



Red Hat Enterprise Linux 8

8.1 リリースノート

Red Hat Enterprise Linux 8.1 リリースノート

Red Hat Enterprise Linux 8 8.1 リリースノート

Red Hat Enterprise Linux 8.1 リリースノート

法律上の通知

Copyright © 2024 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

本リリースノートでは、Red Hat Enterprise Linux 8.1 での改良点および実装された追加機能の概要、本リリースにおける既知の問題などを説明します。また、重要なバグ修正、テクニカルレビュー、非推奨の機能などの詳細も説明します。

目次

RED HAT ドキュメントへのフィードバック (英語のみ)	3
第1章 概要	4
インストーラーおよびイメージの作成	4
Red Hat Enterprise Linux システムロール	4
インフラストラクチャーサービス	4
セキュリティー	4
カーネル	4
ファイルシステムおよびストレージ	4
動的プログラミング言語、Web サーバー、およびデータベースサーバー	5
コンパイラツールセット	5
Identity Management	5
デスクトップ	5
RHEL 7 から RHEL 8 へのインプレースアップグレード	5
関連情報	6
第2章 アーキテクチャー	8
第3章 外部のカーネルパラメーターに対する重要な変更	9
新しいカーネルパラメーター	9
更新されたカーネルパラメーター	11
新しい /proc/sys/kernel パラメーター	12
新しい /proc/sys/net パラメーター	12
更新された /proc/sys/fs パラメーター	12
更新された /proc/sys/kernel パラメーター	13
第4章 RHEL 8 のコンテンツの配布	14
4.1. インストール	14
4.2. リポジトリ	14
4.3. アプリケーションストリーム	15
4.4. YUM/DNF を使用したパッケージ管理	15
第5章 RHEL 8.1.1 リリース	16
5.1. 新機能	16
第6章 RHEL 8.1.0 リリース	19
6.1. 新機能	19
6.2. 新しいドライバー	46
6.3. 更新されたドライバー	48
6.4. バグ修正	49
6.5. テクノロジープレビュー	62
6.6. 非推奨になった機能	73
6.7. 既知の問題	79
第7章 コンテナーに関する主な変更点	101
第8章 国際化	102
8.1. RED HAT ENTERPRISE LINUX 8 の多言語	102
8.2. RHEL 8 における国際化の主な変更点	102
付録A コンポーネント別のチケットリスト	104
付録B 更新履歴	110

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに対するご意見をお聞かせください。ドキュメントの改善点があればお知らせください。これを行うには、以下を行います。

Jira からのフィードバック送信 (アカウントが必要)

1. [Jira](#) の Web サイトにログインします。
2. 上部のナビゲーションバーで **Create** をクリックします。
3. **Summary** フィールドにわかりやすいタイトルを入力します。
4. **Description** フィールドに、ドキュメントの改善に関するご意見を記入してください。ドキュメントの該当部分へのリンクも追加してください。
5. ダイアログの下部にある **Create** をクリックします。

第1章 概要

インストーラーおよびイメージの作成

ユーザーは、キックスタートインストール時にモジュールを無効にすることができます。

詳細は「[インストーラーおよびイメージの作成](#)」を参照してください。

Red Hat Enterprise Linux システムロール

新しいストレージロールが RHEL システムロールに追加されました。

詳細は、「[Red Hat Enterprise Linux システムロール](#)」を参照してください。

インフラストラクチャーサービス

RHEL 8.1には、新しいルーティングプロトコルスタックの **FRR** を導入しています。これは、以前のバージョンの RHEL で使用していた **Quagga** に取って代わります。**FRR** は、複数の IPv4 および IPv6 ルーティングプロトコルをサポートする TCP/IP ベースのルーティングサービスを提供します。

Tuned システムチューニングツールはバージョン 2.12 にリベースされており、CPU リストのネゴシエーションのサポートが追加されました。

chrony スイートがバージョン 3.5 にリベースされており、RHEL 8.1 カーネルのハードウェアタイムスタンプとシステムクロックの同期がより正確になります。

詳細は、「[インフラストラクチャーサービス](#)」を参照してください。

セキュリティ

RHEL 8.1では、コンテナ用の SELinux ポリシーを生成するための新しいツール **udica** が追加されました。**udica** を使用すると、最適なセキュリティポリシーを作成して、ストレージ、デバイス、ネットワークなどのホストシステムリソースにコンテナにアクセスする方法を制御することができます。これにより、セキュリティ違反に対してコンテナのデプロイメントを強化でき、規制コンプライアンスの実現や維持も簡単になります。

fapolicyd ソフトウェアフレームワークは、ユーザー定義ポリシーに基づいたアプリケーションのホワイトリスト化およびブラックリスト化の形式を導入しました。RHEL 8.1 アプリケーションのホワイトリスト機能では、システム上で信頼されていないアプリケーションや悪意のあるアプリケーションを実行しないようにするための最も効率的な方法を利用できます。

セキュリティコンプライアンススイートの **OpenSCAP** が SCAP 1.3 データストリームに対応し、改善されたレポートを提供するようになりました。

詳細は、「[セキュリティ](#)」を参照してください。

カーネル

カーネル用のライブパッチである **kpatch** が利用できるようになり、システムを再起動しなくても、重大かつ重要な CVE の修正を利用できるようになります。

Extended Berkeley Packet Filter (eBPF) は、カーネル領域でのコード実行を可能にするカーネル内の仮想マシンです。**eBPF** は、RHEL の多くのコンポーネントで利用されています。RHEL 8.1では、**BPF Compiler Collection (BCC)** ツールパッケージが、AMD および Intel 64 ビットアーキテクチャーで完全にサポートされ、その他のアーキテクチャーではテクノロジープレビューとして利用できます。さらに、**bpfftrace** トレース言語および **eXpress Data Path (XDP)** 機能がテクノロジープレビューとして利用できます。

詳しい情報は、「[カーネル](#)」および「[カーネル](#)」を参照してください。

ファイルシステムおよびストレージ

LUKS バージョン 2 (LUKS2) 形式は、デバイスの使用中におけるブロックデバイスの再暗号化に対応するようになりました。

詳細は、「[ファイルシステムおよびストレージ](#)」を参照してください。

動的プログラミング言語、Web サーバー、およびデータベースサーバー

以下のコンポーネントの後続のバージョンが、新しいモジュールストリームとして利用できるようになりました。

- **PHP 7.3**
- **Ruby 2.6**
- **Node.js 12**
- **nginx 1.16**

詳細は、「[動的プログラミング言語、Web サーバー、およびデータベースサーバー](#)」を参照してください。

コンパイラーツールセット

RHEL 8.1では、ソフトウェアコレクションとしてパッケージ化された Application Stream の新しいコンパイラーツールセット **GCC Toolset 9**が導入されました。これには、新しいバージョンの開発ツールが含まれます。

また、以下のコンパイラーツールセットもアップグレードされました。

- **LLVM 8.0.1**
- **Rust Toolset 1.37**
- **Go Toolset 1.12.8**

詳細は、「[コンパイラーおよび開発ツール](#)」を参照してください。

Identity Management

Identity Management では、新しいコマンドラインツール **Healthcheck** が導入されました。**Healthcheck** は、IdM 環境の適合性に影響を与える可能性がある問題の特定に役立ちます。

詳細は、「[ID 管理](#)」を参照してください。

Identity Management が、インストールおよび管理用の Ansible ロールおよびモジュールに対応しました。今回の更新で、IdM ベースのソリューションのインストールおよび設定が容易になります。

詳細は、「[ID 管理](#)」を参照してください。

デスクトップ

GNOME クラシック環境のワークスペーススイッチが変更されました。このスイッチは、一番下のバーの右部分に移動され、横線のサムネイルとして設計されています。必要なサムネイルをクリックすることで、ワークスペース間の切り替えが可能です。詳細は、「[デスクトップ](#)」を参照してください。

Direct Rendering Manager (DRM) カーネルグラフィックサブシステムが、アップストリームの Linux カーネルバージョン 5.1 にリベースされました。このバージョンでは、新しい GPU および APU のサポートや、さまざまなドライバー更新など、以前のバージョンから多くの機能強化が行われています。詳細は「[デスクトップ](#)」を参照してください。

RHEL 7 から RHEL 8 へのインプレースアップグレード

以下の主な機能拡張が追加されました。

- 以下のアーキテクチャーにおけるインプレースアップグレードのサポートが追加されました。64 ビット ARM、IBM POWER (little endian)、IBM Z。
- Web コンソールでアップグレード前のシステム評価を実行し、新しい **cockpit-leapp** プラグインを使用して自動修正を適用できるようになりました。
- **/var** ディレクトリーまたは **/usr** ディレクトリーを別のパーティションにマウントできるようになりました。
- UEFI に対応するようになりました。
- **Leapp** は、Supplementary リポジトリからパッケージをアップグレードするようになりました。

サポート対象のアップグレードパスの詳細は、[Supported in-place upgrade paths for Red Hat Enterprise Linux](#) を参照してください。インプレースアップグレードの実行方法は、[Upgrading from RHEL 7 to RHEL 8](#) を参照してください。

CentOS Linux 7 または Oracle Linux 7 を使用している場合は、RHEL 8 にアップグレードする前に、**convert2rhel** ユーティリティを使用してオペレーティングシステムを RHEL 7 に変換できます。手順は、[RPM ベースの Linux ディストリビューションから RHEL への変換](#) を参照してください。

関連情報

- 他のバージョンと比較した Red Hat Enterprise Linux 8.0 の **機能および制限** は、Red Hat ナレッジベースの記事 [Red Hat Enterprise Linux technology capabilities and limits](#) を参照してください。
- Red Hat Enterprise Linux の **ライフサイクル** に関する情報は [Red Hat Enterprise Linux のライフサイクル](#) を参照してください。
- RHEL 8 の **パッケージリスト** は、[パッケージマニフェスト](#) を参照してください。
- RHEL 7 と RHEL 8 の **主な相違点** は、[RHEL 8 の導入における検討事項](#) を参照してください。
- RHEL 7 から RHEL 8 への **インプレースアップグレード** を実行する方法は、[RHEL 7 から RHEL 8 へのアップグレード](#) を参照してください。
- すべての RHEL サブスクリプションで、既知の技術問題の特定、検証、および解決をプロアクティブに行う **Red Hat Insights** サービスが利用できるようになりました。Red Hat Insights クライアントをインストールし、システムをサービスに登録する方法は、[Red Hat Insights を使い始める](#) ページを参照してください。

Red Hat Customer Portal Labs

Red Hat Customer Portal Labs は、カスタマーポータル内のセクションにあるツールセットで、<https://access.redhat.com/labs/> から入手できます。Red Hat Customer Portal Labs のアプリケーションは、パフォーマンスの向上、問題の迅速なトラブルシューティング、セキュリティ問題の特定、複雑なアプリケーションの迅速なデプロイメントおよび設定に役立ちます。最も一般的なアプリケーションには、以下のものがあります。

- [Registration Assistant](#)
- [Product Life Cycle Checker](#)
- [Kickstart Generator](#)

- [Red Hat Satellite Upgrade Helper](#)
- [Red Hat Code Browser](#)
- [JVM Options Configuration Tool](#)
- [Red Hat CVE Checker](#)
- [Red Hat Product Certificates](#)
- [Load Balancer Configuration Tool](#)
- [Yum Repository Configuration Helper](#)

第2章 アーキテクチャー

Red Hat Enterprise Linux 8.1 には、カーネルバージョン 4.18.0-147 が同梱されており、以下のアーキテクチャーに対応します。

- AMD アーキテクチャーおよび Intel 64 ビットアーキテクチャー
- 64 ビット ARM アーキテクチャー
- IBM Power Systems (リトルエンディアン)
- 64 ビット IBM Z

各アーキテクチャーに適切なサブスクリプションを購入してください。詳細は [Get Started with Red Hat Enterprise Linux - additional architectures](#) を参照してください。利用可能なサブスクリプションのリストは、カスタマーポータル [サブスクリプションの使用状況](#) を参照してください。

第3章 外部のカーネルパラメーターに対する重要な変更

本章では、システム管理者向けに、Red Hat Enterprise Linux 8.1 に同梱されるカーネルにおける重要な変更の概要について説明します。変更には、**proc** エントリー、**sysctl** および **sysfs** のデフォルト値、boot パラメーター、カーネル設定オプションの追加や更新、注目すべき動作の変更などが含まれます。

新しいカーネルパラメーター

perf_v4_pmi = [X86,INTEL]

このパラメーターは、Intel PMU カウンターのフリーズ機能を無効にします。
この機能は、Arch Perfmon v4 (Skylake 以降) 以降でのみ存在します。

形式: <bool>

hv_nopvspin [X86,HYPER_V]

このパラメーターは、準仮想スピンロックの最適化を無効にし、ハイパーバイザーがロックの競合時にゲストをアイドル状態にすることができます。

ipcmni_extend [KNL]

このパラメーターは、一意の System V IPC 識別子の最大数を 32,768 から 16,777,216 に拡張します。

kpti = [ARM64]

このパラメーターは、ユーザーおよびカーネルアドレス領域のページテーブルの分離を制御します。
オプションは次のとおりです。

- **デフォルト**: 軽減策が必要なコアで有効になっています。
- **0**: 強制的に無効にします。
- **1**: 強制的に有効にします。

mds = [X86,INTEL]

このパラメーターは、マイクロアーキテクチャーデータサンプリング (MDS) の脆弱性に対する軽減策を制御します。

特定の CPU は、CPU の内部のバッファーに対する悪用から影響を受けやすく、特定の条件で開示ガジェットに情報が転送される可能性があります。脆弱なプロセッサでは、攻撃者が直接アクセス権を持たないデータにアクセスするために、キャッシュ側のチャンネル攻撃で、予測的に転送されるデータを利用できます。

オプションは次のとおりです。

- **full** - 脆弱な CPU で MDS 軽減策を有効にします。
- **full,nosmt** - 脆弱な CPU で MDS 軽減策を有効にし、同時マルチスレッド (SMT) を無効にします。
- **off** - MDS 軽減策を無条件に無効にします。
このパラメーターを指定しない場合は、**mds=full** と同じです。

詳細は、アップストリームの [カーネルドキュメント](#) を参照してください。

mitigations = [X86,PPC,S390,ARM64]

このパラメーターは、CPU の脆弱性に対するオプションの軽減策を制御します。これは、既存のアーキテクチャー固有のオプションの集約となる、一連の、アーキテクチャーに依存しないオプションです。

オプションは次のとおりです。

- **off** - オプションの CPU 軽減策をすべて無効にします。これによりシステムパフォーマンスが向上しますが、ユーザーを複数の CPU の脆弱性にさらす可能性もあります。以下に相当します。
 - **nopti [X86,PPC]**
 - **kpti=0 [ARM64]**
 - **nospectre_v1 [X86,PPC]**
 - **nobp=0 [S390]**
 - **nospectre_v2 [X86,PPC,S390,ARM64]**
 - **spectre_v2_user=off [X86]**
 - **spec_store_bypass_disable=off [X86,PPC]**
 - **ssbd=force-off [ARM64]**
 - **l1tf=off [X86]**
 - **mds=off [X86]**
- **auto** (デフォルト) - すべての CPU 脆弱性を軽減しますが、脆弱な場合でも SMT (Simultaneous multithreading) を有効にしたままにしておきます。このオプションは、カーネルのアップグレードで SMT が自動的に無効化されないようにしたいユーザーや、SMT ベースの攻撃を他の方法で回避できるユーザーを対象にしています。以下に相当します。
 - (デフォルトの動作)
- **auto,nosmt** - 必要に応じて同時マルチスレッディング (SMT) を無効にし、すべての CPU 脆弱性を軽減します。このオプションは、SMT が失われるとしても常に完全な軽減を希望するユーザー向けです。以下に相当します。
 - **l1tf=flush,nosmt [X86]**
 - **mds=full,nosmt [X86]**

novmcoredd [KNL,KDUMP]

このパラメーターは、デバイスダンプを無効にします。

デバイスダンプを使用すると、ドライバーがダンプデータを vmcore に追加できるため、ドライバーが指定したデバッグ情報を収集できます。ドライバーは無制限にデータを追加でき、このデータはメモリーに保存されるため、メモリーに大きな負荷がかかる可能性があります。

デバイスダンプを無効にするとメモリーを節約できますが、ドライバーのデバッグデータは利用できなくなります。

このパラメーターは、**CONFIG_PROC_VMCORE_DEVICE_DUMP** カーネル設定が指定されている場合にのみ使用できます。

nospectre_v1 [X86]

このパラメーターは、Spectre Variant 1 (バインドチェックの回避) の軽減策を無効にします。このオプションを使用すると、システムでデータ漏洩が発生する可能性があります。

psi = [KNL]

このパラメーターは、Pressure Stall Information の追跡を有効または無効にします。
形式: <bool>

random.trust_cpu={on,off} [KNL]

このパラメーターは、CPU の乱数ジェネレーター (ある場合) の使用を信頼する機能を有効化または無効化し、カーネルの暗号論的擬似乱数生成器 (CRNG) を完全にシードします。デフォルトは、**CONFIG_RANDOM_TRUST_CPU** カーネル設定によって制御されます。

vm_debug[=options] [KNL]

CONFIG_DEBUG_VM=y で利用できます。

このパラメーターを有効にすると、システムの起動速度が遅くなる場合があります。特に、大量のメモリーを搭載したシステムでは顕著です。

すべてのオプションはデフォルトで有効になっています。このインターフェイスは、特定の仮想メモリーのデバッグ機能を選択的に有効または無効にできるようにするためのものです。

オプションは次のとおりです。

- **P** - ページ構造の init 時間の Poisoning を有効にします。
- **-** (ダッシュ) - 上記のオプションをすべて無効にします。

更新されたカーネルパラメーター

cgroup_no_v1 = [KNL]

このパラメーターは、バージョン 1 (v1) の cgroup コントローラーと名前付き階層を無効にします。このパラメーターは **cgroup_disable** カーネルパラメーターに似ていますが、cgroup v1 にのみ適用されます。ブラックリストに登録されたコントローラーは、cgroup2 で引き続き使用できます。all オプションはすべてのコントローラーをブラックリストに登録し、named オプションは名前付きマウントを無効にします。all と named の両方を指定すると、すべての v1 階層が無効になります。

形式: { { controller | "all" | "named" } [, { controller | "all" | "named" } ...] }

crashkernel = size[KMG][@offset[KMG]][KNL]

kexec システムコールにより、Linux はパニック時にクラッシュカーネルに切り替えることができます。このパラメーターは、そのカーネルイメージの物理メモリー領域 [offset, offset + size] を予約します。**@offset** を省略すると、適切なオフセットが自動的に選択されます。

[KNL, x86_64] **@offset** が指定されていない場合、最初に 4G 未満のリージョンを選択し、4G 以上のリージョンを予約するようにフォールバックします。

詳細については、アップストリームの [kdump ドキュメント](#) を参照してください。

l1tf = [X86]

このパラメーターは、影響を受ける CPU の L1 Terminal Fault (L1TF) 脆弱性の軽減策を制御します。

オプションは次のとおりです。

- **off** - ハイパーバイザーの軽減策を無効にし、警告を発行しません。また、ハイパーバイザーとベアメタルの両方で、スワップサイズと使用可能な RAM の制限が解除されます。
- **flush** - デフォルトです。
詳細については、アップストリームの [カーネルのドキュメント](#) を参照してください。

nospectre_v2 [X86,PPC_FSL_BOOK3E,ARM64]

このパラメーターは、Spectre Variant 2 (間接的な分岐予測) の脆弱性に対するすべての軽減策を無効にします。

システムは、このパラメーターを使用してデータ漏洩を許可する場合があります。

pci=option[,option...][PCI]

さまざまな PCI サブシステムオプション。

オプションは次のとおりです。

- **force_floating [S390]** - 浮動割り込みを強制的に使用します。
- **nomio [S390]** - メモリー input/output (MIO) 命令を使用しません。

新しい `/proc/sys/kernel` パラメーター

hyperv_record_panic_msg

このパラメーターは、パニックカーネルメッセージ (kmsg) データを Hyper-V に報告するかどうかを制御します。

値は以下のようになります。

- **0**: パニック kmsg データを報告しません。
- **1**: パニック kmsg データを報告します。これがデフォルトの動作です。

新しい `/proc/sys/net` パラメーター

bpf_jit_limit

このパラメーターは、Berkeley Packet Filter Just-in-Time (BPF JIT) コンパイラーへのメモリー割り当てにグローバルな制限を適用し、制限を超えると特権のない JIT 要求を拒否します。

bpf_jit_limit パラメーターには、グローバル制限の値がバイト単位で含まれています。

更新された `/proc/sys/fs` パラメーター

dentry-state

Dentries の割り当ておよび割当解除は動的に行われます。

ユーザーは、`/proc/sys/fs/dentry-state` ファイルを読み取って、次の値を取得できます。

- **nr_dentry** - 割り当てられた dentry の総数 (アクティブ + 未使用) を示します。
- **nr_unused** - あまり使用されていない Dentry の数を表示しますが、今後再利用できるように最近一番使用されていないもの (LRU) に保存されます。

- **age_limit** - メモリーが不足している場合に **dcache** エントリーを再利用できるようになるまでの経過時間を秒単位で示します。
- **want_pages** - **Shrink_dcache_pages()** 関数が呼び出され、**dcache** がまだプルーニングされていない場合にはゼロ以外になります。
- **nr_negative** - 未使用の **dentry** の数を表示します。この数は、どのファイルにもマッピングされていない負の場合もあります。代わりに、ユーザーから提供された存在しないファイルの拒否にかかる時間を短縮できます。

更新された `/proc/sys/kernel` パラメーター

`msg_next_id`、`sem_next_id`、および `shm_next_id`

注記:

1. このカーネルでは、新しいオブジェクトに目的の ID が指定されるとは限りません。間違っただ ID が割り当てられたオブジェクトを処理する方法は、ユーザー空間によって異なります。
2. デフォルト以外の値が指定されたトグルは、プロセス間通信 (IPC) オブジェクトの割り当てが成功した後、カーネルによって `-1` に戻されます。IPC オブジェクト割り当て `syscall` が失敗した場合に、値が変更されないままであるか、`-1` にリセットされるかは未定義です。

第4章 RHEL 8 のコンテンツの配布

4.1. インストール

Red Hat Enterprise Linux 8 は、ISO イメージを使用してインストールします。AMD64、Intel 64 ビット、64 ビット ARM、IBM Power Systems、IBM Z アーキテクチャーで、以下の 2 種類のインストールメディアが利用できます。

- Binary DVD ISO - BaseOS リポジトリおよび AppStream リポジトリが含まれ、リポジトリを追加しなくてもインストールを完了できる完全インストールイメージです。



注記

Binary DVD ISO イメージが 4.7 GB を超え、1 層 DVD に収まらない場合があります。Binary DVD ISO イメージを使用して起動可能なインストールメディアを作成する場合は、2 層 DVD または USB キーが推奨されます。Image Builder ツールを使用すれば、RHEL イメージをカスタマイズできます。Image Builder の詳細は [RHEL システムイメージのカスタマイズの作成](#) を参照してください。

- Boot ISO - インストールプログラムを起動するのに使用する最小限の ISO ブートイメージです。このオプションでは、ソフトウェアパッケージをインストールするのに、BaseOS リポジトリおよび AppStream リポジトリにアクセスする必要があります。リポジトリは、Binary DVD ISO イメージに含まれます。

ISO イメージのダウンロード、インストールメディアの作成、および RHEL インストールの完了の手順については [標準的な RHEL 8 インストールの実行](#) ドキュメントを参照してください。自動化したキックスタートインストールなどの高度なトピックは [高度な RHEL 8 インストールの実行](#) を参照してください。

4.2. リポジトリ

Red Hat Enterprise Linux 8 は、2 つのメインリポジトリで配布されています。

- BaseOS
- AppStream

基本的な RHEL インストールにはどちらのリポジトリも必要で、すべての RHEL サブスクリプションで利用できます。

BaseOS リポジトリのコンテンツは、すべてのインストールのベースとなる、基本的な OS 機能のコアセットを提供します。このコンテンツは RPM 形式で提供されており、RHEL の以前のリリースと同様のサポート条件が適用されます。BaseOS から配布されるパッケージのリストは [パッケージマニフェスト](#) を参照してください。

アプリケーションストリーム (AppStream) リポジトリのコンテンツには、さまざまなワークロードとユースケースに対応するために、ユーザー空間アプリケーション、ランタイム言語、およびデータベースが含まれています。Application Streams は、[モジュール](#) と呼ばれる RPM 形式への拡張、または Software Collections として通常の RPM 形式で利用できます。AppStream で利用可能なパッケージのリストは、[パッケージマニフェスト](#) を参照してください。

また、CodeReady Linux Builder リポジトリは、すべての RHEL サブスクリプションで利用できます。このリポジトリは、開発者向けの追加パッケージを提供します。CodeReady Linux Builder リポジトリに含まれるパッケージには対応しません。

RHEL 8 リポジトリの詳細は [パッケージマニフェスト](#) を参照してください。

4.3. アプリケーションストリーム

Red Hat Enterprise Linux 8 では、アプリケーションストリームの概念が導入されました。ユーザー空間コンポーネントのバージョンが複数配信され、オペレーティングシステムのコアパッケージよりも頻繁に更新されるようになりました。これによりプラットフォームや特定のデプロイメントの基本的な安定性に影響を及ぼすことなく、Red Hat Enterprise Linux をカスタマイズする柔軟性が向上します。

アプリケーションストリームとして使用できるコンポーネントは、モジュールまたは RPM パッケージとしてパッケージ化され、RHEL 8 の AppStream リポジトリを介して配信されます。各 Application Stream コンポーネントには、RHEL 8 と同じか、より短いライフサイクルが指定されています。詳細は [Red Hat Enterprise Linux のライフサイクル](#) を参照してください。

モジュールは、論理ユニット (アプリケーション、言語スタック、データベース、またはツールセット) を表すパッケージの集まりです。これらのパッケージはまとめてビルドされ、テストされ、そしてリリースされます。

モジュールストリームは、アプリケーションストリームコンポーネントのバージョンを表します。たとえば、**postgresql:10** のデフォルトのストリーム以外に、**postgresql** モジュールでは、PostgreSQL データベースサーバーの複数のストリーム (バージョン) を利用できます。システムにインストールできるモジュールストリームは1つだけです。複数のコンテナで異なるバージョンを使用できます。

詳細なモジュールコマンドは [ユーザー空間コンポーネントのインストール、管理、および削除](#) を参照してください。AppStream で利用可能なモジュールのリストは、[Package manifest](#) を参照してください。

4.4. YUM/DNF を使用したパッケージ管理

Red Hat Enterprise Linux 8 へのソフトウェアのインストールは、DNF テクノロジーをベースとした YUM ツールにより行われます。以前のメジャーバージョンの RHEL との一貫性を保つために、**yum** の用語の使用が意図的に準拠しています。ただし、**yum** の代わりに **dnf** を呼び出すと、**yum** は互換性のために **dnf** のエイリアスであるため、コマンドが期待どおりに動作します。

詳細は、以下のドキュメントを参照してください。

- [ユーザー空間コンポーネントのインストール、管理、および削除](#)
- [RHEL 8 の導入における検討事項](#)

第5章 RHEL 8.1.1 リリース

Red Hat は、マイナーリリース (8.Y) において Red Hat Enterprise Linux 8 のコンテンツを四半期ごとに利用できるようにしています。この四半期リリースは、3桁の数字 (8.Y.1) を使用して番号が付けられます。RHEL 8.1.1 リリースの新機能は以下のとおりです。

5.1. 新機能

新しいモジュールストリーム: `postgresql:12`

RHEL 8.1.1 リリースでは、**PostgreSQL 12** が導入されました。これは、バージョン 10 から多くの新機能および機能強化が追加されています。以下は、主な変更点です。

- 標準の PostgreSQL ロギング機能から詳細なセッションおよびオブジェクトロギングを提供する **pgaudit** (PostgreSQL Audit Extension)。
- たとえば、ハッシュパーティション作成のサポートなど、パーティション設定機能の改善
- クエリー並列処理の機能強化
- トランザクション管理を有効にするストアド SQL プロシージャ
- さまざまなパフォーマンス向上:
- 管理機能の強化
- SQL/JSON パス言語サポート
- ストアド生成絡む (stored generated columns)
- 非決定的結合
- GSSAPI 認証またはマルチファクター認証を使用する場合の TCP/IP 接続の暗号化など、新しい認証機能。

PostgreSQL 11 以降のアップストリームで利用できる、Just-In-Time (JIT) のコンパイルサポートは **postgresql:12** モジュールストリームでは提供されていないことに注意してください。

postgresql:12 ストリームをインストールするには、以下を実行します。

```
# yum module install postgresql:12
```

RHEL 8 内で以前の **postgresql** ストリームからアップグレードする場合は、[新しいストリームへの切り替え](#) で説明されている手順に従い、[RHEL 8 バージョンの PostgreSQL への移行](#) で説明されているように **PostgreSQL** データを移行します。

(JIRA:RHELPLAN-26926)

Rust Toolset がバージョン 1.39 にリベース

Rust Toolset は、バージョン 1.39 に更新されました。以下は、主な変更点です。

- **async** - **.await** 構文が、安定した Rust に追加されました。現在、**async** 関数とブロックを定義して **.await** できるようになりました。

- パイプラインのコンパイルの強化により、一部のクレーとグラフの最適なクリーンビルドのビルド時間が 10 ~ 20% 向上します。
- **match** 式の主なパターンに、**by-move** バインディングがある場合、**if** がこれらのバインディングを参照できるようになりました。
- rust はコンパイル時にメモリーの安全性のバグを検出しますが、以前のポロージャーには制限がありました。そのため、未定義の動作が許可され、メモリーの危険性が発生していました。新しい NLL ポロージャーはこれらの問題を見つけることができます。また、移行ステップとして警告していました。これらの警告はハードエラーになりました。
- **rustc** コンパイラーでは、**&T** や **Box<T>** などの一部のタイプの初期化に関数 **mem:: {uninitialized, zeroed}** が使用されるとき、**lint** を使用できるようになりました。
- 以下の関数は、以下の標準ライブラリーの **const fn** になりました。**Vec::new**、**String::new**、**LinkedList::new**、**str::len**、**[T>::len**、**str::as_bytes**、**abswrap**、**ping_abs**、**overflowing_abs**。

Rust Toolset モジュールストリームをインストールするには、**root** で以下のコマンドを実行します。

```
# yum module install rust-toolset
```

使用方法の詳細は [Rust Toolset の使用](#) を参照してください。

(BZ#1680096)

新しいモジュール: **jmc:rhel8**

RHEL 8.1.1 では、HotSpot JVM 用の強力なプロファイラーである JDK Mission Control (JMC) が新しい **jmc** モジュールとして導入されています。JMC では、JDK Flight Recorder が収集した豊富なデータを効率的かつ詳細に分析するための高度なツールセットを利用できます。ツールチェーンを使用すると、開発者および管理者は、実稼働環境でローカルに実行しているまたはデプロイした Java アプリケーションのデータを収集および分析できます。JMC を実行するには、JDK バージョン 8 以降が必要であることに注意してください。ターゲット Java アプリケーションは、最低でも OpenJDK バージョン 11 で実行する必要があります。これにより、JMC が JDK Flight Recorder 機能にアクセスできるようになります。

jmc:rhel8 モジュールストリームには、以下の 2 つのプロファイルがあります。

- JMC アプリケーション全体をインストールする **common** プロファイル
- Java ライブラリーのみをインストールする **core** プロファイル (**jmc-core**)

jmc:rhel8 モジュールストリームの **common** プロファイルをインストールするには、以下を使用します。

```
# yum module install jmc:rhel8/common
```

プロファイル名を **core** に変更して、**jmc-core** パッケージのみをインストールします。

(BZ#1716452)

RHEL 8 で NET Core 3.1 が利用可能になりました

今回の更新で、.NET Core 3.1 Software Development Kit (SDK) および .NET Core 3.1 Runtime が RHEL 8 に追加されました。さらに、Web アプリケーションとサービスを構築するための ASP.NET Core 3.1 フレームワークが利用できるようになりました。

(BZ#1711405)

virtio-win ドライバー用の新しいインストーラー

virtio-win パッケージに対話式の Windows インストーラーが追加されました。これにより、Microsoft Windows をゲストオペレーティングシステムとして使用する仮想マシンで、準仮想化 KVM ドライバーを簡単かつ効率的にインストールできるようになります。

(BZ#1745298)

container-tools が更新されました。

podman、**buildah**、**skopeo**、および **runc** ツールを含む **container-tools** モジュールが更新されました。コンテナのツールは、FIPS モードが有効な状態で構築されるようになりました。また、この更新では、複数のバグとセキュリティー問題が修正されます。

(BZ#1783277)

common は別のパッケージになりました。

common Open Container Initiative (OCI) コンテナランタイムモニターユーティリティーは、別の **common** パッケージに移動されました。**podman** パッケージでは使用できなくなりました。

(BZ#1753209)

第6章 RHEL 8.1.0 リリース

6.1. 新機能

ここでは、Red Hat Enterprise Linux 8.1 に追加された新機能および主要な機能拡張を説明します。

6.1.1. インストーラーおよびイメージの作成

キックスタートインストール時にモジュールを無効化できるようになりました。

この改善により、ユーザーはモジュールからパッケージをインストールしないようにモジュールを無効化できるようになりました。キックスタートインストール中にモジュールを無効にするには、以下のコマンドを使用します。

```
module --name=foo --stream=bar --disable
```

(BZ#1655523)

Blueprint に対する **repo.git** セクションのサポートが利用可能になりました。

新しい **repo.git** Blueprint セクションにより、ユーザーはイメージビルドにさらにファイルを含めることができます。このファイルは、**lorax-composer** ビルドサーバーからアクセス可能な git リポジトリでホストされる必要があります。

(BZ#1709594)

Image Builder が、より多くのクラウドプロバイダーのイメージ作成に対応します。

今回の更新で、Image Builder がイメージを作成できるクラウドプロバイダーの数が増えました。そのため、Google Cloud および Alibaba Cloud でもデプロイ可能な RHEL イメージを作成して、これらのプラットフォームでカスタムインスタンスを実行できるようになりました。

(BZ#1689140)

6.1.2. ソフトウェア管理

dnf-utils の名前が **yum-utils** に変更されました。

今回の更新で、YUM スタックの一部である **dnf-utils** パッケージのバージョンが、**yum-utils** に変更されました。互換性の理由から、このパッケージは依然として **dnf-utils** 名を使用してインストールできます。また、システムのアップグレード時に自動的に元のパッケージを置き換えます。

(BZ#1722093)

6.1.3. サブスクリプション管理

subscription-manager が、ロール、使用率、アドオンの値を報告するようになりました。

この更新により、**subscription-manager** は、カスタマーポータルまたはサテライトのいずれかに登録されている現在の組織で利用可能な各サブスクリプションの Role、Usage、Add-ons の値を表示できるようになりました。

- Role、Usage、Add-ons の値を追加した、利用可能なサブスクリプションを表示するには、以下のコマンドを実行します。

```
# subscription-manager list --available
```

- 追加の Role、Usage、Add-ons 値を含む、消費したサブスクリプションを表示するには、以下を使用します。

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

(BZ#1665167)

6.1.4. インフラストラクチャーサービス

tuned がバージョン 2.12 にリベースされました。

tuned パッケージがアップストリームバージョン 2.12 にアップグレードされ、以前のバージョンのバグ修正および機能拡張が数多く追加されました。主な改善点は以下の通りです。

- 削除、再接続されているデバイスの処理が修正されました。
- CPU リストのネゴシエーションサポートが追加されました。
- **sysctl** ツールから **Tuned** 固有の新しい実装に切り替えることで、ランタイムカーネルパラメーター設定のパフォーマンスが改善しました。

(BZ#1685585)

chrony がバージョン 3.5 にリベースされました。

tuned パッケージがアップストリームバージョン 3.5 にアップグレードされ、以前のバージョンのバグ修正および機能拡張が数多く追加されました。主な改善点は以下の通りです。

- RHEL 8.1 カーネルのハードウェアタイムスタンプとシステムクロックの同期がより正確になるサポートが追加されました。
- ハードウェアのタイムスタンプが大幅に改善されました。
- 利用可能なポーリング間隔の範囲が広がりました。
- フィルターオプションが NTP ソースに追加されました。

(BZ#1685469)

新しい **FRRouting** ルーティングプロトコルスタックが利用できるようになりました。

今回の更新で、**Quagga** が **Free Range Routing (FRRouting (FRR))** に置き換えられました。これは、新しいルーティングプロトコルスタックです。**FRR** は、AppStream リポジトリで利用可能な **fr** パッケージで提供されます。

FRR は、複数の IPv4 および IPv6 ルーティングプロトコル (**BGP**、**IS-IS**、**OSPF**、**PIM**、および **RIP** など) をサポートする TCP/IP ベースのルーティングサービスを提供します。

FRR がインストールされている場合、システムは専用ルーターとして動作します。これにより、内部ネットワークまたは外部ネットワークのいずれかの、その他のルーターとルーティング情報を交換します。

(BZ#1657029)

GNU が **ISO-8859-15** エンコードに対応しました。

今回の更新で、ISO-8859-15 エンコードへの対応が GNU `enscript` プログラムに追加されました。

(BZ#1664366)

`phc2sys` でのシステムクロックオフセットの測定精度の改善しました。

`linuxptp` パッケージの `phc2sys` プログラムが、システムクロックのオフセット測定をより正確に行える方法に対応しました。

(BZ#1677217)

`ptp4l` が `active-backup` モードでのチームインターフェイスに対応しました。

今回の更新で、`PTP Boundary/Ordinary Clock` (`ptp4l`) に、`active-backup` モードのチームインターフェイスサポートが追加されました。

(BZ#1685467)

`macvlan` インターフェイスでの `PTP` 時間同期に対応しました。

今回の更新で、`macvlan` インターフェイスのハードウェアタイムスタンプのサポートが Linux カーネルに追加されました。これにより、`macvlan` インターフェイスは、時刻の同期に `Precision Time Protocol` (PTP) を使用できるようになりました。

(BZ#1664359)

6.1.5. セキュリティー

新しいパッケージ: `fapolicyd`

`fapolicyd` ソフトウェアフレームワークは、ユーザー定義ポリシーに基づいたアプリケーションのホワイトリスト化およびブラックリスト化の形式を導入しました。アプリケーションのホワイトリスト機能では、システム上で信頼されていないアプリケーションや悪意のあるアプリケーションを実行しないようにするための最も効率的な方法を利用できます。

`fapolicyd` フレームワークは、以下のコンテンツを提供します。

- `fapolicyd` サービス
- `fapolicyd` コマンドラインユーティリティ
- `yum` プラグイン
- ルール言語

管理者は、すべてのアプリケーションのパス、ハッシュ、MIME タイプ、信頼に基づいて、`allow` および `deny` 実行ルールを両者とも監査できるようにして定義することができます。

すべての `fapolicyd` 設定は、全体的なシステムのパフォーマンスに影響を与えることに注意してください。パフォーマンスヒットは、ユースケースによって異なります。アプリケーションをホワイトリストにすると、`open()` および `exec()` システムコールが遅くなるため、主にこのようなシステムコールを頻繁に実行するアプリケーションに影響します。

詳細は、man ページの `fapolicyd(8)`、`fapolicyd.rules(5)`、および `fapolicyd.conf(5)` を参照してください。

(BZ#1673323)

新しいパッケージ: **udica**

新しい **udica** パッケージでは、コンテナ用の SELinux ポリシーのツールを利用できます。 **udica** を使用すると、最適なセキュリティポリシーを作成して、ストレージ、デバイス、ネットワークなどのホストシステムリソースにコンテナがアクセスする方法を制御することができます。これにより、セキュリティ違反に対してコンテナのデプロイメントを強化でき、規制コンプライアンスの実現や維持も簡単になります。

詳細は、SELinux タイトルを使用した RHEL 8 の [Creating SELinux policies for containers](#) セクションを参照してください。

(BZ#1673643)

SELinux ユーザースペースツールがバージョン 2.9 へ更新されました。

libsepol、**libselineux**、**libsemanage**、**policycoreutils**、**checkpolicy**、および **mcstrans** SELinux ユーザースペースツールが最新のアップストリームリリース 2.9 にアップグレードされ、以前のバージョンに対するバグ修正および機能拡張が数多く追加されました。

([BZ#1672638](#), [BZ#1672642](#), [BZ#1672637](#), [BZ#1672640](#), [BZ#1672635](#), [BZ#1672641](#))

SETools がバージョン 4.2.2 に更新されました。

SETools の一連のツールとライブラリーが最新のアップストリームリリース 4.2.2 にアップグレードされました。この更新では、以下の変更が行われました。

- ロードソースポリシーのサポートが停止したため、man ページのソースポリシーリファレンスを削除しました。
- エイリアスのロードでのパフォーマンスリグレッションを修正しました。

([BZ#1672631](#))

selinux-policy が 3.14.3 にリベースされました。

selinux-policy パッケージがアップストリームバージョン 3.14.3 にアップグレードされ、以前バージョンの許可ルールに対してバグ修正および機能拡張が数多く追加されました。

([BZ#1673107](#))

新しい SELinux のタイプ: **boltd_t**

新しい SELinux のタイプ **boltd_t** は、Thunderbolt 3 デバイスのマッピングにシステムデーモン **boltd** を制限します。その結果、**boltd** が SELinux 強制モードの制限付きサービスとして実行されるようになりました。

(BZ#1684103)

新しい SELinux ポリシークラス: **bpf**

新しい SELinux ポリシークラス **bpf** が導入されました。 **bpf** クラスを使用すると、ユーザーは SELinux を介して Berkeley Packet Filter (BPF) フローを制御できます。また、Extended Berkeley Packet Filter (eBPF) プログラムと、SELinux が制御するマップの検査と簡単な操作が可能になります。

(BZ#1673056)

OpenSCAP がバージョン 1.3.1 にリベースされました。

openscap パッケージがアップストリームバージョン 1.3.1 にアップグレードされ、以前のバージョンに対するバグ修正および機能拡張が数多く追加されました。主な変更点:

- SCAP 1.3 ソースデータストリームのサポート: 評価、XML スキーマ、およびバリデーション
- テーラリングファイルは ARF の結果ファイルに含まれます
- OVAL 詳細は常に HTML レポートで表示されます。ユーザーは **--oval-results** オプションを指定する必要はありません。
- HTML レポートは、OVAL **extend_definition** 要素を使用して、その他の OVAL 定義に含まれる OVAL テストの OVAL テスト詳細を表示します。
- OVAL テスト ID が HTML レポートに表示されます
- ルール ID が HTML ガイドに表示されます

([BZ#1718826](#))

OpenSCAP が SCAP 1.3 に対応しました。

OpenSCAP スイートは、最新バージョンの SCAP 標準 (SCAP 1.3) に準拠するデータストリームに対応するようになりました。**scap-security-guide** パッケージに含まれるものなど、SCAP 1.3 データストリームを、追加のユーザービリティ制限のない SCAP 1.2 データストリームと同じように使用できるようになりました。

([BZ#1709429](#))

scap-security-guide がバージョン 0.1.46 にリベースされました。

scap-security-guide パッケージがアップストリームバージョン 0.1.46 にアップグレードされ、以前のバージョンに対するバグ修正および機能拡張が数多く追加されました。*SCAP コンテンツが最新バージョンの SCAP 規格に準拠し、SCAP 1.3 * SCAP コンテンツが UBI イメージに対応

([BZ#1718839](#))

OpenSSH が 8.0p1 にリベースされました。

openssh パッケージがアップストリームバージョン 8.0p1 にアップグレードされ、以前のバージョンに対するバグ修正および機能拡張が数多く追加されました。主な変更点:

- **ssh-keygen** ツール用のデフォルトの RSA 鍵サイズを 3072 ビットに増加
- **ShowPatchLevel** 設定オプションのサポートを廃止
- Kerberos クリーンアップ手順の修正など、GSSAPI 鍵交換コードの修正を適用
- **sshd_net_t** SELinux コンテンツへのフォールバックを廃止
- **Match final** ブロックのサポートを追加
- **ssh-copy-id** コマンドのマイナーな問題点を修正
- **scp** ユーティリティに関連する Common Vulnerabilities and Exposures (CVE) を修正 (CVE-2019-6111, CVE-2018-20685, CVE-2019-6109)

このリリースでは、CVE-2019-6111 の緩和策として、**scp** の非互換性が導入されている点に注意してください。スクリプトが、scp ダウンロード中にパスの高度な bash デプロイメントに依存する場合は、**-T** スイッチを使用して、信頼できるサーバーに接続するときにもこれらの緩和策を一時的にオフにする

ことができます。

(BZ#1691045)

libssh がシステム全体の **crypto-policies** に準拠するようになりました。

libssh クライアントとサーバーが、`/etc/libssh/libssh_client.config` ファイルと `/etc/libssh/libssh_server.config` を自動的にロードするようになりました。この設定ファイルには、**libssh** バックエンドのシステム全体の **crypto-policies** コンポーネントで設定されるオプションと、`/etc/ssh/ssh_config` または `/etc/ssh/sshd_config` OpenSSH 設定ファイルで設定したオプションが含まれます。設定ファイルの自動読み込みにより、**libssh** は、**crypto-policies** により設定されたシステム全体の暗号化ポリシー設定を使用するようになりました。この変更により、アプリケーションで使用される暗号化アルゴリズムのセットの制御が簡素化されます。

(BZ#1610883, BZ#1610884)

FROMHOST のケースを維持する **rsyslog** のオプションを利用できるようになりました。

rsyslog サービスへの今回の更新では、**imudp** および **imtcp** モジュールの **FROMHOST** プロパティの大文字と小文字を保持する管理を行うためのオプションが導入されました。**preservecase** 値を **on** に設定すると、**FROMHOST** プロパティが大文字と小文字を区別する状態で処理されます。既存の設定を壊さないように、**preservecase** のデフォルト値は、**imtcp** には **on** で、**imudp** には **off** になっています。

(BZ#1614181)

6.1.6. ネットワーク

PMTU 検出およびルートのリダイレクトが **VXLAN** トンネルおよび **GENEVE** トンネルで対応するようになりました。

Red Hat Enterprise Linux (RHEL) 8.0 のカーネルは、Virtual Extensible LAN (VXLAN) および Generic Network Virtualization Encapsulation (GENEVE) トンネルの Internet Control Message Protocol (ICMP) および ICMPv6 メッセージを処理していませんでした。これにより、8.1 以前の RHEL リリースでは、パス MTU (PMTU) の検出およびルートのリダイレクトは VXLAN トンネルおよび GENEVE トンネルでサポートされていませんでした。今回の更新で、カーネルが ICMP の Destination Unreachable と Redirect Message、および ICMPv6 Packet Too Big エラーメッセージを、PMTU を調整すると転送情報を修正することで、Destination Unreachable エラーメッセージを処理するようになりました。これにより、RHEL 8.1 は、VXLAN と GENEVE トンネルによる PMTU 検出およびルートのリダイレクトに対応しています。

(BZ#1652222)

カーネルにおける **XDP** 機能とネットワーク **eBPF** 機能の主な変更点

kernel パッケージの **XDP** およびネットワーク **eBPF** 機能がアップストリームのバージョン 5.0 にアップグレードされ、以前のバージョンに対してバグ修正および機能拡張が数多く追加されました。

- eBPF プログラムが、TCP/IP スタックと適切に対話でき、フローの分散セクションを実行でき、多くの **bpf** ヘルパー関数が利用可能になり、新しいマップタイプにアクセスできるようになりました。
- XDP メタデータが `AF_XDP` ソケットで使用できるようになりました。

(BZ#1687459)

`ioctl()` の新しい `PTP_SYS_OFFSET_EXTENDED` コントロールにより、測定された `system-PHC` オフセットの正確性が改善されます。

この改善により、`ioctl()` 関数にシステム precision time protocol (PTP) ハードウェアクロック (PHC) オフセットの測定をさらに性格に調整するために、`PTP_SYS_OFFSET_EXTENDED` コントロールが追加されました。たとえば、`chrony` サービスが PHC とシステム間のオフセットを測定するのに使用する `PTP_SYS_OFFSET` コントロールは、十分に正確ではありません。新しい `PTP_SYS_OFFSET_EXTENDED` コントロールを使用では、ドライバーは最小のビットの読み込みを分離できます。これにより、測定されたオフセットの精度が改善されます。ネットワークドライバーは通常、複数の PCI レジスターを読み込みます。また、このドライバーは、システムクロックの2つの値において、PHC タイムスタンプの最も低いビットを読み取りません。

(BZ#1677215)

`ipset` がバージョン 7.1 にリベースされました。

`ipset` パッケージがアップストリームバージョン 7.1 にアップグレードされ、以前のバージョンのバグ修正および機能拡張が数多く追加されました。

- `ipset` プロトコルのバージョン 7 では、`IPSET_CMD_GET_BYNAME` および `IPSET_CMD_GET_BYINDEX` オペレーションが導入されました。また、ユーザー空間コンポーネントは、カーネルコンポーネントに対応する正確な互換性レベルを検出できるようになりました。
- メモリーリークや `user-after-free` バグなど、非常に多くのバグが修正されました。

(BZ#1649090)

6.1.7. カーネル

RHEL 8.1 のカーネルバージョン

Red Hat Enterprise Linux 8.1 には、カーネルバージョン 4.18.0-147 が同梱されています。

(BZ#1797671)

カーネルに対するライブパッチが利用可能になりました。

カーネル用のライブパッチ `kpatch` では、プロセスのリブートまたは再起動なしで、実行中のカーネルにパッチを当てるメカニズムを利用できます。ライブカーネルパッチは、影響度が重大および重要な CVE を修正するための [Extended Update Support \(EUS\)](#) の RHEL で対象となる一部のマイナーリリースストリームに提供されます。

カーネルの RHEL 8.1 バージョンの `kpatch` ストリームに登録するには、[RHEA-2019:3695](#) アドバイザリーによる `kpatch-patch-4_18_0-147` パッケージをインストールします。

詳細は、カーネルの管理、監視、および更新の [カーネルライブパッチを使用したパッチの適用](#) を参照してください。

(BZ#1763780)

RHEL 8 の Extended Berkeley Packet Filter

`eBPF` (extended Berkeley Packet Filter) は、一連の制限付きの関数にアクセスできる制限付きサンドボックス環境において、カーネル領域でのコード実行を可能にするカーネル内の仮想マシンです。この仮想マシンは、特別なアセンブリーのようなコードを実行します。そして、このコードはカーネルに

ロードされ、実行時コンパイラーでネイティブマシンコードに変換されます。eBPF 仮想マシンを使用する Red Hat には、多くのコンポーネントが同梱されています。各コンポーネントの開発フェーズはさまざまです。そのため、現在すべてのコンポーネントが完全にサポートされている訳ではありません。

RHEL 8.1 では、**BPF Compiler Collection (BCC)** ツールパッケージは、AMD および Intel 64 ビットアーキテクチャーで完全にサポートされています。BCC ツールパッケージは、eBPF 仮想マシンを使用する動的なカーネル追跡ユーティリティのコレクションです。

現在、以下の eBPF コンポーネントをテクノロジープレビューとして利用できます。

- 以下のアーキテクチャーの **BCC** ツールパッケージ: 64 ビット ARM アーキテクチャー、IBM Power Systems、リトルエンディアン、および IBM Z
- すべてのアーキテクチャーにおける **BCC** ライブラリー
- **bpftrace** トレース言語
- eXpress Data Path (XDP) 機能

テクノロジープレビューコンポーネントの詳細については、「[カーネル](#)」を参照してください。

(BZ#1780124)

Red Hat Enterprise Linux 8 が、**early kdump** に対応しました。

early kdump 機能により、早い段階でクラッシュしても **vmcore** 情報を取得するのに十分な早さでクラッシュカーネルと **initramfs** の読み込みが行われるようになりました。

early kdump の詳細は `/usr/share/doc/kexec-tools/early-kdump-howto.txt` ファイルを参照してください。

(BZ#1520209)

RHEL 8 が **ipcmni _extend** に対応しました。

Red Hat Enterprise Linux 8 に、新しいカーネルコマンドラインパラメーター **ipcmni _extend** が追加されました。このパラメーターは、現在の最大 32 KB (15 ビット) から 16 MB (24 ビット) までの一意の System V プロセス間通信 (IPC) 識別子の数を拡張します。その結果、アプリケーションが多くの共有メモリーセグメントを生成するユーザーは、32 KB の制限を超過することなく、より強力な IPC 識別子を作成できます。

ipcmni _extend を使用すると、多少のパフォーマンスのオーバーヘッドが生まれます。そのため、アプリケーションが 32 KB 以上の一意の IPC 識別子を必要とする場合にのみ使用するようにしてください。

(BZ#1710480)

永続メモリーの初期化コードが並列初期化に対応しました。

永続メモリーの初期化コードを使用すると、永続メモリーの複数のノードを持つシステムで並列初期化が可能になります。並列初期化では、大量の永続メモリーを使用するシステムの全体的なメモリー初期化時間が大幅に削減されます。その結果、このシステムの起動速度がより高速になりました。

(BZ#1634343)

TPM ユーザー空間ツールが最新バージョンに更新されました。

tpm2-tools ユーザー空間ツールがバージョン 2.0 に更新されました。この更新で、**tpm2-tools** が多くの不具合を修正できるようになりました。

(BZ#1664498)

rngd デーモンが、**root** 以外の特権で実行可能になりました。

乱数ジェネレーターデーモン (**rngd**) は、乱数性のソースによって供給されるデータが十分にランダムなものかどうかをチェックしてから、カーネルの乱数エントロピープールにデータを格納します。今回の更新で、システムセキュリティーを向上するために、**root** 以外のユーザー権限で **rngd** を実行できるようになりました。

(BZ#1692435)

ibmvnic ドライバーの完全サポート

Red Hat Enterprise Linux 8.0 の導入では、IBM POWER アーキテクチャー (**ibmvnic**) 用の IBM Virtual Network Interface Controller (vNIC) ドライバーがテクノロジープレビューとして利用できました。vNIC は、エンタープライズ機能を提供し、ネットワーク管理を簡素化する PowerVM 仮想ネットワークテクノロジーです。SR-IOV NIC と組み合わせると、仮想 NIC レベルで帯域幅制御サービス品質 (QoS) 機能が提供される、高性能で効率的な技術です。vNIC は、仮想化のオーバーヘッドを大幅に削減するため、ネットワーク仮想化に必要な CPU やメモリーなど、待機時間が短縮され、サーバーリソースが少なくなります。

Red Hat Enterprise Linux 8.1 以降、**ibmvnic** デバイスドライバーは IBM POWER9 システムで完全にサポートされています。

(BZ#1665717)

Intel® Omni-Path Architecture (OPA) ホストソフトウェア

Red Hat Enterprise Linux 8.1 は、Intel Omni-Path Architecture (OPA) ホストソフトウェアに完全に対応しています。Intel OPA は、クラスター環境のコンピュートと I/O ノード間の高性能データ転送 (高帯域幅、高メッセージレート、低レイテンシー) のために、初期化とセットアップを行う Host Fabric Interface (HFI) ハードウェアを提供します。

(BZ#1766186)

UBSan が、RHEL 8 のデバッグカーネルで有効化されました。

Undefined Behavior Sanitizer (UBSan) ユーティティーは、ランタイムでの C コード言語で未定義の動作の不具合を明らかにします。コンパイラーの動作が、開発者が予期したものと異なる場合があったため、このユーティティーがデバッグカーネルで有効になりました。特に、コンパイラーの最適化の場合は、詳細なバグが表示されます。これにより、**UBSan** を有効にしてデバッグカーネルを実行すると、システムがこのようなバグを簡単に検出できるようになります。

(BZ#1571628)

RHEL 8 で **fadump** インフラストラクチャーが RHEL 8 での再登録に対応しました。

このサポートは、クラッシュメモリー範囲を更新するメモリーホットアド/リムーブの操作後に、ファームウェア支援ダンプ (**fadump**) インフラストラクチャーの再登録 (登録解除と登録) を行うために追加されました。この機能は、**udev** イベント時に **fadump** をユーザー空間から登録と登録を行う際に、システムの潜在的な問題が発生することを防ぐことを目的としています。

(BZ#1710288)

determine_maximum_mpps.sh スクリプトが RHEL for Real Time 8 に導入されました。

queuelat テストプログラムの使用を容易にするために、**determine_maximum_mpps.sh** スクリプトが導入されました。このスクリプトは **queuelat** を実行して、マシンが 1 秒あたりに処理可能な最大パケットを判断します。

(BZ#1686494)

kernel-rt ソースツリーが、最新の RHEL 8 ツリーに一致するようになりました。

kernel-rt ソースが最新の Red Hat Enterprise Linux カーネルソースツリーをベースとするようにアップグレードされ、以前のバージョンのバグ修正および機能拡張が数多く追加されました。

(BZ#1678887)

ssdd テストが RHEL for Real Time 8 に追加されました。

ssdd テストが追加され、追跡サブシステムのストレステストが可能になりました。このテストは複数の追跡スレッドを実行して、追跡システム内でのロックが適切であることを確認します。

(BZ#1666351)

6.1.8. ハードウェアの有効化

Optane DC 永続メモリー技術用のメモリーモードに完全対応

Intel Optane DC Persistent Memory ストレージデバイスは、データセンタークラスの永続メモリー技術を提供し、トランザクションのスループットを大幅に向上させます。

メモリーモード技術を使用するために、システムに特別なドライバーや特定の認定を設定する必要ありません。メモリーモードは、オペレーティングシステムに対して透過的です。

(BZ#1718422)

IBM Z がシステムブート署名の検証に対応しました。

セキュアブートにより、システムファームウェアでカーネル空間コードの署名に使用された暗号鍵の信頼性をチェックできます。その結果、信頼できるベンダーのコードのみが実行可能なため、この機能によりセキュリティが強化されます。

IBM z15 はセキュアブートを使用する必要があることに注意してください。

(BZ#1659399)

6.1.9. ファイルシステムおよびストレージ

DIF/DIX (Data Integrity Field/Data Integrity Extension) への対応

DIF/DIX は、ハードウェアベンダーが認定している設定でサポートされ、RHEL では特定のホストバスアダプター (HBA) およびストレージアレイ設定に完全に対応しています。

DIF/DIX は、以下の設定ではサポートされません。

- ブートデバイス上での使用はサポートされない。
- 仮想化ゲストではサポートされない。
- DIF/DIX が有効な場合に Automatic Storage Management ライブラリー (ASMLib) を使用することは、Red Hat はサポートしていない。

DIF/DIX は、ストレージデバイスで有効または無効になります。これは、そのアプリケーションまでのさまざまな層 (そのアプリケーションも含む) に関与します。ストレージデバイスで DIF をアクティベートする方法は、デバイスによって異なります。

DIF/DIX 機能の詳細は [DIF/DIX \(別名 PI\) はどのような機能ですか？ Red Hat のサポート対象ですか？](#) を参照してください。

(BZ#1649493)

Optane DC メモリーシステムが **EDAC** レポートに対応するようになりました。

以前では、メモリーアドレスが NVDIMM モジュール内にある場合は、修正済みまたは未修正イベントを報告することができませんでした。今回の更新で、EDAC は正しいメモリーモジュール情報とともにイベントを適切に報告できるようになりました。

(BZ#1571534)

VDO Ansible モジュールが **Ansible** パッケージに移動しました。

以前では、VDO Ansible モジュールは **vdo** RPM パッケージで提供されていました。このリリースから、このモジュールは代わりに **ansible** パッケージで提供されます。

VDO Ansible モジュールファイルの元の場所は以下の通りです。

```
/usr/share/doc/vdo/examples/ansible/vdo.py
```

このファイルの新しい場所は、以下の通りです。

```
/usr/lib/python3.6/site-packages/ansible/modules/system/vdo.py
```

vdo パッケージは引き続き Ansible Playbook を配信します。

Ansible についての詳細は、<http://docs.ansible.com/> を参照してください。

(BZ#1669534)

Aero アダプターが完全対応になりました。

以前テクノロジープレビューとして利用できた以下の Aero アダプターが、完全にサポートされるようになりました。

- PCI ID 0x1000:0x00e2 and 0x1000:0x00e6 (**mpt3sas** ドライバーにより制御)
- PCI ID 0x1000:0x10e5 and 0x1000:0x10e6 (**megaraid_sas** ドライバーにより制御)

(BZ#1663281)

LUKS2 がオンライン再暗号化をサポートするようになりました。

Linux Unified Key Setup バージョン 2 (LUKS2) 形式は、デバイスが使用中の間に暗号化したデバイスの再暗号化に対応しています。たとえば、以下のタスクを実行するにあたり、デバイスでファイルシステムをアンマウントする必要はありません。

- ボリュームキーの変更
- 暗号化アルゴリズムの変更

暗号化されていないデバイスを暗号化する場合は、ファイルシステムのマウントを依然として解除する必要がありますが、暗号化は大幅に速くなります。暗号化の短い初期化後にファイルシステムを再マウントできます。

また、LUKS2 の再暗号化は、より回復力があります。再暗号化プロセスで、パフォーマンスやデータ保護の優先度を設定する複数のオプションを選択できます。

LUKS2 再暗号化を実行するには、**cryptsetup reencrypt** サブコマンドを使用します。Red Hat は、LUKS2 形式への **cryptsetup-reencrypt** ユーティリティーの使用推奨をやめました。

LUKS1 形式は、オンライン再暗号化に対応しておらず、**cryptsetup reencrypt** サブコマンドは LUKS1 と互換性がなことに注意してください。LUKS1 デバイスを暗号化または再暗号化するには、**cryptsetup-reencrypt** ユーティリティーを使用します。

ディスク暗号化の詳細は、[LUKS を使用したブロックデバイスの暗号化](#) ください。

([BZ#1676622](#))

RHEL 8 で利用可能な ext4 の新機能

RHEL 8 において、新しく完全に対応した ext4 の機能は次のとおりです。

- デフォルト以外の機能:
 - **project**
 - **quota**
 - **mmp**
- デフォルト以外のマウントオプション:
 - **bsddf|minixdf**
 - **grpuid|bsdgroups** および **nogrpuid|sysvgroups**
 - **resgid=n** および **resuid=n**
 - **errors={continue|remount-ro|panic}**
 - **commit=nrsec**
 - **max_batch_time=usec**
 - **min_batch_time=usec**
 - **grpquota|noquota|quota|usrquota**
 - **prjquota**
 - **DAX**
 - **lazytime|nolazytime**
 - **discard|nodiscard**
 - **init_itable|noinit_itable**
 - **jqfmt={vfsold|vfvsv0|vfvsv1}**
 - **usrjquota=aquota.user|grpjquota=aquota.group**

機能およびマウントオプションの詳細は、man ページの **ext4** を参照してください。その他の ext4 機能

またはマウントオプション、あるいはこれら両方、機能の組み合わせまたはマウントオプション、あるいはその両方は、Red Hat では完全にサポートされていない場合があります。Red Hat リリースで完全に対応していない機能またはマウントオプションが特別なワークロードに必要な場合は、Red Hat サポートまでご連絡ください。対応リストへの包含を検討させていただきます。

(BZ#1741531)

NVMe over RDMA が **IBM Coral** システムのターゲットモードでの **Infiniband** に対応しました。

RHEL 8.1 では、NVMe over RDMA が、NVMe PCIe 1 をカードにターゲットとして追加して、IBM Coral システムのターゲットモードでの **Infiniband** に対応するようになりました。

(BZ#1721683)

6.1.10. 高可用性およびクラスター

Pacemaker が、**concurrent-fencing** クラスタープロパティを **true** にデフォルト設定するようになりました。

複数のクラスターノードが同時にフェンシングされる必要があり、設定済みの異なるフェンスデバイスを使用する場合、Pacemaker は、以前と同様にシリアル化されるのではなく、フェンシングを同時に実行するようになりました。これにより、複数のノードがフェンスされる必要がある場合、大規模なクラスターへの復旧が素早くなる場合があります。

(BZ#1715426)

共有論理ボリュームの拡張では、すべてのクラスターノードでの更新が不要になりました。

今回のリリースで、共有論理ボリュームを拡張する際に、いずれかのクラスターノードで **lvextend** コマンドを実行した後、すべてのクラスターノードでの更新が不要になりました。GFS2 ファイルシステムのサイズを拡張するすべての手順は、[GFS2 ファイルシステムの拡張](#) を参照してください。

(BZ#1649086)

対応している **RHEL HA** クラスターの最大サイズが **16** ノードから **32** ノードに増大しました。

今回のリリースで、Red Hat は、最大 32 個の完全クラスターノードのクラスターデプロイメントに対応します。

(BZ#1693491)

corosync リンクを追加、変更、および削除するためのコマンドが **pcs** に追加されました

Kronosnet (knet) プロトコルにより、実行中のクラスターで knet リンクを追加および削除できるようになりました。この機能をサポートするために、**pcs** コマンドには、knet リンクを追加、変更、および削除するコマンドと、既存のクラスター内の upd/udpu リンクを変更するコマンドが追加されました。既存クラスターのリンクの追加および変更の詳細は、[既存クラスターへのリンクの追加および変更](#) を参照してください。(BZ#1667058)

6.1.11. 動的プログラミング言語、Web サーバー、およびデータベースサーバー

新しいモジュールストリーム: **php:7.3**

RHEL 8.1 では **PHP 7.3** が導入され、数多くの新機能および機能強化が追加されました。以下は、主な変更点です。

- 強化され、より柔軟になった **heredoc** と **nowdoc** 構文

- PCRE 拡張を PCRE2 にアップグレード
- 改善されたマルチバイト文字列処理
- LDAP コントロールのサポート
- 改善された FastCGI Process Manager (FPM) ロギング
- 非推奨および後方互換性のない変更

詳細は [Migrating from PHP 7.2.x to PHP 7.3.x](#) を参照してください。

RHEL 8 バージョンの **PHP 7.3** は、**Argon2** パスワードハッシュアルゴリズムに対応していません。

php:7.3 ストリームをインストールするには、以下を使用します。

```
# yum module install php:7.3
```

php:7.2 ストリームからアップグレードする場合は、[後続のストリームへの切り替え](#) を参照してください。

([BZ#1653109](#))

新しいモジュールストリーム: **ruby:2.6**

新しいモジュールストリーム **ruby:2.6** が利用できるようになりました。RHEL 8.1 に含まれている **Ruby 2.6.3** は、RHEL 8.0 で配布されていたバージョン 2.5 に対して新機能、機能拡張、バグ、セキュリティ修正などを多数提供しています。

主な機能強化は、以下の通りです。

- 定数名が、ASCII 以外の大文字で開始できるようになりました。
- 制限なしの範囲サポートが追加されました。
- 新しい **Binding#source_location** メソッドを利用できるようになりました。
- **\$SAFE** はプロセスのグローバル状態となります。これは **0** に戻すことができます。

以下のパフォーマンスの向上が実装されています。

- **Proc#call** および **block.call** プロセスが最適化されました。
- 新しいガベッジコレクターマネージドヒープの **theap** (Transient heap) が導入されました。
- 各アーキテクチャーに、コルーチンのネイティブ実装が導入されました。

また、**ruby:2.5** ストリームで提供される **Ruby 2.5** は、バージョン 2.5.5 にアップグレードされ、バグ修正およびセキュリティ修正が数多く追加されました。

ruby:2.6 ストリームをインストールするには、以下を実行します。

```
# yum module install ruby:2.6
```

php:2.5 ストリームからアップグレードする場合は、[後続のストリームへの切り替え](#) を参照してください。

(BZ#1672575)

新しいモジュールストリーム: **nodejs:12**

RHEL 8.1 では **Node.js 12** が導入され、バージョン 10 に対する新機能および機能強化が数多く追加されました。以下は、主な変更点です。

- V8 エンジンがバージョン 7.4 にアップグレード
- 新しいデフォルト HTTP パーサー、**llhttp** (実験対象外になりました)
- ヒープダンプ生成の統合機能
- ECMAScript 2015 (ES6) モジュールのサポート
- ネイティブモジュールのサポートの改善
- ワーカースレッドへのフラグが不要に
- 実験的な診断機能を新たに追加
- パフォーマンスの向上

nodejs:12 ストリームをインストールするには、以下を使用します。

```
# yum module install nodejs:12
```

nodejs:10 ストリームからアップグレードする場合は、[後続のストリームへの切り替え](#) を参照してください。

(BZ#1685191)

Judy-devel が **CRB** で利用可能になりました。

Judy-devel パッケージが、[CodeReady Linux Builder リポジトリ \(CRB\)](#) の **mariadb-devel:10.3** モジュールの一部として利用できるようになりました。そのため、開発者は **Judy** ライブラリーを使用してアプリケーションをビルドできるようになりました。

Judy-devel パッケージをインストールするには、まず **mariadb-devel:10.3** モジュールを有効にします。

```
# yum module enable mariadb-devel:10.3
# yum install Judy-devel
```

(BZ#1657053)

Python 3 の FIPS コンプライアンス

今回の更新で、OpenSSL FIPS モードのサポートが **Python 3** に追加されました。説明:

- FIPS モードでは、**blake2**、**sha3**、および **shake** ハッシュは OpenSSL ラッパーを使用します。ただし、拡張機能 (キー、ツリーハッシュ、カスタムダイジェストサイズなど) は提供しません。
- FIPS モードでは、**hmac.HMAC** クラスは OpenSSL ラッパーまたは OpenSSL ハッシュ名が **digestmod** 引数として指定された文字列でのみインスタンス化できます。**md5** アルゴリズムのデフォルトの代わりに、引数を指定する必要があります。

ハッシュ関数は、OpenSSL FIPS モードでセキュアでないハッシュの使用を可能にする、**usedforsecurity** 引数に対応していることに注意してください。ユーザーは、関連するすべての標準への準拠を確実にする必要があります。

(BZ#1731424)

python3-wheel における FIPS コンプライアンスの変更

python3-wheel パッケージの今回の更新で、FIPS に準拠していないデータの署名および検証を行うための組み込みの実装が削除されます。

(BZ#1731526)

新しいモジュールストリーム: **nginx:1.16**

nginx 1.16 Web およびプロキシサーバーが利用できるようになりました。このサーバーでは、1.14 に対する数多くの新しい機能や強化を提供します。以下に例を示します。

- SSL に関連する多くの更新 (変数からの SSL 証明書および秘密鍵のロード。 **ssl_certificate** ディレクティブおよび **ssl_certificate_key** ディレクティブ、新しい **ssl_early_data** ディレクティブにおける変数サポート)
- 新しい **keepalive** 関連のディレクティブ
- 負荷分散を分配するための新しい **random** ディレクティブ
- 既存のディレクティブに対する新パラメーターおよび改善 (**listen** ディレクティブのポート範囲、**limit_req** ディレクティブの新しい **delay** パラメーター。2 ステージのレート制限を有効化)
- 新しい **\$upstream_bytes_sent** 変数
- UDP (User Datagram Protocol) プロキシの改善

その他の主な変更点は次の通りです。

- **nginx:1.16** ストリームでは、**nginx** パッケージには **nginx-all-modules** パッケージは必要ありません。したがって、**nginx** モジュールを明示的にインストールする必要があります。**nginx** をモジュールとしてインストールする場合、**nginx-all-modules** パッケージは、デフォルトのプロファイルである **common** プロファイルの一部としてインストールされます。
- **ssl** ディレクティブが非推奨となり、代わりに **listen** ディレクティブに **ssl** パラメーターを使用します。
- **nginx** が、設定テスト時に欠如している SSL 証明書を検出するようになりました。
- **listen** ディレクティブでホスト名を使用する場合、**nginx** が、ホスト名が解決するすべてのアドレスに対してリスニングソケットを作成します。

nginx:1.16 ストリームをインストールするには、以下を実行します。

```
# yum module install nginx:1.16
```

nginx:1.14 ストリームからアップグレードする場合は、[後続のストリームへの切り替え](#) を参照してください。

(BZ#1690292)

perl-IO-Socket-SSL がバージョン 2.066 にリベースされました。

perl-IO-Socket-SSL パッケージがバージョン 2.066 にアップグレードされ、以前のバージョンのバグ修正および機能拡張が数多く追加されました。例:

- TLS 1.3 のサポートが改善されました。特に、セッション再利用、クライアントサイドでの自動ハンドシェイク認証が改善されています。
- 複数の曲線、曲線の自動設定、部分的な信頼チェーン、および同一ドメインの RSA 証明書および ECDSA 証明書のサポートを追加しました。

(BZ#1632600)

perl-Net-SSLeay がバージョン 1.88 にリベースされました。

perl-Net-SSLeay パッケージがバージョン 1.88 にアップグレードされ、バグ修正および機能強化が複数追加されました。以下は、主な変更点です。

- 証明書および X509 ストアのスタックの操作、楕円曲線およびグループの選択など、OpenSSL 1.1.1 との互換性を強化
- TLS 1.3 との互換性の向上 (セッションの再使用、ハンドシェイク認証など)
- **cb_data_advanced_put()** サブルーチンでのメモリー漏えいが修正されました。

(BZ#1632597)

6.1.12. コンパイラーおよび開発ツール

GCC Toolset 9 が利用可能になりました。

Red Hat Enterprise Linux 8.1 では、最新バージョンの開発ツールを含むアプリケーションストリームの GCC Toolset 9 が導入されました。

以下のツールおよびバージョンは、GCC Toolset 9 で利用できます。

ツール	バージョン
GCC	9.1.1
GDB	8.3
Valgrind	3.15.0
SystemTap	4.1
Dyninst	10.1.0
binutils	2.32
elfutils	0.176
dwz	0.12

ツール	バージョン
make	4.2.1
strace	5.1
ltrace	0.7.91
annobin	8.79

GCC Toolset 9 は、**AppStream** リポジトリ内の Software Collection の形式で、Application Stream として利用できます。GCC Toolset は、RHEL 7 の [Red Hat Developer Toolset](#) に類似したツールセットです。

GCC Toolset 9 をインストールするには、以下のコマンドを実行します。

```
# yum install gcc-toolset-9
```

GCC Toolset 9 のツールを実行するには、以下のコマンドを実行します。

```
$ scl enable gcc-toolset-9 tool
```

GCC Toolset 9 のツールバージョンが、これらのツールのシステムバージョンよりも優先されるシェルセッションを実行するには、以下のコマンドを実行します。

```
$ scl enable gcc-toolset-9 bash
```

使用方法の詳細は [GCC Toolset の使用](#) を参照してください。

(BZ#1685482)

アップグレードされたコンパイラツールセット

アプリケーションストリームとして配布されている以下のコンパイラツールセットが、RHEL 8.1 でアップグレードされました。

- LLVM コンパイラインフラストラクチャーフレームワーク、C および C++ 言語用の Clang コンパイラ、LLDB デバッガー、コード分析用の関連ツールをバージョン 8.0.1 に提供する Clang と LLVM Toolset。
- Rust プログラミング言語コンパイラ **rustc**、**cargo** ビルドツールおよび依存マネージャー、および必要なライブラリーをバージョン 1.37 に提供します。
- Go Toolset は、Go (**golang**) プログラミング言語ツールおよびライブラリーをバージョン 1.12.8 に提供します。

(BZ#1731502, BZ#1691975, BZ#1680091, BZ#1677819, BZ#1681643)

SystemTap がバージョン 4.1 にリベースされました。

SystemTap インストラクションツールが、アップストリームバージョン 4.1 に更新されました。以下は、主な改善点です。

- eBPF ランタイムバックエンドは、文字列変数やリッチフォーマット印刷などのスクリプト言語のより多くの機能を処理できます。
- トランスレーターのパフォーマンスが大幅に改善されました。
- 最適化された C コードでさらにタイプのデータを DWARF4 debuginfo コンストラクトで抽出できるようになりました。

(BZ#1675740)

DHAT ツールの一般公開

Red Hat Enterprise Linux 8.1 では、**DHAT** ツールが一般公開されました。これは、**valgrind** ツールバージョン 3.15.0 をベースにしています。

valgrind ツール機能の変更および実行機能は以下のとおりです。

- `--tool=exp-dhat` の代わりに `--tool=dhat` を使用します。
- **dchat** ツールがプログラム終了後に最低限のデータを出力するため、`--show-top-n` オプションおよび `--sort-by` オプションは削除されました。
- JavaScript プログラムである新しいビューアーの **dh_view.html** には、プロファイル結果が含まれます。短いメッセージでは、実行終了後に結果を表示する方法が説明されています。
- ビューアーのドキュメントは `/usr/libexec/valgrind/dh_view.html` にあります。
- **DHAT** ツールに関するドキュメントは、`/usr/share/doc/valgrind/html/dh-manual.html` にあります。
- amd64 (x86_64) のサポート: **RDRAND** および **F16C insn** セットの拡張機能が追加されました。
- **cachegrind** では、**cg_annotate** コマンドに新たなオプション `--show-percs` が追加されました。これは、すべてのイベント数の横にパーセンテージを出力します。
- **callgrind** では、**callgrind_annotate** コマンドに新たなオプション `--show-percs` が追加されました。これは、すべてのイベント数の横にパーセンテージを出力します。
- **massif** では、`--read-inline-info` のデフォルト値が **yes** になりました。
- 出力リークが **xtree** 形式になる **memcheck** オプションの `--xtree-leak=yes` では、`--show-leak-kinds=all` オプションが自動的に有効になります。
- 新しいオプションの `--show-error-list=no|yes` を指定すると、実行の最後に、検出されたエラーと使用した抑制が表示されます。以前では、ユーザーは **valgrind** コマンドに `-v` オプションを指定できましたが、混乱する可能性のある情報が多く表示されていました。オプション `-s` は、`--show-error-list=yes` オプションと同等の意味を持ちます。

(BZ#1683715)

elfutils がバージョン 0.176 にリベースされました。

elfutils パッケージが、アップストリームバージョン 0.176 に更新されました。このバージョンでは、さまざまなバグ修正が提供され、以下の脆弱を解決しています。

- [CVE-2019-7146](#)

- [CVE-2019-7149](#)
- [CVE-2019-7150](#)
- [CVE-2019-7664](#)
- [CVE-2019-7665](#)

以下は、主な改善点です。

- **libdw** ライブラリーは、**dwelf_elf_begin()** 関数で拡張されています。これは、圧縮ファイルを扱う **elf_begin()** のバリエーションです。
- 新しい **--reloc-debug-sections-only** オプションが **eu-strip** ツールに追加され、その他のストライプなしでデバッグセクション間の簡単な再配置をすべて解決します。この機能は、特定の状況における **ET_REL** ファイルにのみ関連します。

(BZ#1683705)

glibc の追加メモリー割り当てチェック

アプリケーションメモリーの破損は、アプリケーションとセキュリティー上の問題を引き起こします。検出のコストとの釣り合いの破損などの早期検出では、アプリケーション開発者は大きなメリットが得られます。

検出を改善するために、RHEL のコア C ライブラリーである GNU C ライブラリー (**glibc**) の **malloc** メタデータに、さらに 6 つのメモリー破損チェックが追加されました。これらの追加チェックは、ランタイムパフォーマンスをさほど犠牲にせずに追加されました。

(BZ#1651283)

GDB はより多くの **POWER8** レジスターにアクセスできます。

今回の更新で、GNU デバッガー (GDB) とリモートのスタブ **gdbserver** は、以下の追加レジスターにアクセスでき、IBM の **POWER8** プロセッサ行のセットを登録します。

- **PPR**
- **DSCR**
- **TAR**
- **EBB/PMU**
- **HTM**

(BZ#1187581)

binutils 逆アセンブラーが **NFP** バイナリーファイルを処理できるようになりました。

binutils パッケージの逆アセンブリツールが、Netronome Flow Processor (NFP) ハードウェアシリーズのバイナリーファイルを処理するように拡張されました。この機能は、BPF (**bpftool** Berkeley Packet Filter) コードコンパイラーでさらなる機能を有効にする必要があります。

(BZ#1644391)

部分的に書き込み可能な **GOT** セクションが **IBM Z** アーキテクチャーでサポートされるようになりました。

ローダーのレイジーバインディング機能を使用する IBM Z バイナリーは、部分的に書き込み可能なグローバルオフセットテーブル (GOT) セクションを生成することで強化できるようになりました。これらのバイナリーには、読み取り/書き込み可能な GOT が必要ですが、すべてのエントリーを書き込み可能にする必要はありません。今回の更新により、潜在的な攻撃からエントリーを保護できるようになりました。

(BZ#1525406)

binutils が IBM Z の Arch13 プロセッサに対応するようになりました。

今回の更新で、Arch13 プロセッサに関連する拡張機能のサポートが IBM Z アーキテクチャーの **binutils** パッケージに追加されました。その結果、IBM Z の arch13 対応 CPU で利用可能な機能を使用できるカーネルを構築できるようになりました。

(BZ#1659437)

dyninst がバージョン 10.1.0 にリベースされました。

Dyninst インストゥルメンテーションライブラリーがアップストリームバージョン 10.1.0 に更新されました。以下は、主な変更点です。

- Dyninst は、Linux PowerPC Little Endian (**ppcle**) および 64 ビット ARM (**aarch64**) アーキテクチャーに対応します。
- パラレルコード分析を使用することで、開始時間を改善しました。

(BZ#1648441)

日本語の令和に対する日付形式が更新されました。

GNU C ライブラリーは、2019 年 5 月 1 日をもって、令和に正しい日本語の年号フォーマットを利用できるようになりました。**strftime** 関数および **strptime** 関数によって使用されるデータなど、API データを処理する時間が更新されました。**strftime** が **%EC**、**%EY**、または **%Ey** など、いずれかの年号の変換指定子とともに使用されると、すべての API は令和時代を正しく出力します。

(BZ#1577438)

Performance Co-Pilot がバージョン 4.3.2 にリベースされました。

RHEL 8.1 では、Performance Co-Pilot (PCP) ツールがアップストリームバージョン 4.3.2 に更新されました。以下は、主な改善点です。

- Linux カーネルエントロピー、Pressure Stall Information、Nvidia GPU 統計など、新しいメトリックが追加されました。
- **pcp-dstat**、**pcp-atop**、**perfevent** PMDA などのツールが、新しいメトリックを報告するように更新されました。
- Grafana によるパフォーマンス PCP 統合を行うための **pmseries** ユーティリティーおよび **pmproxy** ユーティリティーが更新されました。

このリリースは、ライブラリー、有線プロトコル、オンディスク PC アーカイブフォーマットと後方互換性があります。

(BZ#1685302)

6.1.13. ID 管理

IdM が、インストールおよび管理用の **Ansible** ロールとモジュールに対応しました。

今回の更新で、Identity Management (IdM) のデプロイメントおよび管理の Ansible ロールとモジュールを提供する **ansible-freeipa** パッケージが導入されました。Ansible ロールを使用して、IdM サーバー、レプリカ、およびクライアントのインストールとアンインストールを行うことができます。Ansible モジュールを使用して、IdM グループ、トポロジー、およびユーザーを管理できます。また、サンプル Playbook も用意されています。

今回の更新で、IdM ベースのソリューションのインストールおよび設定が容易になります。

(JIRA:RHELPLAN-2542)

IdM デプロイメントの全体的な適合性をテストするための新しいツール: **healthcheck**

今回の更新で、Identity Management (IdM) に **Healthcheck** ツールが追加されました。このツールでは、現在の IdM サーバーが正しく設定され、正しく実行されていることを確認するテストを利用できます。

現在対象のチェック項目: * 証明書の設定と有効期限 * レプリケーションエラー * レプリケーショントポロジー * AD Trust 設定 * サービスステータス * 重要な設定ファイルのファイルパーミッション * ファイルシステムの容量

The **Healthcheck** ツールは、コマンドラインインターフェイス (CLI) で利用できます。

(JIRA:RHELPLAN-13066)

IdM は、サーバーがオフライン時の期限切れのシステム証明書の更新を行えるようになりました。

この機能強化により、Identity Management (IdM) がオフラインのときでも、管理者は期限が切れたシステムの証明書を更新できます。システム証明書の期限が切れると、IdM が起動できません。新しい **ipa-cert-fix** コマンドは、新しいプロセスを続行するために日付を手動で設定する回避策に取って代わります。その結果、上述のシナリオのダウンタイムとサポートコストが低減します。

(JIRA:RHELPLAN-13074)

Identity Management が **Windows Server 2019** との信頼に対応しました。

Identity Management を使用すると、Windows Server 2019 によって実行される Active Directory フォレストへの、対応フォレストトラストを確立できるようになりました。対応のフォレストおよびドメイン機能レベルは変更されず、最大レベル Windows Server 2016 まで対応されます。

(JIRA:RHELPLAN-15036)

Samba がバージョン **4.10.4** にリベースされました。

samba パッケージがアップストリームバージョン 4.10.4 にアップグレードされ、以前のバージョンのバグ修正および機能拡張が数多く追加されました。

- Samba 4.10 は Python 3 に完全に対応します。今後の Samba バージョンには、Python 2 のランタイムサポートが含まれないことに注意してください。
- JSON (JavaScript Object Notation) ロギング機能が、認証メッセージの Windows イベント ID とログオンタイプをログに記録するようになりました。
- ユーザー空間における新たな **vfs_glusterfs_fuse** ファイルシステム (FUSE) のモジュールにより、Samba が GlusterFS ポリウムにアクセスする際にパフォーマンスが向上します。このモジュールを有効にするには、`/etc/samba/smb.conf` ファイルの共有の **vfs_objects** パラメー

ターに **glusterfs_fuse** を追加します。**vfs_glusterfs_fuse** は、既存の **vfs_glusterfs** モジュールに置き換えられるわけではないことに注意してください。

- サーバーメッセージブロック (SMB) クライアントの Python バインディングは非推奨となり、今後の Samba リリースでは削除されます。これは、Samba Python バインディングを使用して独自のユーティリティーを書き込むユーザーにのみ適用されます。

smbd サービス、**nmbd** サービス、または **winbind** サービスが起動すると、Samba は **tdb** データベースファイルを自動的に更新します。Samba を起動する前にデータベースファイルをバックアップします。Red Hat は、**tdb** データベースファイルのダウングレードをサポートしていないことに注意してください。

主な変更の詳細は、更新前のアップストリームのリリースノート <https://www.samba.org/samba/history/samba-4.10.0.html> をお読みください。

(BZ#1638001)

OpenLDAP のシステム全体の証明書ストアの場所を更新しました。

OpenLDAP の信頼できる CA のデフォルトの場所が、**/etc/openldap/certs** ではなく、システム全体の証明書ストア (**/etc/pki/ca-trust/source**) を使用するよう更新されました。この変更により、CA 信頼の設定が簡素化されています。

サービス固有の要件がない限り、CA 信頼の設定に特別な設定は不要です。たとえば、LDAP サーバーの証明書を LDAP クライアント接続用にのみ信頼する必要があるとします。この場合は、事前に CA 証明書を設定する必要があります。

(JIRA:RHELPLAN-7109)

IdM CRL マスターの管理を簡素化するために、新しい **ipa-crl-generation** コマンドが導入されました。

今回の更新で、**ipa-crl-generation status/enable/disable** コマンドが導入されました。これらのコマンドは、root ユーザーで実行すると、IdM の Certificate Revocation List (CRL) で簡単に機能します。以前は、CRL 生成マスターの、ある IdM CA サーバーから別のサーバーへの移動が、手動で行われ、エラーを引き起こし易い手順でした。

ipa-crl-generation status コマンドは、現在のホストが CRL 生成マスターかどうかを確認します。**ipa-crl-generation enable** コマンドは、現在のホストが IdM CA サーバーの場合、現在のホストを IdM の CRL 生成マスターにします。**ipa-crl-generation disable** コマンドは、現在のホストでの CRL 生成を停止します。

さらに、**ipa-server-install --uninstall** コマンドには、ホストが CRL 生成マスターであるかどうかについての安全確認が含まれるようになりました。このように、IdM は、システム管理者がトポロジーから CRL 生成マスターを削除しないようにします。

(JIRA:RHELPLAN-13068)

keycloak-httpd-client-install における **OpenID Connect** サポート

keycloak-httpd-client-install アイデンティティプロバイダーは以前、**mod_auth_mellon** 認証モジュールによる SAML (Security Assertion Markup Language) 認証のみをサポートしていました。このリリースでは **mod_auth_openidc** 認証モジュールサポートが導入され、OpenID Connect 認証の設定が可能になります。

keycloak-httpd-client-install アイデンティティプロバイダーでは、**mod_auth_openidc** を設定して、apache インスタンスを OpenID Connect クライアントとして設定できます。

(BZ#1553890)

非表示のレプリカとして **IdM** を設定することがテクノロジープレビューとして利用可能になりました。

この機能拡張により、管理者は Identity Management (IdM) レプリカを隠しレプリカとして設定できるようになりました。隠しレプリカは、稼働中および利用できるすべてのサービスを持つ IdM サーバーです。ただし、DNS のサービスに **SRV** レコードが存在せず、LDAP サーバーロールが有効になっていないため、他のクライアントやマスターには通知されません。そのため、クライアントはサービス検出を使用して隠しレプリカを検出することはできません。

隠しレプリカは主にクライアントを中断できる専用のサービス用に設計されています。たとえば、IdM の完全バックアップは、マスターまたはレプリカ上のすべての IdM サービスをシャットダウンする必要があります。非表示のレプリカを使用するクライアントはないため、管理者はクライアントに影響を与えることなく、このホスト上のサービスを一時的にシャットダウンできます。その他のユースケースには、大量インポートや詳細なクエリーなど、IdM API または LDAP サーバーの高負荷操作が含まれます。

隠しレプリカを新たにインストールするには、**ipa-replica-install -- hidden-replica** コマンドを使用します。既存のレプリカの状態を変更するには、**ipa server-state** コマンドを使用します。

(BZ#1719767)

SSSD がデフォルトで **AD GPO** を強制するようになりました。

SSSD オプション **ad_gpo_access_control** のデフォルト設定が **Enforcing** になりました。RHEL 8 では、**SSSD** は、デフォルトで Active Directory Group Policy Objects (GPO) に基づいてアクセス制御ルールを強制します。

Red Hat は、RHEL 7 から RHEL 8 にアップグレードする前に、Active Directory で GPO が正しく設定されていることを確認することを推奨します。GPO を強制しない場合は、**/etc/sss/sss.conf** ファイルの **ad_gpo_access_control** オプションの値を **permissive** に変更します。

(JIRA:RHELPLAN-51289)

6.1.14. デスクトップ

GNOME クラシックでのワークスペーススイッチの変更を行いました。

GNOME クラシック環境のワークスペーススイッチが変更されました。このスイッチは、一番下のバーの右部分に移動され、横線のサムネイルとして設計されています。ワークスペース間の切り替えは、必要なサムネイルをクリックすることで可能です。または、**Ctrl+Alt+down/up arrow** キーの組み合わせを使用してワークスペース間を切り換えることもできます。アクティブなワークスペースの内容は、**ウィンドウリスト**の形式で、下部のバーの左側に表示されます。

特定のワークスペース内で **Super** キーを押すと、このワークスペースで開いているすべてのウィンドウを含む、**ウィンドウピッカー** が表示されます。ただし、**ウィンドウピッカー** には、以前の RHEL リリースで使用できた以下の要素が表示されなくなりました。

- **dock** (画面の左端にある垂直バー)
- **ワークスペーススイッチ** (画面の右側の垂直バー)
- **検索エントリー**

これらの要素で以前に達成されていた特定のタスクには、以下の方法が採用されます。

- アプリケーションを起動するには、**dock** を使うのではなく、以下を行います。

- トップバーで **アプリケーション** メニューを使用する
- `kdb:[Alt+F2]` キーを押すと、**Enter a Command** 画面が表示されます。この画面で実行可能ファイルの名前を書き込みます。
- 垂直の **ワークスペーススイッチ** を使用せずにワークスペース間の切り換えるには、右下部のバーの水平の **ワークスペーススイッチ** を使用します。
- **検索エントリ** または垂直の **ワークスペーススイッチ** が必要な場合は、GNOME クラシックではなく GNOME Standard 環境を使用します。

(BZ#1704360)

6.1.15. グラフィックインフラストラクチャー

DRM が Linux カーネルバージョン 5.1 にリベースされました。

DRM (**Direct Rendering Manager**) カーネルグラフィックサブシステムが、アップストリームの Linux カーネルバージョン 5.1 にリベースされ、バグ修正および拡張機能が数多く追加されました。以下に例を示します。

- **mgag200** ドライバーが更新されました。このドライバーは、Matrox G200 eh3 GPU を使用する HPE Proliant Gen10 システムのサポートを継続的に提供します。この更新されたドライバーは、現在および新しい Dell EMC PowerEdge サーバーにも対応しています。
- NVIDIA GPU を使用する現在および今後の Lenovo プラットフォームにハードウェア対応を提供するように、**nouveau** ドライバーが更新されました。
- 現在および新しい Intel GPU のサポートを継続するために、**i915** ディスプレイドライバーが更新されました。
- Aspeed ASt BMC ディスプレイチップのバグ修正が追加されました。
- AMD Raven 2 セットの Accelerated Processing Units (APU) のサポートが追加されました。
- AMD Picasso APU のサポートが追加されました。
- AMD Vega GPU のサポートが追加されました。
- Intel Amber Lake-Y および Intel Comet Lake-U GPU のサポートが追加されました。

(BZ#1685552)

AMD Picasso グラフィックカードのサポート

今回の更新で、**amdgpu** グラフィックドライバーが導入されました。これにより、AMD Picasso グラフィックカードが RHEL 8 で完全にサポートされるようになりました。

(BZ#1685427)

6.1.16. Web コンソール

SMT の有効化と無効化

RHEL 8 では、同時マルチスレッド (SMT) 設定が利用できるようになりました。Web コンソールで SMT を無効にすると、以下のような CPU セキュリティー脆弱性を軽減できます。

- [Microarchitectural Data Sampling](#)
- [L1 Terminal Fault Attack](#)

([BZ#1678956](#))

Services ページの検索ボックスの追加

Services ページには、以下別にサービスをフィルタリングするための検索ボックスがあります。

- 名前
- 説明
- 状態

さらに、サービスの状態が1つのリストにマージされました。また、ページ上部のスイッチボタンもタブに変更され、サービス ページのユーザーエクスペリエンスが向上されました。

([BZ#1657752](#))

ファイアウォールゾーンのサポートの追加

Networking ページのファイアウォール設定が、以下に対応するようになりました。

- ゾーンの追加および削除
- 任意のゾーンへのサービスの追加または削除
- **firewalld** サービスに加え、カスタムポートの設定。

([BZ#1678473](#))

仮想マシン設定の改善の追加

この更新では、RHEL 8 Web コンソールに、Virtual Machines ページの多くの改善が含まれます。以下が可能になります。

- さまざまなタイプのストレージプールの管理
- 仮想マシンの自動起動の設定
- 既存の qcow イメージのインポート
- PXE ブートによる仮想マシンのインストール
- メモリー割り当ての変更
- 仮想マシンの一時停止/再開
- キャッシュ特性の設定 (directsync、ライトバック)
- ブート順序の変更

([BZ#1658847](#))

6.1.17. Red Hat Enterprise Linux システムロール

新しいストレージ ロールが RHEL システムロールに追加されました。

rhel-system-roles パッケージが提供する RHEL システム ロールに、**ストレージ** ロールが追加されました。ストレージ ロールは、Ansible を使用してローカルストレージを管理するために使用できます。

現時点で、**ストレージ** ロールは以下のタイプに対応しています。

- ディスク全体におけるファイルシステムの管理
- LVM ボリュームグループの管理
- 論理ボリュームとそのファイルシステムの管理

詳細は、[ファイルシステムの管理](#) および [論理ボリュームの設定および管理](#) を参照してください。

(BZ#1691966)

6.1.18. 仮想化

WALinuxAgent がバージョン 2.2.38 にリベース

WALinuxAgent パッケージがアップストリームバージョン 2.2.38 にアップグレードされました。これにより、以前のバージョンに対して多数のバグ修正と機能拡張が追加されています

さらに、**WALinuxAgent** は Python 2 と互換性がなくなり、アプリケーションは Python 2 に依存しています。その結果、Python 2 で記述されたアプリケーションと拡張機能は、Python 3 に変換して **WALinuxAgent** との互換性を確立する必要があります。

(BZ#1722848)

Windows は必要な **virtio-win** ドライバーを自動的に検出します。

Windows は **virtio-win** ドライバーを自動的に見つけることで、ユーザーがこれらの配置フォルダーを選択しなくても、**ドライバー ISO** から必要なドライバーを自動的に見つけることができます。

[BZ#1223668](#)

KVM が、5 レベルのページングをサポートします。

Red Hat Enterprise Linux 8 では、**KVM** 仮想化は、5 レベルのページング機能をサポートします。これにより、選択したホスト CPU で、ホストおよびゲストのシステムが使用できる物理アドレスおよび仮想アドレス空間が大幅に増加します。

(BZ#1526548)

スマートカード共有が **ActiveClient** ドライバーにより **Windows** ゲストで対応するようになりました。

今回の更新で、Windows ゲスト OS および **ActiveClient** ドライバーを使用する仮想マシンでスマートカードの共有のサポートが追加されました。これにより、これらの仮想マシンでエミュレートされたスマートカードまたは共有スマートカードを使用したユーザーログインに対するスマートカード認証が有効になります。

BZ#1615840

virt-xml に新しいオプションが追加されました。

virt-xml ユーティリティーで、以下のコマンドラインオプションを指定できるようになりました。

- **--no-define - virt-xml** コマンドによる仮想マシン (VM) への変更は永続設定に保存されません。
- **--start** - 要求された変更を行った後に、仮想マシンを起動します。

これらの2つのオプションを指定して、ユーザーは仮想マシンの設定を変更し、変更を永続化せずに、新しい設定で仮想マシンを起動します。たとえば、次の起動で **testguest** 仮想マシンの起動順序がネットワークに変更され、起動を開始します。

```
virt-xml testguest --start --no-define --edit --boot network
```

(JIRA:RHELPLAN-13960)

KVM に対応する IBM z14 GA2 CPU

この更新により、KVM は IBM z14 GA2 CPU モデルに対応しています。これにより、RHEL 8 を使用する IBM z14 GA2 ホストで、ゲスト内に IBM z14 GA2 CPU を持つホスト OS として仮想マシンを作成できます。

(JIRA:RHELPLAN-13649)

NVIDIA NVLink2 が、**IBM POWER9** の仮想マシンと互換性を持つようになりました。

NVLink2 機能に対応した NVIDIA vGPU が、IBM POWER9 システムの RHEL 8 ホストで実行している仮想マシンに割り当てられるようになりました。これにより、これらの仮想マシンが NVLink2 をフルに活用できるようになりました。

(JIRA:RHELPLAN-12811)

6.2. 新しいドライバー

ネットワークドライバー

- シリアルラインインターネットプロトコルのサポート (slip.ko.xz)
- Bosch C_CAN コントローラー用のプラットフォームの CAN バスドライバー (c_can_platform.ko.xz)
- 仮想 CAN インターフェイス (vcan.ko.xz)
- Softing DPRAM CAN ドライバー (softing.ko.xz)
- シリアルライン CAN インターフェイス (slcan.ko.xz)
- EMS Dr.用 CAN ドライバー-Thomas Wuensche CAN/USB インターフェイス (ems_usb.ko.xz)
- esd CAN-USB/2 および CAN-USB/Micro インターフェイス用の CAN ドライバー (esd_usb2.ko.xz)
- プラットフォームバス上の SJA1000 用ソケット CAN ドライバー (sja1000_platform.ko.xz)
- SJA1000 チップを搭載した PLX90xx PCI ブリッジカード用のソケット CAN ドライバー (plx_pci.ko.xz)
- EMS CPC-PCI/PCIe/104P CAN カード用ソケット CAN ドライバー (ems_pci.ko.xz)
- KVASER PCAN PCI カード用の Socket-CAN ドライバー (kvaser_pci.ko.xz)

- Intel® 2.5G Ethernet Linux ドライバー (igc.ko.xz)
- Realtek 802.11ac ワイヤレス PCI ドライバー (rtwpci.ko.xz)
- Realtek 802.11ac ワイヤレスコアモジュール (rtw88.ko.xz)
- MediaTek MT76 デバイスのサポート (mt76.ko.xz)
- MediaTek MT76x0U (USB) サポート (mt76x0u.ko.xz)
- MediaTek MT76x2U (USB) サポート (mt76x2u.ko.xz)

グラフィックスドライバーおよびその他のドライバー

- Virtual Kernel Mode Setting (vkms.ko.xz)
- Intel GTT (Graphics Translation Table) ルーチン (intel-gtt.ko.xz)
- Xen フロントエンド/バックエンドページディレクトリベースの共有バッファ処理 (xen-front-pgdir-shbuf.ko.xz)
- オーディオミュートコントロール用の LED トリガー (ledtrig-audio.ko.xz)
- Host Wireless Adapter Radio Control ドライバー (hwa-rc.ko.xz)
- ネットワークブロックデバイス (nbd.ko.xz)
- Pericom PI3USB30532 Type-C mux ドライバー (pi3usb30532.ko.xz)
- Fairchild FUSB302 Type-C Chip Driver (fusb302.ko.xz)
- TI TPS6598x USB Power Delivery Controller ドライバー (tps6598x.ko.xz)
- Intel PCH Thermal ドライバー (intel_pch_thermal.ko.xz)
- PCIe AER ソフトウェアエラーインジェクター (aer_inject.ko.xz)
- PCI SR-IOV PF デバイス用の単純なスタブドライバー (pci-pf-stub.ko.xz)
- mISDN デジタルオーディオ処理のサポート (mISDN_dsp.ko.xz)
- Cologne Chip HFC-4S/8S チップ用の ISDN レイヤー 1 (hfc4s8s_l1.ko.xz)
- ISDN4Linux: 通話転送のサポート (dss1_divert.ko.xz)
- CAPI4Linux: ユーザー空間/dev/capi20 インターフェイス (capi.ko.xz)
- Gigaset 307x 用 USB ドライバー (bas_gigaset.ko.xz)
- ISDN4Linux: HYSN カード用ドライバー (hysdn.ko.xz)
- mISDN デジタルオーディオ処理のサポート (mISDN_dsp.ko.xz)
- Winbond w6692 ベースのカード用の mISDN ドライバー (w6692.ko.xz)
- CCD の hfc-pci ベースのカード用の mISDN ドライバー (hfcpci.ko.xz)
- hfc-4s/hfc-8s/hfc-e1 ベースのカード用の mISDN ドライバー (hfcmulti.ko.xz)

- NETJet 用 mISDN ドライバー (netjet.ko.xz)
- AVM FRITZ!CARD PCI ISDN カード用 mISDN ドライバー (avmfritz.ko.xz)

ストレージドライバー

- NVMe over Fabrics TCP ホスト (nvme-tcp.ko.xz)
- NVMe over Fabrics TCP ターゲット (nvmet-tcp.ko.xz)
- device-mapper writecache ターゲット (dm-writecache.ko.xz)

6.3. 更新されたドライバー

ネットワークドライバーの更新

- QLogic FastLinQ 4xxxx イーサネットドライバー (qed.ko.xz) がバージョン 8.37.0.20 に更新されました。
- QLogic FastLinQ 4xxxx コアモジュール (qed.ko.xz) がバージョン 8.37.0.20 に更新されました。
- Broadcom BCM573xx ネットワークドライバー (bnxt_en.ko.xz) がバージョン 1.10.0 に更新されました。
- QLogic BCM57710/57711/57711E/57712/57712_MF/57800/57800_MF/57810/57810_MF/57840/57840_ドライバー (bnx2x.ko.xz) がバージョン 1.713.36-0 に更新されました。
- Intel® Gigabit Ethernet ネットワークドライバー (igb.ko.xz) がバージョン 5.6.0-k に更新されました。
- Intel® 10 Gigabit Virtual Function ネットワークドライバー (ixgbev.ko.xz) がバージョン 4.1.0-k-rh8.1.0 に更新されました。
- Intel® 10 Gigabit PCI Express ネットワークドライバー (ixgbe.ko.xz) がバージョン 5.1.0-k-rh8.1.0 に更新されました。
- Intel® Ethernet Switch Host Interface ドライバー (fm10k.ko.xz) がバージョン 0.26.1-k に更新されました。
- Intel® Ethernet Connection E800 Series Linux ドライバー (ice.ko.xz) がバージョン 0.7.4-k に更新されました。
- Intel® Ethernet Connection XL710 ネットワークドライバー (i40e.ko.xz) がバージョン 2.8.20-k に更新されました。
- Netronome Flow Processor (NFP) ドライバー (nfp.ko.xz) がバージョン 4.18.0-147.el8.x86_64 に更新されました。
- Elastic Network Adapter (ENA) (ena.ko.xz) がバージョン 2.0.3K に更新されました。

グラフィックおよびその他ドライバーの更新

- VMware SVGA デバイス (vmwgfx.ko.xz) のスタンドアロン drm ドライバーがバージョン 2.15.0.0 に更新されました。

- HPE ウォッチドッグドライバー (hpwdt.ko.xz) がバージョン 2.0.2 に更新されました。

ストレージドライバーの更新

- HP Smart Array Controller バージョン 3.4.20-170-RH3 (hpsa.ko.xz) のドライバーは、バージョン 3.4.20-170-RH3 に更新されました。
- LSI MPT Fusion SAS 3.0 デバイスドライバー (mpt3sas.ko.xz) がバージョン 28.100.00.00 に更新されました。
- Emulex LightPulse ファイバーチャネル SCSI ドライバー 12.2.0.3 (lpfc.ko.xz) がバージョン 0:12.2.0.3 に更新されました。
- QLogic QEDF 25/40/50/100Gb FCoE ドライバー (qedf.ko.xz) がバージョン 8.37.25.20 に更新されました。
- Cisco FCoE HBA ドライバー (fnic.ko.xz) がバージョン 1.6.0.47 に更新されました。
- QLogic ファイバーチャネル HBA ドライバー (qla2xxx.ko.xz) がバージョン 10.01.00.15.08.1-k1 に更新されました。
- Microsemi Smart Family Controller バージョン 1.2.6-015 (smartpqi.ko.xz) のドライバーがバージョン 1.2.6-015 に更新されました。
- QLogic FastLinQ 4xxxx iSCSI モジュール (qedi.ko.xz) がバージョン 8.33.0.21 に更新されました。
- Broadcom MegaRAID SAS ドライバー (megaraid_sas.ko.xz) がバージョン 07.707.51.00-rc1 に更新されました。

6.4. バグ修正

本パートでは、ユーザーに大きな影響を及ぼしていた Red Hat Enterprise Linux 8.1 のバグで修正されたものを説明します。

6.4.1. インストーラーおよびイメージの作成

カーネル起動パラメーター **version** または **inst.version** を指定しても、インストールプログラムが停止しなくなります。

以前では、**version** または **inst.version** のブートパラメーターを指定してカーネルコマンドラインからインストールプログラムを起動すると、バージョン (例: **anaconda 30.25.6**) が表示され、インストールプログラムが停止していました。

今回の更新で、カーネルコマンドラインからインストールプログラムを起動した場合に、**version** パラメーターおよび **inst.version** パラメーターが無視されるため、インストールプログラムは停止しません。

(BZ#1637472)

ビデオドライバー **xorg-x11-drv-fbdev**、**xorg-x11-drv-vesa**、および **xorg-x11-drv-vmware** がデフォルトでインストールされるようになりました。

以前では、NVIDIA グラフィックスカードの特定のモデルが搭載されたワークステーションと、特定の AMD アクセラレート処理ユニットが搭載されたワークステーションでは、RHEL 8.0 Server のインストール後にグラフィカルログインウィンドウが表示されていませんでした。また、Hyper-V などのグラ

フィックスサポートに EFI を使用する仮想マシンも影響を受けていました。今回の更新で、**xorg-x11-drv-fbdev**、**xorg-x11-drv-vesa**、および **xorg-x11-drv-vmware** ビデオドライバーがデフォルトでインストールされ、RHEL 8.0 以降の Server のインストール後にグラフィカルログインウィンドウが表示されます。

(BZ#1687489)

エラーメッセージを表示せずにレスキューモードが失敗しなくなりました。

以前では、Linux パーティションのないシステムでレスキューモードを実行すると、インストールプログラムで例外が発生して失敗していました。この更新により、Linux パーティションがないシステムが検出されると、インストールプログラムが You don't have any Linux partitions(Linux パーティションがありません) というエラーメッセージを表示します。

(BZ#1628653)

インストールプログラムは、イメージインストールに **lvm_metadata_backup Blivet** フラグを設定するようになりました。

以前では、インストールプログラムはイメージインストールの **lvm_metadata_backup blivet** フラグの設定に失敗していました。これにより、LVM バックアップファイルは、イメージのインストール後に **/etc/lvm/** サブディレクトリー配置されていました。この更新により、インストールプログラムは **lvm_metadata_backup Blivet** フラグを設定します。結果として、イメージのインストール後に **/etc/lvm/** サブディレクトリーに LVM バックアップファイルは配置されません。

(BZ#1673901)

RHEL 8 インストールプログラムが RPM の文字列を処理するようになりました。

以前では、**python3-rpm** ライブラリーが文字列を返すと、インストールプログラムは例外で失敗していました。この更新により、インストールプログラムが RPM の文字列を処理できるようになりました。

(BZ#1689909)

inst.repo カーネル起動パラメーターが、**root** 以外のパスのあるハードドライブのリポジトリーに対して有効になりました。

以前では、**inst.repo=hd:<device>:<path>** カーネルブートパラメーターがハードドライブ上の (ISO イメージではなく) リポジトリーを参照することがあり、**root (/)** 以外のパスが使用されている場合、RHEL 8 のインストールプロセスが完了しませんでした。この更新により、インストールプログラムで、ハードドライブにあるリポジトリー用の **<path>** が伝播され、インストールが正常に行われるようになりました。

(BZ#1689194)

--changesok オプションを指定すると、インストールプログラムが **root** パスワードを変更できるようになりました。

これまで、キックスタートファイルから Red Hat Enterprise Linux 8 をインストールする際に **--changesok** オプションを指定しても、インストールプログラムで **root** パスワードを変更できませんでした。今回の更新で、キックスタートにより **--changesok** オプションが正常に渡されるようになったため、パスワードがキックスタートによって設定されていても、キックスタートファイルで **pwpolicy root --changesok** オプションを指定すれば、GUI を使用して **root** パスワードを変更できます。

(BZ#1584145)

lorax-composer API の使用時にイメージビルドに失敗しなくなりました。

以前では、サブスクライブした RHEL システムから **lorax-composer** API を使用すると、イメージ構築プロセスが常に失敗していました。ホストのサブスクリプション証明書が通っていないため、Anaconda はリポジトリにアクセスできませんでした。この問題を修正するには、**lorax-composer** パッケージ、**pykickstart** パッケージ、および **Anaconda** パッケージを更新します。これにより、サポートされている CDN 証明書を渡すことができます。

(BZ#1663950)

6.4.2. シェルおよびコマンドラインツール

デバッグモードの **systemd** が不要なログメッセージを生成しなくなりました。

デバッグモードで **systemd** システムおよびサービスマネージャーを使用すると、以前では **systemd** により、以下で始まる不要で害のないログメッセージが作成されていました。

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

今回の更新で、**systemd** が修正され、この不要なデバッグメッセージが生成されなくなりました。

(BZ#1658691)

6.4.3. セキュリティー

fapolicyd が RHEL の更新を阻止しなくなりました。

更新で実行中のアプリケーションのバイナリーが置き換えられると、カーネルにより、接尾辞 " (deleted) が追加されて、メモリー内のアプリケーションのバイナリーパスが変更されます。以前のバージョンでは、**fapolicyd** ファイルアクセスポリシーデーモンは、信頼できないアプリケーションなどのように処理し、他のファイルを開き、実行できませんでした。そのため、更新の適用後にシステムを起動できないことがありました。

[RHBA-2020:5241](#) アドバイザリーのリリースでは、**fapolicyd** がバイナリーパスの接尾辞を無視し、バイナリーが信頼データベースに一致するようにします。これにより、**fapolicyd** がルールを正しく適用し、更新プロセスを完了できるようになりました。

(BZ#1897092)

SELinux が原因で **Tomcat** でメールが送信できなかった問題がなくなりました。

この更新以前は、SELinux ポリシーで **tomcat_t** および **pki_tomcat_t** ドメインから SMTP ポートへの接続を許可していませんでした。その結果、SELinux は Tomcat サーバー上のアプリケーションがメールを送信することを拒否しました。今回の **selinux-policy** パッケージの更新により、このポリシーは Tomcat ドメインからのプロセスが SMTP ポートにアクセスすることを許可し、SELinux により Tomcat 上のアプリケーションによるメール送信が阻止されなくなります。

(BZ#1687798)

lockdev が **SELinux** で正しく実行されるようになりました。

以前では、**lockdev_t** の SELinux ポリシーが定義されていても、**lockdev** ツールは **lockdev_t** コンテキストに移行できませんでした。これにより、root ユーザーで使用されると、**lockdev** が **unconfined_t** ドメインで実行されていました。これによりシステムに脆弱性が生じていました。今回の更新で、**lockdev_t** への移行が定義され、**lockdev** を Enforcing モードで正しく使用できるようになりました。

(BZ#1673269)

これで、SELinux で **iotop** が正しく実行されるようになりました。

以前では、**iotop_t** の SELinux ポリシーが定義されていても、**iotop** ツールが **iotop_t** コンテキストに移行することができませんでした。これにより、**iotop** は root ユーザーで使用されると、**unconfined_t** ドメインで実行されていました。これによりシステムに脆弱性が生じていました。今回の更新で、**lockdev_t** への移行が定義され、**lockdev** を Enforcing モードで SELinux とともに正しく使用できるようになりました。

(BZ#1671241)

SELinux が、NFS '**crossmnt**' を適切に処理するようになりました。

プロセスがサーバー上のマウントポイントとしてすでに使用されているサブディレクトリーにアクセスすると、**crossmnt** オプションを指定した NFS プロトコルが内部マウントを自動的に作成します。以前は、これにより、NFS マウントされたディレクトリーにアクセスするプロセスにマウントパーミッションが指定されているかどうかを SELinux に確認させていました。これにより、AVC 拒否が発生していました。現在のバージョンでは、SELinux パーミッションチェックでは、この内部マウントはスキップされます。そのため、サーバー側にマウントされた NFS ディレクトリーにアクセスしても、マウントパーミッションは必要ありません。

(BZ#1647723)

SELinux ポリシーの再読み込みでは、正しくない **ENOMEM** エラーが発生しなくなりました。

以前では SELinux ポリシーをリロードすると、内部セキュリティコンテキストルックアップテーブルが応答しなくなっていました。したがって、ポリシーの再ロード時にカーネルが新しいセキュリティコンテキストに遭遇すると、正しくない Out of memory (ENOMEM) エラーで操作が失敗していました。今回の更新で、内部セキュリティ識別子 (SID) ルックアップテーブルが再設計され、フリーズしなくなりました。その結果、カーネルは、SELinux ポリシーの再読み込み時に誤解を招くエラーを返さなくなりました。

(BZ#1656787)

制限のないドメインが **smc_socket** を使用できるようになりました。

以前では、SELinux ポリシーには、**smc_socket** クラスの許可ルールがありませんでした。したがって、SELinux は、制限のないドメインについて **smc_socket** へのアクセスをブロックしていました。今回の更新で、この許可ルールが SELinux ポリシーに追加されました。これにより、制限のないドメインは **smc_socket** を使用することができます。

(BZ#1683642)

Kerberos クリーンアップの手順は、**krb5.conf** の **GSSAPIDelegateCredentials** およびデフォルトキャッシュと互換性を持つようになりました。

以前では、**default_ccache_name** オプションが **krb5.conf** ファイルに設定されていると、kerberos の認証情報は **GSSAPIDelegateCredentials** オプションおよび **GSSAPICleanupCredentials** オプションで削除されませんでした。このバグは、前述のユースケースで認証情報キャッシュをクリーンアップするためにソースコードを更新することで修正されています。設定後、ユーザーの設定時に認証情報のキャッシュは、終了時にクリーンアップされます。

(BZ#1683295)

OpenSSH が、ラベルが一致しない鍵に対して PKCS #11 の URI を正しく処理するようになりました。

以前では、オブジェクト部分 (鍵ラベル) で PKCS #11 の URI を指定すると、OpenSSH が PKCS #11 の関連オブジェクトを見つけられなくなっていました。今回の更新により、一致するオブジェクトが見つからない場合や、キーが ID によってのみ一致する場合は、ラベルが無視されます。これにより、

OpenSSH は、完全な PKCS #11 の URI を使用して参照されるスマートカードの鍵を使用できるようになりました。

(BZ#1671262)

VMware ホストシステムとの SSH 接続が適切に動作するようになりました。

以前のバージョンの **OpenSSH** スイートでは、SSH パッケージでデフォルトの IPQoS (IP Quality of Service) フラグが変更されていましたが、VMware 仮想化プラットフォームではこれが適切に処理されませんでした。したがって、VMware のシステムとの SSH 接続を確立することができませんでした。この問題は VMware Workstation 15 で修正され、VMware ホストシステムとの SSH 接続が正しく動作するようになりました。

(BZ#1651763)

OpenSSH では、デフォルトで **curve25519-sha256** がサポートされるようになりました。

以前では、SSH 鍵交換アルゴリズム **curve25519-sha256** は、デフォルトのポリシーレベルに準拠する場合でも、OpenSSH のクライアントとサーバーのシステム全体の暗号化ポリシー設定にはありませんでした。そのため、クライアントまたはサーバーが **curve25519-sha256** を使用し、ホストがこのアルゴリズムに対応していない場合は、接続に失敗する可能性があります。**crypto-policies** パッケージの今回の更新でバグが修正され、このシナリオで SSH 接続が失敗しなくなりました。

(BZ#1678661)

OSPP プロファイルおよび PCI-DSS プロファイルの **Ansible Playbook** が、障害の発生後に終了しなくなります。

以前では、OSPP (Security Content Automation Protocol) および Payment Card Industry Data Security Standard (PCI-DSS) プロファイルの Ansible 修正が、修正における正しくない順序や他のエラーにより失敗していました。今回の更新で、生成された Ansible 修復 Playbook の順序およびエラーが修正され、Ansible の修正が正常に機能するようになりました。

(BZ#1741455)

Audit transport=KRB5 が適切に動作するようになりました。

今回の更新以前では、Audit KRB5 トランスポートモードが適切に動作しませんでした。その結果、Kerberos ピア認証を使用した Audit リモートロギングが有効ではありませんでした。今回の更新で問題が修正され、前述のシナリオで Audit リモートロギングが適切に動作するようになりました。

(BZ#1730382)

6.4.4. ネットワーク

カーネルが、**bitmap:ipmac**、**hash:ipmac**、および **hash:mac** IP セットタイプでの宛先 MAC アドレスに対応するようになりました。

以前では、**bitmap:ipmac**、**hash:ipmac**、および **hash:mac** IP セットタイプのカーネル実装のみ、ソース MAC アドレスでの一致のみを許可し、宛先の MAC アドレスは指定可能ですが、セットエントリに対してマッチしていませんでした。これにより、管理者は、これらの IP セットタイプのいずれかで、宛先の MAC アドレスを使用する **iptables** ルールを作成できましたが、指定の仕様に一致するパケットは実際には分類されていませんでした。今回の更新で、カーネルは、宛先 MAC アドレスを比較し、指定の分類がパケットの宛先 MAC アドレスに対応している場合に一致を返すようになりました。これにより、宛先の MAC アドレスに対してパケットに一致するルールが正しく動作するようになりました。

(BZ#1649087)

gnome-control-center アプリケーションが、高度な IPsec 設定の編集に対応しました。

以前では、**gnome-control-center** アプリケーションは、IPsec VPN 接続の高度なオプションのみを表示していました。したがって、ユーザーはこれらの設定を変更できませんでした。今回の更新で、高度な設定のフィールドが編集できるようになり、ユーザーは変更を保存できるようになりました。

(BZ#1697329)

man ページの **iptables-extensions (8)** の **TRACE** ターゲットが更新されました。

以前では、man ページの **iptables-extensions(8)** の **TRACE** ターゲットの説明は、**compat** バリエーションのみを参照していました。しかし、Red Hat Enterprise Linux 8 は **nf_tables** バリエーションを使用します。そのため、man ページでは、**TRACE** イベントを表示する **xtables-monitor** コマンドラインユーティリティーが参照されませんでした。man ページが更新されたため、**xtables-monitor** が記載されているようになりました。

(BZ#1658734)

ipset サービスにおけるエラーロギングが改善されました。

以前では、**ipset** サービスにより、**systemd** ログに、有意の重大度とともにエラーが報告されていませんでした。無効な設定エントリーの重大度レベルは、**情報** 通知のみで、サービスは、使用できない設定のエラーを報告していませんでした。したがって、管理者が **ipset** サービスの設定で問題を特定してトラブルシューティングを行うことは困難でした。今回の更新で、**ipset** が、**systemd** ログで **警告** として設定問題を報告します。サービスが起動できない場合は、詳細を含む **エラー** の重大度とともにエントリーをログに記録するようになりました。このため、**ipset** サービスの設定での問題のトラブルシューティングが可能になりました。

(BZ#1683711)

ipset が、システムの起動時に無効な設定エントリーを無視するようになりました。

ipset サービスは、設定を別のファイルにセットとして保存します。以前では、サービスを起動すると、セットを手動で編集して挿入できる無効なエントリーをフィルタリングせずに、1回の操作ですべてのセットから設定を復元していました。したがって、単一の設定エントリーが無効だった場合でも、このサービスは、それ以上の関連しないセットを復元していませんでした。この問題が修正されました。これにより、**ipset** サービスが復元操作時に無効な設定エントリーを検出して削除し、無効な設定エントリーを無視します。

(BZ#1683713)

ipset list コマンドは、**hash** セットタイプに対して一貫性のあるメモリーを報告します。

エントリーを **hash** セットタイプに追加した場合は、**ipset** ユーティリティーが、さらなるメモリーブロックを割り当てて、新しいエントリーのメモリー内表示のサイズを調整する必要があります。以前は、**ipset** は、現在のメモリー内サイズに値を追加する代わりに、新しいブロックのサイズにのみ、セットごとに割り当てられた合計サイズを設定していました。これにより、**ip list** コマンドで、一貫性のないメモリーサイズが報告されていました。今回の更新で、**ipset** が、メモリー内のサイズを正しく計算するようになりました。これにより、**ipset list** コマンドがセットの正しいメモリー内サイズを表示し、出力が、**ハッシュ** セットタイプに対して実際に割り当てられたメモリーに一致します。

(BZ#1714111)

ICMPv6 Packet Too Big メッセージの受信時にカーネルが **PMTU** を正しく更新するようになりました。

リンクローカルアドレスなどの特定の状況では、複数のルートがソースアドレスに適合できます。以前は、カーネルは、Internet Control Message Protocol Version 6 (ICMPv6) パケットを受信するときに、

入力インターフェイスを確認しませんでした。そのため、ルート検索が、入力インターフェイスに一致しない宛先を返すことがありました。そのため、ICMPv6 **Packet Too Big** メッセージを受信すると、カーネルが別の入力インターフェイスの Path Maximum Transmission Unit (PMTU) を更新する可能性があります。今回の更新で、カーネルはルートの検索時に入力インターフェイスを確認します。その結果、カーネルはソースアドレスに基づいて正しい宛先を更新し、上述のシナリオで PMTU が想定どおりに機能するようになりました。

(BZ#1721961)

`/etc/hosts.allow` ファイルおよび `/etc/hosts.deny` ファイルに、削除された `tcp_wrappers` への古い参照が含まれなくなりました。

以前では、`/etc/hosts.allow` ファイルおよび `/etc/hosts.deny` ファイルには `tcp_wrappers` パッケージに関する古い情報が含まれていました。これは、削除された `tcp_wrappers` に必要なくなったため、RHEL 8 ではファイルが削除されました。

(BZ#1663556)

6.4.5. カーネル

`tpm2-abrmd-selinux` に、`selinux-policy-targeted` に適切な依存関係が含まれるようになりました。

`tpm2-abrmd-selinux` パッケージには、`selinux-policy-targeted` パッケージの代わりに `selinux-policy-base` パッケージの依存関係が含まれていました。したがって、システムに `selinux-policy-targeted` の代わりに `selinux-policy-minimum` をインストールする場合は、`tpm2-abrmd-selinux` パッケージのインストールが失敗していました。今回の更新でバグが修正され、このシナリオにおいて `tpm2-abrmd-selinux` を正しくインストールできます。

(BZ#1642000)

すべての `/sys/kernel/debug` ファイルにアクセスできるようになりました。

以前では、エラーに関係なく、関数の終わりまで、Operation not permitted (EPERM) エラーの戻り値が残っていました。これにより、特定の `/sys/kernel/debug` (`debugfs`) ファイルにアクセスしようとすると、unwarranted EPERM エラーで失敗していました。今回の更新で、EPERM 戻り値を以下のブロックに移動します。したがって、このシナリオで問題なく `debugfs` ファイルにアクセスできます。

(BZ#1686755)

NIC は、41000 および 45000 FastLinQ シリーズの `qede` ドライバーのバグによる影響を受けなくなりました。

以前では、ファームウェアのアップグレードおよびデバッグのデータ収集操作が、41000 および 45000 FastLinQ シリーズの `qede` ドライバーのバグにより失敗していました。これにより、NIC が使用不可能になっていました。ホストの再起動 (PCI リセット) により、NIC が再び動作するようになりました。

この問題は以下のシナリオで発生する可能性があります。

- `inbox` ドライバーを使用した NIC のファームウェアのアップグレード中
- `ethtool -d ethx` コマンドを実行するデバッグデータの収集時
- `ethtool -d ethx` を含む `sosreport` コマンドを実行する際

- I/O タイムアウト、Mail Box Command タイムアウト、Hardware Attention などの inbox ドライバーによる自動デバッグデータ収集の開始時。

この問題を修正するため、Red Hat は Bug Advisory (RHBA) でエラーをリリースしました。RHBA のリリース前に、対応の修正をリクエストするために、<https://access.redhat.com/support> でケースを作成することが推奨されます。

(BZ#1697310)

汎用の **EDAC GHES** ドライバーは、エラーを報告した **DIMM** を検出するようになりました

以前では、**EDAC GHES** ドライバーは、エラーを報告した DIMM を検出できませんでした。したがって、以下のエラーメッセージが表示されていました。

```
DIMM location: not present. DMI handle: 0x<ADDRESS>
```

このドライバーは、**DMI (SMBIOS)** テーブルをスキャンして、DMI (Desktop Management Interface) に一致する特定の DIMM による **0x <ADDRESS>** の処理を検出するように更新されました。その結果、**EDAC GHES** は、ハードウェアエラーを報告した特定の DIMM を正しく検出します。

(BZ#1721386)

podman が、**RHEL 8** で目的のコンテナをチェックポイントできるようになりました。

以前では、CRIU (Checkpoint and Restore In Userspace) パッケージのバージョンが古くなっていました。したがって、CRIU は、コンテナのチェックポイントおよび復元機能に対応しておらず、**podman** ユーティリティーが、チェックポイントコンテナに失敗していました。**podman container checkpoint** コマンドを実行すると、以下のエラーメッセージが表示されていました。

```
'checkpointing a container requires at least CRIU 31100'
```

今回の更新で、CRIU パッケージのバージョンをアップグレードすることでこの問題が修正されます。これにより、**podman** が、コンテナのチェックポイントおよび復元機能に対応するようになりました。

(BZ#1689746)

dracut.conf で **add_dracutmodules+=earlykdump** オプションを指定すると、**early-kdump** および標準の **kdump** が失敗しなくなりました。

以前では、**early-kdump** 用にインストールするカーネルバージョンと、**initramfs** 用に生成されたカーネルバージョンの間で不整合が発生していました。これにより、**early-kdump** が有効になっていると、システムの起動に失敗していました。また、**early-kdump** が標準の **kdump** **initramfs** イメージに含まれていることを検出すると、強制的に終了していました。また、**early-kdump** がデフォルトの **dracut** モジュールとして追加されると、**kdump** **initramfs** を再構築しようとする際に、標準の **kdump** サービスが失敗していました。これにより、**early-kdump** と標準の **kdump** の両方が失敗していました。今回の更新で、**early-kdump** がインストール時に一貫したカーネル名を使用し、実行中のカーネルとバージョンのみが異なります。また、標準の **kdump** サービスは、イメージ生成の失敗を防ぐために、**early-kdump** を強制的にドロップします。これにより、上記のシナリオで、**early-kdump** と標準の **kdump** が失敗しなくなりました。

(BZ#1662911)

SME を有効化した最初のカーネルが、**vmcore** のダンプに成功するようになりました。

以前では、アクティブな **SME** (Secure Memory Encryption) 機能のある最初のカーネルで暗号化されたメモリーが原因で、**kdump** メカニズムに障害が発生していました。したがって、最初のカーネルは、

メモリーの内容 (vmcore) をダンプできませんでした。今回の更新で、暗号化メモリーを再度マッピングし、関連するコードを変更するために、**ioremap_encrypted()** 関数が追加されました。その結果、暗号化されている最初のカーネルのメモリーが適切にアクセスされるようになり、上記のシナリオでクラッシュツールを使用して vmcore をダンプし、解析することができます。

(BZ#1564427)

SEV を有効化した最初のカーネルが、**vmcore** のダンプに成功するようになりました。

以前では、アクティブな SEV (Secure Encrypted Virtualization) 機能のある最初のカーネルで暗号化されたメモリーが原因で、**kdump** メカニズムに障害が発生していました。したがって、最初のカーネルは、メモリーの内容 (vmcore) をダンプできませんでした。今回の更新で、暗号化メモリーを再度マッピングし、関連するコードを変更するために、**ioremap_encrypted()** 関数が追加されました。その結果、最初のカーネルの暗号化されているメモリーが適切にアクセスされるようになり、上記のシナリオでクラッシュツールを使用して vmcore をダンプし、解析することができます。

(BZ#1646810)

カーネルが、**SWIOTLB** 用により多くの領域を予約するようになりました。

以前では、Secure Encrypted Virtualization (SEV) または Secure Memory Encryption (SME) 機能がカーネルで有効になっていると、Software Input Output Translation Lookaside Buffer (SWIOTLB) 技術も有効化されなくてはならず、大量のメモリーが使用されていました。その結果、キャプチャーカーネルが起動に失敗したり、メモリー不足のエラーが発生していました。今回の更新で、SEV/SME がアクティブの際に SWIOTLB の追加の crashkernel メモリーを予約することで、バグが修正されています。その結果、キャプチャーカーネルは SWIOTLB 用に予約されているメモリーが多くなり、上記のシナリオでバグが表示されなくなりました。

(BZ#1728519)

hwlatdetect の実行時に **c-state** 移行が無効にできるようになりました。

リアルタイムのパフォーマンスを実現するには、**hwlatdetect** ユーティリティーがテストの実行中に CPU での節電を無効にできる必要があります。今回の更新で、**hwlatdetect** が、テスト実行中に C-state 移行をオフにできるようになりました。また、**hwlatdetect** がハードウェアの遅延をより正確に検出できるようになりました。

(BZ#1707505)

6.4.6. ハードウェアの有効化

openmpi パッケージをインストールできるようになりました。

opensm パッケージにおけるリベースにより、**soname** メカニズムが変更されていました。その結果、未解決の依存関係が原因で、**openmpi** パッケージをインストールできませんでした。今回の更新でこの問題が修正されています。結果、**openmpi** パッケージが問題なくインストールできるようになりました。

(BZ#1717289)

6.4.7. ファイルシステムおよびストレージ

RHEL 8 インストールプログラムは、エントリー ID を使用してデフォルトのブートエントリーを設定するようになりました。

以前では、**RHEL 8** インストールプログラムは、エントリー ID ではなく、最初のブートエントリーのインデックスをデフォルトとして使用していました。これにより、新しいブートエントリーを追加する

と、最初にソートされ、最初のインデックスに設定されるかのようにデフォルトになっていました。この更新により、このインストールプログラムはエントリー ID を使用してデフォルトのブートエントリーを設定します。その結果、デフォルトの前にブートエントリーが追加され、ソートされた場合でも、デフォルトのエントリーは変更されません。

(BZ#1671047)

SME が **smartpqi** で有効になっていると、システムが正常に起動するようになりました。

以前では、Secure Memory Encryption (SME) 機能が有効で、root ディスクが **smartpqi** ドライバーを使用している場合に、特定の AMD マシンでのシステムの起動が失敗していました。

システムの起動に失敗すると、システムでは、ブートログに以下のようなメッセージが表示されます。

```
smartpqi 0000:23:00.0: failed to allocate PQI error buffer
```

一貫性のある Direct Memory Access (DMA) マスクが設定されていなかったため、Software Input Output Translation Lookaside Buffer (SWIOTLB) にフォールバックしていた **smartpqi** ドライバーによりこの問題が発生します。

今回の更新で、一貫した DMA マスクが正しく設定されるようになりました。その結果、SME がルートディスクの **smartpqi** ドライバーを使用するマシンで有効になっていると、システムが正常に起動されるようになりました。

(BZ#1712272)

FCoE LUN が **bnx2fc** カードで作成した後に消えなくなりました。

以前では、**bnx2fc** カードに FCoE LUN を作成すると、FCoE LUN が正しく割り当てられませんでした。これにより、RHEL 8.0 において **bnx2fc** カードに FCoE LUN を作成した後に、これが消えていました。この更新により、FCoE LUN が正しく割り当てられるようになりました。その結果、FCoE LUN が **bnx2fc** カードに作成された後に、これを検出できるようになりました。

(BZ#1685894)

VDO ボリュームが、異なるエンディアンプラットフォームへ移動した後の重複排除アドバイスを失わなくなりました。

以前では、別のエンディアンを使用するプラットフォームに VDO ボリュームを移動すると、Universal Deduplication Service (UDS) インデックスがすべて重複排除アドバイスを失っていました。そのため、VDO はボリュームを移動する前に保存したデータに対して、新たに書き込まれたデータを重複排除できないため、容量節約が低減していました。

今回の更新で、重複排除アドバイスを失わず、異なるエンディアンを使用するプラットフォーム間で VDO ボリュームを移動できるようになりました。

(BZ#1696492)

kdump サービスが大規模な **IBM POWER** システムで動作するようになりました。

以前では、RHEL8 **kdump** カーネルが起動していませんでした。したがって、大規模な **IBM POWER** システムでの **kdump initrd** ファイルは作成されませんでした。今回の更新で、**squashfs-tools-4.3-19.el8** コンポーネントが追加されました。今回の更新で、利用可能なすべての CPU ではなく、**squashfs-tools-4.3-19.el8** コンポーネントが利用可能なプールから資料できる CPU の数の制限 (128) が追加されました。これにより、リソース不足エラーが修正されました。これにより、大規模な **IBM POWER** システムで **kdump** サービスが機能するようになりました。

(BZ#1716278)

nfs.conf に詳細なデバッグオプションが追加されました。

以前では、**/etc/nfs.conf** ファイルおよび man ページの **nfs.conf (5)** には、以下のオプションが含まれていませんでした。

- `verbosity`
- `rpc-verbosity`

結果として、ユーザーは、これらのデバッグフラグの可用性について認識していませんでした。今回の更新で、これらのフラグは **/etc/nfs.conf** ファイルの **[gssd]** セクションに含まれるようになりました。また man ページの **nfs.conf (8)** にも記載されています。

(BZ#1668026)

6.4.8. 動的プログラミング言語、Web サーバー、およびデータベースサーバー

Socket::inet_aton() が複数のスレッドから安全に使用できるようになりました。

以前は、複数の Perl スレッドからドメイン名を解決するのに使用していた **Socket::inet_aton()** 関数が、安全でない **gethostbyname()** **glibc** 関数を呼び出していました。このため、時折間違った IPv4 アドレスが返されたり、または Perl インタープリターが予期せずに終了したりしていました。今回の更新で、**gethostbyname()** ではなく、スレッドセーフ **getaddrinfo()** **glibc** 関数を使用するように **Socket::inet_aton()** 実装が変更されました。これにより、Perl **Socket** モジュールの **inet_aton()** 関数が、複数のスレッドから安全に使用できます。

(BZ#1699793, BZ#1699958)

6.4.9. コンパイラーおよび開発ツール

gettext は、メモリー不足であっても未変換のテキストを返します。

以前では、テキストローカリゼーションの **gettext()** 関数は、メモリー不足の際にテキストの代わりに NULL 値を返していたため、アプリケーションにテキスト出力やラベルが欠けていました。このバグは修正され、メモリー不足の際に通常通りに **gettext()** が期待通りに未変換のテキストを返すようになりました。

(BZ#1663035)

locale コマンドでは、実行中にエラーに遭遇すると **LOCPATH** が設定されていることについて警告するようになりました。

以前では、**locale** コマンドは、無効な **LOCPATH** が原因でエラーが発生した時に **LOCPATH** 環境変数の診断を行いませんでした。**locale** コマンドは、実行時にエラーが発生する際に **LOCPATH** が設定されていることを警告するように設定されるようになりました。その結果、**locale** は **LOCPATH** と、発生した基本的なエラーを報告するようになりました。

(BZ#1701605)

gdb が、**aarch64 SVE** の **core** ファイルの **z** レジスターを読み取り、正しく表示できるようになりました。

以前では、**gdb** コンポーネントは、**aarch64** スケーラブルベクター拡張 (SVE) アーキテクチャーとの **core** ファイルからの **z** レジスターの読み込みに失敗していました。今回の更新で、**gdb** コンポーネントが、**core** ファイルから **z** レジスターを読み取るようになりました。その結果、**info register** コマンドには **z** レジスターの内容が正常に表示されます。

(BZ#1669953)

GCC がバージョン 8.3.1 にリベースされました。

GNU コンパイラコレクション (GCC) がアップストリームバージョン 8.3.1 に更新されました。このバージョンには、多くのさまざまなバグ修正が含まれています。

(BZ#1680182)

6.4.10. Identity Management

FreeRADIUS が IPv6 アドレスを指すホスト名を解決するようになりました。

以前の RHEL 8 バージョンの FreeRADIUS では、**ipaddr** ユーティリティーは IPv4 アドレスだけに対応していました。したがって、**radiusd** デーモンが IPv6 アドレスを解決するには、RHEL 7 から RHEL 8 ヘシステムのアップグレード後に、設定を手動で更新する必要がありました。この更新で、その基礎となるコードが修正され、FreeRADIUS の **ipaddr** も IPv6 アドレスを使用するようになりました。

(BZ#1685546)

Nuxwdog サービスが HSM 環境での PKI サーバーの起動に失敗しなくなりました。

以前では、バグにより、**keyutils** パッケージが **pki-core** パッケージの依存関係としてインストールされませんでした。さらに、**Nuxwdog** ウォッチドッグサービスでは、ハードウェアセキュリティーモジュール (HSM) を使用する環境で公開鍵基盤 (PKI) サーバーを起動できませんでした。これらの問題は修正されています。その結果、必要な **keyutils** パッケージが依存関係として自動的にインストールされ、**Nuxwdog** が、HSM を使用する環境で想定通りに PKI サーバーを起動するようになりました。

(BZ#1695302)

IdM サーバーが、**FIPS** モードで正しく機能するようになりました。

以前では、Tomcat サーバーの SSL コネクターが完全に実装されていませんでした。結果として、インストールした証明書サーバーがある Identity Management (IdM) サーバーが、FIPS モードが有効になっているマシンでは機能しなくなっていました。このバグは、**JSSTrustManager** および **JSSKeyManager** の追加で修正されています。その結果、上記のシナリオで IdM サーバーが適切に動作するようになります。

IdM サーバーが RHEL 8 の FIPS モードで動作しなくなるバグがいくつかあることに注意してください。今回の更新で、以下のいずれかが修正されています。

(BZ#1673296)

KCM 認証情報キャッシュが、単一の認証情報キャッシュ内で多数の認証情報を行うに適するようになりました。

以前では、Kerberos Credential Manager (KCM) に多くの認証情報が含まれていると、**kinit** のような Kerberos 操作が、データベースのエントリサイズの制限やエントリーの数により失敗していました。

今回の更新で、**sssd.conf** ファイルの **kcm** セクションに、以下の新しい設定オプションが追加されました。

- **max_ccaches** (整数)
- **max_uid_ccaches** (整数)
- **max_ccache_size** (整数)

これにより、KCM が、単一の ccache において多数の認証情報を処理できるようになります。

設定オプションの詳細は、[sssd-kcm man ページ](#) を参照してください。

(BZ#1448094)

Samba が **sss ID** マッピングプラグインを使用する際にアクセスを拒否しなくなりました。

以前では、この設定を使用してドメインメンバーで Samba を実行し、**sss** ID マッピングバックエンドを使用する設定を `/etc/samba/smb.conf` ファイルに追加してディレクトリーを共有すると、ID マッピングバックエンドの変更によりエラーが発生していました。そのため、Samba は、ユーザーまたはグループが存在し、SSSD が認識している場合でも、特定のケースではファイルへのアクセスを拒否していました。この問題が修正されました。その結果、Samba は **sss** プラグインを使用する際にアクセスを拒否しなくなりました。

(BZ#1657665)

SSSD のタイムアウトのデフォルト値が、互いに競合しなくなりました。

以前では、デフォルトのタイムアウト値に競合が発生していました。以下のオプションのデフォルト値が、フェイルオーバー機能を向上させるために変更されました。

- `dns_resolver_op_timeout` - 2s (以前は 6s) に設定
- `dns_resolver_timeout` - 4s (以前は 6s) に設定
- `ldap_opt_timeout` - 8s (以前は 6s) に設定

また、新しい `dns_resolver_server_timeout` オプションがデフォルト値 1000 ms で追加されました。これは、ある DNS サーバーから SSSD が別の DNS サーバーに切り替える際のタイムアウトの長さを指定します。

(BZ#1382750)

6.4.11. デスクトップ

`systemctl isolate multi-user.target` にコンソールプロンプトが表示されるようになりました。

GNOME デスクトップセッションで、GNOME ターミナルから `systemctl isolate multi-user.target` コマンドを実行すると、コンソールプロンプトではなくカーソルのみが表示されていました。今回の更新により `gdm` が修正され、上記の状況で期待通りにコンソールプロンプトが表示されるようになりました。

(BZ#1678627)

6.4.12. グラフィックインフラストラクチャー

'i915' ディスプレイドライバーが、3×4K までの表示設定に対応するようになりました。

以前では、**Xorg** セッションで 'i915' ディスプレイドライバーを使用する場合は、2×4K を超える表示設定を使用できませんでした。今回の更新で、i915 ドライバーが最大 3×4K までのディスプレイ設定に対応しました。

(BZ#1664969)

GPU ドライバーを初期化しても **Linux** ゲストがエラーを表示しなくなりました。

以前では、GPU ドライバーを初期化する際に、Linux ゲストが警告を返していました。これは、Intel

Graphics Virtualization Technology -g (GVT -g) がゲストの **DisplayPort** (DP) インターフェイスのみをシミュレートし、'EDP_psr_imr' および 'EDP_psr_IR' レジスターをデフォルトのメモリーマッピングした I/O (MMIO) 読み取り/書き込みレジスターのままにしているため発生します。この問題を解決するために、ハンドラーがこれらのレジスターに追加され、警告が返されなくなりました。

(BZ#1643980)

6.4.13. Web コンソール

session_recording シェルで RHEL Web コンソールにログインできます。

以前は、**tlog** シェル (セッションの録画を有効にする) ユーザーが RHEL Web コンソールにログインできませんでした。今回の更新でバグが修正されました。この更新をインストールした後に、**tlog-rec-session** シェルを `/etc/shells/` に追加する以前の回避策を元に戻す必要があります。

(BZ#1631905)

6.4.14. 仮想化

pcie-to-pci ブリッジコントローラーへの PCI デバイスのホットプラグが正しく動作するようになりました。

以前では、ゲスト仮想マシンの設定に `pcie-to-pci-bridge` コントローラーが含まれ、ゲストにエンドポイントデバイスが接続されていないと、そのコントローラーへの新しいデバイスのホットプラグが行われませんでした。今回の更新で、PCIe システムのレガシー PCI デバイスをホットプラグする方法が改善され、問題が発生しなくなりました。

(BZ#1619884)

入れ子仮想化を有効化しても、ライブ移行がブロックされなくなりました。

以前では、入れ子仮想化機能はライブ移行と互換性がありませんでした。したがって、RHEL 8 ホストで入れ子仮想化を有効にすると、ホストから仮想マシンを移行したり、仮想マシンの状態のスナップショットをディスクに保存したりできなくなっていました。今回の更新でこの問題が修正され、対象の仮想マシンが移行できるようになりました。

(BZ#1689216)

6.4.15. サポート関連

redhat-support-tool で **sosreport** アーカイブが作成されるようになりました。

以前では、**redhat-support-tool** ユーティリティーは **sosreport** アーカイブを作成できませんでした。その回避策は、**sosreport** コマンドを個別に実行し、**redhat-support-tool addattachment -c** コマンドを実行してアーカイブをアップロードすることでした。また、ユーザーはカスタマーポータルで Web UI を使用してカスタマーケースを作成し、**sosreport** アーカイブをアップロードできます。

さらに、**findkerneldebugs**、**btextract**、**analyze**、または **diagnose** などのコマンドオプションが期待通り機能していませんでしたが、今後のリリースで修正される予定です。

(BZ#1688274)

6.5. テクノロジープレビュー

本パートでは、Red Hat Enterprise Linux 8.1 で利用可能なテクノロジープレビュー機能を説明します。

テクノロジープレビューに対する Red Hat のサポート範囲の詳細は、[テクノロジープレビューのサポート範囲](#) を参照してください。

6.5.1. ネットワーク

TIPC の完全サポート

TIPC (Transparent Inter Process Communication) は、不安定にペアリングされたノードのクラスターで効率的な通信を行うために特別に設計されたプロトコルです。これは、カーネルモジュールとして機能し、**iproute2** パッケージに **tipc** ツールを提供します。これにより、クラスターのどこにいるかわからず、その他のアプリケーションと迅速かつ確実に通信できるアプリケーションを作成できます。この機能は、RHEL 8 で完全に対応するようになりました。

(BZ#1581898)

eBPF がテクノロジープレビューとして利用可能になりました。

Traffic Control (tc) カーネルサブシステムと **tc** ツールは、テクノロジープレビューとして、**eBPF** (extended Berkeley Packet Filtering) プログラムをパケットとして追加し、入力および出力の両方のキューイング規則に対するアクションを実行できます。これにより、カーネルネットワークデータパスでのプログラミング可能なパケット処理が可能になります。

(BZ#1699825)

nmstate がテクノロジープレビューとして利用可能になりました。

Nmstate は、ホストのネットワーク API です。テクノロジープレビューとして利用できる **nmstate** パッケージでは、ライブラリーおよび **nmstatectl** コマンドラインユーティリティを利用でき、ホストのネットワーク設定を宣言型で管理できます。ネットワークの状態は事前定義済みのスキーマで説明されています。現在の状態と、必要な状態への変更の報告は、両者ともこのスキーマに一致します。

詳細は、[/usr/share/doc/nmstate/README.md](#) ファイルおよび [/usr/share/doc/nmstate/examples](#) ディレクトリーにあるサンプルを参照してください。

(BZ#1674456)

AF_XDP がテクノロジープレビューとして利用可能に

AF_XDP (Address Family eXpress Data Path) ソケットは、高性能パケット処理用に設計されています。さらに処理するために、**XDP** を取り入れ、プログラムにより選択されたパケットの効率的なリダイレクトをユーザー空間アプリケーションに付与します。

(BZ#1633143)

XDP がテクノロジープレビューとして利用可能になりました。

eXpress Data Path (XDP) 機能はテクノロジープレビューとして利用でき、カーネルの入力データパスの初期段階にある高性能パケット処理に、**eBPF** (extended Berkeley Packet Filter) プログラムを追加する手段を提供します。これにより、効率的なプログラム可能なパケット分析、フィルタリング、および操作が可能になります。

(BZ#1503672)

KTLS がテクノロジープレビューとして利用可能になりました。

Red Hat Enterprise Linux 8 では、Kernel Transport Layer Security (KTLS) がテクノロジープレビューとして提供されます。KTLS は、AES-GCM 暗号化のカーネルで対称暗号化アルゴリズムまたは複号アルゴリズムを使用して TLS レコードを処理します。KTLS は、この機能に対応するネットワークインター

フェイスコントローラー (NIC) に TLS レコード暗号化をオフロードするインターフェイスも提供します。

(BZ#1570255)

systemd-resolved サービスがテクノロジープレビューとして利用できるようになりました。

systemd-resolved サービスは、ローカルアプリケーションに名前解決を提供します。このサービスは、DNS スタブリゾルバー、LLMNR (Link-Local Multicast Name Resolution)、およびマルチキャスト DNS リゾルバーとレスポンスのキャッシュと検証を実装します。

systemd パッケージが **systemd-resolved** を提供している場合でも、このサービスはサポートされていないテクノロジープレビューであることに注意してください。

(BZ#1906489)

6.5.2. カーネル

RHEL 8 では、**Control Group v2** がテクノロジープレビューとして利用可能に

Control Group v2 メカニズムは、統一された階層制御グループです。**Control Group v2** は、プロセスを階層的に編成し、制御された設定可能な方法で、階層に従ってシステムリソースを分配します。

以前のバージョンとは異なり、**Control Group v2** には階層が1つしかありません。このように階層が単純であるため、Linux カーネルでは次のことが可能になります。

- 所有者のロールに基づいたプロセスの分類
- 複数の階層でポリシーが競合する問題の解消

Control Group v2 では、多くのコントローラーに対応しています。

- CPU コントローラーにより、CPU サイクルの配分が調整されます。このコントローラーには以下が実装されています。
 - 通常のスケジューリングポリシーに対する重みおよび絶対帯域幅制限のモデル
 - 実時間スケジューリングポリシーに対する絶対帯域幅割り当てモデル
- メモリーコントローラーは、メモリー配分を調整します。現在、次の種類のメモリー使用量が追跡されます。
 - ユーザー側のメモリー (ページキャッシュと匿名メモリー)
 - dentry、inode などのカーネルデータ構造
 - TCP ソケットバッファー
- I/O コントローラーは、I/O リソースの配分を制限します。
- ライトバックコントローラーは、メモリーコントローラーおよび I/O コントローラーの両方と対話し、**Control Group v2** に固有のものです。

上記の情報は、[Control Group v2](#) に基づいています。ここでは、個別の **Control Group v2** コントローラーに関する情報を参照できます。

(BZ#1401552)

テクノロジープレビューとしての **kexec fast reboot**

kexec fast reboot 機能は、引き続きテクノロジープレビューとして利用できます。**kexec fast reboot** により、再起動が大幅に速くなります。この機能を使用するには、**kexec** カーネルを手動で読み込み、オペレーティングシステムを再起動します。

(BZ#1769727)

eBPF がテクノロジープレビューとして利用可能になりました。

eBPF (extended Berkeley Packet Filter)は、限られた一連の関数にアクセスできる制限付きサンドボックス環境において、カーネル領域でのコード実行を可能にするカーネル内の仮想マシンです。

仮想マシンには、さまざまな種類のマップの作成に対応した、新しいシステムコール **bpf()** が含まれ、特別なアセンブリーのコードでプログラムをロードすることも可能です。そして、このコードはカーネルにロードされ、実行時コンパイラーでネイティブマシンコードに変換されます。**bpf()** は、root ユーザーなど、**CAP_SYS_ADMIN** が付与されているユーザーのみが利用できます。詳細は、man ページの **bpf(2)** を参照してください。

ロードしたプログラムは、データを受信して処理するために、さまざまなポイント (ソケット、トレースポイント、パケット受信) に割り当てることができます。

eBPF 仮想マシンを使用する Red Hat には、多くのコンポーネントが同梱されています。各コンポーネントの開発フェーズはさまざまです。そのため、現在すべてのコンポーネントが完全にサポートされている訳ではありません。特定のコンポーネントがサポート対象と示されていない限り、すべてのコンポーネントはテクノロジープレビューとして提供されます。

現在、以下の重要な **eBPF** コンポーネントはテクノロジープレビューとして利用できます。

- **BPF Compiler Collection (BCC)** ツールパッケージは、**eBPF** 仮想マシンを使用する動的なカーネル追跡ユーティリティのコレクションです。**BCC** ツールパッケージは、64 ビット ARM アーキテクチャー、IBM Power Systems、リトルエンディアン、および IBM Z のアーキテクチャーでテクノロジープレビューとして利用できます。これは、AMD および Intel 64 ビットアーキテクチャーで完全にサポートされていることにご留意ください。
- **eBPF** 仮想マシンを使用する高レベルの追跡言語 **bpfftrace**。
- XDP (eXpress Data Path) 機能は、**eBPF** 仮想マシンを使用してカーネルでの高速パケット処理を有効にするネットワークングテクノロジーです。

(BZ#1559616)

テクノロジープレビューとして利用できる **Soft-RoCE**

Remote Direct Memory Access (RDMA) over Converged Ethernet (RoCE) は、RDMA over Ethernet を実装するネットワークプロトコルです。Soft-RoCE は、RoCE v1 および RoCE v2 の 2 つのプロトコルバージョンに対応する RoCE のソフトウェア実装です。Soft-RoCE ドライバーの **rdma_rxe** は、RHEL 8 ではサポートされていないテクノロジープレビューとして利用できます。

(BZ#1605216)

6.5.3. ハードウェアの有効化

igc ドライバーが、RHEL 8 ではテクノロジープレビューとして利用できるようになりました。

igc Intel 2.5G Ethernet Linux 有線 LAN ドライバーは、テクノロジープレビューとして、RHEL 8 の全アーキテクチャーで利用できるようになりました。**ethtool** ユーティリティは **igc** 有線 LAN もサポートします。

(BZ#1495358)

6.5.4. ファイルシステムおよびストレージ

NVMe/TCP がテクノロジープレビューとして利用可能になりました。

TCP/IP ネットワーク (NVMe/TCP) および対応する **nvme-tcp.ko** および **nvmet-tcp.ko** カーネルモジュールへのアクセスおよび共有がテクノロジープレビューとして追加されました。

ストレージクライアントまたはターゲットのいずれかとしての NVMe/TCP の使用は、**nvme-cli** パッケージおよび **nvmetcli** パッケージに含まれるツールで管理できます。

NVMe/TCP では、RDMA (Remote Direct Memory Access) およびファイバーチャネル (NVMe/FC) を含む、既存の NVMe over fabrics (NVMe-of) トランスポートとともにストレージトランスポートオプションを利用できます。

(BZ#1696451)

ファイルシステム **DAX** が、テクノロジープレビューとして **ext4** および **XFS** で利用可能になりました。

Red Hat Enterprise Linux 8.1 では、ファイルシステムの DAX がテクノロジープレビューとして利用できます。DAX は、永続メモリーをそのアドレス空間に直接マッピングする手段をアプリケーションに提供します。DAX を使用するには、システムで利用可能な永続メモリーの形式が必要になります。通常は、NVDIMM (Non-Volatile Dual In-line Memory Module) の形式で、DAX に対応するファイルシステムを NVDIMM に作成する必要があります。また、ファイルシステムは **dax** マウントオプションでマウントする必要があります。これにより、**dax** をマウントしたファイルシステムのファイルの **mmap** が、アプリケーションのアドレス空間にストレージを直接マッピングされます。

(BZ#1627455)

OverlayFS

OverlayFS は、ユニオンファイルシステムのタイプです。これにより、あるファイルシステムを別のファイルシステムに重ねることができます。変更は上位のファイルシステムに記録され、下位のファイルシステムは変更しません。これにより、ベースイメージが読み取り専用メディアにあるコンテナや DVD-ROM などのファイルシステムイメージを、複数のユーザーが共有できるようになります。

OverlayFS は、ほとんどの状況で引き続きテクノロジープレビューになります。したがって、カーネルは、この技術がアクティブになると警告を記録します。

以下の制限下で、対応しているコンテナエンジン (**podman**、**cri-o**、または **buildah**) とともに使用すると、OverlayFS に完全対応となります。

- OverlayFS は、コンテナエンジングラフィックドライバーとしての使用、または圧縮された **kdump** **initramfs** などのその他の特殊なユースケースとしての使用のみサポートされています。その使用は主にコンテナ COW コンテンツでサポートされており、永続ストレージではサポートされていません。非 OverlayFS ボリュームに永続ストレージを配置する必要があります。デフォルトのコンテナエンジン設定のみを使用できます。つまり、あるレベルのオーバーレイ、1つの下位ディレクトリー、および下位と上位の両方のレベルが同じファイルシステムにあります。
- 下層ファイルシステムとして使用に対応しているのは現在 XFS のみです。

また、OverlayFS の使用には、以下のルールと制限が適用されます。

- OverlayFS カーネル ABI とユーザー空間の動作については安定しているとみなされていないため、今後の更新で変更が加えられる可能性があります。
- OverlayFS は、POSIX 標準の制限セットを提供します。OverlayFS を使用してアプリケーションをデプロイする前に、アプリケーションを十分にテストしてください。以下のケースは、POSIX に準拠していません。
 - **O_RDONLY** で開いているファイルが少ない場合は、ファイルの読み取り時に **st_atime** の更新を受け取りません。
 - **O_RDONLY** で開いてから、**MAP_SHARED** でマッピングした下位ファイルは、後続の変更と一貫性がありません。
 - 完全に準拠した **st_ino** 値または **d_ino** 値は、RHEL 8 ではデフォルトで有効になっていませんが、モジュールオプションまたはマウントオプションを使用して、この値の完全な POSIX コンプライアンスを有効にできます。一貫した inode 番号を付けるには、**xino=on** マウントオプションを使用します。

redirect_dir=on オプションおよび **index=on** オプションを使用して、POSIX コンプライアンスを向上させることもできます。この 2 つのオプションにより、上位レイヤーの形式は、このオプションなしでオーバーレイと互換性がありません。つまり、**redirect_dir=on** または **index=on** でオーバーレイを作成し、オーバーレイをアンマウントしてから、このオプションなしでオーバーレイをマウントすると、予期しない結果またはエラーが発生することがあります。

- 既存の XFS ファイルシステムがオーバーレイとして使用できるかどうかを確認するには、次のコマンドを実行して、**ftype=1** オプションが有効になっているかどうかを確認します。

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

- SELinux セキュリティラベルは、OverlayFS で対応するすべてのコンテナエンジンでフォルトで有効になっています。
- このリリースの既知の問題は、OverlayFS に関連しています。詳細は [Linux カーネルドキュメントの Non-standard behavior](#) を参照してください。

OverlayFS の詳細は、[Linux カーネルのドキュメント](#) を参照してください。

(BZ#1690207)

Straits がテクノロジープレビューとして利用可能になりました。

Straits は、新しいローカルストレージマネージャーです。ユーザーへの追加機能を備えたストレージプールに、管理されるファイルシステムを提供します。

Straits を使用すると、次のようなストレージタスクをより簡単に実行できます。

- スナップショットおよびシンプロビジョニングを管理する
- 必要に応じてファイルシステムのサイズを自動的に大きくする
- ファイルシステムを維持する

Straits ストレージを管理するには、バックグラウンドサービス **stratisd** と通信する **stratis** ユーティリティを使用します。

Straits はテクノロジープレビューとして提供されます。

詳細については、Stratis のドキュメント ([Stratis ファイルシステムの設定](#)) を参照してください。

(JIRA:RHELPLAN-1212)

IdM ホストにログインした IdM および AD ユーザーが利用可能な Samba サーバーを、テクノロジープレビューとして、IdM ドメインメンバーに設定できるようになりました。

今回の更新で、Identity Management (IdM) ドメインメンバーに Samba サーバーを設定できるようになりました。同じ名前パッケージに含まれる新しい **ipa-client-samba** ユーティリティーは、Samba 固有の Kerberos サービスプリンシパルを IdM に追加し、IdM クライアントを準備します。たとえば、ユーティリティーは、**sss** ID マッピングバックエンドの ID マッピング設定で `/etc/samba/smb.conf` を作成します。その結果、管理者が IdM ドメインメンバーに Samba を設定できるようになりました。

IdM 信頼コントローラーが Global Catalog Service をサポートしないため、AD が登録した Windows ホストは Windows で IdM ユーザーおよびグループを見つけることができません。さらに、IdM 信頼コントローラーは、Distributed Computing Environment / Remote Procedure Calls (DCE/RPC) プロトコルを使用する IdM グループの解決をサポートしません。これにより、AD ユーザーは、IdM クライアントから Samba の共有およびプリンターにしかアクセスできません。

詳細は、[IdM ドメインメンバーでの Samba の設定](#) を参照してください。

(JIRA:RHELPLAN-13195)

6.5.5. 高可用性およびクラスター

Pacemaker の **podman** バンドルがテクノロジープレビューとして利用可能になりました。

Pacemaker コンテナバンドルは、テクノロジープレビューとして利用できるコンテナバンドル機能を使用して、**podman** コンテナプラットフォームで動作するようになりました。この機能はテクノロジープレビューとして利用できますが、例外が1つあります。Red Hat は、Red Hat Openstack 用の Pacemaker バンドルの使用に完全対応します。

(BZ#1619620)

テクノロジープレビューとして利用可能な **corosync-qdevice** のヒューリスティック

ヒューリスティックは、起動、クラスターメンバーシップの変更、**corosync-qnetd** への正常な接続でローカルに実行され、任意で定期的に実行される一連のコマンドです。すべてのコマンドが時間どおりに正常に終了すると (返されるエラーコードがゼロである場合)、ヒューリスティックは渡されますが、それ以外の場合は失敗します。ヒューリスティックの結果は **corosync-qnetd** に送信され、クォーラムとなるべきパーティションを判断するための計算に使用されます。

(BZ#1784200)

新しい **fence-agents-heuristics-ping** フェンスエージェント

Pacemaker は、テクノロジープレビューとして **fence_heuristics_ping** エージェントに対応するようになりました。このエージェントの目的は、実際にはフェンシングを行わず、フェンシングレベルの動作を新しい方法で活用する実験的なフェンスエージェントのクラスを開くことです。

ヒューリスティックエージェントが、実際のフェンシングを行うフェンスエージェントと同じフェンシングレベルで設定されていて、そのエージェントよりも順番が前に設定されているとします。その場合、フェンシングを行うエージェントで **off** 操作を行う前に、ヒューリスティックエージェントで、この操作を行います。このヒューリスティックエージェントが **off** アクションに対して失敗する場合、このフェンシングレベルが成功しないのはすでに明らかです。そのため、Pacemaker フェンシングは、フェンシ

ングを行うエージェントで **off** 操作を行うステップをスキップします。ヒューリスティックエージェントはこの動作を利用して、特定の条件下で、実際のフェンシングを行うエージェントがフェンシングできないようにできます。

サービスを適切に引き継ぐことができないことを事前に把握できる場合は、ノードがピアをフェンシングする意味がないのであれば、ユーザーは特に 2 ノードクラスターでこのエージェントを使用できます。たとえば、ネットワークアップリンクに到達してサービスがクライアントに到達できない場合は、ノードがサービスを引き継ぐ意味はありません。これは、ルーターへの ping が検出できる状況が考えられます。

(BZ#1775847)

6.5.6. ID 管理

Identity Management JSON-RPC API がテクノロジープレビューとして利用可能になりました。

Identity Management (IdM) では API が利用できます。API を表示するために、IdM は、テクノロジープレビューとして API ブラウザーも提供します。

Red Hat Enterprise Linux 7.3 では、複数のバージョンの API コマンドを有効にするために、IdM API が拡張されました。以前は、機能拡張により、互換性のない方法でコマンドの動作が変更することがありました。IdM API を変更しても、既存のツールおよびスクリプトを引き続き使用できるようになりました。これにより、以下が可能になります。

- 管理者は、管理しているクライアント以外のサーバーで、IdM の以前のバージョンもしくは最近のバージョンを使用できます。
- サーバーで IdM のバージョンを変更しても、開発者は特定バージョンの IdM コールを使用できます。

すべてのケースでサーバーとの通信が可能になります。たとえば、ある機能向けの新オプションが新しいバージョンに追加されていて、通信の一方の側でこれを使用していたとしても、特に問題はありませ

ん。

API の使用方法は [Identity Management API を使用して IdM サーバーに接続する \(テクノロジープレビュー\)](#) を参照してください。

(BZ#1664719)

DNSSEC が IdM でテクノロジープレビューとして利用可能になりました。

統合 DNS のある Identity Management (IdM) サーバーは、DNS プロトコルのセキュリティーを強化する DNS に対する拡張セットである DNS Security Extensions (DNSSEC) に対応するようになりました。IdM サーバーでホストされる DNS ゾーンは、DNSSEC を使用して自動的に署名できます。暗号鍵は、自動的に生成およびローテートされます。

DNSSEC で DNS ゾーン的安全性を強化する場合は、以下のドキュメントを参照することが推奨されます。

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

統合 DNS のある IdM サーバーは、DNSSEC を使用して、他の DNS サーバーから取得した DNS 回答を検証することに注意してください。これが、推奨される命名方法に従って設定されていない DNS ゾーンの可用性に影響を与える可能性があります。

(BZ#1664718)

6.5.7. グラフィックインフラストラクチャー

64 ビット ARM アーキテクチャーで VNC リモートコンソールがテクノロジープレビューとして利用可能に

64 ビットの ARM アーキテクチャーでは、Virtual Network Computing (VNC) リモートコンソールがテクノロジープレビューとして利用できます。グラフィックススタックの残りの部分は、現在、64 ビット ARM アーキテクチャーでは検証されていません。

(BZ#1698565)

6.5.8. Red Hat Enterprise Linux システムロール

RHEL システムロールの **postfix** ロールが、テクノロジープレビューとして利用可能になりました。

Red Hat Enterprise Linux システムロールは、Red Hat Enterprise Linux サブシステムの設定インターフェイスを提供します。これにより、Ansible ロールを介したシステム設定が簡単になります。このインターフェイスにより、Red Hat Enterprise Linux の複数のバージョンにわたるシステム設定の管理と、新しいメジャーリリースの導入が可能になります。

rhel-system-roles パッケージは、AppStream リポジトリを介して配布されます。

postfix ロールは、テクノロジープレビューとして利用可能です。

以下のロールが完全にサポートされています。

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

詳細は、ナレッジベースの [RHEL システムロール](#) に関する記事を参照してください。

(BZ#1812552)

rhel-system-roles-sap がテクノロジープレビューとして利用可能になりました。

rhel-system-roles-sap パッケージは、SAP 向けの Red Hat Enterprise Linux (RHEL) システムロールを提供します。これは、RHEL システムの設定を自動化して SAP ワークロードロードを実行するために使用できます。これらのロールは、関連する SAP ノート記載のベストプラクティスに基づいて最適な設定を自動的に適用することで、SAP ワークロードを実行するようにシステムを設定する時間を大幅に短縮できます。アクセスは、RHEL for SAP Solutions 製品に限定されます。サブスクリプションに関するサポートが必要な場合は、Red Hat カスタマーサポートまでご連絡ください。

rhel-system-roles-sap パッケージの以下の新しいロール がテクノロジープレビューとして利用できません。

- **sap-preconfigure**
- **sap-netweaver-preconfigure**
- **sap-hana-preconfigure**

詳細は、[Red Hat Enterprise Linux System Roles for SAP](#) を参照してください。

注記: RHEL 8.1 for SAP Solutions は、Intel 64 アーキテクチャーおよび IBM POWER9 で SAP HANA とともに使用できるように検証される予定です。その他の SAP アプリケーションやデータベース製品 (SAP、SAP ASE など) は、RHEL 8.1 の機能を使用できます。検証されたリリースと SAP サポートの最新情報は、SAP Notes 2369910 および 2235581 を参照してください。

(BZ#1660832)

rhel-system-roles-sap がバージョン 1.1.1 にリベース

[RHBA-2019:4258](#) アドバイザリーにより、**rhel-system-roles-sap** パッケージが更新され、複数のバグ修正が追加されました。以下に例を示します。

- SAP システムロールは、英語以外のロケールのホストで動作します。
- **kernel.pid_max** は **sysctl** モジュールで設定されます。
- **nproc** は、HANA に対して無制限に設定されます (SAP ノート 2772999 のステップ 9 を参照)。
- ハードプロセス制限は、ソフトプロセス制限よりも前に設定されます。
- プロセス制限を設定するコードが、**sap-preconfigure** ロールと同じように動作するようになりました。
- **handlers/main.yml** は、uefi 以外のシステムでのみ機能し、uefi システムではメッセージなしで無視されます。
- **rhel-system-roles** の未使用の依存関係を削除
- **sap_hana_preconfigure_packages** から **libssh2** を削除
- 特定の CPU 設定がサポートされていない場合の障害を防ぐために、追加チェックを追加
- true および false をすべて小文字に変換
- 最小限のパッケージ処理を更新
- ホスト名およびドメイン名を正しく設定
- 多くのマイナーな修正

rhel-system-roles-sap パッケージはテクノロジープレビューとして利用できます。

(BZ#1766622)

6.5.9. 仮想化

Hyper-V の RHEL ゲストで、Intel ネットワークアダプターが SR-IOV をサポートするようになりました。

テクノロジープレビューとして、Hyper-V ハイパーバイザーで実行している Red Hat Enterprise Linux のゲストオペレーティングシステムは、**ixgbevf** および **ixgbevf** ドライバーがサポートする Intel ネットワークアダプターに、シングルルート I/O 仮想化 (SR-IOV) 機能を使用することができるようになりました。この機能は、以下の条件が満たされると有効になります。

- ネットワークインターフェイスコントローラー (NIC) に対して SR-IOV サポートが有効になっている
- 仮想 NIC の SR-IOV サポートが有効になっている
- 仮想スイッチの SR-IOV サポートが有効になっている
- NIC の VF (Virtual Function) は、仮想マシンに接続されている

この機能は現在、Microsoft Windows Server 2019 および 2016 でサポートされています。

(BZ#1348508)

RHEL 8 Hyper-V 仮想マシンで KVM 仮想化が利用可能になりました。

ネストされた KVM 仮想化は、テクノロジープレビューとして、Microsoft Hyper-V ハイパーバイザーで使えるようになりました。これにより、Hyper-V ホストで実行している RHEL 8 ゲストシステムで仮想マシンを作成できます。

この機能は、現在 Intel システムでのみ有効です。また、ネストされた仮想化は、Hyper-V でデフォルトで有効になっていない場合があります。これを有効にするには、以下の Microsoft ドキュメントを参照してください。

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

(BZ#1519039)

KVM 仮想マシンの AMD SEV。

テクノロジープレビューとして、RHEL 8 に、KVM ハイパーバイザーを使用する AMD EPYC ホストマシン用のセキュア暗号化仮想化 (SEV) 機能が同梱されます。仮想マシンで有効になっている場合は、ホストが仮想マシンのデータにアクセスできないように、SEV が仮想マシンメモリーを暗号化します。ホストがマルウェアに感染した場合は、これにより仮想マシンのセキュリティーが向上します。

1 台のホストでこの機能を同時に使用できる仮想マシンの数は、ホストのハードウェアによって決まります。現在の AMD EPYC プロセッサは、SEV を使用して 15 台以下の稼働中の仮想マシンに対応します。

また、SEV が起動できるように設定された仮想マシンでは、ハードメモリー制限のある仮想マシンも設定する必要があります。これを行うには、仮想マシンの XML 設定に以下を追加します。

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

N に推奨される値は、ゲストの RAM + 256 MiB 以上になります。たとえば、ゲストに 2 GiB の RAM が割り当てられている場合、N は 2359296 以上になります。

(BZ#1501618, BZ#1501607, JIRA:RHELPLAN-7677)

Intel vGPU

テクノロジープレビューとして、物理 Intel GPU デバイスを、**仲介デバイス** と呼ばれる複数の仮想デバイスに分割できるようになりました。この仲介デバイスは、仮想 GPU として複数の仮想マシンに割り当てることができます。これにより、この仮想マシンが、1つの物理 Intel GPU のパフォーマンスを共有します。

選択した Intel GPU のみが vGPU 機能と互換性があることに注意してください。また、物理 GPU を仮想マシンに割り当てると、ホストが GPU を使用できなくなるため、ホストのグラフィカルディスプレイ出力が機能しない可能性があります。

(BZ#1528684)

IBM POWER 9 でネストされた仮想化システムが利用可能に

テクノロジープレビューとして、IBM POWER 9 システムで稼働する RHEL 8 ホストマシンでネストされた仮想化機能を使用することが可能になりました。ネストされた仮想化により、KVM 仮想マシンをハイパーバイザーとして機能させることができます。これにより、仮想マシン内で仮想マシンを実行できます。

ネストされた仮想化は、AMD64 および Intel 64 のシステムで引き続きテクノロジープレビューとして利用できます。

また、ネストされた仮想環境では、IBM POWER 9、ホスト、ゲスト、ネストされたゲストのすべてで、以下のいずれかのオペレーティングシステムが実行している必要があります。

- RHEL 8
- RHEL 7 for POWER 9

(BZ#1505999, BZ#1518937)

入れ子仮想マシンの作成

入れ子仮想化はテクノロジープレビューとして、RHEL 8 の KVM 仮想マシンで利用できます。この機能を使用すると、物理ホストで実行される仮想マシンはハイパーバイザーとして動作し、独自の仮想マシンをホストできます。

入れ子仮想化は、AMD64 および Intel 64 アーキテクチャーでのみ利用でき、入れ子ホストは RHEL 7 または RHEL 8 仮想マシンである必要があります。

(JIRA:RHELPLAN-14047)

6.5.10. コンテナ

podman-machine コマンドはサポート対象外です。

仮想マシンを管理するための **podman-machine** コマンドは、テクノロジープレビューとしてのみ利用可能です。代わりに、コマンドラインから直接 Podman を実行してください。

(JIRA:RHELDPCS-16861)

6.6. 非推奨になった機能

ここでは、Red Hat Enterprise Linux 8.1 で **非推奨** となった機能の概要を説明します。

非推奨の機能は、本製品の今後のメジャーリリースではサポートされない可能性が高く、新たに実装することは推奨されません。特定のメジャーリリースにおける非推奨機能の最新情報は、そのメジャーリリースの最新版のリリースノートを参照してください。

非推奨の機能のサポートステータスは、Red Hat Enterprise Linux 8 では変更されていません。サポート期間の詳細は、[Red Hat Enterprise Linux ライフサイクル](#) および [Red Hat Enterprise Linux Application Streams ライフサイクル](#) を参照してください。

現行および今後のメジャーリリースでは、非推奨のハードウェアコンポーネントの新規実装は推奨されません。ハードウェアドライバーの更新は、セキュリティと重大な修正のみに行われます。Red Hat では、このようなハードウェアの早期交換を推奨します。

パッケージが非推奨となり、使用の継続が推奨されない場合があります。製品からパッケージが削除されることもあります。その場合には、製品のドキュメントで、非推奨となったパッケージと同様、同一、またはより高度な機能を提供する最近のパッケージが指定され、詳しい推奨事項が記載されます。

RHEL 7 で使用され、RHEL 8 で **削除された** 機能の詳細は [RHEL 8 の導入における検討事項](#) を参照してください。

6.6.1. インストーラーおよびイメージの作成

複数のキックスタートコマンドおよびオプションが非推奨になりました。

RHEL 8 キックスタートファイルで以下のコマンドとオプションを使用すると、ログに警告が表示されます。

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

特定のオプションだけがリスト表示されている場合は、基本コマンドおよびその他のオプションは引き続き利用でき、非推奨ではありません。

キックスタートの詳細および変更点は、[RHEL 8 の導入における検討事項のキックスタートの変更](#) を参照してください。

(BZ#1642765)

キックスタートコマンド **ignoredisk** の **--interactive** オプションが非推奨になりました。

Red Hat Enterprise Linux の将来のリリースで **--interactive** オプションを使用すると、致命的なインストールエラーが発生します。このオプションを削除するには、キックスタートファイルを変更することが推奨されます。

(BZ#1637872)

6.6.2. ソフトウェア管理

rpmbuild --sign コマンドが非推奨になりました。

今回の更新で、**rpmbuild --sign** コマンドが非推奨となりました。Red Hat Enterprise Linux の今後のリリースでこのコマンドを実行すると、エラーが発生します。代わりに **rpmsign** コマンドを使用することが推奨されます。

(BZ#1688849)

6.6.3. セキュリティー

TLS 1.0 および TLS 1.1 が非推奨になりました。

TLS 1.0 プロトコルおよび TLS 1.1 プロトコルは、システム全体の暗号化ポリシーレベル **DEFAULT** で無効になります。たとえば、Firefox Web ブラウザーのビデオ会議アプリケーションで、非推奨のプロトコルを使用する必要がある場合は、システム全体の暗号化ポリシーを **LEGACY** レベルに変更してください。

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

詳細は、Red Hat カスタマーポータルナレッジベースの記事 [Strong crypto defaults in RHEL 8 and deprecation of weak crypto algorithms](#) および man ページの **update-crypto-policies(8)** を参照してください。

(BZ#1660839)

RHEL 8 で DSA が非推奨になりました。

デジタル署名アルゴリズム (DSA) は、Red Hat Enterprise Linux 8 では非推奨であると考えられています。DSA キーに依存する認証メカニズムはデフォルト設定では機能しません。**OpenSSH** クライアントは、**LEGACY** のシステム全体の暗号化ポリシーレベルでも DSA ホストキーを許可しません。

(BZ#1646541)

NSS で **SSL2 Client Hello** が非推奨に

TLS (Transport Layer Security) プロトコルバージョン 1.2 以前は、**SSL** (Secure Sockets Layer) プロトコルバージョン 2 と後方互換性がある形式の **Client Hello** メッセージを使用してネゴシエーションを開始できます。**NSS** (Network Security Services) ライブラリーでのこの機能への対応は非推奨となっており、デフォルトで無効になっています。

この機能への対応が必要なアプリケーションを有効にするには、新しい API の **SSL_ENABLE_V2_COMPATIBLE_HELLO** を使用する必要があります。この機能への対応は、Red Hat Enterprise Linux 8 の将来のリリースから完全に削除される可能性があります。

(BZ#1645153)

TPM 1.2 が非推奨になりました。

Trusted Platform Module (TPM) のセキュアな暗号化プロセッサの標準バージョンが 2016 年にバージョン 2.0 に更新されました。TPM 2.0 は TPM 1.2 に対する多くの改良を提供しますが、以前のバージョンと後方互換性はありません。TPM 1.2 は RHEL 8 で非推奨となり、次のメジャーリリースで削除される可能性があります。

(BZ#1657927)

6.6.4. ネットワーク

RHEL 8 でネットワークスクリプトが非推奨に

Red Hat Enterprise Linux 8 では、ネットワークスクリプトが非推奨になっており、デフォルトでは提供されなくなりました。基本的なインストールでは、**nmcli** ツール経由で、**NetworkManager** サービスを呼び出す **ifup** スクリプトおよび **ifdown** スクリプトの新しいバージョンが提供されます。Red Hat Enterprise Linux 8 で **ifup** スクリプトおよび **ifdown** スクリプトを実行する場合は、**NetworkManager** を実行する必要があります。

/sbin/ifup-local、**ifdown-pre-local**、および **ifdown-local** の各スクリプトでは、カスタムコマンドが実行されません。

このスクリプトが必要な場合は、次のコマンドを使用すれば、システムに非推奨のネットワークスクリプトをインストールできます。

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

ifup スクリプトと **ifdown** スクリプトが、インストールされている従来のネットワークスクリプトにリンクされます。

従来のネットワークスクリプトを呼び出すと、そのスクリプトが非推奨であることを示す警告が表示されます。

(BZ#1647725)

6.6.5. カーネル

ディスクレスブートを非推奨化しました。

ディスクレスブートにより、複数のシステムがネットワーク経由で root ファイルシステムを共有できます。メリットはありますが、リアルタイムのワークロードでネットワークレイテンシーが発生する可能性が高くなります。RHEL for Real Time 8 の将来のマイナーな更新では、ディスクレスの起動はサポートされなくなります。

(BZ#1748980)

rdma_rxe Soft-RoCE ドライバーが非推奨に

Remote Software Direct Memory Access over Converged Ethernet (Soft-RoCE) は RXE としても知られており、RDMA (Remote Direct Memory Access) をエミュレートする機能です。RHEL 8 では、Soft-RoCE 機能が、サポートされていないテクノロジープレビューとして利用できます。ただし、安定性の問題により、この機能は非推奨になり、RHEL 9 では削除されます。

(BZ#1878207)

6.6.6. ハードウェアの有効化

qla3xxx ドライバーが非推奨になりました。

RHEL 8 では、**qla3xxx** ドライバーが非推奨になりました。このドライバーは、本製品の今後のメジャーリリースではサポートされない可能性が高く、新たに実装することは推奨されません。

(BZ#1658840)

dl2k、**dnet**、**ethoc**、および **dlci** ドライバーは非推奨になりました。

RHEL 8 では、**dl2k**、**dnet**、**ethoc**、および **dlci** ドライバーが非推奨になりました。このドライバーは、本製品の今後のメジャーリリースではサポートされない可能性が高く、新たに実装することは推奨されません。

(BZ#1660627)

6.6.7. ファイルシステムおよびストレージ

elevator カーネルコマンドラインパラメーターが非推奨になりました。

カーネルコマンドラインパラメーターの **elevator** は、すべてのデバイスのディスクスケジューラーを設定するために、以前の RHEL リリースで使用されていました。RHEL 8 では、このパラメーターが非推奨になりました。

アップストリームの Linux カーネルでは、**elevator** パラメーターに対応しなくなりましたが、互換性のために RHEL 8 でも引き続き利用できます。

カーネルは、デバイスのタイプに基づいてデフォルトのディスクスケジューラーを選択することに注意してください。これは通常、最適な設定です。別のスケジューラーが必要な場合は、**udev** ルールまたは Tuned サービスを使用して設定することが推奨されます。選択したデバイスを一致させ、それらのデバイスのスケジューラーのみを切り替えます。

詳しい情報は、[ディスクスケジューラーの設定](#) を参照してください。

(BZ#1665295)

NFSv3 over UDP が無効になりました。

NFS サーバーは、デフォルトで UDP (User Datagram Protocol) ソケットを開いたり、リッスンしなくなりました。バージョン 4 では TCP (Transmission Control Protocol) が必要なため、この変更は NFS バージョン 3 にのみ影響を及ぼします。

RHEL 8 では、NFS over UDP に対応しなくなりました。

(BZ#1592011)

6.6.8. デスクトップ

libgnome-keyring ライブラリーが非推奨になりました。

libgnome-keyring ライブラリーがアップストリームで維持されず、RHEL に必要な暗号化ポリシーに従っていないため、**libsecret** ライブラリーが **libgnome-keyring** ライブラリーを引き継ぎ、**libgnome-keyring** は非推奨となりました。新しい **libsecret** ライブラリーは、必要なセキュリティー標準に準拠する代替ライブラリーです。

(BZ#1607766)

6.6.9. グラフィックインフラストラクチャー

AGP グラフィックカードがサポートされなくなりました。

AGP (Accelerated Graphics Port) バスを使用するグラフィックカードは、Red Hat Enterprise Linux 8 ではサポートされていません。推奨される代替として、PCI-Express バスを備えたグラフィックスカードを使用してください。

(BZ#1569610)

6.6.10. Web コンソール

Web コンソールは、不完全な翻訳への対応を終了しました。

RHEL Web コンソールは、コンソールの翻訳可能な文字列の翻訳率が 50 % 未満の言語に対する翻訳提供を廃止しました。ブラウザがこのような言語に翻訳を要求すると、ユーザーインターフェイスは英語になります。

(BZ#1666722)

6.6.11. 仮想化

virt-manager が非推奨になりました。

Virtual Machine Manager アプリケーション (**virt-manager**) は非推奨になっています。RHEL 8 Web コンソール (**Cockpit**) は、後続のリリースで置き換えられる予定です。したがって、GUI で仮想化を管理する場合は、Web コンソールを使用することが推奨されます。ただし、**virt-manager** で利用可能な機能によっては、RHEL 8 Web コンソールが利用できない場合があります。

(JIRA:RHELPLAN-10304)

RHEL 8 では、仮想マシンのスナップショットへの対応が適切に行われていません。

仮想マシンスナップショットを作成する現在のメカニズムが適切に機能していないため、推奨されなくなりました。これにより、RHEL 8 では、仮想マシンのスナップショットを使用することが推奨されません。

新しい仮想マシンスナップのショットメカニズムは開発中で、RHEL 8 の将来のマイナーリリースで完全に実装される予定です。

(BZ#1686057)

Cirrus VGA 仮想 **GPU** タイプが非推奨に

Red Hat Enterprise Linux の将来のメジャー更新では、KVM 仮想マシンで **Cirrus VGA** GPU デバイスに対応しなくなります。したがって、Red Hat は、Cirrus VGA の代わりに **stdvga** デバイス、**virtio-vga** デバイス、または **qxl** デバイスを使用することを推奨します。

(BZ#1651994)

6.6.12. 非推奨パッケージ

以下のパッケージは非推奨となり、Red Hat Enterprise Linux の今後のメジャーリリースには含まれません。

- 389-ds-base-legacy-tools
- authd

- custodia
- hostname
- libidn
- net-tools
- network-scripts
- nss-pam-ldapd
- sendmail
- yp-tools
- ypbind
- ypserv

6.7. 既知の問題

このパートでは Red Hat Enterprise Linux 8 の既知の問題を説明します。

6.7.1. インストーラーおよびイメージの作成

キックスタートコマンドの **auth** および **authconfig** で AppStream リポジトリが必要になる

インストール中に、キックスタートコマンドの **auth** および **authconfig** で **authselect-compat** パッケージが必要になります。**auth** または **authconfig** を使用したときに、このパッケージがないとインストールに失敗します。ただし、設計上、**authselect-compat** パッケージは AppStream リポジトリでしか利用できません。

この問題を回避するには、BaseOS リポジトリおよび AppStream リポジトリがインストーラーで利用できることを確認するか、インストール中にキックスタートコマンドの **authselect** コマンドを使用します。

(BZ#1640697)

reboot --kexec コマンドおよび **inst.kexec** コマンドが、予測可能なシステム状態を提供しない

キックスタートコマンド **reboot --kexec** またはカーネル起動パラメーター **inst.kexec** で RHEL インストールを実行しても、システムの状態が完全な再起動と同じになるわけではありません。これにより、システムを再起動せずにインストール済みのシステムに切り替えると、予期しない結果が発生することがあります。

kexec 機能は非推奨になり、Red Hat Enterprise Linux の今後のリリースで削除されることに注意してください。

(BZ#1697896)

Anaconda インストールに、最小リソース設定要件の低い制限が含まれています。

Anaconda は最小限のリソース設定を必要とするシステムでインストールを開始し、インストールを成功させるのに必要なリソースに関する以前のメッセージ警告を提供しません。その結果、インストールが失敗し、出力エラーでデバッグや復元の可能性を明確に示すメッセージが提供されない場合があります。

す。この問題を回避するには、システムに、インストールに必要な最小リソース設定 (PPC64 (LE) の場合は 2GB メモリー、x86_64 の場合は 1GB) があることを確認します。これにより、インストールを成功できます。

(BZ#1696609)

reboot --kexec コマンドを使用するとインストールが失敗します。

reboot --kexec コマンドを含むキックスタートファイルを使用すると、RHEL 8 のインストールに失敗します。この問題を回避するには、キックスタートファイルで **reboot --kexec** の代わりに **reboot** コマンドを使用します。

(BZ#1672405)

インストーラーにおける s390x のセキュアなシステム起動のサポート

RHEL 8.1 は、セキュアブートの使用を強制する IBM Z 環境で使用するためにブートディスクの準備をサポートします。インストール時に使用されるサーバーおよびハイパーバイザーの機能により、そのオンディスクフォーマットにセキュアなブートサポートが含まれるかどうかを判断します。インストール時にオンディスク形式に影響を与える方法はありません。

したがって、セキュアブートに対応している環境で RHEL 8.1 をインストールすると、セキュアブートサポートのない環境に移行すると、一部のフェイルオーバーシナリオで起動できても、システムを起動することはできません。

この問題を回避するには、オンディスクブートフォーマットを制御する **zipl** ツールを設定する必要があります。**zipl** を実行する環境がセキュアブートをサポートしている場合でも、以前のオンディスクフォーマットを書き込むように **zipl** を設定できます。RHEL 8.1 のインストールが完了したら、root ユーザーで以下の手動の手順を実行します。

1. 設定ファイル `/etc/zipl.conf` を編集します。
2. "secure=0" を含む行をセクションラベル "defaultboot" に追加します。

Example contents of the ``zipl.conf`` file after the change:

```
[defaultboot]
defaultauto
prompt=1
timeout=5
target=/boot
secure=0
```

3. パラメーターなしで **zipl** ツールを実行します。

これらの手順を実行すると、RHEL 8.1 ブートディスクのオンディスク形式に、セキュアブートサポートが含まれなくなります。その結果、セキュアブートサポートが欠けている環境でもインストールを起動できます。

(BZ#1659400)

SSH から **RHEL 8** 初期セットアップを実行できません。

現在、SSH を使用してシステムにログインする際に、RHEL 8 の初期セットアップインターフェイスは表示されません。これにより、SSH を介して管理される RHEL 8 マシンの初期設定を実行できません。この問題を回避するには、メインのシステムコンソール (ttySO) で初期設定を行ってから、SSH を使用してログインします。

(BZ#1676439)

secure= 起動オプションのデフォルト値は **auto** に設定されていません。

現在、**secure=** 起動オプションのデフォルト値は **auto** に設定されていません。このため、現在のデフォルトが無効であるため、セキュアなブート機能は利用できません。この問題を回避するには、**/etc/zipl.conf** ファイルの **[defaultboot]** セクションに **secure=auto** を手動で設定します。その結果、セキュアなブート機能が利用できるようになります。詳細は、**zipl.conf** の man ページを参照してください。

(BZ#1750326)

Binary DVD.iso ファイルの内容をパーティションにコピーしても、**.treeinfo** ファイルおよび **.discinfo** ファイルがコピーされません。

ローカルインストールで、RHEL 8 Binary DVD.iso イメージファイルの内容をパーティションにコピーする際に、**cp <path>^* <mounted partition>/dir** コマンドの * で、**.treeinfo** ファイルおよび **.discinfo** ファイルがコピーされません。インストールを成功させるには、このファイルが必要です。これにより、BaseOS リポジトリおよび AppStream のリポジトリが読み込まれず、**anaconda.log** ファイルのデバッグ関連のログメッセージでしか問題を確認できません。

この問題を回避するには、不足している **.treeinfo** ファイルおよび **.discinfo** ファイルをパーティションにコピーします。

(BZ#1687747)

Kickstart インストールでは、自己署名 **HTTPS** サーバーを使用できません。

現在では、インストールソースが Kickstart ファイルで指定され、**--noverifyssl** オプションが指定されると、インストーラーは自己署名の **https** サーバーからのインストールに失敗します。

```
url --url=https://SERVER/PATH --noverifyssl
```

この問題を回避するには、Kickstart インストールの開始時に、**inst.noverifyssl** パラメーターをカーネルコマンドラインに追加します。

以下に例を示します。

```
inst.ks=<URL> inst.noverifyssl
```

(BZ#1745064)

6.7.2. ソフトウェア管理

yum repolist が、**skip_if_unavailable=false** で利用できないリポジトリで終了します。

リポジトリ設定オプション **skip_if_unavailable** はデフォルトで以下のように設定されます。

```
skip_if_unavailable=false
```

この設定により、エラーと終了ステータス 1 が付いた最初の利用できないリポジトリで **yum repolist** コマンドを強制的に終了します。したがって、**yum repolist** は、利用可能なリポジトリのリスト表示を続行しません。

各リポジトリの ***.repo** ファイルでこの設定をオーバーライドすることができることに注意してください。

ただし、デフォルトの設定を維持する場合は、以下のオプションを付けて **yum repolist** を使用して問題を回避することができます。

```
--setopt=*.skip_if_unavailable=True
```

(BZ#1697472)

6.7.3. サブスクリプションの管理

syspurpose addons は **subscription-manager attach --auto** 出力に影響しません。

Red Hat Enterprise Linux 8 では、**syspurpose** コマンドラインツールの 4 つの属性 (**role**、**usage**、**service_level_agreement**、および **addons**) が追加されました。現在、**role**、**usage**、および **service_level_agreement** のみが、**subscription-manager attach --auto** コマンドの実行の出力に影響します。**addons** 引数に値を設定しても、自動登録されたサブスクリプションには影響がありません。

(BZ#1687900)

6.7.4. シェルおよびコマンドラインツール

Wayland プロトコルを使用するアプリケーションは、リモートのディスプレイサーバーに転送できません。

Red Hat Enterprise Linux 8.1 では、ほとんどのアプリケーションは、X11 プロトコルの代わりに、デフォルトで **Wayland** プロトコルを使用します。したがって、ssh サーバーは、**Wayland** プロトコルを使用するアプリケーションを転送できませんが、X11 プロトコルを使用するアプリケーションをリモートディスプレイサーバーに転送することは可能です。

この問題を回避するには、アプリケーションを起動する前に環境変数 **gdk_BACKEND=x11** を設定します。その結果、アプリケーションはリモートディスプレイサーバーへ転送できます。

(BZ#1686892)

systemd-resolved.service がシステム起動時の起動に失敗します。

systemd-resolved サービスが起動時に起動できない場合があります。この場合は、以下のコマンドを実行して、システムの起動完了後に手動でサービスを再起動します。

```
# systemctl start systemd-resolved
```

ただし、システムの起動時に **systemd-resolved** が失敗しても、その他のサービスは影響を受けません。

(BZ#1640802)

6.7.5. インフラストラクチャーサービス

dnsmasq での **DNSSEC** のサポート

dnsmasq パッケージには、root サーバーから受け取ったホスト名情報を確認するための Domain Name System Security Extensions (**DNSSEC**) サポートが導入されました。

dnsmasq の DNSSEC 検証は、FIPS 140-2 に準拠していないことに注意してください。連邦情報処理標準 (FIPS) システム上の dnsmasq では DNSSEC を有効化せず、ローカルホスト上のフォワーダーとして適切なリゾルバーを使用します。

(BZ#1549507)

6.7.6. セキュリティー

redhat-support-tool が **FUTURE** 暗号化ポリシーを使用すると機能しない

カスタマーポータル API の証明書が使用する暗号化キーは **FUTURE** のシステム全体の暗号化ポリシーが定義する要件を満たさないため、現時点で **redhat-support-tool** ユーティリティーは、このポリシーレベルでは機能しません。この問題を回避するには、カスタマーポータル API への接続中に **DEFAULT** 暗号化ポリシーを使用します。

(BZ#1802026)

/etc/selinux/config の **SELINUX=disabled** が正常に動作しません。

/etc/selinux/config で **SELINUX=disabled** オプションを使用して SELinux を無効にすると、カーネルが SELinux を有効にして起動し、その後のブートプロセスで無効化モードに切り替わります。これにより、メモリーリークや競合状態が発生し、その結果、カーネルパニックが発生する可能性があります。この問題を回避するには、SELinux を完全に無効にする必要がある場合に [SELinux の使用のシステム起動時に SELinux モードの変更](#) で説明されているように、**selinux=0** パラメーターをカーネルコマンドラインに追加して SELinux を無効にすることが推奨されます。

(JIRA:RHELPLAN-34199)

libselinux-python は、そのモジュールからのみ利用可能

libselinux-python パッケージには、SELinux アプリケーション開発用の Python 2 バインディングのみが含まれ、後方互換性に使用されます。このため、**libselinux-python** コマンドを使用して、デフォルトの RHEL 8 リポジトリで **dnf install libselinux-python** コマンドが利用できなくなりました。

この問題を回避するには、**libselinux-python** モジュールおよび **python27** モジュールの両方を有効にし、以下のコマンドで **libselinux-python** パッケージとその依存関係をインストールします。

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

または、1つのコマンドでインストールプロファイルを使用して **libselinux-python** をインストールします。

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

これにより、各モジュールを使用して **libselinux-python** をインストールできます。

(BZ#1666328)

udica は、**--env container=podman** で開始したときにのみ **UBI 8** コンテナを処理します。

Red Hat Universal Base Image 8 (UBI 8) コンテナは、**podman** の値ではなく、**コンテナ** 環境変数を **oci** 値に設定します。これにより、**udica** ツールがコンテナ JavaScript Object Notation (JSON) ファイルを分析しなくなります。

この問題を回避するには、`--env container=podman` パラメーターを指定して、`podman` コマンドで UBI 8 コンテナを起動します。そのため、`udica` は、上記の回避策を使用している場合に限り、UBI 8 コンテナの SELinux ポリシーを生成することができます。

(BZ#1763210)

`rpm-plugin-selinux` パッケージを削除すると、システムからすべての `selinux-policy` パッケージが削除されます。

`rpm-plugin-selinux` パッケージを削除すると、マシン上で SELinux が無効になります。また、システムからすべての `selinux-policy` パッケージも削除されます。`rpm-plugin-selinux` パッケージを繰り返しインストールしてから、`selinux-policy-targeted` ポリシーが以前に存在していても、`selinux-policy-minimum` SELinux ポリシーをインストールします。ただし、繰り返しインストールしても、ポリシーの変更のために SELinux 設定ファイルが更新されることはありません。このため、`rpm-plugin-selinux` パッケージを再インストールしても、SELinux が無効になります。

この問題を回避するには、以下を実行します。

1. `umount /sys/fs/selinux/` コマンドを実行します。
2. 足りない `selinux-policy-targeted` パッケージを手動でインストールします。
3. ポリシーが `SELINUX=enforcing` と同等になるように `/etc/selinux/config` ファイルを編集します。
4. コマンド `load_policy -i` を実行します。

これにより、SELinux が有効になり、以前と同じポリシーが実行されている状態になります。

(BZ#1641631)

SELinux により、`systemd-journal-gatewayd corosync` が、`corosync` によって作成される共有メモリーファイル上で `newfstatat()` を呼び出さなくなります。

SELinux ポリシーには、`systemd-journal-gatewayd` が `corosync` サービスによって作成されるファイルにアクセスできるルールが含まれません。その結果、SELinux は、`systemd-journal-gatewayd` による、`corosync` で作成された共有メモリーファイルの `newfstatat()` 関数を呼び出しを拒否します。

この問題を回避するには、このシナリオを実現する許可ルールでローカルポリシーモジュールを作成します。SELinux ポリシーの `allow` および `dontaudit` ルールの生成の詳細は、man ページの `audit2allow(1)` を参照してください。以前の回避策により、`systemd-journal-gatewayd` は、Enforcing モードで SELinux により `corosync` で作成した共有メモリーファイルの関数を呼び出すことができます。

(BZ#1746398)

デフォルトのロギング設定がパフォーマンスに与える悪影響。

デフォルトのログ環境設定は、メモリーを 4 GB 以上使用する可能性があり、`rsyslog` で `systemd-journald` を実行している場合は、速度制限値の調整が複雑になります。

詳細は、ナレッジベースの記事 [Negative effects of the RHEL default logging setup on performance and their mitigations](#) を参照してください。

(JIRA:RHELPLAN-10431)

`config.enabled` による `rsyslog` 出力の `Parameter not known` エラー。

rsyslog の出力では、**config.enabled** ディレクティブを使用すると、設定処理エラーで予期しないバグが発生します。これにより、**include()** ステートメントを除き、**config.enabled** ディレクティブを使用すると、**parameter not known** エラーが表示されます。

この問題を回避するには、**config.enabled=on** を設定するか、**include()** ステートメントを使用します。

(BZ#1659383)

特定の **rsyslog** 優先度の文字列が正常に動作しません。

imtcp に **GnuTLS** 優先度文字列を設定して、完成していない暗号化をきめ細かく制御できるようになりました。したがって、**rsyslog** では、以下の優先文字列が正常に動作しません。

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

この問題を回避するには、正しく機能する優先度文字列のみを使用します。

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

したがって、現在の設定は、正しく機能する文字列に限定する必要があります。

(BZ#1679512)

SHA-1 署名を使用するサーバーへの接続が **GnuTLS** で動作しません。

証明書の **SHA-1** 署名は、GnuTLS セキュアな通信ライブラリーにより、セキュアでないものとして拒否されます。したがって、TLS のバックエンドとして GnuTLS を使用するアプリケーションは、このような証明書を提供するピアへの TLS 接続を確立することができません。この動作は、その他のシステム暗号化ライブラリーと一貫性がありません。この問題を回避するには、サーバーをアップグレードして、**SHA-256** または強力なハッシュを使用して署名した証明書を使用するか、**LEGACY** ポリシーに切り替えます。

(BZ#1628553)

TLS 1.3 は、**FIPS** モードの **NSS** で動作しません。

TLS 1.3 は、**FIPS** モードで動作しているシステムでは対応していません。その結果、相互運用性に **TLS 1.3** を必要とする接続が、**FIPS** モードで動作しているシステムで機能しません。

その接続を有効にするには、システムの **FIPS** モードを無効にするか、ピアで **TLS 1.2** のサポートを有効にします。

(BZ#1724250)

OpenSSL が、生の **RSA** または **RSA-PSS** の署名に対応していない **PKCS #11** トークンを誤って処理します。

OpenSSL ライブラリーは、**PKCS #11** トークンの鍵関連の機能を検出しません。したがって、生の **RSA** または **RSA-PSS** の署名に対応しないトークンで署名が作成されると、**TLS** 接続の確立に失敗します。

この問題を回避するには、**/etc/pki/tls/openssl.cnf** ファイルの **crypto_policy** セクションの末尾にある **.include** 行の後に、以下の行を追加します。

```
SignatureAlgorithms =  
RSA+SHA256:RSA+SHA512:RSA+SHA384:ECDSA+SHA256:ECDSA+SHA512:ECDSA+SHA384  
MaxProtocol = TLSv1.2
```

これにより、このシナリオで TLS 接続を確立できます。

([BZ#1685470](#))

OpenSSL TLS ライブラリーは、**PKCS#11** トークンが、生の **RSA** 署名または **RSA-PSS** 署名の作成に対応しているかどうかを検出しません。

TLS-1.3 プロトコルでは、**RSA-PSS** 署名の対応が必要です。**PKCS#11** トークンが、生の **RSA** 署名または **RSA-PSS** 署名に対応していない場合、**OpenSSL TLS** ライブラリーを使用するサーバーアプリケーションは、**PKCS#11** トークンが保持していると、**RSA** 鍵を使用した作業に失敗します。これにより、**TLS** 通信が失敗します。

この問題を回避するには、利用可能な最大の **TLS** プロトコルバージョンとして **TLS-1.2** バージョンを使用するように、サーバーまたはクライアントを設定します。

([BZ#1681178](#))

openssl は、**TLS 1.3** の **CertificateRequest** メッセージに、不正な **status_request** 拡張を生成します。

OpenSSL サーバーは、**status_request** 拡張とクライアント証明書ベースの認証サポートが有効な場合に、**CertificateRequest** メッセージに正しくない **status_request** 拡張を送信します。この場合、**OpenSSL** は **RFC 8446** プロトコルに準拠する実装と同時に動作しません。その結果、**CertificateRequest** メッセージの拡張を適切に検証するクライアントは **OpenSSL** サーバーとの接続を中止します。この問題を回避するには、接続のいずれかで **TLS 1.3** プロトコルのサポートを無効にするか、**OpenSSL** サーバーの **status_request** のサポートを無効にします。これにより、サーバーが不正なメッセージを送信しなくなります。

([BZ#1749068](#))

ssh-keyscan が、**FIPS** モードでサーバーの **RSA** 鍵を取得できません。

FIPS モードで **RSA** 署名の **SHA-1** アルゴリズムが無効になっています。これにより、**ssh-keyscan** ユーティリティーがそのモードで稼働しているサーバーの **RSA** 鍵を取得できなくなります。

この問題を回避するには、代わりに **ECDSA** 鍵を使用するか、サーバーの **/etc/ssh/ssh_host_rsa_key.pub** ファイルから鍵をローカルに取得します。

([BZ#1744108](#))

Audit ルールの **scap-security-guide PCI-DSS** 修復が適切に動作しません。

scap-security-guide パッケージには、修正の組み合わせと、以下のいずれかのシナリオで生じるチェックが含まれます。

- 監査ルールの誤った修正
- 渡されたルールが失敗とマークされた誤検出を含むスキャン評価

これにより、**RHEL 8.1** インストールプロセス時に、インストール済みシステムのスキャンが、失敗またはエラーとして **Audit** ルールを報告します。

この問題を回避するには、ナレッジベースの記事 [RHEL-8.1 workaround for remediating and scanning with the scap-security-guide PCI-DSS profile](#) の手順に従ってください。

(BZ#1754919)

SSG における完全依存ルールの特定のセットが失敗する可能性があります。

ルールとその依存関係の順序付けを定義しないため、ベンチマークの **SCAP Security Guide** (SSG) ルールの修正が失敗する可能性があります。たとえば、特定の順番で複数のルールを実行する必要がある場合、あるルールがコンポーネントをインストールし、別のルールが同じコンポーネントを設定した場合すると、それらは正しくない順序で実行される可能性があります。修正によってエラーが報告されます。この問題を回避するには、修正を回実行して、番目の実行で依存ルールを修正します。

(BZ#1750755)

コンテナのセキュリティおよびコンプライアンススキャンを行うユーティリティーが利用できません。

Red Hat Enterprise Linux 7 では、Atomic テクノロジーに基づいた Docker コンテナのスキャンに、**oscap-docker** ユティリティーを使用できました。Red Hat Enterprise Linux 8 では、Docker 関連、および Atomic 関連の **OpenSCAP** コマンドが利用できません。

この問題に対処するには、カスタマーポータルの記事 [Using OpenSCAP for scanning containers in RHEL 8](#) を参照してください。したがって、現在、RHEL 8 のコンテナのセキュリティおよびコンプライアンススキャンには、未対応の方法と制限のある方法のみを使用できます。

(BZ#1642373)

OpenSCAP が、仮想マシンおよびコンテナのオフラインスキャンを提供しません。

OpenSCAP のコードベースをリファクタリングすると、特定の RPM プロブがオフラインモードで仮想マシンおよびコンテナのファイルシステムをスキャンするのに失敗していました。このため、以下のツールは、**openscap-utils** パッケージである **oscap-vm** および **oscap-chroot** から削除されました。また、**openscap-containers** パッケージも完全に削除されました。

(BZ#1618489)

OpenSCAP の **rpmverifypackage** が正常に動作しません。

rpmverifypackage プロブにより、システムコール **chdir** および **chroot** が 2 回呼び出されます。これにより、カスタムの OVAL (Open Vulnerability and Assessment Language) コンテンツを使用した **OpenSCAP** をスキャンする際にこのプロブを使用していると、エラーが発生します。

この問題を回避するには、コンテンツで OVAL テスト **rpmverifypackage_test** を使用しないようにするか、**rpmverifypackage_test** が使用されていない **scap-security-guide** パッケージのコンテンツのみを使用します。

(BZ#1646197)

SCAP Workbench が、カスタムプロファイルから結果ベースの修正を生成できない

SCAP Workbench ツールを使用してカスタムプロファイルから結果ベースの修正ロールを生成しようとすると、次のエラーが発生します。

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

この問題を回避するには、**oscap** コマンドを、**--tailoring-file** オプションとともに使用します。

(BZ#1640715)

OSCAP Anaconda Addon がすべてのパッケージをテキストモードでインストールしません。

OSCAP Anaconda Addon プラグインは、インストールがテキストモードで実行している場合、システムインストーラーによってインストールに選択されているパッケージのリストを変更することはできません。これにより、キックスタートを使用してセキュリティーポリシープロファイルが指定され、インストールがテキストモードで実行している場合に、インストール中にセキュリティーポリシーに必要な追加パッケージがインストールされません。

この問題を回避するには、グラフィカルモードでインストールを実行するか、キックスタートファイルの **%packages** セクションにあるセキュリティーポリシーで、セキュリティーポリシープロファイルに必要なパッケージをすべて指定します。

これにより、セキュリティーポリシープロファイルで必要となるパッケージは、上記の回避策のいずれかを行わなければ RHEL インストールインストール時にインストールされません。また、インストール後のシステムは、指定のセキュリティーポリシープロファイルと互換性がありません。

(BZ#1674001)

oscap Anaconda Addon がカスタムプロファイルを正しく処理しません。

OSCAP Anaconda Addon プラグインは、個別のファイルでカスタマイズを使用したセキュリティープロファイルを適切に処理しません。これにより、対応する Kickstart セクションで適切に指定しても、RHEL グラフィカルインストールでカスタマイズしたプロファイルは利用できません。

この問題を回避するには、ナレッジベースの記事 [Creating a single SCAP data stream from an original DS and a tailoring file](#) を参照してください。この回避策により、RHEL グラフィカルインストールでカスタマイズした SCAP プロファイルを使用できます。

(BZ#1691305)

6.7.7. ネットワーク

arptables の詳細出力のフォーマットが、RHEL 7 のユーティリティーのフォーマットに一致するようになりました。

RHEL 8 では、**iptables-arptables** パッケージが **arptables** ユーティリティーの代わりに **nftables** を提供します。以前では、**arptables** の詳細な出力は、コンマでのみカウンター値を区切っていました。一方、RHEL 7 の **arptables** は、スペースとコンマの両方で出力を区切っていました。これにより、**arptables -v -L** コマンドの出力を分析した RHEL 7 で作成されたスクリプトを使用した場合は、このスクリプトを調整する必要がありました。この非互換性が修正されました。これにより、RHEL 8.1 の **arptables** が、スペースとコンマの両方でカウンター値を分けるようになりました。

(BZ#1676968)

nftables が多次元の IP セットタイプに対応しません。

nftables パケットフィルタリングフレームワークは、連結と区間を持つセット型に対応しません。これにより、**hash:net,port** などの多次元 IP セットタイプを、**nftables** と共に使用することができません。

この問題を回避するには、多次元 IP セットタイプが必要な場合に、**iptables** フレームワークを **ipset** ツールと共に使用してください。

(BZ#1593711)

GRO が無効になっていると IPsec オフロード中に IPsec ネットワークトラフィックが失敗します。

デバイスで汎用受信オフロード (GRO) が無効になっていると、IPSec オフロードは機能しません。IPsec オフロードがネットワークインターフェイスで設定され、GRO がそのデバイスで無効になっていると、IPsec ネットワークトラフィックに失敗します。

この問題を回避するには、デバイスで GRO を有効にしたままにします。

(BZ#1649647)

6.7.8. カーネル

i40iw モジュールがシステムの起動時に自動的に読み込まれない

多くの i40e NIC で iWarp に対応しておらず、**i40iw** モジュールがサスペンド/レジュームに完全に対応していないため、このモジュールがデフォルトで自動的に読み込まれず、サスペンド/レジューム正しく機能させることができません。この問題を回避するには、`/lib/udev/rules.d/90-rdma-hw-modules.rules` ファイルを手動で編集して、**i40iw** の自動読み込みを有効にします。

また、同じマシンにある i40e デバイスに、別の RDMA デバイスがインストールされている場合に、i40e 以外の RDMA デバイスで、**i40iw** モジュールを含む、有効なすべての RDMA スタックモジュールを読み込む **rdma** サービスが起動します。

(BZ#1623712)

fadump を使用すると、ネットワークインターフェイスの名前が **kdump-`<interface-name>`** に変更されます。

ファームウェアアシストダンプ (**fadump**) を使用して **vmcore** をキャプチャーし、SSH または NFS プロトコルでリモートマシンに保存する場合は、`<interface-name>` がジェネリックであれば (`*eth#` や `net#`)、ネットワークインターフェイスは **kdump- `<interface-name>`** に名前が変更されます。この問題は、初期 RAM ディスク (**initrd**) の **vmcore** 取得スクリプトが、ネットワークインターフェイス名に接尾辞 **kdump-** を追加して、永続的な名前付けを保護するために発生します。同じ **initrd** が通常の起動にも使用されるため、実稼働環境のカーネルのインターフェイス名も変更されます。

(BZ#1745507)

永続メモリーの量が多くなると、システムの起動プロセス時に遅延が発生します。

メモリーの初期化がシリアル化されるため、永続メモリーのサイズが大きいシステムは起動に時間がかかります。したがって、`/etc/fstab` ファイルに永続メモリーのファイルシステムがあると、デバイスが利用できるようになるまで待つ際に、システムがタイムアウトする場合があります。この問題を回避するには、`/etc/systemd/system.conf` ファイルの **DefaultTimeoutStartSec** オプションを十分に大きな値に設定します。

(BZ#1666538)

KSM が、**NUMA** メモリーポリシーを無視することがあります。

merge_across_nodes=1 パラメーターで、カーネル共有メモリー (KSM) 機能を有効にすると、KSM は、**mbind()** 関数が設定したメモリーポリシーを無視し、一部のメモリーから、ポリシーに一致しない **NUMA** (Non-Uniform Memory Access) ノードにページをマージできない場合があります。

この問題を回避するには、KSM を無効にするか、**QEMU** で **NUMA** メモリーバインディングを使用する場合は **merge_across_nodes** パラメーターを **0** に設定します。これにより、**KVM** 仮想マシンに設定した **NUMA** メモリーポリシーが期待どおりに機能します。

(BZ#1153521)

fadump が有効な場合は、システムが起動時に緊急モードに切り替わります。

fadump (kdump) または **dracut squash** モジュールが **initramfs** スキームで有効化されると、システムが緊急モードに切り替わります。これは、**systemd** マネージャーがマウント情報の取得とマウントする LV パーティションの設定に失敗するためです。この問題を回避するには、以下のカーネルコマンドラインパラメーター **rd.lvm.lv=<vg>/<LV>** を追加し、失敗した LV パーティションを適切に検出してマウントします。これにより、上述のシナリオでシステムが正常に起動するようになります。

(BZ#1750278)

kdump カーネルコマンドラインで **irqpoll** を使用すると **vmcore** 生成に失敗します。

AWS (Amazon Web Services) クラウドプラットフォームで実行している 64 ビット ARM アーキテクチャー上には **nvme** ドライバーの既存の根本的な問題があります。この問題により、**irqpoll** **kdump** コマンドライン引数が最初のカーネルにある場合、**vmcore** 生成は失敗します。したがって、カーネルクラッシュ後に **vmcore** が `/var/crash/` ディレクトリーにダンプされません。この問題を回避するには、以下を実行します。

1. `/etc/sysconfig/kdump` ファイルの **KDUMP_COMMANDLINE_REMOVE** キーに **irqpoll** を追加します。
2. **systemctl restart kdump** コマンドを実行して、**kdump** サービスを再起動します。

その結果、最初のカーネルが正常に起動し、カーネルクラッシュ時に **vmcore** がキャプチャーされることが予想されます。

(BZ#1654962)

RHEL 8 で、デバッグカーネルがクラッシュキャプチャー環境で起動に失敗します。

デバッグカーネルのメモリー要求の性質により、デバッグカーネルが使用中で、カーネルパニックが発生すると、問題が発生します。その結果、デバッグカーネルはキャプチャーカーネルとして起動できず、代わりにスタックトレースが生成されます。この問題を回避するには、クラッシュカーネルメモリーを適宜増やします。これにより、デバッグカーネルが、クラッシュキャプチャー環境で正常に起動します。

(BZ#1659609)

softirq の変更により、負荷が大きい場合に **localhost** インターフェイスが **UDP** パケットをドロップできるようになります

Linux カーネルのソフトウェア割り込み (**softirq**) 処理の変更を行い、サービス拒否 (DOS) の影響を軽減します。その結果、これにより、**localhost** インターフェイスが負荷が大きいときに **localhost** インターフェイスが **UDP** (User Datagram Protocol) パケットをドロップする状況が生じます。

この問題を回避するには、ネットワークデバイスのバックログバッファのサイズを `6000:` の値を増やします。

```
echo 6000 > /proc/sys/net/core/netdev_max_backlog
```

Red Hat テストでは、パケットロスを防ぐために十分な値でした。より負荷の高いシステムには、より大きなバックログ値が必要になる場合があります。バックログが増えると、ローカルホストインターフェイスのレイテンシーが増加する可能性があります。

その結果、バッファを増やし、より多くのパケットを処理まで待機できるため、ローカルホストパケットをドロップする可能性を減らすことができます。

(BZ#1779337)

6.7.9. ハードウェアの有効化

HP NMI ウォッチドッグがクラッシュダンプを生成しない場合があります。

HP NMI ウォッチドッグの **hpwdt** ドライバーは、マスク不可割り込み (NMI) が **perfmon** ドライバーにより使用されたため、HPE ウォッチドッグタイマーが生成した NMI を要求できない場合があります。したがって、**hpwdt** は、クラッシュダンプを生成するためにパニックを呼び出さない場合があります。

(BZ#1602962)

QL41000 カードで設定したテストシステムに RHEL 8.1 をインストールするとカーネルパニックが発生します。

QL41000 カードで設定したテストシステム上に RHEL 8.1 をインストールすると、システムは **000000000000003c** カードでカーネル NULL ポインターの逆参照を処理できません。これにより、カーネルパニックのエラーが発生します。この問題に対する回避策はありません。

(BZ#1743456)

cxgb4 ドライバーにより **kdump** カーネルでクラッシュします。

vmcore ファイルに情報を保存しようとする、**kdump** カーネルがクラッシュします。そのため、**cxgb4** ドライバーにより、**kdump** カーネルが、後で分析するためにコアを保存できなくなります。この問題を回避するには、**kdump** カーネルコマンドラインに "novmcoredd" パラメーターを追加して、コアファイルを保存できるようにします。

(BZ#1708456)

6.7.10. ファイルシステムおよびストレージ

特定の SCSI ドライバーが過剰な量のメモリーを使用することがあります。

SCSI ドライバーの中には、RHEL 7 よりも大容量のメモリーを使用しているものがあります。ファイバーチャネルホストバスアダプター (HBA) での vPort 作成など、特定のケースでは、システム設定によってはメモリー使用量が過剰になる可能性があります。

メモリー使用量の増加は、ブロックレイヤーでメモリーの事前割り当てにより発生します。RHEL 8 の各 I/O リクエストに対して、マルチキューブロックデバイススケジューリング (BLK-MQ) とマルチキューの SCSI スタック (SCSI-MQ) の両方がメモリーを事前に割り当てているため、メモリー使用量が増えます。

(BZ#1698297)

UDS が再構築が終了するまで VDO は一時停止できません。

VDO (Virtual Data Optimizer) ボリュームが不完全なシステムのシャットダウン後に起動すると、Universal Deduplication Service (UDS) インデックスが再構築されます。UDS インデックスが再構築されても、**dmsetup suspend** コマンドを使用して VDO ボリュームを一時停止しようとする、suspend コマンドが応答しなくなることがあります。このコマンドは、再構築が完了した後にのみ終了します。

UDS インデックスが大きい VDO ボリュームでは、応答の遅延が明白になります。このため、再構築に時間がかかります。

(BZ#1737639)

NFS 4.0 パッチは、オープンの高ワークロードでパフォーマンスが低下する可能性があります。

以前、場合によっては NFS のオープン操作で、サーバー上のファイルが削除されたり、名前が変更されたりするという事実を見落とすというバグが修正されています。ただし、この修正により、多くのオープンな操作が必要とするワークロードのパフォーマンスが遅くなる可能性があります。この問題を回避するには、NFS バージョン 4.1 以降を使用します。これは、多くの場合においてクライアントに委譲を付与するように改善されています。このため、クライアントがローカルに素早く安全にオープン操作を実行できます。

(BZ#1748451)

6.7.11. 動的プログラミング言語、Web サーバー、およびデータベースサーバー

nginx がハードウェアセキュリティトークンからサーバー証明書をロードできません。

nginx の Web サーバーは、PKCS#11 モジュールを利用してハードウェアセキュリティトークンから直接 TLS 秘密鍵を読み込むようになりました。ただし、現在では、PKCS#11 URI を使用してハードウェアのセキュリティトークンからサーバー証明書を読み込むことはできません。この問題を回避するには、ファイルシステム上にサーバー証明書を保存します。

(BZ#1668717)

php-opcache が PHP 7.2 とともにインストールされると、PHP 7.2 で **php-fpm** により、SELinux AVC 拒否が発生します。

php-opcache パッケージがインストールされると、FastCGI Process Manager (**php-fpm**) により SELinux AVC 拒否が生じます。この問題を回避するには、`/etc/php.d/10-opcache.ini` ファイルのデフォルト設定を以下のように変更します。

```
opcache.huge_code_pages=0
```

この問題は、**php: 7.3** ではなく、**php:7.2** ストリームにのみ影響することに注意してください。

(BZ#1670386)

6.7.12. コンパイラーおよび開発ツール

ltrace ツールが、関数呼び出しを報告しません。

すべての RHEL コンポーネントに適用されるバイナリー強化の改善により、**ltrace** ツールが、RHEL コンポーネントからのバイナリーファイルの関数呼び出しを検出できなくなりました。これにより、**ltrace** の出力では、このようなバイナリーファイルで使用されたときに検出される呼び出しが報告されなくなるため、空になります。現在利用できる回避策はありません。

ただし、**ltrace** では、それぞれの強化フラグを使用せずに構築されたカスタムバイナリーファイルの呼び出しは報告されます。

(BZ#1618748)

6.7.13. ID 管理

GSSAPI 認証を使用する場合に、期限切れのアカウントを持つ AD ユーザーのログインが可能。

アカウントの期限が切れたかどうかを参照するために SSSD が使用する **accountExpires** 属性は、デフォルトでグローバルカタログにレプリケートされません。これにより、GSSAPI 認証を使用する場合に、期限切れのアカウントを持つユーザーはログインできます。この問題に対処するには、**sssd.conf**

ファイルで `ad_enable_gc=False` を指定してグローバルカタログサポートを無効にすることができます。この設定では、GSSAPI 認証を使用する場合に、期限切れのアカウントを持つユーザーはアクセスが拒否されます。

SSSD は、このシナリオにおいて各 LDAP サーバーに個別に接続するため、接続数を増やすことができることに注意してください。

(BZ#1081046)

`--agent-uid pkidbuser` オプションを指定して `cert-fix` ユーティリティーを使用すると、証明書システムが破損します。

`--agent-uid pkidbuser` オプションを指定して `cert-fix` ユーティリティーを使用すると、証明書システムの LDAP 設定が破損します。したがって、証明書システムは不安定になり、システムの復元に手動の操作が必要になる可能性があります。

(BZ#1729215)

`/etc/nsswitch.conf` を変更するには、手動によるシステムの再起動が必要です。

`authselect select profile_id` コマンドの実行など、`/etc/nsswitch.conf` ファイルを変更した場合は、関連するすべてのプロセスで、更新バージョンの `/etc/nsswitch.conf` ファイルが使用されるように、システムを再起動する必要があります。システムを再起動できない場合は、システムを Active Directory (System Security Services Daemon (SSSD) または `winbind`) に追加するサービスを再起動します。

(BZ#1657295)

IdM で AD 信頼のサポートを有効にすると、必要な DNS レコードに関する情報が表示されません。

外部 DNS 管理を使用した Red Hat Enterprise Linux Identity Management (IdM) インストールで Active Directory (AD) 信頼のサポートを有効にすると、必要な DNS レコードに関する情報が表示されません。AD へのフォレストの信頼は、必要な DNS レコードが追加されるまで成功しません。この問題を回避するには、`ipa dns-update-system-records --dry-run` コマンドを実行して、IdM が必要とするすべての DNS レコードのリストを取得します。IdM ドメインの外部 DNS が必要な DNS レコードを定義すると、AD へのフォレスト信頼を確立できるようになります。

(BZ#1665051)

SSSD が、ローカルユーザーの LDAP グループメンバーシップを誤って返します。

SSSD (System Security Services Daemon) がローカルファイルのユーザーに対応している場合、ファイルプロバイダーには、他のドメインのグループメンバーシップが含まれません。これにより、ローカルユーザーが LDAP グループのメンバーである場合、`id local_user` コマンドはユーザーの LDAP グループメンバーシップを返しません。この問題を回避するには、システムが `/etc/nsswitch.conf` ファイルのユーザーのグループメンバーシップを調べるデータベースの順序を元に戻すか、`sss files` を `files sss` に置き換えるか、以下を追加して、暗黙的な `files` ドメインを無効にします。

```
enable_files_domain=False
```

`/etc/sss/sss.conf` ファイルの `[sss]` セクションに移動します。

これにより、`id local_user` が、ローカルユーザーの正しい LDAP グループメンバーシップを返します。

(BZ#1652562)

RHEL 8 で、**systemd-user** のデフォルトの PAM 設定が変更になり、SSSD の動作に影響を及ぼす可能性があります。

Red Hat Enterprise Linux 8 では、プラグ可能な認証モジュール (PAM) スタックが変更されました。たとえば、**systemd** ユーザーセッションは、PAM サービス **systemd-user** を使用して PAM 対話を開始するようになりました。このサービスは、PAM サービスの **system-auth** を再帰的に追加します。ここでは、**pam_sss.so** インターフェイスが含まれる場合もあります。これは、SSSD アクセス制御が常に呼び出されることを意味します。

RHEL 8 システムのアクセス制御ルールを規定する場合は、変更にご注意してください。たとえば、**systemd-user** サービスを、許可されたサービスリストに追加できます。

IPA HBAC、AD GPO などの一部のアクセス制御メカニズムでは、**systemd-user** サービスが、許可されたサービスリストにデフォルトで追加されているため、何もする必要はありません。

(BZ#1669407)

SSSD が同じ優先順位を持つ複数の証明書一致ルールを正しく処理しません。

指定した証明書が、優先順位が同じ複数の証明書の一致ルールに一致する場合、System Security Services Daemon (SSSD) は、いずれか一方のみを使用します。これを回避するには、`|` (or) 演算子で連結した個々のルールのフィルターで設定される LDAP フィルターを持つ 1 つの証明書一致ルールを使用します。証明書一致ルールの例は、man ページの `sss-certamp (5)` を参照してください。

(BZ#1447945)

複数のドメインが定義されている場合は、プライベートグループを `auto_private_group = hybrid` で作成できません。

複数のドメインが定義され、最初のドメイン以外のドメインによってハイブリッドオプションが使用されると、プライベートグループは、`auto_private_group = hybrid` オプションでの作成に失敗します。暗黙的なファイルドメインが **sssd.conf** ファイルの **AD** または **LDAP** ドメインとともに定義され、**MPG_hybrid** としてマークされていない場合、SSSD は `uid=gid` のユーザーに対してプライベートグループを作成し、この `gid` を持つグループは AD または LDAP に存在しません。

`sssd_nss` レスポンダーは、最初のドメインの `auto_private_groups` オプションの値のみを確認します。これにより、RHEL 8 でデフォルトのセットアップを含む、複数のドメインが設定されているセットアップでは、`auto_private_group` オプションを指定しても効果が得られません。

この問題を回避するには、**sssd.conf** の `sssd` セクションで `enable_files_domain = false` を設定します。その結果、`enable_files_domain` オプションが `false` に設定されていると、`sssd` は、アクティブなドメインリストの開始に `id_provider=files` とともにドメインを追加しないため、このバグは発生しません。

(BZ#1754871)

python-ply は FIPS との互換性がありません。

python-ply パッケージの YACC モジュールは、MD5 ハッシュアルゴリズムを使用して YACC 署名のフィンガープリントを生成します。ただし、FIPS モードは、セキュリティ以外のコンテキストでのみ許可される MD5 の使用をブロックします。そのため、**python-ply** は FIPS と互換性がありません。FIPS モードのシステムでは、`ply.yacc()` への呼び出しに失敗し、次のエラーメッセージが表示されます。

```
"UnboundLocalError: local variable 'sig' referenced before assignment"
```

この問題は **python-pyparser** および **python-cffi** のいくつかのユースケースに影響します。この問題を回避するには、`/usr/lib/python3.6/site-packages/ply/yacc.py` ファイルの 2966 行目を変更し、`sig =`

`md5()` を `sig = md5(usedforsecurity=False)` に置き換えます。これにより、`python-ply` を FIPS モードで使用できます。

(BZ#1747490)

6.7.14. デスクトップ

Wayland セッションの制限

Red Hat Enterprise Linux 8 では、GNOME 環境および GNOME Display Manager (GDM) のデフォルトのセッションタイプとして、以前の RHEL メジャーバージョンで使用されていた X11 セッションに代わり、**Wayland** が使用されます。

以下の機能は、**Wayland** の下では現在利用できないか、期待どおりに動作しません。

- **Wayland** では、マルチ GPU の設定が利用できません。
- `xrandr` などの X11 設定ユーティリティーは、ハンドリング、解像度、ローテーション、レイアウトのアプローチが異なるため、**Wayland** では動作しません。ディスプレイ機能は、GNOME 設定を使用して設定できます。
- 画面の録画とリモートデスクトップでは、アプリケーションが **Wayland** のポータル API をサポートする必要があります。特定のレガシーアプリケーションはポータル API をサポートしません。
- **Wayland** では、ポインターのアクセシビリティーは利用できません。
- クリップボードマネージャーは利用できません。
- **Wayland** 上の GNOME Shell は、ほとんどのレガシー X11 アプリケーションで発行されたキーボードグラブを無視します。`/org/gnome/mutter/wayland/xwayland-grab-access-rules` GSettings キーを使用して、X11 アプリケーションがキーボードグラブを発行するようにできます。デフォルトでは、**Wayland** 上の GNOME Shell により、以下のアプリケーションがキーボードグラブを発行できます。
 - **GNOME Boxes**
 - **Vinagre**
 - **Xephyr**
 - **virt-manager**、**virt-viewer**、および **remote-viewer**
 - **vncviewer**
- ゲスト仮想マシン (VM) 内の **Wayland** には安定性とパフォーマンスの問題があります。RHEL は、仮想マシンで実行している場合に X11 セッションに自動的にフォールバックします。

X11 GNOME セッションを使用していた RHEL 7 システムから RHEL 8 にアップグレードすると、システムでは引き続き X11 が使用されます。また、次のグラフィックドライバーが使用されている場合は、自動的に X11 にフォールバックします。

- プロプライエタリー NVIDIA ドライバー
- **cirrus** ドライバー
- **mga** ドライバー

- **aspeed** ドライバー

Wayland の使用は手動で無効にできます。

- GDM の Wayland を無効にするには、`/etc/gdm/custom.conf` ファイルに **WaylandEnable=false** オプションを設定します。
- GNOME セッションで Wayland を無効にするには、ログイン名を入力してから、ログイン画面の歯車メニューでレガシーの X11 オプションを選択します。

Wayland の詳細は <https://wayland.freedesktop.org/> を参照してください。

(BZ#1797409)

ドラッグアンドドロップが、デスクトップとアプリケーション間で機能しません。

gnome-shell-extensions パッケージのバグにより、ドラッグアンドドロップ機能は現在、デスクトップとアプリケーションの間では機能しません。この機能のサポートは、今後のリリースで追加される予定です。

(BZ#1717947)

ソフトウェアリポジトリからの **flatpak** リポジトリの無効化ができません。

現時点で、GNOME Software ユーティリティの Software Repositories ツールで **flatpak** リポジトリを無効化または削除することはできません。

(BZ#1668760)

Generation 2 の RHEL 8 仮想マシンが Hyper-V Server 2016 ホストで起動できない場合があります。

Microsoft Hyper-V Server 2016 ホストで実行している仮想マシンで RHEL 8 をゲストオペレーティングシステムとして使用すると、仮想マシンが起動しなくなり、GRUB ブートメニューに戻る場合があります。さらに、以下のエラーが Hyper-V イベントログに記録されます。

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

このエラーは、Hyper-V ホストの UEFI ファームウェアバグが原因で発生します。この問題を回避するには、Hyper-V Server 2019 をホストとして使用します。

(BZ#1583445)

GNOME Shell on Wayland は、ソフトウェアレンダラーを使用すると、動作が遅くなっています。

ソフトウェアレンダラーを使用すると、GNOME Shell を Wayland コンポジター (**GNOME Shell on Wayland**) として使用しても、画面のレンダリングにキャッシュ可能なフレームバッファを使用しません。これにより、**GNOME Shell on Wayland** が遅くなります。この問題を回避するには、GNOME Display Manager (GDM) ログイン画面に移動し、代わりに X11 プロトコルを使用するセッションに切り替えます。そのため、キャッシュ可能なメモリーを使用する **Xorg** ディスプレイサーバーが使用され、上記の状況の **GNOME Shell on Xorg** は、**GNOME Shell on Wayland** に比べて速く動作します。

(BZ#1737553)

システムクラッシュにより、**fadump** 設定が失われることがあります。

この問題は、ファームウェア支援ダンプ (fadump) が有効になっているシステムで確認されています。

また、ブートパーティションは XFS などのジャーナリングファイルシステムにあります。システムクラッシュにより、ダンプサポートが有効でない古い **initrd** をブートローダーがロードする可能性があります。したがって、復元後、システムは **vmcore** ファイルをキャプチャーせず、**fadump** 設定が失われる可能性があります。

この問題を回避するには、以下を実行します。

- **/boot** が別のパーティションの場合は、以下を実行します。
 1. **kdump** サービスを再起動します。
 2. **root** ユーザーで以下のコマンドを実行するか、**CAP_SYS_ADMIN** 権限を持つユーザーアカウントを使用します。

```
# fsfreeze -f
# fsfreeze -u
```

- **/boot** が別のパーティションではない場合は、システムを再起動します。

(BZ#1723501)

ldap_id_use_start_tls オプションのデフォルト値を使用する場合の潜在的なリスク

ID ルックアップに TLS を使用せずに **ldap://** を使用すると、攻撃ベクトルのリスクが生じる可能性があります。特に、中間者 (MITM) 攻撃は、攻撃者が、たとえば、LDAP 検索で返されたオブジェクトの UID または GID を変更することによってユーザーになりすますことを可能にする可能性があります。

現在、TLS を強制する SSSD 設定オプション **ldap_id_use_start_tls** は、デフォルトで **false** に設定されています。セットアップが信頼できる環境で動作していることを確認し、**id_provider = ldap** に暗号化されていない通信を使用しても安全かどうかを判断してください。注記: **id_provider = ad** および **id_provider = ipa** は、SASL および GSSAPI によって保護された暗号化接続を使用するため、影響を受けません。

暗号化されていない通信を使用することが安全ではない場合は、**/etc/sss/sss.conf** ファイルで **ldap_id_use_start_tls** オプションを **true** に設定して TLS を強制します。デフォルトの動作は、RHEL の将来のリリースで変更される予定です。

(JIRA:RHELPLAN-155168)

6.7.15. グラフィックインフラストラクチャー

radeon がハードウェアを適切なハードウェアリセットに失敗します。

現在、**radeon** カーネルドライバーは、**kexec** コンテキストでハードウェアを正しくリセットしません。代わりに **radeon** がフェイルオーバーします。これにより、**kdump** サービスの残りの部分が失敗します。

この問題を回避するには、**/etc/kdump.conf** ファイルに以下の行を追加して、**kdump** で **radeon** をブラックリストに指定します。

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

マシンと **kdump** を再起動します。**kdump** の起動後、設定ファイルから **force_rebuild 1** 行が削除される可能性があります。

このシナリオでは、**kdump** 中にグラフィックは利用できませんが、**kdump** は正常に動作します。

(BZ#1694705)

6.7.16. Web コンソール

非特権ユーザーがサブスクリプションページにアクセスできます。

管理者以外のユーザーが Web コンソールの **サブスクリプションページ** に移動すると、Web コンソールでは、Cockpit had an unexpected internal error(Cockpit に予期しない内部エラーが発生しました) という一般的なエラーメッセージが表示されます。

この問題を回避するには、権限のあるユーザーで Web コンソールにサインインし、**Reuse my password for privileged tasks** チェックボックスをチェックします。

(BZ#1674337)

6.7.17. 仮想化

cloud-init を使用した **Microsoft Azure** での仮想マシンのプロビジョニングに失敗します。

現在、**cloud-init** ユーティリティを使用して、Microsoft Azure プラットフォームで RHEL 8 仮想マシンをプロビジョニングすることができません。この問題を回避するには、以下のいずれかの方法を使用します。

- **cloud-init** の代わりに **WALinuxAgent** パッケージを使用して、Microsoft Azure に仮想マシンをプロビジョニングします。
- 以下の設定を **/etc/NetworkManager/NetworkManager.conf** ファイルの **[main]** セクションに追加します。

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

(BZ#1641190)

一部の場において **RHEL 7** ホストの **RHEL 8** 仮想マシンは、**1920x1200** を超える解像度で表示できません。

現在、RHEL 7 ホストシステムで RHEL 8 仮想マシンを実行している場合は、kiosk モードでアプリケーションを実行するなど、仮想マシンのグラフィカル出力を表示する方法によっては、1920x1200 を超える解像度を表示することはできません。そのため、ホストハードウェアがより高い解像度に対応している場合でも、この方法での仮想マシンを表示解像度は最大 1920x1200 のみとなります。

(BZ#1635295)

Windows Server 2019 ホストの **RHEL 8** 仮想マシンで **GUI** ディスプレイのパフォーマンスが下がります。

Windows Server 2019 ホストのグラフィカルモードで RHEL 8 をゲストオペレーティングシステムとして使用すると、GUI ディスプレイのパフォーマンスが下がり、ゲストのコンソール出力に現在必要なよりも長い時間がかかります。

これは、Windows 2019 ホストで既知の問題で、Microsoft が修正を保留しています。この問題を回避するには、SSH を使用してゲストに接続するか、ホストとして Windows Server 2016 を使用します。

(BZ#1706541)

RHEL 仮想マシンのインストールが失敗することがあります。

特定の状況下では、**virt-install** ユーティリティーを使用して作成した RHEL 7 および RHEL 8 仮想マシンが、**--location** オプションを指定すると起動に失敗します。

この問題を回避するには、代わりに **--extra-args** オプションを指定して、ネットワークが到達可能なインストールツリーを指定します。以下にその例を示します。

```
--extra-args="inst.repo=https://some/url/tree/path"
```

これにより、RHEL インストーラーはインストールファイルを正しく検出できるようになります。

(BZ#1677019)

QXL では、**Wayland** を使用する仮想マシンの複数のモニターを表示できません。

remote-viewer ユーティリティーを使用して、Wayland ディスプレイサーバーを使用している仮想マシンのモニターを複数表示すると、仮想マシンが応答なくなり、**Waiting for display** というステータスメッセージが永久に表示されます。

この問題を回避するには、Wayland を使用する仮想マシンの GPU デバイスとして **qxl** の代わりに **virtio-gpu** を使用します。

(BZ#1642887)

virsh iface-* コマンドが一貫して動作しません。

現在、**virsh iface-*** コマンド (**virsh iface-start**、**virsh iface-destroy** など) は、設定の依存関係が原因で頻繁に失敗します。したがって、ホストネットワーク接続の設定および管理には **virsh iface-*** コマンドを使用しないことが推奨されます。代わりに、NetworkManager プログラムとその関連管理アプリケーションを使用します。

(BZ#1664592)

cloud-init を使用して **ESXi** 仮想マシンをカスタマイズし、**VM** を再起動すると、**IP** 設定が失われ、仮想マシンの起動が非常に遅くなります。

現在、**cloud-init** サービスを使用して VMware ESXi ハイパーバイザーで実行している仮想マシンを修正し、静的 IP を使用して、仮想マシンをクローンすると、新しいクローンの仮想マシンを再起動するのにかかる時間が非常に長くなる場合があります。これは、**cloud-init** が仮想マシンの静的 IP を DHCP に書き換えてから、利用可能なデータソースを検索しているからです。

この問題を回避するには、仮想マシンを最初に起動してから **cloud-init** をアンインストールします。その結果、その後の再起動の速度は低下しません。

(BZ#1666961, [BZ#1706482](#))

RHEL 8 仮想マシンが、対応するホストで起動できません。

RHEL 8 仮想マシン (VM) で **pseries -rhel7.6.0-sxxm** マシンタイプを使用すると、DD 2.2 または DD2.3 CPU を使用する **Power9 S922LC for HPC** ホストで起動できない場合があります。

代わりに仮想マシンを起動しようとする、以下のエラーメッセージが出力されます。

```
qemu-kvm: Requested safe indirect branch capability level not supported by kvm
```

この問題を回避するには、仮想マシンの XML 設定を次のように設定します。

■

```
<domain type='qemu' xmlns:qemu='http://libvirt.org/schemas/domain/qemu/1.0'>
  <qemu:commandline>
    <qemu:arg value='-machine'/>
    <qemu:arg value='cap-ibs=workaround'/>
  </qemu:commandline>
```

([BZ#1732726](#), [BZ#1751054](#))

IBM POWER 仮想マシンが、ゼロメモリーの **NUMA** ノードで正常に動作しません。

現在、RHEL 8 ホストで実行している IBM POWER 仮想マシン (VM) が、ゼロメモリー (**memory='0'**) を使用する NUMA ノードで設定されていると、仮想マシンが起動しません。したがって、Red Hat は、RHEL 8 ではゼロメモリー NUMA ノードを持つ IBM POWER 仮想マシンを使用しないことを強く推奨しています。

([BZ#1651474](#))

RHEL 7-ALT ホストから **RHEL 8** への **POWER9** ゲストの移行に失敗する

現在のリリースでは、RHEL 7-ALT ホストシステムから RHEL 8 に POWER9 仮想マシンを移行すると、Migration status: active のステータスで応答がなくなります。

この問題を回避するには、RHEL 7-ALT ホストで Transparent Huge Pages (THP) を無効にすることで、移行が正常に完了します。

([BZ#1741436](#))

AMD EPYC でホストパススルーモードを使用する際に、**SMT CPU** トポロジーが仮想マシンで検出されない

AMD EPYC ホストで行われた CPU ホストパススルーモードで仮想マシンを起動すると、**TOPOEXT** 機能フラグは存在しません。したがって、仮想マシンは、コアごとに複数のスレッドを持つ仮想 CPU トポロジーを検出できません。この問題を回避するには、ホストパススルーの代わりに EPYC CPU モデルを使用して仮想マシンを起動します。

([BZ#1740002](#))

多くの **virtio-blk** ディスクを使用すると、仮想マシンが起動しないことがあります。

多数の **virtio-blk** デバイスを仮想マシンに追加すると、プラットフォームで利用可能な割り込みベクトルの数が使い切られる可能性があります。これが発生すると、仮想マシンのゲスト OS は起動できず、**dracut-initqueue[392]: Warning: Could not boot** エラーが表示されます。

([BZ#1719687](#))

第7章 コンテナに関する主な変更点

Red Hat Enterprise Linux (RHEL) 8.1 用の一連のコンテナイメージが利用できます。以下は、主な変更点です。

- ルートレスコンテナは RHEL 8.1 で完全にサポートされています。
ルートレスコンテナは、管理者権限なしで通常のシステムユーザーにより作成および管理されるコンテナです。これにより、ユーザーはコンテナレジストリーに対する認証情報などの ID を維持できます。

podman コマンドおよび buildah コマンドを使用して、ルートレスコンテナを試行できます。詳細:

- ルートレスコンテナについては、[ルートレスコンテナの設定](#) を参照してください。
- buildah については、[Buildah を使用したコンテナイメージの構築](#) を参照してください。
- podman の場合は、[コンテナの構築、実行、および管理](#) を参照してください。
- toolbox RPM パッケージは、RHEL 8.1 で完全にサポートされています。
toolbox コマンドは、Red Hat CoreOS などのコンテナ指向のオペレーティングシステムで使用されるユーティリティです。toolbox では、これらのツールをインストールする必要なく、使用する多くのトラブルシューティングツールを含むコンテナを起動して、ホストオペレーティングシステムのトラブルシューティングとデバッグを実行できます。

toolbox コマンドを実行すると、ホストへの root アクセスを提供する rhel-tools コンテナが起動します。これは、そのホストを修復するため、または、そのホストで作業するために行われます。
- 新しい [run ラベル付きの Running コンテナについてのドキュメンテーション](#) を参照してください。
- podman パッケージがアップストリームバージョン 1.4.2 にアップグレードされました。RHEL 8.0 で使用されていたバージョン 1.0.0 以降の podman に追加された機能の詳細は、最新の [podman release on Github](#) を参照してください。

第8章 国際化

8.1. RED HAT ENTERPRISE LINUX 8 の多言語

Red Hat Enterprise Linux 8 は、複数の言語のインストールと、要件に応じた言語の変更に対応します。

- 東アジア言語 - 日本語、韓国語、簡体字中国語、および繁体字中国語。
- ヨーロッパ言語 - 英語、ドイツ語、スペイン語、フランス語、イタリア語、ポルトガル語、およびロシア語。

次の表は、さまざまな主要言語に提供されるフォントと入力方法を示しています。

言語	デフォルトフォント (フォントパッケージ)	入力メソッド
英語	dejavu-sans-fonts	
フランス語	dejavu-sans-fonts	
ドイツ語	dejavu-sans-fonts	
イタリア語	dejavu-sans-fonts	
ロシア語	dejavu-sans-fonts	
スペイン語	dejavu-sans-fonts	
ポルトガル語	dejavu-sans-fonts	
簡体字中国語	google-noto-sans-cjk-ttc-fonts、 google-noto-serif-cjk-ttc-fonts	ibus-libpinyin、libpinyin
繁体字中国語	google-noto-sans-cjk-ttc-fonts、 google-noto-serif-cjk-ttc-fonts	ibus-libzhuyin、libzhuyin
日本語	google-noto-sans-cjk-ttc-fonts、 google-noto-serif-cjk-ttc-fonts	ibus-kkc、libkkc
韓国語	google-noto-sans-cjk-ttc-fonts、 google-noto-serif-cjk-ttc-fonts	ibus-hangul、libhangu

8.2. RHEL 8 における国際化の主な変更点

RHEL 8 では、RHEL 7 の国際化に以下の変更が加えられています。

- Unicode 11 コンピューティングの業界標準のサポートが追加されました。

- 国際化は複数のパッケージで配布され、より小さなフットプリントのインストールを可能にします。詳細は、[Using langpacks](#) を参照してください。
- 複数のロケールの **glibc** パッケージの更新が、Common Locale Data Repository (CLDR) と同期するようになりました。

付録A コンポーネント別のチケットリスト

コンポーネント	チケット
NetworkManager-libreswan	BZ#1697329
anaconda	BZ#1628653 , BZ#1673901 , BZ#1671047 , BZ#1689909 , BZ#1689194 , BZ#1584145 , BZ#1637472 , BZ#1696609 , BZ#1672405 , BZ#1687747 , BZ#1745064 , BZ#1659400 , BZ#1655523
audit	BZ#1730382
authselect	BZ#1657295
bcc	BZ#1667043
binutils	BZ#1618748 , BZ#1644391 , BZ#1525406 , BZ#1659437
bpfttrace	BZ#1687802
chrony	BZ#1685469
cloud-init	BZ#1641190 , BZ#1666961
cockpit-appstream	BZ#1658847
cockpit	BZ#1631905 , BZ#1678956 , BZ#1657752 , BZ#1678473 , BZ#1666722
corosync	BZ#1693491
criu	BZ#1689746
crypto-policies	BZ#1678661 , BZ#1660839
cryptsetup	BZ#1676622
distribution	BZ#1685191 , BZ#1657927
dnf-plugins-core	BZ#1722093
dnsmasq	BZ#1549507
dyninst	BZ#1648441
elfutils	BZ#1683705
enscript	BZ#1664366

コンポーネント	チケット
fapolicyd	BZ#1673323
freeradius	BZ#1685546
frr	BZ#1657029
gcc-toolset-9	BZ#1685482
gcc	BZ#1680182
gdb	BZ#1669953, BZ#1187581
gdm	BZ#1678627
glibc	BZ#1663035 , BZ#1701605 , BZ#1651283, BZ#1577438
gnome-shell-extensions	BZ#1717947
gnome-shell	BZ#1704360
gnome-software	BZ#1668760
gnutls	BZ#1628553
grub2	BZ#1583445, BZ#1723501
initial-setup	BZ#1676439
ipa	BZ#1665051 , JIRA:RHELPLAN-15036, BZ#1664719 , BZ#1664718 , BZ#1719767
ipset	BZ#1683711 , BZ#1683713 , BZ#1649090
iptables	BZ#1658734 , BZ#1676968
kernel-rt	BZ#1678887

コンポーネント	チケット
kernel	BZ#1647723, BZ#1656787, BZ#1649087, BZ#1721386, BZ#1564427, BZ#1686755, BZ#1664969, BZ#1714111, BZ#1712272, BZ#1646810, BZ#1728519, BZ#1721961, BZ#1654962, BZ#1635295, BZ#1706541, BZ#1666538, BZ#1685894, BZ#1643980, BZ#1602962, BZ#1697310, BZ#1593711, BZ#1649647, BZ#1153521, BZ#1694705, BZ#1698297, BZ#1348508, BZ#1748451, BZ#1743456, BZ#1708456, BZ#1710480, BZ#1634343, BZ#1652222, BZ#1687459, BZ#1571628, BZ#1571534, BZ#1685552, BZ#1685427, BZ#1663281, BZ#1664359, BZ#1677215, BZ#1659399, BZ#1665717, BZ#1581898, BZ#1519039, BZ#1627455, BZ#1501618, BZ#1401552, BZ#1495358, BZ#1633143, BZ#1503672, BZ#1505999, BZ#1570255, BZ#1696451, BZ#1665295, BZ#1658840, BZ#1660627, BZ#1569610
kexec-tools	BZ#1662911, BZ#1750278, BZ#1520209, BZ#1710288
keycloak-httpd-client-install	BZ#1553890
kmod-kvdo	BZ#1696492 , BZ#1737639
kpatch	BZ#1763780
libcacard	BZ#1615840
libdnf	BZ#1697472
libgnome-keyring	BZ#1607766
libselinux-python-2.8-module	BZ#1666328
libsemanage	BZ#1672638
libssh	BZ#1610883
libstoragemgmt	BZ#1626415
libvirt	BZ#1664592, BZ#1526548, BZ#1528684
linuxptp	BZ#1677217, BZ#1685467
lorax	BZ#1663950 , BZ#1709594 , BZ#1689140
lvm2	BZ#1649086
mariadb-10.3-module	BZ#1657053

コンポーネント	チケット
mutter	BZ#1737553
nfs-utils	BZ#1668026, BZ#1592011
nginx	BZ#1668717 , BZ#1690292
nmstate	BZ#1674456
nss	BZ#1724250 , BZ#1645153
openmpi	BZ#1717289
openscap	BZ#1642373, BZ#1618489, BZ#1646197, BZ#1718826 , BZ#1709429
openssh	BZ#1683295 , BZ#1671262, BZ#1651763, BZ#1744108 , BZ#1691045
openssl	BZ#1685470 , BZ#1681178 , BZ#1749068
oscap-anaconda-addon	BZ#1674001 , BZ#1691305
pacemaker	BZ#1715426
pcp	BZ#1685302
pcs	BZ#1619620
perl-IO-Socket-SSL	BZ#1632600
perl-Net-SSLeay	BZ#1632597
perl-Socket	BZ#1699793
php-7.2-module	BZ#1670386
php	BZ#1653109
pki-core	BZ#1695302 , BZ#1673296 , BZ#1729215
pykickstart	BZ#1637872
python-ply	BZ#1747490
python-wheel	BZ#1731526
python3	BZ#1731424

コンポーネント	チケット
qemu-kvm	BZ#1619884 , BZ#1689216 , BZ#1651474 , BZ#1740002 , BZ#1719687 , BZ#1651994
redhat-support-tool	BZ#1688274
rhel-system-roles-sap	BZ#1660832
rhel-system-roles	BZ#1691966
rng-tools	BZ#1692435
rpm	BZ#1688849
rsyslog	JIRA:RHELPLAN-10431 , BZ#1659383 , BZ#1679512 , BZ#1614181
rt-tests	BZ#1686494 , BZ#1707505 , BZ#1666351
ruby-2.6-module	BZ#1672575
s390utils	BZ#1750326
samba	BZ#1638001 , JIRA:RHELPLAN-13195
scap-security-guide	BZ#1741455 , BZ#1754919 , BZ#1750755 , BZ#1718839
scap-workbench	BZ#1640715
selinux-policy	BZ#1673269 , BZ#1671241 , BZ#1683642 , BZ#1641631 , BZ#1746398 , BZ#1673107 , BZ#1684103 , BZ#1673056
setools	BZ#1672631
setup	BZ#1663556
squashfs-tools	BZ#1716278
sssd	BZ#1448094 , BZ#1081046 , BZ#1657665 , BZ#1652562 , BZ#1669407 , BZ#1447945 , BZ#1382750 , BZ#1754871
subscription-manager	BZ#1674337
systemd	BZ#1658691 , BZ#1686892 , BZ#1640802
systemtap	BZ#1675740

コンポーネント	チケット
tpm2-abrmd-selinux	BZ#1642000
tpm2-tools	BZ#1664498
tuned	BZ#1685585
udica	BZ#1763210 , BZ#1673643
valgrind	BZ#1683715
vdo	BZ#1669534
virt-manager	BZ#1677019
virtio-win	BZ#1223668
xorg-x11-drv-qxl	BZ#1642887
xorg-x11-server	BZ#1687489, BZ#1698565
other	BZ#1640697, BZ#1623712, BZ#1745507, BZ#1659609, BZ#1697896, BZ#1732726 , JIRA:RHELPLAN-2542, JIRA:RHELPLAN-13066, JIRA:RHELPLAN-13074, BZ#1731502, BZ#1649493, BZ#1718422 , JIRA:RHELPLAN-7109, JIRA:RHELPLAN-13068, JIRA:RHELPLAN-13960, JIRA:RHELPLAN-13649, JIRA:RHELPLAN-12811, BZ#1766186 , BZ#1741531, BZ#1721683 , BZ#1690207, JIRA:RHELPLAN-1212, BZ#1559616, BZ#1699825 , JIRA:RHELPLAN-14047, BZ#1769727 , BZ#1642765, JIRA:RHELPLAN-10304, BZ#1646541, BZ#1647725, BZ#1686057 , BZ#1748980

付録B 更新履歴

0.5-0

2024年5月9日(木)、Brian Angelica (bangelic@redhat.com)

- [BZ#1690207](#) でテクニカルプレビューを更新しました。

0.4-0

2023年11月10日金曜日、Gabriela Fialová (gfialova@redhat.com)

- RHEL ドキュメントへのフィードバックの提供に関するモジュールを更新しました。

0.3-0

2022年11月15日(火) Gabriela Fialová (gfialova@redhat.com)

- 壊れたリンクを修正しました。

0.2-9

2023年10月13日(金) Gabriela Fialová (gfialova@redhat.com)

- テクノロジープレビュー [JIRA:RHELDOCS-16861](#)(コンテナ)を追加しました。

0.2-8

2023年4月27日(木) Gabriela Fialová (gfialova@redhat.com)

- 既知の問題 [JIRA:RHELPLAN-155168](#) (アイデンティティ管理)を追加

0.2-7

2022年4月29日、金曜日、Lenka Špačková (lspackova@redhat.com)

- [非推奨になった機能](#) の概要を更新。
- [BZ#1605216](#) のタイプミスを修正。
- 無効になっていたリンクを修正。

0.2-6

2021年12月23日(木) Lenka Špačková (lspackova@redhat.com)

- Soft-RoCE ドライバー `rdma_rxe` に関する情報が、テクノロジープレビューの [BZ#1605216](#) および非推奨の機能 [BZ#1878207](#) (カーネル) に追加されました。

0.2-5

2021年8月19日(木) Lucie Maňásková (lmanasko@redhat.com)

- [YUM/DNF を使用したパッケージ管理](#) を、ディストリビューションの章に追加。

0.2-4

2021年5月21日(金) Lenka Špačková (lspackova@redhat.com)

- [概要](#) で OS 変換に関する情報を更新。

0.2-3

2021年4月6日(火) Lenka Špačková (lspackova@redhat.com)

- 対応しているアーキテクチャーのリストを改善

0.2-2

2021年2月25日(木) Lenka Špačková (lspackova@redhat.com)

- CentOS Linux 名を修正。

0.2-1

2021年2月4日(木) Lucie Maňásková (lmanasko@redhat.com)

- 既知の問題(仮想化)を追加。

0.2-0

2021年1月28日(木) Lucie Maňásková (lmanasko@redhat.com)

- 新機能の章を更新。
- テクノロジープレビューの章を更新。

0.1-9

2020年12月10日(木) Lenka Špačková (lspackova@redhat.com)

- SSSDでのAD GPOの処理に関する情報が新機能(Identity Management)に追加されました。

0.1-8

2020年12月1日(火) Lucie Maňásková (lmanasko@redhat.com)

- **fapolicyd**(セキュリティー)の問題に対するバグ修正が追加されました。

0.1-7

2020年10月30日(金) Lenka Špačková (lspackova@redhat.com)

- リポジトリセクションのアプリケーションストリームの説明を更新。

0.1-6

2020年9月15日(火) Jaroslav Klech (jklech@redhat.com)

- カーネルセクションに既知の問題を追加。

0.1-5

2020年4月28日(火) Lenka Špačková (lspackova@redhat.com)

- 概要でインプレースアップグレードに関する情報を更新。

0.1-4

2020年4月9日(木) Lenka Špačková (lspackova@redhat.com)

- 既知の問題(セキュリティー)を2つ追加。

- モジュールをインストールするための統合コマンド。

0.1-3

2020年3月31日(火) Lenka Špačková (lspackova@redhat.com)

- **pcs** に関連する新機能を追加。

0.1-2

2020年3月27日(金) Lucie Maňásková (lmanasko@redhat.com)

- 誤って配置されたテクノロジープレビューの説明を正しい章に移動。

0.1-1

2020年3月20日(金) Lenka Špačková (lspackova@redhat.com)

- **jmc:rhel8** モジュールをインストールするためのコマンドを更新。

0.1-0

2020年3月12日(火) Lenka Špačková (lspackova@redhat.com)

- RHEL System Roles に関する情報を更新。

0.0-9

2020年3月6日(金) Jaroslav Klech (jklech@redhat.com)

- 外部カーネルパラメーターおよび新しいドライバーの章に対する重要な変更を提示しました。

0.0-8

2020年2月12日(水) Jaroslav Klech (jklech@redhat.com)

- アーキテクチャーおよび新機能への完全なカーネルバージョンを指定。

0.0-7

Tue Feb 04 2020, Lucie Maňásková (lmanasko@redhat.com)

- Red Hat Enterprise Linux 8.1.1 リリースノートも併せて参照してください。

0.0-6

Thu Jan 23 2020, Lucie Maňásková (lmanasko@redhat.com)

- テクノロジープレビューのセクションを更新しました。

0.0-5

Fri Dec 20 2019, Lucie Maňásková (lmanasko@redhat.com)

- バージョン 1.1.1 (システムロール) **rhel-system-roles-sap** へのリベースの記述を追加。
- **subscription-manager** により、ロール、使用方法、およびアドオンの値 (Subscription Management) が報告されるという注意書きを追加しました。
- **eBPF (Extended Berkeley Packet Filter)** に関連した注意書きを更新 (カーネル)。

0.0-4

Tue Dec 03 2019, Lucie Maňásková (Imanasko@redhat.com)

- **fadump** に関する既知の問題を追加しました (Kernel)。

0.0-3

Tue Nov 26 2019, Lucie Maňásková (Imanasko@redhat.com)

- バグ修正セクションを更新しました。
- テクノロジープレビューのセクションを更新しました。
- **irqpoll** に関連した既知の問題を追加しました (Kernel)。

0.0-2

Thu Nov 14 2019, Lucie Maňásková (Imanasko@redhat.com)

- TIPC が完全にサポートしているという注意書きを追加しました。
- x86_64 アーキテクチャーでのみ **bcc-tool** に対応するという注意書きを追加しました。
- カーネル **kpatch** のライブパッチについての情報で、概要を更新しました。
- テクノロジープレビューのセクションを更新しました。

0.0-1

Tue Nov 05 2019, Lucie Maňásková (Imanasko@redhat.com)

- Red Hat Enterprise Linux 8.1 リリースノートも併せて参照してください。

0.0-0

Wed Jul 24 2019, Lucie Maňásková (Imanasko@redhat.com)

- Red Hat Enterprise Linux 8.1 ベータリリースノートも併せて参照してください。