



# Red Hat Enterprise Linux 9

## 9.3 リリースノート

Red Hat Enterprise Linux 9.3 のリリースノート



# Red Hat Enterprise Linux 9 9.3 リリースノート

---

Red Hat Enterprise Linux 9.3 のリリースノート

## 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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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 9.3 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
<b>第1章 概要</b> .....	<b>7</b>
1.1. RHEL 9.3 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
<b>第2章 アーキテクチャー</b> .....	<b>12</b>
<b>第3章 RHEL 9 のコンテンツの配布</b> .....	<b>13</b>
3.1. INSTALLATION .....	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
<b>第4章 新機能</b> .....	<b>15</b>
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 .....	16
4.3. RHEL FOR EDGE .....	23
4.4. ソフトウェア管理 .....	24
4.5. シェルおよびコマンドラインツール .....	25
4.6. インフラストラクチャーサービス .....	26
4.7. ネットワーク .....	26
4.8. カーネル .....	34
4.9. ブートローダー .....	38
4.10. ファイルシステムおよびストレージ .....	38
4.11. 高可用性およびクラスター .....	41
4.12. 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 .....	43
4.13. コンパイラーおよび開発ツール .....	47
4.14. IDENTITY MANAGEMENT .....	66
4.15. グラフィックインフラストラクチャー .....	72
4.16. WEB コンソール .....	72
4.17. RED HAT ENTERPRISE LINUX システムロール .....	72
4.18. 仮想化 .....	76
4.19. クラウド環境の RHEL .....	77
4.20. サポート性 .....	78
4.21. コンテナ .....	79
<b>第5章 外部カーネルパラメーターへの重要な変更</b> .....	<b>82</b>
新しいカーネルパラメーター .....	82
更新されたカーネルパラメーター .....	83
削除されたカーネルパラメーター .....	88
<b>第6章 デバイスドライバー</b> .....	<b>89</b>
6.1. 新しいドライバー .....	89
6.2. 更新されたドライバー .....	94
<b>第7章 利用可能な BPF 機能</b> .....	<b>96</b>
<b>第8章 バグ修正</b> .....	<b>115</b>
8.1. インストーラーおよびイメージの作成 .....	115

8.2. セキュリティー	115
8.3. サブスクリプションの管理	121
8.4. ソフトウェア管理	121
8.5. シェルおよびコマンドラインツール	122
8.6. ネットワーク	123
8.7. カーネル	124
8.8. ブートローダー	124
8.9. ファイルシステムおよびストレージ	124
8.10. 高可用性およびクラスター	125
8.11. コンパイラーおよび開発ツール	127
8.12. IDENTITY MANAGEMENT	128
8.13. WEB コンソール	130
8.14. RED HAT ENTERPRISE LINUX システムロール	131
8.15. 仮想化	135
<b>第9章 テクノロジープレビュー</b> .....	<b>137</b>
9.1. インストーラーおよびイメージの作成	137
9.2. セキュリティー	137
9.3. シェルおよびコマンドラインツール	137
9.4. インフラストラクチャーサービス	137
9.5. ネットワーク	138
9.6. カーネル	139
9.7. ファイルシステムおよびストレージ	141
9.8. コンパイラーおよび開発ツール	143
9.9. IDENTITY MANAGEMENT	143
9.10. デスクトップ	145
9.11. 仮想化	146
9.12. クラウド環境の RHEL	148
9.13. コンテナ	148
<b>第10章 非推奨になった機能</b> .....	<b>149</b>
10.1. インストーラーおよびイメージの作成	149
10.2. セキュリティー	150
10.3. サブスクリプションの管理	152
10.4. シェルおよびコマンドラインツール	152
10.5. ネットワーク	153
10.6. カーネル	153
10.7. ファイルシステムおよびストレージ	154
10.8. 動的プログラミング言語、WEB サーバー、およびデータベースサーバー	155
10.9. コンパイラーおよび開発ツール	156
10.10. IDENTITY MANAGEMENT	156
10.11. デスクトップ	158
10.12. グラフィックインフラストラクチャー	158
10.13. RED HAT ENTERPRISE LINUX システムロール	159
10.14. 仮想化	159
10.15. コンテナ	161
10.16. 非推奨のパッケージ	162
<b>第11章 既知の問題</b> .....	<b>176</b>
11.1. インストーラーおよびイメージの作成	176
11.2. セキュリティー	180
11.3. RHEL FOR EDGE	185
11.4. ソフトウェア管理	185
11.5. シェルおよびコマンドラインツール	185

---

11.6. インフラストラクチャーサービス	186
11.7. ネットワーク	187
11.8. カーネル	189
11.9. ファイルシステムおよびストレージ	193
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	195
11.11. IDENTITY MANAGEMENT	195
11.12. デスクトップ	200
11.13. グラフィックインフラストラクチャー	201
11.14. RED HAT ENTERPRISE LINUX システムロール	201
11.15. 仮想化	202
11.16. クラウド環境の RHEL	207
11.17. サポート性	209
11.18. コンテナ	209
<b>付録A コンポーネント別のチケットリスト</b> .....	<b>211</b>
<b>付録B 改訂履歴</b> .....	<b>220</b>





## 多様性を受け入れるオープンソースの強化

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

## RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに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 9.3 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

Image Builder の主なハイライト:

- AWS EC2 AMD または Intel 64 ビットアーキテクチャー AMI イメージが強化され、従来の BIOS ブートに加えて UEFI ブートもサポートされます。

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:

- 次のイメージタイプのサポートが追加されました。
  - **minimal-raw**
  - **edge-vmware**
  - **edge-ami**
- 新しい FIDO Device Onboarding Servers コンテナイメージが利用可能になりました。
  - rhel9/fdo-manufacturing-server
  - rhel9/fdo-owner-onboarding-server
  - rhel9/fdo-rendezvous-server
  - rhel9/fdo-serviceinfo-api-server

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

### セキュリティ

セキュリティ関連の主なハイライト:

- **Keylime** はバージョン 7.3.0 にリベースされました。
- **keylime RHEL システムロール** が利用可能です。このロールを使用すると、Keylime verifier と Keylime registrar をより簡単に設定できます。
- **OpenSSH** は、暗号化を目的として、セキュリティレベルの低い SHA-1 メッセージダイジェストからさらに移行され、追加のシナリオではよりセキュリティレベルの高い SHA-2 が適用されます。
- **pcsc-lite-ccid** USB Chip/Smart Card Interface Device (CCID) および Integrated Circuit Card Device (ICCD) ドライバーがバージョン 1.5.2 にリベースされました。
- RHEL 9.3 では、すべての TLS 1.2 接続に対して FIPS-140-3 標準で必要とされる **Extended Master Secret (EMS) 拡張機能 (RFC 7627)** をサポートするためのさらなる改善が導入されています。
- SELinux ポリシー分析用のグラフィカルツール、コマンドラインツール、およびライブラリーのコレクションである **SEtools** が、バージョン 4.4.3 にリベースされました。

- **OpenSCAP** がバージョン 1.3.8 にリベースされました。
- **SCAP セキュリティーガイド** は、バージョン 0.1.69 にリベースされました。最も注目すべき点は次のとおりです。
  - ANSSI プロファイルがバージョン 2.0 に更新されました。
  - CCN-STIC-610A22 ガイドに合わせて、RHEL 9 に 3 つの新しい SCAP プロファイルが追加されました。

詳細は、[新機能 - セキュリティー](#) を参照してください。

### 動的プログラミング言語、Web サーバー、およびデータベースサーバー 次の Application Streams の新しいバージョンが利用可能になる

- **Redis 7**
- **Node.js 20**

さらに、**Apache HTTP Server** がバージョン 2.4.57 に更新されました。

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 更新されたシステムツールチェーン

RHEL 9.3 では、以下のシステムツールチェーンコンポーネントが更新されました。

- **GCC 11.4.1**

### パフォーマンスツールとデバッガーの更新

RHEL 9.3 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 3.21**
- **SystemTap 4.9**
- **elfutils 0.189**

### 更新されたパフォーマンスモニタリングツール

RHEL 9.3 では、以下のパフォーマンス監視ツールが更新されました。

- **PCP 6.0.5**
- **Grafana 9.2.10**

### 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 9.3 で更新されました。

- **GCC Toolset 13 (新規)**
- **LLVM Toolset 16.0.6**
- **Rust Toolset 1.71.1**
- **Go Toolset 1.20.10**

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。

## RHEL 9 の Java 実装

RHEL 9 AppStream リポジトリには、以下が含まれます。

- **java-21-openjdk** パッケージ。OpenJDK 21 Java Runtime Environment および OpenJDK 21 Java Software Development Kit を提供します。OpenJDK 21.0.1 セキュリティリリースもインストール可能です。OpenJDK 21.0.1 アップデートをインストールして、最新のセキュリティ修正を取得することが推奨されます。
- **java-17-openjdk** パッケージ。OpenJDK 17 Java Runtime Environment および OpenJDK 17 Java Software Development Kit を提供します。
- **java-11-openjdk** パッケージ。OpenJDK 11 Java Runtime Environment および OpenJDK 11 Java Software Development Kit を提供します。
- **java-1.8.0-openjdk** パッケージ。OpenJDK 8 Java Runtime Environment および OpenJDK 8 Java Software Development Kit を提供します。

Red Hat build of OpenJDK パッケージは、ポータブル Linux リリースと RHEL 9.3 以降のリリースの間で単一のバイナリセットを共有します。この更新により、ソース RPM から RHEL 上で OpenJDK パッケージを再構築するプロセスが変更されました。新しい再構築プロセスの詳細は、README.md ファイルを参照してください。これは Red Hat build of OpenJDK の SRPM パッケージで利用可能であるほか、**java-\*-openjdk-headless** パッケージによっても `/usr/share/doc` ツリーの下にインストールされます。

詳細は、[OpenJDK のドキュメント](#) 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 8.6 から RHEL 9.0、RHEL 8.8 から RHEL 9.2、および RHEL 8.9 から RHEL 9.3:
  - 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)
  - 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 8.6 から RHEL 9.0 および RHEL 8.8 から RHEL 9.2

詳細は、[Supported in-place upgrade paths for Red Hat Enterprise Linux](#) を参照してください。

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

SAP HANA で RHEL 9.2 にアップグレードする場合は、アップグレード前に、システムが SAP に対して認定されていることを確認してください。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。

主な機能拡張は、次のとおりです。

- **ftype=0** でフォーマットされた XFS ファイルシステムを備えたシステムで、ディスク容量の要件が大幅に軽減されました。
- アップグレードの目的でアップグレードプロセス中に作成されるディスクイメージのサイズが動的になりました。**LEAPP\_OVL\_SIZE** 環境変数が不要になりました。
- 既存のディスクパーティションに必要な空き領域の計算に関する問題が修正されました。システムの再起動が必要になる前に、不足している空きディスク領域が正しく検出されるようになり、アップグレード RPM トランザクションを続行するのに十分な空き領域がないファイルシステムがレポートに正しく表示されるようになりました。
- カスタム Leapp アクターを使用して、インプレースアップグレードプロセス中にサードパーティードライバーを管理できるようになりました。
- アップグレード前レポートとアップグレードレポートの概要が端末に印刷されるようになりました。
- Red Hat OpenStack Platform の RHEL Real Time および RHEL Real Time for Network Functions Virtualization (NFV) のアップグレードがサポートされるようになりました。

### 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、セキュリティー問題の特定、複雑なアプリケーションの迅速なデプロイメントおよび設定に役立ちます。最も一般的なアプリケーションには、以下のものがあります。

- [Registration Assistant](#)
- [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](#)

## 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 9.3 はカーネルバージョン 5.14.0-362.8.1 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. INSTALLATION

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ローリングストリームで利用できます。ローリングストリームは、RPM またはモジュールとしてパッケージ化されることがあります。

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 9.3 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

#### AWS EC2 イメージのレガシーブートと UEFI ブートの両方をサポート

以前は、RHEL Image Builder は、レガシーブートタイプのみをサポートする EC2 AMD または Intel 64 ビットアーキテクチャーの AMI イメージを作成していました。その結果、セキュアブートなど、UEFI ブートを必要とする特定の AWS 機能を利用できませんでした。この機能拡張により、AWS EC2 AMD または Intel 64 ビットアーキテクチャーの AMI イメージが拡張され、レガシー BIOS ブートに加えて UEFI ブートもサポートされるようになりました。その結果、UEFI でイメージをブートする必要がある AWS の機能を利用できるようになりました。

Jira:RHELDPCS-16339<sup>[1]</sup>

#### キックスタートファイルまたはカーネルドライバーをロードするための待機時間を追加する新しいブートオプション `inst.wait_for_disks=`

場合によっては、ブートプロセス中に **OEMDRV** ラベルが付いたデバイスからキックスタートファイルまたはカーネルドライバーをロードするのに数秒かかることがあります。待機時間を調整するために、新しいブートオプション `inst.wait_for_disks=` を使用できるようになりました。このオプションを使用すると、インストール前に待機する秒数を指定できます。デフォルト時間は **5** 秒に設定されていますが、**0** 秒を使用して遅延を最小限に抑えることができます。このオプションの詳細は、[ストレージブートオプション](#) を参照してください。

Bugzilla:2171811

#### GUI と TUI を使用して RHEL on ARM をインストールする際に、必要なカーネルを選択する機能

以前は、kernel-64k ページサイズの RHEL on ARM をインストールする場合、キックスタート方法以外の選択肢がありませんでした。この更新により、GUI または TUI を使用し、必要なカーネルバージョンを選択して RHEL on ARM をインストールできるようになりました。必要なカーネルを選択するオプションは、Kernel Options の Software Selection 画面で利用できます。

Bugzilla:2164819<sup>[1]</sup>

#### VMware vSphere (OVA) のサポート

この更新により、RHEL Image Builder を使用して VMware vSphere OVA ファイルをビルドするためのサポートが追加されました。Open Virtual Appliance (OVA) ファイルは、VMware vSphere 仮想化アプリケーションが使用する仮想アプライアンスです。OVA ファイルには、OVF 記述子ファイル、1つ以上の仮想マシンディスクイメージファイル (VMDK)、オプションのマニフェストファイル (MF)、および証明書ファイルなどの仮想マシンを記述するために使用されるファイルが含まれます。VMware vSphere (.ova) を使用することで、vSphere GUI クライアントを使用して、イメージを VMware vSphere により簡単にデプロイできます。イメージを起動する前に、結果として得られる仮想マシンをさらにカスタマイズできます。

Jira:RHELDPCS-16877<sup>[1]</sup>

#### DNS 処理を制御するための新しい `network` キックスタートオプション

次の新しいオプションを指定した `network` キックスタートコマンドを使用して、DNS 処理を制御できるようになりました。これらの新しいオプションは `--device` オプションと一緒に使用します。

- **--ipv4-dns-search** および **--ipv6-dns-search** オプションを使用すると、DNS 検索ドメインを手動で設定できます。これらのオプションは、NetworkManager プロパティをミラーリングします。次に例を示します。

```
network --device ens3 --ipv4-dns-search domain1.example.com,domain2.example.com
```

- **--ipv4-ignore-auto-dns** および **--ipv6-ignore-auto-dns** オプションを使用すると、DHCP から DNS 設定を無視できます。引数は必要ありません。

Bugzilla:2065754<sup>[1]</sup>

## 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<sup>[1]</sup>

## 4.2. セキュリティー

### Keylime がバージョン 7.3.0 にリベース

Keylime パッケージはアップストリームバージョン 7.3.0 に更新されました。このバージョンは、さまざまな機能拡張とバグ修正を提供します。最も注目すべき点は、許可リストと除外リストが Keylime ランタイムポリシーに統合されていることです。**convert\_runtime\_policy.py** スクリプトを使用して、2 つのリストを結合できます。

さらに、この更新では、影響度が中程度の 2 つの脆弱性 [CVE-2023-38200](#) および [CVE-2023-38201](#) が修正されます。

Jira:RHEL-476<sup>[1]</sup>

### Keylime のポートには SELinux ポリシーのより厳格なルールがある

Keylime によって使用されるポートは、Keylime SELinux ポリシーで **keylime\_port\_t** というラベルが付けられるようになりました。このポリシーにより、このラベルが付いたポートの TCP 接続が許可されるようになりました。これは、以前の Keylime SELinux ポリシーでは、すべての未定義ポートへの接続が許可されており、Keylime で使用されるポートのほとんどが未定義グループに属していたためです。その結果、この更新により Keylime SELinux ポリシーの粒度が向上し、ポートセキュリティーはより厳密化され、より適切にターゲットを絞ることができます。

Jira:RHEL-595<sup>[1]</sup>

### Audit が FANOTIFY レコードフィールドをサポートするようになる

**audit** パッケージの今回の更新で、**FANOTIFY** Audit レコードフィールドがサポートされるようになりました。Audit サブシステムは、特に次のような追加情報を **AUDIT\_FANOTIFY** レコードに記録するようになりました。

- **fan\_type**: FANOTIFY イベントのタイプを指定する

- **fan\_info**: 追加のコンテキスト情報を指定する
- **sub\_trust** と **obj\_trust**: イベントに関与するサブジェクトとオブジェクトの信頼レベルを示す

これにより、特定のケースで Audit システムがアクセスを拒否した理由をよりよく理解できます。これは、**fapolicyd** フレームワークなどのツールのポリシーを作成する場合に役立ちます。

Jira:RHELPLAN-161087<sup>[1]</sup>

### fapolicyd がトラブルシューティング用のルール番号を提供するようになる

この機能拡張では、新しいカーネルおよび Audit コンポーネントにより、**fapolicyd** サービスが拒否を引き起こすルールの番号を **fanotify** API に送信できるようになります。その結果、**fapolicyd** に関連する問題をより正確にトラブルシューティングできます。

Jira:RHEL-624

### crypto-policies は、FIPS モードでの TLS 1.2 接続に NO-ENFORCE-EMS サブポリシーを提供するようになる

システム全体の暗号化ポリシーに **NO-ENFORCE-EMS** サブポリシーが含まれるようになりました。新しいサブポリシーを適用した後、システムは、FIPS モードでネゴシエートされるすべての TLS 1.2 接続に対して Extended Master Secret (EMS) 拡張機能 (RFC 7627) を必要としなくなりました。これにより、システムは EMS または TLS 1.3 へのサポートなしにレガシーシステムに接続できるようになります。これは FIPS-140-3 標準の要件に違反することに注意してください。**update-crypto-policies --set FIPS:NO-ENFORCE-EMS** コマンドを入力して、サブポリシーを適用できます。

Bugzilla:2216257<sup>[1]</sup>

### GnuTLS は、FIPS モードで EMS を備えた TLS 1.2 が必要

FIPS-140-3 標準に準拠するために、GnuTLS サーバーとクライアントは、FIPS モードでネゴシエートされるすべての TLS 1.2 接続に対して、Extended Master Secret (EMS) 拡張 (RFC 7627) を必要とします。EMS をサポートしていない古いサーバーおよびクライアントとの互換性を維持する必要があり、TLS 1.3 を使用できない場合は、**NO-ENFORCE-EMS** システム全体の暗号化サブポリシーを適用できます。

```
# update-crypto-policies --set FIPS:NO-ENFORCE-EMS
```



#### 警告

EMS なしで TLS 1.2 接続を許可すると、システムは FIPS-140-3 要件を満たさなくなります。

Bugzilla:2157953

### NSS は FIPS モードで EMS を強制するようになる

Network Security Services (NSS) ライブラリーには、FIPS 140-3 標準で義務付けられているすべての TLS 1.2 接続に対して Extended Master Secret (EMS) 拡張 (RFC 7627) を要求する **TLS-REQUIRE-EMS** ポリシーが含まれるようになりました。NSS は、システム全体の暗号化ポリシーが **FIPS** に設定されている場合に、新しいポリシーを使用します。

EMS または TLS 1.3 をサポートしていないレガシーシステムとの相互運用が必要な場合は、**NO-ENFORCE-EMS** システム全体の暗号化サブポリシーを適用できます。このような変更は FIPS-140-3 要件に違反します。

[Bugzilla:2157950](#)

## OpenSSL が FIPS モードでの EMS の無効化をサポートするようになる

`/etc/pki/tls/fips_local.cnf` ファイルを編集することで、FIPS モードで Extended Master Secret (EMS) 拡張 (RFC 7627) なしで TLS 1.2 接続を許可するように OpenSSL 暗号化ライブラリーを設定できるようになりました。任意のテキストエディターで、次のセクションを設定ファイルに追加します。

```
[fips_sect]
tls1-prf-ems-check = 0
activate = 1
```

次に、`/etc/pki/tls/openssl.cnf` ファイルで SSL 設定セクションを見つけます。デフォルトの SSL 設定セクションは **crypto\_policy** です。SSL 設定セクションの最後に、次の行を追加します。

```
Options=RHNoEnforceEMSinFIPS
```

上記の設定変更により、FIPS モードのシステムは、EMS または TLS 1.3 のサポートなしにレガシーシステムに接続できるようになります。



### 警告

**update-crypto-policies --set FIPS:NO-ENFORCE-EMS** コマンドを入力すると、FIPS モードで TLS 1.2 に対する EMS の強制を停止できます。どちらの場合も、そのような設定変更は FIPS-140-3 標準の要件に違反します。

[Bugzilla:2216256<sup>\[1\]</sup>](#)

## OpenSSH は SHA-2 をさらに強制する

暗号化目的で安全性の低い SHA-1 メッセージダイジェストからさらに移行する取り組みの一環として、OpenSSH に次の変更が加えられました。

- **sshd** 起動時に、システムで SHA-1 の使用が設定されているかどうかのチェックを追加しました。SHA-1 が使用できない場合、OpenSSH は操作に SHA-1 を使用しようとしません。これにより、DSS キーが存在する場合はロードしなくなり、また、**rsa-sha2** の組み合わせが使用可能な場合には、その組み合わせを強制的にアドバタイズするようになります。
- SSH 秘密キーの変換では、OpenSSH は RSA キーのテストに明示的に SHA-2 を使用します。
- サーバー側で SHA-1 署名が使用できない場合、**sshd** は SHA-2 を使用してホストキーの証明を確認します。これは、RHEL 8 以前のバージョンのクライアントと互換性がない可能性があります。
- SHA-1 アルゴリズムがクライアント側で使用できない場合、OpenSSH は SHA-2 を使用します。

- クライアント側では、キー証明リクエストで SHA-1 が使用された場合、またはハッシュアルゴリズムが指定されていない場合 (デフォルトを想定)、OpenSSH はサーバーからの SHA-2 ベースのキー証明を許可します。これは、RSA 証明書の既存の例外と合わせており、サポートされている場合は最新のアルゴリズムを使用して接続できるようになります。

[Bugzilla:2070163](#)

### OpenSSL に、Bleichenbacher のような攻撃に対する保護が含まれるようになりました。

このリリースの OpenSSL TLS ツールキットでは、RSA PKCS #1 v1.5 復号化プロセスにおける Bleichenbacher のような攻撃に対する API レベルの保護が導入されています。PKCS #1 v1.5 復号化中にパディングをチェックする際にエラーを検出すると、RSA の復号化がエラーではなく、無作為に生成された確定的なメッセージを返すようになりました。この変更により、[CVE-2020-25659](#) および [CVE-2020-25657](#) などの脆弱性に対する一般的な保護が提供されます。

この保護を無効にするには、次を呼び出します: `EVP_PKEY_CTX_ctrl_str(ctx, "rsa_pkcs1_implicit_rejection"."0")` 関数。これは、RSA 復号化コンテキストで呼び出しますが、これによりシステムがより脆弱になります。

[Bugzilla:2153471](#)

### OpenSSL が、Groups オプションを通じて設定可能な Brainpool 曲線をサポートするようになる

OpenSSL TLS ツールキットのこの更新では、Elliptic Curve Cryptography (ECC) での Brainpool 曲線のサポートが導入されています。さらに、**Groups** 設定オプションを使用して、システム全体の暗号化ポリシーで曲線を制御できます。

次の Brainpool 曲線が OpenSSL ECC で有効化されています。

- **brainpoolP256r1**
- **brainpoolP256t1**
- **brainpoolP320r1**
- **brainpoolP320t1**
- **brainpoolP384r1**
- **brainpoolP384t1**
- **brainpoolP512r1**
- **brainpoolP512t1**

[Bugzilla:2188180](#)

### crypto-policies は、OpenSSL ECC Brainpool 曲線をサポートするようになる

システム全体の暗号化ポリシーのこの更新により、**group** オプションを使用して OpenSSL で次の Brainpool Elliptic Curve Cryptography (ECC) 曲線を制御できるようになりました。

- **BRAINPOOL-P256R1**
- **BRAINPOOL-P384R1**
- **BRAINPOOL-P512R1**

たとえば、次の行を含むサブポリシーを作成することで、OpenSSL でサポートされているすべての Brainpool Elliptic Curve を有効にできます。

```
group = BRAINPOOL-.*+
```

[Bugzilla:2193324](#)<sup>[1]</sup>

### crypto-policies は、デフォルトで OpenSSL と同じグループ順序を使用するようになる

このリリースでは、システム全体の暗号化ポリシー (**crypto-policies**) が、OpenSSL **Groups** 設定オプションのグループ順序を制御します。OpenSSL でのパフォーマンスを維持するために、**crypto-policies** は、OpenSSL の組み込み設定の順序と一致するデフォルトのグループ順序を使用します。その結果、GnuTLS などのグループ順序を制御する **crypto-policies** をサポートする RHEL 暗号化バックエンドは、OpenSSL と同じ順序を使用するようになりました。

[Jira:RHEL-59](#)<sup>[1]</sup>

### crypto-policies permitted\_encetypes が FIPS モードでのレプリケーションを破損しなくなる

この更新前は、RHEL 8 で実行されている IdM サーバーは、FIPS モードで RHEL 9 を実行している IdM レプリカに AES-256-HMAC-SHA-1 で暗号化されたサービスチケットを送信していました。その結果、デフォルトの **permitted\_encetypes krb5** 設定により、FIPS モードでの RHEL 8 IdM サーバーと RHEL 9 IdM レプリカ間のレプリケーションが破損していました。

システム全体の暗号化ポリシーのこの更新により、**permitted\_encetypes krb5** 設定オプションの値が並べ替えられ、より相互運用性の高い暗号化タイプをデフォルトで優先できるようになります。その結果、**permitted\_encetypes** 設定によって、FIPS モードの RHEL 8 IdM サーバーと RHEL 9 IdM レプリカ間のレプリケーションが破損しなくなりました。



#### 注記

Kerberos を使用する場合は、`/etc/crypto-policies/back-ends/krb5.config` ファイル内の **permitted\_encetypes** の値の順序を確認してください。別の順序が必要な場合は、カスタム暗号化サブポリシーを適用します。

[Bugzilla:2225222](#)

### pcsc-lite-ccid が 1.5.2 にリベース

**pcsc-lite-ccid** パッケージがバージョン 1.5.2 に更新されました。このバージョンは、さまざまなバグ修正と機能拡張を提供します。特に、次のとおりです。

- 新しいカードリーダーのサポート
- Alcor Micro AU9560 の修正

[Bugzilla:2209457](#)

### opensc が 0.23 にリベース

**opensc** パッケージがバージョン 0.23 に更新されました。このバージョンは、さまざまなバグ修正と機能拡張を提供します。特に、次のとおりです。

- 対称キーを使用した暗号化と復号化のサポートを追加しました。
- 512 バイトを超える長さのデータへの署名のサポートを追加しました。



- 古いカードドライバーのサポートをデフォルトで無効にしました
- 古いドライバー MioCOS および JCOP のサポートを削除しました。

Jira:RHEL-280<sup>[1]</sup>

### setools が 4.4.3 にリベース

**setools** パッケージがバージョン 4.4.3 に更新されました。このバージョンは、さまざまなバグ修正と機能拡張を提供します。特に、次のとおりです。

- Cython 3.0.0 でのコンパイルを修正しました。
- man ページを改善しました。
- **sediff**、**sesearch**、および **apol** の未使用オプションを削除しました。
- ソースタイプのフロー分析を取得するための **-r** オプションを **seinfoflow** コマンドに追加しました
- 権限のないルールは無効なポリシーとして自動的に拒否されます。

Bugzilla:2231801、Bugzilla:2184140

### 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。

- **qat**
- **systemd-pstore**
- **boothd**
- **fdo-manufacturing-server**
- **fdo-rendezvous-server**
- **fdo-client-linuxapp**
- **fdo-owner-onboarding-server**

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。

Bugzilla:2080443<sup>[1]</sup>、Bugzilla:2026795、Bugzilla:2181565、Bugzilla:2128833

### OpenSCAP が 1.3.8 にリベース

OpenSCAP パッケージがアップストリームバージョン 1.3.8 にリベースされました。このバージョンは、さまざまなバグ修正と機能拡張を提供します。特に、次のとおりです。

- 一部の **systemd** ユニットを無視しないように **systemd** プローブを修正しました
- **shadow** OVAL プローブにオフライン機能を追加しました
- **sysctl** OVAL プローブにオフライン機能を追加しました

- ネットワークファイルシステムのリストに **auristorfs** を追加しました
- **autotailor** ユーティリティーによって生成されたファイルの調整に関する問題の回避策を作成しました。

[Bugzilla:2217442](#)

## SCAP セキュリティーガイドがバージョン 0.1.69 にリベース

SCAP セキュリティーガイド (SSG) パッケージが、アップストリームバージョン 0.1.69 にリベースされました。このバージョンは、さまざまな機能拡張とバグ修正を提供します。最も注目すべき点は、National Cryptologic Center of Spain が 2022 年 10 月に発行した CCN-STIC-610A22 ガイドに準拠した RHEL 9 用の 3 つの新しい SCAP プロファイルが導入されていることです。

プロファイル名	プロファイル ID	ポリシーバージョン
CCN Red Hat Enterprise Linux 9 - Advanced	<b>xccdf_org.ssgproject.content_profile_ccn_advanced</b>	2022-10
CCN Red Hat Enterprise Linux 9 - Basic	<b>xccdf_org.ssgproject.content_profile_ccn_basic</b>	2022-10
CCN Red Hat Enterprise Linux 9 - Intermediate	<b>xccdf_org.ssgproject.content_profile_ccn_intermediate</b>	2022-10

[Bugzilla:2221697](#)

## ANSSI-BP-028 セキュリティープロファイルが、バージョン 2.0 に更新される

SCAP セキュリティーガイド内の以下の French National Agency for the Security of Information Systems (ANSSI) BP-028 は、バージョン 2.0 に合わせて更新されました。

- ANSSI-BP-028 Minimal Level
- ANSSI-BP-028 Intermediary Level
- ANSSI-BP-028 Enhanced Level
- ANSSI-BP-028 High Level

[Bugzilla:2155790](#)

## python3-greenlet-devel が CRB で利用可能になる

**python3-greenlet-devel** パッケージが CodeReady Linux Builder (CRB) リポジトリで利用できるようになりました。これは、明示的に有効にする必要があります。詳細は、ナレッジベース記事 [How to enable and make use of content within CodeReady Linux Builder](#) を参照してください。CRB リポジトリに含まれるパッケージはサポートされていないことに注意してください。

[Bugzilla:2149497](#)

## pam\_wheel.so モジュールで使用されるグループをチェックする SSG ルールが簡素化される

CIS Benchmark では、**sudo** コマンドを優先して **su** コマンドを制限する必要があります。SCAP セキュリティーガイド (SSG) は、**su** コマンドを特定のグループに制限する **pam\_wheel.so** モジュールでこの要件を満たしています。この更新により、このグループが存在し、メンバーが存在しないかを

チェックするルールが改善されました。その結果、ルールはより効率的になり、評価レポートの解釈が簡素化されます。

[Jira:RHEL-1905](#)

## 4.3. RHEL FOR EDGE

### 新しい FIDO Device Onboarding Servers コンテナイメージが利用可能になる

IoT およびエッジコンピューティングデバイスをオンボードするための次の FIDO Device Onboarding Servers コンテナイメージが、[Red Hat Container Catalog](#) で利用できるようになりました。

- rhel9/fdo-manufacturing-server コンテナイメージ
- rhel9/fdo-owner-onboarding-server コンテナイメージ
- rhel9/fdo-rendezvous-server コンテナイメージ
- rhel9/fdo-serviceinfo-api-server コンテナイメージ

[Jira:RHELPLAN-163133<sup>\[1\]</sup>](#)

### minimum-raw イメージタイプが 64 ビット ARM アーキテクチャーをサポートするようになる

この機能拡張により、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をサポートする **minimal-raw** イメージタイプを作成できるようになります。**minimum-raw** イメージは、**xz** 形式で圧縮された、事前にパッケージ化されたブート可能な最小限の RPM イメージです。イメージをブートするには、イメージを解凍し、SD カードなどのブート可能なデバイスにコピーする必要があります。イメージを展開するには、次のコマンドを入力します。

```
$ xz -d <_uuid-minimal-raw.img_.xz>
```

[Jira:RHELPLAN-163665<sup>\[1\]</sup>](#)

### コミット ID が composer-cli CLI の --parent 引数の値としてサポート対象となる

イメージのコミット ID を、**composer-cli** コマンドラインの **--parent** 引数の値として使用できるようになりました。イメージのコミット ID を取得するには、RHEL for Edge Commit イメージをダウンロードして展開します。展開された **.tar** ファイルで **ref** 名とコミット ID を見つけることができます。

[Jira:RHELDOCS-16386<sup>\[1\]</sup>](#)

### RHEL for Edge .ami イメージをビルドするためのサポート

この機能拡張により、オンプレミス RHEL Image Builder を使用して、RHEL for Edge の **.ami** イメージをビルドすることがサポートされるようになりました。初回起動時に、Ignition を使用してブループリントをカスタマイズし、認証情報をイメージに挿入できます。**.ami** イメージを AWS にアップロードし、AWS で EC2 インスタンスを起動できます。

[Jira:RHELDOCS-16708<sup>\[1\]</sup>](#)

### RHEL for Edge の .vmdk イメージをビルドするためのサポート

この機能拡張により、オンプレミス RHEL Image Builder を使用して、RHEL for Edge の **.vmdk** イメージをビルドすることがサポートされるようになりました。Ignition を使用してブループリントをカスタマイズして、初回起動時に認証情報をイメージに挿入できます。イメージを vSphere にロードし、VM

vSphere でイメージをブートできます。このイメージは、ESXi 7.0 U2、ESXi 8.0 以降と互換性があります。VM はバージョン 19 および 20 と互換性があります。

Jira:RHELDPCS-16709<sup>[1]</sup>

## パスワードを設定せずに初期ユーザーとして Edge システムにログインできるようになる

以前は、FDO オンボーディングプロセス中に作成された初期ユーザーとしてログインできませんでした。パスワードを **useradd** コマンドで設定していない場合でも、システムがパスワードを要求していたためです。この機能拡張により、パスワードの設定が任意となり、以前に **useradd** コマンドを使用してパスワードを設定していなかった場合でもログインできるようになりました。パスワードを入力せずに SSH キーを使用してログインすることもできますが、失敗した場合はパスワードの入力を求められることに注意してください。

Jira:RHELDPCS-17101<sup>[1]</sup>

## 4.4. ソフトウェア管理

### アップグレード後に自動再起動するための新しい DNF 自動 **reboot** オプション

この機能拡張により、DNF 自動 **reboot** オプションを使用して、アップグレード後に変更を適用するためにシステムが自動的に再起動するように設定できるようになりました。

**reboot** オプションは次の設定をサポートしています。

- **never** はシステムを再起動しません。これが現在の動作です。
- **when-changed** はアップグレード後に再起動をトリガーします。
- **when-needed** は、変更を適用するために再起動が必要な場合 (systemd またはカーネルがアップグレードされた場合など) にのみ再起動をトリガーします。

**reboot\_command** オプションを使用して、再起動に使用するコマンドをカスタマイズできます。デフォルトの **reboot** コマンドは **shutdown -r** です。

Bugzilla:2124793

### 新しい **--poweroff** オプションを使用すると、更新のインストール後にシステムをシャットダウンできる

この機能拡張により、新しい **--poweroff** オプションが **dnf system-upgrade** プラグインの **reboot** コマンドに追加されました。このオプションを使用すると、更新のインストール後に再起動する代わりにシステムをシャットダウンできます。

Bugzilla:2157844

### 新しい **dnf leaves** および **show-leaves** プラグインが DNF API で利用可能になる

この機能拡張により、以下の新しい DNF プラグインが利用可能になります。これにより、システムにインストール済みのパッケージで、他のインストール済みパッケージの依存関係として不要なものがリスト表示されます。

- **dnf leaves** は、すべてのパッケージをリスト表示します。
- **show-leaves** は、新しくインストールされたパッケージと、トランザクション後に他のインストール済みパッケージの依存関係として不要になったパッケージをリスト表示します。

[Bugzilla:2134638](#)

## 4.5. シェルおよびコマンドラインツール

### NetBackup サービスでバックアップが復元可能になる

NetBackup (NBU) バックアップ方式を使用する場合、ReaR は NetBackup サービスバージョン 10.1.1 のユニットファイルをレスキューイメージに組み込み、レスキューシステムの起動時にこれを起動するようになりました。その結果、回復プロセス中に NBU バックアップ方法を使用してシステムバックアップを復元し、復元を正常に完了できます。

[Bugzilla:2188593](#)

### opencryptoki が 3.21.0 にリベース

**opencryptoki** パッケージはバージョン 3.21.0 にリベースされ、多くの機能拡張とバグ修正が提供されています。最も注目すべき点は、**opencryptoki** が次の機能をサポートするようになったということです。

- ハードウェアセキュリティーモジュール (HSM) マスターキーの同時変更
- 選択したキーを保護されたキーに変換するための **protected-key** オプション
- DH、DSA、汎用シークレットキータイプなどの追加のキータイプ
- EP11 ホストライブラリーバージョン 4
- AES-XTS キーのタイプ
- IBM 固有の Kyber キーのタイプとメカニズム
- 追加の IBM 固有の Dilithium キーラウンド 2 および 3 のバリエーション

さらに、**pkcs11tool** スロットマネージャーは root として実行されなくなり、**opencryptoki** はさらなる強化を提供します。この更新により、次の新しいコマンドのセットも使用できるようになりました。

#### **p11sak set-key-attr**

キーを変更する場合

#### **p11sak copy-key**

キーをコピーする場合

#### **p11sak import-key**

キーをインポートする場合

#### **p11sak export-key**

キーをエクスポートする場合

[Bugzilla:2160061<sup>\[1\]</sup>](#)

### 更新された **systemd-udev** が、InfiniBand インターフェイスに一貫性のあるネットワークデバイス名を割り当てる

RHEL 9 で導入された **systemd** パッケージの新しいバージョンには、更新された **systemd-udev** デバイスマネージャーが含まれています。デバイスマネージャーは、InfiniBand インターフェイスのデフォルト名を、**systemd-udev** が選択した一貫性のある名前に変更します。

[systemd リンクファイルを使用した IPoIB デバイスの名前変更](#) の手順に従って、InfiniBand インターフェイスに名前を付けるためのカスタム命名規則を定義できます。

命名スキームの詳細は、[systemd.net-naming-scheme\(7\)](#) の man ページを参照してください。

[Bugzilla:2136937](#)

## 4.6. インフラストラクチャーサービス

### Postfix が SRV ルックアップをサポートするようになる

この機能拡張により、Postfix DNS サービスレコード解決 (SRV) を使用して、メールクライアントを自動的に設定し、サーバーの負荷を分散できるようになりました。さらに、Postfix 設定で次の SRV 関連のオプションを使用することで、一時的な DNS の問題や SRV レコードの設定ミスによるメール配信の中断を防ぐことができます。

#### `use_srv_lookup`

DNS SRV レコードを使用して、指定したサービスの検出を有効にできます。

#### `allow_srv_lookup_fallback`

カスケードアプローチを使用してサービスを見つけることができます。

#### `ignore_srv_lookup_error`

SRV レコードが利用できない場合やエラーが発生した場合でも、サービス検出は機能し続けることができます。

[Bugzilla:2134789](#)

### Generic LF-to-CRLF ドライバーが cups-filters で利用可能になる

この機能拡張により、CR+LF 文字を含むファイルを受け入れるプリンターで、LF 文字を CR+LF 文字に変換する Generic LF-to-CRLF ドライバーを使用できるようになりました。キャリッジリターン (CR) とラインフィード (LF) は、行の終わりを示す制御文字です。その結果、このドライバーを使用すると、LF 文字で終了するファイルをアプリケーションから CR+LF 文字のみを受け入れるプリンターに送信できます。Generic LF-to-CRLF ドライバーは、RHEL 7 の **text-only** ドライバーの名前が変更されたバージョンです。新しい名前は実際の機能を反映しています。

[Bugzilla:2229784](#)

## 4.7. ネットワーク

### RHEL on ARM は、RHEL 9.3 で wifi アダプターを完全にサポートするようになる

この機能拡張により、**arm64** プラットフォームの複数のカードの wifi アダプターへのアクセスを有効にすることができます。

wifi 接続の設定に関する詳細は、[wifi 接続の管理](#) を参照してください。

[Bugzilla:2208365<sup>\[1\]</sup>](#)

### NetworkManager が、`resolv.conf` で `no-aaaa` オプションをサポートするようになる

**NetworkManager** は、`resolv.conf` ファイルでの `no-aaaa` DNS オプションの追加をサポートするようになりました。DNS オプション設定で `no-aaaa` 値を使用すると、IPv6 DNS 解決を無効にできます。

[Bugzilla:2176137](#)

## nmstate は、静的 DNS 検索と動的 DNS ネームサーバーの混合をサポートするようになる

**nmstate** フレームワークは、静的 Domain Name System (DNS) 検索ドメインと、動的 DNS ネームサーバーの両方をサポートするようになりました。動的 DNS ネームサーバーは、**nmstate** が Dynamic Host Configuration Protocol (DHCP) または **autoconf** メカニズムから取得したものになります。以前は、動的設定が **nmstate** によって破棄されていたため、静的 DNS 検索ドメインは動的 DNS ネームサーバーと共存できませんでした。これにより、ネットワークのセットアップと管理に不要な複雑さと制限が生じることがよくありました。この機能拡張は、DNS 設定の管理の柔軟性を高めることを目的としています。その結果、**nmstate** は、次の順序で DNS 設定を保存するネットワークインターフェイスを見つけようとしています。

1. 優先インターフェイス。現在 DNS 設定を保持しており、DNS に対してまだ有効です。
2. 自動インターフェイス
3. IP が有効なインターフェイス

この機能拡張では、DHCP から学習した DNS ネームサーバーは削除されないことに注意してください。

以下は、この機能を適用するための YAML ファイルの例です。

```
---
dns-resolver:
  config:
    search:
      - example.com
      - example.org
interfaces:
  - name: eth1
    type: ethernet
    state: up
    ipv4:
      enabled: true
      dhcp: true
    ipv6:
      enabled: true
      dhcp: true
      autoconf: true
```

[Bugzilla:2179916](#)

## nmstate は、bridge.vlan-default-pvid NetworkManager 設定オプションをサポートするようになる

この更新により、**nmstate** フレームワークを使用して、**bridge.vlan-default-pvid** NetworkManager 設定オプションを設定できるようになりました。このオプションを使用すると、Linux ブリッジ VLAN フィルタリングを使用するときに、VLAN をサポートするブリッジインターフェイス上のタグなしトラフィックのデフォルトのポート VLAN 識別子 (PVID) を設定できます。この結果を達成するには、次の YAML 設定を使用します。

```
interfaces:
  - name: linux-br0
    type: linux-bridge
    state: up
    bridge:
```

```
options:
  vlan-default-pvid: 5
port:
  - name: eth1
    stp-hairpin-mode: false
    stp-path-cost: 100
    stp-priority: 32
  vlan:
    mode: access
    tag: 100
```

**bridge.vlan-default-pvid** のデフォルト値は1であることに注意してください。VLAN フィルタリングを有効にして0に設定すると、タグなしトラフィックはドロップされます。

[Bugzilla:2180795](#)

### dbus サービスが再起動されるとすぐに、NetworkManager サービスが再起動される

以前は、何らかの理由で **dbus** を再起動した後、**NetworkManager** が停止していました。この動作は最適ではなく、接続の損失を引き起こしました。したがって、この機能拡張により **NetworkManager** が更新され、より堅牢になり、**dbus** の再起動時に自動的に再起動されるようになります。

[Bugzilla:2161915](#)

### nm-cloud-setup ユーティリティーが IMDSv2 設定をサポートするようになる

ユーザーは、**nm-cloud-setup** ユーティリティーを使用して、Instance Metadata Service Version 2 (IMDSv2)を使用する AWS Red Hat Enterprise Linux EC2 インスタンスを設定できます。EC2 メタデータや新機能への不正アクセスを制限するセキュリティ強化に準拠するには、高度な機能を提供するために AWS と Red Hat のサービスを統合する必要があります。この機能拡張により、**nm-cloud-setup** ユーティリティーは保護された IMDSv2 トークンを使用することで、IMDSv2 トークンを取得して保存し、EC2 環境を検証して、利用可能なインターフェイスと IP 設定に関する情報を取得できるようになります。

[Bugzilla:2151986](#)

### 非推奨の ifcfg 形式を使用すると NetworkManager が通知する

**ifcfg** 形式の接続プロファイルは、RHEL 9 で非推奨になりました ([ifcfg 形式の NetworkManager 接続プロファイルが非推奨に](#) を参照)。この更新により、NetworkManager はこの形式の非推奨についてユーザーに通知します。

- NetworkManager は、**/etc/sysconfig/network-scripts/** ディレクトリー内の **ifcfg** 形式で接続プロファイルを処理する場合、次の警告を **systemd** ジャーナルに記録します。

```
Warning: the ifcfg-rh plugin is deprecated, please migrate connections to the keyfile format
using "nmcli connection migrate"
```

- **ifcfg** 形式でサポートされていないプロパティを変更しようとする、**nmcli** ユーティリティーは次のエラーを報告します。

```
Error: Failed to modify connection '<name>': failed to update connection: The ifcfg-rh plugin
doesn't support setting '<property>'. If you are modifying an existing connection profile saved
in ifcfg-rh format, please migrate the connection to keyfile using 'nmcli connection migrate
<connection_uuid>' or via the Update2() D-Bus API and try again.
```



これらの機能拡張の結果、NetworkManager は、非推奨の **ifcfg** 形式の接続プロファイルをまだ使用または変更している場合に、ユーザーに通知するようになりました。

**ifcfg** から **keyfile** 形式へのプロファイルの移行に関する詳細は、[NetworkManager プロファイルの ifcfg から keyfile 形式への移行](#) を参照してください。

[Bugzilla:2190375](#)

### NetworkManager は、ボンディング設定で **lACP\_active** オプションをサポートするようになる

NetworkManager を使用すると、ボンディング設定の **lACP\_active** オプションにより、Link Aggregation Control Protocol Data Units (LACPDU) フレームに対するきめ細かい制御が可能になります。**lACP\_active** オプションは、LACPDU フレームの動作も調整し、ボンディングセットアップでのこれらのフレームの定期的な送信を制御します。ネットワーク設定をカスタマイズするには、**lACP\_active** を **ON** または **OFF** に設定して、LACPDU フレームの定期送信を有効または無効にします。

[Bugzilla:2069001](#)

### NetworkManager は、ボンディングインターフェイスの **ns\_ip6\_target** オプションの設定をサポートするようになる

この機能拡張により、ボンディングインターフェイスの **ns\_ip6\_target** オプションの設定について、NetworkManager で監視ピアとして最大 16 個の IPv6 アドレスを指定することで、**arp\_interval** オプションを設定できるようになります。以前は、NetworkManager で IPv6 監視ピアを指定できませんでした。この更新により、**nmcli** ユーティリティーを使用して **Bond.options** パラメーターの **ns\_ip6\_target** オプションを設定できるようになりました。NetworkManager は、最大 16 個の IPv6 アドレスの指定を有効にすることで、この設定をボンディングインターフェイスに適用します。この機能拡張は、IPv4 と IPv6 の設定に同様に適用されます。

[Bugzilla:2069004](#)

### NetworkManager は、同じネットワークインターフェイス上で静的 IP 設定と DHCP IP 設定の両方をサポートするようになる

**nmstate** ユーティリティーを使用することで、DHCP または Ad-Hoc Network Autoconfiguration (autoconf) が有効なインターフェイスで、**dhcp: true** または **autoconf: true** 値を使用して静的 IP アドレスを割り当てることができるようになりました。

この機能拡張により、**nmstate** は IP アドレスの 2 つのプロパティをサポートします。

- **valid\_lft** は、秒単位での有効な有効期間を意味します
- **preferred\_lft** は、秒単位での優先する有効期間を意味します

両方のパラメーターのデフォルト値は **forever** で、静的を意味します。

上記のプロパティを使用すると、**nmstate** は DHCP/autoconf ベースの IP アドレスを無視して、クエリーされた状態を元に適用した後に動的 IP アドレスが静的 IP に変換されることを回避できます。動的 IP アドレスを持つ DHCP/autoconf 設定を無効にする必要がある場合、**nmstate** は、それらの動的 IP を静的 IP アドレスに変換します。

[Bugzilla:2177733](#)

### **nmstate** は、MAC アドレス識別可能なネットワークインターフェイスをサポートする

**nmstate** ユーティリティーは、インターフェイス名ではなく MAC アドレスを使用して、ネットワークインターフェイスへのネットワーク設定を直接サポートします。

この機能拡張により、基本インターフェイスに2つのプロパティーが導入されました。

- **identifier**: ネットワーク上の **name** または **mac-address** を識別します。デフォルト値は **name** です。
- **profile-name**: 文字列

**identifier** 変数が **mac-address** 値に設定されている場合、**nmstate** は **interface.name** ではなく、**interface.mac-address** を使用して、特定のネットワーク状態のネットワークインターフェイスを選択します。ネットワーク設定を保存するときに、**interface.profile-name** 変数が割り当てられていない場合、**nmstate** は、**interface.name** よりも **interface.profile-name** を優先します。現在のネットワーク状態を確認すると、**interface.profile-name** が **interface.name** と等しい場合、それは非表示のままになります。

[Bugzilla:2183214](#)

### NetworkManager は、ARP チェックが何回失敗した後にボンディングドライバーがポートをダウンとしてマークするかの定義をサポートする

この機能拡張により、NetworkManager のボンディング接続プロファイルに **arp\_missed\_max** オプションが追加されます。アドレス解決プロトコル (ARP) モニターを使用してボンディングのポートが稼働しているかどうかをチェックする場合、**arp\_missed\_max** を設定して、何回チェックが失敗した後にボンディングドライバーがポートをダウンとしてマークするかを定義できるようになりました。

[Bugzilla:2148684](#)

### NetworkManager はリンク関連のプロパティーの指定をサポートする

この機能拡張により、次のネットワークリンクプロパティーが NetworkManager 接続プロファイルに追加されます。

- **link.tx-queue-length** - パケット数で表した送信 (TX) キューの長さのサイズ。
- **link.gro-max-size** - デバイスが受け入れる Generic Receive Offload (GRO) パケットの最大サイズ (バイト単位)。
- **link.gso-max-segments** - デバイスが受け入れる Generic Segmentation Offload (GSO) パケットのセグメントの最大数。
- **link.gso-max-size** - GSO パケットの最大サイズ (バイト単位)。

以前は、これらのカーネル設定は、**ip** コマンドを使用するか、NetworkManager ディスパッチャスクリプトでそのようなコマンドを使用することによってのみ設定できました。この機能拡張により、これらの設定を接続プロファイルで直接設定できるようになりました。

NetworkManager は、これらのプロパティーを **keyfile** 形式の接続プロファイルでのみサポートし、非推奨の **ifcfg** 形式ではサポートしないことに注意してください。

[Bugzilla:2158328](#)

### nmstate で DHCP オプション **dhcp-send-hostname** および **dhcp-custom-hostname** の API サポートが利用可能

この機能拡張により、**nmstate** ユーティリティーは、接続ファイル内で次の2つの DHCP オプションの設定をサポートします。

- **dhcp-send-hostname: true** または **false** の値。DHCP 要求にホスト名または完全修飾ドメイン名 (FQDN) オプションが必要な場合は、そのオプションのホスト名が設定されます。デフォルトは **true** です。
- **dhcp-custom-hostname: <string>**.このオプションを使用して、DHCP リクエストのホスト名または FQDN オプションを設定します。値のタイプは文字列です。

#### DHCPv4 ネットワークプロトコルの場合

- ホスト名が FQDN の場合は、RFC 4702 のオプション (81) **Fully Qualified Domain Name (FQDN)** を参照してください。
- ホスト名が FQDN でない場合は、RFC 2132 のオプション (12) **Host Name** を参照してください。

#### DHCPv6 ネットワークプロトコルの場合

カスタム文字列、空のドメイン名をサポートし、DHCP リクエストのホスト名をオーバーライドします。RFC 4704 のオプション (29) **Fully Qualified Domain Name (FQDN)** を参照してください。

[Bugzilla:2187622](#)

### NetworkManager がバージョン 1.44.0 にリベース

**NetworkManager** パッケージはアップストリームバージョン 1.44.0 にアップグレードされ、以前のバージョンに対するバグ修正や機能拡張がいくつか追加されました。

- **リンク関連のプロパティ**が **NetworkManager** に追加されました。
- **arp\_missed\_max**、**lACP\_active**、および **ns\_ip6\_target** プロパティがボンディング接続プロファイルに追加されました。
- **ipv6.dhcp-pd-hint** 接続プロパティで DHCPv6 接頭辞委任ヒントを設定できるようになりました。
- **/etc/NetworkManager/NetworkManager.conf** ファイルの **[keyfile]** セクションで新しい **rename** パラメーターを有効にすると、プロファイル名 (**connection.id**) を変更すると、NetworkManager によって **/etc/NetworkManager/system-connections/** 内の接続プロファイルの名前が変更されます。外部アプリケーションまたはスクリプトがファイル名に依存している場合は、このパラメーターを有効にしないでください。
- 非パブリックのトップレベルドメイン (TLD) を含むホスト名を設定すると、NetworkManager は完全なホスト名の代わりにこの TLD を DNS 検索ドメインとして使用するようになりました。
- NetworkManager は、**/etc/NetworkManager/NetworkManager.conf** ファイルの **[global-dns]** セクションの DNS オプションを適用するようになりました。
- 他の依存サービスとの競合状態を回避するために、NetworkManager は D-Bus ツリーにデータを設定した後でのみ D-Bus 名を取得するようになりました。これにより、NetworkManager の起動時に遅延が発生する可能性があることに注意してください。
- NetworkManager は、プロファイルの同時変更を防ぐために、**Update2()** D-Bus 呼び出しに **version-id** 引数を追加するようになりました。

- NetworkManager は、DNS からシステムのホスト名を解決するために一時的な IPv6 アドレスを使用しなくなりました。
- 複数接続プロファイルの場合に予期しない動作が発生するのを防ぐために、NetworkManager は、接続ごとではなく、デバイスおよび接続ごとに残っている自動接続再試行回数を追跡するようになりました。
- NetworkManager は、**sysfs** ファイルシステムの代わりにカーネルの **netlink** インターフェイスを使用して、VLAN フィルタリングオプションを設定します。
- **nm-cloud-setup** ユーティリティーは、Amazon EC2 上の Instance Metadata Service Version 2 (IDMSv2) をサポートするようになりました。
- ユーザーは、**nmtui** アプリケーションで wifi および Wireless Wide Area Networks (WWANs) を有効または無効にできるようになりました。
- ボンディング、ブリッジ、およびチーム接続では、**/etc/NetworkManager/NetworkManager.conf** ファイルの **[main]** セクションの **ignore-carrier=no** 設定が使用されるようになりました。

[Bugzilla:2180966](#)

## SCTP は RHEL 9 のカーネルネットワーキングツリーの最新バージョンにリベース

Stream Control Transmission Protocol (SCTP) ネットワーキングサブシステムの注目すべき変更点は次のとおりです。

- 複雑なネットワーク環境内で SCTP トラフィックをセグメント化および分離するための仮想ルーティングおよび転送 (VRF) サポート。
- ネットワーク内で効率的かつ均等なリソース割り当てを確実にするための新しいストリームスケジューラー (**fair capacity** および **weighted fair queueing**)。

[Bugzilla:2189292](#)

## MPTCP は RHEL 9 のカーネルネットワーキングツリーの最新バージョンにリベース

Multipath TCP (MPTCP) プロトコル拡張機能における注目すべき変更点は次のとおりです。

- クライアント側のサポートを含む、TCP fastopen (TFO) 拡張機能のサポート。この機能により、ネットワークの遅延、効率、パフォーマンスが向上します。
- 複数の混合 IPv4/IPv6 サブフローをサポートし、両方の IP バージョンが使用されるネットワークでの柔軟性と適応性を高めます。

[Bugzilla:2193330<sup>\[1\]</sup>](#)

## xdp-tools パッケージがバージョン 1.4.0 にリベース

**xdp-tools** パッケージはバージョン 1.4.0 にアップグレードされ、これにより、複数のバグ修正と拡張機能が提供されています。主な変更点は、以下のとおりです。

- **xdp-bench** ユーティリティーは、マルチバッファ eXpress Data Path (XDP) と、カーネル内の **xdp\_load\_bytes()** ヘルパーのベンチマークのサポートを獲得しました。この機能により、大規模な最大伝送単位 (MTU) を使用してネットワークベンチマークテストを実施できます。
- **xdp-tools** のコマンドラインユーティリティーのロックが改善され、ユーティリティーが正常に終了しなかった場合にロックが古くならないようになりました。

- **libxdp** ライブラリーには、すでに開いている **AF\_XDP** ソケットの追加のファイル記述子を受け入れる新しい **xsk\_umem\_create\_with\_fd()** API が含まれています。プロセスに **CAP\_NET\_RAW** 権限がない場合、この関数を通常の **xsk\_umem\_create()** 関数の代わりに使用できます。

[Bugzilla:2218500](#)

## iproute がバージョン 6.2.0 にリベース

**iproute** パッケージはアップストリームバージョン 6.2.0 にアップグレードされ、以前のバージョンに対するバグ修正や機能拡張がいくつか追加されました。主な変更点は以下のとおりです。

- 新しい **ip stats** コマンドは、インターフェイス統計を管理および表示します。デフォルトでは、**ip stats show** コマンドは、ブリッジやボンディングを含むすべてのネットワークデバイスの統計情報を表示します。**dev** および **group** オプションを使用して出力をフィルタリングできます。詳細は、**ip-stats(8)** の man ページを参照してください。
- **ss** ユーティリティーは、スレッド情報を表示するための **-T (--threads)** オプションを提供するようになりました。これは、**-p (--processes)** オプションを拡張したものです。詳細は、**ss(8)** の man ページを参照してください。
- 新しい **bridge fdb flush** コマンドを使用すると、指定されたオプションに一致する特定の forwarding database (fdb) エントリーを削除できます。詳細は、**bridge(8)** の man ページを参照してください。

[Jira:RHEL-428<sup>\[1\]</sup>](#)

## カーネルは、特定の順序でのボンディングポートのアクティブ化をサポートする

この機能拡張により、**active-backup**、**balance-tlb**、または **balance-alb** モードでボンディングを設定する場合、カーネルの **netlink** インターフェイスは各ポートでの優先順位の設定をサポートします。優先度の値は 32 ビットの整数を使用し、値が大きいほど優先度が高くなります。その結果、ボンディングポートを特定の順序でアクティブ化できるようになりました。

この機能を使用するには、NetworkManager ポート接続プロファイルを作成または変更するときに **bond-port.prio** プロパティを設定して、優先順位を設定できます。

[Bugzilla:2092194<sup>\[1\]</sup>](#)

## 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る

[RHBA-2023:7748](#) アドバイザリーのリリースにより、**firewalld** サービスがアップグレードされ、次の両方の条件が満たされた場合に **iptables** 設定から既存のルールをすべて削除しなくなりました。

- **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-14694<sup>\[1\]</sup>](#)

## 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<sup>[1]</sup>

## 4.8. カーネル

### RHEL 9.3 のカーネルバージョン

Red Hat Enterprise Linux 9.3 は、カーネルバージョン 5.14.0-362.8.1 で配布されます。

[Bugzilla:2232554](#)

### NVIDIA Grace CPU のサポートが追加されました。

Red Hat Enterprise Linux 9.3 では、NVIDIA Grace ARM 64 ビット CPU のサポートが追加されました。

Jira:RHELDPCS-17055<sup>[1]</sup>

### RHEL カーネルが AutoIBRS をサポートするようになる

Automatic Indirect Branch Restricted Speculation (AutoIBRS) は、AMD EPYC 9004 Genoa プロセッサファミリーおよびそれ以降の CPU バージョンによって提供される機能です。AutoIBRS は Spectre v2 CPU の脆弱性に対するデフォルトの軽減策であり、パフォーマンスを向上させ、スケーラビリティを向上させます。

[Bugzilla:1898184](#)<sup>[1]</sup>

### perf がバージョン 6.2 にリベース

**perf** パフォーマンス分析ツールはバージョン 6.2 にリベースされました。多数のマイナーなバグ修正と更新とは別に、**perf list** コマンドは、わかりやすい名前と説明を含む Performance Monitor Unit (PMU) イベントを表示するようになりました。さらに、この更新により、次のプロセッサのサポートが追加されます。

- Intel 第 13 世代 Core プロセッサ (Intel Raptor Lake-S)
- Intel 第 14 世代プロセッサ (Intel Meteor Lake)

- Intel 第5世代 Xeon サーバードプロセッサ (Intel Emerald Rapids)

[Bugzilla:2177180<sup>\[1\]</sup>](#)

### Intel® QAT カーネルドライバーがアップストリームバージョン 6.2 にリベース

Intel® Quick Assist Technology (QAT) は、アップストリームバージョン 6.2 にリベースされました。Intel® QAT には、対称および非対称暗号化、圧縮パフォーマンス、その他の CPU を集中的に使用するタスク用に最適化されたアクセラレーターが含まれています。

リベースには多くのバグ修正と機能拡張が含まれています。最も注目すべき機能拡張は、QAT GEN4 の次のハードウェアアクセラレータデバイスのサポートが利用可能になったことです。

- Intel Quick Assist Technology 401xx デバイス
- Intel Quick Assist Technology 402xx デバイス

[Bugzilla:2144528<sup>\[1\]</sup>](#)

### vTPM 機能が Linux コンテナで使用可能になる

この機能強化により、Linux コンテナおよびその他の仮想環境用の仮想 Trusted Platform Module (vTPM) が導入されます。vTPM は、安全な実行環境に使用する専用の TPM インスタンスを提供する TPM の仮想化バージョンです。vTPM プロキシドライバーを使用すると、プログラムは物理 TPM と対話するのと同じ方法で、エミュレートされた TPM と対話します。

その結果、各仮想マシンは分離され暗号化された専用の vTPM インスタンスを持つことができるようになりました。

[Bugzilla:2210263<sup>\[1\]</sup>](#)

### crash がバージョン 8.0.3 にリベース

crash は、実行中のシステムと、カーネルクラッシュの際に **kdump** によって作成されるコアダンプ ファイルを分析する対話型ユーティリティです。crash ユーティリティは、バージョン 8.0.3 にリベースされ、これには多くのバグ修正と機能拡張が含まれています。最も注目すべき機能拡張は、IPv6 サポートの追加です。

IPv6 をサポートするネットワークインターフェイスの場合、**crash** は、**net** または **net -s** コマンドを使用して IPv6 アドレスを出力します。

- **net** コマンドは、ネットワークデバイス、名前、および IP アドレスのリストを表示します。
- **net -s** コマンドは、以下の情報を表示します。
  - オープンネットワークソケットとソケットアドレス
  - ソケットとソケットアドレスのファミリーとタイプ
  - **INET** および **INET6** ファミリーの送信元および宛先のアドレスとポート

[Bugzilla:2170283](#)

### LVM シンプロビジョニングされたストレージボリュームが **vmcore** ダンプターゲットとしてサポートされる

**kdump** メカニズムは、シンプロビジョニングされた論理ボリュームを **vmcore** ターゲットとしてサポートするようになりました。LVM シンプロビジョニングを設定するには、次の手順を実行します。

1. LVM ボリュームグループを作成します。

```
vgcreate vg00 /dev/sdb
```

2. 利用可能な領域が 10 MB の LVM シンプルを作成します。

```
lvcreate -L 10M -T vg00/thinpool
```

3. 300 MB のファイルシステム領域を持つ LVM シンボリュームを作成します。

```
lvcreate -V 300M -T vg00/thinpool -n thinvol  
mkfs.ext4 /dev/vg00/thinvol
```

4. LVM シンプルのしきい値を設定して、スペースを自動的に拡張します。

```
cat /etc/lvm/lvm.conf  
activation {  
  thin_pool_autoextend_threshold = 70  
  thin_pool_autoextend_percent = 20  
  monitoring = 1  
}
```

5. 最初のカーネルの LVM シンプル監視サービスを有効にします。

```
systemctl enable lvm2-monitor.service  
systemctl start lvm2-monitor.service
```

6. 次の行を **kdump.conf** ファイルに追加して、LVM シンボリュームを **kdump** ターゲットとして設定します。

```
ext4 /dev/vg00/thinvol  
path /
```

7. **kdump** サービスを起動します。

```
kdumpctl restart
```

8. カーネルパニックをトリガーして設定を確認し、**vmcore** が **/dev/vg00/thinvol** に保存されているか確認します。

その結果、この機能拡張により、**kdump** メカニズムはシンプロビジョニングされたストレージボリュームに **vmcore** ダンプファイルを保存する機能を拡張しました。

[Bugzilla:2083475](#)

**makedumpfile** がアップストリームバージョン 1.7.3 にリベースされました。

**makedumpfile** ツールが、アップストリームバージョン 1.7.3 にリベースされました。これは、ページを圧縮するか、不要なメモリーページを除外することでクラッシュダンプファイルを小さくするツールです。リベースには多くのバグ修正と機能拡張が含まれています。

最も注目すべき変更点は、AMD および Intel 64 ビットアーキテクチャー上のスタンドアロンダンプ (**sadump**) メカニズムに 5 レベルのページングモードが追加されたことです。5 レベルのページングモードは、プロセッサのリニアアドレスの幅を拡張し、アプリケーションがより大量のメモリーにア



クセスできるようにします。5レベルのページングにより、仮想アドレスのサイズが48ビットから57ビットに拡張され、物理アドレスのサイズが46ビットから52ビットに拡張されます。

[Bugzilla:2173815](#)

## Red Hat Enterprise Linux は ARM の SystemReady ES および IR 層をサポートする

Red Hat Enterprise Linux では、以前は ARM の SystemReady SR バンドのみがサポートされていましたが、ES および IR をサポートするようになりました。RHEL 9.3 では、NVIDIA Orin、NXP i.MX 8M、および NXP i.MX 8M Mini モジュールが有効になっており、RHEL ハードウェア認定の候補となっています。ハードウェアパートナーは、Red Hat ハードウェア認定ジャーニーに登録することで **認定を提出** できます。お客様は、カタログに記載されているサポート対象のハードウェアを使用して、実稼働環境でのエクスペリエンスを向上させることができます。

[Bugzilla:2195986<sup>\[1\]</sup>](#)

## RHEL on ARM が Bluetooth をサポートするようになる

この機能拡張により、コマンドラインインターフェイスで **bluetoothctl** ツールを使用して Bluetooth デバイスを設定できるようになりました。

[Bugzilla:2187856<sup>\[1\]</sup>](#)

## RHEL on ARM は、RHEL 9.3 で USB 接続カメラを完全にサポートするようになる

この機能拡張により、AMD および Intel 64 ビットアーキテクチャプラットフォーム上の RHEL の **CONFIG\_MEDIA\_SUPPORT** カーネル設定が有効になります。これにより、AMD および Intel 64 ビットアーキテクチャーシステムで USB カメラを使用できるようになります。

[Bugzilla:2192722<sup>\[1\]</sup>](#)

## bpf がバージョン 6.3 にリベース

Berkeley Packet Filter (BPF) 機能は、Linux カーネルバージョン 6.3 にリベースされました。注目すべき変更点と機能拡張は次のとおりです。

- BPF トランポリンが 64 ビット IBM Z アーキテクチャーで使用できるようになりました。
- 新しいマップタイプ - **BPF\_MAP\_TYPE\_USER\_RINGBUF** - および関連ヘルパーが、BPF 固有のリングバッファを介したユーザー空間とカーネル間の通信用に定義されました。
- BPF は、リンクされたリストと **rmtree** という新しい複雑なデータ構造を提供するようになりました。
- プログラムをトレースする BPF トランポリンが **struct** 引数をサポートするようになりました。
- BPF は、NIC でサポートされている XDP 機能をエクスポートする方法を提供するようになりました。
- ハードウェアメタデータは、RX ハッシュおよびタイムスタンプメタデータの初期サポートを備えた BPF カーネル関数 (**kfuncs**) を使用することにより、XDP プログラムに公開されるようになりました。
- BPF は、BPF プログラムの新しい **contrack** モジュールエントリーに送信元および宛先の NAT アドレスとポートを設定するヘルパーを提供するようになりました。

- BPF は、netfilter パケットフィルタリングフレームワークの **nf\_conn:mark** 接続マークに直接書き込むことができるようになりました。

Bugzilla:2178930<sup>[1]</sup>

## 4.9. ブートローダー

### BLS での grub2-mkconfig の新しいデフォルトの動作

ブートローダー仕様(BLS)フレームワークでは、GRUB は起動時に BLS スニペットからブートメニューを動的に生成し、**grub.cfg** ファイルで事前定義されていません。

以前は、**grub2-mkconfig** コマンドは新しい **grub.cfg** ファイルを生成し、常にすべての BLS スニペットのコマンドライン引数を、**/etc/default/grub** ファイルにある **GRUB\_CMDLINE\_LINUX** 変数の値で上書きしていました。

今回のリリースで、**grub2-mkconfig** コマンドはデフォルトで BLS スニペットのカーネルコマンドラインを上書きしなくなりました。ブートローダーメニューの各カーネルは、BLS スニペットからカーネルコマンドラインを取得します。この新しいデフォルト動作は、**GRUB\_ENABLE\_BLSCFG=true** オプションが原因で発生します。

カーネルが BLS スニペットを無視し、代わりに **GRUB\_CMDLINE\_LINUX** からコマンドラインを取得するために **grub.cfg** を再生成するには、**GRUB\_ENABLE\_BLSCFG=false** オプションを設定します。

**GRUB\_CMDLINE\_LINUX** に従って BLS スニペットのカーネルコマンドラインを更新するには、**--update-bls-cmdline** オプションを追加します。

```
# grub2-mkconfig -o /path/to/grub.cfg --update-bls-cmdline
```

また、**grubby** を使用して、個々のカーネルの BLS スニペットに変更を加えることができる点にも留意してください。

```
# grubby --update-kernel /path/to/kernel --args "new args"
```

Jira:RHELDPCS-16752<sup>[1]</sup>

## 4.10. ファイルシステムおよびストレージ

### NFS サーバーは、nfsd 用の Courteous Server コードを実装するようになる

この更新では、RHEL カーネル NFS サーバーに **nfsd** 用の Courteous Server コードの実装が導入されています。この新機能により、NFS サーバーは、クライアントが接続していないときにアクセスの競合がない限り、サーバーとの接続が長期間失われたクライアントのリースを取り消さなくなります。

Bugzilla:2180124

### DAX マウントオプションと reflink が互換性を持つようになりました

この更新により、reflink されたファイルは一般的に DAX モードと互換性を持つようになりました。ファイルシステムの DAX マウントオプション **-o dax=always** は、reflink が有効なファイルシステムと互換性があります。reflink されたファイルは、inode フラグを使用して DAX モードに設定できます。詳細は、**xfs(5)** の man ページを参照してください。

Bugzilla:2192730<sup>[1]</sup>

## RPCSEC GSS Kerberos V5 の新しい暗号化タイプ

RPCSEC GSS Kerberos V5 メカニズムは、RFC 6803 (Kerberos 5 の Camellia Encryption) および RFC 8009 (Kerberos 5 の HMAC-SHA2 を使用した AES 暗号化) で定義された暗号化タイプをサポートするようになりました。

次の暗号化タイプが追加されました。

- **camellia128-cts-cmac**
- **camellia256-cts-cmac**
- **aes128-cts-hmac-sha256-128**
- **aes256-cts-hmac-sha384-192**

これにより、NFS クライアントと NFS サーバーは、GSS コンテキストをネゴシエートするときに、より強力な暗号化タイプを使用できるようになります。

[Bugzilla:2178741](#)

## fuse3 では、umount をトリガーせずにディレクトリーエントリーを無効にできるようになる

この更新により、新しいメカニズムが **fuse3** パッケージに追加されました。これにより、エントリーに存在するマウントの **umount** を自動的にトリガーすることなく、ディレクトリーエントリーを無効にすることができます。

[Bugzilla:2188182](#)

## Stratis ストレージマネージャーが利用可能になる

Stratis はローカルストレージマネージャーです。ユーザーへの追加機能を備えたストレージプールに管理されたファイルシステムを提供します。

- スナップショットおよびシンプロビジョニングを管理する
- 必要に応じてファイルシステムのサイズを自動的に大きくする
- ファイルシステムを維持する
- プールレベルの暗号化
- TMP2 および NBDE のサポート

Stratis ストレージを管理するには、バックグラウンドサービス **stratisd** と通信する **stratis** ユーティリティを使用します。

詳細は、Stratis ドキュメントの [Setting up Stratis file systems](#) を参照してください。

[Bugzilla:2041558](#)

## GFS2 ファイルシステムの設定と操作の改善

GFS2 ファイルシステムに対して次の更新が実装されました。

- **mkfs.gfs2** コマンドは、新しい **-U** オプションをサポートするようになりました。これにより、作成するファイルシステムのファイルシステム UUID を指定できるようになります。このオプションを省略した場合、ファイルシステムの UUID はランダムに生成されます。

- **gfs2\_jadd** コマンドは、以前のリリースよりもはるかに速い速度でジャーナルを作成します。
- GFS2 の man ページが改善されました。

[Bugzilla:2170017](#)

## dmpd がバージョン 1.0.2 にリベース

**dmpd** パッケージがバージョン 1.0.2 にアップグレードされました。主な変更点は、以下のとおりです。

- メモリーの安全性を確保し、複数のスレッドを使用してパフォーマンスを向上させるためにツールを Rust 言語で書き直しました。
- システム起動に伴う LVM プールのアクティブ化の時間を節約するために **thin\_check** ツールと **cache\_check** ツールを改善しました。これらのツールの実行に必要な時間は、以前のバージョンと比較して 10 倍以上改善されました。
- スナップショットのメタデータ **btrees** の共有が失われないように、**thin\_dump** ツールと **thin\_restore** ツールを更新しました。これで、復元されたメタデータには、さらなるスペースが必要なくなりました。
- 新しい **thin\_metadata\_pack** ツールと **thin\_metadata\_unpack** ツールを追加して、シンメタデータを通常、10 分の 1 のサイズに圧縮します。これは汎用コンプレッサーよりも優れています。このツールを使用すると、破損したメタデータを検査のために渡すことが簡単になります。

[Bugzilla:2175198](#)

## SCSI デバイス用に新しいデバイスごとのカウンターが追加される

SCSI 更新の I/O タイムアウト用に、新しいデバイスごとのカウンター **iotmo\_cnt** が追加されました。I/O リクエストの **iorequest\_cnt** 数、**iodone\_cnt** I/O 完了数、**ioerr\_cnt** I/O エラー数に加えて、リクエストのタイムアウト数も確認できます。以下に例を示します。

```
/sys/devices/pci0000:16/0000:16:02.0/0000:17:00.0/host2/target2:2:0/2:2:0:0/iorequest_cnt
/sys/devices/pci0000:16/0000:16:02.0/0000:17:00.0/host2/target2:2:0/2:2:0:0/iodone_cnt
/sys/devices/pci0000:16/0000:16:02.0/0000:17:00.0/host2/target2:2:0/2:2:0:0/iotmo_cnt
/sys/devices/pci0000:16/0000:16:02.0/0000:17:00.0/host2/target2:2:0/2:2:0:0/ioerr_cnt
```

[Bugzilla:2171093](#)<sup>[1]</sup>

## mpathcleanup は、device-mapper-multipath 内のマルチパスデバイスをフラッシュする

**mpathcleanup** ツールは、SCSI ベースのマルチパスデバイス上で動作し、SCSI パスデバイスとともにマルチパスデバイスを削除します。一部のユーザーは、マルチパスデバイスとそのパスデバイスを定期的に削除する必要があります。以前は、マルチパスデバイスを削除するために使用できるツールはなく、この操作にはユーザー定義のスクリプトが必要でした。

この新しいツールを使用すると、ユーザーはマルチパスデバイスとその基盤となるストレージを簡単に削除できるようになり、この操作のためのスクリプトを作成する必要はありません。

[Jira:RHEL-782](#)<sup>[1]</sup>

## nvme-cli がバージョン 2.4 にリベース

**nvme-cli** パッケージは、バージョン 2.4 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。

- TLS over TCP をサポートします。
- **/etc/fstab** ファイルを使用してファイルシステムをマウントするための **systemd** 自動接続サービスの誤った順序を修正しました。
- **u32** 値の出力を修正しました。
- ストレージタグのサイズを正しく検証します。
- ファブリックコントローラーの **nvme effects-log** コマンドをサポートします。

[Bugzilla:2159929<sup>\[1\]</sup>](#)

## 4.11. 高可用性およびクラスター

### 物理ボリュームが欠落している LVM ボリュームグループのフェイルオーバーのサポート

**LVM-activate** リソースエージェントは、ボリュームグループに物理ボリュームがない場合に、ボリュームグループのフェイルオーバーを許可する 2 つの新しいオプションをサポートするようになりました。

- **majoritypvs** オプションを使用すると、ボリュームグループから物理ボリュームが欠落している場合でも、大部分の物理ボリュームが存在する限り、ボリュームグループのシステム ID を変更できます。
- **degraded\_activation** オプションを使用すると、論理ボリューム内のすべてのデータを提供するのに十分なデバイスが RAID で利用可能である限り、レグが欠落しているときにボリュームグループ内の RAID 論理ボリュームをアクティブ化できます。

[Bugzilla:2174911<sup>\[1\]</sup>](#)

### IPaddr2 および IPsrcaddr クラスターリソースエージェントがポリシーベースのルーティングをサポートするようになる

**IPaddr2** および **IPsrcaddr** クラスターリソースエージェントは、ポリシーベースのルーティングをサポートするようになりました。これにより、複雑なルーティングシナリオを設定できるようになります。ポリシーベースのルーティングでは、リソースエージェントの **table** パラメーターを設定する必要があります。

[Bugzilla:2142518](#)

### Filesystem リソースエージェントが EFS ファイルシステムタイプをサポートするようになる

**ocf:heartbeat:Filesystem** クラスターリソースエージェントは、Amazon Elastic File System (EFS) をサポートするようになりました。**Filesystem** リソースを設定するときに **fstype=efs** を指定できるようになりました。

[Bugzilla:2142002](#)

### 新しい pcs 解析では、メタ属性のクローンを指定するときに meta キーワードが必要

**pcs** コマンド形式の一貫性を確保するために、**meta** キーワードを指定せずに **pcs resource clone**、**pcs resource promotable**、および **pcs resource create** コマンドを使用してメタ属性をクローンする設定は非推奨になりました。

以前は、**pcs resource clone** および **pcs resource promotable** コマンドでは、**meta** キーワードが無視されていました。ただし、**pcs resource create** コマンドでは、**clone** キーワードに続く場合に、**meta** キーワードの後に指定されたメタ属性は、クローンではなくリソースに割り当てられました。この更新された解析アルゴリズムでは、**meta** キーワードが **clone** キーワードに続く場合に、その後指定されたメタ属性がクローンに割り当てられます。古い形式に依存する既存のスクリプトとの互換性を維持するには、**pcs resource create** コマンドでクローンリソースを作成するときに **--future** コマンドオプションを指定して、この新しい引数の処理を有効にする必要があります。

次のコマンドは、メタ属性 **mv=v1** を持つリソースとメタ属性 **mv=v2** を持つクローンを作成するようになりました。

```
pcs resource create dummy1 ocf:pacemaker:Dummy meta m1=v1 clone meta m2=v2 --future
```

[Bugzilla:2168155](#)

### 設定されたリソース制約を再作成するための pcs コマンドの表示

新しい **--output-format=cmd** オプションを指定して **pcs constraint** コマンドを使用することにより、別のシステム上で設定済みのリソース制約を再作成するために使用できる **pcs constraint** コマンドを表示できるようになりました。デフォルトの出力形式は、以前のリリースと同様にプレーンテキストで、**--output-format=text** オプションで指定できます。プレーンテキスト形式は、他の **pcs** コマンドの出力形式との一貫性を保つためにわずかに変更されました。

[Bugzilla:2163953](#)

### Pacemaker パッケージがバージョン 2.1.6 にリベース

Pacemaker パッケージはアップストリームバージョン 2.1.6 にアップグレードされ、以前のバージョンに対するバグ修正や機能拡張がいくつか追加されました。

次の機能が追加されました。

- 以前は、Pacemaker Remote 接続が失われると、Pacemaker は常に一時ノード属性をパーズしていましたが、接続がすぐに回復でき、その間にリモートデーモンが再起動されなかった場合、これは不要でした。Pacemaker Remote ノードは、回復可能な短時間の接続停止の後でも一時的なノード属性を保持するようになりました。
- **alert\_snmp.sh.sample** アラートエージェント (Pacemaker に付属のサンプルアラートエージェント) は、SNMPv3 プロトコルと SNMPv2 をサポートするようになりました。この更新により、**alert\_snmp.sh.sample** エージェントを変更せずにコピーして、Pacemaker アラートで SNMPv3 を使用できるようになります。
- Pacemaker アラートとアラート受信者は、**enabled** メタオプションをサポートするようになりました。アラートに対してこのオプションを **false** に設定すると、アラートが無効になります。このオプションをアラートに対して **true** に設定し、特定の受信者に対して **false** に設定すると、その受信者のアラートが無効になります。このオプションのデフォルト値は **true** です。このオプションを使用すると、計画されたメンテナンスなどの理由でアラートを一時的に無効にすることができます。

以下のバグが修正されました。

- Pacemaker Designated Controller の選出は、保留中のアクションがすべて完了し、アクションの結果が失われなくなるようになりました。
- **devices** 属性が設定されていない場合、**fence\_scsi** エージェントは共有 **lvmlckd** デバイスを自動検出できるようになりました。
- リソースのスティッキネスがコロケーションスコアと適切に比較されるようになりました。

- **crm\_resource** コマンドで、アクティブなレプリカが1つだけあるバンドルを禁止または移動できるようになりました。
- 以前は、プロモート可能なクローンインスタンスは、プロモートされたインスタンスが最初となる番号順に割り当てられていました。その結果、プロモートされたクローンインスタンスを起動する必要がある場合、インスタンス番号が変更されたため、プロモートされていないインスタンスが予期せず再起動する場合があります。この修正により、ノードにインスタンス番号を割り当てるときにロールが考慮されるようになり、その結果、不要な再起動が発生しなくなります。

[Bugzilla:2189301](#)

### pcs property コマンドの機能拡張

**pcs property** コマンドは、次の機能拡張をサポートするようになりました。

- **pcs property config --output-format=** オプション
  - **--output-format=cmd** を指定すると、現在のクラスタープロパティ設定から作成された **pcs property set** コマンドが表示されます。このコマンドを使用すると、設定済みのクラスタープロパティを別のシステム上に再作成できます。
  - **--output-format=json** を指定すると、設定済みのクラスタープロパティが JSON 形式で表示されます。
  - 設定済みのクラスタープロパティをプレーンテキスト形式で表示するには、**output-format=text** を指定します。これは、このオプションのデフォルト値です。
- **pcs propertydefaults** コマンド。これは、非推奨となった **pcs property --defaults** オプションに代わるものです。
- **pcs property description** コマンド。これは、クラスタープロパティの意味を説明します。

[Bugzilla:2163914](#)

## 4.12. 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

### 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。

[RHSA-2024:2024](#) の更新では、新しい環境変数

**PYTHON\_EMAIL\_DISABLE\_STRICT\_ADDR\_PARSING** が導入されています。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<sup>[1]</sup>

## 新しい **nodejs:20** モジュールストリームが完全にサポートされるようになりました。

以前はテクノロジープレビューとして利用できた新しい **nodejs:20** モジュールストリームは、[RHEA-2023:7252](#) アドバイザリーのリリースで完全にサポートされています。**nodejs:20** モジュールストリームでは、LTS (Long Term Support) バージョンの **Node.js 20.9** が提供されるようになりました。

RHEL 9.3 に含まれる **Node.js 20** は、RHEL 9.1 以降で利用可能な **Node.js 18** に比べて、多数の新機能、バグ修正、セキュリティ修正、およびパフォーマンスの改善を提供します。

主な変更点は、以下のとおりです。

- **V8** JavaScript エンジンがバージョン 11.3 にアップグレードされました。
- **npm** パッケージマネージャーがバージョン 9.8.0 にアップグレードされました。
- **Node.js** に、新しい実験的な権限モデルが導入されています。
- **Node.js** に、新しい実験的な Single Executable Application (SEA) 機能が導入されています。
- **Node.js** は、Experimental ECMAScript モジュール (ESM) ローダーの改良を提供します。
- **Node.js 18** で実験的な **node:test** モジュールとして導入されたネイティブテストランナーは、現在は安定していると考えられています。
- **Node.js** は、さまざまなパフォーマンスの向上をもたらします。

**nodejs:20** モジュールストリームをインストールするには、以下を使用します。

```
# dnf module install nodejs:20
```

**nodejs:18** ストリームからアップグレードする場合は、[後のストリームへの切り替え](#) を参照してください。

**nodejs** Application Streams 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。

Bugzilla:2186717

## Python tarfile 抽出関数への新しい filter 引数

[CVE-2007-4559](#) の問題を軽減するために、Python は **tarfile** 抽出関数に **filter** 引数を追加します。この引数により、安全性を高めるために **tar** 機能をオフにすることができます ([CVE-2007-4559](#) ディレクトリートラバーサル攻撃のブロックを含む)。フィルターが指定されていない場合、RHEL ではデフォルトで、最も安全ですが制限が最も高い **'data'** フィルターが使用されます。さらに、アプリケーションが影響を受けると、Python は警告を発行します。

警告を非表示にする手順などの詳細は、ナレッジベースの記事 [Mitigation of directory traversal attack in the Python tarfile library \(CVE-2007-4559\)](#) を参照してください。

Jira:RHELDPCS-16405<sup>[1]</sup>

**HTTP::Tiny** Perl モジュールは、デフォルトで TLS 証明書を検証するようになる



HTTPS の使用時に TLS 証明書を検証するために、**HTTP::Tiny** Perl モジュールの **verify\_SSL** オプションのデフォルト値が **0** から **1** に変更されました。この変更により、**HTTP::Tiny** の [CVE-2023-31486](#) と CPAN Perl モジュールの [CVE-2023-31484](#) が修正されます。

TLS 検証のサポートを利用できるようにするために、この更新では次の依存関係を **perl-HTTP-Tiny** パッケージに追加します。

- **perl-IO-Socket-SSL**
- **perl-Mozilla-CA**
- **perl-Net-SSLeay**

[Bugzilla:2228412<sup>\[1\]</sup>](#)

## httpd がバージョン 2.4.57 にリベース

Apache HTTP サーバーはバージョン 2.4.57 に更新され、RHEL 9.1 以降で利用可能なバージョン 2.4.53 にバグ修正、機能拡張、セキュリティー修正が加えられています。

主な機能拡張は、次のとおりです。

- **httpd** で提供される **rotatelogs** ユーティリティーには、最初のログファイルを除くすべてのローテーションされたログファイルを切り詰めるための新しい **-T** オプションが導入されています。
- **mod\_ldap** モジュールの **LDAPConnectionPoolTTL** ディレクティブは、負の値を指定すると、接続された時期に関係なく接続を再利用できるようになりました。以前は、負の値はエラーとして処理されていました。
- **mod\_proxy\_hcheck** モジュールのワーカーが、ワーカーのタイムアウト設定に従って正しくタイムアウトするようになりました。
- **mod\_proxy\_hcheck** モジュールの **hcmethod** パラメーターは、HTTP/1.1 リクエストに新しい **GET11**、**HEAD11**、および **OPTIONS11** メソッドを提供するようになりました。

[Bugzilla:2184403](#)

## httpd の新しい mod\_authnz\_fcgi モジュール

Apache HTTP Server には **mod\_authnz\_fcgi** モジュールが含まれるようになりました。これにより、FastCGI オールライザーアプリケーションがユーザーを認証し、リソースへのアクセスを承認できるようになります。

**mod\_authnz\_fcgi** モジュールはデフォルトではロードされません。このモジュールをロードするには、**/etc/httpd/conf.modules.d/00-optional.conf** ファイル内の次の行のコメントを解除します。

```
LoadModule authnz_fcgi_module modules/mod_authnz_fcgi.so
```

[Bugzilla:2173295<sup>\[1\]</sup>](#)

## nginx:1.22 の新しい ssl\_pass\_phrase\_dialog ディレクティブ

**nginx:1.22** モジュールストリームのこの更新により、新しい **ssl\_pass\_phrase\_dialog** ディレクティブを使用して、暗号化された秘密キーごとに **nginx** の起動時に呼び出される外部プログラムを設定できるようになります。

新しいディレクティブを使用するには、次のいずれかの行を `/etc/nginx/nginx.conf` ファイルに追加します。

- 暗号化された秘密キーファイルごとに外部プログラムを呼び出すには、次のように入力します。

```
ssl_pass_phrase_dialog exec:<path_to_program>;
```

`nginx` は、次の 2 つの引数を使用してこのプログラムを呼び出します。

- **server\_name** 設定で指定されたサーバー名。
  - 次のアルゴリズムのいずれか: **RSA**、**DSA**、**EC**、**DH**、または **UNK** (暗号アルゴリズムが認識できない場合)。
- 暗号化された秘密キーファイルごとにパスフレーズを手動で入力する場合は、次のように入力します。

```
ssl_pass_phrase_dialog builtin;
```

これは、**ssl\_pass\_phrase\_dialog** が設定されていない場合のデフォルトの動作です。

この方法を使用しても、少なくとも 1 つの秘密キーがパスフレーズで保護されている場合、`nginx` サービスは起動に失敗することに注意してください。この場合は、他のいずれかの方法を使用してください。

- **systemctl** ユーティリティーを使用して `nginx` サービスを開始するときに、**systemd** で暗号化された秘密キーごとにパスフレーズの入力を求めるプロンプトを表示するには、次のように入力します。

```
ssl_pass_phrase_dialog exec:/usr/libexec/nginx-ssl-pass-dialog;
```

`nginx` の **ssl\_pass\_phrase\_dialog** ディレクティブは、Apache HTTP Server の **SSLPassPhraseDialog** ディレクティブに似ていることに注意してください。

[Bugzilla:2170808](#)

## 新しい rhel9/squid コンテナイメージ

Red Hat Container Registry で **rhel9/squid** コンテナイメージが利用できるようになりました。**Squid** は、Web クライアント向けの高パフォーマンスのプロキシキャッシュサーバーで、FTP、gopher、および HTTP のデータオブジェクトをサポートします。従来のキャッシュソフトウェアとは異なり、**Squid** はすべてのリクエストを単一の非ブロックの I/O 駆動プロセスで処理します。**Squid** は、メタデータ、特に RAM にキャッシュされたホットオブジェクトを保持し、DNS ルックアップをキャッシュし、ノンブロッキング DNS ルックアップをサポートして、失敗したリクエストのネガティブキャッシュを実装します。

新しいコンテナイメージをプルするには、次を実行します。

```
# podman pull registry.redhat.io/rhel9/squid
```

[Bugzilla:2178953](#)

## 新しいモジュールストリーム: redis:7

高度なキーバリューストアの **Redis 7** が、新しいモジュールストリーム **redis:7** として利用できるようになりました。

**Redis 6** と比較した場合の主な変更点は、以下のとおりです。

- Redis Functions API のサーバーサイドスクリプト
- きめ細かなアクセス制御リスト (ACL) のサポート
- クラスターの共有パブリッシュ/サブスクライブ (pub/sub) サポート
- さまざまな新しいコマンドとコマンド引数

**Redis 7** では、後方互換性のない変更がいくつか導入されています。次に例を示します。

- **Redis 7** は、追加専用ファイル (AOF) をフォルダー内の複数のファイルとして保存するようになりました。
- **Redis 7** は、以前のバージョンと互換性のない Redis Database (RDB) ファイルの新しいバージョン形式を使用しています。

機能と互換性のない変更の完全なリストは、[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。

**redis:7** モジュールストリームをインストールするには、以下を使用します。

```
# dnf module install redis:7
```

**redis** Application Streams 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。

[Bugzilla:2129826](#)

## 4.13. コンパイラーおよび開発ツール

### IBM Z での最適化されたルーチンの使用に影響を与える新しい **glibc** オプション

IBM Z アーキテクチャーでは、**glibc** ライブラリーは、**hwcaps** や **stfle** ビットなどのハードウェア機能に基づいて関数実装を選択します。この更新により、**glibc.cpu.hwcaps** 調整パラメーターを設定することで、ライブラリーによる選択を指示できるようになりました。

[Bugzilla:2169978<sup>\[1\]</sup>](#)

### **glibc** の Intel® Xeon® v5 ベースのハードウェアでの文字列およびメモリールーチンのパフォーマンスが向上

以前は、文字列およびメモリールーチン用に **glibc** によって使用されるデフォルトのキャッシュ量により、Intel® Xeon® v5 ベースのシステムで、パフォーマンスが予想よりも低下していました。今回の更新では、パフォーマンスを向上させるために使用するキャッシュの量が調整されました。

[Bugzilla:2213907](#)

### システム GCC コンパイラーがバージョン 11.4.1 に更新される

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

システム GCC コンパイラーがバージョン 11.4.1 に更新されました。これには、アップストリーム GCC で利用可能な多数のバグ修正と機能拡張が含まれています。

使用方法は、[RHEL 9 での C および C++ アプリケーションの開発](#) を参照してください。

[Bugzilla:2193180](#)

## GCC はレジスター引数の保持をサポートするようになる

今回の更新により、引数レジスターの内容をスタックに保存し、適切な Call Frame Information (CFI) を生成して、アンワインダーがパフォーマンスに悪影響を与えることなく CFI を特定できるようになりました。

[Bugzilla:2168204<sup>\[1\]</sup>](#)

## 64 ビット Intel アーキテクチャー上の GCC の新しい `-mdaz-ftz` オプション

64 ビット Intel アーキテクチャー上の GNU Compiler Collection (GCC) のシステムバージョンは、MXCSR Control と Status Register で flush-to-zero (FTZ) および denormals-are-zero (DAZ) フラグを有効にする `-mdaz-ftz` オプションをサポートするようになりました。

[Bugzilla:2208908](#)

## 新しい GCC Toolset 13

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。

GCC コンパイラーがバージョン 13.1.1 に更新され、アップストリームの GCC で利用できる多くのバグ修正と機能拡張が提供されています。

次のツールとバージョンが GCC Toolset 13 によって提供されます。

ツール	バージョン
GCC	13.1.1
GDB	12.1
binutils	2.40
dwz	0.14
annobin	12.20

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](#) および [GCC Toolset の使用](#) を参照してください。

Bugzilla:2171919<sup>[1]</sup>、Bugzilla:2171930

## GCC Toolset 13: GCC がバージョン 13.1.1 にリベース

GCC Toolset 13 では、GNU Compiler Collection (GCC) がバージョン 13.1.1 に更新されました。主な変更点は、以下のとおりです。

### 一般的な改善

- OpenMP:
  - OpenMP 5.0: Fortran は、いくつかの非長方形ループネストをサポートするようになりました。このようなサポートは、GCC 11 で C/C++ に追加されました。
  - 多くの OpenMP 5.1 機能が追加されました。
  - OpenMP 5.2 機能の初期サポートが追加されました。
- 新しいデバッグ情報圧縮オプション値 **-gz=zstd** が利用できるようになりました。
- **-Ofast**、**-ffast-math**、および **-funsafe-math-optimizations** オプションでは、**-shared** オプションを使用して共有オブジェクトを生成するときに、浮動小数点環境を変更するためのスタートアップコードが追加されなくなりました。
- GCC は、静的分析ツール (GCC の **-fanalyzer** など) の結果をキャプチャーするのに適した JSON ベース形式の、Static Analysis Results Interchange Format (SARIF) を使用して、診断を出力できるようになりました。SARIF を使用して、他の GCC 警告やエラーを機械可読形式でキャプチャーすることもできます。
- リンクタイム最適化の改善が実装されました。

### 新しい言語と言語固有の改善

#### C ファミリー:

- 新しい **-Wxor-used-as-pow** オプションは、ユーザーが指数を意図した可能性がある排他的論理和 (^) 演算子の使用について警告します。
- ファイル記述子である **int** 引数を文書化するために、3 つの新しい関数属性が追加されました。
  - **attribute((fd\_arg(N)))**
  - **attribute((fd\_arg\_read(N)))**
  - **attribute((fd\_arg\_write(N)))**

これらの属性は、ファイル記述子の誤用を検出するために **-fanalyzer** によっても使用されません。

- 新しいステートメント属性、**attribute((assume(EXPR)))**; が、C++23 移植性の前提のために追加されました。この属性は、C または以前の C++ でもサポートされています。

- GCC は、配列の要素にアクセスする目的で、構造体の最後の配列を柔軟な配列メンバーとして扱うタイミングを制御できるようになりました。デフォルトでは、集合体の最後の配列はすべて柔軟な配列メンバーとして扱われます。新しいコマンドラインオプション **-fstrict-flex-arrays** を使用して、どの配列メンバーを柔軟な配列として扱うかを制御します。

C:

- いくつかの C23 機能が実装されました。
  - **nullptr** 定数が導入されました。
  - 基礎となる型を指定するために列挙が強化されました。
  - 可変個引数パラメーターリストの要件が緩和されました。
  - オブジェクト定義の型推論を有効にする **auto** 機能が導入されました。
  - オブジェクト定義に **constexpr** 指定子が導入されました。
  - 複合リテラル用のストレージクラス指定子が導入されました。
  - **typeof** オブジェクト (以前はエクステンションとしてサポート) と **typeof\_unqual** オブジェクトが導入されました。
  - 新しいキーワード **alignas**、**alignof**、**bool**、**false**、**static\_assert**、**thread\_local**、および **true** を追加しました。
  - 関数が呼び出し元に実行を返さないことを指定する **[[noreturn]]** 属性を追加しました。
  - 空のイニシャライザー括弧のサポートが追加されました。
  - **STDC\_VERSION\_\*\_H** ヘッダーバージョンマクロのサポートが追加されました。
  - **ATOMIC\_VAR\_INIT** マクロを削除しました。
  - **<stddef.h>** ヘッダーに **unreachable** マクロを追加しました。
  - トライグラフを削除しました。
  - プロトタイプ化されていない関数を削除しました。
  - **%wN** および **%wfN** 形式の長さ修飾子の **-Wformat** オプションを使用した **printf** および **scanf** 形式のチェックを追加しました。
  - Unicode Standard Annex (UAX) 31 の識別子構文のサポートが追加されました。
  - C23 で採用されている既存の機能は、C23 要件に従うように調整されており、**-std=c2x -Wpedantic** オプションを使用して診断されることはありません。
- 新しい **-Wenum-int-mismatch** オプションは、列挙型と整数型の間の一貫性について警告します。

C++:

- **-fexcess-precision** オプションを使用して、excess precision サポートを実装しました。これは、**-std=c++17** などの厳密な標準モードではデフォルトで有効化されています。デフォルトは **-fexcess-precision=standard** になります。**-std=gnu++20** などの GNU 標準モードでは、デフォルトは **-fexcess-precision=fast** となり、以前の動作が復元されます。

- fexcess-precision** オプションは、次のアーキテクチャーに影響します。
- x87 演算を使用する Intel 32 および 64 ビット、場合によっては Motorola 68000 では、**float** と **double** 式が **long double** 精度で評価されます。
  - 64 ビット IBM Z システムでは、**float** 式が **double** 精度で評価されます。
  - **std::float16\_t** または **std::bfloat16\_t** 型をサポートするいくつかのアーキテクチャーでは、これらの型は **float** 精度で評価されます。
- 以下を含む C++23 の実験的サポートが改善されました。
    - 複合ステートメントの末尾にラベルのサポートが追加されました。
    - 一時オブジェクトへの参照バインドイングを検出するための型特性を追加しました。
    - 揮発性複合操作のサポートが再導入されました。
    - **#warning** ディレクティブのサポートが追加されました。
    - 区切られたエスケープシーケンスのサポートが追加されました。
    - 名前付きユニバーサル文字エスケープのサポートが追加されました。
    - **char8\_t** タイプの互換性と移植性に関する修正を追加しました。
    - 静的 **operator()** 関数オブジェクトを追加しました。
    - 暗黙的なムーブを簡略化しました。
    - 式内の等価性の書き換えは、以前より互換性を破壊しにくくなりました。
    - エンコード不可能なワイド文字リテラルとワイド複数文字リテラルを削除しました。
    - **constexpr** 関数の制限をいくつか緩和しました。
    - 拡張浮動小数点型と標準名。
    - 移植可能な前提条件を実装しました。
    - 移植可能なソースファイルのエンコード標準として UTF-8 のサポートを追加しました。
    - 静的 **operator[]** の添字演算子の多次元サポートを追加しました。
  - 新しい警告:
    - **-Wself-move** は、**std::move** を使用して値がそれ自体に移動されると警告します。
    - **-Wdangling-reference** は、有効期間が終了した一時オブジェクトに参照がバインドされている場合に警告します。
    - **-Wpessimizing-move** および **-Wredundant-move** 警告は、より多くのコンテキストで警告するように拡張されました。
  - 新しい **-nostdlib++** オプションを使用すると、C++ 標準ライブラリーで暗黙的にリンクせずに、**g++** とのリンクが可能になります。

## libstdc++ ランタイムライブラリーの変更

- 以下を含む C++20 の実験的サポートが改善されました。
  - `<format>` ヘッダーと `std::format` 関数を追加しました。
  - `<chrono>` ヘッダーに、`std::chrono::utc_clock` クロック、その他のクロック、タイムゾーン、および `std::format` 関数のサポートが追加されました。
- 以下を含む C++23 の実験的サポートが改善されました。
  - `<ranges>` ヘッダーへの追加:  
`views::zip`、`views::zip_transform`、`views::adjacent`、`views::adjacent_transform`、`views::pairwise`、`views::slide`、`views::chunk`、`views::chunk_by`、`views::repeat`、`views::chunk_by`、`views::cartesian_product`、`views::as_rvalue`、`views::enumerate`、`views::as_const`
  - `<algorithm>` ヘッダーへの追加:  
`ranges::contains`、`ranges::contains_subrange`、`ranges::iota`、`ranges::find_last`、`ranges::find_last_if`、`ranges::find_last_if_not`、`ranges::fold_left`、`ranges::fold_left_first`、`ranges::fold_right`、`ranges::fold_right_last`、`ranges::fold_left_with_iter`、`ranges::fold_left_first_with_iter`
  - `std::expected` クラステンプレートのモノイド操作のサポート。
  - `constexpr` 修飾子を `std::bitset`、`std::to_chars`、および `std::from_chars` 関数に追加しました。
  - 拡張浮動小数点型のライブラリーサポートを追加しました。
- Library Fundamentals Technical Specification (TS) のバージョン 3 から `<experimental/scope>` ヘッダーのサポートが追加されました。
- Concurrency TS のバージョン 2 から `<experimental/synchronized_value>` ヘッダーのサポートが追加されました。
- フリースタANDING環境で以前は利用できなかった多くの機能のサポートが追加されました。以下に例を示します。
  - `std::tuple` クラステンプレートをフリースタANDING環境でのコンパイルに使用できるようになりました。
  - `libstdc++` ライブラリーは、`std::array` や `std::string_view` などのコンポーネントをフリースタANDING環境のサブセットに追加します。
  - `libstdc++` ライブラリーは `-ffreestanding` コンパイラーオプションを尊重するようになったため、`libstdc++` ライブラリーのフリースタANDING環境でのインストールを別途ビルドする必要がなくなりました。`-ffreestanding` を指定してコンパイルすると、`libstdc++` ライブラリーが完全なホスト型実装としてビルドされた場合でも、フリースタANDING環境のサブセットで利用可能な機能が制限されます。

## 新しいターゲットとターゲット固有の改善

### 64 ビット ARM アーキテクチャー

- `-march=` オプションの `armv9.1-a`、`armv9.2-a`、および `armv9.3-a` 引数のサポートが追加されました。

32 ビットおよび 64 ビットの AMD および Intel アーキテクチャー:



- C と C++ の両方の場合、`__bf16` 型は、Streaming SIMD Extensions 2 以降が有効になっているシステムでサポートされます。
- 実際の `__bf16` 型が `AVX512BF16` 命令組み込みに使用されるようになりました。以前は、`short` の typedef である `__bfloat16` が使用されていました。GCC 12 を GCC 13 にアップグレードするときに、`AVX512BF16` 関連のソースコードを調整してください。
- 次の Intel 命令をサポートするための新しい Instruction Set Architecture (ISA) エクステンションが追加されました。
  - `AVX-IFMA` の命令組み込みは、`-mavxifma` コンパイラスイッチを通じて使用できます。
  - `AVX-VNNI-INT8` の命令組み込みは、`-mavxvnniint8` コンパイラスイッチを通じて使用できます。
  - `AVX-NE-CONVERT` の命令組み込みは、`-mavxneconvert` コンパイラスイッチを通じて使用できます。
  - `CMPccXADD` の命令組み込みは、`-mcmpccxadd` コンパイラスイッチを通じて使用できます。
  - `AMX-FP16` の命令組み込みは、`-mamx-fp16` コンパイラスイッチを通じて使用できます。
  - `PREFETCHI` の命令組み込みは、`-mprefetchi` コンパイラスイッチを通じて使用できます。
  - `RAO-INT` の命令組み込みは、`-mraoint` コンパイラスイッチを通じて使用できます。
  - `AMX-COMPLEX` の命令組み込みは、`-mamx-complex` コンパイラスイッチを通じて使用できます。
- GCC は、`-march=znver4` コンパイラスイッチを通じて、`znver4` コアに基づく AMD CPU をサポートするようになりました。このスイッチにより、GCC は自動ベクトル化時に 512 ビットベクトルの使用を考慮します。

### 静的アナライザーの改善

- 静的アナライザーは 20 件の新しい警告を取得しました。
  - `-Wanalyzer-allocation-size`
  - `-Wanalyzer-deref-before-check`
  - `-Wanalyzer-exposure-through-uninit-copy`
  - `-Wanalyzer-imprecise-fp-arithmetic`
  - `-Wanalyzer-infinite-recursion`
  - `-Wanalyzer-jump-through-null`
  - `-Wanalyzer-out-of-bounds`
  - `-Wanalyzer-putenv-of-auto-var`
  - `-Wanalyzer-tainted-assertion`

- ファイル記述子の誤用に関する 7 つの新しい警告:
  - **-Wanalyzer-fd-access-mode-mismatch**
  - **-Wanalyzer-fd-double-close**
  - **-Wanalyzer-fd-leak**
  - **-Wanalyzer-fd-phase-mismatch** (例: ソケットで **listen** を呼び出す前のソケットでの **accept** の呼び出し)
  - **-Wanalyzer-fd-type-mismatch** (例: データグラムソケットでのストリームソケット操作の使用)
  - **-Wanalyzer-fd-use-after-close**
  - **-Wanalyzer-fd-use-without-check**
    - また、**open**、**close**、**creat**、**dup**、**dup2**、**dup3**、**pipe**、**pipe2**、**read**、および **write** 関数の動作の特殊ケーシングの処理も実装されました。
- **<stdarg.h>** ヘッダーの誤用に関する 4 つの新しい警告:
  - **-Wanalyzer-va-list-leak** は、**va\_start** マクロまたは **va\_copy** マクロの後に **va\_end** マクロがないことを警告します。
  - **-Wanalyzer-va-list-use-after-va-end** は、**va\_end** マクロが呼び出された **va\_list** オブジェクトタイプで使用されている **va\_arg** または **va\_copy** マクロについて警告します。
  - **-Wanalyzer-va-arg-type-mismatch** タイプは、プロシージャー間の実行パスでの **va\_arg** マクロの使用状況を、実際に可変個引数呼び出しに渡されたパラメーターの型と照合してチェックします。
  - **-Wanalyzer-va-list-exhausted** は、プロシージャー間実行パス内の **va\_list** オブジェクトタイプで **va\_arg** マクロが何度も使用されている場合に警告します。

- その他多数の改善。

## 後方互換性のない変更

C++ の場合、**std::cout**、**std::cin** などのグローバル **iostream** オブジェクトの構築は、**<iostream>** ヘッダーを含むすべてのソースファイルではなく、標準ライブラリー内で行われるようになりました。この変更により、C++ プログラムの起動パフォーマンスが向上しますが、ランタイムに正しいバージョンの **libstdc++.so** が使用されない場合、GCC 13.1 でコンパイルされたコードがクラッシュすることになります。ランタイムに正しい **libstdc++.so** を使用方法の詳細は、[ドキュメント](#) を参照してください。将来の GCC リリースではこの問題が軽減され、互換性のない以前の **libstdc++.so** ではプログラムをまったく実行できなくなります。

Bugzilla:2172093<sup>[1]</sup>

## GCC Toolset 13: **annobin** がバージョン 12.20 にリベース

GCC Toolset 13 は、**annobin** パッケージのバージョン 12.20 を提供します。主な機能拡張は、次のとおりです。

- **annobin** ノートを別のデバッグ情報ファイルに移動するためのサポートが追加されました。これにより、実行可能バイナリーのサイズが縮小されます。

- 新しい小さな note フォーマットのサポートが追加されたことで、個別の debuginfo ファイルのサイズが縮小され、これらのファイルの作成にかかった時間が短縮されました。

Bugzilla:2171923<sup>[1]</sup>

## GCC Toolset 13: GDB がバージョン 12.1 にリベース

GCC Toolset 13 は、GDB バージョン 12.1 を提供します。

主なバグ修正と機能拡張は、以下のとおりです。

- GDB では、デフォルトでソースコードと逆アセンブラーのスタイルが設定されるようになりました。スタイリングが GDB の自動化またはスクリプト作成を妨げる場合は、**maint set gnu-source-highlight enabled off** および **maint set style disassembler enabled off** コマンドを使用してスタイル設定を無効にできます。
- GDB は、内部エラーが発生するたびにバックトレースを表示するようになりました。これがスクリプトまたは自動化に影響する場合は、**maint set backtrace-on-fatal-signal off** コマンドを使用して、この機能を無効にすることができます。

C/C++ の改善:

- GDB は、C++ テンプレートに関する関数または型を関数のオーバーロードと同様に扱うようになりました。パラメーターリストを省略して、複数のテンプレートタイプで構成される型や関数などのテンプレート関数のファミリーにブレークポイントを設定できます。タブ 補完にも同様の改善が加えられました。

ターミナルユーザーインターフェイス (TUI):

- **tui layout**  
**tui focus**

**tui refresh**

**tui window height**

これらは、古い **layout**、**focus**、**refresh**、および **winheight** TUI コマンドの新しい名前です。古い名前は、これらの新しいコマンドのエイリアスとしてまだ存在しています。

- **tui window width**  
**winwidth**

ウィンドウが水平モードでレイアウトされている場合、新しい **tui window width** コマンドまたは **winwidth** エイリアスを使用して、TUI ウィンドウの幅を調整します。

- **info win**  
このコマンドの出力には、TUI ウィンドウの幅に関する情報が含まれるようになりました。

Machine Interface (MI) の変更:

- MI インタープリターのデフォルトバージョンが 4 (`-i=mi4`) になりました。
- フラグのない **-add-inferior** コマンドは、現在の下位コマンドの接続を継承するようになりました。これにより、バージョン 10 より前の GDB の動作が復元されます。
- **-add-inferior** コマンドは、接続なしで新しい下位コマンドを起動させる **--no-connection** フラグを受け入れるようになりました。

- ブレークポイント出力の **script** フィールド (MI 3 以前では構文的に不正確) は、MI 4 ではリストになりました。これは次のコマンドとイベントに影響します。
  - **-break-insert**
  - **-break-info**
  - **=breakpoint-created**
  - **=breakpoint-modified**  
以前の MI バージョンでの新しい動作を有効にするには、**-fix-breakpoint-script-output** コマンドを使用します。

新しいコマンド:

- **maint set internal-error backtrace [on|off]**  
**maint show internal-error backtrace**
- maint set internal-warning backtrace [on|off]**
- maint show internal-warning backtrace**

GDB は、内部エラーまたは内部警告が発生したときに、自身のバックトレースを出力できるようになりました。これは、内部エラーの場合はデフォルトで有効化され、内部警告の場合はデフォルトで無効化されています。

- **exit**  
既存の **quit** コマンドに加えて、新しい **exit** コマンドを使用して GDB を終了できます。
- **maint set gnu-source-highlight enabled [on|off]**  
**maint show gnu-source-highlight enabled**  
ソースコードにスタイリングを追加するために、GNU Source Highlight ライブラリーを有効または無効化します。無効にすると、ライブラリーは利用可能であっても使用されません。GNU Source Highlight ライブラリーが使用されない場合は、代わりに Python Pygments ライブラリーが使用されます。
- **set suppress-cli-notifications [on|off]**  
**show suppress-cli-notifications**  
CLI に対して通知の出力を抑制するかどうかを制御します。CLI 通知は、選択したコンテキスト (現在の下位、スレッド、フレームなど) を変更したとき、またはデバッグ中のプログラムが停止したときに (ブレークポイントに到達した、ソースステップングの完了、または割り込みなどにより) 発生します。
- **set style disassembler enabled [on|off]**  
**show style disassembler enabled**  
有効化すると、GDB が Python サポートでコンパイルされており、Python Pygments パッケージが利用可能な場合、コマンドは逆アセンブラー出力にスタイリングを適用します。

変更したコマンド:

- **set logging [on|off]**  
非推奨となり、**set logging enabled [on|off]** コマンドに置き換えられました。
- **print**

/x などの基数変更形式を使用した浮動小数点値の出力が、値の基礎となるバイトを必要な基数で表示するように変更されました。

- **clone-inferior**

**clone-inferior** コマンドにより、**TTY**、**CMD**、および **ARGs** の設定が、元の下位設定から新しい設定にコピーされるようになりました。**set environment** コマンドまたは **unset environment** コマンドを使用して加えられた環境変数へのすべての変更も、新しい下位環境変数にコピーされます。

Python API:

- 新しい **gdb.add\_history()** 関数は、**gdb.Value** オブジェクトを取り、それが表す値を GDB の履歴リストに追加します。この関数は、履歴リスト内の新しい項目のインデックスである整数を返します。
- 新しい **gdb.history\_count()** 関数は、GDB の値履歴内の値の数を返します。
- 新しい **gdb.events.gdb\_exiting** イベントは、GDB 終了コードの値を含む読み取り専用属性 **exit\_code** を持つ **gdb.GdbExitingEvent** オブジェクトを使用して呼び出されます。このイベントは、GDB が終了する前の、GDB が内部状態のクリーンアップを開始する前にトリガーされます。
- 新しい **gdb.architecture\_names()** 関数は、考えられるすべての **Architecture.name()** 値を含むリストを返します。各エントリは文字列です。
- 新しい **gdb.Architecture.integer\_type()** 関数は、サイズと符号付きかどうかを指定して整数型を返します。
- 新しい **gdb.TargetConnection** オブジェクトタイプは、(**info connection** コマンドで表示される) 接続を表します。サブクラス **gdb.RemoteTargetConnection** は、**remote** 接続および **extended-remote** 接続を表します。
- **gdb.Inferior** 型には、**gdb.TargetConnection** オブジェクトのインスタンスである **connection** プロパティがあり、この下位型によって使用される接続になります。下層に接続がない場合、これは **None** にすることができます。
- 新しい **gdb.events.connection\_removed** イベントレジストリーは、接続が GDB から削除されると **gdb.ConnectionEvent** イベントを発行します。このイベントには、**connection** プロパティ、つまり削除される接続の **gdb.TargetConnection** オブジェクトがあります。
- 新しい **gdb.connections()** 関数は、現在アクティブなすべての接続のリストを返します。
- 新しい **gdb.RemoteTargetConnection.send\_packet(PACKET)** メソッドは、既存の **maint packet** CLI コマンドと同等です。これを使用して、指定したパケットをリモートターゲットに送信できます。
- 新しい **gdb.host\_charset()** 関数は、現在のホスト文字セットの名前を文字列として返します。
- 新しい **gdb.set\_parameter(NAME, VALUE)** 関数は、GDB パラメーター **NAME** を **VALUE** に設定します。
- 新しい **gdb.with\_parameter (NAME, VALUE)** 関数は、GDB パラメーター **NAME** を一時的に **VALUE** に設定してから、コンテキストが終了するとリセットするコンテキストマネージャーを返します。
- **gdb.Value.format\_string** メソッドは、ブール値である **styling** 引数を取るようになりました。**true** の場合、返される文字列には、スタイリングを適用するためのエスケープシーケンス

を含めることができます。スタイリングは、GDB でスタイリングがオンになっている場合にのみ存在します (**help set styling** を参照してください)。**false** の場合 (**スタイル** 引数が指定されていない場合のデフォルト)、返される文字列にはスタイリングは適用されません。

- 新しい読み取り専用属性 **gdb.InferiorThread.details** は、追加のターゲット固有のスレッド状態情報を含む文字列か、そのような追加情報がない場合は **None** になります。
- 新しい読み取り専用属性 **gdb.Type.is\_scalar** は、スカラー型の場合は **True**、その他すべての型の場合は **False** です。
- 新しい読み取り専用属性 **gdb.Type.is\_signed** は、**Type.is\_scalar** が **True** の場合にのみ読み取られる必要があり、符号付き型の場合は **True**、その他すべての型の場合は **False** になります。非スカラー型に対してこの属性を読み取ろうとすると、**ValueError** が発生します。
- Python で実装された GDB および MI コマンドを追加できるようになりました。

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。

[What has changed in GDB?](#)

Bugzilla:2172096<sup>[1]</sup>

## GCC Toolset 13: binutils がバージョン 2.40 にリベース

GCC Toolset 13 は、**binutils** パッケージバージョン 2.40 を提供します。主な機能拡張は、次のとおりです。

リンカー:

- リンカーの新しい **-w (--no-warnings)** コマンドラインオプションは、警告メッセージやエラーメッセージの生成を抑制します。これは、動作しないことがわかっているバイナリーを作成する必要がある場合に役立ちます。
- ELF リンカーは、次の場合に警告メッセージを生成するようになりました。
  - スタックが実行可能な場合。
  - **Read**、**Write**、**eXecute** の 3 つのパーミッションセットをすべて備えたメモリ常駐セグメントを作成する場合。
  - **eXecute** パーミッションセットを使用してスレッドローカルデータセグメントを作成する場合。  
**--no-warn-exec-stack** または **--no-warn-rwx-segments** オプションを使用すると、これらの警告を無効にできます。
- リンカーは、作成するバイナリーに任意の JSON 形式のメタデータを挿入できるようになりました。

その他のツール:

- Portable Executable (PE) 形式ファイルのファイルヘッダーとセクションヘッダーのフィールドを表示する、**objdump** ツールの新しい **--private** オプション。
- ELF ファイルから ELF セクションヘッダーを削除するための、**objcopy** ユーティリティーおよび **strip** ユーティリティー用の新しい **--strip-section-headers** コマンドラインオプション。

- **objdump** ユーティリティーの新しい **--show-all-symbols** コマンドラインオプションは、アドレスに一致する最初のシンボルのみを表示するデフォルト機能とは対照的に、逆アセンブル時に指定されたアドレスに一致するすべてのシンボルを表示します。
- **nm** ユーティリティーに新しい **-W (--no-weak)** オプションを追加し、弱いシンボルを無視します。
- **objdump** ユーティリティーは、一部のアーキテクチャーの逆アセンブラー出力の構文強調表示をサポートするようになりました。 **--disassembler-color=MODE** コマンドラインオプションを使用します。 **MODE** は、次のいずれかになります。
  - **off**
  - **color** - このオプションは、すべてのターミナルエミュレーターでサポートされています。
  - **extended-color** - このオプションは、すべてのターミナルエミュレーターでサポートされていない 8 ビットカラーを使用します。

[Bugzilla:2171926<sup>\[1\]</sup>](#)

## libabigail がバージョン 2.3 にリベース

**libabigail** パッケージがバージョン 2.3 に更新されました。以下は、主な改善点です。

- BTF debuginfo 形式がサポートされるようになりました。
- Ada 範囲タイプのサポートが改善されました。
- 抑制仕様の新しい **[allow\_type]** ディレクティブがサポートされるようになりました。
- **[supress\_type]** 抑制仕様にさまざまな新しいプロパティを追加しました。
- ABIXML ファイル形式がバージョン 2.2 に更新されました。
- ライブラリーの SONAME は、独自の ABI 変更を反映するために変更されました。

**libabigail** パッケージは、CodeReady Linux Builder (CRB) リポジトリで入手できます。CodeReady Linux Builder リポジトリに含まれるパッケージは、サポート対象外であることに注意してください。

[Bugzilla:2186931](#)

## debugedit の find-debuginfo スクリプトが、-q (--quiet) フラグをサポートするようになる

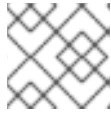
この更新により、**debugedit** ユーティリティーで **find-debuginfo** スクリプトの **-q (--quiet)** フラグを使用して、スクリプトからのエラー以外の出力をサイレンスにすることができます。

[Bugzilla:2177302](#)

## Valgrind がバージョン 3.21.0 にリベース

**Valgrind** がバージョン 3.21.0 に更新されました。主な機能拡張は、次のとおりです。

- **--vgdb-stop-at=event1,event2,...** オプションの新しい **abexit** 値は、ゼロ以外の終了コードなどでプログラムが異常終了したときに **gdbserver** ユーティリティーに通知します。
- 新しい **--enable-debuginfod=[yes|no]** オプションは、**DEBUGINFOD\_URLS** 環境変数にリストされている **debuginfod** サーバーを使用して、Valgrind で実行されているプログラムに不足している DWARF debuginfo 情報を取得するように Valgrind に指示します。このオプションのデフォルト値は **yes** です。



## 注記

**DEBUGINFOD\_URLS** 環境変数は、デフォルトでは設定されません。

- Valgrind は GDB Python コマンドを提供するようになりました。これらの GDB フロントエンドコマンドにより、GDB コマンドラインインターフェイスでの統合が強化されます。この利点としては、GDB のオートコンプリートや、正規表現に一致するコマンドまたはコマンドヘルプを検索するコマンド固有のヘルプなどが挙げられます。関連する監視コマンドについては、GDB は引数を評価して監視コマンドの使用を簡素化します。
- **vgdb** ユーティリティーは、**--multi** オプションを指定して呼び出された場合に、拡張リモートプロトコルをサポートするようになりました。このモードでは GDB **run** コマンドがサポートされているため、1つのターミナルから GDB と Valgrind を実行できます。
- **--realloc-zero-bytes-frees=[yes|no]** オプションを使用すると、**malloc()** 呼び出しをインターセプトするツールのサイズ 0 の **realloc()** 関数の動作を変更できます。
- **memcheck** ツールは、サイズ 0 の **realloc()** 関数の使用のチェックを実行するようになりました。この機能を無効にするには、新しい **--show-realloc-size-zero=[yes|no]** スイッチを使用します。
- **helgrind** ツールの新しい **--history-backtrace-size=value** オプションを使用して、以前のアクセスのスタックトレースに記録するエントリーの数を設定できます。
- **--cache-sim=[yes|no] cachegrind** オプションはデフォルトで **no** になり、その結果、デフォルトでは命令キャッシュ読み取りイベントのみが収集されるようになりました。
- **cg\_annotate**、**cg\_diff**、および **cg\_merge cachegrind** ユーティリティーのソースコードが書き直されたため、ユーティリティーのコマンドラインオプションの処理がより柔軟になりました。たとえば、**--show-percs** と **--no-show-percs** オプション、および既存の **--show-percs=yes** と **--show-percs=no** オプションがサポートされるようになりました。
- **cg\_annotate cachegrind** ユーティリティーは、差分 (**--diff**、**--mod-filename**、および **--mod-funcname** オプションを使用) とマージ (複数のデータファイルを渡すことによる) をサポートするようになりました。さらに、**cg\_annotate** は、ファイルおよび関数レベルでより多くの情報を提供するようになりました。
- **DHAT** ツールに対する新しいユーザー要求により、メモリーブロックのアクセスカウントヒストグラムの 1024 バイト制限をオーバーライドできるようになります。

次の新しいアーキテクチャー固有の命令セットがサポートされるようになりました。

- 64 ット ARM:
  - v8.2 スカラーおよびベクトルの Floating-point Absolute Difference (FABD)、Floating-point Absolute Compare Greater than or Equal (FACGE)、Floating-point Absolute Compare Greater Than (FACGT)、および Floating-point Add (FADD) 命令。
  - v8.2 Floating-point (FP) 比較命令と条件付き比較命令。
  - v8.2 Floating-point (FP) 比較命令のゼロバリエーション。
- 64-ビット IBM Z:
  - **miscellaneous-instruction-extensions facility 3** および **vector-enhancements facility 2** のサポート。これにより、**-march=arch13** または **-march=z15** オプションを使用してコンパイルされたプログラムを Valgrind で実行できるようになります。



- IBM Power:
  - ISA 3.1 のサポートが完了しました。
  - ISA 3.0 は、deliver a random number (darn) 命令をサポートするようになりました。
  - ISA 3.0 は、System Call Vectored (scv) 命令をサポートするようになりました。
  - ISA 3.0 は、コピー、貼り付け、および cpabort 命令をサポートするようになりました。

[Bugzilla:2124346](#)

### systemtap がバージョン 4.9 にリベース

**systemtap** パッケージがバージョン 4.9 にアップグレードされました。主な変更点は、以下のとおりです。

- 新しい Language-Server-Protocol (LSP) バックエンドにより、LSP 対応エディターでの **systemtap** スクリプトのインタラクティブなドラフト作成が容易になります。
- Python/Jupyter インタラクティブノートブックフロントエンドへのアクセス。
- DWARF 5 ビットフィールドの処理が改善されました。

[Bugzilla:2186934](#)

### elfutils がバージョン 0.189 にリベース

**elfutils** パッケージがバージョン 0.189 に更新されました。主な改善点とバグ修正は次のとおりです。

#### libelf

**elf\_compress** ツールは、**ELFCOMPRESS\_ZSTD** ELF 圧縮タイプをサポートするようになりました。

#### libdwfl

**dwfl\_module\_return\_value\_location** 関数は、**DW\_TAG\_unspecified\_type** タイプのタグを指す DWARF Information Entries (DIEs) に対して 0 (戻り値のタイプなし) を返すようになりました。

#### eu-elfcompress

**-t** および **--type=** オプションは、**zstd** 引数を介して Zstandard (**zstd**) 圧縮形式をサポートするようになりました。

[Bugzilla:2182061](#)

### libpfm がバージョン 4.13 にリベース

**libpfm** パッケージがバージョン 4.13 に更新されました。この更新により、**libpfm** は、次のプロセッサマイクロアーキテクチャーのパフォーマンス監視ハードウェアネイティブイベントにアクセスできるようになります。

- AMD Zen 2
- AMD Zen 3
- AMD Zen 4
- ARM Neoverse N1
- ARM Neoverse N2

- ARM Neoverse V1
- ARM Neoverse V2
- IBM z16
- 第 4 世代 Intel® Xeon® スケーラブルプロセッサ

[Bugzilla:2185652](#)、[Bugzilla:2047720](#)、[Bugzilla:2111940](#)、[Bugzilla:2111924](#)、[Bugzilla:2111930](#)、[Bugzilla:2111933](#)、[Bugzilla:2111957](#)、[Bugzilla:2111946](#)

### 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 2
- AMD Zen 3
- ARM Neoverse N1
- ARM Neoverse N2
- ARM Neoverse V1
- ARM Neoverse V2

[Bugzilla:2111923<sup>\[1\]</sup>](#)、[Bugzilla:2111947](#)、[Bugzilla:2111942](#)

### papi は、64 ビット ARM プロセッサの高速パフォーマンスイベントカウント読み取り操作をサポートするようになる

以前の 64 ビット ARM プロセッサでは、すべてのパフォーマンスイベントカウンターの読み取り操作で、リソースを大量に消費するシステムコールを使用する必要がありました。**papi** は、64 ビット ARM 用に更新され、パフォーマンスカウンターで自身を監視するプロセスが、パフォーマンスイベントカウンターのより高速なユーザー空間読み取りを使用できるようになりました。`/proc/sys/kernel/perf_user_access` パラメーターを 1 に設定すると、**papi** が 2 つのカウンターを読み取るための平均クロックサイクル数が 724 サイクルから 29 サイクルに減少します。

[Bugzilla:2186927<sup>\[1\]</sup>](#)

### LLVM Toolset がバージョン 16.0.6 にリベース

LLVM Toolset がバージョン 16.0.6 に更新されました。

主な機能拡張は、次のとおりです。

- 最適化の改善
- 新しい CPU 拡張のサポート
- 新しい C++ バージョンへのサポートの改善

後方互換性のない注目すべき変更は次のとおりです。

- Clang のデフォルトの C++ 標準は、**gnu++14** ではなく **gnu++17** になりました。

- **-Wimplicit-function-declaration**、**-Wimplicit-int**、および **-Wincomplete-function-pointer-types** オプションは、C コードのデフォルトでエラーになるようになりました。これは、configure スクリプトの動作に影響を与える可能性があります。

デフォルトでは、Clang 16 は、GCC Toolset 13 によって提供される **libstdc++** ライブラリーバージョン 13 と **binutils 2.40** を使用します。

詳細は、[LLVM リリースノート](#) および [Clang リリースノート](#) を参照してください。

[Bugzilla:2178796](#)

## Rust Toolset がバージョン 1.71.1 にリベース

Rust Toolset がバージョン 1.71.1 に更新されました。主な変更点は、以下のとおりです。

- パフォーマンスを向上させるための multiple producer, single consumer (mpsc) チャネルの新しい実装
- **crates.io** レジストリーをより効率的に使用するための新しい Cargo **sparse** インデックスプロトコル
- ワンタイム値の初期化のための新しい **OnceCell** および **OnceLock** タイプ
- Foreign Function Interface (FFI) 境界にまたがる強制アンワインドの使用を可能にする新しい **C-unwind** ABI 文字列

詳細は、一連のアップストリームリリース発表を参照してください。

- [Announcing Rust 1.67.0](#)
- [Announcing Rust 1.68.0](#)
- [Announcing Rust 1.69.0](#)
- [Announcing Rust 1.70.0](#)
- [Announcing Rust 1.71.0](#)

[Bugzilla:2191743](#)

## Rust profiler\_builtins ランタイムコンポーネントが利用可能になる

この機能拡張により、Rust **profile\_builtins** ランタイムコンポーネントが利用できるようになりました。このランタイムコンポーネントは、次のコンパイラーオプションを有効にします。

### **-C instrument-coverage**

カバレッジプロファイリングを有効にします。

### **-C profile-generate**

プロファイルガイド付きの最適化を有効にします。

[Bugzilla:2227082<sup>\[1\]</sup>](#)

## Go Toolset がバージョン 1.20.10 にリベース

Go Toolset がバージョン 1.20.10 に更新されました。

主な機能拡張は、次のとおりです。

- 内部表現に依存せずにスライスと文字列を処理するための新しい関数が **unsafe** パッケージに追加されました。
- 比較可能な型は、比較可能な制約を満たすことができるようになりました。
- 新しい **crypto/ecdh** パッケージ。
- **go build** および **go test** コマンドは、**-i** フラグを受け入れなくなりました。
- **go generate** および **go test** コマンドは、**-skip pattern** オプションを受け入れるようになりました。
- **go build**、**go install**、およびその他のビルド関連コマンドは、**-pgo** フラグと **-cover** フラグをサポートするようになりました。
- **go** コマンドは、C ツールチェーンのないシステムではデフォルトで **cgo** を無効にするようになりました。
- **go version -m** コマンドは、より多くの Go バイナリータイプの読み取りをサポートするようになりました。
- **go** コマンドは、C ツールチェーンのないシステムではデフォルトで **cgo** を無効にするようになりました。
- コードカバレッジプロファイルを単体テストからのみ収集するのではなく、アプリケーションと統合テストから収集するためのサポートが追加されました。

Bugzilla:2185259<sup>[1]</sup>

## pcp がバージョン 6.0.5 にリベース

**pcp** パッケージがバージョン 6.0.5 に更新されました。主な変更点は、以下のとおりです。

### コレクターツールの機能

- **pmdaproc** -
  - 最新のカーネルで cgroup ごとの IRQ PSI メトリクスのサポートが追加されました。
  - 新しい **proc.smaps.pss\_dirty** メトリックが追加されました。
- **pmdasmart**: NVME ディスク情報と電源状態メトリクスが追加されました。
- **pmdalinux**:
  - 最新のカーネルでシステム全体の IRQ PSI メトリクスのサポートが追加されました。
  - NUMA 外部メモリー断片化メトリックが追加されました。
  - 新しいネットワーキング (TCP、ICMP) メトリクスが追加されました。
- **pmdaoverhead**: プロセスのグループのオーバーヘッドを測定する新しい PMDA
- **pmdahacluster**: Pacemaker 2.1.5 **crm\_mon** 出力の変更を処理するために更新されました。

### 監視ツールの機能

- **pmieconf**:

- Webhook アクションのサポートが追加されました (Event Driven Ansible)。
- ファイル記述子の制限をチェックする新しい **pmie** ルールが追加されました。
- **pcp2json**: HTTP POST を送信するオプションを備えた拡張 **pcp2json**
- **pcp-atop**: **cgroup**、NUMA メモリー、および NUMA CPU サポートが追加されました。
- **pcp-htop**: 新しいオープンファイル記述子 Meter のサポートが追加されました。
- **pcp-ps**: 複数のアーカイブサンプルを表示する機能が追加されました。

[Bugzilla:2175602](#)

### PCP の **pmie** ユーティリティーが Webhook イベントの生成をサポートするようになる

Performance Co-Pilot (PCP) の Performance Metrics Inference Engine (**pmie**) ユーティリティーは、Webhook イベントの生成をサポートするようになりました。この更新により、設定された **pmie** ルールは、Event-Driven Ansible (EDA) によって使用可能な形式でイベントを生成します。その結果、EDA は PCP ルールに対応できます。

この機能を有効にするには、すべてのローカル **pmie** ルールを、特定のエンドポイント (URL) で Webhook に送信するように設定します。

```
# pmieconf modify global webhook_endpoint https://localhost:443/<endpoint>
# pmieconf modify global webhook_action yes
```

[Bugzilla:2185803](#)

### **grafana** がバージョン 9.2.10 にリベース

**grafana** パッケージがバージョン 9.2.10 に更新されました。主な変更点は、以下のとおりです。

- heatmap パネルは Grafana 全体で使用されるようになりました。
- Geomaps は距離と面積の両方を測定できるようになりました。
- Alertmanager は、**Prometheus Alertmanager** バージョン 0.24 に基づくようになりました。
- Grafana Alerting ルールは、実行エラーまたはタイムアウト時にデフォルトで **Error** 状態を返すようになりました。
- 式を公開ダッシュボードで使用できるようになりました。
- 結合変換で内部結合がサポートされるようになりました。
- パブリックダッシュボードで Grafana ダッシュボードを共有できるようになりました。
- 新しい Prometheus ストリーミングパーサーが、オプトイン機能として利用できるようになりました。

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。

- [What's new in Grafana v9.1](#)
- [What's new in Grafana v9.2](#)

[Bugzilla:2193018](#)

## Grafana は弱い暗号を有効化しなくなる

この更新により、Grafana は安全な通信の暗号化に弱いと考えられる暗号を有効化しなくなりました。影響を受ける暗号は次のとおりです。

- **AES128-GCM-SHA256**
- **AES128-SHA**
- **AECDHE-RSA-AES128-SHA**
- **AES256-GCM-SHA384**
- **AES256-SHA**
- **ECDHE-RSA-AES256-SHA**

Bugzilla:2190025<sup>[1]</sup>

## .NET 8.0 が利用可能になる

Red Hat Enterprise Linux 9.3 には .NET バージョン 8.0 が同梱されています。以下は、主な改善点です。

- C#12 および F#8 言語バージョンのサポートが追加されました。
- .NET ソフトウェア開発キットを直接使用してコンテナイメージをビルドするためのサポートが追加されました。
- ガベージコレクター (GC)、Just-In-Time (JIT) コンパイラー、およびベースライブラリーのパフォーマンスが多くの点で向上しました。

Jira:RHELPLAN-164399<sup>[1]</sup>

## 4.14. IDENTITY MANAGEMENT

### samba がバージョン 4.18.6 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 4.18.6 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拡張機能が提供されています。最も注目すべき変更点:

- 以前のリリースでのセキュリティの向上は、メタデータの高いワークロードの Server Message Block (SMB) サーバーのパフォーマンスに影響を与えました。この更新により、このシナリオでのパフォーマンスが向上します。
- 新しい **wbinfo --change-secret-at=<domain\_controller>** コマンドは、指定されたドメインコントローラーの信頼アカウントのパスワードの変更を強制します。
- デフォルトでは、Samba はアクセス制御リスト (ACL) をファイルの **security.NTACL** 拡張属性に保存します。/etc/samba/smb.conf ファイルの **acl\_xattr:<security\_acl\_name>** 設定を使用して属性名をカスタマイズできるようになりました。カスタム拡張属性名は、**security.NTACL** のように保護された場所ではないことに注意してください。その結果、サーバーにローカルにアクセスできるユーザーはカスタム属性のコンテンツを変更し、ACL を侵害する可能性があります。

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** ユーティリティを使用して **/etc/samba/smb.conf** ファイルを確認します。

[Bugzilla:2190415](#)

### ipaclient ロールにより、IdM レベルでユーザーの subID 範囲を設定できるようになる

この更新により、**ipaclient ansible-freeipa** ロールにより **ipaclient\_subid** オプションが提供され、これを使用して Identity Management (IdM) レベルでサブ ID 範囲を設定できるようになりました。新しいオプションを明示的に **true** に設定しないと、**ipaclient** ロールはデフォルトの動作を維持し、IdM ユーザーに設定された subID 範囲なしでクライアントをインストールします。

以前は、このロールは **sssd authselect** プロファイルを設定し、それによって **/etc/nsswitch.conf** ファイルがカスタマイズされていました。subID データベースは IdM を使用せず、**/etc/subuid** および **/etc/subgid** のローカルファイルのみに依存しました。

[Bugzilla:2175767](#)

### 複数の IdM グループとサービスを単一の Ansible タスクで管理できるようになる

**ansible-freeipa** のこの機能拡張により、単一の Ansible タスクを使用して、複数の Identity Management (IdM) ユーザーグループとサービスを追加、変更、削除できるようになりました。そのためには、**ipagroup** モジュールと **ipaservice** モジュールの **groups** と **services** オプションを使用します。

**ipagroup** で使用できる **groups** オプションを使用すると、特定のグループにのみ適用される複数のグループ変数を指定できます。このグループは、**name** 変数によって定義されます。これは、**groups** オプションの唯一の必須変数です。

同様に、**ipaservice** で使用できる **services** オプションを使用すると、特定のサービスにのみ適用される複数のサービス変数を指定できます。このサービスは、**name** 変数によって定義されます。これは、**services** オプションの唯一の必須変数です。

[Jira:RHELDPCS-16474<sup>\[1\]</sup>](#)

### ansible-freeipa ipaserver ロールが Random Serial Numbers をサポートできるようになる

この更新により、**ansible-freeipa ipaserver** ロールで **ipaserver\_random\_serial\_numbers=true** オプションを使用できるようになりました。このようにして、Ansible を使用して Identity Management (IdM) サーバーをインストールするときに、PKI で証明書とリクエストの完全にランダムなシリアル番号を生成できます。RSNV3 を使用すると、大規模な IdM インストールでの範囲管理を回避し、IdM を再インストールする際の一般的な競合を防ぐことができます。



#### 重要

RSNV3 は、新しい IdM インストールでのみサポートされます。有効にした場合、すべての PKI サービスで RSNV3 を使用する必要があります。

[Jira:RHELDPCS-16462<sup>\[1\]</sup>](#)

### ipa がバージョン 4.10.2 にリベース

**ipa** パッケージがバージョン 4.10.2 にアップグレードされました。主な変更点は、以下のとおりです。

- IdM CLI および Web UI での証明書の検索とリスト表示のパフォーマンスが向上しました。

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

[Bugzilla:2196426](#)

**ipaserver\_remove\_on\_server** オプションと **ipaserver\_ignore\_topology\_disconnect** オプションが、**ipaserver** ロールで利用できるようになりました。

**ipaserver ansible-freeipa** ロールの **remove\_server\_from\_domain** オプションを使用して、Identity Management (IdM) トポロジーからレプリカを削除すると、トポロジーが切断されます。ここで、ドメインの保持したい部分を指定する必要があります。具体的には、次のことを行う必要があります。

- **ipaserver\_remove\_on\_server** 値を指定して、トポロジーのどの部分を保持するかを指定します。
- **ipaserver\_ignore\_topology\_disconnect** を True に設定します。

**remove\_server\_from\_domain** オプションを使用して IdM からレプリカを削除した場合に、接続されたトポロジーが保持される場合は、これらのオプションはどちらも必要ないことに注意してください。

[Bugzilla:2127903](#)

**IdM は min\_lifetime パラメーターをサポートするようになる**

この機能拡張により、**min\_lifetime** パラメーターが **/etc/gssproxy/\*.conf** ファイルに追加されました。**min\_lifetime** パラメーターは、サービスチケットの残りの有効期間がこの値よりも短い場合にサービスチケットの更新をトリガーします。

デフォルトの値は 15 秒です。NFS などのネットワークボリュームクライアントの場合、KDC が一時的に利用できなくなった場合にアクセスが失われるリスクを軽減するには、この値を 60 秒に設定します。

[Bugzilla:2181465](#)

**ipacert Ansible モジュールを使用して IdM 証明書を管理できるようになる**

**ansible-freeipa ipacert** モジュールを使用して、Identity Management (IdM) ユーザー、ホスト、およびサービスの SSL 証明書をリクエストまたは取得できるようになりました。ユーザー、ホスト、およびサービスは、これらの証明書を使用して IdM に対する認証を行うことができます。証明書を取消したり、保留された証明書を復元したりすることもできます。

[Bugzilla:2127907](#)

**optional\_pac\_tkt\_chksum オプションで、krb5 の異なるバージョン間の相互運用性を維持**

**options\_pac\_tkt\_chksum** オプションを使用して、異なるバージョンの **krb5** パッケージを実行している RHEL Kerberos Distribution Center (KDC) サーバー間の相互運用性を維持できるようになりました。具体的には、特権属性証明書 (PAC) チケットの署名検証に関する動作を変更できます。チケットに署名することが予期される Kerberos プリンシパルに対して **optional\_pac\_tkt\_chksum** 文字列属性を **true** に設定すると、KDC は、PAC チケット署名のないチケットを含む Service for User (S4U) リクエストを拒否しません。チケットに署名するプリンシパルは、チケットのターゲットサービスのレルムに応じて、チケット保証サービス (TGS) のプリンシパルまたはレルム間 TGS のプリンシパルです。

**krb5-1.20** リリース以降、MIT Kerberos KDC では、S4U リクエストを正常に処理できるように、Kerberos チケットの暗号化部分に基づくチケット署名が PAC に存在する必要がありました。以前は、特定の KDC が **krb5-1.19** 以前を使用し、他の KDC が **krb5-1.20** 以降を使用する段階的なアップグレー



ドシナリオで、問題が発生していました。新しいバージョンの **krb5** を S4U リクエストに使用する KDC は、サービスが S4U リクエストに古いバージョンの **krb5** を使用している場合に、当該古いバージョンを使用する KDC が提供するサービスチケットを拒否していました。

この機能が Identity Management (IdM) でどのように使用されるかについて、詳細は [こちらのプルリクエスト](#) を参照してください。

[Bugzilla:2178298](#)

## IdM がリソースベースの制約付き委任をサポートするようになる

この更新により、IdM はリソースベースの制約付き委任 (RBCD) をサポートするようになりました。RBCD を使用すると、リソースレベルで委任をきめ細かく制御でき、認証情報が委任されるサービスの所有者がアクセスを設定できます。

RBCD は、IdM と Active Directory (AD) の統合などに役立ちます。これは、ターゲットサービスとプロキシサービスの両方が異なるフォレストに属している場合に AD が RBCD の使用を強制するためです。



### 重要

現在、RBCD ルールで設定できるのは、IdM ドメイン内のサービスのみです。ターゲットサービスが AD ドメインの一部である場合、パーミッションは AD 側でのみ付与できます。AD ドメインコントローラーは IdM サービス情報を解決してルールを作成することができないため、この機能は現在サポートされていません。

委任シナリオの詳細は、[FreeIPA の設計ページ](#) を参照してください。

[Bugzilla:2165880](#)

## RHEL 9.3 は 389-ds-base 2.3.4 を提供します

RHEL 9.3 は、**389-ds-base** パッケージバージョン 2.3.4 とともに配布されます。バージョン 2.3.4 への主なバグ修正および機能強化は、以下のとおりです。

- <https://www.port389.org/docs/389ds/releases/release-2-2-8.html>
- <https://www.port389.org/docs/389ds/releases/release-2-2-9.html>
- <https://www.port389.org/docs/389ds/releases/release-2-3-0.html>
- <https://www.port389.org/docs/389ds/releases/release-2-3-1.html>
- <https://www.port389.org/docs/389ds/releases/release-2-3-2.html>
- <https://www.port389.org/docs/389ds/releases/release-2-3-3.html>
- <https://www.port389.org/docs/389ds/releases/release-2-3-4.html>

[Bugzilla:2188627](#)

## bind 操作が失敗した場合、Directory Server がクライアント接続を閉じることができるようになる

以前は、**bind** 操作が失敗すると、**bind** 戻りコードを無視する一部のアプリケーションが、さらなるリクエストで Director Server をロードする可能性があります。

**cn=config** エントリーの下の新しい **nsldap-close-on-failed-bind** 設定属性を使用すると、サーバーは **bind** 操作が失敗したときに、クライアント接続を閉じることができます。その結果、サーバーの負荷を軽減することができます。

[Bugzilla:1987471](#)

### Automembership プラグインの改善。デフォルトではグループをクリーンアップしなくなる

以前は、automember 再ビルドタスクは、すべての automember ルールを調べてすべてのメンバーシップを削除した後、メンバーシップを最初から再ビルドしていました。したがって、特に他の **be\_txn** プラグインが有効になっている場合、再ビルドタスクは高いコストがかかりました。

この更新により、Automembership プラグインには次の改善が加えられました。

- 一度に許可される再ビルドタスクは1つだけです。
- Automembership プラグインは、デフォルトでは以前のメンバーをクリーンアップしなくなりました。新しい CLI オプション **--cleanup** を使用して、最初から再ビルドする前にメンバーシップを意図的にクリーンアップします。

```
# dsconf slapd-instance_name plugins automember fixup -f objectclass=posixaccount -s sub
--cleanup "ou=people,dc=example,dc=com"
```

- 修正の進行状況を表示するためのログが改善されました。

[Bugzilla:2149025](#)

### 新しい passwordAdminSkipInfoUpdate: on/off 設定オプションが利用可能になる

新しい **passwordAdminSkipInfoUpdate: on/off** 設定を **cn=config** エントリーに追加すると、パスワード管理者が実行するパスワード更新をきめ細かく制御できます。この設定を有効にすると、パスワードの更新で

は、**passwordHistory**、**passwordExpirationTime**、**passwordRetryCount**、**pwdReset**、および **passwdExpWarned** などの特定の属性は更新されません。

[Bugzilla:2166332](#)

### 新しい slapi\_memberof() プラグイン関数が Directory Server プラグインおよびクライアントアプリケーションで使用できるようになる

新しい **slapi\_memberof()** 関数は、指定されたエントリーが直接的または間接的に属するグループの識別名 (DN) を取得します。以前は、MemberOf、Referential Integrity、および ACL プラグインは、そのようなグループを取得する独自のメカニズムを実装していました。この更新により、グループ DN を返すための統合メカニズムを導入する **slapi\_memberof()** 関数を使用できるようになりました。

[Bugzilla:2189946](#)

### Directory Server は、仮想属性 nsRole を、マネージドロールとフィルターされたロールのインデックス付き属性に置き換えるようになる

以前は、仮想属性 **nsRole** をフィルターに含む LDAP 検索は、その属性にインデックスを付けることができないため、時間がかかりました。この更新により、フィルター内の仮想属性 **nsRole** を使用して **ldapsearch** を実行すると、Directory Server は次の方法で **nsRole** 属性を置き換えます。

- マネージドロールの場合、**nsRole** 属性は **nsRoleDN** 属性に置き換えられます。
- フィルターされたロールの場合、**nsRole** 属性は **nsRoleFilter** 属性に置き換えられます。

その結果、検索にインデックスが付けられるため、**nsRole** 属性を使用した検索の応答時間が改善されます。

この更新は、ネストされたロールには適用されないことに注意してください。

[Bugzilla:2189954](#)

## 新しい **nsslapd-umlisteners** 設定オプションが利用可能になる

**nsslapd-umlisteners** 属性は、確立された接続を監視するために Directory Server が使用できるリスナーレッドの数を指定します。属性値を増やすことで、サーバーで多数のクライアント接続が発生した場合の応答時間を改善できます。

[Bugzilla:1975930](#)

## IdM は、PAC の署名に使用される暗号化タイプを制御するオプションをサポートする

デフォルトでは、Kerberos Key Distribution Center (KDC) は、Privilege Attribute Certificate (PAC) の AES HMAC-SHA2 署名を生成します。ただし、この暗号化タイプは Active Directory (AD) ではサポートされていません。その結果、AD クロスレルムの制約付き委任リクエストは正しく処理されません。

この機能拡張により、TGS プリンシパル **krbtgt/[realm]@[realm]** で **pac\_privsvr\_etype** 属性をターゲットレルムに必要な暗号化タイプに設定することで、PAC の署名に使用される暗号化タイプを制御できるようになりました。IdM では、AD 信頼が存在する場合、この文字列属性は自動的に設定されません。

**WARNING:** This update is about standalone MIT realms. Do not change the Kerberos Distribution Center (KDC) configuration in RHEL Identity Management.

たとえば、**MIT** レルムと **AD** レルムの場合、クロスレルム Ticket-Granting Ticket (TGT) で AD 互換の暗号化タイプが使用されるようにするには、管理者は MIT 側でクロスレルム TGS プリンシパルを次のように設定する必要があります。これにより、レルム間 TGT は AES 256 HMAC-SHA1 暗号化タイプを使用し、制約付き委任リクエストが正しく処理されるようになります。

```
kadmin.local <<EOF
setstr krbtgt/AD@IPA pac_privsvr_etype aes256-cts-hmac-sha1-96
setstr krbtgt/IPA@AD pac_privsvr_etype aes256-cts-hmac-sha1-96
EOF
```

[Bugzilla:2060421](#)

## Identity Management API が完全にサポートされるようになる

Identity Management (IdM) API は、RHEL 9.2 ではテクノロジープレビュー機能として利用可能でしたが、RHEL 9.3 以降では完全にサポートされるようになりました。

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](#)

## 4.15. グラフィックインフラストラクチャー

### Intel Arc A シリーズグラフィックスが完全にサポートされるようになる

以前はテクノロジープレビューとして利用可能であった Intel Arc A-Series グラフィックス (Alchemist または DG2) 機能が完全にサポートされるようになりました。Intel Arc A-Series グラフィックスは、ハードウェアアクセラレーションを可能にする GPU で、主に PC ゲームで使用されます。

[Bugzilla:2101598<sup>\[1\]</sup>](#)

## 4.16. WEB コンソール

### Podman ヘルスチェックアクションが利用可能になる

新しいコンテナを作成するときに、次の Podman ヘルスチェックアクションのいずれかを選択できます。

- No action (デフォルト): アクションは実行しません。
- Restart: コンテナを再起動します。
- Stop: コンテナを停止します。
- Force stop: コンテナを強制的に停止します。コンテナが終了するのを待ちません。

[Jira:RHELDPCS-16247<sup>\[1\]</sup>](#)

### Stratis が RHEL Web コンソールで利用可能になる

この更新により、Red Hat Enterprise Linux Web コンソールは、Stratis ストレージを管理できるようになりました。

Stratis の詳細は、[Web コンソールを使用した Stratis ファイルシステムのセットアップ](#) を参照してください。

[Jira:RHELPLAN-122345<sup>\[1\]</sup>](#)

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

### systemd ユニットの管理するための新しい RHEL システムロール

**rhel-system-role** パッケージには、**systemd** RHEL システムロールが含まれるようになりました。このロールを使用して、ユニットファイルをデプロイし、複数のシステム上で **systemd** ユニットの管理できます。**systemd** ユニットファイルとテンプレートを提供し、それらのユニットの状態 (開始、停止、

マスクなど)を指定することで、**systemd** の機能を自動化できます。

[Bugzilla:2224384](#)

### 設定バックアップを無効にする **ssh** ロールの新しいオプション

新しい **ssh\_backup** オプションを **false** に設定することで、古い設定ファイルが上書きされる前にバックアップされることを阻止できるようになりました。以前は、バックアップ設定ファイルが自動的に作成されていましたが、これは不要な場合があります。**ssh\_backup** オプションのデフォルト値は **true** で、元の動作が保持されます。

[Bugzilla:2216753](#)

### **keylime\_server** RHEL システムロール

新しい **keylime\_server** RHEL システムロールを使用すると、Ansible Playbook を使用して、RHEL 9 システム上で verifier および registrar Keylime コンポーネントを設定できます。Keylime は、Trusted Platform Module (TPM) テクノロジーを使用するリモートマシン証明ツールです。

[Bugzilla:2224385](#)

### 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。

- リソースおよびリソース操作のデフォルトの設定 (ルールを含む複数のデフォルトのセットを含む)。
- SBD watchdog カーネルモジュールのロードとブロック。これにより、インストールされているハードウェアウォッチドッグをクラスターで使用できるようになります。
- クラスターホストとクォーラムデバイスへの個別のパスワードの割り当て。これにより、同じクォーラムホストが複数の個別のクラスターに参加し、これらのクラスター上の **hacluster** ユーザーのパスワードが異なるデプロイメントを設定できます。

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。

[Bugzilla:2185065](#), [Bugzilla:2185067](#), [Bugzilla:2216481](#)

### **storage** システムのロールは、RAID LVM ボリュームのストライプサイズの設定をサポートする

この更新により、RAID LVM デバイスの作成時に、カスタムストライプサイズを指定できるようになりました。パフォーマンスを向上させるには、SAP HANA のカスタムストライプサイズを使用してください。RAID LVM ボリュームの推奨ストライプサイズは 64 KB です。

[Bugzilla:2181656](#)

### **network** RHEL システムロールは、DNS レコードの自動更新を制御する **auto-dns** オプションをサポートする

この機能拡張により、定義されたネームサーバーと検索ドメインのサポートが提供されます。DHCP からの **dns record** など、自動的に設定されたネームサーバーと検索ドメインを無効にしなが、**dns** および **dns\_search** プロパティーで指定されたネームサーバーと検索ドメインのみを使用できるようになりました。この機能拡張により、**auto-dns** 設定を変更することで、自動 DNS 記録を自動的に無効にすることができます。

[Bugzilla:2211194](#)

## network RHEL システムロールは、no-aaaa DNS オプションをサポートする

**no-aaaa** オプションを使用して、マネージドノードで DNS 設定を設定できるようになりました。以前は、**getaddrinfo** などの NSS ベースのインターフェイスによってトリガーされる AAAA ルックアップを含む、スタブリゾルバーによって生成される AAAA クエリーを抑制するオプションはありませんでした。DNS ルックアップのみが影響を受けていました。この機能拡張により、スタブリゾルバーによって生成された AAAA クエリーを抑制できるようになりました。

[Bugzilla:2218592](#)

## ad\_integration RHEL システムロールが AD ドメインに再参加できるようになる

この更新により、**ad\_integration** RHEL システムロールを使用して Active Directory (AD) ドメインに再参加できるようになりました。これを行うには、**ad\_integration\_force\_rejoin** 変数を **true** に設定します。**realm\_list** の出力で、ホストがすでに AD ドメインに存在していることが示されている場合、ホストは再参加する前に既存のドメインから離脱します。

[Bugzilla:2211723](#)

## certificate RHEL システムロールで、certmonger の使用時に証明書ファイルモードを変更できるようになる

以前は、**certmonger** プロバイダーを使用して **certificate** RHEL システムロールによって作成された証明書は、デフォルトのファイルモードを使用していました。ただし、ユースケースによっては、より制限のあるモードが必要になる場合があります。この更新により、**mode** パラメーターを使用して別の証明書とキーファイルモードを設定できるようになりました。

[Bugzilla:2218204](#)

## postgresql RHEL システムロールが利用可能になる

新しい **postgresql** RHEL システムロールは、**PostgreSQL** サーバーをインストール、設定、管理、起動します。このロールは、データベースサーバー設定を最適化してパフォーマンスを向上させます。

このロールは、RHEL 8 および RHEL 9 マネージドノード上で現在リリースされサポートされているバージョンの **PostgreSQL** をサポートします。

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。

[Bugzilla:2151373](#)

## podman RHEL システムロールが Quadlets、ヘルスチェック、シークレットをサポートできるようになる

Podman 4.6 以降では、**podman** RHEL システムロールで **podman\_quadlet\_specs** 変数を使用できるようになりました。Quadlet は、ユニットファイルを指定するか、インベントリ内で名前、ユニットのタイプ、仕様を指定して定義できます。ユニットのタイプは次のとおりです：

**container**、**kube**、**network**、および **volume**。Quadlet は RHEL 8 上のルートコンテナでのみ機能することに注意してください。Quadlet は、RHEL 9 上のルートレスコンテナで動作します。

ヘルスチェックは、Quadlet コンテナタイプに対してのみサポートされます。**[Container]** セクションで、**HealthCmd** フィールドを指定してヘルスチェックコマンドを定義し、**HealthOnFailure** フィールドを指定してコンテナが異常な場合のアクションを定義します。可能なオプションは、**none**、**kill**、**restart**、および **stop** です。

`podman_secrets` 変数を使用してシークレットを管理できます。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-154441<sup>[1]</sup>

### restorecon -T 0 を使用した selinux システムロールのパフォーマンスの向上

`selinux` システムロールは、該当するすべてのケースで、`restorecon` コマンドで `-T 0` オプションを使用するようになりました。これにより、ファイル上でデフォルトの SELinux セキュリティーコンテキストを復元するタスクのパフォーマンスが向上します。

Bugzilla:2179460

### rhc システムロールがプロキシサーバータイプの設定をサポートするようになる

`rhc_proxy` パラメーターの下に新しく導入された属性 `スキーム` により、`rhc` システムロールを使用してプロキシサーバータイプを設定できるようになります。デフォルトの `http` と `https` の2つの値を設定できます。

Bugzilla:2211748

### firewall RHEL システムロールは、ipsets に関連する変数をサポートする

`firewall` RHEL システムロールの今回の更新により、`ipsets` を定義、変更、削除できるようになりました。また、これらの `ipsets` をファイアウォールゾーンに追加したり、ファイアウォールゾーンから削除したりすることもできます。あるいは、ファイアウォールのリッチルールを定義するときに、これらの `ipsets` を使用することもできます。

次の変数を使用して、`firewall` RHEL システムロールで `ipsets` を管理できます。

- `ipset`
- `ipset_type`
- `ipset_entries`
- `short`
- `description`
- `state: present` または `state: absent`
- `permanent: true`

この機能拡張の主な利点は次のとおりです。

- 多くの IP アドレスのルールを定義するリッチルールの複雑さを軽減できます。
- 複数のルールを変更せずに、必要に応じてセットに IP アドレスを追加または削除できます。

詳細は、`/usr/share/doc/rhel-system-roles/firewall/` ディレクトリー内のリソースを参照してください。

Bugzilla:2229802

マウントポイントカスタマイズ用の新しいボリュームオプションが RHEL システムロールに追加される

この更新により、マウントディレクトリーに **mount\_user**、**mount\_group**、および **mount\_permissions** パラメーターを指定できるようになりました。

[Bugzilla:2181657](#)

**firewall RHEL システムロールには、競合するサービスを無効にするオプションがあり、firewalld がマスクされている場合でも失敗しなくなる**

以前は、ロールの実行時に **firewalld** サービスがマスクされていた場合、または競合するサービスが存在した場合に、**firewall** システムロールは失敗していました。この更新では、次の2つの主な機能拡張が提供されます。

**linux-system-roles.firewall** ロールは、ロールの実行時に常に **firewalld** サービスのインストール、マスク解除、および有効化を試みます。新しい変数 **firewall\_disable\_conflicting\_services** を Playbook に追加して、競合する既知のサービス (**iptables.service**、**nftables.service**、**ufw.service** など) を無効化できるようになりました。**firewall\_disable\_conflicting\_services** 変数はデフォルトで **false** に設定されます。競合するサービスを無効にするには、変数を **true** に設定します。

[Bugzilla:2222761](#)

**firewall RHEL システムロール設定のリセットには、最小限のダウンタイムが必要**

以前は、**previous: replace** 変数を使用して **firewall** ロール設定をリセットすると、**firewalld** サービスが再起動されました。再起動するとダウンタイムが追加され、**firewalld** がアクティブな接続からのトラフィックをブロックしないオープン接続の期間が長くなります。この機能拡張により、**firewalld** サービスは再起動ではなく再ロードすることで、設定のリセットを完了します。リロードによりダウンタイムが最小限に抑えられ、ファイアウォールルールをバイパスする機会が減ります。その結果、**以前の: replace** 変数を使用して **firewall** ロール設定をリセットすると、ダウンタイムが最小限で済むようになりました。

[Bugzilla:2223764](#)

## 4.18. 仮想化

**sevctl が AMD EPYC Rome および Milan と完全に互換性を持つようになりました。**

今回の更新で、**sevctl** ユーティリティーは、AMD EPYC Rome および AMD EPYC Milan シリーズを含む最新の AMD EPYC コアを正しく認識するようになりました。これにより、**sevctl** を使用して、これらの CPU で利用可能な AMD Secure Encrypted Virtualization (SEV) の機能を設定できます。

ただし、SEV-ES や SEV-SNP などの高度な SEV 機能は、RHEL 9 ではテクノロジープレビューとしてのみ提供されているため、サポート対象外であることに注意してください。

[Bugzilla:2104857<sup>\[1\]</sup>](#)

**virtio-vga および virtio-gpu デバイスが blob リソースをサポートするようになる**

**virtio-vga** および **virtio-gpu** デバイスが **blob** メモリーリソースを使用できるようになり、特定のシナリオでのパフォーマンスが向上します。**blob** リソースを **virtio** グラフィックスデバイスにアタッチするには、仮想マシンの XML 設定内の対応する **<video>** セクションに、**blob="on"** オプションを追加します。以下に例を示します。

```
<video>
  <model type="virtio" heads="1" primary="yes" blob="on"/>
  <address type="pci" domain="0x0000" bus="0x00" slot="0x01" function="0x0"/>
</video>
```



ただし、この機能は現在、IBM Z ホストでは動作しないことに注意してください。

[Bugzilla:2032406](#)

## 第 4 世代 Intel Xeon Scalable プロセッサの仮想化サポート

今回の更新で、RHEL 9 の仮想化により、第 4 世代 Intel Xeon Scalable プロセッサ (旧称 Sapphire Rapids) のサポートが追加されました。その結果、RHEL 9 でホストされる仮想マシンは、**SapphireRapids** CPU モデルを使用し、プロセッサが提供する新機能を活用できるようになりました。

[Bugzilla:1880531](#)<sup>[1]</sup>

### Secure Execution on IBM Z のメモリー回収の改善

Secure Execution on IBM Z で仮想マシン (VM) を使用する場合は、仮想マシンで強化されたメモリー回収をセットアップできるようになりました。仮想マシンが 32 GiB 以上の RAM を使用している場合、この設定により、仮想マシンの再起動または停止のパフォーマンスが向上します。

仮想マシンで強化されたメモリー回収をセットアップするには、XML 設定の `<features></features>` セクションに `<async-teardown enabled='yes'/>` 行を追加します。

[Bugzilla:2168499](#)<sup>[1]</sup>

### 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。

- クラウドイメージをベースにする仮想マシン (VM) の **Create and edit** ボタンを選択すると、VM をインストールする前にすべての VM プロパティを編集できます。
- 仮想マシンの作成時に **raw** ストレージボリュームを作成します。
- 仮想ソケット (vsock) を設定して、ソケット経由でのホストと VM 間の通信を可能にします。仮想ソケットで通信を有効にするには、**socat** などの vsock 対応ソフトウェアが必要であることに注意してください。

[Jira:RHELDPCS-16487](#)<sup>[1]</sup>

## 4.19. クラウド環境の RHEL

### cloud-init は、NetworkManager キーファイルをサポートする

この更新により、**cloud-init** ユーティリティは NetworkManager (NM) キーファイルを使用して、作成されたクラウドインスタンスのネットワークを設定できるようになりました。

デフォルトでは、**cloud-init** は、依然としてネットワークセットアップに **sysconfig** メソッドを使用することに注意してください。代わりに NM キーファイルを使用するように **cloud-init** を設定するには、`/etc/cloud/cloud.cfg` を編集し、**network-manager** をプライマリーネットワークレンダラーとして設定します。

```
# cat /etc/cloud/cloud.cfg

network:
  renderers: ['network-manager', 'eni', 'netplan', 'sysconfig', 'networkd']
```

Bugzilla:2118235<sup>[1]</sup>

## cloud-init が ESXi 上でデフォルトで VMware データソースを使用するようになる

VMware vSphere クラウドプラットフォームなどの VMware ESXi ハイパーバイザーを使用するホスト上に RHEL 仮想マシン (VM) を作成する場合。これにより、**cloud-init** を使用して RHEL の ESXi インスタンスを作成する際のパフォーマンスと安定性が向上します。ただし、ESXi は Open Virtualization Format (OVF) データソースと引き続き互換性があり、VMware のデータソースが使用できない場合でも OVF データソースを使用できる点に留意してください。

Bugzilla:2172341<sup>[1]</sup>

## 4.20. サポート性

### sos がバージョン 4.6 にリベース

設定、診断、およびトラブルシューティングデータを収集するための **sos** ユーティリティーが、バージョン 4.6 にリベースされました。この更新では、次の機能拡張が提供されます。

- **sos** レポートには、起動に関する問題のトラブルシューティングに重要な情報が含まれる可能性がある **/boot/grub2/custom.cfg** ファイルと **/boot/grub2/user.cfg** ファイルの両方の内容が含まれるようになりました。(BZ#2213951)
- OVN-Kubernetes の **sos** プラグインは、相互接続環境の追加のログを収集します。今回の更新により、**ovnkube-node** および **ovnkube-controller** の両方のコンテナが1つにマージされるときに、**sos** は **ovnkube-controller** コンテナからもログを収集します。

さらに、注目すべきバグ修正には以下が含まれます。

- **sos** は、OpenShift Container Platform 4 環境で **cgroup** データを正しく収集するようになりました (BZ#2186361)。
- **sudo** プラグインを有効にして **sos** レポートを収集しているときに、**sos** は **bindpw** オプションを適切に削除するようになりました。(BZ#2143272)
- **subscription\_manager** プラグインは、**/var/lib/rhsm/** パスからプロキシのユーザー名とパスワードを収集しなくなりました。(BZ#2177282)
- **virsh** プラグインは、virt-manager ログ内の SPICE リモートディスプレイパスワードを収集しなくなりました。これにより、**sos** がレポートでパスワードを開示できなくなります。(BZ#2184062)
- **sos** は、**/var/lib/iscsi/nodes/<IQN>/<PortallP>/default** ファイルに以前に表示されていたユーザー名とパスワードをマスクするようになりました。



### 重要

生成されたアーカイブには、機密とみなされるデータが含まれる可能性があります。したがって、アーカイブを第三者に渡す前に必ず内容を確認する必要があります。

(BZ#2187859)

- **sos** は、ログファイルのサイズが超過した場合やプラグインがタイムアウトした場合でも、テールログ収集を完了します。(BZ#2203141)

- Pacemaker クラスターノードで **soscollect** コマンドを入力すると、**sos** は、同じクラスターノードから sos レポートを収集します。(BZ#2186460)
- OpenShift Container Platform 4 環境のホストからデータを収集する場合、**sos** は **sysroot** パスを使用するようになり、正しいデータのみがアセンブルされるようになります。(BZ#2075720)
- **sos report --clean** コマンドは、意図したとおりにすべての MAC アドレスを難読化します。(BZ#2207562)
- **hpssm** プラグインを無効にしても例外は発生しなくなりました。(BZ#2216608)
- **sos clean** コマンドは、サニタイズされたファイルのパーミッションに従います。(BZ#2218279)

**sos** の各リリースの詳細は、[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-156196<sup>[1]</sup>

## 4.21. コンテナ

### Podman は、zstd で圧縮されたイメージのプルとプッシュをサポートする

**zstd** 形式で圧縮されたイメージをプルおよびプッシュできます。zstd 圧縮は、gzip よりも効率的で高速です。イメージのプルとプッシュに関連するネットワークトラフィックとストレージの量を削減できます。

Jira:RHELPLAN-154314<sup>[1]</sup>

### Podman の Quadlet が利用可能になる

Podman v4.6 以降では、Quadlet を使用して、コンテナの説明から **systemd** サービスファイルを自動的に生成できます。Quadlets は、関連するコンテナの詳細に焦点を当てて説明しており、**systemd** でコンテナを実行するという技術的な複雑さを考慮していないため、**podman generate systemd** コマンドよりも使いやすい可能性があります。

詳細は、[Quadlet upstream documentation](#) および [Make systemd better for Podman with Quadlet](#) の記事を参照してください。

Jira:RHELPLAN-154432<sup>[1]</sup>

### 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。この更新では、以前のバージョンに対する一連のバグ修正と機能拡張が適用されます。

Podman v4.6 の注目すべき変更点は次のとおりです。

- **podman kube play** コマンドは、Pod のコンテナ内で使用される環境変数を Kubernetes YAML ファイルに提供する **--configmap=<path>** オプションをサポートするようになりました。
- **podman kube play** コマンドは、**--configmap** オプション用の複数の Kubernetes YAML ファイルをサポートするようになりました。

- **podman kube play** コマンドは、liveness プロブ内の containerPort 名とポート番号をサポートするようになりました。
- **podman kube play** コマンドは、ctrName をエイリアスとして Pod ネットワークに追加するようになりました。
- **podman kube play** および **podman kube generated** コマンドは、SELinux ファイルタイプラベルと ulimit アノテーションをサポートするようになりました。
- 指定された名前のシークレットが存在するかを検証する、新しいコマンド **podman secret exists** が追加されました。
- **podman create**、**podman run**、**podman pod create**、および **podman pod clone** コマンドは、新しいオプション **--shm-size-systemd** をサポートするようになりました。これにより、systemd 固有のマウントの tmpfs サイズを制限できるようになります。
- **podman create** および **podman run** コマンドは、新しいオプション **--security-opt label=nested** をサポートするようになりました。これにより、制限されたコンテナ内で SELinux のラベル付けが可能になります。
- Podman は、Pod 内で実行されているコンテナの自動更新をサポートするようになりました。
- Podman は、安定性を高めるために SQLite データベースをバックエンドとして使用できるようになりました。デフォルトは BoltDB データベースのままです。データベースを選択するには、**containers.conf** ファイルの **database\_backend** フィールドを設定します。
- Podman は、コンテナの説明から **systemd** サービスファイルを自動的に生成する Quadlets をサポートするようになりました。この説明では、関連するコンテナの詳細に焦点を当てており、**systemd** でコンテナを実行する際の技術的な複雑さは隠しています。

注目すべき変更点の詳細は、[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-154438<sup>[1]</sup>

## Podman が Podmash ログインシェルをサポートするようになる

Podman v4.6 以降、**Podmash** ログインシェルを使用して、ユーザーアクセスと制御を管理できるようになりました。標準のシェルコマンド (**/usr/bin/bash** など) の代わりに、**/usr/bin/podmash** コマンドをログインシェルとして使用するよう設定します。ユーザーがシステムセットアップにログインすると、**podmash** コマンドはユーザーのセッションを **podmash** という名前の Podman コンテナ内で実行します。ユーザーがログインするコンテナは、**/etc/containers/systemd/users/** ディレクトリーに作成される Quadlet ファイルを使用して定義されます。これらのファイルで、**[Container]** セッションの **ContainerName** フィールドを **podmash** に設定します。systemd は、ユーザーセッションの開始時に **podmash** を自動的に起動し、すべてのユーザーセッションが終了するまで実行を続けます。

詳細は、[Podman v4.6.0 Introduces Podmash: A Revolutionary Login Shell](#) を参照してください。

Jira:RHELPLAN-163003<sup>[1]</sup>

## Fulcio および Rekor を使用した sigstore 署名のクライアントが利用可能になる

Fulcio および Rekor サーバーを使用すると、秘密鍵を手動で管理する代わりに、OpenID Connect (OIDC) サーバー認証に基づく短期証明書を使用して署名を作成できるようになりました。以前はテクノロジープレビューとして利用可能であった、Fulcio および Rekor を使用した sigstore 署名のクライア

ントが、完全にサポートされるようになりました。この追加機能はクライアント側のサポートのみであり、Fulcio サーバーや Rekor サーバーは含まれません。

**policy.json** ファイルに **fulcio** セクションを追加します。コンテナイメージに署名するには、**podman push --sign-by-sigstore=file.yml** または **skopeo copy --sign-by-sigstore=file.yml** コマンドを使用します。ここで、**file.yml** は sigstore 署名パラメーターファイルです。

署名を検証するには、**policy.json** ファイルに **fulcio** セクションと **rekorPublicKeyPath** または **rekorPublicKeyData** フィールドを追加します。詳細は、**containers-policy.json** の man ページを参照してください。

Jira:RHELPLAN-160660<sup>[1]</sup>

### pasta ネットワーキングモードが利用可能になる

Podman v4.4.1以降、**pasta** ネットワークモードを使用できるようになりました。これは、デフォルトのネットワークモード **slirp4netns** の高性能な代替品であり、IPv6 転送をサポートします。**pasta** ネットワークモードを選択するには、**passt** パッケージをインストールして、**--network=pasta** オプションを指定して **podman run** コマンドを使用します。Podman v4.6 では、**[network]** セクションの **default\_rootless\_network\_cmd** フィールドを使用して、**/etc/containers/containers.conf** 設定ファイルでデフォルトのルートレスネットワークモードを設定できます。

Jira:RHELDPCS-16240<sup>[1]</sup>

### UBI 9 Micro Container Image には、tzdata によってインストールされた zoneinfo が含まれなくなる

この更新により、**tzdata** パッケージによって提供されるタイムゾーン情報が UBI 9 Micro Container Image に含まれなくなり、その結果イメージサイズが縮小されます。UBI 9 Minimal Container と UBI 9 Micro Container は UTC のみであり、ユーザーは必要に応じて **tzdata** パッケージを再インストールして完全な **zoneinfo** を取得する必要があります。

Bugzilla:2223028

## 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 9.3 で配布されるカーネルの重要な変更点を概説します。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

#### amd\_pstate=[X86]

このカーネルパラメーターを使用すると、AMD CPU のパフォーマンスを拡張できます。利用可能な値は次のとおりです。

- **disable** - サポートされているプロセッサのデフォルトのスケーリングドライバーとして **amd\_pstate** を有効にしないでください。
- **passive** - スケーリングドライバーとしてパッシブモードで **amd\_pstate** を使用します。このモードでは、自律的な選択は無効化されています。ドライバーは必要なパフォーマンスレベルを要求します。プラットフォームはそのパフォーマンスレベルに一致させようとします (保証されたパフォーマンスレベルによって満たされる場合)。
- **active - amd\_pstate\_epp** ドライバーインスタンスをスケーリングドライバーとして使用します。ソフトウェアが CPPC ファームウェアに対してパフォーマンス (0x0) またはエネルギー効率 (0xff) に偏りたい場合、ドライバーはハードウェアにヒントを提供します。次に、CPPC 電力アルゴリズムがランタイムワークロードを計算し、リアルタイムコア周波数を調整します。
- **guided** - ガイド付き自律モードを有効にします。ドライバーは最小および最大のパフォーマンスレベルを要求し、プラットフォームはこの範囲内で現在のワークロードに適したパフォーマンスレベルを自律的に選択します。

#### arm64.nosve=[ARM64]

このカーネルパラメーターを使用すると、Scalable Vector Extension のサポートを無条件に無効化できます。

#### arm64.nosme=[ARM64]

このカーネルパラメーターを使用すると、Scalable Matrix Extension のサポートを無条件に無効にすることができます。

#### gather\_data\_sampling=[X86,INTEL]

このカーネルパラメーターを使用すると、Gather Data Sampling (GDS) の軽減策を制御できます。GDS は、ベクトルレジスターに以前に格納されていたデータへの特権のない投機的アクセスを許可するハードウェアの脆弱性です。

この問題は、更新されたマイクロコードではデフォルトで軽減されています。軽減策はパフォーマンスに影響を与える可能性があります。無効にすることができます。マイクロコード軽減策のないシステムが、軽減策として AVX を無効にしています。利用可能な値は次のとおりです。

- **force** - AVX を無効にして、マイクロコード軽減策を使用せずにシステムを軽減します。マイクロコード軽減策が存在する場合は効果がありません。バグのある AVX 列挙によりユーザー空間でクラッシュが発生することが知られています。
- **off** - GDS 軽減策を無効化します。

#### nospectre\_bhb=[ARM64]

このカーネルパラメーターを使用すると、Spectre-BHB (ブランチ履歴インジェクション) の脆弱性に対するすべての軽減策を無効化できます。このオプションを使用すると、システムによりデータ漏洩が発生する可能性があります。

#### trace\_clock=[FTRACE]

このカーネルパラメーターを使用すると、起動時のイベントのトレースに使用されるクロックを設定できます。利用可能な値は次のとおりです。

- **local** - CPU ごとのタイムスタンプカウンターを使用します。
- **global** - イベントのタイムスタンプは CPU 全体で同期されます。ローカルクロックよりも遅い可能性があります、一部の競合状態には適しています。
- **counter** - イベントの単純なカウント (1, 2, ...)。インフラストラクチャーがイベントごとに複数回クロックを取得するため、一部のカウントがスキップされる可能性があることに注意してください。
- **uptime** - タイムスタンプとして jiffies を使用します。
- **perf** - perf が使用するのと同じクロックを使用します。
- **mono** - タイムスタンプに `ktime_get_mono_fast_ns()` 関数を使用します。
- **mono\_raw** - タイムスタンプに `ktime_get_raw_fast_ns()` 関数を使用します。
- **boot** - タイムスタンプに `ktime_get_boot_fast_ns()` 関数を使用します。  
アーキテクチャーによっては、さらに多くのクロックが追加される場合があります。詳細は、[Documentation/trace/ftrace.rst](#) を参照してください。

#### 更新されたカーネルパラメーター

##### cgroup.memory=[KNL]

このカーネルパラメーターを使用すると、**cgroup** メモリーコントローラーにオプションを渡すことができます。

- このパラメーターの形式は次のとおりです: `<string>`  
利用可能な値は次のとおりです。
- **nosocket** - ソケットメモリーアカウンティングを無効にします。
- **nokmem** - カーネルメモリーアカウンティングを無効にします。
- **[新規] nobpf** - BPF メモリーアカウンティングを無効にします。

##### hugetlb\_free\_vmemmap=[KNL]

このカーネルパラメーターは、ブート時に各 hugetlb ページに関連付けられた未使用の **vmemmap** ページを解放する機能を有効にします。このパラメーターが機能するには、**CONFIG\_HUGETLB\_PAGE\_OPTIMIZE\_VMEMMAP** 設定オプションを有効にする必要があります。

このパラメーターの形式は次のとおりです: `{ on | off (default) }`

利用可能な値は次のとおりです。

- **on** - この機能を有効にします。
- **off** - この機能を無効にします。



## 注記

**memory\_hotplug.memmap\_on\_memory** モジュールパラメーターが有効な場合、**vmemmap** ページは追加されたメモリーブロック自体から割り当てられる場合があります。この機能が有効になっている場合でも、これらの **vmemmap** ページは最適化できません。追加されたメモリーブロック自体から割り当てられていない他の **vmemmap** ページは影響を受けません。

### intel\_pstate=[X86]

このカーネルパラメーターを CPU パフォーマンスのスケーリングに使用できます。利用可能な値は次のとおりです。

- **disable** - サポートされているプロセッサのデフォルトのスケーリングドライバーとして **intel\_pstate** を有効にしないでください。
- **[新規] active** - **intel\_pstate** ドライバーを使用して、**cpufreq** のスケーリングガバナー層をバイパスし、p-state 選択用の独自のアルゴリズムを提供します。アクティブモードで **intel\_pstate** によって提供される P-state 選択アルゴリズムには、powersave と performance の 2 つがあります。両方の動作方法は、プロセッサで、そして場合によってはプロセッサモデルで、ハードウェア管理 P-state (HWP) 機能が有効になっているかどうかによって異なります。
- **パッシブ** - **intel\_pstate** をスケーリングドライバーとして使用しますが、(内部ガバナーを有効にするのではなく) 汎用の **cpufreq** ガバナーと連携して動作するように設定します。このモードは、ハードウェア管理の P-state (HWP) 機能と一緒に使用することはできません。
- **force** - **acpi-cpufreq** を優先して **intel\_pstate** をデフォルトで禁止しているシステムで、これを有効化します。**acpi-cpufreq** の代わりに **intel\_pstate** ドライバーを強制すると、OSPM に示される ACPI P-States 情報に依存するプラットフォーム機能 (温度制御や電力制限など) が無効になる可能性があるため、注意して使用する必要があります。このオプションは、**intel\_pstate** ドライバーでサポートされていないプロセッサや、**acpi-cpufreq** の代わりに **pcc-cpufreq** を使用するプラットフォームでは機能しません。
- **no\_hwp** - ハードウェア P ステートコントロール (HWP) が利用可能な場合は有効にしません。
- **hwp\_only** - ハードウェア P state コントロール (HWP) が利用可能な場合は、それをサポートするシステムにのみ **intel\_pstate** をロードします。
- **support\_acpi\_ppc** - **ACPI\_PPC** パフォーマンス制限を強制します。Fixed ACPI Description Table で優先電源管理プロファイルを "エンタープライズサーバー" または "パフォーマンスサーバー" として指定している場合、この機能はデフォルトでオンになります。
- **per\_cpu\_perf\_limits** - **cpufreq sysfs** インターフェイスを使用して、論理 CPU ごとの P-State パフォーマンス制御の制限を許可します。

### kvm-arm.mode=[KVM,ARM]

このカーネルパラメーターを使用すると、KVM/arm64 の動作モードの 1 つを選択できます。利用可能な値は次のとおりです。

- **none** - KVM を強制的に無効にします。
- **nvhe** - 標準の nVHE ベースのモード。protected ゲストはサポートされません。



- **protected** - 状態がホストからプライベートに保たれるゲストをサポートする nVHE ベースのモード。モードを **protected** に設定すると、ホストの **kexec** と休止状態が無効になります。
- **[新規] nested** - ネストされた仮想化をサポートする VHE ベースのモード。少なくとも ARMv8.3 ハードウェアが必要です。 **nested** オプションは実験的なものであるため、細心の注意を払って使用する必要があります。  
ハードウェアのサポートに基づいて、デフォルトは VHE/nVHE になります。

### libata.force=[LIBATA]

このカーネルパラメーターを使用すると、設定を強制できます。

形式は "[ID:]VAL" のコンマ区切りリストで、この場合の ID は PORT[.DEVICE] になります。PORT と DEVICE は、ポート、リンク、またはデバイスに一致する 10 進数です。基本的に、これは **libata** によってコンソールに出力される ATA ID 文字列と一致します。

- ID 部分全体が省略された場合は、最後の **PORT** と **DEVICE** の値が使用されます。
- ID がまだ指定されていない場合、設定はすべてのポート、リンク、デバイスに適用されます。
- **DEVICE** 値のみを省略した場合、パラメーターはポートとその背後にあるすべてのリンクおよびデバイスに適用されます。DEVICE 番号 0 は、最初のデバイスまたは PMP デバイスの背後にある最初のファンアウトリンクを選択します。ホストリンクは選択しません。DEVICE 番号 15 は、ホストリンクとそれに接続されているデバイスを選択します。
- VAL は、強制する設定を指定します。曖昧さがない限り、ショートカット表記が許可されます。たとえば、1.5 と 1.5G はどちらも 1.5Gbps で動作します。  
**libata.force=** パラメーターを使用すると、次の設定を強制できます。
- ケーブルタイプ: 40c、80c、short40c、unk、ign、sata。一致する PORT を持つ任意の ID が使用されます。
- SATA リンク速度制限: 1.5Gbps または 3.0Gbps。
- 転送モード: pio[0-7]、mwdma[0-4] および udma[0-7]。udma[/][16,25,33,44,66,100,133] 表記も使用可能です。
- **nohrst**、**nosrst**、**norst**: ハード、ソフト、および両方のリセットを抑制します。
- **rstonec**: ホットアンプラグリンクリカバリー中にリセットを 1 回だけ試行します。
- **新規 [no]dbdelay**: リンク PHY とデバイス存在検出をデバウンスする前に、追加の 200 ミリ秒の遅延を有効または無効にします。
- **[no]ncq**: NCQ をオンまたはオフにします。
- **[no]ncqtrim**: キューに入れられた DSM TRIM を有効または無効にします。
- **[新規] [no]ncqati**: ATI チップセットの NCQ トリムを有効または無効にします。
- **[新規] [no]trim**: TRIM を有効または無効 (キューから外す) にします。
- **[新規] trim\_zero**: TRIM コマンドがデータをゼロ化することを示します。
- **[新規] max\_trim\_128m**: 最大トリムサイズ制限を 128M に設定します。
- **[新規] [no]dma**: DMA 転送をオンまたはオフにします。

- **atapi\_dmadir**: ATAPI DMADIR ブリッジのサポートを有効にします。
- **atapi\_mod16\_dma**: 16 バイトの倍数ではないコマンドに対する ATAPI DMA の使用を有効にします。
- **[no]dmalog**: ログにアクセスするための READ LOG DMA EXT コマンドの使用を有効または無効にします。
- **[no]iddevlog**: デバイスの識別データログへのアクセスを有効または無効にします。
- **[no]logdir**: 汎用ログディレクトリへのアクセスを有効または無効にします。
- **[新規] max\_sec\_128**: 転送サイズ制限を 128 セクターに設定します。
- **[新規] max\_sec\_1024**: 転送サイズ制限を 1024 セクターに設定またはクリアします。
- **[新規] max\_sec\_lba48**: 転送サイズ制限を 65535 セクターに設定またはクリアします。
- **[新規] [no]lpm**: リンク電源管理を有効または無効にします。
- **[新規] [no]setxfer**: 転送速度モード設定をスキップするかどうかを示します。
- **[新規] [no]fua**: この機能をサポートするデバイスの FUA (Force Unit Access) サポートを無効または有効にします。
- **dump\_id**: IDENTIFY データをダンプします。
- **disable**: このデバイスを無効にします。



### 注記

同じ属性を変更する設定が複数一致する場合は、最後のものが使用されません。

### mitigations=[X86,PPC,S390,ARM64]

このカーネルパラメーターを使用すると、CPU の脆弱性に対するオプションの軽減策を制御できません。これは、既存のアーキテクチャー固有のオプションの集約となる、一連の、アーキテクチャーに依存しないオプションです。利用可能な値は次のとおりです。

- **off** - オプションの CPU 軽減策をすべて無効にします。これによりシステムパフォーマンスが向上しますが、ユーザーを複数の CPU の脆弱性にさらす可能性もあります。**off** 値は以下と同等です。
  - nokaslr の場合、kpti=0 [ARM64]
  - gather\_data\_sampling=off [X86]
  - kvm.nx\_huge\_pages=off [X86]
  - l1tf=off [X86]
  - mds=off [X86]
  - mmio\_stale\_data=off [X86]
  - no\_entry\_flush [PPC]

- no\_uaccess\_flush [PPC]
- nobp=0 [S390]
- nopti [X86,PPC]
- nospectre\_bhb [ARM64]
- nospectre\_v1 [X86,PPC]
- nospectre\_v2 [X86,PPC,S390,ARM64]
- retbleed=off [X86]
- spec\_store\_bypass\_disable=off [X86,PPC]
- spectre\_v2\_user=off [X86]
- srbds=off [X86,INTEL]
- ssbd=force-off [ARM64]
- tsx\_async\_abort=off [X86]

**例外: `kvm.nx_huge_pages=force` の場合、これは `kvm.nx_huge_pages` には影響しません。**

- **auto** (デフォルト) - すべての CPU の脆弱性を軽減しますが、脆弱であっても SMT を有効のままにします。これは、カーネルのアップグレードで SMT が自動的に無効化されないようにしたいユーザーや、SMT ベースの攻撃を他の方法で回避できるユーザーを対象にしています。
- **auto, nosmt** - すべての CPU の脆弱性を軽減し、必要に応じて SMT を無効にします。これは、SMT が失われることを意味する場合でも、常に完全な軽減を求めるユーザーを対象としています。**auto**、**nosmt** オプションは以下と同等です。
  - l1tf=flush,nosmt [X86]
  - mds=full,nosmt [X86]
  - tsx\_async\_abort=full,nosmt [X86]
  - mmio\_stale\_data=full,nosmt [X86]
  - retbleed=auto,nosmt [X86]

### nomodeset

このカーネルパラメーターを使用すると、カーネル modesetting を無効にすることができます。ほとんどのシステムのファームウェアは表示モードをセットアップし、出力用のフレームバッファメモリを提供します。**nomodeset** を使用すると、事前初期化された出力を置き換える可能性がある場合、DRM ドライバーと **fbdev** ドライバーはロードされません。システムフレームバッファのみが使用可能になります。ドライバーは、表示モードの変更や高速レンダリングを実行しません。このパラメーターは、エラーのフォールバックとして、またはテストとデバッグに特に役立ちます。

### rdt=[HW,X86,RDT]

このカーネルパラメーターを使用すると、個々の RDT 機能をオンまたはオフにすることができます。

す。リストには、**cmt**、**mbmtotal**、**mbmlocal**、**l3cat**、**l3cdp**、**l2cat**、**l2cdp**、**mba**、**smba**、**bmec** が含まれません。たとえば、**cmt** をオンにして **mba** をオフにするには、次を使用します。

```
rdt=cmt,!mba
```

#### rodata=[KNL]

このカーネルパラメーターを使用すると、読み取り専用カーネルマッピングを無効にすることができます。利用可能なオプションは次のとおりです。

- **on** - 読み取り専用カーネルメモリーを読み取り専用としてマークします (デフォルト)。
- **off** - デバッグ用に読み取り専用カーネルメモリーを書き込み可能のままにします。
- **[新規] full** - 読み取り専用カーネルメモリーとエイリアスを読み取り専用としてマークします [arm64]。

#### 削除されたカーネルパラメーター

##### nobats=[PPC]

このカーネルパラメーターを使用すると、"クラシック" PPC コアでカーネル lowmem をマッピングするための BAT の使用を禁止できます。

##### noltlbs=[PPC]

このカーネルパラメーターを使用すると、PPC40x および PPC8xx でのカーネル lowmem マッピングで huge page および tlb エントリーの使用を禁止できます。

##### swapaccount=[O|1]=[KNL]

このカーネルパラメーターを使用すると、メモリーリソースコントローラーでのスワップのアカウンティングを有効または無効にできます。詳細は、[Documentation/admin-guide/cgroup-v1/memory.rst](#) を参照してください。

## 第6章 デバイスドライバー

### 6.1. 新しいドライバー

#### ネットワークドライバー

- MediaTek MT7601U (USB) サポート (**mt7601u**) で、MT7601U ベースのワイヤレス USB ドングルのサポートを追加 (64 ビット ARM アーキテクチャーのみ)
- MediaTek MT76x0E (PCIe) サポート (**mt76x0e**) で、MT7610/MT7630 ベースのワイヤレス PCIe デバイスのサポートを追加 (64 ビット ARM アーキテクチャーのみ)
- MediaTek MT76x0U (USB) サポート (**mt76x0u**) で、MT7610U ベースのワイヤレス USB 2.0 ドングルのサポートを追加 (64 ビット ARM アーキテクチャーのみ)
- MediaTek MT76x2E (PCIe) サポート (**mt76x2e**) で、MT7612/MT7602/MT7662 ベースのワイヤレス PCIe デバイスのサポートを追加 (64 ビット ARM アーキテクチャーのみ)
- MediaTek MT76x2U (USB) サポート (**mt76x2u**) で、MT7612U ベースのワイヤレス USB 3.0 ドングルのサポートを追加 (64 ビット ARM アーキテクチャーのみ)
- MediaTek MT7921E (PCIe) サポート (**mt7921e**) で、MT7921E 802.11ax 2x2:2SS ワイヤレスデバイスのサポートを追加 (64 ビット ARM アーキテクチャーのみ)
- Atheros ドライバー 802.11n HTC ベースのワイヤレスデバイス (**ath9k\_htc**) (64 ビット ARM アーキテクチャーのみ)
- Broadcom 802.11n ワイヤレス LAN ドライバー (**brcmsmac**) (64 ビット ARM アーキテクチャーのみ)
- Broadcom 802.11n ワイヤレス LAN ドライバーユーティリティ (**brcmutil**) (64 ビット ARM アーキテクチャーのみ)
- Broadcom 802.11 ワイヤレス LAN fullmac ドライバー (**brcmfmac**) (64 ビット ARM アーキテクチャーのみ)
- Qualcomm Atheros 802.11ac ワイヤレス LAN カード用のコアモジュール (**ath10k\_core**) (64 ビット ARM アーキテクチャーのみ)
- Qualcomm Atheros 802.11ax ワイヤレス LAN カード用のコアモジュール (**ath11k**) (64 ビット ARM アーキテクチャーのみ)
- WWAN フレームワークのデバイスシミュレーター (**wwan\_hwsim**)
- Qualcomm Atheros 802.11ac WLAN PCIe/AHB デバイスのドライバーサポート (**ath10k\_pci**) (64 ビット ARM アーキテクチャーのみ)
- Qualcomm Technologies 802.11ax WLAN PCIe デバイスのドライバーサポート (**ath11k\_pci**) (64 ビット ARM アーキテクチャーのみ)
- Linux 用 Intel® Wireless Wi-Fi ドライバー (**iwlwifi**) (64 ビット ARM アーキテクチャーのみ)
- Linux 用 Intel® Wireless Wi-Fi Link AGN ドライバー (**iwl\_dvm**) (64 ビット ARM アーキテクチャーのみ)
- IOSM Driver (**iosm**)

- Marvell WiFi-Ex Driver バージョン 1.0 (**mwifiex**) (64 ビット ARM アーキテクチャーのみ)
- Marvell WiFi-Ex PCI-Express Driver バージョン 1.0 (**mwifiex\_pcie**) (64 ビット ARM アーキテクチャーのみ)
- Marvell WiFi-Ex SDIO Driver バージョン 1.0 (**mwifiex\_sdio**) (64 ビット ARM アーキテクチャーのみ)
- Marvell WiFi-Ex USB Driver バージョン 1.0 (**mwifiex\_usb**) (64 ビット ARM アーキテクチャーのみ)
- MediaTek PCIe 5G WWAN モデム T7xx ドライバー (**mtk\_t7xx**)
- Network/MBIM over MHI (**mhi\_wwan\_mbim**) (64 ビット ARM アーキテクチャー、IBM Power Systems、リトルエンディアン、AMD および Intel 64 ビットアーキテクチャーのみ)
- rtlwifi 用の PCI 基本ドライバー (**rtl\_pci**) (64 ビット ARM アーキテクチャーのみ)
- Ralink RT2800 ライブラリーバージョン 2.3.0 (**rt2800lib**) (64 ビット ARM アーキテクチャーのみ)
- Ralink RT2800 PCI & PCMCIA ワイヤレス LAN ドライバーバージョン 2.3.0 (**rt2800pcie**) (64 ビット ARM アーキテクチャーのみ)
- Ralink RT2800 USB ワイヤレス LAN ドライバーバージョン 2.3.0 (**rt2800usb**) (64 ビット ARM アーキテクチャーのみ)
- Realtek 802.11ac ワイヤレス 8821c ドライバー (**rtw88\_8821c**) (64 ビット ARM アーキテクチャーのみ)
- Realtek 802.11ac ワイヤレス 8821ce ドライバー (**rtw88\_8821ce**) (64 ビット ARM アーキテクチャーのみ)
- Realtek 802.11ac ワイヤレス 8822b ドライバー (**rtw88\_8822b**) (64 ビット ARM アーキテクチャーのみ)
- Realtek 802.11ac ワイヤレス 8822be ドライバー (**rtw88\_8822be**) (64 ビット ARM アーキテクチャーのみ)
- Realtek 802.11ac ワイヤレス 8822c ドライバー (**rtw88\_8822c**) - (64 ビット ARM アーキテクチャーのみ)
- Realtek 802.11ac ワイヤレス 8822ce ドライバー (**rtw88\_8822ce**) (64 ビット ARM アーキテクチャーのみ)
- Realtek 802.11ac ワイヤレスコアモジュール (**rtw88\_core**) (64 ビット ARM アーキテクチャーのみ)
- Realtek 802.11ac ワイヤレス PCI ドライバー (**rtw88\_pcie**) (64 ビット ARM アーキテクチャーのみ)
- Realtek 802.11ax ワイヤレス 8852A ドライバー (**rtw89\_8852a**) (64 ビット ARM アーキテクチャーのみ)
- Realtek 802.11ax ワイヤレス 8852AE ドライバー (**rtw89\_8852ae**) (64 ビット ARM アーキテクチャーのみ)

- Realtek 802.11ax ワイヤレス 8852B ドライバー (**rtw89\_8852b**) (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のみ)
- Realtek 802.11ax ワイヤレス 8852BE ドライバー (**rtw89\_8852be**) (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のみ)
- Realtek 802.11ax ワイヤレスコアモジュール (**rtw89\_core**) (64 ビット ARM アーキテクチャーのみ)
- Realtek 802.11ax ワイヤレス PCI ドライバー (**rtw89\_pci**) (64 ビット ARM アーキテクチャーのみ)
- Realtek 802.11n PCI ワイヤレスコア (**btcoexist**) (64 ビット ARM アーキテクチャーのみ)
- Realtek 802.11n PCI ワイヤレスコア (**rtlwifi**) (64 ビット ARM アーキテクチャーのみ)
- Realtek 802.11n ワイヤレス 8723d ドライバー (**rtw88\_8723d**) (64 ビット ARM アーキテクチャーのみ)
- Realtek 802.11n ワイヤレス 8723de ドライバー (**rtw88\_8723de**) (64 ビット ARM アーキテクチャーのみ)
- Realtek 8188E 802.11n PCI ワイヤレス (**rtl8188ee**) (64 ビット ARM アーキテクチャーのみ)
- Realtek 8192C/8188C 802.11n PCI ワイヤレス (**rtl8192c-common**) (64 ビット ARM アーキテクチャーのみ)
- Realtek 8192C/8188C 802.11n PCI ワイヤレス (**rtl8192ce**) (64 ビット ARM アーキテクチャーのみ)
- Realtek 8192C/8188C 802.11n USB ワイヤレス (**rtl8192cu**) (64 ビット ARM アーキテクチャーのみ)
- Realtek 8192DE 802.11n Dual Mac PCI ワイヤレス (**rtl8192de**) (64 ビット ARM アーキテクチャーのみ)
- Realtek 8192EE 802.11n PCI ワイヤレス (**rtl8192ee**) (64 ビット ARM アーキテクチャーのみ)
- Realtek 8192S/8191S 802.11n PCI ワイヤレス (**rtl8192se**) (64 ビット ARM アーキテクチャーのみ)
- Realtek 8723BE 802.11n PCI ワイヤレス (**rtl8723be**) (64 ビット ARM アーキテクチャーのみ)
- Realtek 8723E 802.11n PCI ワイヤレス (**rtl8723ae**) (64 ビット ARM アーキテクチャーのみ)
- Realtek 8821ae 802.11ac PCI ワイヤレス (**rtl8821ae**) (64 ビット ARM アーキテクチャーのみ)
- Realtek RTL8723AE/RTL8723BE 802.11n PCI ワイヤレス共通ルーチン (**rtl8723-common**) (64 ビット ARM アーキテクチャーのみ)
- rt2800 MMIO ライブラリーバージョン 2.3.0 (**rt2800mmio**) (64 ビット ARM アーキテクチャーのみ)
- rt2x00 ライブラリーバージョン 2.3.0 (**rt2x00lib**) (64 ビット ARM アーキテクチャーのみ)
- rt2x00 mmio ライブラリーバージョン 2.3.0 (**rt2x00mmio**) (64 ビット ARM アーキテクチャーのみ)

- `rt2x00 pci` ライブラリーバージョン 2.3.0 (**rt2x00pci**) (64 ビット ARM アーキテクチャーのみ)
- `rt2x00 USB` ライブラリーバージョン 2.3.0 (**rt2x00usb**) (64 ビット ARM アーキテクチャーのみ)
- RTL8XXXu USB mac80211 Wireless LAN Driver (**rtl8xxxu**) (64 ビット ARM アーキテクチャーのみ)
- Atheros ワイヤレス 802.11n LAN カード用の共有ライブラリー (**ath9k\_common**) (64 ビット ARM アーキテクチャーのみ)
- Atheros ワイヤレス LAN カードの共有ライブラリー (**ath**) (64 ビット ARM アーキテクチャーのみ)
- Atheros 802.11n ワイヤレス LAN カードのサポート (**ath9k\_hw**) (64 ビット ARM アーキテクチャーのみ)
- Atheros 802.11n ワイヤレス LAN カードのサポート (**ath9k**) (64 ビット ARM アーキテクチャーのみ)
- Linux 用の新しい Intel® ワイヤレス AGN ドライバー (**iwlvm**) (64 ビット ARM アーキテクチャーのみ)
- Thunderbolt/USB4 ネットワークドライバー (**thunderbolt\_net**)
- rtlwifi 用の USB 基本ドライバー (**rtl\_usb**) (64 ビット ARM アーキテクチャーのみ)

### グラフィックドライバーとその他のドライバー

- Atheros AR30xx ファームウェアドライバー 1.0 (**ath3k**) (64 ビット ARM アーキテクチャーのみ)
- BlueFRITZ!USB ドライバーバージョン 1.2 (**bfusb**) (64 ビット ARM アーキテクチャーのみ)
- Bluetooth HCI UART ドライバーバージョン 2.3 (**hci\_uart**) (64 ビット ARM アーキテクチャーのみ)
- Broadcom デバイスバージョン 0.1 の Bluetooth サポート (**btbcm**) (64 ビット ARM アーキテクチャーのみ)
- Intel デバイスバージョン 0.1 の Bluetooth サポート (**btintel**) (64 ビット ARM アーキテクチャーのみ)
- MediaTek デバイスバージョン 0.1 の Bluetooth サポート (**btmtk**) (64 ビット ARM アーキテクチャーのみ)
- Realtek デバイスバージョン 0.1 の Bluetooth サポート (**btrtl**) (64 ビット ARM アーキテクチャーのみ)
- Bluetooth 仮想 HCI ドライバーバージョン 1.5 (**hci\_vhci**) (64 ビット ARM アーキテクチャーのみ)
- Broadcom Blutionium ファームウェアドライバーバージョン 1.2 (**bcm203x**) (64 ビット ARM アーキテクチャーのみ)
- Digianswer Bluetooth USB ドライバーバージョン 0.11 (**bpa10x**) (64 ビット ARM アーキテクチャーのみ)



- 汎用 Bluetooth SDIO ドライバーバージョン 0.1 (**btsdio**) (64 ビット ARM アーキテクチャーのみ)
- 汎用 Bluetooth USB ドライバーバージョン 0.8 (**btusb**) (64 ビット ARM アーキテクチャーのみ)
- Marvell Bluetooth ドライバーバージョン 1.0 (**btmrvl**) (64 ビット ARM アーキテクチャーのみ)
- Marvell BT-over-SDIO ドライバーバージョン 1.0 (**btmrvl\_sdio**) (64 ビット ARM アーキテクチャーのみ)
- BMC IPMI SSIF インターフェイスの Linux デバイスドライバー (**ssif\_bmc**) (64 ビット ARM アーキテクチャーのみ)
- vTPM Driver バージョン 0.1 (**tpm\_vtpm\_proxy**)
- AMD P-state ドライバーテストモジュール (**amd-pstate-ut**) (AMD および Intel 64 ビットアーキテクチャーのみ)
- Compute Express Link (CXL) ACPI ドライバー (**cxl\_acpi**) (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のみ)
- Compute Express Link (CXL) コアドライバー (**cxl\_core**)
- Compute Express Link (CXL) ポートドライバー (**cxl\_port**)
- NVIDIA Tegra GPC DMA Controller ドライバー (**tegra186-gpc-dma**) (64 ビット ARM アーキテクチャーのみ)
- DRM Buddy Allocator (**drm\_buddy**) (64 ビット IBM Z アーキテクチャーのみ)
- DRM ディスプレイアダプターヘルパー (**drm\_display\_helper**) (64 ビット IBM Z アーキテクチャーのみ)
- EVision デバイス用の HID ドライバー (**hid-evision**) (64 ビット ARM アーキテクチャー、IBM Power Systems、リトルエンディアン、AMD および Intel 64 ビットアーキテクチャーのみ)
- Texas Instruments INA3221 HWMon Driver (**ina3221**) (64 ビット ARM アーキテクチャーのみ)
- I3C コア (**i3c**) (64 ビット ARM アーキテクチャーのみ)
- Silvaco デュアルロール I3C マスタードライバー (**svc-i3c-master**) (64 ビット ARM アーキテクチャーのみ)
- Microsoft Azure Network Adapter IB ドライバー (**mana\_ib**) (AMD および Intel 64 ビットアーキテクチャーのみ)
- Soft RDMA トランスポート (**rdma\_rxe**)
- i.MX8MP インターコネクトドライバー - i.MX SOC 用の汎用インターコネクトドライバー (**imx8mp-interconnect**) (64 ビット ARM アーキテクチャーのみ)
- Linux USB Video Class (**uvc**) (64 ビット ARM アーキテクチャー、IBM Power Systems、リトルエンディアン、AMD および Intel 64 ビットアーキテクチャーのみ)
- videobuf2 の共通メモリー処理ルーチン (**videobuf2-memops**) (64 ビット ARM アーキテクチャーのみ)

- cec ドライバーのデバイスノード登録 (**cec**) (64 ビット IBM Z アーキテクチャーのみ)
- メディアドライバーのデバイスノード登録 (**mc**) (64 ビット ARM アーキテクチャーのみ)
- Video for Linux 2 のドライバーヘルパーフレームワーク (**videobuf2-v4l2**) (64 ビット ARM アーキテクチャーのみ)
- メディアバッファコアフレームワーク (**videobuf2-common**) (64 ビット ARM アーキテクチャーのみ)
- USB Video Class ドライバーバージョン 1.1.1 (**uvcvideo**) (64 ビット ARM アーキテクチャーのみ)
- V4L2 DV Timings Helper Functions (**v4l2-dv-timings**) (64 ビット ARM アーキテクチャーのみ)
- Video4Linux2 コアドライバー (**videodev**) (64 ビット ARM アーキテクチャーのみ)
- videobuf2 の vmalloc メモリー処理ルーチン (**videobuf2-vmalloc**) (64 ビット ARM アーキテクチャーのみ)
- SPI NOR のフレームワーク (**spi-nor**) (64 ビット ARM アーキテクチャーのみ)
- Marvell CN10K DRAM サブシステム (DSS) PMU (**marvell\_cn10k\_dds\_pmu**) (64 ビット ARM アーキテクチャーのみ)
- Marvell CN10K LLC-TAD Perf ドライバー (**marvell\_cn10k\_tad\_pmu**) (64 ビット ARM アーキテクチャーのみ)
- Intel Meteor Lake PCH pinctrl/GPIO ドライバー (**pinctrl-meteorlake**) (AMD および Intel 64 ビットアーキテクチャーのみ)
- Intel In Field Scan (IFS) デバイス (**intel\_ifs**) (AMD および Intel 64 ビットアーキテクチャーのみ)
- NVIDIA WMI EC Backlight ドライバー (**nvidia-wmi-ec-backlight**) (AMD および Intel 64 ビットアーキテクチャーのみ)
- QMI エンコーダー/デコーダーヘルパー (**qmi\_helpers**) (64 ビット ARM アーキテクチャーのみ)
- AMD SoundWire ドライバー (**soundwire-amd**) (AMD および Intel 64 ビットアーキテクチャーのみ)
- NVIDIA Tegra114 SPI Controller Driver (**spi-tegra114**) (64 ビット ARM アーキテクチャーのみ)
- STMicroelectronics STUSB160x Type-C コントローラードライバー (**stusb160x**) (64 ビット ARM アーキテクチャーのみ)
- MLX5 VFIO PCI - MLX5 デバイスファミリー用のユーザーレベルのメタドライバー (**mlx5-vfio-pci**)

## 6.2. 更新されたドライバー

### ネットワークドライバーの更新

- Realtek RTL8152/RTL8153 ベースの USB イーサネットアダプター (**r8152**) がバージョン v1.12.13 に更新されました (64 ビット ARM アーキテクチャー、IBM Power Systems、リトルエンディアン、AMD および Intel 64 ビットアーキテクチャーのみ)

ノアイアン、AMD および Intel 64 ビットアーキテクチャーのみ)。

## ストレージドライバーの更新

- Broadcom MegaRAID SAS Driver (**megaraid\_sas**) がバージョン 07.725.01.00-rc1 に更新されました (64 ビット ARM アーキテクチャー、IBM Power Systems、リトルエンディアン、AMD および Intel 64 ビットアーキテクチャーのみ)。
- Microchip Smart Family Controller のドライバー (**smartpqi**) がバージョン 2.1.22-040 に更新されました (64 ビット ARM アーキテクチャー、IBM Power Systems、リトルエンディアン、AMD および Intel 64 ビットアーキテクチャーのみ)。
- Emulex LightPulse Fibre Channel SCSI ドライバー (**lpfc**) がバージョン 0:14.2.0.12 に更新されました (64 ビット ARM アーキテクチャー、IBM Power Systems、リトルエンディアン、AMD および Intel 64 ビットアーキテクチャーのみ)。
- MPI3 Storage Controller Device Driver (**mpi3mr**) がバージョン 8.4.1.0.0 に更新されました。

## 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 拡張機能 v2	available
ISA 拡張機能 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_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_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_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

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_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_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_tstamp, 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_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_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_strncmp, bpf_skb_set_tstamp, 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_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
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_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_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_strncmp, 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_strtoull, 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_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_strtoull, 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_ktime_get_boot_ns, bpf_skb_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_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_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_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_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_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_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_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_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_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_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_strotol, bpf_strtoull, 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_strotol, bpf_strtoull, 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_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_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_sk_select_reuseport, 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_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_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_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_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_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_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
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	サポート対象外
struct_ops	サポート対象外
ext	サポート対象外
lsm	サポート対象外

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_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_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_cgroup, bpf_get_numa_node_id, bpf_probe_read_str, bpf_get_socket_cookie, 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_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_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_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

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
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

マップの種類	Available
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 9.3 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

インストールプログラムは、`url` キックスタートコマンドの `--proxy` オプションを正しく処理するようになる

以前は、インストールプログラムは `url` キックスタートコマンドの `--proxy` オプションを正しく処理していませんでした。その結果、指定されたプロキシを使用してインストールイメージを取得できませんでした。この更新により、この問題は修正され、`--proxy` オプションが期待どおりに機能するようになりました。

[Bugzilla:2177219](#)

`liveimg` の `--noverifyssl` オプションは、HTTPS を使用してダウンロードされたイメージのサーバーの証明書をチェックしなくなる

以前は、インストールプログラムは `liveimg` キックスタートコマンドの `--noverifyssl` オプションを無視していました。したがって、HTTPS プロトコルを使用してダウンロードされたイメージに対してサーバーの証明書を検証できない場合、インストールプロセスは失敗しました。今回の更新により、この問題は修正され、`liveimg` キックスタートコマンドの `--noverifyssl` オプションが正しく機能するようになりました。

[Bugzilla:2157921](#)

Anaconda が、FIPS 要件の LUKS パスフレーズを検証するようになる

以前は、基礎となるツールがこのチェックを実行していても、Anaconda は LUKS パスフレーズの長さが FIPS 要件を満たしているかどうかをチェックしていませんでした。これにより、パスフレーズが 8 文字より短い FIPS モードでインストールすることで、インストーラーが途中で停止していました。

この更新により、パスフレーズの最小長を検証して強制するようにインストールプログラムが改善されました。その結果、インストールプログラムは、LUKS パスフレーズが FIPS モードで使用するには短すぎるかどうかを通知し、予期しない停止を防ぎます。

[Bugzilla:2163497](#)

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

OpenSSL コマンド `cms` および `smime` は、FIPS モードでファイルを暗号化できます

以前は、`cms` および `smime` OpenSSL コマンドのデフォルト設定では、3DES や PKCS #1 v1.5 などのレガシー暗号化アルゴリズムが使用されていました。これらのアルゴリズムは FIPS モードでは無効になります。その結果、FIPS モードのシステムでは、デフォルト設定で `smime` コマンドを使用してファ

イルを暗号化できませんでした。この更新では次の変更が導入されています。

- FIPS モードでは、OpenSSL API はデフォルトで OAEP と RSA キーを使用して CMS データを作成します。
- FIPS モードでは、RSA キーが指定された場合、**cms** OpenSSL コマンドは **aes-128-cbc** および OAEP で暗号化された CMS ファイルを作成します。

ECDSA キーの使用は影響を受けません。非 FIPS モードでは、OpenSSL API と **cms** コマンドは、デフォルトで PKCS#1 v1.5 パディングと 3DES 暗号化を引き続き使用します。

その結果、FIPS モードで **cms** および **smime** OpenSSL コマンドを使用して、ファイルを暗号化できません。

[Bugzilla:2160797](#)

### SELinux で Dovecot でメールのレプリケーションが可能になる

双方向レプリケーションセットを使用して Dovecot 高性能メール配信エージェントを高可用性向けに設定できますが、以前の SELinux ポリシーには、ランタイムファイルシステムのパイプを介して通信する **dovecot-deliver** ユーティリティーに関するルールが含まれていませんでした。その結果、Dovecot でのメールのレプリケーションは機能しませんでした。この更新により、SELinux ポリシーにパーミッションが追加され、その結果、Dovecot でのメールレプリケーションが機能するようになりました。

[Bugzilla:2170495<sup>\[1\]</sup>](#)

### NFS ファイルシステムからのブートは、enforcing モードに設定された SELinux で機能するようになる

以前は、NFS をルートファイルシステムとして使用すると、SELinux ラベルがサーバーから転送されず、SELinux が enforcing モードに設定されているときに起動エラーが発生していました。

この修正により、SELinux は、最初の SELinux ポリシーのロード前に作成された NFS マウントにサポートセキュリティラベルとして正しくフラグを立てるように修正されました。その結果、NFS マウントはサーバーとクライアントの間で SELinux ラベルを転送するようになり、enforcing モードに設定された SELinux でブートを成功させることができます。

[Bugzilla:2218207<sup>\[1\]</sup>](#)

### rabbitmq が IPv6 で失敗しなくなる

以前は、IPv6 を有効にして **rabbitmq** サーバーをデプロイすると、**inet\_gethost** コマンドは **/proc/sys/net/ipv6/conf/all/disable\_ipv6** ファイルにアクセスしようとしていました。その結果、システムは **/proc/sys/net/ipv6/conf/all/disable\_ipv6** へのアクセスを拒否しました。この更新により、システムは **/proc/sys/net/ipv6/conf/all/disable\_ipv6** を読み取ることができるようになり、**rabbitmq** が IPv6 で動作するようになりました。

[Bugzilla:2184999](#)

### cloud-init を介した Insights への登録が SELinux によってブロックされなくなる

以前は、SELinux ポリシーには、**cloud-init** スクリプトによる **Insights-client** サービスの実行を許可するルールが含まれていませんでした。その結果、**cloud-init** スクリプトが **insights-client --register** コマンドの実行を試みると失敗していました。この更新により、欠落していたルールがポリシーに追加され、SELinux を enforcing モードで使用して **cloud-init** を介して Insights に登録できるようになりました。

[Bugzilla:2162663](#)



## staff\_r SELinux ロールのユーザーは scap\_workbench プローブを実行できるようになる

以前は、**selinux-policy** パッケージには、**scap-workbench** ユーティリティを実行するために必要な **staff\_r** SELinux ロールのユーザーに対するルールが含まれていませんでした。その結果、**staff\_r** SELinux ロールのユーザーが実行すると、**scap-workbench** プローブが失敗していました。この更新により、欠落していたルールが **selinux-policy** に追加され、SELinux ユーザーは **scap\_workbench** プローブを実行できるようになりました。

[Bugzilla:2112729](#)

## SELinux ポリシーに insights-client のパーミッションが追加されました

**Insights-client** サービスには、以前のバージョンの **selinux-policy** にはなかった権限が必要です。そのため、**insights-client** の一部のコンポーネントが正しく機能せず、アクセスベクターキャッシュ (AVC) エラーメッセージを報告していました。今回の更新で、SELinux ポリシーに新しいパーミッションが追加されました。その結果、AVC エラーを報告せずに **insights-client** が正常に実行されます。

Jira:RHELPLAN-163014<sup>[1]</sup>、[Bugzilla:2190178](#)、[Bugzilla:2224737](#)、[Bugzilla:2207894](#)、[Bugzilla:2214581](#)

## Keylime 許可リスト生成スクリプトが更新される

Keylime スクリプト **create\_allowlist.sh** は、Keylime ポリシーの許可リストを生成します。RHEL 9.3 では、**create\_runtime\_policy.sh** スクリプトに置き換えられました。このスクリプトは、許可リストを JSON ランタイムポリシーに変換しようとする失敗していました。

今回の更新により、スクリプトは **create\_allowlist.sh** に戻りました。**keylime\_create\_policy** スクリプトを使用して、許可リストと除外リストを JSON ランタイムポリシーに組み合わせることができるようになりました。

Jira:RHEL-11866<sup>[1]</sup>

## Keylime が tls\_dir = default で特定のファイルを必要としなくなる

以前は、Keylime verifier または Keylime registrar 設定で **tls\_dir** 変数が **default** に設定されていた場合、Keylime は、ファイル名が **cacert.crt** ではないカスタムの認証局 (CA) 証明書を拒否していました。今回の更新によってこの問題は発生しなくなり、**tls\_dir = default** と設定されていてもカスタムの CA 証明書ファイルを使用できるようになりました。

Jira:RHELPLAN-157337<sup>[1]</sup>

## 環境変数が、Keylime エージェントのオプションをアンダースコアでオーバーライド可能に

以前は、Keylime エージェント設定オプション名にアンダースコア ( \_ ) が含まれている場合、環境変数を使用してこのオプションをオーバーライドすることはできませんでした。今回の更新により、オプション名にアンダースコアが含まれている場合でも、環境変数によるオーバーライドが正しく機能するようになりました。

Jira:RHEL-395<sup>[1]</sup>

## Keylime registrar は IPv6 アドレスを正しく識別する

以前は、Keylime registrar は IPv6 アドレスを正しく認識できなかったため、リスニングポートをバインドできませんでした。この更新により、Keylime registrar は IPv6 アドレスを適切に識別し、その結果、そのポートに正しくバインドされるようになりました。

Jira:RHEL-392<sup>[1]</sup>

## Keylime エージェントが IPv6 アドレスを正しく処理する

以前は、括弧 [ ] で囲まれていない IPv6 アドレスを使用して Keylime エージェントを登録すると、**keylime\_tenant** ユーティリティーがエラーで失敗していました。今回の更新により、**keylime\_tenant** は、IPv6 アドレスが括弧で囲まれていない場合でも、正しく処理できるようになりました。

[Jira:RHEL-393<sup>\[1\]</sup>](#)

### QEMU VM の新しいイベントにより、Keylime は測定ブート認証に失敗しなくなる

**edk2-ovmf** パッケージの更新により、QEMU によって操作される仮想システムの測定ブートログに新しいタイプのイベントが導入されました。これらのイベントにより、Keylime の測定ブート認証に障害が発生しました。今回の更新により、Keylime はこれらのイベントを正しく処理します。

[Jira:RHEL-947<sup>\[1\]</sup>](#)

### Keylime Webhook Notifier が TLS セッションを正しく終了する

以前は、Keylime Webhook Notifier は TLS セッションを正しく終了しませんでした。これにより、リスナー側で警告が報告されました。この更新によりこの問題が修正され、Webhook Notifier が TLS セッションを正しく終了するようになりました。

[Jira:RHEL-1252<sup>\[1\]</sup>](#)

### gpg-agent が FIPS モードで SSH エージェントとして機能するようになる

以前は、FIPS モードで MD5 ダイジェストが無効になっている場合でも、**gpg-agent** ツールは **ssh-agent** プログラムにキーを追加するときに MD5 フィンガープリントを作成していました。その結果、**ssh-add** ユーティリティーが認証エージェントへのキーの追加に失敗していました。

このリリースでは、**gpg-agent** は MD5 チェックサムを使用しなくなりました。その結果、**gpg-agent** は FIPS モードで実行されているシステムでも SSH 認証エージェントとして機能するようになりました。

[Bugzilla:2073567](#)

### tangd-keygen がデフォルト以外の umask を正しく処理するようになる

以前は、**tangd-keygen** スクリプトは、生成されたキーファイルのファイル権限を変更しませんでした。その結果、他のユーザーへのキーの読み取りを防止するデフォルトのユーザーファイル作成モードマスク (**umask**) が設定されているシステムでは、**tang-show-keys** コマンドはキーを表示する代わりにエラーメッセージ **Internal Error 500** を返していました。今回の更新により、**tangd-keygen** は、生成されたキーファイルのファイルパーミッションを設定するため、スクリプトはデフォルト以外の **umask** を使用するシステムで正しく動作するようになりました。

[Bugzilla:2188743](#)

### fapolicyd サービスが、信頼されたデータベースから削除されたプログラムを実行しなくなる

以前は、**fapolicyd** サービスは、プログラムが信頼されたデータベースから削除された後でも、プログラムを信頼できるものとして誤って処理していました。その結果、**fapolicyd-cli --update** コマンドを入力しても効果はなく、プログラムは削除された後でも実行される可能性があります。この更新により、**fapolicyd-cli --update** コマンドによって信頼されたプログラムデータベースが正しく更新され、削除されたプログラムは実行できなくなりました。

[Jira:RHEL-622](#)

### fapolicyd により、mount および umount 後にシステムがハングしなくなる

以前は、**mount** または **umount** アクションが2回実行され、その後に **fapolicyd-cli --update** コマンドが実行されると、**fapolicyd** サービスが無限ループに入る可能性があります。その結果、システムが応答を停止しました。この更新により、サービスは **fapolicyd-cli --update** コマンドを正しく実行し、任意の数の **mount** または **umount** アクションを処理できるようになりました。

[Jira:RHEL-817](#)

### Keylime は連結された PEM 証明書を受け入れるようになる

以前は、Keylime が1つ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 コンポーネントによって TLS ハンドシェイクの失敗が発生していました。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。この更新により、**keylime-agent-rust** は、中間 CA 証明書を含む複数の証明書を正しく処理できるようになりました。その結果、連結された PEM 証明書を Keylime で使用できるようになりました。

[Jira:RHEL-396<sup>\[1\]</sup>](#)

### Rsyslog は capabilities がなくても起動可能

Rsyslog が通常のユーザーとして実行される場合、またはコンテナ化された環境で実行される場合、**rsyslog** プロセスには capabilities がありません。その結果、Rsyslog は capabilities をドロップできず、起動時に終了しました。今回の更新により、プロセスは、capabilities がない場合に capabilities をドロップしようとしなくなりました。その結果、Rsyslog は capabilities がない場合でも起動できます。

[Jira:RHELPLAN-160541<sup>\[1\]</sup>](#)

### io\_uring は SELinux 拒否なしで動作するようになる

以前は、**io\_uring** カーネルインターフェイスには SELinux ポリシーの **map** パーミッションがありませんでした。その結果、**mmap** システムコールが失敗し、**io\_uring** インターフェイスが正しく動作しませんでした。この更新により、SELinux ポリシーで **map** パーミッションが許可され、インターフェイスは SELinux 拒否なしで動作するようになりました。

[Bugzilla:2187745](#)

### oscap-anaconda-addon で CIS のネットワークサーバーを強化できるようになる

以前は、Network Servers パッケージグループが選択されている場合、CIS セキュリティプロファイル (**cis**、**cis\_server\_l1**、**cis\_workstation\_l1**、または **cis\_workstation\_l2**) を使用して RHEL Network Servers をインストールすることはできませんでした。この問題は、RHEL 9.3 で提供される **oscap-anaconda-addon-2.0.0-17.el9** の **ttft** パッケージを除外することで修正されます。その結果、Network Servers パッケージグループを使用して、CIS で強化された RHEL Network Servers をインストールできます。

[Bugzilla:2172264](#)

### ホームディレクトリーをチェックするルールはローカルユーザーにのみ適用される

**scap-security-guide** パッケージによって提供される複数のコンプライアンスプロファイルには、ユーザーのホームディレクトリーの正しい設定をチェックする次のルールが含まれています。

- **accounts\_umask\_interactive\_users**
- **accounts\_user\_dot\_group\_ownership**
- **accounts\_user\_dot\_user\_ownership**

- **accounts\_user\_interactive\_home\_directory\_exists**
- **accounts\_users\_home\_files\_groupownership**
- **accounts\_users\_home\_files\_ownership**
- **accounts\_users\_home\_files\_permissions**
- **file\_groupownership\_home\_directories**
- **file\_ownership\_home\_directories**
- **file\_permissions\_home\_directories**

これらのルールは、ローカルユーザーの設定を正しくチェックします。以前は、修復スクリプトがリモートユーザーの設定を変更できない場合でも、スキャナーはNSSなどのネットワークソースから提供されたりリモートユーザーの設定を誤ってチェックしていました。これは、OpenSCAP スキャナーが以前に **getpwent()** システムコールを使用していたためです。この更新により、これらのルールの内部実装が **/etc/passwd** ファイルのデータだけに依存するように変更されます。その結果、ルールはローカルユーザーの設定にのみ適用されるようになりました。

[Bugzilla:2203791](#)

### パスワードの有効期間に関するルールはローカルユーザーにのみ適用される

CIS や DISA STIG などの一部のコンプライアンスプロファイルには、ユーザーアカウントのパスワードの有効期間とパスワードの有効期限をチェックする次のルールが含まれています。

- **accounts\_password\_set\_max\_life\_existing**
- **accounts\_password\_set\_min\_life\_existing**
- **accounts\_password\_set\_warn\_age\_existing**
- **accounts\_set\_post\_pw\_existing**

これらのルールは、ローカルユーザーの設定を正しくチェックします。以前は、修復スクリプトがリモートユーザーの設定を変更できない場合でも、スキャナーはNSSなどのネットワークソースから提供されたりリモートユーザーの設定を誤ってチェックしていました。これは、OpenSCAP スキャナーが以前に **getpwent()** システムコールを使用していたためです。

この更新により、これらのルールの内部実装が **/etc/shadow** ファイルのデータだけに依存するように変更されます。その結果、ルールはローカルユーザーの設定にのみ適用されるようになりました。

[Bugzilla:2213958](#)

### Red Hat CVE フィードが更新される

<https://access.redhat.com/security/data/oval/> にある Red Hat Common Vulnerabilities and Exposures (CVE) フィードのバージョン1は廃止され、<https://access.redhat.com/security/data/oval/v2/> にある CVE フィードのバージョン2に置き換えられました。

その結果、**scap-security-guide** パッケージによって提供される SCAP ソースデータストリーム内のリンクは、Red Hat CVE フィードの新しいバージョンにリンクするように更新されました。

[Bugzilla:2223178](#)

### journald 設定に関連するルールで余分な引用符が追加されなくなる

以前は、SCAP セキュリティガイドのルール `journald_compress`、`journald_forward_to_syslog`、および `journald_storage` の修復スクリプトにバグが含まれており、`/etc/systemd/journald.conf` 設定ファイルの設定オプションに余分な引用符が追加されていました。その結果、`journald` システムサービスは設定オプションの解析に失敗し、それらを無視しました。したがって、設定オプションは有効ではありませんでした。これにより、OpenSCAP スキャンの結果が誤った結果である `pass` となりました。この更新により、ルールと修復スクリプトで余分な引用符が追加されなくなりました。その結果、これらのルールは、`journald` の有効な設定を生成するようになりました。

[Bugzilla:2193169](#)

### `/var/lib/fdo` 配下のファイルに正しい SELinux ラベルが付けられるようになる

以前は、FDO プロセスがホスト全体にアクセスできるというセキュリティの問題がありました。今回の更新により、SELinux で `service-info-api` サーバーを使用することで、`/var/lib/fdo` ディレクトリー配下のデバイスに送信するファイルを追加できるようになり、その結果、`/var/lib/fdo` 配下のファイルは正しい SELinux ラベルを取得するようになりました。

[Bugzilla:2229722](#)

## 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<sup>\[1\]</sup>](#)

## 8.4. ソフトウェア管理

### `dnf need-restarting -s` コマンドで `systemd` サービスのリストが正しく表示されるようになる

以前は、`needs-restarting` コマンドを `-s` または `--services` オプションとともに使用すると、`systemd` 以外のプロセスまたは誤動作しているプロセスが検出されたときにエラーが発生していました。今回の更新により、`dnf need-restarting -s` コマンドはそのようなプロセスを無視し、代わりに影響を受ける `systemd` サービスのリストとともに警告を表示します。

[Bugzilla:2203100](#)

### `dnf-automatic` コマンドがトランザクションの終了ステータスを正しく報告するようになる

以前は、`dnf-automatic` コマンドは、トランザクション中の一部のアクションが正常に完了しなかった場合でも、トランザクションの成功終了コードを返しました。これにより、エラータの自動デプロイメントに `dnf-automatic` を使用するマシンでセキュリティリスクが発生する可能性があります。今回の更新により、この問題は修正され、`dnf-automatic` は、トランザクション中にパッケージに関するすべての問題を報告するようになりました。

[Bugzilla:2212262](#)

## 拡張ファイル属性のないファイルシステムに IMA 署名を持つパッケージをインストールしても失敗しなくなる

以前は、RPM は、ファイルが IMA 署名をサポートしていない場合でも、ファイルに IMA 署名を適用しようとしていました。その結果、パッケージのインストールは失敗しました。今回の更新により、RPM は IMA 署名の適用をスキップします。その結果、パッケージのインストールは失敗しなくなりました。

[Bugzilla:2157836](#)

## 8.5. シェルおよびコマンドラインツール

### rsyslog ログサービスがレスキューシステムの起動時に開始されるようになる

以前は、メッセージログ用の **rsyslog** サービスがレスキューシステムで自動的に起動されませんでした。**/dev/log** ソケットは、回復プロセス中にメッセージを受信し続けましたが、このソケットでリッスンするサービスはありませんでした。その結果、**/dev/log** ソケットがメッセージでいっぱいになり、回復プロセスが停止する原因となりました。たとえば、GRUB 設定を再生成する **grub2-mkconfig** コマンドは、マウントされたファイルシステムの数に応じて大量のログメッセージを生成します。ReaR を使用して、多数のファイルシステムがマウントされたシステムをリカバリーすると、多数のログメッセージが **/dev/log** ソケットを埋め尽くし、リカバリープロセスがフリーズしました。

この修正により、レスキューシステムの **systemd** ユニットには、ブート手順にソケットターゲットが含まれるようになり、ブート時にログインソケットが開始するようになりました。その結果、必要に応じて **rsyslog** サービスがレスキュー環境で起動され、リカバリー中にメッセージをログに記録する必要があるプロセスが停止することがなくなります。リカバリープロセスは正常に完了し、レスキュー RAM ディスクの **/var/log/messages** ファイルにログメッセージが表示されます。

[Bugzilla:2172912](#)

### which コマンドが長いパスでも失敗しなくなる

以前は、パスが 256 文字を超えるディレクトリーで **which** コマンドを実行すると、**Can't get current working directory** というエラーメッセージが表示されてコマンドが失敗していました。この修正により、**which** コマンドはパスの長さの制限として **PATH\_MAX** 値を使用するようになりました。その結果、コマンドは失敗しなくなりました。

[Bugzilla:2181974](#)

### ReaR は OUTPUT=USB で UEFI セキュアブートをサポートするようになる

以前は、ブータブルディスクドライブにレスキューイメージを保存する **OUTPUT=USB** ReaR 出力メソッドは、**SECURE\_BOOT\_BOOTLOADER** 設定を尊重していませんでした。その結果、UEFI セキュアブートが有効になっているシステムでは、ブートローダーが署名されていないため、レスキューイメージを含むディスクは起動しませんでした。

この修正により、**OUTPUT=USB** ReaR 出力メソッドは、レスキューディスクの作成時に **SECURE\_BOOT\_BOOTLOADER** 設定で指定したブートローダーを使用するようになりました。署名付き UEFI shim ブートローダーを使用するには、**/etc/rear/local.conf** ファイルで次の設定を変更します。

```
SECURE_BOOT_BOOTLOADER=/boot/efi/EFI/redhat/shimx64.efi
```

この結果、UEFI セキュアブートが有効になっている場合、レスキューディスクは起動可能になります。セキュアブートが有効になっていない場合でも、UEFI を備えたすべてのシステムで変数をこの値に安全に設定できます。一貫性を保つためにも推奨されます。UEFI ブート手順と shim ブートローダーの詳細は、[UEFI: what happens when booting the system](#) を参照してください。

[Bugzilla:2196445](#)

**ReaR** によってリカバリーされたシステムは、すべての VG 論理ボリュームのマウントに失敗しなくなる

`/etc/lvm/devices/system.devices` ファイルは、Logical Volume Manager (LVM) システムデバイスを表し、LVM に対するデバイスの可視性と使用可能性を制御します。デフォルトでは、RHEL 9 では **system.devices** 機能が有効になっており、アクティブになると、この機能は LVM デバイスフィルターを置き換えます。

以前は、ReaR を使用して、元のシステムが使用していたハードウェア ID とは異なるハードウェア ID を持つディスクにシステムをリカバリーすると、リカバリーされたシステムがすべての LVM ボリュームを検出できず、起動に失敗していました。この修正により、ReaR が **system.devices** ファイルを見つけた場合、ReaR はこのファイルをリカバリーの最後にある

`/etc/lvm/devices/system.devices.rearbak` に移動します。その結果、復元されたシステムはデバイスの可視性を制限するために LVM デバイスファイルを使用せず、システムは起動時に復元されたボリュームを検出します。

オプション: デフォルトの動作を復元して LVM デバイスファイルを再生成する場合は、リカバリープロセスの前にディスクを切断した場合に備えて、リカバリーされたシステムを起動し、通常の操作に必要なすべてのディスクデバイスを接続した後、**vgimportdevices -a** コマンドを使用します。

[Bugzilla:2145014](#)

## 8.6. ネットワーク

**Intel Corporation I350 Gigabit Fiber Network Connection** は、カーネル更新後にリンクを提供するようになる

以前は、External Thermal Sensor (ETS) のない Small Formfactor Pluggable (SFP) トランシーバーモジュールを使用したハードウェア設定により、**igb** ドライバーが ETS を読み取るために Inter-Integrated Circuit (I2C) を誤って初期化していました。その結果、接続はリンクを取得できませんでした。このバグ修正により、**igb** ドライバーは、ETS を備えた SFP が使用可能な場合にのみ I2C を初期化します。その結果、接続はリンクを取得します。

[Bugzilla:2173594<sup>\[1\]</sup>](#)

**nm-cloud-setup** サービスは、手動で設定されたセカンダリー IP アドレスをインターフェイスから削除しなくなる

クラウド環境から受け取った情報に基づいて、**nm-cloud-setup** サービスがネットワークインターフェイスを設定します。手動インターフェイス設定のために **nm-cloud-setup** を無効にするオプションもありましたが、特定のシナリオでは競合が発生しました。場合によっては、ホスト上の他のサービスが、セカンダリー IP アドレスの追加など、インターフェイスを独立して設定することがあります。**nm-cloud-setup** は、**systemd** タイマーユニットによって再度トリガーされたときに、これらのセカンダリー IP アドレスを誤って削除しました。**NetworkManager** パッケージのこの更新により、問題が修正されます。**systemd** タイマーユニットが **nm-cloud-setup** をトリガーするのを待つだけで済みます。タイマーを待ちたくない場合は、次のコマンドを使用して **nm-cloud-setup** を手動で有効にすることができます。

```
# systemctl enable nm-cloud-setup.service
```

その結果、**nm-cloud-setup** は、手動で設定されたセカンダリー IP アドレスをインターフェイスから削除しなくなりました。

[Bugzilla:2151040](#)

## 8.7. カーネル

### 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<sup>[1]</sup>

## 8.8. ブートローダー

### GRUB は、非デバッグカーネルバリエーションを正しく処理するようになる

以前は、複数のカーネル RPM がインストールされているシステムで、`dnf install kernel-$VERSION` または `dnf update` コマンドを入力すると、最後にインストールされたカーネルがデフォルトのカーネルとして設定されていました。これは、たとえば、AMD および Intel 64 ビットアーキテクチャー上の標準カーネルとリアルタイムカーネル、または 64 ビット ARM アーキテクチャー上のカーネル (4k) および `kernel-64k` を備えたシステムで発生しました。その結果、システムは将来的な再起動時に不要なカーネルを起動する可能性があります。この更新により、GRUB は `/etc/sysconfig/kernel` 設定ファイル内の `DEFAULTKERNEL` 変数を使用するようになり、デフォルトのカーネルは適切なバリエーションおよび最新バージョンのままになります。

詳細は、ナレッジベースのソリューション記事 [Changing the default kernel in Red Hat Enterprise Linux 8 & 9](#) を参照してください。

Bugzilla:2184069<sup>[1]</sup>

## 8.9. ファイルシステムおよびストレージ

### lpfc ドライバーが D\_ID ポートスワップ中に有効な状態にある

以前は、SAN ブートホストが NetApp ギブバックオペレーションを発行した後、LVM ハングタスクの警告とストールした I/O を引き起こしていました。この問題は、ファイバーチャネル `D_ID` ポートスワップが原因で、DM-Multipath 環境で代替パスが利用可能であった場合でも発生しました。競合状態の結果として、`D_ID` ポートスワップにより `lpfc` ドライバーで不整合な状態が発生し、I/O の発行が妨げられました。

今回の修正により、`lpfc` ドライバーは、`D_ID` ポートスワップが発生すると、必ず有効な状態になるようになりました。その結果、ファイバーチャネル `D_ID` ポートスワップにより、ハングした I/O が発生しなくなりました。

Bugzilla:2173947<sup>[1]</sup>

### multipathd が永続予約登録キーをすべてのパスに追加する

以前は、`multipathd` デーモンが起動して、既存のマルチパスデバイスの1つのパス上の永続予約登録キーを認識したとき、そのデバイスのすべてのパスに登録キーがあるわけではありませんでした。その結果、`multipathd` が停止している間に永続予約を持つマルチパスデバイスに新しいパスが現れた場合、それらのパスには永続予約はセットアップされませんでした。これにより、予約キーによって禁止されているはずのパス上での I/O 処理が可能になりました。

この修正により、`multipathd` がデバイスパス上で永続予約登録キーを見つけた場合、そのキーをすべてのアクティブなパスに追加します。その結果、`multipathd` が実行されていないときにパスデバイス



が最初に表示された場合でも、multipath デバイスではすべてのパス上で永続予約が正しくセットアップされるようになりました。

[Bugzilla:2164869](#)

### 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<sup>\[1\]</sup>](#)

### /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<sup>\[1\]</sup>](#)

## 8.10. 高可用性およびクラスター

### pcs config Checkpoint diff コマンドがすべての設定セクションで正しく機能するようになる

RHEL 9.0 リリースの時点で、**pcs config checkpoint diff** コマンドは、Fencing Levels、Ordering Constraints、Colocation Constraints、Ticket Constraints、Resources Defaults、および Operations Defaults の設定セクションの違いを表示しなくなりました。RHEL 9.1 リリースの時点で、**pcs config checkpoint diff** コマンドは、Resources および Stonith デバイス設定セクションの違いを表示しなくなりました。これは、さまざまな設定セクションをそれぞれ表示するコードが CIB ファイルをロードするための新しいメカニズムに切り替わったため、ロードされたコンテンツがキャッシュされたことが原因です。差分比較に使用された 2 番目のファイルはロードされず、最初のファイルのキャッシュされたコンテンツが代わりに使用されました。その結果、**diff** コマンドは出力を生成しませんでした。この修正により、CIB ファイルの内容はキャッシュされなくなり、**pcs config checkpoint diff** コマンドですべての設定セクションの差異が表示されるようになりました。

[Bugzilla:2175881](#)

### pcsd Web UI はフェンスレベルが設定されているときにクラスターのステータスを表示するようになる

以前は、フェンスレベルが設定されている場合、**pcsd** Web UI にはクラスターのステータスが表示されませんでした。この修正により、フェンスレベルが設定されている場合に、Web UI を使用してクラスターのステータスを表示し、クラスター設定を変更できるようになりました。

[Bugzilla:2182810](#)

### 2 番目のフェンシングデバイスとして設定されたフェンス watchdog は、最初のデバイスがタイムアウトになったときにノードをフェンスするようになる

以前は、watchdog フェンシングデバイスがフェンシングトポロジーの2番目のデバイスとして設定されている場合、フェンシング操作のタイムアウトを計算するときに watchdog タイムアウトは考慮されませんでした。その結果、最初のデバイスがタイムアウトすると、watchdog がノードをフェンスしても、フェンシング操作はタイムアウトになります。この修正により、watchdog タイムアウトがフェンシング操作のタイムアウトに含まれ、最初のデバイスがタイムアウトになった場合にフェンシング操作は成功します。

[Bugzilla:2182482](#)

### リストがノードごとにグループ化されている場合、ルールを含む場所の制約が表示されなくなる

ルールを含むロケーション制約にはノードを割り当てることができません。以前は、ノードごとにリストをグループ化すると、ルールを含む場所の制約が空のノードの下に表示されていました。この修正により、ルールのある場所の制約は表示されなくなり、ルールのある制約が表示されないことを示す警告が表示されます。

[Bugzilla:1423473](#)

### マルチパス SCSI デバイスを更新する pcs コマンドが正しく動作するようになる

Pacemaker CIB ファイルの変更により、**pcs stonith update-scsi-devices** コマンドが設計どおりに動作しなくなり、一部のクラスターリソースが望ましくない再起動を引き起こしました。この修正により、このコマンドは正しく機能し、同じノード上で実行されている他のクラスターリソースを再起動することなく SCSI デバイスを更新できるようになりました。

[Bugzilla:2177996](#)

### pcsd Web UI が開いているときに pcsd-ruby デーモンのメモリーフットプリントが削減されるようになる

以前は、**pcsd** Web UI が開いていると、**pcsd-ruby** デーモンのメモリー使用量が数時間にわたって着実に増加していました。この修正により、**pcsd-ruby** デーモンで実行される Web サーバーが定期的にグレースフルリスタートを実行するようになりました。これにより、割り当てられたメモリーが解放され、メモリーフットプリントが削減されます。

[Bugzilla:1860626<sup>\[1\]</sup>](#)

### azure-events-az リソースエージェントが Pacemaker 2.1 以降でエラーを生成しなくなる

**azure-events-az** リソースエージェントは **crm\_simulate -Ls** コマンドを実行し、出力を解析します。Pacemaker 2.1 以降では、**crm\_simulate** コマンドの出力に **Transition Summary:** というテキストが含まれなくなり、エラーが発生しました。この修正により、このテキストが欠落している場合でもエージェントはエラーを生成しなくなりました。

[Bugzilla:2182415](#)

### mysql リソースエージェントがプロモート可能なクローンリソースで正しく動作するようになる

以前は、プロモートされた値とプロモートされていない値の間でプロモーションスコアが変化するため、**mysql** リソースエージェントは Promoted ロールで動作しているクローンリソースをノード間で移動していました。この修正により、Promoted ロールのノードは Promoted ロールのままになります。

[Bugzilla:2179003<sup>\[1\]</sup>](#)

### fence\_scsi エージェントが共有 lvmlockd デバイスを自動検出できるようになる

以前は、**fence\_scsi** エージェントは共有 **lvmlckd** デバイスを自動検出しませんでした。今回の更新により、**fence\_scsi** は、**devices** 属性が設定されていない場合に、**lvmlckd** デバイスを自動検出できるようになりました。

[Bugzilla:2187327](#)

## 8.11. コンパイラーおよび開発ツール

### **glibc system()** 関数は、以前の信号マスクを無条件に復元するようになる

以前は、**glibc system()** 関数が複数のスレッドから同時に呼び出された場合、**SIGCHLD** シグナルのシグナルマスクが正しく復元されないことがありました。その結果、一部のスレッドでは **glibc system()** 関数から戻った後も、**SIGCHLD** シグナルがブロックされたままでした。

今回の更新により、**glibc system()** 関数は、並列 **system()** 関数呼び出しが実行されている場合でも、以前の信号マスクを無条件に復元するようになりました。その結果、**glibc system()** 関数が複数のスレッドから同時に呼び出された場合に、**SIGCHLD** シグナルが誤ってブロックされることがなくなりました。

[Bugzilla:2177235](#)

### **eu-addr2line -C** が他の引数を正しく認識するようになる

以前は、**elfutils** の **eu-addr2line** コマンドで **-C** 引数を使用すると、次の1文字の引数が消えていました。その結果、**eu-addr2line -Ci** コマンドは **eu-addr2line -C** と同じように動作し、**eu-addr2line -iC** は期待どおりに動作しました。このバグは修正され、**eu-addr2line -Ci** は 両方の引数を認識するようになりました。

[Bugzilla:2182059](#)

### **eu-addr2line -i** は、GCC リンクタイム最適化でコンパイルされたコードを正しく処理するようになる

以前は、**elfutils** に含まれる **libdw** ライブラリーの **dwarf\_getscopes** 関数は、GCC リンクタイム最適化でコンパイルされた関数の抽象オリジン定義を見つけることができませんでした。したがって、**eu-addr2line** コマンドで **-i** 引数を使用すると、**eu-addr2line** は **gcc -flto** でコンパイルされたコードのインライン関数を表示できませんでした。今回の更新により、**libdw dwarf\_getscopes** 関数はインライン化スコープの正しいコンパイル単位を検索し、**eu-addr2line -i** は 期待どおりに動作するようになりました。

[Bugzilla:2236182](#)

### **papi** を使用するプログラムがシャットダウン時に停止しなくなる

以前は、**papi** が一部のコンポーネントを初期化する前に、**papi** はスレッドを初期化していました。このため、配列内の要素の数を記述する特定のコンポーネントのエントリーが正しい値に設定されず、ゼロサイズのメモリー割り当てが試行されました。その結果、その後アクセスしてこれらのゼロサイズのメモリー割り当てを解放すると、プログラムが停止してしまいました。

このバグは修正され、**papi** を使用するプログラムはシャットダウン時に停止しなくなりました。

[Bugzilla:2215582](#)

### OpenJDK XML 署名プロバイダーが FIPS モードで機能するようになる

以前は、OpenJDK XML 署名プロバイダーは FIPS モードで動作できませんでした。FIPS モードのサポートが強化された結果、OpenJDK XML 署名プロバイダーが FIPS モードで有効になりました。

[Bugzilla:2186647](#)

## 8.12. IDENTITY MANAGEMENT

### 通常のユーザーによるページ検索はパフォーマンスに影響を与えなくなる

以前は、Directory Server に検索負荷がかかっている場合、ロックがネットワークイベントをポーリングするスレッドと競合するため、通常のユーザーからのページ検索がサーバーのパフォーマンスに影響を与える可能性がありました。さらに、ページ検索の送信中にネットワークの問題が発生した場合、**nsslapd-iotimeout** パラメーターの期限が切れるまでサーバー全体が応答しなくなっていました。今回の更新では、ネットワークイベントとの競合を避けるために、ロックがいくつかの部分に分割されました。その結果、一般ユーザーからのページ検索中のパフォーマンスへの影響はありません。

[Bugzilla:1974242](#)

### スキーマのレプリケーションが Directory Server で正しく機能するようになる

以前は、Directory Server がスキーマを新しいサーバーに複製するときに、すべてのスキーマがリモートレプリカ上の **99user.ldif** ファイルに追加されました。**X-ORIGIN** キーワードがすべての定義で **user defined** に設定されているため、すべてカスタムスキーマであるように見えました。その結果、Web コンソールに問題が発生する可能性があり、スキーマを監視し、**X-ORIGIN** キーワードに特定の値が含まれることを期待するお客様にも問題が発生する可能性がありました。この更新により、スキーマのレプリケーションが期待どおりに機能するようになりました。

[Bugzilla:1759941](#)

### Directory Server で referral モードが正しく動作するようになる

以前は、CLI は **nsslapd-referral** 設定属性をマッピングツリーではなくバックエンドに設定していました。その結果、referral モードは機能しませんでした。この更新により、**nsslapd-referral** 属性が正しく設定され、referral モードが期待どおりに機能するようになりました。

[Bugzilla:2053204](#)

### LMDB インポートがより高速に動作するようになる

以前は、**entryrdn** インデックスを構築するために、LMDB インポートワーカースレッドは、親エントリーが確実に処理されるように他のワーカースレッドを待機していました。これによりロック競合が発生し、インポートが大幅に遅くなりました。今回の更新により、LMDB データベースを介した LDIF インポートが再設計され、プロバイダスレッドは、エントリー RDN とその親に関するデータを、ワーカースレッドが **entryrdn** インデックスの構築に使用する一時データベースに保存します。その結果、ワーカースレッドの同期が必要なくなり、平均インポート速度が向上しました。

LMDB は同時書き込みトランザクションをサポートしていないため、LMDB インポートのインポート速度は依然として BDB インポートより 3 倍遅いことに注意してください。

[Bugzilla:2116948](#)

### 再起動後に dirsrv サービスが正しく開始されるようになる

以前は、**dirsrv** サービスが **systemd-tmpfiles-setup.service** の終了を明示的に待機しなかったため、再起動後に **dirsrv** サービスの起動に失敗することがありました。これにより競合状態が発生しました。今回の更新により、**dirsrv** サービスは **systemd-tmpfiles-setup.service** が完了するまで待機し、再起動後に起動に失敗することがなくなりました。

[Bugzilla:2179278](#)

### セキュリティーパラメーターの変更が正しく機能するようになる

以前は、`dsconf instance_name security set` コマンドを使用してセキュリティーパラメーターを変更すると、次のエラーが発生して操作が失敗していました。

```
Name 'log' is not defined
```

この更新により、セキュリティーパラメーターの変更が期待どおりに機能するようになりました。

[Bugzilla:2189717](#)

## SSSD が GPO ベースのアクセス制御を評価するときに `sAMAccountName` を使用するようになる

以前は、AD クライアントで `ldap_user_name` が `sAMAccountName` 以外の値に設定されている場合、GPO ベースのアクセス制御が失敗しました。今回の更新により、SSSD が GPO ベースのアクセス制御を評価するときに常に `sAMAccountName` を使用するようになりました。`ldap_user_name` が AD クライアントの `sAMAccountName` とは異なる値に設定されている場合でも、GPO ベースのアクセス制御が正しく機能するようになりました。

[Jira:SSSD-6107](#)

## SSSD は、ユーザーを取得するときに `user_attributes` オプションで重複した属性を処理するようになる

以前は、`sssd.conf` が `user_attributes` オプションに重複した属性を含んでいた場合、SSSD はこれらの重複を正しく処理しませんでした。その結果、それらの属性を持つユーザーを取得できませんでした。この更新により、SSSD は重複を正しく処理できるようになりました。その結果、重複した属性を持つユーザーを取得できるようになりました。

[Jira:SSSD-6177](#)

## 動的 Kerberos PAC チケット署名強制メカニズムにより、IdM でのバージョン間の非互換性が修正される

以前は、Identity Management (IdM) デプロイメントのサーバーが RHEL 9 と RHEL 8 の両方で実行されている場合、特権属性証明書 (PAC) チケット署名サポートのアップストリーム実装によって生じる非互換性により、特定の操作が失敗していました。今回の更新では、RHEL 9 に動的チケット署名強制メカニズムの機能が実装され、このバージョン間の非互換性が修正されました。この機能を実際に有効にするには、以下を行う必要があります。

1. ドメイン内のすべてのサーバーを更新します。
2. すべての IdM Kerberos Distribution Center (KDC) サービスを再起動します。

これら 2 つの操作の順序は重要です。KDC は起動時に、ドメイン内の他のすべてのサーバーのメタデータに対してクエリーを実行し、それらのサーバーがすべて PAC チケット署名をサポートしているかどうかを確認します。サポートしていない場合、署名は強制されません。

制約付き委任リクエストの例など、動的な Kerberos PAC チケット署名強制メカニズムの詳細については、こちらの [ナレッジベースの記事](#) を参照してください。

[Jira:RHELDOCS-17011<sup>\[1\]</sup>](#), [Bugzilla:2182683](#), [Bugzilla:2178298](#)

## SHA-1 署名検証が FIPS モードで許可されるようになる

以前は、Identity Management (IdM) が FIPS モードの場合、SHA-1 署名検証の使用を許可できませんでした。これは、IdM が、SHA-1 署名を許可しない FIPS-140-3 標準を使用しているためです。この状況により、Active Directory (AD) の相互運用性に問題が発生していました。AD は古い FIPS-140-2 標準に

のみ準拠しており、SHA-1 署名を必要とするためです。

この更新により、PKINIT 署名検証に FIPS 例外が導入されました。IdM で FIPS モードが有効になっている場合、当該制約は無視されます。デフォルトモードの制約のみが適用され、FIPS モードの場合でも **SHA1** 暗号化モジュールの使用が許可されます。その結果、FIPS モードでの AD の相互運用性は意図したとおりに機能します。

IdM/AD 信頼のシナリオ、または RHEL 9.2 以降のホストを AD クライアントとして使用するシナリオにおいて FIPS モードで PKINIT をサポートするには、暗号化ポリシーを FIPS:AD-SUPPORT:SHA1 に設定する必要があります。

[Bugzilla:2155607](#)

### IdM admin ユーザーの削除が許可されなくなる

以前は、**admin** グループのメンバーであれば、Identity Management (IdM) の **admin** ユーザーを削除できました。**admin** ユーザーが存在しないと、IdM と Active Directory (AD) の間の信頼が正しく機能しません。この更新により、**admin** ユーザーを削除できなくなりました。結果として、IdM-AD 信頼は正しく機能します。

[Bugzilla:2229712](#)

### ipa-kdb に起因する krb5kdc の障害が発生しなくなる

以前は、**ipa-kdb** ドライバーはサーバーホストオブジェクトの欠如と接続の失敗を区別していませんでした。その結果、LDAP サーバーとの接続の問題によって生成された **NULL** LDAP コンテキストが原因で、**krb5kdc** サーバーが予期せず停止することがありました。

今回の更新により、**ipa-kdb** ドライバーは接続の失敗を正しく識別し、接続の失敗とサーバーホストオブジェクトの欠如を区別できるようになりました。その結果、**krb5kdc** サーバーで障害が発生しなくなりました。

[Bugzilla:2227831](#)

### IdM クライアントインストーラーは、**ldap.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 **ldap.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 **ldap.conf** ファイルに TLS CA 設定をセットアップしません。

[Bugzilla:2094673](#)

### 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。

[RHBA-2023:4359](#) のリリースにより、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](#)

## 8.13. WEB コンソール

## Web コンソールの NBDE バインディング手順が、ルートファイルシステムを持つボリュームグループで機能するようになる

RHEL 9.2 では、ユーザーがルートファイルシステムに Tang キーを追加したかどうかを判断するコードのバグが原因で、LUKS コンテナ上にファイルシステムがまったく存在しない場合、Web コンソールのバインドプロセスがクラッシュしました。**Verify key** ダイアログの **Trust key** ボタンをクリックした後、Web コンソールにエラーメッセージ **TypeError: Qe(...) is undefined** が表示されたため、説明されているシナリオのコマンドラインインターフェイスで必要な手順をすべて実行する必要がありました。

この更新により、Web コンソールはルートファイルシステムへの Tang キーの追加を正しく処理できるようになりました。その結果、Web コンソールは、さまざまなシナリオで Network-Bound Disk Encryption (NBDE) を使用した LUKS 暗号化ボリュームの自動ロック解除に必要なバインド手順をすべて完了します。

[Bugzilla:2203361](#)

## 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。

今回の更新で問題が修正され、3840x2160 などの超高解像度を除き、VNC コンソールが最大解像度で正しく機能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

### storage ロールが、アンマウントせずにマウントされたファイルシステムのサイズを変更できるようになる

以前は、ファイルシステムがオンラインサイズ変更をサポートしていても、**storage** ロールはマウントされたデバイスのサイズを変更できませんでした。その結果、**storage** ロールはサイズ変更の前にすべてのファイルシステムをアンマウントしましたが、実行中のシステムの / ディレクトリーのサイズ変更中など、使用中のファイルシステムでは失敗しました。

この更新により、**storage** ロールは、XFS や Ext4 などのオンラインサイズ変更をサポートする、マウントされたファイルシステムのサイズ変更をサポートするようになりました。その結果、マウントされたファイルシステムをアンマウントせずにサイズ変更できるようになりました。

[Bugzilla:2168692](#)

### podman\_registries\_conf 変数で unqualified-search-registries フィールドが正しく設定されるようになる

以前は、**podman\_registries\_conf** 変数を設定した後、**podman** RHEL システムロールが失敗していました。その結果、**unqualified-search-registries = ["registry.access.redhat.com"]** 設定は、**/etc/containers/registries.conf.d/50-systemroles.conf** ファイルに生成されませんでした。今回の更新で、この問題が修正されています。

[Bugzilla:2211984](#)

## kdump ロールが `authorized_keys` をべき等性を持たせて追加する

以前は、`authorized_key` を追加するタスクは毎回余分な改行文字を追加していました。そのため、ロールはべき等性を持ちませんでした。この修正により、新しい `authorized_key` の追加は正しく機能し、単一のキー値のみをべき等性を持たせて追加します。

[Bugzilla:2232241](#)

## kdump\_authorized\_keys がない場合、kdump システムロールは失敗しない

以前は、`kdump_ssh_user` 変数で定義されたユーザーが `home` ディレクトリー内の `.ssh` ディレクトリーまたは空の `.ssh/authorized_keys` ファイルにアクセスできない場合、`kdump` システムロールは SSH 認可キーを追加できませんでした。この修正により、`kdump` システムロールは認可されたキーを SSH 設定に正しく追加するようになりました。その結果、キーベースの認証は、説明したシナリオで確実に機能します。

[Bugzilla:2232231](#)

## 作成前にメンバーディスクからデータを削除できなかった問題が解消される

以前は、RAID ボリュームを作成するときに、システムは RAID ボリュームを形成する前にメンバーディスクから既存のデータを効果的に削除しませんでした。この更新により、RAID ボリュームは必要に応じてメンバーディスクから既存のデータを削除します。

[Bugzilla:2224090](#)

## 存在しないサービスを使用して firewall RHEL システムロールをチェックモードで実行しても失敗しなくなる

以前は、存在しないサービスを使用してチェックモードで `firewall` ロールを実行すると失敗していました。この修正により、チェックモードの Ansible ベストプラクティスへの準拠性が向上しました。その結果、存在しないサービスが有効または無効でも、チェックモードのロールは失敗しなくなりました。代わりに、サービスが以前の Playbook で定義されていることを確認する警告が表示されます。

[Bugzilla:2222428](#)

## RHEL 7 の firewall RHEL システムロールが存在しない Python パッケージのインストールを試行しなくなる

以前は、RHEL 7 の `firewall` ロールが別のロールから呼び出され、そのロールが `python3` を使用していた場合、`firewall` ロールはそのバージョンの Python の `python3-firewall` ライブラリーをインストールしようとしていました。ただし、そのライブラリーは RHEL 7 では利用できません。そのため、`python3-firewall` ライブラリーが見つからず、次のエラーメッセージが表示されていました。

```
No package matching 'python3-firewall' found available, installed or updated
```

今回の更新で、`firewall` ロールは、`python-firewall` または `python3-firewall` ライブラリーのインストールを試行しなくなりました。その結果、マネージドノードに `python3` がインストールされている場合、RHEL 7 で `firewall` ロールは失敗しません。

[Bugzilla:2216520](#)

## kdump RHEL システムロールの更新

`kdump` RHEL システムロールが新しいバージョンに更新され、次の主な機能拡張が行われました。

- `kexec-tools` をインストールすると、このファイルを管理する必要がなくなるため、ユーティリティースイートは `/etc/sysconfig/kdump` ファイルを生成しなくなります。



- このロールは、`auto_reset_crashkernel` 変数と `dracut_args` 変数をサポートしています。

詳細は、`/usr/share/doc/rhel-system-roles/kdump/` ディレクトリー内のリソースを参照してください。

[Bugzilla:2211187](#)

### rhc ロールを使用して作成された Insights タグが正しく適用されるようになる

以前は、`rhc` ロールを使用して Insights タグを作成した場合、タグは正しいファイルに保存されませんでした。そのため、タグは Insights に送信されず、その結果、タグは Insights インベントリー内のシステムに適用されませんでした。

この修正により、タグが正しく保存され、Insights インベントリーに存在するシステムに適用されるようになりました。

[Bugzilla:2209200](#)

### raid\_chunk\_size パラメーターがエラーメッセージを返さなくなる

以前は、`raid_chunk_size` 属性は、RAID プールおよびボリュームに対して許可されていませんでした。この更新により、制限を受けることなく RAID プールおよびボリュームの `raid_chunk_size` 属性を設定できるようになりました。

[Bugzilla:2193058](#)

### certificate RHEL システムロールは、新しい証明書要求を実行するかどうかを決定するときに証明書のキーサイズをチェックするようになる

以前は、`certificate` RHEL システムロールは、新しい証明書を要求するかどうかを評価するときに、証明書のキーサイズをチェックしませんでした。その結果、同ロールは、新しい証明書要求を発行すべき場合に、発行しないことがありました。この更新により、`certificate` は `key_size` パラメーターをチェックして、新しい証明書要求を実行する必要があるかどうかを判断するようになりました。

[Bugzilla:2186057](#)

### kdump ロールは、複数のキーを `authorized_keys` にべき等性を持たせて追加する

以前は、`authorized_keys` ファイルに複数の SSH キーを同時に追加すると、あるホストのキー値が別のホストに置き換えられていました。この更新では、`lineinfile` モジュールを使用して `authorized_keys` ファイルを管理することで問題が修正されます。`lineinfile` はタスクを順番に繰り返し処理し、既存のキーをチェックし、1つのホストで一度に新しいキーを1つのアトミック操作で書き込みます。その結果、複数のホストに SSH キーを追加すると正しく機能し、別のホストのキーの値を置き換えません。

注記: `serial: 1` play serial キーワードを play レベルで使用して、1度に行われているホストの数を制御します。

[Jira:RHEL-1499<sup>\[1\]</sup>](#)

### kdump ロールにより、`kdump_ssh_server` 認証の `.ssh/authorized_keys` が正常に更新されません。

以前は、`kdump` ロールが `.ssh` ディレクトリーにアクセスして、`kdump_ssh_server` にログインするためのユーザーを安全に認証できませんでした。その結果、`kdump` ロールは、`kdump_ssh_server` が失敗したことを確認するための `.ssh/authorized_keys` ファイルと SSH メカニズムを更新しませんでした。今回の更新でこの問題が修正されています。これにより、`kdump_ssh_server` の `kdump_ssh_user` 認証が確実に機能します。

Jira:RHEL-1397<sup>[1]</sup>

システムロールに対して **kdump** を有効にするには、RHEL 9 以降のバージョンで **failure\_action** 設定パラメーターを使用する必要があります。

以前は、**kdump** 設定中に **default** オプションを使用すると成功せず、以下の警告がログに出力されてきました。

```
kdump: warning: option 'default' was renamed 'failure_action' and will be removed in the future.
please update /etc/kdump.conf to use option 'failure_action' instead.
```

その結果、**default** オプションが使用されていた場合、ロールは **kdump** を正常に有効にしませんでした。今回の更新で問題が修正され、**failure\_action** パラメーターを使用して、複数のシステムにカーネルダンプパラメーターを設定できるようになりました。これにより、上述のシナリオで **kdump** の有効化が正常に機能します。

Jira:RHEL-906<sup>[1]</sup>

**firewall** システムロールの **previous: replaced** パラメーターは、以前の設定を削除せずにオーバーライドするようになりました。

以前は、変数リストに **previous: replaced** パラメーターを追加すると、**firewall** システムロールは既存のユーザー定義設定をすべて削除し、**firewalld** をデフォルト設定にリセットしていました。この修正では、EL7 リリースで導入された **firewalld** のフォールバック設定を使用して、以前の設定を保持します。その結果、変数リストで **previous: replaced** パラメーターを使用すると、**firewall.conf** 設定ファイルはリセット時に削除されませんが、ファイルとファイル内のコメントが保持されます。

Jira:RHEL-1495<sup>[1]</sup>

**firewall** RHEL システムロールは、チェックモードで **previous: replaced** を使用すると、変更を正しく報告します。

以前は、**previous: replaced** ロールは、チェックモードで **previous: replaced** パラメーターを使用するときに、ファイルの変更についてチェックしていませんでした。その結果、ロールでは未定義の変数に関するエラーが発生しました。この修正により、**previous: replaced** パラメーターによるファイルの変更を評価する新しいチェック変数がチェックモードに追加されました。**firewalld.conf** ファイルのチェックでは、**rpm** データベースを評価して、ファイルがパッケージに同梱されているバージョンから変更されているか判断します。その結果、**firewall** ロールは、**previous: replaced** パラメーターを使用する場合、変更を正しく報告するようになりました。

Jira:RHEL-898<sup>[1]</sup>

**firewall** RHEL システムロールは、**Network Manager** インターフェイスにゾーンを割り当てる際に、変更を正しく報告します。

以前は、**Network Manager** インターフェイス割り当ては、変更が存在しない場合でも変更を報告していました。この修正により、**library/firewall\_lib.py** ファイル内の **try\_set\_zone\_of\_interface** モジュールは、インターフェイスのゾーンが変更されたかどうかを示す 2 番目の値を返します。その結果、**Network Manager** によって処理されるインターフェイスにゾーンを割り当てるときに、モジュールは変更を正しく報告するようになりました。

Jira:RHEL-885<sup>[1]</sup>

**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

### NVIDIA グラフィックスデバイスは VM のシャットダウン後も動作し続ける

以前の RHEL カーネルでは、デバイスの電源遷移遅延は PCIe 仕様で要求される遅延とより厳密に一致していました。その結果、一部の NVIDIA GPU は、アタッチされた仮想マシンのシャットダウン後にデバイス割り当てに使用されると、応答しなくなる可能性があります。この更新により、NVIDIA オーディオデバイス機能のデバイス電源遷移遅延が延長されます。結果として、NVIDIA GPU はこのシナリオでも引き続き正しく動作します。

[Bugzilla:2178956](#)<sup>[1]</sup>

フェイルオーバー virtio NICs には、Windows 仮想マシンで IP アドレスが正しく割り当てられるようになりました。

以前は、フェイルオーバー virtio NIC のみを使用して Windows 仮想マシン (VM) を起動すると、仮想マシンは NIC に IP アドレスを割り当てることができませんでした。その結果、NIC はネットワーク接続をセットアップできませんでした。この問題は修正され、VM NIC は、説明されているシナリオで想定されるとおりに、ネットワーク接続をセットアップするようになりました。

[Bugzilla:1969724](#)

インストーラーには、VM に RHEL をインストールするために予期されるシステムディスクが表示される

以前は、`virtio-scsi` デバイスを使用して VM に RHEL をインストールする場合、`device-mapper-multipath` のバグにより、これらのデバイスがインストーラーに表示されない可能性があります。したがって、インストール中に、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、`multipath` コマンドはすべてのデバイスがシリアルを持つと主張していました。このため、インストーラーは、VM に RHEL をインストールするための予期されたシステムディスクを見つけることができませんでした。

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、インストーラーは VM に RHEL をインストールするために予期されるシステムディスクを表示します。

[Bugzilla:1926147](#)<sup>[1]</sup>

Broadcom ネットワークアダプターは、ライブマイグレーション後に Windows 仮想マシン上で正しく動作するようになりました。。

以前は、Broadcom、Qlogic、Marvell などの Broadcom ファミリーのデバイスのネットワークアダプターは、Windows 仮想マシン (VM) のライブマイグレーション中にホットアンプラグできませんでした。その結果、移行が完了した後、アダプターは正しく動作しませんでした。この問題は、Single-root I/O virtualization (SR-IOV) を使用して Windows 仮想マシンにアタッチされているアダプターのみに影響していました。この更新により、基礎となるコードが修正され、問題は発生しなくなりました。

[Jira:RHEL-910](#)、[Bugzilla:2091528](#)、[Bugzilla:2111319](#)

**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 の **virtio-gpu** で **blob** リソースが正しく動作しない

**virtio-gpu** デバイスは現在、IBM Z システムの **blob** メモリーリソースと互換性がありません。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場合、仮想マシンにはグラフィカルな出力が表示されません。

[Jira:RHEL-7135](#)

## 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

#### NVMe over Fibre Channel デバイスが RHEL インストールプログラムでテクノロジープレビューとして利用可能になる

NVMe over Fibre Channel デバイスをテクノロジープレビューとして RHEL インストールに追加できるようになりました。RHEL インストールプログラムでは、Installation Destination 画面でディスクを追加するときに、NVMe Fabrics Devices セクションでこれらのデバイスを選択できます。

[Bugzilla:2107346](#)

### 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<sup>\[1\]</sup>](#)

### 9.3. 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<sup>\[1\]</sup>](#)

### 9.4. 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](#)

## 9.5. 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](#)<sup>[1]</sup>

### 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](#)<sup>[1]</sup>

### 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](#)<sup>[1]</sup>

### 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ができませんでした。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<sup>[1]</sup>

### 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 WWAN 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:RHELDPCS-16760<sup>[1]</sup>, Bugzilla:2123542, Jira:RHEL-6564, Bugzilla:2110561, 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<sup>[1]</sup>

### 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 の今回のリリースで、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、以下のとおりです。

- 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<sup>[1]</sup>

## 9.6. カーネル

Unified Kernel Image を備えた kdump メカニズムはテクノロジープレビューとして利用可能

Unified Kernel Image (UKI) に含まれるカーネルイメージを使用する **kdump** メカニズムは、テクノロジープレビューとして利用できます。UKI は、**initramfs**、**vmlinuz**、およびカーネルコマンドラインを1つのファイルにまとめた1つの実行可能ファイルです。UKI の主な利点は、SecureBoot の暗号化署名をすべてのコンポーネントに一度に拡張できることです。

この機能を使用するには、UKI に含まれるカーネルコマンドラインで **crashkernel=** パラメーターに適切な値を設定します。これにより、**kdump** に必要なメモリーが予約されます。

注記: 現在、Linux カーネルからの **kexec\_file\_load** システムコールは UKI をロードできません。したがって、**kexec\_file\_load** システムコールでクラッシュカーネルをロードする場合は、UKI に含まれるカーネルイメージのみが使用されます。

[Bugzilla:2169720<sup>\[1\]</sup>](#)

## SGX がテクノロジープレビューとして利用可能

**Software Guard Extensions (SGX)** は、ソフトウェアコードおよび公開および修正からのデータを保護する Intel® テクノロジーです。RHEL カーネルは、SGX v1 および v1.5 の機能を部分的に提供します。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。

[Bugzilla:1874182<sup>\[1\]</sup>](#)

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

カーネルの Intel データストリーミングアクセラレータードライバー (IDX) は、現在テクノロジープレビューとして利用できます。これは Intel CPU 統合アクセラレーターであり、プロセスアドレス空間 ID (pasid) 送信と共有仮想メモリー (SVM) を備えた共有ワークキューが含まれています。

[Bugzilla:2030412](#)

## 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ーアクセス (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<sup>\[1\]</sup>](#)

## SGX がテクノロジープレビューとして利用可能

**Software Guard Extensions (SGX)** は、ソフトウェアコードおよび公開および修正からのデータを保護する Intel® テクノロジーです。RHEL カーネルは、SGX v1 および v1.5 の機能を部分的に提供します。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。

Bugzilla:1660337<sup>[1]</sup>

### 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<sup>[1]</sup>

## 9.7. 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<sup>[1]</sup>

### 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<sup>[1]</sup>

### 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<sup>[1]</sup>

## 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 技術提案 8006 では、NVMe-oF の **DH-HMAC-CHAP** インバンド認証プロトコルが定義されており、この拡張機能により提供されます。

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<sup>[1]</sup>

## 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。現在、**io\_uring** コマンドパススルーを使用するには、カスタム SELinux ポリシーモジュールが必要です。カスタム SELinux ポリシーモジュールを作成します。

1. 次の行を **io\_uring\_cmd\_passthrough.cil** ファイルとして保存します。

```
---cut here---
( allow unconfined_domain_type device_node ( io_uring ( cmd )))
( allow unconfined_domain_type file_type ( io_uring ( cmd )))
---cut here---
```

2. ポリシーモジュールをロードします。

```
# semodule -i io_uring_cmd_passthrough.cil
```

Bugzilla:2068237<sup>[1]</sup>

## 9.8. 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](#)

## 9.9. 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](#)

### **Identity Management JSON-RPC API** がテクノロジープレビューとして利用可能

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

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

- 管理者は、管理しているクライアント以外のサーバーで、IdM の以前のバージョンもしくは最近のバージョンを使用できます。

- サーバーで IdM のバージョンを変更しても、開発者は特定バージョンの IdM コールを使用できます。

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

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

[Bugzilla:2084166](#)

### sssd-idp サブパッケージがテクノロジープレビューとして利用可能

SSSD の **sssd-idp** サブパッケージには、Identity Management (IdM) サーバーに対して OAuth2 認証を実行するクライアント側のコンポーネントである **oidc\_child** プラグインおよび **krb5 idp** プラグインが含まれます。この機能は、RHEL 9.1 以降の IdM サーバーのみで使用できます。

[Bugzilla:2065693](#)

### SSSD の内部 krb5 idp プラグインがテクノロジープレビューとして利用可能

SSSD **krb5 idp** プラグインを使用すると、OAuth2 プロトコルを使用して外部アイデンティティプロバイダー (IdP) に対して認証できます。この機能は、RHEL 9.1 以降の IdM サーバーのみで使用できます。

[Bugzilla:2056482](#)

### RHEL IdM では、ユーザー認証をテクノロジープレビューとして外部 ID プロバイダーに委任できる

RHEL IdM では、OAuth2 デバイス認証フローをサポートする外部 ID プロバイダー (IdP) にユーザーを関連付けることができるようになりました。これらのユーザーが RHEL 9.1 以降で利用可能な SSSD バージョンで認証すると、外部 IdP で認証と認可を実行した後、Kerberos チケットを使用した RHEL IdMSingle Sign-On 機能を受け取ります。

主な変更には以下のものがあります。

- **ipa idp-\*** コマンドによる外部 IdP への参照の追加、変更、および削除
- **ipa user-mod --user-auth-type=idp** コマンドを使用したユーザーの IdP 認証の有効化

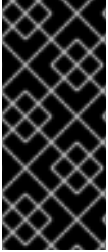
追加情報については、[外部 ID プロバイダーを使用した IdM への認証](#) を参照してください。

[Bugzilla:2069202](#)

### 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<sup>[1]</sup>

## 9.10. 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<sup>[1]</sup>

## 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<sup>[1]</sup>

## 9.11. 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<sup>[1]</sup>

### 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 のセキュリティ証明は含まれません。

[Jira:RHELPLAN-65217<sup>\[1\]</sup>](#)

## ARM 64 で仮想化が利用可能になる

テクノロジープレビューとして、ARM 64 CPU を使用してシステムに KVM 仮想マシンを作成できるようになりました。

[Jira:RHELPLAN-103993<sup>\[1\]</sup>](#)

## AMD64、Intel 64、および ARM 64 で virtio-mem が利用可能になる

RHEL 9 では、テクノロジープレビューとして、AMD64、Intel 64、および ARM 64 システムに **virtio-mem** 機能が追加されました。**virtio-mem** を使用すると、仮想マシンでホストメモリーを動的に追加または削除できます。

**virtio-mem** を使用するには、仮想マシンの XML 設定で **virtio-mem** メモリーデバイスを定義し、**virsh update-memory-device** コマンドを使用して、仮想マシンの実行中にメモリーデバイスのサイズ変更を要求します。このようなメモリーデバイスが実行中の仮想マシンに公開される現在のメモリーサイズを表示するには、仮想マシンの XML 設定を表示します。

ただし、**virtio-mem** は現在、Windows オペレーティングシステムを使用する仮想マシンでは動作しないことに注意してください。

[Bugzilla:2014487](#)、[Bugzilla:2044162](#)、[Bugzilla:2044172](#)

## 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<sup>\[1\]</sup>](#)

## 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<sup>\[1\]</sup>](#)

## 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<sup>[1]</sup>

## 9.12. 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<sup>[1]</sup>

## 9.13. コンテナ

### Podman の SQLite データベースバックエンドがテクノロジープレビューとして利用可能

Podman v4.6 以降、Podman の SQLite データベースバックエンドはテクノロジープレビューとして利用可能です。データベースバックエンドを SQLite に設定するには、`/etc/containers/containers.conf` 設定ファイルに **database\_backend = "sqlite"** オプションを追加します。SQLite データベースバックエンドに切り替える前に、**podman system reset** コマンドを実行してストレージを初期状態にリセットします。すべてのコンテナと Pod を再作成する必要があることに注意してください。SQLite データベースは、優れた安定性と一貫性を保証します。コンテナスタック内の他のデータベースも同様に SQLite に移動されます。BoltDB は引き続きデフォルトのデータベースバックエンドです。

Jira:RHELPLAN-154429<sup>[1]</sup>

**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<sup>[1]</sup>



## 第10章 非推奨になった機能

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。

非推奨のデバイスは完全にサポートされています。つまり、それらのデバイスはテストおよび保守されており、Red Hat Enterprise Linux 9 内でのサポート状況は変更されません。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。

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。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<sup>[1]</sup>

#### 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<sup>[1]</sup>

### 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** 値は非推奨になりました。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](#)

## 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<sup>[1]</sup>

### fapolicyd.rules が非推奨に

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

じ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

SCP (Secure Copy Protocol) には既知のセキュリティ脆弱性があるため、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が低下します。

- **scp** ユーティリティでは、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

[Jira:RHELPLAN-99136<sup>\[1\]</sup>](#)

## 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](#)

## NTLM と Krb4 は Cyrus SASL では非推奨となりました。

NTLM および Kerberos 4 認証プロトコルは非推奨となり、RHEL の今後のメジャーバージョンでは削除される可能性があります。これらのプロトコルはセキュアであると見なされなくなり、アップストリームの実装からはすでに削除されています。

[Jira:RHELDPCS-17380<sup>\[1\]</sup>](#)

## 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<sup>\[1\]</sup>](#)

## 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<sup>[1]</sup>

### **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** コマンドの **--token** オプションは非推奨になりました。

**subscription-manager register** コマンドの **--token=<TOKEN>** オプションは、システムを Red Hat に登録するのに役立つ認証方法です。このオプションは、エンタイトルメントサーバーが提供する機能に応じて異なります。デフォルトのエンタイトルメントサーバー **subscription.rhsm.redhat.com** は、この機能をオフにする予定です。その結果、**subscription-manager register --token=<TOKEN>** を使用しようとする、次のエラーメッセージが表示されて失敗する可能性があります。

```
Token authentication not supported by the entitlement server
```

**subscription-manager register** コマンドのペアのオプション **--username / --password** および **--org / --activationkey** を含めるなど、他の認証方法を使用してシステムの登録を続けることができます。

[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<sup>\[1\]</sup>](#)

### **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](#)

## 10.5. ネットワーク

### RHEL 9 でネットワークチームが非推奨に

**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:1935544](#)<sup>[1]</sup>

### ifcfg 形式の NetworkManager 接続プロファイルが非推奨に

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](#)<sup>[1]</sup>

### firewalld の iptables バックエンドが非推奨に

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](#)

### PF\_KEYv2 カーネル API が非推奨に

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](#)<sup>[1]</sup>

## 10.6. カーネル

ATM カプセル化は DLPI 9 で非推奨になりました

### ATM カプセル化は RHEL7 で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 システムコールは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](#)<sup>[1]</sup>

### RHEL 9 でネットワークチームが非推奨に

**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](#)<sup>[1]</sup>

## 10.7. 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 で削除されました。

**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 では、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**
- **libpmemobj-debug**
- **libpmempool**
- **libpmempool-devel**
- **libpmempool-debug**
- **pmempool**
- **daxio**
- **pmreorder**
- **pmdk-convert**
- **libpmemobj++**
- **libpmemobj++-devel**
- **libpmemobj++-doc**

Jira:RHELDOS-16432<sup>[1]</sup>

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

## 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<sup>[1]</sup>、Jira:RHELPLAN-80695、[Bugzilla:1974657](#)

## 10.9. コンパイラーおよび開発ツール

### 2048 より小さいサイズのキーは Go の FIPS モードの openssl 3.0 で非推奨に

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](#)

### 一部の PKCS1 v1.5 モードは Go の FIPS モードで非推奨に

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<sup>\[1\]</sup>](#)

## 10.10. IDENTITY MANAGEMENT

### OpenDNSSec の SHA-1 を非推奨化

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ドメインは、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 `<g id="1">ファイル</g>` 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



定めます。

```
[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<sup>[1]</sup>

### SSSD files プロバイダーは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<sup>[1]</sup>

### nsslapd-ldapimaprootdn パラメーターが非推奨になる

Directory Server では、**nsslapd-ldapimaprootdn** 設定パラメーターを使用して、システムルートエントリーをルート DN エントリーにマップします。通常、**nsslapd-ldapimaprootdn** パラメーターの値は **nsslapd-rootdn** パラメーターと同じです。さらに、一方の属性を変更してもう一方の属性を変更しない場合、自動バインド設定が機能しなくなり、**dsconf** ユーティリティーと Web コンソールへのアクセスが中断されます。

この更新により、Directory Server は **nsslapd-rootdn** パラメーターのみを使用して、システムルートエントリーをルート DN エントリーにマップします。その結果、**nsslapd-ldapimaprootdn** パラメーターは非推奨となり、ルート DN の変更によって **dsconf** ユーティリティーと Web コンソールへのアクセスが中断されなくなりました。

Bugzilla:2170494

### nsslapd-conntablesize 設定パラメーターが 389-ds-base から削除される

**nsslapd-conntablesize** 設定パラメーターは、RHEL 9.3 の **389-ds-base** パッケージから削除されました。以前は、**nsslapd-conntablesize** 設定属性で、確立された接続を管理する接続テーブルのサイズを指定していました。確立された接続の管理を改善するマルチリスナー機能の導入により、Directory Server は接続テーブルのサイズを動的に計算するようになりました。これにより、接続テーブルサイズの設定が小さすぎて、サーバーがサポートできる接続数に影響を与えた場合の問題も解決されます。RHEL 9.3 以降では、Directory Server がサポートできる TCP/IP 接続の数を管理するには、**nsslapd-maxdescriptors** および **nsslapd-reservedescriptors** 属性のみを使用します。

Bugzilla:2098236

### SMB1 プロトコルは Samba では非推奨に

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

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

Jira:RHELDPCS-16612<sup>[1]</sup>

## 10.11. 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<sup>[1]</sup>

### 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<sup>[1]</sup>

## 10.12. グラフィックインフラストラクチャー

### Motif は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<sup>[1]</sup>

## 10.13. RED HAT ENTERPRISE LINUX システムロール

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

## 10.14. 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

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<sup>[1]</sup>

仮想マシンスナップショットのサポートが限定されました

仮想マシンのスナップショットの作成は、現在、UEFI ファームウェアを使用していない仮想マシンのみでサポートされています。さらに、スナップショット操作中に QEMU モニターがブロックされる可能性があり、これは特定のワークロードのハイパーバイザーのパフォーマンスに悪影響を及ぼします。

また、現在の仮想マシンスナップショットの作成メカニズムは非推奨となり、Red Hat は実稼働環境での仮想マシンスナップショットの使用を推奨していないことにも注意してください。ただし、新しい VM スナップショットメカニズムは開発中であり、RHEL 9 の将来のマイナーリリースで完全に実装される予定です。

Jira:RHELDPCS-16948<sup>[1]</sup>、Bugzilla:1621944

仮想フロッピードライバーが非推奨に

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<sup>[1]</sup>

### libvirtd が非推奨に

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<sup>[1]</sup>

### レガシー CPU モデルは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は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<sup>[1]</sup>

### 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<sup>[1]</sup>

## 10.15. 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<sup>[1]</sup>

### Podman 内の SHA1 ハッシュアルゴリズムが非推奨に

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<sup>[1]</sup>

### rhel9/pause が非推奨に

**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

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<sup>[1]</sup>

### Inkscape および LibreOffice Flatpak イメージが非推奨に

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:RHELDPCS-17102<sup>[1]</sup>

## 10.16. 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。

- adwaita-gtk2-theme
- 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
- 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
- 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
- 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
- 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-qtdoc
- qt5-qtgraphicaleffects
- qt5-qtgraphicaleffects-doc
- 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-qtX11extras
- qt5-qtX11extras-devel
- qt5-qtX11extras-doc
- qt5-qtxmlpatterns
- qt5-qtxmlpatterns-devel
- qt5-qtxmlpatterns-doc
- qt5-qtxmlpatterns-examples
- qt5-rpm-macros
- qt5-srpm-macros
- webkit2gtk3
- webkit2gtk3-devel
- webkit2gtk3-jsc
- webkit2gtk3-jsc-devel
- wpebackend-fdo
- wpebackend-fdo-devel
- xorg-x11-server-Xorg

## 第11章 既知の問題

ここでは、Red Hat Enterprise Linux 9.3 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<sup>[1]</sup>

#### `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<sup>[1]</sup>

#### 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

サードパーティーツールを使用して作成した USB から RHEL インストールを起動すると、インストーラーは **Local Media** インストールソースを検出できません (Red Hat CDN のみが検出されます)。

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<sup>[1]</sup>

### 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<sup>\[1\]</sup>](#)

## 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、インストーラーは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 & Host Name** 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<sup>[1]</sup>

### 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 RM 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場合があります。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

### 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。
- インストーラーの起動オプションを使用して、**%pre** スクリプト用に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ตอบสนอง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](#)

## 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 を使用するようサーバーとクライアントを設定します。

[Bugzilla:1681178<sup>\[1\]</sup>](#)

**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<sup>\[1\]</sup>](#)

### 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検出しない

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_l1.yml
```

**cis\_server\_l1** 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<sup>\[1\]</sup>](#)

## 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<sup>[1]</sup>

### 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<sup>[1]</sup>

### 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<sup>[1]</sup>

### 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のルールの修正が失敗する場合があります。

キックスタートのインストール時に、OpenSCAP ユーティリティーで、サービス **enable** または **disable** 状態の修正が必要でないことが誤って表示されることがあります。これにより、OpenSCAP が、インストール済みシステムのサービスを非準拠状態に設定する可能性があります。回避策として、キックスタートインストール後にシステムをスキャンして修復できます。これにより、サービス関連の問題が修正されます。

[BZ#1834716](#)

## 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<sup>[1]</sup>

## 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](#)<sup>[1]</sup>

### 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** ユーティリティー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<sup>[1]</sup>

## 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

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。攻撃者は UDP を使用し、送信元アドレスをスプーフィングすることで、サービス一覧を要求し、スプーフィング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<sup>\[1\]</sup>](#)

## 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

### `mlx5` ドライバーを使用し、MTU が 3498 バイトを超えた状態で XDP マルチバッファモードを使用するには、RX Striding RQ を無効にする必要がある

次の条件にすべて一致するホストにおいて、マルチバッファモードを使用した eXpress Data Path (XDP) スクリプトの実行が失敗します。

- ホストは `mlx5` ドライバーを使用しています。
- 最大伝送単位 (MTU) の値が 3498 バイトを超えています。
- Mellanox インターフェイスで、受信ストライディング受信キュー (RX Striding RQ) 機能が有効になっています。

すべての条件が当てはまる場合、スクリプトは `link set xdp fd failed` エラーで失敗します。MTU が高いホストで XDP スクリプトを実行するには、Mellanox インターフェイスで RX Striding RQ を無効にします。

```
# ethtool --set-priv-flags <interface_name> rx_striding_rq off
```

これにより、`mlx5` ドライバーを使用し、MTU 値が 3498 バイトを超えるインターフェイスで XDP マルチバッファモードを使用できます。

Jira:RHEL-6496<sup>[1]</sup>

## 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<sup>[1]</sup>

## 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<sup>[1]</sup>

## 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** ユーティリティーおよび **ifdown** ユーティリティーが利用できません。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** ユーティリティー用の NetworkManager ソリューションを提供します。

Bugzilla:2082303

## Mellanox ConnectX-5 アダプターの使用中に mlx5 ドライバーが失敗する

イーサネットスイッチデバイスドライバーモデル (**switchdev**) モードでは、デバイス管理フローステアリング (DMFS) パラメーターと **ConnectX-5** アダプターがサポートするハードウェアを使用して設定されていると、**mlx5** ドライバーが失敗します。その結果、次のエラーメッセージが表示されることがあります。

BUG: Bad page cache in process umount pfn:142b4b

この問題を回避するには、DMFS の代わりにソフトウェア管理フローステアリング (SMFS) パラメーターを使用します。

Jira:RHEL-9897<sup>[1]</sup>

## 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** ドライバーが事前に定義された EEH フリーズの最大値に達するまで、繰り返し I/O エラーを報告します。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<sup>[1]</sup>

## xdp-loader features コマンドが失敗する



**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ました。その結果、**xdp-loader features** コマンドはエラーで失敗します。

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

回避策はありません。その結果、**xdp-loader features** コマンドを使用してインターフェイス機能を表示することはできません。

Jira:RHEL-3382<sup>[1]</sup>

## 11.8. カーネル

カーネルの **kdump** メカニズムにより、64K カーネルで OOM エラーが発生します。

64 ビット ARM アーキテクチャー上の 64K カーネルページサイズは、4KB カーネルよりも多くのメモリーを使用します。その結果、**kdump** はカーネルパニックを引き起こし、メモリー不足 (OOM) エラーでメモリー割り当てが失敗します。回避策として、**crashkernel** 値を手動で 640 MB に設定します。たとえば、**crashkernel=** パラメーターを **crashkernel=2G- :640M** として設定します。

結果として、説明されているシナリオでは、**kdump** メカニズムは 64K カーネルで失敗しません。

Bugzilla:2160676<sup>[1]</sup>

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<sup>[1]</sup>

## 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<sup>[1]</sup>

## 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<sup>[1]</sup>

### 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<sup>[1]</sup>

### **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。

```
# kdumpctl estimate
```

- 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
```

- 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<sup>[1]</sup>

## 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: iwlfwif 0000:09:00.0: Failed to start RT ucode: -110
kernel: iwlfwif 0000:09:00.0: WRT: Collecting data: ini trigger 13 fired (delay=0ms)
kernel: iwlfwif 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<sup>[1]</sup>

### 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<sup>[1]</sup>

### 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、64 ビット ARM CPU のカーネルヘッダーが、ページサイズが 4 キロバイトのカーネルと 64 キロバイトのカーネルの両方で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<sup>[1]</sup>

## 11.9. ファイルシステムおよびストレージ

### CHAP 認証の試行に失敗した後、no authentication メソッドを使用して iSCSI サーバーにログインできない

CHAP 認証を使用して iSCSI ディスクを追加し、間違った認証情報によりログイン試行に失敗した場合は、**no authentication** 方式でのディスクへの再ログインに失敗します。この問題を回避するには、現行セッションを閉じて、**no authentication** メソッドを使用してログインします。

Bugzilla:1983602<sup>[1]</sup>

### デバイス Mapper マルチパスは NVMe/TCP ではサポートされない

**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

てください。

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<sup>[1]</sup>

### 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<sup>[1]</sup>

### 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<sup>[1]</sup>

### 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<sup>[1]</sup>

### 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<sup>[1]</sup>

## kernel-64k ページサイズの ARM ベースのシステムを起動できない

**vdo** パッケージのインストール中に、4k ページサイズのカーネルが依存関係としてインストールされます。その結果、**Software Selection** 画面で 64k ページサイズを選択した場合でも、システムは 4k ページサイズのカーネルで起動します。この問題を回避するには、**Base Environment** で **Minimal Install** を選択し、**Kernel options** でページサイズとして 64k を選択します。システムを初めて起動するときに、DNF パッケージマネージャーを使用して追加のソフトウェアをインストールします。

[Jira:RHEL-8354](#)

## 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<sup>\[1\]</sup>](#)

### FIPS モードで IdM Vault 暗号化および復号化に失敗する

FIPS モードが有効な場合は、OpenSSL RSA-PKCS1v15 パディング暗号化がブロックされます。その結果、現在は IdM が PKCS1v15 パディングを使用してセッションキーをトランスポート証明書でラップするため、Identity Management (IdM) Vault が正しく機能しません。

Jira:RHEL-12143<sup>[1]</sup>

## 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<sup>[1]</sup>

## 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<sup>[1]</sup>

## 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<sup>[1]</sup>

## 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<sup>[1]</sup>

## EMS 強制により、FIPS モードで RHEL 9.2+ 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

TLS **Extended Master Secret** (EMS) 拡張機能 (RFC 7627) は、FIPS 対応の RHEL 9.2 以降のシステムでの TLS 1.2 接続に必須になりました。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](#)

## nsslapd-numlisteners 属性値が 2 より大きい場合、Directory Server が失敗する

**nsslapd-numlisteners** 属性値が 2 より大きい場合、Directory Server は、受け入れたファイル記述子ではなく、リッスンしているファイル記述子を閉じる場合があります。その結果、しばらくすると、Directory Server は一部のポートでのリッスンを停止し、障害が発生します。

この問題を回避するには、**nsslapd-numlisteners** 属性値を 1 に設定します。

[Jira:RHEL-17178<sup>\[1\]</sup>](#)

## 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

RHEL 8 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<sup>\[1\]</sup>](#)

### 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時失敗します。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<sup>\[1\]</sup>](#)

### 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<sup>\[1\]</sup>](#)

### 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<sup>\[1\]</sup>](#)

## 11.14. 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](#)

## 11.15. 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場合失敗します。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<sup>\[1\]</sup>](#)

### 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る

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 **erms** および **fsrm** を手動で有効にします。

[Bugzilla:2077767<sup>\[1\]</sup>](#)

## 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き失敗し、代わりに VM でカーネルパニックが発生します。

[Bugzilla:2005173<sup>\[1\]</sup>](#)

## 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、VM が予期せず終了したり、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](#)

## 多数のキューを使用すると、仮想マシンで障害が発生することがある

仮想 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) が失敗することがあります。

この問題は、vTPM デバイスの制限が原因で発生します。vTPM デバイスには、開いているファイル記述子の最大数に関するハードコーディングされた制限があります。新しいキューごとに複数のファイル記述子が開かれるため、内部の vTPM 制限を超えて VM が失敗する可能性があります。

この問題を回避するには、次の 2 つのオプションのいずれかを選択します。

- vTPM デバイスを有効のままにしますが、使用するキューは 250 未満にします。
- 250 を超えるキューを使用するには、vTPM デバイスを無効にします。

Jira:RHEL-13335<sup>[1]</sup>

### 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せん。

```
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<sup>[1]</sup>

### AMD EPYC CPU を搭載したホストで v2v 変換後に一部の Windows ゲストが起動に失敗する

**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、現在 VM は起動に失敗します。これは、AMD EPYC シリーズ CPU を使用するホストで発生します。

Bugzilla:2168082<sup>[1]</sup>

### 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<sup>\[1\]</sup>](#)

### FIFO スケジューラーを備えた RT 仮想マシンが起動できない

現在、vCPU スケジューラーに **fifo** 設定を使用するようにリアルタイム (RT) 仮想マシンを設定すると、仮想マシンを起動しようとしても応答しません。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す。

[Jira:RHEL-2815<sup>\[1\]</sup>](#)

## 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<sup>[1]</sup>

## 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<sup>[1]</sup>

## 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<sup>[1]</sup>

## 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<sup>[1]</sup>

## ネットワーク負荷が高い状態で Windows 11 または Windows Server 2022 仮想マシンを移行すると失敗することがある

Windows Server 2022 または Windows 11 をゲストオペレーティングシステムとして使用する仮想マシン (VM) をライブマイグレーションする場合に、ネットワークが高いパックスロスによる影響を受けると、マイグレーションが応答しなくなったり、予期せず終了したりすることがあります。

Jira:RHEL-2316<sup>[1]</sup>

## コピー後の仮想マシン移行の再開が失敗する場合があります

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場合がある

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<sup>\[1\]</sup>](#)

### 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<sup>\[1\]</sup>](#)

## 11.16. 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<sup>[1]</sup>

### 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<sup>[1]</sup>

### 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<sup>[1]</sup>

### 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 sulin(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<sup>[1]</sup>

## 11.17. 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<sup>[1]</sup>

## 11.18. 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<sup>[1]</sup>

## 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
<b>389-ds-base</b>	<a href="#">Bugzilla:2188627</a> 、 <a href="#">Bugzilla:1987471</a> 、 <a href="#">Bugzilla:2149025</a> 、 <a href="#">Bugzilla:2166332</a> 、 <a href="#">Bugzilla:2189946</a> 、 <a href="#">Bugzilla:2189954</a> 、 <a href="#">Bugzilla:1975930</a> 、 <a href="#">Bugzilla:1974242</a> 、 <a href="#">Bugzilla:1759941</a> 、 <a href="#">Bugzilla:2053204</a> 、 <a href="#">Bugzilla:2116948</a> 、 <a href="#">Bugzilla:2179278</a> 、 <a href="#">Bugzilla:2189717</a> 、 <a href="#">Bugzilla:2170494</a> 、 <a href="#">Bugzilla:2098236</a> 、 <a href="#">Jira:RHEL-17178</a>
<b>NetworkManager</b>	<a href="#">Bugzilla:2176137</a> 、 <a href="#">Bugzilla:2161915</a> 、 <a href="#">Bugzilla:2151986</a> 、 <a href="#">Bugzilla:2190375</a> 、 <a href="#">Bugzilla:2069001</a> 、 <a href="#">Bugzilla:2069004</a> 、 <a href="#">Bugzilla:2148684</a> 、 <a href="#">Bugzilla:2158328</a> 、 <a href="#">Bugzilla:2180966</a> 、 <a href="#">Bugzilla:2151040</a> 、 <a href="#">Bugzilla:1894877</a>
<b>リリースノート</b>	<a href="#">Jira:RHELDOCS-16861</a> 、 <a href="#">Jira:RHELDOCS-16760</a> 、 <a href="#">Jira:RHELDOCS-16756</a> 、 <a href="#">Jira:RHELDOCS-16612</a> 、 <a href="#">Jira:RHELDOCS-17102</a> 、 <a href="#">Jira:RHELDOCS-16979</a>
<b>anaconda</b>	<a href="#">Bugzilla:2171811</a> 、 <a href="#">Bugzilla:2164819</a> 、 <a href="#">Bugzilla:2177219</a> 、 <a href="#">Bugzilla:2157921</a> 、 <a href="#">Bugzilla:2065754</a> 、 <a href="#">Bugzilla:2107346</a> 、 <a href="#">Bugzilla:2127473</a> 、 <a href="#">Bugzilla:2050140</a> 、 <a href="#">Bugzilla:1877697</a> 、 <a href="#">Jira:RHEL-4707</a> 、 <a href="#">Jira:RHEL-4711</a> 、 <a href="#">Bugzilla:1997832</a> 、 <a href="#">Jira:RHEL-4741</a> 、 <a href="#">Bugzilla:2115783</a> 、 <a href="#">Jira:RHEL-4762</a> 、 <a href="#">Bugzilla:2163497</a> 、 <a href="#">Jira:RHEL-4737</a>
<b>ansible-freeipa</b>	<a href="#">Bugzilla:2175767</a> 、 <a href="#">Bugzilla:2127903</a> 、 <a href="#">Bugzilla:2127907</a>
<b>audit</b>	<a href="#">Jira:RHELPLAN-161087</a>
<b>bacula</b>	<a href="#">Jira:RHEL-6856</a>
<b>bind</b>	<a href="#">Bugzilla:1984982</a>
<b>cloud-init</b>	<a href="#">Bugzilla:2118235</a> 、 <a href="#">Bugzilla:2172341</a> 、 <a href="#">Jira:RHEL-12122</a>
<b>cockpit</b>	<a href="#">Bugzilla:2203361</a>
<b>cockpit-appstream</b>	<a href="#">Bugzilla:2030836</a>
<b>cockpit-machines</b>	<a href="#">Bugzilla:2173584</a>
<b>crash</b>	<a href="#">Bugzilla:2170283</a>
<b>createrepo_c</b>	<a href="#">Bugzilla:2056318</a>
<b>crypto-policies</b>	<a href="#">Bugzilla:2216257</a> 、 <a href="#">Bugzilla:2193324</a> 、 <a href="#">Jira:RHEL-591</a> 、 <a href="#">Bugzilla:2225222</a>

コンポーネント	チケット
<b>cups-filters</b>	<a href="#">Bugzilla:2229784</a>
<b>cyrus-sasl</b>	<a href="#">Bugzilla:1995600</a>
<b>debugedit</b>	<a href="#">Bugzilla:2177302</a>
<b>device-mapper-multipath</b>	<a href="#">Jira:RHEL-782</a> , <a href="#">Bugzilla:2164869</a> , <a href="#">Bugzilla:2033080</a> , <a href="#">Bugzilla:2011699</a> , <a href="#">Bugzilla:1926147</a>
<b>device-mapper-persistent-data</b>	<a href="#">Bugzilla:2175198</a>
<b>dnf</b>	<a href="#">Bugzilla:2124793</a> , <a href="#">Bugzilla:2212262</a> , <a href="#">Bugzilla:2073510</a>
<b>dnf-plugins-core</b>	<a href="#">Bugzilla:2157844</a> , <a href="#">Bugzilla:2134638</a> , <a href="#">Bugzilla:2203100</a>
<b>edk2</b>	<a href="#">Bugzilla:1935497</a>
<b>elfutils</b>	<a href="#">Bugzilla:2182061</a> , <a href="#">Bugzilla:2182059</a>
<b>fapolicyd</b>	<a href="#">Jira:RHEL-624</a> , <a href="#">Jira:RHEL-622</a> , <a href="#">Jira:RHEL-817</a> , <a href="#">Bugzilla:2054740</a> , <a href="#">Jira:RHEL-520</a>
<b>fence-agents</b>	<a href="#">Bugzilla:2187327</a>
<b>fuse3</b>	<a href="#">Bugzilla:2188182</a>
<b>gcc</b>	<a href="#">Bugzilla:2193180</a> , <a href="#">Bugzilla:2168204</a> , <a href="#">Bugzilla:2208908</a>
<b>gcc-toolset-13</b>	<a href="#">Bugzilla:2171919</a>
<b>gcc-toolset-13-annobin</b>	<a href="#">Bugzilla:2171923</a>
<b>gcc-toolset-13-binutils</b>	<a href="#">Bugzilla:2171926</a>
<b>gcc-toolset-13-gcc</b>	<a href="#">Bugzilla:2172093</a>
<b>gcc-toolset-13-gdb</b>	<a href="#">Bugzilla:2172096</a>
<b>gfs2-utils</b>	<a href="#">Bugzilla:2170017</a>
<b>gimp</b>	<a href="#">Bugzilla:2047161</a>
<b>glibc</b>	<a href="#">Bugzilla:2169978</a> , <a href="#">Bugzilla:2213907</a> , <a href="#">Bugzilla:2177235</a>



コンポーネント	チケット
<b>gnupg2</b>	<a href="#">Bugzilla:2073567</a> , <a href="#">Bugzilla:2070722</a>
<b>gnutls</b>	<a href="#">Bugzilla:2157953</a> , <a href="#">Bugzilla:2108532</a>
<b>golang</b>	<a href="#">Bugzilla:2185259</a> , <a href="#">Bugzilla:2111072</a> , <a href="#">Bugzilla:2092016</a>
<b>grafana</b>	<a href="#">Bugzilla:2193018</a> , <a href="#">Bugzilla:2190025</a>
<b>grub2</b>	<a href="#">Bugzilla:2184069</a>
<b>gssproxy</b>	<a href="#">Bugzilla:2181465</a>
<b>gtk3</b>	<a href="#">Jira:RHEL-11924</a>
<b>httpd</b>	<a href="#">Bugzilla:2184403</a> , <a href="#">Bugzilla:2173295</a>
<b>ipa</b>	<a href="#">Bugzilla:2196426</a> , <a href="#">Bugzilla:2165880</a> , <a href="#">Bugzilla:2229712</a> , <a href="#">Bugzilla:2227831</a> , <a href="#">Bugzilla:2084180</a> , <a href="#">Bugzilla:2084166</a> , <a href="#">Bugzilla:2069202</a> , <a href="#">Bugzilla:2094673</a> , <a href="#">Bugzilla:2057471</a> , <a href="#">Jira:RHEL-12154</a> , <a href="#">Jira:RHEL-12143</a> , <a href="#">Jira:RHEL-4955</a>
<b>iproute</b>	<a href="#">Jira:RHEL-428</a>
<b>java-17-openjdk</b>	<a href="#">Bugzilla:2186647</a>
<b>jmc-core</b>	<a href="#">Bugzilla:1980981</a>
<b>kdump-anaconda-addon</b>	<a href="#">Jira:RHEL-11196</a>
<b>kernel</b>	<a href="#">Bugzilla:1898184</a> , <a href="#">Bugzilla:2177180</a> , <a href="#">Bugzilla:2144528</a> , <a href="#">Bugzilla:2210263</a> , <a href="#">Bugzilla:2180124</a> , <a href="#">Bugzilla:2192730</a> , <a href="#">Bugzilla:2178741</a> , <a href="#">Bugzilla:2195986</a> , <a href="#">Bugzilla:2208365</a> , <a href="#">Bugzilla:2187856</a> , <a href="#">Bugzilla:2192722</a> , <a href="#">Bugzilla:2171093</a> , <a href="#">Bugzilla:2189292</a> , <a href="#">Bugzilla:2193330</a> , <a href="#">Bugzilla:2178930</a> , <a href="#">Bugzilla:2092194</a> , <a href="#">Bugzilla:2101598</a> , <a href="#">Bugzilla:2218207</a> , <a href="#">Bugzilla:2173947</a> , <a href="#">Bugzilla:2178956</a> , <a href="#">Bugzilla:2173594</a> , <a href="#">Bugzilla:1613522</a> , <a href="#">Bugzilla:1874182</a> , <a href="#">Bugzilla:1995338</a> , <a href="#">Bugzilla:1570255</a> , <a href="#">Bugzilla:2177256</a> , <a href="#">Bugzilla:2178699</a> , <a href="#">Bugzilla:2023416</a> , <a href="#">Bugzilla:2021672</a> , <a href="#">Bugzilla:2027304</a> , <a href="#">Bugzilla:1660337</a> , <a href="#">Bugzilla:1955275</a> , <a href="#">Bugzilla:2142102</a> , <a href="#">Bugzilla:2068237</a> , <a href="#">Bugzilla:2040643</a> , <a href="#">Bugzilla:2186375</a> , <a href="#">Bugzilla:2183538</a> , <a href="#">Bugzilla:2206599</a> , <a href="#">Bugzilla:2167783</a> , <a href="#">Bugzilla:2000616</a> , <a href="#">Bugzilla:2013650</a> , <a href="#">Bugzilla:2132480</a> , <a href="#">Bugzilla:2059545</a> , <a href="#">Bugzilla:2005173</a> , <a href="#">Bugzilla:2128610</a> , <a href="#">Bugzilla:2129288</a> , <a href="#">Bugzilla:2013884</a> , <a href="#">Bugzilla:2149989</a>
<b>kernel / Networking / IPSec</b>	<a href="#">Jira:RHEL-1015</a>
<b>kernel / Networking / NIC Drivers</b>	<a href="#">Jira:RHEL-6496</a> , <a href="#">Jira:RHEL-9897</a> , <a href="#">Jira:RHEL-15404</a>

コンポーネント	チケット
<b>kernel / Platform Enablement / NVMe</b>	<a href="#">Jira:RHEL-8171</a> 、 <a href="#">Jira:RHEL-8164</a>
<b>kernel / Storage / Storage Drivers</b>	<a href="#">Jira:RHEL-8466</a>
<b>kernel / Virtualization / KVM</b>	<a href="#">Jira:RHEL-7212</a> 、 <a href="#">Jira:RHEL-2815</a>
<b>kernel-rt</b>	<a href="#">Bugzilla:2181571</a>
<b>kernel-rt / Other</b>	<a href="#">Jira:RHEL-9318</a>
<b>kexec-tools</b>	<a href="#">Bugzilla:2083475</a> 、 <a href="#">Bugzilla:2173815</a> 、 <a href="#">Bugzilla:2169720</a> 、 <a href="#">Bugzilla:2160676</a> 、 <a href="#">Bugzilla:2113873</a> 、 <a href="#">Bugzilla:2064708</a>
<b>keylime</b>	<a href="#">Jira:RHEL-595</a> 、 <a href="#">Jira:RHEL-11866</a> 、 <a href="#">Jira:RHEL-392</a> 、 <a href="#">Jira:RHEL-393</a> 、 <a href="#">Jira:RHEL-947</a> 、 <a href="#">Jira:RHEL-1252</a> 、 <a href="#">Jira:RHEL-11867</a> 、 <a href="#">Jira:RHEL-1518</a>
<b>keylime-agent-rust</b>	<a href="#">Jira:RHEL-476</a> 、 <a href="#">Jira:RHEL-395</a> 、 <a href="#">Jira:RHEL-396</a>
<b>kmod</b>	<a href="#">Bugzilla:2103605</a>
<b>kmod-kvdo</b>	<a href="#">Jira:RHEL-8354</a>
<b>krb5</b>	<a href="#">Bugzilla:2178298</a> 、 <a href="#">Bugzilla:2155607</a> 、 <a href="#">Jira:RHEL-4902</a> 、 <a href="#">Bugzilla:2060798</a> 、 <a href="#">Jira:RHEL-4875</a> 、 <a href="#">Jira:RHEL-4889</a> 、 <a href="#">Bugzilla:2060421</a> 、 <a href="#">Bugzilla:2016312</a> 、 <a href="#">Jira:RHEL-4888</a>
<b>libabigail</b>	<a href="#">Bugzilla:2186931</a>
<b>libotr</b>	<a href="#">Bugzilla:2086562</a>
<b>libpfm</b>	<a href="#">Bugzilla:2185652</a>
<b>libvirt</b>	<a href="#">Bugzilla:2032406</a> 、 <a href="#">Bugzilla:2168499</a> 、 <a href="#">Bugzilla:2014487</a> 、 <a href="#">Bugzilla:2143158</a> 、 <a href="#">Bugzilla:2078693</a>
<b>libxcrypt</b>	<a href="#">Bugzilla:2034569</a>
<b>llvm-toolset</b>	<a href="#">Bugzilla:2178796</a>
<b>lvm2</b>	<a href="#">Bugzilla:2038183</a>
<b>mysql</b>	<a href="#">Bugzilla:1991500</a>

コンポーネント	チケット
<b>nfs-utils</b>	<a href="#">Bugzilla:2081114</a>
<b>nginx-1.22-module</b>	<a href="#">Bugzilla:2170808</a>
<b>nmstate</b>	<a href="#">Bugzilla:2179916</a> , <a href="#">Bugzilla:2180795</a> , <a href="#">Bugzilla:2177733</a> , <a href="#">Bugzilla:2183214</a> , <a href="#">Bugzilla:2187622</a>
<b>nodejs</b>	<a href="#">Bugzilla:2186717</a>
<b>nss</b>	<a href="#">Bugzilla:2157950</a>
<b>nvme-cli</b>	<a href="#">Bugzilla:2159929</a>
<b>nvme-stas</b>	<a href="#">Bugzilla:1893841</a>
<b>open-vm-tools</b>	<a href="#">Bugzilla:2037657</a>
<b>opencryptoki</b>	<a href="#">Bugzilla:2160061</a>
<b>opensc</b>	<a href="#">Jira:RHEL-280</a>
<b>openscap</b>	<a href="#">Bugzilla:2217442</a> , <a href="#">Bugzilla:2161499</a>
<b>openslp</b>	<a href="#">Jira:RHEL-6995</a>
<b>openssh</b>	<a href="#">Bugzilla:2070163</a> , <a href="#">Bugzilla:2056884</a>
<b>openssl</b>	<a href="#">Bugzilla:2216256</a> , <a href="#">Bugzilla:2153471</a> , <a href="#">Bugzilla:2188180</a> , <a href="#">Bugzilla:2160797</a> , <a href="#">Bugzilla:2168665</a> , <a href="#">Bugzilla:1975836</a> , <a href="#">Bugzilla:1681178</a> , <a href="#">Bugzilla:1685470</a>
<b>osbuild</b>	<a href="#">Jira:RHEL-4655</a>
<b>osbuild-composer</b>	<a href="#">Bugzilla:2173928</a> , <a href="#">Jira:RHEL-7999</a> , <a href="#">Jira:RHEL-4649</a>
<b>oscap-anaconda-addon</b>	<a href="#">Bugzilla:2172264</a> , <a href="#">Jira:RHEL-1824</a>
<b>pacemaker</b>	<a href="#">Bugzilla:2189301</a> , <a href="#">Bugzilla:2182482</a>
<b>papi</b>	<a href="#">Bugzilla:2111923</a> , <a href="#">Bugzilla:2186927</a> , <a href="#">Bugzilla:2215582</a>
<b>pause-container</b>	<a href="#">Bugzilla:2106816</a>
<b>pcp</b>	<a href="#">Bugzilla:2175602</a> , <a href="#">Bugzilla:2185803</a>

コンポーネント	チケット
<b>pcs</b>	<a href="#">Bugzilla:2168155</a> , <a href="#">Bugzilla:2163953</a> , <a href="#">Bugzilla:2175881</a> , <a href="#">Bugzilla:2182810</a> , <a href="#">Bugzilla:1423473</a> , <a href="#">Bugzilla:2177996</a> , <a href="#">Bugzilla:1860626</a> , <a href="#">Bugzilla:2163914</a>
<b>pcsc-lite-ccid</b>	<a href="#">Bugzilla:2209457</a>
<b>perl-HTTP-Tiny</b>	<a href="#">Bugzilla:2228412</a>
<b>pki-core</b>	<a href="#">Bugzilla:2084181</a>
<b>podman</b>	<a href="#">Jira:RHELPLAN-154314</a> , <a href="#">Jira:RHELPLAN-154432</a> , <a href="#">Jira:RHELPLAN-154441</a> , <a href="#">Jira:RHELPLAN-154438</a> , <a href="#">Jira:RHELPLAN-163003</a> , <a href="#">Jira:RHELPLAN-160660</a> , <a href="#">Jira:RHELPLAN-154429</a> , <a href="#">Bugzilla:2069279</a>
<b>postfix</b>	<a href="#">Bugzilla:2134789</a>
<b>python-greenlet</b>	<a href="#">Bugzilla:2149497</a>
<b>python3.11-lxml</b>	<a href="#">Bugzilla:2157708</a>
<b>qemu-kvm</b>	<a href="#">Bugzilla:1880531</a> , <a href="#">Bugzilla:1965079</a> , <a href="#">Bugzilla:1951814</a> , <a href="#">Bugzilla:2060839</a> , <a href="#">Bugzilla:2014229</a> , <a href="#">Jira:RHEL-7335</a> , <a href="#">Jira:RHEL-7336</a> , <a href="#">Bugzilla:1915715</a> , <a href="#">Jira:RHEL-13335</a> , <a href="#">Jira:RHEL-333</a> , <a href="#">Bugzilla:2176010</a> , <a href="#">Bugzilla:2058982</a> , <a href="#">Bugzilla:2073872</a>
<b>qemu-kvm / Devices</b>	<a href="#">Jira:RHEL-1220</a>
<b>qemu-kvm / Graphics</b>	<a href="#">Jira:RHEL-7135</a>
<b>qemu-kvm / Live Migration</b>	<a href="#">Jira:RHEL-7096</a> , <a href="#">Jira:RHEL-2316</a> , <a href="#">Jira:RHEL-7115</a>
<b>qemu-kvm / Networking</b>	<a href="#">Jira:RHEL-7337</a>
<b>rear</b>	<a href="#">Bugzilla:2188593</a> , <a href="#">Bugzilla:2172912</a> , <a href="#">Bugzilla:2196445</a> , <a href="#">Bugzilla:2145014</a>
<b>redis</b>	<a href="#">Bugzilla:2129826</a>
<b>resource-agents</b>	<a href="#">Bugzilla:2174911</a> , <a href="#">Bugzilla:2142518</a> , <a href="#">Bugzilla:2142002</a> , <a href="#">Bugzilla:2182415</a> , <a href="#">Bugzilla:2179003</a>
<b>restore</b>	<a href="#">Bugzilla:1997366</a>

コンポーネント	チケット
<b>rhel-system-roles</b>	<a href="#">Bugzilla:2224384</a> , <a href="#">Bugzilla:2216753</a> , <a href="#">Bugzilla:2224385</a> , <a href="#">Bugzilla:2185065</a> , <a href="#">Bugzilla:2181656</a> , <a href="#">Bugzilla:2211194</a> , <a href="#">Bugzilla:2218592</a> , <a href="#">Bugzilla:2211723</a> , <a href="#">Bugzilla:2218204</a> , <a href="#">Bugzilla:2151373</a> , <a href="#">Bugzilla:2179460</a> , <a href="#">Bugzilla:2211748</a> , <a href="#">Bugzilla:2229802</a> , <a href="#">Bugzilla:2181657</a> , <a href="#">Bugzilla:2168692</a> , <a href="#">Bugzilla:2211984</a> , <a href="#">Bugzilla:2232241</a> , <a href="#">Bugzilla:2232231</a> , <a href="#">Bugzilla:2224090</a> , <a href="#">Bugzilla:2222761</a> , <a href="#">Bugzilla:2223764</a> , <a href="#">Bugzilla:2222428</a> , <a href="#">Bugzilla:2216520</a> , <a href="#">Bugzilla:2211187</a> , <a href="#">Bugzilla:2209200</a> , <a href="#">Bugzilla:2193058</a> , <a href="#">Bugzilla:2186057</a> , <a href="#">Jira:RHEL-1499</a> , <a href="#">Jira:RHEL-1397</a> , <a href="#">Jira:RHEL-906</a> , <a href="#">Jira:RHEL-1495</a> , <a href="#">Jira:RHEL-898</a> , <a href="#">Jira:RHEL-885</a> , <a href="#">Bugzilla:1999770</a> , <a href="#">Bugzilla:2123859</a> , <a href="#">Jira:RHEL-1172</a> , <a href="#">Bugzilla:2186218</a>
<b>rpm</b>	<a href="#">Bugzilla:2157836</a>
<b>rsyslog</b>	<a href="#">Jira:RHELPLAN-160541</a>
<b>rust</b>	<a href="#">Bugzilla:2191743</a> , <a href="#">Bugzilla:2227082</a>
<b>s390utils</b>	<a href="#">Bugzilla:1932480</a>
<b>samba</b>	<a href="#">Bugzilla:2190415</a>
<b>scap-security-guide</b>	<a href="#">Bugzilla:2221697</a> , <a href="#">Bugzilla:2155790</a> , <a href="#">Jira:RHEL-1905</a> , <a href="#">Bugzilla:2203791</a> , <a href="#">Bugzilla:2213958</a> , <a href="#">Bugzilla:2223178</a> , <a href="#">Bugzilla:2193169</a> , <a href="#">Jira:RHEL-1800</a> , <a href="#">Bugzilla:2038978</a>
<b>selinux-policy</b>	<a href="#">Bugzilla:2080443</a> , <a href="#">Bugzilla:2170495</a> , <a href="#">Bugzilla:2184999</a> , <a href="#">Bugzilla:2162663</a> , <a href="#">Bugzilla:2112729</a> , <a href="#">Jira:RHELPLAN-163014</a> , <a href="#">Bugzilla:2187745</a> , <a href="#">Bugzilla:2229722</a> , <a href="#">Bugzilla:2064274</a>
<b>setools</b>	<a href="#">Bugzilla:2231801</a>
<b>sevctl</b>	<a href="#">Bugzilla:2104857</a>
<b>sos</b>	<a href="#">Bugzilla:1869561</a>
<b>squid-container</b>	<a href="#">Bugzilla:2178953</a>
<b>sssd</b>	<a href="#">Bugzilla:2065693</a> , <a href="#">Bugzilla:2056482</a> , <a href="#">Bugzilla:1608496</a>
<b>stratisd</b>	<a href="#">Bugzilla:2041558</a>
<b>subscription-manager</b>	<a href="#">Bugzilla:2163716</a> , <a href="#">Bugzilla:2136694</a>
<b>sysstat</b>	<a href="#">Jira:RHEL-12009</a>

コンポーネント	チケット
<b>systemd</b>	<a href="#">Bugzilla:2018112</a> 、 <a href="#">Jira:RHEL-6105</a>
<b>systemtap</b>	<a href="#">Bugzilla:2186934</a>
<b>tang</b>	<a href="#">Bugzilla:2188743</a>
<b>tigervnc</b>	<a href="#">Bugzilla:2060308</a>
<b>tuned</b>	<a href="#">Bugzilla:2113900</a>
<b>ubi9-micro-container</b>	<a href="#">Bugzilla:2223028</a>
<b>udisks2</b>	<a href="#">Bugzilla:1983602</a> 、 <a href="#">Bugzilla:2213769</a>
<b>unbound</b>	<a href="#">Bugzilla:2070495</a>
<b>valgrind</b>	<a href="#">Bugzilla:2124346</a>
<b>virt-v2v</b>	<a href="#">Bugzilla:2168082</a>
<b>virtio-win</b>	<a href="#">Bugzilla:1969724</a> 、 <a href="#">Jira:RHEL-11366</a> 、 <a href="#">Jira:RHEL-910</a> 、 <a href="#">Jira:RHEL-1609</a> 、 <a href="#">Jira:RHEL-869</a>
<b>virtio-win / distribution</b>	<a href="#">Jira:RHEL-1517</a> 、 <a href="#">Jira:RHEL-574</a>
<b>virtio-win / virtio-win-prewhql</b>	<a href="#">JIRA:RHEL-1084</a> 、 <a href="#">JIRA:RHEL-935</a> 、 <a href="#">JIRA:RHEL-12118</a> 、 <a href="#">JIRA:RHEL-1212</a>
<b>webkit2gtk3</b>	<a href="#">Jira:RHEL-4157</a>
<b>which</b>	<a href="#">Bugzilla:2181974</a>
<b>xdp-tools</b>	<a href="#">Bugzilla:2218500</a> 、 <a href="#">Jira:RHEL-3382</a>

コンポーネント	チケット
その他	Bugzilla:2232554、Jira:RHELDOCS-17055、Jira:RHELPLAN-163133、Jira:RHELPLAN-163665、Jira:RHELDOCS-16405、Jira:RHELDOCS-16247、Bugzilla:2136937、Jira:RHELDOCS-16474、Jira:RHELDOCS-16462、Jira:RHELDOCS-16386、Jira:RHELPLAN-156196、Jira:RHELDOCS-16708、Jira:RHELDOCS-16709、Jira:RHELDOCS-16339、Jira:RHELDOCS-16877、Jira:RHELPLAN-122345、Jira:RHELDOCS-16487、Jira:RHELDOCS-16752、Jira:RHELDOCS-17101、Bugzilla:2236182、Jira:RHELDOCS-17040、Bugzilla:2020529、Bugzilla:2030412、Jira:RHELPLAN-103993、Jira:RHELPLAN-27394、Jira:RHELPLAN-27737、Jira:RHELDOCS-16861、Jira:RHELDOCS-17050、Bugzilla:1927780、Jira:RHELPLAN-110763、Bugzilla:1935544、Bugzilla:2089200、Jira:RHELDOCS-16948、Jira:RHELPLAN-99136、Jira:RHELDOCS-17380、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:RHELPLAN-157225、Jira:RHELPLAN-157337、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、Bugzilla:2173992、Bugzilla:2185048、Bugzilla:1970830、Jira:RHELDOCS-16574

## 付録B 改訂履歴

### 0.1-9

2024年4月18日(木)、Gabriela Fialová ([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 機能拡張 [RHEL-19142](#) (ネットワーク) を追加しました。

### 0.1-8

2024年4月11日(木)、Gabriela Fialová ([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 機能拡張 [BZ#1513934](#) (IdM) を追加しました。

### 0.1-7

2024年3月14日(木)、Gabriela Fialová ([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 既知の問題 [JIRA:RHEL-25967](#) (カーネル) を追加しました。

### 0.1-6

2024年3月4日(月)、Gabriela Fialová ([gfialova@redhat.com](mailto:gfialova@redhat.com))

- バグ修正 [Jira:SSSD-6096](#) (アイデンティティ管理) を追加しました

### 0.1-5

2024年2月28日(水)、Gabriela Fialová ([gfialova@redhat.com](mailto:gfialova@redhat.com))

- バグ修正 [RHEL-8171](#) (ストレージ) の既知の問題を更新しました

### 0.1-4

2024年2月7日(水)、Lucie Vařáková ([lvarakova@redhat.com](mailto:lvarakova@redhat.com))

- 新機能 [RHEL-14694](#) (ネットワークング) を追加しました

### 0.1-3

2024年2月1日(木)、Gabriela Fialová ([gfialova@redhat.com](mailto:gfialova@redhat.com))

- KI [BZ#1834716](#) (セキュリティ) を追加しました
- 非推奨の機能 [RHELDOCS-16756](#) (コンテナツール) を更新しました

### 0.1-2

2024年1月29日(月)、Gabriela Fialová ([gfialova@redhat.com](mailto:gfialova@redhat.com))

- バグ修正 [RHELPLAN-157337](#) (セキュリティ) を追加しました

### 0.1-1

2024年1月、Lenka Špačková ([lspackova@redhat.com](mailto:lspackova@redhat.com))

- Python [RHELDOCS-17369](#) (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) に関連する機能拡張を追加しました

### 0.1-0

2024年1月10日(水)、Gabriela Fialová ([gfialova@redhat.com](mailto:gfialova@redhat.com))



- 非推奨の機能 [RHELDOCS-17380](#) (セキュリティー) を追加しました

#### 0.0-9

2024年1月2日(火)、Gabriela Fialová ([gfialova@redhat.com](mailto:gfialova@redhat.com))

- 機能拡張 [BZ#2184403](#) の説明を更新しました

#### 0.0-8

2023年11月23日(木)、Gabriela Fialová ([gfialova@redhat.com](mailto:gfialova@redhat.com))

- KI [RHEL-8354](#) (インストーラー) を追加しました

#### 0.0-7

2023年11月22日(水)、Gabriela Fialová ([gfialova@redhat.com](mailto:gfialova@redhat.com))

- IdM KI [RHEL-17178](#) を追加しました

#### 0.0-6

2023年11月21日(火)、David Vozenilek ([dvozenil@redhat.com](mailto:dvozenil@redhat.com))

- システムロール RN を追加しました [BZ#2211723](#)、[BZ#2218204](#)、[BZ#2186057](#)

#### 0.0-5

2023年11月20日(月)、Jana Heves ([jsvarova@redhat.com](mailto:jsvarova@redhat.com))

- KI [RHEL-15404](#) sst\_kernel\_generalists を追加しました

#### 0.0-4

2023年11月19日(日)、Filip Hanzelka ([fhanzelk@redhat.com](mailto:fhanzelk@redhat.com))

- IdM [RHELDOCS-17011](#) に BF を追加しました

#### 0.0-3

2023年11月16日(木)、Marek Suchánek ([muchane@redhat.com](mailto:muchane@redhat.com))

- Inkscape と LibreOffice Flatpak が非推奨になりました [RHELDOCS-17102](#)

#### 0.0-2

2023年11月16日(木)、Lenka Špačková ([lspackova@redhat.com](mailto:lspackova@redhat.com))

- **Node.js 20** が完全にサポートされるようになりました ([BZ#2186717](#))。

#### 0.0-1

2023年11月8日(水)、Gabriela Fialová ([gfialova@redhat.com](mailto:gfialova@redhat.com))

- Red Hat Enterprise Linux 9.3 リリースノートのリリース。

#### 0.0-0

2023年9月27日(水)、Gabriela Fialová ([gfialova@redhat.com](mailto:gfialova@redhat.com))

- Red Hat Enterprise Linux 9.3 ベータ版リリースノートのリリース。

