



Red Hat Enterprise Linux 8

8.4 リリースノート

Red Hat Enterprise Linux 8.4 リリースノート

Red Hat Enterprise Linux 8 8.4 リリースノート

Red Hat Enterprise Linux 8.4 リリースノート

法律上の通知

Copyright © 2024 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

本リリースノートでは、Red Hat Enterprise Linux 8.4 での改良点および実装された追加機能の概要、本リリースにおける既知の問題などを説明します。また、重要なバグ修正、テクニカルレビュー、非推奨の機能などの詳細も説明します。

目次

多様性を受け入れるオープンソースの強化	5
RED HAT ドキュメントへのフィードバック (英語のみ)	6
第1章 概要	7
1.1. RHEL 8.4 における主な変更点	7
1.2. インプレースアップグレードおよび OS 移行	9
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	14
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. セキュリティー	22
4.7. ネットワーク	27
4.8. カーネル	30
4.9. ファイルシステムおよびストレージ	38
4.10. 高可用性およびクラスター	41
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	42
4.12. コンパイラーおよび開発ツール	50
4.13. ID 管理	60
4.14. デスクトップ	65
4.15. グラフィックインフラストラクチャー	66
4.16. WEB コンソール	68
4.17. RED HAT ENTERPRISE LINUX システムロール	69
4.18. 仮想化	71
4.19. クラウド環境の RHEL	72
4.20. サポート性	72
4.21. コンテナ	74
第5章 外部カーネルパラメーターへの重要な変更	76
5.1. 新しいカーネルパラメーター	76
5.2. 新しい /PROC/SYS/USER パラメーター	77
5.3. 新しい /PROC/SYS/VM パラメーター	77
第6章 デバイスドライバー	79
6.1. 新しいドライバー	79
6.2. 更新されたドライバー	80
第7章 バグ修正	81
7.1. インストーラーおよびイメージの作成	81
7.2. ソフトウェア管理	82

7.3. シェルおよびコマンドラインツール	82
7.4. インフラストラクチャーサービス	83
7.5. セキュリティー	83
7.6. ネットワーク	86
7.7. カーネル	88
7.8. ファイルシステムおよびストレージ	90
7.9. 高可用性およびクラスター	90
7.10. 動的プログラミング言語、WEB サーバー、およびデータベースサーバー	90
7.11. コンパイラーおよび開発ツール	90
7.12. ID 管理	92
7.13. グラフィックインフラストラクチャー	93
7.14. RED HAT ENTERPRISE LINUX システムロール	93
7.15. 仮想化	94
7.16. クラウド環境の RHEL	94
7.17. コンテナ	95
第8章 テクノロジープレビュー	96
8.1. インストーラーおよびイメージの作成	96
8.2. ネットワーク	96
8.3. カーネル	99
8.4. ファイルシステムおよびストレージ	100
8.5. 高可用性およびクラスター	103
8.6. ID 管理	104
8.7. デスクトップ	105
8.8. グラフィックインフラストラクチャー	106
8.9. RED HAT ENTERPRISE LINUX システムロール	106
8.10. 仮想化	107
8.11. コンテナ	109
第9章 非推奨になった機能	110
9.1. インストーラーおよびイメージの作成	110
9.2. ソフトウェア管理	111
9.3. シェルおよびコマンドラインツール	111
9.4. セキュリティー	112
9.5. ネットワーキング	113
9.6. カーネル	114
9.7. プラットフォームの有効化	114
9.8. ファイルシステムおよびストレージ	115
9.9. 高可用性およびクラスター	116
9.10. コンパイラーおよび開発ツール	116
9.11. IDENTITY MANAGEMENT	117
9.12. デスクトップ	119
9.13. グラフィックインフラストラクチャー	120
9.14. WEB コンソール	120
9.15. RED HAT ENTERPRISE LINUX システムロール	120
9.16. 仮想化	120
9.17. コンテナ	121
9.18. 非推奨のパッケージ	121
9.19. 非推奨のデバイス	123
第10章 既知の問題	127
10.1. インストーラーおよびイメージの作成	127
10.2. サブスクリプションの管理	131
10.3. インフラストラクチャーサービス	131

10.4. セキュリティー	131
10.5. ネットワーク	137
10.6. カーネル	138
10.7. ハードウェアの有効化	142
10.8. ファイルシステムおよびストレージ	143
10.9. 高可用性およびクラスター	144
10.10. 動的プログラミング言語、WEB サーバー、およびデータベースサーバー	145
10.11. コンパイラーおよび開発ツール	146
10.12. IDENTITY MANAGEMENT	147
10.13. デスクトップ	149
10.14. グラフィックインフラストラクチャー	149
10.15. 仮想化	151
10.16. クラウド環境の RHEL	154
10.17. サポート性	157
第11章 国際化	158
11.1. RED HAT ENTERPRISE LINUX 8 の多言語	158
11.2. RHEL 8 における国際化の主な変更点	158
付録A コンポーネント別のチケットリスト	160
付録B 改訂履歴	169

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[Red Hat CTO である Chris Wright のメッセージ](#) をご覧ください。

RED HAT ドキュメントへのフィードバック (英語のみ)

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

Jira からのフィードバック送信 (アカウントが必要)

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

第1章 概要

1.1. RHEL 8.4 における主な変更点

セキュリティー

Libreswan が提供する **IPsec VPN** が、IKEv2 の TCP カプセル化およびセキュリティーラベルに対応するようになりました。

scap-security-guide パッケージがバージョン 0.1.54 にリベースされ、**OpenSCAP** がバージョン 1.3.4 にリベースされました。これらの更新により、以下を含む大幅に改善が行われます。

- メモリー管理の改善
- RHEL8 ANSSI-BP-028 Minimal、Intermediary および Enhanced プロファイルが追加されました
- RHEL8 STIG プロファイルが DISA STIG v1r1 に更新されました。

fapolicyd フレームワークは、**整合性チェック** を提供し、RPM プラグインは YUM パッケージマネージャーまたは RPM Package Manager によって処理されるシステム更新を登録するようになりました。

rhel8-tang コンテナイメージは、OpenShift Container Platform (OCP) クラスターまたは別の仮想マシンで実行する Clevis クライアントの Tang-server 復号化機能を提供します。

詳細は、「[セキュリティー](#)」を参照してください。

ネットワーク

nmstate はホストのネットワーク API で、RHEL 8.4 で完全に対応しています。**nmstate** パッケージは、ライブラリーと **nmstatectl** コマンドラインユーティリティーを提供し、ホストのネットワーク設定を宣言型で管理できます。

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。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。MPLS サポートは、本リリースではテクノロジープレビューとして提供されています。

iproute2 ユーティリティーには、新しいトラフィック制御 (**tc**) アクションを 3 つ導入します。**mac_push**、**push_eth**、および **pop_eth** は、パケットの先頭に MPLS ラベルを追加し、イーサネットヘッダーの最初にイーサネットヘッダーを構築して、outer Ethernet ヘッダーをドロップします。

bareudp デバイスのサポートが、テクノロジープレビューとして **ip link** コマンドで利用できるようになりました。

本リリースで導入された機能および既存機能の変更に関する詳細は、「[ネットワーク](#)」を参照してください。

カーネル

kpatch-dnf パッケージは、RHEL システムをカーネルライブパッチ更新にサブスクリブするための DNF プラグインを提供します。このプラグインは、現在システムが使用するカーネルの自動サブスクリプションと、将来にカーネルのインストールが可能です。

プロアクティブな圧縮 は、割り当ての要求が実行される **前に**、メモリー圧縮機能を定期的を開始します。したがって、特定のメモリー割り当て要求のレイテンシーは短くなります。

コントロールグループ 技術にスラブメモリーコントローラーの新しい実装が RHEL 8 で利用可能になり

ました。slab メモリーコントローラーにより、スラブ使用率が改善され、ページレベルからオブジェクトレベルへメモリーアカウンティングの移動を有効にします。その結果、カーネルメモリーフットプリントの合計とメモリーの断片化の影響が大幅に低下することができます。

時間名前空間機能は、RHEL 8.4 で利用できます。この機能は、Linux コンテナ内の日時の変更に適しています。チェックポイントからの復元後のコンテナ内クロックの調整も可能になりました。

RHEL 8 は、第 8 世代および 9 世代 Intel Core プロセッサで設定された Error Detection and Correction (EDAC) カーネルモジュールをサポートします。

本リリースで導入された機能および既存機能の変更に関する詳細は、「[カーネル](#)」を参照してください。

高可用性およびクラスター

状態データを維持する永続的な Pacemaker リソースエージェントは、次のモニターの間隔を待たずに、失敗を非同期的に検出し、Pacemaker に即座に挿入することができます。また、永続的なリソースエージェントは、状態データをメンテナンスし、起動、停止、監視などのクラスターアクションの状態のオーバーヘッドを下げ、アクションごとに個別に呼び出しされないため、状態の高いオーバーヘッドのサービスに対してクラスターの応答時間を短縮することもできます。

永続的な Pacemaker リソースエージェントの作成方法は、[Creating a Persistent \(Daemonized\) Pacemaker Resource Agent](#) の記事を参照してください。

動的プログラミング言語、Web サーバー、およびデータベースサーバー

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

- Python 3.9
- SWIG 4.0
- subversion 1.14
- Redis 6
- PostgreSQL 13
- MariaDB 10.5

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

コンパイラーおよび開発ツール

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 10
- LLVM Toolset 11.0.0
- Rust Toolset 1.49.0
- Go Toolset 1.15.7

詳細は、「[コンパイラーおよび開発ツール](#)」を参照してください。

OpenJDK 11 が利用可能に

新しいバージョンの Open Java Development Kit (OpenJDK) 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。

Identity Management

RHEL 8.4 は、Identity Management (IdM) のロールベースアクセス制御 (RBAC)、IdM サーバーのバックアップおよび復元用の Ansible ロール、およびロケーション管理の Ansible モジュールを自動管理するための Ansible モジュールを提供します。

詳細は、「[ID 管理](#)」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 8.4 のアップグレード。
- 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。

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

RHEL 8.4 のリリースに伴い、Red Hat Subscription Manager (RHSM) を使用している場合は、必要な追加のデータファイルが cloud.redhat.com から自動的にダウンロードされます。アップグレードを行わずに古い必要なデータファイルがダウンロードされなくなりました。

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** ユーティリティを使用することに注意してください。サポートされない変換の詳細は、[How to convert from CentOS Linux 6 or Oracle Linux 6 to RHEL 6](#) 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内のセクションにあるツールセットで、<https://access.redhat.com/labs/> から入手できます。Red Hat Customer Portal Labs のアプリケーションは、パフォーマンスの向上、問題の迅速なトラブルシューティング、セキュリティ問題の特

定、複雑なアプリケーションの迅速なデプロイメントおよび設定に役立ちます。最も一般的なアプリケーションには、以下のものがあります。

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

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。
- **RHEL 7 と RHEL 8 の主な相違点** は、[RHEL 8 の導入における検討事項](#) を参照してください。
- **RHEL 7 から RHEL 8 へのインプレースアップグレード** を実行する方法は、[RHEL 7 から RHEL 8 へのアップグレード](#) を参照してください。
- すべての RHEL サブスクリプションで、既知の技術問題の特定、検証、および解決をプロアクティブに行う **Red Hat Insights** サービスが利用できるようになりました。Red Hat Insights クライアントをインストールし、システムをサービスに登録する方法は、[Red Hat Insights を使い始める](#) ページを参照してください。

第2章 アーキテクチャー

Red Hat Enterprise Linux 8.4 ではカーネルバージョン 4.18.0-305 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、カスタマーポータル[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 リポジトリおよび AppStream リポジトリが含まれ、リポジトリを追加しなくてもインストールを完了できる完全インストールイメージです。



注記

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

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

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

3.2. リポジトリ

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

- BaseOS
- AppStream

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

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

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

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

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

3.3. アプリケーションストリーム

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

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

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

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

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

3.4. YUM/DNF を使用したパッケージ管理

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

詳細は、以下のドキュメントを参照してください。

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

第4章 新機能

ここでは、Red Hat Enterprise Linux 8.4 に追加された新機能および主要な機能強化を説明します。

4.1. インストーラーおよびイメージの作成

Anaconda が、元のブートデバイス NVRAM 変数リストを新しい値に置き換えます。

以前は、NVRAM からの起動により、ブートデバイスリストのエントリーが正しくない値を持つために、起動システムが失敗する可能性があります。

今回の更新でこの問題が修正されていますが、ブートデバイス NVRAM 変数の更新時に、以前のデバイスのリストは消去されるようになりました。

(BZ#1854307)

IBM Z への KVM 仮想マシンのグラフィカルインストールが利用できるようになりました。

IBM Z ハードウェアで KVM ハイパーバイザーを使用する場合は、仮想マシンの作成時にグラフィカルインストールを使用できるようになりました。

ユーザーが KVM でインストールを実行し、QEMU が **virtio-gpu** ドライバーを提供すると、インストーラーがグラフィカルコンソールを自動的に起動するようになりました。ユーザーは、仮想マシンのカーネルコマンドラインに **inst.text** または **inst.vnc** ブートパラメーターを追加して、テキストまたは VNC モードに切り替えることができます。

(BZ#1609325)

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 **%s** 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.%s** 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。

\$1 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 **\$2** を使用してください。

(BZ#1897657)

4.2. RHEL FOR EDGE

RHEL for Edge イメージタイプのカスタマイズとしてのカーネル名を指定するサポート

OSTree の作成が **RHEL for Edge** イメージにコミットする場合は、1度に1つのカーネルパッケージのみをインストールでき、コミットの作成は **rpm-ostree** で失敗します。これにより、RHEL for Edge が、リアルタイムカーネル (**kernel-rt**) の代替カーネルを追加できなくなります。今回の機能拡張によ

り、CLI を使用して RHEL for Edge イメージの Blueprint を作成する際に、**customizations.kernel.name** キーを設定して、イメージで使用するカーネルの名前を定義できるようになりました。カーネル名を指定しないと、イメージにはデフォルトのカーネルパッケージが含まれます。

(BZ#1960043)

4.3. ソフトウェア管理

DNF API で、新しい **fill_sack_from_repos_in_cache** 関数がサポートされるようになりました。

今回の更新で、新しい DNF API **fill_sack_from_repos_in_cache** 関数が導入されました。これにより、キャッシュされた **solv**、**solvx** ファイル、および **repomd.xml** ファイルからのみリポジトリを読み込むことができるようになりました。その結果、ユーザーが **dnf** キャッシュを管理する場合は、重複した情報 (**xml** および **solv**) がなく、**xml** を **solv** に処理せずにリソースを保存できます。

(BZ#1865803)

createrepo_c がモジュールメタデータをリポジトリに自動的に追加

以前は、RHEL8 パッケージで **createrepo_c** コマンドを実行して新しいリポジトリを作成すると、このリポジトリにモジュール **repodata** を追加できませんでした。そのため、リポジトリに関するさまざまな問題が生まれました。今回の更新により、**createrepo_c** は以下のようになります。

- モジュールメタデータのスキャン
- 見つかったモジュールの YAML ファイルを単一のモジュールドキュメント **modules.yaml** にマージします。
- このドキュメントをリポジトリに自動的に追加します。

その結果、リポジトリへのモジュールメタデータの配置が自動で、**modifyrepo_c** コマンドを使用して別のステップとして実行する必要がなくなりました。

(BZ#1795936)

DNF 内のシステム間でトランザクションをミラーリングする機能に対応

今回の更新で、ユーザーは DNF 内でトランザクションを保存し、再生できるようになりました。

- DNF 履歴のトランザクションを JSON ファイルに保存するには、**dnf history store** コマンドを実行します。
- トランザクションを同じマシンまたは別のマシン上で再生するには、**dnf history replay** コマンドを実行します。

Comps グループ操作の保存と再生に対応しました。モジュール操作はまだサポートされていないため、保存または再生されていません。

(BZ#1807446)

createrepo_c が 0.16.2 にリベースされました

createrepo_c パッケージがバージョン 0.16.2 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増えられています。

- **createrepo_c** のモジュールメタデータサポートが追加されました。

- さまざまなメモリーリークを修正しました。

(BZ#1894361)

protect_running_kernel 設定オプションが利用可能になりました。

今回の更新で、**dnf** コマンドおよび **microdnf** コマンドの **protect_running_kernel** 設定オプションが導入されました。このオプションは、実行中のカーネルバージョンに対応するパッケージを削除から保護するかどうかを制御します。これにより、ユーザーは実行中のカーネルの保護を無効にできるようになりました。

(BZ#1698145)

4.4. シェルおよびコマンドラインツール

OpenIPMI がバージョン 2.0.29 にリベースされました。

OpenIPMI パッケージがバージョン 2.0.29 にアップグレードされました。以前のバージョンに対する主な変更点は、以下のとおりです。

- メモリーリーク、変数バインディング、およびエラーメッセージの欠落が修正されました。
- **IPMB** のサポートが追加されました。
- **lanserv** での個別グループ拡張の登録に対するサポートが追加されました。

(BZ#1796588)

FreeIPMI がバージョン 1.6.6 にリベースされました。

freeipmi パッケージがバージョン 1.6.6 にアップグレードされました。以前のバージョンに対する主な変更点は、以下のとおりです。

- ソースコードのメモリーリークおよび誤字を修正しました。
- 以下の既知の問題に対する回避策が実装されました。
 - 予期しない完了コード。
 - Dell Poweredge FC830
 - **lan/rmcplusplus ipmb** による順序付けのないパケット。
- 新しい Dell、Intel、および Gigabyte デバイスのサポートが追加されました。
- システム情報およびイベントの解釈のサポートが追加されました。

(BZ#1861627)

opal-prd がバージョン 6.6.3 にリベースされました。

opal-prd パッケージがバージョン 6.6.3 にリベースされました。以下は、主な変更点です。

- **opal-prd** デーモンのオフラインワーカープロセスハンドルページを追加しました。
- **POWER9P** の **opal-gard** のバグを修正し、システムが **gard** レコードのチップターゲットを特定できるようになりました。

- **occ** コマンドの **wait_for_all_occ_init()** で誤検出を修正しました。
- **hw/phys-map** の **OCAPI_MEM BAR** の値が修正されました。
- **hdata/memory.c** の **Inconsistent MSAREA** の警告が修正されました。
- **occ** に協力する場合:
 - センサー値ゼロバグが修正されました。
 - GPU 検出コードが修正されました。
- **MPIPL** 起動時に **sysdump** の取得が省略されました。
- **Mihawk** プラットフォームで **IPMI** の二重解放が修正されました。
- **fsp/dump** で **non-MPIPL scenario** を更新しました。
- **hw/phb4** の場合:
 - AER regs の初期作成前に AER サポートが検証されました。
 - エラーレポートが有効化されました。
- **hdata** に新しい **smp-cable-connector** の VPD キーワードを追加しました。

(BZ#1844427)

opencryptoki がバージョン 3.15.1 にリベースされました。

opencryptoki パッケージがバージョン 3.15.1 にリベースされました。以下は、主な変更点です。

- **C_SetPin** の **segfault** を修正しました。
- **EVP_CipherUpdate** および **EVP_CipherFinal** の使用が修正されました。
- トークンリポジトリを **FIPS** 準拠の暗号化に移行するユーティリティーが追加されました。
- **pkcstok_migrate** ツールの場合:
 - Little Endian プラットフォームでの **NVTOK.DAT** 変換が修正されました。
 - Little Endian プラットフォームでのプライベートおよびパブリックトークンオブジェクトの変換が修正されました。
- 新しいデータ形式での公開トークンオブジェクトの保存が修正されました。
- **dh_pkcs_derive** のパラメーターチェックメカニズムが修正されました。
- ソフトトークンモデル名が修正されました。
- **mech_ec.c** ファイル、および **ICA**、**TPM**、および Soft トークンで、非推奨となった OpenSSL インターフェイスが置き換えられました。
- **sw_crypt.c** ファイルの非推奨の OpenSSL AES/3DES インターフェイスが置き換えられました。
- Soft トークンの ECC メカニズムのサポートが追加されました。

- Soft トークンに IBM 固有の SHA3 HMAC および SHA512/224/256 HMAC メカニズムを追加しました。
- CCA の **CKM_RSA_PKCS** を使用した鍵のラッピングのサポートが追加されました。
- EP11 暗号化スタックの場合:
 - **CKM_DES2_KEY_GEN** を認識するように **ep11_get_keytype** が修正されました。
 - **token_specific_rng** のエラートレースが修正されました。
 - HSM シミュレーションで特定の FW バージョンおよび API を有効にしました。
- **X9.63 KDF** の Endian バグが修正されました。
- **p11sak remove-key** コマンドを処理するエラーメッセージを追加しました。
- C++ でのコンパイルの問題が修正されました。
- **C_Get/SetOperationState** および digest コンテキストの問題が修正されました。
- **pkcscca** 移行が **usr/sb2** で失敗する不具合を修正しました。

(BZ#1847433)

powerpc-utils がバージョン 1.3.8 にリベースされました。

powerpc-utils パッケージがバージョン 1.3.8 にリベースされました。以下は、主な変更点です。

- **Perl** に依存しないコマンドは、**core** サブパッケージに移されました。
- Linux Hybrid Network Virtualization のサポートが追加されました。
- 安全なブートリストを更新しました。
- **vcputat** ユーティリティを追加しました。
- **lparstat** コマンドで **cpu-hotplug** のサポートが追加されました。
- **lparstat** コマンドでスケールされたメトリックを出力するためのスイッチが追加されました。
- 差分、スケールされた時間ベース、および **PURR/SPURR** 値を取得する **helper** 関数が追加されました。
- **ofpathname** ユーティティ:
 - **l2of_scsi()** のスピードを向上しました。
 - **udevadm** の場所を修正しました。
 - **l2od_ide()** および **l2of_scsi()** をサポートするパーティションを追加しました。
 - **SCSI/SATA** ホストのプラグイン ID のサポートが追加されました。
- サポート対象外のコネクタタイプの **segfault** 状態が修正されました。
- **SR_IOV** のハイブリッド仮想ネットワークへの移行をサポートするツールが追加されました。
- **format-overflow** の警告が修正されました。

- **lsdevinfo** ユーティリティーを使用して bash コマンドの置換に関する警告を修正しました。
- 起動時のボンディングインターフェイスのクリーンアップが修正されました。

(BZ#1853297)

新しいカーネルの **cmdline** オプションがネットワークデバイス名を生成するようになりました。

systemd-udev サービスの **net_id** ビルトインは、新しいカーネルの **cmdline** オプション **net.naming-scheme=SCHEME_VERSION** を取得します。**SCHEME_VERSION** の値に基づいて、ユーザーはネットワークデバイス名を生成するアルゴリズムのバージョンを選択できます。

たとえば、RHEL 8.4 の **net_id** ビルトイン機能を使用するには、**SCHEME_VERSION** の値を **rhel-8.4** に設定します。

同様に、**SCHEME_VERSION** の値を、必要な変更または修正が含まれるその他のマイナーリリースに設定できます。

(BZ#1827462)

4.5. インフラストラクチャーサービス

デフォルトの postfix-3.5.8 動作の違い

RHEL-8 の後方互換性を改善するために、**postfix-3.5.8** 更新の動作は、デフォルトのアップストリームの **postfix-3.5.8** の動作とは異なります。デフォルトのアップストリーム **postfix-3.5.8** 動作の場合は、以下のコマンドを実行します。

```
# postconf info_log_address_format=external
```

```
# postconf smtpd_discard_ehlo_keywords=
```

```
# postconf rhel_ipv6_normalize=yes
```

詳細は、**/usr/share/doc/postfix/README-RedHat.txt** ファイルを参照してください。互換性のない機能を使用しない場合や、RHEL-8 後方互換性の優先度が優先される場合は、手順は必要ありません。

(BZ#1688389)

BIND がバージョン 9.11.26 にリベースされました。

bind パッケージがバージョン 9.11.26 に更新されました。以下は、主な変更点です。

- デフォルトの EDNS バッファサイズを 4096 から 1232 バイトに変更しました。この変更により、ネットワーク内の断片化されたパケットが失われなくなります。
- **max-recursion-queries** のデフォルト値が 75 から 100 に増加しました。CVE-2020-8616 に関連します。
- **named** の **lib/dns/rbtdb.c** ファイルでデッドノードの問題が修正されました。
- **lib/dns/rbtdb.c** ファイルで再利用されたデッドノードをクリーンアップする際に、**named** サービスのクラッシュの問題が修正されました。
- **named** サービスで複数のフォワーダーを設定した場合の問題が修正されました。

- 親に DS レコードのない不正な署名ゾーンを割り当てる **named** サービスの問題を修正しました。
- **UDP** での欠落している **DNS cookie response** を修正しました。

([BZ#1882040](#))

unbound 設定がロギング出力が強化されるようになりました。

今回の機能拡張により、**unbound** されていない設定に以下の 3 つのオプションが追加されました。

- **log-servfail** は、**SERVFAIL** エラーコードのクライアントに対する理由を説明するログ行を有効にします。
- **log-local-actions** は、すべてのローカルゾーンアクションのロギングを有効にします。
- **log-tag-queryreply** は、ログファイルのログクエリーとログリプライのタグ付けを有効にします。

([BZ#1850460](#))

複数の脆弱性が **ghostscript-9.27** で修正されました

- **ghostscript-9.27** リリースには、以下の脆弱性のセキュリティー修正が含まれます。
 - [CVE-2020-14373](#)
 - [CVE-2020-16287](#)
 - [CVE-2020-16288](#)
 - [CVE-2020-16289](#)
 - [CVE-2020-16290](#)
 - [CVE-2020-16291](#)
 - [CVE-2020-16292](#)
 - [CVE-2020-16293](#)
 - [CVE-2020-16294](#)
 - [CVE-2020-16295](#)
 - [CVE-2020-16296](#)
 - [CVE-2020-16297](#)
 - [CVE-2020-16298](#)
 - [CVE-2020-16299](#)
 - [CVE-2020-16300](#)
 - [CVE-2020-16301](#)
 - [CVE-2020-16302](#)

- [CVE-2020-16303](#)
- [CVE-2020-16304](#)
- [CVE-2020-16305](#)
- [CVE-2020-16306](#)
- [CVE-2020-16307](#)
- [CVE-2020-16308](#)
- [CVE-2020-16309](#)
- [CVE-2020-16310](#)
- [CVE-2020-17538](#)

([BZ#1874523](#))

tuned がバージョン 2.15-1 にリベースされました。

以下は、主な変更点です。

- Linux サービス制御用の **service** プラグインが追加されました。
- **scheduler** プラグインが改善されました。

([BZ#1874052](#))

DNSTAP が受信する詳細なクエリーを記録するようになりました。

DNSTAP は、受信名クエリーの詳細を監視およびログする高度な方法を提供します。また、**named** サービスから送信された回答も記録します。named サービスの従来のクエリーロギングは、**named** サービスのパフォーマンスに悪影響を与えます。

その結果、DNSTAP は、パフォーマンス低下に影響を与えずに詳細な受信クエリーの継続的なログを実行する方法を提供します。新しい **dnstap-read** ユーティリティを使用すると、別のシステムで実行しているクエリーを分析できます。

([BZ#1854148](#))

SpamAssassin がバージョン 3.4.4 にリベースされました。

SpamAssassin パッケージがバージョン 3.4.4 にアップグレードされました。以下は、主な変更点です。

- **OLEVBMacro** プラグインが追加されました。
- 新しい関数 **check_rbl_ns**、**check_rbl_rcvd**、**check_hashbl_bodyre**、および **check_hashbl_uris** が追加されました。

([BZ#1822388](#))

キーアルゴリズムは **OMAPI** シェルを使用して変更できます。

今回の機能強化により、ユーザーは鍵アルゴリズムを変更できるようになりました。**HMAC-MD5** としてハードコードされた鍵アルゴリズムは、安全とはみなされません。これにより、**omshell** コマンドを使用して鍵アルゴリズムを変更できます。

(BZ#1883999)

Sendmail が TLSFallbackto Professional 設定に対応

今回の機能強化により、発信 TLS 接続が失敗すると、sendmail クライアントはプレーンテキストにフォールバックするようになりました。これにより、相互の TLS 互換性の問題が解消されます。Red Hat は、デフォルトで **TLSFallbacktoSheet** オプションを無効にした sendmail を提供します。

(BZ#1868041)

tcpdump で RDMA 対応デバイスの表示が可能に

今回の機能拡張により、**tcpdump** を使用した RDMA トラフィックの取得がサポートされるようになりました。これにより、**tcpdump** ツールを使用して、オフロードされた RDMA トラフィックをキャプチャーおよび分析できます。これにより、**tcpdump** を使用して RDMA 対応デバイスを表示し、RoCE および VMA トラフィックを取得し、そのコンテンツを分析できます。

(BZ#1743650)

4.6. セキュリティー

Libreswan を 4.3 にリベース

libreswan パッケージがバージョン 4.3 にアップグレードされました。以前のバージョンに対する主な変更点は、以下のとおりです。

- TCP サポート上の IKE および ESP (RFC 8229)
- IKEv2 ラベル付き IPsec サポート
- IKEv2 leftikeport/rightikeport のサポート
- 中間エクスチェンジの実験的サポート
- 負荷分散における拡張リダイレクトサポート
- 相互運用性を強化するために、デフォルトの IKE ライフタイムが 1h から 8h に変更
- **ipsec.secrets** ファイルの **RSA** セクションがなくなりました。
- Windows 10 の再起動を修正しました。
- ECDSA 認証用の証明書送信を修正しました。
- MOBIKE および NAT-T の修正

(BZ#1891128)

IPsec VPN が TCP トランスポートに対応

libreswan パッケージの更新では、RFC 8229 で説明されているように、TCP カプセル化の IPsec ベースの VPN サポートが追加されました。この追加により、ESP (Encapsulating Security Payload) および UDP を使用してトラフィックを防ぐネットワークで IPsec VPN を確立できます。その結果、管理者は、フォールバックまたはメインの VPN トランスポートプロトコルとして TCP を使用するように VPN サーバーおよびクライアントを設定できます。

(BZ#1372050)

Libreswan が、ラベル付き IPsec の IKEv2 に対応

Libreswan Internet Key Exchange (IKE) 実装には、IPsec のセキュリティーラベルの Internet Key Exchange version 2 (IKEv2) に対応するようになりました。今回の更新で、IKEv1 でセキュリティーラベルを使用するシステムが IKEv2 にアップグレードできるようになりました。

(BZ#1025061)

libpwquality が 1.4.4 にリベースされました。

libpwquality パッケージがバージョン 1.4.4 にリベースされました。本リリースには、バグ修正および機能強化が複数追加されました。たとえば、以下の設定オプションが **pwquality.conf** ファイルに追加されました。

- **retry**
- **enforce_for_root**
- **local_users_only**

(BZ#1537240)

p11-kit が 0.23.19 にリベースされました。

p11-kit パッケージが、バージョン 0.23.14 から 0.23.19 にアップグレードされました。新しいバージョンでは複数のバグが修正され、以下のようなさまざまな機能強化が提供されています。

- CVE-2020-29361、CVE-2020-29362、CVE-2020-29363 セキュリティー問題の修正
- **p11-kit** が **meson** ビルドシステムによるビルドをサポートするようになりました。

(BZ#1887853)

pyOpenSSL が 19.0.0 にリベースされました。

pyOpenSSL パッケージが、アップストリームバージョン 19.0.0 にリベースされました。このバージョンでは、主なバグ修正および機能強化が数多く追加されました。

- **openssl** バージョン 1.1.1 での TLS 1.3 サポートが改善されました。
- **X509Store.add_cert** で重複した証明書を追加しようとしてもエラーが発生することがなくなりました。
- コンポーネントに NUL バイトを含む X509 証明書の処理が改善されました。

(BZ#1629914)

SCAP セキュリティーガイドが 0.1.54 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 0.1.54 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。

- **Operating System Protection Profile (OSPP)** は、Red Hat Enterprise Linux 8.4 の Protection Profile for General Purpose Operating Systems に従って更新されました。
- フランス語の ANSSI(National Security Agency) の ANSSI BP-028 の推奨事項に基づくプロファイルの **ANSSI** ファミリーが導入されました。コンテンツには、最小、中間、および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。

- セキュリティー技術実装ガイド (STIG) セキュリティー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 V1R1 からルールが実装されました。

(BZ#1889344)

OpenSCAP が 1.3.4 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 1.3.4 にリベースされました。主な修正と機能強化は、以下のとおりです。

- 大量のファイルを持つシステムのメモリー不足を引き起こしていた特定のメモリーの問題が修正されました。
- OpenSCAP が、GPFS をリモートファイルシステムとして処理するようになりました。
- 定義間の循環依存関係のある OVAL の適切な処理。
- **yamlfilecontent** を向上: **yaml-filter** が更新され、スキーマおよびプローブを拡張し、マップ内の値のセットで機能できるようになりました。
- 多数の警告 (GCC および Clang) が修正されました。
- 数多くのメモリー管理の修正。
- 数多くのメモリーリークの修正。
- XCCDF ファイルのプラットフォーム要素は、XCCDF 仕様に従って適切に解決されるようになりました。
- uClibc との互換性が向上しました。
- ローカルおよびリモートのファイルシステムの検出方法が改善されました。
- キャッシュを手動で開くのではなく、**pkgCacheFile** を使用するように **dpkginfo** プローブが修正されました。
- OpenSCAP スキャンレポートは、有効な HTML5 ドキュメントになりました。
- ファイルプローブの不要な再帰を修正しました。

(BZ#1887794)

RHEL 8 STIG セキュリティープロファイルがバージョン V1R1 に更新されました。

[RHBA-2021:1886](#) アドバイザリーリリースで、SCAP Security Guide の **DISA STIG for Red Hat Enterprise Linux 8** プロファイルが、最新バージョンの **V1R1** に更新されました。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。最初の反復により、STIG に関するカバレッジの約 60% が発生します。

ドラフトプロファイル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。テスト環境で修復を最初に実行します。

(BZ#1918742)

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。

[RHBA-2021:4098](#) アドバイザリーのリリースにより、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#2005431)

ANSSI-BP-028 Minimal、Intermediary、および Enhanced レベルのプロファイルが SCAP セキュリティガイドで利用可能になりました。

新しいプロファイルを使用すると、最小、中間、および強化レベルの GNU/Linux システムの AMD National Security Agency (ANSSI) から推奨事項にシステムを強化できます。これにより、ANSSI Ansible Playbook および ANSSI SCAP プロファイルを使用し、必要な ANSSI 強化レベルに従って、RHEL 8 システムのコンプライアンスを設定および自動化できます。

(BZ#1778188)

sudo 権限で scap-workbench がリモートシステムをスキャンできるようになりました。

scap-workbench GUI ツールが、パスワードなしの **sudo** アクセスを使用したりリモートシステムのスキャンをサポートするようになりました。この機能は、ルートの認証情報を指定することで課されるセキュリティリスクを軽減します。

パスワードなしの **sudo** アクセスと **remediate** オプションを指定して **scap-workbench** を使用する場合は、細心の注意を払ってください。Red Hat は、OpenSCAP スキャナーのみに適切にセキュアなユーザーアカウントを割り当てることを推奨します。

(BZ#1877522)

rhel8-tang コンテナイメージが利用可能になりました。

今回のリリースにより、**rhel8/rhel8-tang** コンテナイメージが **registry.redhat.io** カタログで利用可能になりました。このコンテナイメージは、OpenShift Container Platform (OCP) クラスタまたは別の仮想マシンで実行する Clevis クライアントの Tang-server 復号化機能を提供します。

(BZ#1913310)

Clevis がバージョン 15 にリベースされました。

clevis パッケージがアップストリームバージョン 15 にリベースされました。このバージョンでは、以前のバージョンに対するバグ修正および機能拡張が数多く追加されました。主な変更点は以下の通りです。

- Clevis は汎用 `initramfs` を生成し、カーネルコマンドラインに `rd.neednet=1` パラメーターを自動的に追加しなくなりました。
- `clevis` は、**sss** ピンを使用する間違っただ設定を適切に処理し、`clevis encrypt sss` サブコマンドはエラーの原因を示す出力を返すようになりました。

(BZ#1887836)

Clevis が自動的に `rd.neednet=1` を追加しなくなりました。

Clevis は、ホスト固有の設定オプションなしに汎用 `initrd` (initial ramdisk) を正しく生成するようになりました。その結果、Clevis は `rd.neednet=1` パラメーターがカーネルコマンドラインに自動的に追加されなくなりました。

設定が以前の機能を使用する場合は、`--hostonly-cmdline` 引数を指定して `dracut` コマンドを入力するか、`/etc/dracut.conf.d` に `clevis.conf` ファイルを作成し、`hostonly_cmdline=yes` オプションをファイルに追加します。`initrd` ビルドプロセス中に Tang バインディングが存在する必要があります。

(BZ#1853651)

新しいパッケージ: `rsyslog-udpspooof`

rsyslog-udpspooof サブパッケージが RHEL 8 に戻されました。このモジュールは通常の UDP フォワーダーに似ていますが、**syslog** パケットのソース IP を維持しつつ、異なるネットワークセグメント間で **syslog** のリレーを許可します。

(BZ#1869874)

fapolicyd が 1.0.2 にリベースされました。

fapolicyd パッケージがアップストリームバージョン 1.0.2 にリベースされました。このバージョンでは、以前のバージョンに対するバグ修正および機能拡張が数多く追加されました。主な変更点は以下の通りです。

- 以下を使用して **integrity** チェックを有効にする整合性設定オプションを追加しました。
 - ファイルサイズの比較
 - SHA-256 ハッシュの比較
 - Integrity Measurement Architecture (IMA) サブシステム
- **fapolicyd** RPM プラグインは、YUM パッケージマネージャーまたは RPM Package Manager のいずれかで処理されるシステム更新をすべて登録するようになりました。
- ルールに GID をサブジェクトに含むことができるようになりました。
- デバッグメッセージおよび **syslog** メッセージにルール番号を追加できるようになりました。

(BZ#1887451)

RPM トランザクション中の変更について、新しい RPM プラグインが `fapolicyd` に通知しません。

rpm パッケージの今回の更新で、**fapolicyd** フレームワークと RPM データベースを統合する新しい

RPM プラグインが導入されました。プラグインは、RPM トランザクション中にインストール済みおよび変更されたファイルについて **fapolicyd** に通知します。これにより、**fapolicyd** が整合性チェックに対応するようになりました。

機能は YUM トランザクションに制限されず、RPM の変更点も対応しているため、RPM プラグインは YUM プラグインに代わることに注意してください。

(BZ#1923167)

4.7. ネットワーク

ethtool ユーティリティーの PTP 機能出力形式が変更されました

RHEL 8.4 以降、**ethtool** ユーティリティーはカーネルとの通信に **ioctl()** システムコールではなく **netlink** インターフェイスを使用します。したがって、**ethtool -T <network_controller>** コマンドを使用すると、Precision Time Protocol (PTP) 値の形式が変更されます。

以前は、**ioctl()** インターフェイスでは、**ethtool** は **ethtool -internal** 文字列テーブルを使用して機能ビット名を変換し、**ethtool -T <network_controller>** コマンドは次のように表示しました。

```
Time stamping parameters for <network_controller>:
Capabilities:
hardware-transmit (SOF_TIMESTAMPING_TX_HARDWARE)
software-transmit (SOF_TIMESTAMPING_TX_SOFTWARE)
...
```

netlink インターフェイスを使用すると、**ethtool** はカーネルから文字列を受け取ります。これらの文字列には、内部の **SOF_TIMESTAMPING_*** 名は含まれません。したがって、**ethtool -T <network_controller>** は次のように表示されます。

```
Time stamping parameters for <network_controller>:
Capabilities:
hardware-transmit
software-transmit
...
```

スクリプトまたはアプリケーションで **ethtool** の PTP 機能出力を使用する場合は、それに応じて更新してください。

(JIRA:RHELDOS-18188)

XDP に条件付きサポートを追加

Red Hat は、以下の条件がすべて適用されている場合に限り、eXpress Data Path (XDP) 機能をサポートします。

- AMD または Intel 64 ビットアーキテクチャーに XDP プログラムを読み込みます。
- **libxdp** ライブラリーを使用して、カーネルにプログラムを読み込みます。
- XDP プログラムが XDP ハードウェアオフロードを使用しません

RHEL 8.4 では、**XDP** プログラムで **XDP_TX** および **XDP_REDIRECT** の戻りコードに対応するようになりました。

サポートされていない XDP 機能の詳細は、[テクノロジープレビューとして利用できる XDP 機能](#) を参照してください。

(BZ#1952421)

NetworkManager をバージョン 1.30.0 にリベース

NetworkManager パッケージがアップストリームバージョン 1.30.0 にアップグレードされ、以前のバージョンに対するバグ修正や機能強化が数多く追加されました。

- DHCP サーバー ID NetManager が拒否すべきリース提供を定義するために、**ipv4.dhcp-reject-servers** 接続プロパティが追加されました。
- カスタムの Vendor Class Identifier DHCP オプション値を送信するために、**ipv4.dhcp-vendor-class-identifier** 接続プロパティが追加されました。
- **active_slave** ボンディングオプションが非推奨になりました。代わりに、コントローラー接続で **primary** オプションを設定します。
- **nm-initrd-generator** ユーティリティーが、インターフェイスを示す MAC アドレスに対応するようになりました。
- **nm-initrd-generator** ユーティリティージェネレーターが InfiniBand 接続の作成に対応するようになりました。
- **NetworkManager-wait-online** サービスのタイムアウトが 60 秒に増えました。
- [RFC4361](#) に準拠するように **ipv4.dhcp-client-id=ipv6-duid** 接続プロパティが追加されました。
- **ethtool** オフロード機能が追加されました。
- WPA3 Enterprise Suite-B 192-bit モードに対応するようになりました。
- 仮想イーサネット (**veth**) デバイスのサポートが追加されました。

主な変更の詳細は、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。

- [NetworkManager 1.30.0](#)
- [NetworkManager 1.28.0](#)

(BZ#1878783)

iproute2 ユーティリティーでは、イーサネットヘッダーの前に MPLS ヘッダーを追加するためのトラフィック制御アクションが導入されました。

今回の機能強化により、**iproute2** ユーティリティーが 3 つの新しいトラフィック制御 (**tc**) アクションを提供するようになりました。

- **MAC_push: act_mpls** モジュールは、元のイーサネットヘッダーの前に MPLS ラベルを追加するこのアクションを提供します。
- **push_eth: act_vlan** モジュールは、パケットの最初にイーサネットヘッダーを構築するこのアクションを提供します。
- **pop_eth: act_vlan** モジュールは、外部イーサネットヘッダーを削除するためにこのアクションを提供します。

これらの **tc** アクションは、イーサネットヘッダーの前にマルチプロトコルラベルスイッチ (MPLS) ラベルを追加することで、レイヤー 2 の仮想プライベートネットワーク (L2VPN) を実装するのに役立ちます。これらのアクションは、**tc filters** をネットワークインターフェイスに追加する際に使用できません。

MPLS 自体はテクノロジープレビュー機能であるため、Red Hat は、これらのアクション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。

これらのアクションとそのパラメーターの詳細は、man ページの **tc-mpls(8)** および **tc-vlan(8)** を参照してください。

(BZ#1861261)

nmstate API が完全にサポートされるようになりました。

以前はテクノロジープレビューとして使用されていた nmstate は、ホストのネットワーク API で、RHEL 8.4 で完全対応になりました。**nmstate** パッケージは、ライブラリーと **nmstatectl** コマンドラインユーティリティを提供し、ホストのネットワーク設定を宣言型で管理できます。ネットワークの状態は事前定義済みのスキーマで説明されています。現在の状態と、必要な状態への変更の報告は、両者ともこのスキーマに一致します。

詳細は、`/usr/share/doc/nmstate/README.md` ファイルおよび [ネットワークの設定および管理の nmstatectl](#) セクションを参照してください。

(BZ#1674456)

新しいパッケージ: rshim

rshim パッケージは、Mellanox BlueField rshim user-space ドライバーを提供します。これにより、外部ホストマシンから BlueField SmartNIC ターゲット上の rshim リソースにアクセスできるようになります。現行バージョンの rshim user-space ドライバーは、ブートイメージのプッシュおよび仮想コンソールアクセス用にデバイスファイルを実装します。さらに、BlueField ターゲットに接続する仮想ネットワークインターフェイスを作成し、内部 rshim レジスターにアクセスする方法を提供します。

仮想コンソールまたは仮想ネットワークインターフェイスを機能させるには、ターゲットが **tmfifo** ドライバーを実行している必要があることに注意してください。

(BZ#1744737)

iptraf-ng が 1.2.1 にリベースされました。

iptraf-ng パッケージがアップストリームバージョン 1.2.1 にリベースされ、バグ修正および機能強化が複数追加されました。以下に例を示します。

- **iptraf-ng** アプリケーションは、削除されたインターフェイスの詳細な統計を表示する際に、CPU の使用率が 100% にならなくなりました。
- **printf()** 関数の安全でない処理引数が修正されました。
- IPoIB (IP over InfiniBand) インターフェイスに部分的なサポートが追加されました。カーネルはインターフェイスでソースアドレスを提供しないため、この機能を LAN station monitor モードで使用することはできません。
- **iptraf-ng** が、マルチグラウンドの速度でパケットをキャプチャーできるように、パケットキャプチャーが追加されました。
- **Home**、**End**、**Page up**、および **Page down** のキーボードキーを使用してスクロールできるようになりました。

- アプリケーションは、破棄されたパケット数を表示できるようになりました。

(BZ#1906097)

4.8. カーネル

RHEL 8.4 のカーネルバージョン

Red Hat Enterprise Linux 8.4 には、カーネルバージョン 4.18.0-305 が同梱されています。

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

(BZ#1839151)

Extended Berkeley Packet Filter for RHEL 8.4

extended Berkeley Packet Filter (eBPF) は、限られた一連の関数にアクセスできる制限付きサンドボックス環境において、カーネル領域でのコード実行を可能にするカーネル内の仮想マシンです。この仮想マシンは、特別なアセンブリのようなコードを実行します。

eBPF バイトコードが最初にカーネルにロードされ、その後に検証が行われます。次に実行時のコンパイルでコードがネイティブマシンコードに変換され、その後、仮想マシンがコードを実行します。

Red Hat は、**eBPF** 仮想マシンを使用するコンポーネントを数多く提供しています。各コンポーネントの開発フェーズはさまざまです。そのため、現在すべてのコンポーネントが完全にサポートされている訳ではありません。RHEL 8.4 では、以下の **eBPF** コンポーネントがサポートされています。

- **eBPF** を使用して Linux オペレーティングシステムの I/O 分析、ネットワーク、およびモニタリングを行う **BPF コンパイラコレクション (BCC)** ツールパッケージ。
- **BCC** ライブラリー。これを使用すると、**BCC** ツールパッケージで提供されるツールと同様のツールを開発できます。
- **eBPF for Traffic Control (tc)** 機能。これにより、カーネルネットワークデータパスでのプログラミング可能なパケット処理が可能になります。
- カーネルネットワーキングスタックを処理する前に受信パケットへのアクセスを提供する **eXpress Data Path (XDP)** 機能は、特定の条件でサポートされます。
- **libbpf** パッケージ。 **bpftrace** および **bpf/xdp** 開発のようなアプリケーションに関連する bpf に極めて重要
- **xdp-tools** パッケージが、AMD および Intel 64 ビットアーキテクチャーでサポートされるようになりました。このパッケージには、**XDP** 機能用のユーザー空間サポートユーティリティーが含まれます。これには、 **libxdp** ライブラリー、XDP プログラムを読み込む **xdp-loader** ユーティリティー、パケットフィルタリングの **xdp-filter** サンプルプログラム、XDP が有効になっているネットワークインターフェイスからパケットを取得する **xdpdump** ユーティリティーなどが含まれます。

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

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

- **bpftrace** トレース言語
- **eXpress Data Path (XDP)** パスをユーザー空間に接続する **AF_XDP** ソケット

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

(BZ#1780124)

新しいパッケージ: kmod-redhat-oracleasm

今回の更新で、ASMLib ユーティリティーのカーネルモジュール部分を提供する新しい **kmod-redhat-oracleasm** パッケージが追加されました。Oracle Automated Storage Management (ASM) は、Oracle データベースのデータボリュームマネージャーです。ASMLib は、Linux システムで Oracle ASM デバイスを管理するために使用できるオプションのユーティリティーです。

(BZ#1827015)

xmon プログラムが、攻撃に対して Secure Boot と kernel_lock の耐障害性に対応するように変更

Secure Boot メカニズムが無効になっている場合は、カーネルコマンドラインで **xmon** プログラムを読み書きモード (**xmon=rw**) に設定できます。ただし、**xmon=rw** を指定して Secure Boot モードで起動すると、**kernel_lockdown** 機能は **xmon=rw** を上書きし、これを読み取り専用モードに変更します。Secure Boot の有効化に応じて **xmon** の追加動作を以下に示します。

Secure Boot がオンになっている。

- **xmon=ro** (デフォルト)
- スタックトレースが出力されている
- メモリーの読み取りが動作している
- メモリー書き込みがブロックされている

Secure Boot がオフになっている。

- **xmon=rw** を設定可能
- スタックトレースが常に出力される
- メモリーの読み取りが常に機能する
- メモリー書き込みが、**xmon=rw** でのみ許可されている

xmon へのこれらの変更は、root 権限による攻撃者に対して Secure Boot および **kernel_lock** の耐障害性をサポートすることを目的としています。

カーネルコマンドラインパラメーターの設定方法は、[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。

(BZ#1952161)

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

Red Hat Enterprise Linux 8.4 は、Omni-Path Architecture (OPA) ホストソフトウェアに完全に対応しています。OPA は、クラスター環境のコンピュートと I/O ノード間の高性能データ転送 (高帯域幅、高メッセージレート、低レイテンシー) のために、初期化とセットアップを行う Host Fabric Interface (HFI) ハードウェアを提供します。

Omni-Path Architecture のインストール手順は、[Cornelis Omni-Path ファブリックソフトウェア](#) のリリースノートを参照してください。

(BZ#1960412)

SLAB キャッシュのマージがデフォルトで無効

CONFIG_SLAB_MERGE_DEFAULT カーネル設定オプションが無効になり、SLAB キャッシュはデフォルトでマージされなくなりました。この変更は、キャッシュ使用量のアロケータの信頼性とトレーサビリティを強化することを目的としています。以前の slab-cache マージ動作が望ましい場合は、カーネルコマンドラインに **slub_merge** パラメータを追加して再度有効にできます。カーネルコマンドラインパラメータの設定方法は、[カーネルコマンドラインパラメータの設定](#) を参照してください。

(BZ#1871214)

ima-evm-utils パッケージがバージョン 1.3.2 にリベースされました。

ima-evm-utils パッケージがバージョン 1.3.2 にアップグレードされ、バグ修正および機能強化が複数追加されました。以下は、主な変更点です。

- Trusted Platform Module (TPM2) マルチバンク機能の処理に対するサポートが追加されました。
- ブートアグリゲート値を Platform Configuration Registers (PCR) 8 および 9 に拡張
- CLI パラメータを使用した事前ロードされた OpenSSL エンジン
- Intel Task State Segment (TSS2) PCR 読み込みのサポートが追加されました。
- 元の Integrity Measurement Architecture (IMA) テンプレートのサポートを追加

libimaevm.so.0 および **libimaevm.so.2** ライブラリーは **ima-evm-utils** の一部です。 **libimaevm.so.0** のユーザーは、新しいアプリケーションで **libimaevm.so.2** を使用すると影響を受けません。

(BZ#1868683)

サポートされる CPU アーキテクチャーでの IMA および EVM 機能のレベル化

ARM を除くすべての CPU アーキテクチャーでは、IMA (Integrity Measurement Architecture) および Extended Verification Module (EVM) 技術に類似した機能サポートがあります。有効な機能は、CPU アーキテクチャーごとに異なります。サポートされている各 CPU アーキテクチャーにおける最も重要な変更点を以下に示します。

- IBM Z: IMA アプリおよび信頼できるキーリングの有効化。
- AMD64 および Intel 64: セキュアなブート状態の特定のアーキテクチャーポリシー。
- IBM Power System (little-endian): セキュアで信頼できるブート状態の特定のアーキテクチャーポリシー。
- SHA-256 (サポートされているすべてのアーキテクチャーのデフォルトハッシュアルゴリズム)
- すべてのアーキテクチャーで、測定テンプレートは IMA-SIG に変更になりました。このテンプレートには、存在する場合に署名ビットが含まれます。この形式は **d-ng|n-ng|sig** です。

この更新の目的は、サポートされるすべての CPU アーキテクチャーでユーザー空間アプリケーションを効率的に動作できるように、IMA と EVM の機能差レベルを低減することです。

(BZ#1869758)

RHEL 8 に disabled-by-default として、**プロアクティブコンパクション機能**が追加されました。

実行中のワークロードアクティビティーにより、システムメモリーが断片化されます。断片化により、容量とパフォーマンスの問題が発生する可能性があります。場合によっては、プログラムエラーも可能となります。したがって、カーネルは、メモリー圧縮と呼ばれるリアクティブメカニズムに依存します。メカニズムの元の設計は保存でき、割り当て要求に応じて圧縮アクティビティーが開始されます。ただし、リアクティブ動作は、システムメモリーがすでに断片化されている場合は、割り当てレイテンシーを長くする傾向にあります。**プロアクティブコンパクション**は、割り当て要求を行う **前に** メモリー圧縮作業を **定期的** に開始し、設計を改良します。今回の機能拡張により、オンデマンドでメモリーコンパクションを必要とせずに、メモリーの割り当て要求がメモリーの物理的に連続しているブロックを見つけられる可能性が高くなりました。その結果、特定のメモリー割り当て要求のレイテンシーが短縮されます。



警告

プロアクティブな圧縮処理により、圧縮のアクティビティーが向上する場合があります。これは、異なるプロセスに属するメモリーページが移動および再マッピングされるため、システム全体の影響を与える可能性があります。したがって、**プロアクティブコンパクション**な圧縮を有効にするには、アプリケーションでのレイテンシーが急増しないように注意する必要があります。

(BZ#1848427)

RHEL 8 に **EDAC サポート**が追加されました。

今回の更新で、RHEL 8 は、8 年目および 9 世代 Intel Core プロセッサー (CoffeeLake) で設定された Error Detection and Correction (EDAC) カーネルモジュールに対応します。EDAC カーネルモジュールは主に Error Code Correction (ECC) メモリーを処理し、PCI バスパリティエラーを検出し、報告します。

(BZ#1847567)

新しいパッケージ: kpatch-dnf

kpatch-dnf パッケージは、DNF プラグインを提供します。これにより、カーネルのライブパッチ更新に RHEL システムをサブスクライブできます。サブスクリプションは、今後インストールされるカーネルを含め、システムにインストールされているすべてのカーネルに影響します。kpatch-dnf の詳細は、man ページの dnf-kpatch(8)、または **カーネルの管理、監視、および更新** を参照してください。

(BZ#1798711)

slab メモリー用の新しい cgroups コントローラー実装

コントロールグループ技術にスラブメモリーコントローラーの新しい実装が RHEL 8 で利用可能になりました。現在、単一のメモリースラブには、異なるメモリー **コントロールグループ** が所有するオブジェクトを含めることができます。このスラブのメモリーコントローラーにより、スラブの使用率が改善され (最大 45%)、ページレベルからオブジェクトレベルにメモリーアカウンティングを移動できるようになりました。また、この変更により、各メモリーコントロールグループの重複された CPU ごとの

スラブキャッシュがなくなり、すべてのメモリーコントロールグループに対して CPU ごとのスラブキャッシュの1つとノードごとのスラブキャッシュが1つ確立されます。その結果、カーネルメモリーのフットプリントの合計で大幅に減少し、メモリーの断片化による悪影響を観察できます。

新メモリーアカウンティングおよびより正確なメモリーアカウンティングには、より多くの CPU 時間が必要なことに注意してください。ただし、実際には大きな違いは無視可能であると思われます。

(BZ#1877019)

RHEL 8 にタイム名前空間が追加されました。

名前空間を使用すると、システムの単調増加およびブートタイムクロックが、AMD64、Intel 64、および 64 ビット ARM アーキテクチャーの名前空間ごとのオフセットと連携できるようになります。この機能は、Linux コンテナ内の日時の変更や、チェックポイントから復元後のクロックのコンテナのチューニングに適しています。その結果、ユーザーは個々のコンテナに個別に時間を設定できるようになりました。

(BZ#1548297)

新機能: Free memory page returning

今回の更新で、RHEL 8 ホストカーネルは、仮想マシンが使用していないメモリーページをハイパーバイザーに返すことができるようになりました。これにより、ホストの安定性およびリソースの効率が改善されます。応答するメモリーページが機能するには、仮想マシンで設定する必要があります。また、仮想マシンは **virtio_balloon** デバイスも使用する必要があることに注意してください。

(BZ#1839055)

perf top でのソート順序の変更に対応

今回の更新により、**perf top** が最初の列でソートされるのではなく、グループの複数のイベントがサンプリングされる場合に備えて、任意のイベント列でサンプルをソートできるようになりました。これにより、数字のキーを使用して、一致するデータ列でテーブルをソートします。



注記

列番号は **0** から始まります。

--group-sort-idx コマンドラインオプションを使用すると、コラム番号でソートできます。

(BZ#1851933)

kabi_whitelist パッケージの名前が kabi_stablelist に変更されました。

Red Hat は問題のある言語の置き換えにしたがって、RHEL 8.4 リリースで **kabi_whitelist** パッケージの名前を **kabi_stablelist** に変更します。

(BZ#1867910、[BZ#1886901](#))

BPF がバージョン 5.9 にリベースされました。

RHEL 8 の **bpf** カーネルテクノロジーは、カーネル v5.9 のアップストリームのカウンターパートから最新に起動しました。

この更新により、バグ修正および機能強化が複数追加されました。以下は、主な変更点です。

- マップ要素の Berkeley Packet Filter (BPF) イテレーターが追加され、カーネル内の検査を効率的に行うためにすべての BPF プログラムを繰り返し処理できるようになりました。

- 同じコントロールグループ (cgroup) のプログラムは、cgroup ローカルストレージマップを共有できます。
- BPF プログラムは、ソケットルックアップで実行できます。
- **SO_KEEPALIVE** および関連するオプションは、**bpf_setsockopt()** ヘルパーで利用できます。

一部の BPF プログラムには、ソースコードへの変更が必要な場合があることに注意してください。

(BZ#1874005)

bcc パッケージがバージョン 0.16.0 にリベースされました。

pcp パッケージがバージョン 0.16.0 にアップグレードされ、バグ修正および機能強化が複数追加されました。以下は、主な変更点です。

- ユーティリティー **klockstat** および **funcinterval** を追加しました。
- man ページ **tcpconnect** のさまざまな箇所を修正
- **tcptracer** ツールの出力に IPv6 アドレスの SPORT 列および DPORT 列が表示されるように修正
- 破損した依存関係の修正

(BZ#1879411)

bpftrace がバージョン 0.11.0 にリベースされました。

bpftrace パッケージがバージョン 0.11.0 にアップグレードされ、バグ修正および機能強化が複数追加されました。以下は、主な変更点です。

- ユーティリティー **threadsnoop**、**tcpsynbl**、**tcplife**、**swapin**、**setuids**、**napttime** を追加しました。
- **tcpdrop.bt** および **syncsnoop.bt** ツールの実行の失敗を修正しました。
- IBM Z アーキテクチャーで Berkeley Packet Filter (BPF) プログラムの読み込みに失敗する問題が修正されました。
- シンボルの検索エラーを修正しました。

(BZ#1879413)

libbpf がバージョン 0.2.0.1 にリベースされました。

libbpf パッケージがバージョン 0.2.0.1 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。以下は、主な変更点です。

- BPF Type Format (BTF) 構造アクセスのあるプログラムから **bpf_map** 構造の Berkeley Packet Filter (BPF) マップフィールドにアクセスするためのサポートが追加されました。
- BPF リングバッファの追加
- **bpf** イテレーターインフラストラクチャーを追加しました。
- **bpf_link** の可観測性を改善しました

(BZ#1919345)

perf が、perf を停止または再起動せずに実行中のコレクターへのトレースポイントの追加または削除に対応

以前は、**perf record** のインスタンスからトレースポイントを追加または削除するには、**perf** プロセスを停止する必要がありました。そのため、プロセスが停止したときに発生したパフォーマンスデータが収集されないため、失われていました。今回の更新で、**perf record** プロセスを停止せずに、**perf record** が制御パイプインターフェイスを介して **perf** レコードで収集されるトレースポイントを動的に有効または無効にできるようになりました。

(BZ#1844111)

perf ツールが、トレースデータの絶対タイムスタンプの記録および表示をサポートするようになりました。

今回の更新で、**perf script** は、絶対タイムスタンプを使用してトレースデータを記録し、表示できるようになりました。

注記: 絶対タイムスタンプでトレースデータを表示するには、クロック ID を指定してそのデータを記録する必要があります。

絶対タイムスタンプでデータを記録するには、クロック ID を指定します。

```
# perf record -k CLOCK_MONOTONIC sleep 1
```

指定したクロック ID で記録されたトレースデータを表示するには、次のコマンドを実行します。

```
# perf script -F+tod
```

(BZ#1811839)

dwarves がバージョン 1.19.1 にリベースされました。

dwarves パッケージがバージョン 1.19.1 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。今回の更新では、**ftrace** 関数のサブセットが生成されるように、関連する **ftrace** エントリーを使用して DWARF デバッグデータから機能を確認する新しい方法が導入されました。

(BZ#1903566)

perf が、指定されたイベントを使用してスナップショットをトリガーするボックバッファをサポートするようになりました。

今回の更新により、指定したイベントが検出されると、**perf.data** ファイルにデータを書き込むカスタムブローバッファを作成できるようになりました。これにより、**perf record** は、過剰なオーバーヘッドを生成し、**perf.data** ファイルに継続的にデータを書き込むことで、システムバックグラウンドで継続的に実行できます。

イベント固有のスナップショットを記録する **perf** ツールを使用してカスタムのアップリンクバッファを作成するには、以下のコマンドを使用します。

```
# perf record --overwrite -e _events_to_be_collected_ --switch-output-event
_snapshot_trigger_event_
```

(BZ#1844086)

Kernel DRBG および Jitter エントロピーソースは NIST SP 800-90A および NIST SP 800-90B に準拠しています。

Kernel deterministic Random Bit Generator (DRBG) および Jitter エントロピーソースは、DRBG (NIST SP 800-90A) 仕様に使用されるエントロピーソースの推奨事項に準拠するようになりました。また、ランダムなビットの生成 (NIST SP 800-90B) 仕様に使用されるエントロピーソースの推奨事項に準拠するようになりました。これにより、FIPS モードのアプリケーションは、これらのソースを FIPS 準拠のランダム性やリモートソースとして使用できます。

(BZ#1905088)

kdump が仮想ローカルエリアネットワークタグ付けされたチームネットワークインターフェイスに対応

今回の更新で、**kdump** 用の仮想ローカルエリアネットワークタグ付けされたチームインターフェイスの設定がサポートされるようになりました。これにより、**kdump** が、**vmcore** ファイルをダンプするためにタグ付けされた仮想チームインターフェイスを使用して **vmcore** ファイルをダンプできるようになりました。

(BZ#1844941)

kernel-rt ソースツリーが RHEL 8.4 ツリーに更新

kernel-rt ソースが更新され、最新の Red Hat Enterprise Linux カーネルソースツリーを使用するようになりました。リアルタイムパッチセットも、最新のアップストリームバージョン v5.10-rt7 に更新されました。これらの更新はいずれも、バグ修正および機能強化を多数提供します。

(BZ#1858099, BZ#1858105)

RHEL 8.4 ディストリビューションに **stalld** パッケージが追加されました。

今回の更新で、**stalld** パッケージが RHEL 8.4.0 に追加されました。**stalld** は、低レイテンシーアプリケーションを実行しているシステムでスレッドを監視するデーモンです。これは、指定されたしきい値について CPU にスケジュールせずに、run-queue 上にあるジョブスレッドをチェックします。

停止しているスレッドを検出すると、**stalld** はスケジューリングポリシーを一時的に **SCHED_DEADLINE** に変更し、CPU 時間のスライスをスレッドに割り当てて進めます。時間のスライスの完了またはスレッドブロック時に、スレッドは元のスケジューリングポリシーに戻ります。

(BZ#1875037)

hv_24x7 と hv_gpci PMUs での CPU ホットプラグのサポート

今回の更新で、PMU カウンターが CPU のホットプラグに正しく対応するようになりました。その結果、**hv_gpci** イベントカウンターが無効な CPU で実行されている場合、カウントは別の CPU にリダイレクトされます。

(BZ#1844416)

POWERPC hv_24x7 ネストイベントのメトリックが利用可能に

POWERPC **hv_24x7** ネストイベントのメトリックが **perf** で利用可能になりました。複数のイベントを集計することで、これらのメトリックは **perf** カウンターから取得した値をより明確に理解し、CPU がワークロードをどのように効果的に処理できるかを理解します。

(BZ#1780258)

hwloc がバージョン 2.2.0 にリベースされました。

hwloc パッケージがバージョン 2.2.0 にアップグレードされ、以下の変更が行われました。

- **hwloc** 機能が、合計ディスクサイズやセクターサイズを含む NVMe (Nonvolatile Memory Express) ドライブの詳細を報告

(BZ#1841354)

igc ドライバーが完全にサポートされるようになりました。

igc Intel 2.5G Ethernet Linux 有線 LAN ドライバーは、テクノロジープレビューとして RHEL 8.1 に導入されました。RHEL 8.4 以降では、すべてのアーキテクチャーで完全にサポートされています。**ethtool** ユーティリティーは **igc** 有線 LAN もサポートします。

(BZ#1495358)

4.9. ファイルシステムおよびストレージ

RHEL インストールで、サイズが 16 TiB の swap パーティションの作成に対応

以前は、RHEL のインストール時に、インストーラーにより、自動パーティションおよび手動パーティション設定用に最大 128 GB の swap パーティションが作成されていました。

今回の更新で、自動パーティション設定では、インストーラーは最大 128 GB の swap パーティションの作成を継続しますが、手動パーティション設定の場合は 16 TiB の swap パーティションを作成できるようになりました。

(BZ#1656485)

NVMe デバイスの不適切な削除

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

以下の点に注意してください。

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

(BZ#1634655)

Stratis ファイルシステムのシンボリックリンクパスが変更されました

今回の機能強化により、Stratis ファイルシステムの symlink パスが **/stratis/<stratis-pool>/<filesystem-name>** から **/dev/stratis/<stratis-pool>/<filesystem-name>** に変更になりました。したがって、既存の Stratis シンボリックリンクをすべて移行して、新しいシンボリックリンクパスを使用する必要があります。

含まれる **stratis_migrate_symlinks.sh** 移行スクリプトを使用するか、システムを再起動してシンボリックリンクパスを更新します。**systemd** ユニットファイルまたは **/etc/fstab** ファイルを手動で変更して Stratis ファイルシステムを自動的にマウントする場合は、新しい symlink リンクパスで更新する必要があります。



注記

新しい Stratis symlink パスで設定を更新しない、または自動マウントを一時的に無効にする場合は、次回システムの起動または再起動を行うと、完全に起動しなくなることがあります。

(BZ#1798244)

Stratis が、Supplementary Clevis 暗号化ポリシーへの暗号化プールへのバインディング設定に対応

今回の機能拡張により、Tang サーバー、または Trusted Platform Module (TPM) 2.0 を使用して、暗号化された Stratis プールを Network Bound Disk Encryption (NBDE) にバインドできるようになりました。暗号化された Stratis プールを NBDE または TPM 2.0 にバインドすると、プールの自動アンロックが容易になります。これにより、各システム再起動後にカーネルキーリングの説明を提供することなく Stratis プールにアクセスできます。Stratis プールを通常の Clevis 暗号化ポリシーにバインドすると、プライマリーカーネルキーリング暗号化は削除されないことに注意してください。

(BZ#1868100)

XFS ファイルシステムおよび ext4 ファイルシステムで DAX が有効な場合を制御する新しいマウントオプション

今回の更新では、**FS_XFLAG_DAX** inode フラグと組み合わせると、XFS ファイルシステムおよび ext4 ファイルシステムのファイルに対する Direct Access (DAX) モードを詳細に制御できる新しいマウントオプションが導入されました。本リリース以前は、**dax** マウントオプションを使用してファイルシステム全体に対して DAX が有効になっていました。ダイレクトアクセスモードをファイルごとに有効にできるようになりました。

オンディスクフラグ **FS_XFLAG_DAX** は、特定のファイルまたはディレクトリーに対して DAX を選択的に有効または無効にするために使用されます。**dax** マウントオプションは、フラグを受け入れるかどうかを決定します。

- **-o dax=inode: FS_XFLAG_DAX** に従います。dax オプションを指定しないと、これがデフォルトになります。
- **-o dax=never**: DAX を有効にしません。**FS_XFLAG_DAX** を無視します。
- **-o dax=always**: 常に DAX を有効にし、**FS_XFLAG_DAX** を無視します。
- **-o dax**: "dax=always" のエイリアスであるレガシーオプションです。これは今後削除される可能性があります。したがって、**-o dax=always** が推奨されます。

xfs_io ユーティリティーの **chatter** コマンドを使用して、**FS_XFLAG_DAX** フラグを設定できます。

```
# xfs_io -c "chattr +x" filename
```

(BZ#1838876、BZ#1838344)

SMB Direct がサポートされるようになりました。

今回の更新で、SMB クライアントが SMB Direct に対応するようになりました。

(BZ#1887940)

ファイルシステムをマウントするための新しい API が追加されました。

今回の更新で、ファイルシステムコンテキスト (**struct fs_context**) と呼ばれる内部カーネル構造に基づいてファイルシステムをマウントする新しい API が RHEL 8.4 に追加されました。これにより、ユーザー空間、VFS、およびファイルシステム間でのマウントパラメーターの通信に柔軟性が高まりました。これとともに、ファイルシステムコンテキスト上で動作している以下のシステムコールがあります。

- **fsopen(): fsname** パラメーターに名前が付けられたファイルシステム用に、カーネル内に空のファイルシステム設定コンテキストを作成し、作成モードに追加して、ファイル記述子に接続してから返します。
- **fsmount(): fsopen()** によって返されるファイル記述子を取り、そこに指定されたファイルシステムの **root** のマウントオブジェクトを作成します。
- **fsconfig(): fsopen(2)** または **fspick(2)** コールで設定したファイルシステム設定コンテキストに対して、コマンドと発行のパラメーターを提供します。
- **fspick():** カーネル内に新しいファイルシステム設定コンテキストを作成し、既存のスーパーブロックをそのカーネルにアタッチして再設定できるようにします。
- **move_mount():** マウントを別の場所に移動します。このマウントは、**fsmount()** または **open_tree()** が作成した未割り当てのマウントを **OPEN_TREE_CLONE** システムコールでアタッチするのにも使用できます。
- **open_tree():** パス名で指定されたマウントオブジェクトを選択し、新しいファイル記述子にアタッチするか、クローンを作成して、そのクローンをファイル記述子に接続します。

mount() システムコールに基づく古い API は引き続きサポートされることに注意してください。

詳細は、カーネルソースツリーの **Documentation/filesystems/mount_api.txt** ファイルを参照してください。

(BZ#1622041)

vfat ファイルシステムの mtime が発生しなくなる

今回の更新で、メモリー内とディスク上の書き込み時間との間の **vfat** ファイルシステムの **mtime** の不一致がなくなりました。この差異は、メモリー内とディスク上の **mtime** メタデータの差異によって生じました。

(BZ#1533270)

RHEL 8.4 が close_range() システムコールに対応しました。

今回の更新で、**close_range()** システムコールが RHEL 8.4 にバックポートされました。このシステムコールは、特定の範囲のファイル記述子をすべて効果的に閉じるため、アプリケーションが非常に大きな制限を設定する場合は、さまざまなファイル記述子を閉じる際に存在するタイミングの問題を回避します。

(BZ#1900674)

NFSv4.2 プロトコルを使用したユーザー拡張属性のサポートの追加

今回の更新で、ユーザー拡張属性 (RFC 8276) の NFSV4.2 クライアント側およびサーバー側のサポートが追加され、新たに以下のプロトコル拡張が追加されました。

新しい操作:

- **- GETXATTR** - ファイルの拡張属性を取得します。

- - **SETXATTR** - ファイルの拡張属性を設定します。
- - **LISTXATTR** - ファイルの拡張属性をリスト表示します。
- - **REMOVEXATTR** - ファイルの拡張属性を削除します。

新しいエラーコード:

- - **NFS4ERR-NOXATTR** - **xattr** が存在しません。
- - **NFS4ERR_XATTR2BIG** - **xattr** の値は大きすぎます。

新しい属性:

- - **xattr_support** - per-fs 読み取り専用属性は **xattrs** がサポートされているかどうかを判別します。 **True** に設定すると、オブジェクトのファイルシステムは拡張属性をサポートします。

(BZ#1888214)

4.10. 高可用性およびクラスター

コロケーション制約における非クリティカルなリソースに対応

今回の機能拡張により、コロケーション制約を設定して、制約に依存するリソースが移行しきい値に達すると、Pacemaker はそのリソースをオフラインにし、両方のリソースを別のノードに移動せずに現在のノードにプライマリーリソースを維持するようになります。この動作をサポートするために、コロケーション制約には **influence** オプション (**true** または **false** を設定可能) が設定され、リソースに **critical** なメタ属性があり、リソースには **true** または **false** を設定できるようになりました。 **critical** リソースのメタオプションの値により、リソースが依存するリソースとして関連する全コロケーション制約に対する **influence** オプションのデフォルト値が決まります。

influence 制約オプションの値が **true** の場合、プライマリーリソースと依存するリソース両方をアクティブに維持しようとします。依存するリソースが障害の移行しきい値に達すると、可能な場合は両方のリソースが別のノードに移行します。

influence のコロケーションオプションの値が **false** の場合、Pacemaker は、依存するリソースのステータスが原因でプライマリーリソースを移行しないようにします。この場合、依存するリソースが障害の移行しきい値に達すると、プライマリーリソースがアクティブで、現在のノードに留まると停止します。

デフォルトでは、**critical** リソースのメタオプションの値は **true** に設定され、次に **influence** オプションのデフォルト値が **true** であると判断します。これにより、Pacemaker が両方のリソースをアクティブに維持しようとする以前の動作が保持されます。

(BZ#1371576)

Pacemaker ルールでサポートされる新しい数値のデータ型

PCS が **number** のデータタイプに対応するようになりました。これは、ルールを受け入れる PCS コマンドで Pacemaker ルールを定義する際に使用できます。Pacemaker ルールは、**number** を二重の浮動小数点数として実装し、**integer** を 64 ビット整数として実装します。

(BZ#1869399)

クローンリソースまたは昇格可能なクローンリソースの作成時にカスタムクローン ID を指定する機能

クローンリソースまたは昇格可能なクローンリソースを作成すると、そのクローンリソースはデフォルトで **resource-id-clone** という名前になります。その ID がすでに使用されている場合には、PCS は接尾辞 **-integer** を追加します。整数値は **1** で始まり、追加のクローンごとに1つずつ増えます。**pcs resource create** または **pcs resource clone** コマンドを使用してクローンリソースを作成する場合は、**clone-id** オプションを使用して、クローンリソース ID または昇格可能なクローンリソース ID の名前を指定して上書きできるようになりました。クローンリソースの作成に関する詳細は、[複数のノードでアクティブなクラスターリソースの作成](#) を参照してください。

(BZ#1741056)

Corosync 設定を表示する新しいコマンド

新しい **pcs cluster config [show]** コマンドを使用して、出力形式で **corosync.conf** ファイルの内容を出力できるようになりました。デフォルトでは、**pcs cluster config** コマンドは **テキスト** 出力形式を使用します。これは、**pcs cluster setup** および **pcs cluster config update** コマンドと同じ構造およびオプション名を持つ、人間が判読可能な形式で Corosync 設定を表示します。

(BZ#1667066)

既存クラスターの Corosync 設定を変更する新しいコマンド

新しい **pcs cluster config update** コマンドを使用して、**corosync.conf** ファイルのパラメーターを変更できるようになりました。たとえば、このコマンドを使用すると、**totem** トークンを増大して、一時的にシステムの応答しない際にフェンシングを回避できます。**corosync.conf** ファイルの変更に関する詳細は、[Modifying the corosync.conf file with the pcs command](#) を参照してください。

(BZ#1667061)

既存クラスターでの Corosync トラフィック暗号化の有効化および無効化

以前のバージョンでは、Corosync トラフィック暗号化は新規クラスターの作成時にのみ設定できました。今回の更新により、以下が可能になります。

- **pcs cluster config update** コマンドを使用して、Corosync 暗号暗号およびハッシュの設定を変更できます。
- **pcs cluster authkey corosync** コマンドを使用して Corosync **authkey** を変更できます。

(BZ#1457314)

共有および暗号化された GFS2 ファイルシステム用の新しい **crypt** リソースエージェント

RHEL HA が、新しい **crypt** リソースエージェントに対応するようになりました。これにより、共有および暗号化された GFS2 ファイルシステムを提供するために使用できる LUKS 暗号化ブロックデバイスを設定できるようになりました。**crypt** リソースの使用は現在、GFS2 ファイルシステムでのみサポートされています。暗号化された GFS2 ファイルシステムを設定する方法は、[Configuring an encrypted GFS2 file system in a cluster](#) を参照してください。

(BZ#1471182)

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

新しいモジュール **python39** が導入されました。

RHEL 8.4 では Python 3.9 が導入されました。これは新しいモジュール **python39** および **ubi8/python-39** コンテナイメージで提供されます。

Python 3.8 と比較しての主な機能強化は、以下のとおりです。

- `merge (|)` および `update (|=)` 演算子が **dict** クラスに追加されました。
- 接頭辞と接尾辞を削除するメソッドが文字列に追加されました。
- **list** および **dict** などの、特定の標準タイプに、型ヒントとなる汎用が追加されました。
- IANA タイムゾーンデータベースが、新しい `zoneinfo` モジュールから利用できるようになりました。

Python 3.9 およびこれのためにビルドされたパッケージは、同じシステムの Python 3.6 と Python 3.8 と並行してインストールできます。

python39 モジュールからパッケージをインストールするには、たとえば、以下を使用します。

```
# yum install python39
# yum install python39-pip
```

python39:3.9 モジュールストリームは、自動的に有効になります。

インタプリタを実行するには、たとえば、以下を使用します。

```
$ python3.9
$ python3.9 -m pip --help
```

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

Red Hat は、RHEL 8 のライフサイクルが終了するまで Python 3.6 のサポートを継続することに留意してください。Python 3.8 と同様に、Python 3.9 のライフサイクルは短くなります。[Red Hat Enterprise Linux 8 Application Streams ライフサイクル](#) を参照してください。

(BZ#1877430)

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、RHEL 8 の以下のマイナーリリースで 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 \(CVE-2021-23336\)](#) 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#1928904)

新しいモジュールストリーム: swig:4.0

RHEL 8.4 では、新しいモジュールストリーム **swig:4.0** として利用できる、SWIG (Simplified Wrapper and Interface Generator) バージョン 4.0 が導入されました。

以前リリースされた **SWIG 3.0** への主な変更点は、以下のとおりです。

- サポートされる **Python** バージョンは 2.7 および 3.2 から 3.8 のみです。
- **Python** モジュールが改善されました。生成されたコードが簡素化され、ほとんどの最適化がデフォルトで有効になりました。
- **Ruby 2.7** のサポートが追加されました。
- **PHP 7** でサポートされる唯一のバージョンの PHP は削除されました。 **PHP 5** に対応しなくなりました。
- 大規模なインターフェイスファイルで **SWIG** を実行する場合は、パフォーマンスが大幅に改善されました。
- コマンドラインオプションファイル (応答ファイルとも呼ばれます) のサポートが追加されました。
- JavaScript **Node.js** バージョン 2 から 10 のサポートが追加されました。
- バージョン 4.4 から 5.1 へのサポートが追加されました。

swig:4.0 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 swig:4.0
```

swig:3.0 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。

swig モジュールストリームのサポート期間の詳細は、[Red Hat Enterprise Linux 8 Application Streams ライフサイクル](#) を参照してください。

([BZ#1853639](#))

新しいモジュールストリーム: subversion:1.14

RHEL 8.4 では、新しいモジュールストリーム **subversion:1.14** が導入されました。 **Subversion 1.14** は、最新の LTS (Long Term Support) リリースです。

以下は、RHEL 8.0 で配布された **Subversion 1.10** 以降の主な変更点です。

- **Subversion 1.14** には、**Subversion** を自動化し、お客様のビルドおよびリリースインフラストラクチャーに統合するための **Python 3** バインディングが含まれます。
- 新規の **svnadmin rev-size** コマンドを使用すると、ユーザーはリビジョンの合計サイズを判別できます。
- 新たな **svnadmin build-repcache** コマンドを使用すると、管理者はエントリーが見つからない **rep-cache** データベースに設定することができます。
- 現在の作業コピーステータスの概要を提供するために、新たな実験コマンドが追加されました。

- **svn log**、**svn info**、**svn list** コマンドに対するさまざまな改善が行われました。たとえば、**svn list --human-readable** は、ファイルサイズに人間が判読できる単位を使用するようになりました。
- 大規模な作業コピーの **svn status** に大幅に改良されました。

互換性情報:

- **Subversion 1.10** のクライアントおよびサーバーは **Subversion 1.14** サーバーおよびクライアントと相互運用します。ただし、クライアントとサーバーの両方が最新バージョンにアップグレードされない限り、特定の機能は利用できません。
- **Subversion 1.10** で作成されたリポジトリは、**Subversion 1.14** で正常にロードできます。
- RHEL 8 で配布されるサブバージョン **Subversion 1.14** を使用すると、クライアント側でプレーンテキストのパスワードをキャッシュできます。この動作は **Subversion 1.10** と同じですが、**Subversion 1.14** のアップストリームリリースとは異なります。
- 実験的な **Shelving** 機能が大幅に変更され **Subversion 1.10** で作成された shelves と互換性がありません。詳細およびアップグレードの手順は、[アップストリームのドキュメント](#) を参照してください。
- グローバルルールおよびリポジトリ固有のルールの両方を使用したパスベースの認証設定の解釈が **Subversion 1.14** では変更になりました。影響を受ける設定の詳細については、[アップストリームのドキュメント](#) を参照してください。

subversion:1:14 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 subversion:1.14
```

subversion:1.10 ストリームからアップグレードする場合は、[後続のストリームへの切り替え](#) を参照してください。

subversion モジュールストリームのサポート期間の詳細は、[Red Hat Enterprise Linux 8 Application Streams ライフサイクル](#) を参照してください。

([BZ#1844947](#))

新しいモジュールストリーム: **redis:6**

高度なキー値ストアの **Redis 6** のが、新しいモジュールストリーム **redis:6** として利用できるようになりました。

Redis 5 への主な変更点は、以下の通りです。

- **Redis** がすべてのチャンネルで SSL をサポートするようになりました。
- **Redis** が、コマンド呼び出しおよびキーパターンアクセスのためのユーザーパーミッションを定義するアクセス制御リスト (ACL) をサポートするようになりました。
- **Redis** が新しい **RESP3** プロトコルをサポートするようになりました。これにより、より多くのセマンティック応答が返されます。
- **Redis** がオプションでスレッドを使用して I/O を処理できるようになりました。
- **Redis** は、クライアント側のキー値のキャッシュにサーバー側のサポートを提供するようになりました。

- **Redis** のアクティブな期限切れサイクルが改善され、期限切れの鍵のエビクションが速くなりました。

Redis 6 には **Redis 5** との互換性があります (以下の後方互換性のない変更を除く)。

- セットキーが存在しない場合は、**SPOP <count>** コマンドは `null` を返しなくなりました。**Redis 6** では、コマンドは、このシナリオでは空のセットを返します。これは、引数 **0** で呼び出された場合と同様です。

redis:6 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 redis:6
```

redis:5 ストリームからアップグレードする場合は、[後続のストリームへの切り替え](#) を参照してください。

redis モジュールストリームのサポート期間の詳細は、[Red Hat Enterprise Linux 8 Application Streams ライフサイクル](#) を参照してください。

(BZ#1862063)

新しいモジュールストリーム: postgresql:13

RHEL 8.4 リリースでは、**PostgreSQL 13** が導入されました。これは、バージョン 12 から多くの新機能および機能強化が追加されています。以下は、主な変更点です。

- B-tree インデックスエントリーの重複によるパフォーマンスの向上
- 集約またはパーティション化されたテーブルを使用するクエリーのパフォーマンス向上
- 拡張された統計使用時のクエリー計画の改善
- インデックスの並列化された退避
- インクリメンタルソート

PostgreSQL 11 以降のアップストリームで利用できる、Just-In-Time (JIT) のコンパイルサポートは **postgresql:13** モジュールストリームでは提供されていないことに注意してください。

[PostgreSQL の使用](#) も参照してください。

postgresql:13 ストリームをインストールするには、以下を実行します。

```
# yum module install postgresql:13
```

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

postgresql モジュールストリームのサポート期間の詳細は、[Red Hat Enterprise Linux 8 Application Streams ライフサイクル](#) を参照してください。

(BZ#1855776)

新しいモジュールストリーム: mariadb:10.5

MariaDB 10.5 が、新しいモジュールストリーム **mariadb:10.5** として利用できるようになりました。以前利用できたバージョン 10.3 への主な機能強化は、以下のとおりです。

- MariaDB は、デフォルトで **unix_socket** 認証プラグインを使用するようになりました。このプラグインを使用すると、ローカルの Unix ソケットファイルを介して MariaDB に接続する際に、オペレーティングシステムの認証情報を使用できます。
- MariaDB は、サーバーを再起動しなくても SSL 証明書を再読み込みする新しい **FLUSH SSL** コマンドに対応します。
- MariaDB は、**mariadb-*** という名前のバイナリーと、**mariadb-*** バイナリーを指す **mysql*** シンボリックリンクを追加します。たとえば、**mariadb-admin**、**mariadb-access**、**mariadb-show** を指す、**mysqladmin**、**mysqlaccess**、**mysqlshow** symlink。
- MariaDB は、IPv6 アドレスを保存する新しい **INET6** データ型をサポートします。
- MariaDB は、Perl Compatible Regular Expressions (PCRE) ライブラリーバージョン 2 を使用するようになりました。
- 各ユーザーロールに合わせて、**SUPER** 特権が複数の特権に分割されました。その結果、一部のステートメントが必要な特権が変更されました。
- MariaDB は、新しいグローバル変数 **binlog_row_metadata** を追加して、システム変数およびステータス変数を追加して、ログに記録されるメタデータの量を制御します。
- **eq_range_index_dive_limit** 変数のデフォルト値が **0** から **200** に変更になりました。
- 最後の binlog イベントが接続されたレプリカに送信された後にのみシャットダウンするように、新しい **SHUTDOWN WAIT FOR ALL SLAVES** サーバーコマンドと新しい **mysqladmin shutdown --wait-for-all-slaves** オプションが追加されました。
- 並列レプリケーションでは、**slave_parallel_mode** 変数がデフォルトで **optimistic** になりました。

InnoDB ストレージエンジンには以下の変更が追加されました。

- InnoDB は、インスタント状態の **DROP COLUMN** 操作をサポートし、ユーザーが列の順序を変更できるようになりました。
- 変数のデフォルトが、**innodb_adaptive_hash_index** を **OFF** に、**innodb_checksum_algorithm** を **full_crc32** に変更になりました。
- 一部の InnoDB 変数が削除されるか、非推奨となっています。

MariaDB Galera Cluster がバージョン 4 にアップグレードされ、以下の主な変更点が加えられました。

- Galera は、無制限サイズのトランザクションの複製をサポートする新しいストリーミングレプリケーション機能を追加します。ストリーミングレプリケーションの実行時に、クラスターは小さなフラグメントでトランザクションを複製します。
- Galera がグローバルトランザクション ID (GTID) に完全に対応するようになりました。
- **/etc/my.cnf.d/galera.cnf** ファイルの **wsrep_on** オプションのデフォルト値が **1** から **0** に変更され、必要な追加オプションを設定せずにエンドユーザーが **wsrep** レプリケーションを開始できないようにします。

[MariaDB の使用](#) も参照してください。

mariadb:10.5 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 mariadb:10.5
```

mariadb:10.3 モジュールストリームからアップグレードする場合は、[MariaDB 10.3 から MariaDB 10.5 へのアップグレード](#) を参照してください。

mariadb モジュールストリームのサポート期間の詳細は、[Red Hat Enterprise Linux 8 Application Streams ライフサイクル](#) を参照してください。

(BZ#1855781)

MariaDB 10.5 は、PAM プラグインバージョン 2.0 を提供します。

MariaDB 10.5 は、PAM (Pluggable Authentication Modules) プラグインの新バージョンを追加します。PAM プラグインバージョン 2.0 は、個別の **setuid root** ヘルパーバイナリーを使用して PAM 認証を実行します。これにより、**MariaDB** が追加の PAM モジュールを使用できるようになります。

MariaDB 10.5 では、プラグ可能な認証モジュール (PAM) プラグインとその関連ファイルが新しいパッケージ **mariadb-pam** に移動しました。このパッケージには PAM プラグインバージョンの両方が含まれています。バージョン 2.0 はデフォルトで、バージョン 1.0 は **auth_pam_v1** 共有オブジェクトライブラリーとして利用できます。

MariaDB サーバーでは、デフォルトで **mariadb-pam** パッケージがインストールされないことに注意してください。**MariaDB 10.5** で PAM 認証プラグインを利用できるようにするには、**mariadb-pam** パッケージを手動でインストールします。

既知の問題 **MariaDB** では PAM プラグインバージョン 1.0 が機能しない も参照してください。

(BZ#1936842)

新しいパッケージ: mysql-selinux

RHEL 8.4 は、**MariaDB** および **MySQL** データベースのルールが含まれる SELinux モジュールを提供する新しい **mysql-selinux** パッケージを追加します。このパッケージは、デフォルトでデータベースサーバーを使用してインストールされます。モジュールの優先度は **200** に設定されます。

(BZ#1895021)

python-PyMySQL がバージョン 0.10.1 にリベースされました。

python-PyMySQL クライアントライブラリーを提供する **python-PyMySQL** パッケージが、バージョン 0.10.1 に更新されました。パッケージは、**python36**、**python38**、**python39** モジュールに含まれます。

以下は、主な変更点です。

- 今回の更新で、**ed25519** および **caching_sha2_password** 認証メカニズムに対応するようになりました。
- **python38** および **python39** モジュールに設定したデフォルトの文字は、**utf8mb4** で、アップストリームと一致します。**python36** モジュールは、このモジュールの以前のバージョンとの互換性を維持するために、デフォルトの **latin1** 文字セットを保持します。
- **python36** モジュールでは、**/usr/lib/python3.6/site-packages/pymysql/tests/** ディレクトリーは利用できなくなりました。

(BZ#1820628, BZ#1885641)

新しいパッケージ: python3-pyodbc

今回の更新で、**python3-pyodbc** パッケージが RHEL 8 に追加されました。**pyodbc** Python は、Open Database Connectivity (ODBC) データベースへのアクセスを提供します。このモジュールは Python DB API 2.0 仕様を実装し、サードパーティーの ODBC ドライバーと共に使用できます。たとえば、Performance Co-Pilot (**pcp**) を使用して SQL サーバーのパフォーマンスを監視できるようになりました。

(BZ#1881490)

新しいパッケージ: micropipenv

新しい **micropipenv** パッケージが利用できるようになりました。**Pipenv** および **Poetry** ロックファイルをサポートする **pip** パッケージインストーラーの軽量ラッパーを提供します。

micropipenv パッケージは AppStream リポジトリで配布され、互換性レベル 4 で提供されます。詳細は [Red Hat Enterprise Linux 8 アプリケーションの互換性ガイド](#) を参照してください。

(BZ#1849096)

新しいパッケージ: py3c-devel および py3c-docs

RHEL 8.4 では、新しい **py3c-devel** パッケージおよび **py3c-docs** パッケージが導入され、C 拡張の Python 3 への移植が簡素化されます。これらのパッケージには、詳細なガイドと、移植を容易にするためのマクロセットが含まれています。

py3c-devel および **py3c-docs** パッケージは、サポート対象外の [CodeReady Linux Builder \(CRB\)](#) リポジトリを介して配布されます。

(BZ#1841060)

httpd を設定するための強化された ProxyRemote ディレクティブ

Apache HTTP Server の **ProxyRemote** 設定ディレクティブは、オプションでユーザー名とパスワードの認証情報を取得するように強化されました。これらの認証情報は、HTTP **Basic** 認証を使用してリモートプロキシへの認証に使用されます。この機能は、**httpd 2.5** からバックポートされています。

(BZ#1869576)

非同期証明書は、**SSLProxyMachineCertificateFile** および **SSLProxyMachineCertificatePath** **httpd** 使用できます。

今回の更新により、Apache HTTP Server の **SSLProxyMachineCertificateFile** および **SSLProxyMachineCertificatePath** 設定ディレクティブを使用して、認証局 (CA) や中間証明書などの非エンド (リーフ以外の) 証明書を使用できるようになりました。Apache HTTP サーバーは、**SSLProxyMachineCertificateChainFile** ディレクティブで使用されるかのように、そのような証明書を信頼できる CA として処理するようになりました。以前のバージョンでは、**SSLProxyMachineCertificateFile** ディレクティブおよび **SSLProxyMachineCertificatePath** ディレクティブとともに非エンドの証明書が使用されると、**httpd** は設定エラーで起動できませんでした。

(BZ#1883648)

mod_security モジュールの新しい SecRemoteTimeout ディレクティブ

以前は、Apache HTTP Server の **mod_security** モジュールでリモートルールを取得するデフォルトのタイムアウトを変更できませんでした。今回の更新で、新しい **SecRemoteTimeout** 設定ディレクティブを使用して、カスタムタイムアウトを秒単位で設定できるようになりました。

タイムアウトに達すると、**httpd** がエラーメッセージ **Timeout was reached** で失敗するようになります。

した。このシナリオでは、設定ファイルの構文が有効であっても、**Syntax error** も含まれていることに注意してください。タイムアウト時の **httpd** の動作は、**SecRemoteRulesFailAction** 設定ディレクティブの値によって異なります (デフォルト値は **Abort** です)。

(BZ#1824859)

mod_fcgid モジュールが、FCGI サーバープロセスに最大 1024 個の環境変数を渡すことができるようになりました。

今回の更新で、Apache HTTP Server の **mod_fcgid** モジュールは、FCGI (FCGI) サーバープロセスに最大 1024 個の環境変数を渡すことができるようになりました。以前の 64 環境変数の制限により、FCGI サーバーで実行しているアプリケーションの誤動作が生じる可能性がありました。

(BZ#1876525)

perl-IO-String が AppStream リポジトリで利用可能になりました。

Perl **IO::String** モジュールを提供する **perl-IO-String** パッケージが、サポートされる AppStream リポジトリで配布されるようになりました。RHEL 8 の以前のリリースでは、**perl-IO-String** パッケージは、サポート対象外の CodeReady Linux Builder リポジトリで利用できていました。

(BZ#1890998)

新しいパッケージ: quota-devel

RHEL 8.4 では、**quota-devel** パッケージが導入され、**quota** Remote Procedure Call (RPC) サービスを実装するためのヘッダーファイルが提供されます。

quota-devel パッケージは、サポート対象外の [CodeReady Linux Builder \(CRB\) リポジトリ](#) で配布されることに注意してください。

(BZ#1868671)

4.12. コンパイラーおよび開発ツール

glibc ライブラリーが、最適化した共有ライブラリー実装を読み込む **glibc-hwcap** サブディレクトリーに対応

特定のアーキテクチャーでは、ハードウェアアップグレードにより、**glibc** が以前のハードウェア生成用に最適化されたライブラリーではなく、ベースラインの最適化でライブラリーを読み込むことができました。また、AMD CPU で実行する際に、最適化されたライブラリーはまったく読み込まれていませんでした。

今回の機能拡張により、**glibc** は、**glibc-hwcap** サブディレクトリーで最適化されたライブラリー実装の検索をサポートするようになりました。動的ローダーは、使用中の CPU とそのハードウェア機能に基づいて、サブディレクトリー内のライブラリーファイルをチェックします。この機能は、IBM Power Systems (リトルエンディアン)、IBM Z、64 ビット AMD、および Intel のアーキテクチャーで利用できます。

(BZ#1817513)

glibc 動的ローダーが、ランタイム時に選択した監査モジュールをアクティブに

以前は、**binutils** リンクエディター **ld** が、**--audit** オプションをサポートしてランタイム時にアクティブ化するモジュールを選択していましたが、**glibc** 動的ローダーが要求を無視していました。今回の更新で、**glibc** 動的ローダーが要求を無視せず、指定の監査モジュールを読み込むようになりました。その

結果、ラッパースクリプトを作成せず、同様のメカニズムを使用せずに、特定のプログラムに対して監査モジュールをアクティブにできます。

(BZ#1871385)

glibc で IBM POWER9 のパフォーマンスが向上しました。

今回の更新で、IBM POWER9 の **strlen**、**strcpy**、**stpcpy**、**rawmemchr** 関数の新しい実装が追加されました。その結果、IBM POWER9 ハードウェアでこの機能はより高速に実行するようになり、パフォーマンスが向上します。

(BZ#1871387)

IBM Z での memcpy および memset のパフォーマンスの最適化

今回の機能拡張により、IBM Z プロセッサの小規模な (< 64KiB) と大規模なデータコピーの両方を加速するために、**memcpy** および **memset** API のコアライブラリー実装が調整されました。その結果、インメモリーデータを使用するアプリケーションは、さまざまなワークロードでパフォーマンスが大幅に改善されました。

(BZ#1871395)

GCC が ARMv8.1 LSE アトミック命令に対応

今回の機能強化により、GCC コンパイラーは、ARMv8.1 仕様で追加されたアトミック命令 (LSE)、Large System Extensions (LSE) に対応するようになりました。この手順では、ARMv8.0 Load-Exclusive および Store-Exclusive 命令よりもマルチスレッドアプリケーションのパフォーマンスが向上します。

(BZ#1821994)

GCC が、特定の IBM Z システムでベクトル調整ヒントを生成するようになりました。

今回の更新で、GCC コンパイラーは、IBM z13 プロセッサのベクター負荷およびストア調整ヒントを出力できるようになりました。この機能拡張を使用するには、アセンブラーがこのようなヒントをサポートする必要があります。その結果、特定のベクター操作のパフォーマンスが向上します。

(BZ#1850498)

dyninst がバージョン 10.2.1 にリベースされました。

Dyninst バイナリー分析および変更ツールがバージョン 10.2.1 に更新されました。主なバグ修正と機能強化は、以下のとおりです。

- elfutils **debuginfod** クライアントライブラリーのサポート
- 並行バイナリーコード分析が改善されました。
- 大規模なバイナリーの分析および計測が改善されました。

(BZ#1892001)

elfutils がバージョン 0.182 にリベースされました。

elfutils パッケージがバージョン 0.182 に更新されました。主なバグ修正と機能強化は、以下のとおりです。

- **DW_CFA_AARCH64_negate_ra_state** 命令を認識します。PAC (Pointer Authentication Code) が有効になっていない場合は、**DW_CFA_AARCH64_negate_ra_state** を使用して、64

ビット ARM アーキテクチャーの PAC 用にコンパイルされるアンゴールドコードを使用できません。

- **elf_update** は、**SHF_COMPRESSED** フラグを設定したセクションで不正な **sh_addralign** 値を修正するようになりました。
- **debuginfod-client** が、ZSTD で圧縮されたカーネル ELF イメージに対応するようになりました。
- **debuginfod** にはより効率的なパッケージトラバースがあり、スキャン中のさまざまなエラーに対応します。ゲーミングプロセスはより表示可能で割り込み可能であり、さらに Prometheus メトリックを提供します。

(BZ#1875318)

SystemTap がバージョン 4.4 にリベースされました。

SystemTap 計測ツールがバージョン 4.4 に更新され、バグ修正および機能強化が複数追加されました。以下は、主な変更点です。

- ユーザー空間のプロープにパフォーマンスおよび安定性が改善されました。
- ユーザーは、IBM Power Systems のリトルエンディアンバリエーションである AMD64、Intel 64、IBM Z、およびアーキテクチャーで暗黙的なスレッドローカルストレージ変数にアクセスできるようになりました。
- 浮動小数点値処理の初期サポート。
- グローバル変数を使用するスクリプトの同時実行が改善されました。グローバル変数への同時アクセスを保護するのに必要なロックが最適化され、可能な最低限のリージョンまで超過されました。
- 述語とエピローグの両方を持つエイリアスを定義する新しい構文。
- 新しい **@probewrite** 述語。
- **syscall** の引数は、再度書き込み可能です。

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#1875341)

Valgrind が IBM z14 命令に対応

今回の更新で、Valgrind ツールスイートが IBM z14 プロセッサの命令に対応するようになりました。これにより、Valgrind ツールを使用して、z14 ベクトル命令とその他の z14 命令セットを使用してプログラムをデバッグできるようになりました。

(BZ#1504123)

cmake がバージョン 3.18.2 にリベースされました。

CMake ビルドシステムが、バージョン 3.11.4 からバージョン 3.18.2 にアップグレードされました。RHEL 8.4 では、**cmake-3.18.2-8.el8** パッケージとして利用できます。

バージョン 3.18.2 以降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 x.y.z)** コマンドを使用します。

新機能および非推奨の機能の詳細は、[CMake リリースノート](#) を参照してください。

[\(BZ#1816874\)](#)**libmpc がバージョン 1.1.0 にリベースされました。**

libmpc パッケージがバージョン 1.1.0 にリベースされ、以前のバージョンに対する機能強化およびバグ修正が複数追加されました。詳細は、[GNU MPC 1.1.0 リリースノート](#) を参照してください。

[\(BZ#1835193\)](#)

GCC Toolset 10 の更新

GCC Toolset 10 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 Stream として利用できます。

RHEL 8.4 で導入された主な変更点は、以下のとおりです。

- GCC コンパイラーがアップストリームバージョンに更新され、バグ修正が複数追加されました。
- **elfutils** がバージョン 0.182 に更新されました。
- Dyninst がバージョン 10.2.1 に更新されました。
- SystemTap がバージョン 4.4 に更新されました。

以下のツールおよびバージョンは、GCC Toolset 10 で利用できます。

ツール	バージョン
GCC	10.2.1
GDB	9.2
Valgrind	3.16.0
SystemTap	4.4
Dyninst	10.2.1
binutils	2.35
elfutils	0.182
dwz	0.12
make	4.2.1
strace	5.7
ltrace	0.7.91

ツール	バージョン
annobin	9.29

GCC Toolset 10 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-10
```

GCC Toolset 10 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-10 tool
```

GCC Toolset バージョン 10 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-10 bash
```

詳細は、[Using GCC Toolset](#) を参照してください。

GCC Toolset 10 コンポーネントが、以下のコンテナイメージ 2 つで利用可能になりました。

- GCC コンパイラー、GDB デバッガー、**make** 自動化ツールを含む **rhel8/gcc-toolset-10-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-10-perftools**

コンテナイメージをプルするには、`root` で以下のコマンドを実行します。

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

GCC Toolset 10 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。

コンテナイメージの詳細は、[GCC Toolset コンテナイメージの使用](#) を参照してください。

(BZ#1918055)

GCC Toolset 10: GCC が **bfloat16** に対応するようになりました。

GCC Toolset 10 では、GCC コンパイラーが ACLE Intrinsics による **bfloat16** 拡張に対応するようになりました。今回の機能拡張により、高パフォーマンスコンピューティングが提供されます。

(BZ#1656139)

GCC Toolset 10: GCC が Intel Sapphire Rapids プロセッサーで **ENQCMD** および **ENQCMLS** 命令に対応

GCC Toolset 10 では、GNU コンパイラーコレクション (GCC) が、**ENQCMD** および **ENQCMLS** 命令に対応するようになりました。これを使用して、デバイスにワーク記述子を自動的に送信できます。今回の機能拡張を適用するには、GCC に **-menqcmd** オプションを指定して実行します。

(BZ#1891998)

GCC Toolset 10: Dyninst がバージョン 10.2.1 にリベースされました。

GCC Toolset 10 では、Dyninst バイナリー分析および修正ツールがバージョン 10.2.1 に更新されました。主なバグ修正と機能強化は、以下のとおりです。

- elfutils **debuginfod** クライアントライブラリーのサポート
- 並行バイナリーコード分析が改善されました。
- 大規模なバイナリーの分析および計測が改善されました。

([BZ#1892007](#))

GCC Toolset 10: elfutils がバージョン 0.182 にリベースされました。

GCC Toolset 10 では、**elfutils** パッケージがバージョン 0.182 に更新されました。主なバグ修正と機能強化は、以下のとおりです。

- **DW_CFA_AARCH64_negate_ra_state** 命令を認識します。PAC (Pointer Authentication Code) が有効になっていない場合は、**DW_CFA_AARCH64_negate_ra_state** を使用して、64 ビット ARM アーキテクチャーの PAC 用にコンパイルされるアンゴールドコードを使用できません。
- **elf_update** は、**SHF_COMPRESSED** フラグを設定したセクションで不正な **sh_addralign** 値を修正するようになりました。
- **debuginfod-client** が、ZSTD で圧縮されたカーネル ELF イメージに対応するようになりました。
- **debuginfod** にはより効率的なパッケージトラバースがあり、スキャン中のさまざまなエラーに対応します。ゲーミングプロセスはより表示可能で割り込み可能であり、さらに Prometheus メトリックを提供します。

([BZ#1879758](#))

Go Toolset がバージョン 1.15.7 にリベースされました。

Go Toolset が 1.15.7 にアップグレードされました。主な機能強化は、次のとおりです。

- 新たに実装されたオブジェクトファイル形式および内部フェーズの同時実行が増えると、リンクが速くなり、必要なメモリーが少なくなりました。今回の機能拡張により、内部リンクがデフォルトになりました。この設定を無効にするには、コンパイラーフラグ **-ldflags=-linkmode=external** を使用します。
- ワーストケースのレイテンシーを含め、コア数が多い場合の小さなオブジェクトの割り当てが改善されました。
- **Subject Alternative Names** が指定されていない場合に X.509 証明書の **CommonName** フィールドをホスト名として扱うことが、デフォルトで無効になりました。これを有効にするには、値 **x509ignoreCN=0** を **GODEBUG** 環境変数に追加します。
- **GOPROXY** がエラーを返すプロキシのスキップをサポートするようになりました。
- Go に新しいパッケージ **time/tzdata** が追加されました。これにより、タイムゾーンデータベースがローカルシステムで利用できない場合でも、タイムゾーンデータベースをプログラムに埋め込みできます。

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

(BZ#1870531)

Rust Toolset がバージョン 1.49.0 にリベースされました。

Rust Toolset は、バージョン 1.49.0 に更新されました。以下は、主な変更点です。

- rustdoc のページ項目のパスを使用して、rustdoc でリンクできるようになりました。
- rust テストフレームワークは、スレッド出力を非表示にするようになりました。失敗したテストの出力はターミナルに表示されます。
- **[T; N]: TryFrom<Vec<T>>** を使用して、ベクトルを任意の長さの配列に切り替えることができます。
- **slice::select_nth_unstable** を使用して、順序のパーティションを作成できるようになりました。この関数は、以下のバリエーションでも利用できます。
 - **slice::select_nth_unstable_by** は、コンパレーター関数を提供します。
 - **slice::select_nth_unstable_by_key** は、キー抽出関数を提供します。
- union フィールドのタイプに **ManuallyDrop** を使用できるようになりました。 **impl Drop for Union** を使用して、既存のユニオンに Drop トレイトを追加することも可能です。これにより、特定のフィールドを手動でドロップする必要があるユニーディネートを定義できます。
- Rust Toolset のコンテナイメージが非推奨になり、Rust Toolset は Universal Base Images (UBI) リポジトリに追加されました。

詳細は、[Rust Toolset の使用](#) を参照してください。

(BZ#1896712)

LLVM Toolset がバージョン 11.0.0 にリベースされました。

LLVM Toolset がバージョン 11.0.0 にアップグレードされました。以下は、主な変更点です。

- **-fstack-clash-protection** コマンドラインオプションのサポートが、AMD および Intel 64 ビットアーキテクチャー、IBM Power Systems、リトルエンディアン、および IBM Z に追加されました。新しいコンパイラフラグは、各スタックページを自動的にチェックしてスタッククラッシュ攻撃から保護します。
- 新しいコンパイラフラグ **ffp-exception-behavior={ignore,maytrap,strict}** は、浮動小数点例外動作の指定を有効にします。デフォルト設定は **ignore** です。
- 新しいコンパイラフラグ **ffp-model={precise,strict,fast}** により、単一の目的の浮動小数点オプションを簡素化できます。デフォルト設定は **precise** になります。
- 新しいコンパイラフラグ **-fno-common** がデフォルトで有効になりました。今回の機能強化により、複数の翻訳ユニットで 10 つの変数定義を使用して C で書かれたコードが、複数の有効なリンカーエラーをトリガーするようになりました。この設定を無効にするには、**-fcommon** フラグを使用します。
- LLVM Toolset のコンテナイメージが非推奨になり、LLVM Toolset は Universal Base Images (UBI) リポジトリに追加されました。

詳細は、[LLVM Toolset の使用](#) を参照してください。

(BZ#1892716)

pcp がバージョン 5.2.5 にリベースされました。

pcp パッケージが、バージョン 5.2.5 にアップグレードされました。以下は、主な変更点です。

- SQL Server メトリックはセキュアな接続でサポートされています。
- プロセスごとのネットワークメトリックを使用する **eBPF/BCC** netproc モジュール。
- **hv_24x7 core-level** および **hv_gpcci** イベントメトリックに対する **pmdaperfevent(1)** のサポート。
- NVM Express ディスクに対する新しい Linux プロセスアカウントメトリック、Linux ペイン、Linux カーネルソケットメトリック、Linux マルチパス TCP メトリック、および ZRAM メトリック、および S.M.A.R.T. メトリックのサポートが追加されました。
- システムおよびプロセスメトリックを視覚化する新しい **pcp-htop(1)** ユーティリティ。
- **pmrep/pcp2xxx** 設定を生成する新しい **pmrepconf(1)** ユーティリティ。
- **pmie** サービスを制御する新しい **pmiectl(1)** ユーティリティ。
- **pmlogger** サービスを制御する新しい **pmlogctl(1)** ユーティリティ。
- ログ文字列のメトリックを書き込む新しい **pmlogpaste(1)** ユーティリティ。
- アカウンティング統計とプロセスごとのネットワーク統計レポートを処理する新しい **pcp-atop(1)** ユーティリティ。
- 関数、言語拡張、および REST API をクエリーする **pmseries(1)** ユーティリティが追加されました。
- OOM による強制終了およびソケット接続の飽和を検出する新しい **pmie(1)** ルール。
- **pcp-atopsar(1)**、**pcp-free(1)**、**pcp-dstat(1)**、**pmlogger(1)**、**pmchart(1)** ユーティリティのバグ修正。
- コンテキストごとの派生メトリックに対する REST API および C API のサポート。
- OpenMetrics メトリックのメタデータ (ユニット、セマンティクス) が改善されました。
- **/var** ファイルシステムレイアウトを広範囲にインストールしていました。

([BZ#1854035](#))

grafana-pcp の Vector データソース用に中央 **pmproxy** 経由でリモートホストにアクセスする

一部の環境では、ネットワークポリシーでは、Dashboard ビューアーのブラウザから監視するホストに直接接続が許可されません。今回の更新で、中央 **pmproxy** に接続するために **hostspec** をカスタマイズできるようになりました。これにより、個々のホストにリクエストを転送できるようになりました。

([BZ#1845592](#))

grafana がバージョン 7.3.6 にリベースされました。

grafana パッケージがバージョン 7.3.6 にアップグレードされました。以下は、主な変更点です。

- 新しいパネルエディターと新しいデータ変換機能

- タイムゾーンのサポートの改善
- デフォルトのプロビジョニングパスが、`/usr/share/grafana/conf/provisioning` から `/etc/grafana/provisioning` ディレクトリに変更になりました。この設定は、`/etc/grafana/grafana.ini` 設定ファイルで設定できます。

詳細は、[What's New in Grafana v7.0](#)、[What's New in Grafana v7.1](#)、[What's New in Grafana v7.2](#)、[What's New in Grafana v7.3](#) を参照してください。

(BZ#1850471)

grafana-pcp がバージョン 3.0.2 にリベースされました。

grafana-pcp パッケージがバージョン 3.0.2 にアップグレードされました。以下は、主な変更点です。

- Redis:
 - Grafana でのアラートの作成をサポートします。
 - パフォーマンス上の理由から、Grafana 変数のクエリーの `label_values(metric, label)` の使用が非推奨になりました。`label_values(label)` クエリーは引き続きサポートされます。
- Vector:
 - 派生メトリックをサポートします。これにより、クエリー内で算算演算子および統計関数を使用できます。詳細は、man ページの `pmRegisterDerived(3)` を参照してください。
 - 設定可能な `hostspec`: 中央の `pmproxy` 経由でリモート Performance Metrics Collector Daemon (PMCD) にアクセスできます。
 - パネルのユニットを自動的に設定します。
- ダッシュボード:
 - パフォーマンスの問題を検出し、Utilization Saturation and Errors (USE) メソッドを使用して、チェックリストダッシュボードで考えられる解決策を表示します。
 - **CGroups v2** を使用した新しい MS SQL サーバーダッシュボード、**eBPF/BCC** ダッシュボード、およびコンテナー概要ダッシュボード。
 - すべてのダッシュボードは **データソース** 設定ページの **Dashboards** タブに配置されるようになり、自動的にインポートされません。

アップグレードの注意事項:

Grafana 設定ファイルを更新します。

1. Grafana 設定ファイル `/etc/grafana/grafana.ini` を編集し、以下のオプションが設定されていることを確認します。

```
allow_loading_unsigned_plugins = pcp-redis-datasource
```

2. Grafana サーバーを再起動します。

```
# systemctl restart grafana-server
```

(BZ#1854093)

PCP の SQL Server メトリックにアクセスするための Active Directory 認証

今回の更新で、システム管理者は、Active Directory AD) 認証を使用して SQL Server メトリックに安全に接続するように `pmdamssql(1)` を設定できるようになりました。

([BZ#1847808](#))

grafana-container がバージョン 7.3.6 にリベースされました。

`rhel8/grafana` コンテナイメージは Grafana を提供します。Grafana は、メトリックダッシュボードを使用したオープンソースユーティリティで、Graphite、Elasticsearch、OpenTSDB、Prometheus、InfluxDB、および Performance Co-Pilot(PCP) 用のグラフィックエディターです。`grafana-container` パッケージがバージョン 7.3.6 にアップグレードされました。以下は、主な変更点です。

- `grafana` パッケージがバージョン 7.3.6 に更新されました。
- `grafana-pcp` パッケージがバージョン 3.0.2 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/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#1916154](#))

pcp-container がバージョン 5.2.5 にリベースされました。

`rhel8/pcp` コンテナイメージは、システムパフォーマンス分析ツールキットである Performance Co-Pilot を提供します。`pcp-container` パッケージがバージョン 5.2.5 にアップグレードされました。以下は、主な変更点です。

- `pcp` パッケージが、バージョン 5.2.5 に更新されました。
- コンテナ内で起動する PCP サービスのコンマ区切りのリストを指定する新しい `PCP_SERVICES` 環境変数が導入されました。

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#1916155](#))

JDK Mission Control がバージョン 8.0.0 にリベースされました。

`jmc:rhel8` モジュールストリームによって提供される HotSpot JVM の JMC (JDK Mission Control) プロファイルャーが、バージョン 8.0.0 にアップグレードされました。主な機能強化は、次のとおりです。

- クラスによるメモリー使用量を視覚的に把握するために、`Treemap` ビューアーが `JOverflow` プラグインに追加されました。
- `Threads` グラフは、より多くのフィルタリングおよびズームオプションで強化されました。
- JDK Mission Control は、LZ4 アルゴリズムを使用して圧縮した JDK Flight Recorder を開くためのサポートが追加されました。

- 新規列が **Memory** および **TLAB** に追加され、割り当て不足の領域を特定しやすくなります。
- スタックトレースの視覚化を改善するために、**Graph** ビューが追加されました。
- **Percentage** の列がヒストグラムテーブルに追加されました。

RHEL 8 で JMC を実行するには、JDK バージョン 8 以降を実行する必要があります。ターゲット Java アプリケーションは、最低でも OpenJDK バージョン 8 で実行する必要があります。これにより、JMC が JDK Flight Recorder 機能にアクセスできるようになります。

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

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

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

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

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

(BZ#1919283)

4.13. ID 管理

Identity Management をより包括的に

Red Hat では、意識的な言語の使用に取り組んでいます。これについては、[多様性を受け入れるオープンソースの強化](#) で詳細を参照してください。

Identity Management では、以下のような用語を置き換えています。

- ブラックリスト から ブロックリスト
- ホワイトリスト から 許可リスト
- スレーブからセカンダリー
- 単語 マスター は、コンテキストに応じて、より正確な言語に置き換えられます。
 - マスターからIdM サーバー
 - CA 更新マスター から CA 更新サーバー
 - CRL マスター から CRL パブリッシャーサーバー
 - マルチマスターからマルチサプライヤー

(JIRA:RHELPLAN-73418)

dsidm ユーティリティーがエントリーの名前変更および移動に対応

今回の機能拡張により、**dsidm** ユーティリティーを使用して、Directory Server でユーザー、グループ、POSIX グループ、ロール、および組織単位 (OU) の名前を変更して移動できるようになりました。詳細と例は、Directory Server 管理ガイドの [ユーザー、グループ、POSIX グループ、および OU の名前](#)

[変更](#) を参照してください。

(BZ#1859218)

IdM でのサブ CA の削除

今回の機能拡張により、**ipa ca-del** コマンドを実行して Sub-CA を無効にしていない場合は、サブ CA を削除できないので無効にする必要があるとのエラーが表示されます。最初に **ipa ca-disable** コマンドを実行してサブ CA を無効にしてから、**ipa ca-del** コマンドを使用してこれを削除します。

IdM CA を無効化または削除できない点に留意してください。

(JIRA:RHELPLAN-63081)

IdM が新しい Ansible 管理ロールおよびモジュールに対応

RHEL 8.4 は、Identity Management(IdM) でロールベースのアクセス制御 (RBAC) を自動的に管理する Ansible モジュール、IdM サーバーのバックアップおよび復元用の Ansible ロール、およびロケーション管理用の Ansible モジュールを提供します。

- **ipapermission** モジュールを使用して、IdM RBAC でパーミッションおよびパーミッションメンバーを作成、変更、および削除することができます。
- **ipaprivilege** モジュールを使用して、IdM RBAC で特権および権限メンバーを作成、変更、および削除できます。
- **iparole** モジュールを使用して、IdM RBAC でロールおよびロールメンバーを作成、変更、および削除できます。
- **ipadelegation** モジュールを使用して、IdM RBAC のユーザーに対するパーミッションを委譲できます。
- **ipaselfservice** モジュールを使用して、IdM でセルフサービスアクセスルールを作成、変更、および削除できます。
- **ipabackup** ロールを使用して、IdM サーバーのバックアップを作成し、コピーし、IdM サーバーをローカルまたはコントロールノードから復元できます。
- **ipalocation** モジュールを使用して、データセンターラックなど、ホストの物理的な場所の有無を確認できます。

(JIRA:RHELPLAN-72660)

FIPS モードの IdM が、AD を使用したフォレスト間の信頼に対応

この機能強化により、管理者は FIPS モードが有効な IdM ドメインと Active Directory (AD) ドメインの間でフォレスト間の信頼を確立できるようになりました。IdM で FIPS モードが有効な場合に、共有シークレットを使用して信頼を確立できないことに注意してください。[FIPS コンプライアンス](#) を参照してください。

(JIRA:RHELPLAN-58629)

AD ユーザーは、既知の UPN 接尾辞に属する UPN 接尾辞が付いた IdM にログイン可能に

以前では、Active Directory (AD) ユーザーは、既知の UPN 接尾辞 (例: **ad-example.com**) のサブドメインである UPN (Universal Principal Name) (**sub1.ad-example.com**) で、IdM (Identity Management) にログインできませんでした。これは、内部 Samba プロセスがトポロジー名 (TLN) の重複としてサブド

メインをフィルタリングしていたためです。今回の更新により、既知の UPN 接尾辞に属するかどうかをテストして UPN が検証されるようになりました。これにより、上記のシナリオで下位の UPN 接尾辞を使用してログインできるようになりました。

(BZ#1891056)

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** ライブラリー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。

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。一貫した動作を確認するには、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。

(BZ#1340463)

Active Directory のサイト検出プロセスの改善

SSSD サービスは、接続レス LDAP (CLDAP) と複数のドメインコントローラーで並行して Active Directory サイトを検出し、一部のドメインコントローラーに到達できない状況でサイト検出を加速します。以前のバージョンでは、サイト検出が順次実行され、ドメインコントローラーに到達できない場合、タイムアウトが発生し、SSSD がオフラインになっていました。

(BZ#1819012)

スループットを向上させるために、**nsslapd-nagle** のデフォルト値がオフになっている

以前は、**cn=config** エントリーの **nsslapd-nagle** パラメーターがデフォルトで有効になっていました。その結果、Directory Server は多数の **setsocketopt** システムコールを実行し、サーバーの速度を低下させていました。今回の更新により、**nsslapd-nagle** のデフォルト値が **off** に変更になりました。その結果、Directory Server が実行する **setsocketopt** システムコールの数が減り、1 秒あたりにより多くの操作を処理できるようになります。

(BZ#1996076)

sssd.conf ファイルの [domain] セクションで SSSD ドメインを有効または無効にする

今回の更新で、**sssd.conf** ファイルの該当する [domain] セクションを変更することで、SSSD ドメインを有効または無効にできるようになりました。

以前は、SSSD 設定にスタンドアロンドメインが含まれている場合、**sssd.conf** ファイルの **[sssd]** セクションの **domains** オプションを変更する必要がありました。今回の更新で、ドメイン設定の **enabled=** オプションを true または false に設定できるようになりました。

- **enabled** オプションを true に設定すると、**sssd.conf** ファイルの **[sssd]** セクションの **domains** オプションの下に記載されていない場合でも、ドメインが有効になります。
- **enabled** オプションを false に設定すると、**sssd.conf** ファイルの **[sssd]** セクションの **domains** オプションの下に記載されていない場合でも、ドメインが有効になります。
- **enabled** なオプションが設定されていないと、**sssd.conf** の **[sssd]** セクションの **domains** オプションの設定が使用されます。

(BZ#1884196)

最大オフラインタイムアウトを手動で制御するオプションが追加されました。

offline_timeout の期間は、SSSD によるオンラインに戻るまでの再試行の間隔を決定します。以前のバージョンでは、この間隔の最大許容値は 3600 秒にハードコーディングされていました。これは一般的な用途に適していますが、環境を高速または遅らば問題が発生していました。

今回の更新で、**offline_timeout_max** オプションが各間隔の最大長さを手動で制御できるようになりました。これにより、SSSD でのサーバー動作を追跡できるようになりました。

この値は、**offline_timeout** パラメーターの値に相関して設定する必要があることに注意してください。値が 0 の場合は、インクリメント動作が無効になります。

(BZ#1884213)

SSSD セッションの録画設定で、**scope=all** が設定された **exclude_users** および **exclude_groups** のサポート

Red Hat Enterprise 8.4 では、グループまたはユーザーのセッションの録画を定義する新しい SSSD オプションを利用できるようになりました。

1. **exclude_users**
録画から除外されるユーザーのコンマ区切りリスト。**scope=all** 設定オプションと併用する場合にのみ適用されます。
2. **exclude_groups**
録画から除外されるメンバーであるグループのコンマ区切りリスト。**scope=all** 設定オプションにのみ適用されます。

詳細は、man ページの **sssd-session-recording** を参照してください。

(BZ#1784459)

samba がバージョン 4.13.2 にリベースされました。

samba パッケージがアップストリームバージョン 4.13.2 にアップグレードされ、以前のバージョンに対するバグ修正や機能強化が数多く追加されました。

- 認証されていないユーザーが **netlogon** プロトコルを使用してドメインを引き継ぐセキュリティの問題を回避するには、Samba サーバーが **server schannel** パラメーターのデフォルト値 (**yes**) を使用することを確認してください。**testparm -v | grep 'server schannel'** コマンドを使用して確認します。詳細は [CVE-2020-1472](#) を参照してください。
- Samba "wide links" 機能が VFS モジュールに変換されました。

- [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。](#)
- FIPS モードが有効な RHEL で Samba を使用できるようになりました。FIPS モードの制限により、以下を行います。
 - RC4 暗号化がブロックされているため、NT LAN Manager (NTLM) 認証は使用できません。
 - デフォルトでは、Samba クライアントユーティリティーは AES 暗号による Kerberos 認証を使用します。
 - Samba は、AES 暗号化を使用する Kerberos 認証を使用する Active Directory (AD) または Red Hat Identity Management (IdM) 環境でのみ、ドメインメンバーとして使用できます。Red Hat は、IdM がバックグラウンドで使用するプライマリードメインコントローラー (PDC) 機能のサポートを継続することに留意してください。
- サーバーメッセージブロックバージョン 1 (SMB1) プロトコルでのみ使用できるセキュアでない認証方法に対する以下のパラメーターが非推奨になりました。
 - **client plaintext auth**
 - **client NTLMv2 auth**
 - **client lanman auth**
 - **client use spnego**
- Samba で使用すると、GlusterFS の write-behind パフォーマンス変換の問題が修正され、データ破損を回避できるようになりました。
- ランタイムの最小サポートが Python 3.6 になりました。
- 非推奨の **ldap ssl ads** パラメーターが削除されました。

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

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

([BZ#1878109](#))

SSSD を使用したパスワードなしの sudo 認証用の新しい GSSAPI PAM モジュール

新しい **pam_sss_gss.so** Pluggable Authentication Module (PAM) を使用して、System Security Services Daemon (SSSD) を設定して、Generic Security Service Application Programming Interface (GSSAPI) で PAM 対応サービスに対してユーザーを認証できます。

たとえば、このモジュールを、Kerberos チケットを使用したパスワードなしの **sudo** 認証に使用できます。IdM 環境の追加のセキュリティのために、スマートカードやワンタイムパスワードで認証したユーザーなど、チケットで特定の認証インジケーターを持つユーザーにのみアクセスを付与するように SSSD を設定できます。

詳細は [IdM クライアントでの IdM ユーザーへの sudo アクセスの付与](#) を参照してください。

([BZ#1893698](#))

Directory Server がバージョン 1.4.3.16 にリベースされました。

389-ds-base パッケージがアップストリームバージョン 1.4.3.16 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。主な変更点のリストは、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://www.port389.org/docs/389ds/releases/release-1-4-3-16.html>
- <https://www.port389.org/docs/389ds/releases/release-1-4-3-15.html>
- <https://www.port389.org/docs/389ds/releases/release-1-4-3-14.html>
- <https://www.port389.org/docs/389ds/releases/release-1-4-3-13.html>
- <https://www.port389.org/docs/389ds/releases/release-1-4-3-12.html>
- <https://www.port389.org/docs/389ds/releases/release-1-4-3-11.html>
- <https://www.port389.org/docs/389ds/releases/release-1-4-3-10.html>
- <https://www.port389.org/docs/389ds/releases/release-1-4-3-9.html>

(BZ#1862529)

Directory Server が **RESULT** エントリーで作業および操作時間のログを記録するようになりました。

今回の更新で、Directory Server は、`/var/log/dirsrv/slapd-<instance_name>/access` ファイルの **RESULT** エントリーに追加の時間値をログ記録するようになりました。

- **wtime** の値は、操作が作業キューからワーカースレッドに移動するのにかかった時間を示します。
- **optime** の値は、ワーカースレッドが操作を開始した後に実際に操作が完了するのにかかった時間を示します。

新たに追加されたこの値で、Directory Server が読み込みやプロセス操作を処理する方法などの情報が追加で提供されます。

詳細は、Red Hat Directory Server の設定、コマンド、およびファイルリファレンスの [アクセスログリファレンス](#) を参照してください。

(BZ#1850275)

Directory Server が、インデックス化されていない内部検索を拒否するようになりました。

今回の機能拡張により、**nsslapd-require-internalop-index** パラメーターが **cn=<database_name>,cn=ldbm** データベース、**cn=plugins,cn=config** エントリーに追加されました。プラグインがデータを変更すると、データベースに書き込みロックがあります。大規模なデータベースで、プラグインがインデックス化されていない検索を実行すると、プラグインがすべてのデータベースロックを使用することがあります。これにより、データベースが破損したり、サーバーが応答しなくなります。この問題を回避するには、**nsslapd-require-internalop-index** パラメーターを有効にして、インデックス化されていない内部検索を拒否することができるようになりました。

(BZ#1851975)

4.14. デスクトップ

GNOME で、応答しないアプリケーションタイムアウトを設定できます。

GNOME は、すべてのアプリケーションに定期的にシグナルを送信し、アプリケーションが応答しないかどうかを検出します。GNOME が応答しないアプリケーションを検出すると、アプリケーションを停止するか待機するかを尋ねるアプリケーションウィンドウ上にダイアログが表示されます。

特定のアプリケーションは、時間内にシグナルに応答しません。これにより、GNOME は、アプリケーションが適切に動作している場合でもダイアログを表示します。

今回の更新により、シグナルの間隔を設定できるようになりました。この設定は、GSettings キー **org.gnome.mutter.check-alive-timeout** に保存されます。応答しないアプリケーション検出を完全に無効にするには、キーを 0 に設定します。

GSettings キーの設定に関する詳細は、[コマンドラインで GSettings キーの使用](#) を参照してください。

(BZ#1886034)

4.15. グラフィックインフラストラクチャー

Intel Tiger Lake GPU に対応

今回のリリースで、統合グラフィックスを備えた Intel Tiger Lake CPU マイクロアーキテクチャーのサポートが追加されました。これには、以下の CPU モデルを含む Intel UHD グラフィックと Intel Xe 統合 GPU が含まれます。

- Intel Core i7-1160G7
- Intel Core i7-1185G7
- Intel Core i7-1165G7
- Intel Core i7-1165G7
- Intel Core i7-1185G7E
- Intel Core i7-1185GRE
- Intel Core i7-11375H
- Intel Core i7-11370H
- Intel Core i7-1180G7
- Intel Core i5-1130G7
- Intel Core i5-1135G7
- Intel Core i5-1135G7
- Intel Core i5-1145G7E
- Intel Core i5-1145GRE
- Intel Core i5-11300H
- Intel Core i5-1145G7
- Intel Core i5-1140G7

- Intel Core i3-1115G4
- Intel Core i3-1115G4
- Intel Core i3-1110G4
- Intel Core i3-1115GRE
- Intel Core i3-1115G4E
- Intel Core i3-1125G4
- Intel Core i3-1125G4
- Intel Core i3-1120G4
- Intel Pentium Gold 7505
- Intel Celeron 6305
- Intel Celeron 6305E

Tiger Lake GPU サポートを有効にするために、**i915.alpha_support=1** または **i915.force_probe=*** カーネルオプションを設定する必要がなくなりました。

(BZ#1882620)

11 世代コアマイクロプロセッサを使用する Intel GPU がサポートされるようになりました。

今回のリリースで、Xe gen 12 統合グラフィックで 11 世代コア CPU アーキテクチャー (旧名称 **Rocket Lake**) のサポートが追加されました。これは、以下の CPU モデルにあります。

- Intel Core i9-11900KF
- Intel Core i9-11900K
- Intel Core i9-11900
- Intel Core i9-11900F
- Intel Core i9-11900T
- Intel Core i7-11700K
- Intel Core i7-11700KF
- Intel Core i7-11700T
- Intel Core i7-11700
- Intel Core i7-11700F
- Intel Core i5-11500T
- Intel Core i5-11600
- Intel Core i5-11600K

- Intel Core i5-11600KF
- Intel Core i5-11500
- Intel Core i5-11600T
- Intel Core i5-11400
- Intel Core i5-11400F
- Intel Core i5-11400T

(BZ#1784246, BZ#1784247, BZ#1937558)

NVIDIA Ampere がサポートされるようになりました。

今回のリリースで、GA102 または GA104 チップセットを使用する Nvidia Ampere GPU のサポートが追加されました。これには、以下の GPU モデルが含まれます。

- GeForce RTX 3060 Ti
- GeForce RTX 3070
- GeForce RTX 3080
- GeForce RTX 3090
- RTX A4000
- RTX A5000
- RTX A6000
- Nvidia A40

nouveau グラフィックドライバーでは、まだ Nvidia Ampere ファミリーの 3D アクセラレーションをサポートしていません。

(BZ#1916583)

さまざまな更新後の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最新のアップストリームバージョンに更新されました。

- Matrox **mgag200** ドライバー
- Aspeed **ast** ドライバー

(JIRA:RHELPLAN-72994, BZ#1854354, BZ#1854367)

4.16. WEB コンソール

ソフトウェアの更新ページは、必要な再起動を確認する

今回の更新で、RHEL Web コンソールのソフトウェアの更新ページは、インストール後に更新を有効にするために、一部のサービスまたは実行中のプロセスのみを再起動するだけで十分かどうかを確認します。このような場合、これによりマシンを再起動する必要がなくなります。

(JIRA:RHELPLAN-59941)

Web コンソールでのグラフィカルパフォーマンス分析

今回の更新で、システムグラフページが、マシンのパフォーマンスを分析するための新規の専用ページに置き換えられました。パフォーマンスメトリックを表示するには、**Overview** ページから **View details and history** をクリックします。これは、Utilization Saturation および Errors (USE) メソッドに基づく現在のメトリックおよび履歴イベントを表示します。

(JIRA:RHELPLAN-59938)

Web コンソールは、SSH 鍵の設定を支援します。

以前のバージョンでは、Web コンソールでは、ログイン時にリモート接続のパスワードを再使用する際に、初期ログインパスワードを使用してリモートホストにログインできました。このオプションが削除されました。Web コンソールの代わりに、リモートホストへの自動およびパスワードなしログインを必要とするユーザーの SSH キーの設定が容易になります。

詳細は、[Web コンソールでリモートシステムの管理](#) を参照してください。

(JIRA:RHELPLAN-59950)

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

Logging ロール設定に追加された RELP セキュアなトランスポートサポート

信頼できるイベントロギングプロトコル RELP は、**rsyslog** サーバー間でログメッセージを転送および受信する、安全で信頼できるプロトコルです。今回の機能強化により、**rsyslog** サーバーは、RELP プロトコルを使用してログメッセージを転送および受信できるため、今回の機能強化により、管理者は **rsyslog** ユーザーから高い要求を伴う便利なプロトコルである RELP を活用できるようになりました。

(BZ#1889484)

SSH クライアントの RHEL システムロールに対応

以前は、サーバーおよびクライアントに対して一貫した安定した方法で RHEL SSH を設定するベンダー対応自動化ツールはありませんでした。今回の機能拡張により、RHEL システムロールを使用して、オペレーティングシステムのバージョンとは独立して、システムかつ統一された方法で SSH クライアントを設定できるようになりました。

(BZ#1893712)

従来の RHEL システムロール形式の代替: Ansible コレクション

RHEL 8.4 では、従来の RHEL システムロール形式のオプションとして利用できる、コレクション形式で RHEL システムロールが導入されました。

今回の更新で、namespace およびコレクション名で設定される完全修飾コレクション名 (FQCN) の概念が導入されました。たとえば、カーネルロールの完全修飾名は **redhat.rhel_system_roles.kernel_settings** です。

- 名前空間とコレクション名の組み合わせは、オブジェクトが一意であることを保証します。
- 名前空間とコレクション名の組み合わせにより、競合なしにオブジェクトがコレクションと名前空間の間で共有されるようになります。

RPM パッケージを使用したコレクションをインストールします。Playbook を実行するホストに **python3-jmespath** がインストールされていることを確認します。

```
# yum install rhel-system-roles
```

RPM パッケージには、レガシーの Ansible ロール形式と、新しい Ansible Collection 形式の両方にロールが含まれます。たとえば、network ロールを使用するには、以下の手順を実施します。

レガシー形式:

```
---
- hosts: all
  roles:
  rhel-system-roles.network
```

コレクション形式:

```
---
- hosts: all
  roles:
  redhat.rhel_system_roles.network
```

Automation Hub を使用しており、Automation Hub でホストされるシステムロールコレクションをインストールする場合は、以下のコマンドを入力します。

```
$ ansible-galaxy collection install redhat.rhel_system_roles
```

その後、前述したように、コレクションの形式でロールを使用できます。そのためには、ansible-galaxy コマンドで Ansible Galaxy の代わりに Automation Hub を使用するようにシステムを設定する必要があります。詳細は、[Ansible Galaxy の代わりに Automation Hub を使用するように ansible-galaxy クライアントを設定する](#) を参照してください。

([BZ#1893906](#))

Metrics ロールは、PCP を使用した SQL サーバーのメトリックコレクションの設定および有効化をサポートします。

metrics RHEL システムロールは、SQL Server の **mssql** を Performance Co-Pilot (**pcp**) で接続できるようになりました。SQL Server は、Microsoft からの汎用リレーショナルデータベースです。実行すると、SQL Server は実行中の操作に関する内部統計を更新します。これらの統計には SQL クエリーを使用してアクセスすることができますが、システムおよびデータベース管理者はパフォーマンス分析タスクを記録、レポート、視覚化できるようにシステムおよびデータベース管理者が重要です。今回の機能強化により、メトリック RHEL システムロールを使用して、**mssql** メトリックの記録、レポート、および可視化機能を提供する Performance Co-Pilot (**pcp**) で SQL サーバーである **mssql** を自動化できるようになりました。

([BZ#1893908](#))

Metrics RHEL システムロールで利用可能な exporting-metric-data-to-elasticsearch 機能

Elasticsearch は、一般的で、スケーラブルにスケーラブルなエンジンです。今回の機能拡張により、Metrics RHEL システムロールから Elasticsearch にメトリック値をエクスポートすることにより、ユーザーはグラフィカルインターフェイス、REST API 経由でなど、Elasticsearch インターフェイス経由でメトリックにアクセスできるようになりました。その結果、ユーザーはこれらの Elasticsearch インターフェイスを使用してパフォーマンスの問題の診断や容量計画、ベンチマークなどのその他のパフォーマンス関連のタスクを支援できます。

([BZ#1895188](#))

SSHD RHEL システムロールのサポート

以前は、サーバーおよびクライアントに対して一貫した安定した方法で SSH RHEL System Roles を設定するベンダー対応自動化ツールはありませんでした。今回の機能拡張により、RHEL システムロールを使用して、オペレーティングシステムのバージョンに関係なく、システムかつ統一された方法で **sshd** サーバーを設定できるようになりました。

(BZ#1893696)

Crypto Policies RHEL System Role に対応

今回の機能強化により、RHEL 8 はシステム全体の暗号化ポリシー管理に新しい機能が追加されました。RHEL システムロールを使用すると、任意の数の RHEL 8 システムで暗号化ポリシーを一貫して簡単に設定できるようになりました。

(BZ#1893699)

Logging RHEL システムロールが **rsyslog** の動作に対応しました。

今回の機能強化で、**rsyslog** は Red Hat Virtualization からのメッセージを受信し、そのメッセージを **elasticsearch** に転送するようになりました。

(BZ#1889893)

networking の RHEL システムロールが **ethtool** 設定に対応するようになりました。

今回の機能強化により、**networking** RHEL システムロールを使用して、**NetworkManager** 接続の **ethtool** 結合設定を設定できるようになりました。**interrupt coalescing** 手順を使用すると、システムはネットワークパケットを収集し、複数のパケットに対して割り込みを1つ生成します。これにより、1つのハードウェア割り込みでカーネルに送信されたデータ量が増大し、割り込み負荷が減り、スループットを最大化します。

(BZ#1893961)

4.18. 仮想化

IBM Z 仮想マシンが、最大 248 CPU を実行できるようになりました。

以前のリリースでは、**DIAG318** が有効な IBM Z (s390x) 仮想マシン (VM) で使用できる CPU の数は 240 に制限されていました。現在、Extended-Length SCCB を使用して、IBM Z VM は最大 248 個の CPU を実行できます。

(JIRA:RHELPLAN-44450)

HMAT が RHEL KVM でサポートされるようになりました。

今回の更新で、RHEL KVM で ACPI Heterogeneous Memory Attribute Table (HMAT) に対応するようになりました。ACPI HMAT は、メモリーサイドキャッシュ属性などのメモリー属性に関する情報と、システム物理アドレス (SPA) メモリー範囲に関連する帯域幅およびレイテンシーの詳細を提供することで、メモリーを最適化します。

(JIRA:RHELPLAN-37817)

仮想マシンが Intel Atom P5000 プロセッサの機能を使用できるようになりました。

Snowridge CPU モデル名が仮想マシンで利用できるようになりました。Intel Atom P5000 プロセッサを持つホストでは、仮想マシンの XML 設定の CPU タイプとして **Snowridge** を使用するホストで、これらのプロセッサの新機能が仮想マシンに公開されます。

(JIRA:RHELPLAN-37579)

Windows 10 以降の仮想マシンで、**virtio-gpu** デバイスがより適切に動作するようになりました。

今回の更新で **virtio-win** ドライバーが拡張され、選択した Windows プラットフォームで **virtio-gpu** デバイスのカスタムドライバーも提供されるようになりました。これにより、Windows 10 以降をゲストシステムとして使用する仮想マシンで **virtio-gpu** デバイスのパフォーマンスが向上しました。さらに、デバイスは、将来の機能強化から **virtio-win** にも活用できます。

(BZ#1861229)

第 3 世代 AMD EPYC プロセッサの仮想化サポート

今回の更新で、RHEL 8 の仮想化で、第 3 世代 AMD EPYC プロセッサ (EPYC Milan としても知られる) のサポートが追加されました。これにより、RHEL 8 でホストされる仮想マシンは **EPYC-Milan** CPU モデルを使用し、プロセッサが提供する新機能を使用できるようになりました。

(BZ#1790620)

4.19. クラウド環境の RHEL

AWS 用のゴールドイメージの自動登録

今回の更新で、ユーザーが Red Hat Subscription Management (RHSM) および Red Hat Insights に自動的に登録するように、RHEL 8.4 以降のゴールドイメージを設定できるようになりました。これにより、ゴールドイメージから作成された多数の仮想マシンをより迅速に設定でき、簡単に設定できます。

ただし、RHSM が提供するリポジトリの使用が必要な場合は、`/etc/rhsm/rhsm.conf` の **manage_repos** オプションが **1** に設定されていることを確認してください。詳細は、[Red Hat ナレッジベース](#) を参照してください。

(BZ#1905398, BZ#1932804)

cloud-init が IBM Cloud の Power Systems Virtual Server でサポートされるようになりました。

今回の更新で、**cloud-init** ユーティリティーを使用して、IBM Power Systems ホストでホストされる RHEL 8 仮想マシンを設定し、IBM Cloud Virtual Server サービスで実行できるようになりました。

(BZ#1886430)

4.20. サポート性

sos のバージョン 4.0 へのリベース

sos パッケージがバージョン 4.0 にアップグレードされました。このメジャーバージョンリリースには、多くの新機能および変更が含まれています。

主な変更は以下のとおりです。

- ユーティリティーのメインエントリーポイントとして、以前の **sosreport** バイナリーが新しい **sos** バイナリーに置き換えられました。
- **sos report** を使用して **sosreport tarball** を生成するようになりました。 **sosreport** バイナリーはリダイレクトポイントとして維持され、 **sos report** を呼び出すようになりました。

- `/etc/sos.conf` ファイルが `/etc/sos/sos.conf` に移動され、そのレイアウトは以下のように変更されました。
 - `[general]` セクションの名前が `[global]` に変更され、すべての `sos` コマンドおよびサブコマンドで利用可能なオプションを指定するために使用できます。
 - `[tunables]` セクションの名前が `[plugin_options]` に変更されました。
 - 各 `sos` コンポーネント、`report`、`collect`、および `clean` には、専用のセクションがあります。たとえば、`sos report` は、`global` および `report` からオプションを読み込みます。
- `sos` は Python3 専用のユーティリティになりました。Python2 ではサポートされなくなりました。

sos collect

`sos collect` は、`sos-collector` ユーティリティをメインの `sos` プロジェクトに正式に導入し、複数のノードから同時に `sosreport` を収集するために使用されます。`sos-collector` バイナリは、リダイレクトポイントとして維持され `sos collect` を呼び出します。スタンドアロンの `sos-collector` プロジェクトは、独立して開発されなくなります。`sos collect` の拡張機能には、以下が含まれます。

- `sos collect` は、`sos report` がサポートするすべてのディストリビューション (Policy が定義されているすべてのディストリビューション) でサポートされるようになりました。
- `--insecure-sudo` オプションの名前が `--nopasswd-sudo` に変更されました。
- ノード数に同時に接続するために使用される `--threads` オプションの名前は `--jobs` に変更されました。

sos clean

`sos clean` は、`soscleaner` ユーティリティの機能をメインの `sos` プロジェクトに正式に導入します。このサブコマンドは、IP アドレス、ドメイン名、ユーザー指定のキーワードのクリーニングなど、レポートに対してさらなるデータの難読化を実行します。

注記: `--clean` オプションを `sos report` または `sos collect` コマンドと共に使用すると、`sos clean` が生成されるレポートに適用されます。そのため、レポートを生成し、生成後にはじめてクリーンな機能を適用する必要はありません。

`sos clean` の主な拡張機能には、以下が含まれます。

- IPv4 アドレスの難読化のサポート。これにより、検出されたアドレス間のトポロジー関係の維持が試行される点に注意してください。
- ホスト名およびドメイン名の難読化のサポート。
- ユーザーによって提供されるキーワードの難読化のサポート。
- `sos report` コマンドで使用される `--clean` または `--mask` フラグは、生成されるレポートを難読化します。または、以下のコマンドが既存のレポートを難読化します。

```
[user@server1 ~]$ sudo sos (clean|mask) $archive
```

前者を使用すると、難読化されたレポートアーカイブが1つ作成されますが、後者の場合は難読化されたアーカイブと難読化されていないオリジナルの2つが作成されます。

本リリースに含まれる変更の詳細は、[sos-4.0](#) を参照してください。

(BZ#1966838)

4.21. コンテナ

Podman が Docker 用に作成されたボリュームプラグインに対応

Podman は Docker ボリュームプラグインをサポートするようになりました。ベンダーおよびコミュニティメンバーによって記述されたこれらのボリュームプラグインまたはドライバーは、Podman がコンテナボリュームを作成および管理できます。

podman volume create コマンドが、指定された名前でボリュームプラグインを使用したボリュームの作成をサポートするようになりました。ボリュームプラグインは、**container.conf** 設定ファイルの **[engine.volume_plugins]** セクションで定義する必要があります。

たとえば、以下のようになります。

```
[engine.volume_plugins]
testvol = "/run/docker/plugins/testvol.sock"
```

testvol はプラグインの名前で、**/run/docker/plugins/testvol.sock** はプラグインソケットへのパスです。

podman volume create --driver testvol を使用して、**testvol** プラグインを使用してボリュームを作成できます。

(BZ#1734854)

ubi-micro コンテナイメージが利用できるようになりました。

registry.redhat.io/ubi8/ubi-micro コンテナイメージは、基礎となるホストでパッケージマネージャーを使用して、通常は Buildah を使用するか、Podman でマルチステージビルドをインストールするための最小ベースイメージです。パッケージマネージャーとそのすべての依存関係を除外すると、イメージのセキュリティーレベルが上がります。

(JIRA:RHELPLAN-56664)

自動更新コンテナイメージのサポートが利用可能に

今回の機能強化により、ユーザーは **podman auto-update** コマンドを使用して、自動更新ポリシーに従ってコンテナを自動更新できるようになりました。コンテナには、イメージが更新されているかどうかを確認するために、指定した **"io.containers.autoupdate=image"** ラベルを付ける必要があります。デプロイされている場合には、Podman は新しいイメージをプルし、コンテナを実行する systemd ユニットの再起動します。**podman auto-update** コマンドは systemd に依存し、コンテナを作成するために完全に指定したイメージ名が必要になります。

(JIRA:RHELPLAN-56661)

Podman が、セキュアな短縮名に対応

イメージの短縮名のエイリアスは、**[aliases]** テーブルの **registries.conf** ファイルに設定できるようになりました。short-names モードは以下のようになります。

- Enforcing: イメージのプル中に一致するエイリアスが見つからない場合、Podman はユーザーが非修飾レジストリーのいずれかを選択するよう求めます。選択したイメージが正常にプルされると、Podman は **\$HOME/.config/containers/short-name-aliases.conf** ファイルの新しい short-name エイリアスを自動的に記録します。ユーザーを要求できない場合 (stdin や stdout

など)がTTYではない場合は、Podman は失敗します。**short-name-aliases.conf** ファイルは、両方が同じエイリアスを指定する場合、**registries.conf** ファイルよりも優先されることに注意してください。

- Permissive: Enforcing モードと似ていますが、ユーザーにプロンプトが表示されないと失敗しません。代わりに、Podman は指定された順序で修飾されていないすべてのレジストリーを検索します。エイリアスは記録されないことに注意してください。

たとえば、以下ようになります。

```
unqualified-search-registries=["registry.fedoraproject.org", "quay.io"]
```

```
[aliases]
```

```
"fedora"="registry.fedoraproject.org/fedora"
```

(JIRA:RHELPLAN-39843)

container-tools:3.0 の安定したストリームが利用可能に

Podman、Buildah、Skopeo、および runc ツールを含む **container-tools:3.0** モジュールストリーム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が追加されました。

以前の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。

(JIRA:RHELPLAN-56782)

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 8.4 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。

5.1. 新しいカーネルパラメーター

bgrt_disable = [ACPI, X86]

このパラメーターは、BGRT (Boot Graphics Resource Table) を無効にし、Original Equipment Manufacturer (OEM) ロゴのフォッチを避けるようにします。

radix_hcall_invalidate = on [PPC/PSERIES]

このパラメーターは、Radix GTSE 機能を無効にし、トランスレーションルックアサイドバッファー (TLB: Translation Lookaside Buffer) に hcall を使用します。

disable_tlbie = [PPC]

このパラメーターは、Translation Look-Aside Buffer Invalidate Entry (TLBIE) 命令を無効にします。現在、KVM、ハッシュメモリー管理ユニット (MMU)、または整合性のあるアクセラレーターを使用します。

fw_devlink = [KNL]

このパラメーターは、ファームウェアをスキャンして、コンシューマーとサプライヤーの関係を推測することにより、コンシューマーとサプライヤーデバイス間にデバイスリンクを作成します。この機能は、ドライバーがモジュールとしてロードされ、以下のようなタスクが適切に順序付けされるようにします。

- デバイスのプロービング (サプライヤー、コンシューマーの順)
- サプライヤーブートの状態がクリーンアップされます (すべてのコンシューマーがプローブした後でのみ)。
- Res suspend、resume、および runtime Power Management (PM) (コンシューマーを最初に、その後にサプライヤー)
形式: { off | permissive | on | rpm }
- **off**: ファームウェア情報からデバイスリンクを作成しません。
- **permissive**: ファームウェア情報からデバイスリンクを作成しますが、起動の状態がクリーンアップする (**sync_state()** 呼び出し) にのみ使用してください。
- **on**: ファームウェア情報からデバイスリンクを作成し、それを使用してプローブおよび中断または再開の順序を実施します。
- **rpm**: パイルは **on** になりますが、ランタイム PM にも一致させるためにも使用されます。

デフォルト値は **permissive** です。設定された値は **/proc/cmdline** ファイルで確認できます。

init_on_alloc = [MM]

このパラメーターは、新しく割り当てられたページとヒープオブジェクトをゼロで埋めます。
形式: 0 | 1

カーネル **CONFIG_INIT_ON_ALLOC_DEFAULT_ON** の設定によりデフォルトで設定されます

init_on_free = [MM]

このパラメーターは、空きページとヒープオブジェクトをゼロで埋めます。

形式: 0|1

デフォルトでは、**CONFIG_INIT_ON_FREE_DEFAULT_ON** で設定されます

nofsgsbase [X86]

このパラメーターは、FSGSBASE 命令を無効にします。

nosgx [X86-64,SGX]

このパラメーターは、Intel Software Guard Extensions (SGX) カーネルサポートを無効にします。

rcutree.rcu_min_cached_objs = [KNL]

1つのCPUごとにキャッシュされ、保持されるオブジェクトの最小数。オブジェクトサイズは **PAGE_SIZE** と等しくなります。キャッシュを使用すると、ページアロケーターの無関係を軽減できます。また、メモリーが少ない状態でより適切に動作するアルゴリズム全体にします。

rcuperf.kfree_rcu_test = [KNL]

このパラメーターは、**kfree_rcu()** 関数のパフォーマンスを測定するために使用されます。

rcuperf.kfree_nthreads = [KNL]

kfree_rcu() のループを実行するスレッドの数。

rcuperf.kfree_alloc_num = [KNL]

反復で実施される割り当て数および解放の数。

rcuperf.kfree_loops = [KNL]

割り当ておよび開放の **rcuperf.kfree_alloc_num** 番号を実行するループ数。

rcupdate.rcu_cpu_stall_ftrace_dump = [KNL]

このパラメーターは、Read-copy-update (RCU) CPU が停止された警告をレポートした後に **ftrace** バッファをダンプします。

nopvspin = [X86,KVM]

このパラメーターは、Para-virtualization(PV) の最適化を使用して qspinlock の **qspinlock** パスを無効にします。これにより、ハイパーバイザーがロック競合時にゲストを idle することができます。

5.2. 新しい /PROC/SYS/USER パラメーター

max_time_namespaces

現在のユーザー namespace の任意のユーザーが作成できる最大 namespace 数。

5.3. 新しい /PROC/SYS/VM パラメーター

compaction_proactiveness

このパラメーターは、バックグラウンドでカーネルがメモリーをどれくらい積極的に圧縮するかを決定します。このパラメーターは、[0, 100] の範囲内の値を取り、デフォルト値は 0 に設定されません。デフォルトでこのパラメーターを無効にする要因は、メモリーを移動するために 500 ミリ秒ごとにウェイクアップされる kthread によって、現在確立され、予想されるシステムの動作を壊さないようにすることでした。

圧縮処理には、異なるプロセスに属するページが進められているため、システム全体のシステムに影響を与えることに注意してください。また、アプリケーションの不適切な動作のレイテンシーが発生する可能性があります。カーネルは、プロアクティブなコンパクションが有効ではないことを検知した場合に、CPU サイクルをさまざまなヒューリスティックを使用し、CPU サイクルを防ぎます。

このパラメーターを 100 などの大きな値に設定する場合には注意してください。これにより、バックグラウンドの圧縮アクティビティーが過剰になる可能性があります。

watermark_boost_factor

このパラメーターは、メモリーが断片化される際に回収のレベルを制御します。これは、異なるモビリティのページがページブロック内で混在している場合に回収されるゾーンの高基準の割合を定義します。目的は、将来的に実行する設定が少なく、SLUB 割り当て、THP や hugetlbfs ページなどの今後の高順の割り当ての成功率を増やすことです。

watermark_scale_factor パラメーターに関して、ユニットは 10,000 の割合にあります。!**DISCONTIGMEM** 設定の 15,000 のデフォルト値は、断片化によりページブロックが混在される際に、最大 150% の高ウォーターマークが回収されることを意味します。再利用のレベルは、直近で発生した断片化イベント数により決定されます。この値がページブロックよりも小さい場合は、ページブロックのページブロックが回収されます (64 ビット x86 の 2MB)。ブーストファクター 0 により機能が無効になります。

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

- Realtek 802.11ac wireless 8822b driver (rtw88_8822b.ko.xz)
- Realtek 802.11ac wireless 8822be driver (rtw88_8822be.ko.xz)
- Realtek 802.11ac wireless 8822c driver (rtw88_8822c.ko.xz)
- Realtek 802.11ac wireless 8822ce driver (rtw88_8822ce.ko.xz)
- Realtek 802.11ac wireless core module (rtw88_core.ko.xz)
- Realtek 802.11ac wireless PCI driver (rtw88_pci.ko.xz)
- UDP カプセル化トラフィック用のインターフェイスドライバー (bareudp.ko.xz)

グラフィックドライバーとその他のドライバー

- Regmap SoundWire Module (regmap-sdw.ko.xz)
- Intel® QuickAssist Technology (qat_4xxx.ko.xz)
- Intel® Data Accelerator Driver (idxd.ko.xz)
- Oracle VM VirtualBox Graphics Card (vboxvideo.ko.xz)
- Logitech ゲーミングキーボード上のゲームキーの HID ドライバー (hid-lg-g15.ko.xz)
- HWMON インターフェイスによる RAPL MSR からの AMD Energy レポートのドライバー (amd_energy.ko.xz)
- Elastic Fabric Adapter (EFA) (efa.ko.xz)
- AMD® PCI-E Non-Transparent Bridge Driver (ntb_hw_amd.ko.xz)
- PCIe NTB Performance Measurement Tool (ntb_perf.ko.xz)
- PCIe NTB Simple Pingpong Client (ntb_pingpong.ko.xz)
- PCIe NTB Debugging Tool (ntb_tool.ko.xz)
- Software Queue-Pair Transport over NTB (ntb_transport.ko.xz)
- Intel Elkhart Lake PCH pinctrl/GPIO driver (pinctrl-elkhartlake.ko.xz)
- Dell プラットフォーム設定制御インターフェイス (dell-wmi-sysman.ko.xz)
- DesignWare PWM Controller (pwm-dwc.ko.xz)
- SoundWire bus (soundwire-bus.ko.xz)
- Cadence Soundwire Library (soundwire-cadence.ko.xz)

- SoundWire Generic Bandwidth Allocation ([soundwire-generic-allocation.ko.xz](#))
- Intel Soundwire Init Library ([soundwire-intel.ko.xz](#))
- Fast-charge control for Apple "MFi" devices ([apple-mfi-fastcharge.ko.xz](#))
- TI HD3SS3220 DRP Port Controller Driver ([hd3ss3220.ko.xz](#))
- STMicroelectronics STUSB160x Type-C controller driver ([stusb160x.ko.xz](#))
- Nitro Enclaves Driver ([nitro_enclaves.ko.xz](#))

6.2. 更新されたドライバー

グラフィックおよびその他ドライバーの更新

- VMware SVGA デバイスのスタンドアロンの drm ドライバー ([vmwgfx.ko.xz](#)) がバージョン 2.18.0.0 に更新されました。
- Cisco FCoE HBA ドライバー ([fnic.ko.xz](#)) がバージョン 1.6.0.53 に更新されました。
- Driver for HP Smart Array Controller バージョン 3.4.20-200-RH1 ([hpsa.ko.xz](#)) がバージョン 3.4.20-200-RH1 に更新されました。
- Emulex LightPulse Fibre Channel SCSI ドライバー 12.8.0.5 ([lpfc.ko.xz](#)) がバージョン 0:12.8.0.5 に更新されました。
- LSI MPT Fusion SAS 3.0 Device Driver ([mpt3sas.ko.xz](#)) がバージョン 35.101.00.00 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 10.02.00.104-k に更新されました。
- SCSI デバッグアダプタードライバー ([scsi_debug.ko.xz](#)) がバージョン 0190 に更新されました。
- Driver for Microsemi Smart Family Controller バージョン 1.2.16-012 ([smartpqi.ko.xz](#)) がバージョン 1.2.16-012 に更新されました。
- HPE watchdog driver ([hpwdt.ko.xz](#)) がバージョン 2.0.4 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 8.4 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

Anaconda で、テキストモードで ldl または未フォーマットの DASD ディスクダイアログを表示

以前は、テキストモードでのインストール時に、Anaconda は Linux ディスクレイアウト (**ldl**) またはフォーマットされていない DASD (Direct-Access Storage Device) ディスクのダイアログを表示できませんでした。そのため、ユーザーはインストールにこれらのディスクを使用できませんでした。

今回の更新で、テキストモードの Anaconda が **ldl** ディスクと未フォーマットの DASD ディスクを認識し、インストールの今後の使用のためにユーザーが適切にフォーマットできるダイアログが表示されるようになりました。

(BZ#1874394)

RHEL インストーラー起動オプションを使用して InfiniBand ネットワークインターフェイスを設定すると、RHEL インストーラーが起動しませんでした。

以前では、インストーラー起動オプション (PXE サーバーを使用したインストーラーイメージのダウンロードなど) を使用して RHEL インストールの初期段階で InfiniBand ネットワークインターフェイスを設定する場合、インストーラーはネットワークインターフェイスのアクティブ化に失敗していました。

この問題は、RHEL NetworkManager が InfiniBand モードでネットワークインターフェイスを認識できず、代わりにインターフェイスのイーサネット接続を設定するために発生していました。

その結果、接続のアクティベーションに失敗し、InfiniBand インターフェイスを介した接続が初期段階で必要な場合、RHEL インストーラーはインストールを開始できませんでした。

今回のリリースにより、インストーラーがインストーラー起動オプションを使用して RHEL インストールの初期段階で設定する InfiniBand ネットワークインターフェイスが正常にアクティベートされ、インストールが正常に実行されるようになりました。

(BZ#1890009)

自動パーティション設定が Anaconda でスケジュール可能

以前は、LVM タイプのディスクの自動パーティション設定時に、インストーラーは選択したディスクごとに LVM PV のパーティション作成を試みていました。すでにパーティションレイアウトが存在している場合は、自動パーティション設定のスケジュールが失敗し、エラーメッセージが出力される可能性があります。

今回の更新で、この問題が修正されています。これで、インストーラーで自動パーティション設定をスケジュールできるようになりました。

(BZ#1642391)

Anaconda GUI を使用したワイヤレスネットワークの設定が修正されました。

以前は、Anaconda グラフィカルユーザーインターフェイス (GUI) を使用して無線ネットワークを設定すると、インストールがクラッシュしていました。

今回の更新で、この問題が修正されています。Anaconda GUI を使用すると、インストール時に無線ネットワークを設定できます。

(BZ#1847681)

7.2. ソフトウェア管理

%autopatch rpm マクロで新しい **-m** および **-M** パラメーターに対応

今回の更新で、**-m** (min) および **-M** (max) パラメーターが **%autopatch** マクロに追加され、指定のパラメーターでパッチの範囲だけが適用されるようになりました。

(BZ#1834931)

POPT がバージョン 1.18 にリベースされました。

popt パッケージがアップストリームバージョン 1.18 にアップグレードされ、以前のバージョンに対する主な変更点が加えられました。

- 全体的なコードベースクリーンアップおよび最新化。
- **alias exec** コマンドの特権の削除に失敗する問題が修正されました。
- リソースリークを含むさまざまなバグが修正されました。

(BZ#1843787)

7.3. シェルおよびコマンドラインツール

snmpbulkget が存在しない PID に有効な出力を提供

snmpbulkget コマンドは、存在しない PID に有効な出力を提供しませんでした。そのため、このコマンドは **結果が見つからない**ため、出力で失敗していました。

今回の更新で、**snmpbulkget** が存在しない PID に有効な出力を提供するようになりました。

(BZ#1817190)

CRON コマンドは、トリガー条件に従ってメールを送信するようになりました。

以前のリリースでは、Relax-and-Recover (**ReaR**) ユーティリティーが誤って設定されている場合、**CRON** コマンドは、メールを介して管理者に送信されたエラーメッセージをトリガーしていました。そのため、**ReaR** で設定が実行されていない場合でも、管理者はメールを受け取りました。

今回の更新により、**CRON** コマンドが変更され、トリガー条件に基づいてメールを送信するようになりました。

(BZ#1729499)

ReaR のバックアップメカニズムとして、BPM バージョン 8.2 を使用できるようになりました。

以前では、バックアップメソッドとして VC を使用する場合、Relax-and-Recover (**ReaR**) ユーティリティーは、レスキューシステムで **vxpbx_exchanged** サービスを開始しませんでした。そのため、RPI 8.2 のあるレスキューシステムのバックアップからの復元に失敗し、以下のエラーメッセージが、RPI サーバーに記録されていました。

**Error bpbrm (pid=...) cannot execute cmd on clientInfo tar (pid=...) done. status: 25: cannot connect on socket
Error bpbrm (pid=...) client restore EXIT STATUS 25: cannot connect on socket**

今回の更新で、**ReaR** は **vxbpx_exchanged** 関連ファイルをレスキューシステムに追加して、レスキューシステムの起動時にサービスを開始するようになりました。

(BZ#1898080)

libvpd がバージョン 2.2.8 にリベースされました。

以下は、主な変更点です。

- **sqlite** 操作を非同期にすることで、**vpdupdate** のパフォーマンスが向上しました。

(BZ#1844429)

ReaR ユーティリティーが、**LUKS2** で暗号化されたパーティションを使用してシステムを復元するようになりました。

以前のリリースでは、**Rear** (Relax-and-Recover) ユーティリティーでバックアップするためにシステムに **LUKS2** で暗号化したパーティションが少なくとも1つ存在する場合、**Relax-and-Recover (Rear)** ユーティリティーで **ReaR** が **LUKS2** 暗号化したパーティションに対応していないことをユーザーに通知していませんでした。したがって、**ReaR** ユーティリティーは復元フェーズでシステムの元の状態を再作成できませんでした。

今回の更新で、**ReaR** ユーティリティーに、基本的な **LUKS2** 設定、エラーの確認、および改善された出力に対応するようになりました。**ReaR** ユーティリティーは、基本的な **LUKS2** 暗号化パーティションを使用してシステムを復元するか、逆のケースでユーザーに通知するようになりました。

(BZ#1832394)

texlive が **Poppler** で正しく機能するようになりました。

以前は、**Poppler** ユーティリティーが API の変更の更新が行われていました。そのため、これらの API の変更が原因で **Texlive** ビルドは機能していませんでした。今回の更新で、**Texlive** ビルドが新しい **Poppler** ユーティリティーで正常に機能するようになりました。

(BZ#1889802)

7.4. インフラストラクチャーサービス

RPZ がワイルドカード文字で機能

以前は、**lib/dns/rpz.c** ファイルの **dns_rpz_find_name** 関数は、同じ接尾辞のレコードが存在する場合にワイルドカード文字を考慮していませんでした。そのため、ワイルドカード文字を含む一部のレコードは無視されました。今回の更新で、**dns_rpz_find_name** 関数が修正され、ワイルドカード文字を考慮するようになりました。

(BZ#1876492)

7.5. セキュリティー

pkcs11 のパディングが改善

以前は、**pkcs11** トークンラベルにスマートカードの追加のパディングが含まれていました。そのため、誤ったパディングにより、**label** 属性に基づいてカードの一致が生じる可能性がありました。今回の更新で、全カードでパディングが修正され、**PKCS #11** の URI を定義し、アプリケーション内のそれ

らに対して一致させることが期待どおりに機能するようになりました。

(BZ#1877973)

sealert 接続の問題処理を修正しました。

以前では、**setroubleshoot** デーモンのクラッシュにより、**sealert** プロセスが応答しなくなる可能性があります。そのため、GUI で分析が表示されず応答なくなり、コマンドラインツールは何も出力せず、強制終了されるまで実行を継続していました。今回の更新で、**sealert** と **setroubleshootd** 間の接続問題の処理が改善されました。**sealert** は、**setroubleshoot** デーモンがクラッシュした場合にエラーメッセージを報告し、終了するようになりました。

(BZ#1875290)

setroubleshoot による監査レコード分析の最適化

以前は、**setroubleshoot-3.3.23-1** で導入された新機能はパフォーマンスに悪影響を与えるため、AVC 分析は以前よりも 8 回遅くなります。今回の更新で、AVC 分析時間が大幅に削減される最適化機能が提供されます。

(BZ#1794807)

SELinux ポリシーインターフェイスパーサーを修正しました。

以前のバージョンでは、ポリシーインターフェイスパーサーは構文エラーメッセージが表示されていました。これには、カスタムポリシーをインストールすると、インターフェイスファイルに **ifndef** ブロックが含まれていました。今回の更新で、インターフェイスファイルの解析が改善され、この問題が解決されました。

(BZ#1868717)

setfiles がラベル付けエラーで停止しない

以前のバージョンでは、**setfiles** ユーティリティは、ファイルの再ラベルに失敗したたびに停止していました。そのため、誤ったファイルにラベルが付けられたファイルがターゲットディレクトリーに残されていました。今回の更新により、**setfiles** は再ラベルできないファイルを省略し、**setfiles** はターゲットディレクトリーのすべてのファイルを処理するようになりました。

(BZ#1926386)

SELinux ポリシーストアを再構築すると、電源が切れやすくなります。

以前のリリースでは、キャッシュの書き込みにより、SELinux-policy を再構築すると、電源障害に耐えませんでした。そのため、ポリシーの再ビルド時に、SELinux ポリシーストアが機能しなくなると、電源障害が発生する可能性があります。今回の更新で、**libsemanage** ライブラリーは、メタデータおよびキャッシュされたファイルデータに対する保留中の変更はすべて、これを使用する前にポリシーストアが含まれるファイルシステムに書き込まれるようになりました。その結果、ポリシーストアは電源障害や他の中断に対してより回復できるようになりました。

(BZ#1913224)

libselinux で SELinux ユーザーのデフォルトコンテキストが正しく判断されるようになりました。

以前は、**libselinux** ライブラリーは、非推奨の **security_compute_user()** 関数を使用したため、一部のシステムで SELinux ユーザーのデフォルトのコンテキストを決定できませんでした。そのため、複雑なセキュリティーポリシーがあるシステムでは、一部のシステムサービスが利用できませんでした。今回の更新で、**libselinux** は **security_compute_user()** を使用なくなり、ポリシーの複雑性に関わらず、SELinux ユーザーのデフォルトのコンテキストが適切に判断されるようになりました。

(BZ#1879368)

SELinux により rsync モードの geo レプリケーションが失敗しなくなりました。

以前は、SELinux ポリシーでは、**rsync_t** で実行しているプロセスが **security.trusted** 拡張属性の値を設定できませんでした。そのため、Red Hat Gluster Storage(RHGS) の geo レプリケーションに失敗していました。この更新には、**rsync_t** プロセスによる **security.trusted** の設定を可能にする新しい SELinux ブール値 **rsync_sys_admin** が含まれます。その結果、**rsync_sys_admin** ブール値を有効にすると、**rsync** は **security.trusted** 拡張属性を設定して geo レプリケーションに失敗しなくなりました。

(BZ#1889673)

OpenSCAP はメモリーが不足することなく、ファイルが大量に含まれるシステムをスキャンできるように。

以前のリリースでは、メモリーが少なく、ファイルが多数あるシステムをスキャンすると、OpenSCAP スキャナーが原因でシステムのメモリーがなくなることがありました。今回の更新で、OpenSCAP スキャナーのメモリー管理が改善されました。その結果、スキャナーは、たとえば、**Server with GUI** および **Workstation** を使用するパッケージグループなど、多数のファイルのスキャンする際に、RAM が少ないシステムでメモリーが不足することがなくなりました。

(BZ#1824152)

FAT が設定された CIS 予約システムが起動時に失敗しなくなる

以前は、SCAP Security Guide (SSG) の Center for Internet Security (CIS) プロファイルには、FAT ファイルシステムへのアクセスを行うカーネルモジュールの読み込みを無効にするルールが含まれていました。そのため、SSG がこのルールを修正した場合、システムは EFI システムパーティション (ESP) を含む FAT12, FAT16 および FAT32 ファイルシステムでフォーマットされたパーティションにアクセスできませんでした。これにより、システムが起動できませんでした。今回の更新で、ルールがプロファイルから削除されました。その結果、このファイルシステムを使用するシステムは起動に失敗しなくなりました。

(BZ#1927019)

OVAL チェックは GPFS をリモートとして考慮する

以前は、OpenSCAP スキャナーはマウントされた General Parallel File Systems (GPFS) をリモートファイルシステム (FS) として識別しませんでした。これにより、ローカルシステムのみ適用される OVAL チェックであっても、OpenSCAP は GPFS をスキャンしていました。これにより、スキャナーがリソースが不足し、スキャンの完了に失敗することがありました。今回の更新で、GPFS がリモート FS のリストに含まれるようになりました。その結果、OVAL チェックは GPFS をリモートの FS として正しく考慮し、スキャンは高速です。

(BZ#1840579)

fapolicyd-selinux SELinux ポリシーがすべてのファイルタイプに対応するようになりました。

以前では、**fapolicyd-selinux** SELinux ポリシーはすべてのファイルタイプに対応しませんでした。そのため、**fapolicyd** サービスは、**sysfs** などのスケジュールされていない場所にあるファイルにアクセスできませんでした。今回の更新で、**fapolicyd** サービスはすべてのファイルシステムタイプに準拠し、分析するようになりました。

(BZ#1940289)

fapolicyd が RHEL の更新を阻止しなくなりました。

更新で実行中のアプリケーションのバイナリーが置き換えられると、カーネルにより、接尾辞

(deleted) が追加されて、メモリー内のアプリケーションのバイナリーパスが変更されます。以前は、**fapolicyd** ファイルのアクセスポリシーデーモンは、このようなアプリケーションを信頼できないものとして処理していました。そのため、**fapolicyd** は、これらのアプリケーションがその他のファイルを開いて実行できませんでした。今回の更新で、**fapolicyd** はバイナリーパスの接尾辞を無視し、バイナリーが信頼データベースに一致するようになりました。これにより、**fapolicyd** がルールを正しく適用し、更新プロセスを完了できるようになりました。

(BZ#1896875)

USBGuard が 1.0.0-1 にリベースされました

usbguard パッケージが、アップストリームバージョン 1.0.0-1 にリベースされました。今回の更新では、改良されたバグ修正およびバグ修正が追加されました。主要な変更点は次のとおりです。

- 安定したパブリック API により、後方互換性が確保されます。
- **rules.d** ディレクトリー内のルールファイルは、英数字順を読み込むようになりました。
- 1つのルールで複数のデバイスのポリシーを変更できない場合に一部のユースケースが修正されました。
- ラベルによるルールのフィルタリングでエラーが生成されなくなりました。

(BZ#1887448)

USBGuard が Audit メッセージを送信できるようになりました。

サービスのハードニングの一環として、**CAP_AUDIT_WRITE** がない間、**usbguard.service** の機能に制限されていました。そのため、システムサービスとして実行している **usbguard** は、Audit イベントを送信できませんでした。今回の更新で、サービス設定が更新され、USBGuard が Audit メッセージを送信できるようになりました。

(BZ#1940060)

tangd が無効な要求を適切に処理するようになりました。

以前では、**tangd** デーモンは、一部の無効なリクエストのエラー終了コードを返していました。これにより、**tangd.socket@.service** が失敗し、そのユニットの数が増加すると問題が発生する可能性があります。今回の更新で、**tangd** サーバー自体が反応している場合にのみ、エラーコードで **tangd** が終了します。これにより、**tangd** が無効な要求を適切に処理します。

(BZ#1828558)

7.6. ネットワーク

ipset ルックアップに関連するルールを使用した RHEL 7 から RHEL 8 への iptables ルールセットの移行に失敗しなくなりました。

以前は、**ipset** カウンターは、**iptables** ルールセットからカウンターが有効な **ipset** コマンドを参照する間、追加の制約がすべて一致する場合にのみ更新されました。したがって、**ipset** ルックアップに関連するルール (**-m set --match-set xxx src --bytes-gt 100** など) は、一致するように変更されることはありません。これは、**ipset** のカウンターが追加されないためです。今回の更新で、**ipset** ルックアップに関連するルールが含まれる **iptables** ルールセットの移行が期待どおりに機能するようになりました。

(BZ#1806882)

iptraf-ng が raw メモリーコンテンツを公開しなくなりました。

以前は、**iptraf-ng** のフィルターに **%p** を設定すると、アプリケーションはステータスバーに raw メモリーコンテンツを表示していました。そのため、不要な情報が表示されます。今回の更新で、**iptraf-ng** プロセスが、ステータスバーに生のメモリーコンテンツが表示されなくなりました。

(BZ#1842690)

Anaconda ip 起動オプションで DHCP を使用すると、ネットワークアクセスが利用できるようになりました。

初期 RAM ディスク (**initrd**) は、NetworkManager を使用してネットワークを管理します。以前は、RHEL 8.3 ISO ファイルが提供する **dracut** NetworkManager モジュールは誤って、Anaconda 起動オプションの **ip** オプションの最初のフィールドが常に設定されていると仮定していました。そのため、DHCP を使用して **ip=::::<host_name>::dhcp** を設定すると、NetworkManager は IP アドレスを取得せず、Anaconda でネットワークを利用できませんでした。この問題が修正されました。これにより、上記のシナリオで RHEL 8.4 ISO を使用してホストをインストールすると、Anaconda **ip** 起動オプションが期待どおりに機能します。

(BZ#1900260)

nfp ドライバーを使用する Netronome ネットワークカードで XDP プログラムのアンロードに失敗しなくなりました。

以前は、Netronome ネットワークカードの **nfp** ドライバーにバグが含まれていました。そのため、**XDP_FLAGS_REPLACE** フラグで **IFLA_XDP_EXPECTED_FD** 機能を使用して XDP プログラムをロードした場合、eXpress Data Path (XDP) プログラムのアンロードに失敗していました。たとえば、これは **libxdp** ライブラリーを使用してロードされた XDP プログラムに該当します。このバグは修正されています。その結果、Netronome ネットワークカードから XDP プログラムをアンロードすることは期待どおりに機能します。

(BZ#1880268)

NetworkManager が、全インターフェイスで DHCP および逆引き DNS ルックアップを使用してホスト名の取得を試みるようになりました。

以前は、ホスト名が **/etc/hostname** ファイルに設定されていない場合、NetworkManager は、最も低いメトリック値を持つインターフェイスを介して、DHCP または逆引き DNS ルックアップを使用してホスト名の取得を試みていました。そのため、デフォルトルートなしでは、ネットワークでホスト名を自動的に割り当てることができませんでした。今回の更新で動作が変更され、NetworkManager はまずデフォルトのルートインターフェイスを使用してホスト名の取得を試みるようになりました。このプロセスに失敗すると、NetworkManager はその他の利用可能なインターフェイスを試行します。これにより、**/etc/hostname** に設定されていない場合、NetworkManager は、全インターフェイスの DHCP および逆引き DNS ルックアップを使用してホスト名を取得しようとします。

NetworkManager が以前の動作を使用するように設定するには、以下を行います。

1. 以下の内容で **/etc/NetworkManager/conf.d/10-hostname.conf** ファイルを作成します。

```
[connection-hostname-only-from-default]
hostname-only-from-default=1
```

2. **NetworkManager** サービスを再読み込みします。

```
# systemctl reload NetworkManager
```

(BZ#1766944)

7.7. カーネル

IBM Z システムで、カーネルが誤検出の警告を返しなくなりました。

以前は、RHEL 8 の IBM Z システムに、ユーザーアクセスを許可する **ZONE_DMA** メモリーゾーンの許可されるエントリーがありませんでした。したがって、カーネルは以下のような誤検出の警告を返します。

```
...
Bad or missing usercopy whitelist? Kernel memory exposure attempt detected from SLUB object
'dma-kmalloc-192' (offset 0, size 144)!
WARNING: CPU: 0 PID: 8519 at mm/usercopy.c:83 usercopy_warn+0xac/0xd8
...
```

sysfs インターフェイスから特定のシステム情報にアクセスする際に警告が表示されました。たとえば、**debuginfo.sh** スクリプトを実行するなどです。

今回の更新で、ユーザー空間アプリケーションがバッファにアクセスできるように、Direct Memory Access (DMA) バッファにフラグが追加されました。

これにより、上記のシナリオで警告メッセージが表示されなくなります。

(BZ#1660290)

RHEL システムが **tboot** GRUB エントリーから期待どおりに起動する

以前は、バージョン 1.9.12-2 の **tboot** ユーティリティーが原因で、Trusted Platform Module (TPM) 2.0 が有効になっている一部の RHEL システムがレガシーモードでの起動に失敗していました。その結果、**tboot** Grand Unified Bootloader (GRUB) エントリーから起動しようとする、システムが停止しました。RHEL 8 の新しいバージョンと **tboot** ユーティリティーの更新により、問題が修正され、RHEL システムが期待どおりに起動します。

(BZ#1947839)

負荷の高いコンテナシナリオでカーネルがメモリーを正常に回収

コンテナ内の I/O およびメモリーにボリュームが制約された場合、データの競合状態によりメモリーのロックアップの回収を行うカーネルコード。データ競合は、次のような場合に発生する現象です。

- 少なくとも 2 つの CPU スレッドが同じデータセットを同時に変更しようとしています。
- これらの CPU スレッドのいずれかは、データセットで書き込み操作を試行します。

データセットを変更するための各スレッドの正確なタイミングをもとに、結果は A、B、または AB (無決定) になります。

コンテナがメモリー不足の状態にあると、複数の OOM (Out of Memory) による強制終了が生じ、コンテナがロックされ、応答しなくなりました。本リリースでは、ロックおよび最適化用の RHEL カーネルコードが更新されました。その結果、カーネルは応答しなくなり、データは競合状態に影響しなくなります。

(BZ#1860031)

メモリーがオフラインの RHEL 8 により、カーネルパニックが発生しなくなりました。

以前は、開始し、オフラインとマークされたメモリーで RHEL 8 を実行すると、カーネルが初期化されていないメモリーページにアクセスしようとする場合があります。これにより、カーネルパニックが発生していました。今回の更新で、アイドルページ追跡のカーネルメカニズムが修正され、問題が発生

しなくなりました。

(BZ#1867490)

NUMA システムでは、予期しないメモリーレイアウトが発生しなくなりました。

以前のリリースでは、**CONFIG_NODES_SPAN_OTHER_NODES** オプションがないため、**ARM64** および **S390** アーキテクチャーでは NUMA システムで予期しないメモリーレイアウトが発生していました。そのため、異なる NUMA ノードのメモリー領域が交差し、低い NUMA ノードのメモリーリージョンが高 NUMA に追加されていました。

今回の更新により、NUMA システムはメモリーレイアウトの問題が発生しなくなりました。

(BZ#1844157)

rngd サービスが poll() システムコールでビジー状態にならない

バージョン 4.18.0-193.10 で始まるカーネル用に、FIPS モードの新しいカーネルエントロピーソースが追加されました。そのため、**/dev/random** デバイスの **poll()** システムコールで、**rngd** サービスがビジーウェイトしていました。これにより、システムが FIPS モードにある場合に CPU 時間を 100% 消費していました。今回の更新で、FIPS モードでは、**/dev/random** デバイスの **poll()** ハンドラーが、特に **/dev/random** デバイスに対して開発されたデフォルトのハンドラーからハンドラーに変更されました。その結果、上記のシナリオで **rngd** サービスが **poll()** でビジーウェイトしなくなりました。

(BZ#1884857)

SCHED_DEADLINE スケジューラーの HRTICK サポートが有効になっている

以前は、**SCHED_DEADLINE** ポリシーで設定した特定のタスクでは、高解像度のシステムタイマー (**HRTICK**) の機能が配置されませんでした。そのため、**SCHED_DEADLINE** スケジューラーを使用するこれらのタスクのスロットリングメカニズムは、これらのタスクに設定したすべてのランタイムを使用しました。この動作により、リアルタイム環境で予期しないレイテンシーが急増していました。

今回の更新により、高解像度のプリエンプションを提供する **HRTICK** 機能が有効になりました。**HRTICK** は高解像度タイマーを使用して、タスクがランタイムを完了するときにスロットリングメカニズムを強制します。その結果、この問題は上記のシナリオで発生しなくなりました。

(BZ#1885850)

tpm2-abrmd のバージョン 2.3.3.2 へのリベース

tpm2-abrmd パッケージがバージョン 2.3.3.2 にアップグレードされ、バグ修正が複数追加されました。以下は、主な変更点です。

- 一時的なハンドルの使用を修正しました。
- TPM Command Transmission Interface (TCTI) での部分的な読み込みを修正しました。
- アクセスブローカーのリファクタリング

(BZ#1855177)

cxgb4 ドライバーにより、kdump カーネルでクラッシュが発生しなくなる

以前のリリースでは、**vmcore** ファイルに情報を保存しようとする、**kdump** カーネルがクラッシュしていました。そのため、**cxgb4** ドライバーにより、**kdump** カーネルが、後で分析するためにコアを保存できなくなります。この問題を回避するには、**kdump** カーネルコマンドラインに **novmcoredd** パラメーターを追加して、コアファイルを保存できるようにします。

[RHSA-2020:1769](#) アドバイザリーのリリースにより、**kdump** カーネルがこの状況を適切に処理し、クラッシュしなくなりました。

([BZ#1708456](#))

7.8. ファイルシステムおよびストレージ

EREMOTE エラーで SMB ターゲットへのアクセスが失敗しなくなりました。

以前は、**cifsacl** マウントオプションを使用した RHEL SMB クライアントに VC 名前空間をマウントすることはできず、リストが **EREMOTE** エラーで失敗していました。今回の更新で、カーネルが **EREMOTE** に対応するように修正され、SMB 共有にアクセスできるようになりました。

([BZ#1871246](#))

NFS readdir 関数のパフォーマンス向上

以前は、ディレクトリーをリスト表示する NFS クライアントのプロセスが、リストを完了するのに時間がかかり、永久に完了しなくなる可能性があります。今回の更新で、次のシナリオで、NFS クライアントディレクトリーのリスティングが改善しました。

- 100,000 個以上のファイルを含む大きなディレクトリーのリスト表示。
- 修正されるディレクトリーのリスト。

([BZ#1893882](#))

7.9. 高可用性およびクラスター

corosync.conf ファイルのデフォルトのトークンタイムアウト値が 1 秒から 3 秒に増加

以前は、**corosync.conf** ファイルの TOTEM トークンのタイムアウト値が 1 秒に設定されていました。この短いタイムアウトにより、クラスターはすぐに応答しますが、ネットワークの遅延があると、予想外のフェイルオーバーが生じる可能性があります。デフォルト値は 3 秒に設定され、迅速な応答とより幅広い適用性との間でトレードオフを提供します。token のタイムアウト値を変更する方法は、[How to change totem token timeout value in a RHEL 5, 6, 7, or 8 High Availability cluster?](#) を参照してください。

([BZ#1870449](#))

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

perl-Time-HiRes がインストールされている場合にインプレースアップグレードが可能に

以前は、RHEL 8 で配布される **perl-Time-HiRes** パッケージには、RHEL 7 バージョンのパッケージに含まれていたエポック番号がありませんでした。そのため、**perl-Time-HiRes** のインストール時に、RHEL 7 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せんでした。欠落していたエポック番号が追加され、**perl-Time-HiRes** のインストール時にインプレースアップグレードに失敗しなくなりました。

([BZ#1895852](#))

7.11. コンパイラーおよび開発ツール

glibc DNS スタブリゾルバーが、同じトランザクション ID で並列クエリーを正しく処理

今回の更新以前は、GNU C ライブラリー **glibc** の DNS スタブリゾルバーは、同じトランザクション ID を持つ並列クエリーへの応答を処理しませんでした。そのため、トランザクション ID が等しい場合、2 番目の並列応答がクエリーと一致せず、タイムアウトおよび再試行が生じました。

今回の更新で、2 つ目の並列応答が有効と認識されるようになりました。その結果、**glibc** DNS スタブリゾルバーは、認識されない応答による過剰なタイムアウトを回避します。

(BZ#1868106)

fgetsgent() および **fgetsgent_r()** で設定ファイルの読み取りがより堅牢になりました。

/etc/gshadow ファイルで特に構造化エントリー、または読み込み時のファイルサイズの変更によって、**fgetsgent()** および **fgetsgent_r()** 関数が無効なポインターを返すことがありました。そのため、これらの関数を使用して **/etc/gshadow**、または **/etc/** 内のその他の設定ファイルをセグメンテーションフォールトで読み取り、失敗していました。今回の更新で **fgetsgent()** および **fgetsgent_r()** が変更され、設定ファイルの読み込みがより堅牢となりました。その結果、アプリケーションが正常に設定ファイルを読み取るようになりました。

(BZ#1871397)

glibc 文字列は、AMD64 および Intel 64 プロセッサのシステムキャッシュに悪影響を及ぼさなくなりました。

以前では、文字列関数の **glibc** の実装は、64 ビット AMD および Intel プロセッサのスレッドで利用可能な最終レベルのキャッシュを誤って予測していました。その結果、大規模なバッファで **memcpy** 関数を呼び出すと、システムの全体的なキャッシュパフォーマンスに影響を与えるか、**memcpy** システムコールの速度が低下していました。

今回の更新で、システム内で報告されたハードウェアスレッド数で、最後のレベルのキャッシュサイズがスケーリングされなくなりました。その結果、文字列関数が大規模なバッファのキャッシュをバイパスし、残りのシステムキャッシュに悪影響を及ぼさなくなりました。

(BZ#1880670)

glibc 動的ローダーが、**libc.so.6** の特定の失敗を回避するようになりました。

以前は、**libc.so.6** 共有オブジェクトがメインプログラム (**glibc** バージョン情報を表示するなど) として実行されると、**glibc** 動的ローダーは、**LD_PRELOAD** 環境変数を使用してロードされたオブジェクトに関連する **libc.so.6** を正しく再配置しませんでした。そのため、**LD_PRELOAD** が設定されている場合、**libc.so.6** を呼び出すと、**libc.so.6** がセグメンテーションフォールトで予期せずに終了することがありました。今回の更新でバグが修正され、動的ローダーが **libc.so.6** の再配置を正しく処理するようになりました。その結果、上記の問題が発生しなくなりました。

(BZ#1882466)

glibc 動的リンカーが、静的スレッドローカルストレージ領域の一部を静的 TLS 割り当てに制限するようになりました。

以前は、**glibc** 動的リンカーは、動的 TLS に利用可能なすべての静的スレッドローカルストレージ (TLS) 空間を最初に使用していました。そのため、動的 TLS の割り当てで、利用可能なすべての静的 TLS 領域の消費がすでに使用されているため、**dlopen** 関数を使用してランタイム時に追加の共有オブジェクトを読み込むことが失敗することがありました。この問題は、特に 64 ビット ARM アーキテクチャーおよび IBM Power Systems で発生しました。

動的リンカーは、静的 TLS 領域の一部を静的 TLS 割り当てに制限し、動的 TLS の最適化にこの領域を使用しなくなりました。これにより、**dlopen** 呼び出しは、デフォルト設定でより多くのケースで成功します。デフォルト設定よりも多くの静的 TLS を必要とするアプリケーションでは、新しい **glibc.rtld.optional_static_tls** 調整可能パラメーターを使用できます。

(BZ#1871396)

glibc 動的リンカーが、64 ビット ARM バリエーション呼び出し規則の Lazy binding を無効にするようになりました。

以前は、**glibc** 動的リンカーは、64 ビット ARM(AArch64) バリエーション呼び出し規則を使用して、関数の Lazy バインディングを無効にしませんでした。その結果、このような関数呼び出しにおける動的リンカーの破損した引数により、誤った結果やプロセスが失敗する可能性があります。今回の更新で、上記のシナリオで動的リンカーがレイジーバインディングを無効にし、関数の引数が正しく渡されるようになりました。

(BZ#1893662)

gcc がバージョン 8.4 にリベースされました。

GNU コンパイラコレクション (GCC) がアップストリームバージョン 8.4 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#1868446)

7.12. ID 管理

Samba wide links 機能が VFS モジュールに変換されました

以前は、**wide links** パラメーターは **smbd** サービスのコア機能の一部でした。この機能を有効にすると安全ではないため、**widelinks** という名前の個別の仮想ファイルシステム (VFS) モジュールに移動されました。後方互換性を確保するために、RHEL 8.4 の Samba は、設定内に **wide links = yes** が設定されている共有用にこのモジュールを自動的に読み込みます。

重要: Red Hat は、安全ではない **wide links** 機能を使用しないことを推奨します。代わりに、**bind mount** を使用して、ファイル階層の一部を Samba で共有されるディレクトリーにマウントします。bind マウントの設定に関する詳細は、man ページの **mount(8)** の **Bind mount operation** セクションを参照してください。

bind mount に **wide links** を使用する設定に切り替えるには、次のコマンドを実行します。

1. ファイル共有以外のすべてのシンボリックリンクでは、リンクを **bind mount** に置き換えます。詳細は、man ページの **mount(8)** の **Bind mount operation** セクションを参照してください。
2. **/etc/samba/smb.conf** ファイルからすべての **wide links = yes** エントリーを削除します。
3. Samba をリロードします。

```
# smbcontrol all reload-config
```

(BZ#1925192)

ネットワーク接続のアイドルタイムアウトが、リソースエラーとして報告されなくなる

以前は、Directory Server は、アイドル状態のネットワーク接続がタイムアウトしたときにリソースが一時的に利用不可だったという誤解を招くエラーを報告していました。今回の更新で、ネットワーク接続のアイドルタイムアウトのエラーマクロが **EAGAIN** から **ETIMEDOUT** に変更され、タイムアウトを説明する正確なエラーメッセージが Directory Server アクセスログに書き込まれます。

(BZ#1859301)

PKI CA に接続された PKI ACME Responder が発行した証明書で OCSP 検証に失敗しなくなりました。

以前のバージョンでは、PKI CA が提供するデフォルトの ACME 証明書プロファイルには、実際の OCSP サービスを参照していないサンプル OCSP URL が含まれていました。これにより、PKI ACME Responder が PKI CA 発行者を使用するように設定されている場合に、この Responder が発行する証明書は OCSP 検証に失敗する可能性があります。今回の更新で、ACME 証明書プロファイルのハードコーディングされた URL が削除され、カスタマイズしない場合にプロファイル設定ファイルを修正するアップグレードスクリプトが追加されました。

(BZ#1868233)

7.13. グラフィックインフラストラクチャー

最新の Intel ノートパソコンで、ディスプレイバックライトが確実に機能するようになりました。

Intel CPU を使用する最近のラップトップでは、ディスプレイのバックライトを制御するプロプライエタリーインターフェイスが必要です。以前は、RHEL はプロプライエタリーインターフェイスに対応しておらず、ノートパソコンでは信頼できない VESA インターフェイスの使用を試みていました。そのため、RHEL は、このノートパソコンでディスプレイのバックライトを制御しませんでした。

今回の更新で、RHEL はプロプライエタリーバックライトインターフェイスへの対応が追加され、結果としてディスプレイ制御が期待どおりに機能するようになりました。

(BZ#1885406)

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

tests_luks.yml により NVME ディスクでパーティションケースが失敗することがなくなりました。

以前は、NVME ディスクは `virtio/scsi` で使用されるパーティション命名規則とは異なるパーティション命名規則を使用しており、Storage ロールはこれを反映しませんでした。その結果、NVME ディスクで Storage ロールを実行すると、クラッシュしていました。今回の修正により、Storage RHEL システムロールが `blivet` モジュールからパーティション名を取得するようになりました。

(BZ#1865990)

selinux RHEL システムロールが、`present` という名前の変数を使用しなくなりました。

以前は、`selinux` RHEL システムロールにある一部のタスクは、文字列 `present` ではなく、`present` という名前の変数を使用していました。これにより、`selinux` の RHEL システムロールは、変数 `present` がないことを示すエラーを返していました。今回の更新でこの問題が修正され、これらのタスクを変更して文字列 `present` を使用するようになりました。これにより、`selinux` の RHEL システムロールが期待どおりに機能し、エラーメッセージが表示されません。

(BZ#1926947)

rsyslog-gnutls パッケージがない場合の logging 出力が失敗しなくなりました。

logging RHEL システムロールがセキュアなりモート入出力とセキュアな転送出力を提供するように設定されている場合には、グローバル `tls rsyslog-gnutls` パッケージが必要です。以前のバージョンでは、`tls rsyslog-gnutls` パッケージが、以前のバージョンに無条件でインストールされるように変更されました。そのため、`tls rsyslog-gnutls` パッケージが管理ノードで利用できなくなると、`logging` の一部としてセキュアなりモート入力とセキュアな転送出力が含まれていなくても、ロギングロールの設定に失敗していました。今回の更新で、セキュアな接続が設定されているかどうかを確認し、グローバ

ル `tls logging_pki_files` 変数をチェックし、この問題を修正しています。`rsyslog-gnutls` パッケージは、セキュアな接続が設定されている場合にのみインストールされます。その結果、欠落している `rsyslog-gnutls` パッケージでロギング出力が失敗しなくなったため、`elasticsearch` を統合するように Red Hat Enterprise Virtualization Hypervisor を設定する操作になりました。

(BZ#1927943)

7.15. 仮想化

Windows Server 2019 ホストの RHEL 8 ゲストコンソールへの接続の速度が低下することがなくなりました。

以前では、Windows Server 2019 ホストで、RHEL 8 をマルチユーザーモードでゲストオペレーティングシステムとして使用すると、ゲストのコンソール出力へ接続するのに想定よりもはるかに長い時間がかかっていました。今回の更新で、Hyper-V ハイパーバイザー上の VRAM のパフォーマンスが改善され、この問題が修正されます。

(BZ#1908893)

QXL で、Wayland を使用する仮想マシンの複数のモニターを表示できるようになりました。

以前では、`remote-viewer` ユーティリティーを使用して、Wayland ディスプレイサーバーを使用している仮想マシンのモニターを複数表示すると、仮想マシンが応答しなくなり、`Waiting for display` というステータスメッセージが永久に表示されていました。基礎となるコードが修正され、上記の問題が発生しなくなりました。

(BZ#1642887)

7.16. クラウド環境の RHEL

ハイバネート後に GPU 最適化の Azure インスタンスが正しく動作するようになりました。

NV6 などの GPU 最適化仮想マシン (VM) のサイズを使用する Microsoft Azure インスタンスで RHEL 8 をゲストオペレーティングシステムとして実行している場合、以前は仮想マシンの GPU が正常に機能しませんでした。これが発生すると、カーネルは以下のメッセージをログに記録します。

```
hv_irq_unmask() failed: 0x5
```

今回の更新で、Microsoft Azure の影響を受ける仮想マシンが再開後に GPU を正しく処理し、問題が発生しなくなりました。

(BZ#1846838)

TX/RX パケットカウンターは、仮想マシンがハイバネートから再開した後に想定どおりに増大します。

以前は、CX4 VF NIC を使用する RHEL 8 仮想マシンが Microsoft Azure でハイバネートから再開すると、TX/RX パケットカウンターの増加は停止していました。今回の更新で問題が解決し、パケットカウンターが意図された通りに増大するようになりました。

(BZ#1876527)

RHEL 8 仮想マシンが Azure でハイバネートからの再開に失敗しなくなりました。

以前は、`SR-IOV` が有効になっている RHEL 8 仮想マシンが Microsoft Azure でハイバネートし、割り当て解除すると、Virtual Function (VF)、`vmbus` デバイスの GUID が変更になりました。したがって、仮

想マシンが再起動すると、再開に失敗し、予期せずに終了していました。今回の更新で、**vmbus** デバイスの VF が変更されなくなり、仮想マシンがハイバネートから正常に再開されるようになりました。

(BZ#1876519)

Hyper-V および KVM ゲストで冗長エラーメッセージを削除

以前は、KVM または Hyper-V 仮想マシンで RHEL 8 ゲストオペレーティングシステムを実行している場合は、以下のエラーメッセージが **/var/log/messages** ファイルで報告されていました。

```
serial8250: too much work for irq4
```

これは冗長なエラーメッセージであり、削除されました。

問題の詳細は、[Red Hat ナレッジベースソリューション](#) を参照してください。

(BZ#1919745)

7.17. コンテナ

podman system connection add がデフォルトの接続を自動的に設定

以前は、**podman system connection add** コマンドは、最初の接続をデフォルト接続に自動的に設定しませんでした。したがって、**podman system connection default <connection_name>** コマンドを手動で実行して、デフォルトの接続を設定する必要があります。今回の更新で、**podman system connection add** コマンドが期待どおりに機能するようになりました。

(BZ#1881894)

podman run --pid=host はルートレスモードで動作します

以前は、rootless ユーザーとして **podman run --pid=host** コマンドを実行しても機能しませんでした。その結果、OCI パーミッションエラーが発生しました。

```
$ podman run --rm --pid=host quay.io/libpod/testimage:20200929 cat -v /proc/self/attr/current
```

```
Error: container_linux.go:370: starting container process caused: process_linux.go:459: container init caused: readonly path /proc/bus: operation not permitted: OCI permission denied
```

今回の更新で、この問題が修正されています。

(BZ#1940854)

第8章 テクノロジープレビュー

本パートでは、Red Hat Enterprise Linux 8.4 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に対する Red Hat のサポート範囲の詳細は、[テクノロジープレビューのサポート範囲](#) を参照してください。

8.1. インストーラーおよびイメージの作成

Red Hat Connector がテクノロジープレビューとして利用可能になりました。

Red Hat Insights とサブスクリプションコンテンツを使用できるように、1つのコマンドを使用して RHEL システムに接続できるようになりました。Red Hat Enterprise Linux 8.4 ではテクノロジープレビューとして提供され、Red Hat コネクター (**rhc**) CLI は登録体験を通知し、**subscription-manager** および **insights-client** コマンドを個別に実行して Red Hat に接続する必要がなくなります。Red Hat コネクターと Smart Management サブスクリプションでは、クラウドから直接問題を修正することもできます。

詳細は [Red Hat Connector Configuration Guide](#) を参照してください。

(BZ#1957316)

8.2. ネットワーク

UDP トンネルで MPLS トラフィックをテクノロジープレビューとして追加するための bareudp デバイスサポートの導入

bareudp デバイスのサポートが、テクノロジープレビューとして **ip link** コマンドで利用できるようになりました。**bareudp** デバイスは、UDP トンネル内の unicast および multicast multi protocol label switching (MPLS) および IPv4/IPv6 などの異なる L3 プロトコルでトラフィックをルーティングするための L3 カプセル化サポートを提供します。**tc** フィルターおよびアクションの追加で、UDP で MPLS パケットのルーティングを開始できます。

たとえば、新規の **bareudp** デバイスを作成するには、以下のコマンドを使用します。

```
# ip link add dev bareudp0 type bareudp dstport 6635 ethertype mpls_uc
```

bareudp0 デバイスを使用して UDP トンネルの MPLS 受信パケットをルーティングするには、次のコマンドを使用します。

```
# tc qdisc add dev enp1s0 ingress
# tc filter add dev enp1s0 ingress proto mpls_uc matchall \
> action tunnel_key set src_ip 2001:db8::22 dst_ip 2001:db8::21 id 0 \
> action mirrored egress redirect dev bareudp0
```

bareudp デバイスの作成時に使用するオプションおよびパラメーターの詳細は、man ページの **ip-link(8)** の **Bareudp Type Support** セクションを参照してください。

(BZ#1849815)

AF_XDP がテクノロジープレビューとして利用可能に

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

(BZ#1633143)

KTLS がテクノロジープレビューとして利用可能に

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

(BZ#1570255)

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めます。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せん)。
- 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 mirrored 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、Botem 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)

Multipath TCP サポート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。

マルチパス TCP (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 8.4 では、以下のような追加の機能を提供します。

- 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)

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

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

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

([BZ#1906489](#))

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、RHEL 8.4 の 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。

SGX (Software Guard Extensions) 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。本リリースでは、SGX v1 および v1.5 のカーネルサポートを開始します。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、実行時コンパイラーでネイティブマシンコードに変換されます。**bpf()** は、root ユーザーなど、**CAP_SYS_ADMIN** が付与されているユーザーのみが利用できます。詳細は、man ページの **bpf(2)** を参照してください。

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

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

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

- **bpfftrace**。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りました。

現時点で、カーネルのデータストリーミングナビゲーター (DSA) ドライバーがテクノロジープレビューとして利用できます。DSA 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)

8.4. ファイルシステムおよびストレージ

NVMe/TCP がテクノロジープレビューとして利用可能になりました。

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

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

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

(BZ#1696451)

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

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

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

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

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

また、OverlayFS の使用には、以下のルールと制限が適用されます。

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

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

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

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

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

- SELinux セキュリティーラベルは、OverlayFS で対応するすべてのコンテナエンジンでデフォルトで有効になっています。
- このリリースの既知の問題は、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)

Stratis がテクノロジープレビューとして利用可能になりました。

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

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

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

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

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)

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

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

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

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

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

(JIRA:RHELPLAN-13195)

8.5. 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テクノロジープレビューとして利用可能になりました。

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。Red Hat Enterprise Linux 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ます。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** 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)

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

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

(BZ#1619620)

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

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

(BZ#1784200)

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

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

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

サービスを適切に引き継ぐことができないことを事前に把握できる場合は、ノードがピアをフェンシングする意味がないのであれば、ユーザーは特に2ノードクラスターでこのエージェントを使用できま

す。たとえば、ネットワークアップリンクに到達してサービスがクライアントに到達できない場合は、ノードがサービスを引き継ぐ意味はありません。これは、ルーターへの ping が検出できる状況が考えられます。

(BZ#1775847)

8.6. ID 管理

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

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

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

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

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

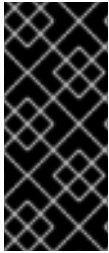
統合 DNS のある IdM サーバーは、DNSSEC を使用して、他の DNS サーバーから取得した DNS 回答を検証することに注意してください。これが、推奨される命名方法に従って設定されていない DNS ゾーンの可用性に影響を与える可能性があります。

(BZ#1664718)

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 **acmelPAServerCert** 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)

Intel Tiger Lake グラフィックがテクノロジープレビューとして利用可能に

Intel Tiger Lake UP3 および UP4 Xe グラフィック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。

Intel Tiger Lake グラフィックでハードウェアアクセラレーションを有効にするには、カーネルコマンドラインに以下のオプションを追加します。

```
i915.force_probe=pci-id
```

このオプションでは、**pci-id** を以下のいずれかに置き換えます。

- Intel GPU の PCI ID
- すべての高品質ハードウェアで **i915** ドライバーを有効にする * 文字

(BZ#1783396)

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。

- フェンシングなしおよびリソースを使用しないクラスターの設定
- 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

(BZ#1893743)

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

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

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

postfix ロールは、テクノロジープレビューとして利用可能です。

以下のロールが完全にサポートされています。

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

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

(BZ#1812552)

8.10. 仮想化

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

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

(BZ#1519039)

KVM 仮想マシンの AMD SEV。

テクノロジープレビューとして、RHEL 8 に、KVM ハイパーバイザーを使用する AMD EPYC ホストマシン用のセキュア暗号化仮想化 (SEV) 機能が同梱されます。仮想マシンで有効になっている場合は、ホ

ホストが仮想マシンのデータにアクセスできないように、SEV が仮想マシンメモリを暗号化します。ホストがマルウェアに感染した場合は、これにより仮想マシンのセキュリティが向上します。

1台のホストでこの機能を同時に使用できる仮想マシンの数は、ホストのハードウェアによって決まります。現在の AMD EPYC プロセッサは、SEV を使用して 509 台以下の稼働中の仮想マシンに対応します。

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

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

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

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

Intel vGPU

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

選択した Intel GPU のみが vGPU 機能と互換性があることに注意してください。

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、**ixgbevf** および **ixgbevf** ドライバーがサポートする Intel ネットワークアダプターに、シングルルート I/O 仮想化 (SR-IOV) 機能を使用することができるようになりました。この機能は、以下の条件が満たされると有効になります。

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

- NIC 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 がテクノロジーレビューとして利用可能になりました。

テクノロジーレビューとして、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を確保できます。

(BZ#1904496)

8.11. コンテナ

CNI プラグインはテクノロジーレビューとして Podman で利用可能になりました。

CNI プラグインは、テクノロジーレビューとして Podman ルートモードで使用できるようになりました。この機能を有効にするには、ユーザーは独自のルートレス CNI インフラストラクチャーコンテナイメージをビルドする必要があります。

(BZ#1932083)

crun がテクノロジーレビューとして利用可能になりました。

crun OCI ランタイムが、**container-tools:rhel8** モジュールがテクノロジーレビューとして利用できるようになりました。**crun** コンテナランタイムは、コンテナ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をサポートします。これは、setgid が設定されたディレクトリーまたはユーザーがグループアクセスのみを持つディレクトリーにおけるボリュームマウントに役立ちます。現在、**crun** または **runc** ランタイムはいずれも **cgroupsv2** に完全に対応していません。

(BZ#1841438)

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

registry.redhat.io/rhel8/podman コンテナイメージは、**podman** パッケージをコンテナ化した実装です。**podman** ツールは、コンテナおよびイメージの管理、それらのコンテナにマウントされたボリューム、およびコンテナのグループから作成された Pod を管理するために使用されます。

(JIRA:RHELPLAN-56659)

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、テクノロジーレビューとしてのみ利用可能です。代わりに、コマンドラインから直接 Podman を実行してください。

(JIRA:RHELDPCS-16861)

第9章 非推奨になった機能

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。

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

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

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

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

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

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。このコマンドに関連するセクションは、[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 の以前のバックエンド `lorax-composer` は非推奨とみなされます。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提供され、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` コマンドが非推奨となりました。Red Hat Enterprise Linux の今後のリリースでこのコマンドを実行すると、エラーが発生します。代わりに `rpmsign` コマンドを使用することが推奨されます。

(BZ#1688849)

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)

`curl` の Metalink サポートが無効になりました。

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

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

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

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

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

以下に例を示します。

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

(BZ#1999620)

9.4. セキュリティー

NSS SEED 暗号が非推奨になりました。

Mozilla Network Security Services (**NSS**) ライブラリーでは、今後のリリースで SEED 暗号化を使用する TLS 暗号スイートのサポートがなくなります。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ナレッジベースの記事 [Strong crypto defaults in RHEL 8 and deprecation of weak crypto algorithms](#) および man ページの [update-crypto-policies\(8\)](#) を参照してください。

(BZ#1660839)

RHEL 8 で DSA が非推奨になりました。

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

(BZ#1646541)

NSS で SSL2 Client Hello が非推奨に

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

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

(BZ#1645153)

TPM 1.2 が非推奨になりました。

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

(BZ#1657927)

/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)

ipa SELinux モジュールが selinux-policy から削除されました。

ipa SELinux モジュールは、維持されなくなったため、**selinux-policy** パッケージ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またはインターフェイスを使用する必要がある場合は、**ipa-selinux** パッケージ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なくなります。したがって、このツールは新規のデプロイメントには推奨されません。このパッケージの代わりとして、Red Hat は **perf** コマンドラインツールの使用を推奨します。

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 [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)

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。この取り組みの詳細は、[多様性を受け入れるオープンソースの強化](#) を参照してください。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:RHELDPCS-17641)

9.6. カーネル

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

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

(BZ#1748980)

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. プラットフォームの有効化

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)

9.8. ファイルシステムおよびストレージ

elevator カーネルコマンドラインパラメーターが非推奨になりました。

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

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

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

詳しい情報は、[ディスクスケジューラーの設定](#) を参照してください。

(BZ#1665295)

LVM mirror が非推奨化されました。

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

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

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

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

(BZ#1827628)

peripety が非推奨になりました。

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)

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)

9.9. 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。

(BZ#1851335)

9.10. コンパイラーおよび開発ツール

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、**--allowerase** オプションを **dnf** に渡して **gdb.i686** を削除し、**gdb.x86_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)

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)

9.11. 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_ssh_authorizedkeys** 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_ssh_authorizedkeys 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_ssh_authorizedkeys** を使用できるようにするには、man ページの **sss_ssh_authorizedkeys(1)** に記載されているように、**AuthorizedKeysCommand** **/usr/bin/sss_ssh_authorizedkeys** 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。キー変更の手順については、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** は含めません。
4. 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 の時点では、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 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ず。
- **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)

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ドメインコントローラー (PDC) として実行し、バックアップドメインコントローラー (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ー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ー (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 が非推奨に

winbind サービスを実行せずに、Samba **smbd** サービスが AD からユーザーおよびグループ情報を取得できるように、**libwbclient** パッケージの SSSD 実装が追加されました。Samba では、**winbind** サービスが実行しており、AD との通信を処理する必要があるため、セキュリティ上の理由から、関連するコードが **smbd** から削除されました。この追加機能は SSSD の一部ではなく、新しいバージョンの Samba では **libwbclient** の SSSD 実装を使用することはできません。そのため、**libwbclient** の SSSD 実装は非推奨になりました。

(BZ#1881992)

SMB1 プロトコルは Samba では非推奨に

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

セキュリティ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で SMB1 が無効になっています。

Jira:RHELDPCS-16612

9.12. 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標準に準拠する代替ライブラリーです。

(BZ#1607766)

9.13. 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)

9.14. 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)

9.15. 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ー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)

9.16. 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 でも引き続きサポートされますが、Red Hat 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)

virt-manager が非推奨になりました。

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

(JIRA:RHELPLAN-10304)

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

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

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

(BZ#1686057)

Cirrus VGA 仮想 GPU タイプが非推奨に

Red Hat Enterprise Linux の将来のメジャー更新では、KVM 仮想マシンで Cirrus VGA GPU デバイスに対応しなくなります。したがって、Red Hat では、Cirrus VGA の代わりに `stdvga` または `virtio-vga` デバイスの使用を推奨しています。

(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)

9.17. コンテナ

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

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

- 389-ds-base-legacy-tools
- authd
- custodia
- firewire
- geoipupdate
- hostname
- isl
- isl-devel
- libavc1394
- libdc1394
- libdwarf
- libdwarf-devel
- libdwarf-static
- libdwarf-tools
- libidn
- libpng12
- libraw1394
- lorax-composer
- mailman
- mailx - replaced by s-nail
- mercurial
- ncompress
- net-tools
- netcf
- netcf-libs
- network-scripts
- nss_nis
- nss-pam-ldapd

- openssh-ldap
- parfait
- peripety
- perl-prefork
- perl-Sys-Virt
- python3-nose
- python3-pymongo
- python3-pytoml - python3-toml に置き換え
- python3-virtualenv - 代わりに Python 3 で **venv** モジュールを使用します。
- redhat-support-lib-python
- redhat-support-tool
- scala
- sendmail
- yp-tools
- ypbind
- ypserv
- xdelta
- xinetd

9.19. 非推奨のデバイス

このセクションでは、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。

PCI ID は、**vendor:device:subvendor:subdevice** の形式です。**subdevice** エントリーまたは **subvendor:subdevice** エントリーがリストにない場合は、そのような不明なエントリーの値を持つ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。

デバイスの種別	ドライバー	Device	デバイス ID
PCI	bnx2		
PCI	hpsa		0x103C:0x3239:0x103C:0x21C4

デバイスの種別	ドライバー	Device	デバイス ID
PCI	hpsa		0x103C:0x3239:0x103C:0x21C9
PCI	hpsa		0x103C:0x3239:0x103C:0x21CC
PCI	hpsa		0x103C:0x3239:0x103C:0x21CD
PCI	hpsa		0x103C:0x3239:0x103C:0x21CE
PCI	hpsa		0x103C:0x323a:0x103C:0x3233
PCI	hpsa		0x103C:0x323a:0x103C:0x3241
PCI	hpsa		0x103C:0x323a:0x103C:0x3243
PCI	hpsa		0x103C:0x323a:0x103C:0x3245
PCI	hpsa		0x103C:0x323a:0x103C:0x3247
PCI	hpsa		0x103C:0x323a:0x103C:0x3249
PCI	hpsa		0x103C:0x323a:0x103C:0x324A
PCI	hpsa		0x103C:0x323a:0x103C:0x324B
PCI	hpsa		0x103C:0x323b:0x103C:0x3350
PCI	hpsa		0x103C:0x323b:0x103C:0x3351
PCI	hpsa		0x103C:0x323b:0x103C:0x3352
PCI	hpsa		0x103C:0x323b:0x103C:0x3353
PCI	hpsa		0x103C:0x323b:0x103C:0x3354
PCI	hpsa		0x103C:0x323b:0x103C:0x3355
PCI	hpsa		0x103C:0x323b:0x103C:0x3356
PCI	hpsa		0x103C:0x333f:0x103c:0x333f
PCI	hpsa		0x9005:0x0290:0x9005:0x0580
PCI	hpsa		0x9005:0x0290:0x9005:0x0581

デバイスの種別	ドライバー	Device	デバイス ID
PCI	hpsa		0x9005:0x0290:0x9005:0x0582
PCI	hpsa		0x9005:0x0290:0x9005:0x0583
PCI	hpsa		0x9005:0x0290:0x9005:0x0584
PCI	hpsa		0x9005:0x0290:0x9005:0x0585
PCI	lpfc		0x10df:0x0724
PCI	lpfc		0x10df:0xe200
PCI	lpfc		0x10df:0xe220
PCI	lpfc		0x10df:0xf011
PCI	lpfc		0x10df:0xf015
PCI	lpfc		0x10df:0xf100
PCI	lpfc		0x10df:0xfc40
PCI	megaraid_sas		0x1000:0x005b
PCI	mpt3sas		0x1000:0x006E
PCI	mpt3sas		0x1000:0x0080
PCI	mpt3sas		0x1000:0x0081
PCI	mpt3sas		0x1000:0x0082
PCI	mpt3sas		0x1000:0x0083
PCI	mpt3sas		0x1000:0x0084
PCI	mpt3sas		0x1000:0x0085
PCI	mpt3sas		0x1000:0x0086
PCI	mpt3sas		0x1000:0x0087
PCI	myri10ge		

デバイスの種別	ドライバー	Device	デバイス ID
PCI	netxen_nic		
PCI	sfc		0x1924:0x0803
PCI	sfc		0x1924:0x0813
PCI	qla2xxx		0x1077:0x2031
PCI	qla2xxx		0x1077:0x2532
PCI	qla2xxx		0x1077:0x8031

第10章 既知の問題

このパートでは Red Hat Enterprise Linux 8.4 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

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

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

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

(BZ#1640697)

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

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

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

(BZ#1697896)

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)

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)

Anaconda でカスタムパーティションの暗号化が表示されない

システムのインストール時に **カスタム** のパーティション設定を選択すると、**Encrypt my data** データラジオボタンは利用できません。その結果、インストールの完了時にデータは暗号化されません。

この問題を回避するには、暗号化する各デバイスのカスタムパーティション設定画面で暗号化を設定します。Anaconda はダイアログを終了する際にパスフレーズの入力を要求します。

(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が自動パーティション設定を試みます

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と、キックスタートファイルでパーティションコマンドを指定しない場合でも、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す。このインストールプログラムは、キックスタートファイルで **autopart** コマンドを使用しているかのように動作し、予期しないパーティションが作成されます。この問題を回避するには、キックスタートファイルで **reqpart** コマンドを使用して、手動パーティション設定を対話的に設定できます。

(BZ#1954408)

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)

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)

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、GUI インストールプログラムにより、リポジトリメタデータの更新が開始されます。更新プロセスは、登録およびサブスクリプションプロセスの一部ではないため、Red Hat への接続ウィンドウで **登録解除** ボタンが有効になります。ネットワーク接続によっては、更新プロセスが完了するのに1分以上かかることがあります。更新プロセスが完了する前に **登録解除** ボタンをクリックすると、登録解除プロセスで、インストールプログラムが CDN との通信に必要な証明書と CDN リポジトリファイルが削除されるため、GUI インストールが失敗する可能性があります。

この問題を回避するには、Red Hat への接続ウィンドウで **登録** ボタンをクリックした後に、GUI インストールで次の手順を実行します。

1. Red Hat への接続画面から **完了** をクリックして、**インストールの概要** 画面に戻ります。
2. **インストールの概要** ウィンドウで、**インストールソース** および **ソフトウェアの選択** の斜体のステータスメッセージに処理情報が表示されていないことを確認します。
3. インストールソースとソフトウェアの選択のカテゴリーが準備できたら、Red Hat への接続をクリックします。
4. **登録解除** ボタンをクリックします。

これらの手順を完了したら、GUI のインストール時にシステムの登録を安全に解除できます。

(BZ#1821192)

複数の組織に属するユーザーアカウントの登録に失敗していました

現在、複数の組織に属するユーザーアカウントでシステムを登録しようとする、登録プロセスが失敗し、**You must specify an organization for new units**(新しいユニットの組織を指定する必要があります)。というメッセージが表示されます。

この問題を回避するには、以下のいずれかを行います。

- 複数の組織に属さない別のユーザーアカウントを使用します。
- GUI および Kickstart インストールの **Connect to Red Hat** 機能で利用できる **アクティベーションキー** 認証方法を使用します。
- **Connect to Red Hat** の登録手順を省略し、**Subscription Manager** を使用してインストール後にシステムを登録します。

(BZ#1822880)

グラフィカルインストーラーの使用時に、Red Hat Insights クライアントがオペレーティングシステムの登録に失敗する

現在、Insights クライアントを示すエラーでインストールに失敗します。

この問題を回避するには、インストーラーでシステムを登録する前に、**Connect to Red Hat** ステップで、**Connect to Red Hat Insights** プシジョンの選択を解除します。

その結果、以下のコマンドを使用してインストールを完了し、その後 Insights に登録できます。

```
# insights-client --register
```

(BZ#1931069)

autopart ユーティリティーを使用したインストールは、一貫性のないディスクセクターサイズで失敗します

複数の整合性のないディスクセクターサイズを持つ **autopart** を使用した RHEL のインストールに失敗します。回避策として、デフォルトの **LVM** スキームではなく、**autopart --type=plain** などの **plain** パーティションスキームを使用します。もう1つのオプションとして、たとえば **hdparm --set-sector-size=<SIZE> <DEVICE>** を実行して、セクターサイズの再設定を試行することがあげられます。

キックスタートインストールの回避策として、以下を実行します。

- **ignoredisk --drives=..** を指定して、パーティション設定に使用するディスクを制限します。または **--only-use=..** です。
- 作成した各 LVM 物理ボリュームに使用するディスクを指定します: **partition pv.1 --ondisk=..**

手動インストールの回避策として、以下を実行します。

- グラフィカルモードまたはテキストモードでの手動インストール中に、同じセクターサイズのディスクのみを選択します。
- 一貫性のないセクターサイズを持つディスクがインストールに選択されている場合は、作成された各 LVM ボリュームグループが同じセクターサイズの物理ボリュームを使用するように制限します。これは、カスタムパーティション設定スポークのグラフィカルモードでのみ実行できます。

(BZ#1935722)

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)

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)

rhel-guest-image-8.4 イメージで grub2- mkconfig を使用して grub.cfg を再構築できない

rhel-guest-image-8.4 タイプには、**/etc/default/grub** ファイルに **GRUB_DEFAULT=saved** エントリは含まれません。したがって、新しいカーネルをインストールし、**grub2 -mkconfig -o /boot/grub2/grub.cfg** コマンドを使用して **grub** を再構築すると、再起動後に新しいカーネルでシステムが起動しません。この問題を回避するには、**GRUB_DEFAULT=saved** を **/etc/default/grub** ファイルに追加します。その結果、システムは新しいカーネルで起動するはずですが、

(BZ#2227218)

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. 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)

10.4. セキュリティー

ユーザーは、ロックされたユーザーとして **sudo** コマンドを実行できます。

ALL キーワードで **sudoers** パーミッションが定義されているシステムでは、パーミッションを持つ **sudo** ユーザーは、アカウントがロックされているユーザーとして **sudo** コマンドを実行できます。そのため、ロックされたアカウントと期限切れのアカウントを使用して、コマンドを実行し続けることができます。

この問題を回避するには、**/etc/shells** 内の有効なシェルの適切な設定と併せて、新たに実装した **runas_check_shell** オプションを有効にします。これにより、攻撃者が **bin** などのシステムアカウントでコマンドを実行するのを防ぎます。

(BZ#1786990)

libselinux-python は、そのモジュールからのみ利用可能

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

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

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


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

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

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

(BZ#1666328)

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

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

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

(BZ#1763210)

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/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)

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

/etc/selinux/config で **SELINUX=disabled** オプションを使用して SELinux を無効にすると、カーネルが SELinux を有効にして起動し、その後のブートプロセスで無効化モードに切り替わります。これにより、メモリーリークが生じる可能性があります。

この問題を回避するには、SELinux を完全に無効にする必要がある場合に [SELinux の使用のシステム起動時に SELinux モードの変更](#) で説明されているように、**selinux=0** パラメーターをカーネルコマンドラインに追加して SELinux を無効にすることが推奨されます。

(JIRA:RHELPLAN-34199)

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 Camellia 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)

SHA-1 署名を使用するサーバーへの接続が GnuTLS で動作しません。

証明書の SHA-1 署名は、GnuTLS セキュアな通信ライブラリーにより、セキュアでないものとして拒否されます。したがって、TLS のバックエンドとして GnuTLS を使用するアプリケーションは、このような証明書を提供するピアへの TLS 接続を確立することができません。この動作は、その他のシステム暗号化ライブラリーと一貫性がありません。

この問題を回避するには、サーバーをアップグレードして、SHA-256 または強力なハッシュを使用し、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)

Libreswan は、leftikeport および rightikeport を無視します。

Libreswan は、host-to-host Libreswan 接続では **leftikeport** および **rightikeport** オプションを無視します。これにより、Libreswan は、**leftikeport** と **rightikeport** の設定に関係なく、デフォルトのポートを使用します。現在利用できる回避策はありません。

(BZ#1934058)

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)

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)

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ません。

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](#))

systemd が任意のパスからコマンドを実行できない

SELinux ポリシーパ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](#))

selinux-policy により、IPsec が TCP で機能しないようにします。

RHEL 8.4 の **libreswan** パッケージは、TCP カプセル化を使用した IPsec ベースの VPN に対応します。ただし、**selinux-policy** パッケージでは、この更新が反映されません。これにより、Libreswan が TCP を使用するよう設定すると、**ipsec** サービスは指定の TCP ポートにバインドできません。

この問題を回避するには、カスタムの SELinux ポリシーを使用します。

1. テキストエディターで新しい **.cil** ファイルを開きます。以下に例を示します。

```
# vim local_ipsec_tcp_listen.cil
```

2. 以下のルールを挿入します。

```
(allow ipsec_t ipsecnat_port_t (tcp_socket (name_bind name_connect)))
```

3. ファイルを保存してから閉じます。
4. ポリシーモジュールをインストールします。

```
# semodule -i local_ipsec_tcp_listen.cil
```

5. **ipsec** サービスを再起動します。

```
# systemctl restart ipsec
```

これにより、Libreswan は、一般的に使用される **4500/tcp** ポートにバインドおよび接続できます。

([BZ#1931848](#))

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](#))

CIS プロファイルで rpm_verify_permissions が失敗する

rpm_verify_permissions ルールでは、ファイルパーミッションがパッケージのデフォルトパーミッションと比較されます。ただし、**scap-security-guide** パッケージで提供される Center for Internet Security (CIS) プロファイルでは、一部のファイルパーミッションがデフォルトよりも厳格なものに変更されます。その結果、**rpm_verify_permissions** を使用した特定ファイルの検証が失敗します。

この問題を回避するには、これらのファイルに以下のパーミッションがあることを手作業で確認します。

- **/etc/cron.d** (0700)
- **/etc/cron.hourly** (0700)
- **/etc/cron.monthly** (0700)
- **/etc/crontab** (0600)
- **/etc/cron.weekly** (0700)
- **/etc/cron.daily** (0700)

([BZ#1843913](#))

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

キックスタートは、**com_redhat_oscaped** ではなく、**org_fedora_oscaped** 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](#))

SSG における相互依存ルールの特定のセットが失敗する可能性があります。

ルールとその依存関係の順序付けを定義しないため、ベンチマークの **SCAP Security Guide (SSG)** ルールの修正が失敗する可能性があります。たとえば、特定の順番で複数のルールを実行する必要がある場合、あるルールがコンポーネントをインストールし、別のルールが同じコンポーネントを設定した場合すると、それらは正しくない順序で実行される可能性があり、修正によってエラーが報告されます。この問題を回避するには、修正を再実行して、番目の実行で依存ルールを修正します。

([BZ#1750755](#))

OSCAP Anaconda Addon がすべてのパッケージをテキストモードでインストールしません。

OSCAP Anaconda Addon プラグインは、インストールがテキストモードで実行している場合、システムインストーラーによってインストールに選択されているパッケージのリストを変更することはできません。これにより、キックスタートを使用してセキュリティーポリシープロファイルが指定され、インストールがテキストモードで実行している場合に、インストール中にセキュリティーポリシーに必要な追加パッケージがインストールされません。

この問題を回避するには、グラフィカルモードでインストールを実行するか、キックスタートファイルの **%packages** セクションにあるセキュリティーポリシーで、セキュリティーポリシープロファイルに必要なパッケージをすべて指定します。

これにより、セキュリティーポリシープロファイルで必要となるパッケージは、上記の回避策のいずれかを行わなければ RHEL インストールインストール時にインストールされません。また、インストール後のシステムは、指定のセキュリティーポリシープロファイルと互換性がありません。

([BZ#1674001](#))

oscap Anaconda Addon がカスタムプロファイルを正しく処理しません。

OSCAP Anaconda Addon プラグインは、個別のファイルでカスタマイズを使用したセキュリティープロファイルを適切に処理しません。これにより、対応する Kickstart セクションで適切に指定しても、RHEL グラフィカルインストールでカスタマイズしたプロファイルは利用できません。

この問題を回避するには、ナレッジベースの記事 [Creating a single SCAP data stream from an original DS and a tailoring file](#) を参照してください。この回避策により、RHEL グラフィカルインストールでカスタマイズした SCAP プロファイルを使用できます。

([BZ#1691305](#))

キックスタートインストール時のサービス関連のルールの修正が失敗する場合があります。

キックスタートのインストール時に、OpenSCAP ユーティリティーで、サービス **enable** または **disable** 状態の修正が必要でないことが誤って表示されることがあります。これにより、OpenSCAP が、インストール済みシステムのサービスを非準拠状態に設定する可能性があります。回避策として、キックスタートインストール後にシステムをスキャンして修復できます。これにより、サービス関連の問題が修正されます。

([BZ#1834716](#))

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)

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)

10.5. 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ブ化します。

```
# 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)

GRO が無効になっていると、IPsec オフロード中に IPsec ネットワークトラフィックが失敗します。

デバイスで汎用受信オフロード (GRO) が無効になっていると、IPSec オフロードは機能しません。IPsec オフロードがネットワークインターフェイスで設定され、GRO がそのデバイスで無効になっていると、IPsec ネットワークトラフィックに失敗します。

この問題を回避するには、デバイスで GRO を有効にしたままにします。

(BZ#1649647)

10.6. カーネル

特定の BCC ユーティリティーで安全性に問題のない警告を表示する

一部のコンパイラ固有のカーネルヘッダーでのマクロの再定義が原因一部の BPF Compiler Collection(BCC) ユーティリティーでは、以下の警告が表示されます。

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

この警告は安全性に問題がないため、無視することができます。

(BZ#1907271)

vmcore キャプチャーはメモリーのホットプラグまたはアンプラグの操作後に失敗します。

メモリーのホットプラグまたはホットアンプラグ操作の実行後に、メモリーのレイアウト情報を含むデバイスツリーを更新するとイベントが発生します。これにより、**makedumpfile** ユーティリティーは存在しない物理アドレスにアクセスしようとします。以下の条件を満たすと問題が発生します。

- IBM Power System (little endian) で RHEL 8 を実行する。
- システムで **kdump** サービスまたは **fadump** サービスが有効になっている。

このような場合に、メモリーホットプラグまたはホットアンプラグの操作後にカーネルクラッシュが発生すると、カーネルのキャプチャーで **vmcore** の保存に失敗します。

この問題を回避するには、ホットプラグまたはホットアンプラグ後に **kdump** サービスを再起動します。

```
# systemctl restart kdump.service
```

これにより、上記のシナリオで **vmcore** が正常に保存されます。

(BZ#1793389)

kdump が SSH または NFS ダンプターゲットで vmcore のダンプに失敗する

新しいバージョンの **dracut-network** は、**ipcalc** を必要とする **dhcp-client** の依存関係を破棄します。したがって、NIC ポートが静的 IP に設定され、**kdump** が SSH または NFS のダンプ出力先でダンプするように設定されると、**kdump** が失敗し、以下のエラーメッセージが表示されます。

```
ipcalc: command not found
```

この問題を回避するには、以下を実行します。

1. **ipcalc** パッケージを手動でインストールします。

```
dnf install ipcalc
```

2. **kdump** 用の **initramfs** を再構築します。


```
kdumpctrl rebuild
```

3. **kdump** サービスを再起動します。

```
systemctl restart kdump
```

これにより、上記のシナリオで **kdump** が正常に実行されます。

(BZ#1931266)

RHEL 8 で、デバッグカーネルがクラッシュキャプチャー環境で起動に失敗します。

デバッグカーネルのメモリー要求の性質により、デバッグカーネルが使用中で、カーネルパニックが発生すると、問題が発生します。その結果、デバッグカーネルはキャプチャーカーネルとして起動できず、代わりにスタックトレースが生成されます。この問題を回避するには、クラッシュカーネルメモリーを適宜増やします。これにより、デバッグカーネルが、クラッシュキャプチャー環境で正常に起動します。

(BZ#1659609)

起動時にクラッシュカーネルのメモリー割り当てに失敗する

一部の Ampere Altra システムでは、BIOS 設定で 32 ビットのリージョンが無効になっていると、メモリー割り当てが失敗します。したがって、従来のメモリーではメモリー割り当てを確保するのに十分な大きさがいないため、**kdump** サービスの起動に失敗します。

この問題を回避するには、以下のように BIOS で 32 ビット CPU 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、クラッシュカーネルは 32 ビット領域内でメモリーを割り当て、**kdump** サービスが期待どおりに動作します。

(BZ#1940674)

特定のカーネルドライバーがそのバージョンを表示しません

RHEL 8.4 では、多くのネットワークカーネルドライバーに対応する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バージョンを表示しません。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**を表示します。この問題を回避するには、ユーザーは以下のコマンドを実行できます。

```
# modinfo <AFFECTED_DRIVER> | grep rhelversion
```

その結果、必要に応じて、影響を受けるカーネルドライバーのバージョンを判断できます。

ドライバーバージョンの文字列の変更量は、ドライバー自体の変更量には実際に行われないことに注意してください。

(BZ#1944639)

irqpoll を使用すると vmcore の生成に失敗する

Amazon Web Services (AWS) クラウドプラットフォームで実行している 64 ビット ARM アーキテクチャー上には **nvme** ドライバーの既存の問題があります。この問題により、最初のカーネルに **irqpoll** カーネルコマンドラインパラメーターを指定すると **vmcore** の生成に失敗します。したがって、カーネルクラッシュ後に **vmcore** が `/var/crash/` ディレクトリーにダンプされません。この問題を回避するには、以下を実行します。

1. `/etc/sysconfig/kdump` ファイルの **KDUMP_COMMANDLINE_REMOVE** 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. **systemctl restart kdump** コマンドを実行して、**kdump** サービスを再起動します。

その結果、最初のカーネルが正常に起動し、カーネルクラッシュ時に **vmcore** がキャプチャーされることが予想されます。

kdump サービスは、大量のクラッシュカーネルメモリーを使用して **vmcore** ファイルをダンプできることに注意してください。キャプチャーカーネルには、**kdump** サービス用のメモリーが十分あることを確認します。

(BZ#1654962)

HP NMI ウォッチドッグが常にクラッシュダンプを生成しない

特定の場合において、HP NMI ウォッチドッグの **hpwdt** ドライバーは、マスク不可割り込み (NMI) が **perfmon** ドライバーにより使用されたため、HPE ウォッチドッグタイマーが生成した NMI を要求できません。

欠落している NMI は、以下の 2 つの条件のいずれかによって開始されます。

1. Integrated Lights-Out (iLO) サーバー管理ソフトウェアの **NMI 生成** ボタン。このボタンはユーザーがトリガーします。
2. **hpwdt** ウォッチドッグ。デフォルトでは、有効期限により NMI がサーバーに送信されます。

通常、両方のシーケンスは、システムが応答しない場合に発生します。通常、これらの状況の NMI ハンドラーは **kernel panic()** 関数を呼び出します。また、設定されていれば、**kdump** サービスが **vmcore** ファイルを生成します。

ただし、NMI が見つからないため、**kernel panic()** は呼び出されず、**vmcore** が収集されません。

最初のケース (1.) でシステムが応答しない場合は、その状態のままになります。このシナリオを回避するには、仮想 **電源** ボタンを使用してサーバーをリセットするか、電源を切って入れ直します。

2 つ目のケース (2.) では、欠落している NMI が Automated System Recovery (ASR) からのリセットの後 9 秒後に続きます。

HPE Gen9 Server ラインでは、1 桁台の割合でこの問題が発生します。Gen10 の周波数がさらに小さくなる。

(BZ#1602962)

tuned-adm profile powersave コマンドを使用すると、システムが応答しなくなります。

tuned-adm profile powersave コマンドを実行すると、古い Thunderx (CN88xx) プロセッサを持つ Penguin Valkyrie 2000 2 ソケットシステムが応答しなくなります。これにより、作業を再開するためシステムを再起動することになります。この問題を回避するには、システムが上記の仕様と一致する場合には **powersave** プロファイルの使用を避けてください。

(BZ#1609288)

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 0x30000000-0x31ffffff] not reserved in ACPI namespace
[ 2.827911] acpi PNP0A08:00: ECAM at [mem 0x30000000-0x31ffffff] for [bus 00-1f]
```

ただし、カーネルは依然として **0x30000000-0x31ffffff** 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 0x30000000-0x31ffffff not reserved in ACPI namespace" appears during system boot](#) を参照してください。

(BZ#1868526)

デフォルト設定の **hwloc** コマンドは、単一の CPU Power9 および Power10 LPAR で動作しません。

バージョン 2.2.0 の **hwloc** パッケージでは、Power9 / Power10 CPU を実行する単一ノードの Non-Uniform Memory Access (NUMA) システムは disallowed とみなされます。そのため、**hwloc** コマンドがすべて機能せず、以下のエラーメッセージが表示されます。

```
Topology does not contain any NUMA node, aborting!
```

この問題を回避するには、以下のいずれかのオプションを使用できます。

- 環境変数 **HWLOC_ALLOW=all** を設定する

- さまざまな **hwloc** コマンドで **disallowed** フラグを使用する

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)

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 (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 **openlib** 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 (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_ucx_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)

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)

10.7. 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 4 4 17 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)

10.8. ファイルシステムおよびストレージ

/boot ファイルシステムを LVM に配置することができません。

/boot ファイルシステムを LVM 論理ボリュームに配置することはできません。この制限は、以下の理由により存在します。

- EFI システムでは、**EFI システムパーティション** が従来の **/boot** ファイルシステムとして機能します。uEFI 標準では、特定の GPT パーティションタイプと、このパーティションの特定のファイルシステムタイプが必要です。
- RHEL 8 は、システムブートエントリーに **Boot Loader Specification (BLS)** を使用します。この仕様では、プラットフォームのファームウェアが **/boot** ファイルシステムを読み込める必要があります。EFI システムでは、プラットフォームファームウェアは uEFI 標準で定義された **/boot** 設定のみを読み取ることができます。
- GRUB 2 ブートローダーでの LVM 論理ボリュームに対するサポートは完全ではありません。Red Hat は、uEFI や BLS などの標準があるので、この機能のユースケース数が減少しているため、サポートを改善する予定はありません。

Red Hat では、LVM での **/boot** のサポートを提供する予定はありません。代わりに、Red Hat は、**/boot** ファイルシステムを LVM 論理ボリュームに配置する必要がないシステムスナップショットおよびロールバックを管理するツールを提供します。

(BZ#1496229)

LVM で、複数のブロックサイズを持つボリュームグループが作成できない

vgcreate または **vgextend** などの LVM ユーティリティーでは、物理ボリューム (PV) の論理ブロックサイズが異なるボリュームグループ (VG) を作成できなくなりました。別のブロックサイズの PV で基礎となる論理ボリューム (LV) を拡張するとファイルシステムがマウントに失敗するため、LVM はこの変更を採用しました。

ブロックサイズが混在する VG の作成を再度有効にするには、**lvm.conf** ファイルの **allow_mixed_block_sizes=1** オプションを設定します。

(BZ#1768536)

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、**writocache** 論理ボリュームに名前を付けることはできません。
- **writocache** を指定した論理ボリュームは、シンプルまたは 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 **writocache** 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](#))

NFS 4.0 パッチにより、オープンな高ワークロードでパフォーマンスが低下する可能性があります。

以前、場合によっては NFS のオープン操作で、サーバー上のファイルが削除されたり、名前が変更されたりするという事実を見落とすというバグが修正されています。ただし、この修正により、多くのオープンな操作が必要とするワークロードのパフォーマンスが遅くなる可能性があります。この問題を回避するには、NFS バージョン 4.1 以降を使用します。これは、多くの場合においてクライアントに委譲を付与するように改善されています。このため、クライアントがローカルに素早く安全にオープン操作を実行できます。

([BZ#1748451](#))

複数のクォータタイプが指定されている場合、**xfs_quota state** はすべての猶予時間を出力しない

現在、**xfs_quota state** コマンドは、複数のクォータタイプを指定するオプションで、クォータの猶予期間を予想通りに出力しません。この問題を回避するには、コマンドオプションに必要なクォータタイプを個別に指定します。つまり、**xfs_quota state -g**、**xfs_quota state -p** または **xfs_quota state -u** になります。

([BZ#1949743](#))

10.9. 高可用性およびクラスター

停止操作で **crm_mon** 出力を解析する **ocf: Sets:pgsql** リソースエージェントおよび RHEL 8.4 のシャットダウンプロセス中に停止に失敗する場合があります。

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。その結果、リソースの停止などのシャットダウンの進捗は監視できません。また、`resource-agents` パッケージで配布される `ocf:heartbeat:pgsql` エージェントなどの停止操作で `crm_mon` 出力を解析するリソースエージェントなど、クラスターの停止が失敗する可能性があるため、クラスターの問題が発生する可能性があります。

`z-stream` が利用可能になるまで、`ocf:heartbeat:pgsql` リソースエージェントを使用するクラスターは RHEL 8.4 にアップグレードされないことが推奨されます。

(BZ#1948620)

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

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_DEEPCBIND` 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_DEEPCBIND` 動的リンカーオプションを使用できるようになります。`APR_DEEPCBIND` 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)

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、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。

MariaDB 10.5 は、PAM プラグインバージョン 2.0 を提供します。 も併せて参照してください。

(BZ#1942330)

pyodbc が MariaDB 10.3 で動作しない

現在、**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。**MariaDB 10.3** サーバーおよび **MariaDB 10.5** サーバーの以前のバージョンは、この問題の影響を受けません。

根本的な原因は **mariadb-connector-odbc** パッケージにあり、影響を受けるパッケージバージョンは以下のとおりです。

- **pyodbc-4.0.30**
- **mariadb-server-10.3.27**
- **mariadb-connector-odbc-3.0.7**

(BZ#1944692)

10.11. コンパイラーおよび開発ツール

GCC Toolset 10: Valgrind は IBM z15 アーキテクチャーのサポートを誤って報告していました

Valgrind は、特定の IBM z15 プロセッサ機能をサポートしていませんが、GCC Toolset 10 Valgrind のバグでは、z15 対応システム上で実行する際に z15 サポートを報告することになります。これにより、Valgrind で利用可能な場合に z15 機能の使用を試みるソフトウェアです。この問題を回避するには、z15 プロセッサで実行している場合に、`/usr/bin/valgrind` 経由でアクセス可能な Valgrind のシステムバージョンを使用します。このビルドは z15 サポートを報告しません。

(BZ#1937340)

PCP の pmproxy におけるメモリーリーク

pmproxy サービスでは、5.3.0 より前のバージョンの Performance Co-Pilot (PCP) でメモリーリークが発生します。PCP バージョン 5.3.0 は、RHEL 8.4 および RHEL 8 の以前のマイナーバージョンでは使用できません。これにより、RHEL 8 ユーザーのメモリー使用量は予想よりも多くなる可能性があります。

この問題を回避するには、**pmproxy** のメモリー使用量を制限します。

1. 以下のコマンドを実行して `/etc/systemd/system/pmproxy.service.d/override.conf` ファイルを作成します。

```
# systemctl edit pmproxy
```

2. 以下の内容を `override.conf` に追加して変更を保存します。

```
[Service]
MemoryMax=10G
```

要件に従って 10G の値を置き換えます。

3. **pmproxy** サービスを再起動します。

```
# systemctl restart pmproxy
```

これにより、**pmproxy** のメモリー使用量が指定の制限に達すると、**pmproxy** サービスが再起動します。

(BZ#1991659)

10.12. IDENTITY MANAGEMENT

すべての KRA メンバーが非表示レプリカの場合は、KRA のインストールに失敗します。

最初の KRA インスタンスが非表示レプリカにインストールされている場合、Key Recovery Authority (KRA) がすでに存在するクラスターでは **ipa-kra-install** ユーティリティーで問題が発生します。そのため、これ以上、追加の KRA インスタンスをクラスターに追加することはできません。

この問題を回避するには、新しい KRA インスタンスを追加する前に、KRA ロールが割り当てられた非表示レプリカを解除します。**ipa-kra-install** が正常に終了してから、レプリカを再度非表示にできません。

(BZ#1816784)

--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)

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、12800000008 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パースで、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パ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、それら进行处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パ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、スパースファイル进行处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** などのスパ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)

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)

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)

バージョン 1.2.2 へのリベース後の authselect のダウングレードにより、システム認証の破損

authselect パッケージが、最新のアップストリームバージョン **1.2.2** にリベースされました。**authselect** のダウングレードはサポートされておらず、**root** を含むすべてのユーザーに対してシステム認証が破損しています。

authselect パッケージを **1.2.1** 以前にダウングレードした場合は、この問題を回避するために以下の手順を実行します。

1. GRUB ブート画面で、起動するカーネルのバージョンを含む **Red Hat Enterprise Linux** を選択し、**e** を押してエントリーを編集します。
2. **linux** で始まる行の末尾で、**single** を、別の単語で入力し、**Ctrl+x** を押して起動プロセスを開始します。
3. シングルユーザーモードでの起動時に、**root** パスワードを入力します。
4. 以下のコマンドを使用して **authselect** 設定を復元します。

```
# authselect select sssd --force
```

(BZ#1892761)

pki-ca パッケージのバージョンが 10.10.5 よりも古い場合、IdM サーバーの RHEL 8.3 から RHEL 8.4 へのアップグレードに失敗する

pki-ca パッケージのバージョンが 10.10.5 よりも古い場合、IdM サーバーのアップグレードプログラム **ipa-server-upgrade** は失敗します。必要なファイルがこれらのバージョンに存在しないため、IdM サーバーのアップグレードは、パッケージのインストール時と **ipa-server-upgrade** または **ipactl** の実行時の両方において、正常に完了しません。

この問題を解決するには、**pki-*** パッケージをバージョン 10.10.5 以降にアップグレードし、**ipa-server-upgrade** コマンドを再度実行します。

(BZ#1957768)

ldap_id_use_start_tls オプションのデフォルト値を使用する場合の潜在的なリスク

ID ルックアップに TLS を使用せずに `ldap://` を使用すると、攻撃ベクトルのリスクが生じる可能性があります。特に、中間者 (MITM) 攻撃は、攻撃者が、たとえば、LDAP 検索で返されたオブジェクトの UID または GID を変更することによってユーザーになりすますことを可能にする可能性があります。

現在、TLS を強制する SSSD 設定オプション `ldap_id_use_start_tls` は、デフォルトで `false` に設定されています。セットアップが信頼できる環境で動作していることを確認し、`id_provider = ldap` に暗号化されていない通信を使用しても安全かどうかを判断してください。注記: `id_provider = ad` および `id_provider = ipa` は、SASL および GSSAPI によって保護された暗号化接続を使用するため、影響を受けません。

暗号化されていない通信を使用することが安全ではない場合は、`/etc/sss/sss.conf` ファイルで `ldap_id_use_start_tls` オプションを `true` に設定して TLS を強制します。デフォルトの動作は、RHEL の将来のリリースで変更される予定です。

(JIRA:RHELPLAN-155168)

10.13. 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 ユーティリティーの 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)

10.14. グラフィックインフラストラクチャー

`radeon` がハードウェアを適切なハードウェアリセットに失敗します。

現在、`radeon` カーネルドライバーは、`kexec` コンテキストでハードウェアを正しくリセットしません。代わりに `radeon` がフェイルオーバーします。これにより、`kdump` サービスの残りの部分が失敗します。

この問題を回避するには、`/etc/kdump.conf` ファイルに以下の行を追加して、`kdump` で `radeon` を無効にします。

```
dracut_args --omit-drivers "radeon"  
force_rebuild 1
```

マシンと `kdump` を再起動します。`kdump` の起動後、設定ファイルから `force_rebuild 1` 行が削除される可能性があります。

このシナリオでは、`kdump` 中にグラフィックは利用できませんが、`kdump` は正常に動作します。

(BZ#1694705)

1つの MST トポロジーで複数の HDR ディスプレイを使用すると、電源が入らないことがあります。

nouveau ドライバーの NVIDIA Turing GPUs を使用するシステムで、**DisplayPort** ハブ (ラップトップのドックなど) を使用して HDR プラグインのサポートがあるモニターを複数接続すると、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)

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時、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)

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)

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)

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に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)

10.15. 仮想化

virsh iface-* コマンドが一貫して動作しません。

現在、**virsh iface-*** コマンド (**virsh iface-start**、**virsh iface-destroy** など) は、設定の依存関係が原因で頻繁に失敗します。したがって、ホストネットワーク接続の設定および管理には **virsh iface-*** コマンドを使用しないことが推奨されます。代わりに、NetworkManager プログラムとその関連管理アプリケーションを使用します。

(BZ#1664592)

多数の virtio-blk ディスクを使用すると、仮想マシンが起動しないことがあります。

多数の virtio-blk デバイスを仮想マシンに追加すると、プラットフォームで利用可能な割り込みベクトルの数が使い切られる可能性があります。これが発生すると、仮想マシンのゲスト OS は起動できず、**dracut-initqueue[392]: Warning: Could not boot** エラーが表示されます。

(BZ#1719687)

virtio-blk を使用して仮想マシンに LUN デバイスを割り当てると機能しません。

q35 マシンタイプは、移行用の virtio 1.0 デバイスをサポートしないため、RHEL 8 では virtio 1.0 で非推奨となった機能はサポートされません。特に、RHEL 8 ホストで virtio-blk デバイスから SCSI コマンドを送信することはできません。したがって、virtio-blk コントローラーを使用する場合は、物理ディスクを LUN デバイスとして仮想マシンに割り当てると失敗します。

物理ディスクをゲストオペレーティングシステムを通して渡すことは引き続き可能ですが、**device='lun'** オプションではなく、**device='disk'** オプションで設定する必要があることに留意してください。

(BZ#1777138)

ホストで TSX が無効になっていると、Cooperlake を使用する仮想マシンを起動できません。

現在、ホストで **TSX** CPU フラグが無効化されている場合、**Cooperlake** CPU モデルを使用する仮想マシンは起動に失敗します。代わりに、ホストに以下のエラーメッセージが表示されます。

```
the CPU is incompatible with host CPU: Host CPU does not provide required features: hle, rtm
```

このようなホストで仮想マシンが **Cooperlake** を使用できるようにするには、VM の XML 設定の VM 設定で HLE、RTM、および TAA_NO のフラグを無効化します。

```
<feature policy='disable' name='hle'/>
<feature policy='disable' name='rtm'/>
<feature policy='disable' name='taa-no'/>
```


[\(BZ#1860743\)](#)

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エントで 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。

[\(BZ#1924016\)](#)

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。

[\(BZ#1554735\)](#)

`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。

[\(BZ#1910848\)](#)

AMD EPYC でホストパススルーモードを使用する際に、SMT CPU トポロジーが仮想マシンで検出されません。

AMD EPYC ホストで行われた CPU ホストパススルーモードで仮想マシンを起動すると、**TOPOEXT** 機能フラグは存在しません。したがって、仮想マシンは、コアごとに複数のスレッドを持つ仮想 CPU トポロジーを検出できません。この問題を回避するには、ホストパススルーの代わりに EPYC CPU モデ

ルを使用して仮想マシンを起動します。

(BZ#1740002)

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)

仮想マシンから `macvtap` インターフェイスを削除すると、`macvtap` 接続がすべてリセットされます。

現在、複数の `macvtap` デバイスを持つ実行中の仮想マシン (VM) から `macvtap` インターフェイスを削除すると、他の `macvtap` インターフェイスの接続設定もリセットされます。結果として、仮想マシンでネットワークの問題が発生する可能性があります。

(BZ#1332758)

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)

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 (以降) カーネルを使用しているにもかかわらず、RHEL 8.5 (以降) **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)

10.16. クラウド環境の RHEL

Azure および Hyper-V で kdump が起動しないことがあります。

Microsoft Azure または Hyper-V ハイパーバイザー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)

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)

特定の NIC を搭載した RHEL 8 仮想マシンの Azure のリモートマシンへのコアダンプには、予想よりも長い時間がかかっていました。

現在、**kdump** ユーティリティーを使用した、Microsoft Azure ハイパーバイザー上の 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。これにより、ダンプファイルは即座にではなく、約 200 秒後に保存されます。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す。

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

(BZ#1854037)

nm-cloud-setup ユーティリティーは、Microsoft Azure に誤ったデフォルトルートを設定しません。

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。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、接続が破損していました。現在利用できる回避策はありません。

(BZ#1912236)

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されることがあります。

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-0x60022480{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=xfst --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)

RHEL 8 仮想マシンの場合、AWS ARM64 インスタンスでネットワークパフォーマンスが低下する。

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るとき、または、**iommu.strict** カーネルパラメーターが定義されないとき、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す。

この問題を回避するには、パラメーターを **iommu.strict=0** に変更します。ただし、これにより、仮想マシンのセキュリティも減らすこともできます。

(BZ#1836058)

FIPS モードが有効な場合に RHEL 8 ゲストが休止状態に

現在、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する仮想マシンをハイバネートすることはできません。

(BZ#1934033, BZ#1944636)

バックアップ AMI から作成された EC2 インスタンスで SSH キーが正しく生成されない

現在、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 キーを削除しますが、新しい SSH キーは作成しません。その結果、仮想マシンがホストに接続できない場合があります。

この問題を回避するには、**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)

バックアップ AMI から作成された EC2 インスタンスで SSH キーが正しく生成されない

現在、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 キーを削除しますが、新しい SSH キーは作成しません。その結果、仮想マシンがホストに接続できない場合があります。

この問題を回避するには、**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#1963981](#))

10.17. サポート性

redhat-support-tool が **FUTURE** 暗号化ポリシーを使用すると機能しません。

カスタマーポータル API の証明書が使用する暗号化キーは **FUTURE** のシステム全体の暗号化ポリシーが定義する要件を満たさないので、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、カスタマーポータル API への接続中に **DEFAULT** 暗号化ポリシーを使用します。

([BZ#1802026](#))

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#1859301 , BZ#1862529 , BZ#1859218 , BZ#1850275 , BZ#1851975
KVM Hypervisor	JIRA:RHELPLAN-44450
NetworkManager	BZ#1900260 , BZ#1878783 , BZ#1766944 , BZ#1912236
OpenIPMI	BZ#1796588
SLOF	BZ#1910848
accel-config	BZ#1843266
anaconda	BZ#1890009 , BZ#1874394 , BZ#1642391 , BZ#1609325 , BZ#1854307 , BZ#1821192 , BZ#1822880 , BZ#1914955 , BZ#1847681 , BZ#1903786 , BZ#1931069 , BZ#1954408 , BZ#1897657
apr	BZ#1819607
authselect	BZ#1892761
bcc	BZ#1879411
bind	BZ#1876492 , BZ#1882040 , BZ#1854148
bpfttrace	BZ#1879413
clevis	BZ#1887836 , BZ#1853651
cloud-init	BZ#1886430 , BZ#1750862 , BZ#1957532 , BZ#1963981
cmake	BZ#1816874
cockpit	BZ#1666722
corosync-qdevice	BZ#1784200
corosync	BZ#1870449
createrepo_c	BZ#1795936 , BZ#1894361

コンポーネント	チケット
crun	BZ#1841438
crypto-policies	BZ#1919155 , BZ#1660839
dhcp	BZ#1883999
distribution	BZ#1877430 , BZ#1855776 , BZ#1855781 , BZ#1657927
dnf	BZ#1865803 , BZ#1807446 , BZ#1698145
dwarves	BZ#1903566
dyninst	BZ#1892001 , BZ#1892007
edk2	BZ#1935497
elfutils	BZ#1875318 , BZ#1879758
fapolicyd	BZ#1940289 , BZ#1896875 , BZ#1887451
fence-agents	BZ#1775847
freeipmi	BZ#1861627
freeradius	BZ#1723362
gcc	BZ#1868446 , BZ#1821994 , BZ#1850498 , BZ#1656139 , BZ#1891998
gdb	BZ#1853140
ghostscript	BZ#1874523
glibc	BZ#1868106 , BZ#1871397 , BZ#1880670 , BZ#1882466 , BZ#1871396 , BZ#1893662 , BZ#1817513 , BZ#1871385 , BZ#1871387 , BZ#1871395
gnome-shell-extensions	BZ#1717947
gnome-software	BZ#1668760
gnutls	BZ#1628553
go-toolset	BZ#1870531
grafana-container	BZ#1916154

コンポーネント	チケット
grafana-pcp	BZ#1845592 , BZ#1854093
grafana	BZ#1850471
grub2	BZ#1583445
httpd	BZ#1869576 , BZ#1883648
hwloc	BZ#1841354 , BZ#1917560
ima-evm-utils	BZ#1868683
ipa	BZ#1891056 , BZ#1340463 , BZ#1816784 , BZ#1924707 , BZ#1664719 , BZ#1664718
iproute	BZ#1849815
iptraf-ng	BZ#1842690 , BZ#1906097
jmc	BZ#1919283
kernel-rt	BZ#1858099
kernel	BZ#1806882 , BZ#1846838 , BZ#1884857 , BZ#1876527 , BZ#1660290 , BZ#1885850 , BZ#1649647 , BZ#1838876 , BZ#1871246 , BZ#1893882 , BZ#1876519 , BZ#1860031 , BZ#1844416 , BZ#1780258 , BZ#1851933 , BZ#1885406 , BZ#1867490 , BZ#1908893 , BZ#1919745 , BZ#1867910 , BZ#1887940 , BZ#1874005 , BZ#1871214 , BZ#1622041 , BZ#1533270 , BZ#1900674 , BZ#1869758 , BZ#1861261 , BZ#1848427 , BZ#1847567 , BZ#1844157 , BZ#1844111 , BZ#1811839 , BZ#1877019 , BZ#1548297 , BZ#1844086 , BZ#1839055 , BZ#1905088 , BZ#1882620 , BZ#1784246 , BZ#1916583 , BZ#1924230 , BZ#1793389 , BZ#1944639 , BZ#1694705 , BZ#1748451 , BZ#1654962 , BZ#1708456 , BZ#1812577 , BZ#1666538 , BZ#1602962 , BZ#1609288 , BZ#1730502 , BZ#1865745 , BZ#1868526 , BZ#1910358 , BZ#1924016 , BZ#1906870 , BZ#1940674 , BZ#1930576 , BZ#1907271 , BZ#1942888 , BZ#1836058 , BZ#1934033 , BZ#1519039 , BZ#1627455 , BZ#1501618 , BZ#1495358 , BZ#1633143 , BZ#1570255 , BZ#1814836 , BZ#1696451 , BZ#1348508 , BZ#1839311 , BZ#1783396 , JIRA:RHELPLAN-57712 , BZ#1837187 , BZ#1904496 , BZ#1660337 , BZ#1665295 , BZ#1569610
kexec-tools	BZ#1844941 , BZ#1931266 , BZ#1854037
kmod-redhat-oracleasm	BZ#1827015
kpatch	BZ#1798711

コンポーネント	チケット
krb5	BZ#1877991
libbpf	BZ#1919345
libgnome-keyring	BZ#1607766
libguestfs	BZ#1554735
libmpc	BZ#1835193
libpcap	BZ#1743650
libpwquality	BZ#1537240
libreswan	BZ#1891128 , BZ#1372050 , BZ#1025061 , BZ#1934058 , BZ#1934859
libselinux-python-2.8-module	BZ#1666328
libselinux	BZ#1879368
libsemanage	BZ#1913224
libvirt	BZ#1664592 , BZ#1332758 , BZ#1528684
libvpd	BZ#1844429
llvm-toolset	BZ#1892716
lvm2	BZ#1496229 , BZ#1768536
mariadb-connector-odbc	BZ#1944692
mariadb	BZ#1936842 , BZ#1944653 , BZ#1942330
mesa	BZ#1886147
micropipenv	BZ#1849096
mod_fcgid	BZ#1876525
mod_security	BZ#1824859
mutter	BZ#1886034

コンポーネント	チケット
mysql-selinux	BZ#1895021
net-snmp	BZ#1817190
nfs-utils	BZ#1592011
nispor	BZ#1848817
nmstate	BZ#1674456
nss_nis	BZ#1803161
nss	BZ#1817533 , BZ#1645153
opal-prd	BZ#1844427
opencryptoki	BZ#1847433
opencv	BZ#1886310
openmpi	BZ#1866402
opensc	BZ#1877973 , BZ#1947025
openscap	BZ#1824152 , BZ#1887794 , BZ#1840579
openssl	BZ#1810911
osbuild-composer	BZ#1951964
oscap-anaconda-addon	BZ#1843932 , BZ#1665082 , BZ#1674001 , BZ#1691305 , BZ#1834716
p11-kit	BZ#1887853
pacemaker	BZ#1371576 , BZ#1948620
pcp-container	BZ#1916155
pcp	BZ#1854035 , BZ#1847808
pcs	BZ#1869399 , BZ#1741056 , BZ#1667066 , BZ#1667061 , BZ#1457314 , BZ#1839637 , BZ#1619620 , BZ#1851335
perl-IO-String	BZ#1890998

コンポーネント	チケット
perl-Time-HiRes	BZ#1895852
pki-core	BZ#1868233 , BZ#1729215
podman	BZ#1734854 , BZ#1881894 , BZ#1932083
policycoreutils	BZ#1868717 , BZ#1926386
popt	BZ#1843787
postfix	BZ#1688389 , BZ#1711885
powerpc-utils	BZ#1853297
py3c	BZ#1841060
pyOpenSSL	BZ#1629914
pykickstart	BZ#1637872
pyodbc	BZ#1881490
python-PyMySQL	BZ#1820628
python-blivet	BZ#1656485
qemu-kvm	BZ#1790620 , BZ#1719687 , BZ#1860743 , BZ#1740002 , BZ#1651994
quota	BZ#1868671
rear	BZ#1729499 , BZ#1898080 , BZ#1832394
redhat-support-tool	BZ#1802026
redis	BZ#1862063
resource-agents	BZ#1471182
rhel-system-roles	BZ#1865990 , BZ#1926947 , BZ#1889484 , BZ#1927943 , BZ#1893712 , BZ#1893743 , BZ#1893906 , BZ#1893908 , BZ#1895188 , BZ#1893696 , BZ#1893699 , BZ#1889893 , BZ#1893961
rpm	BZ#1834931 , BZ#1923167 , BZ#1688849

コンポーネント	チケット
rshim	BZ#1744737
rsyslog	BZ#1869874 , JIRA:RHELPLAN-10431 , BZ#1679512
rust-toolset	BZ#1896712
samba	BZ#1944657 , BZ#2009213 , JIRA:RHELPLAN-13195 , Jira:RHELDOCS-16612
scap-security-guide	BZ#1889344 , BZ#1927019 , BZ#1918742 , BZ#1778188 , BZ#1843913 , BZ#1858866 , BZ#1750755
scap-workbench	BZ#1877522
selinux-policy	BZ#1889673 , BZ#1860443 , BZ#1931848 , BZ#1461914
sendmail	BZ#1868041
setroubleshoot	BZ#1875290 , BZ#1794807
skopeco	BZ#1940854
sos	BZ#1966838
spamassassin	BZ#1822388
spice	BZ#1849563
sssd	BZ#1819012 , BZ#1884196 , BZ#1884213 , BZ#1784459 , BZ#1893698 , BZ#1881992
stalld	BZ#1875037
stratisd	BZ#1798244 , BZ#1868100
subscription-manager	BZ#1905398
subversion	BZ#1844947
sudo	BZ#1786990
swig	BZ#1853639
systemd	BZ#1827462

コンポーネント	チケット
systemtap	BZ#1875341
tang-container	BZ#1913310
tang	BZ#1828558
texlive	BZ#1889802
tpm2-abrmd	BZ#1855177
tuned	BZ#1874052
udica	BZ#1763210
unbound	BZ#1850460
usbguard	BZ#1887448 , BZ#1940060
valgrind	BZ#1504123 , BZ#1937340
virtio-win	BZ#1861229
wayland	BZ#1673073
xdp-tools	BZ#1880268
xfspgrog	BZ#1949743
xorg-x11-drv-qxl	BZ#1642887
xorg-x11-server	BZ#1698565

コンポーネント	チケット
その他	BZ#1839151 , BZ#1780124 , JIRA:RHELPLAN-59941, JIRA:RHELPLAN-59938, JIRA:RHELPLAN-59950, BZ#1952421 , JIRA:RHELPLAN-37817, BZ#1918055 , JIRA:RHELPLAN-56664, JIRA:RHELPLAN-56661, JIRA:RHELPLAN-39843, BZ#1925192 , JIRA:RHELPLAN-73418, JIRA:RHELPLAN-63081, BZ#1935686 , BZ#1634655 , JIRA:RHELPLAN-56782, JIRA:RHELPLAN-72660, JIRA:RHELPLAN-72994, JIRA:RHELPLAN-37579, BZ#1952161 , BZ#1640697 , BZ#1659609 , BZ#1687900 , BZ#1697896 , JIRA:RHELPLAN-59111, BZ#1757877 , BZ#1777138 , JIRA:RHELPLAN-27987, JIRA:RHELPLAN-28940, JIRA:RHELPLAN-34199, JIRA:RHELPLAN-57914, BZ#1897383 , BZ#1741436 , BZ#1971061 , JIRA:RHELPLAN-58629, BZ#1960412 , BZ#1959020 , BZ#1690207 , JIRA:RHELPLAN-1212, BZ#1559616 , BZ#1889737 , BZ#1812552 , JIRA:RHELPLAN-14047, BZ#1769727 , JIRA:RHELPLAN-27394, JIRA:RHELPLAN-27737, JIRA:RHELPLAN-56659, BZ#1906489 , BZ#1957316 , BZ#1960043 , BZ#1642765 , JIRA:RHELPLAN-10304, BZ#1646541 , BZ#1647725 , BZ#1932222 , BZ#1686057 , BZ#1748980 , JIRA:RHELPLAN-71200, BZ#1827628 , JIRA:RHELPLAN-45858, BZ#1871025 , BZ#1871953 , BZ#1874892 , BZ#1893767 , BZ#1916296 , BZ#1926114 , BZ#1904251 , JIRA:RHELPLAN-59825, BZ#1920624 , JIRA:RHELPLAN-70700, BZ#1929173

付録B 改訂履歴

0.3-6

2024年5月9日(木) Brian Angelica (bangelic@redhat.com)

- [JIRAの機能強化が更新されました:RHELDOCS-18188](#) (ネットワーク)。

0.3-5

2024年5月9日(木) Brian Angelica (bangelic@redhat.com)

- [BZ#1690207](#) でテクニカルプレビューを更新。

0.3-4

2024年5月9日(木)、 Gabriela Fialova (gfialova@redhat.com)

- 既知の問題 [BZ#1730502](#) (ストレージ) を更新。

0.3-3

2024年2月7日(水)、 Lucie Vařáková (lvarakova@redhat.com)

- 非推奨の機能 [JIRA:RHELDOCS-17641](#) (Networking) を追加。

0.3-2

2023年11月10日金曜日、 Gabriela Fialová (gfialova@redhat.com)

- RHEL ドキュメントへのフィードバックの提供に関するモジュールを更新しました。

0.3-1

2022年11月15日(火) Gabriela Fialová (gfialova@redhat.com)

- 壊れたリンクを修正しました。

0.3-0

2023年10月13日(金) Gabriela Fialová (gfialova@redhat.com)

- テクノロジープレビュー [JIRA:RHELDOCS-16861](#) (コンテナ) を追加しました。

0.2-9

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:RHELDOCS-16612](#) (Samba) を追加しました。

0.2-8

2022年8月5日金曜日、 Lucie Vařáková (lvarakova@redhat.com)

- 既知の問題 [BZ#2028361](#) (インストーラーおよびイメージの作成) を追加。

0.2-7

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

0.2-6

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。

0.2-5

2022 年 12 月 8 日 (木) Marc Muehlfeld (mmuehlfeld@redhat.com)

- 既知の問題 [BZ#2132754](#) (ネットワーキング) を追加しました。

0.2-4

6 月 9 日 (木)、Lucie Vařáková (lmanasko@redhat.com)

- 新機能 [BZ#1996076](#) (アイデンティティ管理) を追加。

0.2-3

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。

0.2-2

2022 年 3 月 24 日 (木) Jaroslav Klech (jklech@redhat.com)

- バグ修正 [BZ#1947839](#) (カーネル) を追加しました。

0.2-1

2022 年 3 月 21 日 (月) Jaroslav Klech (jklech@redhat.com)

- 既知の問題 (カーネル) を削除しました。

0.2-0

2022 年 2 月 4 日 (金) Jaroslav Klech (jklech@redhat.com)

- 非推奨の機能 [BZ#1871863](#) (ハードウェアの有効化) を追加しました。
- [非推奨のパッケージ](#) を更新しました。
- 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。

0.1-9

2022 年 1 月 20 日 (木) Lucie Maňásková (lmanasko@redhat.com)

- 既知の問題 [BZ#2028361](#) (インストーラーおよびイメージの作成) を追加。

0.1-8

2021 年 12 月 23 日 (木) Lenka Špačková (lspackova@redhat.com)

- Soft-RoCE ドライバー `rdma_rxe` に関する情報が、テクノロジープレビューの [BZ#1605216](#) および非推奨の機能 [BZ#1878207](#) (カーネル) に追加されました。

0.1-7

2021年12月22日(水) Lenka Špačková (lspackova@redhat.com)

- [BZ#2005431](#) (セキュリティー) の拡張機能を追加。
- [非推奨のパッケージ](#) を更新しました。

0.1-6

2021年10月29日(木) Jaroslav Klech (jklech@redhat.com)

- `fw_devlink` パラメーターを更新しました (外部カーネルパラメーターに対する重要な変更)。

0.1-5

2021年10月7日(木) Lenka Špačková (lspackova@redhat.com)

- 既知の問題 [BZ#1942330](#) を更新 (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)。

0.1-4

2021年10月5日(火) Lucie Maňásková (lmanasko@redhat.com)

- [非推奨の機能 BZ#1999620](#) (シェルおよびコマンドラインツール) が追加されました。

0.1-3

2021年9月17日(金) Lucie Maňásková (lmanasko@redhat.com)

- 既知の問題 [BZ#1987087](#) (インストーラー) を追加。

0.1-2

2021年9月7日(火) Lucie Maňásková (lmanasko@redhat.com)

- 既知の問題 [BZ#1961722](#) (仮想化) を更新しました。

0.1-1

2021年9月3日(金) Lenka Špačková (lspackova@redhat.com)

- 既知の問題 [BZ#1995558](#) (仮想化) を更新しました。

0.1-0

2021年8月30日(月) Lenka Špačková (lspackova@redhat.com)

- 既知の問題 [BZ#1995558](#) (仮想化) を追加しました。
- バグ修正 [BZ#1940854](#) (コンテナ) を追加しました。

0.0-9

2021年8月20日(金) Lucie Maňásková (lmanasko@redhat.com)

- [YUM/DNF を使用したパッケージ管理](#) を、ディストリビューションの章に追加。
- [BZ#1708456](#) (カーネル) のテキストを更新。

- 新しい機能 [BZ#1888214](#) (ファイルシステムおよびストレージ) を追加。
- 既知の問題 [BZ#1991659](#) (コンパイラーおよび開発ツール) を追加。
- テクノロジープレビュー機能 [JIRA:RHELPLAN-58596](#) (アイデンティティ管理) を追加。

0.0-8

2021年8月10日(火) Lucie Maňásková (Imanasko@redhat.com)

- 新機能 [BZ#1905398](#) (クラウド環境の RHEL) を更新。

0.0-7

2021年8月3日(火) Lucie Maňásková (Imanasko@redhat.com)

- 既知の問題 [BZ#1935722](#) (インストーラーおよびイメージの作成) を追加。
- 既知の問題 [BZ#1961722](#) (仮想化) を追加。

0.0-6

2021年7月23日(金) Lucie Maňásková (Imanasko@redhat.com)

- 既知の問題 [BZ#1924230](#) (セキュリティ) を追加。
- 既知の問題 [BZ#1957768](#) (アイデンティティ管理) を追加。

0.0-5

2021年7月16日(金) Lucie Maňásková (Imanasko@redhat.com)

- 既知の問題 [BZ#1959020](#) (仮想化) を追加。
- 既知の問題 [BZ#1963981](#) (クラウド環境の RHEL) を追加。
- 新機能 [BZ#1340463](#) (アイデンティティ管理) を追加。
- 無効なリリースノートおよびその改訂履歴エントリを削除。

0.0-4

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

- 新機能 [BZ#1966838](#) (サポート機能) を追加。
- **sfc** で非推奨デバイスを更新。
- その他の小さな改善点。

0.0-3

2021年6月16日(水) Lucie Maňásková (Imanasko@redhat.com)

- 非推奨機能 [BZ#1929173](#) (ネットワーク) を追加。
- 非推奨機能 [BZ#1920624](#) (コンパイラーと開発ツール) を追加。
- 新機能 [JIRA:RHELPLAN-63081](#) (アイデンティティ管理) を追加。

- 既知の問題 [BZ#1949743](#) (ファイルシステムおよびストレージ) を追加。
- 既知の問題 [BZ#1332758](#) (仮想化) を追加。
- 既知の問題 [BZ#1957532](#) (クラウド環境の RHEL) を追加。
- その他の小さな改善点。

0.0-2

2021年6月4日(金) Lenka Špačková (lspackova@redhat.com)

- [BZ#1849815](#) の注記を修正。
- さまざまなフォーマットの改善。

0.0-1

2021年5月18日(水) Lucie Maňásková (لمانasko@redhat.com)

- Red Hat Enterprise Linux 8.4 リリースノートも併せて参照してください。

0.0-0

2021年3月31日(水) Lucie Maňásková (لمانasko@redhat.com)

- Red Hat Enterprise Linux 8.4 Beta リリースノートをリリースしました。