



Red Hat Enterprise Linux 8

8.0 リリースノート

Red Hat Enterprise Linux 8.0 リリースノート

Red Hat Enterprise Linux 8 8.0 リリースノート

Red Hat Enterprise Linux 8.0 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 8.0 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
ディストリビューション	4
ソフトウェア管理	4
シェルおよびコマンドラインツール	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	4
デスクトップ	4
インストーラーおよびイメージの作成	5
カーネル	5
ファイルシステムおよびストレージ	5
セキュリティー	5
ネットワーク	5
仮想化	6
コンパイラーおよび開発ツール	6
高可用性およびクラスター	6
関連情報	6
第2章 アーキテクチャー	8
第3章 RHEL 8 のコンテンツの配布	9
3.1. インストール	9
3.2. リポジトリ	9
3.3. アプリケーションストリーム	10
3.4. YUM/DNF を使用したパッケージ管理	10
第4章 RHEL 8.0.1 リリース	11
4.1. 新機能	11
4.2. 既知の問題	12
第5章 RHEL 8.0.0 リリース	13
5.1. 新機能	13
5.2. バグ修正	75
5.3. テクノロジープレビュー	79
5.4. 非推奨になった機能	87
5.5. 既知の問題	92
第6章 コンテナに関する主な変更点	115
第7章 国際化	116
7.1. RED HAT ENTERPRISE LINUX 8 の多言語	116
7.2. RHEL 8 における国際化の主な変更点	116
付録A コンポーネント別のチケットリスト	118
謝辞	124
付録B 更新履歴	125

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

Red Hat Enterprise Linux 8.0 は、Fedora 28 およびアップストリームカーネル 4.18 を基に、従来および新しいワードロードに対応するのに必要なツールを備え、ハイブリッドクラウドデプロイメント全体に、安定し、安全で、統一された基盤を提供します。このリリースの主な機能は以下のようになります。

ディストリビューション

- コンテンツは、**BaseOS** リポジトリおよびアプリケーションストリーム (**AppStream**) リポジトリから入手できます。
- **AppStream** リポジトリは、従来の RPM 形式の新しい拡張機能 (**モジュール**) に対応します。これにより、同じコンポーネントのメジャーバージョンを複数インストールできるようになります。

詳細は [3章 RHEL 8 のコンテンツの配布](#) を参照してください。

ソフトウェア管理

- **YUM** パッケージマネージャーは **DNF** テクノロジーに基づいており、モジュール式コンテンツ、パフォーマンス向上、ツールングとの統合に使用される、安定し、優れた API を提供します。

詳細は、「[ソフトウェア管理](#)」を参照してください。

シェルおよびコマンドラインツール

- RHEL 8 では、バージョン管理システムの **Git 2.18**、**Mercurial 4.8**、および **Subversion 1.10** が提供されます。

詳細は「[シェルおよびコマンドラインツール](#)」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

- RHEL 8 におけるデフォルトの **Python** 実装は **Python 3.6** で、**Python 2.7** への対応は限定されます。デフォルトでは、いずれのバージョンもインストールされません。
- RHEL に **Node.js** が追加されました。RHEL 7 以降、**PHP 7.2**、**Ruby 2.5**、**Perl 5.26**、**SWIG 3.0** などの **動的プログラミング言語** が利用できるようになりました。
- RHEL 8 では、データベースサーバーの **MariaDB 10.3**、**MySQL 8.0**、**PostgreSQL 10**、**PostgreSQL 9.6**、および **Redis 5** が同梱されています。
- RHEL 8 では **Apache HTTP Server 2.4** が提供され、新しい **Web サーバー (nginx 1.14)** が導入されました。
- **Squid** がバージョン 4.4 に更新され、新しい **プロキシキャッシュサーバー (Varnish Cache 6.0)** が追加されました。

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。

デスクトップ

- **GNOME Shell** がバージョン 3.28 にリベースしました。

- GNOME セッションおよび GNOME Display Manager では、デフォルトのディスプレイサーバーとして **Wayland** が使用されます。RHEL 7 でデフォルトのディスプレイサーバーとして使用されていた **X.Org** サーバーも利用できます。

詳細は「[デスクトップ](#)」を参照してください。

インストーラーおよびイメージの作成

- **Anaconda** インストーラーは、**LUKS2** のディスク暗号化を使用し、**NVDIMM** デバイスへのインストールに対応するようになりました。
- **Image Builder** ツールでは、さまざまな形式にカスタマイズしたシステムイメージを作成できます。これには、さまざまなプロバイダーのクラウドへのデプロイメント用イメージも含まれます。
- RHEL 8 の **IBM Z** では、Hardware Management Console (**HMC**) および Support Element (**SE**) を使用した、DVD からのインストールが可能になりました。

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。

カーネル

- **eBPF (extended Berkeley Packet Filtering)**機能を使用すると、ユーザー空間でカスタムプログラムを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)に接続し、データを受信して処理できます。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。
- 効率的なカーネルトレースおよび操作プログラムを作成するツールである **BPF コンパイラーコレクション (BCC)** は、**テクノロジープレビュー** として利用できます。

詳細は「[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

- 従来の **LUKS (LUKS1)** 形式が、**LUKS バージョン 2 (LUKS2)** 形式に置き換わりました。**dm-crypt** サブシステムおよび **cryptsetup** ツールでは、暗号化ボリュームのデフォルト形式として **LUKS2** が使用され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。

セキュリティ

- システム全体の **暗号化ポリシー** により、コア暗号化サブシステムが設定します。デフォルトでは、**TLS**、**IPsec**、**SSH**、**DNSSEC**、および **Kerberos** のプロトコルが対象となります。管理者は、新しい **update-crypto-policies** コマンドを使用すれば、モード (default、legacy、future、および fips) の切り替えが簡単に行えます。
- **PKCS #11** を使用した **スマートカード** および **Hardware Security Modules (HSM)** のサポート手順が、システム全体で統一されています。

詳細は「[セキュリティ](#)」を参照してください。

ネットワーク

- **iptables** に代わり、**nftables** フレームワークが、デフォルトのネットワークパケットフィルタリング機能を提供します。
- **firewalld** デーモンで、デフォルトのバックエンドとして **nftables** が使用されるようになりました。

- 複数のコンテナへのネットワーク接続を有効にする仮想ネットワークドライバー **IPVLAN** に対応するようになりました。
- eXpress Data Path (**XDP**)、XDP for Traffic Control (**tc**)、および Address Family eXpress Data Path (**AF_XDP**)、そして拡張 Berkeley Packet Filtering (**eBPF**) 機能の一部が **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。詳細は、テクニカルプレビューの「[ネットワーク](#)」を参照してください。

追加機能の新機能の「[ネットワーク](#)」を参照してください。

仮想化

- RHEL 8 の仮想マシンでより近代的な PCI Express ベースのマシントップ (Q35) に対応するようになり、新規に作成すると自動的に設定されるようになりました。これにより、さまざまな機能改善と、仮想デバイスの互換性が追加されました。
- RHEL 8 Web コンソール (**Cockpit** としても知られています) から、仮想マシンを作成して管理できるようになりました。
- **QEMU** エミュレーターに **sandboxing** 機能が追加されました。これにより、**QEMU** が実行するシステムコールに設定可能な制限が課され、仮想マシンがより安全になりました。

詳細は「[仮想化](#)」を参照してください。

コンパイラーおよび開発ツール

- バージョン 8.2 に基づいた **GCC** コンパイラーは、新しい C++ 言語の標準仕様バージョンに対応します。最適化の改善と、新しいコードの堅牢化技術、警告の改善、そして新しいハードウェア機能に対応します。
- コード生成、操作、およびデバッグに使用するさまざまなツールが、**DWARF5** デバッグの情報形式を実験的に処理できるようになりました。
- **eBPF** トレースでのカーネルへの対応が、**BCC**、**PCP**、**SystemTap** などのツールで利用できます。
- バージョン 2.28 に基づいた **glibc** ライブラリーでは、Unicode 11 への対応、新しい Linux システムコール、DNS スタブリゾルバーにおける重要な改善、セキュリティの強化、パフォーマンスの改善が追加されました。
- RHEL 8 では、OpenJDK 11、OpenJDK 8、IcedTea-Web、およびさまざまな **Java** ツール (**Ant**、**Maven**、または **Scala** など) が提供されます。

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。

高可用性およびクラスター

- **Pacemaker** クラスターリソースマネージャーがアップストリームのバージョン 2.0.0 にアップグレードしており、バグ修正および機能強化が数多く追加されました。
- RHEL 8 の **pcs** 設定システムで、Corosync 3、**knet**、およびノード名に完全に対応するようになりました。

詳細は「[高可用性およびクラスター](#)」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 テクノロジーの機能と制限](#) 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。
- 現在対応しているアップグレードパスが [Supported in-place upgrade paths for Red Hat Enterprise Linux](#) にリスト表示されます。
- 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の特典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](#)
- [Kickstart Generator](#)
- [Product Life Cycle Checker](#)
- [Red Hat Product Certificates](#)
- [Red Hat Satellite Upgrade Helper](#)
- [Red Hat CVE Checker](#)
- [JVM Options Configuration Tool](#)
- [Load Balancer Configuration Tool](#)
- [Red Hat Code Browser](#)
- [Yum Repository Configuration Helper](#)

第2章 アーキテクチャー

Red Hat Enterprise Linux 8.0 ではカーネルバージョン 4.18.0-80 が使用されており、以下のアーキテクチャーのサポートが提供されています。

- 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 RHEL 8 のコンテンツの配布

3.1. 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。

3.2. 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。

3.3. 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 と同じく、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。

3.4. 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4章 RHEL 8.0.1 リリース

4.1. 新機能

RHEL システムロールの更新

RHEL サブシステムの設定インターフェイスを提供する **rhel-system-roles** パッケージが更新されました。以下は、主な変更点です。

- **network** ロールで不明なプロファイルの処理が改善しました。**absent** への永続的な状態を設定することで、既存の NetworkManager のディスク上のプロファイル設定を削除すると、プロファイルの永続的な設定のみが削除され、現在のランタイム設定は変更しません。その結果、上記の状況では、対応するネットワークデバイスがダウンしなくなります。
- **network** ロールの VLAN インターフェイスおよび MACVLAN インターフェイスに最大転送単位 (MTU) のサイズを指定することが修正されました。これにより、**network** ロールを使用して VLAN インターフェイスおよび MACVLAN インターフェイスに MTU サイズを設定すると、以下のエラーメッセージで失敗します。

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

- **selinux** ロールおよび **timesync** ロールでは、ドキュメント化された入力変数がすべてデフォルトファイル (**defaults/main.yml**) に記載されるようになりました。これにより、ロールが対応する入力変数は、各デフォルトファイルの内容を調べれば簡単に確認できるようになりました。
- **kdump** ロールおよび **timesync** ロールは、チェックモードで失敗しないように修正されました。

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

sos-collector がバージョン 1.7 にリベース

sos-collector パッケージが、RHEL 8.0.1 でバージョン 1.7 に更新されました。以下は、主な変更点です。

- **sos-collector** は、通常の RHEL ノードと同じように、Red Hat Enterprise Linux CoreOS (RHCOS) ノードから **sosreport** を収集できるようになりました。ユーザーは、**sos-collector** の実行方法を変更する必要はありません。ノードが RHCOS または RHEL を自動的に識別します。
- RHCOS ノードから収集すると、**sos-collector** はノードに一時的なコンテナを作成し、**support-tools** コンテナを使用して **sosreport** を生成します。このコンテナは、完了後に削除されます。
- **--cluster-type=none** オプションを使用すると、ノードで実行する **sosreport** コマンドに対するクラスター関連の確認または修正をすべてスキップし、**--nodes** パラメーターで渡されるノードの静的なリストから簡単に収集できるようになります。
- Red Hat Satellite は、Satellite および Capsules から **sosreport** を収集できる、現在対応しているクラスタータイプです。

([BZ#1695764](#))

アップグレードされたコンパイラーツールセット

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 8.0.1 でアップグレードされました。

- Rust Toolset は、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 1.35 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 1.11.6 に提供します。

(BZ#1731500)

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#1713186)

4.2. 既知の問題

IPSec トンネルのパフォーマンス低下

RHEL 8.0.1 で **aes256_sha2** または **aes-gcm256** の IPSec 暗号セットを使用すると、IPSec トンネルのパフォーマンスに影響を及ぼします。特定の VPN 設定を使用すると、IPSec トンネルのパフォーマンスが 10% 低下します。このリグレッションは、Microarchitectural Data Sampling (MDS) の軽減策により発生しているわけではありません。この軽減策の有無に関わらず、このリグレッションは確認されています。

(BZ#1731362)

第5章 RHEL 8.0.0 リリース

5.1. 新機能

ここでは、Red Hat Enterprise Linux 8 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。

5.1.1. Web コンソール



注記

Web コンソールのサブスクリプションページは、新しい **subscription-manager-cockpit** パッケージで提供されるようになりました。

Web コンソールに firewall インターフェイスが追加

RHEL 8 Web コンソールの **Networking** ページに **Firewall** セクションが追加されました。このセクションでは、ファイアウォールの有効化または無効化、およびファイアウォールルールの追加、削除、変更を行います。

(BZ#1647110)

Web コンソールがデフォルトで利用可能

RHEL 8 Web コンソールのパッケージ (Cockpit と呼ばれます) は、Red Hat Enterprise Linux のデフォルトリポジトリに同梱されるようになったため、登録済みの RHEL 8 システムにすぐにインストールできます。

さらに、RHEL 8 の最小インストール以外のインストールでは、Web コンソールが自動的にインストールされ、コンソールに必要なファイアウォールポートが自動的に開くようになりました。ログイン前に、Web コンソールを有効にしたり、Web コンソールにアクセスしたりする方法を示すシステムメッセージも追加されました。

(JIRA:RHELPLAN-10355)

Web コンソール用の IdM 統合が改善

システムが Identity Management (IdM) ドメインに登録されていると、RHEL 8 Web コンソールはデフォルトで、ドメインで集中管理されている IdM リソースを使用するようになりました。これには、以下の利点があります。

- IdM ドメインの管理者は、Web コンソールを使用して、ローカルマシンを管理できます。
- コンソールの Web サーバーでは、IdM 認証局 (CA) が発行した証明書に自動的に切り替わり、ブラウザーにより許可されます。
- IdM ドメインに Kerberos チケットがあると、Web コンソールにアクセスする際にログイン認証情報を指定する必要がなくなりました。
- IdM ドメインで認識されている SSH ホストは、手動で SSH 接続を追加しなくても Web コンソールにアクセスできます。

IdM と Web コンソールの統合が適切に機能するようにするには、IdM マスターシステムの **enable-admins-sudo** オプションを付けて、**ipa-advise** ユーティリティを実行する必要があります。

(JIRA:RHELPLAN-3010)

Web コンソールにモバイルブラウザーとの互換性が追加

今回の更新で、モバイルブラウザーのバリエーションで、Web コンソールメニューおよびページを移動できるようになりました。これにより、モバイルデバイスから RHEL 8 Web コンソールを使用してシステムを管理できます。

(JIRA:RHELPLAN-10352)

Web コンソールのフロントページに、不足している更新およびサブスクリプションを表示

RHEL 8 Web コンソールが管理するシステムに、古いパッケージまたは失効したサブスクリプションがあると、システムの Web コンソールのフロントページに警告が表示されるようになりました。

(JIRA:RHELPLAN-10353)

Web コンソールが PBD 登録に対応

今回の更新で、RHEL 8 Web コンソールインターフェイスを使用して、管理システムのディスクに PBD (Policy-Based Decryption) ルールを適用できるようになりました。これは、Clevis 複号クライアントを使用して、LUKS で暗号化されたディスクパーティションの自動ロック解除など、Web コンソールのさまざまなセキュリティ管理機能を容易にします。

(JIRA:RHELPLAN-10354)

Web コンソールを使用して仮想マシンが管理可能に

RHEL 8 Web コンソールインターフェイスに、**Virtual Machines** ページを追加できるようになりました。これにより、libvirt ベースの仮想マシンを作成および管理できるようになりました。

(JIRA:RHELPLAN-2896)

5.1.2. インストーラーおよびイメージの作成

IBM Z で、SE および HMC を使用して DVD から RHEL をインストールする方法に完全対応

IBM Z ハードウェアで、**Support Element (SE)** および **Hardware Management Console (HMC)** を使用して DVD から Red Hat Enterprise Linux 8 をインストールすることに完全に対応するようになりました。この追加機能により、SE および HMC を使用した IBM Z のインストールプロセスが簡略化されます。

Binary DVD から起動すると、インストーラーにより、追加のカーネルパラメーターを入力するように求められます。DVD をインストールソースとして設定するには、**inst.repo=hmc** をカーネルパラメーターに追加します。インストーラーは、SE および HMC のファイルアクセスを有効にし、DVD から stage2 のイメージをフェッチし、ソフトウェア選択で DVD のパッケージへのアクセスを提供します。

この新機能により、外部ネットワークを設定する必要がなくなるため、インストールのオプションが増えます。

(BZ#1500792)

インストーラーが LUKS2 ディスク暗号化フォーマットに対応

Red Hat Enterprise Linux 8 インストーラーは、デフォルトで LUKS2 フォーマットを使用しますが、**Anaconda** の Custom Partitioning ウィンドウ、キックスタートの **autopart** コマンド、**logvol** コマンド、**part** コマンド、および **RAID** コマンドで新しいオプションを使用して、LUKS バージョンを選択できます。

LUKS2 は改良や機能を数多く提供します。たとえば、ディスク上のフォーマット機能を拡張し、メタデータを格納する柔軟な方法を提供します。

(BZ#1547908)

Anaconda で RHEL 8 のシステムの目的に対応

以前は、**Anaconda** から、**Subscription Manager** にシステムの目的の情報を提供していませんでした。Red Hat Enterprise Linux 8.0 では、**Anaconda** の **システムの目的** ウィンドウまたはキックスタートの **syspurpose** コマンドを使用して、システムの使用目的を設定できるようになりました。そのシステムの目的情報は、インストールが完了し、システムをサブスクライブする際に **Subscription Manager** が使用します。

(BZ#1612060)

Pykickstart で RHEL 8 のシステムの目的に対応

以前は、**pykickstart** ライブラリーにより、**Subscription Manager** にシステムの目的の情報が提供されていました。Red Hat Enterprise Linux 8.0 では、**pykickstart** が、新しい **syspurpose** コマンドを解析し、完全または一部が自動化されているインストール時に、システムの使用目的を記録します。その後、その情報は **Anaconda** に渡され、新しくインストールしたシステムに保存され、システムのサブスクライブ時に **Subscription Manager** で利用できるようになりました。

(BZ#1612061)

Anaconda で RHEL 8 の新しいカーネル起動パラメーターに対応

以前は、カーネル起動パラメーターからベースリポジトリだけを指定できました。Red Hat Enterprise Linux 8 では、新しいカーネルパラメーター **inst.addrepo=<name>,<url>** により、インストール時に追加リポジトリを指定できるようになりました。

このパラメーターでは、リポジトリの名前と、リポジトリを指定する URL の 2 つの値が必要になります。詳細は [inst.addrepo](#) を参照してください。

(BZ#1595415)

Anaconda で RHEL 8 の統合 ISO に対応

Red Hat Enterprise Linux 8.0 では、統合 ISO が、インストールソースリポジトリ BaseOS および AppStream を自動的に読み込みます。

この機能は、インストール時に読み込まれる最初のベースリポジトリに対して有効です。たとえば、リポジトリを設定せずにインストールを起動し、GUI にベースリポジトリとなる統合 ISO がある場合、または、統合 ISO を指定する **inst.repo=** オプションを使用してインストールを起動する場合などです。これにより、AppStream リポジトリは、**Installation Source** GUI ウィンドウの **Additional Repositories** セクションで有効になります。AppStream リポジトリを削除したり、その設定を変更することはできませんが、インストールソースで無効にできます。別のベースリポジトリを使用してインストールを起動し、そのあとで統合 ISO に変更した場合はこの機能が無効になります。これを行うと、ベースリポジトリが置き換えられます。ただし、AppStream リポジトリは置き換えられず、元のファイルを指定します。

(BZ#1610806)

Anaconda がキックスタートスクリプトにモジュールパッケージをインストール可能

Anaconda インストーラーは、モジュール、ストリーム、プロファイルなど、アプリケーションストリームに関連するすべての機能を処理するように拡張されました。キックスタートスクリプトは、モジュールとストリームの組み合わせを有効にし、モジュールプロファイルをインストールし、モジュールパッケージをインストールします。詳細は [高度な RHEL インストールの実行](#) を参照してください。

(JIRA:RHELPLAN-1943)

RHEL 8 インストールオプションで **nosmt** 起動オプションが利用可能に

nosmt 起動オプションは、新たにインストールした RHEL 8 システムに渡されるインストールオプションで利用できます。

(BZ#1677411)

RHEL 8 が、ローカルのハードドライブのリポジトリからのインストールに対応

以前は、ハードドライブから RHEL をインストールする場合は、インストールソースに ISO イメージが必要でした。ただし、ファイルシステムによっては、RHEL 8 イメージが大きすぎる場合があります。たとえば、FAT32 ファイルシステムでは、4 GiB 以上のファイルを保存できません。

RHEL 8 では、ローカルのハードドライブのリポジトリからインストールできます。ISO イメージの代わりにディレクトリを指定するだけで行えます。たとえば、`inst.repo=hd:<device>:<path to the repository>` です。

(BZ#1502323)

Image Builder を使用したシステムイメージのカスタマイズが RHEL 8 で利用可能に

RHEL イメージのカスタマイズには、Image Builder ツールを使用できます。Image Builder は、**lorax-composer** パッケージの AppStream で利用できます。

Image Builder を使用すると、追加パッケージを含むカスタムのシステムイメージを作成できます。Image Builder 機能は、以下から使用できます。

- Web コンソールのグラフィカルユーザーインターフェイス
- **composer-cli** ツールのコマンドラインインターフェイス

Image Builder の出力の形式には、以下のようなものがあります。

- ライブ ISO ディスクイメージ
- 仮想マシンまたは OpenStack で直接使用する qcow2 ファイル
- ファイルシステムのイメージファイル
- Azure、VMWare、および AWS のクラウドイメージ

Image Builder の詳細は [RHEL システムイメージのカスタマイズ](#) を参照してください。

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

新しいキックスタートコマンドの追加: **authselect** および **modules**

このリリースでは、次のキックスタートコマンドが追加されました。

- **authselect: authselect** コマンドを使用して、インストール中にシステム認証オプションを設定します。非推奨の **auth** または **authconfig** Kickstart コマンドの代わりに **authselect** を使用できます。詳細については、高度なインストールの実行ガイドの **authselect** セクションを参照してください。
- **module: module** コマンドを使用して、キックスタートスクリプト内でパッケージモジュールストリームを有効にします。詳細は、高度なインストールの実行ガイドの **module** セクションを参照してください。

(BZ#1972210)

5.1.3. カーネル

RHEL 8.0 のカーネルバージョン

Red Hat Enterprise Linux 8.0 には、カーネルバージョン 4.18.0-80 が同梱されています。

(BZ#1797671)

ARM 52 ビット物理アドレス指定が可能に

今回の更新で、64 ビット ARM アーキテクチャー用の 52 ビット物理アドレッシング (PA) に対応するようになりました。これにより、以前の 48 ビットの PA よりも大きなアドレス空間が提供されます。

(BZ#1643522)

IOMMU コードで RHEL 8 の 5 レベルのページテーブルに対応

Linux カーネルの I/O メモリー管理ユニット (IOMMU) コードは、Red Hat Enterprise Linux 8 で 5 レベルのページテーブルに対応するようになりました。

(BZ#1485546)

5 レベルのページングに対応

Red Hat Enterprise Linux 8 で 5 レベルのページングに対応するために、新しい **P4d_t** ソフトウェアページのテーブルタイプが Linux カーネルに追加されました。

(BZ#1485532)

メモリー管理で 5 レベルのページテーブルに対応

Red Hat Enterprise Linux 7 での既存のメモリーバスには、48/46 ビットの仮想または物理のメモリーアドレス機能があり、Linux カーネルが、4 つのレベルのページテーブルを実装して、物理アドレスへの仮想アドレスを管理します。物理バスのアドレス線は、物理メモリーの容量の上限を 64 TB に制限します。

この制限は、57/52 ビットの仮想および物理のメモリーアドレスにより、128 PiB の仮想アドレス空間と、4 PB の物理メモリーまで容量が拡張されました。

拡張アドレス範囲を使用して、Red Hat Enterprise Linux 8 のメモリー管理では、5 レベルのページテーブルの実装への対応が追加され、拡張アドレス範囲を処理します。

(BZ#1485525)

RHEL 8 で、**kernel-signing-ca.cer** が **kernel-core** に移動

Red Hat Enterprise Linux 7 では、すべてのバージョンで、公開鍵 **kernel-signing-ca.cer** が **kernel-doc** パッケージに含まれていました。ただし、Red Hat Enterprise Linux 8 では、すべてのアーキテクチャーで、**kernel-signing-ca.cer** が **kernel-core** パッケージに移動しました。

(BZ#1638465)

Spectre V2 の軽減策のデフォルトが IBRS から Retpolines へ変更

第 6 世代 Intel Core プロセッサが搭載されたシステムに対する Spectre V2 脆弱性 (CVE-2017-5715) のデフォルトの軽減策と、その類似策 [1] が、Red Hat Enterprise Linux 8 の Indirect Branch Speculation (IBRS) から Retpolines に変更になりました。Red Hat は、Linux コミュニティーで使用されるデフォルトに合わせて、損失したパフォーマンスを復元する Intel 社の推奨事項により、この変更

を実装しました。ただし、場合によっては、Retpolines を使用すると、Spectre V2 が完全に軽減されない場合があります。Intel 社の Retpoline ドキュメント [2] は、露出時のすべてのケースを説明します。本書は、攻撃のリスクが低いことも示しています。

完全な Spectre V2 軽減策が必要なユースケースでは、**spectre_v2=ibrs** フラグを追加して、カーネル起動ラインを介して IBRS を選択できます。

カーネルモジュールが Retpoline に対応するように構築されていない場合は、**/sys/devices/system/cpu/vulnerabilities/spectre_v2** ファイルで脆弱性、および **/var/log/messages** ファイルで問題のあるモジュールを特定します。詳細は [How to determine which modules are responsible for spectre_v2 returning "Vulnerable: Retpoline with unsafe module\(s\)"?](#) を参照してください。

[1]第 6 世代 Intel Core プロセッサと、その類似の派生製品は、Intel 社の Retpolines ドキュメントで Skylake-generation と呼ばれているものです。

[2] [Retpoline: A Branch Target Injection Mitigation - White Paper](#)

(BZ#1651806)

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

Red Hat Enterprise Linux 8 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タと 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。

Intel Omni-Path Architecture のインストール方法は、https://www.intel.com/content/dam/support/us/en/documents/network-and-i-o/fabric-products/Intel_OP_Software_RHEL_8_RN_K51383.pdf を参照してください。

(BZ#1683712)

NUMA が RHEL 8 でより多くのノードに対応

今回の更新で、64 ビット ARM アーキテクチャーのシステム上の Red Hat Enterprise Linux 8 では、NUMA (Non-Uniform Memory Access) ノードの数が、4 NUMA ノードから 8 NUMA ノードまで増えました。

(BZ#1550498)

RHEL 8 で IOMMU パススルーがデフォルトで有効に

入出力メモリー管理ユニット (IOMMU) パススルーが、デフォルトで有効になります。これにより、ホストに対してダイレクトメモリーアクセス (DMA) の再マッピングが無効になるため、AMD システムのパフォーマンスが改善します。今回の更新で、DMA 再マッピングがデフォルトで無効になっている Intel システムとの互換性を持たせるようになりました。ハイパーバイザーを含むカーネルコマンドラインで **iommu.passthrough=off** パラメーターまたは **iommu=nopt** パラメーターを指定してそのような動作を無効 (にし、DMA 再マッピングを有効) にできます。

(BZ#1658391)

RHEL8 カーネルが、5 レベルのページテーブルに対応

Red Hat Enterprise Linux カーネルは、最大 5 レベルのページテーブルが搭載される将来の Intel プロセッサに完全に対応するようになりました。これにより、プロセッサは、最大 4PB の物理メモリーと 128PB の仮想アドレス領域に対応できるようになります。大容量のメモリーを使用するアプリ

ケーションは、4レベルのページテーブルに制約を課さずに、システムが提供するメモリーをできるだけ多く使用できるようになりました。

(BZ#1623590)

RHEL8 カーネルは、将来の Intel CPU 用に拡張 IBRS に対応

Red Hat Enterprise Linux カーネルは、Spectre V2 の脆弱性を軽減するために、強化された IBRS (Indirect Branch Speculation) 機能の使用に対応するようになりました。これを有効にすると、IBRS は Spectre V2 を軽減する点で、Retpolines (デフォルト) よりも優秀で、Intel Control-flow Enforcement テクノロジーに干渉しません。これにより、今後の Intel CPU では、Spectre V2 の軽減策を有効にすることによるパフォーマンス低下が少なくなります。

(BZ#1614144)

eBPF ベースのプログラムおよびマップの調査および操作に使用する **bpftool** が追加

eBPF (extended Berkeley Packet Filtering) に基づくプログラムおよびマップの検査と簡単な操作を行う **bpftool** ユーティリティーが Linux カーネルに追加されました。**bpftool** はカーネルソースツリーの一部で、**bpftool** パッケージにより提供されます。これは、カーネルパッケージのサブパッケージとして含まれています。

(BZ#1559607)

kernel-rt ソースが更新

kernel-rt ソースが更新され、最新の RHEL カーネルソースツリーを使用するようになりました。最新のカーネルソースツリーは、現在、アップストリーム v4.18 のリアルタイムパッチセットを使用しています。これにより、以前のバージョンにバグ修正および機能強化が数多く追加されました。

(BZ#1592977)

5.1.4. ソフトウェア管理

YUM パフォーマンスの改善、およびモジュール式コンテンツへの対応

Red Hat Enterprise Linux 8 へのソフトウェアのインストールは、DNF テクノロジーに基づいた、新しいバージョンの YUM ツール (YUM v4) で行われます。

YUM v4 には、RHEL 7 で使用されていた、以前のバージョンの YUM v3 に対して、以下の利点が追加されました。

- パフォーマンスの向上
- モジュール式コンテンツへの対応
- ツーリングと統合するために適切に設計され、安定した API

新しい YUM v4 ツールと、以前のバージョンである RHEL 7 の YUM v3 ツールとの間における相違点の詳細は、[Changes in DNF CLI compared to YUM](#) を参照してください。

コマンドラインから使用したり、設定ファイルを編集または作成する場合、YUM v4 は YUM v3 と互換性があります。

ソフトウェアをインストールするには、RHEL 7 と同じ **yum** コマンドとオプションを使用できます。

選択した yum プラグインおよびユーティリティーは、新しい DNF バックエンドに移植されており、RHEL 7 と同じ名前です。このパッケージは互換性を持ったシンボリックリンクを提供するため、バイナリー、設定ファイル、ディレクトリーは通常の場所で確認できます。

YUM v3 が提供するレガシーの Python API は利用できなくなりました。YUM v4 (DNF Python API) が提供する安定し、完全に対応する新しい API に、使用しているプラグインおよびスクリプトを移行することが推奨されます。DNF Python API は、[DNF API Reference](#) から入手できます。

Libdnf および Hawkey API (C および Python の両方) は安定していないため、Red Hat Enterprise Linux 8 のライフサイクル期間中に変更する可能性があります。

YUM パッケージの変更点および利用可能なツールの詳細は、[RHEL 8 の導入における検討事項](#) を参照してください。

YUM v3 機能の一部の挙動が、YUM v4 と異なる可能性があります。この変更がワークフローに影響を及ぼす場合は、[カスタマーポータルでサポートケースを作成および管理する](#) に従って、Red Hat サポートケースを作成してください。

(BZ#1581198)

RHEL 8 における主な RPM 機能

Red Hat Enterprise Linux 8 には RPM 4.14 が同梱されています。このバージョンでは、RHEL 7 で利用できた RPM 4.11 に対する機能拡張が数多く追加されました。以下は、主な変更点です。

- **debuginfo** パッケージの複数バージョンを同時にインストール可能
- 弱い依存関係への対応
- 豊富なブール型依存関係への対応。
- 4 GB を超えるファイルのパッケージ化への対応。
- ファイルトリガーへの対応

以下は、主な変更点です。

- 厳密なスペックパーサー
- 非冗長モードで出力を確認する簡易署名
- マクロの追加および廃止

(BZ#1581990)

RPM が、インストールを開始する前にパッケージ全体のコンテンツを検証

Red Hat Enterprise Linux 7 では、RPM ユーティリティーにより、解凍時に各ファイルのペイロードの内容が検証されていました。以下の理由により、この方法は十分ではありません。

- ペイロードが破損していても、スクリプトアクションを実行した後にしか通知されず、元に戻せません。
- ペイロードが破損していると、以前のバージョンのファイルを一部置き換えた後でパッケージのアップグレードが中止するため、インストールが正常に機能しません。
- 個々のファイルのハッシュは、圧縮されていないデータに対して実行するため、RPM は、デコンプレッサーの脆弱性に対して脆弱です。

Red Hat Enterprise Linux 8 では、インストール前に、利用可能な最善のハッシュを使用して、パッケージ全体が個別の手順で検証されます。

Red Hat Enterprise Linux 8 に構築したパッケージでは、圧縮したペイロードに新しい **SHA-256** ハッシュが使用されています。署名付きパッケージでは、署名によりペイロードハッシュの保護が強化されるため、パッケージヘッダーでは、署名やその他のハッシュを壊さずにペイロードハッシュを変更することはできません。以前のパッケージでは、設定で無効にされていない限り、ヘッダーとペイロードの **MD5** ハッシュを使用します。

さらに、`%_pkgverify_level` マクロを使用して、インストール前に署名検証を強制することを有効にしたり、ペイロード検証を完全に無効にしたりできます。さらに、`%_pkgverify_flags` マクロを使用して、どのハッシュおよび署名を許可するかを制限できます。たとえば、弱い **MD5** ハッシュの使用を無効にすることもできますが、これにより、古いパッケージとの互換性がなくなります。

(JIRA:RHELPLAN-10596)

5.1.5. インフラストラクチャーサービス

RHEL 8 で推奨される Tuned プロファイルの主な変更点

今回の更新で、(`tuned-adm recommend` コマンドにより報告される) 推奨される Tuned プロファイルが、次のルールに基づいて選択されるようになりました。一致する最初の規則が有効になります。

- (`syspurpose show` コマンドにより報告される) **syspurpose** ロールには **atomic** が含まれ、以下のようになります。
 - Tuned がベアメタルで実行している場合は、**atomic-host** プロファイルが選択されます。
 - Tuned が仮想マシンで実行している場合は、**atomic-guest** プロファイルが選択されます。
- Tuned が仮想マシンで実行している場合は、**virtual-guest** プロファイルが選択されます。
- **syspurpose** ロールに **desktop** または **workstation** が含まれ、シャーシタイプ (`dmidecode` が報告) が **Notebook**、**Laptop**、または **Portable** の場合は、**balanced** プロファイルが選択されます。
- 上記のどのルールにも一致しない場合は **throughput-performance** プロファイルが選択されません。

(BZ#1565598)

named が作業ディレクトリーに生成するファイルを書き込み可能に

以前は、**named** デーモンが、作業ディレクトリーにデータを保存していましたが、Red Hat Enterprise Linux では、このディレクトリーは読み取り専用でした。今回の更新で、選択したファイルのパスが、書き込みが許可されているサブディレクトリーに変更になりました。これにより、Unix および SELinux のパーミッションでは、デフォルトのディレクトリーへの書き込みが許可されています。ただし、ディレクトリーに配布されているファイルは引き続き **named** に対して読み取り専用になります。

(BZ#1588592)

Geolite Databases が Geolite2 Databases に置き換え

Red Hat Enterprise Linux 7 で提供していた Geolite Databases が、Red Hat Enterprise Linux 8 で Geolite2 Databases に置き換えられました。

Geolite Databases は **GeoIP** パッケージで提供されていました。アップストリームでは、従来のデータベースが同梱されていたこのパッケージに対応しなくなりました。

Geolite2 Databases は、複数のパッケージで提供されます。**libmaxminddb** パッケージには、ライブラリーと、アドレスの手動検索を可能にする **mmdblookup** コマンドラインツールが同梱されています。従来の **GeoIP** パッケージの **geoipupdate** バイナリーは **geoipupdate** パッケージで提供されているため、従来のデータベースと新しい Geolite2 データベースの両方をダウンロードできるようになりました。

(JIRA:RHELPLAN-6746)

journald が CUPS ログに対応

RHEL 8 では、RHEL 7 で使用されていた **/var/log/cups** ディレクトリーの特定ファイルに、CUPS ログが保存されなくなりました。RHEL 8 では、どのタイプの CUPS ログも、他のプログラムとともに **systemd** の **journald** デーモンが集中管理します。CUPS ログにアクセスするには、**journalctl -u cups** コマンドを使用してください。詳細は、[systemd ジャーナルの CUPS ログへのアクセス](#) を参照してください。

(JIRA:RHELPLAN-12764)

RHEL 8 における主な BIND 機能

RHEL 8 には、バージョン 9.11 に BIND (Berkeley Internet Name Domain) が含まれています。このバージョンの DNS サーバーには、バージョン 9.10 と比較して、複数の新機能と機能変更が追加されています。

新機能:

- セカンダリーサーバー **Catalog Zones** をプロビジョニングする新しい方法が追加されました。
- Domain Name System Cookies は、**named** サービスおよび **dig** ユーティリティーにより送信されるようになりました。
- **Response Rate Limiting** 機能は、DNS 増幅攻撃の軽減を支援できます。
- RPZ の (response-policy zone) のパフォーマンスが改善しました。
- **map** と呼ばれる新しいゾーンファイルが追加されています。このフォーマットに保存されるゾーンファイルは、メモリーに直接マッピングされます。これにより、ゾーンが読み込む速度が大幅に改善します。
- DNS データの検索、および DNS Security Extensions (DNSSEC) 検証を実行する **dig** のようなセマンティクスを使用する、**delv** ツール (ドメインエンティティー検索および検証) が追加されています。
- 新しい **mdig** コマンドが利用できるようになりました。このバージョンのコマンドは、**dig** コマンドの中で、1つのクエリーを送って、応答を待ってから次のクエリーを送る前に、パイプラインで複数のクエリーを送って応答を待ちます。
- 再帰リゾルバーのパフォーマンスを改善する新しい **prefetch** オプションが追加されました。
- ビュー間でゾーンデータを共有できる、新しい **in-view** ゾーンオプションが追加されました。このオプションが追加されると、新しいビューは、メモリーに複数のコピーを保存せずに、同じゾーンを確実に保存できます。
- ゾーンに最大の TTL を強制する新しい **max-zone-ttl** オプションが追加されました。高い TTL を含むゾーンを読み込むと、読み込みに失敗します。動的 DNS (DDNS) により高い TTL を設定することは可能ですが、TTL は切り捨てられます。

- 新しいクォータは、再帰リゾルバーが、サービス拒否攻撃が発生している権威サーバーに送信するクエリーを制限するために追加されました。
- **nslookup** ユーティリティーは、デフォルトで IPv6 アドレスと IPv4 アドレスの両方を検索するようになりました。
- **named** サービスは、起動する前に、その他のネームサーバープロセスが実行しているかどうかを確認します。
- 署名付きゾーンを読み込むと、**named** が、Resource Record Signature (RSIG) の開始時間が将来時にあるかどうかを確認し、存在する場合はすぐに RRSIG を再生成するようになりました。
- ゾーン転送は、ネットワーク使用量を低減するメッセージ圧縮を改善するために、より小さいサイズのメッセージを使用するようになりました。

機能変更:

- 静的チャンネルに対するバージョン **3 の XML** スキーマでは、高速解析を行うために、HTTP インターフェイスにより、新しい統計および平坦化した XML ツリーが提供されます。古いバージョンの **2 XML** スキーマに対応しなくなりました。
- **named** サービスは、デフォルトで IPv6 および IPv4 のインターフェイスでリッスンするようになりました。
- **named** サービスは GeolIP に対応しなくなりました。クエリー送信者の推定位置で定義される Access control lists (ACL) は利用できなくなりました。

(JIRA:RHELPLAN-1820)

5.1.6. シェルおよびコマンドラインツール

nobody ユーザーが nfsnobody に置き換え

Red Hat Enterprise Linux 7 では、以下のように設定されていました。

- ID が 99 の **nobody** のユーザーおよびグループのペア。
- ID が 65534 の **nfsnobody** ユーザーとグループのペア。この ID は、デフォルトのカーネルオーバーフロー ID です。

これはいずれも、Red Hat Enterprise Linux 8 では、**nobody** ユーザーおよびグループのペア (ID 65534) に統合されました。新規インストールでは、**nfsnobody** ペアが作成されなくなりました。

この変更により、**nobody** が所有し、NFS とは無関係のファイルに対する曖昧さが軽減されます。

(BZ#1591969)

RHEL 8 のバージョン管理システム

RHEL 8 は、次のバージョン管理システムを提供します。

- **Git 2.18** は、分散アーキテクチャーを持つ分散型リビジョン管理システムです。
- **Mercurial 4.8** は、大規模プロジェクトを効率的に処理するために設計された、軽量の分散バージョン管理システムです。
- **Subversion 1.10** は、集中型管理システムです。

RHEL 7 で利用できた Concurrent Versions System (CVS) および Revision Control System (RCS) は、RHEL 8 では配布されていません。

(BZ#1693775)

Subversion 1.10 への主な変更点

Subversion 1.10 には、RHEL 7 で配布されたバージョン 1.7 以降に追加された新機能と、次の互換性の変更が含まれています。

- 言語バインディングに対応するのに使用される **Subversion** ライブラリーにおける非互換性のため、**Subversion 1.10** の **Python 3** バインディングは利用できません。したがって、**Subversion** に **Python** バインディングを必要とするアプリケーションには対応していません。
- **Berkeley DB** に基づくリポジトリには対応しなくなりました。移行する前に、**svnadmin dump** コマンドを使用して、**Subversion 1.7** で作成したリポジトリをバックアップします。RHEL 8 をインストールした後、**svnadmin load** コマンドを使用してリポジトリを復元します。
- RHEL 7 の **Subversion 1.7** クライアントがチェックアウトした既存のワーキングコピーは、**Subversion 1.10** で使用する前に新しい形式にアップグレードする必要があります。RHEL 8 をインストールしたら、各ワーキングコピーで **svn upgrade** コマンドを実行します。
- **https://** を使用してリポジトリにアクセスするスマートカード認証には対応しなくなりました。

(BZ#1571415)

dstat への主な変更点

RHEL 8 には、新しいバージョンの **dstat** ツールが同梱されています。このツールは、Performance Co-Pilot (PCP) ツールキットに含まれるようになりました。**/usr/bin/dstat** ファイルおよび **dstat** パッケージの名前は、**pcp-system-tools** パッケージで提供されるようになりました。

新しいバージョンの **dstat** では、RHEL 7 で利用可能な **dstat** に、以下の機能強化が追加されました。

- **python3** への対応
- 履歴分析
- リモートホスト分析
- 設定ファイルの構文
- 新パフォーマンスのメトリクス

(BZ#1684947)

5.1.7. 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

Python 3 が RHEL 8 におけるデフォルトの Python 実装に

Red Hat Enterprise Linux 8 には、**Python 3.6** が同梱されています。このパッケージは、デフォルトでインストールされていない可能性があります。**Python 3.6** をインストールする場合は、**yum install python3** コマンドを使用します。

Python 2.7 は、**python2** パッケージで入手できます。ただし、**Python 3** への移行をより円滑に進められるように、**Python 2** のライフサイクルは短くなっています。

デフォルトの **python** パッケージまたはバージョンを指定しない **/usr/bin/python** 実行ファイルは、いずれも RHEL 8 では配布されません。**python3** または **python2** を直接使用することが推奨されます。もしくは、管理者が、**alternatives** コマンドを使用して、バージョン管理外の **python** コマンドを設定できます。

詳細は、[Python の概要](#) を参照してください。

(BZ#1580387)

Python スクリプトは、RPM ビルド時にインタープリターディレクティブでメジャーバージョンを指定する必要があります

RHEL 8 では、実行可能な Python スクリプトは、少なくともメジャー Python バージョンを明示的に指定するインタープリターディレクティブ (hashbang) を使用することが期待されています。

BRP (buildroot policy) スクリプト **/usr/lib/rpm/redhat/brp-mangle-shebangs** は、RPM パッケージを構築する際に自動的に実行します。このスクリプトは、すべての実行可能ファイルのインタープリターディレクティブを修正しようとします。スクリプトが、Python のメジャーバージョンを指定していないあいまいな Python インタープリターディレクティブを検出すると、エラーが生成され、RPM ビルドが失敗します。このようなあいまいなインタープリターディレクティブの例には、次のようなものがあります。

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

RPM ビルド時にこれらのビルドエラーを引き起こす Python スクリプトのインタープリターディレクティブを変更するには、**platform-python-devel** パッケージの **pathfix.py** スクリプトを使用します。

```
pathfix.py -pn -i %[_python3] PATH ...
```

複数の **PATH** を指定できます。**PATH** がディレクトリーの場合、**pathfix.py** は Python スクリプトを再帰的にスキャンして、**^[a-zA-Z0-9_]+\.\$** パターンに一致するものを探します。これは、曖昧な hashbang があるものだけではありません。**%prep** セクション、または **%install** セクションに、**pathfix.py** を実行するコマンドを追加します。

詳細は、[Python スクリプトでのインタープリターディレクティブの処理](#) を参照してください。

(BZ#1583620)

PHP への主な変更点

Red Hat Enterprise Linux 8 には **PHP 7.2** が同梱されています。このバージョンには、RHEL 7 で利用できた **PHP 5.4** に対する重要な変更が追加されています。

- **PHP** はデフォルトで FastCGI Process Manager (FPM) を使用します (スレッド化された **httpd** で安全に使用できます)。
- **php_value** 変数と **php-flag** 変数が **httpd** 設定ファイルで使用されなくなり、代わりにプール設定の **/etc/php-fpm.d/*.conf** で設定する必要があります。
- **PHP** スクリプトのエラーと警告のログは、**/var/log/httpd/error.log** ではなく **/var/log/php-fpm/www-error.log** ファイルに記録されます。

- PHP の **max_execution_time** 設定変数を変更する時は、変更した値に合わせて **httpd ProxyTimeout** 設定を増やす必要があります。
- **PHP** スクリプトを実行するユーザーが、FPM プール設定 (**apache** ユーザーがデフォルトとなる **/etc/php-fpm.d/www.conf** ファイル) に設定されるようになりました。
- 設定を変更した場合、または新しい拡張機能をインストールした場合は、**php-fpm** サービスを再起動する必要があります。
- **zip** 拡張が、**php-common** から、別のパッケージ **php-pecl-zip** に移動しました。

以下の拡張機能が削除されました。

- **aspell**
- **mysql** (拡張機能の **mysqli** および **pdo_mysql** は、**php-mysqlnd** パッケージで引き続き利用できます)
- **memcache**

(BZ#1580430, [BZ#1691688](#))

Ruby への主な変更点

RHEL 8 では、RHEL 7 に同梱されていた **Ruby 2.5** に新機能および機能強化を追加した **Ruby 2.0.0** が提供されます。以下は、主な変更点です。

- インクリメンタルガベージコレクターが追加されました。
- **Refinements** 構文が追加されました。
- シンボルは、ガベージコレクションが行われるようになりました。
- 安全レベルの **\$SAFE=2** および **\$SAFE=3** が廃止されました。
- **Fixnum** クラスと **Bignum** クラスが、**Integer** クラスに統合されました。
- **Hash** クラスの最適化、インスタンス変数へのアクセスの向上、ならびに **Mutex** クラスの小型化および高速化により、パフォーマンスが向上しました。
- 古い API が非推奨になりました。
- **RubyGems**、**Rake**、**RDoc**、**Psych**、**Minitest**、**test-unit** などのバンドルされたライブラリーが更新されました。
- **Ruby** とともに配布されていた **mathn**、**DL**、**ext/tk**、**XMLRPC** などのライブラリーは非推奨になり、同梱されなくなりました。
- **SemVer** バージョン管理スキームが、**Ruby** バージョン管理に使用されるようになりました。

(BZ#1648843)

Perl への主な変更点

RHEL 8 で提供される **Perl 5.26** では、RHEL 7 で提供されていたバージョンに以下のような変更が追加されました。

- **Unicode 9.0** に対応するようになりました。

- 新しい **SystemTap** のプローブ **op-entry**、**loading-file**、および **loaded-file** が提供されるようになりました。
- パフォーマンスを向上させるために、スカラーの割り当て時に、コピーオンライトメカニズムが使用されます。
- IPv4 ソケットおよび IPv6 ソケットを透過的に処理するために **IO::Socket::IP** モジュールが追加されました。
- 構造化された方法で **perl -V** データにアクセスするために、**Config::Perl::V** モジュールが追加されました。
- Comprehensive Perl Archive Network (CPAN) リポジトリからモジュールを取得、抽出、ビルド、およびインストールする **cpanm** ユーティリティを同梱する **perl-App-cpanminus** パッケージが追加されました。
- セキュリティ上の理由により、**@INC** モジュールの検索パスから、現在のディレクトリ (.) が削除されました。
- 上記の動作上の変更によりファイルの読み込みに失敗した時に、**do** ステートメントが非推奨の警告を返すようになりました。
- **do subroutine(LIST)** 呼び出しに対応しなくなり、構文エラーが発生するようになりました。
- ハッシュがデフォルトでランダム化されるようになりました。ハッシュから鍵と値が返される順序は、**perl** の実行ごとに変わります。ランダム化を無効にするには、**PERL_PERTURB_KEYS** 環境変数を **0** に設定します。
- 正規表現のパターンで、エスケープされていないリテラルの { 文字が使用できなくなりました。
- **\$_** 変数に対する語彙的なスコープへの対応が削除されました。
- 配列またはハッシュに **defined** 演算子を使用すると、致命的なエラーが発生します。
- **UNIVERSAL** モジュールから関数をインポートすると、致命的なエラーが発生します。
- **find2perl** ツール、**s2p** ツール、**a2p**、**c2ph** ツール、および **pstruct** ツールが削除されました。
- **\${^ENCODING}** 機能が削除されました。**encoding** プラグマのデフォルトモードに対応しなくなりました。**UTF-8** 以外のエンコーディングでソースコードを記述する場合は、エンコーディングの **Filter** オプションを使用します。
- アップストリームに合わせて、**perl** パッケージが変更になりました。**perl** パッケージはコアモジュール群をインストールし、**/usr/bin/perl** インタープリターは **perl-interpreter** パッケージで提供されます。以前のリリースでは、**perl** パッケージに最小限のインタープリターだけが同梱され、**perl-core** パッケージにインタープリターとコアモジュールの両方が同梱されていました。
- Perl モジュールの **IO::Socket::SSL** は、**./certs/my-ca.pem** ファイルまたは **./ca** ディレクトリから認証局の証明書、**./certs/server-key.pem** ファイルからサーバーの秘密鍵、**./certs/server-cert.pem** ファイルからサーバーの証明書、**./certs/client-key.pem** ファイルからクライアントの秘密鍵、**./certs/client-cert.pem** ファイルからクライアント証明書を読み込まなくなりました。代わりにファイルのパスを明示的に指定します。

Node.js が RHEL に新登場

JavaScript プログラミング言語で高速でスケーラブルなネットワークアプリケーションを構築するソフトウェア開発プラットフォームである **Node.js** が RHEL で初めて提供されます。以前は、Software Collection からしか入手できませんでした。RHEL 8 では **Node.js 10** が提供されます。

(BZ#1622118)

SWIG への主な変更点

RHEL 8 には、SWIG (Simplified Wrapper and Interface Generator) バージョン 3.0 が含まれています。RHEL 7 に同梱されていたバージョン 2.0 に新機能、機能拡張、およびバグ修正が数多く追加されました。特に注目すべきは、C++11 標準仕様への対応が実装されたことです。**SWIG** は、**Go 1.6**、**PHP 7**、**Octave 4.2**、および **Python 3.5** に対応するようになりました。

(BZ#1660051)

Apache httpd への主な変更点

RHEL 8 には、Apache HTTP Server 2.4.37 が同梱されています。このバージョンには、RHEL 7 で利用できた **httpd** に対する次の変更が含まれています。

- **httpd** モジュールに同梱される **mod_http2** パッケージにより、HTTP/2 に対応するようになりました。
- (**Let's Encrypt** などの証明書プロバイダーで使用するため) 自動証明書管理環境 (ACME) プロトコルを使用した、TLS 証明書の自動プロビジョニングおよび更新が、**mod_md** パッケージで対応するようになりました。
- Apache HTTP Server が、**PKCS#11** モジュールを利用して、ハードウェアのセキュリティートークンから、TLS 証明書および秘密鍵を直接読み込むようになりました。これにより、**mod_ssl** 設定で、**PKCS#11** の URL を使用して、**SSLCertificateKeyFile** ディレクティブおよび **SSLCertificateFile** ディレクティブに、TLS 秘密鍵と、必要に応じて TLS 証明書をそれぞれ指定できるようになりました。
- Apache HTTP Server にデフォルトで設定されているマルチプロセッシングモジュール (MPM) が、(**prefork** として知られる) マルチプロセスのフォークモデルから、高パフォーマンスのマルチスレッドモデル **event** へ変更しました。スレッドセーフではないサードパーティーのモジュールは、交換または削除する必要があります。設定した MPM を変更する場合は、**/etc/httpd/conf.modules.d/00-mpm.conf** ファイルを編集します。詳細は、man ページの **httpd.conf(5)** を参照してください。

httpd の変更とその使用方法の詳細は、[Apache HTTP Web サーバーの設定](#) を参照してください。

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

nginx Web サーバーが RHEL に新登場

RHEL 8 では、HTTP などのプロトコルに対応する Web サーバーおよびプロキシサーバー **nginx 1.14** が導入され、同時実行性とパフォーマンスが高くなり、メモリー使用量が少なくなりました。**nginx** は、以前は、Software Collection からしか入手できませんでした。

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ートークンから直接 TLS 秘密鍵を読み込むようになりました。これにより、**nginx** 設定で、**PKCS#11** の URL を使用して TLS 秘密鍵を **ssl_certificate_key** ディレクティブに指定できるようになりました。

(BZ#1545526)

RHEL 8 のデータベースサーバー

RHEL 8 は、次のデータベースサーバーを提供します。

- **MySQL 8.0**。マルチユーザー、マルチスレッドの SQL データベースサーバーです。**MySQL** サーバーデーモンである **mysqld** と、多数のクライアントプログラムで設定されます。
- **MariaDB 10.3**。マルチユーザー、マルチスレッドの SQL データベースサーバーです。あらゆる実用的な目的に対応するために、**MariaDB** には、**MySQL** とバイナリー互換性があります。
- **PostgreSQL 10** および **PostgreSQL 9.6**。高度なオブジェクトリレーショナルデータベース管理システム (DBMS) です。
- **Redis 5**。高度な鍵と値のストアです。鍵には、文字列、ハッシュ、リスト、セット、およびソートセットを含めることができるため、データ構造サーバーと呼ばれています。RHEL で、**Redis** が初めて提供されます。

NoSQL の **MongoDB** データベースサーバーは、Server Side Public License (SSPL) を使用するため、RHEL 8.0 には同梱されていません。

(BZ#1647908)

MySQL 8.0 への主な変更点

RHEL 8 に同梱される **MySQL 8.0** には、たとえば次のような機能強化が含まれます。

- **MySQL** に、データベースオブジェクトに関する情報を格納するトランザクショナルデータディクショナリーが組み込まれました。
- **MySQL** がロールに対応するようになりました。ロールは特権の集まりです。
- デフォルトの文字セットが、**latin1** から **utf8mb4** に変更しました。
- 非再帰的および再帰的な Common Table Expression に対応するようになりました。
- **MySQL** が、クエリーで関連する行の集合に対して、行ごとに計算を実行する window 関数に対応するようになりました。
- locking read ステートメントにより、**InnoDB** で、**NOWAIT** オプションおよび **SKIP LOCKED** オプションに対応するようになりました。
- GIS 関連の関数が改善しました。
- JSON 関数が強化されました。
- 新しい **mariadb-connector-c** パッケージは、**MySQL** と **MariaDB** に共通のクライアントライブラリーを提供します。このライブラリーは、データベースサーバーの **MySQL** および **MariaDB** の全バージョンで使用できます。その結果、RHEL 8 に同梱される **MySQL** サーバーおよび **MariaDB** サーバーのいずれかに構築されるアプリケーションの1つに接続できます。

さらに、RHEL 8 に同梱されている **MySQL 8.0** サーバーでは、デフォルトの認証プラグインに **mysql_native_password** を使用するように設定されています。RHEL 8 のクライアントツールおよびライブラリーは、アップストリームの **MySQL 8.0** バージョンでデフォルトで使用されている **caching_sha2_password** メソッドと互換性がないためです。

デフォルトの認証プラグインを **caching_sha2_password** に変更するには、**/etc/my.cnf.d/mysql-default-authentication-plugin.cnf** ファイルを次のように変更します。

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

[MySQL の使用](#) も参照してください。

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

MariaDB 10.3 への主な変更点

MariaDB 10.3 では、RHEL 7 に同梱されていたバージョン 5.5 に新機能が数多く追加されました。以下は、主な変更点です。

- 共通テーブル式 (CTE)
- システムのバージョンが管理されたテーブル
- **FOR** ループ
- 非表示の列
- シーケンス
- **InnoDB** のインスタント **ADD COLUMN**
- ストレージエンジンに依存しないカラム圧縮
- 並列レプリケーション
- マルチソースのレプリケーション

さらに、新しい **mariadb-connector-c** パッケージは、**MySQL** と **MariaDB** に共通のクライアントライブラリーを提供します。このライブラリーは、データベースサーバーの **MySQL** および **MariaDB** の全バージョンで使用できます。その結果、RHEL 8 に同梱される **MySQL** サーバーおよび **MariaDB** サーバーのいずれかに構築されるアプリケーションの1つに接続できます。

その他の主な変更点は次の通りです。

- 同期マルチクラスターでもある **MariaDB Galera Cluster** が、**MariaDB** の標準に含まれるようになりました。
- **InnoDB** は、**XtraDB** の代わりに、デフォルトのストレージエンジンとして使用されます。
- **mariadb-bench** サブパッケージが削除されました。
- プラグインの成熟度に対するデフォルトの許可レベルは、サーバーの成熟度よりレベルが1つ低くなるように変更になりました。その結果、以前は動作していた、成熟度レベルが低いプラグインが、読み込まれなくなりました。

[MariaDB の使用](#) も参照してください。

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

PostgreSQL への主な変更点

RHEL 8.0 は、**postgresql** モジュールの2つのストリームで、**PostgreSQL** データベースサーバーのバージョンを2つ (**PostgreSQL 10** (デフォルトストリーム) および **PostgreSQL 9.6**) 提供します。RHEL 7 には **PostgreSQL** バージョン 9.2 が含まれます。

PostgreSQL 9.6 への主な変更点。以下は例になります。

- 一連の動作の並列実行 - **scan**、**join**、および **aggregate**
- 同期レプリケーションの機能強化
- フレーズを検索できるように、フルテキスト検索が改善
- **postgres_fdw** データ連携ドライバーが、リモートの **join**、**sort**、**UPDATE**、および **DELETE** の操作に対応
- (特に、マルチ CPU ソケットサーバーのスケーラビリティに関する) 重要なパフォーマンスの向上

PostgreSQL 10 への主な機能拡張。以下は例になります。

- **publish** キーワードおよび **subscribe** キーワードを使用した論理レプリケーション
- **SCRAM-SHA-256** メカニズムを基にした強力なパスワード認証
- 宣言型テーブルのパーティション
- 改善されたクエリーの並列処理
- 重要な一般的なパフォーマンスの向上
- 改善された監視および制御

[PostgreSQL の使用](#) も参照してください。

(BZ#1660041)

Squid への主な変更点

RHEL 8.0 は、Web クライアント、対応する FTP、Gopher、および HTTP のデータオブジェクト用の高パフォーマンスのプロキシキャッシュサーバーである **Squid 4.4** で配布します。このリリースは、RHEL 7 で利用可能なバージョン 3.5 に新しい機能、機能強化、バグ修正を多数提供します。

以下は、主な変更点です。

- ヘルパーのキューサイズが変更可能
- ヘルパー同時実行チャンネルへの変更
- ヘルパーバイナリーへの変更
- Internet Content Adaptation Protocol (ICAP) の保護
- 対称型マルチプロセッシング (SMP) への対応が改善
- プロセス管理が改善
- SSL への対応が削除
- Edge Side Includes (ESI) カスタムパーサーが削除
- 複数の設定変更

(BZ#1656871)

Varnish Cache が RHEL に新登場

高パフォーマンスの HTTP 逆ポリシーである **Varnish Cache** が、RHEL で初めて提供されました。以前は、Software Collection からしか入手できませんでした。**Varnish Cache** は、将来の同等の要求で応答時間およびネットワークの帯域幅を削減するのに使用されるメモリーに、ファイルまたはファイルの断片を保存します。RHEL 8.0 は、**Varnish Cache 6.0** で配布されています。

(BZ#1633338)

5.1.8. デスクトップ

RHEL 8 の GNOME シェル (バージョン 3.28)

Red Hat Enterprise Linux (RHEL) 8 では、GNOME シェルのバージョン 3.28 が利用できます。以下は、主な機能強化です。

- GNOME Boxes の新機能
- 新しいオンスクリーンキーボード
- デバイスへの対応が拡張 (最も大きな統合は Thunderbolt 3 インターフェイス)
- GNOME ソフトウェア、dconf-editor、および GNOME ターミナルの改善

(BZ#1649404)

Wayland がデフォルトのディスプレイサーバーに

Red Hat Enterprise Linux 8 では、GNOME セッションおよび GNOME Display Manager (GDM) のデフォルトのディスプレイサーバーでは、以前の RHEL メジャーバージョンで使用されていた **X.org** に代わり、**Wayland** が使用されます。

X.org と比べると、**Wayland** では、複数の利点および改善点が提供されます。以下に例を示します。

- より強力なセキュリティーモデル
- 改善されたマルチモニター処理
- 改善されたユーザーインターフェイス (UI) スケーリング
- デスクトップでウィンドウ処理を直接制御

以下の機能は、現在利用できない、または期待通りに機能しない状態です。

- **Wayland** では、マルチ GPU の設定が利用できません。
- **Wayland** では、**NVIDIA** バイナリードライバーが有効ではありません。
- **xrandr** ユーティリティーは、解像度、ローテーション、およびレイアウトの処理方法が異なるため、**Wayland** では有効ではありません。画面を操作するその他の **X.org** ユーティリティーも、**Wayland** では機能しません。
- 画面の録画、リモートデスクトップ、およびアクセシビリティーは、**Wayland** では正常に機能しない場合があります。
- クリップボードマネージャーは利用できません。

- **Wayland** は、仮想マシンビューアーなどの X11 アプリケーションのキーボードグラブを無視します。
- ゲスト仮想マシンの **Wayland** には安定性とパフォーマンスに問題があるため、仮想環境では X11 セッションを使用することが推奨されます。

X.org 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 **X.org** 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 **X.org** にフォールバックします。

- NVIDIA バイナリードライバー
- **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 legacy X11 オプションを選択します。

Wayland の詳細は [Wayland](#) を参照してください。

(BZ#1589678)

デフォルトでは有効でないリポジトリにある RPM パッケージの検索

デスクトップ用の追加のリポジトリは、デフォルトでは有効になりません。無効化は、対応する `.repo` ファイルの `enabled=0` 行で指定されています。PackageKit を使用して、そのようリポジトリからパッケージをインストールしようとする、アプリケーションが利用できないことを知らせるエラーメッセージが PackageKit により示されます。パッケージを利用できるようにするには、各 `.repo` ファイルの `enabled=0` の行を `enabled=1` に変更します。

(JIRA:RHELPLAN-2878)

パッケージ管理用の GNOME Software

Red Hat Enterprise Linux 7 のグラフィック環境でパッケージを管理するツール群を提供していた **gnome-packagekit** パッケージが利用できなくなりました。Red Hat Enterprise Linux 8 では、アプリケーションと `gnome-shell` 拡張機能のインストールと更新を可能にする **GNOME Software** ユーティリティーにより同様の機能が提供されます。**GNOME Software** は、**gnome-software** パッケージで配布されます。

(JIRA:RHELPLAN-3001)

Wayland の GNOME Shell で利用可能な分数スケール

Wayland の **GNOME Shell** セッションで、分数のスケール機能が利用できます。この機能は、GUI を分数でスケールでき、特定のディスプレイでスケールした GUI の出現を改善します。

この機能は現在試験的なものなので、デフォルトでは無効になっていることに注意してください。

分数スケールを有効にするには、次のコマンドを実行します。

-

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

(BZ#1668883)

5.1.9. ハードウェアの有効化

fwupd を使用したファームウェア更新が利用可能

RHEL 8 は、**fwupd** デーモンを使用した UEFI Capsule、Device Firmware Upgrade (DFU) などのファームウェア更新に対応します。デーモンは、セッションソフトウェアがローカルマシンのデバイスファームウェアを自動的に更新できるようにします。

更新を表示して適用する場合は、以下を使用できます。

- GUI ソフトウェアマネージャー (GNOME ソフトウェアなど)
- **fwupdmgr** コマンドラインツール

メタデータファイルは Linux Vendor Firmware Service (LVFS) セキュアポータルから自動的にダウンロードされ、D-Bus 経由で **fwupd** に送信されます。適用が必要な更新がダウンロードされ、ユーザー通知と更新の詳細が表示されます。更新を実行する前に、ユーザーはファームウェアの更新操作に明示的に合意する必要があります。

LVFS へのアクセスは、デフォルトでは無効になっていることに注意してください。

LVFS へのアクセスを有効にするには、GNOME ソフトウェアで **sources** ダイアログのスライダーをクリックするか、**fwupdmgr enable-remote lvfs** コマンドを実行します。**fwupdmgr** を使用して更新リストを取得する場合は、LVFS を有効にするかどうか尋ねられます。

LVFS にアクセスすると、ハードウェアベンダーからファームウェアの更新を直接入手できます。このような更新は、Red Hat QA による検証が行われていないことに注意してください。

(BZ#1504934)

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)

5.1.10. ID 管理

Directory Server で新しいパスワード構文の確認

この機能拡張により、Directory Server に新しいパスワード構文確認機能が追加されました。管理者は、たとえば、辞書の確認を有効にし、文字のシーケンスや回文を使用して許可または拒否を行います。したがって、これを有効にすると、Directory Server のパスワードポリシー構文の確認により、より安全なパスワードが強制されます。

(BZ#1334254)

Directory Server が提供する内部操作ログへの対応が改善

サーバーとクライアントが開始した Directory Server の操作により、バックグラウンドで追加の操作が発生します。以前は、サーバーが、内部操作の **Internal** 接続キーワードのみを記録し、操作 ID は常に **-1** に設定されていました。この機能拡張により、Directory Server は実際の接続および操作の ID のログを記録できるようになりました。この操作の原因となったサーバーまたはクライアントの操作に対する内部操作を追跡できるようになりました。

(BZ#1358706)

tomcatjss ライブラリーに、AIA 拡張のレスポnderを使用した OCSP 確認に対応

この機能拡張により、**tomcatjss** ライブラリーは、証明書の Authority Information Access (AIA) 拡張のレスポnderを使用した Online Certificate Status Protocol (OCSP) の確認に対応します。これにより、Red Hat Certificate System の管理者は、AIA 拡張の URL を使用して OCSP の確認を設定できるようになりました。

(BZ#1636564)

pki subsystem-cert-find コマンドおよび pki subsystem-cert-show コマンドが証明書のシリアル番号を表示

この機能強化により、証明書システムの **pki subsystem-cert-find** コマンドおよび **pki subsystem-cert-show** コマンドでは、出力の証明書のシリアル番号が表示されるようになりました。シリアル番号は重要な情報であり、他の複数のコマンドで必要になることがよくあります。これにより、証明書のシリアル番号を識別するのが簡単になりました。

(BZ#1566360)

証明書システムで pki user コマンドおよび pki group コマンドが非推奨に

今回の更新で、証明書システムの新しい **pki <subsystem>-user** コマンドおよび **pki <subsystem>-group** コマンドが、**pki user** コマンドおよび **pki group** コマンドに置き換まりました。以前のコマンドも引き続き使用できますが、コマンドが非推奨であるというメッセージと、代替となる新しいコマンドが表示されます。

(BZ#1394069)

証明書システムでシステム証明書のオフライン更新に対応

この機能強化により、管理者はオフライン更新機能を使用して、証明書システムで設定されているシステム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、Certificate System が起動できなくなります。この機能強化により、管理者が期限切れのシステム証明書を置き換える必要がなくなりました。

(BZ#1669257)

証明書システムで、外部 CA 署名用に SKI 拡張を使用して CSR を作成できるようになる

この機能強化により、証明書システムで、外部認証局 (CA) 署名用の SKI (Subject Key Identifier) 拡張子を持つ証明書署名要求 (CSR) の作成に対応します。特定の CA は、特定の値で、または CA 公開鍵から派生したこの拡張子を必要とします。これにより、管理者は **pkispawn** ユーティリティーに渡される設定ファイルの **pki_req_ski** パラメーターを使用して、SKI 拡張子を持つ CSR を作成できるようになりました。

(BZ#1656856)

SSSD が、AD ドメインのフォールバックとして [nss] セクションの fallback_homedir 値を使用しなくなりました。

RHEL 7.7 以前では、Active Directory (AD) プロバイダーの SSSD **fallback_homedir** パラメーターには

デフォルト値がありませんでした。**fallback_homedir** が設定されていない場合は、**/etc/sss/sss.conf** ファイルの **[nss]** セクションにある同じパラメーターの値が SSSD によって使用されていました。セキュリティを向上させるために、RHEL 7.7 の SSSD では、**fallback_homedir** のデフォルト値が導入されました。これにより、**[nss]** に設定された値に戻らなくなりました。AD ドメインの **fallback_homedir** パラメーターにデフォルトとは異なる値を使用する場合は、ドメインのセクションで手動で設定する必要があります。

(BZ#2213183)

SSSD で、複数のスマートカード認証デバイスの中から1つを選択可能

デフォルトでは、SSSD (System Security Services Daemon) が、スマートカード認証用デバイスを自動的に検出しようとします。複数のデバイスを接続している場合は、最初に見つかったデバイスを SSSD が選択します。したがって、特定のデバイスを選択することはできません。失敗する可能性もあります。

この更新では、**sss.conf** 設定ファイルの **[pam]** セクションに、新しい **p11_uri** オプションを設定できるようになりました。このオプションを使用すると、スマートカード認証に使用するデバイスを定義できます。

たとえば、OpenSC PKCS#11 モジュールにより検出されたスロット ID 2 のリーダーを選択するには、以下を追加します。

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

sss.conf の **[pam]** セクションに追加します。

詳細は、**man sss.conf** ページを参照してください。

(BZ#1620123)

SSSD によりローカルユーザーがキャッシュに格納され、nss_sss モジュールにより処理される

RHEL 8 の System Security Services Daemon (SSSD) は、デフォルトで **/etc/passwd** ファイルおよび **/etc/groups** ファイルから、ユーザーとグループにサービスを提供します。**sss** の **nsswitch** モジュールは、**/etc/nsswitch.conf** のファイルに先行します。

SSSD を通して、ローカルユーザーにサービスを提供する利点は、**nss_sss** モジュールに **memory-mapped cache** があるため、NSS (Name Service Switch) への各要求でディスクにアクセスしてファイルを開くのと比較して、Name Service Switch (NSS) ルックアップが高速になることでした。これまで、ディスクへのアクセスの高速化は、Name サービスキャッシュデーモン (**nscd**) により行われていました。ただし、SSSD と **nscd** の両方が独自の独立したキャッシュを使用するため、**nscd** と SSSD を並行して使用するのには容易ではありません。そのため、SSSD が、リモートドメイン (LDAP、Active Directory など) のユーザーにもサービスを提供している環境で **nscd** を使用すると、予期しない動作が発生する可能性があります。

今回の更新で、RHEL 8 でローカルユーザーおよびグループの解決が速くなりました。**root** ユーザーは、SSSD に処理されることがないため、SSSD の潜在的なバグが **root** の解決に影響を及ぼすことはありません。また、SSSD が実行していない場合は、問題を回避するために、**nss_sss** モジュールが **nss_files** にフォールバックすることでこの状況が適切に処理されます。SSSD を設定する必要はありません。files ドメインは自動的に追加されます。

(JIRA:RHELPLAN-10439)

KCM が、KEYRING をデフォルトの認証情報キャッシュストレージとして置き換え

RHEL 8 でデフォルトの認証情報キャッシュストレージは、**sssd-kcm** デーモンにより強化されている Kerberos Credential Manager (KCM) です。KCM では、以前使用されていた KEYRING の制限 (名前空間がないためにコンテナ化された環境での使用や、クォータの表示および管理が困難など) が解消されています。

今回の更新で、RHEL 8 には、コンテナ環境により適した認証情報キャッシュが含まれ、将来のリリースでより多くの機能を構築するための基盤が提供されます。

(JIRA:RHELPLAN-10440)

Active Directory ユーザーが Identity Management を管理可能

今回の更新で、RHEL 8 は、Identity Management (IdM) グループのメンバーとして、Active Directory (AD) ユーザーのユーザー ID のオーバーライドを追加できるようになりました。ID オーバーライドは、特定の AD ユーザーまたはグループのプロパティが特定の ID ビュー (この場合は Default Trust View) 内でどのように見えるかを記述するレコードです。この更新により、IdM LDAP サーバーは、IdM グループのアクセス制御ルールを AD ユーザーに適用できます。

AD ユーザーは、IdM UI のセルフサービス機能 (SSH キーのアップロード、個人のデータの変更など) を使用できるようになりました。AD 管理者は、アカウントおよびパスワードを 2 つ使用しなくても、IdM を完全に管理できるようになります。現在、AD ユーザーが、IdM で選択されている機能を利用できない場合があります。

(JIRA:RHELPLAN-10442)

sssctl が、IdM ドメインの HBAC ルールレポートを出力

今回の更新で、System Security Services Daemon (SSSD) の **sssctl** ユーティリティーは、Identity Management (IdM) ドメインのアクセス制御レポートを出力できるようになりました。この機能は、規制上の理由から、特定のクライアントマシンにアクセスできるユーザーとグループのリストを表示するニーズを満たします。IdM クライアントで **sssctl access-report domain_name** を実行すると、クライアントマシンに適用する IdM ドメインで解析されたホストベースのアクセス制御 (HBAC) ルールのサブセットを表示します。

IdM 以外のプロバイダーは、この機能に対応していません。

(JIRA:RHELPLAN-10443)

Identity Management パッケージがモジュールとして利用可能

RHEL 8 では、Identity Management (IdM) サーバーおよびクライアントのインストールに必要なパッケージがモジュールとして公開されています。**client** ストリームは、**idm** モジュールのデフォルトのストリームなので、このストリームを有効にしなくてもクライアントのインストールに必要なパッケージをダウンロードできます。

IdM サーバーのモジュールストリームは **DL1** ストリームと呼ばれています。このストリームには、さまざまなタイプの IdM サーバーに対応するプロファイル (server、dns、adtrust、client、および default) が含まれています。**DL1** ストリームの特定プロファイルに含まれるパッケージをダウンロードする場合は、以下を行います。

1. ストリームを有効にします。
2. ストリーム経由で配信される RPM に切り替えます。
3. **yum module install idm:DL1/profile_name** コマンドを実行します。

特定のストリームを有効にし、そこからパッケージをダウンロードし、新しいモジュールストリームに切り替えるには、以下を行います。

1. インストールされている関連コンテンツをすべて削除し、現在のモジュールストリームを無効にします。
2. 新しいモジュールストリームを有効にします。

(JIRA:RHELPLAN-10438)

RHEL 8 のセッション記録ソリューションが追加

Red Hat Enterprise Linux 8 (RHEL 8) にセッション記録ソリューションが追加されました。新しい **tlog** パッケージと、それに関連付けられた Web コンソールセッションプレイヤーにより、ユーザー端末セッションを録画および再生できるようになりました。録画は、システムセキュリティーサービスデーモン (SSSD) サービスを介して、ユーザー別またはユーザーグループ別に設定できます。端末への入出力はすべてキャプチャーされ、テキストベースの形式でシステムジャーナルに保存されます。セキュリティー上の理由から、未加工のパスワードやその他の機密情報を傍受されないように、入力はデフォルトでは非アクティブになっています。

このソリューションを、セキュリティーが重要なシステムでユーザーセッションを監査するのに使用できます。セキュリティー違反が発生した場合に、記録したセッションをフォレンジック分析として評価できます。システム管理者は、セッション記録をローカルに設定し、**tlog-play** ユーティリティーを使用して、RHEL 8 Web コンソールインターフェイスまたはコマンドラインインターフェイスから、結果を表示できるようになりました。

(JIRA:RHELPLAN-1473)

authselect がユーザー認証の設定を簡略化

今回の更新で、**authconfig** ユーティリティーに代わる、RHEL 8 ホストでのユーザー認証の設定を簡素化する **authselect** ユーティリティーが導入されました。**authselect** には、システム管理者に対して PAM 設定の変更を簡単に行える、より安全な PAM スタック管理アプローチがあります。**authselect** を使用すると、パスワード、証明書、スマートカード、フィンガープリントなどの認証方法を設定できます。**authselect** では、リモートドメインに参加するのに必要なサービスを設定しないことに注意してください。このタスクは、**realmd**、**ipa-client-install** のような専門のツールにより実行されます。

(JIRA:RHELPLAN-10445)

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)

5.1.11. コンパイラーおよび開発ツール

Boost がバージョン 1.66 へ更新

Boost C++ ライブラリーがアップストリームバージョン 1.66 に更新されました。Red Hat Enterprise Linux 7 に含まれる **Boost** のバージョンは 1.53 です。詳細は、アップストリームの changelog ([Boost Version History](#)) を参照してください。

今回の更新で、以前のバージョンとの互換性を損なう以下の変更が行われています。

- splay コンテナの **bs_set_hook()** 関数、**splay_set_hook()** 関数と、**Intrusive** ライブラリーの **splaytree_algorithms()** 関数の追加パラメーター **bool splay = true** が削除されました。
- JSON ファイルのコメントまたは文字列の連結が、**Property Tree** ライブラリーのパーサーで対応されなくなりました。
- **Math** ライブラリーのディストリビューションや特別な関数の一部を、ドキュメントに記載されているとおりに動作し、最大の有限値を返さずに **overflow_error** を送出するように修正されました。
- **Math** ライブラリーの一部のヘッダーを、**libs/math/include_private** ディレクトリーに移動しました。
- **Regex** ライブラリーの **basic_regex<>::mark_count()** 関数および **basic_regex<>::subexpression(n)** 関数の挙動が、ドキュメントに合わせて変更になりました。
- **Variant** ライブラリーで可変テンプレートを使用すると、メタプログラミング機能が壊れる可能性があります。
- API **boost::python::numeric** が削除されました。代わりに **boost::python::numpy** を使用できます。
- 非オブジェクト型へのポインターに対する算術演算が、**Atomic** ライブラリーでは提供されなくなりました。

(BZ#1494495)

Unicode 11.0.0 への対応

Red Hat Enterprise Linux のコア C ライブラリーの **glibc** が、Unicode 標準バージョン 11.0.0 に対応するように更新されました。このため、文字セット間の書き直しおよび変換を含むワイド文字およびマルチバイト文字の API のすべてが、この規格に準拠した正確な情報を提供します。

(BZ#1512004)

boost が Python から独立

今回の更新で、**boost** パッケージをインストールしても、**Boost.Python** ライブラリーが依存関係としてインストールされなくなりました。**Boost.Python** を使用するには、**boost-python3** パッケージまたは **boost-python3-devel** パッケージを明示的にインストールする必要があります。

(BZ#1616244)

新しい **compat-libgfortran-48** パッケージが利用可能に

Fortran ライブラリーを使用する Red Hat Enterprise Linux 6 および 7 のアプリケーションとの互換性のために、**libgfortran.so.3** ライブラリーを提供する **compat-libgfortran-48** 互換性パッケージが新たに利用できるようになりました。

(BZ#1607227)

GCC で Retpoline に対応

今回の更新で、GCC への retpoline に対応するようになりました。retpoline は、CVE-2017-5715 で説明されているように、Spectre Variant 2 攻撃を軽減するためのオーバーヘッドを減らすために、カーネルで使用されるソフトウェア構造です。

(BZ#1535774)

toolchain コンポーネントにおける 64 ビットの ARM アーキテクチャーへの対応の強化

ツールチェーンコンポーネントの **GCC** および **binutils** は、64 ビットの ARM アーキテクチャーへの対応が拡張されました。以下に例を示します。

- **GCC** と **binutils** で Scalable Vector Extension (SVE) に対応するようになりました。
- ARM v8.2 で提供されていた **FP16** データ型への対応が **GCC** に追加されました。**FP16** データ型は特定のアルゴリズムのパフォーマンスを改善します。
- **binutils** のツールで、ポインター認証を含む ARM v8.3 アーキテクチャーの定義に対応するようになりました。ポインター認証機能により、悪意のあるコードが、独自の関数ポインターを作成してプログラムまたはカーネルの通常の実行を妨害するのを防ぎます。その結果、コードのさまざまな場所に分岐する際に信頼できるアドレスだけが使用され、セキュリティーが向上します。

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

IBM POWER システム用 **glibc** の最適化

今回の更新で、IBM POWER 8 と IBM POWER 9 の両アーキテクチャー向けに最適化された新しいバージョンの **glibc** が提供されるようになりました。これにより、IBM POWER 8 システムおよび IBM POWER 9 システムが、実行時に最適化された適切な **glibc** バリエーションに自動的に切り替わるようになりました。

(BZ#1376834)

GNU C ライブラリーがバージョン 2.28 に更新

Red Hat Enterprise Linux 8 には、バージョン 2.28 の GNU C ライブラリー (**glibc**) が同梱されます。以下は、主な改善点です。

- セキュリティーの強化機能:
 - **AT_SECURE** フラグが付いた安全なバイナリーファイルは、**LD_LIBRARY_PATH** 環境変数を無視します。
 - シャットダウンのスピードを上げ、危険にさらされた環境でより多くのコードが実行されることを回避するため、スタックの確認の失敗に対してバックトレースが表示されなくなりました。
- パフォーマンスの向上:
 - スレッドローカルキャッシュにより、**malloc()** 関数のパフォーマンスが向上しました。
 - ライブラリーパフォーマンスの特性を変更する **GLIBC_TUNABLES** 環境変数が追加されました。
 - スレッドセマフォの実装が改善され、スケーラブルな **pthread_rwlock_xxx()** 関数が新たに追加されました。
 - **math** ライブラリーのパフォーマンスが向上しました。
- Unicode 11.0.0 への対応が追加されました。
- 標準規格の ISO/IEC/IEEE 60559:2011、IEEE 754-2008、および ISO/IEC TS 18661-3:2015 で定義されている 128 ビット浮動小数点への対応が改善されました。

- **/etc/resolv.conf** 設定ファイルに関連するドメインネームサービス (DNS) スタブリゾルバーが改善しました。
 - ファイルを変更すると、設定が自動的に再読み込みされるようになりました。
 - 任意の検索ドメインに対応するようになりました。
 - **rotate** オプションの適切なランダム選択が追加されました。
- 開発用の機能が新たに追加されました。以下は例となります。
 - カーネルコールの **preadv2** および **pwritev2** で使用される Linux のラッパー関数
 - 新しい関数 (**reallocarray()**、**explicit_bzero()** など)
 - **posix_spawnattr_setflags()** 関数の新しいフラグ (**POSIX_SPAWN_SETSID** など)

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

RHEL で CMake が利用可能に

CMake ビルドシステムバージョン 3.11 は、Red Hat Enterprise Linux 8 の **cmake** パッケージで入手できます。

(BZ#1590139, BZ#1502802)

make バージョン 4.2.1

Red Hat Enterprise Linux 8 には、**make** ビルドツールバージョン 4.2.1 が同梱されています。以下は、主な変更点です。

- レシピに失敗すると、makefile の名前とレシピの行番号が表示されます。
- ターゲットの追跡を可能にするために、**--trace** オプションが追加されました。このオプションを使用すると、起動時に、非表示になっているものも含むすべてのレシピが、レシピが置かれているファイル名、行番号、そしてレシピを呼び出す前提条件とともに出力されます。
- 明示的ルールと暗黙的ルールを混在させても、**make** が実行を終了することはなくなりました。代わりに警告が表示されます。この構文は非推奨になっており、将来、完全に削除される可能性があります。
- テキストをファイルに書き込む **\$(file ...)** 関数が追加されました。テキスト引数なしで呼び出しを行うと、ファイルを開くだけですぐに閉じます。
- 新しいオプション **--output-sync** または **-O** を使用すると、複数のジョブからの出力をジョブごとにグループ化するため、並列ビルドのデバッグを容易にできます。
- **--debug** オプションは、現在有効なデバッグ設定を無効にする **n** (none) フラグも受け入れるようになりました。
- BSD makefiles との互換性を高めるために、**\$(shell ...)** 関数の代わりに **!=** シェル割り当て演算子が追加されました。演算子と関数の詳細と相違点は、GNU の **make** マニュアルを参照してください。これにより、**variable!=value** のように、感嘆符で終わり、その後に代入が続く名前の変数は、新しい構文として解釈されるようになりました。以前の動作に戻すには、**variable! =value** のように、感嘆符の後に空白を追加します。
- POSIX 規格で定義されている代入演算子 **::=** が追加されました。

- **.POSIX** 変数を指定している場合、**make** は、バックスラッシュと改行を処理する POSIX 標準規格の要件に従います。このモードでは、バックスラッシュ前の後続スペースが保持され、各バックスラッシュとそれに続く改行文字と空白文字が、空白文字1つに変換されます。
- **MAKEFLAGS** 変数および **MFLAGS** 変数の動作が、より正確に定義されました。
- 新しい変数 **GNUMAKEFLAGS** は、**MAKEFLAGS** と同じ **make** フラグとして解析されます。これにより、GNU の **make** 固有のフラグを **MAKEFLAGS** 以外で保存でき、makefile の移植性が向上します。
- ホストアーキテクチャーを含む新しい変数 **MAKE_HOST** が追加されました。
- 新しい変数 **MAKE_TERMOUT** と **MAKE_TERMERR** には、それぞれ **make** が標準出力とエラーを端末に書き込んでいるかどうかを示されます。
- makefile の **MAKEFLAGS** 変数に **-r** オプションおよび **-R** オプションを設定すると正しく機能し、すべての組み込み規則と変数がそれぞれ適切に削除されるようになりました。
- **.RECIPEPREFIX** 設定はレシピごとに記録されるようになりました。さらに、そのレシピでデプロイメントされた変数も、そのレシピの接頭辞設定を使用します。
- **.RECIPEPREFIX** 設定と、ターゲット固有の変数はすべて、コメントとしてではなく、makefile にあるように **-p** オプションの出力に表示されます。

(BZ#1641015)

SystemTap バージョン 4.0

Red Hat Enterprise Linux 8 には、**SystemTap** 計測ツールのバージョン 4.0 が同梱されています。以下は、主な改善点です。

- extended Berkeley Packet Filter (eBPF) バックエンド (特に文字列と関数) が改良されました。このバックエンドを使用するには、**SystemTap** を起動する際に **--runtime=bpf** オプションを追加します。
- Prometheus モニタリングシステムで使用するエクスポートネットワークサービスが新たに追加されました。
- システムコールプロービングの実装は、必要に応じてカーネルトレースポイントを使用するように改善されました。

(BZ#1641032)

binutils バージョン 2.30 の改良

Red Hat Enterprise Linux 8 には、バージョン 2.30 の **binutils** パッケージが同梱されています。以下は、主な改善点です。

- 新しい IBM Z アーキテクチャー拡張への対応が改善されました。

リンカー:

- リンカーは、デフォルトで、コードと、読み取り専用データを別々のセグメントに配置するようになりました。その結果、動的ローダーが、読み取り専用データを含むメモリーページの実行を無効にするため、作成される実行ファイルは大きくなり、実行の安全性が向上しました。
- バイナリーファイルに関する動的ローダーへのヒントを提供する GNU Property ノートに対応するようになりました。

- 以前は、リンカーが、Indirect Branch Tracking (IBT) テクノロジーに無効な実行コードを生成していました。そのため、生成された実行ファイルを開始できませんでした。このバグは修正されています。
- 以前は、**gold** リンカーが、プロパティノートを不適切にマージしていました。このため、生成されたコードで誤ったハードウェア機能が有効になり、コードが突然終了することがありました。このバグは修正されています。
- 以前は、**gold** リンカーが、アーキテクチャーに合わせた調整を行うために、最後にパディングバイトを持つノートセクションを作成していました。動的ローダーではパディングが予想されていなかったため、読み込んでいるプログラムが突然終了していました。このバグは修正されています。

その他のツール:

- **readelf** ツールおよび **objdump** ツールに、リンクをたどって別ファイルのデバッグ情報に移動し、それらの情報を表示するオプションが追加されました。
- 新しい **--inlines** オプションは、**objdump** ツールの既存の **--line-numbers** オプションを拡張したもので、インライン関数のネスト情報を表示します。
- **nm** ツールに、シンボルの名前の後にバージョン情報を表示する **--with-version-strings** オプションが新たに追加されました。
- ARMv8-R アーキテクチャーと、Cortex-R52、Cortex-M23、および Cortex-M33 のプロセッサへの対応がアセンブラーに追加されました。

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

Performance Co-Pilot バージョン 4.3.0

Red Hat Enterprise Linux 8 には、**Performance Co-Pilot (PCP)** バージョン 4.3.0 が同梱されています。以下は、主な改善点です。

- **pcp-dstat** ツールに、履歴分析とコンマ区切り値 (CSV) 形式の出力が追加されました。
- ログユーティリティーで、メトリックラベルとヘルプテキストレコードを使用できます。
- **pmdaperfevent** ツールが、低レベルの同時マルチスレッド (SMT) で正しい CPU 番号を報告するようになりました。
- **pmdapostgresql** ツールが PostgreSQL シリーズ 10.x に対応するようになりました。
- **pmdaredis** ツールが Redis シリーズ 5.x に対応するようになりました。
- 動的プロセスフィルタリングと、各プロセスのシステムコール、ucall、および ustat により、**pmdabcc** ツールが強化されました。
- **pmdammv** ツールが、メトリックラベルをエクスポートするようになり、フォーマットのバージョンが 3 に増えました。
- **pmdagfs2** ツールで、glock および glock ホルダーの計測に対応するようになりました。
- SELinux ポリシーにいくつかの変更が加えられました。

(BZ#1641034)

メモリー保護キー

今回の更新で、スレッドごとにページ保護フラグの変更を可能にするハードウェア機能が有効になりました。**pkey_alloc()** 関数、**pkey_free()** 関数、および **pkey_mprotect()** 関数に、**glibc** システムコールラッパーが新たに追加されました。さらに、各スレッドの保護フラグへのアクセスを許可するために、**pkey_set()** 関数および **pkey_get()** 関数が追加されました。

(BZ#1304448)

GCC が IBM Z で z13 にデフォルト設定

今回の更新で、IBM Z アーキテクチャーの GCC はデフォルトで z13 プロセッサのコードを構築し、このコードは z14 プロセッサ用に調整されました。これは、**-march=z13** オプションおよび **-mtune=z14** オプションの使用と同じです。ターゲットアーキテクチャーおよびチューニングのオプションを明示的に使用することで、このデフォルトを上書きできます。

(BZ#1571124)

elfutils がバージョン 0.174 へ更新

Red Hat Enterprise Linux 8 では、**elfutils** パッケージのバージョン 0.174 が利用できます。以下は、主な変更点です。

- **eu-readelf** ツールでは、変数を示す際に、負の値を、符号がない大きな値として使用したり、符号がない大きな値を、負の値として使用していました。これは修正され、**eu-readelf** が、定数値型のサイズおよび定数値型のサイズと符号属性を調べて、正しく表示するようになりました。
- CU がない **.debug_line** データを読み込む新しい関数 **dwarf_next_lines()** が **libdw** ライブラリーに追加されました。この関数は、**dwarf_getsrclines()** 関数および **dwarf_getsrcfiles()** 関数の代わりに使用できます。
- 以前は、セクションが 65280 を超えるファイルでは、**libelf** ライブラリーおよび **libdw** ライブラリー、ならびにそのライブラリーを使用するツールでエラーが発生する可能性があります。このバグは修正されています。これにより、ELF ファイルヘッダーで拡張された **shnum** 値および **shstrndx** 値は正しく処理されます。

(BZ#1641007)

Valgrind がバージョン 3.14 へ更新

Red Hat Enterprise Linux 8 には、Valgrind 実行コード分析ツールバージョン 3.14 が同梱されています。以下は、主な変更点です。

- **--keep-debuginfo** オプションが追加され、アンロードされたコードのデバッグ情報を保持できるようになりました。これにより、保存されるスタックトレースには、メモリーから削除されたコードのファイルおよび行の情報が含まれる可能性があります。
- ソースファイル名と行番号に基づいた抑制が追加されました。
- **Helgrind** ツールは、**--delta-stacktrace** オプションにより、完全な履歴のスタックトレースの計算を指定するようになりました。特に、**--history-level=full** とともにこのオプションを使用すると、**Helgrind** のパフォーマンスを最大 25% 向上させることができます。
- Intel および AMD 64 ビットのアーキテクチャーで最適化されたコードで、**Memcheck** ツールの誤検出の割合が減少し、ARM 64 ビットアーキテクチャーが減少しました。定義確認処理を制御する **--expensive-definedness-checks** は使用できますが、パフォーマンスに影響を及ぼします。

- Valgrind では、IBM Power Systems (リトルエンディアン) の命令で認識できるものが増えました。
- Valgrind が、IBM Z アーキテクチャーの z13 プロセッサで、整数および文字列の vector 命令の大部分を処理するようになりました。

新しいオプションと既知の制限の詳細は、man ページの **valgrind(1)** を参照してください。

(BZ#1641029, BZ#1501419)

GDB バージョン 8.2

Red Hat Enterprise Linux 8 には、GDB デバッガーのバージョン 8.2 が同梱されています。以下は、主な変更点です。

- IPv6 プロトコルは、GDB および **gdbserver** によるリモートデバッグに対応しています。
- デバッグ情報なしのデバッグが改善されました。
- GDB ユーザーインターフェイスのシンボル補完が、ABI タグや名前空間などの、より構文的な設定を使用することで、より良い提案を提供するように改善されました。
- コマンドが、バックグラウンドで実行できるようになりました。
- Rust プログラミング言語で作成されたプログラムのデバッグが可能になりました。
- **_Alignof** 演算子、**alignof** 演算子、C++ rvalue 参照、および C99 変数長自動配列へのパーサーの対応により、C 言語および C++ 言語のデバッグが改善されました。
- GDB 拡張スクリプトが、Guile スクリプト言語を使用できるようになりました。
- 拡張機能用の Python スクリプト言語インターフェイスが改善され、API 機能、フレームデコレーター、フィルター、およびアンワインダーが新たに追加されました。また、GDB 設定の **.debug_gdb_scripts** セクションにあるスクリプトも自動的に読み込まれます。
- GDB は、Python バージョン 3 を使用して、プリンター、フレームデコレーター、フィルター、アンワインダーなどのスクリプトを実行するようになりました。
- ARM アーキテクチャーおよび 64 ビットの ARM アーキテクチャーが改善され、Thumb 32 ビット命令およびシステムコール命令を含む、プロセスの実行記録および再生が追加されました。
- GDB は、64 ビット ARM アーキテクチャーで Scalable Vector Extension (SVE) に対応するようになりました。
- Intel PKU レジスターおよび Intel プロセッサトレースへの対応が追加されました。
- 記録と再生の機能で、Intel ベースのシステムの **rdrand** および **rdseed** の命令を含むように拡張されました。
- IBM Z アーキテクチャーの GDB の機能は、トレースポイントと高速トレースポイント、ベクトルレジスターと ABI、および **Catch** システムコールに対応するように拡張されました。また、GDB が、アーキテクチャーの新しい命令に対応するようになりました。
- GDB は、64 ビット ARM アーキテクチャーで、SystemTap の静的ユーザー空間プローブ (SDT) を使用できるようになりました。

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

RHEL の `glibc` のローカライゼーションが複数のパッケージで配布

RHEL 8.0 では、1つの `glibc-common` パッケージで `glibc` のロケールと翻訳が提供されなくなりました。代わりに、すべてのロケールと言語が `glibc-langpack-CODE` パッケージで利用できるようになりました。また、多くの場合はデフォルトですべてのロケールがインストールされず、インストーラーで選択した言語だけがインストールされます。その他に必要なロケールパッケージを個別にインストールする必要があります。または、以前と同じようにインストールされたすべての `glibc` ロケールを含むロケールアーカイブを取得する場合は、`glibc-all-langpacks` をインストールします。

詳細は、[Using langpacks](#) を参照してください。

(BZ#1512009)

GCC バージョン 8.2

Red Hat Enterprise Linux 8 では、GCC ツールチェーンは GCC 8.2 リリースシリーズに基づいています。以下は、主な変更点です。

- エイリアス解析、ベクトル化機能の改善、同一コードの折りたたみ、プロシーチャー間解析、ストアマージの最適化パスなど、一般的な最適化が多数追加されました。
- Address Sanitizer が改善されました。Leak Sanitizer と Undefined Behavior Sanitizer が追加されました。
- デバッグ情報が DWARF5 形式で生成できるようになりました。この機能は実験的なものです。
- ソースコードのカバレッジ解析ツール GCOV が、さまざまな改良とともに拡張されました。
- 特定の、より多くのプログラムエラーを静的に検出するために、新しい警告と改善された診断が追加されました。
- GCC が、生成したコードをさらに強化するツールを提供するように拡張されました。セキュリティに関する改善には、オーバーフロー確認用の組み込み機能、スタッククラッシュに対する追加の保護、制御フロー命令のターゲットアドレスの確認、バインドされた文字列操作関数の警告、範囲外の配列インデックスの検出に関する警告などがあります。

アーキテクチャーおよびプロセッサへの対応の改善点は次のとおりです。

- Intel AVX-512 アーキテクチャー、その多数のマイクロアーキテクチャー、および Intel Software Guard Extensions (SGX) にアーキテクチャー固有の新しいオプションが複数追加されました。
- コード生成は、現在、64 ビットの ARM アーキテクチャー LSE 拡張、ARMv8.2-A 16 ビット浮動小数点拡張 (FPE)、およびアーキテクチャーのバージョン ARMv8.2-A、ARMv8.3-A、および ARMv8.4-A が対象となります。
- IBM Z アーキテクチャーのプロセッサ z13 および z14 に対応するようになりました。

以下は、言語と標準規格に関連した主な変更点です。

- C 言語でコンパイルする際に使用されるデフォルトの標準規格が、GNU 拡張機能が含まれる C17 に変更になりました。
- C++ 言語でコードをコンパイルする際に使用されるデフォルトの標準規格が、GNU 拡張機能が含まれる C++14 に変更になりました。
- C++ ランタイムライブラリーで、C++11 および C++14 の標準規格に対応するようになりました。

- C++ コンパイラーが、C++14 標準規格を実装するようになりました。
- C 言語規格 C11 への対応が改善されました。
- 新しい `__auto_type` の GNU C 拡張機能が、C 言語の C++11 の `auto` キーワード機能のサブセットを提供します。
- ISO/IEC TS 18661-3:2015 標準規格が指定する型名 `_FloatN` および `_FloatNx` が、C フロントエンドで認識されるようになりました。
- 空のクラスを引数として渡すと、プラットフォーム ABI で要求される、Intel 64 アーキテクチャーおよび AMD64 アーキテクチャーで領域を使用しなくなります。
- C++11 の `alignof` 演算子により返される値は、C の `_Alignof` 演算子と一致し、最小の配置を返すように修正されました。適切な配置を見つけるには、GNU 拡張機能 `__alignof__` を使用します。
- Fortran 言語コード用の `libgfortran` ライブラリーのメインバージョンが 5 に変更になりました。
- Ada (GNAT)、GCC Go、および Objective C/C++ 言語に対応しなくなりました。Go コード開発には Go Toolset を使用してください。

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

Go 暗号ライブラリー FIPS モードが、システム設定を受け入れるようになる

以前は、Go の標準暗号化ライブラリーは、ライブラリーを使用しているアプリケーションのビルド時に明示的に無効になっていない限り、常に FIPS モードを使用していました。これにより、Go ベースのアプリケーションのユーザーが、FIPS モードを使用するかどうかを制御できませんでした。この変更により、システムが FIPS モードで設定されていない場合、ライブラリーはデフォルトで FIPS モードになりません。これにより、RHEL システムの Go ベースのアプリケーションのユーザーが、Go 暗号化ライブラリーの FIPS モードの使用をさらに制御できます。

(BZ#1633351)

strace がバージョン 4.24 に更新

Red Hat Enterprise Linux 8 には、`strace` ツールバージョン 4.24 が同梱されています。以下は、主な変更点です。

- `-e inject=` オプションで、システムコールの改ざん機能が追加されました。これには、エラー、戻り値、遅延、およびシグナルの挿入が含まれます。
- システムコール修飾の構文が改善されました。
 - 正規表現でシステムコールをフィルタリングするために、`-e trace=/regex` オプションが追加されました。
 - `-e trace=` オプションのシステムコール修飾の前に疑問符を付けると、その修飾がいずれのシステムコールに一致しなくても `strace` を続行できます。
 - `-e trace` オプションのシステムコール修飾に、パーソナリティーの指定が追加されました。
- 終了理由の `kvm vcpu` のデコードが追加されました。これには、`-e kvm=vcpu` オプションを使用します。

- **-k** オプションを使用している場合は、**elfutils** の **libdw** ライブラリーがスタックのアンワインドに使用されるようになりました。また、シンボルのデマングルは、**libiberty** ライブラリーを使用して実行します。
- 以前は、**-r** オプションを指定すると、**strace** が **-t** オプションを無視していました。この問題が修正され、干渉を受けなくなりました。
- 追加モードで出力ファイルを開く **-A** オプションが追加されました。
- **xlat** 出力フォーマットを設定する **-X** オプションが追加されました。
- **-yy** オプションを使用したソケットアドレスのデコードが改善されました。また、**-yy** モードに、ブロックデバイスとキャラクターデバイスの番号出力が追加されました。
- IBM Z アーキテクチャーの **strace** ツールを1つ使用して、64 ビットと 32 ビットの両方のバイナリーを追跡できるようになりました。これにより、個別の **strace32** パッケージが RHEL 8 から削除されました。

また、以下の項目のデコードが追加、改善、または更新されました。

- **netlink** プロトコル、メッセージ、および属性
- **arch_prctl**、**bpf**、**getsockopt**、**io_pgetevent**、**keyctl**、**prctl**、**pkey_alloc**、**pkey_free**、**pkey_mprotect**、**ptrace**、**rseq**、**setsockopt**、**socket**、**statx**、その他のシステムコール
- **ioctl** システムコールの複数のコマンド
- さまざまな型の定数
- システムコールの **execveat**、**inotify_add_watch**、**inotify_init**、**select**、**symlink**、**symlinkat**、および間接引数が付いた **mmap** システムコールのパス追跡
- シグナルコードのリスト

(BZ#1641014)

RHEL 8 のコンパイラーツールセット

RHEL 8.0 は、以下のコンパイラーツールセットを Application Streams として提供します。

- Clang および LLVM Toolset 7.0.1 は、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 言語および C++ 言語用の Clang コンパイラー、LLDB デバッガー、コード解析の関連ツールを提供します。 [Using Clang and LLVM Toolset](#) を参照してください。
- Rust Toolset 1.31 は、Rust プログラミング言語コンパイラー **rustc**、**cargo** ビルドツールおよび依存マネージャー、**cargo-vendor** プラグイン、および必要なライブラリーを提供します。 [Using Rust Toolset](#) を参照してください。
- Go Toolset 1.11.5 は、Go プログラミング言語ツールおよびライブラリーを提供します。Go は、**golang** としても知られています。 [Using Go Toolset](#) を参照してください。

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

RHEL 8 における Java 実装および Java ツール

RHEL 8 AppStream リポジトリには、以下が含まれます。

- **java-11-openjdk** パッケージ。OpenJDK 11 Java Runtime Environment および OpenJDK 11 Java Software Development Kit を提供します。
- **java-1.8.0-openjdk** パッケージ。OpenJDK 8 Java Runtime Environment および OpenJDK 8 Java Software Development Kit を提供します。
- **icedtea-web** パッケージ (Java Web Start の実装を提供)。
- Java ライブラリーおよびコマンドラインツールを提供する **ant** モジュール。Java アプリケーションのコンパイル、アセンブル、テスト、および実行を行います。**ant** がバージョン 1.10 に更新されました。
- **maven** モジュールは、ソフトウェアプロジェクトの管理および解釈を行うツールを提供します。**maven** は、以前は、Software Collection から、またはサポート対象外の Optional チャンネルからしか入手できませんでした。
- **scala** モジュールは、Java プラットフォーム用の汎用プログラミング言語を提供します。**scala** は、以前は、Software Collection からしか入手できませんでした。

また、**java-1.8.0-ibm** パッケージは、Supplementary リポジトリを介して配布されます。Red Hat は、このリポジトリのパッケージをサポートしていません。

(BZ#1699535)

std::string および std::list における C++ ABI の変更

RHEL 7 (GCC 4.8) と RHEL 8 (GCC 8) との間で変更した **libstdc++** ライブラリーの **std::string** クラスおよび **std::list** クラスの Application Binary Interface (ABI) は、C++11 標準に従います。**libstdc++** ライブラリーは、古い ABI および新しい ABI の両方に対応しますが、その他の C++ システムライブラリーには対応しません。そのため、このライブラリーに動的にリンクするアプリケーションを再構築する必要があります。これは、C++98 を含むすべての C++ 標準モードに影響します。RHEL 7 で Red Hat Developer Toolset コンパイラーを使用して構築したアプリケーションにも影響します。このコンパイラーは、古い ABI を維持して、システムライブラリーとの互換性を維持します。

(BZ#1704867)

5.1.12. 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)

XFS が、共有コピーオンライトのデータエクステンメントに対応

XFS ファイルシステムは、共有コピーオンライトのデータエクステンメント機能に対応します。この機能により、2つ以上のファイルで共通のデータブロックセットを共有できます。共通ブロックを共有しているファイルのいずれかを変更すると、XFS が共通ブロックへのリンクを解除して、新しいファイルを作成します。これは、その他のファイルシステムに見られるコピーオンライト (COW) 機能と似ています。

共有コピーオンライトのデータエクステンメントには次の特徴があります。

高速

共有コピーを作成しても、ディスク I/O を使用しません。

容量を効果的に使用

共有ブロックは、ディスク容量を追加で使用しません。

透過的

共通ブロックを共有するファイルは、通常のファイルと同じように機能します。

ユーザー空間ユーティリティーは、次の目的で共有コピーオンライトのデータエクステンメントを使用できません。

- `cp --reflink` コマンドなどを使用した効果的なファイルのクローン作成
- ファイルごとのスナップショット

この機能は、Overlayfs、NFS などのカーネルサブシステムで、より効率的な操作に使用されます。

共有コピーオンライトのデータエクステンメントは、`xfsprogs` パッケージの **4.17.0-2.el8** 以降のバージョンで、XFS ファイルシステムを作成する際にデフォルトで有効になります。

直接アクセス (DAX) デバイスは、現在、共有コピーオンライトのデータエクステンメントを使用する XFS に対応していません。この機能を使用せずに XFS ファイルシステムを作成する場合は、次のコマンドを使用します。

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

Red Hat Enterprise Linux 7 は、読み取り専用モードでのみ、共有コピーオンライトのデータエクステンメントを使用する XFS ファイルシステムをマウントできます。

(BZ#1494028)

XFS ファイルシステムの最大サイズが 1024 TiB に

XFS ファイルシステムで対応可能な最大サイズが 500 TiB から 1024 TiB に増えました。

500 TiB を超えるファイルシステムでは、以下が必要です。

- メタデータの CRC 機能と、フリーの inode btree 機能の両方が、ファイルシステムフォーマットで有効になっています。
- 割り当てグループのサイズは、最低 512 GiB です。

RHEL 8 では、**mkfs.xfs** ユーティリティーが、デフォルトでこの要件を満たすファイルシステムを作成します。

この要件を満たしていない小規模なファイルシステムを、500 TiB を超えるサイズに拡張することには対応していません。

(BZ#1563617)

ext4 ファイルシステムが metadata checksum に対応

この更新により、**ext4** メタデータは **checksum** により保護されます。これにより、ファイルシステムが破損メタデータを認識させます。これにより破損を回避し、ファイルシステムの耐障害性を高めます。

(BZ#1695584)

VDO がすべてのアーキテクチャーに対応

Virtual Data Optimizer (VDO) が、RHEL 8 で対応しているすべてのアーキテクチャーで利用可能になりました。

対応しているアーキテクチャーのリストは、[2章 アーキテクチャー](#) を参照してください。

(BZ#1534087)

BOOM ブートマネージャーが、ブートエントリーを作成するプロセスを簡素化

BOOM は、Linux システム用のブートマネージャーで、ブートエントリー設定の BootLoader 仕様に対応するブートローダーを使用します。柔軟なブート設定が可能になり、ブートエントリーの新規作成や変更が容易になります。たとえば、LVM を使用して作成したシステムのスナップショットイメージを起動するためのエントリーです。

BOOM は、既存のブートローダー設定を変更せず、追加エントリーを挿入するだけです。既存の設定は維持され、ディストリビューションの統合 (カーネルのインストールや更新のスクリプトなど) は、以前と同じように引き続き機能します。

BOOM には、ブートエントリーを作成するタスクを容易にする、単純化されたコマンドラインインターフェイス (CLI) および API があります。

(BZ#1649582)

LUKS2 が、ボリューム暗号化のデフォルトフォーマットに

RHEL 8 では、レガシーの LUKS (LUKS1) 形式に代わり、LUKS バージョン 2 (LUKS2) の形式が使用されます。**dm-crypt** サブシステムおよび **cryptsetup** ツールでは、暗号化ボリュームのデフォルト形式として LUKS2 が使用されるようになりました。LUKS2 は、部分的なメタデータ破損イベントが発生した場合に備えて、暗号化されたボリュームにメタデータの冗長性と自動回復を提供します。

内部の柔軟なレイアウトにより、LUKS2 は将来の機能も可能にします。これは、**libcryptsetup** に組み込まれた一般的なカーネルキーリングトークンによる自動ロック解除に対応し、カーネルキーリング保持サービスに保存されているパズルピースを使用して LUKS2 ボリュームのロックを解除します。

以下は、その他の主な機能強化です。

- ラップ鍵暗号方式を使用した保護鍵の設定
- Policy-Based Decryption (Clevis) とのより簡単な統合
- 最大 32 個の鍵スロット (LUKS1 は鍵スロットを 8 個のみ提供します)

詳細は、man ページの **cryptsetup(8)** および **cryptsetup-reencrypt(8)** を参照してください。

(BZ#1564540)

Broadcom Emulex および Marvell Qlogic のファイバーチャンネルアダプターで NVMe/FC に完全対応

NVMe に対応する Broadcom Emulex アダプターおよび Marvell Qlogic Fibre Channel 32Gbit アダプターとともに使用すると、イニシエーターモードで NVMe/FC (NVMe over Fibre Channel) トランスポートタイプに完全に対応するようになりました。

Red Hat Enterprise Linux に同梱されていた RDMA (Remote Direct Memory Access) プロトコルに加えて、NVMe over Fibre Channel が、NVMe (Nonvolatile Memory Express) プロトコルのファブリックトランスポートタイプとして追加されました。

NVMe/FC を有効にするには、以下を行います。

- **lpfc** ドライバーで NVMe/FC を有効にするには、`/etc/modprobe.d/lpfc.conf` ファイルに以下のオプションを追加します。

```
lpfc_enable_fc4_type=3
```

- **qla2xxx** ドライバーで NVMe/FC を有効にするには、`/etc/modprobe.d/qla2xxx.conf` ファイルに以下のオプションを追加します。

```
qla2xxx.ql2xnvmeenable=1
```

その他の制限:

- NVMe/FC は、マルチパスに対応していません。
- NVMe/FC は、NVMe クラスタリングに対応していません。
- NVMe/FC は、**kdump** に対応していません。
- SAN (Storage Area Network) の NVMe/FC からのシステム起動には対応していません。

(BZ#1649497)

新しい `scan_lvs` 設定

新しい `lvm.conf` 設定ファイルの設定 `scan_lvs` が追加され、デフォルトで 0 に設定されています。新しいデフォルト動作では、LVM が、論理ボリュームに存在する可能性のある物理ボリュームを検索しないようにします。つまり、より多くの物理ボリュームについて、アクティブな論理ボリュームをスキャンしません。デフォルト設定では、LVM が論理ボリュームに物理ボリュームを作成しないようにします。

論理ボリュームに物理ボリュームを重ねるのは、仮想マシンイメージを論理ボリュームに配置する方法で行うことができます。この場合、ホストが物理ボリュームにアクセスするのは安全ではありません。この安全ではないアクセスを回避することが、新しいデフォルト動作の主な理由です。また、アクティブな論理ボリュームが多数ある環境では、LVM が行うデバイススキャンの量が大幅に減ります。

この設定を 1 に設定すると、以前の動作を復元できます。

(BZ#1676598)

DM Multipath 設定ファイルの `overrides` セクションが追加

`/etc/multipath.conf` ファイルには、全デバイスの設定値を設定できる **overrides** セクションが追加されました。この属性は、デバイスを含むパスに対して、`/etc/multipath.conf` ファイルの **multipaths** セクションに指定した属性で上書きした場合を除き、DM Multipath によりすべてのデバイスに使用されます。この機能は、現在是对応していない設定ファイルの **devices** セクションの **all_devs** パラメーターに代わるものです。

(BZ#1643294)

NVDIMM デバイスからのインストールおよび起動に対応

以前は、どのモードでも、NVDIMM (Nonvolatile Dual Inline Memory Module) デバイスがインストーラーから無視されていました。

今回の更新で、NVDIMM デバイスに対応するカーネル実装により、システムパフォーマンスの機能が改善し、データベースまたは解析のワークロードなど、書き込みが集中するアプリケーションへのファイルシステムのアクセスが強化され、CPU オーバーヘッドが削減しました。

今回の更新で、以下に対応するようになりました。

- キックスタートの **nvdimm** コマンドおよび GUI を使用したインストールに NVDIMM デバイスを使用。インストール時に、セクターモードの NVDIMM デバイスからインストールおよび起動して、セクターモードに NVDIMM デバイスに再設定できます。
- NVDIMM デバイスを扱うコマンドで、Anaconda の **Kickstart** スクリプトを拡張。
- システムコンポーネント **grub2**、**efibootmgr**、および **efivar** が、NVDIMM デバイスを処理して起動する機能。

(BZ#1499442)

DM Multipath のマージナルパスの検出が改善

multipathd サービスでは、マージナルパスの検出が改善しました。これにより、マルチパスデバイスが、繰り返し失敗する可能性があるパスを回避して、パフォーマンスを向上します。マージナルパスは、永続的で断続的な I/O エラーがあるパスです。

マージナルパスの動作は、`/etc/multipath.conf` ファイルの以下のオプションで制御します。

- **marginal_path_double_failed_time**
- **marginal_path_err_sample_time**
- **marginal_path_err_rate_threshold**
- **marginal_path_err_recheck_gap_time**

以下の場合、DM Multipath はパスを無効にし、サンプル期間中に繰り返し I/O でテストします。

- **multipath.conf** オプションが設定されている
- 設定した期間内にパスが 2 回失敗する
- その他のパスが利用できる

このテスト時に、パスのエラー率が、設定されたエラー率よりも大きいと、設定したギャップ時間中 DM Multipath がパスを無視し、それが復旧できるぐらいに適切に機能しているかどうかを再確認します。

詳細は、man ページの **multipath.conf** を参照してください。

(BZ#1643550)

ブロックデバイスにおけるマルチキュースケジューリング

Red Hat Enterprise Linux 8 では、ブロックデバイスがマルチキュースケジューリングを使用するようになりました。これにより、高速ソリッドステートドライブ (SSD) およびマルチコアシステムでの拡張が向上します。

RHEL 7 以前のバージョンで利用できた従来のスケジューラーが削除されました。RHEL 8 は、マルチキュースケジューラーのみに対応します。

(BZ#1647612)

5.1.13. 高可用性およびクラスター

新しい pcs コマンドで、利用可能なウォッチドッグデバイスのリストを表示し、ウォッチドッグデバイスをテスト

Pacemaker を使用して SBD を設定するために、動作するウォッチドッグデバイスが必要です。このリリースでは、ローカルノードで利用可能なウォッチドッグデバイスをリスト表示する **pcs stonith sbd watchdog list** コマンドと、ウォッチドッグデバイスをテストする **pcs stonith sbd watchdog test** コマンドに対応します。**sbd** コマンドラインツールの詳細は、man ページの **sbd(8)** を参照してください。

(BZ#1578891)

pcs コマンドで、オペレーションと、その間隔によるリソースのフィルタリングの不具合に対応

Pacemaker は、リソース名およびノードのほかに、リソース操作別にリソースの不具合を追跡するようになりました。したがって、**pcs resource failcount show** コマンドでは、不具合のフィルタリングに、リソース、ノード、操作、および間隔を選択できます。これにより、リソースおよびノード別、もしくはリソース、ノード、操作、および間隔別に細かく集計した不具合を表示するオプションを提供します。また、**pcs resource cleanup** コマンドを使用すれば、リソース、ノード、操作、および間隔で不具合をフィルタリングできます。

(BZ#1591308)

corosync ログでタイムスタンプが有効になっている

以前は、**corosync** ログにタイムスタンプが含まれていなかったため、他のノードやデーモンからのログに関連付けることは困難でした。このリリースでは、タイムスタンプが **corosync** ログに記録されます。

(BZ#1615420)

コマンド pcs cluster setup、pcs cluster node add、pcs cluster node remove の新しい形式

Red Hat Enterprise Linux 8 の **pcs** は、Corosync 3、**knet**、およびノード名に完全に対応します。ノード名は必須となり、ノード識別子となるノードアドレスを置き換えます。ノードアドレスは任意となりました。

- **pcs host auth** コマンドで、ノードアドレスがデフォルトでノード名となります。
- **pcs cluster setup** コマンドおよび **pcs cluster node add** コマンドでは、ノードアドレスが、**pcs host auth** コマンドで指定したノードアドレスにデフォルト設定されます。

この変更により、クラスターのセットアップ、クラスターへのノードの追加、およびクラスターからのノードの削除を行うコマンドの形式が変更になりました。新しいコマンド形式の詳細は、**pcs cluster setup** コマンド、**pcs cluster node add** コマンド、および **pcs cluster node remove** コマンドのヘルプ表示を参照してください。

(BZ#1158816)

新しい pcs コマンド

Red Hat Enterprise Linux 8 には、以下のコマンドが追加されました。

- RHEL 8 に、新しいコマンド **pcs cluster node add-guest | remove-guest** が追加されました。これは、RHEL 7 の **pcs cluster remote-node add | remove** コマンドに代わるものです。
- RHEL 8 に、新しいコマンド **pcs quorum unblock** が追加されました。これは、RHEL 7 の **pcs cluster quorum unblock** コマンドに代わるものです。
- **pcs resource cleanup** コマンドの機能を複製するため、**pcs resource failcount reset** コマンドが削除されました。
- RHEL 8 に、RHEL 7 の **pcs resource [show]** コマンドに代わる新しいコマンドが追加されました。
 - RHEL 8 の **pcs resource [status]** コマンドは、RHEL 7 の **pcs resource [show]** コマンドに代わるものです。
 - RHEL 8 の **pcs resource config** コマンドは、RHEL 7 の **pcs resource [show] --full** コマンドに代わるものです。
 - RHEL 8 の **pcs resource config resource id** コマンドは、RHEL 7 の **pcs resource show resource id** コマンドに代わるものです。
- RHEL 8 には、RHEL 7 の **pcs stonith [show]** コマンドに代わる新しいコマンドが追加されました。
 - RHEL 8 の **pcs stonith [status]** コマンドは、RHEL 7 の **pcs stonith [show]** コマンドに代わるものです。
 - RHEL 8 の **pcs stonith config** コマンドは、RHEL 7 の **pcs stonith [show] --full** コマンドに代わるものです。
 - RHEL 8 の **pcs stonith config resource id** コマンドは、RHEL 7 の **pcs stonith show resource id** コマンドに代わるものです。

(BZ#1654280)

RHEL 8 の Pacemaker 2.0.0

pacemaker パッケージがアップストリームバージョンの Pacemaker 2.0.0 にアップグレードし、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。

- Pacemaker の詳細ログが、(**/var/log** に直接置かれるのではなく、**/var/log/cluster** にある **corosync** ログを組み合わせる) デフォルトで **/var/log/pacemaker/pacemaker.log** になりました。
- ログの読み取りをより直観的にするために、Pacemaker デーモンプロセスの名前が変更になりました。たとえば、**pengine** の名前は **pacemaker-schedulerd** に変更になりました。

- 非推奨のクラスタープロパティ **default-resource-stickiness** および **is-managed-default** に対応しなくなりました。代わりに、**resource-stickiness** プロパティおよび **is-managed** プロパティをリソースのデフォルトに設定する必要があります。非推奨の構文を持つ既存の設定 (新たに作成されたものではありません) は自動的に更新され、対応している構文を使用するようになります。
- 変更の詳細なリストは [Pacemaker 2.0 upgrade in Red Hat Enterprise Linux 8](#) を参照してください。

Red Hat Enterprise Linux 7 以前を使用して既存のクラスターをアップグレードするユーザーは、すべてのクラスターノードで RHEL をアップグレードする前と後に、任意のクラスターノードで **pcs cluster cib-upgrade** を実行することが推奨されます。

(BZ#1543494)

マスターリソースから、昇格可能なクローンリソースに名前が変更

Red Hat Enterprise Linux (RHEL) 8 は Pacemaker 2.0 に対応しています。ここでは、マスター/スレーブのリソースが、別の種類のリソースではなくなり、**promotable** メタ属性が **true** に設定されている標準のクローンリソースになりました。今回の更新に対応するために、以下の変更が行われました。

- **pcs** コマンドでマスターリソースを作成することができなくなりました。代わりに、**昇格可能なクローンリソース**を作成できます。関連キーワードおよびコマンドが、**master** から **promotable** に変わりました。
- 既存のすべてのマスターリソースが昇格可能なクローンリソースとして表示されます。
- RHEL7 クラスターでは昇格可能なクローンに対応しないため、Web UI で RHEL7 クラスターを管理する場合は、マスターリソースが引き続きマスターと呼ばれます。

(BZ#1542288)

クラスターのノードを認証する新しいコマンド

Red Hat Enterprise Linux (RHEL) 8 では、クラスターでノードの認証に使用されるコマンドに以下の変更が加えられています。

- 認証用の新しいコマンドは **pcs host auth** です。このコマンドにより、ホスト名、アドレス、および **pcsd** ポートを指定できます。
- **pcs cluster auth** コマンドは、ローカルクラスターのノードのみを認証し、ノードリストは許可しません、
- 各ノードにアドレスを指定できるようになりました。これにより、**pcs/pcsd** は指定したアドレスを使用して各ノードと通信します。このアドレスは、**corosync** が内部で使用しているアドレスとは異なる場合があります。
- **pcs pcsd clear-auth** コマンドは、**pcs pcsd deauth** コマンドおよび **pcs host deauth** コマンドに置き換えられました。この新しいコマンドを使用すると、ユーザーは全ホストの認証解除だけでなく、1台のホストでも認証解除できます。
- 以前は、ノード認証が双方向で行われており、**pcs cluster auth** コマンドを実行すると、指定したすべてのノードが互いに認証されていました。ただし、**pcs host auth** コマンドを使用すると、指定したノードに対してローカルホストだけが認証されます。これにより、このコマンドを実行する際に、どのノードをどのノードに認証させるかをより適切に制御できます。クラスターを設定する場合や、ノードを追加する際に、**pcs** がクラスターのトークンを自動的に同期するため、クラスター内のすべてのノードが以前と同じように自動的に認証され、クラスターノードが互いに通信できます。

これらの変更は後方互換性がないことに注意してください。RHEL 7 システムで認証されたノードは、再認証する必要があります。

(BZ#1549535)

pcs コマンドで、フェンシング履歴の表示、クリーンアップ、および同期に対応

Pacemaker の fence デーモンは、実行したすべての fence アクションの履歴 (保留、成功、および失敗) を追跡します。このリリースでは、**pcs** コマンドを使用して、ユーザーは次の方法でフェンシング履歴にアクセスできます。

- **pcs status** コマンドは、失敗した保留中のフェンシングアクションを表示します。
- **pcs status --full** コマンドは、フェンシング履歴全体を表示します。
- **pcs stonith history** コマンドは、フェンシング履歴を表示してクリーンアップするオプションを提供します。
- フェンシング履歴は自動的に同期されますが、**pcs stonith history** コマンドで **update** オプションに対応するようになり、必要に応じてフェンシング履歴を手動で同期できるようになりました。

(BZ#1620190, BZ#1615891)

5.1.14. ネットワーク

nftables で、iptables が、デフォルトのネットワークパケットフィルタリングのフレームワークとして置き換え

nftables フレームワークは、パケットの分類機能を提供し、**iptables** ツール、**ip6tables** ツール、**arptables** ツール、および **ebtables** ツールの後継となります。利便性、機能、パフォーマンスにおいて、以前のパケットフィルタリングツールに多くの改良が追加されました。以下に例を示します。

- 線形処理の代わりにルックアップテーブルを使用
- **IPv4** プロトコルおよび **IPv6** プロトコルに対する1つのフレームワーク
- 完全ルールセットのフェッチ、更新、および保存を行わず、すべてアトミックに適用されるルール
- ルールセットにおけるデバッグおよびトレースへの対応 (**nfttrace**) およびトレースイベントの監視 (**nft** ツール)
- より統一されたコンパクトな構文、プロトコル固有の拡張なし
- サードパーティーのアプリケーション用 Netlink API

iptables と同様、**nftables** は、チェーンを保存するテーブルを使用します。このチェーンには、アクションを実行する個々のルールが含まれます。**nft** ツールは、以前のパケットフィルタリングフレームワークのツールをすべて置き換えます。**libnftables** ライブラリーは、**libmnl** ライブラリーの Netlink API **nftables** で、低レベルの対話のために使用できます。

iptables ツール、**ip6tables** ツール、**ebtables** ツール、および **arptables** ツールは、**nftables** ベースの同じ名前のドロップインツールに置き換えられました。外部の挙動は従来のもと同じですが、内部的には必要に応じて互換インターフェイスを通して、従来の **netfilter** カーネルモジュールを使用した **nftables** を使用します。

nftables ルールセットに対するモジュールの効果は、**nft list ruleset** コマンドを使用して確認できます。これらのツールは、テーブル、チェーン、およびルールを **nftables** ルールセットに追加するため、**nft flush ruleset** コマンドなどの **nftables** ルールセット操作は、先に別の従来のコマンドを使用してインストールしたルールセットに影響を及ぼす可能性があることに注意してください。

どの種類のツールが存在するかをすばやく特定するために、バージョン情報にバックエンド名が追加されるようになりました。RHEL 8 では、**nftables** ベースの **iptables** ツールで、次のバージョン文字列が出力されます。

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

一方、従来の **iptables** ツールが存在する場合は、次のバージョン情報が出力されます。

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

(BZ#1644030)

RHEL 8 における TCP の新機能

Red Hat Enterprise Linux 8 には、TCP ネットワーキングスタックバージョン 4.18 が同梱され、より高いパフォーマンスおよび安定性と、より優れたスケラビリティが提供されます。特に、入力接続率が高い、ビジー状態の TCP サーバーのパフォーマンスが向上します。

また、2つの新しい TCP 輻輳制御アルゴリズム (**BBR** および **NV**) が利用でき、ほとんどのシナリオで、キュービックよりも待ち時間が少なく、スループットも良くなります。

(BZ#1562998)

firewalld がデフォルトで nftables を使用

今回の更新で、**nftables** フィルタリングサブシステムが、**firewalld** デーモンのデフォルトのファイアウォールバックエンドになります。バックエンドを変更するには、`/etc/firewalld/firewalld.conf` ファイルの **FirewallBackend** オプションを使用します。

この変更により、**nftables** を使用した時の動作に以下のような差異が生じます。

1. **iptables** ルールの実行は、常に **firewalld** ルールの前に行われます。
 - **iptables** の **DROP** は、**firewalld** からパケットを隠すことを意味します。
 - **iptables** の **ACCEPT** は、パケットが引き続き **firewalld** ルールに準拠することを意味します。
2. 他の **firewalld** 機能は **nftables** を使用しますが、**firewalld** ダイレクトルールは、**iptables** を介して実装されます。
3. ダイレクトルールの実行は、確立された接続に行われる **firewalld** の一般的な承認の前に行われます。

(BZ#1509026)

RHEL 8 の wpa_supplicant への主な変更点

Red Hat Enterprise Linux (RHEL) 8 の **wpa_supplicant** パッケージは、**CONFIG_DEBUG_SYSLOG** が有効になった状態で構築されています。これにより、`/var/log/wpa_supplicant.log` ファイルの内容を

確認する代わりに、`journalctl` ユーティリティーを使用して `wpa_supplicant` ログを読み取ることができます。

(BZ#1582538)

NetworkManager で SR-IOV 仮想関数への対応

Red Hat Enterprise Linux 8.0 では、**NetworkManager** で、SR-IOV (Single Root I/O virtualization) に対応するインターフェイスに対する仮想関数の数を設定できます。また、**NetworkManager** では、MAC アドレス、VLAN、**spoof checking** 設定、許可されるビットレートなど、仮想関数の属性の一部を設定できます。SR-IOV に関連するプロパティーはすべて、**sriov** 接続の設定で利用できます。詳細は、man ページの **nm-settings(5)** を参照してください。

(BZ#1555013)

IPVLAN 仮想ネットワークドライバへの対応

Red Hat Enterprise Linux 8.0 のカーネルには、IPVLAN 仮想ネットワークドライバへの対応が含まれます。今回の更新で、IPVLAN の仮想ネットワークインターフェイスカード (NIC) により、ローカルネットワークに1つの MAC アドレスを公開する複数のコンテナへのネットワーク接続が可能になります。これにより、ピアネットワーク機器で対応している MAC アドレス数の制限を改善するために、1つのホストで多数のコンテナを使用できるようになります。

(BZ#1261167)

NetworkManager で、接続に対するワイルドカードインターフェイスの名前一致に対応

以前は、インターフェイス名の完全一致のみを使用して、特定のインターフェイスへの接続を制限していました。今回の更新で、接続には、ワイルドカードに対応する新しい **match.interface-name** プロパティーが含まれるようになりました。この更新により、ワイルドカードパターンを使用した柔軟な方法で、接続用インターフェイスを選択できるようになります。

(BZ#1555012)

Networking Stack 4.18 の改良

Red Hat Enterprise Linux 8.0 には、アップストリームバージョン 4.18 にアップグレードされたネットワークスタックが含まれています。以下は、主な変更点です。

- **UDP_GSO** などの新しいオフロード機能、および一部のデバイスドライバに **GRO_HW** が追加されました。
- ユーザーデータグラムプロトコル (UDP) のスケーラビリティが大幅に向上しました。
- 汎用ビジーポーリングコードを改善しました。
- IPv6 プロトコルのスケーラビリティが改善しました。
- ルーティングコードのスケーラビリティが改善しました。
- 新しいデフォルト送信キュースケジューリングアルゴリズム **fq_codel** が追加されました。転送遅延が改善されます。
- 一部の送信キュースケジューリングアルゴリズムのスケーラビリティが向上しました。たとえば、**pfifo_fast** がロックされなくなりました。
- IP 再構築ユニットのスケーラビリティの強化 - ガベージコレクションのカーネルスレッドおよび IP フラグメントは、タイムアウト時にのみ有効期限が切れます。したがって、DoS 下の CPU 使用率は大幅に低く、永続可能なフラグメントの最大破棄率は IP 再設定ユニットに設定

されたメモリ一量により制限されます。

(BZ#1562987)

iptables を nftables に変換する新しいツール

今回の更新で、既存の **iptables** ルールまたは **ip6tables** ルールを、**nftables** で同等のルールに変換する **iptables-translate** ツールおよび **ip6tables-translate** ツールが追加されました。拡張機能によっては変換機能がない場合もあります。対応する機能がない拡張機能が存在する場合は、ツールにより、その前に **#** 記号が付いた未変換ルールが出力されます。以下に例を示します。

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

また、ユーザーは、**iptables-restore-translate** ツールおよび **ip6tables-restore-translate** ツールを使用して、ルールのダンプを変換できます。その前に、**iptables-save** コマンドまたは **ip6tables-save** コマンドを使用して、現在のルールのダンプを出力できます。以下に例を示します。

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

(BZ#1564596)

NetworkManager により VPN に追加された新機能

Red Hat Enterprise Linux 8.0 の **NetworkManager** は、VPN に以下の新機能を提供します。

- インターネット鍵交換バージョン 2 (IKEv2) プロトコルへの対応
- **rightid**、**leftcert**、**narrowing**、**rekey**、**fragmentation** などの **Libreswan** オプションが追加されました。対応しているオプションの詳細は、man ページの **nm-settings-libreswan** を参照してください。
- デフォルトの暗号を更新しました。つまり、暗号化方式を指定しないと、**NetworkManager-libreswan** プラグインにより、**Libreswan** アプリケーションがシステムのデフォルト暗号を選択します。ただし、ユーザーが IKEv1 アグレッシブモード設定を選択した場合は異なります。この場合は、**ike = aes256-sha1;modp1536** 値および **eps = aes256-sha1** 値が **Libreswan** に渡されます。

(BZ#1557035)

新しいデータチャンクタイプ I-DATA が SCTP に追加

今回の更新で、新しいデータチャンクタイプ (**I-DATA**)、およびストリームスケジューラーが SCTP (Stream Control Transmission Protocol) に追加されました。以前は、ユーザーが送信する順序で SCTP がユーザーメッセージを送信していました。このため、SCTP ユーザーメッセージが大きくなると、送信が完了するまで、ストリーム内の他のすべてのメッセージがブロックされていました。**I-DATA** チャンクを使用している場合は、Transmission Sequence Number (TSN) フィールドがオーバーロードされません。そのため、SCTP ではさまざまな方法でストリームをスケジューリングできるようになり、**I-DATA** ではユーザーメッセージのインターリーブ (RFC 8260) が可能になりました。両方のピアが **I-DATA** チャンクタイプに対応する必要があることに注意してください。

(BZ#1273139)

NetworkManager が、ethtool オフロード機能の設定に対応

この機能強化により、**NetworkManager** は、**ethtool** オフロード機能の設定に対応するため、init スクリプトまたは **NetworkManager** ディスパッチャースクリプトを使用しなくなりました。その結果、以下のいずれかの方法を使用して、接続プロファイルの一部としてオフロード機能を設定できるようになりました。

- **nmcli** ユーティリティーを使用
- `/etc/NetworkManager/system-connections/` ディレクトリーの鍵ファイルを編集
- `/etc/sysconfig/network-scripts/ifcfg-*` ファイルを編集

この機能は、現在、グラフィカルインターフェイスと **nmtui** ユーティリティーでは対応していないことに注意してください。

(BZ#1335409)

RHEL 8 における TCP BBR への対応

Red Hat Enterprise Linux (RHEL) 8 では、新しい TCP 輻輳制御アルゴリズムである BBR (Bottleneck Bandwidth and Round-trip time) に対応するようになりました。BBR は、ボトルネックリンクおよびラウンドトリップ時間 (RTT) の帯域幅を決定します。ほとんどの輻輳アルゴリズムは、パケットロス (デフォルトの Linux TCP 輻輳制御アルゴリズムである CUBIC を含む) に基づいており、高スループットのリンクの問題をかかえています。BBR は、損失イベントに直接反応せず、利用可能な帯域幅をそれと一致させて、TCP ペーシングレートを調整します。TCP BBR を使用する場合は、関係するすべてのインターフェイスの **fq** キュー設定に切り替える必要があります。

ユーザーが明示的に **fq** を使用し、**fq_codel** は使用しないことに注意してください。

詳細は、man ページの **tc-fq** を参照してください。

(BZ#1515987)

RHEL 8 における **lksctp-tools** (バージョン 1.0.18)

lksctp-tools パッケージ (バージョン 3.28) は Red Hat Enterprise Linux (RHEL) 8 で利用できます。以下は、主な機能強化およびバグ修正です。

- Travis CI および Coverity Scan との統合
- **sctp_peeloff_flags** 関数への対応
- 利用可能なカーネル機能の表示
- Coverity Scan の問題に対する修正

(BZ#1568622)

RHEL 8 のデフォルトでの SCTP モジュールのブラックリスト登録

セキュリティを強化するために、一連のカーネルモジュールが **kernel-modules-extra** パッケージに移動しました。デフォルトではインストールされません。これにより、root 以外のユーザーは、デフォルトでブラックリストに指定されているため、これらのコンポーネントを読み込むことができません。これらのカーネルモジュールを使用するには、システム管理者が **kernel-modules-extra** をインストールし、モジュールのブラックリストを明示的に削除する必要があります。これにより、root 以外のユーザーは、自動的にソフトウェアコンポーネントを読み込むことができます。

(BZ#1642795)

driverctl 0.101 への主な変更点

Red Hat Enterprise Linux 8.0 には、**driverctl** 0.101 が同梱されています。このバージョンは以下のバグ修正が含まれます。

- **shellcheck** の警告が修正されました。
- bash-completion は、**driverctl-bash-completion.sh** の代わりに **driverctl** としてインストールされます。
- 非 PCI バスの **load_override** 関数が修正されました。
- **driverctl** サービスは、systemd ターゲットの **basic.target** に到達する前に、すべてのオーバーライドを読み込みます。

(BZ#1648411)

firewalld へのリッチルール優先度の追加

リッチルールに **priority** オプションが追加されました。これにより、ルール実行時に好ましい優先順位を定義し、リッチルールに対するより高度な制御が可能になります。

(BZ#1648497)

RHEL 8 で NVMe over RDMA に対応

Red Hat Enterprise Linux (RHEL) 8 の RDMA (Remote Direct Memory Access) の NVMe (Nonvolatile Memory Express) では、イニシエーターモードでのみ Infiniband、RoCEv2、および iWARP に対応します。

マルチパスは、フェイルオーバーモードでのみ対応していることに注意してください。

その他の制限:

- Kdump は、NVMe/RDMA では対応しません。
- RDMA での NVMe デバイスからの起動には対応していません。

(BZ#1680177)

nf_tables バックエンドが、dmesg を使用したデバッグに対応していない

Red Hat Enterprise Linux 8.0 では、**dmesg** ユーティリティの出力を使用したファイアウォールのデバッグに対応しないファイアウォールに、**nf_tables** バックエンドを使用します。ファイアウォールルールをデバッグするには、**xtables-monitor -t** コマンドまたは **nft monitor trace** コマンドを使用して、ルール評価イベントをデコードします。

(BZ#1645744)

Red Hat Enterprise Linux が VRF に対応

RHEL 8.0 のカーネルは、仮想ルーティングおよび転送 (VRF) に対応します。管理者は、VRF デバイスを、**ip** ユーティリティを使用して設定したルールと組み合わせて、Linux ネットワークスタックに VRF ドメインを作成できます。これらのドメインはレイヤー 3 のトラフィックを分離するため、管理者は異なるルーティングテーブルを作成し、ホストにあるさまざまな VRF ドメインで同じ IP アドレスを再利用できます。

(BZ#1440031)

RHEL 8 の iproute バージョン 4.18

iproute パッケージには、Red Hat Enterprise Linux (RHEL) 8 のバージョン 4.18 が同梱されています。最も注目すべき変更は、ethX:Y とマークされたインターフェイスエイリアス (eth0:1 など) に対応しなくなったことです。この問題を回避するには、**ip link show** に移動する前に、エイリアスの接尾辞 (コロンと、その後続く番号) を削除する必要があります。

(BZ#1589317)

5.1.15. セキュリティー

RHEL 8.0 リリースの SWID タグ

ISO/IEC 19770-2:2015 メカニズムを使用して RHEL 8.0 インストールの識別を可能にするために、`/usr/lib/swidtag/redhat.com/com.redhat.RHEL-8-<architecture>.swidtag` ファイルおよび `/usr/lib/swidtag/redhat.com/com.redhat.RHEL-8.0-<architecture>.swidtag` ファイルに、ソフト識別 (SWID) タグがインストールされます。このタグの親ディレクトリーも、`/etc/swid/swidtags.d/redhat.com` シンボリックリンクからたどることができます。

SWID タグファイルの XML 署名は、**xmlsec1 verify** コマンドを使用して確認できます。以下に例を示します。

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

コード署名認証局の証明書は、カスタマーポータル[の製品署名キー](#) ページからも入手できます。

(BZ#1636338)

システム全体の暗号化ポリシーがデフォルトで適用

暗号化ポリシーは、Red Hat Enterprise Linux 8 でコア暗号化サブシステムを設定するコンポーネントで、TLS、IPsec、DNSSEC、Kerberos、および SSH の各プロトコルに対応します。これにより、管理者が **update-crypto-policies** コマンドで選択できる小規模セットのポリシーを提供します。

デフォルト のシステム全体の暗号化ポリシーは、現在の脅威モデルに安全な設定を提供します。TLS プロトコル 1.2 と 1.3、IKEv2 プロトコル、および SSH2 プロトコルが使用できます。RSA 鍵と Diffie-Hellman パラメーターの長さが 2047 ビット以上であれば許可されます。

詳細は、Red Hat ブログの記事 [Consistent security by crypto policies in Red Hat Enterprise Linux 8](#) と、man ページの **update-crypto-policies(8)** を参照してください。

(BZ#1591620)

OpenSSH がバージョン 7.8p1 にリベース

openssh パッケージが、アップストリームバージョン 7.8p1 にアップグレードされました。以下は、主な変更点です。

- **SSH version 1** プロトコルに対応しなくなりました。
- **hmac-ripemd160** メッセージ認証コードに対応しなくなりました。
- RC4 (**arcfour**) 暗号に対応しなくなりました。
- **Blowfish** 暗号に対応しなくなりました。

- **CAST** 暗号に対応しなくなりました。
- **UseDNS** オプションのデフォルト値が **no** に変更になりました。
- **DSA** 公開鍵アルゴリズムがデフォルトで無効になりました。
- **Diffie-Hellman** パラメーターの最小モジュールサイズが 2048 ビットに変更になりました。
- **ExposeAuthInfo** 設定オプションのセマンティクスが変更になりました。
- **UsePrivilegeSeparation=sandbox** オプションが必須になったため、無効にできなくなりました。
- 許可される最小の **RSA** キーサイズが 1024 ビットに設定されました。

(BZ#1622511)

OpenSSH サーバーの鍵の自動生成を `sshd-keygen@.service` が処理

サーバーホストの鍵の RSA、ECDSA、および ED25519 がない場合は、**OpenSSH** が自動的に作成します。RHEL 8 でホストの鍵の作成を設定するには、インスタンス化したサービス `sshd-keygen@.service` を使用します。

たとえば、RSA 鍵タイプの自動作成を無効にするには、次のコマンドを実行します。

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

詳細は `/etc/sysconfig/ssh` ファイルを参照してください。

(BZ#1228088)

SSH 認証で ECDSA 鍵に対応

このリリースの **OpenSSH** スイートでは、PKCS #11 スマートカードに保存されている ECDSA 鍵に対応するようになりました。これにより、SSH 認証に RSA 鍵と ECDSA 鍵の両方を使用できるようになりました。

(BZ#1645038)

libssh が SSH をコア暗号化コンポーネントとして実装

この変更により、Red Hat Enterprise Linux 8 のコア暗号化コンポーネントとして **libssh** が追加されました。**libssh** ライブラリーに Secure Shell (SSH) プロトコルが実装されました。

libssh のクライアント側は、システム全体の暗号化ポリシーを介して **OpenSSH** の設定に従いますが、サーバー側の設定はシステム全体の暗号化ポリシーでは変更できないことに注意してください。

(BZ#1485241)

暗号化ライブラリーにおける TLS 1.3 への対応

今回の更新で、主要なすべてのバックエンド暗号ライブラリーで、TLS (Transport Layer Security) 1.3 が有効になっています。これにより、オペレーティングシステムの通信層でのレイテンシーを短縮し、RSA-PSS、X25519 などの新しいアルゴリズムを使用して、アプリケーションのプライバシーとセキュリティを強化します。

(BZ#1516728)

NSS がデフォルトで SQL を使用

Network Security Services (NSS) ライブラリーは、デフォルトで信頼データベースに SQL ファイル形式を使用するようになりました。以前のリリースでデフォルトのデータベース形式として使用されていた DBM ファイル形式は、マルチプロセスによる同時アクセスに対応せず、アップストリームでは非推奨となっていました。したがって、NSS 信頼データベースを使用するアプリケーションが鍵や証明書を保存し、取消し情報がデフォルトで SQL 形式のデータベースを作成します。レガシーの DBM 形式でデータベースを作成しようとする試みは失敗します。既存の DBM データベースを読み取り専用モードで開くと、自動的に SQL 形式に変換されます。Red Hat Enterprise Linux 6 以降、NSS では SQL ファイル形式に対応します。

(BZ#1489094)

スマートカードおよび HSM の PKCS #11 がシステム全体で統一

今回の更新で、PKCS #11 の暗号化トークンインターフェイスを使用するスマートカードおよび HSM (Hardware Security Modules) の使用方法が統一されました。つまり、ユーザーおよび管理者は、システムに関連するすべてのツールに同じ構文を使用できるようになりました。以下は、主な機能強化です。

- 管理者およびアプリケーション作成者用に、RHEL サーバーで、トークンの簡略化された使用可能性を保証する PKCS #11 Uniform Resource Identifier (URI) スキームに対応する。
- システム全体で、スマートカードと HSM の登録に **pkcs11.conf** を使用する。
- NSS、GnuTLS、および OpenSSL (**openssl-pkcs11** エンジンを使用) のアプリケーションで利用できる、HSM およびスマートカードに対する対応が統一される。
- Apache HTTP サーバー (**httpd**) が、HSM にシームレスに対応する。

詳細は、man ページの **pkcs11.conf(5)** を参照してください。

(BZ#1516741)

Firefox が、システム全体に登録された PKCS #11 ドライバーと連携

Firefox Web ブラウザーは **p11-kit-proxy** モジュールを自動的に読み込み、**pkcs11.conf** ファイルを介して **p11-kit** でシステム全体に登録されるすべてのスマートカードを自動的に検出します。TLS クライアント認証を使用した場合、その他に必要な設定はありません。また、サーバーがスマートカードを要求する際に、スマートカードの鍵が自動的に使用されます。

(BZ#1595638)

OpenSC で RSA-PSS に対応

今回の更新で、**OpenSC** スマートカードドライバーに、RSA-PSS 暗号化署名スキームへの対応が追加されました。新しいスキームでは、クライアントソフトウェアでの TLS 1.3 への対応に必要な安全な暗号化アルゴリズムが有効になります。

(BZ#1595626)

RHEL 8 の Libreswan への主な変更点

libreswan パッケージがアップストリームバージョン 3.27 にアップグレードし、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。以下は、主な変更点です。

- **authby=rsa-sha2** を介した RSA-PSS (RFC 7427) への対応、**authby=ecdsa-sha2** を介した ECDSA (RFC 7427) への対応、**dh31** キーワードを使用した CURVE25519 への対応、**chacha20_poly1305** 暗号化キーワードを介した IKE および ESP の CHACHA20-

POLY1305 が、IKEv2 プロトコルに追加されました。

- アップストリームは、KLIPS を完全に非推奨としているため、**Libreswan** で、代替の KLIPS カーネルモジュールに対応しなくなりました。
- (RFC 8247 によると) Diffie-Hellman グループ DH22、DH23、および DH24 に対応しなくなりました。

authby=rsasig は、常に RSA v1.5 メソッドを使用するように変更されており、**authby=rsa-sha2** オプションは RSASSA-PSS メソッドを使用することに注意してください。RFC 8247 によると、**authby=rsa-sha1** オプションは有効ではありません。これは、**Libreswan** が、デジタル署名のある SHA-1 に対応しなくなるためです。

(BZ#1566574)

システム全体の暗号化ポリシーが、Libreswan のデフォルトの IKE バージョンを IKEv2 に変更

Libreswan IPsec 実装のデフォルト IKE バージョンは、IKEv1 (RFC 2409) から IKEv2 (RFC 7296) に変更になりました。IPsec で使用するデフォルトの IKE および ESP/AH のアルゴリズムが、システム全体の暗号化ポリシーである RFC 8221 および RFC 8247 に準拠するように更新されました。256 ビットの暗号鍵サイズは、128 ビットの鍵サイズよりも推奨されるようになりました。

デフォルトの IKE および ESP/AH の暗号には、暗号用の AES-GCM、CHACHA20POLY1305、および AES-CBC が含まれます。整合性の確認には、AEAD および SHA-2 を提供します。Diffie-Hellman グループに、DH19、DH20、DH21、DH14、DH15、DH16、および DH18 が含まれるようになりました。

AES_CTR、3DES、SHA1、DH2、DH5、DH22、DH23、および DH24 の各アルゴリズムは、デフォルトの IKE ポリシーおよび ESP/AH ポリシーから削除されました。DH22、DH23、および DH24 を除いたアルゴリズムは、IPsec 設定ファイルの **ike=** オプションまたは **phase2alg=esp=ah=** オプションで有効にできます。

IKEv1 プロトコルを必要とする IPsec VPN 接続を設定するには、接続設定ファイルに **ikev2=no** オプションを追加します。詳細は、man ページの **ipsec.conf(5)** を参照してください。

(BZ#1645606)

Libreswan における IKE バージョン関連の変更

この機能強化により、Libreswan は、異なるインターネット鍵交換 (IKE) 設定を処理します。

- デフォルトのインターネット鍵交換 (IKE) のバージョンが 1 から 2 に変更になりました。
- 接続で IKEv1 プロトコルまたは IKEv2 プロトコルが使用できるようになりましたが、両方を使用することはできません。
- **ikev2** オプションの解釈が変更しました。
 - **insist** の値は IKEv2 専用として解釈されます。
 - **no** および **never** の値は IKEv1 専用として解釈されます。
 - **propose**、**yes**、および **permit** の値は有効ではなくなり、この値から生成された IKE のバージョンが明確ではないため、エラーが発生します。

(BZ#1648776)

RHEL 8 の OpenSCAP における新機能

OpenSCAP スイートがアップストリームバージョン 1.3.0 にアップグレードし、以前のバージョンに対する機能拡張が数多く追加されました。以下は、主な変更点です。

- API と ABI が統合され、更新、非推奨、未使用のシンボルが削除されています。
- プローブは、独立したプロセスとしてではなく、**oscap** プロセス内のスレッドとして実行します。
- コマンドラインインターフェイスが更新しました。
- **Python 2** バインディングが、**Python 3** バインディングに置き換えられました。

(BZ#1614273)

SCAP セキュリティーガイドがシステム全体の暗号化ポリシーに対応

scap-security-guide パッケージが更新され、コア暗号化サブシステムの設定に、事前定義のシステム全体の暗号化ポリシーを使用するようになりました。システム全体の暗号化ポリシーと競合する、または上書きしたセキュリティコンテンツが削除されました。

この変更は、**scap-security-guide** のセキュリティコンテンツにのみ適用され、OpenSCAP スキャナーやその他の SCAP コンポーネントを更新する必要がないことに注意してください。

(BZ#1618505)

OpenSCAP コマンドラインインターフェイスが改善

oscap モジュールおよびサブモジュールで、詳細モードが利用できるようになりました。ツールの出力で形式が改善しました。

コマンドラインインターフェイスの使いやすさを向上させるために、非推奨のオプションが削除されました。

次のオプションは使用できなくなりました。

- **oscap xccdf generate report** の **--show** は完全に削除されました。
- **oscap oval eval** の **--probe-root** は削除されました。代わりに、環境変数 **OSCAP_PROBE_ROOT** を設定できます。
- **oscap xccdf eval** の **--sce-results** は、**--check-engine-results** に置き換えられました。
- **validate-xml** サブモジュールが、CPE モジュール、OVAL モジュール、および XCCDF モジュールから削除されました。**validate** サブモジュールを使用して、XML スキーマおよび XSD スキーマトロンに対して SCAP コンテンツを検証できます。
- **oscap oval list-probes** コマンドが削除されました。代わりに、**oscap --version** を使用して、利用可能なプローブのリストを表示できます。

OpenSCAP では、**--profile 'all'** を使用して、プロファイルに関係なく、特定の XCCDF ベンチマーク上のルールをすべて評価できます。

(BZ#1618484)

SCAP セキュリティーガイド PCI-DSS プロファイルがバージョン 3.2.1 に準拠

scap-security-guide パッケージが、Red Hat Enterprise Linux 8 の PCI-DSS (Payment Card Industry Data Security Standard) プロファイルを提供し、最新の PCI-DSS バージョン (3.2.1) に合わせて更新されました。

(BZ#1618528)

SCAP セキュリティーガイドが OSPP 4.2 に対応

scap-security-guide パッケージが、Red Hat Enterprise Linux 8 用の OSPP (Protection Profile for General purpose Operating Systems) プロファイルバージョン 4.2 のドラフトを提供します。このプロファイルは、汎用オペレーティングシステムの保護プロファイル (プロテクションプロファイルバージョン 4.2) の NIAP 設定付録に記載されている必須の設定管理を反映しています。SCAP セキュリティーガイドは、OSPP に定義されている要件をユーザーが満たせるように、自動化された確認とスクリプトを提供します。

(BZ#1618518)

RHEL 8 の rsyslog への主な変更点

rsyslog パッケージがアップストリームバージョン 8.37.0 にアップグレードし、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。以下は、主な変更点です。

- **rsyslog** の内部メッセージの処理が強化されました。レートを制限できるようになり、デッドロックの可能性が修正されました。
- 一般的に強化されたレート制限。実際の **スパムの送信元** がログに記録されました。
- 特大サイズのメッセージ処理が改善されました。ユーザーは、コアと特定モジュールの両方で、別々の操作でメッセージを処理する方法を設定できるようになりました。
- 個別のファイルを作成する代わりに、**mmnormalize** ルールベースが **config** ファイルに組み込まれるようになりました。
- JSON の変数を含むすべての **config** 変数は、大文字と小文字を区別しません。
- PostgreSQL への出力にさまざまな改善が追加されました。
- 追加の設定ファイルの条件付き読み込み、ステートメントの実行、**config** へのテキストの挿入など、**config** 処理を制御するシェル変数を使用できるようになりました。この機能を過度に使用すると、**rsyslog** での問題のデバッグが非常に困難になる可能性があることに注意してください。
- **config** で、4 桁のファイル作成モードを指定できるようになりました。
- RELP (Reliable Event Logging Protocol) の入力、指定したアドレスにのみバインドできるようになりました。
- メール出力の **enable.body** オプションのデフォルト値が、ドキュメントに合わせて変更になりました。
- ユーザーは、MongoDB 出力で無視する必要がある挿入エラーコードを指定できるようになりました。
- Parallel TCP (pTCP) 入力には、より良い負荷分散に、設定可能なバックログがあります。
- **journald** がそのファイルをローテートすると発生する可能性がある重複レコードを回避するには、**imjournal** オプションが追加されています。このオプションを使用するとパフォーマンスに影響を及ぼします。

ナレッジベースの記事 [Configuring system logging without journald](#) に記載されるように、システムに **rsyslog** を設定すると、パフォーマンスが向上します。

(BZ#1613880)

新しい rsyslog モジュール: omkafka

kafka 集中型データストレージシナリオを有効にするには、新しい **omkafka** モジュールを使用して、kafka インフラストラクチャーにログを転送できるようになりました。

(BZ#1542497)

rsyslog imfile でシンボリックリンクへの対応

今回の更新で、rsyslog imfile モジュールのパフォーマンスが向上し、設定オプションが増えました。これにより、より複雑なファイルにこのモジュールを使用してユースケースを監視できるようになりました。たとえば、設定されたパスのいずれの場所にも、グロブパターンを使用したファイルのモニターを使用し、データのスループットが増えたときに、シンボリックリンクのターゲットをローテートできるようになりました。

(BZ#1614179)

デフォルトの rsyslog 設定ファイルの形式がレガシーではない

rsyslog パッケージの設定ファイルが、デフォルトで非レガシー形式を使用するようになりました。レガシー形式も使用できますが、現在の設定ステートメントとレガシーの設定ステートメントを混在する場合は制約がいくつかあります。以前の RHEL リリースから引き継がれた設定は修正する必要があります。詳細は、man ページの **rsyslog.conf(5)** を参照してください。

(BZ#1619645)

Audit 3.0 で audispd が auditd に置き換え

今回の更新で、audispd 機能が auditd に移行しました。したがって、plugins.d 設定オプションが auditd.conf に追加されました。また、plugins.d ディレクトリーが、/etc/audit に移動しました。auditd と、そのプラグインの現在のステータスは、service auditd state コマンドを実行すれば確認できます。

(BZ#1616428)

tangd_port_t で Tang のデフォルトポートの変更が可能に

今回の更新で、SELinux Enforcing モードで制限のある tangd サービスの実行を可能にする SELinux タイプ tangd_port_t が追加されました。この変更により、Tang サーバーを設定してユーザー定義のポートをリッスンするのを簡素化し、SELinux が提供するセキュリティーレベルを Enforcing モードで維持します。

詳細は [ポリシーベースの複号を使用して暗号化ボリュームの自動アンロックの設定](#) を参照してください。

(BZ#1664345)

新しい SELinux ブール値

今回の SELinux システムポリシーの更新により、以下のブール値が追加されました。

- colord_use_nfs
- mysql_connect_http
- pdns_can_network_connect_db

- `ssh_use_tcpd`
- `sslh_can_bind_any_port`
- `sslh_can_connect_any_port`
- `virt_use_pcsd`

ブール値のリストとその意味を取得し、有効かどうかを調べるには、**selinux-policy-devel** パッケージをインストールして、以下のコマンドを実行します。

```
# semanage boolean -l
```

(JIRA:RHELPLAN-10347)

SELinux で `systemd` の `No New Privileges` に対応

今回の更新で、新旧のコンテキスト間で `nnp_nosuid_transition` が許可されている場合に、**No New Privileges** (NNP) または `nosuid` で SELinux ドメインの移行を可能にする `nnp_nosuid_transition` ポリシー機能が追加されました。**selinux-policy** パッケージに、**NNP** セキュリティー機能を使用する `systemd` サービスのポリシーが含まれるようになりました。

次のルールでは、サービスにこの機能を許可しています。

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

以下に例を示します。

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

ディストリビューションポリシーには、`m4` マクロインターフェイスも含まれています。これは、`init_nnp_daemon_domain()` 関数を使用するサービスの SELinux セキュリティーポリシーで使用できます。

(BZ#1594111)

`mmap` システムコールに対する新しいマップパーミッションの確認への対応

SELinux の `map` パーミッションは、ファイル、ディレクトリー、ソケットなどへのメモリーマップアクセスを制御するために追加されました。これにより、SELinux ポリシーによって、メモリーが、さまざまなファイルシステムオブジェクトに直接アクセスできないようになり、そのようなすべてのアクセスが再検証されるようになります。

(BZ#1592244)

SELinux で `process` クラスの `getrlimit` アクセス権に対応

今回の更新で、新しい SELinux アクセス制御確認 (`process:getrlimit`) が追加されました。これは、`prlimit()` 関数に追加されました。これにより、SELinux ポリシーの開発者は、`process:setrlimit` パーミッションを使用して、あるプロセスが別のプロセスのリソース制限を読み込み、変更するかどうかを制御できます。プロセスが `prlimit()` を介して、自身のリソース制限を操作することは、SELinux が制限しないことに注意してください。詳細は、`man` ページの `prlimit(2)` および `getrlimit(2)` を参照してください。

(BZ#1549772)

selinux-policy が VxFS ラベルに対応

今回の更新で、VxFS (Veritas File System) セキュリティー拡張属性 (xattrs) に対応するようになりました。これにより、汎用の vxfs_t タイプではなく、ファイルシステムのオブジェクトで適切な SELinux ラベルを保存できます。これにより、SELinux に完全に対応する VxFS を使用するシステムの安全性が向上します。

(BZ#1483904)

コンパイル時のセキュリティー強化フラグの適用で一貫性が増す

コンパイル時のセキュリティーの強化フラグは、RHEL 8 ディストリビューションの RPM パッケージでより一貫して適用され、**redhat-rpm-config** パッケージによりセキュリティーの強化フラグが自動的に提供されるようになりました。適用されたコンパイル時フラグは、Common Criteria (CC) 要件を満たすのに役立ちます。以下のセキュリティー強化フラグが適用されます。

- バッファオーバーフローエラーの検出 - **D_FORTIFY_SOURCE=2**
- C++ アレイ、ベクトル、および文字列の確認を強化している標準ライブラリー - **D_GLIBCXX_ASSERTIONS**
- Stack Smashing Protector (SSP) の場合 - **fstack-protector-strong**
- 例外の強化の場合 - **fexceptions**
- Control-Flow Integrity (CFI) の場合 - **fcf-protection=full** (AMD および Intel 64 ビットのアーキテクチャーのみ)
- Address Space Layout Randomization (ASLR) の場合 - **fPIE** (実行ファイル用) または **fPIC** (ライブラリー用)
- スタッククラッシュの脆弱性に対する保護 - **fstack-clash-protection** (ARM を除く)
- システムの起動時にすべてのシンボルを解決するフラグにリンク (**-WI**、**-z,now**)

詳細は、man ページの **gcc(1)** を参照してください。

(JIRA:RHELPLAN-2306)

5.1.16. 仮想化

RHEL 8 の qemu-kvm 2.12

Red Hat Enterprise Linux 8 には **qemu-kvm** 2.12 が同梱されています。このバージョンでは複数のバグが修正され、Red Hat Enterprise Linux 7 で利用可能なバージョン 1.5.3 に機能拡張が数多く追加されました。

具体的には、以下の機能が追加されました。

- Q35 ゲストマシンのタイプ
- UEFI ゲストブート
- ゲストの NUMA チューニングとピンニング
- vCPU のホットプラグとホットアンプラグ
- ゲストの I/O スレッド

qemu-kvm 2.12 で利用可能な機能の一部は、Red Hat Enterprise Linux 8 では対応していません。詳細は、Red Hat カスタマーポータルでの [Feature support and limitations in RHEL 8 virtualization](#) を参照してください。

(BZ#1559240)

仮想化システムで Q35 マシンタイプへの対応

Red Hat Enterprise Linux 8 では、より新しい PCI Express ベースのマシンタイプである **Q35** に対応するようになりました。これにより、仮想デバイスの機能とパフォーマンスにさまざまな改善が行われ、最新の広範囲なデバイスで仮想システムへの互換性が保証されます。また、Red Hat Enterprise Linux 8 で作成された仮想マシンは、デフォルトで **Q35** を使用するように設定されています。

また、以前のデフォルトの **PC** マシンタイプは非推奨となったため、Q35 に対応しない以前のオペレーティングシステムを仮想化する場合にのみ使用してください。

(BZ#1599777)

コピー後の仮想マシンの移行

RHEL 8 では、KVM 仮想マシン (VM) のコピー後の移行を実行できます。コピー後の移行を使用すると、移行元ホストで移行中の VM の vCPU が一時停止され、最小限のメモリーページのみが転送され、移行先ホストで仮想マシンの vCPU がアクティブ化され、仮想マシンが移行先で実行されている間に残りのメモリーページが転送されます。

これにより、移行された仮想マシンのダウンタイムが大幅に短縮され、移行元の仮想マシンのメモリーページがどれほど急速に変化するかに関係なく、移行が確実に完了します。そのため、標準のプレコピー移行では移行できない、頻繁に使用される仮想マシンの移行に最適です。

JIRA:RHELPLAN-160541

virtio-gpu が KVM 仮想化でサポートされるようになりました

KVM 仮想マシン (VM) 用に **virtio-gpu** ディスプレイデバイスが導入されました。**virtio-gpu** は、仮想マシンのグラフィックパフォーマンスを向上させ、将来実装される仮想 GPU デバイスのさまざまな機能強化も可能にします。

JIRA:RHELPLAN-160541

KVM は RHEL 8 で UMIP に対応

ユーザー空間のアプリケーションが、システム全体の設定にアクセスしないようにする UMIP (User-Mode Instruction Prevention) 機能に、KVM 仮想化システムが対応するようになりました。これにより、権限昇格攻撃の潜在的なベクトルが減少するため、KVM ハイパーバイザーとそのゲストマシンがより安全になります。

(BZ#1494651)

KVM ゲストクラッシュレポートの情報が追加される

ゲストが予期せず終了したり、応答しなくなった場合に、KVM ハイパーバイザーが生成するクラッシュ情報が拡張されました。これにより、KVM 仮想化のデプロイメントにおける問題の診断と修正が容易になります。

(BZ#1508139)

NVIDIA vGPU と VNC コンソールの互換性を持たせる

NVIDIA 仮想 GPU (vGPU) 機能を使用するときに、VNC コンソールを使用して、ゲストの視覚的な出力を表示できるようになりました。

(BZ#1497911)

仮想化システムで Ceph に対応

今回の更新で、Red Hat で対応する全 CPU アーキテクチャーの KVM 仮想システムで、Ceph ストレージに対応できるようになりました。

(BZ#1578855)

IBM Z の KVM 仮想マシン用の対話型ブートローダー

IBM Z ホストで KVM 仮想マシンを起動するとき、QEMU ブートローダーファームウェアが、ゲスト OS の対話型コンソールインタフェースを表示するようになりました。これにより、ホスト環境にアクセスせずにゲスト OS の起動問題をトラブルシューティングすることができるようになりました。

(BZ#1508137)

仮想マシンで IBM z14 ZR1 に対応

KVM ハイパーバイザーが、IBM z14 ZR1 サーバーの CPU モデルに対応するようになりました。これにより、IBM Z システムで実行する KVM 仮想マシンでこの CPU の機能を使用できるようになります。

(BZ#1592337)

KVM が IBM Z で Telnet 3270 に対応

IBM Z システムのホストとして RHEL 8 を使用する場合は、Telnet 3270 クライアントを使用して、ホストの仮想マシンに接続できるようになりました。

(BZ#1570029)

QEMU サンドボックス化が追加

Red Hat Enterprise Linux 8 では、QEMU エミュレーターでサンドボックス機能が追加されました。これにより、QEMU が実行するシステムコールに制限が設定でき、仮想マシンの安全性が高まりました。この機能はデフォルトで有効になっており、設定されています。

(JIRA:RHELPLAN-10628)

PV TLB Flush Hyper-V 啓発

RHEL 8 は、PV TLB Flush Hyper-V Enlightenment 機能を追加します。これにより、KVM ハイパーバイザー上のオーバーコミット環境で実行する Windows 仮想マシン (VM) のパフォーマンスが向上します。

JIRA:RHELPLAN-160541

IBM POWER における KVM 仮想マシンの新しいマシンタイプ

IBM POWER 8 および IBM POWER 9 のシステムで実行している KVM ハイパーバイザーに対して、複数の新しい rhel-pseries マシンタイプが有効になりました。これにより、IBM POWER システムの RHEL 8 にホストされている仮想マシンが、このマシンタイプの CPU 機能を正しく使用できます。また、これにより、IBM POWER の仮想マシンを、最新バージョンの KVM ハイパーバイザーに移行できます。

(BZ#1585651, BZ#1595501)

Intel Xeon SnowRidge で GFNI および CLDEMOT の命令セットが有効化

Intel Xeon SnowRidge システムの RHEL 8 ホストで稼働している仮想マシンで、GFNI および CLDEMOT 命令セットを使用できるようになりました。これにより、特定のシナリオでこのような仮想マシンのパフォーマンスが大幅に向上する可能性があります。

(BZ#1494705)

OVMF に対して IPv6 を有効にする

IPv6 プロトコルが、OVMF (Open Virtual Machine Firmware) で有効にできるようになりました。これにより、OVMF を使用する仮想マシンが、IPv6 が提供するさまざまなネットワーク起動の改善を活かすことができるようになります。

(BZ#1536627)

NVMe デバイス用の VFIO ベースのブロックドライバーが追加

QEMU エミュレーターには、NVMe (Non-volatile Memory Express) デバイス用の、VFIO (Virtual Function I/O) ベースのドライバーが追加されました。ドライバーは、仮想マシン (VM) に接続されている NVMe デバイスと直接通信し、カーネルシステムレイヤーと、その NVMe ドライバーを使用しないようにします。これにより、仮想マシン内の NVMe デバイスのパフォーマンスが向上します。

(BZ#1519004)

Hyper-V Generic UIO ドライバーのマルチチャネルへの対応

RHEL 8 は、Hyper-V Generic Userspace I/O (UIO) ドライバーのマルチチャネル機能に対応するようになりました。これにより、Hyper-V ハイパーバイザーで実行している RHEL 8 仮想マシンは、この仮想マシンのネットワーク機能を強化するデータプレーン開発キット (DPDK) の Netvsc Poll Mode ドライバー (PMD) を使用できます。

ただし、Netvsc インターフェイスのステータスは、現在、実行中で使用可能な場合でも、ダウンしていると表示されます。

(BZ#1650149)

Huge page の対応が改善

RHEL 8 を仮想ホストとして使用する場合は、仮想マシンのメモリーに対応するページのサイズを、CPU が対応する任意のサイズに変更できます。これにより、仮想マシンのパフォーマンスが大きく改善します。

仮想マシンのメモリーページのサイズを設定するには、仮想マシンの XML 設定を編集して、`<hugepages>` 要素を `<memoryBacking>` セクションに追加します。

(JIRA:RHELPLAN-14607)

POWER 9 ホスト上の仮想マシンは THP を使用できます

IBM POWER 9 アーキテクチャーで実行している RHEL 8 ホストでは、仮想マシン (VM) は透過的なヒュージページ (THP) 機能の恩恵を受けます。THP を使用すると、ホストカーネルが巨大なメモリーページをプロセスに動的に割り当てることができるため、大量のメモリーを備えた仮想マシンのパフォーマンスが向上します。

JIRA:RHELPLAN-160541

5.1.17. サポート性

sosreport が eBPF ベースのプログラムおよびマップを報告

sosreport ツールが拡張され、Red Hat Enterprise Linux 8 に読み込まれた eBPF (extended Berkeley Packet Filtering) プログラムおよびマップをすべて報告できるようになりました。

(BZ#1559836)

5.2. 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 8.0 のバグで修正されたものを説明します。

5.2.1. デスクトップ

PackageKit が rpm パッケージで動作可能

今回の更新で、**rpm** パッケージでの操作への対応が **PackageKit** に追加されました。

(BZ#1559414)

5.2.2. グラフィックインフラストラクチャー

QEMU が 8 バイトの **ggtt** エントリーを正しく処理しない

QEMU が、8 バイトの **ggtt** エントリーの書き込みを、連続した 4 バイトの書き込みに分割することがあります。これらの部分的な書き込みは、それぞれ別々のホストの **ggtt** を発生させます。2 つの **ggtt** 書き込みが不適切に結合される場合があります。したがって、マシンアドレスへの変換に失敗し、エラーログが発生します。

(BZ#1598776)

5.2.3. ID 管理

Enterprise Security Client が、トークン検出に **opensc** ライブラリーを使用

Red Hat Enterprise Linux 8.0 は、スマートカードの **opensc** ライブラリーにのみ対応します。今回の更新で、Enterprise Security Client (ESC) は、削除された **coolkey** ライブラリーの代わりに **opensc** を使用してトークンを検出します。その結果、アプリケーションで、対応しているトークンを正しく検出します。

(BZ#1538645)

証明書システムが、デバッグログのローテートに対応

以前は、証明書システムが、ログローテーションに対応しないカスタムのロギングフレームワークを使用していました。これにより、`/var/log/pki/instance_name/ca/debug` などのデバッグログが無限に大きくなります。今回の更新で、証明書システムは、ログローテーションに対応する **java.logging.util** フレームワークを使用します。これにより、`/var/lib/pki/instance_name/conf/logging.properties` ファイルでログローテーションを設定できます。

ログローテーションの詳細は、**java.util.logging** パッケージのドキュメントを参照してください。

(BZ#1565073)

証明書システムが、サービスの起動時に **SetAllPropertiesRule** 操作の警告をログに記録しなくなる

以前は、証明書システムは、サービスの開始時に `/var/log/messages` ログファイルの `SetAllPropertiesRule` 操作で警告を記録していました。問題が修正され、上記の警告は記録されなくなりました。

(BZ#1424966)

Certificate System KRA クライアントは、鍵要求の応答を正しく解析する

以前は、証明書システムが新しい JSON ライブラリーに切り替えていました。その結果、特定のオブジェクトのシリアライズが異なり、Python の鍵回復機関 (KRA) クライアントが、鍵要求の応答を解析できませんでした。クライアントは、古い JSON ライブラリーと新しい JSON ライブラリーの両方を使用した応答に対応するように修正されました。これにより、Python KRA クライアントはキー要求 応答を正しく解析します。

(BZ#1623444)

5.2.4. コンパイラーおよび開発ツール

GCC が範囲外のアクセスに関する誤検出警告を生成しなくなる

以前は、最適化レベルのオプション `-O3` でコンパイルする際に、コンパイルしたコードがこれを含まない場合でも、GNU コンパイラーコレクション (GCC) が範囲外のアクセスに関する誤検出の警告を返すことがありました。最適化が修正され、GCC が誤検出警告を表示しないようになりました。

(BZ#1246444)

Itrace が大きな構造を正しく表示する

以前は、`Itrace` ツールが、関数から返される大きな構造を正しく出力できませんでした。`Itrace` での大規模な構造の処理が改善され、正しく出力されるようになりました。

(BZ#1584322)

GCC 組み込み関数 `__builtin_clz` が、IBM Z で正しい値を返す

以前は、IBM Z アーキテクチャーの `FLOGR` 命令が、GCC コンパイラーにより誤って省略されていました。これにより、この命令を使用する `__builtin_clz` 関数は、GCC オプション `-funroll-loops` でコードがコンパイルされると、誤った結果が返される場合があります。このバグは修正され、関数で正しい結果が提供されるようになりました。

(BZ#1652016)

GDB は、バッチモードで最後のコマンドが失敗した場合に、ゼロ以外の終了ステータスを提供

以前は、バッチモードで実行する際に、コマンド内のエラーに関係なく、GDB が常にステータス `0` で終了していました。したがって、コマンドが成功したかどうかを確認できませんでした。この動作が変更され、最後のコマンドでエラーが発生すると、GDB がステータス `1` で終了するようになりました。これにより、すべてのコマンドが実行する前の動作との互換性が維持されるようになりました。したがって、GDB バッチモードの実行が成功したかどうかを判断できるようになりました。

(BZ#1491128)

5.2.5. ファイルシステムおよびストレージ

印刷レベルを高くしても、`iscsiadm` が突然終了しなくなる

以前は、`iscsiadm` の `iscsiadm --shell` コマンドが、`iscsiadm --shell --help` コマンドと同様に、`iscsiadm --shell --help` コマンドを実行すると、`iscsiadm` が突然終了するようになりました。

以前は、**--print** オプションまたは **-P** オプションで 0 を超える出力レベルを指定すると、**iscsiadm** ユーティリティーが突然終了していました。この問題は修正され、すべての印刷レベルが期待どおりに機能するようになりました。

(BZ#1582099)

multipathd が、パスの WWID を取得できない場合に、パスを無効にしない

multipathd サービスは、以前は、パスの WWID の取得に失敗したことを、空の WWID の取得として処理していました。**multipathd** がパスの WWID を取得できない場合は、そのパスが無効になる可能性があります。

今回の更新で、変更したかどうかを確認する際に WWID を取得できないと、**multipathd** は引き続き古い WWID を使用します。

したがって、WWID が変更したかどうかを確認する際に **multipathd** が WWID を取得できない場合にパスが無効になることがなくなりました。

(BZ#1673167)

5.2.6. 高可用性およびクラスター

新しい /etc/sysconfig/pcsd オプションにより、クライアントが起動する SSL/TLS 再ネゴシエーションの拒否

サーバーで TLS 再ネゴシエーションを有効にすると、クライアントが、新しいハンドシェイクを開始する再ネゴシエーション要求を送信できるようになります。ハンドシェイクの計算要件は、クライアントよりもサーバーの方が高くなります。これにより、サーバーが DoS 攻撃に対して脆弱になります。今回の修正で、**/etc/sysconfig/pcsd** 設定ファイルに **PCSD_SSL_OPTIONS** を設定すると、再ネゴシエーションを拒否する **OP_NO_RENEGOTIATION** オプションを受け付けるようになりました。ただし、クライアントは、引き続き、すべての接続でハンドシェイクを実行して、サーバーへの接続を複数開くことができます。

(BZ#1566430)

削除されたクラスターノードがクラスターのステータスに表示されない

以前は、**pcs cluster node remove** コマンドでノードが削除されると、削除したノードは **pcs status** ディスプレイの出力に表示されたままになります。この修正により、削除されたノードはクラスターのステータスには表示されなくなりました。

(BZ#1595829)

フェンスエージェントが、新しい優先パラメーター名または非推奨のパラメーター名のいずれかを使用して設定可能に

古いパラメーター名は非推奨として引き続き対応していますが、フェンスエージェントパラメーター名の多くが変更になりました。以前は、**--force** オプションと併用しないと、**pcs** で新しいパラメーターを設定できませんでした。この修正により、**pcs** は、非推奨パラメーターへの対応を維持しながら、名前を変更したフェンスエージェントパラメーターに対応するようになりました。

(BZ#1436217)

pcs コマンドが、クラスターの表示用に XML ステータスを正しく読み込むようになる

pcs コマンドは、**crm_mon** ユーティリティーを実行し、XML 形式のクラスターのステータスを取得します。**crm_mon** ユーティリティーは、標準出力に XML を出力し、標準エラー出力に警告を出力します。以前は、**pcs** では 1 つのストリームで XML と警告が混在していたため、これを XML として解析で

きませんでした。この修正により、**pcs** で標準出力とエラー出力が分離し、クラスタの XML ステータスの読み取りが期待どおりに機能するようになりました。

(BZ#1578955)

既存のクラスタのノードを持つ新規クラスタを作成する際に、クラスタを破棄することが非推奨に

以前は、**pcs cluster setup** コマンドを実行する場合や、Web UI の **pcsd** でクラスタを作成する場合に、ユーザーが既存のクラスタからノードを指定すると、**pcs** がそれをエラーとして報告し、そのノードでクラスタを破棄することを推奨していました。したがって、ノードでクラスタを破棄した場合に、ノードが残りのノードに含まれていたため、クラスタを分割しても、破棄されたノードがクラスタの一部とみなされていました。この修正により、クラスタからノードを削除することが推奨され、クラスタを破損することなく問題に対応する方法がよりの確に示されるようになります。

(BZ#1596050)

pcs コマンドが対話的に認証情報を要求しなくなる

root 以外のユーザーが root パーミッションを必要とする **pcs** コマンドを実行すると、**pcs** はローカルで実行している **pcsd** デーモンに接続し、そのデーモンにそのコマンドを渡します。これは、root 権限で **pcsd** デーモンを実行し、コマンドを実行することができるためです。以前は、ユーザーがローカルの **pcsd** デーモンで認証されていないと、**pcs** がユーザー名とパスワードを対話的に要求していました。これはユーザーが混乱し、**pcs** を実行しているスクリプトで必要とされる特別な処理です。この修正では、ユーザーが認証されないと、**pcs** がエラーで終了し、何を実行するか (root で **pcs** を実行するか、新しい **pcs client local-auth** コマンドを使用して認証する) が表示されていました。このため、**pcs** コマンドが対話的に認証情報を要求せず、ユーザーエクスペリエンスが改善されます。

(BZ#1554310)

crypto-policies が **FUTURE** に設定されていると、**pcsd** デーモンがデフォルトの自己生成 SSL 証明書で開始

crypto-policies を **FUTURE** に設定するには、SSL 証明書の RSA 鍵の長さが 3072b 以上である必要があります。以前は、このポリシーが 2048b 鍵で SSL 証明書を生成するため、このポリシーを設定すると **pcsd** デーモンが起動しませんでした。今回の更新で、**pcsd** が自己生成した SSL 証明書の鍵のサイズが 3072b に拡大し、**pcsd** がデフォルトの自己生成 SSL 証明書で起動するようになりました。

(BZ#1638852)

ネットワークの準備ができると **pcsd** サービスが開始するようになる

以前は、ユーザーが特定の IP アドレスにバインドするように **pcsd** を設定すると、システムの起動時に **pcsd** が起動しようとしたときにそのアドレスの準備が間に合わず、**pcsd** が開始できなかったため、**pcsd** を手動で起動する必要がありました。今回の修正で、**pcsd.service** が **network-online.target** に依存するようになります。これにより、ネットワークが準備され、IP アドレスにバインドすると、**pcsd** が起動します。

(BZ#1640477)

5.2.7. ネットワーク

弱い TLS アルゴリズムが **glib-networking** に許可されなくなる

以前は、**glib-networking** パッケージが、RHEL 8 のシステム全体の暗号化ポリシーと互換性がありませんでした。したがって、**glib** ライブラリーをネットワークに使用するアプリケーションでは、管理者が意図しているアルゴリズムよりも弱いアルゴリズムを使用した TLS (Transport Layer Security) 接続

が許可される可能性があります。今回の更新で、システム全体の暗号化ポリシーが適用され、ネットワークに **glib** を使用するアプリケーションでは、ポリシーに従って許可される TLS 接続のみが許可されます。

(BZ#1640534)

5.2.8. セキュリティー

SELinux ポリシーにより、iscsiuio プロセスが検出ポータルに接続可能に

以前は、SELinux ポリシーは **iscsiuio** プロセスへの制約が多すぎて、このプロセスが **mmap** システムコールを使用して **/dev/uid*** デバイスにアクセスできませんでした。これにより、検出ポータルへの接続に失敗していました。今回の更新で、足りないルールが SELinux ポリシーに追加され、上記のシナリオで期待どおりに **iscsiuio** プロセスが機能するようになりました。

(BZ#1626446)

5.2.9. サブスクリプション管理

dnf および yum が、subscription-manager の値に関係なくリポジトリにアクセス可能に

以前は、**dnf** コマンドまたは **yum** コマンドが、**subscription-manager** サービスが追加した URL の **https://** 接頭辞を無視していました。更新した **dnf** コマンドまたは **yum** コマンドは、無効な **https://** の URL を無視しません。したがって、**dnf** と **yum** はリポジトリにアクセスできませんでした。この問題を解決するには、新しい設定変数 **proxy_scheme** が **/etc/rhsm/rhsm.conf** ファイルに追加され、値を **http** または **https** に設定できます。値を指定しないと、**subscription-manager** がデフォルトで、より一般的に使用される **http** を設定します。

プロキシーが **http** を使用すると、ほとんどの場合は **/etc/rhsm/rhsm.conf** で設定を変更しないようにする必要があります。プロキシーが **https** を使用する場合は、**proxy_scheme** の値を **https** に更新する必要があります。いずれの場合も、ユーザーは **subscription-manager repos --list** コマンドを実行するか、**rhsmcertd** デーモンプロセスが **/etc/yum.repos.d/redhat.repo** を適切に再生成するまで待機する必要があります。

(BZ#1654531)

5.2.10. 仮想化

Azure での一時ディスクのマウントが、より確実に機能するようになる

以前は、Microsoft Azure プラットフォームで実行している仮想マシンへの一時ディスクのマウントは、仮想マシンが停止 (割り当て解除) になって起動すると、失敗していました。今回の更新で、上記の状況で、再接続したディスクが正しく処理され、問題が発生しないようにします。

(BZ#1615599)

5.3. テクノロジープレビュー

本パートでは、Red Hat Enterprise Linux 8.0 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機能に対する Red Hat のサポート範囲の詳細は、[テクノロジープレビュー機能のサポート範囲](#) を参照してください。

5.3.1. カーネル

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 Filtering)機能は、テクノロジープレビューとしてネットワークングおよびトレースの両方に利用できます。eBPF を使用すると、ユーザー空間はカスタムプログラムを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) に接続してデータを受信して処理できるようにします。この機能には、新しいシステムコール `bpf()` が含まれます。これは、様々な種類のマップの作成と、様々な種類のプログラムの更新に対応します。`bpf()` は、root 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。

(BZ#1559616)

BCC がテクノロジープレビューとして利用可能に

BPF コンパイラコレクション (BCC) は、Red Hat Enterprise Linux 8 でテクノロジープレビューとして利用できる、効率的なカーネルの追跡および操作プログラムを作成するユーザー空間ツールパッケージです。**BCC** は、**eBPF (extended Berkeley Packet Filtering)** を使用して、Linux オペレーティングシステムの I/O 解析、ネットワークング、およびモニタリング用のツールを提供します。

(BZ#1548302)

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
- 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)

early kdump が Red Hat Enterprise Linux 8 でテクノロジープレビューとして利用可能に

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)

ibmvnic デバイスドライバーがテクノロジープレビューとして利用可能に

Red Hat Enterprise Linux 8.0 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す。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。

(BZ#1524683)

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)

5.3.2. 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)

5.3.3. ハードウェアの有効化

クラスター対応の MD RAID1 がテクノロジープレビューとして利用可能に

RAID1 クラスターは、カーネル領域ではデフォルトで有効になっていません。RAID1 クラスターを試す場合は、最初に RAID1 クラスターをモジュールとしてカーネルを構築する必要があります。以下の手順を使用します。

1. **make menuconfig** コマンドを入力します。
2. **make && make modules && make modules_install && make install** コマンドを入力します。
3. **reboot** コマンドを入力します。

(BZ#1654482)

5.3.4. ID 管理

DNSSEC が IdM でテクノロジープレビューとして利用可能に

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)

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

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)

5.3.5. ファイルシステムおよびストレージ

Aero アダプターがテクノロジープレビューとして利用可能に

以下の Aero アダプター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。

- 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)

Stratis が利用可能に

Stratis 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。

Stratis 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

Stratis 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。

Stratis 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)

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、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)

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

Red Hat Enterprise Linux 8.0 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)

5.3.6. 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)

5.3.7. ネットワーク

XDP がテクノロジープレビューとして利用可能に

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)

eBPF がテクノロジープレビューとして利用可能に

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)

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)

KTLS がテクノロジープレビューとして利用可能に

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)

TIPC がテクノロジープレビューとして利用可能に

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てもかわらず、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、テクノロジープレビューとして利用できます。

(BZ#1581898)

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)

5.3.8. 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**
- **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#2213183)

5.3.9. 仮想化

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)

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)

RHEL 8 Hyper-V 仮想マシンで KVM 仮想化が利用可能

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用でき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)

5.3.10. 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)

5.4. 非推奨になった機能

ここでは、Red Hat Enterprise Linux 8.0 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。

5.4.1. 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 `ignoredisk` の `--interactive` オプションが非推奨に

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)

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

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)

5.4.2. ファイルシステムおよびストレージ

NFSv3 over UDP が無効に

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)

elevator カーネルコマンドラインパラメーターが非推奨に

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。

詳細は、[Why does the 'elevator=' parameter no longer work in RHEL8](#) を参照してください。

(BZ#1665295)

VDO パッケージの VDO Ansible モジュール

VDO Ansible モジュールは、現在、RPM パッケージ **vdo** で提供されます。今後のリリースで、VDO Ansible モジュールは Ansible RPM パッケージに移動します。

(BZ#1669537)

5.4.3. 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)

5.4.4. カーネル

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)

5.4.5. セキュリティー

RHEL 8 で DSA が非推奨に

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)

TLS 1.0 および TLS 1.1 が非推奨に

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の上レヅジベース [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](#))

5.4.6. 仮想化

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ではない

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](#))

virt-manager が非推奨に

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、Red Hat Enterprise Linux 8.0 では、機能によっては、**virt-manager** またはコマンドラインからしかアクセスできません。

([JIRA:RHELPLAN-10304](#))

5.4.7. 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

5.5. 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。

5.5.1. 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ない

現在、tlog の録画を有効にしていると、RHEL Web コンソールのログインに失敗します。RHEL Web コンソールにログインする場合は、`/etc/shells` ディレクトリーにユーザーのシェルが存在している必要があります。ただし、**tlog-rec-session** を `/etc/shells` に追加する場合は、chsh ユーティリティーを使用して、**tlog-rec-session** のシェルを、`/etc/shells` の別のシェルに変更することで、録画を無効にできません。Red Hat は、この理由により、**tlog-rec-session** を `/etc/shells` に追加することは推奨していません。

(BZ#1631905)

5.5.2. 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 リポジトリーが必要になる

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 リポジトリーでしか利用できません。

この問題を回避するには、BaseOS リポジトリーおよび AppStream リポジトリー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)

ビデオドライバー `xorg-x11-drv-fbdev`、`xorg-x11-drv-vesa`、および `xorg-x11-drv-vmware` がデフォルトではインストールされない

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ません。

この問題を回避するには、ワークステーションマシンで RHEL 8.0 **Workstation** のインストールを実行します。ワークステーションで RHEL 8.0 **Server** のインストールが必要な場合は、`yum -y groupinstall base-x` コマンドを実行して、インストール後に `base-x` パッケージグループを手動でインストールします。

さらに、Hyper-V などのグラフィックに EFI を使用する仮想マシンも影響を受けます。Hyper-V で **Server with GUI** ベース環境を選択すると、システムの再起動時にブラックスクリーンが表示され、口

ログインできない場合があります。Hyper-v でこの問題を回避するには、以下の手順に従ってマルチユーザーモードまたはシングルユーザーモードを有効にします。

1. 仮想マシンを起動します。
2. システムの起動プロセス中に、キーボードの上矢印キーおよび下矢印キーを使用して必要なカーネルを選択します。
3. キーボードの **e** キーを押して、カーネルコマンドラインを編集します。
4. **systemd.unit=multi-user.target** を、GRUB のカーネルコマンドラインに追加します。
5. **Ctrl-X** を押して、仮想マシンを起動します。
6. ログインしたら、**yum -y groupinstall base-x** コマンドを実行します。
7. 仮想マシンを再起動してグラフィカルモードにアクセスします。

(BZ#1687489)

reboot --kexec コマンドを使用するとインストールが失敗する

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)

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ない

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#1692746)

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る

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。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)

5.5.3. 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)

システムにデバイスが多数接続されていると、システムが応答しなくなることがある

Red Hat Enterprise Linux 8 に多数のデバイスを設定すると、システムコンソールにコンソールメッセージが多数表示されます。たとえば、論理ユニット番号 (LUN) が多数あり、その各 LUN へのパスが複数あると、このような状況が発生します。カーネルが行っている他の作業に加えて、コンソールメッセージがあふれるとカーネルがハングするため、カーネルのウォッチドッグによりカーネルパニック発生する場合があります。

スキャンはブートサイクルの早い段階で行われるため、接続されているデバイスが多くなるとシステムが応答しなくなります。これは通常、起動時に発生します。

システム起動後のデバイススキャンイベント中に、マシンで **kdump** が有効になっていると、ハードロックアップにより **vmcore** イメージがキャプチャーされます。

この問題を回避するには、ウォッチドッグロックアップタイマーを増やします。これを行うには、カーネルのコマンドラインに **watchdog_thresh=N** オプションを追加します。**N** を、次の秒数に置き換えます。

- デバイスが 1000 台未満の場合は、**30** を使用してください。
- デバイスが 1000 台以上の場合は、**60** を使用してください。

ストレージの場合、デバイスの数はすべての LUN へのパスの数です。通常、これは **/dev/sd*** デバイスの数です。

回避策を適用した後、大量のデバイスを設定してもシステムが応答しなくなることはなくなりました。

(BZ#1598448)

KSM が、NUMA メモリーポリシーを無視することがある

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)

qede ドライバーが NIC をハングし、その NIC を使用できなくなるようにする

バグにより、QLogic シリーズの 41000 および 45000 の NIC 用の **qede** ドライバーで、ファームウェアアップグレードおよびデバッグのデータ収集操作に失敗し、ホストの再起動 (PCI リセット) が NIC を再操作するまで NIC が使用できなくなるか、ハング状態になる可能性があります。

この問題は、以下のすべてのシナリオで検出されました。

- inbox ドライバーを使用して NIC のファームウェアをアップグレードした場合
- **ethtool -d ethx** コマンドを実行してデバッグデータを収集した場合
- **ethtool -d ethx** を追加して **sosreport** コマンドを実行した場合
- inbox ドライバーが、IO タイムアウト、Mail Box Command のタイムアウト、Hardware Attention などの自動デバッグデータ収集を開始した場合

Red Hat の今後のエラータは、この問題に対処するために Red Hat Bug Advisory (RHBA) 経由でリリースされます。この問題を回避するには、RHBA がリリースされるまで、<https://access.redhat.com/support> で、この問題に対するサポート対象の修正を要求するケースを作成します。

(BZ#1697310)

radix ツリーシンボルが kernel-abi-whitelists に追加

Red Hat Enterprise Linux 8 の **kernel-abi-whitelists** パッケージに、以下の radix ツリーシンボルが追加されました。

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

上記のシンボルは本来存在しなければならないものではなく、RHEL8 ホワイトリストから削除されます。

(BZ#1695142)

podman が、RHEL 8 でコンテナのチェックポイントに失敗する

Red Hat Enterprise Linux 8 では、CRIU (Checkpoint and Restore In Userspace) パッケージのバージョン

ンが廃止されました。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ます。**podman container checkpoint** コマンドを実行すると、エラーメッセージ `checkpointing a container requires at least CRIU 31100` が表示されます。

(BZ#1689746)

dracut.conf で `add_dracutmodules+=earlykdump` オプションを使用すると、**early-kdump** および標準の **kdump** が失敗する

現在、**early-kdump** 用にインストールするカーネルバージョンと、**initramfs** 用に生成されるカーネルバージョンの間で不整合が発生します。これにより、**early-kdump** を有効にしてシステムを起動すると、**early-kdump** が失敗します。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、強制的に終了します。また、**early-kdump** を、デフォルトの **dracut** モジュールとして追加すると、**kdump** `initramfs` を再構築しようとする際に、標準の **kdump** サービスが失敗します。これにより、**early-kdump** と標準の **kdump** の両方が失敗します。この問題を回避するには、**dracut.conf** ファイルに `add_dracutmodules+=earlykdump` または同等の設定を追加しないでください。これにより、デフォルトで **early-kdump** が **dracut** に含まれていないため、問題が発生しなくなります。ただし、**early-kdump** が必要な場合は、手動で作成する必要があります。

(BZ#1662911)

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する

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)

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)

5.5.4. ソフトウェア管理

root 以外のユーザーで `yum list` を実行すると、YUM がクラッシュする

libdnf パッケージを更新してから、root 以外のユーザーで `yum list` コマンドを実行すると、YUM が突然終了することがあります。このバグが発生した場合は、問題を解決するために root で `yum list` を実行してください。これにより、root 以外のユーザーで `yum list` を実行しようとしても、YUM クラッシュが発生しなくなります。

(BZ#1642458)

YUM v4 が、デフォルトで利用できないリポジトリをスキップ

YUM v4 では、すべてのリポジトリで `skip_if_unavailable=True` 設定がデフォルトに設定されます。

そのため、必要なリポジトリが利用できない場合、リポジトリのパッケージはインストール、検索、または更新の操作で考慮されません。その後、一部の **yum** コマンドおよび yum ベースのスク립トは、利用できないリポジトリが存在する場合でも、終了コード 0 で成功します。

現在、**libdnf** パッケージを更新する以外に回避策はありません。

(BZ#1679509)

5.5.5. インフラストラクチャーサービス

nslookup ユーティリティーおよび **host** ユーティリティーが、再帰を利用できないネームサーバーからの返信を無視する

他のネームサーバーが設定されていて、ネームサーバーで再帰が使用できない場合は、**nslookup** ユーティリティーおよび **host** ユーティリティーが、そのネームサーバーが最後に設定されているものでない限り、このようなネームサーバーからの返信を無視します。前回設定したネームサーバーでは、**recursion available** フラグがなくても応答が許可されます。ただし、最後に設定したネームサーバーが応答しない場合や到達できない場合は、名前解決に失敗します。

この問題を回避するには、以下のコマンドを実行します。

- 設定したネームサーバーが、**recursion available** フラグが設定されている状態で常に応答することを確認します。
- すべての内部クライアントに再帰を許可します。

この問題のトラブルシューティングには、**dig** ユーティリティーを使用して、再帰が利用可能かどうかを検出することもできます。

(BZ#1599459)

5.5.6. シェルおよびコマンドラインツール

net-snmp パッケージの Python バインディングが利用できない

Net-SNMP のツールスイートは、RHEL 8 のデフォルトの **Python** の実装である **Python 3** にバインディングを提供しません。これにより、**python-net-snmp** パッケージ、**python2-net-snmp** パッケージ、または **python3-net-snmp** パッケージが RHEL 8 では使用できません。

(BZ#1584510)

デバッグモードの **systemd** が不要なログメッセージを生成する

デバッグモードの **systemd** システムおよびサービスマネージャーは、以下で始まる不要なログメッセージを生成します。

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

以下を実行してメッセージをリスト表示します。

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

これらのデバッグメッセージは無害で、無視しても問題はありません。

現在利用できる回避策はありません。

(BZ#1658691)

KEYBD トラップを使用した ksh がマルチバイト文字を誤って処理

Korn Shell (KSH) は、**KEYBD** トラップが有効な場合にマルチバイト文字を正しく処理できません。したがって、たとえばユーザーが日本語の文字を入力すると、**ksh** には間違った文字列が表示されます。この問題を回避するには、以下の行をコメントアウトして、`/etc/kshrc` ファイルの **KEYBD** トラップを無効にします。

```
trap keybd_trap KEYBD
```

詳細は、ナレッジベースソリューション [ksh displays multibyte characters incorrectly when 'KEYBD trap' is enabled in profile file](#) を参照してください。

(BZ#1503922)

5.5.7. 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

データベースサーバーを並行してインストールできない

RPM パッケージが競合しているため、RHEL 8.0 では、 **mariadb** モジュールと **mysql** モジュールを同時にインストールすることができません。

設計上、同じモジュールの複数のバージョン (ストリーム) を並行してインストールすることはできません。たとえば、**postgresql** モジュールから利用可能なストリーム (**10** (デフォルト) または **9.6**) の中から 1 つ選択する必要があります。RHEL 6 および RHEL 7 用の Red Hat Software Collections では、コンポーネントの並列インストールが可能です。RHEL 8 では、コンテナ内で異なるバージョンのデータベースサーバーを使用できます。

(BZ#1566048)

mod_cgid ログイングの問題

mod_cgid Apache httpd モジュールを、スレッド化マルチプロセッシングモジュール (MPM) で使用する場合 (RHEL 8 のデフォルト) は、以下のログイング問題が発生します。

- CGI スクリプトの **stderr** 出力には、標準のタイムスタンプ情報が付いていません。
- **VirtualHost** に特有のログファイルに CGI スクリプトの **stderr** 出力が設定されていても、正しくリダイレクトされません。

(BZ#1633224)

Perl モジュール **IO::Socket::SSL** で TLS 1.3 に対応しない

セッション再開、ハンドシェイク後認証などの TLS 1.3 プロトコルの新機能は、RHEL 8 の **OpenSSL** ライブラリーに実装されますが、Perl モジュール **Net::SSLeay** には実装されないため、**IO::Socket::SSL** Perl モジュールでは利用できません。クライアント証明書の認証が失敗し、TLS 1.2 プロトコルを使用した場合よりもセッションの再確立が遅くなる可能性があります。

この問題を回避するには、**IO::Socket::SSL** オブジェクトを作成する場合に、**SSL_version** オプションを **!TLSv1_3** 値に設定して、TLS 1.3 を無効にします。

(BZ#1632600)

生成された Scala のドキュメントが読み込めない

scaladoc コマンドを使用してドキュメントを生成すると、JavaScript リソースが見つからないため、生成される HTML ページが使用できなくなります。

(BZ#1641744)

5.5.8. デスクトップ

qxl が Wayland ベースの仮想マシンで動作しない

qxl ドライバーは、特定のハイパーバイザーでカーネルモード設定機能を提供できません。したがって、Wayland プロトコルに基づくグラフィックは、**qxl** を使用する仮想マシンでは利用できず、Wayland ベースのログイン画面は起動しません。

この問題を回避するには、以下のいずれかを使用します。

- QXL (QuarkXpress Element Library) グラフィックをベースとした仮想マシンで、**Wayland の GNOME Shell** の代わりに **Xorg** ディスプレイサーバー。

または

- 仮想マシンの **qxl** ドライバーではなく **virtio** ドライバー。

(BZ#1641763)

systemctl isolate multi-user.target の実行時にコンソールプロンプトが表示されない

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ます。この問題を回避するには、**Ctrl+Alt+F2** キーを押します。これにより、コンソールプロンプトが表示されます。

動作は、**Wayland の GNOME Shell** および **X.Org** ディスプレイサーバーの両方に適用されます。

(BZ#1678627)

5.5.9. グラフィックインフラストラクチャー

X.Org で実行しているデスクトップの画面解像度を下げるとハングする

X.Org ディスプレイサーバーで **GNOME** デスクトップを使用する場合は、画面の解像度を低い値に変更するとデスクトップが応答しなくなります。この問題を回避するには、画面の解像度を **800 × 600** ピクセルより小さい値に設定しないでください。

(BZ#1655413)

radeon がハードウェアを正しくリセットできない

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)

5.5.10. ハードウェアの有効化

ARP リンクモニターの使用時にバックアップスレーブ MII ステータスが機能しない

デフォルトでは、i40e ドライバーにより管理されるデバイスはソースプルーニングを実行し、受信フィルターのいずれかに一致するソースの Media Access Control (MAC) アドレスを持つパケットを破棄します。これにより、チャンネルボンディングでアドレス解決プロトコル (ARP) の監視を使用する場合に、バックアップスレーブの媒体独立インタフェース (MII) 状態が機能しません。この問題を回避するには、以下のコマンドでソースのプルーニングを無効にします。

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

これにより、バックアップスレーブの MII のステータスが期待どおりに機能します。

(BZ#1645433)

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る

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)

5.5.11. ID 管理

KCM 認証情報キャッシュは、1つの認証情報キャッシュ内で多数の認証情報を行うには適していない

Kerberos Credential Manager (KCM) で処理できる **ccache** サイズは、最大 64 kB となります。**sssd-kcm** コンポーネントと、基本的なデータベースとの間でデータを転送するのに使用するバッファにハードコード化された制限があるため、含まれる認証情報が多くなりすぎると、**kinit** などの Kerberos 操作が失敗します。

この問題を回避するには、**/etc/sss/sss.conf** ファイルの **kcm** セクションに **ccache_storage = memory** オプションを追加します。これにより、**kcm** レスポンダーが、永続的ではなく、認証情報キャッシュをメモリ内に格納するようになります。これを行った場合は、システムを再起動するか、**sssd-kcm** で認証情報キャッシュを消去します。

(BZ#1448094)

/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

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)

タイムアウト値が競合しているため、SSSD がサーバーに接続できない

System Security Services Daemon (SSSD) が使用するフェイルオーバー操作に関連するデフォルトのタイムアウト値の一部が競合しています。これにより、全体のタイムアウトとなる接続操作になる前に、SSSD が1台のサーバーと通信するために予約されているタイムアウト値により、SSSD がその他のサーバーを試行しなくなります。この問題を回避するために、`ldap_opt_timeout` タイムアウトパラメーターの値を `dns_resolver_timeout` パラメーターの値より高く設定し、`dns_resolver_timeout` パラメーターの値を `dns_resolver_op_timeout` パラメーターの値よりも高く設定します。

(BZ#1382750)

SSSD は、ID オーバーライドで一意的な証明書のみを検索できる

複数の ID オーバーライドに同じ証明書が含まれる場合、SSSD (System Security Services Daemon) は証明書に一致するユーザーのクエリーを解決できません。これらのユーザーを検索しようとしても、ユーザーは返されません。ユーザー名または UID を使用してユーザーを検索すると、期待通りに機能します。

(BZ#1446101)

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ない

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)

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が、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)

sudo ルールがグループ名を参照すると、id_provider=ad で sudo ルールが機能しないことがある

SSSD (System Security Services Daemon) は、キャッシュを使用して AD と SSSD の間の通信を最適化するため、`initgroups` 操作中は Active Directory グループ名を解決しません。キャッシュエントリーにはセキュリティー識別子 (SID) のみが含まれ、グループが名前または ID で要求されるまでグループ名は含まれません。したがって、`sudo` の実行前にグループが完全に解決されていなければ、`sudo` ルールは AD グループとは一致しません。

この問題を回避するには、最適化を無効にする必要があります。`/etc/sss/sss.conf` ファイルを開き、`[domain/example.com]` セクションに `ldap_use_tokengroups = false` パラメーターを追加します。

(BZ#1659457)

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る

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。ここには、**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)

IdM サーバーが FIPS で機能しない

Tomcat 用の SSL コネクタの実装が不完全なため、証明書サーバーをインストールした 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ません。

(BZ#1673296)

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

Active Directory (AD) ドメインに参加している RHEL ホストのファイルサーバーとして Samba を使用するには、SSSD を使用して AD からユーザーおよびグループを管理している場合でも、Samba Winbind サービスを実行している必要があります。`realm join --client-software=sss` コマンドを使用するか、このコマンドで `--client-software` パラメーターを指定せずにドメインに参加すると、**realm** は `/etc/sss/sss.conf` ファイルのみを作成します。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を変更する際にエラーが発生する場合があります。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ます。

以前のバージョンの RHEL からアップグレードし、`/etc/sss/sss.conf` ファイルの `ldap_id_mapping` パラメーターを **True** (デフォルト) に設定すると、回避策が利用できなくなります。この場合は、問題が修正されるまで、ホストを RHEL 8 にアップグレードしないでください。

その他のシナリオでは、以下の回避策があります。

- 新規インストールの場合は、`realm join --client-software=winbind` コマンドを使用してドメインに参加します。これにより、全ユーザーおよびグループの検索に、SSSD の代わりに Winbind を使用するようにシステムが設定されます。この場合、Samba は、`--automatic-id-mapping` オプションを **yes** (デフォルト) に設定したか、**no** に設定したかに応じて、`/etc/samba/smb.conf` の `rid` または `ad` の ID マッピングプラグインを使用します。SSSD を将来のシステムまたはその他のシステムで使用する予定がある場合は、`--automatic-id-mapping=no` を使用すると移行が容易になりますが、すべてのユーザーおよびグループ用に、AD に POSIX の UID および GID を保存する必要があります。

- 以前のバージョンの RHEL からアップグレードし、`/etc/sss/sss.conf` ファイルの `ldap_id_mapping` パラメーターが `False` に設定され、システムが ID マッピングに AD の `uidNumber` 属性および `gidNumber` 属性を使用する場合は、以下を行います。
 1. `/etc/samba/smb.conf` ファイルの `idmap config <domain> : backend = sss` エントリーを、`idmap config <domain> : backend = ad` に変更します。
 2. `systemctl status winbind` コマンドを使用して Winbind を再起動します。

(BZ#1657665)

nuxwdog サービスが HSM 環境では失敗するため、非 HSM 環境に keyutils パッケージをインストールする必要がある

`nuxwdog` ウォッチドッグサービスが証明書システムに統合されました。このため、`nuxwdog` が別のパッケージとして提供されなくなりました。ウォッチドッグサービスを使用するには、`pki-server` パッケージをインストールしてください。

`nuxwdog` サービスには、以下の既知の問題があります。

- ハードウェアストレージモジュール (HSM) を使用していると、`nuxwdog` サービスは機能しません。この問題の回避策はありません。
- 非 HSM 環境の Red Hat Enterprise Linux 8.0 では、`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ません。このパッケージを手動でインストールするには、`dnf install keyutils` コマンドを使用します。

(BZ#1652269)

AD ユーザーの ID オーバーライドの追加が IdM CLI でのみ有効

現在、管理ロールへのアクセスを付与するために、Active Directory (AD) ユーザーの ID オーバーライドを Identity Management (IdM) グループに追加すると、IdM Web UI で失敗します。この問題を回避するには、代わりに IdM コマンドラインインターフェイス (CLI) を使用します。

`ipa` ユーティリティーを使用して特定の操作を実行した後に、IdM サーバーに `ipa-idoverride-memberof-plugin` パッケージをインストールした場合、Red Hat は、`ipa` ユーティリティーのキャッシュを削除して、IdM サーバーのメタデータに関する表示を強制的に更新することを推奨します。

これを行うには、`ipa` ユーティリティーを実行するユーザーの `~/.cache/ipa` ディレクトリーの内容を削除します。たとえば、`root` の場合は以下ようになります。

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

(BZ#1651577)

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ない

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)

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)

5.5.12. コンパイラーおよび開発ツール

GCC により生成された合成関数により SystemTap が混乱する

GCC の最適化により、その他の関数を部分的にインラインにコピーした合成関数を生成する場合があります。SystemTap や GDB などのツールは、これらの合成関数と実際の関数を区別できません。これにより、SystemTap は、合成関数と実関数の両方のエントリーポイントにプローブを置くことができるため、1つの実関数呼び出しに対して、複数のプローブを数えあげます。

この問題を回避するには、SystemTap スクリプトを、再帰の検出、インライン化された部分関数に関連するプローブの非表示などの方法で調整する必要があります。たとえば、スクリプトは、以下のようになります。

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

説明されている問題は、次のように回避を試行できます。

```
global in_can_nice%

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

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

このスクリプト例では、不明な kprobes や kretprobes、または、本物の意図的な再帰など、考えられるすべてのシナリオが考慮されているわけではありません。

(BZ#1169184)

ltrace ツールが、関数呼び出しを報告しない

すべての RHEL コンポーネントに適用されるバイナリ強化の改善により、`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、各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, BZ#1655368)

5.5.13. ファイルシステムおよびストレージ

iscsiuio パッケージを使用して iSCSI ターゲットを検出できない

Red Hat Enterprise Linux 8 では、PCI レジスター領域への同時アクセスが許可されていません。そのため、**could not set host net params (err 29)** エラーが設定され、ディスカバリーポータルへの接続に失敗します。この問題を回避するには、iSCSI オフロードに、カーネルコマンドラインでカーネルパラメーター **iomem=relaxed** を設定します。これは特に、**bnx2i** ドライバーを使用したオフロードに関係します。このため、ディスカバリーポータルへの接続は成功し、**iscsiuio** パッケージが正しく機能するようになりました。

(BZ#1626629)

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う

VDO (Virtual Data Optimizer) は、プラットフォームに、ネイティブのエンディアン形式で Universal Deduplication Service (UDS) インデックスヘッダーを書き込みます。VDO は、UDS インデックスが破損し、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、そのインデックスを新しい空のインデックスで上書きします。

そのため、上書き前に UDS インデックスに保存されている重複排除アドバイスはすべて失われます。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、容量が節約されます。

(BZ#1696492)

XFS DAX マウントオプションは、共有コピーオンライトデータエクステントと互換性がない

共有コピーオンライトのデータエクステント機能でフォーマットされた XFS ファイルシステムは、**-o dax** マウントオプションと互換性がありません。これにより、そのようなファイルシステムを **-o dax** でマウントすると失敗します。

この問題を回避するには、**reflink=0** メタデータオプションを使用してファイルシステムをフォーマットし、共有コピーオンライトデータエクステントを無効にします。

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

このため、**-o dax** でファイルシステムをマウントすることに成功しました。

詳細は [Creating a file system DAX namespace on an NVDIMM](#) を参照してください。

(BZ#1620330)

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る

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#1733278)

5.5.14. ネットワーク

nftables が多次元の IP セットタイプに対応しない

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)

man ページの iptables-extensions (8) の TRACE ターゲットが nf_tables バリエーションを参照していない

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ますが、Red Hat Enterprise Linux (RHEL) 8.0 は **nf_tables** バリエーションを使用します。RHEL の **nftables** ベースの **iptables** ユーティリティーは、内部的に **meta nftrace** 式を使用します。したがって、カーネルは **TRACE** イベントをカーネルログに出力せず、代わりにユーザー空間に送信します。ただし、man ページでは、この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を参照しません。

(BZ#1658734)

RHEL 8 で、スイッチが長期間使用できなくなってから 802.3ad ボンディングのステータスが Churned と表示

現在、802.3ad ネットワークボンディングを設定し、長期間スイッチがダウンしている場合、Red Hat Enterprise Linux は、接続が稼働状態に戻った後も、ボンディングのステータスを Churned として適切に表示します。Churned のステータスは、重要なリンク停止が発生したことを管理者に通知する目的があるため、これは意図している動作です。このステータスを削除するには、ネットワークボンドを再起動するか、ホストを再起動します。

(BZ#1708807)

ebtables コマンドが broute テーブルに対応していない

Red Hat Enterprise Linux 8.0 の **nftables** ベースの **ebtables** コマンドは、**broute** テーブルに対応していません。したがって、ユーザーはこの機能を使用できません。

(BZ#1649790)

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する

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)

NetworkManager がデフォルトで 内部 DHCP プラグインを使用するようになる

NetworkManager は、DHCP プラグインの **internal** および **dhclient** に対応します。デフォルトでは、Red Hat Enterprise Linux (RHEL) 7 の NetworkManager は **dhclient** を使用し、RHEL 8 では **内部** プラグインを使用します。特定の状況では、プラグインの動作が異なります。たとえば、**dhclient** は、`/etc/dhcp/` ディレクトリーで指定されている追加設定を使用できます。

RHEL 7 から RHEL 8 にアップグレードした際に NetworkManager の動作が異なる場合は、**dhclient** プラグインを使用するために、`/etc/NetworkManager/NetworkManager.conf` ファイルの **[main]** セクションに、以下の設定を追加します。

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

(BZ#1571655)

gnome-control-center を使用して IPsec ベースの VPN の高度なオプションを変更できない

gnome-control-center アプリケーションを使用して **IPsec** ベースの **VPN** 接続を設定すると、**Advanced** ダイアログには設定が表示されますが、変更することはできません。したがって、ユーザーは高度な IPsec オプションを変更できません。この問題を回避するには、**nm-connection-editor** ツールまたは **nmcli** ツールを使用して、高度なプロパティを設定します。

(BZ#1697326)

/etc/hosts.allow ファイルおよび /etc/hosts.deny ファイルに不正確な情報が含まれる

Red Hat Enterprise Linux (RHEL) 8 では `tcp_wrappers` パッケージが削除されましたが、`/etc/hosts.allow` および `/etc/hosts.deny` のファイルは削除されません。これにより、これらのファイルには古い情報が含まれ、RHEL 8 には該当しません。

この問題を回避するには、サービスへのアクセスのフィルタリングにファイアウォールルールを使用します。ユーザー名およびホスト名に基づくフィルタリングには、アプリケーション固有の設定を使用します。

(BZ#1663556)

IP デフラグが、ネットワークトラフィックのオーバーロードでは持続できない

Red Hat Enterprise Linux 8 では、ガベージコレクションのカーネルスレッドが削除され、IP フラグメントはタイムアウト時にのみ有効期限が切れます。その結果、サービス拒否攻撃 (DoS) の CPU 使用率は大幅に低くなり、持続可能なフラグメントの最大破棄率は IP 再構築ユニットに設定されたメモリー量により制限されます。デフォルト設定では、パケットの破棄、パケットの並び替え、または断片化が同時に多数するフローが発生したときに断片化されたトラフィックを必要とするワークロードにより、関連するパフォーマンスが低下する可能性があります。

この場合は、`/proc/sys/net/ipv4` ディレクトリーの IP 断片化キャッシュの適切なチューニングを使用して **ipfrag_high_thresh** 変数を設定し、メモリー量と **ipfrag_time** 変数を制限して、IP 断片化を秒単位で保持できます。以下に例を示します。

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

上記は、IPv4 トラフィックに適用されます。IPv6 で関連するパラメーターは、`/proc/sys/net/ipv6/` ディレクトリーの **ip6frag_high_thresh** および **ip6frag_time** になります。

高速に断片化されたトラフィックに依存するワークロードでは、安定性とパフォーマンスの問題が発生する場合があります。特にパケットが破棄される場合、実稼働環境ではこのようなデプロイメントは推奨されません。

(BZ#1597671)

RHEL 8 でのネットワークインターフェイス名の変更

Red Hat Enterprise Linux 8 では、RHEL 7 と同じ一貫性のあるネットワークデバイス命名スキームがデフォルトで使用されます。ただし、一部のカーネルドライバ (**e1000e**、**nfp**、**qede**、**sfc**、**tg3**、**bnxt_en** など) では、RHEL 8 の新規インストールで一貫した名前が変更になりました。ただし、RHEL 7 からアップグレードする場合はこの名前が保持されます。

(BZ#1701968)

5.5.15. セキュリティー

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)

libssh がシステム全体の暗号化ポリシーに準拠しない

libssh ライブラリーは、システム全体の暗号化ポリシー設定には従いません。これにより、管理者が、**update-crypto-policies** コマンドを使用して暗号ポリシーレベルを変更しても、対応しているアルゴリズムのセットは変更しません。

この問題を回避するには、**libssh** を使用するアプリケーションごとに、公開された一連のアルゴリズムを個別に設定する必要があります。これにより、システムがポリシーレベルの LEGACY または FUTURE に設定されていると、**OpenSSH** と比較したときに、**libssh** を使用するアプリケーションの動作が矛盾します。

(BZ#1646563)

特定の rsyslog 優先度の文字列が正常に動作しない

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)

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)

OpenSCAP の rpmverifypackage が正常に動作しない

rpmverifypackage プロンプトにより、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ルー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)

RHEL 8 のキックスタートが、com_redhat_oscap の代わりに org_fedora_oscap を使用

キックスタートは、**com_redhat_oscap** ではなく、**org_fedora_oscap** として Open Security Content Automation Protocol (OSCAP) Anaconda アドオンを参照します。これが、混乱を招く可能性があります。これは、Red Hat Enterprise Linux 7 との後方互換性を維持するために行われます。

(BZ#1665082)

OpenSCAP の rpmverifyfile が機能しない

OpenSCAP スキャナーは、オフラインモードで現在の作業ディレクトリーを正しく変更せず、**fchdir** 関数を、OpenSCAP **rpmverifyfile** プロンプトの正しい引数で呼び出さないようにします。そのため、

SCAP コンテンツで `rpmverifyfile_test` を使用すると、`oscap-chroot` コマンドを使用した任意のファイルシステムのスキャンに失敗します。したがって、上記のシナリオでは、`oscap-chroot` が中断します。

(BZ#1636431)

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ない

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)

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ない

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。そのため、RHEL 8 では、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、`oscap-docker` または同等のユーティリティーを使用できません。

(BZ#1642373)

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ない

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)

PKCS#11 デバイスに保存されている RSA 秘密鍵と RSA-PSS 証明書を使用すると、Apache の httpd が起動しない

PKCS#11 標準は、RSA と RSA-PSS の鍵オブジェクトを区別せず、両方に `CKK_RSA` タイプを使用します。ただし、OpenSSL は、RSA 鍵および RSA-PSS 鍵に異なるタイプを使用します。その結果、`openssl-pkcs11` エンジンが、PKCS#11 RSA 鍵オブジェクトの OpenSSL に提供すべき種類を指定できません。現在、エンジンは鍵の種類を、すべての PKCS#11 `CKK_RSA` オブジェクトの RSA 鍵として設定します。OpenSSL が、証明書から取得した RSA-PSS 公開鍵の種類を、エンジンが提供する RSA 秘密鍵オブジェクトに含まれる種類と比較すると、種類が異なります。したがって、証明書と秘密鍵は一致しません。OpenSSL 関数 `X509_check_private_key()` で実行した確認は、このシナリオでエラーを返します。`httpd` の Web サーバーは、この関数をスタートアッププロセスで呼び出し、提供された証明書と鍵が一致するかどうかを確認します。この確認は、PKCS#11 モジュールに保存されている RSA-PSS 公開鍵と RSA 秘密鍵を含む証明書では常に失敗するため、`httpd` はこの設定の使用を開始できません。この問題に対する回避策はありません。

(BZ#1664802)

対応する公開鍵が PKCS#11 デバイスに保存されていない状態で ECDSA 秘密鍵を使用すると、httpd が起動しない

RSA 鍵とは異なり、ECDSA 秘密鍵には、公開鍵情報が含まれているとは限りません。この場合、ECDSA 秘密鍵から公開鍵を取得することはできません。このため、PKCS#11 デバイスは、公開鍵オブジェクトまたは証明書オブジェクトのいずれかであっても、別のオブジェクトに公開鍵情報を格納します。OpenSSL は、秘密鍵に公開鍵情報を含めるために、エンジンが提供する **EVP_PKEY** 構造を想定します。OpenSSL に提供する **EVP_PKEY** 構造を満たすと、**openssl-pkcs11** パッケージのエンジンは、一致する公開鍵オブジェクトのみから公開鍵情報を取得し、現在の証明書オブジェクトを無視します。

OpenSSL がエンジンから ECDSA 秘密鍵を要求すると、指定された **EVP_PKEY** 構造は、公開鍵を含む一致する証明書が利用可能な場合でも、PKCS#11 デバイ스에公開鍵がない場合は、公開鍵情報を含みません。これにより、Apache **httpd** の Web サーバーは、公開鍵を必要とする **X509_check_private_key()** 関数を (起動プロセスで) 呼び出すため、このシナリオで **httpd** が起動しなくなりました。この問題を回避するには、ECDSA 鍵を使用する際に、秘密鍵と公開鍵の両方を PKCS#11 デバイスに保存します。これにより、ECDSA 鍵が PKCS#11 デバイスに保存されると、**httpd** が正常に起動します。

(BZ#1664807)

OpenSSH が、ラベルが一致しない鍵の PKCS #11 の URI を処理しない

OpenSSH スイートでは、鍵のペアをラベルで識別できます。ラベルは、スマートカードに保存されている秘密鍵と公開鍵で異なる場合があります。したがって、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 で適切なオブジェクトを見つけるのを防ぐことができます。

この問題を回避するには、オブジェクト部分を使用せずに PKCS #11 の URI を指定します。これにより、OpenSSH は、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ます。

(BZ#1671262)

iptables-ebtables の出力が、ebtables と一部互換性がない

RHEL 8 では、**ebtables** コマンドは、**iptables-ebtables** パッケージが提供します。ここには、このツールが **nftables** ベースで再実装されています。このツールには別のコードベースがあり、その出力は、側面が異なる場合があるため、無視できるか、設計上の選択を慎重に検討する必要があります。

したがって、**ebtables** 出力を解析するスクリプトを移行する際に、以下を反映するスクリプトを調整します。

- MAC アドレスの書式が、長さが固定されるように変更されました。octet 値では、2 文字の書式を維持するために、必要に応じて、個々のバイト値の前にゼロが含まれます。
- IPv6 接頭辞の形式が、RFC 4291 に準拠するように変更になりました。スラッシュ文字の後ろの終了部分には、IPv6 アドレスフォーマットのネットマスクが含まなくなりましたが、接頭辞長は含まれます。スラッシュ文字の後ろの終了部分は、IPv6 アドレスフォーマットのネットマスクが含まなくなりましたが、接頭辞長を含みます。この変更は、有効な (左連続の) マスクにしか適用されませんが、それ以外の場合は、古い形式で印刷されます。

(BZ#1674536)

OpenSSH では、デフォルトで **curve25519-sha256** に対応しない

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。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。

この問題を回避するには、OpenSSH のクライアントおよびサーバーの **/etc/crypto-policies/back-**

ends/ ディレクトリーの **openssh.config** ファイルおよび **opensshserver.config** ファイルを変更して、システム全体の暗号化ポリシーの設定を手動で上書きします。この設定は、システム全体の暗号化ポリシーの変更ごとに上書きされることに注意してください。詳細は、man ページの **update-crypto-policies(8)** を参照してください。

(BZ#1678661)

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

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)

VMware ホストシステムとの SSH 接続が機能しない

現在のバージョンの **OpenSSH** スイートでは、SSH パケットでデフォルトの IPQoS (IP Quality of Service) フラグが変更しましたが、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。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。

この問題を回避するには、**ssh_config** ファイルに **IPQoS=throughput** を追加します。これにより、VMware ホストシステムとの SSH 接続が適切に機能します。

詳細は、ナレッジベースの記事 [RHEL 8 Running in VMWare Workstation Unable to Connect via SSH to Other Hosts](#) を参照してください。

(BZ#1651763)

5.5.16. サブスクリプション管理

service-level の設定と設定解除に成功したことを示すメッセージが表示されない

candlepin サービスに **syspurpose** 機能がない場合、サブスクリプションマネージャーは異なるコードパスを使用して **service-level** の引数を設定します。このコードパスは、操作の結果を表示しません。これにより、サブスクリプションマネージャーがサービスレベルを設定するときにメッセージが表示されません。これは特に、**service-level** セットに誤字がある場合や、実際に利用できない場合に問題となります。

(BZ#1661414)

syspurpose アドオン が subscription-manager attach --auto 出力に影響しない

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)

5.5.17. 仮想化

cloud-init を使用してカスタマイズされ、クローンされた起動が非常に遅い ESXi 仮想マシン

現在、**cloud-init** 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て静的 IP を使用し、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)

ネストされた仮想化ブロックのライブ移行の有効化

現在、ネストされた仮想化機能はライブ移行と互換性がありません。したがって、RHEL 8 ホストでネストされた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ります。

ネストされた仮想化は、現在 RHEL 8 でテクノロジープレビューとして提供されているため、サポート対象外であることに注意してください。さらに、ネスト化された仮想化は、デフォルトでは無効になっています。これを有効にする場合は、モジュールパラメーターの **kvm_intel.nested** または **kvm_amd.nested** を使用します。

(BZ#1689216)

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する

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)

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る

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)

virsh iface-* コマンドが一貫して動作しない

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)

Azure 用の Linux 仮想マシン拡張機能が機能しないことがある

RHEL 8 には、デフォルトで **python2** パッケージが含まれません。これにより、RHEL 8 仮想マシンで Azure 用の Linux 仮想マシン拡張機能 (**azure-linux-extensions** と呼ばれる) の実行に失敗する場合があります。

azure-linux-extensions が期待通りに機能する可能性をあげるには、手動で RHEL 8 仮想マシンに **python2** をインストールします。

```
# yum install python2
```

(BZ#1561132)

5.5.18. サポート関連

redhat-support-tool が、opencase から sosreport を自動的に収集しない

redhat-support-tool コマンドは、**sosreport** アーカイブを作成できません。この問題を回避するには、**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か、カスタマーポータル Web UI を使用します。これにより、ケースが作成され、**sosreport** がアップロードされます。

findkerneldebugs コマンド、**btextract** コマンド、**analyze**、**diagnose** コマンドは期待通り機能せず、今後のリリースで修正されることに注意してください。

(BZ#1688274)

第6章 コンテナに関する主な変更点

Red Hat Enterprise Linux (RHEL) 8.0 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。

- RHEL 8.0 には、Docker が同梱されていません。コンテナを使用するには、ツールの **podman**、**buildah**、**skopeo**、および **runc** が必要です。RHEL 8 でこのツールとコンテナを使用する方法は [コンテナの構築、実行、および管理](#) を参照してください。
- **podman** ツールが、完全に対応されるようになりました。**podman** ツールでは、1つのノードにある Pod、コンテナイメージ、およびコンテナが管理されます。これは、**libpod** ライブラリーでビルドされます。このライブラリーでは、コンテナおよびコンテナのグループ (Pod と呼ばれています) の管理が有効になります。

podman の使用方法は [コンテナの構築、実行、および管理](#) を参照してください。

- RHEL 8 GA 以降、Red Hat Universal Base Images (UBI) が新たに利用できるようになりました。UBI は、標準および最小の RHEL ベースイメージなど、以前提供されていた Red Hat イメージの一部を置き換えるものです。以前の Red Hat イメージとは異なり、UBI は自由に再配布できます。つまり、あらゆる環境で使用でき、どこでも共有できます。Red Hat のお客様でなくても使用できます。

UBI の詳細は [コンテナの構築、実行、および管理](#) を参照してください。

- RHEL 8 GA 時に、AppStream コンポーネントを提供するコンテナイメージが追加されました。このコンテナイメージは、RHEL 7 の **Red Hat Software Collections** で配布されています。このような RHEL 8 イメージは、すべて **ubi8** ベースイメージに基づいています。
- 64 ビットの ARM アーキテクチャーに対するコンテナイメージの ARM は、RHEL 8 で完全に対応されています。
- RHEL 8 では、**rhel-tools** コンテナが削除されました。**sos** ツールおよび **redhat-support-tool** ツールは、**support-tools** コンテナで提供されています。システム管理者は、このイメージを、システムツールのコンテナイメージをビルドするためのベースとして使用することもできます。
- ルートレスコンテナは、RHEL 8 ではテクノロジープレビューとして利用できます。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。

第7章 国際化

7.1. 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

7.2. 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 [glibs localization for RHEL is distributed in multiple packages](#) 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
389-ds-base	BZ#1334254, BZ#1358706
NetworkManager	BZ#1555013, BZ#1555012, BZ#1557035, BZ#1335409, BZ#1571655
PackageKit	BZ#1559414
WALinuxAgent	BZ#1561132
anaconda	BZ#1499442, BZ#1500792, BZ#1547908, BZ#1612060, BZ#1595415, BZ#1610806, BZ#1533904, BZ#1672405 , JIRA:RHELPLAN-1943, BZ#1677411, BZ#1502323, BZ#1696609
audit	BZ#1616428
authselect	BZ#1657295
bcc	BZ#1548302
bind	BZ#1588592
boom-boot	BZ#1649582
boost	BZ#1494495, BZ#1616244
cloud-init	BZ#1615599, BZ#1641190
cmake	BZ#1590139
cockpit	BZ#1619993, BZ#1631905
criu	BZ#1689746
crypto-policies	BZ#1591620, BZ#1645606, BZ#1678661 , BZ#1660839
cryptsetup	BZ#1564540
device-mapper-multipath	BZ#1643550, BZ#1673167
distribution	BZ#1516728, BZ#1516741, BZ#1566048
dnf	BZ#1622580, BZ#1647760, BZ#1581191
driverctl	BZ#1648411

コンポーネント	チケット
edk2	BZ#1536627
esc	BZ#1538645
firewalld	BZ#1509026, BZ#1648497
gcc	BZ#1169184, BZ#1607227, BZ#1535774, BZ#1504980, BZ#1571124, BZ#1246444, JIRA:RHELPLAN-7437, BZ#1652016
gdb	BZ#1491128
gdm	BZ#1589678, BZ#1641763, BZ#1678627
glib-networking	BZ#1640534
glibc	BZ#1512004, BZ#1376834, BZ#1512010, BZ#1304448, BZ#1512009, BZ#1512006, BZ#1514839, BZ#1533608
gnome-control-center	BZ#1697326
go-toolset-1.10-golang	BZ#1633351
grub2	BZ#1583445
httpd	BZ#1633224, BZ#1632754
ipa-idoverride-memberof	BZ#1651577
ipa	BZ#1664718 , BZ#1664719 , BZ#1665051
iproute	BZ#1640991, BZ#1589317
iptables	BZ#1644030, BZ#1564596, BZ#1646159, BZ#1658734 , BZ#1649790, BZ#1674536
iscsi-initiator-utils	BZ#1626629, BZ#1582099
kernel-rt	BZ#1592977

コンポーネント	チケット
kernel	BZ#1598448, BZ#1559607, BZ#1643522, BZ#1485546, BZ#1562998, BZ#1494651, BZ#1485532, BZ#1494028, BZ#1563617, BZ#1485525, BZ#1261167, BZ#1562987, BZ#1273139, BZ#1401552, BZ#1638465, BZ#1598776, BZ#1503672, BZ#1633143, BZ#1596240, BZ#1534870, BZ#1153521, BZ#1515987, BZ#1642795, BZ#1570255, BZ#1645744, BZ#1440031, BZ#1649647, BZ#1494705, BZ#1650149, BZ#1655413, BZ#1651806, BZ#1620330, BZ#1665295, BZ#1505999, BZ#1645433, BZ#1663281, BZ#1695142, BZ#1627455, BZ#1581898, BZ#1597671, BZ#1550498, BZ#1658391, BZ#1623590, BZ#1614144, BZ#1519039, BZ#1524683, BZ#1694705
kexec-tools	BZ#1520209, BZ#1662911
kmod-kvdo	BZ#1534087, BZ#1639512, BZ#1696492
ksh	BZ#1503922
libdnf	BZ#1642458, BZ#1679509
libreswan	BZ#1566574, BZ#1648776, BZ#1657854
libssh	BZ#1485241
libvirt	BZ#1528684
lksctp-tools	BZ#1568622
ltrace	BZ#1618748, BZ#1584322
lvm2	BZ#1676598 , BZ#1643543, BZ#1643545, BZ#1643547, BZ#1643549, BZ#1643562, BZ#1643576
mariadb	BZ#1637034
mdadm	BZ#1654482
mutter	BZ#1668883
net-snmp	BZ#1584510
nfs-utils	BZ#1592011, BZ#1639432
nftables	BZ#1593711
nginx	BZ#1545526

コンポーネント	チケット
nodejs-10-module	BZ#1622118
nss	BZ#1489094, BZ#1645153
nuxwdog	BZ#1652269
openldap	BZ#1570056
opensc	BZ#1595638, BZ#1595626
openscap	BZ#1614273, BZ#1618484, BZ#1646197, BZ#1636431, BZ#1618489, BZ#1642373, BZ#1618464
openssh	BZ#1622511, BZ#1228088, BZ#1645038, BZ#1671262, BZ#1651763
openssl-pkcs11	BZ#1664802 , BZ#1664807
openssl	BZ#1685470
oscap-anaconda-addon	BZ#1665082
pacemaker	BZ#1543494
pcs	BZ#1578891, BZ#1591308, BZ#1615420, BZ#1158816, BZ#1542288, BZ#1549535, BZ#1620190, BZ#1566430, BZ#1595829, BZ#1436217, BZ#1578955, BZ#1596050, BZ#1554310, BZ#1638852, BZ#1640477, BZ#1619620
perl-IO-Socket-SSL	BZ#1632600
perl	BZ#1511131
pki-core	BZ#1565073, BZ#1623444, BZ#1566360, BZ#1394069, BZ#1669257 , BZ#1656856, BZ#1673296
postgresql-9.6-module	BZ#1660041
pykickstart	BZ#1637872, BZ#1612061
python-rtslib	BZ#1666377
qemu-kvm	BZ#1559240, BZ#1508139, BZ#1497911, BZ#1578855, BZ#1651994, BZ#1621817, BZ#1508137, BZ#1592337, BZ#1570029, BZ#1689216 , BZ#1585651, BZ#1519004
redhat-release	BZ#1636338

コンポーネント	チケット
redhat-support-tool	BZ#1688274
rsyslog	BZ#1613880 , BZ#1542497 , BZ#1614179 , BZ#1619645 , BZ#1679512 , JIRA:RHELPLAN-10431
scala-2.10-module	BZ#1641744
scap-security-guide	BZ#1618505 , BZ#1618528 , BZ#1618518
scap-workbench	BZ#1640715
selinux-policy	BZ#1664345 , BZ#1594111 , BZ#1592244 , BZ#1549772 , BZ#1483904 , BZ#1626446
setup	BZ#1591969 , BZ#1663556
sos	BZ#1559836
squid	BZ#1656871
sssd	BZ#1448094 , BZ#1382750 , BZ#1446101 , BZ#1447945 , BZ#1620123 , BZ#1652562 , BZ#1659457 , BZ#1669407 , BZ#1657665
subscription-manager	BZ#1654531 , BZ#1661414
subversion	BZ#1571415
swig-3.0-module	BZ#1660051
systemd	BZ#1658691
tomcatjss	BZ#1424966 , BZ#1636564
tuned	BZ#1565598
valgrind	BZ#1500481 , BZ#1538009
varnish	BZ#1633338
vdo	BZ#1669537
virt-manager	BZ#1599777 , BZ#1643609
wpa_supplicant	BZ#1582538 , BZ#1537143

コンポーネント	チケット
xorg-x11-server	BZ#1687489, BZ#1698565
その他	JIRA:RHELPLAN-10347, BZ#1646563, JIRA:RHELPLAN-2306, BZ#1640697, BZ#1623712, BZ#1649404, BZ#1581198, BZ#1581990, BZ#1649497, BZ#1695584 , BZ#1654280, BZ#1643294, BZ#1647612, BZ#1641015, BZ#1641032, BZ#1641004, BZ#1641034, BZ#1647110, BZ#1641007, BZ#1641029, BZ#1641022, JIRA:RHELPLAN-1212, BZ#1649493, BZ#1559616, BZ#1699825 , BZ#1646541, BZ#1647725, BZ#1686057 , BZ#1582530, BZ#1581496, BZ#1650618, BZ#1650675, BZ#1650701 , JIRA:RHELPLAN-10439, JIRA:RHELPLAN-10440, JIRA:RHELPLAN-10442, JIRA:RHELPLAN-10443, JIRA:RHELPLAN-10438, JIRA:RHELPLAN-2878, JIRA:RHELPLAN-10355, JIRA:RHELPLAN-3010, JIRA:RHELPLAN-10352, JIRA:RHELPLAN-10353, JIRA:RHELPLAN-1473, JIRA:RHELPLAN-10445, JIRA:RHELPLAN-3001, JIRA:RHELPLAN-6746, JIRA:RHELPLAN-10354, JIRA:RHELPLAN-2896, JIRA:RHELPLAN-10304, JIRA:RHELPLAN-10628, JIRA:RHELPLAN-10441, JIRA:RHELPLAN-10444, JIRA:RHELPLAN-1842, JIRA:RHELPLAN-10596, JIRA:RHELPLAN-7291, JIRA:RHELPLAN-12764, BZ#1680177 , JIRA:RHELPLAN-14607, JIRA:RHELPLAN-1820, BZ#1684947 , BZ#1683712 , BZ#1659609, BZ#1504934, BZ#1642765, BZ#1641014, BZ#1692746, BZ#1687900 , BZ#1690207, BZ#1693775, BZ#1580387, BZ#1583620, BZ#1580430, BZ#1648843, BZ#1647908, BZ#1649891, BZ#1695698 , BZ#1697896, BZ#1698613 , BZ#1699535, BZ#1701968 , BZ#1704867

謝辞

RHEL 8 Readiness Challenge の一環としてフィードバックをご提供いただいた皆様、誠に有難うございました。以下の3名が特に貢献度が高かった方々です。

- Sterling Alexander
- John Pittman
- Jake Hunsaker

付録B 更新履歴

0.1-7

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。

0.1-6

2023年11月2日木曜日、Gabriela Fialová (gfialova@redhat.com)

- RHEL ドキュメントで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。

0.1-5

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。

0.1-4

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

0.1-3

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。

0.1-2

2022年3月17日(木) Lucie Maňásková (Imanasko@redhat.com)

[JIRA:RHELPLAN-14323](#)、[JIRA:RHELPLAN-14329](#)、および [JIRA:RHELPLAN-14330](#) を新機能セクション(仮想化)に追加しました。

0.1-1

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。

0.1-0

2021年9月23日(木) Lucie Maňásková (Imanasko@redhat.com)

- 無効な新機能の説明(仮想化)を削除しました。

0.0-9

2021年8月19日(木) Lucie Maňásková (Imanasko@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。

0.0-8

2021年6月23日(水) Lucie Maňásková (Imanasko@redhat.com)

- 新機能セクション(インストーラー)を更新しました。

0.0-7

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

0.0-6

2021年1月28日(木) Lucie Maňásková (Imanasko@redhat.com)

- テクノロジープレビューの章を更新。

0.0-5

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。

0.0-4

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。

0.0-3

2020年3月12日(火) Lenka Špačková (lspackova@redhat.com)

- 欠落している **postfix** RHEL システムロールをテクノロジープレビューに追加しました。

0.0-2

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。

0.0-1

2019年7月30日(火) Lucie Maňásková (Imanasko@redhat.com)

- Red Hat Enterprise Linux 8.0.1 リリースノートも併せて参照してください。

0.0-0

2019年5月7日(火) Ioanna Gkioka (igkioka@redhat.com)

- Red Hat Enterprise Linux 8.0 リリースノートも併せて参照してください。