=802-3-ethernet.s390-$name
val=$(nmcli --get-values "$field"connection show "$inactive_connection")
nmcli connection modify "$active_connection" "$field" $val
done
```

- c. 重启 **kdump** 服务以使更改生效：

```
# kdumpectl restart
```

[Bugzilla:2064708](#)

### KTLS 不支持将 TLS 1.3 卸载到 NIC

内核传输层安全(kTLS)不支持将 TLS 1.3 卸载到 NIC。因此，即使 NIC 支持 TLS 卸载，软件加密也会与 TLS 1.3 一起使用。要临时解决这个问题，如果需要卸载，禁用 TLS 1.3。因此，您只能卸载 TLS 1.2。当使用 TLS 1.3 时，性能较低，因为无法卸载 TLS 1.3。

[Bugzilla:2000616](#)

### 默认情况下，Delay Accounting 功能不会显示 SWAPIN 和 IO% 统计列

**Delayed Accounting** 功能与早期版本不同，它们会被默认禁用。因此，**iostat** 应用程序不显示 **SWAPIN** 和 **IO%** 统计列，并显示以下警告：

```
CONFIG_TASK_DELAY_ACCT not enabled in kernel, cannot determine SWAPIN and IO%
```

**Delay Account** 功能使用 **taskstats** 接口，为属于线程组的所有任务或线程提供延迟统计。当任务等待 kernel 资源可用时，会延迟执行，例如：等待空闲 CPU 运行的任务。统计有助于设置任务的 CPU 优先级、I/O 优先级和 **rss** 限制值。

作为临时解决方案，您可以在运行时或引导时启用 **delayacct** 引导选项。

- 要在运行时启用 **delayacct**，请输入：

```
echo 1 > /proc/sys/kernel/task_delayacct
```

请注意，这个命令可启用系统范围功能，但只适用于您在运行此命令后启动的任务。

- 要在引导时永久启用 **delayacct**，请使用以下步骤之一：

- 编辑 `/etc/sysctl.conf` 文件以覆盖默认参数：
  - a. 在 `/etc/sysctl.conf` 文件中添加以下条目：
 

```
kernel.task_delayacct = 1
```

如需更多信息，请参阅 [如何在 Red Hat Enterprise Linux 上设置 sysctl 变量](#)。
  - b. 重启系统以使更改生效。
- 在内核命令行中添加 `delayacct` 选项。  
如需更多信息，请参阅 [配置内核命令行参数](#)。

因此，`iotop` 应用程序会显示 **SWAPIN** 和 **IO%** 统计列。

Bugzilla:2132480

### kdump 机制无法捕获 LUKS 加密目标上的 vmcore 文件

当在使用 Linux Unified Key Setup(LUKS)加密分区的系统中运行 `kdump` 时，系统需要特定的可用内存。当可用内存小于所需内存量时，`systemd-cryptsetup` 服务将无法挂载分区。因此，第二个内核无法捕获 LUKS 加密目标上的崩溃转储文件(`vmcore`)。

使用 `kdumpctl estimate` 命令，您可以查询 **推荐的 crashkernel 值**，这是 `kdump` 所需的内存大小。

要解决这个问题，请按照以下步骤在 LUKS 加密目标中为 `kdump` 配置所需的内存：

1. 输出估计的 `crashkernel` 值：

```
# kdumpctl estimate
```

2. 通过增大 `crashkernel` 值来配置所需的内存量：

```
# grubby --args=crashkernel=652M --update-kernel=ALL
```

3. 重启系统以使更改生效。

```
# reboot
```

因此，`kdump` 在带有 LUKS 加密分区的系统上可以正常工作。

Bugzilla:2017401

### 在引导时分配崩溃内核内存失败

在某些 Ampere Altra 系统中，在可用内存低于 1GB 时为 `kdump` 分配崩溃内核内存会失败。因此，`kdumpctl` 命令无法启动 `kdump` 服务。

要解决这个问题，请执行以下操作之一：

- 减少 `crashkernel` 参数的值（最少 240 MB）以满足大小要求，例如 `crashkernel=240M`）。
- 使用 `crashkernel=x,high` 选项为 `kdump` 保留大于 4 GB 的崩溃内核内存。

因此，`kdump` 的崩溃内核内存分配不会在 Ampere Altra 系统上失败。

[Bugzilla:2065013](#)

### 启用 VMD 时，RHEL 无法识别 NVMe 磁盘

当您重置或重新附加驱动程序时，卷组管理设备(VMD)域目前不会软重置。因此，硬件无法正确检测并枚举其设备。因此，启用了 VMD 的操作系统无法识别 NVMe 磁盘，特别是在重置服务器或使用虚拟机时。

[Bugzilla:2128610](#)

### iwl7260-firmware 破坏了 Intel Wi-Fi 6 AX200、AX210 和 Lenovo ThinkPad P1 Gen 4 上的 Wi-Fi

在将 **iwl7260-firmware** 或 **iwl7260-wifi** 驱动程序更新到 RHEL 9.1 及之后的版本提供的版本后，硬件会进入不正确的内部状态。错误地报告其状态。因此，Intel Wifi 6 卡可能无法正常工作，并显示错误信息：

```
kernel: iwlwifi 0000:09:00.0: Failed to start RT ucode: -110
kernel: iwlwifi 0000:09:00.0: WRT: Collecting data: ini trigger 13 fired (delay=0ms)
kernel: iwlwifi 0000:09:00.0: Failed to run INIT ucode: -110
```

一个没有确认的临时解决方法是关闭并再次打开系统。不要重启。

[Bugzilla:2129288](#)

### kmod 中的 weak-modules 不能与模块间依赖一起工作

**kmod** 软件包提供的 **weak-modules** 脚本决定了哪些模块与安装的内核 kABI 兼容。但是，在检查模块的内核兼容性时，**weak-modules** 会从它们构建的内核的高版本到低版本来处理模块符号依赖关系。因此，针对不同内核版本构建的具有相互依赖关系的模块可能会被解释为不兼容，因此 **weak-modules** 脚本无法在此场景下工作。

要临时解决这个问题，请在安装新内核前针对最新的库存内核进行构建或放置额外的模块。

[Bugzilla:2103605](#)

### 使用 Mellanox ConnectX-5 适配器时，mlx5 驱动程序会失败

在以太网交换机设备驱动程序型号(**switchdev**)模式下，当使用设备管理的流控制(DMFS)参数和 **ConnectX-5** 适配器支持的硬件配置时，**mlx5** 驱动程序会失败。因此，您可以看到以下错误信息：

```
BUG: Bad page cache in process umount pfn:142b4b
```

要临时解决这个问题，请使用软件管理的流控制 (SMFS) 参数，而不是 DMFS。

[Bugzilla:2180665](#)

### 在具有大量核数的系统上实时内核的硬件认证可能需要传递 skew\_tick=1 引导参数来避免锁争用

具有大量插槽和大量核数的大型系统可能会因为 **xtime\_lock** 上的锁争用而经历延迟峰值，**xtime\_lock** 用于计时系统中。因此，在多处理系统上的硬件认证中可能会发生延迟峰值和延迟。作为临时解决方案，您可以通过添加 **skew\_tick=1** 引导参数来偏移每个 CPU 的计时器刻度，以便在不同的时间启动。

要避免锁冲突，请启用 **skew\_tick=1**：

1. 使用 **grubby** 启用 **skew\_tick=1** 参数。

```
# grubby --update-kernel=ALL --args="skew_tick=1"
```



2. 重启以使更改生效。
3. 运行 `cat /proc/cmdline` 命令来验证新设置。

请注意，启用 `skew_tick=1` 会导致大量功耗增加，因此只有在您运行延迟敏感实时工作负载时才必须启用。

Bugzilla:2214508

### DKMS 对 64 位 ARM CPU 上正确编译的驱动程序提供一条不正确的程序失败警告

动态内核模块支持(`dkms`)工具无法识别用于 4 KB 内核和 64 KB 页面大小的 64 位 ARM CPU 的内核标头。因此，当执行内核更新且 `kernel-64k-devel` 软件包未安装时，`dkms` 提供一条错误警告，关于为什么程序在正确编译的驱动程序上失败。要临时解决这个问题，请安装 `kernel-headers` 软件包，其包含两种类型的 ARM CPU 架构的头文件，且不特定于 `dkms` 及其要求。

Jira:RHEL-25967

## 11.8. 文件系统和存储

### 在出现不成功的 CHAP 验证尝试后，Anaconda 无法使用 no authentication 方法登录 iSCSI 服务器

当您使用 CHAP 身份验证添加 iSCSI 磁盘时，如果因为凭证不正确而导致登录失败，使用 `no authentication` 方法尝试重新登录也将失败。要解决这个问题，请先关闭当前会话，再使用 `no authentication` 方法登录。

Bugzilla:1983602

### NVMe/TCP 不支持设备映射器多路径

使用带有 `nvme-tcp` 驱动程序的设备映射器多路径可能会导致 Call Trace 警告和系统不稳定。要临时解决这个问题，NVMe/TCP 用户必须启用原生 NVMe 多路径，且不能在 NVMe 中使用 `device-mapper-multipath` 工具。

默认情况下，RHEL 9 中启用了原生 NVMe 多路径。如需更多信息，请参阅[在 NVMe 设备上启用多路径](#)。

Bugzilla:2033080

### blk-availability systemd 服务停用了复杂的设备堆栈

在 `systemd` 中，默认的块停用代码并不总是正确处理虚拟块设备的复杂堆栈。在一些配置中，虚拟设备在关闭过程中可能无法被删除，这会导致记录错误信息。要临时解决这个问题，请执行以下命令来停用复杂块设备堆栈：

```
# systemctl enable --now blk-availability.service
```

因此，复杂虚拟设备堆栈会在关闭过程中被正确停用，且不会生成错误消息。

Bugzilla:2011699

### 对于启用了配额的情况下挂载的 XFS 文件系统，不再可能禁用配额记帐

从 RHEL 9.2 开始，不再可能对启用了配额的情况下挂载的 XFS 文件系统禁用配额记帐。

要临时解决这个问题，请去掉配额选项，重新挂载文件系统来禁用配额记帐。



Bugzilla:2160619

### 当在 `/etc/fstab` 中将 NVMe-FC 设备添加为挂载点时系统无法引导

通过 `/etc/fstab` 文件挂载的光纤通道(NVMe-FC)设备上的 Non-volatile Memory Express 无法在引导时挂载，系统进入紧急模式。这是因为 `nvme-cli nvmf-autoconnect systemd` 服务中的一个已知的 bug。

Bugzilla:2168603

### 对 NVMe 设备的 udev 规则更改

对 NVMe 设备有一个 udev 规则更改，即添加了 `OPTIONS="string_escape=replace"` 参数。如果您设备的序列号前面有空格，则这会导致某些厂商的对按 id 命名的磁盘进行更改。

Bugzilla:2185048

## 11.9. 动态编程语言、网页和数据库服务器

### python3.11-lxml 不提供 lxml.isoschematron 子模块

`python3.11-lxml` 软件包不与 `lxml.isoschematron` 子模块一起分发，因为它不是开源许可证。子模块实现 ISO 架构支持。作为替代方案，`lxml.etree.Schematron` 类种提供了 pre-ISO-Schematron 验证。`python3.11-lxml` 软件包的其余内容不受影响。

Bugzilla:2157708

### MySQL 和 MariaDB 中的 `--ssl-fips-mode` 选项不会改变 FIPS 模式

RHEL 中 `MySQL` 和 `MariaDB` 中的 `--ssl-fips-mode` 选项与上游中的工作方式不同。

在 RHEL 9 中，如果您使用 `--ssl-fips-mode` 作为 `mysqld` 或 `mariadb` 守护进程的参数，或者在 `MySQL` 或 `MariaDB` 服务器配置文件中 `ssl-fips-mode`，则 `--ssl-fips-mode` 不会更改这些数据库服务器的 FIPS 模式。

相反：

- 如果将 `--ssl-fips-mode` 设为 `ON`，则 `mysqld` 或 `mariadb` 服务器守护进程不会启动。
- 如果您在启用了 FIPS 的系统上将 `--ssl-fips-mode` 设为 `OFF`，则 `mysqld` 或 `mariadb` 服务器守护进程仍然在 FIPS 模式下运行。

这是预期的，因为应该为整个 RHEL 系统启用或禁用 FIPS 模式，而不是为特定组件。

因此，不要在 RHEL 中的 `MySQL` 或 `MariaDB` 中使用 `--ssl-fips-mode` 选项。相反，请确保在整个 RHEL 系统上启用 FIPS 模式：

- 最好安装启用了 FIPS 模式的 RHEL。在安装过程中启用 FIPS 模式可确保系统使用 FIPS 批准的算法生成所有的密钥，并持续监控测试。有关在 FIPS 模式下安装 RHEL 的详情，请参考 [在 FIPS 模式下安装系统](#)。
- 或者，您可以按照 [将系统切换到 FIPS 模式](#) 中的流程，为整个 RHEL 系统切换 FIPS 模式。

Bugzilla:1991500

## 11.10. 编译器和开发工具

### 某些基于符号的探测无法在 64 位 ARM 架构上的 SystemTap 中工作

内核配置禁用 **SystemTap** 所需要的某些功能。因此，一些基于符号的探测无法在 64 位 ARM 架构上工作。因此，受影响的 **SystemTap** 脚本可能无法运行，或可能不会收集对所需探测点的点击。

请注意，随着 [RHBA-2022:5259](#) 公告的发布，已针对其余架构对此 bug 进行了修复。

Bugzilla:2083727

## GCC Toolset 12 中的 GCC : 对 Intel Sapphire Rapids 处理器的 CPU 检测可能会失败

对 Intel Sapphire Rapids 处理器的 CPU 检测依赖于 **AVX512\_VP2INTERSECT** 功能的存在。此功能已从 GCC 的 GCC Toolset 12 版本中删除，因此，对 Intel Sapphire Rapids 处理器的 CPU 检测可能会失败。

Bugzilla:2141718

## 11.11. 身份管理

### 在 Directory 服务器中为后缀配置引用失败

如果您在 Directory 服务器中设置了后端引用，使用 **dsconf <instance\_name> backend suffix set --state referral** 命令设置后端状态会失败，并显示以下错误：

```
Error: 103 - 9 - 53 - Server is unwilling to perform - [] - need to set nsslapd-referral before moving to referral state
```

因此，为后缀配置引用会失败。要临时解决这个问题：

1. 手动设置 **nsslapd-referral** 参数：

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com
dn: cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping tree,cn=config
changetype: modify
add: nsslapd-referral
nsslapd-referral: ldap://remote_server:389/dc=example,dc=com
```

2. 设置后端状态：

```
# dsconf <instance_name> backend suffix set --state referral
```

因此，您可以使用临时解决方案为后缀配置引用。

Bugzilla:2063140

### dsconf 实用程序没有为 entryUUID 插件创建修复任务的选项

**dsconf** 实用程序没有提供为 **entryUUID** 插件创建修复任务的选项。因此，管理员无法使用 **dsconf** 创建任务来自动将 **entryUUID** 属性添加到现有条目。作为临时解决方案，请手动创建任务：

```
# ldapadd -D "cn=Directory Manager" -W -H ldap://server.example.com -x
dn: cn=entryuuid_fixup_<time_stamp>,cn=entryuuid task,cn=tasks,cn=config
objectClass: top
objectClass: extensibleObject
```

```
basedn: <fixup base tree>
cn: entryuuid_fixup_<time_stamp>
filter: <filtered_entry>
```

创建了任务后，Directory 服务器会修复缺少或无效 **entryUUID** 属性的条目。

[Bugzilla:2047175](#)

### MIT Kerberos 不支持 PKINIT 的 ECC 证书

MIT Kerberos 不对评论文档实施 RFC5349 请求，它描述了公钥 Cryptography 中的 elliptic-curve 加密 (ECC) 支持。因此，RHEL 使用的 MIT **krb5-pkinit** 软件包不支持 ECC 证书。如需更多信息，请参阅 [Kerberos \(PKINIT\)对公共密钥加密加密支持\(ECC\) 支持](#)。

[Bugzilla:2106043](#)

### 必须在 RHEL 9 客户端上设置 DEFAULT:SHA1 子策略，以使 PKINIT 能够针对 AD KDC 工作

现在，RHEL 9 中弃用了 SHA-1 摘要算法，对公共密钥 Cryptography for Public Key Cryptography 的 CMS 消息使用更强大的 SHA-256 算法签名。

但是，Active Directory (AD) Kerberos Distribution Center (KDC) 仍然使用 SHA-1 摘要算法为 CMS 信息签名。因此，RHEL 9 Kerberos 客户端无法通过对 AD KDC 使用 PKINIT 来验证用户。

要临时解决这个问题，使用以下命令在 RHEL 9 系统上启用对 SHA-1 算法的支持：

```
# update-crypto-policies --set DEFAULT:SHA1
```

[Bugzilla:2060798](#)

### 如果 RHEL 9 Kerberos 代理与非 RHEL-9 和非 AD Kerberos 代理通信，则用户的 PKINIT 身份验证会失败

如果 RHEL 9 Kerberos 代理（客户端或 Kerberos 分发中心(KDC) 与不是 Active Directory (AD) 代理的非 RHEL-9 Kerberos 代理交互，则用户的 PKINIT 身份验证会失败。要临时解决这个问题，请执行以下操作之一：

- 将 RHEL 9 代理的 crypto-policy 设置为 **DEFAULT:SHA1** 以允许验证 SHA-1 签名：

```
# update-crypto-policies --set DEFAULT:SHA1
```

- 更新非 RHEL-9 和非 AD 代理，以确保它不使用 SHA-1 算法为 CMS 数据签名。因此，将您的 Kerberos 客户端或 KDC 软件包更新至使用 SHA-256 而不是 SHA-1 的版本：
  - CentOS 9 Stream: krb5-1.19.1-15
  - RHEL 8.7: krb5-1.18.2-17
  - RHEL 7.9: krb5-1.15.1-53
  - Fedora Rawhide/36: krb5-1.19.2-7
  - Fedora 35/34 : krb5-1.19.2-3

因此，用户的 PKINIT 身份验证可以正常工作。

请注意，对于其他操作系统，这是 krb5-1.20 版本，可确保代理使用 SHA-256 而不是 SHA-1 为 CMS 数据进行签名。

另请参阅 [必须在 RHEL 9 客户端上设置 DEFAULT:SHA1 子策略，以使 PKINIT 能够针对 AD KDC 工作](#)。

[Bugzilla:2077450](#)

## AD 信任的 FIPS 支持需要 AD-SUPPORT 加密子策略

Active Directory(AD)使用 AES SHA-1 HMAC 加密类型，默认情况下在 RHEL 9 上不允许 FIPS 模式。如果要使用带有 AD 信任的 RHEL 9 IdM 主机，请在安装 IdM 软件前支持 AES SHA-1 HMAC 加密类型。

由于 FIPS 合规性是一个涉及技术和机构协议的过程，因此，请在启用 **AD-SUPPORT** 子策略前咨询 FIPS 审核员，以允许采取技术措施支持 AES SHA-1 HMAC 加密类型，然后安装 RHEL IdM：

```
# update-crypto-policies --set FIPS:AD-SUPPORT
```

[Bugzilla:2057471](#)

## Heimdal 客户端无法针对 RHEL 9 KDC 使用 PKINIT 来验证用户

默认情况下，Heimdal Kerberos 客户端通过使用 Modular Exponential (MODP) Diffie-Hellman Group 2 用于互联网密钥交换 (IKE) 启动 IdM 用户的 PKINIT 身份验证。但是，RHEL 9 上的 MIT Kerberos 分配中心 (KDC) 仅支持 MODP 组 14 和 16。

因此，pre-authentication 请求会失败并显示 **krb5\_get\_init\_creds: PREAUTH\_FAILED** 错误，在 RHEL MIT KDC 中 **不接受 Key 参数**。

要临时解决这个问题，请确保 Heimdal 客户端使用 MODP Group 14。将客户端配置文件的 **libdefaults** 部分中的 **pkinit\_dh\_min\_bits** 参数设置为 1759：

```
[libdefaults]
pkinit_dh_min_bits = 1759
```

因此，Heimdal 客户端可以针对 RHEL MIT KDC 完成 PKINIT 预验证。

[Bugzilla:2106296](#)

## FIPS 模式下的 IdM 不支持使用 NTLMSSP 协议来建立双向跨林信任

在活动目录(AD)和启用了 FIPS 模式的身份管理(IdM)之间建立双向跨林信任会失败，因为新技术局域网管理器安全支持提供程序 (NTLMSSP)身份验证不符合 FIPS。FIPS 模式下的 IdM 不接受在尝试验证时 AD 域控制器使用的 RC4 NTLM 哈希。

[Bugzilla:2124243](#)

## IdM 到 AD 跨域 TGS 请求失败

IdM Kerberos 票据中的 Privilege Attribute 证书(PAC) 信息现在使用 AES SHA-2 HMAC 加密进行签名，这是 Active Directory (AD) 不支持的。

因此，IdM 到 AD 跨域 TGS 请求（即双向信任设置）失败，并显示以下错误：

```
Generic error (see e-text) while getting credentials for <service principal>
```

[Bugzilla:2060421](#)

## IdM Vault 加密和解密在 FIPS 模式下失败

如果启用了 FIPS 模式，则 OpenSSL RSA-PKCS1v15 填充加密会被阻止。因此，身份管理(IdM) Vault 无法正常工作，因为 IdM 目前正在使用 PKCS1v15 填充来用传输证书包装会话密钥。

[Bugzilla:2089907](#)

## 升级后，没有 SID 的用户无法登录到 IdM

将 IdM 副本升级到 RHEL 9.2 后，IdM Kerberos 分发中心(KDC)可能无法向没有为其账户分配安全标识符(SID)的用户发出票据授予票据(TGT)。因此，用户无法登录到其帐户。

要临时解决这个问题，请以 IdM 管理员身份在拓扑中的另一个 IdM 副本上运行以下命令来生成 SID：

```
# ipa config-mod --enable-sid --add-sids
```

之后，如果用户仍然无法登录，请检查目录服务器错误日志。您可能需要调整 ID 范围，来包含用户 POSIX 身份。

如需更多信息，请参阅 [升级到 RHEL9 时，IDM 用户无法再登录](#) 知识库解决方案。

Jira:RHELPLAN-157939

## 迁移的 IdM 用户可能会因为不匹配的域 SID 而无法登录

如果您使用 `ipa migrate-ds` 脚本将用户从一个 IdM 部署迁移到另一个，则这些用户可能会在使用 IdM 服务时有问题，因为它们之前存在的安全标识符(SID)没有当前 IdM 环境的域 SID。例如，这些用户可以使用 `kinit` 工具检索 Kerberos 票据，但不能登录。要临时解决这个问题，请参阅以下知识库文章：[Migrated IdM 用户因为不匹配的域 SID 而无法登录](#)。

Jira:RHELPLAN-109613

## 由于生成用户 PAC 的加密类型不兼容，MIT krb5 用户无法获取 AD TGT

在 MIT `krb5 1.20` 以及后续的软件包中，默认在所有 Kerberos 票据中都包括特权属性证书(PAC)。MIT Kerberos 分发中心(KDC)选择可用的最强的加密类型，来在 PAC 中生成 KDC 校验和，这目前是 RFC8009 中定义的 **AES HMAC-SHA2** 加密类型。但是，活动目录(AD)不支持这个 RFC。因此，在 AD-MIT 跨领域设置中，MIT `krb5` 用户无法获取 AD 票据授予票据(TGT)，因为 MIT KDC 生成的跨领域 TGT 在 PAC 中包含不兼容的 KDC 校验和类型。

要临时解决这个问题，对于 `/var/kerberos/krb5kdc/kdc.conf` 配置文件的 `[realms]` 部分中的 MIT 领域，请将 `disable_pac` 参数设为 `true`。因此，MIT KDC 会生成没有 PAC 的票据，这意味着 AD 会跳过失败的校验和验证，MIT `krb5` 用户可以获取 AD TGT。

[Bugzilla:2016312](#)

## 对 `ldap_id_use_start_tls` 选项使用默认值时的潜在风险

当对身份查询使用没有 TLS 的 `ldap://` 时，可能会对攻击向量构成风险。特别是中间人(MITM)攻击，例如，它可能允许攻击者通过更改 LDAP 搜索中返回的对象的 UID 或 GID 来冒充用户。

目前，SSSD 配置选项强制 TLS `ldap_id_use_start_tls` 的默认值为 `false`。确保您的设置在可信环境中执行，并决定是否可以对 `id_provider = ldap` 使用未加密的通信。注意 `id_provider = ad` 和 `id_provider = ipa` 不受影响，因为它们使用 SASL 和 GSSAPI 保护的加密连接。

如果使用未加密的通信不安全，请在 `/etc/sss/sss.conf` 文件中将 `ldap_id_use_start_tls` 选项设为 `true` 来强制使用 TLS。计划在以后的 RHEL 版本中更改默认行为。

Jira:RHELPLAN-155168

## 将 FIPS 模式下的 RHEL 9 副本添加到用 RHEL 8.6 或更早版本初始化的 FIPS 模式下的 IdM 部署会失败

略旨在遵守 FIPS 140-3 的默认 RHEL 9 FIPS 加密策略不允许使用 AES HMAC-SHA1 加密类型的密钥派生功能，如 5.1 章节 RFC3961 所定义的。

当在 FIPS 模式下将 RHEL 9 身份管理(IdM)副本添加到 FIPS 模式下的 RHEL 8 IdM 环境（其中，第一个服务器安装在 RHEL 8.6 系统或更早的版本上）中时，这个约束是一个阻止因素。这是因为在 RHEL 9 和之前的 RHEL 版本之间没有通用的加密类型，它们通常使用 AES HMAC-SHA1 加密类型，但不使用 AES HMAC-SHA2 加密类型。

您可以通过在服务器上输入以下命令来查看 IdM 主密钥的加密类型：

```
# kadmin.local getprinc K/M | grep -E '^Key:'
```

要临时解决这个问题，在 RHEL 9 副本上启用 AES HMAC-SHA1：

```
update-crypto-policies --set FIPS:AD-SUPPORT
```

### WARNING

这个临时解决方案可能会违反 FIPS 合规性。

因此，向 IdM 部署添加 RHEL 9 副本可以正确进行。

请注意，目前有一个正在进行的工作来提供一个在 RHEL 7 和 RHEL 8 服务器上生成缺少的 AES HMAC-SHA2 加密的 Kerberos 密钥的流程。这将在 RHEL 9 副本上取得 FIPS 140-3 合规性。但是，这个过程将无法完全自动化，因为 Kerberos 密钥加密的设计无法将现有的密钥转换为不同的加密类型。唯一的方法是要求用户更新其密码。

[Bugzilla:2103327](#)

## SSSD 可正确注册 DNS 名称

在以前的版本中，如果 DNS 被错误建立，第一次尝试注册 DNS 名称时，SSSD 总是失败。要临时解决这个问题，这个更新提供了一个新的参数 `dns_resolver_use_search_list`。设置 `dns_resolver_use_search_list = false`，以避免使用 DNS 搜索列表。

[Bugzilla:1608496](#)

## 当以引用模式启动时，目录服务器会意外终止

由于一个程序错误，全局引用模式无法在 Directory Server 中工作。如果您以 `dirsrv` 用户身份使用 `refer` 选项启动 `ns-slapd` 进程，则目录服务器会忽略端口设置并意外终止。尝试以 `root` 用户身份运行进程会更改 SELinux 标签，并可以防止服务以后以正常模式启动。没有可用的临时解决方案。

[Bugzilla:2053204](#)

## 目录服务器只能从 `/var/lib/dirsrv/slapd-instance_name/ldif/` 导入 LDIF 文件

从 RHEL 8.3 开始，红帽目录服务器(MQTTs)使用自己的私有目录，并且为 LDAP 服务默认启用 `PrivateTmp` `systemd` 指令。因此，RHDS 只能从 `/var/lib/dirsrv/slapd-instance_name/ldif/` 目录导入 LDIF 文件。如果 LDIF 文件存储在不同的目录中，如 `/var/tmp/`、`/tmp` 或 `/root`，则导入会失败，并显示类似如下的错误：

```
Could not open LDIF file "/tmp/example.ldif", errno 2 (No such file or directory)
```



要临时解决这个问题，请完成以下步骤：

1. 将 LDIF 文件移到 `/var/lib/dirsrv/slapd-instance_name/ldif/` 目录中：

```
# mv /tmp/example.ldif /var/lib/dirsrv/slapd-instance_name__/ldif/
```

2. 设置允许 `dirsrv` 用户读文件的权限：

```
# chown dirsrv /var/lib/dirsrv/slapd-instance_name/ldif/example.ldif
```

3. 恢复 SELinux 上下文：

```
# restorecon -Rv /var/lib/dirsrv/slapd-instance_name/ldif/
```

如需更多信息，请参阅 [LDAP 服务无法访问主机 /tmp 和 /var/tmp 目录下的文件](#)。

[Bugzilla:2075525](#)

### 因为 EMS 强制，使用 RHEL 9.2+ IdM 服务器在 FIPS 模式下安装 RHEL 7 IdM 客户端会失败

对于启用了 FIPS 的 RHEL 9.2 及更新系统上的 TLS 1.2 连接，TLS **Extended Master Secret** (EMS) 扩展 (RFC 7627) 现在是强制的。这符合 FIPS-140-3 要求。但是，RHEL 7.9 及较低版本中提供的 `openssl` 版本不支持 EMS。因此，使用在 RHEL 9.2 及更新版本上运行的启用了 FIPS 的 IdM 服务器安装 RHEL 7 身份管理 (IdM) 客户端会失败。

如果在安装 IdM 客户端前将主机升级到 RHEL 8 不是一个选项，请通过在 FIPS 加密策略之上应用 `NO-ENFORCE-EMS` 子策略，删除 RHEL 9 服务器上 EMS 使用的要求来临时解决此问题：

```
# update-crypto-policies --set FIPS:NO-ENFORCE-EMS
```

请注意，这个删除不符合 FIPS 140-3 要求。因此，您可以建立并接受不使用 EMS 的 TLS 1.2 连接，RHEL 7 IdM 客户端的安装可以成功。

[Bugzilla:2220915](#)

## 11.12. 桌面

### 升级到 RHEL 9 后将禁用 Firefox 附加组件

如果您从 RHEL 8 升级到 RHEL 9，则之前在 Firefox 中启用的所有附加组件都会被禁用。

要临时解决这个问题，请手动重新安装或更新附加组件。因此，附加组件会如预期启用。

[Bugzilla:2013247](#)

### 升级到 RHEL 9 后，VNC 没有运行

从 RHEL 8 升级到 RHEL 9 后，VNC 服务器无法启动，即使之前启用它。

要临时解决这个问题，在系统升级后手动启用 `vncserver` 服务：

```
# systemctl enable --now vncserver@:port-number
```

现在，每个系统引导后都会启用 VNC 并按预期启动。

[Bugzilla:2060308](#)

## 用户创建屏幕没有响应

当使用图形用户界面安装 RHEL 时，用户创建屏幕没有响应。因此，在安装过程中创建用户更为困难。

要临时解决这个问题，请使用以下解决方案之一创建用户：

- 在 VNC 模式下运行安装并重新定义 VNC 窗口的大小。
- 完成安装过程后创建用户。

[BZ#2122636](#)

## 11.13. 图形基础结构

### NVIDIA 驱动程序可能会恢复到 X.org

在某些情况下，专有 NVIDIA 驱动程序会禁用 Wayland 显示协议并恢复到 X.org 显示服务器：

- 如果 NVIDIA 驱动程序的版本低于 470。
- 如果系统是使用混合图形的笔记本电脑。
- 如果您还没有启用所需的 NVIDIA 驱动程序选项。

另外，启用 Wayland，但如果 NVIDIA 驱动程序的版本低于 510，则桌面会话默认使用 X.org。

Jira:RHELPLAN-119001

### 使用 NVIDIA 在 Wayland 上无法使用 night Light

当您的系统上启用了专有 NVIDIA 驱动程序时，Wayland 会话将无法使用 GNOME 的 Night Light 功能。NVIDIA 驱动程序目前不支持 Night Light。

Jira:RHELPLAN-119852

### x.org 配置工具无法在 Wayland 下工作

用于操作屏幕的 x.org 实用程序无法在 Wayland 会话中工作。值得注意的是，**xrandr** 实用程序无法在 Wayland 下工作，因为其处理、解析、轮转和布局的不同方法。

Jira:RHELPLAN-121049

## 11.14. WEB 控制台

### VNC 控制台在某些分辨率下无法正常工作

当在某些显示分辨率下使用虚拟网络计算(VNC)控制台时，您可能会遇到鼠标偏移问题，或者您可能只看到一部分界面。因此，可能无法使用 VNC 控制台。要临时解决这个问题，您可以尝试扩展 VNC 控制台的大小，或使用控制台选项卡中的 Desktop Viewer 来启动远程查看器。

[Bugzilla:2030836](#)

## 11.15. RED HAT ENTERPRISE LINUX 系统角色

### metrics 系统角色无法用于禁用的事实收集



由于性能或其他原因，可能会在您的环境中禁用 Ansible 事实收集。在这种配置中，目前无法使用 **metrics** 系统角色。要临时解决这个问题，请启用事实缓存，或者不使用 **metrics** 系统角色（如果无法使用事实收集）。

[Bugzilla:2078999](#)

### 如果 **firewalld.service** 被屏蔽，则使用 **firewall** RHEL 系统角色失败

如果 **firewalld.service** 在 RHEL 系统上被屏蔽，则 **firewall** RHEL 系统角色失败。要临时解决这个问题，请对 **firewalld.service** 取消屏蔽：

```
systemctl unmask firewalld.service
```

[Bugzilla:2123859](#)

### 无法使用环境名称注册系统

当在 **rhc\_environment** 中指定环境名称时，**rhc** 系统角色注册系统失败。作为临时解决方案，请在注册时使用环境 ID 而不是环境名称。

[Bugzilla:2187539](#)

## 11.16. 虚拟化

### 在某些情况下，通过 **https** 或 **ssh** 安装虚拟机会失败

目前，当尝试通过 **https** 或 **ssh** 连接从 ISO 源安装客户机操作系统时，**virt-install** 工具会失败 - 例如使用 **virt-install --cdrom https://example/path/to/image.iso**。所描述的操作意外终止并显示 **internal error: process exited while connecting to monitor** 错误，而不是创建虚拟机(VM)

同样，使用 RHEL 9 web 控制台安装客户机操作系统失败，如果使用 **https** 或 **ssh** URL 或 **Download OS** 功能，则会显示 **Unknown driver 'https'** 错误。

要临时解决这个问题，请在主机上安装 **qemu-kvm-block-curl** 和 **qemu-kvm-block-ssh** 来启用 **https** 和 **ssh** 协议支持。或者，使用不同的连接协议或不同的安装源。

[Bugzilla:2014229](#)

### 在虚拟机中使用 **NVIDIA** 驱动程序会禁用 **Wayland**

目前，**NVIDIA** 驱动程序与 **Wayland** 图形会话不兼容。因此，使用 **NVIDIA** 驱动程序的 RHEL 客户机操作系统会自动禁用 **Wayland** 并加载 **Xorg** 会话。这主要在以下情况下发生：

- 当您通过 **NVIDIA** GPU 设备传递给 RHEL 虚拟机(VM)
- 当您为 RHEL 虚拟机分配 **NVIDIA** vGPU mediated 设备

[Jira:RHELPLAN-117234](#)

### 在 **AMD Milan** 系统上有时无法提供 **Milan VM CPU** 类型

在某些 **AMD Milan** 系统上，默认在 BIOS 中禁用了增强 **REP MOVSB(erm)**和 **Fast Short REP MOVSB(fsrm)**功能标记。因此，在这些系统上可能无法使用 **Milan** CPU 类型。另外，在具有不同功能标志设置的 **Milan** 主机之间的虚拟机实时迁移可能会失败。要临时解决这个问题，在主机的 BIOS 中手动打开 **erm** 和 **fsrm**。

[Bugzilla:2077767](#)

## 带有故障切换设置的 **hostdev** 接口在热拔后无法进行热插

从正在运行的虚拟机(VM)中删除带有故障切换配置的 **hostdev** 网络接口后，该接口目前无法重新连接到同一正在运行的虚拟机。

[Bugzilla:2052424](#)

## 带有故障切换 VF 的虚拟机实时复制迁移失败

目前，如果虚拟机使用启用了虚拟功能(VF)故障转移功能的设备，则试图对一个正在运行的虚拟机(VM)进行 post-copy 迁移会失败。要临时解决这个问题，请使用标准迁移类型，而不要使用 post-copy 迁移方式。

[Bugzilla:1817965](#)

## 主机网络无法在实时迁移过程中 ping 使用 VF 的虚拟机

当使用配置的虚拟功能 (VF) 实时迁移虚拟机时，如使用虚拟 SR-IOV 软件的虚拟机，虚拟机的网络不对其它设备看到，如 **ping** 之类的命令无法访问虚拟机。完成迁移后，问题将不再发生。

[Bugzilla:1789206](#)

## 在 Windows 虚拟机上不分配 IP 地址故障转移 virtio NIC

目前，当使用故障转移 virtio NIC 启动 Windows 虚拟机时，虚拟机无法为 NIC 分配 IP 地址。因此，NIC 无法设置网络连接。目前，还没有临时解决方案。

[Bugzilla:1969724](#)

## 禁用 AVX 会导致虚拟机无法引导

在使用具有高级向量扩展(AVX)支持的 CPU 的主机上，尝试引导明确禁用 AVX 的虚拟机当前会失败，并触发虚拟机中的内核 panic。

[Bugzilla:2005173](#)

## 在网络接口重置后，Windows VM 无法获取 IP 地址

有时，Windows 虚拟机在自动网络接口重置后无法获取 IP 地址。因此，虚拟机无法连接到网络。要临时解决这个问题，在 Windows 设备管理器中禁用并重新启用网络适配器驱动程序。

[Bugzilla:2084003](#)

## 实时迁移后，Broadcom 网络适配器在 Windows 虚拟机上无法正常工作

目前，广播设备（如 Broadcom、Qlogic 或 Marvell）的网络适配器无法在 Windows 虚拟机实时迁移过程中热拔。因此，在迁移完成后，适配器可以正常工作。

此问题只会影响那些使用单根 I/O 虚拟化 (SR-IOV) 附加到 Windows 虚拟机的适配器。

[Bugzilla:2090712](#), [Bugzilla:2091528](#), [Bugzilla:2111319](#)

## Windows Server 2016 虚拟机有时会在热插拔 vCPU 后停止工作

目前，将 vCPU 分配给运行 Windows Server 2016 客户机操作系统的虚拟机(VM)可能会导致各种问题，如虚拟机意外终止、变得没有响应或重启。

[Bugzilla:1915715](#)

## 使用大量队列可能会导致 Windows 虚拟机失败

当启用了虚拟可信平台模块(vTPM)设备, 且将 *multi-queue virtio-net* 功能配置为使用超过 250 个队列时, Windows 虚拟机(VM)可能会失败。

这个问题是由 vTPM 设备的限制造成的。vTPM 设备对于打开的文件描述符的最大数量有一个硬性的限制。因为会为每个新队列打开多个文件描述符, 因此可能会超过内部 vTPM 的限值, 从而导致虚拟机失败。

要临时解决这个问题, 请选择以下两个选项之一:

- 保持 vTPM 设备启用, 但使用少于 250 个队列。
- 禁用 vTPM 设备以使用超过 250 个队列。

[Bugzilla:2020146](#)

### 在具有 NVIDIA passthrough 设备的虚拟机上的冗余错误消息

当使用运行 RHEL 9.2 操作系统的 Intel 主机时, 具有 passthrough NVIDIA GPU 设备的虚拟机(VM)会频繁记录以下出错信息:

```
Spurious APIC interrupt (vector 0xFF) on CPU#2, should never happen.
```

但是, 这个错误消息不会影响虚拟机的功能, 可以忽略。详情请查看 [红帽知识库](#)。

[Bugzilla:2149989](#)

### 在带有 AMD EPYC CPU 的主机上进行 v2v 转换后, 一些 Windows 客户机不能启动

在使用 **virt-v2v** 工具将使用 Windows 11 或 Windows Server 2022 的虚拟机(VM)转换为客户机操作系统后, 虚拟机当前不能启动。这会在使用 AMD EPYC 系列 CPU 的主机上发生。

[Bugzilla:2168082](#)

### 在主机上重启 OVS 服务可能会阻止在其上运行的虚拟机的网络连接

当 Open vSwitch (OVS)服务在主机上重启时或崩溃时, 在此主机上运行的虚拟机(VM)无法恢复网络设备的状态。因此, 虚拟机可能无法完全接收数据包。

此问题只会影响在 **virtio** 网络堆栈中使用压缩 **virtqueue** 格式的系统。

要临时解决这个问题, 使用 **virtio** 网络设备定义中的 **packed=off** 参数来禁用压缩的 **virtqueue**。当禁用压缩的 **virtqueue** 时, 网络设备的状态在某些情况下可以从 RAM 中恢复。

[Bugzilla:1947422](#)

### Nvidia GPU 驱动程序在虚拟机关闭后停止工作

RHEL 内核已采用了上游 Linux 更改, 使设备电源转换延迟更接近 PCIe 规格所要求的延迟。因此, 由于 GPU 的音频功能, 一些 Nvidia GPU 可能会在关闭虚拟机后停止工作。

要临时解决这个问题, 请从虚拟机上去除 GPU 的音频功能。另外, 由于设备分配的 DMA 隔离要求 (即 IOMMU 分组), 将音频功能绑定到 **vfio-pci** 驱动程序, 这将允许 GPU 功能继续被分配, 且工作正常。

[Bugzilla:2178956](#)

### nodedev-dumpxml 无法正确列出某些介质设备的属性

目前, **nodedev-dumpxml** 无法正确列出使用 **nodedev-create** 命令创建的介质设备的属性。要临时解决这个问题, 请使用 **nodedev-define** 和 **nodedev-start** 命令。

[Bugzilla:2143158](#)

### 恢复中断的复制后虚拟机迁移可能会失败

如果虚拟机(VM)的复制后迁移中断，然后在同一传入端口上立即恢复，则迁移可能会失败，并显示以下错误 **Address already in use**

要临时解决这个问题，请在恢复后复制迁移或切换到迁移恢复的另一个端口前至少等待 10 秒。

[Bugzilla:2178376](#)

### 重启 virtqemud 或 libvirtd 后，无法附加 virtiofs 设备

目前，重启 **virtqemud** 或 **libvirtd** 服务会防止 **virtiofs** 存储设备附加到主机上的虚拟机。

[Bugzilla:2078693](#)

### virsh blkio tune --weight 命令无法设置正确的 cgroup I/O 控制器值

目前，使用 **virsh blkio tune --weight** 命令设置 VM 权重无法按预期工作。该命令无法在 cgroup I/O 控制器接口文件中设置正确的 **io.bfq.weight** 值。目前还没有临时解决方案。

Jira:RHELPLAN-83423

### 将 Watchdog 卡热插到虚拟机失败

目前，如果没有可用的 PCI 插槽，向正在运行的虚拟机(VM)添加 Watchdog 卡会失败，并显示以下错误：

```
Failed to configure watchdog
ERROR Error attempting device hotplug: internal error: No more available PCI slots
```

要临时解决这个问题，请在添加 Watchdog 卡前关闭虚拟机。

[Bugzilla:2173584](#)

### NUMA 节点映射在 AMD EPYC CPU 上无法正常工作

QEMU 无法正确处理 AMD EPYC CPU 上的 NUMA 节点映射。因此，如果使用 NUMA 节点配置，具有这些 CPU 的虚拟机(VM)的性能可能会受到负面影响。另外，虚拟机在启动过程中会显示类似如下的警告。

```
sched: CPU #4's llc-sibling CPU #3 is not on the same node! [node: 1 != 0]. Ignoring dependency.
WARNING: CPU: 4 PID: 0 at arch/x86/kernel/smpboot.c:415 topology_sane.isra.0+0x6b/0x80
```

要临时解决这个问题，请不要将 AMD EPYC CPU 用于 NUMA 节点配置。

[Bugzilla:2176010](#)

### 虚拟机迁移过程中的 NFS 故障会导致迁移失败和源虚拟机 coredump

目前，如果 NFS 服务或服务器在虚拟机(VM)迁移过程中关闭，则源虚拟机的 QEMU 在重新开始运行时无法重新连接到 NFS 服务器。因此，迁移会失败，并在源虚拟机上发起 coredump。目前，还没有可用的临时解决方案。

[Bugzilla:2058982](#)

### PCIe ATS 设备无法在 Windows 虚拟机上工作

当您在带有 Windows 客户机操作系统的虚拟机的 XML 配置中配置 PCIe 地址转换服务(ATS)设备时，在引导虚拟机后，客户机不会启用 ATS 设备。这是因为 Windows 目前不支持 **virtio** 设备上的 ATS。

[Bugzilla:2073872](#)

### kdump 在带有 AMD SEV-SNP 的虚拟机上失败

目前，kdump 在使用带有 Secure Nested Paging (SNP) 功能的 AMD Secure Encrypted Virtualization (SEV) 的 RHEL 9 虚拟机(VM)上失败。

Jira:RHEL-10019

## 11.17. 云环境中的 RHEL

### 在 Nutanix AHV 中使用 LVM 克隆或恢复 RHEL 9 虚拟机会导致非 root 分区消失

当在 Nutanix AHV 虚拟机监控程序上托管的虚拟机中运行 RHEL 9 客户机操作系统时，从快照中恢复虚拟机或克隆虚拟机目前会导致虚拟机中的非 root 分区在虚拟机中使用逻辑卷管理(LVM)时消失。因此，会出现以下问题：

- 从快照恢复虚拟机后，虚拟机无法引导，而是进入紧急模式。
- 通过克隆创建的虚拟机无法引导，而是进入紧急模式。

要临时解决这个问题，在虚拟机的紧急模式下执行以下操作：

1. 删除 LVM 系统设备文件：**rm /etc/lvm/devices/system.devices**
2. 重新创建 LVM 设备设置：**vgimportdevices -a**
3. 重启虚拟机

这样，克隆或恢复的虚拟机可以正确引导。

另外，为了避免这个问题发生，请在克隆虚拟机或创建虚拟机快照前进行以下操作：

1. 在 **/etc/lvm/lvm.conf** 文件中取消注释 **use\_devicesfile = 0** 行
2. 重启虚拟机

[Bugzilla:2059545](#)

### 在 ESXi 上自定义 RHEL 9 客户机有时会导致网络问题

目前，在 VMware ESXi hypervisor 中自定义 RHEL 9 客户机操作系统无法正常工作。因此，如果客户机使用这样的密钥文件，它有不正确的网络设置，如 IP 地址或网关。

有关详情和临时解决方案说明，请参阅 [VMware 知识库](#)。

[Bugzilla:2037657](#)

### 如果 RHEL 实例是由 cloud-init 提供的，且使用 NFSv3 挂载条目配置的，则其在 Azure 上无法引导

目前，如果 VM 是由 **cloud-init** 工具提供的，且虚拟机的客户机操作系统在 **/etc/fstab** 文件中有 NFSv3 挂载条目，则在 Microsoft Azure 云平台上引导 RHEL 虚拟机(VM)会失败。

[Bugzilla:2081114](#)

## 在 VMware 主机上的 RHEL 虚拟机中设置静态 IP 无法正常工作

目前，当使用 RHEL 作为 VMware 主机上虚拟机(VM)的客户机操作系统时，DatasourceOVF 功能无法正常工作。因此，如果您使用 **cloud-init** 实用程序将虚拟机的网络设置为静态 IP，然后重启虚拟机，则虚拟机的网络将更改为 DHCP。

要临时解决这个问题，请查看 [VMware 知识库](#)。

[Bugzilla:1750862](#)

## 11.18. 支持性

### 在 IBM Power Systems Little Endian 上运行 **sos report** 时超时

当在具有带有数百或数千个 CPU 的 IBM Power Systems, Little Endian 上运行 **sos report** 命令时，处理器插件会在收集 `/sys/devices/system/cpu` 目录的大量内容时达到默认的 300 秒超时时间。作为临时解决方案，请相应地增加插件的超时时间：

- 对于一次性设置，请运行：

```
# sos report -k processor.timeout=1800
```

- 对于永久性更改，请编辑 `/etc/sos/sos.conf` 文件的 `[plugin_options]` 部分：

```
[plugin_options]
# Specify any plugin options and their values here. These options take the form
# plugin_name.option_name = value
#rpm.rpmva = off
processor.timeout = 1800
```

示例值设为 1800。特定的超时值高度依赖于特定的系统。要适当地设置插件超时，您可以首先通过运行以下命令来估算收集一个没有超时的插件所需的时间：

```
# time sos report -o processor -k processor.timeout=0 --batch --build
```

[Bugzilla:1869561](#)

## 11.19. 容器

### 在较旧的容器镜像中运行 **systemd** 无法正常工作

在较旧的容器镜像（如 **centos:7**）中运行 **systemd** 将无法正常工作：

```
$ podman run --rm -ti centos:7 /usr/lib/systemd/systemd
Storing signatures
Failed to mount cgroup at /sys/fs/cgroup/systemd: Operation not permitted
[!!!!!!] Failed to mount API filesystems, freezing.
```

要临时解决这个问题，请使用以下命令：

```
# mkdir /sys/fs/cgroup/systemd
# mount none -t cgroup -o none,name=systemd /sys/fs/cgroup/systemd
# podman run --runtime /usr/bin/crun --annotation=run.oci.systemd.force_cgroup_v1=/sys/fs/cgroup -
-rm -ti centos:7 /usr/lib/systemd/systemd
```

Jira:RHELPLAN-96940



## 附录 A. 按组件划分的问题单列表

在本文档中列出了 Bugzilla 和 JIRA 问题单以供参考。这些链接会指向本文档中描述问题单的发行注记。

| 组件                                      | 票证   |
|---|--|
| <b>389-ds-base</b>                      | <a href="#">Bugzilla:2096795</a> , <a href="#">Bugzilla:1859271</a> , <a href="#">Bugzilla:2057070</a> , <a href="#">Bugzilla:2093981</a> , <a href="#">Bugzilla:1132524</a> , <a href="#">Bugzilla:2136610</a> , <a href="#">Bugzilla:2142639</a> , <a href="#">Bugzilla:1878808</a> , <a href="#">Bugzilla:1924569</a> , <a href="#">Bugzilla:1956987</a> , <a href="#">Bugzilla:1952241</a> , <a href="#">Bugzilla:2063140</a> , <a href="#">Bugzilla:2047175</a> , <a href="#">Bugzilla:2053204</a>                                    |
| <b>Doc-administration-guide</b>         | <a href="#">Bugzilla:2075525</a>   |
| <b>NetworkManager</b>                   | <a href="#">Bugzilla:2134897</a> , <a href="#">Bugzilla:2081302</a> , <a href="#">Bugzilla:2019306</a> , <a href="#">Bugzilla:2128809</a> , <a href="#">Bugzilla:2110307</a> , <a href="#">Bugzilla:2117352</a> , <a href="#">Bugzilla:2029636</a> , <a href="#">Bugzilla:2073512</a> , <a href="#">Bugzilla:2128216</a> , <a href="#">Bugzilla:1894877</a> , <a href="#">Bugzilla:2151040</a>   |
| <b>aardvark-dns</b>                     | <a href="#">Jira:RHELPLAN-138024</a>   |
| <b>anaconda</b>                         | <a href="#">Bugzilla:2052938</a> , <a href="#">Bugzilla:2158210</a> , <a href="#">Bugzilla:1991843</a> , <a href="#">Bugzilla:2127100</a> , <a href="#">Bugzilla:2093793</a> , <a href="#">Bugzilla:2107346</a> , <a href="#">Bugzilla:2050140</a> , <a href="#">Bugzilla:1877697</a> , <a href="#">Bugzilla:1914955</a> , <a href="#">Bugzilla:1929105</a> , <a href="#">Bugzilla:1997832</a> , <a href="#">Bugzilla:2125542</a> , <a href="#">Bugzilla:2115783</a> , <a href="#">Bugzilla:2164216</a> , <a href="#">Bugzilla:2163497</a> |
| <b>ansible-collection-microsoft-sql</b> | <a href="#">Bugzilla:2151282</a> , <a href="#">Bugzilla:2151283</a> , <a href="#">Bugzilla:2151284</a> , <a href="#">Bugzilla:2153428</a> , <a href="#">Bugzilla:2163709</a>   |
| <b>ansible-freeipa</b>                  | <a href="#">Bugzilla:2127913</a>   |
| <b>bacula</b>                           | <a href="#">Bugzilla:2089395</a>   |
| <b>bind</b>                             | <a href="#">Bugzilla:1984982</a>   |
| <b>chrony</b>                           | <a href="#">Bugzilla:2133754</a>   |
| <b>clevis</b>                           | <a href="#">Bugzilla:2126533</a> , <a href="#">Bugzilla:2159728</a> , <a href="#">Bugzilla:2159735</a>   |
| <b>cloud-init</b>                       | <a href="#">Bugzilla:1750862</a>   |
| <b>cockpit</b>                          | <a href="#">Bugzilla:2207498</a>   |
| <b>cockpit-appstream</b>                | <a href="#">Bugzilla:2030836</a>   |
| <b>cockpit-machines</b>                 | <a href="#">Bugzilla:2173584</a>   |
| <b>contrack-tools</b>                   | <a href="#">Bugzilla:2132398</a>   |
| <b>crash</b>                            | <a href="#">Bugzilla:2119685</a>   |



| 组件                             | 票证  |
|--------------------------------|---|
| <b>crypto-policies</b>         | <a href="#">Bugzilla:2152635</a>  |
| <b>cyrus-sasl</b>              | <a href="#">Bugzilla:1995600</a>  |
| <b>device-mapper-multipath</b> | <a href="#">Bugzilla:2033080</a> , <a href="#">Bugzilla:2011699</a> , <a href="#">Bugzilla:1926147</a>                                    |
| <b>dnf</b>                     | <a href="#">Bugzilla:2131288</a> , <a href="#">Bugzilla:2121662</a> , <a href="#">Bugzilla:2122626</a> , <a href="#">Bugzilla:2073510</a> |
| <b>dnf-plugins-core</b>        | <a href="#">Bugzilla:2139326</a>  |
| <b>edk2</b>                    | <a href="#">Bugzilla:1935497</a>  |
| <b>fapolicyd</b>               | <a href="#">Jira:RHEL-192</a> , <a href="#">Bugzilla:2054740</a> , <a href="#">Bugzilla:2070655</a>                                       |
| <b>firefox</b>                 | <a href="#">Bugzilla:2013247</a>  |
| <b>firewalld</b>               | <a href="#">Bugzilla:2125371</a> , <a href="#">Bugzilla:2077512</a> , <a href="#">Bugzilla:2122678</a>                                    |
| <b>frr</b>                     | <a href="#">Bugzilla:2129731</a> , <a href="#">Bugzilla:2129743</a>   |
| <b>gcc</b>                     | <a href="#">Bugzilla:2110583</a> , <a href="#">Bugzilla:2117632</a> , <a href="#">Bugzilla:2141718</a>                                    |
| <b>gdm</b>                     | <a href="#">Bugzilla:2131203</a>  |
| <b>gimp</b>                    | <a href="#">Bugzilla:2047161</a>  |
| <b>git</b>                     | <a href="#">Bugzilla:2139379</a>  |
| <b>git-lfs</b>                 | <a href="#">Bugzilla:2139383</a>  |
| <b>glibc</b>                   | <a href="#">Bugzilla:2129005</a> , <a href="#">Bugzilla:2155352</a>   |
| <b>gnome-shell-extensions</b>  | <a href="#">Bugzilla:2154358</a> , <a href="#">Bugzilla:2160553</a>   |
| <b>gnupg2</b>                  | <a href="#">Bugzilla:2070722</a> , <a href="#">Bugzilla:2073567</a>   |
| <b>gnutls</b>                  | <a href="#">Bugzilla:2084161</a> , <a href="#">Bugzilla:2042009</a>   |
| <b>golang</b>                  | <a href="#">Bugzilla:2133019</a> , <a href="#">Bugzilla:2175173</a> , <a href="#">Bugzilla:2111072</a> , <a href="#">Bugzilla:2092016</a> |
| <b>grafana</b>                 | <a href="#">Bugzilla:2116847</a>  |
| <b>grafana-pcp</b>             | <a href="#">Bugzilla:2116848</a>  |

| 组件                          | 票证   |
|-----------------------------|--|
| <b>grub2</b>                | <a href="#">Bugzilla:2026579</a>   |
| <b>grubby</b>               | <a href="#">Bugzilla:2127453</a>   |
| <b>gssproxy</b>             | <a href="#">Bugzilla:2184333</a>   |
| <b>ipa</b>                  | <a href="#">Bugzilla:2143224</a> , <a href="#">Bugzilla:2162677</a> , <a href="#">Bugzilla:2084180</a> , <a href="#">Bugzilla:2084166</a> , <a href="#">Bugzilla:2069202</a> , <a href="#">Bugzilla:2094673</a> , <a href="#">Bugzilla:2057471</a> , <a href="#">Bugzilla:2124243</a> , <a href="#">Bugzilla:2089907</a>   |
| <b>iproute</b>              | <a href="#">Bugzilla:2155604</a>   |
| <b>java-1.8.0-openjdk</b>   | <a href="#">Bugzilla:2188023</a>   |
| <b>java-17-openjdk</b>      | <a href="#">Bugzilla:2186803</a> , <a href="#">Bugzilla:2186810</a> , <a href="#">Bugzilla:2186806</a>   |
| <b>jmc</b>                  | <a href="#">Bugzilla:2122401</a>   |
| <b>jmc-core</b>             | <a href="#">Bugzilla:1980981</a>   |
| <b>kdump-anaconda-addon</b> | <a href="#">Bugzilla:2017401</a>   |
| <b>kernel</b>               | <a href="#">Bugzilla:2153073</a> , <a href="#">Bugzilla:2143850</a> , <a href="#">Bugzilla:1871126</a> , <a href="#">Bugzilla:1871143</a> , <a href="#">Bugzilla:2075216</a> , <a href="#">Bugzilla:2100606</a> , <a href="#">Bugzilla:2104468</a> , <a href="#">Bugzilla:2111048</a> , <a href="#">Bugzilla:2150284</a> , <a href="#">Bugzilla:2066372</a> , <a href="#">Bugzilla:2107347</a> , <a href="#">Bugzilla:2140899</a> , <a href="#">Bugzilla:2069758</a> , <a href="#">Bugzilla:1613522</a> , <a href="#">Bugzilla:1874182</a> , <a href="#">Bugzilla:1995338</a> , <a href="#">Bugzilla:1570255</a> , <a href="#">Bugzilla:2023416</a> , <a href="#">Bugzilla:2021672</a> , <a href="#">Bugzilla:2021672</a> , <a href="#">Bugzilla:2027304</a> , <a href="#">Bugzilla:1660337</a> , <a href="#">Bugzilla:1955275</a> , <a href="#">Bugzilla:2142102</a> , <a href="#">Bugzilla:2041690</a> , <a href="#">Bugzilla:2040643</a> , <a href="#">Bugzilla:2167783</a> , <a href="#">Bugzilla:2000616</a> , <a href="#">Bugzilla:2013650</a> , <a href="#">Bugzilla:2132480</a> , <a href="#">Bugzilla:2059545</a> , <a href="#">Bugzilla:1960467</a> , <a href="#">Bugzilla:2005173</a> , <a href="#">Bugzilla:2128610</a> , <a href="#">Bugzilla:2129288</a> , <a href="#">Bugzilla:2013884</a> , <a href="#">Bugzilla:2149989</a> , <a href="#">Bugzilla:2168603</a> , <a href="#">Bugzilla:2168603</a> , <a href="#">Bugzilla:2173947</a> , <a href="#">Bugzilla:2178956</a> , <a href="#">Bugzilla:2180665</a> , <a href="#">JIRA:RHEL-6496</a> |
| <b>kexec-tools</b>          | <a href="#">Bugzilla:2085347</a> , <a href="#">Bugzilla:2076416</a> , <a href="#">Bugzilla:2160676</a> , <a href="#">Bugzilla:2080110</a> , <a href="#">Bugzilla:2139000</a> , <a href="#">Bugzilla:2113873</a> , <a href="#">Bugzilla:2064708</a> , <a href="#">Bugzilla:2065013</a>  |
| <b>keylime</b>              | <a href="#">Bugzilla:2150830</a> , <a href="#">Bugzilla:2138167</a> , <a href="#">Bugzilla:2140670</a> , <a href="#">Bugzilla:2142009</a>  |
| <b>kmod</b>                 | <a href="#">Bugzilla:2103605</a>   |
| <b>krb5</b>                 | <a href="#">Bugzilla:2068535</a> , <a href="#">Bugzilla:2106043</a> , <a href="#">Bugzilla:2060798</a> , <a href="#">Bugzilla:2077450</a> , <a href="#">Bugzilla:2106296</a> , <a href="#">Bugzilla:2060421</a> , <a href="#">Bugzilla:2016312</a> , <a href="#">Bugzilla:2103327</a>  |

| 组件                   | 票证   |
|----------------------|--|
| <b>libdnf</b>        | <a href="#">Bugzilla:2124480</a>   |
| <b>libnvme</b>       | <a href="#">Bugzilla:2139752</a>   |
| <b>libotr</b>        | <a href="#">Bugzilla:2086562</a>   |
| <b>libreswan</b>     | <a href="#">Bugzilla:2128669</a>   |
| <b>libsepol</b>      | <a href="#">Bugzilla:2145224</a>   |
| <b>libssh</b>        | <a href="#">Bugzilla:2026449</a> , <a href="#">Bugzilla:2068475</a>  |
| <b>libvirt</b>       | <a href="#">Bugzilla:2014487</a> , <a href="#">Bugzilla:2143158</a> , <a href="#">Bugzilla:2078693</a>   |
| <b>libxcrypt</b>     | <a href="#">Bugzilla:2034569</a>   |
| <b>llvm-toolset</b>  | <a href="#">Bugzilla:2118567</a>   |
| <b>lvm2</b>          | <a href="#">Bugzilla:1878893</a> , <a href="#">Bugzilla:2038183</a>  |
| <b>mod_security</b>  | <a href="#">Bugzilla:2143211</a>   |
| <b>mysql</b>         | <a href="#">Bugzilla:1991500</a>   |
| <b>nfs-utils</b>     | <a href="#">Bugzilla:2143747</a> , <a href="#">Bugzilla:2081114</a>  |
| <b>nginx</b>         | <a href="#">Bugzilla:2096174</a>   |
| <b>nmstate</b>       | <a href="#">Bugzilla:2095207</a> , <a href="#">Bugzilla:2120473</a> , <a href="#">Bugzilla:2044150</a> , <a href="#">Bugzilla:2058292</a> ,<br><a href="#">Bugzilla:2130240</a> , <a href="#">Bugzilla:2162401</a> |
| <b>nodejs</b>        | <a href="#">Bugzilla:2178088</a>   |
| <b>nss</b>           | <a href="#">Bugzilla:2091905</a>   |
| <b>nvme-cli</b>      | <a href="#">Bugzilla:2139753</a>   |
| <b>nvme-stas</b>     | <a href="#">Bugzilla:1893841</a>   |
| <b>open-vm-tools</b> | <a href="#">Bugzilla:2037657</a>   |
| <b>openblas</b>      | <a href="#">Bugzilla:2112099</a> , <a href="#">Bugzilla:2115737</a>  |
| <b>opencryptoki</b>  | <a href="#">Bugzilla:2110314</a>   |

| 组件                          | 票证   |
|-----------------------------|--|
| <b>openscap</b>             | <a href="#">Bugzilla:2159286</a> , <a href="#">Bugzilla:2161499</a>  |
| <b>openslp</b>              | <a href="#">Bugzilla:2184570</a>   |
| <b>openssh</b>              | <a href="#">Bugzilla:2056884</a>   |
| <b>openssl</b>              | <a href="#">Bugzilla:2129063</a> , <a href="#">Bugzilla:2188046</a> , <a href="#">Bugzilla:2060044</a> , <a href="#">Bugzilla:1975836</a> , <a href="#">Bugzilla:2168665</a> , <a href="#">Bugzilla:1681178</a> , <a href="#">Bugzilla:1685470</a> |
| <b>openssl-ibmca</b>        | <a href="#">Bugzilla:2110378</a>   |
| <b>osbuild-composer</b>     | <a href="#">Bugzilla:2173928</a>   |
| <b>oscap-anaconda-addon</b> | <a href="#">Bugzilla:2165920</a> , <a href="#">Bugzilla:2172264</a>  |
| <b>pacemaker</b>            | <a href="#">Bugzilla:2133546</a> , <a href="#">Bugzilla:2125344</a> , <a href="#">Bugzilla:2125337</a>   |
| <b>pam</b>                  | <a href="#">Bugzilla:2126640</a>   |
| <b>passt</b>                | <a href="#">Bugzilla:2131015</a>   |
| <b>pause-container</b>      | <a href="#">Bugzilla:2106816</a>   |
| <b>pcp</b>                  | <a href="#">Bugzilla:2117074</a>   |
| <b>pcs</b>                  | <a href="#">Bugzilla:2116295</a> , <a href="#">Bugzilla:2112270</a> , <a href="#">Bugzilla:1620043</a> , <a href="#">Bugzilla:1796827</a> , <a href="#">Bugzilla:2092950</a>   |
| <b>pki-core</b>             | <a href="#">Bugzilla:1849834</a>   |
| <b>podman</b>               | <a href="#">Jira:RHELPLAN-136602</a> , <a href="#">Jira:RHELPLAN-136607</a> , <a href="#">Bugzilla:2119200</a> , <a href="#">Jira:RHELPLAN-136611</a> , <a href="#">Bugzilla:2069279</a>   |
| <b>postgresql</b>           | <a href="#">Bugzilla:2128410</a>   |
| <b>powerpc-utils</b>        | <a href="#">Bugzilla:2125152</a>   |
| <b>powertop</b>             | <a href="#">Bugzilla:2044132</a>   |
| <b>python-blivet</b>        | <a href="#">Bugzilla:2103800</a>   |
| <b>python-sqlalchemy</b>    | <a href="#">Bugzilla:2152649</a>   |
| <b>python3.11</b>           | <a href="#">Bugzilla:2127923</a>   |

| 组件                         | 票证  |
|----------------------------|---|
| <b>python3.11-lxml</b>     | <a href="#">Bugzilla:2157708</a>  |
| <b>qemu-kvm</b>            | <a href="#">Bugzilla:2116496</a> , <a href="#">Bugzilla:1965079</a> , <a href="#">Bugzilla:1951814</a> , <a href="#">Bugzilla:2060839</a> , <a href="#">Bugzilla:2014229</a> , <a href="#">Bugzilla:2052424</a> , <a href="#">Bugzilla:1817965</a> , <a href="#">Bugzilla:1789206</a> , <a href="#">Bugzilla:2090712</a> , <a href="#">Bugzilla:1915715</a> , <a href="#">Bugzilla:2020146</a> , <a href="#">Bugzilla:1947422</a> , <a href="#">Bugzilla:2178376</a> , <a href="#">Bugzilla:2176010</a> , <a href="#">Bugzilla:2058982</a>  |
| <b>realtime-tests</b>      | <a href="#">Bugzilla:2041637</a>  |
| <b>rear</b>                | <a href="#">Bugzilla:2172589</a> , <a href="#">Bugzilla:2160748</a>   |
| <b>restore</b>             | <a href="#">Bugzilla:1997366</a>  |
| <b>rhel-system-roles</b>   | <a href="#">Bugzilla:2131293</a> , <a href="#">Bugzilla:2133858</a> , <a href="#">Bugzilla:2078999</a> , <a href="#">Bugzilla:2119102</a> , <a href="#">Bugzilla:2128843</a> , <a href="#">Bugzilla:2130010</a> , <a href="#">Bugzilla:2130329</a> , <a href="#">Bugzilla:2130344</a> , <a href="#">Bugzilla:2130357</a> , <a href="#">Bugzilla:2133528</a> , <a href="#">Bugzilla:2133930</a> , <a href="#">Bugzilla:2134202</a> , <a href="#">Bugzilla:2137663</a> , <a href="#">Bugzilla:2140795</a> , <a href="#">Bugzilla:2141330</a> , <a href="#">Bugzilla:2143768</a> , <a href="#">Bugzilla:2165175</a> , <a href="#">Bugzilla:2140804</a> , <a href="#">Bugzilla:2126959</a> , <a href="#">Bugzilla:2143816</a> , <a href="#">Bugzilla:2153030</a> , <a href="#">Bugzilla:2153043</a> , <a href="#">Bugzilla:2162782</a> , <a href="#">Bugzilla:2167528</a> , <a href="#">Bugzilla:2168735</a> , <a href="#">Bugzilla:2160152</a> , <a href="#">Bugzilla:1999770</a> , <a href="#">Bugzilla:2123859</a> , <a href="#">Bugzilla:2187539</a> , <a href="#">Bugzilla:2186218</a> |
| <b>rpm</b>                 | <a href="#">Bugzilla:2150804</a> , <a href="#">Bugzilla:2111251</a> , <a href="#">Bugzilla:2144005</a>  |
| <b>rsyslog</b>             | <a href="#">Bugzilla:2124849</a> , <a href="#">Bugzilla:2127404</a> , <a href="#">Bugzilla:2124488</a> , <a href="#">Bugzilla:2157659</a>   |
| <b>rteval</b>              | <a href="#">Bugzilla:2081325</a>  |
| <b>rust</b>                | <a href="#">Bugzilla:2123900</a>  |
| <b>s390utils</b>           | <a href="#">Bugzilla:2044204</a> , <a href="#">Bugzilla:1932480</a>   |
| <b>samba</b>               | <a href="#">Bugzilla:2131993</a> , <a href="#">Jira:RHELDPCS-16612</a>  |
| <b>scap-security-guide</b> | <a href="#">Bugzilla:2158405</a> , <a href="#">Bugzilla:2122325</a> , <a href="#">Bugzilla:2169414</a> , <a href="#">Bugzilla:2105162</a> , <a href="#">Bugzilla:2120978</a> , <a href="#">Bugzilla:2038978</a>   |
| <b>selinux-policy</b>      | <a href="#">Bugzilla:2151841</a> , <a href="#">Bugzilla:1972222</a> , <a href="#">Bugzilla:2064274</a>  |
| <b>scs</b>                 | <a href="#">Bugzilla:2164987</a> , <a href="#">Bugzilla:2134906</a> , <a href="#">Bugzilla:1869561</a>  |
| <b>sssd</b>                | <a href="#">Bugzilla:1507035</a> , <a href="#">Bugzilla:2087247</a> , <a href="#">Bugzilla:1766490</a> , <a href="#">Bugzilla:2065693</a> , <a href="#">Bugzilla:2056482</a> , <a href="#">Bugzilla:1608496</a>   |

| 组件                          | 票证   |
|-----------------------------|--|
| <b>stratisd</b>             | <a href="#">Bugzilla:2039957</a> , <a href="#">Bugzilla:2039955</a> , <a href="#">Bugzilla:2041558</a> |
| <b>subscription-manager</b> | <a href="#">Bugzilla:2108549</a> , <a href="#">Bugzilla:2163716</a> , <a href="#">Bugzilla:2136694</a> |
| <b>swig</b>                 | <a href="#">Bugzilla:2139101</a>   |
| <b>synce4l</b>              | <a href="#">Bugzilla:2143264</a>   |
| <b>systemd</b>              | <a href="#">Bugzilla:2217931</a> , <a href="#">Bugzilla:2018112</a>                                    |
| <b>systemtap</b>            | <a href="#">Bugzilla:2083727</a>   |
| <b>tang</b>                 | <a href="#">Bugzilla:2095474</a> , <a href="#">Bugzilla:2188743</a>                                    |
| <b>tigervnc</b>             | <a href="#">Bugzilla:2060308</a>   |
| <b>tomcat</b>               | <a href="#">Bugzilla:2160511</a>   |
| <b>toolbox</b>              | <a href="#">Bugzilla:2163752</a>   |
| <b>tuna</b>                 | <a href="#">Bugzilla:2122781</a> , <a href="#">Bugzilla:2121517</a> , <a href="#">Bugzilla:2062865</a> |
| <b>tuned</b>                | <a href="#">Bugzilla:2133815</a> , <a href="#">Bugzilla:2113900</a>                                    |
| <b>tzdata</b>               | <a href="#">Bugzilla:2157982</a>   |
| <b>udisks2</b>              | <a href="#">Bugzilla:1983602</a>   |
| <b>unbound</b>              | <a href="#">Bugzilla:2070495</a>   |
| <b>usbguard</b>             | <a href="#">Bugzilla:2155910</a> , <a href="#">Bugzilla:2042345</a> , <a href="#">Bugzilla:2097419</a> |
| <b>virt-v2v</b>             | <a href="#">Bugzilla:2168082</a>   |
| <b>virtio-win</b>           | <a href="#">Bugzilla:1969724</a> , <a href="#">Bugzilla:2084003</a>                                    |
| <b>vsftpd</b>               | <a href="#">Bugzilla:2018284</a>   |
| <b>wsmancli</b>             | <a href="#">Bugzilla:2127416</a>   |
| <b>xdp-tools</b>            | <a href="#">Bugzilla:2160066</a>   |

| 组件 | 票证   |
|----|--|
| 其他 | <p>Bugzilla:2177782, Jira:RHELPLAN-137505, Jira:RHELPLAN-139125, Bugzilla:2046653, Jira:RHELPLAN-133650, Jira:RHELPLAN-139430, Jira:RHELPLAN-137416, Jira:RHELPLAN-137411, Jira:RHELPLAN-137406, Jira:RHELPLAN-137403, Jira:RHELPLAN-159146, Jira:RHELPLAN-139448, Jira:RHELPLAN-151481, Jira:RHELPLAN-150266, Jira:RHELPLAN-147982, Jira:RHELPLAN-147428, Jira:RHELPLAN-139659, Jira:RHELPLAN-149091, Jira:RHELPLAN-139655, Jira:RHELPLAN-139424, Jira:RHELPLAN-136489, Jira:RHELPLAN-59528, Bugzilla:2209419, Bugzilla:2190123, Jira:RHELPLAN-135600, Jira:RHELPLAN-148303, Bugzilla:2020529, Bugzilla:2030412, Jira:RHELPLAN-103993, Jira:RHELPLAN-122345, Jira:RHELPLAN-27394, Jira:RHELPLAN-27737, Jira:RHELPLAN-148394, Bugzilla:1927780, Jira:RHELPLAN-110763, Bugzilla:1935544, Bugzilla:2089200, Jira:RHELPLAN-15509, Jira:RHELPLAN-99136, Jira:RHELPLAN-103232, Bugzilla:1899167, Bugzilla:1979521, Jira:RHELPLAN-100087, Jira:RHELPLAN-100639, Bugzilla:2058153, Jira:RHELPLAN-113995, Jira:RHELPLAN-98983, Jira:RHELPLAN-131882, Jira:RHELPLAN-137660, Jira:RHELPLAN-139805, Jira:RHELPLAN-147725, Jira:RHELPLAN-153267, Jira:RHELDOCS-16300, Jira:RHELPLAN-157225, Jira:RHELPLAN-157337, Bugzilla:1640697, Bugzilla:1697896, Bugzilla:2047713, Jira:RHELPLAN-96940, Jira:RHELPLAN-117234, Jira:RHELPLAN-119001, Jira:RHELPLAN-119852, Bugzilla:2077767, Bugzilla:2053598, Bugzilla:2082303, Jira:RHELPLAN-121049, Jira:RHELPLAN-157939, Jira:RHELPLAN-109613, Bugzilla:2160619, Bugzilla:2173992, Bugzilla:2185048, Jira:RHELPLAN-83423</p> |

## 附录 B. 修订历史

### 0.3-8

2024 年 6 月 11 日星期二, Brian Angelica ([bangelic@redhat.com](mailto:bangelic@redhat.com))

- 添加已弃用的功能 [RHELDOCS-18049](#) (Shells 和命令行工具)。

### 0.3-7

2024 年 6 月 11 日星期二, Brian Angelica ([bangelic@redhat.com](mailto:bangelic@redhat.com))

- 添加了一个已知问题 [RHEL-24847](#) (Shells 和命令行工具)。

### 0.3-6

2024 年 5 月 16 日星期四, Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 添加了一个已知问题 [RHEL-10019](#) (虚拟化)。

### 0.3-5

2024 年 4 月 25 日星期四, Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 添加了一个改进 [BZ#2136610](#) (身份管理)。

### 0.3-4

2024 年 4 月 18 日星期四, Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 添加了一个改进 [RHEL-19142](#) (网络)。

### 0.3-3

2024 年 3 月 14 日星期四, Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 添加了一个改进 [RHEL-18359](#) (内核)。
- 添加了一个已知问题 [RHEL-25967](#) (内核)。

### 0.3-2

2024 年 3 月 4 日星期一, Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 添加了一个 Bug 修复 [JIRA:SSSD-6096](#) (身份管理)。

### 0.3.1

2024 年 2 月 1 日星期四, Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 添加了一个已知问题 [BZ#1834716](#) (安全)。

### 0.3-0

2024 年 1 月 12 日星期五, Marc Muehlfeld ([mmuehlfeld@redhat.com](mailto:mmuehlfeld@redhat.com))

- 添加了一个已知问题 [JIRA:RHEL-6496](#) (网络)。

### 0.2-9

2023 年 12 月 12 日星期二, Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 添加了一个技术预览 [BZ#2166771](#) (安全)。



- 添加了一个技术预览 [BZ#2162677](#)(IdM)。

## 0.2-8

2023 年 12 月 7 日星期四, Lucie Vařáková([lvarakova@redhat.com](mailto:lvarakova@redhat.com))

- 添加了一个新功能 [BZ#2044200](#) (内核)。

## 0.2-7

2023 年 11 月 20 日星期一, Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 添加了一个增强 [BZ#2165827](#) (身份管理)。

## 0.2-6

2023 年 11 月 13 日星期一, Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 添加了一个技术预览 [JIRA:RHELDOCS-17040](#) (虚拟化)

## 0.2-5

2023 年 11 月 10 日星期五, Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 更新了对 RHEL 文档提供反馈的模块。

## 0.2-4

2023 年 11 月 10 日星期五, Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 添加了一个技术预览 [JIRA:RHELDOCS-17050](#) (虚拟化)。

## 0.2-3

2023 年 11 月 2 日星期四, Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 更新了 [BZ#2125371](#)(网络)中的文档文本。

## 0.2-2

2023 年 10 月 13 日星期五, Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 添加了一个技术预览 [JIRA:RHELDOCS-16861](#) (容器)。

## 0.2-1

2023 年 9 月 25 日, Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 添加了一个已知问题 [BZ#2122636](#) (桌面)。

## 0.2-0

2023 年 9 月 13 日, Lenka Špačková ([lspackova@redhat.com](mailto:lspackova@redhat.com))

- 修复了 [BZ#2220915](#) 中的命令格式问题。

## 0.1-9

2023 年 9 月 8 日, Marc Muehlfeld ([mmuehlfeld@redhat.com](mailto:mmuehlfeld@redhat.com))

- 添加了一个已弃用的功能发行注记 [JIRA:RHELDOCS-16612](#) (Samba)。

- 更新了 [对红帽文档提供反馈](#) 部分。

#### 0.1-8

2023 年 9 月 5 日, Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 添加了一个增强 [BZ#2075017](#) (idm\_ds)。

#### 0.1-7

2023 年 8 月 31 日, Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 添加了一个已知问题 [BZ#2230431](#) (plumbers)。

#### 0.1-6

2023 年 8 月 29 日, Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 添加了一个已知问题 [BZ#2220915](#) (IdM)。

#### 0.1-5

2023 年 8 月 25 日, Lucie Vařáková([lvarakova@redhat.com](mailto:lvarakova@redhat.com))

- 添加了一个已知问题 [BZ#2214508](#) (内核)。

#### 0.1.4

2023 年 8 月 17 日, Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 添加了一个增强 [BZ#2136937](#) (Plumbers)。

#### 0.1.3

2023 年 8 月 14 日, Lenka Špačková([lspackova@redhat.com](mailto:lspackova@redhat.com))

- 修复了 [BZ#2128410](#) 中的一个拼写错误。

#### 0.1.2

2023 年 8 月 9 日, Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 更新了一个 bug 修复 [BZ#2155910](#) (CS)。

#### 0.1.1

2023 年 8 月 7 日, Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 更新了一个已弃用的功能发行注记 [BZ#2214130](#) (CS)。

#### 0.1.0

2023 年 8 月 3 日, Lenka Špačková([lspackova@redhat.com](mailto:lspackova@redhat.com))

- 修复了 [BZ#2142639](#) 和 [BZ#2119102](#) 中的格式化问题。
- 改进了抽象。

#### 0.0.9

2023 年 8 月 2 日, Marc Muehlfeld([mmuehlfeld@redhat.com](mailto:mmuehlfeld@redhat.com))

- 更新了一个已弃用的功能发行注记 [BZ#21004077](#) (NetworkManager)

- 更新了一个已弃用的功能发行注记 [BZ#1894877](#) (NetworkManager)。

### 0.0.8

2023 年 8 月 1 日, Mirek Jahoda ([mjahoda@redhat.com](mailto:mjahoda@redhat.com))

- 通过 bug 修复 [BZ#2207498](#), 使用 NBDE 替换了 web 控制台已知问题(RHEL web 控制台)。

### 0.0.7

2023 年 7 月 27 日 Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 根据 DDF 反馈, 修改了内核中的 3 个增强, 编译器和开发工具中的 1 个增强。

### 0.0.6

2023 年 7 月 25 日, Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 添加了一个已知问题 [BZ#2109231](#) (安装程序)。

### 0.0.5

2023 年 6 月 22 日, Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 添加了一个改进 [BZ#2087247](#) (IdM)。

### 0.0.4

2023 年 6 月 8 日, Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 添加了一个增强 [BZ#2190123](#) (内核)。

### 0.0.3

2023 年 6 月 6 日, Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 添加了 [RHELPLAN-159146](#) (IdM)。

### 0.0.2

2023 年 6 月 5 日, Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 添加了一个 KI [BZ#2176010](#)( virt)。

### 0.0.1

2023 年 5 月 10 日, Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- Red Hat Enterprise Linux 9.2 发行注记发布。

### 0.0.0

2023 年 3 月 29 日 Gabriela Fialová([gfialova@redhat.com](mailto:gfialova@redhat.com))

- Red Hat Enterprise Linux 9.2 Beta 发行注记发布。