Red Hat Enterprise Linux 8

8.0 发行注记

Red Hat Enterprise Linux 8.0 发行注记

Last Updated: 2024-05-12

Red Hat Enterprise Linux 8 8.0 发行注记

Red Hat Enterprise Linux 8.0 发行注记

法律通告

摘要

本发行注记提供了已在 Red Hat Enterprise Linux 8.0 中实现的改进和附加组件的高级信息，并记录了此发行版中的已知问题，以及重要的 bug 修复、技术预览、已弃用的功能和其他详情。

目录

对红帽文档提供反馈

我们感谢您对文档提供反馈信息。请让我们了解如何改进文档。要做到这一点：

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。包括到文档相关部分的链接。
5. 点对话框底部的 **Create**。

第 1 章 概述

基于 Fedora 28 和上游内核 4.18，Red Hat Enterprise Linux 8.0 为用户提供了跨混合云部署的一个稳定、安全、一致的基础，以及支持传统和新兴工作负载所需的工具。该版本的主要内容包括：

分发

- 内容可以通过 **BaseOS**和应用程序流(**AppStream**)存储库获得。
- **AppStream** 存储库支持传统 RPM 格式 - *模块* 的新扩展。这允许可安装一个组件的多个主版本。

如需更多信息，请参阅 第 3 章 *RHEL 8 中的内容发布*。

软件管理

- **YUM** 软件包管理器现在基于 **DNF** 技术，为模块化内容、更高的性能以及与工具集成的设计良好的稳定 API 提供支持。

详情请查看 第 5.1.4 节 “软件管理”。

Shell 和命令行工具

- RHEL 8 提供如下 **版本控制系统** ：**Git 2.18**、**Mercurial 4.8** 和 **Subversion 1.10**。

详情请查看 第 5.1.6 节 “Shell 和命令行工具”。

动态编程语言、网页和数据库服务器

- **Python 3.6** 是 RHEL 8 **中默认的 Python** 实现；对 **Python 2.7** 提供了有限的支持。默认情况下不安装任何 Python 版本。
- RHEL 中新增加了**Node.js**。从 RHEL 7 开始，其他 **动态编程语言** 已更新： **PHP 7.2**、Ruby **2.5**、**Perl 5.26**、**SWIG 3.0** 现已可用。
- 以下 **数据库服务器** 随 RHEL 8 一起发布：Maria **DB 10.3**、**MySQL 8.0**、**PostgreSQL 10**、**PostgreSQL 9.6** 和 **Redis 5**。
- RHEL 8 提供了 **Apache HTTP Server 2.4** 并引进了新的 **Web Server**，**nginx 1.14**。
- **Squid** 已更新至版本 4.4，**现在包括了新的代理服务器** ：V **arnish Cache 6.0**。

如需更多信息，请参阅 第 5.1.7 节 “动态编程语言、网页和数据库服务器”。

Desktop

- **GNOME Shell** 已更新至版本 3.28。
- GNOME 会话和 GNOME 显示管理器将 **Wayland** 用作其默认显示服务器。**X.Org 服务器**（RHEL 7 中的默认显示服务器）也可用。

如需更多信息，请参阅 第 5.1.8 节 “Desktop”。

安装程序和镜像创建

- **Anaconda** 安装程序可以使用 **LUKS2** 磁盘加密，并可以在 **NVDIMM** 设备 上安装该系统。

- **Image Builder** 工具允许用户创建各种格式的自定义系统镜像，包括准备在不同供应商的云上部署的镜像。
- 在 RHEL 8 中，可以使用**IBM Z** 上的硬件管理控制台(**HMC**)和支持元素(**SE**)从 DVD 安装。

详情请查看 第 5.1.2 节 “安装程序和镜像创建”。

内核

- 扩展的 Berkeley Packet Filtering(**eBPF**) 功能使用户空间可以将自定义程序附加到各种点（套接字、追踪点、数据包接收）来接收和处理数据。此功能 **作为技术预览提供**。
- BPF 编译器集(**BCC**)是用于创建高效内核跟踪和操作程序的工具，可作为 **技术预览** 使用。

如需更多信息，请参阅 第 5.3.1 节 “内核”。

文件系统和存储

- LUKS 版本 2(**LUKS2**)格式取代了传统的 LUKS(LUKS1)格式。**dm-crypt** 子系统和 **cryptsetup** 工具现在使用 LUKS2 作为加密卷的默认格式。

如需更多信息，请参阅 第 5.1.12 节 “文件系统和存储”。

安全性

- 默认情况下应用系统范围的**加密策略** （该策略会配置核心加密子系统，包括TLS、IPsec、SSH、DNSSEC 和 Kerberos 协议）。通过新的 **update-crypto-policies** 命令，管理员可以轻松地在模式间切换：default、legacy、futon 和 fips。
- 现在，在系统中支持 **智能卡** 和带有 **PKCS #11** 的硬件安全模块(**HSM**)。

如需更多信息，请参阅 第 5.1.15 节 “安全性”。

网络

- **nftables** 框架替代了默认网络数据包过滤功能中的 **iptables**。
- **firewalld** 守护进程现在使用 **nftables** 作为其默认后端。
- 引入了对 **IPVLAN** 虚拟网络驱动程序的支持，其对多个容器启用了网络连接。
- eXpress Data Path(**XDP**)、用于流量控制(**tc**)的 XDP和Address Family eXpress Data Path(**AF_XDP**)作为扩展 Berkeley Packet Filtering(eBPF) 功能的一部分， 可作为**技术预览**来提供。如需了解更多详细信息，请参阅技术预览中的 第 5.3.7 节 “网络”。

如需了解更多功能，请参阅新功能中的 第 5.1.14 节 “网络”。

虚拟化

- 现在，在 RHEL 8 中创建的虚拟机中，支持并自动配置一个基于 PCI Express 的机器类型(**Q35**)。这在功能和虚拟设备的兼容性方面提供了各种改进。
- 现在，可以使用 RHEL 8 web 控制台（也称为 **Cockpit** ）来创建和管理虚拟机。
- **QEMU** 模拟器引入了 **沙盒功能**，其为调用 QEMU 的系统可以执行的操作提供了可配置的限制，从而使虚拟机更加安全。

如需更多信息，请参阅 第 5.1.16 节 “虚拟化”。

编译器和开发工具

- 基于版本 8.2 的 **GCC** 编译器支持最新的 C++ 标准版本、更好的优化、新的代码强化技术、改进的警告和新的硬件特性。
- 现在，各种用于代码生成、操作和调试的工具可以实验性地处理 **DWARF5** 调试信息格式。
- 一些工具（如 **BCC**、PCP 和 **SystemTap** ）提供了对 **eBPF 跟踪** 的内核支持。
- 基于版本 2.28 的 **glibc** 库增加了对 Unicode 11、较新的 Linux 系统调用、DNS 存根解析器的关键改进、额外的安全强化功能，以及改善性能方面的支持。
- RHEL 8 提供了 OpenJDK 11、OpenJDK 8、IcedTea-Web 和各种 **Java** 工具，如 **Ant**、**Maven** 或 **Scala**。

详情请查看 第 5.1.11 节 “编译器和开发工具”。

高可用性和集群

- **Pacemaker** 集群资源管理器已升级到上游版本 2.0.0，其提供了许多程序错误修复和增强。
- 在 RHEL 8 中，pcs **配置系统** 完全支持 Corosync 3、 **knet** 和节点名称。

如需更多信息，请参阅 第 5.1.13 节 “高可用性和集群”。

其他资源

- 与其他版本系统相比， Red Hat Enterprise Linux 8 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 8 **提供软件包列表**。
- **RHEL 7 和 RHEL 8 的主要区别** 包括在使用 RHEL 8 时的注意事项。
- 有关如何从 **RHEL 7 原位升级到 RHEL 8** 的说明，请参阅 从 RHEL 7 升级到 RHEL 8 的文档。
- 有关当前支持的升级路径列表，请参阅支持的 Red Hat Enterprise Linux 原位升级路径 。
- **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。

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
- Product Life Cycle Checker
- Red Hat Product Certificates
- Red Hat Satellite Upgrade Helper

- Red Hat CVE Checker
- JVM Options Configuration Tool
- 负载均衡配置工具
- Red Hat Code Browser
- Yum Repository Configuration Helper

第 2 章 构架

Red Hat Enterprise Linux 8.0 与内核版本 4.18.0-80一同发布，它支持以下构架：

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

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 。有关可用订阅列表，请查看客户门户网站中的 订阅使用。

第 3 章 RHEL 8 中的内容发布

3.1. 安装

Red Hat Enterprise Linux 8 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：

- 二进制 DVD 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。

注意

二进制 DVD ISO 镜像大于 4.7 GB，因此它可能不适用于单层 DVD。当使用二进制 DVD ISO 镜像创建可引导安装介质时，建议使用双层 DVD 或者 USB 设备。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。软件仓库是二进制 DVD ISO 镜像的一部分。

有关下载 ISO 镜像、创建安装介质和完成 RHEL 安装的指令，请参阅 执行标准 RHEL 8 安装 文档。有关自动的 Kickstart 安装和其他高级主题，请参阅 执行高级 8 RHEL 安装 文档。

3.2. 软件仓库

Red Hat Enterprise Linux 8 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。有关通过 BaseOS 发布的软件包的列表，请查看 软件包清单。

Application Stream 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。Application Streams（应用程序流）以熟悉的 RPM 格式，作为RPM 格式的扩展，名为 *模块（modules）*，或作为 Software Collections（软件集合）。有关 AppStream 中可用软件包的列表，请查看 软件包清单。

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 8 软件仓库的详情，请查看 软件包清单。

3.3. 应用程序流

Red Hat Enterprise Linux 8 引进了应用程序流（Application Streams）的概念。现在，用户空间组件的多个版本的发布和更新频率会比核心操作系统的发布和更新频率快。这为自定义 Red Hat Enterprise Linux 提供了更大的灵活性，不会影响平台或特定部署的基本稳定性。

作为 Application Streams 提供的组件可打包为模块（module）或 RPM 软件包，并通过 RHEL 8 中的 AppStream 软件仓库提供。每个 Application Stream 组件都有其特定的生命周期，可能和 RHEL 8 的生命周期相同或更短。详情请查看 Red Hat Enterprise Linux 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。

模块流代表 Application Stream 组件的版本。例如，PostgreSQL 数据库服务器的几个流（版本）在带有默认的 **postgresql:10** 流的 **postgresql** 模块中提供。在系统中只能安装 一 个模块流。不同的容器可以使用不同的版本。

详细的模块命令，请参考 安装、管理和删除用户空间组件文档。有关 AppStream 中可用模块的列表，请查看 软件包清单。

3.4. 使用 YUM/DNF 的软件包管理

在 Red Hat Enterprise Linux 8 上，安装软件是由 **YUM** 工具来保证的，该工具基于 **DNF** 技术。我们有意坚持使用 **yum** 术语，以便与以前的 RHEL 主版本保持一致。但是，如果您输入 **dnf** 而不是 **yum**，则命令可以按预期工作，因为 **yum** 是 **dnf** 的别名，以便兼容。

如需了解更多详细信息，请参阅以下文档：

- 安装、管理和删除用户空间组件
- 使用 RHEL 8 时的注意事项

第 4 章 RHEL 8.0.1 发行版本

4.1. 新功能

RHEL 系统角色已更新

为 RHEL 子系统提供配置接口的 **rhel-system-roles** 软件包已更新。主要变更包括：

- 改进了 **网络** 角色中缺失的配置文件的处理。当通过将 persistent 状态设置为 **absent** 来删除现有的 NetworkManager on-disk 配置文件的配置时，现在只删除配置文件的持久配置，当前的运行时配置保持不变。因此，在上述情况下，对应的网络设备不再失效。
- 为 **网络** 角色中的 VLAN 和 MACVLAN 接口指定最大传输单元(MTU)的大小已被修复。因此，**使用网络** 角色在 VLAN 和 MACVLAN 接口上设置 MTU 大小不再出现以下错误消息的失败：

  ```
  failure: created connection failed to normalize: nm-connection-error-quark:
  connection.type: property is missing (6)
  ```

- **selinux** 和 **timesync** 角色现在将所有记录的输入变量包含在其默认文件中（**defaults/main.yml**）。这样便可通过检查各自默认文件的内容来轻松地确定角色支持哪些输入变量。
- **kdump** 和 **timesync** 角色已被修复，在检查模式中不会失败。

(BZ#1685902,BZ#1674004,BZ#1685904)

SOS-collector rebase 到版本 1.7

在 RHEL 8.0.1 **中，sos-collector** 软件包已更新至版本 1.7。主要变更包括：

- **SOS-collector** 现在可以从 Red Hat Enterprise Linux CoreOS(RHCOS)节点收集 sosreport，与从常规的 RHEL 节点收集sosreports的方式相同。用户无需更改他们运行 **sos-collector** 的方式。自动识别节点是RHCOS还是RHEL。
- 从 RHCOS 节点收集时，**sos-collector** 将在节点上创建一个临时容器，并使用 **support-tools** 容器生成 sosreport。该容器将在完成后被移除。
- 使用 **--cluster-type=none** 选项时，用户可以跳过在节点上运行的 **sosreport** 命令的所有与集群相关的检查或修改，只需从通过 **--nodes** 参数传递的节点静态列表中收集。
- Red Hat Satellite 现在是一个支持的集群类型，允许从 Satellite 和任何Capsules收集 sosreport。

(BZ#1695764)

升级的编译器工具集

使用 RHEL 8.0.1 升级了以下作为 Application Streams 分发的编译器工具集：

- Rust Toolset（提供 Rust 编程语言编译器 **rustc** ）、**cargo** 构建工具和依赖项管理器以及版本 1.35所需的库
- Go Toolset，它为版本 1.11.6 提供了 Go(**golang**)编程语言工具和库。

(BZ#1731500)

启用和禁用 SMT

RHEL 8 现在提供了并发多线程(SMT)配置。在 web 控制台中禁用 SMT 可让您缓解一系列 CPU 安全漏洞，例如：

- Microarchitectural Data Sampling
- L1 Terminal Fault Attack

(BZ#1713186)

4.2. 已知问题

IPSec 隧道中性能降低

使用 RHEL 8.0.1 中设置的 **aes256_sha2** 或 **aes-gcm256** IPSec 密码对 IPSec 隧道有负面影响。具有特定 VPN 设置的用户在 IPSec 隧道上的性能会下降 10%。这种回归并非由 Microarchitectural Data Sampling(MDS)缓解导致的，可以在缓解时看到缓解措施。

(BZ#1731362)

第 5 章 RHEL 8.0.0 发行版本

5.1. 新功能

这部分论述了 Red Hat Enterprise Linux 8 中引入的新功能和主要增强。

5.1.1. Web 控制台

注意

Web 控制台的订阅页面现在由新的 **subscription-manager-cockpit** 软件包提供。

在 web 控制台中添加了防火墙界面

RHEL 8 web 控制台中的 **Networking** 页面现在包含 **Firewall** 部分。在本节中，用户可以启用或禁用防火墙，以及添加、删除和修改防火墙规则。

(BZ#1647110)

Web 控制台现在默认可用

RHEL 8 web 控制台（也称为 Cockpit）的软件包现在是 Red Hat Enterprise Linux 默认软件仓库的一部分，因此可以立即在注册的 RHEL 8 系统中安装。

另外，在非最小安装 RHEL 8 中，会自动安装 web 控制台，控制台所需的防火墙端口会自动打开。登录前还添加了系统消息，提供有关如何启用或访问 Web 控制台的信息。

(JIRA:RHELPLAN-10355)

Web 控制台更好的 IdM 集成

如果您的系统在 Identity Management(IdM)域中注册，RHEL 8 web 控制台现在默认使用域的集中管理的 IdM 资源。这包括以下优点：

- IdM 域的管理员可以使用 Web 控制台来管理本地计算机。
- 控制台的 Web 服务器会自动切换到由 IdM 证书颁发机构(CA)发布并被浏览器接受的证书。
- IdM 域中具有 Kerberos 票据的用户不需要提供登录凭据来访问 Web 控制台。
- Web 控制台可以访问 IdM 域已知的 SSH 主机，而无需手动添加 SSH 连接。

请注意，要使 IdM 与 web 控制台集成正常工作，用户首先需要在 IdM master 系统中运行带有 **enable-admins-sudo** 选项的 **ipa-advise** 工具。

(JIRA:RHELPLAN-3010)

Web 控制台现在与移动浏览器兼容

有了这个更新，Web 控制台菜单和页面可以在移动浏览器变体上导航。这样就可以从移动设备使用 RHEL 8 web 控制台管理系统。

(JIRA:RHELPLAN-10352)

Web 控制台前端页面现在显示缺少的更新和订阅

如果由 RHEL 8 web 控制台管理的系统有过时的软件包或 lapsed 订阅，现在会在系统的 web 控制台前页中显示警告信息。

(JIRA:RHELPLAN-10353)

Web 控制台现在支持 PBD 注册

在这个版本中，您可以使用 RHEL 8 web 控制台界面将基于策略的 Decryption(PBD)规则应用到受管系统上的磁盘。这可使用 Clevis 解密客户端协助 web 控制台中的各种安全管理功能，如自动解锁 LUKS 加密的磁盘分区。

(JIRA:RHELPLAN-10354)

虚拟机现在可以使用 Web 控制台进行管理

Virtual Machines 页面现在可以添加到 RHEL 8 web 控制台界面中，用户可以创建和管理基于 libvirt 的虚拟机。

(JIRA:RHELPLAN-2896)

5.1.2. 安装程序和镜像创建

IBM Z 现在完全支持使用 SE 和 HMC 从 DVD 安装 RHEL

现在完全支持在 IBM Z 硬件上使用 **Support Element(SE)**和 **硬件管理控制台(HMC)** 从 DVD 安装 Red Hat Enterprise Linux 8。这个额外的功能简化了使用 **SE** 和 **HMC** 的 IBM Z 上的安装过程。

使用二进制 DVD 引导时，安装程序会提示用户输入附加内核参数。要将 DVD 设为安装源，请在内核参数后附加 **inst.repo=hmc**。然后安装程序启用了 **SE** 和 **HMC** 文件访问，从 DVD 中获取 stage2 镜像，并提供对 DVD 中软件包的访问以供软件选择。

这个新功能消除了外部网络设置的要求，并扩展了安装选项。

(BZ#1500792)

安装程序现在支持 LUKS2 磁盘加密格式

Red Hat Enterprise Linux 8 安装程序现在默认使用 LUKS2 格式，但您可以从 **Anaconda 的** Custom Partitioning 窗口中选择 LUKS 版本，或使用 Kickstart 的 **autopart**、**logvol**、**part** 和 **RAID** 命令中的新选项。

LUKS2 提供了许多改进和功能，例如，它扩展了磁盘上格式的功能，并提供灵活的元数据存储方式。

(BZ#1547908)

Anaconda 支持 RHEL 8 中的系统目的

在以前的版本中，**Anaconda** 不会为 **Subscription Manager** 提供系统目的信息。在 Red Hat Enterprise Linux 8.0 中，您可以使用 **Anaconda 的 System Purpose** 窗口或 Kickstart 的 **syspurpose** 命令在安装过程中设置系统的预期目的。安装完成后，**Subscription Manager** 在订阅系统时使用系统用途信息。

(BZ#1612060)

pykickstart 支持 RHEL 8 中的系统目的

在以前的版本中，**pykickstart** 库无法为 **Subscription Manager** 提供系统目的信息。在 Red Hat Enterprise Linux 8.0 中，**pykickstart** 解析新的 **syspurpose** 命令，并记录系统在自动化和部分自动化安装过程中的预期用途。然后，该信息被传递给 **Anaconda**，保存在新安装的系统上，并在订阅系统时对 **Subscription Manager** 可用。

(BZ#1612061)

Anaconda 支持 RHEL 8 中的新内核引导参数

在以前的版本中，您只能从内核引导参数指定基本存储库。在 Red Hat Enterprise Linux 8 中，一个新的内核参数，**inst.addrepo=<name>,<url>** 允许您在安装过程中指定附加软件仓库。

这个参数有两个强制值：仓库名称和指向存储库的 URL。如需更多信息，请参阅 https://anaconda-installer.readthedocs.io/en/latest/boot-options.html#inst-addrepo

(BZ#1595415)

Anaconda 支持 RHEL 8 中的统一 ISO

在 Red Hat Enterprise Linux 8.0 中，统一的 ISO 会自动加载 BaseOS 和 AppStream 安装源存储库。

这个功能适用于安装时载入的第一个基本存储库。例如：如果您在没有配置软件仓库的情况下引导安装，且在 GUI 中将统一 ISO 作为基本软件仓库，或者使用指向统一 ISO 的 **inst.repo=** 选项引导安装。因此，AppStream 软件仓库会在 **Installation Source** GUI 窗口的 **Additional Repositories** 部分启用。您不能删除 AppStream 存储库或更改其设置，但您可以在 **安装源中禁用它**。如果您使用不同基础程序库引导安装，然后将其改为统一 ISO，则该功能将不起作用。如果这样做，基本软件仓库将被替换。但是 AppStream 软件仓库不会被替换并指向原始文件。

(BZ#1610806)

Anaconda 可以在 Kickstart 脚本中安装模块化软件包

Anaconda 安装程序已扩展，以处理与应用程序流相关的所有功能：模块、流和配置集。Kickstart 脚本现在可以启用模块和流组合、安装模块配置集以及安装模块化软件包。如需更多信息，请参阅 执行高级 RHEL 安装。

(JIRA:RHELPLAN-1943)

现在，RHEL 8 安装选项中提供了 nosmt 引导选项

nosmt 引导选项包括在传递给新安装的 RHEL 8 系统的安装选项中。

(BZ#1677411)

RHEL 8 支持从本地硬盘中的软件仓库安装

在以前的版本中，从硬盘安装 RHEL 需要 ISO 镜像作为安装源。但是，RHEL 8 ISO 镜像对于某些文件系统来说可能太大，例如： FAT32 文件系统无法存储大于 4 GiB 的文件。

在 RHEL 8 中，您可以使用本地硬盘中的软件仓库启用安装。您只需要指定目录而不是 ISO 镜像。例如：'inst.repo=hd:<device>:<path to the repository>'

(BZ#1502323)

RHEL 8 中提供了使用 Image Builder 进行自定义系统镜像创建

Image Builder 工具可让用户创建自定义的 RHEL 镜像。镜像构建程序位于 AppStream 中的 **lorax-composer** 软件包中。

使用镜像构建器，用户可以创建包含其他软件包的自定义系统镜像。镜像构建器功能可以通过以下方式访问：

- web 控制台中的图形用户界面

- **composer-cli** 工具里的命令行界面。

镜像构建器输出格式包括：

- 实时 ISO 磁盘镜像
- 可以直接用于虚拟机或 OpenStack 的 qcow2 文件
- 文件系统镜像文件
- Azure、VMWare 和 AWS 的云镜像

如需了解更多有关镜像构建器的信息，请参阅文档标题 组成自定义的 RHEL 系统镜像。

(JIRA:RHELPLAN-7291, BZ#1628645, BZ#1628646, BZ#1628647, BZ#1628648)

添加了新的 kickstart 命令： authselect 和 modules

在这个版本中，添加了以下 kickstart 命令：

- **authselect** ：使用 **authselect** 命令在安装过程中设置系统身份验证选项。您可以使用 **authselect** 作为已弃用的 **auth** 或 **authconfig** Kickstart 命令的替代。如需更多信息，请参阅 执行高级安装 指南中的 **authselect** 部分。
- **module** ：使用 **module** 命令在 kickstart 脚本中启用软件包模块流。如需更多信息，请参阅 执行高级安装 指南中的 **module** 部分。

(BZ#1972210)

5.1.3. 内核

RHEL 8.0 中的内核版本

Red Hat Enterprise Linux 8.0 带有内核版本 4.18.0-80。

(BZ#1797671)

ARM 52 位物理寻址现已可用

在这个版本中，支持 64 位 ARM 架构的 52 位物理寻址（PA）。这比之前的 48 位 PA 提供更大的地址空间。

(BZ#1643522)

IOMMU 代码支持 RHEL 8 中的 5 级页表

Linux 内核中的 I/O 内存管理单元(IOMMU)代码已更新，以支持 Red Hat Enterprise Linux 8 中的 5 级页表。

(BZ#1485546)

支持 5 级分页

在 Linux 内核中添加了新的 **P4d_t** 软件页表类型，以便在 Red Hat Enterprise Linux 8 中支持 5 级分页。

(BZ#1485532)

内存管理支持 5 级页表

使用 Red Hat Enterprise Linux 7 时，现有内存总线有 48/46 位虚拟/物理内存寻址能力，Linux 内核实施了 4 个级别的页表，以管理这些虚拟地址到物理地址。物理总线寻址行会使物理内存上限限制为 64TB。

这些限制已扩展到 57/52 位的虚拟/物理内存寻址，其中包括 128 PiB 虚拟地址空间和 4 PB 物理内存容量。

通过扩展地址范围，Red Hat Enterprise Linux 8 中的内存管理增加了对 5 级页面表实现的支持，以便可以处理扩展的地址范围。

(BZ#1485525)

kernel-signing-ca.cer 在 RHEL 8 中被移到 kernel-core

在所有 Red Hat Enterprise Linux 7 版本中，**kernel-signing-ca.cer** 公钥都位于 **kernel-doc** 包中。但是，在 Red Hat Enterprise Linux 8 中，**kernel-signing-ca.cer** 已重新定位到每个构架的 **kernel-core** 软件包中。

(BZ#1638465)

spectre V2 缓解默认从 IBRS 改为 Retpolines

具有第 6 代 Intel Core Processors 的系统及其衍生产品[1]的 Spectre V2 漏洞(CVE-2017-5715)的默认缓解方案已从 Indirect Branch Restricted Speculation(IBRS)改为 Red Hat Enterprise Linux 8 中的 Retpolines。红帽已根据 Intel 的建议进行了此更改，以与 Linux 社区中使用的默认值保持一致，以恢复丢失的性能。但请注意，在某些情况下使用 Retpolines 可能无法完全缓解 Spectre V2。Intel 的回复文档 [2] 描述了任何暴露情况。本文档还说明攻击的风险较低。

对于需要完成 Spectre V2 缓解方案的用例，用户可通过添加 **spectre_v2=ibrs** 标志，通过内核引导行选择 IBRS。

如果没有使用 Retpoline 支持构建一个或多个内核模块，**/sys/devices/system/cpu/vulnerabilities/spectre_v2** 文件将指示漏洞，**/var/log/messages** 文件将识别出错模块。请参阅 如何确定哪些模块负责 spectre_v2 返回 "Vulnerable: Retpoline with insecure module(s)"？

[1] "6 代 Intel 核心处理器及其近乎衍生产品"是 Intel 的 Retpolines 文档所谓的"Skylake-generation"。

[2] Retpoline：一个分支目标入侵缓解 - 白皮书

(BZ#1651806)

Intel® Omni-Path Architecture (OPA) 主机软件

Red Hat Enterprise Linux 8 完全支持 Intel Omni-Path 架构(OPA)主机软件。

Intel OPA 为在集群环境中的计算和 I/O 节点之间的高性能数据传输（高带宽、高消息率、低延迟）提供主机 Fabric Interface（HFI）硬件初始化和设置。

有关安装 Intel Omni-Path 架构文档的说明，请参阅 ：
https://www.intel.com/content/dam/support/us/en/documents/network-and-i-o/fabric-products/Intel_OP_Software_RHEL_8_RN_K51383.pdf

(BZ#1683712)

NUMA 支持 RHEL 8 中的更多节点

在这个版本中，在具有 64 位 ARM 架构的系统中，非一致性内存访问(NUMA)节点数从 4 个 NUMA 节点增加到 Red Hat Enterprise Linux 8 中的 8 NUMA 节点。

(BZ#1550498)

现在，RHEL 8 中默认启用 IOMMU passthrough

默认情况下，启用了输入/输出内存管理单元(IOMMU)passthrough。这提高了 AMD 系统的性能，因为主机禁用 Direct Memory Access(DMA)重新映射。在这个版本中，Intel 系统也默认禁用了 DMA 重新映射的 Intel 系统的一致性。用户可以通过在内核命令行上，包括 hypervisor，指定 **iommu.passthrough=off** 或 **iommu=nopt** 参数来禁用此类行为（并启用 DMA 重新映射）。

(BZ#1658391)

RHEL8 内核现在支持 5 级页表

Red Hat Enterprise Linux 内核现在完全支持具有最多 5 级页表的未来 Intel 处理器。这可让处理器支持多达 4PB 的物理内存和 128PB 虚拟地址空间。使用大量内存的应用程序现在可以尽可能多地使用系统提供的内存，而不受 4 级页表的限制。

(BZ#1623590)

RHEL8 内核支持针对将来 Intel CPU 的增强 IBRS

Red Hat Enterprise Linux 内核现在支持使用增强的 Indirect Branch Restricted Speculation(IBRS)功能来缓解 Spectre V2 漏洞。启用后，IBRS 的性能将优于 Retpolines（默认）来缓解 Spectre V2，并且不会影响 Intel Control-flow Enforcement 技术。因此，在未来 Intel CPU 上启用 Spectre V2 的性能损失会较小。

(BZ#1614144)

bpftool 用于检查和操作基于 eBPF 的程序和映射

Linux 内核中添加了 **bpftool** 工具，用于根据扩展的 Berkeley Packet 过滤(eBPF)检查和简单操作程序和映射。**bpftool** 是内核源树的一部分，由 **bpftool** 软件包提供，该软件包作为 **kernel** 软件包的子软件包提供。

(BZ#1559607)

kernel-rt 源已更新

kernel-rt 源已更新为使用最新的 RHEL 内核源树。最新的内核源树现在使用上游 v4.18 实时补丁集，与之前的版本相比，它提供了很多程序错误修复和增强。

(BZ#1592977)

5.1.4. 软件管理

YUM 性能改进和支持模块化内容

在 Red Hat Enterprise Linux 8 中，安装软件是由新版本的 **YUM** 工具可保证的，它基于 **DNF** 技术(**YUM v4**)。

yum v4 比 RHEL 7 上之前使用的 **YUM v3** 有以下优点：

- 提高了性能
- 支持模块化内容
- 设计良好的稳定 API，用于与工具集成

有关 RHEL 7 中新 **YUM v4** 工具和之前版本的 **YUM v3** 之间区别的详细信息，请参阅 与 YUM 相比，DNF CLI 中的变化。

yum v4 在使用命令行、编辑或者创建配置文件时与 **YUM v3** 兼容。

要安装软件，您可以使用 **yum** 命令及其特定选项，方式与在 RHEL 7 中相同。

所选的一些 yum 插件和工具已被移植到新的 DNF 后端，可使用与 RHEL 7 中相同的名称进行安装。它们也提供兼容性符号链接，因此可在通常的位置找到 二进制文件、配置文件和目录。

请注意，**YUM v3** 提供的旧版本的 Python API 不再可用。我们建议用户将插件和脚本迁移到 **YUM v4**（DNF Python API）提供的新 API 中，它是稳定的且被完全支持。DNF Python API 请参见 DNF API 参考。

Libdnf 和 Hawkey API（C 和 Python）不稳定，在 Red Hat Enterprise Linux 8 生命周期中可能会改变。

有关 **YUM** 软件包和工具可用性更改的详情，请参阅使用 RHEL 8 的注意事项。

YUM v3 的一些功能在 **YUM v4** 中的行为可能不同。如果此类更改对您的工作流有影响，请向红帽支持创建一个问题单，请参阅 如何打开和管理客户门户网站中的支持问题单所述？

(BZ#1581198)

RHEL 8 中的主要 RPM 功能

Red Hat Enterprise Linux 8 使用 RPM 4.14。这个版本比 RPM 4.11 提供了很多改进，具体信息包括在 RHEL 7 中。最显著的功能包括：

- **debuginfo** 软件包可以并行安装
- 支持弱依赖项
- 支持富或布尔值依赖项
- 支持大小超过 4 GB 的打包文件
- 支持文件触发器

另外，最显著的变化包括：

- 更严格的 spec-parser
- 简化对非详细模式输出的签名检查
- 在宏中添加和弃用

(BZ#1581990)

RPM 现在会在开始安装前验证整个软件包内容

在 Red Hat Enterprise Linux 7 中，**RPM** 实用程序在解压缩时验证了单个文件的内容。然而，由于多个原因，这是不够的：

- 如果有效负载损坏，则只有在执行脚本操作后才会注意到它，而这些脚本操作是不可避免的。
- 如果有效负载损坏，则软件包升级会在替换之前版本的某些文件后中止，这会破坏正常安装。
- 单个文件的哈希在解压缩的数据上执行，这使得 **RPM** 易受解压缩器漏洞的影响。

在 Red Hat Enterprise Linux 8 中，使用可用的最佳哈希在单独步骤安装前验证整个软件包。

在 Red Hat Enterprise Linux 8 上构建的软件包在压缩有效负载上使用一个新的 **SHA-256** 哈希。在经过签名的包中，有效负载哈希还受到签名的保护，因此在不破坏包头上的签名和其他哈希的情况下无法更改载荷哈希。较旧的软件包使用标头和有效负载的 **MD5** 哈希，除非配置禁用。

%_pkgverify_level 宏还可用于在安装前启用强制签名验证或完全禁用有效负载验证。此外，**%_pkgverify_flags** 宏可用于限制允许哪些哈希和签名。例如，可以禁用弱 **MD5** 哈希的使用，但代价是与较旧的软件包兼容。

(JIRA:RHELPLAN-10596)

5.1.5. 基础架构服务

RHEL 8 中推荐的 Tuned 配置集的显著变化

在这个版本中，根据以下规则选择推荐的 Tuned 配置集（由 **tuned-adm recommend** 命令报告） - 匹配的第一个规则生效：

- 如果 **syspurpose**（由 **syspurpose show** 命令报告）包含 **atomic**，且同时：
  - 如果 Tuned 在裸机中运行，则会选择 **atomic-host** 配置集
  - 如果 Tuned 在虚拟机中运行，则会选择 **atomic-guest** 配置集
- 如果 Tuned 在虚拟机中运行，则会选择 **virtual-guest** 配置集
- 如果 **syspurpose** 角色包含 **desktop** 或 **workstation** ，并且 chassis 类型（由 **dmidecode**报告）是**Notebook**、**Laptop** 或 **Portable**，则会选择 **balanced** 配置集。
- 如果以上规则都不匹配，则会选择 **throughput-performance** 配置集

(BZ#1565598)

named 生成的文件可以在工作目录中写

在以前的版本中，**命名** 守护进程在工作目录中存储一些数据，这个目录在 Red Hat Enterprise Linux 中只读。在这个版本中，所选文件的路径已被更改到子目录中，允许写入。现在，默认目录 Unix 和 SELinux 权限允许写入目录。在该目录中分发的文件对 **named** 仍为只读文件。

(BZ#1588592)

Geolite Databases 已被 Geolite2 Databases 替代

Red Hat Enterprise Linux 7 中存在的 Geolite Databases 被 Red Hat Enterprise Linux 8 中的 Geolite2 Databases 替代。

Geolite Databases 由 **GeoIP** 软件包提供。上游不再支持此软件包与传统数据库。

Geolite2 数据库由多个软件包提供。**libmaxminddb** 软件包包括 库和 **mmdblookup** 命令行工具，该工具支持手动搜索地址。传统 **GeoIP** 软件包中的 **geoipupdate** 二进制文件现在由 **geoipupdate** 软件包提供，能够下载传统的数据库和新的 Geolite2 数据库。

(JIRA:RHELPLAN-6746)

CUPS 日志由 journald 处理

在 RHEL 8 中，CUPS 日志不再存储在 RHEL 7 中的 **/var/log/cups** 目录中的特定文件中。在 RHEL 8

中，所有类型的 CUPS 日志都与其他程序的日志一起集中记录在 systemd **journald** 守护进程中。若要访问 CUPS 日志，请使用 **journalctl -u cups** 命令。如需更多信息，请参阅访问 systemd 日志中的 CUPS 日志。

(JIRA:RHELPLAN-12764)

RHEL 8 中的主要 BIND 功能

RHEL 8 在包含 BIND（Berkeley Internet Name Domain）版本 9.11。与版本 9.10 相比，这个版本的 DNS 服务器引入了多个新功能和功能变化。

新特性：

- 添加了一个新的置备二级服务器的方法,称为 **Catalog Zones**。
- Domain Name System Cookies 现在由 **named** 服务和 **dig** 工具发送。
- 现在，**Response Rate Limiting** 功能可以帮助缓解 DNS 扩展攻击。
- 提高了响应策略区域(RPZ)的性能。
- 添加了名为 **map** 的新区域文件格式。使用此格式存储的区域数据可以直接映射到内存中，这样区域加载就能大大加快。
- 添加了一个名为 **delv** （域实体查找和验证）的新工具，它具有类似于 dig 的语义，用于查找 DNS 数据并执行内部 DNS 安全扩展(DNSSEC)验证。
- 现在提供了一个新的 **mdig** 命令。此命令是 'dig' 命令的一个版本，它发送多个管道查询并等待响应，而不是发送一个查询并等待响应，然后再发送下一个查询。
- 添加了一个新的 **prefetch** 选项，它可提高递归解析器性能。
- 添加了一个新的 **in-view** 区域选项，它允许在视图间共享区域数据。当使用这个选项时，多个视图可以在不需要在内存中存储多个副本的情况下为相同的区域服务。
- 添加了一个新的 **max-zone-ttl** 选项，它强制执行 zone 的最大 TTL。当加载包含更高 TTL 的区域时，加载会失败。带有更高 TTL 的动态 DNS（DDNS）更新会被接受，但 TTL 会被截断。
- 添加了新的配额，将递归解析器发送的查询限制到遇到拒绝服务攻击的权威服务器。
- 现在，**nslookup** 程序默认查找 IPv6 和 IPv4 地址。
- **named** 服务现在在启动前检查其他名称服务器进程是否正在运行。
- 加载签名区域时，**named** 现在会检查 Resource Record Signature's(RSIG)起始时间是否在将来，如果是，它会立即重新生成 RRSIG。
- 区域传送现在使用较小的消息大小来改进消息压缩，这降低了网络使用量。

功能更改：

- HTTP 接口提供统计频道的版本 **3 XML** 架构，包括新的统计信息和用于快速解析的扁平化 XML 树。旧版 **2 XML** 模式不再被支持。
- **named** 服务现在默认侦听 IPv6 和 IPv4 接口。
- **named** 服务不再支持 GeoIP。由查询发送方假定位置定义的访问控制列表（ACL）不可用。

(JIRA:RHELPLAN-1820)

5.1.6. Shell 和命令行工具

nobody 用户替换 **nfsnobody**

在 Red Hat Enterprise Linux 7 中，有：

- ID 为 99 的 **nobody** 用户和组对，以及
- ID 为 65534 的 **nfsnobody** 用户和组对（这是默认的内核溢出 ID）也是如此。

它们都已合并到 **nobody** 用户和组对中，该对使用 Red Hat Enterprise Linux 8 中的 65534 ID。新安装不再创建 **nfsnobody** 对。

这一更改减少了对 **nobody** 所有但与 NFS 无关的文件的混淆。

(BZ#1591969)

RHEL 8 中的版本控制系统

RHEL 8 提供以下版本控制系统：

- **Git 2.18**，一种分布式版本控制系统，具有分散架构。
- **Mercurial 4.8** 是轻量级分布式版本控制系统，旨在高效地处理大型项目。
- **Subversion 1.10**，一个集中版本的控制系统。

请注意，RHEL 7 提供的 Concurrent Versions System(CVS)和修订控制系统(RCS)没有与 RHEL 8 一起发布。

(BZ#1693775)

Subversion 1.10中的显著变化

Subversion 1.10 引入了一些自 RHEL 7 中发布的 1.7 版本以来的新功能，以及以下兼容性变化：

- 由于 **Subversion** 库中不兼容用于支持语言绑定，因此不支持 **Subversion 1.10** 的 **Python 3** 绑定。因此，**Subversion** 需要 **Python** 绑定的应用程序不被支持。
- 不再支持基于 **Berkeley DB** 的软件仓库。在迁移前,请使用 **svnadmin dump** 命令备份使用 **Subversion 1.7** 创建的库。安装 RHEL 8 后,请使用 **svnadmin load** 命令恢复存储库。
- RHEL 7 中的 **Subversion 1.7** 客户端签出的现有工作副本必须升级至新格式，然后才能从 **Subversion 1.10** 使用。安装 RHEL 8 后，请在每个工作副本中运行 **svn upgrade** 命令。
- 不再支持通过 **https://** 访问软件仓库的智能卡验证。

(BZ#1571415)

dstat中的显著变化

RHEL 8 提供了 **dstat** 工具的新版本。这个工具现在是 Performance Co-Pilot(PCP)工具包的一部分。**/usr/bin/dstat** 文件和 **dstat** 软件包的名称现在由 **pcp-system-tools** 软件包提供。

与 RHEL 7 提供的 **dstat** 相比，新版本的 **dstat** 引入了以下改进：

- **python3** 支持
- 历史分析
- 远程主机分析
- 配置文件插件
- 新的性能指标

(BZ#1684947)

5.1.7. 动态编程语言、网页和数据库服务器

Python 3 是 RHEL 8 中默认的 Python 实现

Red Hat Enterprise Linux 8 与 **Python 3.6** 一起分发。但默认情况下可能不会安装该软件包。要安装 **Python 3.6**，请使用 **yum install python3** 命令。

python2 软件包中提供了 **Python 2.7**。但是，**Python 2** 的生命周期会更短，其目的在于帮助客户更顺利地过渡到 **Python 3**。

RHEL 8 不与默认的 **python** 软件包或未指定版本的 **/usr/bin/python** 执行文件一起分发。建议客户直接使用 **python3** 或 **python2**。另外,管理员也可以使用 **alternatives** 命令来配置 **python** 命令。

如需更多信息，请参阅 Python 简介。

(BZ#1580387)

在 RPM 构建时，Python 脚本必须在解释器指令中指定主版本

在 RHEL 8 中，可执行的 Python 脚本预期使用明确指定至少主 Python 版本的解释器指令(hashbangs)。

构建任何 RPM 软件包时，会自动运行 **/usr/lib/rpm/redhat/brp-mangle-shebangs** buildroot 策略(BRP)脚本。这个脚本会尝试更正所有可执行文件中的解释器指令。当脚本遇到不指定 Python 主版本的 Python 解释器指令时，它会生成错误，RPM 构建失败。这种模糊的解释解释器指令的示例包括：

- **#! /usr/bin/python**
- **#! /usr/bin/env python**

要修改在 RPM 构建时导致这些构建错误的 Python 脚本中的解释器指令，请使用 **platform-python-devel** 软件包中的 **pathfix.py** 脚本：

```
pathfix.py -pn -i %{__python3} PATH ...
```

可以指定多个 *PATH*。如果 *PATH* 是一个目录，**pathfix.py** 递归扫描与模式 **^[a-zA-Z0-9_]+\.py$** 匹配的任何 Python 脚本，而不仅仅扫描那些具有模糊哈希bang的 Python 脚本。将运行 **pathfix.py** 的命令添加到 **%prep** 部分，或者在 **%install** 部分末尾。

如需更多信息，请参阅 处理 Python 脚本中的解释器指令。

(BZ#1583620)

PHP中的显著变化

Red Hat Enterprise Linux 8 带有 **PHP 7.2**。这个版本在 RHEL 7 中与 **PHP 5.4** 相比包括以下主要变化：

- **PHP** 默认使用 FastCGI 进程管理器(FPM)（与线程 **httpd**一起使用的安全）
- **httpd** 配置文件中不应使用 **php_value** 和 **php-flag** 变量，它们应该改为在池配置中设置：**/etc/php-fpm.d/*.conf**
- **PHP** 脚本错误和警告记录到 **/var/log/php-fpm/www-error.log** 文件中，而不是 **/var/log/httpd/error.log**
- 更改 PHP **max_execution_time** 配置变量时，应增加 **httpd ProxyTimeout** 设置以匹配
- 现在，运行 **PHP** 脚本的用户在 FPM 池配置中配置（ **/etc/php-fpm.d/www.conf** 文件），**apache** 用户是默认设置。
- 需要在配置更改或者安装新扩展后重启 **php-fpm** 服务
- **zip** 扩展已经从 **php-common** 软件包移动到独立的软件包 **php-pecl-zip**。

删除了以下扩展：

- **aspell**
- **MySQL**（注意 **mysqli** 和 **pdo_mysql** 扩展仍可用，它们由 **php-mysqlnd** 软件包提供）
- **memcache**

(BZ#1580430, BZ#1691688)

Ruby中的显著变化

RHEL 8 提供了 **Ruby 2.5**，它比 RHEL 7 提供的 **Ruby 2.0.0** 提供了很多新功能和增强。主要变更包括：

- 添加了增量垃圾收集器。
- 添加了 **Refinements** 语法。
- 现在会对符号进行垃圾收集。
- **$SAFE=2** 和 **$SAFE=3** 安全等级现已过时。
- **Fixnum** 和 **Bignum** 类已统一为 **Integer** 类。
- 通过优化 **Hash** 类、改善对实例变量的访问以及 **Mutex** 类小且更快速，性能得以提高。
- 某些旧的 API 已被弃用。
- 捆绑的库，如 **RubyGems**、**Rake**、**RDoc**、**Psych**、**Minitest** 和 **test-unit** 已更新。
- 其他库，如 **mathn**、**DL**、**ext/tk** 和 **XMLRPC**（之前与 **Ruby** 一起分发）已被弃用或不再包含。
- **SemVer** 版本控制方案现在用于 **Ruby** 版本。

(BZ#1648843)

Perl中的显著变化

RHEL 8 提供了 **Perl 5.26**，与 RHEL 7 中的版本相比有以下变化：

- **Unicode 9.0** 现在被支持。

- 提供了新的 **op-entry**、**load-file** 和 **load-file SystemTap** 探测。
- 在分配 scalar 时，使用写时复制机制来提高性能。
- 添加了用于处理 IPv4 和 IPv6 套接字的 **IO::Socket::IP** 模块。
- 添加了 **Config::Perl::V** 模块，来以结构化方式访问 **perl -V** 数据。
- 添加了一个新的 **perl-App-cpanminus** 软件包，其中包含用于从全面的 Perl 存档网络(CPAN)存储库获取、提取、构建和安装模块的 **cpanm** 实用程序。
- 出于安全考虑，当前目录**.** 已从 **@INC** 模块搜索路径中删除。
- 现在，因为上面描述的行为更改，当 **do** 语句无法加载文件时，会返回弃用警告。
- **doroutine(LIST)** 调用不再被支持，并会产生语法错误。
- 现在，默认随机化哈希。每次运行 **perl** 时，从哈希值返回键和值的顺序都会变化。若要禁用随机化，可将 **PERL_PERTURB_KEYS** 环境变量设置为 **0**。
- 不再允许正则表达式模式中未转义的字面 **{** 字符。
- 已删除对 **$_** 变量的字典范围支持。
- 在数组或哈希中使用 **defined** 操作符会导致致命错误。
- 从 **UNIVERSAL** 模块中导入功能会导致严重错误。
- **find2perl**、**s2p**、**a2p**、**c2ph** 和 **pstruct** 工具已被删除。
- **${^ENCODING}** 工具已被删除。**encoding** pragma 的默认模式不再被支持。要在 **UTF-8** 之外的其他编码中写入源代码，请使用编码的 **Filter** 选项。
- **perl** 打包现在与上游一致。**perl** 软件包也会安装核心模块，而 **/usr/bin/perl** 解释器则由 **perl-interpreter** 软件包提供。在以前的版本中，**perl** 软件包只包括一个最小解释器，而 **perl-core** 软件包则包括解释器和核心模块。
- **IO::Socket::SSL** Perl 模块不再从 **./certs/my-ca.pem** 文件或 **./ca** 目录加载证书颁发机构证书, 从 **./certs/server-key.pem** 文件加载服务器私钥、从 **./certs/server-cert.pem** 文件加载服务器证书、从 **./certs/client-key.pem** 文件加载客户端私钥,以及从 **./certs/client-cert.pem** 文件加载客户端证书。明确指定文件的路径。

(BZ#1511131)

Node.js 包括在 **RHEL** 中

Node.js 是一个软件开发平台,用于使用 JavaScript 编程语言开发快速、可扩展的网络应用程序。它首次在 RHEL 中提供。之前，它只通过 Software Collection 提供。RHEL 8 提供 **Node.js 10**。

(BZ#1622118)

SWIG 中的显著变化

RHEL 8 包含简化的 wrapper 和 Interface Generator(SWIG)版本 3.0，它比 RHEL 7 中发布的版本 2.0 提供了大量新功能、增强和程序错误修复。最值得注意的是，实现了对 C++11 标准的支持。**SWIG** 现在也支持 **Go 1.6**、**PHP 7**、**Octave 4.2** 和 **Python 3.5**。

(BZ#1660051)

Apache httpd中的显著变化

RHEL 8 随 Apache HTTP 服务器 2.4.37 一起发布。此版本在 RHEL 7 中对 **httpd** 进行了以下更改：

- HTTP/2 支持现在由 **mod_http2** 软件包提供，这是 **httpd** 模块的一部分。
- 现在，**mod_md** 软件包支持使用自动证书管理环境(ACME)协议的自动 TLS 证书置备和续订（与证书提供程序，如 **Let's Encrypt** 一起使用）
- Apache HTTP 服务器现在支持直接从 **PKCS#11** 模块的硬件安全令牌加载 TLS 证书和私钥。现在，**mod_ssl**配置可以使用**PKCS#11** URL来别识 TLS 私钥，也可选择使用**SSLCertificateKeyFile**和**SSLCertificateFile** 指令中的 TLS 证书。
- 使用 Apache HTTP 服务器默认配置的多处理模块(MPM)已从多进程、分叉模型（称为 **prefork**）改为高性能多线程模型 **event**。任何不是线程的第三方模块都需要被替换或删除。要更改配置的 MPM，请编辑 **/etc/httpd/conf.modules.d/00-mpm.conf** 文件。有关详细信息，请参见 **httpd.conf(5)** 手册页。

有关 **httpd** 中的变化及其用法的更多信息，请参阅 设置 Apache HTTP Web 服务器。

(BZ#1632754, BZ#1527084, BZ#1581178)

RHEL 提供了新的 nginx web 服务器

RHEL 8 引入了 **nginx 1.14**，这是支持 HTTP 和其他协议的 Web 和代理服务器，其重点在于高并发性、性能和低内存用量。**nginx** 以前仅作为 Software Collection 提供。

nginx web 服务器现在支持直接从 **PKCS#11** 模块的硬件安全令牌加载 TLS 私钥。因此， **nginx** 配置可以使用 **PKCS#11** URL 来识别 **ssl_certificate_key** 指令中的 TLS 私钥。

(BZ#1545526)

RHEL 8 中的数据库服务器

RHEL 8 提供以下数据库服务器：

- **MySQL 8.0**，一个多用户、多线程 SQL 数据库服务器。它由 **MySQL** 服务器守护进程（**mysqld**）和多个客户端程序组成。
- **MariaDB 10.3**，一个多用户、多线程 SQL 数据库服务器。就所有实际用途而言， **Maria** DB 与 **MySQL** 二进制兼容。
- **PostgreSQL 10** 和 **PostgreSQL 9.6**，一种高级对象关系数据库管理系统(DBMS)。
- **Redis 5**，一种高级的键值存储。它通常被称为数据结构服务器，因为键可以包含字符串、散列、列表、集合和排序的集合。**Redis** 在 RHEL 中第一次提供。

请注意，RHEL 8.0 中不包括 NoSQL **MongoDB** 数据库服务器，因为它使用服务器幻灯片公共许可证(SSPL)。

(BZ#1647908)

MySQL 8.0中的显著变化

RHEL 8 提供了 **MySQL 8.0**，它提供以下改进：

- **MySQL** 现在包含一个事务数据字典，用于存储数据库对象的信息。
- **MySQL** 现在支持角色，它们是特权的集合。

- 默认字符集已从 **latin1** 改为 **utf8mb4**。
- 添加了对通用表表达式（非递归和递归）的支持。
- **MySQL** 现在支持窗口功能，它使用相关行对查询中每一行执行计算。
- **InnoDB** 现在支持带有锁定读语句的 **NOWAIT** 和 **SKIP LOCKED** 选项。
- 改进了与 GIS 相关的功能。
- JSON 功能已被改进。
- 新的 **mariadb-connector-c** 软件包为 **MySQL** 和 **MariaDB** 提供通用客户端库。此库可用于任何版本的 **MySQL** 和 **MariaDB** 数据库服务器。因此，用户可以将一个应用程序构建连接到 RHEL 8 发布的任何 **MySQL** 和 **MariaDB** 服务器。

此外，随 RHEL 8 发布的 **MySQL 8.0** 服务器被配置为使用 **mysql_native_password** 作为默认身份验证插件，因为 RHEL 8 中的客户端工具和库与 **caching_sha2_password** 方法不兼容，这在上游 **MySQL 8.0 版本中** 默认使用。

要将默认身份验证插件更改为 **caching_sha2_password**，请编辑 **/etc/my.cnf.d/mysql-default-authentication-plugin.cnf** 文件：

```
[mysqld]
default_authentication_plugin=caching_sha2_password
```

另请参阅 使用 MySQL。

(BZ#1649891, BZ#1519450, BZ#1631400)

MariaDB 10.3的显著变化

与 RHEL 7 提供的版本 5.5 相比，**MariaDB 10.3** 提供了多个新功能，例如：

- 常见表表达式
- system-versioned 表
- **FOR** 循环
- 不可见的栏
- 序列
- **InnoDB**的即时 **ADD COLUMN**
- 独立于存储引擎的栏压缩
- 并行复制
- 多源复制

此外，新的 **mariadb-connector-c** 软件包为 **MySQL** 和 **MariaDB** 提供通用客户端库。此库可用于任何版本的 **MySQL** 和 **MariaDB** 数据库服务器。因此，用户可以将一个应用程序构建连接到 RHEL 8 发布的任何 **MySQL** 和 **MariaDB** 服务器。

其他显著变化包括：

- **MariaDB Galera 集群** （一个同步的多 master 集群）现在是 **MariaDB** 的标准部分。
- **InnoDB** 是默认存储引擎，而不是 **XtraDB**。
- 已删除 mariadb-bench 子包。
- 默认允许插件成熟度等级已改为比服务器成熟度低 一 个等级。因此，在之前使用的，但成熟度较低的插件将不再加载。

另请参阅 使用 MariaDB。

(BZ#1637034, BZ#1519450, BZ#1688374)

PostgreSQL中的显著变化

RHEL 8.0 提供了 **PostgreSQL** 数据库服务器的两个版本，在 **postgresql** 模块的两个流：**PostgreSQL 10** （默认流）和 **PostgreSQL 9.6** 中分发。RHEL 7 包含 **PostgreSQL** 版本 9.2。

PostgreSQL 9.6 中的显著变化是：

- 可以并行执行的顺序操作： **scan**、**join** 和 **aggregate**
- 同步复制的改进
- 改进了全文本搜索功能用于使用短语进行搜索
- **postgres_fdw** 数据联合驱动程序现在支持远程 **join**、**sort**、**UPDATE** 和 **DELETE** 操作
- 显著提高性能，特别是在多 CPU 套接字服务器上的可扩展性方面

PostgreSQL 10 的主要改进包括：

- 使用 **publish** 和 **subscribe** 关键字进行逻辑复制
- 基于 **SCRAM-SHA-256** 机制更强大的密码身份验证
- 声明性表分区
- 改进了查询并行性
- 显著的常规性能改进
- 改进了监控和控制

另请参阅 使用 PostgreSQL。

(BZ#1660041)

Squid中的显著变化

RHEL 8.0 提供了 **Squid 4.4**，它是一个用于 Web 客户端的高性能代理缓存服务器，它支持 FTP、Gopher 和 HTTP 数据对象。与 RHEL 7 提供的版本 3.5 相比，这个版本包括的新功能、改进和程序错误修复。

主要变更包括：

- 可配置的 helper 队列大小

- 对 helper 并发频道的更改
- 更改 helper 二 进制文件
- 安全互联网内容适配器协议(ICAP)
- 改进了对 Symmetric Multi Processing（SMP）的支持
- 改进的进程管理
- 删除了对 SSL 的支持
- 删除了 Edge Side Inudes(ESI)自定义解析程序
- 多配置更改

(BZ#1656871)

RHEL 提供了新的 Varnish Cache

RHEL 首次提供了**Varnish Cache**，它是一个高性能 HTTP 反向代理。之前，它只通过 Software Collection 提供。**Varnish Cache** 在内存中存储文件或碎片，以降低未来等效请求的响应时间和网络带宽消耗。RHEL 8.0 提供了 **Varnish Cache 6.0**。

(BZ#1633338)

5.1.8. Desktop

GNOME Shell，RHEL 8 中的 3.28 版本

GNOME Shell 在 Red Hat Enterprise Linux(RHEL)8 中提供 3.28 版本。主要改进包括：

- 新的 GNOME Boxes 功能
- 新屏幕键盘
- 扩展设备支持，最重要的是 Thunderbolt 3 接口的集成
- GNOME 软件、dconf-editor 和 GNOME Terminal 的改进

(BZ#1649404)

Wayland 是默认的显示服务器

在 Red Hat Enterprise Linux 8 中，GNOME 会话和 GNOME 显示管理器(GDM)使用 **Wayland** 作为其默认显示服务器，而不是 **X.org** 服务器，这些服务器与之前的 RHEL 主版本一起使用。

与 **X.org** 相比，**Wayland** 提供多种优势和改进。最值得注意的是：

- 更强大的安全模式
- 改进了多显示器处理
- 改进了用户界面(UI)扩展
- 桌面可以直接控制窗口处理。

请注意，以下功能目前不可用，或者无法按预期工作：

- Wayland 不支持多GPU设置。
- NVIDIA 二进制驱动程序在 Wayland 下无法正常工作。
- **xrandr** 实用程序不会在 Wayland 下工作，因为其处理、解决方案、轮转和布局的方法不同。请注意，操控屏幕的其他 X.org 实用程序在 Wayland 下也无法工作。
- 屏幕记录、远程桌面和可访问性在 Wayland 下并不总是能正常工作。
- 没有可用的剪贴板管理器。
- Wayland 忽略 X11 应用发布的键盘 grab，如虚拟机查看器。
- 客户机虚拟机(VM)中的 Wayland 具有稳定性和性能问题，因此建议在虚拟环境中使用 X11 会话。

如果您从使用 X.org GNOME 会话的 RHEL 7 系统升级到 RHEL 8，则您的系统将继续使用 X.org。使用以下图形驱动程序时，系统还会自动回退到 X.org ：

- NVIDIA 二进制驱动程序
- **cirrus** 驱动程序
- **mga** 驱动程序
- **aspeed** 驱动程序

您可以手动禁用 Wayland 的使用：

- 要在 GDM 中禁用 Wayland，请在 **/etc/gdm/custom.conf** 文件中设置 **WaylandEnable=false** 选项。
- 要在 GNOME 会话中禁用 Wayland，请在输入登录名称后使用登录屏幕上的 cogwheel 菜单来选择旧的 X11 选项。

有关 Wayland 的详情，请参考 https://wayland.freedesktop.org/。

(BZ#1589678)

查找默认未启用的软件仓库中的 RPM 软件包

默认情况下不启用桌面的其他存储库。禁用通过对应的 **.repo** 文件中的 **enabled=0** 行来指示。如果您尝试使用 PackageKit 从此类存储库安装软件包，PackageKit 会显示错误消息，指出该应用不可用。要使软件包可用，请将之前在对应 **.repo** 文件中的 **enabled=0** 行替换为 **enabled=1**。

(JIRA:RHELPLAN-2878)

GNOME Sofware 用于软件包管理

在 Red Hat Enterprise Linux 7 上的图形环境中为软件包管理提供工具集合的 **gnome-packagekit** 软件包不再可用。在 Red Hat Enterprise Linux 8 中，**GNOME Software** 工具程序提供了类似的功能，它可让您安装和更新应用程序及 gnome-shell 扩展。**GNOME 软件** 在 **gnome-software** 软件包中分发。

(JIRA:RHELPLAN-3001)

Wayland 上的 GNOME Shell 提供部分扩展

在 **GNOME Shell on Wayland** 会话中，提供了部分扩展功能。该功能使得按部分扩展 GUI 成为可能，这可以提高特定显示器上扩展的 GUI 的外观。

请注意，这个功能当前被视为实验性功能，因此在默认情况下是禁用的。

要启用分数扩展，请运行以下命令：

```
# gsettings set org.gnome.mutter experimental-features "['scale-monitor-framebuffer']"
```

(BZ#1668883)

5.1.9. 硬件启用

使用 **fwupd** 进行固件更新可用

RHEL 8 支持固件更新，如 UEFI 胶囊、设备固件升级(DFU)等，使用 **fwupd** 守护进程。守护进程允许会话软件自动更新本地计算机上的设备固件。

要查看和应用更新，您可以使用：

- GUI 软件管理器，如 GNOME 软件
- **fwupdmgr** 命令行工具

元数据文件从 Linux 供应商固件服务(LVFS)安全门户自动下载，并通过 D-Bus 提交至 **fwupd**。需要应用的更新将下载，显示用户通知和更新详细信息。用户必须在执行更新前明确同意固件更新操作。

请注意，默认禁用对 LVFS 的访问。

要启用对 LVFS 的访问，请单击 GNOME 软件 **源** 对话框中的滑块，或者运行 **fwupdmgr enable-remote lvfs** 命令。如果您使用 **fwupdmgr** 获取更新列表，系统会询问您是否要启用 LVFS。

通过 LVFS 的访问权限，您将直接从硬件供应商获得固件更新。请注意，这些更新尚未由红帽 QA 验证。

(BZ#1504934)

完全支持 Optane DC Persistent Memory 技术的内存模式

Intel Optane DC 持久内存存储设备提供数据中心级别的持久内存技术，这可以显著提高事务吞吐量。

要使用内存模式技术，您的系统不需要任何特殊驱动程序或特定认证。内存模式对操作系统是透明的。

(BZ#1718422)

5.1.10. 身份管理

目录服务器中的新密码语法检查

此增强为目录服务器添加新密码语法检查。例如，管理员可以启用字典检查，允许或拒绝使用字符序列和像素组。因此，如果启用，Directory 服务器中的密码策略语法检查会强制执行更安全的密码。

(BZ#1334254)

目录服务器现在提供改进的内部操作日志记录支持

目录服务器中的几个操作由服务器和客户端发起，在后台执行其他操作。在以前的版本中，服务器仅记录 **Internal** 连接关键字的内部操作，操作 ID 始终设置 **-1**。在这个版本中，Directory 服务器会记录真正的连接和操作 ID。现在，您可以将内部操作跟踪到导致此操作的服务器或客户端操作。

(BZ#1358706)

tomcatjss 库支持使用 AIA 扩展中的响应器进行 OCSP 检查

有了这个增强，**tomcatjss** 库支持使用证书的授权信息访问(AIA)扩展的响应程序的在线证书状态协议(OCSP)检查。因此，红帽认证系统的管理员可以配置使用 AIA 扩展中的 URL 的 OCSP 检查。

(BZ#1636564)

pki subsystem-cert-find 和 pki subsystem-cert-show 命令现在显示证书的序列号

有了这个增强，证书系统中的 **pki subsystem-cert-find** 和 **pki subsystem-cert-show** 命令在其输出中显示证书的序列号。序列号是重要的信息，通常由多个其他命令要求。因此，现在更容易识别证书的序列号。

(BZ#1566360)

在证书系统中已弃用了 pki user 和 pki group 命令

有了这个更新，新的 **pki < *subsystem*>-user** 和 **pki *<subsystem>*-group** 命令替换了证书系统中的 **pki user** 和 **pki group** 命令。替换的命令仍有效，但它们显示命令已弃用并引用新命令的消息。

(BZ#1394069)

证书系统现在支持系统证书离线续订

在这个版本中，管理员可以使用脱机续订功能续订证书系统中配置的系统证书。当系统证书过期时，证书系统无法启动。因为这个改进，管理员不再需要临时解决方案来替换过期的系统证书。

(BZ#1669257)

证书系统现在可以为外部 CA 签名使用 SKI 扩展创建 CSR

在这个版本中，证书系统支持为外部证书颁发机构(CA)签名使用对象密钥标识符(SKI)扩展创建证书签名请求(CSR)。某些 CA 需要使用特定值或派生自 CA 公钥的扩展。因此，管理员现在可以在传递给 **pkispawn** 工具的配置文件中使用 **pki_req_ski** 参数来创建带有 SKI 扩展名的 CSR。

(BZ#1656856)

SSSD 不再使用 [nss] 部分中的 fallback_homedir 值作为 AD 域的回退

在 RHEL 7.7 之前，Active Directory(AD)供应商中的 SSSD **fallback_homedir** 参数没有默认值。如果没有设置 **fallback_homedir**，则使用 SSSD 代替 **/etc/sssd/sssd.conf** 文件中的 **[nss]** 部分中的同一参数的值。为提高安全性，RHEL 7.7 中的 SSSD 引入了 **fallback_homedir** 的默认值。因此，SSSD 不再回退到 **[nss]** 部分中设置的值。如果要使用与 AD 域中 **fallback_homedir** 参数的默认值不同的值，您必须在域的部分中手动设置它。

(BZ#1652719)

SSSD 现在允许您选择多个智能卡验证设备之一

默认情况下，系统安全服务后台程序(SSSD)会尝试自动检测智能卡身份验证的设备。如果连接了多个设备，SSSD 会选择它检测到的第一个设备。因此，您无法选择特定的设备，有时会导致失败。

在这个版本中，您可以为 **sssd.conf** 配置文件的 **[pam]** 部分配置一个新的 **p11_uri** 选项。这个选项允许您定义用于智能卡验证的设备。

例如，要选择一个由 OpenSC PKCS#11 模块检测到插槽 ID **2** 的读取器，请添加：

```
p11_uri = library-description=OpenSC%20smartcard%20framework;slot-id=2
```

sssd.conf 的 **[pam]** 部分：

详情请查看 **man sssd.conf** 页面。

(BZ#1620123)

本地用户由 SSSD 缓存并通过 nss_ss 模块提供服务

在 RHEL 8 中，系统安全服务守护进程(SSSD)默认为 **/etc/passwd** 和 **/etc/groups** 文件中的用户和组提供服务。**sss** nsswitch 模块先于 **/etc/nsswitch.conf** 中的文件。

通过 SSSD 为本地用户提供服务的好处是 **nss_** ss 模块有一个快速的 **内存映射缓存**，与与访问磁盘以及每个 NSS 请求打开的文件相比，其加速了名称服务交换(NSS)查找，。在以前的版本中，名称服务缓存守护进程(**nscd**)帮助加快访问磁盘的过程。但是，与 SSSD 并行使用 **nscd** 非常麻烦，因为 SSSD 和 **nscd** 使用自己的独立缓存。因此，在设置中使用 **nscd** （SSSD 还为远程域的用户服务，如 LDAP 或 Active Directory）可能会导致无法预计的行为。

因此，在 RHEL 8 中，本地用户和组的解析速度更快。请注意，**root** 用户永远不会被 SSSD 处理，因此 **root** 解析不会受到 SSSD 中的潜在错误的影响。请注意，如果 SSSD 没有运行， **nss_sss** 模块会通过回退到 **nss_files** 来优雅地处理这种情况，以避免出现问题。您不必以任何方式配置 SSSD，文件域会被自动添加。

(JIRA:RHELPLAN-10439)

KCM 替换 KEYRING 作为默认的凭证缓存存储

在 RHEL 8 中，默认凭证缓存存储是 Kerberos 凭证管理器(KCM)，它由 **sssd-kcm** deamon 支持。KCM 克服了之前使用的 KEYRING 的限制，例如难以在容器化环境中使用，因为它没有命名空间，也无法查看和管理配额。

在这个版本中，RHEL 8 包含一个更适合容器化环境的凭证缓存，它为在以后的版本中构建更多功能提供了基础。

(JIRA:RHELPLAN-10440)

Active Directory 用户现在可以管理身份管理

在这个版本中，RHEL 8 允许作为 Identity Management(IdM)组的成员为 Active Directory(AD)用户添加用户 ID 覆盖。ID 覆盖是指描述特定 AD 用户或组属性在特定 ID 视图中应如下所示的记录，本例中为默认信任视图。更新后，IdM LDAP 服务器可以为 AD 用户应用 IdM 组的访问控制规则。

AD 用户现在可以使用 IdM UI 的自助服务功能，例如上传其 SSH 密钥或更改其个人数据。AD 管理员可以在没有两个不同的帐户和密码的情况下完全管理 IdM。请注意，当前 IdM 中选定的功能可能仍对 AD 用户不可用。

(JIRA:RHELPLAN-10442)

sssctl 为 IdM 域打印 HBAC 规则报告

在这个版本中，System Security Services Daemon(SSSD)的 **sssctl** 实用程序可以打印 Identity Management(IdM)域的访问控制报告。此功能满足特定环境的需求，出于法规原因，查看可访问特定客户端计算机的用户和组列表。在 IdM 客户端上运行 **sssctl access-report domain_name** 会输出应用于客户端计算机的 IdM 域中的基于主机的访问控制(HBAC)规则的解析子集。

请注意,除了 IdM 外，其它供应商都不支持这个特性。

(JIRA:RHELPLAN-10443)

身份管理软件包作为模块提供

在 RHEL 8 中，安装 Identity Management(IdM)服务器和客户端所需的软件包作为一个模块提供。**client** 流是 **idm** 模块的默认流，您可以下载安装客户端所需的软件包，而无需启用流。

IdM 服务器模块流称为 **DL1** 流。流包含多个与不同类型的 IdM 服务器对应的配置集：server、dns、adtrust、client 和 default。要在 **DL1** 流的特定配置集中下载软件包：

1. 启用流。
2. 切换到通过流提供的 RPM。
3. 运行 **yum module install idm:DL1/profile_name** 命令。

在启用了特定流并从中下载软件包后，要切换到新的模块流：

1. 删除所有相关安装内容，并禁用当前的模块流。
2. 启用新模块流。

(JIRA:RHELPLAN-10438)

为 RHEL 8 添加了会话记录解决方案

在 Red Hat Enterprise Linux 8（RHEL 8）添加了记录会话记录解决方案。新的 **tlog** 软件包及其关联的 Web 控制台会话播放器启用，以记录和回放用户终端会话。可以通过 System Security Services Daemon（SSSD）服务针对每个用户或用户组配置记录。所有终端输入和输出都会捕获并存储在系统日志中基于文本的格式。出于安全原因，输入默认为不活动，因为安全原因不会截获原始密码和其他敏感信息。

该解决方案可用于审核安全敏感系统上的用户会话。如果出现安全问题，可以检查记录的会话作为分析的一 部分。系统管理员现在可以在本地配置会话记录，并使用 **tlog-play** 工具从 RHEL 8 web 控制台界面或通过命令行界面查看结果。

(JIRA:RHELPLAN-1473)

authselect 简化了用户身份验证的配置

在这个版本中引进了 **authselect** 工具，它简化了 RHEL 8 主机的用户身份验证配置，替换了 **authconfig** 工具程序。**authselect** 提供了一种更安全的 PAM 堆栈管理方法，使系统管理员 PAM 配置更改变得更加简单。**Authselect** 可用于配置身份验证方法，如密码、证书、智能卡和指纹。请注意，**authselect** 不配置加入远程域所需的服务。此任务由特殊工具执行，比如 **realmd** 或 **ipa-client-install**。

(JIRA:RHELPLAN-10445)

SSSD 现在默认强制使用 AD GPOs

SSSD 选项 **ad_gpo_access_control** 的默认设置现在是 **enforcing**。在 RHEL 8 中，SSSD 默认根据活动目录组策略对象(GPO)强制实施访问控制规则。

红帽建议确保在从 RHEL 7 升级到 RHEL 8 前，在 Active Directory 中正确配置 GPO。如果您不想强制执行 GPO，请将 **/etc/sssd/sssd.conf** 文件中的 **ad_gpo_access_control** 选项的值改为 **permissive**。

(JIRA:RHELPLAN-51289)

5.1.11. 编译器和开发工具

将更新至版本 1.66

Boost C++ 库已更新至上游版本 1.66。Red Hat Enterprise Linux 7 中包含的 **Boost** 版本为 1.53。详情请查看上游变更日志: https://www.boost.org/users/history/

在这个版本中引进了与以前版本的兼容性更改：

- **bs_set_hook()** 函数、来自 splay 容器的 **splay_set_hook（）**，以及 **Intrusive** 库中 **splaytree_algorithms（）** 函数中的 **bool splay = true** 额外参数已被删除。
- JSON 文件中的注释或字符串串联不再被 **Property Tree** 库中的解析器支持。
- **Math** 库中的一些发行版和特殊函数已被修复，可以像记录一样运作，并引发 **overflow_error**，而不是返回最大有限值。
- **Math** 库中的一些标头已移到 **libs/math/include_private** 目录中。
- **Regex** 库中的 **basic_regex<>::mark_count()** 和**basic_regex<>::subexpression(n)** 函数已被更改为与其文档匹配。
- 在 **Variant** 库中 variadic 模板的使用可能会破坏元编程函数。
- **boost::python::numeric** API 已被删除。用户可以改为使用 **boost::python::numpy**。
- Atomic 库中不再提供指向非对象类型的指针的算术运算。

(BZ#1494495)

Unicode 11.0.0 支持

Red Hat Enterprise Linux 核心 C 库 **glibc** 已更新为支持 Unicode 标准版本 11.0.0。因此，所有宽度字符和多字节字符 API 包括字符集之间的转换和转换提供符合此标准的准确和正确的信息。

(BZ#1512004)

boost 软件包现在独立于 Python

在这个版本中，安装 **boost** 软件包不再安装 **Boost.Python** 库作为依赖项。要使用 **Boost.Python**，您需要显式安装 **boost-python3** 或 **boost-python3-devel** 软件包。

(BZ#1616244)

提供了新的 **compat-libgfortran-48** 软件包

为了与使用 Fortran 库的 Red Hat Enterprise Linux 6 和 7 应用程序兼容，现在提供了一个新的 **compat-libgfortran-48** 兼容性软件包，它提供 **libgfortran.so.3** 库。

(BZ#1607227)

GCC 中的 Retpoline 支持

在这个版本中，增加了对 GCC 的 retpoline 的支持。retpoline 是一个内核使用的软件结构，用于降低缓解 CVE-2017-5715 中描述的 Spectre 变体 2 攻击的开销。

(BZ#1535774)

增强了对工具链组件中的 64 位 ARM 架构的支持

工具链组件 **GCC** 和 **binutils** 现在为 64 位 ARM 架构提供扩展支持。例如：

- **GCC** 和 **binutils** 现在支持可扩展 Vector 扩展(SVE)。

- **GCC** 添加了对 ARM v8.2 提供的 **FP16** 数据类型的支持。**FP16** 数据类型提高了特定算法的性能。

- **binutils** 中的工具现在支持 ARM v8.3 架构定义，包括 Pointer Authentication。Pointer Authentication 功能可通过设计自己的功能指针防止恶意代码破坏程序或内核的正常执行。因此，仅在分支到代码中的不同位置时使用可信地址，这样可以提高安全性。

(BZ#1504980, BZ#1550501, BZ#1504995, BZ#1504993, BZ#1504994)

IBM POWER 系统的 glibc 优化

这个版本提供了一个新版本的 **glibc**，它针对 IBM POWER 8 和 IBM POWER 9 架构进行了优化。因此，IBM POWER 8 和 IBM POWER 9 系统现在会在运行时自动切换到适当的优化 **glibc** 变体。

(BZ#1376834)

GNU C 库更新至 2.28 版本

Red Hat Enterprise Linux 8 包含 GNU C 库(glibc)的版本 2.28。主要改进包括：

- 安全强化功能：
  - 安全标有 **AT_SECURE** 标志的二进制文件忽略 **LD_LIBRARY_PATH** 环境变量。
  - 检查故障的堆栈不再打印回溯追踪，从而加快关机并避免在受到影响的环境中运行更多代码。
- 性能改进：
  - **malloc（）** 函数的性能已使用线程本地缓存进行了改进。
  - 添加 **GLIBC_TUNABLES** 环境变量，以改变库性能特性。
  - 线程 semaphores 的实现已改进，添加了新的可扩展 **pthread_rwlock_xxx（）** 函数。
  - 数学库的性能有所改进。
- 添加了对 Unicode 11.0.0 的支持。
- 改进了对由 ISO/IEC/IEEE 60559:2011、IEEE 754-2008 和 ISO/IEC TS 18661-3:2015 标准定义的 128 位浮动点数的支持。
- 与 **/etc/resolv.conf** 配置文件相关的域名服务(DNS)根解析器改进：
  - 更改文件时将自动重新加载配置。
  - 添加了对任意数量的搜索域的支持。
  - 为 **rotate** 选项添加了正确的随机选择。
- 添加了新的开发功能，包括：
  - **preadv2** 和 **pwritev2** 内核调用的 Linux 包装器函数
  - 包括 **reallocarray（）** 和**explicit_bzero()** 的新函数
  - **posix_spawnattr_setflags（）** 函数的新标记，如 **POSIX_SPAWN_SETSID**

(BZ#1512010, BZ#1504125, BZ#506398)

RHEL 中可用的 CMake

在 Red Hat Enterprise Linux 8 中提供了 CMake 构建系统版本 3.11 作为 **cmake** 软件包。

(BZ#1590139, BZ#1502802)

make 版本 4.2.1

Red Hat Enterprise Linux 8 带有 **make** 构建工具版本 4.2.1。主要变更包括：

- 当方法失败时，会显示方法的 makefile 的名称和行号。
- 添加了 **--trace** 选项来启用目标追踪。当使用此选项时，会先打印每个方法，即使这个方法的文件名和行号被禁止，以及该方法所在的行号，以及导致调用它的前提条件。
- 混合显式和隐式规则不再导致 **make** 终止执行。相反，会输出警告信息。请注意，此语法已被弃用，未来可能会完全删除。
- **$(file ...)** 功能已被添加来将文本写入到文件中。如果不使用文本参数调用，则仅打开并立即关闭文件。
- 新选项 **--output-sync** 或 **-O** 可使每个作业对多个作业的输出进行分组，从而简化并行构建的调试。
- **--debug** 选项现在还接受 **n**（none）标记来禁用所有当前启用的调试设置。
- **!=** shell 分配运算符已添加为 **$(shell ...)** 功能的替代选择，以提高与 BSD makefile 的兼容性。有关运算符和函数之间的更多详细信息和不同之处，请参阅 GNU make manual。
  请注意，名称以感叹号结尾且紧接分配的变量（如 **variable!=value** ）现在被解释为新语法。要恢复之前的行为,请在声明标记后添加一个空格,比如 **variable! =value**。
- 添加了 POSIX 标准定义的 **::=** 分配操作器。
- 当 **.POSIX** 变量被指定后，**make** 观察处理反斜杠和新行的 POSIX 标准要求。在此模式下，保留反斜杠之前的任何尾随空格，每个反斜杠加上新行并将空格字符转换为单个空格字符。
- **MAKEFLAGS** 和 **MFLAGS** 的行为现在被精确定义。
- 一个新的变量 **GNUMAKEFLAGS** 被解析为与 **MAKEFLAGS** 相同的 **make** 标记。因此，GNU **make** 特定标记可以在 **MAKEFLAGS** 之外存储，并增加了 makefiles 的可移植性。
- 添加了包含主机架构的新变量 **MAKE_HOST**。
- 新变量 **MAKE_TERMOUT** 和 **MAKE_TERMERR** 表示 **make** 是否将标准输出和错误写到终端。
- 在 makefile 中的 **MAKEFLAGS** 变量中设置 **-r** 和 **-R** 选项现在可以正常工作，并分别删除了所有内置规则和变量。
- 现在，每种方法都记住了 **.RECIPEPREFIX** 设置。此外，在该方法中扩展的变量也使用该方法前缀设置。
- **-p** 选项的输出中会显示 **.RECIPEPREFIX** 设置和所有特定于目标的变量，就像在 makefile 中一样，而不是注释。

(BZ#1641015)

SystemTap 版本 4.0

Red Hat Enterprise Linux 8 使用 **SystemTap** 工具工具版本 4.0 进行发布。主要改进包括：

- 扩展的 Berkeley Packet 过滤器(eBPF)后端已被改进，特别是字符串和功能。若要使用此后端，请使用 **--runtime=bpf** 选项启动 **SystemTap**。
- 添加了一个新的导出网络服务，用于 Prometheus 监控系统。
- 系统调用探测实施已被改进，必要时可使用内核追踪点。

(BZ#1641032)

binutils 版本 2.30 中的改进

Red Hat Enterprise Linux 8 包含 **binutils** 软件包的版本 2.30。主要改进包括：

- 改进了对新的 IBM Z 架构扩展的支持。

链接器：

- 默认情况下，链接器将代码和只读数据放在单独的片段中。因此，创建的可执行文件更大且更易于运行，因为动态加载器可以禁用任何包含只读数据的内存页面的执行。
- 添加了对 GNU 属性说明的支持，该注释为动态加载器提供有关二进制文件的提示。
- 在以前的版本中，链接程序为 Intel Indirect Branch Tracking(IBT)技术生成无效的可执行代码。因此，生成的可执行文件无法启动。这个程序错误已被解决。
- 在以前的版本中，**gold** 链接器不正确地合并了属性备注。因此，生成的代码中可能会启用错误的硬件功能，代码可能会意外终止。这个程序错误已被解决。
- 在以前的版本中，**金级** 链路器创建了末尾带有填充字节的备注部分，以便根据架构实现对齐。由于动态加载器没有期望 padding，所以coud 会意外终止它正在加载的程序。这个程序错误已被解决。

其他工具：

- **readelf** 和 **objdump** 工具现在可以选择在单独的调试信息文件中追踪链接并在其中显示信息。
- 新的 **--inlines** 选项扩展 **objdump** 工具的现有 **--line-numbers** 选项，以显示内嵌功能的信息。
- **nm** 工具获得了一个新的选项 **--with-version-strings**，来在其名称后显示符号的版本信息（如果存在的话）。
- 在 assembler 中添加了对 ARMv8-R52、Cortex-M23 和 Cortex-M33 处理器的支持。

(BZ#1641004, BZ#1637072, BZ#1501420, BZ#1504114, BZ#1614908, BZ#1614920)

Performance Co-Pilot 版本 4.3.0

Red Hat Enterprise Linux 8 提供了 **Performance Co-Pilot** (PCP)版本 4.3.0。主要改进包括：

- **pcp-dstat** 工具现在包含历史分析和 Comma 分隔的值(CSV)格式输出。
- 日志实用程序可以使用指标标签并帮助文本记录。
- **pmdaperfevent** 工具现在在较低 Simultaneous Multi Threading(SMT)级别报告正确的 CPU 号码。

- **pmdapostgresql** 工具现在支持 **Postgres** 系列 10.x。
- **pmdaredis** 工具现在支持 **Redis** 系列 5.x。
- 通过动态进程过滤和按进程系统调用、ucalls 和 ustat 增强了 **pmdabcc** 工具。
- **pmdammv** 工具现在导出指标标签，格式版本增加到 3。
- **pmdagfs2** 工具支持额外的 glock 和 glock 拥有者指标。
- 对 SELinux 策略进行了几个修复。

(BZ#1641034)

内存保护密钥

在这个版本中，启用了允许每个线程页面保护标志更改的硬件功能。已为 **pkey_alloc（）**、**pkey_free（）** 和 **pkey_mprotect（）** 函数添加了新的 **glibc** 系统调用包装器。另外，已添加了 **pkey_set（）** 和 **pkey_get（）** 函数，以允许访问每个线程的保护标记。

(BZ#1304448)

GCC 现在默认为 IBM Z 中的 z13

在这个版本中，IBM Z 架构代码中的默认 GCC 为 z13 处理器构建代码，并为 z14 处理器调整代码。这等同于使用 **-march=z13** 和 **-mtune=z14** 选项。用户可以通过使用目标架构和调整选项来显式覆盖此默认设置。

(BZ#1571124)

elfutils 更新至版本 0.174

在 Red Hat Enterprise Linux 8 中，**selfutils** 软件包包括在 0.174 版中。主要变更包括：

- 在以前的版本中，**eu-readelf** 工具可能会显示一个带有负值的变量，就像它有一个很大的未签名值，或者以负值的形式显示一个大的未签名值。这已被修正，**eu-readelf** 现在会查找恒定值类型的大小和签名状态，以正确显示它们。
- 用于读取缺少 CU 的 **.debug_line** 数据的新函数 **dwarf_next_lines()** 已被添加到 **libdw** 库中。此函数可用作 **dwarf_getsrclines（）** 和 **dwarf_getsrcfiles（）** 函数的替代。
- 在以前的版本中，具有超过 65280 部分的文件可能会在 **libelf** 和 **libdw** 库以及所有使用它们的工具中导致错误。这个程序错误已被解决。因此，ELF 文件标头中扩展的 **shnum** 和 **shstrndx** 值会被正确处理。

(BZ#1641007)

Valgrind 更新至 3.14 版本

Red Hat Enterprise Linux 8 使用 Valgrind 可执行代码分析工具 3.14。主要变更包括：

- 添加了一个新的 **--keep-debuginfo** 选项，用于为卸载的代码保留调试信息。因此，保存的堆栈跟踪可以包含内存中不再存在的代码的文件和行信息。
- 添加了基于源文件名和行号的压缩。
- **Helgrind** 工具已使用 **--delta-stacktrace** 选项进行了扩展，以指定完整历史记录堆栈跟踪的计算。值得注意的是，将这个选项与 **--history-level=full** 结合使用可以将 **Helgrind** 的性能提升高达 25%。

- 在 **Memcheck** 工具中，Intel 和 AMD 64 位 arcitects 上优化的代码和 ARM 64 位构架的正率已被降低。请注意，您可以使用 **--expensive-definedness-checks** 来控制定义的检查处理，并通过牺牲性能提高速率。
- Valgrind 现在可以识别 IBM Power Systems 的 little-endian 变体的更多指令。
- Valgrind 现在可以处理 IBM Z 架构 z13 处理器的大部分整数和字符串向量指令。

有关新选项及其已知限制的更多信息，请参阅 **valgrind(1)** 手册页。

(BZ#1641029, BZ#1501419)

GDB 版本 8.2

Red Hat Enterprise Linux 8 带有 GDB debugger 版本 8.2 Notable 更改，其中包括：

- IPv6 协议支持使用 GDB 和 **gdbserver** 进行远程调试。
- 改进了在没有调试信息的情况下进行调试。
- GDB 用户界面中的符号补全已被改进，通过使用更多语法结构（如 ABI 标签或命名空间）来提供更好的建议。
- 现在可以在后台执行命令。
- 现在，可以使用 Rust 编程语言创建的调试程序。
- 调试 C 和 C++ 语言已使用对 **_Alignof** 和 **alignof** 运算符、C++ rvalue 引用和 C99 变量长度自动数组的解析器支持进行了改进。
- GDB 扩展脚本现在可以使用 Guile 脚本语言。
- 新的 API 功能、帧解码器、过滤器和解压器改进了扩展的 Python 脚本语言接口。另外，GDB 配置的 **.debug_gdb_scripts** 部分中的脚本会自动加载。
- GDB 现在使用 Python 版本 3 运行其脚本，包括友善打印机、帧解码器、过滤器和解压器。
- 通过流程执行记录和重播（包括 Thumb 32 位和系统调用说明）改进了 ARM 和 64 位 ARM 架构。
- GDB 现在支持 64 位 ARM 架构上的可扩展 Vector 扩展(SVE)。
- 添加了对 Intel PKU 寄存器和 Intel Processor Trace 的支持。
- 记录和重播功能已扩展，以包含基于 Intel 的系统上的 **rdrand** 和 **rdseed** 指令。
- IBM Z 构架中的 GDB 功能增加了，支持追踪点、快速追踪点、向量寄存器和 ABI 以及 **Catch** 系统调用。另外，GDB 现在支持更最新的架构指令。
- GDB 现在可以使用 64 位 ARM 架构中的 SystemTap 静态用户空间探测(SDT)。

(BZ#1641022, BZ#1497096, BZ#1505346, BZ#1592332, BZ#1550502)

RHEL 的 glibc 本地化在多个软件包中分发

在 RHEL 8 中，**glibc** 区域和翻译不再由单一 **glibc-common** 软件包提供。现在，本地化的内容和语言都位于 **glibc-langpack-*CODE*** 软件包中。另外，在大多数情形中，默认情况下不安装所有区域，仅安装安装程序中选择的区域。用户必须单独安装所需的所有更多区域设置软件包，或者如果用户希望安装 **glibc-**

all-langpacks，以获取包含之前安装的所有 **glibc** 区域的区域归档。

如需更多信息，请参阅 请使用语言包。

(BZ#1512009)

GCC 版本 8.2

在 Red Hat Enterprise Linux 8 中，GCC 工具链基于 GCC 8.2 版本系列。主要变更包括：

- 添加了大量常规优化，如别名分析、向量器改进、相同的代码折叠、流程间分析、存储合并优化通过等。
- 改进了 Address Sanitizer。添加了 Leak Sanitizer 和 Undefined Behavior Sanitizer。
- 现在可使用 DWARF5 格式生成调试信息。这个功能是实验性的。
- 源代码覆盖分析工具 GCOV 已进行了各种改进。
- 为静态检测更多编程错误，添加了新的警告并改进了诊断。
- GCC 已被扩展，提供一些工具以确保增加生成的代码的强化。与安全性相关的改进包括内置的溢出检查、对堆栈冲突的额外保护、控制流指令的目标地址、有界字符串操作功能的警告，以及检测边界外数组索引的警告。

架构和处理器支持的改进包括：

- 添加了多个适用于 Intel AVX-512 架构、微架构和 Intel Software Guard 扩展(SGX)的新架构特定选项。
- 现在，代码生成可以针对 64 位 ARM 架构 LSE 扩展、ARMv8.2-A 16 位 Floating-Point 扩展(FPE)和 ARMv8.2-A、ARMv8.3-A 和 ARMv8.4-A 架构版本。
- 添加了对 IBM Z 架构的 z13 和 z14 处理器的支持。

与语言和标准有关的显著变化包括：

- C 语言编译代码时使用的默认标准已改为使用 GNU 扩展的 C17。
- C++ 语言编译代码时使用的默认标准已改为使用 GNU 扩展的 C++14。
- C++ 运行时程序库现在支持 C++11 和 C++14 标准。
- C++ 编译器现在实施 C++14 标准。
- 改进了对 C 语言标准 C11 的支持。
- 新的 **__auto_type** GNU C 扩展在 C 语言中提供了 C++11 **auto** 关键字的功能子集。
- 由 ISO/IEC TS 18661-3:2015 标准指定的 **_FloatN** 和 **_FloatNx** 类型名称现在由 C 前端识别。
- 现在，传递空类作为参数不包括在 Intel 64 和 AMD64 构架中，如平台 ABI 要求。
- 更正了 C++11 **alignof** 运算符返回的值，以便与 C **_Alignof** 运算符匹配，并返回最小对齐。要查找首选的对齐，请使用 GNU 扩展名 **__alignof__**。
- Fortran 语言代码的 **libgfortran** 库的主要版本已改为 5。

- 对 Ada(GNAT)、GCC Go 和 Objective C/C++ 语言的支持已被删除。使用 Go Toolset 进行 Go 代码开发。

(JIRA:RHELPLAN-7437, BZ#1512593, BZ#1512378)

Go 加密库 FIPS 模式现在遵循系统设置

在以前的版本中，Go 标准加密库总是使用其 FIPS 模式，除非在使用库构建应用程序时明确禁用它。因此，基于 Go 的应用程序的用户无法控制是否使用 FIPS 模式。在这个版本中，当系统没有以 FIPS 模式配置时，库不会被默认为 FIPS 模式。因此，RHEL 系统中基于 Go 的应用程序的用户对 Go 加密库的 FIPS 模式的使用具有更多控制。

(BZ#1633351)

strace 更新至版本 4.24

Red Hat Enterprise Linux 8 带有 **strace** 工具版本 4.24。主要变更包括：

- 系统调用修改功能已添加 **-e inject=** 选项。这包括注入错误、返回值、延迟和信号。
- 改进了系统调用资格语法：
  - 为使用正则表达式过滤系统调用，添加了 **-e trace=/regex** 选项。
  - 在 **-e trace=** 选项中为系统调用资格前添加问号可让 **strace** 继续，即使资格与任何系统调用不匹配。
  - 在 **-e trace** 选项中为系统调用资格中添加了个性化名称。
- 添加 **kvm vcpu** 退出原因解码。为此，可使用 **-e kvm=vcpu** 选项。
- **elfutils** 中的 **libdw** 库现在用于使用 **-k** 选项时的堆栈展开。此外，符号解调通过 **libiberty** 库来执行。
- 在以前的版本中，**r** 选项会导致 **strace** 忽略 **-t** 选项。这个问题已被解决，两个选项现在都独立。
- 添加了 **-A** 选项，以附加模式打开输出文件。
- 添加了 **-X** 选项来配置 **xlat** 输出格式。
- 改进了使用 **-yy** 选项解码套接字地址。此外，在 **-yy** 模式中添加了块和字符设备编号打印。
- 现在，可以在 IBM Z 构架上使用单个 **strace** 工具跟踪 64 位和 32 位二进制文件。因此，RHEL 8 中不再存在独立的 **strace32** 软件包。

另外，还添加了、改进或更新以下项目的解码：

- **netlink** 协议、消息和属性
- **arch_prctl**,**bpf**,**getockopt**,**io_pgetevent**,**keyctl**,**prctl**,**pkey_alloc**,**pkey_free**,**pkey_mprotect**,**ptrace**,**rseq**,**setsockopt**,**socket**,**statx** 和其他系统调用
- 为 **ioctl** 系统调用有多个命令
- 各种类型的常量
- **execveat**、**inotify_add_watch**、**inotify_init**、**select**、**symlink**、**symlinkat**系统调用带有间接参数的 **mmap** 系统调用的路径追踪

- 信号代码列表

(BZ#1641014)

RHEL 8 中的编译器工具集

RHEL 8.0 提供了以下编译器工具集作为 Application Streams：

- clang 和 LLVM Toolset 7.0.1，提供 LLVM 编译器基础架构框架、Clang 编译器和 C++ 语言的 Clang 编译器、LLDB 调试器以及用于代码分析的相关工具。请参阅 使用 Clang 和 LLVM Toolset 文档。
- Rust Toolset 1.31 提供了 Rust 编程语言编译器 **rustc**、**cargo** 构建工具和依赖项管理器、**cargo-vendor** 插件和所需的库。请参阅 使用 Rust Toolset 文档。
- Go Toolset 1.11.5，它提供 Go 编程语言工具和库。Go 也称为 **golang**。请参阅 使用 Go Toolset 文档。

(BZ#1695698, BZ#1613515, BZ#1613516, BZ#1613518)

RHEL 8 中的 Java 实现和 Java 工具

RHEL 8 AppStream 软件仓库包括：

- **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。
- **icedtea-web** 软件包提供了一个 Java Web Start 的实现。
- **ant** 模块，它为编译、汇编、测试和运行 Java 应用程序提供了 Java 库和命令行工具。 **Ant** 更新至 1.10 版本。
- **maven** 模块，提供软件项目管理和理解工具。**Maven** 之前只作为软件集合或不支持的可选渠道提供。
- **scala** 模块，为 Java 平台提供通用编程语言。 **Scala** 以前仅作为 Software Collection 提供。

此外，**java-1.8.0-ibm** 软件包通过 Supplementary 存储库分发。请注意，红帽不支持这个存储库中的软件包。

(BZ#1699535)

std::string 和 std::list 中 C++ ABI 的更改

libstdc++ 库中的 **std::string** 和 **std::list** 类的应用二进制接口(ABI)在 RHEL 7 (GCC 4.8)和 RHEL 8(GCC 8)之间的更改符合 C++11 标准。**libstdc++** 库支持新旧 ABI，但其他一些 C++ 系统库则不支持。因此，需要重建针对这些库动态链接的应用程序。这会影响所有 C++ 标准模式，包括 C++98。它还影响了使用红帽开发人员工具集编译器构建的适用于 RHEL 7 的应用程序，该编译器保留旧 ABI，以保持与系统库的兼容性。

(BZ#1704867)

5.1.12. 文件系统和存储

支持 Data Integrity Field/Data Integrity Extension（DIF/DIX）

只有在硬件厂商已验证，并完全支持在 RHEL 中的特定主机总线适配器（HBA）和存储阵列，则支持 DIF/DIX。

在以下配置中不支持 DIF/DIX：

- 不支持在引导设备中使用。
- 在虚拟客户机中不支持。
- 当启用了 DIF/DIX 时，红帽不支持使用 Automatic Storage Management 库（ASMLib）。

在涉及应用程序之前（包括应用程序）的不同层的存储设备上启用或禁用 DIF/DIX。在存储设备中激活 DIF 的方法取决于设备。

有关 DIF/DIX 功能的详情，请参考 什么是 DIF/DIX。

(BZ#1649493)

XFS 现在支持共享的 copy-on-write 数据扩展

XFS 支持共享的 copy-on-write 数据扩展功能这个功能可让两个或者多个文件共享一组通用的数据块。当任何一个共享通用块更改的文件时, XFS 会破坏到通用块的链接并创建新文件。这与其他文件系统中找到的 copy-on-write（COW）功能类似。

共享复制时写入数据扩展：

速度快

创建共享副本不会利用磁盘 I/O。

空间效率

共享块不消耗额外的磁盘空间。

透明

共享常见块操作的文件，类似常规文件。

用户空间工具可以使用共享的复制时写入数据扩展：

- 高效的文件克隆，例如使用 **cp --reflink** 命令
- 针对文件的快照

内核子系统（如 Overlayfs 和 NFS）也使用这个功能来更有效地操作。

现在，从 **xfsprogs** 软件包版本 **4.17.0-2.el8** 开始，在创建 XFS 文件系统时默认启用共享的写时复制数据扩展。

请注意，直接访问(DAX)设备目前不支持带有共享写时复制数据扩展的 XFS。要创建没有这个功能的 XFS 文件系统，请使用以下命令：

```
# mkfs.xfs -m reflink=0 block-device
```

Red Hat Enterprise Linux 7 可以使用共享复制时写入数据扩展以只读模式挂载 XFS 文件系统。

(BZ#1494028)

最大 XFS 文件系统大小为 1024 TiB

XFS 文件系统最多支持的大小从 500 TiB 增加到 1024 TiB。

大于 500 TiB 的文件系统需要：

- 元数据 CRC 功能和空闲内节点 btree 功能都以文件系统格式启用，以及
- 分配组大小至少为 512 GiB。

在 RHEL 8 中，**mkfs.xfs** 工具默认创建满足这些要求的文件系统。

不支持增大一个小于 500 TiB 的新文件系统，它无法满足这些要求。

(BZ#1563617)

ext4 文件系统现在支持 元数据校验和

有了这个更新，**ext4** 元数据被 **checksums**保护。这可让文件系统识别损坏的元数据，从而避免损坏并增加文件系统的弹性。

(BZ#1695584)

VDO 现在支持所有构架

Virtual Data Optimizer(VDO)现在包括在 RHEL 8 支持的所有构架中。

有关支持的构架列表，请参阅 第 2 章 *构架*。

(BZ#1534087)

BOOM 引导管理器简化了创建引导条目的过程

BOOM 是 Linux 系统的引导管理器，使用引导装载程序支持 BootLoader 规范用于引导条目配置。它启用了灵活的引导配置，并简化了新引导条目或修改的引导条目的创建：例如，为使用 LVM 创建的系统引导快照镜像。

BOOM 不会修改现有的引导装载程序配置，仅插入附加条目。保持现有配置，并且所有发行版集成（如内核安装和更新脚本）都可以象以前一样继续运行。

BOOM 具有简化的命令行界面(CLI)和 API，可简化创建引导条目的任务。

(BZ#1649582)

LUKS2 现在是加密卷的默认格式

在 RHEL 8 中，LUKS 版本2 (LUKS2) 格式替换了传统的 LUKS(LUKS1)格式。**dm-crypt** 子系统和 **cryptsetup** 工具现在使用 LUKS2 作为加密卷的默认格式。LUKS2 在出现部分元数据崩溃事件时，为加密卷提供元数据冗余和自动恢复功能。

由于内部布局，LUKS2 也是将来功能的启用器。它支持通过 **libcryptsetup** 中构建的通用 kernel-keyring 令牌自动锁定，该令牌允许用户使用存储在 kernel-keyring 保留服务中的密语解锁 LUKS2 卷。

其他显著改进包括：

- 使用嵌套密钥加密方案进行保护的密钥设置。
- 更轻松地与基于策略的加密（Clevis）集成。
- 最多 32 个密钥插槽 - LUKS1 只提供 8 个密钥插槽。

详情请查看 **cryptsetup(8)** 和 **cryptsetup-reencrypt(8)** man page。

(BZ#1564540)

Broadcom Emulex 和 Marvell Qlogic Fibre Channel 适配器完全支持 NVMe/FC

现在，当与 Broadcom Emulex 和 Marvell Qlogic Fibre Channel 32Gbit 适配器支持时，Initiator 模式中完全支持 NVMe over Fibre Channel 32Gbit 传输类型。

NVMe over Fibre Channel 是之前在 Red Hat Enterprise Linux 中引入的 Nonvolatile Memory Express(NVMe)协议之外的额外光纤传输类型。

启用 NVMe/FC:

- 要在 **lpfc** 驱动程序中启用 NVMe/FC，编辑 **/etc/modprobe.d/lpfc.conf** 文件并添加以下选项：

```
lpfc_enable_fc4_type=3
```

- 要在 **qla2xxx** 驱动程序中启用 NVMe/FC，编辑 **/etc/modprobe.d/qla2xx.conf** 文件并添加以下选项：

```
qla2xxx.ql2xnvmeenable=1
```

其他限制：

- NVMe/FC 不支持多路径。
- NVMe/FC 不支持 NVMe 集群。
- **kdump** 不支持 NVMe/FC。
- 不支持从 Storage Area Network (SAN) NVMe/FC 引导。

(BZ#1649497)

新的 scan_lvs 配置设置

添加了一个新的 **lvm.conf** 配置文件设置 **scan_lvs**，默认设置为 0。新的默认行为阻止 LVM 查找 LV 上可能存在的 PV；也就是说，它不会扫描活动 LV 以获得更多 PV。默认设置还可阻止 LVM 在 LV 之上创建 PV。

LV 上方的 PV 可以通过放置在 LV 上的虚拟机镜像来进行，在这种情况下，主机无法安全地访问 PV。避免这种不安全的访问是新默认行为的主要原因。另外，在具有许多活动 LV 的环境中，LVM 可以显著减少设备扫描量。

通过将此设置更改为 1，可以恢复之前的行为。

(BZ#1676598)

新的 DM 多路径配置文件中的 overrides 部分

/etc/multipath.conf 文件现在包含一个 **overrides** 部分，允许您为所有设备设置配置值。这些属性可由 DM 多路径为所有设备使用，除非被 **/etc/multipath.conf** 文件的 **multipaths** 部分中指定的属性覆盖，用于包含该设备的路径。此功能替换了 配置文件的 **devices** 部分的 **all_devs** 参数，该参数不再被支持。

(BZ#1643294)

现在支持使用 NVDIMM 设备安装和引导

在更新之前，安装程序会忽略任何模式中的 Nonvolatile Dual Inline Memory Module(NVDIMM)设备。

在这个版本中，内核的改进支持 NVDIMM 设备提供更好的系统性能功能，并增强了对写密集型应用程序（如数据库或分析工作负载）的文件系统访问，并减少了 CPU 开销。

这个版本引进了对以下支持：

- 使用 **nvdimm** Kickstart 命令和 GUI 安装的 NVDIMM 设备的使用可让您在扇区模式下从 NVDIMM 设备安装和引导，并在安装过程中将 NVDIMM 设备重新配置为扇区模式。
- 使用命令处理 NVDIMM 设备，**Anaconda** 的 **Kickstart** 脚本的扩展。
- **grub2**、**efibootmgr** 和 **efivar** 系统组件能够处理和从 NVDIMM 设备引导。

(BZ#1499442)

改进了在 **DM** 多路径中发现的边缘路径

multipathd 服务现在支持改进的路径检测。这有助于多路径设备避免可能重复失败的路径，并提高性能。边缘路径（Marginal paths）是带有持久性但可能会出现 I/O 错误的路径。

/etc/multipath.conf 文件中的以下选项控制边路径行为：

- **marginal_path_double_failed_time**,
- **marginal_path_err_sample_time**,
- **marginal_path_err_rate_threshold**，和
- **marginal_path_err_recheck_gap_time**.

DM 多路径在配置的示例时间中禁用路径并使用重复 I/O 进行测试，如果：

- 设置了 **multipath.conf** 选项，
- 路径在配置的时间里失败两次，同时
- 其它路径可用。

如果在本次测试中路径超过配置的 err 速率，DM 多路径会在配置的空白时间忽略它，然后重新测试以查看它是否足够正常工作。

如需更多信息，请参阅 **multipath.conf** man page。

(BZ#1643550)

块设备的多队列调度

块设备现在在 Red Hat Enterprise Linux 8 中使用多队列调度。这可让块层性能针对使用快速固态驱动器（SSD）和多核系统进行正常扩展。

RHEL 7 及更早的版本中提供的传统调度程序已被删除。RHEL 8 只支持多队列调度程序。

(BZ#1647612)

5.1.13. 高可用性和集群

新的 **pcs** 命令列出可用的 **watchdog** 设备并测试 **watchdog** 设备

要使用 Pacemaker 配置 SBD，需要一个正常工作的 watchdog 设备。此发行版本支持 **pcs stonith sbd watchdog list** 命令，来列出本地节点上的可用 watchdog 设备，支持 **pcs stonith sbd watchdog test** 命令来测试 watchdog 设备。有关 **sbd** 命令行工具的详情请参考 **sbd**(8)man page。

(BZ#1578891)

pcs 命令现在支持按操作及其间隔过滤资源故障

Pacemaker 现在会根据资源名称和节点的每个资源操作跟踪资源故障。**pcs resource failcount show** 命令现在允许按资源、节点、操作和间隔过滤故障。它提供了一个选项，可以显示每个资源和节点聚合的故障，或按资源、节点、操作及其间隔详细显示。另外，**pcs resource cleanup** 命令现在允许按资源、节点、操作和间隔过滤失败。

(BZ#1591308)

corosync 日志中启用的时间戳

corosync 日志以前不包含时间戳，因此很难将它与其他节点和守护进程的日志关联。有了此版本，时间戳存在于 **corosync** 日志中。

(BZ#1615420)

pcs cluster setup、**pcs cluster node add** 和 **pcs cluster node remove** 命令有新格式

在 Red Hat Enterprise Linux 8 中，**pcs** 完全支持 Corosync 3、**knet** 和节点名称。现在，节点名称是必需的，并替换节点标识符角色中的节点地址。节点地址现在是可选的。

- 在 **pcs host auth** 命令中，节点地址默认为节点名称。
- 在 **pcs cluster setup** 和 **pcs cluster node add** 命令中，节点地址默认为 **pcs host auth** 命令中指定的节点地址。

随着这些更改，设置集群的命令格式、在集群中添加节点以及从集群中删除节点已更改。详情请参阅 **pcs cluster setup**、**pcs cluster node add** 和 **pcs cluster node remove** 命令的帮助信息。

(BZ#1158816)

新 **pcs** 命令

Red Hat Enterprise Linux 8 包括以下新命令。

- RHEL 8 引入了一个新命令 **pcs cluster node add-guest | remove-guest**，它在 RHEL 7 中取代了 **pcs cluster remote-node add | remove** 命令。
- RHEL 8 引入了一个新命令 **pcs quorum unblock**，它取代了 RHEL 7 中的 **pcs cluster quorum unblock** 命令。
- **pcs resource failcount reset** 命令已被删除，因为它与 **pcs resource cleanup** 命令的功能重复。
- RHEL 8 引入了一个新命令，其替代了 RHEL 7 中的 **pcs resource [show]** 命令：
  - RHEL 8 中的 **pcs resource [status]** 命令替代了 RHEL 7 中的 **pcs resource [show]** 命令。
  - RHEL 8 中的 **pcs resource config** 命令替换了 RHEL 7 中的 **pcs resource [show] --full** 命令。
  - RHEL 8 中的 **pcs resource config *resource id*** 命令取代了了 RHEL 7 中的 **pcs resource show *resource id*** 命令。

- RHEL 8 引入了一个新命令，其替换了 RHEL 7 中的 **pcs stonith [show]** 命令：
  - RHEL 8 中的 **pcs stonith [status]** 命令取代了 RHEL 7 中的 **pcs stonith [show]** 命令。
  - RHEL 8 中的 **pcs stonith config** 命令取代了 RHEL 7 中的 **pcs stonith [show] --full** 命令。
  - RHEL 8 中的 **pcs stonith config *resource id*** 命令取代了 RHEL 7 中的 **pcs stonith show *resource id*** 命令。

(BZ#1654280)

RHEL 8 中的 Pacemaker 2.0.0

pacemaker 软件包已升级到 Pacemaker 2.0.0 的上游版本，与之前的版本相比，它提供了一些程序错误修复和增强：

- Pacemaker 详情日志默认是 **/var/log/pacemaker/pacemaker.log** （不直接在 **/var/log** 中，或者与 **/var/log/cluster**下的 **corosync** 日志合并）。
- Pacemaker 守护进程进程已被重命名为更直观地读取日志。例如，**pengine** 已重命名为 **pacemaker-schedulerd**。
- 对已弃用的 **default-resource-stickiness** 和 **is-managed-default** 集群属性的支持已被丢弃。相反，**resource-stickiness** 和 **is-managed** 属性应该在资源默认值中设置。语法已弃用的现有配置（但不是新创建的配置）将自动更新，以使用支持的语法。
- 如需更完整的更改列表，请参阅 Red Hat Enterprise Linux 8 中的 Pacemaker 2.0 升级 。

建议在使用 Red Hat Enterprise Linux 7 或更早版本升级现有集群的用户，在所有集群节点上升级 RHEL 前后，在任何集群节点上运行 **pcs cluster cib-upgrade**。

(BZ#1543494)

重命名为可升级克隆资源的 master 资源

Red Hat Enterprise Linux (RHEL) 8 支持 Pacemaker 2.0，其中的 master/slave 资源不再是一个独立的资源类型，而是带有 **promotable** meta-attribute 设置为 **true** 的标准克隆资源。实施以下更改后，支持此次更新：

- 不再可以使用 **pcs** 命令创建 master 资源。相反，可以创建 **可升级的** 克隆资源。相关的关键字和命令已从 **master** 改为 **可升级**。
- 所有现有的 master 资源都显示为 promotable 克隆资源。
- 在 Web UI 中管理 RHEL7 集群时，master 资源仍被称为 master，因为 RHEL7 集群不支持可升级克隆。

(BZ#1542288)

用于验证集群中的节点的新命令

Red Hat Enterprise Linux(RHEL)8 对用来验证集群中节点的命令进行了以下更改。

- 新命令为 **pcs host auth**。用户可通过此命令指定主机名、地址和 **pcsd** 端口。
- **pcs cluster auth** 命令只验证本地集群中的节点，不接受节点列表
- 现在可以为每个节点指定地址。**pcs**/**pcsd** 随后将使用指定地址与每个节点通信。这些地址可能与 **corosync** 内部使用的不同。

- **pcs pcsd clear-auth** 命令已被 **pcs pcsd deauth** 和 **pcs host deauth** 命令替代。新命令允许用户验证单个主机以及所有主机。

- 在以前的版本中，节点身份验证是双向的，运行 **pcs cluster auth** 命令会导致所有指定节点相互身份验证。但是，**pcs host auth** 命令只使本地主机向指定节点进行身份验证。这样，在运行此命令时，可以更好地控制针对其他节点进行身份验证的节点。在集群设置本身，以及添加节点时，**pcs** 会自动同步集群上的令牌，因此集群中的所有节点仍然会像以前那样自动验证，且集群节点可以相互通信。

请注意,这些更改不是后向兼容的。在 RHEL 7 系统中验证的节点需要再次进行身份验证。

(BZ#1549535)

pcs 命令现在支持隔离历史记录的显示、清理和同步

Pacemaker 的隔离守护进程跟踪所有执行的隔离操作（待处理、成功和失败）。有了此版本，**pcs** 命令允许用户通过以下方式访问隔离历史记录：

- **pcs status** 命令显示失败和待处理的隔离操作

- **pcs status --full** 命令显示整个隔离历史记录

- **pcs stonith history** 命令提供显示和清理隔离历史记录的选项

- 虽然隔离历史记录是自动同步的，但 **pcs stonith history** 命令现在支持一个**update**选项，其允许用户在必要手手动同步隔离历史记录。

(BZ#1620190, BZ#1615891)

5.1.14. 网络

nftables 取代 **iptables** 作为默认的网络数据包过滤框架

nftables 框架提供数据包分类工具,它是 **iptables**、**ip6tables**、**arptables** 和 **ebtables** 工具的指定后台。与之前的数据包过滤工具相比，它在方便、特性和性能方面提供了大量改进，最重要的是：

- 查找表而不是线性处理

- **IPv4** 和 **IPv6** 使用同一个协议框架

- 规则会以原子方式应用，而不是提取、更新和存储完整的规则集

- 支持在规则集中的调试和追踪(**nftrace**)以及监控追踪事件（在 **nft** 工具中）

- 更加一致和压缩的语法，没有特定协议的扩展

- 用于第三方应用程序的 Netlink API

与 **iptables** 类似，**nftables** 使用表来存储链。链包含执行动作的独立规则。**nft** 工具取代了之前数据包过滤框架中的所有工具。**libnftables** 库可用于通过 **libmnl** 库与 **nftables** Netlink API 进行低级交互。

iptables、**ip6tables**、**ebtables** 和 **arptables** 工具被基于 nftables 的置入替换替换为同名。虽然外部行为与其旧的对等点相同，但它们在需要时通过兼容性接口在内部使用 **nftables** 与传统的 **netfilter** 内核模块。

可以使用 **nft list ruleset** 命令查看模块对 **nftables** 规则集的影响。由于这些工具将表、链和规则添加到 **nftables** 规则集，请注意 **nftables** 规则集操作（如 **nft flush ruleset** 命令）可能会影响使用之前独立的传统命令安装的规则集。

为了帮助快速识存在该工具的哪个变体，版本信息已被更新，其中包含了后端名称。在 RHEL 8 中，基于 nftables 的 **iptables** 工具打印以下版本字符串：

```
$ iptables --version
iptables v1.8.0 (nf_tables)
```

为了进行比较，如果存在旧的 **iptables** 工具，则会打印以下版本信息：

```
$ iptables --version
iptables v1.8.0 (legacy)
```

(BZ#1644030)

RHEL 8 中的主要 TCP 特性

Red Hat Enterprise Linux 8 提供了 TCP 网络堆栈版本 4.18，它提供更高的性能、更好的可伸缩性和稳定性。性能会提高，特别是对于入站连接率高的忙碌的 TCP 服务器的性能。

此外，还提供两种新的 TCP 拥塞算法： **BBR** 和 **NV**，提供比 cubic 更低的延迟和更好的吞吐量。

(BZ#1562998)

firewalld 默认使用 **nftables**

有了此更新，过滤子系统的 **nftables** 是 **firewalld** 守护进程默认的防火墙后端。要更改后端，请使用 **/etc/firewalld/firewalld.conf** 文件中的 **FirewallBackend** 选项。

在使用 **nftables** 时，这个更改的行为会有以下区别：

1. **iptables** 规则的执行总是在 **firewalld** 规则之前发生
   - **iptables** 中的 **DROP** 表示 **firewalld**从未看到数据包
   - **iptables** 中的 **ACCEPT** 表示数据包仍会遵守 **firewalld** 规则
2. **firewalld** 直接规则仍然通过 **iptables** 实施，而其他 **firewalld** 功能则使用 **nftables**
3. 直接执行规则是在 **firewalld** 通用接受已建立的连接前执行

(BZ#1509026)

RHEL 8 中的 **wpa_supplicant** 的显著变化

在 Red Hat Enterprise Linux(RHEL)8 中，**wpa_supplicant** 软件包是启用了 **CONFIG_DEBUG_SYSLOG** 构建的。这允许使用 **journalctl** 实用程序读取 **wpa_supplicant** 日志，而不必检查 **/var/log/wpa_supplicant.log** 文件的内容。

(BZ#1582538)

NetworkManager 现在支持 SR-IOV 虚拟功能

在 Red Hat Enterprise Linux 8.0 中，**NetworkManager** 允许为支持单根 I/O 虚拟化(SR-IOV)的接口配置虚拟功能(VF)的数量。另外，**NetworkManager** 允许配置 VF 的一些属性，如 MAC 地址、VLAN、**欺骗**检查设置以及允许的字节速率。请注意，与 SR-IOV 相关的所有属性都位于 **sriov** 连接设置中。详情请查

看 **nm-settings(5)手册** 页。

(BZ#1555013)

IPVLAN 虚拟网络驱动程序现在被支持

在 Red Hat Enterprise Linux 8.0 中,内核包含对 IPVLAN 虚拟网络驱动程序的支持。在这个版本中，IPVLAN 虚拟网络接口卡(NIC)为本地网络公开单个 MAC 地址的多个容器启用网络连接。这使得单个主机可以克服对同级网络设备支持的 MAC 地址数量可能存在的限制。

(BZ#1261167)

NetworkManager 支持连接的通配符接口名称匹配

在以前的版本中，只能使用接口名的完全匹配来限制到给定接口的连接。在这个版本中，连接具有支持通配符的新 **match.interface-name** 属性。在这个版本中，用户可以使用通配符模式以更灵活的方式选择连接的接口。

(BZ#1555012)

网络堆栈 4.18 的改进

Red Hat Enterprise Linux 8.0 包括了升级到上游版本 4.18 的网络堆栈，它提供了几个程序错误修复和增强。主要变更包括：

- 引入了新的卸载功能，如 **UDP_GSO**，以及某些设备驱动程序 **GRO_HW**。
- 改进了用户数据报协议(UDP)的显著可扩展性。
- 改进了通用忙碌轮询代码。
- 改进了 IPv6 协议的可扩展性。
- 路由代码的可扩展性得到改进。
- 添加了一个新的默认传输队列调度算法**fq_codel**，它可改进传输延迟。
- 提高了部分传输队列调度算法的可扩展性.例如： **pfifo_fast** 现在没有锁定。
- 通过移除垃圾回收内核线程和 ip 片段只在超时后过期，改进了 IP 重新装配单元的可扩展性。因此，DoS 下的 CPU 使用率要低得多，且最大可持续发展片段率会受到为 IP 回配单元配置的内存量的限制。

(BZ#1562987)

将 **iptables** 转换为 **nftables**的新工具

此更新添加了**iptables-translate** 和 **ip6tables-translate** 工具，来将现有的 **iptables** 或 **ip6tables** 规则转换为 **nftables** 的对等规则。请注意，一些扩展可能缺少响应的转换支持。如果存在这样的扩展，工具会输出带有 **#** 符号前缀的未转换的规则。例如：

```
| % iptables-translate -A INPUT -j CHECKSUM --checksum-fill
| nft # -A INPUT -j CHECKSUM --checksum-fill
```

此外，用户可以使用 **iptables-restore-translate** 和 **ip6tables-restore-translate** 工具来转换规则转储。请注意，在此之前，用户可以使用 **iptables-save** 或 **ip6tables-save** 命令来打印当前规则的转储。例如：

```
| % sudo iptables-save >/tmp/iptables.dump
```

```
| % iptables-restore-translate -f /tmp/iptables.dump
| # Translated by iptables-restore-translate v1.8.0 on Wed Oct 17 17:00:13 2018
| add table ip nat
| ...
```

(BZ#1564596)

使用 NetworkManager添加到 VPN 的新功能

在 Red Hat Enterprise Linux 8.0 中，**NetworkManager** 为 VPN 提供以下新功能：

- 支持互联网密钥交换版本 2(IKEv2)协议。
- 添加了一些更多 **Libreswan** 选项，如 **rightid**、**leftcert**、**narrowing**、**rekey**、**fragmentation** 选项。有关支持选项的详情，请查看 **nm-settings-libreswan** man page。
- 更新了默认密码。这意味着，当用户没有指定密码时，**NetworkManager-libreswan** 插件允许 **Libreswan** 应用程序选择系统默认密码。唯一的例外是在用户选择 IKEv1 主动模式配置时。在这种情况下，**ike = aes256-sha1;modp1536** 和 **eps = aes256-sha1** 值被传递给 **Libreswan**。

(BZ#1557035)

新数据块类型 I-DATA，添加到 SCTP

在这个版本中，在流控制传输协议(SCTP)中添加了一个新的数据块类型、**I-DATA** 和流调度程序。在以前的版本中，SCTP 会按照用户发送的顺序发送用户消息。因此，一个大型 SCTP 用户消息会阻止任何流中的所有其他信息，直到完全发送为止。使用 **I-DATA** 块时，传输序列号(TSN)字段不会被过载。因此，SCTP 现在可以以不同的方式调度流，**I-DATA** 允许用户消息交集(RFC 8260)。请注意，两个同级都必须支持 **I-DATA** 块类型。

(BZ#1273139)

NetworkManager 支持配置 ethtool offload 功能

在这个版本中，**NetworkManager** 支持配置 **ethtool** offload 功能，用户不再需要使用 init 脚本或 **NetworkManager** 分配程序脚本。现在,用户可以使用以下方法之一将下载功能配置为连接配置集的一部分：

- 使用 **nmcli** 工具
- 编辑 **/etc/NetworkManager/system-connections/** 目录中的密钥文件
- 编辑 **/etc/sysconfig/network-scripts/ifcfg-*** 文件

请注意，图形界面和 **nmtui** 程序目前不支持这个 特性。

(BZ#1335409)

RHEL 8 中的 TCP BBR 支持

现在，Red Hat Enterprise Linux(RHEL)8 支持一个新的 TCP 拥塞控制算法，即 Bottleneck Bandwidth 和 round-trip 时间(BBR)。BBR 尝试确定电压链路的带宽和往返用时(RTT)。大多数拥塞算法基于数据包丢失（包括 CUBIC，默认的 Linux TCP 拥塞控制算法），它们在高吞吐量链路上有问题。BBR 不直接响应丢失事件，它会调整 TCP 架构率使其与可用带宽匹配。TCP BBR 的用户应该在所有相关接口上切换到 **fq** 队列设置。

请注意，用户应明确使用 **fq** 而不是 **fq_codel**。

详情请查看 **tc-fq** man page。

(BZ#1515987)

lksctp-tools，RHEL 8 中的 1.0.18 版本

lksctp-tools 软件包版本 3.28 包括在 Red Hat Enterprise Linux(RHEL)8 中。主要改进和程序错误修复包括：

- 与 Travis CI 和覆盖范围扫描集成
- 支持 **sctp_peeloff_flags** 功能
- 指明哪些内核功能可用
- 覆盖范围扫描问题修复

(BZ#1568622)

在 RHEL 8 中默认将 SCTP 模块列入黑名单

为提高安全性，已将一组内核模块移到 **kernel-modules-extra** 软件包中。默认情况下不安装它们。因此，非 root 用户无法加载这些组件，因为它们会被默认列入黑名单。要使用其中一个内核模块，系统管理员必须安装 **kernel-modules-extra** 并明确删除模块黑名单。因此，非 root 用户将自动加载软件组件。

(BZ#1642795)

driverctl 0.101 中的显著变化

Red Hat Enterprise Linux 8.0 提供了 **driverctl** 0.101。此版本包括以下程序错误修复：

- **shellcheck** 警告已修复。
- bash-completion 被安装为 **driverctl** 而不是 **driverctl-bash-completion.sh**。
- 非 PCI 总线的 **load_override** 功能已修复。
- **driverctl** 服务在到达 **basic.target** systemd 目标前加载所有覆盖。

(BZ#1648411)

为 firewalld 添加了富规则优先级

priority 选项已添加到富规则中。这使得用户可以在规则执行期间定义所需的优先级顺序，并提供对富规则的更多高级控制。

(BZ#1648497)

RHEL 8 支持 NVMe over RDMA

在 Red Hat Enterprise Linux(RHEL)8 中，通过远程直接内存访问(RDMA)的 Nonvolatile Memory Express(NVMe)只支持 Infiniband、RoCEv2 和 iWARP（发起端模式）。

请注意，只有故障切换模式支持多路径。

其他限制：

- kdump 不支持 NVMe/RDMA。

- 不支持通过 RDMA 从 NVMe 设备引导。

(BZ#1680177)

nf_tables 后端不支持使用 dmesg进行调试

Red Hat Enterprise Linux 8.0 将 **thenf_tables** 后端用于防火墙，这些防火墙不支持使用 **dmesg** 实用程序来调试防火墙。要调试防火墙规则，请使用 **xtables-monitor -t** 或 **nft monitor trace** 命令解码规则评估事件。

(BZ#1645744)

Red Hat Enterprise Linux 支持 VRF

RHEL 8.0 中的内核支持虚拟路由和转发(VRF)。VRF 设备以及使用 **ip** 实用程序设置的规则可让管理员在 Linux 网络堆栈中创建 VRF 域。这些域隔离第 3 层的流量，因此管理员可以创建不同的路由表，并在一个主机上的不同 VRF 域内重复使用相同的 IP 地址。

(BZ#1440031)

iproute, RHEL 8 版本 4.18

iproute 软件包附带了 Red Hat Enterprise Linux(RHEL)8 中的 4.18 版本。最显著的变化是标记为 ethX:Y 的接口别名不再被支持，如 eth0:1。要临时解决这个问题，用户应删除别名后缀，该后缀是冒号，在输入 **ip link show** 之前是以下编号：

(BZ#1589317)

5.1.15. 安全性

RHEL 8.0 发行版本的 SWID 标签

要使用 ISO/IEC 19770-2:2015 机制启用 RHEL 8.0 安装的识别，软件识别(SWID)标签被安装在 **/usr/lib/swidtag/redhat.com/com.redhat.com/com.redhat.RHEL-8-<architecture>.swidtag** 和 **/usr/lib/swidtag/redhat.com/com.redhat.RHEL-8.0-<architecture>.wid.tag**中。这些标签的父目录也可以通过 **/etc/swid/swidtags.d/redhat.com** 符号链接找到。

SWID 标签文件的 XML 签名可使用 **xmlsec1 verify** 命令验证，例如：

```
xmlsec1 verify --trusted-pem /etc/pki/swid/CA/redhat.com/redhatcodesignca.cert
/usr/share/redhat.com/com.redhat.RHEL-8-x86_64.swidtag
```

代码签名证书颁发机构的证书也可以从客户门户网站上的 产品签名密钥 页面中获得。

(BZ#1636338)

默认应用系统范围的加密策略

crypto-policies 是 Red Hat Enterprise Linux 8 中的一个组件，它配置核心加密子系统，包括 TLS、IPsec、DNSSEC、Kerberos 和 SSH 协议。它提供一组小的策略，管理员可以使用 **update-crypto-policies** 命令选择这些策略。

DEFAULT 系统范围的加密策略为当前的威胁模型提供安全设置。它允许 TLS 1.2 和 1.3 协议，以及 IKEv2 和 SSH2 协议。如果 2047 位大于 2047，则接受 RSA 密钥和 Diffie-Hellman 参数。

如需更多信息，请参阅红帽博客上的 Red Hat Enterprise Linux 8 中的加密策略的一致安全性 文章和 **update-crypto-policies(8)** 手册页。

(BZ#1591620)

openssh rebase 到版本 7.8p1

openssh 软件包已升级到上游版本 7.8p1。主要变更包括：

- 删除了对 **SSH 版本 1** 协议的支持。
- 删除了对 **hmac-ripemd160** 消息验证代码的支持。
- 删除了对 RC4（**arcfour**）加密的支持。
- 删除了对 **Blowfish** 加密的支持。
- 删除了对 **CAST** 加密的支持。
- 将 **UseDNS** 选项的默认值改为**no**。
- 默认禁用 **DSA** 公钥算法。
- 将 **Diffie-Hellman** 参数的最小 modulus 大小改为 2048 字节。
- 更改了 **ExposeAuthInfo** 配置选项的语义。
- 现在，**usePrivilegeSeparation=sandbox** 选项是强制的且无法禁用。
- 最小的 **RSA** 密钥大小为 1024 位。

(BZ#1622511)

OpenSSH 服务器密钥生成现在由**sshd-keygen@.service** 处理

如果缺少 RSA、ECDSA 和 ED25519 服务器主机密钥，**OpenSSH** 会自动创建它们。要在 RHEL 8 中配置主机密钥创建，使用 **sshd-keygen@.service** 实例化服务。

例如，禁用自动创建 RSA 密钥类型：

```
# systemctl mask sshd-keygen@rsa.service
```

如需更多信息，请参阅 **/etc/sysconfig/sshd** 文件。

(BZ#1228088)

ECDSA 密钥支持 SSH 身份验证

此 **OpenSSH** 套件发行版本引进了对保存在 PKCS #11 智能卡中的 ECDSA 密钥的支持。现在，用户可以同时使用 RSA 和 ECDSA 密钥进行 SSH 身份验证。

(BZ#1645038)

libssh 实现 SSH 用于核心加密组件

在 Red Hat Enterprise Linux 8 中使用 **libssh** 作为核心加密组件。**libssh** 库实施 Secure Shell(SSH)协议。

请注意，**libssh** 的客户端遵循通过系统范围的加密策略为 **OpenSSH** 配置的设置，但服务器端的配置无法通过系统范围的加密策略进行更改。

(BZ#1485241)

加密库中的 TLS 1.3 支持

在这个版本中，在所有主要后端加密库中默认启用传输层安全(TLS)1.3。这通过利用新的算法（如 RSA-PSS 或 X25519）实现了跨操作系统通信层的低延迟并增强应用程序的隐私和安全。

(BZ#1516728)

NSS 现在默认使用 SQL

网络安全服务(NSS)库现在默认使用 SQL 文件格式作为信任数据库。DBM 文件格式用作之前版本中的默认数据库格式，它不支持多个进程同时访问同一数据库，它在上游中已被弃用。因此，使用 NSS 信任数据库存储密钥、证书和撤销信息的应用程序现在默认使用 SQL 格式创建数据库。尝试使用旧的 DBM 格式创建数据库会失败。现有 DBM 数据库以只读模式打开，它们会自动转换为 SQL 格式。请注意，自 Red Hat Enterprise Linux 6 之后，NSS 支持 SQL 文件格式。

(BZ#1489094)

PKCS #11 对智能卡和 HSM 的支持在系统中一致

在这个版本中，使用智能卡和带有 PKCS #11 加密令牌接口的硬件安全模块(HSM)会变得一致。这意味着用户和管理员可以对系统中所有相关工具使用相同的语法。主要改进包括：

- 支持 PKCS #11 Uniform Resource Identifier(URI)方案，用于确保为管理员和应用程序作者在 RHEL 服务器中启用令牌。
- 使用 **pkcs11.conf** 的智能卡和 HSMs 的系统范围注册方法。
- NSS、GnuTLS 和 OpenSSL（通过 **openssl-pkcs11** 引擎）应用程序提供了对 HSM 和智能卡的一致支持。
- Apache HTTP 服务器(**httpd**)现在无缝支持 HSM。

如需更多信息，请参阅 **pkcs11.conf(5)** man page。

(BZ#1516741)

Firefox 现在可用于系统范围内注册的 PKCS #11 驱动程序

Firefox Web 浏览器会自动加载 **p11-kit-proxy** 模块，并且通过 **pkcs11.conf** 文件在 **p11-kit** 中注册的每个智能卡会自动检测到。要使用 TLS 客户端身份验证，不需要额外的设置，当服务器请求智能卡的密钥时，会自动使用它们。

(BZ#1595638)

OpenSC现在支持 RSA-PSS

在这个版本中，在 **OpenSC** 智能卡驱动程序中添加了对 RSA-PSS 加密签名方案的支持。新方案启用客户端软件中支持 TLS 1.3 所需的安全加密算法。

(BZ#1595626)

RHEL 8 中 Libreswan 的显著变化

libreswan 软件包已升级到上游版本 3.27，它提供很多程序错误修复和增强。最显著的变化包括：

- 支持通过 **authby=rsa-sha2** 的 RSA-PSS (RFC 7427)，通过 **authby=ecdsa-sha2** 的 ECDSA (RFC 7427), 使用 **dh31** 关键字的 CURVE25519 ，通过 **chacha20_poly1305** 加密关键字的 IKE 和 ESP 的 CHACHA20-POLY1305 已为 IKEv2 协议添加。
- 从 **Libreswan** 中删除了对替代 KLIPS 内核模块的支持，因为上游完全弃用了 KLIPS。
- Diffie-Hellman 组不再支持 DH22、DH23 和 DH24（与 RFC 8247 不同）。

请注意，**authby=rsasig** 已更改为始终使用 RSA v1.5 方法， **authby=rsa-sha2** 选项使用 RSASSA-PSS 方法。按照 RFC 8247，**authby=rsa-sha1** 选项。这就是 **Libreswan** 不再支持带有数字签名的 SHA-1 的原因。

(BZ#1566574)

系统范围的加密策略将 Libreswan 中的默认 IKE 版本改为 IKEv2

Libreswan IPsec 实现中的默认 IKE 版本已从 IKEv1(RFC 2409)改为 IKEv2(RFC 7296)。用于 IPsec 的默认 IKE 和 ESP/AH 算法已更新，以符合系统范围的加密策略、RFC 8221 和 RFC 8247。现在，加密密钥大小为 256 位，优先于 128 位的密钥大小。

默认 IKE 和 ESP/AH 密码现在包括 AES-GCM、CHACHA20POLY1305 和 AES-CBC 用于加密。为了进行完整性检查，它们提供了 AEAD 和 SHA-2。Diffie-Hellman 组现在包含 DH19、DH20、DH21、DH14、DH15、DH16 和 DH18。

以下算法已从默认的 IKE 和 ESP/AH 策略中删除：AES_CTR、3DES、SHA1、DH2、DH5、DH22、DH23 和 DH24。除了 DH22、DH23 和 DH24 除外，这些算法可以通过 IPsec 配置文件中的 **ike=** 或 **phase2alg=/esp=/ah=** 选项启用。

要配置仍然需要 IKEv1 协议的 IPsec VPN 连接，请在连接配置文件中添加 **ikev2=no** 选项。详情请查看 **ipsec.conf(5)** man page。

(BZ#1645606)

Libreswan 中的 IKE 版本相关更改

在这个版本中，Libreswan 以不同的方式处理互联网密钥交换(IKE)设置：

- 默认互联网密钥交换(IKE)版本已从 1 改为 2。
- 连接现在可以使用 IKEv1 或 IKEv2 协议，但不能同时使用两者。
- 更改了对 **ikev2** 选项的解释：
  - **insist** 的值被解释为 IKEv2-only。
  - 值 **no** 和 **never** 解释为 IKEv1-only。
  - **propose** 的值、**yes** 和 **permit** 不再有效，并导致一个错误，因为它不知道这些值是从哪个 IKE 版本从生成的

(BZ#1648776)

RHEL 8 中的 OpenSCAP 的新功能

OpenSCAP 套件已升级到上游版本 1.3.0，它引进了许多与之前版本相比的改进。最显著的功能包括：

- API 和 ABI 已合并 - 已删除更新、弃用和/或未使用的符号。
- 探测不作为独立的进程运行，而是作为 **oscap** 进程中的线程运行。

- 命令行界面已被更新。
- **Python 2** 绑定已被 **Python 3** 绑定替代。

(BZ#1614273)

SCAP 安全指南 现在支持系统范围的加密策略

scap-security-guide 软件包已被更新，使用预定义的系统范围的加密策略来配置核心加密子系统。与系统范围的加密策略冲突或覆盖的安全内容已被删除。

请注意，这个更改仅适用于 **scap-security-guide 中的** 安全内容，您不需要更新 OpenSCAP 扫描程序或其他 SCAP 组件。

(BZ#1618505)

改进了 OpenSCAP 命令行界面

现在，所有 **oscap** 模块和子模块中都提供了详细模式。工具输出改进了格式。

已删除已弃用的选项，以提高命令行界面的可用性。

以下选项不再可用：

- **oscap xccdf generate report** 中的**--show** 已被完全删除。
- **oscap oval eval** 中的 **--probe-root** 已被删除。它可以通过设置环境变量 **OSCAP_PROBE_ROOT** 来替换。
- **oscap xccdf eval** 中的 **--sce-results** 已被 **--check-engine-results** 替代
- **validate-xml** 子模块已从 CPE、OVAL 和 XCCDF 模块中删除。 **validate** 可用来针对 XML 模式和 XSD 模式验证 SCAP 内容。
- **oscap oval list-probes** 命令已被删除，可用探测列表可以改为使用 **oscap --version** 显示。

OpenSCAP 允许使用 **--profile '(all)'** 来评估给定 XCCDF 基准中的所有规则，无论配置集是什么。

(BZ#1618484)

SCAP 安全指南 PCI-DSS 配置集与版本 3.2.1 一致

scap-security-guide 软件包为 Red Hat Enterprise Linux 8 提供 PCI-DSS（支付卡行业数据安全标准）配置文件，此配置文件已更新，以与最新的 PCI-DSS 版本 - 3.2.1 保持一致。

(BZ#1618528)

SCAP 安全指南支持 OSPP 4.2

scap-security-guide 软件包为 Red Hat Enterprise Linux 8 提供 OSPP（常规目的操作系统保护配置文件）配置文件版本 4.2 的一个草案。这个配置集反映了在 NIAP 配置中识别到 General Purpose Operating Systems（Protection Profile 版本 4.2）的保护配置集中指定的强制配置控制。SCAP 安全指南提供自动检查和脚本，帮助用户满足 OSPP 中定义的要求。

(BZ#1618518)

RHEL 8 中 rsyslog 的显著变化

rsyslog 软件包已升级到上游版本 8.37.0，它提供很多程序错误修复和增强。最显著的变化包括：

- 加强了 **rsyslog** 内部消息的处理；对它们进行速率限制的可能性；修复了可能的死锁。
- 总体上增强了速率限制；现在记录实际的 *垃圾邮件源*。
- 改进了对超大小消息的处理 - 用户现在可以设置如何在核心模块和具有独立操作的某些模块中处理它们。
- 现在可将 **mmnormalize** 规则基础嵌入到 **config** 文件中，而不是为其创建单独的文件。
- 现在，所有 **配置** 变量（包括 JSON 中的变量）都不区分大小写。
- PostgreSQL 输出的各种改进。
- 添加了可以使用 shell 变量控制 **config** 处理的可能性，例如，额外配置文件的有条件加载、执行语句，或在 **config** 中包含文本。请注意，过度使用此功能可能会导致 **rsyslog** 调试问题非常困难。
- 现在可以在 **config** 中指定 4 位文件创建模式。
- 现在，可靠的事件日志记录协议(RELP)输入只能在指定地址上绑定。
- 邮件输出的 **enable.body** 选项的默认值现在与文档一致
- 用户现在可以指定在 MongoDB 输出中应该忽略的插入错误代码。
- 并行 TCP(pTCP)输入现已是可配置的积压，以更好地负载平衡。
- 为了避免在 **journald** 轮转其文件时可能会出现的重复记录，添加了 **imjournal** 选项。请注意，使用这个选项可能会影响性能。

请注意，可以配置带有 **rsyslog** 的系统来提供更好的性能，如 配置没有 journald 或具有最小化 journald 使用的系统日志 知识库文章中所述。

(BZ#1613880)

新 rsyslog 模块： **omkafka**

要启用 **kafka** 集中式数据存储场景，您现在可以使用新的 **omkafka** 模块将日志转发到 **kafka** 基础架构。

(BZ#1542497)

rsyslog imfile 现在支持符号链接

有了这个更新，**rsyslog imfile** 模块提供了更好的性能和更多配置选项。这可让您将模块用于更复杂的文件监控用例。例如，您现在可以在配置的路径的任何位置使用带有 glob 模式的文件监视器，并使用提高数据吞吐量来轮转符号链接目标。

(BZ#1614179)

现在，默认的 **rsyslog** 配置文件格式为非传统的

rsyslog 软件包中的配置文件现在默认使用非传统的格式。旧格式仍可使用，但混合当前和旧版配置语句具有多个限制。需要检查来自以前的 RHEL 版本的配置。详情请查看 **rsyslog.conf(5)** 手册页。

(BZ#1619645)

Audit 3.0 使用 **auditd** 替换 **audispd**

在这个版本中，**audispd** 的功能已移到 **auditd**。因此，**audispd** 配置选项现在是 **auditd.conf** 的一部分。另外，**plugins.d** 目录已移到 **/etc/audit** 下。**auditd** 和它的插件的当前状态通过运行 **service auditd state** 命令检查。

(BZ#1616428)

tangd_port_t 允许更改 Tang 的默认端口

在这个版本中引进了 **tangd_port_t** SELinux 类型，允许 **tangd** 服务作为 SELinux 强制模式限制运行。这一更改有助于简化将 Tang 服务器配置为侦听用户定义的端口，同时还会将 SELinux 提供的安全级别保持在 enforcing 模式。

如需更多信息，请参阅使用 基于策略的解密配置加密卷的自动解锁 部分。

(BZ#1664345)

新 SELinux 布尔值

这个 SELinux 系统策略更新引进了以下布尔值：

- colord_use_nfs
- mysql_connect_http
- pdns_can_network_connect_db
- ssh_use_tcpd
- sslh_can_bind_any_port
- sslh_can_connect_any_port
- virt_use_pcscd

要获得包括布尔值的列表，并找出它们是否启用或禁用，请安装 **selinux-policy-devel** 软件包并使用：

```
# semanage boolean -l
```

(JIRA:RHELPLAN-10347)

SELinux 现在支持 systemd No new Privileges

此更新引进了 **nnp_nosuid_transition** 策略功能，它可在 **No New Privileges** (NNP)或 **nosuid** 下启用 SELinux 域转换，如果在旧上下文和新上下文之间允许 **nnp_nosuid_transition**。**selinux-policy** 软件包现在包含使用 **NNP** 安全功能的 **systemd** 服务的策略。

以下规则描述了允许该服务此功能：

```
allow source_domain  target_type:process2 { nnp_transition nosuid_transition };
```

例如：

```
allow init_t fprintd_t:process2 { nnp_transition nosuid_transition };
```

发行版策略现在还包含 m4 宏接口，可用于使用 **init_nnp_daemon_domain（）** 函数的服务的 SELinux 安全策略。

(BZ#1594111)

支持 mmap syscall 中的新映射权限检查

添加了 SELinux **map** 权限，以控制对文件、目录、套接字等的内存映射访问。这允许 SELinux 策略阻止直接访问各种文件系统对象，并确保每个此类访问都已重新验证。

(BZ#1592244)

SELinux 现在支持 process 类中的 getrlimit 权限

此更新引入了一个新的 SELinux 访问控制检查，**process:getrlimit**，并为 **prlimit（）** 函数进行了添加。这使得 SELinux 策略开发人员能够控制一个进程何时尝试读取，然后使用 **process:setrlimit** 权限修改另一个进程的资源限制。请注意，SELinux 不会限制通过 **prlimit（）** 操作自己资源限制的进程。如需更多信息，请参阅 **prlimit(2)** 和 **getrlimit(2)** man page。

(BZ#1549772)

selinux-policy 现在支持 VxFS 标签

这个版本引进了对 Veritas 文件系统(VxFS)安全扩展属性(xattrs)的支持。这允许在文件系统中为对象存储正确的 SELinux 标签，而不是通用 vxfs_t 类型。因此，完全支持 SELinux 的 VxFS 系统更安全。

(BZ#1483904)

编译时安全强化标志会以更一致的方式应用

在 RHEL 8 分发中的 RPM 软件包中，编译时安全强化标记会更加一致，**redhat-rpm-config** 软件包现在会自动提供安全强化标记。应用编译时间标志还有助于满足通用标准(CC)要求。应用以下安全强化标记：

- 用于检测 buffer-overflow 错误： **D_FORTIFY_SOURCE=2**
- 检查 C++ 数组、向量和字符串的标准库强化： **D_GLIBCXX_ASSERTIONS**
- 对于 Stack Smashing Protector(SSP)： **fstack-protector-strong**
- 增强例外： **fexceptions**
- 对于 Control-Flow Integrity(CFI)： **fcf-protection=full** （仅在 AMD 和 Intel 64 位构架中）
- 对于地址空间布局随机化(ASLR)： **fPIE** （对于可执行文件）或 **fPIC** （用于库）
- 对于保护 Stack Clash 漏洞：**fstack-clash-protection** (ARM)
- 在启动时解析所有符号的链接标志： **-Wl,-z,now**

如需更多信息，请参阅 **gcc(1)** 手册页。

(JIRA:RHELPLAN-2306)

5.1.16. 虚拟化

RHEL 8 中的 qemu-kvm 2.12

Red Hat Enterprise Linux 8 与 **qemu-kvm** 2.12 一起分发。这个版本修复了多个程序错误，并在 Red Hat Enterprise Linux 7 中提供 1.5.3 版本中添加了许多改进。

值得注意的是，引入了以下功能：

- Q35 客户机机器类型
- UEFI 客户机引导
- guest 中的 NUMA 调整和固定
- vCPU 热插拔
- 客户机 I/O 线程

请注意，Red Hat Enterprise Linux 8 不支持 **qemu-kvm** 2.12 中的一些功能。详情请查看红帽客户门户网站中的"RHEL 8 虚拟化功能支持和限制"。

(BZ#1559240)

虚拟化现在支持 Q35 机器类型

Red Hat Enterprise Linux 8 引入了对 **Q35** 的支持，它是一种基于更现代 PCI Express 的机器类型。这在虚拟设备的特性和性能方面提供了各种改进，并确保更广泛的现代设备与虚拟化兼容。另外，Red Hat Enterprise Linux 8 中创建的虚拟机被设置为默认使用 **Q35**。

另请注意，以前的默认 **PC** 机器类型已被弃用，只有在虚拟化不支持 Q35 的旧操作系统时使用。

(BZ#1599777)

复制后虚拟机迁移

RHEL 8 使执行 KVM 虚拟机(VM)的复制后迁移成为可能。使用时，复制后迁移会暂停在源主机上迁移虚拟机的 vCPU，仅传输最小的内存页，激活目标主机上虚拟机的 vCPU，并在虚拟机在目标上运行时传输剩余的内存页。

这可显著降低迁移的虚拟机的停机时间，还可保证迁移完成，无论源虚拟机的内存页如何快速变化。因此，它适合迁移持续大量使用的虚拟机，而使用标准的预复制迁移无法迁移。

(JIRA:RHELPLAN-14323)

KVM 虚拟化现在支持 virtio-gpu

已为 KVM 虚拟机引进了 **virtio-gpu** 显示设备。**virtio-gpu** 提高了虚拟机图形性能，并可以实现虚拟 GPU 设备的各种增强功能。

(JIRA:RHELPLAN-14329)

KVM 在 RHEL 8 中支持 UMIP

KVM 虚拟化现在支持用户模型讲师(UMIP)功能，这有助于防止用户空间应用程序访问系统范围设置。这减少了权限升级攻击的潜在向量，因此 KVM 管理程序及其客户机机器更安全。

(BZ#1494651)

KVM 客户机崩溃报告中的其他信息

如果客户机意外终止或变得无响应，则 KVM 管理程序生成的崩溃信息已扩展。这使得诊断和修复 KVM 虚拟化部署中的问题变得更加简单。

(BZ#1508139)

NVIDIA vGPU 现在与 VNC 控制台兼容

使用 NVIDIA 虚拟 GPU(vGPU)功能时，现在可以使用 VNC 控制台来显示客户机的视觉输出。

(BZ#1497911)

Ceph 由虚拟化支持

在这个版本中，Red Hat 支持的所有 CPU 架构上的 KVM 虚拟化均支持 Ceph 存储。

(BZ#1578855)

IBM Z 中 KVM 虚拟机的交互式引导装载程序

在 IBM Z 主机上引导 KVM 虚拟机时，QEMU 引导装载程序固件现在可以显示客户机操作系统的互动控制台界面。这样便可在不访问主机环境的情况下对 guest OS 引导问题进行故障排除。

(BZ#1508137)

虚拟机支持 IBM z14 ZR1

KVM 系统管理程序现在支持 IBM z14 ZR1 服务器的 CPU 模型。这可在 IBM Z 系统上运行的 KVM 虚拟机中使用此 CPU 的功能。

(BZ#1592337)

KVM 支持 IBM Z 上的 Telnet 3270

当将 RHEL 8 用作 IBM Z 系统中的主机时，现在可以使用 **Telnet 3270** 客户端连接到主机上的虚拟机。

(BZ#1570029)

添加了 QEMU 沙盒

在 Red Hat Enterprise Linux 8 中，QEMU 模拟器引入了沙盒功能。QEMU 沙盒为调用 QEMU 可以执行的系统提供可配置的限制，从而使虚拟机更安全。请注意，这个功能会被默认启用和配置。

(JIRA:RHELPLAN-10628)

PV TLB Flush Hyper-V enlightenment

RHEL 8 添加 **PV TLB Flush** Hyper-V Enlightenment 功能。这提高了在 KVM hypervisor 上过度承诺环境中运行的 Windows 虚拟机(VM)的性能。

(JIRA:RHELPLAN-14330)

IBM POWER 中 KVM 虚拟机的新机器类型

为 IBM POWER 8 和 IBM POWER 9 系统上运行的 KVM 系统管理程序启用了多个新的 rhel-pseries 机器类型。这使得在 IBM POWER 系统的 RHEL 8 中托管的虚拟机(VM)可以正确使用这些机器类型的 CPU 功能。此外，这还允许将 IBM POWER 上的虚拟机迁移到最新版本的 KVM 系统管理程序。

(BZ#1585651, BZ#1595501)

为 Intel Xeon SnowRidge 启用 GFNI 和 CLDEMOT 指令集

在 Intel Xeon SnowRidge 系统上的 RHEL 8 主机上运行的虚拟机(VM)现在可以使用 GFNI 和 CLDEMOT 指令集。在某些情况下，这可能会显著提高此类虚拟机的性能。

(BZ#1494705)

为 OVMF 启用 IPv6

现在，Open Virtual Machine Firmware(OVMF)上启用了 IPv6 协议。这使得使用 OVMF 的虚拟机可以利用 IPv6 提供的各种网络引导改进。

(BZ#1536627)

添加了 NVMe 设备的基于 VFIO 的块驱动程序

QEMU 模拟器为 Non-volatile Memory Express(NVMe)设备引入了基于虚拟功能 I/O(VFIO)的驱动程序。驱动程序直接与附加到虚拟机(VM)的 NVMe 设备通信，并避免使用内核系统层及其 NVMe 驱动程序。因此，这会提高虚拟机中 NVMe 设备的性能。

(BZ#1519004)

支持 Hyper-V Generic UIO 驱动程序

RHEL 8 现在支持 Hyper-V Generic 用户空间 I/O(UIO)驱动程序的多频道功能。这使得在 Hyper-V 管理程序上运行的 RHEL 8 虚拟机可以使用 Data Plane Development Kit(DPDK)Netvsc Poll Mode 驱动程序(PMD)，它可增强这些虚拟机的网络功能。

但请注意，Netvsc 接口状态当前显示为 Down，即使它正在运行并且可以使用。

(BZ#1650149)

改进了巨页支持

当使用 RHEL 8 作为虚拟化主机时，用户可以将虚拟机(VM)的内存页大小修改为 CPU 支持的任何大小。这可以显著提高虚拟机的性能。

要配置虚拟机内存页面的大小，请编辑虚拟机的 XML 配置，并将 <hugepages> 元素添加到 <memoryBacking> 部分。

(JIRA:RHELPLAN-14607)

POWER 9 主机上的虚拟机可以使用 THP

在 IBM POWER 9 架构上运行的 RHEL 8 主机中，虚拟机(VM)功能得益于透明大内存页(THP)功能。THP 可让主机内核动态为进程分配大内存页面，从而提高具有大量内存的虚拟机性能。

(JIRA:RHELPLAN-13440)

5.1.17. 支持性

sosreport 可以报告基于 eBPF 的程序和映射

sosreport 工具已被改进来报告 Red Hat Enterprise Linux 8 载入的扩展 Berkeley Packet Filtering(eBPF)程序和映射。

(BZ#1559836)

5.2. 程序错误修复

这部分论述了 Red Hat Enterprise Linux 8.0 中修复的错误，它们对用户有严重影响。

5.2.1. Desktop

PackageKit 现在可在 rpm 软件包上运行

在这个版本中，在 **rpm** 软件包上操作的支持已添加到 **PackageKit** 中。

(BZ#1559414)

5.2.2. 图形基础结构

QEMU 无法正确处理 8 字节 ggtt 条目

QEMU 有时会将一个 8 字节的 **ggtt** 条目写入到两个连续的 4 字节写入操作中。这些部分写入各自可能会触发单独的主机 **ggtt** 写入。有时，两个 **ggtt** 写被错误地合并。因此，转换到机器地址会失败，并会出现错误日志。

(BZ#1598776)

5.2.3. 身份管理

企业安全客户端使用 opensc 库进行令牌检测

Red Hat Enterprise Linux 8.0 仅支持用于智能卡的 **opensc** 库。有了此更新，企业安全客户端(ESC)使用 **opensc** 而不是删除的 **coolkey** 库进行令牌检测。因此，应用程序可以正确地检测到支持的令牌。

(BZ#1538645)

证书系统现在支持轮转调试日志

在以前的版本中，证书系统使用自定义日志记录框架，它不支持日志轮转。因此，调试日志（如 **/var/log/pki/*instance_name*/ca/debug** ）无限增长。在这个版本中，证书系统使用 **java.logging.util** 框架，它支持日志轮转。因此，您可以在 **/var/lib/pki/*instance_name*/conf/logging.properties** 文件中配置日志轮转。

有关日志轮转的更多信息，请参阅 **java.util.logging** 软件包的文档。

(BZ#1565073)

证书系统不再记录服务启动时 SetAllPropertiesRule 操作警告

在以前的版本中，当服务启动时，证书系统会在 **/var/log/messages** 日志文件中记录 **SetAllPropertiesRule** 操作的警告信息。这个问题已被解决，上面提到的警告不再被记录。

(BZ#1424966)

证书系统 KRA 客户端正确地解析 Key Request 响应

在以前的版本中，证书系统切换到新的 JSON 库。因此，某些对象的序列化会有所不同，Python 密钥恢复颁发机构(KRA)客户端无法解析 **Key Request** 响应。客户端已被修改，以支持使用旧和新的 JSON 库的响应。因此，Python KRA 客户端会正确解析 **密钥请求** 响应。

(BZ#1623444)

5.2.4. 编译器和开发工具

GCC 不再针对出站访问产生假的正警告

在以前的版本中，当使用 **-O3** 优化级别选项编译时，GNU Compiler Collection(GCC)偶尔会返回有关出站访问的假警告，即使编译的代码没有包含它。这个优化已被修复，GCC 不再显示假的正警告。

(BZ#1246444)

ltrace 可正确显示大型结构

在以前的版本中，**ltrace** 工具无法正确打印函数返回的大型结构。**ltrace** 中大型结构的处理已得到改进，它们现在可以正确打印。

(BZ#1584322)

GCC 内置功能 __builtin_clz 在 IBM Z 上返回正确的值

在以前的版本中，GCC 编译器错误地折叠 IBM Z 架构的 **FLOGR** 指令。因此，使用这个指令的 **__builtin_clz** 功能可能会在使用 **-funroll-loops** GCC 选项编译代码时返回错误的结果。这个程序错误已被解决，这个功能现在提供了正确的结果。

(BZ#1652016)

GDB 在批处理模式中的最后一个命令失败时提供非零退出状态

在以前的版本中，在以批处理模式运行时，GDB 始终以状态 **0** 退出，无论命令中出现什么错误。因此，无法确定命令是否成功。这个行为已被更改，GDB 现在会在上一命令发生错误时退出状态 **1**。这样可保持与之前执行所有命令的行为的兼容性。因此，现在可以确定 GDB 批处理模式是否成功执行。

(BZ#1491128)

5.2.5. 文件系统和存储

更高的打印级别不再导致 iscsiadm 意外终止

在以前的版本中，当用户使用 **--print** 或 **-P** 选项指定大于 0 的打印级别时，**iscsiadm** 程序会意外终止。这个问题已被解决，所有打印级别现在都可以正常工作。

(BZ#1582099)

当 multipathd 无法获取路径的 WWID 时，它不再禁用该路径

在以前的版本中，**multipathd** 服务会把一个路径的 WWID 视为获得空 WWID 的失败尝试。如果 **multipathd** 无法获取路径的 WWID，有时会禁用该路径。

在这个版本中，如果在检查是否改变时无法获取 WWID，则 **multipathd** 会继续使用旧的 WWID。

因此，当当检查 WWID 是否已更改时，当 **multipathd** 不能得到 WWID 时，它不再禁用路径。

(BZ#1673167)

5.2.6. 高可用性和集群

新的 /etc/sysconfig/pcsd 选项拒绝客户端发起的 SSL/TLS 重新协商

当在服务器上启用 TLS 重新协商时，客户端可以发送重新协商请求，该请求将启动新的握手。握手的计算要求在服务器上高于客户端。这使得服务器易受 DoS 攻击。在这个版本中，**/etc/sysconfig/pcsd** 配置文件中设置 **PCSD_SSL_OPTIONS** 会接受 **OP_NO_RENEGOTIATION** 选项来拒绝重新协商。请注意，客户端仍然可以打开多个与在所有服务器中执行握手的服务器的连接。

(BZ#1566430)

删除的集群节点不再显示在集群状态中

在以前的版本中，当使用 **pcs cluster node remove** 命令删除一个节点时，删除的节点会在 **pcs status** 显示的输出中保持可见。在这个版本中，删除的节点不再显示在集群状态中。

(BZ#1595829)

现在可以使用更新的首选参数名称或已弃用的参数名称来配置隔离代理

大量隔离代理参数已被重命名，而旧参数名称仍被支持为已弃用。在以前的版本中，**pcs** 无法设置新参数，除非与 **--force** 选项一同使用。在这个版本中，pc **s** 现在支持重命名的隔离代理参数，同时保持对已弃用参数的支持。

(BZ#1436217)

pcs 命令现在可以正确地读取集群的 **XML** 状态。

pcs 命令以 XML 格式运行 **crm_mon** 实用程序来获取群集的状态。**crm_mon** 实用程序将 XML 打印到标准输出，并将警告打印到标准错误输出。以前，**pcs** 将 XML 和警告混合到一个流中，无法将其作为 XML 解析。在这个版本中，标准和错误输出在 **pcs** 中分离，并读取集群的 XML 状态可以正常工作。

(BZ#1578955)

在从现有集群创建新集群时，用户不再建议销毁集群

在以前的版本中，当用户在运行 **pcs cluster setup** 命令时指定现有集群中的节点或者使用 **pcsd** Web UI 创建集群时，pcs 将其报告为一个错误，并建议用户销毁节点上的集群。因此，用户会销毁节点上的集群，破坏节点所属的集群节点，因为剩余的节点仍会将销毁的节点视为集群的一部分。在这个版本中，建议用户从其集群中删除节点，从而更好地通知他们如何在不中断集群的情况下解决问题。

(BZ#1596050)

pcs 命令不再以交互方式请求凭证

当非 root 用户运行需要 root 权限的 **pcs** 命令时，**pcs** 会连接到本地运行的 **pcsd** 守护进程，并将命令传给它，因为 **pcsd** 守护进程使用 root 权限来运行，所以可以运行该命令。在以前的版本中，如果用户没有认证到本地 **pcsd** 守护进程，则 **pcs** 会要求输入用户名和密码。这给用户造成混淆，要求在运行 **pcs** 的脚本中进行特殊处理。在这个版本中，如果用户未通过身份验证，则 **pcs** 退出并显示应执行的操作： 无论是以 root 身份运行 **pcs**，还是使用新的 **pcs client local-auth** 命令进行身份验证。因此，**pcs** 命令不会以互动方式询问凭证，改进用户体验。

(BZ#1554310)

现在，当 **crypto-policies** 被设置为 **FUTURE** 时，**pcsd** 守护进程会使用其默认自生成的 **SSL** 证书启动。

FUTURE 的 **crypto-policies** 设置需要 SSL 证书中的 RSA 密钥至少为 3072b 长。在以前的版本中，当设置了此策略时，**pcsd** 守护进程不会启动，因为它使用 2048b 密钥生成 SSL 证书。有了这个更新，**pcsd** 自生成的 SSL 证书的密钥大小已增加到 3072b，并且 **pcsd** 现在使用其默认自生成的 SSL 证书启动。

(BZ#1638852)

pcsd 服务现在在网络就绪时启动

在以前的版本中，当用户配置了 **pcsd** 来绑定到特定 IP 地址，且当 **pcsd** 试图启动时，地址在启动过程中未准备好，则 **pcsd** 无法启动，需要手动干预来启动 **pcsd**。有了此修复，**pcsd.service** 依赖于 **network-online.target**。因此，当网络就绪且可绑定到 IP 地址时，**pcsd** 可以启动。

(BZ#1640477)

5.2.7. 网络

glib-networking不再允许弱 **TLS** 算法

在以前的版本中，**glib-networking** 软件包与 RHEL 8 系统范围的 Crypto 策略不兼容。因此，使用 **glib** 库进行联网的应用程序可能会允许使用较弱算法的传输层安全(TLS)连接。有了此更新，会应用系统范围的加密策略，现在为网络使用 **glib** 的应用程序只允许按照策略可接受的 TLS 连接。

(BZ#1640534)

5.2.8. 安全性

SELinux 策略现在允许 iscsiuio 进程连接到发现门户

在以前的版本中，对于 **iscsiuio** 进程，SELinux 策略的限制太强，这些进程无法使用 **mmap** 系统调用访问 **/dev/uio*** 设备。因此，到发现门户的连接会失败。此更新在 SELinux 策略中添加了缺少的规则，在上述场景中，**iscsiuio** 进程可以按预期正常工作。

(BZ#1626446)

5.2.9. 订阅管理

dnf 和 yum 现在无论 subscription-manager 值都可以访问存储库

在以前的版本中，**dnf** 或 **yum** 命令会忽略 subscription-manager 服务添加的 URL 中的 **https://** 前缀。更新的 **dnf** 或 **yum** 命令不会忽略无效的 **https://** URL。因此，**dnf** 和 **yum** 无法访问存储库。为解决这个问题，新的配置变量 **proxy_scheme** 已添加到 **/etc/rhsm/rhsm.conf** 文件中，其值可以设置为 **http** 或 **https**。如果没有指定值，subscription-manager 会使用更常用的默认值设置 **http**。

请注意，如果代理使用 **http**，大多数用户不应更改 **/etc/rhsm/rhsm.conf** 中的配置中的任何内容。如果代理使用 **https**，用户应将 **proxy_scheme** 的值更新为 **https**。然后，在这两种情况下，用户都需要运行 **subscription-manager repos --list** 命令，或等待 **rhsmcertd** 守护进程进程正确重新生成 **/etc/yum.repos.d/redhat.repo**。

(BZ#1654531)

5.2.10. 虚拟化

在 Azure 上挂载临时磁盘现在可以更可靠

在以前的版本中，如果虚拟机"停止（取消分配）"，然后启动，在 Microsoft Azure 平台上运行的虚拟机(VM)中挂载临时磁盘会失败。在这个版本中，在上述情况下可以正确处理磁盘重新连接，这可防止问题的发生。

(BZ#1615599)

5.3. 技术预览

这部分提供了 Red Hat Enterprise Linux 8.0 中所有可用的技术预览列表。

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。

5.3.1. 内核

eBPF 作为技术预览

扩展的 Berkeley 数据包过滤(eBPF)功能作为一个技术预览用于网络和追踪。**eBPF** 可让用户空间将自定义程序附加到多个点（套接字、追踪点、数据包接收）上，从而接收和处理数据。该功能包括一个新的系统调用 **bpf（）**，它支持创建各种映射类型，并将各种类型的程序插入到内核中。请注意，只有具有

CAP_SYS_ADMIN 能力的用户（如一个 root 用户）才可以成功使用 **bpf()** syscall。更多信息，请参阅 **bpf**(2) man page。

(BZ#1559616)

BCC 作为技术预览提供

BPF Compiler Collection(BCC) 是一个用户空间工具工具包，用于创建高效的内核跟踪和操作程序，在 Red Hat Enterprise Linux 8 中作为技术预览提供。**BCC** 提供用于使用 **扩展 Berkeley Packet 过滤 (eBPF)** 的 Linux 操作系统的 I/O 分析、网络和监控工具。

(BZ#1548302)

Control Group v2 在 RHEL 8 中作为技术预览提供

控制组 v2 机制是统一的层次结构控制组。**控制**组 v2 以分层方式组织进程，并以受控和可配置的方式在层次结构中分发系统资源。

与上一版本不同，Control Group v2 只具有一个层次结构。这个单一层次结构使 Linux 内核能够：

- 根据拥有者的角色对进程进行分类。
- 解决多个分级冲突策略的问题。

控制组 v2 支持大量控制器：

- CPU 控制器规定了 CPU 周期的分布。这个控制器实现：
  - 常规调度策略的权重和绝对带宽限制模型。
  - 实时调度策略的绝对带宽分配模型。
- 内存控制器规定了内存分布。目前，会追踪以下类型的内存用量：
  - Userland memory - 页面缓存和匿名内存。
  - 内核数据结构，如密度和内节点。
  - TCP 套接字缓冲。
- I/O 控制器规定了 I/O 资源的分配。
- 回写控制器与 Memory 和 I/O 控制器交互，并且特定于 Control Group v2。

以上信息基于链接 ：https://www.kernel.org/doc/Documentation/cgroup-v2.txt。您可以参考同一链接来获取特定的 Control Group v2 控制器的更多信息。

(BZ#1401552)

在 Red Hat Enterprise Linux 8 中， 早期 kdump 作为技术预览提供

早期 kdump 功能允许崩溃内核和 initramfs 加载足够早的负载，以便在早期崩溃时捕获 **vmcore** 信息。有关 **early-kdump** 的详情，请查看 **/usr/share/doc/kexec-tools/early-kdump-howto.txt** 文件。

(BZ#1520209)

ibmvnic 设备驱动程序作为技术预览提供

使用 Red Hat Enterprise Linux 8.0 ， IBM POWER 架构的 IBM Virtual Network Interface

Controller(vNIC)驱动程序 **ibmvnic** 作为技术预览提供。vNIC 是一个提供企业功能并简化网络管理的 PowerVM 虚拟网络技术。当与 SR-IOV NIC 结合，在虚拟 NIC 级别提供带宽控制服务质量(QoS)功能时，它是一种高性能、有效的技术。vNIC 可以显著减少虚拟化开销，从而降低延迟，减少服务器资源，包括网络虚拟化所需的 CPU 和内存。

(BZ#1524683)

soft-RoCE 作为技术预览提供

通过融合以太网的远程直接内存访问(RDMA)是一种网络协议，其通过以太网实现 RDMA。soft-RoCE 是 RoCE 的软件实施，它支持两种协议版本：RoCE v1 和 RoCE v2。在 RHEL 8 中，Soft-RoCE 驱动程序 **rdma_rxe** 作为不受支持的技术预览提供。

(BZ#1605216)

5.3.2. 图形基础结构

VNC 远程控制台作为 64 位 ARM 架构的一个技术预览提供

在 64 位 ARM 架构中,虚拟网络计算(VNC)远程控制台可作为技术预览使用。请注意,在 64 位 ARM 架构中,目前图形堆栈的其它部分没有被验证。

(BZ#1698565)

5.3.3. 硬件启用

集群感知的 MD RAID1 作为技术预览提供。

默认情况下，在内核空间中不启用 RAID1 集群。如果要尝试使用 RAID1 集群，首先需要构建将 RAID1 集群作为模块的内核，使用以下步骤：

1. 输入 **make menuconfig** 命令。
2. 输入 **make && make modules && make modules_install && make install** 命令。
3. 输入 **reboot** 命令。

(BZ#1654482)

5.3.4. 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）服务器现在支持 DNS 安全扩展（DNSSEC），这是一组增强 DNS 协议安全性的 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转的。

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 Operational Practices, Version 2: http://tools.ietf.org/html/rfc6781#section-2
- Secure Domain Name System (DNS) Deployment Guide: http://dx.doi.org/10.6028/NIST.SP.800-81-2
- DNSSEC Key Rollover Timing Considerations: http://tools.ietf.org/html/rfc7583

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。

(BZ#1664718)

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 Red Hat Enterprise Linux 7.3 中，IdM API 被改进来启用多个 API 命令版本。在以前的版本中，增强功能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)。

(BZ#1664719)

5.3.5. 文件系统和存储

Aero 适配器作为技术预览提供

以下 Aero 适配器作为技术预览提供：

- PCI ID 0x1000:0x00e2 和 0x1000:0x00e6，由 **mpt3sas** 驱动程序控制
- PCI ID 0x1000:Ox10e5 和 0x1000:0x10e6，由 **megaraid_sas** 驱动程序控制

(BZ#1663281)

Stratis 现在可用

Stratis 是一个新的本地存储管理器。它在存储池的上面为用户提供额外的功能。

Stratis 可让您更轻松地执行存储任务，比如：

- 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：建立 Stratis 文件系统。

（JIRA:RHELPLAN-1212）

OverlayFS

OverlayFS 是一种联合文件系统。它允许您在另一个文件系统上覆盖一个文件系统。更改记录在上面的文件系统中，而较小的文件系统则未修改。这允许多个用户共享文件系统镜像，如容器或 DVD-ROM，基础镜像使用只读介质。

在大多数情况下，OverlayFS 仍是一个技术预览。因此，当这个技术被激活时，内核会记录警告信息。

与支持的容器引擎（**podman**、**cri-o** 或 **buildah**）一同使用时，对 OverlayFS 提供的全面支持包括以下限制：

- OverlayFS 仅支持作为容器引擎图形驱动程序或其他专用用例使用，如 squashed **kdump** initramfs。它主要用于容器 COW 内容，不支持持久性存储。您必须将任何持久性存储放在非 OverlayFS 卷中。您只能使用默认容器引擎配置：一个级别的覆盖、一个较低 dir 以及较低级别和上一级都位于同一个文件系统中。
- 目前只支持 XFS 作为较低层文件系统使用。

另外，以下规则和限制适用于使用 OverlayFS:

- OverlayFS 内核 ABI 和用户空间的行为被视为不稳定，将来的更新可能会改变。
- OverlayFS 提供一组受限的 POSIX 标准。在使用 OverlayFS 部署前，先测试您的应用程序。以下情况与 POSIX 不兼容：
  - **O_RDONLY** 打开的较低文件在读取文件时不会接收 **st_atime** 更新。
  - 使用 **O_RDONLY** 打开的较低文件，然后与 **MAP_SHARED** 映射与后续修改不一致。
  - RHEL 8 中不默认启用完全兼容 **st_ino** 或 **d_ino** 值，但您可以使用模块选项或挂载选项为它们启用完整的 POSIX 合规性。
    要获得一致的内节点编号，请使用 **xino=on** 挂载选项。

    您还可以使用 **redirect_dir=on** 和 **index=on** 选项提高 POSIX 合规性。这两个选项使上层的格式与没有这些选项的 overlay 不兼容。也就是说，如果您使用 **redirect_dir=on** 或 **index=on** 创建覆盖，卸载覆盖，然后在没有这些选项的情况下挂载覆盖，则可能会出现意外的结果或错误。
- 要确定现有 XFS 文件系统是否有资格用作 overlay，请使用以下命令查看是否启用了 **ftype=1** 选项：

```
# xfs_info /mount-point | grep ftype
```

- 使用 OverlayFS 在所有支持的容器引擎中默认启用 SELinux 安全标签。
- 本发行版本中与 OverlayFS 相关的几个已知问题。详情请查看 Linux 内核文档 中的 *非标准行为*。

有关 OverlayFS 的更多信息，请参阅 Linux 内核文档。

（BZ#1690207）

现在 ext4 和 XFS 作为技术预览提供文件系统 DAX

在 Red Hat Enterprise Linux 8.0 中，文件系统 DAX 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内存，通常使用一个或多个非线内存模块 (NVDIMM)，且必须在 NVDIMM 上创建支持 DAX 的文件系统。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一个文件 **mmap** 会把存储直接映射到应用程序的地址空间中。

(BZ#1627455)

5.3.6. 高可用性和集群

pacemaker podman bundles 作为技术预览

pacemaker 容器捆绑包现在在 **podman** 容器平台上运行，容器捆绑包功能作为一个技术预览可用。其中一个功能例外于技术预览：红帽完全支持在 Red Hat Openstack 中使用 Pacemaker 捆绑包。

（BZ#1619620）

5.3.7. 网络

XDP 作为技术预览提供

eXpress Data Path（XDP）功能作为技术预览提供。它提供了在内核入口数据路径早期为高性能数据包处理附加扩展 Berkeley Packet Filter（eBPF）程序的方法，从而可以进行高效的可编程数据包分析、过滤和操作。

(BZ#1503672)

用于 tc 的 eBPF 作为技术预览

作为技术预览，流量控制(tc)内核子系统和 **tc** 工具附加扩展 Berkeley Packet 过滤(eBPF)程序作为数据包分类器，以及用于入口和出站队列规则的操作。这可实现内核网络数据路径内的可编程数据包处理。

(BZ#1699825)

AF_XDP 作为技术预览

Address Family eXpress Data Path (**AF_XDP**) 是设计用于处理高性能数据包。它包含 **XDP**，并允许通过编程方式将选定的数据包高效地重定向到用户空间应用，以便进一步处理。

（BZ#1633143）

KTLS 作为技术预览提供

在 Red Hat Enterprise Linux 8 中，KTLS 是作为技术预览提供的。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 还提供将 TLS 记录加密卸载到支持此功能的网络接口控制器(NIC)的接口。

(BZ#1570255)

TIPC 作为技术预览提供

Transparent Inter Process Communication(**TIPC**)是一种专门为松散配对节点集群内的高效通信而设计的协议。它可作为内核模块使用，并在 **iproute2** 软件包中提供了一个 **tipc** 工具，以便设计人员能够快速并可靠地创建与其它应用程序通信的应用程序，而无论它们在集群中的位置。此功能作为技术预览提供。

(BZ#1581898)

systemd-resolved 服务现在作为技术预览提供

systemd-resolved 服务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）以及多播 DNS 解析器和响应程序。

请注意，即使 **systemd** 软件包提供了 **systemd -resolved**，这个服务仍是一个不受支持的技术预览。

(BZ#1906489)

5.3.8. Red Hat Enterprise Linux System Roles

RHEL 系统角色的 postfix 角色作为技术预览

Red Hat Enterprise Linux 系统角色为 Red Hat Enterprise Linux 子系统提供了一个配置界面，这有助于通过包含 Ansible 角色来简化系统配置。这个界面支持在多个 Red Hat Enterprise Linux 版本间管理系统配置，并使用新的主发行版本。

rhel-system-roles 软件包通过 AppStream 软件仓库发布。

postfix 角色是作为技术预览提供的。

以下角色被完全支持：

- **kdump**
- **network**
- **selinux**
- **timesync**

如需更多信息，请参阅有关 RHEL 系统角色的知识库文章。

(BZ#1812552)

5.3.9. 虚拟化

用于 KVM 虚拟机的 AMD SEV

作为技术预览，RHEL 8 为使用 KVM 管理程序的 AMD EPYC 主机引入了安全加密虚拟化(SEV)功能。如果在虚拟机(VM)上启用，SEV 会加密虚拟机内存，因此主机不能访问虚拟机上的数据。如果主机被恶意软件成功损坏，这可以提高虚拟机的安全性。

请注意，在单一主机上同时可以使用此功能的虚拟机数量是由主机硬件决定的。当前的 AMD EPYC 处理器支持使用 SEV 运行最多 15 个正在运行的虚拟机。

也请注意，对于将 SEV 配置为可以引导的虚拟机，还必须使用硬内存限制配置虚拟机。要做到这一点，请在虚拟机 XML 配置中添加以下内容：

```
<memtune>
  <hard_limit unit='KiB'>N</hard_limit>
</memtune>
```

建议 N 的值等于或大于客户机 RAM + 256 MiB。例如：如果为客户端分配 2 GiB RAM，则 N 应该为 2359296 或更高。

(BZ#1501618, BZ#1501607)

Intel vGPU

作为技术预览，现在可以将物理 Intel GPU 设备划分为多个虚拟设备（称为 **mediated devices**）。然后可将这些 mediated devices 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,只有所选 Intel GPU 与 vGPU 功能兼容。另外,为虚拟机分配物理 GPU 使得主机无法使用 GPU,并可能阻止主机的图形显示输出工作。

(BZ#1528684)

IBM POWER 9 上现在可用的嵌套虚拟化

作为技术预览，现在可以使用在 IBM POWER 9 系统上运行的 RHEL 8 主机机器上的嵌套虚拟化功能。嵌套虚拟化使得 KVM 虚拟机(VM)充当虚拟机监控程序，允许在虚拟机内运行虚拟机。

请注意，嵌套虚拟化在 AMD64 和 Intel 64 系统中也是一个技术预览。

另请注意，要使嵌套虚拟化在 IBM POWER 9、主机、客户机和嵌套客户机上工作，目前都需要运行以下操作系统之一：

- RHEL 8
- 用于 POWER 9 的 RHEL 7

(BZ#1505999, BZ#1518937)

KVM 虚拟化可用于 RHEL 8 Hyper-V 虚拟机

作为技术预览，现在可将嵌套的 KVM 虚拟化用于 Microsoft Hyper-V hypervisor。因此，您可以在运行在 Hyper-V 主机的 RHEL 8 虚拟机中创建虚拟机。

请注意目前，这个功能仅适用于 Intel 系统。另外，在一些情况下，Hyper-V 中不默认启用嵌套虚拟化。要启用它，请参阅以下文档：

https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/user-guide/nested-virtualization

(BZ#1519039)

5.3.10. 容器

podman-machine 命令不被支持

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:RHELDOCS-16861)

5.4. 过时的功能

这部分提供了在 Red Hat Enterprise Linux 8.0 中 *弃用* 的功能的概述。

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。

Red Hat Enterprise Linux 8 中已弃用功能的支持保持改变。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 7 中存在但已在 RHEL 8 中*删除*的功能的详情，请参考 采用 RHEL 8 的注意事项。

5.4.1. 安装程序和镜像创建

ignoredisk Kickstart 命令的 --interactive 选项已被弃用

在以后的 Red Hat Enterprise Linux 版本中使用 **--interactive** 选项会导致严重安装错误。建议您修改 Kickstart 文件删除该选项。

(BZ#1637872)

弃用了一些 Kickstart 命令和选项

使用 RHEL 8 Kickstart 文件中的以下命令和选项会在日志中显示警告信息。

- **auth** 或 **authconfig**
- **device**
- **deviceprobe**
- **dmraid**
- **install**
- **lilo**
- **lilocheck**
- **mouse**
- **multipath**
- **bootloader --upgrade**
- **ignoredisk --interactive**
- **partition --active**
- **reboot --kexec**

如果只列出具体选项，则基础命令及其它选项仍可用且没有弃用。

有关 Kickstart 中的详情和相关更改，请参阅*使用 RHEL 8 的参考事项*文档中的 Kickstart 更改部分。

(BZ#1642765)

5.4.2. 文件系统和存储

禁用了 NFSv3 over UDP

默认情况下，NFS 服务器不再默认在 User Datagram Protocol(UDP)套接字上打开或监听。这个变化只影响 NFS 版本 3，因为版本 4 需要传输控制协议(TCP)。

RHEL 8 不再支持通过 UDP 的 NFS。

(BZ#1592011)

elevator 内核命令行参数已弃用

在之前的 RHEL 版本中使用 **elevator** 内核命令行参数为所有设备设置磁盘调度程序。在 RHEL 8 中，该参数已弃用。

上游 Linux 内核删除了对 **elevator** 参数的支持，但出于兼容性的原因，在 RHEL 8 中仍提供此支持。

请注意，内核会根据设备类型选择默认磁盘调度程序。这通常是最佳设置。如果您需要不同的调度程序，红帽建议您使用 **udev** 规则或 Tuned 服务来配置它。匹配所选设备并只为那些设备切换调度程序。

如需更多信息，请参阅以下文章： 为什么 'elevator=' 参数在 RHEL8 中不再起作用。

(BZ#1665295)

VDO 软件包中的 VDO Ansible 模块

VDO Ansible 模块目前由 **vdo** RPM 软件包提供。在以后的发行版本中，VDO Ansible 模块将移到 Ansible RPM 软件包中。

(BZ#1669537)

5.4.3. 网络

在 RHEL 8 中已弃用网络脚本

网络脚本在 Red Hat Enterprise Linux 8 中已弃用，且不再默认提供。基本安装提供了 **ifup** 和 **ifdown** 脚本的新版本，它们通过 **nmcli** 工具调用 **NetworkManager** 服务。在 Red Hat Enterprise Linux 8 中，要运行 **ifup** 和 **ifdown** 脚本，NetworkManager 必须正在运行。

请注意， **/sbin/ifup-local** 、 **ifdown-pre-local** 和 **ifdown-local** 脚本中的自定义命令不会执行。

如果需要这些脚本，您仍可以使用以下命令在系统中安装已弃用的网络脚本：

```
~]# yum install network-scripts
```

ifup 和 **ifdown** 脚本链接到已安装的旧网络脚本。

调用旧的网络脚本会显示一个关于它们已过时的警告。

(BZ#1647725)

5.4.4. 内核

rdma_rxe Soft-RoCE 驱动程序已被弃用

软件直接内存通过融合以太网(Soft-RoCE)（也称为 RXE）是模拟远程直接内存访问(RDMA)的功能。在 RHEL 8 中，Soft-RoCE 功能作为一个不受支持的技术预览提供。但是，由于稳定性问题，此功能已被弃用，并将在 RHEL 9 中删除。

(BZ#1878207)

5.4.5. 安全性

在 RHEL 8 中弃用 DSA

数字签名算法(DSA)在 Red Hat Enterprise Linux 8 中被视为已弃用。依赖于 DSA 密钥的身份验证机制在默认配置中不起作用。请注意，即使使用系统范围的**LEGACY**加密策略级别中，**OpenSSH**客户端都不接受 DSA 主机密钥。

(BZ#1646541)

在 NSS中弃用了SSL2 Client Hello

传输层安全(**TLS**)协议版本 1.2 及更早版本允许以向后兼容安全套接字层(**SSL**)协议版本 2 的方式与 **客户端 Hello** 消息进行协商。网络安全服务(**NSS**)库中对这个功能的支持已被弃用，默认是禁用的。

需要这个功能支持的应用程序需要使用新的 **SSL_ENABLE_V2_compatible_HELLO** API 启用它。以后的 Red Hat Enterprise Linux 8 版本中可以完全删除对这个功能的支持。

(BZ#1645153)

TLS 1.0 和 TLS 1.1 已弃用

TLS 1.0 和 TLS 1.1 协议在 **DEFAULT** 系统范围的加密策略级别被禁用。如果需要使用启用的协议，如 Firefox 网页浏览器中的视频检查程序，把系统范围的加密策略切换到 **LEGACY** 级别：

```
# update-crypto-policies --set LEGACY
```

如需更多信息，请参阅红帽客户门户网站中的 RHEL 8 中的 强加密默认值和弱加密算法 知识库中的文章以及 **update-crypto-policies(8)**手册页。

(BZ#1660839)

5.4.6. 虚拟化

RHEL 8 不支持虚拟机快照

当前创建虚拟机(VM)快照的机制已经被弃用，因为它无法可靠工作。因此，建议在 RHEL 8 中使用虚拟机快照。

请注意，一个新的 VM 快照机制正在开发中，并将在以后的 RHEL 8 次要发行本中完全实现。

(BZ#1686057)

Cirrus VGA 虚拟 GPU 类型已弃用

随着 Red Hat Enterprise Linux 的主要更新，**Cirrus VGA** GPU 设备将在 KVM 虚拟机中不再被支持。因此,，红帽建议您使用 **stdvga**、**virtio-vga** 或者 **qxl** 设备而不是 Cirrus VGA。

(BZ#1651994)

virt-manager 已被弃用

虚拟机管理器（也称 **virt-manager**） 已弃用。RHEL 8 web 控制台（也称 **Cockpit** ）旨在在以后的版本中成为它替换。因此，建议您使用 web 控制台使用 GUI 管理虚拟化。但是，在 Red Hat Enterprise Linux 8.0 中，一些功能只能从 **virt-manager** 或命令行访问。

（JIRA:RHELPLAN-10304）

5.4.7. 已弃用的软件包

下列软件包已弃用，可能不会包括在 Red Hat Enterprise Linux 未来的主发行版本中：

- 389-ds-base-legacy-tools
- authd
- custodia
- hostname
- libidn
- net-tools
- network-scripts
- nss-pam-ldapd
- sendmail
- yp-tools
- ypbind
- ypserv

5.5. 已知问题

这部分论述了 Red Hat Enterprise Linux 8 中已知的问题。

5.5.1. Web 控制台

无法使用 session_recording shell 登录到 RHEL web 控制台

目前，对于启用了 tlog 记录的用户，RHEL web 控制台登录会失败。RHEL web 控制台要求用户的 shell 存在于 **/etc/shells** 目录中，以允许成功登录。但是，如果将 **tlog-rec-session** 添加到 **/etc/shells** 中，则记录的用户可以通过使用 "chsh" 工具将 shell 从 **tlog-rec-session** 改为 **/etc/shells** 中的另一个 shell 来禁用记录。因此，红帽不推荐将 **tlog-rec-session** 添加到 **/etc/shells** 中。

(BZ#1631905)

5.5.2. 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。

(BZ#1640697)

xorg-x11-drv-fbdev,**xorg-x11-drv-vesa** 和 **xorg-x11-drv-vmware** 视频驱动程序不会被默认安装

带有特定类型 NVIDIA 图形卡和带有特定 AMD 加速处理单元的工作站不会在 RHEL 8.0 服务器安装后显示图形登录窗口。

要临时解决这个问题，请在 workstation 计算机上执行 RHEL 8.0 **工作站** 安装。如果工作站上需要安装 RHEL 8.0 **服务器**，请在安装后手动安装 **base-x** 软件包组，方法是运行 **yum -y groupinstall base-x** 命令。

此外，依赖 EFI 图形支持（如 Hyper-V）的虚拟机也受到影响。如果您在 Hyper-V 上选择了 **Server with GUI** base 环境，则可能无法登录，因为重启时会显示一个黑色屏幕。要在 Hyper-v 中临时解决这个问题，请按照以下步骤启用多或单用户模式：

1. 重启虚拟机。
2. 在启动过程中，使用键盘上的上下箭头键选择所需的内核。
3. 按键盘上的 **e** 键编辑内核命令行。
4. 将 **systemd.unit=multi-user.target** 添加到 GRUB 中的内核命令行。
5. 按 **Ctrl-X** 启动虚拟机。
6. 登录后，运行 **yum -y groupinstall base-x** 命令。
7. 重新引导虚拟机以访问图形模式。

(BZ#1687489)

使用 **reboot --kexec** 命令安装失败

当使用包含 **reboot --kexec** 命令的 Kickstart 文件时，RHEL 8 安装会失败。要避免这个问题，请在 Kickstart 文件中使用 **reboot** 命令而不是 **reboot --kexec**。

(BZ#1672405)

将 **Binary DVD.iso** 文件的内容复制到分区会省略 **.treeinfo** 和 **.discinfo** 文件

在本地安装过程中，当将 RHEL 8 Binary DVD.iso 镜像文件的内容复制到分区时，**cp <path>/* <mounted 分区>/dir** 命令中的 * 无法复制 **.treeinfo** 和 **.discinfo** 文件。成功安装时需要这些文件。因此，BaseOS 和 AppStream 软件仓库不会被加载，在 **anaconda.log** 文件中与 debug 相关的日志消息是问题的唯一记录。

要临时解决这个问题，请将缺少的 **.treeinfo** 和 **.discinfo** 文件复制到分区。

(BZ#1692746)

Anaconda 安装包括最小资源设置要求的低限制

Anaconda 以最少的资源设置在系统中启动安装，并且不要提供有关成功执行安装所需的资源的先前消息警告。因此，安装可能会失败，输出错误不会为可能的调试和恢复提供清晰的信息。要临时解决这个问题，请确保系统具有安装所需的最少资源设置：2GB 内存在 PPC64(LE)和 1GB on x86_64 上。因此，应该可以成功执行安装。

(BZ#1696609)

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。

(BZ#1697896)

5.5.3. 内核

i40iw 模块不会在引导时自动载入

由于许多 i40e NIC 不支持 iWarp，并且 **i40iw** 模块没有全面支持 suspend/resume，因此此模块默认不会自动加载，以确保暂停/恢复正常工作。要临时解决这个问题，请手动编辑 **/lib/udev/rules.d/90-rdma-hw-modules.rules** 文件，以启用 **i40iw** 的自动负载。

另请注意，如果同一机器上安装了另一个 RDMA 设备，非i40e RDMA 设备会触发 **rdma** 服务，它会加载所有已启用的 RDMA 堆栈模块，包括 **i40iw** 模块。

(BZ#1623712)

当很多设备被连接时，系统有时会变得无响应

当 Red Hat Enterprise Linux 8 配置大量设备时，系统控制台中会出现大量控制台信息。例如，当存在大量逻辑单元号(LUN)时，每个 LUN 都有多个路径。除了内核正在执行的其他工作外，控制台消息的激增可能会导致内核监视功能强制出现内核 panic，因为内核似乎处于挂起状态。

因为扫描在引导周期早期发生，所以连接很多设备时系统会变得无响应。这通常在系统启动时发生。

如果在启动后在机器上在设备扫描事件中启用了 **kdump**，硬锁定会导致一个 **vmcore** 镜像的捕获。

要临时解决这个问题，增大 watchdog 锁定计时器。为此，请将 **watchdog_thresh=*N*** 选项添加到内核命令行。将 ***N*** 替换为秒数：

- 如果少于一千台设备，请使用 **30** 个设备。
- 如果您有超过一千台设备，请使用 **60** 个设备。

对于存储，设备数量是到所有 LUN 的路径数：通常，**/dev/sd*** 设备的数量。

应用临时解决方案后，系统在配置大量设备时不再响应。

(BZ#1598448)

KSM 有时会忽略 NUMA 内存策略

当内核共享内存(KSM)功能通过 **merge_across_nodes=1** 参数启用时，KSM 会忽略 mbind（）函数设置的内存策略，并且可能会将某些内存区域的页面合并到与策略不匹配的非一致性内存访问(NUMA)节点。

要临时解决这个问题，如果使用 NUMA 内存与 QEMU 绑定，请禁用 KSM 或将 **merge_across_nodes** 参数设置为 **0**。因此，为 KVM 虚拟机配置的 NUMA 内存策略可以正常工作。

(BZ#1153521)

qede 驱动程序挂起 NIC 并使其不可用

由于一个错误，41000 和 45000 QLogic 系列 NIC 的 **qede** 驱动程序可能会导致固件升级和调试数据收集操作失败，并使 NIC 不可用或处于挂起状态，直到主机重新引导（PCI 重置）使 NIC 再次正常运行。

在以下情况下都检测到了这个问题：

- 当使用 inbox 驱动程序升级 NIC 的固件时
- 在收集调试数据时，运行 **ethtool -d ethx** 命令
- 运行 **sosreport** 命令，因为它包含 **ethtool -d ethx**。

- 当 inbox 驱动程序启动自动调试数据收集，如 IO 超时、Mail Box 命令超时和硬件属性。

红帽的未来勘误将通过红帽漏洞公告(RHBA)发布来解决这个问题。要临时解决这个问题，请在 https://access.redhat.com/support 中创建一个问题单来请求受支持的修复程序，直到 RHBA 发布为止。

(BZ#1697310)

radix 树符号被添加到 kernel-abi-whitelists

在 Red Hat Enterprise Linux 8 中的 **kernel-abi-whitelists** 软件包中添加了以下 radix 树符号：

- **__radix_tree_insert**
- **__radix_tree_next_slot**
- **radix_tree_delete**
- **radix_tree_gang_lookup**
- **radix_tree_gang_lookup_tag**
- **radix_tree_next_chunk**
- **radix_tree_preload**
- **radix_tree_tag_set**

上面的符号不应存在，将从 RHEL8 白名单中删除。

(BZ#1695142)

Podman 无法检查 RHEL 8 中的容器

Checkpoint 和 Restore in Userspace(CRIU)软件包的版本在 Red Hat Enterprise Linux 8 中已经过时。因此，CRIU 不支持容器检查点和恢复功能，**podman** 实用程序无法检查点容器。在运行 **podman container checkpoint** 命令时，会显示以下出错信息：'checkpointing a container requires at least CRIU 31100'

(BZ#1689746)

如果在 dracut.conf中使用 add_dracutmodules+=earlykdump 选项，则 early-kdump 和标准的 kdump 会失败

目前，在为 **early-kdump** 安装的内核版本和为其生成的内核版本 **initramfs** 之间会发生不一致。因此，启用了 **early-kdump** 的引导会失败，**early-kdump** 会失败。另外，如果 **early-kdump** 检测到它包含在标准 **kdump** initramfs 镜像中，它会强制退出。因此，如果将 **early-kdump** 作为默认的 **dracut** 模块添加，则标准的 **kdump** 服务在尝试重建 **kdump** 时也会失败。因此，**early-kdump** 和标准 **kdump** 都会失败。要临时解决这个问题，请不要在 **dracut.conf** 文件中添加 **add_dracutmodules+=earlykdump** 或任何等同的配置。因此，**dracut** 默认不包含 **early-kdump**，这可以防止问题的发生。但是，如果需要 **early-kdump** 镜像，则必须手动创建它。

(BZ#1662911)

Debug 内核无法在 RHEL 8 的崩溃捕获环境中引导

由于 debug 内核的内存需求特性，会在使用 debug 内核并触发内核 panic 时出现问题。因此，调试内核无法作为捕获内核引导，而是生成一个堆栈追踪。要临时解决这个问题，相应地增大崩溃内核内存。因此，debug 内核可以在崩溃捕获环境中成功引导。

(BZ#1659609)

当使用 **fadump** 时，网络接口被重命名为 **kdump-<interface-name>**

当固件辅助转储(**fadump**)用来捕获 vmcore，并使用 SSH 或 NFS 协议将其保存到远程机器时，如果 **<interface-name>** 是通用的，则网络接口被重命名为 **kdump-<interface-name>**。这是因为初始 RAM 磁盘(**initrd**)中的 vmcore 捕获脚本在网络接口名称中添加 kdump- 前缀来保护持久性命名。同一 **initrd** 也用于常规引导，因此生产内核的接口名称也会更改。

(BZ#1745507)

5.5.4. 软件管理

在非 **root** 用户下运行 **yum list** 会导致 YUM 崩溃

当在 **libdnf** 软件包已更新后在非 root 用户下运行 **yum list** 命令时，YUM 可能会意外终止。如果您碰到这个 bug，请在 root 下运行 **yum list** 来解决此问题。因此，在非 root 用户下后续尝试运行 **yum list** 不再会导致 YUM 崩溃。

(BZ#1642458)

默认情况下，**YUM v4** 跳过不可用的软件仓库

YUM v4 默认为所有存储库的"skip_if_unavailable=True"设置。因此，如果所需的存储库不可用，则不会在安装、搜索或更新操作中考虑存储库中的软件包。因此，即使不存在存储库，某些 **yum** 命令和基于 yum 的脚本也会成功使用退出代码 0。

目前，除了更新 **libdnf** 软件包外，没有其他可用的临时解决方案。

(BZ#1679509)

5.5.5. 基础架构服务

nslookup 和 **host** 实用程序会忽略递归不可用的名称服务器的回复

如果配置了名称服务器，且递归不能用于名称服务器，则 **nslookup** 和 **host** 工具会忽略来自此类名称服务器的回复，除非它是最后配置的那个。如果是上次配置的名称服务器，即使没有 **递归可用** 标志，也会接受答案。但是，如果最后配置的域名服务器无法响应或者无法访问，名字解析会失败。

要临时解决这个问题：

- 确定配置的域名服务器总是使用 **recursion available** 进行回复。
- 允许为所有内部客户端进行 recursion。

若要对问题进行故障排除，您还可以使用 **dig** 实用程序检测递归是否可用。

(BZ#1599459)

5.5.6. Shell 和命令行工具

绑定了 **net-snmp** 软件包的 **Python** 不可用

Net-SNMP 工具套件不为 **Python 3** 提供绑定，这是 RHEL 8 中默认的 **Python** 实现。因此,RHEL 8 不提供 **python-net-snmp**、**python2-net-snmp** 或 **python3-net-snmp** 软件包。

(BZ#1584510)

调试模式中的 systemd 生成不必要的日志消息

调试模式中的 **systemd** 系统和服务管理器会生成不必要的日志消息，其开头如下：

```
"Failed to add rule for system call ..."
```

运行以下命令列出信息：

```
journalctl -b _PID=1
```

这些调试消息毫无危害，您可以放心忽略它们。

目前，还没有可用的临时解决方案。

(BZ#1658691)

带有 KEYBD 陷阱字节字符的 ksh

当启用 **KEYBD** 陷阱时，Korn Shell(KSH)无法正确处理多字节字符。因此，当用户输入日语字符时，**ksh** 显示一个不正确的字符串。要临时解决这个问题，注释掉以下行，在 **/etc/kshrc** 文件中禁用 **KEYBD** 陷阱：

```
trap keybd_trap KEYBD
```

如需了解更多详细信息，请参阅相关的 知识库解决方案。

(BZ#1503922)

5.5.7. 动态编程语言、网页和数据库服务器

数据库服务器无法并行安装

由于 RPM 软件包冲突，RHEL 8.0 中无法并行安装 **mariadb** 和 **mysql** 模块。

按照设计，无法并行安装同一模块的多个版本(stream)。例如，您只需要从 **postgresql** 模块中选择一个可用的流，可以是 **10**（默认）或 **9.6**。在 RHEL 6 和 RHEL 7 的 Red Hat Software Collections 中可以并行安装组件。在 RHEL 8 中，可在容器中使用不同版本的数据库服务器。

(BZ#1566048)

mod_cgid 日志记录中的问题

如果在线程多处理模块(MPM)下使用 **mod_cgid** Apache httpd 模块（这是 RHEL 8 中的默认情况），则会出现以下日志问题：

- CGI 脚本的 **stderr** 输出未使用标准时间戳信息作为前缀。
- 如果已配置，CGI 脚本的 **stderr** 输出将无法正确重定向到特定于 **VirtualHost** 的日志文件。

(BZ#1633224)

IO::Socket::SSL Perl 模块不支持 TLS 1.3

TLS 1.3 协议的新功能（如会话恢复或后握验证）是在 RHEL 8 **OpenSSL** 库中实现的，但未在 **Net::SSLeay** Perl 模块中实现，因此在 **IO::Socket::SSL** Perl 模块中不可用。因此，客户端证书身份验证可能会失败，重新建立会话可能比 TLS 1.2 协议慢。

要临时解决这个问题，请在创建 **IO::Socket::SSL** 对象时将 **SSL_version** 选项设置为 **!TLSv1_3** 值来禁用 TLS 1.3 的使用。

(BZ#1632600)

生成的 Scala 文档不可读取

使用 **scaladoc** 命令生成文档时，因为缺少 JavaScript 资源，生成的 HTML 页面将无法使用。

(BZ#1641744)

5.5.8. Desktop

QXL 无法在基于 Wayland 的虚拟机上工作

qxl 驱动程序无法在特定系统管理程序上提供内核模式设置功能。因此，使用 **qxl** 的虚拟机(VM)不支持基于 Wayland 协议的图形，基于 Wayland 的登录屏幕也不会启动。

要临时解决这个问题，请使用 ：

- 基于 QuarkXpress Element Library(QXL)图形的虚拟机上的 **Xorg** 显示服务器而不是 **Wayland 上的 GNOME Shell**。

或者

- 虚拟机的 **virtio** 驱动程序而不是 **qxl** 驱动程序。

(BZ#1641763)

运行 **systemctl isolate multi-user.target**时不会显示控制台提示符

当在 GNOME 桌面会话中运行 **systemctl isolate multi-user.target** 命令时，只会显示光标，而不是控制台提示符。要临时解决这个问题，请按 **Ctrl+Alt+F2** 键。因此，会出现控制台提示符。

其行为适用于 **Wayland 上的 GNOME Shell**和 **X.Org** 显示服务器。

(BZ#1678627)

5.5.9. 图形基础结构

X.Org 上运行的桌面在更改为低屏幕分辨率时挂起

将 GNOME 桌面与 **X.Org** 显示服务器搭配使用时，如果您尝试将屏幕分辨率更改为低值，桌面将变得无响应。要临时解决这个问题，请不要将屏幕分辨率设置为小于 800 x 600 像素的值。

(BZ#1655413)

radeon 无法正确重置硬件

radeon 内核驱动程序目前没有在 kexec 上下文中正确重置硬件。相反， **radeon** 无法工作，从而导致剩余的 **kdump** 服务失败。

要临时解决这个问题，请通过在 **/etc/kdump.conf** 文件中添加以下行来在 kdump 中将 **radeon** 列入黑名单：

```
dracut_args --omit-drivers "radeon"
force_rebuild 1
```

重启机器和 kdump。启动 kdump 后，**force_rebuild 1** 行可能会从配置文件中删除 。

请注意，在这种情况下，kdump 不会提供图形，但 kdump 可成功运行。

(BZ#1694705)

5.5.10. 硬件启用

使用 ARP 链接监控器时备份从 MII 状态不起作用

默认情况下，由 i40e 驱动程序管理的设备进行源修剪，这会丢弃具有源 Media Access Control(MAC)地址与其中一个接收过滤器匹配的数据包。因此，在通道绑定中使用地址解析协议(ARP)监控时，备份介质独立接口(MII)状态将无法正常工作。要临时解决这个问题，请使用以下命令禁用源修剪：

```
# ethtool --set-priv-flags <ethX> disable-source-pruning on
```

因此，备份从 MII 状态可以正常工作。

(BZ#1645433)

在某些情况下，HP NMI watchdog 不会生成崩溃转储

HP NMI watchdog 的 **hpwdt** 驱动程序有时无法声明由 HPE watchdog 计时器生成的不可屏蔽中断(NMI)，因为 NMI 被 **perfmon** 驱动程序使用。

因此，**hpwdt** 在某些情况下无法调用 panic 来生成崩溃转储。

(BZ#1602962)

5.5.11. 身份管理

KCM 凭证缓存不适用于单个凭证缓存中的大量凭证

Kerberos 凭据管理器(KCM)可以处理最多 64 kB 的 ccache 大小。如果它包含太多凭证，Kerberos 操作（如 **kinit** ）会失败，因为用于在 **sssd-kcm** 组件和底层数据库间传输数据的缓冲区存在硬编码限制。

要临时解决这个问题，请在 **/etc/sssd/sssd.conf** 文件的 **kcm** 部分添加 **ccache_storage = memory** 选项。这指示 **kcm** 响应器仅将凭证缓存存储在内存中，而不是永久存储。如果您这样做，重启系统或 **sssd-kcm** 会清除凭证缓存。

(BZ#1448094)

更改 /etc/nsswitch.conf 需要手动重启系统

对 **/etc/nsswitch.conf** 文件的任何更改（例如运行 **authselect select profile_id** 命令）都需要重启系统，以便所有相关进程使用更新版本的 **/etc/nsswitch.conf** 文件。如果无法重新启动系统，请重新启动将您的系统加入 Active Directory 的服务，即 **系统安全服务后台程序** (SSSD)或 **winbind**。

(BZ#1657295)

冲突的超时值阻止 SSSD 连接到服务器

与 System Security Services Daemon(SSSD)使用的故障转移操作相关的一些默认超时值相互冲突。因此，为 SSSD 与单个服务器进行通信的超时值会阻止 SSSD 在连接操作超时前尝试其他服务器。要临时解决这个问题，请设置 **ldap_opt_timeout** 超时参数的值，高于 **dns_resolver_timeout** 参数的值，并设置 **dns_resolver_timeout** 参数的值高于 **dns_resolver_op_timeout** 参数的值。

(BZ#1382750)

SSSD 只能查找 ID 覆盖中的唯一证书

当多个 ID 覆盖包含同一证书时，系统安全服务守护进程(SSSD)将无法解析与证书匹配的用户的查询。尝试查找这些用户不会返回任何用户。请注意，通过使用用户名或 UID 查找用户可以正常工作。

(BZ#1446101)

SSSD 无法正确处理具有相同优先级的多个证书匹配规则

如果给定证书与多个具有相同优先级的证书匹配规则匹配，系统安全服务守护进程(SSSD)只使用其中一个规则。作为临时解决方案，请使用单个证书匹配规则，该规则 LDAP 过滤器由与 | （或）运算符串联的单独规则的过滤器组成。有关证书匹配规则的示例，请参阅 sss-certamp(5)man page。

(BZ#1447945)

SSSD 为本地用户返回不正确的 LDAP 组成员资格

如果系统安全服务守护进程(SSSD)从本地文件为用户提供服务，则文件提供商不包含来自其他域的组成员资格。因此，如果本地用户是 LDAP 组的成员，**id local_user** 命令不会返回用户的 LDAP 组成员资格。要临时解决这个问题，可以恢复系统在 **/etc/nsswitch.conf** 文件中查找用户的组成员资格的数据库的顺序，将 **sss files** 替换为 **files sss**，或隐式禁用 **files** 域，方法是添加

```
enable_files_domain=False
```

到 **/etc/sssd/sssd.conf** 文件中的 **[sssd]** 部分。

因此，**id local_user** 会为本地用户返回正确的 LDAP 组成员资格。

(BZ#1652562)

如果 sudo 规则引用组名称，则 sudo 规则可能无法用于 id_provider=ad

在 **initgroups** 操作过程中，系统安全服务守护进程(SSSD)不会解析 Active Directory 组名称，因为使用缓存优化了 AD 和 SSSD 之间的通信。缓存条目仅包含安全标识符(SID)，在按名称或 ID 请求组之前，不包含组名称。因此，sudo 规则与 AD 组不匹配，除非在运行 sudo 之前完全解析了这些组。

要临时解决这个问题，您需要禁用优化：打开 **/etc/sssd/sssd.conf** 文件，并在 **[domain/example.com]** 部分中添加 **ldap_use_tokengroups = false** 参数。

(BZ#1659457)

RHEL 8 中已更改了 systemd-user 的默认 PAM 设置，这可能会影响 SSSD 行为

Red Hat Enterprise Linux 8 中更改了可插拔验证模块(PAM)堆栈。例如，systemd 用户会话现在使用 **systemd -user** PAM 服务启动 PAM 对话。此服务现在递归包含 **system-auth** PAM 服务，其中可能包括 **pam_sss.so** 接口。这意味着始终调用 SSSD 访问控制。

请注意为 RHEL 8 系统设计访问控制规则时的更改。例如，您可以将 **systemd-user** 服务添加到允许的服务列表中。

请注意，对于某些访问控制机制，如 IPA HBAC 或 AD GPOs，默认情况下 **systemd-user** 服务已添加到允许的服务列表中，您不需要进行任何操作。

(BZ#1669407)

IdM 服务器在 FIPS 中无法工作

由于 Tomcat 的 SSL 连接器实现不完整，装有证书服务器的身份管理(IdM)服务器在启用了 FIPS 模式的机器上无法正常工作。

(BZ#1673296)

使用 sss ID 映射插件时，Samba 拒绝访问

要将 Samba 用作加入 Active Directory(AD)域的 RHEL 主机上的文件服务器，必须运行 Samba Winbind 服务，即使 SSSD 用于管理 AD 中的用户和组。如果您使用 **realm join --client-software=sssd** 命令加入域，或者在此命令中未指定 **--client-software** 参数，则 **realm** 仅创建 **/etc/sssd/sssd.conf** 文件。当您使用此配置在域成员上运行 Samba 并添加使用 **sss** ID 映射后端到 **/etc/samba/smb.conf** 文件以共享目录的配置时，ID 映射后端的更改可能会导致错误。因此，Samba 在某些情况下拒绝访问文件，即使存在用户或组且 SSSD 已知的。

如果您计划从以前的 RHEL 版本升级，**/etc/sssd/sssd.conf** 文件中的 **ldap_id_mapping** 参数被设置为 **True**（这是默认值），则没有可用的临时解决方案。在这种情况下，不要将主机升级到 RHEL 8，直到问题解决为止。

在其他情况下可能解决这一问题：

- 对于新安装，请使用 **realm join --client-software=winbind** 命令加入域。这会将系统配置为在所有用户和组查找中使用 Winbind 而不是 SSSD。在这种情况下，Samba 根据您是否将 **--automatic-id-mapping** 选项设置为 **yes**（默认）或 **no** 来在 **/etc/samba/smb.conf** 中使用 **rid** 或 **ad** ID 映射插件。如果您计划在将来或其他系统中使用 SSSD，使用 **--automatic-id-mapping=no** 允许更轻松地迁移，但要求您在 AD 中为所有用户和组存储 POSIX UID 和 GID。

- 当从以前的 RHEL 版本升级时，如果 **/etc/sssd/sssd.conf** 文件中的 **ldap_id_mapping** 参数被设置为 **False**，系统使用 AD 中的 **uidNumber** 和 **gidNumber** 属性进行 ID 映射：

  1. 将 **/etc/samba/smb.conf** 文件中的 **idmap config <domain> : backend = sss** 条目改为 **idmap config <domain> : backend = ad**

  2. 使用 **systemctl status winbind** 命令重新启动 Winbind。

(BZ#1657665)

nuxwdog 服务在 HSM 环境中失败，且需要在非 HSM 环境中安装 keyutils 软件包

nuxwdog watchdog 服务已集成到证书系统中。因此，**nuxwdog** 不再作为单独的软件包提供。要使用 watchdog 服务，请安装 **pki-server** 软件包。

请注意，**nuxwdog** 服务有以下已知问题：

- 如果您使用硬件存储模块(HSM)，**nuxwdog** 服务将无法工作。对于这个问题，没有可用的临时解决方案。

- 在非 HSM 环境中，Red Hat Enterprise Linux 8.0 不自动安装 **keyutils** 软件包作为依赖项。要手动安装 软件包，请使用 **dnf install keyutils** 命令。

(BZ#1652269)

仅可在 IdM CLI 中添加 AD 用户的 ID 覆盖

目前，将 Active Directory(AD)用户的 ID 覆盖添加到 Identity Management(IdM)组中，以便在 IdM Web UI 中授予对管理角色的访问权限。要临时解决这个问题，请使用 IdM 命令行界面(CLI)。

请注意，如果在之前使用 **ipa** 工具执行了某些操作后，在 IdM 服务器上安装了 **ipa-idoverride-memberof-plugin** 软件包，红帽建议清理 **ipa** 工具的缓存，以强制它刷新其有关 IdM 服务器元数据的视图。

为此，请删除在其下执行 **ipa** 实用程序的用户 **~/.cache/ipa** 目录的内容。例如，对于 root：

```
# rm -r /root/.cache/ipa
```

(BZ#1651577)

在 IdM 中启用 AD 信任时，没有显示有关所需 DNS 记录的信息

当通过外部 DNS 管理启用对 Red Hat Enterprise Linux Identity Management(IdM)安装中的 Active Directory(AD)信任时，不会显示有关所需 DNS 记录的信息。只有添加所需的 DNS 记录后，林信任才会成功。要临时解决这个问题，请运行 'ipa dns-update-system-records --dry-run' 命令，以获取 IdM 所需的所有 DNS 记录列表。当 IdM 域的外部 DNS 定义所需的 DNS 记录时，有可能建立对 AD 的林信任。

(BZ#1665051)

对 **ldap_id_use_start_tls** 选项使用默认值时的潜在风险

当使用没有 TLS 的 **ldap://** 进行身份查找时，可能会对攻击向量构成风险。特别是中间人(MITM)攻击，例如，攻击者可以通过更改 LDAP 搜索中返回的对象的 UID 或 GID 来冒充用户。

目前，强制 TLS 的 SSSD 配置选项 **ldap_id_use_start_tls** 默认为 **false**。确保您的设置在可信环境中操作，并决定对 **id_provider = ldap** 使用未加密的通信是否是安全的。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/sssd/sssd.conf** 文件中将 **ldap_id_use_start_tls** 选项设置为 **true** 来强制使用 TLS。计划在以后的 RHEL 版本中更改的默认行为。

(JIRA:RHELPLAN-155168)

5.5.12. 编译器和开发工具

GCC confuse SystemTap 生成的复合功能

GCC 优化可以为其他函数的部分内嵌副本生成复合函数。SystemTap 和 GDB 等工具无法区分这些复合函数和实际功能。因此，SystemTap 可以在复合和真实功能入口点上放置探测，从而为单个实际函数调用注册多个探测命中。

要临时解决这个问题，SystemTap 脚本必须经过相应调整，如检测递归和抑制与内联部分功能相关的探测。例如，一个脚本

```
probe kernel.function("can_nice").call { }
```

可以尝试避免描述的问题，如下所示：

```
global in_can_nice%

probe kernel.function("can_nice").call {
  in_can_nice[tid()] ++;
  if (in_can_nice[tid()] > 1) { next }
  /* code for real probe handler */
}
```

```
probe kernel.function("can_nice").return {
  in_can_nice[tid()] --;
}
```

请注意，这个示例脚本没有考虑所有可能的情况，如错过的 kprobes 或 kretprobes 或真正的预期递归。

(BZ#1169184)

ltrace 工具不报告函数调用

由于改进了应用于所有 RHEL 组件的二进制强化，**ltrace** 工具无法再检测 RHEL 组件的二进制文件中的功能调用。因此，**ltrace** 输出为空，因为在用于此类二进制文件时，其不报告任何检测到的调用。目前还没有可用的临时解决方案。

作为备注，**ltrace** 可以正确报告自定义二进制文件中的调用，而无需相应的强化标记。

(BZ#1618748, BZ#1655368)

5.5.13. 文件系统和存储

无法使用 iscsiuio 软件包发现 iSCSI 目标

Red Hat Enterprise Linux 8 不允许并发访问 PCI 寄存器区域。因此，**无法设置主机 net 参数(err 29)**错误，到发现门户的连接会失败。要临时解决这个问题，请在内核命令行中设置 iSCSI 卸载的内核参数 **iomem=relaxed**。这专门涉及使用 **bnx2i** 驱动程序进行卸载。因此，到发现门户的连接现在是成功的，**iscsiuio** 软件包现在可以正常工作。

(BZ#1626629)

迁移到不同的终端平台后，VDO 卷会丢失重复数据删除建议

Virtual Data Optimizer(VDO)以您的平台原生的 endian 格式写入通用重复数据删除服务(UDS)索引标头。如果您将 VDO 卷移至使用其他 endian 的平台，VDO 会考虑 UDS 索引损坏，并使用新的空白索引覆盖它。

因此，覆盖前保存在 UDS 索引中的任何重复数据删除建议都会丢失。然后 VDO 无法根据在移动卷前存储的数据删除新写入的数据，从而减少空间。

(BZ#1696492)

XFS DAX 挂载选项与共享写时复制数据扩展不兼容

使用共享写时复制数据扩展功能格式化的 XFS 文件系统与 **-o dax** 挂载选项不兼容。因此，使用 **-o dax** 挂载这样的文件系统会失败。

要临时解决这个问题，使用 **reflink=0** metadata 选项格式化文件系统以禁用共享复制时写入数据扩展：

```
# mkfs.xfs -m reflink=0 block-device
```

因此，使用 **-o dax** 挂载文件系统会成功。

如需更多信息，请参阅 在 NVDIMM 上创建文件系统 DAX 命名空间。

(BZ#1620330)

某些 SCSI 驱动程序有时可能会使用过多的内存

某些 SCSI 驱动程序使用的内存比 RHEL 7 中的内存更大。在某些情况下，比如在光纤通道主机总线适配器(HBA)上创建 vPort，内存用量可能会过大，具体取决于系统配置。

内存用量增加是由块层中内存预分配造成的。多队列块设备调度(BLK-MQ)和多队列 SCSI 堆栈(SCSI-MQ)预分配 RHEL 8 中每个 I/O 请求的内存，从而提高了内存用量。

(BZ#1733278)

5.5.14. 网络

nftables 不支持多组 **IP** 设置类型

nftables 数据包过滤框架不支持设置具有串联和间隔的类型。因此，您无法使用带有 **nftables** 的多维 IP 设置类型，如 **hash:net,port**。

要临时解决这个问题，如果您需要多组 IP 设置类型，请将 **iptables** 框架与 **ipset** 工具一起使用。

(BZ#1593711)

iptables-extensions(8)man page 中的 **TRACE** 目标不参考 **the nf_tables** 变体

iptables-extensions(8) 手册页中 **TRACE** 目标的描述只指向 **compat** 变体，但 Red Hat Enterprise Linux(RHEL)8.0 使用 **nf_tables** 变体。RHEL 中的基于 **nftables**的 **iptables** 实用程序在内部使用 **meta nftrace** 表达式。因此，内核不会在内核日志中打印 **TRACE** 事件，而是将它们发送到用户空间。但是，man page 不引用 **xtables-monitor** 命令行实用程序来显示这些事件。

(BZ#1658734)

RHEL 8 在交换机在长时间不可用后显示 802.3ad 绑定的状态为"Churned"

目前，当您配置 802.3ad 网络绑定并且交换机长时间停机时，Red Hat Enterprise Linux 会正确地将绑定的状态显示为"Churned"，即使连接返回到工作状态后也是如此。但是，这是预期的行为，因为"Churned"状态旨在告知管理员发生了重大链接中断。要清除此状态，请重新启动网络绑定或重新引导主机。

(BZ#1708807)

ebtables 命令不支持 **broute** 表

Red Hat Enterprise Linux 8.0 中的 **nftables**-based **ebtables** 命令不支持 **broute** 表。因此，用户无法使用这个功能。

(BZ#1649790)

当禁用 **GRO** 时，**IPsec** 网络流量在 **IPsec** 卸载过程中失败

当在该设备中禁用通用接收 Offload（GRO）时，IPsec 卸载将不会正常工作。如果在一个网络接口中配置了 IPsec 卸载，且在该设备中禁用 GRO，IPsec 网络流量会失败。

要临时解决这个问题，在该设备中启用 GRO。

(BZ#1649647)

NetworkManager 现在默认使用 **内部 DHCP** 插件

NetworkManager 支持 **internal** 和 **dhclient** DHCP 插件。默认情况下, Red Hat Enterprise Linux(RHEL)7 中的 **NetworkManager** 使用 **dhclient**，RHEL 8 使用 **internal** 插件。在某些情况下，插件的行为不同。例如： **dhclient** 可以使用在 **/etc/dhcp/** 目录里指定的附加设置。

如果您从 RHEL 7 升级到 RHEL 8，且 **NetworkManager** 的行为不同，请在 **/etc/NetworkManager/NetworkManager.conf** 文件中的 **[main]** 部分添加以下设置以使用 **dhclient** 插件：

```
[main]
dhcp=dhclient
```

(BZ#1571655)

不能使用 gnome-control-center 更改 基于 IPsec 的 VPN 高级选项

当使用 **gnome-control-center** 应用程序配置 **基于 IPsec 的 VPN** 连接时，**高级** 对话框将仅显示配置，但不允许进行任何更改。因此，用户无法更改任何高级 IPsec 选项。要临时解决这个问题，请使用 **nm-connection-editor** 或 **nmcli** 工具来执行高级属性的配置。

(BZ#1697326)

/etc/hosts.allow 和 /etc/hosts.deny 文件包含不准确的信息

在 Red Hat Enterprise Linux(RHEL)8 中删除 tcp_wrappers 软件包，但不删除其文件 /etc/hosts.allow 和 /etc/hosts.deny。因此，这些文件包含过时的信息，不适用于 RHEL 8。

要临时解决这个问题，请使用防火墙规则过滤对服务的访问。要根据用户名和主机名进行过滤，请使用特定于应用程序的配置。

(BZ#1663556)

在网络流量过载下，IP 碎片整理无法持续

在 Red Hat Enterprise Linux 8 中，垃圾回收内核线程已被删除，IP 片段仅在超时后过期。因此，在 Denial of Service(DoS)下的 CPU 使用率低得多，且最大可持续发展片段丢弃率会受到为 IP 回配单元配置的内存量的限制。使用默认设置工作负载需要存在数据包丢弃的碎片流量时，数据包重新排序或多个并发碎片流可能会发生相关的性能回归。

在这种情况下，用户可以在 **/proc/sys/net/ipv4** 目录设置 **ipfrag_high_thresh** 变量来限制内存量和 **ipfrag_time** 变量，从而在内存中保持每秒的 IP 碎片。例如，

echo 419430400 > /proc/sys/net/ipv4/ipfrag_high_thresh echo 1 > /proc/sys/net/ipv4/ipfrag_time

以上命令适用于 IPv4 流量。对于 IPv6，相关可调项是：**/proc/sys/net/ipv6/** 目录中的 **ip6frag_high_thresh** 和 **ip6frag_time**。

请注意，任何依赖于高速碎片流量的工作负载都可能导致稳定性和性能问题，特别是在数据包丢弃方面，因此生产环境中强烈建议进行此类部署。

(BZ#1597671)

RHEL 8 中的网络接口名称更改

在 Red Hat Enterprise Linux 8 中，默认使用与 RHEL 7 相同的一致的网络设备命名方案。但是，有些内核驱动程序，如 **e1000e**、**nfp**、**qede**、**sfc**、**tg3** 和 **bnxt_en** 在 RHEL 8 的新安装中更改了它们的一致性名称。但是，名称会在从 RHEL 7 升级时保留。

(BZ#1701968)

5.5.15. 安全性

libselinux-python 只能通过其模块提供

libselinux-python 软件包只包含用于开发 SELinux 应用程序的 Python 2 绑定，它用于向后兼容。因此，通过 **dnf install libselinux-python** 命令，默认的 RHEL 8 软件仓库不再提供 **libselinux-python**。

要临时解决这个问题，请启用 **libselinux-python** 和 **python27** 模块，并使用以下命令安装 **libselinux-python** 软件包及其相依性软件包：

```
# dnf module enable libselinux-python
# dnf install libselinux-python
```

或者，使用它的安装配置集在一个命令中安装 **libselinux-python**:

```
# dnf module install libselinux-python:2.8/common
```

因此，您可以使用相关的模块安装 **libselinux-python**。

(BZ#1666328)

libssh 不遵循系统范围的加密策略

libssh 库不遵循系统范围的加密策略设置。因此，当管理员使用 **update-crypto-policies** 命令更改 crypto 策略级别时，支持的算法集合不会被改变。

要临时解决这个问题，需要使用 **libssh** 的每个应用程序单独设置一组公告的算法。因此，当系统被设置为 criACY 或 FUTURE 策略级别时，使用 **libssh** 的应用程序与 **OpenSSH** 相比的行为不一致。

(BZ#1646563)

某些 rsyslog 优先级字符串无法正常工作

对于允许对加密进行精细控制的 **imtcp** 的 **GnuTLS** 优先级字符串的支持并不完整。因此，以下优先级字符串无法在 **rsyslog** 中正常工作：

```
NONE:+VERS-ALL:-VERS-TLS1.3:+MAC-ALL:+DHE-RSA:+AES-256-GCM:+SIGN-RSA-SHA384:+COMP-ALL:+GROUP-ALL
```

要临时解决这个问题，请只使用正确的优先级字符串：

```
NONE:+VERS-ALL:-VERS-TLS1.3:+MAC-ALL:+ECDHE-RSA:+AES-128-CBC:+SIGN-RSA-SHA1:+COMP-ALL:+GROUP-ALL
```

因此，当前的配置必须仅限于可正常工作的字符串。

(BZ#1679512)

默认日志设置在性能上的负面影响

默认日志环境设置可能会消耗 4 GB 内存甚至更多，当 **systemd-journald** 使用 **rsyslog** 运行时，速率限制值的调整会很复杂。

如需更多信息，请参阅 RHEL 默认日志设置对性能的负面影响及环境方案。

（JIRA:RHELPLAN-10431）

OpenSCAP rpmverifypackage 无法正常工作

rpmverifypackage 探测调用 **chdir** 和 **chroot** 系统调用两次。因此，在使用自定义 Open Vulnerability 和评估语言(OVAL)内容的 **OpenSCAP** 扫描中使用探测时会出现错误。

要临时解决这个问题，请不要在您的内容中使用 **rpmverifypackage_test** OVAL 测试，或者仅使用未使用 **rpmverifypackage_test** 的 **scap-security-guide** 软件包中的内容。

(BZ#1646197)

SCAP Workbench 无法从定制的配置集生成基于结果的补救方法

当尝试使用 **SCAP Workbench** 工具从自定义配置文件生成基于结果的补救角色时，会发生以下错误：

```
Error generating remediation role .../remediation.sh: Exit code of oscap was 1: [output truncated]
```

要临时解决这个问题，请使用带有 **--tailoring-file** 选项的 **oscap** 命令。

(BZ#1640715)

Kickstart 在 RHEL 8 中使用 org_fedora_oscap 而不是 com_redhat_oscap

Kickstart 将 Open Security Content Automation Protocol(OSCAP)Anaconda 附加组件引用为 **org_fedora_oscap** 而不是 **com_redhat_oscap**，这可能会导致混乱。这样做可以保持与 Red Hat Enterprise Linux 7 的向后兼容性。

(BZ#1665082)

OpenSCAP rpmverifyfile 无法正常工作

OpenSCAP 扫描程序无法以脱机模式正确更改当前工作目录，并且不使用 **OpenSCAP rpmverifyfile** 探测中的正确参数调用 **fchdir** 功能。因此，如果在 SCAP 内容中使用 **rpmverifyfile_test**，则使用一个 **oscap-chroot** 命令扫描任意文件系统会失败。因此，在上述场景中，**oscap-chroot** 中止。

(BZ#1636431)

OpenSCAP 不提供虚拟机和容器的离线扫描

重构 **OpenSCAP** 代码库会导致某些 RPM 探测无法在离线模式下扫描虚拟机和容器文件系统。因此，以下工具已从 **openscap-utils** 软件包中删除： **oscap-vm** 和 **oscap-chroot**。另外，opens **cap-containers** 软件包已被完全删除。

(BZ#1618489)

不提供容器安全和合规性扫描的实用程序

在 Red Hat Enterprise Linux 7 中，**oscap-docker** 工具可用于根据原子技术扫描 Docker 容器。在 Red Hat Enterprise Linux 8 中，Docker 和 Atomic 相关的 **OpenSCAP** 命令不可用。因此，RHEL 8 当前无法使用, **oscap-docker** 或等同的工具用于容器的安全性和合规性扫描。

(BZ#1642373)

OpenSSL TLS 库不会检测到 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 **raw RSA** 或 **RSA-PSS** 签名，使用 **OpenSSL TLS** 库的服务器应用程序将无法使用 **RSA** 密钥（如果 **PKCS#11** 令牌持有）。因此，**TLS** 通信将失败。

要临时解决这个问题，将服务器或客户端配置为使用 **TLS-1.2** 版本作为可用最高 **TLS** 协议版本。

(BZ#1681178)

如果 Apache httpd 使用存储在 PKCS#11 设备和 RSA-PSS 证书中的 RSA 私钥，则无法启动

PKCS#11 标准不会区分 RSA 和 RSA-PSS 密钥对象，并为这两者使用 **CKK_RSA** 类型。但是，OpenSSL 将不同类型的用于 RSA 和 RSA-PSS 密钥。因此，**openssl-pkcs11** 引擎无法决定为 OpenSSL 提供 PKCS#11 RSA 密钥对象的类型。目前，引擎会将所有 PKCS#11 **CKK_RSA** 对象的密钥类型设置为 RSA 密钥。当 OpenSSL 将从证书获得的 RSA-PSS 公钥的类型与引擎提供的 RSA 私钥对象中包含的类型进行比较时，则得出不同类型的结果。因此，证书和私钥不匹配。**X509_check_private_key（）** OpenSSL 函数中执行的检查在此场景中返回一个错误。**httpd** Web 服务器在其启动过程中调用此功能，以检查提供的证书和密钥是否匹配。由于对包含 RSA-PSS 公钥和 PKCS#11 模块中存储的 RSA-PSS 公钥和 RSA 私钥的证书进行这个检查总会失败，所以 **httpd** 无法开始使用此配置。这个问题没有可用的临时解决方案。

(BZ#1664802)

如果 httpd 使用 ECDSA 私钥，且未存储在 PKCS#11 设备中的对应公钥，则 httpd 无法启动

与 RSA 密钥不同，ECDSA 私钥不一定包含公钥信息。在这种情况下，您无法从 ECDSA 私钥获取公钥。因此，PKCS#11 设备将公钥信息存储在单独的对象中，无论是公钥对象还是证书对象。OpenSSL 期望引擎提供的 **EVP_PKEY** 结构用于存放公钥信息的私钥。填写要提供给 OpenSSL 的 **EVP_PKEY** 结构时，**openssl-pkcs11** 软件包中的引擎会尝试仅从匹配的公钥对象获取公钥信息，并忽略当前的证书对象。

当 OpenSSL 从引擎请求 ECDSA 私钥时，提供的 **EVP_PKEY** 结构不包含公钥信息，如果 PKCS#11 设备中没有公钥，即使有匹配的证书可用，也是如此。因此，因为 Apache **httpd** web 服务器调用 **X509_check_private_key（）** 函数，因此在启动过程中需要公钥，**httpd** 无法在此场景中启动。要临时解决这个问题，使用 ECDSA 密钥将私钥和公钥保存在 PKCS#11 设备中。因此，当 ECDSA 密钥存储在 PKCS#11 设备中时，**httpd** 可正确启动。

(BZ#1664807)

OpenSSH 无法正确处理不匹配标签的密钥的 PKCS #11 URI

OpenSSH 套件可以通过标签识别密钥对。该标签可能会在保存在智能卡中的私钥和公钥上有所不同。因此，使用对象部分（密钥标签）指定 PKCS #11 URI 可以阻止 OpenSSH 在 PKCS #11 中查找适当的对象。

要临时解决这个问题，请指定 PKCS #11 URIs，但没有对象部分。因此，OpenSSH 可以在使用 PKCS #11 URI 引用的智能卡上使用密钥。

(BZ#1671262)

iptables-ebtables 输出不与 ebtables 100% 兼容

在 RHEL 8 中，**ebtables** 命令由 **iptables-ebtables** 软件包提供，该软件包包含一个基于 **nftables** 的工具的重新实现。此工具具有不同的代码库，其输出在诸多方面有偏差，这些代码可忽略或谨慎设计选择。

因此,在迁移用于解析 **ebtables** 输出脚本时，需要对脚本进行以下调整：

- 将 MAC 地址格式化改为固定长度。必要时，在独立的字节值的开始包含一个 0，用于维护每个字段值带有两个字符。
- 已更改 IPv6 前缀的格式以符合 RFC 4291。斜杠字符后的结尾部分不再包含 IPv6 地址格式的子网掩码，而是一个前缀长度。这个更改只适用于有效的（left-contiguous）掩码，其他更改则仍然以旧格式打印。

(BZ#1674536)

OpenSSH 中默认不支持 curve25519-sha256

OpenSSH 客户端和服务器的系统范围的加密策略配置中缺少 **curve25519-sha256** SSH 密钥交换算法，即使它符合默认的策略级别。因此，如果客户端或服务器使用 **curve25519-sha256** 且主机不支持这个算法，则连接可能会失败。

要临时解决这个问题，您可以通过为 OpenSSH 客户端和服务器修改 **/etc/crypto-policies/back-ends/** 目录中的 **openssh.config** 和 **opensshserver.config** 文件来手动覆盖系统范围加密策略的配置。请注意，这个配置在每次系统范围的加密策略更改中都会被覆盖。如需更多信息，请参阅 **update-crypto-policies(8)** man page。

(BZ#1678661)

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。

(BZ#1685470)

与 VMware 托管系统的 SSH 连接无法正常工作

OpenSSH 套件的当前版本在 SSH 数据包中引入了对默认 IP 服务质量(IPQoS)标记的更改，这是 VMware 虚拟化平台无法正确处理的。因此，无法与 VMware 上的系统建立 SSH 连接。

要临时解决这个问题，请在 **ssh_config** 文件中包含 **IPQoS=throughput**。因此，与 VMware 托管系统的 SSH 连接可以正常工作。

如需更多信息，请参阅 在 VMWare 工作站中运行的 RHEL 8 无法通过 SSH 连接到其他主机 知识库解决方案。

(BZ#1651763)

5.5.16. 订阅管理

没有打印成功设置和取消设置 服务级别的消息

当 **candlepin** 服务没有"syspurpose"功能时，订阅管理器会使用不同的代码路径来设置 **service-level** 参数。这个代码路径不会打印操作的结果。因此，当订阅管理器设置了服务级别时，不会显示任何信息。当 **service-level** 集有拼写错误或不可用时，这尤其成问题。

(BZ#1661414)

syspurpose addons 对 **subscription-manager attach --auto** 输出没有影响。

在 Red Hat Enterprise Linux 8 中，添加了 **syspurpose** 命令行工具的四个属性：**role**、**usage**、**service_level_agreement** 和 **addons**目前，只有 **role**、**usage** 和 **service_level_agreement** 会影响到运行 **subscription-manager attach --auto** 命令的输出。试图为 **addons** 参数设置值的用户不会观察到对自动附加的订阅有任何影响。

(BZ#1687900)

5.5.17. 虚拟化

ESXi 虚拟机使用 cloud-init 自定义并克隆的启动非常慢

目前，如果 **cloud-init** 服务用于修改在 VMware ESXi 管理程序上运行的虚拟机(VM)，以使用静态 IP，然后克隆虚拟机，则新的克隆虚拟机在某些情况下需要很长时间才能重新引导。这是因为 **cloud-init** 将虚拟机的静态 IP 重写为 DHCP，然后搜索可用的数据源。

要临时解决这个问题，您可以在虚拟机第一次引导后卸载 **cloud-init**。因此，后续重启不会减慢。

(BZ#1666961, BZ#1706482)

启用嵌套虚拟化阻止实时迁移

目前，嵌套虚拟化功能与实时迁移不兼容。因此，在 RHEL 8 主机上启用嵌套虚拟化会阻止从主机迁移任何虚拟机(VM)，并保存虚拟机状态快照到磁盘。

请注意，嵌套虚拟化目前在 RHEL 8 中作为技术预览提供，因此不受支持。此外，默认禁用嵌套虚拟化。如果要启用它，请使用 **kvm_intel.nested** 或 **kvm_amd.nested** 模块参数。

(BZ#1689216)

使用 cloud-init 在 Microsoft Azure 上置备虚拟机会失败

目前，无法使用 **cloud-init** 工具在 Microsoft Azure 平台上置备 RHEL 8 虚拟机(VM)。要临时解决这个问题，请使用以下方法之一：

- 使用 **WALinuxAgent** 软件包而不是 **cloud-init** 在 Microsoft Azure 上调配虚拟机。
- 在 **/etc/NetworkManager/NetworkManager.conf** 文件中的 **[main]** 部分添加以下设置：

```
[main]
dhcp=dhclient
```

(BZ#1641190)

第二代 RHEL 8 虚拟机有时无法在 Hyper-V Server 2016 主机上引导

当使用 RHEL 8 作为在 Microsoft Hyper-V Server 2016 主机上运行的虚拟机(VM)中的客户机操作系统时，虚拟机在某些情况下无法引导，并返回到 GRUB 引导菜单。另外，会在 Hyper-V 事件日志中记录以下错误：

```
The guest operating system reported that it failed with the following error code: 0x1E
```

这个错误是由 Hyper-V 主机上的 UEFI 固件错误造成的。要临时解决这个问题，使用 Hyper-V Server 2019 作为主机。

(BZ#1583445)

virsh iface-* 命令无法一致性地工作

因为配置的依赖关系，目前**virsh iface-*** 命令（如 **virsh iface-start** 和 **virsh iface-destroy** 会经常失败。因此，建议您不要使用 **virsh iface-*** 命令配置和管理主机网络连接。反之，使用 NetworkManager 程序及其相关管理程序。

(BZ#1664592)

Azure 的 Linux 虚拟机扩展有时无法正常工作

默认情况下，RHEL 8 不包含 **python2** 软件包。因此，在某些情况下，在 RHEL 8 虚拟机上为 Azure 运行 Linux 虚拟机扩展（也称为 **azure-linux-extensions** ）会失败。

要提高 **azure-linux-extensions** 按预期工作的可能性，请手动在 RHEL 8 虚拟机上安装 **python2** ：

```
# yum install python2
```

(BZ#1561132)

5.5.18. 支持性

redhat-support-tool 不从 **opencase**中自动收集 **sosreport**

redhat-support-tool 命令无法创建 **sosreport** 归档。要临时解决这个问题，请单独运行 **sosreport** 命令，然后输入 **redhat-support-tool addattachment -c** 命令以上传归档或在客户门户网站中使用 Web UI。因此，会创建一个问题单并上传 **sosreport**。

请注意，**findkerneldebugs**、 **btextract**、**analyze** 、**diagnose** 命令无法按预期工作，并将在以后的版本中修复。

(BZ#1688274)

第 6 章 容器的显著变化

一组容器镜像可用于 Red Hat Enterprise Linux(RHEL)8.0。主要变更包括：

- Docker 不包含在 RHEL 8.0 中。要与容器一起工作，请使用 **podman**、**buildah**、**skopeo** 和 **runc** 工具。
  有关这些工具以及 RHEL 8 中使用容器的详情，请参考 构建、运行和管理容器。

- **podman** 已作为完全支持的功能发布。
  podman 管理单节点上的 pod、容器镜像和容器。它在 **libpod** 库基础上构建，支持管理容器和容器组，称为 pod。

  了解如何使用 **podman**,查看 构建、运行和管理容器。

- 在 RHEL 8 GA 中，红帽通用基础镜像(UBI)现已推出。UBIs 替换了以前提供的一些镜像，如标准和最小 RHEL 基础镜像。
  与旧的红帽镜像不同，UBI 可以自由重新分发。这意味着它们可以在任何环境和任意位置共享。即使您不是红帽客户，您也可以使用它们。

  对于 UBI 文档，请参阅 构建、运行和管理容器。

- 在 RHEL 8 GA 中，还提供了提供 AppStream 组件的容器镜像，为此，容器镜像与 RHEL 7 中的 **Red Hat Software Collections** 一起分发。所有 RHEL 8 镜像均基于 **ubi8** 基础镜像。

- RHEL 8 完全支持 64 位 ARM 架构的容器镜像 ARM。

- **rhel-tools** 容器已在 RHEL 8 中删除。 **support-tools** 容器提供了 **sos** 和 **redhat-support-tool** 工具程序。系统管理员也可以使用此镜像来构建系统工具容器镜像。

- 在 RHEL 8 中，对无根容器的支持作为技术预览提供。
  无根容器是由普通系统用户在没有管理权限的情况下创建和管理的容器。

第 7 章 国际化

7.1. RED HAT ENTERPRISE LINUX 8 国际语言

Red Hat Enterprise Linux 8 支持多种语言的安装，并根据您的需要更改语言。

- 东亚语言 - 日语、韩语、简体中文和繁体中文。
- 欧洲语言 - 英语、德语、西班牙语、法语、意大利语、葡萄牙语和俄语。

下表列出了为各种主要语言提供的字体和输入法。

语言	默认字体（字体软件包）	输入法
English	dejavu-sans-fonts	
法语	dejavu-sans-fonts	
德语	dejavu-sans-fonts	
意大利语	dejavu-sans-fonts	
俄语	dejavu-sans-fonts	
西班牙语	dejavu-sans-fonts	
葡萄牙语	dejavu-sans-fonts	
简体中文	google-noto-sans-cjk-ttc-fonts, google-noto-serif-cjk-ttc-fonts	ibus-libpinyin, libpinyin
繁体中文	google-noto-sans-cjk-ttc-fonts, google-noto-serif-cjk-ttc-fonts	ibus-libzhuyin, libzhuyin
日语	google-noto-sans-cjk-ttc-fonts, google-noto-serif-cjk-ttc-fonts	ibus-kkc, libkc
韩语	google-noto-sans-cjk-ttc-fonts, google-noto-serif-cjk-ttc-fonts	ibus-hangul, libhangu

7.2. RHEL 8 中国际化的显著变化

RHEL 8 与 RHEL 7 相比，对国际化进行了以下更改：

- 添加了对 **Unicode 11** 计算行业标准的支持。
- 国际化发布在多个软件包中，这样就可以进行较小的内存占用安装。
  如需更多信息，请参阅 RHEL 的 **glibc** 本地化在多个软件包中分发。

- 现在，多个区域的 **glibc** 软件包更新与 Common Locale Data Repository(CLDR)同步。

附录 A. 按组件划分的问题单列表

组件	票证
389-ds-base	BZ#1334254, BZ#1358706
NetworkManager	BZ#1555013, BZ#1555012, BZ#1557035, BZ#1335409, BZ#1571655
PackageKit	BZ#1559414
WALinuxAgent	BZ#1561132
anaconda	BZ#1499442, BZ#1500792, BZ#1547908, BZ#1612060, BZ#1595415, BZ#1610806, BZ#1533904, BZ#1672405, JIRA:RHELPLAN-1943, BZ#1677411, BZ#1502323, BZ#1696609
audit	BZ#1616428
authselect	BZ#1657295
bcc	BZ#1548302
bind	BZ#1588592
boom-boot	BZ#1649582
boost	BZ#1494495, BZ#1616244
cloud-init	BZ#1615599, BZ#1641190
cmake	BZ#1590139
cockpit	BZ#1619993, BZ#1631905
criu	BZ#1689746
crypto-policies	BZ#1591620, BZ#1645606, BZ#1678661, BZ#1660839
cryptsetup	BZ#1564540
device-mapper-multipath	BZ#1643550, BZ#1673167
distribution	BZ#1516728, BZ#1516741, BZ#1566048
dnf	BZ#1622580, BZ#1647760, BZ#1581191
driverctl	BZ#1648411

组件	票证
edk2	BZ#1536627
esc	BZ#1538645
firewalld	BZ#1509026, BZ#1648497
gcc	BZ#1169184, BZ#1607227, BZ#1535774, BZ#1504980, BZ#1571124, BZ#1246444, JIRA:RHELPLAN-7437, BZ#1652016
gdb	BZ#1491128
gdm	BZ#1589678, BZ#1641763, BZ#1678627
glib-networking	BZ#1640534
glibc	BZ#1512004, BZ#1376834, BZ#1512010, BZ#1304448, BZ#1512009, BZ#1512006, BZ#1514839, BZ#1533608
gnome-control-center	BZ#1697326
go-toolset-1.10-golang	BZ#1633351
grub2	BZ#1583445
httpd	BZ#1633224, BZ#1632754
ipa-idoverride-memberof	BZ#1651577
ipa	BZ#1664718, BZ#1664719, BZ#1665051
iproute	BZ#1640991, BZ#1589317
iptables	BZ#1644030, BZ#1564596, BZ#1646159, BZ#1658734, BZ#1649790, BZ#1674536
iscsi-initiator-utils	BZ#1626629, BZ#1582099
kernel-rt	BZ#1592977

组件	票证
内核	BZ#1598448, BZ#1559607, BZ#1643522, BZ#1485546, BZ#1562998, BZ#1494651, BZ#1485532, BZ#1494028, BZ#1563617, BZ#1485525, BZ#1261167, BZ#1562987, BZ#1273139, BZ#1401552, BZ#1638465, BZ#1598776, BZ#1503672, BZ#1633143, BZ#1596240, BZ#1534870, BZ#1153521, BZ#1515987, BZ#1642795, BZ#1570255, BZ#1645744, BZ#1440031, BZ#1649647, BZ#1494705, BZ#1650149, BZ#1655413, BZ#1651806, BZ#1620330, BZ#1665295, BZ#1505999, BZ#1645433, BZ#1663281, BZ#1695142, BZ#1627455, BZ#1581898, BZ#1597671, BZ#1550498, BZ#1658391, BZ#1623590, BZ#1614144, BZ#1519039, BZ#1524683, BZ#1694705
kexec-tools	BZ#1520209, BZ#1662911
kmod-kvdo	BZ#1534087, BZ#1639512, BZ#1696492
ksh	BZ#1503922
libdnf	BZ#1642458, BZ#1679509
libreswan	BZ#1566574, BZ#1648776, BZ#1657854
libssh	BZ#1485241
libvirt	BZ#1528684
lksctp-tools	BZ#1568622
ltrace	BZ#1618748, BZ#1584322
lvm2	BZ#1676598, BZ#1643543, BZ#1643545, BZ#1643547, BZ#1643549, BZ#1643562, BZ#1643576
mariadb	BZ#1637034
mdadm	BZ#1654482
mutter	BZ#1668883
net-snmp	BZ#1584510
nfs-utils	BZ#1592011, BZ#1639432
nftables	BZ#1593711
nginx	BZ#1545526

组件	票证
nodejs-10-module	BZ#1622118
nss	BZ#1489094, BZ#1645153
nuxwdog	BZ#1652269
openldap	BZ#1570056
opensc	BZ#1595638, BZ#1595626
openscap	BZ#1614273, BZ#1618484, BZ#1646197, BZ#1636431, BZ#1618489, BZ#1642373, BZ#1618464
openssh	BZ#1622511, BZ#1228088, BZ#1645038, BZ#1671262, BZ#1651763
openssl-pkcs11	BZ#1664802, BZ#1664807
openssl	BZ#1685470
oscap-anaconda-addon	BZ#1665082
pacemaker	BZ#1543494
pcs	BZ#1578891, BZ#1591308, BZ#1615420, BZ#1158816, BZ#1542288, BZ#1549535, BZ#1620190, BZ#1566430, BZ#1595829, BZ#1436217, BZ#1578955, BZ#1596050, BZ#1554310, BZ#1638852, BZ#1640477, BZ#1619620
perl-IO-Socket-SSL	BZ#1632600
Perl	BZ#1511131
pki-core	BZ#1565073, BZ#1623444, BZ#1566360, BZ#1394069, BZ#1669257, BZ#1656856, BZ#1673296
postgresql-9.6-module	BZ#1660041
pykickstart	BZ#1637872, BZ#1612061
python-rtslib	BZ#1666377
qemu-kvm	BZ#1559240, BZ#1508139, BZ#1497911, BZ#1578855, BZ#1651994, BZ#1621817, BZ#1508137, BZ#1592337, BZ#1570029, BZ#1689216, BZ#1585651, BZ#1519004
redhat-release	BZ#1636338

组件	票证
redhat-support-tool	BZ#1688274
rsyslog	BZ#1613880, BZ#1542497, BZ#1614179, BZ#1619645, BZ#1679512, JIRA:RHELPLAN-10431
scala-2.10-module	BZ#1641744
scap-security-guide	BZ#1618505, BZ#1618528, BZ#1618518
scap-workbench	BZ#1640715
selinux-policy	BZ#1664345, BZ#1594111, BZ#1592244, BZ#1549772, BZ#1483904, BZ#1626446
setup	BZ#1591969, BZ#1663556
scs	BZ#1559836
squid	BZ#1656871
sssd	BZ#1448094, BZ#1382750, BZ#1446101, BZ#1447945, BZ#1620123, BZ#1652562, BZ#1659457, BZ#1669407, BZ#1657665
subscription-manager	BZ#1654531, BZ#1661414
subversion	BZ#1571415
swig-3.0-module	BZ#1660051
systemd	BZ#1658691
tomcatjss	BZ#1424966, BZ#1636564
tuned	BZ#1565598
valgrind	BZ#1500481, BZ#1538009
varnish	BZ#1633338
vdo	BZ#1669537
virt-manager	BZ#1599777, BZ#1643609
wpa_supplicant	BZ#1582538, BZ#1537143

组件	票证
xorg-x11-server	BZ#1687489, BZ#1698565
其他	JIRA:RHELPLAN-10347, BZ#1646563, JIRA:RHELPLAN-2306, BZ#1640697, BZ#1623712, BZ#1649404, BZ#1581198, BZ#1581990, BZ#1649497, BZ#1695584, BZ#1654280, BZ#1643294, BZ#1647612, BZ#1641015, BZ#1641032, BZ#1641004, BZ#1641034, BZ#1647110, BZ#1641007, BZ#1641029, BZ#1641022, JIRA:RHELPLAN-1212, BZ#1649493, BZ#1559616, BZ#1699825, BZ#1646541, BZ#1647725, BZ#1686057, BZ#1582530, BZ#1581496, BZ#1650618, BZ#1650675, BZ#1650701, JIRA:RHELPLAN-10439, JIRA:RHELPLAN-10440, JIRA:RHELPLAN-10442, JIRA:RHELPLAN-10443, JIRA:RHELPLAN-10438, JIRA:RHELPLAN-2878, JIRA:RHELPLAN-10355, JIRA:RHELPLAN-3010, JIRA:RHELPLAN-10352, JIRA:RHELPLAN-10353, JIRA:RHELPLAN-1473, JIRA:RHELPLAN-10445, JIRA:RHELPLAN-3001, JIRA:RHELPLAN-6746, JIRA:RHELPLAN-10354, JIRA:RHELPLAN-2896, JIRA:RHELPLAN-10304, JIRA:RHELPLAN-10628, JIRA:RHELPLAN-10441, JIRA:RHELPLAN-10444, JIRA:RHELPLAN-1842, JIRA:RHELPLAN-10596, JIRA:RHELPLAN-7291, JIRA:RHELPLAN-12764, BZ#1680177, JIRA:RHELPLAN-14607, JIRA:RHELPLAN-1820, BZ#1684947, BZ#1683712, BZ#1659609, BZ#1504934, BZ#1642765, BZ#1641014, BZ#1692746, BZ#1687900, BZ#1690207, BZ#1693775, BZ#1580387, BZ#1583620, BZ#1580430, BZ#1648843, BZ#1647908, BZ#1649891, BZ#1695698, BZ#1697896, BZ#1698613, BZ#1699535, BZ#1701968, BZ#1704867

致谢

感谢 RHEL 8 准备挑战过程中提供反馈的所有人。前 3 名获奖者是：

- sterling Alexander
- John Pittman
- Jake Hunsaker

附录 B. 修订历史记录

0.1-7

2024 年 5 月 9 日星期四，Brian Angelica (bangelic@redhat.com)

- 在 BZROX0207 中更新了技术预览。

0.1-6

2023 年 11 月 10 日星期五, Gabriela Fialová(gfialova@redhat.com)

- 更新了对 RHEL 文档提供反馈的模块。

0.1-5

2023 年 10 月 13 日星期五, Gabriela Fialová(gfialova@redhat.com)

- 添加了一个技术预览 JIRA:RHELDOCS-16861（容器）。

0.1-4

2023 年 4 月 27 日星期四, Gabriela Fialová(gfialova@redhat.com)

- 添加了一个已知问题 JIRA:RHELPLAN-155168 （身份管理）。

0.1-3

2022 年 4 月 29 日星期五, Lenka Špačková (lspackova@redhat.com)

- 更新了 已弃用功能 介绍。
- 修复了 BZ#1605216 中的拼写错误。
- 修复了损坏的链接。

0.1-2

2022 年 3 月 17 日星期四, Lucie Maňásková (lmanasko@redhat.com)

向新特性部分中添加了 JIRA:RHELPLAN-14323,JIRA:RHELPLAN-14329, JIRA:RHELPLAN-14330 (虚拟化)。

0.1-1

2021 年 12 月 23 日，Lenka paová(lspackova@redhat.com)

- 将 Soft-RoCE 驱动程序(**rdma_rxe**)的信息添加到了技术预览 BZ#1605216 和已弃用的功能 BZ#1878207 (内核)。

0.1-0

2021 年 9 月 23 日, Lucie Maňásková (lmanasko@redhat.com)

- 删除了无效的新功能描述（虚拟化）。

0.0-9

Thu Aug 19 2021, Lucie Maňásková (lmanasko@redhat.com)

- 向分发一章中添加了使用 YUM/DNF 的软件包管理。

0.0-8

Wed Jun 23 2021, Lucie Maňásková (lmanasko@redhat.com)

- 更新了新功能部分(Installer)。

0.0-7

Tue Apr 06 2021, Lenka Špačková (lspackova@redhat.com)

- 改进了支持的构架列表。

0.0-6

Thu Jan 28 2021, Lucie Maňásková (lmanasko@redhat.com)

- 更新了技术预览章节。

0.0-5

Thu Dec 10 2020, Lenka Špačková (lspackova@redhat.com)

- 向新功能（身份管理）中添加了有关在 SSSD 中处理 AD GPO 的信息。

0.0-4

Tue Apr 28 2020, Lenka Špačková (lspackova@redhat.com)

- 在概述中更新有关原位升级的信息。

0.0-3

Thu Mar 12 2020, Lenka Špačková (lspackova@redhat.com)

- 向技术预览中添加了缺少的 **postfix** RHEL 系统角色。

0.0-2

Wed Feb 12 2020, Jaroslav Klech (jklech@redhat.com)

- 为架构和新功能提供了完整的内核版本.

0.0-1

Tue Jul 30 2019, Lucie Maňásková (lmanasko@redhat.com)

- 发行 Red Hat Enterprise Linux 8.0.1 发行注记。

0.0-0

2019 年 5 月 7 日星期二，Ioanna Gkioka(igkioka@redhat.com)

- 发布 Red Hat Enterprise Linux 8.0 发行注记。