



Red Hat Enterprise Linux 9

9.4 リリースノート

Red Hat Enterprise Linux 9.4 リリースノート

Red Hat Enterprise Linux 9 9.4 リリースノート

Red Hat Enterprise Linux 9.4 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

このリリースノートでは、Red Hat Enterprise Linux 9.4 での改良点および実装された追加機能の概要、この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、非推奨機能などの詳細も説明します。Red Hat Enterprise Linux のインストールは、Installation を参照してください。

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 9.4 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	9
1.3. RED HAT CUSTOMER PORTAL LABS	10
1.4. 関連情報	11
第2章 アーキテクチャー	12
第3章 RHEL 9 のコンテンツの配布	13
3.1. インストール	13
3.2. リポジトリ	13
3.3. APPLICATION STREAMS (APPSTREAM)	14
3.4. YUM/DNF を使用したパッケージ管理	14
第4章 新機能	15
4.1. インストーラーおよびイメージの作成	15
4.2. セキュリティー	15
4.3. RHEL FOR EDGE	22
4.4. シェルおよびコマンドラインツール	22
4.5. インフラストラクチャーサービス	23
4.6. ネットワーク	24
4.7. カーネル	30
4.8. ブートローダー	32
4.9. ファイルシステムおよびストレージ	32
4.10. 高可用性およびクラスター	35
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	35
4.12. コンパイラーおよび開発ツール	41
4.13. IDENTITY MANAGEMENT	46
4.14. WEB コンソール	52
4.15. RED HAT ENTERPRISE LINUX システムロール	52
4.16. 仮想化	57
4.17. クラウド環境の RHEL	59
4.18. コンテナ	60
第5章 外部カーネルパラメーターへの重要な変更	63
新しいカーネルパラメーター	63
更新されたカーネルパラメーター	65
削除されたカーネルパラメーター	67
新しい sysctl パラメーター	67
sysctl パラメーターの更新	67
第6章 デバイスドライバー	69
6.1. 新しいドライバー	69
6.2. 更新されたドライバー	74
第7章 利用可能な BPF 機能	75
第8章 バグ修正	94
8.1. インストーラーおよびイメージの作成	94
8.2. セキュリティー	95

8.3. サブスクリプションの管理	97
8.4. ソフトウェア管理	98
8.5. シェルおよびコマンドラインツール	98
8.6. ネットワーク	101
8.7. カーネル	102
8.8. ファイルシステムおよびストレージ	103
8.9. 高可用性およびクラスター	105
8.10. 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	106
8.11. コンパイラーおよび開発ツール	106
8.12. IDENTITY MANAGEMENT	108
8.13. WEB コンソール	111
8.14. RED HAT ENTERPRISE LINUX システムロール	111
8.15. 仮想化	113
第9章 テクノロジーレビュー	116
9.1. インストーラーおよびイメージの作成	116
9.2. セキュリティー	118
9.3. RHEL FOR EDGE	118
9.4. シェルおよびコマンドラインツール	119
9.5. インフラストラクチャーサービス	119
9.6. ネットワーク	119
9.7. カーネル	121
9.8. ファイルシステムおよびストレージ	122
9.9. コンパイラーおよび開発ツール	123
9.10. IDENTITY MANAGEMENT	124
9.11. デスクトップ	125
9.12. WEB コンソール	126
9.13. 仮想化	126
9.14. クラウド環境の RHEL	128
9.15. コンテナ	128
第10章 非推奨の機能	130
10.1. インストーラーおよびイメージの作成	130
10.2. セキュリティー	132
10.3. サブスクリプションの管理	134
10.4. シェルおよびコマンドラインツール	134
10.5. インフラストラクチャーサービス	135
10.6. ネットワーク	136
10.7. カーネル	137
10.8. ファイルシステムおよびストレージ	138
10.9. 動的プログラミング言語、WEB サーバー、およびデータベースサーバー	139
10.10. コンパイラーおよび開発ツール	140
10.11. IDENTITY MANAGEMENT	140
10.12. デスクトップ	142
10.13. グラフィックインフラストラクチャー	142
10.14. RED HAT ENTERPRISE LINUX システムロール	143
10.15. 仮想化	143
10.16. コンテナ	145
10.17. 非推奨のパッケージ	147
第11章 既知の問題	162
11.1. インストーラーおよびイメージの作成	162
11.2. セキュリティー	167
11.3. RHEL FOR EDGE	172

11.4. ソフトウェア管理	172
11.5. シェルおよびコマンドラインツール	173
11.6. インフラストラクチャーサービス	175
11.7. ネットワーク	176
11.8. カーネル	176
11.9. ファイルシステムおよびストレージ	181
11.10. 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	182
11.11. IDENTITY MANAGEMENT	183
11.12. デスクトップ	187
11.13. グラフィックインフラストラクチャー	188
11.14. WEB コンソール	189
11.15. RED HAT ENTERPRISE LINUX システムロール	189
11.16. 仮想化	190
11.17. クラウド環境の RHEL	196
11.18. サポート性	198
11.19. コンテナ	198
付録A コンポーネント別のチケットリスト	200
付録B 改訂履歴	210

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、用語の置き換えは、今後の複数のリリースにわたって段階的に実施されます。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 ドキュメントに関するご意見やご感想をお寄せください。また、改善点があればお知らせください。

Jira からのフィードバック送信 (アカウントが必要)

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

第1章 概要

1.1. RHEL 9.4 における主な変更点

インストーラーおよびイメージの作成

RHEL Image Builderの主なハイライト:

- RHEL 9.4 以降のリリースディストリビューションでは、オペレーティングシステム用に予約されている特定のパスを除き、任意のカスタムマウントポイントを指定できます。
- **auto-lvm**、**lvm**、**raw** などのさまざまなパーティション設定モードを作成できます。
- ルールを追加および削除するための **selected** オプションと **unselected** オプションを使用して、プロファイルの調整オプションをカスタマイズし、ブループリントのカスタマイズにオプションを追加できます。

詳細は、[新機能 - インストーラーとイメージの作成](#) を参照してください

RHEL for Edge

RHEL for Edge の主なハイライト:

- FIPS 準拠の RHEL for Edge イメージを作成できるようになりました。
- このテクノロジープレビューでは、Sqlite または Postgresql データベースから Owner Voucher を保存およびクエリーすることで、FDO オンボーディングプロセスを使用できます。

詳細は、[新機能 - RHEL for Edge](#) を参照してください。

セキュリティ

SELinux ユーザー空間リリース 3.6 では、SELinux ポリシーをさらにカスタマイズするための拒否ルールが導入されます。

Keylime サーバーコンポーネント (verifier および registrar) は、コンテナとして利用できます。

Rsyslog ログ処理システムでは、カスタマイズ可能な TLS/SSL 暗号化設定と、機能のドロップに関連する追加オプションが導入されます。

OpenSSL TLS ツールキットは、プロバイダー固有の設定ファイル用のドロップインディレクトリーを追加します。

Linux カーネル暗号化 API (libkcap) 1.4.0 では、新しいツールとオプションが導入されます。特に、新しい **-T** オプションを使用すると、ハッシュ合計の計算でターゲットファイル名を指定できます。

stunnel TLS/SSL トンネリングサービス 5.71 では、FIPS モードにある OpenSSL 1.1 以降のバージョンの動作が変更されます。この変更に加えて、バージョン 5.71 では、最新の PostgreSQL クライアントのサポートなど、多くの新機能が提供されます。

詳細は、[新機能 - セキュリティ](#) 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

次の Application Streams の新しいバージョンが利用可能になる

- Python 3.12
- Ruby 3.3

- PHP 8.2
- nginx 1.24
- MariaDB 10.11
- PostgreSQL 16

以下のコンポーネントがアップグレードされました。

- Git がバージョン 2.43.0 へ
- Git LFS がバージョン 3.4.1 へ

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

コンパイラーおよび開発ツール

パフォーマンスツールとデバッガーの更新

RHEL 9.4 では、以下のパフォーマンスツールおよびデバッガーが更新されました。

- Valgrind 3.22
- SystemTap 5.0
- elfutils 0.190

更新されたパフォーマンスモニタリングツール

RHEL 9.4 では、以下のパフォーマンスモニタリングツールが更新されました。

- PCP 6.2.0

更新されたコンパイラーツールセット

次のコンパイラーツールセットが RHEL 9.4 で更新されました。

- GCC Toolset 13
- LLVM Toolset 17.0.6
- Rust Toolset 1.75.1
- Go Toolset 1.21.7

詳しい変更点については、[新機能 - コンパイラーと開発ツール](#) を参照してください。

Identity Management

Identity Management の主なハイライト:

- SSSD でパスワードレス認証を有効にして設定し、YubiKey などの FIDO2 仕様と互換性のある生体認証デバイスを使用できます。

詳細は、[新機能 - Identity Management](#) を参照してください。

仮想化

RHEL 9.4 では、64 ビット ARM アーキテクチャー上の KVM 仮想マシンの完全なサポートが導入されています。

さらに、仮想マシンの外部スナップショットが完全にサポートされ、多くのスナップショット操作のデフォルトのメカニズムになりました。

このリリースで導入された仮想化機能の詳細は、[新機能 - 仮想化](#) を参照してください。

コンテナ

主な変更点は、以下のとおりです。

- マルチアーキテクチャーコンテナイメージを作成するための **podman farm build** コマンドが、テクノロジープレビューとして利用できます。
- Podman は、事前に定義された設定セットをロードするための **containers.conf** モジュールをサポートするようになりました。
- Container Tools パッケージが更新されました。
- Podman v4.9 RESTful API では、イメージをレジストリーにプルまたはプッシュするときに、進捗データが表示されるようになりました。
- SQLite は、Podman のデフォルトのデータベースバックエンドとして完全にサポートされるようになりました。
- **Containerfile** は、複数行の HereDoc 命令をサポートするようになりました。
- ネットワーク名としての **pasta** は非推奨になりました。
- BoltDB データベースバックエンドは非推奨になりました。
- **container-tools:4.0** モジュールは非推奨になりました。
- Container Network Interface (CNI) ネットワークスタックは非推奨となり、今後のリリースでは削除される予定です。

詳細については、[新機能 - コンテナ](#) を参照してください。

1.2. インプレースアップグレード

RHEL 8 から RHEL 9 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。

- 次のアーキテクチャー上の RHEL 8.8 から RHEL 9.2、および RHEL 8.10 から RHEL 9.4:
 - 64 ビット Intel
 - 64 ビット AMD
 - 64-bit ARM
 - IBM POWER 9 (リトルエンディアン) 以降
 - z13 を除く IBM Z アーキテクチャー
- SAP HANA を搭載したシステム上の RHEL 8.8 から RHEL 9.2 および RHEL 8.10 から RHEL 9.4

インプレースアップグレードの実行方法は、[RHEL 8 から RHEL 9 へのアップグレード](#) を参照してください。

SAP 環境があるシステムでインプレースアップグレードを実行する手順については、[SAP 環境を RHEL 8 から RHEL 9 にインプレースアップグレードする方法](#) を参照してください。

Red Hat がインプレースアップグレードプロセスをサポートする方法は、[インプレースアップグレード サポートポリシー](#) を参照してください。

主な機能拡張は、次のとおりです。

- アップグレード後の **systemd** サービスの予想される状態を決定するための新しいロジックが実装されました。
- ローカルに保存された DNF リポジトリをインプレースアップグレードに使用できるようになりました。
- プロキシを使用してアップグレードできるように DNF を設定できるようになりました。
- HTTPS を使用してアクセスしたカスタム DNF リポジトリでインプレースアップグレードを実行する際の問題が修正されました。
- **/etc/pki/tls/openssl.cnf** 設定ファイルが変更されている場合は、アップグレード後に問題が発生しないように、アップグレード中にそのファイルがターゲットのデフォルトの OpenSSL 設定ファイルに置き換えられるようになりました。詳細は、アップグレード前のレポートを参照してください。

RHEL7 から RHEL 9 へのインプレースアップグレード

RHEL7 から RHEL 9 へのインプレースアップグレードを直接実行することはできません。ただし、RHEL 7 から RHEL 8 へのインプレースアップグレードを実行してから、RHEL 9 への 2 回目のインプレースアップグレードを実行することはできます。詳細は、[RHEL7 から RHEL8 へのアップグレード](#) 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セクションにあるツールセットで、<https://access.redhat.com/labs/> から入手できます。Red Hat Customer Portal Labs のアプリケーションは、パフォーマンスの向上、問題の迅速なトラブルシューティング、セキュリティ問題の特定、複雑なアプリケーションの迅速なデプロイメントおよび設定に役立ちます。最も一般的なアプリケーションには、以下のものがあります。

- [Registration Assistant](#)
- [Kickstart Generator](#)
- [Red Hat Product Certificates](#)
- [Red Hat CVE Checker](#)
- [Kernel Oops Analyzer](#)
- [Red Hat Code Browser](#)
- [VNC Configurator](#)
- [Red Hat OpenShift Container Platform Update Graph](#)
- [Red Hat Satellite Upgrade Helper](#)
- [JVM Options Configuration Tool](#)

- [Load Balancer Configuration Tool](#)
- [Red Hat OpenShift Data Foundation サポートおよび相互運用性チェッカー](#)
- [Ansible Automation Platform Upgrade Assistant](#)
- [Ceph Placement Groups \(PGs\) per Pool Calculator](#)
- [Yum Repository Configuration Helper](#)

1.4. 関連情報

他のバージョンと比較した Red Hat Enterprise Linux 9 の **機能および制限** は、Red Hat ナレッジベースの記事 [Red Hat Enterprise Linux テクノロジーの機能と制限](#) を参照してください。

Red Hat Enterprise Linux の **ライフサイクル** に関する情報は [Red Hat Enterprise Linux のライフサイクル](#) を参照してください。

パッケージマニフェスト ドキュメントは、ライセンスとアプリケーションの互換性レベルを含む、RHEL 9 の **パッケージリスト** を提供します。

アプリケーションの互換性レベル は、[Red Hat Enterprise Linux 9: アプリケーション互換性ガイド](#) ドキュメントで説明されています。

削除された機能を含む主な RHEL 8 と RHEL 9 の **相違点** は、[RHEL 9 の導入における考慮事項](#) で説明されています。

RHEL 8 から RHEL 9 への **インプレースアップグレード** を実行する方法は、[Upgrading from RHEL 8 to RHEL 9](#) 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ます。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。



注記

公開リリースノートには、元の追跡チケットにアクセスするためのリンクが含まれていますが、非公開リリースノートは表示できないため、リンクは含まれていません。[1]

[1] 公開リリースノートには、元の追跡チケットにアクセスするためのリンクが含まれていますが、非公開リリースノートは表示できないため、リンクは含まれていません。

第2章 アーキテクチャー

Red Hat Enterprise Linux 9.4 はカーネルバージョン 5.14.0-427.13.1 とともに配布されており、次のアーキテクチャーの最低限必要なバージョン (括弧内に記載) でサポートを提供します。

- AMD および Intel 64 ビットアーキテクチャー (x86-64-v2)
- 64 ビット ARM アーキテクチャー (ARMv8.0-A)
- IBM Power Systems (リトルエンディアン) (POWER9)
- 64 ビット IBM Z (z14)

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。

第3章 RHEL 9 のコンテンツの配布

3.1. インストール

Red Hat Enterprise Linux 9 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。

- インストール 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。[製品のダウンロード](#) ページでは、インストール ISO は **バイナリー DVD** と呼ばれます。



注記

インストール用 ISO イメージのサイズは複数 GB であるため、光学メディア形式には適合しない場合があります。インストール 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、USB キーまたは USB ハードドライブを使用することが推奨されます。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 [Composing a customized RHEL system image](#) 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、Installation ISO イメージの一部です。インストール中に Red Hat CDN または Satellite に登録して、Red Hat CDN または Satellite から最新の BaseOS および AppStream コンテンツを使用することもできます。

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、RHEL 9 インストールの完了の方法は、[標準的な RHEL 9 インストールの実行](#) 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 9 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 9 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、基本的なオペレーティングシステム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。詳細は、[対象範囲の詳細](#) を参照してください。

AppStream リポジトリには、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同梱されます。

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は、サポート対象外です。

RHEL 9 リポジトリとそれらが提供するパッケージの詳細は、[パッケージマニフェスト](#) を参照してください。

3.3. APPLICATION STREAMS (APPSTREAM)

複数のバージョンのユーザー空間コンポーネントが Application Streams として提供され、BaseOS リポジトリよりも頻繁に更新されます。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盤となる安定性に影響を及ぼさずに、RHEL をより柔軟にカスタマイズできます。

Application Streams は、通常の RPM 形式で、モジュールと呼ばれる RPM 形式への拡張として、Software Collections として、または Flatpak として利用できます。

各 Application Streams コンポーネントには、RHEL 9 と同じか、より短いライフサイクルが指定されています。RHEL のライフサイクル情報は、[Red Hat Enterprise Linux のライフサイクル](#) を参照してください。

RHEL 9 では、従来の **dnf install** コマンドを使用して RPM パッケージとしてインストールできる最初の Application Streams バージョンを提供することで、Application Streams エクスペリエンスを向上させています。



注記

RPM 形式を使用する初期 Application Streams の中には、Red Hat Enterprise Linux 9 よりも短いライフサイクルのものがあります。

追加の Application Streams バージョンの中には、将来のマイナー RHEL 9 リリースで、ライフサイクルが短いモジュールとして配布されるものがあります。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。

Application Streams のどのバージョンをインストールするかについては、まず [Red Hat Enterprise Linux Application Streams ライフサイクル](#) を確認してください。

代替コンパイラーやコンテナツールなど、迅速な更新を必要とするコンテンツは、代替バージョンを並行して提供しない Rolling Streams で利用できます。Rolling Streams は、RPM またはモジュールとしてパッケージ化される場合があります。

RHEL 9 で使用可能な Application Streams とそのアプリケーション互換性レベルについては、[パッケージマニフェスト](#) を参照してください。アプリケーションの互換性レベルは、[Red Hat Enterprise Linux 9: アプリケーション互換性ガイド](#) ドキュメントで説明されています。

3.4. YUM/DNF を使用したパッケージ管理

Red Hat Enterprise Linux 9 では、ソフトウェアインストールは **DNF** により保証されます。Red Hat は、以前の RHEL のメジャーバージョンとの整合性を保つため、**yum** コマンドの使用を引き続きサポートします。**yum** の代わりに **dnf** と入力しても、どちらも互換性のためのエイリアスなので、コマンドは期待通りに動作します。

RHEL 8 と RHEL 9 は **DNF** をベースにしていますが、RHEL 7 で使用していた **YUM** との互換性があります。

詳細は、[DNF ツールを使用したソフトウェアの管理](#) を参照してください。

第4章 新機能

ここでは、Red Hat Enterprise Linux 9.4 に追加された新機能および主要な機能拡張を説明します。

4.1. インストーラーおよびイメージの作成

SCAP セキュリティープロファイルのカスタマイズされたファイルをブループリントに追加するためのサポート

この機能拡張により、次のオプションを使用して、プロファイルのカスタマイズ済み調整オプションを **osbuild-composer** ブループリントのカスタマイズに追加できるようになりました。

- **selected**: 追加するルールの一覧に使用します。
- **unselected**: 削除するルールの一覧に使用します。

デフォルトの **org.ssgproject.content** ルールの名前空間の場合、この名前空間にあるルールの接頭辞を省略できます。たとえば、**org.ssgproject.content_grub2_password** と **grub2_password** は機能的に同等です。

そのブループリントからイメージをビルドすると、新しいテーラリングプロファイル ID を持つテーラリングファイルが作成され、**/usr/share/xml/osbuild-oscaped-tailoring/tailoring.xml** としてイメージに保存されます。新しいプロファイル ID には、ベースプロファイルの接尾辞として **_osbuild_tailoring** が追加されます。たとえば、**cis** ベースプロファイルを使用する場合は、**xccdf_org.ssgproject.content_profile_cis_osbuild_tailoring** となります。

Jira:RHELDOCS-17792^[1]

最小限の RHEL インストールでは、**s390utils-core** パッケージのみがインストールされるようになる

RHEL 8.4 以降では、**s390utils-base** パッケージは、**s390utils-core** パッケージと補助 **s390utils-base** パッケージに分割されています。そのため、RHEL インストールを **minimal-environment** に設定すると、必要な **s390utils-core** パッケージのみがインストールされ、補助 **s390utils-base** パッケージはインストールされません。最小限の RHEL インストールで **s390utils-base** パッケージを使用する場合は、RHEL インストールの完了後にパッケージを手動でインストールするか、キックスタートファイルを使用して **s390utils-base** を明示的にインストールする必要があります。

Bugzilla:1932480^[1]

4.2. セキュリティー

Keylime verifier および registrar コンテナが利用可能になる

Keylime サーバーコンポーネント (verifier および registrar) を、コンテナとして設定できるようになりました。コンテナ内で実行されるように設定した場合、Keylime registrar はホストにバイナリーがなくてもコンテナからテナントシステムを監視します。コンテナのデプロイメントにより、Keylime コンポーネントの分離性、モジュール性、再現性が向上します。

Jira:RHELDOCS-16721^[1]

libkcapi が、ハッシュ合計の計算でターゲットファイル名を指定するオプションを提供するようになりました

この **libkcapi** (Linux カーネル暗号化 API) パッケージの更新では、ハッシュ合計の計算でターゲット

ファイル名を指定するための新しいオプション **-T** が導入されます。このオプションの値は、処理済みの HMAC ファイルで指定されたファイル名をオーバーライドします。このオプションは、**-c** オプションとの併用でのみ使用できます。以下に例を示します。

```
$ sha256hmac -c <hmac_file> -T <target_file>
```

[Jira:RHEL-15298^{\[1\]}](#)

crypto-policies を使用した SSH の MAC のより細かい制御

システム全体の暗号化ポリシー (**crypto-policies**) で、SSH プロトコルのメッセージ認証コード (MAC) の追加オプションを設定できるようになりました。この更新により、**crypto-policies** オプション **ssh_etm** が、トライステート **etm@SSH** オプションに変換されました。以前の **ssh_etm** オプションは非推奨になりました。

ssh_etm は、次のいずれかの値に設定できるようになりました。

ANY

encrypt-then-mac と **encrypt-and-mac** の両方の MAC を許可します。

DISABLE_ETM

encrypt-then-mac MAC を許可しません。

DISABLE_NON_ETM

encrypt-then-mac を使用しない MAC を許可しません。

インプリシットな MAC を使用する暗号は、認証された暗号化を使用するため、常に許可されることに注意してください。

[Jira:RHEL-15925](#)

semanage fcontext コマンドがローカルの変更を並べ替えなくなりました

semanage fcontext -l -C コマンドは、**file_contexts.local** ファイルに保存されているローカルファイルコンテキストの変更をリスト表示します。**restorecon** ユーティリティは、**file_contexts.local** 内のエントリーを最も新しいエントリーから最も古いエントリーへと順番に処理します。以前の **semanage fcontext -l -C** は、エントリーを間違った順序でリスト表示していました。処理順序とリスト表示順序の不一致により、SELinux ルールの管理時に問題が発生していました。この更新により、**semanage fcontext -l -C** は、最も古いルールから最も新しいルールへと、期待どおりの正しい順序でルールを表示するようになりました。

[Jira:RHEL-24462^{\[1\]}](#)

SELinux ポリシーに制限されている追加サービス

この更新により、次の **systemd** サービスを制限する追加のルールが SELinux ポリシーに追加されます。

- **nvme-stas**
- **rust-afterburn**
- **rust-coreos-installer**
- **bootc**

その結果、これらのサービスは **unconfined_service_t** SELinux ラベルでは実行されなくなり、SELinux enforcing モードで正常に実行されます。

[Jira:RHEL-12591^{\[1\]}](#)

SAP HANA サービス用の新しい SELinux ポリシーモジュール

この更新により、SAP HANA サービスの SELinux ポリシーに追加のルールが追加されます。その結果、サービスは **sap_unconfined_t** ドメインの SELinux enforcing モードで正常に実行されるようになりました。

[Jira:RHEL-21452](#)

glusterd SELinux モジュールは別の glusterfs-selinux パッケージへ移動

この更新により、**glusterd** SELinux モジュールは別の **glusterfs-selinux** パッケージで管理されるようになりました。したがって、このモジュールは **selinux-policy** パッケージの一部ではなくなりました。**glusterd** モジュールに関するアクションについては、**glusterfs-selinux** パッケージをインストールして使用します。

[Jira:RHEL-1548](#)

OpenSSL 用の fips.so ライブラリーは別パッケージとして提供される

OpenSSL は、FIPS プロバイダーとして **fips.so** 共有ライブラリーを使用します。この更新により、認定のために米国立標準技術研究所 (NIST) に提出された **fips.so** の最新バージョンが、別のパッケージに含まれるようになりました。これにより、OpenSSL の将来のバージョンでは、認定済みのコードまたは認定中のコードが使用されるようになります。

[Jira:RHEL-23474^{\[1\]}](#)

chronyd-restricted サービスは SELinux ポリシーによって制限される

この更新では、新しい **chronyd-restricted** サービスを制限する追加のルールが SELinux ポリシーに追加されます。その結果、サービスは SELinux で正常に実行されるようになりました。

[Jira:RHEL-18219](#)

OpenSSL がプロバイダー設定用のドロップインディレクトリーを追加する

OpenSSL TLS ツールキットは、暗号化アルゴリズムを提供するモジュールのインストールおよび設定のためのプロバイダー API をサポートしています。この更新により、メインの OpenSSL 設定ファイルを変更せずに、プロバイダー固有の設定を **/etc/pki/tls/openssl.d** ディレクトリー内の個別の **.conf** ファイルに配置できるようになります。

[Jira:RHEL-17193](#)

SELinux ユーザー空間コンポーネントが 3.6 にリベース

SELinux ユーザー空間コンポーネント

libsepol、**libselinux**、**libsemanage**、**policycoreutils**、**checkpolicy**、および **mcstrans** ライブラリーパッケージが 3.6 にリベースされました。このバージョンでは、さまざまなバグ修正、最適化、機能拡張が行われていますが、特に注目すべき点は次のとおりです。

- CIL の **deny** ルールのサポートが追加されました。
- CIL の **notself** および **other** キーワードのサポートが追加されました。

- 現在のシステムで実行されたポリシーの再ロードの数を出力する **getpolicyload** バイナリーを追加しました。

[Jira:RHEL-16233](#)

GnuTLS が 3.8.3 にリベース

GnuTLS パッケージはアップストリームバージョン 3.8.3 にリベースされました。このバージョンでは、さまざまなバグ修正と機能拡張が行われていますが、特に注目すべき点は次のとおりです。

- **gnutls_hkdf_expand** 関数は、RFC 5869 2.3 に準拠するために、ハッシュダイジェストサイズの 255 倍以下の長さの引数のみを受け入れるようになりました。
- **TLS PSK** ユーザー名の長さ制限が 65535 文字に増加されました。
- **gnutls_session_channel_binding** API 関数は、RFC9622 4.2 に従って **GNUTLS_CB_TLS_EXPORTER** が要求されたときに追加のチェックを実行します。
- **GNUTLS_NO_STATUS_REQUEST** フラグと **%NO_STATUS_REQUEST** 優先度修飾子が追加され、クライアント側で **status_request** TLS エクステンションを無効にできるようになりました。
- GnuTLS は、TLS バージョンが 1.3 より古い場合、Change Cipher Spec メッセージの内容が 1 に等しいかチェックするようになりました。
- ClientHello エクステンションの順序は、デフォルトでランダム化されます。
- GnuTLS は、以前は機能しなかった PKCS #11 トークンでの EdDSA キー生成をサポートするようになりました。

[Jira:RHEL-14891^{\[1\]}](#)

nettle が 3.9.1 にリベース

nettle ライブラリーパッケージが 3.9.1 にリベースこのバージョンでは、さまざまなバグ修正、最適化、機能拡張が行われていますが、特に注目すべき点は次のとおりです。

- バルーンパスワードハッシュの追加
- SIV-GCM 認証付き暗号化モードの追加
- Offset Codebook Mode 認証付き暗号化モードの追加
- 64 ビット IBM Z、AMD、Intel 64 ビットアーキテクチャーでの SHA-256 ハッシュ関数のパフォーマンスの向上
- IBM Power Systems、Little Endian、AMD、Intel 64 ビットアーキテクチャーにおける Poly1305 ハッシュ関数のパフォーマンスの向上

[Jira:RHEL-14890^{\[1\]}](#)

p11-kit が 0.25.3 にリベース

p11-kit パッケージが、アップストリームバージョン 0.25.3 に更新されました。このパッケージには、PKCS#11 モジュールを管理するための **p11-kit** ツール、信頼ポリシーストアを操作するための **trust** ツール、および **p11-kit** ライブラリーが含まれます。注目すべき機能拡張は次のとおりです。

- PKCS #11 バージョン 3.0 のサポートを追加しました。

- **pkcs11.h** ヘッダーファイル:
 - ChaCha20/Salsa20、Poly1305、IBM 固有のメカニズムと属性を追加しました。
 - メッセージベースの暗号化に AES-GCM メカニズムパラメーターを追加しました。
- **p11-kit** ツール:
 - トークンのオブジェクトをリスト表示および管理するためのユーティリティーコマンド (**list-tokens**、**list-mechanisms**、**list-objects**、**import-object**、**export-object**、**delete-object**、**generate-keypair**) を追加しました。
 - トークンの PKCS#11 プロファイルを管理するためのユーティリティーコマンド (**list-profiles**、**add-profile**、**delete-profile**) を追加しました。
 - マージされた設定を出力するための **print-config** コマンドを追加しました。
- **trust** ツール:
 - **.p11-kit** ファイルの形式を検証するための **check-format** コマンドを追加しました。

[Jira:RHEL-14834^{\[1\]}](#)

libkcapi が 1.4.0 にリベース

Linux カーネル暗号化 API へのアクセスを提供する **libkcapi** ライブラリーは、アップストリームバージョン 1.4.0 にリベースされました。この更新には、さまざまな機能拡張とバグ修正が含まれています。主なものは次のとおりです。

- **sm3sum** および **sm3hmac** ツールを追加しました。
- **kcapi_md_sm3** および **kcapi_md_hmac_sm3** API を追加しました。
- SM4 の便利な機能を追加しました。
- リンク時最適化 (LTO) のサポートを修正しました。
- LTO リグレッションテストを修正しました。
- **kcapi-enc** による任意サイズの AEAD 暗号化のサポートを修正しました。

[Jira:RHEL-5367^{\[1\]}](#)

OpenSSH でのユーザーとグループの作成に **sysusers.d** 形式を使用します

以前、OpenSSH は静的な **useradd** スクリプトを使用していました。この更新により、OpenSSH は **sysusers.d** 形式を使用してシステムユーザーを宣言するようになり、システムユーザーをイントロスペクションできるようになりました。

[Jira:RHEL-5222](#)

OpenSSH は認証における人為的な遅延を制限します

ログイン失敗後の OpenSSH の応答は、ユーザー列挙攻撃を防ぐために人為的に遅延されます。この更新では、特権アクセス管理 (PAM) の処理などでリモート認証に時間がかかりすぎる場合に、このような遅延に対して上限を導入します。

[Jira:RHEL-2469^{\[1\]}](#)

stunnel が 5.71 にリベースされま3した

stunnel TLS/SSL トンネリングサービスが、アップストリームバージョン 5.71 にリベースされました。

主な新機能は次のとおりです。

- 最新の PostgreSQL クライアントのサポートが追加されました。
- **protocolHeader** サービスレベルオプションを使用して、カスタム **connect** プロトコルネゴシエーションヘッダーを挿入できます。
- **protocolHost** オプションを使用して、クライアントの SMTP プロトコルネゴシエーションの HELO/EHLO 値を制御できます。
- クライアントサイドの **protocol = ldap** に関するクライアントサイドサポートが追加されました。
- サービスレベルの **sessionResume** オプションを使用して、セッション再開を設定できるようになりました。
- **CApath** を使用したサーバーモードでのクライアント証明書の要求に対するサポートが追加されました (以前は **CAfile** のみがサポートされていました)。
- ファイルの読み取りとロギングのパフォーマンスが向上しました。
- **retry** オプションの設定可能な遅延のサポートが追加されました。
- クライアントモードでは、**verifyChain** が設定されている場合に OCSP ステージングの要求および検証が行われます。
- サーバーモードでは、OCSP ステージングは常に利用可能です。
- 不確定の OCSP 検証により TLS ネゴシエーションが中断されます。これを無効にするには、**OCSPRequire = no** と設定します。

Jira:RHEL-2468^[1]

Rsyslog で機能をドロップするための新しいオプション

次のグローバルオプションを使用して、機能をドロップするときの Rsyslog の動作を設定できるようになりました。

libcapng.default

機能のドロップ中にエラーが発生した場合の Rsyslog のアクションを決定します。デフォルト値は **on** です。この場合、**libcapng-related** 関連のエラーが発生した場合に Rsyslog が終了します。

libcapng.enable

Rsyslog が起動中に機能をドロップするかどうかを決定します。このオプションを無効にすると、**libcapng.default** は影響を与えません。

Jira:RHEL-943^[1]

audit が 3.1.2 にリベース

Linux の Audit システムがバージョン 3.1.2 に更新されました。これにより、以前にリリースされたバージョン 3.0.7 に対するバグ修正、機能拡張、およびパフォーマンス向上が実現しました。主な機能拡張は、次のとおりです。

- **auparse** ライブラリーは、名前のないソケットと匿名ソケットを解釈するようになりました。
- **ausearch** および **aureport** ツールの **start** オプションと **end** オプションで、新しいキーワード **this-hour** を使用できます。
- **io_uring** 非同期 I/O API のサポートが追加されました。
- シグナル用のユーザーフレンドリーなキーワードが、**auditctl** プログラムに追加されました。
- **auparse** での破損したログの処理が改善されました。
- **auditd** サービスで、**ProtectControlGroups** オプションがデフォルトで無効になりました。
- 除外フィルターのルールチェックが修正されました。
- **OPENAT2** フィールドの解釈が改善されました。
- **audispd af_unix** プラグインがスタンドアロンプログラムに移動しました。
- Python バインディングが変更され、Python API から Audit ルールが設定できなくなりました。この変更は、Simplified Wrapper and Interface Generator (SWIG) のバグのために行われました。

Jira:RHEL-14896^[1]

Rsyslog が 8.2310 にリベース

Rsyslog ログ処理システムがアップストリームバージョン 8.2310 にリベースされました。この更新では、重要な機能拡張とバグ修正が導入されました。主な機能拡張は、次のとおりです。

カスタマイズ可能な TLS/SSL 暗号化設定

以前のバージョンでは、個別の接続に対する TLS/SSL 暗号化設定はグローバル設定に限定されていました。最新バージョンでは、Rsyslog 内の個々の接続それぞれに固有の TLS/SSL 設定を定義できるようになりました。これには、セキュリティと柔軟性を強化するために、さまざまな CA 証明書、秘密鍵、公開鍵、および CRL ファイルを指定することが含まれます。詳細な情報と使用方法については、**rsyslog-doc** パッケージで提供されるドキュメントを参照してください。

機能のドロップ機能の改良

機能のドロップに関連する追加オプションを設定できるようになりました。**libcpng.enable** グローバルオプションを **off** に設定することで、機能のドロップを無効にできます。詳細は、[RHEL-943](#) を参照してください。

[Jira:RHEL-937](#), [Jira:RHEL-943](#)

SCAP セキュリティガイドが 0.1.72 にリベース

SCAP セキュリティガイド (SSG) パッケージが、アップストリームバージョン 0.1.72 にリベースされました。このバージョンでは、バグ修正とさまざまな機能拡張が行われています。主なものは次のとおりです。

- CIS プロファイルは最新のベンチマークに合わせて更新されます。
- PCI DSS プロファイルは、PCI DSS ポリシーバージョン 4.0 に準拠します。
- STIG プロファイルは、最新の DISA STIG ポリシーに準拠します。

詳細は、[SCAP Security Guide release notes](#) を参照してください。

[Jira:RHEL-21425](#)

4.3. RHEL FOR EDGE

FIPS 対応 RHEL for Edge イメージのビルドのサポート

この機能拡張により、次のイメージタイプに対して FIPS 対応の RHEL for Edge イメージをビルドするためのサポートが追加されます。

- **edge-installer**
- **edge-simplified-installer**
- **edge-raw-image**
- **edge-ami**
- **edge-vmware**



重要

FIPS モードを有効にできるのは、イメージのプロビジョニングプロセス中のみです。非 FIPS イメージのビルドを開始した後に FIPS モードに変更することはできません。

[Jira:RHELDPCS-17263^{\[1\]}](#)

4.4. シェルおよびコマンドラインツール

openCryptoki がバージョン 3.22.0 にリベース

opencryptoki パッケージがバージョン 3.22.0 に更新されました。主な変更点は、以下のとおりです。

- **CPACF** で保護されたキーを使用することで、**AES-XTS** キータイプのサポートを追加しました。
- 証明書オブジェクト管理のサポートが追加されました。
- **no-login** オプションによるパブリックセッションのサポートが追加されました。
- セキュリティーオフィサー (SO) としてログインするためのサポートが追加されました。
- **Edwards** キーと **Montgomery** キーのインポートおよびエクスポートのサポートが追加されました。
- **RSA-PSS** キーと証明書のインポートのサポートが追加されました。
- セキュリティー上の理由から、AES-XTS キーの 2 つのキー部分を同一にすることはできません。この更新では、同一になるのを確実に避けるために、キー生成およびインポートプロセスに対するチェックが追加されます。
- さまざまなバグ修正が実装されました。

[Jira:RHEL-11412^{\[1\]}](#)

4.5. インフラストラクチャーサービス

synce4l がバージョン 1.0.0 にリベース

synce4l プロトコルがバージョン 1.0.0 に更新されました。この更新では、カーネルの Digital Phase Locked Loop (DPLL) インターフェイスのサポートが追加されます。

[Jira:RHEL-10089^{\[1\]}](#)

chrony がバージョン 4.5 にリベース

chrony スイートがバージョン 4.5 に更新されました。主な変更点は、以下のとおりです。

- インターネット上の NTS の信頼性を向上させるために、Network Time Security (NTS) クッキーを短縮する AES-GCM-SIV 暗号のサポートが追加されました (インターネットでは、一部のプロバイダーが長い Network Time Protocol (NTP) メッセージをブロックまたはレート制限します)。
- ホスト名で指定された NTP ソースの IP アドレスの定期的な更新を追加しました。デフォルトの間隔は 2 週間です。**chrony.conf** ファイルに **refresh 0** パラメーターを追加することで無効にできます。
- 到達不能な NTP ソースの自動置換が改善されました。
- **chronyc** ユーティリティーによって行われる重要な変更のロギングが改善されました。
- ソース選択の失敗と **falseticker** のロギングが改善されました。
- 遅延ハードウェア送信タイムスタンプのタイムアウトを設定するための **hwtsttimeout** ディレクティブを追加しました。
- ハードウェアタイムスタンプで PTP の精度に到達するために、Precision Time Protocol (PTP) 透過クロックが提供する修正に対する実験的なサポートが追加されました。
- **root** 権限なしで **chronyd** サービスを起動できる最小限のクライアント専用設定用の代替サービスとして、**chronyd-restricted** サービスを追加しました。
- **interleaved** モードの **presend** オプションを修正しました。
- IP アドレスで指定された変更済みソースの **sourcedir** ディレクトリーからのリロードを修正しました。

[Jira:RHEL-6522](#)

linuxptp がバージョン 4.2 にリベース

linuxptp プロトコルがバージョン 4.2 に更新されました。主な変更点は、以下のとおりです。

- **phc2sys** ユーティリティーに複数のドメインのサポートが追加されました。
- Precision Time Protocol (PTP) の親データセット (クロッククラスなど) におけるクロックの更新および変更に関する通知のサポートが追加されました。
- PTP Power Profile (IEEE C37.238-2011 および IEEE C37.238-2017) のサポートが追加されました。

[Jira:RHEL-2026](#)

4.6. ネットワーク

nft ユーティリティーが、nftables ルールに含まれる状態をリセットできるようになりました

この機能拡張により、**nft reset** コマンドを使用して、**nftables** ルールに含まれる状態をリセットできるようになりました。たとえば、この機能を使用して、カウンターとクォータステートメントの値をリセットします。

Jira:RHEL-5980^[1]

Marvell Octeon PCIe Endpoint Network Interface Controller ドライバーが利用可能になる

この機能拡張により、**octeon_ep** ドライバーが追加されました。Marvell の Octeon PCIe Endpoint ネットワークインターフェイスカードのネットワークに使用できます。ホストドライバーは、PCI Express (PCIe) エンドポイントネットワークインターフェイス (NIC) として機能し、24 N2 コアのインフラストラクチャプロセッサファミリーである Marvell OCTEON TX2 CN106XX をサポートします。OCTEON TX2 ドライバーを PCIe NIC として使用することで、セキュリティーファイアウォール、5G Open Radio Access Network (ORAN) と Virtual RAN (VRAN) アプリケーション、データ処理オフロードアプリケーションなど、さまざまな製品の PCIe エンドポイントとして OCTEON TX2 を使用できます。

現在、以下のデバイスでご利用いただけます。

- Network controller: Cavium, Inc.Device b100
- Network controller: Cavium, Inc.Device b200
- Network controller: Cavium, Inc.Device b400
- Network controller: Cavium, Inc.Device b900
- Network controller: Cavium, Inc.Device ba00
- Network controller: Cavium, Inc.Device bc00
- Network controller: Cavium, Inc.Device bd00

Jira:RHEL-9308^[1]

NetworkManager は、高度なハードウェアオフロード用の switchdev モードの設定をサポートするようになりました

この機能拡張により、NetworkManager 接続プロファイルで次の新しいプロパティを設定できるようになります。

- **sriov.eswitch-mode**
- **sriov.eswitch-inline-mode**
- **sriov.eswitch-encap-mode**

これらのプロパティを使用して、スマートネットワークインターフェイスコントローラー (Smart NIC) の eSwitch を設定できます。たとえば、高度なハードウェアオフロード機能を使用するには、**sriov.eswitch-mode** 設定を使用して、モードを **legacy SR-IOV** から **switchdev** に変更します。

Jira:RHEL-1441

NetworkManager は ethtool チャンネル設定の変更をサポートします

ネットワークインターフェイスには、複数の割り込み要求 (IRQ) と、**channels** と呼ばれる関連パケットキューを設定できます。この機能拡張により、NetworkManager 接続プロファイルは、接続プロパティ **ethtool.channels-rx**、**ethtool.channels-tx**、**ethtool.channels-other**、および **ethtool.channels-combined** を使用して、インターフェイスに割り当てるチャンネルの数を指定できるようになりました。

[Jira:RHEL-1471^{\[1\]}](#)

Nmstate は設定を元に戻すための YAML ファイルを作成できるようになりました

この機能拡張により、Nmstate は、現在のネットワーク設定と、適用する新しい設定を含む YAML ファイルとの相違点を含む "元に戻すための設定ファイル" を作成できるようになります。YAML ファイルを適用した後、設定が想定どおりに機能しない場合は、元に戻すための設定ファイルを使用して以前の設定を復元できます。

1. 適用する設定を含む YAML ファイル (たとえば、**new.yml**) を作成します。
2. **new.yml** 内の意図した設定と現在の状態の相違点を含む、元に戻すための設定ファイルを作成します。

```
# nmstatectl gr new.yml > revert.yml
```

3. **new.yml** の設定を適用します。
4. 以前の状態に戻す場合は、**revert.yml** を適用します。

あるいは、Nmstate API を使用して元に戻す設定を作成する場合は、**NetworkState::generate_revert(current)** 呼び出しを使用することもできます。

[Jira:RHEL-1434](#)

Nmstate API は IPsec 設定に基づいて VPN 接続を設定します

Libreswan ユーティリティーは、VPN を設定するための IPsec の実装です。この更新により、**nmstatectl** を使用して、設定モード (トンネルとトランスポート) およびネットワークレイアウト (**host-to-host**、**host-to-host**、**subnet-to-subnet**) に加え、IPsec ベースの認証タイプを設定できるようになりました。

[Jira:RHEL-1605](#)

nmstate が priority ボンディングプロパティをサポートするようになりました

この更新では、設定ファイルの **ports-config** セクションの **priority** プロパティを使用して、**nmstate** フレームワーク内のボンディングポートの優先順位を設定できるようになりました。YAML ファイルの例は次のようになります。

```
---
interfaces:
- name: bond99
  type: bond
  state: up
  link-aggregation:
    mode: active-backup
```

```
ports-config:
- name: eth2
  priority: 15
```

ボンディングされたインターフェイス内のアクティブポートがダウンすると、RHEL カーネルは、すべてのバックアップポートのプールから、**priority** プロパティの数値が最も高い次のアクティブポートを選択します。

priority プロパティは、ボンディングインターフェイスの次のモードに関連します。

- **active-backup**
- **balance-tlb**
- **balance-alb**

[Jira:RHEL-1438^{\[1\]}](#)

NetworkManager Wi-Fi 接続が新しい MAC アドレスベースのプライバシーオプションをサポートします

この機能拡張により、無作為に生成された MAC アドレスを Wi-Fi ネットワークのサービスセット識別子 (SSID) に関連付けるように NetworkManager を設定できるようになりました。これにより、接続プロファイルを削除して再作成しても、ランダムでありつつ一貫性のある MAC アドレスを Wi-Fi ネットワークで永続的に使用できます。この新機能を使用するには、Wi-Fi 接続プロファイルの **802-11-wireless.cloned-mac-address** プロパティを **stable-ssid** に設定します。

[Jira:RHEL-16470](#)

VLAN インターフェイスの新しい nmstate 属性の導入

今回の **nmstate** フレームワークの更新により、以下の VLAN 属性が導入されました。

- **registration-protocol**: VLAN Registration Protocol。有効な値は **gvrp** (GARP VLAN Registration Protocol)、**mvrp** (Multiple VLAN Registration Protocol)、および **none** です。
- **reorder-headers**: 出力パケットヘッダーを並び替えます。有効な値は **true** および **false** です。
- **loose-binding**: プライマリーデバイスの操作状態に対してインターフェイスを緩やかにバインドします。有効な値は **true** および **false** です。

YAML 設定ファイルは以下の例のようになります。

```
---
interfaces:
- name: eth1.101
  type: vlan
  state: up
  vlan:
    base-iface: eth1
    id: 101
    registration-protocol: mvrp
    loose-binding: true
    reorder-headers: true
```

[Jira:RHEL-19142](#)

ipv4.dhcp-client-id を none に設定すると、client-identifier が送信されません

NetworkManager で **client-identifier** オプションが設定されていない場合、実際の値は、使用されている DHCP クライアントのタイプ (NetworkManager の **internal** DHCP クライアントや **dhclient** など) によって異なります。通常、DHCP クライアントは **client-identifier** を送信します。したがって、ほとんどの場合、**none** オプションを設定する必要はありません。したがって、このオプションは、クライアントが **client-identifier** を送信しないことを要求する、一部の特殊な DHCP サーバー設定の場合にのみ役立ちます。

[Jira:RHEL-1469](#)

nmstate が MACsec インターフェイスの作成をサポートするようになりました

この更新により、**nmstate** フレームワークのユーザーは、Open Systems Interconnection (OSI) モデルのレイヤー 2 で通信を保護するように、MACsec インターフェイスを設定できるようになりました。そのため、後からレイヤー 7 で個々のサービスを暗号化する必要はありません。また、この機能により、関連する課題 (エンドポイントごとに大量の証明書进行管理するなど) がなくなります。

詳細は、[nmstatectl を使用した MACsec 接続の設定](#) を参照してください。

[Jira:RHEL-1420](#)

netfilter の更新

RHEL 9 では、**kernel** パッケージがバージョン 5.14.0-405 にアップグレードされました。その結果、リベースにより、RHEL カーネルの **netfilter** コンポーネントにおいて複数の機能拡張とバグ修正も行われました。以下は、主な変更点です。

- **nftables** サブシステムは、トンネルパケットのさまざまな内部ヘッダーフィールドにマッチできます。これにより、特にトンネリングプロトコルが使用される環境において、ネットワークトラフィックのより詳細かつ効果的な制御が可能になります。

[Jira:RHEL-16630^{\[1\]}](#)

firewalld が不要なファイアウォールルールのフラッシュを実行しなくなりました

firewalld サービスは、以下の条件を両方満たす場合、**iptables** 設定から既存のルールすべてを削除しません。

- **firewalld** が **nftables** バックエンドを使用している。
- **--direct** オプションを使用して作成されたファイアウォールルールが存在しない。

この変更は、不要な操作 (ファイアウォールルールのフラッシュ) を削減し、他のソフトウェアとの統合を改善することを目的としています。

[Jira:RHEL-427^{\[1\]}](#)

ss ユーティリティーが TCP バインドされた非アクティブソケットの可視性を改善する

iproute2 スイートは、TCP/IP ネットワークトラフィックを制御するためのユーティリティーのコレクションを提供します。TCP バインドされた非アクティブソケットは、IP アドレスとポート番号に接続されていますが、両方とも TCP ポートに接続もリッスンもされていません。ソケットサービス (**ss**) ユーティリティーは、TCP バインドされた非アクティブソケットをダンプするためのカーネルのサポートを追加します。次のコマンドオプションを使用して、これらのソケットを表示できます。

- **ss --all**: TCP バインドされた非アクティブソケットを含むすべてのソケットをダンプします。

- **ss --bound-inactive**: バインドされた非アクティブソケットのみをダンプします。

[Jira:RHEL-21223^{\[1\]}](#)

Nmstate API が SR-IOV VLAN 802.1ad のタグ付けをサポートするようになりました

この機能拡張により、Nmstate API を使用して、ハードウェアアクセラレーションされた Single-Root I/O Virtualization (SR-IOV) 仮想ローカルエリアネットワーク (VLAN) 802.1ad のタグ付けを有効にできるようになりました。このタグ付けは、ファームウェアでこの機能がサポートされているカードで有効にできます。

[Jira:RHEL-1425](#)

TCP Illinois 輻輳アルゴリズムカーネルモジュールが再度有効化されました

TCP Illinois は TCP プロトコルのバリエーションです。インターネットサービスプロバイダー (ISP) などのお客様は、TCP Illinois アルゴリズムがないとパフォーマンスが最適に保たれず、Bandwidth and Round-trip propagation time (BBR) アルゴリズムを使用してもネットワークトラフィックが適切にスケーリングされず、結果として待ち時間が長くなります。したがって、TCP Illinois アルゴリズムを使用することで、平均スループットが若干向上し、ネットワークリソースがより公平に割り当てられ、互換性も実現できます。

[Jira:RHEL-5736^{\[1\]}](#)

iptables ユーティリティーがバージョン 1.8.10 にリベース

iptables ユーティリティーは、ファイアウォールを管理するためのパケットフィルタリングのルールを定義します。このユーティリティーはリベースされました。主な変更点は、以下のとおりです。

注目すべき機能:

- **sctp** マッチでの新しいチャンクタイプのサポートの追加
- ip6tables オプトイン列を空の場合にそろえると、出力を **jc --iptables** にパイプするときに役立つ
- より安定した出力のために、**--numeric** で数値プロトコル番号を出力する
- 出力フォーマットの改善による ***tables-translate** ユーティリティーの翻訳の増加
- man ページのいくつかの改善

注目すべき修正点:

- **iptables-restore** エラーメッセージが COMMIT 行を誤って指している
- ebtables での **-p Length** の一致が壊れている
- 複数のルールで使用されている場合、マッチ間で壊れた ebtables が **ebtables-restore** によって復元される
- すでに存在するチェーンの数に応じてチェーンの名前を変更すると、プログラムがクラッシュする可能性がある
- 重大でないメモリーリーク
- nft-variants への切り替え後、ebtables で broute テーブルのサポートが失われる

- '-c' オプションによる ip6tables ルールカウンター設定が壊れている
- 存在しないチェーンをリストするときに予期しないエラーメッセージが表示される
- among match が使用されている場合、ebtables ルールの比較で誤検出が発生する可能性がある
- チェーンの名前を無効な名前に変更することを禁止する
- 無効なチェーン名を検出するために、iptables-restore 入力の "chain lines" をより厳密にチェックする
- 機能しないビルトインチェーンポリシーカウンター

[Jira:RHEL-14147](#)

nftables がバージョン 1.0.9 にリベース

nftables ユーティリティーがバージョン 1.0.9 にアップグレードされ、複数のバグ修正と機能拡張が提供されています。主な変更点は、以下のとおりです。

- **--optimize** コマンドオプションの改善
- Python nftables クラスの拡張
- **iptables-nft** によって作成されたルールを処理する際の動作の改善
- vxlan-encapsulated ヘッダーのフィールドへのアクセスに対するサポート
- GRE、Geneve、GRETAP プロトコルの初期サポート
- ルールカウンター、クォータをリセットするための新しい **reset rule(s)** コマンド
- 新しい **destroy** コマンドは存在するものだけを削除する
- 最後にパケットを確認したときに、新しい **last** ステートメントを記録する
- netdev-family チェーンからデバイスを追加および削除する
- ebtables の broute 機能をエミュレートする新しい **meta broute** 式
- さまざまなメモリーリークを修正する
- コーナーケースでのエラーメッセージの間違ったロケーションを修正する
- JSON 出力に set および map ステートメントがない

[Jira:RHEL-14191](#)

firewalld がバージョン 1.3 にリベース

firewalld パッケージがバージョン 1.3 にアップグレードされ、複数のバグ修正と機能拡張が提供されます。主な変更点は、以下のとおりです。

- 新しい **--reset-defaults** CLI オプション: このオプションは、**firewalld** サービスの設定をデフォルトにリセットします。これにより、ユーザーは **firewalld** の設定を消去し、デフォルト設定からやり直すことができます。
- **Ingress-zone=ZONE** のポリシーに対して **--add-masquerade** CLI オプションを有効にしま

す。ここで、**ZONE** には **--add-interface** CLI オプションで割り当てられたインターフェイスがあります。これにより制限がなくなり、一般的なシナリオで (ソースの代わりに) インターフェイスを使用できるようになります。

これらの機能を導入する理由:

- ファイアウォールをデフォルト設定にリセットするために **--reset-defaults** が実装されました。
- インターフェイスを使用すると、ファイアウォールの設定に影響を与えずに IP アドレスを変更できます。

その結果、ユーザーは次のアクションを実行できます。

- 設定をリセットする。
- ポリシーを使用する際に **--add-maquerade** と **--add-interface** を組み合わせて使用する。

[Jira:RHEL-14485](#)

4.7. カーネル

RHEL 9.4 のカーネルバージョン

Red Hat Enterprise Linux 9.4 は、カーネルバージョン 5.14.0-427.13.1 で配布されます。

rteval は、デフォルトの測定 CPU リストに対する任意の CPU の追加および削除をサポートするようになりました

rteval ユーティリティーを使用すると、**--measurement-cpulist** パラメーターの使用時に、新しいリスト全体を指定するのではなく、デフォルトの測定 CPU リストに CPU を追加 (+ 記号を使用) または削除 (- 記号を使用) することができます。さらに、分離されたすべての CPU のセットをデフォルトの測定 CPU リストに追加するために **--measurement-run-on-isolcpus** が導入されました。このオプションは、分離された CPU 上で実行されるリアルタイムアプリケーションの最も一般的なユースケースをカバーします。その他のユースケースでは、より一般的な機能が必要になります。たとえば、一部のリアルタイムアプリケーションでは、ハウスキーピングのために分離 CPU を 1 つ使用していました。当該 CPU は、デフォルトの測定 CPU リストから除外する必要がありました。その結果、デフォルトの測定 CPU リストに任意の CPU を追加するだけでなく、柔軟な方法で任意の CPU を削除することもできるようになりました。削除は追加よりも優先されます。このルールは、+/- 記号を使用して指定した CPU と、**--measurement-run-on-isolcpus** で定義した CPU の両方に適用されます。

[Jira:RHEL-9912^{\[1\]}](#)

rtla がアップストリーム kernel ソースコードのバージョン 6.6 にリベース

rtla ユーティリティーが最新のアップストリームバージョンにアップグレードされ、複数のバグ修正および機能拡張が追加されました。主な変更点は、以下のとおりです。

- メインの **rtla** スレッドとは別に、実行する **rtla** スレッドの追加コントロールグループを指定する **-C** オプションが追加されました。
- **rtla** スレッドをハウスキーピング CPU に配置し、測定スレッドを異なる CPU に配置する **--house-keeping** オプションが追加されました。
- **timerlat hist** および **timerlat top** スレッドをユーザー空間で実行できるように、**timerlat** トレーサーのサポートが追加されました。

Jira:RHEL-10079^[1]

cyclicdeadline がレイテンシーのヒストグラムの生成をサポートするようになりました

このリリースでは、**cyclicdeadline** ユーティリティーはレイテンシーのヒストグラムの生成をサポートします。この機能を使用すると、最悪のケースの数値を1つだけ取得するのではなく、さまざまなサイズのレイテンシースパイクの頻度についてより詳細な情報を得ることができます。

Jira:RHEL-9910^[1]

SGX が完全にサポートされるようになりました

Software Guard Extensions(SGX) は、ソフトウェアコードおよび公開および修正からのデータを保護する Intel® テクノロジーです。

RHEL カーネルは、SGX バージョン 1 および 2 の機能を提供します。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で SGX テクノロジーを使用できるようになります。バージョン 2 では、**Enclave Dynamic Memory Management(EDMM)** が追加されています。主な変更には以下のものがあります。

- 初期化されたエンクレーブに属する通常のエンクレーブページの EPCM 権限を変更します。
- 初期化されたエンクレーブへの通常のエンクレーブページの動的追加。
- より多くのスレッドを収容できるように初期化されたエンクレーブを拡張します。
- 初期化されたエンクレーブから通常のページと TCS ページを削除します。

このリリースでは、SGX はテクノロジープレビューから完全にサポートされる機能に移行します。

Bugzilla:2041883^[1]

Intel データストリーミングアクセラレータードライバーが完全にサポートされるようになりました

Intel データストリーミングアクセラレータードライバー (IDX) は、Intel CPU 統合アクセラレーターを提供するカーネルドライバーです。これには、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が含まれます。

このリリースでは、IDX はテクノロジープレビューから完全にサポートされる機能に移行します。

Jira:RHEL-10097^[1]

eBPF 機能が Linux カーネルバージョン 6.6 にリベース

注目すべき変更点と機能拡張は次のとおりです。

- BPF プログラムでのデータおよび可変サイズのアクセスによる、より人間工学的で脆弱性の少ない反復が可能になる **skb** および **xdp** タイプの新しい動的ポインター (**dynptrs**)。
- 新しい BPF **netfilter** プログラムタイプと、事前ルーティングや転送などの **netfilter** フックに BPF プログラムをフックするための最小限のサポート。
- カーネルポインター (**kptrs**) に対する複数の改善点:
 - **kptrs** は、より多くの map タイプで使用できます。
 - タスク **kptrs** に対して RCU セマンティクスが有効化されます。

- BPF **list** および **rbtree** の両方にノードを追加するのに役立つ、新しい参照カウンタローカル **kptrs**。
- ロード時に、BPF プログラムは特定の **kfunc** が存在するかどうかを検出できます。
- **dynptr**、**cgroups**、**sockets**、**cpumasks** を使用するためのいくつかの新しい **kfuncs**。
- 複数の **uprobes** と **usdt** プローブをアタッチするための新しい BPF リンク。これにより、速度が大幅に向上し、余分なファイル記述子 (FD) が節約されます。
- BPF **map** 要素カウントは、すべてのプログラムタイプで有効になります。
- すべての BPF **map** タイプのメモリー使用量レポートがより正確になります。
- **bpf_fib_lookup** BPF ヘルパーには、ルーティングテーブル ID が含まれます。
- **BPF_OBJ_PIN** および **BPF_OBJ_GET** コマンドは **O_PATH** FD をサポートします。

[Jira:RHEL-10691^{\[1\]}](#)

libbpf-tools パッケージが IBM Z で利用可能になりました

BPF Compiler Collection (BCC) 用のコマンドラインツールを提供する **libbpf-tools** パッケージが、IBM Z アーキテクチャーで利用できるようになりました。その結果、IBM Z で **libbpf-tools** のコマンドを使用できるようになりました。

[Jira:RHEL-16325^{\[1\]}](#)

4.8. ブートローダー

起動前段階での DEP/NX サポート

Data Execution Prevention (DEP)、No Execute (NX)、または Execute Disable (XD) と呼ばれるメモリー保護機能は、実行不可としてマークされたコードの実行をブロックします。DEP/NX は、RHEL のオペレーティングシステムレベルで利用可能になりました。

このリリースでは、GRUB および **shim** ブートローダーに DEP/NX サポートが追加されました。これにより、特定の脆弱性 (DEP/NX 保護がなければ特定の攻撃を開始する可能性のある、悪意のある EFI ドライバーなど) を起動前の段階で防ぐことができます。

[Jira:RHEL-10288^{\[1\]}](#)

4.9. ファイルシステムおよびストレージ

ファイルシステムのサイズ制限の設定がサポートされるようになる

この更新により、ユーザーはファイルシステムを作成または変更する際に、ファイルシステムのサイズ制限を設定できるようになりました。**stratisd** サービスは、動的なファイルシステムの拡張を可能にしますが、XFS ファイルシステムを過度に拡張すると重大なパフォーマンスの問題が発生する可能性があります。この機能の追加により、XFS ファイルシステムが特定のしきい値を超えて拡張された際に発生する可能性のあるパフォーマンスの問題が対処されます。ファイルシステムのサイズ制限を設定することで、ユーザーはこのような問題を回避し、最適なパフォーマンスを確保できます。さらに、この機能により、ユーザーがファイルシステムのサイズに上限を設定できるようになり、効率的なリソース割り当てが保証されるため、プールのモニタリングとメンテナンスが向上します。

[Jira:RHEL-12898](#)

lvconvert を使用して標準 LV をシン LV に変換できるようになりました

標準論理ボリューム (LV) をシンプルデータとして指定することで、**lvconvert** コマンドを使用して標準 LV をシン LV に変換できるようになりました。この更新により、既存の LV を変換してシンプロビジョニング機能を使用できるようになります。

[Jira:RHEL-8357](#)

multipathd が、NVMe デバイスの FPIN-Li イベントの検出をサポートするようになりました

以前は、**multipathd** コマンドは SCSI デバイス上の Integrity Fabric Performance Impact Notification (PFIN-Li) イベントのみを監視していました。**multipathd** は、ファイバーチャネルファブリックが送信するリンク整合性イベントをリッスンし、それを使用してパスをマージナルとしてマークすることができました。この機能は、SCSI デバイス上のマルチパスデバイスでのみサポートされていました。この機能の使用の制限により、**multipathd** は Non-volatile Memory Express (NVMe) デバイスパスをマージナルとしてマークすることができませんでした。

この更新により、**multipathd** は、SCSI デバイスと NVMe デバイスの両方で FPIN-Li イベントの検出をサポートするようになりました。その結果、マルチパスは、他のパスが利用可能である間は、ファブリック接続が良好でないパスを使用しなくなりました。これにより、そのような状況での IO 遅延を回避できます。

[Jira:RHEL-6678](#)

max_retries オプションが multipath.conf の defaults セクションに追加されました

この機能拡張により、**multipath.conf** ファイルの **defaults** セクションに **max_retries** オプションが追加されました。デフォルトではこのオプションは設定されておらず、5 回の再試行を指定する SCSI レイヤーのデフォルト値が使用されます。このオプションの有効な値は **0** から **5** です。このオプションを設定すると、SCSI デバイスの **max_retries sysfs** 属性のデフォルト値がオーバーライドされます。この属性は、特定のエラータイプが発生した場合に、SCSI 層が失敗を返す前に I/O コマンドを再試行する回数を制御します。

マルチパスのパスチェッカーが成功を返すものの、デバイスへの I/O がハングしているという問題に遭遇した場合、このオプションを設定して、別のパスで I/O が再試行されるまでの時間を短縮できます。

[Jira:RHEL-1729^{\[1\]}](#)

auto_resize オプションが multipath.conf の defaults セクションに追加されました

以前は、マルチパスデバイスのサイズを変更するには、**multipathd resize map <name>** コマンドを手動で実行する必要がありました。この更新により、**multipath.conf** ファイルの **defaults** セクションに **auto_resize** オプションが追加されました。このオプションは、**multipathd** コマンドがマルチパスデバイスのサイズを自動的に変更できるタイミングを制御します。**auto_resize** のさまざまな値を以下に示します。

- デフォルトでは、**auto_resize** は **never** に設定されています。この場合、**multipathd** は変更なしで動作します。
- **auto_resize** が **grow_only** に設定されている場合、**multipathd** は、デバイスのパスのサイズが拡大した際にマルチパスデバイスのサイズを自動的に変更します。
- **auto_resize** が **grow_shrink** に設定されている場合、**multipathd** は、デバイスのパスのサイズが縮小した際にマルチパスデバイスを自動的に縮小します。

したがって、このオプションを有効にすると、マルチパスデバイスのサイズを手動で変更する必要がなくなります。

[Jira:RHEL-986^{\[1\]}](#)

Arcus NVMeoFC multipath.conf 設定の変更がカーネルに含まれるようになりました

Device-mapper-multipath に、HPE Alletra 9000 NVMeoFC アレイ用の設定が組み込まれるようになりました。Arcus は、NVMeoFC の ANA (非対称名前空間アクセス) のサポートを追加しました。これは SCSI の ALUA に似ています。RHEL ホストがこの機能を使用し、利用可能な場合は ANA 最適化パスにのみ I/O を送信するには、**multipath.conf** の変更が必要です。この変更なしでは、デバイスマッパーは ANA 最適化パスと ANA 非最適化パスの両方に I/O を送信していました。



注記

この変更は NVMeoFC のみに適用されます。FCP **multipath.conf** コンテンツには、この ALUA をサポートする設定が以前から含まれていました。

[Jira:RHEL-1830](#)

stratis-cli がバージョン 3.6.0 にリベース

stratis-cli パッケージがバージョン 3.6.0 にアップグレードされました。主なバグ修正と機能拡張は、以下のとおりです。

- **stratis-cli** コマンドラインインターフェイスは、ファイルシステムの作成時にファイルシステムのサイズ制限を設定するための追加オプションをサポートします。**set-size-limit** と **unset-size-limit** という 2 つの新しいファイルシステムコマンドは、ファイルシステムの作成後にファイルシステムのサイズ制限を設定または設定解除します。
- **stratis-cli** では、手動入力を使用してカーネルキーリングにキーを設定するときに、パスワード検証が組み込まれるようになりました。
- **stratis-cli** は、プールを停止するときに名前または UUID でプールを指定することをサポートするようになりました。
- **stratis-cli** の更新にはさまざまな内部改善も含まれ、パッケージ設定で少なくとも Python 3.9 バージョンの要件が強制されるようになりました。

[Jira:RHEL-2265^{\[1\]}](#)

boom がバージョン 1.6.0 にリベース

boom パッケージがバージョン 3.6.0 にアップグレードされました。主な機能拡張は、次のとおりです。

- **systemd** コマンドがサポートするマルチボリュームスナップショットブート構文のサポート。
- ブートエントリーでマウントする追加のボリュームを指定する、**new --mount** オプションおよび **--no-fstab** オプションの追加。

[Jira:RHEL-16813](#)

SAN からの NVMe-FC ブートが完全にサポートされるようになりました

Red Hat Enterprise Linux 9.2 でテクノロジープレビューとして導入された Non-volatile Memory Express (NVMe) over Fibre Channel (NVMe/FC) ブートが、完全にサポートされるようになりました。一部の NVMe/FC ホストバスアダプターは、NVMe/FC ブート機能をサポートします。NVMe/FC ブート機能を有効にするようにホストバスアダプター (HBA) をプログラミングする方法の詳細は、NVMe/FC ホストバスアダプターの製造元のドキュメントを参照してください。

Jira:RHEL-1492^[1]

4.10. 高可用性およびクラスター

pcs は時刻プロパティに関して ISO 8601 の duration 仕様をサポートします

pcs コマンドラインインターフェイスで、Pacemaker の時刻プロパティの値を ISO 8601 の duration 仕様標準に従って指定できるようになりました。

Jira:RHEL-7672

新しい pcsd Web UI 機能のサポート

pcsd Web UI では、次の機能がサポートされるようになりました。

- クラスターリソースの現在実行中のノードからの移動
- ノード上でのリソース実行の禁止
- クラスターステータスの経過時間とクラスターの状態がリロードされるタイミングを示す、クラスターステータスの表示
- クラスターステータス表示のリロードの要求

Jira:RHEL-7582, Jira:RHEL-7739

TLS 暗号リストがデフォルトでシステム全体の暗号化ポリシーを使用するようになりました

以前は、**pcsd** TLS 暗号リストはデフォルトで **DEFAULT:!RC4:!3DES:@STRENGTH** に設定されていました。この更新により、暗号リストはデフォルトでシステム全体の暗号化ポリシーによって定義されるようになりました。現在設定されている暗号化ポリシーによっては、このアップグレードにより **pcsd** デーモンが受け入れる TLS 暗号が変更される可能性があります。暗号化ポリシーフレームワークの詳細は、**crypto-policies**(7) の man ページを参照してください。

Jira:RHEL-7724

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

Python 3.12 が RHEL 9 で利用できるようになりました

RHEL 9.4 では、新しいパッケージ **python3.12** とそのためにビルドされた一連のパッケージ、および **ubi9/python-312** コンテナイメージによって提供される Python 3.12 が導入されています。

以前にリリースされた Python 3.11 と比較しての主な機能拡張は次のとおりです。

- Python では、ジェネリッククラスと関数に対して新しい **type** ステートメントと新しい型パラメーター構文が導入されています。
- フォーマットされた文字列リテラル (f-strings) が文法で形式化され、パーサーに直接統合できるようになりました。
- Python が、インタープリターごとに固有のグローバルインタープリターロック (GIL) を提供するようになりました。
- Python コードからバッファープロトコルを使用できるようになりました。

- セキュリティーを強化するために、SHA1、SHA3、SHA2-384、SHA2-512、および MD5 暗号化アルゴリズムの組み込みの **hashlib** 実装が、HACL* プロジェクトの正式に検証されたコードに置き換えられました。OpenSSL が組み込みの実装を提供しない場合は、フォールバックとして引き続き組み込みの実装を使用できます。
- **CPython** のディクショナリー、リスト、セットの内包表記がインライン化されました。これにより、内包表記の実行速度が大幅に向上します。
- **CPython** が Linux **perf** プロファイラーをサポートするようになりました。
- **CPython** が、サポート対象のプラットフォームでスタックオーバーフローの保護を提供するようになりました。

Python 3.12 とそのためにビルドされたパッケージは、同じシステム上に Python 3.9 および Python 3.11 と並行してインストールできます。

python3.12 スタック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。

```
# dnf install python3.12
# dnf install python3.12-pip
```

インタープリターを実行するには、たとえば、以下を使用します。

```
$ python3.12
$ python3.12 -m pip --help
```

詳細については、[Python のインストールと使用](#) を参照してください。

Python 3.12 のサポート期間については、[Red Hat Enterprise Linux Application Streams ライフサイクル](#) を参照してください。

[Jira:RHEL-14941](#)

メールアドレスの解析を制御するための Python の新しい環境変数

[CVE-2023-27043](#) の問題を軽減するために、メールアドレスをより厳密に解析するための後方互換性のない変更が Python 3 に導入されました。

この更新では、新しい **PYTHON_EMAIL_DISABLE_STRICT_ADDR_PARSING** 環境変数が導入されました。この変数を **true** に設定すると、比較的厳密ではない以前の解析動作が、システム全体のデフォルトになります。

```
export PYTHON_EMAIL_DISABLE_STRICT_ADDR_PARSING=true
```

ただし、該当する関数を個別に呼び出すと、より厳密な動作が有効になる可能性があります。

次の内容で **/etc/python/email.cfg** 設定ファイルを作成しても、同じ結果を得ることができます。

```
[email_addr_parsing]
PYTHON_EMAIL_DISABLE_STRICT_ADDR_PARSING = true
```

詳細は、ナレッジベース記事 [Mitigation of CVE-2023-27043 introducing stricter parsing of email addresses in Python](#) を参照してください。

[Jira:RHELDPCS-17369^{\[1\]}](#)

新しいモジュールストリーム: ruby:3.3

RHEL 9.4 では、新しい **ruby:3.3** モジュールストリームに Ruby 3.3.0 が導入されました。このバージョンでは、RHEL 9.1 で配布される **Ruby 3.1** に対するパフォーマンスの向上、バグおよびセキュリティ修正、および新機能がいくつか追加されました。

主な機能拡張は、次のとおりです。

- **Ripper** の代わりに新しい **Prism** パーサーを使用できます。**Prism** は、移植可能で、エラー耐性があり、メンテナンス性に優れた Ruby 言語の再帰下降パーサーです。
- Ruby の just-in-time (JIT) コンパイラ実装である YJIT は、もはや実験的機能ではなく、大幅なパフォーマンスの向上をもたらします。
- **Regexp** 一致アルゴリズムが改善され、潜在的な正規表現サービス拒否 (ReDoS) の脆弱性の影響が軽減されました。
- 新しい実験的な RJIT (純粋な Ruby の JIT) コンパイラーが MJIT を置き換えます。実稼働環境では YJIT を使用してください。
- 新しい M:N スレッドスケジューラーが利用可能になりました。

その他の主な変更点:

- 今後は、**Bison** の代わりに **Lrama** LALR パーサージェネレーターを使用する必要があります。
- いくつかの非推奨のメソッドと定数が削除されました。
- **Racc** gem はデフォルトの gem からバンドルされた gem に昇格しました。

ruby:3.3 モジュールストリームをインストールするには、次のコマンドを使用します。

```
# dnf module install ruby:3.3
```

以前の **ruby** モジュールストリームからアップグレードするには、[後続のストリームへの切り替え](#) を参照してください。

Ruby 3.3 のサポート期間については、[Red Hat Enterprise Linux Application Streams ライフサイクル](#) を参照してください。

Jira:RHEL-17089^[1]

新しいモジュールストリーム: php:8.2

RHEL 9.4 は、PHP 8.2 を新しい **php:8.2** モジュールストリームとして追加します。

このリリースでの改善点は次のとおりです。

- 読み取り専用クラス
- いくつかの新しいスタンドアロンタイプ
- 新しい **Random** エクステンション
- 特性の制約

php:8.2 モジュールストリームをインストールするには、次のコマンドを使用します。

-

```
# dnf module install php:8.2
```

php:8.1 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。

RHEL 9 での PHP の使用方法の詳細は、[Using the PHP scripting language](#) を参照してください。

php 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 Application Streams のライフサイクル](#) を参照してください。

Jira:RHEL-14699^[1]

perl-DateTime-TimeZone モジュールの **name()** メソッドは、タイムゾーン名を返すようになりました。

perl-DateTime-TimeZone モジュールがバージョン 2.62 に更新され、**name()** メソッドによって返される値がタイムゾーンエイリアスからメインのタイムゾーン名に変更されました。

詳細と例については、ナレッジベースの記事 [Change in the perl-DateTime-TimeZone API related to time zone name and alias](#) を参照してください。

Jira:RHEL-35685

新しいモジュールストリーム: nginx:1.24

nginx 1.24 Web およびプロキシサーバーが、**nginx:1.24** モジュールストリームとして利用できるようになりました。この更新では、以前にリリースされたバージョン 1.22 に対するバグ修正、セキュリティ修正、新機能、および機能拡張がいくつか提供されます。

Transport Layer Security (TLS) に関連する新機能と変更点:

- **ssl_session_cache** ディレクティブで共有メモリーを使用する場合、TLS セッションチケットの暗号鍵が自動的にローテーションされるようになりました。
- Secure Sockets Layer (SSL) プロキシを使用した設定でメモリー使用量が最適化されました。
- **resolver** ディレクティブの **ipv4=off** パラメーターを使用して、解決時に IPv4 アドレスの検索を無効にできるようになりました。
- nginx は、PROXY v2 TLV プロトコルに表示される Type-Length-Value (TLV) フィールドの値を格納する **\$proxy_protocol_tlv_*** 変数をサポートするようになりました。
- **ngx_http_gzip_static_module** モジュールはバイト範囲をサポートするようになりました。

その他の変更点:

- ヘッダー行は、内部 API でリンクされたリストとして表されるようになりました。
- nginx は、**ngx_http_perl_module** の **\$r->header_in()** メソッドと、**\$http_...**、**\$sent_http_...**、**\$sent_trailer_...**、**\$upstream_http_...**、および **\$upstream_trailer_...** 変数の検索中に、FastCGI、SCGI、および uwsgi バックエンドに渡される同一の名前のヘッダー文字列を連結するようになりました。
- nginx は、リスニングソケットのプロトコルパラメーターが再定義されると警告を表示するようになりました。

- nginx は、クライアントによってパイプラインが使用された場合、接続を遅延して閉じるようになりました。
- さまざまな SSL エラーのロギングレベルが、たとえば **Critical** から **Informational** に引き下げられました。

nginx:1.24 ストリームをインストールするには、以下を実行します。

```
# dnf module install nginx:1.24
```

nginx 1.22 ストリームからアップグレードするには、[後続のストリームに切り替えます](#)。

詳細については、[NGINX のセットアップと設定](#) を参照してください。

nginx モジュールストリームのサポート期間については、[Red Hat Enterprise Linux アプリケーションストリームのライフサイクル](#) を参照してください。

Jira:RHEL-14713^[1]

新しいモジュールストリーム: mariadb:10.11

MariaDB 10.11 が、新しいモジュールストリーム **mariadb:10.11** として利用できるようになりました。以前に利用可能であったバージョン 10.5 に対する主な機能拡張は、以下のとおりです。

- 新しい **sys_schema** 機能。
- アトミックデータ定義言語 (DDL) ステートメント。
- 新しい **GRANT ... TO PUBLIC** 特権。
- **SUPER** 特権と **READ ONLY ADMIN** 特権の分離。
- 新しい **UUID** データベースデータ型。
- Secure Socket Layer (SSL) プロトコルのバージョン 3 のサポート。MariaDB サーバーの起動には、正しく設定された SSL が必要になりました。
- **natural_sort_key()** 関数による自然なソート順序のサポート。
- 任意のテキストフォーマットのための新しい **SFORMAT** 関数。
- UTF-8 文字セットと UCA-14 照合の変更。
- **/usr/share/** ディレクトリーで利用可能な **systemd** ソケットのアクティベーションファイル。アップストリームとは異なり、これらのファイルは RHEL のデフォルト設定の一部ではないことに注意してください。
- **MySQL** の代わりに **MariaDB** 文字列を含むエラーメッセージ。
- 中国語で利用可能なエラーメッセージ。
- デフォルトの logrotate ファイルへの変更。
- MariaDB および MySQL クライアントの場合、コマンドラインで指定した接続プロパティ (例: **--port=3306**) によって、クライアントとサーバー間の通信のプロトコルタイプ (**tcp**、**socket**、**pipe**、**memory** など) が強制されるようになりました。

MariaDB 10.11 の変更点の詳細は、[MariaDB 10.5 と MariaDB 10.11 の主な違い](#) を参照してください。

MariaDB の詳細は、[MariaDB の使用](#) を参照してください。

mariadb:10.11 ストリームをインストールするには、以下を使用します。

```
# dnf module install mariadb:10.11
```

MariaDB 10.5 からアップグレードする場合は、[MariaDB 10.5 から MariaDB 10.11 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 Application Streams のライフサイクル](#) を参照してください。

[Jira:RHEL-3638](#)

新しいモジュールストリーム: **postgresql:16**

RHEL 9.4 では、PostgreSQL 16 が **postgresql:16** モジュールストリームとして導入されています。PostgreSQL 16 では、バージョン 15 に比べていくつかの新機能と機能拡張が提供されます。

主な機能拡張は、次のとおりです。

- 強化された一括ロードによりパフォーマンスが向上します。
- **libpq** ライブラリーが、接続レベルの負荷分散をサポートするようになりました。より効率的な負荷分散のために、新しい **load_balance_hosts** オプションを使用できます。
- カスタム設定ファイルを作成し、それを **pg_hba.conf** ファイルと **pg_ident.conf** ファイルに追加できるようになりました。
- PostgreSQL は、**pg_hba.conf** ファイル内のデータベースおよびロールエントリーに対する正規表現の一致をサポートするようになりました。

その他の変更点は次のとおりです。

- PostgreSQL は **postmaster** バイナリーとともに配布されなくなりました。提供されている **systemd** ユニットファイル (**systemctl start postgres** コマンド) を使用して **postgresql** サーバーを起動するユーザーは、この変更の影響を受けません。以前に **postmaster** バイナリーを介して **postgresql** サーバーを直接起動していた場合は、今後は代わりに **postgres** バイナリーを使用する必要があります。
- PostgreSQL は、パッケージで PDF 形式のドキュメントを提供しなくなりました。代わりに [オンラインドキュメント](#) を使用してください。

[PostgreSQL の使用](#) も参照してください。

postgresql:16 ストリームをインストールするには、次のコマンドを使用します。

```
# dnf module install postgresql:16
```

RHEL 9 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、[PostgreSQL の RHEL9 バージョンへの移行](#) 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については、[Red Hat Enterprise Linux アプリケーションストリームのライフサイクル](#) を参照してください。

Jira:RHEL-3635

Git がバージョン 2.43.0 にリベース

Git バージョン管理システムがバージョン 2.43.0 に更新され、以前にリリースされたバージョン 2.39 に対するバグ修正、機能拡張、およびパフォーマンス向上が実現しました。

主な機能拡張は、次のとおりです。

- **git check-attr** コマンドで新しい **--source** オプションを使用して、現在の作業ディレクトリーではなく、提供されたツリー状のオブジェクトから **.gitattributes** ファイルを読み取ることができるようになりました。
- Git は、**WWW-Authenticate** レスポンスタイプヘッダーからの情報を認証情報ヘルパーに渡すことができるようになりました。
- コミットが空の場合、**git format-patch** コマンドは空のファイルを作成する代わりに、コミットのヘッダーを含む出力ファイルを書き込むようになりました。
- **git blame --contents=<file> <revision> -- <path>** コマンドを使用して、**<file>** の内容を起点として **<revision>** に至る履歴を通じて、行の起源を検索できるようになりました。
- **git log --format** コマンドは、**--decorate** オプションが提供する機能を拡張するためのさらなるカスタマイズ用に、**%(decorate)** プレースホルダーを受け入れるようになりました。

Jira:RHEL-17100^[1]

Git LFS がバージョン 3.4.1 にリベース

Git Large File Storage (LFS) エクステンションがバージョン 3.4.1 に更新されました。これにより、以前にリリースされたバージョン 3.2.0 に対するバグ修正、機能拡張、およびパフォーマンス向上が実現しました。

主な変更点は、以下のとおりです。

- **git lfs push** コマンドが、標準入力から参照とオブジェクト ID を読み取ることができるようになりました。
- Git LFS は、Git に依存せずに代替リモートを処理するようになりました。
- Git LFS は、認証情報ヘルパーとして **WWW-Authenticate** レスポンスタイプヘッダーをサポートするようになりました。

Jira:RHEL-17101^[1]

4.12. コンパイラーおよび開発ツール

LLVM Toolset がバージョン 17.0.6 にリベース

LLVM Toolset がバージョン 17.0.6 に更新されました。

主な機能拡張は、次のとおりです。

- 不透明ポインターの移行が完了しました。
- ミドルエンド最適化におけるレガシーパスマネージャーのサポートが削除されました。

Clang の変更:

- C++20 コルーチンは実験的なものとはみなされなくなりました。
- 最適化されていないビルドでの **std::move** 関数などのコード生成が改善されました。

詳細は、LLVM および Clang のアップストリームリリースノートを参照してください。

[Jira:RHEL-9283](#)

Rust Toolset がバージョン 1.75.0 にリベース

Rust Toolset がバージョン 1.75.0 に更新されました。

主な機能拡張は、次のとおりです。

- 定数評価時間が無制限になる
- よりクリーンなパニックメッセージ
- Cargo レジストリー認証
- **async fn** と特性内の不透明な戻り値のタイプ

[Jira:RHEL-12963](#)

Go Toolset がバージョン 1.21.0 にリベース

Go Toolset がバージョン 1.21.0 に更新されました。

主な機能拡張は、次のとおりです。

- **min**、**max**、**clear** ビルトインが追加されました。
- プロファイルガイド最適化の公式サポートが追加されました。
- パッケージの初期化順序がより正確に定義されるようになりました。
- 型推論が改善されました。
- 下位互換性のサポートが改善されました。

詳細は、アップストリームの Go リリースノートを参照してください。

[Jira:RHEL-11871](#)^[1]

Clang リソースディレクトリーが移動しました

Clang が内部ヘッダーとライブラリーを保存する Clang リソースディレクトリーが、`/usr/lib64/clang/17` から `/usr/lib/clang/17` に移動しました。

[Jira:RHEL-9346](#)

elfutils がバージョン 0.190 にリベース

elfutils パッケージが、バージョン 0.190 に更新されました。以下は、主な改善点です。

- **libelf** ライブラリーが、相対再配置 (REL R) をサポートするようになりました。

- **libdw** ライブラリーが、**.debug_[ct]u_index** セクションを認識するようになりました。
- **eu-readelf** ユーティリティーが、ELF セクションを使用せずに動的セグメントを使用してシンボルを表示する、新しい **-Ds, --use-dynamic --symbol** オプションをサポートするようになりました。
- **eu-readelf** ユーティリティーが、**.gdb_index** バージョン 9 を表示できるようになりました。
- 新しい **eu-scrlines** ユーティリティーは、指定された DWARF または ELF ファイルに関連付けられたソースファイルのリストをコンパイルします。
- **debuginfod** サーバースキーマが変更され、ファイル名の表現が 60% 圧縮されました (インデックスの再作成が必要です)。

[Jira:RHEL-12489](#)

systemtap がバージョン 5.0 にリベース

systemtap パッケージが、バージョン 5.0 に更新されました。主な機能拡張は、次のとおりです。

- カーネルとユーザー間のトランスポートの速度および信頼性の向上。
- DWARF5 debuginfo 形式のサポートの拡張。

[Jira:RHEL-12488](#)

GCC Toolset 13 の更新

GCC Toolset 13 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。これは、AppStream リポジトリ内の Software Collection の形式で Application Stream として利用できます。

RHEL 9.4 で導入された注目すべき変更点は次のとおりです。

- GCC コンパイラーがバージョン 13.2.1 に更新され、アップストリーム GCC で利用可能な多くのバグ修正と機能拡張が提供されます。
- **binutils** は、**-march=znver5** コンパイラースイッチを通じて、**znver5** コアに基づく AMD CPU をサポートするようになりました。
- **annobin** がバージョン 12.32 に更新されました。
- GCC の **annobin** プラグインは、オブジェクトファイルに保存するメモに対して、より圧縮された形式をデフォルトで使用するようになりました。その結果、特に大規模で複雑なプログラムでは、オブジェクトファイルが小さくなり、リンク時間が短縮されます。

GCC Toolset 13 では、次のツールとバージョンが提供されています: GCC:: 13.2.1 GDB:: 12.1 binutils:: 2.40 dwz:: 0.14 annobin:: 12.32

GCC Toolset 13 をインストールするには、**root** として次のコマンドを実行します。

```
# dnf install gcc-toolset-13
```

GCC Toolset 13 からツールを実行するには、以下を使用します。

```
$ scl enable gcc-toolset-13 tool
```

GCC Toolset 13 のツールバージョンがこれらのツールのシステムバージョン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、以下を使用します。

```
$ scl enable gcc-toolset-13 bash
```

詳細は、[GCC Toolset 13](#) を参照してください。

Jira:RHEL-23798^[1]

GCC と `-fstack-protector` フラグを使用してコンパイルすると、64 ビット ARM での動的スタック割り当ての保護に失敗しなくなる

以前は、64 ビット ARM アーキテクチャーでは、`-fstack-protector` フラグを指定したシステム GCC コンパイラーは、C99 可変長配列または `alloca()` 割り当てオブジェクトを含む関数でバッファオーバーフローを検出できませんでした。その結果、攻撃者はスタック上に保存されたレジスターを上書きする可能性があります。この更新により、64 ビット ARM でのバッファオーバーフロー検出が修正されました。その結果、システム GCC でコンパイルされたアプリケーションはより安全になります。

Jira:RHEL-17638^[1]

GCC ツールセット 13: GCC と `-fstack-protector` フラグを使用してコンパイルすると、64 ビット ARM での動的スタック割り当ての保護に失敗しなくなる

以前は、64 ビット ARM アーキテクチャーでは、`-fstack-protector` フラグを指定した GCC コンパイラーは、C99 可変長配列または `alloca()` 割り当てオブジェクトを含む関数でバッファオーバーフローを検出できませんでした。その結果、攻撃者はスタック上に保存されたレジスターを上書きする可能性があります。この更新により、64 ビット ARM でのバッファオーバーフロー検出が修正されました。その結果、GCC でコンパイルされたアプリケーションはより安全になります。

Jira:RHEL-16998

pcp がバージョン 6.2.0 に更新される

pcp パッケージがバージョン 6.2.0 に更新されました。以下は、主な改善点です。

- `pcp-htop` は、ユーザー定義のタブをサポートするようになりました。
- `pcp-atop` は、新しい棒グラフ視覚化モードをサポートするようになりました。
- OpenMetrics PMDA メトリクスラベルとロギングが改善されました。
- 追加の Linux カーネル仮想メモリーメトリクスが追加されました。
- 新しいツール:
 - `pmlogredact`
 - `pcp-buddyinfo`
 - `pcp-meminfo`
 - `pcp-netstat`
 - `pcp-slabinfo`
 - `pcp-zoneinfo`

[Jira:RHEL-2317^{\[1\]}](#)

新しい grafana-selinux パッケージ

以前は、**grafana-server** のデフォルトのインストールは **unconfined_service_t** SELinux タイプとして実行されていました。この更新では、**grafana-server** 用の SELinux ポリシーが含まれ、**grafana-server** とともにデフォルトでインストールされる、新しい **grafana-selinux** パッケージが追加されます。その結果、**grafana-server** は **grafana_t** SELinux タイプとして実行されるようになりました。

[Jira:RHEL-7505](#)

papi は、新しいプロセッサマイクロアーキテクチャーをサポートする

この機能拡張により、次のプロセッサマイクロアーキテクチャー上の **papi** イベントプリセットを使用して、パフォーマンス監視ハードウェアにアクセスできるようになります。

- AMD Zen 4
- 第 4 世代 Intel® Xeon® スケーラブルプロセッサ

[Jira:RHEL-9333^{\[1\]}](#)、[Jira:RHEL-9335](#)、[Jira:RHEL-9334](#)

新しいパッケージ: maven-openjdk21

maven:3.8 モジュールストリームに、**maven-openjdk21** サブパッケージが含まれるようになりました。このサブパッケージは、OpenJDK 21 用の Maven JDK バインディングを提供し、システム OpenJDK 21 を使用するように Maven を設定します。

[Jira:RHEL-13046^{\[1\]}](#)

新しいパッケージ: libzip-tools

RHEL 9.4 では、**zipcmp**、**zipmerge**、**ziptool** などのユーティリティを提供する **libzip-tools** パッケージが導入されています。

[Jira:RHEL-17567](#)

cmake がバージョン 3.26 にリベース

cmake パッケージがバージョン 3.26 に更新されました。以下は、主な改善点です。

- C17 および C18 の言語規格のサポートが追加されました。
- **cmake** は、**/etc/os-release** ファイルでオペレーティングシステムの識別情報をクエリーできるようになりました。
- CUDA 20 および **nvtx3** ライブラリーのサポートが追加されました。
- Python 安定アプリケーションバイナリーインターフェイスのサポートが追加されました。
- Simplified Wrapper and Interface Generator (SWIG) ツールに Perl 5 のサポートが追加されました。

[Jira:RHEL-7393](#)

valgrind が 3.22 に更新される

valgrind パッケージがバージョン 3.22 に更新されました。以下は、主な改善点です。

- **valgrind memcheck** は、C 関数 **memalign**、**posix_memalign**、および **aligned_alloc** に指定された値と、C++17 で調整された **new** Operator が有効なアラインメント値であるかどうかをチェックするようになりました。
- **valgrind memcheck** は、C++14 サイズおよび C++17 に揃えられた **new** および **delete** Operator の不一致検出をサポートするようになりました。
- DWARF デバッグ情報の遅延読み取りのサポートが追加され、**debuginfo** パッケージがインストールされている場合に起動が高速化されました。

[Jira:RHEL-12490](#)

libbigail がバージョン 2.4 にリベース

libbigail パッケージがバージョン 2.4 に更新されました。

主な機能拡張は、次のとおりです。

- **abidiff** ツールが 2 セットのバイナリーの比較をサポートするようになりました。
- 柔軟な配列データメンバーに関連する無害な変更レポートを抑制するためのサポートが追加されました。
- **enum** 型に関する無害な変更レポートを抑制するためのサポートが改善されました。
- 匿名の **enum** 型、**union** 型、**struct** 型への変更の表現が改善されました。

[Jira:RHEL-12491](#)

4.13. IDENTITY MANAGEMENT

SSSD で新しいパスワードレスの認証方法が利用可能になる

この更新により、SSSD でパスワードレス認証を有効にして設定し、FIDO2 仕様と互換性のある生体認証デバイス (YubiKey など) を使用できるようになりました。事前に FIDO2 トークンを登録し、この登録情報を RHEL IdM、Active Directory、または LDAP ストアのユーザーアカウントに保存する必要があります。RHEL は、**libfido2** ライブラリーを使用して FIDO2 の互換性を実装していますが、現在は USB ベースのトークンのみをサポートしています。

[Jira:RHELDPCS-17841^{\[1\]}](#)

ansible-freeipa ipauser および ipagroup モジュールが、新しい renamed 状態をサポートするようになる

この更新により、**ansible-freeipa ipauser** モジュールの **renamed** 状態を使用して、既存の IdM ユーザーのユーザー名を変更できるようになりました。また、**ansible-freeipa ipagroup** モジュールでこの状態を使用して、既存の IdM グループのグループ名を変更することもできます。

[Jira:RHEL-4962](#)

Identity Management ユーザーは、外部アイデンティティプロバイダーを使用して IdM に認証できるようになる

この機能拡張により、Identity Management (IdM) ユーザーを、OAuth 2 デバイス認証フローをサポートする外部アイデンティティプロバイダー (IdP) に関連付けることができるようになりました。このような IdP の例には、Red Hat build of Keycloak、Microsoft Entra ID (以前の Azure Active Directory)、GitHub、および Google が含まれます。

IdM に IdP 参照と関連付けられた IdP ユーザー ID が存在する場合は、それらを使用して IdM ユーザーが外部 IdP で認証できるようにすることができます。外部 IdP で認証と認可を実行した後、IdM ユーザーはシングルサインオン機能を備えた Kerberos チケットを受け取ります。ユーザーは、RHEL 9.1 以降で使用可能な SSSD バージョンで認証する必要があります。

Jira:RHELPLAN-169666^[1]

ipa がバージョン 4.11 にリベース

ipa パッケージがバージョン 4.10 から 4.11 に更新されました。主な変更点は、以下のとおりです。

- FIDO2 ベースのパスキーのサポート。
- Kerberos サービス用のリソースベースの制約付き委任 (RBCD) の初期実装。
- **ipalib.api** を自動的に設定、接続、切断するためのコンテキストマネージャー。
- IdM レプリカのインストールが、Kerberos 認証だけでなく、すべての IPA API および CA 要求に関しても、選択したサーバーに対して実行されるようになりました。
- **ansible-freeipa** パッケージが、バージョン 1.11 から 1.12.1 にリベースされました。
- **ipa-healthcheck** パッケージが、バージョン 0.12 から 0.16 にリベースされました。

詳細は、[アップストリームのリリースノート](#) を参照してください。

Jira:RHEL-11652

期限切れの KCM Kerberos チケットの削除

以前は、Kerberos Credential Manager (KCM) に新しい認証情報を追加しようとした際にすでにストレージ領域の上限に達していた場合、新しい認証情報は拒否されていました。ユーザーのストレージ領域は、**max_uid_ccaches** 設定オプション (デフォルト値は 64) によって制限されます。この更新により、ストレージ領域の上限にすでに達している場合は、最も古い期限切れの認証情報が削除され、新しい認証情報が KCM に追加されます。期限切れの認証情報がない場合、操作は失敗し、エラーが返されます。この問題を防ぐには、**kdestroy** コマンドを使用して認証情報を削除し、領域を解放します。

Jira:SSSD-6216

IdM は Ansible モジュール **idoverrideuser**、**idoverridegroup**、**idview** をサポートするようになりました

この更新により、**ansible-freeipa** パッケージに次のモジュールが含まれるようになりました。

idoverrideuser

Identity Management (IdM) LDAP サーバーに保存されているユーザーのユーザー属性 (ユーザーのログイン名、ホームディレクトリー、証明書、SSH キーなど) をオーバーライドできます。

idoverridegroup

IdM LDAP サーバーに保存されているグループの属性 (グループ名、GID、説明など) をオーバーライドできます。

idview

ユーザーおよびグループ ID のオーバーライドを整理し、特定の IdM ホストに適用できます。

将来的には、これらのモジュールを使用して、AD ユーザーがスマートカードを使用して IdM にログインできるようになります。

[Jira:RHEL-16934](#)

idp Ansible モジュールを使用すると、IdM ユーザーを外部 IdP に関連付けることができます

この更新により、**idp ansible-freeipa** モジュールを使用して、Identity Management (IdM) ユーザーを、OAuth 2 デバイス認可フローをサポートする外部アイデンティティプロバイダー (IdP) に関連付けることができます。IdM に IdP 参照および関連付けられた IdP ユーザー ID が存在する場合は、それらを使用して IdM ユーザーの IdP 認証を有効にすることができます。

外部 IdP で認証と認可を実行した後、IdM ユーザーはシングルサインオン機能を備えた Kerberos チケットを受け取ります。ユーザーは、RHEL 8.7 以降で使用可能な SSSD バージョンで認証する必要があります。

[Jira:RHEL-16939](#)

getcert add-ca は、証明書がすでに存在するか追跡されている場合に新しい戻りコードを返します

この更新により、すでに存在するか追跡されている証明書を追加または追跡しようとすると、**getcert** コマンドは特定の戻りコード **2** を返します。以前は、エラー状態が発生すると、このコマンドは戻りコード **1** を返していました。

[Jira:RHEL-22302](#)

DNS ゾーン管理の委譲が ansible-freeipa で有効になりました

dnszone ansible-freeipa モジュールを使用して、DNS ゾーン管理を委譲できるようになりました。**dnszone** モジュールの **permission** または **managedby** 変数を使用して、ゾーンごとのアクセス委譲権限を設定します。

[Jira:RHEL-19134](#)

すべての LDAP クライアントに OTP の使用を強制する

[RHBA-2024:2558](#) アドバイザリーのリリースにより、RHEL IdM では、2 要素 (OTP) 認証が設定されたユーザーアカウントの LDAP サーバー認証におけるデフォルトの動作を設定できるようになりました。OTP が強制されている場合、LDAP クライアントは、関連付けられた OTP トークンを持つユーザーに対して、単一要素認証 (パスワード) を使用して LDAP サーバーに対して認証することはできません。この方法は、データなしの OID 2.16.840.1.113730.3.8.10.7 を使用した特別な LDAP 制御を使用することで、Kerberos バックエンドを通じてすでに実施されています。

- すべての LDAP クライアントに OTP の使用を強制するには、管理者は次のコマンドを使用できます。

```
$ ipa config-mod --addattr ipaconfigstring=EnforceLDAPOTP
```

- すべての LDAP クライアントに対して以前の OTP 動作に戻すには、次のコマンドを使用します。

```
$ ipa config-mod --delattr ipaconfigstring=EnforceLDAPOTP
```

[Jira:RHEL-23377^{\[1\]}](#)

runasuser_group パラメーターが ansible-freeipa ipasudorule で利用できるようになる

この更新では、**ansible-freeipa ipasudorule** モジュールを使用して、**sudo** ルールの RunAs ユーザーのグループを設定できるようになりました。このオプションは、Identity Management (IdM) コマンドラインインターフェイスと IdM Web UI ですでに利用可能です。

[Jira:RHEL-19130](#)

389-ds-base がバージョン 2.4.5 にリベース

389-ds-base パッケージがバージョン 2.4.5 に更新されました。バージョン 2.3.4 への主なバグ修正および機能拡張は、以下のとおりです。

- <https://www.port389.org/docs/389ds/releases/release-2-3-5.html>
- <https://www.port389.org/docs/389ds/releases/release-2-3-6.html>
- <https://www.port389.org/docs/389ds/releases/release-2-3-7.html>
- <https://www.port389.org/docs/389ds/releases/release-2-4-0.html>
- <https://www.port389.org/docs/389ds/releases/release-2-4-1.html>
- <https://www.port389.org/docs/389ds/releases/release-2-4-2.html>
- <https://www.port389.org/docs/389ds/releases/release-2-4-3.html>
- <https://www.port389.org/docs/389ds/releases/release-2-4-4.html>
- <https://www.port389.org/docs/389ds/releases/release-2-4-5.html>

[Jira:RHEL-15907](#)

ns-slapd プロセスでは、Transparent Huge Pages がデフォルトで無効になりました

大規模なデータベースキャッシュを使用すると、Transparent Huge Pages (THP) によって、高い負荷がかかった状態の Directory Server のパフォーマンスに悪影響が生じる可能性があります。たとえば、メモリーフットプリントの増加、CPU 使用率の上昇、レイテンシースパイクなどが挙げられます。この機能拡張により、**dirsrv systemd** ユニットの `/usr/lib/systemd/system/dirsrv@.service.d/custom.conf` ドロップイン設定ファイルに、**ns-slapd** プロセスの THP を無効にする新しい設定オプション **THP_DISABLE=1** が追加されました。

さらに、Directory Server ヘルスチェックツールが THP 設定を検出するようになりました。システム全体および Directory Server インスタンスに対して THP を有効にした場合、ヘルスチェックツールは有効になっている THP について通知し、無効にする方法に関する推奨事項を出力します。

[Jira:RHEL-5142](#)

新しい lastLoginHistSize 設定属性が Account Policy プラグインで利用できるようになりました

以前は、ユーザーがバインドを正常に実行した場合、最後のログインの時刻しか利用可能できませんでした。この更新では、新しい **lastLoginHistSize** 設定属性を使用して、成功したログインの履歴を管理できます。デフォルトでは、成功したログインのうち最後の 5 回が保存されます。

lastLoginHistSize 属性で成功したログインの統計情報を収集するには、Account Policy プラグインの **alwaysRecordLogin** 属性を有効にする必要があります。

詳細は、[lastLoginHistSize](#) を参照してください。

Jira:RHEL-5133^[1]

MFA バインドを識別する、アクセスログ内の新しい **notes=M** メッセージ

この更新により、MFA プラグインなどの事前バインド認証プラグインを使用してユーザーアカウントの二要素認証を設定すると、**BIND** 操作中に、Directory Server ログファイルに次のメッセージが記録されるようになりました。

- アクセスログには、新しい **notes=M** note メッセージが記録されます。

```
[time_stamp] conn=1 op=0 BIND dn="uid=jdoe,ou=people,dc=example,dc=com" method=128
version=3
[time_stamp] conn=1 op=0 RESULT err=0 tag=97 nentries=0 wtime=0.000111632
optime=0.006612223 etime=0.006722325 notes=M details="Multi-factor Authentication"
dn="uid=jdoe,ou=people,dc=example,dc=com"
```

- セキュリティーログには、新しい **SIMPLE/MFA** バインドメソッドが記録されます。

```
{ "date": "[time_stamp] ", "utc_time": "1709327649.232748932", "event": "BIND_SUCCESS",
  "dn": "uid=djoe,ou=people,dc=example,dc=com", "bind_method": "SIMPLE/MFA",
  "root_dn": false, "client_ip": "::1", "server_ip": "::1", "ldap_version": 3, "conn_id": 1, "op_id": 0,
  "msg": "" }
```

アクセスログとセキュリティーログにこのようなメッセージを記録するには、バインドが事前バインド認証プラグインの一部である場合、事前バインド認証プラグインで SLAPI API を使用してフラグを設定する必要があります。

Jira:RHELDOCS-17838^[1]

新しい **inchainMatch** マッチングルールが利用可能になる

この更新により、クライアントアプリケーションは新しい **inchainMatch** マッチングルールを使用して、LDAP エントリーの祖先を検索できるようになります。**inchainMatch** マッチングルールでは、**member**、**manager**、**parentOrganization**、**memberof** 属性が使用可能であり、次の検索を実行できます。

- ユーザーがメンバーになっているすべての直接または間接のグループを検索します。
- 特定のユーザーがマネージャーであるすべての直接または間接ユーザーを検索します。
- エントリーが属するすべての直接または間接的な組織を検索します。
- 特定のグループのすべての直接または間接メンバーを検索します。

パフォーマンス上の理由から、クライアントアプリケーションが **inchainMatch** マッチングルールを使用してこれらの属性に対する検索を実行する場合は、**member**、**manager**、**parentOrganization**、および **memberof** 属性をインデックス化する必要があることに注意してください。

Directory Server は、デフォルトで有効になっている In Chain プラグインを使用して、**inchainMatch** マッチングルールを実装します。ただし、**inchainMatch** の計算にはコストがかかるため、アクセス制御命令 (ACI) によってマッチングルールの使用が制限されます。

詳細は、[inchainMatch マッチングルールを使用して LDAP エントリーの祖先を検索する](#) を参照してください。

Jira:RHELDOCS-17256^[1]

HAProxy プロトコルが 389-ds-base パッケージでサポートされるようになりました

以前は、Directory Server はプロキシクライアントと非プロキシクライアント間の着信接続を区別していませんでした。この更新により、新しい多値設定属性 **nsslapd-haproxy-trusted-ip** を使用して、信頼できるプロキシサーバーのリストを設定できるようになりました。**cn=config** エントリーの下に **nsslapd-haproxy-trusted-ip** が設定されている場合、Directory Server は HAProxy プロトコルを使用して追加の TCP ヘッダー経由でクライアント IP アドレスを受信します。これにより、アクセス制御命令 (ACI) を正しく評価し、クライアントトラフィックをログに記録できるようになります。

信頼されていないプロキシサーバーがバインド要求を開始した場合、Directory Server は要求を拒否し、エラーログファイルに次のメッセージを記録します。

```
[time_stamp] conn=5 op=-1 fd=64 Disconnect - Protocol error - Unknown Proxy - P4
```

詳細は、[nsslapd-haproxy-trusted-ip](#) を参照してください。

[Jira:RHEL-5130](#)

samba がバージョン 4.19.4 にリベース

samba パッケージはアップストリームバージョン 4.19.4 にアップグレードされ、以前のバージョンに対するバグ修正と機能拡張が提供されています。主な変更点は以下のとおりです。

- 一貫したユーザーエクスペリエンスを実現するために、**smbget** ユーティリティーのコマンドラインオプションは名前が変更され、削除されました。ただし、これにより、ユーティリティーを使用する既存のスクリプトまたはジョブが破損する可能性があります。新しいオプションの詳細は、**smbget --help** コマンドと **smbget(1)** の man ページを参照してください。
- **winbind debug traceid** オプションが有効になっている場合、**winbind** サービスは次のフィールドもログに記録するようになりました。
 - **traceid**: 同じリクエストに属するレコードを追跡します。
 - **depth**: リクエストのネストレベルを追跡します。
- Samba は独自の暗号化実装を使用しなくなり、代わりに GnuTLS ライブラリーが提供する暗号化機能を全面的に使用するようになりました。
- **directory name cache size** オプションが削除されました。

Samba 4.11 以降はサーバーメッセージブロックバージョン 1 (SMB1) プロトコルが非推奨となり、今後のリリースで削除されることに注意してください。

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

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

[Jira:RHEL-16476](#)

Identity Management API が完全にサポートされるようになりました

Identity Management (IdM) API は、RHEL 9.2 でテクノロジープレビューとして利用可能でした。RHEL 9.3 以降では完全にサポートされています。

IdM API が拡張されて API コマンドの複数のバージョンが有効になった場合でも、ユーザーは既存のツールとスクリプトを使用できます。これらの機能拡張により、コマンドの動作が互換性のない方法で変更されることはありません。これには次の利点があります。

- 管理者は、管理しているクライアントではなくサーバー上で、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。

たとえば、一方が機能の新しいオプションを導入した新しいバージョンを使用している場合でも、サーバーとの通信は可能です。

注記

IdM API は JSON-RPC インターフェイスを提供しますが、このタイプのアクセスはサポートされていません。Red Hat では、代わりに Python を使用して API にアクセスすることを推奨します。Python を使用すると、サーバーからのメタデータの取得などの重要な部分が自動化され、使用可能なすべてのコマンドをリスト表示できるようになります。

[Bugzilla:1513934](#)

4.14. WEB コンソール

RHEL Web コンソールで、Ansible およびシェルスクリプトを生成できるようになりました

Web コンソールで、**kdump** 設定ページの自動化スクリプトに簡単にアクセスしてコピーできるようになりました。生成されたスクリプトを使用して、特定の **kdump** 設定を複数のシステムに実装できます。

[Jira:RHELDOCS-17060^{\[1\]}](#)

Storage でのストレージ管理およびパーティションサイズ変更の簡素化

Web コンソールの Storage セクションが再設計されました。新しいデザインにより、すべてのビューの視認性が向上しました。概要ページでは、すべてのストレージオブジェクトが包括的なテーブルに表示されるようになり、直接操作を実行しやすくなります。任意の行をクリックすると、詳細情報と追加のアクションが表示されます。さらに、Storage セクションからパーティションのサイズを変更できるようになりました。

[Jira:RHELDOCS-17056^{\[1\]}](#)

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

ad_integration RHEL システムロールは、動的 DNS 更新オプションの設定をサポートするようになりました

この更新により、**ad_integration** RHEL システムロールは、Active Directory (AD) との統合時に、SSSD を使用した動的 DNS 更新の設定オプションをサポートするようになりました。デフォルトでは、SSSD は以下の場合に DNS レコードの自動更新を試みます。

- アイデンティティプロバイダーがオンラインになるとき (常に)
- 指定した間隔 (任意の設定)。デフォルトでは、AD プロバイダーは 24 時間ごとに DNS レコードを更新します。

ad_integration の新しい変数を使用して、これらの設定やその他の設定を変更できます。たとえ

ば、`ad_dyndns_refresh_interval` を **172800** に設定すれば、DNS レコードの更新間隔を 48 時間に変更できます。ロール変数の詳細は、`/usr/share/doc/rhel-system-roles/ad_integration/` ディレクトリー内のリソースを参照してください。

[Jira:RHELDPCS-17372](#)^[1]

Storage RHEL システムロールが共有 LVM デバイス管理をサポートするようになりました

RHEL システムロールが、共有論理ボリュームとボリュームグループの作成と管理をサポートするようになりました。

[Jira:RHEL-1535](#)

Microsoft SQL Server 2022 が RHEL 9 で利用可能になる

`mssql-server` システムロールが RHEL 9 で利用できるようになりました。このロールは 2 つの変数を追加します。

1. `mssql_run_selinux_confined` は、SQL Server を制限されたアプリケーションとして実行するかどうかを制御します。`true` に設定すると、ロールは `mssql-server-selinux` パッケージをインストールします。`false` に設定すると、ロールは `mssql-server-selinux` パッケージを削除します。デフォルト設定は、RHEL 9 マネージドノードの場合は `true`、その他のマネージドノードの場合は `false` です。
2. `mssql_manage_selinux` は、SELinux を設定するかどうかを制御します。`true` に設定すると、変数は `mssql_run_selinux_confined` 変数の値に基づいて、Enforcing モードまたは Permissive モードを設定します。

[Jira:RHEL-16342](#)

rhc システムロールが RHEL 7 システムをサポートするようになる

`rhc` システムロールを使用して RHEL 7 システムを管理できるようになりました。RHEL 7 システムを Red Hat Subscription Management (RHSM) および Insights に登録し、`rhc` システムロールを使用してシステムの管理を開始します。

現在 RHEL 7 では Insights Remediation 機能を利用できないため、`rhc_insights.remediation` パラメーターを使用しても RHEL 7 システムには影響しません。

[Jira:RHEL-16976](#)

fapolicyd を設定するための新しい RHEL システムロール

新しい `fapolicyd` RHEL システムロールを使用すると、Ansible Playbook を使用して `fapolicyd` フレームワークを管理および設定できます。`fapolicyd` ソフトウェアフレームワークは、ユーザー定義のポリシーに基づいてアプリケーションの実行を制御します。

[Jira:RHEL-16542](#)

RHEL システムロールが LVM スナップショット管理をサポートするようになる

この機能拡張により、新しい `snapshot` RHEL システムロールを使用して、LVM スナップショットを作成、設定、管理できます。

[Jira:RHEL-16552](#)

Nmstate API と `network` RHEL システムロールが新しいルートタイプをサポートするようになりました

この機能拡張により、Nmstate API と **network** RHEL システムロールで次のルートタイプを使用できるようになりました。

- **blackhole**
- **prohibit**
- **unreachable**

[Jira:RHEL-19579^{\[1\]}](#)

ad_integration RHEL システムロールは、カスタム SSSD ドメイン設定をサポートするようになりました。

以前は、**ad_integration** RHEL システムロールを使用する場合、ロールを使用して **sssd.conf** ファイルのドメイン設定セクションにカスタム設定を追加することはできませんでした。この機能拡張により、**ad_integration** ロールは **sssd.conf** ファイルを変更できるようになり、結果としてカスタム SSSD 設定を使用できるようになります。

[Jira:RHEL-17668](#)

ad_integration RHEL システムロールがカスタム SSSD 設定をサポートするようになる

以前は、**ad_integration** RHEL システムロールを使用する場合、ロールを使用して **sssd.conf** ファイルの **[sssd]** セクションにカスタム設定を追加することはできませんでした。この機能拡張により、**ad_integration** ロールは **sssd.conf** ファイルを変更できるようになり、結果としてカスタム SSSD 設定を使用できるようになります。

[Jira:RHEL-21133](#)

rhc ロールに、表示名を設定するための新しい **rhc_insights.display_name** オプションが追加されました

新しい **rhc_insights.display_name** パラメーターを使用して、Red Hat Insights に登録されているシステムの表示名を設定または更新できるようになりました。このパラメーターを使用すると、希望に応じてシステムに名前を付けて、Insights Inventory 内のシステムを簡単に管理することができます。システムがすでに Red Hat Insights に接続されている場合は、パラメーターを使用して既存の表示名を更新します。登録時に表示名が明示的に設定されていない場合、表示名はデフォルトでホスト名に設定されます。表示名をホスト名に自動的に戻すことはできませんが、手動でホスト名に設定することは可能です。

[Jira:RHEL-16964](#)

fapolicyd を設定するための新しい RHEL システムロール

新しい **fapolicyd** RHEL システムロールを使用すると、Ansible Playbook を使用して **fapolicyd** フレームワークを管理および設定できます。**fapolicyd** ソフトウェアフレームワークは、ユーザー定義のポリシーに基づいてアプリケーションの実行を制御します。

[Jira:RHEL-16541](#)

logging RHEL システムロール用の新しい **logging_preserve_fqdn** 変数

以前は、**logging** システムロールを使用して完全修飾ドメイン名 (FQDN) を設定することはできませんでした。この更新により、オプションの **logging_preserve_fqdn** 変数が追加されています。これを使用すると、**rsyslog** の **preserveFQDN** 設定オプションを設定して、syslog エントリーで短い名前ではなく完全な FQDN を使用できます。

[Jira:RHEL-15932](#)

logging ロールは、出力モジュールの一般的なキューと一般的なアクションパラメーターをサポートします。

以前は、**logging** ロールで一般的なキューパラメーターと一般的なアクションパラメーターを設定することはできませんでした。この更新により、**logging** RHEL システムロールは、出力モジュールの一般的なキューパラメーターと一般的なアクションパラメーターの設定をサポートするようになりました。

[Jira:RHEL-15439](#)

postgresql RHEL システムロールが PostgreSQL 16 をサポートするようになりました

PostgreSQL サーバーをインストール、設定、管理、起動する **postgresql** RHEL システムロールが、PostgreSQL 16 をサポートするようになりました。

このシステムロールの詳細は、[postgresql RHEL システムロールを使用した PostgreSQL のインストールと設定](#) を参照してください。

[Jira:RHEL-18962](#)

ファイルシステムを作成せずにボリュームを作成する機能のサポート

この機能拡張により、**fs_type=unformatted** オプションを指定することで、ファイルシステムを作成せずに新しいボリュームを作成できるようになりました。

同様に、セーフモードが無効になっていることを確認したうえで、同じ方法を使用することで既存のファイルシステムを削除できます。

[Jira:RHEL-16212](#)

新しい ha_cluster システムロール機能のサポート

ha_cluster システムロールは、次の機能をサポートするようになりました。

- **dlm** や **gfs2** などの Resilient Storage パッケージを含むリポジトリの有効化。リポジトリにアクセスするには、Resilient Storage サブスクリプションが必要です。
- フェンシングレベルの設定。これにより、クラスターは複数のデバイスを使用してノードをフェンスできます。
- ノード属性の設定。

これらの機能を実装するために設定するパラメーターの詳細は、[ha_cluster RHEL システムロールを使用した高可用性クラスターの設定](#) を参照してください。

[Jira:RHEL-15876^{\[1\]}](#), [Jira:RHEL-22106](#), [Jira:RHEL-15910](#)

ForwardToSyslog フラグが journald システムロールでサポートされるようになりました

journald RHEL システムロールでは、**journald_forward_to_syslog** 変数は、受信したメッセージを従来の **syslog** デーモンに転送するかどうかを制御します。この変数のデフォルト値は **false** です。この機能拡張により、インベントリーで **journald_forward_to_syslog** を **true** に設定することで、**ForwardToSyslog** フラグを設定できるようになりました。その結果、Splunk などのリモートロギングシステムを使用する場合、ログは **/var/log** ファイルで利用できるようになります。

[Jira:RHEL-21117](#)

Ansible ホスト名を設定するための rhc ロールの新しい rhc_insights.ansible_host オプション

新しい `rhc_insights.ansible_host` パラメーターを使用して、Red Hat Insights に登録されているシステムの Ansible ホスト名を設定または更新できるようになりました。このパラメーターを設定すると、`/etc/insights-client/insights-client.conf` ファイル内の `ansible_host` 設定が、選択した Ansible ホスト名に変更されます。システムがすでに Red Hat Insights に接続されている場合、このパラメーターによって既存の Ansible ホスト名が更新されます。

[Jira:RHEL-16974](#)

新しい mssql_ha_prep_for_pacemaker 変数

以前は、`microsoft.sql.server` RHEL システムロールには、Pacemaker 用に SQL Server を設定するかどうかを制御する変数がありませんでした。この更新では、`mssql_ha_prep_for_pacemaker` が追加されます。システムを Pacemaker 用に設定せず、別の HA ソリューションを使用する場合は、変数を `false` に設定します。

[Jira:RHEL-19091](#)

sshd ロールは証明書ベースの SSH 認証を設定するようになりました

`sshd` RHEL システムロールを使用すると、SSH 証明書を使用して認証する複数の SSH サーバーを設定および管理できるようになりました。これにより、証明書は信頼できる CA によって署名され、きめ細かいアクセス制御、有効期限、集中型管理が提供されるため、SSH 認証がより安全になります。

[Jira:RHEL-5972](#)

logging システムロールでは、rsyslog_max_message_size の代わりに、logging_max_message_size パラメーターを使用します。

以前は、`rsyslog_max_message_size` パラメーターがサポートされていなかったにもかかわらず、`logging` RHEL システムロールは、`logging_max_message_size` パラメーターではなく `rsyslog_max_message_size` を使用していました。この機能拡張により、ログメッセージの最大サイズを設定するために `rsyslog_max_message_size` ではなく、`logging_max_message_size` が使用されるようになります。

[Jira:RHEL-15037](#)

ratelimit_burst 変数は、logging システムロールで ratelimit_interval が設定されている場合のみ使用されます。

以前は、`logging` RHEL システムロールで、`ratelimit_interval` 変数が設定されていない場合、ロールは `ratelimit_burst` 変数を使用して `rsyslog ratelimit.burst` を設定していました。しかし、`ratelimit_interval` も設定する必要があるため、効果はありませんでした。

この機能拡張では、`ratelimit_interval` が設定されていない場合、ロールは `ratelimit.burst` を設定しません。`ratelimit.burst` を設定する場合は、`ratelimit_interval` 変数および `ratelimit_burst` 変数の両方を設定する必要があります。

[Jira:RHEL-19046](#)

selinux ロールは、存在しないモジュールを指定するとメッセージを出力するようになりました。

このリリースでは、`selinux_modules.path` 変数に存在しないモジュールを指定すると、`selinux` RHEL システムロールによってエラーメッセージが出力されます。

[Jira:RHEL-19043](#)

selinux ロールは、無効モードでの SELinux の設定をサポートするようになりました。

この更新により、**selinux** RHEL システムロールは、SELinux が無効に設定されているノード上の SELinux ポート、ファイルコンテキスト、およびブールマッピングの設定をサポートするようになりました。これは、システム上で SELinux を許可モードまたは強制モードに有効にする前の設定シナリオに役立ちます。

[Jira:RHEL-15870](#)

RHEL System ロールの **metrics** は、**PMIE Webhook** の設定をサポートするようになりました。

この更新により、**metrics** RHEL System Role の **metrics_webhook_endpoint** 変数を使用して、`global webhook_endpoint` PMIE 変数を自動的に設定できるようになりました。これにより、重要なパフォーマンスイベントに関するメッセージを受信する環境用のカスタム URL を提供できるようになります。これは通常、Event-Driven Ansible などの外部ツールで使用されます。

[Jira:RHEL-13760](#)

bootloader RHEL システムロール

この更新では、**bootloader** RHEL システムロールが導入されます。この機能を使用すると、RHEL システム上のブートローダーとカーネルの安定した一貫した設定を実現できます。要件、ロール変数、およびサンプル Playbook の詳細は、`/usr/share/doc/rhel-system-roles/bootloader/` ディレクトリーの README リソースを参照してください。

[Jira:RHEL-16336](#)

4.16. 仮想化

ARM 64 で仮想化がサポートされるようになりました

この更新により、ARM 64 (別名 AArch64) CPU を使用するシステム上で KVM 仮想マシンを作成するためのサポートが導入されました。ただし、AMD64 および Intel 64 システムで利用可能な特定の仮想化機能は、ARM 64 では動作が異なったり、サポートされなかったりする場合があります。

詳細は、[ARM 64 での仮想化が AMD64 および Intel 64 とどのように異なるか](#) を参照してください。

[Jira:RHEL-14097](#)

仮想マシンの外部スナップショット

この更新により、非推奨になった内部スナップショットメカニズムに代わる、仮想マシン (VM) の外部スナップショットメカニズムが導入されました。その結果、完全にサポートされる仮想マシンスナップショットを作成、削除、および復元できるようになりました。外部スナップショットは、コマンドラインインターフェイスと RHEL Web コンソールの両方でより確実に動作します。これは、ライブスナップショットと呼ばれる実行中の仮想マシンのスナップショットにも適用されます。

ただし、一部のコマンドとユーティリティーでは内部スナップショットが作成される場合があることに注意してください。スナップショットが完全にサポートされていることを確認するには、スナップショットが **external** として設定されていることを確認します。以下に例を示します。

```
# virsh snapshot-dumpxml VM-name snapshot-name | grep external
<disk name='vda' snapshot='external' type='file'>
```

[Jira:RHEL-7528](#)

RHEL は仮想マシンのマルチ FD 移行をサポートするようになりました

この更新により、仮想マシンの複数のファイル記述子 (マルチ FD) の移行がサポートされるようになりました。マルチ FD 移行では、複数の並列接続を使用して仮想マシンを移行するため、利用可能なネットワーク帯域幅をすべて活用してプロセスを高速化できます。

この機能は高速ネットワーク (20 Gbps 以上) で使用することを推奨します。

Jira:RHELDPCS-16970^[1]

仮想マシンの移行で、ポストコピープリエンプションがサポートされるようになりました

仮想マシンのポストコピーライブマイグレーションで **postcopy-preempt** 機能が使用されるようになりました。これにより、移行のパフォーマンスと安定性が向上しました。

Jira:RHEL-13004^[1], Jira:RHEL-7100

Secure Execution VMs on IBM Z が暗号化コプロセッサをサポートするようになりました。

この更新により、IBM Secure Execution on IBM Z を使用して、暗号化コプロセッサを仲介デバイスとして仮想マシン (VM) に割り当てることができるようになりました。

暗号化コプロセッサを仲介デバイスとして Secure Execution 仮想マシンに割り当てることで、仮想マシンのセキュリティを損なうことなくハードウェア暗号化を使用できるようになります。

Jira:RHEL-11597^[1]

KVM ゲストでサポートされる第 4 世代 AMD EPYC プロセッサ

第 4 世代 AMD EPYC プロセッサ (AMD Genoa と呼ばれる) のサポートが、KVM ハイパーバイザーとカーネルコード、および libvirt API に追加されました。これにより、KVM 仮想マシンは第 4 世代 AMD EPYC プロセッサを使用できるようになります。

Jira:RHEL-7568

RHEL Web コンソールの新しい仮想化機能

今回の更新で、RHEL Web コンソールに Virtual Machines ページに新機能が追加されました。以下を実行することができます。

- 仮想マシン (VM) の作成中に SSH 公開鍵を追加します。この公開鍵は、新しく作成された仮想マシン上の指定された root 以外のユーザーの `~/.ssh/authorized_keys` ファイルに保存され、指定されたユーザーアカウントへの即時 SSH アクセスが提供されます。
- 新しいストレージプールを作成するときに、**pre-formatted block device** タイプを選択します。これは、raw ディスクデバイスの意図しない再フォーマットを防ぐため、**physical disk device** タイプよりも堅牢な代替手段です。

この更新により、仮想マシンページのいくつかのデフォルトの動作も変更されます。

- **Add disk** ダイアログでは、**Always attach** オプションがデフォルトで設定されるようになりました。
- **Create snapshot** アクションでは、RHEL 9 で非推奨となった内部スナップショットの代わりに、外部スナップショットが使用されるようになりました。外部スナップショットは信頼性が高く、**qcow2** イメージだけでなく、**raw** イメージでも機能します。実行中の仮想マシンのメモリー状態を保持する場合は、メモリースナップショットファイルのロケーションを選択することもできます。

Jira:RHELDPCS-17000^[1]

virtio-mem が AMD64 および Intel 64 システムでサポートされるようになりました

この更新により、AMD64 および Intel 64 システムでの **virtio-mem** 機能のサポートが RHEL 9 に導入されました。**virtio-mem** を使用すると、仮想マシン (VM) のホストメモリーを動的に追加または削除できます。

virtio-mem の詳細は、[virtio-mem を使用した仮想マシンメモリーの追加および削除](#) を参照してください。

Jira:RHELDPCS-17053^[1]

Web コンソールで SPICE を VNC に置き換えることができるようになる

この更新により、Web コンソールを使用して、既存の仮想マシン (VM) で SPICE リモート表示プロトコルを VNC プロトコルに置き換えることができるようになりました。

RHEL 9 では SPICE プロトコルのサポートが削除されているため、SPICE プロトコルを使用する仮想マシンは RHEL 9 ホスト上で起動できません。たとえば、RHEL 8 仮想マシンはデフォルトで SPICE を使用するため、正常に RHEL 9 へ移行するには、SPICE から VNC に切り替える必要があります。

Jira:RHEL-17434

virtio-blk ディスクデバイスの I/O パフォーマンスの向上

この更新により、**virtio-blk** ディスクデバイス内の各 virtqueue に個別の IOThread を設定できるようになります。この設定により、集中的な I/O ワークロード中に複数の CPU を搭載した仮想マシンのパフォーマンスが向上します。

Jira:RHEL-7416

VNC ビューアーが、ramfb のライブマイグレーション後に仮想マシンディスプレイを正しく初期化する

この更新により、仮想マシン (VM) のプライマリーディスプレイとして設定できる **ramfb** フレームバッファデバイスが強化されます。以前は、**ramfb** は移行できなかったため、**ramfb** を使用する仮想マシンではライブマイグレーション後に空白の画面が表示されていました。現在、**ramfb** は、ライブマイグレーションと互換性があります。その結果、移行が完了すると仮想マシンデスクトップが表示されます。

Jira:RHEL-7478

4.17. クラウド環境の RHEL

EC2 上の RHEL インスタンスが IPv6 IMDS 接続をサポートするようになる

この更新により、Amazon Elastic Cloud コンピュート (EC2) 上の RHEL 8 および 9 インスタンスは IPv6 プロトコルを使用してインスタンスメタデータサービス (IMDS) に接続できるようになります。その結果、EC2 上の **cloud-init** を使用して、デュアルスタック IPv4 および IPv6 接続を備えた RHEL インスタンスを設定できるようになります。さらに、IPv6 のみのサブネットで **cloud-init** を使用して RHEL の EC2 インスタンスを起動することもできます。

Jira:RHEL-7278

生成された設定ファイルを削除するための新しい cloud-init clean オプション

cloud-init ユーティリティーに **cloud-init clean --configs** オプションが追加されました。このオプショ

ンを使用すると、インスタンス上の **cloud-init** によって生成された不要な設定ファイルを削除できます。たとえば、ネットワークのセットアップを定義する **cloud-init** 設定ファイルを削除するには、次のコマンドを使用します。

```
cloud-init clean --configs network
```

Jira:RHEL-7311^[1]

4.18. コンテナ

Podman が containers.conf モジュールをサポートするようになりました

Podman モジュールを使用して、事前に定義された設定セットをロードできます。Podman モジュールは、TOML 形式の **containers.conf** ファイルです。

このモジュールは、次のディレクトリまたはそのサブディレクトリにあります。

- rootless ユーザーの場合: **\$HOME/.config/containers/containers.conf.modules**
- root ユーザーの場合: **/etc/containers/containers.conf.modules**、または **/usr/share/containers/containers.conf.modules**

podman --module <your_module_name> コマンドを使用してオンデマンドでモジュールをロードし、システム設定ファイルおよびユーザー設定ファイルをオーバーライドできます。モジュールを操作する際には、次の点に留意してください。

- **--module** オプションを使用して、モジュールを複数回指定できます。
- **<your_module_name>** が絶対パスの場合、設定ファイルは直接ロードされます。
- 相対パスは、前述の3つのモジュールディレクトリを基準にして解決されます。
- **\$HOME** 内のモジュールは、**/etc/** および **/usr/share/** ディレクトリ内のモジュールをオーバーライドします。

詳細は、[アップストリームのドキュメント](#) を参照してください。

Jira:RHELPLAN-167829^[1]

Container Tools パッケージが更新される

Podman、Buildah、Skopeo、crun、runc ツールを含む、更新された Container Tools RPM メタパッケージが利用可能になりました。以前のバージョンに対する主なバグ修正および機能拡張は、以下のとおりです。

Podman v4.9 の主な変更点:

- **podman --module <your_module_name>** コマンドを使用して Podman でモジュールをオンデマンドでロードし、システム設定ファイルおよびユーザー設定ファイルをオーバーライドできるようになりました。
- **create**、**set**、**remove**、**update** サブコマンドのセットを備えた新しい **podman farm** コマンドが追加されました。これらのコマンドを使用すると、さまざまなアーキテクチャーの podman を実行しているマシンにビルドを委託できます。

- Docker compose などの外部 compose プロバイダーを使用して Compose ワークロードを実行する、新しい **podman-compose** コマンドが追加されました。
- **podman build** コマンドは、**--layer-label** および **--cw** オプションをサポートするようになりました。
- **podman generate systemd** コマンドは非推奨になりました。Quadlet を使用して、**systemd** でコンテナと Pod を実行してください。
- **podman build** コマンドは、HereDoc 構文を使用した **Containerfiles** をサポートするようになりました。
- **podman kube play** コマンドは、新しい **--publish-all** オプションをサポートするようになりました。このオプションを使用して、ホスト上のすべての containerPorts を公開します。

注目すべき変更の詳細は、[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。

Jira:RHELPLAN-167796^[1]

Podman v4.9 RESTful API が進捗のデータを表示するようになりました

この機能拡張により、Podman v4.9 RESTful API では、イメージをレジストリーにプルまたはプッシュするときに、進捗データが表示されるようになりました。

Jira:RHELPLAN-167823^[1]

Toolbx が利用可能になる

Toolbx を使用すると、ベースオペレーティングシステムに影響を与えることなく、開発およびデバッグのツール、エディター、ソフトウェア開発キット (SDK) を、Toolbx の完全にミュータブルなコンテナにインストールできます。Toolbx コンテナは、**registry.access.redhat.com/ubi9.4/toolbox:latest** イメージをベースとしています。

Jira:RHELDPCS-16241^[1]

SQLite は、Podman のデフォルトのデータベースバックエンドとして完全にサポートされるようになりました

Podman v4.9 では、以前テクノロジープレビューとして利用可能だった Podman の SQLite データベースバックエンドが完全にサポートされるようになりました。SQLite データベースは、コンテナメタデータを操作する際に、より優れた安定性、パフォーマンス、一貫性を提供します。SQLite データベースバックエンドは、RHEL 9.4 の新規インストールのデフォルトバックエンドです。以前の RHEL バージョンからアップグレードする場合、デフォルトバックエンドは BoltDB になります。

containers.conf ファイルの **database_backend** オプションを使用してデータベースバックエンドを明示的に設定した場合、Podman は指定されたバックエンドを引き続き使用します。

Jira:RHELPLAN-168180^[1]

管理者は nftables を使用してファイアウォールルールの分離を設定できます。

iptables がインストールされていないシステムでも、Podman コンテナネットワークスタックである Netavark を使用できます。以前は、Netavark の前身であるコンテナネットワークインターフェイス (CNI) ネットワーキングを使用すると、**iptables** がインストールされていないシステムでコンテナネットワークを設定する方法はありませんでした。この機能強化により、Netavark ネットワークスタックは、**nftables** のみがインストールされているシステムでも動作し、自動的に生成されたファイアウォールルールの分離が改善されます。

Jira:RHELDOCS-16955^[1]

Containerfile が複数行の命令をサポートするようになりました

Containerfile ファイルで複数行の HereDoc 命令 (Here Document 表記) を使用すると、このファイルを簡素化し、複数の **RUN** ディレクティブを実行することで発生するイメージレイヤーの数を減らすことができます。

たとえば、元の **Containerfile** が次の **RUN** ディレクティブを含むとします。

```
RUN dnf update
RUN dnf -y install golang
RUN dnf -y install java
```

複数の **RUN** ディレクティブの代わりに、HereDoc 表記を使用できます。

```
RUN <<EOF
dnf update
dnf -y install golang
dnf -y install java
EOF
```

Jira:RHELPLAN-168185^[1]

gvisor-tap-vsock パッケージが利用可能になる

gvisor-tap-vsock パッケージは、**libslirp** ユーザーモードネットワークライブラリーと VPNKit ツールおよびサービスの代替手段です。このパッケージは Go で書かれており、gVisor のネットワークスタックに基づいています。**libslirp** と比較して、**gvisor-tap-vsock** ライブラリーは、設定可能な DNS サーバーと動的ポート転送をサポートします。podman-machine の仮想マシンで **gvisor-tap-vsock** ネットワークライブラリーを使用できます。仮想マシンを管理するための **podman machine** コマンドは、現在 Red Hat Enterprise Linux ではサポートされていません。

Jira:RHELPLAN-167396^[1]

第5章 外部カーネルパラメーターへの重要な変更

この章では、システム管理者向けに、Red Hat Enterprise Linux 9.4 で配布されるカーネルの重要な変更点の概要を説明します。これらの変更には、たとえば、追加または更新された **proc** エントリー、**sysctl** エントリー、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、または注目すべき動作の変更などが含まれます。

新しいカーネルパラメーター

accept_memory=

[MM]

値:

lazy (デフォルト)

デフォルトでは、起動時間が長くなるのを避けるために、受け入れられないメモリーは遅延して受け入れられます。**lazy** オプションは、すべてのメモリーが最終的に受け入れられるまで、実行時のオーバーヘッドをいくつか追加します。ほとんどの場合、オーバーヘッドはごくわずかです。

eager

一部のワークロードやデバッグの目的で、**accept_memory=eager** を使用して、起動時にすべてのメモリーを一度に受け入れることができます。

arm64.nomops

[ARM64]

メモリーコピーおよびメモリーセット命令のサポートを無条件に無効にします。

cgroup_favordynmods=

[KNL]

favordynmods を有効または無効にします。

値:

- **true**
- **false**

デフォルトの値は **CONFIG_CGROUP_FAVOR_DYNMODS** です。

early_page_ext

[KNL]

より早いブート割り当てをカバーするために、**page_ext** の初期化を早い段階で強制します。

副次的影響として、これを達成するために一部の最適化が無効になる可能性があることに注意してください。たとえば、並列化されたメモリー初期化が無効になります。したがって、特にメモリー容量の多いシステムでは、起動プロセスに時間がかかる可能性があります。

CONFIG_PAGE_EXTENSION=y で利用可能です。

fw_devlink.sync_state=

[KNL]

プローブ可能なすべてのデバイスがプローブを完了したときに、このパラメーターは、**sync_state()** 呼び出しをまだ受信していないデバイスに対して何を行うかを制御します。

値:

strict (デフォルト)

コンシューマーがプローブに成功するまで待機し続けます。

timeout

コンシューマーの待機をあきらめ、**deferred_probe_timeout**の有効期限が切れた後、または **CONFIG_MODULES** が **false** の場合は **late_initcall()** によって、**sync_state()** 呼び出しをまだ受信していないデバイスで **sync_state()** を呼び出します。

ia32_emulation=

[X86-64]

値:

true

32 ビットプログラムのロードと 32 ビット syscall の実行を許可し、基本的に起動時に **IA32_EMULATION_DEFAULT_DISABLED** をオーバーライドします。

false

IA32 エミュレーションを無条件に無効にします。

kunit.enable=

[KUNIT]

KUnit テストの実行を有効にします。完全に有効にするには、**CONFIG_KUNIT** を設定する必要があります。

KUNIT_DEFAULT_ENABLED を使用してデフォルト値をオーバーライドできます。

デフォルトは 1 (有効化済み) です。

mtrr=debug

[X86]

起動時に MTRR レジスターに関連するデバッグ情報を出力できるようにします。

rcupdate.rcu_cpu_stall_cputime=

[KNL]

サンプリング期間中の CPU 時間、割り込みおよびタスクの数に関する統計を提供します。複数の連続した RCU ストールの場合、すべてのサンプリング期間は最初の RCU ストールタイムアウトの半分で開始されます。

rcupdate.rcu_exp_stall_task_details=

[KNL]

高速 RCU CPU ストール警告中に、現在の高速 RCU 猶予期間をブロックしているすべてのタスクのスタックダンプを出力します。

spec_rstack_overflow=

[X86]

AMD Zen CPU 上の RAS オーバーフロー軽減策を制御します。

値:

off

軽減策を無効にします。

microcode

マイクロコードの軽減策のみを有効にします。

safe-ret (デフォルト)

ソフトウェアのみの安全な RET 軽減策を有効にします。

ibpb

カーネルエントリーで IBPB を発行して軽減策を有効にします。

ibpb-vmexit

VMEXIT でのみ IBPB を発行します。この軽減策はクラウド環境に特有のものです。

workqueue.unbound_cpus=

[KNL,SMP]

バインドされていないワークキューで使用する 1 つまたは複数の CPU を制限するように指定します。

値: CPU のリスト。

デフォルトでは、すべてのオンライン CPU がバインドされていないワークキューに使用できます。

更新されたカーネルパラメーター

amd_iommu=

[HW, X86-64]

システム内の AMD IOMMU ドライバーにパラメーターを渡します。

値:

fullflush

非推奨。 **iommu.strict=1** と同等です。

off

システム内で見つかった AMD IOMMU を初期化しないでください。

force_isolation

すべてのデバイスのデバイス分離を強制します。IOMMU ドライバーは、必要に応じて分離要件を引き上げることができなくなりました。このオプションは **iommu=pt** をオーバーライドしません。

force_enable

IOMMU を有効にするとバグが発生することがわかっているプラットフォームで、IOMMU を強制的に有効にします。このオプションは注意して使用してください。

新規: pgtbl_v1 (デフォルト)

DMA-API にはバージョン 1 ページテーブルを使用します。

新規: pgtbl_v2

DMA-API にはバージョン 2 ページテーブルを使用します。

新規: **irtcachedis**

割り込み再マッピングテーブル (IRT) キャッシュを無効にします。

nosmt

[KNL, PPC, S390]

対称マルチスレッド (SMT) を無効にします。 **smt=1** と同等です。

[KNL, X86, PPC]

対称マルチスレッド (SMT) を無効にします。

nosmt=force

SMT を強制的に無効にします。 **sysfs** 制御ファイルを使用して元に戻すことはできません。

page_reporting.page_reporting_order=

[KNL]

最小ページのレポート順序。

値: 整数

最小ページのレポート順序を調整します。

新規: **MAX_ORDER** を超えるとページレポートが無効になります。

tsc=

TSC のクロックソース安定性チェックを無効にします。

値:

[x86] **reliable**

tsc クロックソースを信頼できるものとしてマークします。これにより、実行時のクロックソース検証と起動時に実行される安定性チェックが無効になります。古いハードウェアおよび仮想化環境で高分解能タイマーモードを有効にするために使用されます。

[x86] **noirqtime**

irq アカウンティングの実行に、TSC を使用しないでください。RDTSC が遅れており、このアカウンティングによってオーバーヘッドが追加される可能性があるプラットフォームで、実行時に **IRQ_TIME_ACCOUNTING** を無効にするために使用されます。

[x86] **unstable**

TSC クロックソースを不安定としてマークします。これにより、起動時に TSC が無条件に不安定であるとマークされ、TSC ウォッチドッグがそれを検知すると、それ以上の不安定さが回避されません。

[x86] **nowatchdog**

クロックソースウォッチドッグを無効にします。レイテンシー要件が厳しい状況 (クロックソースウォッチドッグからの割り込みが許容されない場合) で使用されます。

[x86] **recalibrate**

MSR または CPUID(0x15) を使用して TSC 周波数が HW または FW から取得されるシステム上で、HW タイマー (HPET または PM タイマー) に対する再キャリブレーションを強制します。差が 500 ppm を超える場合は警告します。

新規: [x86] watchdog

TSC をウォッチドッグクロックソースとして使用し、他の HW タイマー (HPET または PM タイマー) をチェックします。ただし、TSC が信頼できると判断されたシステムでのみ使用します。以前の **tsc=nowatchdog** は、これを抑制します。後の **tsc=nowatchdog** は、これをオーバーライドします。コンソールメッセージには、このような抑制またはオーバーライドがフラグ付けされません。

usbcore.authorized_default=

[USB]

デフォルトの USB デバイス認証。

値:

新規: -1 (デフォルト)

承認済み (1 と同じ)。

0

非承認

1

承認済み

2

デバイスが内部ポートに接続する場合は承認されます。

削除されたカーネルパラメーター

- **cpu0_hotplug**
- **sysfs.deprecated**

新しい sysctl パラメーター**io_uring_group**

値:

1

io_uring インスタンスを作成するには、プロセスが特権 (**CAP_SYS_ADMIN**) を持っているか、**io_uring_group** グループに属している必要があります。

-1 (デフォルト)

CAP_SYS_ADMIN 権限を持つプロセスのみが **io_uring** インスタンスを作成できます。

numa_balancing_promote_rate_limit_MBps

異なるメモリータイプ間のプロモートおよびデモートのスループットが高すぎると、アプリケーションのレイテンシーに悪影響が出る可能性があります。このパラメーターを使用して、プロモーションのスループットをレート制限できます。ノードあたりの最大プロモーションスループット (MB/秒) は、設定された値以下に制限されます。

経験則としては、これを PMEM ノードの書き込み帯域幅の 1/10 未満に設定します。

sysctl パラメーターの更新

io_uring_disabled

io_uring_disabled

すべてのプロセスが新しい **io_uring** インスタンスを作成するのを防ぎます。これを有効にすると、カーネルの攻撃対象領域が縮小されます。

値:

新規: 0

すべてのプロセスは通常どおり **io_uring** インスタンスを作成できます。

新規: 1

io_uring_group グループに属さない非特権プロセスでは、**io_uring** の作成は無効になります。**io_uring_setup()** は **-EPERM** で失敗します。既存の **io_uring** インスタンスは引き続き使用できます。

詳細は、**io_uring_group** のドキュメントを参照してください。

新規: 2 (デフォルト)

io_uring の作成は、すべてのプロセスで無効化されています。**io_uring_setup()** は常に **-EPERM** で失敗します。既存の **io_uring** インスタンスは引き続き使用できます。

第6章 デバイスドライバー

6.1. 新しいドライバー

表6.1 暗号化ドライバー

説明	名前	アーキテクチャーに限定
IAA Compression Accelerator Crypto Driver	iaa_crypto	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
Intel® QuickAssist Technology - 0.6.0	intel_qat	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
Intel® QuickAssist Technology - 0.6.0	qat_4xxx	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
Intel® QuickAssist Technology - 0.6.0	qat_c3xxx	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
Intel® QuickAssist Technology - 0.6.0	qat_c3xxxvf	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
Intel® QuickAssist Technology - 0.6.0	qat_c62x	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
Intel® QuickAssist Technology - 0.6.0	qat_c62xvf	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
Intel® QuickAssist Technology - 0.6.0	qat_dh895xcc	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
Intel® QuickAssist Technology - 0.6.0	qat_dh895xccvf	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

表6.2 ネットワークドライバー

説明	名前	アーキテクチャーに限定
	bcm-phy-ptp	64 ビット ARM アーキテクチャー、IBM Power Systems、AMD および Intel 64 ビットアーキテクチャー
	mt7925-common	64 ビット ARM アーキテクチャー、AMD および Intel 64 ビットアーキテクチャー

説明	名前	アーキテクチャーに限定
	mt7925e	64 ビット ARM アーキテクチャー、AMD および Intel 64 ビットアーキテクチャー
	mt792x-lib	64 ビット ARM アーキテクチャー、AMD および Intel 64 ビットアーキテクチャー
PCI バス上の Bosch M_CAN コントローラー用 CAN バスドライバー	m_can_pci	IBM Power Systems、AMD、Intel 64 ビットアーキテクチャー
Bosch M_CAN コントローラー用 CAN バスドライバー	m_can	IBM Power Systems、AMD、Intel 64 ビットアーキテクチャー
8 台のデバイス USB2CAN インターフェイス用 CAN ドライバー	usb_8dev	IBM Power Systems、AMD、Intel 64 ビットアーキテクチャー
EMS Dr.用 CAN ドライバーThomas Wuensche CAN/USB インターフェイス	ems_usb	IBM Power Systems、AMD、Intel 64 ビットアーキテクチャー
Kvaser CAN/USB デバイス用の CAN ドライバー	kvaser_usb	IBM Power Systems、AMD、Intel 64 ビットアーキテクチャー
PEAK-System USB アダプター用 CAN ドライバー	peak_usb	IBM Power Systems、AMD、Intel 64 ビットアーキテクチャー
Intel® Infrastructure Data Path Function Linux Driver	idpf	64 ビット ARM アーキテクチャー、IBM Power Systems、AMD および Intel 64 ビットアーキテクチャー
Marvell 88Q2XXX 100/1000BASE-T1 Automotive Ethernet PHY driver	marvell-88q2xxx	64 ビット ARM アーキテクチャー、IBM Power Systems、AMD および Intel 64 ビットアーキテクチャー
Marvell Octeon EndPoint NIC Driver	octeon_ep	64 ビット ARM アーキテクチャー、IBM Power Systems、AMD および Intel 64 ビットアーキテクチャー
Microchip 251x/25625 CAN driver	mcp251x	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
Microchip MCP251xFD Family CAN controller driver	mcp251xfd	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

説明	名前	アーキテクチャーに限定
NXP imx8 DWMAC Specific Glue layer	dwmac-imx	64 ビット ARM アーキテクチャー
	bcm-phy-ptp	64 ビット ARM アーキテクチャー、IBM Power Systems、AMD および Intel 64 ビットアーキテクチャー
Realtek 802.11ax wireless 8852C driver	rtw89_8852c	64 ビット ARM アーキテクチャー、AMD および Intel 64 ビットアーキテクチャー
Realtek 802.11ax wireless 8852CE driver	rtw89_8852ce	64 ビット ARM アーキテクチャー、AMD および Intel 64 ビットアーキテクチャー
シリアルライン CAN インターフェイス	slcan	IBM Power Systems、AMD、Intel 64 ビットアーキテクチャー
PEAK PCAN PCIe/M.2 FD ファミリーカード用ソケット CAN ドライバー	peak_pciefd	IBM Power Systems、AMD、Intel 64 ビットアーキテクチャー
	bcm-phy-ptp	64 ビット ARM アーキテクチャー、IBM Power Systems、AMD および Intel 64 ビットアーキテクチャー
	mt7925-common	64 ビット ARM アーキテクチャー、AMD および Intel 64 ビットアーキテクチャー
	mt7925e	64 ビット ARM アーキテクチャー、AMD および Intel 64 ビットアーキテクチャー
	mt792x-lib	64 ビット ARM アーキテクチャー、AMD および Intel 64 ビットアーキテクチャー

表6.3 プラットフォームドライバー

説明	名前	アーキテクチャーに限定
AMD HSMP Platform Interface Driver - 2.0	amd_hsmp	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
AMD Platform Management Framework Driver	amd-pmf	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

説明	名前	アーキテクチャーに限定
Intel TPMI enumeration module	intel_vsec_tpmi	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
Intel TPMI SST Driver	isst_tpmi	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
Intel TPMI UFS Driver	intel-uncore-frequency-tpmi	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
Intel Uncore Frequency Common Module	intel-uncore-frequency-common	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
Intel Uncore Frequency Limits Driver	intel-uncore-frequency	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
Intel WMI Thunderbolt force power driver	intel-wmi-thunderbolt	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
Mellanox PMC driver	mlxbf-pmc	64 ビット ARM アーキテクチャー
	intel-hid	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
	isst_tpmi_core	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

表6.4 グラフィックドライバーとその他のドライバー

説明	名前	アーキテクチャーに限定
AMD XCP Platform Devices	amdxcpc	64 ビット ARM アーキテクチャー、IBM Power Systems、AMD および Intel 64 ビットアーキテクチャー
DRM execution context	drm_exec	
Range suballocator helper	drm_suballoc_helper	64 ビット ARM アーキテクチャー、IBM Power Systems、AMD および Intel 64 ビットアーキテクチャー

説明	名前	アーキテクチャーに限定
	regmap-ram	64 ビット ARM アーキテクチャー、IBM Power Systems、AMD および Intel 64 ビットアーキテクチャー
	regmap-raw-ram	64 ビット ARM アーキテクチャー、IBM Power Systems、AMD および Intel 64 ビットアーキテクチャー
	regmap-ram	64 ビット ARM アーキテクチャー、IBM Power Systems、AMD および Intel 64 ビットアーキテクチャー
	regmap-raw-ram	64 ビット ARM アーキテクチャー、IBM Power Systems、AMD および Intel 64 ビットアーキテクチャー
	regmap-ram	64 ビット ARM アーキテクチャー、IBM Power Systems、AMD および Intel 64 ビットアーキテクチャー
	regmap-raw-ram	64 ビット ARM アーキテクチャー、IBM Power Systems、AMD および Intel 64 ビットアーキテクチャー
Arm FF-A interface driver	ffa-module	64 ビット ARM アーキテクチャー
NVIDIA BlueField-3 GPIO Driver	gpio-mlxbf3	64 ビット ARM アーキテクチャー
パススルーデバイスの I/O アドレス空間管理	iommufd	
CS42L43 Core Driver	cs42l43	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
CS42L43 SoundWire Driver	cs42l43-sdw	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
MEI GSC Proxy	mei_gsc_proxy	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
	pwrseq_emmc	64 ビット ARM アーキテクチャー
	pwrseq_simple	64 ビット ARM アーキテクチャー

説明	名前	アーキテクチャーに限定
SDHCI platform driver for Synopsys DWC MSHC	sdhci-of-dwcmshc	64 ビット ARM アーキテクチャー
	arm_cspmu_module	64 ビット ARM アーキテクチャー
NVIDIA pinctrl driver	pinctrl-mlxbf3	64 ビット ARM アーキテクチャー
NXP i.MX93 power domain driver	imx93-pd	64 ビット ARM アーキテクチャー
Intel RAPL TPMI Driver	intel_rapl_tpmi	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
Mellanox BlueField power driver	pwr-mlxbf	64 ビット ARM アーキテクチャー
NXP i.MX93 src driver	imx93-src	64 ビット ARM アーキテクチャー
Provide Trusted Security Module attestation reports via configfs	tsm	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

6.2. 更新されたドライバー

表6.5 ストレージドライバーの更新

説明	名前	現行バージョン	アーキテクチャーに限定
Broadcom MegaRAID SAS Driver	megaraid_sas	07.727.03.00-rc1	64 ビット ARM アーキテクチャー、IBM Power Systems、AMD および Intel 64 ビットアーキテクチャー
Driver for Microchip Smart Family Controller	smartpqi	2.1.24-046	64 ビット ARM アーキテクチャー、IBM Power Systems、AMD および Intel 64 ビットアーキテクチャー
Emulex LightPulse ファイバーチャネル SCSI ドライバー	lpfc	0:14.2.0.16	64 ビット ARM アーキテクチャー、IBM Power Systems、AMD および Intel 64 ビットアーキテクチャー
MPI3 Storage Controller Device Driver	mpi3mr	8.5.0.0.50	

第7章 利用可能な BPF 機能

この章では、Red Hat Enterprise Linux 9 のこのマイナーバージョンのカーネルで利用可能な Berkeley Packet Filter (BPF) 機能の完全なリストを提供します。表には次のリストが含まれます。

- システム設定とその他のオプション
- 利用可能なプログラムの種類とサポートされているヘルパー
- 利用可能なマップの種類

この章には、**bpftool feature** コマンドの自動生成された出力が含まれています。

表7.1 システム設定とその他のオプション

オプション	値
unprivileged_bpf_disabled	2 (特権ユーザーに限定された bpf() syscall、管理者は変更可能)
JIT コンパイラー	1 (有効)
JIT コンパイラーの強化	1 (権限のないユーザーに対して有効)
JIT コンパイラー kallsyms エクスポート	1 (ルートで有効)
非特権ユーザーの JIT のメモリー制限	528482304
CONFIG_BPF	y
CONFIG_BPF_SYSCALL	y
CONFIG_HAVE_EBPF_JIT	y
CONFIG_BPF_JIT	y
CONFIG_BPF_JIT_ALWAYS_ON	y
CONFIG_DEBUG_INFO_BTF	y
CONFIG_DEBUG_INFO_BTF_MODULES	y
CONFIG_CGROUPS	y
CONFIG_CGROUP_BPF	y
CONFIG_CGROUP_NET_CLASSID	y
CONFIG_SOCK_CGROUP_DATA	y

オプション	値
CONFIG_BPF_EVENTS	y
CONFIG_KPROBE_EVENTS	y
CONFIG_UPROBE_EVENTS	y
CONFIG_TRACING	y
CONFIG_FTRACE_SYSCALLS	y
CONFIG_FUNCTION_ERROR_INJECTION	y
CONFIG_BPF_KPROBE_OVERRIDE	n
CONFIG_NET	y
CONFIG_XDP_SOCKETS	y
CONFIG_LWTUNNEL_BPF	y
CONFIG_NET_ACT_BPF	m
CONFIG_NET_CLS_BPF	m
CONFIG_NET_CLS_ACT	y
CONFIG_NET_SCH_INGRESS	m
CONFIG_XFRM	y
CONFIG_IP_ROUTE_CLASSID	y
CONFIG_IPV6_SEG6_BPF	y
CONFIG_BPF_LIRC_MODE2	n
CONFIG_BPF_STREAM_PARSER	y
CONFIG_NETFILTER_XT_MATCH_BPF	m
CONFIG_BPFILTER	n
CONFIG_BPFILTER_UMH	n

オプション	値
CONFIG_TEST_BPF	m
CONFIG_HZ	1000
bpf() syscall	available
大きなプログラムサイズの制限	available
有界ループのサポート	available
ISA エクステンション v2	available
ISA エクステンション v3	available

表7.2 利用可能なプログラムの種類とサポートされているヘルパー

プログラムの種類	利用可能なヘルパー
socket_filter	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_perf_event_output, bpf_skb_load_bytes, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_get_socket_uid, bpf_skb_load_bytes_relative, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

プログラムの種類	利用可能なヘルパー
kprobe	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task, bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str, bpf_perf_event_read_value, bpf_get_stack, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strotol, bpf_strtoul, bpf_send_signal, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_send_signal_thread, bpf_jiffies64, bpf_get_ns_current_pid_tgid, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_get_task_stack, bpf_copy_from_user, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_get_attach_cookie, bpf_task_pt_regs, bpf_get_branch_snapshot, bpf_find_vma, bpf_loop, bpf_strcmp, bpf_copy_from_user_task, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

プログラムの種類	利用可能なヘルパー
sched_cls	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_skb_store_bytes, bpf_l3_csum_replace, bpf_l4_csum_replace, bpf_tail_call, bpf_clone_redirect, bpf_get_cgroup_classid, bpf_skb_vlan_push, bpf_skb_vlan_pop, bpf_skb_get_tunnel_key, bpf_skb_set_tunnel_key, bpf_redirect, bpf_get_route_realm, bpf_perf_event_output, bpf_skb_load_bytes, bpf_csum_diff, bpf_skb_get_tunnel_opt, bpf_skb_set_tunnel_opt, bpf_skb_change_proto, bpf_skb_change_type, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_skb_change_tail, bpf_skb_pull_data, bpf_csum_update, bpf_set_hash_invalid, bpf_get_numa_node_id, bpf_skb_change_head, bpf_get_socket_cookie, bpf_get_socket_uid, bpf_set_hash, bpf_skb_adjust_room, bpf_skb_get_xfrm_state, bpf_skb_load_bytes_relative, bpf_fib_lookup, bpf_skb_cgroup_id, bpf_get_current_cgroup_id, bpf_skb_ancestor_cgroup_id, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_sk_fullsock, bpf_tcp_sock, bpf_skb_ecn_set_ce, bpf_get_listener_sock, bpf_skc_lookup_tcp, bpf_tcp_check_syncookie, bpf_strtol, bpf_strtoul, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_tcp_gen_syncookie, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_sk_assign, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_csum_level, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_skb_cgroup_classid, bpf_redirect_neigh, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_redirect_peer, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_check_mtu, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strcmp, bpf_skb_set_timestamp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_tcp_raw_gen_syncookie_ipv4, bpf_tcp_raw_gen_syncookie_ipv6, bpf_tcp_raw_check_syncookie_ipv4, bpf_tcp_raw_check_syncookie_ipv6, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

プログラムの種類	利用可能なヘルパー
sched_act	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_skb_store_bytes, bpf_l3_csum_replace, bpf_l4_csum_replace, bpf_tail_call, bpf_clone_redirect, bpf_get_cgroup_classid, bpf_skb_vlan_push, bpf_skb_vlan_pop, bpf_skb_get_tunnel_key, bpf_skb_set_tunnel_key, bpf_redirect, bpf_get_route_realm, bpf_perf_event_output, bpf_skb_load_bytes, bpf_csum_diff, bpf_skb_get_tunnel_opt, bpf_skb_set_tunnel_opt, bpf_skb_change_proto, bpf_skb_change_type, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_skb_change_tail, bpf_skb_pull_data, bpf_csum_update, bpf_set_hash_invalid, bpf_get_numa_node_id, bpf_skb_change_head, bpf_get_socket_cookie, bpf_get_socket_uid, bpf_set_hash, bpf_skb_adjust_room, bpf_skb_get_xfrm_state, bpf_skb_load_bytes_relative, bpf_fib_lookup, bpf_skb_cgroup_id, bpf_get_current_cgroup_id, bpf_skb_ancestor_cgroup_id, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_sk_fullsock, bpf_tcp_sock, bpf_skb_ecn_set_ce, bpf_get_listener_sock, bpf_skc_lookup_tcp, bpf_tcp_check_syncookie, bpf_strtol, bpf_strtoul, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_tcp_gen_syncookie, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_sk_assign, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_csum_level, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_skb_cgroup_classid, bpf_redirect_neigh, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_redirect_peer, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_check_mtu, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strcmp, bpf_skb_set_timestamp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_tcp_raw_gen_syncookie_ipv4, bpf_tcp_raw_gen_syncookie_ipv6, bpf_tcp_raw_check_syncookie_ipv4, bpf_tcp_raw_check_syncookie_ipv6, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

プログラムの種類	利用可能なヘルパー
tracepoint	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task, bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str, bpf_perf_event_read_value, bpf_get_stack, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strotol, bpf_strtoul, bpf_send_signal, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_send_signal_thread, bpf_jiffies64, bpf_get_ns_current_pid_tgid, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_get_task_stack, bpf_copy_from_user, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_get_attach_cookie, bpf_task_pt_regs, bpf_get_branch_snapshot, bpf_find_vma, bpf_loop, bpf_strcmp, bpf_copy_from_user_task, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
xdp	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_redirect, bpf_perf_event_output, bpf_csum_diff, bpf_get_current_task, bpf_get_numa_node_id, bpf_xdp_adjust_head, bpf_redirect_map, bpf_xdp_adjust_meta, bpf_xdp_adjust_tail, bpf_fib_lookup, bpf_get_current_cgroup_id, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_sk_lookup_tcp, bpf_tcp_check_syncookie, bpf_strotol, bpf_strtoul, bpf_tcp_gen_syncookie, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_check_mtu, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strcmp, bpf_xdp_get_buff_len, bpf_xdp_load_bytes, bpf_xdp_store_bytes, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_tcp_raw_gen_syncookie_ipv4, bpf_tcp_raw_gen_syncookie_ipv6, bpf_tcp_raw_check_syncookie_ipv4, bpf_tcp_raw_check_syncookie_ipv6, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

プログラムの種類	利用可能なヘルパー
perf_event	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task, bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str, bpf_perf_event_read_value, bpf_perf_prog_read_value, bpf_get_stack, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strotol, bpf_strtoul, bpf_send_signal, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_send_signal_thread, bpf_jiffies64, bpf_read_branch_records, bpf_get_ns_current_pid_tgid, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_get_task_stack, bpf_copy_from_user, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_get_attach_cookie, bpf_task_pt_regs, bpf_get_branch_snapshot, bpf_find_vma, bpf_loop, bpf_strncmp, bpf_copy_from_user_task, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
cgroup_skb	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_perf_event_output, bpf_skb_load_bytes, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_get_socket_uid, bpf_skb_load_bytes_relative, bpf_skb_cgroup_id, bpf_get_current_cgroup_id, bpf_get_local_storage, bpf_skb_ancestor_cgroup_id, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_sk_fullsock, bpf_tcp_sock, bpf_skb_ecn_set_ce, bpf_get_listener_sock, bpf_skc_lookup_tcp, bpf_strotol, bpf_strtoul, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_sk_cgroup_id, bpf_skb_ancestor_cgroup_id, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

プログラムの種類	利用可能なヘルパー
cgroup_sock	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_get_cgroup_classid, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_get_current_cgroup_id, bpf_get_local_storage, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_sk_storage_get, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_netns_cookie, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_loop, bpf_strncmp, bpf_get_retval, bpf_set_retval, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
lwt_in	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_cgroup_classid, bpf_get_route_realm, bpf_perf_event_output, bpf_skb_load_bytes, bpf_csum_diff, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_skb_pull_data, bpf_get_numa_node_id, bpf_lwt_push_encap, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

プログラムの種類	利用可能なヘルパー
lwt_out	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_cgroup_classid, bpf_get_route_realm, bpf_perf_event_output, bpf_skb_load_bytes, bpf_csum_diff, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_skb_pull_data, bpf_get_numa_node_id, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
lwt_xmit	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_skb_store_bytes, bpf_l3_csum_replace, bpf_l4_csum_replace, bpf_tail_call, bpf_clone_redirect, bpf_get_cgroup_classid, bpf_skb_get_tunnel_key, bpf_skb_set_tunnel_key, bpf_redirect, bpf_get_route_realm, bpf_perf_event_output, bpf_skb_load_bytes, bpf_csum_diff, bpf_skb_get_tunnel_opt, bpf_skb_set_tunnel_opt, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_skb_change_tail, bpf_skb_pull_data, bpf_csum_update, bpf_set_hash_invalid, bpf_get_numa_node_id, bpf_skb_change_head, bpf_lwt_push_encap, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_csum_level, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

プログラムの種類	利用可能なヘルパー
sock_ops	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_setsockopt, bpf_sock_map_update, bpf_getsockopt, bpf_sock_ops_cb_flags_set, bpf_sock_hash_update, bpf_get_current_cgroup_id, bpf_get_local_storage, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_tcp_sock, bpf_strtol, bpf_strtoul, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_netns_cookie, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_load_hdr_opt, bpf_store_hdr_opt, bpf_reserve_hdr_opt, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
sk_skb	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_skb_store_bytes, bpf_tail_call, bpf_perf_event_output, bpf_skb_load_bytes, bpf_get_current_task, bpf_skb_change_tail, bpf_skb_pull_data, bpf_get_numa_node_id, bpf_skb_change_head, bpf_get_socket_cookie, bpf_get_socket_uid, bpf_skb_adjust_room, bpf_sk_redirect_map, bpf_sk_redirect_hash, bpf_get_current_cgroup_id, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_skc_lookup_tcp, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

プログラムの種類	利用可能なヘルパー
cgroup_device	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_get_cgroup_classid, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_current_cgroup_id, bpf_get_local_storage, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtol, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
sk_msg	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_cgroup_classid, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_msg_redirect_map, bpf_msg_apply_bytes, bpf_msg_cork_bytes, bpf_msg_pull_data, bpf_msg_redirect_hash, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_msg_push_data, bpf_msg_pop_data, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtol, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_netns_cookie, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

プログラムの種類	利用可能なヘルパー
raw_tracepoint	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task, bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str, bpf_perf_event_read_value, bpf_get_stack, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strotol, bpf_strtoul, bpf_send_signal, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_send_signal_thread, bpf_jiffies64, bpf_get_ns_current_pid_tgid, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_get_task_stack, bpf_copy_from_user, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_task_pt_regs, bpf_get_branch_snapshot, bpf_find_vma, bpf_loop, bpf_strncmp, bpf_copy_from_user_task, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
cgroup_sock_addr	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_get_cgroup_classid, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_setsockopt, bpf_getsockopt, bpf_bind, bpf_get_current_cgroup_id, bpf_get_local_storage, bpf_sk_lookup_tcp, bpf_sk_lookup_udp, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_skc_lookup_tcp, bpf_strotol, bpf_strtoul, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_netns_cookie, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_get_retval, bpf_set_retval, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

プログラムの種類	利用可能なヘルパー
lwt_seg6local	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_cgroup_classid, bpf_get_route_realm, bpf_perf_event_output, bpf_skb_load_bytes, bpf_csum_diff, bpf_skb_under_cgroup, bpf_get_hash_recalc, bpf_get_current_task, bpf_skb_pull_data, bpf_get_numa_node_id, bpf_lwt_seg6_store_bytes, bpf_lwt_seg6_adjust_srh, bpf_lwt_seg6_action, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strotol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
lirc_mode2	サポート対象外
sk_reuseport	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_skb_load_bytes, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_socket_cookie, bpf_skb_load_bytes_relative, bpf_get_current_cgroup_id, bpf_sk_select_reuseport, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strotol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

プログラムの種類	利用可能なヘルパー
flow_dissector	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_skb_load_bytes, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_current_cgrouop_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgrouop_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
cgroup_sysctl	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_get_cgroup_classid, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_current_cgrouop_id, bpf_get_local_storage, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_sysctl_get_name, bpf_sysctl_get_current_value, bpf_sysctl_get_new_value, bpf_sysctl_set_new_value, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgrouop_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

プログラムの種類	利用可能なヘルパー
raw_tracepoint_wri table	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task, bpf_current_task_under_cgroup, bpf_get_numa_node_id, bpf_probe_read_str, bpf_perf_event_read_value, bpf_get_stack, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strotol, bpf_strtoul, bpf_send_signal, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_send_signal_thread, bpf_jiffies64, bpf_get_ns_current_pid_tgid, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_get_task_stack, bpf_copy_from_user, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_task_pt_regs, bpf_get_branch_snapshot, bpf_find_vma, bpf_loop, bpf_strncmp, bpf_copy_from_user_task, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
cgroup_sockopt	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_get_cgroup_classid, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_current_cgroup_id, bpf_get_local_storage, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_tcp_sock, bpf_strotol, bpf_strtoul, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_netns_cookie, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_loop, bpf_strncmp, bpf_get_retval, bpf_set_retval, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
tracing	
struct_ops	
ext	
lsm	

プログラムの種類	利用可能なヘルパー
sk_lookup	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_perf_event_output, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_current_cgrouop_id, bpf_sk_release, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgrouop_id, bpf_sk_assign, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_ktime_get_coarse_ns, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_skc_to_unix_sock, bpf_loop, bpf_strncmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete
syscall	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_probe_read, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_pid_tgid, bpf_get_current_uid_gid, bpf_get_current_comm, bpf_perf_event_read, bpf_perf_event_output, bpf_get_stackid, bpf_get_current_task, bpf_current_task_under_cgrouop, bpf_get_numa_node_id, bpf_probe_read_str, bpf_get_socket_cookie, bpf_perf_event_read_value, bpf_get_stack, bpf_get_current_cgrouop_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_sk_storage_get, bpf_sk_storage_delete, bpf_send_signal, bpf_skb_output, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_send_signal_thread, bpf_jiffies64, bpf_get_ns_current_pid_tgid, bpf_xdp_output, bpf_get_current_ancestor_cgrouop_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_skc_to_tcp6_sock, bpf_skc_to_tcp_sock, bpf_skc_to_tcp_timewait_sock, bpf_skc_to_tcp_request_sock, bpf_skc_to_udp6_sock, bpf_get_task_stack, bpf_d_path, bpf_copy_from_user, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_task_storage_get, bpf_task_storage_delete, bpf_get_current_task_btf, bpf_sock_from_file, bpf_for_each_map_elem, bpf_snprintf, bpf_sys_bpf, bpf_btf_find_by_name_kind, bpf_sys_close, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_get_func_ip, bpf_task_pt_regs, bpf_get_branch_snapshot, bpf_skc_to_unix_sock, bpf_kallsyms_lookup_name, bpf_find_vma, bpf_loop, bpf_strncmp, bpf_xdp_get_buff_len, bpf_copy_from_user_task, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_skc_to_mptcp_sock, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

プログラムの種類	利用可能なヘルパー
netfilter	bpf_map_lookup_elem, bpf_map_update_elem, bpf_map_delete_elem, bpf_ktime_get_ns, bpf_get_prandom_u32, bpf_get_smp_processor_id, bpf_tail_call, bpf_get_current_task, bpf_get_numa_node_id, bpf_get_current_cgroup_id, bpf_map_push_elem, bpf_map_pop_elem, bpf_map_peek_elem, bpf_spin_lock, bpf_spin_unlock, bpf_strtol, bpf_strtoul, bpf_probe_read_user, bpf_probe_read_kernel, bpf_probe_read_user_str, bpf_probe_read_kernel_str, bpf_jiffies64, bpf_get_current_ancestor_cgroup_id, bpf_ktime_get_boot_ns, bpf_ringbuf_output, bpf_ringbuf_reserve, bpf_ringbuf_submit, bpf_ringbuf_discard, bpf_ringbuf_query, bpf_snprintf_btf, bpf_per_cpu_ptr, bpf_this_cpu_ptr, bpf_get_current_task_btf, bpf_for_each_map_elem, bpf_snprintf, bpf_timer_init, bpf_timer_set_callback, bpf_timer_start, bpf_timer_cancel, bpf_task_pt_regs, bpf_loop, bpf_strcmp, bpf_kptr_xchg, bpf_map_lookup_percpu_elem, bpf_dynptr_from_mem, bpf_ringbuf_reserve_dynptr, bpf_ringbuf_submit_dynptr, bpf_ringbuf_discard_dynptr, bpf_dynptr_read, bpf_dynptr_write, bpf_dynptr_data, bpf_ktime_get_tai_ns, bpf_user_ringbuf_drain, bpf_cgrp_storage_get, bpf_cgrp_storage_delete

表7.3 利用可能なマップの種類

マップの種類	Available
ハッシュ	はい
array	はい
prog_array	はい
perf_event_array	はい
percpu_hash	はい
percpu_array	はい
stack_trace	はい
cgroup_array	はい
lru_hash	はい
lru_percpu_hash	はい
lpm_trie	はい
array_of_maps	はい
hash_of_maps	はい

マップの種類	Available
devmap	はい
sockmap	はい
cpumap	はい
xskmap	はい
sockhash	はい
cgroup_storage	はい
reuseport_sockarray	はい
percpu_cgroup_storage	はい
queue	はい
stack	はい
sk_storage	はい
devmap_hash	はい
struct_ops	はい
ringbuf	はい
inode_storage	はい
task_storage	はい
bloom_filter	はい
user_ringbuf	はい
cgrp_storage	はい

第8章 バグ修正

ここでは、ユーザーに重大な影響を与えるバグで、Red Hat Enterprise Linux 9.4 で修正されたものについて説明します。

8.1. インストーラーおよびイメージの作成

Anaconda は、Installation Destination 画面にマルチパスストレージデバイスの WWID 識別子を表示します

以前、Anaconda は、マルチパスストレージデバイスのデバイス番号、WWPN、LUN などの詳細を表示していませんでした。その結果、Installation Destination > Add a disk 画面から正しいインストール先を選択することは困難でした。この更新により、Anaconda がマルチパスストレージデバイスの WWID 識別子を表示するようになりました。その結果、高度なストレージデバイスの画面で、必要なインストール先を簡単に識別して選択できるようになりました。

Jira:RHEL-11384^[1]

インストーラーはキックスタートファイルで追加のタイムゾーン定義を受け入れるようになりました

Anaconda は、タイムゾーンの選択を検証する方法を、より制限的な別の方法に切り替えました。このため、日本などの一部のタイムゾーン定義は、以前のバージョンでは受け入れられていたにもかかわらず、有効ではなくなりました。これらの定義を含むレガシーキックスタートファイルを更新する必要があります。更新しない場合、デフォルトで **Americas/New_York time** ゾーンになっていました。

有効なタイムゾーンのリストは、以前は **pytz** Python ライブラリーの **pytz.common_timezones** から取得されていました。この更新により、**timezone** キックスタートコマンドの検証設定が **pytz.all_timezones** を使用するように変更されます。これは **common_timezones** リストのスーパーセットであり、より多くのタイムゾーンを指定できるようになります。この変更により、Red Hat Enterprise Linux 6 用に作成された古いキックスタートファイルでも引き続き有効なタイムゾーンが指定されるようになります。

注記: この変更は **timezone** キックスタートコマンドにのみ適用されます。グラフィカルおよびテキストベースの対話型インターフェイスでのタイムゾーンの選択は変更されません。有効なタイムゾーンが選択されている Red Hat Enterprise Linux 9 の既存のキックスタートファイルは、更新する必要がありません。

Jira:RHEL-13150^[1]

インストーラーが、複数のポートと BOOTIF オプションを使用してボンディングデバイスを正しく作成するようになりました

以前は、複数のポートを持つボンディングネットワークデバイスと **BOOTIF** ブートオプションを使用してインストールを起動した場合、インストールプログラムが誤った接続プロファイルを作成していました。その結果、BOOTIF オプションで使用されるデバイスは、ポートの1つとして設定されていたにもかかわらず、ボンディングデバイスに追加されませんでした。

この更新により、**BOOTIF** ブートオプションの使用時に、インストールプログラムが **initramfs** にプロファイルを作成するようになりました。その結果、指定したすべてのポートが、インストールされたシステムのボンディングデバイスに追加されます。

Jira:RHEL-4766

Anaconda は、インストールイメージの起動に失敗したときに表示される、誤解を招くエラーメッセージを置き換えます

以前は、たとえば **inst.stage2** または **inst.repo** で指定された **stage2** のソースが見つからないことが原因で、インストールプログラムがインストールイメージの起動に失敗した場合、Anaconda は誤解を招く以下のエラーメッセージを表示していました。

```
/run/anaconda/initrd_errors.txt: No such file or directory
```

この更新により、Anaconda は混乱を最小限に抑えるために、適切な警告メッセージを発行します。

[Jira:RHEL-5638](#)

新しいバージョンの **xfsprogs** が **/boot** サイズを縮小しなくなる

以前は、RHEL 9.3 の 5.19 バージョンの **xfsprogs** パッケージにより、**/boot** サイズが縮小していました。その結果、RHEL 9.2 バージョンと比較すると、**/boot** パーティション上の使用可能な領域に違いが生じました。この修正により、すべてのイメージの **/boot** パーティションが 500 MiB ではなく 600 MiB に増加し、**/boot** パーティションはスペースの問題の影響を受けなくなりました。

[Jira:RHEL-7999](#)

8.2. セキュリティー

Libreswan は IPv6 SAN エクステンションを受け入れる

以前は、IPv6 アドレスの **subjectAltName (SAN)** エクステンションを含む証明書を使用して証明書ベースの認証をセットアップすると、IPsec 接続が失敗しました。この更新により、**pluto** デーモンは IPv4 だけでなく IPv6 SAN も受け入れるように変更されました。その結果、証明書に埋め込まれた IPv6 アドレスを ID として IPsec 接続が正しく確立されるようになりました。

[Jira:RHEL-12278](#)

SELinux ポリシーに、**ip vrf** による仮想ルーティングを管理するためのルールが追加されました

ip vrf コマンドを使用して、他のネットワークサービスの仮想ルーティングを管理できます。以前は、**selinux-policy** にはこの用途をサポートするルールは含まれていませんでした。この更新により、SELinux ポリシールールによって、**ip** ドメインから **httpd**、**sshd**、および **named** ドメインへの明示的な遷移が可能になります。これらの遷移は、**ip** コマンドが **setexeccon** ライブラリー呼び出しを使用する場合に適用されます。

[Jira:RHEL-14246^{\[1\]}](#)

unconfined_login が **off** に設定されている場合、SELinux ポリシーは制限のないユーザーの SSH ログインを拒否します。

以前は、**unconfined_login** ブール値が **off** に設定されている場合に、制限のないユーザーが SSH 経由でログインすることを拒否するルールが SELinux ポリシーにありませんでした。その結果、**unconfined_login** が **off** に設定されている場合でも、ユーザーは **SSHD** を使用して制限のないドメインにログインできます。この更新により、SELinux ポリシーにルールが追加され、その結果、**unconfined_login** が **off** の場合、ユーザーは **sshd** 経由で制限なしでログインできなくなります。

[Jira:RHEL-1551](#)

SELinux ポリシーにより **rsyslogd** が制限されたコマンドを実行できる

以前は、SELinux ポリシーには、**rsyslogd** デーモンが **systemctl** などの SELinux 制限コマンドを実行できるようにするルールがありませんでした。その結果、**omprog** ディレクティブの引数として実行さ

れたコマンドは失敗しました。この更新により、SELinux ポリシーにルールが追加され、**omprog** の引数として実行される **/usr/libexec/rsyslog** ディレクトリー内の実行可能ファイルが **syslogd_unconfined_script_t** の制限のないドメインに含まれるようになります。その結果、**omprog** の引数として実行されたコマンドは正常に終了します。

[Jira:RHEL-11174](#)

SELinux ポリシーにより **rsyslogd** が制限されたコマンドを実行できる

以前は、SELinux ポリシーには、**rsyslogd** デーモンが **systemctl** などの SELinux 制限コマンドを実行できるようにするルールがありませんでした。その結果、**omprog** ディレクティブの引数として実行されたコマンドは失敗しました。この更新により、SELinux ポリシーにルールが追加され、**omprog** の引数として実行される **/usr/libexec/rsyslog** ディレクトリー内の実行可能ファイルが **syslogd_unconfined_script_t** の制限のないドメインに含まれるようになります。その結果、**omprog** の引数として実行されたコマンドは正常に終了します。

[Jira:RHEL-10087](#)

kmod が SELinux MLS ポリシーで実行される

以前は、SELinux は **/var/run/tmpfiles.d/static-nodes.conf** ファイルにプライベートタイプを割り当てていませんでした。その結果、**kmod** ユーティリティーは SELinux マルチレベルセキュリティ (MLS) ポリシーで動作しない可能性があります。この更新により、SELinux ポリシーに **/var/run/tmpfiles.d/static-nodes.conf** の **kmod_var_run_t** ラベルが追加され、その結果、**kmod** が SELinux MLS ポリシーで正常に実行されるようになりました。

[Jira:RHEL-1553](#)

selinux-autorelabel が SELinux MLS ポリシーで実行される

以前は、SELinux ポリシーは **/usr/libexec/selinux/selinux-autorelabel** ユーティリティーにプライベートタイプを割り当てていませんでした。その結果、**selinux-autorelabel.service** は SELinux マルチレベルセキュリティ (MLS) ポリシーで動作しなくなる可能性があります。この更新により、**semanage_exec_t** ラベルが **/usr/libexec/selinux/selinux-autorelabel** に追加され、その結果、**selinux-autorelabel.service** が SELinux MLS ポリシーで正常に実行されるようになりました。

[Jira:RHEL-14289](#)

/bin = /usr/bin ファイルのコンテキスト同等性ルールが SELinux ポリシーに追加される

以前は、SELinux ポリシーには **/bin = /usr/bin** ファイルコンテキスト同等性ルールが含まれていませんでした。その結果、**restorecond** デーモンは正しく動作しませんでした。この更新により、ポリシーに不足しているルールが追加され、その結果、**restorecond** は SELinux enforcing モードで正しく動作するようになりました。

重要

この変更は、**/bin** 内のパターンのファイルコンテキスト仕様を使用するすべてのローカルポリシーモジュールをオーバーライドします。

[Jira:RHEL-5032](#)

SELinux ポリシーには追加のサービスとアプリケーションに関するルールが含まれています

selinux-policy パッケージのこのバージョンには追加のルールが含まれています。特に注目すべきは、**sysadm_r** ロールのユーザーは次のコマンドを実行できることです。

- **sudo traceroute** (RHEL-14077)

- **sudo tcpdump** (RHEL-15432)

[Jira:RHEL-15432](#)

SELinux ポリシーが QAT ファームウェアの権限を追加する

以前は、**Intel VT-d** カーネルオプションを有効にして Intel QuickAssist Technology (QAT) を更新すると、SELinux 権限がないために拒否されていました。この更新により、**qat** サービスに権限が追加されます。その結果、QAT を正しく更新できるようになります。

[Jira:RHEL-19051^{\[1\]}](#)

Rsyslog は omprog を使用して特権コマンドを実行できます

以前は、Rsyslog の **omprog** モジュールは、特定の外部プログラム、特に特権コマンドを含むプログラムを実行できませんでした。その結果、特権コマンドを含むスクリプトを **omprog** を介して使用することは制限されていました。この更新では、SELinux ポリシーが調整されました。調整された SELinux ポリシーとの互換性を確保するために、スクリプトを **/usr/libexec/rsyslog** ディレクトリに配置してください。そうすることで、Rsyslog は **omprog** モジュールを使用して、特権コマンドを含むスクリプトを実行できます。

[Jira:RHEL-5196](#)

semanage fcontext コマンドがローカルの変更を並べ替えなくなりました

semanage fcontext -l -C コマンドは、**file_contexts.local** ファイルに保存されているローカルファイルコンテキストの変更をリスト表示します。**restorecon** ユーティリティーは、**file_contexts.local** 内のエントリーを最も新しいエントリーから最も古いエントリーへと順番に処理します。以前の **semanage fcontext -l -C** は、エントリーを間違った順序でリスト表示していました。処理順序とリスト表示順序の不一致により、SELinux ルールの管理時に問題が発生していました。この更新により、**semanage fcontext -l -C** は、最も古いルールから最も新しいルールへと、期待どおりの正しい順序でルールを表示するようになりました。

[Jira:RHEL-25263^{\[1\]}](#)

オフセット付きの CardOS 5.3 カードが OpenSC で問題を引き起こさなくなりました

以前は、単一の PKCS #15 ファイルの異なるオフセットに証明書を保存する一部の CardOS 5.3 カードでは、ファイルのキャッシュが正しく機能しませんでした。これは、ファイルのキャッシュがファイルのオフセット部分を無視し、結果としてキャッシュのオーバーライドが繰り返され、ファイルのキャッシュから無効なデータが読み取られることが原因でした。この問題はアップストリームで特定および修正されました。この更新以降、CardOS 5.3 カードはファイルのキャッシュに関して正常に動作するようになりました。

[Jira:RHEL-4079^{\[1\]}](#)

8.3. サブスクリプションの管理

subscription-manager は端末に不要なテキストを保持しなくなる

RHEL 9.1以降、**subscription-manager** は操作の処理中に進行状況情報を表示します。以前は、一部の言語 (通常は非ラテン語) では、操作の終了後に進行状況メッセージがクリーンアップされませんでした。この更新により、操作の終了時にすべてのメッセージが適切にクリーンアップされます。

以前に進行状況メッセージを無効にしたことがある場合は、次のコマンドを入力して再度有効にできます。

```
# subscription-manager config --rhsm.progress_messages=1
```

Bugzilla:2136694^[1]

8.4. ソフトウェア管理

librhsm ライブラリーは、**librhsm** がコンテナ内で実行されている場合に、正しい `/etc/rhsm-host` 接頭辞を返すようになりました

librhsm がコンテナ内で実行されている場合、**librhsm** ライブラリーは、CA 証明書へのパス接頭辞を `/etc/rhsm` から `/etc/rhsm-host` パスに書き換えます。以前は、文字列操作の間違いにより、**librhsm** は `/etc/rhsm-host-host` という誤った接頭辞を返していました。この更新により、この問題は修正され、**librhsm** ライブラリーが `/etc/rhsm-host` という正しい接頭辞を返すようになりました。

[Jira:RHEL-14224](#)

systemd は **librepo** が作成した `/run/user/0` ディレクトリーを正しく管理するようになりました

以前は、`root` でログインする前に Insights クライアントから **librepo** 関数が呼び出されると、`/run/user/0` ディレクトリーが間違った SELinux コンテキストタイプで作成される場合がありました。これにより、`root` からのログアウト後に **systemd** がディレクトリーをクリーンアップすることができませんでした。

この更新により、**librepo** パッケージは、SELinux ポリシーで定義されたデフォルトのファイルシステムのラベル付けルールに従って、デフォルトの作成タイプを設定するようになりました。その結果、**systemd** は **librepo** が作成した `/run/user/0` ディレクトリーを正しく管理するようになりました。

[Jira:RHEL-11240](#)

systemd は **libdnf** が作成した `/run/user/0` ディレクトリーを正しく管理するようになりました

以前は、`root` でログインする前に Insights クライアントから **libdnf** 関数が呼び出されると、`/run/user/0` ディレクトリーが間違った SELinux コンテキストタイプで作成される場合がありました。これにより、`root` からのログアウト後に **systemd** がディレクトリーをクリーンアップすることができませんでした。

この更新により、**libdnf** パッケージは、SELinux ポリシーで定義されたデフォルトのファイルシステムのラベル付けルールに従って、デフォルトの作成タイプを設定するようになりました。その結果、**systemd** は **libdnf** が作成した `/run/user/0` ディレクトリーを正しく管理するようになりました。

[Jira:RHEL-11238](#)

dnf needs-restarting --reboothint コマンドは、CPU マイクロコードを更新するために再起動を推奨するようになりました

CPU マイクロコードを完全に更新するには、システムを再起動する必要があります。以前は、更新された CPU マイクロコードを含む **microcode_ctl** パッケージをインストールしたときに、**dnf needs-restarting --reboothint** コマンドは再起動を推奨しませんでした。この更新により、この問題は修正され、**dnf needs-restarting --reboothint** は CPU マイクロコードを更新するために再起動を推奨するようになりました。

[Jira:RHEL-4600](#)

8.5. シェルおよびコマンドラインツール

top -u コマンドは、プロセスをメモリー順に並べ替えるときに、少なくとも1つのプロセスを表示するようになりました

以前は、**-u <user>** パラメーターを指定して **top** コマンドを実行する際に、**user** がコマンドを実行しているユーザーと異なっていた場合、プロセスをメモリー順に並べ替えるために **M** キーを押すとすべてのプロセスが消えていました。この更新により、**top** コマンドは、プロセスをメモリー順に並べ替えるときに、少なくとも1つのプロセスを表示するようになりました。



注記

カーソルの位置を保持するため、すべてのプロセスが表示されるわけではありません。結果を上スクロールすると、残りのプロセスが表示されます。

[Jira:RHEL-16278](#)

BIOS と UEFI の両方のブートローダーがインストールされている場合に ReaR が BIOS ブートローダーの存在を確認するようになりました

以前は、ハイブリッドブートローダーセットアップ (**UEFI** と **BIOS**) で、**UEFI** を使用して起動すると、Relax-and-Recover (ReaR) によって **UEFI** ブートローダーのみが復元され、**BIOS** ブートローダーは復元されませんでした。これにより、システムに **GUID Partition Table** (GPT) と **BIOS** ブートパーティションは存在するものの、**BIOS** ブートローダーは存在しない状態が発生していました。この状況では、ReaR はレスキューイメージの作成に失敗し、**rear mkbackup** または **rear mkrescue** コマンドを使用してバックアップまたはレスキューイメージを作成しようとすると、次のエラーメッセージが表示されて失敗していました。

```
ERROR: Cannot autodetect what is used as boot loader, see default.conf about 'BOOTLOADER'.
```

この更新により、ReaR が **UEFI** と **BIOS** の両ブートローダーの存在を確認してそれらを復元するようになり、**GPT** の **BIOS** ブートパーティションを持つシステムで **BIOS** ブートローダーが検出されない場合でも失敗しなくなりました。その結果、**UEFI** と **BIOS** のハイブリッドブートローダーセットアップを備えたシステムで、複数回のバックアップおよび復元が可能になりました。

[Jira:RHEL-16864^{\[1\]}](#)

ReaR はリカバリー中に logbsize、sunit、swidth マウントオプションを使用しなくなりました

以前は、**MKFS_XFS_OPTIONS** 設定を使用して、元のパラメーターとは異なるパラメーターで **XFS** ファイルシステムを復元する場合、Relax-and-Recover (ReaR) は、元のファイルシステムには適用できるものの復元されたファイルシステムには適用できないマウントオプションを使用して、このファイルシステムをマウントしていました。その結果、ReaR が **mount** コマンドを実行したときに、ディスクレイアウトの再作成が失敗し、次のエラーメッセージが表示されていました。

```
wrong fs type, bad option, bad superblock on and missing codepage or helper program, or other error.
```

カーネルログには、次のいずれかのメッセージが表示されていました。

```
logbuf size must be greater than or equal to log stripe size
```

```
alignment check failed: sunit/swidth vs. agsize
```

この更新により、ReaR は再作成された **XFS** ファイルシステムをマウントするときに、**logbsize**、**sunit**、および **swidth** マウントオプションを使用しなくなります。その結果、**MKFS_XFS_OPTIONS** 設定を使用すると、ディスクレイアウトの再作成が成功します。

Jira:RHEL-10478^[1]

シンプルのメタデータサイズが小さいシステムで ReaR リカバリーが失敗しなくなりました

以前は、ReaR は、シンプルを含む LVM ボリュームグループのレイアウトを保存するときに、プールメタデータボリュームのサイズを保存していませんでした。システムがデフォルト以外のプールメタデータサイズを使用していた場合でも、ReaR はリカバリー時にデフォルトサイズでプールを再作成していました。

その結果、元のプールメタデータサイズがデフォルトサイズよりも小さく、ボリュームグループに空き領域がない場合、システムのリカバリー時にレイアウトの再作成が失敗し、次の例のようなメッセージがログに記録されていました。

```
Insufficient free space: 230210 extents needed, but only 230026 available
```

または、以下を実行します。

```
Volume group "vg" has insufficient free space (16219 extents): 16226 required.
```

この更新により、リカバリー後のシステムが、元のシステムと同サイズのメタデータボリュームを含むようになります。その結果、シンプルのメタデータサイズが小さく、ボリュームグループに追加の空き領域がないシステムのリカバリーが、正常に完了します。

Jira:RHEL-6984

ReaR は、レスキューシステムとリカバリーされたシステムで NetBackup の **bprestore** コマンドからのログを保存するようになりました

以前は、NetBackup 統合 (**BACKUP=NBU**) を使用する場合、ReaR はリカバリー時に、**bprestore** コマンドからのログを終了時に削除されるディレクトリーに追加していました。さらに、ReaR は、コマンドによって生成された追加のログを、リカバリーされたシステムの **/usr/opensv/netbackup/logs/bprestore/** ディレクトリーに保存していませんでした。

その結果、リカバリー時に **bprestore** コマンドが失敗した場合は、**-d** または **-D** オプションを指定して **rear recover** コマンドを実行しない限り、ログは削除されていました。さらに、リカバリーが正常に完了した場合でも、**/usr/opensv/netbackup/logs/bprestore/** ディレクトリー配下のログは再起動後に失われ、調査できませんでした。

この更新により、ReaR は **bprestore** コマンドのログを、レスキューシステムの **/var/lib/rear/restore** ディレクトリーに保存するようになりました。ログは、**rear recover** コマンドが終了した後も、レスキューシステムが再起動するまで保持されます。システムがリカバリーされた場合、さらなる調査が必要な場合に備えて、**/usr/opensv/netbackup/logs/bprestore/** のすべてのログが **/var/lib/rear/restore** のログとともに **/var/log/rear/recover/restore** ディレクトリーにコピーされます。

Jira:RHEL-17393

設定ファイルで **TMPDIR** 変数が設定されている場合、ReaR はリカバリー中に失敗しなくなりました

以前は、ReaR のデフォルト設定ファイル **/usr/share/rear/conf/default.conf** には次の手順が記載されていました。


```
# To have a specific working area directory prefix for Relax-and-Recover
# specify in /etc/rear/local.conf something like
#
# export TMPDIR="/prefix/for/rear/working/directory"
#
# where /prefix/for/rear/working/directory must already exist.
# This is useful for example when there is not sufficient free space
# in /tmp or $TMPDIR for the ISO image or even the backup archive.
```

上記の手順は正しく機能しませんでした。これは、**TMPDIR** 変数がレスキュー環境で同じ値を持っていたためです。**TMPDIR** 変数で指定されたディレクトリーがレスキューイメージに存在しない場合、この値は不適切でした。

その結果、レスキューイメージを起動したときに、**/etc/rear/local.conf** ファイルで **TMPDIR** を設定してエクスポートすると、次のエラーが発生しました。

```
mktemp: failed to create file via template '/prefix/for/rear/working/directory/tmp.XXXXXXXXXX': No
such file or directory
cp: missing destination file operand after '/etc/rear/mappings/mac'
Try 'cp --help' for more information.
No network interface mapping is specified in /etc/rear/mappings/mac
```

または、**rear recover** の実行中に次のエラーが発生し、その後キャンセルしていました。

```
ERROR: Could not create build area
```

この更新により、ReaR がレスキュー環境の **TMPDIR** 変数をクリアするようになりました。ReaR は、**/etc/rear/local.conf** で変数が設定されているのかも検出し、変数が設定されている場合は警告を出力します。**/usr/share/rear/conf/default.conf** 内のコメントは変更され、**TMPDIR** を **/etc/rear/local.conf** で設定するのではなく、**rear** を実行する前に環境内で設定してエクスポートするように指示するようになりました。

/etc/rear/local.conf でコマンド **export TMPDIR=...** を使用すると、ReaR は次の警告を出力します。

```
Warning: Setting TMPDIR in a configuration file is deprecated. To specify a working area directory
prefix, export TMPDIR before executing 'rear'
```

その結果、説明した設定でリカバリーが成功します。

/etc/rear/local.conf などの設定ファイルで **TMPDIR** を設定することが非推奨となりました。この機能は今後のリリースで削除される予定です。代わりに、**/etc/rear/local.conf** から当該設定を削除し、ReaR を呼び出す前に環境内で **TMPDIR** を設定およびエクスポートすることを推奨します。

[Jira:RHEL-24847](#)

8.6. ネットワーク

wwan_hwsim は **kernel-modules-internal** パッケージに含まれるようになりました

wwan_hwsim カーネルモジュールは、ワイヤレス広域ネットワーク (WWAN) デバイスを使用するさまざまなネットワークシナリオをシミュレーションおよびテストするためのフレームワークを提供します。以前は、**wwan_hwsim** は **kernel-modules-extra** パッケージの一部でした。しかし、このリリースでは、同様の指向の他のユーティリティーが含まれる **kernel-modules-internal** パッケージに移動しました。PCI モデムの WWAN 機能はまだ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。

Jira:RHEL-24618^[1]

xdp-loader features コマンドが期待通りに動作するようになりました

xdp-loader ユーティリティーは、**libbpf** の以前のバージョンに対してコンパイルされていました。その結果、**xdp-loader features** はエラーで失敗していました。

```
Cannot display features, because xdp-loader was compiled against an old version of libbpf without support for querying features.
```

このユーティリティーは、正しい **libbpf** バージョンに対してコンパイルされるようになりました。その結果、コマンドは期待どおりに動作するようになりました。

Jira:RHEL-3382

Mellanox ConnectX-5 アダプターが DMFS モードで動作する

以前は、イーサネットスイッチデバイスドライバーモデル (**switchdev**) モードを使用しているときに、**ConnectX-5** アダプターのデバイス管理フローステアリング (**DMFS**) モードで設定されていると、**mlx5** ドライバーが失敗していました。したがって、以下のエラーメッセージが表示されていました。

```
mlx5_core 0000:5e:00.0: mlx5_cmd_out_err:780:(pid 980895):
DELETE_FLOW_TABLE_ENTRY(0x938) op_mod(0x0) failed, status bad resource(0x5), syndrome
(0xabe70a), err(-22)
```

その結果、**ConnectX-5** アダプターのファームウェアバージョンを 16.35.3006 以降に更新すると、エラーメッセージは表示されなくなります。

Jira:RHEL-9897^[1]

8.7. カーネル

crash がバージョン 8.0.4 にリベース

crash ユーティリティーがバージョン 8.0.4 にアップグレードされ、複数のバグが修正されました。主な修正は次のとおりです。

- カーネルパニック中にパニックが発生していない CPU が停止に失敗した場合の、セグメンテーション違反を修正しました。
- **panic_on_oops** カーネルパラメーターが無効な場合、重大なエラーが生じててもカーネルパニックが発生していませんでした。
- **crash** ユーティリティーは、**CONFIG_SLAB_FREELIST_HARDENED=y** 設定オプションでコンパイルされたカーネルの、ハッシュ化されたフリーリストポインターを適切に解決しませんでした。
- カーネルモジュールのメモリーレイアウト用語を変更しました。この変更により、**crash** ユーティリティーのメモリー関連の要素をより適切に示すために、**module_layout** が **module_memory** に置き換われました。この変更を行わないと、**crash** はセッションを開始できず、次のようなエラーメッセージが表示されます。

```
crash: invalid structure member offset: module_core_size
FILE: kernel.c LINE: 3787 FUNCTION: module_init()
```

Jira:RHEL-9009

tuna は、必要に応じて GUI を起動します

以前は、サブコマンドなしで **tuna** ユーティリティーを実行すると、GUI が起動していました。ディスプレイがある場合、この動作は望ましいものでした。逆の場合、ディスプレイのないマシンの **tuna** は適切に終了しませんでした。この更新により、**tuna** はディスプレイがあるかどうかを検出し、それに応じて GUI を起動するかどうかを決定するようになりました。

Jira:RHEL-8859^[1]

Intel TPM チップが正しく検出されるようになる

以前は、AMD Trusted Platform Module (TPM) チップのバグ修正の副次的影響が Intel TPM チップにも影響を及ぼしていました。その結果、RHEL は特定の Intel TPM チップを検出できませんでした。

この更新では、AMD TPM のバグ修正が改訂されました。その結果、RHEL は Intel TPM チップを正しく検出するようになりました。

Jira:RHEL-18985^[1]

VMD が有効になっている場合、RHEL は以前は NVMe ディスクを認識できなかった

ドライバーをリセットまたは再アタッチすると、Volume Management Device (VMD) ドメインは以前はソフトリセットされませんでした。その結果、ハードウェアはデバイスを適切に検出して列挙できませんでした。この更新により、特にサーバーをリセットするときや仮想マシンを操作するときに、VMD が有効になっているオペレーティングシステムが NVMe ディスクを正しく認識するようになりました。

Bugzilla:2128610^[1]

8.8. ファイルシステムおよびストレージ

multipathd は、キューに入れられた未処理の I/O を持つデバイスを正常に削除できるようになりました

以前は、**multipathd** コマンドは、デバイスを削除する前に **queue_if_no_path** パラメーターを無効にしませんでした。このような動作は、キューに入れられた未処理の I/O が、パーティションデバイスではなくマルチパスデバイス自体に対して存在する場合を除いて不可能でした。結果として、**multipathd** がハングし、マルチパスデバイスを維持できなくなっていました。この更新により、**multipathd** は、**multipath -F**、**multipath -f <device>**、**multipathd remove maps**、**multipathd remove map <device>** などの削除コマンドを実行する前に、キューイングを無効にするようになりました。その結果、**multipathd** は、キューに入れられた未処理の I/O を持つデバイスを正常に削除できるようになりました。

Jira:RHEL-4998^[1]

dm-crypt および dm-verity デバイスの

no_read_workqueue、no_write_workqueue、try_verify_in_taskle オプションは一時的に無効になっています

以前は、**no_read_workqueue** または **no_write_workqueue** オプションを使用して作成された **dm-crypt** デバイスと、**try_verify_in_tasklet** オプションを使用して作成された **dm-verity** デバイスによってメモリー破損が発生していました。その結果、ランダムにカーネルメモリーが破損し、さまざまなシ

ステムの問題が発生しました。この更新により、これらのオプションは一時的に無効になります。この修正により、一部のワークロードで **dm-verity** と **dm-crypt** のパフォーマンスが低下する可能性があることに注意してください。

Jira:RHEL-23572^[1]

Multipathd はデバイスが誤って I/O をキューイングしていないかチェックするようになりました

以前は、次の条件下では、マルチパスデバイスが (失敗するように設定されていたにもかかわらず) I/O のキューイングを再開していました。

- マルチパスデバイスの **queue_if_no_paths** パラメーターが複数回の再試行に設定されていた。
- 動作中のパスがなく、I/O をキューイングしなくなったパスデバイスが、マルチパスデバイスから削除された。

今回の更新により、問題が修正されました。その結果、キューイングが無効化され、使用可能なパスがない状態でパスが削除された場合、マルチパスデバイスは I/O のキューイングを再開しなくなりました。

Jira:RHEL-17234^[1]

nvmf_log_connect_error からの重複エントリーの削除

以前は、重複コミットマージエラーにより、**nvmf_log_connect_error** カーネル関数でログメッセージが繰り返されていました。その結果、ファブリックに接続された不揮発性メモリーエクスプレス (NVMe) デバイスにカーネルが接続できなかった場合、**Connect command failed** メッセージが 2 回表示されていました。この更新により、重複したログメッセージがカーネルから削除され、エラーごとに 1 つのログメッセージのみが表示されるようになりました。

Jira:RHEL-21545^[1]

名前空間の追加や削除時にカーネルがクラッシュしなくなりました

以前は、NVMe 名前空間が素早く追加されたり削除されたりすると、名前空間をプローブするために使用される連続したコマンドの間に、名前空間が消えていました。特定のケースでは、ストレージ配列は **invalid namespace** エラーを返さず、代わりにゼロで埋められたバッファを返しました。その結果、**divide-by-zero** によってカーネルがクラッシュしました。この更新により、カーネルは、ストレージに対して発行された Identify Namespace データ構造に対する応答からのデータを検証するようになりました。その結果、カーネルはクラッシュしなくなりました。

Jira:RHEL-14751^[1]

データデバイスの新しく割り当てられたセクションが適切に整列されるようになりました

以前は、Stratis プールが拡張されると、プールの新しいリージョンを割り当てることが可能でした。しかし、新しく割り当てられたリージョンは、以前に割り当てられたリージョンと正しく整列していませんでした。その結果、パフォーマンスが低下し、また **sysfs** 内の Stratis シンプルの **alignment_offset** ファイルにゼロ以外のエントリーが発生する場合があります。この更新により、プールが拡張されると、データデバイスの新しく割り当てられたリージョンが、以前に割り当てられたリージョンと適切に整列するようになりました。その結果、パフォーマンスの低下が発生せず、**sysfs** 内の Stratis シンプルの **alignment_offset** ファイルにゼロ以外のエントリーが発生しなくなりました。

Jira:RHEL-16736

`/etc/fstab` にマウントポイントとして NVMe-FC デバイスを追加すると、システムが正常に起動する

以前は、`nvme-cli nvmf-autoconnect systemd` サービスの既知の問題により、Non-volatile Memory Express over Fibre Channel (NVMe-FC) デバイスを `/etc/fstab` ファイルのマウントポイントとして追加すると、システムが起動できませんでした。その結果、システムは緊急モードに入っていました。この更新により、NVMe-FC デバイスをマウントした際に、システムが問題なく起動するようになりました。

Jira:RHEL-8171^[1]

オペレーティングシステムのインストール中に LUN が表示されるようになる

以前は、特に iSCSI iBFT (Boot Firmware Table) に保存されている CHAP (Challenge-Handshake Authentication Protocol) 認証による iSCSI ハードウェアオフロードが関係する場合、システムはファームウェアソースからの認証情報を使用していませんでした。その結果、インストール中に iSCSI ログインが失敗しました。

udisks2-2.9.4-9.el9 ファームウェア認証の修正により、この問題は解決され、インストールおよび初回起動時に LUN が表示されるようになりました。

Bugzilla:2213769^[1]

8.9. 高可用性およびクラスター

`--force` を指定しない `tls` および `keep_active_partition_tie_breaker` クォーラムデバイスオプションの設定

以前は、クォーラムデバイスを設定するときに、`--force` オプションを指定せずにクォーラムデバイスモデル `net` の `tls` および `keep_active_partition_tie_breaker` オプションを設定することができませんでした。この更新により、これらのオプションを設定する際に `--force` を指定する必要がなくなりました。

Jira:RHEL-7746

クローンリソースおよびバンドルリソースの移動および禁止に関する問題が修正されました

このバグ修正により、バンドルされたリソースとクローンリソースの移動に関する 2 つの制限が解消されます。

- ユーザーがバンドルされたリソースをバンドルから移動しようとしたり、バンドル内での実行を禁止しようとしたりすると、`pcs` が制約を作成していました。しかし、その制約は効果がありませんでした。このため、エラーメッセージが表示されて移動が失敗していました。この修正により、`pcs` は、バンドルされたリソースのバンドルからの移動や、バンドルされたリソースの禁止を許可しなくなりました。また、バンドルされたリソースをバンドルから移動できないことを示すエラーメッセージを出力するようになりました。
- ユーザーがバンドルリソースまたはクローンリソースを移動しようとする、バンドルリソースまたはクローンリソースを移動できないことを示すエラーメッセージが表示され、`pcs` が終了していました。この修正により、`move` コマンドの検証が緩和されます。クローンリソースおよびバンドルリソースを移動できるようになりました。クローンリソースを移動する場合、クローンのインスタンスが複数実行されている場合は、宛先ノードを指定する必要があります。移動できるのはレプリカが 1 つのバンドルのみです。

Jira:RHEL-7744

`pcs status` コマンドの出力に、期限切れの制約に関する警告が表示されなくなりました

以前は、クラスターリソースを移動すると一時的な場所の制約が作成され、制約の有効期限が切れた後でも **pcs status** コマンドによって警告が表示されていました。この修正により、**pcs status** コマンドは期限切れの制約を除外し、コマンド出力に警告メッセージが生成されなくなりました。

[Jira:RHEL-7669](#)

SBD フェンシングで必要な場合、**auto_tie_breaker** クォーラムオプションを無効にできなくなりました

以前は、SBD フェンシングが正しく機能するためにクラスター設定で **auto_tie_breaker** クォーラムオプションが必要な場合でも、**pcs** でこのオプションを無効にすることができました。この修正により、SBD フェンシングで **auto_tie_breaker** オプションを有効にする必要があるシステムで **auto_tie_breaker** を無効にしようとすると、**pcs** はエラーメッセージを生成するようになりました。

[Jira:RHEL-7730](#)

8.10. 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

DAV リポジトリの場所が正規表現の一致を使用して設定されている場合に **httpd** が正常に動作します

以前は、Apache HTTP Server で正規表現の一致 (**LocationMatch** など) を使用して Distributed Authoring and Versioning (DAV) リポジトリが設定されている場合、**mod_dav httpd** モジュールがパス名からリポジトリのルートを持定できませんでした。その結果、サードパーティープロバイダー (たとえば、サブバージョンの **mod_dav_svn** モジュール) からのリクエストを **httpd** が処理できませんでした。

この更新により、**httpd.conf** ファイルの新しい **DevBasePath** ディレクティブを使用して、リポジトリのルートパスを指定できるようになりました。以下に例を示します。

```
<LocationMatch "^/repos/">
  DAV svn
  DavBasePath /repos
  SVNParentPath /var/www/svn
</LocationMatch>
```

これにより、DAV リポジトリの場所が正規表現の一致を使用して設定されている場合でも、**httpd** はリクエストを正常に処理します。

[Jira:RHEL-6600](#)

8.11. コンパイラーおよび開発ツール

システムアップグレードの中断後に **ldconfig** がクラッシュしなくなりました

以前は、システムアップグレードの中断後、**/usr/lib64** ディレクトリに残された不完全な共有オブジェクトを処理する際に、**ldconfig** ユーティリティがセグメンテーション違反で予期せず終了していました。この更新により、**ldconfig** はシステムのアップグレード中に書き込まれた一時ファイルを無視するようになりました。その結果、システムアップグレードの中断後に **ldconfig** がクラッシュしなくなりました。

[Jira:RHEL-14383](#)

glibc は **malloc** アリーナのチューニングに設定されたプロセッサの数を使用するようになる

以前は、**glibc** はスレッドごとの CPU アフィニティマスクを使用して、**malloc** の最大アリーナ数を調整していました。その結果、スレッドアフィニティマスクをシステム内の CPU の小さなサブセットに制限すると、パフォーマンスが低下する可能性があります。

glibc は、最大アリーナ数を決定するために設定された CPU の数を使用するように変更されました。その結果、スレッドごとの CPU アフィニティマスクが制限された状態で実行されている場合でも、アプリケーションはより多くのアリーナを使用するため、パフォーマンスの低下は発生しなくなります。

[Jira:RHEL-17157^{\[1\]}](#)

依存関係のサイクルに関係する共有オブジェクトに **dlclose** を使用するアプリケーションとの **glibc** 互換性が向上しました

以前は、**glibc** の **dlclose** 関数を使用して依存関係サイクル内の共有オブジェクトをアンロードすると、そのオブジェクトの ELF デストラクターが、他のすべてのオブジェクトがアンロードされるまで呼び出されない場合があります。ELF デストラクターの実行が遅くなることで、アプリケーションでクラッシュやその他のエラーが発生しました。これは、最初にアンロードした共有オブジェクトの依存関係がすでに初期化解除されていることが原因でした。

この更新では、**glibc** が修正され、他の ELF デストラクターを実行する前に、アンロードする直近のオブジェクトの ELF デストラクターを呼び出すようになりました。その結果、依存関係サイクルに関係する共有オブジェクトに **dlclose** を使用するアプリケーションとの互換性が向上し、クラッシュが発生しなくなりました。

[Jira:RHEL-2491^{\[1\]}](#)

make はディレクトリーの実行を思考しなくなる

以前は、**make** は実行しようとしている実行可能ファイルが、実際に実行可能ファイルであることを確認していませんでした。その結果、パスに実行可能ファイルと同じ名前のディレクトリーが含まれている場合、**make** は代わりにそのディレクトリーを実行しようとしていました。この更新により、**make** は実行可能ファイルを検索するときに追加のチェックを行うようになりました。その結果、**make** はディレクトリーを実行しようとしなくなりました。

[Jira:RHEL-22829](#)

glibc ワイド文字書き込みパフォーマンスの向上

以前は、**glibc** のワイド **stdio** ストリーム実装では、デフォルトのバッファサイズがワイド文字の書き込み操作に十分な大きさであるとは扱われず、代わりに 16 バイトのフォールバックバッファが使用されていたため、パフォーマンスに悪影響がありました。この更新により、バッファ管理が修正され、書き込みバッファ全体が使用されるようになりました。その結果、**glibc** ワイド文字書き込みパフォーマンスが向上します。

[Jira:RHEL-19862^{\[1\]}](#)

glibc **getaddrinfo** 関数が **ncsd** キャッシュ情報を正しく読み取るようになる

以前は、**glibc** の **getaddrinfo** 関数のバグにより、リストアドレス情報構造体に空の要素が返されることがありました。この更新により、**getaddrinfo** 関数が修正され、**ncsd** キャッシュデータを正しく読み取り、変換できるようになり、結果として正しいアドレス情報が返されるようになりました。

[Jira:RHEL-16643](#)

依存関係のサイクルに関係する共有オブジェクトに **dlclose** を使用するアプリケーションとの **glibc** 互換性が向上しました

以前は、**glibc** の **dlclose** 関数を使用して依存関係サイクル内の共有オブジェクトをアンロードすると、そのオブジェクトの ELF デストラクターが、他のすべてのオブジェクトがアンロードされるまで呼び出されない場合があります。ELF デストラクターの実行が遅くなることで、アプリケーションでクラッシュやその他のエラーが発生しました。これは、最初にアンロードした共有オブジェクトの依存関係がすでに初期化解除されていることが原因でした。

この更新では、**glibc** が修正され、他の ELF デストラクターを実行する前に、アンロードする直近のオブジェクトの ELF デストラクターを呼び出すようになりました。その結果、依存関係サイクルに関する共有オブジェクトに **dlclose** を使用するアプリケーションとの互換性が向上し、クラッシュが発生しなくなりました。

[Jira:RHEL-12362](#)

キャッシュ有効期限情報の不一致により **ncsd** の起動が失敗しなくなる

以前は、永続キャッシュファイル内のキャッシュ有効期限情報が一貫していないため、**glibc** Name Service Switch Caching Daemon (**ncsd**) の起動に失敗する可能性があります。この更新により、**ncsd** はタイミング情報が一貫していないキャッシュエントリを削除対象としてマークし、スキップするようになりました。その結果、キャッシュ有効期限情報の不一致により **ncsd** の起動が失敗することがなくなりました。

[Jira:RHEL-3397](#)

一貫して高速な **glibc** スレッドローカルストレージパフォーマンス

以前は、**glibc** ダイナミックリンカーは、**dlopen()** 関数を使用して TLS を持つ共有オブジェクトがロードされた後、特定のスレッドローカルストレージ (TLS) メタデータを調整しなかったため、TLS アクセスが遅くなっていました。この更新により、ダイナミックリンカーは、**dlopen()** 呼び出しによって発生した TLS の変更に対して、TLS メタデータを更新するようになりました。その結果、TLS アクセスは一貫して高速になります。

[Jira:RHEL-2123](#)

8.12. IDENTITY MANAGEMENT

割り当てられたメモリーが操作完了後に解放されるようになりました

以前は、各操作に対して KCM によって割り当てられたメモリーは、接続が閉じるまで解放されませんでした。その結果、接続を開いて同じ接続で多くの操作を実行するクライアントアプリケーションでは、接続が閉じるまで割り当てられたメモリーが解放されないため、メモリーが著しく増加しました。この更新により、操作に割り当てられたメモリーは、操作が完了するとすぐに解放されるようになりました。

[Jira:SSSD-7015](#)

IdM クライアントは、信頼できる AD ユーザーの名前に大文字と小文字が混在している場合でも、当該 AD ユーザーの情報を適切に取得する

以前は、ユーザーの検索または認証を試行した際に、その信頼できる Active Directory (AD) ユーザーの名前に大文字と小文字が混在しており、かつ IdM でオーバーライドが設定されていた場合、エラーが返され、ユーザーは IdM リソースにアクセスできませんでした。

この更新により、大文字と小文字を区別する比較は、大文字と小文字を区別しない比較に置き換えられました。その結果、IdM クライアントは、ユーザー名に大文字と小文字が混在しており、IdM でオーバーライドが設定されている場合でも、AD の信頼済みドメインのユーザーを検索できるようになりました。

[Jira:SSSD-6096](#)

SSSD は、パスワード変更時に猶予ログインが残っていない場合にエラーを正しく返します

以前は、ユーザーの LDAP パスワードの有効期限が切れた場合、猶予ログインが残っていないためにユーザーの最初のバインドが失敗した後でも、SSSD はパスワードの変更を試みていました。しかし、ユーザーに返されたエラーには失敗の理由が示されていませんでした。この更新により、バインドが失敗した場合、パスワード変更要求は中断されるようになりました。SSSD は、猶予ログインが残っていないため、別の方法でパスワードを変更する必要があることを示すエラーメッセージを返します。

[Jira:SSSD-6184](#)

realm leave コマンドを使用したドメインからのシステム削除

以前は、**sssd.conf** ファイルの **ad_server** オプションに複数の名前が設定されている場合、**realm leave** コマンドを実行すると解析エラーが発生し、システムがドメインから削除されませんでした。この更新により、**ad_server** オプションが適切に評価されるようになり、正しいドメインコントローラー名が使用され、システムがドメインから適切に削除されるようになりました。

[Jira:SSSD-6081](#)

KCM は正しい **sssd.kcm.log** ファイルにログを記録します

以前、**logrotate** は Kerberos Credential Manager (KCM) ログファイルを正しくローテーションしていましたが、KCM はログを誤って古いログファイル **sssd_kcm.log.1** に書き込んでいました。KCM を再起動した場合、正しいログファイルが使用されました。この更新により、**logrotate** が呼び出されるとログファイルがローテーションされ、KCM は **sssd_kcm.log** ファイルに正しくログを記録するようになりました。

[Jira:SSSD-6652](#)

realm leave --remove コマンドが認証情報を要求しなくなる

以前は、**realm** ユーティリティーは、**realm leave** 操作を実行するときに、有効な Kerberos チケットが利用可能かどうかを正しく確認していませんでした。その結果、有効な Kerberos チケットが利用可能であったにもかかわらず、ユーザーはパスワードの入力を求められました。この更新により、**realm** は有効な Kerberos チケットがあるかどうかを正しく検証するようになり、**realm leave --remove** コマンドを実行するときにユーザーにパスワードの入力を要求しなくなりました。

[Jira:SSSD-6425](#)

KDC は、一般的な制約付き委任リクエストを処理するときに追加のチェックを実行するようになりました

以前は、Red Hat Enterprise Linux 8 で実行中の KDC が発行した Kerberos チケットの **forwardable** フラグに脆弱性があり、検出されることなく不正な変更を加えることが可能でした。この脆弱性により、特定の権限を持たないユーザーによるなりすまし攻撃が発生する可能性があります。この更新により、KDC は、一般的な制約付き委任リクエストを処理するときに追加のチェックを実行するようになりました。これにより、不正なフラグ変更を検出および拒否し、脆弱性を排除します。

[Jira:RHEL-9984^{\[1\]}](#)

ドメインに対して SID が生成されている場合、forwardable フラグのチェックは無効になります

以前、CVE-2020-17049 の修正を提供する更新では、KDC が一般的な制約付き委任リクエストを処理するときに、Kerberos PAC を使用して、チケットの **forwardable** フラグに関する特定のチェックを実行していませんでした。ただし、PAC は、過去に SID 生成タスクが実行されたドメインでのみ生成されま

す。このタスクは、Red Hat Enterprise Linux (RHEL) 8.5 以降で作成されたすべての IdM ドメインに対しては自動的に実行されますが、古いバージョンで初期化されたドメインでは、このタスクを手動で実行する必要があります。

RHEL 8.4 以前で初期化された IdM ドメインに対して SID 生成タスクを手動で実行しなかった場合、Kerberos チケットから PAC が欠落し、一般的な制約付き委任リクエストがすべて拒否されます。これには、一般的な制約付き委任に依存する IdM の HTTP API が含まれます。

この更新により、ドメインに対して SID が生成されなかった場合、**forwardable** フラグのチェックが無効になります。一般的な制約付き委任に依存するサービス (IdM HTTP API など) は引き続き動作します。ただし、Red Hat では、特にドメインにカスタムの一般的な制約付き委任ルールが設定されている場合は、できるだけ早くドメインで SID 生成タスクを実行することを推奨しています。このタスクが完了するまでは、ドメインは CVE-2020-17049 に対して脆弱なままとなります。

[Jira:RHEL-22313](#)

FIPS モードで IdM Vault 暗号化および復号化に失敗しなくなりました

以前は、IdM Vault は、デフォルトのパディングラッピングアルゴリズムとして OpenSSL RSA-PKCS1v15 を使用していました。しかし、RHEL の FIPS 認定モジュールはいずれも FIPS で承認されたアルゴリズムとして PKCS#1 v1.5 をサポートしていなかったため、IdM Vault は FIPS モードで失敗していました。この更新により、IdM Vault は、RSA-OAEP パディングラッピングアルゴリズムをフォールバックとしてサポートするようになりました。その結果、IdM Vault の暗号化と復号化が FIPS モードで正しく動作するようになりました。

[Jira:RHEL-12143^{\[1\]}](#)

ページ結果検索を中止した後に Directory Server が失敗しなくなりました

以前は、競合状態が、ページ化された結果の検索を中止する際にヒープ破損と Directory Server 障害の原因となっていました。この更新により、競合状態が修正され、Directory Server 障害は発生しなくなりました。

[Jira:RHEL-16830^{\[1\]}](#)

nsslapd-numlisteners 属性値が 2 より大きい場合、Directory Server は失敗しなくなる

以前は、**nsslapd-numlisteners** 属性値が 2 より大きい場合、Directory Server は受け入れられたファイル記述子ではなく、リスニング中のファイル記述子を閉じることがありました。その結果、Directory Server でセグメンテーションエラーが発生してしまいました。この更新により、Directory Server は正しい記述子を閉じ、ポートを正しくリッスンし続けます。

[Jira:RHEL-17175](#)

自動バインド操作が、他の接続で実行される操作に影響を与えなくなる

以前は、自動バインド操作の進行中に、Directory Server は接続時の新しい操作のリッスンを停止していました。この更新により、自動バインド操作は他の接続で実行される操作に影響を与えなくなります。

[Jira:RHEL-5111](#)

IdM クライアントインストーラーは、Idap.conf ファイルで TLS CA 設定を指定しなくなる

以前は、IdM クライアントインストーラーは **Idap.conf** ファイルで TLS CA 設定を指定していました。この更新により、OpenLDAP はデフォルトのトラストストアを使用し、IdM クライアントインストーラーは **Idap.conf** ファイルに TLS CA 設定をセットアップしません。

[Bugzilla:2094673](#)

8.13. WEB コンソール

VNC コンソールがほとんどの解像度で機能するようになる

以前は、特定のディスプレイ解像度で Virtual Network Computing (VNC) コンソールを使用すると、マウスオフセットの問題が発生したり、インターフェイスの一部しか表示されなかったりしました。そのため、VNC コンソールを使用することはできませんでした。

この更新により、問題は修正され、3840 x 2160 ピクセルなどの超高解像度を除き、ほとんどの解像度で VNC コンソールが正しく動作するようになりました。

カーソルの録画した位置と表示位置の間には、小さいオフセットが引き続き存在する可能性があることに注意してください。ただし、これは VNC コンソールのユーザービリティに大きな影響を与えることはありません。

[Bugzilla:2030836](#)

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

SBD の `delay-start` 値が高い場合に、クラスターの起動がタイムアウトしなくなりました

以前は、ユーザーが `ha_cluster` システムロールを使用してクラスターで SBD フェンシングを設定し、`delay-start` オプションを 90 秒に近い値または 90 秒を超える値に設定すると、クラスターの起動がタイムアウトしていました。これは、デフォルトの `systemd` 起動タイムアウトが 90 秒であり、システムが SBD の起動遅延値より前にこの起動タイムアウトに達していたためです。この修正により、`ha_cluster` システムロールは、`delay-start` の値を上回るように `systemd` の `sbd.service` 起動タイムアウトをオーバーライドします。これにより、`delay-start` オプションの値が高い場合でも、システムを正常に起動できるようになります。

[Jira:RHEL-18026^{\[1\]}](#)

ネットワーク ロールが、`0.0.0.0/0` または `::/0` のルーティングルールを検証する

以前は、ルーティングルールで `from:` または `to:` 設定が `0.0.0.0/0` または `::/0` アドレスに設定されていた場合、`network` RHEL システムロールはルーティングルールを設定できず、設定が無効として拒否されていました。この更新により、`network` ロールでは、ルーティングルールの検証で `from:` と `to:` に `0.0.0.0/0` と `::/0` が許可されるようになりました。その結果、ロールは検証エラーを発生させることなくルーティングルールを正常に設定します。

[Jira:RHEL-1683](#)

読み取りスケールクラスターの実行と `mssql-server-ha` のインストールに特定の変数が不要になる

以前は、`mssql` RHEL システムロールを使用して、特定の変数 (`mssql_ha_virtual_ip`、`mssql_ha_login`、`mssql_ha_login_password`、および `mssql_ha_cluster_run_role`) なしで読み取りスケールクラスターを設定すると、ロールは失敗し、“Variable not defined” というエラーメッセージが表示されました。ただし、これらの変数は読み取りスケールクラスターを実行するために必要ではありません。このロールは、読み取りスケールクラスターに必要な `mssql-server-ha` のインストールも試みました。この修正により、これらの変数の要件は削除されました。その結果、読み取りスケールクラスターは、エラーメッセージなしで正常に実行されます。

[Jira:RHEL-3540](#)

kexec_crash_size ファイルがビジー状態のときでも、Kdump システムロールが正常に動作する

`/sys/kernel/kexec_crash_size` ファイルは、クラッシュカーネルメモリーに割り当てられたメモリーリージョンのサイズを提供します。

以前は、`/sys/kernel/kexec_crash_size` ファイルがビジー状態のときに Kdump システムロールが失敗していました。この更新により、システムロールはファイルが利用可能になるとファイルの読み取りを再試行します。その結果、ファイルがビジー状態のときにシステムロールが失敗しなくなりました。

[Jira:RHEL-3353](#)

selinux ロールは **item** ループ変数を使用しなくなりました

以前は、**selinux** RHEL システムロールは **item** ループ変数を使用していました。これにより、別のロールから **selinux** ロールを呼び出したときに、次の警告メッセージが表示される場合があります。

```
[WARNING]: TASK: fedora.linux_system_roles.selinux : Restore SELinux labels on filesystem tree:
The loop variable 'item' is already in use.
You should set the `loop_var` value in the `loop_control` option for the task to something else to avoid
variable collisions and unexpected behavior.
```

このリリースでは、**selinux** ロールは `__selinux_item` をループ変数として使用します。その結果、別のロールから **selinux** ロールを呼び出しても、**item** 変数がすでに使用されているという警告が表示されなくなりました。

[Jira:RHEL-19040](#)

ha_cluster システムロールが **qnetd** ホスト上のファイアウォールを正しく設定するようになりました

以前は、ユーザーが **qnetd** ホストを設定し、**ha_cluster** システムロールを使用して **ha_cluster_manage_firewall** 変数を **true** に設定しても、そのロールによってファイアウォールの高可用性サービスが有効になりませんでした。この修正により、**ha_cluster** システムロールは **qnetd** ホスト上のファイアウォールを正しく設定するようになりました。

[Jira:RHEL-17875](#)

postgresql RHEL システムロールは、正しいバージョンの PostgreSQL をインストールするようになりました。

以前は、RHEL マネージドノードで **postgresql_version: "15"** 変数を定義して **postgresql** RHEL システムロールを実行しようとする、PostgreSQL バージョン 15 ではなくバージョン 13 がインストールされていました。このバグは修正され、**postgresql** ロールは変数に設定されたバージョンをインストールするようになりました。

[Jira:RHEL-5274](#)

keylime_server ロールが **registrar** サービスのステータスを正しく報告する

以前は、**keylime_server** ロール Playbook が誤った情報を提供すると、ロールは誤って開始が成功したと報告していました。この更新により、ロールは誤った情報が提供された場合に失敗を正しく報告するようになり、開いているポートを待機する際のタイムアウトが約 300 秒から約 30 秒に短縮されました。

[Jira:RHEL-15909](#)

podman RHEL システムロールは、ルートレスコンテナに対して `linger` を適切に設定およびキャンセルできるようになりました。

以前は、**podman RHEL システムロールは、ルートレスコンテナに対して `linger` を適切に設定およびキャンセルしていませんでした。**その結果、ルートレスユーザーのシークレットまたはコンテナをデプロイすると、場合によってはエラーが発生し、場合によってはリソースを削除するときに `linger` をキャンセルできませんでした。この更新により、**podman RHEL システムロールは、シークレットまたはコンテナリソースの管理を行う前にルートレスユーザーに対して `linger` が有効になっていることを確認し、管理するシークレットまたはコンテナリソースがなくなったときにルートレスユーザーに対して `linger` がキャンセルされるようにします。**その結果、このロールはルートレスユーザーの残留を正しく管理します。

[Jira:RHEL-22228](#)

`nbde_server` ロールがソケットオーバーライドで動作するようになる

以前は、**`nbde_server` RHEL システムロールは、`tangd` ソケットオーバーライドディレクトリー内の唯一のファイルはカスタムポートの `override.conf` ファイルであると想定していました。**その結果、ポートのカスタマイズがないと、ロールは他のファイルをチェックせずにディレクトリーを削除し、システムはその後の実行でディレクトリーを再作成しました。

このリリースでは、ポートオーバーライドファイルの属性の変更や他のファイルがある場合のディレクトリーの削除を防止するためにロールが修正されました。その結果、**`tangd` ソケットオーバーライドファイルがロールの外部でも管理されていると、ロールは正しく機能します。**

[Jira:RHEL-25508](#)

ボリューム `quadlet` サービス名が失敗しなくなる

以前は、ボリュームサービス名を開始すると、"`Could not find the requested service NAME.volume: host`" のようなエラーが発生しました。この更新により、ボリューム `quadlet` サービス名が **`basename-volume.service`** に変更されました。その結果、ボリュームサービスはエラーなしで開始されます。

詳細は、[ボリュームユニット](#) の man ページを参照してください。

[Jira:RHEL-21401](#)

Ansible はシークレットで使用するために JSON 文字列を保存するようになる

以前は、値がループで使用され、**`data: "{{ value }}"`** に類似した文字列である場合、Ansible は JSON 文字列を対応する JSON オブジェクトに変換していました。その結果、JSON 文字列をシークレットとして渡して値を保持できませんでした。この更新により、**`podman_secret` モジュールに渡すときに、データ値が文字列にキャストされます。**その結果、JSON 文字列はシークレットで使用するためにそのまま保存されます。

[Jira:RHEL-22309](#)

`rhc_auth` にアクティベーションキーが含まれている場合、`rhc` システムロールが登録済みシステムで失敗しなくなる

以前は、**`rhc_auth` パラメーターで指定されたアクティベーションキーを使用して登録済みシステムで Playbook ファイルを実行すると、エラーが発生していました。**この問題は解決されています。**`rhc_auth` パラメーターにアクティベーションキーが提供されていても、すでに登録されているシステムで Playbook ファイルを実行できるようになりました。**

[Bugzilla:2186218](#)

8.15. 仮想化

FIFO スケジューラーを備えた RT 仮想マシンが正しく起動するようになる

以前は、リアルタイム (RT) 仮想マシン (VM) で vCPU スケジューラーに **fifo** 設定を使用するように設定した後、仮想マシンを起動しようとすると応答しなくなりました。代わりに、仮想マシンに **Guest has not initialized the display (yet)** というエラーが表示されました。この更新により、エラーが修正され、説明した状況で vCPU スケジューラー用に **fifo** を設定することが期待どおりに動作するようになりました。

Jira:RHEL-2815^[1]

ダンプの失敗によって IBM Z VMs with Secure Execution がブロックされなくなりました。

以前は、IBM Z virtual machine (VM) with Secure Execution のダンプが失敗すると、仮想マシンは一時停止状態のままになり、実行がブロックされていました。たとえば、ディスクに十分なスペースがない場合、**virsh dump** コマンドを使用して仮想マシンをダンプすることは失敗します。

基礎となるコードが修正され、ダンプ失敗後も Secure Execution 仮想マシンは正常に操作を再開します。

Jira:RHEL-16695^[1]

仮想マシンに RHEL をインストールするための予想されるシステムディスクをインストールプログラムが表示するようになりました

以前は、**virtio-scsi** デバイスを使用して仮想マシンに RHEL をインストールすると、**device-mapper-multipath** のバグにより、これらのデバイスがインストールプログラムに表示されない可能性があります。その結果、インストール時に、シリアルセットを持つデバイスとシリアルセットを持たないデバイスがある場合、シリアルセットを持つすべてのデバイスを **multipath** コマンドが要求していました。このため、仮想マシンに RHEL をインストールするために必要なシステムディスクをインストールプログラムが検出できませんでした。

今回の更新により、**multipath** はシリアルのないデバイスを World Wide Identifier (WWID) を持たないものとして正しく設定し、無視します。インストール時に、**multipath** は **multipathd** がマルチパスデバイスのバインドに使用するデバイスのみを要求し、インストールプログラムは仮想マシンに RHEL をインストールするための予想されるシステムディスクを表示します。

Bugzilla:1926147^[1]

多数のキューを使用しても仮想マシンで障害が発生しなくなりました

以前は、Virtual Trusted Platform Module (vTPM) デバイスが有効で、マルチキュー **virtio-net** 機能が 250 を超えるキューを使用するように設定されている場合、仮想マシン (VM) で障害が発生することがありました。

この問題は、vTPM デバイスの制限が原因で発生していました。この更新により、問題が修正され、250 を超えるキューを使用する、vTPM が有効な仮想マシンが確実に動作するようになりました。

Jira:RHEL-13335^[1]

AMD EPYC CPU を搭載したホストで v2v 変換を行った後、Windows ゲストの起動がより安定します。

virt-v2v ユーティリティーを使用して、Windows 11 または Windows Server 2022 をゲスト OS として使用する仮想マシン (VM) を変換した後、以前は仮想マシンの起動に失敗していました。これは、AMD EPYC シリーズ CPU を使用するホストで発生していました。現在、基礎となるコードが修正され、仮想マシンは説明した状況で期待どおりに起動します。

Bugzilla:2168082^[1]

nodedev-dumpxml は、特定の仲介デバイスの属性を正しくリストします。

この更新より前は、**nodedev-dumpxml** ユーティリティーは、**nodedev-create** コマンドを使用して作成された仲介デバイスの属性を正しくリストしませんでした。この問題は修正され、**nodedev-dumpxml** は、影響を受ける仲介デバイスの属性を適切に表示するようになりました。

[Bugzilla:2143158](#)

virtqemud または **libvirtd** を再起動した後、**virtiofs** デバイスを接続できませんでした

以前は、**virtqemud** サービスまたは **libvirtd** サービスを再起動すると、**virtiofs** ストレージデバイスをホスト上の仮想マシン (VM) に接続できなくなりました。このバグは修正されており、説明されているシナリオで期待どおりに **virtiofs** デバイスをアタッチできるようになりました。

[Bugzilla:2078693](#)

仮想マシンへの Watchdog カードのホットプラグが失敗しなくなる

以前は、使用可能な PCI スロットがない場合、実行中の仮想マシン (VM) に Watchdog カードを追加すると、以下のエラーが発生し、失敗していました。

```
Failed to configure watchdog
ERROR Error attempting device hotplug: internal error: No more available PCI slots
```

今回の更新で問題が修正され、実行中の仮想マシンに Watchdog カードを追加すると、想定どおりに機能するようになりました。

[Bugzilla:2173584](#)

IBM Z 上の **virtio-gpu** で **blob** リソースが正しく動作するようになる

以前は、**virtio-gpu** デバイスは IBM Z システム上の **blob** メモリーリソースと互換性がありませんでした。その結果、IBM Z ホスト上で **virtio-gpu** を使用して仮想マシン (VM) を設定し、**blob** リソースを使用すると、仮想マシンにグラフィカル出力がありませんでした。

この更新により、**virtio** デバイスにオプションの **blob** 属性が追加されました。**blob** を **on** に設定すると、デバイス内の **blob** リソースが使用できるようになります。これにより、**virtio-gpu** デバイスで説明した問題が防止され、ゲストとホスト間のピクセルデータのコピーが削減または排除されるため、ディスプレイパスも高速化されます。**blob** リソースのサポートには QEMU バージョン 6.1 以降が必要であることに注意してください。

[Jira:RHEL-7135](#)

virtio-win ドライバーを再インストールしても、ゲストの DNS 設定がリセットされなくなりました

以前は、Windows ゲストオペレーティングシステムを使用する仮想マシンで、ネットワークインターフェイスコントローラー (NIC) の **virtio-win** ドライバーを再インストールまたはアップグレードすると、ゲストの DNS 設定がリセットされていました。その結果、Windows ゲストのネットワーク接続が失われる場合があります。

この更新により、上記の問題が修正されました。したがって、**virtio-win** の最新バージョンを再インストールまたはアップグレードすると、問題は発生しなくなります。ただし、**virtio-win** の以前のバージョンからアップグレードしても問題は解決されず、Windows ゲストで DNS リセットが引き続き発生する可能性があることに注意してください。

[Jira:RHEL-1860^{\[1\]}](#)

第9章 テクノロジープレビュー

ここでは、Red Hat Enterprise Linux 9 で利用可能なテクノロジープレビュー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。

9.1. インストーラーおよびイメージの作成

RHEL インストール用の NVMe over TCP がテクノロジープレビューとして利用可能になる

このテクノロジープレビューにより、ファームウェアの設定後、NVMe over TCP ボリュームを使用して RHEL をインストールできるようになりました。**Installation Destination** 画面からディスクを追加するときに、**NVMe Fabrics Devices** セクションで NVMe 名前空間を選択できます。

Jira:RHEL-10216^[1]

起動可能な OSTree ネイティブコンテナのインストールがテクノロジープレビューとして利用可能になる

ostreecontainer キックスタートコマンドが、Anaconda 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。このコマンドを使用すると、OCI イメージにカプセル化された OSTree コミットから、オペレーティングシステムをインストールできます。キックスタートインストールを実行する場合、**ostreecontainer** とともに次のコマンドが利用できます。

- graphical、text、または cmdline
- ostreecontainer
- clearpart、zerombr
- autopart
- part
- logvol、volgroup
- reboot および shutdown
- lang
- rootpw
- sshkey
- bootloader **---append** オプションパラメーターを指定する形でのみ使用できます。
- user

user コマンド内でグループを指定する場合、ユーザーアカウントはコンテナイメージ内にすでに存在するグループにのみ割り当てることができます。ここにリストされていないキックスタートコマンドを **ostreecontainer** コマンドと併用することもできますが、それらのコマンドがパッケージベースのインストールで期待どおりに動作する保証はありません。

ただし、次のキックスタートコマンドの **ostreecontainer** との併用はサポートされていません。

- %packages (必要なパッケージがコンテナイメージにすでに存在している必要があります)

- url (インストール用に **stage2** イメージを取得する必要がある場合、たとえば PXE インストールでは、Kickstart ファイル内で **stage2** の URL を指定する代わりに、カーネルで **inst.stage2=** を使用します)
- liveimg
- vnc
- authconfig および authselect (代わりにコンテナイメージに関連する設定を提供します)
- module
- repo
- zipl
- zfc

部分的なキックスタートファイルを使用する対話形式のインストールでは、起動可能な OSTree ネットワークコンテナのインストールはサポートされません。

注記: マウントポイントをカスタマイズする場合は、マウントポイントを **/mnt** ディレクトリー内に定義し、マウントポイントディレクトリーがコンテナイメージの **/var/mnt** 内に存在することを確認する必要があります。

Jira:RHEL-2250^[1]

Anaconda の **bootupd** / **bootupctl** によるブートローダーのインストールおよび設定が、テクノロジープレビューとして利用可能になる

ostreecontainer キックスタートコマンドが Anaconda でテクノロジープレビューとして利用可能になったため、このコマンドを使用して、OCI イメージにカプセル化された OSTree コミットからオペレーティングシステムをインストールできます。Anaconda は、キックスタートで明示的なブートローダー設定がなくても、コンテナイメージに含まれる **bootupd** / **bootupctl** ツールを使用して、ブートローダーのインストールおよび設定を自動的に調整します。

Jira:RHEL-17205^[1]

bootc image builder ツールはテクノロジープレビューとして利用可能になる

現在テクノロジープレビューとして利用可能な **bootc image builder** ツールは、**bootc** コンテナ入力から互換性のあるディスクイメージを簡単に作成してデプロイするためのコンテナとして機能します。**bootc image builder** を使用してコンテナイメージを実行した後、必要なアーキテクチャーのイメージを生成できます。その後、生成されたイメージを仮想マシン、クラウド、またはサーバーにデプロイできます。新しい更新が必要になるたびに **bootc image builder** を使用してコンテンツを再生成するのではなく、**bootc** を使用してイメージを簡単に更新できます。

Jira:RHELDPCS-17468^[1]

新しい **rhel9/bootc-image-builder** コンテナイメージがテクノロジープレビューとして利用可能になる

RHEL のイメージモード用の **rhel9/bootc-image-builder** コンテナイメージには、起動可能なコンテナイメージ (**rhel-bootc** など) を QCOW2、AMI、VMDK、ISO などのさまざまなディスクイメージ形式に変換するイメージビルダーの最小バージョンが含まれています。

Jira:RHELDPCS-17733^[1]

9.2. セキュリティー

gnutls がテクノロジープレビューとして kTLS を使用するようになる

更新された **gnutls** パッケージは、テクノロジープレビューとして、暗号化チャネルでのデータ転送を加速するためにカーネル TLS (kTLS) を使用できます。kTLS を有効にするには、**modprobe** コマンドを使用して **tls.ko** カーネルモジュールを追加し、次の内容を含むシステム全体の暗号化ポリシー用の新しい設定ファイル `/etc/crypto-policies/local.d/gnutls-ktls.txt` を作成します。

```
[global]
ktls = true
```

現在のバージョンは、TLS **KeyUpdate** メッセージによるトラフィックキーの更新をサポートしていません。これは、AES-GCM 暗号スイートのセキュリティーに影響を与えることに注意してください。詳細は、[RFC 7841 - TLS 1.3](#) ドキュメントを参照してください。

Bugzilla:2108532^[1]

io_uring インターフェイスがテクノロジープレビューとして利用可能

io_uring は、新しく効果的な非同期 I/O インターフェイスであり、現在テクノロジープレビューとして利用可能です。デフォルトでは、この機能は無効にされています。このインターフェイスを有効にするには、**kernel.io_uring_disabled** sysctl 変数を次のいずれかの値に設定します。

0

すべてのプロセスは通常どおり **io_uring** インスタンスを作成できます。

1

io_uring の作成は、特権のないプロセスに対しては無効化されています。呼び出しプロセスに **CAP_SYS_ADMIN** 機能による特権が与えられていない限り、**io_uring_setup** は **-EPERM** エラーで失敗します。既存の **io_uring** インスタンスは引き続き使用できます。

2

io_uring の作成は、すべてのプロセスで無効化されています。**io_uring_setup** は常に **-EPERM** で失敗します。既存の **io_uring** インスタンスは引き続き使用できます。これはデフォルト設定です。

この機能を使用するには、匿名 i ノードで **mmap** システムコールを有効にするための SELinux ポリシーの更新バージョンも必要です。

io_uring コマンドパススルーを使用すると、アプリケーションは **nvme** などの基盤となるハードウェアにコマンドを直接発行できます。

Jira:RHEL-11792^[1]

9.3. RHEL FOR EDGE

テクノロジープレビューとして、FDO が SQL バックエンドからの Owner Voucher の保存とクエリーを提供するようになる

このテクノロジープレビューでは、SQL バックエンドから Owner Voucher を保存およびクエリーするために、FDO **manufacturer-server**、**onboarding-server**、および **rendezvous-server** が利用できます。その結果、FDO サーバーのオプションで、認証情報やその他のパラメーターとともに SQL データストアを選択して、Owner Voucher を保存できるようになります。

Jira:RHELDPCS-17752^[1]

9.4. シェルおよびコマンドラインツール

RHEL 9 でテクノロジープレビューとして利用可能な GIMP

GNU Image Manipulation Program (GIMP) 2.99.8 が、テクノロジープレビューとして RHEL 9 で利用できるようになりました。**gimp** パッケージバージョン 2.99.8 は、改善された一連の改良を含むリリース前のバージョンですが、機能のセットが制限され、安定性の保証は保証されません。公式の GIMP 3 のリリース後すぐに、今回のリリース前のバージョンの更新として RHEL 9 に導入されます。

RHEL 9 では、RPM パッケージとして **gimp** を簡単にインストールできます。

[Bugzilla:2047161^{\[1\]}](#)

9.5. インフラストラクチャーサービス

TuneD 用のソケット API がテクノロジープレビューとして利用可能になる

UNIX ドメインソケットを通じて TuneD を制御するためのソケット 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。ソケット API は D-Bus API と 1対1でマッピングされ、D-Bus が利用できない場合に代替通信方法を提供します。ソケット API を使用すると、TuneD デーモンを制御してパフォーマンスを最適化したり、さまざまなチューニングパラメーターの値を変更したりできます。ソケット API はデフォルトでは無効になっていますが、**tuned-main.conf** ファイルで有効にできます。

[Bugzilla:2113900](#)

9.6. ネットワーク

WireGuard VPN はテクノロジープレビューとして利用可能になる

Red Hat がサポートしていないテクノロジープレビューとして提供している WireGuard は、Linux カーネルで実行する高パフォーマンスの VPN ソリューションです。最新の暗号を使用し、その他の VPN ソリューションよりも簡単に設定できます。さらに、WireGuard のコードベースが小さくなり、攻撃の影響が減るため、セキュリティが向上します。

詳細は [Setting up a WireGuard VPN](#) を参照してください。

[Bugzilla:1613522^{\[1\]}](#)

kTLS がテクノロジープレビューとして利用可能になる

RHEL はカーネル Transport Layer Security (kTLS) をテクノロジープレビューとして提供します。kTLS は AES-GCM 暗号のカーネル内の対称暗号化または復号化アルゴリズムを使用して TLS レコードを処理します。また、kTLS には、この機能を提供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ためのインターフェイスが組み込まれています。

[Bugzilla:1570255^{\[1\]}](#)

systemd-resolved サービスがテクノロジープレビューとして利用可能になる

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-resolved 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。

[Bugzilla:2020529](#)

PRP および HSR プロトコルがテクノロジープレビューとして利用可能になる

この更新では、次のプロトコルを提供する **hsr** カーネルモジュールが追加されます。

- Parallel Redundancy Protocol (PRP)
- 高可用性 Seamless Redundancy (HSR)

IEC 62439-3 標準はこれらのプロトコルを定義しており、この機能を使用してイーサネットネットワークでゼロ損失冗長性を設定できます。

[Bugzilla:2177256](#)^[1]

NetworkManager と Nmstate API は MACsec ハードウェアオフロードをサポートします

ハードウェアが MACsec ハードウェアオフロードをサポートしている場合は、NetworkManager と Nmstate API の両方を使用してこの機能を有効にできます。その結果、暗号化などの MACsec 操作を CPU からネットワークインターフェイスカードにオフロードできるようになります。

この機能は、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。

[Jira:RHEL-24337](#)

NetworkManager で HSR および PRP インターフェイスを設定できます

高可用性 Seamless Redundancy (HSR) と Parallel Redundancy Protocol (PRP) は、単一のネットワークコンポーネントの障害に対してシームレスなフェイルオーバーを提供するネットワークプロトコルです。どちらのプロトコルもアプリケーション層に対して透過的です。すなわち、メインパスと冗長パス間の切り替えはユーザーが認識することなく非常に迅速に行われるため、ユーザーが通信の中断やデータの損失を経験することはありません。**NetworkManager** サービスで **nmcli** ユーティリティと DBus メッセージシステムを使用して、HSR および PRP インターフェイスの有効化および設定を行うことができます。

[Jira:RHEL-5852](#)

NIC への IPsec カプセル化のオフロードがテクノロジープレビューとして利用可能になる

今回の更新で、IPsec パケットオフロード機能がカーネルに追加されました。以前は、暗号化をネットワークインターフェイスコントローラー (NIC) にオフロードすることしかできませんでした。今回の機能拡張により、カーネルは IPsec カプセル化プロセス全体を NIC にオフロードし、ワークロードを軽減できるようになりました。

IPsec カプセル化プロセスを NIC にオフロードすると、カーネルによるこのようなパケットの監視およびフィルタリング機能も低下することに注意してください。

[Bugzilla:2178699](#)^[1]

RHEL のモデム用のネットワークドライバーはテクノロジープレビューとして利用可能

デバイスメーカーは、デフォルト設定として連邦通信委員会 (FCC) ロックをサポートしています。FCC は、モデムと通信するためのチャンネルを WWAN ドライバーが提供する特定のシステムに、WWAN ドライバーをバインドするロックを提供します。メーカーは、モデム PCI ID に基づいて、ModemManager 用のロック解除ツールを Red Hat Enterprise Linux に統合します。ただし、WWAN ドライバーに互換性があり、機能している場合でも、事前にロックを解除していないとモデムは使用できないままになります。Red Hat Enterprise Linux は、機能が限定された次のモデム用ドライバーをテクノロジープレビューとして提供します。

- Qualcomm MHI WWAM MBIM - Telit FN990Axx
- Intel IPC over Shared Memory (IOSM) - Intel XMM 7360 LTE Advanced
- Mediatek t7xx (WWAN) - Fibocom FM350GL
- Intel IPC over Shared Memory (IOSM) - Fibocom L860GL modem

Jira:RHELDOCS-16760^[1]、Jira:RHEL-6564、Bugzilla:2110561、Bugzilla:2123542、Bugzilla:2222914

Segment Routing over IPv6 (SRv6) はテクノロジープレビューとして利用可能

RHEL カーネルは、テクノロジープレビューとして Segment Routing over IPv6 (SRv6) を提供します。この機能を使用すると、エッジコンピューティングのトラフィックフローを最適化したり、データセンターのネットワークプログラマビリティを向上したりできます。ただし、最も重要な使用例は、5G デプロイメントシナリオにおけるエンドツーエンド (E2E) ネットワークスライシングです。この領域では、SRv6 プロトコルは、特定のアプリケーションまたはサービスのネットワーク要件に対処するために、プログラム可能なカスタムネットワークスライスとリソース予約を提供します。同時に、このソリューションは単一目的のアプライアンスにデプロイでき、より小さい計算フットプリントのニーズを満たします。

Bugzilla:2186375^[1]

kTLS がバージョン 6.3 にリベース

Kernel Transport Layer Security (kTLS) 機能はテクノロジープレビューです。RHEL 9.3 では、kTLS が 6.3 アップストリームバージョンにリベースされました。主な変更点は次のとおりです。

- TX デバイスオフロードによる 256 ビットキーのサポートを追加しました。
- さまざまなバグ修正を提供しました。

Bugzilla:2183538^[1]

9.7. カーネル

Soft-iWARP ドライバーがテクノロジープレビューとして利用可能になる

Soft-iWARP(siw) は、Linux 用のソフトウェア、インターネットワイドエリア RDMA プロトコル (iWARP)、カーネルドライバーです。soft-iWARP は、TCP/IP ネットワークスタックで iWARP プロトコルスイートを実装します。このプロトコルスイートはソフトウェアで完全に実装されており、特定のリモートダイレクトメモリアクセス (RDMA) ハードウェアを必要としません。soft-iWARP を使用すると、標準のイーサネットアダプターを備えたシステムが iWARP アダプターまたは他のシステムに接続でき、すでに Soft-iWARP がインストールされている別のシステムに接続できます。

Bugzilla:2023416^[1]

rvu_af、rvu_nicpf、および rvu_nicvf がテクノロジープレビューとして利用可能

次のカーネルモジュールは、Marvell OCTEON TX2 インフラストラクチャプロセッサファミリーのテクノロジープレビューとして利用できます。

rvu_nicpf

Marvell OcteonTX2 NIC 物理機能ドライバー

rvu_nicvf

Marvell OcteonTX2 NIC 仮想機能ドライバー

rvu_nicvf

Marvell OcteonTX2 RVU 管理機能ドライバー

Bugzilla:2040643^[1]

python-drng がテクノロジープレビューとして利用可能になる

python-drng パッケージは、プログラマビリティを重視した高度なデバッグユーティリティを提供します。Python コマンドラインインターフェイスを使用して、ライブカーネルとカーネルダンプの両方をデバッグできます。さらに、**python-drng** は、デバッグタスクを自動化し、Linux カーネルの複雑な分析を実行するためのスクリプト機能を提供します。

Jira:RHEL-6973^[1]

IAA 暗号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

Intel® In-Memory Analytics Accelerator (Intel® IAA) は、プリミティブな分析機能とともに、非常にスループットの高い圧縮と展開を提供するハードウェアアクセラレーターです。

RHEL 9.4 では、圧縮および展開の操作を CPU からオフロードする **iaa_crypto** ドライバーがテクノロジープレビューとして導入されました。このドライバーは、RFC 1951 に記載されている DEFLATE 圧縮標準と互換性のある圧縮および展開をサポートします。**iaa_crypto** ドライバーは、高レベル圧縮デバイス (**zswap** など) の下のレイヤーとして機能するように設計されています。

IAA 暗号ドライバーの詳細は、以下を参照してください。

- [Intel® In-Memory Analytics Accelerator \(Intel® IAA\) User Guide](#)
- [IAA Compression Accelerator Crypto Driver](#)

Jira:RHEL-20145^[1]

9.8. ファイルシステムおよびストレージ

DAX がテクノロジープレビューとして ext4 および XFS で利用可能になる

RHEL 9 では、DAX ファイルシステムがテクノロジープレビューとして提供されています。DAX は、アプリケーションが永続メモリーをそのアドレス空間に直接マップするための手段を提供します。DAX を使用するには、システムに何らかの形式の永続メモリー (通常は1つ以上の不揮発性デュアルインラインメモリーモジュール (NVDIMM) の形式) が必要であり、DAX 互換ファイルシステムを NVDIMM 上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。

Bugzilla:1995338^[1]

NVMe-oF Discovery Service 機能がテクノロジープレビューとして利用可能になる

NVMexpress.org Technical Proposals (TP) 8013 および 8014 で定義されている NVMe-oF Discovery Service の機能は、テクノロジープレビューとして利用できます。これらの機能をプレビューするには、**nvme-cli 2.0** パッケージを使用して、TP-8013 または TP-8014 を実装する NVMe-oF ターゲットデバイスにホストを割り当てます。TP-8013 および TP-8014 の詳細は、<https://nvmexpress.org/specations/> Web サイトから NVM Express 2.0 認定 TP を参照してください。

Bugzilla:2021672^[1]

NVMe-stas 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る

Linux の Central Discovery Controller (CDC) クライアントである **nvme-stas** 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、非同期イベント通知 (AEN)、自動化された NVMe サブシステム接続制御、エラー処理とレポート、および Automatic (**zeroconf**) 手動設定を処理します。

このパッケージは、Storage Appliance Finder (**stafd**) と Storage Appliance Connector (**stacd**) の 2 つのデーモンで構成されています。

Bugzilla:1893841^[1]

NVMe TP 8006 インバンド認証がテクノロジープレビューとして利用可能になる

Non-Volatile Memory Express (NVMe) の実装 NVMe over Fabrics (NVMe-oF) のインバンド認証である TP 8006 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るようになりました。NVMe Technical Proposal 8006 で、この機能強化で提供される NVMe-oF の **DH-HMAC-CHAP** インバンド認証プロトコルが定義されています。

詳細は、**nvme-connect(1)** の man ページの **dhchap-secret** および **dhchap-ctrl-secret** オプションの説明を参照してください。

Bugzilla:2027304^[1]

9.9. コンパイラーおよび開発ツール

jmc-core および owasp-java-encoder がテクノロジープレビューとして利用可能

RHEL 9 は、AMD および Intel 64 ビットアーキテクチャー用のテクノロジープレビューとして、**jmc-core** および **owasp-java-encoder** パッケージとともに配布されます。

jmc-core は、Java Development Kit (JDK) Mission Control のコア API を提供するライブラリーです。これには、JDK Flight Recording ファイルの解析および書き込み用のライブラリーや、Java Discovery Protocol (JDP) による Java Virtual Machine (JVM) 検出のライブラリーが含まれます。

owasp-java-encoder パッケージは、Java の高パフォーマンスな低オーバーヘッドコンテキストエンコーダーのコレクションを提供します。

RHEL 9.2 以降、**jmc-core** および **owasp-java-encoder** は CodeReady Linux Builder (CRB) リポジトリで使用できるため、明示的に有効にする必要があることに注意してください。詳細は、[CodeReady Linux Builder 内でコンテンツを有効にして利用する方法](#) を参照してください。

Bugzilla:1980981

libabigail: 柔軟な配列変換の警告抑制がテクノロジープレビューとして利用可能になる

この更新により、バイナリーを比較するときに、次の抑制仕様を使用して、真のフレキシブル配列に変換された偽のフレキシブル配列に関連する警告を抑制できます。

```
[suppress_type]
  type_kind = struct
  has_size_change = true
  has_strict_flexible_array_data_member_conversion = true
```

Jira:RHEL-16629^[1]

9.10. IDENTITY MANAGEMENT

DNSSEC が IdM でテクノロジープレビューとして利用可能

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) を実装するようになりました。IdM サーバーでホストされる DNS ゾーンは、DNSSEC を使用して自動的に署名できます。暗号鍵は、自動的に生成およびローテートされます。

DNSSEC で DNS ゾーンを保護する場合は、以下のドキュメントを参照することが推奨されます。

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

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

[Bugzilla:2084180](#)

ACME がテクノロジープレビューとして利用可能

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

RHEL では、ACME サービスは Red Hat Certificate System (RHCS) PKI ACME レスポンダーを使用します。RHCS ACME サブシステムは、IdM デプロイメントのすべての認証局 (CA) サーバーに自動的にデプロイされますが、管理者が有効にするまでリクエストに対応しません。RHCS は、ACME 証明書を発行する際に **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
```

Bugzilla:2084181^[1]

9.11. 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 デスクトップ環境は、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できます。

VNC を使用して 64 ビット ARM サーバーのデスクトップセッションに接続できるようになりました。その結果、グラフィカルアプリケーションを使用してサーバーを管理できます。

64 ビット ARM では、限定されたグラフィカルアプリケーションのセットを使用できます。以下に例を示します。

- Firefox Web ブラウザー
- Red Hat Subscription マネージャー (**subscription-manager-cockpit**)
- ファイアウォール設定 (**firewall-config**)
- ディスク使用状況アナライザー (**baobab**)

Firefox を使用して、サーバー上の Cockpit サービスに接続できます。

LibreOffice などの特定のアプリケーションは、コマンドラインインターフェイスのみを提供し、グラフィカルインターフェイスは無効になっています。

Jira:RHELPLAN-27394^[1]

テクノロジープレビューとして利用可能な IBM Z アーキテクチャー用の GNOME

GNOME デスクトップ環境は、テクノロジープレビューとして IBM Z アーキテクチャーで利用できません。

VNC を使用して IBM Z サーバーのデスクトップセッションに接続できるようになりました。その結果、グラフィカルアプリケーションを使用してサーバーを管理できます。

IBM Z では、限定されたグラフィカルアプリケーションのセットを使用できます。たとえば、次のようになります。

- Firefox Web ブラウザー
- Red Hat Subscription マネージャー (**subscription-manager-cockpit**)
- ファイアウォール設定 (**firewall-config**)
- ディスク使用状況アナライザー (**baobab**)

Firefox を使用して、サーバー上の Cockpit サービスに接続できます。

LibreOffice などの特定のアプリケーションは、コマンドラインインターフェイスのみを提供し、グラフィカルインターフェイスは無効になっています。

Jira:RHELPLAN-27737^[1]

9.12. WEB コンソール

RHEL Web コンソールで WireGuard 接続を管理できるようになりました

RHEL 9.4 以降では、RHEL Web コンソールを使用して WireGuard VPN 接続を作成および管理できます。なお、WireGuard テクノロジーとその Web コンソール統合は、どちらもサポート対象外のテクノロジープレビューです。

Jira:RHELDPCS-17520^[1]

9.13. 仮想化

入れ子仮想マシンの作成

入れ子 KVM 仮想化は、RHEL 9 で Intel、AMD64、および IBM Z 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 9 ホストで実行中の RHEL 7、RHEL 8、または RHEL 9 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:RHELDPCS-17040^[1]

KVM 仮想マシンの AMD SEV および SEV-ES

RHEL 9 は、テクノロジープレビューとして、KVM ハイパーバイザーを使用する AMD EPYC ホストマシンに、セキュア暗号化仮想化 (SEV) 機能を提供します。仮想マシンで有効になっている場合は、SEV が仮想マシンのメモリーを暗号化して、ホストから仮想マシンへのアクセスを防ぎます。これにより、仮想マシンのセキュリティが向上します。

さらに、強化された SEV (Encrypted State) バージョンの SEV (SEV-ES) もテクノロジープレビューとして提供されます。SEV-ES は、仮想マシンの実行が停止すると、すべての CPU レジスターの内容を暗号化します。これにより、ホストが仮想マシンの CPU レジスターを変更したり、そこから情報を読み取ったりできなくなります。

SEV および SEV-ES は、第 2 世代の AMD EPYC CPU (コードネーム Rome) 以降のみで動作することに注意してください。また、RHEL 9 には SEV および SEV-ES の暗号化が含まれますが、SEV および SEV-ES のセキュリティー証明は含まれません。

Jira:RHELPLAN-65217^[1]

RHEL ゲストのインテル TDX

テクノロジープレビューとして、Intel Trust Domain Extension (TDX) 機能が RHEL 9.2 以降のゲストオペレーティングシステムで使用できるようになりました。ホストシステムが TDX をサポートしている場合は、トラストドメイン (TD) と呼ばれる、ハードウェアから分離された RHEL 9 仮想マシン (VM) をデプロイできます。ただし、TDX は現在 **kdump** では機能せず、TDX を有効にすると VM 上で **kdump** が失敗することに注意してください。

Bugzilla:1955275^[1]

RHEL の Unified Kernel Image がテクノロジープレビューとして利用可能になる

テクノロジープレビューとして、RHEL カーネルを仮想マシン (VM) の Unified Kernel Image (UKI) として入手できるようになりました。Unified Kernel Image は、カーネル、initramfs、およびカーネルコマンドラインを単一の署名付きバイナリーファイルに結合します。

UKI は、仮想化環境やクラウド環境、特に強力なセキュアブート機能が必要な機密 VM で使用できます。UKI は、RHEL 9 リポジトリの **kernel-uki-virt** パッケージとして利用できます。

現在、RHEL UKI は、UEFI ブート設定のみで使用できます。

Bugzilla:2142102^[1]

Intel vGPU がテクノロジープレビューとして利用可能になる

テクノロジープレビューとして、物理 Intel GPU デバイスを、**mediated devices** 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。

この機能は非推奨となり、RHEL 9.3 リリースで完全に削除されたことに注意してください。

Jira:RHELDPCS-17050^[1]

ARM 64 上の CPU クラスタ

テクノロジープレビューとして、CPU トポロジーで複数の ARM 64 CPU クラスタを使用する KVM 仮想マシンを作成できるようになりました。

Jira:RHEL-7043^[1]

Mellanox の Virtual Function による仮想マシンのライブマイグレーションがテクノロジープレビューとして利用可能になりました

テクノロジープレビューとして、Mellanox ネットワークデバイスの Virtual Function (VF) がアタッチされた仮想マシン (VM) のライブマイグレーションを利用できるようになりました。

現在、この機能は Mellanox CX-7 ネットワークデバイスでのみ利用できます。Mellanox CX-7 ネットワークデバイス上の VF は、ライブマイグレーションに必要な機能を追加する新しい **mlx5_vfio_pci** ドライバーを使用します。この新しいドライバーは、**libvirt** によって VF に自動的にバインドされます。

Jira:RHEL-13007^[1]

9.14. クラウド環境の RHEL

RHEL がテクノロジープレビューとして Azure Confidential VM で利用可能になる

更新された RHEL カーネルを使用すると、RHEL 機密仮想マシン (VM) を Microsoft Azure 上でテクノロジープレビューとして作成して実行できるようになりました。新しく追加された Unified Kernel Image (UKI) により、暗号化された機密 VM イメージを Azure 上で起動できるようになりました。UKI は、RHEL 9 リポジトリの **kernel-uki-virt** パッケージとして利用できます。

現在、RHEL UKI は、UEFI ブート設定のみで使用できます。

Jira:RHELPLAN-139800^[1]

9.15. コンテナ

podman-machine コマンドはサポート対象外です。

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

Jira:RHELDPCS-16861^[1]

マルチアーキテクチャーイメージのビルド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

マルチアーキテクチャーコンテナイメージの作成に使用できる **podman farm build** コマンドが、テクノロジープレビューとして利用できます。

ファームとは、UNIX podman ソケットが実行されているマシンのグループです。ファーム内のノードには、さまざまなアーキテクチャーのさまざまなマシンが存在する場合があります。**podman farm build** コマンドは、**podman build --arch --platform** コマンドよりも高速です。

podman farm build を使用して、次のアクションを実行できます。

- ファーム内のすべてのノードにイメージをビルドします。
- ノードをマニフェストリストにバンドルします。
- すべてのファームノードで **podman build** コマンドを実行します。
- **--tag** オプションを使用して指定されたレジストリーにイメージをプッシュします。
- マニフェストリストをローカルに作成します。
- マニフェストリストをレジストリーにプッシュします。
マニフェストリストには、ファーム内に存在するネイティブアーキテクチャータイプごとに 1 つのイメージが含まれます。

Jira:RHELPLAN-154436^[1]

新しい rhel9/rhel-bootc コンテナイメージがテクノロジープレビューとして利用可能になる

rhel9/rhel-bootc コンテナイメージが、テクノロジープレビューとして Red Hat Container Registry で利用できるようになりました。RHEL のブート可能なコンテナイメージを使用すると、コンテナとまったく同じように、オペレーティングシステムを構築、テスト、およびデプロイできます。RHEL のブート可能なコンテナイメージは、機能拡張により、既存のアプリケーションの Universal Base Images (UBI) とは異なるものとなっています。すなわち、RHEL のブート可能なコンテナイメージに

は、カーネル、initrd、ブートローダー、ファームウェアなど、起動に必要な追加コンポーネントが含まれています。既存のコンテナイメージに変更はありません。詳細は、[Red Hat Ecosystem Catalog](#) を参照してください。

Jira:RHELDPCS-17803^[1]

composefs ファイルシステムがテクノロジープレビューとして利用可能になる

composefs 読み取り専用ファイルシステムが、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、これは一般的に bootc/ostree および podman プロジェクトでのみ使用されることを目的としています。composefs を使用すると、これらのプロジェクトを使用して、読み取り専用のイメージを作成して使用したり、イメージ間でファイルデータを共有したり、実行時にイメージを検証したりできます。その結果、完全に検証されたファイルシステムツリーがマウントされ、同一ファイルが適宜きめ細かく共有されるようになります。

Jira:RHEL-18157^[1]

第10章 非推奨の機能

ここでは、Red Hat Enterprise Linux 9 で **非推奨** となった機能の概要を説明します。

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

非推奨の機能のサポートステータスは、Red Hat Enterprise Linux 9 では変更されていません。サポート期間については、[Red Hat Enterprise Linux のライフサイクル](#) および [Red Hat Enterprise Linux アプリケーションストリームのライフサイクル](#) を参照してください。

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

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

RHEL 8 には存在するが RHEL 9 では **削除** された機能については、[RHEL 9 を導入する際の考慮事項](#) を参照してください。

10.1. インストーラーおよびイメージの作成

非推奨のキックスタートコマンド

以下のキックスタートコマンドが非推奨になりました。

- **timezone --ntpservers**
- **timezone --nontp**
- **logging --level**
- **%packages --excludeWeakdeps**
- **%packages --instLangs**
- **%Anaconda**
- **pwpolicy**

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、非推奨ではないことに注意してください。キックスタートファイルで非推奨のコマンドを使用すると、ログに警告が出力されます。**inst.ksstrict** 起動オプションを使用して、非推奨のコマンド警告をエラーにすることもできます。

Bugzilla:1899167^[1]

edge-commit および edge-container ブループリントのユーザーとグループのカスタマイズは非推奨になる

ブループリントでユーザーまたはグループのカスタマイズを指定することは、**edge-commit** および **edge-container** イメージタイプでは非推奨になりました。これは、イメージをアップグレードし、ブループリントでユーザーを再度指定しないと、ユーザーのカスタマイズが失われるためです。

既存の OSTree コミットのデプロイに使用されるブループリントでのユーザーまたはグループのカスタマイズの指定 (**edge-raw-image**、**edge-installer**、**edge-simplified-installer** イメージタイプなど) は、引き続きサポートされることに注意してください。

[Bugzilla:2173928](#)

initial-setup パッケージが非推奨になる

initial-setup パッケージは、Red Hat Enterprise Linux 9.3 で非推奨になり、次の RHEL メジャーリリースで削除される予定です。代わりに、グラフィカルユーザーインターフェイスの **gnome-initial-setup** を使用します。

Jira:RHELDPCS-16393^[1]

inst.geoloc ブートオプションの provider_hostip 値と provider_fedora_geoip 値が非推奨になる

inst.geoloc= ブートオプションの GeolP API を指定した **provider_hostip** 値と **provider_fedora_geoip** 値が非推奨になる代わりに、**geolocation_provider=URL** オプションを使用して、インストールプログラム設定ファイルに必要な位置情報を設定できます。**inst.geoloc=0** オプションを使用して位置情報を無効にすることもできます。

[Bugzilla:2127473](#)

グローバルホットキーを使用して Anaconda GUI からスクリーンショットをキャプチャーすることが非推奨になりました

以前は、ユーザーはグローバルホットキーを使用して Anaconda GUI のスクリーンショットをキャプチャーできました。つまり、ユーザーはインストール環境からスクリーンショットを手動で抽出し、任意の用途で 사용할ことが可能でした。この機能は非推奨になりました。

Jira:RHELDPCS-17166^[1]

Anaconda の組み込みヘルプが非推奨になる

Anaconda のインストール時に利用可能な、全 Anaconda ユーザーインターフェイスのスポークおよびハブの組み込みドキュメントは非推奨になりました。代わりに、Anaconda ユーザーインターフェイスは自己記述型になり、ユーザーは将来の RHEL メジャーリリースで公式の [RHEL ドキュメント](#) を参照できます。

Jira:RHELDPCS-17309^[1]

NVDIMM デバイスのサポートが非推奨になる

以前は、インストールプログラムにより、インストール中に NVDIMM デバイスを再設定することができました。キックスタートおよび GUI インストール中の NVDIMM デバイスに対する当該サポートは非推奨になり、次の RHEL メジャーリリースで削除される予定です。セクターモードの NVDIMM デバイスは、インストールプログラムで引き続き表示され、使用可能です。

[Jira:RHELDPCS-17702](#)

インストール環境でドライバー更新ディスクから更新されたドライバーをロードできない

インストールの初期 RAM ディスクから同じドライバーがすでにロードされている場合、ドライバー更新ディスクからの新しいバージョンのドライバーがロードされない可能性があります。そのため、ドライバーの最新バージョンをインストール環境に適用できません。

回避策として、**modprobe.blacklist=** カーネルコマンドラインオプションを **inst.dd** オプションと一緒に使用します。たとえば、ドライバー更新ディスクから **virtio_blk** ドライバーの更新バージョンが確実に

にロードされるようにするには、`modprobe.blacklist=virtio_blk` を使用し、通常の手順を続行してドライバー更新ディスクからドライバーを適用します。その結果、システムはドライバーの更新バージョンをロードし、それをインストール環境で使用できるようになります。

[Jira:RHEL-4762](#)

10.2. セキュリティー

SHA-1 は暗号化の目的で非推奨になる

暗号化を目的とした SHA-1 メッセージダイジェストの使用は、RHEL 9 では非推奨になりました。SHA-1 によって生成されたダイジェストは、ハッシュ衝突の検出に基づく多くの攻撃の成功例が記録化されているため、セキュアであるとは見なされません。RHEL コア暗号コンポーネントは、デフォルトで SHA-1 を使用して署名を作成しなくなりました。RHEL 9 のアプリケーションが更新され、セキュリティー関連のユースケースで SHA-1 が使用されないようになりました。

例外の中でも、HMAC-SHA1 メッセージ認証コードと Universal Unique Identifier (UUID) 値は、SHA-1 を使用して作成できます。これは、これらのユースケースが現在セキュリティーリスクをもたらさないためです。SHA-1 は、Kerberos や WPA-2 など、相互運用性および互換性に関する重要な懸念事項に関連する限られたケースでも使用できます。詳細は、[RHEL 9 セキュリティーの強化ドキュメント](#) の [FIPS 140-3 に準拠していない暗号化を使用する RHEL アプリケーションのリスト](#) を参照してください。

既存またはサードパーティーの暗号署名を検証するために SHA-1 を使用する必要がある場合は、次のコマンドを入力して有効にできます。

```
# update-crypto-policies --set DEFAULT:SHA1
```

または、システム全体の暗号化ポリシーを **LEGACY** ポリシーに切り替えることもできます。**LEGACY** は、セキュアではない他の多くのアルゴリズムも有効にすることに注意してください。

[Jira:RHELPLAN-110763^{\[1\]}](#)

fapolicyd.rules が非推奨になる

実行ルールの許可と拒否を含むファイルの `/etc/fapolicyd/rules.d/` ディレクトリーは、`/etc/fapolicyd/fapolicyd.rules` ファイルを置き換えます。`fagenrules` スクリプトは、このディレクトリー内のすべてのコンポーネントルールファイルを `/etc/fapolicyd/compiled.rules` ファイルにマージするようになりました。`/etc/fapolicyd/fapolicyd.trust` のルールは引き続き `fapolicyd` フレームワークによって処理されますが、下位互換性を確保するためのみに使用されます。

[Bugzilla:2054740](#)

RHEL 9 で SCP が非推奨になる

SCP (Secure Copy Protocol) には既知のセキュリティー脆弱性があるため、非推奨となりました。SCP API は RHEL 9 ライフサイクルで引き続き利用できますが、システムセキュリティーが低下します。

- `scp` ユーティリティーでは、SCP はデフォルトで SSH ファイル転送プロトコル (SFTP) に置き換えられます。
- OpenSSH スイートは、RHEL 9 では SCP を使用しません。
- `libssh` ライブラリーで SCP が非推奨になりました。

[Jira:RHELPLAN-99136^{\[1\]}](#)

OpenSSL では、FIPS モードでの RSA 暗号化にパディングが必要です。

OpenSSL は、FIPS モードでのパディングなしの RSA 暗号化をサポートしなくなりました。パディングを使用しない RSA 暗号化は一般的ではないため、ほとんど使用されません。RSA (RSASVE) によるキーのカプセル化はパディングを使用しませんが、引き続きサポートされていることに注意してください。

[Bugzilla:2168665](#)

OpenSSL で Engines API が非推奨になる

OpenSSL 3.0 TLS ツールキットでは、Engines API が非推奨になりました。エンジンインターフェイスはプロバイダー API に置き換えられました。アプリケーションと既存のエンジンのプロバイダーへの移行が進行中です。非推奨の Engines API は、今後のメジャーリリースで削除される可能性があります。

Jira:RHELDOCS-17958^[1]

openssl-pkcs11 が非推奨になる

非推奨になった OpenSSL エンジンのプロバイダー API への継続的な移行の一環として、**pkcs11-provider** パッケージが **openssl-pkcs11** パッケージ (**engine_pkcs11**) を置き換えます。**openssl-pkcs11** パッケージは非推奨になりました。**openssl-pkcs11** パッケージは、今後のメジャーリリースで削除される可能性があります。

Jira:RHELDOCS-16716^[1]

RHEL 8 および 9 OpenSSL 証明書および署名コンテナが非推奨になる

Red Hat Ecosystem Catalog の **ubi8/openssl** および **ubi9/openssl** リポジトリで利用可能な OpenSSL ポータブル証明書および署名コンテナは、需要が低いため非推奨になりました。

Jira:RHELDOCS-17974^[1]

Digest-MD5 SASL では非推奨となりました。

SASL (Simple Authentication Security Layer) フレームワークの Digest-MD5 認証メカニズムは非推奨になり、将来バージョンのメジャーリリースでは **cyrus-sasl** パッケージから削除される可能性あり

[Bugzilla:1995600^{\[1\]}](#)

OpenSSL は、MD2、MD4、MDC2、Whirlpool、Blowfish、CAST、DES、IDEA、RC2、RC4、RC5、SEED、および PBKDF1 を非推奨にします。

OpenSSL プロジェクトは、セキュアではない、一般的ではない、またはその両方であるという理由で、一連の暗号アルゴリズムを非推奨にしました。Red Hat もそれらのアルゴリズムの使用を推奨せず、RHEL 9 では、新しいアルゴリズムを使用するために暗号化されたデータを移行するためにそれらを提供しています。ユーザーは、自分のシステムのセキュリティのためにこれらのアルゴリズムに依存してはいけません。

アルゴリズム MD2、MD4、MDC2、Whirlpool、Blowfish、CAST、DES、IDEA、RC2、RC4、RC5、SEED、および PBKDF1 の実装は、OpenSSL のレガシープロバイダーに移行されました。

レガシープロバイダーをロードし、非推奨のアルゴリズムのサポートを有効にする方法については、**/etc/pki/tls/openssl.cnf** 設定ファイルを参照してください。

[Bugzilla:1975836](#)

/etc/system-fips が非推奨に

`/etc/system-fips` ファイルで FIPS モードが削除されることを示すサポートにより、ファイルは今後の RHEL バージョンに含まれなくなります。FIPS モードで RHEL をインストールするには、システムのインストール時に `fips=1` パラメーターをカーネルコマンドラインに追加します。`fips-mode-setup --check` コマンドを使用して、RHEL が FIPS モードで動作しているかどうかを確認できます。

Jira:RHELPLAN-103232^[1]

libcrypt.so.1 が非推奨に

`libcrypt.so.1` ライブラリーは現在非推奨であり、RHEL の将来のバージョンで削除される可能性があります。

Bugzilla:2034569

10.3. サブスクリプションの管理

`subscription-manager register` の非推奨の `--token` オプションは、2024 年 11 月末に機能しなくなります

`subscription-manager register` コマンドの非推奨の `--token=<TOKEN>` オプションは、2024 年 11 月末以降はサポート対象の認証方法ではなくなります。デフォルトのエンタitlementサーバー `subscription.rhsm.redhat.com` では、トークンベースの認証が許可されなくなります。したがって、`subscription-manager register --token=<TOKEN>` を使用すると、次のエラーメッセージが表示されて登録が失敗します。

```
Token authentication not supported by the entitlement server
```

システムを登録するには、サポート対象の他の認可方法を使用します。たとえば、`subscription-manager register` コマンドにペアのオプション `--username / --password` または `--org / --activationkey` を含めます。

Bugzilla:2163716

10.4. シェルおよびコマンドラインツール

`dump` からの `dump` ユーティリティーが非推奨になりました。

ファイルシステムのバックアップに使用される `dump` ユーティリティーが非推奨になり、RHEL 9 では使用できなくなります。

RHEL 9 では、使用方法に基づいて、`tar`、`dd`、または `bacula` のバックアップユーティリティーを使用することが推奨されています。これにより、`ext2`、`ext3`、および `ext4` のファイルシステムで完全に安全なバックアップが提供されます。

`dump` パッケージの `restore` ユーティリティーは、RHEL 9 で引き続き利用可能で、サポートされており、`restore` パッケージとして利用できます。

Bugzilla:1997366^[1]

Bacula の SQLite データベースバックエンドは廃止されました

Bacula バックアップシステムは、複数のデータベースバックエンド (PostgreSQL、MySQL、および SQLite) をサポートしていました。SQLite バックエンドは廃止され、RHEL の今後のリリースではサポートされなくなります。代わりに、他のバックエンド (PostgreSQL または MySQL) のいずれかに移行し、新しい展開では SQLite バックエンドを使用しないでください。

Jira:RHEL-6856

sysstat パッケージの %vmeff メトリクスは非推奨になりました

ページ再利用効率を測定する **sysstat** パッケージの **%vmeff** メトリクスは、将来の RHEL メジャーバージョンではサポートされなくなります。**sysstat** は、新しいカーネルバージョンで提供されるすべての関連する **/proc/vmstat** 値を解析しないため、**sar -B** コマンドによって返される **%vmeff** 列の値は正しくありません。

/proc/vmstat ファイルから **%vmeff** 値を手動で計算できます。詳細は、[Why the sar\(1\) tool reports %vmeff values beyond 100 % in RHEL 8 and RHEL 9?](#) を参照してください。

Jira:RHELDPCS-17015^[1]

ReaR 設定ファイルでの TMPDIR 変数の設定が非推奨になる

export TMPDIR=... などのステートメントを使用して、**/etc/rear/local.conf** または **/etc/rear/site.conf** ReaR 設定ファイルで **TMPDIR** 環境変数を設定することは非推奨となりました。

ReaR 一時ファイルのカスタムディレクトリーを指定するには、ReaR を実行する前にシェル環境で変数をエクスポートします。たとえば、**export TMPDIR=...** ステートメントを実行してから、同じシェルセッションまたはスクリプトで **rear** コマンドを実行します。

Jira:RHELDPCS-18049^[1]

RHEL 9 で cgroups v1 が非推奨になりました

cgroups は、プロセス追跡、システムリソースの割り当て、およびパーティション設定に使用されるカーネルサブシステムです。systemd サービスマネージャーは、**cgroups v1** モードと **cgroups v2** モードでの起動をサポートします。Red Hat Enterprise Linux 9 では、デフォルトのモードは **v2** です。Red Hat Enterprise Linux 10 では、systemd は **cgroups v1** モードでの起動をサポートせず、**cgroups v2** モードのみが利用可能になります。

Jira:RHELDPCS-17545^[1]

10.5. インフラストラクチャーサービス

クライアントサイドおよびサーバーサイドの DHCP パッケージが非推奨になりました

Internet Systems Consortium (ISC) は、2022 年末をもって ISC DHCP のメンテナンスを終了することを発表しました。そのため、Red Hat は、RHEL 9 でのクライアントサイドおよびサーバーサイドの DHCP パッケージの使用を非推奨とし、今後の RHEL メジャーバージョンでは配布しないことを決定しました。お客様は、**dhcpcd** や **ISC Kea** などの利用可能な代替手段への移行を準備する必要があります。

Jira:RHELDPCS-17135^[1]

sendmail、libotr、mod_security、spamassassin パッケージが非推奨になりました

以下のパッケージは RHEL 9 では非推奨となり、今後の RHEL メジャーバージョンでは配布されません。

- sendmail - Red Hat では、postfix メールデーモン (サポート対象) への移行を推奨しています。
- libotr
- mod_security

- spamassassin

Jira:RHEL-22385^[1]

10.6. ネットワーク

RHEL 9 でネットワークチームが非推奨になりました

teamd サービスおよび **libteam** ライブラリーは、Red Hat Enterprise Linux 9 では非推奨になり、次回のメジャーリリースでは削除される予定です。代替として、ネットワークチームの代わりにボンディングを設定します。

Red Hat は、機能が類似するボンディングとチームの機能を 2 つ管理しなくてもいいように、カーネルベースのボンディングに注力しています。ボンディングコードは、顧客の採用率が高く、堅牢で、活発なコミュニティー開発が行われています。その結果、ボンディングコードは拡張、更新されます。

ボンディングにチームを移行する方法は、[Migrating a network team configuration to network bond](#) を参照してください。

Bugzilla:1935544^[1]

ifcfg 形式の NetworkManager 接続プロファイルが非推奨になりました

RHEL 9.0 以降では、**ifcfg** 形式の接続プロファイルは非推奨になりました。次の RHEL メジャーリリースでは、この形式のサポートが削除されます。ただし、RHEL 9 では、既存のプロファイルを変更すると、NetworkManager は引き続きこの形式で既存のプロファイル処理および更新します。

デフォルトでは、NetworkManager は接続プロファイルをキーファイル形式で `/etc/NetworkManager/system-connections/` ディレクトリーに保存するようになりました。**ifcfg** 形式とは異なり、キーファイル形式は、NetworkManager が提供するすべての接続設定をサポートします。キーファイル形式とプロファイルの移行方法の詳細は、[NetworkManager connection profiles in keyfile format](#) を参照してください。

Bugzilla:1894877^[1]

firewalld の iptables バックエンドが非推奨になりました

RHEL 9 では、**iptables** フレームワークは非推奨になりました。結果として、**iptables** バックエンドと、**firewalld** の **直接インターフェイス** も非推奨になりました。**直接インターフェイス** の代わりに、**firewalld** のネイティブ機能を使用して、必要なルールを設定できます。

Bugzilla:2089200

firewalld のロックダウン機能が非推奨になりました

firewalld のロックダウン機能は非推奨になりました。理由は、**root** として実行中のプロセスが自身を許可リストに追加するのを防げないためです。ロックダウン機能は、今後の RHEL メジャーリリースで削除される可能性があります。

Jira:RHEL-17708

connection.master、connection.slave-type、connection.autoconnect-slaves プロパティーが非推奨になりました

Red Hat では、意識的な言語の使用に取り組んでいます。この取り組みに関する詳細は、[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、**connection.master**、**connection.slave-type**、**connection.autoconnect-slaves** プロパティーの

名前が変更されました。下位互換性を確保するために、古いプロパティ名を新しいプロパティ名にマップするエイリアスが作成されました。

- **connection.master** は **connection.controller** のエイリアスです
- **connection.slave-type** は **connection.port-type** のエイリアスです
- **connection.autoconnect-slaves** は、**connection.autoconnect-ports** のエイリアスです

connection.master、**connection.slave-type**、および **connection.autoconnect-slaves** エイリアスは非推奨となり、今後の RHEL バージョンでは削除されることに注意してください。

Jira:RHEL-17619^[1]

PF_KEYv2 カーネル API が非推奨になりました

アプリケーションは、**PV_KEYv2** および新しい **netlink** API を使用して、カーネルの IPsec 実装を設定できます。**PV_KEYv2** はアップストリームで積極的に保守されておらず、最新の暗号、オフロード、拡張シーケンス番号サポートなどの重要なセキュリティー機能が欠けています。その結果、RHEL 9.3 以降、**PV_KEYv2** API は非推奨となり、次の RHEL メジャーリリースで削除される予定です。アプリケーションでこのカーネル API を使用する場合は、代替として最新の **netlink** API を使用するよう移行してください。

Jira:RHEL-1015^[1]

10.7. カーネル

RHEL 9 で ATM カプセル化が非推奨になりました

非同期転送モード (ATM) カプセル化により、ATM アダプテーションレイヤー 5(AAL-5) のレイヤー 2(ポイントツーポイントプロトコル、イーサネット) またはレイヤー 3(IP) 接続が可能になります。Red Hat は、RHEL7 以降 ATMNIC ドライバーのサポートを提供していません。ATM 実装のサポートは RHEL 9 で廃止されています。これらのプロトコルは現在、ADSL テクノロジーをサポートし、メーカーによって段階的に廃止されているチップセットのみで使用されています。したがって、ATM カプセル化は Red Hat Enterprise Linux 9 では非推奨です。

詳細は、[PPP Over AAL5](#)、[Multiprotocol Encapsulation over ATM Adaptation Layer 5](#)、および [Classical IP and ARP over ATM](#) を参照してください。

Bugzilla:2058153

kexec-tools の kexec_load システムコールが非推奨になりました

2 番目のカーネルをロードする **kexec_load** システムコールは、将来の RHEL リリースではサポートされなくなります。**kexec_file_load** システムコールは **kexec_load** に代わるもので、現在はすべてのアーキテクチャーのデフォルトのシステムコールです。

詳細は、[Is kexec_load supported in RHEL9?](#) を参照してください。

Bugzilla:2113873^[1]

RHEL 9 でネットワークチームが非推奨になりました

teamd サービスおよび **libteam** ライブラリーは、Red Hat Enterprise Linux 9 では非推奨になり、次のメジャーリリースでは削除される予定です。代替として、ネットワークチームの代わりにボンディングを設定します。

Red Hat は、機能が類似するボンディングとチームの機能を 2 つ管理しなくてもいいように、カーネルベースのボンディングに注力しています。ボンディングコードは、顧客の採用率が高く、堅牢で、活発なコミュニティ開発が行われています。その結果、ボンディングコードは拡張、更新されます。

ボンディングにチームを移行する方法は、[Migrating a network team configuration to network bond](#) を参照してください。

Bugzilla:2013884^[1]

10.8. ファイルシステムおよびストレージ

lvm2-activation-generator およびその生成されたサービスが RHEL 9.0 で削除される

lvm2-activation-generator プログラムとその生成されたサービス **lvm2-activation**、**lvm2-activation-early**、および **lvm2-activation-net** は、RHEL 9.0 で削除されています。サービスをアクティベートするために使用される **lvm.conf event_activation** 設定は機能しなくなりました。ボリュームグループを自動アクティブ化する唯一の方法は、イベントベースのアクティブ化です。

Bugzilla:2038183

RHEL 9 で永続メモリー開発キット (pmdk) とサポートライブラリーが非推奨になりました

pmdk は、システム管理者とアプリケーション開発者の永続メモリーデバイスの管理とアクセスを簡素化するためのライブラリーとツールのコレクションです。**pmdk** およびサポートライブラリーは、RHEL 9 で非推奨になりました。これには、**-debuginfo** パッケージも含まれます。

pmdk によって生成された以下の一覧のバイナリーパッケージ (**nvml** ソースパッケージを含む) が非推奨になりました。

- **libpmem**
- **libpmem-devel**
- **libpmem-debug**
- **libpmem2**
- **libpmem2-devel**
- **libpmem2-debug**
- **libpmemblk**
- **libpmemblk-devel**
- **libpmemblk-debug**
- **libpmemlog**
- **libpmemlog-devel**
- **libpmemlog-debug**
- **libpmemobj**
- **libpmemobj-devel**

- `libmemobj-debug`
- `libmempool`
- `libmempool-devel`
- `libmempool-debug`
- `pmempool`
- `daxio`
- `pmreorder`
- `pmdk-convert`
- `libmemobj++`
- `libmemobj++-devel`
- `libmemobj++-doc`

[Jira:RHELDOCS-16432^{\[1\]}](#)

md-linear および md-faulty モジュールが非推奨になりました

以下の MD RAID カーネルモジュールが非推奨となり、今後の RHEL メジャーリリースで削除される予定です。

- **CONFIG_MD_LINEAR** または **md-linear** モジュールは、複数のドライブを連結して、単一のメンバーディスクがいっぱいになったときに、すべてのディスクがいっぱいになるまでデータが次のディスクに書き込まれるようにします。
- 読み取りまたは書き込みエラーを時々返すブロックデバイスをテストするための **CONFIG_MD_FAULTY** または **md-faulty** モジュール。これは、テストに役立ちます。

[Jira:RHEL-30730^{\[1\]}](#)

VDO sysfs パラメーターが非推奨になりました

Virtual Data Optimizer (VDO) **sysfs** パラメーターは非推奨となり、今後の RHEL メジャーリリースで削除される予定です。 **log_level** を除き、 **kvdo** モジュールのすべてのモジュールレベルの **sysfs** パラメーターが削除されます。個々の **dm-vdo** ターゲットでは、VDO に固有のすべての **sysfs** パラメーターも削除されます。すべての DM ターゲットに共通するパラメーターには変更はありません。現在、削除されたモジュールレベルのパラメーターを更新することによって設定されている **dm-vdo** ターゲットの設定値は、変更できなくなります。

dm-vdo ターゲットの統計情報と設定値は、 **sysfs** 経由ではアクセスできなくなります。しかし、これらの値は、 **dmsetup message stats**、 **dmsetup status**、 および **dmsetup table** の **dmsetup** コマンドを使用して引き続きアクセスできます。

[Jira:RHEL-30525](#)

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

libdb が非推奨になりました

RHEL 8 および RHEL 9 は、現在、LGPLv2 ライセンスで配布される Berkeley DB (**libdb**) バージョン 5.3.28 を提供しています。アップストリームの Berkeley DB バージョン 6 は、より厳しい AGPLv3 ライセンスで利用できます。

libdb パッケージは、RHEL 9 で非推奨になり、将来バージョンの RHEL では利用できない可能性があります。

また、RHEL 9 では、**libdb** から暗号アルゴリズムが削除され、RHEL 9 では複数の **libdb** 依存関係が削除されています。

libdb のユーザーは、別の鍵値データベースに移行することが推奨されます。詳細は、ナレッジベースの記事 [Available replacements for the deprecated Berkeley DB \(libdb\) in RHEL](#) を参照してください。

Bugzilla:1927780^[1]、Jira:RHELPLAN-80695、[Bugzilla:1974657](#)

10.10. コンパイラーおよび開発ツール

Go の FIPS モードの openssl 3.0 で、2048 より小さいサイズのキーが非推奨になりました

2048 ビットより小さい鍵サイズは **openssl** 3.0 で廃止され、Go の FIPS モードでは機能しなくなりました。

[Bugzilla:2111072](#)

Go の FIPS モードで、一部の PKCS1 v1.5 モードが非推奨になりました

一部の **PKCS1** v1.5 モードは、**FIPS-140-3** で暗号化が承認されておらず、無効になっています。Go の FIPS モードでは機能しなくなります。

Bugzilla:2092016^[1]

32 ビットパッケージが非推奨になりました

32 ビットの multilib パッケージに対するリンクは非推奨になりました。***.i686** パッケージは Red Hat Enterprise Linux 9 のライフサイクル期間中はサポートされ続けますが、RHEL の次のメジャーバージョンでは削除されます。

Jira:RHELDOCS-17917^[1]

10.11. IDENTITY MANAGEMENT

OpenDNSSec の SHA-1 が非推奨になりました

OpenDNSSec は、**SHA-1** アルゴリズムを使用したデジタル署名および認証レコードのエクスポートに対応しています。**SHA-1** アルゴリズムの使用に対応しなくなりました。RHEL 9 リリースでは、OpenDNSSec の **SHA-1** が非推奨になり、今後のマイナーリリースで削除される可能性があります。また、OpenDNSSec のサポートは、Red Hat Identity Management との統合に限定されます。OpenDNSSec はスタンドアロンでは対応していません。

[Bugzilla:1979521](#)

SSSD 暗黙的なファイルプロバイダーメインは、デフォルトで無効になっています。

`/etc/shadow` などのローカルファイルからユーザー情報を取得する SSSD 暗黙的な **ファイル** プロバイダードメイン、および `/etc/group` からグループ情報を取得する SSSD 暗黙的な **ファイル** プロバイダードメインは、デフォルトで無効になりました。

SSSD を使用してローカルファイルからユーザーおよびグループ情報を取得するには、次のコマンドを実行します。

1. SSSD を設定します。以下のいずれかのオプションを選択します。
 - a. **sssd.conf** 設定ファイルで **id_provider=files** を使用して、ローカルドメインを明示的に設定します。

```
[domain/local]
id_provider=files
...
```

- b. **sssd.conf** 設定ファイルで **enable_files_domain=true** を設定して、**ファイル** プロバイダーを有効にします。

```
[sssd]
enable_files_domain = true
```

2. ネームサービススイッチを設定します。

```
# authselect enable-feature with-files-provider
```

[Jira:RHELPLAN-100639^{\[1\]}](#)

SSSD files プロバイダーが非推奨になりました

SSSD **files** プロバイダーは Red Hat Enterprise Linux (RHEL) 9 で非推奨になりました。 **files** プロバイダーは、RHEL の将来のリリースから削除される可能性があります。

[Jira:RHELPLAN-139805^{\[1\]}](#)

AD および IdM の enumeration 機能が非推奨になりました

enumeration 機能を使用すると、Active Directory (AD)、Identity Management (IdM)、および LDAP プロバイダーに対して、引数なしで **getent passwd** または **getent group** コマンドを使用することで、すべてのユーザーまたはグループをリスト表示できます。Red Hat Enterprise Linux (RHEL) 9 では、AD および IdM に対する **enumeration** 機能のサポートは廃止されました。RHEL 10 では、AD および IdM に対する **enumeration** 機能は削除されます。

[Jira:SSSD-6596](#)

libsss_simpleifp サブパッケージが非推奨になりました

libsss_simpleifp.so ライブラリーを提供する **libsss_simpleifp** サブパッケージは、Red Hat Enterprise Linux (RHEL) 9 で非推奨になりました。 **libsss_simpleifp** サブパッケージは、RHEL の今後のリリースから削除される可能性があります。

[Jira:SSSD-6601](#)

Samba で SMB1 プロトコルが非推奨になりました

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

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

Jira:RHELDPCS-16612^[1]

10.12. デスクトップ

GTK 2 が非推奨になりました

レガシー GTK 2 ツールキットと、以下の関連パッケージが非推奨になりました。

- **adwaita-gtk2-theme**
- **gnome-common**
- **gtk2**
- **gtk2-immodules**
- **hexchat**

現在、他にも複数のパッケージが GTK 2 に依存しています。今後の RHEL メジャーリリースで非推奨パッケージへの依存が発生しないよう、これらは変更されます。

GTK 2 を使用するアプリケーションを維持する場合、Red Hat は、アプリケーションを GTK 4 に移植することを推奨します。

Jira:RHELPLAN-131882^[1]

LibreOffice が非推奨になりました

LibreOffice RPM パッケージは非推奨となり、今後の RHEL メジャーリリースで削除される予定です。LibreOffice は、RHEL 7、8、および 9 のライフサイクル全体を通じて引き続き完全にサポートされません。

Red Hat は、RPM パッケージの代わりに、The Document Foundation が提供する次のいずれかのソースから LibreOffice をインストールすることを推奨します。

- Flathub リポジトリの公式 Flatpak パッケージ:
<https://flathub.org/apps/org.libreoffice.LibreOffice>
- 公式 RPM パッケージ: <https://www.libreoffice.org/download/download-libreoffice/>

Jira:RHELDPCS-16300^[1]

10.13. グラフィックインフラストラクチャー

Motif が非推奨になりました

アップストリームの Motif コミュニティーでの開発は非アクティブであるため、Motif ウィジェットツールキットは RHEL で非推奨になりました。

開発バリエーションおよびデバッグバリエーションを含む、以下の Motif パッケージが非推奨になりました。

- **motif**

- **openmotif**
- **openmotif21**
- **openmotif22**

さらに、**motif-static** パッケージが削除されました。

Red Hat は、GTK ツールキットを代替として使用することを推奨します。GTK は Motif と比較してメンテナンス性が高く、新機能を提供します。

[Jira:RHELPLAN-98983^{\[1\]}](#)

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

mssql_ha_cluster_run_role は廃止される

mssql_ha_cluster_run_role 変数は非推奨になりました。代わりに、**mssql_manage_ha_cluster** 変数を使用します。

[Jira:RHEL-19092](#)

RHEL 9 ノードでチームを設定すると、**network** システムロールが非推奨の警告を表示します。

ネットワークチーム機能は、RHEL 9 では非推奨になりました。その結果、RHEL 8 制御ノードで **network** RHEL システムロールを使用して RHEL 9 ノードでネットワークチームを設定すると、非推奨についての警告が表示されます。

[Bugzilla:1999770](#)

10.15. 仮想化

SHA1 ベースの署名を使用した SecureBoot イメージ検証が非推奨になりました

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。

[Bugzilla:1935497^{\[1\]}](#)

仮想フロッピードライバーが非推奨になりました

仮想フロッピーディスクデバイスを制御する **isa-fdc** ドライバーが非推奨になり、今後の RHEL ではサポートされなくなります。そのため、移行した仮想マシンとの前方互換性を確保するため、Red Hat では、RHEL 9 でホストされている仮想マシンでのフロッピーディスクデバイスの使用を推奨しません。

[Bugzilla:1965079](#)

qcow2-v2 イメージ形式が非推奨になりました

RHEL 9 では、仮想ディスクイメージの qcow2-v2 形式が非推奨になり、将来バージョンの RHEL ではサポートされなくなります。また、RHEL 9 Image Builder は、qcow2-v2 形式のディスクイメージを作成できません。

Red Hat では、qcow2-v2 の代わりに、qcow2-v3 の使用を推奨しています。qcow2-v2 イメージを、それ以降の形式に変換する場合は、**qemu-img amend** コマンドを使用します。

[Bugzilla:1951814](#)

virt-manager が非推奨になりました

Virtual Machine Manager アプリケーション (**virt-manager**) は非推奨になっています。RHEL Web コンソール (**Cockpit**) は、後続のリリースで置き換えられる予定です。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 Web コンソールで利用できない場合があります。

Jira:RHELPLAN-10304^[1]

libvirtd が非推奨になりました

モノリシック **libvirt** デーモン **libvirtd** は、RHEL 9 で非推奨になり、RHEL の将来のメジャーリリースで削除される予定です。ハイパーバイザーで仮想化を管理するために **libvirtd** を引き続き使用することに注意してください。ただし、Red Hat では、新しく導入されたモジュラー **libvirt** デーモンに切り替えることを推奨します。手順と詳細は、[RHEL 9 の仮想化の設定と管理](#) に関するドキュメントを参照してください。

Jira:RHELPLAN-113995^[1]

レガシー CPU モデルが非推奨になりました

かなりの数の CPU モデルが非推奨になり、RHEL の将来のメジャーリリースで仮想マシン (VM) での使用がサポートされなくなります。非推奨のモデルは次のとおりです。

- Intel の場合: Intel Xeon 55xx および 75xx プロセッサファミリー (Nehalem と呼ばれます) より前のモデル
- AMD の場合: AMD Opteron G4 より前のモデル
- IBM Z の場合: IBM z14 より前のモデル

VM が非推奨の CPU モデルを使用しているかどうかを確認するには、**virsh dominfo** ユーティリティを使用し、**Message** セクションで次のような行を探します。

```
tainted: use of deprecated configuration settings
deprecated configuration: CPU model 'i486'
```

[Bugzilla:2060839](#)

RDMA ベースのライブマイグレーションが非推奨になりました

この更新により、リモートダイレクトメモリーアクセス (RDMA) を使用した実行中の仮想マシンの移行は非推奨になりました。その結果、**rdma://** 移行 URI を使用して RDMA 経由の移行を要求することは可能ですが、この機能は RHEL の将来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-153267^[1]

Intel vGPU 機能が削除される

以前は、テクノロジープレビューとして、物理 Intel GPU デバイスを、**mediated devices** と呼ばれる複数の仮想デバイスに分割することができました。続いて、これらの仲介デバイスは、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した。その結果、これらの仮想マシンは単一の物理 Intel GPU のパフォーマンスを共有しましたが、この機能と互換性があるのは一部の Intel GPU のみでした。

RHEL 9.3 以降、Intel vGPU 機能は完全に削除されました。

Bugzilla:2206599^[1]

pmem デバイスパススルーが非推奨になりました

この更新により、不揮発性メモリーライブラリー (nvm) パッケージは非推奨となり、RHEL の今後のメジャーバージョンでは削除される予定です。その結果、パッケージが削除された場合、永続メモリー (pmem) デバイスを仮想マシンに渡すことができなくなります。揮発性メモリーまたはファイルでバックアップした、エミュレートされた NVDIMM デバイスは引き続き使用できますが、永続的に設定することはできない点に注意してください。

Jira:RHELDPCS-17989

Windows Server 2012 または Windows 8 のゲストオペレーティングシステムとしての使用はサポート対象外

Microsoft が以下のバージョンの Windows のサポートを終了したため、Red Hat も今回の更新でこれらのバージョンをゲストオペレーティングシステムとして使用するサポートを終了しました。

- Windows 8
- Windows 8.1
- Windows Server 2012
- Windows Server 2012 R2

Jira:RHEL-11810

10.16. コンテナ

RHEL 7 ホストでの RHEL 9 コンテナの実行がサポート対象外

RHEL 7 ホストでは、RHEL 9 コンテナの実行に対応していません。正常に動作するかもしれませんが、保証されません。

詳細は、[Red Hat Enterprise Linux Container Compatibility Matrix](#) を参照してください。

Jira:RHELPLAN-100087^[1]

Podman 内の SHA1 ハッシュアルゴリズムが非推奨になりました

ルートレスネットワーク namespace のファイル名を生成するために使用される SHA1 アルゴリズムは Podman ではサポートされなくなりました。したがって、Podman 4.1.1 以降に更新する前に起動されたルートレスコンテナは、ネットワークに参加している場合は (slirp4netns を使用するだけでなく) 再起動して、アップグレード後に起動したコンテナに接続できるようにする必要があります。

Bugzilla:2069279^[1]

rhel9/pause が非推奨になりました

rhel9/pause コンテナイメージが非推奨になりました

Bugzilla:2106816

CNI ネットワークスタックが非推奨になりました

Container Network Interface (CNI) ネットワークスタックは非推奨となり、RHEL の今後のマイナーリリースでは Podman から削除される予定です。以前は、コンテナは DNS 経由のみで単一の

Container Network Interface (CNI) プラグインに接続していました。Podman v.4.0 では、新しい Netavark ネットワークスタックが導入されました。Netavark ネットワークスタックは、Podman およびその他の Open Container Initiative (OCI) コンテナ管理アプリケーションとともに使用できます。Podman 用の Netavark ネットワークスタックは、高度な Docker 機能とも互換性があります。複数のネットワーク内のコンテナは、それらのネットワークのいずれかにあるコンテナにアクセスできます。

詳細は、[CNI から Netavark へのネットワークスタックの切り替え](#) を参照してください。

Jira:RHELDOCS-16756^[1]

Inkscape および LibreOffice Flatpak イメージが非推奨になりました

テクノロジープレビューとして利用可能な **rhel9/inkscape-flatpak** および **rhel9/libreoffice-flatpak** Flatpak イメージは非推奨になりました。

Red Hat は、これらのイメージに代わる次の代替手段を推奨します。

- **rhel9/inkscape-flatpak** を置き換えるには、**inkscape** RPM パッケージを使用します。
- **rhel9/libreoffice-flatpak** を置き換えるには、[LibreOffice 非推奨化に関するリリースノート](#) を参照してください。

Jira:RHELDOCS-17102^[1]

ネットワーク名としての **pasta** が非推奨になりました

ネットワーク名の値としての **pasta** に関するサポートは非推奨になり、Podman の次のメジャーリリースであるバージョン 5.0 では使用できなくなります。**pasta** というネットワーク名の値は、**podman run --network** コマンドと **podman create --network** コマンドを使用して、Podman 内に一意のネットワークモードを作成するために使用できます。

Jira:RHELDOCS-17038^[1]

BoltDB データベースバックエンドが非推奨になりました

BoltDB データベースバックエンドは、RHEL 9.4 以降では非推奨です。RHEL の今後のバージョンでは、BoltDB データベースバックエンドが削除され、Podman では利用できなくなります。Podman の場合は、RHEL 9.4 以降ではデフォルトとなっている SQLite データベースバックエンドを使用してください。

Jira:RHELDOCS-17495^[1]

CNI ネットワークスタックが非推奨になりました

Container Network Interface (CNI) ネットワークスタックは非推奨となり、今後のリリースでは削除される予定です。代わりに Netavark ネットワークスタックを使用してください。詳細は、[CNI から Netavark へのネットワークスタックの切り替え](#) を参照してください。

Jira:RHELDOCS-17518^[1]

Podman v5.0 における今後の非推奨項目

RHEL 9.5 および RHEL 10.0 ベータ版でリリースされる予定の Podman v5.0 では、以下が非推奨になります。

- BoltDB データベースバックエンドは非推奨になります。新しい SQLite データベースバックエンドが利用可能になりました。

- **containers.conf** ファイルは読み取り専用になります。システム接続とファーム情報は、Podman によってのみ管理される **podman.connections.json** ファイルに保存されます。Podman は、**[engine.service_destinations]** や **[farms]** セクションなどの古い設定オプションを引き続きサポートします。必要に応じて手動で接続またはファームを追加することはできませんが、**podman system connection rm** コマンドを使用して **containers.conf** ファイルから接続を削除することはできません。

RHEL 10.0 ベータ版では以下の変更が予定されています。

- **pasta** ネットワークモードは、ルートレスコンテナのデフォルトのネットワークモードになります。**slirp4netns** ネットワークモードは非推奨となります。
- **cgroupv1** は非推奨となります。
- **CNI** ネットワークスタックは非推奨となります。

Jira:RHELDPCS-17462^[1]

rhel9/openssl が非推奨になりました

rhel9/openssl コンテナイメージが非推奨になりました。

Jira:RHELDPCS-18106^[1]

10.17. 非推奨のパッケージ

このセクションでは、非推奨となり、将来バージョンの Red Hat Enterprise Linux には含まれない可能性があるパッケージのリストを示します。

RHEL 8 と RHEL 9 との間でパッケージを変更する場合は、**RHEL 9 の導入における考慮事項** ドキュメントの [パッケージの変更](#) を参照してください。



重要

非推奨パッケージのサポート状況は、RHEL 9 内でも変更されません。サポート期間の詳細は、[Red Hat Enterprise Linux のライフサイクル](#) および [Red Hat Enterprise Linux アプリケーションストリームのライフサイクル](#) を参照してください。

次のパッケージは RHEL 9 で非推奨になりました。

- **aacraid**
- **adwaita-gtk2-theme**
- **af_key**
- **anaconda-user-help**
- **autocorr-af**
- **autocorr-bg**
- **autocorr-ca**
- **autocorr-cs**

- autocorr-da
- autocorr-de
- autocorr-dsb
- autocorr-el
- autocorr-en
- autocorr-es
- autocorr-fa
- autocorr-fi
- autocorr-fr
- autocorr-ga
- autocorr-hr
- autocorr-hsb
- autocorr-hu
- autocorr-is
- autocorr-it
- autocorr-ja
- autocorr-ko
- autocorr-lb
- autocorr-lt
- autocorr-mn
- autocorr-nl
- autocorr-pl
- autocorr-pt
- autocorr-ro
- autocorr-ru
- autocorr-sk
- autocorr-sl
- autocorr-sr
- autocorr-sv

- autocorr-tr
- autocorr-vi
- autocorr-vro
- autocorr-zh
- cheese
- cheese-libs
- clutter
- clutter-gst3
- clutter-gtk
- cogl
- daxio
- dbus-glib
- dbus-glib-devel
- dhcp-client
- dhcp-common
- dhcp-relay
- dhcp-server
- enchant
- enchant-devel
- eog
- evolution
- evolution-bogofilter
- evolution-devel
- evolution-help
- evolution-langpacks
- evolution-mapi
- evolution-mapi-langpacks
- evolution-pst
- evolution-spamassassin

- festival
- festival-data
- festvox-slt-arctic-hts
- flite
- flite-devel
- firewire-core
- gedit
- gedit-plugin-bookmarks
- gedit-plugin-bracketcompletion
- gedit-plugin-codecomment
- gedit-plugin-colorpicker
- gedit-plugin-colorschemer
- gedit-plugin-commander
- gedit-plugin-drawspaces
- gedit-plugin-findinfiles
- gedit-plugin-joinlines
- gedit-plugin-multiedit
- gedit-plugin-sessionsaver
- gedit-plugin-smartspaces
- gedit-plugin-synctex
- gedit-plugin-terminal
- gedit-plugin-textsize
- gedit-plugin-translate
- gedit-plugin-wordcompletion
- gedit-plugins
- gedit-plugins-data
- ghostscript-x11
- gnome-common
- gnome-photos

- gnome-photos-tests
- gnome-screenshot
- gnome-themes-extra
- gtk2
- gtk2-devel
- gtk2-devel-docs
- gtk2-immodule-xim
- gtk2-immodules
- highcontrast-icon-theme
- inkscape
- inkscape-docs
- inkscape-view
- iptables-devel
- iptables-libs
- iptables-nft
- iptables-nft-services
- iptables-utils
- libdb
- libgdata
- libgdata-devel
- libpmem
- libpmem-debug
- libpmem-devel
- libpmem2
- libpmem2-debug
- libpmem2-devel
- libpmemblk
- libpmemblk-debug
- libpmemblk-devel

- libpmemlog
- libpmemlog-debug
- libpmemlog-devel
- libpmemobj
- libpmemobj-debug
- libpmemobj-devel
- libpmempool
- libpmempool-debug
- libpmempool-devel
- libreoffice
- libreoffice-base
- libreoffice-calc
- libreoffice-core
- libreoffice-data
- libreoffice-draw
- libreoffice-emailmerge
- libreoffice-filters
- libreoffice-gdb-debug-support
- libreoffice-graphicfilter
- libreoffice-gtk3
- libreoffice-help-ar
- libreoffice-help-bg
- libreoffice-help-bn
- libreoffice-help-ca
- libreoffice-help-cs
- libreoffice-help-da
- libreoffice-help-de
- libreoffice-help-dz
- libreoffice-help-el

- libreoffice-help-en
- libreoffice-help-eo
- libreoffice-help-es
- libreoffice-help-et
- libreoffice-help-eu
- libreoffice-help-fi
- libreoffice-help-fr
- libreoffice-help-gl
- libreoffice-help-gu
- libreoffice-help-he
- libreoffice-help-hi
- libreoffice-help-hr
- libreoffice-help-hu
- libreoffice-help-id
- libreoffice-help-it
- libreoffice-help-ja
- libreoffice-help-ko
- libreoffice-help-lt
- libreoffice-help-lv
- libreoffice-help-nb
- libreoffice-help-nl
- libreoffice-help-nn
- libreoffice-help-pl
- libreoffice-help-pt-BR
- libreoffice-help-pt-PT
- libreoffice-help-ro
- libreoffice-help-ru
- libreoffice-help-si
- libreoffice-help-sk

- libreoffice-help-sl
- libreoffice-help-sv
- libreoffice-help-ta
- libreoffice-help-tr
- libreoffice-help-uk
- libreoffice-help-zh-Hans
- libreoffice-help-zh-Hant
- libreoffice-impress
- libreoffice-langpack-af
- libreoffice-langpack-ar
- libreoffice-langpack-as
- libreoffice-langpack-bg
- libreoffice-langpack-bn
- libreoffice-langpack-br
- libreoffice-langpack-ca
- libreoffice-langpack-cs
- libreoffice-langpack-cy
- libreoffice-langpack-da
- libreoffice-langpack-de
- libreoffice-langpack-dz
- libreoffice-langpack-el
- libreoffice-langpack-en
- libreoffice-langpack-eo
- libreoffice-langpack-es
- libreoffice-langpack-et
- libreoffice-langpack-eu
- libreoffice-langpack-fa
- libreoffice-langpack-fi
- libreoffice-langpack-fr

- libreoffice-langpack-fy
- libreoffice-langpack-ga
- libreoffice-langpack-gl
- libreoffice-langpack-gu
- libreoffice-langpack-he
- libreoffice-langpack-hi
- libreoffice-langpack-hr
- libreoffice-langpack-hu
- libreoffice-langpack-id
- libreoffice-langpack-it
- libreoffice-langpack-ja
- libreoffice-langpack-kk
- libreoffice-langpack-kn
- libreoffice-langpack-ko
- libreoffice-langpack-lt
- libreoffice-langpack-lv
- libreoffice-langpack-mai
- libreoffice-langpack-ml
- libreoffice-langpack-mr
- libreoffice-langpack-nb
- libreoffice-langpack-nl
- libreoffice-langpack-nn
- libreoffice-langpack-nr
- libreoffice-langpack-nso
- libreoffice-langpack-or
- libreoffice-langpack-pa
- libreoffice-langpack-pl
- libreoffice-langpack-pt-BR
- libreoffice-langpack-pt-PT

- libreoffice-langpack-ro
- libreoffice-langpack-ru
- libreoffice-langpack-si
- libreoffice-langpack-sk
- libreoffice-langpack-sl
- libreoffice-langpack-sr
- libreoffice-langpack-ss
- libreoffice-langpack-st
- libreoffice-langpack-sv
- libreoffice-langpack-ta
- libreoffice-langpack-te
- libreoffice-langpack-th
- libreoffice-langpack-tn
- libreoffice-langpack-tr
- libreoffice-langpack-ts
- libreoffice-langpack-uk
- libreoffice-langpack-ve
- libreoffice-langpack-xh
- libreoffice-langpack-zh-Hans
- libreoffice-langpack-zh-Hant
- libreoffice-langpack-zu
- libreoffice-math
- libreoffice-ogltrans
- libreoffice-opensymbol-fonts
- libreoffice-pdfimport
- libreoffice-pyuno
- libreoffice-sdk
- libreoffice-sdk-doc
- libreoffice-ure

- libreoffice-ure-common
- libreoffice-wiki-publisher
- libreoffice-writer
- libreoffice-x11
- libreoffice-xsltfilter
- libreofficekit
- libsoup
- libsoup-devel
- libuser
- libuser-devel
- libwpe
- libwpe-devel
- mcpp
- mod_auth_mellon
- motif
- motif-devel
- pmdk-convert
- pmempool
- python3-pytz
- q1a4xxx
- qt5
- qt5-assistant
- qt5-designer
- qt5-devel
- qt5-doctools
- qt5-linguist
- qt5-qdbusviewer
- qt5-qt3d
- qt5-qt3d-devel

- qt5-qt3d-doc
- qt5-qt3d-examples
- qt5-qtbase
- qt5-qtbase-common
- qt5-qtbase-devel
- qt5-qtbase-doc
- qt5-qtbase-examples
- qt5-qtbase-gui
- qt5-qtbase-mysql
- qt5-qtbase-odbc
- qt5-qtbase-postgresql
- qt5-qtbase-private-devel
- qt5-qtbase-static
- qt5-qtconnectivity
- qt5-qtconnectivity-devel
- qt5-qtconnectivity-doc
- qt5-qtconnectivity-examples
- qt5-qtdeclarative
- qt5-qtdeclarative-devel
- qt5-qtdeclarative-doc
- qt5-qtdeclarative-examples
- qt5-qtdeclarative-static
- qt5-qtimageformats
- qt5-qtimageformats-doc
- qt5-qtlocation
- qt5-qtlocation-devel

- qt5-qtlocation-doc
- qt5-qtlocation-examples
- qt5-qtmultimedia
- qt5-qtmultimedia-devel
- qt5-qtmultimedia-doc
- qt5-qtmultimedia-examples
- qt5-qtquickcontrols
- qt5-qtquickcontrols-doc
- qt5-qtquickcontrols-examples
- qt5-qtquickcontrols2
- qt5-qtquickcontrols2-devel
- qt5-qtquickcontrols2-doc
- qt5-qtquickcontrols2-examples
- qt5-qtscript
- qt5-qtscript-devel
- qt5-qtscript-doc
- qt5-qtscript-examples
- qt5-qtsensors
- qt5-qtsensors-devel
- qt5-qtsensors-doc
- qt5-qtsensors-examples
- qt5-qtserialbus
- qt5-qtserialbus-devel
- qt5-qtserialbus-doc
- qt5-qtserialbus-examples
- qt5-qtserialport
- qt5-qtserialport-devel
- qt5-qtserialport-doc
- qt5-qtserialport-examples

- qt5-qtsvg
- qt5-qtsvg-devel
- qt5-qtsvg-doc
- qt5-qtsvg-examples
- qt5-qttools
- qt5-qttools-common
- qt5-qttools-devel
- qt5-qttools-doc
- qt5-qttools-examples
- qt5-qttools-libs-designer
- qt5-qttools-libs-designercomponents
- qt5-qttools-libs-help
- qt5-qttools-static
- qt5-qttranslations
- qt5-qtwayland
- qt5-qtwayland-devel
- qt5-qtwayland-doc
- qt5-qtwayland-examples
- qt5-qtwebchannel
- qt5-qtwebchannel-devel
- qt5-qtwebchannel-doc
- qt5-qtwebchannel-examples
- qt5-qtwebsockets
- qt5-qtwebsockets-devel
- qt5-qtwebsockets-doc
- qt5-qtwebsockets-examples
- qt5-qtqmlextras
- qt5-qtqmlextras-devel
- qt5-qtqmlextras-doc

- qt5-qtxmlpatterns
- qt5-qtxmlpatterns-devel
- qt5-qtxmlpatterns-doc
- qt5-qtxmlpatterns-examples
- qt5-rpm-macros
- qt5-srpm-macros
- team
- tigervnc
- tigervnc-icons
- tigervnc-license
- tigervnc-selinux
- tigervnc-server
- tigervnc-server-minimal
- tigervnc-server-module
- webkit2gtk3
- webkit2gtk3-devel
- webkit2gtk3-jsc
- webkit2gtk3-jsc-devel
- wpebackend-fdo
- wpebackend-fdo-devel
- xorg-x11-server-Xorg
- yp-tools
- ypbind
- ypserv

第11章 既知の問題

ここでは、Red Hat Enterprise Linux 9.4 の既知の問題について説明します。

11.1. 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ルプログラム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。

Bugzilla:1640697^[1]

`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。

Bugzilla:1697896^[1]

Anaconda がアプリケーションとして実行されているシステムでの予期しない SELinux ポリシー

Anaconda がすでにインストールされているシステムでアプリケーションとして実行されている場合 (たとえば、`-image anaconda` オプションを使用してイメージファイルに別のインストールを実行する場合)、システムはインストール中に SELinux のタイプと属性を変更することを禁止されていません。そのため、SELinux ポリシーの特定の要素は、Anaconda が実行されているシステムで変更される可能性があります。

この問題を回避するには、実稼働システムで Anaconda を実行しないでください。代わりに、一時的な仮想マシンで Anaconda を実行して、実稼働システムの SELinux ポリシーを変更しないようにします。`boot.iso` や `dvd.iso` からのインストールなど、システムインストールプロセスの一部として `anaconda` を実行しても、この問題の影響は受けません。

Bugzilla:2050140

サードパーティーのツールを使用して作成した USB からインストールを起動する際に、Local Media のインストールソースが検出されない

サードパーティーのツールを使用して作成された USB から RHEL インストールを起動すると、インストールプログラムが `Media` インストールソースを検出できません (Red Hat CDN のみが検出されません)。

この問題は、デフォルトの起動オプション `int.stage2=` が `iso9660` イメージ形式の検索を試みるためです。ただし、サードパーティーツールは、別の形式の ISO イメージを作成する可能性があります。

回避策として、以下のソリューションのいずれかを使用します。

- インストールの起動時に **Tab** キーをクリックしてカーネルコマンドラインを編集し、起動オプション **inst.stage2=** を **inst.repo=** に変更します。
- Windows で起動可能な USB デバイスを作成するには、Fedora Media Writer を使用します。
- Rufus などのサードパーティーツールを使用して起動可能な USB デバイスを作成する場合は、最初に Linux システムで RHEL ISO イメージを再生成し、サードパーティーのツールを使用して起動可能な USB デバイスを作成します。

指定の回避策を実行する手順の詳細は、[Installation media is not auto-detected during the installation of RHEL 8.3](#) を参照してください。

Bugzilla:1877697^[1]

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。

Jira:RHEL-4707

iso9660 ファイルシステムで、ハードドライブがパーティション分割されたインストールが失敗する

ハードドライブが **iso9660** ファイルシステムでパーティションが設定されているシステムには、RHEL をインストールできません。これは、**iso9660** ファイルシステムパーティションを含むハードディスクを無視するように設定されている、更新されたインストールコードが原因です。これは、RHEL が DVD を使用せずにインストールされている場合でも発生します。

この問題を回避するには、インストールの開始前に、キックスタートファイルに次のスクリプトを追加して、ディスクをフォーマットします。

メモ: 回避策を実行する前に、ディスクで利用可能なデータのバックアップを作成します。**wipefs** は、ディスク内の全データをフォーマットします。

```
%pre
wipefs -a /dev/sda
%end
```

その結果、インストールでエラーが発生することなく、想定どおりに機能します。

Jira:RHEL-4711

Anaconda が管理者ユーザーアカウントの存在の確認に失敗する

グラフィカルユーザーインターフェイスを使用して RHEL をインストールしている場合に、管理者アカウントが作成されていると、Anaconda が確認に失敗します。その結果、管理者ユーザーアカウントがなくても、システムをインストールできてしまう可能性があります。

この問題を回避するには、管理者ユーザーアカウントを設定するか、root パスワードを設定して、root アカウントのロックを解除します。その結果、インストール済みシステムで管理タスクを実行できます。

[Bugzilla:2047713](#)

新しい XFS 機能により、バージョン 5.10 よりも古いファームウェアを持つ PowerNV IBM POWER システムが起動しなくなる

PowerNV IBM POWER システムは、ファームウェアに Linux カーネルを使用し、GRUB の代わりに Petitboot を使用します。これにより、ファームウェアカーネルのマウント `/boot` が発生し、Petitboot が GRUB 設定を読み取り、RHEL を起動します。

RHEL 9 カーネルでは、XFS ファイルシステムに **bigtime=1** 機能および **inobtcount=1** 機能が導入されています。これは、バージョン 5.10 よりも古いファームウェアのカーネルが理解できません。

この問題を回避するには、`/boot` に別のファイルシステム (ext4 など) を使用できます。

[Bugzilla:1997832^{\[1\]}](#)

rpm-ostree ペイロードをインストールすると、RHEL for Edge インストーラーイメージがマウントポイントの作成に失敗する

RHEL for Edge インストーラーイメージなどで使用される **rpm-ostree** ペイロードをデプロイする場合、インストールプログラムがカスタムパーティションの一部のマウントポイントを適切に作成しません。その結果、インストールは以下のエラーで中止されます。

```
The command 'mount --bind /mnt/sysimage/data /mnt/sysroot/data' exited with the code 32.
```

この問題を回避するには、以下を実行します。

- 自動パーティション設定スキームを使用し、手動でマウントポイントを追加しないでください。
- マウントポイントは、`/var` ディレクトリー内のみを手動で割り当てます。たとえば、`/var/my-mount-point` や、`/boot`、`/var` などの標準ディレクトリーです。

その結果、インストールプロセスは正常に終了します。

[Jira:RHEL-4741](#)

ネットワークに接続されているが、DHCP または静的 IP アドレスが設定されていない場合、NetworkManager はインストール後に起動に失敗する

RHEL 9.0 以降、特定の **ip=** またはキックスタートネットワーク設定が設定されていない場合、Anaconda はネットワークデバイスを自動的にアクティブ化します。Anaconda は、イーサネットデバイスごとにデフォルトの永続的な設定ファイルを作成します。接続プロファイルには、**ONBOOT** と **autoconnect** の値が **true** に設定されています。その結果、インストールされたシステムの起動中に、RHEL がネットワークデバイスをアクティブ化し、**networkManager-wait-online** サービスが失敗します。

回避策として、以下のいずれかを実行します。

- 使用する 1 つの接続を除いて、**nmcli** ユーティリティーを使用してすべての接続を削除します。以下に例を示します。
 - a. すべての接続プロファイルを一覧表示します。

```
# nmcli connection show
```

- b. 不要な接続プロファイルを削除します。


```
# nmcli connection delete <connection_name>
```

<connection_name> を、削除する接続の名前に置き換えます。

- 特定の **ip=** またはキックスタートネットワーク設定が設定されていない場合は、Anaconda の自動接続ネットワーク機能を無効にします。
 - a. Anaconda GUI で、**Network & Hostname** に移動します。
 - b. 無効にするネットワークデバイスを選択します。
 - c. **Configure** をクリックします。
 - d. **General** タブで、**Connect automatically with priority** チェックボックスをオフにします。
 - e. **Save** をクリックします。

Bugzilla:2115783^[1]

キックスタートインストールでネットワーク接続の設定に失敗する

Anaconda は、NetworkManager API を通じてのみキックスタートネットワーク設定を実行します。Anaconda は、**%pre** キックスタートセクションの後にネットワーク設定を処理します。その結果、キックスタート **%pre** セクションの一部のタスクがブロックされます。たとえば、**%pre** セクションからのパッケージのダウンロードは、ネットワーク設定が利用できないため失敗します。

この問題を回避するには、以下を実行します。

- たとえば、**%pre** スクリプトの一部として **nmcli** ツールを使用して、ネットワークを設定します。
- インストールプログラムのブートオプションを使用して、**%pre** スクリプトのネットワークを設定します。

その結果、**%pre** セクションのタスクにネットワークを使用できるようになり、キックスタートインストールプロセスが完了します。

Bugzilla:2173992

RHEL Image Builderを使用して rpm-ostree イメージをビルドする場合、FIPS モードの有効化はサポート対象外

現在、RHEL Image Builderを使用して **rpm-ostree** イメージをビルドするときに FIPS モードを有効にすることはサポートされていません。

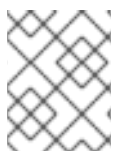
Jira:RHEL-4655

stig プロファイル修復でビルドされたイメージが FIPS エラーで起動に失敗する

FIPS モードは、RHEL Image Builder ではサポートされていません。**xccdf_org.ssgproject.content_profile_stig** プロファイル修復でカスタマイズされた RHEL Image Builder を使用すると、システムは次のエラーで起動に失敗します。

```
Warning: /boot//.vmlinuz-<kernel version>.x86_64.hmac does not exist
FATAL: FIPS integrity test failed
Refusing to continue
```

`/boot` ディレクトリーが別のパーティションにあるため、システムイメージのインストール後に **fips-mode-setup --enable** コマンドを使用して FIPS ポリシーを手動で有効にしても機能しません。FIPS が無効になっている場合、システムは正常に起動します。現在、使用可能な回避策はありません。



注記

イメージのインストール後に、**fips-mode-setup --enable** コマンドを使用して、FIPS を手動で有効にすることができます。

[Jira:RHEL-4649](#)

ドライバーディスクメニューがコンソールでユーザー入力を表示できない

ドライバーディスクを使用して、カーネルコマンドラインで **inst.dd** オプションを使用して RHEL インストールを開始すると、コンソールにユーザー入力が表示されません。そのため、アプリケーションがユーザー入力に応答せず、応答を停止しているように見えますが、出力は表示されます。これはユーザーにはわかりにくい動作です。ただし、この動作は機能に影響を与えず、**Enter** を押すとユーザー入力が登録されます。

回避策として、予想される結果を確認するには、コンソールでユーザー入力が存在しないことを無視し、入力の追加が終了したら **Enter** を押します。

[Jira:RHEL-4737](#)

%packages セクションに systemd サービスファイルを含むパッケージがないため、キックスタートインストールが失敗する

キックスタートファイルで **services --enabled=...** ディレクティブを使用して **systemd** サービスを有効にし、指定したサービスファイルを含むパッケージが **%packages** セクションに含まれていない場合、RHEL のインストールプロセスは次のエラーで失敗します。

```
Error enabling service <name_of_the_service>
```

この問題を回避するには、キックスタートの **%packages** セクションに、サービスファイルを含むパッケージを追加します。こうすることで、インストール中に期待されるサービスが有効になり、RHEL のインストールが完了します。

[Jira:RHEL-9633^{\[1\]}](#)

bootc-image-builder はプライベートレジストリーからのイメージの構築をサポートしていません

現在、**bootc-image-builder** を使用してプライベートレジストリーから取得されるベースディスクイメージを構築することはできません。この問題を回避するには、プライベートレジストリーをローカルホストにコピーし、次の引数を使用してイメージをビルドします。

- **--local**
- **localhost/<image name>:tag** as the image

たとえば、イメージをビルドするには、次のようにします。

```
sudo podman run \
--rm \
-it \
--privileged \
```

```
--pull=newer \
--security-opt label=type:unconfined_t \
-v ./config.toml:/config.toml \
-v ./output:/output \
-v /var/lib/containers/storage:/var/lib/containers/storage \
registry.redhat.io/rhel9/bootc-image-builder:latest
--type qcow2 \
--local \
quay.io/<namespace>/<image>:<tag>
```

[Jira:RHEL-34054](#)

11.2. セキュリティー

OpenSSL は PKCS #11 トークンが raw RSA または RSA-PSS 署名の作成をサポートしているか検出しない

TLS 1.3 プロトコルには、RSA-PSS 署名のサポートが必要です。PKCS #11 トークンが生の RSA または RSA-PSS 署名をサポートしていない場合、キーが PKCS #11 トークンによって保持されている場合、OpenSSL ライブラリーを使用するサーバーアプリケーションは RSA キーを操作できません。これにより、上記のシナリオで TLS 通信に失敗します。

この問題を回避するには、利用可能な最高の TLS プロトコルバージョンとして TLS バージョン 1.2 を使用するようにサーバーとクライアントを設定します。

[Bugzilla:1681178^{\[1\]}](#)

OpenSSL が、生の RSA または RSA-PSS の署名に対応していない PKCS #11 トークンを誤って処理する

OpenSSL ライブラリーは、PKCS #11 トークンの鍵関連の機能を検出しません。したがって、生の RSA または RSA-PSS の署名に対応しないトークンで署名が作成されると、TLS 接続の確立に失敗します。

この問題を回避するには、`/etc/pki/tls/openssl.cnf` ファイルの `crypto_policy` セクションの末尾にある `.include` 行の後に、以下の行を追加します。

```
SignatureAlgorithms =
RSA+SHA256:RSA+SHA512:RSA+SHA384:ECDSA+SHA256:ECDSA+SHA512:ECDSA+SHA384
MaxProtocol = TLSv1.2
```

これにより、このシナリオで TLS 接続を確立できます。

[Bugzilla:1685470^{\[1\]}](#)

特定の構文を使用すると、scp は自身にコピーされたファイルを空にする

scp ユーティリティーが Secure copy protocol (SCP) からよりセキュアな SSH ファイル転送プロトコル (SFTP) に変更されました。したがって、ある場所からファイルを同じ場所にコピーすると、ファイルの内容が消去されます。この問題は以下の構文に影響します。

```
scp localhost:/myfile localhost:/myfile
```

この問題を回避するには、この構文を使用して、ソースの場所と同じ宛先にファイルをコピーしないでください。

この問題は、以下の構文に対して修正されました。

- **scp /myfile localhost:/myfile**
- **scp localhost:~/myfile ~/myfile**

[Bugzilla:2056884](#)

OSCAP Anaconda アドオンは、グラフィカルインストールで調整されたプロファイルをフェッチしない

OSCAP Anaconda アドオンには、RHEL グラフィカルインストールでセキュリティープロファイルの調整を選択または選択解除するオプションがありません。RHEL 8.8 以降、アドオンはアーカイブまたは RPM パッケージからインストールするときにデフォルトで調整を考慮しません。その結果、インストールでは、OSCAP に合わせたプロファイルを取得する代わりに、次のエラーメッセージが表示されます。

```
There was an unexpected problem with the supplied content.
```

この問題を回避するには、キックスタートファイルの `%addon org_fedora_oscap` セクションにパスを指定する必要があります。次に例を示します。

```
xccdf-path = /usr/share/xml/scap/sc_tailoring/ds-combined.xml  
tailoring-path = /usr/share/xml/scap/sc_tailoring/tailoring-xccdf.xml
```

その結果、OSCAP 調整プロファイルのグラフィカルインストールは、対応するキックスタート仕様のみで使用できます。

[Jira:RHEL-1824](#)

Ansible 修復には追加のコレクションが必要

ansible-core パッケージによる Ansible Engine の置き換えにより、RHEL サブスクリプションで提供される Ansible モジュールのリストが削減されました。これにより、**scap-security-guide** パッケージに含まれる Ansible コンテンツを使用する修復を実行するには、**rhc-worker-playbook** パッケージからのコレクションが必要です。

Ansible 修復の場合は、以下の手順を実行します。

1. 必要なパッケージをインストールします。

```
# dnf install -y ansible-core scap-security-guide rhc-worker-playbook
```

2. `/usr/share/scap-security-guide/ansible` ディレクトリーに移動します。

```
# cd /usr/share/scap-security-guide/ansible
```

3. 追加の Ansible コレクションへのパスを定義する環境変数を使用して、関連する Ansible Playbook を実行します。

```
# ANSIBLE_COLLECTIONS_PATH=/usr/share/rhc-worker-  
playbook/ansible/collections/ansible_collections/ ansible-playbook -c local -i localhost, rhel9-  
playbook-cis_server_11.yml
```

`cis_server_11` を、システムを修正するプロファイルの ID に置き換えます。

これにより、Ansible コンテンツは正しく処理されます。



注記

rhc-worker-playbook で提供されるコレクションのサポートは、**scap-security-guide** から取得する Ansible コンテンツの有効化だけに限定されます。

[Jira:RHEL-1800](#)

Keylime は連結された PEM 証明書を受け入れない

Keylime が単一のファイルに連結された PEM 形式の複数の証明書として証明書チェーンを受信すると、**keylime-agent-rust** Keylime コンポーネントは署名検証中に提供されたすべての証明書を正しく使用せず、TLS ハンドシェイクが失敗します。その結果、クライアントコンポーネント (**keylime_verifier** および **keylime_tenant**) は Keylime エージェントに接続できません。この問題を回避するには、複数の証明書ではなく1つの証明書だけを使用します。

[Jira:RHELPLAN-157225](#)^[1]

Keylime は、ダイジェストがバックスラッシュで始まるランタイムポリシーを拒否する

ランタイムポリシーを生成する現在のスクリプト **create_runtime_policy.sh** は、SHA チェックサム関数 (**sha256sum** など) を使用してファイルダイジェストを計算します。ただし、入力ファイル名にバックスラッシュまたは **\n** が含まれている場合、チェックサム関数は出力のダイジェストの前にバックスラッシュを追加します。このような場合、生成されたポリシーファイルの形式は不正になります。不正な形式のポリシーファイルが提供されると、Keylime テナントは、エラーメッセージ **me.tenant - ERROR - Response code 400: Runtime policy is malformed** (または同様のエラーメッセージ) を生成します。この問題を回避するには、**sed -i 's/^\//g' <malformed_file_name>** コマンドを入力して、不正な形式のポリシーファイルからバックスラッシュを手動で削除します。

[Jira:RHEL-11867](#)^[1]

Keylime エージェントが更新後に verifier からのリクエストを拒否する

Keylime エージェント (**keylime-agent-rust**) の API バージョン番号が更新されると、エージェントは別のバージョンを使用するリクエストを拒否します。その結果、Keylime エージェントが verifier に追加されて更新されると、verifier は古い API バージョンを使用してエージェントに接続しようとします。エージェントはこのリクエストを拒否し、認証に失敗します。この問題を回避するには、エージェント (**keylime-agent-rust**) を更新する前に verifier (**keylime-verifier**) を更新します。その結果、エージェントが更新されると、verifier は API の変更を検出し、それに応じて保存されているデータを更新します。

[Jira:RHEL-1518](#)^[1]

trustdb にファイルが見つからないため、fapolicyd が拒否される

fapolicyd が Ansible DISA STIG プロファイルとともにインストールされると、競合状態により **trustdb** データベースが **rpmdb** データベースと同期しなくなります。その結果、**trustdb** 内のファイルが見つからないと、システムで拒否が発生します。この問題を回避するには、**fapolicyd** を再起動するか、Ansible DISA STIG プロファイルを再度実行します。

[Jira:RHEL-24345](#)^[1]

fapolicyd ユーティリティーは、変更されたファイルの実行を誤って許可する

正しくは、ファイルの IMA ハッシュはファイルに変更が加えられた後に更新され、**fapolicyd** は変更されたファイルの実行を阻止する必要があります。ただし、IMA ポリシーのセットアップと **evctml** ユー

ティリティーによるファイルハッシュの違いにより、これは起こりません。その結果、変更されたファイルの拡張属性で IMA ハッシュは更新されません。その結果、**fapolicyd** は、変更されたファイルの実行を誤って許可します。

Jira:RHEL-520^[1]

デフォルトの SELinux ポリシーにより、制限のない実行ファイルがスタックを実行可能にする

SELinux ポリシーの **selinuxuser_execstack** ブール値のデフォルトの状態は on です。これは、制限のない実行ファイルがスタックを実行可能にすることを意味します。実行可能ファイルはこのオプションを使用しないでください。また、ハードコーディングされていない実行ファイルや攻撃の可能性を示している可能性があります。ただし、他のツール、パッケージ、およびサードパーティー製品との互換性のため、Red Hat はデフォルトポリシーのブール値を変更できません。そのような互換性の側面に依存しない場合は、コマンド **setsebool -P selinuxuser_execstack off** を入力して、ローカルポリシーでブール値をオフにすることができます。

Bugzilla:2064274

STIG プロファイルの SSH タイムアウトルールが誤ったオプションを設定する

OpenSSH の更新は、次の米国国防情報システム局のセキュリティ技術実装ガイド (DISA STIG) プロファイルのルールに影響を与えました。

- RHEL 9 用 DISA STIG (**xccdf_org.ssgproject.content_profile_stig**)
- RHEL 9 用、GUI の DISA STIG (**xccdf_org.ssgproject.content_profile_stig_gui**)

これらの各プロファイルでは、次の 2 つのルールが影響を受けます。

Title: Set SSH Client Alive Count Max to zero
CCE Identifier: CCE-90271-8
Rule ID: xccdf_org.ssgproject.content_rule_sshd_set_keepalive_0

Title: Set SSH Idle Timeout Interval
CCE Identifier: CCE-90811-1
Rule ID: xccdf_org.ssgproject.content_rule_sshd_set_idle_timeout

SSH サーバーに適用すると、これらの各ルールは、以前のように動作しなくなったオプション (**ClientAliveCountMax** および **ClientAliveInterval**) を設定します。その結果、OpenSSH は、これらのルールで設定されたタイムアウトに達したときに、アイドル状態の SSH ユーザーを切断しなくなりました。回避策として、これらのルールは、ソリューションが開発されるまで、DISA STIG for RHEL 9 および DISA STIG with GUI for RHEL 9 プロファイルから一時的に削除されました。

Bugzilla:2038978

GnuPG は crypto-policies によって許可されていない場合でも、SHA-1 署名の使用を誤って許可する

GNU Privacy Guard (GnuPG) 暗号化ソフトウェアは、システム全体の暗号化ポリシーで定義されている設定に関係なく、SHA-1 アルゴリズムを使用する署名を作成および検証できます。したがって、**DEFAULT** の暗号化ポリシーで暗号化の目的で SHA-1 を使用できます。これは、署名に対するこのセキュアではないアルゴリズムのシステム全体での非推奨とは一致しません。

この問題を回避するには、SHA-1 を含む GnuPG オプションを使用しないでください。これにより、セキュアでない SHA-1 署名を使用して GnuPG がデフォルトのシステムセキュリティを下げるのを防ぎます。

[Bugzilla:2070722](#)

OpenSCAP のメモリー消費の問題

メモリーが限られているシステムでは、OpenSCAP スキャナが途中で停止するか、結果ファイルが生成されない可能性があります。この問題を回避するには、スキャンプロファイルをカスタマイズして、/ ファイルシステム全体の再帰を含むルールの選択を解除します。

- `rpm_verify_hashes`
- `rpm_verify_permissions`
- `rpm_verify_ownership`
- `file_permissions_unauthorized_world_writable`
- `no_files_unowned_by_user`
- `dir_perms_world_writable_system_owned`
- `file_permissions_unauthorized_suid`
- `file_permissions_unauthorized_sgid`
- `file_permissions_ungroupowned`
- `dir_perms_world_writable_sticky_bits`

詳細とその他の回避策については、関連する [ナレッジベースの記事](#) を参照してください。

[Bugzilla:2161499](#)

キックスタートインストール時のサービス関連ルールの修正が失敗する場合がある

キックスタートインストール時に、OpenSCAP ユーティリティーで、サービスの **enable** または **disable** 状態の修正は不要であると誤って表示されることがあります。その結果、OpenSCAP によって、インストール済みシステム上のサービスが非準拠状態に設定され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。

[Jira:RHELPLAN-44202^{\[1\]}](#)

CNSA 1.0 により FIPS:OSPP ホストの相互運用性が影響を受ける

OSPP サブポリシーは、Commercial National Security Algorithm (CNSA) 1.0 に準拠しています。これは、**FIPS:OSPP** ポリシーとサブポリシーの組み合わせを使用するホストの相互運用性に影響します。主に影響を受ける点は次のとおりです。

- RSA キーの最小サイズは 3072 ビットとすることが必要です。
- アルゴリズムネゴシエーションでは、AES-128 暗号、secp256r1 楕円曲線、および FFDHE-2048 グループがサポートされなくなりました。

[Jira:RHEL-2735^{\[1\]}](#)

SELinux ポリシーにルールがないため、SQL データベースへの権限がブロックされる

SELinux ポリシーの権限ルールが欠落していると、SQL データベースへの接続がブロックされます。そ

の結果、FIDO Device Onboard (FDO) サービスの **fdo-manufacturing-server.service**、**fdo-owner-onboarding-server.service**、および **fdo-rendezvous-server.service** が、PostgreSQL や SQLite などの FDO データベースに接続できません。したがって、システムは、所有権バウチャーの保存などの認証情報やその他のパラメーターにサポートされているデータベースを使用して FDO を起動することができません。

この問題を回避するには、次の手順に従います。

1. **local_fdo_update.cil** という名前の新しいファイルを作成し、欠落している SELinux ポリシールールを入力します。

```
(allow fdo_t etc_t (file (write)))
(allow fdo_t fdo_conf_t (file (append create rename setattr unlink write )))
(allow fdo_t fdo_var_lib_t (dir (add_name remove_name write )))
(allow fdo_t fdo_var_lib_t (file (create setattr unlink write )))
(allow fdo_t krb5_keytab_t (dir (search)))
(allow fdo_t postgresql_port_t (tcp_socket (name_connect)))
(allow fdo_t sssd_t (unix_stream_socket (connectto)))
(allow fdo_t sssd_var_run_t (sock_file (write)))
```

2. ポリシーモジュールパッケージをインストールします。

```
# semodule -i local_fdo_update.cil
```

これにより、FDO が PostgreSQL データベースに接続できるようになります。また、SQLite データベースファイルの想定される配置先である **/var/lib/fdo/** の SQLite 権限に関連する問題も修正されます。

[Jira:RHEL-28814](#)

11.3. RHEL FOR EDGE

open-vm-tools パッケージは、**edge-vsphere** イメージでは使用できない

現在、**open-vm-tools** パッケージは、デフォルトでは **edge-vsphere** イメージにインストールされていません。この問題を回避するには、ブループリントのカスタマイズにパッケージを追加します。**edge-vsphere** イメージタイプを使用する場合は、RHEL for Edge Container イメージまたは RHEL for Edge Commit イメージのブループリントに **open-vm-tools** を追加します。

[Jira:RHELDPCS-16574^{\[1\]}](#)

11.4. ソフトウェア管理

インストールプロセスが応答しなくなることがある

RHEL をインストールすると、インストールプロセスが応答しなくなることがあります。**/tmp/packaging.log** ファイルは、最後に以下のメッセージを表示します。

```
10:20:56,416 DDEBUG dnf: RPM transaction over.
```

この問題を回避するには、インストールプロセスを再起動します。

[Bugzilla:2073510](#)

ローカルリポジトリで `createrepo_c` を実行すると、重複した `repodata` ファイルが生成される

ローカルリポジトリで `createrepo_c` コマンドを実行すると、`repodata` ファイルの重複コピーが生成されます。コピーの1つは圧縮されており、もう1つは圧縮されていません。回避策はありませんが、重複したファイルは無視しても問題ありません。`createrepo_c` コマンドは、`createrepo_c` を使用して作成されたリポジトリに依存する要件と他のツールの違いにより、重複したコピーを生成します。

[Bugzilla:2056318](#)

11.5. シェルおよびコマンドラインツール

ifcfg ファイルを使用したネットワークインターフェイスの名前変更に失敗する

RHEL 9 では、`initscripts` はデフォルトでインストールされません。その結果、`ifcfg` ファイルを使用したネットワークインターフェイスの名前変更に失敗します。この問題を解決するには、`udev` ルールを使用するか、ファイルをリンクしてインターフェイスの名前を変更することが推奨されます。詳細は、[一貫したネットワークインターフェイスデバイスの命名](#) および `systemd.link(5)` の man ページを参照してください。

推奨される方法のいずれも使用できない場合は、`initscripts` パッケージをインストールします。

[Bugzilla:2018112^{\[1\]}](#)

RHEL 9 では、chkconfig パッケージがデフォルトでインストールされない

システムサービス用のランレベル情報を更新およびクエリーする `chkconfig` パッケージは、RHEL 9 ではデフォルトでインストールされません。

サービスを管理するには、`systemctl` コマンドを使用するか、`chkconfig` パッケージを手動でインストールします。

`systemd` の詳細は、[systemd の概要](#) を参照してください。`systemctl` ユーティリティの使用方法については、[systemctl を使用したシステムサービスの管理](#) を参照してください。

[Bugzilla:2053598^{\[1\]}](#)

initscripts パッケージがデフォルトでインストールされない

デフォルトでは、`initscripts` パッケージはインストールされません。これにより、`ifup` ユーティリティおよび `ifdown` ユーティリティが利用できません。別の方法として、`nmcli connection up` コマンドおよび `nmcli connection down` コマンドを使用して、接続を有効および無効にします。提案された代替案がうまくいかない場合は、問題を報告し、`NetworkManager-initscripts-updown` パッケージをインストールしてください。これは、`ifup` および `ifdown` ユーティリティ用の NetworkManager ソリューションを提供します。

[Bugzilla:2082303](#)

コンソール `keymap` を設定するには、最小限のインストールで `libxkbcommon` ライブラリーが必要です。

RHEL 9 では、特定の `systemd` ライブラリーの依存関係が動的リンクから動的ロードに変換され、システムが実行時にライブラリーを開いて使用できるようになりました。今回の変更により、必要なライブラリーをインストールしない限り、このようなライブラリーに依存する機能は使用できなくなります。これは、最小限のインストール設定を使用するシステムにおけるキーボードレイアウトの設定にも影響します。その結果、`localectl --no-convert set-x11-keymap gb` コマンドに失敗します。

この問題を回避するには、**libxkbcommon** ライブラリーをインストールします。

```
# dnf install libxkbcommon
```

[Jira:RHEL-6105](#)

sysstat パッケージの %vmeff メトリックに誤った値が表示される

sysstat パッケージは、ページ再利用効率を測定するための **%vmeff** メトリックを提供します。**sysstat** は、新しいカーネルバージョンで提供されるすべての関連する **/proc/vmstat** 値を解析しないため、**sar -B** コマンドによって返される **%vmeff** 列の値は正しくありません。この問題を回避するには、**/proc/vmstat** ファイルから **%vmeff** 値を手動で計算します。詳細は、[Why the sar\(1\) tool reports %vmeff values beyond 100 % in RHEL 8 and RHEL 9?](#) を参照してください。

[Jira:RHEL-12009](#)

Service Location Protocol (SLP) は UDP を介した攻撃に対して脆弱である

OpenSLP は、プリンターやファイルサーバーなどのローカルエリアネットワーク内のアプリケーションに動的設定メカニズムを提供します。ただし、SLP は、インターネットに接続されたシステムで UDP を介した反射型/増幅型サービス拒否攻撃に対して脆弱です。SLP を使用すると、認証されていない攻撃者は、SLP 実装によって設定された制限なしで新しいサービスを登録できます。UDP を使用して送信元アドレスを偽装することにより、攻撃者はサービスリストを要求し、偽装されたアドレスにサービス拒否を作成できます。

外部の攻撃者が SLP サービスにアクセスできないようにするには、インターネットに直接接続されているなど、信頼できないネットワークで実行されているすべてのシステムで SLP を無効にします。または、この問題を回避するには、UDP および TCP ポート 427 でトラフィックをブロックまたはフィルタリングするようにファイアウォールを設定します。

[Jira:RHEL-6995^{\[1\]}](#)

セキュアブートが有効になっている UEFI システム上の ReaR レスキューイメージは、デフォルト設定では起動に失敗する

rear mkrescue または **rear mkbackup** コマンドを使用した ReaR イメージの作成が失敗し、次のメッセージが表示されます。

```
grub2-mkstandalone may fail to make a bootable EFI image of GRUB2 (no /usr/*/grub*/x86_64-efi/moddep.lst file)
(...)
grub2-mkstandalone: error: /usr/lib/grub/x86_64-efi/modinfo.sh doesn't exist. Please specify --target or --directory.
```

不足しているファイルは、**grub2-efi-x64-modules** パッケージの一部です。このパッケージをインストールすると、エラーなしでレスキューイメージが正常に作成されます。**UEFI** セキュアブートが有効になっている場合は、レスキューイメージは署名されていないブートローダーを使用するため起動できません。

この問題を回避するには、**/etc/rear/local.conf** または **/etc/rear/site.conf** ReaR 設定ファイルに次の変数を追加します。

```
UEFI_BOOTLOADER=/boot/efi/EFI/redhat/grubx64.efi
SECURE_BOOT_BOOTLOADER=/boot/efi/EFI/redhat/shimx64.efi
```

提案された回避策を使用すると、**grub2-efi-x64-modules** パッケージのないシステムでもイメージを正

常に生成でき、セキュアブートが有効になっているシステムで起動できるようになります。さらに、システムのリカバリー中に、リカバリーされたシステムのブートローダーは **EFI shim** ブートローダーに設定されます。

UEFI、**セキュアブート**、**shim ブートローダー** の詳細は、ナレッジベースの記事 [UEFI: what happens when booting the system](#) を参照してください。

Jira:RHELDOCS-18064^[1]

sar および **iostat** ユーティリティーによって生成された **%util** 列が無効

sar または **iostat** ユーティリティーを使用してシステム使用状況の統計情報を収集する場合、**sar** または **iostat** によって生成された **%util** 列に無効なデータが含まれることがあります。

Jira:RHEL-26275^[1]

lsb-release バイナリーは RHEL 9 では利用できない

`/etc/os-release` の情報は、以前は **lsb-release** バイナリーを呼び出すことで入手できました。このバイナリーは **redhat-lsb package** に含まれていましたが、このパッケージは RHEL 9 では削除されました。現在は、`/etc/os-release` ファイルを読み取ることで、オペレーティングシステムに関する情報 (ディストリビューション、バージョン、コード名、関連するメタデータなど) を表示できるようになりました。このファイルは Red Hat が提供しており、このファイルに対する変更は **redhat-release** パッケージを更新するたびに上書きされます。ファイルの形式は **KEY=VALUE** であり、シェルスクリプトのデータを安全に取得できます。

Jira:RHELDOCS-16427^[1]

11.6. インフラストラクチャーサービス

bind および **unbound** の両方が **SHA-1** ベースの署名の検証を無効化する

bind および **unbound** コンポーネントは、すべての **RSA/SHA1** (アルゴリズム番号 5) および **RSASHA1-NSEC3-SHA1** (アルゴリズム番号 7) 署名の検証サポートを無効にし、署名の **SHA-1** 使用は **DEFAULT** システム全体の暗号化ポリシーで制限されます。

その結果、**SHA-1**、**RSA/SHA1**、および **RSASHA1-NSEC3-SHA1** ダイジェストアルゴリズムで署名された特定の **DNSSEC** レコードは、Red Hat Enterprise Linux 9 で検証できず、影響を受けるドメイン名が脆弱になります。

この問題を回避するには、**RSA/SHA-256** や楕円曲線キーなどの別の署名アルゴリズムにアップグレードします。

影響を受け脆弱なトップレベルドメインの詳細とリストについては、[RSASHA1 で署名された DNSSEC レコードがソリューションを検証できない](#) を参照してください。

Bugzilla:2070495

同じ書き込み可能ゾーンファイルが複数のゾーンで使用されていると、**named** が起動しない

BIND では、複数のゾーンに同じ書き込み可能ゾーンファイルを使用することができません。そのため、**named** で変更可能なファイルへのパスを共有するゾーンが複数存在すると、**named** が起動できなくなります。この問題を回避するには、**in-view** 節を使用して、複数のビュー間で1つのゾーンを共有し、異なるゾーンに異なるパスを使用するようにします。たとえば、パスにビュー名を含めます。

書き込み可能なゾーンファイルは通常、動的更新が許可されたゾーン、セカンダリーゾーン、または **DNSSEC** が管理するゾーンで使用されることに注意してください。

[Bugzilla:1984982](#)

libotr は FIPS に準拠していない

libotr ライブラリーとオフザレコード (OTR) メッセージング用のツールキットは、インスタントメッセージングの会話にエンドツーエンドの暗号化を提供します。ただし、**libotr** ライブラリーは **gcry_pk_sign()** および **gcry_pk_verify()** 関数を使用しているため、連邦情報処理標準 (FIPS) に準拠していません。その結果、FIPS モードでは **libotr** ライブラリーを使用できません。

[Bugzilla:2086562](#)

11.7. ネットワーク

kTLS は、TLS 1.3 の NIC へのオフロードをサポートしない

Kernel Transport Layer Security(kTLS) は、TLS 1.3 の NIC へのオフロードをサポートしていません。そのため、NIC が TLS オフロードをサポートしていても、TLS 1.3 によるソフトウェア暗号化が使用されます。この問題を回避するには、オフロードが必要な場合は TLS 1.3 を無効にしてください。その結果、TLS 1.2 のみをオフロードすることができます。TLS 1.3 が使用されている場合、TLS 1.3 をオフロードすることができないため、パフォーマンスが低下します。

[Bugzilla:2000616^{\[1\]}](#)

セッションキーの更新に失敗すると、接続が切断される

カーネルトランスポートレイヤーセキュリティ (kTLS) プロトコルは、対称暗号で使用されるセッションキーの更新をサポートしていません。その結果、ユーザーはキーを更新することができず、接続が切断されてしまいます。この問題を回避するには、kTLS を無効にしてください。その結果、この回避策により、セッションキーを正常に更新できます。

[Bugzilla:2013650^{\[1\]}](#)

11.8. カーネル

カーネルページサイズに依存する顧客アプリケーションは、ページサイズカーネルを 4k から 64k に移行するときに更新が必要になる場合がある

RHEL は、4k と 64k の両方のページサイズのカーネルと互換性があります。4K カーネルページサイズに依存する顧客アプリケーションは、4K から 64K ページサイズカーネルに移行するときに更新が必要になる場合があります。この既知の例には、**jemalloc** および依存アプリケーションが含まれます。

jemalloc メモリーアロケータライブラリーは、システムのランタイム環境で使用されるページサイズの影響を受けます。このライブラリーは、たとえば、**--with-lg-page=16** または **env JEMALLOC_SYS_WITH_LG_PAGE=16** (**jemallocator** Rust クレートの場合) で設定されている場合、4k および 64k ページサイズのカーネルと互換性があるように構築できます。その結果、ランタイム環境のページサイズと、**jemalloc** に依存するバイナリーのコンパイル時に存在したページサイズとの間に不一致が発生する可能性があります。その結果、**jemalloc** ベースのアプリケーションを使用すると、次のエラーが発生します。

```
<jemalloc>: Unsupported system page size
```

この問題を回避するには、次のいずれかの方法を使用します。

- 適切なビルド設定または環境オプションを使用して、4k および 64k ページサイズと互換性のあるバイナリーを作成します。

- 最終的な 64k カーネルおよびランタイム環境で起動した後、**jemalloc** を使用するユーザー空間パッケージをビルドします。

たとえば、同じく **jemalloc** を使用する **fd-find** ツールを、**cargo** Rust パッケージマネージャーを使用して構築できます。最後の 64k 環境では、**cargo** コマンドを入力して、すべての依存関係の新しいビルドをトリガーし、ページサイズの不一致を解決します。

```
# cargo install fd-find --force
```

Bugzilla:2167783^[1]

dnf を使用して最新のリアルタイムカーネルにアップグレードしても、複数のカーネルバージョンは並行してインストールされない

dnf パッケージマネージャーを使用して最新のリアルタイムカーネルをインストールするには、パッケージの依存関係を解決して、新しいカーネルバージョンと現在のカーネルバージョンを同時に保持する必要があります。デフォルトでは、**dnf** は、アップグレード中に古い **kernel-rt** パッケージを削除します。

回避策として、現在の **kernel-rt** パッケージを **/etc/yum.conf** 設定ファイルの **installonlypkgs** オプションに追加します (例: **installonlypkgs=kernel-rt**)。

installonlypkgs オプションは、**dnf** で使用されるデフォルトのリストに **kernel-rt** を追加します。**installonlypkgs** ディレクティブにリストされているパッケージは自動的に削除されないため、複数のカーネルバージョンの同時インストールがサポートされます。

複数のカーネルをインストールすると、新しいカーネルバージョンを使用するときにフォールバックオプションを使用できるようになります。

Bugzilla:2181571^[1]

デフォルトでは、Delay Accounting 機能は SWAPIN および IO% 統計列を表示しない

初期のバージョンとは異なり、**Delayed Accounting** 機能はデフォルトで無効になっています。その結果、**iotop** アプリケーションは **SWAPIN** および **IO%** 統計列を表示せず、次の警告を表示します。

```
CONFIG_TASK_DELAY_ACCT not enabled in kernel, cannot determine SWAPIN and IO%
```

taskstats インターフェイスを使用する **Delay Accounting** 機能は、スレッドグループに属するすべてのタスクまたはスレッドの遅延統計を提供します。タスク実行の遅延は、カーネルリソースが利用可能になるのを待つときに発生します。たとえば、空き CPU が実行されるのを待っているタスクです。統計は、タスクの CPU 優先度、I/O 優先度、および **rss** 制限値を適切に設定するのに役立ちます。

回避策として、実行時または起動時に、**delayacct** ブートオプションを有効にすることができます。

- 実行時に **delayacct** を有効にするには、次のように入力します。

```
echo 1 > /proc/sys/kernel/task_delayacct
```

このコマンドはシステム全体で機能を有効にしますが、このコマンドの実行後に開始したタスクに対してのみ有効であることに注意してください。

- 起動時に **delayacct** を永続的に有効にするには、次のいずれかの手順を使用します。
 - **/etc/sysctl.conf** ファイルを編集して、デフォルトのパラメーターをオーバーライドします。

- a. 次のエントリーを `/etc/sysctl.conf` ファイルに追加します。

```
kernel.task_delayacct = 1
```

詳細は、[Red Hat Enterprise Linux で sysctl 変数を設定する方法](#) を参照してください。

- b. システムを再起動して、変更を反映させます。
- カーネルコマンドラインに **delayacct** オプションを追加します。
詳細は、[カーネルコマンドラインパラメーターの設定](#) を参照してください。

その結果、**iotop** アプリケーションは **SWAPIN** および **IO%** 統計列を表示します。

Bugzilla:2132480^[1]

コア数が多いシステム上のリアルタイムカーネルのハードウェア認定には、`skew-tick=1` ブートパラメーターを渡すことが必要になる場合がある

多数のソケットとコアカウントが大きい大規模なシステムまたは中規模のシステムでは、タイムキーピングシステムで使用される `xtime_lock` のロック競合により、レイテンシーの急増が発生する可能性があります。その結果、レイテンシーの急増およびハードウェア認証のレイテンシーは、マルチプロセッシングシステムで発生する可能性があります。回避策として、`skew_tick=1` ブートパラメーターを追加することで、CPU ごとにタイマーティックをオフセットし、別のタイミングで開始できます。

ロックの競合を回避するには、`skew_tick=1` を有効にします。

- grubby** で `skew_tick=1` パラメーターを有効にします。

```
# grubby --update-kernel=ALL --args="skew_tick=1"
```

- 変更を有効にするために再起動します。
- ブート中に渡すカーネルパラメーターを表示して、新しい設定を確認します。

```
cat /proc/cmdline
```

`skew_tick=1` を有効にすると、消費電力が大幅に増加するため、レイテンシーの影響を受けるリアルタイムワークロードを実行している場合にのみ有効にする必要があります。

Jira:RHEL-9318^[1]

kdump メカニズムは、LUKS 暗号化ターゲットで **vmcore** ファイルをキャプチャーできない

Linux Unified Key Setup (LUKS) で暗号化されたパーティションを使用するシステムで **kdump** を実行する場合、システムには一定量の使用可能なメモリーが必要です。使用可能なメモリーが必要なメモリー量より少ない場合、**systemd-cryptsetup** サービスはパーティションのマウントに失敗します。その結果、2 番目のカーネルは LUKS 暗号化ターゲット上のクラッシュダンプファイルのキャプチャーに失敗します。

回避策として、**Recommended crashkernel value** をクエリーし、メモリーサイズを適切な値まで徐々に増やします。**Recommended crashkernel value** は、必要なメモリーサイズを設定するための参考として役立ちます。

- クラッシュカーネルの推定値を出力します。

```
# kdumpectl estimate
```

-
2. **crashkernel** の値を増やして、必要なメモリー量を設定します。

```
# grubby --args=crashkernel=652M --update-kernel=ALL
```

3. システムを再起動して、変更を反映させます。

```
# reboot
```

これにより、LUKS で暗号化したパーティションがあるシステムで **kdump** が正常に機能します。

Jira:RHEL-11196^[1]

kdump サービスが IBM Z システムで **initrd** ファイルの構築に失敗する

64 ビットの IBM Z システムでは、**s390-** subchannels などの **znet** 関連の設定情報が非アクティブな **NetworkManager** 接続プロファイルに存在する場合、**kdump** サービスは初期 RAM ディスク (**initrd**) のロードに失敗します。その結果、**kdump** メカニズムは次のエラーで失敗します。

```
dracut: Failed to set up znet
kdump: mkdumprd: failed to make kdump initrd
```

回避策として、次のいずれかの解決策を使用してください。

- **znet** 設定情報を持つ接続プロファイルを再利用して、ネットワークボンディングまたはブリッジを設定します。

```
$ nmcli connection modify enc600 master bond0 slave-type bond
```

- 非アクティブな接続プロファイルからアクティブな接続プロファイルに **znet** 設定情報をコピーします。

- a. **nmcli** コマンドを実行して、**NetworkManager** 接続プロファイルを照会します。

```
# nmcli connection show

NAME                UUID                TYPE  Device
bridge-br0         ed391a43-bdea-4170-b8a2 bridge  br0
bridge-slave-enc600 caf7f770-1e55-4126-a2f4 ethernet enc600
enc600             bc293b8d-ef1e-45f6-bad1 ethernet --
```

- b. 非アクティブな接続からの設定情報でアクティブなプロファイルを更新します。

```
#!/bin/bash
inactive_connection=enc600
active_connection=bridge-slave-enc600
for name in nettype subchannels options; do
field=802-3-ethernet.s390-$name
val=$(nmcli --get-values "$field"connection show "$inactive_connection")
nmcli connection modify "$active_connection" "$field" $val
done
```

- c. 変更を有効にするために **kdump** サービスを再起動します。

```
# kdumpctl restart
```

[Bugzilla:2064708](#)

iwl7260-firmware により、Intel Wi-Fi 6 AX200、AX210、および Lenovo ThinkPad P1 Gen 4 で Wi-Fi が切断される

iwl7260-firmware または **iwl7260-wifi** ドライバーを RHEL 9.1以降で提供されるバージョンに更新すると、ハードウェアが不正な内部状態になります。その状態を誤って報告します。その結果、Intel Wifi 6 カードが機能せず、次のエラーメッセージが表示される場合があります。

```
kernel: iwlwifi 0000:09:00.0: Failed to start RT ucode: -110
kernel: iwlwifi 0000:09:00.0: WRT: Collecting data: ini trigger 13 fired (delay=0ms)
kernel: iwlwifi 0000:09:00.0: Failed to run INIT ucode: -110
```

未確認の回避策は、システムの電源をオフにしてから再度オンにすることです。再起動しないでください。

[Bugzilla:2129288^{\[1\]}](#)

kmod の weak-modules がモジュールの相互依存関係で機能しない

kmod パッケージによって提供される **weak-modules** スクリプトは、どのモジュールがインストールされたカーネルと kABI 互換であるかを判別します。ただし、モジュールのカーネル互換性をチェックしている間、**weak-modules** はモジュールシンボルの依存関係を、それらがビルドされたカーネルの上位リリースから下位リリースへと処理します。結果として、異なるカーネルリリースに対して構築された相互依存関係を持つモジュールは互換性がないと解釈される可能性があるため、**weak-modules** はこのシナリオでは機能しません。

この問題を回避するには、新しいカーネルをインストールする前に、最新のストックカーネルに対して追加のモジュールをビルドまたは配置します。

[Bugzilla:2103605^{\[1\]}](#)

Intel® i40e アダプターが IBM Power10 で永続的に失敗する

IBM Power10 システムで **i40e** アダプターに I/O エラーが発生すると、Enhanced I/O Error Handling (EEH) カーネルサービスがネットワークドライバーのリセットトリガーをトリガーします。しかし、EEH は、**i40e** ドライバーが事前定義された最大値に達して EEH が応答を停止するまで、I/O エラーを繰り返し報告します。その結果、デバイスは EEH によって永続的に失敗します。

[Jira:RHEL-15404^{\[1\]}](#)

64 ビット ARM CPU で正しくコンパイルされたドライバーでのプログラム失敗に関して dkms が誤った警告を出す

Dynamic Kernel Module Support (**dkms**) ユーティリティーは、4 kB および 64 kB のページサイズのカーネルで 64 ビット ARM CPU のカーネルヘッダーが動作することを認識しません。その結果、**dkms** は、カーネルの更新時に **kernel-64k-devel** パッケージがインストールされていない場合、正しくコンパイルされたドライバーでプログラムが失敗した理由に関して誤った警告を出します。この問題を回避するには、**kernel-headers** パッケージをインストールします。このパッケージは、両タイプの ARM CPU アーキテクチャー用のヘッダーファイルを含むもので、**dkms** とその要件に特化したものではありません。

[Jira:RHEL-25967^{\[1\]}](#)

11.9. ファイルシステムおよびストレージ

デバイスマッパーマルチパスは NVMe/TCP ではサポートされていません

nvme-tcp ドライバーで Device Mapper Multipath を使用すると、コールトレースの警告とシステムの不安定性が発生する可能性があります。この問題を回避するには、NVMe/TCP ユーザーはネイティブ NVMe マルチパスを有効にする必要があります、NVMe で **device-mapper-multipath** ツールを使用しないでください。

デフォルトでは、ネイティブ NVMe マルチパスは RHEL 9 で有効になっています。詳細は、[Enabling multipathing on NVMe devices](#) を参照してください。

Bugzilla:2033080^[1]

blk-availability systemd サービスは、複雑なデバイススタックを非アクティブ化する

systemd では、デフォルトのブロック非アクティブ化コードは、仮想ブロックデバイスの複雑なスタックを常に正しく処理するとは限りません。一部の設定では、シャットダウン中に仮想デバイスが削除されない場合があります、エラーメッセージがログに記録されます。この問題を回避するには、次のコマンドを実行して、複雑なブロックデバイススタックを非アクティブ化します。

```
# systemctl enable --now blk-availability.service
```

その結果、複雑な仮想デバイススタックはシャットダウン中に正しく非アクティブ化され、エラーメッセージは生成されません。

Bugzilla:2011699^[1]

クォータを有効にしてマウントされた XFS ファイルシステムでは、クォータアカウンティングを無効化できなくなる

RHEL 9.2 以降、クォータを有効にしてマウントされた XFS ファイルシステムで、クォータアカウンティングを無効にすることはできなくなりました。

この問題を回避するには、クォータオプションを削除してファイルシステムを再マウントし、クォータアカウンティングを無効にします。

Bugzilla:2160619^[1]

NVMe デバイスの udev ルールの変更

NVMe デバイスの udev ルールに変更があり、**OPTIONS="string_escape=replace"** パラメーターが追加されました。これにより、デバイスのシリアル番号の先頭に空白がある場合、一部のベンダーではディスク ID による名前が変更されます。

Bugzilla:2185048

NVMe/FC デバイスはキックスタートファイルで確実に使用できない

NVMe/FC デバイスは、キックスタートファイルの解析中または事前スクリプトの実行中に利用できなくなる可能性があり、キックスタート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、ブート引数を **inst.wait_for_disks=30** に更新します。このオプションでは 30 秒の遅延が発生しますが、NVMe/FC デバイスの接続に十分な時間が確保されます。この回避策を適用し、NVMe/FC デバイスが時間内に接続されると、キックスタートインストールは問題なく続行します。

Jira:RHEL-8164^[1]

qedi ドライバー使用時のカーネルパニック

qedi iSCSI ドライバーの使用時、OS の起動後にカーネルがパニックになります。この問題を回避するには、カーネルブートコマンドラインに **kfence.sample_interval=0** を追加して、**kfence** ランタイムメモリーエラー検出機能を無効にします。

Jira:RHEL-8466^[1]

vdo がインストールされている場合、ARM ベースのシステムが 64k ページサイズのカーネルで更新できない

vdo パッケージのインストール時に、RHEL は **kmod-kvdo** パッケージと **4k** ページサイズのカーネルを依存関係としてインストールします。その結果、**kmod-kvdo** が 64k カーネルと競合するため、RHEL 9.3 から 9.x への更新が失敗します。この問題を回避するには、更新を試みる前に **vdo** パッケージとその依存関係を削除します。

Jira:RHEL-8354

lldpad が qedf アダプターでも自動的に有効になる

QLogic Corp. の FastLinQ QL45000 Series 10/25/40/50GbE を使用すると、FCOE コントローラーが、Red Hat Virtualization を実行しているシステムで **lldpad** デモンを自動的に有効にします。その結果、I/O 操作はエラーで中断します (例: **[qedf_ah_abort:xxxx]:1: Aborting io_req=ff5d85a9dcf3xxxx**)。

この問題を回避するには、Link Layer Discovery Protocol (LLDP) を無効にしてから、**vdsmd** 設定レベルで設定可能なインターフェイスに対して有効にします。詳細は、<https://access.redhat.com/solutions/6963195> を参照してください。

Jira:RHEL-8104^[1]

iommu が有効化されている場合はシステムが起動しない

BNX2I アダプターの使用中に AMD プラットフォームで Input-Output Memory Management Unit (IOMMU) を有効にすると、システムは Direct Memory Access Remapping (DMAR) タイムアウトエラーで起動に失敗します。この問題を回避するには、カーネルコマンドラインオプション **iommu=off** を使用して、起動前に IOMMU を無効にします。その結果、システムはエラーなしで起動します。

Jira:RHEL-25730^[1]

11.10. 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

python3.11-lxml は lxml.isoschematron サブモジュールを提供しない

python3.11-lxml パッケージは、オープンソースライセンスの下にないため、**lxml.isoschematron** サブモジュールなしで配布されます。サブモジュールは ISO Schematron サポートを実装します。代わりに、ISO-Schematron 前の検証を **lxml.etree.Schematron** クラスで利用できます。**python3.11-lxml** パッケージの残りのコンテンツは影響を受けません。

Bugzilla:2157708

MySQL および MariaDB の --ssl-fips-mode オプションでは FIPS モードが変更されない

MySQL の **--ssl-fips-mode** オプションと RHEL の **MariaDB** は、アップストリームとは異なる動作をします。

RHEL 9 では、`--ssl-fips-mode` を `mysqld` デーモンまたは `mariadb` デーモンの引数として使用する場
合や、**MySQL** または **MariaDB** サーバー設定ファイルに `ssl-fips-mode` を使用すると、`--ssl-fips-
mode` はこれらのデータベースサーバーの FIPS モードを変更しません。

代わりに、以下のようになります。

- `--ssl-fips-mode` を **ON** に設定すると、`mysqld` サーバーデーモンまたは `mariadb` サーバーデーモンは起動しません。
- FIPS が有効なシステムで `--ssl-fips-mode` を **OFF** に設定すると、`mysqld` サーバーデーモンまたは `mariadb` サーバーデーモンは FIPS モードで稼働します。

これは、特定のコンポーネントではなく、RHEL システム全体で FIPS モードを有効または無効にする必要があるためです。

したがって、RHEL の **MySQL** または **MariaDB** では `--ssl-fips-mode` オプションを使用しないでください。代わりに、FIPS モードが RHEL システム全体で有効になっていることを確認します。

- FIPS モードが有効な RHEL をインストールすることが推奨されます。インストール時に FIPS モードを有効にすると、システムは FIPS で承認されるアルゴリズムと継続的な監視テストですべての鍵を生成するようになります。FIPS モードで RHEL をインストールする方法は、[FIPS モードでのシステムのインストール](#) を参照してください。
- または、[FIPS モードへのシステムの切り替え](#) の手順に従って、RHEL システム全体の FIPS モードを切り替えることができます。

[Bugzilla:1991500](#)

11.11. IDENTITY MANAGEMENT

MIT Kerberos は PKINIT の ECC 証明書をサポートしません

MIT Kerberos は、初期認証 (PKINIT) の公開鍵暗号化における楕円曲線暗号化 (ECC) サポートの設計を説明するコメントドキュメントに、RFC5349 要求を実装していません。したがって、RHEL で使用される MIT `krb5-pkinit` パッケージは ECC 証明書に対応していません。詳細は、[Elliptic Curve Cryptography \(ECC\) Support for Public Key Cryptography for Initial Authentication in Kerberos \(PKINIT\)](#) を参照してください。

[Jira:RHEL-4902](#)

PKINIT が AD KDC に対して機能するように、DEFAULT:SHA1 サブポリシーを RHEL 9 クライアントに設定する必要がある

SHA-1 ダイジェストアルゴリズムは RHEL 9 で非推奨になり、初期認証 (PKINIT) の公開鍵暗号化の CMS メッセージは、より強力な SHA-256 アルゴリズムで署名されるようになりました。

しかし、Active Directory (AD) Kerberos Distribution Center (KDC) は引き続き SHA-1 ダイジェストアルゴリズムを使用して CMS メッセージに署名します。その結果、RHEL 9 Kerberos クライアントは、AD KDC に対して PKINIT を使用してユーザーを認証できません。

この問題を回避するには、次のコマンドを使用して、RHEL 9 システムで SHA-1 アルゴリズムのサポートを有効にします。

```
# update-crypto-policies --set DEFAULT:SHA1
```

[Bugzilla:2060798](#)

RHEL 9 Kerberos エージェントが RHEL-9 以外および AD 以外の Kerberos エージェントと通信すると、ユーザーの PKINIT 認証に失敗する

クライアントまたは Kerberos Distribution Center (KDC) のいずれかの RHEL 9 Kerberos エージェントが、Active Directory (AD) エージェントではない RHEL-9 Kerberos エージェントとやりとりすると、ユーザーの PKINIT 認証に失敗します。この問題を回避するには、以下のいずれかのアクションを実行します。

- RHEL 9 エージェントの `crypto-policy` を **DEFAULT:SHA1** に設定して、SHA-1 署名の検証を許可します。

```
# update-crypto-policies --set DEFAULT:SHA1
```

- RHEL 9 以外および AD 以外のエージェントを更新して、SHA-1 アルゴリズムを使用して CMS データを署名しないようにします。そのためには、Kerberos パッケージを SHA-1 の代わりに SHA-256 を使用するバージョンに更新します。
 - CentOS 9 Stream: krb5-1.19.1-15
 - RHEL 8.7: krb5-1.18.2-17
 - RHEL 7.9: krb5-1.15.1-53
 - Fedora Rawhide/36: krb5-1.19.2-7
 - Fedora 35/34: krb5-1.19.2-3

その結果、ユーザーの PKINIT 認証が正しく機能します。

他のオペレーティングシステムでは、エージェントが SHA-1 ではなく SHA-256 で CMS データを署名するように krb5-1.20 リリースであることに注意してください。

PKINIT が AD KDC に対して機能するように、[DEFAULT:SHA1 サブポリシー](#)を RHEL 9 クライアントに設定する必要があるも併せて参照してください。

[Jira:RHEL-4875](#)

AD 信頼の FIPS サポートには、AD-SUPPORT 暗号サブポリシーが必要

Active Directory (AD) は、AES SHA-1 HMAC 暗号化タイプを使用します。これは、デフォルトで RHEL 9 の FIPS モードでは許可されていません。AD トラストで RHEL 9 ホストを使用する場合は、IdM ソフトウェアをインストールする前に、AESSHA-1HMAC 暗号化タイプのサポートを有効にしてください。

FIPS 準拠は技術的合意と組織的合意の両方を伴うプロセスであるため、**AD-SUPPORT** サブポリシーを有効にして技術的手段が AES SHA-1 HMAC 暗号化タイプをサポートできるようにする前に、FIPS 監査人に相談してから、RHEL IdM をインストールしてください。

```
# update-crypto-policies --set FIPS:AD-SUPPORT
```

[Bugzilla:2057471](#)

Heimdal クライアントは、RHEL 9 KDC に対して PKINIT を使用してユーザーを認証できない

デフォルトでは、Heimdal Kerberos クライアントは、Internet Key Exchange (IKE) に Modular Exponential (MODP) Diffie-Hellman Group 2 を使用して、IdM ユーザーの PKINIT 認証を開始します。ただし、RHEL 9 の MIT Kerberos Distribution Center (KDC) は、MODP Group 14 および 16 のみに対応しています。

したがって、Heimdal クライアントで **krb5_get_init_creds: PREAUTH_FAILED** エラーが発生し、RHEL MIT KDC では **Key parameters not accepted** が発生します。

この問題を回避するには、Heimdal クライアントが MODP Group 14 を使用していることを確認してください。クライアント設定ファイルの **libdefaults** セクションで **pkinit_dh_min_bits** パラメーターを 1759 に設定します。

```
[libdefaults]
pkinit_dh_min_bits = 1759
```

その結果、Heimdal クライアントは、RHEL MIT KDC に対する PKINIT 事前認証を完了します。

[Jira:RHEL-4889](#)

FIPS モードの IdM は、双方向のフォレスト間信頼を確立するための NTLMSSP プロトコルの使用をサポートしない

FIPS モードが有効な Active Directory (AD) と Identity Management (IdM) との間で双方向のフォレスト間の信頼を確立すると、New Technology LAN Manager Security Support Provider (NTLMSSP) 認証が FIPS に準拠していないため、失敗します。FIPS モードの IdM は、認証の試行時に AD ドメインコントローラーが使用する RC4 NTLM ハッシュを受け入れません。

[Jira:RHEL-12154](#)^[1]

SID のないユーザーは、アップグレード後に IdM にログインできない

IdM レプリカを RHEL 9.2 にアップグレードした後、IdM Kerberos Distribution Center (KDC) は、アカウントにセキュリティー識別子 (SID) が割り当てられていないユーザーに Ticket-Granting Ticket (TGT) を発行できない場合があります。その結果、ユーザーは自分のアカウントにログインできなくなります。

この問題を回避するには、トポロジー内の別の IdM レプリカで IdM 管理者として次のコマンドを実行して SID を生成します。

```
# ipa config-mod --enable-sid --add-sids
```

その後もユーザーがログインできない場合は、Directory Server のエラーログを調べてください。ユーザーの POSIX ID を含めるように ID 範囲を調整する必要がある場合があります。

詳細は、ナレッジベースのソリューション記事 [When upgrading to RHEL9, IDM users are not able to login anymore](#) を参照してください。

[Jira:RHELPLAN-157939](#)^[1]

ドメイン SID の不一致により、移行した IdM ユーザーがログインできない可能性がある

ipa migrate-ds スクリプトを使用して IdM デプロイメントから別のデプロイメントにユーザーを移行する場合、そのユーザーの以前のセキュリティー識別子 (SID) には現在の IdM 環境のドメイン SID がないため、ユーザーが IdM サービスを使用する際に問題が発生する可能性があります。たとえば、これらのユーザーは **kinit** ユーティリティーを使用して Kerberos チケットを取得できますが、ログインできません。この問題を回避するには、ナレッジベースの記事 [Migrated IdM users unable to log in due to mismatching domain SIDs](#) を参照してください。

[Jira:RHELPLAN-109613](#)^[1]

ユーザー PAC を生成する暗号化タイプに互換性がないため、MIT krb5 ユーザーは AD TGT の取得に失敗する

MIT **krb5 1.20** 以降のパッケージでは、デフォルトですべての Kerberos チケットに特権属性証明書 (PAC) が含まれています。MIT Kerberos Distribution Center (KDC) は、PAC で KDC チェックサムを生成するために使用できる最も強力な暗号化タイプを選択します。これは現在、RFC8009 で定義されている **AES HMAC-SHA2** 暗号化タイプです。ただし、Active Directory (AD) はこの RFC をサポートしていません。その結果、AD-MIT クロスレルム設定では、MIT KDC によって生成されたクロスレルム TGT に互換性のない KDC チェックサムタイプが PAC に含まれているため、MIT **krb5** ユーザーは AD チケット認可チケット (TGT) を取得できません。

この問題を回避するには、`/var/kerberos/krb5kdc/kdc.conf` 設定ファイルの `[realms]` セクションで MIT レルムの `disable_pac` パラメーターを `true` に設定します。その結果、MIT KDC は PAC なしでチケットを生成します。これは、AD が失敗したチェックサム検証をスキップし、MIT **krb5** ユーザーが AD TGT を取得できることを意味します。

[Bugzilla:2016312](#)

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^{\[1\]}](#)

RHEL 8.6 以前で初期化された FIPS モードの IdM デプロイメントに FIPS モードの RHEL 9 レプリカを追加すると失敗する

FIPS 140-3 への準拠を目的としたデフォルトの RHEL 9 FIPS 暗号化ポリシーでは、RFC3961 のセクション 5.1 で定義されている AES HMAC-SHA1 暗号化タイプのキー派生関数の使用が許可されていません。

この制約は、最初のサーバーが RHEL 8.6 システム以前にインストールされている FIPS モードの RHEL 8 IdM 環境に、FIPS モードの RHEL 9 Identity Management (IdM) レプリカを追加する際の障害となります。これは、AES HMAC-SHA1 暗号化タイプを一般的に使用し、AES HMAC-SHA2 暗号化タイプを使用しない、RHEL 9 と以前の RHEL バージョンの間に共通の暗号化タイプがないためです。

サーバーで次のコマンドを入力すると、IdM マスターキーの暗号化タイプを表示できます。

```
# kadmin.local getprinc K/M | grep -E '^Key:'
```

この問題を回避するには、RHEL 9 レプリカで AES HMAC-SHA1 の使用を有効にします。

```
update-crypto-policies --set FIPS:AD-SUPPORT
```

WARNING

この回避策は FIPS 準拠に違反する可能性があります。

その結果、RHEL 9 レプリカの IdM デプロイメントへの追加が正しく進行します。

RHEL 7 および RHEL 8 サーバー上で欠落している AES HMAC-SHA2 暗号化 Kerberos キーを生成する手順を提供する作業が進行中であることを注意してください。これにより、RHEL 9 レプリカで FIPS 140-3 準拠が達成されます。ただし、Kerberos キー暗号化の設計により、既存のキーを別の暗号化タイプに変換することができないため、このプロセスは完全には自動化されません。唯一の方法は、ユーザーにパスワードの更新を求めることです。

[Jira:RHEL-4888](#)

SSSD は DNS 名を適切に登録する

以前は、DNS が正しく設定されていない場合、SSSD は DNS 名の登録の最初の試行で常に失敗していました。この問題を回避するために、この更新では新しいパラメーター `dns_resolver_use_search_list` が提供されます。DNS 検索リストの使用を回避するには、`dns_resolver_use_search_list = false` を設定します。

[Bugzilla:1608496^{\[1\]}](#)

EMS 強制により、FIPS モードで RHEL 9.2 以降の IdM サーバーを使用した RHEL 7 IdM クライアントのインストールが失敗する

FIPS 対応の RHEL 9.2 以降のシステムで、TLS 1.2 接続に TLS **Extended Master Secret** (EMS) エクステンション (RFC 7627) が必須になりました。これは FIPS-140-3 要件に準拠しています。ただし、RHEL 7.9 以前で利用可能な `openssl` バージョンは EMS をサポートしていません。その結果、RHEL 9.2 以降で実行されている FIPS 対応の IdM サーバーに RHEL 7 Identity Management (IdM) クライアントをインストールすると失敗します。

IdM クライアントをインストールする前にホストを RHEL 8 にアップグレードできない場合は、FIPS 暗号化ポリシーに加えて NO-ENFORCE-EMS サブポリシーを適用して、RHEL 9 サーバーでの EMS 使用の要件を削除することで問題を回避します。

```
# update-crypto-policies --set FIPS:NO-ENFORCE-EMS
```

この削除は FIPS 140-3 要件に反することに注意してください。その結果、EMS を使用しない TLS 1.2 接続を確立して受け入れることができ、RHEL 7 IdM クライアントのインストールは成功します。

[Jira:RHEL-4955](#)

オンラインバックアップとオンライン自動メンバーシップ再ビルドタスクは、2つのロックを取得し、デッドロックを引き起こす可能性があります。

オンラインバックアップとオンライン自動メンバーシップ再ビルドタスクが同じ2つのロックを逆の順序で取得しようとする、回復不能なデッドロックが発生し、サーバーを停止して再起動する必要が生じる可能性があります。この問題を回避するには、オンラインバックアップとオンライン自動メンバーシップ再ビルドタスクを並行して起動しないでください。

[Jira:RHELDPCS-18065^{\[1\]}](#)

11.12. デスクトップ

RHEL 9 へのアップグレード後に VNC が実行されていない

RHEL8 から RHEL 9 にアップグレードした後、以前に有効にされていたとしても、VNC サーバーは起動に失敗します。

この問題を回避するには、システムのアップグレード後に **vncserver** サービスを手動で有効にします。

```
# systemctl enable --now vncserver@:port-number
```

その結果、VNC が有効になり、システムが起動するたびに期待どおりに起動します。

[Bugzilla:2060308](#)

User Creation 画面が応答しない

グラフィカルユーザーインターフェイスを使用して RHEL をインストールすると、User Creation の画面が応答しなくなります。そのため、インストール中にユーザーを作成するのが困難です。

この問題を回避するには、以下のソリューションのいずれかを使用してユーザーを作成します。

- VNC モードでインストールを実行し、VNC ウィンドウのサイズを変更します。
- インストールプロセスの完了後にユーザーを作成します。

[Jira:RHEL-11924](#)^[1]

WebKitGTK が IBM Z で Web ページの表示に失敗する

WebKitGTK Web ブラウザーエンジンは、IBM Z アーキテクチャーで Web ページを表示しようとするとき失敗します。Web ページは空白のまま、WebKitGTK プロセスが予期せず終了します。

その結果、WebKitGTK を使用して Web ページを表示するアプリケーションの次のような特定の機能を使用できなくなります。

- Evolution メールクライアント
- GNOME オンラインアカウント設定
- GNOME ヘルプアプリケーション

[Jira:RHEL-4157](#)

11.13. グラフィックインフラストラクチャー

NVIDIA ドライバーが X.org に戻る可能性がある

特定の条件下では、プロプライエタリー NVIDIA ドライバーは Wayland ディスプレイプロトコルを無効にし、X.org ディスプレイサーバーに戻ります。

- NVIDIA ドライバーのバージョンが 470 未満の場合。
- システムがハイブリッドグラフィックスを使用するラップトップの場合。
- 必要な NVIDIA ドライバーオプションを有効にしていない場合。

また、Wayland は有効になっていますが、NVIDIA ドライバーのバージョンが 510 未満の場合には、デスクトップセッションはデフォルトで X.org を使用します。

[Jira:RHELPLAN-119001](#)^[1]

ナイトライトは、NVIDIA の Wayland では利用できない

システムで独自の NVIDIA ドライバーが有効になっている場合、GNOME のナイトライト機能は Wayland セッションでは使用できません。NVIDIA ドライバーは、現在 **Night Light** をサポートしていません。

Jira:RHELPLAN-119852^[1]

Wayland では X.org 設定ユーティリティーが動作しない

画面を操作するための X.org ユーティリティーは、Wayland セッションでは機能しません。特に、**xrandr** ユーティリティーは、処理、解像度、回転、およびレイアウトへのアプローチが異なるため、Wayland では機能しません。

Jira:RHELPLAN-121049^[1]

11.14. WEB コンソール

RHEL Web コンソールの VNC コンソールが ARM64 で正しく動作しない

現在、ARM64 アーキテクチャー上の RHEL Web コンソールに仮想マシン (VM) をインポートし、VNC コンソールでその仮想マシンと対話しようとする、コンソールが入力に反応しません。

さらに、ARM64 アーキテクチャーの Web コンソールで仮想マシンを作成すると、VNC コンソールに入力の最後の行が表示されません。

Jira:RHEL-31993^[1]

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

firewalld.service がマスクされている場合、firewall RHEL システムロールの使用は失敗する

RHEL システム上で **firewalld.service** がマスクされている場合、**firewall** RHEL システムロールは失敗します。この問題を回避するには、**firewalld.service** のマスクを解除します。

```
systemctl unmask firewalld.service
```

Bugzilla:2123859

環境名でシステムを登録できない

rhc_environment に環境名を指定すると、**rhc** システムロールはシステムの登録に失敗します。回避策として、登録時に環境名の代わりに環境 ID を使用します。

Jira:RHEL-1172

高可用性モードの Microsoft SQL Server 2022 は、SELinux で制限されたアプリケーションとして実行できない

RHEL 9.4 以降の Microsoft SQL Server 2022 は、SELinux で制限されたアプリケーションとしての実行がサポートされています。ただし、Microsoft SQL Server の制限により、高可用性モードでは、同サービスを SELinux で制限されたアプリケーションとして実行することができません。この問題を回避するには、サービスを高可用性とする必要がある場合、Microsoft SQL Server を制限のないアプリケーションとして実行します。

mssql RHEL システムロールを使用して同サービスをインストールする場合、この制限は Microsoft SQL Server のインストールにも影響することに注意してください。

Jira:RHELDOCS-17719^[1]

mssql RHEL システムロールが、AD 統合を使用して Microsoft SQL Server を設定できない

Microsoft SQL Server サービスは、Active Directory (AD) との統合に必要な **adutil** ツールを提供しません。したがって、RHEL 9 管理対象ノードでこのシナリオを設定するために **mssql** RHEL システムロールを使用することはできません。回避策はありません。この RHEL システムロールは、RHEL 9 で AD 統合なしで Microsoft SQL Server を設定する場合にのみ使用できます。

Jira:RHELDOCS-17720^[1]

imuxsock 入力基本タイプを設定すると問題が発生する

logging RHEL システムロールを使用して **imuxsock** 入力基本タイプを設定し、**use_imuxsock** オプションを指定すると、その結果となるマネージドノードの設定で問題が発生します。ロールは **name** パラメーターを設定しますが、**imuxsock** 入力タイプは **name** パラメーターをサポートしません。その結果、**rsyslog** ログイングユーティリティーは、**parameter 'name' not known — typo in config file?** エラーを出力します。

Jira:RHELDOCS-18329^[1]

11.16. 仮想化

https または ssh 経由での仮想マシンのインストールに失敗する場合がある

現在、**virt-install** ユーティリティーは、https または ssh 接続を介して ISO ソースからゲストオペレーティングシステム (OS) をインストールしようとするとう失敗します。たとえば、**virt-install--cdromhttps://example/path/to/image.iso** を使用します。仮想マシンを作成する代わりに、上述の操作は **internal error: process exited while connecting to monitor** というメッセージが表示されて予期せず終了します。

同様に、https または ssh URL、あるいは **Download OS** 機能を使用した場合、RHEL 9 Web コンソールを使用したゲストオペレーティングシステムのインストールが失敗し、**Unknown driver 'https'** エラーが表示されます。

この問題を回避するには、ホストに **qemu-kvm-block-curl** および **qemu-kvm-block-ssh** をインストールして、https および ssh プロトコルのサポートを有効にします。別の接続プロトコルまたは別のインストールソースを使用することもできます。

Bugzilla:2014229

仮想マシンで NVIDIA ドライバーを使用すると Wayland が無効になる

現在、NVIDIA ドライバーは Wayland グラフィカルセッションと互換性がありません。これにより、NVIDIA ドライバーを使用する RHEL ゲストオペレーティングシステムは、Wayland を自動的に無効にし、代わりに Xorg セッションを読み込みます。これは主に以下のシナリオで生じます。

- NVIDIA GPU デバイスを RHEL 仮想マシンに渡す場合
- NVIDIA vGPU 仲介デバイスを RHEL 仮想マシンに割り当てる場合

Jira:RHELPLAN-117234^[1]

Milan 仮想マシンの CPU タイプは、AMD Milan システムで利用できないことがあります。

一部の AMD Milan システムでは、Enhanced REP MOVSB (**erms**) および Fast Short REP MOVSB (**fsrm**) 機能フラグがデフォルトで BIOS で無効になっています。したがって、**Milan** CPU タイプは、こ

これらのシステムで利用できない可能性があります。さらに、機能フラグ設定が異なる Milan ホスト間の仮想マシンのライブマイグレーションが失敗する可能性があります。これらの問題を回避するには、ホストの BIOS で **erns** および **fsrcm** を手動で有効にします。

Bugzilla:2077767^[1]

フェイルオーバー設定のある hostdev インターフェイスは、ホットアンプラグされた後にホットプラグすることはできない

フェイルオーバー設定の **hostdev** ネットワークインターフェイスを実行中の仮想マシン (VM) から削除した後、現在、インターフェイスを同じ実行中の VM に再接続することはできません。

Jira:RHEL-7337

フェイルオーバー VF を使用した VM のコピー後のライブマイグレーションが失敗する

現在、VM が仮想機能 (VF) フェイルオーバー機能が有効になっているデバイスを使用している場合、実行中の仮想マシン (VM) のコピー後移行の試行は失敗します。この問題を回避するには、コピー後の移行ではなく、標準の移行タイプを使用します。

Jira:RHEL-7335

ライブマイグレーション中にホストネットワークが VF と VM に ping できない

設定済みの仮想機能 (VF) で仮想マシン (仮想 SR-IOV ソフトウェアを使用する仮想マシンなど) のライブマイグレーションを行う場合、仮想マシンのネットワークは他のデバイスに表示されず、**ping** などのコマンドで仮想マシンに到達できません。ただし、移行が終了すると、問題は発生しなくなります。

Jira:RHEL-7336

AVX を無効にすると、仮想マシンが起動できなくなる

Advanced Vector Extensions (AVX) をサポートする CPU を使用するホストマシンで、現在、AVX を明示的に無効にして VM を起動しようとするとう失敗し、代わりに VM でカーネルパニックが発生します。

Bugzilla:2005173^[1]

ネットワークインターフェイスのリセット後に Windows VM が IP アドレスの取得に失敗する

ネットワークインターフェイスの自動リセット後に、Windows 仮想マシンが IP アドレスの取得に失敗することがあります。その結果、VM はネットワークに接続できません。この問題を回避するには、Windows デバイスマネージャーでネットワークアダプタードライバーを無効にしてから再度有効にします。

Jira:RHEL-11366

vCPU をホットプラグした後、Windows Server 2016 VM が動作を停止することがある

現在、Windows Server 2016 ゲストオペレーティングシステムで実行中の仮想マシン (VM) に vCPU を割り当てると、仮想マシンが予期せず停止したり、応答しなくなったり、再起動したりするなど、さまざまな問題が発生する可能性があります。

Bugzilla:1915715

NVIDIA パススルーデバイスを備えた VM での冗長エラーメッセージ。

RHEL 9.2 以降のオペレーティングシステムを搭載した Intel ホストマシンを使用している場合、パススルー NVIDIA GPU デバイスを備えた仮想マシン (VM) で、次のエラーメッセージが頻繁に記録されます。

Spurious APIC interrupt (vector 0xFF) on CPU#2, should never happen.

ただし、このエラーメッセージは VM の機能には影響しないため、無視してかまいません。詳細は、[Red Hat ナレッジベース](#) を参照してください。

Bugzilla:2149989^[1]

ホストで OVS サービスを再起動すると、実行中の VM でネットワーク接続がブロックされることがある

ホストで Open vSwitch (OVS) サービスが再起動またはクラッシュすると、このホストで実行されている仮想マシン (VM) はネットワークデバイスの状態を回復できません。その結果、仮想マシンがパケットを完全に受信できなくなる可能性があります。

この問題は、**virtio** ネットワークスタックで圧縮された virtqueue 形式を使用するシステムのみに影響します。

この問題を回避するには、**virtio** ネットワークデバイス定義で **packed=off** パラメーターを使用して、圧縮された virtqueue を無効にします。圧縮された virtqueue を無効にすると、状況によっては、ネットワークデバイスの状態を RAM から回復できます。

[Jira:RHEL-333](#)

中断されたコピー後の VM 移行の回復が失敗することがある

仮想マシン (VM) のコピー後の移行が中断された後、同じ受信ポートですぐに再開されると、移行は **Address already in use** のエラーで失敗する可能性があります。

この問題を回避するには、コピー後の移行を再開する前に少なくとも 10 秒待つか、移行の回復のために別のポートに切り替えます。

[Jira:RHEL-7096](#)

AMD EPYC CPU で NUMA ノードマッピングが正しく機能しない

QEMU は、AMD EPYC CPU の NUMA ノードマッピングを正しく処理しません。これにより、NUMA ノード設定を使用する場合、これらの CPU を持つ仮想マシン (VM) のパフォーマンスに悪影響が及ぶ可能性があります。さらに、VM は起動時に以下のような警告を表示します。

```
sched: CPU #4's llc-sibling CPU #3 is not on the same node! [node: 1 != 0]. Ignoring dependency.  
WARNING: CPU: 4 PID: 0 at arch/x86/kernel/smpboot.c:415 topology_sane.isra.0+0x6b/0x80
```

この問題を回避するには、NUMA ノード設定に AMD EPYC CPU を使用しないでください。

[Bugzilla:2176010](#)

VM 移行中の NFS 障害により、移行が失敗してソース仮想マシンのコアダンプが発生する

現在、仮想マシン (VM) の移行中に NFS サービスまたはサーバーがシャットダウンした場合、ソース VM の QEMU は、実行を再開したときに NFS サーバーに再接続できません。その結果、移行に失敗し、ソース VM でコアダンプが開始されます。現在、使用可能な回避策はありません。

[Bugzilla:2058982](#)

PCIe ATS デバイスが Windows 仮想マシンで動作しない

Windows ゲストオペレーティングシステムを使用して仮想マシン (VM) の XML 設定で PCIe アドレス変換サービス (ATS) デバイスを設定しても、ゲストが仮想マシンの起動後に ATS デバイスを有効にしません。これは、Windows が現在 **virtio** デバイス上の ATS をサポートしていないためです。

詳細は、[Red Hat ナレッジベース](#) を参照してください。

[Bugzilla:2073872](#)

virsh blkio tune --weight コマンドが正しい cgroup I/O コントローラー値を設定できない

現在、**virsh blkio tune --weight** コマンドを使用して VM weight を設定しても、期待どおりに機能しません。このコマンドは、cgroup I/O コントローラーインターフェイスファイルに正しい **io.bfq.weight** 値を設定できません。現時点では回避策はありません。

[Bugzilla:1970830](#)

NVIDIA A16 GPU を使用して仮想マシンを起動すると、ホスト GPU が動作を停止する場合があります

現在、NVIDIA A16 GPU パススルーデバイスを使用する仮想マシンを起動すると、ホストシステム上の NVIDIA A16 GPU 物理デバイスが動作を停止する場合があります。

この問題を回避するには、ハイパーバイザーを再起動し、GPU デバイスの **reset_method** を **bus** に設定します。

```
# echo bus > /sys/bus/pci/devices/<DEVICE-PCI-ADDRESS>/reset_method
# cat /sys/bus/pci/devices/<DEVICE-PCI-ADDRESS>/reset_method
bus
```

詳細は、[Red Hat ナレッジベースの記事](#) を参照してください。

[Jira:RHEL-7212^{\[1\]}](#)

ストレージエラーが原因で Windows 仮想マシンが応答しなくなる可能性がある

Windows ゲストオペレーティングシステムを使用する仮想マシン (VM) では、I/O 負荷が高いときにシステムが応答しなくなることがあります。このような場合、システムは **viostor Reset to device, \Device\RaidPort3, was issued** エラーをログに記録します。

[Jira:RHEL-1609^{\[1\]}](#)

特定の PCI デバイスを搭載した Windows 10 仮想マシンが起動時に応答しなくなることがある

現在、Windows 10 ゲストオペレーティングシステムを使用する仮想マシン (VM) は、ローカルディスクバックエンドを備えた **virtio-win-scsi** PCI デバイスが仮想マシンにアタッチされている場合、起動中に応答しなくなる可能性があります。この問題を回避するには、**multi_queue** オプションを有効にして仮想マシンを起動します。

[Jira:RHEL-1084^{\[1\]}](#)

virtio-win ドライバー用の virtio-win-guest-tool の修復機能が機能しない

現在、Virtio Balloon Driver などの **virtio-win** ドライバーに対して **virtio-win-guest-tool** の **Repair** ボタンを使用しても、このボタンは効果がありません。そのため、ドライバーをゲスト上で削除した後は再インストールできません。

[Jira:RHEL-1517^{\[1\]}](#)

メモリーバルーンデバイスセットが設定された Windows 11 仮想マシンが再起動中に予期せず終了することがある

現在、Windows 11 ゲストオペレーティングシステムとメモリーバルーンデバイスを使用する仮想マシン (VM) の再起動が、**DRIVER POWER STAT FAILURE** ブルースクリーンエラーで失敗する場合があります。

[Jira:RHEL-935^{\[1\]}](#)

コピー後の仮想マシン移行の再開が失敗することがある

現在、仮想マシンのコピー後の移行を実行するときに、移行の RECOVER フェーズでプロキシーネットワークに障害が発生すると、仮想マシンが応答しなくなり、移行を再開できなくなります。代わりに、`recovery` コマンドは次のエラーを表示します。

```
error: Requested operation is not valid: QEMU reports migration is still running
```

[Jira:RHEL-7115](#)

virtio バルーンドライバーが Windows 10 仮想マシンで動作しないことがある

特定の状況下では、`virtio-balloon` ドライバーが Windows 10 ゲストオペレーティングシステムを使用する仮想マシン (VM) 上で正しく動作しません。その結果、そのような仮想マシンは割り当てられたメモリーを効率的に使用できない可能性があります。

[Jira:RHEL-12118](#)

Windows 仮想マシンの virtio ファイルシステムのパフォーマンスは最適ではない

現在、Windows ゲストオペレーティングシステムを使用する仮想マシン (VM) 上で `virtio` ファイルシステム (`virtiofs`) が設定されている場合、仮想マシン内の `virtiofs` のパフォーマンスは、Linux ゲストを使用する仮想マシンよりも大幅に低下します。

[Jira:RHEL-1212^{\[1\]}](#)

Windows 仮想マシンのストレージデバイスのホットアンプラグが失敗する可能性がある

Windows ゲストオペレーティングシステムを使用する仮想マシン (VM) で、仮想マシンの実行中にストレージデバイスを削除すると (デバイスのホットアンプラグとも呼ばれる)、失敗する場合があります。その結果、ストレージデバイスは仮想マシンにアタッチされたままになり、ディスクマネージャーサービスが応答しなくなる可能性があります。

[Jira:RHEL-869](#)

CPU を Windows 仮想マシンにホットプラグするとシステム障害が発生する可能性がある

`Huge Page` が有効になっている Windows 仮想マシンに最大数の CPU をホットプラグすると、ゲストオペレーティングシステムが次の `Stop` エラーでクラッシュする場合があります。

```
PROCESSOR_START_TIMEOUT
```

[Jira:RHEL-1220](#)

Windows 仮想マシンの virtio ドライバーの更新が失敗する可能性がある

Windows 仮想マシン (VM) で KVM 準仮想化 (`virtio`) ドライバーを更新すると、更新によりマウスが動作しなくなり、新しくインストールされたドライバーが署名されない可能性があります。この問題は、`virtio-win.iso` ファイルの一部である `virtio-win-guest-tools` パッケージからインストールし

て、**virtio** ドライバーを更新する際に発生します。

この問題を回避するには、Windows Device Manager を使用して **virtio** ドライバーを更新します。

Jira:RHEL-574^[1]

vhost-kernel を使用する仮想マシンで TX キューのサイズを変更できない

現在、**virtio** ネットワークドライバーのバックエンドとして **vhost-kernel** を使用する KVM 仮想マシンでは、TX キューサイズをセットアップすることができません。したがって、TX キューにはデフォルト値の 256 しか使用できず、仮想マシンのネットワークスループットを最適化できない可能性があります。

Jira:RHEL-1138^[1]

仮想マシンが AMD SRSO の脆弱性を誤って報告する

AMD Zen 3 および 4 の CPU アーキテクチャーを搭載した RHEL 9 ホスト上で実行している RHEL 9.4 仮想マシン (VM) は、投機的リターンスタックオーバーフロー (SRSO) 攻撃に対する脆弱性を誤って報告します。

```
# lscpu | grep rstack
```

```
Vulnerability Spec rstack overflow: Vulnerable: Safe RET, no microcode
```

この問題は **cpuid** フラグの欠落によって発生し、次の条件下では仮想マシンで脆弱性は実際に完全に軽減されます。

- ホストには、ここで説明されているように、更新された **linux-firmware** パッケージがあります ([cve-2023-20569](#))。
- ホストカーネルでは緩和策が有効になっており、これがデフォルトの動作です。緩和策が有効になっている場合は、ホスト上の **lscpu** コマンド出力に **Safe RET** が表示されます。

Jira:RHEL-26152^[1]

大量の vCPU と仮想ディスクを持つ仮想マシンは失敗する可能性がある

現在、RHEL 仮想マシン (VM) に大量の vCPU と仮想ディスクを割り当てると、仮想マシンの起動に失敗する場合があります。

この問題を回避するには、可能であれば、ブロックデバイスの代わりに Small Computer System Interface (SCSI) 仮想ストレージデバイスを使用します。詳細は、[CLI を使用して vHBA デバイスで SCSI ベースのストレージプールの作成](#) を参照してください。

仮想ブロックデバイスを使用する必要がある場合は、**-global virtio-blk-pci.vectors=<number-of-vectors>** QEMU オプションを使用して仮想マシンを起動し、割り込みベクターの数を減らすこともできます。仮想マシンが正常に起動できる、十分に少ない数の割り込みベクターを見つけるようにしてください。

Jira:RHEL-32990^[1]

e1000e または **igb** モデルインターフェイスのステータスが **down** の場合でも、リンクステータスは仮想マシン上に **up** と表示されます。

仮想マシンを起動する前に、**e1000** または **igb** モデルのネットワークインターフェイスのイーサネットリンクのステータスを **down** に設定します。設定したにもかかわらず、仮想マシンの起動後、ネット

ワークインターフェイスは **up** ステータスを維持します。これは、イーサネットリンクのステータスを **down** に設定し、仮想マシンを停止して再起動すると、自動的に **up** に設定されるためです。その結果、ネットワークインターフェイスの正しい状態が維持されません。回避策として、次のコマンドを使用して、仮想マシン内のネットワークインターフェイスのステータスを **down** に設定します。

```
# ip link set dev eth0 down
```

または、仮想マシンの実行中にこのネットワークインターフェイスを削除して再度追加してみることもできます。

[Jira:RHEL-21867](#)

NBD を使用して TLS 接続経由で仮想マシンストレージを移行すると正しく動作しない

現在、TLS 接続を介して Network Block Device (NBD) プロトコルを使用して仮想マシン (VM) とそのストレージデバイスを移行する場合は、TLS ハンドシェイクでのデータ競合により、移行が成功したように見えることがあります。ただし、これにより、宛先仮想マシン上の QEMU プロセスがそれ以上のやり取りに応答しなくなります。

ネットワークを信頼できる場合は、仮想マシンストレージの移行中に使用される NBD プロトコルに TLS 接続ではなくプレーンテキストを使用することで、この問題を回避できます。

[Jira:RHEL-33440](#)

AMD SEV-SNP を搭載した仮想マシンで Kdump が失敗する

現在、Secure Nested Paging (SNP) 機能を備えた AMD Secure Encrypted Virtualization (SEV) を使用する RHEL 9 仮想マシン (VM) では kdump が失敗します。

[Jira:RHEL-10019^{\[1\]}](#)

11.17. クラウド環境の RHEL

Nutanix AHV で LVM を使用する RHEL 9 仮想マシンのクローンを作成または復元すると、ルート以外のパーティションが表示されなくなる

Nutanix AHV ハイパーバイザーをホストとする仮想マシン (VM) で RHEL 9 ゲストオペレーティングシステムを実行する場合、スナップショットから VM を復元するか VM をクローンすると、ゲストが論理ボリューム管理 (LVM) を使用している場合は VM 内の非ルートパーティションを消失させることがあります。これにより、以下の問題が発生します。

- スナップショットから仮想マシンを復元すると、仮想マシンは起動できず、緊急モードに入ります。
- クローンを作成して作成した仮想マシンは起動できず、緊急モードに入ります。

これらの問題を回避するには、仮想マシンの緊急モードで以下を行います。

1. 以下の LVM システムデバイスファイルを削除します: **rm/etc/lvm/devices/system.devices**
2. LVM デバイス設定を再作成します: **vgimportdevices -a**
3. 仮想マシンを再起動します。

これにより、クローン化または復元された VM を正しく起動できます。

または、問題が発生しないようにするには、VM のクローンを作成する前、または VM のスナップショットを作成する前に、次の手順を実行します。

1. `/etc/lvm/lvm.conf` ファイルの `use_devicesfile = 0` 行のコメントを外します
2. 仮想マシンを再起動します。

Bugzilla:2059545^[1]

ESXi で RHEL 9 ゲストをカスタマイズすると、ネットワークの問題が発生することがある

現在、VMware ESXi ハイパーバイザーでの RHEL 9 ゲストオペレーティングシステムのカスタマイズは、NetworkManager キーファイルでは正しく機能しません。その結果、ゲストがそのようなキーファイルを使用している場合、IP アドレスやゲートウェイなどのネットワーク設定が正しくなくなります。

詳細と回避策は、[VMware ナレッジベース](#) を参照してください。

Bugzilla:2037657^[1]

cloud-init によってプロビジョニングされ、NFSv3 マウントエントリーで設定された場合、Azure で RHEL インスタンスが起動しない

現在、仮想マシンが `cloud-init` ツールによってプロビジョニングされ、仮想マシンのゲストオペレーティングシステムで `/etc/fstab` ファイルに NFSv3 マウントエントリーがある場合、Microsoft Azure クラウドプラットフォームで RHEL 仮想マシンの起動に失敗します。

Bugzilla:2081114^[1]

VMware ホストの RHEL 仮想マシンで静的 IP を設定できない

現在、VMware ホストで RHEL 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。

この問題を回避するには、[VMware のナレッジベース](#) を参照してください。

Jira:RHEL-12122

kmemleak オプションが有効になっていると、大規模な仮想マシンがデバッグカーネルで起動できない場合がある

RHEL 9 仮想マシンをデバッグカーネルで起動しようとする時、マシンカーネルが `kmemleak=on` 引数を使用している場合、次のエラーで起動が失敗することがあります。

```
Cannot open access to console, the root account is locked.
See slogin(8) man page for more details.
```

```
Press Enter to continue.
```

この問題は主に、ブートシーケンスに多くの時間を費やす大規模な仮想マシンに影響します。

この問題を回避するには、マシン上の `/etc/fstab` ファイルを編集し、追加のタイムアウトオプションを `/boot` および `/boot/efi` マウントポイントに追加します。以下に例を示します。

```
UUID=e43ead51-b364-419e-92fc-b1f363f19e49 /boot xfs defaults,x-systemd.device-timeout=600,x-systemd.mount-timeout=600 0 0
```

```
UUID=7B77-95E7 /boot/efi vfat defaults,uid=0,gid=0,umask=077,shortname=winnt,x-systemd.device-timeout=600,x-systemd.mount-timeout=600 0 2
```

Jira:RHELDOCS-16979^[1]

11.18. サポート性

IBM Power Systems (Little Endian) で `sos report` を実行するとタイムアウトする

数百または数千の CPU を搭載した IBM Power Systems (Little Endian) で `sos report` コマンドを実行すると、`/sys/devices/system/cpu` ディレクトリーの膨大なコンテンツを収集する際のプロセッサプラグインはデフォルトのタイムアウトである 300 秒に達します。回避策として、それに応じてプラグインのタイムアウトを増やします。

- 1 回限りの設定の場合は、次を実行します。

```
# sos report -k processor.timeout=1800
```

- 永続的な変更を行うには、`/etc/sos/sos.conf` ファイルの `[plugin_options]` セクションを編集します。

```
[plugin_options]
# Specify any plugin options and their values here. These options take the form
# plugin_name.option_name = value
#rpm.rpmva = off
processor.timeout = 1800
```

値の例は 1800 に設定されています。特定のタイムアウト値は、特定のシステムに大きく依存します。プラグインのタイムアウトを適切に設定するには、次のコマンドを実行して、タイムアウトなしで 1 つのプラグインを収集するために必要な時間を最初に見積もることができます。

```
# time sos report -o processor -k processor.timeout=0 --batch --build
```

Bugzilla:1869561^[1]

11.19. コンテナ

古いコンテナイメージ内で `systemd` を実行すると動作しない

古いコンテナイメージ (例:`centos:7`) で `systemd` を実行しても動作しません。

```
$ podman run --rm -ti centos:7 /usr/lib/systemd/systemd
Storing signatures
Failed to mount cgroup at /sys/fs/cgroup/systemd: Operation not permitted
[!!!!!!] Failed to mount API filesystems, freezing.
```

この問題を回避するには、以下のコマンドを使用します。

```
# mkdir /sys/fs/cgroup/systemd
# mount none -t cgroup -o none,name=systemd /sys/fs/cgroup/systemd
# podman run --runtime /usr/bin/crun --annotation=run.oci.systemd.force_cgroup_v1=/sys/fs/cgroup -rm -ti centos:7 /usr/lib/systemd/systemd
```

Jira:RHELPLAN-96940^[1]

ルートファイルシステムはデフォルトでは拡張されない

cloud-init を含まないベースコンテナイメージを使用して、**bootc-image-builder** を使用して AMI または QCOW2 コンテナイメージを作成すると、起動時にルートファイルシステムのサイズがプロビジョニングされた仮想ディスクのフルサイズまで動的に拡張されません。

この問題を回避するには、次の利用可能なオプションのいずれかを適用します。

- イメージに **cloud-init** を含めます。
- コンテナイメージにカスタムロジックを含めて、ルートファイルシステムを拡張します。次に例を示します。

```
/usr/bin/growpart /dev/vda 4  
unshare -m bin/sh -c 'mount -o remount,rw /sysroot && xfs_growfs /sysroot'
```

- **/var/lib/containers** などのセカンダリーファイルシステムに追加のスペースを使用するためのカスタムロジックを含めます。



注記

デフォルトでは、物理ルートストレージは **/sysroot** パーティションにマウントされません。

Jira:RHEL-33208

付録A コンポーネント別のチケットリスト

参考のために、Bugzilla および JIRA チケットのリストをこのドキュメントに記載します。リンクをクリックすると、チケットについて説明したこのドキュメントのリリースノートにアクセスできます。

コンポーネント	チケット
389-ds-base	Jira:RHEL-15907 , Jira:RHEL-5142 , Jira:RHEL-5133 , Jira:RHEL-5130 , Jira:RHEL-16830 , Jira:RHEL-17175 , Jira:RHEL-5111
NetworkManager	Jira:RHEL-1441 , Jira:RHEL-1471 , Jira:RHEL-16470 , Jira:RHEL-1469 , Jira:RHEL-24337 , Jira:RHEL-5852 , Bugzilla:1894877 , Jira:RHEL-17619
リリースノート	Jira:RHELDOCS-17841 , Jira:RHELDOCS-16861 , Jira:RHELDOCS-16760 , Jira:RHELDOCS-17520 , Jira:RHELDOCS-17803 , Jira:RHELDOCS-16756 , Jira:RHELDOCS-16612 , Jira:RHELDOCS-17102 , Jira:RHELDOCS-17166 , Jira:RHELDOCS-17309 , Jira:RHELDOCS-17545 , Jira:RHELDOCS-17518 , Jira:RHELDOCS-17989 , Jira:RHELDOCS-17702 , Jira:RHELDOCS-17917 , Jira:RHELDOCS-16979
anaconda	Jira:RHEL-11384 , Jira:RHEL-13150 , Jira:RHEL-4766 , Jira:RHEL-5638 , Jira:RHEL-10216 , Jira:RHEL-2250 , Jira:RHEL-17205 , Bugzilla:2127473 , Bugzilla:2050140 , Bugzilla:1877697 , Jira:RHEL-4707 , Jira:RHEL-4711 , Bugzilla:1997832 , Jira:RHEL-4741 , Bugzilla:2115783 , Jira:RHEL-4762 , Jira:RHEL-4737 , Jira:RHEL-9633
ansible-collection-microsoft-sql	Jira:RHEL-16342 , Jira:RHEL-19092 , Jira:RHEL-19091 , Jira:RHEL-3540
ansible-freeipa	Jira:RHEL-4962 , Jira:RHEL-16934 , Jira:RHEL-16939 , Jira:RHEL-19134 , Jira:RHEL-19130
audit	Jira:RHEL-14896
bacula	Jira:RHEL-6856
bcc	Jira:RHEL-16325
bind	Bugzilla:1984982
boom-boot	Jira:RHEL-16813
bootc-image-builder-container	Jira:RHEL-34054
certmonger	Jira:RHEL-22302
chrony	Jira:RHEL-6522

コンポーネント	チケット
clang	Jira:RHEL-9346
cloud-init	Jira:RHEL-7278 、 Jira:RHEL-7311 、 Jira:RHEL-12122
cmake	Jira:RHEL-7393
cockpit-appstream	Bugzilla:2030836
cockpit-machines	Jira:RHEL-17434 、 Bugzilla:2173584 、 Jira:RHEL-31993
crash	Jira:RHEL-9009
createrepo_c	Bugzilla:2056318
crypto-policies	Jira:RHEL-15925 、 Jira:RHEL-2735
cyrus-sasl	Bugzilla:1995600
device-mapper-multipath	Jira:RHEL-6678 、 Jira:RHEL-1729 、 Jira:RHEL-4998 、 Jira:RHEL-986 、 Jira:RHEL-1830 、 Jira:RHEL-17234 、 Bugzilla:2033080 、 Bugzilla:2011699 、 Bugzilla:1926147
distribution	Jira:RHEL-17089 、 Jira:RHEL-6973 、 Jira:RHEL-18157 、 Jira:RHEL-22385
dnf	Bugzilla:2073510
dnf-plugins-core	Jira:RHEL-4600
edk2	Bugzilla:1935497
elfutils	Jira:RHEL-12489
fapolicyd	Bugzilla:2054740 、 Jira:RHEL-24345 、 Jira:RHEL-520
firewalld	Jira:RHEL-427 、 Jira:RHEL-14485 、 Jira:RHEL-17708
gcc	Jira:RHEL-17638
gcc-toolset-13-binutils	Jira:RHEL-23798
gcc-toolset-13-gcc	Jira:RHEL-16998
gimp	Bugzilla:2047161

コンポーネント	チケット
git	Jira:RHEL-17100
git-lfs	Jira:RHEL-17101
glibc	Jira:RHEL-14383 , Jira:RHEL-17157 , Jira:RHEL-2491 , Jira:RHEL-19862 , Jira:RHEL-16643 , Jira:RHEL-12362 , Jira:RHEL-3397 , Jira:RHEL-2123
gnupg2	Bugzilla:2070722
gnutls	Jira:RHEL-14891 , Bugzilla:2108532
golang	Jira:RHEL-11871 , Bugzilla:2111072 , Bugzilla:2092016
grafana	Jira:RHEL-7505
grub2	Jira:RHEL-10288
gtk3	Jira:RHEL-11924
httpd	Jira:RHEL-6600
ipa	Jira:RHEL-11652 , Jira:RHEL-23377 , Bugzilla:1513934 , Jira:RHEL-9984 , Jira:RHEL-22313 , Jira:RHEL-12143 , Bugzilla:2084180 , Bugzilla:2094673 , Bugzilla:2057471 , Jira:RHEL-12154 , Jira:RHEL-4955
iptables	Jira:RHEL-14147
jmc-core	Bugzilla:1980981
kdump-anaconda-addon	Jira:RHEL-11196
kernel	Jira:RHEL-11597 , Bugzilla:2041883 , Bugzilla:1613522 , Bugzilla:1995338 , Bugzilla:1570255 , Bugzilla:2177256 , Bugzilla:2178699 , Bugzilla:2023416 , Bugzilla:2021672 , Bugzilla:2027304 , Bugzilla:1955275 , Bugzilla:2142102 , Bugzilla:2040643 , Bugzilla:2186375 , Bugzilla:2183538 , Bugzilla:2206599 , Bugzilla:2167783 , Bugzilla:2000616 , Bugzilla:2013650 , Bugzilla:2132480 , Bugzilla:2059545 , Bugzilla:2005173 , Bugzilla:2128610 , Bugzilla:2129288 , Bugzilla:2013884 , Bugzilla:2149989
kernel / BPF	Jira:RHEL-10691
kernel / Crypto	Jira:RHEL-20145
kernel / DMA Engine	Jira:RHEL-10097

コンポーネント	チケット
kernel / Debugging-Tracing / rtda	Jira:RHEL-10079
kernel / Kernel-Core	Jira:RHEL-25967
kernel / Networking / IPSec	Jira:RHEL-1015
kernel / Networking / NIC Drivers	Jira:RHEL-9308 、 Jira:RHEL-24618 、 Jira:RHEL-9897
kernel / Networking / Netfilter	Jira:RHEL-16630
kernel / Networking / Protocol / tcp	Jira:RHEL-21223 、 Jira:RHEL-5736
kernel / Platform Enablement / NVMe	Jira:RHEL-21545 、 Jira:RHEL-14751 、 Jira:RHEL-8171 、 Jira:RHEL-8164
kernel / Platform Enablement / ppc64	Jira:RHEL-15404
kernel / Security / TPM	Jira:RHEL-18985
kernel / Storage / Device Mapper / Crypt	Jira:RHEL-23572
kernel / Storage / Multiple Devices (MD)	Jira:RHEL-30730
kernel / Storage / Storage Drivers	Jira:RHEL-8466 、 Jira:RHEL-8104 、 Jira:RHEL-25730
kernel / Virtualization	Jira:RHEL-1138
kernel / Virtualization / KVM	Jira:RHEL-2815 、 Jira:RHEL-13007 、 Jira:RHEL-7212 、 Jira:RHEL-26152 、 Jira:RHEL-10019
kernel-rt	Bugzilla:2181571
kernel-rt / Other	Jira:RHEL-9318
kexec-tools	Bugzilla:2113873 、 Bugzilla:2064708
keylime	Jira:RHEL-11867 、 Jira:RHEL-1518

コンポーネント	チケット
kmod	Bugzilla:2103605
kmod-kvdo	Jira:RHEL-8354
krb5	Jira:RHEL-4902 , Bugzilla:2060798 , Jira:RHEL-4875 , Jira:RHEL-4889 , Bugzilla:2016312 , Jira:RHEL-4888
libabigail	Jira:RHEL-16629
libdnf	Jira:RHEL-11238
libkcapi	Jira:RHEL-15298 , Jira:RHEL-5367
libotr	Bugzilla:2086562
librepo	Jira:RHEL-11240
libreswan	Jira:RHEL-12278
librhsm	Jira:RHEL-14224
libsepol	Jira:RHEL-16233
libvirt	Bugzilla:2143158 , Bugzilla:2078693
libvirt / General	Jira:RHEL-7568 , Jira:RHEL-7043
libvirt / Storage	Jira:RHEL-7528 , Jira:RHEL-7416
libxcrypt	Bugzilla:2034569
libzip	Jira:RHEL-17567
linuxptp	Jira:RHEL-2026
llvm-toolset	Jira:RHEL-9283
lvm2	Jira:RHEL-8357 , Bugzilla:2038183
make	Jira:RHEL-22829
mariadb	Jira:RHEL-3638
maven	Jira:RHEL-13046

コンポーネント	チケット
mysql	Bugzilla:1991500
nettle	Jira:RHEL-14890
nfs-utils	Bugzilla:2081114
nftables	Jira:RHEL-5980 , Jira:RHEL-14191
nginx	Jira:RHEL-14713
nmstate	Jira:RHEL-1434 , Jira:RHEL-1605 , Jira:RHEL-1438 , Jira:RHEL-19142 , Jira:RHEL-1420 , Jira:RHEL-1425
nvme-cli	Jira:RHEL-1492
nvme-stas	Bugzilla:1893841
open-vm-tools	Bugzilla:2037657
opencryptoki	Jira:RHEL-11412
opensc	Jira:RHEL-4079
openscap	Bugzilla:2161499
openslp	Jira:RHEL-6995
openssh	Jira:RHEL-5222 , Jira:RHEL-2469 , Bugzilla:2056884
openssl	Jira:RHEL-23474 , Jira:RHEL-17193 , Bugzilla:2168665 , Bugzilla:1975836 , Bugzilla:1681178 , Bugzilla:1685470
osbuild	Jira:RHEL-4655
osbuild-composer	Bugzilla:2173928 , Jira:RHEL-7999 , Jira:RHEL-4649
oscap-anaconda-addon	Jira:RHEL-1824 , Jira:RHELPLAN-44202
p11-kit	Jira:RHEL-14834
papi	Jira:RHEL-9333
pause-container	Bugzilla:2106816
pcp	Jira:RHEL-2317

コンポーネント	チケット
pcs	Jira:RHEL-7672 , Jira:RHEL-7582 , Jira:RHEL-7724 , Jira:RHEL-7746 , Jira:RHEL-7744 , Jira:RHEL-7669 , Jira:RHEL-7730
php	Jira:RHEL-14699
pki-core	Bugzilla:2084181
podman	Jira:RHELPLAN-167829 , Jira:RHELPLAN-167796 , Jira:RHELPLAN-167823 , Jira:RHELPLAN-168180 , Jira:RHELPLAN-168185 , Jira:RHELPLAN-154436 , Bugzilla:2069279
polycoreutils	Jira:RHEL-24462 , Jira:RHEL-25263
postgresql	Jira:RHEL-3635
procps-ng	Jira:RHEL-16278
python3.11-lxml	Bugzilla:2157708
qemu-kvm	Jira:RHEL-11597 , Jira:RHEL-16695 , Bugzilla:1965079 , Bugzilla:1951814 , Bugzilla:2060839 , Bugzilla:2014229 , Jira:RHEL-7335 , Jira:RHEL-7336 , Bugzilla:1915715 , Jira:RHEL-13335 , Jira:RHEL-333 , Bugzilla:2176010 , Bugzilla:2058982 , Bugzilla:2073872 , Jira:RHEL-7478 , Jira:RHEL-32990
qemu-kvm / Devices	Jira:RHEL-1220
qemu-kvm / Graphics	Jira:RHEL-7135
qemu-kvm / Live Migration	Jira:RHEL-13004 , Jira:RHEL-7096 , Jira:RHEL-7115
qemu-kvm / Networking	Jira:RHEL-7337 , Jira:RHEL-21867
qemu-kvm / Storage / NBD	Jira:RHEL-33440
realtime-tests	Jira:RHEL-9910
rear	Jira:RHEL-16864 , Jira:RHEL-10478 , Jira:RHEL-6984 , Jira:RHEL-17393 , Jira:RHEL-24847
restore	Bugzilla:1997366
rhel-bootc-container	Jira:RHEL-33208

コンポーネント	チケット
rhel-system-roles	Jira:RHEL-1535、 Jira:RHEL-16976、 Jira:RHEL-16542、 Jira:RHEL-16552、 Jira:RHEL-19579、 Jira:RHEL-17668、 Jira:RHEL-21133、 Jira:RHEL-16964、 Jira:RHEL-16541、 Jira:RHEL-15932、 Jira:RHEL-15439、 Jira:RHEL-18962、 Jira:RHEL-16212、 Jira:RHEL-15876、 Jira:RHEL-21117、 Jira:RHEL-16974、 Jira:RHEL-5972、 Jira:RHEL-15037、 Jira:RHEL-19046、 Jira:RHEL-18026、 Jira:RHEL-1683、 Jira:RHEL-3353、 Jira:RHEL-19043、 Jira:RHEL-19040、 Jira:RHEL-17875、 Jira:RHEL-5274、 Jira:RHEL-15870、 Jira:RHEL-15909、 Jira:RHEL-22228、 Jira:RHEL-25508、 Jira:RHEL-21401、 Jira:RHEL-22309、 Bugzilla:1999770、 Bugzilla:2123859、 Jira:RHEL-1172、 Bugzilla:2186218
rsyslog	Jira:RHEL-943、 Jira:RHEL-937、 Jira:RHEL-5196
rteval	Jira:RHEL-9912
rust	Jira:RHEL-12963
s390utils	Bugzilla:1932480
samba	Jira:RHEL-16476
scap-security-guide	Jira:RHEL-21425、 Jira:RHEL-1800、 Bugzilla:2038978
selinux-policy	Jira:RHEL-12591、 Jira:RHEL-21452、 Jira:RHEL-1548、 Jira:RHEL-18219、 Jira:RHEL-14246、 Jira:RHEL-1551、 Jira:RHEL-11174、 Jira:RHEL-10087、 Jira:RHEL-1553、 Jira:RHEL-14289、 Jira:RHEL-5032、 Jira:RHEL-15432、 Jira:RHEL-19051、 Jira:RHEL-11792、 Bugzilla:2064274、 Jira:RHEL-28814
sos	Bugzilla:1869561
sssd	Jira:SSSD-7015、 Bugzilla:1608496
sssd_kcm	Jira:SSSD-7015
stratis-cli	Jira:RHEL-2265
stratisd	Jira:RHEL-12898、 Jira:RHEL-16736
stunnel	Jira:RHEL-2468
subscription-manager	Bugzilla:2163716、 Bugzilla:2136694
synce4l	Jira:RHEL-10089

コンポーネント	チケット
sysstat	Jira:RHEL-12009 , Jira:RHEL-26275
systemd	Bugzilla:2018112 、 Jira:RHEL-6105
systemtap	Jira:RHEL-12488
tigervnc	Bugzilla:2060308
tuna	Jira:RHEL-8859
tuned	Bugzilla:2113900
udisks2	Bugzilla:2213769
unbound	Bugzilla:2070495
vdo	Jira:RHEL-30525
virt-v2v	Bugzilla:2168082
virtio-win	Jira:RHEL-11810 、 Jira:RHEL-11366 、 Jira:RHEL-1609 、 Jira:RHEL-869
virtio-win / distribution	Jira:RHEL-1517 , Jira:RHEL-1860 , Jira:RHEL-574
virtio-win / virtio-win-prewhql	JIRA:RHEL-1084 、 JIRA:RHEL-935 、 JIRA:RHEL-12118 、 JIRA:RHEL-1212
webkit2gtk3	Jira:RHEL-4157
xdp-tools	Jira:RHEL-3382

コンポーネント	チケット
その他	Jira:RHELDOCS-17369、 Jira:RHELDOCS-17263、 Jira:RHELDOCS-17060、 Jira:RHELDOCS-17056、 Jira:RHELDOCS-16721、 Jira:RHELDOCS-17372、 Jira:RHELDOCS-16970、 Jira:RHELPLAN-169666、 Jira:RHELDOCS-17000、 Jira:RHELDOCS-16241、 Jira:RHELDOCS-16955、 Jira:RHELDOCS-17053、 Jira:RHELDOCS-17792、 Jira:SSSD-6184、 Jira:RHELDOCS-17040、 Bugzilla:2020529、 Jira:RHELPLAN-27394、 Jira:RHELPLAN-27737、 Jira:RHELDOCS-16861、 Jira:RHELDOCS-17050、 Jira:RHELDOCS-17520、 Jira:RHELDOCS-17752、 Jira:RHELDOCS-17803、 Jira:RHELDOCS-17468、 Jira:RHELDOCS-17733、 Bugzilla:1927780、 Jira:RHELPLAN-110763、 Bugzilla:1935544、 Bugzilla:2089200、 Jira:RHELPLAN-99136、 Jira:RHELPLAN-103232、 Bugzilla:1899167、 Bugzilla:1979521、 Jira:RHELPLAN-100087、 Jira:RHELPLAN-100639、 Bugzilla:2058153、 Jira:RHELPLAN-113995、 Jira:RHELPLAN-98983、 Jira:RHELPLAN-131882、 Jira:RHELPLAN-139805、 Jira:RHELDOCS-16756、 Jira:RHELPLAN-153267、 Jira:RHELDOCS-16300、 Jira:RHELDOCS-16432、 Jira:RHELDOCS-16393、 Jira:RHELDOCS-16612、 Jira:RHELDOCS-17102、 Jira:RHELDOCS-17015、 Jira:RHELDOCS-18049、 Jira:RHELDOCS-17135、 Jira:RHELDOCS-17545、 Jira:RHELDOCS-17038、 Jira:RHELDOCS-17495、 Jira:RHELDOCS-17518、 Jira:RHELDOCS-17462、 Jira:RHELDOCS-18106、 Jira:RHELPLAN-157225、 Bugzilla:1640697、 Bugzilla:1697896、 Bugzilla:2047713、 Jira:RHELPLAN-96940、 Jira:RHELPLAN-117234、 Jira:RHELPLAN-119001、 Jira:RHELPLAN-119852、 Bugzilla:2077767、 Bugzilla:2053598、 Bugzilla:2082303、 Jira:RHELPLAN-121049、 Jira:RHELPLAN-157939、 Jira:RHELPLAN-109613、 Bugzilla:2160619、 Jira:RHELDOCS-18064、 Jira:RHELDOCS-16427、 Bugzilla:2173992、 Bugzilla:2185048、 Bugzilla:1970830、 Jira:RHELDOCS-16574、 Jira:RHELDOCS-17719、 Jira:RHELDOCS-17720

付録B 改訂履歴

0.0-7

Wed Jun 12 2024、 Brian Angelica (bangelic@redhat.com)

- [RHELPLAN-169666](#) (Identity Management)の機能拡張を更新しました。

0.0-6

2024年5月29日(水) Gabriela Fialová (gfialova@redhat.com)

- 機能拡張 [RHEL-12490](#) (コンパイラおよび開発ツール) を追加
- 機能拡張 [RHEL-12491](#) (コンパイラおよび開発ツール) を追加
- 機能拡張 [RHEL-13760](#) (RHEL システムロール) を更新
- インプレースアップグレードセクションの更新

0.0-5

2024年5月28日(火) Lenka Špačková (lspackova@redhat.com)

- [RHEL-16629](#) (コンパイラおよび開発ツール) のフォーマットを修正

0.0-4

2024年5月23日(木)、 Gabriela Fialová (gfialova@redhat.com)

- 機能拡張 [RHEL-23798](#) (コンパイラおよび開発ツール) を更新

0.0-3

2024年5月21日(火) Lenka Špačková (lspackova@redhat.com)

- 機能拡張 [RHEL-35685](#) (動的プログラミング言語、Web およびデータベースサーバー) を追加

0.0-2

2024年5月16日(木)、 Gabriela Fialová (gfialova@redhat.com)

- 機能拡張 [RHEL-16336](#) (RHEL システムロール) を追加
- 機能拡張 [RHEL-13760](#) (RHEL システムロール) を追加

0.0-1

2024年5月1日(水) Gabriela Fialová (gfialova@redhat.com)

- Red Hat Enterprise Linux 9.4 リリースノートのリリース。

0.0-0

2024年3月27日水曜日、 Gabriela Fialová (gfialova@redhat.com)

- Red Hat Enterprise Linux 9.4 ベータ版リリースノートのリリース。

