



Red Hat

Red Hat Enterprise Linux 8

8.5 リリースノート

Red Hat Enterprise Linux 8.5 リリースノート

Red Hat Enterprise Linux 8 8.5 リリースノート

Red Hat Enterprise Linux 8.5 リリースノート

Legal Notice

Copyright © 2025 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.

Abstract

本リリースノートでは、Red Hat Enterprise Linux 8.5 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。

Table of Contents

RED HAT ドキュメントへのフィードバック (英語のみ)	5
第1章 概要	6
1.1. RHEL 8.5 における主な変更点	6
1.2. インプレースアップグレードおよび OS 移行	8
1.3. RED HAT CUSTOMER PORTAL LABS	9
1.4. 関連情報	10
第2章 アーキテクチャー	11
第3章 RHEL 8 のコンテンツの配布	12
3.1. インストール	12
3.2. リポジトリ	12
3.3. アプリケーションストリーム	13
3.4. YUM/DNF を使用したパッケージ管理	13
第4章 新機能	14
4.1. インストーラーおよびイメージの作成	14
4.2. RHEL FOR EDGE	15
4.3. ソフトウェア管理	15
4.4. シェルおよびコマンドラインツール	16
4.5. インフラストラクチャーサービス	19
4.6. SECURITY	20
4.7. ネットワーキング	27
4.8. カーネル	29
4.9. ファイルシステムおよびストレージ	32
4.10. 高可用性およびクラスター	33
4.11. 動的プログラミング言語、WEB サーバー、およびデータベースサーバー	34
4.12. コンパイラーおよび開発ツール	39
4.13. IDENTITY MANAGEMENT	51
4.14. デスクトップ	56
4.15. グラフィックインフラストラクチャー	60
4.16. RED HAT ENTERPRISE LINUX システムロール	62
4.17. 仮想化	65
4.18. サポート性	65
4.19. コンテナー	66
第5章 外部カーネルパラメーターへの重要な変更	68
新しいカーネルパラメーター	68
更新されたカーネルパラメーター	72
第6章 デバイスドライバー	75
6.1. 新しいドライバー	75
6.2. 更新されたドライバー	76
第7章 バグ修正	77
7.1. インストーラーおよびイメージの作成	77
7.2. シェルおよびコマンドラインツール	77
7.3. インフラストラクチャーサービス	80
7.4. SECURITY	80
7.5. カーネル	82
7.6. ファイルシステムおよびストレージ	84
7.7. 高可用性およびクラスター	84

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	84
7.9. コンパイラーおよび開発ツール	85
7.10. IDENTITY MANAGEMENT	86
7.11. RED HAT ENTERPRISE LINUX システムロール	87
7.12. クラウド環境の RHEL	90
7.13. コンテナー	91
第8章 テクノロジープレビュー	92
8.1. シェルおよびコマンドラインツール	92
8.2. ネットワーキング	92
8.3. カーネル	94
8.4. ファイルシステムおよびストレージ	95
8.5. 高可用性およびクラスター	98
8.6. IDENTITY MANAGEMENT	99
8.7. デスクトップ	101
8.8. グラフィックインフラストラクチャー	101
8.9. RED HAT ENTERPRISE LINUX システムロール	101
8.10. 仮想化	102
8.11. コンテナー	103
第9章 非推奨の機能	105
9.1. インストーラーおよびイメージの作成	105
9.2. ソフトウェア管理	106
9.3. シェルおよびコマンドラインツール	106
9.4. SECURITY	107
9.5. ネットワーキング	109
9.6. カーネル	110
9.7. ファイルシステムおよびストレージ	110
9.8. 高可用性およびクラスター	112
9.9. コンパイラーおよび開発ツール	113
9.10. IDENTITY MANAGEMENT	113
9.11. デスクトップ	116
9.12. グラフィックインフラストラクチャー	116
9.13. WEB コンソール	117
9.14. RED HAT ENTERPRISE LINUX システムロール	117
9.15. 仮想化	117
9.16. サポート性	118
9.17. コンテナー	119
9.18. 非推奨のパッケージ	119
9.19. 非推奨のデバイスおよび非保守のデバイス	144
第10章 既知の問題	147
10.1. インストーラーおよびイメージの作成	147
10.2. サブスクリプションの管理	150
10.3. ソフトウェア管理	150
10.4. シェルおよびコマンドラインツール	151
10.5. インフラストラクチャーサービス	151
10.6. SECURITY	151
10.7. ネットワーク	156
10.8. カーネル	157
10.9. ハードウェアの有効化	162
10.10. ファイルシステムおよびストレージ	162
10.11. 動的プログラミング言語、WEB サーバー、およびデータベースサーバー	164
10.12. コンパイラーおよび開発ツール	164

10.13. IDENTITY MANAGEMENT	165
10.14. デスクトップ	168
10.15. グラフィックインフラストラクチャー	169
10.16. 仮想化	170
10.17. クラウド環境の RHEL	173
10.18. サポート関連	175
10.19. コンテナー	175
第11章 国際化	178
11.1. RED HAT ENTERPRISE LINUX 8 の多言語	178
11.2. RHEL 8 における国際化の主な変更点	178
付録A コンポーネント別のチケットリスト	180
付録B 改訂履歴	188

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。以下の方法で送信してください。

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

1.1. RHEL 8.5 における主な変更点

インストーラーおよびイメージの作成

RHEL 8.5 では、Image Builder が以下の機能をサポートしています。

- ファイルシステムの設定をカスタマイズする機能
- 利用可能な正式なリポジトリを上書きする機能
- ブート可能なインストライメージを作成し、ベアメタルシステムにインストールする機能

詳しくは、[「インストーラーおよびイメージの作成」](#)をご覧ください。

RHEL for Edge

RHEL 8.5 には、デバイスへの無人インストールと RHEL for Edge イメージへのイメージのプロビジョニングのために最適化された RHEL for Edge Simplified Installer イメージが導入されています。

詳細は、[「RHEL for Edge」](#) を参照してください。

Security

システム全体の暗号化ポリシーは、カスタムポリシーのディレクティブのスコープおよびワイルドカードをサポートします。異なるバックエンドに対して、異なるアルゴリズムセットを有効化できるようになりました。

Rsyslog ログ処理アプリケーションがバージョン 8.2102.0-5 に更新されました。今回の更新で、OpenSSL ネットワークストリームドライバーなどの改善が追加されました。これにより、OpenSSL ライブラリーを使用して TLS で保護されるトранスポートが Rsyslog に実装されます。

SCAP セキュリティーガイド プロジェクトでは、新規プロファイルおよび既存のプロファイルの改善点が複数追加されました。

- Australian Cyber Security Centre Information Security Manual(ACSC ISM) に準拠した新しいプロファイル。
- Center for Internet Security(CIS) プロファイルが 4 つの異なるプロファイル (Workstation Level 1、Workstation Level 2、Server Level 1、Server Level 2) に再構築。
- Security Technical Implementation Guide(STIG) セキュリティープロファイルがバージョン V1R3 に更新されました。
- **Server with GUI** インストールと互換性のある新しい STIG プロファイル。
- 新しい French National Security Agency(ANSSI)High Level プロファイル。これは、**SCAP Security Guide** のすべての ANSSI-BP-028 v1.2 強化レベルに対してプロファイルの可用性を完了します。

これらの機能強化により、これらのセキュリティーベースラインのいずれかに準拠するシステムをインストールし、OpenSCAP スイートを使用して、関連する機関が定義するセキュリティー制御のリスクベースのアプローチを使用して、セキュリティーコンプライアンスと修復を確認できます。

詳細は、[新機能 - セキュリティ](#) を参照してください。

新しい RHEL VPN システムロールを使用すると、多数のホストで、安全で適切な IPsec トンネリングおよび仮想プライベートネットワーク (VPN) ソリューションの設定が容易になります。詳細は、[新機能 - Red Hat Enterprise Linux システムロール](#) を参照してください。

ネットワーキング

NetworkManager は、全トライアックを受け入れるデバイスの設定に対応するようになりました。この機能は、**nmcli** ユーティリティーなどを使用すると設定できます。

firewalld サービスは、ゾーン内の異なるインターフェイスまたはソース間のトライアック転送に対応します。

firewalld サービスは、ゾーン間で転送されるトライアックのフィルターに対応します。

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。

- Ruby 3.0
- nginx 1.20
- Node.js 16

以下のコンポーネントがアップグレードされました。

- PHP: バージョン 7.4.19 へ
- Squid: バージョン 4.15 へ
- Mutt: バージョン 2.0.7 へ

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

コンパイラーおよび開発ツール

以下のコンパイラーツールセットが更新されました。

- GCC Toolset 11
- LLVM Toolset 12.0.1
- Rust Toolset 1.54.0
- Go Toolset 1.16.7

詳しくは [新機能 - コンパイラーおよび開発ツール](#) をご覧ください。

OpenJDK の更新

- Open Java Development Kit 17(OpenJDK 17) が利用できるようになりました。このリリースで導入された機能や既存の機能の変更に関する詳細は、[OpenJDK 11 のドキュメント](#) を参照してください。
- OpenJDK 11 がバージョン 11.0.13 に更新されました。このリリースで導入された機能や既存の機能の変更に関する詳細は、[OpenJDK 11 のドキュメント](#) を参照してください。

- OpenJDK 8 がバージョン 8.0.312 に更新されました。このリリースで導入された機能や既存の機能の変更に関する詳細は、[OpenJDK 11 のドキュメント](#) を参照してください。

Red Hat Enterprise Linux システムロール

Postfix RHEL システムロールに完全対応

Timesync RHEL システムロールに追加された NTS (Network Time Security) オプション

Storage RHEL システムロールが LVM VDO ボリュームをサポートし、ボリュームサイズをパーセンテージで表現

新しい RHEL VPN システムロールを使用すると、多数のホストで、安全で適切な IPsec トンネリングおよび仮想プライベートネットワーク (VPN) ソリューションの設定が容易になります。

8.5 GA リリースでは、High Availability Cluster RHEL システムロールはテクノロジープレビューとして利用可能

詳細は、[新機能 - Red Hat Enterprise Linux システムロール](#) および [テクノロジープレビュー - Red Hat Enterprise Linux システムロール](#) を参照してください。

1.2. インプレースアップグレードおよび OS 移行

RHEL 7 から RHEL 8 へのインプレースアップグレード

現在サポートされているインプレースアップグレードパスは次のとおりです。

- 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 8.4 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 (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 8.8.4 のアップグレード。これは、これらのアーキテクチャーの最終のインプレースアップグレードパスです。
- SAP HANA のシステムにおける RHEL 7.7 から RHEL 8.2 へのアップグレード。RHEL 8.2 にアップグレードした後に SAP HANA のシステムに対応していることを確認するには、RHEL 8.2 Update Services for SAP Solutions (E4S) リポジトリを有効にします。

RHEL 8.4 へのアップグレード後もシステムがサポートされるようにするには、最新の RHEL 8.5 バージョンに更新するか、RHEL 8.4 Extended Update Support (EUS) リポジトリを有効にします。SAP HANA を使用するシステムで、RHEL 8.2 Update Services for SAP Solutions (E4S) リポジトリを有効にします。

詳細は [Red Hat Enterprise Linux のサポート対象のインプレースアップグレードパス](#) 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。SAP 環境があるシステムでインプレースアップグレードを実行する手順については、[How to in-place upgrade SAP environments from RHEL 7 to RHEL 8](#) を参照してください。

主な機能拡張は、次のとおりです。

- AWS 上の Pay-As-You-Go インスタンス上の SAP HANA を Red Hat Update Infrastructure (RHUI) を使用してインプレースアップグレードすることが可能になりました。
- インプレースアップグレード時に、EUS または E4S リポジトリを有効にできるようになりました。
- **yum install leapp-upgrade** コマンドで Leapp ユーティリティーがインストールできるようになりました。この変更に伴い、**leapp-repository** および **leapp-repository-deps** RPM パッケージ

ジは、それぞれ **leapp-upgrade-el7toel8** および **leapp-upgrade-el7toel8-deps** に名称変更されました。古いパッケージがすでにシステムにインストールされている場合は、**yum update** を実行すると自動的に新しいパッケージに置き換えられます。

- Leapp のレポート、ログ、その他の生成されたドキュメントは、言語設定に関わらず、英語で表示されます。
- アップグレード後、残った Leapp パッケージをシステムから削除するには、**/etc/dnf/dnf.conf** 設定ファイルの **exclude** リストから手動で削除する必要があります。
- **leapp-data15.tar.gz** アーカイブに含まれる **repomap.csv** ファイルは非推奨になり、**repomap.json** ファイルに置き換えられました。非推奨のファイルは、2022 年 3 月まで利用可能です。
- IBM POWER 9(リトルエンディアン) と IBM Z(Structure A) のアーキテクチャーは、ライフサイクルを終えました。新しいアップグレードパス、機能、およびバグフィックスを含む、インプレースアップグレードの後続リリースには、これらのアーキテクチャーは含まれません。

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 8.4 にアップグレードするには、[Upgrading from RHEL 6 to RHEL 8](#) の手順に従います。

別の Linux ディストリビューションから RHEL への移行

CentOS Linux 8 または Oracle Linux 8 を使用している場合は、Red Hat がサポートする **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 8 に変換できます。詳細は、[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。

CentOS Linux または Oracle Linux の旧バージョン (バージョン 6 または 7) を使用している場合は、お使いのオペレーティングシステムを RHEL に移行してから、RHEL 8 へのインプレースアップグレードを実行できます。CentOS Linux 6 および Oracle Linux 6 変換は、サポート対象外の **Convert2RHEL** ユーティリティーを使用することに注意してください。サポートされていない変換の詳細については、[How to perform an unsupported conversion from a RHEL-derived Linux distribution to RHEL](#) を参照してください。

Red Hat が他の Linux ディストリビューションから RHEL への移行は、[Convert2RHEL サポートポリシー](#) を参照してください。

1.3. RED HAT CUSTOMER PORTAL LABS

Red Hat Customer Portal Labs は、カスタマーポータル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、セキュリティー問題の特定、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 Enterprise Linux Upgrade Helper](#)
- [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](#)
- [Red Hat Memory Analyzer](#)
- [Kernel Oops Analyzer](#)
- [Red Hat Product Errata Advisory Checker](#)
- [Red Hat メモリー不足アナライザー](#)

1.4. 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。
- 削除された機能を含む主な[RHEL 7 と RHEL 8 の相違点](#)は、[RHEL 8 の導入における考慮事項](#) で説明されています。
- [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、[Upgrading from RHEL 7 to RHEL 8](#) 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。

第2章 アーキテクチャー

Red Hat Enterprise Linux 8.5 ではカーネルバージョン 4.18.0-348 が使用されており、以下のアーキテクチャーに対応します。

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

各アーキテクチャーに適切なサブスクリプションを購入してください。詳細は [Get Started with Red Hat Enterprise Linux - additional architectures](#) を参照してください。利用可能なサブスクリプションのリストは、カスタマーportalの [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 リポジトリーより AppStream リポジトリーや追加リポジトリが含まれ、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 リポジトリーや AppStream リポジトリー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、[インストールメディアから RHEL を対話形式でインストールする](#) ドキュメントを参照してください。自動キックスタートインストールやその他の高度なトピックについては、[RHEL の自動インストール](#) ドキュメント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。複数のコンテナー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 新機能

ここでは、Red Hat Enterprise Linux 8.5 に追加された新機能および主要な機能拡張を説明します。

4.1. インストーラーおよびイメージの作成

RHEL for Edge が単純化されたインストーラーをサポートするようになりました。

今回の機能拡張により、Image Builder が RHEL for Edge Simplified Installer(**edge-simplified-installer**) および RHEL for Edge Raw Images(**edge-raw-image**) をビルドできるようになりました。

RHEL for Edge Simplified Installer を使用すると、新しい Blueprint オプション **installation_device** を指定し、デバイスへの無人インストールを実行できます。raw イメージを作成するには、既存の OSTree コミットを指定する必要があります。これにより、既存のコミットがデプロイされた raw イメージが作成されます。インストーラーは、この raw イメージを指定されたインストールデバイスに使用します。

また、Image Builder を使用して RHEL for Edge Raw Images をビルドすることもできます。これらは、既存の OSTree コミットがデプロイされたパーティションレイアウトを含む圧縮された raw イメージです。RHEL for Edge Raw Images をインストールして、ハードドライブにフラッシュするか、仮想マシンで起動することができます。

([BZ#1937854](#))

非推奨のカーネルブート引数の警告

inst. 接頭辞 (**ks**、**stage2**、**repo** など) なしの Anaconda ブート引数は、RHEL7 の起動が非推奨になりました。これらの引数は、次の RHEL メジャーリリースで削除されます。

今回のリリースにより、**inst** 接頭辞なしでブート引数を使用すると、適切な警告メッセージが表示されるようになりました。警告メッセージは、インストールの起動時に **dracut** に表示されます。また、インストールプログラムがターミナルで開始される際にも表示されます。

以下は、ターミナルに表示される警告メッセージの例です。

非推奨のブート引数 **ks** は、**inst.** 接頭辞とともに使用する必要があります。代わりに **inst.ks** を使用してください。**inst.** 接頭辞のない Anaconda ブート引数は非推奨となり、今後のメジャーリリースで削除されます。

以下は、**dracut** に表示される警告メッセージの例です。

ks が非推奨になりました。**inst.** 接頭辞のない Anaconda ブート引数の使用はすべて非推奨となり、今後のメジャーリリースで削除されます。代わりに **inst.ks** を使用してください。

([BZ#1897657](#))

Red Hat Connector が完全にサポートされました。

Red Hat Connector (**rhc**) を使用してシステムに接続することができます。Red Hat Connector はコマンドラインインターフェイスとデーモンで設定されており、ユーザーは Insights の Web ユーザーインターフェイス (console.redhat.com) 内で、Insights の修復 Playbook をホスト上で直接実行することができます。Red Hat Connector は、RHEL 8.4 でテクノロジープレビューとして提供され、RHEL 8.5 では完全にサポートされています。

([BZ#1957316](#))

利用可能な正式なリポジトリを上書きする機能

デフォルトでは、**osbuild-composer** バックエンドには、**/usr/share/osbuild-composer/repositories** ディレクトリーで定義されている正式なリポジトリの独自のセットがあります。したがって、**/etc/yum.repos.d/** ディレクトリーにあるシステムリポジトリを継承しません。正式なリポジトリを上書きできるようになりました。これには、**/etc/osbuild-composer/repositories** にオーバーライドを定義します。その結果、そのディレクトリーにあるファイルは、**/usr** ディレクトリーにあるファイルよりも優先されます。

(BZ#1915351)

Image Builder がファイルシステム設定に対応しました。

今回の機能拡張により、Blueprint でカスタムファイルシステム設定を指定でき、必要なディスクレイアウトでイメージを作成できるようになりました。したがって、デフォルト以外のレイアウトを持つことで、セキュリティーベンチマーク、既存設定との一貫性、パフォーマンス、およびディスク不足エラーに対する保護に関してメリットが得られます。

Blueprint でファイルシステム設定をカスタマイズするには、以下のカスタマイズを設定します。

```
[[customizations.filesystem]]
mountpoint = "MOUNTPOINT"
size = MINIMUM-PARTITION-SIZE
```

(BZ#2011448)

Image Builder が起動可能なインストーライメージの作成に対応

この機能拡張により、Image Builder を使用して、ルートファイルシステムを含む **tarball** ファイルで構成されるブート可能な ISO イメージを作成できるようになりました。その結果、起動可能な ISO イメージを使用して、**tarball** ファイルシステムをベアメタルシステムにインストールすることができます。

(BZ#2019318)

4.2. RHEL FOR EDGE

Greenboot サービスはデフォルトで有効になりました。

以前のリリースでは、greenboot サービスはデフォルトの事前設定に存在しないため、greenboot パッケージのインストール時に、ユーザーはこれらの greenboot サービスを手動で有効にする必要がありました。今回の更新で、greenboot サービスがデフォルトの事前設定に存在するようになり、ユーザーは手動で有効化する必要がなくなりました。

(BZ#1935177)

4.3. ソフトウェア管理

RPM に **sqlite** データベースバックエンドに対する読み取り専用サポートが追加されました。

sqlite に基づく RPM データベースをクエリーする機能は、コンテナーなどの他のルートディレクトリーを検査する際に必要になります。この更新により、RPM **sqlite** データベースバックエンドに対して読み取り専用サポートが追加されます。これにより、ホストの RHEL 8 から、UBI 9 または Fedora コンテナーにインストールされているパッケージをクエリーできるようになりました。Podman でこれを行うには、以下を実行します。

1. **podman mount** コマンドを使用して、コンテナーのファイルシステムをマウントします。

- マウントされた場所をポイントする **--root** オプションを指定して、**rpm -qa** コマンドを実行します。

RHEL 8 の RPM は、引き続き BerkeleyDB データベース (**bdb**) バックエンドを使用することに注意してください。

(BZ#1938928)

libmodulemd がバージョン 2.12.1 にリベースされました。

libmodulemd パッケージがバージョン 2.12.1 にリベースされました。主な変更点は、以下のとおりです。

- modulemd-obsolete** ドキュメントタイプのバージョン 1 のサポートを追加。これにより、別のストリームを廃止するストリームや、ライフサイクル終了に到達したストリームに関する情報が提供されます。
- モジュールビルドシステムのモジュールストリームコンテンツのパッケージャーの説明を提供する **modulemd-packager** ドキュメントタイプのバージョン 3 のサポートを追加
- バージョン 2 の **modulemd** ドキュメントタイプの **static_context** 属性のサポートを追加。そのため、モジュールコンテキストはモジュールビルドシステムで生成されるのではなく、パッケージャーによって定義されるようになりました。
- モジュールストリーム値は、常に引用符で囲まれた文字列としてシリアル化されるようになりました。

(BZ#1894573)

libmodulemd がバージョン 2.13.0 にリベースされました。

libmodulemd パッケージがバージョン 2.13.0 にリベースされ、以前のバージョンに対する主な変更点が加えられています。

- モジュールからモジュール化解除されたパッケージをリストから除外するサポートを追加
- modulemd-validator** ツールの新しい **--type** オプションを使用して、**modulemd-packager- v3** ドキュメントを検証するためのサポートを追加。
- 整数値を解析する機能強化
- 各種の **modulemd-validator** の問題を修正

(BZ#1984402)

sslverifystatus が **dnf** 設定に追加されました

今回の更新により、**sslverifystatus** オプションを有効にすると、**dnf** は **Certificate Status Request** TLS 拡張 (OCSP stapling) を使用して各サーバー証明書の失効ステータスを確認します。これにより、取り消された証明書が発生すると、**dnf** はそのサーバーからダウンロードを拒否します。

(BZ#1814383)

4.4. シェルおよびコマンドラインツール

ReaR はバージョン 2.6 に更新されました。

ReaR(Relax-and-Recover) がバージョン 2.6 に更新されました。主なバグ修正と機能拡張は、以下のとおりです。

- **eMMC** デバイスのサポートを追加
- デフォルトでは、すべてのカーネルモジュールはレスキューシステムに含まれます。特定のモジュールを含めるには、設定ファイルに **MODULES** アレイ変数を **MODULES=(mod1 mod2)** に設定します。
- AMD および Intel の 64 ビットアーキテクチャー、IBM Power Systems、Little Endian では、ブートローダーのインストール場所を制御するために、新しい設定変数 **GRUB2_INSTALL_DEVICES** が導入されています。詳細は、[/usr/share/rear/conf/default.conf](#) の説明を参照してください。
- マルチパスデバイスのバックアップが改善
- **/media**、**/run**、**/mnt**、**/tmp** のディレクトリーにはリムーバブルメディアまたは一時ファイルが含まれることが認識されるため、これらのディレクトリー配下のファイルは、自動的にバックアップから除外されます。[/usr/share/rear/conf/default.conf](#) の **AUTOEXCLUDE_PATH** 変数の説明を参照してください。
- **CLONE_ALL_USERS_GROUPS=true** がデフォルトになりました。詳細は、[/usr/share/rear/conf/default.conf](#) の説明を参照してください。

(BZ#1988493)

modulemd-tools パッケージが利用できるようになりました。

今回の更新で、**modulemd-tools** パッケージが導入され、**modulemd** YAML ファイルを解析および生成するためのツールを提供するようになりました。

modulemd-tools をインストールするには、以下を使用します。

```
# yum install modulemd-tools
```

(BZ#1924850)

opencryptoki がバージョン 3.16.0 にリベースされました。

opencryptoki がバージョン 3.16.0 にアップグレードされました。主なバグ修正と機能拡張は、以下のとおりです。

- **EP11** コアプロセッサーで、**protected-key** オプションと **attribute-bound keys** のサポートを強化。
- **cycle-count-accurate** (CCA) プロセッサーでセキュアな鍵オブジェクトのインポートおよびエクスポートを強化。

(BZ#1919223)

lsvpd がバージョン 1.7.12 にリベースされました。

lsvpd がバージョン 1.7.12 にアップグレードされました。主なバグ修正と機能拡張は、以下のとおりです。

- **sysvpd** の UUID プロパティーが追加されました。

- NVMe ファームウェアバージョンが改善されました。
- PCI デバイスメーカー解析ロジックを修正
- **lsvpd** 設定ファイルに **recommends clause** を追加

(BZ#1844428)

ppc64-diag がバージョン 2.7.7 にリベースされました。

ppc64-diag がバージョン 2.7.7 にアップグレードされました。主なバグ修正と機能拡張は、以下のとおりです。

- ユニットテストケースが改善されました。
- **sysvpd** の UUID プロパティーが追加されました。
- **rtas_errd** サービスは、Linux コンテナーでは実行されません。
- 廃止されたロギングオプションは、**systemd** サービスファイルでは利用できなくなりました。

(BZ#1779206)

ipmi_power と ipmi_boot モジュールが、redhat.rhel_mgmtCollection で利用可能です

この更新により、Intelligent Platform Management Interface (IPMI) Ansible モジュールがサポートされます。IPMI は、ベースボード管理コントローラー (BMC) デバイスと通信するための一連の管理インターフェイスの仕様です。IPMI モジュールである **ipmi_power** と **ipmi_boot** は、**redhat.rhel_mgmtCollection** に含まれており、**ansible-collection-redhat-rhel_mgmt** パッケージをインストールすることで入手できます。

(BZ#1843859)

udftools 2.3 が RHEL に追加されました。

udftools パッケージは、Universal Disk Format(UDF) ファイルシステムを操作するユーザー空間ユーティリティーを提供します。今回の機能拡張により、**udftools** は以下のツールセットを提供します。

- **cdrwtool** - 空白、フォーマット、クイック設定、および DVD-R/CD-R/CD-RW メディアに書き込むなどのアクションを実行します。
- **mkfs.udf**、**mkudffs** - Universal Disk Format(UDF) ファイルシステムを作成します。
- **pktsetup** - パケットデバイスを設定し、破棄します。
- **udfinfo** - Universal Disk Format(UDF) ファイルシステムに関する情報を表示します。
- **udflabel** - Universal Disk Format(UDF) ファイルシステムラベルを表示または変更します。
- **wrudf** - 既存の Universal Disk Format(UDF) ファイルシステムで、**cp**、**rm**、**mkdir**、**rmdir**、**ls**、および **cd** 操作でインタラクティブシェルを提供します。

(BZ#1882531)

RHEL 8.5 に Tesseract 4.1.1 が追加されました。

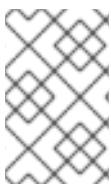
Tesseract はオープンソースの OCR(optical character reading) エンジンで、以下の機能があります。

- **tesseract** バージョン 4 以降では、文字認識は Long Short-Term Memory(LSTM) ニューラル ネットワークに基づいています。
- UTF-8 をサポートします。
- プレーンテキスト、hOCR(HTML)、PDF、および TSV 出力形式をサポートします。

(BZ#1826085)

シンプールを使用する LVM を復元する際のエラーは発生しません。

今回の機能拡張により、ReaR は、カーネルメタデータを持つシンプールおよびその他の論理ボリュームタイプ(RAID やキャッシュなど)がボリュームグループ(VG)で使用されるかどうかを検出し、lvcreate コマンドを使用して VG のすべての論理ボリューム(LV)を再作成するモードに切り替わるようになりました。したがって、シンプールを使用する LVM はエラーなしで復元されます。



注記

この新しいメソッドは、LVM UUID などの LV プロパティーをすべて保持しません。再作成されたストレージレイアウトが要件と一致するかどうかを判断するために、実稼働環境で ReaR を使用する前に、バックアップからの復元をテストする必要があります。

(BZ#1747468)

Net-SNMP が RSA および ECC 証明書を検出するようになりました。

以前は、Net-Simple Network Management Protocol(Net-SNMP) は Rivest、Shamir、Adleman(RSA) 証明書のみを検出しました。今回の機能拡張により、ECC(Elliptic Curve Cryptography) がサポートされるようになりました。その結果、Net-SNMP が RSA および ECC 証明書を検出するようになりました。

(BZ#1919714)

FCoE オプションが rd.fcoe に変更

以前は、**dracut.cmdline** の man ページで FCoE(Fibre Channel over Ethernet) をオフにするコマンドとして **rd.nofcoe=0** が記載されていました。

今回の更新で、コマンドは **rd.fcoe** に変更されました。FCoE を無効にするには、**rd.fcoe=0** コマンドを実行します。

FCoE の詳細は [Fibre Channel over Ethernet の設定](#) を参照してください。

(BZ#1929201)

4.5. インフラストラクチャーサービス

linuxptp がバージョン 3.1にリベースされました。

linuxptp パッケージがバージョン 3.1に更新されました。主なバグ修正と機能拡張は、以下のとおりです。

- Precision Time Protocol(PTP) ハードウェアロックを Pulse Per Second(PPS) シグナルに同期するための **ts2phc** プログラムが追加されました。
- automotive プロファイルのサポートを追加

- クライアントイベント監視のサポートを追加

(BZ#1895005)

chrony がバージョン 4.1 にリベースされました。

chrony がバージョン 4.1 に更新されました。主なバグ修正と機能拡張は、以下のとおりです。

- Network Time Security(NTS) 認証のサポートを追加 詳細は、[chrony における Network Time Security \(NTS\) の概要](#) を参照してください。
- デフォルトでは、認証されたネットワークタイムプロトコル(NTP) ソースは、非認証の NTP ソースで信頼されます。 **chrony.conf** ファイルに **autoselectmode ignore** 引数を追加して、元の動作を復元します。
- **RIPemd** キー (**RMD128**、**RMD160**、**RMD256**、**RMD320**) での認証のサポートが利用できなくなりました。
- NTPv4 パケットにおける長い非標準 MAC のサポートが利用できなくなりました。 **chrony 2.x (MD5/SHA1 以外の 鍵)** を使用している場合は、バージョン 3 オプションで **chrony** を設定する必要があります。

(BZ#1895003)

powertop がバージョン 2.14 にリベースされました。

powertop がバージョン 2.14 にアップグレードされました。これは、Alder Lake、Sapphire Rapids、および Rocket Lake プラットフォームのサポートを追加する更新です。

(BZ#1834722)

Tuned が不要な IRQ をハウスキーピング CPU に移動します。

i40e、**iavf**、**mlx5** などのネットワークデバイスドライバーは、オンラインの CPU を評価してキュー、したがって作成する **MSIX** ベクターの数を判断します。

分離した CPU が多くハウスキーピング CPU が非常に少ない低レイテンシー環境では、Tuned がこれらのデバイス IRQ をハウスキーピング CPU に移動しようとすると、CPU あたりのベクターの制限により失敗します。

今回の機能拡張により、Tuned はハウスキーピング CPU に応じてネットワークデバイスチャネル(したがって MSIX ベクター)の数を明示的に調整するようになりました。したがって、すべてのデバイス IRQ はハウスキーピング CPU に移動し、低レイテンシーを実現できるようになりました。

(BZ#1951992)

4.6. SECURITY

libreswan を 4.4 にリベース

libreswan パッケージがアップストリームバージョン 4.4 にアップグレードされ、多くの機能強化とバグ修正が行われました。以下に例を示します。

- IKEv2 プロトコル:
 - トランスポートモード およびホスト間接続での TCP カプセル化に関する修正を導入しました。

- リダイレクトの統計情報を表示する **ipsec whack** コマンドに、**--globalstatus** オプションを追加しました。
- ipsec.conf** 設定ファイルの **vhost** および **vnet** の値は、IKEv2 接続では使用できなくなりました。
- pluto** IKE デーモン:
 - 非標準の IKE ポートを使用するホスト間接続の修正を導入しました。
 - 最適な初期接続を選択するために、ピア ID(**IKEv2 IDr** または **IKEv1 Aggr**) を追加しました。
 - Libreswan が対応する機能をまだ提供していないため、**interface-ip=** オプションを無効にしました。
 - トランSPORTモード の NAT に対する **ipsec_updown** スクリプトの **PLUTO_PEER_CLIENT** 変数を修正しました。
 - PLUTO_CONNECTION_TYPE** 変数を **transport** または **tunnel** に設定します。
 - テンプレート化されないワイルドカード ID の接続がマッチするようになりました。

(BZ#1958968)

gnutls が 3.6.16 にリベース

gnutls パッケージがバージョン 3.6.16 に更新されました。主なバグ修正と機能拡張は、以下のとおりです。

- gnutls_x509_crt_export2()** 関数は、成功した場合に、内部の base64 ブロブのサイズの代わりに 0 を返すようになりました。これは、**gnutls_x509_crt_export2(3)** man ページのドキュメントと一致しています。
- OCSP(Online Certificate Status Protocol) の must-stapling に従わないことによる証明書検証の失敗に、**GNUTLS_CERT_INVALID** フラグが正しく付くようになりました。
- これまで、**-VERS-TLS1.2** オプションで TLS 1.2 を明示的に無効にしても、TLS 1.3 が有効な場合にサーバーが TLS 1.2 を提供していました。バージョンネゴシエーションが修正され、TLS 1.2 を正しく無効にできるようになりました。

(BZ#1956783)

socat が 1.7.4 にリベースされました。

socat パッケージがバージョン 1.7.3 から 1.7.4 にアップグレードされ、バグ修正および改善点が数多く追加されました。以下に例を示します。

- GOPEN** および **UNIX-CLIENT** アドレスが **SEQPACKET** ソケットに対応するようになりました。
- アドレスのリッスンまたは許可の場合、汎用 **setsockopt-int** および関連オプションは、接続されたソケットに適用されます。リスニングソケットで設定オプションを有効にするために、**setsockopt-listen** オプションが利用できるようになりました。
- ファイルに転送されたデータのローダーンプの **-r** オプションおよび **-R** オプションを追加。

- **ip-transparent** オプションと **IP_TRANSPARENT** ソケットオプションを追加。
- **OPENSSL-CONNECT** は SNI 機能を自動的に使用し、**openssl-no-sni** オプションが SNI をオフにするようになりました。**openssl-snihost** オプションは、**openssl-commonname** オプションまたはサーバー名の値を上書きします。
- **accept-timeout** オプションおよび **listen-timeout** オプションが追加されました。
- **ip-add-source-membership** オプションが追加されました。
- **UDP-DATAGRAM** アドレスは、1.7.3 で行っていた応答のピアポートの確認を行わなくなりました。シナリオに以前の動作が必要な場合は、**sourceport** オプションを使用します。
- 新しい **proxy-authorization-file** オプションは、ファイルから **PROXY-CONNECT** 認証情報を読み取り、プロセステーブルからこのデータを非表示にできるようにします。
- **VSOCK-CONNECT** および **VSOCK-LISTEN** アドレスの **AF_VSOCK** サポートが追加されました。

(BZ#1947338)

crypto-policies が 20210617 にリベースされました。

crypto-policies パッケージがアップストリームバージョン 20210617 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。主な変更点は以下のとおりです。

- スコープ指定されたポリシーを使用して、異なるバックエンドに対して、異なるアルゴリズムセットを有効化できるようになりました。各設定ディレクティブは、特定のプロトコル、ライブラリー、またはサービスに制限できるようになりました。利用可能なスコープの全リストと新しい構文の詳細は、**crypto-policies(7)** の man ページを参照してください。たとえば、以下のディレクティブは、**libssh** ライブラリーと OpenSSH スイートの両方に対して、SSH プロトコルでの AES-256-CBC 暗号の使用を許可します。

cipher@SSH = AES-256-CBC+
- ディレクティブで、ワイルドカードを使用して複数の値を指定するアスタリスクを使用できるようになりました。たとえば、以下のディレクティブは、**libssh** を使用したアプリケーションのすべての CBC モード暗号を無効にします。

cipher@libssh = -* -CBC

今後の更新では、現在のワイルドカードで一致した新しいアルゴリズムが導入される可能性があることに注意してください。

(BZ#1960266)

crypto-policies が、カスタムポリシーで AES-192 暗号に対応するようになりました。

システム全体の暗号化ポリシーが、カスタムポリシーとサブポリシーの **cipher** オプションに対し、**AES-192-GCM**、**AES-192-CCM**、**AES-192-CTR** および **AES-192-CBC** の値に対応するようになりました。これにより、Libreswan アプリケーションの **AES-192-GCM** および **AES-192-CBC** 暗号、**libssh** ライブラリーの **AES-192-CTR** および **AES-192-CBC** 暗号、および **crypto-policies** を介して OpenSSH スイートを有効にできます。

(BZ#1876846)

FUTURE 暗号化ポリシーで CBC 暗号を無効にする

今回の **crypto-policies** パッケージの更新により、**FUTURE** ポリシーで CBC (cipher block chaining) モードを使用する暗号が無効になります。**FUTURE** の設定は、近い将来の攻撃に耐えうるものでなければならず、今回の変更は現在の進捗を反映したものです。そのため、**FUTURE** ポリシーが有効な場合、**crypto-policies** を遵守するシステムコンポーネントは CBC モードを使用できません。

(BZ#1933016)

新しいカーネル AVC トレースポイントの追加

今回の機能強化により、SELinux 拒否の監査時にトリガーする新しい **avc:selinux_audited** カーネルトレースポイントが追加されました。この機能により、SELinux 拒否の低レベルのデバッグがより容易になります。**perf** などのツールには、新しいトレースポイントが利用できます。

(BZ#1954024)

SCAP セキュリティガイドの新しい ACSC ISM プロファイル

scap-security-guide パッケージで、Australian Cyber Security Centre (ACSC) Information Security Manual (ISM) 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。この機能強化により、このセキュリティベースラインに準拠するシステムをインストールし、OpenSCAP スイートを使用して、ACSC が定義するセキュリティ制御のリスクベースのアプローチを使用して、セキュリティコンプライアンスと修復を確認できます。

(BZ#1955373)

SCAP セキュリティガイドが 0.1.57 にリベースされました。

scap-security-guide パッケージがアップストリームバージョン 0.1.57 にリベースされ、バグ修正および改善が複数追加されました。以下に例を示します。

- Australian Cyber Security Centre (ACSC) Information Security Manual (ISM) プロファイルが導入されました。このプロファイルは Essential Eight プロファイルを拡張し、ISM で定義されているより多くのセキュリティ制御を追加します。
- CCenter for Internet Security (CIS) プロファイルが、公式の CIS ベンチマークで定義されているように、強化レベルとシステムタイプ (サーバーおよびワークステーション) に対応する 4 つのプロファイルに再構築されました。
- セキュリティ技術実装ガイド (STIG) セキュリティプロファイルが更新され、最近リリースされたバージョン V1R3 からルールが実装されました。
- Security Technical Implementation Guide with GUI (STIG with GUI) セキュリティプロファイルが導入されました。プロファイルは STIG プロファイルに由来し、**Server with GUI** パッケージの選択を選択する RHEL インストールと互換性があります。
- French National Security Agency (ANSSI) の ANSSI BP-028 の推奨事項に基づく ANSSI High level プロファイルが導入されました。これには、高い強化レベルのルールを実装するプロファイルが含まれます。

(BZ#1966577)

OpenSCAP が 1.3.5 にリベースされました。

OpenSCAP パッケージがアップストリームバージョン 1.3.5 にリベースされました。主なバグ修正と機能強化は、以下のとおりです。

- oval および xccdf モジュールの validate コマンドで、デフォルトで Schematron ベースの検証を有効にします。
- SCAP 1.3 ソースデータストリーム Schematron を追加。
- XML 署名の検証を追加
- SOURCE_DATE_EPOCH** に mtime をクランプするのを許可しました。
- severity** および **role** 属性が追加されました。
- Rule and Group(XCCDF) の **requires** および **conflicts** 要素のサポート。
- HTML レポートでの Kubernetes 修復
- gpfs**、**proc**、および **sysfs** のファイルシステムをローカル以外のものとして処理しています。
- arg=val というスタイルの一般的なオプションの処理を修正
- StateType** Operator の動作が修正されました。
- XPath 式 (**xmlfilecontent**) で名前空間が無視され、不完全な XPath クエリーを許可します。
- 突出したデータの存在に関する警告を生じさせる問題を修正
- stig-viewer 機能で複数のセグメンテーション違反と破損テストを修正
- TestResult/benchmark/@href** 属性を修正
- メモリー管理の多くの問題が修正されました。
- 多くのメモリリークを修正

(BZ#1953092)

デジタル署名された SCAP ソースデータストリームの検証

Security Content Automation Protocol (SCAP) 1.3 仕様に準拠するため、OpenSCAP が、デジタルに署名された SCAP ソースデータストリームのデジタル署名を検証するようになりました。これにより、デジタル署名された SCAP ソースデータストリームを評価する際に、OpenSCAP がデジタル署名を検証します。署名の検証は、ファイルの読み込み時に自動的に実行されます。無効な署名を持つデータストリームが拒否され、OpenSCAP はコンテンツを評価しません。OpenSCAP は、OpenSSL 暗号ライブラリーと共に [XML Security Library](#) を使用して、デジタル署名を検証します。

oscap xccdf eval コマンドに **--skip-signature-validation** オプションを追加すると、署名の検証をスキップできます。



重要

OpenSCAP は、**KeyInfo** 署名要素の一部で署名の検証に使用される証明書または公開鍵の信頼性に対応していません。不正な攻撃者によって変更および署名されたデータストリームの評価を防ぐために、ご自身で鍵を検証する必要があります。

(BZ#1966612)

Server with GUI インストールと互換性のある新しい DISA STIG プロファイル。

新しいプロファイル **DISA STIG with GUI** が SCAP Security Guide に追加されました。プロファイルは **DISA STIG** プロファイルに由来し、**Server with GUI** パッケージグループを選択した RHEL インストールと互換性があります。DISA STIG ではグラフィカルユーザーインターフェイスをアンインストールする必要があるため、これまで存在した **stig** プロファイルは **Server with GUI** と互換性がありませんでした。ただし、評価中に Security Officer によって適切に文書化されている場合には、この設定を上書きできます。これにより、新しいプロファイルは、RHEL システムを DISA STIG プロファイルに準拠した **Server with GUI** としてインストールする際に役立ちます。

(BZ#1970137)

STIG セキュリティープロファイルがバージョン V1R3 に更新されました。

SCAP セキュリティーガイドの **DISA STIG for Red Hat Enterprise Linux 8** プロファイルが更新され、最新バージョンの **V1R3** に合わせて更新されました。このプロファイルはより安定し、DISA (Defense Information Systems Agency) が提供する RHEL 8 STIG (Security Technical Implementation Guide) のマニュアルベンチマークにより適切に調整されるようになりました。

この 2 番目の反復により、STIG に関するカバレッジの約 90% が発生します。古バージョンが有効でなくなったため、このプロファイルの現行バージョンのみを使用する必要があります。

警告



自動修正によりシステムが機能しなくなる場合があります。まずテスト環境で修復を実行してください。

(BZ#1993056)

SCAP セキュリティーガイドの 3 つの新しい CIS プロファイル

Center for Internet Security (CIS) Red Hat Enterprise Linux 8 Benchmark に準拠する 3 つの新しいコンプライアンスプロファイルが SCAP セキュリティーガイドに導入されました。CIS RHEL 8 Benchmark は、サーバーとワークステーションデプロイメントには、異なる設定の推奨事項を提供し、デプロイメントごとに 2 つの設定レベルレベル 1 とレベル 2 を定義します。従来 RHEL8 に同梱されていた CIS プロファイルは、サーバーレベル 2 しか表しませんでした。新しい 3 つのプロファイルは CIS RHEL8 Benchmark プロファイルのスコープを完了し、CIS の推奨事項に対してシステムをより簡単に評価できるようになりました。

現在利用可能なすべての CIS RHEL 8 プロファイルは次のとおりです。

ワークステーションレベル 1	<code>xccdf_org.ssgproject.content_profile_cis_workstation_l1</code>
ワークステーションレベル 2	<code>xccdf_org.ssgproject.content_profile_cis_workstation_l2</code>
サーバーレベル 1	<code>xccdf_org.ssgproject.content_profile_cis_server_l1</code>

サーバーレベル 2	xccdf_org.ssgproject.content_profile_cis
-----------	---

(BZ#1993197)

同様のシステムコールをグループ化して Audit の修復のパフォーマンスを強化

以前のバージョンでは、Audit の修正は、プロファイルが監査する各システムコールに対して個々のルールを生成していました。これが、パフォーマンスを低下させる多くの監査ルールの原因でした。今回の機能強化により、Audit の修復では、同じフィールドを持つ同様のシステムコールのルールを1つのルールにグループ化でき、パフォーマンスが向上します。

1つにグループ化されたシステムコールの例:

```
-a always, exit -F arch=b32 -S chown, fchown, fchownat, lchown -F auid>=1000 -F auid!=unset -F key=perm_mod
```

```
-a always, exit -F arch=b32 -S unlink, unlinkat, rename, renameat, rmdir -F auid>=1000 -F auid!=unset -F key=delete
```

```
-a always, exit -F arch=b32 -S chown, fchown, fchownat, lchown -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccessful-perm-change
```

```
-a always, exit -F arch=b32 -S unlink, unlinkat, rename, renameat -F auid>=1000 -F auid!=unset -F exit=-EACCES -F auid>=1000 -F auid!=unset -F key=unsuccessful-delete
```

(BZ#1876483)

ANSSI-BP-028 High Level のプロファイルを追加

French National Security Agency (ANSSI) の ANSSI BP-028 の推奨事項に基づく ANSSI High level プロファイルが導入されました。これは、SCAP Security Guideのすべての ANSSI-BP-028 v1.2 強化レベルに対してプロファイルの可用性を完了します。新しいプロファイルを使用すると、システムを GNU/Linux システムの ANSSI からの推奨事項の高い強化レベルに強化できます。これにより、ANSSI Ansible Playbook および ANSSI SCAP プロファイルを使用し、RHEL 8 システムの最も厳しい強化レベルへのコンプライアンスを設定および自動化できます。

(BZ#1955183)

Rsyslog TCP および RELP トラフィックを暗号化する OpenSSL が追加されました。

OpenSSL ネットワークストリームドライバーが Rsyslog に追加されました。このドライバーは、OpenSSL ライブラリーを使用して TLS で保護されるトランスポートを実装します。これにより、GnuTLS ライブラリーを使用するストリームドライバーよりも、追加機能が提供されます。これにより、Rsyslog ネットワークストリームドライバーとして OpenSSL または GnuTLS のどちらかを使用できるようになりました。

(BZ#1891458)

rsyslog が 8.2102.0-5 にリベースされました。

rsyslog パッケージがアップストリームバージョン 8.2102.0-5 にリベースされ、以前のバージョンに対する主な変更点が加えられています。

- 変数が存在するかどうかを確認するために **exists()** スクリプト関数を追加 (例: `$!path!var`)。
- omrelp** および **imrelp** モジュールの新たな設定パラメーター **tls.tlscfgcmd** で OpenSSL 設定コマンドを設定するサポートが追加されました。
- リモートサーバーに送信されるレート制限 syslog メッセージに対して **omfwd** モジュールに新しいレート制限オプションが追加されました。
 - RateLimit.interval** はレート制限間隔を秒単位で指定します。
 - RateLimit.burst** は、メッセージの数でレート制限バーストを指定します。
- さまざまな改善と共に **immark** モジュールが書き直されました。
- imptcp** モジュールに **max sessions** 設定パラメーターを追加。最大はインスタンスごとに測定されますが、全インスタンスでグローバルに測定されません。
- rsyslog-openssl** サブパッケージが追加されました。このネットワークストリームドライバーは、OpenSSL ライブラリーを使用して TLS で保護されるトランスポートを実装します。
- MaxBytesPerMinute** オプションおよび **MaxLinesPerMinute** オプションと共に、**imfile** モジュールに1分あたりのレート制限が追加されました。これらのオプションは整数値を受け入れ、1分で送信される可能性のあるバイト数または行数を制限します。
- streamdriver.TlsVerifyDepth** オプションを使用して証明書チェーンの検証の最大深さを設定する **imtcp** モジュールおよび **omfwd** モジュールのサポートを追加

(BZ#1932795)

4.7. ネットワーキング

NetworkManager での ethtool の pause パラメーターのサポート

特定のケースで特定のネットワークインターフェイスで、自動一時停止以外のパラメーターを明示的に設定する必要があります。以前は、NetworkManager は、**nmstate** の **ethtool** の制御フローパラメーターを一時停止できませんでした。pause パラメーターのオートネゴシエーションを無効にし、RX/TX 一時停止のサポートを明示的に有効にするには、以下のコマンドを使用します。

```
# nmcli connection modify enp1s0 ethtool.pause-autoneg no ethtool.pause-rx true ethtool.pause-tx true
```

(BZ#1899372)

プロミスキャスモードに物理インターフェイスおよび仮想インターフェイスを設定するための NetworkManager の新しいプロパティー

今回の更新で、**802-3-ethernet.accept-all-mac-addresses** プロパティーが NetworkManager に追加され、物理インターフェイスおよび仮想インターフェイスを **accept all MAC addresses** モードで設定できるようになりました。今回の更新により、カーネルは、**accept all MAC addresses** モードで、現在のインターフェイスの MAC アドレスを対象とするネットワークパッケージを受け入れることができます。**eth1** で **accept all MAC addresses** モードを有効にするには、次のコマンドを使用します。

```
$ sudo nmcli c add type ethernet ifname eth1 connection.id eth1 802-3-ethernet.accept-all-mac-addresses true
```

(BZ#1942331)

NetworkManager がバージョン 1.32.10 にリベースされました。

NetworkManager パッケージがアップストリームバージョン 1.32.10 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。

主な変更の詳細は、このバージョンの [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。

(BZ#1934465)

NetworkManager が、ファイアウォールバックエンドとして nftables に対応

今回の機能強化により、NetworkManager への **nftables** ファイアウォールフレームワークのサポートが追加されました。デフォルトのバックエンドを **iptables** から **nftables** に切り替えるには、以下を行います。

1. 以下の内容で `/etc/NetworkManager/conf.d/99-firewall-backend.conf` ファイルを作成します。

```
[main]
firewall-backend=nftables
```

2. NetworkManager サービスを再読み込みします。

```
# systemctl reload NetworkManager
```

(BZ#1548825)

firewalld がバージョン 0.9.3 にリベース

firewalld 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拡張機能とバグ修正が数多く追加されました。

詳細は、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。

- [firewalld 0.9.3 リリースノート](#)
- [firewalld 0.9.2 リリースノート](#)
- [firewalld 0.8.6 リリースノート](#)
- [firewalld 0.8.5 リリースノート](#)
- [firewalld 0.8.4 リリースノート](#)

(BZ#1872702)

firewalld ポリシーオブジェクト機能の提供開始

これまで、**firewalld** を使用して、仮想マシン、コンテナー、およびゾーン間を流れるトラフィックをフィルタリングすることはできませんでした。今回の更新では、**firewalld** ポリシーオブジェクト機能が導入され、**firewalld** でのフォワードおよびアウトプットフィルタリングが可能になりました。

(BZ#1492722)

マルチパス TCP が完全にサポートされるようになりました

RHEL 8.5 以降、Multipath TCP (MPTCP) が完全にサポートされています。MPTCP はネットワーク内のリソース使用状況を改善し、ネットワーク障害に対する耐障害性を確保します。たとえば、RHEL サーバーで Multipath TCP を使用すると、MPTCP v1 対応のスマートカードは、サーバーで実行中のアプリケーションに接続し、サーバーへの接続を中断せずに Wi-Fi とセルラーネットワークを切り替えることができます。

RHEL 8.5 では、以下のような追加の機能が導入されました。

- 複数の同時アクティブなサブストリーム
- アクティブバックアップサポート
- ストリームのパフォーマンスが向上
- バッファーの自動調整の `receive` and `send` によるメモリー使用量の向上
- SYN クッキーのサポート

サーバーで実行中のアプリケーションが MPTCP をネイティブにサポートするか、管理者が **eBPF** プログラムをカーネルにロードして、**IPPROTO_TCP** を **IPPROTO_MPTCP** に動的に変更する必要があることに注意してください。

詳細は、[Multipath TCP の使用](#) を参照してください。

(JIRA:RHELPLAN-57712)

RHEL で代わりのネットワークインターフェイスの命名が利用可能に

代替インターフェイスの命名は RHEL カーネル設定であり、次のネットワーク上の利点を提供します。

- ネットワークインターフェイスカード (NIC) 名の長さは任意です。
- 1つの NIC に同時に複数の名前を付けることができます。
- コマンドのハンドルとしての代替名の使用。

(BZ#2164986)

4.8. カーネル

RHEL 8.5 のカーネルバージョン

Red Hat Enterprise Linux 8.5 には、カーネルバージョン 4.18.0-348 が同梱されています。

(BZ#1839151)

Intel Sapphire Rapids プロセッサーの EDAC に対応しました。

今回の機能拡張により、Intel Sapphire Rapids プロセッサー用の Error Detection And Correction (EDAC) デバイスのサポートが提供されます。EDAC は主に Error Code Correction (ECC) メモリーを処理し、PCI バスパリティーエラーを検出し、報告します。

(BZ#1837389)

bpftrace パッケージがバージョン 0.12.1 にリベースされました。

bpftrace パッケージがバージョン 0.12.1 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。以前のバージョンでの大きな変更には、以下の変更が含まれます。

- 新しい **builtin** パスを追加しました。これは、パスストラクチャーからフルパスを表示する、信頼できる新しい方法です。
- kfunc** プローブおよび **tracepoint** カテゴリーに対するワイルドカードサポートが追加されました。

(BZ#1944716)

vmcore キャプチャーは、CPU のホットアドまたはホットリムーブ操作後に期待どおりに機能します。

以前は、IBM POWER システムでは、CPU またはメモリーのホットプラグまたは削除操作を行うたびに、**kdump.service** を再読み込みしない限り、デバイスツリーの CPU データが古くなりました。最新の CPU 情報を再読み込みするため、**kdump.service** はデバイスノードを解析して CPU 情報を取得します。ただし、CPU ノードの一部は、ホットリムーブ中に失われています。そのため、**kdump.service** の再読み込みと CPU **hot-removal** との間で競合状態が発生すると同時に、ダンプが失敗する場合があります。その後のクラッシュでは、**vmcore** ファイルがキャプチャーされない可能性があります。

この更新により、CPU ホットプラグ後に **kdump.service** を再読み込みする必要がなくなり、上述の例で **vmcore** キャプチャーが期待どおりに機能するようになりました。

注: この改善は、ファームウェア支援ダンプ (**fadump**) で期待どおりに機能します。スタンダード **kdump** の場合は、ホットプラグ の動作中に **kdump.service** の再読み込みが行われます。

(BZ#1922951)

kdumpctl コマンドが、新しい **kdumpctl estimate** ユーティリティーに対応するようになりました。

kdumpctl コマンドが、**kdumpctl estimate** ユーティリティーに対応するようになりました。既存の **kdump** 設定に基づいて、**kdumpctl estimate** は、**kdump** × モリー割り当てに適した推定値を出力します。

クラッシュカーネルの最小サイズは、ハードウェアおよびマシンの仕様により異なります。そこで、以前は、正確な **crashkernel=** 値を推定することが困難でした。

今回の更新で、**kdumpctl estimate** ユーティリティーが推定値を提供するようになりました。この値は、ベストエフォートの推奨予測値であり、実現可能な **crashkernel=** 値の設定に役立ちます。

(BZ#1879558)

IBM TSS 2.0 パッケージが 1.6.0 にリベースされました。

IBM の Trusted Computing Group(TCG)Software Stack(TSS)2.0 バイナリーパッケージが 1.6.0 にアップグレードされました。今回の更新で、AMD64 および Intel 64 のアーキテクチャーに IBM TSS 2.0 サポートが追加されました。

これは、Trusted Platform Modules (TPM) 2.0 のユーザー空間 TSS で、よりシンプルなインターフェイスで TCG TSS ワーキンググループの Enhanced System Application Interface (ESAPI)、System Application Interface (SAPI)、および TPM Command Transmission Interface (TCTI) API と同等の機能(ただし API 互換ではない)を実装します。

これは、アプリケーションおよびプラットフォームが TPM を共有してセキュアなアプリケーションに統合できるようにするセキュリティーミドルウェアです。

今回のリベースで、以前のバージョンに対する多くのバグ修正および機能拡張が追加されました。主な変更点には、以下の新しい属性が含まれます。

- **tsscertifyx509**: x509 証明書を検証します。
- **tssgetcryptolibrary**: 現在の暗号化ライブラリーを表示します。
- **tssprintattr**: TPM 属性をテキストとして出力
- **tsspublicname**: エンティティーのパブリック名を計算します。
- **tsssetcommandcodeauditstatus**: TPM2_SetCommandCodeAuditStatus によりコードを消去または設定します。
- **tsstpmcmd**: 帯域内 TPM シミュレーターシグナルを送信します。

(BZ#1822073)

schedutil CPU 周波数ガバナーが RHEL 8 で利用できるようになりました。

schedutil CPU ガバナーは、CPU スケジューラーで利用可能な CPU 使用率データを使用します。 **schedutil** は CPU スケジューラーの一部で、スケジューラーの内部データ構造に直接アクセスできます。 **schedutil** は、システムロードに応じて CPU の周波数をどのように上げ下げるかを制御します。 **schedutil** 周波数ガバナーは、デフォルトでは有効にならないため、手動で選択する必要があります。

CPU ごとに1つの **policyX** ディレクトリーが存在します。 **schedutil** は、カーネル内の既存の **CPUFreq** ガバナーの **policyX/scaling_governors** リストで利用でき、**/sys/devices/system/cpu/cpufreq/policyX** ポリシーに割り当てられます。ポリシーファイルは上書きして変更できます。

intel_pstate スケーリングドライバーを使用する場合は、**intel_pstate** が利用可能になり、ガバナーによりリスト表示されるように、**intel_pstate=passive** コマンドライン引数を設定する必要がある場合があります。 **intel_pstate** は、最新の CPU を備えた Intel ハードウェアにおけるデフォルトです。

(BZ#1938339)

rt-tests-2.1 アップストリームバージョンにリベースされた **rt-tests** スイート

rt-tests スイートが、**rt-tests-2.1** バージョンにリベースされ、バグ修正および機能拡張が数多く追加されました。以前のバージョンでの大きな変更には、以下のものが含まれます。

- **rt-tests** スイート内のさまざまなプログラムを修正します。
- 一般的なオプションセットとプログラムがより均一になるように修正されています。たとえば、**oslat** プログラムのオプション **-t --runtime** オプションの名前が **-D** に変更され、残りのスイートに合わせて実行時間を指定します。
- **json** 形式で出力する新機能を実装しました。

(BZ#1954387)

Intel® QuickAssist テクノロジーライブラリー (QATlib) がバージョン 21.05 にリベースされました。

qatlib パッケージがバージョン 21.05 にリベースされ、バグ修正および機能拡張が数多く追加されました。主な変更点は、以下のとおりです。

- 複数の暗号化アルゴリズムのサポートを追加しました。
 - AES-CCM 192/256
 - ChaCha20-Poly1305
 - PKE 8K (RSA, DH, ModExp, ModInv)
- 異なるノードでのデバイス列挙の修正
- 32 ビットビルドの **pci_vfio_set_command** の修正

QATlib のインストール方法は、[RHEL 8 で Intel®QuickAssist テクノロジースタックが正常に機能していることの確認](#) を参照してください。

(BZ#1920237)

4.9. ファイルシステムおよびストレージ

クオータタイプが複数指定されている場合に、**xfs_quota state** がすべての猶予時間を出力するようになりました。

xfs_quota state コマンドは、コマンドラインで指定された複数のクオータタイプの猶予時間を出力するようになりました。以前は、**-g**、**-p**、または **-u** が複数指定されていても、1つしか表示されていませんでした。

(BZ#1949743)

rpc.gssd デーモンに追加した **-H** オプションと、**/etc/nfs.conf** ファイルに追加した **set-home** オプション

このパッチでは、**-H** オプションが **rpc.gssd** に、**set-home** オプションが **/etc/nfs.conf** に追加されますが、デフォルトの挙動は変更しません。

デフォルトでは、**rpc.gssd** は、ユーザーのホームディレクトリーが Kerberos セキュリティーを持つ NFS 共有にある場合に発生する可能性があるデッドロックを回避するため、**\$HOME** を **/** に設定します。**-H** オプションが **rpc.gssd** に追加されているか、**set-home=0** が **/etc/nfs.conf** に追加されている場合、**rpc.gssd** は **\$HOME** を **/** に設定しません。

このオプションを使用すると、**\$HOME/.k5identity** で Kerberos k5ID ファイルを使用できます。また、NFS ホームディレクトリーが Kerberos セキュリティー機能の NFS 共有にないことを前提とします。このオプションは、k5ID ファイルが必要など、特定の環境でのみ使用できます。詳細は、man ページの **k5identity** を参照してください。

(BZ#1868087)

Storage RHEL システムロールが LVM VDO ボリュームをサポートするようになりました

Virtual Data Optimizer (VDO) は、ストレージボリュームの使用を最適化するのに役立ちます。今回の機能強化により、管理者は **storage** システムロールを使用して、論理ボリュームマネージャー (LVM) の VDO ボリュームで **圧縮** および **重複排除** を管理できるようになりました。

(BZ#1882475)

LVM system.devices ファイル機能が RHEL 8 で利用可能になりました

RHEL 8.5 では、LVM **system.devices** ファイル機能が導入されました

た。**/etc/lvm/devices/system.devices** ファイルにデバイスのリストを作成することにより、LVM が認識して使用する特定のデバイスを選択し、LVM が不要なデバイスを使用しないようにすることができます。

system.devices ファイル機能を有効にするには、**lvm.conf** 設定ファイルで **use_devicesfile=1** を設定し、**system.devices** ファイルにデバイスを追加します。**system.devices** ファイル機能が有効になっている場合、LVM はデバイスフィルター設定を無視します。警告メッセージが表示されないようにするには、**lvm.conf** ファイルからフィルター設定を削除します。

詳細は、**lvmdevices (8)** の man ページを参照してください。

(BZ#1922312)

quota が HPE XFS に対応

quota ユーティリティーは、HPE XFS ファイルシステムのサポートを提供するようになりました。その結果、HPE XFS のユーザーは、クォータ ユーティリティーを通じてユーザーおよびグループのディスク使用状況を監視および管理できるようになります。

(BZ#1945408)

4.10. 高可用性およびクラスター

pcs cluster setup コマンドのローカルモードバージョンが完全にサポートされるようになりました。

デフォルトでは、**pcs cluster setup** コマンドは、すべての設定ファイルをクラスターノードに自動的に同期します。RHEL 8.3 以降、**pcs cluster setup** コマンドは **--corosync-conf** オプションをテクノロジープレビューとして提供していました。この機能は、RHEL 8.5 で完全にサポートされるようになりました。このオプションを指定すると、コマンドが **local** モードに切り替わります。このモードでは、**pcs** コマンドラインインターフェイスは他のノードと通信せずに **corosync.conf** ファイルを作成し、ローカルノード上の指定されたファイルに保存します。これにより、スクリプトで **corosync.conf** ファイルを作成し、スクリプトでそのファイルを処理できます。

(BZ#1839637)

クラスターノードのサブセットでフェンシング用にウォッチドッグのみの SBD を設定する機能

以前のバージョンでは、ウォッチドッグのみの SBD 設定を使用するには、クラスター内のすべてのノードで SBD を使用する必要がありました。一部のノードはサポートしているが、他のノード（リモートノード）では他のフェンシングが必要なクラスターで SBD が使用できませんでした。ユーザーは、新しい **fence_watchdog** エージェントを使用して、ウォッチドッグのみの SBD 設定を設定できるようになりました。これにより、一部のノードのみがフェンシングにウォッチドッグのみの SBD を使用し、その他のノードが他のフェンシングタイプを使用するクラスター設定が可能になります。クラスターはこのようなデバイスを1つしか持たず、これは **watchdog** という名前にする必要があります。

(BZ#1443666)

その他のすべてのリソースを再起動せずに SCSI フェンスデバイスを更新する新たな **pcs** コマンド

pcs stonith update コマンドを使用して SCSI フェンスデバイスを更新すると、stonith リソースが実行されているのと同じノードで実行中の全リソースを再起動することになります。新しい **pcs stonith update-scsi-devices** コマンドを使用すると、他のクラスターリソースを再起動せずに SCSI デバイスを更新できます。

[\(BZ#1872378\)](#)

pcs resource safe-disable コマンドの新しい縮小出力表示オプション

pcs resource safe-disable コマンドおよび **pcs resource disable --safe** コマンドは、エラーレポートの後に長いシミュレーション結果を出力します。これらのコマンドに、エラーのみを出力する **--brief** を指定できるようになりました。エラーレポートには、影響を受けるリソースのリソース ID が常に含まれるようになりました。

[\(BZ#1909901\)](#)

pcs がロール名として Promoted および Unpromoted を受け入れるようになりました。

Pacemaker 設定でロールが設定される場合、**pcs** コマンドラインインターフェイスで **Promoted** および **Unpromoted** を受け入れるようになりました。これらのロール名は、**Master** および **Slave** Pacemaker ロールと機能的に同等です。**Master** および **Slave** は、設定表示とヘルプテキストでこれらのロールの名前のままでです。

[\(BZ#1885293\)](#)

新しい pcs リソースステータス表示コマンド

pcs resource status コマンドおよび **pcs stonith status** コマンドで、以下のオプションが使用できるようになりました。

- **pcs resource status node=node_id** コマンドおよび **pcs stonith status node=node_id** コマンドを使用すると、特定ノードに設定したリソースの状態を表示できます。これらのコマンドを使用すると、クラスターとリモートノードの両方でリソースのステータスを表示できます。
- **pcs resource status resource_id** コマンドおよび **pcs stonith status resource_id** コマンドを使用すると、1つのリソースの状況を表示できます。
- **pcs resource status tag_id** コマンドおよび **pcs stonith status tag_id** コマンドを使用すると、指定したタグで、すべてのリソースの状態を表示できます。

[\(BZ#1290830、BZ#1285269\)](#)

自動アクティベーションを制御する新しい LVM ボリュームグループフラグ

LVM ボリュームグループは、ボリュームグループから作成した論理ボリュームを起動時に自動的にアクティブにするかどうかを制御する **setautoactivation** フラグに対応するようになりました。クラスターで Pacemaker が管理するボリュームグループを作成する場合は、データの破損を防ぐために、ボリュームグループで **vgcreate --setautoactivation n** コマンドを実行して、このフラグを **n** に設定します。Pacemaker クラスターで使用される既存のボリュームグループがある場合は、**vgchange --setautoactivation n** でフラグを設定します。

[\(BZ#1899214\)](#)

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

nodejs:16 モジュールストリームに完全に対応しました。

以前はテクノロジープレビューとして利用できた **nodejs:16** モジュールストリームは、[RHSA-2021-5171](#) アドバイザリーのリリースで完全に対応しています。**nodejs:16** モジュールストリームでは、LTS (Long Term Support) バージョンの **Node.js 16.13.1** が提供されるようになりました。

RHEL 8.5 に含まれる **Node.js 16** は、RHEL 8.3 以降で利用可能な **Node.js 14** 上で、多くの新機能と、バグおよびセキュリティーの修正を提供します。

主な変更点は、以下のとおりです。

- **V8** エンジンがバージョン 9.4 にアップグレードされました。
- **npm** パッケージマネージャーがバージョン 8.1.2 にアップグレードされました。
- 新しい **Timers Promises** API は、**Promise** オブジェクトを返すタイマー関数の代替セットを提供します。
- **Node.js** が実験的な新しい **Web Streams** API を提供するようになりました。
- **Node.js** には、**Corepack** が含まれるようになりました。これは、手動でインストールすることなく、指定したプロジェクトに設定されたパッケージマネージャーを使用できるようにする試験的なツールです。
- **Node.js** は、ESM ローダーフック API を提供する試験的な ECMAScript モジュール (ESM) を提供します。これにより、ESM ローダーフックが統合されます。

nodejs:16 モジュールストリームをインストールするには、以下を使用します。

```
# yum module install nodejs:16
```

nodejs:14 ストリームからアップグレードする場合は、[Switching to a later stream](#) を参照してください。

(BZ#1953991、BZ#2027610)

新しいモジュールストリーム: **ruby:3.0**

RHEL 8.5 では、新しい **ruby:3.0** モジュールストリームに **Ruby 3.0.2** が導入されました。このバージョンでは、RHEL 8.3 で配布される **Ruby 2.7** に対するパフォーマンスの向上、バグおよびセキュリティー修正、および新機能が数多く追加されました。

主な機能拡張は、次のとおりです。

- 同時実行機能および並列処理機能:
 - スレッドセーフ実行を提供する Actor-model 抽象化である **Ractor** は実験的な機能として提供されます。
 - **Fiber Scheduler** が実験的な機能として導入されました。**Fiber Scheduler** はブロッキング操作をインターセプトし、既存のコードを変更せずに軽量の同時実行を可能にします。
- 静的な分析機能:
 - **Ruby** プログラムの構造を説明する **RBS** 言語が導入されました。**RBS** で書かれたタイプ定義を解析するために **rbs** gem が追加されました。
 - **TypeProf** ユーティリティーが導入されました。これは **Ruby** コードのタイプ解析ツールです。
- **case/in** 式を使用したパターンの照合は実験的なものではありません。

- 実験的な機能である 1 行パターンの一致が再設計されました。
- 検索パターンが実験的な機能として追加されました。

以下のパフォーマンスの向上が実装されています。

- Interactive Ruby Shell (IRB)** に長いコードを貼り付けることができるようになりました。
- measure** コマンドが、時間測定のために **IRB** に追加されました。

その他の主な変更点は次の通りです。

- キーワード引数は他の引数から分離されています。
- \$HOME/.gem/** ディレクトリーがすでに存在しない限り、ユーザーがインストールした gems のデフォルトディレクトリーは **\$HOME/.local/share/gem/** になります。

ruby:3.0 モジュール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 ruby:3.0
```

以前の **ruby** モジュールストリームからアップグレードするには、[Switching to a later stream](#) を参照してください。

(BZ#1938942)

Python **urllib** 解析関数のデフォルト区切り文字の変更点

Python **urllib** ライブラリーの [Web Cache Poisoning CVE-2021-23336](#) を緩和するため、**urllib.parse.parse_qs** および **urllib.parse.parse_qs** 関数のデフォルト区切り文字が、アンパサンド (&) とセミコロン (;) の両方が単一のアンパサンドに変更されます。

この変更は、RHEL 8.4 のリリースで Python 3.6 に実装され、Python 3.8 および Python 2.7 にバックポートされます。

デフォルトの区切り文字の変更は後方互換性がない可能性があるため、Red Hat はデフォルトの区切り文字が変更された Python パッケージの動作を設定する手段を提供します。さらに、影響を受ける **urllib** 解析関数は、お客様のアプリケーションが変更の影響を受けることを検知すると警告を発行します。

詳細は、[Mitigation of Web Cache Poisoning in the Python urllib library \(CVE-2021-23336\)](#) ナレッジ記事を参照してください。

Python 3.9 は影響を受けませんが、新しいデフォルトの区切り文字 (&) がすでに含まれているため、Python コードで **urllib.parse.parse_qs** および **urllib.parse.parse_qs** 関数を呼び出す時に、セパレーターパラメーターを渡すことしか変更できません。

(BZ#1935686、BZ#1931555、BZ#1969517)

Python **ipaddress** モジュールで、IPv4 アドレスでゼロを使用できなくなりました。

[CVE-2021-29921](#) を緩和するために、Python **ipaddress** モジュールが、**AddressValueError: Leading zeros** の先頭のゼロを持つ IPv4 アドレスを拒否するようになりました。

この変更は、**python38** および **python39** モジュールに導入されました。RHEL に同梱される以前の Python バージョンは、CVE-2021-29921 の影響を受けません。

以前の動作に依存するお客様は、IPv4 アドレス入力を事前に処理して、先頭のゼロオフを削除します。以下に例を示します。

```
>>> def reformat_ip(address): return '.'.join(part.lstrip('0') if part != '0' else part for part in address.split('.'))
>>> reformat_ip('0127.0.0.1')
'127.0.0.1'
```

読みやすために明示的なループで先頭のゼロを取り除くには、以下を使用します。

```
def reformat_ip(address):
    parts = []
    for part in address.split('.'):
        if part != "0":
            part = part.lstrip('0')
        parts.append(part)
    return '.'.join(parts)
```

(BZ#1986007、BZ#1970504、BZ#1970505)

php:7.4 モジュールストリームがバージョン 7.4.19 にリベース

php:7.4 モジュールストリームが提供する PHP スクリプト言語が、バージョン 7.4.6 から 7.4.19 にアップグレードされました。今回の更新で、複数のセキュリティー修正およびバグ修正が提供されます。

(BZ#1944110)

新しいパッケージ: pg_repack

postgresql:12 および **postgresql:13** モジュールストリームに、新しい **pg_repack** パッケージが追加されました。**pg_repack** パッケージは **PostgreSQL** 拡張を提供し、テーブルおよびインデックスから bloat を取り除き、必要に応じてクラスターインデックスの物理順序を復元します。

(BZ#1967193、BZ#1935889)

新しいモジュールストリーム: nginx:1.20

nginx 1.20 Web およびプロキシサーバーが **nginx:1.20** 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。この更新では、以前にリリースされたバージョン 1.18 に対するバグ修正、セキュリティ修正、新機能、および機能拡張が数多く提供されます。

新機能:

- **nginx** が、OCSP (Online Certificate Status Protocol) を使用したクライアント SSL 証明書の検証に対応するようになりました。
- **nginx** が、最小限の空き領域に基づくキャッシュクリアに対応するようになりました。これに対応するのは、**proxy_cache_path** ディレクティブの **min_free** パラメーターとして実装されています。
- 新しい **ngx_stream_set_module** モジュールが追加されました。これにより、変数の値を設定できるようになりました。

拡張されたディレクティブ:

- **ssl_conf_command**、**ssl_reject_handshake** など、新しいディレクティブが複数利用できるようになりました。
- **proxy_cookie_flags** ディレクティブが変数に対応するようになりました。

HTTP/2 のサポートが改善されました。

- **ngx_http_v2** モジュールに、**linger_close** ディレクティブ、**linger_time** ディレクティブ、**linger_timeout** ディレクティブが含まれるようになりました。
- HTTP/2 での接続の処理は、HTTP/1.x に合わせて行われました。**nginx 1.20** では、削除した **http2_recv_timeout** ディレクティブ、**http2_idle_timeout** ディレクティブ、および **http2_max_requests** ディレクティブの代わりに、**keepalive_timeout** ディレクティブおよび **keepalive_requests** ディレクティブを使用します。

nginx:1.20 ストリームをインストールするには、以下を実行します。

```
# yum module install nginx:1.20
```

nginx:1.20 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#1945671)

squid:4 モジュールストリームがバージョン 4.15 にリベース

squid:4 モジュールストリームで利用可能な **Squid** プロキシサーバーが、バージョン 4.11 からバージョン 4.15 にアップグレードされました。今回の更新で、さまざまなバグ修正およびセキュリティー修正が提供されます。

(BZ#1964384)

mutt がバージョン 2.0.7 にリベース

Mutt メールクライアントはバージョン 2.0.7 に更新され、機能強化およびバグ修正が数多く追加されました。

主な変更点は、以下のとおりです。

- **Mutt** は、**XOAUTH2** メカニズムを使用して **OAuth 2.0** 承認プロトコルをサポートするようになりました。Mutt は、IMAP、POP、および SMTP プロトコルの **OAUTHBEARER** 認証メカニズムもサポートするようになりました。OAuth ベースの機能は、外部スクリプトから提供されます。これにより、認証トークンを使用して **Gmail** などのさまざまなクラウドメールアドレスと **Mutt** を接続できます。OAuth サポートで **Mutt** を設定する方法の詳細は、[How to set up Mutt with Gmail using OAuth2 authentication](#) を参照してください。
- **Mutt** では、ドメインリテラルのメールアドレスがサポートされるようになりました(例: `user@[IPv6:fcXX:...]`)。
- 新しい **\$ssl_use_tlsv1_3** 設定変数により、TLS 1.3 接続がメールサーバーでサポートされる場合、この接続が可能になります。この変数はデフォルトで有効になっています。
- 新しい **\$imap_deflate** 変数により、**COMPRESS=DEFLATE** 圧縮がサポートされるようになりました。デフォルトでは、この変数は無効です。

- **\$ssl_starttls** 変数は、暗号化されていない IMAP **PREAUTH** 接続の中止を制御しなくなりました。**STARTTLS** プロセスに依存している場合は、代わりに **\$ssl_force_tls** 変数を使用してください。

新しい **Mutt** バージョンに更新した後でも、RHEL ユーザーが既存の環境で問題に遭遇するのを防ぐために、**ssl_force_tls** 設定変数はデフォルトで **no** になっていることに注意してください。**Mutt** のアップストリームバージョンでは、**ssl_force_tls** がデフォルトで有効になりました。

(BZ#1912614、BZ#1890084)

4.12. コンパイラーより開発ツール

Go Toolset がバージョン 1.16.7 にリベース

Go Toolset がバージョン 1.16.7 にアップグレードされました。主な変更点は、以下のとおりです。

- **GO111MODULE** 環境変数が、デフォルトで **on** に設定されるようになりました。この設定を元に戻すには、**GO111MODULE** を **auto** に変更します。
- Go リンカーでは、使用するリソースが少なくなり、コードの堅牢性および保守性が向上します。これは、サポートされているすべてのアーキテクチャーおよびオペレーティングシステムに該当します。
- 新しい **embed** パッケージを使用すると、プログラムのコンパイル中に埋め込みファイルにアクセスできます。
- **io/ioutil** パッケージのすべての機能が **io** パッケージおよび **os** パッケージに移動しました。**io/ioutil** を引き続き使用できますが、**io** および **os** パッケージはより適切な定義を提供します。
- Delve デバッガーが 1.6.0 にリベースされ、Go 1.16.7 Toolset に対応するようになりました。

詳細は、[Using Go Toolset](#) を参照してください。

(BZ#1938071)

Rust Toolset がバージョン 1.54.0 にリベース

Rust Toolset が、バージョン 1.54.0 に更新されました。主な変更点は、以下のとおりです。

- Rust 標準ライブラリーが **wasm32-unknown-unknown** ターゲットで利用できるようになりました。今回の機能拡張により、新たに安定化した組み込み関数を含む WebAssembly バイナリを生成できます。
- Rust には、アレイの **Intoliterator** 実装が含まれるようになりました。今回の機能拡張により、**Intoliterator** トレイトを使用して、配列を値ごとに反復し、配列をメソッドに渡すことができます。ただし、**array.into_iter()** は、Rust の 2021 版まで値を参照ごとに繰り返します。
- **or** パターンの構文により、パターンのどこにでもネスト化できるようになりました。(例: **Pattern(1)|Pattern(2)** ではなく **Pattern(1|2)**)。
- Unicode 識別子に、Unicode Standard Annex #31 で定義された有効な識別子文字がすべて含まれるようになりました。
- メソッドとトレイトの実装は安定化されました。

- 増分コンパイルはデフォルトで再度有効になります。

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

(BZ#1945805)

LLVM Toolset がバージョン 12.0.1 にリベース

LLVM Toolset がバージョン 12.0.1 にアップグレードされました。主な変更点は、以下のとおりです。

- 新しいコンパイラーフラグ **-march=x86-64-v[234]** が追加されました。
- Clang コンパイラーのコンパイラーフラグ **-fasynchronous-unwind-tables** が、Linux AArch64/PowerPC のデフォルトになりました。
- Clang コンパイラーは、C++20 可能性属性 `[[likely]]` および `[[unlikely]]` に対応するようになりました。
- 新しい関数属性 **tune-cpu** が追加されました。これにより、マイクロアーキテクチャーの最適化は **target-cpu** 属性または TargetMachine CPU とは独立して適用できます。
- セキュリティーを強化するために、新しいサニタイザー **-fsanitize=unsigned-shift-base** が整数サニタイザー **-fsanitize=integer** に追加されました。
- PowerPC ターゲットでのコード生成が最適化されました。
- WebAssembly バックエンドが LLVM で有効にされるようになりました。今回の機能拡張により、LLVM および Clang で WebAssembly バイナリーを生成できます。

詳細は、[LLVM Toolset の使用](#) を参照してください。

(BZ#1927937)

cmake がバージョン 3.20.2 にリベースされました。

Cmake が 3.18.2 から 3.20.2 にリベースされました。バージョン 3.20.2 以前を必要とするプロジェクトで CMake を使用するには、`cmake_minimum_required(version 3.20.2)` コマンドを使用します。

主な変更点は、以下のとおりです。

- C++23 コンパイラーモードは、ターゲットプロパティー **CXX_STANDARD**、**CUDA_STANDARD**、**OBJCXX_STANDARD**、またはコンパイル機能の **cxx_std_23** メタ機能を使用して指定できるようになりました。
- CUDA 言語サポートにより、NVIDIA CUDA コンパイラーをシンボリックリンクにすることができるようになりました。
- Intel oneAPI NextGen LLVM コンパイラーが、**IntelLLVM** コンパイラ ID でサポートされるようになりました。
- Cmake は、Android NDK のツールチェーンファイルとマージすることで、Android のクロスコンパイルを容易にします。
- cmake(1)** を実行してプロジェクトビルドシステムを生成すると、ハイフンで始まる不明なコマンドライン引数が拒否されるようになりました。

新機能および非推奨になった機能の詳細は、[CMake Release Notes](#) を参照してください。

(BZ#1957947)

新しい GCC Toolset 11

GCC Toolset 11 は最新バージョンの開発ツールを提供するコンパイラーツールセットです。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 Streams として利用できます。

GCC Toolset 10 以降、以下のコンポーネントがリベースされています。

- GCC のバージョンを 11.2 に変更
- GDB: バージョン 10.2 へ
- Valgrind: バージョン 3.17.0 へ
- SystemTap: バージョン 4.5 へ
- binutils のバージョンを 2.36 に変更
- elfutils をバージョン 0.185 に更新
- dwz: バージョン 0.14 へ
- Annobin をバージョン 9.85 へ

コンポーネントの完全なリストについては、[GCC Toolset 11](#) を参照してください。

GCC Toolset 11 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-11
```

GCC Toolset 11 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-11 tool
```

GCC Toolset バージョン 11 のツールバージョンが、このようなツールのシステムバージョンをオーバーライドするシェルセッションを実行するには、次のコマンドを実行します。

```
$ scl enable gcc-toolset-11 bash
```

詳細は、[Using GCC Toolset](#) を参照してください。

GCC Toolset 11 コンポーネントが、以下のコンテナーアイメージ 2 つでも利用可能になりました。

- GCC コンパイラ、GDB デバッガー、make 自動化ツールを含む **rhel8/gcc-toolset-11-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-11-perf-tools**

コンテナーア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>
```

GCC Toolset 11 コンテナイメージのみがサポートされるようになりました。以前のバージョンの GCC Toolset コンテナイメージが非推奨になりました。

(BZ#1953094)

.NET がバージョン 6.0 に更新されました。

Red Hat Enterprise Linux 8.5 には .NET バージョン 6.0 が同梱されています。以下は、主な改善点です。

- 64 ビット Arm (aarch64) に対応
- IBM Z および LinuxONE (s390x) に対応

詳細は、[.NET 6.0 RPM パッケージリリースノート](#) および [.NET 6.0 コンテナリリースノート](#) を参照してください。

(BZ#2022794)

GCC Toolset 11: GCC がバージョン 11.2 にリベースされました。

GCC Toolset 11 では、GCC パッケージがバージョン 11.2 に更新されました。主なバグ修正と機能拡張は、以下のとおりです。

一般的な改善

- GCC は、DWARF バージョン 5 のデバッグフォーマットをデフォルトで使用するようになりました。
- 診断で表示される列番号は、デフォルトでは実際の列番号を表し、複数列の文字を尊重します。
- 直線コードベクターライザは、機能全体を考慮してベクターリングを行います。
- 同じ変数を比較する一連の条件式は、それぞれに比較式が含まれていれば、switch ステートメントに変換することができます。
- プロセッサー間の最適化の改善:
 - **fipa-modref** オプションで制御される新しい IPA-modref パスは、関数呼び出しの副作用を追跡し、ポイントツー分析の精度を向上させます。
 - **fipa-icf** オプションで制御される同一コードのフォールディングパスが大幅に改善され、統一された関数の数が増え、コンパイル時のメモリー使用量が削減されました。
- リンクタイム最適化の改善:
 - リンク時のメモリー割り当てを改善し、ピークのメモリー使用量を削減しました。
- IDE で新しい **GCC_EXTRA_DIAGNOSTIC_OUTPUT** 環境変数を使用すると、ビルドフラグを調整することなく、機械的に読める修正のヒントを要求することができます。
- **-fanalyzer** オプションで実行されるスタティックアナライザが大幅に改善され、多数のバグフィックスと機能強化が行われました。

言語固有の改善

C ファミリー

- C および C++ コンパイラは、OpenMP 5.0 仕様の OpenMP コンストラクトおよびアロケータルーチンにおいて、非矩形のループネストをサポートしています。
- 属性:
 - 新しい **no_stack_protector** 属性は、スタック保護 (**-fstack-protector**) をかけてはいけない関数を示します。
 - 改良された **malloc** 属性は、アロケータとデアロケータの API ペアを識別するために使用することができます。
- 新しい警告:
 - **-Wsizeof-array-div(-Wall** オプションで有効) は、2つの **sizeof** 演算子の除算について、最初の演算子が配列に適用され、除算値が配列要素のサイズと一致しない場合に警告を発します。
 - **-Wstringop-overread** は、デフォルトで有効になっており、引数として渡された配列の最後を超えて読み取ろうとする文字列関数の呼び出しについて警告します。
- 警告の強化:
 - **-Wfree-nonheap-object** は、動的メモリー割り当て関数から返されていないポインターを使用した割り当て解除関数の呼び出しのインスタンスをより多く検出します。
 - **-Wmaybe-uninitialized** は、初期化されていないメモリーへのポインターや参照が、**const**-qualified 引数を取る関数に渡すことを診断します。
 - **-Wuninitialized** は、初期化されていない動的に割り当てられたメモリーからの読み取りを検出します。

C

- **-std=c2x** および **-std=gnu2x** オプションにより、ISO C 規格の次期 C2X 改訂版の新機能がサポートされています。以下に例を示します。
 - 標準属性がサポートされています。
 - **__has_c_attribute** プリプロセッサー演算子がサポートされています。
 - ラベルは、宣言の前や複合ステートメントの最後に表示されることがあります。

C++

- デフォルトのモードは **-std=gnu++17** に変更されます。
- C++ ライブラリーの **libstdc++** では、C++17 のサポートが改善されました。
- C++20 の新機能がいくつか実装されています。なお、C++20 のサポートは実験的なものです。各機能の詳細については、[C++20 Language Features](#) を参照してください。
- C++ フロントエンドは、今後予定されている C++23 ドラフト機能の一部を実験的にサポートしています。
- 新しい警告:

- **-Wctad-maybe-unsupported** はデフォルトでは無効で、控除ガイドのない型でクラステンプレート引数の控除を行うことについて警告します。
- **-Wrange-loop-construct** は、**-Wall** で有効になり、範囲ベースの for ループが不必要でリソース効率の悪いコピーを作成している場合に警告を発します。
- **-Wmismatched-new-delete** は **-Wall** で有効になり、不一致な形式の new 演算子や他の不一致な割り当て関数から返されたポインターを持つ delete 演算子の呼び出しについて警告します。
- **-Wvexing-parse** はデフォルトで有効になっており、最も厄介な構文解析ルールを警告します。つまり、宣言が変数定義のように見えても、C++言語では関数宣言として解釈される必要がある場合です。

アーキテクチャー固有の改善

64 ビット ARM アーキテクチャー

- Armv8-R アーキテクチャーは、**-march=armv8-r** オプションでサポートされています。
- GCC は、加算、減算、乗算、および複素数の累積と減算を行う演算を自動ベクトル化することができます。

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

- Sapphire Rapids、Alder Lake、および Rocket Lake のインテル製 CPU に対応しています。
- Intel AVX-VNNI の新しい ISA 拡張サポートが追加されました。**-mavxvnni** コンパイラースイッチは、AVX-VNNI の組込みを制御します。
- znver3 コアを搭載した AMD CPU は、新たな **-march=znver3** オプションによりサポートされます。
- [x86-64 psABI サプリメント](#) で定義されている 3 つのマイクロアーキテクチャレベルは、新しい **-march=x86-64-v2**、**-march=x86-64-v3**、および **-march=x86-64-v4** オプションでサポートされています。

(BZ#1946782)

GCC Toolset 11: dwz が DWARF 5 に対応

GCC Toolset 11 では、**dwz** ツールが DWARF Version 5 デバッグ形式に対応するようになりました。

(BZ#1948709)

GCC Toolset 11: GCC が AIA ユーザー割り込みをサポートしました。

GCC Toolset 11 では、GCC が Accelerator Interfacing Architecture(AIA) のユーザー割り込みをサポートするようになりました。

(BZ#1927516)

GCC Toolset 11: 汎用 SVE チューニングのデフォルトが改善されました。

GCC Toolset 11 では、64 ビットの ARM アーキテクチャーにおいて、一般的な SVE のチューニングのデフォルトが改善されました。

(BZ#1979715)

SystemTap がバージョン 4.5 にリベース

SystemTap パッケージがバージョン 4.5 に更新されました。主なバグ修正と機能拡張は、以下のとおりです。

- 32 ビット浮動小数点変数は自動的に倍精度変数に拡張され、その結果、**\$context** 変数として直接アクセスできます。
- 列挙の値は **\$context** 変数としてアクセスできます。
- BPF uconversions tapset は拡張され、**user_long_error()** など、ユーザー空間の値にアクセスするためのより多くの tapset 関数が含まれています。
- 大規模なサーバーで安定した操作を提供するために、同時実行制御が大幅に改善されました。

詳細は、アップストリームの [SystemTap 4.5 リリースノート](#) を参照してください。

(BZ#1933889)

elfutils がバージョン 0.185 にリベース

elfutils パッケージがバージョン 0.185 に更新されました。主なバグ修正と機能拡張は、以下のとおりです。

- eu-elflint** および **eu-readelf** ツールは、ELF セクションの **SHF_GNU_RETAIN** および **SHT_X86_64_UNWIND** フラグを認識して表示するようになりました。
- DEBUGINFOD SONAME** マクロが **debuginfod.h** に追加されました。このマクロは、**dlopen** 関数とともに使用して、**libdebuginfod.so** ライブラリーをアプリケーションから直接読み込むことができます。
- 新しい関数 **debuginfod_set_verbose_fd** が **debuginfod-client** ライブラリーに追加されました。この関数は、詳細な出力を別のファイルにリダイレクトすることで、**debuginfod_find_*** クエリー機能を強化します。
- DEBUGINFOD_VERBOSE** 環境変数を設定すると、**debuginfod** クライアントが接続しているサーバーおよびこれらのサーバーの HTTP 応答に関する詳細情報が表示されるようになりました。
- debuginfod** サーバーは、新しいスレッドビジーメトリックと、より詳細なエラーメトリックを提供します。これにより、**debuginfod** サーバー上で実行されるプロセスの検査が容易になります。
- libdw** ライブラリーは、**DW_FORM_indirect** の場所値を透過的に処理し、**dwarf_whatform** 関数が属性の実際の FORM を返すようになりました。
- ネットワークトラフィックを減らすために、**debuginfod-client** ライブラリーは負の結果をキャッシュに保存し、クライアントオブジェクトは既存の接続を再利用できます。

(BZ#1933890)

Valgrind がバージョン 3.17.0 にリベース

Valgrind パッケージがバージョン 3.17.0 に更新されました。主なバグ修正と機能拡張は、以下のとおりです。

- Valgrind は、DWARF バージョン 5 デバッグ形式を読み取ることができます。

- Valgrind は、**debuginfod** サーバーへのデバッグクエリーをサポートします。
- ARMv8.2 プロセッサー命令は、部分的にサポートされます。
- POWER10 プロセッサーにおける Power ISA v.3.1 命令は、一部サポートされます。
- IBM z14 プロセッサー命令がサポートされます。
- ほとんどの IBM z15 命令は、サポートされています。Valgrind ツールスイートは、IBM z15 プロセッサーの miscellaneous-instruction-extensions facility 3 と vector-enhancements facility 2 をサポートします。これにより、Valgrind は、GCC **-march=z15** でコンパイルしたプログラムを正しく実行し、パフォーマンスとデバッグのエクスペリエンスを向上させます。
- track-fds=yes オプションは -q (--quiet) を尊重し、デフォルトで標準ファイル記述子の **stdin**、**stdout**、および **stderr** を無視します。標準のファイル記述子を追跡するには、--track-fds=all オプションを使用します。
- DHAT ツールには、--mode=copy および --mode=ad-hoc の 2 つの新しい操作モードがあります。

(BZ#1933891)

Dyninst がバージョン 11.0.0 にリベース

Dyninst パッケージがバージョン 11.0.0 に更新されました。主なバグ修正と機能拡張は、以下のとおりです。

- debuginfod** サーバーのサポートと、個別の **debuginfo** ファイル取得のサポート。
- プロシージャリンクエージテーブル (PLT) スタブへの間接呼び出しの検出を改善しました。
- C++ の名前のデマングルを改善しました。
- コードエミット時のメモリリークを修正しました。

(BZ#1933893)

IBM POWER10 の GDB で DAWR 機能が改善

この機能拡張により、IBM POWER10 プロセッサーでは、新しいハードウェアオッチポイント機能が GDB で有効になりました。たとえば、DAWR/DAWRX レジスターの新しいセットが追加されました。

(BZ#1854784)

GCC Toolset 11: GDB がバージョン 10.2 にリベースされました。

GCC Toolset 11 では、GDB パッケージがバージョン 10.2 に更新されました。主なバグ修正と機能拡張は、以下のとおりです。

新機能

- マルチスレッドシンボルの読み込みは、この機能をサポートするアーキテクチャーでデフォルトで有効になっています。この変更により、多くのシンボルを持つプログラムのパフォーマンスが向上します。
- テキスト形式のユーザーインターフェイス (TUI) ウィンドウは水平方向に配置できます。
- GDB は、複数のターゲット接続を同時にデバッグしますが、このサポートは実験的なものであ

り、制限があります。たとえば、各 inferior を異なるマシンで実行する別のリモートサーバーに接続するか、1つの inferior を使用してローカルのネイティブプロセスまたは他のプロセスのデバッグを行うことができます。

新規コマンドおよび改善されたコマンド

- 新しい **tui new-layout name window weight [window weight...]** コマンドは、新しいテキストユーザーインターフェイス (TUI) レイアウトを作成します。また、レイアウト名と表示ウィンドウを指定することもできます。
- alias [-a] [--] alias = command [default-args]** コマンドが改善され、新規エイリアスの作成時にデフォルトの引数を指定できます。
- set exec-file-mismatch** コマンドおよび **show exec-file-mismatch** コマンドは、新しい **exec-file-mismatch** オプションを設定して表示します。GDB が実行中のプロセスに割り当てるとき、このオプションは、GDB がロードされた現在の実行可能ファイルと、プロセスの開始に使用される実行ファイルとの間に不一致を検出すると、GDB が反応するかを制御します。

Python API

- gdb.register_window_type** 関数は、Python で新しい TUI ウィンドウを実装します。
- 動的タイプをクエリーできるようになりました。**gdb.Type** クラスのインスタンスは、新しいブール値属性 **dynamic** を指定でき、**gdb.Type.sizeof** 属性には動的タイプの値 **None** を指定できます。**Type.fields()** が動的タイプのフィールドを返す場合、その **bitpos** 属性の値は **None** になります。
- 新しい **gdb.COMMAND_TUI** 定数は、コマンドの TUI ヘルプクラスのメンバーとして Python コマンドを登録します。
- 新しい **gdb.PendingFrame.architecture()** メソッドは、保留中のフレームのアーキテクチャーを取得します。
- 新しい **gdb.Architecture.registers** メソッドは、**gdb.RegisterDescriptorIterator** オブジェクトを返す **gdb.RegisterDescriptor** Iterator オブジェクトを返します。このようなオブジェクトはレジスターの値を提供しませんが、アーキテクチャーで利用可能なレジスターを理解するのに役立ちます。
- 新しい **gdb.Architecture.register_groups** メソッドは、**gdb.RegisterGroupIterator** オブジェクトを返す **gdb.RegisterGroup** Iterator オブジェクトを返します。このようなオブジェクトは、どの登録グループがアーキテクチャーで利用可能なかを理解するのに役立ちます。

(BZ#1954332)

GCC Toolset 11: SystemTap がバージョン 4.5 にリベース

GCC Toolset 11 では、SystemTap パッケージがバージョン 4.5 に更新されました。主なバグ修正と機能拡張は、以下のとおりです。

- 32 ビット浮動小数点変数は自動的に倍精度変数に拡張され、その結果、**\$context** 変数として直接アクセスできるようになりました。
- 列挙 の値は **\$context** 変数としてアクセスできるようになりました。
- BPF uconversions tapset は拡張され、**user_long_error()** など、ユーザー空間の値にアクセスするためのより多くの tapset 関数が含まれるようになりました。

- 大規模なサーバーで安定した操作を提供するために、同時実行制御が大幅に改善されました。

詳細は、アップストリームの [SystemTap 4.5 リリースノート](#) を参照してください。

(BZ#1957944)

GCC Toolset 11: **elfutils** がバージョン 0.185 にリベースされました。

GCC Toolset 11 では、**elfutils** パッケージがバージョン 0.185 に更新されました。主なバグ修正と機能拡張は、以下のとおりです。

- eu-elflint** および **eu-readelf** ツールは、ELF セクションの **SHF_GNU_RETAIN** および **SHT_X86_64_UNWIND** フラグを認識して表示するようになりました。
- DEBUGINFOD SONAME** マクロが **debuginfod.h** に追加されました。このマクロは、**dlopen** 関数とともに使用して、**libdebuginfod.so** ライブラリーをアプリケーションから直接読み込むことができます。
- 新しい関数 **debuginfod_set_verbose_fd** が **debuginfod-client** ライブラリーに追加されました。この関数は、詳細な出力を別のファイルにリダイレクトすることで、**debuginfod_find_*** クエリー機能を強化します。
- DEBUGINFOD_VERBOSE** 環境変数を設定すると、**debuginfod** クライアントが接続しているサーバーおよびこれらのサーバーの HTTP 応答に関する詳細情報が表示されるようになりました。
- debuginfod** サーバーは、新しいスレッドビジーメトリックと、より詳細なエラーメトリックを提供します。これにより、**debuginfod** サーバー上で実行されるプロセスの検査が容易になります。
- libdw** ライブラリーは、**DW_FORM_indirect** の場所値を透過的に処理し、**dwarf_whatform** 関数が属性の実際の FORM を返すようになりました。
- debuginfod-client** ライブラリーは負の結果をキャッシュに保存し、クライアントオブジェクトは既存の接続を再利用できるようになりました。これにより、ライブラリー利用時の不要なネットワークトラフィックを防ぐことができます。

(BZ#1957225)

GCC Toolset 11: Valgrind がバージョン 3.17.0 にリベースされました。

GCC Toolset 11 では、Valgrind パッケージがバージョン 3.17.0 に更新されました。主なバグ修正と機能拡張は、以下のとおりです。

- Valgrind は、DWARF バージョン 5 デバッグ形式を読み取ることができるようになりました。
- Valgrind は、**debuginfod** サーバーへのデバッグクエリーをサポートするようになりました。
- Valgrind は、ARMv8.2 プロセッサー命令を部分的にサポートするようになりました。
- Valgrind は、IBM z14 プロセッサー命令をサポートするようになりました。
- Valgrind は、POWER10 プロセッサーの Power ISA v.3.1 命令を部分的にサポートするようになりました。

- **--track-fds=yes** オプションは **-q (--quiet)** を尊重し、デフォルトで標準ファイル記述子の **stdin**、**stdout**、および **stderr** を無視するようになりました。標準のファイル記述子を追跡するには、**--track-fds=all** オプションを使用します。
- DHAT ツールは、**--mode=copy** および **--mode=ad-hoc** の 2 つの新しい操作モードを持つようになりました。

(BZ#1957226)

GCC Toolset 11: Dyninst がバージョン 11.0.0 にリベースされました。

GCC Toolset 11 では、Dyninst パッケージがバージョン 11.0.0 に更新されました。主なバグ修正と機能拡張は、以下のとおりです。

- **debuginfod** サーバーのサポートと、個別の **debuginfo** ファイル取得のサポート。
- プロシージャリンクエージテーブル (PLT) スタブへの間接呼び出しの検出を改善しました。
- C++ の名前のデマングルを改善しました。
- コードエミット時のメモリリークを修正しました。

(BZ#1957942)

Fujitsu A64FX の PAPI ライブラリーサポートを追加

Fujitsu A64FX の PAPI ライブラリのサポートが追加されました。この機能を使用すると、開発者はハードウェアの統計を収集できます。

(BZ#1908126)

PCP パッケージが 5.3.1 にリベース

PCP (Performance Co-Pilot) パッケージがバージョン 5.3.1 にリベースされました。このリリースには、バグ修正、機能拡張、および新機能が含まれます。主な変更点は、以下のとおりです。

- スケーラビリティが向上します。数百ものホストのパフォーマンスマトリックの集中ロギング (**pmlogger** フーム) およびパフォーマンスルールによる自動監視 (**pmie** フーム) がサポートされるようになりました。
- **pmproxy** サービスおよび **libpcp_web** API ライブラリのメモリリークを解決し、インストールメンテーションと新しいマトリックを **pmproxy** に追加しました。
- 過去のソケット統計情報用の新しい **pcp-ss** ツール。
- **pcp-htop** ツールの改善
- 有線の PCP プロトコルに拡張機能が追加され、より高い解像度のタイムスタンプをサポートされるようになりました。

(BZ#1922040)

grafana パッケージがバージョン 7.5.9 にリベース

grafana パッケージがバージョン 7.5.9 にリベースされました。主な変更点は、以下のとおりです。

- 新規時系列パネル (ベータ)

- 新規パイチャートパネル(ベータ)
- Loki のアラートサポート
- 複数の新しいクエリー変換

詳細は [What's New in Grafana v7.4](#) および [What's New in Grafana v7.5](#) を参照してください。

(BZ#1921191)

grafana-pcp パッケージが 3.1.0 にリベース

grafana-pcp パッケージがバージョン 3.1.0 にリベースされました。主な変更点は、以下のとおりです。

- Performance Co-Pilot(PCP)Vector Checklist ダッシュボードは、新しい時系列パネルを使用し、グラフに単位を表示し、更新されたヘルプテキストが含まれます。
- PCP Vector Host Overview および PCP Checklist ダッシュボードに **pmproxy** URL および **hostspec** 変数を追加します。
- すべてのダッシュボードでデータソースの選択を表示します。
- 含まれているすべてのダッシュボードを読み取り専用としてマークします。
- Grafana 8 との互換性の追加

(BZ#1921190)

grafana-container がバージョン 7.5.9 にリベース

rhel8/grafana コンテナイメージは Grafana を提供します。主な変更点は、以下のとおりです。

- grafana** パッケージがバージョン 7.5.9 に更新されました。
- grafana-pcp** パッケージがバージョン 3.1.0 に更新されました。
- コンテナーが、コンテナーの起動時にカスタム Grafana プラグインをインストールするために **GF_INSTALL_PLUGINS** 環境変数をサポートするようになりました。

リベースは、Red Hat コンテナレジストリーの **rhel8/grafana** イメージを更新します。

このコンテナイメージをプルするには、以下のコマンドを実行します。

```
# podman pull registry.redhat.io/rhel8/grafana
```

(BZ#1971557)

pcp-container がバージョン 5.3.1 にリベース

rhel8/pcp コンテナイメージは、Performance Co-Pilot を提供します。 **pcp-container** パッケージがバージョン 5.3.1 にアップグレードされました。主な変更点は、以下のとおりです。

- pcp** パッケージが、バージョン 5.3.1 に更新されました。

リベースは、Red Hat コンテナレジストリーの **rhel8/pcp** イメージを更新します。

このコンテナイメージをプルするには、以下のコマンドを実行します。

```
# podman pull registry.redhat.io/rhel8/pcp
```

(BZ#1974912)

新しい pcp-ss PCP ユーティリティーが利用可能に

pcp-ss PCP ユーティリティーは、**pmdasockets(1)** PMDA が収集したソケット統計を報告します。このコマンドは、多くの **ss** コマンドラインオプションとレポート形式と互換性があります。また、ライブモードのローカルまたはリモート監視と、以前に記録された PCP アーカイブからの過去の再生の利点も提供します。

(BZ#1879350)

PCP で電力消費メトリックが利用可能に

新しい **pmda-denki** Performance Metrics Domain Agent (PMDA) は、消費電力に関連するメトリックを報告します。具体的には以下の項目が報告されます。

- 最近のインテル製 CPU で利用可能な RAPL(Running Average Power Limit) 測定値に基づく消費メトリック
- バッテリーを搭載したシステムで利用可能な、バッテリー放電に基づく消費メトリック

(BZ#1629455)

4.13. IDENTITY MANAGEMENT

IdM は新しいパスワードポリシーオプションをサポートするように

今回の更新で、Identity Management(IdM) が追加の **libpwquality** ライブラリオプションに対応するようになりました。

--maxrepeat

同じ文字の最大数を連続して指定します。

--maxsequence

単調な文字シーケンスの最大長を指定します (abcd)。

--dictcheck

パスワードが辞書の単語であるかどうかを確認します。

--usercheck

パスワードにユーザー名が含まれるかどうかを確認します。

ipa pwpolicy-mod を使用して、このオプションを適用します。たとえば、マネージャーグループのユーザーが提案するすべての新しいパスワードに、ユーザー名の確認を適用します。

```
*$ ipa pwpolicy-mod --usercheck=True managers*
```

新しいパスワードポリシーオプションのいずれかが設定されている場合、**--minlength** オプションの値に関係なく、パスワードの最小長は 6 文字になります。新しいパスワードポリシー設定は、新しいパスワードにのみ適用されます。

RHEL 7 サーバーと RHEL 8 サーバーが混在する環境では、新しいパスワードポリシー設定は、RHEL

8.4 以降で実行されているサーバーのみに適用されます。ユーザーが IdM クライアントにログインし、IdM クライアントが RHEL 8.3 以前で実行している IdM サーバーと通信している場合は、システム管理者が設定した新しいパスワードポリシーの要件は適用されません。一貫した動作を確保するには、すべてのサーバーを RHEL 8.4 以降にアップグレードまたは更新します。

(JIRA:RHELPLAN-89566)

各要求に一意の ID タグを追加して、SSSD デバッグのロギングを改善

SSSD は非同期的に要求を処理するため、異なるリクエストからのメッセージが同じログファイルに追加されるため、バックエンドログの個々のリクエストのログエントリーを追跡するのは容易ではありません。デバッグルогの可読性を向上させるために、ログメッセージに **RID#<integer>** の形式で一意のリクエスト ID が追加されました。これにより、個々の要求に関連するログを分離でき、複数の SSSD コンポーネントからのログファイル全体でリクエストを最初から最後まで追跡できます。

たとえば、SSSD ログファイルからの以下の出力例は、2 つの異なる要求について一意の識別子の RID#3 および RID#4 を示しています。

```
(2021-07-26 18:26:37): [be[testidm.com]] [dp_req_destructor] (0x0400): RID#3 Number of active DP
request: 0
(2021-07-26 18:26:37): [be[testidm.com]] [dp_req_reply_std] (0x1000): RID#3 DP Request
AccountDomain #3: Returning [Internal Error]: 3,1432158301,GetAccountDomain() not supported
(2021-07-26 18:26:37): [be[testidm.com]] [dp_attach_req] (0x0400): RID#4 DP Request Account #4:
REQ_TRACE: New request. sssd.nss CID #1 Flags [0x0001].
(2021-07-26 18:26:37): [be[testidm.com]] [dp_attach_req] (0x0400): RID#4 Number of active DP
request: 1
```

(JIRA:RHELPLAN-92473)

IdM が、automember および server Ansible モジュールに対応可能に

今回の更新で、**ansible-freeipa** パッケージに **ipaautomember** モジュールおよび **ipaserver** モジュールが含まれるようになりました。

- **ipaautomember** を使用して、自動メンバールールおよび条件の追加、削除、および変更を行うことができます。その結果、条件を満たす今後の IdM ユーザーおよびホストが、IdM グループに自動的に割り当てられます。
- **ipaserver** モジュールを使用すると、IdM トポロジーでサーバーの有無に関わらず、さまざまなパラメーターを保証できます。また、レプリカを非表示にしたり、表示したりすることもできます。

(JIRA:RHELPLAN-96640)

IdM パフォーマンスベースライン

今回の更新により、4 つの CPU と 8 GB の RAM を搭載した RHEL 8.5 の IdM サーバーで、130 の IdM クライアントを同時に登録できることが確認されました。

(JIRA:RHELPLAN-97145)

SSSD の Kerberos キャッシュのパフォーマンスが改善されました。

システムセキュリティーサービスデーモン (SSSD) の Kerberos Cache Manager (KCM) サービスに、新しいオペレーション **KCM_GET_CRED_LIST** が追加されました。この機能強化により、クレデンシャルキャッシュを繰り返し使用する際の入出力操作の回数が減り、KCM のパフォーマンスが向上しました。

(BZ#1956388)

SSSD がデフォルトでバックトレースをログ記録するようになる

この改善により、SSSD は詳細なデバッグルогをメモリー内のバッファーに保存し、障害発生時にログファイルに追加できるようになりました。デフォルトでは、以下のエラーレベルが原因でバックトレースが発生します。

- レベル 0: 致命的な障害
- レベル 1: 重大な障害
- レベル 2: 重大な障害

この動作は、**sssd.conf** 設定ファイルの対応するセクションにある **debug_level** オプションを設定することで、SSSD プロセスごとに変更できます。

- デバッグルогを 0 に設定すると、レベル 0 のイベントのみがバックトレースをトリガーします。
- デバッグルогを 1 に設定すると、レベル 0 と 1 でバックトレースが発生します。
- デバッグルогを 2 以上に設定すると、レベル 0 から 2 のイベントでバックトレースが発生します。

sssd.conf の対応するセクションで **debug_backtrace_enabled** オプションを **false** に設定することで、SSSD プロセスごとにこの機能を無効にできます。

```
[sssd]
debug_backtrace_enabled = true
debug_level=0
...
[nss]
debug_backtrace_enabled = false
...
[domain/idm.example.com]
debug_backtrace_enabled = true
debug_level=2
...
...
```

(BZ#1949149)

SSSD KCM がチケットを付与するチケットの自動更新に対応しました

この機能拡張により、システムセキュリティーサービスデーモン (SSSD) の Kerberos Cache Manager(KCM) サービスが、Identity Management(IdM) サーバー上の KCM クレデンシャルキャッシュに保存されているチケットを付与するチケット (TGT) を自動更新するように設定できるようになりました。更新は、チケットの有効期間の半分に達したときにのみ行われます。自動更新を使用するためには、IdM サーバー上のキー配布センター (KDC) が、更新可能な Kerberos チケットをサポートするように設定されている必要があります。

/etc/sssd/sssd.conf ファイルの [kcm] セクションを変更することで、TGT の自動更新を有効にすることができます。例えば、ファイルに以下のオプションを追加することで、60 分ごとに KCM に保存された

更新可能な TGT をチェックし、チケットの有効期間の半分に達した場合に自動更新を試みるよう SSSD を設定できます。

```
[kcm]
tgt_renewal = true
krb5_renew_interval = 60m
```

既存のドメインから更新用の **krb5** オプションを継承するように SSSD を設定することもできます。

```
[kcm]
tgt_renewal = true
tgt_renewal_inherit = domain-name
```

詳細については、**sssd-kcm** の man ページの **Renewals** セクションを参照してください。

([BZ#1627112](#))

Samba がバージョン 4.14.4 にリベース

The `_samba_` packages have been upgraded to upstream version 4.14.4, which provides bug fixes and enhancements over the previous version:

- Active Directory(AD) の公開プリンターの信頼性が向上し、AD で公開されている情報に新たなプリンター機能が追加されました。また、Samba は ARM64 アーキテクチャーの Windows ドライバーに対応するようになりました。
- ctdb isnotrecmaster** コマンドが削除されました。別の方法としては、**ctdb pnn** または **ctdb recmaster** コマンドを使用します。
- クラスター化されたトリビアルデータベース (CTDB) の **ctdb natgw master** および **slave-only** パラメーターの名前が **ctdb natgw leader** および **follower-only** に変更されました。

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

Samba を更新したら、**testparm** ユーティリティーを使用して `/etc/samba/smb.conf` ファイルを確認します。

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。

([BZ#1944657](#))

dnaInterval 設定属性がサポートされるようになりました。

今回の更新で、Red Hat Directory Server は、**cn=<DNA_config_entry>,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config** エントリーで Distributed Numeric Assignment (DNA) プラグインの **dnaInterval** 属性の設定をサポートするようになりました。DNA プラグインは、指定された属性に一意の値を生成します。レプリケーション環境では、サーバーは同じ範囲を共有できます。異なるサーバーで重複を回避するために、**dnaInterval** 属性を設定して一部の値をスキップできます。たとえば、間隔が **3** で、範囲の最初の番号が **1** の場合、範囲で使用される次の数字は **4**、続いて **7**、次が **10** になります。

詳細は **dnaInterval** パラメーターの説明を参照してください。

(BZ#1938239)

Directory Server がバージョン 1.4.3.27 にリベース

389-ds-base パッケージがアップストリームバージョン 1.4.3.27 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。

- <https://directory.fedoraproject.org/docs/389ds/releases/release-1-4-3-24.html>
- <https://directory.fedoraproject.org/docs/389ds/releases/release-1-4-3-23.html>
- <https://directory.fedoraproject.org/docs/389ds/releases/release-1-4-3-22.html>
- <https://directory.fedoraproject.org/docs/389ds/releases/release-1-4-3-21.html>
- <https://directory.fedoraproject.org/docs/389ds/releases/release-1-4-3-20.html>
- <https://directory.fedoraproject.org/docs/389ds/releases/release-1-4-3-19.html>
- <https://directory.fedoraproject.org/docs/389ds/releases/release-1-4-3-18.html>
- <https://directory.fedoraproject.org/docs/389ds/releases/release-1-4-3-17.html>

(BZ#1947044)

Directory Server が一時パスワードに対応

今回の機能拡張により、管理者はグローバルおよびローカルパスワードポリシーで一時パスワードルールを設定できるようになりました。これらのルールを使用すると、管理者がユーザーのパスワードをリセットすると、パスワードは一時的で、特定時間と定義された試行回数にのみ有効となるように設定できます。さらに、管理者がパスワードを変更した時点から有効期限が開始されないように設定できます。これにより、Directory Server では、一時パスワードを使用して、決められた時間または試行回数のみユーザーを認証できます。ユーザーが認証に成功すると、Directory Server により、このユーザーはパスワードの変更のみが許可されます。

(BZ#1626633)

IdM KDC は、セキュリティを強化するために PAC 情報を含む Kerberos チケットを発行するようになりました

今回の更新により、セキュリティを強化するために、RHEL Identity Management (IdM) は、新しいデプロイメントでデフォルトで特権属性証明書 (PAC) 情報を含む Kerberos チケットを発行するようになりました。PAC には、セキュリティ識別子 (SID)、グループメンバーシップ、ホームディレクトリー情報など、Kerberos プリンシパルに関する豊富な情報が含まれています。その結果、Kerberos チケットは悪意のあるサーバーによる操作の影響を受けにくくなります。

Microsoft Active Directory (AD) がデフォルトで使用する SID は、再利用されることのないグローバルに一意の識別子です。SID は複数の名前空間を表します。各ドメインには、各オブジェクトの SID の接頭辞である SID があります。

RHEL 8.5 以降、IdM サーバーまたはレプリカをインストールすると、インストールスクリプトはデフォルトでユーザーおよびグループの SID を生成します。これにより、IdM が PAC データを操作できるようになります。RHEL 8.5 より前に IdM をインストールし、AD ドメインとの信頼を設定していない場合は、IdM オブジェクトの SID が生成されていない可能性があります。IdM オブジェクトの SID の生成に関する詳細は、[IdM でのセキュリティ識別子 \(SID\) の有効化](#) を参照してください。

Kerberos チケットの PAC 情報を評価することで、リソースアクセスをより詳細に制御できます。たと

えば、あるドメインの **Administrator** アカウントは、他のドメインの **Administrator** アカウントとは一意に異なる SID を持っています。AD ドメインへの信頼がある IdM 環境では、UID が 0 のすべての Linux **root** アカウントなど、さまざまな場所で繰り返される可能性のある単純なユーザー名や UID ではなく、グローバルに一意の SID に基づいてアクセス制御を設定できます。

(Jira:RHELPLAN-159143)

Directory Server は、ロックの枯渇によって生じるデータベースの破損を防ぐ監視設定を提供します。

今回の更新で、**nsslapd-db-locks-monitoring-enable** パラメーターが **cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config** エントリーに追加されています。有効になっている場合 (デフォルト)、アクティブなデータベースロックの数が **nsslapd-db-locks-monitoring-threshold** で設定されているパーセンテージしきい値よりも大きい場合、**DirectoryServer** はすべての検索を中止します。問題が発生した場合、管理者は **cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config** エントリーの **nsslapd-db-locks** パラメーターで、データベースロックの数を増やすことができます。これにより、データの破損を防ぐことができます。さらに、管理者はスレッドがチェック間でスリープする間隔をミリ秒単位で設定できるようになりました。

詳細は、[Red Hat Directory Server 設定、コマンド、およびファイルリファレンス](#) のパラメーターの説明を参照してください。

(BZ#1812286)

Directory Server は、レトロエンジログデータベースから属性と接尾辞を除外できます。

今回の機能拡張で、**nsslapd-exclude-attrs** および **nsslapd-exclude-suffix** パラメーターが **Directory Server** に追加されました。これらのパラメーターを **cn=Retro Changelog Plugin,cn=plugins,cn=config** エントリーに設定し、レトロエンジログデータベースから特定の属性または接尾辞を除外できます。

(BZ#1850664)

Directory Server で **entryUUID** 属性のサポート

今回の機能拡張により、**Directory Server** は、[RFC 4530](#) に準拠するように **entryUUID** 属性をサポートするようになりました。たとえば、**entryUUID** サポートにより、OpenLDAP からの移行が容易になります。デフォルトでは、**Directory Server** は **entryUUID** 属性を新規エントリーのみに追加します。既存のエントリーに手動で追加するには、**dsconf <instance_name> plugin entryuuid fixup** コマンドを使用します。

(BZ#1944494)

nsSSLPersonalitySSL の設定に役立つ新しいメッセージを追加

これまででは、TLS 証明書のニックネームが設定パラメーター **nsSSLPersonalitySSL** の値と一致しない場合に、RHDS インスタンスの起動に失敗することが多くありました。この不一致は、お客様が以前のインスタンスから NSS DB をコピーしたり、証明書のデータをエクスポートしたりした際に、**nsSSLPersonalitySSL** の値を適切に設定しなかった場合に発生しました。今回の更新では、**nsSSLPersonalitySSL** を正しく設定するための追加メッセージがログに記録されます。

(BZ#1895460)

4.14. デスクトップ

ログイン画面でネットワークに接続できるようになりました。

今回の更新で、GNOME Display Manager (GDM) ログイン画面で、ネットワークに接続し、特定のネットワークオプションを設定できるようになりました。その結果、ホームディレクトリーがリモートサーバに保存されているエンタープライズユーザーとしてログインできます。

ログイン画面では、次のネットワークオプションに対応しています。

- 有線ネットワーク
- ワイヤレスネットワーク (パスワードで保護されているネットワークを含む)
- 仮想プライベートネットワーク (VPN)

ログイン画面では、追加のネットワーク設定の画面を開くことができません。このため、ログイン画面では次のネットワークオプションを使用できません。

- キャプティブポータルを開くネットワーク
- モデム接続
- エンタープライズの WPA または WPA2 暗号を使用した、事前設定されていないワイヤレスネットワーク

ログイン画面のネットワークオプションは、デフォルトでは無効になっています。ネットワーク設定を有効にするには、以下の手順を使用します。

1. 以下の内容で `/etc/polkit-1/rules.d/org.gnome.gdm.rules` を作成します。

```
polkit.addRule(function(action, subject) {
    if (action.id == "org.freedesktop.NetworkManager.network-control" &&
        subject.user == "gdm") {
        return polkit.Result.YES;
    }

    return polkit.Result.NOT_HANDLED;
});
```

2. GDM を再起動します。

```
# systemctl restart gdm
```



警告

GDM を再起動すると、グラフィカルユーザーセッションがすべて終了します。

3. ログイン画面で、トップパネルの右側にあるメニューのネットワーク設定にアクセスします。

(BZ#1935261)

ログイン時のシステムセキュリティ一分類の表示

GNOME Display Manager(GDM) ログイン画面を設定して、事前定義済みメッセージを含むオーバーレイバナーを表示できるようになりました。これは、ログイン前にユーザーがシステムのセキュリティ一分類を読み取る必要があるデプロイメントに役立ちます。

オーバーレイバナーを有効にし、セキュリティ一分類メッセージを設定するには、以下の手順に従います。

1. **gnome-shell-extension-heads-up-display** パッケージをインストールします。

```
# yum install gnome-shell-extension-heads-up-display
```

2. 以下の内容で **/etc/dconf/db/gdm.d/99-hud-message** ファイルを作成します。

```
[org/gnome/shell]
enabled-extensions=['heads-up-display@gnome-shell-extensions.gcampax.github.com']

[org/gnome/shell/extensions/heads-up-display]
message-heading="Security classification title"
message-body="Security classification description"
```

以下の値を、システムのセキュリティ一分類を記述するテキストに置き換えます。

Security classification title

セキュリティ一分類を示す簡単な見出し。

Security classification description

さまざまなガイドラインへの参照など、追加情報を提供する長いメッセージ。

3. **dconf** データベースを更新します。

```
# dconf update
```

4. システムを再起動します。

(BZ#1651378)

Flicker free boot が利用可能

これで、システムでフリッカーフリー起動を有効にできるようになります。flicker free boot が有効になると、システムの起動プロセス中にグラフィカルな急激な変化がなくなり、システムの起動時にディスプレイの表示がすぐに消えることはありません。

フリッカーフリー起動を有効にするには、以下の手順を使用します。

1. ブートローダーメニューをデフォルトで非表示に設定します。

```
# grub2-editenv - set menu_auto_hide=1
```

2. ブートローダー設定を更新します。

- UEFI システムの場合:

```
# grub2-mkconfig -o /etc/grub2-efi.cfg
```

- レガシー BIOS システムの場合:

```
# grub2-mkconfig -o /etc/grub2.cfg
```

3. システムを再起動します。

その結果、システムの起動時にブートローダーメニューが表示されず、ブートプロセスがグラフィカルにスムーズになります。

ブートローダーメニューを表示するには、システムの電源を入れてから **Esc** を繰り返し押します。

(JIRA:RHELPLAN-99148)

絵文字のサポートを更新

このリリースでは、Unicode の絵文字のサポートを、絵文字規格のバージョン 11 からバージョン 13 に更新しています。その結果、RHEL ではより多くの絵文字を使用できるようになりました。

絵文字機能を提供する以下のパッケージがリベースされました。

パッケージ	以前のバージョン	リベース後のバージョン
cldr-emoji-annotation	33.1.0	38
google-noto-emoji-fonts	20180508	20200723
unicode-emoji	10.90.20180207	13.0

(JIRA:RHELPLAN-61867)

すべてのユーザーにデフォルトのデスクトップセッションを設定できます。

今回の更新で、ログインしていないすべてのユーザーに対して事前に選択されているデフォルトのデスクトップセッションを設定できるようになりました。

ユーザーがデフォルト以外のセッションを使用してログインしても、選択した内容は次回のログイン時に持続します。

デフォルトのセッションを設定するには、以下の手順を使用します。

1. 設定ファイルテンプレートをコピーします。

```
# cp /usr/share/accountsservice/user-templates/standard \
/etc/accountsservice/user-templates/standard
```

2. 新しい **/etc/accountsservice/user-templates/standard** を編集します。 **Session=gnome** の行で、**gnome** を、デフォルトとして設定するセッションに置き換えます。
3. オプションです。特定のユーザーのデフォルトセッションの例外を設定するには、以下の手順に従います。

- a. テンプレートファイルを **/var/lib/AccountsService/users/user-name** にコピーします。

```
# cp /usr/share/accountsservice/user-templates/standard \
/var/lib/AccountsService/users/user-name
```

- b. 新しいファイルで、 `${USER}` や `${ID}` などの変数を、ユーザーの値に置き換えてください。
- c. **Session** の値を編集します。

(BZ#1812788)

4.15. グラフィックインフラストラクチャー

新しい GPU のサポート

以下の新しい GPU に対応しました。

Intel グラフィック:

- Alder Lake-S (ADL-S)

Alder Lake-S グラフィックのサポートはデフォルトで無効になっています。これを有効にするには、カーネルコマンドラインに次のオプションを追加します。

i915.force_probe=PCI_ID

`PCI_ID` を、Intel GPU の PCI デバイス ID に置き換えるか、* 文字に置き換えて、**i915** ドライバーを使用するすべてのアルファベット品質のハードウェアに対応できるようにします。

- Elkhart Lake (EHL)
- TGP プラットフォームコントローラーハブ (PCH) を使用した Comet Lake Refresh (CML-R)

AMD グラフィック:

- Cezzane と Barcelo
- Sienna Cichlid
- Dimgrey Cavefish

(JIRA:RHELPLAN-99040, BZ#1784132, BZ#1784136, BZ#1838558)

Wayland セッションは、プロプライエタリーの NVIDIA ドライバーで利用できます。

独自仕様の NVIDIA ドライバーが、Xwayland でのハードウェアアクセラレーションされた OpenGL および Vulkan レンダリングに対応するようになりました。その結果、独自の NVIDIA ドライバーで、GNOME Wayland セッションを有効にできるようになりました。以前は、ドライバーで利用できるのは従来の X11 セッションのみでした。以前のバージョンの RHEL から更新する場合に中断が起こらないように、X11 はデフォルトセッションのままになります。

NVIDIA 所有者ドライバーで Wayland を有効にするには、以下の手順を使用します。

1. カーネルコマンドラインに次のオプションを追加して、DRM (Direct Rendering Manager) カーネルモード設定を有効にします。

nvidia-drm.modeset=1

カーネルオプションの有効化の詳細は、[カーネルコマンドラインパラメーターの設定](#) を参照してください。

2. システムを再起動します。
Wayland セッションがログイン画面で利用できるようになりました。
3. (必要に応じて) システムのサスPENDまたは休止時にビデオ割り当てが失われないようにするには、ドライバーで電源管理オプションを有効にします。詳しくは [電源管理サポートの設定](#) を参照してください。

プロプライエタリーの NVIDIA ドライバーで DRM カーネルモード設定の使用に関連する制限は、[Direct Rendering Manager Kernel Modesetting \(DRM KMS\)](#) を参照してください。

(JIRA:RHELPLAN-99049)

GPU サポートの改善

これで、次の新しい GPU 機能が有効になります。

- PSR (Panel Self Refresh) が、Intel Tiger Lake 以降のグラフィックで有効になり、消費電力が改善されます。
- Intel Tiger Lake、Ice Lake、およびその後続のグラフィックでは、DP-MST (DisplayPort Multi-Stream Transport) 伝送方法で High Bit Rate 3 (HBR3) モードを使用できるようになりました。これにより、docks で特定の表示機能に対応できるようになります。
- Modesetting が、NVIDIA Ampere GPU で有効になりました。これには、ハイブリッドグラフィックスシステムを含む GA102、GA104、および GA107 のモデルが含まれます。
- Intel 統合グラフィックスと NVIDIA Ampere GPU を搭載したほとんどのラップトップが、いずれかの GPU を使用して外部ディスプレイに出力できるようになりました。

(JIRA:RHELPLAN-99043)

更新されたグラフィックドライバー

以下のグラフィックドライバーが更新されました。

- **amdgpu**
- **ast**
- **i915**
- **mgag2000**
- **nouveau**
- **vmwgfx**
- **vmwgfx**
- Mesa ライブラリー
- Vulkan パッケージ

(JIRA:RHELPLAN-99044)

Intel Tiger Lake グラフィックスに完全対応

以前はテクノロジープレビューとして利用できた Intel Tiger Lake UP3 および UP4 Xe グラフィックスに完全に対応するようになりました。ハードウェアアクセラレーションは、これらの GPU でデフォルトで有効にされています。

(BZ#1783396)

4.16. RED HAT ENTERPRISE LINUX システムロール

timesync_max_distance パラメーターを使用して、ルートの最大距離を設定できます。

今回の更新で、**timesync** RHEL システムロールが、新しい **timesync_max_distance** パラメーターを使用して、**ntpd** の **tos maxdist** および **chronyd** サービスの **maxdistance** パラメーターを設定できるようになりました。**timesync_max_distance** パラメーターは、Network Time Protocol(NTP) サーバーから測定値を受け入れる最大ルート距離を設定します。デフォルト値は 0 で、プロバイダー固有のデフォルト値を保持します。

(BZ#1938016)

Elasticsearch がサーバーのリストを受け入れることができる

以前は、Logging RHEL システムロールの Elasticsearch 出力の **server_host** パラメーターは、1つのホストの文字列値しか受け付けませんでした。今回の機能拡張により、複数のホストをサポートするために文字列のリストも受け入れるようになりました。その結果、1つの Elasticsearch 出力ディクショナリーで複数の Elasticsearch ホストを設定できるようになりました。

(BZ#1986463)

timesync RHEL システムロールに Network Time Security (NTS) オプションが追加されました

クライアントサーバーで **NTS** を有効にするために、Timesync RHEL システムロールに **NTS** オプションが追加されました。NTS は、Network Time Protocol (NTP) に指定される新たなセキュリティーメカニズムで、クライアント固有の設定がなくても NTP クライアントの同期をセキュアにでき、大量のクライアントにスケーリングできます。**NTS** オプションは、バージョン 4.0 以降の **chrony** NTP プロバイダーでのみ対応しています。

(BZ#1970664)

SSHD RHEL システムロールが非排他的設定スニペットをサポートするようになりました

この機能を使用すると、名前空間を使用して以前の設定を書き換えることなく、さまざまなロールや Playbook で SSHD を設定できます。名前空間はドロップインディレクトリーと似ており、SSHD 用に非排他設定スニペットを定義します。そのため、設定ファイル全体ではなく、設定のごく一部のみを設定する必要がある場合は、別のロールの SSHD RHEL システムロールを使用できます。

(BZ#1970642)

SELinux ロールが SELinux モジュールを管理可能

SELinux の RHEL システムロールは、SELinux モジュールを管理することができます。今回の更新で、ユーザーは **.pp** または **.cil** ファイルから独自のカスタムモジュールを提供できるようになりました。これにより、より柔軟な SELinux ポリシー管理が可能になりました。

(BZ#1848683)

chrony インターリーブモード、NTP フィルタリング、およびハードウェアのタイムスタンプを管理可能

今回の更新で、**timesync** RHEL システムロールを使用すると、Network Time Protocol(NTP) インターリープモード、NTP 測定の追加フィルタリング、およびハードウェアのタイムスタンプを設定できるようになりました。バージョン 4.0 の **chrony** パッケージでは、ローカルネットワークのクロックの非常に精度の高い安定した同期を実現するために、これらの機能のサポートが追加されました。

- NTP のインターリープモードを有効にするには、サーバーがこの機能をサポートしているのを確認し、**timesync_ntp_servers** リストでサーバーの **xleave** オプションを **yes** に設定します。デフォルト値は **no** です。
- クロック更新ごとの NTP 測定数を設定するには、設定する NTP サーバーの **filter** オプションを設定します。デフォルト値は **1** です。
- NTP 用にハードウェアのタイムスタンプを有効にする必要があるインターフェイスのリストを設定するには、**timesync_ntp_hwts_interfaces** パラメーターを使用します。特殊な値 **["*"]** は、これをサポートするすべてのインターフェイスでタイムスタンプを有効にします。デフォルトは **[]** です。

(BZ#1938020)

timesync ロールにより、chrony のカスタム設定が可能

以前のリリースでは、**timesync** ロールを使用してカスタマイズした **chrony** 設定を指定する方法はありませんでした。今回の更新で、**timesync_chrony_custom_settings** パラメーターが追加されました。これにより、ユーザーは以下のような **chrony** のカスタマイズ設定を指定できます。

```
timesync_chrony_custom_settings:
  - "logdir /var/log/chrony"
  - "log measurements statistics tracking"
```

(BZ#1938023)

timesync ロールが、ハイブリッドのエンドツーエンドの遅延メカニズムに対応

今回の機能拡張により、**timesync_ptp_domains** で新しい **hybrid_e2e** オプションを使用して、**timesync** ロールでハイブリッドのエンドツーエンド遅延メカニズムを有効にすることができます。ハイブリッドのエンドツーエンドの遅延メカニズムはユニキャスト遅延リクエストを使用します。これは、大規模なネットワークでマルチキャストトラフィックを減らすのに便利です。

(BZ#1957849)

ethtool がパケットロスレートおよびレイテンシーの削減に対応

Tx または Rx バッファーは、トラフィックのバーストを処理するためにネットワークアダプターが割り当てるメモリー領域です。パケットロスレートを減らし、許容可能なネットワークレイテンシーを達成するには、これらのバッファーのサイズを適切に管理することが重要になります。

ethtool ユーティリティーは、指定したネットワークデバイスの **ring** オプションを設定して、パケットロスレートまたはレイテンシーを減らすようになりました。

サポート対象の **ring** パラメーターのリストは以下のとおりです。

- **rx** - Rx リングのリングエントリーの数を変更します。
- **rx-jumbo** - Rx ジャンボリングのリングエントリーの数を変更します。
- **rx-mini** - Rx ミニリングのリングエントリー数を変更します。

- **tx**: Tx リングのリングエントリーの数を変更します。

(BZ#1959649)

新しい `ipv6_disabled` パラメーターが利用可能

今回の更新により、`ipv6_disabled` パラメーターを使用して、アドレスの設定時に IPv6 を無効にできるようになりました。

(BZ#1939711)

RHEL システムロールが VPN 管理をサポートするようになりました

以前のリリースでは、Linux で安全で適切な IPsec トンネリングおよび仮想プライベートネットワーク (VPN) ソリューションを設定するのが困難でした。この機能拡張により、VPN RHEL システムロールを使用して、多数のホスト間でホスト間およびメッシュ接続用の VPN トンネルをより簡単にセットアップおよび設定できるようになりました。これにより、RHEL システムロールプロジェクト内で、VPN および IPsec トンネリング設定用の一貫した設定インターフェイスが得られます。

(BZ#1943679)

storage RHEL システムロールが `filesystem` の再ラベル付けに対応

以前のバージョンでは、`storage` ロールは再ラベル付けをサポートしていました。今回の更新で問題が修正され、`filesystem` ラベルの再ラベル付けがサポートされるようになりました。これには、`storage_volumes` で新しいラベル文字列を `fs_label` パラメーターに設定します。

(BZ#1876315)

Storage システムロールで、パーセンテージで表現されたボリュームサイズのサポートが利用可能になりました

この機能拡張により、Storage RHEL システムロールにサポートが追加され、LVM ボリュームサイズをプールの合計サイズのパーセンテージとして表現できるようになります。LVM ボリュームのサイズは、人間が判読できるファイルシステムのサイズ (10g、50 GiB など) に加えて、プール/VG サイズの割合 (50% など) で指定できます。

(BZ#1894642)

Microsoft SQL Server Management 用の新しい Ansible ロール

新しい `microsoft.sql.server` ロールは、IT およびデータベース管理者が、Red Hat Enterprise Linux で SQL Server の設定、設定、およびパフォーマンスチューニングに関連するプロセスを自動化するのに役立ちます。

(BZ#2013853)

RHEL システムロールが Ansible 2.8 をサポートしない

今回の更新では、Ansible 2.8 が製品ライフサイクルの終了を迎えたため、サポートされません。RHEL のシステムロールは Ansible 2.9 に対応しています。

(BZ#1989199)

RHEL システムロールの `postfix` ロールに完全対応

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。

RHEL 8.5 の時点では、**postfix** ロールは完全にサポートされます。

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](#))

4.17. 仮想化

Web コンソールでの仮想マシンの管理の強化

RHEL 8 Web コンソールの仮想マシン (VM) セクションのデザインが変更され、より使いやすくなりました。また、以下のような変更や機能も導入されています。

- 仮想マシンのステータス、ディスク、ネットワーク、コンソール情報など、関連するすべての仮想マシン情報が1つのページに表示されるようになりました。
- Web コンソールを使用して仮想マシンのライブマイグレーションを実施できるようになりました。
- Web コンソールで、仮想マシンのネットワークインターフェイスの MAC アドレスを編集できるようになりました。
- Web コンソールを使用して、仮想マシンに接続されているホストデバイスのリストを表示することができます

(JIRA:RHELPLAN-79074)

zPCI デバイスの割り当て

zPCI デバイスは、IBM Z ハードウェアで実行している RHEL 8 にホストされている仮想マシンに、仲介デバイスとして接続できるようになりました。たとえば、これを使用すると、仮想マシンで NVMe フラッシュドライブを使用できます。

(JIRA:RHELPLAN-59528)

4.18. サポート性

sos がバージョン 4.1 にリベース

sos パッケージがバージョン 4.1 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。主な機能拡張は、次のとおりです。

- Red Hat Update Infrastructure(**RHUI**) プラグインは、**sos** パッケージにネイティブに実装されるようになりました。**rhui-debug.py** python バイナリーを使用すると、**sos** は、メインの設定ファイル、**rhui-manager** ログファイル、またはインストール設定など、**RHUI** からレポートを収集できます。
- sos** では、コマンド実行のタイムアウトを手動で設定する **--cmd-timeout** グローバルオプションが導入されました。デフォルト値 (-1) は、一般的なコマンドタイムアウト (300 秒) を延長します。

([BZ#1928679](#))

4.19. コンテナー

デフォルトのコンテナーアイメージの署名検証ができるようになりました。

以前では、Red Hat コンテナーレジストリーのポリシー YAML ファイルは、/etc/containers/registries.d/ ディレクトリーに手動で作成する必要がありました。現在、**registry.access.redhat.com.yaml** ファイルおよび **registry.redhat.io.yaml** ファイルは **containers-common** に同梱されています。**podman image trust** を使用して、RHEL でコンテナーアイメージの署名を検証できるようになりました。

(JIRA:RHELPLAN-75166)

container-tools:rhel8 モジュールが更新されました。

Podman、Buildah、Skopeo、および runc ツールを含む **container-tools:rhel8** モジュールが利用できるようになりました。今回の更新で、以前のバージョンに対するバグ修正および機能拡張のリストが追加されました。

(JIRA:RHELPLAN-76515)

containers-common が利用可能に

containers-common パッケージが **container-tools:rhel8** モジュールに追加されました。**containers-common** パッケージには、Podman、Buildah、Skopeo などのコンテナーツールのエコシステムに関する一般的な設定ファイルおよびドキュメントが含まれています。

(JIRA:RHELPLAN-77542)

カーネルにおけるオーバーレイファイルシステムのネイティブサポートが利用可能

オーバーレイファイルシステムのサポートがカーネル 5.11 から利用可能になりました。非ルートユーザーは、ルートレスで (ユーザーとして) 実行しても、ネイティブなオーバーレイ性能を発揮します。従って、この機能強化により、バインドマウントを必要とせずにオーバーレイファイルシステムを使用したい非 root ユーザーに、より良いパフォーマンスを提供します。

(JIRA:RHELPLAN-77241)

podman コンテナーアイメージが利用可能に

これまでテクノロジープレビューとして提供されていた **registry.redhat.io/rhel8/podman** コンテナーアイメージが、完全にサポートされるようになりました。**registry.redhat.io/rhel8/podman** コンテナーアイメージは、**podman** パッケージをコンテナ化した実装です。**podman** ツールは、コンテナーおよびイメージ、それらのコンテナーにマウントされたボリューム、およびコンテナーのグループから作成された Pod を管理します。

(JIRA:RHELPLAN-57941)

ユニバーサルベースイメージが Docker Hub で利用可能に

これまでユニバーサルベースイメージは、Red Hat コンテナーカタログからしか入手できませんでした。現在、ユニバーサルベースイメージは Docker Hub からも入手可能です。

詳細は、[Red Hat Brings Red Hat Universal Base Image to Docker Hub](#) を参照してください。

(JIRA:RHELPLAN-85064)

Podman で CNI プラグインが利用できるようになりました。

Podman のルートレスモードで CNI プラグインが使用できるようになりました。ルートレスネットワークコマンドは、システムでその他の要件がなくても機能するようになりました。

(BZ#1934480)

Podman はバージョン 3.3.1 に更新されました。

Podman ユーティリティーがバージョン 3.3.1 に更新されました。主な機能拡張は、次のとおりです。

- Podman は、**--restart** オプションで作成されたコンテナーを、システム再起動後に再起動できるようになりました。
- **podman container checkpoint** および **podman container restore** コマンドは、Pod 内にあるコンテナーのチェックポイントとリストアをサポートし、それらのコンテナーを Pod にリストアするようになりました。さらに、**podman container restore** コマンドは、エクスポートされたチェックポイントからリストアされたコンテナーに転送されるポートを変更する **--publish** オプションをサポートするようになりました。

(JIRA:RHELPLAN-87877)

crun OCI ランタイムが利用可能

crun OCI ランタイムが、**container-tools:rhel8** モジュールで利用できるようになりました。crun コンテナーランタイムは、コンテナーがルートレスユーザーの追加グループにアクセスできるようにするアノテーションに対応しています。これは、`setgid` が設定されたディレクトリーでボリュームをマウントする場合、またはユーザーがグループアクセスのみを持つ場合にコンテナー操作を行う際に役立ちます。

(JIRA:RHELPLAN-75164)

podman UBI イメージが利用可能に

UBI の一環として、`registry.access.redhat.com/ubi8/podman` が利用可能になりました。

(JIRA:RHELPLAN-77489)

container-tools:rhel8 モジュールが更新されました。

Podman、Buildah、Skopeo、および runc ツールを含む **container-tools:rhel8** モジュールが利用できるようになりました。今回の更新で、以前のバージョンに対するバグ修正および機能拡張のリストが追加されました。

詳細は、[RHEA-2021: 85361-01](#) を参照してください。

(BZ#2009153)

ubi8/nodejs-16 および ubi8/nodejs-16-minimal コンテナーアイメージが完全にサポートされるようになりました。

テクノロジープレビューとして以前利用できた **ubi8/nodejs-16** コンテナーアイメージおよび **ubi8/nodejs-16-minimal** コンテナーアイメージは、[RHBA-2021: 5260](#) アドバイザリーのリリースで完全に対応しています。このコンテナーアイメージには、LTS (Long Term Support) バージョンである **Node.js 16.13** が含まれます。

(BZ#2001020)

第5章 外部カーネルパラメーターへの重要な変更

本章では、システム管理者向けに、Red Hat Enterprise Linux 8.5 に同梱されるカーネルの重要な変更点の概要を説明します。変更には、たとえば、**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。

新しいカーネルパラメーター

idxd.sva = [HW]

形式: <bool>

このパラメーターでは、**idxd** カーネルモジュールに対する SVA (Shared Virtual Memory) サポートを強制的に無効にできます。

デフォルト値は **true** (1) です。

lsm.debug = [SECURITY]

このパラメーターを使用すると、Linux Security Module (LSM) 初期化デバッグ出力を有効にできます。

lsm = **lsm1**...,**lsmN** [SECURITY]

このパラメーターでは、Linux セキュリティモジュール (LSM) の初期化の順序を選択できます。このパラメーターは、**CONFIG_LSM** オプションおよび **security=** パラメーターを上書きします。

rcutree.qovId = [KNL]

このパラメーターでは、キューに入れられた RCU (Read-copy-update) コールバックのしきい値を設定できます。このしきい値を超えると、RCU の force-quiescent-state スキャンにより、**cond_resched()** システムコールのヘルプが積極的に追加され、IPI の設定により、CPU がより速く静止状態に到達できるようになります。

このパラメーターは、システムの起動時に **rcutree.qhimark** パラメーターに基づいて設定するため、このパラメーターを 0 より小さい値に設定できます。また、このパラメーターをゼロに設定すると、より積極的なヘルプの参加が無効になります。

rcutree.rcu_unlock_delay = [KNL]

このパラメーターでは、設定の布尔値が **CONFIG_RCU_STRICT_GRACE_PERIOD=y** に設定されているカーネル単位で、**rcu_read_unlock()**-time 遅延を指定できます。

デフォルト値は 0 です。

遅延が大きいほど、RCU (Read-copy-update) ポインターリークをキャッチする可能性が高くなります。これは、関連する **rcu_read_unlock()** が完了した後の RCU 保護ポインターの欠陥による使用です。

rcutortureirqreader = [KNL]

このパラメーターを使用すると、割り込み要求 (IRQ) ハンドラーまたはタイマーハンドラーから RCU (Read-copy-update) リーダーを実行できます。

rcutorture.leakpointer = [KNL]

このパラメーターを使用すると、リーダーから RCU (Read-copy-update) 保護ポインターをリークできます。これによりスプラットが発生することがあります。このようなリークを検出するためには、**CONFIG_RCU_STRICT_GRACE_PERIOD=y** などの設定の機能をテストすることを目的としています。

rcutorture.read_exit = [KNL]

このパラメーターでは、read-then-exit kthread の数を設定して、Read-copy-update (RCU) 更新プログラムと task-exit 処理の相互作用をテストできます。

rcutorture.read_exit_burst = [KNL]

このパラメーターでは、指定した read-then-exit エピソードで、一連の read-then-exit kthread が起動する回数を指定できます。

rcutorture.read_exit_delay = [KNL]

このパラメーターでは、連続した read-then-exit テストエピソードの間の遅延を秒単位で指定できます。

rcutorture.stall_cpu_block = [KNL]

このパラメーターでは、停止中にスリープを設定できます。その結果、他のストール関連のアクティビティーのほかに、プリエンプティブな RCU (Read-copy-update) からの警告が発生する可能性があります。

rcutorture.stall_gp_kthread = [KNL]

このパラメーターでは、RCU (Read-copy-update) 猶予期間 kthread 内で強制スリープし、RCU CPU ストール警告をテストする期間 (秒) を指定できます。

このパラメーターをゼロに設定すると、機能が無効になります。

stall_cpu パラメーターと **stall_gp_kthread** パラメーターの両方が指定されている場合は、最初に kthread が不足し、次に CPU が不足します。

rcupdate.rcu_cpu_stall_suppress_at_boot = [KNL]

このパラメーターを使用すると、システムの初期起動時に発生する RCU CPU stall 警告メッセージおよび rcutre writer stall 警告を抑制できます。これは、**init** タスクが起動する前の時間です。

rcupdate.rcu_task_ipi_delay = [KNL]

このパラメーターでは、指定の猶予期間の開始から始まり、RCU (Read-copy-update) タスクが IPI の送信を回避する時間を jiffies 単位で設定できます。

大きな数を設定すると、リアルタイムワークロードへの阻害を回避できますが、猶予期間は長くなります。

refscale.holdoff = [KNL]

このパラメーターでは、test-start holdoff 期間を設定できます。このパラメーターの目的は、干渉を防ぐために、システムの起動が完了するまでテストの開始を遅らせることです。

refscale.loops = [KNL]

このパラメーターでは、テスト中の同期プリミティブでループの数を設定できます。この数を増やすと、ループの開始や終了のオーバーヘッドによるノイズが低減します。

デフォルト値では、パスごとのノイズがすでに、約 2020 x 86 ラップトップでほんの数ピコ秒に低減されています。

refscale.nreaders = [KNL]

このパラメーターでは、リーダーの数を設定できます。

デフォルト値の -1 は N を選択します。N は、CPU 数の約 75% です。

refscale.nruns = [KNL]

このパラメーターでは、実行数を設定できます。各実行数はコンソールログにダンプされます。

refscale.readdelay = [KNL]

このパラメーターでは、読み取り側の critical-section 持続時間 (マイクロ秒) を設定できます。

refscale.scale_type = [KNL]

このパラメーターを使用すると、テストする読み取り保護の実装を指定できます。

refscale.shutdown = [KNL]

このパラメーターを使用すると、パフォーマンステストの終了時にシステムをシャットダウンできます。

デフォルト値は 1 で、システムがシャットダウンします。refscale はカーネルに組み込まれています。

この値は 0 で、システムは実行したままになります。refscale はモジュールとして構築されます。

refscale.verbose = [KNL]

このパラメーターを使用すると、追加の `printf()` ステートメントを有効にできます。

scftorture.holdoff = [KNL]

このパラメーターでは、テストを開始する前に保留する秒数を指定できます。

このパラメーターは、モジュール挿入の場合はデフォルトで 0 になり、組み込みの `smp_call_function()` テストの場合は 10 秒になります。

scftorture.longwait = [KNL]

このパラメーターを使用すると、選択した制限までランダムに選択される非常に長い待機を要求できます(秒単位)。

デフォルト値はゼロで、この機能は無効になります。

ゼロ以外の秒数を少しでも要求すると、RCU (Read-copy-update) の CPU stall 警告、ソフトウェアロックに関するエラーなどが発生する可能性があることに注意してください。

scftorture.threads = [KNL]

このパラメーターでは、`smp_call_function()` 関数ファミリーを呼び出すために起動する kthread の数を指定できます。

デフォルトの -1 は、CPU 数と同じ kthread 数を指定します。

scftorture.onoff_holdoff = [KNL]

このパラメーターでは、CPU ホットプラグ操作を開始する前に、テストの開始後待機する秒数を指定できます。

scftorture.onoff_interval = [KNL]

このパラメーターでは、CPU ホットプラグによる連続した操作の間待機する秒数を指定できます。

デフォルト値はゼロで、CPU ホットプラグの操作が無効になります。

scftorture.shutdown_secs = [KNL]

このパラメーターでは、テスト開始後の秒数を指定できます。テストが終了すると、システムがシャットダウンします。

デフォルト値をゼロにすると、システムのシャットダウンを回避できます。ゼロ以外の値は自動テストに役立ちます。

scftorture.stat_interval = [KNL]

このパラメーターでは、現在のテスト統計をコンソールに出力する間隔を秒数で指定できます。値をゼロにすると、統計出力が無効になります。

scftorture.stutter_cpus = [KNL]

このパラメーターでは、テスト対象の CPU セットへの変更ごとに待機する jiffies 数を指定できます。

scftorture.use_cpus_read_lock = [KNL]

このパラメーターでは、デフォルトの `preempt_disable()` システムコールの代わりに `use_cpus_read_lock()` システムコールを使用して、`smp_call_function*()` 関数のいずれかを呼び出す際に CPU ホットプラグを無効にできます。

`scftorture.verbose = [KNL]`

このパラメーターを使用すると、追加の `printf()` ステートメントを有効にできます。

`scftorture.weight_single = [KNL]`

このパラメーターでは、ゼロの wait パラメーターを持つ `smp_call_function_single()` 関数に使用する確率の重み付けを指定します。

値を -1 にすると、その他のすべての重みが -1 の場合にデフォルトが選択されます。ただし、少なくとも 1 つの重みにその他の値がある場合は、-1 に設定すると、重みがゼロになります。

`scftorture.weight_single_wait = [KNL]`

このパラメーターでは、ゼロ以外の wait パラメーターを持つ `smp_call_function_single()` 機能に使用する確率の重み付けを指定します。詳細は `weight_single` を参照してください。

`scftorture.weight_many = [KNL]`

このパラメーターでは、ゼロの wait パラメーターを持つ `smp_call_function_many()` 関数に使用する確率の重み付けを指定します。

この重みの確率を高く設定すると、システムに深刻な IPI (Inter-processor Interrupt) 負荷がかかる可能性があることに注意してください。

詳細は `weight_single` を参照してください。

`scftorture.weight_many_wait = [KNL]`

このパラメーターでは、ゼロ以外の wait パラメーターを持つ `smp_call_function_many()` 関数に使用する確率の重み付けを指定します。

詳細は、`weight_single` および `weight_many` を参照してください。

`scftorture.weight_all = [KNL]`

このパラメーターでは、ゼロの wait パラメーターを持つ `smp_call_function_all()` 関数に使用する可能性の加重を指定します。

詳細は、`weight_single` および `weight_many` を参照してください。

`scftorture.weight_all_wait = [KNL]`

このパラメーターでは、ゼロ以外の wait パラメーターを持つ `smp_call_function_all()` 関数に使用する確率加重を指定します。

詳細は、`weight_single` および `weight_many` を参照してください。

`sched_energy_aware`

このパラメーターは、Energy Aware Scheduling (EAS) を有効または無効にします。

EAS は、Energy Model が利用可能な非対称 CPU トポロジーを持つプラットフォームで自動的に起動します。

お使いのプラットフォームが EAS の要件を満たしているが、これを使用しない場合は、この値を 0 に変更します。

`torture.disable_onoff_at_boot = [KNL]`

このパラメーターを使用すると、`init` タスクが起動するまで、CPU ホットプラグコンポーネントの機能を防ぐことができます。

torture.ftrace_dump_at_shutdown = [KNL]

このパラメーターを使用すると、エラーが発生していないくとも、torture-test のシャットダウン時に **ftrace** バッファーをダンプできます。

多くの torture テストが同時に実行されている場合 (特に、rotating-rust ストレージを使用するシステムで) は、これにより操作が非常にコストがかかる場合があります。

更新されたカーネルパラメーター**iommu.forcedac = [ARM64, X86]**

このパラメーターを使用すると、PCI デバイスへの IOVA (Input-output Virtual Address) の割り当てを制御できます。

形式: {0|1}

- **0** - まず 32 ビットの DMA (Direct Memory Access) アドレスを割り当ててから、必要に応じてフルレンジに戻すようにします。
- **1** - 使用可能な全範囲から直接割り当てます。このオプションは、32 ビットを超えるアドレス指定に対応する PCI カードに対して Dual Address Cycle を強制します。

page_poison = [KNL]

この boot-time パラメーターを使用すると、**CONFIG_PAGE_POISONING=y** 設定で利用可能なバディアロケーターのポイズニング状態を変更できます。

- **オフ** - ポイズニングを無効にします (デフォルト)。
- **on** - ポイズニングを有効にします。

rcuscale.gp_async = [KNL]

このパラメーターを使用すると、**call_rcu()** などの非同期の猶予期間プリミティブのパフォーマンスを測定できます。

rcuscale.gp_async_max = [KNL]

このパラメーターでは、ライター・スレッドごとの未解決のコールバックの最大数を指定できます。ライタースレッドがこの制限を超えると、対応する **rcu_barrier()** フレーバーを呼び出して、事前にポストされたコールバックをドレイン (解放) できるようにします。

rcuscale.gp_exp = [KNL]

このパラメーターを使用すると、非同期の猶予期間プリミティブのパフォーマンスを測定できます。

rcuscale.holdoff = [KNL]

このパラメーターでは、test-start holdoff 期間を設定できます。このパラメーターの目的は、干渉を防ぐために、システムの起動が完了するまでテストの開始を遅らせることです。

rcuscale.kfree_rcu_test = [KNL]

このパラメーターを使用すると、**kfree_rcu()** フラッディングのパフォーマンスを測定できます。

rcuscale.kfree_nthreads = [KNL]

このパラメーターでは、**kfree_rcu()** 機能のループを実行しているスレッドの数を指定できます。

rcuscale.kfree_alloc_num = [KNL]

このパラメーターでは、1回の反復で実行する割り当てと解放の数を指定できます。

rcuscale.kfree_loops = [KNL]

このパラメーターでは、**rcuscale.kfree_alloc_num** 数の割り当てと解放を行うループの数を指定できます。

rcuscale.nreaders = [KNL]

このパラメーターでは、RCU (Read-copy-update) リーダーの数を設定できます。
値 -1 は N を選択します。N は CPU の数です。

rcuscale.nwriters = [KNL]

このパラメーターでは、RCU (Read-copy-update) ライターの数を設定できます。
値の動作は **rcuscale.nreaders=N** と同じです。N は CPU の数になります。

rcuscale.perf_type = [KNL]

このパラメーターでは、テストする RCU (Read-copy-update) 実装を指定できます。

rcuscale.shutdown = [KNL]

パフォーマンステストが完了したら、このパラメーターを使用してシステムをシャットダウンできます。これは、自動テストをハンズオフする場合に役立ちます。

rcuscale.verbose = [KNL]

このパラメーターを使用すると、追加の **printf()** ステートメントを有効にできます。

rcuscale.writer_holdoff = [KNL]

このパラメーターでは、猶予期間の間に、マイクロ秒単位で holdoff を書き込むことができます。
デフォルト値はゼロで、no holdoff を意味します。

security = [SECURITY]

このパラメーターでは、システムの起動時に有効にする従来のメジャーセキュリティーモジュールを選択できます。

これは、**lsm** パラメーターで非推奨になりました。

split_lock_detect = [X86]

このパラメーターでは、スプリットロック検出、またはバスロック検出を有効にできます。
これを有効にし、ハードウェアのサポートがある場合は、キャッシュラインの境界を越えてデータにアクセスするアトミック命令の結果が以下のようになります。

- スプリットロック検出用のアライメントチェック例外
- バスロック検出用のデバッグ例外
値:
 - **off** - 機能は有効になりません。
 - **warn** - カーネルはアプリケーションに関するレート制限付きの警告を出力し、#AC 例外または#DB 例外をトリガーします。このモードは、スプリットロック検出またはバスロック検出に対応する CPU のデフォルトです。ハードウェアで両方の機能が有効になっている場合、デフォルトの動作は#ACになります。
 - **fatal** - カーネルは、#AC 例外または#DB 例外をトリガーするアプリケーションに **SIGBUS** シグナルを送信します。ハードウェアで両方の機能が有効になっている場合、デフォルトの動作は#ACになります。
 - **ratelimit:N** - バスロック検出では、システム全体のレート制限を1秒あたり N 個のバスロックに設定します (0 < N ≤ 1000)。スプリットロック検出の場合は N/A。

カーネルまたはファームウェアで #AC 例外がヒットすると (ユーザー モードでの実行中には実行されない)、カーネルが **warn** モードまたは **fatal** モードで oops を発生させます。

バスロックの #DB 例外は、CPL > 0 の場合にのみトリガーされます。

usb-storage.quirks =

k = NO_SAME (**WRITE_SAME** を使用しない、UAS のみ)

第6章 デバイスドライバー

6.1. 新しいドライバー

ネットワークドライバー

- SYNOPSYS DESIGNWARE Ethernet XPCS ドライバー (pcs-xpcs.ko.xz)
- INTEL 10/100/1000 Ethernet PCI ドライバー (dwmac-intel.ko.xz)
- STMMAC 10/100/1000 Ethernet デバイスドライバー (stmmac.ko.xz)
- Crypto IPSEC for Chelsio Terminator カード (ch_ipsec.ko.xz): 1.0.0.
- Chelsio NIC TLS ULD ドライバー (ch_ktls.ko.xz): 1.0.0.
- Microsoft Azure Network Adapter ドライバー (mana.ko.xz)
- Qualcomm Atheros 802.11ax ワイヤレス LAN カード用のコアモジュール (ath11k.ko.xz)
- Qualcomm Technologies 802.11ax WLAN PCIe デバイスのドライバーサポート (ath11k_pci.ko.xz)
- MAC からオプションの PHY への接続 (phylink.ko.xz)

グラフィックドライバーとその他のドライバー

- In-Band ECC を使用する Intel クライアント SoC 用 MC ドライバー (igen6_edac.ko.xz)
- Regmap SoundWire MBQ Module (regmap-sdw-mbq.ko.xz)
- Intel Platform Monitoring Technology PMT ドライバー (intel_pmt.ko.xz)
- Intel PMT Crashlog ドライバー (intel_pmt_crashlog.ko.xz)
- UV システムの Sysfs 構造 (uv_sysfs.ko.xz)
- Intel PMT Telemetry ドライバー (intel_pmt_telemetry.ko.xz)
- Intel PMT Class ドライバー (intel_pmt_class.ko.xz)
- AMD PMC Driver (amd-pmc.ko.xz)
- MHI Host Interface (mhi.ko.xz)
- Modem Host Interface (MHI) PCI コントローラードライバー (mhi_pci_generic.ko.xz)
- vDPA Device Simulator for block device (vdpa_sim_blk.ko.xz): 0.1
- vDPA Device Simulator for networking device (vdpa_sim_net.ko.xz): 0.1
- vp-vdpa (vp_vdpa.ko.xz): 1
- Mellanox VDPA ドライバー (mlx5_vdpa.ko.xz)
- Basic STM フレーミングプロトコルドライバー (stm_p_basic.ko.xz)
- MIPI SyS-T STM フレーミングプロトコルドライバー (stm_p_sys-t.ko.xz)

- QMI encoder/decoder ヘルパー (qmi_helpers.ko.xz)
- ACPI DPTF プラットフォーム電源 ドライバー (dptf_power.ko.xz)
- ACPI Platform プロファイル sysfs インターフェイス (platform_profile.ko.xz)
- Intel Emmitsburg PCH pinctrl/GPIO ドライバー (pinctrl-emmitsburg.ko.xz)
- Intel Alder Lake PCH pinctrl/GPIO ドライバー (pinctrl-alderlake.ko.xz)
- MPI3 Storage Controller Device Driver (mpi3mr.ko.xz): 00.255.45.01
- 実行される CPU IO に基づいてパスを選択する device-mapper マルチパスセレクター (dm-io-affinity.ko.xz)
- デバイスマッパー測定サービス時間指向パスセレクター (dm-historical-service-time.ko.xz)

6.2. 更新された ドライバー

ネットワーク ドライバー

- Mellanox 第 5 世代ネットワークアダプター (ConnectX シリーズ) コアドライバー (mlx5_core.ko.xz) がバージョン 4.18.0-348.el8.x86_64 に更新されました。
- realtek RTL8152/RTL8153 Based USB Ethernet Adapters (r8152.ko.xz) がバージョン v1.11.11 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

- LSI MPT Fusion SAS 3.0 Device Driver (mpt3sas.ko.xz) がバージョン 37.101.00.00 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 0:12.8.0.10 に更新されました。
- QLogic Fibre Channel HBA ドライバー (qla2xxx.ko.xz) がバージョン 10.02.00.106-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 2.1.8-045 に更新されました。
- Broadcom MegaRAID SAS ドライバー (megaraid_sas.ko.xz) がバージョン 07.717.02.00-rh1 に更新されました。

第7章 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 8.5 のバグで修正されたものを説明します。

7.1. インストーラーおよびイメージの作成

Insights クライアントがシステムの登録に失敗した場合に RHEL インストールを中断しなくなりました。

以前は、Red Hat Insights クライアントがインストール時にシステムの登録に失敗すると、最終的に RHEL インストールがエラーと共に失敗していました。今回の更新で、Insights クライアントが失敗しても、システムがインストールを完了するようになりました。インストール時のエラーに関して通知されるため、エラーは後で個別に処理できます。

([BZ#1931069](#))

Anaconda では、カスタムパーティショニング画面で自動的に作成されたディスクレイアウトのデータ暗号化が可能です。

これまで、カスタムパーティショニング画面でディスクレイアウトを自動作成した際に、暗号化されたディスクレイアウトを要求することはできませんでした。今回の更新では、Anaconda はカスタムパーティショニング画面で、自動的に作成されたディスクレイアウトを暗号化するオプションを提供します。

([BZ#1903786](#))

キックスタートファイルにパーティション設定スキームが指定されていない場合は、インストールプログラムは自動パーティション設定を試みない

キックスタートファイルを使用して自動インストールを実行する場合、キックスタートファイルでパーティション設定スキームを指定しないと、インストールプログラムが自動パーティション設定を実行しません。インストールプロセスが中断され、ユーザーがパーティションの設定を行うことができます。

([BZ#1954408](#))

RHEL-Edge コンテナーアイメージが nginx を使用し、ポート 8080 で動作するようになりました。

以前のバージョンでは、**edge-container** イメージタイプは root 以外のモードで実行できませんでした。そのため、Red Hat OpenShift 4 は **edge-container** イメージタイプを使用できませんでした。今回の機能強化により、コンテナーは **nginx** HTTP サーバーを使用してコミットと、コンテナー内の root 以外のユーザーとしてサーバーが実行できるようにする設定ファイルを提供し、Red Hat OpenShift 4 で使用できるようになりました。内部 Web サーバーは、**80** ではなくポート **8080** を使用するようになりました。

([BZ#1945238](#))

7.2. シェルおよびコマンドラインツール

opal-prd がバージョン 6.7.1 にリベース

opal-prd がバージョン 6.7.1 にアップグレードされました。主なバグ修正と機能拡張は、以下のとおりです。

- **xscom OPAL**呼び出しによる **xscom** エラーログの問題を修正しました。

- **DEBUG** ビルドでデッドロックが修正されました。
- **fast-reboot** が **core/platform** で失敗した場合は、**full_reboot** にフォールバックします。
- **core/cpu** の **next_ungarded_primary** が修正されました。
- Self-Boot Engine (SBE) におけるレートリミットタイマーの要求とタイマーの状態が改善されました。

(BZ#1921665)

libservicelog がバージョン 1.1.19 にリベース

libservicelog がバージョン 1.1.19 にアップグレードされました。主なバグ修正と機能拡張は、以下のとおりです。

- 出力アライメントの問題を修正
- **servicelog_open()** の失敗時の **segfault** が修正されました。

(BZ#1844430)

ipmitool sol activate コマンドがクラッシュしなくなる

以前は、RHEL 7 から RHEL 8 にアップグレードした後に、IBM DataPower アプライアンスのリモートコンソールにアクセスしようとすると、**ipmitool sol activate** コマンドがクラッシュしていました。

今回の更新によりバグが修正され、**ipmitool** を使用して再度リモートコンソールにアクセスできるようになりました。

(BZ#1951480)

Relax-and-Recover(ReaR) パッケージが bootlist 実行可能ファイルに依存するようになりました。

以前のリリースでは、ReaR は、IBM Power Systems(リトルエンディアン)アーキテクチャーに bootlist 実行ファイルなしにレスキューイメージを生成することができました。したがって、**powerpc-utils-core** パッケージがインストールされていない場合、レスキューイメージには bootlist 実行可能ファイルが含まれませんでした。

今回の更新で、ReaR パッケージは bootlist 実行ファイルに依存するようになりました。この依存関係により、bootlist 実行ファイルが存在するようになります。bootlist 実行ファイルがない場合に、ReaR はレスキューイメージを作成しません。これにより、無効なレスキューイメージが作成されるのを防ぐことができます。

(BZ#1983013)

特権のないリモートユーザーで rsync を ReaR で使用できるようになりました。

以前のバージョンでは、rsync を使用してシステムデータのバックアップおよび復元を行うと (**BACKUP=RSYNC**)、rsync へのパラメーターは誤って引用され **--fake-super** パラメーターがリモートの rsync プロセスに渡されませんでした。そのため、ファイルメタデータが正しく保存および復元されませんでした。

今回の更新で、以下のバグが修正されました。

- ReaR は rsync の正しいパラメーターを使用します。

- バックアップおよび復元中のエラー検出の rsync コードが改善されました。
 - バックアップ中に rsync エラーが検出されると、ReaR はエラーメッセージを出して中止します。
 - 復元中に rsync エラーが検出される場合、ReaR は警告メッセージを表示します。

`/etc/rear/local.conf` ファイルで **BACKUP_INTEGRITY_CHECK=1** を設定して、警告をエラーメッセージにします。

([BZ#1930662](#))

ReaR の使用時にネットワーク共有のバックアップデータが失われなくなりました

以前のリリースでは、NFSなどのネットワークファイルシステムを使用して ReaR バックアップを保存した場合、エラーが発生すると、ReaR は NFS がマウントされたディレクトリーを削除していました。そのため、バックアップデータの喪失が発生していました。

今回の更新により、ReaR が新しい方法を使用してネットワーク共有をアンマウントするようになりました。この新しいメソッドでは、マウントポイントが削除されても、マウントされたファイルシステムの内容は削除されません。ReaR の使用時にネットワーク共有のバックアップデータが失われる問題が修正されました。

([BZ#1958247](#))

ReaR は、ESP を使用するマシンのバックアップおよび復旧に使用できるようになりました。

以前は、Unified Extensible Firmware Interface(UEFI) ファームウェアを使用しているマシンの EFI システムパーティションにソフトウェア RAID(MDRAID) が使用される場合、ReaR は Extensible Firmware Interface(EFI) エントリーを作成しませんでした。ソフトウェア RAID で UEFI ファームウェアおよび EFI システムパーティションを使用するシステムが ReaR を使用して復元されると、回復したシステムは起動できず、ブート EFI 変数の修正に手動の介入が必要でした。

今回の更新で、ソフトウェア RAID デバイスのブート EFI エントリー作成のサポートが ReaR に追加されました。ReaR を使用して、復元後の手動介入なしに、ソフトウェア RAID で EFI システムパーティション (ESP) を使用するマシンをバックアップおよび復元できるようになりました。

([BZ#1958222](#))

openslp パッケージに `/etc/slp.spi` ファイルを追加

以前は、`/etc/slp.spi` ファイルが **openslp** パッケージで欠落していました。そのため、`/usr/bin/slptool` コマンドが出力を生成しませんでした。今回の更新で、`/etc/slp.spi` が **openslp** に追加されました。

([BZ#1965649](#))

マルチパスを持つ IBM Power Systems(リトルエンディアン) アーキテクチャマシンを ReaR を使用して安全に復旧できるようになりました。

以前は、ReaR がシステムを復旧する際に、`/sys` ファイルシステムが chroot にマウントされませんでした。IBM Power Systems(Little Endian) アーキテクチャーの **of pathname** 実行ファイルは、ブートローダーのインストール時に失敗していました。その結果、エラーは検出されず、復元したシステムは起動できませんでした。

今回の更新で、ReaR はリカバリー chroot に `/sys` ファイルシステムをマウントするようになりました。ReaR は、Power Systems(リトルエンディアン) アーキテクチャーのマシン上のレスキューステムに **of pathname** が存在するようにします。

(BZ#1983003)

エイリアスを使用した場合に、**which** ユーティリティーが構文エラーメッセージで中止されなくなりました

以前は、**A=B which ls** などのエイリアスで **which** コマンドを使用しようとすると、**which** ユーティリティーは構文エラーメッセージ **bash: syntax error near unexpected token `'** で中断します。

このバグは修正され、**which** がエラーメッセージなしでコマンドのフルパスを正しく表示するようになりました。

(BZ#1940468)

7.3. インフラストラクチャーサービス

/var/lib/chrony のパーミッションを変更

以前は、エンタープライズセキュリティースキナーは、**/var/lib/chrony** ディレクトリーに誰でも読み取り/実行可能なフラグを付けていました。今回の更新で、**/var/lib/chrony** ディレクトリーのパーミッションが変更され、root および chrony ユーザーのみにアクセスを制限するようになりました。

(BZ#1939295)

7.4. SECURITY

明示的に信頼される場合、**GnuTLS** は SHA-1 で署名された CA を拒否しなくなりました。

以前は、CA が明示的に信頼できる場合でも、**GnuTLS** ライブラリーはすべての認証局 (CA) の署名ハッシュの強度をチェックしていました。そのため、SHA-1 アルゴリズムで署名された CA を含むチェーンは、エラーメッセージ **certificate's signature hash strength is unacceptable** と共に拒否されました。今回の更新で、**GnuTLS** は署名ハッシュの強度チェックから信頼される CA を除外するため、弱いアルゴリズムを使用して署名されている CA を含む証明書チェーンでも拒否しなくなりました。

(BZ#1965445)

FIPS モードでハードウェア最適化が有効

これまでの Federal Information Processing Standard(FIPS 140-2) では、ハードウェアの最適化を使用することは認められていませんでした。そのため、FIPS モードの場合、**libgcrypt** パッケージではこの操作が無効になっていました。今回の更新により、FIPS モードでのハードウェアの最適化が可能になり、その結果、すべての暗号処理が高速に実行されるようになりました。

(BZ#1976137)

leftikeport オプションおよび **rightikeport** オプションが正しく機能する

以前は、Libreswan は、ホスト間の Libreswan 接続では **leftikeport** オプションと **rightikeport** オプションを無視していました。これにより、Libreswan は、デフォルト以外のオプション設定にかかわらず、デフォルトのポートを使用していました。今回の更新で問題が修正され、デフォルトオプションの **leftikeport** および **rightikeport** 接続オプションを使用できるようになりました。

(BZ#1934058)

SELinux ポリシーにより GDM が GRUB `boot_success` フラグを設定できなかった

以前は、SELinux ポリシーにより、パワーオフおよび再起動操作時に、GNOME Display Manager(GDM) は GRUB **boot_success** フラグを設定できませんでした。その結果、GRUB メニューが次回の起動時に表示されました。今回の更新で、SELinux ポリシーで、GDM が GRUB **boot_success** フラグを設定できる新しい **xdm_exec_bootloader** ブール値(デフォルトで有効)が導入されました。これにより、初回ブートで GRUB ブートメニューが表示され、フリッカーのない起動をサポートする機能が正しく機能するようになりました。

(BZ#1994096)

selinux-policy は、TCP カプセル化を使用した IPsec ベースの VPN をサポートします。

RHEL 8.4 以降、**libreswan** パッケージは、TCP カプセル化を使用した IPsec ベースの VPN に対応していましたが、**selinux-policy** パッケージはこの更新を反映しませんでした。その結果、Libreswan が TCP を使用するように設定されている場合、**ipsec** サービスは指定の TCP ポートにバインドできませんでした。今回の **selinux-policy** パッケージの更新により、**ipsec** サービスが一般的に使用されている TCP ポート **4500** にバインドして接続できるようになったため、IPsec ベースの VPN で TCP カプセル化を使用できるようになりました。

(BZ#1931848)

SELinux ポリシーが **staff_u** ユーザーの **unconfined_r** への切り替えを防止するようになりました。

以前のバージョンでは、**secure_mode** ブール値を有効にすると、**staff_u** ユーザーが誤って **unconfined** ロールに切り替えることができました。これにより、**staff_u** ユーザーは、システムのセキュリティに影響する特権操作を実行できました。今回の修正により、SELinux ポリシーにより、**staff_u** ユーザーが **newrole** コマンドを使用して **unconfined_r** ロールに切り替えられなくなりました。したがって、特権のないユーザーは特権操作を実行できません。

(BZ#1947841)

OSCAP Anaconda Addon がカスタマイズされたプロファイルを処理

以前は、**OSCAP Anaconda Addon** プラグインは、個別のファイルでカスタマイズしたセキュリティープロファイルを正しく処理しませんでした。したがって、対応する Kickstart セクションで指定した場合でも、カスタマイズしたプロファイルは RHEL グラフィカルインストールで利用できませんでした。処理が修正され、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るようになりました。

(BZ#1691305)

STIG プロファイルおよびその他の SCAP コンテンツの評価中に OpenSCAP が失敗しなくなりました。

以前は、OpenSCAP の暗号ライブラリーの初期化は、OpenSCAP で適切に実行されませんでした(特に **filehash58** プローブ)。その結果、**filehash58_test** Open Vulnerability Assessment Language(OVAL) テストを含む SCAP コンテンツの評価中にセグメンテーション違反が発生しました。これは、Red Hat Enterprise Linux 8 用の STIG プロファイルの評価に影響を及ぼしました。評価は予期せず失敗し、結果が生成されませんでした。**openscap** パッケージの新しいバージョンで、ライブラリーを初期化するプロセスが修正されました。その結果、RHEL 8 用の STIG プロファイル、および **filehash58_test** OVAL テストを含むその他の SCAP コンテンツの評価時に、OpenSCAP が失敗しなくなりました。

(BZ#1959570)

Ansible は必要な場合にのみバナーファイルを更新します

以前のバージョンでは、バナー修復に使用する Playbook は常にファイルを削除し、再作成していました。

た。そのため、バナーファイルの inode はその必要がなくても常に変更されていました。今回の更新により、Ansible 修復 Playbook が改善され、**copy** モジュールを使用して、既存のコンテンツを目的のコンテンツと比較し、必要な場合にのみファイルを更新するようになりました。その結果、既存のコンテンツが目的のコンテンツとは異なる場合にのみ、バナーファイルは更新されます。

(BZ#1857179)

USB デバイスが DISA STIG プロファイルで正常に動作するようになりました。

以前は、DISA STIG プロファイルが **USBGuard** サービスを有効にしましたが、初期接続された USB デバイスを設定していませんでした。したがって、**USBGuard** サービスは、特に許可されていないデバイスをすべてブロックしていました。これにより、スマートカードなどの一部の USB デバイスが到達不能になっていました。今回の更新で、DISA STIG プロファイルの適用時に初期 USBGuard 設定が生成され、接続された USB デバイスを使用できるようになりました。その結果、USB デバイスがブロックされず、正常に機能します。

(BZ#1946252)

OSCAP Anaconda Addon が選択されたすべてのパッケージをテキストモードでインストール

以前は、**OSCAP Anaconda Addon** プラグインは、テキストモードで実行している際インストールの開始前に、特定のパーティションレイアウトまたはパッケージのインストールと削除が必要なルールを評価しませんでした。そのため、キックスタートを使用してセキュリティーポリシープロファイルが指定され、インストールがテキストモードで実行されている場合は、選択したセキュリティープロファイルに必要な追加パッケージがインストールされませんでした。**OSCAP Anaconda Addon** は、インストールがグラフィカルまたはテキストベースであるかに関係なく、インストールの開始前に必要なチェックを実行するようになりました、選択したすべてのパッケージもテキストモードでインストールされます。

(BZ#1674001)

rpm_verify_permissions が CIS プロファイルから削除

ファイルパーミッションとパッケージのデフォルトパーミッションを比較する **rpm_verify_permissions** ルールは、Center for Internet Security (CIS) Red Hat Enterprise Linux 8 Benchmark から削除されました。今回の更新で、CIS プロファイルが CIS RHEL 8 ベンチマークと整合し、その結果、このルールは CIS に応じてシステムを強化するユーザーに影響を与えなくなりました。

(BZ#1843913)

7.5. カーネル

アップストリームパッチをやり直すことで、一部の **systemd** サービスやユーザー空間のワークロードを期待どおりに実行できるようになります。

バックアップとして提供されている **mknod()** システムコールのアップストリームの変更により、**open()** システムコールでは、デバイスノードに対する特権が **mknod()** よりも高くなりました。そのため、複数のユーザー空間のワークロードや、コンテナー内の一部の **systemd** サービスが応答しなくなりました。今回の更新で、間違った動作が元に戻り、クラッシュが発生しなくなりました。

(BZ#1902543)

メモリーアカウンティング操作でパフォーマンスリグレッションが改善されました。

スラブメモリーコントローラーは、スラブごとのメモリーアカウンティング操作の周波数を増加させていました。その結果、メモリーアカウンティング操作の数が増えたため、パフォーマンスが低下しました。この問題を修正するため、メモリーアカウンティング操作が、できるだけ多くのキャッシングを使用

し、アトミック操作を最小限にとどめるように合理化されました。その結果、わずかなパフォーマンスのリグレッションが持続します。ただし、ユーザーエクスペリエンスは大幅に改善されています。

(BZ#1959772)

複数の SysRq-T マジックキーを発行すると、ハードロックが発生し、システムパニックが発生しなくなりました。

システムに複数の SysRq-T マジックキーシーケンスを発行すると、シリアルコンソールの速度や、出力される情報のボリュームによっては、中断が長期間無効になります。この長期間の無効割り込み時間により、多くの場合、ハードロックアップが発生し、その後システムパニックが発生します。今回の更新で、SysRq-T キーシーケンスが追加され、割り込みが無効になっている期間が大幅に短縮されました。その結果、上記のシナリオでハードロックアップやシステムパニックが発生することはありません。

(BZ#1954363)

特定の BCC ユーティリティーが macro redefined 警告を表示しない

一部のコンパイラー固有のカーネルヘッダーのマクロ再定義により、いくつかの BPF Compiler Collection (BCC) ユーティリティーが以下の影響なしの警告が表示していました。

```
warning: '__no_SANITIZE_address' macro redefined [-Wmacro-redefined]
```

今回の更新で、マクロの再定義を削除して問題が修正されています。その結果、該当する BCC ユーティリティーは、このシナリオで警告を表示しなくなりました。

(BZ#1907271)

kdump が SSH または NFS ターゲットの vmcore のダンプに失敗しない

以前は、ネットワークインターフェイスカード (NIC) ポートを静的 IP アドレスに設定し、**kdump** が SSH または NFS ダンプターゲット上の **vmcore** をダンプするように設定すると、**kdump** サービスは以下のエラーメッセージと共に開始しました。

```
ipcalc: command not found
```

したがって、SSH または NFS ダンプターゲット上の **kdump** は最終的に失敗していました。

今回の更新で問題が修正され、**kexec-tools** ユーティリティーが IP アドレスとネットマスクの計算を **ipcalc** ツールに依存しなくなりました。これにより、SSH または NFS ダンプターゲットを使用する場合に **kdump** が期待どおりに機能します。

(BZ#1931266)

特定のネットワークカーネルドライバーが、そのバージョンを適切に表示する

RHEL 8.4 では、多くのネットワークカーネルドライバーのモジュールバージョン管理の動作が変更になりました。そのため、これらのドライバーはそのバージョンを表示しませんでした。また、**ethtool -i** コマンドを実行すると、ドライバーはドライバーバージョンではなく カーネル バージョンを表示しました。今回の更新で、カーネルモジュールの文字列を提供することでバグが修正されました。その結果、影響を受けるカーネルドライバーのバージョンを判別できます。

(BZ#1944639)

hwloc コマンドが、単一の CPU Power9 および Power10 の論理パーティションで正しいデータを返すようになりました。

バージョン 2.2.0 の **hwloc** ユーティリティーを使用すると、Power9 または Power10 CPU を実行する単一ノードの Non-Uniform Memory Access(NUMA) システムは、disallowed と見なされました。そのため、NODE0(ソケット 0、CPU 0) がオフラインで、**hwloc** ソースコードは NODE0 がオンラインであることを想定するので、すべての **hwloc** コマンドは機能しませんでした。以下のエラーメッセージが表示されました。

Topology does not contain any NUMA node, aborting!

今回の更新により、**hwloc** が修正され、ソースコードはクエリー前に NODE0 がオンラインかどうかを確認します。NODE0 がオンラインではない場合、コードは次のオンライン NODE に進みます。

その結果、**hwloc** コマンドは上記のシナリオでエラーを返しなくなりました。

(BZ#1917560)

7.6. ファイルシステムおよびストレージ

getaddrinfo() から取得したレコードにデフォルトの TTL が含まれるようになりました

以前は、API は Time-to-Live (TTL) 情報を伝達しなかったため、たとえ DNS から取得されたものであっても、**getaddrinfo()** によって取得されたアドレスレコードに対して TTL が設定されていませんでした。その結果、SRV や AFSDB レコードなど、DNS から直接取得したコンポーネントがレコードに含まれていない限り、**key.dns_resolver** upcall プログラムは **dns_resolver** レコードに有効期限を設定しませんでした。今回の更新により、**getaddrinfo()** からのレコードには、未設定の有効期限を防ぐために 10 分のデフォルト TTL が含まれるようになりました。

(BZ#1661674)

7.7. 高可用性およびクラスター

ocf:heartbeat:pgsql リソースエージェントおよび一部のサードパーティーエージェントがシャットダウンプロセス中に停止に失敗しなくなりました。

RHEL 8.4 GA リリースでは、Pacemaker の **crm_mon** コマンドラインツールが変更され、Pacemaker のシャットダウンを開始するときに通常のクラスター情報ではなくシャットダウンのメッセージが表示されます。そのため、リソースの停止などのシャットダウンの進捗を監視できませんでした。この場合、resource-agents パッケージで配布される **ocf:heartbeat:pgsql** エージェントやカスタムまたはサードパーティーエージェントなど、停止操作で **crm_mon** 出力を解析するリソースエージェントは、クラスターの停止に失敗する可能性があり、クラスターの問題が発生する可能性がありました。このバグは修正され、上記の問題が発生しなくなりました。

(BZ#1948620)

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

pyodbc が MariaDB 10.3 で再び動作するようになりました。

pyodbc モジュールは、RHEL 8.4 リリースに含まれる **MariaDB 10.3** サーバーでは機能しませんでした。**mariadb-connector-odbc** パッケージの根本的な原因が修正され、**pyodbc** が期待通りに **MariaDB 10.3** と動作するようになりました。

MariaDB 10.3 サーバーの以前のバージョンおよび **MariaDB 10.5** サーバーは、この問題の影響を受けなかった点に注意してください。

(BZ#1944692)

7.9. コンパイラーや開発ツール

GCC Toolset 11: GCC 11 が DWARF 4 にデフォルト設定

アップストリームの GCC 11 は、デフォルトで DWARF 5 デバッグ形式を使用していますが、GCC Toolset 11 の GCC は、RHEL 8 コンポーネント (**rpmbuild** など) との互換性を維持するために DWARF 4 にデフォルト設定されます。

(BZ#1974402)

パラメーターフームワークが **GLIBC_TUNABLES** を正しく解析するようになりました。

以前のバージョンでは、パラメーターフームワークは、setuid プログラムの setuid が設定されていない子オブジェクトについて **GLIBC_TUNABLES** 環境変数を正しく解析しませんでした。そのため、setuid プログラムの setuid が設定されていない子オブジェクトでは、すべてのパラメーターが残ることがありました。今回の更新で、**GLIBC_TUNABLES** 環境変数のパラメーターが正しく解析されるようになりました。その結果、特定されたパラメーターの制限されたサブセットのみが、setuid プログラムの setuid が設定されていない子オブジェクトによって継承されるようになりました。

(BZ#1934155)

glibc の **semctl** システムコールラッパーが **SEM_STAT_ANY** を **SEM_STAT** のように処理するようになりました。

以前は、**glibc** の **semctl** システムコールラッパーは、カーネル引数 **SEM_STAT_ANY** を **SEM_STAT** のように処理していました。そのため、**glibc** は結果オブジェクトの **struct semid_ds** のアドレスをカーネルに渡さなかったため、カーネルが更新に失敗していました。今回の更新で、**glibc** は **SEM_STAT_ANY** を **SEM_STAT** のように処理し、その結果、アプリケーションは **SEM_STAT_ANY** を使用して **struct semid_ds** データを取得できるようになりました。

(BZ#1912670)

glibc に **IPPROTO_ETHERNET**、**IPPROTO_MPTCP**、および **INADDR_ALLSNOOPERS_GROUP** の定義が含まれるようになりました。

以前は、**Glibc** システムライブラリーヘッダー (**/usr/include/netinet/in.h**) には **IPPROTO_ETHERNET**、**IPPROTO_MPTCP**、および **INADDR_ALLSNOOPERS_GROUP** の定義が含まれていませんでした。そのため、これらの定義を必要とするアプリケーションをコンパイルできませんでした。今回の更新で、システムライブラリーヘッダーに、**IPPROTO_ETHERNET**、**IPPROTO_MPTCP**、および **INADDR_ALLSNOOPERS_GROUP** の新しいネットワーク定数定義が追加され、アプリケーションを正しくコンパイルするようになりました。

(BZ#1930302)

gcc がバージョン 8.5 にリベース

GNU コンパイラーコレクション (GCC) がアップストリームバージョン 8.5 にリベースされ、以前のバージョンにバグ修正が数多く追加されました。

(BZ#1946758)

OpenSSL aes-cbc モード使用時の間違ったファイルの復号化

OpenSSL EVP **aes-cbc** モードはファイルを正しく復号しませんでした。Go CryptoBlocks インターフェイスが完全なブロックを想定しているのに対し、パディングを処理することを想定しているためです。この問題は、OpenSSL で EVP 操作を実行する前にパディングを無効にすることによって修正されました。

(BZ#1979100)

7.10. IDENTITY MANAGEMENT

ブートストラップスクリプトの実行時に、FreeRADIUS がデフォルトの証明書を誤って生成しなくなりました。

FreeRADIUS が起動するたびに、ブートストラップスクリプトが実行されます。以前のバージョンでは、このスクリプトは **/etc/raddb/certs** ディレクトリーに新しいテスト証明書を生成し、その結果、これらのテスト証明書は無効なので、FreeRADIUS サーバーが起動できないことがありました。たとえば、証明書の有効期限が切れている場合があります。今回の更新により、ブートストラップスクリプトは **/etc/raddb/certs** ディレクトリーをチェックし、テストまたはカスタマー証明書が含まれている場合は、スクリプトが実行されず、FreeRADIUS サーバーが正しく起動するはずです。

テスト証明書は、FreeRADIUS の設定中のテスト目的のみで、実際の環境では使用しないでください。ユーザーの証明書が使用されたら、ブートストラップスクリプトを削除する必要があります。

(BZ#1954521)

FreeRADIUS がコアダンプファイルの作成に失敗しなくなる

以前では、**allow_core_dumps** を **yes** に設定した場合に、FreeRADIUS がコアダンプファイルを作成していました。そのため、いずれかのプロセスに失敗した場合は、コアダンプファイルが作成されませんでした。今回の更新で、**allow_core_dumps** を **yes** に設定すると、いずれかのプロセスが失敗した場合に、FreeRADIUS がコアダンプファイルを作成するようになりました。

(BZ#1977572)

SSSD が、/etc/krb5.conf の Kerberos キータブ名のデフォルト設定を正しく評価

以前は、**krb5.keytab** ファイルの標準以外の場所を定義した場合は、SSSD はこの場所を使用せず、代わりにデフォルトの **/etc/krb5.keytab** の場所を使用していました。したがって、システムへのログイン試行時に、**/etc/krb5.keytab** にエントリーが含まれていないため、ログインに失敗していました。

今回の更新で、SSSD は **/etc/krb5.conf** の **default_keytab_name** 変数を評価し、この変数で指定された場所を使用するようになりました。**default_keytab_name** 変数が設定されていない場合にのみ、SSSD はデフォルトの **/etc/krb5.keytab** の場所を使用します。

(BZ#1737489)

sudo コマンドを実行しても、KRB5CCNAME 環境変数をエクスポートしなくなりました。

以前のバージョンでは、**sudo** コマンドの実行後に、環境変数 **KRB5CCNAME** は、元のユーザーの Kerberos 認証情報キャッシュを参照していましたが、ターゲットユーザーがアクセスできない場合がありました。そのため、このキャッシュにアクセスできないため、Kerberos 関連の操作が失敗する可能性

がありました。今回の更新で、**sudo** コマンドを実行しても **KRB5CCNAME** 環境変数が設定されなくなり、ターゲットユーザーがデフォルトの Kerberos 認証情報キャッシュを使用できるようになりました。

(BZ#1879869)

Kerberos が許可された暗号化タイプのみを要求

以前では、**default_tgs_enctypes** または **default_tkt_enctypes** パラメーターが設定されていないと、RHEL は **/etc/krb5.conf** ファイルの **permitted_enctypes** パラメーターで指定された許可された暗号化タイプを適用しませんでした。そのため、Kerberos クライアントは RC4 などの非推奨の暗号スイートをリクエストでき、他のプロセスが失敗する可能性がありました。今回の更新で、RHEL は **permitted_enctypes** に設定した暗号化タイプをデフォルトの暗号化タイプにも適用し、プロセスは許可された暗号化タイプのみを要求できるようになりました。

Red Hat Identity Management(IdM) を使用して、Active Directory(AD) との信頼を設定する場合は、RHEL 8 で非推奨となった RC4 暗号スイートが、ユーザー、サービス、および AD フォレストの AD ドメイン間の信頼のデフォルト暗号化タイプであることに注意してください。以下のオプションのいずれかを使用できます。

- (推奨): AD で強力な AES 暗号化タイプを有効にします。詳細は、Microsoft のアーティクル [AD DS: Security: Kerberos "Unsupported etype" error when accessing a resource in a trusted domain](#) を参照してください。
- AD ドメインのメンバーである必要のある RHEL ホストで **update-crypto-policies --set DEFAULT:AD-SUPPORT** コマンドを使用して、AD との後方互換性を確保するために非推奨の RC4 暗号化タイプを有効にします。

(BZ#2005277)

レプリケーションセッションの更新速度が向上しました。

これまで、エンジログに大きな更新が含まれている場合、レプリケーションセッションはエンジログの先頭から開始されていました。これでセッションが遅くなってしまいました。これは、レプリケーションセッション中に、エンジログからの更新情報を格納するために小さなバッファーを使用したことが原因です。この更新で、レプリケーションセッションは、開始の時点でバッファーが更新を保存するのに十分な大きさであることをチェックします。レプリケーションセッションは、すぐに更新の送信を開始します。

(BZ#1898541)

プラグインで作成したデータベースのインデックスが有効になる

これまで、サーバープラグインが独自のデータベースインデックスを作成した場合、それらのインデックスを手動で有効にする必要がありました。今回の更新では、デフォルトでインデックスの作成直後から有効になります。

(BZ#1951020)

7.11. RED HAT ENTERPRISE LINUX システムロール

同じ出力を実行する際にロールタスクが変わらない

以前のバージョンでは、同じ入力をもう一度実行しても、変更がない場合でも、ロールタスクの一部は **CHANGED** として報告されていました。そのため、ロールはべき等性を持ちませんでした。この問題を修正するには、以下のアクションを実行します。

- 設定変数の変更を確認してから、それらを適用します。この検証には `--check` オプションを使用できます。
- **Last Modified: \$date** ヘッダーを設定ファイルに追加しないでください。

その結果、ロールタスクはべき等になります。

(BZ#1960375)

Postfix ドキュメントに relayhost パラメーターが不適切に定義されなくなりました

以前は、Postfix RHEL システムロールの `relayhost` パラメーターは、`rhel-system-roles` で提供される `doc /usr/share/doc/rhel-system-roles/postfix/README.md` ドキュメントで `relay_host` として定義されていました。今回の更新で問題が修正され、Postfix ドキュメントに `relayhost` パラメーターが正しく定義されるようになりました。

(BZ#1866544)

Postfix RHEL システムロールの README.md の "Role Variables" セクションで欠落していた変数がなくなりました

以前は、`postfix_check`、`postfix_backup`、`postfix_backup_multiple` などの Postfix RHEL システムロール変数は、Role Variables セクションでは使用できません。そのため、Postfix ロールのドキュメントを参照できませんでした。今回の更新で、Postfix README セクションに、ロール変数のドキュメントが追加されました。ロール変数は、`rhel-system-roles` が提供する `doc/usr/share/doc/rhel-system-roles/postfix/README.md` ドキュメントで文書化され、ユーザーが利用できるようになりました。

(BZ#1961858)

postfix ロール README がプレーンロール名を使用しなくなる

以前は、`/usr/share/ansible/roles/rhel-system-roles.postfix/README.md` で提供される例では、`rhel-system-roles.postfix` ではなく、プレーンバージョンのロール名 `postfix` を使用していました。そのため、ユーザーはドキュメントを参照し、完全修飾ロール名 (FQRN) ではなくプレーンロール名を誤って使用していました。今回の更新で問題が修正され、ドキュメントでは FQRN、`rhel-system-roles.postfix` の例が含まれ、ユーザーは Playbook を正しく作成できるようになりました。

(BZ#1958963)

timesync の出力ログでは、有害なエラーしか報告されません

以前では、`timesync` RHEL システムロールは、`ignore_errors` ディレクティブを使用し、多くのタスクでタスクの失敗を個別にチェックしていました。そのため、成功したロール実行の出力ログには、安全性に問題のないエラーが多数含まれていました。ユーザーはこれらのエラーを無視することは安全ですが、それでも見ることは負担でした。今回の更新により、関連するタスクが `ignore_errors` を使用しないように書き換えられました。その結果、出力ログがクリーンになり、ロール停止エラーのみが報告されるようになりました。

(BZ#1938014)

requirements.txt ファイルが Ansible コレクションで欠落しなくなりました。

以前のバージョンでは、python 依存関係を指定する `requirements.txt` ファイルが Ansible コレクションで欠落していました。今回の修正により、正しい依存関係がある欠落していたファイルが `/usr/share/ansible/collections/ansible_collections/redhat/rhel_system_roles/requirements.txt` パスに追加されました。

(BZ#1954747)

storage_pools に **type: partition** が設定されているときに **Traceback** が確認されなくなりました

以前のバージョンでは、Playbook の **storage_pools** に変数 **type** を **partition** として設定すると、この Playbook の実行に失敗し、**traceback** が示されていました。今回の更新で問題が修正され、**Traceback** エラーが表示されなくなりました。

(BZ#1854187)

SELinux ロールが不要なリロードを実行しなくなる

以前は、**SELinux** ロールは、**SELinux** ポリシーを再読み込みする前に、実際に変更が適用されたかどうかを確認しませんでした。これにより、**SELinux** ポリシーが不必要に再読み込みされ、システムリソースに影響がありました。今回の修正により、**SELinux** ロールは Ansible ハンドラーおよび条件を使用して、変更がある場合にのみポリシーが再読み込みされるようになりました。これにより、**SELinux** ロールはより迅速に実行されます。

(BZ#1757869)

RHEL6 ホストにインストールされた **sshd_config** ファイルで **sshd** ロールの起動に失敗することがなくなりました。

これまで、管理ノードが RHEL6 を実行している場合、インストールタスクによって追加されたマッチ条件で、OpenSSH のバージョンが"Match all"をサポートしていませんでした。その結果、RHEL6 ホストにインストールされた **sshd_config** ファイルでの **sshd** の起動に失敗しました。この更新では、RHEL6 の **sshd_config** 設定ファイルの "Match all" を "Match address *" に置き換えることで、この問題を修正しました。この条件は OpenSSH のバージョンでサポートされるためです。その結果、RHEL6 ホストにインストールされた **sshd_config** ファイルを使用して、**sshdRHEL** システムロールが正常に起動します。

(BZ#1990947)

README.md の例にある **SSHD** ロール名に誤りが無くなる

これまで、**sshd** の **README.md** ファイルの中で、**willshersystems.sshd** という名前のロールを呼び出す例を紹介していました。今回の更新により問題が修正され、参照例では **rhel_system_roles.sshd** というロールが正しく参照されるようになりました。

(BZ#1952090)

tls が **false** の場合、**key/certs** のソースファイルがコピーされなくなりました。

これまで、**logging** RHEL システムロールの **elasticsearch** 出力において、コントロールホスト上の **key/certs** ソースファイルのパスが Playbook で設定されている場合、**tls** が **false** に設定されていても、管理下のホストにコピーされていました。そのため、**key/cert** のファイルパスが設定され、**tls** が **false** に設定されていると、コピー元のファイルが存在しないため、コマンドが失敗してしまいます。今回の更新により問題が修正され、**tls** パラメーターが **true** に設定されている場合にのみ、**key/certs** のコピーが実行されるようになりました。

(BZ#1994580)

metric ロールで対象ホストのログを有効にするタスクが動作するようになりました。

これまで、**metric** RHEL システムロールのバグにより、**enabling the performance metric logging** タスクで対象となるホストを参照することができませんでした。その結果、パフォーマンスマトリックのロギング用のコントロールファイルが生成されませんでした。この更新により問題が修正され、対象

となるホストが正しく参照されるようになりました。その結果、コントロールファイルが正常に作成され、パフォーマンスマトリックのロギング実行が可能になります。

(BZ#1967335)

sshd_hostkey_group および sshd_hostkey_mode 変数が Playbook で設定可能になりました。

以前は、**sshd_hostkey_group** および **sshd_hostkey_mode** 変数が **defaults** および **vars** ファイルの両方で意図せずに定義されていました。そのため、ユーザーは Playbook でこれらの変数を設定することができませんでした。この修正により、**vars** ファイルで定数値を定義するため **sshd_hostkey_group** が **_sshd_hostkey_group** に、**sshd_hostkey_mode** が **_sshd_hostkey_mode** に名前が変更されます。**default** ファイルでは、**sshd_hostkey_group** には **_sshd_hostkey_group** が、**sshd_hostkey_mode** には **_sshd_hostkey_mode** が設定されています。その結果、ユーザーは Playbook で **sshd_hostkey_group** および **sshd_hostkey_mode** 変数を設定できるようになりました。

(BZ#1966711)

README.md の RHEL システムロールの内部リンクがリンク切れにならない

これまで、**README.md** ファイルで利用できる内部リンクがリンク切れになっていました。そのため、ユーザーが特定のセクションのドキュメントリンクをクリックしても、特定の **README.md** セクションにはリダイレクトされませんでした。今回の更新では、この問題が修正され、内部リンクが正しいセクションを示すようになりました。

(BZ#1962976)

7.12. クラウド環境の RHEL

nm-cloud-setup ユーティリティーが Microsoft Azure に正しいデフォルトルートを設定する

以前のリリースでは、Microsoft Azure では、**nm-cloud-setup** ユーティリティーがクラウド環境の正しいゲートウェイを検出できませんでした。これにより、ユーティリティーは誤ったデフォルトルートを設定し、接続に失敗していました。今回の更新でこの問題が修正されています。その結果、**nm-cloud-setup** ユーティリティーが Microsoft Azure に正しいデフォルトルートを設定するようになりました。

(BZ#1912236)

SSH 鍵が、バックアップ AMI から作成された EC2 インスタンスで正しく生成されるようになりました。

以前のリリースでは、バックアップ Amazon Machine Image (AMI) から RHEL 8 の新しい Amazon EC2 インスタンスを作成する場合、**cloud-init** は仮想マシン上の既存の SSH キーを削除しましたが、新しい SSH キーは作成しませんでした。したがって、仮想マシンがホストに接続できない場合がありました。

この問題は、新たに作成された RHEL 8.5 仮想マシンに対して修正されました。RHEL 8.4 以前からアップグレードした仮想マシンの場合、この問題は手動で回避する必要があります。

これを行うには、**cloud.cfg** ファイルを編集し、**ssh_genkeytypes: ~** 行を **ssh_genkeytypes: ['rsa', 'ecdsa', 'ed25519']** に変更します。これにより、上記の状況で RHEL 8 仮想マシンをプロビジョニングする際に、SSH キーを削除して正しく生成できるようになります。

(BZ#1957532)

AWS ARM64 インスタンスで実行している RHEL 8 が、指定されたネットワーク速度に到達できるようになりました。

Amazon Web Services (AWS) ARM64 インスタンスで実行される仮想マシンで 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場合は、**iommu.strict=1** カーネルパラメーターが使用されたとき、または、**iommu.strict** カーネルパラメーターが定義されなかったとき、これまで仮想マシンのネットワークパフォーマンスは予想されるよりも低くなりました。

この問題は、Red Hat が提供する RHEL 8.5 Amazon Machine Images (AMI) では発生しなくなりました。別のタイプのイメージでは、パラメーターを **iommu.strict=0** に変更することで問題を回避できます。ここでは、以下の点を説明します。

- RHEL 8.4 以前のイメージ
- **yum update** を使用して以前のバージョンからアップグレードした RHEL 8.5 イメージ
- Red Hat 以外の RHEL 8.5 イメージ

(BZ#1836058)

RHEL 8 の仮想マシンを Azure 上のリモートマシンにコアダンプする操作が、より確実に動作するようになりました。

以前のリリースでは、**kdump** ユーティリティーを使用した、Microsoft Azure ハイパーテイラー上の RHEL 8 仮想マシンのコアダンプファイルのリモートマシンへの保存は、仮想マシンがネットワークアクセラレーションを有効化して NIC を使用している場合は適切に動作しませんでした。これにより、ダンプファイルは即座にではなく、約 200 秒後に保存されました。さらに、ダンプファイルを保存する前に、以下のエラーメッセージがコンソールに記録されました。

device (eth0): linklocal6: DAD failed for an EUI-64 address

今回の更新で、基礎となるコードが修正されたため、上記の状況では、ダンプファイルがすぐに保存されるようになりました。

(BZ#1854037)

FIPS モードが有効になると、RHEL 8 ゲストの休止状態が正しく機能するようになりました。

仮想マシンが FIPS モードを使用している場合は、RHEL 8 をゲストオペレーティングシステムとして使用していた仮想マシンを休止状態にすることはできませんでした。基本となるコードが修正され、影響を受ける仮想マシンが正しく休止状態にできるようになりました。

(BZ#1934033, BZ#1944636)

7.13. コンテナー

UBI 9-Beta コンテナーは、RHEL 7 および 8 ホストで実行できます。

UBI 9-Beta コンテナーイメージでは、**containers-common** パッケージに間違った seccomp プロファイルが設定されていました。そのため、コンテナーが特定のシステムコールに対応できず、障害が発生しました。今回の更新で、この問題が修正されています。

(BZ#2019901)

第8章 テクノロジープレビュー

本パートでは、Red Hat Enterprise Linux 8.5 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。

8.1. シェルおよびコマンドラインツール

ReaR は、64 ビット IBM Z アーキテクチャーでテクノロジープレビューとして利用できます。

Basic Relax and Recover (ReaR) 機能が、64 ビットの IBM Z アーキテクチャーでテクノロジープレビューとして利用できるようになりました。IBM Z では、z/VM 環境でのみ ReaR レスキューイメージを作成できます。論理パーティション (LPAR) のバックアップおよび復元はテストされていません。

現在利用できる出力方法は、Initial Program Load (IPL) のみです。IPL は、**zIPL** ブートローダーで使用できるカーネルと初期 ramdisk (initrd) を生成します。

詳細は、[64 ビット IBM Z アーキテクチャーで ReaR レスキューイメージの使用](#) を参照してください。

(BZ#1868421)

8.2. ネットワーキング

KTLS がテクノロジープレビューとして利用可能に

RHEL は、テクノロジープレビューとして KTLS (Kernel Transport Layer Security) を提供します。

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)

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)

テクノロジープレビューとして利用できる XDP 機能

Red Hat は、以下の eXpress Data Path (XDP) 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ます。

- AMD および Intel 64 ビット以外のアーキテクチャーで XDP プログラムを読み込む。**libxdp** ライブラリーは、AMD および Intel 64 ビット以外のアーキテクチャーでは使用できません。
- XDP ハードウェアオフロード。

(BZ#1889737)

TC のマルチプロトコルラベルスイッチがテクノロジープレビューとして利用可能に

Multi-protocol Label Switching (MPLS) は、エンタープライズネットワーク全体でトラフィックフローをルーティングするカーネル内データ転送メカニズムです。MPLS ネットワークでは、パケットを受信するルーターは、パケットに割り当てられたラベルに基づいて、パケットの追加のルートを決定します。ラベルを使用すると、MPLS ネットワークは特定の特性を持つパケットを処理する機能があります。たとえば、特定ポートから受信したパケットの管理や、特定のタイプのトラフィックを一貫した方法で伝送する **tc filters** を追加できます。

パケットがエンタープライズネットワークに入ると、MPLS ルーターは、パケット上で複数の操作を実行します。ラベルの追加には **push**、**swap** (ラベルの更新)、ラベルの削除の **pop** などが含まれます。MPLS では、RHEL の1つまたは複数のラベルに基づいて、アクションをローカルに定義できます。ルーターを設定し、トラフィック制御 (**tc**) フィルターを設定して、**label**、**traffic class**、**bottom of stack**、**time to live** などの MPLS ラベルスタックエントリー (**lse**) 要素に基づいて、パケットに対して適切なアクションを実行するように設定することができます。

たとえば、次のコマンドは、フィルターを **enp0s1** ネットワークインターフェイスに追加して、最初のラベル 12323 と 2 番目のラベル 45832 を持つ着信パケットと一致させます。一致するパケットでは、以下のアクションが実行されます。

- 最初の MPLS TTL はデクリメントされます (TTL が 0 に達するとパケットがドロップされます)。
- 最初の MPLS ラベルが 549386 に変更
- 作成されるパケットは **enp0s2** 経由で送信されます。宛先 MAC アドレス 00:00:5E:00:53:01、およびソース MAC アドレス 00:00:5E:00:53:02。

```
# tc filter add dev enp0s1 ingress protocol mpls_uc flower mpls lse depth 1 label 12323 lse
depth 2 label 45832 \
action mpls dec_ttl pipe \
action mpls modify label 549386 pipe \
action pedit ex munge eth dst set 00:00:5E:00:53:01 pipe \
action pedit ex munge eth src set 00:00:5E:00:53:02 pipe \
action mirred egress redirect dev enp0s2
```

(BZ#1814836, [BZ#1856415](#))

act_mpls モジュ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。

act_mpls モジュールが、テクノロジープレビューとして **kernel-modules-extra** rpm で利用可能になりました。モジュールを使用すると、トラフィック制御 (TC) フィルターを使用した Multiprotocol Label Switching (MPLS) アクション (TC フィルターを使用した MPLS ラベルスタックエントリーの push や pop など) の適用が可能になります。また、このモジュールでは、Label、Traffic Class、Bottom of Stack、および Time to Live フィールドを独立して設定できます。

(BZ#1839311)

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 サブリゾルバー、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ダー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)

nispor パッケ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。

nispor パッケージ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。これは、Linux ネットワーク状態クエリーの統合インターフェイスです。これにより、Python および C API と rust crate を使用して、実行中のすべてのネットワークのステータスにクエリーを実行することができます。**nispor** は、**nmstate** ツールの依存関係として機能します。

nispor パッケージは、**nmstate** の依存関係、または個々のパッケージとしてインストールできます。

- **nispor** を個別のパッケージとしてインストールするには、次のコマンドを実行します。

```
# yum install nispor
```

- **nispor** を **nmstate** の依存関係としてインストールするには、次のコマンドを実行します。

```
# yum install nmstate
```

nispor は依存関係としてリスト表示されます。

nispor の使用の詳細は、</usr/share/doc/nispor/README.md> ファイルを参照してください。

(BZ#1848817)

8.3. カーネル

kexec fast reboot 機能は、テクノロジープレビューとしてご利用いただけます。

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** は、カーネルが先に BIOS (Basic Input/Output System) を経由せずに、2 番目のカーネルを直接起動できるようにすることで、ブートプロセスの時間を大幅に短縮します。この機能を使用するには、以下を実行します。

1. **kexec** カーネルを手動で読み込みます。
2. オペレーティングシステムを再起動します。

(BZ#1769727)

accel-config パッケ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。

accel-config パッケージが、テクノロジープレビューとして、Intel **EM64T** および **AMD64** アーキテクチャーで利用可能になりました。このパッケージは、Linux カーネルでデータストリーミング (DSA) サブシステムを制御し、設定するのに役立ちます。また、**sysfs** (pseudo-filesystem) を介してデバイスを設定し、設定を **json** 形式で保存および読み込みます。

(BZ#1843266)

SGX 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。

Software Guard Extensions (SGX) は、ソフトウェアコードおよび公開および修正からのデータを保護する Intel® テクノロジーです。RHEL カーネルは、SGX v1 および v1.5 を部分的にサポートします。バージョン 1 では、**Flexible Launch Control** メカニズムを使用するプラットフォームが SGX テクノロジーを使用できるようにします。

(BZ#1660337)

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。
- **AF_XDP**。これは、eXpress Data Path (XDP) パスを、パケット処理のパフォーマンスを優先するアプリケーションのユーザー空間に接続するためのソケットです。

(BZ#1559616)

カーネルの Intel データストリーミングタブレットドライバーがテクノロジープレビューとして利用可能になる

カーネルの Intel データストリーミングアクセラレータードライバー (IDXD) は、現在テクノロジープレビューとして利用できます。これは Intel CPU が統合され、プロセスアドレス空間 ID (pasid) の送信および共有仮想メモリー (SVM) の共有ワークキューをサポートします。

(BZ#1837187)

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)

stmmac ドライバーがテクノロジープレビューとして利用可能に

Red Hat は、チップ (SoC) での Intel® Elkhart Lake システムの **stmmac** の使用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ます。

(BZ#1905243)

8.4. ファイルシステムおよびストレージ

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

Red Hat Enterprise Linux 8 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ーや DVD-ROM などのファイルシステムイメージを、複数のユーザーが共有できるようになります。

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

以下の制限下で、対応しているコンテナーエンジン (**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、コンテナーエンジングラフドライバーとしての使用、または圧縮された **kdump** **initramfs** などのその他の特殊なユースケースとしての使用のみサポートされています。その使用は主にコンテナー COW コンテンツでサポートされており、永続ストレージではサポートされていません。非 OverlayFS ボリュームに永続ストレージを配置する必要があります。デフォルトのコンテナー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** の更新を受け取ません。
 - O_RDONLY** で開いてから、**MAP_SHARED** でマッピングした下位ファイルは、後続の変更と一貫性がありません。
 - 完全に準拠した **st_ino** 値または **d_ino** 値は、RHEL 8 ではデフォルトで有効になっていませんが、モジュールオプションまたはマウントオプションを使用して、この値の完全な POSIX コンプライアンスを有効にできます。
一貫した inode 番号を付けるには、**xino=on** マウントオプションを使用します。

redirect_dir=on オプションおよび **index=on** オプションを使用して、POSIX コンプライアンスを向上させることもできます。この 2 つのオプションにより、上位レイヤーの形式は、このオプションなしでオーバーレイと互換性がありません。つまり、**redirect_dir=on**

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

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

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

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

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

(BZ#1690207)

Straits がテクノロジープレビューとして利用可能になりました。

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** ユーティリティーを使用します。

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 8.3 は Stratis をバージョン 2.1.0 に更新した。詳細は、[Stratis 2.1.0 リリースノート](#) を参照してください。

(JIRA:RHELPLAN-1212)

テクノロジープレビューとして、IdM ドメインメンバーで Samba サーバーを設定できるようになりました。

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

IdM 信頼コントローラーが Global Catalog Service をサポートしないため、AD が登録した Windows ホストは Windows で IdM ユーザーおよびグループを見つけることができません。さらに、IdM 信頼コントローラーは、Distributed Computing Environment / Remote Procedure Calls (DCE/RPC) プロトコル

を使用する IdM グループの解決をサポートしません。これにより、AD ユーザーは、IdM クライアントから Samba の共有およびプリンターにしかアクセスできません。

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

(JIRA:RHELPLAN-13195)

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)

8.5. 高可用性およびクラスター

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

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

(BZ#1619620)

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

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

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)

リソース移動後の場所の制約の自動削除がテクノロジープレビューとして利用可能

pcs resource move コマンドを実行すると、現在実行しているノードでリソースが実行されないように、制約がリソースに追加されます。**pcs resource move** コマンドの **--autodelete** オプションが、テクノロジープレビューとして利用可能になりました。このオプションを指定すると、リソースを移動すると、コマンドが作成する場所の制約が自動的に削除されます。

(BZ#1847102)

8.6. IDENTITY MANAGEMENT

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。

以前では、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。これらの機能拡張により、互換性のない方法でコマンドの動作が変更することがありました。IdM API を変更しても、既存のツールおよびスクリプトを引き続き使用できるようになりました。これにより、以下が可能になります。

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

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

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

(BZ#1664719)

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

統合 DNS のある Identity Management (IdM) サーバーは、DNS プロトコルのセキュリティを強化する 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 ゾーンを保護する場合は、以下のドキュメントを参照することが推奨されます。

- [DNSSEC Operational Practices, Version 2](#)
- [Secure Domain Name System \(DNS\) Deployment Guide](#)

- [DNSSEC Key Rollover Timing Considerations](#)

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)

ACME はテクノロジープレビューとしてご利用いただけます。

Automated Certificate Management Environment (ACME) サービスが、テクノロジープレビューとして Identity Management (IdM) で利用可能になりました。ACME は、自動化識別子の検証および証明書の発行に使用するプロトコルです。この目的は、証明書の有効期間を短縮し、証明書のライフサイクル管理での手動プロセスを回避することにより、セキュリティーを向上させることです。

RHEL では、ACME サービスは Red Hat Certificate System (RHCS) PKI ACME レスポンダーを使用します。RHCS ACME サブシステムは、IdM デプロイメントのすべての認証局 (CA) サーバーに自動的にデプロイされますが、管理者が有効にするまでリクエストに対応しません。RHCS は、ACME 証明書を発行する際に **acmeIPAServerCert** プロファイルを使用します。発行された証明書の有効期間は 90 日です。ACME サービスの有効化または無効化は、IdM デプロイメント全体に影響します。



重要

ACME は、すべてのサーバーが RHEL 8.4 以降を実行している IdM デプロイメントでのみ有効にすることが推奨されます。以前の RHEL バージョンには ACME サービスが含まれていないため、バージョンが混在するデプロイメントで問題が発生する可能性があります。たとえば、ACME のない CA サーバーは、異なる DNS サブジェクト代替名 (SAN) を使用しているため、クライアント接続が失敗する可能性があります。



警告

現在、RHCS は期限切れの証明書を削除しません。ACME 証明書は 90 日後に期限切れになるため、期限切れの証明書が蓄積され、パフォーマンスに影響を及ぼす可能性があります。

- IdM デプロイメント全体で ACME を有効にするには、**ipa-acme-manage enable** コマンドを使用します。

```
# ipa-acme-manage enable
The ipa-acme-manage command was successful
```

- IdM デプロイメント全体で ACME を無効にするには、**ipa-acme-manage disable** コマンドを使用します。

```
# ipa-acme-manage disable
The ipa-acme-manage command was successful
```

- ACME サービスがインストールされ、有効または無効であるかを確認するには、**ipa-acme-manage status** コマンドを使用します。

```
# ipa-acme-manage status
ACME is enabled
The ipa-acme-manage command was successful
```

(JIRA:RHELPLAN-58596)

8.7. 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うになりました。

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#1667225, BZ#1667516, [BZ#1724302](#))

IBM Z の GNOME デスクトップがテクノロジープレビューとして利用可能に

Firefox Web ブラウザーを含む GNOME デスクトップが、IBM Z アーキテクチャーでテクノロジープレビューとして利用できるようになりました。VNC を使用して GNOME を実行するリモートグラフィカルセッションに接続し、IBM Z サーバーを設定および管理できるようになりました。

(JIRA:RHELPLAN-27737)

8.8. 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)

8.9. RED HAT ENTERPRISE LINUX システムロール

HA クラスターの RHEL システムロールがテクノロジープレビューとして利用可能に

高可用性クラスター (HA クラスター) 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。現在、以下の重要な設定を利用することができます。

- ノード、フェンスデバイス、リソース、リソースグループ、およびリソースクローン (メタ属性およびリソース操作を含む) の設定
- クラスタープロパティーの設定
- マルチリンククラスターの設定

- カスタムクラスター名およびノード名の設定
- システムの起動時にクラスターが自動的に起動するかどうかの設定
- 基本的な corosync クラスターと pacemaker クラスターのプロパティー、stonith、リソースの設定。

現在、**ha_cluster** システムロールは制約をサポートしていません。制約を設定してからロールを実行すると、制約や、ロールでサポートされていない設定も削除されます。

現在、**ha_cluster** システムロールは SBD をサポートしていません。

(BZ#1893743、BZ#1978726)

8.10. 仮想化

KVM 仮想マシンの AMD SEV および SEV-ES

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、SEV が仮想マシンのメモリーを暗号化して、ホストから仮想マシンへのアクセスを防ぎます。これにより、仮想マシンのセキュリティーが向上します。

さらに、強化された SEV (Encrypted State) バージョンの SEV (SEV-ES) もテクノロジープレビューとして提供されます。SEV-ES は、仮想マシンの実行が停止すると、すべての CPU レジスターの内容を暗号化します。これにより、ホストが仮想マシンの CPU レジスターを変更したり、そこから情報を読み取ったりできなくなります。

SEV および SEV-ES は、第 2 世代の AMD EPYC CPU (コードネーム Rome) 以降でのみ機能することに注意してください。また、RHEL 8 には SEV および SEV-ES の暗号化が含まれますが、SEV および SEV-ES のセキュリティー証明は含まれません。

(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することができます。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。

さらに、Intel vGPU が操作する VNC コンソールを有効にすることもできます。これを有効にすると、ユーザーは仮想マシンの VNC コンソールに接続し、Intel vGPU がホストする仮想マシンのデスクトップを確認できます。ただし、これは現在 RHEL ゲストオペレーティングシステムでのみ動作します。

(BZ#1528684)

入れ子仮想マシンの作成

入れ子 KVM 仮想化は、RHEL 8 で Intel、AMD64、および IBM Z システムホストで実行している KVM 仮想マシン用のテクノロジープレビューとして提供されます。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。

(JIRA:RHELPLAN-14047, JIRA:RHELPLAN-24437)

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、**ixgbbevf** および **ixgbevf** ドライバーがサポートする Intel ネットワークアダプターに、シングルルート I/O 仮想化 (SR-IOV) 機能を使用することができるようになりました。この機能は、以下の条件が満たされると有効になります。

- ネットワークインターフェイスコントローラー (NIC) に対して SR-IOV サポートが有効になっている
- 仮想 NIC の SR-IOV サポートが有効になっている
- 仮想スイッチの SR-IOV サポートが有効になっている
- NIC からの VF (Virtual Function) が仮想マシンに割り当てられている

この機能は現在、Microsoft Windows Server 2019 および 2016 で対応しています。

(BZ#1348508)

RHEL 仮想マシンで、ESX ハイパーバイザーおよび SEV-ES 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。

テクノロジープレビューとして、RHEL 8.4 以降では、AMD Secure Encrypted Virtualization-Encrypted State (SEV-ES) を有効にして、VMware の ESXi ハイパーバイザー (バージョン 7.0.2 以降) で RHEL 仮想マシンのセキュリティーを確保できます。

(BZ#1904496)

virtiofs を使用したホストと仮想マシン間でのファイルの共有

RHEL 8 では、テクノロジープレビューとして virtio ファイルシステム (**virtiofs**) が追加されました。 **virtiofs** を使用すると、ホストシステムと仮想マシン (VM) との間で、ファイルを効率的に共有できます。

(BZ#1741615)

RHEL 8 Hyper-V 仮想マシンで KVM 仮想化が利用可能に

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 および AMD 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)

8.11. コンテナー

Toolbox がテクノロジープレビューとして利用可能になりました。

これまでの Toolbox ユーティリティーは、RHEL CoreOS の github.com/coreos/toolbox をベースにしていました。今回のリリースでは、Toolbox は github.com/containers/toolbox に置き換えられました。

(JIRA:RHELPLAN-77238)

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:RHELDOCS-16861)

第9章 非推奨の機能

ここでは、Red Hat Enterprise Linux 8 で **非推奨** となった機能の概要を説明します。

非推奨のデバイスは完全にサポートされています。つまり、非推奨のデバイスはテストおよび保守されています。デバイスが非推奨になっても、Red Hat Enterprise Linux 8 内でのサポート状況は変わりません。ただし、非推奨のデバイスは、次のメジャーバージョンのリリースではサポートされない可能性が高く、最新または今後のメジャーバージョンの新規 RHEL デプロイメントには推奨されません。

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。サポート期間の詳細は、[Red Hat Enterprise Linux のライフサイクル](#) および [Red Hat Enterprise Linux アプリケーションストリームのライフサイクル](#) を参照してください。

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。特定の状況下では、製品からパッケージが削除されることがあります。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。

RHEL 8 で使用され、RHEL 9 で削除された機能の詳細は [RHEL 9 の導入における検討事項](#) を参照してください。

9.1. 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)

キックスタートの `autostep` コマンドが非推奨に

`autostep` コマンドが非推奨になりました。このコマンドに関するセクションは、[RHEL 8 のドキュメント](#) から削除されました。

(BZ#1904251)

RHEL 8 で非推奨となった Image Builder `lorax-composer` バックエンド

Image Builder の `lorax-composer` バックエンドは、非推奨になりました。Red Hat Enterprise Linux 8 の残りのライフサイクルでは一部の修正のみが行われ、今後のメジャーリリースから削除される予定です。Red Hat では、`lorax-composer` をアンインストールして `osbuild-composer` バックエンドを代わりにインストールすることを推奨します。

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

(BZ#1893767)

9.2. 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` コマンドが非推奨になりました。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)

9.3. シェルおよびコマンドラインツール

`OpenEXR` コンポーネントが非推奨になりました。

`OpenEXR` コンポーネントが非推奨になりました。そのため、`EXR` イメージ形式のサポートは `imagecodecs` モジュールから削除されました。

(BZ#1886310)

`dump` からの `dump` ユーティリティーが非推奨になりました。

ファイルシステムのバックアップに使用される **dump** ユーティリティーが非推奨になり、RHEL 9 では使用できなくなります。

RHEL 9 では、使用方法に基づいて、**bacula**、**tar**、または **dd** のバックアップユーティリティーを使用することが推奨されています。これにより、ext2、ext3、および ext4 のファイルシステムで完全で安全なバックアップが提供されます。

dump パッケージの **restore** ユーティリティーは、RHEL 9 で引き続き利用可能で、サポートされており、**restore** パッケージとして利用できます。

(BZ#1997366)

hidepid=n マウントオプションが、RHEL 8 **systemd** で未サポート

マウントオプションの **hidepid=n** は、**/proc/[pid]** ディレクトリーの情報にアクセスできるユーザーを制御しますが、RHEL 8 で提供されている **systemd** インフラストラクチャーと互換性がありません。

また、このオプションを使用すると、**systemd** が起動する特定のサービスで SELinux の AVC 拒否メッセージが生成され、その他の操作が完了しないようにする場合があります。

詳細は、関連 [RHEL7 および RHEL8](#) では、**/proc** を **hidepid=2** でマウントすることが推奨されますか？を参照してください。

(BZ#2038929)

9.4. SECURITY

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。NSS がサポートを削除した際に SEED 暗号に依存するデプロイメントを円滑に移行させるため、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。

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の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ー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)

crypto-policies から派生したプロパティーが非推奨に

カスタムポリシーにおける **crypto-policies** ディレクトリのスコープの導入により、**tls_cipher**、**ssh_cipher**、**ssh_group**、**ike_protocol**、および **sha1_in_dnssec** の派生プロパティーが非推奨になりました。さらに、スコープを指定しない **protocol** プロパティーの使用も非推奨になりました。推奨される代替は、**crypto-policies(7)** の man ページを参照してください。

(BZ#2011208)

/etc/selinux/config を使用して SELinux を無効にするランタイムが非推奨になりました。

/etc/selinux/config ファイルの **SELINUX=disabled** オプションを使用して SELinux を無効にするランタイムが非推奨になりました。RHEL 9 では、/etc/selinux/config でのみ SELinux を無効にすると、システムは SELinux が有効化されますが、ポリシーが読み込まれずには開始します。

SELinux を完全に無効にする必要がある場合には、Red Hat は、**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 を無効にすることを推奨します。これは、[SELinux の使用](#) タイトルの [システムの起動時に SELinux モードの変更](#) セクションで説明されています。

(BZ#1932222)

selinux-policy から ipa SELinux モジュールが削除されました。

ipa SELinux はメンテナンスされなくなったため、**selinux-policy** から削除されました。この機能は、**ipa-selinux** サブパッケージに含まれるようになりました。

ローカルの SELinux ポリシーで、**ipa** モジュールからタイプやインターフェイスを使用する必要がある場合は、**ipa-selinux** をインストールします。

(BZ#1461914)

9.5. ネットワーキング

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)

dropwatch ツールが非推奨に

`dropwatch` ツールが非推奨になりました。このツールは今後のリリースではサポートされませんので、新規デプロイメントには推奨できません。このパッケージの代わりに、Red Hat は `perf` コマンドラインツールを使用することを推奨します。

`perf` コマンドラインツールの使用方法の詳細は、Red Hat カスタマーポータルの [Getting started with Perf](#) セクションまたは `perf` の `man` ページを参照してください。

(BZ#1929173)

cgdcbxd パッケージが非推奨に

コントロールグループデータセンターブリッジング交換デーモン (`cgdcbxd`) は、データセンターのブリッジ (DCB) のネットリンクイベントをモニターし `net_prio control` グループサブシステムを管理するサービスです。RHEL 8.5 以降では、`cgdcbxd` パッケージは非推奨となり、次の RHEL メジャーリリースで削除されます。

(BZ#2006665)

xinetd が非推奨に

`xinetd` サービスが非推奨になり、RHEL 9 では削除される予定です。代わりに `systemd` を使用します。詳細は、[xinetd サービスを systemd に変換する方法](#) を参照してください。

(BZ#2009113)

slaves という用語は、`nmstate` API では非推奨です。

Red Hat では、意識的な言語の使用に取り組んでいます。したがって、**slaves** という用語は `Nmstate` API では非推奨です。`nmstatectl` を使用する場合は、**port** という用語を使用します。

(JIRA:RHELDODCS-18764)

9.6. カーネル

カーネルライブパッチが、すべての RHEL マイナーリリースに対応するようになりました。

RHEL 8.1 以降、カーネルライブパッチは、影響度が重大および重要な Common Vulnerabilities and Exposures (CVE) を修正するために、Extended Update Support (EUS) ポリシーの対象となる RHEL の一部のマイナーリリースストリームに提供されています。同時にカバーされるカーネルとユースケースの最大数に対応するため、各ライブパッチのサポート期間は、カーネルのマイナー、メジャー、および zStream の各バージョンで 12 カ月から 6 カ月に減少します。これは、カーネルライブパッチがリリースされると、過去 6 カ月間に配信されたすべてのマイナーリリースとスケジュール済みのエラータカーネルが含まれます。たとえば、8.4.x には 1 年間のサポートウィンドウがありますが、8.4.x+1 には 6 カ月があります。

この機能の詳細は、[Applying patches with kernel live patching](#) を参照してください。

利用可能なカーネルライブパッチの詳細は、[Kernel Live Patch life cycles](#) を参照してください。

(BZ#1958250)

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)

Linux `firewire` サブシステムおよび関連するユーザー空間コンポーネントは、RHEL 8 では非推奨になりました。

`firewire` サブシステムは、IEEE 1394 バスでリソースを使用し、維持するインターフェイスを提供します。RHEL 9 では、`firewire` は、`kernel` パッケージで対応しなくなります。`firewire` には、`libavc1394`、`libdc1394`、`libraw1394` パッケージで提供されるユーザー空間コンポーネントが複数含まれることに注意してください。これらのパッケージも非推奨になります。

(BZ#1871863)

`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)

9.7. ファイルシステムおよびストレージ

`async` 以外の VDO 書き込みモードが非推奨に

VDO は、RHEL 8 で複数の書き込みモードに対応します。

- **sync**
- **async**
- **async-unsafe**
- **auto**

RHEL 8.4 以降、以下の書き込みモードが非推奨になりました。

sync

VDO レイヤー上のデバイスは、VDO が同期されているかどうかを認識できないため、デバイスは VDO **sync** モードを利用できません。

async-unsafe

VDO は、ACID (Atomicity, Consistency, Isolation, and Durability) に準拠する **async** モードの回避策としてこの書き込みモードを追加しました。Red Hat は、ほとんどのユースケースで **async-unsafe** を推奨せず、それに依存するユーザーを認識しません。

auto

この書き込みモードは、他の書き込みモードのいずれかのみを選択します。VDO が1つの書き込みモードのみに対応している場合は、不要になりました。

この書き込みモードは、今後の RHEL メジャーリリースで削除されます。

推奨される VDO 書き込みモードが **async** になりました。

VDO 書き込みモードの詳細は、[VDO 書き込みモードの選択](#) を参照してください。

(JIRA:RHELPLAN-70700)

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)

cramfs が非推奨になりました。

ユーザーの不足により、**cramfs** カーネルモジュールが非推奨になりました。代替策として **squashfs** が推奨されます。

(BZ#1794513)

VDO マネージャー が非推奨に

python ベースの VDO 管理ソフトウェアは非推奨となり、RHEL 9 から削除される予定です。RHEL 9 では、LVM-VDO 統合に置き換えられます。そのため、**lvcreate** コマンドを使用して VDO ボリュームを作成することが推奨されます。

VDO 管理ソフトウェアを使用して作成した既存のボリュームは、**lvm2** パッケージが提供する **/usr/sbin/lvm_import_vdo** スクリプトを使用して変換できます。LVM-VDO 実装の詳細は、[LVM 上の VDO の概要](#) を参照してください。

(BZ#1949163)

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 RAID1 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)

peripety が非推奨に

peripety パッケージは、RHEL 8.3 以降で非推奨になりました。

Peripety ストレージイベント通知デーモンは、システムストレージログを構造化されたストレージイベントに解析します。ストレージの問題を調査するのに役立ちます。

(BZ#1871953)

9.8. 高可用性およびクラスター

clutter ツールに対応する **pcs** コマンドが非推奨になりました。

クラスター設定フォーマットを分析する **clutter** ツールに対応する **pcs** コマンドが非推奨になりました。これらのコマンドにより、コマンドが非推奨になり、コマンドに関連するセクションが **pcs** ヘルプ表示と、**pcs(8)** man ページから削除されていることを示す警告が出力されるようになりました。

以下のコマンドが非推奨になりました。

- **pcs config import-cman**: CMAN / RHEL6 HA クラスター設定のインポート
- **pcs config export**: クラスター設定を、同じクラスターを再作成する **pcs** コマンドのリストにエクスポート

(BZ#1851335)

9.9. コンパイラーおよび開発ツール

libdwarf が非推奨に

RHEL 8 では、**libdwarf** ライブラリーが非推奨になりました。ライブラリーは、将来のメジャーリリースでサポートされない可能性があります。代わりに、ELF/DWARF ファイルを処理するアプリケーションに **elfutils** および **libdw** ライブラリーを使用してください。

libdwarf-tools dwarfdump プログラムの代替は、**binutils readelf** プログラムまたは **elfutils eu-readelf** プログラムになります。どちらも **--debug-dump** フラグを渡すことで使用されます。

(BZ#1920624)

gdb.i686 パッケージが非推奨に

RHEL 8.1 では、別のパッケージの依存関係の問題が原因で、32 ビットバージョンの GNU Debugger(GDB) **gdb.i686** が同梱されていました。RHEL 8 は 32 ビットハードウェアに対応していないため、RHEL 8.4 以降、**gdb.i686** パッケージは非推奨になりました。64 ビットバージョンの GDB (**gdb.x86_64**) は、32 ビットアプリケーションをデバッグできます。

gdb.i686 を使用する場合は、以下の重要な問題に注意してください。

- **gdb.i686** パッケージは更新されなくなりました。代わりに **gdb.x86_64** をインストールする必要があります。
- **gdb.i686** をインストールしている場合は、**gdb.x86_64** をインストールすると、**dnf** が **package gdb-8.2-14.el8.x86_64 obsoletes gdb < 8.2-14.el8 provided by gdb-8.2-12.el8.i686** を報告します。これは想定される状況です。**gdb.i686** をアンインストールするか、**--allowrasing** オプションを **dnf** に渡して **gdb.i686** を削除し、**gdb.x8_64** をインストールします。
- ユーザーは、64 ビットシステム(つまり、**libc.so.6()(64-bit)** パッケージのある)に **gdb.i686** パッケージをインストールすることができなくなります。

(BZ#1853140)

9.10. IDENTITY MANAGEMENT

openssh-ldap が非推奨に

openssh-ldap サブパッケージは、Red Hat Enterprise Linux 8 で非推奨になり、RHEL 9 で削除されます。**openssh-ldap** サブパッケージはアップストリームでは維持されないため、Red Hat は SSSD と

sss_sshAuthorizedKeys ヘルパーを使用することを推奨しています。これは、他の IdM ソリューションよりも適切に統合でき、安全です。

デフォルトでは、**ldap** および **ipa** プロバイダーはユーザー オブジェクトの **sshPublicKey** LDAP 属性を読み取ります (利用可能な場合)。AD (Active Directory) には公開鍵を保存するためのデフォルトの LDAP 属性がないため、**ad** プロバイダーまたは IdM の信頼されるドメインのデフォルト SSSD 設定を使用して AD から SSH 公開鍵を取得することはできません。

sss_sshAuthorizedKeys ヘルパーが SSSD から鍵を取得できるようにするには、**sssd.conf** ファイルの **services** オプションに **ssh** を追加して **ssh** レスポンダーを有効にします。詳細は **man** ページの **sssd.conf(5)** を参照してください。

sshd が **sss_sshAuthorizedKeys** を使用できるようにするには、**man** ページの **sss_sshAuthorizedKeys(1)** に記載されているように、**AuthorizedKeysCommand** **/usr/bin/sss_sshAuthorizedKeys** および **AuthorizedKeysCommandUser nobody** オプションを **/etc/ssh/sshd_config** ファイルに追加します。

(BZ#1871025)

DES および 3DES 暗号化タイプが削除されました。

RHEL 7 以降、セキュリティ上の理由から、データ暗号化標準 (DES) アルゴリズムが非推奨になり、デフォルトで無効化になりました。Kerberos パッケージの最近のリベースで、RHEL 8 からシングル DES (DES) およびトリプル DES (3DES) の暗号化タイプが削除されました。

DES または 3DES の暗号化のみを使用するようにサービスまたはユーザーが設定されている場合、以下のようないサービスの中断が発生する可能性があります。

- Kerberos 認証エラー
- **unknown enctype** 暗号化工エラー
- DES で暗号化されたデータベースマスター キー (K/M) を使用した KDC (Kerberos Distribution Center) が起動しない

アップグレードを準備するには、以下の操作を実施します。

1. KDC が **krb5check** オープンソース Python スクリプトで DES または 3DES 暗号化を使用しているかどうかを確認します。GitHub の [krb5check](#) を参照してください。
2. Kerberos プリンシパルで DES または 3DES 暗号化を使用している場合は、Advanced Encryption Standard (AES) などのサポート対象の暗号化タイプでキーを変更します。キーの再生成の手順については、MIT Kerberos ドキュメントの [Retiring DES](#) を参照してください。
3. アップグレードの前に以下の Kerberos オプションを一時的に設定して、DES および 3DES からの独立性をテストします。
 - a. KDC の **/var/kerberos/krb5kdc/kdc.conf** で、**supported_enctypes** を設定し、**des** または **des3** は含まれません。
 - b. すべてのホストについて、**/etc/krb5.conf** および **/etc/krb5.conf.d** のすべてのファイルで、**allow_weak_crypto** を **false** に設定します。デフォルトは **false** です。
 - c. すべてのホストについて、**/etc/krb5.conf** および **/etc/krb5.conf.d** のすべてのファイルで、**permitted_enctypes**、**default_tgs_enctypes**、**default_tkt_enctypes** を設定します。また、**des** または **des3** は含まれません。

- 前の手順で Kerberos 設定をテストしてサービスが中断されない場合は、サービスを削除してアップグレードします。最新の Kerberos パッケージにアップグレードした後は、この設定は必要ありません。

(BZ#1877991)

ctdb サービスのスタンドアロン使用が非推奨になりました。

RHEL 8.4 以降、以下の条件がすべて適用されている場合に限り、**ctdb** クラスター Samba サービスを使用することが推奨されます。

- **ctdb** サービスは、resource-agent **ctdb** を使用して **pacemaker** リソースとして管理されます。
- **ctdb** サービスは、Red Hat Gluster Storage 製品または GFS2 ファイルシステムが提供する GlusterFS ファイルシステムのいずれかが含まれるストレージボリュームを使用します。

ctdb サービスのスタンドアロンユースケースは非推奨となり、Red Hat Enterprise Linux の次期メジャーリリースには含まれません。Samba のサポートポリシーの詳細は、ナレッジベースの記事 [Support Policies for RHEL Resilient Storage - ctdb General Policies](#) を参照してください。

(BZ#1916296)

WinSync による IdM との間接的な AD 統合が非推奨に

WinSync は、さまざまな機能制限のため、RHEL 8 では積極的に開発されなくなりました。

- WinSync は、1つの Active Directory (AD) ドメインのみをサポートします。
- パスワードの同期には、AD ドメインコントローラに追加のソフトウェアをインストールする必要があります。

リソースとセキュリティの分離を強化したより強固なソリューションとして、レッドハットは Active Directory との間接的な統合にフォレスト間の信頼を使用することを推奨しています。 [間接的な統合](#) のドキュメントを参照してください。

(JIRA:RHELPLAN-100400)

Samba を PDC または BDC として実行することは非推奨になりました。

管理者が Samba を NT4 のようなプライマリードメインコントローラ (PDC) として実行し、バックアップドメインコントローラ (BDC) を実行できるようにする従来のドメインコントローラモードが非推奨になりました。これらのモードを設定するためのコードおよび設定は、今後の Samba リリースで削除されます。

RHEL 8 の Samba バージョンが PDC モードおよび BDC モードを提供している限り、Red Hat は、NT4 ドメインに対応する Windows バージョンを使用する既存のインストールでのみ、これらのモードをサポートします。Red Hat は、新規の Samba NT4 ドメインのセットアップを推奨しません。なぜなら、Microsoft のオペレーティングシステム (Windows 7 以降) および Windows Server 2008 R2 は、NT4 ドメインをサポートしないからです。

PDC を使用して Linux ユーザーのみを認証する場合、Red Hat は、RHEL サブスクリプションに含まれる [Red Hat Identity Management \(IdM\)](#) への移行を推奨します。ただし、Windows システムを IdM ドメインに参加させることはできません。Red Hat は、引き続き IdM が使用する PDC 機能のサポートを継続することに注意してください。

Red Hat は、Samba を AD ドメインコントローラ (DC) として実行することはサポートしていません。

(BZ#1926114)

SSSD バージョンの libwbclient が削除される

libwbclient パッケージの SSSD 実装は、RHEL 8.4 で非推奨になりました。最新バージョンの Samba で使用できないため、**libwbclient** の SSSD 実装が削除されています。

(BZ#1947671)

Samba で SMB1 プロトコルが非推奨になりました

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:RHELODCS-16612

9.11. 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)

9.12. 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)

Motif が非推奨になりました。

Motif ウィジェットツールキットが非推奨になりました。アップストリームの Motif コミュニティーの開発は非アクティブです。

以下の Motif パッケージ (開発バリアントおよびデバッグバリアントを含む) は非推奨となりました。

- **motif**
- **motif-static**
- **openmotif**
- **openmotif21**
- **openmotif22**

Red Hat は、GTK ツールキットを代替として使用することを推奨します。GTK は Motif と比較してメンテナンス性が高く、新機能を提供します。

(JIRA:RHELPLAN-98983)

9.13. 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。ブラウザー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)

9.14. RED HAT ENTERPRISE LINUX システムロール

geoipupdate パッケージが非推奨に

geoipupdate パッケージにはサードパーティのサブスクリプションが必要で、プロプライエタリーコンテンツもダウンロードします。したがって、**geoipupdate** パッケージは非推奨となり、次の RHEL メジャーバージョンで削除されます。

(BZ#1874892)

9.15. 仮想化

SPICE が非推奨になりました

SPICE リモートディスプレイプロトコルが非推奨になりました。その結果、SPICE は RHEL 8 で引き続きサポートされますが、Red Hat では、リモートディスプレイストリーミングに代替ソリューションを使用することを推奨しています。

- リモートコンソールへのアクセスには、VNC プロトコルを使用します。
- 高度なリモートディスプレイ機能には、RDP、HP RGS、または Mechdyne TGX などのサードパーティーツールを使用します。

SPICE で使用される **QXL** グラフィックスデバイスも非推奨になっていることに注意してください。

(BZ#1849563)

virsh iface-* コマンドが非推奨になりました。

virsh iface-start、**virsh iface-destroy** などの **virsh iface-*** コマンドは非推奨になり、将来のメジャーバージョンの RHEL では削除される予定です。また、このようなコマンドは設定の依存関係により頻繁に失敗します。

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、関連する管理アプリケーション (**nmcli** など) を使用します。

(BZ#1664592)

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 Web コンソール (`Cockpit`) は、後続のリリースで置き換える予定です。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 Web コンソールで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。

(BZ#1686057)

Cirrus VGA 仮想 GPU タイプが非推奨に

Red Hat Enterprise Linux の今後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` または `virtio-vga` デバイスの使用を推奨します。

(BZ#1651994)

IBM POWER 上の KVM が非推奨に

IBM POWER ハードウェアでの KVM 仮想化の使用は非推奨になりました。その結果、IBM POWER の KVM は、RHEL 8 でも引き続きサポートされますが、RHEL の今後のメジャーリリースではサポートされなくなります。

(JIRA:RHELPLAN-71200)

SHA1 ベースの署名を使用した SecureBoot イメージ検証が非推奨に

UEFI (PE/COFF) 実行ファイルでの SHA1 ベースの署名を使用した SecureBoot イメージ検証の実行は非推奨になりました。代わりに、Red Hat は、SHA2 アルゴリズムまたはそれ以降に基づく署名を使用することを推奨します。

(BZ#1935497)

9.16. サポート性

-s スプリットオプションは、-f オプションではサポートされない

Red Hat Secure FTP にファイルをアップロードして Red Hat Support に提供する際に、`redhat-support-tool addattachment -f` コマンドを実行することができます。しかし、インフラストラクチャーの変更により、大きなファイルを分割して Red Hat Secure FTP にアップロードするために、このコマンドで `-s` オプションを使用することはできなくなりました。

(BZ#2013335)

`redhat-support-tool diagnose <file_or_directory>` コマンドが非推奨に

Red Hat サポートツール は、これまでファイルやディレクトリーの高度な診断サービスに使用されていた `redhat-support-tool diagnose <file_or_directory>` コマンドをサポートしなくなりました。`redhat-support-tool diagnose` コマンドは、引き続きプレーンテキスト解析をサポートします。

(BZ#2019786)

9.17. コンテナー

Podman varlink ベースの API v1.0 が削除されました

Podman varlink ベースの API v1.0 は、以前のリリースの RHEL 8 で非推奨となりました。Podman v2.0 には、新しい Podman v2.0 RESTful API が導入されました。Podman v3.0 のリリースでは、varlink ベースの API v1.0 が完全に削除されました。

(JIRA:RHELPLAN-45858)

container-tools:1.0 が非推奨に

container-tools:1.0 モジュールは非推奨となり、セキュリティ更新を受信しなくなります。**container-tools:2.0** や **container-tools:3.0** などの新しいサポートされる安定したモジュールストリームを使用することが推奨されます。

(JIRA:RHELPLAN-59825)

9.18. 非推奨のパッケージ

このセクションでは、非推奨となり、将来バージョンの Red Hat Enterprise Linux には含まれない可能性があるパッケージのリストを示します。

RHEL 7 と RHEL 8 との間でパッケージを変更する場合は、[RHEL 8 の導入における考慮事項](#) ドキュメントの [パッケージの変更](#) を参照してください。

以下のパッケージは非推奨となり、RHEL 8 のライフサイクルが終了するまでサポートされ続けます。

- 389-ds-base-legacy-tools
- adobe-source-sans-pro-fonts
- adwaita-qt
- amanda
- amanda-client
- amanda-libs
- amanda-server
- ant-contrib
- antlr3
- antlr32
- aopalliance
- apache-commons-collections
- apache-commons-compress
- apache-commons-exec
- apache-commons-jxpath

- apache-commons-parent
- apache-ivy
- apache-parent
- apache-resource-bundles
- apache-sshd
- apiguardian
- assertj-core
- authd
- auto
- autoconf213
- autogen
- base64coder
- batik
- bea-stax
- bea-stax-api
- bind-sdb
- bouncycastle
- bsh
- buildnumber-maven-plugin
- byaccj
- cal10n
- cbi-plugins
- cdparanoia
- cdparanoia-devel
- cdparanoia-libs
- cdrdao
- cmirror
- codehaus-parent
- codemodel

- compat-exiv2-026
- compat-guile18
- compat-libpthread-nonshared
- compat-openssl10
- compat-sap-c++-10
- createrepo_c-devel
- ctags
- ctags-ctags
- custodia
- dbus-c++
- dbus-c++-devel
- dbus-c++-glib
- dbxtool
- dirsplt
- dleyna-connector-dbus
- dleyna-core
- dleyna-renderer
- dleyna-server
- dnssec-trigger
- dptfxtract
- drpm
- drpm-devel
- dvd+rw-tools
- dyninst-static
- eclipse-ecf
- eclipse-emf
- eclipse-license
- ed25519-java
- ee4j-parent

- elfutils-devel-static
- elfutils-libelf-devel-static
- enca
- enca-devel
- environment-modules-compat
- evince-browser-plugin
- exec-maven-plugin
- farstream02
- felix-osgi-compendium
- felix-osgi-core
- felix-osgi-foundation
- felix-parent
- file-roller
- fipscheck
- fipscheck-devel
- fipscheck-lib
- firewire
- forge-parent
- fuse-sshfs
- fusesource-pom
- future
- gamin
- gamin-devel
- gavl
- gcc-toolset-10
- gcc-toolset-10-annobin
- gcc-toolset-10-binutils
- gcc-toolset-10-binutils-devel
- gcc-toolset-10-build

- gcc-toolset-10-dwz
- gcc-toolset-10-dyninst
- gcc-toolset-10-dyninst-devel
- gcc-toolset-10-elfutils
- gcc-toolset-10-elfutils-debuginfod-client
- gcc-toolset-10-elfutils-debuginfod-client-devel
- gcc-toolset-10-elfutils-devel
- gcc-toolset-10-elfutils-libelf
- gcc-toolset-10-elfutils-libelf-devel
- gcc-toolset-10-elfutils-libs
- gcc-toolset-10-gcc
- gcc-toolset-10-gcc-c++
- gcc-toolset-10-gcc-gdb-plugin
- gcc-toolset-10-gcc-gfortran
- gcc-toolset-10-gdb
- gcc-toolset-10-gdb-doc
- gcc-toolset-10-gdb-gdbserver
- gcc-toolset-10-libasan-devel
- gcc-toolset-10-libatomic-devel
- gcc-toolset-10-libitm-devel
- gcc-toolset-10-liblsan-devel
- gcc-toolset-10-libquadmath-devel
- gcc-toolset-10-libstdc++-devel
- gcc-toolset-10-libstdc++-docs
- gcc-toolset-10-libtsan-devel
- gcc-toolset-10-libubsan-devel
- gcc-toolset-10-ltrace
- gcc-toolset-10-make
- gcc-toolset-10-make-devel

- gcc-toolset-10-perf-tools
- gcc-toolset-10-runtime
- gcc-toolset-10-strace
- gcc-toolset-10-systemtap
- gcc-toolset-10-systemtap-client
- gcc-toolset-10-systemtap-devel
- gcc-toolset-10-systemtap-initscript
- gcc-toolset-10-systemtap-runtime
- gcc-toolset-10-systemtap-sdt-devel
- gcc-toolset-10-systemtap-server
- gcc-toolset-10-toolchain
- gcc-toolset-10-valgrind
- gcc-toolset-10-valgrind-devel
- gcc-toolset-9
- gcc-toolset-9-annobin
- gcc-toolset-9-build
- gcc-toolset-9-perf-tools
- gcc-toolset-9-runtime
- gcc-toolset-9-toolchain
- GConf2
- GConf2-devel
- genisoimage
- genwqe-tools
- genwqe-vpd
- genwqe-zlib
- genwqe-zlib-devel
- geoipupdate
- geronimo-annotation
- geronimo-jms

- geronimo-jpa
- geronimo-parent-poms
- gfbgraph
- gflags
- gflags-devel
- glassfish-annotation-api
- glassfish-el
- glassfish-fastinfoset
- glassfish-jaxb-core
- glassfish-jaxb-txw2
- glassfish-jsp
- glassfish-jsp-api
- glassfish-legal
- glassfish-master-pom
- glassfish-servlet-api
- glew-devel
- glib2-fam
- glog
- glog-devel
- gmock
- gmock-devel
- gnome-boxes
- gnome-menus-devel
- gnome-online-miners
- gnome-shell-extension-disable-screenshield
- gnome-shell-extension-horizontal-workspaces
- gnome-shell-extension-no-hot-corner
- gnome-shell-extension-window-grouper
- gnome-themes-standard

- gnupg2-smime
- gobject-introspection-devel
- google-gson
- gphoto2
- gssntlmssp
- gtest
- gtest-devel
- gtkmm24
- gtkmm24-devel
- gtkmm24-docs
- gtksourceview3
- gtksourceview3-devel
- gtkspell
- gtkspell-devel
- gtkspell3
- guile
- gutenprint-gimp
- gvfs-afc
- gvfs-afp
- gvfs-archive
- hawtjni
- highlight-gui
- hivex-devel
- hostname
- hplip-gui
- httpcomponents-project
- icedax
- icu4j
- idm-console-framework

- iptables
- ipython
- isl
- isl-devel
- isorelax
- istack-commons-runtime
- istack-commons-tools
- iwl3945-firmware
- iwl4965-firmware
- iwl6000-firmware
- jacoco
- jaf
- jakarta-oro
- janino
- jansi-native
- jarjar
- java_cup
- java-atk-wrapper
- javacc
- javacc-maven-plugin
- javaewah
- javaparser
- javapoet
- javassist
- jaxen
- jboss-annotations-1.2-api
- jboss-interceptors-1.2-api
- jboss-logmanager
- jboss-parent

- jctools
- jdepend
- jdependency
- jdom
- jdom2
- jetty
- jffi
- jflex
- jgit
- jline
- jnr-netdb
- jolokia-jvm-agent
- js-uglify
- jsch
- json_simple
- jss-javadoc
- jtidy
- junit5
- jvnet-parent
- jzlib
- kernel-cross-headers
- ksc
- ldapjdk-javadoc
- lensfun
- lensfun-devel
- libaec
- libaec-devel
- libappindicator-gtk3
- libappindicator-gtk3-devel

- libavc1394
- libblocksruntime
- libcacard
- libcacard-devel
- libcgroup
- libchamplain
- libchamplain-devel
- libchamplain-gtk
- libcroco
- libcroco-devel
- libcxl
- libcxl-devel
- libdap
- libdap-devel
- libdazzle-devel
- libdbusmenu
- libdbusmenu-devel
- libdbusmenu-doc
- libdbusmenu-gtk3
- libdbusmenu-gtk3-devel
- libdc1394
- libdnet
- libdnet-devel
- libdv
- libdwarf
- libdwarf-devel
- libdwarf-static
- libdwarf-tools
- libepubgen-devel

- libertas-sd8686-firmware
- libertas-usb8388-firmware
- libertas-usb8388-olpc-firmware
- libgdither
- libGLEW
- libgovirt
- libguestfs-benchmarking
- libguestfs-devel
- libguestfs-gfs2
- libguestfs-gobject
- libguestfs-gobject-devel
- libguestfs-java
- libguestfs-java-devel
- libguestfs-javadoc
- libguestfs-man-pages-ja
- libguestfs-man-pages-uk
- libguestfs-tools
- libguestfs-tools-c
- libhugetlbfs
- libhugetlbfs-devel
- libhugetlbfs-utils
- libIDL
- libIDL-devel
- libidn
- libiec61883
- libindicator-gtk3
- libindicator-gtk3-devel
- libiscsi-devel
- libjose-devel

- libldb-devel
- liblogging
- libluksmeta-devel
- libmcpp
- libmemcached
- libmetalink
- libmodulemd1
- libmongocrypt
- libmtp-devel
- libmusicbrainz5
- libmusicbrainz5-devel
- libnbd-devel
- liboauth
- liboauth-devel
- libpfm-static
- libpng12
- libpurple
- libpurple-devel
- libraw1394
- libsass
- libsass-devel
- libselinux-python
- libsqlite3x
- libtalloc-devel
- libtar
- libtdb-devel
- libtevent-devel
- libunwind
- libusal

- libvarlink
- libvirt-admin
- libvirt-bash-completion
- libvirt-daemon-driver-storage-gluster
- libvirt-daemon-driver-storage-iscsi-direct
- libvirt-devel
- libvirt-docs
- libvirt-gconfig
- libvirt-gobject
- libvirt-lock-sanlock
- libvncserver
- libwinpr-devel
- libwmf
- libwmf-devel
- libwmf-lite
- libXNVCtrl
- libyami
- log4j12
- lorax-composer
- lua-guestfs
- lucene
- mailman
- mailx
- make-devel
- maven-antrun-plugin
- maven-assembly-plugin
- maven-clean-plugin
- maven-dependency-analyzer
- maven-dependency-plugin

- maven/doxia
- maven/doxia-sitetools
- maven/install-plugin
- maven/invoker
- maven/invoker-plugin
- maven/parent
- maven/plugins-pom
- maven/reporting-api
- maven/reporting-impl
- maven/scm
- maven/script-interpreter
- maven/shade-plugin
- maven/shared
- maven/verifier
- maven2
- meanwhile
- mercurial
- metis
- metis-devel
- mingw32-bzip2
- mingw32-bzip2-static
- mingw32-cairo
- mingw32-expat
- mingw32-fontconfig
- mingw32-freetype
- mingw32-freetype-static
- mingw32-gstreamer1
- mingw32-harfbuzz
- mingw32-harfbuzz-static

- mingw32-icu
- mingw32-libjpeg-turbo
- mingw32-libjpeg-turbo-static
- mingw32-libpng
- mingw32-libpng-static
- mingw32-libtiff
- mingw32-libtiff-static
- mingw32-openssl
- mingw32-readline
- mingw32-sqlite
- mingw32-sqlite-static
- mingw64-adwaita-icon-theme
- mingw64-bzip2
- mingw64-bzip2-static
- mingw64-cairo
- mingw64-expat
- mingw64-fontconfig
- mingw64-freetype
- mingw64-freetype-static
- mingw64-gstreamer1
- mingw64-harfbuzz
- mingw64-harfbuzz-static
- mingw64-icu
- mingw64-libjpeg-turbo
- mingw64-libjpeg-turbo-static
- mingw64-libpng
- mingw64-libpng-static
- mingw64-libtiff
- mingw64-libtiff-static

- mingw64-nettle
- mingw64-openssl
- mingw64-readline
- mingw64-sqlite
- mingw64-sqlite-static
- modello
- mojo-parent
- mongo-c-driver
- mousetweaks
- mozjs52
- mozjs52-devel
- mozjs60
- mozjs60-devel
- mozvoikko
- msv-javadoc
- msv-manual
- munge-maven-plugin
- nbd
- nbdkit-devel
- nbdkit-example-plugins
- nbdkit-gzip-plugin
- ncompress
- net-tools
- netcf
- netcf-devel
- netcf-libs
- network-scripts
- nkf
- nss_nis

- nss-pam-ldapd
- objectweb-asm
- objectweb-pom
- ocaml-bisect-ppx
- ocaml-camlp4
- ocaml-camlp4-devel
- ocaml-lwt
- ocaml-mmap
- ocaml-ocplib-endian
- ocaml-ounit
- ocaml-result
- ocaml-seq
- opencv-contrib
- opencv-core
- opencv-devel
- openhpi
- openhpi-libs
- OpenIPMI-perl
- openssh-cav
- openssh-ldap
- openssl-ibmpkcs11
- opentest4j
- os-maven-plugin
- pakchois
- pandoc
- paranamer
- parfait
- parfait-examples
- parfait-javadoc

- pcp-parfait-agent
- pcp-pmda-rpm
- pcsc-lite-doc
- peripety
- perl-B-Debug
- perl-B-Lint
- perl-Class-Factory-Util
- perl-Class-ISA
- perl-DateTime-Format-HTTP
- perl-DateTime-Format-Mail
- perl-File-CheckTree
- perl-homedir
- perl-libxml-perl
- perl-Locale-Codes
- perl-Mozilla-LDAP
- perl-NKF
- perl-Object-HashBase-tools
- perl-Package-DeprecationManager
- perl-Pod-LaTeX
- perl-Pod-Plainer
- perl-prefork
- perl-String-CRC32
- perl-SUPER
- perl-Sys-Virt
- perl-tests
- perl-YAML-Syck
- phodav
- pidgin
- pidgin-devel

- pidgin-sipe
- pinentry-emacs
- pinentry-gtk
- pipewire0.2-devel
- pipewire0.2-libs
- plexus-ant-factory
- plexus-bsh-factory
- plexus-cli
- plexus-component-api
- plexus-component-factories-pom
- plexus-components-pom
- plexus-i18n
- plexus-interactivity
- plexus-pom
- plexus-velocity
- plymouth-plugin-throbgress
- powermock
- ptscotch-mpich
- ptscotch-mpich-devel
- ptscotch-mpich-devel-parmetis
- ptscotch-openmpi
- ptscotch-openmpi-devel
- purple-sipe
- python-nss-doc
- python-redis
- python-schedutils
- python-slip
- python-varlink
- python2-mock

- python3-click
- python3-cpio
- python3-custodia
- python3-flask
- python3-gevent
- python3-gobject-base
- python3-hivex
- python3-html5lib
- python3-hypothesis
- python3-ipatests
- python3-itsdangerous
- python3-jwt
- python3-libguestfs
- python3-mock
- python3-networkx-core
- python3-nose
- python3-nss
- python3-openipmi
- python3-pillow
- python3-pydbus
- python3-pymongo
- python3-pyOpenSSL
- python3-pytoml
- python3-reportlab
- python3-schedutils
- python3-scons
- python3-semantic_version
- python3-syspurpose
- python3-virtualenv

- python3-webencodings
- python3-werkzeug
- qemu-kvm-block-gluster
- qemu-kvm-block-iscsi
- qemu-kvm-tests
- qpdf
- qpid-proton
- qrencode
- qrencode-devel
- qrencode-libs
- qt5-qtcanvas3d
- qt5-qtcanvas3d-examples
- rarian
- rarian-compat
- re2c
- redhat-menus
- redhat-support-lib-python
- redhat-support-tool
- reflections
- regexp
- relaxngDatatype
- rhsm-gtk
- rpm-plugin-prioreset
- rsyslog-udpspoof
- ruby-hivex
- ruby-libguestfs
- rubygem-abrt
- rubygem-abrt-doc
- rubygem-mongo

- rubygem-mongo-doc
- samba-pidl
- samba-test
- samba-test-libs
- sane-frontends
- sanlk-reset
- scala
- scotch
- scotch-devel
- SDL_sound
- selinux-policy-minimum
- sendmail
- sgabios
- sgabios-bin
- shrinkwrap
- sisu-mojos
- SLOF
- sonatype-oss-parent
- sonatype-plugins-parent
- sparsehash-devel
- spec-version-maven-plugin
- spice
- spice-client-win-x64
- spice-client-win-x86
- spice-glib
- spice-glib-devel
- spice-gtk
- spice-gtk-tools
- spice-gtk3

- spice-gtk3-devel
- spice-gtk3-vala
- spice-parent
- spice-protocol
- spice-qxl-wddm-dod
- spice-server-devel
- spice-streaming-agent
- spice-vdagent-win-x64
- spice-vdagent-win-x86
- sssd-libwbclient
- stax-ex
- stax2-api
- stringtemplate
- stringtemplate4
- subscription-manager-initial-setup-addon
- subscription-manager-migration
- subscription-manager-migration-data
- subversion-javahl
- SuperLU
- SuperLU-devel
- supermin-devel
- swig
- swig-doc
- swig-gdb
- system-storage-manager
- testng
- timedatex
- treelayout
- trousers

- tycho
- uglify-js
- univocity-output-tester
- univocity-parsers
- usbguard-notifier
- usbredir-devel
- utf8cpp
- uthash
- velocity
- vinagre
- vino
- virt-dib
- virt-p2v-maker
- vm-dump-metrics-devel
- weld-parent
- wodim
- woodstox-core
- xdelta
- xmlgraphics-commons
- xmlstreambuffer
- xinetd
- xorg-x11-apps
- xorg-x11-drv-qxl
- xorg-x11-server-Xspice
- xpp3
- xsane-gimp
- xsom
- xz-java
- yajl-devel

- `yp-tools`
- `ypbind`
- `ypserv`

9.19. 非推奨のデバイスおよび非保守のデバイス

このセクションは、

- RHEL 8 のライフサイクルが終了するまで継続してサポートされるデバイス (ドライバー、アダプター) を説明しますが、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いため、新たに実装することは推奨されません。記載以外のデバイスのサポートは変更しません。これは **非推奨 デバイス** です。
- RHEL 8 では入手可能ですが、ルーチンベースでのテストや更新は行われていません。Red Hat は、独自の判断でセキュリティーバグなどの深刻なバグを修正する場合があります。このようなデバイスは実稼働環境では使用しなくなり、次のメジャーリリースでは無効になる可能性が高くなります。これは **未管理 デバイス** です。

PCI デバイス ID は、`vendor:device:subvendor:subdevice` の形式です。デバイス ID が記載されていない場合は、対応するドライバーに関連するすべてのデバイスが非推奨になっています。ご使用のシステムでハードウェアの PCI ID を確認するには、`lspci -nn` コマンドを実行します。

表9.1非推奨のデバイス

デバイス ID	ドライバー	デバイス名
	<code>bnx2</code>	QLogic BCM5706/5708/5709/5716 Driver
	<code>hpsa</code>	Hewlett-Packard Company: Smart アレイコントローラー
0x10df:0x0724	<code>lpfc</code>	Emulex Corporation: OneConnect FCoE Initiator (Skyhawk)
0x10df:0xe200	<code>lpfc</code>	Emulex Corporation: LPe15000/LPe16000 Series 8Gb/16Gb Fibre Channel Adapter
0x10df:0xf011	<code>lpfc</code>	Emulex Corporation: Saturn: LightPulse Fibre Channel Host Adapter
0x10df:0xf015	<code>lpfc</code>	Emulex Corporation: Saturn: LightPulse Fibre Channel Host Adapter
0x10df:0xf100	<code>lpfc</code>	Emulex Corporation: LPe12000 Series 8Gb Fibre Channel Adapter
0x10df:0xfc40	<code>lpfc</code>	Emulex Corporation: Saturn-X: LightPulse Fibre Channel Host Adapter
0x10df:0xe220	<code>be2net</code>	Emulex Corporation: OneConnect NIC (Lancer)

デバイス ID	ドライバー	デバイス名
0x1000:0x005b	megaraid_sas	Broadcom / LSI: MegaRAID SAS 2208 [Thunderbolt]
0x1000:0x006E	mpt3sas	Broadcom / LSI: SAS2308 PCI-Express Fusion-MPT SAS-2
0x1000:0x0080	mpt3sas	Broadcom / LSI: SAS2208 PCI-Express Fusion-MPT SAS-2
0x1000:0x0081	mpt3sas	Broadcom / LSI: SAS2208 PCI-Express Fusion-MPT SAS-2
0x1000:0x0082	mpt3sas	Broadcom / LSI: SAS2208 PCI-Express Fusion-MPT SAS-2
0x1000:0x0083	mpt3sas	Broadcom / LSI: SAS2208 PCI-Express Fusion-MPT SAS-2
0x1000:0x0084	mpt3sas	Broadcom / LSI: SAS2208 PCI-Express Fusion-MPT SAS-2
0x1000:0x0085	mpt3sas	Broadcom / LSI: SAS2208 PCI-Express Fusion-MPT SAS-2
0x1000:0x0086	mpt3sas	Broadcom / LSI: SAS2308 PCI-Express Fusion-MPT SAS-2
0x1000:0x0087	mpt3sas	Broadcom / LSI: SAS2308 PCI-Express Fusion-MPT SAS-2
	myri10ge	Myricom 10G driver (10GbE)
	netxen_nic	QLogic/NetXen (1/10) GbE Intelligent Ethernet Driver
0x1077:0x2031	qla2xxx	QLogic Corp.: ISP8324-based 16Gb Fibre Channel to PCI Express Adapter
0x1077:0x2532	qla2xxx	QLogic Corp.: ISP2532-based 8Gb Fibre Channel to PCI Express HBA
0x1077:0x8031	qla2xxx	QLogic Corp.: 8300 Series 10GbE Converged Network Adapter (FCoE)
	qla3xxx	QLogic ISP3XXX ネットワーク ドライバー v2.03.00-k5
0x1924:0x0803	sfc	Solarflare Communications: SFC9020 10G Ethernet Controller
0x1924:0x0813	sfc	Solarflare Communications: SFL9021 10GBASE-T Ethernet Controller

デバイス ID	ドライバー	デバイス名
	Soft-RoCE (rdma_rxe)	
	HNS-RoCE	

表9.2 未管理デバイス

デバイス ID	ドライバー	デバイス名
	e1000	Intel® PRO/1000 ネットワーク ドライバー
	mptbase	Fusion MPT SAS ホスト ドライバー
	mptsas	Fusion MPT SAS ホスト ドライバー
	mptscsih	Fusion MPT SCSI ホスト ドライバー
	mptspi	Fusion MPT SAS ホスト ドライバー
0x1000:0x0071 ^[a]	megaraid_sas	Broadcom / LSI: MR SAS HBA 2004
0x1000:0x0073 ^[a]	megaraid_sas	Broadcom / LSI: MegaRAID SAS 2008 [Falcon]
0x1000:0x0079 ^[a]	megaraid_sas	Broadcom / LSI: MegaRAID SAS 2108 [Liberator]

[a] RHEL 8.0 で無効になり、顧客の要求により RHEL 8.4 で再度有効になりました。

第10章 既知の問題

このパートでは Red Hat Enterprise Linux 8.5 の既知の問題を説明します。

10.1. インストーラーおよびイメージの作成

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 以降、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き、GUI インストールプログラムにより、リポジトリー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 との通信に必要とする証明書と CDN リポジトリー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)

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と、登録プロセスが失敗し、You must specify an organization for new units 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)

USB CD-ROM ドライブが Anaconda のインストールソースとして利用できない

USB CD-ROM ドライブがソースで、キックスタート `ignoredisk --only-use=` コマンドを指定すると、インストールに失敗します。この場合、Anaconda はこのソースディスクを見つけ、使用できません。

この問題を回避するには、**harddrive --partition=sdX --dir=/** コマンドを使用して USB CD-ROM ドライブからインストールします。その結果、インストールは失敗しなくなりました。

(BZ#1914955)

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 リポジトリでのみ使用可能です。

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)

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)

iso9660 ファイルシステムで、ハードドライブがパーティション分割されたインストールが失敗する

ハードドライブが **iso9660** ファイルシステムでパーティションが設定されているシステムには、RHEL をインストールできません。これは、**iso9660** ファイルシステムパーティションを含むハードディスクを無視するように設定されている、更新されたインストールコードが原因です。これは、RHEL が DVD を使用せずにインストールされている場合でも発生します。

この問題を回避するには、インストールの開始前に、キックスタートファイルに次のスクリプトを追加して、ディスクをフォーマットします。

メモ: 回避策を実行する前に、ディスクで利用可能なデータのバックアップを作成します。**wipefs** は、ディスク内の全データをフォーマットします。

```
%pre
wipefs -a /dev/sda
%end
```

その結果、インストールでエラーが発生することなく、想定どおりに機能します。

(BZ#1929105)

HASH MMU モードの IBM 電源システムが、メモリー割り当ての障害で起動できない

HASH メモリー割り当てユニット (MMU) モードの IBM Power Systems は、最大 192 コアの **kdump** に対応します。そのため、**kdump** が 192 コア以上で有効になっていると、メモリー割り当て失敗が原因でシステムの起動が失敗します。この制限は、**HASH MMU** モードの起動初期段階での RMA メモリーの割り当てによるものです。この問題を回避するには、**kdump** を使用する代わりに、**fadump** を有効にした **Radix MMU** モードを使用します。

(BZ#2028361)

Blueprint とキックスタートファイルの両方で同じユーザー名を追加すると、Edge イメージのインストールが失敗します。

RHEL for Edge イメージをインストールするには、**rhel-edge-container** イメージを構築する Blueprint を作成し、RHEL for Edge イメージをインストールするようにキックスタートファイルを作成する必要があります。ユーザーが Blueprint とキックスタートファイルの両方で同じユーザー名、パスワード、および SSH キーを追加すると、RHEL for Edge イメージのインストールに失敗します。現在、回避策はありません。

(BZ#1951964)

新しい **osbuild-composer** バックエンドが、アップグレード時に **lorax-composer** から Blueprint 状態に複製されない。

lorax-composer バックエンドから新しい **osbuild-composer** バックエンドにアップグレードする Image Builder ユーザーは、Blueprint が消えてしまう可能性があります。その結果、アップグレードが完了すると、Blueprint が自動的に表示されなくなります。この問題を回避するには、以下の手順を実行します。

前提条件

- **composer-cli** CLI ユーティリティーがインストールされている。

手順

1. 以下のコマンドを実行して、以前の **lorax-composer** ベースの Blueprint を新しい **osbuild-composer** バックエンドに読み込みます。

```
$ for blueprint in $(find /var/lib/lorax/composer/blueprints/git/workspace/master -name '*.toml'); do composer-cli blueprints push "${blueprint}"; done
```

これにより、**osbuild-composer** バックエンドで同じ Blueprint が利用できるようになりました。

関連情報

- この既知の問題の詳細は、[Image Builder Blueprint](#) が、Red Hat Enterprise Linux 8.3 への更新後に表示されなくなる 参照してください。

(BZ#1897383)

Anaconda がアプリケーションとして実行されているシステムでの予期しない SELinux ポリシー

Anaconda がすでにインストールされているシステムでアプリケーションとして実行されている場合(たとえば、`-image anaconda` オプションを使用してイメージファイルに別のインストールを実行する場合)、システムはインストール中に SELinux のタイプと属性を変更することを禁止されていません。そのため、SELinux ポリシーの特定の要素は、Anaconda が実行されているシステムで変更される可能性があります。この問題を回避するには、実稼働システムで Anaconda を実行せず、一時的な仮想マシンで実行します。そうすることで、実稼働システムの SELinux ポリシーは変更されません。`boot.iso` や `dvd.iso` からのインストールなど、システムインストールプロセスの一部として `anaconda` を実行しても、この問題の影響は受けません。

(BZ#2050140)

10.2. サブスクリプションの管理

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

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

(BZ#1687900)

10.3. ソフトウェア管理

システム上で CodeReady Linux Builder リポジトリが利用できない場合に `libdnf-devel` のアップグレードに失敗する

`libdnf-devel` パッケージが、BaseOS から CodeReady Linux Builder リポジトリに移動しました。そのため、システムで CodeReady Linux Builder リポジトリが利用できない場合に、`libdnf-devel` のアップグレードに失敗します。

この問題を回避するには、CodeReady Linux Builder リポジトリを有効にするか、アップグレード前に `libdnf-devel` パッケージを削除します。

(BZ#1960616)

`cr_compress_file_with_stat()` がメモリリークを引き起こす可能性がある

`createrepo_c` ライブラリーには API `cr_compress_file_with_stat()` 関数があります。この関数は、`char **dst` を 2 番目のパラメーターとして宣言します。他のパラメーターによって、`cr_compress_file_with_stat()` は、入力パラメーターとして `dst` を使用するか、割り当てられた文字列を返すために使用します。`dst` の内容をいつ解放するかユーザーに通知しないため、この予測できない動作によりメモリリークが発生する可能性があります。

この問題を回避するために、`dst` パラメーターを入力としてのみ使用する新しい API `cr_compress_file_with_stat_v2` 関数が追加されました。これは `char *dst` として宣言されます。これにより、メモリリークが回避されます。

cr_compress_file_with_stat_v2 関数は一時的で、RHEL 8 のみに存在することに注意してください。後で、**cr_compress_file_with_stat()** が代わりに修正されます。

(BZ#1973588)

10.4. シェルおよびコマンドラインツール

coreutils は、誤解を招く EPERM エラーコードを報告することがあります。

statx() システムコールを使用して、GNU コアユーティリティー (**coreutils**) が起動しました。 **seccomp** フィルターが、不明なシステムコールに対して EPERM エラーコードを返す場合、 EPERM は動作中の **statx()** の syscall が返す実際の **Operation not permitted** エラーと区別できないため、**coreutils** は、誤解を招く EPERM エラーコードを報告します。

この問題を回避するには、**seccomp** フィルターを更新して、**statx()** の syscall を許可するか、不明の syscall の ENOSYS エラーコードを返すようにします。

(BZ#2030661)

10.5. インフラストラクチャーサービス

FIPS モードの Postfix TLS フィンガープリントアルゴリズムを SHA-256 に変更する必要があります。

RHEL 8 のデフォルトでは、**postfix** は後方互換性に TLS を使用する MD5 フィンガープリントを使用します。ただし、FIPS モードでは、MD5 ハッシュ関数が利用できないため、デフォルトの postfix 設定で TLS が誤って機能する可能性があります。この問題を回避するには、postfix 設定ファイルのハッシュ関数を SHA-256 に変更する必要があります。

詳細は、関連するナレッジベースの記事 [Fix postfix TLS in the FIPS mode by switch to SHA-256 instead of the MD5](#) を参照してください。

(BZ#1711885)

brltty パッケージは multilib 対応ではない

brltty パッケージの 32 ビット版と 64 ビット版の両方をインストールすることはできません。32 ビット版 (**brltty.i686**) または 64 ビット版 (**brltty.x86_64**) いずれかのパッケージをインストールすることができます。64 ビット版を推奨します。

(BZ#2008197)

10.6. SECURITY

/etc/passwd- のファイル権限が CIS RHEL 8 Benchmark 1.0.0 と合致しない

CIS Benchmark の問題により、**/etc/passwd-** バックアップファイルの権限を保証する SCAP ルールの修正によって、権限が **0644** に設定されます。ただし、**CIS Red Hat Enterprise Linux 8 Benchmark 1.0.0** では、そのファイルに対するファイルパーミッション **0600** が必要です。そのため、修正後、**/etc/passwd-** のファイル権限はベンチマークに合うように設定されません。

(BZ#1858866)

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 コンテナーを処理します。

Red Hat Universal Base Image 8 (UBI 8) コンテナーは、**podman** の値ではなく、コンテナー 環境変数を **oci** 値に設定します。これにより、**udica** ツールがコンテナー 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 コンテナーを起動します。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 コンテナーの SELinux ポリシーを生成することができます。

(BZ#1763210)

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)

/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 システムの起動時に 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)

crypto-policies が Camellia 暗号を誤って許可する。

RHEL 8 システム全体の暗号化ポリシーでは、製品ドキュメントで説明されているように、すべてのポリシーレベルで Camellia 暗号を無効にする必要があります。ただし、Kerberos プロトコルでは、デフォルトでこの Camellia 暗号が有効になります。

この問題を回避するには、**NO-CAMELLIA** サブポリシーを適用します。

```
# update-crypto-policies --set DEFAULT:NO-CAMELLIA
```

これまでに上記のコマンドで、**DEFAULT** から切り替えたことがある場合は、**DEFAULT** を暗号化レベルの名前に置き換えます。

その結果、この回避策を使用して Cemellia 暗号を無効にしている場合に限り、システム全体の暗号化ポリシーを使用する全ポリシーで、この暗号化を適切に拒否できます。

([BZ#1919155](#))

IKEv2 で複数のラベルが付いた IPsec 接続が正常に動作しない

Libreswan が **IKEv2** プロトコルを使用する場合、IPsec のセキュリティーラベルは複数の接続では正しく機能しません。これにより、ラベルが付いた IPsec を使用する Libreswan は、最初の接続のみを確立できますが、後続の接続を確立することができません。複数の接続を使用するには、**IKEv1** プロトコルを使用します。

([BZ#1934859](#))

OpenSC pkcs15-init によるスマートカードのプロビジョニングプロセスが適切に動作しない

file_caching オプションは、デフォルトの OpenSC 設定で有効になっているため、キャッシュ機能は **pkcs15-init** ツールから一部のコマンドを適切に処理しません。したがって、OpenSC を使用したスマートカードのプロビジョニングプロセスは失敗します。

この問題を回避するには、以下のスニペットを **/etc/opensc.conf** ファイルに追加します。

```
app pkcs15-init {
    framework pkcs15 {
        use_file_caching = false;
    }
}
```

pkcs15-init を使用したスマートカードのプロビジョニングは、前述の回避策を適用している場合に限り機能します。

([BZ#1947025](#))

SHA-1署名を使用するサーバーへの接続が GnuTLS で動作しない

証明書の SHA-1 署名は、GnuTLS セキュアな通信ライブラリーにより、セキュアでないものとして拒否されます。したがって、TLS のバックエンドとして GnuTLS を使用するアプリケーションは、このような証明書を提供するピアへの TLS 接続を確立することができません。この動作は、その他のシステム暗号化ライブラリーと一貫性がありません。

この問題を回避するには、サーバーをアップグレードして、SHA-256 または強力なハッシュを使用して署名した証明書を使用するか、LEGACY ポリシーに切り替えます。

([BZ#1628553](#))

FIPS モードの OpenSSL が、特定の D-H パラメーターのみを受け入れます。

FIPS モードでは、OpenSSL を使用する TLS クライアントは **bad dh value** エラーを返し、手動で生成されたパラメーターを使用するようにサーバーへの TLS 接続を中止します。これは、FIPS 140-2 に準拠するよう設定されている場合、OpenSSL が NIST SP 800-56A rev3 付録 D (RFC 3526 で定義されたグループ 14、15、16、17、18、および RFC 7919 で定義されたグループ) に準拠した Diffie-Hellman パラメーターでのみ機能するためです。また、OpenSSL を使用するサーバーは、その他のパラメーターをすべて無視し、代わりに同様のサイズの既知のパラメーターを選択します。この問題を回避するには、準拠するグループのみを使用します。

(BZ#1810911)

IKE over TCP 接続がカスタム TCP ポートで機能しない

tcp-remoteport Libreswan 設定オプションが適切に動作しません。したがって、デフォルト以外の TCP ポートを指定する必要があるシナリオでは、IKE over TCP 接続を確立することができません。

(BZ#1989050)

SELinux Audit ルールと SELinux ブール値設定での競合

Audit ルールリストに、**subj_*** または **obj_*** フィールドを含む Audit ルールが含まれ、SELinux ブール値の設定が変更された場合は、SELinux ブール値を設定するとデッドロックが発生します。その結果、システムが応答しなくなり、復旧に再起動が必要になります。この問題を回避するには、**subj_*** または **obj_*** フィールドを含む Audit ルールをすべて無効にするか、SELinux ブール値を変更する前にこのようなルールを一時的に無効にします。

[RHSA-2021:2168](#) アドバイザリーのリリースにより、カーネルはこの状況を適切に処理し、デッドロックが発生しなくなりました。

(BZ#1924230)

systemd が任意のパスからコマンドを実行できない

SELinux ポリシーアッケージにはこのようなルールが含まれていないため、**systemd** サービスは **/home/user/bin** の任意のパスからコマンドを実行できません。そのため、システム以外のパスで実行されるカスタムサービスは失敗し、SELinux がアクセスを拒否すると、AVC (アクセスベクターキャッシュ) 監査メッセージをログに記録します。この問題を回避するには、以下のいずれかを実行します。

- **-c** オプションを指定し、シェルスクリプトを使用してコマンドを実行します。以下に例を示します。

bash -c command

- **/bin**、**/sbin**、**/usr/sbin**、**/usr/local/bin**、**/usr/local/sbin** の共通のディレクトリーを使用して共通のパスからコマンドを実行します。

(BZ#1860443)

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る。

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、修正を回実行して、番目の実行で依存ルールを修正します。

(BZ#1750755)

Server with GUI または Workstation ソフトウェアの選択と CIS セキュリティープロファイルを使用したインストールはできません。

CIS セキュリティープロファイルは、**Server with GUI** および **Workstation** ソフトウェアの選択と互換性がありません。そのため、**Server with GUI** ソフトウェアの選択と CIS プロファイルを使用した RHEL 8 のインストールはできません。CIS プロファイルと、これらのソフトウェアの選択のいずれかを使用したインストール試行では、エラーメッセージが生成されます。

package xorg-x11-server-common has been added to the list of excluded packages, but it can't be removed from the current software selection without breaking the installation.

この問題を回避するには、**Server with GUI** または **Workstation** ソフトウェアの選択で CIS セキュリティープロファイルを使用しないでください。

(BZ#1843932)

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)

usbguard-notifier がジャーナルに記録するエラーメッセージが多すぎる

usbguard-notifier サービスには、**usbguard-daemon** IPC インターフェイスに接続するためのプロセス間通信 (IPC) のパーミッションがありません。したがって、**usbguard-notifier** はインターフェイスへの接続に失敗し、対応するエラーメッセージがジャーナルに書き込まれます。**usbguard-notifier** は **-wait** オプションで始まるため、デフォルトでは接続障害後に毎秒 **usbguard-notifier** が IPC インターフェイスへの接続を試みるため、ログにはこれらのメッセージが過剰に含まれます。

この問題を回避するには、**usbguard-notifier** が実行されているユーザーまたはグループが IPC インターフェイスに接続するのを許可します。たとえば、以下のエラーメッセージには、GNOME Display Manager(GDM) の UID および GID 値が含まれます。

IPC connection denied: uid=42 gid=42 pid=8382, where uid and gid 42 = gdm

不足しているパーミッションを **gdm** ユーザーに付与するには、**usbguard** コマンドを使用して **usbguard** デーモンを再起動します。

```
# usbguard add-user gdm --group --devices listen
# systemctl restart usbguard
```

不足しているパーミッションを付与した後に、エラーメッセージがログに表示されなくなります。

(BZ#2000000)

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)

10.7. ネットワーク

nm-cloud-setup サービスは、手動で設定されたセカンダリー IP アドレスをインターフェイスから削除する

クラウド環境から受け取った情報に基づいて、**nm-cloud-setup** サービスがネットワークインターフェイスを設定します。インターフェイスを手動で設定するには、**nm-cloud-setup** を無効にします。ただし、場合によっては、ホスト上の他のサービスもインターフェイスを設定できます。たとえば、これらのサービスはセカンダリー IP アドレスを追加できます。**nm-cloud-setup** がセカンダリー IP アドレスを削除しないようにするには、

1. **nm-cloud-setup** サービスおよびタイマーを停止して無効にします。

```
# systemctl disable --now nm-cloud-setup.service nm-cloud-setup.timer
```

2. 使用可能な接続プロファイルを表示します。

```
# nmcli connection show
```

3. 影響を受ける接続プロファイルを再アクティビ化します。

```
# nmcli connection up "<profile_name>"
```

その結果、このサービスは、手動で設定されたセカンダリー IP アドレスをインターフェイスから削除しなくなりました。

(BZ#2132754)

NetworkManager は、特定の順番でボンドおよびチームポートをアクティベートしない

NetworkManager は、インターフェイス名のアルファベット順にインターフェイスをアクティビ化します。ただし、起動中にインターフェイスが後で表示される場合、たとえば、カーネルがインターフェイスを検出するのにより多くの時間が必要な場合、NetworkManager は後でこのインターフェイスをアクティビ化します。NetworkManager は、ボンドおよびチームポートの優先順位の設定に対応していません。したがって、NetworkManager がこのデバイスのポートをアクティビ化する順番は、常に予測できるとは限りません。この問題を回避するには、ディスパッチャスクリプトを作成します。

このようなスクリプトの例は、チケット内の対応する [コメント](#) を参照してください。

(BZ#1920398)

IPv6_rpfilter オプションが有効になっているシステムでネットワークスループットが低下

firewalld.conf ファイルで **IPv6_rpfilter** オプションが有効になっているシステムでは、100-Gbps リンクなどの高いトラフィックシナリオの場合、現時点でパフォーマンスは最適ではなくネットワークスループットが低下します。この問題を回避するには、**IPv6_rpfilter** オプションを無効にします。これを

行うには、`/etc/firewalld/firewalld.conf` ファイルに次の行を追加します。

```
IPv6_rpfilter=no
```

その結果、システムはパフォーマンスが向上しますが、同時にセキュリティーは低下します。

(BZ#1871860)

10.8. カーネル

同一の `crash` 拡張機能を再読み込みすると、セグメンテーションフォルトが発生する場合がある

読み込み済みのクラッシュ拡張ファイルのコピーを読み込むと、セグメンテーションフォルトが発生する場合があります。現在、`crash` ユーティリティーは、元のファイルが読み込まれているかどうかを検出します。その結果、`crash` ユーティリティーに同一のファイルが2つ共存するため、名前空間コリジョンが発生し、クラッシュユーティリティーが起動してセグメンテーションフォルトが発生します。

この問題を回避するには、クラッシュ拡張ファイルを一度だけ読み込みます。その結果、セグメンテーションフォルトは上記のシナリオでは発生しなくなりました。

(BZ#1906482)

`vmcore` キャプチャーが、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する

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)

RHEL 8 で、デバッグカーネルがクラッシュキャプチャー環境で起動に失敗する

デバッグカーネルはメモリーを大量に消費するので、デバッグカーネルが使用中で、カーネルパニックが発生すると、問題が発生します。その結果、デバッグカーネルはキャプチャーカーネルとして起動できず、代わりにスタックトレースが生成されます。この問題を回避するには、必要に応じてクラッシュカーネルメモリーを増やします。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)

起動時にクラッシュカーネルメモリーの割り当てに失敗する

特定の Ampere Altra システムでは、BIOS 設定で 32 ビットリージョンが無効になっていると、起動時にクラッシュカーネルメモリーを割り当てることに失敗します。したがって、**kdump** サービスが起動できません。これは、クラッシュカーネルメモリーを含むのに十分な大きさのフラグメントがない場合に、4 GB 未満のリージョンのメモリーの断片化によって生じます。

この問題を回避するには、以下のように BIOS で 32 ビットのメモリーリージョンを有効にします。

1. システムで BIOS 設定を開きます。
2. **Chipset** メニューを開きます。
3. **Memory Configuration** で、**Slave 32-bit** オプションを有効にします。

これにより、32 ビットリージョン内のクラッシュカーネルメモリー割り当てに成功し、**kdump** サービスが期待どおりに機能します。

(BZ#1940674)

デフォルトのクラッシュカーネルメモリーを使用する KVM 仮想マシンで kdump が失敗する

一部の KVM 仮想マシンでは、**kdump** にデフォルトメモリー量を使用してカーネルクラッシュダンプをキャプチャーすると **kdump** に失敗します。その結果、クラッシュカーネルにより、以下のエラーが表示されます。

```
/bin/sh: error while loading shared libraries: libtinfo.so.6: cannot open shared object file: No such file or directory
```

この問題を回避するには、**kdump** のサイズ要件に合うように、**crashkernel=** オプションを 32M 以上増やします。たとえば、**final** の値は、現在の値と 32M の合計になります。

crashkernel=auto パラメーターの場合、以下のようになります。

1. 現在のメモリーサイズを確認し、次のようにして 32M 増加します。

```
echo $(( $(cat /sys/kernel/kexec_crash_size)/1048576+32))M
```

2. **kernel** **crashkernel** パラメーターを **crashkernel=x** に設定します。**x** は増加したサイズになります。

(BZ#2004000)

QAT マネージャーが LKCF のスペアデバイスを残さない

Intel® QuickAssist Technology(QAT) マネージャー (**qatmgr**) はユーザー空間プロセスであり、デフォルトではシステム内のすべての QAT デバイスを使用します。これにより、Linux Kernel Cryptographic Framework(LKCF) には QAT デバイスが残っていません。この動作は予想され、大多数のユーザーはユーザースペースからのアクセラレーションを使用するため、この状況を回避する必要はありません。

(BZ#1920086)

カーネル ACPI ドライバーは、PCIe ECAM メモリーリージョンにアクセスできないことを報告します。

ファームウェアが提供する Advanced Configuration and Power Interface (ACPI) テーブルは、PCI バスデバイスの現在のリソース設定 (_CRS) メソッドにおいて PCI バス上のメモリーリージョンを定義しません。したがって、システムの起動時に以下の警告メッセージが表示されます。

```
[ 2.817152] acpi PNP0A08:00: [Firmware Bug]: ECAM area [mem 0x30000000-0x31ffff] not reserved in ACPI namespace
[ 2.827911] acpi PNP0A08:00: ECAM at [mem 0x30000000-0x31ffff] for [bus 00-1f]
```

ただし、カーネルは依然として **0x30000000-0x31ffff** メモリーリージョンにアクセスできます。また、そのメモリーリージョンを PCI Enhanced Configuration Access Mechanism (ECAM) に適切に割り当てるすることができます。以下の出力で 256 バイトオフセットで PCIe 設定領域にアクセスして、PCI ECAM が正常に機能することを確認できます。

```
03:00.0 Non-Volatile memory controller: Sandisk Corp WD Black 2018/PC SN720 NVMe SSD (prog-if 02 [NVM Express])
```

```
...
Capabilities: [900 v1] L1 PM Substates
L1SubCap: PCI-PM_L1.2- PCI-PM_L1.1- ASPM_L1.2+ ASPM_L1.1- L1_PM_Substates+
PortCommonModeRestoreTime=255us PortTPowerOnTime=10us
L1SubCtl1: PCI-PM_L1.2- PCI-PM_L1.1- ASPM_L1.2- ASPM_L1.1-
T_CommonMode=0us LTR1.2_Threshold=0ns
L1SubCtl2: T_PwrOn=10us
```

これにより、警告メッセージを無視します。

問題の詳細は、["Firmware Bug: ECAM area **mem 0x30000000-0x31ffff** not reserved in ACPI namespace" appears during system boot](#) を参照してください。

(BZ#1868526)

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る

tuned-adm profile powersave コマンドを実行すると、古い Thunderx (CN88xx) プロセッサーを持つ 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)

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時に **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** に **irqpoll** を追加します。

```
KDUMP_COMMANDLINE_REMOVE="hugepages hugepagesz slub_debug quiet
log_buf_len swiotlb"
```

2. **/etc/sysconfig/kdump** ファイルの **KDUMP_COMMANDLINE_APPEND** から **irqpoll** を削除します。

```
KDUMP_COMMANDLINE_APPEND="irqpoll nr_cpus=1 reset_devices
cgroup_disable=memory udev.children-max=2 panic=10 swiotlb=noforce novmcoredd"
```

3. **kdump** サービスを再起動します。

```
systemctl restart kdump
```

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。

この既知の問題の関連情報は、[irqpoll カーネルコマンドラインパラメーターにより、vmcore 生成エラーが発生する場合がある](#) を参照してください。

(BZ#1654962)

HP NMI ウオ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 ウオッチドッグの **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 ウオ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** ウオッチドッグ。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)

仮想マシンへの仮想機能の割り当て時に接続に失敗する

ionic デバイスドライバーを使用する Pensando ネットワークカードは、VLAN タグ設定要求を許可し、ネットワーク仮想機能 (**VF**) を **VM** に割り当てる間にネットワーク接続の設定を試行します。この機能はカードのファームウェアではサポートされていないため、このようなネットワーク接続は失敗します。

(BZ#1930576)

OPEN MPI ライブラリーは、デフォルトの PML でランタイムが失敗する可能性があります。

OPEN Message Passing Interface (OPEN MPI) 実装 4.0.x シリーズでは、UCX (Unified Communication X) がデフォルトの PML (ポイントツーポイント) です。OPEN MPI 4.0.x シリーズの新しいバージョンでは、**openib** Byte Transfer Layer (BTL) が非推奨になりました。

ただし、OPEN MPI は **同種** クラスター (同じハードウェアおよびソフトウェア設定) で実行される場合も、UCX は MPI **openib** の一方向操作に BTL を使用します。これにより、実行エラーが発生す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。

- 以下のパラメーターを使用して **mpirun** コマンドを実行します。

```
-mca btl openib -mca pml ucx -x UCX_NET_DEVICES=mlx5_ib0
```

詳細は以下のようになります。

- mca btl openib** パラメーターは **openib** BTL を無効にします。
- mca pml ucx** パラメーターは、**ucx** PML を使用するように OPEN MPI を設定します。
- x UCX_NET_DEVICES=** パラメーターは、指定したデバイスを使用するように UCX を制限します。

OPEN MPI は、**異種** クラスター (ハードウェアおよびソフトウェア設定に異なる) を実行する場合は、デフォルトの PML として UCX を使用します。これにより、OPEN MPI ジョブが不安定なパフォーマンス、応答しない動作で実行されたり、またはクラッシュによる不具合とともに実行される可能性があります。この問題を回避するには、UCX の優先度を以下のように設定します。

- 以下のパラメーターを使用して **mpirun** コマンドを実行します。

```
-mca pml_uxc_priority 5
```

これにより、OPEN MPI ライブラリーは、UCX を介して利用可能な別のトランスポート層を選択することができます。

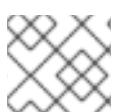
(BZ#1866402)

Solarflare が、最大数の VF(Virtual Function) の作成に失敗する

Solarflare NIC は、リソースが十分にないため、最大数の VF の作成に失敗します。PCIe デバイスが作成できる VF の最大数は、**/sys/bus/pci/devices/PCI_ID/sriov_totalvfs** ファイルで確認できます。この問題を回避するには、起動時に **Solarflare Boot Manager** から、または Solarflare **sfboot** ユーティリティの使用により、VF の数または VF MSI 割り込みの値を低い値に調整できます。デフォルトの VF MSI 割り込みの値は **8** です。

- sfboot** を使用して VF MSI 割り込み値を調整するには、以下を実行します。

```
# sfboot vf-msix-limit=2
```



注記

VF MSI 割り込みの値を調整すると、VF のパフォーマンスに影響します。

調整されるパラメーターの詳細は、**Solarflare Server Adapter user guide** を参照してください。

(BZ#1971506)

10.9. ハードウェアの有効化

デフォルトの 7417 printk 値により、一時的にシステムが応答しなくなる

デフォルトの 7417 printk 値を使用することで、カーネルアクティビティーのデバッグを改善できます。ただし、シリアルコンソールと組み合わせると、この printk 設定により、RHEL システムが一時的に応答しなくなるような激しい I/O がバーストする可能性があります。この問題を回避するには、新しい **optimize-serial-console** TuneD プロファイルを追加し、デフォルトの printk 値を 4417 に減らします。ユーザーは、以下のようにシステムをインストルメント化できます。

```
# tuned-adm profile throughput-performance optimize-serial-console
```

再起動後も printk 値を短くすると、システムがハングする可能性が低くなります。

この設定変更は、余分なデバッグ情報が失われる代償を伴うことに注意してください。

(JIRA:RHELPLAN-28940)

10.10. ファイルシステムおよびストレージ

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。

- **pvmove** コマンドを使用すると、**writecache** 論理ボリュームに名前を付けることはできません。
- **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。

以下の制限は、**cache** メソッドにも適用されます。

- **cache** または **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。

(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 **raid1** に変換するには、ミラーリングされた LVM デバイスの RAID1 デバイスへの変換を参照してください。

(BZ#1730502)

/boot ファイルシステムを LVM に配置することができない

/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)

GRUB が、ブート時の初回の失敗後にディスクへのアクセスを再試行する

ストレージエリアネットワーク (SAN) が **open** および **read** のディスク呼び出しを確認しない場合があります。以前は、GRUB ツールが **grub_rescue** プロンプトに切り替わり起動に失敗していました。今回の更新で、ディスク **open** および **read** の初回呼び出しが失敗した後、GRUB は最大 20 回ディスクへのアクセスを再試行します。これらの試行後に GRUB ツールがまだディスクの **open** または **read** ができない場合は、**grub_rescue** モードに切り替わります。

(BZ#1987087)

XFS クオータ警告が頻繁にトリガーされる

クオータタイマーを使用すると、クオータ警告が頻繁にトリガーされるため、ソフトクオータが必要以上に速く実行されます。この問題を回避するには、警告のトリガーを妨げるソフトクオータを使用しないでください。その結果、警告メッセージの量はソフトクオータ制限を強制せず、設定されたタイムアウトを尊重するようになります。

(BZ#2059262)

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

OQGraph プラグインが有効な場合に、**MariaDB 10.5** が存在しないテーブルの破棄について警告しない

OQGraph ストレージエンジンプラグインが **MariaDB 10.5** サーバーに読み込まれると、**MariaDB** は存在しないテーブルの削除について警告しません。特に、ユーザーが **DROP TABLE** コマンドまたは **DROP TABLE** または **DROP TABLE IF EXISTS** SQL コマンドを使用して存在しないテーブルをドロップしようとすると、**MariaDB** はエラーメッセージを返したり警告をログに記録したりしません。

OQGraph プラグインは **mariadb-oqgraph-engine** パッケージにより提供されることに注意してください。デフォルトではインストールされません。

(BZ#1944653)

MariaDB では PAM プラグインバージョン 1.0 が機能しない

MariaDB 10.3 は、PAM (Pluggable Authentication Modules) プラグインバージョン 1.0 を提供します。**MariaDB 10.5** は、プラグインバージョン 1.0 および 2.0 を提供します。バージョン 2.0 がデフォルトです。

RHEL 8 では、**MariaDB** PAM プラグインバージョン 1.0 は機能しません。この問題を回避するには、**mariadb:10.5** モジュールストリームによって提供される PAM プラグインバージョン 2.0 を使用します。

(BZ#1942330)

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)

OpenLDAP ライブラリー間のシンボルの競合により、**httpd** でクラッシュが発生することがある

OpenLDAP が提供する **libldap** ライブラリーと **libldap_r** ライブラリーの両方が、単一のプロセス内にロードされ、使用されると、これらのライブラリー間でシンボルの競合が発生する可能性があります。そのため、**httpd** 設定によって **mod_security** または **mod_auth_openidc** モジュールもロードされると、PHP **ldap** 拡張機能を使用する Apache **httpd** 子プロセスが突然終了する可能性があります。

Apache Portable Runtime (APR) ライブラリーに対する RHEL 8.3 の更新では、**APR_DEEPBIND** 環境変数を設定することでこの問題を回避できます。これにより、**httpd** モジュールのロード時に **RTLD_DEEPBIND** 動的リンクオプションを使用できるようになります。**APR_DEEPBIND** 環境変数を有効にすると、競合するライブラリーをロードする **httpd** 設定でクラッシュが発生しなくなります。

(BZ#1819607)

10.12. コンパイラーや開発ツール

同じ入力ストリームで CryptBlocks を複数回使用すると、暗号化が正しく行われない

Go FIPS モードを有効にすると、AES CBC CryptBlocks は初期化ベクターを誤って再初期化します。これにより、入力ストリームで CryptBlocks を複数回使用すると、ファイルが誤って暗号化されます。この問題を回避するには、**aes-cbc** インターフェイスで IV を再初期化しないでください。このアクションにより、ファイルを正しく暗号化できます。

(BZ#1972825)

10.13. IDENTITY MANAGEMENT

Windows Server 2008 R2 以前に対応しなくなった

RHEL 8.4 以降では、Identity Management (IdM) は、Windows Server 2008 R2 以前のバージョンを実行している Active Directory ドメインコントローラーとの間で Active Directory への信頼を確立することに対応していません。RHEL IdM との信頼関係を確立する際に、SMB 暗号化が必要になりました。これは、Windows Server 2012 以降でのみ利用可能です。

(BZ#1971061)

--agent-uid pkidbuser オプションを指定して **cert-fix** ユーティリティーを使用すると、証明書システムが破損します。

--agent-uid pkidbuser オプションを指定して **cert-fix** ユーティリティーを使用すると、証明書システムの LDAP 設定が破損します。したがって、証明書システムは不安定になり、システムの復元に手動の操作が必要になる可能性があります。

(BZ#1729215)

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)

IdM ホストの **/var/log/lastlog** 分析ファイルが、パフォーマンスの問題を引き起こす可能性があります。

IdM のインストール時に、利用できる合計 10,000 の範囲からの 200,000 の UID の範囲が無作為に選択され、割り当てられます。このようにランダムな範囲を選択すると、今後別の 2 つの IdM ドメインを統合する場合に、ID の競合が発生する可能性を大幅に削減できます。

ただし、UID が多いと、**/var/log/lastlog** ファイルで問題が発生する可能性があります。たとえば、1280000008 の UID を持つユーザーが IdM クライアントにログインすると、ローカルの **/var/log/lastlog** ファイルサイズは、約 400 GB に増えます。実際のファイルはスペースで、その領域をすべて使用しません。ただし、一部のアプリケーションはデフォルトではスペースファイルを識別するように設計されています。そのため、それらを処理する特定のオプションが必要になる場合があります。たとえば、設定が複雑でバックアップ、コピーアプリケーションがスペースファイルを正しく処理しない場合、ファイルはサイズが 400 GB であるかのようにコピーされます。この動作により、パフォーマンスの問題が発生す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。

- 標準パッケージの場合は、そのドキュメントを参照して、スペースファイルを処理するオプションを特定します。
- カスタムアプリケーションの場合、`/var/log/lastlog` などのスペースファイルを正しく管理できることを確認してください。

(JIRA:RHELPLAN-59111)

FIPS モードは、共有シークレットを使用したフォレスト間の信頼を確立することをサポートしません。

NTLMSSP 認証は FIPS に準拠していないため、FIPS モードでフォレスト間の信頼を確立できません。この問題を回避するには、FIPS モードが有効な IdM ドメインと AD ドメインとの間に信頼を確立する際に、Active Directory (AD) 管理アカウントで認証します。

(BZ#1924707)

FreeRADIUS サーバーが FIPS モードでの実行に失敗する

デフォルトでは、FIPS モードでは、OpenSSL は MD5 ダイジェストアルゴリズムの使用を無効にします。RADIUS プロトコルでは、RADIUS クライアントと RADIUS サーバー間のシークレットを暗号化するために MD5 が必要であるため、FreeRADIUS サーバーが FIPS モードで失敗します。

この問題を回避するには、以下の手順に従います。

手順

- radiusd サービスの環境変数 `RADIUS_MD5_FIPS_OVERRIDE` を作成します。

```
systemctl edit radiusd

[Service]
Environment=RADIUS_MD5_FIPS_OVERRIDE=1
```

- 変更を適用するには、`systemd` 設定を再読み込みし、`radiusd` サービスを開始します。

```
# systemctl daemon-reload
# systemctl start radiusd
```

- デバッグモードで FreeRADIUS を実行するには、以下を実行します。

```
# RADIUS_MD5_FIPS_OVERRIDE=1 radiusd -X
```

FreeRADIUS は FIPS モードで実行できますが、FIPS モードでは弱い暗号と関数が使用されるため、これは FIPS に準拠するわけではありません。

FIPS モードでの FreeRADIUS 認証の設定方法は、[FIPS モードで FreeRADIUS 認証を設定する方法](#) を参照してください。

(BZ#1958979)

Samba をプリントサーバーとして実行する際にアクションが必要

今回の更新で、`samba` パッケージが `/var/spool/samba/` ディレクトリーを作成しなくなりました。プリントサーバーとして Samba を使用し、`[printers]` 共有の `/var/spool/samba/` を使用してプリントジョブをスプールすると、SELinux は Samba ユーザーがこのディレクトリーにファイルを作成しないよう

にします。したがって、印刷ジョブが失敗し、**auditd** サービスは **/var/log/audit/audit.log** に **denied** メッセージを記録します。システムを RHEL 8.5 に更新した後にこの問題を回避するには、以下を行います。

1. **/etc/samba/smb.conf** ファイルで **[printers]** 共有を探します。
2. 共有定義に **path = /var/spool/samba/** が含まれる場合は、設定を更新して、**path** パラメーターを **/var/tmp/** に設定します。
3. **smbd** サービスを再起動します。

```
# systemctl restart smbd
```

RHEL 8.5 に Samba を新たにインストールした場合は、何もする必要はありません。RHEL 8.5 の **samba-common** パッケージが提供するデフォルトの **/etc/samba/smb.conf** ファイルは、すでに **/var/tmp/** ディレクトリーを使用してプリントジョブをスプールします。

(BZ#2009213)

NSS で有効になっている暗号の **default** キーワードは、他の暗号と組み合わせても機能しません

Directory Server では、**default** キーワードを使用して、ネットワークセキュリティーサービス (NSS) で有効になっているデフォルトの暗号を参照することができます。しかし、コマンドラインまたは Web コンソールを使用してデフォルトの暗号および追加の暗号を有効にする場合、Directory Server は **default** キーワードの解決に失敗します。その結果、サーバーは追加で指定された暗号のみを有効にし、以下のエラーを記録します。

```
Security Initialization - SSL alert: Failed to set SSL cipher preference information: invalid ciphers
<default,+__cipher_name__>; format is +cipher1,-cipher2... (Netscape Portable Runtime error 0 - no
error)
```

回避策としては、追加で有効にしたいものも含めて、NSS でデフォルトで有効になっているすべての暗号を指定してください。

(BZ#1817505)

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/sssd/sssd.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)

グループのサイズが 1500 人を超えると、SSSD が取得するメンバーリストが不完全なものになる

SSSD と Active Directory の統合時に、グループサイズが 1500 メンバーを超えると、SSSD が取得するメンバーリストが不完全なものになります。この問題は、1回のクエリーで取得できるメンバーの数を制限する Active Directory の MaxValRange ポリシーが、デフォルトで 1500 に設定されているために発生します。

この問題を回避するには、Active Directory の MaxValRange 設定を変更して、より大きなグループサイズに対応できるようにします。

(JIRA:RHELDODCS-19603)

10.14. デスクトップ

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)

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)

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)

Flatpak の現在の制限

Flatpak パッケージマネージャーを使用して特定のアプリケーションをインストールできます。ただし、Flatpak には現在、特定の機能または機能がありません。以下に例を示します。

- Flatpak には CVE と changelog 機能のパリティーがありません。Flatpak アプリケーションに GNOME Software アプリケーションを使用すると、現在、それぞれのパッケージまたは CVE に関する情報は提供されません。
- Red Hat Flatpak リモート リポジトリを追加する場合、GPG キーチェックはデフォルトで無効になっています。
- Flatpak アプリケーションには固有のアイコンがありません。GNOME Software では、アプリ

ケーションは **rpm** バージョンと **Flatpak** バージョンの両方を表示します。回避策として、それぞれのアイコンの **Show Details** をクリックすると、アプリケーションの生成元を見つけることができます。

- **Flatpak** アプリケーションは Kerberos チケットを処理できません。
- **Flatpak** アプリケーションからの印刷は現在利用できません。
- Red Hat **Flatpak リモート** は自動的に追加されません。それらを有効にするには、製品ドキュメント [Red Hat Flatpak リモートの有効化](#) の指示に従ってください。

(JIRA:RHELPLAN-100230)

10.15. グラフィックインフラストラクチャー

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、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)

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/ksmp.conf** ファイルに以下の行を追加して、**kdmp** で **radeon** を無効にします。

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

マシンと **kdmp** を再起動します。**kdmp** 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、**kdmp** 中にグラフィックは利用できませんが、**kdmp** は正常に動作します。

(BZ#1694705)

ビデオメモリーが少なくなったため、ESXi の GUI がクラッシュする可能性がある

vCenter Server 7.0.1 を使用する VMware ESXi 7.0.1 ハイパーバイザーの RHEL 仮想マシンでグラフィカルユーザーインターフェイス (GUI) には、一定量のビデオメモリーが必要です。複数のコンソールまたは高解像度のモニターを仮想マシンに接続する場合、GUI には少なくとも 16 MB のビデオメモリーが必要です。ビデオメモリーが少ないので GUI を起動すると、GUI が突然終了する可能性があります。

この問題を回避するには、仮想マシンに 16 MB 以上のビデオメモリーを割り当てるようハイパーバイザーを設定します。その結果、仮想マシンの GUI がクラッシュしなくなりました。

(BZ#1910358)

VNC Viewer が、IBM Z で 16 ビットのカラーデプスで誤った色を表示

VNC Viewer アプリケーションは、16 ビットのカラーデプスで IBM Z サーバーの VNC セッションに接続すると、誤った色を表示します。

この問題を回避するには、VNC サーバーで 24 ビットのカラーデプスを設定します。Xvnc サーバーの場合は、Xvnc 設定で **-depth 16** オプションを **-depth 24** に置き換えます。

その結果、VNC クライアントで色が正しく表示されますが、サーバーでは、より多くのネットワーク帯域幅が使用されます。

(BZ#1886147)

VGA ディスプレイを備えた MatroxGPU が出力を表示しない

以下のシステム設定を使用すると、ディスプレイにグラフィカル出力が表示されない場合があります。

- Matrox MGA G200 ファミリーの GPU
- VGA コントローラーで接続されたディスプレイ
- UEFI のレガシーモードへの切り替え

したがって、この設定で RHEL を使用またはインストールすることはできません。

この問題を回避するには、以下の手順に従います。

1. ブートローダーメニューにシステムを起動します。
2. カーネルコマンドラインに **nomodeset** オプションを追加します。

これにより、RHEL がブートし、グラフィカル出力が想定どおりに表示されますが、最大解像度は制限されます。

(BZ#1953926)

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と、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)

ARM でハードウェアアクセラレーションがサポートされない

組み込みグラフィックドライバーは、64 ビット ARM アーキテクチャー上のハードウェアアクセラレーションまたは Vulkan API に対応していません。

ARM でハードウェアアクセラレーションまたは Vulkan を有効にするには、プロプライエタリーの Nvidia ドライバーをインストールします。

(JIRA:RHELPLAN-57914)

10.16. 仮想化

PowerVM での IBMVFC デバイスのホットアンプラグに失敗する

PowerVM ハイパーバイザー上の RHEL 8 ゲストオペレーティングシステムで仮想マシン (VM) を使用する場合は、実行中の仮想マシンから IBM Power Virtual Fibre Channel (IBMVFC) デバイスを削除しようとすると失敗します。代わりに、**outstanding translation** エラーが表示されます。

この問題を回避するには、仮想マシンのシャットダウン時に IBMVFC デバイスを削除します。

(BZ#1959020)

ibmvfc ドライバーを使用すると、IBM POWER ホストがクラッシュする可能性がある

PowerVM 論理パーティション (LPAR) で RHEL 8 を実行すると、現在、**ibmvfc** ドライバーの問題により、さまざまなエラーが発生する可能性があります。結果として、ホストのカーネルは、以下のような特定の状況下でパニックになる可能性があります。

- Live Partition Mobility (LPM) 機能の使用
- ホストアダプターのリセット
- SCSI エラー処理機能 (SCSI EH) 機能の使用

(BZ#1961722)

IBM POWER Systems で perf kvm レコードを使用すると、仮想マシンがクラッシュする可能性があります。

IBM POWER ハードウェアのリトルエンディアンバリアントで RHEL 8 ホストを使用する場合は、**perf kvm record** コマンドを使用して KVM 仮想マシンのイベントサンプルを収集すると、仮想マシンが応答しなくなることがあります。この状況は、以下の場合に発生します。

- **perf** ユーティリティーは権限のないユーザーによって使用され、-p オプションは仮想マシンを識別するために使用されます (**perf kvm record -e trace_cycles -p 12345**)。
- 仮想マシンが **virsh** シェルを使用して起動している。

この問題を回避するには、**perf kvm** ユーティリティーに -i オプションを指定して、**virsh** シェルを使用して作成した仮想マシンを監視します。以下に例を示します。

```
# perf kvm record -e trace_imc/trace_cycles/ -p <guest pid> -i
```

-i オプションを使用する場合、子タスクはカウンターを継承しないため、スレッドは監視されないことに注意してください。

(BZ#1924016)

virtio-blk を使用して仮想マシンに LUN デバイスを割り当てる機能しません。

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)

iommu_platform=on が IBM POWER で起動に失敗する

RHEL 8 は現在、IBM POWER システムの仮想マシン用の **iommu_platform=on** パラメーターに対応していません。これにより、IBM POWER ハードウェアでこのパラメーターを使用して仮想マシンを起動すると、仮想マシンがシステムの起動プロセス時に応答しなくなります。

(BZ#1910848)

特定の CPU モデルの使用時に Hyper-V を有効化した Windows Server 2016 仮想マシンが起動に失敗する

現在、Windows Server 2016 をゲストオペレーティングシステムとして使用し、Hyper-V ロールが有効になっていて、以下の CPU モデルのいずれかを使用する仮想マシンを起動できません。

- EPYC-IBPB
- EPYC

この問題を回避するには、EPYC-v3 CPU モデルを使用するか、仮想マシンの `xsaves` CPU フラグを手動で有効にします。

(BZ#1942888)

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)

virt-customize を使用すると、**guestfs-firstboot** が失敗することがあります。

virt-customize ユーティリティーを使用して仮想マシン (VM) ディスクイメージを変更すると、SELinux パーミッションが正しくないために **guestfs-firstboot** サービスが失敗します。これにより、ユーザーの作成やシステム登録の失敗など、仮想マシンの起動時にさまざまな問題が発生します。

この問題を回避するには、**--selinux-relabel** オプションを **virt-customize** コマンドに追加します。

(BZ#1554735)

macvtap 仮想ネットワークから正引きインターフェイスを削除すると、このネットワークの接続数がすべてリセットされます。

現在、複数のフォワードインターフェイスを持つ **macvtap** 仮想ネットワークからフォワードインターフェイスを削除すると、ネットワークの他のフォワードインターフェイスの接続ステータスもリセットされます。したがって、ライブネットワーク XML の接続情報が正しくありません。ただし、これは仮想ネットワークの機能に影響を与えるわけではないことに注意してください。この問題を回避するには、ホストで **libvirtd** サービスを再起動します。

(BZ#1332758)

SLOF が指定された仮想マシンは netcat インターフェイスでの起動に失敗する

netcat(**nc**) インターフェイスを使用して、現在 Slimline Open Firmware(SLOF) プロンプトで待機中の仮想マシンのコンソールにアクセスすると、ユーザー入力は無視され、仮想マシンが応答しないままとなります。この問題を回避するには、仮想マシンに接続する場合は **nc -C** オプションを使用するか、代わりに **telnet** インターフェイスを使用します。

(BZ#1974622)

RHEL 8 ゲストで特定の状況で **virtiofs** ディレクトリーのマウントに失敗する

現在、**virtiofs** 機能を使用して仮想マシンにホストディレクトリーを提供する場合は、仮想マシンが RHEL 8.4 カーネルを使用しているにもかかわらず、RHEL 8.5 **selinux-policy** パッケージを使用している場合は、Operation not supported エラーで、仮想マシンへのディレクトリーのマウントに失敗します。

この問題を回避するには、ゲストを再起動して、ゲストで利用可能な最新のカーネルで起動します。

(BZ#1995558)

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る

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)

AMD EPYC でホストパススルーモードを使用する際に、SMT CPU トポロジーが仮想マシンで検出されない

AMD EPYC ホストで行われた CPU ホストパススルーモードで仮想マシンを起動すると、**TOPOEXT** 機能フラグは存在しません。したがって、仮想マシンは、コアごとに複数のスレッドを持つ仮想 CPU トポロジーを検出できません。この問題を回避するには、ホストパススルーモードの代わりに EPYC CPU モデルを使用して仮想マシンを起動します。

(BZ#1740002)

10.17. クラウド環境の RHEL

VMware ホストの RHEL 8 仮想マシンで静的 IP を設定できませんでした。

現在、VMware ホストで RHEL 8 を仮想マシンのゲストオペレーティングシステムとして使用すると、DatasourceOVF 機能は正しく機能しません。これにより、**cloud-init** ユーティリティーを使用して、仮想マシンのネットワークを静的 IP に設定し、仮想マシンを再起動すると、仮想マシンのネットワークが DHCP に変更されます。

(BZ#1750862)

Azure および Hyper-V で **kdump** が起動しないことがある

Microsoft Azure または Hyper-V ハイパーバイザでホストされている RHEL 8 ゲストオペレーティングシステムでは、実行後通知が有効な場合に **kdump** カーネルの起動が失敗することがあります。

この問題を回避するには、**crash kexec post notifiers** を無効にします。

```
# echo N > /sys/module/kernel/parameters/crash_kexec_post_notifiers
```

(BZ#1865745)

複数のゲストディスクで Hyper-V 仮想マシンを起動する際に、SCSI ホストアドレスが変更することがある

現在、Hyper-V ハイパーバイザーで RHEL 8 仮想マシンを起動すると、場合によっては、**Host**, **Bus**, **Target**, **Lun** (HBTL) SCSI アドレスのホスト部分が変わることがあります。したがって、仮想マシンで HBTL SCSI 識別またはデバイスノードで設定した自動タスクは一貫して動作しません。これは、仮想マシンに複数のディスクがある場合、またはディスクに異なるサイズがある場合に発生します。

この問題を回避するには、以下のいずれかの方法でキックスタートファイルを変更します。

方法 1: SCSI デバイスに永続的な識別子を使用

たとえば、以下の powershell スクリプトを使用すると、特定のデバイス識別子を特定できます。

```
# Output what the /dev/disk/by-id/<value> for the specified hyper-v virtual disk.
# Takes a single parameter which is the virtual disk file.
# Note: kickstart syntax works with and without the /dev/ prefix.
param (
    [Parameter(Mandatory=$true)][string]$virtualdisk
)

$what = Get-VHD -Path $virtualdisk
$part = $what.DiskIdentifier.ToLower().split('-')

$p = $part[0]
$s0 = $p[6] + $p[7] + $p[4] + $p[5] + $p[2] + $p[3] + $p[0] + $p[1]

$p = $part[1]
$s1 = $p[2] + $p[3] + $p[0] + $p[1]

[string]::format("/dev/disk/by-id/wwn-0x{0}{1}{2}", $s0, $s1, $part[4])
```

このスクリプトは、ハイパー ホストで使用することができます。以下に例を示します。

```
PS C:\Users\Public\Documents\Hyper-V\Virtual hard disks> .\by-id.ps1 .\Testing_8\disk_3_8.vhdx
/dev/disk/by-id/wwn-0x60022480e00bc367d7fd902e8bf0d3b4
PS C:\Users\Public\Documents\Hyper-V\Virtual hard disks> .\by-id.ps1 .\Testing_8\disk_3_9.vhdx
/dev/disk/by-id/wwn-0x600224807270e09717645b1890f8a9a2
```

その後、以下のようにキックスタートファイルでディスクの値を使用できます。

```
part / --fstype=xfs --grow --asprimary --size=8192 --ondisk=/dev/disk/by-id/wwn-0x600224807270e09717645b1890f8a9a2
part /home --fstype="xfs" --grow --ondisk=/dev/disk/by-id/wwn-0x60022480e00bc367d7fd902e8bf0d3b4
```

これらの値は仮想ディスクごとに固有であるため、仮想マシンインスタンスごとに設定を行う必要があります。そのため、**%include** 構文を使用して、ディスク情報を別のファイルに配置すると便利です。

方法 2: デバイス選択をサイズで設定

サイズに基づいてディスク選択を設定するキックスタートファイルには、以下のような行を含める必要があります。

...

```

# Disk partitioning information is supplied in a file to kick start
%include /tmp/disks

...
# Partition information is created during install using the %pre section
%pre --interpreter /bin/bash --log /tmp/ks_pre.log

# Dump whole SCSI/IDE disks out sorted from smallest to largest ouputting
# just the name
disks=(`lsblk -n -o NAME -l -b -x SIZE -d -l 8,3` ) || exit 1

# We are assuming we have 3 disks which will be used
# and we will create some variables to represent
d0=${disks[0]}
d1=${disks[1]}
d2=${disks[2]}

echo "part /home --fstype="xfs" --ondisk=$d2 --grow" >> /tmp/disks
echo "part swap --fstype="swap" --ondisk=$d0 --size=4096" >> /tmp/disks
echo "part / --fstype="xfs" --ondisk=$d1 --grow" >> /tmp/disks
echo "part /boot --fstype="xfs" --ondisk=$d1 --size=1024" >> /tmp/disks

%end

```

(BZ#1906870)

10.18. 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portal 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ので、現時点で **redhat-support-tool** ユーティリティーは、このポリシーレベルでは機能しません。

この問題を回避するには、カスタマーportal API への接続中に **DEFAULT** 暗号化ポリシーを使用します。

(BZ#1802026)

10.19. コンテナー

fuse-overlayfs を持つルートレスコンテナーが削除されたファイルを認識しない

RHEL8.4 以前では、ルートレスイメージやコンテナーは **fuse-overlayfs** ファイルシステムを使用して作成または保存されていました。RHEL 8.5 以降でこのようなイメージやコンテナーを使用すると、カーネルが提供する **overlayfs** の実装を使用している権限のないユーザーが、RHEL 8.4 のコンテナーやイメージからファイルやディレクトリーを削除した場合に問題が発生する可能性があります。この問題は、root アカウントで作成されたコンテナーには該当しません。

例えば、コンテナーやイメージから削除するファイルやディレクトリーは、**fuse-overlayfs** ファイルシステムを使用する場合、ホワイトアウトフォーマットを使用して、そのようにマークされます。しかし、フォーマットの違いにより、カーネルの **overlayfs** の実装では、**fuse-overlayfs** で作成されたホワイ

トアウトフォーマットを認識できません。その結果、削除されたファイルやディレクトリーはそのまま表示されます。この問題は、root アカウントで作成されたコンテナーには該当しません。

この問題を回避するには、次のいずれかのオプションを使用します。

1. **podman unshare rm -rf \$HOME/.local/share/containers/*** コマンドを使用して、非特権ユーザーのコンテナーとイメージをすべて削除します。ユーザーが次に Podman を実行すると、**\$HOME/.local/share/containers** ディレクトリーが再作成されるので、コンテナーを再作成する必要があります。
2. 引き続き、ルートレスユーザーで **podman** コマンドを実行します。更新されたバージョンの **podman** がシステム上で初めて起動されると、**\$HOME/.local/share/containers** ディレクトリー内のすべてのファイルをスキャンし、**fuse-overlayfs** を使用するかどうかを検出します。Podman は、後でスキャンを再実行しないように、スキャンの結果を記録します。その結果、削除されたファイルは表示されません。

fuse-overlayfs がまだ必要かどうかを検出するのに必要な時間は、スキャンが必要なコンテナーやイメージ内のファイルやディレクトリーの数に依存します。

(JIRA:RHELPLAN-92741)

古いコンテナーアイメージ内で **systemd** を実行すると動作しない

古いコンテナーアイメージ (例:**centos:7**) で **systemd** を実行しても動作しません。

```
$ podman run --rm -ti centos:7 /usr/lib/systemd/systemd
Storing signatures
Failed to mount cgroup at /sys/fs/cgroup/systemd: Operation not permitted
[!!!!!!] Failed to mount API filesystems, freezing.
```

この問題を回避するには、以下のコマンドを使用します。

```
# mkdir /sys/fs/cgroup/systemd
# mount none -t cgroup -o none,name=systemd /sys/fs/cgroup/systemd
# podman run --runtime /usr/bin/crun --annotation=run.oci.systemd.force_cgroup_v1=/sys/fs/cgroup -
-rm -ti centos:7 /usr/lib/systemd/systemd
```

(JIRA:RHELPLAN-96940)

ベータ版 GPG キーで署名されたコンテナーアイメージがプルできない

現在、RHEL Beta のコンテナーアイメージをプルしようとすると、**podman** が **Error: Source image rejected: None of the signatures were accepted** のエラーメッセージと共に終了します。現在のビルトでは、RHEL ベータ版の GPG キーをデフォルトで信頼しないように設定されているため、イメージのプルに失敗します。

回避策としては、Red Hat Beta GPG キーがローカルシステムに保存されていることを確認し、**podman image trust set** コマンドで適切な beta 名前空間の既存の信頼範囲を更新します。

ベータ版の GPG キーがローカルに保存されていない場合は、以下のコマンドを実行することで、そのキーをプルすることができます。

```
sudo wget -O /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-beta
https://www.redhat.com/security/data/f21541eb.txt
```

Beta GPG キーを信頼済みとしてネームスペースに追加するには、次のいずれかのコマンドを使用します。

```
$ sudo podman image trust set -f /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-beta
registry.access.redhat.com/namespace
```

および

```
$ sudo podman image trust set -f /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-beta
registry.redhat.io/namespace
```

namespaceを **ubi9-beta** または **rhel9-beta** に置き換えてください。

([BZ#2020301](#))

第11章 国際化

11.1. 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、libhangul

11.2. 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。
- 多くの **glibc** ロケールが Unicode Common Locale Data Repository (CLDR) と同期されています。

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#1898541 、 BZ#1951020 、 BZ#1938239 、 BZ#1947044 、 BZ#1626633 、 BZ#1812286 、 BZ#1850664 、 BZ#1944494 、 BZ#1895460 、 BZ#1817505
NetworkManager	BZ#1912236 、 BZ#1899372 、 BZ#1942331 、 BZ#1934465 、 BZ#1548825 、 BZ#1920398
SLOF	BZ#1910848
accel-config	BZ#1843266
accountsservice	BZ#1812788
anaconda	BZ#1914955 、 BZ#1931069 、 BZ#1903786 、 BZ#1954408 、 BZ#1821192 、 BZ#1822880 、 BZ#1929105 、 BZ#1897657
ansible-collection-redhat-rhel_mgmt	BZ#1843859
apr	BZ#1819607
bpftrace	BZ#1944716
brltty	BZ#2008197
chrony	BZ#1939295 、 BZ#1895003
cloud-init	BZ#1957532 、 BZ#1750862
cmake	BZ#1957947
cockpit	BZ#1666722
container-tools-rhel8-module	BZ#2009153
containers-common	BZ#2020301 、 BZ#2019901
coreutils	BZ#2030661

コンポーネント	チケット
corosync-qdevice	BZ#1784200
crash	BZ#1906482
createrepo_c	BZ#1973588
crypto-policies	BZ#1960266 , BZ#1876846 , BZ#1933016 , BZ#1919155 , BZ#1660839
distribution	BZ#1953991 、 BZ#1657927
dotnet6.0	BZ#2022794
dracut	BZ#1929201
dwz	BZ#1948709
dyninst	BZ#1933893 , BZ#1957942
edk2	BZ#1741615 、 BZ#1935497
elfutils	BZ#1933890 , BZ#1957225
fence-agents	BZ#1775847
firewalld	BZ#1872702 , BZ#1492722 , BZ#1871860
freeradius	BZ#1954521 , BZ#1977572 , BZ#1723362 , BZ#1958979
gcc-toolset-11-gdb	BZ#1954332
gcc-toolset-11	BZ#1953094
gcc	BZ#1974402 , BZ#1946758 , BZ#1946782 , BZ#1927516 , BZ#1979715
gdb	BZ#1854784 、 BZ#1853140
glibc	BZ#1934155 、 BZ#1912670 、 BZ#1930302
gnome-shell-extensions	BZ#1717947
gnome-shell	BZ#1935261 , BZ#1651378
gnome-software	BZ#1668760

コンポーネント	チケット
gnutls	BZ#1965445 、 BZ#1956783 、 BZ#1628553
go-toolset	BZ#1938071
golang	BZ#1979100 , BZ#1972825
grafana-container	BZ#1971557
grafana-pcp	BZ#1921190
grafana	BZ#1921191
grub2	BZ#1583445
hwloc	BZ#1917560
ipa	BZ#1924707 , BZ#1664719 , BZ#1664718
ipmitool	BZ#1951480
kernel	BZ#1944639 , BZ#1907271 , BZ#1902543 , BZ#1959772 , BZ#1954363 , BZ#1924230 , BZ#1954024 , BZ#1837389 , BZ#1570255 , BZ#1938339 , BZ#1865745 , BZ#1836058 , BZ#1906870 , BZ#1934033 , BZ#1924016 , BZ#1942888 , BZ#1868526 , BZ#1812577 , BZ#1694705 , BZ#1910358 , BZ#1953926 , BZ#1730502 , BZ#1930576 , BZ#1609288 , BZ#1793389 , BZ#1654962 , BZ#1666538 , BZ#1602962 , BZ#1940674 , BZ#1920086 , BZ#1971506 , BZ#1605216 , BZ#1519039 , BZ#1627455 , BZ#1501618 , BZ#1633143 , BZ#1814836 , BZ#1696451 , BZ#1348508 , BZ#1839311 , BZ#1783396 , JIRA:RHELPLAN-57712 , BZ#1837187 , BZ#1904496 , BZ#1660337 , BZ#1905243 , BZ#1878207 , BZ#1665295 , BZ#1871863 , BZ#1569610 , BZ#1794513
kexec-tools	BZ#1922951 , BZ#1879558 , BZ#1854037 , BZ#1931266 , BZ#2004000
krb5	BZ#1956388 , BZ#1877991
libcomps	BZ#1960616
libgcrypt	BZ#1976137
libgnome-keyring	BZ#1607766
libguestfs	BZ#1554735
libmodulemd	BZ#1894573 , BZ#1984402

コンポーネント	チケット
librepo	BZ#1814383
libreswan	BZ#1958968, BZ#1934058 , BZ#1934859 , BZ#1989050
libselinux-python-2.8-module	BZ#1666328
libservicelog	BZ#1844430
libvirt	BZ#1664592, BZ#1332758 , BZ#1528684
linuxptp	BZ#1895005
llvm-toolset	BZ#1927937
lsvpd	BZ#1844428
lvm2	BZ#1899214 、 BZ#1496229 、 BZ#1768536
mariadb-connector-odbc	BZ#1944692
mariadb	BZ#1944653 、 BZ#1942330
mesa	BZ#1886147
modulemd-tools	BZ#1924850
mutt	BZ#1912614
net-snmp	BZ#1919714
nfs-utils	BZ#1868087, BZ#1592011
nginx	BZ#1945671
nispor	BZ#1848817
nodejs-16-container	BZ#2001020
nss_nis	BZ#1803161
nss	BZ#1817533 , BZ#1645153

コンポーネント	チケット
opal-prd	BZ#1921665
opencryptoki	BZ#1919223
opencv	BZ#1886310
openmpi	BZ#1866402
opensc	BZ#1947025
openscap	BZ#1959570, BZ#1953092 , BZ#1966612
openslp	BZ#1965649
openssl	BZ#1810911
osbuild-composer	BZ#1945238, BZ#1937854 , BZ#1915351 , BZ#1951964
oscap-anaconda-addon	BZ#1691305, BZ#1674001 , BZ#1843932 , BZ#1665082
pacemaker	BZ#1948620 、 BZ#1443666
papi	BZ#1908126
pcp-container	BZ#1974912
pcp	BZ#1922040, BZ#1879350 , BZ#1629455
pcs	BZ#1839637、 BZ#1872378 、 BZ#1909901 、 BZ#1885293 、 BZ#1290830 、 BZ#1619620 、 BZ#1847102 、 BZ#1851335
pg_repack	BZ#1967193
php	BZ#1944110
pki-core	BZ#1729215
podman	JIRA:RHELPLAN-77542、JIRA:RHELPLAN-77241、 BZ#1934480 、 JIRA:RHELPLAN-77238、JIRA:RHELPLAN-77489、JIRA:RHELPLAN-92741
postfix	BZ#1711885
powertop	BZ#1834722

コンポーネント	チケット
ppc64-diag	BZ#1779206
pykickstart	BZ#1637872
qatlib	BZ#1920237
qemu-kvm	BZ#1740002、BZ#1719687、BZ#1651994
quota	BZ#1945408
rear	BZ#1983013、BZ#1930662、BZ#1958247、BZ#1988493、BZ#1958222、BZ#1983003、BZ#1747468、BZ#1868421
redhat-release	BZ#1935177
redhat-support-tool	BZ#1802026
restore	BZ#1997366
rhel-system-roles	BZ#1960375、BZ#1866544、BZ#1961858、BZ#1958963、BZ#1938014、BZ#1954747、BZ#1854187、BZ#1757869、BZ#1990947、BZ#1952090、BZ#1994580、BZ#1967335、BZ#1966711、BZ#1962976、BZ#1938016、BZ#1986463、BZ#1970664、BZ#1970642、BZ#1848683、BZ#1938020、BZ#1938023、BZ#1957849、BZ#1959649、BZ#1939711、BZ#1943679、BZ#1882475、BZ#1876315、BZ#1894642、BZ#1989199、BZ#1893743
rpm	BZ#1938928、BZ#1688849
rsyslog	BZ#1891458、BZ#1932795、BZ#1679512、JIRA:RHELPLAN-10431
rt-tests	BZ#1954387
ruby	BZ#1938942
rust-toolset	BZ#1945805
samba	BZ#1944657、BZ#2009213、JIRA:RHELPLAN-13195、Jira:RHELEDOCS-16612
scap-security-guide	BZ#1857179、BZ#1946252、 BZ#1955373、BZ#1966577、BZ#1970137、BZ#1993056、BZ#1993197、BZ#1876483、BZ#1955183、BZ#1843913、BZ#1858866、BZ#1750755

コンポーネント	チケット
selinux-policy	BZ#1994096 、 BZ#1860443 、 BZ#1931848 、 BZ#1947841 、 BZ#1461914
socat	BZ#1947338
sos	BZ#1928679
spice	BZ#1849563
squid	BZ#1964384
sssd	BZ#1737489 , BZ#1879869 , BZ#1949149 , BZ#1627112 , BZ#1947671
systemtap	BZ#1933889 , BZ#1957944
tboot	BZ#1947839
tesseract	BZ#1826085
tss2	BZ#1822073
tuned	BZ#1951992
udftools	BZ#1882531
udica	BZ#1763210
usbguard	BZ#2000000
valgrind	BZ#1933891 , BZ#1957226
vdo	BZ#1949163
wayland	BZ#1673073
xfsprogs	BZ#1949743
xorg-x11-server	BZ#1698565

コンポーネント	チェック
その他	BZ#2005277 , BZ#1839151 , JIRA:RHELPLAN-89566, JIRA:RHELPLAN-92473, JIRA:RHELPLAN-96640, JIRA:RHELPLAN-97145, BZ#1935686, BZ#1986007, JIRA:RHELPLAN-75166, JIRA:RHELPLAN-76515, JIRA:RHELPLAN-57941, JIRA:RHELPLAN-85064, JIRA:RHELPLAN-87877, JIRA:RHELPLAN-75164, BZ#2011448 , JIRA:RHELPLAN-99040, JIRA:RHELPLAN-99049, JIRA:RHELPLAN-99043, JIRA:RHELPLAN-99044, JIRA:RHELPLAN-99148, JIRA:RHELPLAN-61867, BZ#2013853 , BZ#1957316 , JIRA:RHELPLAN-79074, BZ#2019318 , JIRA:RHELPLAN-59528, JIRA:RHELPLAN-95056, BZ#1971061 , BZ#1959020, BZ#1897383 , BZ#1961722, BZ#1777138, BZ#1640697, BZ#1659609, BZ#1687900 , BZ#1697896, BZ#1757877, BZ#1741436, JIRA:RHELPLAN-59111, JIRA:RHELPLAN-27987, JIRA:RHELPLAN-28940, JIRA:RHELPLAN-34199, JIRA:RHELPLAN-57914, JIRA:RHELPLAN-96940, BZ#1987087, BZ#1974622, BZ#1995558, BZ#2028361, BZ#1690207, JIRA:RHELPLAN-1212, BZ#1559616, BZ#1889737 , BZ#1812552 , JIRA:RHELPLAN-14047, BZ#1769727 , JIRA:RHELPLAN-27394, JIRA:RHELPLAN-27737, BZ#1906489 , JIRA:RHELPLAN-58596, BZ#1642765, JIRA:RHELPLAN-10304, BZ#1646541, BZ#1647725, BZ#1932222 , BZ#1686057 , BZ#1748980 , BZ#1958250 , JIRA:RHELPLAN-71200, BZ#1827628, JIRA:RHELPLAN-45858, BZ#1871025 , BZ#1871953 , BZ#1874892, BZ#1893767 , BZ#1916296, JIRA:RHELPLAN-100400, BZ#1926114 , BZ#1904251, BZ#2011208 , JIRA:RHELPLAN-59825, BZ#1920624 , JIRA:RHELPLAN-70700, BZ#1929173 , BZ#2006665 , JIRA:RHELPLAN-98983, BZ#2013335, BZ#2019786, BZ#2009113, BZ#2038929

付録B 改訂履歴

0.3-6

2025年11月26日水曜日、Valentina Adhirova (vaahiro@redhat.com)

- 拡張機能 [BZ#1922312](#) (ファイルシステムとストレージ) を移動しました。
- 拡張機能 [BZ#1945408](#) (ファイルシステムとストレージ) を移動しました。

0.3-5

2025年1月30日木曜日、Gabriela Fialová (gfialova@redhat.com)

- 既知の問題 [JIRA:RHELODCS-19603](#) (IdM SSSD) を追加しました。

0.3-4

2024年12月4日水曜日、Gabriela Fialová (gfialova@redhat.com)

- カスタマーポータルのラボセクションを更新しました
- インストールセクションを更新しました

0.3-3

2024年6月10日水曜日、Gabriela Fialová (gfialova@redhat.com)

- 機能拡張 [BZ#2083077](#) (ファイルシステムとストレージ) を追加しました。

0.3-2

2024年5月9日(木) Brian Angelica (bangelic@redhat.com)

- [BZ#1690207](#) のテクノロジープレビューを更新しました。

0.3-1

2024年5月9日(木)、Gabriela Fialová (gfialova@redhat.com)

- 既知の問題 [BZ#1730502](#) (ストレージ) を更新。

0.3-0

2024年2月7日(水)、Lucie Vařáková (lvarakova@redhat.com)

- 非推奨の機能 [JIRA:RHELODCS-17641](#) (Networking) を追加。

0.2-9

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。

0.2-8

2022年11月15日(火) Gabriela Fialová (gfialova@redhat.com)

- 壊れたリンクを修正しました。

0.2-7

2023年10月13日(金) Gabriela Fialová (gfialova@redhat.com)

- テクノロジープレビュー [JIRA:RHEODOCS-16861](#)(コンテナー)を追加しました。

0.2-6

2023年9月8日、 Marc Muehlfeld (mmuehlfeld@redhat.com)

- 非推奨機能のリリースノート [JIRA:RHEODOCS-16612](#)(Samba)を追加しました。

0.2-5

2023年6月7日(水) Lucie Vařáková (lvarakova@redhat.com)

- 新機能 [JIRA:RHELPLAN-159143](#)(アイデンティティ管理)を追加。

0.2-4

2023年4月27日(木) Gabriela Fialová (gfialova@redhat.com)

- 既知の問題 [JIRA:RHELPLAN-155168](#)(アイデンティティ管理)を追加

0.2-3

2023年4月13日(木) Gabriela Fialová (gfialova@redhat.com)

- DFとKIの2つの壊れたリンクを修正しました。

0.2-2

2023年1月30日月曜日、 Lucie Vařáková (lvarakova@redhat.com)

- 新機能 [BZ#2164986](#)(ネットワーキング)を追加。

0.2-1

2022年12月8日木曜日、 Marc Muehlfeld (mmuehlfeld@redhat.com)

- 既知の問題 [BZ#2132754](#)(ネットワーキング)を追加。

0.2-0

7月29日金曜日、 Lucie Vařáková (lvarakova@redhat.com)

- バグ修正 [BZ#1661674](#)(ファイルシステムとストレージ)を追加。

0.1-9

6月9日(木)、 Lucie Vařáková (lvarakova@redhat.com)

- 既知の問題 [BZ#2059262](#)(ファイルシステムおよびストレージ)を追加。
- バグ修正 [BZ#1940468](#)(シェルとコマンドラインツール)を追加。

0.1-8

4月29日(金)、 Lenka Špačková (lspackova@redhat.com)

- 非推奨になった機能の概要を更新。

- [BZ#1605216](#) のタイプミスを修正。
- 無効になっていたリンクを修正。

0.1-7

2022 年 4 月 27 日 (水) Gabriela Fialová (gfialova@redhat.com)

- [BZ#2050140](#) を既知の問題 (インストーラー) に追加しました。

0.1-6

2022 年 4 月 1 日 (金) Gabriela Fialová (gfialova@redhat.com)

- 追加された [JIRA:RHELPLAN-57712](#) は、テクノロジープレビューから拡張機能 (ネットワーキング) に移動しました。

0.1-5

2022 年 3 月 22 日 (火) Lucie Maňásková (lmanasko@redhat.com)

- 既知の問題 (デスクトップ) に [JIRA:RHELPLAN-100230](#) を追加しました。

0.1-5

2022 年 3 月 21 日 (月) Jaroslav Klech (jklech@redhat.com)

- 既知の問題 (カーネル) から BZ#1666538 を削除。

0.1-4

2022 年 3 月 17 日 (木) Jaroslav Klech (jklech@redhat.com)

- 既知の問題 (カーネル) から BZ#1947839 を削除。

0.1-3

2022 年 2 月 15 日 (火) Lucie Maňásková (lmanasko@redhat.com)

- バグ修正 [BZ#1934033](#) (クラウド環境の RHEL) および [BZ#2019901](#) (コンテナー) を追加しました。
- 非推奨の機能 [BZ#1997366](#) および [BZ#2038929](#) (シェルおよびコマンドラインツール)、[BZ#2009113](#) (ネットワーキング)、[BZ#1871863](#) (カーネル)、[BZ#1794513](#) (ファイルシステムおよびストレージ)、および [BZ#1664592](#) (仮想化) が追加されました。
- その他の若干の更新

0.1-2

2022 年 2 月 4 日 (金) Jaroslav Klech (jklech@redhat.com)

- 非推奨のパッケージのリストを更新しました。
- 非推奨の機能 [BZ#1871863](#) (カーネル) が追加されました。
- 非推奨の機能 [BZ#2038929](#) (シェルおよびコマンドラインツール) が追加されました。

0.1-1

2022年2月3日(木) Gabriela Fialová (gfialova@redhat.com)

- 非推奨機能 [BZ#2009113](#) (ネットワーク) を追加。
- 非推奨の機能 [BZ#1794513](#) (ファイルシステムおよびストレージ) が追加されました。

0.1-0

2022年2月1日(火) Lucie Maňásková (lmanasko@redhat.com)

- 非推奨の機能 [BZ#1997366](#) (シェルおよびコマンドラインツール) が追加されました。
- [BZ#1664592](#) を既知の問題から非推奨の機能(仮想化)に変更しました。

0.0-9

2022年1月27日(木) Lucie Maňásková (lmanasko@redhat.com)

- 既知の問題(シェルおよびコマンドラインツール)に [BZ#2030661](#) が追加されました。
- 非推奨デバイスのリストを更新しました。

0.0-8

2022年1月17日(月) Lucie Maňásková (lmanasko@redhat.com)

- 新しい機能(コンテナー)に [BZ#2009153](#) が追加されました。
- 既知の問題(インストーラーおよびイメージの作成)に [BZ#2028361](#) を追加しました。
- 非推奨デバイスのリストを更新しました。

0.0-7

2021年12月21日(火) 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。
- テクノロジープレビューから、完全にサポートされている機能(コンテナー)に、**ubi8/nodejs-16** および **ubi8/nodejs-16-minimal** コンテナーアイメージの [BZ#2001020](#) を移動しました。

0.0-6

2021年12月16日(木) Lenka Špačková (lspackova@redhat.com)

- テクノロジープレビューから、完全に対応した機能(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)に、**nodejs:16** モジュールストリームの [BZ#1953991](#) を移行しました。

0.0-5

2021年12月10日(火) Lucie Maňásková (lmanasko@redhat.com)

- 非推奨の機能 [BZ#1827628](#) (ファイルシステムおよびストレージ) が追加されました。
- 既知の問題(カーネル)への [BZ#1654962](#) を追加しました。
- その他の若干の更新

0.0-4

2021年11月22日(月) Lucie Maňásková (lmanasko@redhat.com)

- 新機能 [BZ#1922951](#)(カーネル) が更新されました。
- 新しい機能 [BZ#1934480](#)(コンテナー) を追加しました。
- その他の若干の更新

0.0-3

2021年11月19日(金) Lucie Maňásková (lmanasko@redhat.com)

- バグ修正(カーネル)に [BZ#1959772](#) および [BZ#1954363](#) が追加されました。
- バグ修正(Identity Management)への [BZ#1977572](#) を追加しました。
- 新しい機能(コンパイラおよび開発ツール)に [BZ#2022794](#) が追加されました。
- 外部カーネルパラメーターの変更に関する情報を追加しました。

0.0-2

2021年11月17日(水) Prerana Sharma (presharm@redhat.com)

- 付録 A で [BZ#1944716](#)(bpftrace) を追加しました。

0.0-1

2021年11月10日(水) Lucie Maňásková (lmanasko@redhat.com)

- Red Hat Enterprise Linux 8.5 リリースノートのリリース

0.0-0

2021年10月6日(水) Lucie Maňásková (lmanasko@redhat.com)

- Red Hat Enterprise Linux 8.5 Beta リリースノートのリリース