Red Hat Enterprise Linux 10

10.1 Release Notes

Release Notes for Red Hat Enterprise Linux 10.1

Last Updated: 2026-02-02

Red Hat Enterprise Linux 10 10.1 Release Notes

Release Notes for Red Hat Enterprise Linux 10.1

Legal Notice

Abstract

The Release Notes provide high-level coverage of the improvements and additions that have been implemented in Red Hat Enterprise Linux 10.1 and document known issues in this release, as well as notable fixed issues, Technology Previews, deprecated functionalities, functionalities removed in RHEL 10, and other details. For information about installing Red Hat Enterprise Linux, see Installation.

Table of Contents

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION 5

CHAPTER 1. OVERVIEW 6
1.1. MAJOR CHANGES IN RHEL 10.1 6
Installer and image creation 6
Security 6
Kernel 6
Dynamic programming languages, web and database servers 6
Compilers and development tools 7
System toolchain 7
Performance tools and debuggers 7
Performance monitoring tools 7
.NET 10.0 is now available on RHEL 7
Compiler toolsets 7
The web console 8
1.2. IN-PLACE UPGRADE 8
In-place upgrade from RHEL 9 to RHEL 10 8
In-place upgrade from RHEL 8 to RHEL 10 9
1.3. RED HAT CUSTOMER PORTAL LABS 9
1.4. ADDITIONAL RESOURCES 9

CHAPTER 2. ARCHITECTURES 11

CHAPTER 3. DISTRIBUTION OF CONTENT IN RHEL 10 12
3.1. INSTALLATION 12
3.2. REPOSITORIES 12
3.3. APPLICATION STREAMS 12

CHAPTER 4. IMPORTANT CHANGES TO EXTERNAL KERNEL PARAMETERS 14
New kernel parameters 14
Updated kernel parameters 15
Updated sysctl parameters 17

CHAPTER 5. DEVICE DRIVERS 18
5.1. NEW DRIVERS 18
5.2. UPDATED DRIVERS 21

CHAPTER 6. NEW FEATURES AND ENHANCEMENTS 22
6.1. INSTALLER AND IMAGE CREATION 22
6.2. SECURITY 24
6.3. SOFTWARE MANAGEMENT 30
6.4. SHELLS AND COMMAND-LINE TOOLS 31
6.5. INFRASTRUCTURE SERVICES 33
6.6. NETWORKING 35
6.7. KERNEL 39
6.8. BOOT LOADER 44
6.9. FILE SYSTEMS AND STORAGE 44
6.10. HIGH AVAILABILITY AND CLUSTERS 45
6.11. DYNAMIC PROGRAMMING LANGUAGES, WEB AND DATABASE SERVERS 49
6.12. COMPILERS AND DEVELOPMENT TOOLS 49
6.13. IDENTITY MANAGEMENT 54
6.14. SSSD 59
6.15. DESKTOP 60

6.16. THE WEB CONSOLE 60
6.17. RED HAT ENTERPRISE LINUX SYSTEM ROLES 60
6.18. VIRTUALIZATION 64
6.19. RHEL IN CLOUD ENVIRONMENTS 68
6.20. SUPPORTABILITY 68
6.21. CONTAINERS 69
6.22. RHEL LIGHTSPEED 71
6.23. AI ACCELERATOR DRIVER AVAILABILITY 72

CHAPTER 7. TECHNOLOGY PREVIEW FEATURES 74
7.1. INSTALLER AND IMAGE CREATION 74
7.2. SOFTWARE MANAGEMENT 74
7.3. SHELLS AND COMMAND-LINE TOOLS 74
7.4. KERNEL 74
7.5. FILE SYSTEMS AND STORAGE 75
7.6. DYNAMIC PROGRAMMING LANGUAGES, WEB AND DATABASE SERVERS 76
7.7. COMPILERS AND DEVELOPMENT TOOLS 76
7.8. IDENTITY MANAGEMENT 77
7.9. VIRTUALIZATION 78
7.10. CONTAINERS 80
7.11. TECHNOLOGY PREVIEW FEATURES IDENTIFIED IN PREVIOUS RELEASES 81
7.11.1. Networking 81

CHAPTER 8. DEVELOPER PREVIEW FEATURES 82
8.1. RHEL LIGHTSPEED 82

CHAPTER 9. REMOVED FEATURES 83
9.1. INSTALLER AND IMAGE CREATION 83
9.2. NETWORKING 83
9.3. COMPILERS AND DEVELOPMENT TOOLS 83
9.4. IDENTITY MANAGEMENT 83
9.5. RED HAT ENTERPRISE LINUX SYSTEM ROLES 84
9.6. VIRTUALIZATION 84

CHAPTER 10. DEPRECATED FEATURES 85
10.1. INSTALLER AND IMAGE CREATION 85
10.2. SECURITY 85
10.3. SOFTWARE MANAGEMENT 86
10.4. INFRASTRUCTURE SERVICES 86
10.5. NETWORKING 87
10.6. FILE SYSTEMS AND STORAGE 87
10.7. HIGH AVAILABILITY AND CLUSTERS 87
10.8. COMPILERS AND DEVELOPMENT TOOLS 88
10.9. THE WEB CONSOLE 88
10.10. RED HAT ENTERPRISE LINUX SYSTEM ROLES 89
10.11. VIRTUALIZATION 89
10.12. CONTAINERS 91
10.13. DEPRECATED FEATURES IDENTIFIED IN PREVIOUS RELEASES 92
10.13.1. SSSD 92
10.14. DEPRECATED PACKAGES 92

CHAPTER 11. KNOWN ISSUES 94
11.1. INSTALLER AND IMAGE CREATION 94
11.2. SECURITY 97

11.3. SHELLS AND COMMAND-LINE TOOLS 99
11.4. INFRASTRUCTURE SERVICES 99
11.5. FILE SYSTEMS AND STORAGE 100
11.6. HIGH AVAILABILITY AND CLUSTERS 100
11.7. COMPILERS AND DEVELOPMENT TOOLS 100
11.8. IDENTITY MANAGEMENT 101
11.9. SSSD 102
11.10. DESKTOP 102
11.11. GRAPHICS INFRASTRUCTURES 103
11.12. THE WEB CONSOLE 103
11.13. RED HAT ENTERPRISE LINUX SYSTEM ROLES 103
11.14. VIRTUALIZATION 104
11.15. RHEL IN CLOUD ENVIRONMENTS 108
11.16. CONTAINERS 109
11.17. RHEL LIGHTSPEED 110
11.18. KNOWN ISSUES IDENTIFIED IN PREVIOUS RELEASES 110
11.18.1. Networking 110
11.18.2. Virtualization 111
CHAPTER 12. FIXED ISSUES 112
12.1. INSTALLER AND IMAGE CREATION 112
12.2. SECURITY 112
12.3. SOFTWARE MANAGEMENT 113
12.4. SHELLS AND COMMAND-LINE TOOLS 113
12.5. INFRASTRUCTURE SERVICES 114
12.6. NETWORKING 114
12.7. KERNEL 116
12.8. BOOT LOADER 116
12.9. FILE SYSTEMS AND STORAGE 116
12.10. HIGH AVAILABILITY AND CLUSTERS 117
12.11. COMPILERS AND DEVELOPMENT TOOLS 118
12.12. IDENTITY MANAGEMENT 120
12.13. SSSD 123
12.14. RED HAT ENTERPRISE LINUX SYSTEM ROLES 124
12.15. VIRTUALIZATION 128
12.16. RHEL IN CLOUD ENVIRONMENTS 130
12.17. SUPPORTABILITY 130
12.18. CONTAINERS 132
12.19. RHEL LIGHTSPEED 133
CHAPTER 13. AVAILABLE BPF FEATURES 135
APPENDIX A. LIST OF TICKETS BY COMPONENT 154
APPENDIX B. REVISION HISTORY 163

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our documentation. Let us know how we can improve it.

Submitting feedback through Jira (account required)

1. Log in to the Jira website.
2. Click **Create** in the top navigation bar.
3. Enter a descriptive title in the **Summary** field.
4. Enter your suggestion for improvement in the **Description** field. Include links to the relevant parts of the documentation.
5. Click **Create** at the bottom of the dialogue.

CHAPTER 1. OVERVIEW

1.1. MAJOR CHANGES IN RHEL 10.1

Installer and image creation

Key highlights for RHEL installer:

- The newly created users will have administrative privileges by default, unless you deselect the option.
- You can now set the required time zone by using new options instead of the time zone map.
- The remote desktop protocol (RDP) for graphical remote access replaces VNC.

Key highlights for RHEL image builder:

- RHEL image builder has a new CLI experience available as a Technology Preview
- RHEL image builder **cockpit-composer** package is removed and replaced with the new **cockpit-image-builder** plugin.
- System images created with the RHEL image builder, such as AWS or KVM formats, do not have a separate /**boot** partition.
- RHEL image builder now supports WSL2 images.

For more information, see New features and enhancements - Installer and image creation .

Security

The system-wide **cryptographic policies** enable post-quantum cryptography (PQC) algorithms in all policies by default.

OpenSSL 3.5 introduces support for the ML-KEM, ML-DSA, and SLH-DSA **post-quantum algorithms** and adds the hybrid ML-KEM algorithms to the default TLS group list.

RHEL 10.1 introduces also support for **RPMv6** signatures. This new format enables multiple signatures in an RPM package. You can use **Sequoia PGP** tools to sign RPM packages with PQC algorithms and create or verify OpenPGPv6 signatures.

See New features - Security for more information.

Kernel

RHEL 10.1 extends kernel-focused observability and energy tracking. Enhancements include upstream alignment for **perf** and BPF, broader **uncore** and **core** counters, Intel RAPL energy events, Intel Trace Hub (NPK) device IDs, AMD per-core energy tracking, and modernized debugging with **python-drgn** alongside crash tooling updates. Entropy generation improves through **rng-tools**, delivering more consistent performance insights across current hardware generations.

Dynamic programming languages, web and database servers

Later versions of the following Application Streams are now available:

- **Node.js 24**

Later versions of the following web servers are now available:

- **Apache HTTP Server 2.4.63**

See New features - Dynamic programming languages, web and database servers for more information.

Compilers and development tools

System toolchain

The following system toolchain components are available with RHEL 10.1:

- GCC 14.3
- glibc 2.39
- Annobin 12.99
- binutils 2.41

Performance tools and debuggers

The following performance tools and debuggers are available with RHEL 10.1:

- GDB 16.3
- Valgrind 3.25.1
- SystemTap 5.3
- Dyninst 13.0.0
- elfutils 0.193
- libabigail 2.8

Performance monitoring tools

The following performance monitoring tools are available with RHEL 10.1:

- PCP 6.3.7
- Grafana 10.2.6

.NET 10.0 is now available on RHEL

Red Hat Enterprise Linux (RHEL) supports .NET, a general-purpose development platform that features automatic memory management and modern programming languages, allowing you to build high-quality applications efficiently. This update adds support for the most recent version, .NET 10.0 (Long-Term Support), expanding the versions available on RHEL. Other supported versions include .NET 9.0 (Standard-Term Support) and the previous long-term support version, .NET 8.0.

For more information, see Release Notes for .NET 10.0 RPM packages and Release Notes for .NET 10.0 containers

Compiler toolsets

The following compiler toolsets are available with RHEL 10.1:

- GCC Toolset 15
    - GCC 15.1
    - Binutils 2.44
      Note that **Annobin** and **dwz** are not provided in GCC Toolset starting with version 15.
- LLVM Toolset 20.1.8

- Rust Toolset 1.88.0
- Go Toolset 1.24

For detailed changes, see New features - Compilers and development tools .

The web console

The **cockpit** packages have been upgraded to version 344, which provides many improvements, most notably the upgrade to the **Patternfly 6 system design**.

See New features - The web console for more information.

1.2. IN-PLACE UPGRADE

In-place upgrade from RHEL 9 to RHEL 10

The supported in-place upgrade paths currently are:

- From RHEL 9.6 to RHEL 10.0 and RHEL 9.7 to RHEL 10.1 on the following architectures:
  - AMD and Intel 64-bit architectures (x86-64-v3)
  - The 64-bit ARM architecture (ARMv8.0-A)

    IMPORTANT

    For the 64-bit ARM architecture, in-place upgrades are supported only on systems that run the **4k** page size kernel. The Leapp utility does not support in-place upgrades if the system is booted with the **64k** page size kernel.

  - IBM Power Systems, Little Endian (POWER9)
  - 64-bit IBM Z (IBM z15 or IBM LinuxONE III or later)

For more information, see Supported in-place upgrade paths for Red Hat Enterprise Linux .

For instructions on performing an in-place upgrade, see Upgrading from RHEL 9 to RHEL 10 .

Notable enhancements and bug fixes include:

- Fix in-place upgrades on systems that use the **fapolicyd** software framework.
- Disable the **localpkg_gpgcheck** DNF option when performing the upgrade allowing the required installation of bundled **leapp-deps-el10** and **leapp-repository-deps-el10** metapackages.
- Introduce the LiveMode feature as a Technology Preview. LiveMode allows you to upgrade by using the standard booting process. You can also use LiveMode for troubleshooting and testing. For more information, see Configuring the upgrade with LiveMode.
- Inhibit the upgrade on systems that use deprecated **network-legacy** dracut module to prevent kernel panic.
- Migrate SSSD configuration during the in-place upgrade.

- Enable upgrades on PAYG RHEL systems that use Red Hat Upgrade Infrastructure (RHUI) on Amazon Web Services (AWS), Azure, and Google Cloud.

In-place upgrade from RHEL 8 to RHEL 10

It is not possible to perform an in-place upgrade directly from RHEL 8 to RHEL 10. However, you can perform an in-place upgrade from RHEL 8 to RHEL 9 and then perform a second in-place upgrade to RHEL 10. For more information, see In-place upgrades over multiple RHEL major versions by using Leapp.

1.3. RED HAT CUSTOMER PORTAL LABS

Red Hat Customer Portal Labs is a set of tools in a section of the Customer Portal available at https://access.redhat.com/labs/. The applications in Red Hat Customer Portal Labs can help you improve performance, quickly troubleshoot issues, identify security problems, and quickly deploy and configure complex applications. Some of the most popular applications are:

- Registration Assistant
- Kickstart Generator
- Red Hat Product Certificates
- Red Hat CVE Checker
- Kernel Oops Analyzer
- Red Hat Satellite Upgrade Helper
- JVM Options Configuration Tool
- Load Balancer Configuration Tool
- Ceph Placement Groups (PGs) per Pool Calculator
- Yum Repository Configuration Helper
- Red Hat Out of Memory Analyzer
- Postfix Configuration Helper
- System Unit Generator
- Rsyslog Configuration Helper
- Red Hat IdM Upgrade Helper

1.4. ADDITIONAL RESOURCES

Red Hat Insights is now **Red Hat Lightspeed**. This is a change in name only and all the same product features, functionalities, and capabilities you have relied on under the Red Hat Insights name remain under the name Red Hat Lightspeed. With Red Hat Lightspeed, which is included with all RHEL subscriptions, you can proactively identify, examine, and resolve known technical issues. For instructions on how to install the client and register your system to the service, see the Red Hat Lightspeed documentation page.

NOTE

Public release notes include links to access the original tracking tickets, but private release notes are not viewable so do not include links.[1]

[1] Public release notes include links to access the original tracking tickets, but private release notes are not viewable so do not include links.

CHAPTER 2. ARCHITECTURES

Red Hat Enterprise Linux 10.1 is distributed with the kernel version 6.12.0-124.8.1, which provides support for the following architectures at the minimum required version (stated in parentheses):

- AMD and Intel 64-bit architectures (x86-64-v3)
- The 64-bit ARM architecture (ARMv8.0-A)
- IBM Power Systems, Little Endian (POWER10 and later)
- 64-bit IBM Z (IBM z15 or IBM LinuxONE III)

Make sure you purchase the appropriate subscription for each architecture.

CHAPTER 3. DISTRIBUTION OF CONTENT IN RHEL 10

3.1. INSTALLATION

Red Hat Enterprise Linux 10 is installed using ISO images. Two types of ISO image are available for the AMD64, Intel 64-bit, 64-bit ARM, IBM Power Systems, and IBM Z architectures:

- Installation ISO: A full installation image that contains the BaseOS and AppStream repositories and allows you to complete the installation without additional repositories. On the Product Downloads page, the **Installation ISO** is referred to as **Binary DVD**.

NOTE

The Installation ISO image is in multiple GB size, and as a result, it might not fit on optical media formats. A USB key or USB hard drive is recommended when using the Installation ISO image to create bootable installation media. You can also use the Image Builder tool to create customized RHEL images. For more information about Image Builder, see the *Composing a customized RHEL system image* document.

- Boot ISO: A minimal boot ISO image that is used to boot into the installation program. This option requires access to the BaseOS and AppStream repositories to install software packages. The repositories are part of the Installation ISO image. You can also register to Red Hat CDN or Satellite during the installation to use the latest BaseOS and AppStream content from Red Hat CDN or Satellite.

3.2. REPOSITORIES

Red Hat Enterprise Linux 10 is distributed through two main repositories:

- BaseOS
- AppStream

Both repositories are required for a basic RHEL installation, and are available with all RHEL subscriptions.

Content in the BaseOS repository is intended to provide the core set of the underlying operating system functionality that provides the foundation for all installations. This content is available in the RPM format and is subject to support terms similar to those in previous releases of RHEL.

Content in the AppStream repository includes additional user-space applications, runtime languages, and databases in support of the varied workloads and use cases.

In addition, the CodeReady Linux Builder repository is available with all RHEL subscriptions. It provides additional packages for use by developers. Packages included in the CodeReady Linux Builder repository are unsupported.

For more information about RHEL 10 repositories and the packages they provide, see the Package manifest.

3.3. APPLICATION STREAMS

Multiple versions of user-space components are delivered as Application Streams and updated more frequently than the core operating system packages. This provides greater flexibility to customize RHEL without impacting the underlying stability of the platform or specific deployments.

Application Streams are available in the following formats:

- RPM format
- Software Collections
- Flatpaks

NOTE

In previous RHEL major versions, some Application Streams were available as modules as an extension to the RPM format. In RHEL 10, Red Hat does not intend to provide any Application Streams that use modularity as the packaging technology and, therefore, no modular content is being distributed with RHEL 10.

Each Application Stream component has a given life cycle, either the same as RHEL 10 or shorter.

RHEL 10 improves the Application Streams experience by providing initial Application Stream versions that can be installed as RPM packages using the **dnf install** command.

NOTE

Certain initial Application Streams in the RPM format have a shorter life cycle than Red Hat Enterprise Linux 10.

Always determine what version of an Application Stream you want to install.

Content that needs rapid updating, such as alternate compilers and container tools, is available in Rolling Streams that will not provide alternative versions in parallel.

CHAPTER 4. IMPORTANT CHANGES TO EXTERNAL KERNEL PARAMETERS

This chapter provides system administrators with a summary of significant changes in the kernel distributed with Red Hat Enterprise Linux 10.1. These changes could include, for example, added or updated **proc entries**, **sysctl**, and **sysfs** default values, boot parameters, kernel configuration options, or any noticeable behavior changes.

New kernel parameters

indirect_target_selection=

[X86, Intel] Mitigation control for the Indirect Target Selection (ITS) issue on Intel CPUs. Updated microcode is required for the IBPB fix.

Values:

on (default)

Enable mitigation.

off

Disable mitigation.

force

Force ITS bug state and deploy the default mitigation.

vmexit

Deploy mitigation only for guest/host isolation.

stuff

Use RSB-fill when retpoline is deployed; otherwise use the default mitigation.

See: Documentation/admin-guide/hw-vuln/indirect-target-selection.rst.

sdw_mclk_divider=

[SDW, Intel] Specify the MCLK divider for Intel SoundWire buses when the BIOS does not provide the clock rate properly.

thp_shmem=

[KNL, MM] Control the default huge page policy per size for the internal shmem mount.

Format:

<size>[KMG]<size>[KMG]:<policy>;<size>[KMG]-<size>[KMG]:<policy>

Policies: **always**, **inherit**, **never**, **within_size**, **advise**.

You can specify this multiple times to set policies for multiple THP sizes. See: Documentation/admin-guide/mm/transhuge.rst.

transparent_hugepage_shmem=

[KNL, MM] Control the huge page allocation policy for the internal shmem mount.

Values: **always**, **within_size**, **advise**, **never**, **deny**, **force**. See: Documentation/admin-guide/mm/transhuge.rst.

tsa=

[X86, AMD] Control mitigation for Transient Scheduler Attacks on AMD CPUs.

Values:

off
: Disable the mitigation.

on (default)
: Enable the mitigation.

user
: Mitigate only user/kernel transitions.

vm
: Mitigate only guest/host transitions.

For guidance, see vendor documentation for transient scheduler attacks.

Updated kernel parameters

init=

[KNL] Format: **<full_path>** Run the specified binary instead of **/sbin/init** as the system init process.

intremap=

[X86-64, Intel-IOMMU, EARLY]

Values:

on
: Enable Interrupt Remapping (default).

off
: Disable Interrupt Remapping.

kvm-arm.mode=

[KVM, ARM, EARLY] Select the Arm KVM virtualization mode.

Values:

nvhe
: Standard nVHE-based mode without protected guests.

protected
: Support guests with state private from the host, using VHE or nVHE depending on hardware support. Disables kexec and hibernation on the host. To force nVHE on VHE hardware, add **arm64_sw.hvhe=0 id_aa64mmfr1.vh=0** to the command line.

nested
: VHE-based mode with nested virtualization. Requires Armv8.4 hardware (FEAT_NV2). Experimental; use with extreme caution. Defaults to VHE or nVHE based on hardware support.

kvm-arm.vgic_v3_group0_trap=

[KVM, ARM, EARLY] Trap guest accesses to GICv3 group-0 registers to the host for stricter isolation and debugging.

libata.force=

[SATA/ATA] Per-port options:

New: **external**

Mark the port as external (hotplug-capable). Other options remain available (for example, **max_sec_lba48**, **[no]lpm**, **[no]setxfer**).

nohz

[KNL] Disable the tick when a single task runs, and offload other kernel work such as RCU callbacks. Equivalent to **nohz_full**. A residual 1 Hz tick is offloaded to workqueues; affine these to housekeeping CPUs via the global workqueue CPU mask. See also **rcu_nocbs=** and **isolcpus=**.

pci=

[PCI] ACS configuration example updated to include a device selector: **pci=config_acs=10x@pci:0:0**:: Configure supported devices to enable P2P Request Redirect, disable Translation Blocking, and leave Source Validation unchanged along the specified device path.

pcie=

[PCIE] New system-wide flag: **notph**:: Disable PCIe TLP Processing Hints support when **CONFIG_PCIE_TPH** is enabled.

pcie_aspm=

[PCIE] Forcibly enable or ignore PCIe Active State Power Management. Behavior unchanged; use alongside **pcie=** flags where needed.

preempt=

[KNL, Scheduler] Preemption control modes. New: **lazy**:: Scheduler-controlled mode similar to **full**. The task gets one HZ tick to yield before the scheduler forces preemption. A preemption is counted when the task returns to user space.

print-fatal-signals=

[KNL] Enable debugging output for fatal signals.

skew_tick=

[KNL, EARLY] Offset the periodic timer tick per CPU to reduce contention in large systems and with **CONFIG_MAXSMP**.

slub_debug=

[MM] SLUB allocator debugging; **slub_nomerge** remains a legacy alias. See **Documentation/mm/slub.rst**.

spectre_v2=

[X86] Selecting a specific mitigation does not force-enable user-space mitigations. Selecting **on** enables kernel protections and mitigates user-to-user task attacks. Selecting **off** disables both.

tsc=

[X86] Disable clocksource stability checks for TSC.

Values:

reliable
: Mark the TSC clocksource as reliable.

noirqtime
: Do not use TSC for IRQ accounting.

unstable
: Mark the TSC clocksource as unstable.

nowatchdog
: Disable the clocksource watchdog.

recalibrate
: Recalibrate against HPET or PM timer if TSC frequency came from MSR or CPUID(0x15); warn if the difference is more than 500 ppm.

watchdog
: Use TSC as the watchdog clocksource on systems where TSC is trustworthy.

Note: An earlier **tsc=nowatchdog** suppresses **watchdog**. A later **tsc=nowatchdog** overrides it; the kernel logs any suppression or override.

transparent_hugepage_shmem=

[KNL, MM] Values: **always**, **within_size**, **advise**, **never**, **deny**, **force**. Controls the internal shmem mount policy. See **Documentation/admin-guide/mm/transhuge.rst**.

mitigations=

[Multi-arch] Selecting **off** disables a set of kernel and user-space mitigations. The equivalence list now includes **indirect_target_selection=off** on X86, in addition to existing entries such as **kpti=0** on Arm64, **gather_data_sampling=off**, **kvm.nx_huge_pages=off**, **l1tf=off**, **mds=off**, and related X86 flags.

Updated sysctl parameters

timer_migration

When set to a nonzero value, the kernel attempts to migrate timers away from idle CPUs to help those CPUs remain in low-power states longer.

Default: **1** (enabled).

CHAPTER 5. DEVICE DRIVERS

5.1. NEW DRIVERS

Table 5.1. Accelerator drivers

Description	Name	Limited to architectures
Driver for Intel NPU (Neural Processing Unit) - 1.0.0	intel_vpu	AMD and Intel 64-bit architectures

Table 5.2. Bluetooth drivers

Description	Name	Limited to architectures
Bluetooth support for MediaTek devices ver 0.1	btmtk	AMD and Intel 64-bit architectures, 64-bit ARM architecture

Table 5.3. Character device drivers

Description	Name	Limited to architectures
SNP SVSM vTPM (virtual Trusted Platform Module) driver	tpm_svsm	AMD and Intel 64-bit architectures

Table 5.4. DMA drivers

Description	Name	Limited to architectures
AMD AE4DMA driver	ae4dma	AMD and Intel 64-bit architectures
AMD PassThru DMA driver	ptdma	AMD and Intel 64-bit architectures

Table 5.5. Firmware control drivers

Description	Name	Limited to architectures
Firmware control access framework	fwctl	AMD and Intel 64-bit architectures, 64-bit ARM architecture
mlx5 ConnectX firmware control driver	mlx5_fwctl	AMD and Intel 64-bit architectures, 64-bit ARM architecture

Table 5.6. Graphics drivers and miscellaneous drivers

Description	Name	Limited to architectures
Chrontel ch7006 TV encoder driver	ch7006	AMD and Intel 64-bit architectures
Cirrus driver for QEMU emulated device	cirrus-qemu	AMD and Intel 64-bit architectures
DRM GPUSVM	drm_gpusvm	AMD and Intel 64-bit architectures, 64-bit ARM architecture
Quirks for panel backlight overrides	drm_panel_backlight_quirks	AMD and Intel 64-bit architectures
Silicon Image sil164 TMDS transmitter driver	sil164	AMD and Intel 64-bit architectures

Table 5.7. HID drivers

Description	Name	Limited to architectures
HID driver for Corsair Void headsets	hid-corsair-void	AMD and Intel 64-bit architectures
Intel Touch Host Controller driver	intel-thc	AMD and Intel 64-bit architectures
Intel QuickI2C driver	intel-quicki2c	AMD and Intel 64-bit architectures
Intel QuickSPI driver	intel-quickspi	AMD and Intel 64-bit architectures

Table 5.8. Media drivers

Description	Name	Limited to architectures
Conexant cx231xx USB video device driver - 0.0.3	cx231xx	AMD and Intel 64-bit architectures
Conexant CX25840 audio/video decoder driver	cx25840	AMD and Intel 64-bit architectures
cx23415/6/8 driver	cx2341x	AMD and Intel 64-bit architectures

Description	Name	Limited to architectures
Driver for various TV and TV+FM radio tuners	tuner	AMD and Intel 64-bit architectures
Videobuf2 DMA scatter/gather memory handling	videobuf2-dma-sg	AMD and Intel 64-bit architectures
I2C Hauppauge EEPROM decoder driver	tveeprom	AMD and Intel 64-bit architectures

Table 5.9. Network drivers

Description	Name	Limited to architectures
Aeonsemi AS21xxx PHY driver	as21xxx	AMD and Intel 64-bit architectures
Intel MLD wireless driver for Linux	iwlmld	AMD and Intel 64-bit architectures, 64-bit ARM architecture
MaxLinear MXL86110 PHY driver	mxl-86110	AMD and Intel 64-bit architectures
Microchip PHY RDS PTP driver	microchip_rds_ptp	AMD and Intel 64-bit architectures
Realtek PHY driver	realtek	AMD and Intel 64-bit architectures
Socket CAN driver for Geschwister Schneider and candleLight USB CAN interfaces	gs_usb	AMD and Intel 64-bit architectures, IBM Power Systems (ppc64le)

Table 5.10. Platform drivers

Description	Name	Limited to architectures
AMD 3D V-Cache Performance Optimizer driver	amd_3d_vcache	AMD and Intel 64-bit architectures

Table 5.11. USB drivers

Description	Name	Limited to architectures

Description	Name	Limited to architectures
Thunderbolt 3 USB Type-C Alternate Mode	typec_thunderbolt	AMD and Intel 64-bit architectures

Table 5.12. vDPA drivers

Description	Name	Limited to architectures
vDPA Device in Userspace	vduse	AMD and Intel 64-bit architectures, 64-bit ARM architecture

5.2. UPDATED DRIVERS

Table 5.13. Storage driver updates

Description	Name	Current version	Limited to architectures
Broadcom MegaRAID SAS driver	megaraid_sas	07.734.00.00-rc1	AMD and Intel 64-bit architectures, 64-bit ARM architecture, IBM Power Systems (ppc64le)
Cisco FCoE HBA driver	fnic	1.8.0.2	AMD and Intel 64-bit architectures
Driver for Microchip Smart Family Controller	smartpqi	2.1.34-035	AMD and Intel 64-bit architectures, 64-bit ARM architecture, IBM Power Systems (ppc64le)
Emulex LightPulse Fibre Channel SCSI driver	lpfc	0:14.4.0.9	AMD and Intel 64-bit architectures, 64-bit ARM architecture, IBM Power Systems (ppc64le)
LSI MPT Fusion SAS 3.0 device driver	mpt3sas	52.100.00.00	AMD and Intel 64-bit architectures, 64-bit ARM architecture, IBM Power Systems (ppc64le)
MPI3 Storage Controller device driver	mpi3mr	8.15.0.5.50	AMD and Intel 64-bit architectures

CHAPTER 6. NEW FEATURES AND ENHANCEMENTS

This version adds the following major new features and enhancements.

6.1. INSTALLER AND IMAGE CREATION

New boot menu entry for **fips=1** added to ISO installations

With this update, the DVD and Boot ISO image installations provide a new boot menu entry for setting the **fips=1** kernel boot option. This simplifies the process, as enabling FIPS mode during the RHEL installation ensures that the system generates all keys with FIPS-approved algorithms and continuous monitoring tests in place. By using this boot option, you start the installation with the **fips=1** kernel parameter and you can target the system’s compliance with Federal Information Processing Standards (FIPS) 140 requirements.

Jira:RHEL-91929

Soft reboots are now available in RHEL

Systemd now offers soft reboots, a capability for **rebooting** userspace without requiring full system downtime. Key enhancements include:

- Reduced downtime: Perform a soft reboot to update system state without the time-consuming process of a full reboot, which benefits scheduled maintenance and troubleshooting.
- Flexible patching: Apply certain userspace updates, such as **openssl**, **glibc**, and **dbus-broker**, without requiring a full system reboot.
- Image mode integration: In image mode, soft reboots either restart userspace when no update is staged or seamlessly switch to a staged update if one is present, excluding kernel changes.
- Improved immutability experience: Soft reboots simplify the adoption of new image versions on immutable systems by reducing the need for frequent full reboots.

Known limitations:

- Kernel modules: Changes to kernel modules might result in mismatches with the running kernel after a soft reboot.
- Kernel and firmware updates: Soft reboots do not apply kernel, kpatch, or firmware initialization changes.

Jira:RHELDOCS-20453[1]

The **rpm** command is now available in the installation environment

Previously, the **rpm** command was not included in the installation environment. With this update, the **rpm** command is now included. Users can use this command when installing RHEL, for example, in the **%post** Kickstart scripts.

Jira:RHEL-101695[1]

The blueprint file customization now supports a **URI** field for referencing files from external sources

This update adds the **URI** field support to the blueprint file customization structure. As a result, you can reference and source files from external locations rather than only those included directly in the blueprint, providing more flexible customization of the build system and a more adaptable build experience.

Jira:RHELDOCS-21016[1]

RHEL image builder supports a new image type **vagrant-libvirt** for vagrant

With this update, RHEL image builder supports the **libvirt** hypervisor, and you can easily run RHEL virtual machines by using Vagrant. This enhancement provides pre-configured images to ensure a consistent and streamlined setup. It also grants sudo privileges to the **vagrant** user within the Vagrant box, making it easier to manage and run administrative tasks. These enhancements deliver a more efficient and seamless experience when working with RHEL virtual machines in Vagrant environments.

Jira:RHELDOCS-21025[1]

RHEL Image Builder now supports WSL2 images

You can now use the RHEL image builder to create Windows Subsystem for Linux (WSL2). The image type is available in the **wsl** format, and to consume the image, deploy it by double-clicking the generated file.

Jira:RHELDOCS-20633[1]

RHEL Image Builder GUI supports modularized content discovery

Starting from RHEL 9.7, RHEL Image Builder Graphical User Interface (GUI) supports modularized content discovery. This capability introduces the following enhancements:

- When creating RHEL operating system images, you can use the RHEL Image Builder GUI to discover and include modularized content from various repositories, including RHEL AppStream and third-party repositories, for example, Extra Packages for Enterprise Linux (EPEL).
- Enhanced modularity support in RHEL. Application Streams use DNF modularity and **modulemd** metadata to provide flexible package management. You can specify version streams and use case profiles in the modules with support for default streams and profiles.
- DNF modularity implementation updates. The **@** character syntax for specifying RPM groups enables and installs module streams, providing compatibility for Kickstart files.

Jira:RHELDOCS-21026[1]

image-installer provides a new boot menu entry for **fips=1**

In this update, the **image-installer** ISO image type provides a new boot menu entry for setting the **fips=1** kernel boot option during installation. This simplifies the process, as in RHEL 10, you cannot switch an installed system to FIPS mode, and you must add **fips=1** to the kernel command line when starting the installation. By setting **fips=1** for the installation, you can target the system's compliance with Federal Information Processing Standards (FIPS) 140 requirements.

Jira:RHEL-104075

Logical volume devices in **/etc/fstab** now use UUID in the **fs_spec** field

After installation, the system writes logical volume (LV) devices in **/etc/fstab** by using UUID in the **fs_spec** field. This change provides the following benefits:

- Ensures consistency across all device entries in **/etc/fstab**.
- Supports LV or volume group (VG) renaming without changes in **/etc/fstab**.
- Keeps **/etc/fstab** valid after re-encrypting devices with LUKS.
- Preserves correct mapping of the root (/) and other mounts across re-provisioning, even if device-mapper paths change.
- Offers predictable and portable configurations as UUIDs are globally unique identifiers stored in the file system superblock.

Jira:RHEL-87651[1]

6.2. SECURITY

RHEL 10.1 crypto-policies enable PQC algorithms by default

The system-wide cryptographic policies in RHEL 10.1 extend support for post-quantum cryptography (PQC) and enable PQC algorithms by default in all predefined policies. The most notable enhancements and fixes over the version in RHEL 10.0 include:

- Hybrid Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) and pure Module-Lattice-Based Digital Signature Standard (ML-DSA) post-quantum cryptographic algorithms are enabled in LEGACY, DEFAULT, and FUTURE cryptographic policies with the highest priorities.
- The new NO-PQ subpolicy simplifies turning off the PQC algorithms.
- The TEST-PQ subpolicy no longer enables PQC algorithms as a Technology Preview, but you can use it to enable pure ML-KEM in OpenSSL.
- The FIPS cryptographic policy enables hybrid ML-KEM and pure ML-DSA post-quantum cryptographic algorithms.
- The new OpenSSL group selection syntax prioritizes post-quantum groups over classical ones. The behavior of earlier releases can be achieved only by disabling all PQ groups.
- The PQC algorithms are enabled for the Sequoia PGP tool in all policies.
- ML-DSA algorithms are enabled for GnuTLS TLS connections by default, and you can control them through the **MLDSA44**, **MLDSA65**, and **MLDSA87** values.
- The ML-DSA-44, ML-DSA-65, and ML-DSA-87 PQC algorithms are enabled for NSS TLS connections in all cryptographic policies.
- The **mlkem768x25519**, **secp256r1mlkem768**, and **secp384r1mlkem1024** hybrid ML-KEM groups are enabled for NSS TLS negotiations.

Jira:RHEL-113008, Jira:RHEL-97763, Jira:RHEL-103962, Jira:RHEL-86059, Jira:RHEL-101123, Jira:RHEL-85078, Jira:RHEL-98732, Jira:RHEL-106868, Jira:RHEL-92148

AD-SUPPORT-LEGACY subpolicy re-added to crypto-policies

The AD-SUPPORT-LEGACY cryptographic subpolicy, which is used to support legacy RC4 encryption for interoperability with outdated Active Directory implementations, is re-added to RHEL.

Jira:RHEL-93323[1]

OpenSSL rebased to 3.5

OpenSSL is rebased to upstream version 3.5. This version provides important fixes and enhancements, most notably the following:

- Added support for the ML-KEM, ML-DSA, and SLH-DSA post-quantum algorithms.
- Added the hybrid ML-KEM algorithms to the default TLS group list.
- Enhanced TLS configuration options.
- Added support for the QUIC transport protocol according to the IETF RFC 9000 draft.
- Added support for opaque symmetric key objects in the form of the EVP_SKEY data structure.
- Disabled the SHA-224 digest.
- SHAKE-128 and SHAKE-256 implementations no longer have a default digest length. Therefore, these algorithms cannot be used with the **EVP_DigestFinal**/**_ex()** function unless the **xoflen** parameter is set.
- Added a capability for a client to send multiple key shares in TLS 1.3 connections.

Jira:RHEL-80811

NSS rebased to 3.112

The NSS cryptographic toolkit packages have been rebased to upstream version 3.112, which provides many improvements and fixes. Most notably, the following:

- Added support for the Module-Lattice-Based Digital Signature Algorithm (ML-DSA), which is a post-quantum cryptography (PQC) standard.
- Added hybrid support for SSL for the MLKEM1024 key encapsulation mechanism.

The following known issues occur in this version:

- Updating the NSS database password corrupts the ML-DSA seed. For more information, see RHEL-114443.

Jira:RHEL-103352

libreswan rebased to 5.3

The **libreswan** packages are rebased to the 5.3 upstream version.

Jira:RHEL-102733[1]

GnuTLS rebased to 3.8.10

The **gnutls** package is rebased to the 3.8.10 upstream release, which includes the following enhancements:

- You can set TLS certificate compression methods with the **cert-compression-alg** configuration option in the **gnutls** priority file.

- You can use all variants of ML-DSA private key formats defined in the **draft-ietf-lamps-dilithium-certificates-12** document.
- You can use the ML-DSA-44, ML-DSA-65, and ML-DSA-87 signature algorithms in TLS.
- You can use PKCS#11 modules to override the default cryptographic backend as a Technology Preview. You can test this feature by specifying the **[provider]** section in the system-wide configuration to set the path and pin to the module.

Jira:RHEL-102557[1]

Sequoia PGP updated to support OpenPGP v6

With this update, the **sequoia-sq** and **sequoia-sqv** can handle post-quantum cryptography (PQC) keys. The **rpm-sequoia** package newly supports verifications of OpenPGP v6 signatures. As a result, you can use quantum-resistant digital signatures conforming to the Commercial National Security Algorithm Suite (CNSA) 2.0 standard.

Jira:RHEL-101952, Jira:RHEL-101906, Jira:RHEL-101905, Jira:RHEL-92148

selinux-policy rebased to 42.1

The **selinux-policy** packages are rebased to upstream version 42.1. This version contains many fixes and improvements, including packaging improvements. Notably, SELinux types related to the **systemd** generators have been added to the SELinux policy.

Jira:RHEL-54303

OpenSSL supports sslkeylogfile

OpenSSL supports the **sslkeylogfile** format for TLS. As a result, you can log all secrets produced by SSL connections by setting the **SSLKEYLOGFILE** environment variable.

IMPORTANT

Enabling the **SSLKEYLOGFILE** variable poses an explicit security risk. Recording the exchanged keys during an SSL session allows anyone with read access to the file to decrypt application traffic sent over that session. Use this feature only in test and debug environments.

Jira:RHEL-90853

NSS supports ML-DSA keys

With this update, the Network Security Services (NSS) database now supports using Module-Lattice-Based Digital Signature Algorithm (ML-DSA) keys. ML-DSA is a new signing algorithm approved by the National Institute of Standards and Technology (NIST) as resistant to attacks from a Cryptographically Relevant Quantum Computer (CRQC).

Jira:RHEL-64738

Hybrid ML-KEM cryptography works in FIPS mode

With this release, Hybrid Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) post-quantum cryptographic algorithms are supported in FIPS mode of RHEL. OpenSSL is able to fetch the Elliptic Curve Diffie-Hellman (ECDH) part of the new hybrid post-quantum groups from the FIPS provider when the system is running in FIPS mode. As a result, the OpenSSL library uses FIPS-compliant cryptography for the ECDH part of the hybrid post-quantum key exchanges.

Jira:RHEL-94614

OpenSSL 3.5 uses standard format for ML-KEM and ML-DSA

In RHEL 10.0, the **oqsprovider** library used a pre-standard format for the Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) and the Module-Lattice-Based Digital Signature Algorithm (ML-DSA) private keys. With the rebase to OpenSSL 3.5, you must convert the ML-KEM and ML-DSA keys to the standard format by using the following command:

```
# openssl pkcs8 -in <old_private_key> -nocrypt -topk8 -out <standard_private_key>
```

Replace **<old_private_key>** with the path to the non-standard private key, and **<standard_private_key>** with the path where the standard key will be saved.

Jira:RHEL-82676

SCAP Security Guide rebased to 0.1.78

For additional information, see the SCAP Security Guide release notes .

Jira:RHEL-111008

SELinux policy modules related to EPEL packages moved to **-extra** subpackages in the CRB repository

In RHEL 10.0, SELinux policy modules related only to packages contained in the Extra Packages for Enterprise Linux (EPEL) repository and not to any RHEL package were moved from the **selinux-policy** package to the **selinux-policy-epel** package. This reduced the size of **selinux-policy**, enabling the system to perform operations such as rebuilding and loading the SELinux policy faster.

In RHEL 10.1, the modules from **selinux-policy-epel** are moved to the following **-extra** subpackages in the RHEL CodeReady Linux Builder (CRB) repository:

- **selinux-policy-targeted-extra**
- **selinux-policy-mls-extra**

This change enables the automatic installation of **-extra** SELinux policy modules when users enable the EPEL repository.

Jira:RHEL-89587

setroubleshoot-server no longer requires initscripts

Before this update, the **%post** and **%postun** scriptlets for the **setroubleshoot-server** SELinux diagnostic tool called **/sbin/service**. With this update, the scriptlets now directly call **auditctl** for reloading the **auditd** service, and bypass the use of **/sbin/service**. This enhancement simplifies the dependency structure and streamlines the execution of the scriptlets.

Jira:RHEL-90842

OpenSSH ignores invalid RSA hostkeys in **known_hosts**

Before this update, if **known_hosts** contained only a bad hostkey, the SSH connection failed with a **bad**

hostkey: Invalid key length message when OpenSSH received a server hostkey, even if the server had valid hostkeys available. With this update, OpenSSH ignores RSA hostkeys that are invalid due to being too short in the **known_hosts** file. As a result, instead of a failed SSH connection, OpenSSH receives new keys and can establish a connection.

Jira:RHEL-83644[1]

Three RHEL services removed from SELinux permissive mode

The following SELinux domains for RHEL services have been removed from SELinux permissive mode:

- **gnome_remote_desktop_t**
- **pcmsensor_t**
- **samba_bgqd_t**

Previously, these services from packages recently added to RHEL 10 were temporarily set to SELinux permissive mode, which allows gathering information about additional denials while the rest of the system is in SELinux enforcing mode. This temporary setting has now been removed, and as a result, these services now run in SELinux enforcing mode.

Jira:RHEL-82672[1]

GnuTLS supports ML-DSA keys in TLS connections.

With this update, the GnuTLS library supports using X.509 certificates with Module-Lattice-Based Digital Signature Algorithm (ML-DSA) keys in TLS 1.3 connections. For resistance against attacks by quantum computers, the certificate chain and the TLS handshake must be authenticated with a post-quantum algorithm, such as ML-DSA.

Jira:RHEL-64740

OpenSSH server supports Kerberos authentication indicators

When in Match configuration, OpenSSH server supports authentication indicators from Kerberos tickets. If the **GSSAPIIndicators** option is defined in **sshd** configuration, a Kerberos ticket that has indicators but does not match the policy is denied. If at least one indicator is configured, whether for access or denial, tickets without authentication indicators are explicitly rejected. For more information, see the **sshd_config(5)** man page on your system.

Jira:RHEL-40790

DNS over TLS is generally available in RHEL 10.1

Encrypted DNS (eDNS) is generally available to secure all DNS communication by using the DNS-over-TLS (DoT) protocol. You can use eDNS to secure new RHEL installations during boot time, which ensures no plain text DNS traffic is ever sent. You can also convert an existing RHEL system to use eDNS.

To perform a new installation with eDNS, specify the DoT-enabled DNS server by using the kernel command line. If you require a custom CA certificate bundle, you can install it only by using the **%certificate** section in the Kickstart file. Currently, the custom CA bundle can be installed only through Kickstart installation.

On an existing system, configure NetworkManager to use a new DNS plugin, **dnsconfd**, which manages the local DNS resolver (**unbound**) for eDNS. Add kernel arguments to configure eDNS for the early boot process, and optionally install a custom CA bundle.

As a result, you can encrypt all RHEL DNS traffic end-to-end by using the DoT protocol and configure policies to prevent any fallback to insecure protocols. See Securing system DNS traffic with encrypted DNS for more details.

Jira:RHELDOCS-21104[1]

New package: fips-provider-next

The **fips-provider-next** package provides the next version of the FIPS provider that is submitted to the National Institute of Standards and Technology (NIST) for validation. The package is not installed by default because the **openssl-fips-provider** package is the validated OpenSSL FIPS provider. To switch from **openssl-fips-provider** to **fips-provider-next**:

```
# dnf swap openssl-fips-provider fips-provider-next
```

Jira:RHEL-105014[1]

Rsyslog imuxsock provides the new ratelimit.discarded counter

With this update, the **imuxsock** Rsyslog module includes a new counter, **ratelimit.discarded**, which tracks the number of messages dropped due to rate-limiting on the UNIX socket. This enhancement provides administrators with visibility into message loss due to rate-limiting, enabling them to fine-tune their rate-limiting settings and prevent critical logs from being discarded.

Jira:RHEL-96589[1]

The SELinux policy adds rules and type for the qgs daemon

The **qgs** daemon was added to RHEL with the **linux-sgx** package, which supports TDX confidential virtualization. The **qgs** daemon communicates with QEMU over a UNIX domain socket when the guest operating system requests attestation of the virtual machine (VM). To make this possible, the SELinux policy adds a new **qgs_t** type, access rules, and permissions.

Jira:RHEL-87742

audit.cron helps to set up time-based auditd log rotation

With this update, the **auditd.cron** file has been added to the **audit** packages. This enhancement provides a clear, documented example of how to configure time-based **auditd** log rotation by using existing tools. As a result, administrators have a simple, official guide to set up **auditd** log rotation based on time.

Jira:RHEL-77141[1]

Additional services confined in the SELinux policy

This update adds additional rules to the SELinux policy that confine the following **systemd** services:

- **switcheroo-control**
- **tuned-ppd**

As a result, these services no longer run with the **unconfined_service_t** SELinux label, which violated the CIS Server Level 2 benchmark "Ensure No Daemons are Unconfined by SELinux" rule, and run successfully in SELinux enforcing mode.

Jira:RHEL-69450, Jira:RHEL-83267

Rsyslog **imfile** provides the new **deleteStateOnFileMove** option

With this update, the new **deleteStateOnFileMove** parameter has been added to the **imfile** module, available as both a module-level and a per-action option. This enhancement addresses the issue of orphaned state files accumulating in the **spool**/ directory when monitored log files are rotated or moved. By enabling this parameter, you can automatically clean up these obsolete files when log files are moved, preventing disk space from being wasted and simplifying management.

Jira:RHEL-92757[1]

6.3. SOFTWARE MANAGEMENT

RPM supports **spec**-local file attributes and dependency generators

File attributes and their dependency generators are usually shipped in separate packages that you must install before building a package that uses these attributes. However, you might need a file attribute to take effect during the build of the package that ships this attribute. You might also need the file attribute just for building the package, without shipping the attribute at all.

With this update, you can register **spec**-local file attributes and generators by performing the following actions:

1. Define the **%_local_file_attrs** macro. **%_local_file_attrs** accepts a colon-separated list of new attribute names to register directly in your **spec** file.

2. Define one or more dependency generator macros for each attribute, such as **%__NAME_provides** or **%__NAME_path**, where **NAME** is the name of the local file attribute.

RPM then uses the file attributes for dependency generation when the **spec** file is built. As a result, you can create build-time file attributes that are not necessarily meant for installation.

For example, the following **spec** file snippet generates the provides for each packaged file by using the **foobar.sh** script bundled with your package’s sources:

```
Source1: foobar.sh
[...]
%define _local_file_attrs foobar
%define __foobar_provides %{SOURCE1}
%define __foobar_path .*
```

Jira:RHEL-84057

RPM records a checksum of the original package during installation

With this update, RPM records the SHA256 and SHA512 digests of the entire **.rpm** package during its installation. You can then retrieve these digests from the RPM database to verify that the installed package corresponds to a specific **.rpm** file. As a result, you can improve the integrity of your RHEL system by retrospectively verifying that the installed package set matches, bit-by-bit, a known set of **.rpm** packages, such as the ones available in a DNF repository.

To print the package digests of an installed package, use the following command:

```
$ rpm -q --qf "[%{packagedigestalgos:hashalgo} %{packagedigests}\n]" <package_name>
```

You can also customize which digest types are recorded in the database by configuring the new **%_pkgverify_digests** macro, for example:

```
%_pkgverify_digests 8:10
```

Jira:RHEL-84062

System clock skew is reported during dnf transactions

Significant clock skew between a system and the entitlement server can make content repositories unavailable, even on properly registered systems. This is difficult to troubleshoot, particularly when a negative skew makes entitlements appear to start in the future.

With this enhancement, when **subscription-manager** detects a clock skew greater than 2 seconds, the following message is printed to stdout during a **dnf** transaction:

```
The system clock is skewed. There is a time difference of X.Y seconds with the entitlement server.
Please check your clock settings to ensure access to all entitled content.
```

Additional **DEBUG** logging is written to the **/var/log/rhsm/rhsm.log** file when the skew exceeds 2 seconds, changing to a **WARNING** if it exceeds 15 minutes.

For instructions on how to keep your RHEL 10 system clock synchronized with an NTP server, see Configuring time synchronization.

Jira:RHEL-13374[1]

6.4. SHELLS AND COMMAND-LINE TOOLS

Support added for post-quantum cryptography in tog-pegasus

Previously, there was no mechanism to support a classic certificate chain and the **ML-DSA** certificate at the same time.

With this update, two new files **/etc/pki/Pegasus/server-fallback.pem** and **/etc/pki/Pegasus/file-fallback.pem** are provided for **tog-pegasus** server. These files are used to enable loading of classic certificate and key when there is a requirement to use an **ML-DSA** certificate and classic certificate chain at the same time. For more information, see **/usr/share/doc/tog-pegasus/README.RedHat.SSL** .

Jira:RHEL-93093[1]

Support added for post-quantum cryptography in sblim-sfcb

Previously, the package did not use post-quantum key exchange by default if the peer supports it. Also, there was no mechanism to support a classic certificate chain and the ML-DSA certificate at the same time.

With this update, two new configuration options **sslKeyFallbackFilePath** and **sslCertificateFallbackFilePath** are introduced in **sblim-sfcb** server configuration file. These options are disabled by default, but can be used to enable loading of classic certificate and key when there is a requirement to use an **ML-DSA** certificate and classic certificate chain at the same time.

The ECDH ephemeral key generation which prevents post-quantum key exchange by default was disabled in the **sblim-sfcb** server.

Jira:RHEL-93092[1]

Support added for post-quantum cryptography in **openwsman**

Previously,the package did not use post-quantum key exchange by default if the peer supports it. Also, there was no mechanism to support a classic certificate chain and the ML-DSA certificate at the same time.

With this update, two new configuration options **ssl_cert_fallback_file** and **ssl_key_fallback_file** are introduced in **openwsman** server configuration file. These options are disabled by default, but can be used to enable loading of classic certificate and key when there is a requirement to use an **ML-DSA** certificate and classic certificate chain at the same time.

The outdated SSL initialization which prevents post-quantum key exchange by default was removed from the **openwsman** server.

Jira:RHEL-93091[1]

openCryptoki provided in version 3.25.0

The **openCryptoki** packages are provided in version 3.25.0. Support has been added for the following:

- In EP11:
  - PKCS#11 v3.0 SHA3 and SHA3-HMAC mechanisms
  - PKCS#11 v3.0 SHA3 mechanisms and MGFs for RSA-OAEP
  - PKCS#11 v3.0 SHA3 variants of RSA-PKCS and ECDSA mechanisms
  - Opaque secure key blob import via C_CreateObject
- In ICA/Soft:
  - PKCS#11 v3.0 SHAKE key derivation
  - The CKM_AES_KEY_WRAP[_*] mechanisms
  - The CKM_ECDH_AES_KEY_WRAP mechanism
  - Key wrapping with AES-GCM
- In CCA:
  - CCA AES CIPHER secure key types
  - The CKM_ECDH1_DERIVE mechanism
  - Newer CCA versions on s390x and non-s390x platforms
  - CKM_AES_GCM for single-part operations only

- CCA/Soft/ICA: The CKM_RSA_AES_KEY_WRAP mechanism.
- P11KMIP: Added a tool for importing and exporting PKCS#11 keys to a KMIP server.
- ICA: Report mechanisms depending on whether libica is in FIPS mode.

Jira:RHEL-73343[1]

6.5. INFRASTRUCTURE SERVICES

RHEL is now equipped with **dyninst** version 13.0.0

The **dyninst** framework is rebased to upstream version 13.0.0 This version offers the following list of enhancements:

- improved support for AMD GPU binaries.
- improved parsing of x86 instructions and C++ DWARF constructs.

For more information, see the upstream documentation.

Jira:RHEL-87001

RHEL is now equipped with **SystemTap** version 5.3

SystemTap is rebased to version 5.3, and its multithreaded parsing capability now improves startup performance by reducing initialization time by several seconds.

Jira:RHEL-86999

elfutils is now rebased to version 0.193

elfutils 0.193 is now available in RHEL 10.1. The notable changes in this update include:

- **debuginfod** now supports CORS (web application access) in the web API and provides a **--cors** option. The new **--listen-address** option enables binding the HTTP listen socket to a specific IPv4 or IPv6 address. The **debuginfod** client now caches **x-debuginfod-*** HTTP headers alongside downloaded files.
- **libdw** library adds the **dwarf_language** and **dwarf_language_lower_bound** functions, with improved support for DWARF6 language metadata and new language constants for Nim, Dylan, Algol68, V, and Mojo. The **dwarf_srclang** function is forward-compatible with DWARF6 language constants.
- **libdwfl_stacktrace** experimental interface can unwind stack samples into call chains and cache ELF data for multiple processes. This interface initially supports **perf_events** stack sample data and is provided as a Technology Preview.
- **libelf** library has a more robust implementation of **elf_scnshndx** for ELF files with more than 64K sections.
- **readelf** tool improves handling of corrupt ELF data. The output of the **--section-headers** option now includes a key to explain section flag meanings.

Jira:RHEL-86966

valgrind has been upgraded to upstream version 3.25.1

The upgrade from version 3.24.0 (RHEL 10.0) to the upstream version 3.25.1 (RHEL 10.1) provides the following notable enhancements:

- Added support for zstd-compressed debug sections.
- Extended to Linux syscalls: landlock*, io_pgetevents, open_tree, move_mount, fsopen, fsconfig, fsmount, fspick, userfaultfd.
- Enhanced file-descriptor tracking: **--track-fds=yes** and **--track-fds=all** apply the same behavior to inherited file descriptors as to standard input, standard output, and standard error.
- New option **--modify-fds=high** (use with **--track-fds=yes**) allocates higher-numbered descriptors first to help detect descriptor reuse issues.
- Helgrind configuration: warnings for **pthread_cond_signal** and **pthread_cond_broadcast** with an unlocked mutex are now controlled by **--check-cond-signal-mutex=yes|no** (default: no).

Architecture-specific enhancements:

- New IBM Z (**s390x**) NNPA hardware support.

Jira:RHEL-86988

valgrind package split into subpackages for flexible installation

Before this update, the **valgrind** package included all core functionality, post-processing scripts, GDB integration, and documentation in a single package which required you to install all components, even if you only needed specific features.

With this update, the **valgrind** package has been split into multiple subpackages. You can install only the components you require, such as the core **valgrind** functionality, post-processing scripts, GDB integration, or documentation.

Jira:RHEL-75470[1]

jemalloc 5.3.0 is integrated within Varnish

Before this update, some users reported excessive memory usage in Varnish following upgrades to newer versions of Red Hat Enterprise Linux. Despite setting explicit memory limits (for example, **-s malloc,1G**), memory consumption continued to grow over time.

With this enhancement, the **jemalloc** memory allocator library (version 5.3.0) is integrated within the Varnish package, replacing default **glibc malloc**. The integration of **jemalloc** 5.3.0 results in lower memory consumption, better performance, and greater memory stability for Varnish deployments, especially in high-load or long-running environments.

Jira:RHEL-45756[1]

The BrowseOptionsUpdate directive is now available in RHEL

The **BrowseOptionsUpdate** directive determines the source and update frequency of default printing options. It specifies whether the system retrieves options from a local system or a remote printing server, and if it updates them at service startup, at certain intervals, or not at all.

You can now add the **BrowseOptionsInterval** directive and its value to the **/etc/cups/cups-browsed.conf** file to achieve the required behavior. The directive offers these values:

- **None** (default): A local file, created from previous sessions, loads default options.
- **Static**: The **cups-browsed** service retrieves default options from the remote server when it starts.
- **Dynamic**: The system updates default options according to the **BrowseInterval** value, also defined in the **/etc/cups/cups-browsed.conf** file.

Note: You need to restart the service after changing the **BrowseOptionsInterval** directive values.

Jira:RHEL-87180[1]

6.6. NETWORKING

NetworkManager and Nmstate support configuring IPv4 forwarding per interface

With this enhancement, NetworkManager can enable and disable IPv4 forwarding per network interface. This enables granular control directly in NetworkManager connection profiles, and updating **sysctl** kernel settings is no longer required. If you enable the **ipv4.forwarding** parameter in a profile, the corresponding interface acts as a router and forwards IPv4 packets. With the default value **auto**, NetworkManager enables IPv4 forwarding if any shared connection is active and, in other cases, it uses the kernel default value.

This feature is also available in Nmstate.

Jira:RHEL-89582

KTLS now supports rekeying for TLS 1.3

Kernel Transport Layer Security (KTLS), which is an unsupported Technology Preview in RHEL, now supports in-kernel rekeying for TLS 1.3. Previously, long-lived sessions with large data transfers were not possible because only a limited number of bytes could be sent with the initial key. With this enhancement, updates now occur seamlessly during an active session, supporting the transfer of large amounts of data without applications needing to restart connections. Note that, to use this feature, user-space libraries, such as OpenSSL and GnuTLS, must also support KTLS rekeying capability.

This enhancement supports rekeying only for TLS 1.3 and not renegotiation in TLS 1.2.

Jira:RHEL-86020[1]

Nmstate now supports the mtu and quickack route options

With this enhancement, you can use Nmstate to set the **mtu** and **quickack** route options. These settings are important for optimizing the network performance if the maximum transmission unit is different from the default and for tuning the TCP acknowledgment behavior. As a result, you now have more precise control over network traffic behavior.

Jira:RHEL-84768

Nmstate now supports configuring FEC settings for network interfaces

With this enhancement, you can now use Nmstate to apply Forward Error Correction (FEC) modes, such as **RS-FEC**, **Base-R** and **Disabled** to interfaces. These settings are crucial for improving data transmission reliability by detecting and correcting errors without retransmission. As a result, you can now use Nmstate to apply FEC settings instead of manually configuring them or using platform-specific tools.

Jira:RHEL-84766

An NBFT parser was added to **nm-initrd-generator**

NVMe Boot Firmware Table (NBFT) is a standard method for firmware to pass network and storage configuration from the pre-boot environment directly to the operating system by using an ACPI table. The **nm-initrd-generator** utility now uses this parser to automatically detect and apply this configuration, and creates the necessary connections without manual setup. This implementation replaces the **95nvmf** module in **dracut** and relies on **systemd** automation for a more streamlined and robust boot sequence.

Jira:RHEL-83058

NetworkManager now supports fixed subnet IDs for downstream interfaces when using IPv6 prefix delegation

With this enhancement, you can now specify a fixed subnet ID for downstream interfaces in NetworkManager when you use IPv6 prefix delegation. In previous releases, when you rebooted the system, the subnet ID for these interfaces could change. With a fixed subnet ID, IPv6 addresses assigned to devices in the downstream network do not change when you reboot the RHEL host.

Jira:RHEL-81948

Nmstate now supports configuring routes by using a MAC address instead of an interface name

With Nmstate, you can create a network connection by assigning it to the MAC address of an interface. With this enhancement, you can use the profile name instead of the interface name in the **next-hop-interface** parameter in the routing configuration. With this feature, you can create static routes without knowing the interface name.

Jira:RHEL-80547[1]

Nmstate can assign settings to network interfaces based on PCI addresses

With this enhancement, you can use Nmstate to set up network interfaces based on their PCI address instead of a device name. Use this feature to ensure consistent configuration across nodes in a cluster. For further details, see Configuring an Ethernet connection with a dynamic IP address by using nmstatectl with a device path and Configuring an Ethernet connection with a static IP address by using nmstatectl with a device path.

Jira:RHEL-80116

Nmstate now supports egress and ingress priority mapping for VLAN interfaces

NetworkManager already supports configuring traffic priority mapping for VLAN interfaces. With this enhancement, the Nmstate library can also handle both egress and ingress priority quality of service (QoS) mapping rules. As a result, you can use Nmstate to create VLANs and define bidirectional priority mapping, helping manage traffic more precisely and efficiently.

Jira:RHEL-78334[1]

nmtui now supports configuring the loopback interface

NetworkManager already supports configuring the loopback interface by using the **nmcli** utility. This enhancement adds the same functionality to the **nmtui** application. As a result, you can configure IP addresses and routes on the loopback interface.

Jira:RHEL-70484

The NetworkManager-libreswan plugin supports using the Libreswan default values

With this enhancement, you can set the **no-nm-default** property in Libreswan VPN connection profiles to **true** to use Libreswan's instead of NetworkManager's default values. This ensures the compatibility with configurations defined for native Libreswan. As a result, you can now, for example, configure subnet-to-subnet tunnels.

Jira:RHEL-34057

Bond configurations in Nmstate support optimization settings

With this enhancement, the Nmstate API supports the following bond options:

- **lacp_active**: Defines whether or not the Linux kernel periodically sends Link Aggregation Control Protocol Data Unit (LACPDU) frames. You can use this setting only in the 802.3ad bond mode.
- **ns_ip6_target**: Lists the IPv6 addresses to use as IPv6 monitoring peers when you set the **arp_interval** parameter to a value larger than 0.

As a result, administrators can use these settings to optimize a network bond to ensure stable connections, efficient bandwidth, and IPv6 compatibility.

Jira:RHEL-1415

iproute rebased to version 6.14.0

The **iproute** package has been updated to upstream version 6.14.0.

Notable enhancements:

- The **ip nexthop** command supports 16-bit **nexthop** weights.
- The **ip link rmnet** command supports flag handling.
- The **ip lwtunnel** command supports setting and getting the 'tunsrc' attribute.
- The **ip monitor** command adds support for monitoring multicast addresses (**ip monitor maddress**).
- The **ip rule** command supports the 'dscp' selector.
- The **ip rule** command supports flow labels.
- The **ip route** command supports IPv6 flow labels.
- The **ip address** and **ip link show** commands support the 'down' filter.
- The **tc flower** filter supports matching on tunnel metadata.
- The **tc fq** queuing discipline supports the **TCA_FQ_OFFLOAD_HORIZON** attribute.

- The **tc** utility supports the **Hold**/**Release** mechanism in Time-Sensitive Networking (TSN) as specified in the IEEE 802.1Q-2018 standard.
- The **rdma monitor** command adds support for monitoring Remote Direct Memory Access (RDMA) events.
- The **vdpa** utility supports setting the MAC address.
- Several man pages were improved.

Notable bug fixes:

- Some memory leaks were fixed.
- The error checking of the **ip netconf** command was fixed to prevent unnecessarily strict errors.
- Custom **iproute2** settings in the **/etc/iproute2/** directory work as expected.

Jira:RHEL-90493

New network packet drop reasons and MIB counters

The kernel’s networking stack now provides more detailed reasons when it drops network packets. This enhancement also adds two new Management Information Base (MIB) counters: **LINUX_MIB_PAWS_TW_REJECTED** and **LINUX_MIB_PAWS_OLD_ACK**. As a result, debugging and diagnosing network problems, is now easier.

Jira:RHEL-88891[1]

The nft monitor trace command now displays connection tracking information

You can now use the **nft monitor trace** command to display details about connection tracking. This feature simplifies debugging connections and helps to better understand connection states.

Jira:RHEL-87758[1]

The fwctl subsystem has been added to the kernel

If the kernel lock-down feature is enabled, the kernel does not allow access to **resource0** files in the **/sys/** directory and PCI config spaces for security reasons. The **fwctl** kernel subsystem manages communication with the firmware in software-defined devices, such as the **mlx5** network interface controller. This subsystem establishes a standardized and secure Remote Procedure Call (RPC) interface, that enables user-space applications to interact with device firmware for diagnostics, configuration, and updates. In addition to the new subsystem, the **mstflint** utility now also uses the **fwctl** subsystem, and the utility functions fully in these secure environments.

Jira:RHEL-86015[1]

The ice driver now supports reducing the MSI-X vector usage for a PF to free vectors for associated VF

With this enhancement, you can now reduce the Message Signaled Interrupts eXtended (MSI-X) vectors allocated to a physical function (PF) to ensure that a sufficient number of vectors are available for associated virtual functions (VFs). For details, see Reducing the MSI-X vector usage for a physical function to free vectors for associated virtual functions.

Jira:RHEL-80554[1]

The **named** and **dnssec** utilities now support OpenSSL providers for hardware tokens

Before this update, support for using hardware security tokens to store private keys for DNSSEC zone signing was unavailable after the removal of OpenSSL ENGINEs. This functionality was required both for directly using hardware tokens with the **named** service and for the DNSSEC feature in the **ipa-server-dns** package.

With this update, the **named** and **dnssec** command-line utilities have been updated to support OpenSSL providers.

As a result, you can use OpenSSL providers to access both hardware and software tokens to store private keys. This restores the ability to use hardware tokens directly in the **named** service and enables the DNSSEC zone signing feature in the **ipa-server-dns** package.

Jira:RHEL-33729

NetworkManager and Nmstate support configuring IPv4 forwarding per interface

With this enhancement, NetworkManager can enable and disable IPv4 forwarding per network interface. This enables granular control directly in NetworkManager connection profiles, and updating **sysctl** kernel settings is no longer required. If you enable the **ipv4.forwarding** parameter in a profile, the corresponding interface acts as a router and forwards IPv4 packets. With the default value **auto**, NetworkManager enables IPv4 forwarding if any shared connection is active and, in other cases, it uses the kernel default value.

This feature is also available in Nmstate.

Jira:RHEL-59083

6.7. KERNEL

Kernel version in RHEL 10.1

Red Hat Enterprise Linux 10.1 is distributed with the kernel version 6.12.0-124.8.1.

Perf core counters supported on Intel Panther Lake CPUs

Previously, users could not monitor hardware events by using perf core counters on Intel Panther Lake CPUs. With the addition of Panther Lake support in the **perf** package, users can access hardware event monitoring on this microarchitecture.

Jira:RHEL-47451[1]

The default measurement module for rteval is now rtla timerlat for better tracing of problem latencies

With this enhancement, you should be able to easily identify the source of problem latencies. The required cyclictest measurement module can be chosen using the rteval.config file.

Jira:RHEL-97541[1]

kpatch-dnf plugin is updated with improved kernel management

Before this update, the **kpatch-dnf** plugin did not align kernel upgrades with **kpatch** support. As a consequence, administrators might install or upgrade to kernels that were not supported by **kpatch**, thereby increasing the risk of running unsupported kernels and reducing system stability.

With this update, the **kpatch-dnf** plugin enables administrators to focus kernel updates on those supported by **kpatch**. As a result, system upgrades are more reliable, and overall stability is improved.

Jira:RHEL-85686[1]

perf tool rebased to upstream v6.14

The **perf** tool and its kernel backend are rebased to align with upstream version v6.14. This update introduces several enhancements and bug fixes. Most notably, the following:

- Fixed the memory leak issue in the RAPL code.
- Added the per-core energy tracking support for AMD.
- Addressed memory leaks in **perf trace**.
- Added Processor Trace Trigger Tracing (PTTT) support in the **perf** tool.
- Supports the RDPMC metrics in clear mode.
- Added RAPL energy events support in the **perf** tool for the ARL-U platform.

These changes improve performance analysis and resolve known issues in the **perf** tool.

Jira:RHEL-77936[1]

Added support for virtio devices

Before this update, **virtio** devices inside of KVM guests were all listed as type **generic-ccw**. With this enhancement, you can easily identify which device type is connected at which device number by using the **lszdev** command:

```
# lszdev
TYPE ID ON PERS NAMES

virtio-balloon 0.0.0007 yes no
virtio-blk 0.0.0000 yes no vda
virtio-console 0.0.0004 yes no
virtio-gpu 0.0.0002 yes no
virtio-input 0.0.0005 yes no
virtio-input 0.0.0006 yes no
virtio-net 0.0.0001 yes no enc1
virtio-scsi 0.0.0003 yes no
virtio-vsock 0.0.0008 yes no
```

This enhancement also introduces additional **chpstat** fixes for Red Hat Enterprise Linux 10.0.z, improving DPU utilization scaling in reports (**s390utils** and **s390-tools**).

Jira:RHEL-73341[1]

Intel Arrow Lake U RAPL energy events support in kernel

The **kernel** package now supports RAPL (Running Average Power Limit) energy performance counters for the Intel Arrow Lake U microarchitecture. With this enhancement, the **perf** tool identifies power-consumption events for Arrow Lake U platforms to monitor energy usage for CPU cores, GPUs, packages, and system domains.

Jira:RHEL-53584[1]

Adaptive PEBS enables counter snapshotting support in **perf** on Intel Panther Lake

Before this update, the Linux kernel's perf tool relied on software-based sample reads to collect performance event data. This approach introduced minor timing gaps and additional overhead when reading counters after an event overflow. With this update, adaptive PEBS counter snapshotting is available on Intel Panther Lake CPUs. With this feature, the kernel captures programmable counters, fixed-function counters, and performance metrics directly in the PEBS record by using the PEBS format version 6.

As a result, counter snapshotting provides a more accurate and lower-overhead alternative to software sample reads, improving performance monitoring and analysis capabilities.

Jira:RHEL-47443[1]

Intel Trace Hub supports Intel Panther Lake

Before this update, the **kernel** package did not support Intel Panther Lake (P, H, U variants) in Intel Trace Hub. With this update, device IDs for Panther Lake platforms are added to Intel Trace Hub in the **kernel** package.

As a result, systems based on Panther Lake can use Intel Trace Hub features for enhanced debugging and tracing capabilities.

Jira:RHEL-47423[1]

Perf uncore event support for Intel Clearwater Forest

The **perf** package adds uncore event monitoring on Clearwater Forest microarchitecture. With this enhancement, the **perf** package supports the uncore event monitoring on Clearwater Forest systems. As a result, users can perform advanced performance analysis and debugging on supported hardware.

Jira:RHEL-45094[1]

Perf core event support for Intel Clearwater Forest

The **perf** package adds core event monitoring on Clearwater Forest microarchitecture. As a result, users can monitor and analyze core-level performance events on Intel Clearwater Forest systems by using **perf**.

Jira:RHEL-45092[1]

AMD Milan CPUs support per-core energy tracking with RAPL **perf** events

Before this update, energy monitoring on AMD systems was limited to package-level measurements. With this update, the **kernel** package supports per-core energy tracking through Running Average Power Limit (RAPL) performance events on AMD Milan CPUs. As a result, you can measure and analyze energy consumption at the individual core level for more granular performance and power management.

Jira:RHEL-24184[1]

Intel Arrow Lake H microarchitecture support added to **intel_th**

Before this update, Intel Trace Hub did not recognize Arrow Lake H NPK device IDs, which limited trace and debugging capabilities for systems using this hardware. With this update, the **intel_th** package supports the Intel Arrow Lake H microarchitecture in Intel Trace Hub. With the new support, users have enhanced tracing and debugging features on Arrow Lake H platforms.

Jira:RHEL-20109[1]

PerfMon support enabled for Intel Arrow Lake H in kernel

With this update, the **kernel** package provides PerfMon support for Core, Uncore, Cstate, and MSR features on the Intel Arrow Lake H microarchitecture. As a result, you can monitor and analyze performance metrics specific to Arrow Lake H systems by using the **perf** tool.

Jira:RHEL-20093[1]

KVM modules are integrated into the Realtime kernel package

This update removes the generation of KVM module packages for the Realtime kernel in RHEL, aligning with the decision to make the Realtime kernel a deployment option for base RHEL. This change streamlines the deployment process, integrating KVM modules directly into the Realtime kernel package and eliminating the separate **kernel-rt-kvm** package. As a result, users will experience a more seamless and efficient setup when deploying the Realtime kernel on RHEL, improving the overall user experience.

Jira:RHEL-62687[1]

Added Processor Trace Trigger Tracing (PTTT) support in the perf tool

With this update, performance analysis is elevated through the introduction of Processor Trace (PT) Trigger tracing. This enables software to select specific events as trigger points for pausing and resuming tracing activity, thereby enhancing the efficiency and accuracy of performance monitoring. This leads to more efficient and targeted tracing, ultimately offering a clearer comprehension of their application’s performance.

Jira:RHEL-45090[1]

python-drgn rebased to version 0.0.31

python-drgn has been rebased to version 0.0.31. This update introduces several enhancements and new features:

- Added support for **debuginfod**, which enables automatic retrieval of debugging information from debuginfod servers.
- A new Module API, which provides improved extensibility and integration capabilities.
- Kernel stack unwinding without debugging symbols, allowing stack traces to be generated even when debug symbols are unavailable.

For a complete list of changes, see the upstream changelogs:

- 0.0.31: https://github.com/osandov/drgn/releases/tag/v0.0.31
- 0.0.30: https://github.com/osandov/drgn/releases/tag/v0.0.30

Jira:RHEL-86265

eBPF subsystem rebased to version 6.14.

The eBPF subsystem is rebased to the Linux kernel upstream version v6.14. This version includes the following changes and enhancements:

- Support for **uprobe** session probes.
- Support for **bpf_fastcall**, a special annotation for eBPF helpers and kernel functions (**kfuncs**), which allows optimizing the execution of such helpers and functions.
- New **kmem_cache** eBPF iterator to allow eBPF programs to iterate over entries in **/proc/slabinfo** or **/sys/kernel/slab**.
- Support for a private stack in eligible eBPF programs, which allows preventing the kernel stack overflows in nested eBPF programs.
- eBPF verifier improvement, which allows programs to avoid a NULL check on statically known map lookup keys.
- Removal of **"helper that may corrupt user memory!"** warning message when using **bpf_probe_write_user**.
- Prevent infinite loops when using a combination of tail calls and **freplace**.
- Avoid potential kernel crashes when attaching eBPF programs to raw tracepoints with NULL arguments.
- The **bpf_timer** destroy procedure used to cause the issues but that has been fixed by the rebase.
- The **bpf_local_storage** in preventing the **kmalloc**, causing **"sleeping function called from invalid context"** issues while using eBPF on the real-time kernel.

Jira:RHEL-78201[1]

perf tool rebased to upstream v6.15

The **perf** tool and its kernel backend are rebased to align with upstream version v6.15. This update introduces several enhancements and bug fixes. Most notably, the following:

- Added the **--code-with-type** option to **perf annotate**, enabling decoding of data structures from pointers.
- Refactored s390 **cpum_sf** and **cpum_cf** components.
- Addressed memory leaks in **perf trace**.
- Introduced hardware event support for RISCV CPUs.
- Extended functionality for the **python-perf** module.
- Enhanced **perf report** to display workload per parent and child processes.
- Updated PMU events and metrics for various Intel CPUs.
- Enabled Processor Trace (PT) Trigger tracing on Intel platforms.

These changes improve performance analysis, extend hardware support, and resolve known issues in the **perf** tool.

Jira:RHEL-78197[1]

crash rebased to 9.0.0

The **crash** package, which provides a kernel analysis utility for live systems and various types of dump files, has been rebased to upstream version 9.0.0. This version provides several fixes and enhancements, most notably the following:

- The internal **gdb** database has been updated to version 16.2.
- The **crash** utility now supports cross-compilations.

Jira:RHEL-76107

Default configuration now disables jitter entropy source in **rng-tools**

The jitter entropy source is now disabled by default in **rng-tools**. Modern CPUs provide a hardware entropy source, and most virtual machines offer the **/dev/hwrng** device as an entropy source from the virtual host. In these environments, the jitter entropy source consumes unnecessary CPU cycles. For older hardware without a hardware entropy source, you can explicitly enable the jitter entropy source in **/etc/sysconfig/rngd**.

As a result, the **rngd** daemon no longer consumes CPU cycles unnecessarily on systems that have hardware entropy sources.

Jira:RHEL-91113

stalld no longer conflicts with the working of the dl-server

With this release, the **stalld** functionality detects the **dl-server** in the host kernel and boosts only the tasks that the **dl-server** fails to run. Currently, **dl-server** does not boost FIFO tasks. You might prefer to keep using **stalld** in a system upgrade and disable **dl-server**. The **dl-server** is the only entity responsible for running the starving tasks.

Jira:RHEL-73883

6.8. BOOT LOADER

Secure boot **shim** signing for RHEL 10 on **x86_64** and **aarch64**

RHEL 10 requires a signed **shim** binary to enable secure boot on AMD and Intel 64-bit architectures and on the 64-bit ARM architecture. Without a signed and trusted **shim**, systems with enforced secure boot did not boot, which affected both enterprise and cloud deployments.

With this release, the **shim** package was signed and updated for **x86_64** and **aarch64**. On **x86_64**, **shim** is signed by Microsoft Windows UEFI Driver Publisher and includes Red Hat Secure Boot CA 5 and CA 8 in the vendor database. On **aarch64**, **shim** is signed by Microsoft UEFI CA 2023 and includes Red Hat Secure Boot CA 8. The SBAT entries were updated to the latest levels.

As a result, RHEL boots with the secure boot feature enabled. Additionally, the fallback works properly, and all other boot loader components are correctly signed.

Jira:RHEL-81188

6.9. FILE SYSTEMS AND STORAGE

multipathd supports file-based sockets

With this update, the **multipathd** daemon listens for commands on a file-based socket **/run/multipathd.socket** in addition to the abstract namespace socket. You can communicate with the host's **multipathd** daemon from within a container by using a bind mount for the new socket file.

Jira:RHEL-82180[1]

LVM RAID repairs volumes after multiple simultaneous device failures

With this enhancement, you can use the **lvconvert --repair /dev/*VG-name*/*LV-name*** command to reintegrate missing RAID devices back into a striped RAID (raid4, raid5, and raid6). This repair process works even when the number of temporarily missing devices exceeds the fault tolerance of the RAID level, allowing for recovery once the devices reappear. Note that you must unmount and deactivate the volume and the file system on top before repairing them.

Jira:RHEL-89832

6.10. HIGH AVAILABILITY AND CLUSTERS

The IPaddr2 resource agent now detects network link failures

Before this update, the **IPaddr2** resource agent did not monitor the link state of the network interface. As a consequence, an **IPaddr2** resource continued to report success on a node even if the underlying interface was in a **DOWN** or **LOWERLAYERDOWN** state, preventing the cluster from recovering the resource on another node.

With this release, the **IPaddr2** agent has been enhanced to check the interface's link status.

As a result, an **IPaddr2** resource correctly fails if its network interface goes down, allowing for a proper failover. You can disable this new default behavior by setting the **check_link_status=false** parameter in the resource configuration.

Jira:RHEL-85014[1]

AWS resource agents reuse IMDS tokens to improve reliability

Before this update, the AWS resource agents requested a new Instance Metadata Service (IMDS) token for every operation. This could lead to a large number of API calls on a single node, which increased the risk of resource failures, especially in environments with many AWS resources.

With this update, the AWS resource agents cache and reuse IMDS tokens until they expire.

As a result, the volume of API calls to the AWS metadata service is significantly reduced. This improves the performance and reliability of AWS resources in high-availability clusters.

Jira:RHEL-81237[1]

The awsvip resource agent allows specifying a network interface

Before this update, the **awsvip** resource agent always assigned the virtual IP address to the primary network interface of an EC2 instance. It was not possible to use a secondary network interface for the resource.

With this enhancement, an **interface** parameter has been added to the **awsvip** agent.

By using this parameter, you can specify to which network interface the agent should assign the virtual IP, which enables more flexible network configurations in AWS.

Jira:RHEL-81236[1]

The fence_sbd agent can automatically detect the SBD device

Before this update, when configuring a **fence_sbd** resource, you were required to explicitly specify the SBD device path by using the **devices** parameter.

With this update, the **fence_sbd** agent can now retrieve the device configuration from the system.

As a result, if you do not set the **devices** parameter when creating the **fence_sbd** resource, the agent automatically uses the device specified in the **SBD_DEVICE** variable within the **/etc/sysconfig/sbd** file.

Jira:RHEL-79799[1]

Watchdog device listing provides more detailed information

Before this update, when listing available watchdog devices, the output only displayed the device path, such as **/dev/watchdog0**. This made it difficult for administrators to distinguish between multiple devices on the same system.

With this update, the output includes the device path, identity, and driver for each watchdog. This allows for easy identification and selection of the correct device.

Jira:RHEL-76176

New fence agent for Nutanix AHV virtualization is now available

Previously, Red Hat High Availability Add-On did not provide a dedicated fence agent for Nutanix Acropolis Hypervisor (AHV) environments.

With this enhancement, the **fence_nutanix** agent is added.

As a result, you can now configure STONITH for cluster nodes running on the Nutanix AHV platform, enabling fully supported high-availability deployments.

Jira:RHEL-68322[1]

pcs warns users before removing the last fencing device

Before this update, **pcs** allowed users to disable or remove the last fencing device from a cluster without a warning. This could inadvertently leave the cluster in an unsupported state without any STONITH or SBD fencing configured.

With this enhancement, **pcs** now includes a safety check to prevent the accidental removal of all fencing mechanisms.

As a result, if you attempt an action that would leave the cluster without any fencing, **pcs** displays an error and blocks the change by default. For example, this occurs when you try to remove the last STONITH resource while SBD is disabled. You can override this safety check to force the change if needed.

Jira:RHEL-66607

pcs provides more detailed error messages for failed CIB updates

Previously, when a CIB update failed when using the **pcs cluster edit** or **pcs cluster cib-push** commands, the error message provided by Pacemaker was generic. It did not explain the specific reason for the failure, which made troubleshooting the invalid configuration difficult.

With this enhancement, **pcs** is updated to request a detailed validation check from Pacemaker upon a failed CIB push.

As a result, when a CIB update is rejected, **pcs** now displays a specific error message explaining what is wrong with the configuration.

Jira:RHEL-63186

The pcs alert config command now supports multiple output formats

Previously, the **pcs alert config** command displayed its output only in a human-readable plain text format. This format was not suitable for machine parsing or for easily replicating the configuration.

With this enhancement, a new **--output-format** option has been added to the **pcs alert config** command.

As a result, you can now display the configured alerts in one of three formats:

- **text**: Displays the output in plain text. This is the default format.
- **json**: Displays the output in a machine-readable JSON format, which is useful for scripting and automation.
- **cmd**: Displays the output as a series of **pcs** commands, which you can use to re-create the same alert configuration on a different system.

Jira:RHEL-44347

The pcs resource meta command is improved to support bundles and prevent guest node misconfiguration

Previously, the **pcs resource meta** command did not support managing meta attributes for bundle resources. Additionally, the command did not prevent users from incorrectly modifying the connection parameters of a guest node, which could lead to a misconfigured resource.

With this enhancement, the **pcs resource meta** command has been rewritten.

As a result, you can now use **pcs resource meta** to update meta attributes for bundle resources. In addition to this, when using the command on a guest node, it now prevents unintended changes to connection parameters, avoiding potential misconfigurations.

Jira:RHEL-35407

A new pcs command is available for renaming a cluster

Previously, it was not possible to change the name of an existing cluster by using **pcs** commands. Administrators had to perform a series of manual steps, which were complex and could lead to errors.

With this enhancement, the **pcs cluster rename** command has been introduced.

As a result, you can now easily change the name of an existing cluster. To rename your cluster, run the following command:

```
pcs cluster rename <new-name>
```

Jira:RHEL-22423

The **pcs node attribute** and **pcs node utilization** commands now support multiple output formats

Previously, the **pcs node attribute** and **pcs node utilization** commands displayed their output only in a human-readable plain text format. This format was not suitable for machine parsing or for easily replicating the configuration.

With this enhancement, a new **--output-format** option has been added to the **pcs node attribute** and **pcs node utilization** commands.

As a result, you can now display the configured node attributes and utilization in one of three formats:

- **text**: Displays the output in plain text. This is the default format.
- **json**: Displays the output in a machine-readable JSON format, which is useful for scripting and automation.
- **cmd**: Displays the output as a series of **pcs** commands, which you can use to re-create the same configuration on a different system.

Jira:RHEL-21050

pcs automatically validates the CIB for potential issues

Previously, the **pcs** utility did not automatically run advanced validation checks on the Cluster Information Base (CIB). As a consequence, certain cluster misconfigurations could remain undetected during routine operations.

With this enhancement, **pcs** has been updated to integrate Pacemaker's CIB validation tool into its workflow.

As a result, **pcs** now automatically performs a validation check and displays the results when you run the **pcs status**, **pcs cluster edit**, or **pcs cluster cib-push** commands.

Jira:RHEL-7681

New crypt resource agent for managing encrypted volumes

Previously, Red Hat High Availability Add-On did not provide a resource agent for managing encrypted devices. This made it difficult to configure volumes encrypted with **cryptsetup** as highly available resources within a Pacemaker cluster.

With this update, the new **crypt** resource agent has been introduced.

As a result, you can configure encrypted local or network volumes as cluster resources. The **crypt** agent uses **cryptsetup** to manage these devices. It supports unlocking volumes with a standard **key_file** and also supports network-bound unlocking using **tang**/**clevis**.

Jira:RHEL-13089[1]

6.11. DYNAMIC PROGRAMMING LANGUAGES, WEB AND DATABASE SERVERS

The PostGIS extension is available for PostgreSQL

This enhancement adds the PostGIS extension to PostgreSQL. With this extension, PostgreSQL supports geographic objects, enabling spatial queries and analysis for Geographic Information System (GIS) applications, such as mapping, geolocation, and distance calculations within a relational database.

Jira:RHEL-81633[1]

6.12. COMPILERS AND DEVELOPMENT TOOLS

glibc now supports **sched_setattr** and **sched_getattr** for advanced scheduler options

Previously, **glibc** provided access to only a limited set of Linux scheduler options through functions defined in **<sched.h>**. This limitation required applications to use direct system calls or Linux kernel headers to access advanced scheduling features.

With this enhancement, the extensible scheduler configuration mechanism from **sched_setattr** and **sched_getattr** is now available through the **glibc <sched.h>** header file. This change includes support for additional scheduling policies, such as **SCHED_DEADLINE**.

As a result, applications can select from a wider range of scheduling options without relying on direct system calls or kernel-specific headers, improving portability and flexibility for developers.

Jira:RHEL-58357

Geomap support added for PCP Valkey datasource in **grafana-pcp**

Previously, users could not visualize PCP metrics on a map in Grafana because the PCP Valkey data source did not provide the longitude and latitude labels required for geomap panels. This limitation made it difficult to compare the performance of monitored systems across different locations.

To create a geomap visualization for PCP metrics in Grafana:

1. Create a new panel.
2. Select the geomap panel type.
3. Enter the metric you want to visualize in the query window, as you would for other PCP visualizations.
4. In the Format drop-down menu below the query window, select Geomap.
5. Grafana will automatically detect the longitude and latitude labels and place the data on the map.
6. For additional options and customization, see the Grafana documentation.

With this enhancement, the PCP Valkey datasource in **grafana-pcp** includes longitude and latitude labels from PCP metrics, allowing instances to be accurately placed on a geomap. Users can create geomap visualizations in Grafana to compare system performance geographically.

Jira:RHEL-77946[1]

llvm-toolset rebased to LLVM 20

The **llvm-toolset** is updated to LLVM 20, delivering improved code generation, performance optimizations, and expanded language front-end and library support across C, C++, and Rust workflows. This rebase aligns dependent components in RHEL, including rebuilds for **rust**, **annobin**, **bcc**, **bpftrace**, **qt5-qttools**, and **mesa**. The build is validated with **llvm-20.1.8-1.el10**.

The notable changes are:

- Backend improvements, including fixes for the **ppc64le**
- Optimizations and diagnostics enhancements in Clang and LLVM passes for general performance and reliability
- Toolchain ecosystem refresh with coordinated package rebuilds for compatibility with LLVM 20
- Continued deprecation of older targets, consistent with upstream direction for ARM and MIPS in this stream

Jira:RHEL-80988

GDB now supports IBM’s z17 CPU architecture

The **gdb** package is enhanced to support binaries that use new hardware instructions introduced with IBM’s z17 CPU architecture. This update enables developers and system administrators to debug applications compiled for the latest IBM Z hardware on RHEL 10.1.

Jira:RHEL-56897[1]

GCC Toolset 15 is now available

With this update, **gcc-toolset-15** is now available in RHEL 10.1. The toolset includes the latest supported versions of GCC and related utilities, enabling developers to build, test, and deploy applications by using up-to-date compiler technology.

Jira:RHEL-81745[1]

glibc provides the GLIBC_ABI_GNU2_TLS symbol on x86_64

glibc includes the GLIBC_ABI_GNU2_TLS symbol on x86_64 systems. Programs that use the gnu2 thread-local storage access convention might require this symbol to start. Before this update, if **glibc** did not provide this symbol, affected programs would fail to launch. With this update, programs that depend on GLIBC_ABI_GNU2_TLS start and run as expected.

Jira:RHEL-109625

glibc adds GLIBC_ABI_DT_X86_64_PLT symbol support for x86_64

Before this update, programs that required the **GLIBC_ABI_DT_X86_64_PLT** symbol failed to start when it was not available in **glibc**. With this enhancement, **glibc** includes the **GLIBC_ABI_DT_X86_64_PLT** symbol for **x86_64** systems. With this enhancement, programs requiring this symbol to start now run as expected.

Jira:RHEL-109621

glibc header files updated to align with Linux 6.12 UAPI

The **glibc** header files in Red Hat Enterprise Linux 10 are updated to incorporate the latest Linux User-space API (UAPI)constants for **MAP_***, **PIDFD_***, **SCHED_***, and **SYS_***, from Linux kernel version 6.12. As a result, developers can access new and revised UAPI constants when building applications, ensuring consistency and compatibility with the latest kernel features.

Jira:RHEL-107695

gdb is rebased to version 16.3

This update of **gdb** to version 16.3 in RHEL 10.1 provides the following notable enhancements:

- Removed support for Intel MPX.
- Added support for tagged data pointers, including Intel's Linear Address Masking (LAM) and aarch64's Memory Tagging Extension (MTE).
- Enabled background DWARF reading for improved performance.
- Enhanced Intel Process Trace (**record btrace**):
  - Asynchronous event printing enabled with **set record btrace pt event-tracing**.
  - Ptwrite payloads can now be accessed in Python as **RecordAuxiliary** objects.
- Improved Python integration:
  - Stop events now include a **details** attribute, mirroring MI "*stopped" events.
  - **gdb.Progspace()** no longer creates objects directly; objects must be obtained with other APIs.
  - User-defined attributes can be added to **gdb.Inferior** and **gdb.InferiorThread** objects.
  - Introduced new event source: **gdb.tui_enabled**.
  - Added **gdb.record.clear**, which clears the current recording's trace data.
  - Added modules for handling missing objfiles and debug information.
  - New class **gdb.missing_debug.MissingDebugInfo** can be subclassed to handle missing debug information.
  - New attribute **gdb.Symbol.is_artificial**.
  - New constants for symbol lookup across multiple domains.
  - New function **gdb.notify_mi(NAME, DATA)** emits custom async notifications.
  - New attribute **gdb.Value.bytes** for reading and writing value contents.
  - Added **gdb.interrupt** to simulate a CTRL-C interrupt.
  - New attribute **gdb.InferiorThread.ptid_string** provides the target ID.
- Debug Adapter Protocol (DAP) changes:
  - Updated "scopes" request to include global variables and last return value.

  - "launch" and "attach" requests can be used at any time, effective after "configurationDone".
  - "variables" request no longer returns artificial symbols.
  - Added "process" event and support for the "cancel" request.
  - "attach" request now supports specifying the program.
- Introduced new commands for styling, language frame mismatch warnings, missing objfile handlers, and function call timeouts.
- Enhanced and renamed several commands, including improved error handling for **disassemble** and renaming **set unwindonsignal** to **set unwind-on-signal**.
- Expanded remote packet support, including new packets for file status and memory fetch, and new stop reasons such as **clone**.
- Introduced per-thread event reporting options and address tagging checks.

Jira:RHEL-91382

GCC tuning for IBM z16 is default on s390x

The default tuning for code generated by the **gcc** compiler on the s390x architecture in RHEL 10.1 now aligns with IBM z16.

Before this update, the default tuning for s390x code generation in **gcc** was set for older IBM architectures.

With this update, code compiled with **gcc** on s390x in RHEL 10.1 is tuned for IBM z16 by default. If you need to optimize for a different architecture, you can override this setting by specifying the required architecture with the **-mtune** flag during **gcc** invocation.

Jira:RHEL-86679[1]

Initial support for IBM Z z17 added to glibc

The dynamic loader in **glibc** is enhanced to support detecting IBM z17 CPUs or their specific features. As a result, any IBM z17-optimized libraries installed in the **/usr/lib64/glibc-hwcap/z17/** directory are loaded automatically on z17 systems. This update improves hardware compatibility and performance for IBM Z z17 platforms.

Jira:RHEL-72564[1]

Rust Toolset rebased to version 1.88.0

RHEL 10.1 is distributed with Rust Toolset in version 1.88.0. This update includes the following notable enhancements:

- Rust 2024 Edition is now stable. This is a major opt-in release that enables significant language changes and is the largest edition released to date.
- Use the 2024 Edition with **let** chains, allowing fluent **&&**-chaining of **let** statements within **if** and **while** conditions to reduce nesting and improve readability.
- For high-performance computing, when you enable target features, you can call multiple **std::arch** intrinsics directly in safe Rust, which gives you direct access to specific CPU features.

- **async** closures are now supported, providing first-class solutions for asynchronous programming. These closures allow borrowing from captures and properly express higher-ranked function signatures with the AsyncFn traits.
- Trait upcasting allows coercing a reference to a trait object to a reference of its supertrait, simplifying common patterns, especially with the **Any** trait.
- Cargo now automatically cleans its cache, removing old downloaded files not accessed in 1-3 months, which helps manage disk space.

Rust Toolset is a rolling Application Stream, and Red Hat only supports the latest version. For more information, see the Red Hat Enterprise Linux Application Streams Life Cycle document.

Jira:RHEL-81600

tzdata includes the NEWS file

With this update, the tzdata package includes its NEWS file with each release to provide precise descriptions of time zone data changes. As a result, you can review what changed in detail. Users can review the included NEWS file to understand what changed in the update.

Jira:RHEL-105042[1]

Red Hat build of OpenJDK 25 is available

Red Hat introduces the latest long term support (LTS) release of the Red Hat build of OpenJDK (Open Java Development Kit) 25, a free and open source implementation of the Java Platform, Standard Edition (Java SE). Red Hat build of OpenJDK 25 is available starting from RHEL 10.1. For more information about OpenJDK Life Cycle, Support Policy, and all supported configurations, see the OpenJDK Life Cycle and Support Policy .

OpenJDK 25 includes several enhancements and additions to the Java specification, multiple bug and stabilization fixes, and general performance improvements and new features, such as the following improvements:

- Java Flight Recorder enhancements (cooperative sampling, method timing and tracing)
- Generational Shenandoah garbage collector
- Late barrier expansion and region pinning for the G1 garbage collector
- Ahead-Of-Time class loading and linking
- Compact object headers
- Synchronize virtual threads without pinning
- Compact source files and instance main methods
- Unnamed variables and patterns
- Scoped values
- Stream Gatherers
- Launch multi-file source-code programs

For the complete list of new features since the last LTS release, see JEPs in JDK 25 integrated since JDK 21.

Jira:RHEL-100678[1]

6.13. IDENTITY MANAGEMENT

ipa-healthcheck now warns about expiring certificates

With this update, the **ipa-healthcheck** tool now evaluates user-provided HTTP, DS, and PKINIT certificates for expiration and provides warnings 28 days before their expiration date. This is to prevent certificate expirations going potentially unnoticed, which can lead to downtime.

Jira:RHELDOCS-20303[1]

ansible-freeipa rebased to 1.15.1

The **ansible-freeipa** package, which provides modules and roles to manage Red Hat Identity Management (IdM) environments, has been rebased from version 1.13.2 to 1.15.1. The update includes the following enhancement:

- The **freeipa.ansible_freeipa** collection that the **ansible-freeipa** RPM package provides is now compatible with the namespace and name of the **redhat.rhel_idm** collection provided by Red Hat Ansible Automation Hub (RH AAH). If you have installed the RPM package, you can now run playbooks that reference the AAH roles and modules. Note that internally, the namespace and names from the RPM package are used.

Jira:RHELDOCS-20257[1]

Healthcheck warns if krbLastSuccessfulAuth is enabled

Enabling the **krbLastSuccessfulAuth** setting in the **ipaConfigString** attribute can lead to performance issues if large numbers of users are authenticating at the same time. Therefore, it is disabled by default. With this update, **Healthcheck** displays a message if **krbLastSuccessfulAuth** is enabled, warning about the possible performance problems.

Jira:RHEL-84771[1]

IdM now supports UIDs up to Linux maximum UID limit for legacy systems compatibility

With this update, you can now use User and Group IDs up to 4,294,967,293, or 2^32-1. This aligns IdM’s maximum with the Linux UID limit and can be useful in rare cases where the standard IdM range, up to 2,147,483,647, is insufficient. Specifically, it enables IdM deployment alongside legacy systems that require the full 32-bit POSIX ID space.

> **WARNING**
>
> In standard deployments, IdM reserves the 2,147,483,648 - 4,294,836,223 range for subIDS. Using the 2^31 to 2^32-1 UID range requires disabling the subID feature and therefore conflicts with modern Linux capabilities.

To enable UIDs up to 2^32-1:

1. Disable the subordinate ID feature:

```
$ ipa config-mod --addattr ipaconfigstring=SubID:Disable
```

2. Remove any existing subordinate ID ranges:

```
$ ipa idrange-del <id_range>
```

3. On the IdM server, ensure the internal DNA plugin configuration is correctly removed:

```
# ipa-server-upgrade
```

4. Add a new local ID range that covers the 2^31 to 2^32-1 space. Ensure that you define RID bases for this new range so that IdM can generate SIDs properly for users and groups.

NOTE

You can only disable the subordinate ID feature if no subordinate IDs have been allocated yet.

Jira:RHEL-67686[1]

samba rebased to version 4.22.4

The **samba** package has been updated to upstream version 4.22.4. This version provides bug fixes and enhancements, most notably the following:

- Samba supports Server message block version 3 (SMB3) directory leases. With this enhancement, clients can cache directory listings, which reduces network traffic and improves performance.

- Samba supports querying domain controller (DC) information by using TCP-based LDAP or LDAPS, as an alternative to the traditional UDP method on port 389. This enhancement improves compatibility with firewall-restricted environments. You can configure the protocol by using the **client netlogon ping protocol** parameter (default value: **CLADP**).

- The following configuration parameters are removed:

  - **nmbd_proxy_logon**: This setting was used to forward NetLogon authentication requests to a Windows NT4 primary domain controller (PDC) before Samba introduced its own NetBIOS over TCP/IP (NBT) server.

  - **cldap port**: Connectionless Lightweight Directory Access Protocol (CLDAP) always uses UDP port 389. Additionally, the Samba code did not use this parameter consistently, so the behavior was inconsistent.

  - **fruit:posix_rename**: This option of the **vfs_fruit** module is removed because it could result in problems with Windows clients. As a possible workaround to prevent the creation of **.DS_Store** files on network mounts, use the **defaults write com.apple.desktopservices DSDontWriteNetworkStores true** command on MacOS.

Note that the server message block version 1 (SMB1) protocol has been deprecated since Samba 4.11 and will be removed in a future release.

Before starting Samba, back up the database files. Samba automatically updates its **tdb** database files when the **smbd**, **nmbd**, or **winbind** services start. Red Hat does not support downgrading **tdb** database files.

After updating Samba, use the **testparm** utility to verify the **/etc/samba/smb.conf** file.

Jira:RHEL-89870

Identity Management Upgrade Helper

The Identity Management Upgrade Helper is a new application that simplifies upgrading your IdM environment to a newer RHEL version. It provides an upgrade plan with step-by-step instructions that are specific to your upgrade path. As a result, you can use the app to prepare your deployment, set up a new replica, and decommission an old server with clear instructions.

To use this app, see Identity Management Upgrade Helper on the Red Hat Customer Portal.

Jira:RHELDOCS-21103[1]

You can now use dsconf or the web console to exclude subtrees from the attribute uniqueness verification

With this update, you can configure the **uniqueness-exclude-subtrees** parameter for the Attribute Uniqueness plugin directly through the **dsconf** utility and web console. Before this update, **uniqueness-exclude-subtrees** was set only by using the **ldapmodify** utility.

Use the **--exclude-subtree** option for the **dsconf plugin attr-uniq set** command to set the distinguished name (DN) under which the plugin skips uniqueness verification of the attribute's value. Alternatively, go to the **Plugins** menu in the web console, add or edit the Attribute Uniqueness plugin configuration and set the **Excluded Subtrees** field.

Jira:RHEL-67006

389-ds-base rebased to version 3.1.3

The **389-ds-base** package has been updated to version 3.1.3. This version provides various bug fixes and enhancements, most notably:

- Support of Session Tracking Control internet draft
- The **nsslapd-pwdPBKDF2NumIterations** configuration attribute for PBKDF2-* plugins
- Log buffering for the error log
- Support of **CRYPT-YESCRYPT** as a password storage scheme
- JSON format for access and error logs
- Various **dsidm** bug fixes:
    - **dsidm** no longer fails with the **argument must be a string or a number** error.
    - **dsidm get_dn** no longer fails for an organizational unit, service and POSIX group.

- **dsidm uniquegroup members** correctly displays the unique group members.
- **dsidm role rename-by-dn** correctly renames a role.
- **dsidm -j account get-by-dn** and **dsidm -j role get-by-dn** returns the output in JSON format.
- **dsidm role subtree-status** correctly displays a subtree status.
- **dsidm role create-nested** and **dsidm role create-filtered** create nested and filtered roles.
- **dsidm role delete** properly deletes a role.
- **dsidm user rename** renames the user correctly.
- **dsidm account unlock** re-enables user accounts that reached the inactivity limit correctly.

Jira:RHEL-80162

Custom matching rules in the Attribute Uniqueness plugin to search uniqueness attributes

With this update, in Attribute Uniqueness plugin configuration, you can specify a matching rule for the attribute you want to enforce uniqueness on. For example, when you want to override the attribute’s syntax from **case exact** or **case ignore**.

Specify attributes and their matching rules in the plugin configuration, as follows:

```
uniqueness-attribute-name: <attribute>:<Matching rule OID>:
```

Before this update, if you used the attribute **cn** with a **case exact** syntax, the Attribute Uniqueness plugin could not find a matching value if the case was different between the two values being compared. Now you can set the matching rule and make it **case ignore** and the plugin will see that the values match:

```
uniqueness-attribute-name: cn:caseIgnoreMatch:
```

Jira:RHEL-109018[1]

JSON format is available for the access and error logs in **389-ds-base**

With this update, you can use the following commands to configure JSON format for the access and error log files:

```
# dsconf <instance_name> logging access set log-format json
# dsconf <instance_name> logging error set log-format json
```

These commands set the **nsslapd-accesslog-log-format** or **nsslapd-errorlog-json-format** configuration attributes to **json**. As a result, access and error logging becomes more consumable by standard parsing tools.

Note that when you change the format setting, Directory Server rotates the current log file.

Jira:RHEL-80252

The new **list --full-dn** option is available for the **dsidm** utility

With this update, you can use the **list --full-dn** option to get the list of full distinguished names (DN) of the entries of the same type. For example, to see the role DNs, use the following command:

```
# dsidm <instance_name> -b dc=example,dc=com role list --full-dn
```

Before this update, you had no option to determine DNs of these entries with the **dsidm** tool because the existing **list** option only displays relative distinguished name (RDN) values.

Jira:RHEL-74270

389-ds-base log files now contain a session identifier for bind or modify operations

With this enhancement, the replication plugin works with the session tracking feature, correlating replication agreement activities with consumer operations in **389-ds-base**.

On the supplier side, when the replication debug level is enabled, the supplier error log contains messages as follows:

```
[time_stamp] - DEBUG - NSMMReplicationPlugin - repl5_inc_run - "EWBpte8J8Wx 2" -
agmt="cn=004" (localhost:39004): State: wait_for_changes -> ready_to_acquire_replica
```

On the consumer side, without any debug log level, the access logs contain messages as follows:

```
[time_stamp] conn=2 op=7 SRCH base="dc=example,dc=com" scope=2 filter="(objectClass=\*)"
attrs="distinguishedName"
[time_stamp]] conn=2 op=7 RESULT err=0 tag=101 nentries=1 wtime=0.000189515
optime=0.000171470 etime=0.000358345 notes=U,P details="Partially Unindexed Filter,Paged
Search" pr_idx=0 pr_cookie=-1 sid="EWBpte8J8Wx   2"
```

As a result, you can trace the origin of connections or operations more effectively. This improves the overall efficiency and troubleshooting capabilities in connections or operations deployments.

Jira:RHEL-31959[1]

ACME server adds support for the ES256 signature algorithm

Previously, the Automatic Certificate Management Environment (ACME) server did not support the ES256 signature algorithm for JSON Web Key (JWK) validation. This lack of support prevented certain clients, such as the Caddy web server, from successfully obtaining certificates.

With this update, the ACME server has been enhanced to support the ES256 signature algorithm for JWK validation.

As a result, the server can interoperate with clients that use ES256, such as the Caddy web server, allowing them to successfully obtain certificates and establish secure HTTPS communication.

Jira:RHEL-98721[1]

IdM-to-IdM migration now available

IdM-to-IdM migration, previously available as a Technology Preview, is now fully supported with this release. You can use the **ipa-migrate** command to migrate all IdM-specific data, such as SUDO rules, HBAC, DNA ranges, hosts, services, and more, from one IdM server to another. This can be useful, for example, when moving IdM from a development or staging environment into a production one.

Jira:RHELDOCS-19500[1]

HSM is now fully supported in IdM

Hardware Security Modules (HSM) are now fully supported in Identity Management (IdM). You can store your key pairs and certificates for your IdM Certificate Authority (CA) and Key Recovery Authority (KRA) on an HSM. This adds physical security to the private key material.

IdM relies on the networking features of the HSM to share the keys between machines to create replicas. The HSM provides additional security without visibly affecting most IdM operations. When using low-level tools the certificates and keys are handled differently but this is seamless for most users.

> **NOTE**
>
> Migration of an existing CA or KRA to an HSM-based setup is not supported. You need to reinstall the CA or KRA with keys on the HSM.

You need the following:

- A supported HSM.
- The HSM Public-Key Cryptography Standard (PKCS) #11 library.
- An available slot, token, and the token password.

To install a CA or KRA with keys stored on an HSM, you must specify the token name and the path to the PKCS #11 library. For example:

```
ipa-server-install -r EXAMPLE.TEST -U --setup-dns --allow-zone-overlap --no-forwarders -N --auto-reverse --random-serial-numbers -—token-name=HSM-TOKEN --token-library-path=/opt/nfast/toolkits/pkcs11/libcknfast.so --setup-kra
```

Jira:RHELDOCS-17465[1]

6.14. SSSD

Improved smart card authentication for environments with multiple PKCS#11 tokens

SSSD smart card authentication has been enhanced to handle authentication in environments that have multiple PKCS#11 tokens inserted simultaneously. This improves authentication, especially in STIG compliant environments that require multiple user accounts, each with distinct privileges and often tied to a separate PKI token.

Previously, SSSD might fail to authenticate if the first checked token did not contain a matching certificate, because SSSD did not continue searching for the appropriate certificate on other available tokens. With this update, SSSD scans all inserted PKCS#11 tokens for a matching authentication certificate, so that users can authenticate successfully.

Jira:RHEL-4976

The new SSSD option **ldap_read_rootdse** to control RootDSE reads

With this update, SSSD provides a new option, **ldap_read_rootdse**, to control how SSSD reads Root Directory Service Entry (RootDSE) from the LDAP server. By default, SSSD attempts to read the RootDSE anonymously before the user authenticates. However, this default behavior might conflict with strict security policies that typically restrict all anonymous binds to the LDAP server.

To manage this behavior, you can configure the **ldap_read_rootdse** option to **authenticated** to instruct SSSD to read the RootDSE only after a successful user authentication, or set it to **never** to completely prevent SSSD from attempting the read.

Jira:RHEL-13086[1]

6.15. DESKTOP

OpenGL and Vulkan are supported by default in Toolbx containers based on UBI

Before this update, you had to manually install Mesa-related packages to enable OpenGL and Vulkan support, which was not intuitive or documented.

With this enhancement, OpenGL and Vulkan work by default inside Toolbx containers created from updated UBI-based toolbox images, matching the behavior on Red Hat Enterprise Linux Workstation hosts. This includes only the free software drivers provided by Mesa, not proprietary ones such as NVIDIA.

As a result, OpenGL and Vulkan applications can run inside Toolbx containers without additional configuration, improving usability and consistency with the host system.

Jira:RHEL-85074

6.16. THE WEB CONSOLE

cockpit rebased to version 344

The **cockpit** packages have been rebased to version 344, which provides many improvements and fixes compared to version 334 in RHEL 10.0, most notably:

- Improved UI to the new style based on the PatternFly 6 design system.
- Added support for the SMART (Self-Monitoring, Analysis and Reporting Technology) standard and the Stratis 3.8+ pool format in the Storage component.
- Improved graphical VNC, control VNC, and serial consoles in the Virtual machines component.
- Added support for IPv6 addresses for WireGuard VPNs in the Networking component.
- All web console pages can be branded through the **branding.css** style-sheet file.

Jira:RHEL-87394

new subpackage: cockpit-ws-selinux

The SELinux policy for the **cockpit_ws** processes is provided in a separate subpackage **cockpit-ws-selinux**. This prevents the RHEL web console from failing when run on a system without SELinux installed, because the package manager installs the **selinux_policy** packages as dependencies. See the **cockpit_ws_selinux(8)** man page on your system for more information.

Jira:RHEL-92061

6.17. RED HAT ENTERPRISE LINUX SYSTEM ROLES

Introduced a variable MaxRetention to configure the maximum retention parameter

With this update, users can configure the maximum retention parameter for **journald**, enabling time-based deletion of journal files. This enhancement provides flexibility in managing log data according to specific data retention policies, allowing both time-based log deletion and size-based deletion. It helps with compliance with data retention requirements and improves overall system performance by preventing excessive log storage.

Jira:RHEL-102635

metrics role supports enabling additional PCP PMDA

With this update, the **rhel-system-roles** package adds the **metrics_optional_domains** variable to the **metrics** system role. A domain is a set of metrics managed by a Performance Metrics Domain Agent (PMDA), such as a database, specialized hardware, or an application. Use this variable to enable additional PMDAs. The role adds these PMDAs to the default set (for example, the kernel) and the PMDAs that the role manages explicitly (for example, SQL Server databases). As a result, users can enable the domains they require for their specific use cases, improving flexibility in data collection and monitoring.

Jira:RHEL-101724

Ability to configure the default kernel in **rhel-system-roles**

Previously, users could not specify which kernel should be set as the default during system boot. This limitation prevented administrators from managing the default kernel selection through automation.

With this update, the **rhel-system-roles** package introduces the ability to configure the default boot loader kernel by using a new **default** option. Users can now designate a single kernel as the default by setting the **default** boolean parameter in the kernel settings. The system validates that only one kernel can be marked as default, and applies the selection by using **grubby --set-default** as required.

This enhancement improves flexibility and simplifies automation when managing kernel versions in RHEL.

Jira:RHEL-101671[1]

The ad_integration RHEL system role can control the SSSD domain section naming and consolidate duplicates

With this update, users can control the name of the section used in the SSSD config file for the domain or realm-specific settings, as managed by the **ad_dyndns_update** and **ad_integration_sssd_custom_settings** parameters. By default, the **ad_integration** role uses the lower case of the **ad_integration_realm** variable. However if users want to use the actual case of **ad_integration_realm**, users can use a new option **ad_integration_sssd_realm_preserve_case = true** to preserve the case of the realm. This might leave the SSSD config file with multiple sections for the realm. Use the new **ad_integration_sssd_remove_duplicate_sections** setting to consolidate all of the settings from the multiple sections into the chosen section. As a result, the **ad_integration** system role can manage domain and realm sections in the SSSD config file correctly.

Jira:RHEL-99087

The journald RHEL system role can monitor disk space

With this update, you can configure the **SystemKeepFree** option in the **journald.conf** journal service to set a maximum size for the system journal. This improves overall system stability and performance. As a result, you can use the **journald_system_keep_free** variable to configure size limit. The value is specified in megabytes. There is no default value - by default, it will use the **journald** default value.

Jira:RHEL-95846

Introducing flexibility for package installation in ad_integration role

Previously, the **ad_integration** role always attempted to install the required packages, for example, **realmd**, **sssd-ad**, **adcli**, and many more that are listed in **__ad_integration_packages**. In environments where external systems handled package management, for example, via configuration management outside of this role, pre-baked images, or immutable systems, this step was redundant and undesirable.

With this update, users can now manage package installations through other means and only want this role to join a domain, offering them flexibility. The notable enhancements are:

- New Variable: Introduced a new boolean variable **ad_integration_manage_packages** to control whether the role installs packages.
- Default Value: The default value is set to **true** in **defaults/main.yml** to ensure backward compatibility. Existing playbooks using this role will continue to function as before without modification.
- Conditional Task: Added a **when: ad_integration_manage_packages | bool** condition to the "Ensure required packages are installed" task in **tasks/main.yml**. The task will now only run if the flag is **true** (the default).
- Documentation: Updated **README.md** to include the new **ad_integration_manage_packages** variable, explaining its purpose and default value.

Jira:RHEL-88312

The firewall RHEL system role now supports including other services

With this enhancement, you can include other services when you use the **firewall** RHEL system role to create **firewalld** service definitions. For example, you can create a service **webserver** that includes the **http** and **https** services. If you then enable the **webserver** service, **firewalld** open the ports defined in **http** and **https** services. For further details, see Creating a custom firewalld service by using the firewall RHEL system role.

Jira:RHEL-84953[1]

The podman role generates all TOML compliant configuration file

Before this update, the current Jinja-based formatter did not support many TOML features, including tables and inline tables, which were required to configure all aspects of **podman**. With this enhancement, all features of TOML are supported by using a true TOML formatter instead of a simple Jinja template. As a result, the **podman** role can generate any TOML compliant configuration file that **podman** can use.

The **podman** role needs to preserve certain features of the old formatter. Therefore, the TOML formatter is disabled by default. For the particular use cases that you need to use the old formatter for and information about how you can convert your inventory data to use the new and improved formatter, see the README file.

To use the new TOML formatter in all cases, set the **podman_use_new_toml_formatter** to **true**:

```
podman_use_new_toml_formatter: true
```

Jira:RHEL-84932[1]

Metrics role now supports Apache Spark metric collection and export

Previously, users could not directly collect or export Apache Spark metrics by using the metrics role. With this update, the **rhel-system-roles** package adds support to gather and update metrics from Apache Spark. Two new boolean parameters are introduced:

- **metrics_into_spark**: false This enables exporting metric values into Spark.
- **metrics_from_spark**: false This enables gathering metrics from Spark.

You can now both retrieve metrics from Spark and send metrics information into Spark, improving integration and monitoring capabilities for Spark workloads.

Jira:RHEL-78262[1]

Enables IPv4-only operation for the **chronyd** service when using the **rhel-system-roles.timesync** role

With this update, users can customize the **chronyd** configuration on RHEL 10.1 when IPv6 is disabled on a node. The enhancement provides two options: add a setting to the **timesync** role to disable IPv6, or pass a parameter to set the OPTIONS value for **chronyd**. These options enable IPv4-only operation for the **chronyd** service when using the **rhel-system-roles.timesync** role. This improves time synchronization accuracy and stability for environments where IPv6 is disabled.

Jira:RHEL-85689[1]

The **ha_cluster** RHEL System Role can now export resource definitions

Previously, the **ha_cluster** RHEL System Role's export functionality did not include variables related to cluster resources, such as primitives, groups, and clones. This made it difficult to use the role to get a complete, reusable definition of an existing cluster's configuration.

With this enhancement, the export functionality of the **ha_cluster** RHEL System Role has been updated to gather and export cluster resource definitions.

As a result, you can now use the **ha_cluster** RHEL System Role to export a complete cluster configuration that is compatible with the role's input format. The exported data now includes the following variables:

- **ha_cluster_resource_primitives**
- **ha_cluster_resource_groups**
- **ha_cluster_resource_clones**
- **ha_cluster_resource_bundles**

Jira:RHEL-46225

The **ha_cluster** RHEL System Role can now export operating system and **pcsd** configurations

Previously, when using the **ha_cluster** RHEL System Role to export the configuration of an existing cluster, the export did not include important operating system level settings such as repository, firewall, or SELinux configurations. This resulted in an incomplete definition, making it difficult to fully re-create a cluster from the exported data.

With this enhancement, the **ha_cluster** role's export functionality now gathers and exports operating system level and **pcsd** daemon configurations from cluster nodes.

As a result, you can generate a more complete cluster definition from an existing deployment. This is useful for recreating the cluster or for bringing a cluster that was not created with the **ha_cluster** role under its management. The exported data now includes the following variables:

- **ha_cluster_enable_repos**
- **ha_cluster_enable_repos_resilient_storage**
- **ha_cluster_manage_firewall**
- **ha_cluster_manage_selinux**
- **ha_cluster_install_cloud_agents**
- **ha_cluster_pcs_permission_list**

Jira:RHEL-46224

6.18. VIRTUALIZATION

virtio-mem is available on IBM Z

With this update, **virtio-mem**, a paravirtualized memory device, can be used on IBM Z hardware. By using **virtio-mem**, you can dynamically add or remove host memory in virtual machines.

Jira:RHEL-72994[1]

New command for IBM Z hosts: **virsh hypervisor-cpu-models**

This update introduces the **virsh hypervisor-cpu-models** command. You can use this command on the IBM Z architecture to display which CPU models your hypervisor recognizes.

Jira:RHEL-58151[1]

virt-v2v can now convert VMware VMs that use NVMe disks

With this update, the **libvirt** toolset can correctly detect non-volatile memory express (NVMe) disks when analyzing the configuration of virtual machines (VMs) created on the VMware hypervisor. As a result, it is now possible to use the **virt-v2v** utility to convert such VMs for the KVM hypervisor.

Jira:RHEL-7390

Fast initialization NetKVM parameter

This update adds a Fast Initialization (**FastInit**) parameter for NetKVM drivers. Enabling this parameter ensures that the driver allocates only a part of the required memory blocks to virtual queues, and then indicates readiness to the kernel. The remaining memory blocks are then initialized in the background.

This makes starting or restarting the network in Windows virtual machines significantly faster, especially when the network back end uses a high number of virtual queues. However, it might also negatively impact performance before the background memory allocation is finished.

FastInit is enabled by default, but you can disable it by using the Device Manager app in the Windows guest operating system.

Jira:RHEL-40693

virtio-mem can be used with Windows virtual machines

With this update, you can use **virtio-mem**, a paravirtualized memory device, with Windows virtual machines (VMs) running on a RHEL 10 host. With the **virtio-mem** device, you can dynamically add or remove host memory in VMs.

For a list of supported Windows versions, see: Certified Guest Operating Systems .

Jira:RHELDOCS-18640[1]

Performance-enhanced PCI translation for IBM Z guests

With this update, virtual machines (VMs) on IBM Z hosts can use identity-mapped direct memory access (DMA) for PCI devices. This feature significantly improves the performance of PCI device passthrough. Note that to use the feature, your system must be configured as follows:

- The **iommu.passthrough=1** parameter must be set up on the kernel command line of the VM.
- The VM must have fully NUMA-pinned memory.
- The RHEL host system must not be using logical partitioning (LPAR).

Jira:RHEL-52964[1]

virtio based keyboard driver improvements

With this update, the new **virtio** based keyboard driver enables capturing early keyboard input in a virtual machine, especially in firmware setup screens and in GRUB boot loader.

Jira:RHEL-50[1]

New option for VM live migration: --available-switchover-bandwidth

When live-migrating a virtual machine (VM) by using the **virsh migrate --live** command, you can now add the **--available-switchover-bandwidth** option to specify the bandwidth at which the migration switches over to the destination host in the pre-copy process. By default, the hypervisor measures the available bandwidth automatically, but when this might not reliably ensure that the live migration finishes successfully, using **--available-switchover-bandwidth** can fix the issue.

Jira:RHEL-20294

VMs can now use MSDM ACPI tables

On certain Windows guest operating systems, license activation requires the guest to be configured with a Microsoft Data Management (MSDM) Advanced Configuration and Power Interface (ACPI) table. For this purpose, you can now set up a MSDM ACPI table on virtual machines (VMs) hosted on RHEL. To do so, use the following lines in the XML configuration of the VM:

```
<acpi>
  <table type="msdm">/path/to/table</table>
</acpi>
```

Jira:RHEL-81041

Support for disabling the firmware configuration screen for UEFI virtual machines

With this update, a new capability to block the firmware configuration screen for UEFI virtual machines (VMs) has been added to QEMU. Blocking the firmware configuration screen can prevent unauthorized access to the firmware configuration application and ensures the VM boots only from authorized devices.

You can block the firmware configuration screen by adding the following line to the QEMU command line.

```
-fw_cfg name=opt/<hostname>/FirmwareSetupSupport,string=no
```

Replace *<hostname>* with your host ID.

Jira:RHEL-63645[1]

Fine-grained configuration of VM actions on host shutdown

With this update, it is possible to configure the **libvirt** drivers on how to handle virtual machines (VMs) when the host shuts down. For example, you can configure the VM memory to be saved when the host shuts down, and for VMs to be automatically started from the saved memory when the host starts. For the specific configuration options, see the **auto_shutdown** parameters in the **/etc/libvirt/virtqemud.conf** file.

Note that this feature implements the same functionality provided by the **libvirt-guests** service, as configured in the **/etc/sysconfig/libvirt-guests** file. As a consequence, you cannot use **auto_shutdown** configuration in **virtqemud.conf** at the same time as **libvirt-guests.service**.

For new deployments, using **auto_shutdown** in **virtqemud.conf** is recommended instead of **libvirt-guests.service**, and it will replace **libvirt-guests.service** completely in a future major release of RHEL.

Jira:RHEL-71662

New QEMU configuration parameter: migrate_tls_priority

With this update, you can configure the **migrate_tls_priority** parameter in the **/etc/libvirt/qemu.conf** file. You can use this parameter to work around QEMU issues with TLS when live migrating virtual machines. To obtain the recommended value to set if the default does not work on your deployment, contact Red Hat customer support.

Jira:RHEL-104382

New features for virtual machines on 64-bit ARM hosts

The following features are now supported for virtual machines on RHEL hosts that use the 64-bit ARM architecture (**aarch64**):

- Live snapshots
- Pre-copy migration with the following options:

  - TLS encryption and XBZRLE compression
  - Dirty rate monitoring
  - Auto-converge

- Multi-FD migration with the following options:
  - TLS encryption and XBZRLE compression
  - Auto-converge
  - Zero-copy
- Post-copy migration with the following options:
  - TLS encryption and XBZRLE compression
  - Recovery
  - Preemption
- Live migration with **virtiofs**

Jira:RHELDOCS-20674[1]

Direct kernel boot supported for SecureBoot VMs

With this update, you can set up direct kernel boot in virtual machines (VM) that are configured with the SecureBoot feature. To do so, use the **<shim>** parameter in the XML configuration of the VM, for example as follows:

```
<os firmware="efi">
  ...
  <shim>/var/lib/libvirt/images/BOOTX64.EFI</shim>
</os>
```

Jira:RHEL-68043

Support for multiple I/O threads in virtio-scsi devices

With this update, you can configure multiple I/O threads for a single **virtio-scsi** device. To do so, use the **<iothreads>** parameter in the XML configuration of the virtual machine to which the device is attached. This provides additional options for fine-tuning the performance and scalability of your virtual SCSI devices.

Jira:RHEL-77552

SCSI passthrough support for virtual machines

With this update, RHEL now supports SCSI passthrough for virtual machines (VMs). With this feature, VMs can directly access host SCSI devices, such as tape drives and Storage Area Network (SAN) Logical Unit Numbers (LUNs).

As a result, you can configure VMs to use specialized storage devices that require direct SCSI access, including support for both single-path and multipathed vDisks.

Note that for SCSI passthrough to work, the host must use a supported RHEL and kernel version. For details, see Required RHEL versions for SGIO support in Virtual Machines .

Jira:RHELDOCS-21410[1]

SCSI3 Persistent Reservation support for virtual machines

With this update, RHEL supports SCSI3 Persistent Reservation (S3-PR) for virtual machines (VMs). This feature makes it possible for multiple VMs to coordinate access to shared storage devices, which is essential for Linux clustering solutions, such as Pacemaker, and for Windows Server Failover Clustering.

As a result, VMs can register and manage persistent reservations on storage devices, which prevent conflicts when multiple VMs access the same storage. S3-PR support is available for both single-path and multipathed vDisks.

Note that for S3-PR to work, the host must use a supported RHEL and kernel version. For details, see Required RHEL versions for SGIO support in Virtual Machines .

Jira:RHELDOCS-21467[1]

6.19. RHEL IN CLOUD ENVIRONMENTS

Enhanced automatic registration for eligible RHEL images

With this update, RHEL instances based on eligible images from eligible marketplaces automatically receive content and updates from Red Hat content delivery network (CDN) instead of the Red Hat Update Infrastructure (RHUI). The RHUI repositories are turned off by default.

This ensures automatic access to latest updates for users of subscribed RHEL instances.

For additional details, see Understanding auto-registration.

Jira:RHELDOCS-21241[1]

RHEL is available on Azure confidential VMs

You can create and run RHEL confidential virtual machines (CVMs) on Microsoft Azure by using RHEL CVM images. The images support full disk encryption through the Confidential OS disk encryption feature in Azure.

Jira:RHELDOCS-21373[1]

New package: **azure-vm-utils**

This update adds the **azure-vm-utils** package, which provides a collection of utilities and **udev** rules to optimize the experience of using RHEL 10 as a guest operating system on Microsoft Azure.

Jira:RHEL-73904[1]

6.20. SUPPORTABILITY

sos now collects the Satellite metrics file for improved support diagnostics

The **foreman-installer** plugin of **sos** now collects the **satellite_metrics.yml** file located at **/var/lib/foreman-maintain/** directory. It provides insight into which features of Satellite are in use and in what scale.

Jira:RHEL-71825

6.21. CONTAINERS

A new **rhel10/valkey-8** container image is generally available in RHEL

The newly available **rhel10/valkey-8** container image allows atomic operations and supports various data types such as strings, hashes, lists, sets, and sorted sets. The image offers high performance because of its in-memory data set, which can be persisted to disk or by appending commands to a log.

Jira:RHELDOCS-20640[1]

Improved support for reproducible container builds

Reproducible builds ensure that a given set of inputs consistently generates the same output. This enhancement addresses several factors that previously complicated reproducibility in container image builds. While using **-source-date-epoch** and **-rewrite-timestamp** improves the reproducibility of builds and better aligns with common practices such as setting and looking for **$SOURCE_DATE_EPOCH**, it cannot guarantee complete reproducibility.

Jira:RHEL-88522

New artifact endpoints for Podman RESTFUL API

Podman RESTFUL API now includes new artifact endpoints, enabling programmatic management of OCI artifacts. This enhancement simplifies integration of OCI artifact operations into existing systems and scripts.

Jira:RHEL-88473

The Container Tools packages have been updated

The updated Container Tools RPM meta-package, which contains the Podman, Buildah, Skopeo, **crun**, and **runc** tools, is available. The Buildah package has been updated to version v1.41.0, and Skopeo has been updated to version 1.20.0.

Podman release v5.6 contains the following notable bug fixes and enhancements over the previous version:

- A new set of commands for managing Quadlets has been added as **podman quadlet install** (install a new Quadlet for the current user), **podman quadlet list** (list installed Quadlets), **podman quadlet print** (print the contents of a Quadlet file), and **podman quadlet rm** (remove a Quadlet).
- The **podman kube play** command can restrict container execution to specific CPU cores and specific memory nodes using the **io.podman.annotations.cpuset/$ctrname** and **io.podman.annotations.memory-nodes/$ctrname** annotations.
- The **podman kube play** command supports the **lifecycle.stopSignal** field in Pod YAML, allowing the signal used to stop containers to be specified.
- The **podman volume import** and **podman volume export** commands are available in the remote Podman client.
- The **podman volume create** command accepts two new options, **--uid** and **--gid**, to set the UID and GID the volume will be created with.

- The **podman secret create** command has a new option, **--ignore**, causing the command to succeed even if a secret with the given name already exists.
- The **podman pull** command has a new option, **--policy**, to configure pull policy.
- The **podman update** command has a new option, **--latest**, to update the latest container instead of specifying a specific container.
- A full set of API endpoints for interacting with artifacts has been added, including inspecting artifacts (**GET /libpod/artifacts/{name}/json**), listing all artifacts (**GET /libpod/artifacts/json**), pulling an artifact (**POST /libpod/artifacts/pull**), removing an artifact (**DELETE /libpod/artifacts/{name}**), adding an artifact (or appending to an existing artifact) from a tar file in the request body (**POST /libpod/artifacts/add**), pushing an artifact to a registry (**/libpod/artifacts/{name}/push**), and retrieving the contents of an artifact (**GET /libpod/artifacts/{name}/extract**).
- A new command has been added, **podman artifact extract**, to copy some or all of the contents of an OCI artifact to a location on disk.
- The **--mount** option to **podman create**, **podman run**, and **podman pod create** supports a new mount type, **--mount type=artifact**, to mount OCI artifacts into containers.
- The **podman artifact add** command features two new options, **--append** to add new files to an existing artifact, and **--file-type** to specify the MIME type of the file added to the artifact.
- The **podman artifact rm** command features a new option, **--all**, to remove all artifacts in the local store.
- The **podman kube generate** and **podman kube play** commands supports a new annotation, **io.podman.annotation.pids-limit/$containername**, preserving the PID limit for containers across **kube generate** and **kube play**.
- Quadlet **.container** units support three new keys, **Memory=** (set maximum memory for the created container), **ReloadCmd** (execute a command via systemd **ExecReload**), and **ReloadSignal** (stop the container with the given signal via systemd **ExecReload**).
- Quadlet **.container**, **.image**, and **.build** units support two new keys, **Retry** (number of times to retry pulling image on failure) and **RetryDelay** (delay between retries).
- Quadlet **.pod** units support a new key, **HostName=**, to set the pod's hostname.
- Quadlet files support a new option, **UpheldBy**, in the **Install** section, corresponding to the systemd **Upholds** option.
- The names of Quadlet units specified as systemd dependencies are automatically translated, for example **Wants=my.container** is valid.

For more information about notable changes, see upstream release notes.

Jira:RHEL-88463

The ADD and COPY instructions now support the --link option

Buildah and Podman now support the **--link** flag for ADD and COPY instructions in Containerfiles, which causes the new content to be added as its own layer in the built image.

Jira:RHEL-88308

StrictForwardPorts is now available in firewalld

When the **StrictForwardPorts** option in the **/etc/firewalld/firewalld.conf** configuration file is set to **yes**, port forwarding from Podman is no longer possible, and attempting to start a container or pod with the **-p** or **-P** options returns errors. All ports must be forwarded by using **firewalld**. This ensures that containers cannot allow traffic through the firewall without administrator intervention. See the **netavark-firewalld** man page for more details.

Jira:RHEL-27842

New rhel10/nodejs-24 and rhel10/nodejs-24-minimal container images available

The real-time **registry.redhat.io/rhel10/nodejs-24** and **registry.redhat.io/rhel10/nodejs-24-minimal** container images are now available in the Red Hat Container Registry.

Node.js is a platform built on Chrome’s JavaScript runtime for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, ideal for data-intensive real-time applications that run across distributed systems.

Jira:RHELDOCS-20749[1]

RHEL image mode supports creating root-level directories and symlinks at runtime

With this release, you can use RHEL image mode to create root-level directories and symbolic links after system deployment, then return the filesystem to read-only mode. As a result, you can use a single base image across multiple deployment environments with different file system requirements.

Jira:RHELDOCS-21230[1]

bootc-image-builder uses the local container storage by default

With this release, the **bootc-image-builder** tool operates in local mode by default, which means it no longer pulls container images from remote registries. To build disk images, you must pre-load the base bootc container image in the local container registry of the system before building disk images. If you have existing workflows that relied on automatic image pulling, you must update them. This change improves security by reducing external network dependencies during the build process.

Jira:RHELDOCS-21218[1]

6.22. RHEL LIGHTSPEED

The command-line assistant supports image mode for RHEL

With this enhancement, you can customize your Containerfile to include the **command-line-assistant** package, create a disk image from a container image, and boot a system with that image. As a result, the system image has the command-line assistant preinstalled, and you can use it after you register your system with **subscription-manager**.

Jira:RHELDOCS-20546[1]

The command-line assistant context limit increased to 32KB input

Before this update, the command-line assistant had a 2KB input context limit, causing it to fail when input exceeded this limit. As a consequence, user experience was limited, preventing thorough log analysis due to the 2KB input context limit. With this release, the command-line assistant input context limit has been increased from 2KB to 32KB. As a result, the command-line assistant now supports larger input contexts, enabling better log analysis and potential issue detection.

Jira:RHELDOCS-20421[1]

The command-line assistant for RHEL Lightspeed has better error handling and exit codes

With this enhancement, the command-line assistant brings better error handling and exit codes, such as:

- Output different error messages based on different types of errors that can occur during CLA runtime.
- Try to output an error message that corresponds to the actual cause of the error, and log it.
- Implement different exit codes based on different types of issues.

Jira:RHELDOCS-21313[1]

Command-line assistant -w option displays current output

Before this update, when you tried to use the **-w** option without the current enable-capture mode, the command-line assistant incorrectly displayed output from an earlier session. With this update, the terminal capture log file is actively verified before outputting from the **-w** option. As a result, the mentioned problem is fixed, and the displayed output is accurate.

Jira:RHELDOCS-21315[1]

6.23. AI ACCELERATOR DRIVER AVAILABILITY

Accelerator drivers available through Red Hat

With RHEL 10.1, third-party accelerator drivers and compute stacks, for example CUDA from NVIDIA and ROCm from AMD, are directly available to install from Red Hat. The kernel drivers are built and signed within the Red Hat infrastructure and work with secure boot. In addition, a new AppStream component, **rhel-drivers**, eases the installation of these third-party drivers and regular updates are through the existing **dnf** update process.

For instructions about installing AI accelerator drivers on RHEL, see the following Red Hat blog post: The new and simplified AI accelerator driver experience on Red Hat Enterprise Linux .

Jira:RHELDOCS-21377[1]

Simplified third-party driver installation with rhel-drivers

RHEL 10.1 introduces the **rhel-drivers** installer, which is available in the AppStream repository. With this tool, you can more easily install third-party hardware drivers for GPUs and AI accelerators by using a single, uniform command-line interface. The **rhel-drivers** tool manages the installation of complex driver stacks, such as the NVIDIA kernel module and CUDA libraries, by pulling packages directly from the RHEL Extensions and Supplementary channels.

Before this release, installing specialized hardware drivers on RHEL was a manual and inconsistent process. You had to find, download, and manage driver installations from various vendor websites. This approach created significant friction when setting up systems for high-performance computing or AI and machine learning workloads. With **rhel-drivers**, you can more easily, consistently, and reliably install and manage RHEL-distributed partner drivers. This streamlines system provisioning, ensures that you receive the latest supported driver versions directly from Red Hat repositories, and eliminates the need for manual downloads.

For example, you can install all necessary drivers with just two commands:

```
# dnf install rhel-drivers
# rhel-drivers install --auto-detect
```

Jira:RHEL-113198[1]

CHAPTER 7. TECHNOLOGY PREVIEW FEATURES

This part provides a list of all Technology Preview features available in Red Hat Enterprise Linux 10.

For information on Red Hat scope of support for Technology Preview features, see Technology Preview Features Support Scope.

7.1. INSTALLER AND IMAGE CREATION

image-builder-cli replaces **osbuild-composer** and **composer-cli** (Technology Preview)

With this release, you can install and use the new **image-builder-cli** package to build an image with one command. The new tool supports containers and enhances your user experience to create a container image that you can use to build other images. This capability is a Technology Preview feature. For more details, see Installing RHEL image builder .

Jira:RHELDOCS-20354[1]

7.2. SOFTWARE MANAGEMENT

Support for signing packages with Sequoia PGP is available as a Technology Preview

The **macros.rpmsign-sequoia** macro file that configures RPM to use Sequoia PGP instead of GnuPG for signing packages is now available as a Technology Preview. To enable its usage, perform the following steps:

1. Install the following packages:

```
# dnf install rpm-sign sequoia-sq
```

2. Copy the **macros.rpmsign-sequoia** file to the **/etc/rpm/** directory:

```
$ cp /usr/share/doc/rpm/macros.rpmsign-sequoia /etc/rpm/
```

Jira:RHEL-56363[1]

7.3. SHELLS AND COMMAND-LINE TOOLS

RHEL 10.1 provides ReaR on aarch64 as a Technology Preview

RHEL 10.1 introduces the Relax and Recover (ReaR) package for the 64-bit ARM architecture (**aarch64**) as a Technology Preview. ReaR is a disaster recovery tool that produces a bootable image that you can use to restore the system from a backup. You can currently use the following output methods with ReaR on **aarch64**: ISO, USB, and PXE.

For more information about ReaR, see the article What is Relax and Recover(ReaR) and how to use it for disaster recovery?

Jira:RHEL-84286[1]

7.4. KERNEL

The Red Hat Enterprise Linux for Real Time on ARM64 is now available as a Technology Preview

With this Technology Preview, the Red Hat Enterprise Linux for Real Time is now enabled for ARM64. The ARM64 is enabled on ARM (AARCH64), for both 4k and 64k ARM kernels.

Jira:RHELDOCS-19635[1]

7.5. FILE SYSTEMS AND STORAGE

ublk_drv driver is available as a Technology Preview

The **ublk_drv** kernel module is now enabled as a Technology Preview. It provides the **ublk** framework with which you can create and build high-performance block devices from userspace. Currently, **ublk** requires userspace implementations, such as the Userspace Block Driver (**ublksrv**) or the Rust-based **ublk** (**rublk**), to function effectively.

Jira:RHELDOCS-19891[1]

NVMe/TCP using TLS is available as a Technology Preview

Encrypting Non-volatile Memory Express (NVMe) over TCP (NVMe/TCP) network traffic using TLS configured with Pre-Shared Keys (PSK) has been added as a Technology Preview in RHEL 10.0. For instructions, see Configuring an NVMe/TCP host using TLS with Pre-Shared-Keys .

Jira:RHELDOCS-19968[1]

xfs_scrub utility is available as a Technology Preview

You can check all the metadata on a mounted XFS file system by using the **xfs_scrub** utility as a Technology Preview. It functions similarly to the **xfs_repair -n** command for an unmounted XFS filesystem. For details, see the **xfs_scrub(8)** man page on your system. Note that currently only the scrub feature is available in RHEL 10 kernels and online repair is not enabled.

Jira:RHELDOCS-20041[1]

Limited shrinking of XFS file systems is available as Technology Preview

You can reduce the size of XFS file systems by using the **xfs_growfs** utility as a Technology Preview. You can remove blocks from the end of the file system by using **xfs_growfs**, provided that all of the following conditions are true:

- No metadata or data is allocated within the range to be removed.
- The requested size is within the last allocation group.

Jira:RHELDOCS-20042[1]

Mounting XFS file systems with blocks larger than system page is available as Technology Preview

You can now mount XFS file systems created with a block size larger than the system page size as a Technology Preview. For example, a file system with 16-KB blocks can now be mounted on a system with a 4-KB page size, such as x86_64.

Jira:RHELDOCS-20043[1]

io-uring interface is available as a Technology Preview

The **io_uring**, which is an asynchronous I/O interface, is available as a Technology Preview. By default, this feature is disabled in RHEL 10. You can enable this interface by setting the **kernel/io_uring_disabled** variable:

- For all users:

```
# echo 0 > /proc/sys/kernel/io_uring_disabled
```

- For root only:

```
# echo 1 > /proc/sys/kernel/io_uring_disabled
```

You can also disable **io_uring** for all processes:

```
# echo 2 > /proc/sys/kernel/io_uring_disabled
```

Jira:RHEL-65347

NVMe/TCP Boot with NBFT is available as a Technology Preview

NVMe/TCP Boot by using the NVM Express Boot Specification (NBFT) is available on select server platforms as a Technology Preview. Consult your server manufacturer for platform-specific details and compatibility information.

Jira:RHELDOCS-21587[1]

7.6. DYNAMIC PROGRAMMING LANGUAGES, WEB AND DATABASE SERVERS

Node.js 24 is available as a Technology Preview

A new **nodejs24** component is available as a Technology Preview in Red Hat Enterprise Linux 10.1. This update introduces Node.js 24, which includes new features, bug fixes, security updates, and performance improvements compared to Node.js 22 in RHEL 10.0.

Currently, the **nodejs24** package provides versioned binaries (**/usr/bin/node-24**, **/usr/bin/npm-24**, and **/usr/bin/npx-24**). To use these binaries, update the shebang lines in your scripts to reference the version-specific paths. The ability for **nodejs24** to provide the base binaries (**/usr/bin/node** and related files) might be included in a future update.

To install the **nodejs24** package, enter:

```
# dnf install nodejs24
```

For information about the length of support for the **nodejs** Application Streams, see Red Hat Enterprise Linux Application Streams Life Cycle.

Jira:RHEL-90826

7.7. COMPILERS AND DEVELOPMENT TOOLS

eu-stacktrace available as a Technology Preview

The **eu-stacktrace** utility, which has been distributed through the **elfutils** package since version 0.192, is available as a Technology Preview feature. **eu-stacktrace** is a prototype utility that uses the **elfutils** toolkit's unwinding libraries to support a sampling profiler to unwind frame pointer-less stack sample data.

Jira:RHELDOCS-19072[1]

7.8. IDENTITY MANAGEMENT

DNS over TLS (DoT) in IdM deployments is available as a Technology Preview

Encrypted DNS using DNS over TLS (DoT) is now available as a Technology Preview in Identity Management (IdM) deployments. You can now encrypt all DNS queries and responses between DNS clients and IdM DNS servers.

To start using this functionality, install the **ipa-server-encrypted-dns** package on IdM servers and replicas, and the **ipa-client-encrypted-dns** package on IdM clients. Administrators can enable DoT during the installation by using the **--dns-over-tls** option.

IdM configures Unbound as a local caching resolver and BIND to receive DoT requests. This functionality is available through the command-line interface (CLI) and noninteractive installations of IdM.

The following options were added to installation utilities for IdM servers, replicas, clients, and the integrated DNS service:

- **--dot-forwarder** to specify an upstream DoT-enabled DNS server.
- **--dns-over-tls-key** and **--dns-over-tls-cert** to configure DoT certificates.
- **--dns-policy** to set a DNS security policy to either allow fallback to unencrypted DNS or enforce strict DoT usage.

By default, IdM uses the **relaxed** DNS policy, which allows fallback to unencrypted DNS. You can enforce encrypted-only communication by using the new **--dns-policy** option with the **enforced** setting.

You can also enable DoT on an existing IdM deployment by reconfiguring the integrated DNS service by using **ipa-dns-install** with the new DoT options.

See Securing DNS with DoT in IdM for more details.

Jira:RHEL-67912

DNSSEC available as Technology Preview in IdM

Identity Management (IdM) servers with integrated DNS now implement DNS Security Extensions (DNSSEC), a set of extensions to DNS that enhance security of the DNS protocol. DNS zones hosted on IdM servers can be automatically signed using DNSSEC. The cryptographic keys are automatically generated and rotated.

Users who decide to secure their DNS zones with DNSSEC are advised to read and follow these documents:

- DNSSEC Operational Practices, Version 2

- Secure Domain Name System (DNS) Deployment Guide
- DNSSEC Key Rollover Timing Considerations

Note that IdM servers with integrated DNS use DNSSEC to validate DNS answers obtained from other DNS servers. This might affect the availability of DNS zones that are not configured in accordance with recommended naming practices.

Jira:RHELDOCS-20690[1]

Encrypted DNS with DoT is now available in ansible-freeipa installations of IdM as a Technology Preview

You can now use Ansible to ensure that all DNS queries and responses between DNS clients and Identity Management (IdM) DNS servers are encrypted. Encrypted DNS using DNS over TLS (DoT) has been available as a Technology Preview in IdM deployments since RHEL 10. In RHEL 10.1, the functionality is available as a Technology Preview in the **freeipa.ansible_freeipa** collection.

To enable DoT during a deployment of IdM by using **ansible-freeipa** use the following options:

- **ipaserver_dns_over_tls** with the **freeipa.ansible_freeipa.ipaserver** role for a new server.
- **ipareplica_dns_over_tls** with the **freeipa.ansible_freeipa.ipareplica** role for a replica.
- **dot_forwarder** to specify an upstream DoT-enabled DNS server.
- **dns_over_tls_key** and **dns_over_tls_cert** to configure DoT certificates.

Additionally, you can set the **dns_policy** variable to enforce DoT-only communication, overriding the default behavior that allows fallback to unencrypted DNS.

Jira:RHELDOCS-20258[1]

7.9. VIRTUALIZATION

TDX is available on RHEL hosts as a Technology Preview

As a Technology Preview, you can enable Trust Domain Extensions (TDX) on RHEL hosts. TDX is a hardware-based security feature that provides strong memory encryption and integrity protection for virtual machines, isolating them from the hypervisor and other system software.

TDX is available only with Intel CPUs.

Jira:RHEL-111863[1]

VDUSE for RHEL networking is available as a Technology Preview

The virtio Data Path Acceleration (vDPA) device in userspace (VDUSE) feature is now available as a Technology Preview for RHEL networking. VDUSE is a Linux kernel mechanism, which allocates user-space for vDPA devices specifically. This mechanism enables a user-space process to register a **virtio-class** device, such as a NIC or block device, with the kernel in a controlled manner. As a result, you can use it on virtual machines or the host through standard vDPA or virtio interfaces.

Jira:RHEL-76477[1]

AMD SEV, SEV-ES, and SEV-SNP for KVM virtual machines are available as a Technology Preview

As a Technology Preview, RHEL provides the Secure Encrypted Virtualization (SEV) feature for AMD EPYC host machines that use the KVM hypervisor. If enabled on a virtual machine (VM), SEV encrypts the VM's memory to protect the VM from access by the host. This increases the VM security.

In addition, the enhanced Encrypted State version of SEV (SEV-ES) is also provided as Technology Preview. SEV-ES encrypts all CPU register contents when a VM stops running. This prevents the host from modifying the VM's CPU registers or reading any information from them.

RHEL also provides the Secure Nested Paging (SEV-SNP) feature as Technology Preview. SNP enhances SEV and SEV-ES by improving its memory integrity protection, which helps to prevent hypervisor-based attacks, such as data replay or memory re-mapping.

Note that:

- SEV and SEV-ES work only on the 2nd generation of AMD EPYC CPUs (codenamed Rome) or later.
- SEV-SNP works only on 3rd generation AMD EPYC CPUs (codenamed Milan) or later.

Also note that RHEL includes SEV, SEV-ES, and SEV-SNP encryption, but not the SEV, SEV-ES, and SEV-SNP security attestation and live migration.

Jira:RHELDOCS-16800[1]

Creating nested virtual machines

Nested KVM virtualization is provided as a Technology Preview for KVM virtual machines (VMs) running on Intel, AMD64, and IBM Z hosts with RHEL 10. With this feature, a RHEL 7, RHEL 8, or RHEL 9 VM that runs on a physical RHEL 10 host can act as a hypervisor, and host its own VMs.

Jira:RHELDOCS-20080[1]

New package: **trustee-guest-components**

As a Technology Preview, this update adds the **trustee-guest-components** package. This makes it possible for confidential virtual machines to attest themselves and get confidential resources from a Trustee server.

Jira:RHEL-73770[1]

Virtual Socket to TCP bridge is available as a Technology Preview

As a Technology Preview, you can use a Virtual Socket (**vsock**) to TCP bridge. By using this bridge, you can securely expose a virtual machine (VM) service, such as SSH, to the host machine without configuring any IP networking.

To bridge your host's connection directly to the SSH service inside the VM over the hypervisor's private **vsock** channel, you can use a relay tool such as **socat**.

Jira:RHEL-91041

SEV-SNP is available on RHEL hosts as a Technology Preview

As a Technology Preview, you can enable Secure Encrypted Virtualization-Secure Nested Paging (SEV-SNP) on RHEL hosts. SEV-SNP is a hardware-based security feature that provides strong memory encryption and integrity protection for virtual machines, isolating them from the hypervisor and other system software.

SEV-SNP is available only with AMD CPUs, and you must use the **snphost** package to configure the feature on the host.

Jira:RHELDOCS-19757[1]

CCA in ARM virtual machines is available as a Technology Preview

As a Technology Preview, you can enable Confidential Compute Architecture (CCA) in RHEL 10.1 virtual machines (VMs). CCA, built on top of Realm Management Extension (RME), helps to maintain data privacy while it is in use within a virtual machine.

Currently, CCA can only be enabled in ARM VMs as a Technology Preview and not in a RHEL host.

Jira:RHEL-83042

7.10. CONTAINERS

Partial pulls for **zstd:chunked** are available as a Technology Preview

You can pull only the changed parts of the container images compressed with the **zstd:chunked** format, reducing network traffic and necessary storage. You can enable partial pulls by adding the **enable_partial_images = "true"** setting to the **/etc/containers/storage.conf** file. This functionality is available as a Technology Preview.

Jira:RHEL-32266

The **podman artifact** command is available as a Technology Preview

The **podman artifact** command, which you can use to work with OCI artifacts at the command-line level, is available as a Technology Preview. For further informal, reference the man page.

Jira:RHEL-70218

The **vrf** option for the **podman network create** is available as a Technology Preview

The **podman network create** command now provides the **vrf** value for the **--opt** option, as a Technology Preview. The **vrf** value assigns a virtual routing and forwarding instance (VRF) to the bridge interface. It accepts the name of the VRF and defaults to none.

This option can only be used with the Netavark network backend.

Jira:RHEL-89373

Podman compatibility with Docker API is available as a Technology Preview

Podman supports the following Docker API versions as a Technology Preview:

- Docker API 1.41
- Docker API 1.43

Jira:RHEL-88122

7.11. TECHNOLOGY PREVIEW FEATURES IDENTIFIED IN PREVIOUS RELEASES

This part provides a list of all Technology Previews available in Red Hat Enterprise Linux 10.

For information on Red Hat scope of support for Technology Preview features, see Technology Preview Features Support Scope.

7.11.1. Networking

WireGuard VPN is available as a Technology Preview

WireGuard, which Red Hat provides as an unsupported Technology Preview, is a high-performance VPN solution that runs in the Linux kernel. It uses modern cryptography and is easier to configure than other VPN solutions. Additionally, the small code-basis of WireGuard reduces the surface for attacks and, therefore, improves the security.

For further details, see Setting up a WireGuard VPN .

Jira:RHELDOCS-20056[1]

KTLS available as a Technology Preview

In RHEL, Kernel Transport Layer Security (KTLS) is provided as a Technology Preview. KTLS handles TLS records by using the symmetric encryption or decryption algorithms in the kernel for the AES-GCM cipher. KTLS also includes the interface for offloading TLS record encryption to Network Interface Controllers (NICs) that provides this functionality.

Note that specific uses cases of kernel TLS offload might have a higher support status.

Jira:RHELDOCS-20440[1]

The PRP and HSR protocols are now available as a Technology Preview

This update adds the **hsr** kernel module that provides the following protocols:

- Parallel Redundancy Protocol (PRP)
- High-availability Seamless Redundancy (HSR)

The IEC 62439-3 standard defines these protocols, and you can use this feature to configure redundancy with zero-time recovery in Ethernet networks.

Jira:RHELDOCS-20472[1]

NetworkManager enables configuring HSR and PRP interfaces

High-availability Seamless Redundancy (HSR) and Parallel Redundancy Protocol (PRP) are network protocols that provide seamless failover against failure of any single network component. Both protocols are transparent to the application layer, meaning that users do not experience any disruption in communication or any loss of data, because a switch between the main path and the redundant path happens very quickly and without awareness of the user. Now it is possible to enable and configure HSR and PRP interfaces using the **NetworkManager** service through the **nmcli** utility and the DBus message system.

Jira:RHEL-5852

CHAPTER 8. DEVELOPER PREVIEW FEATURES

This part provides a list of all Developer Preview features available in Red Hat Enterprise Linux 10.1.

For information about Red Hat scope of support for Developer Preview features, see Developer Preview - Scope of Support.

8.1. RHEL LIGHTSPEED

The MCP server for Red Hat Enterprise Linux is available as a Developer Preview

This Developer Preview introduces a new Model Context Protocol (MCP) server for RHEL, which is designed to bridge the gap between RHEL systems and Large Language Models (LLMs). By using the MCP server for RHEL, you can enable AI applications to perform context-aware troubleshooting on RHEL systems, including log and performance analysis.

Jira:RHELDOCS-21153[1]

CHAPTER 9. REMOVED FEATURES

All removed features were deprecated in earlier releases and are no longer supported. For information regarding functionality that is present in RHEL 9 but has been *removed* in RHEL 10, see Considerations in adopting RHEL 10.

9.1. INSTALLER AND IMAGE CREATION

The cockpit-composer package is removed

The **cockpit-composer** package is removed and is no longer supported. From now on, use **cockpit-image-builder**.

Jira:RHELDOCS-20167[1]

gdisk is removed from boot.iso in RHEL 10

The **gdisk** partitioning utility is removed from the **boot.iso** image type in RHEL 10. You still can use **gdisk** in your Kickstarts. For the **boot.iso** image type, other tools are available for handling GPT disks, for example, the **parted** utility.

Jira:RHELDOCS-18904[1]

9.2. NETWORKING

Network team driver was removed

The **teamd** service and the **libteam** library were removed in Red Hat Enterprise Linux 10. As a replacement, configure a bond instead of a network team.

Red Hat focuses its efforts on kernel-based bonding to avoid maintaining two features, bonds and teams, that have similar functions. The bonding code has a high customer adoption, is robust, and has an active community development. As a result, the bonding code receives enhancements and updates.

If you use RHEL 9 with a network team and plan to upgrade to RHEL 10, migrate the network team configuration to network bond before you upgrade.

Jira:RHELDOCS-20862[1]

9.3. COMPILERS AND DEVELOPMENT TOOLS

32-bit ARM and MIPS backends are removed from llvm

The **llvm** package in RHEL 10.1 removed the 32-bit ARM and MIPS backends. This change reduces build time and maintenance effort for the toolchain. If you require these backends, you should use alternative build targets or earlier package versions.

Jira:RHEL-86089

9.4. IDENTITY MANAGEMENT

The referral mode is removed from 389-ds-base

Before this update, Directory Server supported starting the server in the referral mode. However, because of the stability issues, this feature is no longer supported and was removed.

Note that you can continue using the **nsslapd-referralmode** and **nsslapd-referral** attributes. Directory Server still can return referrals when the requested distinguished name (DN) is not in any of the suffixes the server maintains.

Jira:RHEL-35241

nsslapd-subtree-rename-switch is removed from **389-ds-base**

Before this update, you could configure Directory Server to prevent moving entries between sub-trees in a database. Because of the stability issues, this feature is removed. Consequently, the **nsslapd-subtree-rename-switch** parameter no longer exists.

As a result, you can no longer deactivate moving the entries between sub-trees. As an alternative, you can deactivate moving the entries by creating an access control instruction (ACI).

Jira:RHEL-77490[1]

9.5. RED HAT ENTERPRISE LINUX SYSTEM ROLES

The **mssql_accept_microsoft_odbc_driver_17_for_sql_server_eula** variable has been deprecated

With a future major update of RHEL, the **mssql_accept_microsoft_odbc_driver_17_for_sql_server_eula** variable will no longer be supported in the **mssql** system role because the role can now install the **odbc** driver for **mssql_tools** version 17 and 18. Therefore, you must use the **mssql_accept_microsoft_odbc_driver_for_sql_server_eula** variable without the version number instead.

Important: If you use the deprecated variable with the version number **mssql_accept_microsoft_odbc_driver_17_for_sql_server_eula**, the role notifies you to use the new variable **mssql_accept_microsoft_odbc_driver_for_sql_server_eula**. However, the deprecated variable continues to work.

Jira:RHEL-69315

9.6. VIRTUALIZATION

Live VM dumps are no longer supported

With this update, the **-- live** option for the **virsh dump** command has become unsupported. If you attempt to create a virtual machine dump by using **virsh dump** with the **--live** option, the command will fail.

Jira:RHELDOCS-21349[1]

CHAPTER 10. DEPRECATED FEATURES

Deprecated functionalities are fully supported, which means that they are tested and maintained, and their support status remains unchanged within Red Hat Enterprise Linux 10. However, they will likely not be supported in a future major version release, and are not recommended for new deployments on the current or future major versions of Red Hat Enterprise Linux.

Features can be deprecated during a major version’s release cycle.

A deprecated feature is listed in all future release notes until it is removed. For a complete list of deprecated features, see the release notes for the latest minor version. For information about the length of support, see Red Hat Enterprise Linux Life Cycle and Red Hat Enterprise Linux Application Streams Life Cycle.

10.1. INSTALLER AND IMAGE CREATION

The squashfs package has been deprecated

The **squashfs** package has been deprecated, and will be removed in a future major RHEL release. As an alternative, **dracut** has support for mounting **erofs**.

Jira:RHELDOCS-18903[1]

The module Kickstart command has been deprecated

Anaconda has deprecated its support for DNF modularity, and as a consequence the **module** Kickstart command has been deprecated. This might impact you if you are using modules in the **%packages** section of your Kickstart files or the **module** Kickstart command. This change is implemented for simplifying the installation process and ensuring a more consistent experience moving forward.

Jira:RHEL-34829

The inst.gpt boot option is now deprecated

The **inst.gpt** boot option is now deprecated and will be removed in the future releases. To specify a preferred disk label type, use the **inst.disklabel** boot option. Specify **gpt** or **mbr** to create GPT or MBR disk labels.

Jira:RHELDOCS-18491[1]

10.2. SECURITY

oqsprovider and liboqs are deprecated

The **oqsprovider** and **liboqs** packages, which provided post-quantum cryptography (PQC) for OpenSSL 3.0, are deprecated and might be removed in a future major release. Instead, use the PQC functionality provided by OpenSSL 3.5.

Jira:RHEL-97489[1]

X25519-MLKEM768 deprecated and aliased to MLKEM768-X25519 in crypto-policies

The **X25519-MLKEM768** value in system-wide cryptographic policies is deprecated and aliased to the **MLKEM768-X25519** value. This unifies the concatenation order, allowing both variants to work.

Jira:RHEL-99813

ENGINE API in OpenSSL is deprecated

In RHEL 10, ENGINE API is deprecated and is planned to be removed in a future major release. No new applications should be built by using the ENGINE API. To keep application binary interface (ABI) and existing applications working, OpenSSL still exports the ENGINE symbols. To prevent new applications from using ENGINE API, OpenSSL sets the **OPENSSL_NO_ENGINE** flag system-wide, and the header **engine.h** that exposes the ENGINE API has been removed.

Jira:RHEL-45704

crypto-policies now set **allow-rsa-pkcs1-encrypt = false** for GnuTLS

In RHEL 10, the GnuTLS library blocks encryption and decryption with the RSA PKCS #1 v1.5 padding by default. Except for the LEGACY policy, the **allow-rsa-pkcs1-encrypt = false** option is specified in all system-wide cryptographic policies (DEFAULT, FUTURE, and FIPS).

Jira:RHEL-64746

HMAC-SHA-1 in FIPS mode is deprecated

The HMAC-SHA-1 cryptographic algorithm is deprecated in FIPS mode, and it might be removed in a future release. Outside FIPS mode, support for HMAC-SHA-1 is preserved.

Jira:RHELDOCS-18674

10.3. SOFTWARE MANAGEMENT

Modularity is deprecated

In RHEL 10, the modularity functionality is deprecated and will be removed in a future major release. Therefore, the DNF **module** command displays a deprecation warning.

NOTE

In previous RHEL major versions, some Application Streams were available as modules as an extension to the RPM format. In RHEL 10, Red Hat does not intend to provide any Application Streams that use modularity as the packaging technology. Therefore, no modular content is being distributed with RHEL 10.

Jira:RHELDOCS-20138[1]

10.4. INFRASTRUCTURE SERVICES

FTP clients and Servers software are now deprecated

The following FTP clients and servers software are deprecated and will be removed in the future major version of RHEL:

- **ftp**
- **lftp**
- **vsftpd**

These FTP protocol implementations are no longer under active development. We recommend that customers plan to migrate workflows based on FTP to one of either:

- OpenSSH and the **sftp** command, which provides an interactive interface for secure file transfer over the SSH protocol.
- WebDAV based on Apache httpd - various client implementations are available.

Jira:RHELDOCS-20610[1]

10.5. NETWORKING

ipset has been unmaintained

In RHEL 10, the **ipset** utility is unmaintained and is planned to be removed in a future major release. Red Hat will provide only critical bug fixes during the current release lifecycle. As an alternative to **ipset**, you can use the **nftables** sets functionality instead.

Jira:RHELDOCS-20147[1]

The BIND auto-dnssec parameter is deprecated

Starting with RHEL 9.7, the BIND **auto-dnssec** parameter is deprecated and will be removed in a future release. As a replacement, use the **dnssec-policy** parameter to specify a complete Key and Signing Policy (KASP) that groups all related configurations into a single, intuitive block.

For further details and information about migrating to **dnssec-policy**, see DNSSEC Key and Signing Policy in the BIND 9 upstream documentation.

Jira:RHELDOCS-21532[1]

10.6. FILE SYSTEMS AND STORAGE

The squashfs package has been deprecated

SquashFS is deprecated and will be removed in the next major release. It will no longer receive enhancements and is in RHEL 10 for specific use cases that are internal to Red Hat. Consider using EROFS as an alternative solution.

Jira:RHELDOCS-18450[1]

10.7. HIGH AVAILABILITY AND CLUSTERS

Deprecated High Availability Add-On features

The following features have been deprecated in Red Hat Enterprise Linux 10 and will be removed in the next major release:

- Specifying rules as multiple arguments. Use a single string argument instead.
- Specifying **score** as a standalone value in **pcs constraint location add** and **pcs constraint colocation ad**. Use **score=value** instead.

- Specifying the **--wait** option in resource commands except **pcs resource restart | move**, and in the commands **pcs cluster node add-guest | add-remote**. Use the following commands instead:
  - **pcs status wait** to wait for the cluster to settle into stable state.
  - **pcs status query resource** commands to verify that the resource is in the expected state after the wait.
- Using the **--force** flag to confirm potentially destructive actions such as **pcs cluster destroy**, **pcs quorum unblock**, **pcs stonith confirm**, **pcs stonith sbd device setup**, and **pcs stonith sbd watchdog test** commands. You should now use the **--yes** flag to confirm potentially destructive actions and reserve use of the **--force** flag to override validation errors.
- Using the **--force** flag to confirm overwriting files in **pcs cluster report**. Use the **--overwrite** flag instead.
- Assigning and unassigning ACL roles without specifying the **user** or **group** keyword.
- Configuring a score parameter in order constraints. The **pcs** command-line interface now produces a warning when a user attempts to configure a score parameter in order constraints.

Jira:RHELDOCS-19607[1]

10.8. COMPILERS AND DEVELOPMENT TOOLS

GCC Toolset 15 environment script replaces Software Collections (scl-enable)

Previously, the **scl enable gcc-toolset-15 <command>** command was used to manage the development environment for GCC Toolset 15 on Red Hat Enterprise Linux. In RHEL 10, Software Collections are no longer used for this purpose. As a consequence, the **scl enable** option does not work with **gcc-toolset-15**.

Use the new **gcc-toolset-15-env** script, which runs the specified command with the GCC toolset environment:

```
gcc-toolset-15-env <command>
```

If a command is not specified, the script opens a default shell (**sh**) in the GCC toolset environment.

As a result, users must use **gcc-toolset-15-env** instead of **scl enable** to access GCC Toolset 15 in RHEL 10.

Jira:RHEL-88743[1]

The utmp and utmpx interfaces in glibc are deprecated

The **utmp** and **utmpx** interfaces provided by the **glibc** library include a counter that counts time since the UNIX epoch. This counter will overflow on February 07, 2106. Therefore, **utmp** and **utmpx** are deprecated in RHEL 10 and will be removed in RHEL 11.

Jira:RHELDOCS-18080[1]

10.9. THE WEB CONSOLE

The host switcher in the RHEL web console is deprecated

The host switcher that provides connections to multiple machines through SSH from a single RHEL web console session is deprecated and disabled by default. Due to the web technology limitations, this feature cannot be secure.

In the short term, you can enable the host switcher after assessing the risks in your scenario with the **AllowMultiHost** option in the **cockpit.conf** file:

```
[WebService]
AllowMultiHost=yes
```

As more secure alternatives, you can use:

- the web console login page (with the secure limit of one host in a web browser session)
- the Cockpit Client flatpak

Jira:RHEL-4032[1]

10.10. RED HAT ENTERPRISE LINUX SYSTEM ROLES

The sshd variable deprecated and replaced by sshd_config

To unify coding standards across the RHEL system roles, the **sshd** variable has been replaced by the **sshd_config** variable. The **sshd** variable is now deprecated and might be removed from the **sshd** Ansible role in a future major version of RHEL.

Jira:RHEL-73440[1]

10.11. VIRTUALIZATION

Specific IBM z16 CPU features have been deprecated.

With this update, the **te** and **cte** CPU features have been deprecated for IBM z16 KVM VMs. Note, however, that migrating a virtual machine with CPU model **host-model** from an IBM z16 host to an IBM z17 host does not require any adjustments to CPU feature settings.

Jira:RHEL-89426[1]

The rtl8139 NIC has been deprecated for VMs

With this update, the **rtl8139** network interface controller type has been deprecated, and will become unsupported for use in virtual machines in a future major release of RHEL. If you require using a non-virtio NIC type on your host, use the **e1000** or **e1000e** NIC instead.

Jira:RHEL-45624

libslirp has been deprecated

In RHEL 10, the **libslirp** networking back end has become deprecated, and will be removed in a future major version release.

Jira:RHEL-45147

The i440fx virtual machine type has been deprecated

In RHEL 10, the **i440fx** machine types for virtual machines (VMs) have become deprecated, and will be removed in a future major version of RHEL.

In addition, the **i440fx-rhel7.6** machine type has been replaced by **i440fx-rhel10.0**. As a consequence, a VM with a **i440fx-rhel7.6** machine type will not boot correctly after live migrating to a RHEL 10 host. Workaround: Restart the VM after live migration.

Jira:RHELDOCS-18672[1]

Legacy vCPU models are now deprecated

Several virtual CPU models are now deprecated and will become unsupported for use in virtual machines (VMs) in a future major release of RHEL. Notably, the deprecated models include the following:

- Intel Xeon 55xx and 75xx Processor families (also known as Nehalem)
- Intel Xeon v2 (also known as Ivy Bridge)
- AMD Opteron G4 and G5

To view the complete list of deprecated CPU models, use the following command:

```
# /usr/libexec/qemu-kvm -cpu help | grep depre | grep -v - -v
```

To check whether a running VM is using a deprecated CPU model, use the **virsh dominfo** utility, and look for a line similar to the following in the **Messages** section:

```
tainted: use of deprecated configuration settings
deprecated configuration: CPU model 'Nehalem'
```

Jira:RHEL-28971[1]

virt-manager has been deprecated

The Virtual Machine Manager application, also known as **virt-manager**, has been deprecated. The RHEL web console, also known as **Cockpit**, is intended to become its replacement in a subsequent release. It is, therefore, recommended that you use the web console for managing virtualization in a GUI. Note, however, that some features available in **virt-manager** might not be yet available in the RHEL web console.

Jira:RHELDOCS-20688[1]

libvirtd has become deprecated

The monolithic **libvirt** daemon, **libvirtd**, has been deprecated in RHEL 9, and will be removed in a future major release of RHEL. Note that you can still use **libvirtd** for managing virtualization on your hypervisor, but Red Hat recommends switching to the newly introduced modular **libvirt** daemons. For instructions and details, see the RHEL 9 Configuring and Managing Virtualization document.

Jira:RHELDOCS-20689[1]

SecureBoot image verification by using SHA1-based signatures is deprecated

Performing SecureBoot image verification by using SHA1-based signatures on UEFI (PE/COFF) executables has become deprecated. Instead, Red Hat recommends using signatures based on the SHA-2 algorithm, or later.

Jira:RHELDOCS-20691[1]

The virtual floppy driver has become deprecated

The **isa-fdc** driver, which controls virtual floppy disk devices, is now deprecated, and will become unsupported in a future release of RHEL. Therefore, to ensure forward compatibility with migrated virtual machines (VMs), Red Hat discourages using floppy disk devices in VMs hosted on RHEL 10.1.

Jira:RHELDOCS-20692[1]

qcow2-v2 image format is deprecated

With RHEL 10.1, the qcow2-v2 format for virtual disk images has become deprecated, and will become unsupported in a future major release of RHEL. In addition, the RHEL 10.1 Image Builder cannot create disk images in the qcow2-v2 format.

Instead of qcow2-v2, Red Hat strongly recommends using qcow2-v3. To convert a qcow2-v2 image to a later format version, use the **qemu-img amend** command.

Jira:RHELDOCS-20693[1]

10.12. CONTAINERS

tzdata package is no longer installed by default in the minimal container images

The **tzdata** package is no longer installed in the **registry.access.redhat.com/ubi10-minimal** container image. As a consequence, if you migrate your minimal container builds from a previous RHEL release to RHEL 10.0, and you enter the **microdnf reinstall tzdata** command to reinstall the **tzdata** package, you get an error message because the **tzdata** package is no longer installed by default. In this case, enter the **microdnf install tzdata** command to install **tzdata**.

Jira:RHELDOCS-18700[1]

The Podman v5.0 deprecations

In RHEL 10.0, the following is deprecated in Podman v5.0:

- The system connections and farm information stored in the **containers.conf** file are now read-only. The system connections and farm information will now be stored in the **podman.connections.json** file, managed only by Podman. Podman continues to support the old configuration options such as **[engine.service_destinations]** and the **[farms]** section. You can still add connections or farms manually if needed; however, it is not possible to delete a connection from the **containers.conf** file with the **podman system connection rm** command.

- The **slirp4netns** network mode is deprecated and will be removed in a future major release of RHEL. The **pasta** network mode is the default network mode for rootless containers.

- The **containernetworking-plugins** package and the CNI network stack are no longer supported.

  - If you upgrade from the previous RHEL versions to RHEL 10.0 or if you have a fresh installation of RHEL 10.0, the CNI is no longer available. As a result, you have to run the

podman rmi --all --force command to remove all images and containers that are using those images.

  - If present, the **cni** value in the containers.conf file for the **network_backend** option must be changed to **netavark** or can be unset.

Jira:RHEL-40641

The podman-tests package has been deprecated

The **podman-tests** package has been deprecated in the AppStream repository. The package is now available in the CodeReady Linux Builder (CRB). More information about the CRB repository can be found at https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html/package_manifest/repositc repository.

Jira:RHEL-67860

nodejs-18 and nodejs-18-minimal are deprecated

The **nodejs-18** and **nodejs-18-minimal** container images are now deprecated and will no longer receive feature updates. Use **nodejs-22** and **nodejs-22-minimal** instead.

Jira:RHELDOCS-20283[1]

10.13. DEPRECATED FEATURES IDENTIFIED IN PREVIOUS RELEASES

This part provides an overview of functionality that has been *deprecated* in Red Hat Enterprise Linux 10.

10.13.1. SSSD

The SMB1 protocol is deprecated in Samba

Starting with Samba 4.11, the insecure Server Message Block version 1 (SMB1) protocol is deprecated and will be removed in a future release.

To improve the security, by default, SMB1 is disabled in the Samba server and client utilities.

Jira:RHELDOCS-16612[1]

10.14. DEPRECATED PACKAGES

This section lists packages that have been deprecated and will probably not be included in a future major release of Red Hat Enterprise Linux.

> **IMPORTANT**
>
> The support status of deprecated packages remains unchanged within RHEL 10.

The following packages have been deprecated in RHEL 10:

- daxio
- ftp

- gvisor-tap-vsock-gvforwarder
- lftp
- libpmem
- libpmem2
- libpmemblk
- libpmemlog
- libpmemobj
- libpmemobj-cpp
- libpmempool
- libslirp
- nvml
- pmempool
- pmreorder
- sdl2-compat
- vsftpd
- wget

CHAPTER 11. KNOWN ISSUES

This version of Red Hat Enterprise Linux 10.1 is affected by the following newly identified and previously known issues. A known issue is listed in all future release notes until resolved, at which point it is published as a fixed issue. If you encountered an issue that is not listed in this section, report it by using the button in the top right corner of this page.

11.1. INSTALLER AND IMAGE CREATION

Crash dumps are not performed by default

By default, crash dumps do not occur for default installation methods using RHEL Image Mode, because the **crashkernel=** kernel argument is not set. To work around this problem, set a **crashkernel=** kernel argument at build or during installation time.

Jira:RHEL-82380

Podman and bootc do not share the same registry login process

Podman and **bootc** use different registry login processes when pulling images. As a consequence, if you login to an image by using Podman, logging to a registry for **bootc** will not work on that image. When you install an image mode for RHEL system, and login to registry.redhat.io by using the following command:

```
# podman login registry.redhat.io <username_password>
```

And then you attempt to switch to the **registry.redhat.io/rhel9/rhel-bootc** image with the following command:

```
# bootc switch registry.redhat.io/rhel9/rhel-bootc:9.4
```

You should be able to see the following message:

```
Queued for next boot: registry.redhat.io/rhel9/rhel-bootc:9.4
```

However, an error is displayed:

```
ERROR Switching: Pulling: Creating importer: Failed to invoke skopeo proxy method OpenImage:
remote error: unable to retrieve auth token: invalid username/password: unauthorized: Please login to
the Red Hat Registry using your Customer Portal credentials. Further instructions can be found here:
https://access.redhat.com/RegistryAuthentication
```

Workaround: Follow the steps Configuring container pull secrets to use authenticated registries with **bootc**.

Jira:RHELDOCS-18471[1]

cloud-init growpart skips with composefs is enabled

When composefs is enabled, if you generate an image from the generic base image, then the rootfs will not grow the filesystem, prompting an error similar to:

```
2024-04-30 17:27:53,543 - cc_growpart.py[DEBUG]: '/' SKIPPED: stat of 'overlay' failed: [Errno 2] No
such file or directory: 'overlay'
```

Workaround: You can add a custom growpart, by specifying the **rootfs** default size in the container, instead of dynamically choosing 100G at instance creation time to be able to write a partitioning config in the container.

Jira:RHEL-34859

Unable to build ISOs from a signed container

Trying to build an ISO disk image from a GPG or a simple signed container results in an error, similar to the following:

```
manifest - failed
Failed
Error: cannot run osbuild: running osbuild failed: exit status 1
2024/04/23 10:56:48 error: cannot run osbuild: running osbuild failed: exit status 1
```

This happens because the system fails to get the image source signatures.

Workaround: You can either remove the signature from the container image or build a derived container image. For example, to remove the signature, you can run the following command:

```
$ sudo skopeo copy --remove-signatures containers-storage:registry.redhat.io/rhel9/rhel-bootc:9.4
containers-storage:registry.redhat.io/rhel9/rhel-bootc:9.4
$ sudo podman run \
      --rm \
      -it \
      --privileged \
      --pull=newer \
      --security-opt label=type:unconfined_t \
      -v /var/lib/containers/storage:/var/lib/containers/storage \
      -v ~/images/iso:/output \
      quay.io/centos-bootc/bootc-image-builder \
      --type iso --local \
      registry.redhat.io/rhel9/rhel-bootc:9.4
```

To build a derived container image, and avoid adding a simple GPG signatures to it, see the Signing container images product documentation.

Jira:RHEL-34807

Hostname resolution fails with encrypted DNS and custom CA in boot options

While using the **inst.repo=** or **inst.stage2=** boot options in the kernel command line along with a remote installation URL, an encrypted DNS, and a custom CA certificate in the Kickstart file, the installation program attempts to download the **install.img** stage2 image before processing the Kickstart file. Consequently, the hostname resolution fails, leading to display of some errors before successfully fetching the stage2 image. Workaround: Define the installation source in the Kickstart file instead of the kernel command line.

Jira:RHEL-80672

The installation program becomes unresponsive during final RPM installation stage

An installation program might become unresponsive during the RPM installation process at the final stage. Before the issue occurs, you might see the repeated **Configuring rootfiles.noarch** messages. Workaround: Restart the installation process.

Jira:RHEL-67865[1]

Disabled keyboard layout switching by using shortcut during installation

To prevent confusion caused by a broken keyboard shortcut to change keyboard layout, this feature has been disabled in Anaconda. You cannot change keyboard layouts by using shortcuts during installation. Workaround: Use the keyboard layout icon on the top bar to switch layouts.

Jira:RHEL-74504

Bonding device with LACP takes longer to become operational, causing subscription failures

When configuring a bonding device with LACP by using both kernel command-line boot options and a Kickstart file, the connection is created during the **initramfs** stage but reactivated in Anaconda. As a consequence, it causes a temporary disruption that leads to system subscription failure via the **rhsm** Kickstart command.

Workaround: Add **--no-activate** to the Kickstart network configuration to keep the network operational. As a result, the system subscription completes successfully.

Jira:RHELDOCS-19853[1]

The **services** Kickstart command fails to disable the **firewalld** service

A bug in Anaconda prevents the **services --disabled=firewalld** command from disabling the **firewalld** service in Kickstart. Workaround: Use the **firewall --disabled** command instead. As a result, the **firewalld** service is disabled properly.

Jira:RHEL-83577

Installation program fails if **/boot** partition is not created when using **ostreecontainer**

When using the **ostreecontainer** Kickstart command to install a bootable container, the installation fails if the **/boot** partition is not created. This issue occurs because the installation program requires a dedicated **/boot** partition to proceed with the container deployment.

Workaround: Ensure that a **/boot** partition is defined in the Kickstart file or manually created during the installation process.

Jira:RHEL-66155

Kickstart installation fails with an unknown disk error when **ignoredisk** command precedes **iscsi** command

Installing RHEL by using the Kickstart method fails if the **ignoredisk** command is placed before the **iscsi** command. This issue occurs because the **iscsi** command attaches the specified iSCSI device during command parsing, while the **ignoredisk** command resolves device specifications simultaneously. If the **ignoredisk** command references an iSCSI device name before it is attached by the **iscsi** command, the installation fails with an "unknown disk" error.

Workaround: Ensure that the **iscsi** command is placed before the **ignoredisk** command in the Kickstart file to reference the iSCSI disk and enable successful installation.

Jira:RHEL-58827

The USB CD-ROM drive is not available as an installation source in Anaconda

Installation fails when the USB CD-ROM drive is the source for it and the Kickstart **ignoredisk --only-use=** command is specified. In this case, Anaconda cannot find and use this source disk.

Workaround: Use the **harddrive --partition=sdX --dir=**/ command to install from USB CD-ROM drive. As a result, the installation does not fail.

Jira:RHEL-58829

Driver disk menu fails to display user inputs on the console

When you start RHEL installation by using the **inst.dd** option on the kernel command line with a driver disk, the console fails to display the user input. Consequently, it seems that the application does not respond to the user input and stops responding, but displays the output which is confusing for users. However, this behavior does not affect the functionality, and user input gets registered after pressing **Enter**.

Workaround: To see the expected results, ignore the absence of user inputs in the console and press **Enter** when you finish adding inputs.

Jira:RHEL-58828

Insufficient disk space can cause deployment failure

Deploying a bootc container image on a package mode system without enough free disk space can result in installation errors and prevent the system from booting. Ensure adequate disk space is available for the image to install and adjust the provision logical volume before deployment.

Jira:RHELDOCS-19948[1]

Anaconda might not work correctly on s390x and ppc64le architectures

Image mode for RHEL supports **pp64le** and **s390x** architectures besides the already supported **x86_64** and ARM architectures. However, Anaconda might not function correctly on s390x and ppc64le architectures.

Jira:RHELDOCS-19496[1]

RHEL images on Azure marked as LVM require default layout resizing

When using **system-reinstall-bootc** or **bootc install** on Azure, RHEL images marked as LVM will require resizing the default layout.

Workaround: Use RHEL images labeled as RAW. This does not require resizing the default layout.

Jira:RHELDOCS-19945[1]

11.2. SECURITY

sq cannot generate keys in FIPS mode

The **sq** utility from the Sequoia PGP toolset uses the deprecated OpenSSL API for key generation. Consequently, you cannot generate keys with **sq** on the system running in FIPS mode.

Jira:RHEL-85985

GnuTLS cannot convert ML-DSA private keys to public ones

GnuTLS lacks an algorithm to convert a private ML-DSA key in the expanded form to a public ML-DSA key. Consequently, operations requiring both keys fail when only the expanded private key is provided.

Workaround: Use the **openssl** command to convert such a private key to a public key: **openssl dsa -in *<private_key>* -pubout -out *<public_key>***. As a result, the public key is available for use in other operations.

Jira:RHEL-102992

Updating the NSS database password corrupts the ML-DSA seed

Generating an ML-DSA key begins with a seed, which is sufficient to derive the key. However, the keys can also be expanded to accelerate subsequent operations. If you have ML-DSA keys in an NSS database, either generated or imported, both the expanded format and the seed are likely stored. Due to a bug in how NSS handles database re-encryption, if you change the password of the database, the seed attribute is not updated to accommodate the new password, and its value is permanently lost, even with the knowledge of the previous password.

To work around this problem, export the key before updating the password and re-import it after the update.

Jira:RHEL-114443

PQC for **rpm-sequoia** is always enabled in **crypto-policies**

In RHEL 10.1, the **rpm-sequoia** fails to verify dual-signed RPM packages if one of the algorithms used for signing is disabled in system-wide cryptographic policies. This problem is common on systems that have post-quantum (PQ) algorithms disabled and cannot install packages signed with both classic and PQ cryptography.

To prevent breaking the system, the enablement of PQ algorithms for **rpm-sequoia** is hard-coded on the **crypto-policies** level. As a result, PQ algorithms for **rpm-sequoia** are enabled regardless of any settings in **crypto-policies**.

Jira:RHEL-112392

SELinux policy rules for four **libvirt** services temporarily changed into permissive mode

Previously, the SELinux policy was changed to reflect the replacement of the legacy monolithic **libvirtd** daemon with a new set of modular daemons. Because this change requires testing of many scenarios, the following services have been temporarily changed into SELinux permissive mode:

- **virtqemud**
- **virtvboxd**
- **virtstoraged**
- **virtsecretd**

To prevent harmless AVC denials, **dontaudit** rules have been added to the SELinux policy for these services.

Jira:RHEL-77808[1]

Cryptographic tokens do not work in FIPS mode with **pkcs11-provider**

When the system runs in FIPS mode, the **pkcs11-provider** OpenSSL provider does not work correctly and the OpenSSL TLS toolkit falls back to the default provider. Consequently, OpenSSL fails to load PKCS #11 keys, and cryptographic tokens do not work in this scenario.

Workaround: Set the **pkcs11-module-assume-fips = true** parameter in the PKCS #11 section of the **openssl.cnf** file. See the **pkcs11-provider(7)** man page on your system for more information. With this configuration change, **pkcs11-provider** works in FIPS mode.

Jira:RHEL-68621

11.3. SHELLS AND COMMAND-LINE TOOLS

uname command produces an unknown output

The **uname** command displays unknown output with flags **--hardware-platform** and **--processor**. In the previous RHEL versions, **uname -i** and **uname -p** were aliases for **uname -m** and are not portable even across GNU or Linux distributions.

As a workaround, you can use the **-m** flag instead of the **-i** and **-p** flags.

Jira:RHEL-74146

11.4. INFRASTRUCTURE SERVICES

Hot-plugged memory is not available to VMs running on IBM Z by default

RHEL provides default udev rules that automatically set memory online when you hot plug memory to virtual machines (VMs) with **virtio-mem**. However, current udev rules do not include VMs running on IBM Z. As a consequence, after hot-plugging memory to VMs running on IBM Z with **virtio-mem**, the memory is not immediately available in the VM.

To work around this problem, set the **memhp_default_state=online** kernel parameter in the VM and reboot it. For example:

```
# grubby --update-kernel=ALL --args=memhp_default_state=online
```

As a result, the hot-plugged memory is available in the VM.

Jira:RHEL-92781

Nginx does not support PKCS #11 and TPM

The OpenSSL engines API was deprecated in RHEL 9 and removed from Nginx in RHEL 10. The corresponding functionality using the current OpenSSL providers API is not yet available. As a consequence, the Nginx HTTP server does not work with hardware security modules (HSMs) through PKCS #11 and Trusted Platform Module (TPM) devices.

Jira:RHEL-33742

Using the incorrect Perl database driver for MariaDB and MySQL can lead to unexpected results

The MariaDB database is a fork of MySQL. Over time, these services developed independently and are no longer fully compatible. These differences also affect the Perl database drivers. Consequently, if you use the **DBD::mysql** driver in a Perl application to connect to a MariaDB database, or the

DBD::MariaDB driver to connect to a MySQL database, operations can lead to unexpected results. For example, the driver can return incorrect data from read operations. To avoid such problems, use the Perl driver in your application that matches the database service.

Red Hat only supports the following scenarios:

- The Perl **DBD::MariaDB** driver with a MariaDB database
- The Perl **DBD::mysql** driver with a MySQL database

Note that RHEL 8 contained only the **DBD::mysql** driver. If you plan to upgrade to RHEL 9 and then to RHEL 10 and your application uses a MariaDB database, install the **perl-DBD-MariaDB** package after the upgrade and modify your application to use the **DBD::MariaDB** driver.

For further details, see the Red Hat Knowledgebase solution Support of MariaDB/MySQL cross-database connection from Perl db drivers.

Jira:RHELDOCS-19770[1]

11.5. FILE SYSTEMS AND STORAGE

Inconsistent NVMe device names after reboot

A new kernel feature that enables asynchronous NVMe namespace scans is introduced in RHEL 10, to accelerate NVMe disk detection. As a consequence of the asynchronous scans, the **/dev/nvmeXnY** device files might point to different namespaces after each reboot. This can lead to inconsistent device names. At this time, there is no known workaround for this issue.

Jira:RHEL-85845[1]

iSCSI-backed logical volumes fail to activate after a reboot

During installation, a logical volume spanning a local disk and an iSCSI device can fail to activate the iSCSI device in the installed system. This occurs where a non-root filesystem LVM logical volume is located both on a local disk and on an iSCSI device, which results in the iSCSI device not getting configured with **node.startup=onboot** by the installation program. As a result, the system cannot access the volume after reboot, because it doesn’t get automatically activated upon boot.

Workaround: Manually create the logical volume after the installation or update the iSCSI node configuration by setting **node.startup=automatic** in the relevant file in the **/var/lib/iscsi/nodes/** directory.

Jira:RHEL-53719

11.6. HIGH AVAILABILITY AND CLUSTERS

ACL roles should not reference location constraints with two rules

In Red Hat Enterprise Linux 10, more than one top-level rule in a location constraint is not supported. When upgrading from RHEL 9 to RHEL 10, verify that any ACL roles you have configured do not reference a location constraint with two rules and are still valid.

Jira:RHEL-62722

11.7. COMPILERS AND DEVELOPMENT TOOLS

The new version of TBB is incompatible

RHEL 10 includes the Threading Building Blocks (TBB) library version 2021.11.0, which is incompatible with the versions distributed with previous releases of RHEL. You must rebuild applications that use TBB to make them run on RHEL 10.

Jira:RHEL-33633

11.8. IDENTITY MANAGEMENT

An error message is displayed during migration from BDB to LMDB

When you run the **dsctl dblib bdb2mdb** command to migrate from Berkeley Database (BDB) to Lightning Memory-Mapped Database Manager (LMDB) and you have not enabled the replication, the following error message is displayed in the output:

```
Error: 97 - 1 - 53 - Server is unwilling to perform - [] - Unauthenticated binds are not allowed
```

Note that you can ignore the error message. The error occurs because Directory Server attempts to find the **replication_changelog.db** file that is not mandatory when the replication is disabled. This error does not prevent the migration from BDB to LMDB.

There is currently no workaround for this issue.

Jira:RHELDOCS-19016[1]

IdM in FIPS mode does not support using the NTLMSSP protocol to establish a two-way cross-forest trust

Establishing a two-way cross-forest trust between Active Directory (AD) and Identity Management (IdM) with FIPS mode enabled fails because the New Technology LAN Manager Security Support Provider (NTLMSSP) authentication is not FIPS-compliant. IdM in FIPS mode does not accept the RC4 NTLM hash that the AD domain controller uses when attempting to authenticate.

Jira:RHEL-12154[1]

Installing a RHEL 7 IdM client with a RHEL 10 IdM server in FIPS mode fails due to EMS enforcement

The TLS **Extended Master Secret** (EMS) extension (RFC 7627) is now mandatory for TLS 1.2 connections on FIPS-enabled RHEL 10 systems. This is in accordance with FIPS-140-3 requirements. However, the **openssl** version available in RHEL 7.9 and lower does not support EMS. In consequence, installing a RHEL 7 Identity Management (IdM) client with a FIPS-enabled IdM server running on RHEL 10 fails.

Workaround: Upgrade the host to RHEL 8 or later before installing an IdM client on it.

Jira:RHELDOCS-19015[1]

Automatic host keytab renewal via **adcli** run by SSSD is failing

In direct SSSD-AD integration, SSSD checks daily if the machine account password is older than the configured age in days and, if needed, tries to renew it. The configured age is set by the **ad_maximum_machine_account_password_age** value, with a default of **30** days. A value of **0** disables the renewal attempt.

However, currently there is an issue and the automatic renewal of the machine account password fails. If the password expires, this might result in the host losing access to the AD domain.

Workaround: Renew the password manually or via another means. Do not rely on the SSSD automatic renewal.

Jira:RHELDOCS-19172[1]

ldapmodify does not delete a single specific value from any attribute in cn=config

Currently, when you try to delete a value from any attribute in **cn=config**, the value remains in the attribute and the server might require a restart to fully remove it.

Workaround: Remove the entire attribute, including all its values, by performing a modify operation without specifying any values. Then re-add the values you need. Alternatively, use the following **dsconf** command to remove a specific value without a server restart:

```
# dsconf <instance_name> config delete <attribute_name>=<undesired_value>
```

Jira:RHEL-25071

11.9. SSSD

SSSD retrieves incomplete list of members if the group size exceeds 1500 members

During the integration of SSSD with Active Directory, SSSD retrieves incomplete group member lists when the group size exceeds 1500 members. This issue occurs because Active Directory’s MaxValRange policy, which restricts the number of members retrievable in a single query, is set to 1500 by default.

Workaround: Change the MaxValRange setting in Active Directory to accommodate larger group sizes.

Jira:RHELDOCS-19603[1]

11.10. DESKTOP

Plymouth duplicates log entries of the kernel log ringbuffer

Plymouth, an application which provides a graphical boot experience for Red Hat Enterprise Linux, has a "console syndication" feature that outputs log messages to all configured consoles during boot. The kernel can natively output log messages only to the last configured console. In the default configuration, the kernel is muted, but removing the **quiet** argument from the kernel command line unmutes the kernel, and causes both Plymouth and the kernel to send the boot log messages to the last-configured console. As a result, log messages might be duplicated on the last-configured console (for example ttyS0). Plymouth further duplicates these log entries by replaying the entire contents of the kernel log ringbuffer during boot and shutdown. To work around this problem, disable Plymouth.

Jira:RHEL-60198[1]

Standard mouse cursor is offset in VMs when using Mutter

When you use a standard mouse within a virtual machine (VM) configuration in the Mutter compositing window manager, you might notice an offset between the physical mouse cursor and the actual pointer within the virtual environment. The actual pointer might not even be visible in the virtual environment.

Workaround: If your scenario requires precise input, use a tablet as an input device in the VM configuration.

Jira:RHEL-69291

11.11. GRAPHICS INFRASTRUCTURES

Standard mouse cursor is offset in VMs when using Mutter

When you use a standard mouse within a virtual machine (VM) configuration in the Mutter compositing window manager, you might notice an offset between the physical mouse cursor and the actual pointer within the virtual environment. The actual pointer might not even be visible in the virtual environment.

Workaround: If your scenario requires precise input, use a tablet as an input device in the VM configuration.

Jira:RHEL-45898

11.12. THE WEB CONSOLE

VNC console in the RHEL web console does not work correctly on ARM64

Currently, when you import a virtual machine (VM) in the RHEL web console on ARM64 architecture and then you try to interact with it in the VNC console, the console does not react to your input.

Additionally, when you create a VM in the web console on ARM64 architecture, the VNC console does not display the last lines of your input.

Jira:RHEL-31993[1]

11.13. RED HAT ENTERPRISE LINUX SYSTEM ROLES

Ansible **rpm_key** modules fail to work with the OpenPGP v6 **RPM-GPG-KEY-redhat-release** key

RHEL 10.1 uses the Red Hat RPM signing key extended with a post-quantum public key and stored in the **/etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release** file in the OpenPGP v6 format. Because the Ansible **rpm_key** modules use the GnuPG tools, which cannot handle post-quantum keys and OpenPGP v6, the modules fail to work with this key.

Jira:RHEL-126844

PostgreSQL, MariaDB, and MySQL do not work with RHEL in image mode

The PostgreSQL, MariaDB, and MySQL database management systems do not use the **sysusers.d** directories to populate users and working directories. MariaDB and MySQL also do not use the **tmpfiles.d** directory. As a consequence, the database user can be missing and the database systems are not able to initialize because their working directory is missing. There is currently no workaround for this issue.

Jira:RHELDOCS-21374[1]

ansible-core does not install **sshpass** as a dependency

The **ansible-core** package does not install the **sshpass** package as a dependency. Consequently, you cannot use Ansible to manage systems over SSH with an SSH password.

Workaround: On the control node, manually install **sshpass** after you install **ansible-core**. As a result, you can use Ansible in the scenario described above.

Jira:RHEL-86829[1]

11.14. VIRTUALIZATION

Windows VMs might become unresponsive due to storage errors

On virtual machines (VMs) that use Windows guest operating systems, the system in some cases becomes unresponsive when under high I/O load. When this happens, the system logs a **viostor Reset to device, \Device\RaidPort3, was issued** error. There is currently no workaround for this issue.

Jira:RHEL-1609[1]

Windows 10 VMs with certain PCI devices might become unresponsive on boot

Currently, a virtual machine (VM) that uses a Windows 10 guest operating system might become unresponsive during boot if a **virtio-win-scsi** PCI device with a local disk back end is attached to the VM.

Workaround: Boot the VM with the **multi_queue** option enabled.

Jira:RHEL-1084[1]

Using virtiofs with the rsync and du commands can result in too many open files errors

The **virtiofsd** daemon keeps file descriptors open until the guest invalidates its cache. When tracking a large number of files, this can cause **virtiofsd** on the host to exceed the maximum open files limit.

As a consequence, when sharing a large number of files with the guest with **virtiofs**, using the **rsync** and **du** commands in the shared directory can result in the **too many open files** error.

To work around this problem, increase the **virtiofsd** maximum open files limit in the XML configuration of the guest. For example:

```
<filesystem type='mount' accessmode='passthrough'>
...
<binary path='/usr/libexec/virtiofsd' xattr='on'>
  <openfiles max='2000000'/>
</binary>
...
</filesystem>
```

In this example, the **<openfiles max>** attribute is set to two million files.

For more information, see the Virtiofs 'too many open files' errors with rsync and du commands KCS Solution.

Jira:RHEL-99895[1]

VMs with SEV-SNP enabled fail to boot with arch-capabilities=on

Virtual machines (VMs) with SEV-SNP enabled fail to boot when using the **arch-capabilities=on** CPU flag.

To work around this problem, disable the **arch-capabilities** feature in the CPU section of the VM's configuration:

```
<cpu mode='host-passthrough' check='none'>
  <feature name='arch-capabilities' policy='disable'/>
</cpu>
```

Jira:RHEL-100313[1]

Installing the VirtIO-Win bundle cannot be canceled

Currently, if you start the installation of **virtio-win** drivers from the VirtIO-Win installer bundle in a Windows guest operating system, clicking the **Cancel** button during the installation does not correctly stop it. The installation wizard interface displays a "Setup Failed" screen, but the drivers are installed and the IP address of the guest is reset.

Jira:RHEL-53962, Jira:RHEL-53965

A virtual machine with a large amount of bootable data disks might fail to start

If you attempt to start a virtual machine (VM) with a large amount of bootable data disks, the VM might fail to boot with this error: **Something has gone seriously wrong: import_mok_state() failed: Volume Full**

Workaround: Decrease the number of bootable data disks and use one system disk. To ensure the system disk is first in the boot order, add **boot order=1** to the device definition of the system disk in the XML configuration. For example:

```
<disk type='file' device='disk'>
  <driver name='qemu' type='qcow2'/>
  <source file='/path/to/disk.qcow2'/>
  <target dev='vda' bus='virtio'/>
  <boot order='1'/>
</disk>
```

Set boot order only for the system disk.

Jira:RHEL-68418

VMs with large memory cannot boot on SEV-SNP host with AMD Genoa CPUs

Currently, virtual machines (VMs) cannot boot on hosts that use a 4th Generation AMD EPYC processor (also known as Genoa) and have the AMD Secure Encrypted Virtualization with Secure Nested Paging (SEV-SNP) feature enabled. Instead of booting, a kernel panic occurs in the VM.

Jira:RHEL-32892[1]

The virtio balloon driver sometimes does not work on Windows 10 and Windows 11 VMs

Under certain circumstances, the **virtio-balloon** driver does not work correctly on virtual machines (VMs) that use a Windows 10 or Windows 11 guest operating system. As a consequence, such VMs might not use their assigned memory efficiently.

Jira:RHEL-12118

Windows 11 VMs with a memory balloon device set might close unexpectedly during reboot

Currently, rebooting virtual machines (VMs) that use a Windows 11 guest operating system and a memory balloon device in some cases fails with a **DRIVER POWER STAT FAILURE** blue-screen error.

Jira:RHEL-935[1]

Windows VM with VBS and IOMMU device fails to boot

When you boot a Windows VM with Virtualization Based Security (VBS) enabled and an Input-Output Memory Management Unit (IOMMU) device by using the **qemu-kvm** utility, the booting sequence only shows the boot screen, resulting in an incomplete booting process.

Workaround: Ensure the VM domain XML is configured as below:

```
<features>
  <ioapic driver='qemu'/>
</features>
<devices>
<iommu model='intel'>
   <driver intremap='on' eim='off' aw_bits='48'/>
   <alias name='iommu0'/>
</iommu>
<memballoon model='virtio'>
   <alias name='balloon0'/>
   <address type='pci' domain='0x0000' bus='0x03' slot='0x00' function='0x0'/>
   <driver iommu='on' ats='on'/>
</memballoon>
</devices>
```

Otherwise, the Windows VM cannot boot.

Jira:RHEL-45585[1]

Windows VM running on Sapphire Rapids CPU with hypervisor launch type set to auto might fail to boot when restarted

If you set the hypervisor launch type to **auto** in a Windows virtual machine (VM) running on a Sapphire Rapids CPU, the VM might fail to boot when it is restarted. For example, you can set the hypervisor launch type to **auto** by using the **bcdedit /set hypervisorlaunchtype Auto** command.

Workaround: Do not set the hypervisor launch type to **auto** in the Windows VM.

Jira:RHEL-67699[1]

Hot-plugging vCPUs and memory to Windows guests with VBS does not work

Currently, Windows Virtualization-based Security (VBS) is not compatible with hot-plugging CPU and memory resources. As a consequence, attempting to attach memory or vCPUs to a running Windows virtual machine (VM) with VBS enabled only adds the resources to the VM after the guest system is restarted.

Jira:RHEL-66229, Jira:RHELDOCS-19066

VMs with 5-level page merging and a lot of memory sometimes fail to start

VMs with the following configuration fail to boot if you set the **host-phys-bits-limit** parameter to **49** or more:

- The VM has more than 1 TB of assigned memory
- The VM uses the 5-level page merging feature
- The host uses System Management Mode (SMM) in its firmware

Instead, attempting to boot the VM fails with **ERROR: Out of aligned pages**.

Workaround: Set the **host-phys-bits-limit** parameter to 48 or less.

Jira:RHEL-82685

Cloning or restoring RHEL 9 virtual machines that use LVM on Nutanix AHV causes non-root partitions to disappear

When running a RHEL 9 guest operating system on a virtual machine (VM) hosted on the Nutanix AHV hypervisor, restoring the VM from a snapshot or cloning the VM currently causes non-root partitions in the VM to disappear if the guest is using Logical Volume Management (LVM). As a consequence, the following problems occur:

- After restoring the VM from a snapshot, the VM cannot boot, and instead enters emergency mode.
- A VM created by cloning cannot boot, and instead enters emergency mode.

To work around these problems, do the following in emergency mode of the VM:

1. Remove the LVM system devices file:

   ```
   # rm /etc/lvm/devices/system.devices
   ```

2. Re-create LVM device settings:

   ```
   # vgimportdevices -a
   ```

3. Reboot the VM

This makes it possible for the cloned or restored VM to boot up correctly.

Alternatively, to prevent the issue from occurring, do the following before cloning a VM or creating a VM snapshot:

1. Uncomment the **use_devicesfile = 0** line in the **/etc/lvm/lvm.conf** file.
2. Regenerate initramfs. To do so, use the following steps in the VM and replace ***<kernelVersion>*** with the full version of the kernel that you want to rebuild.
   a. Back up the current **initramfs** configuration:

      ```
      # cp /boot/initramfs-<kernelVersion>.img /boot/initramfs-<kernelVersion>.img.bak
      ```

   b. Build **initramfs**:

```
# dracut -f /boot/initramfs-<kernelVersion>.img  <kernelVersion>
```

3. Reboot the VM to verify successful boot.

Jira:RHELPLAN-114103[1]

Live VM memory dumps and VM snapshots might not work correctly on IBM Z

Currently, attempting to create a memory dump of a running VM by using the **virsh dump --live**` command on an IBM Z host may cause the VM to become unresponsive. In rare cases, creating a snapshot of a running VM can also cause the VM to become unresponsive on IBM Z.

Jira:RHELDOCS-21707[1]

11.15. RHEL IN CLOUD ENVIRONMENTS

RDMA devices currently do not work on vSphere

When using a RHEL 10 instance on the VMware vSphere platform, the **vmw_pvrdma** module currently does not install properly. As a consequence, VMware paravirtual remote direct memory access (PVRDMA) devices do not work on the affected instances.

Jira:RHEL-41133[1]

The leapp upgrade fails when upgrading from RHEL 9.6 to RHEL 10.0 for the cloud-init network configuration

If you deploy RHEL 9.6 with the **cloud-init** default configuration and with **sysconfig** as the default network configuration directory, the **sysconfig** configuration files do not support the **ifcfg** legacy format for RHEL 10.0. Consequently, the **leapp** upgrade fails when upgrading from RHEL 9.6 to RHEL 10.0 for the legacy network configuration files, such as ifcfg-*<enp1s0>*.

Workaround: Convert the **sysconfig** configuration files into the NetworkManager native **keyfile** format:

1. Modify the connection:

```
# nmcli connection modify "System <enp1s0>" connection.id "cloud-init <enp1s0>"
```

2. Migrate the connection:

```
# nmcli connection migrate /etc/sysconfig/network-scripts/ifcfg-<enp1s0>
```

3. Move the connection profile:

```
# sudo mv /etc/NetworkManager/system-connections/"cloud-init <enp1s0>.nmconnection"
/etc/NetworkManager/system-connections/cloud-init-<enp1s0>.nmconnection
```

4. Reload the network connection settings:

```
# nmcli conn reload
```

As a result, the leapp upgrade from RHEL 9.6 to RHEL 10.0 now works with the updated configuration.

Jira:RHEL-82209[1]

Upgrading a RHEL 9.6 guest on VMware ESXi to RHEL 10.0 causes **cloud-init** to rewrite the network configuration

After a upgrading a RHEL guest on the VMware ESXi hypervisor from RHEL 9.6 to RHEL 10.0, the **cloud-init** tool currently cannot detect the VMware data source and cannot restore its configuration from the cache. As a consequence, **cloud-init** reverts to the **None** data source, and rewrites the network configuration of the guest.

Workaround: Remove the **disable_vmware_customization** flag from the **/etc/cloud/cloud.cfg** file before you reboot the guest during the upgrade process. As a result, the upgraded guest will retain its previous network configuration.

Jira:RHEL-82210[1]

kdump fails to complete on the Azure Confidential VMs

When you experience a kernel crash on a Red Hat Enterprise Linux VM on the Azure Confidential VM instances, in this case DCv5 and ECv5 series, the **kdump** process might not complete and the VM becomes unresponsive. As a result, after a forced reboot, there is a **vmcore-incomplete** file.

Jira:RHEL-75576[1]

BIOS or UEFI supported Hyper-V Windows Server 2016 VM fails to boot if a host uses the AMD EPYC CPU processor

With the Hyper-V enabled setting, Hyper-V Windows Server 2016 VM fails to boot on the AMD EPYC CPU host.

Workaround: Check for the following log message:

```
kvm: Booting SMP Windows KVM VM with !XSAVES && XSAVEC.
If it fails to boot try disabling XSAVEC in the VM config.
```

And try adding **xsavec=off** to **-cpu cmdline** to boot Hyper-V Windows Server 2016 VM.

Jira:RHEL-38957[1]

11.16. CONTAINERS

FIPS bootc image creation fails on FIPS enabled host

Building a disk image on a host by using Podman with enabled the FIPS mode fails with the exit code 3 because of the update-crypto-policies package:

```
# Enable the FIPS crypto policy
# crypto-policies-scripts is not installed by default in RHEL-10
RUN dnf install -y crypto-policies-scripts && update-crypto-policies --no-reload --set FIPS
```

Workaround: Build the bootc image with FIPS mode disabled.

Jira:RHELDOCS-19539

11.17. RHEL LIGHTSPEED

The command-line assistant cannot verify the Satellite Server certificate

The command-line assistant does not recognize the Satellite certificate authority (CA) certificate for the Red Hat Satellite Server. The Satellite CA certificate is used to issue and sign certificates for hosts that register with and are managed by Satellite. As a consequence, the command-line assistant cannot establish secure connections to the Satellite Server, which prevents it from functioning correctly.

Work around: copy the Satellite CA certificate to the system truststore and update the CA trust database:

```
$ sudo cp /etc/rhsm/ca/katello /etc/pki/ca-trust/source/anchors/*
$ sudo update-ca-trust
```

Jira:RHELDOCS-21325[1]

Command-line assistant configuration file changes are not applied immediately

When making changes in the **etc/xdg/command-line-assistant/config.toml** configuration file, it takes around 30 to 60 seconds for the command-line assistant daemon to recognize the changes, instead of applying the changes immediately. The command-line assistant is also missing the **reload** functionality.

Workaround: Follow the steps:

1. Make the changes that you need to the **config.toml** configuration file.

2. Run the following command:

```
# systemctl restart clad
```

Jira:RHELDOCS-19734[1]

Security risk when using unsupported AI models

Do not use unsupported models. Changing the RHEL Offline Container model to unsupported models might allow the execution of arbitrary code or compromise the integrity of Red Hat Enterprise Linux (RHEL).

No known workaround exists.

Jira:RHELDOCS-21726[1]

11.18. KNOWN ISSUES IDENTIFIED IN PREVIOUS RELEASES

This part describes known issues in Red Hat Enterprise Linux 10.1.

11.18.1. Networking

Failure to update the session key causes the connection to break

Kernel Transport Layer Security (kTLS) protocol does not support updating the session key, which is used by the symmetric cipher. Consequently, the user cannot update the key, which causes a connection break.

Workaround: Disable kTLS. As a result, with the workaround, it is possible to successfully update the session key.

Jira:RHELDOCS-20686[1]

kTLS does not support offloading of TLS 1.3 to NICs

Kernel Transport Layer Security (kTLS) does not support offloading of TLS 1.3 to NICs. Consequently, software encryption is used with TLS 1.3 even when the NICs support TLS offload.

Workaround: Disable TLS 1.3 if offload is required. As a result, you can offload only TLS 1.2. When TLS 1.3 is in use, there is lower performance, since TLS 1.3 cannot be offloaded.

Jira:RHELDOCS-20687[1]

11.18.2. Virtualization

The **Extended Master Secret** TLS Extension is now enforced on FIPS-enabled systems

With the release of the RHSA-2023:3722 advisory, the TLS **Extended Master Secret** (EMS) extension (RFC 7627) is mandatory for TLS 1.2 connections on FIPS-enabled RHEL 9 systems. This is in accordance with FIPS-140-3 requirements. TLS 1.3 is not affected.

Legacy clients that do not support EMS or TLS 1.3 now cannot connect to FIPS servers running on RHEL 9 and 10. Similarly, RHEL 9 and 10 clients in FIPS mode cannot connect to servers that only support TLS 1.2 without EMS. This in practice means that these clients cannot connect to servers on RHEL 6, RHEL 7 and non-RHEL legacy operating systems. This is because the legacy 1.0.x versions of OpenSSL do not support EMS or TLS 1.3.

In addition, connecting from a FIPS-enabled RHEL client to a hypervisor such as VMWare ESX now fails with a **Provider routines::ems not enabled** error if the hypervisor uses TLS 1.2 without EMS. To work around this problem, update the hypervisor to support TLS 1.3 or TLS 1.2 with the EMS extension. For VMWare vSphere, this means version 8.0 or later.

For more information, see TLS Extension "Extended Master Secret" enforced with Red Hat Enterprise Linux 9.2 and later.

Jira:RHEL-13340[1]

CHAPTER 12. FIXED ISSUES

This version provides the following fixed issues and other problems that have a significant impact.

12.1. INSTALLER AND IMAGE CREATION

Installation no longer fails if a VDO logical volume is present

Before this update, installing RHEL failed when users attempted to remove a pre-existing Logical Volume Manager Virtual Data Optimizer (LVM VDO) volume on systems without the **dm_vdo** kernel module. With this update, installation succeeds when removing an LVM VDO volume on systems without VDO support.

Jira:RHEL-84685[1]

Enhanced installation program to enable container-based deployments on **s390x**

The RHEL installation program now supports deploying bootable containers in Image Mode on the **s390x** architectures by using the **ostreecontainer** Kickstart command. This enhancement removes previous limitations and ensures consistent deployment options across supported architectures. Users can now automate installations on **s390x** systems by using container-based workflows.

Jira:RHEL-88558

The installation program now respects the **BOOTIF** boot argument

Previously, the RHEL installation program ignored the **BOOTIF=<MAC>** boot argument and activated all the available network interfaces. With this fix, the installation program now properly processes the **BOOTIF** argument and ensures that only the designated network device is activated during the installation process.

Jira:RHEL-69400[1]

12.2. SECURITY

fapolicyd no longer causes the RPM database to crash with repeated updates

Before this update, repeated updates of the RPM database when **fapolicyd** was in enforcing mode caused a bus error (SIGBUS), which caused the RPM database to terminate unexpectedly. With this release, fapolicyd SIGBUS protection for RPM database updates has been improved. As a result, the RPM database no longer crashes when repeatedly updating it with **fapolicyd** enabled.

Jira:RHEL-94540[1]

SSH connection fail no longer displays verbose help message

Before this update, when SSH connection failed, a message with common SSH errors and a link to Red Hat help was displayed. As a consequence, the help message in the error output broke user scripts and automation. With this update, the help message displays only when SSH is run with log level **debug1** or higher. As a result, the error output does not include any unexpected messages by default.

Jira:RHEL-93957

fapolicyd-cli --file add no longer fails when processing non-regular files

Before this update, the **fapolicyd-cli --file add** command failed to add directories containing non-regular files, such as sockets, to the trust database. With this update, the problem is resolved, and **fapolicyd-cli --file add** no longer fails in the described scenario.

Jira:RHEL-105425

GnuTLS supports standard ML-DSA formats

In RHEL 10.0, GnuTLS tools used non-standard serialization formats for ML-DSA private keys. Consequently, the **certtool -p** command exported ML-DSA private keys that were not compatible with IETF-compliant implementations. Likewise, keys exported by other software did not work with GnuTLS. With this update, GnuTLS support standard ML-DSA formats and generate interoperable private keys.

Jira:RHEL-85829

OpenSSL stores ML-KEM and ML-DSA private keys in standard formats

In RHEL 10.0, the open quantum-safe provider for OpenSSL (**oqsprovider**) generated private keys in a format that did not conform to any of the file formats proposed by the IETF LAMPS workgroup. Consequently, the key files were unreadable by other applications that follow the IETF standard and could not be handled by applications that require providing the key in the seed format for import. With this update, OpenSSL no longer uses **oqsprovider** and its post-quantum cryptography (PQC) implementation generates the keys in standard formats. As a result, you can use OpenSSL ML-KEM and ML-DSA keys for storing long-term secrets.

Jira:RHEL-72719

12.3. SOFTWARE MANAGEMENT

DNF Automatic uses the correct RHEL minor version when the EPEL repository is enabled

Before this update, when you used the DNF Automatic tool with the Extra Packages for Enterprise Linux (EPEL) repository enabled, the tool expanded the EPEL metalink URL to a wrong address. This caused the tool to download packages for the minor version of RHEL 10 that is still in development instead of the last released minor version. With this update, updating of the **releasever_major** and **releasever_minor** variables was fixed in the **dnf.cli.cli.Cli._read_conf_file()** method. As a result, DNF Automatic correctly detects a minor release number and downloads an EPEL repository matching the RHEL major and minor release version.

Jira:RHEL-106141

12.4. SHELLS AND COMMAND-LINE TOOLS

/var/lib/tftpboot directory is created by default in Image Mode deployments

Previously, in Image Mode deployments, installing the **tftp-server** package did not create the **/var/lib/tftpboot** directory. This occurred because changes to the **/var** directory were not applied when additional packages were added to existing Image Mode deployments.

With this update, the **/var/lib/tftpboot** directory is automatically created in all Image Mode deployments.

Jira:RHEL-79983[1]

The IPL output method of ReaR on IBM Z follows RHEL 9 file naming conventions

Before this update, when using the deprecated **IPL** output method of Relax-and-Recover (ReaR) on

IBM Z, the resulting recovery kernel was named **kernel-$RAMDISK_SUFFIX** and the RAM disk image was named **initramfs-$RAMDISK_SUFFIX.img**. This naming convention differed from RHEL 9, which uses **vmlinuz-$kernel_version** for the kernel and **initrd.cgz** for the RAM disk image. Consequently, custom scripts used to process these recovery images required manual adjustment after an upgrade from RHEL 9 to RHEL 10 due to the file name changes.

With this update, the RHEL 9 file naming behavior is restored for the deprecated **IPL** output method on IBM Z, preserving backward compatibility with previous major releases of RHEL. The kernel image is now named **vmlinuz-$kernel_version** and the RAM disk image is named **initrd.cgz**.

As a result, the kernel and **initramfs** images are named consistently with RHEL 9, eliminating the need to adapt scripts when upgrading from RHEL 9 to RHEL 10.1. This represents a change in behavior from RHEL 10.0. The **RAMDISK** output method should be used instead of the deprecated **IPL** method, as **RAMDISK** uses the naming convention **kernel-$RAMDISK_SUFFIX** and **initramfs-$RAMDISK_SUFFIX.img** and is uniform across all supported architectures.

Jira:RHEL-102563[1]

12.5. INFRASTRUCTURE SERVICES

The chronyc reload sources command now correctly handles hostname-specified sources

Previously, the **chronyc reload sources** command in **chronyd** incorrectly reloaded sources from the **sourcedir** directory specified in the **chrony.conf** file. This behavior caused the **chronyd** to duplicate sources when a hostname resolved to multiple IP addresses, resulting in an unexpected increase in the number of sources.

With this update, the **chronyc reload sources** command correctly handles sources specified with a hostname. As a result, reloading of sources does not change the number of used sources.

Jira:RHEL-95017

12.6. NETWORKING

The custom iproute2 settings in /etc/iproute2/ works as expected

Previously, if you updated to RHEL 10.0, the **iproute2** package stored the default configuration in the **/usr/share/iproute2/** directory. Additionally, if you had a custom configuration in **/etc/iproute2/**, the update renamed these files and appended the **.rpmsave** suffix. As a consequence, the custom settings were no longer applied. If you update to the RHEL 10.1 version of the **iproute2** package, the installation script in the package no longer renames custom configuration files and, if it finds files with **.rpmsave** suffix in **/etc/iproute2/**, the script removes this suffix. As a result, custom settings work again as expected.

Note that the **iproute2** default settings remain in **/usr/share/iproute2/**.

Jira:RHEL-99163[1]

The kernel no longer panics if you reduce the number of SR-IOV VFs at runtime

In previous releases, the Linux kernel could panic if all of the following conditions applied:

- The host has Input-Output Memory Management Unit (IOMMU) enabled.

- A network driver uses a page pool.
- You reduced the number of Single Root I/O Virtualization (SR-IOV) Virtual Functions (VFs) of the network interface that uses this driver.

With this update, the kernel tracks which DMA-mapped memory pages belong to a page pool. When a page pool is destroyed, for example by removing a VF, the memory pages are DMA-unmapped. This prevents attempts to unmap the memory pages after the VF has already been removed. As a result, the kernel no longer panics if you reduce the number of SR-IOV VFs at runtime.

Jira:RHEL-68401[1]

The NAT engine now checks for address collisions in reply direction

Before this update, the network address translation (NAT) engine did not check for address collisions in the reply direction. This led to connection failures when new incoming connections used the same source addresses and source ports as existing connections. With this release, the NAT engine now checks the reply direction and detects the collision, and the source port the new connection is internally remapped to a new available port number. As a result, the connection proceeds as expected.

Jira:RHEL-99656[1]

The nft_fib expression now returns consistent results for both IPv4 and IPv6 in VRF domains

Before this update, the Netfilter "nft_fib" expression returned **unicast** instead of **local** if a device was part of a Virtual Routing and Forwarding (VRF) domain. Additionally, the **fib daddr . iif** expression behaved differently for IPv4 and IPv6 packets arriving on a VRF interface. For an incoming IPv6 packet, it incorrectly returned the name of the underlying physical interface, whereas for an IPv4 packet, it correctly returns the name of the VRF device itself. With this update, the **nft_fib** expression now provides consistent results for both IPv4 and IPv6 when the device is part of a VRF domain.

Jira:RHEL-88574[1]

Network authentication methods using PKCS #11 with wpa_supplicant has been fixed

In RHEL 10, engines that are not compatible with the Federal Information Processing Standard (FIPS), such as the OpenSSL engine API, have been removed. Consequently, the dependent **wpa_supplicant** service could not load X.509 certificates and keys stored in PKCS #11 URI format. This prevented any EAP-TLS authentication method and variants using PKCS #11 did not connect to the relevant network. To fix this problem, **wpa_supplicant** now depends on the **pkcs11-provider** package and uses the same-named library to load X.509 certificates and keys from a PKCS #11 storage. As a result, network authentication methods using PKCS #11 work as expected.

Jira:RHEL-86951

VMware vCenter now correctly removes a SATA disk from a running RHEL VM

When using the VMware vCenter interface to remove a SATA disk from a running RHEL 10 guest on the VMware ESXi hypervisor, the disk previously did not get removed fully. It stopped being functional and disappeared from the guest in the vCenter interface, but the SCSI interface still detected the disk as attached in the guest. This update fixes the issue, and the SATA disk is fully removed in the described scenario.

Jira:RHEL-79913[1]

12.7. KERNEL

Updated the **stalld** scheduling policy regression to prevent performance degradation

Before this update, the Node Tuning Operator CI was broken because of a change in **stalld** scheduling policy., This change caused the service to revert to SCHED_OTHER instead of SCHED_FIFO after starting. Consequently, real-time workloads could experience performance degradation, and you could not merge PR. With this update, the **systemd** unit file sets **stalld** priority to 10, ensuring that **stalld** runs with SCHED_FIFO. This restores expected behavior and improves performance for real-time workloads.

Jira:RHEL-109112

osnoise/cpus allows setting a long comma-separated list of CPUs

Before this update, you could not set a lengthy comma-separated list of CPUs in osnoise/cpus because of an invalid argument error. This restriction impacted latency debugging and troubleshooting. With this release, you can input a long comma-separated list of CPUs in osnoise/cpus to enhance RTLA latency debugging and troubleshooting.

Jira:RHEL-86520[1]

rtla timerlat now handles high-frequency sampling on systems with 100+ CPUs

Before this update, **rtla timerlat** could not process timerlat samples with 100us period or faster on systems with more than 100 CPUs due to insufficient **tracefs** buffer handling. As a consequence, samples were dropped and **timerlat** measurements became inaccurate, affecting real-time performance analysis. With this release, **timerlat** samples are collected directly on measurement CPUs, eliminating buffer overflow issues. As a result, rtla timerlat provides accurate measurements on high-core-count systems, enabling reliable real-time performance analysis.

Jira:RHEL-77357[1]

rtla timerlat does not reset osnoise stop tracing threshold during startup

Before this update, using the **rtla timerlat** multiple times without clearing the stop_tracing flags would leave or left **RTLA** in an inconsistent state. As a consequence, tracing did not stop correctly in case stop tracing was not requested via the **-a**, **-T**, or **-i** options. This led to inaccurate data being reported, since **RTLA** exited when it shouldn't have. With this update, **rtla-timerlat** resets stop tracing variables, preventing early exit, and as a result, program stability is improved.

Jira:RHEL-73865[1]

12.8. BOOT LOADER

The GRUB2 **net_del_dns** command deletes the DNS server correctly

Before this update, if you attempted to delete the DNS server by using the **net_del_dns** command, it added the DNS server back erroneously because of incorrect implementation, and returned an error. With this fix, the **add** command was replaced by the **remove** command in the **net_del_dns** implementation. As a result, you can delete the DNS server by using the **net_del_dns** command.

Jira:RHEL-4378

12.9. FILE SYSTEMS AND STORAGE

multipathd can monitor devices with offline paths

Before this update, when a user created a multipath device while some paths to the device were in the offline state, the **multipathd** daemon did not monitor the device or its paths. Consequently, if paths failed, they were never restored, even if they became available again. With this update, the **multipathd** daemon monitors the multipath device and its offline paths. **multipathd** also adds the paths to the multipath device if they become online.

Jira:RHEL-82535[1]

The RHEL installation program removes corrupted LVM thin volumes

Previously, the presence of corrupted LVM thin volumes caused storage configuration errors, blocking the installation process. With this fix, the RHEL installation program now detects and removes broken thin volumes. As a result, users do not have to intervene in the installation process manually.

Jira:RHEL-84663

12.10. HIGH AVAILABILITY AND CLUSTERS

pcs commands no longer fail due to improperly capitalized **target-role** values

Before this update, if a resource's **target-role** meta-attribute was set to a value that was not capitalized, such as **stopped** instead of **Stopped**, **pcs** failed to parse the cluster status. This parsing error caused **pcs status query resource** commands and commands for deleting resources, including **pcs resource delete**, to fail.

With this update, the cluster status parsing logic in **pcs** has been made more flexible.

As a result, **pcs** commands function correctly even when a resource has a **target-role** meta-attribute with an improperly capitalized value.

Jira:RHEL-92043

fence_ibm_powervs supports plain text token files

Before this update, the **fence_ibm_powervs** agent could only read authentication tokens from files that were formatted as JSON. It failed to read tokens from plain text files.

With this update, the file reading logic in the agent has been corrected.

As a result, the **fence_ibm_powervs** agent can use token files that are in either JSON or plain text format.

Jira:RHEL-88569[1]

Pacemaker Remote nodes are no longer fenced unnecessarily when quorum is lost

Before this update, in certain cluster configurations, a Pacemaker Remote node could be fenced when its partition lost quorum, even if the resource managing that node could be safely restarted on a different, quorate node. This behavior caused unnecessary downtime for the services running on the Pacemaker Remote node.

With this update, a new cluster property, **fence-remote-without-quorum**, has been introduced to control this behavior.

As a result, with the default **fence-remote-without-quorum=false** setting, Pacemaker no longer fences a remote node if its managing resource can be recovered on a quorate node, thus improving service availability.

Jira:RHEL-86146[1]

Pacemaker no longer requires manual IPC buffer tuning for large clusters

Before this update, in clusters with a large number of nodes or resources, Pacemaker's internal communication could exceed the default buffer size. This would result in logged errors and could cause command-line tools to be slow or unresponsive. Users sometimes had to manually increase the **PCMK_ipc_buffer** setting to resolve these issues.

With this update, Pacemaker's inter-process communication (IPC) code has been enhanced to handle large messages without a fixed buffer limit.

As a result, the **PCMK_ipc_buffer** setting is no longer needed and has been deprecated. Command-line tools are more responsive on complex clusters, and buffer size errors are no longer logged.

Jira:RHEL-86144[1]

systemd resources with long start or stop times are handled correctly

Before this update, Pacemaker polled for the result of start and stop actions on **systemd** resources with a fixed timeout. If a resource took longer to start or stop than this timeout, Pacemaker incorrectly marked the resource as failed.

With this update, Pacemaker listens for DBus messages from **systemd** to be notified when a start or stop action completes.

As a result, Pacemaker correctly detects the status of long-running **systemd** services, and resources are no longer marked as failed due to a timeout.

Jira:RHEL-71181[1]

12.11. COMPILERS AND DEVELOPMENT TOOLS

glibc package updated to include bug fixes and enhancements from the upstream 2.39 release

Upstream development delivered multiple bug fixes and enhancements to **glibc** 2.39. As a consequence, RHEL 10 **glibc** became outdated relative to the upstream release, resulting in gaps in features and unresolved bugs. To address this, the fixes and enhancements from the **glibc** 2.39 upstream release branch were backported to RHEL 10. As a result, RHEL 10 **glibc** now provides feature and bug parity with the upstream **glibc** 2.39 release branch as of August 20, 2025.

Jira:RHEL-109536

Certain programs no longer crash when running the glibc dynamic linker in auditing mode

Previously, the **glibc** dynamic linker in **LD_AUDIT** mode could allocate internal data structures by using the main **calloc** function before the linker initialized the main **malloc** subsystem. As a consequence, certain programs terminated unexpectedly in the **calloc** function during startup. With this update, the process startup sequence has been rearranged so that **calloc** memory allocation occurs before switching to the main **malloc** function, using the internal **malloc** implementation during startup. As a result, programs no longer crash during startup in the **calloc** function when the dynamic linker is in auditing mode.

Jira:RHEL-109703[1]

Improved support for recursive **dlopen** calls in audit modules in **glibc**

Previously, recursive **dlopen** calls from auditors could trigger an **r_state == RT_CONSISTENT** assertion failure in glibc's **dl-open.c**. As a consequence, applications exited unexpectedly when auditors were active. With this update, the dynamic linker reports consistency of its internal data structures earlier during an in-progress **dlopen** call. As a result, recursive **dlopen** operations for auditors are supported in more cases.

Jira:RHEL-109702

glibc: **ctype.h** macros caused segmentation faults in multithreaded programs with multiple **libc.so**

Previously, the internal state for **<ctype.h>** in secondary C library copies created by audit or with **dlmopen** failed to initialize for threads created with **pthread_create**. As a consequence, using **<ctype.h>** functionality, either directly or indirectly, in secondary threads and namespaces resulted in program crashes.

With this update, the internal state for **<ctype.h>** is initialized to refer to the **C** locale for secondary threads and namespaces. As a result, using functionality from **<ctype.h>** in these scenarios no longer causes crashes.

Jira:RHEL-72018

getent group now returns complete member lists when NSS merge encounters **ERANGE** in glibc

Before this update, a merge between two group entries could fail due to a too-small internal buffer on systems where Name Service Switch (NSS) merged groups from more than two sources. In such cases, glibc incorrectly skipped the merge instead of retrying with a larger buffer. As a consequence, in some cases, querying group membership produced incomplete or empty results in environments with multiple group databases.

With this update, glibc correctly handles merge failures and retries with an appropriately sized buffer instead of skipping the result. As a result, group membership queries reliably return the full set of members when groups are merged from more than two services.

Jira:RHEL-114264[1]

glibc audit logging provides complete object life cycle tracking

Before this update, the glibc dynamic linker called **la_objclose** for the proxy **ld.so** link map in a secondary namespace without a preceding **la_objopen**. This resulted in incomplete object life cycle reporting for tools that rely on **la_objopen** to track shared objects.

As a consequence, auditing tools that rely on **la_objopen** to establish tracking failed to monitor proxy link maps reliably, resulting in gaps in visibility and possible misinterpretation of unload events.

With this release, the glibc dynamic linker generates **la_objopen** events for all applicable link maps, including the proxy **ld.so** in secondary namespaces, ensuring a consistent sequence for the auditing interface.

As a result, audit tools can track proxy link maps throughout their complete life cycle with consistent **la_objopen** and **la_objclose** event pairs, improving the reliability of audit tools and diagnostics.

Jira:RHEL-109693

12.12. IDENTITY MANAGEMENT

ipa-cacert-manage install now permits duplicate CA subjects

Previously, attempting to add a CA certificate with an identical subject but a different private key by using **ipa-cacert-manage install** failed with the message **subject public key info mismatch**, as IdM prohibited duplicate subjects.

This update relaxes that restriction, allowing **ipa-cacert-manage install** to accept duplicate CA subjects. However, the following limitations remain:

- Certificates cannot be added with different trust flags.
- The CAs must share the same nickname.
- An Authority Key Identifier (AKI) extension is mandatory for all CAs. Its absence leads to an unexpected chain of trust behavior.

Jira:RHEL-84648[1]

dsconf replication get-ruv no longer returns an error

Before this update, one of the replication functions did not call a required function. As a result, when you ran **dsconf <instance_name> replication get-ruv --suffix dc=example,dc=com**, an error was displayed. With this update, the command returns a Replica Update Vector (RUV) value as expected.

Jira:RHEL-112722

Newly created user password policies are displayed correctly

Before this update, the **cosAttribute** attribute in the Class of Service (CoS) template had the **operational** modifier instead of **operational-default**. As a consequence, when both subtree and user password policies existed, the **pwdpolicysubentry** attribute pointed to the subtree password policy instead of the user password policy. With this release, the CoS template uses the **operational-default** modifier. As a result, the user policy is displayed correctly.

> NOTE
>
> This issue affected only displaying the policies, not the actual password policy logic.

Jira:RHEL-97565

ipa-healthcheck now ignores the **replica busy** condition

Before this update, in a topology with more than two suppliers, the **ipa-healthcheck** tool reported an error about replication agreement status when a supplier was receiving updates from another node. It is a standard replication situation and, with this release, **ipa-healthcheck** no longer reports an error when replicas are busy.

Jira:RHEL-89774[1]

Directory Server no longer fails during cleanup at shutdown on instance with LMDB

Before this update, a race condition occurred during cleanup at shutdown on an instance with Lightning Memory-Mapped Database Manager (LMDB). With this update, Directory Server no longer calls **lmdb** when the database environment is closed.

Jira:RHEL-86878

LMDB monitoring statistics are now displayed correctly

Before this update, when you tried to retrieve the monitoring statistics on an instance with Lightning Memory-Mapped Database Manager (LMDB) database type, a key error occurred. With this update, Directory Server ensures backend and monitor keys match the configured database implementation. As a result, global monitoring statistics are displayed correctly.

Jira:RHEL-83850

389-ds-base no longer fails during the LMDB offline import

Before this update, a race condition occurred when a worker thread read an entry before another process finished writing the entry. As a result, offline import on an instance with the Lightning Memory-Mapped Database Manager (LMDB) backend caused a segmentation fault.

With this update, Directory Server ensures thread-safe access by locking the worker queue before writing entries, and the server no longer fails during the LMDB offline import.

Jira:RHEL-5117

The Directory Server web console now shows the server version

Before this update, the web console did not display the server version in the **Server Settings>General Settings**. With this update, the server version is displayed correctly.

Jira:RHEL-101783[1]

Directory Server correctly displays the number of child entries under a specific node

Before this update, the **numSubordinates** and **numTombstoneSubordinates** attributes were wrongly computed during import. Consequently, when you compared the number of child entries under a specific node, the wrong values were displayed.

With this update, Directory Server computes **numSubordinates** and **numTombstoneSubordinates** correctly.

Jira:RHEL-101727

Directory Server no longer fails during NDN cache operations

Before this update, the **arc-swap** library, which was used in the Rust dependency of **389-ds-base**, could cause a failure in Directory Server during NDN cache operations. With this release, Directory Server uses an updated version of Rust dependency (concread) 0.5.7 that does not contain the **arc-swap** library. As a result, Directory Server no longer fails.

Jira:RHEL-95441

Directory Server correctly displays membership in nested groups

Before this update, Directory Server displayed an incorrect value of the **memberOf** attribute in that entry under the following conditions:

- An entry was a member of groups that had multiple nested levels
- Groups were part of other different groups that had multiple paths in the membership relations.

With this update, the **memberOf** distinguished name (DN) value is added systematically, and the entry membership in groups is displayed correctly.

Jira:RHEL-89748

Directory Server no longer fails when adding **nsslapd-referral**

Before this update, when you tried to configure Directory Server to use a referral, the server failed due to incorrect handling of the paged search result.

With this update, If the search result code is **LDAP_REFERRAL**, the paged search result returns the correct value and the server no longer fails.

Jira:RHEL-87352

The RootDN Access Control plugin with wildcards for IP addresses no longer fails

Before this update, if you tried to set IP addresses with wildcards for the RootDN Access Control plugin configuration, the attempt failed with the **Invalid IP address** error. With this release, the validation function was updated. As a result, the attempt to set values with wildcards no longer fails.

Jira:RHEL-86313

The Directory Server monitoring information is available as expected when NDN cache is disabled

Before this update, when the Normalized DN (NDN) cache was disabled, the **dsconf *<instance_name>* monitor dbmon** command failed with an error because of improper handling of the backend get-tree command failures. This release adds a rollback functionality to prevent orphaned backends when the tree creation fails during a backend creation. As a result, Directory Server monitoring information is returned as expected.

Jira:RHEL-79079

The Databases menu opens as expected in the Directory Server web console

Before this update, you could not open the **Databases** menu in the Directory Server web console if the database name that you created had an incorrect suffix syntax, for example, the name included **dc=**. With this update, Directory Server uses a rollback functionality when mapping tree creation fails during backend creation to prevent orphaned backends. As a result, the **Databases** menu opens as expected.

Jira:RHEL-76832[1]

NDN cache no longer causes increased memory consumption in Directory Server

Before this update, the concread Rust dependency of **389-ds-base** allowed the Normalized DN (NDN) cache to hold the memory even of the evicted entries. As a consequence, NDN cache could increase memory consumption.

With this update, Directory Server uses an updated version of concread Rust dependency and NDN cache works as expected without the server performance impact.

Jira:RHEL-74085

Password modify extended operation skips password policy checks correctly for the root DN and password administrators

Before this update, when the root DN or a password administrator used a password modify extended operation to change a password, they could not bypass Directory Server's password policies restrictions. As a consequence, they could not update passwords that did not comply with password policy requirements.

With this release, the password policies are checked correctly when the Bind DN is the root DN or a password administrator. As a result, the root DN and password administrators can successfully update passwords without policy restrictions.

Jira:RHEL-67022

dsconf correctly returns replication monitoring information

Before this update, if a supplier was configured with a replica starting with **0**, such as **010** or **020**, the **dsconf <instance_name> replication monitor** command failed to retrieve information about time of a delay or the replication status.

With this update, non-significant zeros (**0**) at the beginning of replica ID are ignored while processing the replica ID within the replica update vector (RUV). As a result, **dsconf <instance_name> replication monitor** provides the expected information.

Jira:RHEL-67003

The error log in **389-ds-base** now contains full message about replication

Before this update, when you configured replication, the error log file contained incomplete messages about replication. With this release, the error log contains full messages with the actual values.

Jira:RHEL-61327

dsctl healthcheck no longer reports an incorrect database type on instances with LMDB

With this update, the correct configuration parameter is checked, and **dsctl healthcheck** no longer reports the wrong database type.

Jira:RHELDOCS-19014[1]

12.13. SSSD

Unprivileged processes can now renew host keytabs

Before this update, unprivileged processes lacked the ability to renew host keytab because the keytab file was only accessible by the **root** user. This issue prevented unprivileged processes from renewing their host keytab. With the release of the RHBA-2025:21019 advisory, **realmd** supports renewing the host keytab with appropriate policy-kit settings for unprivileged processes. As a result, unprivileged processes and users can now renew host keytab with ease.

Jira:RHEL-117645

12.14. RED HAT ENTERPRISE LINUX SYSTEM ROLES

Specifying multiple users no longer causes resources to be associated with the wrong user

Previously, when managing resources for two different users, both **vars** and **set_fact** were used to set the **__podman_user** and **__podman_user_home_dir** variables. This led to unpredictable and undefined behavior as the system used the old values from the first user for the second user, causing the second user's configuration to incorrectly reference the first user's data.

With this fix, the role sets the **podman_user** variable only with **set_fact**, and the **__podman_user_home_dir** variable only with **vars**. Also, the code has been refactored to use **__podman_handle_user** instead of **__podman_user** where the role could use **vars**. As a result, you keep data for multiple users separate and ensure consistent configurations.

Jira:RHEL-105093

The **postfix** RHEL system role auto-detects if an IPv6 interface is disabled

The default **postfix** configuration uses the **inet_interfaces = localhost** setting which tells **postfix** to listen on all interfaces resolving to **localhost** including both IPv4 and IPv6 interfaces. Before this update, a problem occurred if IPv6 was disabled on the host. In this situation, the **postfix** role and its command-line tools, such as **postconf**, returned an error. The entire role failed. With this release, the role determines if IPv6 is disabled. If so, then it sets **inet_protocols = ipv4** so that **postfix** only uses the IPv4 interface. As a result, the **postfix** role works even when IPv6 is disabled.

Jira:RHEL-103887

selinux role no longer produces error due to undefined **tempdir** path in Ansible check mode

Before this update, the **tempdir** path was not defined in Ansible check mode, and the **__selinux_item.path** could be undefined. Consequently, when running in check mode, the **selinux** RHEL system role produced an error that various variables are undefined. With this update, the role skips tasks that require the **tempdir.path** to be defined, and can handle cases where variables are undefined. As a result, the role works correctly in check mode.

Jira:RHEL-103573

Improved removal of kernel options with values in **rhel-system-roles**

Previously, kernel boot options specified as key=value could not be removed when users provided only the key, resulting in persistent unwanted boot parameters and inconsistent management of kernel options by name. With this update, the regular expression in the **mod_boot_args** function was updated to match and remove kernel options with values correctly, and automated tests were added to verify correct behavior.

As a result, kernel options can now be reliably removed by name, even when set as key=value, ensuring accurate configuration and improved system management.

Jira:RHEL-101676

Ensures **/var/lib/pcsd** directory is available when needed by the **ha_cluster** RHEL system role

Before this update, the **/var/lib/pcsd** directory was created during the installation of **pcs**, but newer versions rely on the **systemd** service to create this directory when the **pcsd** service starts. As a result, the directory might not exist at the time the role attempts to access it, causing errors or failures in execution.

With this update, the role explicitly ensures that the **/var/lib/pcsd** directory exists before using it. As a result, it prevents runtime issues due to the missing directory and improving the reliability of role execution.

Jira:RHEL-100819[1]

LVM RAID now supports encrypted and partitioned devices

Before this update, the LVM RAID code assumed that disks specified in **raid_disks** were the parent devices of the PVs for all LVM RAID setups. This was not applicable for encrypted or partitioned devices. As a consequence, errors occurred when encrypted LUKS layers added an extra storage layer, or when direct partitions were used without a parent device. With this release, PV resolution in LVM RAID is improved to support encrypted and partitioned devices. As a result, you can now specify the PV partition instead of the underlying disk.

This fix also adds error handling for missing or invalid RAID disk entries and introduces corresponding tests to ensure stability.

Jira:RHEL-95883

RAID now reports clear errors for invalid or unsupported configurations

Before this update, invalid RAID levels or insufficient disks could be specified without raising clear errors. This resulted in failed or inconsistent array creation. As a consequence, the error messages were unclear, and RAID setup was less reliable. With this release, RAID parameters are validated before array creation, and a minimum disk count is enforced. As a result, clear errors are raised, and attempts to create a RAID with inadequate disks are blocked.

This fix also removes the deprecated **process_device_numbers** helper and uses **unify_raid_level** instead. In addition, failure tests for invalid RAID level and insufficient disks scenarios are also added.

Jira:RHEL-95757

encryption_key is no longer masked

Before this update, the **encryption_key** parameter was incorrectly marked as **no_log**. This caused the key file path to be replaced by a placeholder string, preventing disk encryption from working. With this update, the **encryption_key** parameter is no longer marked with the **no_log** flag, and you can now perform disk encryption by using a key file successfully.

Jira:RHEL-95729

selinux role persistently sets kernel SELinux parameters

Before this update, the **selinux** RHEL system role did not set the kernel SELinux parameter when changing the SELinux state to and from disabled. As a consequence, the SELinux state change was not persistent upon reboot. This update ensures that the kernel SELinux parameter is correctly set when the role changes SELinux state to and from disabled. As a result, the SELinux state change to and from disabled is persistent upon reboot.

Jira:RHEL-93294

The systemd role uses file basename to construct the path to the destination

Before this update, if a user specified a file or a template source within a nested directory, the **systemd** RHEL system role used the whole path instead of the basename for the destination file. As a consequence, files and templates were placed in the same directory structure on the destination, which

systemd does not support. With this release, the role uses basenames for destination files in nested directories. As a result, users can use nested directories with the role.

Jira:RHEL-88774[1]

The **timesync** RHEL system role no longer removes the **OPTIONS="-F 2"** default setting from **/etc/sysconfig/chronyd**

Before this update, the **timesync** system role replaced the default **OPTIONS=** setting for the **chronyd** service with **""**. As a consequence, this removed the default **OPTIONS="-F 2"** setting which weakened the security of **chronyd**. With this release, **-F 2** is added as the default setting for **OPTIONS**, and the user can override or extend this setting. As a result, the **timesync** role now applies the correct security settings while still allowing user customization.

Jira:RHEL-88297

The **network** RHEL system role no longer shows errors due to incorrect routing rule validation

Before this update, the validation part in the **network** RHEL system role incorrectly checked for routing rule attributes at the top-level **NM** module instead of the **NM.IPRoutingRule** class. This caused validation failures and the role displayed errors. With this update, the role uses the API correctly and no longer shows incorrect validation errors.

Jira:RHEL-88286[1]

The **network** RHEL system role now uses a more robust interface identification method

Before this update, when both an interface name and a MAC address were provided for a network interface, the validation process performed two separate lookups: one using the interface name and another using the MAC address. This could lead to validation failures because a lookup by MAC address might match the interface's current MAC address rather than its permanent hardware MAC address.

With this update, the validation logic has been improved. The network role now uses the interface name as the only identifier to look up the network device. It then retrieves the MAC address associated with that interface and compares it to the user-provided MAC address for validation. This approach is more reliable, because interface names are unique kernel identifiers, preventing mismatches caused by temporary MAC address changes.

Jira:RHEL-88263[1]

The **qdevice** daemon now restarts automatically after certificate changes

Previously, after updating the TLS certificates used for communication between the quorum device daemon (**qnetd**) and the cluster nodes (**qdevice**), the **qdevice** daemon was not automatically restarted. The daemon would continue to use the old certificates, causing communication with the quorum device to fail.

With this update, the **qdevice** daemon on cluster nodes automatically restarts after its certificates are changed. This ensures that the new certificates are loaded immediately and that communication with the quorum device is maintained.

Jira:RHEL-88249

Boolean values are correctly rendered in TOML files

Before this update, boolean values in TOML files were incorrectly formatted, causing improper handling of boolean options. As a consequence, users experienced configuration issues. With this release, the format of boolean options in TOML files has been corrected. As a result, end users can now correctly configure boolean options in their TOML files.

Jira:RHEL-85704[1]

Boolean values are correctly rendered in TOML files

Before this update, incorrect boolean conversion in a Jinja2 template caused **True** to be written as **"True"**. As a consequence, users received an error due to incorrectly formatted configuration file, causing a container service failure. With this release, improper boolean conversion in a Jinja2 template has been fixed. As a result, Podman configuration now correctly converts boolean values in a Jinja2 template.

Jira:RHEL-84942[1]

podman RHEL system role no longer fails with UNREACHABLE errors when removing resources

Before this update, when disabling linger for non-root users, the system did not wait long enough for the user state to change to **closing**. As a result, the **systemd-logind** service was restarted prematurely to force the linger state to be canceled. On some systems, this triggered a timer that terminated the root session, including the active sshd connection. This caused the Ansible Playbook to fail with an **UNREACHABLE** error. With this release, the system waits significantly longer for linger to be properly canceled, and **systemd-logind** is restarted only if absolutely necessary. As a result, the role no longer fails with **UNREACHABLE** errors when removing resources

Jira:RHEL-84912[1]

The ha_cluster RHEL System Role now works with a system-wide HTTP proxy configured

Previously, when a system-wide HTTP proxy was configured, the **ha_cluster** RHEL System Role would incorrectly attempt to use the proxy for local communication with the **pcsd** daemon via a UNIX socket. This caused the role to fail.

With this release, the role has been modified to explicitly disable proxy usage for local **pcsd** communication.

As a result, the **ha_cluster** RHEL System Role works as expected on systems with a system-wide HTTP proxy defined.

Jira:RHEL-81918

GSSAPIIndicators added to sshd role

A new configuration option **GSSAPIIndicators** for setting Generic Security Services Application Programming Interface (GSS-API) was added to RHEL 10. This update adds the **GSSAPIIndicators** configuration option to the **sshd** RHEL system role. As a result, you can configure **GSSAPIIndicators** on RHEL 10 systems by using RHEL system roles.

Jira:RHEL-107047

bootloader role rejects boolean or null type values

Before this update, the user could specify values such as **value: on** or **value: yes** expecting that these would be converted to strings **"on"** or **"yes"**. But instead, YAML treats these as YAML bool type and writes them as the string **"True"**. Consequently, users who were unaware of YAML boolean handling could not set values such as **"on"** or **"off"**. With this update, the **bootloader** RHEL system role rejects any value of boolean or **null** type. As a result, users must enter such YAML boolean type values as quoted strings to write them to the boot loader configuration. The readme file is updated with this information.

Jira:RHEL-107013

sudo role no longer hangs when parsing Alias values

Before this update, the regular expression in the **sudo** RHEL system role was not taking into consideration that Alias values, such as **Cmnd_Alias**, do not have to have spaces on either side of the equal sign **=**. Consequently, the regular expression never terminated, and the role appeared to hang. With this update, the role ensures that the regular expression complies with the eBNF definition of the field from the **sudoers** file specification. As a result, the Alias values are parsed correctly with and without spaces around **=**.

Jira:RHEL-106261[1]

The podman RHEL system role does not report changed: true when managing authentication and configuration files

Before this update, the **podman** RHEL system role changed the parent path mode every time it ran if it managed both authentication and configuration files because it used two different modes for the common parent path for various configuration and authentication files.

With this fix, the role does not report **changed: true** unnecessarily because it uses a consistent mode for the parent path.

Jira:RHEL-84922[1]

The systemd role unmasks and starts units in a single run

Before this update, the **systemd** RHEL system role failed to enable and start services when units were masked because the role could not unmask the units first. As a result, users had to run the role twice. With this release, the **systemd** role correctly unmasks and starts services, eliminating the need for double runs.

Jira:RHEL-88760[1]

Minor volume size mismatch no longer cause incorrect role reporting

Before this update, when creating or resizing volumes, the system allowed up to a 2% difference between the requested size and the actual size. This adjustment made the volume fit into the available pool free space. As a consequence, the sizes did not match when the role was run again, causing the role to incorrectly assume that something had changed. With this release, small size differences no longer cause the role to misinterpret changes. As a result the role now reports the correct state.

Jira:RHEL-90216[1]

12.15. VIRTUALIZATION

Local kdump no longer fails on virtual machines with AMD SEV-SNP

Before this update, local kdump failed on RHEL 10 virtual machines (VMs) that used the AMD Secure Encrypted Virtualization (SEV) with the Secure Nested Paging (SNP) feature. As a consequence, you could not capture kernel crash dumps on VMs with AMD SEV-SNP enabled.

With this release, the underlying code has been fixed. As a result, local kdump no longer fails on VMs with AMD SEV-SNP.

Jira:RHEL-67539[1]

The **--migrate-disks-detect-zeroes** option no longer fails for VM migration

Before this update, when migrating virtual machines (VMs) on RHEL 10, the **--migrate-disks-detect-zeroes** option might not have worked, and the migration might have proceeded without zeroed block detection on the specified disk. This problem was caused by a bug in QEMU where mirroring jobs relied on punching holes, resulting in a sparse destination file.

With this release, QEMU has been fixed to preserve sparseness if the destination system reports that it reads all zeroes, and when no extra effort is made to further sparsify the image. As a result, the **--migrate-disks-detect-zeroes** option works as expected for VM migration.

Jira:RHEL-88435

VMs sending misaligned discard I/O requests no longer pause when **discard_granularity** is not configured

Before this update, the host kernel failed misaligned discard I/O requests and QEMU used the **werror= policy** parameter to respond to such failures. When **werror** was set to **stop**: **werror=stop**, a failed discard request caused the virtual machine (VM) to pause. As a consequence, it was not possible to correct this situation and resume the VM again.

With this release, QEMU has been updated to silently ignore misaligned discard I/O requests, so that guests without a correct discard_granularity value do not pause. As a result, VMs sending discard I/O requests no longer pause when **discard_granularity** is not configured. However, it is still preferable to configure the **discard_granularity** value, so that discard requests have their intended effect instead of being ignored when misaligned.

Jira:RHEL-87642[1]

virtiofsd no longer crashes when accessing shared directories with many open files

Before this update, when accessing a **virtiofs** shared directory with a large number of open files from a virtual machine (VM), the operation might have failed with the following error: **Too many open files**, and the **virtiofsd** process crashed.

With this release, the underlying code has been fixed. As a result, accessing a **virtiofs** shared directory with a large number of open files from a VM might still result in an error in the VM, but the **virtiofsd** process no longer crashes, keeping the **virtiofs** shared directory accessible in the VM.

Jira:RHEL-87161[1]

QEMU no longer prevents using SEV-SNP

Previously, when attempting to start a virtual machine (VM) with AMD SEV-SNP enabled, QEMU checked the incorrect capability of KVM, and the guest failed to start. As a consequence, running VMs with AMD SEV-SNP configured was not possible with RHEL10. This problem has been fixed, and running VMs with SEV-SNP works as expected now.

Jira:RHEL-58928[1]

Network boot for VMs now works correctly without an RNG device

Previously, when a virtual machine (VM) did not have an RNG device configured and its CPU model did not support the RDRAND feature, it was not possible to boot the VM from the network. With this update, the problem has been fixed, and VMs that do not support RDRAND can boot from the network even without an RNG device configured.

Note, however, that adding an RNG device is highly encouraged for VMs that use a CPU model that does not support RDRAND, in order to increase security when booting from the network.

Jira:RHEL-66234

RHEL 10 guests no longer crash on restart in Google Cloud and Alibaba

When using a RHEL 10.0 instance on Google Cloud or the Alibaba Cloud, restarting the instance previously caused a kernel panic in the guest operating system if the **virtio-net** driver was in use. This issue has been fixed and RHEL 10 guests no longer crash in the described scenario.

Jira:RHEL-56981[1]

Secure Execution VMs can now boot with file-backed memory backing

Previously, if you configured a virtual machine (VM) with enabled Secure Execution to use file-backed memory backing, the VM failed to boot, and instead displayed a **Protected boot has failed** error. Now, the VM boots as expected.

Jira:RHEL-58218

Enabling 3D support no longer prevents installing a RHEL 10 guest on ESXi

Before this update, if you selected the Enable 3D support option in VMware ESXI for installing a RHEL 10 guest operating system, the installation did not start correctly, and instead showed a blank screen. This issue has been fixed, and you can now install RHEL 10 guests in the described scenario.

Jira:RHEL-88668[1]

12.16. RHEL IN CLOUD ENVIRONMENTS

Nested VM with KVM virtualization and OVMF now boots successfully on Azure or Hyper-V when using an AMD EPYC processor

Previously, a nested virtual machine (VM) with Open Virtual Machine Firmware (OVMF) failed to boot when run on a RHEL VM with KVM virtualization enabled on Microsoft Azure or Hyper-V that used an AMD EPYC processor. The VM failed to boot up with following log message:

> Code=qemu-kvm: ../hw/core/cpu-sysemu.c:76 Aborted (core dumped) .

With this update, the problem has been fixed, and the nested VM boots as expected in the described circumstances.

Jira:RHEL-29919[1]

12.17. SUPPORTABILITY

The **coredump** plugin now correctly limits the number of collected **coredump** files

Previously, the **coredump** plugin collected **coredumpctl dump** outputs, which could lead to unnecessary large archives. With this update, the plugin defaults to collecting the three most recent **coredump** files. Additionally, the plugin continues to provide summary information from **coredumpctl info** and includes symlinks to help map collected dumps to their metadata entries.

Users can further filter collected dumps by using the **executable** option, which accepts a case-insensitive Python regular expression applied to the EXE field of **coredumpctl list**. You can further use the **dumps** option to limit the number of last coredumps.

Jira:RHEL-62972[1]

Plugin option overrides in **sos report** no longer disable unrelated options configured in **/etc/sos/sos.conf** or a preset

Previously, when executing the **sos report** command with a **-k** option specifying a particular plugin setting , the **sos** utility would incorrectly ignore other valid plugin options defined in **/etc/sos/sos.conf** or in a preset. This led to scenarios where global settings or user-defined presets, were silently disabled despite being correctly configured in the **[plugin_options]** section of the configuration file or in a preset.

This behavior affected customers attempting to collect full System Activity Reporter (SAR) data as outlined in Red Hat Knowledgebase Solution 1418303 . When any **-k** option was used at runtime, the **sar.all_sar** setting reverted to **off**, resulting in incomplete data collection.

With this update, the **sos** tool now correctly merges options provided via the **-k** flag with those defined in the configuration file, ensuring that unrelated plugin options are preserved and applied as expected. This fix restores consistency and ensures comprehensive SAR data collection when configured.

Jira:RHEL-67097[1]

sos-audit package now includes required GPLv2 LICENSE file

Previously, while the **sos-audit** package was always part of the **sos** project and built from the same SRPM containing the license, the resulting **sos-audit** RPM package could be installed separately from the main **sos** RPM. This meant users installing only the **sos-audit** subpackage would not find the license readily available. This omission affected all versions of **sos-audit** up to the current release across RHEL 8 and RHEL 9.

With this update, the **sos-audit** package now correctly includes the GPLv2 **LICENSE** file.

Jira:RHEL-73028

iscsi plugin no longer collects plain text CHAP credentials in sosreport

Previously, the **iscsi** plugin in **sos** collected sensitive CHAP authentication credentials in **iscsi** configuration files in plain text when generating a report that posed a security risk. With this update, the **iscsi** plugin has been modified to obscure sensitive fields, ensuring that CHAP usernames and passwords are redacted or excluded from the collected output.

Jira:RHEL-81187[1]

THP plugin now collects complete configuration to accurately reflect Transparent Huge Pages state

Previously, the memory plugin of **sos** collected only the **enabled** file from **/sys/kernel/mm/transparent_hugepage/** to determine the state of Transparent Huge Pages (THP). However, recent kernel behavior changes have made this approach insufficient. For example, it is possible for **enabled** to be set to **[never]** while **shmem_enabled** is set to **[always]**, resulting in THP being active for shared memory segments despite appearing disabled.

With this update, the THP plugin now collects all relevant files under **/sys/kernel/mm/transparent_hugepage/**, providing a complete and accurate view of how and where THP is enabled.

Jira:RHEL-81634[1]

per-user SSH configuration is now disabled by default

Previously, the **ssh** plugin in **sos** collected detailed information from all local user **.ssh** directories by default. This resulted in significantly prolonged execution time, especially in environments with a large number of local users. With this update, the **ssh** plugin no longer collects per-user **.ssh** configuration data by default. To capture user configurations, enable it explicitly by setting **ssh.userconfs=on**.

Jira:RHEL-84078

sos collect command in the sos 4.10 version no longer produces xz/bz2 tar archive

Before this update, the **sos collect** command returned a compressed tar archive, such as **tar.xz** or **tar.bz2**. With this release, the **sos collect** now produces uncompressed **tar** archives instead of compressed ones, saving time and resources.

Jira:RHELDOCS-21013[1]

12.18. CONTAINERS

Event logs from podman events command are now available

Previously, an error in the **journald** driver prevented the preservation of network event attributes, so these events were not included in logs. With this update, **podman events** now displays **network create** and **network rm** events.

Jira:RHEL-110318

You can now set /sys/fs/cgroup/io.max within the container

Before this update, when using **runc** as the container runtime, you could not set **/sys/fs/cgroup/io.max** inside the container. With this fix, the issue is resolved, and the value of **/sys/fs/cgroup/io.max** now matches in the **podman update** command.

Jira:RHEL-81042[1]

Parent directories can be created now for the mount targets with mode 0755

In this update, build failures were occurring due to modifications in the handling of **--mount** parameter permissions in **quay.io/buildah/stable:v1 v1.41.3**. Previously, specifying UID as an argument resulted in incorrect permissions for the secret. Consequently, users were unable to access build secrets due to incorrect permissions after the **buildah** update.

With this release, Buildah has updated secret permissions for Buildah v1.41.3, using **secret-permissions** instead of **mount**. As a result, Buildah now sets the expected permissions for secrets correctly when using the UID argument in the **--mount** parameter, resolving mount failures.

Jira:RHEL-115167

12.19. RHEL LIGHTSPEED

Command-line assistant shows a meaningful error message when you try to delete a non-existent chat history

Before this update, users could delete a non-existent chat history without receiving an error message. This enhancement implements an error message for such cases.

Jira:RHELDOCS-21314[1]

Adding a description to an unnamed chat triggers a warning

Before this update, if you added a description to a chat without specifying a name for the chat, there was no error message displayed, nor was the chat with your custom description. With this update, the command-line assistant displays a warning in such cases.

Jira:RHELDOCS-21316[1]

c history shows complete history by default

Before this update, running the **c history** command without any options returned no history, confusing users. With this update, the default option for **--all** has been added. As a result, you can easily view all history with the single command: **c history**.

Jira:RHELDOCS-21317[1]

Command-line assistant no longer displays errors for invalid queries

Before this update, an incorrect data structure for terminal output in response led to unprocessable error messages for user queries. With this enhancement, the chat interface’s terminal output structure has been actively addressed, preventing the command-line assistant from displaying errors for invalid query requests, thereby enhancing your user experience.

Jira:RHELDOCS-21318[1]

Interactive shell starts correctly after a terminal restart

Before this update, the user’s **.bashrc** file did not include a reference to the **.bashrc.d** directory, preventing the **source** command from locating the CLA integration script. As a consequence, users could not access an interactive shell. With this update, a check has been added to ensure that the files necessary for shell integration are loaded. As a result, the interactive shell starts upon terminal restart.

Jira:RHELDOCS-21319[1]

Backend timeout works correctly in query.py

Before this update, extending the backend timeout in the **query.py** script did not work correctly. The script continued to generate timeout messages every 30 seconds because an internal timeout remained set at 30 seconds by default. With this enhancement, you can extend the backend timeout to any value that suits you by configuring this in the **/etc/xdg/command-line-assistant/config.toml** file, improving your response time.

Jira:RHELDOCS-21320[1]

cla chat displays help when run without arguments

Before this update, using **cla chat** without providing additional input caused user confusion, as they expected interactive AI assistance but received no response. With this update, when you use **cla chat** without arguments, the command-line assistant provides help and indicates additional input, improving your user experience with CLA’s interactive mode.

Jira:RHELDOCS-21322[1]

CHAPTER 13. AVAILABLE BPF FEATURES

This chapter provides the complete list of Berkeley Packet Filter (BPF) features available in the kernel of this minor version of Red Hat Enterprise Linux 10 The tables include the lists of:

- System configuration and other options
- Available program types and supported helpers
- Available map types

This chapter contains automatically generated output of the **bpftool feature** command.

Table 13.1. System configuration and other options

Option	Value
unprivileged_bpf_disabled	2 (bpf() syscall restricted to privileged users, admin can change)
JIT enable	1 (enabled)
JIT harden	1 (enabled for unprivileged users)
JIT kallsyms	1 (enabled for root)
Memory limit for JIT for unprivileged users	69267617742848
CONFIG_BPF	y
CONFIG_BPF_SYSCALL	y
CONFIG_HAVE_EBPF_JIT	y
CONFIG_BPF_JIT	y
CONFIG_BPF_JIT_ALWAYS_ON	y
CONFIG_DEBUG_INFO_BTF	y
CONFIG_DEBUG_INFO_BTF_MODULES	y
CONFIG_CGROUPS	y
CONFIG_CGROUP_BPF	y
CONFIG_CGROUP_NET_CLASSID	y
CONFIG_SOCK_CGROUP_DATA	y

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

Option	Value
CONFIG_TEST_BPF	m
CONFIG_HZ	100
bpf() syscall	available
Large insn size limit	available
Bounded loop support	available
ISA extension v2	available
ISA extension v3	available

Table 13.2. Available program types and supported helpers

Program type	Available helpers
socket_filter	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_perf_event_output, bpf_skb_load_bytes, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_get_socket_uid, bpf_skb_load_bytes_relative, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

Program type	Available helpers
kprobe	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task, bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str, bpf_perf_event_read_value, bpf_get_stack, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_send_signal, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_send_signal_thread, bpf_jiffies64, bpf_get_ns_current_pid_tgid, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_get_task_stack, bpf_copy_from_user, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_get_attach_cookie, bpf_task_pt_regs, bpf_get_branch_snapshot, bpf_find_vma, bpf_loop, bpf_strncmp, bpf_copy_from_user_task, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

Program type	Available helpers
sched_cls	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_skb_store_bytes, bpf_l3_csum_replace, bpf_l4_csum_replace, bpf_tail_call, bpf_clone_redirect, bpf_get_cgroup_classid, bpf_skb_vlan_push, bpf_skb_vlan_pop, bpf_skb_get_tunnel_key, bpf_skb_set_tunnel_key, bpf_redirect, bpf_get_route_realm, bpf_perf_event_output, bpf_skb_load_bytes, bpf_csum_diff, bpf_skb_get_tunnel_opt, bpf_skb_set_tunnel_opt, bpf_skb_change_proto, bpf_skb_change_type, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_skb_change_tail, bpf_skb_pull_data, bpf_csum_update, bpf_set_hash_invalid, bpf_get_numa_node_id, bpf_skb_change_head, bpf_get_socket_cookie, bpf_get_socket_uid, bpf_set_hash, bpf_skb_adjust_room, bpf_skb_get_xfrm_state, bpf_skb_load_bytes_relative, bpf_fib_lookup, bpf_skb_cgroup_id, bpf_get_current_cgroup_id, bpf_skb_ancestor_cgroup_id, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_sk_fullsock, bpf_tcp_sock, bpf_skb_ecn_set_ce, bpf_get_listener_sock, bpf_skc_lookup_tcp, bpf_tcp_check_syncookie, bpf_strtol, bpf_strtoul, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_tcp_gen_syncookie, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_sk_assign, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_csum_level, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_skb_cgroup_classid, bpf_redirect_neigh, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_redirect_peer, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_check_mtu, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_skb_set_tstamp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_tcp_raw_gen_syncookie_ipv4, bpf_tcp_raw_gen_syncookie_ipv6, bpf_tcp_raw_check_syncookie_ipv4, bpf_tcp_raw_check_syncookie_ipv6, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

Program type	Available helpers
sched_act	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_skb_store_bytes, bpf_l3_csum_replace, bpf_l4_csum_replace, bpf_tail_call, bpf_clone_redirect, bpf_get_cgroup_classid, bpf_skb_vlan_push, bpf_skb_vlan_pop, bpf_skb_get_tunnel_key, bpf_skb_set_tunnel_key, bpf_redirect, bpf_get_route_realm, bpf_perf_event_output, bpf_skb_load_bytes, bpf_csum_diff, bpf_skb_get_tunnel_opt, bpf_skb_set_tunnel_opt, bpf_skb_change_proto, bpf_skb_change_type, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_skb_change_tail, bpf_skb_pull_data, bpf_csum_update, bpf_set_hash_invalid, bpf_get_numa_node_id, bpf_skb_change_head, bpf_get_socket_cookie, bpf_get_socket_uid, bpf_set_hash, bpf_skb_adjust_room, bpf_skb_get_xfrm_state, bpf_skb_load_bytes_relative, bpf_fib_lookup, bpf_skb_cgroup_id, bpf_get_current_cgroup_id, bpf_skb_ancestor_cgroup_id, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_sk_fullsock, bpf_tcp_sock, bpf_skb_ecn_set_ce, bpf_get_listener_sock, bpf_skc_lookup_tcp, bpf_tcp_check_syncookie, bpf_strtol, bpf_strtoul, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_tcp_gen_syncookie, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_sk_assign, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_csum_level, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_skb_cgroup_classid, bpf_redirect_neigh, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_redirect_peer, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_check_mtu, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_skb_set_tstamp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_tcp_raw_gen_syncookie_ipv4, bpf_tcp_raw_gen_syncookie_ipv6, bpf_tcp_raw_check_syncookie_ipv4, bpf_tcp_raw_check_syncookie_ipv6, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

Program type	Available helpers
tracepoint	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task, bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str, bpf_perf_event_read_value, bpf_get_stack, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_send_signal, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_send_signal_thread, bpf_jiffies64, bpf_get_ns_current_pid_tgid, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_get_task_stack, bpf_copy_from_user, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_get_attach_cookie, bpf_task_pt_regs, bpf_get_branch_snapshot, bpf_find_vma, bpf_loop, bpf_strncmp, bpf_copy_from_user_task, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
xdp	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_redirect, bpf_perf_event_output, bpf_csum_diff, bpf_get_current_task, bpf_get_numa_node_id, bpf_xdp_adjust_head, bpf_redirect_map, bpf_xdp_adjust_meta, bpf_xdp_adjust_tail, bpf_fib_lookup, bpf_get_current_cgroup_id, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_skc_lookup_tcp, bpf_tcp_check_syncookie, bpf_strtol, bpf_strtoul, bpf_tcp_gen_syncookie, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_check_mtu, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_xdp_get_buff_len, bpf_xdp_load_bytes, bpf_xdp_store_bytes, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_tcp_raw_gen_syncookie_ipv4, bpf_tcp_raw_gen_syncookie_ipv6, bpf_tcp_raw_check_syncookie_ipv4, bpf_tcp_raw_check_syncookie_ipv6, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

Program type	Available helpers
perf_event	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task, bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str, bpf_perf_event_read_value, bpf_perf_prog_read_value, bpf_get_stack, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_send_signal, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_send_signal_thread, bpf_jiffies64, bpf_read_branch_records, bpf_get_ns_current_pid_tgid, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_get_task_stack, bpf_copy_from_user, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_get_attach_cookie, bpf_task_pt_regs, bpf_get_branch_snapshot, bpf_find_vma, bpf_loop, bpf_strncmp, bpf_copy_from_user_task, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
cgroup_skb	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_perf_event_output, bpf_skb_load_bytes, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_get_socket_uid, bpf_skb_load_bytes_relative, bpf_skb_cgroup_id, bpf_get_current_cgroup_id, bpf_get_local_storage, bpf_skb_ancestor_cgroup_id, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_sk_fullsock, bpf_tcp_sock, bpf_skb_ecn_set_ce, bpf_get_listener_sock, bpf_skc_lookup_tcp, bpf_strtol, bpf_strtoul, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_sk_cgroup_id, bpf_sk_ancestor_cgroup_id, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

Program type	Available helpers
cgroup_sock	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_get_cgroup_classid, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_get_current_cgroup_id, bpf_get_local_storage, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_sk_storage_get, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_netns_cookie, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_loop, bpf_strncmp, bpf_get_retval, bpf_set_retval, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
lwt_in	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_cgroup_classid, bpf_get_route_realm, bpf_perf_event_output, bpf_skb_load_bytes, bpf_csum_diff, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_skb_pull_data, bpf_get_numa_node_id, bpf_lwt_push_encap, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

Program type	Available helpers
lwt_out	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_cgroup_classid, bpf_get_route_realm, bpf_perf_event_output, bpf_skb_load_bytes, bpf_csum_diff, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_skb_pull_data, bpf_get_numa_node_id, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
lwt_xmit	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_skb_store_bytes, bpf_l3_csum_replace, bpf_l4_csum_replace, bpf_tail_call, bpf_clone_redirect, bpf_get_cgroup_classid, bpf_skb_get_tunnel_key, bpf_skb_set_tunnel_key, bpf_redirect, bpf_get_route_realm, bpf_perf_event_output, bpf_skb_load_bytes, bpf_csum_diff, bpf_skb_get_tunnel_opt, bpf_skb_set_tunnel_opt, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_skb_change_tail, bpf_skb_pull_data, bpf_csum_update, bpf_set_hash_invalid, bpf_get_numa_node_id, bpf_skb_change_head, bpf_lwt_push_encap, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_csum_level, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

Program type	Available helpers
sock_ops	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_setsockopt, bpf_sock_map_update, bpf_getsockopt, bpf_sock_ops_cb_flags_set, bpf_sock_hash_update, bpf_get_current_cgroup_id, bpf_get_local_storage, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_tcp_sock, bpf_strtol, bpf_strtoul, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_netns_cookie, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_load_hdr_opt, bpf_store_hdr_opt, bpf_reserve_hdr_opt, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
sk_skb	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_skb_store_bytes, bpf_tail_call, bpf_perf_event_output, bpf_skb_load_bytes, bpf_get_current_task, bpf_skb_change_tail, bpf_skb_pull_data, bpf_get_numa_node_id, bpf_skb_change_head, bpf_get_socket_cookie, bpf_get_socket_uid, bpf_skb_adjust_room, bpf_sk_redirect_map, bpf_sk_redirect_hash, bpf_get_current_cgroup_id, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_skc_lookup_tcp, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

Program type	Available helpers
cgroup_device	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_get_cgroup_classid, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_current_cgroup_id, bpf_get_local_storage, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
sk_msg	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_cgroup_classid, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_msg_redirect_map, bpf_msg_apply_bytes, bpf_msg_cork_bytes, bpf_msg_pull_data, bpf_msg_redirect_hash, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_msg_push_data, bpf_msg_pop_data, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_netns_cookie, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

Program type	Available helpers
raw_tracepoint	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task, bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str, bpf_perf_event_read_value, bpf_get_stack, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_send_signal, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_send_signal_thread, bpf_jiffies64, bpf_get_ns_current_pid_tgid, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_get_task_stack, bpf_copy_from_user, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_task_pt_regs, bpf_get_branch_snapshot, bpf_find_vma, bpf_loop, bpf_strncmp, bpf_copy_from_user_task, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
cgroup_sock_addr	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_get_cgroup_classid, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_setsockopt, bpf_getsockopt, bpf_bind, bpf_get_current_cgroup_id, bpf_get_local_storage, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_skc_lookup_tcp, bpf_strtol, bpf_strtoul, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_netns_cookie, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_get_retval, bpf_set_retval, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

Program type	Available helpers
lwt_seg6local	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_cgroup_classid, bpf_get_route_realm, bpf_perf_event_output, bpf_skb_load_bytes, bpf_csum_diff, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_skb_pull_data, bpf_get_numa_node_id, bpf_lwt_seg6_store_bytes, bpf_lwt_seg6_adjust_srh, bpf_lwt_seg6_action, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
lirc_mode2	not supported
sk_reuseport	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_skb_load_bytes, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_skb_load_bytes_relative, bpf_get_current_cgroup_id, bpf_sk_select_reuseport, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

Program type	Available helpers
flow_dissector	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_skb_load_bytes, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
cgroup_sysctl	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_get_cgroup_classid, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_current_cgroup_id, bpf_get_local_storage, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_sysctl_get_name, bpf_sysctl_get_current_value, bpf_sysctl_get_new_value, bpf_sysctl_set_new_value, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

Program type	Available helpers
raw_tracepoint_writable	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task, bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str, bpf_perf_event_read_value, bpf_get_stack, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_send_signal, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_send_signal_thread, bpf_jiffies64, bpf_get_ns_current_pid_tgid, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_get_task_stack, bpf_copy_from_user, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_task_pt_regs, bpf_get_branch_snapshot, bpf_find_vma, bpf_loop, bpf_strncmp, bpf_copy_from_user_task, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
cgroup_sockopt	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_get_cgroup_classid, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_current_cgroup_id, bpf_get_local_storage, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_tcp_sock, bpf_strtol, bpf_strtoul, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_netns_cookie, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_loop, bpf_strncmp, bpf_get_retval, bpf_set_retval, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
tracing	
struct_ops	
ext	
lsm	

Program type	Available helpers
sk_lookup	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_current_cgroup_id, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_sk_assign, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
syscall	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task, bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str, bpf_get_socket_cookie, bpf_perf_event_read_value, bpf_get_stack, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_send_signal, bpf_skb_output, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_send_signal_thread, bpf_jiffies64, bpf_get_ns_current_pid_tgid, bpf_xdp_output, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_get_task_stack, bpf_d_path, bpf_copy_from_user, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf, bpf_sock_from_file, bpf_for_each_map_elem, bpf_snprintf, bpf_sys_bpf, bpf_btf_find_by_name_kind, bpf_sys_close, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_task_pt_regs, bpf_get_branch_snapshot, bpf_skc_to_unix_sock, bpf_kallsyms_lookup_name, bpf_find_vma, bpf_loop, bpf_strncmp, bpf_xdp_get_buff_len, bpf_copy_from_user_task, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

Program type	Available helpers
netfilter	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

Table 13.3. Available map types

Map type	Available
hash	yes
array	yes
prog_array	yes
perf_event_array	yes
percpu_hash	yes
percpu_array	yes
stack_trace	yes
cgroup_array	yes
lru_hash	yes
lru_percpu_hash	yes
lpm_trie	yes
array_of_maps	yes
hash_of_maps	yes

Map type	Available
devmap	yes
sockmap	yes
cpumap	yes
xskmap	yes
sockhash	yes
cgroup_storage	yes
reuseport_sockarray	yes
percpu_cgroup_storage	yes
queue	yes
stack	yes
sk_storage	yes
devmap_hash	yes
struct_ops	yes
ringbuf	yes
inode_storage	yes
task_storage	yes
bloom_filter	yes
user_ringbuf	yes
cgrp_storage	yes
arena_map	yes

APPENDIX A. LIST OF TICKETS BY COMPONENT

Bugzilla and JIRA tickets are listed in this document for reference. The links lead to the release notes in this document that describe the tickets.

Component	Tickets
389-ds-base	Jira:RHEL-67006, Jira:RHEL-80162, Jira:RHEL-109018, Jira:RHEL-80252, Jira:RHEL-74270, Jira:RHEL-31959, Jira:RHEL-112722, Jira:RHEL-97565, Jira:RHEL-89774, Jira:RHEL-86878, Jira:RHEL-83850, Jira:RHEL-5117, Jira:RHEL-101783, Jira:RHEL-101727, Jira:RHEL-95441, Jira:RHEL-89748, Jira:RHEL-87352, Jira:RHEL-86313, Jira:RHEL-79079, Jira:RHEL-76832, Jira:RHEL-74085, Jira:RHEL-67022, Jira:RHEL-67003, Jira:RHEL-61327, Jira:RHEL-35241, Jira:RHEL-77490, Jira:RHEL-25071
NetworkManager	Jira:RHEL-89582, Jira:RHEL-83058, Jira:RHEL-81948, Jira:RHEL-70484, Jira:RHEL-59083, Jira:RHEL-5852
NetworkManager-libreswan	Jira:RHEL-34057
Release Notes	Jira:RHELDOCS-18640, Jira:RHELDOCS-21373, Jira:RHELDOCS-21377, Jira:RHELDOCS-21013, Jira:RHELDOCS-19016, Jira:RHELDOCS-20690, Jira:RHELDOCS-19072, Jira:RHELDOCS-19891, Jira:RHELDOCS-19968, Jira:RHELDOCS-20041, Jira:RHELDOCS-20042, Jira:RHELDOCS-20043, Jira:RHELDOCS-19635, Jira:RHELDOCS-20472, Jira:RHELDOCS-20167, Jira:RHELDOCS-20862, Jira:RHELDOCS-18080, Jira:RHELDOCS-19607, Jira:RHELDOCS-18674, Jira:RHELDOCS-18672, Jira:RHELDOCS-18450, Jira:RHELDOCS-20688, Jira:RHELDOCS-20689, Jira:RHELDOCS-20691, Jira:RHELDOCS-20692, Jira:RHELDOCS-20693, Jira:RHELDOCS-20147, Jira:RHELDOCS-20610, Jira:RHELDOCS-21532, Jira:RHELDOCS-16612, Jira:RHELDOCS-19015, Jira:RHELDOCS-19172, Jira:RHELDOCS-19603, Jira:RHELDOCS-19770, Jira:RHELDOCS-20686, Jira:RHELDOCS-20687, Jira:RHELDOCS-21410, Jira:RHELDOCS-21467, Jira:RHELDOCS-21153
anaconda	Jira:RHEL-87651, Jira:RHEL-88558, Jira:RHEL-80672, Jira:RHEL-67865, Jira:RHEL-74504, Jira:RHEL-69400, Jira:RHEL-83577, Jira:RHEL-66155, Jira:RHEL-58827, Jira:RHEL-58829, Jira:RHEL-58828
ansible-collection-microsoft-sql	Jira:RHEL-69315
ansible-core	Jira:RHEL-126844, Jira:RHEL-86829
audit	Jira:RHEL-77141
azure-vm-utils	Jira:RHEL-73904

Component	Tickets
bind	Jira:RHEL-33729
bootc-image-builder-container	Jira:RHEL-34807
buildah	Jira:RHEL-115167
chrony	Jira:RHEL-95017
cloud-init	Jira:RHEL-82209, Jira:RHEL-82210
cockpit	Jira:RHEL-87394, Jira:RHEL-92061, Jira:RHEL-4032
cockpit-machines	Jira:RHEL-31993
container-tools	Jira:RHEL-67860
coreutils	Jira:RHEL-74146
crash	Jira:RHEL-76107
crypto-policies	Jira:RHEL-113008, Jira:RHEL-93323, Jira:RHEL-99813, Jira:RHEL-64746, Jira:RHEL-112392
cups-browsed	Jira:RHEL-87180
device-mapper-multipath	Jira:RHEL-82180, Jira:RHEL-82535
distribution	Jira:RHEL-100678, Jira:RHEL-113198, Jira:RHEL-73904, Jira:RHEL-73770
dnf	Jira:RHEL-106141
dogtag-pki	Jira:RHEL-98721
dyninst	Jira:RHEL-87001
edk2	Jira:RHEL-50, Jira:RHEL-63645, Jira:RHEL-66234, Jira:RHEL-68418, Jira:RHEL-82685
elfutils	Jira:RHEL-86966
fapolicyd	Jira:RHEL-94540, Jira:RHEL-105425

Component	Tickets
fence-agents	Jira:RHEL-79799, Jira:RHEL-68322, Jira:RHEL-88569
fips-provider-next	Jira:RHEL-105014
gcc	Jira:RHEL-86679
gcc-toolset-15	Jira:RHEL-81745, Jira:RHEL-88743
gdb	Jira:RHEL-56897, Jira:RHEL-91382
glibc	Jira:RHEL-58357, Jira:RHEL-109625, Jira:RHEL-109621, Jira:RHEL-107695, Jira:RHEL-72564, Jira:RHEL-109536, Jira:RHEL-109703, Jira:RHEL-109702, Jira:RHEL-72018, Jira:RHEL-114264, Jira:RHEL-109693
gnutls	Jira:RHEL-102557, Jira:RHEL-64740, Jira:RHEL-85829, Jira:RHEL-102992
grafana-pcp	Jira:RHEL-77946
grub2	Jira:RHEL-4378
ipa	Jira:RHEL-67686, Jira:RHEL-84648, Jira:RHEL-67912, Jira:RHEL-12154
ipa-healthcheck	Jira:RHEL-84771
iproute	Jira:RHEL-90493, Jira:RHEL-99163
java-25-openjdk	Jira:RHEL-100678
kernel	Jira:RHELPLAN-114103
kernel / BPF	Jira:RHEL-78201
kernel / Core	Jira:RHEL-83042
kernel / Debugging-Tracing / Perf	Jira:RHEL-47451, Jira:RHEL-77936, Jira:RHEL-53584, Jira:RHEL-47443, Jira:RHEL-47423, Jira:RHEL-45094, Jira:RHEL-45092, Jira:RHEL-24184, Jira:RHEL-20109, Jira:RHEL-20093, Jira:RHEL-45090, Jira:RHEL-78197
kernel / Debugging-Tracing / rtla	Jira:RHEL-86520, Jira:RHEL-77357, Jira:RHEL-73865
kernel / Desktop / Graphics	Jira:RHEL-88668

Component	Tickets
kernel / Networking	Jira:RHEL-88891, Jira:RHEL-68401
kernel / Networking / NIC Drivers	Jira:RHEL-80554, Jira:RHEL-56981
kernel / Networking / Netfilter	Jira:RHEL-87758, Jira:RHEL-99656, Jira:RHEL-88574
kernel / Networking / kTLS	Jira:RHEL-86020
kernel / Other	Jira:RHEL-65347
kernel / RDMA	Jira:RHEL-86015
kernel / Storage / Storage Drivers	Jira:RHEL-85845
kernel / Virtualization	Jira:RHEL-76477
kernel / Virtualization / ESXi	Jira:RHEL-41133
kernel / Virtualization / Hyper-V	Jira:RHEL-75576, Jira:RHEL-29919
kernel / Virtualization / KVM	Jira:RHEL-52964, Jira:RHEL-67539, Jira:RHEL-100313, Jira:RHEL-58218, Jira:RHEL-32892, Jira:RHEL-45585, Jira:RHEL-38957
kernel-rt / Other	Jira:RHEL-62687
kpatch	Jira:RHEL-85686
libreswan	Jira:RHEL-102733
libslirp	Jira:RHEL-45147
libvirt	Jira:RHEL-58151, Jira:RHEL-7390, Jira:RHEL-81041, Jira:RHEL-71662
libvirt / General	Jira:RHEL-111863, Jira:RHEL-72994, Jira:RHEL-104382, Jira:RHEL-68043, Jira:RHEL-89426
libvirt / Live Migration	Jira:RHEL-20294
libvirt / Storage	Jira:RHEL-77552

Component	Tickets
llvm	Jira:RHEL-80988, Jira:RHEL-86089
lorax-templates-rhel	Jira:RHEL-91929, Jira:RHEL-101695
lvm2	Jira:RHEL-89832
mesa	Jira:RHEL-45898
mutter	Jira:RHEL-69291
nginx	Jira:RHEL-33742
nmstate	Jira:RHEL-84768, Jira:RHEL-84766, Jira:RHEL-80547, Jira:RHEL-80116, Jira:RHEL-78334, Jira:RHEL-1415
nodejs	Jira:RHEL-90826
nodejs24	Jira:RHEL-90826
nss	Jira:RHEL-103352, Jira:RHEL-64738, Jira:RHEL-114443
opencryptoki	Jira:RHEL-73343
openssh	Jira:RHEL-83644, Jira:RHEL-40790, Jira:RHEL-93957
openssl	Jira:RHEL-80811, Jira:RHEL-90853, Jira:RHEL-94614, Jira:RHEL-82676, Jira:RHEL-45704, Jira:RHEL-72719
openwsman	Jira:RHEL-93091
osbuild	Jira:RHEL-104075
pacemaker	Jira:RHEL-86146, Jira:RHEL-86144, Jira:RHEL-71181, Jira:RHEL-62722
pcs	Jira:RHEL-76176, Jira:RHEL-66607, Jira:RHEL-63186, Jira:RHEL-44347, Jira:RHEL-35407, Jira:RHEL-22423, Jira:RHEL-21050, Jira:RHEL-7681, Jira:RHEL-92043
pkcs11-provider	Jira:RHEL-68621
plymouth	Jira:RHEL-60198

Component	Tickets
podman	Jira:RHEL-88522, Jira:RHEL-88473, Jira:RHEL-88463, Jira:RHEL-88308, Jira:RHEL-27842, Jira:RHEL-110318, Jira:RHEL-81042, Jira:RHEL-32266, Jira:RHEL-70218, Jira:RHEL-89373, Jira:RHEL-88122, Jira:RHEL-40641
postgresql16-postgis	Jira:RHEL-81633
pykickstart	Jira:RHEL-34829
python-blivet	Jira:RHEL-84685, Jira:RHEL-84663, Jira:RHEL-53719
python-drgn	Jira:RHEL-86265
qemu-kvm	Jira:RHEL-88435, Jira:RHEL-87642, Jira:RHEL-58928, Jira:RHEL-67699, Jira:RHEL-66229
qemu-kvm / Devices / CPU Models	Jira:RHEL-28971
qemu-kvm / Networking	Jira:RHEL-45624
realmd	Jira:RHEL-117645
rear	Jira:RHEL-102563, Jira:RHEL-84286
resource-agents	Jira:RHEL-85014, Jira:RHEL-81237, Jira:RHEL-81236, Jira:RHEL-13089
rhel-bootc-container	Jira:RHEL-82380, Jira:RHEL-34859
rhel-drivers	Jira:RHEL-113198
rhel-system-roles	Jira:RHEL-102635, Jira:RHEL-101724, Jira:RHEL-101671, Jira:RHEL-99087, Jira:RHEL-95846, Jira:RHEL-88312, Jira:RHEL-84953, Jira:RHEL-84932, Jira:RHEL-78262, Jira:RHEL-85689, Jira:RHEL-46225, Jira:RHEL-46224, Jira:RHEL-105093, Jira:RHEL-103887, Jira:RHEL-103573, Jira:RHEL-101676, Jira:RHEL-100819, Jira:RHEL-95883, Jira:RHEL-95757, Jira:RHEL-95729, Jira:RHEL-93294, Jira:RHEL-88774, Jira:RHEL-88297, Jira:RHEL-88286, Jira:RHEL-88263, Jira:RHEL-88249, Jira:RHEL-85704, Jira:RHEL-84942, Jira:RHEL-84912, Jira:RHEL-81918, Jira:RHEL-107047, Jira:RHEL-107013, Jira:RHEL-106261, Jira:RHEL-84922, Jira:RHEL-88760, Jira:RHEL-90216, Jira:RHEL-73440
rng-tools	Jira:RHEL-91113
rpm	Jira:RHEL-84057, Jira:RHEL-84062, Jira:RHEL-56363

Component	Tickets
rsyslog	Jira:RHEL-96589, Jira:RHEL-92757
rteval	Jira:RHEL-97541
rust	Jira:RHEL-81600
rust-rpm-sequoia	Jira:RHEL-101952
rust-sequoia-sq	Jira:RHEL-85985
s390utils	Jira:RHEL-73341
samba	Jira:RHEL-89870
sblim-sfcb	Jira:RHEL-93092
scap-security-guide	Jira:RHEL-111008
selinux-policy	Jira:RHEL-54303, Jira:RHEL-89587, Jira:RHEL-82672, Jira:RHEL-87742, Jira:RHEL-69450, Jira:RHEL-77808
setroubleshoot	Jira:RHEL-90842
shim	Jira:RHEL-81188
sos	Jira:RHEL-71825, Jira:RHEL-62972, Jira:RHEL-67097, Jira:RHEL-73028, Jira:RHEL-81187, Jira:RHEL-81634, Jira:RHEL-84078
sssd	Jira:RHEL-4976, Jira:RHEL-13086
stalld	Jira:RHEL-109112, Jira:RHEL-73883
subscription-manager	Jira:RHEL-13374
systemd	Jira:RHEL-92781
systemtap	Jira:RHEL-86999
tbb	Jira:RHEL-33633
tftp	Jira:RHEL-79983
tog-pegasus	Jira:RHEL-93093

Component	Tickets
toolbox-container	Jira:RHEL-85074
trustee-guest-components	Jira:RHEL-73770
tuned	Jira:RHEL-79913
tzdata	Jira:RHEL-105042
valgrind	Jira:RHEL-86988, Jira:RHEL-75470
varnish	Jira:RHEL-45756
virt-v2v	Jira:RHEL-13340
virtio-win	Jira:RHEL-1609
virtio-win / user-mode	Jira:RHEL-91041
virtio-win / virtio-win-prewhql	Jira:RHEL-40693, Jira:RHEL-1084, Jira:RHEL-53962, Jira:RHEL-12118, Jira:RHEL-935
virtiofsd	Jira:RHEL-87161, Jira:RHEL-99895
wpa_supplicant	Jira:RHEL-86951

Component	Tickets
other	Jira:RHELDOCS-21104, Jira:RHELDOCS-20640, Jira:RHELDOCS-20749, Jira:RHELDOCS-20453, Jira:RHELDOCS-20303, Jira:RHELDOCS-20257, Jira:RHELDOCS-21103, Jira:RHELDOCS-20546, Jira:RHELDOCS-20421, Jira:RHELDOCS-21016, Jira:RHELDOCS-21025, Jira:RHELDOCS-20633, Jira:RHELDOCS-21026, Jira:RHELDOCS-18640, Jira:RHELDOCS-21241, Jira:RHELDOCS-21230, Jira:RHELDOCS-21218, Jira:RHELDOCS-21313, Jira:RHELDOCS-20674, Jira:RHELDOCS-19500, Jira:RHELDOCS-21013, Jira:RHELDOCS-21314, Jira:RHELDOCS-21315, Jira:RHELDOCS-21316, Jira:RHELDOCS-21317, Jira:RHELDOCS-21318, Jira:RHELDOCS-21319, Jira:RHELDOCS-21320, Jira:RHELDOCS-21322, Jira:RHELDOCS-16800, Jira:RHELDOCS-17465, Jira:RHELDOCS-19891, Jira:RHELDOCS-19968, Jira:RHELDOCS-20041, Jira:RHELDOCS-20042, Jira:RHELDOCS-20043, Jira:RHELDOCS-20080, Jira:RHELDOCS-19635, Jira:RHELDOCS-20258, Jira:RHELDOCS-20354, Jira:RHELDOCS-19757, Jira:RHELDOCS-20167, Jira:RHELDOCS-21349, Jira:RHEL-97489, Jira:RHELDOCS-20138, Jira:RHELDOCS-18700, Jira:RHELDOCS-18903, Jira:RHELDOCS-18904, Jira:RHELDOCS-18491, Jira:RHELDOCS-18672, Jira:RHELDOCS-18450, Jira:RHELDOCS-20147, Jira:RHELDOCS-20283, Jira:RHELDOCS-20610, Jira:RHELDOCS-16612, Jira:RHELDOCS-21325, Jira:RHELDOCS-19015, Jira:RHELDOCS-19172, Jira:RHELDOCS-19603, Jira:RHELDOCS-18471, Jira:RHELDOCS-19770, Jira:RHELDOCS-19539, Jira:RHELDOCS-19734, Jira:RHELDOCS-19948, Jira:RHELDOCS-19496, Jira:RHELDOCS-19945

APPENDIX B. REVISION HISTORY

0.0-5

Thu 22 Jan 2026, Valentina Ashirova (vaashiro@redhat.com)

- Added a Known Issue RHELDOCS-21726 (RHEL Lightspeed)

0.0-4

Tue 20 Jan 2026, Valentina Ashirova (vaashiro@redhat.com)

- Updated the In-place upgrade section
- Updated the Architectures section

0.0-3

Wed 14 Jan 2026, Valentina Ashirova (vaashiro@redhat.com)

- Added a Technology Preview RHELDOCS-21587 (File systems and storage)

0.0-2

Fri 09 Jan 2026, Valentina Ashirova (vaashiro@redhat.com)

- Added a Developer Preview RHELDOCS-21153 (RHEL Lightspeed)

0.0-1

Wed 17 Dec 2025, Valentina Ashirova (vaashiro@redhat.com)

- Added Enhancements RHELDOCS-21410 and RHELDOCS-21467 (Virtualization)

0.0-0

Wed 12 Nov 2025, Valentina Ashirova (vaashiro@redhat.com)

- Release of the Red Hat Enterprise Linux 10.1 Release Notes.