



Red Hat Enterprise Linux 8

8.2 リリースノート

Red Hat Enterprise Linux 8.2 リリースノート

Red Hat Enterprise Linux 8 8.2 リリースノート

Red Hat Enterprise Linux 8.2 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 8.2 での改良点および実装された追加機能の概要、本リリースにおける既知の問題などを説明します。また、重要なバグ修正、テクニカルレビュー、非推奨の機能などの詳細も説明します。

目次

RED HAT ドキュメントへのフィードバック (英語のみ)	3
第1章 概要	4
インストーラーおよびイメージの作成	4
インフラストラクチャーサービス	4
セキュリティー	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
コンパイラツールセット	4
Identity Management	4
Web コンソール	5
デスクトップ	5
インプレースアップグレード	5
関連情報	6
Red Hat Customer Portal Labs	6
第2章 アーキテクチャー	7
第3章 RHEL 8 のコンテンツの配布	8
3.1. インストール	8
3.2. リポジトリ	8
3.3. アプリケーションストリーム	9
3.4. YUM/DNF を使用したパッケージ管理	9
第4章 RHEL 8.2.1 リリース	10
4.1. 新機能	10
第5章 RHEL 8.2.0 リリース	12
5.1. 新機能	12
5.2. 外部カーネルパラメーターの重要な変更	51
5.3. デバイスドライバー	57
5.4. バグ修正	59
5.5. テクノロジープレビュー	74
5.6. 非推奨になった機能	85
5.7. 既知の問題	92
第6章 国際化	120
6.1. RED HAT ENTERPRISE LINUX 8 の多言語	120
6.2. RHEL 8 における国際化の主な変更点	120
付録A コンポーネント別のチケットリスト	122
付録B 改訂履歴	129

RED HAT ドキュメントへのフィードバック (英語のみ)

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

Jira からのフィードバック送信 (アカウントが必要)

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

第1章 概要

インストーラーおよびイメージの作成

RHEL 8.2 では、パッケージのインストール前に、システムを登録し、RHEL サブスクリプションを割り当て、Red Hat コンテンツ配信ネットワーク (CDN) からインストールすることができます。インストール時に、システムを Red Hat Insights に登録することもできます。インタラクティブな GUI インストールと自動キックスタートインストールが、これらの新機能をサポートしています。

詳細は、「[インストーラーおよびイメージの作成](#)」を参照してください。

インフラストラクチャーサービス

Tuned システムチューニングツールがバージョン 2.13 にリベースされ、アーキテクチャー依存チューニングと複数の include ディレクティブのサポートが追加されました。

詳細は、「[インフラストラクチャーサービス](#)」を参照してください。

セキュリティー

システム全体の暗号化ポリシーでカスタマイゼーションがサポートされるようになりました。管理者は、完全なポリシーを定義するか、特定の値のみを変更できるようになりました。

RHEL 8.2 には、SELinux ポリシーの分析とデータフローの検査のためのツールを提供する **setools-gui** パッケージと **setools-console-analyses** パッケージが含まれています。

SCAP Security Guide が Australian Cyber Security Centre (ACSC) **Essential Eight** Maturity Model に準拠するプロファイルを提供するようになりました。

詳細は、「[セキュリティー](#)」を参照してください。

動的プログラミング言語、Web サーバー、およびデータベースサーバー

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

- Python 3.8
- Maven 3.6

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

コンパイラツールセット

RHEL 8.2 では以下のコンパイラツールセットが更新されました。

- GCC Toolset 9
- Clang and LLVM Toolset 9.0.1
- Rust Toolset 1.41
- Go Toolset 1.13

詳細は、「[コンパイラーおよび開発ツール](#)」を参照してください。

Identity Management

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

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

詳細は、「[ID 管理](#)」を参照してください。

Web コンソール

Web コンソールが、PatternFly 4 ユーザーインターフェイスシステムデザインを使用するように再設計されました。

セキュリティを強化するため、Web コンソールにセッションタイムアウトが追加されました。

詳細は、「[Web コンソール](#)」を参照してください。

デスクトップ

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

Direct Rendering Manager (DRM) カーネルグラフィックサブシステムが、アップストリームの Linux カーネルバージョン 5.3 にリベースされました。このバージョンでは、新しい GPU および APU のサポートや、さまざまなドライバー更新など、以前のバージョンから多くの機能強化が行われています。

インプレースアップグレード

RHEL 7 から RHEL 8 へのインプレースアップグレード

サポート対象のインプレースアップグレードパスは、以下のとおりです。

- 64 ビット Intel、IBM POWER 8 (little endian)、および IBM Z アーキテクチャーでの RHEL 7.9 から RHEL 8.2 のアップグレード。
- カーネルバージョン 4.14 を必要とするアーキテクチャー (64-bit ARM、IBM POWER 9 (リトルエンディアン)、および IBM Z (Structure A)) での RHEL 7.6 から RHEL 8.2 のアップグレード。上記のアーキテクチャーは RHEL 7 では完全にサポートされていますが、RHEL 7.7 以降はマイナーリリースの更新を受け取らなくなっている点に注意してください。

詳細は [Supported in-place upgrade paths for Red Hat Enterprise Linux](#) を参照してください。インプレースアップグレードの実行方法は、[RHEL 7 から RHEL 8 へのアップグレード](#) を参照してください。

主な機能強化は、次のとおりです。

- RHEL 7 から RHEL 8 へのインプレースアップグレードに、追加のカスタムリポジトリを使用できるようになりました。また、Red Hat Subscription Manager を使用せずにアップグレードすることも可能です。
- Leapp ユーティリティーを使用して、独自のアクターを作成し、カスタムまたはサードパーティーのアプリケーションを移行できます。

詳細は [Red Hat Enterprise Linux のインプレースアップグレードのカスタマイズ](#) を参照してください。

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

RHEL 6 から RHEL 8 へのインプレースアップグレード

RHEL 6.10 から RHEL 8.2 にアップグレードするには、[RHEL 6 から RHEL 8 へのアップグレード](#) の手順に従います。

CentOS Linux 6 または Oracle Linux 6 を使用している場合は、RHEL 8 にアップグレードする前に、サポート対象外の `convert2rhel` ユーティリティを使用してオペレーティングシステムを RHEL 6 に変換できます。手順は、[CentOS または Oracle Linux から RHEL に変換する方法](#) を参照してください。

関連情報

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

Red Hat Customer Portal Labs

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

- [Registration Assistant](#)
- [Product Life Cycle Checker](#)
- [Kickstart Generator](#)
- [Kickstart Converter](#)
- [Red Hat Satellite Upgrade Helper](#)
- [Red Hat Code Browser](#)
- [JVM Options Configuration Tool](#)
- [Red Hat CVE Checker](#)
- [Red Hat Product Certificates](#)
- [Load Balancer Configuration Tool](#)
- [Yum Repository Configuration Helper](#)

第2章 アーキテクチャー

Red Hat Enterprise Linux 8.2 ではカーネルバージョン 4.18.0-193 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 と同じか、より短いライフサイクルが指定されています。詳細は [Red Hat Enterprise Linux のライフサイクル](#) を参照してください。

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

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

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

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 8.2.1 リリース

Red Hat は、マイナーリリース (8.Y) において Red Hat Enterprise Linux 8 のコンテンツを四半期ごとに利用できるようにしています。この四半期リリースは、3桁の数字 (8.Y.1) を使用して番号が付けられます。RHEL 8.2.1 リリースの新機能は以下のとおりです。

4.1. 新機能

JDK Mission Control がバージョン 7.1.1 にリベースされました。

jmc:rhel8 モジュールストリームが提供する HotSpot JVM の JMC (JDK Mission Control) プロファイルが、RHEL 8.2.1 リリースでバージョン 7.1.1 にアップグレードされました。

この更新には、以下のようなバグ修正および機能拡張が数多く含まれています。

- 複数のルールの最適化
- Standard Widget Toolkit (SWT) に基づく新しい JOverflow ビュー
- 新しいフレームグラフビュー
- High Dynamic Range (HDR) ヒストグラムを使用した、レイテンシーの視覚化の新たな方法

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

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

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

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

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

(BZ#1792519)

Rust Toolset がバージョン 1.43 にリベースされました。

Rust Toolset は、バージョン 1.43 に更新されました。以下は、主な変更点です。

- 便利な行番号が、呼び出された **Option** および **Result** パニックメッセージに含まれるようになりました。
- サブスライスパターンでの一致のサポートが拡張されました。
- **matches!** マクロは、ブール値を返すパターンを提供します。
- **項目** のフラグメントは、特性、impls、および extern ブロックに挿入できます。
- プリミティブ型の推論が改善されました。
- フロートおよび整数の関連する定数。

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

■

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

使用方法は、[Using Rust Toolset](#) を参照してください。

(BZ#1811997)

コンテナーレジストリーが **skopeo sync** コマンドをサポートするようになりました。

今回の機能強化により、ユーザーは **skopeo sync** コマンドを使用して、コンテナーレジストリーとローカルレジストリーを同期できるようになりました。**skopeo sync** コマンドは、ローカルコンテナーレジストリーミラーを同期し、エアギャップ環境内で実行されるレジストリーを設定するのに役立ちます。

skopeo sync コマンドでは、ソース (**--src**) および宛先 (**--dst**) トランスポートを別々に指定する必要があります。利用可能なソースおよび宛先のトランスポートは **docker** (コンテナーレジストリーでホストされるリポジトリー) と **dir** (ローカルディレクトリーパス内のディレクトリー) です。ソーストランスポートには、**yaml** (ローカルYAMLファイルパス) も含まれます。**skopeo sync** の使用方法は、**skopeo--sync** の man ページを参照してください。

(BZ#1811779)

設定ファイル **container.conf** が利用できるようになりました。

今回の機能強化により、ユーザーおよび管理者はコンテナーエンジンのデフォルトの設定オプションとコマンドラインフラグを指定できるようになりました。コンテナーエンジンは、**/usr/share/containers/containers.conf** ファイルと **/etc/containers/containers.conf** ファイルが存在する場合は読み取ります。ルートレスモードでは、コンテナーエンジンが **\$HOME/.config/containers/containers.conf** ファイルを読み取ります。

containers.conf ファイルで指定されるフィールドは、デフォルトのオプションと、以前読み込まれた **containers.conf** ファイルのオプションを上書きします。**container.conf** ファイルは Podman と Buildah 間で共有され、**libpod.conf** ファイルに取って代わります。

(BZ#11826486)

レジストリーサーバーへのログインが可能になりました。

今回の機能強化により、**skopeo login** および **skopeo logout** コマンドを使用して、指定されたレジストリーサーバーからログインおよびログアウトできるようになりました。**skopeo login** コマンドは、ユーザー名とパスワードを標準入力から読み取ります。ユーザー名とパスワードは、**--username** (または **-u**) および **--password** (または **-p**) オプションを使用して設定することもできます。

認証ファイルのパスを指定するには、**--authfile** フラグを設定します。デフォルトのパスは **\${XDG_RUNTIME_DIR}/containers/auth.json** です。**skopeo login** および **skopeo logout** の使用方法は、**skopeo-login** と **skopeo-logout** の man ページを参照してください。

(JIRA:RHELPLAN-47311)

podman ストレージをリセットできるようになりました。

今回の機能強化により、ユーザーは **podman system reset** コマンドを使用して、**podman** ストレージを初期状態にリセットできるようになりました。**podman system reset** コマンドは、すべての Pod、コンテナー、イメージ、およびボリュームを削除します。詳細は、**podman-system-reset** man ページを参照してください。

(JIRA:RHELPLAN-48941)

第5章 RHEL 8.2.0 リリース

5.1. 新機能

ここでは、Red Hat Enterprise Linux 8.2 に追加された新機能および主要な機能強化を説明します。

5.1.1. インストーラーおよびイメージの作成

システムを登録し、RHEL サブスクリプションを割り当て、Red Hat CDN からインストールする機能

RHEL 8.2 では、パッケージのインストール前に、システムを登録し、RHEL サブスクリプションを割り当て、Red Hat コンテンツ配信ネットワーク (CDN) からインストールすることができます。インタラクティブな GUI インストールと、自動キックスタートインストールが、この機能をサポートしています。次の利点があります。

- サイズの小さい Boot ISO イメージファイルを使用することで、サイズの大きい Binary DVD ISO イメージファイルをダウンロードする必要がなくなります。
- CDN は最新のパッケージを使用するため、インストール直後に、完全にサブスクライブされた、最新のシステムになります。インストール後にパッケージの更新をインストールする必要はありません。
- 登録は、パッケージのインストール前に実行されるため、インストールプロセスが短く、簡素化されます。
- Red Hat Insights の統合サポートが利用可能です。

(BZ#1748281)

インストール時にシステムを Red Hat Insights に登録する機能

RHEL 8.2 では、インストール時にシステムを Red Hat Insights に登録することができます。インタラクティブな GUI インストールと、自動キックスタートインストールが、この機能をサポートしています。

次の利点があります。

- 事業運営に影響が及ぶ前に、問題を特定し、優先順位を付け、解決することが容易になります。
- 予測分析により、プロアクティブにセキュリティー、パフォーマンス、可用性、および安定性に対する脅威を特定して修復します。
- お使いの環境における問題と予定外のダウンタイムを回避します。

(BZ#1746391)

Image Builder では、Azure イメージを作成するために cloud-init サポートが利用できるようになりました。

今回の機能強化により、Image Builder で作成された Azure イメージで cloud-init サポートが利用できるようになりました。その結果、顧客に対し、迅速なプロビジョニング付きのオンプレミスイメージの作成、およびカスタムデータの追加機能が利用可能になりました。

(BZ#1754711)

新しいキックスタートコマンドを追加: rhsm および zipl

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。

- **rhsm:** **rhsm** コマンドを使用して、インストール中にシステムを Red Hat に登録します。
- **zipl:** **zipl** コマンドを使用して、IBM Z システムで **zipl** 設定を指定します。

(BZ#1972214)

5.1.2. ソフトウェア管理

User-Agent ヘッダー文字列に **/etc/os-release** ファイルから読み込んだ情報を含むようになりました。

今回の機能強化により、通常は DNF が作成する HTTP リクエストに含まれる **User-Agent** ヘッダー文字列が、**/etc/os-release** ファイルから読み込む情報に拡張されました。

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

(BZ#1676891)

すべての **dnf-automatic.timer** タイマーユニットが、デフォルトでリアルタイムクロックを使用するようになりました。

従前は、**dnf-automatic.timer** タイマーユニットはモノトニッククロックを使用していました。これにより、システムの起動後の予期せぬアクティベーションが生じていました。今回の更新で、タイマーユニットが午前 6 時から午前 7 時の間に実行されるようになりました。この時間帯にシステムがオフになっている場合、タイマーユニットはシステムの起動後 1 時間以内にアクティベートされます。

(BZ#1754609)

createrepo_c ユーティリティーが、そのメタデータに認められない制御文字を含むパッケージをスキップするようになりました。

有効な XML を確保するために、パッケージメタデータには、以下の例外を除き、制御文字を含めることはできません。

- 水平タブ
- 改行文字
- キャリッジリターン文字

今回の更新で、**createrepo_c** ユーティリティーが新たに作成されたりポジトリで認められない制御文字を含むメタデータをパッケージに含まなくなり、次のエラーメッセージを返すようになりました。

```
C_CREATEREPOLIB: Critical: Cannot dump XML for PACKAGE_NAME (PACKAGE_SUM):  
Forbidden control chars found (ASCII values <32 except 9, 10 and 13)
```

(BZ#1743186)

5.1.3. シェルおよびコマンドラインツール

opencv がバージョン 3.4.6 にリベースされました。

opencv パッケージがアップストリームバージョン 3.4.6 にアップグレードされました。以下は、主な変更点です。

- **OPENCV_OPENCL_BUILD_EXTRA_OPTIONS** および **OPENCV_OPENCL_DEVICE_MAX_WORK_GROUP_SIZE** などの新しい Open CL パラメーターのサポートが追加されました。
- **objdetect** モジュールが QR コード検出アルゴリズムに対応するようになりました。
- **MatSize::dims** または **VideoCapture::getBackendName** などの複数の新規メソッドが追加されました。
- **drawFrameAxes** または **getVersionMajor** などの新規関数が複数追加されました。
- SSSE3 命令の使用時に **GaussianBlur** 関数、**v_load_deinterleave** および **v_store_interleave** 組み込み関数の改善機能など、さまざまなパフォーマンスが改善されました。

(BZ#1694647)

5.1.4. インフラストラクチャーサービス

graphviz-python3 が CRB リポジトリで配布されるようになりました。

今回の更新で、**graphviz-python3** パッケージが RHEL 8 に追加されました。このパッケージは、Python から Graphviz グラフ描画ソフトウェアを使用するのに必要なバインディングを提供します。

graphviz-python3 パッケージは、サポート対象外の [CodeReady Linux Builder リポジトリ \(CRB\)](#) で配布されている点に留意してください。

(BZ#1704875)

tuned がバージョン 2.13.0 にリベースされました。

tuned パッケージがアップストリームバージョン 2.13.0 にアップグレードされました。主な機能強化は、次のとおりです。

- アーキテクチャーに依存したチューニングフレームワークが追加されました。
- 複数の include ディレクティブのサポートが追加されました。
- **sap-hana**、**latency-performance**、および **realtime** プロファイルのチューニングが更新されました。

(BZ#1738250)

powertop がバージョン 2.11 にリベースされました。

powertop パッケージがバージョン 2.11 にアップグレードされました。これは、以下の主要な変更を提供するものです。

- EHL、TGL、ICL/ICX プラットフォームのサポート

(BZ#1716721)

BIND が **GeoLite Legacy GeoIP** に代わって **.GeoIP2** に対応するようになりました。

BIND では、**GeoLite Legacy GeoIP** ライブラリーに対応しなくなりました。今回の更新で、**GeoLite Legacy GeoP** が **GeoIP2** に置き換えられました。これは、**libmaxminddb** データ形式で提供されます。

新しい形式では、設定変更が必要になる場合があること、また、この形式では、以下のレガシー GeoIP アクセス制御リスト (ACL) 設定をサポートしないことに留意してください。

- `geoip netspeed`
- `geoip org`
- ISO 3166 Alpha-3 国名コード

(BZ#1564443)

stale-answer が、DDoS 攻撃の場合に古いキャッシュレコードを提供するようになりました。

以前は、分散型サービス拒否 (DDoS) 攻撃により、権威サーバーが SERVFAIL エラーで失敗していました。今回の更新で、**stale-answer** 機能が新しいレスポンスを取得するまで期限切れのレコードを提供するようになりました。

serve-stale 機能を有効または無効にするには、次のいずれかを使用します。

- 設定ファイル
- リモート制御チャンネル (`rndc`)

(BZ#1664863)

BIND がバージョン 9.11.13 にリベースされました。

bind パッケージが、バージョン 9.11.13 にアップグレードされました。以下は、主な変更点です。

- **tcp-highwater** 統計情報変数が追加されました。この変数は、実行時に記録された同時 TCP クライアントの最大数を表示します。
- **SipHash-2-4** ベースの DNS Cookies (RFC 7873) アルゴリズムが追加されました。
- **minimal-responses** 設定オプションがどのように設定されているかに関わらず、プライミングクエリーをルーティングするためのグルーアドレスが返されます。
- **named-checkconf** コマンドが、**DNS64** ネットワーク接頭辞の有効性を確保するようになりました。
- **trusted-keys** および **managed-keys** ステートメントが両方とも同じ名前に設定されている場合、RFC 5011 ごとの自動ロールオーバーに失敗することがなくなりました。代わりに、警告メッセージがログに記録されます。
- **dig** および **nslookup** ユーティリティーにおける国際化ドメイン名 (IDN) について、処理が端末で実行されない場合 (スクリプトで実行されるなどの場合)、デフォルトで無効になりました。**dig** における IDN 処理は、**+idnin** および **+idnout** オプションを使用して切り替えることができます。

(BZ#1704328)

5.1.5. セキュリティー

RHEL 8 に **DISA STIG** プロファイルが追加されました。

セキュリティ技術実装ガイド (Security Technical Implementation Guide (STIG)) は、DISA (Defense Information Systems Agency) が発行する一連のベースライン推奨事項で、この推奨事項なしでは脆弱となる可能性のある情報システムおよびソフトウェアのセキュリティを強化します。今回のリリース

には、このセキュリティーポリシーのプロファイルおよびキックスタートファイルが含まれます。今回の機能強化により、Red Hat Enterprise Linux 8 のシステムにおけるコンプライアンスの確認、システムの修正、および DISA STIG に準拠するシステムのインストールが可能になります。

(BZ#1755447)

crypto-policies がカスタマイズ可能になりました。

今回の更新で、あらゆるポリシーレベルの特定のアルゴリズムまたはプロトコルを調整することや、現在のシステム全体の暗号化ポリシーとして新しい完全なポリシーファイルを設定することができるようになりました。これにより、管理者はさまざまなシナリオごとに必要に応じて、システム全体の暗号化ポリシーをカスタマイズできます。

RPM パッケージは、それらが提供するポリシーを `/usr/share/crypto-policies/policies` ディレクトリーに保存する必要があります。`/etc/crypto-policies/policies` ディレクトリーには、ローカルのカスタムポリシーが含まれます。

詳細は、man ページの **update-crypto-policies(8) の Custom Policies** セクション、および man ページの **update-crypto-policies(8) の Crypto Policy Definition Format** セクションを参照してください。

(BZ#1690565)

SCAP セキュリティーガイドが ACSC Essential Eight に対応するようになりました。

scap-security-guide パッケージで、Australian Cyber Security Centre (ACSC) Essential Eight 準拠のプロファイルと、これに対応するキックスタートファイルを利用できるようになりました。今回の機能強化により、このセキュリティーベースラインに準拠するシステムをインストールできるようになりました。さらに、ACSC が定義する、この最小セキュリティー制御の仕様を使用してセキュリティーコンプライアンスと修正を確認するために **OpenSCAP** スイートを利用することができます。

(BZ#1755194)

コンテナのセキュリティースキャンおよびコンプライアンススキャンを行う **oscap-podman が利用できるようになりました。**

この **openscap** パッケージの更新により、コンテナのセキュリティースキャンおよびコンプライアンススキャン用に新しいユーティリティーが導入されました。**oscap-podman** ツールが、RHEL 7 におけるコンテナおよびコンテナイメージのスキャンを提供する **oscap-docker** ユーティリティーと同等のものを提供します。

(BZ#1642373)

setroubleshoot が **execmem アクセス拒否を分析して、これに反応できるようになりました。**

今回の更新で、新しい **setroubleshoot** プラグインが導入されました。このプラグインは **execmem** のアクセス拒否 (AVC) を分析し、適切なアドバイスを提供できます。その結果、**setroubleshoot** は、アクセスが許可される場合にはブール値の切り替えを示唆でき、ブール値でアクセスを許可できない場合には問題を報告できるようになりました。

(BZ#1649842)

新しいパッケージ、setools-gui** および **setools-console-analyses****

RHEL 7 に含まれていた **setools-gui** パッケージが、RHEL 8 に導入されています。グラフィカルツールは、とりわけ特殊な SELinux ポリシーを持つマルチレベルシステムで関係やデータフローを検査する際に役立ちます。**setools-gui** パッケージの **apol** グラフィカルツールで、SELinux ポリシーの要素を検査し、分析することができます。**setools-console-analyses** パッケージのツールを使用すると、ドメイン移行と SELinux ポリシー情報のフローを分析できます。

(BZ#1731519)

SELinux の制限のあるユーザーが、ユーザーセッションサービスを管理できるようになりました。

以前は、制限のあるユーザーはユーザーセッションサービスを管理できませんでした。その結果、**systemctl --user** コマンドまたは **busctl --user** コマンドを実行することや、RHEL Web コンソールで作業することができませんでした。今回の更新で、制限のあるユーザーがユーザーセッションを管理できるようになりました。

(BZ#1727887)

lvmdbusd サービスが SELinux で制限されました。

lvmdbusd サービスが D-Bus API を論理ボリュームマネージャー (LVM) に提供します。以前は、**lvmdbusd** デーモンは、**lvm_t** の SELinux ポリシーが定義されていても、**lvm_t** コンテキストに移行することができませんでした。これにより、**lvmdbusd** デーモンは **unconfined_service_t** ドメインで実行され、SELinux は **lvmdbusd** を非制限としてラベル付けしていました。今回の更新で、**lvmdbusd** 実行ファイルが定義済みの **lvm_exec_t** コンテキストを持つようになり、強制モードの SELinux で **lvmdbusd** を正常に使用できるようになりました。

(BZ#1726166)

semanage が、SCTP ポートと DCCP ポートのリッスンと変更をサポートするようになりました。

以前は、**semanage port** では、リッスンおよび変更ができるのは TCP ポートと UDP ポートのみでした。今回の更新で、SCTP および DCCP プロトコルのサポートが **semanage port** に追加されています。これにより、管理者は、両マシンが SCTP 経由で通信できるかどうかを確認し、SCTP 機能を完全に有効にして SCTP ベースのアプリケーションを正常にデプロイできるようになりました。

(BZ#1563742)

semanage export が、**permissive** ドメインに関連するカスタマイゼーションを表示するようになりました。

今回の更新で、**semanage** ユーティリティー (SELinux の **policycoreutils** パッケージの一部) が **permissive** ドメインに関連するカスタマイゼーションを表示できるようになりました。システム管理者が、**semanage export** コマンドを使用して、マシン間で許容ローカル変更を転送できるようになりました。

(BZ#1417455)

udica が、SELinux の拒否から生成された新しい許可ルールを既存のコンテナポリシーに追加できるようになりました。

udica ユーティリティーにより生成されたポリシーの下で動作するコンテナが SELinux の拒否をトリガーする場合、**udica** がポリシーを更新できるようになりました。新しいパラメーター **-a** または **--append-rules** を使用して AVC ファイルからルールを追加できます。

(BZ#1732704)

新しい SELinux タイプが追加され、サービスが制限のある状態で実行できるようになりました。

今回の更新では、新しい SELinux タイプが導入され、**unconfined_service_t** ドメインで実行するのではなく、以下のサービスが SELinux Enforcing モードで制限のあるサービスとして実行できるようになりました。

- **lldpd** が **lldpad_t** として実行されるようになりました。
- **rrdcached** が **rrdcached_t** として実行されるようになりました。
- **stratisd** が **stratisd_t** として実行されるようになりました。
- **timedatex** が **timedatex_t** として実行されるようになりました。

([BZ#1726246](#), [BZ#1726255](#), [BZ#1726259](#), [BZ#1730204](#))

Clevis で、所定の LUKS デバイス用のポリシーをリスト表示できます。

今回の更新で、**clevis luks list** コマンドが、所定の LUKS デバイス用の PBD ポリシーをリスト表示するようになりました。これにより、使用中の Clevis ピンやピン設定 (Tang サーバーアドレス、**tpm2** ポリシーの詳細、SSS しきい値など) に関する情報を簡単に見つけることができます。

([BZ#1766526](#))

Clevis で、キーのステータスを報告し、期限切れのキーを再バインドする新しいコマンドが提供されます。

clevis luks report コマンドで、特定のバインディングのキーがローテーションを要するかどうかを報告させることが簡単にできるようになりました。Tang サーバーにおける通常のキーローテーションにより、Network-Bound Disk Encryption (NBDE) デプロイメントのセキュリティが強化されるため、クライアントは期限切れのキーの検出を提供する必要があります。キーの有効期限が切れると、Clevis は **clevis luks regen** コマンドを使用して現在のキーで期限切れのキースロットを再バインドするよう推奨します。これにより、キーローテーションのプロセスが大幅に簡素化されます。

([BZ#1564559](#), [BZ#1564566](#))

Clevis が、LUKS デバイスの特定のスロットをバインドするために使用されるパスフレーズを抽出できるようになりました。

Clevis ポリシーベースの復号化フレームワークに対する今回の更新で、LUKS デバイスの特定のスロットをバインドするために使用されるパスフレーズを抽出できるようになりました。以前は、LUKS インストールパスフレーズが消去されると、Clevis は、再暗号化、ユーザーパスフレーズを使用した新しいキースロットの有効化、管理者が **sss** しきい値を変更する必要がある場合の Clevis の再バインドなどの、LUKS 管理タスクを実行できませんでした。今回の更新で、特定のスロットをバインドするために使用されるパスフレーズを示す **clevis luks pass** コマンドが導入されています。

([BZ#1436780](#))

Clevis ではシステムの起動時に複数の LUKS デバイスを復号化するサポートを改善されました。

clevis パッケージが更新され、システムの起動時に複数の LUKS 暗号化デバイスを復号するサポートが改善されました。この機能が改善される前は、管理者がシステム設定に複雑な変更を加えて、起動時に Clevis が複数のデバイスを適切に復号化できるようにする必要がありました。今回のリリースでは、**clevis luks bind** コマンドを使用して復号化を設定し、**dracut -fv --regenerate-all** コマンドで **initramfs** を更新できます。

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#1784524](#))

openssl-pkcs11 が 0.4.10 にリベースされました。

openssl-pkcs11 パッケージがアップストリームバージョン 0.4.10 にアップグレードされ、以前のバー

ジョンに対するバグ修正や機能強化が数多く追加されました。**openssl-pkcs11** パッケージでは、エンジンインターフェイスを介した PKCS #11 モジュールへのアクセスが提供されます。新しいバージョンで導入された主な変更点は以下のとおりです。

- ECDSA 秘密鍵の読み込み時に秘密鍵に対応する公開鍵オブジェクトが利用できない場合、エンジンは一致する証明書 (ある場合) から公開鍵を読み込みます。
- **openssl-pkcs11** エンジンは所定の PKCS #11 URI に一致するすべてのトークンを検索するため、一般的な PKCS #11 URI (**pkcs11:type=public** など) を使用できます。
- システムは、単一デバイスが URI 検索と一致する場合にのみ PIN でのログインを試みます。これにより、PIN を一致するすべてのトークンに提供することで、認証の失敗を防ぎます。
- デバイスへのアクセス時に、**openssl-pkcs11** エンジンが、RSA メソッド構造に **RSA_FLAG_FIPS_METHOD** フラグを付けるようになりました。FIPS モードでは、OpenSSL は、RSA メソッド構造でフラグを設定する必要があります。エンジンは、デバイスが FIPS で認定されているかどうかを検出できないことに留意してください。

(BZ#1745082)

rsyslog が 8.1911.0 にリベースされました。

rsyslog ユーティリティーがアップストリームバージョン 8.1911.0 にアップグレードされ、以前のバージョンに対するバグ修正や機能強化が数多く追加されました。以下は、主な機能強化のリストです。

- 新しい **omhttp** モジュールを使用すると、HTTP REST インターフェイスを介してメッセージを送信できます。
- ファイル入力モジュールが強化され、安定性、エラー報告、およびトランケーション検出が改善されています。
- あらゆるアクションで使用できる新しい **action.resumeIntervalMax** パラメーターにより、再試行間隔の増加に特定の値の上限を付けることができます。
- TLS の新しい **StreamDriver.PermitExpiredCerts** オプションは、証明書が期限切れの場合でも接続を許可します。
- 設定された外部ファイルのコンテンツに基づいて、出力を一時停止および再開できるようになりました。これは、反対側が常にメッセージを許可するが、すべてのメッセージを処理できない場合には断りなくドロップするという状況で有用です。
- ファイル出力モジュールのエラー報告が改善され、実際のファイル名と、エラーの原因に関する詳細情報が含まれるようになりました。
- ディスクキューがマルチスレッドを実行するようになり、パフォーマンスが向上します。
- さらに厳格な TLS 操作モードを設定できます。**extendedKeyUsage** 証明書フィールドを確認するモードと、**CN/SAN** 証明書フィールドをより厳格に確認するモードがあります。

(BZ#1740683)

rsyslog で、HTTP REST インターフェイスを介して通信するための **omhttp** プラグインが提供されるようになりました。

rsyslog パッケージの今回の更新で、Ceph ストレージプラットフォーム、Amazon Simple Storage Service (Amazon S3)、Grafana Loki などの、Representational State Transfer (REST) API を使用するサービスと互換性のある出力を生成する新しい **omhttp** プラグインを使用できるようになりました。こ

の新しい HTTP 出力モジュールでは、設定可能な REST パスおよびメッセージ形式、複数のバッチ形式のサポート、圧縮、および TLS 暗号化が提供されます。

詳細は、**rsyslog-doc** パッケージでシステムにインストールされた `/usr/share/doc/rsyslog/html/configuration/modules/omhttp.html` ファイルを参照してください。

(BZ#1676559)

rsyslog の omelasticsearch が rebindinterval に対応するようになりました。

rsyslog パッケージの今回の更新で、**omelasticsearch** モジュールにおける定期的な再接続の時間の設定がサポートされるようになりました。このパラメーターをシナリオに応じて設定すると、Elasticsearch ノードのクラスターにレコードを送信する場合のパフォーマンスを向上させることができます。**rebindinterval** パラメーターの値は、**rsyslog** が接続を閉じてから新しい接続を確立するまでにノードに送信された操作の数を示します。デフォルト値の **-1** は、**rsyslog** が接続を再確立していないことを意味します。

(BZ#1692073)

rsyslog mmkubernetes がメタデータキャッシュの有効期限を提供するようになりました。

rsyslog パッケージの今回の更新で、メタデータキャッシュの有効期限を設定するために、**mmkubernetes** モジュールの 2 つのパラメーターを新しく使用できるようになりました。これにより、削除された Kubernetes オブジェクトは、**mmkubernetes** 静的キャッシュから確実に削除されます。**cacheentryttl** パラメーターの値は、キャッシュエントリーの最大期間 (秒単位) を示します。**cacheexpireinterval** パラメーターの値は、次のとおりです。

- キャッシュ有効期限チェックを無効にする場合は **-1**
- キャッシュ有効期限チェックを有効にする場合は **0**
- 定期的なキャッシュ有効期限チェック (秒単位) の場合は、0 よりも大きい数

(BZ#1692072)

audit がバージョン 3.0-0.14 にリベースされました。

audit パッケージがアップストリームバージョン 3.0-0.14 にアップグレードされ、以前のバージョンに対するバグ修正や機能強化が数多く追加されました。主要な変更点は次のとおりです。

- **syslog** プラグインのフィールドを解釈するオプションが追加されました。
- **30-ospp-v42.rules** ファイルが、さらに細分されました。
- サンプルルールが `/usr/share/audit/sample-rules/` ディレクトリーに移動しました。
- リモートロギング用の Audit KRB5 トランスポートモードが修正されました。

(BZ#1757986)

Audit に、カーネル v5.5-rc1 の多くの改善点が含まれるようになりました。

Linux カーネルに対する今回の追加には、Audit サブシステムに関連し、バージョン 4.18 と 5.5-rc1 の間に導入された、多くの機能強化、バグ修正、およびクリーンアップが含まれます。以下のリストは、重要な変更点を抜粋したものです。

- フィルタリングで **exe** フィールドの使用を拡大
- v3 名前空間機能のサポート

- リモートファイルシステムでのフィルタリングの改善
- **gid** フィルタールールの修正
- メモリー解放後の使用における破損およびメモリーリークの修正
- イベントレコードの関連付けの改善
- **fanotify** インターフェイス、Audit 設定オプション、および **syscall** インターフェイスのクリーンアップ
- Extended Verification Module (EVM) の戻り値の修正
- 複数のレコード形式の修正とクリーンアップ
- 仮想ファイルシステム (VFS) 監査の簡素化および修正

(BZ#1716002)

fapolicyd が 0.9.1-2 にリベースされました。

RHEL アプリケーションのホワイトリスト化を提供する **fapolicyd** パッケージが、アップストリームバージョン 0.9.1-2 にアップグレードされました。主なバグ修正と機能強化は、以下のとおりです。

- プロセス ID が修正されました。
- サブジェクト部分とオブジェクト部分が、ルールで厳密に配置されるようになりました。いずれの部分もコロンで区切られており、必要なパーミッション (execute, open, any) が含まれています。
- サブジェクト属性とオブジェクト属性が統合されています。
- 新しいルールの形式は以下のとおりです。

```
DECISION PERMISSION SUBJECT : OBJECT
```

以下に例を示します。

```
allow perm=open exe=/usr/bin/rpm : all
```

(BZ#1759895)

sudo が 1.8.29-3.el8 にリベースされました。

sudo パッケージがアップストリームバージョン 1.8.29-3 にアップグレードされ、以前のバージョンに対するバグ修正や機能強化が数多く追加されました。新しいバージョンで導入された主な変更点は以下のとおりです。

- **sudo** が、利用可能な場合には、標準出力や標準エラー出力ではなく、ユーザー端末に Pluggable Authentication Module (PAM) メッセージを書き出すようになりました。これにより、ファイルおよびパイプに送信された PAM 出力とコマンドの出力の混乱が回避されます。
- LDAP および SSSD からの **notBefore** および **notAfter** オプションが、**sudo -l** command コマンドで、正常に動作、表示するようになりました。
- **cvtsudoers** コマンドが、LDIF から **sudoers** および JSON 形式への変換時に LDAP Data Interchange Format (LDIF) 以外の入力を拒否するようになりました。

- **sudoers** の新しい **log_allowed** および **log_denied** 設定を使用すると、許可されたコマンドと拒否されたコマンドのロギングおよび監査を無効にできます。
- **runas_spec** 仕様にグループが一切存在しない場合であっても、**sudo** で **-g** オプションを使用して、ターゲットユーザーグループに一致するグループを指定できるようになりました。以前は、グループがターゲットユーザーのプライマリーグループと一致する場合にのみ、これを実行できました。
- **sudo** で **sssd.conf** の **ipa_hostname** の値にホスト名を一致させることを妨げていたバグを修正しました。
- **Runas** 仕様が **ALL** キーワードでの **root** アクセスを認めていない場合に **sudo** ユーザーが **root** としてコマンドを実行できるという脆弱性が修正されました (CVE-2019-14287)。
- **ALL** キーワードを使用するなど、許容 **sudoers** エントリーに対する不明なユーザーおよびグループ ID の使用が無効になりました。**runas_allow_unknown_id** 設定で有効にできます (CVE-2019-19232)。

(BZ#1733961)

pam_namespace モジュールで **tmpfs** の追加マウントオプションを指定可能になりました。

/etc/security/namespace.conf 設定ファイルでは、さまざまなオプションが利用できるようになりました。setuid ビットの効果を無効にする **nosuid** マウントオプション。実行可能ファイルの実行を無効にする **noexec** マウントオプション。ならびにマウントされた **tmpfs** ファイルシステムでファイルが文字列またはブロックデバイスとして解釈されるのを防ぐ **nODEV** マウントオプション。

その他のマウントオプションは、**tmpfs(5)** man ページで指定されています。

(BZ#1252859)

pam_faillock が、**faillock.conf** 設定ファイルから設定を読み込めるようになりました。

プラグ可能な認証モジュール (PAM) の一部である **pam_faillock** モジュールが、**/etc/security/faillock.conf** に位置する設定ファイルから設定を読み込めるようになりました。これにより、**faillock.conf** ファイルを編集するだけで、認証失敗時にアカウントのロックアウトを設定し、この機能にユーザープロファイルを提供し、さまざまな PAM 設定を処理することが簡単にできるようになります。

(BZ#1537242)

5.1.6. ネットワーク

ユーザー空間のアプリケーションが、カーネルで選択した **netns** ID を取得できるようになりました。

ユーザー空間のアプリケーションは、カーネルに新しい **netns** ID を選択するよう要求し、これをネットワーク名前空間に割り当てることができます。今回の機能強化により、ユーザーはカーネルに **RTM_NETNSID netlink** メッセージを送信するときに、**NLM_F_ECHO** フラグを指定できるようになりました。これで、カーネルが **netlink** メッセージをユーザーに送り戻します。このメッセージには、カーネルが選択した値に設定された **netns** ID が含まれています。その結果、ユーザー空間アプリケーションには、カーネルが選択した **netlink** ID を識別するための信頼できるオプションが追加されました。

(BZ#1763661)

firewalld がバージョン 0.8 にリベースされました。

firewalld パッケージが、バージョン 0.8 に更新されました。以下は、主な変更点です。

- この **firewalld** のバージョンには、バージョン 0.7.0 以降のすべてのバグ修正が含まれます。
- **firewalld** が **nftables** サブシステムへの **libnftables** JSON インターフェイスを使用するようになりました。これにより、ルールアプリケーションのパフォーマンスおよび信頼性が向上します。
- サービス定義では、新しい **helper** 要素が **module** を置き換えます。
- このバージョンでは、カスタムヘルパーが標準のヘルパーモジュールを使用できます。

(BZ#1740670)

ndptool が IPv6 ヘッダーで宛先アドレスを指定できるようになりました。

今回の更新で、**ndptool** ユーティリティーが、IPv6 ヘッダーでアドレスを指定することで、特定の宛先に Neighbor Solicitation (NS) または Neighbor Advertisement (NA) メッセージを送信できるようになりました。その結果、メッセージをリンクローカルアドレス以外のアドレスに送信できます。

(BZ#1697595)

nftables が、多次元 IP セットタイプをサポートするようになりました。

今回の機能強化により、**nftables** パケットフィルタリングフレームワークが、連結と区間を持つセットタイプをサポートするようになりました。その結果、管理者は、多次元 IP セットタイプを作成するための回避策が不要になりました。

(BZ#1593711)

nftables がバージョン 0.9.3 にリベースされました

nftables パッケージがアップストリームバージョン 0.9.3 にアップグレードされ、以前のバージョンに対するバグ修正や機能強化が数多く追加されました。

- JSON API が **libnftables** ライブラリーに追加されました。このライブラリーは、サードパーティーアプリケーションから **nftables** ルールセットを管理する高度なインターフェイスを提供します。Python で新しい API を使用するには、**python3-nftables** パッケージをインストールします。
- ステートメントは、**192.0.2.0/24** および **192.0.2.0-192.0.2.30** などの IP 接頭辞および範囲をサポートします。
- オペレーティングシステムのフィンガープリントのサポートが追加され、予想されるオペレーティングシステムに基づいてパケットをマークできるようになりました。詳細は、**nft(8)** man ページの **osf expression** セクションを参照してください。
- パケットヘッダーを変更せずにパケットをローカルソケットにリダイレクトするように、透過型プロキシ機能が追加されました。詳細は、**nft(8)** man ページの **tproxy statement** セクションを参照してください。
- デフォルトでは、**nft** は、nft チェーンの作成中に優先順位セットのテキスト名を表示します。標準優先度の数値を表示するには、**-y** オプションを使用します。
- セキュリティマークのサポートが追加されました。
- 動的セット更新のサポートが改善され、パケットパスから更新を設定するようになりました。

- トランスポートヘッダーポートマッチのサポートが追加されました。

主な変更の詳細は、更新前にアップストリームのリリースノートを参照してください。

- <https://lore.kernel.org/netfilter-devel/20190624164910.defehs5giqziqnir@salvia/>
- <https://lore.kernel.org/netfilter-devel/20190819115807.myv6owxzbj2bthd@salvia/>
- <https://lore.kernel.org/netfilter-devel/20191202211737.xvmd6e6xxj4xvvl@salvia/>

(BZ#1643192)

firewalld サービスに対するルールが、標準ポート以外で動作するサービスについて接続追跡ヘルパーを使用できるようになりました。

firewalld サービス内のユーザー定義ヘルパーが、標準のカーネルヘルパーモジュールを使用できるようになりました。これにより、管理者は、非標準ポートで動作するサービスについて接続追跡ヘルパーを使用する **firewalld** ルールを作成することができます。

(BZ#1733066)

whois パッケージが利用できるようになりました。

今回の機能強化により、RHEL 8.2.0 で **whois** パッケージが利用できるようになりました。その結果、特定のドメイン名または IP アドレスについての情報を取得できるようになりました。

(BZ#1734183)

tc の eBPF が完全にサポートされました。

Traffic Control (tc) カーネルサブシステムと **tc** ツールは、extended Berkeley Packet Filtering (eBPF) 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することができます。これにより、カーネルネットワークデータパス内でプログラム可能なパケット処理が可能になりました。以前はテクノロジープレビューとして利用できた **tc** 用の eBPF は、RHEL 8.2 で完全に対応になりました。

(BZ#1755347)

5.1.7. カーネル

RHEL 8.2 のカーネルバージョン

Red Hat Enterprise Linux 8.2 には、カーネルバージョン 4.18.0-193 が同梱されています。

[外部カーネルパラメーターの重要な変更](#) および [デバイスドライバー](#) も参照してください。

(BZ#1797671)

RHEL 8.2 向け Extended Berkeley Packet Filter

extended Berkeley Packet Filter (eBPF) は、限られた一連の関数にアクセスできる制限付きサンドボックス環境において、カーネル領域でのコード実行を可能にするカーネル内の仮想マシンです。この仮想マシンは、特別なアセンブリのようなコードを実行します。**eBPF** バイトコードが最初にカーネルにロードされ、その後に検証が行われます。次に実行時のコンパイルでコードがネイティブマシンコードに変換され、その後、仮想マシンがコードを実行します。

Red Hat は、**eBPF** 仮想マシンを使用するコンポーネントを数多く提供しています。各コンポーネントの開発フェーズはさまざまです。そのため、現在すべてのコンポーネントが完全にサポートされている訳ではありません。RHEL 8.2 では、以下の **eBPF** コンポーネントがサポートされています。

- **BPF Compiler Collection (BCC)** ツールパッケージ。これは、効率的なカーネル追跡とプログラムの操作に **eBPF** 仮想マシンを使用する、動的カーネル追跡ユーティリティのユーザー空間コレクションです。**BCC** は、**eBPF** を使用して、Linux オペレーティングシステムの I/O 分析、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。
- **BCC** ライブラリー。これを使用すると、**BCC** ツールパッケージで提供されるツールと同様のツールを開発できます。
- **eBPF for Traffic Control (tc)**機能。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。

特定のコンポーネントがサポート対象と示されていない限り、その他のすべての **eBPF** コンポーネントはテクノロジープレビューとして提供されます。

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

- **bpfftrace** トレース言語
- **eXpress Data Path (XDP)**機能

テクノロジープレビューのコンポーネントに関する詳細情報は、[テクノロジープレビュー](#) を参照してください。

([BZ#1780124](#))

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

Red Hat Enterprise Linux 8.2 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。

([BZ#1833541](#))

Control Group v2 は、RHEL 8 で完全に対応されるようになりました。

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 は、非常に多くのコントローラーに対応します。以下に例を示します。

- 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

- `cpuset` コントローラーは、プロセッサやプロセスのメモリー配置を、`cpuset` インターフェイスファイルで指定されたリソースのみに制限します。
- メモリーコントローラーは、メモリー配分を調整します。現在、次の種類のメモリー使用量が追跡されます。
 - ユーザー側のメモリー (ページキャッシュと匿名メモリー)
 - `dentry`、`inode` などのカーネルデータ構造
 - TCP ソケットバッファー
- I/O コントローラーは、I/O リソースの配分を制限します。
- ライトバックコントローラーは、メモリーコントローラーおよび I/O コントローラーの両方と対話し、**Control Group v2** に固有のものです。

上記の情報は、[Control Group v2](#) のアップストリームのドキュメントに基づいています。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。

アップストリームドキュメントに記載の全機能が RHEL 8 に実装されているわけではないので注意してください。

(BZ#1401552)

空きリストのランダム化により、ダイレクトマップ方式のメモリーサイドキャッシュのパフォーマンスおよび使用状況が改善されました。

今回の機能強化により、ページアロケーターを有効にして空きリストをランダム化し、ダイレクトマップ方式のメモリー側キャッシュの平均使用率を高めることができるようになりました。カーネルコマンドラインオプション `page_alloc.shuffle` により、ページアロケーターは空きリストのランダム化が可能になり、ブール値フラグが **True** に設定されます。`sysfs` ファイル

は、`/sys/module/page_alloc/parameters/shuffle` にあります。このファイルは、フラグの状態を読み取り、空きリストをシャッフルします。このため、Dynamic Random Access Memory (DRAM) がキャッシュされ、DRAM と永続メモリー間のレイテンシー幅が低減されます。その結果、大容量で帯域幅が狭い永続メモリーを汎用サーバープラットフォームで利用できます。

(BZ#1620349)

TPM ユーザー空間ツールが最新バージョンに更新されました。

`tpm2-tools` ユーザー空間ツールがバージョン 3.2.1 に更新されました。今回の更新では、特に Platform Configuration Register コードおよび `man` ページのクリーンアップに関連するバグ修正が複数追加されました。

(BZ#1725714)

C620 シリーズの PCH チップセットが Intel Trace Hub 機能に対応するようになりました。

今回の更新で、C620 シリーズプラットフォームコントローラーハブ (PCH) (別称 Lewisburg PCH) の Intel トレースハブ (TH) のハードウェアサポートが追加されました。C620 シリーズ PCH を使用するユーザーは Intel TH を使用できるようになりました。

(BZ#1714486)

`perf` ツールは、CLX-AP および `cpx` プロセッサのダイ別イベント集計をサポートするようになりました。

今回の更新で、`perf` ツールは、ダイが複数ある一部の CPU でダイ別のイベント数集計をサポートする

ようになりました。このモードを有効にするには、Xeon Cascade Lake-AP (CLX-AP) および Cooper Lake (CPX) システムプロセッサの **-a** オプションに、**--per-die** オプションが追加されました。今回の更新では、このサポートが追加され、ダイ間でバランスが取られていない状態が検出されるようになりました。**perf stat** コマンドはイベント数をキャプチャーし、出力を以下のように表示します。

```
# perf stat -e cycles --per-die -a -- sleep 1
Performance counter stats for 'system wide':
S0-D0      8      21,029,877   cycles
S0-D1      8      19,192,372   cycles
```

(BZ#1660368)

IBM Z では `crashkernel=auto` のしきい値が減少します。

IBM Z システムの **crashkernel=auto** カーネルコマンドラインパラメーターのしきい値の下限が 4G から 1G に減ります。今回の実装により、IBM Z は AMD64 および Intel 64 システムのしきい値に合わせて、**crashkernel=auto** のしきい値の下限に対して同じ予約ポリシーが使用できるようになりました。これにより、クラッシュカーネルは、RAM が 4GB 未満のシステムで **kdump** のメモリーを自動的に予約できます。

(BZ#1780432)

numactl マニュアルエントリにより、メモリー使用量の出力が明らかになります。

RHEL 8 の今回のリリースでは、**numactl** の man ページで、メモリー使用量の情報はシステム上の常駐ページしか反映しないということが明示的に説明されています。これは、メモリー使用量の情報が常駐ページに関連するのかわかりにくく、仮想メモリーに関連するのかわかりにくく、ユーザーが混乱する可能性をなくすために追加されました。

(BZ#1730738)

kexec-tools ドキュメントが更新され、Kdump FCoE ターゲットのサポートが含まれるようになりました。

今回のリリースでは、`/usr/share/doc/kexec-tools/supported-kdump-targets.txt` ファイルが更新され、Kdump Fibre Channel over Ethernet (FCoE) ターゲットのサポートが含まれるようになりました。その結果、FCoE ターゲットサポートでの **kdump** クラッシュダンプ機構の状態と詳細について、理解を深めることが可能になりました。

(BZ#1690729)

ファームウェア支援ダンプが PowerNV をサポートするようになりました。

ファームウェア支援ダンプ (**fadump**) メカニズムが PowerNV プラットフォームでサポートされるようになりました。この機能は、IBM POWER9 FW941 ファームウェアバージョン以降でサポートされます。システム障害の発生時には、**fadump** は **vmcore** ファイルとともに、**opalc** もエクスポートします。**opalc** ファイルには、障害時の OpenPower Abstraction Layer (OPAL) メモリーの状態に関する情報が含まれます。**opalc** ファイルは、OPAL ベースシステムのクラッシュのデバッグに役立ちます。

(BZ#1524687)

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

kernel-rt ソースが更新され、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。リアルタイムパッチセットも、最新のアップストリーム v5.2.21-rt13 バージョンに更新されました。これらの更新はいずれも、バグ修正および機能強化を多数提供します。

(BZ#1680161)

rngd が root 以外の権限で実行可能になりました。

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

(BZ#1692435)

POWER 9 の RHEL 8.2 以降で仮想永続メモリーがサポートされるようになりました

IBM POWER9 ハードウェアで PowerVM ハイパーバイザーを使用して RHEL 8.2 以降のホストを実行している場合、ホストは Virtual Persistent Memory (vPMEM) 機能を使用できるようになりました。vPMEM を使用すると、物理サーバーの電源がオフになるまで、アプリケーションおよびパーティションの再起動後もデータが保持されます。その結果、vPMEM を使用するワークロードの再起動が大幅に高速化されます。

システムが vPMEM を使用できるようにするには、次の要件を満たす必要があります。

- ハードウェア管理コンソール (HMC) V9R1 M940 以降
- ファームウェアレベル FW940 以降
- E980 システムファームウェア FW940 以降
- L922 システムファームウェア FW940 以降
- PowerVM レベル V3.1.1

現在、vPMEM を使用する RHEL 8 でいくつかの既知の問題が発生していることに注意してください。詳細は、次のナレッジベース記事を参照してください。

- [Hot plug/unplug of pmem memory can cause kernel panic on POWER9](#)
- [Booting of the capture kernel takes a very long time using vPMEM namespaces as a dump target for kdump/fadump](#)

(BZ#1859262)

5.1.8. ファイルシステムおよびストレージ

LVM が dm-writecache キャッシュメソッドをサポートするようになりました。

LVM キャッシュボリュームでは、既存のホットスポット (**dm-cache**) メソッドに加えて、**dm-writecache** キャッシュメソッドを提供するようになりました。

dm-cache

このメソッドは、高速なボリュームで頻繁に使用されるデータをキャッシュして、このようなデータへのアクセス時間を短縮します。このメソッドは、読み取りおよび書き込みの両方の操作をキャッシュします。

dm-writecache

このメソッドは、書き込み操作のみをキャッシュします。高速ボリュームは通常、SSD または永続メモリー (PMEM) ディスクを指します。これは、まず書き込み操作を格納してから、これらをバックグラウンドで低速ディスクに移行します。

キャッシュメソッドを設定するには、**lvconvert** ユーティリティーで **--type cache** または **--type writecache** オプションを使用します。

詳細は、[Enabling caching to improve logical volume performance](#) を参照してください。

(BZ#1600174)

VDO async ポリシーが **ACID** に準拠するようになりました。

今回のリリースでは、VDO **async** の書き込みモードが Atomicity, Consistency, Isolation, Durability (ACID) に準拠するようになりました。VDO が **async** モードでデータを書き込み中にシステムが予期せず停止した場合に、復元データの一貫性が常に保たれるようになりました。

ACID に準拠するようになり、**async** のパフォーマンスが以前のリリースより低下しています。以前のパフォーマンスに戻すには、VDO ボリュームの書き込みモードを、ACID に準拠していない **async-unsafe** モードに変更してください。

詳細は [VDO 書き込みモードの選択](#) を参照してください。

(BZ#1657301)

VDO ボリュームをインポートできるようになりました。

vdo ユーティリティーでは、システムに登録されていない、既存の VDO ボリュームをインポートできるようになりました。VDO ボリュームをインポートするには、**vdo import** コマンドを使用します。

また、**vdo import** コマンドを使用して、VDO ボリュームの UUID (Universally Unique Identifier) を変更できます。

(BZ#1713749)

mountstats および **nfsiostat** の出力で、新しい **per-op** エラーカウンターが利用できるようになりました。

NFS クライアントシステムでは、マイナーなサポート機能を利用できます。**nfs-utils** の **mountstats** および **nfsiostat** コマンドの出力に、**per-op** のエラー数が追加されました。今回の機能拡張により、これらのツールで、**per-op** のエラー数と割合を表示できるようになり、NFS クライアントマシンの特定の NFS マウントポイントの問題を絞り込むのに役立てることができます。この新しい統計は、Red Hat Enterprise Linux 8.2 カーネルでのカーネルの変更内容に依存することに注意してください。

(BZ#1719983)

XFS で **cgroup** 対応のライトバック IO が利用できるようになりました。

今回のリリースでは、XFS は **cgroup** 対応のライトバック IO をサポートします。通常、**cgroup** のライトバックには、基礎となるファイルシステムからの明示的なサポートが必要です。これまで、XFS のライトバック IO は **root** の **cgroup** の属性のみでした。

(BZ#1274406)

FUSE ファイルシステムが **copy_file_range()** を実装するようになりました。

copy_file_range() システムコールは、ファイルシステムが効率的なデータコピーメカニズムを実装する方法を提供します。GlusterFS は FUSE (Filesystem in Userspace) フレームワークを使用しているので、今回の更新でこのメカニズムを利用するようになりました。FUSE ファイルシステムの読み取り/書き込み機能には複数データのコピーが必要になるため、**copy_file_range()** を使用するとパフォーマンスが大幅に向上します。

(BZ#1650518)

mountstats および **nfsiostat** コマンドに対して **per-op** 統計がサポートされるようになりました。

NFS クライアントシステムに対するサポート機能が使用できるようになりました。**/proc/self/mountstats** ファイルには、**per-op** エラーカウンターが追加されています。今回の更新では、各 **per-op** 統計の行にある 9 番目の番号で、ステータス値がゼロ未満で完了した操作の数を示します。このステータス値はエラーを示します。詳細は、この新しいエラー数を表示する **nfs-utils** の **mountstats** および **nfsiostat** プログラムの更新を参照してください。

(BZ#1636572)

新しい **mount stats** の **lease_time** および **lease_expired** は **/proc/self/mountstats** ファイルにあります。

NFSv4.x クライアントシステムのサポート機能を使用できるようになりました。**/proc/self/mountstats** ファイルには、**nfsv4:** で始まる行の最後に **lease_time** と **lease_expired** のフィールドがあります。**lease_time** フィールドは NFSv4 リースタイムの秒数を指します。**lease_expired** フィールドは、リースの有効期限が切れてからの秒数を指します。リースの期限が切れていない場合は 0 です。

(BZ#1727369)

NVMe デバイスの不適切な削除

今回の機能拡張により、オペレーティングシステムに事前に通知しなくても、Linux オペレーティングシステムから NVMe デバイスを削除できるようになりました。これにより、順番に削除するためにデバイスの準備を行う必要がないため、NVMe デバイスの保守性が向上し、サーバーのダウンタイムをなくしてサーバーの可用性を確保できます。

以下の点に注意してください。

- NVMe デバイスの削除には、**kernel-4.18.0-193.13.2.el8_2.x86_64** バージョン以降が必要です。
- NVMe デバイスを正常に削除するには、ハードウェアプラットフォームまたはプラットフォームで実行されているソフトウェアからの追加要件が必要になる場合があります。
- システム操作に不可欠な NVMe デバイスの削除には対応していません。たとえば、オペレーティングシステムまたはスワップパーティションを含む NVMe デバイスを削除することはできません。

(BZ#1634655)

5.1.9. 高可用性およびクラスター

他のリソースに影響を与えない場合にのみリソースを無効にする新しいコマンドオプション。

場合によっては、他のリソースに影響を与えない場合にのみリソースを無効化する必要があります。これを確認することは、複雑なリソース関係が設定されている場合、手作業では不可能です。このニーズに対応するため、**pcs resource disable** コマンドが、以下のオプションをサポートするようになりました。

- **pcs resource disable --simulate** は、クラスター設定を変更せずに、特定のリソースを無効にします。
- **pcs resource disable --safe** は、あるノードから別のノードへの移行など、その他のリソースが影響を受けない場合にのみ、特定のリソースを無効にします。

- **pcs resource disable --safe --no-strict** は、他のリソースが停止または降格されない場合限り、特定のリソースを無効にします。

さらに、**pcs resource safe-disable** コマンドが、**pcs resource disable --safe** のエイリアスとして導入されました。

(BZ#1631519)

リソース間の関係を表示する新しいコマンド。

新しい **pcs resource relations** コマンドにより、ツリー構造内のクラスターリソース間の関係を表示することができます。

(BZ#1631514)

プライマリーサイトおよび復旧サイトクラスターの両方のステータスを表示する新しいコマンド。

クラスターをリカバリーサイトとして使用するよう設定している場合は、**pcs dr** コマンドを使用して、そのクラスターをリカバリーサイトクラスターとして設定できるようになりました。そうすることで、**pcs dr** コマンドを使用して、プライマリーサイトクラスターとリカバリーサイトクラスターの両方のステータスを単一のノードから表示することができます。

(BZ#1676431)

制約をリスト表示する際に、期限切れのリソース制約はデフォルトで非表示になりました。

リソース制約をリスト表示する場合に、デフォルトで期限切れの制約が表示されなくなりました。期限切れの制約を含めるには、**pcs constraint** コマンドの **--all** オプションを使用します。これにより、期限切れの制約のリストが表示され、制約とそれに関連するルールが (**expired**) として表示されます。

(BZ#1442116)

Pacemaker が、ノードの正常なシャットダウン時にリソースが停止したままになる設定に対応しました。

クラスターノードがシャットダウンした場合、Pacemaker のデフォルトの応答は、そのノードで実行中のすべてのリソースを停止し、別の場所でリソースを復元します。一部のユーザーは、不具合に対してのみ高可用性を確保し、クリーンなシャットダウンを予定された停止として扱うことを好みます。これに対応するため、Pacemaker は **shutdown-lock** および **shutdown-lock-limit** クラスタープロパティをサポートするようになりました。これは、シャットダウン時にノードでアクティブであったリソースは、ノードが次に再参加するまで停止したままにすることを必須と指定するものです。ユーザーは、人的介入なしに、クリーンシャットダウンを予定されていた停止として使用できるようになりました。ノードの正常なシャットダウン時にリソースを停止したままにする設定の情報は、[ノードの正常なシャットダウン時に停止したままになるようにリソースを設定](#) を参照してください。

(BZ#1712584)

単一ノードでのクラスター環境の実行サポート

1つのメンバーのみを持つクラスターは、クラスター環境でリソースを起動し、実行できるようになりました。これにより、ユーザーはバックアップに単一のノードを使用する複数ノードクラスターに別の障害復旧サイトを設定できます。1つのノードで設定されるクラスターは、それ自体にフォールトトレランスがないことに注意してください。

(BZ#1700104)

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

新しいモジュール `python38` が導入されました。

RHEL 8.2 では Python 3.8 が導入されました。これは新しいモジュール `python38` および `ubi8/python-38` コンテナイメージで提供されます。

Python 3.6 と比較しての主な機能強化は、以下のとおりです。

- 新しい Python モジュール (`contextvars`、`dataclasses`、または `importlib.resources` など)
- ウォルラス演算子 (`:=`) と呼ばれる代入式、位置のみのパラメーターなどの新しい言語機能
- `breakpoint()` 組み込み関数、`=` 形式の文字列書式指定、Python および拡張モジュールのデバッグビルドと非デバッグビルド間の互換性による、開発者エクスペリエンスの向上
- パフォーマンスの向上
- オプションの静的タイプヒントのサポートの改善
- 簡単にデバッグできるようにフォーマットされた文字列リテラルへの `=` 指定子の追加 (f-strings)
- パッケージ (`pip`、`requests`、`Cython`) のバージョンの更新

Python 3.8 およびこれのためにビルドされたパッケージは、同じシステムの Python 3.6 と並行してインストールできます。

`python38` モジュールには、システムツール (RPM、DNF、SELinux など) にバインドされる、`python36` モジュール向けに提供されているのと同じバイナリーは含まれないことに留意してください。

`python38` モジュールからパッケージをインストールするには、たとえば、以下を使用します。

```
# yum install python38
# yum install python38-Cython
```

`python38:3.8` モジュールストリームは、自動的に有効になります。

インタプリターを実行するには、たとえば、以下を使用します。

```
$ python3.8
$ python3.8 -m cython --help
```

詳細は、[Python のインストールおよび使用](#) を参照してください。

Red Hat は、RHEL 8 のライフサイクルが終了するまで Python 3.6 のサポートを継続することに留意してください。Python 3.8 のライフサイクルは短くなります。[Red Hat Enterprise Linux 8 Application Streams ライフサイクル](#) を参照してください。

(BZ#1747329)

`mod_wsgi` インストールの変更

以前は、`yum install mod_wsgi` コマンドを使用して `mod_wsgi` モジュールのインストールを試行すると、必ず `python3-mod_wsgi` パッケージがインストールされていました。RHEL 8.2 では、Python 3.6 に加えて、Python 3.8 が導入されました。今回の更新で、インストールする `mod_wsgi` のバージョン指定が必要になりました。指定しない場合、エラーメッセージが返されます。

`mod_wsgi` の Python 3.6 バージョンをインストールするには、次を実行します。

```
# yum install python3-mod_wsgi
```

`mod_wsgi` の Python 3.8 バージョンをインストールするには、次を実行します。

```
# yum install python38-mod_wsgi
```

`python3-mod_wsgi` と `python38-mod_wsgi` パッケージは互いに競合するため、Apache HTTP Server の制限により、システムには一方の `mod_wsgi` モジュールしかインストールできないことに留意してください。

今回の変更が原因で、[BZ#1829692](#) に記載されている既知の依存関係の問題が発生します。

(BZ#1779705)

IBM Z の `zlib` における、ハードウェアアクセラレートされた `deflate` のサポート。

今回の更新で、IBM Z メインフレームの `zlib` ライブラリーに、ハードウェアアクセラレートされた `deflate` アルゴリズムのサポートが追加されました。その結果、IBM Z ベクターマシンでの圧縮および圧縮解除のパフォーマンスが改善されました。

(BZ#1659433)

IBM Power Systems (リトルエンディアン) で `gzip` を圧縮解除した場合のパフォーマンス向上

今回の更新で、32 ビットの Cyclic Redundancy Check (CRC32) の最適化が IBM Power Systems (リトルエンディアン) の `zlib` ライブラリーに追加されました。その結果、`gzip` ファイルの圧縮解除のパフォーマンスが改善されました。

(BZ#1666798)

新しいモジュールストリーム: `maven:3.6`。

RHEL 8.2 では、新しいモジュールストリーム `maven:3.6` が導入されています。このバージョンの Maven 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では、RHEL 8.0 で配布される `maven:3.5` ストリームについて、数多くのバグ修正とさまざまな機能強化が提供されます。

`maven:3.6` ストリームをインストールするには、以下を使用します。

```
# yum module install maven:3.6
```

`maven:3.5` ストリームからアップグレードする場合は、[後続のストリームへの切り替え](#) を参照してください。

(BZ#1783926)

`mod_md` が ACMEv2 プロトコルをサポートするようになりました。

`mod_md` モジュールがバージョン 2.0.8 に更新されました。今回の更新で、Automatic Certificate Management Environment (ACME) 証明書発行および管理プロトコルのバージョン 2 のサポートなど、多くの機能が追加されました。これは、Internet Engineering Task Force (IETF) 規格 (RFC 8555) です。元の ACMEv1 プロトコルは引き続きサポートされますが、一般的なサービスプロバイダーでは非推奨となっています。

(BZ#1747923)

PHP 7.3 の新しい拡張機能

php:7.3 モジュールストリームが更新され、2つの新しい PHP 拡張 (**rrd** および **Xdebug**) を提供するようになりました。

rrd 拡張は、**RRDtool** C ライブラリーへのバインディングを提供します。**RRDtool** は、時系列データのデータロギングおよびグラフ化を行うハイパフォーマンスシステムです。

Xdebug 拡張は、デバッグと開発を支援するために含まれています。拡張機能は開発のみを目的として提供されるということに留意してください。製品環境では使用できません。

RHEL 8 での PHP のインストールおよび使用の詳細は、[Using the PHP scripting language](#) を参照してください。

(BZ#1769857, BZ#1764738)

新しいパッケージ: perl-LDAP および perl-Convert-ASN1。

今回の更新で、RHEL 8 に **perl-LDAP** パッケージと **Perl-Convert-ASN1** パッケージが追加されました。**perl-LDAP** パッケージは、Perl 言語用の LDAP クライアントを提供します。**perl-LDAP** には、Basic Encoding Rules (BER) と Distinguished Encoding Rules (DER) を使用して Abstract Syntax Notation One (ASN.1) データ構造をエンコードおよびデコードする **perl-Convert-ASN1** パッケージが必要です。

(BZ#1663063, BZ#1746898)

sscg がパスワードで保護された秘密鍵ファイルの生成をサポートするようになりました。

sscg ユーティリティーが、パスワードで保護された秘密鍵ファイルを生成できるようになりました。これにより、秘密鍵の保護レベルがさらに追加されます。これは、FreeRADIUS などの一部のサービスで必要です。

(BZ#1717880)

5.1.11. コンパイラーおよび開発ツール

grafana がバージョン 6.3.6 にリベースされました。

grafana パッケージがバージョン 6.3.6 にアップグレードされ、バグ修正および機能強化が複数追加されました。以下は、主な変更点です。

- Database - パフォーマンスを向上させるために、システム統計クエリーを再書き込みします。
- Explore:
 - Safari ブラウザーの分割表示でのクエリーフィールドレイアウトを修正します。
 - サポートされているデータソースの Live オプションを追加し、目的を共有するため **orgId** を URL に追加します。
 - ラベルエンドポイントの新しい **loki**、**start**、および **end** パラメーターのサポートを追加します。
 - Explore での生クエリーモードの切り替えのサポートを追加し、メトリックとログ間の切り替えを可能にします。
 - ログ行のコンテキストを表示します。フィールドまたはラベルで指定された場合は、ログレベルを解析しません。

- 新しい **LogQL** フィルタリング構文をサポートします。
- Grafana/UI から新しい **TimePicker** を使用します。
- **LogRow** Highlighter で改行を処理します。
- ダッシュボードパネルへのブラウザーバックを修正します。
- ロググラフでシリーズレベルでのフィルタリングを修正します。
- 読み込み時およびグラフ/テーブルを畳む時の問題を修正します。
- ログ行の選択/コピーを修正します。
- ダッシュボード - プロパティが足りないパネルリンクでローディングエラーを起こしたダッシュボード **init** を修正し、またコンマ区切り値 (CSV) のデータリンクへのエクスポート時に、ダッシュボードのタイムゾーン設定を修正します。
- エディター - 行全体しかコピーされていなかった問題を修正します。
- LDAP - **multi ldap** および **ldap** 認証コンポーネントを統合します。
- Profile/UserAdmin - 32 ビットビルドで **grafana-server** をクラッシュさせるユーザーエージェントパーサーを修正します。
- Prometheus:
 - **Prometheus** データソースへの切り替え時にパネルエディターがクラッシュするのを防ぎ、**brace-insertion** 動作が気にならないように変更します。
 - **label_replace** でクエリーを修正し、クエリーエディターのロード時に \$1 のマッチを削除します。
 - エディターのマルチラインクエリーを常時許可し、ステップの調整にタイムゾーンを考慮します。
 - ダッシュボード範囲の代わりに、オーバーライドされた **\$__range** のパネル範囲を使用します。
 - 時系列ラベルクエリーに時間範囲フィルターを追加し、挿入された **PromQL** 変数内の | リテラルをエスケープします。
 - Explore にコロンを含むメトリックのラベルの追加中に修正します。
- Auth - API キーの期限切れを許可し、HTTP API でユーザー認証トークンをリッスンしながら、デバイス、OS、およびブラウザーを返し、UI でユーザー認証トークンのリストと取り消しをサポートします。
- DataLinks - スコープ付き変数をデータリンクに正確に適用し、グラフコンテンツメニューでデータポイントのタイプスタンプを表示中にタイムゾーンをフォローし、変数挿入時にデータポイントタイムスタンプを正確に使用し、**\${__series_name}** の誤った挿入を修正します。
- Graph - シリーズ行アイコンをクリックして凡例の問題とウィンドウで表示される水平スクロールバーの問題を修正し、新しいグラデーション塗り潰しオプションを追加します。
- Graphite - 単一値配列変数の glob を防止し、最後に動かしたエイリアスと関数の問題を修正し、**seriesByTag** と変数パラメーターのある関数の問題を修正し、**/metrics/find** リクエストの **POST** を使用します。

- TimeSeries - 値がすべて数値であると想定します。
- Gauge/BarGauge - 失われたしきい値の問題と **avg** 統計で Gauge をロードする問題を修正します。
- PanelLinks - パネルリンク (ドリルダウンリンク) のある Gauge および Bar Gauge パネルのクラッシュの問題を修正し、パネルの説明がない場合のレンダリングの問題を修正します。
- OAuth - SameSite クッキーポリシーに起因する **missing saved state** OAuth ログインの失敗を修正し、DS プロキシの **OAuth** リフレッシュで更新された誤ったユーザートークンを修正します。
- Auth Proxy - キャッシュキーの一部として追加のヘッダーを含めます。
- **cli: dev** モードにおける認識を修正し、sql エラーでの **encrypt-datasource-passwords** の失敗の問題を修正します。
- Permissions - 非管理者ユーザーに対しナビゲーションでプラグインを表示しますが、プラグイン設定は非表示にします。
- TimePicker - ドロップダウンのクイック範囲の高さの最大値を増加させ、ポップオーバーのカスタム範囲のスタイルの問題を修正します。
- Loki - Explore で、リアルタイム表示のログを正しい順番で表示します。
- Timerange - カスタム時間範囲が協定世界時 (UTC) に従っていなかったバグを修正します。
- **remote_cache** - **redis connstr** 解析を修正します。
- Alerting - ルールを警告するタグを追加し、所定のすべてのメールアドレスに対するメール通知の送信を試行します。また、警告ルールのテストを改善し、**Discord** アラート通知者のコンテンツフィールドの設定をサポートします。
- Alertmanager - ラベル名の下線付きの禁則文字を置き換えます。
- AzureMonitor - Azure Logs のクラッシュする組み込み Grafana 変数またはマクロ名を変更します。
- CloudWatch - Amazon Web Services (AWS) Cloudwatch Expressions でリージョンを可視化し、AWS **DocDB** メトリックを追加します。
- GraphPanel - レジェンドテーブルとソート列が非表示の場合に、シリーズをソートしません。
- InfluxDB - Explore におけるログの可視化をサポートします。
- MySQL/Postgres/MSSQL - 日、週、年間隔でのマクロな解析を追加し、クライアント証明書の定期的なリロードのサポートを追加します。
- Plugins - **plugin.json** ファイルで **dataFormats** リストを **skipDataQuery** フラグに置き換えます。
- Refresh picker - 空の間隔を処理します。
- Singlestat - **y** 最小/最大設定をシングルスタットのスパークラインに追加します。
- Templating - ページのリロード後に多値変数で **__text** を正常に表示し、多値変数のフィルタリングされた値すべての選択をサポートします。

- Frontend - Json ツリーコンポーネントが機能していない問題を修正します。
- InfluxDB - ラベルの値フィルターでシングルクォートがエスケープされていない問題を修正します。
- Config - **defaults.ini** ファイルの **remote_cache** の **connectionstring** オプションを修正します。
- Elasticsearch - Explore で、ワイルドカードとして送られなければならない空のクエリー (テンプレート変数経由) を修正し、デフォルトの最大同時シャード要求数を修正し、ログの視覚化をサポートします。
- TablePanel - アノテーション表示を修正します。
- Grafana-CLI - コマンドライン経由で受け取るフラグを修正します。 **RPM/DEB** パッケージ内の **grafana-cli** 用のラッパーと **config/homepath** は、グローバルフラグになりました。
- HTTPServer - **X-XSS-Protection** ヘッダー形式を修正し、新しいヘッダー **X-Content-Type-Options**、**X-XSS-Protection** および **Strict-Transport-Security** を返すオプションを選択肢とします。また、**Strict-Transport-Security** ヘッダーを修正し、Grafana にカスタムの URL パス接頭辞を提供します。

([BZ#1725278](#))

pcp がバージョン 5.0.2 にリベースされました。

pcp パッケージがバージョン **pcp-5.0.2** にアップグレードされ、バグ修正および機能強化が複数追加されました。以下は、主な変更点です。

- **pcp-webapp*** パッケージが、**grafana-pcp** パッケージと **pmproxy** に置き換えられました。
- **pcp-collectl** ツールが **pmrep** 設定に置き換えられました。
- 新規および改善された Performance Metric Domain Agents (PMDA) は、以下のとおりです。
 - **pmdamssql** - Microsoft SQL Server 実装用の新しい PMDA。
 - **pmdanetcheck** - ネットワークチェックを実行する新しい PMDA。
 - **pmdaopenmetrics** - **prometheus** エージェントの名前を **openmetrics** に変更します。
 - **pmdanfsclient** - **per-op** および **per-mount rpc** エラーメトリックを追加します。
 - **pmdalmsensors** - 名前解析とエラー処理を改善します。
 - **pmdaperfevent** - マルチノードシステムで **hv_24x7** ネストイベントをサポートします。
 - **pmdalinux**:
 - スパースな、または不規則な NUMA ノードを適切に処理します。
 - **per-cpu** NUMA 統計情報に CPU の **instname** を使用し、**instid** を使用しません。
 - アクティブなスラブと合計スラブを **slabinfo v2** 解析に追加します。
 - 複数の unix ソケット、**icmp6** メトリック、**hugepage** メトリック値の計算、および、CPU 数が大きい割り込みコードでの **segfault** を修正します。

- **--container** 名前空間でさらにネットワークメトリックを取得します。
- **pmdabcc** - **bcc** 0.10.0 以降のバージョンの **tracepoints** モジュールを修正します。
- **pmdabpftrace** - **bpftrace** スクリプトのメトリックの新しい PMDA。
- **pmdaproc** -
 - **pidlist** 更新でメモリーリークを修正します。
 - **cgroups_scan** での過度な統計呼び出しを回避します。
 - **cgroup** パス、およびアンエスケープインスタンス名のみを保持します。
- **pmdaroot** - キャッシュされた、または非アクティブな **cgroup** 動作の処理を改善し、また、**cgroup** fs 変更のコンテナー **indom** も更新します。
- コレクター (サーバー) ツールの修正は以下のとおりです。
 - **pmproxy** - **/metrics** エンドポイント経由の Openmetrics サポート、**pmseries/grafana** REST API の統合、および新しい **async PMWEBAPI(3)** REST API 実装を追加します。
 - **selinux** - 数多くの **pcp** ポリシーを更新します。
 - **python pmdas** - 認証サポートを有効化し、新しい **set_comm_flags** メソッドによる通信フラグを設定します。
 - **python api- pmdaGetContext()** をエクスポートし、デバッグラッパーを追加します。
 - **perl api** - **python** ラッパーを使用するのと同様に、PMDA ストア用のコンテキスト設定を保証します。
 - **systemd** - すべてのサービスに 120 秒のタイムアウトを追加し、**pmlogger** サービスの起動の失敗を修正します。
- 分析 (クライアント) ツールの修正は次のとおりです。
 - **pmchart** - 取得エラーの状況におけるチャートの自動スケーリングを修正します。
 - **pmrep** - **collectl-dm-sD** および **collectl-sD** の **wait.formula** を修正します。
 - **pmseries** - **delta** キーワードのサポートとタイムスタンプの改善を提供します。
 - **pcp-atop** - 書き込みモード (**-w**) を修正して、**proc** メトリックと **hotproc** メトリックの比較を処理します。
 - **pcp-atopsar** - いくつかのコマンドライン引数の誤処理を修正します。
 - **pcp-dstat** - CSV 出力のヘッダーの位置ずれと、**--bits** コマンドラインオプションの処理を修正します。
 - **libpcp** - ローカルコンテキストのある **cockpit-pcp segv** と、破損アーカイブを処理する複数アーカイブの再生エラーを修正します。

([BZ#1723598](#))

RHEL 8.2 で **grafana-pcp** が利用できるようになりました。

grafana-pcp パッケージは、新しい **grafana** データソース、ならびに **PCP** と **grafana** を接続するアプリケーションプラグインを提供します。**grafana-pcp** パッケージを使用すると、過去の **PCP** メトリックとリアルタイムの **PCP** メトリックを、**pmseries** クエリ言語と対応する **pmwebapi** ライブサービスを使用して分析できます。詳細は、[Performance Co-Pilot Grafana Plugin](#) を参照してください。

(BZ#1685315)

GCC Toolset 9 の更新

GCC Toolset 9 は最新バージョンの開発ツールを提供するコンパイラツールセットです。このツールセットは、**AppStream** リポジトリにおいて、Software Collection の形式で、Application Stream として利用できます。

RHEL 8.2 で導入された主な変更点は、以下のとおりです。

- GCC コンパイラーがバージョン 9.2.1 に更新され、アップストリームの GCC で利用可能なバグ修正および機能拡張が数多く追加されました。
- GCC Toolset 9 コンポーネントが、以下のコンテナイメージ 2 つで利用可能になりました。
 - GCC コンパイラー、GDB デバッガー、**make** 自動化ツールを含む **rhel8/gcc-toolset-9-toolchain**
 - SystemTap や Valgrind などのパフォーマンス監視ツールを含む **rhel8/gcc-toolset-9-perftools**
コンテナイメージをプルするには、root で以下のコマンドを実行します。

```
# podman pull registry.redhat.io/<image_name>
```

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

ツール	バージョン
GCC	9.2.1
GDB	8.3
Valgrind	3.15.0
SystemTap	4.1
Dyninst	10.1.0
binutils	2.32
elfutils	0.176
dwz	0.12
make	4.2.1
strace	5.1

ツール	バージョン
ltrace	0.7.91
annobin	9.08

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

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

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

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

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

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

詳細は、[Using GCC Toolset](#) を参照してください。

(BZ#1789401)

GCC Toolset 9 が NVIDIA PTX ターゲットオフロードに対応します。

GCC Toolset 9 の GCC コンパイラーが、NVIDIA PTX の OpenMP ターゲットオフロードに対応するようになりました。

(BZ#1698607)

RHEL 8.2 で GCC コンパイラーが更新されました。

システム GCC コンパイラーバージョン 8.3.1 が更新され、アップストリームの GCC で利用可能なバグ修正および機能拡張が数多く追加されました。

GNU コンパイラーコレクション (GCC) には、C、C++、および Fortran のプログラミング言語でアプリケーションを開発するためのツールが含まれます。

使用方法は、[RHEL 8 での C および C++ アプリケーションの開発](#) を参照してください。

(BZ#1747157)

glibc の最大 Fastbin サイズを変更するためのチューナブル (調整可能) パラメーターが追加されました。

malloc 関数は、一連の fastbin を使用して、再利用可能なメモリーを指定した最大チャンクサイズ分まで格納します。デフォルトの最大チャンクサイズは、32 ビットシステムの場合は 80 バイト、64 ビットシステムの場合は 160 バイトです。今回の機能拡張により、新しい **glibc.malloc.mxfast** tunable が **glibc** に導入され、最大 fastbin サイズを変更できるようになりました。

(BZ#1764218)

ベクトル化された math ライブラリーは、GCC Toolset 9 の GNU Fortran 向けに有効化されています。

今回の機能強化により、GCC Toolset の GNU Fortran は、ベクトル化された math ライブラリー **libmvec** のルーチンを使用できるようになりました。以前のリリースでは、GCC Toolset の Fortran コンパイラーには、GNU C ライブラリー **glibc** が提供する **libmvec** からのルーチンを使用する前に、Fortran ヘッダーファイルが必要でした。

(BZ#1764238)

glibc.malloc.tcache 調整可能パラメーターが強化されました。

glibc.malloc.tcache_count 調整可能パラメーターは、スレッドごとのキャッシュ (tcache) に格納できる各サイズのメモリーチャンクの最大数を設定できます。今回の更新で、**glibc.malloc.tcache_count** 調整可能パラメーターの上限が 127 から 65535 に増えました。

(BZ#1746933)

glibc 動的ローダーが強化され、メカニズムをプリロードする非継承ライブラリーが提供されるようになりました。

今回の機能強化により、**--preload** オプションにプリロードするライブラリーのコンマ区切りリストを続けることで、ユーザープログラムを呼び出すローダーを呼び出せるようになりました。この機能により、ユーザーは非継承ライブラリーのプリロードリストを使用して、ローダーから直接プログラムを呼び出すことができます。

以前は、環境を介してすべての子プロセスが継承した LD_PRELOAD 環境変数を使用する必要がありました。

(BZ#1747453)

GDB は IBM Z アーキテクチャーで ARCH(13) 拡張をサポートするようになりました。

GNU Debugger (GDB) は、今回の機能強化により、IBM Z アーキテクチャーに ARCH (13) 拡張で実装した新しい命令をサポートするようになりました。

(BZ#1768593)

elfutils がバージョン 0.178 にリベースされました。

elfutils パッケージがバージョン 0.178 にアップグレードされ、バグ修正および機能強化が複数追加されました。以下は、主な変更点です。

- **elfclassify**: ELF オブジェクトの分析ツールが追加されました。
- **debuginfod**: HTTP 経由でファイルおよび RPM アーカイブから ELF、DWARF、およびソースをインデックス化して自動的にフェッチする新しいサーバー、クライアントツール、およびライブラリー。
- **libeb1** が **libdw.so** に直接コンパイルされるようになりました。
- **eu-readelf** には、注記、セクション番号、シンボルテーブルのフラグが新たに複数追加されました。
- **libdw** のマルチスレッドサポートが改善されました。
- **libdw** の他の GNU DWARF 拡張サポートが追加されました。

(BZ#1744992)

SystemTap がバージョン 4.2 にリベースされました。

SystemTap インストラクションツールがバージョン 4.2 に更新されました。主な機能強化は、次のとおりです。

- バックトレースに、ソースのファイル名と行番号を含めることができるようになりました。
- ループ、タイミング、その他のプロセスなど、BPF (Berkeley Packet Filter) バックエンド拡張が多数利用できるようになりました。
- SystemTap スクリプトの管理サービスが新たに利用できるようになりました。このサービスは、メトリックを Prometheus 互換のモニタリングシステムに送信します。
- SystemTap は、**debuginfod** と呼ばれる **elfutils** 用の新しい HTTP ファイルサーバーの機能を継承しました。このサーバーは、デバッグリソースを SystemTap に自動的に送信します。

(BZ#1744989)

IBM Z シリーズパフォーマンスカウンターが強化されました。

IBM Z シリーズタイプ 0x8561、0x8562、および 0x3907 (z14 ZRI) マシンが **libpfm** で認識されるようになりました。IBM Z シリーズで楕円曲線暗号 (ECC) 操作を監視するパフォーマンスイベントが利用できるようになりました。これにより、IBM Z シリーズマシンで追加のサブシステムを監視できるようになりました。

(BZ#1731019)

Rust Toolset がバージョン 1.41 にリベースされました。

Rust Toolset は、バージョン 1.41 に更新されました。以下は、主な変更点です。

- 孤立したルールは厳密性が低くなったので、新規特性の実装が簡素化されました。
- これで、**#[non_exhaustive]** 属性を **struct**、**enum** または **enum** バリエーションにアタッチできるようになりました。
- Foreign Function Interface (FFI) で **Box<T>** を使用すると、より多くの保証が提供されます。**Box<T>** には、FFI の **T*** ポインターと同じ Application Binary Interface (ABI) があります。
- rust はコンパイル時にメモリーの安全性のバグを検出しますが、以前のポロージャーには制限がありました。そのため、未定義の動作が許可され、メモリーの危険性が発生していました。新しい NLL (Non-lexical lifetimes) ポロージャーは、メモリーの安全性の問題を、ハードエラーとして報告する可能性があります。これは、Rust 2015 および Rust 2018 エディションに適用されるようになりました。以前のリリースでは、Rust 2015 は、NLL ポロージャーは、このような問題に関する警告のみを表示していました。

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

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

使用方法は、[Rust Toolset の使用](#) を参照してください。

(BZ#1776847)

LLVM Toolset がバージョン 9.0.1 にリベースされました。

LLVM Toolset がバージョン 9.0.1 にアップグレードされました。今回の更新で、**asm goto** ステートメントに対応するようになりました。この変更により、AMD64 および Intel 64 アーキテクチャーで Linux カーネルをコンパイルできます。

llvm-toolset モジュールをインストールするには、`root` で以下のコマンドを実行します。

```
# yum module install llvm-toolset
```

詳細は、[LLVM Toolset の使用](#) を参照してください。

(BZ#1747139)

Go Toolset がバージョン 1.13 にリベースされました。

Go Toolset がバージョン 1.13 にアップグレードされました。主な機能強化は、次のとおりです。

- RHEL システムが FIPS モードで起動すると、Go が FIPS 認定の暗号化モジュールを使用できるようになりました。**GOLANG_FIPS=1** 環境変数を使用してこのモードを手動で有効にできません。
- Delve デバッガーのバージョン 1.3.2 が Go で利用可能になりました。これは、Go (**golang**) プログラミング言語のソースレベルのデバッガーです。

go-toolset モジュールをインストールするには、`root` で以下のコマンドを実行します。

```
# yum module install go-toolset
```

Delve デバッガーをインストールするには、`root` で以下のコマンドを実行します。

```
# yum install delve
```

Delve を使用して **helloworld.go** プログラムをデバッグするには、以下のコマンドを実行します。

```
$ dlv debug helloworld.go
```

Go Toolset の詳細は、[Using Go Toolset](#) を参照してください。

Delve の詳細は、アップストリームの [Delve documentation](#) を参照してください。

(BZ#1747150)

OpenJDK が secp256k1 もサポートするようになりました。

以前のバージョンでは、Open Java Development Kit (OpenJDK) は NSS ライブラリーの曲線しか使用できませんでした。そのため、OpenJDK は、ECC (elliptic curve cipher) に secp256r1、secp384r1、および secp521r1 の曲線のみを提供していました。今回の更新により、OpenJDK は、内部の ECC 実装を使用し、secp256k1 曲線もサポートします。

(BZ#1746875, BZ#1746879)

5.1.12. ID 管理

IdM が新しい Ansible 管理モジュールに対応するようになりました。

今回の更新で、Ansible Playbook を使用して一般的な Identity Management (IdM) タスクを自動化する複数の **ansible-freeipa** モジュールが導入されました。

- **ipausers** モジュールはユーザーの追加と削除を自動化します。

- **ipagroup** モジュールは、ユーザーグループに対するユーザーとユーザーグループの追加や削除を自動化します。
- **ipahost** モジュールはホストの追加と削除を自動化します。
- **ipahostgroup** モジュールは、ホストグループに対するホストとホストグループの追加や削除を自動化します。
- **ipasudorule** モジュールは、**sudo** コマンドと **sudo** ルールの管理を自動化します。
- **ipapwpolicy** モジュールは、IdM のパスワードポリシーの設定を自動化します。
- **ipahbacrule** モジュールは、IdM のホストベースのアクセス制御管理を自動化します。

2 つ以上の **ipauser** 呼び出しを **users** 変数を含む呼び出し 1 つに統合するか、これらのユーザーを含む JSON ファイルを使用できる点に注意してください。同様に 2 つ以上の **ipahost** 呼び出しを **hosts** 変数を含む 1 つの呼び出しに統合するか、ホストを含む JSON ファイルを使用します。**ipahost** モジュールは、ホストの IPv4 と IPv6 アドレスの有無も確認できます。

(JIRA:RHELPLAN-37713)

IdM Healthcheck は DNS レコードのスクリーニングに対応するようになりました。

今回の更新で、Identity Management (IdM) サーバーで DNS レコードのスタンドアロンの手動テストが導入されました。

このテストでは **Healthcheck** ツールを使用し、**etc/resolv.conf** ファイルのローカルリゾルバーで DNS クエリーを実行します。このテストでは、自動検出に必要な DNS レコードが解決可能であることを確認します。

(JIRA:RHELPLAN-37777)

SSSD を使用した RHEL の AD への直接統合が FIPS に対応するようになりました。

今回の機能強化により、SSSD (System Services Security Daemon) で、認証メカニズムが FIPS (Federal Information Processing Standard) で承認された暗号化タイプを使用する Active Directory (AD) デプロイメントと統合できるようになりました。この機能拡張により、FIPS 基準を満たす必要がある環境で、RHEL システムを AD に直接統合できます。

(BZ#1841170)

SMB1 プロトコルは、Samba サーバーおよびクライアントユーティリティーでデフォルトで無効になっています。

Server Message Block バージョン 1 (SMB1) プロトコルが非推奨になったため、Samba 4.11 では、**server min protocol** および **client min protocol** パラメーターのデフォルト値が **NT1** から **SMB2_02** に変更になりました。**/etc/samba/smb.conf** ファイルにこれらのパラメーターを設定していない場合は、以下のとおりです。

- SMB1 のみをサポートするクライアントは、Samba サーバーに接続できなくなりました。
- **smbclient** などの Samba クライアントユーティリティーと **libsmbclient** ライブラリーは、SMB1 のみをサポートするサーバーへの接続に失敗します。

Red Hat は、SMB1 プロトコルの使用を推奨していません。ただし、環境で SMB1 が必要な場合は、手動でプロトコルを再度有効にすることができます。

Samba サーバーで SMB1 を再度有効にするには、以下を実行します。

- 以下の設定を `/etc/samba/smb.conf` ファイルに追加します。

```
server min protocol = NT1
```

- **smb** サービスを再起動します。

```
# systemctl restart smb
```

Samba クライアントユーティリティーおよび **libsmbclient** ライブラリーの SMB1 を再度有効にするには、以下を実行します。

- 以下の設定を `/etc/samba/smb.conf` ファイルに追加します。

```
client min protocol = NT1
```

- **smb** サービスを再起動します。

```
# systemctl restart smb
```

SMB1 プロトコルは今後の Samba リリースで削除される予定ですので留意してください。

([BZ#1785248](#))

samba がバージョン 4.11.2 にリベースされました。

samba パッケージがアップストリームバージョン 4.11.2 にアップグレードされ、以前のバージョンに対するバグ修正や機能強化が数多く追加されました。以下は、主な変更点です。

- デフォルトで、Server Message Block バージョン 1 (SMB1) プロトコルが、Samba サーバー、クライアントユーティリティー、および **libsmbclient** ライブラリーで無効になりました。ただし、引き続き、**server min protocol** と **client min protocol** パラメーターを手動で **NT1** に設定し、再度 SMB1 を有効にすることは可能です。Red Hat は、SMB1 プロトコルを再度有効にすることは推奨していません。
- **lanman auth** および **encrypt passwords** パラメーターは非推奨となっています。これらのパラメーターは、セキュアでない認証を有効とするもので、非推奨の SMB1 プロトコルでしか利用できません。
- **-o** パラメーターが、**onode** Clustered Trivial Database (CTDB) ユティリティーから削除されました。
- Samba は暗号化に GnuTLS ライブラリーを使用するようになりました。これにより、RHEL の FIPS モードが有効な場合には、Samba は FIPS 標準に準拠します。
- CPU スレッドの 90% を超えて使用される場合に、**ctdbd** サービスがログ記録するようになりました。
- 非推奨の Python 2 サポートが削除されました。

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

主な変更の詳細は、更新前にアップストリームのリリースノート <https://www.samba.org/samba/history/samba-4.11.0.html> を参照してください。

(BZ#1754409)

Directory Server がバージョン 1.4.2.4 にリベースされました。

389-ds-base パッケージがアップストリームバージョン 1.4.2.4 にアップグレードされ、以前のバージョンに対するバグ修正や機能強化が数多く追加されました。主な変更点の一覧については、更新前にアップストリームのリリースノートを参照してください。

[389 Directory Server リリースノートアーカイブ](#) には、次のリリースのリリースノートが含まれています。* 389 Directory Server 1.4.2.4 * 389 Directory Server 1.4.2.3 * 389 Directory Server 1.4.2.2 * 389 Directory Server 1.4.2.1

(BZ#1748994)

一部のレガシースクリプトが Directory Server で置き換えられました。

今回の機能強化では、Directory Server ではサポートされていないレガシースクリプトの **dbverify**、**validate-syntax.pl**、**cl-dump.pl**、**fixup-memberuid.pl**、および **repl-monitor.pl** の代替が提供されます。これらのスクリプトは以下のコマンドに置き換えられました。

- **dbverify**: `dsctl instance_name dbverify`
- **validate-syntax.pl**: `dsconf schema validate-syntax`
- **cl-dump.pl**: `dsconf replication dump-changelog`
- **fixup-memberuid.pl**: `dsconf plugin posix-winsync fixup`
- **repl-monitor.pl**: `dsconf replication monitor`

すべてのレガシースクリプトとその代替のリストについては、[Command-line utilities replaced in Red Hat Directory Server 11](#) を参照してください。

(BZ#1739718)

非表示のレプリカとしての IdM の設定を完全にサポートするようになりました。

RHEL 8.2 の Identity Management (IdM) では、非表示のレプリカとしての IdM サーバーの設定を完全にサポートします。非表示のレプリカは、稼働中および利用できるすべてのサービスが含まれる IdM サーバーです。ただし、DNS のサービスに **SRV** レコードが存在せず、LDAP サーバーロールが有効になっていないため、他のクライアントやマスターには通知されません。そのため、クライアントはサービス検出を使用して隠しレプリカを検出することはできません。

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

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

詳細は、[IdM 非表示レプリカのインストール](#) を参照してください。

(BZ#1719767)

Kerberos チケットポリシーが認証インジケータをサポートするようになりました。

認証インジケータは、チケットを取得するために使用される事前認証メカニズムに基づいて Kerberos チケットに割り当てられます。

- **otp** - 2 要素認証 (パスワード + OTP)
- **radius** - RADIUS 認証
- **pkinit** - PKINIT、スマートカード、または証明書での認証
- **hardened** - 強化パスワード (SPAKE または FAST)

Kerberos Distribution Center (KDC) は、認証インジケータに基づくサービスチケット要求で、サービスアクセス制御、チケットの最大有効期間、および更新可能な期間などのポリシーを強制できます。

今回の機能強化により、管理者はユーザーのチケットから特定の認証インジケータを要求することで、サービスチケットの発行をより詳細に制御できるようになりました。

(BZ#1777564)

krb5 パッケージは FIPS に準拠するようになりました。

今回の機能強化により、準拠していない暗号は禁止されるようになりました。これにより、管理者は FIPS が集計する環境で Kerberos を使用できます。

(BZ#1754690)

Directory Server は、システム全体の暗号化ポリシーに基づいて `sslVersionMin` パラメータを設定します。

デフォルトでは、Directory Server は、システム全体の暗号化ポリシーに基づいて **`sslVersionMin`** パラメータの値を設定するようになりました。`/etc/crypto-policies/config` ファイルでポリシープロファイルを以下のように設定します。

- **DEFAULT**、**FUTURE**、または **FIPS**、Directory Server は **`sslVersionMin`** を **TLS1.2** に設定します。
- **LEGACY**、Directory Server は **`sslVersionMin`** を **TLS1.0** に設定します。

ように、**`sslVersionMin`** は、crypto ポリシーで定義されている値よりも高い値に手動で設定できます。

```
# dsconf -D "cn=Directory Manager" __ldap://server.example.com__ security set --tls-protocol-min TLS1.3
```

(BZ#1828727)

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)

5.1.13. デスクトップ

デュアル GPU システムで Wayland が有効になりました。

以前は、GNOME 環境では、ノートパソコンおよびその他の Graphical Processing Unit (GPU) を 2 つ備えたシステムでの X11 セッションがデフォルトとなっていました。今回のリリースで、GNOME は、デュアル GPU システムでの Wayland セッションがデフォルトになりました。これは、シングル GPU システムの場合と同じ動作です。

(BZ#1749960)

5.1.14. グラフィックインフラストラクチャー

新しいグラフィックカードのサポートが追加されました。

以下のグラフィックカードがサポートされるようになりました。

- Intel Comet Lake H および U プロセッサに含まれる Intel HD Graphics 610、620、および 630
- Intel Ice Lake UHD Graphics 910 および Iris Plus Graphics 930、940、950
Intel Ice Lake グラフィックのサポートを有効にするために **alpha_support** カーネルオプションを設定する必要がなくなりました。
- 以下のモデルを含む AMD Navi 10 ファミリー
 - Radeon RX 5600
 - Radeon RX 5600 XT
 - Radeon RX 5700
 - Radeon RX 5700 XT
 - Radeon Pro W5700
- 以下のモデルを含む Nvidia Turing TU116 ファミリー
nouveau グラフィックドライバーでは、まだ Nvidia Turing TU116 ファミリーの 3D アクセラレーションのサポートはありません。
 - GeForce GTX 1650 Super
 - GeForce GTX 1660
 - GeForce GTX 1660 Super
 - GeForce GTX 1660 Ti
 - GeForce GTX 1660 Ti Max-Q

さらに、以下のグラフィックドライバーが更新されました。

- Matrox **mgag2000** ドライバー
- Aspeed **ast** ドライバー
- Intel **i915** ドライバー

(JIRA:RHELPLAN-41384)

5.1.15. Web コンソール

管理者は、RHEL 8 Web コンソールに対する認証にクライアント証明書を使用できるようになりました。

今回の Web コンソールの機能強化により、システム管理者はクライアント証明書を使用して、証明書認証が組み込まれたブラウザを使用して RHEL 8 システムにローカルまたはリモートでアクセスできるようになりました。追加のクライアントソフトウェアは必要ありません。これらの証明書は、一般的にスマートカードまたは Yubikey によって提供されるか、ブラウザにインポートすることができます。

証明書でログインする場合、ユーザーは現在、Web コンソールで管理アクションを実行できません。ただし、パスワードで認証した後は、**sudo** コマンドを使用して、Terminal ページで管理アクションを実行することができます。

(JIRA:RHELPLAN-2507)

TLS クライアント証明書を使用して Web コンソールにログインするオプション。

今回の更新で、ブラウザまたはスマートカードや YubiKey などのデバイスで利用できる TLS クライアント証明書でログインするように Web コンソールを設定できるようになりました。

(BZ#1678465)

Web コンソールのログインの変更

RHEL Web コンソールが更新され、以下が変更されました。

- Web コンソールは、アクティブではなくなってから 15 分後に、現行セッションから自動的にユーザーをログアウトさせます。`/etc/cockpit/cockpit.conf` ファイルで、タイムアウトを分単位で設定できます。
- SSH と同様に、Web コンソールが、オプションでログイン画面にバナーファイルのコンテンツを表示できるようになりました。`/etc/cockpit/cockpit.conf` ファイルで機能を設定する必要があります。

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

(BZ#1754163)

RHEL Web コンソールが、PatternFly 4 ユーザーインターフェイス設計システムを使用するように再設計されました。

新たな設計は、アクセス性を向上させるものであり、また、OpenShift 4 の設計に一致します。更新には以下が含まれます。

- 概要ページが完全に再設計されました。たとえば、情報はよりわかりやすいパネルにグループ化され、正常性情報はより目立つようになりました。また、リソースグラフは独自のページに移動し、ハードウェア情報ページは簡単に検索できるようになりました。
- ナビゲーションメニューの新しい [Search] フィールドを使用して、キーワードに基づく特定のページを簡単に見つけることができます。

PatternFly の詳細については、[PatternFly のプロジェクト](#) のページを参照してください。

(BZ#1784455)

Virtual Machines ページの更新

Web コンソールの **Virtual Machines** ページでは、ストレージについて複数の改善がありました。

- ストレージボリュームの作成が、libvirt 対応のすべてのタイプで機能するようになりました。
- ストレージプールは LVM または iSCSI に作成できます。

さらに、**Virtual Machines** ページが、仮想ネットワークインターフェイスの作成および削除をサポートするようになりました。

([BZ#1676506](#), [BZ#1672753](#))

Web コンソールの Storage ページの更新

ユーザービリティテストで、RHEL Web コンソールの **Storage** ページにおけるデフォルトマウントポイントの概念を把握することが難しく、多くの混乱を招いていたことがわかりました。今回の更新で、ファイルシステムのマウント時に、Web コンソールが **デフォルト** を提示しなくなりました。新しいファイルシステムを作成する場合、常にマウントポイントの指定が必要になりました。

さらに、Web コンソールでは、設定 (**/etc/fstab**) とランタイム状態 (**/proc/mounts**) との違いが非表示になりました。Web コンソールで追加された変更は、常に設定およびランタイム状態の両方に適用されます。設定とランタイム状態が相互に異なる場合、Web コンソールは警告を表示し、ユーザーが簡単に同期し直せるようにします。

([BZ#1784456](#))

5.1.16. 仮想化

インストールツリーから RHEL 仮想マシンを作成しようとすると、より有用なエラーメッセージが返されるようになりました。

virt-install ユーティリティを使用して作成した RHEL 7 および RHEL 8 仮想マシンで、**--location** オプションを指定すると起動できない場合があります。今回の更新で、この問題の回避策を示す **virt-install** エラーメッセージが追加されました。

([BZ#1677019](#))

Intel Xeon Platinum 9200 シリーズプロセッサは KVM ゲストでサポートされるようになりました。

Intel Xeon Platinum 9200 シリーズプロセッサ (以前は **Cascade Lake** と呼ばれる) のサポートが KVM ハイパーバイザーおよびカーネルコード、および libvirt API に追加されました。これにより、KVM 仮想マシンが Intel Xeon Platinum 9200 シリーズプロセッサを使用できるようになります。

([JIRA:RHELPLAN-13995](#))

EDK2 がバージョン **stable201908** にリベースされました。

EDK2 パッケージがバージョン **stable201908** にアップグレードされ、複数の機能強化が提供されました。主なものは以下のとおりです。

- EDK2 に OpenSSL-1.1.1 のサポートが含まれるようになりました。
- アップストリームプロジェクトのライセンス要件に従い、EDK2 パッケージのライセンスが **BSD and OpenSSL and MIT** から **BSD-2-Clause-Patent and OpenSSL and MIT** に変更になりました。

[\(BZ#1748180\)](#)

入れ子仮想マシンの作成

今回の更新で、RHEL 8 により Intel 64 ホスト上で稼働する KVM 仮想マシン (VM) で、入れ子仮想化が完全にサポートされるようになりました。この機能を使用すると、物理 RHEL 8 ホストで実行中の RHEL 7 または RHEL 8 仮想マシンがハイパーバイザーとして機能し、独自の仮想マシンをホストできます。

AMD64 システムでは、入れ子 KVM 仮想化は引き続きテクノロジープレビューであることに留意してください。

[\(JIRA:RHELPLAN-14047, JIRA:RHELPLAN-24437\)](#)

5.1.17. コンテナ

`/etc/containers/registries.conf` のデフォルトのレジストリー検索リストが更新されました。

`/etc/containers/registries.conf` のデフォルトの `registries.search` リストが更新され、Red Hat およびそのパートナーがキュレート、パッチ修正、保守を行うコンテナイメージを提供する信頼できるレジストリーだけが追加されています。

Red Hat は、以下を含む完全修飾イメージ名を常に使用することを推奨します。

- レジストリーサーバー (完全 DNS 名)
- 名前空間
- イメージ名
- タグ (例: `registry.redhat.io/ubi8/ubi:latest`)

略称名を使用する場合には、レジストリーから `foobar` という名前のイメージをプルして、`myregistry.com` から取得していると想定するなど、なりすましの固有リスクが常に存在します。`myregistry.com` が検索リストの最初でない場合には、攻撃者は検索リストの前のほうに別の `foobar` イメージを配置することができてしまいます。目的のコンテンツではなく、攻撃者が配置したイメージやコードを間違えて実行する可能性があります。Red Hat は、不明なユーザーや匿名ユーザーが任意の名前でアカウントを作成できないように、信頼できるレジストリーだけを追加することを推奨します。これにより、イメージがなりすましやスクワッピングに使用されたり、または別の方法でセキュリティーを低下されないようにします。

[\(BZ#1810053\)](#)

Podman は `oci-systemd-hook` に依存しなくなりました。

Podman は、`container-tools:rhel8` および `container-tools:2.0` モジュールストリームから削除済みの `oci-systemd-hook` パッケージを必要としない、または依存しなくなりました。

[\(BZ#1645280\)](#)

5.2. 外部カーネルパラメーターの重要な変更

本章では、システム管理者向けに、Red Hat Enterprise Linux 8.2 に配布されるカーネルの重要な変更点の概要を説明します。これらの変更には、追加もしくは更新された `proc` エントリー、`sysctl` および `sysfs` のデフォルト値、ブートパラメーター、カーネル設定オプション、または重要な動作の変更などが含まれます。

5.2.1. 新しいカーネルパラメーター

cpuidle.governor = [CPU_IDLE]

使用する **cpuidle** ガバナーの名前

deferred_probe_timeout = [KNL]

これは、プローブへの依存関係で遅延しているプローブが待機をやめるまでのタイムアウトを秒単位で設定するデバッグパラメーターです。

オプトインされている特定の依存関係 (サブシステムまたはドライバー) のみは無視されます。タイムアウト 0 の場合は、**initcalls** の最後にタイムアウトします。また、このパラメーターは、再試行後も遅延しているプローブリストにデバイスをダンプします。

kvm.nx_huge_pages = [KVM]

このパラメーターは、**X86_BUG_ITLB_MULTIHIT** バグのソフトウェアによる回避策を制御します。オプションは次のとおりです。

- **force** - 常に回避策をデプロイします。
- **off** - 回避策をデプロイしません。
- **auto** (デフォルト) - **X86_BUG_ITLB_MULTIHIT** の存在をベースに回避策をデプロイします。

ソフトウェアによる回避策がホストに対して有効化されている場合、ゲストはネスト化されたゲストに対して回避策を有効化する必要はありません。

kvm.nx_huge_pages_recovery_ratio = [KVM]

このパラメーターは、Huge Page に定期的にザッピングされる 4KiB ページの数を制御します。0 の値ではリカバリーが無効になります。値が 0 以外の N の場合、カーネルベースの仮想マシン (KVM) は、毎分 4KiB ページの N 分の 1 をザッピングします。デフォルト値は 60 です。

page_alloc.shuffle = [KNL]

ページアロケーターが空きリストをランダム化するかどうかを制御するブール値フラグ。

ランダム化は、カーネルがダイレクトマップ方式のメモリーサイドキャッシュのあるプラットフォームで実行中であることを検知すると、自動的に有効にされます。このパラメーターを使用して、この動作をオーバーライドすることや、無効にすることができます。

フラグの状態は、**/sys/module/page_alloc/parameters/shuffle** ファイルの **sysfs** pseudo ファイルシステムから読み込むことができます。

panic_print =

パニックが発生したときにシステム情報を出力するビットマスク。
ユーザーは、以下のビットを組み合わせで選択できます。

- ビット 0 - すべてのタスク情報の出力
- ビット 1 - システムメモリー情報の出力
- ビット 2 - タイマー情報の出力
- ビット 3 - **CONFIG_LOCKDEP** カーネル設定がオンの場合にロック情報を出力
- ビット 4 - **ftrace** バッファの出力

- ビット5-バッファのすべての **printk** メッセージを出力

rcutree.sysrq_rcu = [KNL]

新しい猶予期間がまだ開始していない原因を特定することを目的に、**sysrq** キーを強制的に使用して Tree RCU の **rcu_node** ツリーをダンプします。

rcutorture.fwd_progress = [KNL]

この観念をサポートするリードコピー更新 (RCU) のタイプについて、RCU の猶予期間の進捗テストを有効にします。

rcutorture.fwd_progress_div = [KNL]

進捗テストの緊密なループを実行する、CPU のストール警告期間の分数を指定します。

rcutorture.fwd_progress_holdoff = [KNL]

連続する進捗テスト間で待機する秒数

rcutorture.fwd_progress_need_resched = [KNL]

cond_resched() 呼び出しを、進捗テストの緊密なループ中に、**need_resched()** の確認に入れます。

tsx = [X86]

このパラメーターは、Transactional Synchronization Extensions (TSX) 制御に対応する Intel プロセッサで TSX 機能を制御します。オプションは次のとおりです。

- **on** - システムで TSX を有効にします。すべての既知のセキュリティー脆弱性に対する軽減策はありますが、TSX は以前の予測関連の CVE を複数助長しました。その結果、有効にしたままにすると、これに関連する未知のセキュリティーリスクが存在する可能性があります。
- **off** - システムで TSX を無効にします。このオプションは、Microarchitectural Data Sampling (MDS) に対する脆弱性がない新しい CPU でのみ実行可能です。つまり、そういった CPU には **MSR_IA32_ARCH_CAPABILITIES.MDS_NO=1** があり、マイクロコード更新を介して新しい **IA32_TSX_CTRL** モデル固有レジスター (MSR) を取得します。この新しい MSR により、TSX 機能の信頼できる非アクティブ化が実現します。
- **auto** - システムで、**X86_BUG_TAA** が存在する場合に TSX を無効にし、それ以外の場合に TSX を有効にします。

このパラメーターを指定しない場合は、**tsx=off** と同じです。

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

tsx_async_abort = [X86,INTEL]

このパラメーターは、TSX Async Abort (TAA) 脆弱性の軽減策を制御します。

Micro-architectural Data Sampling (MDS) と同様に、Transactional Synchronization Extensions (TSX) に対応した特定の CPU は、CPU 内部バッファに対する不正使用に関して脆弱です。この不正使用では、特定の条件下で情報を開示ガジェットに転送することが可能です。

脆弱なプロセッサでは、攻撃者が直接アクセス権を持たないデータにアクセスするために、キャッシュ側のチャネル攻撃で、予測的に転送されるデータを利用できます。

オプションは次のとおりです。

- **full** - TSX が有効な場合に、脆弱性のある CPU で TAA 軽減策を有効にします。

- **full,nosmt** - 脆弱性のある CPU で TAA 軽減策を有効にし、同時マルチスレッディング (SMT) を無効にします。TSX が無効になっていると、CPU がクロススレッド TAA 攻撃に対して脆弱ではないため、SMT は無効になりません。
- **off** - TAA 軽減策を無条件で無効にします。
MDS の影響を受けるマシンでは、MDS 軽減策をアクティブにすることで **tsx_async_abort=off** パラメーターが阻害される可能性があります。これは両方の脆弱性が同じメカニズムで軽減されるためです。したがって、この軽減策を無効にするには、**mds=off** パラメーターも指定する必要があります。

このオプションを指定しない場合、**tsx_async_abort=full** と同じになります。MDS の影響を受ける CPU と MDS 軽減策をデプロイする CPU では、TAA 軽減策は必要なく、その他の軽減策も提供されません。

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

5.2.2. 更新されたカーネルパラメーター

intel_iommu = [DMAR]

Intel IOMMU ドライバー Direct Memory Access Remapping (DMAR)。オプションは次のとおりです。

- **sm_on** [デフォルトではオフ] - デフォルトで、ハードウェアがスケラブルモードのトランザクションに対応していると通知する場合に、スケラブルモードが無効になります。このオプションが設定されていると、スケラブルモードは、これをサポートすると主張するハードウェアで使用されます。

isolcpus = [KNL,SMP,ISOL]

このパラメーターは、特定のセットの CPU を阻害から分離します。

- **managed_irq** - 分離された CPU が、管理割り込みによって対象とならないようにするサブパラメーター。これには、分離された CPU を含む割り込みマスクがあります。管理された割り込みのアフィニティはカーネルが処理し、**/proc/irq/*** インターフェイス経由で変更することはできません。
この分離はベストエフォートであり、デバイスキューに自動的に割り当てられる割り込みマスクに分離されたハウスキーピング CPU が含まれる場合にのみ有効です。ハウスキーピング CPU がオンラインの場合、この割り込みはハウスキーピング CPU に転送され、ハウスキーピング CPU で送信される I/O は分離された CPU を妨害できません。

キューのアフィニティマスクが分離された CPU しか含まない場合、このパラメーターは割り込みルーティングの決定には影響を与えません。ただし、割り込みは、それらの分離された CPU で実行しているタスクが I/O を送信する場合にのみ配信されます。ハウスキーピング CPU 上で送信された I/O は、これらのキューに影響を与えません。

mds = [X86,INTEL]

オプションの変更は次のとおりです。

- **off** - TSX Async Abort (TAA) の影響を受けるマシンでは、TAA 軽減策をアクティブにすることで、**mds=off** が阻害される可能性があります。これは両方の脆弱性が同じメカニズムで軽減されるためです。したがって、この軽減策を無効にするには、**tsx_async_abort=off** カーネルパラメーターも指定する必要があります。

このパラメーターを指定しない場合は、**mds=full** と同じです。

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

mem_encrypt = [X86-64]

AMD Secure Memory Encryption (SME) の制御

...

メモリー暗号化を有効にできる場合についての詳細は、アップストリームの [カーネルドキュメント](#) を参照してください。

mitigations =

オプションの変更は次のとおりです。

- **off** - オプションの CPU 軽減策をすべて無効にします。これによりシステムパフォーマンスが向上しますが、ユーザーを複数の CPU の脆弱性にさらす可能性もあります。以下に相当します。
 - **nopti [X86,PPC]**
 - **kpti=0 [ARM64]**
 - **nospectre_v1 [X86,PPC]**
 - **nobp=0 [S390]**
 - **nospectre_v2 [X86,PPC,S390,ARM64]**
 - **spectre_v2_user=off [X86]**
 - **spec_store_bypass_disable=off [X86,PPC]**
 - **ssbd=force-off [ARM64]**
 - **l1tf=off [X86]**
 - **mds=off [X86]**
 - **tsx_async_abort=off [X86]**
 - **kvm.nx_huge_pages=off [X86]**
以下の例外があります。
 - **kvm.nx_huge_pages=force** の場合、**kvm.nx_huge_pages** に影響を与えません。
- **auto,nosmt** - 必要に応じて同時マルチスレッディング (SMT) を無効にし、すべての CPU 脆弱性を軽減します。このオプションは、SMT が失われるとしても常に完全な軽減を希望するユーザー向けです。以下に相当します。
 - **l1tf=flush,nosmt [X86]**
 - **mds=full,nosmt [X86]**
 - **tsx_async_abort=full,nosmt [X86]**

rcutree.jiffies_till_sched_qs = [KNL]

このパラメーターは、リードコピー更新 (RCU) が `rcu_note_context_switch()` および

cond_resched() 関数から静止状態のヘルプを求め始める前に指定の猶予期間で必要とされる期間を `jiffy` 単位で設定します。指定されていない場合、カーネルは、直近で設定された **rcutree.jiffies_till_first_fqs** および **rcutree.jiffies_till_next_fqs** カーネルパラメーターをベースに値を計算します。

この計算値は、**rcutree.jiffies_to_sched_qs** カーネルパラメーターで確認できません。**rcutree.jiffies_to_sched_qs** を設定しようとする、上書きされます。

tsc =

このパラメーターは、Time Stamp Counter (TSC) のクロックソース安定性チェックを無効にします。

形式は、<文字列> です。

オプションは次のとおりです。

- **reliable** [x86] - TSC クロックソースを信頼できるものとしてマークします。このオプションは、ランタイム時のクロックソース検証と、起動時に実行される安定性チェックを無効にします。また、このオプションでは、古いハードウェアおよび仮想化環境の high-resolution timer モードが有効になります。
- **noirqtime** [x86] - Interrupt Request (IRQ) の計算に TSC を使用しません。Read Time-Stamp Counter (RDTS) が遅れているプラットフォームで **IRQ_TIME_ACCOUNTING** を無効にする時間を実行するために使用されます。この計算はオーバーヘッドに追加される可能性があります。
- **unstable** [x86] - TSC クロックソースを不安定としてマークします。このオプションは、システムの起動時に TSC を無条件で不安定としてマークし、TSC ウォッチドッグの通知が出た後の追加のウォブルを回避します。
- **nowatchdog** [x86] - クロックソースウォッチドッグを無効にします。このオプションは、クロックソースウォッチドッグからの割り込みが許可されない厳密なレイテンシー要件のある状況で使用されます。

5.2.3. 新しい /proc/sys/kernel パラメーター

panic_print

パニックが発生したときにシステム情報を出力するビットマスク。
ユーザーは、以下のビットを組み合わせで選択できます。

- ビット 0 - すべてのタスク情報の出力
 - ビット 1 - システムメモリー情報の出力
 - ビット 2 - タイマー情報の出力
 - ビット 3 - **CONFIG_LOCKDEP** カーネル設定項目がオンの場合にロック情報を出力
 - ビット 4 - **ftrace** バッファの出力
- たとえば、パニック時にタスクとメモリーの情報を出力するには、以下を実行します。

```
# echo 3 > /proc/sys/kernel/panic_print
```

5.2.4. 更新された /proc/sys/kernel パラメーター

threads-max

このパラメーターは、**fork()** 関数が作成できるスレッドの最大数を制御します。初期化時に、カーネルは、最大数のスレッドが作成された場合でも、スレッド構造が利用可能な RAM ページの一部 (8 分の 1) だけを占有することのないように、この値を設定します。

threads-max に書き込める最小値は 1 です。最大値は、定数 **FUTEX_TID_MASK (0x3fffffff)** で指定されます。

この範囲外の値が **threads-max** に書き込まれた場合、エラー **EINVAL** が発生します。

5.2.5. 更新された /proc/sys/net パラメーター

bpf_jit_enable

このパラメーターは、**Berkeley Packet Filter Just-in-Time (BPF JIT)** コンパイラーを有効にします。

BPF は、さまざまなフックポイントでバイトコードの実行を可能にする、柔軟で効率的なインフラストラクチャーです。これは、ネットワーク (**XDP**、**tc** など)、トレース (**kprobes**、**uprobes**、**tracepoints** など)、およびセキュリティー (**seccomp** など) のような、数多くの Linux カーネルサブシステムで使用されます。

LLVM には、制限付きの C を一連の **BPF** 命令にコンパイルできる **BPF** バックエンドがあります。プログラムが **bpf()** システムコールを介してロードし、カーネルの検証器を渡した後、**JIT** はこれらの **BPF** プログレットをネイティブな CPU 命令に変換します。

JIT には複数のフレーバーがあり、新しい **eBPF JIT** は現在以下の CPU アーキテクチャーでサポートされます。

- **x86_64**
- **arm64**
- **ppc64** (little および big endians 両方)
- **s390x**

5.3. デバイスドライバー

本章では、Red Hat Enterprise Linux 8.2 で新規または更新されたデバイスドライバーをすべてとりあげます。

5.3.1. 新しいドライバー

ネットワークドライバー

- gVNIC ドライバー (gve.ko.xz)
- Broadcom UniMAC MDIO バスコントローラー (mdio-bcm-unimac.ko.xz)
- Software iWARP Driver (siw.ko.xz)

グラフィックドライバーとその他のドライバー

- DRM VRAM メモリーマネジメントヘルパー (drm_vram_helper.ko.xz)

- haltpoll ガバナー用 cpuidle ドライバー (cpuidle-haltpoll.ko.xz)
- stm_ftrace ドライバー (stm_ftrace.ko.xz)
- stm_console ドライバー (stm_console.ko.xz)
- System Trace Module デバイスクラス (stm_core.ko.xz)
- dummy_stm デバイス (dummy_stm.ko.xz)
- stm_heartbeat ドライバー (stm_heartbeat.ko.xz)
- Intel® Trace Hub Global Trace Hub ドライバー (intel_th_gth.ko.xz)
- Intel® Trace Hub PTI/LPP 出力ドライバ (intel_th_pti.ko.xz)
- Intel® Trace Hub コントローラードライバー (intel_th.ko.xz)
- Intel® Trace Hub Memory Storage Unit ドライバー (intel_th_msu.ko.xz)
- Intel® Trace Hub Software Trace Hub ドライバー (intel_th_sth.ko.xz)
- Intel® Trace Hub Memory Storage Unit ソフトウェアシンク (intel_th_msu_sink.ko.xz)
- Intel® Trace Hub PCI コントローラードライバー (intel_th_pci.ko.xz)
- Intel® Trace Hub ACPI コントローラードライバー (intel_th_acpi.ko.xz)
- Intel 10nm サーバードライバ MC Driver (i10nm_edac.ko.xz)
- Device DAX: ダイレクトアクセスマッピングデバイス (dax_pmem_core.ko.xz)
- PMEM DAX: 永続メモリーへのダイレクトアクセス (dax_pmem.ko.xz)
- PMEM DAX: 非推奨の /sys/class/dax インターフェイスのサポート (dax_pmem_compat.ko.xz)
- Intel PMC Core platform init (intel_pmc_core_pltdrv.ko.xz)
- MSR インターフェイス経由 Intel RAPL (Running Average Power Limit) コントロール (intel_rapl_msr.ko.xz)
- Intel Runtime Average Power Limit (RAPL) 共通コード (intel_rapl_common.ko.xz)

ストレージドライバー

- MD のクラスタリングサポート (md-cluster.ko.xz)

5.3.2. 更新されたドライバー

ネットワークドライバーの更新

- VMware vmxnet3 virtual NIC ドライバー (vmxnet3.ko.xz) がバージョン 1.4.17.0-k に更新されました。
- Intel® 10 Gigabit Virtual Function Network Driver (ixgbevf.ko.xz) がバージョン 4.1.0-k-rh8.2.0 に更新されました。

- Intel® 10 Gigabit PCI Express Network Driver (ixgbe.ko.xz) がバージョン 5.1.0-k-rh8.2.0 に更新されました。
- Intel® Ethernet Connection E800 Series Linux Driver (ice.ko.xz) がバージョン 0.8.1-k に更新されました。
- Netronome Flow Processor (NFP) ドライバー (nfp.ko.xz) がバージョン 4.18.0-185.el8.x86_64 に更新されました。
- Elastic Network Adapter (ENA) (ena.ko.xz) がバージョン 2.1.0K に更新されました。

グラフィックおよびその他ドライバーの更新

- HPE ウォッチドッグドライバー (hpwdt.ko.xz) がバージョン 2.0.3 に更新されました。
- Intel I/OAT DMA Linux ドライバー (ioatdma.ko.xz) がバージョン 5.00 に更新されました。

ストレージドライバーの更新

- Driver for HPE Smart Array Controller (hpsa.ko.xz) がバージョン 3.4.20-170-RH4 に更新されました。
- LSI MPT Fusion SAS 3.0 Device Driver (mpt3sas.ko.xz) がバージョン 32.100.00.00 に更新されました。
- QLogic FCoE Driver (bnx2fc.ko.xz) がバージョン 2.12.10 に更新されました。
- Emulex LightPulse Fibre Channel SCSI ドライバー (lpfc.ko.xz) がバージョン 0:12.6.0.2 に更新されました。
- QLogic FastLinQ 4xxxx FCoE Module (qedf.ko.xz) がバージョン 8.42.3.0 に更新されました。
- QLogic Fibre Channel HBA Driver (qla2xxx.ko.xz) がバージョン 10.01.00.21.08.2-k に更新されました。
- Driver for Microsemi Smart Family Controller バージョン (smartpqi.ko.xz) がバージョン 1.2.10-025 に更新されました。
- QLogic FastLinQ 4xxxx iSCSI Module (qedi.ko.xz) がバージョン 8.37.0.20 に更新されました。
- Broadcom MegaRAID SAS Driver (megaraid_sas.ko.xz) がバージョン 07.710.50.00-rc1 に更新されました。

5.4. バグ修正

本パートでは、ユーザーに大きな影響を及ぼしていた Red Hat Enterprise Linux 8.2 のバグで修正されたものを説明します。

5.4.1. インストーラーおよびイメージの作成

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

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

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

(BZ#1637472)

インストーラーにおける s390x のセキュアブートのサポート

以前は、RHEL 8.1 は、セキュアブートの使用を強制する IBM Z 環境で使用するためにブートディスクの準備をサポートしていました。インストール時に使用されるサーバーおよびハイパーバイザーの機能により、そのオンディスクフォーマットにセキュアブートのサポートが含まれるかを判断していました。インストール時にオンディスク形式に影響を与える方法はありませんでした。したがって、セキュアブートに対応している環境で RHEL 8.1 をインストールした場合、セキュアブートに対応していない環境に移行すると、一部のフェイルオーバーシナリオで起動したようにシステムを起動することはできませんでした。

今回の更新で、**zipl** ツールのセキュアブートオプションを設定できるようになりました。これを行うには、以下のいずれかを使用します。

- キックスタート **zipl** コマンドと、そのオプション (**--secure-boot**、**--no-secure-boot**、**--force-secure-boot** など) の 1 つ。
- GUI の **Installation Summary** ウィンドウから、**System > Installation Destination > Full disk summary and boot loader** リンクを選択してシステムの起動デバイスを設定することができます。その結果、セキュアブートに対応していない環境でも、インストールを起動できるようになりました。

(BZ#1659400)

セキュアブート機能が利用できるようになりました。

以前は、**secure=** ブートオプションのデフォルト値が **auto** に設定されていなかったため、セキュアブート機能が利用できませんでした。今回の更新で、事前設定されていない限り、デフォルト値が **auto** に設定され、セキュアブート機能が利用できるようになりました。

(BZ#1750326)

/etc/sysconfig/kernel ファイルが **new-kernel-pkg** スクリプトを参照しなくなりました。

以前は、**/etc/sysconfig/kernel** ファイルが **new-kernel-pkg** スクリプトを参照していました。ただし、**new-kernel-pkg** スクリプトは RHEL 8 システムに含まれていません。今回の更新で、**new-kernel-pkg** スクリプトの参照が **/etc/sysconfig/kernel** ファイルから削除されました。

(BZ#1747382)

インストールで、許容される最大値を超える数のデバイスを **boot-device NVRAM** 変数に設定しません。

以前は、RHEL 8 のインストールプログラムが、許容される最大値を超える数のデバイスを **boot-device NVRAM** 変数に設定していました。これにより、デバイスが最大値を超えるシステムでインストールが失敗していました。今回の更新で、RHEL 8 インストールプログラムが、最大デバイス設定を確認し、許可された数のデバイスしか追加しないようになりました。

(BZ#1748756)

インストールが、ネットワーク上にないキックスタートファイルの URL コマンドで指定されたイメージの場所に対して機能します。

以前は、イメージのリモートの場所によってトリガーされるネットワークアクティベーションが、ネッ

トワーク上にないキックスタートファイルの URL コマンドで指定されていた場合、インストールがプロセスの初期段階で失敗していました。今回の更新で問題が修正され、CD-ROM やローカルブロックデバイスなど、ネットワーク上にないキックスタートファイルの URL コマンドを使用してイメージの場所を提供するインストールが期待どおりに機能するようになりました。

(BZ#1649359)

RHEL 8 インストールプログラムは、未フォーマットのデバイスの ECKD DASD のみを確認します。

以前は、未フォーマットのデバイスを確認する際に、インストールプログラムがすべての DASD デバイスを確認していました。しかし、インストールプログラムが確認する必要があるのは、ECKD DASD デバイスのみでした。これにより、SWAPGEN のある FBA DASD デバイスが使用されると、インストールはトレースバックで失敗していました。今回の更新で、インストールプログラムが FBA DASD デバイスを確認しなくなり、インストールが正常に実行されるようになりました。

(BZ#1715303)

5.4.2. ソフトウェア管理

yum repolist が、利用できない最初のリポジトリで終了しないようになりました。

以前は、リポジトリ設定オプション **skip_if_unavailable** はデフォルトで以下のように設定されていました。

```
skip_if_unavailable=false
```

この設定により、エラーと終了ステータス 1 が付いた利用できない最初のリポジトリで **yum repolist** コマンドが強制終了していました。したがって、**yum repolist** は、利用可能なリポジトリのリスト表示を続行しませんでした。

今回の更新で、**yum repolist** が修正され、一切のダウンロードを必要としなくなりました。これにより、**yum repolist** はメタデータを必要とする出力を提供しなくなり、コマンドは、期待どおりに利用可能なリポジトリをリスト表示し続けます。

利用可能なパッケージの数は、利用可能なメタデータを引き続き必要とする **yum repolist --verbose** または **yum repoinfo** によってしか返されないことに留意してください。したがって、これらのコマンドは利用できない最初のリポジトリで終了します。

(BZ#1697472)

5.4.3. シェルおよびコマンドラインツール

ReaR が更新されました。

RHEL 8.2 では、**ReaR** (Relax-and-Recover) ユーティリティーに多くの更新が導入されました。

build ディレクトリーの処理が変更されました。以前のリリースでは、**ReaR** に障害が発生した場合に、build ディレクトリーが一時的な場所に保持されていました。今回の更新で、非対話的な実行で build ディレクトリーがデフォルトで削除され、ディスク領域が消費されなくなりました。

KEEP_BUILD_DIR 設定変数のセマンティクスが拡張され、新しい **errors** 値が追加されるようになりました。**KEEP_BUILD_DIR** 変数を以下の値に設定できます。

- **errors** - エラーのある build ディレクトリーを保存を保存してデバッグする (以前の動作)

- **y (true)** - 常に build ディレクトリーを保持する
- **n (false)** - build ディレクトリーを保持しない

デフォルト値は空の文字列です。これは、**ReaR** を対話的に (端末内で) 実行した場合に **errors** を、**ReaR** を非対話的に実行する場合は **false** として処理されます。**KEEP_BUILD_DIR** は、デバッグモード (-d) とデバッグスクリプトモード (-D) では自動的に **true** に設定され、この動作は変更されていません。

以下は、主なバグ修正です。

- NetBackup 8.0 のサポートが修正されました。
- **ReaR** は、ユーザー、グループ、グループごとのユーザー数が多いシステムで、**xrealloc: cannot allocate** と同様なバッシュエラーで中断されなくなりました。
- **bconsole** コマンドで、プロンプトが表示され、Bacula 統合の使用時に復元操作を実行できるようになりました。
- **ReaR** は、**docker** サービスが実行中ではあるものの、**docker** の root ディレクトリーが定義されていない状況や、**docker** サービスのステータスが判断できない場合に、正しくファイルをバックアップするようになりました。
- 移行モードでシンプールを使用したり、システムを復旧したりする場合にリカバリーに失敗しなくなりました。
- LVM を使用したリカバリープロセス中に **initramfs** の再ビルドに時間がかかる問題が解決されました。
- **ReaR** は、UEFI ブートローダーの使用時に、AMD および Intel 64 ビットアーキテクチャーで、正しく機能するブータブル ISO イメージを作成するようになりました。このセットアップでレスキューイメージを起動すると、エラーメッセージ **Unknown command 'configfile' (...)** **Entering rescue mode...** で Grub で中止されなくなりました。この設定での GRUB_RESCUE のサポート (XFS ファイルシステムのサポートがないため、以前は失敗する可能性がありました) も修正されました。

(BZ#1729501)

mlocate パッケージのインストール時に **mlocate-updatedb.timer** が有効化されるようになりました。

以前のバージョンでは、**mlocate** パッケージのインストール後に **mlocate-updatedb.timer** が無効になるため、ファイルデータベースの再インデックスは自動的に実行されませんでした。今回の更新で、**mlocate-updatedb.timer** タイマーが **90-default.preset** ファイルに含まれるようになり、**mlocate** パッケージのインストール後にデフォルトで有効になります。その結果、ファイルデータベースが自動的に更新されます。

(BZ#1817591)

5.4.4. インフラストラクチャーサービス

dnsmasq が非再帰 DNS クエリーを正しく処理するようになりました。

以前は、**dnsmasq** は、非再帰クエリーをすべてアップストリームサーバーに転送し、その結果、さまざまな応答が発生していました。今回の更新で、DHCP ホストリース名や **/etc/hosts** ファイルから読み込まれたホストなど、ローカルで既知の名前に対する非再帰的クエリーが **dnsmasq** で処理されるよう

になり、アップストリームサーバーに転送されなくなりました。その結果、既知の名前への再帰クエリーと同じ応答が返されます。

(BZ#1700916)

dhclient が、システム時間の変更後に IP アドレスの更新に失敗しなくなりました。

以前は、システム時間を変更すると、カーネルによる削除の影響で、割り当てられた IP アドレスがシステムで失われる可能性がありました。今回の更新で、**dhclient** が巻き戻しジャンプ検出にモニタリングタイマーを使用し、システム時刻で連続しないジャンプがあった場合のリース拡張に **DHCPREQUEST** メッセージを発行するようになりました。その結果、上述のシナリオでシステムが IP アドレスを失わなくなりました。

(BZ#1729211)

ipcalc が、/31 ネットワークの正しいブロードキャストアドレスを返すようになりました。

今回の更新で、**ipcalc** ユーティリティーが正しく RFC 3021 規格に従うように修正されています。これにより、インターフェイスで /31 接頭辞を使用すると、**ipcalc** は正しいブロードキャストアドレスを返します。

(BZ#1638834)

/etc/services に適切な NRPE ポート定義が含まれるようになりました。

今回の更新で、適切な Nagios Remote Plug-in Executor (NRPE) サービスポート定義が **/etc/services** ファイルに追加されました。

(BZ#1730396)

postfix DNS リゾルバーコードは、**res_query** ではなく、**res_search** を使用するようになりました。

DNS リゾルバーコードは、**postfix** の前回の更新後、**res_search** 関数ではなく、**res_query** 関数を使用していました。これにより、以下の **postfix** 設定を使用する場合は、DNS リゾルバーで、現在のドメインおよび親ドメインのホスト名が検索されませんでした。

```
# postfixconf -e "smtp_host_lookup = dns"
# postfixconf -e "smtp_dns_resolver_options = res_defnames, res_dnsrch"
```

たとえば、

```
# postfixconf -e "relayhost = [smtp]"
```

と **example.com** 形式のドメイン名を使用している場合には、DNS リゾルバーは **smtp.example.com** の SMTP サーバーをリレーに使用しませんでした。

今回の更新で、DNS リゾルバーコードが **res_query** の代わりに **res_search** を使用するように変更され、現在のドメインおよび親ドメインのホスト名が正しく検索されるようになりました。

(BZ#1723950)

Postfix で **PCRE**、**CDB**、および **SQLite** が使用できるようになりました。

RHEL 8 では、**postfix** パッケージは複数のサブパッケージに分けられており、各サブパッケージが特定のデータベースのプラグインを提供しています。以前は、**postfix-pcre** プラグイン、**postfix-cdb** プラグイン、**postfix-sqlite** プラグインを含む RPM パッケージは提供されていませんでした。したがって、

Postfix では、これらのプラグインを使用するデータベースを使用できませんでした。今回の更新で、PCRE、CDB、および SQLite プラグインを含む RPM パッケージが AppStream リポジトリに追加されました。そのため、適切な RPM パッケージのインストール後、これらのプラグインを使用できません。

(BZ#1745321)

5.4.5. セキュリティー

fapolicyd が RHEL の更新を阻止しなくなりました。

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

[RHBA-2020:5243](#) アドバイザリーのリリースでは、**fapolicyd** がバイナリーパスの接尾辞を無視し、バイナリーが信頼データベースに一致するようにします。これにより、**fapolicyd** がルールを正しく適用し、更新プロセスを完了できるようになりました。

(BZ#1897091)

openssl-pkcs11 が、複数のデバイスへのログイン試行によるデバイスのロックをしなくなりました。

以前は、**openssl-pkcs11** エンジンが、PKCS #11 URI が提供する最初の検索結果へのログインを試みていました。この際、最初の検索結果が目的のデバイスではなく、PIN が別のデバイスに適合するものだったとしても、提供された PIN を使用していました。これらの認証試行の失敗により、デバイスはロックされていました。

openssl-pkcs11 は、提供された PKCS #11 URI が単一のデバイスと一致する場合にのみ、ログインを試みるようになりました。PKCS #11 検索で複数のデバイスが検出される場合には、エンジンが意図的に失敗するようになりました。このため、**openssl-pkcs11** を使用してデバイスにログインする場合には、単一のデバイスにのみ一致する PKCS #11 URI を提供しなければなりません。

(BZ#1705505)

rpmverifyfile を使用する OpenSCAP オフラインスキャンが正常に動作するようになりました。

今回の更新以前は、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** プローブの正しい引数で呼び出していませんでした。OpenSCAP スキャナーが、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するように修正され、また、**fchdir** 関数が **rpmverifyfile** で正しい引数を使用するように修正されました。その結果、OVAL **rpmverifyfile** を含む SCAP コンテンツを、OpenSCAP で任意のファイルシステムのスキャンに使用できるようになりました。

(BZ#1636431)

PKCS #11 デバイスに保存された公開鍵とは一致しない ECDSA 秘密鍵を使用する場合でも、httpd が正常に起動するようになりました。

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

していました。OpenSSL に提供する **EVP_PKEY** 構造を満たす場合、**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ていました。

OpenSSL がエンジンから ECDSA 秘密鍵を要求する場合、指定された **EVP_PKEY** 構造には、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 デバイスに公開鍵がなければ、公開鍵情報が含まれていませんでした。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** が起動しなくなっていました。この問題は、公開鍵オブジェクトが利用できない場合に、証明書から EC 公開鍵を読み込むことで解決されています。これにより、ECDSA 鍵が PKCS#11 デバイスに保存されている場合、**httpd** が正常に起動するようになりました。

(BZ#1664807)

Audit ルールの **scap-security-guide PCI-DSS** 修正が適切に動作するようになりました。

以前は、**scap-security-guide** パッケージには、修正の組み合わせと、以下のいずれかのシナリオで生じるチェックが含まれていました。

- 監査ルールの誤った修正
- 渡されたルールが失敗とマークされた誤検出を含むスキャン評価

これにより、RHEL インストールプロセス時に、インストール済みシステムのスキャンが、失敗またはエラーとして Audit ルールを報告していました。

今回の更新で、修復が修正され、PCI-DSS セキュリティポリシーでインストールしたシステムのスキャンで、Audit ルールの誤検出が報告されなくなりました。

(BZ#1754919)

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するようになりました。

以前は、**OpenSCAP** のコードベースをリファクタリングすると、特定の RPM プロー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、次のツール (**oscap-vm** および **oscap-chroot**) を **openscap-utils** パッケージに含めることができませんでした。さらに、**openscap-containers** パッケージは、RHEL 8 から完全に削除されました。今回の更新で、プローブの問題が修正されました。

これにより、RHEL 8 には、**openscap-utils** パッケージに **oscap-podman** ツール、**oscap-vm** ツール、および **oscap-chroot** ツールが含まれるようになりました。

(BZ#1618489)

OpenSCAP rpmverifypackage が正常に動作するようになりました。

以前は、**rpmverifypackage** プローブにより、システムコール **chdir** および **chroot** が 2 回呼び出されていました。これにより、カスタムの OVAL (Open Vulnerability and Assessment Language) コンテンツを使用した **OpenSCAP** をスキャンする際にこのプローブを使用していると、エラーが発生していました。**rpmverifypackage** プローブが、システムコール **chdir** および **chroot** を正常に使用するよう修正されました。その結果、**rpmverifypackage** が正常に動作するようになりました。

(BZ#1646197)

5.4.6. ネットワーク

qdisc_run 関数におけるロックによって、カーネルのクラッシュが発生しなくなりました。

以前は、**pfifo_fast** キュー規律がトラフィックのデキュー中にリセットされる場合の競合状態が、開放後のパケット転送につながっていました。これにより、カーネルが突然終了することがありました。今回の更新で、**qdisc_run** 関数のロックが改善されました。これにより、上述のシナリオでカーネルがクラッシュしなくなりました。

(BZ#1744397)

org.fedoraproject.FirewallD1.config.service の DBus API が期待どおりに動作するようになりました。

以前は、**org.fedoraproject.FirewallD1** 内の DBus API **getIncludes**、**setIncludes**、および **queryIncludes** 関数が、間違ったインデックスのために **org.fedoraproject.FirewallD1.Exception: list index out of range** というエラーメッセージを返していました。今回の更新で、DBus API 関数 **getIncludes**、**setIncludes**、および **queryIncludes** が期待どおりに動作するようになりました。

(BZ#1737045)

RHEL で、**ipvs** モジュールのアンロード時にカーネル警告をログに記録しなくなりました。

以前は、IP 仮想サーバー (**ipvs**) モジュールが誤った参照カウントを使用していたため、モジュールのアンロード時に競合状態が発生していました。これにより、RHEL がカーネル警告をログ記録していました。今回の更新で競合状態が修正されています。その結果、**ipvs** モジュールのアンロード時に、カーネルが警告をログ記録しなくなりました。

(BZ#1687094)

nft ユーティリティーが、最初のオプション以外の引数の後に、引数をコマンドラインオプションとして解釈することがなくなりました。

以前は、**nft** ユーティリティーは **nft** コマンド内のどの場所でもオプションを受け入れていました。たとえば、管理者はオプション以外の引数の間や、またはその後に、オプションを使用することができました。これにより、ダッシュ記号のために、**nft** は負の優先順位の値をオプションとして解釈し、コマンドに失敗していました。**nft** ユーティリティーのコマンドラインパーサーが、最初のオプション以外の引数を一度読み込んだ後は、ダッシュ記号で始まる引数を解釈しないよう更新されました。その結果、管理者が、負の優先順位の値を **nft** に渡すための回避策をとる必要がなくなりました。

この変更により、すべてのコマンドオプションは、最初のオプション以外の引数より前に **nft** に渡さなければならなくなりましたので、留意してください。この更新をインストールした後でも確実に **nftables** スクリプトが期待どおりに動作するよう、更新前に、スクリプトがこの新しい条件を満たしていることを確認してください。

(BZ#1778883)

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

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

(BZ#1663556)

ゾーンのドリフトを無効にする設定パラメーターを **firewalld** に追加されました

以前のリリースでは、**firewalld** サービスに、ゾーンのドリフトとして知られる、文書化されていない

動作が含まれていました。RHEL 8.0 では、セキュリティーに悪影響を及ぼす可能性があるという理由から、この動作が削除されました。その結果、この動作を使用して汎用ゾーンまたはフォールバックゾーンを設定するホストでは、以前許可されていた接続を **firewalld** が拒否するようになりました。今回の更新で、ゾーンのドリフト動作が設定可能な機能として再度追加されました。これにより、ゾーンのドリフトを使用するか、動作を無効化してよりセキュアなファイアウォール設定を使用するかを、ユーザーが決定できるようになりました。

RHEL 8.2 では、`/etc/firewalld/firewalld.conf` ファイルの新しい **AllowZoneDrifting** パラメーターはデフォルトで **yes** に設定されています。このパラメーターが有効な場合、以下のメッセージが **firewalld** ログに記録されるので注意してください。

```
WARNING: AllowZoneDrifting is enabled. This is considered an insecure configuration option. It will be removed in a future release. Please consider disabling it now.
```

(BZ#1772208)

5.4.7. カーネル

サブセクションのメモリーホットプラグが完全にサポートされるようになりました。

以前のリリースでは、デュアルインラインモジュール (DIMM) やインターリーブなど、調整された物理メモリー領域が 64MiB メモリー境界に設定されているものがありました。ただし、Linux ホットプラグサブシステムは 128MiB のメモリーサイズを使用するため、新しいデバイスをホットプラグすると、複数のメモリー領域が1つのホットプラグメモリーウィンドウに重なっていました。そのため、以下または同様の呼び出しトレースの、利用可能な永続メモリー名前空間をリスト表示できませんでした。

```
WARNING: CPU: 38 PID: 928 at arch/x86/mm/init_64.c:850
add_pages+0x5c/0x60
[.]
RIP: 0010:add_pages+0x5c/0x60
[.]
Call Trace:
 devm_memremap_pages+0x460/0x6e0
 pmem_attach_disk+0x29e/0x680 [nd_pmem]
 ? nd_dax_probe+0xfc/0x120 [libnvdimm]
 nvdimm_bus_probe+0x66/0x160 [libnvdimm]
```

今回の更新で問題が修正され、複数のメモリー領域が単一のホットプラグメモリーウィンドウを共有できるように Linux ホットプラグサブシステムをサポートするようになりました。

(BZ#1724969)

データの破損の場合、WARN メッセージではなく BUG がトリガーされるようになりました。

今回の機能強化により、`lib/list_debug.c` でリストが破損すると、BUG がトリガーされ、**vmcore** でレポートが生成されるようになりました。以前は、データの破損に遭遇すると、単純な WARN が生成されていましたが、これは見過ごされやすいものでした。**CONFIG_BUG_ON_DATA_CORRUPTION** を設定することで、カーネルがデータ破損に反応して、クラッシュを作成し、BUG をトリガーするようになりました。これにより、さらなる損傷が回避され、セキュリティーリスクが低減されます。**kdump** が **vmcore** を生成するようになり、データ破損のバグレポートが改善されました。

(BZ#1714330)

Intel Carlsville カードのサポートが追加されましたが、RHEL 8.2 で検証されていません。

Intel Carlsville カードのサポートを利用できるようになりましたが、Red Hat Enterprise Linux 8.2 でテストされていません。

(BZ#1720227)

RPS および XPS は、分離された CPU にジョブを配置しなくなりました

以前は、Receive Packet Steering (RPS) ソフトウェアキューメカニズムと Transmit Packet Steering (XPS) 送信キュー選択メカニズムが、分離された CPU を含むすべての CPU セットにジョブを割り当てていました。その結果、遅延の影響を受けやすいワークロードが RPS または XPS ジョブが実行されているのと同じ CPU を使用していた場合は、リアルタイム環境で予期しない遅延スパイクが発生する可能性があります。今回の更新により、**store_rps_map()** 関数には、RPS 設定の目的で分離された CPU が含まれなくなりました。同様に、XPS 設定に使用されるカーネルドライバーは、CPU 分離を考慮しています。その結果、RPS および XPS は、説明されているシナリオで分離された CPU にジョブを配置しなくなりました。`/sys/devices/pci*/net/dev/queues/rx-*/rps_cpus` ファイルで分離された CPU を設定すると、次のエラーが表示されます。

```
Error: "-bash: echo:write error: Invalid argument"
```

ただし、`/sys/devices/pci*/net/dev/queues/tx-*/xps_cpus` ファイルで分離された CPU を手動で設定すると、分離された CPU に XPS ジョブが正常に割り当てられます。

分離された CPU を使用する環境でのネットワークワークロードでは、パフォーマンスが多少変動する可能性があることに注意してください。

(BZ#1867174)

5.4.8. ファイルシステムおよびストレージ

SCSI ドライバーがメモリーを過剰に使用しなくなりました。

以前は、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した。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っていました。

メモリー使用量の増加は、ブロックレイヤーでのメモリーの事前割り当てにより発生していました。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たため、メモリー使用量が増えていました。

今回の更新で、ブロック層がメモリーの事前割り当て量を制限するようになり、その結果、SCSI ドライバーが過剰な量のメモリーを使用することはなくなりました。

(BZ#1698297)

UDS が再構築を終了する前に VDO を一時停止できるようになりました。

以前のリリースでは、UDS インデックスの再構築中に VDO ボリュームを一時停止しようとする **dmsetup suspend** コマンドが応答しなくなりました。このコマンドは、再構築後にのみ終了します。

今回の更新で、この問題が修正されています。**dmsetup suspend** コマンドは、応答しなくなることなく UDS の再構築が完了する前に終了できます。

(BZ#1737639)

5.4.9. 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

mod_cgid ログイングの問題が修正されました。

今回の更新以前は、**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、以下のログイング問題が発生していました。

- CGI スクリプトの **stderr** 出力に、標準のタイムスタンプ情報が付いていませんでした。
- **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でした。

今回の更新で問題が修正され、**mod_cgid** ログイングが期待どおりに動作するようになりました。

(BZ#1633224)

5.4.10. コンパイラーおよび開発ツール

dlopen の失敗時に、未配置で初期化されていない共有オブジェクトが失敗することがなくなりました。

以前は、**dlopen** 呼び出しに失敗すると、**glibc** 動的リンカーは、エラー報告前に **NODELETE** マークの付いた共有オブジェクトを削除しませんでした。そのため、未配置で初期化されていない共有オブジェクトがプロセスイメージに残され、最終的にはアサーションの失敗またはクラッシュにつながっていました。今回の更新で、動的ローダーは **dlopen** 失敗時に、保留中の **NODELETE** 状態を使用して共有オブジェクトを削除した後で、それらを永続的に **NODELETE** としてマークするようになりました。その結果、再配置されていないオブジェクトがプロセスに取り残されることはありません。また、ELF コンストラクターおよびデストラクターの実行中に遅延結合が失敗すると、プロセスが終了するようになりました。

(BZ#1410154)

64 ビット ARM アーキテクチャーの Advanced SIMD 関数は遅延解決時に誤ってコンパイルしなくなりました。

以前のリリースでは、Advanced SIMD の新しいベクトルプロシーチャーコール標準 (PCS) で、Advanced SIMD 機能の遅延解決時に、特定の呼出先退避レジスターが正しく保存、復元されませんでした。そのため、ランタイム時にバイナリーが誤作動する可能性がありました。今回の更新では、シンボルテーブルの Advanced SIMD と SVE ベクトル関数は **.variant_pcs** とマーク付されたので、動的なリンカーはこのような関数を初期にバインドするようになります。

(BZ#1726641)

sudo ラッパースクリプトはオプションを解析するようになりました。

以前のリリースでは、**/opt/redhat/devtoolset*/root/usr/bin/sudo** ラッパースクリプトは **sudo** オプションを正しく解析しませんでした。そのため、一部の **sudo** オプション (**sudo -i** など) を実行できませんでした。今回の更新で、**sudo** オプションが正しく解析され、**sudo** ラッパースクリプトが **/usr/bin/sudo** と同様に動作するようになりました。

(BZ#1774118)

glibc での TLS 変数の調整が修正されました。

以前のリリースでは、特定の条件下で、調整されたスレッドローカルストレージ (TLS) データは、想定外の調整なしにインスタンス化される可能性がありました。今回の更新で、POSIX Thread Library **libpthread** が拡張され、どのような条件下でも正しい調整を行うようになりました。これにより、調整された TLS データは、正しく調整されすべてのスレッドに対して適切にインスタンス化されるようになりました。

(BZ#1764214)

EINTR または EAGAIN エラーの発生後に pututxline 呼び出しを繰り返しても、utmp ファイルが破損しなくなりました。

pututxline 関数がロックを取得しようとして、時間内に成功しなかった場合には、関数は **EINTR** または **EAGAIN** のエラーコードを返します。以前のリリースでは、は、**pututxline** がすぐに呼び出されて、ロックを取得するように管理できた場合に、**utmp** ファイルにすでに割り当てられている対応のスロットを使用せず、別のエントリーを追加していました。そのため、このような未使用のエントリーにより、**utmp** ファイルのサイズが大幅に増大していました。今回の更新で問題が修正され、エントリーが **utmp** ファイルに正しく追加されるようになりました。

(BZ#1749439)

内部障害が発生しても mtrace がハングしなくなりました。

以前のリリースでは、**mtrace** ツールの実装に不具合が原因で、メモリートレースがハングする可能性がありました。この問題を修正するために、**mtrace** メモリートレースの実装が堅牢化され、内部障害の発生時にもハングが発生しなくなりました。その結果、**mtrace** を呼び出しても、ハングしなくなり、制限時間内に完了するようになりました。

(BZ#1764235)

fork 関数は pthread_atfork を使用すると発生する特定のデッドロックを回避するようになりました。

以前のリリースは、プログラムが **atfork** ハンドラーを登録し、非同期署名ハンドラーから **fork** を呼び出すと、内部実装依存ロックの不具合により、プログラムがフリーズする可能性がありました。今回の更新で、**フォーク** とその **atfork** ハンドラーの実装を調整し、シングルスレッドプログラムでデッドロックを回避できるようになりました。

(BZ#1746928)

strstr は、省略されたパターンに対して、正しい一致内容を返すようになりました。

一部の IBM Z プラットフォーム (z15、以前は arch13) では、**strstr** 関数は、ページの境界をまたぐ検索パターンを処理する場合に CPU レジスターが適切に更新されませんでした。これにより、**strstr** は誤った一致内容を返していました。今回の更新でこの問題が修正され、上記のシナリオで **strstr** が期待どおりに動作するようになりました。

(BZ#1777241)

glibc における C.UTF-8 ロケールソースの省略表現が修正されました。

以前は、C.UTF-8 ソースロケールに不具合があると、U+10000 を超えるすべての Unicode コードポイントに照合の重み付けが欠けていました。これにより、U+10000 を超えるすべてのコードポイントが期待どおりに照合されませんでした。C.UTF-8 ソースロケールが修正され、新たにコンパイルされたバイナリーロケールでは、すべての Unicode コードポイントが照合の重み付けを持つようになりました。コンパイルされた C.UTF-8 ロケールは、この修正により 5.3MiB 大きくなります。

(BZ#1361965)

setpwent() を呼び出しせずに getpwent() が呼び出されても glibc が失敗しなくなりました。

/etc/nsswitch.conf ファイルが Berkeley DB (**db**) パスワードプロバイダーを参照する場合は、最初に **setpwent()** を1度のみ呼び出すことなく、**getpwent()** 関数を使用してデータを要求できます。**endpwent()** 関数を呼び出すとき、最初に **setpwent()** を呼び出さずに **getpwent()** に呼び出しを行うと、**glibc** が失敗していました。これは、**endpwent()** が新しいクエリーを許可するように内部をリ

セットできなかったためです。今回の更新でこの問題が修正されています。その結果、**endpwent()** で1つのクエリーを終了すると、**setpwent()** を呼び出さなくても、**getpwent()** への呼び出しがさらに新しいクエリーを開始します。

(BZ#1747502)

ltrace は、強化されたバイナリーでのシステムコールを追跡できるようになりました。

以前のリリースでは、**ltrace** を使用して、AMD および Intel 64 ビットアーキテクチャー上のシステムバイナリーなど、特定の強化されたバイナリーで結果が生成されませんでした。今回の更新で、**ltrace** が、強化されたバイナリーでのシステムコールを追跡できるようになりました。

(BZ#1655368)

Intel の JCC の不具合により、GCC コンパイラーでパフォーマンスが大幅に低下することがなくなりました。

特定の Intel CPU は、マシン命令が誤って実行される原因となるジャンプ条件コード (JCC) バグの影響を受けます。したがって、影響を受けた CPU は、プログラムを適切に実行しない可能性があります。完全な修正には、脆弱な CPU のマイクロコードの更新が含まれており、パフォーマンス低下につながる可能性があります。今回の更新で、パフォーマンスの低下を軽減するのに役立つアセンブラーでの回避策が可能になりました。回避策は、デフォルトでは有効になっていません。

回避策を適用するには、GCC で **-Wa,-mbranches-within-32B-boundaries** コマンドラインオプションを使用して、プログラムを再コンパイルします。このコマンドラインオプションで再コンパイルしたプログラムは、JCC の不具合の影響を受けませんが、システムを完全に保護するにはマイクロコードの更新が依然として必要になります。

回避策を適用するとプログラムのサイズが増大します。再コンパイルを行わなかった場合よりは軽微ではあるものの、パフォーマンスがわずかに低下する可能性があることに留意してください。

(BZ#1777002)

並行ビルド中に **make** の速度が低下することがなくなりました。

以前のバージョンでは、並行ビルドの実行中に、シーケンスの実行を待機すると、**make** サブプロセスが一時的に応答しなくなる可能性があります。その結果、**-j** の値が高いビルドは遅延するか、効果的な **-j** の値が低いで実行されました。今回の更新では、**make** のジョブ制御ロジックがブロックされなくなりました。その結果、**-j** の値が高いビルドが完全な **-j** の速度で実行されます。

(BZ#1774790)

ltrace ツールが関数呼び出しを正しく報告するようになりました。

すべての RHEL コンポーネントに適用されるバイナリー強化を改善したことが原因で、以前は、**ltrace** ツールで RHEL コンポーネントからのバイナリーファイルの関数呼び出しを検出できませんでした。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ず、空になっていました。今回の更新で、**ltrace** が関数呼び出しを処理する方法が修正され、上記の問題が発生しなくなりました。

(BZ#1618748)

5.4.11. ID 管理

dsctl ユーティリティーが、名前にハイフンが付いたインスタンスの管理に失敗しなくなりました。

以前は、**dsctl** ユーティリティーは Directory Server インスタンス名のハイフンを正しく解析しません

でした。そのため、管理者は **dsctl** を使用して名前にハイフンを含むインスタンスを管理することができませんでした。今回の更新でこの問題が修正され、上記のシナリオで **dsctl** が期待どおりに動作するようになりました。

(BZ#1715406)

Directory Server インスタンス名が、最大 103 文字になりました。

LDAP クライアントが Directory Server への接続を確立すると、サーバーはクライアントアドレスに関連する情報をローカルバッファに保存します。以前は、このバッファのサイズが小さすぎて、46 文字を超える LDAP API パス名を格納できませんでした。たとえば、Directory Server インスタンスの名前が長すぎる場合などがこれに該当します。これにより、バッファオーバーフローが原因で、サーバーが突然終了していました。今回の更新で、Netscape Portable Runtime (NSPR) ライブラリーがパス名用にサポートするバッファサイズが最大サイズに増えます。その結果、上述のシナリオで Directory Server がクラッシュしなくなりました。

NSPR ライブラリーの制限により、インスタンス名は最大 103 文字になる可能性があることに留意してください。

(BZ#1748016)

pkidestroy ユーティリティーが正しいインスタンスを選択するようになりました。

以前のリリースでは、半分削除されたインスタンスで **pkidestroy --force** コマンドを実行すると、**-i instance** でインスタンス名を指定しても、デフォルトで **pki-tomcat** インスタンスが選択されていました。

これにより、目的のインスタンスではなく、**pki-tomcat** インスタンスが削除され、**--remove-logs** オプションを指定しても、目的のインスタンスのログが削除されませんでした。**pkidestroy** は、正しいインスタンス名を適用し、目的のインスタンスの残り物のみを削除するようになりました。

(BZ#1698084)

sssd-ldap の man ページで ldap_user_authorized_service の説明が更新されました。

RHEL 8 では、プラグ可能な認証モジュール (PAM) スタックが変更されました。たとえば、**systemd** ユーザーセッションは、PAM サービス **systemd-user** を使用して PAM 対話を開始するようになりました。このサービスは、PAM サービスの **system-auth** を再帰的に追加します。ここでは、**pam_sss.so** インターフェイスが含まれる場合もあります。これは、SSSD アクセス制御が常に呼び出されることを意味します。

RHEL 8 システムのアクセス制御ルールを規定する場合は、変更にご注意してください。たとえば、**systemd-user** サービスを、許可されたサービスリストに追加できます。

IPA HBAC、AD GPO などの一部のアクセス制御メカニズムでは、**systemd-user** サービスが、許可されたサービスリストにデフォルトで追加されているため、何もする必要はありません。

sssd-ldap の man ページが更新され、上記の情報が追加されるようになりました。

(BZ#1669407)

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るようになりました。

以前のリリースでは、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。IdM で必要とされる全 DNS レコードのリストを取得するには、**ipa dns-update-system-records --dry-run** コマンドを手動で入力する必要がありました。

今回の更新で、**ipa-adtrust-install** コマンドは、DNS ゾーンに手動で追加する DNS サービスレコードを正しくリスト表示するようになりました。

(BZ#1665051)

再帰 DNS クエリーは、統合 DNS を備えた IdM サーバーでデフォルトで無効になりました。

以前は、統合 DNS を備えた Identity Management (IdM) サーバーを使用する場合、再帰クエリーがデフォルトで有効になっていました。その結果、サーバーが DNS 増幅攻撃に使用される可能性があります。今回の更新により、再帰 DNS クエリーがデフォルトで無効になり、サーバーを DNS 増幅攻撃に使用できなくなりました。

IdM サーバーの統合 DNS 設定を変更することで、特定のクライアントの再帰クエリーを手動で許可できます。たとえば、信頼できるクライアントのみがキャッシュと再帰にアクセスできるように、誰でもサーバーにクエリーを実行して信頼できるデータを取得できるようにするには、**trusted** アクセス制御リスト (ACL) にクライアントをリストします。

1. **/etc/named/ipa-ext.conf** ファイルに **trusted** ACL を作成します。

```
acl "trusted" {
    192.168.0.0/16;
    10.153.154.0/24;
    localhost;
    localnets;
};
```

2. **trusted** ACL を **/etc/named/ipa-options-ext.conf** ファイルに追加します。

```
allow-query { any; };
allow-recursion { trusted; };
allow-query-cache { trusted; };
```

(BZ#2151696)

5.4.12. デスクトップ

Wayland 上の GNOME Shell でソフトウェアレンダラーを使用しても、動作が遅くならなくなりました。

以前は、GNOME Shell の Wayland バックエンドは、ソフトウェアレンダラーの使用時にキャッシュ可能なフレームバッファを使用していませんでした。そのため、Wayland 上のソフトウェアレンダリングされた GNOME Shell は、X.org バックエンド上のソフトウェアレンダリングされた GNOME Shell と比較して遅くなっていました。

今回の更新で、Wayland 上の GNOME Shell に中間シャドウフレームバッファが追加されました。その結果、Wayland 上のソフトウェアレンダリングされた GNOME Shell が、X.org 上の GNOME Shell と同じように実行できるようになりました。

(BZ#1737553)

5.4.13. 仮想化

第 10 世代の Intel Core プロセッサにおける仮想マシンの起動に失敗しなくなりました。

以前は、第 10 世代 Intel Core プロセッサを使用するホストモデル (Icelake-Server ともいう) では、仮想マシン (VM) の起動に失敗していました。今回の更新で、**libvirt** が QEMU でサポートされていな

い **pconfig** CPU 機能の無効化を試みなくなりました。その結果、第 10 世代 Intel プロセッサを実行するホストモデルでの仮想マシンの起動に失敗しなくなりました。

([BZ#1749672](#))

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が正しく機能するようになりました。

以前のリリースでは、**cloud-init** コーティリティーを使用して、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でした。今回の更新で、Azure エンドポイントの **cloud-init** 処理が修正され、Azure での RHEL 8 仮想マシンのプロビジョニングが期待通りに実行されるようになりました。

([BZ#1641190](#))

RHEL 7 ホスト上の RHEL 8 仮想マシンが、1920x1200 を超える解像度で確実に表示できません。

以前は、RHEL 7 ホストシステムで RHEL 8 仮想マシンを実行している場合は、kiosk モードでアプリケーションを実行するなど、仮想マシンのグラフィカル出力を表示する方法によっては、1920x1200 を超える解像度を表示することはできませんでした。そのため、ホストハードウェアがより高い解像度に対応している場合でも、この方法での仮想マシンを表示できる解像度は最大 1920x1200 のみでした。今回の更新で、上述の問題が発生しないように DRM ドライバーおよび QXL ドライバーが調整されています。

([BZ#1635295](#))

cloud-init を使用する ESXi 仮想マシンのカスタマイゼーションと、仮想マシンの再起動が正常に動作するようになりました。

以前は、**cloud-init** サービスを使用して VMware ESXi ハイパーバイザーで実行している仮想マシンを修正し、静的 IP を使用して仮想マシンをクローンすると、新しいクローンの仮想マシンを再起動するのにかかる時間が非常に長くなる場合があります。今回の更新で、**cloud-init** が仮想マシンの静的 IP を DHCP に書き直さないように修正され、上記の問題が発生しなくなりました。

([BZ#1666961](#), [BZ#1706482](#))

5.4.14. コンテナ

quay.io レジストリーからイメージをプルする場合に、意図しないイメージが生じなくなりました。

以前のリリースでは、`/etc/containers/registries.conf` 内のデフォルトのレジストリー検索リストに quay.io コンテナイメージレジストリーを記載した場合に、短縮名を使用すると、スプーフィングされたイメージをプルできていました。この問題を修正するため、quay.io コンテナイメージレジストリーが、`/etc/containers/registries.conf` のデフォルトのレジストリー検索リストから削除されました。その結果、quay.io レジストリーからイメージをプルする場合、ユーザーは完全なレジストリー名 (`quay.io/myorg/myimage` など) を指定することが必要になりました。Quay.io レジストリーは、`/etc/containers/registries.conf` のデフォルトのレジストリー検索リストに再び追加し、短縮名を使用したコンテナイメージのプルを有効化し直すことができますが、セキュリティリスクが生じる可能性があるため、これは推奨されません。

([BZ#1784267](#))

5.5. テクノロジープレビュー

本パートでは、Red Hat Enterprise Linux 8.2 で利用可能なテクノロジープレビュー機能のリストを提示します。

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。

5.5.1. ネットワーク

nmstate がテクノロジープレビューとして利用可能になりました。

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

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

(BZ#1674456)

AF_XDP がテクノロジープレビューとして利用可能に

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

(BZ#1633143)

XDP がテクノロジープレビューとして利用可能になりました。

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

(BZ#1503672)

KTLS がテクノロジープレビューとして利用可能になりました。

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

(BZ#1570255)

テクノロジープレビューとして、NetworkManager サポートで、dracut ユーティリティーが initrd イメージの作成をサポートするようになりました。

デフォルトでは、**dracut** ユーティリティーはシェルスクリプトを使用して初期 RAM ディスク (**initrd**) のネットワークを管理します。特定のケースでは、ネットワークの設定に NetworkManager を使用するオペレーティングシステムから RAM ディスクに、システムが切り替わると、問題が発生する可能性があります。たとえば、RAM ディスクのスクリプトが IP アドレスをすでに要求していた場合でも、NetworkManager が別の DHCP 要求を送信する可能性があります。RAM ディスクからのこの要求はタイムアウトする可能性があります。

このような問題を解決するために、RHEL 8.2 の **dracut** では、RAM ディスクで NetworkManager を使用できるようになりました。以下のコマンドを使用して機能を有効にし、RAM ディスクイメージを再作成します。

```
echo 'add_dracutmodules+=" network-manager "' > /etc/dracut.conf.d/enable-nm.conf
dracut -vf --regenerate-all
```

Red Hat はテクノロジープレビュー機能をサポートしないことに留意してください。ただし、この機能についてのフィードバックを提供する場合は、Red Hat サポートにご連絡ください。

(BZ#1626348)

mlx5_core ドライバーがテクノロジープレビューとして Mellanox ConnectX-6 Dx ネットワークアダプターに対応

今回の機能拡張により、Mellanox ConnectX-6 Dx ネットワークアダプターの PCI ID が **mlx5_core** ドライバーに追加されました。このアダプターを使用するホストでは、RHEL は **mlx5_core** ドライバーを自動的に読み込みます。Red Hat は、この機能をサポート対象外のテクノロジープレビューとして提供していることに注意してください。

(BZ#1687434)

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)

5.5.2. カーネル

kexec fast reboot 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。

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

(BZ#1769727)

eBPF がテクノロジープレビューとして利用可能になりました。

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

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

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

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

現在、以下の主要 eBPF コンポーネントが、テクノロジープレビューとして利用可能です。

- **bpftrace**。これは、eBPF 仮想マシンを使用する高レベルの追跡言語です。
- XDP (eXpress Data Path) 機能は、eBPF 仮想マシンを使用してカーネルでの高速パケット処理を有効にするネットワーキングテクノロジーです。

(BZ#1559616)

libbpf 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。

libbpf パッケージは現在、テクノロジープレビューとして利用できます。**libbpf** パッケージは、**bpftrace** および **bpf/xdp** 開発のようなアプリケーションに関連する bpf に極めて重要です。

これは、bpf-next Linux ツリー **bpf-next/tools/lib/bpf** ディレクトリーのミラーと、それがサポートするヘッダーファイルです。パッケージのバージョンは、Application Binary Interface (ABI) のバージョンを反映しています。

(BZ#1759154)

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

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

(BZ#1495358)

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)

5.5.3. ファイルシステムおよびストレージ

NVMe/TCP がテクノロジープレビューとして利用可能になりました。

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

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

NVMe/TCP ターゲットテクノロジープレビュー機能はテスト目的としてのみ同梱されており、現時点ではフルサポートの予定はありません。

(BZ#1696451)

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

Red Hat Enterprise Linux 8.2 では、ファイルシステムの DAX がテクノロジープレビューとして利用できます。DAX は、永続メモリーをそのアドレス空間に直接マッピングする手段をアプリケーションに提供します。DAX を使用するには、システムで利用可能な永続メモリーの形式が必要になります。通常は、NVDIMM (Non-Volatile Dual In-line Memory Module) の形式で、DAX に対応するファイルシステムを NVDIMM に作成する必要があります。また、ファイルシステムは **dax** マウントオプションでマウントする必要があります。これにより、dax をマウントしたファイルシステムのファイルの **mmap** が、アプリケーションのアドレス空間にストレージを直接マッピングされます。

(BZ#1627455)

OverlayFS

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

<https://www.kernel.org/doc/Documentation/filesystems/overlayfs.txt> を参照してください

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、コンテナエンジンのグラフィックドライバーとしての使用のみの対応となります。その使用は、コンテナの COW コンテンツのみに対応し、永続ストレージには対応していません。非 OverlayFS ボリュームに永続ストレージを配置する必要があります。デフォルトのコンテナエンジン設定のみを使用できます。つまり、あるレベルのオーバーレイ、1つの下位ディレクトリー、および下位と上位の両方のレベルが同じファイルシステムにあります。
- 下層ファイルシステムとして使用に対応しているのは現在 XFS のみです。

また、OverlayFS の使用には、以下のルールと制限が適用されます。

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

redirect_dir=on オプションおよび **index=on** オプションを使用して、POSIX コンプライアンスを向上させることもできます。この 2つのオプションにより、上位レイヤーの形式

は、このオプションなしでオーバーレイと互換性がありません。つまり、**redirect_dir=on** または **index=on** でオーバーレイを作成し、オーバーレイをアンマウントしてから、このオプションなしでオーバーレイをマウントすると、予期しない結果またはエラーが発生することがあります。

- XFS で使用されるコマンド:
 - XFS ファイルシステムは、オーバーレイとして使用する **-n ftype=1** オプションを有効にして作成する必要があります。
 - システムのインストール時に作成された rootfs およびファイルシステムを使用して、Anaconda キックスタートに **--mkfsoptions=-n ftype=1** パラメーターを設定します。
 - インストール後に新しいファイルシステムを作成する場合は、**# mkfs -t xfs -n ftype=1 /PATH/TO/DEVICE** コマンドを実行します。
 - 既存の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、**# xfs_info /PATH/TO/DEVICE | grep ftype** 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。
- 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デフォルトで有効になっています。
- このリリースでは、OverlayFS に関連する既知の問題がいくつかあります。詳細は、Linux カーネルドキュメントの **Non-standard behavior** (<https://www.kernel.org/doc/Documentation/filesystems/overlayfs.txt>) を参照してください。

(BZ#1690207)

Stratis 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。

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。

RHEL 8.2 は Stratis をバージョン 2.0.0 に更新します。このバージョンでは、信頼性と Stratis DBus API が改善されます。

(JIRA:RHELPLAN-1212)

IdM がテクノロジープレビューとして、IdM ドメインメンバーでの Samba サーバー設定に対応しました。

今回の更新で、Identity Management (IdM) ドメインメンバーに Samba サーバーを設定できるようにな

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

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

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

(JIRA:RHELPLAN-13195)

5.5.4. 高可用性およびクラスター

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

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

(BZ#1619620)

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

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

(BZ#1784200)

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

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

ヒューリスティックエージェントが、実際のフェンシングを行うフェンスエージェントと同じフェンシングレベルで設定されていて、そのエージェントよりも順番が前に設定されているとします。その場合、フェンシングを行うエージェントで **off** 操作を行う前に、ヒューリスティックエージェントで、この操作を行います。このヒューリスティックエージェントが **off** アクションに対して失敗する場合、このフェンシングレベルが成功しないのはすでに明らかです。そのため、Pacemaker フェンシングは、フェンシングを行うエージェントで **off** 操作を行うステップをスキップします。ヒューリスティックエージェントはこの動作を利用して、特定の条件下で、実際のフェンシングを行うエージェントがフェンシングできないようにできます。

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

(BZ#1775847)

5.5.5. ID 管理

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

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

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

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

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

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

(BZ#1664719)

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

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

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

- 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)

テクノロジープレビューとして、公開鍵インフラストラクチャーの全体的な正常性を確認できるようになりました。

今回の更新で、公開鍵インフラストラクチャー (PKI) Healthcheck ツールが、RHEL 8.1 で導入された Identity Management (IdM) Healthcheck ツールに PKI サブシステムの正常性を報告するようになり

した。IdM Healthcheck を実行すると、PKI Healthcheck が呼び出され、これにより、PKI サブシステムの正常性レポートを収集して返します。

pki-healthcheck ツールは、デプロイ済みのあらゆる RHEL IdM サーバーまたはレプリカで利用可能です。**pki-healthcheck** が提供するすべてのチェックは、**ipa-healthcheck** ツールにも統合されます。**ipa-healthcheck** は、**idm:DL1** モジュールストリームから個別にインストールできます。

pki-healthcheck は、スタンドアローンの Red Hat Certificate System (RHCS) インフラストラクチャーでも動作可能であることに留意してください。

(BZ#1303254)

5.5.6. デスクトップ

64 ビット ARM アーキテクチャーの GNOME がテクノロジープレビューとして利用できるよくなりました。

GNOME デスクトップ環境がテクノロジープレビューとして、64 ビット ARM アーキテクチャーで利用可能になりました。これにより、管理者は VNC セッションを使用して、グラフィカルユーザーインターフェイス (GUI) からサーバーをリモートで設定し、管理できます。

そのため、64 ビット ARM アーキテクチャーで新しい管理アプリケーションが利用できるようになりました。たとえば、**Disk Usage Analyzer (baobab)**、**Firewall Configuration (firewall-config)**、**Red Hat Subscription Manager (subscription-manager)**、または **Firefox Web ブラウザー** などです。**Firefox** を使用すると、管理者はローカルの Cockpit デーモンにリモートで接続できます。

(JIRA:RHELPLAN-27394, BZ#1667516, BZ#1667225, BZ#1724302)

5.5.7. グラフィックインフラストラクチャー

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

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

(BZ#1698565)

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

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

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

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

postfix ロールは、テクノロジープレビューとして利用可能です。

以下のロールが完全にサポートされています。

- **kdump**

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

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

(BZ#1812552)

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

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

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

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

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

注記: RHEL 8.2 for SAP Solutions は、Intel 64 アーキテクチャーおよび IBM POWER9 で SAP HANA とともに使用できるように検証される予定です。他の SAP アプリケーションやデータベース製品 (例: SAP NetWeaver および SAP ASE) のサポートは GA リリースに関連付けられており、お客様は、RHEL 8.2 の機能を一般提供 (GA) 後に利用できます。検証されたリリースと SAP サポートの最新情報は、SAP Notes 2369910 および 2235581 を参照してください。

(BZ#1660832)

5.5.9. 仮想化

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

テクノロジープレビューとして、Hyper-V ハイパーバイザーで実行している Red Hat Enterprise Linux のゲストオペレーティングシステムは、**ixgbevf** および **i40evf** ドライバーがサポートする Intel ネットワークアダプターに、シングルルート I/O 仮想化 (SR-IOV) 機能を使用することができるようになりました。この機能は、以下の条件が満たされると有効になります。

- ネットワークインターフェイスコントローラー (NIC) に対して SR-IOV サポートが有効になっている
- 仮想 NIC の SR-IOV サポートが有効になっている
- 仮想スイッチの SR-IOV サポートが有効になっている

- NIC の VF (Virtual Function) は、仮想マシンに接続されている

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

(BZ#1348508)

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

ネストされた KVM 仮想化は、テクノロジープレビューとして、Microsoft Hyper-V ハイパーバイザーで使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)

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 509 台以下の稼働中の仮想マシンに対応します。

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

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

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

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

Intel vGPU

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

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

(BZ#1528684)

5.5.10. コンテナ

skopeo コンテナイメージが、テクノロジープレビューとして利用可能になりました。

registry.redhat.io/rhel8/skopeo コンテナイメージは、**skopeo** パッケージをコンテナ化した実装です。**skopeo** は、コンテナイメージおよびイメージリポジトリでさまざまな操作を実行するコマンドラインツールユーティリティです。このコンテナイメージを使用すると、コンテナイメージを検査して、認証されていない、あるコンテナレジストリーから別のコンテナレジストリーにコピーすることができます。

([BZ#1627900](#))

buildah コンテナイメージが、テクノロジープレビューとして利用可能になりました。

registry.redhat.io/rhel8/buildah コンテナイメージは、**buildah** パッケージをコンテナ化した実装です。**buildah** は、OCI コンテナイメージを簡単にビルドできるようにするツールです。このコンテナイメージを使用すると、システムに **buildah** パッケージをインストールしなくても、コンテナイメージをビルドできます。このユースケースでは、root 以外のユーザーとして rootless モードでこのイメージを実行することを説明しません。

([BZ#1627898](#))

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:RHELDOS-16861](#))

5.6. 非推奨になった機能

ここでは、Red Hat Enterprise Linux 8.2 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。

5.6.1. インストーラーおよびイメージの作成

複数のキックスタートコマンドおよびオプションが非推奨になりました。

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

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

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

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

(BZ#1642765)

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

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

(BZ#1637872)

5.6.2. ソフトウェア管理

`rpmbuild --sign` が非推奨になりました。

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

(BZ#1688849)

5.6.3. シェルおよびコマンドラインツール

`curl` の Metalink サポートが無効になりました。

Metalink を使用してダウンロードされたコンテンツのクレデンシャルとファイルハッシュの不一致を処理する方法で、curl 機能に欠陥が見つかりました。この欠陥により、悪意のある攻撃者がホスティングサーバーを制御して次のことが可能になります。

- ユーザーをだまして悪意のあるコンテンツをダウンロードさせる
- ユーザーの知らないうちに、提供された認証情報への不正アクセスを取得する

この脆弱性による最大の脅威は、機密性と完全性です。これを回避するために、curl の Metalink サポートは Red Hat Enterprise Linux 8.2.0.z から無効になっています。

回避策として、Metalink ファイルをダウンロードした後、次のコマンドを実行します。

```
wget --trust-server-names --input-metalink`
```

以下に例を示します。

```
wget --trust-server-names --input-metalink <(curl -s $URL)
```

(BZ#1999620)

5.6.4. セキュリティー

NSS SEED 暗号が非推奨になりました。

Mozilla Network Security Services (**NSS**) ライブラリーでは、今後のリリースで SEED 暗号化を使用する TLS 暗号スイートのサポートがなくなります。SEED 暗号に依存するデプロイメントの場合には、Red Hat は、他の暗号スイートのサポートを有効にすることを推奨します。他の暗号スイートを有効にすることで、NSS によるサポートが取り消された場合にも、スムーズな移行が可能になります。

RHEL では、SEED 暗号はデフォルトですでに無効にされています。

(BZ#1817533)

TLS 1.0 および TLS 1.1 が非推奨になりました。

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

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

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

(BZ#1660839)

RHEL 8 で DSA が非推奨になりました。

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

(BZ#1646541)

NSS で SSL2 Client Hello が非推奨に

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

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

(BZ#1645153)

TPM 1.2 が非推奨になりました。

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

(BZ#1657927)

5.6.5. 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)

5.6.6. カーネル

ディスクレスブートを使用した RHEL for Real Time 8 のインストールが非推奨になりました。

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

(BZ#1748980)

qla3xxx ドライバーが非推奨になりました。

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

(BZ#1658840)

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

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

(BZ#1660627)

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)

5.6.7. ファイルシステムおよびストレージ

elevator カーネルコマンドラインパラメーターが非推奨になりました。

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

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

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

詳しい情報は、[ディスクスケジューラーの設定](#) を参照してください。

(BZ#1665295)

LVM mirror が非推奨化されました。

LVM **mirror** セグメントタイプが非推奨になりました。**mirror** のサポートは、RHEL の今後のメジャーリリースで削除されます。

Red Hat は、セグメントタイプが **mirror** ではなく、**raid1** の LVM RAID 1 デバイスを使用することを推奨します。**raid1** のセグメントタイプは、デフォルトの RAID 設定タイプで、**mirror** の代わりに、推奨のソリューションとしてこのタイプが使用されます。

mirror デバイスを **raid1** に変換するには、[ミラーリングされた LVM デバイスの RAID1 論理ボリュームへの変換](#) を参照してください。

LVM **mirror** には既知の問題が複数あります。詳細は、[ファイルシステムおよびストレージの既知の問題](#) を参照してください。

(BZ#1827628)

NFSv3 over UDP が無効になりました。

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

RHEL 8 では、NFS over UDP に対応しなくなりました。

(BZ#1592011)

5.6.8. Identity Management

SMB1 プロトコルは Samba では非推奨に

Samba 4.11 以降、安全でない Server Message Block バージョン 1 (SMB1) プロトコルは非推奨となり、今後のリリースでは削除される予定です。

セキュリティーを向上させるために、デフォルトでは、Samba サーバーおよびクライアントユーティリティーで SMB1 が無効になっています。

(JIRA:RHELDPCS-16612)

5.6.9. デスクトップ

libgnome-keyring ライブラリーが非推奨になりました。

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

(BZ#1607766)

5.6.10. グラフィックインフラストラクチャー

AGP グラフィックカードがサポートされなくなりました。

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

(BZ#1569610)

5.6.11. Web コンソール

Web コンソールは、不完全な翻訳への対応を終了しました。

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

(BZ#1666722)

5.6.12. 仮想化

virt-manager が非推奨になりました。

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

(JIRA:RHELPLAN-10304)

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

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

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

(BZ#1686057)

Cirrus VGA 仮想 GPU タイプが非推奨に

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

(BZ#1651994)

cpu64-rhel6 CPU モデルが非推奨になり、削除されました。

cpu64-rhel6 QEMU 仮想 CPU モデルは、RHEL 8.1 で非推奨になり、RHEL 8.2 から削除されました。ホストマシンにある CPU に応じて、QEMU および libvirt が提供する他の CPU モデルを使用することが推奨されます。

(BZ#1741346)

5.6.13. 非推奨のパッケージ

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

5.7. 既知の問題

このパートでは Red Hat Enterprise Linux 8.2 の既知の問題を説明します。

5.7.1. インストーラーおよびイメージの作成

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

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

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

(BZ#1640697)

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

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

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

(BZ#1697896)

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

Anaconda は最小限のリソース設定を必要とするシステムでインストールを開始し、インストールを成功させるのに必要なリソースに関する以前のメッセージ警告を提供しません。その結果、インストールが失敗し、出力エラーでデバッグや復元の可能性を明確に示すメッセージが提供されない場合があります。この問題を回避するには、システムに、インストールに必要な最小リソース設定 (PPC64 (LE) の場合は 2GB メモリー、x86_64 の場合は 1GB) があることを確認します。これにより、インストールを成功できます。

(BZ#1696609)

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

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

(BZ#1672405)

SSH から RHEL 8 初期セットアップを実行できません

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

(BZ#1676439)

インストールプログラムでは、ネットワークアクセスがデフォルトで有効になっていません。

一部のインストール機能、たとえば、コンテンツ配信ネットワーク (CDN) を使用したシステムの登録、NTP サーバーサポート、およびネットワークインストールソースなどには、ネットワークアクセスが必要です。ただし、ネットワークアクセスはデフォルトでは有効になっていません。そのためこの機能は、ネットワークアクセスが有効になるまで使用できません。

この問題を回避するには、インストールの開始時にネットワークアクセスを有効にする起動オプション **ip=dhcp** を追加します。オプションで、起動オプションを使用して、ネットワーク上にあるキックスタートファイルまたはリポジトリを渡しても、問題が解決されます。結果として、ネットワークベースのインストール機能を使用できます。

(BZ#1757877)

複数の組織に属するユーザーアカウントの登録に失敗していました

現在、複数の組織に属するユーザーアカウントでシステムを登録しようとする、登録プロセスが失敗し、**You must specify an organization for new units**(新しいユニットの組織を指定する必要があります)。というメッセージが表示されます。

この問題を回避するには、以下のいずれかを行います。

- 複数の組織に属さない別のユーザーアカウントを使用します。
- GUI および Kickstart インストールの Connect to Red Hat 機能で利用できる **アクティベーションキー** 認証方法を使用します。
- Connect to Red Hat の登録手順を省略し、Subscription Manager を使用してインストール後にシステムを登録します。

(BZ#1822880)

Binary DVD ISO イメージを使用した GUI インストールを、CDN 登録なしで続行できないことがありました。

Binary DVD ISO イメージファイルを使用して GUI インストールを実行する場合は、Red Hat 機能への接続機能を使用してシステムを登録するまで、インストーラーの競合状態によりインストールが続行できなくなることがあります。この問題を回避するには、以下の手順を実施します。

1. GUI インストールのインストール **インストール概要** 画面から **インストールソース** を選択します。
2. **自動検出したインストールメディア** が選択されていることを確認します。
3. **完了** をクリックして選択を確定し、**インストール概要** 画面に戻ります。
4. インストール **概要** 画面に、**ローカルメディア** が **インストールソース** ステータスとして表示されることを確認します。

これにより、Red Hat への接続機能を使用してシステムを登録しなくてもインストールに進むことができます。

(BZ#1823578)

Binary DVD.iso ファイルのコンテンツをパーティションにコピーしても、.treeinfo ファイルおよび .discinfo ファイルはコピーされません。

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

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

(BZ#1687747)

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

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

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

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

以下に例を示します。

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

(BZ#1745064)

リポジトリの更新が完了する前に CDN を使用した登録解除を試みると、GUI インストールが失敗することがあります。

RHEL 8.2 では、システムを登録し、コンテンツ配信ネットワーク (CDN) を使用してサブスクリプションを割り当てると、GUI インストールプログラムにより、リポジトリメタデータの更新が開始されます。更新プロセスは、登録およびサブスクリプションプロセスの一部ではないため、Red Hat への接続ウィンドウで **登録解除** ボタンが有効になります。ネットワーク接続によっては、更新プロセスが完了するのに1分以上かかることがあります。更新プロセスが完了する前に **登録解除** ボタンをクリックすると、登録解除プロセスで、インストールプログラムが CDN との通信に必要な証明書と CDN リポジトリファイルが削除されるため、GUI インストールが失敗する可能性があります。

この問題を回避するには、Red Hat への接続ウィンドウで **登録** ボタンをクリックした後に、GUI インストールで次の手順を実行します。

1. Red Hat への接続画面から **完了** をクリックして、インストールの概要画面に戻ります。
2. インストールの概要 ウィンドウで、インストールソース および ソフトウェアの選択 の斜体のステータスメッセージに処理情報が表示されていないことを確認します。
3. インストールソースとソフトウェアの選択のカテゴリーが準備できたら、Red Hat への接続をクリックします。

4. 登録解除 ボタンをクリックします。

これらの手順を完了したら、GUI のインストール時にシステムの登録を安全に解除できます。

(BZ#1821192)

5.7.2. サブスクリプション管理

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

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

(BZ#1687900)

キックスタートファイルを使用して RHEL をインストールすると、マルチパスストレージデバイスのデータが失われる

キックスタートファイルを使用して RHEL をインストールすると、ホストに接続されているマルチパスストレージデバイスのデータが失われます。この問題は、**ignoredisk --drives** コマンドを使用して指定したマルチパスストレージデバイスをインストーラーが無視できないため発生します。その結果、デバイス上のデータが失われます。

この問題を回避するには、インストール前にデバイスを切り離すか、**ignoredisk --only-use** コマンドを使用してインストールするデバイスを指定します。

(BZ#1862131)

5.7.3. シェルおよびコマンドラインツール

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

Red Hat Enterprise Linux 8 では、ほとんどのアプリケーションは、X11 プロトコルの代わりに、デフォルトで Wayland プロトコルを使用します。したがって、ssh サーバーは、リモートディスプレイサーバーに対し、Wayland プロトコルを使用するアプリケーションを転送できませんが、X11 プロトコルを使用するアプリケーションを転送することは可能です。

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

(BZ#1686892)

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

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

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

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

(BZ#1640802)

5.7.4. セキュリティー

Audit の実行可能な監視機能がシンボリックリンクで機能しない

-w オプションによって提供されるファイルモニタリングでは、パスを直接追跡できません。デバイスと inode へのパスを解決して、実行したプログラムとの比較を行う必要があります。実行可能なシンボリックリンクを監視する監視機能は、メモリーで実行されるプログラムではなく、デバイスとシンボリックリンク自体の inode を監視します。これは、シンボリックリンクの解決から確認できます。監視機能がシンボリックリンクを解決して作成される実行プログラムを取得する場合でも、ルールは別のシンボリックリンクから呼び出されるマルチコールバイナリーでトリガーされます。これにより、誤検出でログがいっぱいになります。したがって、Audit の実行可能な監視機能は、シンボリックリンクでは機能しません。

この問題を回避するには、プログラム実行可能ファイルの解決されたパスに対して監視を設定し、**comm=** や **proctitle=** フィールドに記載されている最後のコンポーネントを使用して、生成されるログメッセージをフィルターします。

(BZ#1846345)

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

/etc/selinux/config で **SELINUX=disabled** オプションを使用して SELinux を無効にすると、カーネルが SELinux を有効にして起動し、その後のブートプロセスで無効化モードに切り替わります。これにより、メモリーリークが生じる可能性があります。

この問題を回避するには、SELinux を完全に無効にする必要がある場合に [SELinux の使用のシステム起動時に SELinux モードの変更](#) で説明されているように、**selinux=0** パラメーターをカーネルコマンドラインに追加して SELinux を無効にすることが推奨されます。

(JIRA:RHELPLAN-34199)

libselinux-python は、そのモジュールからのみ利用可能

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

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

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

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

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

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

(BZ#1666328)

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

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

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

(BZ#1763210)

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

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

この問題を回避するには、以下を実行します。

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

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

(BZ#1641631)

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

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

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

(BZ#1746398)

SELinux が原因で **auditd** によるシステムの停止や電源オフができません。

SELinux ポリシーには、Audit デーモンが **power_unit_file_t systemd** ユニットを起動できるようにするルールがありません。したがって、Logging ディスクパーティションに領域が残っていない場合など、**auditd** がシステムを停止したり、電源をオフにしたりできるように設定した場合でも、システムの停止や電源オフができません。

この問題を回避するには、カスタムの SELinux ポリシーモジュールを作成します。こうすることで、**auditd** は回避策を適用している場合に限り、システムを正常に停止したり、電源をオフにしたりできます。

(BZ#1826788)

ユーザーは、ロックされたユーザーとして `sudo` コマンドを実行できます。

ALL キーワードで **sudoers** パーミッションが定義されているシステムでは、パーミッションを持つ **sudo** ユーザーは、アカウントがロックされているユーザーとして **sudo** コマンドを実行できます。そのため、ロックされたアカウントと期限切れのアカウントを使用して、コマンドを実行し続けることができます。

この問題を回避するには、`/etc/shells` 内の有効なシェルの適切な設定と併せて、新たに実装した **runas_check_shell** オプションを有効にします。これにより、攻撃者が **bin** などのシステムアカウントでコマンドを実行するのを防ぎます。

(BZ#1786990)

デフォルトのロギング設定がパフォーマンスに与える悪影響

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

詳細は、ナレッジベースの記事 [Negative effects of the RHEL default logging setup on performance and their mitigations](#) を参照してください。

(JIRA:RHELPLAN-10431)

config.enabled による rsyslog 出力の Parameter not known エラー。

rsyslog の出力では、**config.enabled** ディレクティブを使用すると、設定処理エラーで予期しないバグが発生します。これにより、**include()** ステートメントを除き、**config.enabled** ディレクティブを使用すると、**parameter not known** エラーが表示されます。

この問題を回避するには、**config.enabled=on** を設定するか、**include()** ステートメントを使用します。

(BZ#1659383)

特定の rsyslog 優先度の文字列が正常に動作しません。

imtcp に **GnuTLS** 優先度文字列を設定して、完成していない暗号化をきめ細かく制御できるようになりました。したがって、**rsyslog** では、以下の優先文字列が正常に動作しません。

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

この問題を回避するには、正しく機能する優先度文字列のみを使用します。

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

したがって、現在の設定は、正しく機能する文字列に限定する必要があります。

(BZ#1679512)

SHA-1 署名を使用するサーバーへの接続が GnuTLS で動作しません。

証明書の SHA-1 署名は、GuTLS セキュアな通信ライブラリーにより、セキュアでないものとして拒否されます。したがって、TLS のバックエンドとして GnuTLS を使用するアプリケーションは、このような証明書を提供するピアへの TLS 接続を確立することができません。この動作は、その他のシステム暗号化ライブラリーと一貫性がありません。この問題を回避するには、サーバーをアップグレードして、SHA-256 または強力なハッシュを使用して署名した証明書を使用するか、LEGACY ポリシーに切り替えます。

(BZ#1628553)

TLS 1.3 は、FIPS モードの NSS で動作しません。

TLS 1.3 は、FIPS モードで動作しているシステムでは対応していません。その結果、相互運用性に TLS 1.3 を必要とする接続が、FIPS モードで動作しているシステムで機能しません。

その接続を有効にするには、システムの FIPS モードを無効にするか、ピアで TLS 1.2 のサポートを有効にします。

(BZ#1724250)

OpenSSL が、生の RSA または RSA-PSS の署名に対応していない PKCS #11 トークンを誤って処理します。

OpenSSL ライブラリーは、PKCS #11 トークンの鍵関連の機能を検出しません。したがって、生の RSA または RSA-PSS の署名に対応しないトークンで署名が作成されると、TLS 接続の確立に失敗します。

この問題を回避するには、`/etc/pki/tls/openssl.cnf` ファイルの `crypto_policy` セクションの末尾にある `.include` 行の後に、以下の行を追加します。

```
SignatureAlgorithms =
RSA+SHA256:RSA+SHA512:RSA+SHA384:ECDSA+SHA256:ECDSA+SHA512:ECDSA+SHA384
MaxProtocol = TLSv1.2
```

これにより、このシナリオで TLS 接続を確立できます。

(BZ#1685470)

OpenSSL が、TLS 1.3 の CertificateRequest メッセージに、不正な status_request 拡張を生成します。

OpenSSL サーバーは、`status_request` 拡張とクライアント証明書ベースの認証サポートが有効な場合に、`CertificateRequest` メッセージに正しくない `status_request` 拡張を送信します。この場合、OpenSSL は RFC 8446 プロトコルに準拠する実装と同時に動作しません。その結果、`CertificateRequest` メッセージの拡張を適切に検証するクライアントは OpenSSL サーバーとの接続を中止します。この問題を回避するには、接続のいずれかで TLS 1.3 プロトコルのサポートを無効にするか、OpenSSL サーバーの `status_request` のサポートを無効にします。これにより、サーバーが不正なメッセージを送信しなくなります。

(BZ#1749068)

ssh-keyscan が、FIPS モードでサーバーの RSA 鍵を取得できません。

FIPS モードで RSA 署名の SHA-1 アルゴリズムが無効になっています。これにより、`ssh-keyscan` ユーティリティーがそのモードで稼働しているサーバーの RSA 鍵を取得できなくなります。

この問題を回避するには、代わりに ECDSA 鍵を使用するか、サーバーの `/etc/ssh/ssh_host_rsa_key.pub` ファイルから鍵をローカルに取得します。

(BZ#1744108)

Libreswan が、すべての設定において `seccomp=enabled` で正常に動作しません。

Libreswan SECCOMP サポート実装で許可された `syscall` のセットは現在、完全ではありません。したがって、SECCOMP が `ipsec.conf` ファイルで有効となっている場合、`syscall` のフィルタリングは、**pluto** デモンの正常な機能に必要な `syscall` まで拒否します。つまり、デモンは強制終了され、**ipsec** サービスが再起動されます。

この問題を回避するには、**seccomp=** オプションを設定して、**disabled** 状態に戻します。SECCOMP サポートは、**ipsec** を正常に実行するため、無効のままにしておく必要があります。

(BZ#1777474)

SSG における相互依存ルールの特定のセットが失敗する可能性があります。

ルールとその依存関係の順序付けを定義しないため、ベンチマークの **SCAP Security Guide (SSG)** ルールの修正が失敗する可能性があります。たとえば、特定の順番で複数のルールを実行する必要がある場合、あるルールがコンポーネントをインストールし、別のルールが同じコンポーネントを設定した場合すると、それらは正しくない順序で実行される可能性があり、修正によってエラーが報告されます。この問題を回避するには、修正を再実行して、番目の実行で依存ルールを修正します。

(BZ#1750755)

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ません。

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)

OSCAP Anaconda Addon がすべてのパッケージをテキストモードでインストールしません。

OSCAP Anaconda Addon プラグインは、インストールがテキストモードで実行している場合、システムインストーラーによってインストールに選択されているパッケージのリストを変更することはできません。これにより、キックスタートを使用してセキュリティポリシープロファイルが指定され、インストールがテキストモードで実行している場合に、インストール中にセキュリティポリシーに必要な追加パッケージがインストールされません。

この問題を回避するには、グラフィカルモードでインストールを実行するか、キックスタートファイルの **%packages** セクションにあるセキュリティポリシーで、セキュリティポリシープロファイルに必要なパッケージをすべて指定します。

これにより、セキュリティポリシープロファイルで必要となるパッケージは、上記の回避策のいずれかを行わなければ RHEL インストールインストール時にインストールされません。また、インストール後のシステムは、指定のセキュリティポリシープロファイルと互換性がありません。

(BZ#1674001)

oscap Anaconda Addon がカスタムプロファイルを正しく処理しません。

OSCAP Anaconda Addon プラグインは、個別のファイルでカスタマイズを使用したセキュリティープロファイルを適切に処理しません。これにより、対応する Kickstart セクションで適切に指定しても、RHEL グラフィカルインストールでカスタマイズしたプロファイルは利用できません。

この問題を回避するには、ナレッジベースの記事 [Creating a single SCAP data stream from an original DS and a tailoring file](#) を参照してください。この回避策により、RHEL グラフィカルインストールでカスタマイズした SCAP プロファイルを使用できます。

(BZ#1691305)

GnuTLS が NSS サーバーとの現行セッションを再開できません。

TLS (Transport Layer Security) 1.3 セッションを再開するとき、**GnuTLS** クライアントは 60 ミリ秒に加えて、サーバーがセッション再開データを送信するために推定されるラウンドトリップタイムの分だけ待機します。サーバーがこの時間内に再開データを送信しないと、クライアントは現行セッションを再開せずに新しいセッションを作成します。これは、通常のセッションネゴシエーションのパフォーマンスに若干影響することを除いて、深刻な悪影響を与えることはありません。

(BZ#1677754)

リモートシステムを --sudo でスキャンすると、oscap-ssh ユーティリティーが失敗します。

oscap-ssh ツールで **--sudo** オプションを使用してリモートシステムの Security Content Automation Protocol (SCAP) スキャンを実行すると、リモートシステムの **oscap** ツールが、**root** ユーザーとして、スキャン結果ファイルとレポートファイルを一時ディレクトリーに保存します。リモートマシンの **umask** 設定を変更した場合は、**oscap-ssh** がこれらのファイルにアクセスできない可能性があります。この問題を回避するには、ソリューション "[oscap-ssh --sudo](#)" fails to retrieve the result files with "[scp: ...: Permission denied](#)" error で説明されているように、**oscap-ssh** ツールを変更します。これにより、**oscap** はターゲットユーザーとしてファイルを保存し、**oscap-ssh** は普通にファイルにアクセスします。

(BZ#1803116)

OpenSCAP が、YAML マルチライン文字列から空の行を削除することにより、誤検出が生成されます。

OpenSCAP がデータストリームから Ansible 修正を生成すると、YAML マルチライン文字列から空の行が削除されます。一部の Ansible 修正にはリテラル設定のファイルコンテンツが含まれているため、空の行を削除すると、対応する修正に影響します。空の行に何の効果もないとしても、これにより **openscap** ユーティリティーが、対応する Open Vulnerability and Assessment Language (OVAL) チェックに失敗します。この問題を回避するには、ルールの説明を確認し、空の行がないために失敗したスキャン結果をスキップします。または、Bash の修正ではこれらの誤検出が発生しないので、Ansible の修正の代わりに Bash の修正を使用します。

(BZ#1795563)

OSPP ベースのプロファイルに、GUI パッケージグループとの互換性がありません。

Server with GUI パッケージグループでインストールされる **GNOME** パッケージには、Operating System Protection Profile (OSPP) に準拠しない **nfs-utils** パッケージが必要です。これにより、OSPP または OSPP ベースのプロファイル (Security Technical Implementation Guide (STIG) など) を使用したシステムのインストール時に **Server with GUI** パッケージグループを選択すると、インストールが中断されます。OSPP ベースのプロファイルがインストール後に適用される場合、システムは起動できません。この問題を回避するには、**Server with GUI** パッケージグループや、または OSPP プロファイルと

OSPP ベースのプロファイルを使用する際に GUI をインストールするその他のグループをインストールしないでください。代わりに**Server**または**Minimal Install**パッケージグループを使用すると、システムは問題なくインストールされ、正常に機能します。

(BZ#1787156)

RHEL 8 システムに Server with GUI パッケージグループが含まれる場合には e8 プロファイルを使用して修復できません。

OpenSCAP Anaconda Add-on を使用して、**Verify Integrity with RPM** グループからルールを選択するプロファイルが含まれる **Server With GUI** パッケージグループでシステムを強化すると、システム上のメモリーを過剰に必要とします。この問題は、OpenSCAP スキャナーが原因で発生します。詳細は、[Scanning large numbers of files with OpenSCAP causes systems to run out of memory](#) を参照してください。このため、RHEL8 Essential Eight (e8) プロファイルを使用してシステムを正常に強化できません。この問題を回避するには、サーバーなどの小規模なパッケージグループを選択して、インストール後に必要な追加パッケージをインストールします。その結果、システムにはパッケージ数が少なくなり、スキャンに必要なメモリーが少なくなるため、システムを自動的に強化できます。

(BZ#1816199)

OpenSCAP で多数のファイルをスキャンすると、システムがメモリーを使い切ってしまう。

OpenSCAP スキャナーは、スキャンが完了するまで、収集したすべての結果をメモリーに保存します。そのため、たとえば、**GUI** および **Workstation** を使用する大規模なパッケージグループから、大量のファイルをスキャンする場合、RAM が少ないシステムでメモリーが不足する可能性があります。この問題を回避するには、RAM が少ないシステムで、より小さいパッケージグループ (例: **Server** および **Minimal Install**) を使用します。大規模なパッケージグループを使用する必要がある場合は、システムで仮想環境またはステージング環境に十分なメモリーがあるかどうかをテストしてください。または、スキャンプロファイルを調整して、/ ファイルシステム全体の再帰を行う、以下のルールの選択を解除することができます。

- **rpm_verify_hashes**
- **rpm_verify_permissions**
- **rpm_verify_ownership**
- **file_permissions_unauthorized_world_writable**
- **no_files_unowned_by_user**
- **dir_perms_world_writable_system_owned**
- **file_permissions_unauthorized_suid**
- **file_permissions_unauthorized_sgid**
- **file_permissions_ungroupowned**
- **dir_perms_world_writable_sticky_bits**

選択を外すことで、OpenSCAP スキャンで、システムがメモリー不足になるのを防ぎます。

(BZ#1824152)

5.7.5. ネットワーク

GRO が無効になっていると、IPsec オフロード中に IPsec ネットワークトラフィックが失敗します。

デバイスで汎用受信オフロード (GRO) が無効になっていると、IPSec オフロードは機能しません。IPsec オフロードがネットワークインターフェイスで設定され、GRO がそのデバイスで無効になっていると、IPsec ネットワークトラフィックに失敗します。

この問題を回避するには、デバイスで GRO を有効にしたままにします。

(BZ#1649647)

iptables では、指定のチェーンタイプが不明な場合に、モジュールがチェーン更新コマンドを読み込むように要求されません。

注記: この問題は、サービスのデフォルト設定を使用している場合は、**iptables systemd** サービスを停止すると、機能的な影響のない偽のエラーが発生します。

iptables-nft でチェーンのポリシーを設定すると、関連付けられたカーネルモジュールがすでにロードされていない場合には、作成される更新チェーンコマンドのカーネルへの送信に失敗します。この問題を回避するには、以下のコマンドを使用してモジュールを読み込みます。

```
+
# iptables -t nat -n -L
# iptables -t mangle -n -L
```

(BZ#1812666)

nft_compat モジュールによる、アドレスファミリー固有の LOG バックエンドモジュールの自動読み込みがハングする可能性があります。

nft_compat モジュールが、ネットワークの名前空間 (**netns**) で並行して操作が実行されている時に、アドレスファミリー固有の **LOG** ターゲットバックエンドを読み込むと、ロックの競合が発生する可能性があります。そのため、アドレスファミリー固有の **LOG** ターゲットバックエンドを読み込むとハングする可能性があります。この問題を回避するには、**iptables-restore** ユーティリティを実行する前に、関連する **LOG** ターゲットバックエンド (**nf_log_ipv4.ko**、**nf_log_ipv6.ko** など) を手動で読み込みます。こうすることで、**LOG** ターゲットバックエンドの読み込みが停止しなくなります。ただし、システムの起動時に問題が発生した場合は、回避策はありません。

libvirtd などの他のサービスは、**iptables** コマンドも実行するため、問題が発生する可能性があることに注意してください。

(BZ#1757933)

5.7.6. カーネル

誤ってパッチが削除されると、huge_page_setup_helper.py でエラーが表示されます。

huge_page_setup_helper.py スクリプトを更新するパッチが誤って削除されました。これにより、**huge_page_setup_helper.py** の実行後に、以下のエラーメッセージが表示されます。

```
SyntaxError: Missing parentheses in call to 'print'
```

この問題を回避するには、RHEL 8.1 から **huge_page_setup_helper.py** スクリプトをコピーし、これを **/usr/bin/** ディレクトリーにインストールします。

1. RHEL-8.1.0 のインストールメディアまたは [Red Hat カスタマーポータル](#) から、**libhugetlbfs-utils-2.21-3.el8.x86_64.rpm** パッケージをダウンロードします。
2. **rpm2cpio** コマンドを実行します。

```
# rpm2cpio libhugetlbfs-utils-2.21-3.el8.x86_64.rpm | cpio -D / -iduv
*/huge_page_setup_helper.py'
```

このコマンドは、RHEL 8.1 RPM から **huge_page_setup_helper.py** スクリプトをデプロイメントして、`/usr/bin/` ディレクトリーに保存します。

これにより、**huge_page_setup_helper.py** スクリプトが正しく動作するようになります。

(BZ#1823398)

永続メモリーのサイズが大きくなると、システムの起動プロセス時に遅延が発生します。

メモリーの初期化がシリアル化されるため、永続メモリーのサイズが大きいシステムは起動に時間がかかります。したがって、`/etc/fstab` ファイルに永続メモリーのファイルシステムがあると、デバイスが利用できるようになるまで待つ際に、システムがタイムアウトする場合があります。この問題を回避するには、`/etc/systemd/system.conf` ファイルの **DefaultTimeoutStartSec** オプションを十分に大きな値に設定します。

(BZ#1666538)

KSM が、NUMA メモリーポリシーを無視することがあります。

merge_across_nodes=1 パラメーターで、カーネル共有メモリー (KSM) 機能を有効にすると、KSM は、`mbind()` 関数が設定したメモリーポリシーを無視し、一部のメモリーから、ポリシーに一致しない NUMA (Non-Uniform Memory Access) ノードにページをマージできない場合があります。

この問題を回避するには、KSM を無効にするか、QEMU で NUMA メモリーバインディングを使用する場合は **merge_across_nodes** パラメーターを **0** に設定します。これにより、KVM 仮想マシンに設定した NUMA メモリーポリシーが期待どおりに機能します。

(BZ#1153521)

RHEL 8 で、デバッグカーネルがクラッシュキャプチャー環境で起動に失敗します。

デバッグカーネルのメモリー要求の性質により、デバッグカーネルが使用中で、カーネルパニックが発生すると、問題が発生します。その結果、デバッグカーネルはキャプチャーカーネルとして起動できず、代わりにスタ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)

zlib は、一部の圧縮機能で **vmcore** キャプチャーの速度を低下させる可能性があります。

kdump 設定ファイルはデフォルトで、**lzo** 圧縮形式 (`makedumpfile -l`) を使用します。**zlib** 圧縮形式 (`makedumpfile -c`) を使用して設定ファイルを変更すると、**vmcore** のキャプチャープロセスの速度を低下させる代わりに、圧縮の因子が改善される可能性が高くなっています。これにより、**lzo** と比較して、**kdump** が **zlib** で **vmcore** をキャプチャーするのに最大 4 倍の時間がかかります。

このように、Red Hat は、速度が主要な要因である場合に、デフォルトの **lzo** を使用することを推奨します。ただし、ターゲットマシンで利用可能な領域が少ない場合は、**zlib** の方が適しています。

(BZ#1790635)

vmcore キャプチャーはメモリーのホットプラグまたはアンプラグの操作を実行した後に失敗します。

メモリーのホットプラグまたはホットアンプラグ操作の実行後に、メモリーのレイアウト情報を含むデバイスツリーを更新するとイベントが発生します。これにより、**makedumpfile** ユーティリティーは存在しない物理アドレスにアクセスしようとします。以下の条件を満たすと問題が発生します。

- IBM Power System (little endian) で RHEL 8 を実行する。
- システムで **kdump** サービスまたは **fadump** サービスが有効になっている。

このような場合に、メモリーホットプラグまたはホットアンプラグの操作後にカーネルクラッシュが発生すると、カーネルのキャプチャーで **vmcore** の保存に失敗します。

この問題を回避するには、ホットプラグまたはホットアンプラグ後に **kdump** サービスを再起動します。

```
# systemctl restart kdump.service
```

これにより、上記のシナリオで **vmcore** が正常に保存されます。

(BZ#1793389)

fadump ダンプメカニズムが、ネットワークインターフェイスの名前を **kdump-`<interface-name>`** に変更します。

ファームウェア支援ダンプ (**fadump**) を使用して **vmcore** を取得し、SSH または NFS プロトコルを使用してこれをリモートマシンに保存すると、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。名前変更は、`<interface-name>` が、`*eth#` や `net#` などのように一般的な場合に起こります。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頭辞 `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)

fadump が有効な場合、システムが起動時に緊急モードに切り替わります。

fadump (**kdump**) または **dracut** squash モジュールが **initramfs** スキームで有効化されると、システムが緊急モードに切り替わります。これは、**systemd** マネージャーがマウント情報の取得とマウントする LV パーティションの設定に失敗するためです。この問題を回避するには、以下のカーネルコマンドラインパラメーター **rd.lvm.lv=`<vg>/<LV>`** を追加し、失敗した LV パーティションを適切に検出してマウントします。これにより、上述のシナリオでシステムが正常に起動するようになります。

(BZ#1750278)

irqpoll を使用すると **vmcore** の生成に失敗します。

Amazon Web Services (AWS) クラウドプラットフォームで実行している 64 ビット ARM アーキテクチャー上には **nvme** ドライバーの既存の問題があります。この問題により、最初のカーネルに **irqpoll** カーネルコマンドラインパラメーターを指定すると **vmcore** の生成に失敗します。したがって、カーネルクラッシュ後に **vmcore** が `/var/crash/` ディレクトリーにダンプされません。この問題を回避するには、以下を実行します。

1. `/etc/sysconfig/kdump` ファイルの **KDUMP_COMMANDLINE_REMOVE** キーに **irqpoll** を追加します。

2. **systemctl restart kdump** コマンドを実行して、**kdump** サービスを再起動します。

その結果、最初のカーネルが正常に起動し、カーネルクラッシュ時に **vmcore** がキャプチャーされることが予想されます。

kdump サービスは、大量のクラッシュカーネルメモリーを使用して **vmcore** ファイルをダンプできることに注意してください。キャプチャーカーネルには、**kdump** サービス用のメモリーが十分あることを確認します。

(BZ#1654962)

ダンプターゲットとして vPMEM メモリーを使用すると、カーネルクラッシュキャプチャープロセスが遅延します。

仮想永続メモリー (vPEM) 名前空間を **kdump** または **fadump** ターゲットとして使用する場合には、**papr_scm** モジュールは強制的に、vPMEM がサポートするメモリーのマッピングを解除して再マッピングし、メモリーをリニアマップに再追加します。したがって、この動作により、ハイパーバイザーコール (HCall) が POWER Hypervisor にトリガーされ、カーネルブートのキャプチャーにかかる合計時間が大幅に長くなります。そのため、**kdump** または **fadump** のダンプターゲットとして vPMEM 名前空間を使用しないことが推奨されます。

vPMEM を使用する場合に、この問題を回避するには、以下のコマンドを実行します。

1. **/etc/dracut.conf.d/99-pmem-workaround.conf** ファイルを作成し、以下を追加します。

```
add_drivers+="nd_pmem nd_btt libnvdimm papr_scm"
```

2. 初期 RAM ディスク (initrd) のファイルシステムを再構築します。

```
# touch /etc/kdump.conf
# systemctl restart kdump.service
```

(BZ#1792125)

HP NMI ウォッチドッグが常にクラッシュダンプを生成しない

特定の場合において、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 を要求できません。

欠落している NMI は、以下の 2 つの条件のいずれかによって開始されます。

1. Integrated Lights-Out (iLO) サーバー管理ソフトウェアの **NMI 生成** ボタン。このボタンはユーザーがトリガーします。
2. **hpwdt** ウォッチドッグ。デフォルトでは、有効期限により NMI がサーバーに送信されます。

通常、両方のシーケンスは、システムが応答しない場合に発生します。通常、これらの状況の NMI ハンドラーは **kernel panic()** 関数を呼び出します。また、設定されていれば、**kdump** サービスが **vmcore** ファイルを生成します。

ただし、NMI が見つからないため、**kernel panic()** は呼び出されず、**vmcore** が収集されません。

最初のケース (1.) でシステムが応答しない場合は、その状態のままになります。このシナリオを回避するには、仮想 **電源** ボタンを使用してサーバーをリセットするか、電源を切って入れ直します。

2 つ目のケース (2.) では、欠落している NMI が Automated System Recovery (ASR) からのリセットの後 9 秒後に続きます。

HPE Gen9 Server ラインでは、1 桁台の割合でこの問題が発生します。Gen10 の周波数がさらに小さくなる。

(BZ#1602962)

tuned-adm profile powersave コマンドを使用すると、システムが応答しなくなります。

tuned-adm profile powersave コマンドを実行すると、古い Thunderx (CN88xx) プロセッサを持つ Penguin Valkyrie 2000 2 ソケットシステムが応答しなくなります。これにより、作業を再開するためシステムを再起動することになります。この問題を回避するには、システムが上記の仕様と一致する場合には **powersave** プロファイルの使用を避けてください。

(BZ#1609288)

cxgb4 ドライバーにより kdump カーネルでクラッシュします。

vmcore ファイルに情報を保存しようとする、**kdump** カーネルがクラッシュします。そのため、**cxgb4** ドライバーにより、**kdump** カーネルが、後で分析するためにコアを保存できなくなります。この問題を回避するには、kdump カーネルコマンドラインに **novmcoredd** パラメーターを追加して、コアファイルを保存できるようにします。

(BZ#1708456)

ICE ドライバー NIC ポートをモード 5 (balance-tlb) ボンディングマスターインターフェイスに追加しようとする、と失敗する場合があります。

ICE ドライバー NIC ポートをモード 5 (balance-tlb) ボンディングマスターインターフェイスに追加しようとする、**Master 'bond0', Slave 'ens1f0': Error: Enslave failed** のエラーで失敗する可能性があります。そのため、NIC ポートをボンディングマスターインターフェイスに NICE ポートを追加するときに断続的に問題が発生します。この問題を回避するには、インターフェイスの追加をもう一度試してください。

(BZ#1791664)

type='hostdev' の仮想マシンへの Virtual Function のアタッチに失敗する場合があります。

Assignment with <interface type='hostdev'> メソッドに従って、.XML ファイルを使用して仮想マシンに Virtual Function (VF) をアタッチすると、失敗する場合があります。**Assignment with <interface type='hostdev'>** メソッドを使用すると、仮想マシンに渡す VF NIC に仮想マシンをアタッチできなくなります。この問題を回避するには、**Assignment with <hostdev>** メソッドを使用する、.XML ファイルで、仮想マシンに VF をアタッチしてください。こうすることで、**virsh attach-device** コマンドがエラーなしで成功します。**Assignment with <hostdev>** と **Assignment with <interface type='hostdev'>** (SRIOV デバイスのみ) の相違点については、[PCI Passthrough of host network devices](#) を参照してください。

(BZ#1792691)

5.7.7. ファイルシステムおよびストレージ

/boot ファイルシステムを LVM に配置することができません。

/boot ファイルシステムを LVM 論理ボリュームに配置することはできません。この制限は、以下の理由により存在します。

- EFI システムでは、**EFI システムパーティション**が従来の **/boot** ファイルシステムとして機能します。uEFI 標準では、特定の GPT パーティションタイプと、このパーティションの特定のファイルシステムタイプが必要です。
- RHEL 8 は、システムブートエントリーに **Boot Loader Specification (BLS)** を使用します。この仕様では、プラットフォームのファームウェアが **/boot** ファイルシステムを読み込める必要があります。EFI システムでは、プラットフォームファームウェアは uEFI 標準で定義された **/boot** 設定のみを読み取ることができます。
- GRUB 2 ブートローダーでの LVM 論理ボリュームに対するサポートは完全ではありません。Red Hat は、uEFI や BLS などの標準があるので、この機能のユースケース数が減少しているため、サポートを改善する予定はありません。

Red Hat では、LVM での **/boot** のサポートを提供する予定はありません。代わりに、Red Hat は、**/boot** ファイルシステムを LVM 論理ボリュームに配置する必要がないシステムスナップショットおよびロールバックを管理するツールを提供します。

(BZ#1496229)

LVM で、複数のブロックサイズを持つボリュームグループが作成できない

vgcreate または **vgextend** などの LVM ユーティリティーでは、物理ボリューム (PV) の論理ブロックサイズが異なるボリュームグループ (VG) を作成できなくなりました。別のブロックサイズの PV で基礎となる論理ボリューム (LV) を拡張するとファイルシステムがマウントに失敗するため、LVM はこの変更を採用しました。

ブロックサイズが混在する VG の作成を再度有効にするには、**lvm.conf** ファイルの **allow_mixed_block_sizes=1** オプションを設定します。

(BZ#1768536)

接続されている LUN が過剰にあると、DM Multipath の起動に失敗する可能性があります。

システムに接続されている論理ユニット (LUN) が多過ぎると、**multipathd** サービスがタイムアウトし、起動に失敗する可能性があります。問題を引き起こす LUN の正確な数は、デバイスの数、ストレージレイの応答時間、メモリーおよび CPU の設定、システムの負荷など、複数の要素によって異なります。

この問題を回避するには、**multipathd** ユニットファイルのタイムアウトの値を増やします。

1. ユニットエディターで **multipathd** ユニットを開きます。

```
# systemctl edit multipathd
```

2. 以下の設定を入力し、タイムアウト値を上書きします。

```
[Service]
TimeoutSec=300
```

Red Hat は、デフォルト値の 90 から 300 に値を増やすことを推奨しますが、90 を超える値をテストすることもできます。

3. エディターでファイルを保存します。
4. **systemd** ユニットを再度読み込んで、変更を適用します。

```
# systemctl daemon-reload
```


結果、**multipathd** は、多数の LUN を使用して正常に起動できるようになりました。

(BZ#1797660)

LVM writecache の制限

writecache LVM キャッシュメソッドには以下の制限がありますが、**cache** メソッドには存在しません。

- 論理ボリュームが **writecache** を使用している場合には、論理ボリュームのスナップショットを取得できません。
- 論理ボリュームがアクティブな場合には、**writecache** の割り当てまたは割り当て解除ができません。
- アクティブではない論理ボリュームに **writecache** を割り当てる場合は、既存のファイルシステムのブロックサイズに一致する **writecache** ブロックサイズを使用する必要があります。詳細は、man ページの **lvmscache(7)** を参照してください。
- **writecache** がアタッチされている間は、論理ボリュームのサイズを変更することはできません。
- **writecache** を使用するデバイスでは、**pvmove** コマンドは使用できません。
- **writecache** を指定した論理ボリュームは、シンプールまたは VDO と組み合わせて使用できません。

(JIRA:RHELPLAN-27987, [BZ#1798631](#), BZ#1808012)

LUKS ボリュームを格納する LVM mirror デバイスが応答しなくなることがあります。

セグメントタイプが **mirror** のミラーリング LVM デバイスで LUKS ボリュームを格納すると、特定の条件下で応答しなくなる可能性があります。デバイスが応答しなくなると、すべての I/O 操作を拒否します。

耐障害性のソフトウェア定義ストレージに、LUKS ボリュームをスタックする必要がある場合に、この問題を回避するには、Red Hat はセグメントタイプが **mirror** ではなく **raid1** の LVM RAID 1 デバイスを使用することを推奨します。

raid1 のセグメントタイプは、デフォルトの RAID 設定タイプで、**mirror** の代わりに、推奨のソリューションとしてこのタイプが使用されます。

mirror デバイスを **raid** に変換するには、[ミラーリングされた LVM デバイスの RAID1 論理ボリュームへの変換](#) を参照してください。

(BZ#1730502)

NFS 4.0 パッチにより、オープンな高ワークロードでパフォーマンスが低下する可能性があります。

以前、場合によっては NFS のオープン操作で、サーバー上のファイルが削除されたり、名前が変更されたりするという事実を見落とすというバグが修正されています。ただし、この修正により、多くのオープンな操作が必要とするワークロードのパフォーマンスが遅くなる可能性があります。この問題を回避するには、NFS バージョン 4.1 以降を使用します。これは、多くの場合においてクライアントに委譲を付与するように改善されています。このため、クライアントがローカルに素早く安全にオープン操作を実行できます。

(BZ#1748451)

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

32 ビットアプリケーションで呼び出されると `getpwnam()` が失敗する場合がある

NIS のユーザーが `getpwnam()` 関数を呼び出す 32 ビットアプリケーションを使用する場合は、`nss_nis.i686` パッケージがないと呼び出しに失敗します。この問題を回避するには、`yum install nss_nis.i686` コマンドを使用して、不足しているパッケージを手動でインストールします。

(BZ#1803161)

nginx がハードウェアセキュリティートークンからサーバー証明書をロードできません。

nginx の Web サーバーは、PKCS#11 モジュールを利用してハードウェアセキュリティ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。ただし、現在では、PKCS#11 URI を使用してハードウェアのセキュリティートークンからサーバー証明書を読み込むことはできません。この問題を回避するには、ファイルシステム上にサーバー証明書を保存します。

(BZ#1668717)

php-opcache が PHP 7.2 とともにインストールされると、PHP 7.2 で php-fpm により、SELinux AVC 拒否が発生します。

php-opcache パッケージがインストールされると、FastCGI Process Manager (`php-fpm`) により SELinux AVC 拒否が生じます。この問題を回避するには、`/etc/php.d/10-opcache.ini` ファイルのデフォルト設定を以下のように変更します。

```
opcache.huge_code_pages=0
```

この問題は、`php: 7.3` ではなく、`php:7.2` ストリームにのみ影響することに注意してください。

(BZ#1670386)

mod_wsgi パッケージが依存関係としてインストールされると、このパッケージ名が欠落します。

BZ#1779705 に記載されているように、`mod_wsgi` のインストールが変更され、`python3-mod_wsgi` パッケージにより `mod_wsgi` 名が提供されなくなりました。`mod_wsgi` モジュールのインストール時に、完全なパッケージ名を指定する必要があります。今回の変更により、サードパーティーパッケージの依存関係で問題が発生します。

`mod_wsgi` という名前の依存関係が必要なサードパーティーのパッケージをインストールしようとする、以下のようなエラーが返されます。

```
Error:
Problem: conflicting requests
- nothing provides mod_wsgi needed by package-requires-mod_wsgi.el8.noarch
```

この問題を回避するには、以下のいずれかを実行します。

- a. `python3-mod_wsgi` の完全なパッケージ名が必要なパッケージをリビルドします (または、サードパーティーベンダーに新しいビルドをリクエストしてください)。
- b. `mod_wsgi` のパッケージ名でメタパッケージを作成します。
 1. `mod_wsgi` の名前を渡す空のメタパッケージをご自身で構築します。

2. このメタパッケージを含むリポジトリの `.repo` 設定ファイルに `module_hotfixes=True` の行を追加します。
3. `python3-mod_wsgi` を手作業でインストールします。

(BZ#1829692)

5.7.9. 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ため、1つの実関数呼び出しに対して、複数のプローブを登録します。

この問題を回避するには、SystemTap スクリプトを変更して再帰を検出し、インライン化された部分関数に関連するプローブの配置を防ぎます。

このサンプルスクリプト

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

は、以下のように変更できます。

```
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、真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)

5.7.10. ID 管理

`/etc/nsswitch.conf` を変更するには、手動によるシステムの再起動が必要です。

`authselect select profile_id` コマンドの実行など、`/etc/nsswitch.conf` ファイルを変更した場合は、関連するすべてのプロセスで、更新バージョンの `/etc/nsswitch.conf` ファイルが使用されるように、システムを再起動する必要があります。システムを再起動できない場合は、システムを Active Directory (**System Security Services Daemon** (SSSD) または **winbind**) に追加するサービスを再起動します。

(BZ#1657295)

`files` ドメインが有効である場合は、SSSD が、ローカルユーザーの LDAP グループメンバーシップを誤って返します。

System Security Services Daemon (SSSD) が、ローカルファイルからユーザーを提供し、**sssd.conf** ファイルの [domain/LDAP] セクションの **ldap_rfc2307_fallback_to_local_users** 属性が True に設定されている場合は、ファイルのプロバイダーには他のドメインのグループメンバーシップが含まれません。これにより、ローカルユーザーが LDAP グループのメンバーである場合、**id local_user** コマンドはユーザーの LDAP グループメンバーシップを返しません。この問題を回避するには、暗示の **files** ドメインを無効にするため、次の

```
enable_files_domain=False
```

を **/etc/sss/sss.conf** ファイルの **[sss]** セクションに追加します。

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)

SSSD が同じ優先順位を持つ複数の証明書一致ルールを正しく処理しません。

指定した証明書が、優先順位が同じ複数の証明書の一致ルールに一致する場合、System Security Services Daemon (SSSD) は、いずれか一方のみを使用します。これを回避するには、**|** (or) 演算子で連結した個々のルールのフィルターで設定される LDAP フィルターを持つ 1 つの証明書一致ルールを使用します。証明書一致ルールの例は、man ページの **sss-certamp (5)** を参照してください。

(BZ#1447945)

複数のドメインが定義されている場合は、プライベートグループを **auto_private_group = hybrid** で作成できません。

複数のドメインが定義され、最初のドメイン以外のドメインによってハイブリッドオプションが使用されると、プライベートグループは、**auto_private_group = hybrid** オプションでの作成に失敗します。暗黙的なファイルドメインが **sssd.conf** ファイルの AD または LDAP ドメインとともに定義され、**MPG_HYBRID** としてマークされていない場合、SSSD は **uid=gid** のユーザーに対してプライベートグループを作成し、この **gid** を持つグループは AD または LDAP に存在しません。

sssd_nss レスポンダーは、最初のドメインの **auto_private_groups** オプションの値のみを確認します。これにより、RHEL 8 でデフォルトのセットアップを含む、複数のドメインが設定されているセットアップでは、**auto_private_group** オプションを指定しても効果が得られません。

この問題を回避するには、**sssd.conf** の **sss** セクションで **enable_files_domain = false** を設定します。その結果、**enable_files_domain** オプションが **false** に設定されていると、**sssd** は、アクティブなドメインリストの開始に **id_provider=files** とともにドメインを追加しないため、このバグは発生しません。

(BZ#1754871)

python-ply は FIPS との互換性がありません。

python-ply パッケージの YACC モジュールは、MD5 ハッシュアルゴリズムを使用して YACC 署名のフィンガープリントを生成します。ただし、FIPS モードは、セキュリティ以外のコンテキストでのみ許可される MD5 の使用をブロックします。そのため、**python-ply** は FIPS と互換性がありません。FIPS モードのシステムでは、**ply.yacc()** への呼び出しに失敗し、次のエラーメッセージが表示されます。

```
UnboundLocalError: local variable 'sig' referenced before assignment
```

この問題は **python-pyparser** および **python-cffi** のいくつかのユースケースに影響します。この問題を回避するには、**/usr/lib/python3.6/site-packages/ply/yacc.py** ファイルの 2966 行目を変更し、**sig =**

`md5()` を `sig = md5(usedforsecurity=False)` に置き換えます。これにより、`python-ply` を FIPS モードで使用できます。

(BZ#1747490)

FreeRADIUS が 249 文字を超える Tunnel-Passwords を断りなく切り捨てます。

Tunnel-Password が 249 文字を超える場合、FreeRADIUS サービスはそのパスワードを断りなく切り捨てます。これにより、他のシステムと矛盾する想定外のパスワードになる可能性があります。

この問題を回避するには、249 文字以下のパスワードを選択します。

(BZ#1723362)

すべての KRA メンバーが非表示レプリカの場合は、KRA のインストールに失敗します。

最初の KRA インスタンスが非表示レプリカにインストールされている場合、Key Recovery Authority (KRA) がすでに存在するクラスターでは `ipa-kra-install` ユーティリティで問題が発生します。そのため、これ以上、追加の KRA インスタンスをクラスターに追加することはできません。

この問題を回避するには、新しい KRA インスタンスを追加する前に、KRA ロールが割り当てられた非表示レプリカを解除します。`ipa-kra-install` が正常に終了してから、レプリカを再度非表示にできません。

(BZ#1816784)

Directory Server は、検索フィルターで使用されている属性がスキーマに欠如している場合に警告します。

`nsslapd-verify-filter-schema` パラメーターを `warn-invalid` に設定している場合、Directory Server は、スキーマで定義されていない属性で検索操作を処理し、警告を記録します。この設定では、属性がスキーマに定義されているかどうかにかかわらず、Directory Server は検索結果で要求された属性を返します。

Directory Server の今後のバージョンでは、`nsslapd-verify-filter-schema` のデフォルト設定が厳格なチェックを実施するように変更されます。新しいデフォルトでは、スキーマで欠如している属性について警告し、リクエストを拒否するか、部分的な結果のみを返します。

(BZ#1790259)

ipa-healthcheck-0.4 は古いバージョンの ipa-healthcheck を廃止しません

Healthcheck ツールは、`ipa-healthcheck` と `ipa-healthcheck-core` 2つのサブパッケージに分割されました。ただし、`ipa-healthcheck-core` サブパッケージのみが古いバージョンの `ipa-healthcheck` に正しく設定されています。その結果、Healthcheck を更新すると `ipa-healthcheck-core` のみがインストールされ、更新後に `ipa-healthcheck` コマンドが機能しなくなります。

この問題を回避するには、`yum install ipa-healthcheck-0.4` を使用して `ipa-healthcheck-0.4` サブパッケージを手動でインストールします。

(BZ#1852244)

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)

5.7.11. デスクトップ

Wayland セッションの制限

Red Hat Enterprise Linux 8 では、GNOME 環境および GNOME Display Manager (GDM) のデフォルトのセッションタイプとして、以前の RHEL メジャーバージョンで使用されていた X11 セッションに代わり、**Wayland** が使用されます。

以下の機能は、**Wayland** の下では現在利用できないか、期待どおりに動作しません。

- **xrandr** などの X11 設定ユーティリティーは、ハンドリング、解像度、ローテーション、レイアウトのアプローチが異なるため、**Wayland** では動作しません。ディスプレイ機能は、GNOME 設定を使用して設定できます。
- 画面の録画とリモートデスクトップでは、アプリケーションが **Wayland** のポータル API をサポートする必要があります。特定のレガシーアプリケーションはポータル API をサポートしません。
- **Wayland** では、ポインターのアクセシビリティは利用できません。
- クリップボードマネージャーは利用できません。
- **Wayland** 上の GNOME Shell は、ほとんどのレガシー X11 アプリケーションで発行されたキーボードグラブを無視します。**/org/gnome/mutter/wayland/xwayland-grab-access-rules** GSettings キーを使用して、X11 アプリケーションがキーボードグラブを発行するようにできます。デフォルトでは、**Wayland** 上の GNOME Shell により、以下のアプリケーションがキーボードグラブを発行できます。
 - **GNOME Boxes**
 - **Vinagre**
 - **Xephyr**
 - **virt-manager**、**virt-viewer**、および **remote-viewer**
 - **vncviewer**
- ゲスト仮想マシン (VM) 内の **Wayland** には安定性とパフォーマンスの問題があります。RHEL は、仮想マシンで実行している場合に X11 セッションに自動的にフォールバックします。

X11 GNOME セッションを使用していた RHEL 7 システムから RHEL 8 にアップグレードすると、システムでは引き続き X11 が使用されます。また、次のグラフィックドライバーが使用されている場合は、自動的に X11 にフォールバックします。

- プロプライエタリー NVIDIA ドライバー

- **cirrus** ドライバー
- **mga** ドライバー
- **aspeed** ドライバー

Wayland の使用は手動で無効にできます。

- GDM の Wayland を無効にするには、`/etc/gdm/custom.conf` ファイルに **WaylandEnable=false** オプションを設定します。
- GNOME セッションで Wayland を無効にするには、ログイン名を入力してから、ログイン画面の歯車メニューでレガシーの X11 オプションを選択します。

Wayland の詳細は <https://wayland.freedesktop.org/> を参照してください。

(BZ#1797409)

ドラッグアンドドロップが、デスクトップとアプリケーション間で機能しません。

gnome-shell-extensions パッケージのバグにより、ドラッグアンドドロップ機能は現在、デスクトップとアプリケーションの間では機能しません。この機能のサポートは、今後のリリースで追加される予定です。

(BZ#1717947)

ソフトウェアリポジトリからの flatpak リポジトリの無効化ができません。

現時点で、GNOME Software ユーティリティの Software Repositories ツールで **flatpak** リポジトリを無効化または削除することはできません。

(BZ#1668760)

Generation 2 の RHEL 8 仮想マシンが Hyper-V Server 2016 ホストで起動できない場合があります。

Microsoft Hyper-V Server 2016 ホストで実行している仮想マシンで RHEL 8 をゲストオペレーティングシステムとして使用すると、仮想マシンが起動しなくなり、GRUB ブートメニューに戻る場合があります。さらに、以下のエラーが Hyper-V イベントログに記録されます。

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

このエラーは、Hyper-V ホストの UEFI ファームウェアバグが原因で発生します。この問題を回避するには、Hyper-V Server 2019 をホストとして使用します。

(BZ#1583445)

システムクラッシュにより、fadump 設定が失われることがあります。

この問題は、ファームウェア支援ダンプ (fadump) が有効になっているシステムで確認されています。また、ブートパーティションは XFS などのジャーナリングファイルシステムにあります。システムクラッシュにより、ダンプサポートが有効でない古い **initrd** をブートローダーがロードする可能性があります。したがって、復元後、システムは **vmcore** ファイルをキャプチャーせず、fadump 設定が失われる可能性があります。

この問題を回避するには、以下を実行します。

- `/boot` が別のパーティションの場合は、以下を実行します。

1. `kdump` サービスを再起動します。
2. `root` ユーザーで以下のコマンドを実行するか、`CAP_SYS_ADMIN` 権限を持つユーザーアカウントを使用します。

```
# fsfreeze -f
# fsfreeze -u
```

- `/boot` が別のパーティションではない場合は、システムを再起動します。

(BZ#1723501)

5.7.12. グラフィックインフラストラクチャー

sudo コマンドを使用してグラフィカルアプリケーションを実行できません。

権限が昇格されたユーザーで、グラフィカルアプリケーションを実行しようとする、エラーメッセージが表示され、アプリケーションを開くことができません。この障害は、**Xauthority** ファイルで、通常ユーザーの認証情報を使用して認証するように、**Xwayland** に制限が加えられているため発生します。

この問題を回避するには、**sudo -E** コマンドを使用して、`root` ユーザーとしてグラフィカルアプリケーションを実行します。

(BZ#1673073)

radeon がハードウェアを適切なハードウェアリセットに失敗します。

現在、**radeon** カーネルドライバーは、`kexec` コンテキストでハードウェアを正しくリセットしません。代わりに **radeon** がフェイルオーバーします。これにより、`kdump` サービスの残りの部分が失敗します。

この問題を回避するには、`/etc/kdump.conf` ファイルに以下の行を追加して、`kdump` で **radeon** をブラックリストに指定します。

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

マシンと `kdump` を再起動します。`kdump` の起動後、設定ファイルから **force_rebuild 1** 行が削除される可能性があります。

このシナリオでは、`kdump` 中にグラフィックは利用できませんが、`kdump` は正常に動作します。

(BZ#1694705)

1つの MST トポロジーで複数の HDR ディスプレイを使用すると、電源が入らないことがあります。

nouveau ドライバーの NVIDIA Turing GPUs を使用するシステムで、**DisplayPort** ハブ (ラップトップのドックなど) を使用して HDR プラグインのサポートがあるモニターを複数接続すると、以前の RHEL リリースではできていたにも拘らず、すべてのディスプレイの電源が入らないことがあります。これは、全ディスプレイをサポートする帯域幅がハブ上にないと、システムが誤って判断してしまうことが原因で発生します。

(BZ#1812577)

5.7.13. Web コンソール

非特権ユーザーがサブスクリプションページにアクセスできます。

管理者以外のユーザーが Web コンソールのサブスクリプションページに移動すると、Web コンソールでは、**Cockpit had an unexpected internal error** (Cockpit に予期しない内部エラーが発生しました) という一般的なエラーメッセージが表示されます。

この問題を回避するには、権限のあるユーザーで Web コンソールにサインインし、**Reuse my password for privileged tasks** チェックボックスをチェックします。

(BZ#1674337)

5.7.14. 仮想化

Windows Server 2019 ホストの RHEL 8 仮想マシンで GUI ディスプレイのパフォーマンスが下がります。

Windows Server 2019 ホストのグラフィカルモードで RHEL 8 をゲストオペレーティングシステムとして使用すると、GUI ディスプレイのパフォーマンスが下がり、ゲストのコンソール出力に現在必要なよりも長い時間がかかります。

これは、Windows 2019 ホストで既知の問題で、Microsoft が修正を保留しています。この問題を回避するには、SSH を使用してゲストに接続するか、ホストとして Windows Server 2016 を使用します。

(BZ#1706541)

QXL で、Wayland を使用する仮想マシンの複数のモニターを表示できません。

remote-viewer ユーティリティを使用して、Wayland ディスプレイサーバーを使用している仮想マシンのモニターを複数表示すると、仮想マシンが応答なくなり、**Waiting for display** というステータスメッセージが永久に表示されます。

この問題を回避するには、Wayland を使用する仮想マシンの GPU デバイスとして **qxl** の代わりに **virtio-gpu** を使用します。

(BZ#1642887)

virsh iface-* コマンドが一貫して動作しません。

現在、**virsh iface-*** コマンド (**virsh iface-start**、**virsh iface-destroy** など) は、設定の依存関係が原因で頻繁に失敗します。したがって、ホストネットワーク接続の設定および管理には **virsh iface-*** コマンドを使用しないことが推奨されます。代わりに、NetworkManager プログラムとその関連管理アプリケーションを使用します。

(BZ#1664592)

RHEL 8 仮想マシンが、Witherspoon ホストで起動できないことがあります。

RHEL 8 仮想マシン (VM) で **pseries -rhel7.6.0-sxxm** マシンタイプを使用すると、DD 2.2 または DD2.3 CPU を使用する **Power9 S922LC for HPC** ホストで起動できない場合があります。

代わりに仮想マシンを起動しようとする、以下のエラーメッセージが出力されます。

```
qemu-kvm: Requested safe indirect branch capability level not supported by kvm
```

この問題を回避するには、仮想マシンの XML 設定を次のように設定します。

-

```
<domain type='qemu' xmlns:qemu='http://libvirt.org/schemas/domain/qemu/1.0'>
  <qemu:commandline>
    <qemu:arg value='-machine'/>
    <qemu:arg value='cap-ibs=workaround'/>
  </qemu:commandline>
```

(BZ#1732726, BZ#1751054)

IBM POWER 仮想マシンが、空の NUMA ノードで正常に動作しません。

現在、RHEL 8 ホストで実行している IBM POWER 仮想マシン (VM) が、メモリーを使用せず (**memory='0'**)、CPU も使用しない NUMA ノードで設定されていると、仮想マシンが起動しません。したがって、Red Hat は、RHEL 8 では、そういった空の NUMA ノードを持つ IBM POWER 仮想マシンを使用しないことを強く推奨しています。

(BZ#1651474)

AMD EPYC でホストパススルーモードを使用する際に、SMT CPU トポロジーが仮想マシンで検出されません。

AMD EPYC ホストで行われた CPU ホストパススルーモードで仮想マシンを起動すると、**TOPOEXT** 機能フラグは存在しません。したがって、仮想マシンは、コアごとに複数のスレッドを持つ仮想 CPU トポロジーを検出できません。この問題を回避するには、ホストパススルーの代わりに EPYC CPU モデルを使用して仮想マシンを起動します。

(BZ#1740002)

RHEL 8.2 仮想マシンのディスク識別子は、仮想マシンの再起動で変わる可能性があります。

Hyper-V ハイパーバイザーのゲストオペレーティングシステムとして RHEL 8.2 で仮想マシン (VM) を使用する場合は、仮想マシンの再起動時に仮想マシンの仮想ディスクのデバイス識別子が変わることがあります。たとえば、最初は **/dev/sda** として識別されていたディスクが、**dev/sdb** になる場合があります。これにより、仮想マシンが起動に失敗し、仮想マシンのディスクを参照するスクリプトが動作しなくなる可能性があります。

この問題を回避するには、Red Hat は、仮想マシンのディスクに永続的な名前を設定することを強く推奨します。詳細は、Microsoft Azure のドキュメント (<https://docs.microsoft.com/en-us/azure/virtual-machines/troubleshooting/troubleshoot-device-names-problems>) を参照してください。

(BZ#1777283)

多数の virtio-blk ディスクを使用すると、仮想マシンが起動しないことがあります。

多数の virtio-blk デバイスを仮想マシンに追加すると、プラットフォームで利用可能な割り込みベクトルの数が使い切られる可能性があります。これが発生すると、仮想マシンのゲスト OS は起動できず、**dracut-initqueue[392]: Warning: Could not boot** エラーが表示されます。

(BZ#1719687)

virtio-blk を使用して仮想マシンに LUN デバイスを割り当てると機能しません。

q35 マシンタイプは、移行用の virtio 1.0 デバイスをサポートしないため、RHEL 8 では virtio 1.0 で非推奨となった機能はサポートされません。特に、RHEL 8 ホストで virtio-blk デバイスから SCSI コマンドを送信することはできません。したがって、virtio-blk コントローラーを使用する場合は、物理ディスクを LUN デバイスとして仮想マシンに割り当てると失敗します。

物理ディスクをゲストオペレーティングシステムを通して渡すことは引き続き可能ですが、**device='lun'** オプションではなく、**device='disk'** オプションで設定する必要があることに留意してください。

(BZ#1777138)

RHEL 7-ALT ホストから RHEL 8 への POWER9 ゲストの移行に失敗する

現在のリリースでは、RHEL 7-ALT ホストシステムから RHEL 8 に POWER9 仮想マシンを移行すると、Migration status: active のステータスで応答がなくなります。

この問題を回避するには、RHEL 7-ALT ホストで Transparent Huge Pages (THP) を無効にすることで、移行が正常に完了します。

(BZ#1741436)

5.7.15. サポート関連

redhat-support-tool が FUTURE 暗号化ポリシーを使用すると機能しません。

カスタマーポータル API の証明書が使用する暗号化キーは **FUTURE** のシステム全体の暗号化ポリシーが定義する要件を満たさないため、現時点で **redhat-support-tool** ユーティリティは、このポリシーレベルでは機能しません。

この問題を回避するには、カスタマーポータル API への接続中に **DEFAULT** 暗号化ポリシーを使用します。

(BZ#1802026)

5.7.16. コンテナ

UDICA が 1.0 の安定したストリームと連携するように想定されていません。

UDICA (コンテナの SELinux ポリシーを生成するツール) は、**container-tools:1.0** モジュールストリームの podman 1.0.x で実行されるコンテナで機能するように想定されていません。

(JIRA:RHELPLAN-25571)

Podman での FIPS サポートに関する注意事項

Federal Information Processing Standard (FIPS) を使用するには、認定済みモジュールを使用する必要があります。以前のバージョンでは、Podman は起動時に適切なフラグを有効にして、コンテナに認定モジュールを正しくインストールしていました。ただし、本リリースでは、Podman は FIPS システム全体の crypto-policy の形式でシステムによって提供される追加アプリケーションヘルパーを適切に設定しません。認定モジュールでは、システム全体の crypto-policy を設定する必要はありませんが、適切な方法で暗号化モジュールを使用するアプリケーションが強化されます。この問題を回避するには、他のアプリケーションコードを実行する前に、コンテナを変更して **update-crypto-policies --set FIPS** コマンドを実行します。

(BZ#1804193)

第6章 国際化

6.1. 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

6.2. RHEL 8 における国際化の主な変更点

RHEL 8 では、RHEL 7 の国際化に以下の変更が加えられています。

- Unicode 11 コンピューティングの業界標準のサポートが追加されました。

- 国際化は複数のパッケージで配布され、より小さなフットプリントのインストールを可能にします。詳細は、[Using langpacks](#) を参照してください。
- 複数のロケールの **glibc** パッケージの更新が、Common Locale Data Repository (CLDR) と同期するようになりました。

付録A コンポーネント別のチケットリスト

本書には Bugzilla と JIRA ID が記載されています。一般にアクセス可能な Bugzilla バグには、チケットへのリンクが含まれます。

コンポーネント	チケット
389-ds-base	BZ#1715406 , BZ#1748016 , BZ#1790259 , BZ#1748994 , BZ#1739718
NetworkManager	BZ#1626348
anaconda	BZ#1747382 , BZ#1637472 , BZ#1748756 , BZ#1649359 , BZ#1715303 , BZ#1696609 , BZ#1672405 , BZ#1687747 , BZ#1745064 , BZ#1659400 , BZ#1821192 , BZ#1822880 , BZ#1823578 , BZ#1748281 , BZ#1746391
audit	BZ#1757986
authselect	BZ#1657295
bind	BZ#1564443 , BZ#1664863 , BZ#1704328
binutils	BZ#1777002 , BZ#1618748
buildah-container	BZ#1627898
clevis	BZ#1766526 , BZ#1564559 , BZ#1436780 , BZ#1784524
cloud-init	BZ#1641190 , BZ#1666961
cockpit-appstream	BZ#1676506
cockpit	BZ#1678465 , BZ#1754163 , BZ#1666722
container-tools-rhel8-module	BZ#1784267
corosync-qdevice	BZ#1784200
createrepo_c	BZ#1743186
crypto-policies	BZ#1690565 , BZ#1660839
device-mapper-multipath	BZ#1797660
dhcp	BZ#1729211
distribution	BZ#1657927

コンポーネント	チケット
dnf	BZ#1676891 , BZ#1754609
dnsmasq	BZ#1700916
edk2	BZ#1748180
elfutils	BZ#1744992
fapolicyd	BZ#1759895
fence-agents	BZ#1775847
firewalld	BZ#1737045 , BZ#1740670 , BZ#1733066
freeradius	BZ#1723362
gcc-toolset-9	BZ#1774118
gcc	BZ#1726641 , BZ#1698607 , BZ#1747157
gdb	BZ#1768593
gdm	BZ#1749960
glibc	BZ#1410154 , BZ#1764214 , BZ#1749439 , BZ#1764235 , BZ#1746928 , BZ#1777241 , BZ#1361965 , BZ#1747502 , BZ#1764218 , BZ#1764238 , BZ#1746933 , BZ#1747453
gnome-shell-extensions	BZ#1717947
gnome-shell	BZ#1724302
gnome-software	BZ#1668760
gnutls	BZ#1628553 , BZ#1677754
go-toolset	BZ#1747150
grafana-pcp	BZ#1685315
grafana	BZ#1725278
graphviz	BZ#1704875
grub2	BZ#1583445 , BZ#1723501

コンポーネント	チケット
---------	------

httpd-2.4-module	BZ#1747923
httpd	BZ#1633224
initial-setup	BZ#1676439
ipa	BZ#1665051 , BZ#1816784 , BZ#1719767 , BZ#1777564 , BZ#1664719 , BZ#1664718
ipcalc	BZ#1638834
java-11-openjdk	BZ#1746875
kernel-rt	BZ#1680161
kernel	BZ#1744397, BZ#1698297, BZ#1687094, BZ#1720227, BZ#1846345, BZ#1635295, BZ#1793389, BZ#1706541, BZ#1666538, BZ#1602962, BZ#1649647, BZ#1153521, BZ#1694705, BZ#1348508, BZ#1748451, BZ#1708456, BZ#1654962, BZ#1609288, BZ#1777283, BZ#1791664, BZ#1792125, BZ#1792691, BZ#1812666, BZ#1812577, BZ#1757933, BZ#1763661, BZ#1780432, BZ#1401552, BZ#1716002, BZ#1593711, BZ#1620349, BZ#1724969, BZ#1714330, BZ#1714486, BZ#1660368, BZ#1524687, BZ#1274406, BZ#1650518, BZ#1636572, BZ#1727369, BZ#1519039, BZ#1627455, BZ#1501618, BZ#1495358, BZ#1633143, BZ#1503672, BZ#1570255, BZ#1696451, BZ#1665295, BZ#1658840, BZ#1660627, BZ#1569610, BZ#1730502
kexec-tools	BZ#1750278, BZ#1690729
kmod-kvdo	BZ#1737639 , BZ#1657301
krb5	BZ#1754690
libbpf	BZ#1759154
libdnf	BZ#1697472
libgnome-keyring	BZ#1607766
libndp	BZ#1697595
libpfm	BZ#1731019

コンポーネント	チケット
libreswan	BZ#1777474
libselinux-python-2.8-module	BZ#1666328
libvirt	BZ#1749672 , BZ#1664592, BZ#1528684
llvm-toolset	BZ#1747139
lorax	BZ#1754711
ltrace	BZ#1655368
lvm2	BZ#1600174, BZ#1496229, BZ#1768536
make	BZ#1774790
maven	BZ#1783926
mod_wsgi	BZ#1829692 , BZ#1779705
mutter	BZ#1737553
nfs-utils	BZ#1719983 , BZ#1592011
nftables	BZ#1778883 , BZ#1643192
nginx	BZ#1668717
nmstate	BZ#1674456
nss_nis	BZ#1803161
nss	BZ#1724250 , BZ#1817533 , BZ#1645153
numactl	BZ#1730738
opencv	BZ#1694647
openscap	BZ#1636431, BZ#1618489, BZ#1646197, BZ#1803116 , BZ#1795563 , BZ#1824152 , BZ#1642373
openssh	BZ#1744108

コンポーネント	チケット
openssl-pkcs11	BZ#1705505 , BZ#1664807 , BZ#1745082
openssl	BZ#1685470 , BZ#1749068
oscap-anaconda-addon	BZ#1665082 , BZ#1674001 , BZ#1691305 , BZ#1787156 , BZ#1816199
pacemaker	BZ#1712584 , BZ#1700104
pam	BZ#1252859 , BZ#1537242
pcp	BZ#1723598
pcs	BZ#1631519 , BZ#1631514 , BZ#1676431 , BZ#1442116 , BZ#1619620
perl-LDAP	BZ#1663063
php-7.2-module	BZ#1670386
php-pecl-xdebug	BZ#1769857
pki-core	BZ#1698084 , BZ#1303254
podman	BZ#1804193 , BZ#1645280
policycoreutils	BZ#1563742 , BZ#1417455
postfix	BZ#1723950 , BZ#1745321
powertop	BZ#1716721
pykickstart	BZ#1637872
python-ply	BZ#1747490
python38-3.8-module	BZ#1747329
qemu-kvm	BZ#1651474 , BZ#1740002 , BZ#1719687 , BZ#1651994 , BZ#1741346
rear	BZ#1729501
redhat-release	BZ#1817591
redhat-support-tool	BZ#1802026
rhel-system-roles-sap	BZ#1660832

コンポーネント	チケット
rng-tools	BZ#1692435
rpm	BZ#1688849
rsyslog	JIRA:RHELPLAN-10431, BZ#1659383 , BZ#1679512 , BZ#1740683 , BZ#1676559 , BZ#1692073 , BZ#1692072
rust-toolset	BZ#1776847
s390utils	BZ#1750326
samba	BZ#1754409 , JIRA:RHELPLAN-13195
scap-security-guide	BZ#1755447 , BZ#1754919 , BZ#1750755 , BZ#1755194
scap-workbench	BZ#1640715
selinux-policy	BZ#1641631 , BZ#1746398 , BZ#1826788 , BZ#1727887 , BZ#1726166 , BZ#1726246
setools	BZ#1731519
setroubleshoot-plugins	BZ#1649842
setup	BZ#1730396 , BZ#1663556
skopeo-container	BZ#1627900
skopeo	BZ#1810053
sscg	BZ#1717880
sssd	BZ#1669407 , BZ#1652562 , BZ#1447945 , BZ#1754871
subscription-manager	BZ#1674337
sudo	BZ#1786990 , BZ#1733961
systemd	BZ#1686892 , BZ#1640802
systemtap	BZ#1744989
tpm2-tools	BZ#1725714
tuned	BZ#1738250

コンポーネント	チケット
udica	BZ#1763210 , BZ#1732704
vdo	BZ#1713749
virt-manager	BZ#1677019
wayland	BZ#1673073
whois	BZ#1734183
xorg-x11-drv-qxl	BZ#1642887
xorg-x11-server	BZ#1698565
zlib	BZ#1659433 , BZ#1666798
その他	BZ#1640697 , BZ#1659609 , BZ#1687900 , BZ#1697896 , BZ#1797409 , BZ#1790635 , BZ#1823398 , BZ#1745507 , BZ#1732726 , BZ#1757877 , JIRA:RHELPLAN-25571 , BZ#1777138 , JIRA:RHELPLAN-27987 , BZ#1797671 , BZ#1780124 , JIRA:RHELPLAN-2507 , JIRA:RHELPLAN-37713 , JIRA:RHELPLAN-37777 , BZ#1841170 , JIRA:RHELPLAN-13995 , BZ#1785248 , BZ#1755347 , BZ#1784455 , BZ#1784456 , BZ#1789401 , JIRA:RHELPLAN-41384 , BZ#1690207 , JIRA:RHELPLAN-1212 , BZ#1559616 , BZ#1812552 , JIRA:RHELPLAN-14047 , BZ#1769727 , JIRA:RHELPLAN-27394 , BZ#1642765 , JIRA:RHELPLAN-10304 , BZ#1646541 , BZ#1647725 , BZ#1686057 , BZ#1748980 , BZ#1827628

付録B 改訂履歴

0.3-5

2023年12月7日木曜日、Lucie Vařáková (lvarakova@redhat.com)

- 非推奨の機能 [JIRA:RHELDOCS-16612](#) (Identity Management)を追加しました。

0.3-4

2023年11月10日金曜日、Gabriela Fialová (gfialova@redhat.com)

- RHEL ドキュメントへのフィードバックの提供に関するモジュールを更新しました。

0.3-3

2022年11月15日(火) Gabriela Fialová (gfialova@redhat.com)

- 壊れたリンクを修正しました。

0.3-2

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。

0.3-1

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

0.3-0

2023年4月13日(木) Gabriela Fialová (gfialova@redhat.com)

- [BZ#1811997](#) を更新し、Rust Toolset スプラッシュページへのリンクを追加しました。

0.2-9

2023年3月1日、水曜日、Lucie Vařáková (lvarakova@redhat.com)

- バグ修正 [BZ#2151696](#) (アイデンティティ管理)を追加しました。

0.2-8

2022年4月29日、金曜日、Lenka Špačková (lspackova@redhat.com)

- 非推奨になった機能の概要を更新。
- [BZ#1605216](#) のタイプミスを修正。
- 無効になっていたリンクを修正。

0.2-7

2021年12月23日(木) Lenka Špačková (lspackova@redhat.com)

- Soft-RoCE ドライバー `rdma_rxe` に関する情報が、テクノロジープレビューの [BZ#1605216](#) および非推奨の機能 [BZ#1878207](#) (カーネル) に追加されました。

0.2-6

2021年10月5日(火) Lucie Maňásková (Imanasko@redhat.com)

- 非推奨の機能 [BZ#1999620](#) (シェルおよびコマンドラインツール) が追加されました。

0.2-5

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。

0.2-4

2021年7月9日(金) Lucie Maňásková (Imanasko@redhat.com)

- [JIRA:RHELPLAN-34199](#) (セキュリティー) を更新しました。

0.2-3

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。

0.2-2

2021年5月21日(金) Lenka Špačková (lspackova@redhat.com)

- [概要](#) で OS 変換に関する情報を更新。

0.2-1

2021年4月6日(火) Lenka Špačková (lspackova@redhat.com)

- 対応しているアーキテクチャーのリストを改善

0.2-0

2021年3月11日(木) Lucie Maňásková Lucie Maňásková (Imanasko@redhat.com)

- 新機能セクション (ファイルシステムとストレージ) を更新しました。

0.1-9

2021年2月25日(木) Lenka Špačková (lspackova@redhat.com)

- CentOS Linux 名を修正。

0.1-8

2021年2月10日(水) Lucie Maňásková (Imanasko@redhat.com)

- 既知の問題 (仮想化) を追加。

0.1-7

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。

0.1-6

2020年12月10日(木) Lenka Špačková (lspackova@redhat.com)

- SSSDでのAD GPOの処理に関する情報が新機能 (Identity Management) に追加されました。

0.1-5

2020年12月1日(火) Lucie Maňásková (lmanasko@redhat.com)

- fapolicyd (セキュリティー) の問題に対するバグ修正が追加されました。
- 既知の問題 (インストーラー) を追加しました。

0.1-4

2020年11月24日(火) Lucie Maňásková (lmanasko@redhat.com)

- 新機能セクション (ネットワーク) を更新しました。

0.1-3

2020年10月30日(金) Lenka Špačková (lspackova@redhat.com)

- リポジトリセクションのアプリケーションストリームの説明を更新。

0.1-2

2020年10月5日(月) Lucie Maňásková (lmanasko@redhat.com)

- バグ修正 (ネットワーク) を追加しました。

0.1-1

2020年9月29日(火) Lenka Špačková (lspackova@redhat.com)

- RHEL 7.9 のリリースでインプレースアップグレードパスを更新しました。

0.1-0

2020年8月27日(木) Lucie Maňásková (lmanasko@redhat.com)

- バグ修正 (カーネル) を追加しました。

0.0-9

2020年8月10日(月) Lucie Maňásková (lmanasko@redhat.com)

- 既知の問題 (ID 管理) を追加しました。

0.0-8

2020年7月21日(火) Lucie Maňásková (lmanasko@redhat.com)

- Red Hat Enterprise Linux 8.2.1 リリースノートも併せて参照してください。

0.0-7

2020年7月16日(木) Lucie Maňásková (lmanasko@redhat.com)

- テクノロジープレビュー (ネットワーク) を追加。
- 新機能セクションを更新。

0.0-6

2020 年 6 月 25 日 (木) Jaroslav Klech (jklech@redhat.com)

- カーネルパラメーターの章を詳細化。
- デバイスドライバーの章にさまざまな改善を追加。

0.0-5

2020 年 6 月 19 日 (金) Lucie Maňásková (Imanasko@redhat.com)

- 新しい既知の問題を追加。
- 他のリリースノートへの複数の更新。

0.0-4

2020 年 6 月 4 日 (木) Lucie Maňásková (Imanasko@redhat.com)

- 新機能セクションを更新。
- 既知の問題 (コンテナー) を追加。

0.0-3

2020 年 5 月 20 日 (水) Lenka Špačková (lspackova@redhat.com)

- 既知の問題 (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) を追加。
- バグ修正 (コンパイラーおよび開発ツール) を追加。
- 他のリリースノートへの複数の更新。

0.0-2

Tue Apr 28 2020, Lucie Maňásková (Imanasko@redhat.com)

- Red Hat Enterprise Linux 8.2 リリースノートも併せて参照してください。

0.0-1

Mon Mar 09 2020, Jaroslav Klech (jklech@redhat.com)

- 外部カーネルパラメーターおよび新しいドライバーの章に対する重要な変更を提示しました。

0.0-0

Tue Jan 21 2020, Lucie Maňásková (Imanasko@redhat.com)

- Red Hat Enterprise Linux 8.2 Beta リリースノートをリリースしました。