



Red Hat Enterprise Linux 8

8.3 リリースノート

Red Hat Enterprise Linux 8.3 リリースノート

Red Hat Enterprise Linux 8 8.3 リリースノート

Red Hat Enterprise Linux 8.3 リリースノート

法律上の通知

Copyright © 2024 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

本リリースノートでは、Red Hat Enterprise Linux 8.3 での改良点および実装された追加機能の概要、本リリースにおける既知の問題などを説明します。また、重要なバグ修正、テクニカルレビュー、非推奨の機能などの詳細も説明します。

目次

RED HAT ドキュメントへのフィードバック (英語のみ)	3
第1章 概要	4
インストーラーおよびイメージの作成	4
RHEL for Edge	4
インフラストラクチャーサービス	4
セキュリティー	4
動的プログラミング言語、Web サーバー、およびデータベースサーバー	4
コンパイラツールセット	5
ID 管理	5
Web コンソール	5
仮想化	5
デスクトップおよびグラフィックス	5
インプレースアップグレードおよび OS 移行	5
OpenJDK 11 が利用可能に	6
関連情報	7
Red Hat Customer Portal Labs	7
第2章 アーキテクチャー	8
第3章 RHEL 8 のコンテンツの配布	9
3.1. インストール	9
3.2. リポジトリ	9
3.3. アプリケーションストリーム	10
3.4. YUM/DNF を使用したパッケージ管理	10
第4章 RHEL 8.3.1 リリース	11
4.1. 新機能	11
第5章 RHEL 8.3.0 リリース	15
5.1. 新機能	15
5.2. 外部カーネルパラメーターの重要な変更	71
5.3. デバイスドライバー	77
5.4. バグ修正	79
5.5. テクノロジープレビュー	93
5.6. 非推奨になった機能	104
5.7. 既知の問題	113
第6章 国際化	140
6.1. RED HAT ENTERPRISE LINUX 8 の多言語	140
6.2. RHEL 8 における国際化の主な変更点	140
付録A コンポーネント別のチケットリスト	142
付録B 改訂履歴	150

RED HAT ドキュメントへのフィードバック (英語のみ)

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

Jira からのフィードバック送信 (アカウントが必要)

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

第1章 概要

インストーラーおよびイメージの作成

RHEL 8.3 では、インストールを開始する前に root パスワードを設定し、ユーザーアカウントを作成できます。以前は、インストールプロセス開始後に、root パスワードを設定してユーザーアカウントを作成していました。非常に優れたバックエンドに基づいてカスタマイズしたイメージを作成し、RHEL Web コンソールでイメージをクラウドにプッシュすることもできます。

RHEL for Edge

RHEL 8.3 では、Edge サーバーで RHEL をリモートでインストールするための **RHEL for Edge** が導入されました。RHEL for Edge は、Image Builder を使用して設定できる rpm-ostree イメージです。キックスタートファイルを使用してイメージをインストールし、イメージを管理し、イメージを更新して以前の機能状態にイメージをロールバックすることができます。

RHEL for Edge の主な特色を以下に示します。

- アトミックアップグレード。各更新の状態が分かっており、デバイスを再起動するまで変更が加えられません。
- 回復性を確保するためのカスタムヘルスチェックおよびインテリジェントなロールバック。
- コンテナ中心のワークフロー: アプリケーションの更新からコア OS の更新を分離し、異なるバージョンのアプリケーションをテストし、デプロイすることができます。
- 低帯域幅環境向けに最適化された OTA ペイロード。

詳細は、「[RHEL for Edge](#)」を参照してください。

インフラストラクチャーサービス

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

セキュリティー

RHEL 8.3 は、**Clevis** および **Tang** を使用した PBD (Policy-Based Decryption) ソリューションの自動デプロイメント用の Ansible ロールを提供します。また、このバージョンの **rhel-system-roles** パッケージには、**Rsyslog** を介して RHEL ログイング用の Ansible ロールも含まれます。

scap-security-guide パッケージがバージョン 0.1.50 にリベースされ、**OpenSCAP** がバージョン 1.3.3 にリベースされました。これらの更新では、CIS RHEL 7 Benchmark v2.2.0 に準拠するプロファイルや、北米の医療機関で必要な HIPAA Health Insurance Portability and Accountability Act (HIPAA) に一致するプロファイルなど、大幅に改良されました。

今回の更新で、**SCAP Workbench** ツールを使用して、カスタマイズしたプロファイルから結果ベースの修復ロールを生成できるようになりました。

USBGuard フレームワークは、独自の SELinux ポリシーを提供し、GUI でデスクトップユーザーに通知されるようになり、バージョン 0.7.8 には多くの改善およびバグ修正が含まれています。

動的プログラミング言語、Web サーバー、およびデータベースサーバー

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

- **nginx 1.18**
- **Node.js 14**

- Perl 5.30
- PHP 7.4
- Ruby 2.7

RHEL 8.3 では、以下のコンポーネントが更新されました。

- Git がバージョン 2.27 へ
- squid がバージョン 4.11 へ

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

コンパイラツールセット

RHEL 8.3 では、以下のコンパイラツールセットが更新されました。

- GCC Toolset 10
- LLVM Toolset 10.0.1
- Rust Toolset 1.45.2
- Go Toolset 1.14.7

詳細は、「[コンパイラおよび開発ツール](#)」を参照してください。

ID 管理

RHEL 8 では、Rivest Cipher 4 (RC4) 暗号スイート、ユーザー、サービスのデフォルト暗号化タイプ、および AD フォレスト内の Active Directory (AD) ドメイン間の信頼が非推奨になりました。互換性の理由から、今回の更新で新しい暗号化サブポリシー **AD-SUPPORT** が導入され、非推奨の RC4 暗号化タイプのサポートが可能になります。新しいサブポリシーを使用すると、RHEL Identity Management (IdM) および SSSD Active Directory 統合ソリューションで RC4 を使用できます。

詳細は、「[ID 管理](#)」を参照してください。

Web コンソール

Web コンソールには、ユーザーセッション内から管理アクセスと制限付きアクセスを切り替えるオプションがあります。

仮想化

IBM Z ハードウェアでホストされる仮想マシン (VM) が、IBM Secure Execution 機能を使用できるようになりました。これにより、ホストが侵害された場合の仮想マシンの攻撃に対する耐性が高まり、信頼できないホストが仮想マシンから情報を取得するのを防ぐことができます。さらに、DASD デバイスが IBM Z の仮想マシンに割り当てられるようになりました。

デスクトップおよびグラフィックス

IBM Z システムで GNOME デスクトップを使用できるようになりました。

Direct Rendering Manager (DRM) カーネルグラフィックサブシステムが、アップストリームの Linux カーネルバージョン 5.6 にリベースされました。このバージョンでは、新しい GPU および APU のサポートや、さまざまなドライバー更新など、以前のバージョンから多くの機能強化が行われています。

詳細は、「[デスクトップ](#)」および「[グラフィックインフラストラクチャー](#)」を参照してください。

インプレースアップグレードおよび OS 移行

RHEL 7 から RHEL 8 へのインプレースアップグレード

現在サポートされているインプレースアップグレードパスは次のとおりです。

- 64 ビット Intel、IBM POWER 8 (little endian)、および IBM Z アーキテクチャーでの RHEL 7.8 から RHEL 8.2 のアップグレード。
- カーネルバージョン 4.14 を必要とするアーキテクチャー (IBM POWER 9 (リトルエンディアン) および IBM Z (Structure A)) での RHEL 7.6 から RHEL 8.2 のアップグレード。
- SAP HANA のシステムにおける RHEL 7.7 から RHEL 8.2 へのアップグレード。

RHEL 8.2 へのアップグレード後もシステムがサポートされるようにするには、最新の RHEL 8.3 バージョンに更新するか、RHEL 8.2 Extended Update Support (EUS) リポジトリを有効にします。SAP HANA を使用するシステムで、RHEL 8.2 Update Services for SAP Solutions (E4S) リポジトリを有効にします。

詳細は [Red Hat Enterprise Linux のサポート対象のインプレースアップグレードパス](#) を参照してください。インプレースアップグレードの実行方法は、[RHEL 7 から RHEL 8 へのアップグレード](#) を参照してください。

主な機能強化は、次のとおりです。

- **Leapp** は、アップグレードの続行方法を判断するために、true/false の質問を生成してユーザー入力をサポートするようになりました。
- Satellite Web UI を使用して、複数のホストを同時にアップグレードできるようになりました。
- Red Hat Update Infrastructure (RHUI) を使用して、AWS および Microsoft Azure のオンデマンドインスタンスでサポートされるインプレースアップグレードがサポートされるようになりました。
- [RHBA-2021:0569](#) アドバイザリーのリリースでは、**Leapp** のアップグレード前のレポート用のカスタムスクリプトを作成できます。詳細は、[Automating your Red Hat Enterprise Linux pre-upgrade report workflow](#) を参照してください。

RHEL 6 から RHEL 8 へのインプレースアップグレード

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

別の Linux ディストリビューションから RHEL への移行

CentOS Linux 8 または Oracle Linux 8 を使用している場合は、Red Hat がサポートする **Convert2RHEL** ユーティリティを使用してオペレーティングシステムを RHEL 8 に変換できます。詳細は、[RPM ベースの Linux ディストリビューションから RHEL への変換](#) を参照してください。

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

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

OpenJDK 11 が利用可能に

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

関連情報

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

Red Hat Customer Portal Labs

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

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

第2章 アーキテクチャー

Red Hat Enterprise Linux 8.3 ではカーネルバージョン 4.18.0-240 が使用されており、以下のアーキテクチャーに対応します。

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

各アーキテクチャーに適切なサブスクリプションを購入してください。詳細は [Get Started with Red Hat Enterprise Linux - additional architectures](#) を参照してください。利用可能なサブスクリプションのリストは、カスタマーポータル[のサブスクリプションの使用状況](#)を参照してください。

第3章 RHEL 8 のコンテンツの配布

3.1. インストール

Red Hat Enterprise Linux 8 は、ISO イメージを使用してインストールします。AMD64、Intel 64 ビット、64 ビット ARM、IBM Power Systems、IBM Z アーキテクチャーで、以下の 2 種類のインストールメディアが利用できます。

- Binary DVD ISO - BaseOS リポジトリおよび AppStream リポジトリが含まれ、リポジトリを追加しなくてもインストールを完了できる完全インストールイメージです。



注記

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

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

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

3.2. リポジトリ

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

- BaseOS
- AppStream

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

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

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

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

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

3.3. アプリケーションストリーム

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

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

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

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

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

3.4. YUM/DNF を使用したパッケージ管理

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

詳細は、以下のドキュメントを参照してください。

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

第4章 RHEL 8.3.1 リリース

Red Hat は、マイナーリリース (8.Y) において Red Hat Enterprise Linux 8 のコンテンツを四半期ごとに利用できるようにしています。この四半期リリースは、3桁の数字 (8.Y.1) を使用して番号が付けられます。RHEL 8.3.1 リリースの新機能は以下のとおりです。

4.1. 新機能

複数のデスクトップアプリケーションの Flatpak パッケージ

Flatpak は、グラフィカルアプリケーションをコンテナとして実行するシステムです。Flatpak を使用すると、ホストオペレーティングシステムとは独立してアプリケーションをインストールおよび更新できます。

今回の更新で、Red Hat Container Catalog で以下のアプリケーションの Flatpak コンテナイメージが提供されます。

アプリケーション名	Flatpak コンテナ ID
Firefox	org.mozilla.firefox
GIMP	org.gimp.GIMP
Inkscape	org.inkscape.Inkscape
Thunderbird	org.mozilla.Thunderbird

Red Hat Container Catalog で利用可能な Flatpak コンテナをインストールするには、以下の手順に従います。

1. Flatpak クライアントの最新バージョンがシステムにインストールされていることを確認します。

```
# yum update flatpak
```

2. RHEL Flatpak リポジトリを有効にします。

```
# flatpak remote-add rhel https://flatpaks.redhat.io/rhel.flatpakrepo
```

3. RHEL アカountの認証情報を指定します。

```
# podman login registry.redhat.io
```

デフォルトで、Podman はユーザーがログアウトするまで認証情報を保存します。

4. 必要に応じて、認証情報を永続的に保存します。

```
$ cp $XDG_RUNTIME_DIR/containers/auth.json \
  $HOME/.config/flatpak/oci-auth.json
```

5. Flatpak コンテナイメージをインストールします。


```
$ flatpak install rhel container-id
```

(JIRA:RHELPLAN-

30958、[BZ#1920689](#)、[BZ#1921179](#)、[BZ#1921802](#)、[BZ#1916412](#)、[BZ#1921812](#)、[BZ#1920604](#))

Rust Toolset がバージョン 1.47.0 にリベース

Rust Toolset が、バージョン 1.47.0 に更新されました。以下は、主な変更点です。

- compile-time 評価された関数 **const fn** が改善され、**if**、**while**、**match** などの制御フロー機能を使用できるようになりました。
- 新しい **#[track_caller]** アノテーションを関数に配置できるようになりました。アノテーションが付けられた関数からパニックが発生すると、呼び出し元がソースとして報告されます。
- Rust Standard Library は、任意の長さの配列にトレイトを実装するようになりました。以前のリリースでは、アレイに実装されているトレイトの多くは、長さが 0 から 32 までしか含まれていませんでした。

使用方法の詳細は [Rust Toolset の使用](#) を参照してください。

(BZ#1883839)

Logging システムロールが、出力でプロパティーベースのフィルターをサポート

今回の更新で、プロパティーベースのフィルターが Logging システムロールのファイル出力、転送出力、および `remote_files` の出力に追加されました。この機能は **rsyslog** のサブロールに基づいて提供され、Logging RHEL システムロールを使用して設定できます。その結果、ユーザーは、ホスト名、タグなどのプロパティーでログメッセージをフィルタリングする機能を利用でき、メッセージ自体がログの管理に役立ちます。

(BZ#1889492)

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

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

(BZ#1889893)

ubi8/pause コンテナイメージが利用可能に

Podman は、**k8s.gcr.io/pause** コンテナイメージの代わりに **ubi8/pause** を使用して、Pod のネットワーク名前空間情報を保持するようになりました。

(BZ#1690785)

Podman がバージョン 2.1 にリベース

Podman ユーティリティーがバージョン 2.1 に更新されました。主な機能強化は、次のとおりです。

- 変更点:
 - Podman を 2.05 から 2.2.1 に、Buildah を 1.15.1 から 1.19 に、Skopeo を 1.1.1 から 1.2.1 に、Udica を 0.2.2 から 0.2.3 に、CRIU を 0.3.4 から 3.15 に更新
 - Docker 互換のボリューム API エンドポイント (Create、Inspect、List、Remove、Prune) が利用可能に

- コンテナの `systemd` ユニットファイルを生成するための API エンドポイントを追加
 - **podman play kube** コマンドが、コンテナの CPU およびメモリー制限の設定に対応
 - **podman play kube** コマンドが、Podman の名前付きボリュームを使用した永続ボリューム要求に対応
 - **podman play kube** コマンドが **--configmap** オプションを使用した Kubernetes configmaps に対応
 - 短縮名のエイリアスの実験的なサポートが追加。これはデフォルトでは有効になっていませんが、環境変数 **CONTAINERS_SHORT_NAME_ALIASING** をオンに設定すると有効にすることができます。詳細は、[Container image short name in Podman](#) を参照してください。
 - 新しい **podman image command** が追加されました。これにより、イメージをマウントでき、読み取り専用で、コンテナを作成せずにそのコンテンツを検査できます。
 - **podman save** コマンドおよび **podman load** コマンドが、複数のイメージを含むアーカイブを作成および読み込むことができるようになりました。
 - Podman は、ネットワークエラーが原因でプルが失敗する場合に、最大 3 回でイメージのプルを再試行するようになりました。
- バグ修正:
 - `cgroups v1` システムでコンテナで `systemd` を実行するバグが修正されました。

Buildah ツールがバージョン 1.19 に更新されました。主な機能強化は、次のとおりです。

- 変更点:
 - **buildah inspect** コマンドは、マニフェストの検査に対応
 - **buildah push** コマンドは、マニフェストリストとダイジェストのプッシュに対応
 - **--manifest** フラグのサポートを追加
 - アーキテクチャーおよび OS を選択するために、**--arch** オプション、**--os** オプション、および **--variant** オプションを追加
 - ユーザーが `stdin` をコンテナに指定可能に
 - **FROM** を **--from** オプションで上書き可能に
 - 代替の **.dockerignore** フラグを使用するように **--ignorefile** フラグを追加
 - 短縮名のエイリアス設定
 - **buildah pull** コマンドに **--policy** オプションを追加
 - **buildah mount** コマンドを修正して、ID ではなくコンテナ名を表示
 - `buildah` 補完の強化
 - **--omit-timestamp** フラグではなく **--timestamp** を使用
 - コピーにパイプを使用

- **buildah bud** コマンドに **--omit-timestamp** フラグを追加
- VFS 追加イメージストアをコンテナに追加
- マウントオプションで読み取り専用をエイリアスとして ro に許可
- buildah、bud: 並列実行の **--jobs = N** オプションに対応

Skopeo ツールがバージョン 1.2.1 に更新されました。主な機能強化は、次のとおりです。

- 変更点:
 - Travis を使用してアップストリームおよび安定した skopeo イメージのマルチアーキテクチャビルドを追加
 - 同期時のダイジェストのサポートを追加
 - エミュレートコピー **--all** に **--all** 同期フラグを追加
 - **skopeo inspect** コマンドに **--format** オプションを追加

Udica ツールがバージョン 0.2.3 に更新されました。主な機能強化は、次のとおりです。

- 変更点:
 - ホストポートではなく、コンテナポートを有効化
 - **--version** オプションを追加

CRIU ツールはバージョン 3.15 に更新されました。主な機能強化は、次のとおりです。

- 変更点:
 - 初期 cgroup2 サポート
 - Legalized swrk API と、この API 経由で fd を継承する機能を追加
 - 外部バインドマウントおよび tasks-to-cgroups バインディング
 - **ibcriu.so** (RPC ラッパー) およびプラグイン

(JIRA:RHELPLAN-55998)

第5章 RHEL 8.3.0 リリース

5.1. 新機能

ここでは、Red Hat Enterprise Linux 8.3 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。

5.1.1. インストーラーおよびイメージの作成

Anaconda がバージョン 33.16 にリベース

今回のリリースで、Anaconda がバージョン 33.16 にリベースされました。このバージョンは、以前のバージョンに対する以下の主な機能強化を提供します。

- インストールプログラムは、複数の行に静的 IPv6 アドレスを表示し、ウィンドウのサイズを変更しなくなりました。
- インストールプログラムが、対応している NVDIMM デバイスセクターサイズを表示するようになりました。
- IPv6 静的設定のあるインストール済みのシステムで、ホスト名が正しく設定されるようになりました。
- ディスク暗号化パスフレーズで ASCII 以外の文字を使用できるようになりました。
- インストールプログラムでは、/boot、/tmp、およびすべての /var および /usr マウントポイントに、/usr/local および /var/www 以外の新しいファイルシステムを作成する適切な推奨事項が表示されます。
- インストールプログラムはキーボードレイアウトを正しくチェックし、キーボードのレイアウト (ALT+SHIFT) を使用して異なるレイアウトと言語間を切り換える場合に、キーボードレイアウト画面を変更しないようになりました。
- 既存の RAID1 パーティションを使用するシステムでは、レスキューモードが失敗しなくなりました。
- コンテナの LUKS バージョンへの変更は、**手動パーティション設定** 画面で利用できるようになりました。
- インストールプログラムでは、**btrfs-progs** パッケージなしでインストールが正常に終了します。
- インストールプログラムでは、暗号化されたコンテナ用にデフォルトの LUKS2 バージョンが使用されるようになりました。
- キックスタートファイルが、論理ボリュームグループ (VG) の物理ボリューム (PV) を **ignoredisk** リストに配置すると、インストールプログラムがクラッシュしなくなりました。
- システムルートに新しいマウントパス **/mnt/sysroot** が追加されました。このパスは、ターゲットシステムの / をマウントするのに使用されます。通常、物理ルートとシステムの root は同じであるため、**/mnt/sysroot** は **/mnt/sysimage** と同じファイルシステムに接続されます。唯一の例外は、デプロイメントに基づいてシステムの root が変更する rpm-ostree システムのみです。次に、**/mnt/sysroot** は、**/mnt/sysimage** のサブディレクトリーに割り当てられます。chroot には **/mnt/sysroot** を使用することが推奨されます。

([BZ#1691319](#)、[BZ#1679893](#)、[BZ#1684045](#)、[BZ#1688478](#)、[BZ#1700450](#)、[BZ#1720145](#)、[BZ#1723888](#)、[BZ#1754977](#)、[BZ#1755996](#)、[BZ#1784360](#)、[BZ#1796310](#)、[BZ#1871680](#))

RHEL インストールプログラムの GUI の変更点

RHEL インストールプログラムには、インストール概要画面に以下のユーザー設定が追加されました。

- root パスワード
- ユーザーの作成

この変更により、インストールを開始する前に root パスワードを設定し、ユーザーアカウントを作成できるようになりました。以前は、インストールプロセス開始後に、root パスワードを設定してユーザーアカウントを作成していました。

root パスワードは、管理者 (スーパーユーザーまたは root としても知られている) アカウントにログインし、システム管理タスクに使用されます。コマンドラインからログインするには、ユーザー名を使用します。グラフィカル環境をインストールする場合、グラフィカルログインマネージャーは、フルネームを使用します。詳細は [標準的な RHEL インストールの実行](#) を参照してください。

(JIRA:RHELPLAN-40469)

Image Builder バックエンド **osbuild-composer** が **lorax-composer** に置き換え

osbuild-composer バックエンドは、**lorax-composer** を置き換えます。新しいサービスには、イメージビルド向けの REST API が含まれます。その結果、ユーザーはより信頼性の高いバックエンドと予測可能な出力イメージの利点を活用できます。

(BZ#1836211)

Image Builder の **osbuild-composer** が、イメージタイプのセットに対応

osbuild-composer バックエンドの置換により、**osbuild-composer** で以下のイメージタイプがサポートされます。

- TAR アーカイブ (.tar)
- QEMU QCOW2 (.qcow2)
- VMware Virtual Machine Disk (.vmdk)
- Amazon Machine Image (.ami)
- Azure Disk Image (.vhd)
- OpenStack Image (.qcow2)

以下の出力は現在サポートされていません。

- ext4-filesystem
- partitioned-disk
- Alibaba Cloud
- Google GCE

(JIRA:RHELPLAN-42617)

Image Builder が、GUI でクラウドへのプッシュに対応しました。

今回の機能拡張により、イメージの作成時に、GUI Image Builder で **Azure** および **AWS** サービスクラウドにプッシュするオプションをユーザーが選択できるようになりました。その結果、ユーザーはアップロードとインスタンス化をより簡単に活用できます。

(JIRA:RHELPLAN-30878)

5.1.2. RHEL for Edge

RHEL for Edge イメージの概要

今回のリリースで、Edge サーバー用にカスタマイズした RHEL イメージを作成できるようになりました。

Image Builder を使用して RHEL for Edge イメージを作成し、RHEL インストーラーを使用して AMD および Intel 64 ビットシステムにデプロイできます。Image Builder は、RHEL for Edge イメージを **rhel-edge-commit** として **.tar** ファイルに生成します。

RHEL for Edge イメージは、Edge サーバーで RHEL をリモートにインストールするシステムパッケージを含む **rpm-ostree** イメージです。

システムパッケージには以下が含まれます。

- ベース OS パッケージ
- コンテナエンジンとしての podman

イメージをカスタマイズして、要件に応じて OS コンテンツを設定できます。また、それらを物理マシンおよび仮想マシンにデプロイできます。

RHEL for Edge イメージを使用すると、以下を実行できます。

- アトミックアップグレード。各更新の状態が分かっており、デバイスを再起動するまで変更が加えられません。
- アップグレードに失敗した場合の回復性のために、Greenboot およびインテリジェントロールバックを使用したカスタムヘルスチェック。
- コンテナ中心のワークフロー: アプリケーションの更新からコア OS の更新を分離し、異なるバージョンのアプリケーションをテストし、デプロイすることができます。
- 低帯域幅環境向けに最適化された OTA ペイロード。
- Greenboot を使用して回復性を確保するためのカスタムヘルスチェック。

RHEL for Edge イメージの作成、インストール、および管理に関する詳細は、[RHEL for Edge イメージの作成、インストール、および管理](#) を参照してください。

(JIRA:RHELPLAN-56676)

5.1.3. ソフトウェア管理

best dnf 設定オプションのデフォルト値が True から False に変更されました。

今回の更新で、元の dnf 動作を維持するために、デフォルトの設定ファイルで、**best** な dnf 設定オプションの値が **True** に設定されました。その結果、デフォルトの設定ファイルを使用するユーザーには、動作は変更されません。

独自の設定ファイルを指定する場合は、元の動作を維持するために **best=True** オプションが存在することを確認してください。

(BZ#1832869)

dnf reposync コマンドの新しい **--norepopath** オプションが利用できるようになりました。

以前のバージョンでは、**reposync** コマンドは、デフォルトでダウンロードした各リポジトリに対して、**--download-path** ディレクトリにサブディレクトリを作成していました。今回の更新で、**--norepopath** オプションが追加され、**reposync** はサブディレクトリを作成しません。これにより、**--download-path** で指定されたディレクトリにリポジトリが直接ダウンロードされます。このオプションは、YUM v3 にも存在します。

(BZ#1842285)

libdnf プラグインを有効または無効にする機能

以前は、サブスクリプションチェックは **libdnf** プラグインの RHEL バージョンにハードコードされていました。今回の更新で、**microdnf** ユーティリティーが **libdnf** プラグインを有効または無効にすることができ、DNF と同じようにサブスクリプションチェックを無効にできるようになりました。サブスクリプションの確認を無効にするには、**--disableplugin=subscription-manager** コマンドを使用します。すべてのプラグインを無効にするには、**--noplugins** コマンドを使用します。

(BZ#1781126)

5.1.4. シェルおよびコマンドラインツール

ReaR が更新されました。

RHEL 8.3 では、**ReaR** (Relax-and-Recover) ユーティリティーに多くの更新が導入されました。以下は、主な変更点です。

- 外部バックアップソフトウェアとして、サードパーティーの Rubrik Cloud Data Management (CDM) のサポートが追加されました。これを使用するには、設定ファイルの **BACKUP** オプションを **CDM** に設定します。
- IBM POWER (リトルエンディア) アーキテクチャーに 4 GB を超えるファイルを使用したレスキューイメージの作成が有効になりました。
- **ReaR** で作成したディスクレイアウトには、Rancher 2 Longhorn iSCSI デバイスおよびファイルシステムのエントリが含まれなくなりました。

(BZ#1743303)

smartmontools がバージョン 7.1 にリベース

smartmontools パッケージがバージョン 7.1 にアップグレードされ、バグ修正および機能強化が複数追加されました。以下は、主な変更点です。

- ドライブデータベースに追加する HDD、SSD、および USB
- 新しいオプション **-j** および **--json** を使用して JSON 出力モードを有効にします。
- 一部の SAS SSD からの未完了の **Log** サブページの応答に対する回避策。
- **READ CAPACITY** コマンドの処理を向上
- ログページのデコードに関するさまざまな改善。

[\(BZ#1671154\)](#)

opencryptoki がバージョン 3.14.0 にリベースされました。

opencryptoki パッケージがバージョン 3.14.0 にアップグレードされ、バグ修正および機能強化が複数追加されました。以下は、主な変更点です。

- EP11 暗号化サービスの機能拡張:
 - Dilithium サポート
 - EdDSA (Edwards-curve digital signature algorithm) サポート
 - 非 SHA1 ハッシュおよびマスク生成関数 (MGF) を持つ RSA-OAEP (Rivest-Shamir-Adleman 最適な非対称暗号化パディング) のサポート
- 強化されたプロセスおよびスレッドロック
- **Enhancedbtree** およびオブジェクトロック
- 新しい IBM Z ハードウェア z15 のサポート
- 信頼できるプラットフォームモジュール (TPM)、IBM 暗号化アーキテクチャー (ICA)、および統合暗号化サービスファシリティ (ICSF) 向けの複数のトークンインスタンスのサポート
- **openCryptoki** トークンリポジトリのトークンキーをリスト表示する新しいツール **p11sak** を追加
- トークンリポジトリを FIPS 準拠の暗号化に移行するユーティリティを追加
- **pkcsep11_migrate** ツールを修正
- ICSF ソフトウェアの若干の修正

[\(BZ#1780293\)](#)

gpgme がバージョン 1.13.1 にリベース

gpgme パッケージがアップストリームバージョン 1.13.1 にアップグレードされました。以下は、主な変更点です。

- 新しいコンテキストフラグ **no-symkey-cache** (GnuPG 2.2.7 以降と併用すると効果がある)、**request-origin** (GnuPG 2.2.6 以降と併用すると効果がある)、**auto-key-locate** および **trust-model** が導入されました。
- Web ブラウザーのネイティブメッセージングサーバーとしての新しいツール **gpgme-json** が追加されました。現時点では、公開鍵の暗号化および復号がサポートされています。
- 非表示の受信者オプションを含むダイレクトキー仕様をサポートする新しい暗号化 API が導入されました。これにより、サブキーも使用できます。

[\(BZ#1829822\)](#)

5.1.5. インフラストラクチャーサービス

powertop がバージョン 2.12 にリベースされました。

powertop パッケージがバージョン 2.12 にアップグレードされました。以前に利用可能なバージョン 2.11 に対する主な変更点は、以下のとおりです。

- SATA リンク PM に DIPM (Device Interface Power Management) の使用。
- Intel Comet Lake モバイルシステムおよびデスクトップシステム、Skylake サーバー、および Atom ベースの Tremont アーキテクチャー (Jasper Lake) のサポート。

(BZ#1783110)

tuned がバージョン 2.14.0 にリベースされました。

tuned パッケージがアップストリームバージョン 2.14.0 にアップグレードされました。主な機能強化は、次のとおりです。

- **optimize-serial-console** プロファイルが導入されました。
- 読み込まれた post プロファイルのサポートが追加されました。
- **irqbalance** 設定を処理する **irqbalance** プラグインが追加されました。
- Marvell ThunderX および AMD ベースのプラットフォーム用のアーキテクチャー固有のチューニングが追加されました。
- スケジューラープラグインは、CPU アフィニティー設定の **cgroups-v1** をサポートするように拡張されました。

(BZ#1792264)

tcpdump がバージョン 4.9.3 にリベース

tcpdump ユーティリティーがバージョン 4.9.3 に更新され、CVE (Common Vulnerabilities and Exposures) を修正できるようになりました。

(BZ#1804063)

libpcap がバージョン 1.9.1 にリベース

libpcap パッケージがバージョン 1.9.1 に更新され、CVE (Common Vulnerabilities and Exposures) を修正できるようになりました。

(BZ#1806422)

iperf3 がクライアント側で **sctp** オプションに対応

今回の機能拡張により、ユーザーはネットワークスループットをテストするクライアント側の TCP (Transmission Control Protocol) の代わりに SCTP (Stream Control Transmission Protocol) を使用できるようになりました。

iperf3 の以下のオプションが、テストのクライアント側で利用できるようになりました。

- **--sctp**
- **--xbind**
- **--nstreams**

詳細は、man ページの **iperf3** の **Client Specific Options** を参照してください。

(BZ#1665142)

iperf3 が SSL に対応

今回の機能強化により、ユーザーはクライアントとサーバー間の RSA 認証を使用して、サーバーへの接続を正当なクライアントのみに制限できるようになりました。

iperf3 の以下のオプションがサーバー側で利用できるようになりました。

- **--rsa-private-key-path**
- **--authorized-users-path**

iperf3 の以下のオプションが、通信のクライアント側で利用できるようになりました。

- **--username**
- **--rsa-public-key-path**

(BZ#1700497)

bind が 9.11.20 にリベース

pcp パッケージがバージョン 9.11.20 にアップグレードされ、バグ修正および機能強化が複数追加されました。以下は、主な変更点です。

- 競合状態を修正することで、CPU コアが多数あるシステムの信頼性が向上します。
- 詳細なエラーレポート: **dig** およびその他のツールが存在する場合は、拡張 DNS エラー (EDE) オプションを出力できるようになりました。
- インバウンド DNS Zone Transfer Protocol (AXFR) 転送のメッセージ ID は、一致しない場合にチェックされ、ログに記録されます。

(BZ#1818785)

printk の値を減らして I/O をシリアルコンソールに減らす新しい optimize-serial-console TuneD プロファイル

今回の更新で、新しい **optimize-serial-console** TuneD プロファイルが利用できるようになりました。シナリオによっては、カーネルドライバーが大量の I/O 操作をシリアルコンソールに送信することができます。I/O がシリアルコンソールに書き込まれる間、そのような動作は一時的に応答しなくなることがあります。**optimize-serial-console** プロファイルでは、デフォルトの 7417 から 4417 への **printk** 値を減らすことで、この I/O を減らします。システムでこの変更を行う場合は、シリアルコンソールを使用すると、以下のようにシステムをインストルメント化できます。

```
# tuned-adm profile throughput-performance optimize-serial-console
```

その結果、ユーザーは再起動後も維持する **printk** の値が少なくなるため、システムがハングする可能性が低くなります。

この TuneD プロファイルは、デバッグ情報を削除して、シリアルコンソールに書き込まれた I/O の量を減らします。このデバッグ情報を収集する必要がある場合は、このプロファイルが有効になっておらず、**printk** の値が 7417 に設定されていることを確認する必要があります。**printk** 実行の値を確認するには、次のコマンドを実行します。

```
# cat /proc/sys/kernel/printk
```

(BZ#1840689)

AMD ベースのプラットフォームに新しい TuneD プロファイルが追加

RHEL 8.3 では、AMD ベースのプラットフォームのチューニングを組み込むように **throughput-performance** TuneD プロファイルが更新されました。パラメーターを手動で変更する必要はありません。チューニングは **AMD** システムに自動的に適用されます。**AMD Epyc Naples** および **Rome** システムは、デフォルトの **throughput-performance** プロファイルで以下のパラメーターを変更します。

sched_migration_cost_ns=5000000 および **kernel.numa_balancing=0**

この機能拡張により、システムパフォーマンスが最大 5% 向上します。

(BZ#1746957)

Memcached がバージョン 1.5.22 にリベース

memcached パッケージがバージョン 1.5.22 にアップグレードされました。以前のバージョンに対する主な変更点は、以下のとおりです。

- TLS が有効になっています。
- **-o inline_ascii_response** オプションは削除されました。
- **-Y [authfile]** オプションが ASCII プロトコルの認証モードとともに追加されました。
- **memcached** は、再起動間のキャッシュを回復できるようになりました。
- 実験的なメタコマンドが新たに追加されました。
- さまざまなパフォーマンス向上。

(BZ#1809536)

5.1.6. セキュリティー

Cyrus SASL は、**SASL/GSSAPI** プラグインおよび **SASL/GSS-SPNEGO** プラグインを使用したチャンネルバインディングをサポートするようになりました。

今回の更新で、**SASL/GSSAPI** プラグインおよび **SASL/GSS-SPNEGO** プラグインを使用したチャンネルバインディングのサポートが追加されました。その結果、**openldap** ライブラリーで使用すると、この機能は、**Cyrus SASL** が LDAP 接続に強制チャンネルバインディングを持つ Microsoft Active Directory および Microsoft Windows システムとの互換性を維持およびアクセスできるようになります。

(BZ#1817054)

Libreswan を 3.32 にリベース

今回の更新で、Libreswan がアップストリームバージョン 3.32 にリベースされ、新機能およびバグ修正が複数追加されました。主な変更には以下のものがあります。

- Libreswan では、別の FIPS 140-2 認定が不要になりました。
- Libreswan は、RFC 8247 の暗号化推奨事項を実装し、SHA-1 および RSA-PKCS v1.5 から SHA-2 および RSA-PSS に、優先順位を変更するようになりました。

- Libreswan は、ファイアウォールルールの書き込みを簡素化する XFRMi 仮想 ipsecXX インターフェイスをサポートします。
- 完全なメッシュ暗号化ネットワークでクラッシュおよび再起動するノードの復旧が改善されました。

(BZ#1820206)

libssh ライブラリーがバージョン 0.9.4 にリベース

SSH プロトコルを実装する **libssh** ライブラリーがバージョン 0.9.4 にアップグレードされました。

この更新には、以下を含むバグ修正および機能拡張が含まれています。

- PEM ファイルに **Ed25519** キーのサポートが追加されました。
- **diffie-hellman-group14-sha256** キー交換アルゴリズムのサポートが追加されました。
- **libssh** クライアント設定ファイルの **Match** キーワードに **localuser** のサポートが追加されました。
- **一致** 条件のキーワード引数は、大文字と小文字が区別されるようになりました (キーワードは大文字と小文字を区別しませんが、キーワードの引数は大文字と小文字を区別しません)。
- CVE-2019-14889 および CVE-2020-1730 を修正しました。
- 既知のホストファイルに指定されているパス文字列に不足しているディレクトリーを再帰的に作成するサポートが追加されました。
- コメントと空白を含む PEM ファイルに **OpenSSH** キーのサポートが追加されました。
- **libssh** サーバー設定に含まれる **OpenSSH** サーバー設定を削除しました。

(BZ#1804797)

gnutls が 3.6.14 にリベース

gnutls パッケージがアップストリームバージョン 3.6.14 にリベースされました。このバージョンでは、主なバグ修正および機能強化が数多く追加されました。

- **gnutls** は、無効な文字またはフォーマットを含む **Time** フィールドのある証明書を拒否するようになりました。
- **gnutls** は、信頼できる CA 証明書で最小キーサイズをチェックするようになりました。
- 暗号化された秘密鍵を表示すると、**certtool** ユーティリティーにプレーンテキストの記述が含まれなくなりました。
- **gnutls** を使用するサーバーが OCSP ステージングサポートを公開するようになりました。
- **gnutls** を使用するクライアントは、リクエストでのみ OCSP ステージングを送信するようになりました。

(BZ#1789392)

gnutls FIPS DH チェックが NIST SP 800-56A リビジョンに準拠3

gnutls パッケージの今回の更新で、NIST Special Publication 800-56A リビジョン 3、セクション 5.7.1.1 および 5.7.1.2 ステップ 2 で必要なチェックが提供されます。この変更は、今後の FIPS 140-2 認

定に必要です。その結果、**gnutls** は、FIPS モードで動作する際に、Diffie-Hellman 鍵交換中に RFC 7919 および RFC 3526 から 2048 ビット以上のパラメーターのみを受け入れるようになりました。

(BZ#1849079)

gnutls が NIST SP 800-56A リビジョン 3 に従って検証を実行

gnutls パッケージの今回の更新で、NIST Special Publication 800-56A リビジョン 3、セクション 5.6.2.2.2 および 5.6.2.1.3、ステップ 2 で必要なチェックが追加されました。さらに、今後の FIPS 140-2 認定に **gnutls** を準備します。その結果、**gnutls** は、FIPS モードで動作する際に、Diffie-Hellman 鍵交換中に公開鍵の生成および受け取りについて追加の検証ステップを実行します。

(BZ#1855803)

update-crypto-policies および fips-mode-setup が crypto-policies-scripts に移動

crypto-policies パッケージに含まれていた **update-crypto-policies** スクリプトおよび **fips-mode-setup** スクリプトは、別の RPM サブパッケージ **crypto-policies-scripts** に移動しました。パッケージは、通常のインストールの Recommends 依存関係を使用して自動的にインストールされます。これにより、**ubi8/ubi-minimal** イメージが有効になり、Python 言語インタプリターが含まれなくなり、イメージサイズが削減されます。

(BZ#1832743)

opensc がバージョン 0.20.0 にリベース

opensc パッケージがバージョン 0.20.0 にリベースされ、複数のバグやセキュリティー上の問題に対応します。以下は、主な変更点です。

- 今回の更新で、CVE-2019-6502、CVE-2019-15946、CVE-2019-15945、CVE-2019-19480、CVE-2019-19481、および CVE-2019-19479 のセキュリティー問題が修正されています。
- OpenSC モジュールは、**C_WrapKey** 関数および **C_UnwrapKey** 関数に対応するようになりました。
- これで、この機能を使用して、カードリーダーの挿入および取り外しを期待どおりに検出できるようになりました。
- **pkcs11-tool** ユーティリティーが、**CKA_ALLOWED_MECHANISMS** 属性に対応するようになりました。
- 今回の更新で、**OsEID** カードのデフォルトの検出が可能になりました。
- OpenPGP Card v3 は、**Elliptic Curve Cryptography (ECC)** に対応するようになりました。
- PKCS#11 URI が、省略記号を使用してリーダー名を切り捨てるようになりました。

(BZ#1810660)

stunnel がバージョン 5.56 にリベース

今回の更新で、**stunnel** 暗号化ラッパーがアップストリームバージョン 5.56 にリベースされ、新機能やバグ修正が複数追加されました。主な変更には以下のものがあります。

- 発行したセッションチケットの機密性および整合性保護を制御する新しい **ticketKeySecret** オプションおよび **ticketMacSecret** オプション。これらのオプションにより、クラスターの他のノードのセッションを再開できます。

- OpenSSL 1.1.0 以降の elliptic 曲線のリストを制御する新しい **curves** オプション。
- 許可される TLS 1.3 暗号スイートのリストを制御する新しい **ciphersuites** スイートオプション。
- OpenSSL 1.1.0 以降に **sslVersion**、**sslVersionMin**、および **sslVersionMax** を追加。

(BZ#1808365)

libkcapi がバージョン 1.2.0 にリベース

libkcapi パッケージがアップストリームバージョン 1.2.0 にリベースされ、これには若干の変更が含まれています。

(BZ#1683123)

setools が 4.3.0 にリベース

SELinux ポリシー解析を容易にするために設計されたツールの集合である **setools** パッケージが、バージョン 4.3.0 にアップグレードされました。

この更新には、以下を含むバグ修正および機能拡張が含まれています。

- Type Enforcement (TE) ルールの **sediff** メソッドが改訂され、メモリーおよびランタイムの問題が大幅に削減されました。
- **seinfo**、**sediff**、および **apol** に、**infiniband** コンテキストサポートが追加されました。
- オンラインドキュメントを表示するために使用される Qt アシスタントツールの場所の **apol** 設定が追加されました。
- 以下の **sediff** の問題が修正されました。
 - 要求されない場合に表示するプロパティヘッダー。
 - **type_transition** ファイルの名前比較。
- マップソケットの **sendto** 情報フローの方向のパーミッションが修正されました。
- 完全な Python コレクションを作成するために **TypeAttribute** クラスにメソッドを追加しました。
- **Genfscon** は、**libsepol** から削除された固定値を使用するのではなく、クラスを検索します。

setools パッケージには、以下のパッケージが必要です。

- **setools-console**
- **setools-console-analyses**
- **setools-gui**

(BZ#1820079)

個々の CephFS ファイルおよびディレクトリーに SELinux ラベルを追加

CephFS (Ceph File System) は、ファイルの拡張属性に SELinux ラベルを保存するように最近有効化されています。以前のリリースでは、CephFS ボリュームのすべてのファイルに、共通のラベル **system_u:object_r:cephfs_t:s0** のラベルが付けられました。今回の機能拡張により、個別のファイル

のラベルを変更し、SELinux が移行ルールに基づいて新たに作成されたファイルのラベルを定義することができるようになりました。以前にラベル付けされていないファイルには、明示的に変更されるまで **system_u:object_r:cephfs_t:s0** ラベルが設定されていることに注意してください。

([BZ#1823764](#))

OpenSCAP がバージョン 1.3.3 にリベースされました。

openscap パッケージがアップストリームバージョン 1.3.3 にアップグレードされ、以前のバージョンに対するバグ修正および機能拡張が数多く追加されました。主な変更点:

- コマンドラインインターフェイス (CLI) を使用してテーリングファイルを生成できるようにする **autotailor** スクリプトが追加されました。
- Extensible Configuration Checklist Description Format (XCCDF) TestResult の起動および終了のタイムスタンプにタイムゾーン部分が追加されました。
- **yamfilecontent** 独立したプローブがドラフト実装として追加されました。
- XCCDF に **urn:xccdf:fix:script:kubernetes** 修正タイプが導入されました。
- **machineconfig** 修正を生成する機能を追加します。
- **oscap-podman** ツールがあいまいなスキャンターゲットを検出できるようになりました。
- **rpmverifyfile** プローブが、**/bin** ディレクトリーからファイルを検証できるようになりました。
- **textfilecontent58** プローブで複雑な正規表現が実行する時のクラッシュが修正されました。
- XCCDF レポートの評価特性が、**system_info** プローブからの OVAL エンティティーと一致するようになりました。
- **textfilecontent58** プローブのオフラインモードでのファイルパスパターンの一致が修正されました。
- **systemdunitdependency** プローブの無限再帰が修正されました。

([BZ#1829761](#))

SCAP セキュリティーガイドで、CIS RHEL 8 Benchmark v1.0.0 に一致するプロファイルを提供

今回の更新により、**scap-security-guide** パッケージで CIS Red Hat Enterprise Linux 8 Benchmark v1.0.0 に一致するプロファイルが提供されるようになりました。このプロファイルを使用すると、Center for Internet Security (CIS) のガイドラインに従ってシステムの設定を強化できます。このため、CIS Ansible Playbook および CIS SCAP プロファイルを使用して、RHEL 8 システムの CIS への準拠を設定および自動化できます。

CIS プロファイルの **rpm_verify_permissions** ルールは正常に機能しません。

([BZ#1760734](#))

scap-security-guide が HIPAA を実装するプロファイルを提供

scap-security-guide パッケージの今回の更新で、Health Insurance Portability and Accountability Act (HIPAA) プロファイルが RHEL 8 セキュリティーコンプライアンスコンテンツに追加されました。このプロファイルは、[The HIPAA Privacy Rule](#) の Web サイトで説明されている推奨事項を実装します。

HIPAA Security Rule は、対象となるエンティティーによって作成、受信、使用、または維持される個人の電子個人健康情報を保護するための米国の国内基準を確立します。セキュリティールールには、適切な管理、物理、および技術の保護手段が必要です。これにより、電子的に保護された健全性情報の機密性、整合性、およびセキュリティを確保できます。

(BZ#1832760)

scap-security-guide が 0.150 にリベース

scap-security-guide パッケージ (Linux システムのセキュリティポリシーの最新セットを含む) が、バージョン 0.150 にアップグレードされました。

この更新には、バグ修正および機能強化が含まれています。主な変更点は、以下のとおりです。

- Ansible のコンテンツが改善されました。多数のルールには、Ansible の初回修正が含まれていますが、バグ修正についてはその他のルールが更新されています。
- RHEL7 システムのスキャン用に **scap-security-guide** コンテンツを修正し、改善します。これには、以下が含まれます。
 - **scap-security-guide** パッケージで、CIS RHEL 7 Benchmark v2.2.0 に一致するプロファイルを提供するようになりました。CIS プロファイルの **rpm_verify_permissions** ルールが正常に動作しないため、既知の問題である **CIS プロファイルの rpm_verify_permissions が失敗する** を参照してください。
 - SCAP セキュリティガイドプロファイルは、開始すべきでないサービスを適切に無効およびマスクするようになりました。
 - **scap-security-guide** パッケージの **audit_rules_privileged_commands** ルールが特権コマンドに対して適切に機能するようになりました。
 - **scap-security-guide** パッケージの **dconf_gnome_login_banner_text** ルールの修正が正しく失敗しなくなりました。

(BZ#1815007)

SCAP Workbench が、カスタムプロファイルから結果ベースの修正を生成可能に

今回の更新で、**SCAP Workbench** ツールを使用して、カスタマイズしたプロファイルから結果ベースの修復ルールを生成できるようになりました。

(BZ#1640715)

新しい Ansible ロール は、Clevis クライアントの自動デプロイメントを提供します。

rhel-system-roles パッケージの今回の更新で、**nbde_client** RHEL システムロールが導入されました。この Ansible ロールを使用すると、複数の Clevis クライアントを自動的にデプロイできます。

(BZ#1716040)

新しい Ansible ロールで、Tang サーバーを設定可能に

今回の機能拡張により、新しい **nbde_server** システムロールを使用して、自動ディスク暗号化ソリューションの一部として、Tang サーバーをデプロイおよび管理できるようになりました。**rhel-system-roles** パッケージに含まれる **nbde_server** Ansible ロールでは、以下の機能がサポートされません。

- Tang 鍵のローテーション

- Tang 鍵のデプロイおよびバックアップ

詳細は [Tang サーバー鍵のローテーション](#) を参照してください。

(BZ#1716039)

clevis がバージョン 13 にリベース

clevis パッケージがバージョン 13 にリベースされ、バグ修正および機能強化が複数追加されました。以下は、主な変更点です。

- **clevis luks unlock** は、非対話モードでキーファイルを使用してデバイスで使用できます。
- **clevis encrypt tpm2** は、入力が JSON 配列として指定される場合、**pcr_ids** フィールドを解析します。
- man ページの **clevis-luks-unbind(1)** は、LUKS v1 のみを参照しなくなりました。
- **clevis luks bind** は、指定したパスワードが正しくないと、非アクティブなスロットには書き込まれません。
- **clevis luks bind** が、システムが英語以外のロケールを使用する間に機能するようになりました。
- **tpm2-tools** 4.x のサポートを追加

(BZ#1818780)

clevis luks edit を使用して特定のピン設定を編集可能に

clevis パッケージの今回の更新で、特定のピン設定を編集できるように、新しい **clevis luks edit** サブコマンドが追加されました。たとえば、Tang サーバーの URL アドレスと、TPM2 設定で **pcr_ids** パラメーターを変更できるようになりました。新しい **sss** ピンを追加および削除し、**sss** ピンのしきい値を変更することもできます。

(BZ#1436735)

clevis luks bind -y で自動バインディングが可能に

今回の機能強化により、Clevi は **-y** パラメーターを使用した自動バインディングをサポートするようになりました。**clevis luks bind** コマンドで **-y** オプションを使用して、その後のプロンプトに **yes** と自動的に回答できるようになりました。たとえば、Tang ピンを使用する場合は、Tang 鍵を手動で信頼する必要がなくなりました。

(BZ#1819767)

fapolicyd がバージョン 1.0 にリベース

fapolicyd パッケージがバージョン 1.0 にリベースされ、バグ修正および機能強化が複数追加されました。以下は、主な変更点です。

- 複数のスレッド同期の問題が解決されました。
- データベースのサイズと読み込み時間を短縮し、パフォーマンスが向上しました。
- 信頼バックエンドをカスタマイズするために、**fapolicyd.conf** ファイルの **fapolicyd** パッケージの新しい信頼オプションが追加されました。信頼できるファイル、バイナリー、およびスクリプトはすべて、新しい **/etc/fapolicyd/fapolicyd.trust** ファイルに追加できます。

- CLI を使用して **fapolicyd.trust** ファイルを管理できます。
- CLI を使用してデータベースをクリーンアップまたはダンプできます。
- **fapolicyd** パッケージは、スクリプトのデコードをより適切に行うためにマジックデータベースを上書きします。CLI は、上書きに応じて、file コマンドに類似したファイルの MIME タイプを出力します。
- **/etc/fapolicyd/fapolicyd.rules** ファイルは、値のグループを属性値としてサポートします。
- **fapolicyd** デーモンには、**audit/sylog** イベントの形式を設定する **syslog_format** オプションがあります。

(BZ#1817413)

fapolicyd が、fapolicyd-selinux に独自の SELinux ポリシーを提供

今回の機能拡張により、**fapolicyd** フレームワークが独自の SELinux セキュリティポリシーを提供するようになりました。デーモンは **fapolicyd_t** ドメインで制限され、ポリシーは **fapolicyd-selinux** サブパッケージを使用してインストールされます。

(BZ#1714529)

usbguard がバージョン 0.7.8 にリベース

usbguard パッケージがバージョン 0.7.8 にリベースされ、バグ修正および機能強化が複数追加されました。以下は、主な変更点です。

- **/etc/usbguard/usbguard-daemon.conf** ファイルの **HidePII=true|false** パラメーターは、監査エントリから個人的識別可能な情報を非表示にできるようになりました。
- **/etc/usbguard/usbguard-daemon.conf** ファイルの **AuthorizedDefault=keep|none|all|internal** パラメーターにより、コントローラーデバイスの承認状態が事前定義される可能性があります。
- 新しい **with-connect-type** ルール属性を使用すると、デバイスの接続タイプを区別できるようになりました。
- **-t** オプションを使用して、一時ルールを追加できるようになりました。一時的なルールは、デーモンが再起動するまでメモリー内に留まります。
- **usbguard list-rules** が、特定のプロパティに従ってフィルターできるようになりました。
- **usbguard generate-policy** が、特定のデバイスのポリシーを生成できるようになりました。
- **usbguard allow|block|reject** コマンドはルール文字列を処理できるようになり、指定されたルール文字列に一致する各デバイスにターゲットが適用されます。
- 新しいサブパッケージ **usbguard-notifier** および **usbguard-selinux** が含まれます。

(BZ#1738590)

USBGuard は、企業デスクトップユーザーの改善を多数提供

USBGuard プロジェクトへの追加には、企業のデスクトップユーザーのユーザービリティを向上させる機能拡張およびバグ修正が含まれます。重要な変更には以下が含まれます。

- `/etc/usbguard/rules.conf` ルールファイルを正常に維持するために、ユーザーは `RuleFolder=/etc/usbguard/rules.d/` ディレクトリーに複数の設定ファイルを定義できます。デフォルトでは、`RuleFolder` は `/etc/usbguard-daemon.conf` ファイルに指定されます。
- `usbguard-notifier` ツールは、GUI 通知を提供するようになりました。このツールは、デバイスがプラグインまたはプラグアウトされたとき、およびデバイスがユーザーによって許可、ブロック、または拒否されたかどうかをユーザーに通知します。
- `usbguard-daemon` は `#` で始まる行を解析しなくなったため、設定ファイルにコメントを追加できるようになりました。

(BZ#1667395)

USBGuard が `usbguard-selinux` に独自の SELinux ポリシーを提供するようになりました。

今回の機能拡張により、USBGuard フレームワークが独自の SELinux セキュリティポリシーを提供するようになりました。デーモンは `usbguard_t` ドメインで制限され、ポリシーは `usbguard-selinux` サブパッケージからインストールされます。

(BZ#1683567)

`libcap` がアンビエント機能に対応

この更新で、ユーザーはログイン時にアンビエント機能を付与できるようになり、適切に設定されたプロセスに root アクセス権を持たせる必要をなくすることができます。

(BZ#1487388)

`libseccomp` ライブラリーがバージョン 2.4.3 にリベース

`seccomp` システムコールのフィルタリングメカニズムへのインターフェイスを提供する `libseccomp` ライブラリーが、バージョン 2.4.3 にアップグレードされました。

この更新では、バグ修正および機能強化が数多く追加されました。以下は、主な変更点です。

- Linux v5.4-rc4 の `syscall` テーブルを更新しました。
- 存在しないシステムコールの `__NR_x` 値を定義しなくなりました。
- `__SNR_x` が内部で使用されるようになりました。
- `__SNR_ppoll` の `define` が追加されました。
- `s390/s390x shm*` システムコールの多重化の問題が修正されました。
- `libseccomp` ツールのコンパイルから `static` フラグが削除されました。
- 関連するシステムコールへの `io-uring` サポートが追加されました。
- v2.4.0 リリースに導入された Python モジュールの命名の問題を修正しました。このモジュールは、以前と同じ `seccomp` という名前になります。
- `scmp_bpf_sim` ツールの `clang` によって識別されるメモリーリークの可能性が修正されました。

(BZ#1770693)

`omamqp1` モジュールのサポート

今回の更新で、**AMQP 1.0** プロトコルがバスの宛先へのメッセージの送信をサポートするようになりました。以前のリリースでは、OpenStack は **AMQP1** プロトコルを通信標準として使用していました。このプロトコルは AMQP メッセージでメッセージをログに記録できるようになりました。今回の更新で、**rsyslog-omamqp1** サブパッケージが導入され、**omamqp1** 出力モードが提供され、これはメッセージをログに記録し、バスの宛先に送信します。

(BZ#1713427)

OpenSCAP はリモートコンテンツを圧縮

今回の更新で、OpenSCAP がリモートコンテンツを転送するのに **gzip** 圧縮を使用します。リモートコンテンツの最も一般的なタイプはテキストベースの CVE フィードです。これは、時間の経過とともにサイズが増え、通常はすべてのスキャンに対してダウンロードする必要があります。**gzip** 圧縮は、圧縮されていないコンテンツに必要な帯域幅を 10% に減らします。その結果、スキャンされたシステムと、リモートコンテンツをホストするサーバーとの間のチェーン全体で帯域幅要件が減少します。

(BZ#1855708)

SCAP セキュリティガイドで NIST-800-171 に一致するプロファイルが提供されるようになりました。

今回の更新で、**scap-security-guide** パッケージが、NIST-800-171 標準に一致するプロファイルを提供するようになりました。このプロファイルを使用すると、非対応情報システムの CUI (Controlled Unclassified Information) のセキュリティ要件に従ってシステム設定を強化できます。これにより、NIST-800-171 標準に合わせてシステムを簡単に設定することができます。

(BZ#1762962)

5.1.7. ネットワーク

IPv4 接続追跡モジュールおよび IPv6 接続追跡モジュールは、`nf_conntrack` モジュールにマージされています。

今回の機能拡張で、Netfilter 接続追跡モジュール `nf_conntrack_ipv4` と `nf_conntrack_ipv6` が `nf_conntrack` カーネルモジュールにマージされます。この変更により、アドレスファミリー固有のモジュールを RHEL 8.3 で機能しないので、`nf_conntrack` モジュールのみをブラックリストに登録して、IPv4 プロトコルと IPv6 プロトコルの両方の接続追跡サポートを無効にできます。

(BZ#1822085)

firewalld がバージョン 0.8.2 にリベース

firewalld パッケージがアップストリームバージョン 0.8.2 にアップグレードし、以前のバージョンにバグ修正が数多く追加されました。詳細は、[firewalld 0.8.2 Release Notes](#) を参照してください。

(BZ#1809636)

NetworkManager がバージョン 1.26.0 にリベース

NetworkManager パッケージがアップストリームバージョン 1.26.0 にアップグレードされ、以前のバージョンに対するバグ修正や機能強化が数多く追加されました。

- NetworkManager は、デバイスの非アクティブ化時に、オートネゴシエーション、速度、およびデュプレックスの設定を元の値にリセットします。
- 以前のアクティベーション試行がすべて失敗した場合、Wi-Fi プロファイルが自動的に接続されるようになりました。これは、ネットワークへの自動接続の最初の失敗が自動化をブロックしなくなったことを意味します。副次的効果として、以前にブロックされていた既存の Wi-Fi プ

ロファイルが自動的に接続されました。

- **nm-settings-nmcli(5)** および **nm-settings-dbus(5)** の man ページが追加されました。
- 多数のブリッジパラメーターのサポートが追加されました。
- VRF (Virtual Routing and Forwarding) インターフェイスのサポートが追加されました。詳細は、[異なるインターフェイスで同じ IP アドレスを連続して再利用](#) を参照してください。
- Wi-Fi ネットワークの Opportunistic Wireless Encryption (OWE) モードに対応するようになりました。
- NetworkManager は、[RFC 3021](#) に応じて、IPv4 ポイントツーポイントリンクの 31 ビットの接頭辞に対応するようになりました。
- **nmcli** ユーティリティーは、**nmcli connection modify <connection_name> remove <setting>** コマンドを使用した設定の削除をサポートするようになりました。
- マスターデバイスがない場合は、NetworkManager がスレーブデバイスを作成し、アクティベートしなくなりました。

主な変更の詳細は、アップストリームのリリースノートを参照してください。

- [NetworkManager 1.26.0](#)
- [NetworkManager 1.24.0](#)

([BZ#1814746](#))

XDP に条件付きサポートを追加

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

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

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

([BZ#1889736](#))

xdp-tools が部分的にサポートされます。

XDP (kernel eXpress Data Path) 機能のユーザー空間サポートユーティリティーを含む **xdp-tools** パッケージが、AMD および Intel 64 ビットアーキテクチャーでサポートされるようになりました。これには、**libxdp** ライブラリー、XDP プログラムを読み込む **xdp-loader** ユーティリティー、パケットフィルタリングの **xdp-filter** サンプルプログラムが含まれます。XDP が有効になっているネットワークインターフェイスからパケットをキャプチャーするための **xdpdump** ユーティリティーは、引き続きテクノロジープレビューである点に注意してください。(BZ#1820670)

デフォルトでは、**dracut** ユーティリティーが初期 RAM ディスクで **NetworkManager** を使用するようになりました。

以前は、**dracut** ユーティリティーはシェルスクリプトを使用して初期 RAM ディスク (**initrd**) のネットワークを管理していました。場合によっては、問題が発生することがあります。たとえば、RAM ディスクのスクリプトが IP アドレスをすでに要求している場合でも、**NetworkManager** が別の DHCP 要求を送信すると、タイムアウトが発生する可能性があります。

今回の更新で、デフォルトで **dracut** が初期 RAM ディスクに **NetworkManager** を使用し、システムで問題が発生しなくなりました。以前の実装に切り替え、RAM ディスクイメージを再作成する場合には、以下のコマンドを使用します。

```
# echo 'add_dracutmodules+= "network-legacy "' > /etc/dracut.conf.d/enable-network-legacy.conf

# dracut -vf --regenerate-all
```

(BZ#1626348)

カーネルコマンドラインのネットワーク設定が、**ip** パラメーターに統合されました。

カーネルコマンドラインでネットワーク設定を設定する **ipv6**、**netmask**、**gateway**、および **hostname** パラメーターは、**ip** パラメーターに統合されました。**ip** パラメーターでは、以下のような異なる形式を使用できます。

```
ip=__IP_address__:__peer__:__gateway_IP_address__:__net_mask__:__host_name__:__interface_name__:__configuration_method__
```

個々のフィールドや、このパラメーターで使用できるその他の形式の詳細は、**dracut.cmdline(7)** の man ページの **ip** パラメーターの説明を参照してください。

RHEL 8 では、**ipv6** パラメーター、**netmask** パラメーター、**gateway** パラメーター、および **hostname** パラメーターは利用できなくなりました。

(BZ#1905138)

5.1.8. カーネル

RHEL 8.3 のカーネルバージョン

Red Hat Enterprise Linux 8.3 には、カーネルバージョン 4.18.0-240 が同梱されています。

(BZ#1839151)

Extended Berkeley Packet Filter for RHEL 8.3

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

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

Red Hat は、**eBPF** 仮想マシンを使用するコンポーネントを数多く提供しています。各コンポーネントの開発フェーズはさまざまです。そのため、現在すべてのコンポーネントが完全にサポートされている訳ではありません。RHEL 8.3 では、以下の **eBPF** コンポーネントがサポートされています。

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

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

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

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

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

([BZ#1780124](#))

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

Red Hat Enterprise Linux 8.3 は、Omni-Path Architecture (OPA) ホストソフトウェアに完全に対応しています。OPA は、クラスター環境のコンピュートと I/O ノード間の高性能データ転送 (高帯域幅、高メッセージレート、低レイテンシー) のために、初期化とセットアップを行う Host Fabric Interface (HFI) ハードウェアを提供します。

([BZ#1893174](#))

TSX がデフォルトで無効になる

RHEL 8.3 以降、カーネルはデフォルトで **Intel® Transactional Synchronization Extensions (TSX)** 技術が無効にし、OS セキュリティーを強化するようになりました。この変更は、**TSX** の無効化に対応する CPU に適用されます。これには、第 2 世代の Intel® Xeon® Scalable Processor (以前は Intel® C620 Series Chipsets で Cascade Lake として知られる) が含まれます。

アプリケーションが **TSX** を使用しないユーザーの場合は、第 2 世代 Intel® Xeon® Scalable Processor で **TSX Asynchronous Abort (TAA)** の軽減策であるデフォルトのパフォーマンスペナルティーが削除されます。

この変更は、RHEL カーネルの動作をアップストリームに合わせて調整します。この場合、**TSX** は Linux 5.4 以降デフォルトでは無効になっています。

TSX を有効にするには、**tsx=on** パラメーターをカーネルコマンドラインに追加します。

(BZ#1828642)

RHEL 8.3 がページ所有者の追跡機能に対応

今回の更新により、ページの所有者追跡機能を使用して、ページ割り当てレベルでカーネルメモリー使用率を確認できるようになりました。

ページトラッカーを有効にするには、以下の手順を実行します。

```
# grubby --args="page_owner=on" --update-kernel=0
# reboot
```

その結果、ページの所有者トラッカーはカーネルメモリー消費を追跡します。これは、カーネルメモリーリークをデバッグし、大量のメモリーを使用するドライバーを検出するのに役立ちます。

(BZ#1825414)

AMD EPYC™ 7003 シリーズプロセッサの EDAC に対応

今回の機能拡張により、AMD EPYC™ 7003 シリーズプロセッサ用の Error Detection And Correction (EDAC) デバイスのサポートが提供されます。以前は、AMD EPYC™ 7003 シリーズプロセッサをベースとするシステムで、修正 (CE) および未修正 (UE) メモリーエラーが報告されていませんでした。今回の更新で、このエラーが EDAC を使用して報告されるようになりました。

(BZ#1735611)

flamegraph が perf ツールに対応

今回の更新で、**perf** コマンドラインツールが、システムのパフォーマンスのグラフィカル表現を作成するためのフレームグラフをサポートするようになりました。**perf** データは、同様のスタックのバックトレースが設定されたサンプルにグループ化されます。その結果、このデータは視覚的な表現に変換され、計算量の多いコード領域を簡単に識別できるようになります。**perf** ツールを使用してフレームグラフを生成するには、以下のコマンドを実行します。

```
$ perf script record flamegraph -F 99 -g -- stress --cpu 1 --vm-bytes 128M --timeout 10s
stress: info: [4461] dispatching hogs: 1 cpu, 0 io, 0 vm, 0 hdd
stress: info: [4461] successful run completed in 10s
[ perf record: Woken up 1 times to write data ]
[ perf record: Captured and wrote 0.060 MB perf.data (970 samples) ]
$ perf script report flamegraph
dumping data to flamegraph.html
```

注記: flamegraphs を生成するには、**js-d3-flame-graph** rpm をインストールします。

(BZ#1281843)

/dev/random および /dev/urandom が、Kernel Crypto API DRBG が条件付きで電源が入るようになる

FIPS モードでは、疑似乱数ジェネレーター **/dev/random** および **/dev/urandom** は、Kernel Crypto API Deterministic Random Generator (DRBG) によって電源が活用されます。FIPS モードのアプリケーションは、上記のデバイスを FIPS 準拠のノイズ源として使用するため、デバイスは FIPS で承認される

アルゴリズムを使用する必要があります。この目的を達成するために、必要なフックが `/dev/random` ドライバーに追加されました。その結果、このフックは FIPS モードで有効になり、`/dev/random` および `/dev/urandom` が Kernel Crypto API DRBG に接続します。

(BZ#1785660)

libbpf が完全対応

bpftrace、**bpf/xdp** 開発などの bpf 関連アプリケーションに不可欠な **libbpf** パッケージが完全にサポートされるようになりました。

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

(BZ#1759154)

lshw ユーティリティーが追加の CPU 情報を提供

今回の機能拡張により、List Hardware ユーティリティー (**lshw**) により多くの CPU 情報が表示されます。CPU の **version** フィールドは、システムプロセッサのファミリー、モデル、およびステッピングの詳細を **version: <family>.<model>.<stepping>** として数値形式で提供するようになりました。

(BZ#1794049)

kernel-rt ソースツリーが RHEL 8.3 ツリーに更新

kernel-rt ソースが更新され、最新の Red Hat Enterprise Linux カーネルソースツリーを使用するようになりました。リアルタイムパッチセットも、最新のアップストリームバージョン v5.6.14-rt7 に更新されました。これらの更新はいずれも、バグ修正および機能強化を多数提供します。

(BZ#1818138、BZ#1818142)

tpm2-tools がバージョン 4.1.1 にリベース

tpm2-tools パッケージがバージョン 4.1.1 にアップグレードされ、コマンドの追加、更新、および削除が数多く追加されました。詳細は、[Updates to tpm2-tools package in RHEL8.3](#) を参照してください。

(BZ#1789682)

Mellanox ConnectX-6 Dx ネットワークアダプターが完全にサポートされるようになりました。

今回の機能拡張により、Mellanox ConnectX-6 Dx ネットワークアダプターの PCI ID が **mlx5_core** ドライバーに追加されました。このアダプターを使用するホストでは、RHEL は **mlx5_core** ドライバーを自動的に読み込みます。この機能は以前のリリースではテクノロジープレビュー機能として利用できていましたが、RHEL 8.3 で完全にサポートされるようになりました。

(BZ#1782831)

mlxsw ドライバーがバージョン 5.7 にリベース

mlxsw ドライバーは、アップストリームバージョン 5.7 にアップグレードされ、以下の新機能を追加します。

- バッファの占有データを提供する共有バッファの占有機能。
- **layer 2**、**layer 3**、**tunnels**、および **access control list** のドロップの監視を可能にするパケットドロップ機能。

- パケットトラップポリシーのサポート。
- Link Layer Discovery Protocol (LLDP) エージェントを使用したデフォルトのポート優先度設定のサポート。
- ETS (Enhanced Transmission Selection) および Token Bucket Filter (TBF) キュー処理のサポート。
- 初期のパケットドロップを防ぐために、Red Hat のキューイング規則の **nodrop** モードが有効になります。
- トラフィッククラス SKB 編集アクション **actionskbedit** 優先順位昨日は、パケットメタデータの変更を有効にし、**pedit** Traffic Class Offloading (TOS) で補完します。

(BZ#1821646)

クラッシュカーネルが、kdump 用のメモリー予約を拡張

今回の機能拡張により、**crashkernel=auto** 引数は、4GB から 64GB のメモリー容量を持つマシンでより多くのメモリーを予約するようになりました。以前は、メモリーが限定されたため、クラッシュカーネルはカーネルスペースおよびユーザー空間のメモリーとしてクラッシュダンプをキャプチャーできませんでした。これにより、クラッシュカーネルが OOM (Out-of-memory) エラーが発生しました。この更新は、上記のシナリオで OOM エラーの発生を軽減し、**kdump** のメモリー容量を適宜拡張するのに役立ちます。

(BZ#1746644)

5.1.9. ファイルシステムおよびストレージ

LVM が VDO ボリュームを管理可能に

LVM が、VDO (Virtual Data Optimizer) セグメントタイプをサポートするようになりました。これにより、LVM ユーティリティを使用して、ネイティブの LVM 論理ボリュームとして VDO ボリュームを作成および管理できるようになりました。

VDO は、インラインブロックレベルの重複排除、圧縮、およびシンプロビジョニング機能を提供します。

詳細は [RHEL での論理ボリュームの重複排除および圧縮](#) を参照してください。

(BZ#1598199)

SCSI スタックが、高パフォーマンスアダプターにより適切に動作するようになりました。

SCSI スタックのパフォーマンスが向上しました。その結果、次世代の高パフォーマンスホストバスアダプター (HBA) が、RHEL でより高い IOPS (I/Os 毎秒) 可能になりました。

(BZ#1761928)

megaraid_sas ドライバーが最新バージョンに更新されました。

megaraid_sas ドライバーがバージョン 07.713.01.00-rc1 に更新されました。この更新では、パフォーマンスの向上、サポートされている MegaRAID アダプターの安定性、およびリッチ機能セットに関連するバグ修正および機能強化が複数追加されました。

(BZ#1791041)

Stratis が、エラーのプール名をリスト表示

既存の Stratis プールが使用しているブロックデバイスに Stratis プールを作成しようとする、**stratis** ユーティリティーが既存プールの名前を報告するようになりました。以前は、ユーティリティーはプールの UUID ラベルのみをリスト表示していました。

(BZ#1734496)

FPIN ELS フレーム通知のサポート

lpfc Fibre Channel (FC) ドライバーが、リンクの整合性に関する Fabric Performance Impact Notifications (FPIN) に対応するようになりました。これは、リンクレベルの問題を特定し、スイッチがより信頼できるパスを選択できるようになります。

(BZ#1796565)

LVM オンディスクのメタデータのデバッグを行うための新しいコマンド

lvm2 パッケージで利用できる **pvck** ユーティリティーは、物理ボリュームで LVM のディスク上のメタデータをデバッグまたはレスキューするための低レベルのコマンドを提供するようになりました。

- メタデータを抽出するには、**pvck --dump** コマンドを使用します。
- メタデータを修復するには、**pvck --repair** コマンドを使用します。

詳細は、man ページの **pvck(8)** を参照してください。

(BZ#1541165)

LVM RAID は、デバイスのデータの破損によるデータ損失を防ぐために DM 整合性に対応します。

LVM RAID 設定に Device Mapper (DM) の整合性を追加して、データ損失を防ぐことができます。整合性層は、デバイスのデータの破損を検出し、RAID レイヤーに、LVM RAID 全体で破損したデータを修正するよう警告します。

RAID は、デバイス障害によるデータ損失を防ぎますが、LVM RAID アレイに整合性を追加すると、デバイスのデータが破損するため、データ損失を防ぐことができます。新しい LVM RAID を作成する際に整合性層を追加するか、すでに存在する LVM RAID に追加できます。

(JIRA:RHELPLAN-39320)

Resilient Storage (GFS2) は、AWS、Azure、および Aliyun パブリッククラウドでサポートされます。

Resilient Storage (GFS2) は、それらのプラットフォームで共有ブロックデバイスのサポートの導入により、3つの主要なパブリッククラウド、Amazon (AWS)、Microsoft (Azure)、および Alibaba (Aliyun) でサポートされるようになりました。これにより、GFS2 は、オンプレミスおよびパブリッククラウドの両方を使用するオプションが含まれる、実際のハイブリッドクラウドクラスターファイルシステムになりました。Microsoft Azure および AWS で共有ブロックストレージを設定する方法については、[Microsoft Azure への RHEL 8 のデプロイ](#) および [Amazon Web Services への RHEL 8 のデプロイ](#) を参照してください。Alibaba Cloud で共有ブロックストレージを設定する方法は、[Configuring Shared Block Storage for a Red Hat High Availability Cluster on Alibaba Cloud](#) を参照してください。

(BZ#1900019)

ユーザー空間が最新の **nfsdclid** デーモンに対応

ユーザー空間は、最新の **nfsdclid** デーモンしかサポートするようになりました。これは、namespace 対応のクライアント追跡メソッドのみです。今回の機能拡張により、データの破損なしに、クライアントがコンテナ化された **knfsd** デーモンから開放またはロックするようになります。

(BZ#1817756)

nconnect が複数の同時接続に対応

今回の機能拡張により、**nconnect** 機能を使用して NFS サーバーへの同時接続を複数作成できるようになり、異なる負荷分散機能が可能になりました。**nconnect=X** NFS マウントオプションを使用して **nconnect** 機能を有効にします。ここで、**X** は、使用する同時接続の数です。現在の制限は 16 です。

(BZ#1683394, BZ#1761352)

クライアント情報追跡用の **nfsdclid** デーモンに対応

今回の機能強化により、**nfsdclid** デーモンが安定したストレージのクライアントベース情報を追跡するデフォルトのメソッドになりました。その結果、コンテナで実行中の NFS v4 により、クライアントはサーバーの再起動後に開いたり、ロックの返還を要求できます。

(BZ#1817752)

5.1.10. 高可用性およびクラスター

pacemaker がバージョン 2.0.4 にリベース

Pacemaker クラスターリソースマネージャーがアップストリームのバージョン 2.0.4 にアップグレードされ、多くのバグ修正が提供されています。

(BZ#1828488)

新しい **priority-fencing-delay** クラスタープロパティー

Pacemaker は、新しい **priority-fencing-delay** クラスタープロパティーをサポートするようになりました。これにより、スプリットブレインが発生した場合に、実行中のリソースが最も少ないノードがフェンシングされるように、2 ノードクラスターを設定できます。

priority-fencing-delay プロパティーは期間に設定できます。このプロパティーのデフォルト値は 0 (無効) です。このプロパティーがゼロ以外の値に設定されている場合や、**priority** メタ属性が 1 つ以上のリソースに対して設定されている場合は、スプリットブレインが発生すると、実行されているすべてのリソースの中で、最も優先順位が高い組み合わせのノードが存続する可能性が高くなります。

たとえば、**pcs resource defaults priority=1** と **pcs property set priority-fencing-delay=15s** を設定し、他の優先度が設定されていない場合には、他のノードはフェンシングを開始するまで 15 秒間待機するため、最も多くのリソースを実行するノードが存続する可能性が高くなります。特定のリソースが他のリソースよりも重要である場合は、優先度を高く設定できます。

昇格可能なクローンに優先順位が設定されている場合は、そのクローンのマスターロールを実行しているノードの優先度が 1 ポイント追加されます。

priority-fencing-delay で設定した遅延は、フェンスデバイスプロパティー **pcmk_delay_base** および **pcmk_delay_max** の遅延に追加されます。この動作により、両方のノードの優先度が同等の場合、またはノードの損失以外の理由で両方のノードをフェンシングする必要がある場合 (例: **on-fail=fencing** がリソースモニター操作作用に設定されている)、ある程度の遅延を許容します。組み合わせで使用する場合には、優先ノードが優先されるよう、**priority-fencing-delay** プロパティーを、**pcmk_delay_base** および **pcmk_delay_max** の最大遅延よりもはるかに大きい値に設定することを推奨します (値を 2 倍すると完全に安全となります)。

(BZ#1784601)

リソースおよび操作のデフォルトの複数のセットを管理する新しいコマンド

リソースおよび操作のデフォルトの複数のセットの作成、リスト表示、変更、および削除が可能になりました。デフォルト値のセットを作成する場合は、**resource** および **op** 式が含まれるルールを指定できます。たとえば、たとえば、特定タイプのすべてのリソースにデフォルトのリソース値を設定できます。既存のデフォルト値をリスト表示するコマンドには、出力に複数のデフォルトセットが含まれるようになりました。

- **pcs resource [op] defaults set create** コマンドは、デフォルト値の新しいセットを作成します。このコマンドでルールを指定する場合は、**and**、**or**、および括弧を含む **resource** および **op** 式のみが許可されます。
- **pcs resource [op] defaults set delete | remove** コマンドは、デフォルト値を削除します。
- **pcs resource [op] defaults set update** コマンドは、セットのデフォルト値を変更します。

(BZ#1817547)

クラスターリソースのタグ付けのサポート

pcs tag コマンドを使用して、Pacemaker クラスターにクラスターリソースをタグ付けできるようになりました。この機能により、1つのコマンドで特定のリソースセットを管理できます。**pcs tag** コマンドを使用して、リソースタグを削除または変更したり、タグ設定を表示したりすることもできます。

pcs resource enable、**pcs resource disable**、**pcs resource manage**、および **pcs resource unmanage** コマンドは、引数としてタグ ID を受け入れます。

(BZ#1684676)

Pacemaker が、完全に停止するのではなく、昇格したリソースの降格により復元に対応しました。

Pacemaker クラスターに昇格可能なリソースを設定し、そのリソースに対してプロモートまたは監視アクションが失敗した場合や、リソースを実行しているパーティションがクォーラムを失った場合に、リソースが完全に停止されるようにできるようになりました。

この機能は、unpromoted モードでリソースが引き続き利用可能である場合に便利です。たとえば、データベースマスターのパーティションがクォーラムを失った場合、データベースリソースが **Master** ロールを失ったものの、読み取り専用モードで稼働させる方が望ましいといえます。したがって、読み取りが必要なアプリケーションのみが失われたクォーラムであっても動作し続けます。この機能は、正常な降格がリカバリーで十分で、完全な再起動よりも速くなる場合にも有用です。

この機能をサポートするために、以下を行います。

- **on-fail** 操作の meta-attribute は、以下の例のように **promote** アクションとともに使用すると **demote** 値を受け入れるようになりました。

```
pcs resource op add my-rsc promote on-fail="demote"
```

- **on-fail** 操作の meta-attribute は、以下の例のように、両方の **interval** がゼロ以外の値に設定され、**role** が **Master** に設定された状態で **monitor** アクションで使用した場合に **demote** 値を受け入れるようになりました。

```
pcs resource op add my-rsc monitor interval="10s" on-fail="demote" role="Master"
```

- **no-quorum-policy** クラスタープロパティーで **demote** の値が使用されるようになりました。クラスターパーティションがクォーラムを失うと、昇格したリソースが降格されますが、実行したままになり、他のリソースはすべて停止します。

操作に **denote** メタ属性を指定しても、リソースの昇格は影響を受けません。影響を受けるノードのプロモーションスコアが引き続き最高となっている場合は、再度昇格するように選択されます。

(BZ#1837747、BZ#1843079)

Pacemaker との同期を改善するための新たな **SBD_SYNC_RESOURCE_STARTUP** SBD 設定パラメーター

SBD と Pacemaker の同期をより適切に制御するために、`/etc/sysconfig/sbd` ファイルが **SBD_SYNC_RESOURCE_STARTUP** パラメーターに対応するようになりました。RHEL 8.3 以降の Pacemaker パッケージおよび SBD パッケージがインストール済みで、**SBD_SYNC_RESOURCE_STARTUP=true** が設定されている場合、SBD はデーモンの状態についての情報を Pacemaker デーモンに問い合わせます。

この設定では、Pacemaker デーモンは、SBD がコンダクトされるまで待機してから、サブデーモンを起動し、最終終了します。その結果、SBD がそれとアクティブに通信できない場合に Pacemaker はリソースを実行せず、SBD への正常なシャットダウンを報告するまで Pacemaker は終了しません。これにより、Pacemaker が最後に切断される前にリソースが実行されなかった場合は、SBD が正常なシャットダウンの瞬間を検出できなかったときに、思いも寄らないことが起こる可能性は低くなります。このようなことが発生すると、不要な再起動がトリガーされます。定義されたハンドシェイクを使用して正常なシャットダウンを検出すると、メンテナンスモードでも機能します。実行中のリソースが残っていない状況に基づいて正常なシャットダウンを検出する以前の方法は、リソースの実行がシャットダウンで触れられないため、メンテナンスモードで無効化されていました。

また、この機能を有効にすると、SBD と Pacemaker の両方が起動しても SBD が pacemaker と通信できない場合に、スプリットブレインの状況が発生しないようにします。これは、SELinux ポリシーなどの理由で発生する可能性があります。この場合、Pacemaker では、SBD が機能していないと仮定します。この新機能を有効にすると、Pacemaker は SBD が接続するまで起動を完了しません。この新機能のもう1つの利点は、SBD を有効にすると、ハートビートを使用して Pacemaker を繰り返し通信し、いつでも Pacemaker が応答しなくなった場合にノードをパニックできることです。



注記

`/etc/sysconfig/sbd` ファイルを編集するか、PCS を介して SBD を設定している場合、RPM のアップグレードは新しい **SBD_SYNC_RESOURCE_STARTUP** パラメーターにプルされません。このような場合は、この機能を実装するには、`/etc/sysconfig/sbd.rpmnew` ファイルから手動で追加するか、**sbd(8)man** ページの **Configuration via environment** セクションで説明されている設定の手順に従う必要があります。

(BZ#1718324、BZ#1743726)

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

新しいモジュールストリーム: **ruby:2.7**

RHEL 8.3 では、新しい **ruby:2.7** モジュールストリームに Ruby 2.7.1 が導入されました。このバージョンでは、RHEL 8.1 で配布される Ruby 2.6 に対するパフォーマンスの向上、バグおよびセキュリティー修正、および新機能が数多く追加されました。

主な機能強化は、次のとおりです。

- 新しい Compaction Garbage Collector (GC) が導入されました。この GC は、断片化されたメモリー領域をデフラグできます。

- Racc (Ruby yet Another Compiler-Compiler) は、1 トークンの LALR(1) (Look-Ahead Left-to-Right) - パーサージェネレーターのコマンドラインインターフェイスを提供するようになりました。
- バンドルされた REPL (Read-Eval-Print Loop) 環境であるインタラクティブな Ruby Shell (**irb**) が、マルチライン編集をサポートするようになりました。
- 機能プログラミング言語で頻繁に使用されるパターンマッチングが実験的な機能として導入されています。
- デフォルトのブロックパラメーターとしての番号付きパラメーターは、実験的な機能として導入されました。

以下のパフォーマンスの向上が実装されています。

- ファイバーの作成を加速するために、ファイバーキャッシュ戦略が変更されました。
- **CGI.escapeHTML** メソッドのパフォーマンスが改善されました。
- **Monitor** クラスおよび **MonitorMixin** モジュールのパフォーマンスが改善されました。

また、キーワード引数および位置引数の自動変換が非推奨になりました。Ruby 3.0 では、位置引数とキーワード引数が区切られます。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。

実験的な機能に対して警告を非表示にするには、**-W:no-experimental** コマンドラインオプションを使用します。非推奨の警告を無効にするには、**-W:no-deprecated** コマンドラインオプションを使用するか、コードに **Warning[:deprecated] = false** を追加します。

ruby:2.7 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。

```
# yum module install ruby:2.7
```

php:2.6 ストリームからアップグレードする場合は、[後続のストリームへの切り替え](#) を参照してください。

(BZ#1817135)

新しいモジュールストリーム: **nodejs:14**

新しいモジュールストリーム **nodejs:14** が利用できるようになりました。RHEL 8.3 に含まれている **Node.js 14** は、RHEL 8.1 で配布される **Node.js 12** に新機能およびバグ修正が数多く追加されました。

以下は、主な変更点です。

- V8 エンジンがバージョン 8.3 にアップグレードされました。
- 実験的な新しい WebAssembly System Interface (WASI) が実装されました。
- 実験的な新しい Async Local Storage API が導入されました。
- 診断レポート機能が安定しました。
- ストリーム API が強化されました。
- 実験的なモジュールの警告が削除されました。

[RHEA-2020:5101](#) アドバイザリーのリリースでは、RHEL 8 は **Node.js 14.15.0** を提供します。これは、安定性が改善された最近の LTS (Long Term Support) です。

nodejs:14 モジュールストリームをインストールするには、以下を使用します。

```
# yum module install nodejs:14
```

nodejs:12 ストリームからアップグレードする場合は、[後続のストリームへの切り替え](#) を参照してください。

([BZ#1815402](#)、[BZ#1891809](#))

git がバージョン 2.27 にリベース

git パッケージがアップストリームバージョン 2.27 にアップグレードされました。以前に利用できたバージョン 2.18 への主な変更点は、以下のとおりです。

- **git checkout** コマンドは、2つの異なるコマンドに分割されました。
 - ブランチ管理用の **git switch**
 - ディレクトリーツリー内での変更を管理するための **git restore**
- **git rebase** コマンドの動作は、以前の **patch+apply** ワークフローではなく、デフォルトで **merge** ワークフローに基づいています。以前の動作を保持するには、**apply** する **rebase.backend** 設定変数を設定します。
- **git difftool** コマンドをリポジトリ以外でも使用できるようになりました。
- より具体的なケースで **user.{name,email}** を上書きするように、新しい設定変数 **{author,committer}.{name,email}** が導入されました。
- ユーザーがプロキシとの通信に SSL を設定できるようにする新しいオプションが追加されました。
- **git fast-export** および **git fast-import** ユーティリティーで、UTF-8 文字エンコーディングでのログメッセージによるコミットの処理が改善されました。
- **lfs** 拡張が、新しい **git-lfs** パッケージとして追加されました。Git Large File Storage (LFS) は、大規模なファイルを **Git** 内のテキストポインターに置き換え、ファイルの内容をリモートサーバーに保存します。

([BZ#1825114](#)、[BZ#1783391](#))

Python の変更点

RHEL 8.3 では、**python38:3.8** モジュールストリームに以下の変更が加えられています。

- **Python** インタープリターがバージョン 3.8.3 に更新されました。これにより、バグ修正が提供されています。
- **python38-pip** パッケージがバージョン 19.3.1 に更新され、**pip** は **manylinux2014 wheel** をインストールするようになりました。

python3 パッケージが提供する **Python 3.6** インタープリターのパフォーマンスが大幅に改善しました。

ubi8/python-27、**ubi8/python-36**、および **ubi8/python-38** コンテナイメージが、カスタムパッケー

ジインデックスまたは PyPI ミラーからの **pipenv** ユーティリティーのインストールに対応するようになりました。以前のリリースでは、**pipenv** はアップストリームの PyPI リポジトリからのみダウンロードでき、アップストリームリポジトリが利用できないと、インストールに失敗していました。

([BZ#1847416](#)、[BZ#1724996](#)、[BZ#1827623](#)、[BZ#1841001](#))

新しいモジュールストリーム: php:7.4

RHEL 8.3 では **PHP 7.4** が導入され、バージョン 7.3 に対するバグ修正および機能拡張が数多く追加されました。

本リリースでは、新しい実験的な拡張である Foreign Function Interface (FFI) が導入されています。これにより、ネイティブ機能呼び出し、ネイティブ変数へのアクセス、C ライブラリーで定義されたデータ構造の作成およびアクセスが可能になります。FFI 拡張は **php-ffi** パッケージで利用できます。

以下の拡張機能が削除されました。

- **wddx** 拡張 (**php-xml** パッケージから削除)
- **php-recode** パッケージから削除された **recode** 拡張機能。

php:7.4 モジュールストリームをインストールするには、以下を使用します。

```
# yum module install php:7.4
```

php:7.3 ストリームからアップグレードする場合は、[後続のストリームへの切り替え](#) を参照してください。

RHEL 8 での PHP の使用方法の詳細は、[PHP スクリプト言語の使用](#) を参照してください。

([BZ#1797661](#))

新しいモジュールストリーム: nginx:1.18

nginx 1.18 Web およびプロキシサーバーが利用できるようになりました。バージョン 1.16 に対するバグ修正、セキュリティ修正、および拡張機能が数多く追加されました。以下は、主な変更点です。

- HTTP リクエストレートおよび接続制限に対する機能強化が実装されました。たとえば、**limit_rate** ディレクティブおよび **limit_rate_after** ディレクティブは、新しい **\$limit_req_status** 変数および **\$limit_conn_status** 変数を含む変数をサポートするようになりました。さらに、**limit_conn_dry_run** ディレクティブおよび **limit_req_dry_run** ディレクティブにドライランモードが追加されました。
- 新しい **auth_delay** ディレクティブが追加されました。これにより、承認されていないリクエストの遅延処理が可能になります。
- 以下のディレクティブは、**grpc_pass**、**proxy_upload_rate**、および **proxy_download_rate** の変数に対応するようになりました。
- 追加の PROXY プロトコル変数 (**\$proxy_protocol_server_addr** および **\$proxy_protocol_server_port**) が追加されました。

nginx:1.18 ストリームをインストールするには、以下を実行します。

```
# yum module install nginx:1.18
```


nginx:1.16 ストリームからアップグレードする場合は、[後続のストリームへの切り替え](#) を参照してください。

([BZ#1826632](#))

新しいモジュールストリーム: perl:5.30

RHEL 8.3 では **Perl 5.30** が導入され、以前にリリースされた **Perl 5.26** にバグ修正および機能拡張が数多く追加されました。新しいバージョンは、特定の言語機能も非推奨または削除します。大きな影響を及ぼす主な変更点には、以下が含まれます。

- **Math::BigInt::CalcEmu** モジュール、**arybase** モジュール、および **B::Debug** モジュールが削除された
- ファイル記述子が **close-on-exec** フラグで開くようになった
- 同じシンボルをファイルおよびディレクトリーリハンドルとして開くことはできなくなった
- サブルーチンの属性は、サブルーチンの署名の前に置く必要がある
- **:locked** 属性および **:uniq** 属性が削除された
- フォーマットでのコンマなしの変数リストが許可されなくなった
- 裸の **<<** ヒアドキュメント演算子は使用できなくなった
- エスケープされていない左中括弧 (**{**) 文字を正規表現パターンで使用することは非推奨になっていたが、使用できなくなった
- **AUTOLOAD()** サブルーチンがメソッド以外の関数に継承できなくなった
- **sort** プラグマが **sort** アルゴリズムを指定できなくなった
- **B::OP::terse()** サブルーチンが **B::Concise::b_terse()** サブルーチンに置き換えられた
- **File::Glob::glob()** 関数が **File::Glob::bsd_glob()** 関数に置き換えられた
- **dump()** 関数が、**CORE::dump()** として完全修飾して呼び出さなければいけなくなった
- **yada-yada** 演算子 (**...**) がステートメントで、式として使用できなくなった
- **\$[** 変数にゼロ以外の値を割り当てると、致命的なエラーが返されるようになった
- **\$*** および **\$#** 変数が許可されなくなった
- 偽の条件ブランチで **my()** 関数を使用した変数の宣言が許可されなくなった
- **:utf8** バンドルの **sysread()** 関数および **syswrite()** 関数は、致命的なエラーを返すようになる
- **pack()** 関数が不適切な形式の UTF-8 形式を返さなくなった
- **IV_MAX** より値が大きい Unicode コードポイントが許可されなくなった
- Unicode 12.1 に対応するようになった

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

Perl 5.30 は、s2i が有効な **ubi8/perl-530** コンテナイメージとしても利用できます。

([BZ#1713592](#)、[BZ#1732828](#))

新しいモジュールストリーム: **perl-libwww-perl:6.34**

RHEL 8.3 では、新しい **perl-libwww-perl:6.34** モジュールストリームが導入され、RHEL 8 で利用可能なすべてのバージョンの **Perl** に **perl-libwww-perl** パッケージが提供されています。RHEL 8.0 以降の、モジュールでない **perl-libwww-perl** パッケージは、5.26 以外の **Perl** ストリームで使用できません。新しいデフォルトの **perl-libwww-perl:6.34** ストリームは廃止されました。

([BZ#1781177](#))

新しいモジュールストリーム: **perl-IO-Socket-SSL:2.066**

新しい **perl-IO-Socket-SSL:2.066** モジュールストリームが利用できるようになりました。本モジュールでは、**perl-IO-Socket-SSL** パッケージおよび **perl-Net-SSLeay** パッケージを提供し、RHEL 8 で利用可能なすべての **Perl** ストリームと互換性があります。

([BZ#1824222](#))

squid:4 モジュールストリームがバージョン 4.11 にリベース

squid:4 モジュールストリームが提供する **Squid** プロキシサーバーが、バージョン 4.4 からバージョン 4.11 にアップグレードされました。本リリースでは、複数のバグ修正およびセキュリティ修正と、新しい設定オプションなどのさまざまな機能拡張が提供されます。

([BZ#1829467](#))

httpd:2.4 モジュールストリームの変更点

RHEL 8.3 では、**httpd:2.4** モジュールストリームで利用可能な Apache HTTP Server に以下のような変更が追加されました。

- **mod_http2** モジュールがバージョン 1.15.7 にリベース
- **H2Upgrade** ディレクティブおよび **H2Push** ディレクティブの設定変更
- HTTP/2 ペイロードフレームのパディングを制御する新しい **H2Padding** 設定ディレクティブ
- 多くのバグ修正。

([BZ#1814236](#))

httpd の **CustomLog** ディレクティブから **journald** へのロギングのサポート

CustomLog ディレクティブの新規オプションを使用して、Apache HTTP Server から **journald** にアクセス (transfer) ログを出力できるようになりました。

サポートされる構文は以下のとおりです。

```
CustomLog journald:priority format|nickname
```

priority は、**LogLevel** ディレクティブで使用されるように **debug** する優先度の高い文字列です。

たとえば、**combined** ログ形式を使用して **journald** にログインするには、以下を使用します。

```
CustomLog journald:info combined
```

このオプションを使用する場合は、フラットファイルに直接ログインする場合よりもサーバーのパフォーマンスが低くなる可能性があることに注意してください。

([BZ#1209162](#))

5.1.12. コンパイラーおよび開発ツール

.NET 5 が RHEL で利用可能に

.NET 5 は、Red Hat Enterprise Linux 7、Red Hat Enterprise Linux 8、および OpenShift Container Platform で利用できます。.NET 5 には、新しい言語バージョン (C# 9 および F# 5.0) が含まれます。ベースライブラリー、GC および JIT でパフォーマンスが大幅に改善されました。.NET 5 では単一ファイルアプリケーション、.NET アプリケーションを1つの実行可能ファイルとして配布でき、すべての依存関係が含まれます。.NET 5 用の UBI8 イメージは Red Hat コンテナレジストリーから利用でき、OpenShift で使用できます。

.NET 5 を使用するには、**dotnet-sdk-5.0** パッケージをインストールします。

```
$ sudo dnf install -y dotnet-sdk-5.0
```

詳細は [.NET 5 のドキュメント](#) を参照してください。

([BZ#1944677](#))

新しい GCC Toolset 10

GCC Toolset 10 は最新バージョンの開発ツールを提供するコンパイラーツールセットです。このツールセットは、**AppStream** リポジトリーにおいて、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 10.2.1 に更新され、アップストリームの GCC で利用可能なバグ修正および機能拡張が数多く追加されました。

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

ツール	バージョン
GCC	10.2.1
GDB	9.2
Valgrind	3.16.0
SystemTap	4.3
Dyninst	10.1.0
binutils	2.35
elfutils	0.180
dwz	0.12

ツール	バージョン
make	4.2.1
strace	5.7
ltrace	0.7.91
annobin	9.29

GCC Toolset 10 をインストールするには、`root` で以下のコマンドを実行します。

```
# yum install gcc-toolset-10
```

GCC Toolset 10 のツールを実行するには、以下のコマンドを実行します。

```
$ scl enable gcc-toolset-10 tool
```

GCC Toolset バージョン 10 のツールバージョンが、このようなツールのシステムバージョンをオーバーライドするシェルセッションを実行するには、次のコマンドを実行します。

```
$ scl enable gcc-toolset-10 bash
```

詳細は、[Using GCC Toolset](#) を参照してください。

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

- GCC コンパイラー、GDB デバッガー、**make** 自動化ツールを含む **rhel8/gcc-toolset-10-toolchain**
- SystemTap や Valgrind などのパフォーマンス監視ツールを含む **rhel8/gcc-toolset-10-perftools**
コンテナイメージをプルするには、`root` で以下のコマンドを実行します。

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

GCC Toolset 10 コンテナイメージのみがサポートされるようになりました。以前のバージョンの GCC Toolset コンテナイメージが非推奨になりました。

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

(BZ#1842656)

Rust Toolset がバージョン 1.45.2 にリベース

Rust Toolset が、バージョン 1.45.2 に更新されました。以下は、主な変更点です。

- 依存関係を表示するサブコマンド **cargo tree** が **cargo** に組み込まれるようになりました。
- 浮動小数点値から整数にキャストすると、まとまったキャストが生成されます。以前は、ターゲット整数タイプの省略された浮動小数点値が範囲外であった場合、結果はコンパイラーで未定義の動作でした。有限でない浮動小数点の値も未定義になりました。今回の機能拡張によ

り、有限値は整数の最小範囲または最大範囲に切り捨てられるようになりました。正の値と負の値はデフォルトで、それぞれ最大整数と最小整数値である Not-a-Number(NaN) の値をゼロにまとめられます。

- 式、パターン、ステートメントの関数のような手順マクロが拡張され、安定化されるようになりました。

使用方法の詳細は [Rust Toolset の使用](#) を参照してください。

(BZ#1820593)

LLVM Toolset がバージョン 10.0.1 にリベース

LLVM Toolset がバージョン 10.0.1 にアップグレードされました。今回の更新で、**clang-libs** パッケージに個別のコンポーネントライブラリーが含まれなくなりました。その結果、アプリケーションにアプリケーションをリンクできなくなりました。**clang** ライブラリーにアプリケーションをリンクするには、**libclang-cpp.so** パッケージを使用します。

詳細は、[LLVM Toolset の使用](#) を参照してください。

(BZ#1820587)

Go Toolset がバージョン 1.14.7 にリベース

Go Toolset がバージョン 1.14.7 の主な変更点にアップグレードされました。

- Go モジュールシステムが完全にサポートされるようになりました。
- SSL バージョン 3.0 (SSLv3) はサポートされなくなりました。Delve デバッガーの主な機能強化は、以下のとおりです。
- raw メモリーを調べる新しいコマンド **examinemem** (または **x**)
- プログラムの停止ごとに式の **display** 値を出力する新しいコマンド
- デバッグしたプログラムの Teletypewriter (TTY) を指定するための新しい **--tty** フラグ
- Arm64 の新しいコアダンプサポート
- goroutine ラベルを出力する新しい機能
- Debug Adapter Protocol (DAP) サーバーのリリース
- **dlv trace** および **trace** REPL (read-eval-print-loop) コマンドからの出力が改善されました。

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

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

(BZ#1820596)

SystemTap がバージョン 4.3 にリベース

SystemTap 計測ツールがバージョン 4.3 に更新され、バグ修正および機能強化が複数追加されました。以下は、主な変更点です。

- ユーザー空間プローブは、**readelf -n** の 16 進数 **buildid** でターゲットに設定できます。この代替パス名を使用すると、一致するバイナリーを任意の名前でプローブできるため、1つのスクリプトで異なるバージョンの範囲をターゲットにできます。この機能は、elfutils **debuginfod**

サーバーと連携して機能します。

- スクリプト関数はプローブ `$context` 変数を使用してプローブされた場所内の変数にアクセスできます。これにより、SystemTap スクリプトは共通のロジックを使用してさまざまなプローブと連携できます。
- `try-catch` ステートメントやエラープローブを含む `stapbpf` プログラムが改善され、BPF バックエンドで実行されているスクリプトで適切なエラー耐性を持たせるようになりました。

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

(BZ#1804319)

Valgrind がバージョン 3.16.0 にリベース

Valgrind 実行コード分析ツールがバージョン 3.16.0 に更新されました。これにより、以前のバージョンに対するバグ修正や機能強化が数多く追加されました。

- 次のいずれかの方法により、Valgrind でプログラムを実行しながら、多くのコマンドラインオプションの値を動的に変更できるようになりました。`vgdb` 経由、Valgrind `gdbserver` に接続されている `gdb` 経由、またはプログラムクライアントリクエスト経由。動的に変更可能なオプションのリストを表示するには、`valgrind --help-dyn-options` コマンドを実行します。
- `Cachegrind` (`cg_annotate`) および `Callgrind` (`callgrind_annotate`) ツールについては、`--auto` と `--show-percs` オプションがデフォルトで `yes` になりました。
- `Memcheck` ツールを使用すると、最適化されたコードの誤検出エラーが少なくなります。特に、`Memcheck` はコンパイラーが `A && B` チェックを `B && A` に変換した場合に適切に処理されるようになりました。`B` 未定義で、`A` が `false` でした。`Memcheck` は、部分的に定義された値の整数等号チェックと不等号チェックも処理します。
- 実験的なスタックおよびグローバルアレイチェックツール (`exp-sgcheck`) が削除されました。スタックおよびグローバルアレイのオーバーランを検出する方法は、GCC の `AddressSanitizer` (ASAN) 機能を使用することです。これには、`-fsanitize=address` オプションでコードを再ビルドする必要があります。

(BZ#1804324)

elfutils がバージョン 0.180 にリベース

`elfutils` パッケージがバージョン 0.180 にアップグレードされ、バグ修正および機能強化が複数追加されました。以下は、主な変更点です。

- GCC LTO (リンク時間最適化) で構築されたコードのデバッグ情報への対応が改善されました。`eu-readelf` および `libdw` ユーティリティーで `.gnu.debuglto_` セクションの読み取りと処理が可能になり、コンパイルユニット (CU) 全体に定義される関数のファイル名を正しく解決できるようになりました。
- `eu-nm` ユーティリティーは、`V` を弱いオブジェクト、`C` を共通のシンボルとして明示的に特定するようになりました。
- `debuginfod` サーバーは `.deb` アーカイブをインデックス化できるようになりました。また、`-Z EXT[=CMD]` オプションを使用して他のパッケージアーカイブ形式を追加するための汎用拡張があります。たとえば、`-Z '.tar.zst=zstdcat'` は、`zstdcat` ユーティリティーを使用して、`.tar.zst` 拡張で終わるアーカイブをデプロイメントする必要があることを示しています。

- **debuginfo-client** ツールには複数の新しいヘルパー関数があります。たとえば、**debuginfod_set_user_data**、**debuginfod_get_user_data**、**debuginfod_get_url**、**anddebuginfod_add_http_header** となります。**file://** URL もサポートしています。

(BZ#1804321)

GDB が IBM z15 でのプロセスレコードおよび再生に対応

今回の機能強化により、GNU Debugger (GDB) は、IBM z15 プロセッサのほとんどの新しい命令で、プロセスレコードおよび再生をサポートするようになりました (以前は arch13 と呼ばれます)。現在、以下の指示はサポートされていません。SORTL (sort list)、DFLTCC (deflate conversion call)、KDSA (コンピュートデジタル署名認証)。

(BZ#1659535)

Marvell ThunderX2 パフォーマンス監視イベントが papi で更新されました。

今回の機能拡張により、未コアイベントを含む ThunderX2 に固有のパフォーマンスイベントが多数更新されました。その結果、開発者は Marvell ThunderX2 システムでシステムパフォーマンスをより適切に調査できます。

(BZ#1726070)

glibc math ライブラリーが IBM Z に対して最適化

今回の機能拡張により、IBM Z マシンのパフォーマンスを向上するために、libm **math** 機能が最適化されました。以下は、主な変更点です。

- 不要な浮動小数点コントロールレジスターセットおよび抽出を回避するように、丸めるモード処理を改善
- z196 整数と浮動小数点間の変換の悪用

(BZ#1780204)

追加の libffi 固有の一時ディレクトリーが利用可能

以前は、強化されたシステムでは、システム全体の一時ディレクトリーには、**libffi** ライブラリーとの使用に適したパーミッションがない可能性があります。

今回の機能強化により、システム管理者は、**write** および **exec** マウントまたは **selinux** パーミッションの両方を持つ libffi 固有の一時ディレクトリーを参照するように **LIBFFI_TMPDIR** 環境変数を設定できるようになりました。

(BZ#1723951)

strstr() および strcasestr() のパフォーマンスを改善

今回の更新で、サポートされる複数のアーキテクチャーで **strstr()** 関数および **strcasestr()** 関数のパフォーマンスが改善されました。その結果、文字列およびメモリー操作ルーチンを使用したすべてのアプリケーションのパフォーマンスを大幅に向上させるようになりました。

(BZ#1821531)

glibc が、省略されたロケールアーカイブの読み込みを正しく処理

アップグレード時またはディスク障害の発生時に、システムロケールのアーカイブが切り捨てられていたり、システムロケールのアーカイブが切り捨てられると、アーカイブの読み込み時にプロセスが予期せず終了する可能性があります。今回の機能拡張により、ロケールアーカイブのロードに一貫性チェッ

クが追加されます。その結果、プロセスはアーカイブ切り捨てを検出し、アーカイブされていないロケールまたはデフォルトの POSIX ロケールのいずれかにフォールバックできるようになりました。

(BZ#1784525)

GDB が debuginfod に対応

今回の機能強化により、GNU Debugger (GDB) は、elfutils **debuginfod** クライアントライブラリーを使用して、集中型サーバーからデバッグ情報パッケージをダウンロードできるようになりました。

(BZ#1838777)

pcp がバージョン 5.1.1-3 にリベース

pcp パッケージが、バージョン 5.1.1-3 にアップグレードされました。以下は、主な変更点です。

- サービスユニットを更新し、すべての PCP サービスの **systemd** 統合および信頼性が向上します。アーカイブログローテーションおよび時間圧縮が向上しました。**pmproxy** プロトコルで検出に関するバグの修正。
- **pmrep** およびエクスポートツールのメトリックラベルレポートとともに、**pcp-atop**、**pcp-dstat**、**pmrep**、および関連の監視ツールが改善されました。
- **bpfftrace**、**OpenMetrics**、MMV、Linux カーネルエージェントなどのコレクションエージェントが改善されました。**Open vSwitch** および **RabbitMQ** サーバーの新しいメトリックコレクター。
- スタンドアロンの **pmmgr** デーモンに代わる、新しいホスト検出 **pmfind systemd** サービス。

(BZ#1792971)

grafana がバージョン 6.7.3 にリベース

grafana パッケージがバージョン 6.7.3 にアップグレードされました。以下は、主な変更点です。

- 汎用 **OAuth** ロールマッピングのサポート
- 新規ログパネル
- テーブルパネルに複数行のテキスト表示
- 新しい通貨および energy ユニット

(BZ#1807323)

grafana-pcp がバージョン 2.0.2 にリベース

grafana-pcp パッケージがバージョン 2.0.2 にアップグレードされました。以下は、主な変更点です。

- フレームグラフでグラフ付けされるマルチディメンションの **eBPF** マップをサポートします。
- クエリーエディターで自動補完キャッシュを削除し、PCP メトリックを動的に表示できるようにします。

(BZ#1807099)

新しい rhel8/pcp コンテナイメージ

Red Hat コンテナレジストリーで **rhel8/pcp** コンテナイメージが利用できるようになりました。イメージには Performance Co-Pilot (PCP) ツールキットが含まれます。これには、**pcp-zeroconf** パッケージおよび **OpenMetrics** PMDA が事前にインストールされます。

(BZ#1497296)

新しい **rhel8/grafana** コンテナイメージ

rhel8/grafana コンテナイメージが、Red Hat コンテナレジストリーで利用できるようになりました。Grafana は、メトリックダッシュボードを含むオープンソースユーティリティーで、**Graphite**、**Elasticsearch**、**OpenTSDB**、**Prometheus**、**InfluxDB**、および **PCP** 監視ツールのグラフィックエディターです。

(BZ#1823834)

5.1.13. ID 管理

IdM のバックアップユーティリティーが、必要なレプリカロールを確認するようになりました。

ipa-backup ユーティリティーは、認証局 (CA)、Domain Name System (DNS)、Key Recovery Agent (KRA) などの IdM クラスターで使用されるすべてのサービスが、バックアップを実行しているレプリカにインストールされているかどうかを確認するようになりました。レプリカにこれらのサービスがすべてインストールされていない場合、そのホストで取得したバックアップはクラスターのフル復元には十分ではないため、**ipa-backup** ユーティリティーは警告を表示して終了します。

たとえば、IdM デプロイメントで統合認証局 (CA) を使用している場合、CA 以外のレプリカのバックアップは CA データを取得しません。Red Hat は、**ipa-backup** を実行するレプリカに、クラスターで使用される IdM サービス がすべてインストールされていることを確認することを推奨します。

詳細は、[IdM バックアップによるデータ損失の準備](#) を参照してください。

(BZ#1810154)

新しいパスワード有効期限の通知ツール

ipa-client-epn パッケージが提供する Password Notification (EPN) は、パスワードの期限が切れる Identity Management (IdM) ユーザーのリストを構築するために使用できるスタンドアロンツールです。

IdM 管理者は、EPN を使用して以下を行うことができます。

- JSON 形式で影響を受けるユーザーのリストを表示します。これはランタイム時に計算されません。
- 特定の日または日付の範囲に送信される電子メール数を計算する。
- パスワード期限切れのメール通知をユーザーに送信する。

Red Hat は、追加する **ipa-epn.timer systemd** タイマーを使用して、IdM クライアントまたはレプリカから1日1回起動することを推奨します。

(BZ#913799)

JSS が FIPS 準拠の SSLContext を提供するようになりました。

以前のリリースでは、Tomcat は Java Cryptography Architecture (JCA) SSLContext クラスの SSLEngine ディレクティブを使用していました。デフォルトの SunJSSE 実装は連邦情報処理標準 (FIPS) に準拠していないため、PKI は JSS 経由で FIPS 準拠の実装を提供するようになりました。

(BZ#1821851)

公開鍵インフラストラクチャーの全体的な正常性を確認できるようになりました。

今回の更新で、公開鍵インフラストラクチャー (PKI) Healthcheck ツールが、RHEL 8.1 で導入された Identity Management (IdM) Healthcheck ツールに PKI サブシステムの正常性を報告するようになりました。IdM Healthcheck を実行すると、PKI Healthcheck が呼び出され、これにより、PKI サブシステムの正常性レポートを収集して返します。

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

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

(BZ#1770322)

RSA PSS のサポート

今回の機能強化により、PKI は RSA PSS (Probabilistic Signature Scheme) 署名アルゴリズムに対応するようになりました。

この機能を有効にするには、特定のサブシステムに対して **pkispawn** スクリプトファイルに以下の行を設定します (**pki_use_pss_rsa_signing_algorithm=True**)。

これにより、このサブシステムに対する既存のデフォルト署名アルゴリズム (**CS.cfg** 設定ファイルで指定) はすべて、対応する PSS バージョンを使用します。たとえば、**SHA256withRSA** が **SHA256withRSA/PSS** になります。

(BZ#1824948)

Directory Server は、サービスの起動時にプライベートキーと証明書をプライベート名前空間にエクスポートします。

Directory Server は、レプリカ合意などの発信接続に OpenLDAP ライブラリーを使用します。これらのライブラリーはネットワークセキュリティーサービス (NSS) データベースに直接アクセスできないため、Directory Server は、TLS 暗号化サポートのあるインスタンスで秘密鍵と証明書を抽出し、OpenLDAP ライブラリーが暗号化された接続を確立できるようにします。以前は、Directory Server は、**cn=config** エントリー (デフォルト: **/etc/dirsrv/slapd-<instance_name>/**) の **nsslapd-certdir** パラメーターで設定したディレクトリーに、秘密鍵と証明書をデプロイメントしました。これにより、Directory Server は、このディレクトリーに **Server-Cert-Key.pem** および **Server-Cert.pem** を保存します。今回の機能強化により、Directory Server が、**systemd** が **/tmp/** ディレクトリーにマウントされるプライベート名前空間に秘密鍵と証明書を抽出するようになりました。その結果、セキュリティーが強化されました。

(BZ#1638875)

ディスクの監視のしきい値に達すると、Directory Server がインスタンスの読み取り専用モードに切り替わるようになりました。

今回の更新で、**nsslapd-disk-monitoring-readonly-on-threshold** パラメーターを **cn=config** エントリーに追加されました。この設定を有効にすると、Directory Server は、ディスクの監視が有効で、空きディスク領域が **nsslapd-disk-monitoring-threshold** に設定した値よりも小さい場合に、すべての

データベースを読み取り専用に切り替えます。**nsslapd-disk-monitoring-readonly-on-threshold** を **on** に設定すると、Directory Server がインスタンスを正常にシャットダウンするまでデータベースを変更することはできません。これにより、データの破損を防ぐことができます。

(BZ#1728943)

samba がバージョン 4.12.3 にリベース

samba パッケージがアップストリームバージョン 4.12.3 にアップグレードされ、以前のバージョンに対するバグ修正や機能強化が数多く追加されました。

- 組み込み暗号機能は、GnuTLS 機能に置き換えられました。これにより、サーバーメッセージブロックバージョン 3 (SMB3) のパフォーマンスが改善され、コピー速度が大幅に改善されます。
- ランタイムの最小サポートが Python 3.5 になりました。
- 以前の書き込みキャッシュの概念が原因で、**write cache size** パラメーターが削除され、メモリ制約のあるシステムのパフォーマンスが低下します。
- DES 暗号化タイプによる Kerberos チケットを使用した接続の認証のサポートが削除されました。
- **vfs_netatalk** 仮想ファイルシステム (VFS) モジュールが削除されました。
- **ldap ssl ads** パラメーターは非推奨としてマークされ、今後の Samba バージョンで削除されます。または、LDAP トラフィックを暗号化する方法および詳細は、[samba: removal of "ldap ssl ads" smb.conf option](#) ソリューションを参照してください。
- デフォルトでは、RHEL 8.3 の Samba は非推奨の RC4 暗号スイートに対応しなくなりました。Kerberos 認証に RC4 を必要とする AD で Samba をドメインメンバーとして実行する場合は、**update-crypto-policies --set DEFAULT:AD-SUPPORT** コマンドを使用して RC4 暗号化タイプのサポートを有効にします。

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

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

(BZ#1817557)

cockpit-session-recording がバージョン 4 にリベース

cockpit-session-recording モジュールがバージョン 4 にリベースされました。このバージョンでは、以前のバージョンに対する以下の主な変更点があります。

- **metainfo** ファイルの親 ID を更新しました。
- パッケージマニフェストを更新しました。
- CentOS7 で正しいパスを解決するように **rpmmacro** を修正しました。
- バイトアレイでエンコードされたジャーナルデータを処理します。
- 非推奨の React ライフサイクル機能からコードを移動しました。

(BZ#1826516)

krb5 がバージョン 1.18.2 にリベース

krb5 パッケージがアップストリームバージョン 1.18.2 にアップグレードされました。主なバグ修正と機能強化は、以下のとおりです。

- シングルおよびトリプル DES 暗号化タイプが削除されました。
- Active Directory のサポート対象バージョンに必要がなく、ドラフト 9 の PKINIT が削除されました。
- NegoEx メカニズムプラグインがサポートされるようになりました。
- ホスト名正規化のフォールバックがサポートされるようになりました (`dns_canonicalize_hostname = fallback`)。

(BZ#1802334)

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

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

- **config** モジュールにより、IdM 内でグローバル設定パラメーターを設定できます。
- **dnsconfig** モジュールを使用すると、グローバル DNS 設定を変更できます。
- **dnsforwardzone** モジュールを使用すると、IdM から DNS フォワーダーを追加または削除できます。
- **dnsrecord** を使用すると、DNS レコードを管理できます。アップストリームの **ipa_dnsrecord** とは対照的に、1つの実行での複数のレコード管理が可能になり、より多くのレコードタイプをサポートします。
- **dnszone** モジュールにより、DNS サーバーでゾーンを設定できます。
- **service** モジュールにより、サービスの有無について確認することができます。
- **vault** モジュールを使用すると、vault と vault のメンバーが存在することを確認できます。

ipagroup モジュールおよび **ipahostgroup** モジュールは、ユーザーとホストグループのメンバーシップマネージャーを組み込むように拡張されています。グループメンバーシップマネージャーは、グループにメンバーを追加したり、グループからメンバーを削除できるユーザーまたはグループです。詳細は、それぞれの `/usr/share/doc/ansible-freeipa/README-*` ファイルの **Variables** セクションを参照してください。

(JIRA:RHELPLAN-49954)

IdM が、証明書管理用の新しい Ansible システムロールに対応

Identity Management (IdM) は、証明書管理タスクを自動化するために新しい Ansible システムロールをサポートします。新しいロールには以下の利点があります。

- このロールは、証明書の発行および更新を自動化するのに役立ちます。
- ロールは、**ipa** 認証局が証明書を発行するように設定できます。このようにして、既存の IdM インフラストラクチャーを使用して、証明書トラストチェーンを管理できます。

- このロールでは、サービスの停止や開始など、証明書の発行前および発行後に実行するコマンドを指定できます。

(JIRA:RHELPLAN-50002)

Identity Management が FIPS に対応

今回の機能拡張により、Identity Management (IdM) の認証メカニズムを使用して、連邦情報処理標準 (FIPS) によって承認された暗号化タイプを使用できるようになりました。IdM と Active Directory との間のフォレスト間の信頼は、FIPS に準拠していないことに注意してください。

FIPS が必要なにもかかわらず、AD 信頼を必要としないお客様は、FIPS モードで IdM をインストールできるようになりました。

(JIRA:RHELPLAN-43531)

idm:DL1 の OpenDNSSEC がバージョン 2.1 にリベース

idm:DL1 モジュールストリームの OpenDNSSEC コンポーネントが、現在の長期アップストリームサポートバージョンの 2.1 バージョンシリーズにアップグレードされました。opendnssec は、DNSSEC (Domain Name System Security Extensions) の導入を促進するオープンソースプロジェクトで、インターネットセキュリティをさらに強化します。opendnssec 2.1 では、以前のバージョンに対するバグ修正および機能拡張が数多く追加されました。詳細は、アップストリームのリリースノート <https://www.opendnssec.org/archive/releases/> を参照してください。

(JIRA:RHELPLAN-48838)

IdM が、新しいシステム全体の暗号化ポリシーで非推奨の RC4 暗号スイートに対応

今回の更新で、Identity Management (IdM) で Rivest Cipher 4 (RC4) 暗号スイートを有効にする新しい **AD-SUPPORT** 暗号化サブポリシーが導入されました。

IdM-Active Directory (AD) フォレスト間の信頼のコンテキストで管理者は、AD が Advanced Encryption Standard (AES) を使用するように設定されていない場合に、新しい **AD-SUPPORT** サブポリシーをアクティブにできます。具体的には、以下の条件のいずれかが当てはまる場合は、Red Hat では新しいサブポリシーを有効にすることを推奨します。

- AD のユーザーまたはサービスアカウントには RC4 暗号化キーがあり、AES 暗号化キーがありません。
- 個々の Active Directory ドメイン間の信頼リンクには RC4 暗号化鍵があり、AES 暗号化キーがありません。

DEFAULT 暗号化ポリシーに加えて **AD-SUPPORT** サブポリシーを有効にするには、以下を入力します。

```
# update-crypto-policies --set DEFAULT:AD-SUPPORT
```

または、強力な AES 暗号化タイプをサポートするように AD フォレスト内の AD ドメイン間で信頼をアップグレードするには、Microsoft の記事を参照してください: [AD DS: セキュリティー: 信頼されたドメインのリソースにアクセスしているときの KerberosUnsupported etype エラー](#)。

(BZ#1851139)

新しい Microsoft LDAP チャンネルバインディングおよび LDAP 署名要件への調整

最近の Microsoft の更新では、Active Directory (AD) が、LDAP チャンネルバインディングおよび LDAP 署名のデフォルトの Windows 設定を使用しないクライアントにフラグを付けます。これにより、AD と

の直接統合または間接統合に System Security Services Daemon (SSSD) を使用する RHEL システムは、Generic Security Services Application Program Interface (GSSAPI) を使用する Simple Authentication and Security Layer (SASL) 操作が正常に実行されると、AD でエラーイベント ID をトリガーする可能性があります。

これらの通知を回避するには、GSSAPI の代わりに Simple および Protected GSSAPI Negotiation Mechanism (GSS-SPNEGO) SASL メカニズムを使用するようにクライアントアプリケーションを設定します。SSSD を設定するには、`ldap_sasl_mech` オプションを **GSS-SPNEGO** に設定します。

さらに、チャンネルバインディングが AD 側で適用されている場合、以下のように SSL/TLS で SASL を使用するシステムを設定します。

1. RHEL 8.3 以降に同梱された **cyrus-sasl** パッケージ、**openldap** パッケージ、および **krb5-libs** パッケージの最新バージョンをインストールします。
2. `/etc/openldap/ldap.conf` ファイルで、**SASL_CBINDING** オプションを **tls-endpoint** に設定して正しいチャンネルバインディングタイプを指定します。

詳細は、[Is of Microsoft Security Advisory ADV190023 | LDAP Channel Binding and LDAP Signing on RHEL and AD integration](#) を参照してください。

(BZ#1873567)

SSSD、`adcli`、および `realmd` が、新しいシステム全体の暗号化ポリシーで非推奨の RC4 暗号スイートに対応するようになりました。

今回の更新で、以下のユーティリティーの Rivest Cipher 4 (RC4) 暗号スイートを有効にする新しい **AD-SUPPORT** 暗号化サブポリシーが導入されました。

- System Security Services Daemon (SSSD)
- `adcli`
- `realmd`

管理者は、以下のシナリオで Active Directory (AD) が Advanced Encryption Standard (AES) を使用するように設定されていない場合に、新しい **AD-SUPPORT** サブポリシーをアクティブにすることができます。

- SSSD は、AD に直接接続された RHEL システムで使用されます。
- `adcli` は、AD ドメインに参加したり、ホストキーなどのホスト属性を更新するために使用されます。
- `realmd` は、AD ドメインに参加させるために使用されます。

Red Hat は、以下の条件のいずれかが当てはまる場合は、新しいサブポリシーを有効にすることを推奨します。

- AD のユーザーまたはサービスアカウントには RC4 暗号化キーがあり、AES 暗号化キーがありません。
- 個々の Active Directory ドメイン間の信頼リンクには RC4 暗号化鍵があり、AES 暗号化キーがありません。

DEFAULT 暗号化ポリシーに加えて **AD-SUPPORT** サブポリシーを有効にするには、以下を入力します。

■


```
# update-crypto-policies --set DEFAULT:AD-SUPPORT
```

(BZ#1866695)

authselect に新しい minimal プロファイル

authselect ユーティリティーには **minimal** プロファイルが新しくなりました。このプロファイルを使用すると、他の認証プロバイダーを使用する代わりに、ローカルユーザーおよびグループだけがシステムファイルから直接提供できます。したがって、**SSSD**、**winbind** パッケージ、および **fprintd** パッケージを安全に削除し、ディスクおよびメモリー領域の保存に最小限のインストールを必要とするシステムでこのプロファイルを使用できます。

(BZ#1654018)

SSSD が、パスワードをローテーションする際に Samba の **secrets.tdb** ファイルを更新するようになりました。

sssd.conf ファイルの新しい **ad_update_samba_machine_account_password** オプションが RHEL で利用できるようになりました。Samba の使用時にマシンのドメインパスワードをローテーションする際に、SSSD を使用して Samba の **secrets.tdb** ファイルを自動的に更新できます。

ただし、SELinux が Enforcing モードの場合、SSSD は **secrets.tdb** ファイルの更新に失敗します。そのため、Samba は新しいパスワードにアクセスできません。この問題を回避するには、SELinux を Permissive モードに設定します。

(BZ#1793727)

SSSD がデフォルトで AD GPO を強制するようになりました。

SSSD オプション **ad_gpo_access_control** のデフォルト設定が **Enforcing** になりました。RHEL 8 では、SSSD は、デフォルトで Active Directory Group Policy Objects (GPO) に基づいてアクセス制御ルールを強制します。

Red Hat は、RHEL 7 から RHEL 8 にアップグレードする前に、Active Directory で GPO が正しく設定されていることを確認することを推奨します。GPO を強制しない場合は、**/etc/sss/sss.conf** ファイルの **ad_gpo_access_control** オプションの値を **permissive** に変更します。

(JIRA:RHELPLAN-51289)

Directory Server が pwdReset 操作属性をサポート

今回の機能拡張で、Directory Server に **pwdReset** 操作属性のサポートが追加されました。管理者がユーザーのパスワードを変更すると、Directory Server は、ユーザーのエントリーの **pwdReset** を **true** に設定します。アプリケーションはこの属性を使用して、管理者がユーザーのパスワードをリセットしたかどうかを識別できます。

pwdReset は運用属性であるため、ユーザーはこれを編集できないことに注意してください。

(BZ#1775285)

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

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

- **wtime** の値は、操作が作業キューからワーカースレッドに移動するのにかかった時間を示します。

- **optime** の値は、ワーカーレッドが操作を開始した後に実際に操作が完了するのにかかった時間を示します。

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

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

(BZ#1850275)

5.1.14. デスクトップ

単一アプリケーションセッションが利用できるように

kiosk モード と呼ばれる 1 つのアプリケーションセッションで GNOME を起動できるようになりました。このセッションでは、GNOME は、設定したアプリケーションのフルスクリーンウィンドウのみを表示します。

単一アプリケーションセッションを有効にするには、以下を行います。

1. **gnome-session-kiosk-session** パッケージをインストールします。

```
# yum install gnome-session-kiosk-session
```

2. 単一アプリケーションセッションを開くユーザーの **\$HOME/.local/bin/redhat-kiosk** ファイルを作成して編集します。
ファイルに、起動するアプリケーションの実行ファイル名を入力します。

たとえば、**Text Editor** アプリケーションを起動するには、以下を実行します。

```
#!/bin/sh
gedit &
```

3. ファイルを実行可能にします。

```
$ chmod +x $HOME/.local/bin/redhat-kiosk
```

4. GNOME ログイン画面で、歯車ボタンメニューから **Kiosk** セッションを選択し、シングルアプリケーションユーザーとしてログインします。

(BZ#1739556)

TigerVNC がバージョン 1.10.1 にリベース

tigervnc スイートがバージョン 1.10.1 にリベースされました。この更新には、修正および改善点が多数含まれています。以下に例を示します。

- TigerVNC は、**systemd** サービスマネージャーを使用した仮想ネットワークコンピューティング (VNC) サーバーの起動のみをサポートするようになりました。
- ネイティブビューアーの **WinVNC**、および Xvnc/libvnc.so で完全な Unicode に対応するようになりました。

- ネイティブクライアントは、サーバー証明書を検証する際にシステムトラストストアを認識するようになりました。
- Java Web サーバーが削除されました。
- **X0vncserver** は、ローカル接続のみを許可するよう設定できるようになりました。
- **X0vncserver** は、ディスプレイの一部だけが共有されている場合に限り修正を受け取れていません。
- ポーリングは **WinVNC** でデフォルトになりました。
- VMware の VNC サーバーとの互換性が改善しました。
- macOS での一部の入力メソッドとの互換性が改善されました。
- JPEG artefacts の自動 repair が改善されました。

([BZ#1806992](#))

5.1.15. グラフィックインフラストラクチャー

新しいグラフィックカードのサポートが追加されました。

以下のグラフィックカードが完全にサポートされるようになりました。

- 以下のモデルを含む AMD Navi 14 ファミリー
 - Radeon RX 5300
 - Radeon RX 5300 XT
 - Radeon RX 5500
 - Radeon RX 5500 XT
- 以下のモデルを含む AMD Renoir APU ファミリー
 - Ryzen 3 4300U
 - Ryzen 5 4500U、4600U、および 4600H
 - Ryzen 7 4700U、4800U、および 4800H
- 以下のモデルを含む AMD Dali APU ファミリー:
 - Athlon Silver 3050U
 - Athlon Gold 3150U
 - Ryzen 3 3250U

さらに、以下のグラフィックドライバーが更新されました。

- Matrox **mgag200** ドライバー

([JIRA:RHELPLAN-55009](#))

ハードウェアアクセラレーション (Nvidia Volta および Turing)

nouveau グラフィックドライバーが、Nvidia Volta および Turing GPU ファミリーのハードウェアアクセラレーションをサポートするようになりました。その結果、3D グラフィックスを使用するデスクトップおよびアプリケーションが GPU で効率的にレンダリングされるようになりました。また、これにより他のタスクの CPU が解放され、システム全体の応答性が向上します。

(JIRA:RHELPLAN-57564)

XWayland でのディスプレイテアリングの減少

XWayland ディスプレイバックエンドは、XPresent 拡張を有効にします。XPresent を使用すると、アプリケーションはウィンドウコンテンツを効率的に更新できるため、表示表示が削減されます。

この機能は、3D エディターなど、フルスクリーンの OpenGL アプリケーションのユーザーインターフェイスのレンダリングを大幅に改善します。

(JIRA:RHELPLAN-57567)

Intel Tiger Lake GPU に対応

今回の更新で、GPU の Intel Tiger Lake ファミリーがサポートされるようになりました。これには、CPU モデルで見つかった Intel UHD Graphics および Intel Xe GPU が含まれます (<https://ark.intel.com/content/www/us/en/ark/products/codename/88759/tiger-lake.html>)。

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

この機能拡張は、[RHSA-2021:0558](#) 非同期アドバイザリーの一部としてリリースされています。

(BZ#1882620)

5.1.16. Web コンソール

Web コンソールセッション内からの権限の設定

Web コンソールには、ユーザーセッション内から管理アクセスと制限付きアクセスを切り替えるオプションがあります。Web コンソールセッションの **Administrative access** または **Limited access** をクリックして、モードを切り替えることができます。

(JIRA:RHELPLAN-42395)

ログ検索の改善

今回の更新により、Web コンソールでは、ユーザーがログ間で検索する方法に関する新しい方法をサポートする検索ボックスが導入されました。検索ボックスでは、ログメッセージの検索、サービスの指定、または特定のログフィールドによるエントリーの検索をサポートします。

(BZ#1710731)

概要ページでは、より多くの Insights レポートが表示されます。

今回の更新により、マシンが Red Hat Insights に接続されると、Web コンソールの **Overview** ページの **Health** カードに、ヒットの数とその優先度に関する詳細情報が表示されるようになりました。

(JIRA:RHELPLAN-42396)

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

RHEL システムロールに追加された ターミナルログロール

今回の機能拡張により、**rhel-system-roles** パッケージに同梱された RHEL システムロールに新しいターミナルログ (TLOG) ロールが追加されました。ユーザーは、**tlog** ロールを使用して、Ansible を使用してセッションの録画を設定し、設定できるようになりました。

現在、**tlog** ロールは次のタスクに対応します。

- **systemd** ジャーナルに記録データのログを記録するように **tlog** を設定する
- SSSD を使用して、明示的なユーザーおよびグループに対するセッション録画を有効にする

([BZ#1822158](#))

RHEL Logging システムロールが Ansible で利用可能になりました。

Logging システムロールを使用すると、ローカルおよびリモートホストにさまざまなロギング設定を一貫してデプロイできます。RHEL ホストをサーバーとして設定して、多くのクライアントシステムからログを収集できます。

([BZ#1677739](#))

rhel-system-roles-sap に完全対応

以前はテクノロジープレビューとして利用できた **rhel-system-roles-sap** パッケージが完全にサポートされるようになりました。SAP 向けの Red Hat Enterprise Linux (RHEL) システムロールを提供します。これを使用して、RHEL システムの設定を自動化して SAP ワークロードを実行できます。これらのロールは、関連する SAP ノート記載のベストプラクティスに基づいて最適な設定を自動的に適用することで、SAP ワークロードを実行するようにシステムを設定する時間を大幅に短縮できます。アクセスは、RHEL for SAP Solutions 製品に限定されます。サブスクリプションに関するサポートが必要な場合は、Red Hat カスタマーサポートまでご連絡ください。

rhel-system-roles-sap パッケージの以下の新しいロールが完全にサポートされています。

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

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

([BZ#1660832](#))

Ansible で RHEL システムロール metrics が利用可能

metric の RHEL システムロールを使用すると、ローカルおよびリモートホスト用に設定できます。

- **pcp** アプリケーションによるパフォーマンス分析サービス
- **grafana** サーバーを使用したこのデータの視覚化
- これらのサービスを手動で設定せずに、**redis** データソースを使用したこのデータのクエリー。

([BZ#1890499](#))

rhel-system-roles-sap がアップグレード

rhel-system-roles-sap パッケージがアップストリームバージョン 2.0.0 にアップグレードし、バグ修正および機能強化が複数追加されました。以下は、主な変更点です。

- ホスト名の設定とチェックを改善
- **uuid** ステータスの検出および処理を改善
- **--check(-c)** オプションのサポートを追加
- **nofile** 制限を 32800 から 65536 へ増加
- **nfs-utils** ファイルを **sap_preconfigure_packages*** に追加します。
- **firewalld** を無効にします。この変更により、**firewalld** は、インストールされている場合に限り無効にします。
- RHEL 8.0 および RHEL 8.1 の **setup** パッケージの最小バージョンを追加します。
- **tmpfiles.d/sap.conf** ファイル処理を改善
- 単一のステップの実行または SAP ノートのチェックをサポートします。
- 必要な **compat-sap-c++** パッケージの追加
- パッケージ最小インストール処理の改善
- RHEL システムロールの適用後に再起動が必要であるかどうかを検出する
- SELinux 状態の設定をサポートします。デフォルトの状態は、"**disabled**" です。
- 同じ IP アドレスを持つ複数の行がある場合に失敗しなくなりました。
- **sap_ip** を含む行が複数ある場合に **/etc/hosts** が変更されなくなりました。
- RHEL 7.7 での HANA のサポート
- **ppc64le** プラットフォームの SAP HANA に必要な IBM サービスおよび生産ツール用のリポジトリの追加をサポート

([BZ#1844190](#))

RHEL システムロール **storage** がファイルシステム管理に対応

今回の機能強化により、管理者は、**storage** RHEL システムロールを使用して以下を実行できるようになりました。

- **ext4** ファイルシステムのサイズ変更
- LVM ファイルのサイズ変更
- swap パーティションを作成します (存在する場合)。デフォルトのパラメーターを使用するブロックデバイスに swap パーティションを変更します。

([BZ#1959289](#))

5.1.18. 仮想化

互換性のない TSC 設定を持つホストへの仮想マシンの移行が速くなりました。

以前のリリースでは、TSC (Time Stamp Counter) 設定がないホストに仮想マシンを移行すると、プロセスでレイテンシーが発生していました。今回の更新により、移行の試行により、移行プロセスが開始する前にエラーが生成されるようになりました。

(JIRA:RHELPLAN-45950)

第2世代 AMD EPYC プロセッサの仮想化サポート

今回の更新で、RHEL 8 の仮想化で、第2世代 AMD EPYC プロセッサ (EPYC Rome としても知られる) のサポートが追加されました。これにより、RHEL 8 でホストされる仮想マシンは **EPYC-Rome** CPU モデルを使用し、プロセッサが提供する新機能を使用できるようになりました。

(JIRA:RHELPLAN-45959)

新しいコマンド: `virsh iothreadset`

今回の更新で、**`virsh iothreadset`** コマンドが追加されました。これは、動的な IOThread ポーリングを設定するのに使用できます。これにより、IOThread の CPU 消費よりも、レイテンシーが低くなるため、I/O 集約型のワークロードで仮想マシンを設定できるようになります。特定のオプションは、man ページの `virsh` を参照してください。

(JIRA:RHELPLAN-45958)

第10世代 Intel Core プロセッサでは、UMIP が KVM で対応

今回の更新で、第10世代 Intel Core プロセッサ (Ice Lake Server と呼ばれる) で実行されるホストの、User-mode Instruction Prevention (UMIP) 機能が、KVM で対応するようになりました。 **`sgdt`**、 **`sidt`**、 **`sldt`**、 **`smsw`**、 **`str`** などの特定の命令が、現在の Privilege Level (CPL) が 0 よりも大きい場合に、UMIP 機能は一般的な保護例外を発行します。その結果、UMIP は、承認されていないアプリケーションが、特権昇格攻撃を開始するために使用できる特定のシステム全体の設定にアクセスできないことで、システムセキュリティを確保します。

(JIRA:RHELPLAN-45957)

libvirt ライブラリーがメモリー帯域幅の割り当てに対応

libvirt が、メモリー帯域幅の割り当て (MBA) に対応するようになりました。MBA の場合、`<cputune>` セクションの `<memorytune>` 要素を使用して、vCPU スレッドのホストメモリー帯域幅の一部を割り当てることができます。

MBA は、Intel Xeon v4 プロセッサにある既存の Cache QoS Enforcement (CQE) 機能 (Broadwell サーバーとも呼ばれる) の拡張機能です。CPU アフィニティーに関連付けられたタスクの場合、MBA が使用するメカニズムは CQE と同じになります。

(JIRA:RHELPLAN-45956)

RHEL 6 仮想マシンが Q35 マシンタイプに対応

ゲスト OS として RHEL 6 を使用する RHEL 8 でホストされる仮想マシン (VM) が、より現代的な PCI Express ベースのマシンタイプである Q35 を使用できるようになりました。これにより、仮想デバイスの機能とパフォーマンスに様々な改善が行われ、最新の広範囲なデバイスで RHEL 6 VM への互換性が保証されます。

(JIRA:RHELPLAN-45952)

ログに記録されたすべての QEMU イベントにタイムスタンプが付けられました。これにより、`/var/log/libvirt/qemu/` ディレクトリーに保存されているログを使用して、仮想マシンをより簡単にトラブルシューティングできます。

QEMU ログに spice-server イベントのタイムスタンプが含まれる

今回の更新で、タイムスタンプが `spice-server` のイベントログに追加されます。そのため、ログに記録されたすべての QEMU イベントにはタイムスタンプが含まれるようになりました。これにより、`/var/log/libvirt/qemu/` ディレクトリーに保存されているログを使用して、仮想マシンをより簡単にトラブルシューティングできます。

(JIRA:RHELPLAN-45945)

bochs-display デバイスへの対応

RHEL 8.3 以降では、Bochs 表示デバイスが導入され、現在使用されている `stdvga` デバイスよりも安全です。**bochs-display** と互換性があるすべての仮想マシンはデフォルトでこれを使用することに注意してください。これには、主に UEFI インターフェイスを使用する仮想マシンが含まれます。

(JIRA:RHELPLAN-45939)

仮想マシンの MDS 保護の最適化

今回の更新で、RHEL 8 ホストは、[Microarchitectural Data Sampling](#) (MDS) に対して脆弱であるかどうかを仮想マシン (VM) に通知します。脆弱ではない仮想マシンは MDS に対する計測値を使用しません。これにより、パフォーマンスが向上します。

(JIRA:RHELPLAN-45937)

RBD での QCOW2 ディスクイメージの作成がサポートされるようになりました。

今回の更新により、RADOS Block Device (RBD) ストレージに QCOW2 ディスクイメージを作成できるようになりました。その結果、仮想マシンは、QCOW2 イメージでストレージバックエンドに RBD サーバーを使用することができます。

ただし、RBD ストレージ上の QCOW2 ディスクイメージの書き込みパフォーマンスは、現在予定よりも低いことに注意してください。

(JIRA:RHELPLAN-45936)

サポートされる VFIO デバイスの最大数が 64 に増加

今回の更新で、VFIO を使用する 64 PCI デバイスを RHEL 8 ホストの 1 台の仮想マシンに割り当てることができるようになりました。これは、RHEL 8.2 以前の 32 から増えています。

(JIRA:RHELPLAN-45930)

QEMU/KVM で discard コマンドと write-zeroes コマンドに対応

今回の更新で、`virtio-blk` の `discard` コマンドおよび `write-zeroes` コマンドが QEMU/KVM でサポートされるようになりました。したがって、仮想マシンは `virtio-blk` デバイスを使用して SSD の未使用のセクターを破棄し、割りがついているときにセクターをゼロで埋めます。これは、SSD のパフォーマンスを向上させる場合や、ドライブが安全に消去されていることを確認するために使用できます。

(JIRA:RHELPLAN-45926)

RHEL 8 が IBM POWER9 XIVE に対応

今回の更新で、IBM POWER9 の External Interrupt Virtualization Engine (XIVE) 機能のサポートが RHEL 8 に導入されています。これにより、IBM POWER9 システムの RHEL 8 ハイパーバイザーで実行している仮想マシンは、XIVE を使用して、I/O 集約型仮想マシンのパフォーマンスを向上できます。

(JIRA:RHELPLAN-45922)

仮想マシンの Control Group v2 に対応

今回の更新で、libvirt スイートがコントロールグループ v2 をサポートするようになりました。これにより、RHEL 8 でホストされる仮想マシンは、コントロールグループ v2 のリソース制御機能を利用できます。

(JIRA:RHELPLAN-45920)

準仮想化 IPI が Windows 仮想マシンで対応

今回の更新で、Windows 仮想マシン (VM) のサポートされるハイパーバイザー Enlightenment に `hv_ipi` フラグが追加されました。これにより、ハイパーコールを介してプロセッサ間割り込み (IPI) を送信できます。これにより、Windows OS を実行している仮想マシンでは、IPI を高速に実行することができます。

(JIRA:RHELPLAN-45918)

ディスクキャッシュが有効な仮想マシンの移行が可能になりました

今回の更新で、RHEL 8 KVM ハイパーバイザーが、ディスクキャッシュのライブマイグレーションと互換性があります。その結果、ディスクキャッシュが有効にされている仮想マシンのライブマイグレーションが可能になりました。

(JIRA:RHELPLAN-45916)

非特権セッションの仮想マシンで macvtap インターフェイスを使用できるようになりました。

特権プロセスで事前に作成した macvtap インターフェイスを仮想マシンで使用することができます。特に、`libvirtd` の非特権 `user` セッションで起動した仮想マシンが、macvtap インターフェイスを使用できるようにします。

そのためには、まず特権環境で macvtap インターフェイスを作成し、非特権セッションで `libvirtd` を実行するユーザーに所有されるように設定します。これは、Web コンソールなどの管理アプリケーションを使用するか、`root` としてコマンドラインユーティリティを使用して実行できます。以下に例を示します。

```
# ip link add link en2 name mymacvtap0 address 52:54:00:11:11:11 type macvtap mode bridge
# chown myuser /dev/tap$(cat /sys/class/net/mymacvtap0/ifindex)
# ip link set mymacvtap0 up
```

その後、仮想マシンの `<interface>` 設定のサブ要素 `<target>` を変更し、新たに作成された macvtap インターフェイスを参照します。

```
<interface type='ethernet'>
  <model type='virtio'>
    <mac address='52:54:00:11:11:11'>
      <target dev='mymacvtap0' managed='no'>
    </interface>
```

この設定では、`libvirtd` がユーザー `myuser` として実行すると、仮想マシンは起動時に既存の macvtap インターフェイスを使用します。

(JIRA:RHELPLAN-45915)

仮想マシンは、第 10 世代 Intel Core プロセッサの機能を使用できるようになりました。

Icelake-Server および **Icelake-Client** CPU モデル名が仮想マシンで利用可能になりました。第 10 世代 Intel Core プロセッサを持つホストで、**Icelake-Server** または **Icelake-Client** を仮想マシンの XML 設定の CPU タイプとして使用すると、これらの CPU の新機能が仮想マシンに公開されます。

(JIRA:RHELPLAN-45911)

QEMU が LUKS 暗号化に対応

今回の更新で、LUKS (Linux Unified Key Setup) 暗号化を使用して仮想ディスクを作成できるようになりました。仮想マシンの XML 設定に **<encryption>** フィールドを追加して、ストレージボリュームの作成時にディスクを暗号化できます。また、XML 設定ファイルにディスクのドメイン定義に **<encryption>** フィールドを追加して、**LUKS** 暗号化仮想ディスクを完全に仮想マシンに透過的にすることもできます。

(JIRA:RHELPLAN-45910)

nbdkit のログの改善

nbdkit サービスロギングが冗長化されるように変更されました。その結果、**nbdkit** は重要なメッセージのみをログに記録し、**virt-v2v** 変換中に作成されたログは短くなり、解析が容易になりました。

(JIRA:RHELPLAN-45909)

仮想マシンの SELinux セキュリティーラベルおよびパーミッションの一貫性を強化

今回の更新で、**libvirt** サービスは、ファイルに関連する SELinux セキュリティーラベルとパーミッションを記録し、ファイルの変更後にラベルを復元できるようになりました。したがって、たとえば **libguestfs** ユーティリティーを使用して、特定のユーザーが所有している仮想マシン (VM) のディスクイメージを変更しても、イメージの所有者が **root** に変更されなくなりました。

この機能は、NFS などの拡張ファイル属性に対応していないファイルシステムでは機能しないことに注意してください。

(JIRA:RHELPLAN-45908)

QEMU が XTS 暗号の **gcrypt** ライブラリーを使用するようになりました。

今回の更新で、QEMU エミュレーターが、**gcrypt** ライブラリーが提供する XTS 暗号モード実装を使用するように変更になりました。これにより、ホストストレージが QEMU のネイティブ **luks** 暗号化ドライバーを使用する仮想マシンの I/O パフォーマンスが改善されます。

(JIRA:RHELPLAN-45904)

Windows Virtio ドライバーが Windows の更新を使用して更新可能に

今回の更新で、QEMU の起動時に、新しい仕様 **SMBIOS** 文字列がデフォルトで開始されるようになりました。**SMBIOS** フィールドで提供されるパラメーターにより、仮想マシンで実行している仮想ハードウェアの ID を生成できます。これにより、Windows Update は仮想ハードウェアと RHEL ハイパーバイザーマシンタイプを特定し、Windows 10+、Windows Server 2016、および Windows Server 2019+ を実行している仮想マシンで Virtio ドライバーを更新できます。

(JIRA:RHELPLAN-45901)

新しいコマンド: **virsh guestinfo**

virsh guestinfo コマンドが RHEL 8.3 に導入されました。これにより、仮想マシンに関する以下のタイプの情報を報告できます。

- ゲスト OS およびファイルシステム情報

- アクティブなユーザー
- 使用されるタイムゾーン

virsh guestinfo を実行する前に、**qemu-guest-agent** パッケージがインストールされていることを確認します。また、以下のように、仮想マシンの XML 設定で **guest_agent** チャンネルを有効にする必要があります。

```
<channel type='unix'>
  <target type='virtio' name='org.qemu.guest_agent.0'>
</channel>
```

(JIRA:RHELPLAN-45900)

BFLOAT16 入力の VNNI が KVM で対応する

今回の更新で、**BFLOAT16** 入力に対応する Vector Neural Network Instructions (VNNI) は、**AVX512_BF16** 命令とも呼ばれる、KVM が 3rd Gen Intel Xeon スケーラブルプロセッサ (Cooper Lake と呼ばれる) で実行されるホストの KVM でサポートされるようになりました。その結果、ゲストソフトウェアは、仮想 CPU 設定で有効にすることで、仮想マシン内の **AVX512_BF16** 命令を使用できるようになりました。

(JIRA:RHELPLAN-45899)

新しいコマンド: virsh pool-capabilities

RHEL 8.3 には、**virsh pool-capabilities** コマンドオプションが追加されました。このコマンドは、ホスト上のストレージプールおよび各プール内のストレージボリュームの作成に使用できる情報を表示します。これには以下が含まれます。

- ストレージプールの種類
- ストレージプールのソース形式
- ターゲットストレージボリュームの形式タイプ

(JIRA:RHELPLAN-45884)

Intel Xeon Platinum 9200 シリーズプロセッサを持つ仮想マシンで CPUID.1F のサポート

今回の更新で、RHEL 8 でホストされる仮想マシンは、拡張 Topology Enumeration リーフ機能 (CPUID.1F) を使用して、複数のロールの仮想 CPU トポロジーで設定できるようになりました。この機能は、以前は Cascade Lake として知られる Intel Xeon Platinum 9200 シリーズプロセッサでサポートされます。これにより、Intel Xeon Platinum 9200 シリーズプロセッサを使用するホストで、ホストの物理 CPU トポロジーをミラーリングする vCPU トポロジーを作成できるようになりました。

(JIRA:RHELPLAN-37573、JIRA:RHELPLAN-45934)

仮想マシンが第 3 世代 Intel Xeon Scalable プロセッサの機能を使用可能に

Cooperlake CPU モデル名が仮想マシン (VM) で利用可能になりました。**Cooperlake** を仮想マシンの XML 設定の CPU タイプとして使用すると、ホストがこの CPU を使用する場合は、仮想マシンに公開される 3 世代 Intel Xeon Scalable Processor の新機能が作成されます。

(JIRA:RHELPLAN-37570)

KVM で Intel Optane 永続メモリーに対応

今回の更新で、RHEL 8 でホストされる仮想マシンは、Intel Crystal Ridge として知られる Intel Optane 永続メモリー技術の利点を活用できるようになりました。Intel Optane 永続メモリーストレージデバイスは、データセンタークラスの永続メモリー技術を提供し、トランザクションのスループットを大幅に向上させます。

(JIRA:RHELPLAN-14068)

仮想マシンで Intel プロセッサトレースを使用可能に

今回の更新で、RHEL 8 でホストされる仮想マシン (VM) が Intel プロセッサトレース (PT) 機能を使用できるようになりました。ホストが Intel PT をサポートする CPU を使用する場合は、特殊な Intel ソフトウェアを使用して、仮想マシンの CPU のパフォーマンスに関するさまざまなメトリックスを収集できます。また、仮想マシンの XML 設定で **intel-pt** 機能も有効にする必要があります。

(JIRA:RHELPLAN-7788)

DASD デバイスが IBM Z の仮想マシンに割り当てられる

DASD (ダイレクトアクセスストレージデバイス) は、特定のストレージ機能を提供します。この **vfio-ccw** 機能を使用すると、DASD を仲介デバイスとして IBM Z ホストの仮想マシンに割り当てることができます。たとえば、仮想マシンは z/OS データセットにアクセスするか、割り当てられた DASD を z/OS マシンと共有できます。

(JIRA:RHELPLAN-40234)

IBM Z でサポートされる IBM Secure Execution

IBM Z ハードウェアを使用して RHEL 8 ホストを実行する場合は、仮想マシンの IBM Secure Execution を設定して、仮想マシンのセキュリティーを強化できます。IBM Secure Execution (Protected Virtualization と呼ばれる) は、ホストシステムが仮想マシンの状態とメモリーのコンテンツにアクセスできないようにします。

その結果、ホストが危険にさらされても、ゲストオペレーティングシステムを攻撃するベクトルとして使用できません。さらに、セキュア実行を使用して、信頼できないホストが仮想マシンから機密情報を取得しないようにすることもできます。

(JIRA:RHELPLAN-14754)

5.1.19. クラウド環境の RHEL

cloud-utils-growpart が 0.31 にリベース

cloud-utils-growpart パッケージがバージョン 0.31 にアップグレードされ、バグ修正および機能強化が複数追加されました。以下は、主な変更点です。

- GPT ディスクが 2TB を超えるのを妨げるバグが修正されました。
- **growpart** 操作は、開始セクターとサイズが同じ場合に失敗しなくなりました。
- 以前は **sgdisk** ユーティリティーを使用してパーティションのサイズを変更できない場合があります。この問題は修正されています。

(BZ#1846246)

5.1.20. コンテナ

skopeo コンテナイメージが利用可能に

registry.redhat.io/rhel8/skopeo コンテナイメージは、**skopeo** パッケージをコンテナ化した実装です。**skopeo** ツールは、コンテナイメージおよびイメージリポジトリでさまざまな操作を実行するコマンドラインユーティリティです。このコンテナイメージを使用すると、レジストリー内のコンテナイメージを検査し、レジストリーからコンテナイメージを削除し、認証されていないあるコンテナレジストリーから別のコンテナレジストリーにコンテナイメージをコピーすることができます。**registry.redhat.io/rhel8/skopeo** コンテナイメージをプルするには、アクティブな Red Hat Enterprise Linux サブスクリプションが必要です。

(BZ#1627900)

buildah コンテナイメージが利用可能に

registry.redhat.io/rhel8/buildah コンテナイメージは、**buildah** パッケージをコンテナ化した実装です。**buildah** ツールは、OCI コンテナイメージの構築を容易にします。このコンテナイメージを使用すると、システムに **buildah** パッケージをインストールしなくても、コンテナイメージをビルドできます。このユースケースでは、root 以外のユーザーとして rootless モードでこのイメージを実行することを説明しません。**registry.redhat.io/rhel8/buildah** コンテナイメージをプルするには、アクティブな Red Hat Enterprise Linux サブスクリプションが必要です。

(BZ#1627898)

Podman v2.0 RESTful API が利用可能に

varlink ライブラリーに基づいて古いリモート API に代わる、新しい REST ベースの Podman 2.0 API。新規 API はルートフル環境およびルートレス環境の両方で機能し、docker 互換レイヤーを提供します。

(JIRA:RHELPLAN-37517)

Podman のインストールには、**container-selinux** は必要ありません。

今回の機能拡張により、コンテナビルド時に **container-selinux** パッケージのインストールはオプションになりました。その結果、Podman に他のパッケージの依存関係が少なくなる

(BZ#1806044)

5.2. 外部カーネルパラメーターの重要な変更

本章では、システム管理者向けに、Red Hat Enterprise Linux 8.3 に同梱されるカーネルの重要な変更点の概要を説明します。変更には、たとえば、**proc** エントリー、**sysctl** および **sysfs** のデフォルト値、ブートパラメーター、カーネル設定オプション、または重要な動作の変更などが含まれます。

新しいカーネルパラメーター

acpi_no_watchdog = [HW,ACPI,WDT]

このパラメーターを使用すると、Advanced Configuration and Power Interface (ACPI) ベースのウォッチドッグインターフェイス (WDAT) を無視し、ネイティブドライバーが代わりにウォッチドッグデバイスを制御できるようにします。

dfitcc = [HW,S390]

このパラメーターは、IBM Z アーキテクチャーの **zlib** ハードウェアサポートを設定します。

形式: { on | off | def_only | inf_only | always }

オプションは次のとおりです。

- **on** (デフォルト): レベル 1 の圧縮および圧縮解除の IBM Z **zlib** ハードウェアサポート

- **off** - IBM Z **zlib** ハードウェアのサポートはありません。
- **def_only: deflate** アルゴリズムの IBM Z **zlib** ハードウェアサポート (レベル1の圧縮)
- **inf_only: inflate** アルゴリズムの IBM Z **zlib** ハードウェアサポート (デプロイメント)
- **always: on** と似ていますが、選択した圧縮レベルを無視し、常にハードウェアサポートを使用します (デバッグに使用)。

irqchip.gicv3_pseudo_nmi = [ARM64]

このパラメーターは、カーネル内の擬似マスク不可割り込み (NMI) のサポートを有効にします。このパラメーターを使用するには、**CONFIG_ARM64_PSEDO_NMI** 設定項目でカーネルを構築する必要があります。

panic_on_taint =

add_taint() で、条件付きで **panic()** を呼び出すビットマスク。
形式: <hex>[,nouserntaint]

TAINT フラグのセットのフラグで **add_taint()** システムコールが呼び出される際に、このフラグのセットを表す 16 進数のビットマスク。オプションの **nouserntaint** スイッチは、**/proc/sys/kernel/tainted** ファイルに書き込むことでユーザー空間の強制を阻止します。これは、**panic_on_taint** のビットマスクに一致するフラグセットに通知します。

詳細は、[アップストリームのドキュメント](#) を参照してください。

prot_virt = [S390]

形式: <bool>

このパラメーターにより、ハードウェアサポートが存在する場合には、ハイパーバイザーから分離されている保護されている仮想マシンをホストできます。

rcutree.use_softirq = [KNL]

このパラメーターは、Tree-RCU の **softirq** 処理の排除を有効にします。このパラメーターをゼロに設定すると、すべての **RCU_SOFTIRQ** 処理を per-CPU rcuc kthreads に移動します。**rcutree.use_softirq** をゼロ以外の値 (デフォルト) に設定すると、**RCU_SOFTIRQ** がデフォルトで使用されます。**rcutree.use_softirq=0** を指定して rcuc kthreads を使用します。

split_lock_detect = [X86]

このパラメーターにより、分割されたロック検出が有効になります。有効化されていると、ハードウェアサポートが有効な場合は、キャッシュライン境界でデータにアクセスするアトミックな命令により、チェック例外が調整されます。オプションは次のとおりです。

- **off** - 有効ではありません
- **warn**: カーネルは、Alignment Check Exception (#AC) をトリガーするアプリケーションについてレート制限警告を生成します。このモードは、分割されたロック検出をサポートする CPU のデフォルトです。
- **fatal**: カーネルは、#AC 例外をトリガーするアプリケーションに Buss エラー (SIGBUS) シグナルを送信します。
#AC 例外がユーザーモードで実行されていない場合、カーネルは **warn** または **fatal** モードのいずれかで oops エラーを発行します。

srbds = [X86,INTEL]

このパラメーターは、特殊レジスタバッファデータサンプリング (SRBDS) の軽減策を制御します。

特定の CPU は、マイクロアーキテクチャーデータサンプリング (MDS) のような不正使用に対して脆弱です。このような不正使用が発生すると、乱数ジェネレーターからビットが漏洩する可能性があります。

デフォルトでは、マイクロコードはこの問題を軽減します。ただし、マイクロコードの修正により、**RDRAND** および **RDSEED** 命令の処理が大幅に遅くなる可能性があります。そのため、これにより、**urandom** カーネル乱数のソースデバイスからのスループットが低下します。

マイクロコードの軽減策を無効にするには、以下のオプションを設定します。

- **off** - 軽減策を無効にし、**RDRAND** および **RDSEED** のパフォーマンスへの影響を排除します。

svm = [PPC]

形式: { on | off | y | n | 1 | 0 }

このパラメーターは、pSeries システムでの Protected Execution Facility の使用を制御します。

nopv = [X86,XEN,KVM,HYPER_V,VMWARE]

このパラメーターは、PV ドライバーなしでゲストを汎用ゲストとして実行するように強制する PV の最適化を無効にします。

現在サポートされているのは、XEN HVM、KVM、HYPER_V、および VMWARE ゲストです。

更新されたカーネルパラメーター**hugepagesz = [HW]**

このパラメーターは、Huge Page サイズを指定します。このパラメーターを **hugepages** パラメーターと共に使用して、指定されたサイズの Huge Page 数を事前に割り当てます。

以下のようにペアに **hugepagesz** パラメーターと **hugepages** パラメーターを指定します。

```
hugepagesz=2M hugepages=512
```

hugepagesz パラメーターは、特定の Huge Page サイズのコマンドラインで1回だけ指定できません。有効な Huge Page サイズはアーキテクチャーに依存します。

hugepages = [HW]

このパラメーターは、事前に割り当てる Huge Page の数を指定します。このパラメーターは、通常、有効な **hugepagesz** パラメーターまたは **default_hugepagesz** パラメーターに従います。ただし、ヒュージページが最初に、または HugeTLB コマンドラインパラメーターのみの場合、割り当てるデフォルトサイズの **hugepages** の数を暗黙的に指定します。デフォルトサイズの Huge Page の数が暗黙的に指定されると、デフォルトサイズの **hugepagesz + hugepages** パラメーターのペアで上書きすることはできません。

たとえば、2M のデフォルトの Huge Page サイズを持つアーキテクチャーの場合は、以下のようになります。

```
hugepages=256 hugepagesz=2M hugepages=512
```

上記の例からの設定により、256 2M の Huge Page の割り当てと、**hugepages=512** パラメーターが無視される警告メッセージが表示されます。**hugepages** が無効な **hugepagesz** の前に付けられると、**hugepages** は無視されます。

default_hugepagesz = [HW]

このパラメーターは、デフォルトの Huge Page サイズを指定します。**default_hugepagesz** は、コマンドラインで1回のみ指定できます。オプションとして、**default_hugepagesz** を **hugepages** パラメーターに従って実行し、デフォルトサイズの特定の Huge Page を事前割り当てできます。また、事前に割り当てるために、デフォルトサイズの Huge Page の数を暗黙的に指定することもできます。

たとえば、2M のデフォルトの Huge Page サイズを持つアーキテクチャーの場合は、以下のようになります。

```
hugepages=256
default_hugepagesz=2M hugepages=256
hugepages=256 default_hugepagesz=2M
```

上記の例からの設定により、256 2M の Huge Page が割り当てられます。有効なデフォルトの Huge Page サイズはアーキテクチャーに依存します。

efi = [EFI]

形式: { "old_map", "nochunk", "noruntime", "debug", "nosoftreserve" }

オプションは次のとおりです。

- **old_map** [X86-64] - 古い ioremap ベースの EFI ランタイムサービスへのマッピングに切り替えます。32-bit は依然として、デフォルトでこれを使用します。
- **nochunk**: EFI ブートスタブの chunks 内のファイルの読み取りを無効にします。チャンクすると一部のファームウェアの実装で問題が発生する可能性があります。
- **noruntime** - EFI ランタイムサービスのサポートの無効化
- **debug** - その他のデバッグ出力を有効にする
- **nosoftreserve**: **EFI_MEMORY_SP** (Specific Purpose) 属性により、カーネルがメモリーマッピングドライバーを要求に予約することがあります。**efi=nosoftreserve** を指定してこの予約を無効にし、ベースタイプ (例:**EFI_CONVENTIONAL_MEMORY** / "System RAM") でメモリーを処理します。

intel_iommu = [DMAR]

Intel IOMMU ドライバー Direct Memory Access Remapping (DMAR)。

追加されたオプションは以下のとおりです。

- **nobounce** (デフォルト off)- Thunderbolt デバイスなどの信頼できないデバイスのバウンズバッファーを無効にします。これにより、信頼できないデバイスを信頼できるデバイスとして扱います。したがって、この設定により、ダイレクトメモリーアクセス (DMA) 攻撃のセキュリティリスクにさらされる可能性があります。

mem = nn[KMG] [KNL,BOOT]

このパラメーターにより、特定のメモリー量の使用が強制されます。

以下のように、使用するメモリー量。

1. テスト用です。

2. カーネルがシステムメモリー全体を表示できない場合
3. **mem** 境界を超えたメモリーはハイパーバイザーから除外されてから、KVM ゲストに割り当てられます。

[X86] 最大アドレスの制限として機能します。物理アドレス領域の競合を避けるために **memmap** パラメーターと併用します。 **memmap** を使用しない場合には、PCI (Peripheral Component Interconnect) デバイスが未使用の RAM に属するアドレスに配置できました。

この設定は、上記の場合 3 以降、ブート時の場合にのみ有効であることに注意してください。これは、ハイパーバイザーのシステムメモリーが十分でない場合に、ブート後にメモリーを追加する必要がある場合があります。

pci = [PCI]

さまざまな Peripheral Component Interconnect (PCI) サブシステムオプション。

ここでの一部のオプションは、特定のデバイスまたはデバイスセット (<pci_dev>) 上で動作します。これは、以下のいずれかの形式で指定されます。

```
[<domain>:]<bus>:<dev>.<func>[/<dev>.<func>]*
pci:<vendor>:<device>[:<subvendor>:<subdevice>]
```

最初の形式は、PCI バス/デバイス/機能アドレスを指定して、新しいハードウェアが挿入される場合、マザーボードファームウェアが変更したり、他のカーネルパラメーターによって生じることが原因で変更する可能性があることに注意してください。ドメインを指定しないと、ゼロになります。オプションで、ベースアドレスの後に、複数のデバイス/機能アドレス経由のデバイスへのパスを指定できます (これは、再番号の問題に対して堅牢です)。2 つ目の形式は、設定領域から ID を使用してデバイスを選択します。これは、システム内の複数のデバイスに一致する可能性があります。

オプションは次のとおりです。

- **hpmiosize**: ホットプラグブリッジのメモリーマッピング I/O (MMIO) ウィンドウ用に予約されるバス領域の固定量。デフォルトサイズは 2 メガバイトです。
- **hpmmioprefsize**: ホットプラグブリッジの MMIO_PREF ウィンドウ用に予約されるバス領域の固定量。デフォルトサイズは 2 メガバイトです。

pcie_ports = [PCIE]

PCIe (peripheral Component Interconnect Express) ポートサービスを処理する。

オプションは次のとおりです。

- **native**: プラットフォームが OS ホットプラグを付与していない場合でも、ネイティブの PCIe サービス (PME、AER、DPC、PCIPC、PCIPC) を使用します。この設定は、プラットフォームがこれらのサービスも使用しようとする、競合が発生する可能性があります。
- **DPC-native** - DPC にネイティブ PCIe サービスのみを使用します。この設定では、ファームウェアが AER または DPC を使用する場合は競合が発生する可能性があります。
- **compat**: ネイティブ PCIe サービス (PME、AER、DPC、PCIPC ホットプラグ) を無効にします。

rcu_nocbs = [KNL]

引数は CPU リストです。文字列 all を使用して、システム上のすべての CPU を指定できます。

usbcore.authorized_default = [USB]

デフォルトの USB デバイス認証。
オプションは次のとおりです。

- **-1** (デフォルト) - ワイヤレス USB を除く認証
- **0** - 非承認
- **1** - 承認
- **2** - デバイスが内部ポートに接続されている場合に承認される

`usbcore.old_scheme_first = [USB]`

このパラメーターにより、古いデバイスの初期化スキームを開始できます。この設定は、低スピードデバイスおよびフルスピードデバイス (デフォルトは 0 = off) のみに適用されます。

`usbcore.quirks = [USB]`

組み込み USB コア quirk リストを拡張する quirk エントリーのリスト。リストエントリーはコンマで区切られます。各エントリーには **VendorID:ProductID:Flags** の形式 (例: **quirks=0781:5580:bk,0a5c:5834:gij**) があります。ID は 4 桁の 16 進数で、**フラグ** は文字のセットです。各文字は、組み込みの quirk を変更します。クレデンシャルが設定されると、明確で消去された場合は設定を行います。

追加されたフラグ:

- **o: USB_QUIRK_HUB_SLOW_RESET** (ポートリセット後に、ハブで遅延が必要)

新しい `/proc/sys/fs` パラメーター

`protected_fifos`

このパラメーターは Openwall ソフトウェアの制限をベースとしており、プログラムが通常のファイルの作成を目的とする攻撃者が制御する FIFO への意図しない書き込みを回避するための保護を提供します。

オプションは次のとおりです。

- **0**: FIFO への書き込みは無制限です。
- **1**: ディレクトリーの所有者によって所有されていない限り、すべてのユーザーが書き込み可能なスティッキーディレクトリーの FIFO で **O_CREAT** フラグが開かないようにします。
- **2**: グループの書き込み可能なスティッキーディレクトリーに適用されます。

`protected_regular`

このパラメーターは `protected_fifos` パラメーターと似ていますが、プログラムが作成予定の攻撃者が制御する通常のファイルに書き込むのを防ぎます。

オプションは次のとおりです。

- **0**: 通常のファイルへの書き込みは無制限です。
- **1**: ディレクトリーの所有者によって所有されていない限り、すべてのユーザーが書き込み可能なスティッキーディレクトリーの通常のファイルで **O_CREAT** フラグが開かないようにします。
- **2**: グループの書き込み可能なスティッキーディレクトリーに適用されます。

5.3. デバイスドライバー

5.3.1. 新しいドライバー

ネットワークドライバー

- Kvaser CAN/USB デバイス向け CAN ドライバー (kvaser_usb.ko.xz)
- Theobroma Systems UCAN デバイスのドライバー (ucan.ko.xz)
- Pensando Ethernet NIC Driver (ionic.ko.xz)

グラフィックドライバーとその他のドライバー

- Generic Remote Processor Framework (remoteproc.ko.xz)
- Package Level C-state Idle Injection for Intel® CPUs (intel_powerclamp.ko.xz)
- X86 PKG TEMP Thermal Driver (x86_pkg_temp_thermal.ko.xz)
- INT3402 Thermal driver (int3402_thermal.ko.xz)
- ACPI INT3403 サーマルドライバー (int3403_thermal.ko.xz)
- Intel® acpi thermal rel misc dev driver (acpi_thermal_rel.ko.xz)
- INT3400 Thermal driver (int3400_thermal.ko.xz)
- Intel® INT340x common thermal zone handler (int340x_thermal_zone.ko.xz)
- Processor Thermal Reporting Device Driver (processor_thermal_device.ko.xz)
- Intel® PCH Thermal ドライバー (intel_pch_thermal.ko.xz)
- DRM gem ttm ヘルパー (drm_ttm_helper.ko.xz)
- cec ドライバーのデバイスノード登録 (cec.ko.xz)
- Fairchild FUSB302 Type-C Chip Driver (fusb302.ko.xz)
- VHOST IOTLB (vhost_iotlb.ko.xz)
- virtio 向け vDPA ベースの vhost バックエンド (vhost_vdpa.ko.xz)
- VMware 仮想 PTP クロックドライバー (ptp_vmw.ko.xz)
- Intel® LPSS PCI ドライバー (intel-lpss-pci.ko.xz)
- Intel® LPSS コアドライバー (intel-lpss.ko.xz)
- Intel® LPSS ACPI ドライバー (intel-lpss-acpi.ko.xz)
- Mellanox ウォッチドッグドライバー (mlx_wdt.ko.xz)
- Mellanox FAN ドライバー (mlxreg-fan.ko.xz)
- Mellanox regmap I/O アクセスドライバー (mlxreg-io.ko.xz)

- Intel® speed select interface pci mailbox driver (isst_if_mbox_pci.ko.xz)
- Intel® スピード選択インターフェイス mailbox ドライバー (isst_if_mbox_msr.ko.xz)
- Intel® speed select interface mmio driver (isst_if_mmio.ko.xz)
- Mellanox LED regmap ドライバー (leds-mlxreg.ko.xz)
- vDPA Device Simulator (vdpa_sim.ko.xz)
- Intel® Tiger Lake PCH pinctrl/GPIO ドライバー (pinctrl-tigerlake.ko.xz)
- PXA2xx SSP SPI Controller (spi-pxa2xx-platform.ko.xz)
- PXA のドライバーの CE4100/LPSS PCI-SPI glue コード (spi-pxa2xx-pci.ko.xz)
- Hyper-V PCI Interface (pci-hyperv-intf.ko.xz)
- virtio デバイス向け vDPA バスドライバー (virtio_vdpa.ko.xz)

5.3.2. 更新されたドライバー

ネットワークドライバーの更新

- VMware vmxnet3 virtual NIC ドライバー (vmxnet3.ko.xz) がバージョン 1.5.0.0-k に更新されました。
- realtek RTL8152/RTL8153 Based USB Ethernet Adapters (r8152.ko.xz) がバージョン 1.09.10 に更新されました。
- Broadcom BCM573xx ネットワークドライバー (bnxt_en.ko.xz) がバージョン 1.10.1. に更新されました。
- Netronome Flow Processor (NFP) ドライバー (nfp.ko.xz) がバージョン 4.18.0-240.el8.x86_64 に更新されました。
- Intel® Ethernet Switch Host Interface Driver (fm10k.ko.xz) がバージョン 0.27.1-k に更新されました。
- Intel® Ethernet Connection E800 Series Linux Driver (ice.ko.xz) がバージョン 0.8.2-k に更新されました。

ストレージドライバーの更新

- Emulex LightPulse Fibre Channel SCSI ドライバー (lpfc.ko.xz) がバージョン 0:12.8.0.1 に更新されました。
- QLogic FCoE ドライバー (bnx2fc.ko.xz) がバージョン 2.12.13 に更新されました。
- LSI MPT Fusion SAS 3.0 Device Driver (mpt3sas.ko.xz) がバージョン 34.100.00.00 に更新されました。
- Driver for HP Smart Array Controller バージョン (hpsa.ko.xz) がバージョン 3.4.20-170-RH5 に更新されました。
- QLogic Fibre Channel HBA ドライバー (qla2xxx.ko.xz) がバージョン 10.01.00.25.08.3-k に更新されました。

- Broadcom MegaRAID SAS ドライバー (megaraid_sas.ko.xz) がバージョン 07.714.04.00-rh1 に更新されました。

グラフィックおよびその他ドライバーの更新

- VMware SVGA デバイスのスタンドアロンの drm ドライバー (vmwgfx.ko.xz) がバージョン 2.17.0.0 に更新されました。
- Chelsio Terminator カードの暗号化 Co-processor (chcr.ko.xz) がバージョン 1.0.0.0-ko に更新されました。

5.4. バグ修正

本パートでは、ユーザーに大きな影響を及ぼしていた Red Hat Enterprise Linux 8.3 のバグで修正されたものを説明します。

5.4.1. インストーラーおよびイメージの作成

RHEL 8 初期セットアップが SSH で適切に動作するように

以前は、SSH を使用してシステムにログインすると、RHEL 8 の初期セットアップインターフェイスが表示されませんでした。これにより、SSH を介して管理される RHEL 8 マシンの初期設定を実行できません。この問題は修正され、SSH を介して実行された場合に RHEL 8 の初期設定が正しく機能するようになりました。

([BZ#1676439](#))

reboot --kexec コマンドを使用するとインストールが失敗しました。

以前では、**reboot --kexec** コマンドを含むキックスタートファイルを使用すると、RHEL 8 のインストールに失敗していました。

今回の更新で、**reboot --kexec** を使用したインストールが予想通りに機能するようになりました。

([BZ#1672405](#))

America/New York タイムゾーンを正しく設定可能

以前では、インタラクティブな Anaconda インストールプロセスでは、キックスタートファイルの使用時に **America/New York** タイムゾーンをユーザーが設定できませんでした。今回の更新で、タイムゾーンがキックスタートファイルで指定されていない場合に、インタラクティブなインストーラーの希望のタイムゾーンとして **America/New York** を設定できるようになりました。

([BZ#1665428](#))

SELinux コンテキストが正しく設定されるようになる

以前では、SELinux が Enforcing モードの場合、一部のフォルダーおよびファイルの誤った SELinux コンテキストにより、インストール後にこれらのファイルにアクセスしようとする際に、予期しない AVC 拒否が発生していました。

今回の更新で、Anaconda は正しい SELinux コンテキストを設定します。これにより、ファイルシステムを手動で再ラベル付けせずにフォルダーおよびファイルにアクセスできるようになりました。

([BZ#1775975](#))

自動パーティション作成が有効な /boot パーティションを作成

以前は、自動パーティション設定を使用して、または事前設定されたパーティションを持つキックスタートファイルを使用してシステムに RHEL をインストールすると、インストーラーは無効な **/boot** パーティションを含むパーティション設定スキームを作成していました。したがって、パーティション設定スキームの検証に失敗していたため、自動インストールプロセスが途中で終了しました。今回の更新で、Anaconda が、有効な **/boot** パーティションを含むパーティション設定スキームを作成するようになりました。その結果、自動インストールは期待どおりに完了します。

(BZ#1630299)

Binary DVD ISO イメージを使用した GUI インストールが、CDN の登録なしで正常に完了するようになりました。

以前のバージョンでは、Binary DVD ISO イメージファイルを使用して GUI インストールを実行する場合は、Red Hat 機能への接続機能を使用してシステムを登録するまで、インストーラーの競合状態によりインストールが続行できなくなることがあります。

今回の更新で、Red Hat への接続機能を使用してシステムを登録しなくてもインストールに進むことができるようになりました。

(BZ#1823578)

キックスタートで iSCSI デバイスまたは FCoE デバイスを作成し、`ignoredisk --only-use` コマンドでインストールプロセスを停止しなくなりました。

以前では、キックスタートで作成された iSCSI デバイスまたは FCoE デバイスが `ignoredisk --only-use` コマンドで使用されると、インストールプログラムは、**Disk "disk/by-id/scsi-360a9800042566643352b476d674a774a"** で指定されている error のようなエラーを出して失敗しました。これによりインストールプロセスが停止しました。

今回の更新で、この問題が修正されています。インストールプログラムは機能を続行します。

(BZ#1644662)

CDN を使用したシステム登録に失敗し、エラーメッセージ `Name or service not known` が表示

コンテンツ配信ネットワーク (CDN) を使用してシステムを登録しようとする、登録プロセスが失敗し、`Name or service not known` というエラーメッセージが表示されていました。

この問題は、空のカスタムサーバー URL とカスタムベース URL の値がシステム登録のデフォルト値をオーバーライドするために発生しました。

今回の更新により、空の値はデフォルト値を上書きせず、システム登録が正常に完了するようになりました。

(BZ#1862116)

5.4.2. ソフトウェア管理

dnf-automatic が、正しい GPG 署名を持つパッケージのみを更新

以前は、`dnf-automatic` 設定ファイルは、更新を実行する前にダウンロードしたパッケージの GPG 署名を確認しませんでした。そのため、リポジトリ設定に GPG 署名の確認 (`gpgcheck=1`) が必要ですが、インポートされていない鍵で署名されていない更新または更新は、`dnf-automatic` でインストールできませんでした。今回の更新で問題が修正され、`dnf-automatic` は、更新を実行する前にダウンロードしたパッケージの GPG 署名をチェックするようになりました。その結果、正しい GPG 署名を使用した更新は、GPG 署名の確認が必要なリポジトリからのみインストールされます。

(BZ#1793298)

末尾にコンマがあると `append` タイプオプションでエントリーが削除されなくなる

以前のバージョンでは、末尾のコンマ (リストの最後に空のエントリー) を `append` タイプオプション (例: `exclude`、`excludepkgs`、`includepkgs`) に追加すると、オプション内のすべてのエントリーが削除されることがありました。また、2つのコンマ (空のエントリー) を追加すると、コンマを使用した後のエントリーのみが使用されることがありました。

今回の更新で、先頭にコンマ以外の空のエントリー (リストの先頭に空のエントリー) は無視されます。その結果、先頭のコンマだけが `append` タイプオプションから既存のエントリーを削除し、ユーザーはこのエントリーを上書きできるようになりました。

(BZ#1788154)

5.4.3. シェルおよびコマンドラインツール

ReaR ディスクレイアウトに **Rancher 2 Longhorn iSCSI デバイス** および **ファイルシステムのエントリー** が含まれなくなりました。

今回の更新で、**ReaR** が作成したディスクレイアウトから **Rancher 2 Longhorn iSCSI デバイス** および **ファイルシステムのエントリー** が削除されました。

(BZ#1843809)

IBM POWER (リトルエンディアン) では、**4 GB を超えるファイルを使用したレスキューイメージの作成が有効** になりました。

以前のリリースでは、**ReaR** ユーティリティーは、**IBM POWER (リトルエンディアン)** アーキテクチャーに **4GB を超えるファイルを含むレスキューイメージを作成** できませんでした。今回の更新で問題が修正され、**IBM POWER**、**リトルエンディアン** に **4 GB を超えるファイルでレスキューイメージを作成** できるようになりました。

(BZ#1729502)

5.4.4. セキュリティー

SELinux により、**systemd-journal-gatewayd** が **corosync** で使用される **/dev/shm/** ファイルで **newfststat()** を呼び出さなくなりました。

以前のバージョンでは、**SELinux** ポリシーには、**systemd-journal-gatewayd** が **corosync** サービスによって作成されるファイルにアクセスできるルールが含まれません。その結果、**SELinux** は、**systemd-journal-gatewayd** による、**corosync** で作成された共有メモリーファイルの **newfststat()** 関数を呼び出しを拒否します。今回の更新で、**SELinux** は、**systemd-journal-gatewayd** が、**corosync** で作成された共有メモリーファイルの **newfststat()** を呼び出すのを回避しなくなりました。

(BZ#1746398)

Libreswan が、**すべての設定で `seccomp=enabled` で動作** するようになりました。

この更新以前は、**Libreswan** SECCOMP サポート実装で許可された **syscall** のセットが、**RHEL** ライブラリーの新たな使用と一致しませんでした。したがって、**SECCOMP** が **ipsec.conf** ファイルで有効になっていると、**syscall** のフィルタリングは **pluto** デーモンの正常な機能に必要な **syscall** に拒否されました。このデーモンは強制終了され、**ipsec** サービスが再起動されました。今回の更新で、新しく要求されたシステムコールがすべて許可され、**Libreswan** が **seccomp=enabled** オプションで正しく機能するようになりました。

(BZ#1544463)

auditd によるシステムの停止や電源オフが SELinux によって阻止されなくなりました

以前のバージョンでは、SELinux ポリシーに、Audit デーモンが `power_unit_file_t systemd` ユニットの起動できるようにするルールがありませんでした。したがって、Logging ディスクパーティションに領域が残っていない場合などに `auditd` がシステムの停止や電源オフを行うことができるように設定されていても、これを行うことができませんでした。

`selinux-policy` パッケージの今回の更新で、不足しているルールが追加され、`auditd` が Enforcing モードでのみシステムを停止したり、電源をオフにできるようになりました。

(BZ#1826788)

IPTABLES_SAVE_ON_STOP が正常に動作するように

以前のバージョンでは、保存された IP テーブルコンテンツを持つファイルが間違った SELinux コンテキストを受け取るため、`iptables` サービスの `IPTABLES_SAVE_ON_STOP` 機能は機能しませんでした。これにより、`iptables` スクリプトはパーミッションを変更できず、その後のスクリプトでは変更を保存できませんでした。今回の更新で、`iptables.save` ファイルおよび `ip6tables.save` ファイルの適切なコンテンツを定義し、ファイル名の移行ルールが作成されます。これにより、`iptables` サービスの `IPTABLES_SAVE_ON_STOP` 機能が正しく機能します。

(BZ#1776873)

NSCD データベースが異なるモードを使用できるようになりました。

`nsswitch_domain` 属性のドメインは、Name Service Cache Daemon (NSCD) サービスにアクセスできます。各 NSCD データベースは `thenscd.conf` ファイルで設定され、共有プロパティーはデータベースが `shared` メモリーまたは Socket モードを使用するかどうかを決定します。以前のバージョンでは、すべての NSCD データベースは `thenscd_use_shm` ブール値に応じて同じアクセスモードを使用する必要がありました。現在は、Unix ストリームソケットが常に許可されるようになったため、異なる NSCD データベースが異なるモードを使用できるようになりました。

(BZ#1772852)

oscap-ssh ユーティリティーが、--sudo でリモートシステムをスキャンすると、正しく機能するようになりました。

`oscap-ssh` ツールで `--sudo` オプションを使用してリモートシステムの Security Content Automation Protocol (SCAP) スキャンを実行すると、リモートシステムの `oscap` ツールが、`root` ユーザーとして、スキャン結果ファイルとレポートファイルを一時ディレクトリーに保存します。以前は、リモートマシンの `umask` 設定を変更すると、`oscap-ssh` がこれらのファイルにアクセスできないことがありました。今回の更新で問題が修正され、`oscap` がターゲットユーザーとしてファイルを保存し、`oscap-ssh` が通常ファイルにアクセスします。

(BZ#1803116)

OpenSCAP が、リモートファイルシステムを正しく処理

以前は、マウント仕様が 2 つのスラッシュで開始されなかった場合に、OpenSCAP はリモートファイルシステムを確実に検出できませんでした。これにより、OpenSCAP が、一部のネットワークベースのファイルシステムをローカルとして処理していました。今回の更新で、OpenSCAP はマウント仕様ではなく、ファイルシステムタイプを使用してファイルシステムを特定します。これにより、OpenSCAP がリモートファイルシステムを正しく処理するようになりました。

(BZ#1870087)

OpenSCAP で YAML の複数行文字列から空の行が削除されなくなる

以前のバージョンでは、OpenSCAP は YAML の複数ライン文字列から、データストリームから生成さ

れた Ansible 修正から空の行を削除していました。この影響を受ける Ansible の修正により、**openscap** ユーティリティーが対応する Open Vulnerability and Assessment Language (OVAL) チェックに失敗し、誤検出結果が生成されます。この問題は修正され、**openscap** は YAML の複数行の文字列から空の行を削除しなくなりました。

(BZ#1795563)

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

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

(BZ#1824152)

config.enabled がステートメントを適切に制御

以前のバージョンでは、**rsyslog** は、ステートメントの設定処理中に **config.enabled** ディレクティブを誤って評価していました。そのため、**include()** を除く各ステートメントに対して **parameter not known** エラーが表示されました。今回の更新で、すべてのステートメントに対して設定が同等に処理されます。その結果、**config.enabled** はエラーを表示せずに、ステートメントを正しく無効または有効にするようになりました。

(BZ#1659383)

fapolicyd が RHEL の更新を阻止しなくなりました。

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

[RHBA-2020:5242](#) アドバイザリーのリリースでは、**fapolicyd** がバイナリーパスの接尾辞を無視し、バイナリーが信頼データベースに一致するようにします。これにより、**fapolicyd** がルールを正しく適用し、更新プロセスを完了できるようになりました。

(BZ#1897090)

e8 プロファイルを使用して、Server with GUI で RHEL 8 システムを修復可能に

OpenSCAP Anaconda Add-on を使用して、**Verify Integrity with RPM** グループからルールを選択するプロファイルが含まれる **Server With GUI** パッケージグループでシステムを強化すると、システム上のメモリーが過剰に必要ではなくなりました。この問題の原因は OpenSCAP スキャナーでした。詳細は [Scanning large numbers of files with OpenSCAP causes systems to run out of memory](#) を参照してください。これにより、RHEL 8 Essential Eight (e8) プロファイルを使用してシステムを強化すると、**Server With GUI** も機能するようになりました。

(BZ#1816199)

5.4.5. ネットワーク

nft_compat モジュールによる iptables 拡張機能モジュールの自動読み込みがハングしなくなる

以前は、ネットワーク名前空間 (netns) が並行して操作中に **nft_compat** モジュールが拡張モジュール

がロードされた場合、その拡張が初期化中に **pernet** サブシステムを登録すると、ロックの競合が発生する可能性があります。これにより、**modprobe** コマンドと呼ばれるカーネルがハングします。これは、**libvirt** などの他のサービスが、**iptables** コマンドも実行するその他のサービスが原因である可能性があります。この問題が修正されました。その結果、**nft_compat** モジュールによる **iptables** 拡張機能モジュールの読み込みでハングしなくなりました。

(BZ#1757933)

firewalld サービスは、サービスが停止すると **ipsets** を削除するようになりました。

以前は、**firewalld** サービスを停止すると、**ipsets** が削除されませんでした。今回の更新でこの問題が修正されています。これにより、**firewalld** が停止すると、**ipset** がシステムに残らなくなりました。

(BZ#1790948)

firewalld がシャットダウン後に **ipset** エントリーを保持しない

以前は、**firewalld** をシャットダウンすると、**ipset** エントリーが削除されませんでした。したがって、**firewalld** サービスを停止した後も、**ipset** エントリーはカーネル内でアクティブな状態のままになりました。今回の修正により、**firewalld** をシャットダウンすると、**ipset** エントリーが期待どおりに削除されるようになりました。

(BZ#1682913)

firewalld が、リロード後に **ipset** エントリーを復元するようになりました。

以前は、再読み込み後に **firewalld** がランタイム **ipset** エントリーを保持しませんでした。したがって、ユーザーは不足しているエントリーをもう一度手動で追加する必要がありました。今回の更新で、再読み込み後に **ipset** エントリーを復元するように **firewalld** が変更されました。

(BZ#1809225)

nftables サービスおよび **firewalld** サービスが相互に排他的に

以前は、**nftables** サービスと **firewalld** サービスを同時に有効にできました。これにより、**nftables** は **firewalld** ルールセットを上書きしていました。今回の更新で、**nftables** サービスおよび **firewalld** サービスは相互排他的になり、同時に有効にできなくなりました。

(BZ#1817205)

5.4.6. カーネル

huge_page_setup_helper.py スクリプトが正しく動作する

Python 3 の **huge_page_setup_helper.py** スクリプトしたパッチが、誤って削除されました。これにより、**huge_page_setup_helper.py** の実行後に、以下のエラーメッセージが表示されます。

```
SyntaxError: Missing parentheses in call to 'print'
```

今回の更新で、**libhugetlbfs.spec** ファイルを更新してこの問題が修正されています。その結果、**huge_page_setup_helper.py** に上記のシナリオでエラーが表示されなくなりました。

(BZ#1823398)

大量の永続メモリーを備えたシステムは、タイムアウトなしでより迅速に起動します

元のソースコードではノードごとに1つの初期化スレッドしか許可されていなかったため、大量の永続

メモリーを備えたシステムは起動に長い時間がかかりました。たとえば、4ノードシステムの場合は、4つのメモリー初期化スレッドがありました。したがって、`/etc/fstab` ファイルに永続メモリーのファイルシステムがあると、デバイスが利用できるようになるまで待つ際に、システムがタイムアウトする場合があります。今回の更新では、ソースコードが単一ノード内で複数のメモリー初期化スレッドを許可するようになったため、問題が修正されました。その結果、システムはより迅速に起動し、説明されているシナリオではタイムアウトは発生しません。

(BZ#1666538)

bcc スクリプトが正常に BPF モジュールをコンパイル

スクリプトコードをコンパイルして Berkeley Packet Filter (BPF) モジュールを作成すると、**bcc** ツールキットはデータタイプ定義にカーネルヘッダーを使用していました。一部のカーネルヘッダーには、**KBUILD_MODNAME** マクロを定義する必要がありました。そのため、**KBUILD_MODNAME** を追加しない **bcc** スクリプトでは、さまざまな CPU アーキテクチャー全体で BPF モジュールをコンパイルする可能性がありました。以下の **bcc** スクリプトは影響を受けます。

- **bindsnoop**
- **sofdsnoop**
- **solisten**
- **tcpaccept**
- **tcpconnect**
- **tcpconnlst**
- **tcpdrop**
- **tcpretrans**
- **tcpsubnet**
- **tcptop**
- **tcptracer**

今回の更新で、**bcc** のデフォルトの **cflags** パラメーターに **KBUILD_MODNAME** を追加することで、この問題が修正されています。その結果、この問題は上記のシナリオで表示されなくなります。また、カスタマースクリプトは、**KBUILD_MODNAME** 自体を定義する必要はありません。

(BZ#1837906)

IBM Z で bcc-tools および bpfftrace が適切に動作するようになりました。

以前のバージョンでは、機能のバックポートにより、**ARCH_HAS_NON_OVERLAPPING_ADDRESS_SPACE** カーネルオプションが導入されました。ただし、IBM Z アーキテクチャー用の **bcc-tools** パッケージおよび **bpfftrace** トレース言語パッケージでは、このオプションが適切にサポートされませんでした。そのため、**bpff()** システムコールが **Invalid argument exception** で失敗し、**bpfftrace** が BPF プログラムを読み込む際に **Error loading program** を示すエラーを出して失敗しました。今回の更新により、**ARCH_HAS_NON_OVERLAPPING_ADDRESS_SPACE** オプションが削除されました。その結果、上記のシナリオで問題が表示されなくなりました。

(BZ#1847837、BZ#1853964)

エントロピーがないためブートプロセスが失敗しなくなる

以前のバージョンでは、エントロピーがないためにブートプロセスが失敗しました。カーネルがブートプロセスの早い段階でエントロピーを収集できるように、より良いメカニズムが使用されるようになりました。これは、ハードウェア固有の割り込みに依存しません。今回の更新では、起動の初期で無作為な生成のセキュリティーを保護するのに十分なエントロピーの可用性を確保することでこの問題が修正されています。その結果、この修正によりキックスタートのタイムアウトや起動速度が遅くなり、起動プロセスが期待どおりに機能します。

(BZ#1778762)

kexec を使用した再起動が期待どおりに動作する

以前のバージョンでは、Amazon EC2 Nitro プラットフォームでカーネルを再起動すると、カーネル実行パスの `shutdown()` の呼び出し中に `remove` モジュール (`rmmod`) が呼び出されませんでした。したがって、**kexec** システムコールを使用してカーネルを再起動すると、失敗していました。今回の更新で、安全なカーネル実行を可能にする PCI `shutdown()` ハンドラーを追加してこの問題が修正されます。その結果、Amazon EC2 Nitro プラットフォームで **kexec** を使用した再起動が繰り返し行われなくなりました。

(BZ#1758323)

ダンプターゲットとして vPMEM メモリーを使用した繰り返し再起動が期待どおりに機能するようになりました。

以前は、**kdump** または **fadump** のダンプターゲットとして仮想永続メモリー (vPMEM) 名前空間を使用すると、**papr_scm** モジュールが vPMEM がサポートするメモリーのマッピングを解除し、メモリーをリニアマップに再追加していました。

その結果、この動作により、ハイパーバイザーコール (HCall) がトリガーされ、POWER Hypervisor がトリガーされます。その結果、キャプチャーカーネルブートが大幅に遅くなり、ダンプファイルを保存するのに時間がかかります。今回の更新で問題が修正され、上記のシナリオで起動プロセスが期待どおりに動作するようになりました。

(BZ#1792125)

ICE ドライバーの NIC ポートをモード 5 ボンディングマスターインターフェイスに追加しようとしても失敗しなくなりました。

以前のバージョンでは、**ICE** ドライバー NIC ポートをモード 5 (**balance-tlb**) ボンディングマスターインターフェイスに追加しようとする、**Master 'bond0', Slave 'ens1f0': Error: Enslave failed** のエラーで失敗する可能性があります。そのため、NIC ポートをボンディングマスターインターフェイスに NICE ポートを追加するときに断続的に問題が発生していました。今回の更新で問題が修正され、インターフェイスの追加は失敗しなくなりました。

(BZ#1791664)

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

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

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

(BZ#1708456)

5.4.7. 高可用性およびクラスター

ファイルシステムエージェントで GFS2 ファイルシステムを使用する場合は、**fast_stop** オプションがデフォルトで **no** に設定されます。

以前は、ファイルシステムエージェントで GFS2 ファイルシステムを使用すると、**fast_stop** オプションがデフォルトで **yes** に設定されます。この値により、GFS2 ファイルシステムのマウント解除にかかる時間が原因で、不要なフェンスイベントが発生する可能性があります。今回の更新で、このオプションはデフォルトで **no** に設定されます。他のすべてのファイルシステムでは、デフォルトで **yes** に設定されます。

(BZ#1814896)

fence_compute および **fence_evacuate** エージェントは、より標準的な方法で **insecure** オプションを解釈するようになりました。

以前は、**fence_compute** および **fence_evacuate** エージェントは、**--insecure** がデフォルトで指定されたかのように動作していました。今回の更新により、コンピュートまたは退避サービスに有効な証明書を使用しないお客様は、**insecure=true** を設定し、CLI から手動で実行する場合に **--insecure** オプションを使用する必要がありました。これは、他のすべてのエージェントの動作に一貫性があります。

(BZ#1830776)

5.4.8. 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 による CPU 消費の最適化

以前の **libdb** データベースの更新により、trickle スレッドにおける CPU 消費が過剰になっていました。今回の更新で、CPU 使用率が最適化されました。

(BZ#1670768)

did_you_mean Ruby gem には、重要なライセンスを持つファイルが含まれなくなりました。

以前は、**ruby:2.5** モジュールストリームで利用可能な **did_you_mean** gem には、自分以外のライセンスを持つファイルが含まれていました。今回の更新で、影響を受けるファイルが削除されます。

(BZ#1846113)

nginx が PKCS#11 URI を使用して、ハードウェアセキュリティトークンからサーバー証明書をロードできるようになりました。

nginx Web サーバーの **ssl_certificate** ディレクティブは、PKCS#11 モジュールを利用してハードウェアセキュリティトークンから直接 TLS サーバー証明書を読み込むことをサポートします。以前は、PKCS#11 URI を使用して、ハードウェアセキュリティトークンからサーバー証明書を読み込むことができませんでした。

(BZ#1668717)

5.4.9. コンパイラーおよび開発ツール

DT_FILTER を使用する共有ライブラリーの読み込み中に **glibc** 動的ローダーが失敗しなくなりました。

この更新以前は、共有オブジェクトの動的ローダー実装にフィルターとして不具合があると、フィルターを使用する共有ライブラリーがロードされ、コンストラクターを持つ際に動的ローダーが失敗していました。今回のリリースにより、フィルターの動的ローダー実装 (**DT_FILTER**) が修正され、このよ

うな共有ライブラリーを正しく処理できるようになりました。その結果、上述のシナリオで動的ローダーが想定どおりに機能するようになりました。

(BZ#1812756)

glibc が、getmntent() リストから擬似マウントを削除可能

カーネルには、ユーザー空間に公開されるテーブルの **automount** 疑似エントリーが含まれます。そのため、**getmntent()** API を使用するプログラムは、通常のマウントとこの擬似マウントの両方をリストに表示されるようになりました。擬似マウントは、実際のマウントには対応せず、有効な情報も含まれます。

今回の更新で、mount エントリーに **automount(8)** 設定にある **ignore** マウントオプションがある場合、**glibc** ライブラリーでこの擬似マウントが **getmntent()** リストから削除されるようになりました。以前の動作を想定するプログラムでは、異なる API を使用する必要があります。

(BZ#1743445)

movv1qi パターンにより、IBM Z の自動ベクターコードに誤コンパイルが発生しなくなりました。

この更新以前は、**movv1qi** パターンに対して誤った負荷命令が生成されていました。これにより、自動アクターのフィルターが有効になっていると、IBM Z システムで不適切なコンパイルが実行される可能性があります。今回の更新で **movv1qi** パターンが修正され、コードコンパイルおよび実行が正常に実行されるようになりました。

(BZ#1784758)

PAPI_event_name_to_code() が複数のスレッドで適切に動作するようになりました。

この更新以前は、PAPI 内部コードがスレッドの調整を正しく処理しませんでした。したがって、複数のスレッドが **PAPI_event_name_to_code()** 操作を使用すると、競合状態が発生し、操作に失敗していました。今回の更新で、PAPI 内部コードで複数のスレッドの処理が強化されます。その結果、**PAPI_event_name_to_code()** 操作を使用したマルチスレッドコードが正しく機能するようになりました。

(BZ#1807346)

IBM Power Systems の glibc math 関数のパフォーマンスを改善

以前は、**glibc** math 関数は、IBM Power Systems で不要な浮動小数点のステータスの更新とシステムコールを実行していたため、パフォーマンスに悪影響を及ぼしていました。今回の更新で、不要な浮動小数点ステータスの更新が削除され、**ceil()**、**ceilf()**、**fegetmode()**、**fesetmode()**、**fesetenv()**、**fegetexcept()**、**feenableexcept()**、**fedisableexcept()**、**fegetround()**、**fesetround()** の実装が改善されました。その結果、IBM Power Systems では、math ライブラリーのパフォーマンスが向上しました。

(BZ#1783303)

メモリー保護鍵が IBM Power で対応

IBM Power Systems では、メモリー保護キーインターフェイス **pkey_set** および **pkey_get** は、以前はスタブ機能でした。そのため、常に失敗していました。今回の更新でインターフェイスが実装され、GNU C ライブラリー (**glibc**) が IBM Power Systems でのメモリー保護キーをサポートするようになりました。

現在、メモリー保護キーにはハッシュベースのメモリー管理ユニット (MMU) が必要なため、カーネルパラメーター **disable_radix** で特定のシステムを起動する必要があるかもしれません。

(BZ#1642150)

papi-testsuite および **papi-devel** が、必要な **papi-libs** パッケージをインストールするようになりました。

以前は、**papi-testsuite** および **papi-devel RPM** パッケージは、一致する **papi-libs** パッケージの依存関係を宣言しませんでした。そのため、テストは実行に失敗し、開発者はアプリケーションで利用可能な **papi** 共有ライブラリーの必要なバージョンがありませんでした。

今回の更新で、**papi-testsuite** パッケージまたは **papi-devel** パッケージのいずれかをインストールすると、**papi-libs** パッケージもインストールされます。そのため、**papi-testsuite** に、テストを実行できるように正しいライブラリーが設定され、**papi-devel** を使用する開発者では、適切なバージョンの **papi** 共有ライブラリーに実行ファイルがリンクされるようになりました。

(BZ#1664056)

複数のアーキテクチャー用の **lldb** パッケージのインストールでファイルの競合が発生しなくなる

以前では、**lldb** パッケージは、アーキテクチャーに依存しない場所にアーキテクチャー依存ファイルをインストールしていました。そのため、パッケージの 32 ビットバージョンと 64 ビットバージョンの両方をインストールすると、ファイルの競合が発生していました。今回の更新で、アーキテクチャー依存の場所のファイルがパッケージ化されるようになりました。これにより、上記のシナリオでの **lldb** のインストールが正常に完了します。

(BZ#1841073)

getaddrinfo がメモリー割り当てエラーを適切に処理

以前は、メモリー割り当てに失敗すると、GNU C ライブラリー **glibc** の **getaddrinfo** 関数は、内部リゾルバーコンテキストをリリースしませんでした。したがって、**getaddrinfo** は、呼び出しスレッドの残りのライフタイム中は **/etc/resolv.conf** ファイルを再読み込みできず、メモリーリークが発生していました。

今回の更新により、リゾルバーコンテキストの追加リリース操作でエラー処理パスが変更されました。その結果、**getaddrinfo** は、断続的なメモリー割り当ての失敗後も新しい設定値で **/etc/resolv.conf** を再読み込みします。

(BZ#1810146)

glibc が、IFUNC リゾルバーの順序によって発生する特定の障害を回避

以前は、GNU C ライブラリー **glibc** の **librt** および **libpthread** ライブラリーの実装には、以下の関数の間接関数 (IFUNC) リゾルバーが含まれていました。

time_gettime、**clock_getcpu**、**clockid**、**clock_nanosleep**、**time_settime**、**vfork**。場合によっては、IFUNC リゾルバーは、**librt** ライブラリーおよび **libpthread** ライブラリーが再配置される前に実行できることがあります。その結果、早いプログラムの起動時にアプリケーションが **glibc** 動的ローダーで失敗していました。

今回のリリースにより、これらの関数の実装が **glibc** の **libc** コンポーネントに移動し、上記の問題が発生しなくなりました。

(BZ#1748197)

アサーションの失敗が **pthread_create** の実行中に発生しない

以前のバージョンでは、**glibc** 動的ローダーは内部 Thread Local Storage (TLS) モジュール ID カウンターへの変更をロールバックしませんでした。これにより、**pthread_create** 関数のアサーションエ

ラーが生じ、特定の 방법으로 **dlopen** 関数が失敗していました。今回の修正により、**glibc** 動的ローダーは、特定の障害が発生しなくなった後に、後で TLS モジュール ID カウンターを更新するようになりました。その結果、アサーションエラーが生じなくなりました。

(BZ#1774115)

glibc が **nss_db** を使用して 32 ビットアプリケーションに正しい依存関係をインストールするようになりました。

以前は、**nss_db.x86_64** パッケージは **nss_db.i686** パッケージで依存関係を宣言しませんでした。したがって、32 ビット環境 **glibc.i686** がインストールされている場合でも、自動インストールでは **nss_db.i686** をシステムにインストールしませんでした。そのため、**nss_db** を使用する 32 ビットアプリケーションは、正確なユーザーデータベーススルックアップを実行しなくなり、同じ設定の 64 ビットアプリケーションが正常に機能します。

今回の更新で、**glibc** パッケージの弱い依存関係が、**glibc.i686** および **nss_db** の両方がシステムにインストールされていると、**nss_db.i686** パッケージのインストールがトリガーされます。これにより、システム管理者が **nss_db.i686** パッケージを明示的にインストールしていない場合でも、**nss_db** を使用する 32 ビットアプリケーションが正しく機能するようになりました。

(BZ#1807824)

Odia 言語で更新された **glibc** ロケール情報

以前 Orissa と呼ばれる Indian 状態の名前が Odisha に変更され、公式の言語の名前が Oriya から Odia に変更になりました。今回の更新で、**glibc** ロケール情報に言語の新しい名前が反映されるようになりました。

(BZ#1757354)

LLVM サブパッケージが、アーキテクチャーに依存しない場所に **arch** 依存ファイルをインストールするようになりました。

以前では、LLVM サブパッケージは、アーキテクチャーに依存しない場所に **arch** 依存ファイルをインストールしていました。これにより、32 ビットおよび 64 ビットバージョンの LLVM のインストール時に競合が生じました。今回の更新で、パッケージファイルがアーキテクチャー依存の場所に正しくインストールされるようになり、バージョンの競合を避けるようになりました。

(BZ#1820319)

glibc でパスワードおよびグループ検索が失敗しなくなりました。

以前は、**glibc** ライブラリーの **nss_compat** モジュールが、パスワードおよびグループエントリーの処理中に誤ったエラーコードで **errno** 状態を上書きしていました。その結果、アプリケーションはバッファを予想通りにサイズ変更せず、パスワードおよびグループの検索に失敗していました。今回の更新で問題が修正され、ルックアップが期待どおりに完了するようになりました。

(BZ#1836867)

5.4.10. ID 管理

SSSD がデフォルトでワイルドカード文字が含まれるすべてのルールをダウンロードしない

以前は、**ldap_sudo_include_regexp** オプションがデフォルトで誤って **true** に設定されていました。これにより、SSSD が SSSD ルールの実行または更新を開始した場合、SSSD は **sudoHost** 属性にワイルドカード文字 (*) が含まれるすべてのルールをダウンロードしていました。今回の更新でバグが修正され、**ldap_sudo_include_regexp** オプションがデフォルトで **false** に設定されるようになりました。その結果、上記の問題が発生しなくなりました。

(BZ#1827615)

krb5 が、許可された暗号化タイプのみを要求する

以前のバージョンでは、**default_tgs_etypes** 属性または **default_tkt_etypes** 属性が設定されていない場合、`/etc/krb5.conf` ファイルの **permitted_etypes** 変数で指定された暗号化タイプは、デフォルトの暗号化タイプに適用されませんでした。そのため、Kerberos クライアントは RC4 などの非推奨の暗号スイートを要求することができ、これにより、他のプロセスが失敗する可能性があります。今回の更新で、**allow_etypes** 変数で指定した暗号化タイプもデフォルトの暗号化タイプに適用され、許可される暗号化タイプのみが要求されるようになりました。

RHEL 8 で非推奨となった RC4 暗号スイートは、ユーザー、サービス、および AD フォレスト内の Active Directory (AD) ドメインとの間の信頼のデフォルト暗号化タイプです。

- AD フォレストの AD ドメイン間で強固な AES 暗号化タイプに対応させるには、Microsoft の記事 [AD DS: Security: Kerberos "Unsupported etype" error when accessing a resource in a trusted domain](#) を参照してください。
- AD との後方互換性のために、IdM サーバーで非推奨の RC4 暗号化タイプのサポートを有効にするには、**update-crypto-policies --set DEFAULT:AD-SUPPORT** コマンドを使用します。

(BZ#1791062)

KDC で LDAP バックエンドからパスワード有効期間のポリシーを正常に適用されるようになりました

以前のバージョンは、Kerberos LDAP バックエンドにより、パスワードポリシーが正しく適用されていなかったため、IPA 以外の Kerberos Distribution Center (KDC) ではパスワードの最大有効期間を保証できませんでした。今回の更新で、Kerberos LDAP バックエンドが修正され、パスワードの有効期間が期待どおりに機能するようになりました。

(BZ#1784655)

SSSD を使用する AD クライアントにパスワード有効期限の通知が送信されます

以前のバージョンでは、SSSD を使用する Active Directory クライアント (IdM 以外) には、パスワード有効期限の通知が送信されませんでした。これは Kerberos 認証情報を取得するために、SSSD インターフェイスに最近変更が加えられたためです。

Kerberos インターフェイスが更新され、有効期限の通知が正しく送信されるようになりました。

(BZ#1820311)

間接的な CoS 定義を使用時の、Directory Server でのメモリーリークを修正

以前のバージョンでは、間接的な CoS (Class of Service) 定義を処理すると、Directory Server では、間接 CoS 定義を使用する検索操作ごとにメモリーリークが発生していました。今回の更新で、Directory Server は、処理後にデータベースエントリーに関連する CoS 内部構造をすべて解放するようになりました。その結果、間接的な CoS 定義の使用時にサーバーでのメモリーリークがなくなりました。

(BZ#1816862)

AD ユーザーの ID オーバーライドの追加が IdM Web UI で機能するようになりました。

以前は、IdM Web UI の使用時に、管理ロールへのアクセスを付与する目的で、Default Trust View の Active Directory (AD) ユーザーの ID オーバーライドを Identity Management (IdM) グループに追加していました。今回の更新でバグが修正されました。これにより、このシナリオで Web UI と IdM コマンドラインインターフェイス (CLI) の両方を使用できるようになりました。

[\(BZ#1651577\)](#)

FreeRADIUS がパッケージのインストール時に証明書を生成しなくなりました

以前のバージョンでは、FreeRADIUS がパッケージのインストール時に証明書を生成していたため、以下の問題が発生することがありました。

- キックスタートを使用して FreeRADIUS がインストールされている場合には、システムのエントローピーが十分ではない場合に証明書が生成される可能性があり、インストールに失敗したり、セキュアな証明書が少なくなることがありました。
- パッケージは、ターゲットマシンではなくビルダーマシンでパッケージインストールが行われるため、コンテナなどのイメージの一部としてビルドすることは容易ではありませんでした。イメージから起動するすべてのインスタンスに同じ証明書情報があります。
- 証明書を手動で削除し、再生成する必要があるため、エンドユーザーが環境で簡単な仮想マシンを生成することは容易ではありませんでした。

今回の更新で、FreeRADIUS インストールで、デフォルトの自己署名 CA 証明書または下位 CA 証明書が生成されなくなりました。FreeRADIUS が **systemd** から起動されると、以下を行います。

- 必要な証明書がすべて見つからない場合は、デフォルト証明書のセットが生成されます。
- 予想される証明書の1つまたは複数が存在する場合は、新しい証明書を生成しません。

[\(BZ#1672285\)](#)

FreeRADIUS が FIPS 準拠の Diffie-Hellman パラメーターを生成するようになりました。

openssl が **dhparam** を介して Diffie-Hellman (dh) パラメーターを生成できない新しい FIPS 要件により、dh パラメーターの生成は FreeRADIUS ブートストラップスクリプトから削除され、**rfc3526-group-18-8192.dhparam** ファイルはすべてのシステムの FreeRADIUS パッケージに含まれており、FreeRADIUS が FIPS モードで起動するようになっています。

/etc/raddb/certs/bootstrap および **/etc/raddb/certs/Makefile** をカスタマイズして、必要に応じて DH パラメーターの生成を復元できるようにしてください。

[\(BZ#1859527\)](#)

Healthcheck の更新により ipa-healthcheck-core および ipa-healthcheck の両方が適切に更新されるようになりました。

以前では、**yum update healthcheck** を入力すると、**ipa-healthcheck** パッケージが更新されず、**ipa-healthcheck-core** パッケージに置き換えられました。そのため、更新後に **ipa-healthcheck** コマンドが機能しませんでした。

今回の更新でバグが修正され、**ipa-healthcheck** を更新すると、**ipa-healthcheck** パッケージと **ipa-healthcheck-core** パッケージの両方が正しく更新されるようになりました。これにより、更新後に **Healthcheck** ツールが正しく動作します。

[\(BZ#1852244\)](#)

5.4.11. グラフィックインフラストラクチャー

ハイブリッド Nvidia GPU を備えたラップトップがサスペンドから正常に再開できるようになりました。

以前は、この問題を解決するために、特定のハードウェア構成で、特定のオペレーティング

以前では、**nouveau** グラフィックスドライバーが、省電力モードからの特定のラップトップでハイブリッド Nvidia GPU の電源をオンにできないことがありました。その結果、ラップトップはサスペンドから再開できませんでした。

今回の更新で、Runtime Power Management (**runpm**) システムの複数の問題が修正されました。その結果、ハイブリッドグラフィックスを備えたラップトップがサスペンドから正常に再開できるようになりました。

(JIRA:RHELPLAN-57572)

5.4.12. 仮想化

デフォルトの CPU モデルを使用する仮想マシンの移行がより確実に機能するようになりました。

以前は、特定の CPU モデルなしで仮想マシン (VM) が作成された場合、QEMU が **libvirt** サービスに見えないデフォルトモデルを使用していました。したがって、仮想マシンのデフォルトの CPU モデルをサポートしないホストに仮想マシンを移行する可能性があります。これにより、移行後にゲスト OS でクラッシュおよび誤った動作が生じることがありました。

今回の更新で、**libvirt** が、仮想マシンの XML 設定でデフォルトとして **qemu64** モデルを明示的に使用するようになりました。その結果、ユーザーがデフォルトの CPU モデルを持つ仮想マシンをそのモデルをサポートしないホストに移行しようとすると、**libvirt** がエラーメッセージを正しく生成します。

ただし、Red Hat は、仮想マシンに特定の CPU モデルを使用することを強く推奨します。

(JIRA:RHELPLAN-45906)

5.4.13. コンテナ

Podman での FIPS サポートに関する注意事項

Federal Information Processing Standard (FIPS) を使用するには、認定済みモジュールを使用する必要があります。以前のバージョンでは、Podman は起動時に適切なフラグを有効にして、コンテナに認定モジュールを正しくインストールしていました。ただし、本リリースでは、Podman は FIPS システム全体の crypto-policy の形式でシステムによって提供される追加アプリケーションヘルパーを適切に設定しません。認定モジュールでは、システム全体の crypto-policy を設定する必要はありませんが、適切な方法で暗号化モジュールを使用するアプリケーションが強化されます。この問題を回避するには、他のアプリケーションコードを実行する前に、コンテナを変更して **update-crypto-policies --set FIPS** コマンドを実行します。今回の修正では、**update-crypto-policies --set FIPS** コマンドが不要になりました。

(BZ#1804193)

5.5. テクノロジープレビュー

本パートでは、Red Hat Enterprise Linux 8.3 で利用可能なテクノロジープレビュー機能のリストを提示します。

テクノロジープレビュー機能に対する Red Hat のサポート範囲の詳細は、[テクノロジープレビュー機能のサポート範囲](#) を参照してください。

5.5.1. ネットワーク

xt_u32 Netfilter モジュールを有効化

xt_u32 Netfilter モジュールは **kernel-modules-extra** rpm で利用できるようになりました。このモジュールは、他のプロトコルベースのパケットフィルターからアクセスできないデータに基づいて、パケットの転送に役立ちます。したがって、**nftables** の手動移行が容易になります。ただし、Netfilter モジュール **xt_u32** は Red Hat ではサポートされていません。

(BZ#1834769)

nmstate がテクノロジープレビューとして利用可能になりました。

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

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

(BZ#1674456)

AF_XDP がテクノロジープレビューとして利用可能に

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

(BZ#1633143)

XDP がテクノロジープレビューとして利用可能になりました。

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

(BZ#1503672)

KTLS がテクノロジープレビューとして利用可能になりました。

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

(BZ#1570255)

テクノロジープレビューとして利用できる XDP 機能

Red Hat は、以下の eXpress Data Path (XDP) 機能をサポート対象外のテクノロジープレビューとして提供します。

- AMD および Intel 64 ビット以外のアーキテクチャーで XDP プログラムを読み込む。 **libxdp** ライブラリーは、AMD および Intel 64 ビット以外のアーキテクチャーでは使用できません。
- **XDP_TX** および **XDP_REDIRECT** の戻りコード。
- XDP ハードウェアオフロード。この機能を使用する前は、[Unloading XDP programs on Netronome network cards that use the nfp driver fails](#) を参照してください。

(BZ#1889737)

act_mpls モジュールがテクノロジープレビューとして利用可能になりました。

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

(BZ#1839311)

Multipath TCP がテクノロジープレビューとして利用可能

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

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

詳細は、[Multipath TCP の使用](#) を参照してください。

(JIRA:RHELPLAN-41549)

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

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

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

(BZ#1906489)

5.5.2. カーネル

kexec fast reboot 機能がテクノロジープレビューとして利用可能になりました。

kexec fast reboot 機能は、引き続きテクノロジープレビューとして利用できます。**kexec fast reboot** は、カーネルが先に BIOS (Basic Input/Output System) を経由せずに、2 番目のカーネルを直接起動できるようにすることで、ブートプロセスの時間を大幅に短縮します。この機能を使用するには、以下を実行します。

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

(BZ#1769727)

eBPF がテクノロジープレビューとして利用可能になりました。

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

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

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

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

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

- **bpftrace**。これは、eBPF 仮想マシンを使用する高レベルの追跡言語です。
- **AF_XDP**。これは、eXpress Data Path (XDP) パスを、パケット処理のパフォーマンスを優先するアプリケーションのユーザー空間に接続するためのソケットです。

(BZ#1559616)

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

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

(BZ#1495358)

テクノロジープレビューとして利用できる Soft-RoCE

Remote Direct Memory Access (RDMA) over Converged Ethernet (RoCE) は、RDMA over Ethernet を実装するネットワークプロトコルです。Soft-RoCE は、RoCE v1 および RoCE v2 の 2 つのプロトコルバージョンに対応する RoCE のソフトウェア実装です。Soft-RoCE ドライバーの **rdma_rxe** は、RHEL 8 ではサポートされていないテクノロジープレビューとして利用できます。

(BZ#1605216)

5.5.3. ファイルシステムおよびストレージ

NVMe/TCP がテクノロジープレビューとして利用可能になりました。

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

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

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

(BZ#1696451)

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

Red Hat Enterprise Linux 8 では、ファイルシステムの DAX がテクノロジープレビューとして利用でき

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

(BZ#1627455)

OverlayFS

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

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

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

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

また、OverlayFS の使用には、以下のルールと制限が適用されます。

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

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

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

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

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

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

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

(BZ#1690207)

Stratis がテクノロジープレビューとして利用可能になりました。

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

Stratis を使用すると、次のようなストレージタスクをより簡単に実行できます。

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

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

Stratis はテクノロジープレビューとして提供されます。

詳細については、Stratis のドキュメント ([Stratis ファイルシステムの設定](#)) を参照してください。

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

(JIRA:RHELPLAN-1212)

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

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

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

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

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

(JIRA:RHELPLAN-13195)

5.5.4. 高可用性およびクラスター

pcs cluster setup コマンドのローカルモードバージョンがテクノロジープレビューとして利用可能になりました。

デフォルトでは、**pcs cluster setup** コマンドは、すべての設定ファイルをクラスターノードに自動的に同期します。Red Hat Enterprise Linux 8.3 以降、**pcs cluster setup** コマンドは、**--corosync-conf** オプションをテクノロジープレビューとして提供します。このオプションを指定すると、コマンドが **local** モードに切り替わります。このモードでは、**pcs** は他のノードと通信せずに **corosync.conf** ファイルを作成し、ローカルノード上の指定されたファイルに保存します。これにより、スクリプトで **corosync.conf** ファイルを作成し、スクリプトでそのファイルを処理できます。

(BZ#1839637)

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

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

(BZ#1619620)

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

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

(BZ#1784200)

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

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

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

サービスを適切に引き継ぐことができないことを事前に把握できる場合は、ノードがピアをフェンシングする意味がないのであれば、ユーザーは特に 2 ノードクラスターでこのエージェントを使用できま

す。たとえば、ネットワークアップリンクに到達してサービスがクライアントに到達できない場合は、ノードがサービスを引き継ぐ意味はありません。これは、ルーターへの ping が検出できる状況が考えられます。

(BZ#1775847)

5.5.5. ID 管理

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

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

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

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

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

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

(BZ#1664719)

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

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

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

- DNSSEC Operational Practices, Version 2 - <http://tools.ietf.org/html/rfc6781#section-2>
- Secure Domain Name System (DNS) Deployment Guide - <http://dx.doi.org/10.6028/NIST.SP.800-81-2>
- DNSSEC Key Rollover Timing Considerations - <http://tools.ietf.org/html/rfc7583>

統合 DNS のある IdM サーバーは、DNSSEC を使用して、他の DNS サーバーから取得した DNS 回答を検証することに注意してください。これが、推奨される命名方法に従って設定されていない DNS ゾーンの可用性に影響を与える可能性があります。

(BZ#1664718)

5.5.6. デスクトップ

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

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

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

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

IBM Z の GNOME デスクトップがテクノロジープレビューとして利用可能に

Firefox Web ブラウザーを含む GNOME デスクトップが、IBM Z アーキテクチャーでテクノロジープレビューとして利用できるようになりました。VNC を使用して GNOME を実行するリモートグラフィカルセッションに接続し、IBM Z サーバーを設定および管理できるようになりました。

(JIRA:RHELPLAN-27737)

5.5.7. グラフィックインフラストラクチャー

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

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

(BZ#1698565)

Intel Tiger Lake グラフィックがテクノロジープレビューとして利用可能に

Intel Tiger Lake UP3 および UP4 Xe グラフィックがテクノロジープレビューとして利用できるようになりました。

Intel Tiger Lake グラフィックでハードウェアアクセラレーションを有効にするには、カーネルコマンドラインに以下のオプションを追加します。

```
i915.force_probe=pci-id
```

このオプションでは、**pci-id**を以下のいずれかに置き換えます。

- Intel GPU の PCI ID
- すべての高品質ハードウェアで **i915** ドライバーを有効にする * 文字

(BZ#1783396)

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

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

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

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

postfix ロールは、テクノロジープレビューとして利用可能です。

以下のロールが完全にサポートされています。

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

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

(BZ#1812552)

5.5.9. 仮想化

RHEL 8 Hyper-V 仮想マシンで KVM 仮想化が利用可能に

ネストされた KVM 仮想化は、テクノロジープレビューとして、Microsoft Hyper-V ハイパーバイザーで使用できるようになりました。これにより、Hyper-V ホストで実行している RHEL 8 ゲストシステムで仮想マシンを作成できます。

この機能は、現在 Intel システムでのみ有効です。また、ネストされた仮想化は、Hyper-V でデフォルトで有効になっていない場合があります。これを有効にするには、以下の Microsoft ドキュメントを参照してください。

<https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/user-guide/nested-virtualization>

(BZ#1519039)

KVM 仮想マシンの AMD SEV。

テクノロジープレビューとして、RHEL 8 に、KVM ハイパーバイザーを使用する AMD EPYC ホストマシン用のセキュア暗号化仮想化 (SEV) 機能が同梱されます。仮想マシンで有効になっている場合は、ホストが仮想マシンのデータにアクセスできないように、SEV が仮想マシンメモリーを暗号化します。ホストがマルウェアに感染した場合は、これにより仮想マシンのセキュリティーが向上します。

1台のホストでこの機能を同時に使用できる仮想マシンの数は、ホストのハードウェアによって決まります。現在の AMD EPYC プロセッサは、SEV を使用して 509 台以下の稼働中の仮想マシンに対応します。

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

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

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

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

Intel vGPU

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

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

(BZ#1528684)

入れ子仮想マシンの作成

入れ子 KVM 仮想化は、RHEL 8 で AMD64 および IBM Z システムホストで実行している KVM 仮想マシン用のテクノロジープレビューとして提供されます。この機能を使用すると、物理 RHEL 8 ホストで実行中の RHEL 7 または RHEL 8 仮想マシンがハイパーバイザーとして機能し、独自の仮想マシンをホストできます。

RHEL 8.2 以降では、Intel 64 ホストで実行している仮想マシンでは、入れ子になった仮想化が完全にサポートされることに注意してください。

(JIRA:RHELPLAN-14047, JIRA:RHELPLAN-24437)

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

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

- ネットワークインターフェイスコントローラー (NIC) に対して SR-IOV サポートが有効になっている
- 仮想 NIC の SR-IOV サポートが有効になっている
- 仮想スイッチの SR-IOV サポートが有効になっている
- NIC からの VF (Virtual Function) が仮想マシンに割り当てられている

この機能は現在、Microsoft Windows Server 2019 および 2016 で対応しています。

(BZ#1348508)

5.5.10. コンテナ

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

registry.redhat.io/rhel8/podman コンテナイメージは、**podman** パッケージをコンテナ化した実装です。**podman** ツールは、コンテナおよびイメージの管理、それらのコンテナにマウントされたボリューム、およびコンテナのグループから作成された Pod を管理するために使用されます。**podman** は、コンテナライフサイクル管理の **libpod** ライブラリーに基づいています。**libpod** ライブラリーは、コンテナ、Pod、コンテナイメージ、およびボリュームを管理するための API を提供します。このコンテナイメージを使用すると、システムに **podman** パッケージをインストールしなくても、コンテナイメージの作成、変更、実行を行うことができます。このユースケースでは、root 以外のユーザーとして rootless モードでこのイメージを実行することを説明します。**registry.redhat.io/rhel8/podman** コンテナイメージをプルするには、アクティブな Red Hat Enterprise Linux サブスクリプションが必要です。

(BZ#1627899)

crun 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。

crun OCI ランタイムが **container-roots:rhel8** モジュールに追加されました。**crun** が、cgroupsV2 での実行権限を提供します。**crun** は、コンテナがルートレスユーザーの追加のグループにアクセスできるアノテーションをサポートします。これは、ユーザーにグループアクセスのみが割り当てられているディレクトリーへのボリュームマウントや、setgid のあるディレクトリーに役に立ちます。

(BZ#1841438)

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)

5.6. 非推奨になった機能

ここでは、Red Hat Enterprise Linux 8 で **非推奨** となった機能の概要を説明します。

非推奨の機能は、本製品の今後のメジャーリリースではサポートされない可能性が高く、新たに実装することは推奨されません。特定のメジャーリリースにおける非推奨機能の最新情報は、そのメジャーリリースの最新版のリリースノートを参照してください。

非推奨の機能のサポートステータスは、Red Hat Enterprise Linux 8 では変更されていません。サポート期間の詳細は、[Red Hat Enterprise Linux ライフサイクル](#) および [Red Hat Enterprise Linux Application Streams ライフサイクル](#) を参照してください。

現行および今後のメジャーリリースでは、非推奨のハードウェアコンポーネントの新規実装は推奨されません。ハードウェアドライバーの更新は、セキュリティと重大な修正のみに行われます。Red Hat では、このようなハードウェアの早期交換を推奨します。

パッケージが非推奨となり、使用の継続が推奨されない場合があります。製品からパッケージが削除されることもあります。その場合には、製品のドキュメントで、非推奨となったパッケージと同様、同一、またはより高度な機能を提供する最近のパッケージが指定され、詳しい推奨事項が記載されます。

RHEL 7 で使用され、RHEL 8 で **削除された** 機能の詳細は [RHEL 8 の導入における検討事項](#) を参照してください。

5.6.1. インストーラーおよびイメージの作成

複数のキックスタートコマンドおよびオプションが非推奨になりました。

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

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

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

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

(BZ#1642765)

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

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

(BZ#1637872)

RHEL 8 で非推奨となった Image Builder `lorax-composer` バックエンド

Image Builder の以前のバックエンド `lorax-composer` は非推奨とみなされます。Red Hat Enterprise Linux 8 の残りのライフサイクルでは一部の修正のみが提供され、今後のメジャーリリースから削除される予定です。Red Hat では、`lorax-composer` をアンインストールして `osbuild-composer` バックエンドを代わりにインストールすることを推奨します。

詳細は、[RHEL システムイメージのカスタマイズ](#) を参照してください。

(BZ#1893767)

5.6.2. ソフトウェア管理

rpmbuild --sign が非推奨になりました。

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

(BZ#1688849)

5.6.3. シェルおよびコマンドラインツール

curl の Metalink サポートが無効になりました。

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

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

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

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

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

以下に例を示します。

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

(BZ#1999620)

5.6.4. インフラストラクチャーサービス

mailman が非推奨に

今回の更新で、**mailman** パッケージが非推奨と識別され、Red Hat Enterprise Linux の将来のメジャーリリースで利用できなくなりました。

(BZ#1890976)

5.6.5. セキュリティー

NSS SEED 暗号が非推奨になりました。

Mozilla Network Security Services (**NSS**) ライブラリーでは、今後のリリースで SEED 暗号化を使用する TLS 暗号スイートのサポートがなくなります。NSS がサポートを削除した際に SEED 暗号に依存するデプロイメントを円滑に移行させるため、Red Hat は、他の暗号スイートのサポートを有効にすることを推奨します。

RHEL では、SEED 暗号はデフォルトですでに無効にされています。

(BZ#1817533)

TLS 1.0 および TLS 1.1 が非推奨になりました。

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

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

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

(BZ#1660839)

RHEL 8 で DSA が非推奨になりました。

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

(BZ#1646541)

NSS で SSL2 Client Hello が非推奨に

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

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

(BZ#1645153)

TPM 1.2 が非推奨になりました。

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

(BZ#1657927)

5.6.6. ネットワーク

RHEL 8 でネットワークスクリプトが非推奨に

Red Hat Enterprise Linux 8 では、ネットワークスクリプトが非推奨になっており、デフォルトでは提供されなくなりました。基本的なインストールでは、**nmcli** ツール経由で、**NetworkManager** サービスを呼び出す **ifup** スクリプトおよび **ifdown** スクリプトの新しいバージョンが提供されます。Red Hat Enterprise Linux 8 で **ifup** スクリプトおよび **ifdown** スクリプトを実行する場合は、**NetworkManager** を実行する必要があります。

/sbin/ifup-local、**ifdown-pre-local**、および **ifdown-local** の各スクリプトでは、カスタムコマンドが実行されません。

このスクリプトが必要な場合は、次のコマンドを使用すれば、システムに非推奨のネットワークスクリプトをインストールできます。

```
~]# yum install network-scripts
```

ifup スクリプトと **ifdown** スクリプトが、インストールされている従来のネットワークスクリプトにリンクされます。

従来のネットワークスクリプトを呼び出すと、そのスクリプトが非推奨であることを示す警告が表示されます。

(BZ#1647725)

5.6.7. カーネル

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

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

(BZ#1748980)

qla3xxx ドライバーが非推奨になりました。

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

(BZ#1658840)

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

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

(BZ#1660627)

rdma_rxe Soft-RoCE ドライバーが非推奨に

Remote Software Direct Memory Access over Converged Ethernet (Soft-RoCE) は RXE としても知られており、RDMA (Remote Direct Memory Access) をエミュレートする機能です。RHEL 8 では、Soft-RoCE 機能が、サポートされていないテクノロジープレビューとして利用できます。ただし、安定性の問題により、この機能は非推奨になり、RHEL 9 では削除されます。

(BZ#1878207)

5.6.8. ファイルシステムおよびストレージ

elevator カーネルコマンドラインパラメーターが非推奨になりました。

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

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

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

詳しい情報は、[ディスクスケジューラーの設定](#) を参照してください。

(BZ#1665295)

LVM mirror が非推奨化されました。

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

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

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

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

(BZ#1827628)

peripety が非推奨になりました。

peripety パッケージは、RHEL 8.3 以降で非推奨になりました。

Peripety ストレージイベント通知デーモンは、システムストレージログを構造化されたストレージイベントに解析します。ストレージの問題を調査するのに役立ちます。

(BZ#1871953)

NFSv3 over UDP が無効になりました。

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

RHEL 8 では、NFS over UDP に対応しなくなりました。

(BZ#1592011)

cramfs が非推奨になりました。

ユーザーの不足により、**cramfs** カーネルモジュールが非推奨になりました。代替策として **squashfs** が推奨されます。

(BZ#1794513)

5.6.9. Identity Management

openssh-ldap が非推奨に

openssh-ldap サブパッケージは、Red Hat Enterprise Linux 8 で非推奨になり、RHEL 9 で削除されます。**openssh-ldap** サブパッケージはアップストリームでは維持されないため、Red Hat は SSSD と **sss_ssh_authorizedkeys** ヘルパーを使用することを推奨しています。これは、他の IdM ソリューションよりも適切に統合でき、安全です。

デフォルトでは、**ldap** および **ipa** プロバイダーはユーザーオブジェクトの **sshPublicKey** LDAP 属性を読み取ります (利用可能な場合)。AD (Active Directory) には公開鍵を保存するためのデフォルトの LDAP 属性がないため、**ad** プロバイダーまたは IdM の信頼されるドメインのデフォルト SSSD 設定を使用して AD から SSH 公開鍵を取得することはできません。

sss_ssh_authorizedkeys ヘルパーが SSSD から鍵を取得できるようにするには、**sssd.conf** ファイルの **services** オプションに **ssh** を追加して **ssh** レスポンダーを有効にします。詳細は man ページの **sssd.conf(5)** を参照してください。

sshd が **sss_ssh_authorizedkeys** を使用できるようにするには、man ページの **sss_ssh_authorizedkeys(1)** に記載されているように、**AuthorizedKeysCommand /usr/bin/sss_ssh_authorizedkeys** および **AuthorizedKeysCommandUser nobody** オプションを **/etc/ssh/sshd_config** ファイルに追加します。

(BZ#1871025)

DES および 3DES 暗号化タイプが削除されました。

RHEL 7 以降、セキュリティ上の理由から、データ暗号化標準 (DES) アルゴリズムが非推奨になり、デフォルトで無効化になりました。Kerberos パッケージの最近のリベースで、RHEL 8 からシングル DES (DES) およびトリプル DES (3DES) の暗号化タイプが削除されました。

DES または 3DES の暗号化のみを使用するようにサービスまたはユーザーが設定されている場合、以下のようなサービスの中断が発生する可能性があります。

- Kerberos 認証エラー
- **unknown enctype** 暗号化エラー
- DES で暗号化されたデータベースマスターキー (K/M) を使用した KDC (Kerberos Distribution Center) が起動しない

アップグレードを準備するには、以下の操作を実施します。

1. KDC が **krb5check** オープンソース Python スクリプトで DES または 3DES 暗号化を使用しているかどうかを確認します。GitHub の [krb5check](#) を参照してください。
2. Kerberos プリンシパルで DES または 3DES 暗号化を使用している場合は、Advanced Encryption Standard (AES) などのサポート対象の暗号化タイプでキーを変更します。キー変更の手順については、MIT Kerberos ドキュメントの [Retiring DES](#) を参照してください。
3. アップグレードの前に以下の Kerberos オプションを一時的に設定して、DES および 3DES からの独立性をテストします。
 - a. KDC の **/var/kerberos/krb5kdc/kdc.conf** で、**supported_enctypes** を設定し、**des** または **des3** は含まれません。
 - b. すべてのホストについて、**/etc/krb5.conf** および **/etc/krb5.conf.d** のすべてのファイルで、**allow_weak_crypto** を **false** に設定します。デフォルトは **false** です。
 - c. すべてのホストについて、**/etc/krb5.conf** および **/etc/krb5.conf.d** のすべてのファイルで、**permitted_enctypes**、**default_tgs_enctypes**、**default_tkt_enctypes** を設定します。また、**des** または **des3** は含めません。

4. 前の手順で Kerberos 設定をテストしてサービスが中断されない場合は、サービスを削除してアップグレードします。最新の Kerberos パッケージにアップグレードした後は、この設定は必要ありません。

(BZ#1877991)

SMB1 プロトコルは Samba では非推奨に

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

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

(JIRA:RHELDPCS-16612)

5.6.10. デスクトップ

libgnome-keyring ライブラリーが非推奨になりました。

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

(BZ#1607766)

AlternateTab 拡張が削除される

AlternateTab GNOME Shell 拡張機能を提供する **gnome-shell-extension-alternate-tab** パッケージが削除されました。

window-switching 動作を設定するには、キーボード設定のキーボードショートカットを設定します。詳細は、[Gnome 3.32 以降での Alternate-Tab の使用](#) を参照してください。(BZ#1922488)

5.6.11. グラフィックインフラストラクチャー

AGP グラフィックカードがサポートされなくなりました。

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

(BZ#1569610)

5.6.12. Web コンソール

Web コンソールは、不完全な翻訳への対応を終了しました。

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

(BZ#1666722)

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

geoipupdate パッケージが非推奨に

geoipupdate パッケージにはサードパーティーのサブスクリプションが必要で、プロプライエタリーコンテンツもダウンロードします。したがって、**geoipupdate** パッケージは非推奨となり、次の RHEL メジャーバージョンで削除されます。

(BZ#1874892)

5.6.14. 仮想化

SPICE が非推奨になりました

SPICE リモートディスプレイプロトコルが非推奨になりました。その結果、SPICE は RHEL 8 でも引き続きサポートされますが、Red Hat はリモートディスプレイストリーミングに代替ソリューションを使用することを推奨しています。

- リモートコンソールへのアクセスには、VNC プロトコルを使用します。
- 高度なリモートディスプレイ機能には、RDP、HP RGS、または Mechdyne TGX などのサードパーティーツールを使用します。

SPICE で使用される **QXL** グラフィックスデバイスも非推奨になっていることに注意してください。

(BZ#1849563)

virt-manager が非推奨になりました。

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

(JIRA:RHELPLAN-10304)

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

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

新しい仮想マシンスナップのショットメカニズムは開発中で、RHEL 8 の将来のマイナーリリースで完全に実装される予定です。

(BZ#1686057)

Cirrus VGA 仮想 GPU タイプが非推奨に

Red Hat Enterprise Linux の将来のメジャー更新では、KVM 仮想マシンで **Cirrus VGA** GPU デバイスに対応しなくなります。したがって、Red Hat では、Cirrus VGA の代わりに **stdvga** または **virtio-vga** デバイスの使用を推奨しています。

(BZ#1651994)

5.6.15. コンテナ

Podman varlink ベースの REST API V1 が非推奨に

Podman varlink ベースの REST API V1 が非推奨になり、Podman varlink ベースの REST API

Podman varlink ベースの REST API V1 がアップストリームで非推奨となり、新しい Podman REST API V2 が使用されるようになりました。この機能は、Red Hat Enterprise Linux 8 の今後のリリースで削除されます。

(JIRA:RHELPLAN-60226)

5.6.16. 非推奨のパッケージ

以下のパッケージは非推奨となり、Red Hat Enterprise Linux の今後のメジャーリリースには含まれません。

- 389-ds-base-legacy-tools
- authd
- custodia
- hostname
- libidn
- lorax-composer
- mercurial
- net-tools
- network-scripts
- nss-pam-ldapd
- sendmail
- yp-tools
- ypbind
- ypserv

5.7. 既知の問題

このパートでは Red Hat Enterprise Linux 8.3 の既知の問題を説明します。

5.7.1. インストーラーおよびイメージの作成

キックスタートコマンドの `auth` および `authconfig` で AppStream リポジトリーが必要になる

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 リポジトリーでしか利用できません。

この問題を回避するには、BaseOS リポジトリーおよび AppStream リポジトリーがインストーラーで利用できることを確認するか、インストール中にキックスタートコマンドの `authselect` コマンドを使用します。

(BZ#1640697)

reboot --kexec コマンドおよび inst.kexec コマンドが、予測可能なシステム状態を提供しない

キックスタートコマンド **reboot --kexec** またはカーネル起動パラメーター **inst.kexec** で RHEL インストールを実行しても、システムの状態が完全な再起動と同じになるわけではありません。これにより、システムを再起動せずにインストール済みのシステムに切り替えると、予期しない結果が発生することがあります。

kexec 機能は非推奨になり、Red Hat Enterprise Linux の今後のリリースで削除されることに注意してください。

(BZ#1697896)

インストールプログラムでは、ネットワークアクセスがデフォルトで有効になっていません。

一部のインストール機能、たとえば、コンテンツ配信ネットワーク (CDN) を使用したシステムの登録、NTP サーバーサポート、およびネットワークインストールソースなどには、ネットワークアクセスが必要です。ただし、ネットワークアクセスはデフォルトでは有効になっていません。そのためこの機能は、ネットワークアクセスが有効になるまで使用できません。

この問題を回避するには、インストールの開始時にネットワークアクセスを有効にする起動オプション **ip=dhcp** を追加します。オプションで、起動オプションを使用して、ネットワーク上にあるキックスタートファイルまたはリポジトリを渡しても、問題が解決されます。結果として、ネットワークベースのインストール機能を使用できます。

(BZ#1757877)

新しい osbuild-composer バックエンドが、アップグレード時に lorax-composer から Blueprint 状態に複製されない。

lorax-composer バックエンドから新しい **osbuild-composer** バックエンドにアップグレードする Image Builder ユーザーは、Blueprint が消えてしまう可能性があります。その結果、アップグレードが完了すると、Blueprint が自動的に表示されなくなります。この問題を回避するには、以下の手順を実行します。

前提条件

- **composer-cli** CLI ユーティリティーがインストールされている。

手順

1. 以下のコマンドを実行して、以前の **lorax-composer** ベースの Blueprint を新しい **osbuild-composer** バックエンドに読み込みます。

```
$ for blueprint in $(find /var/lib/lorax/composer/blueprints/git/workspace/master -name
*.toml); do composer-cli blueprints push "${blueprint}"; done
```

これにより、**osbuild-composer** バックエンドで同じ Blueprint が利用できるようになりました。

関連情報

- この既知の問題の詳細は、[Image Builder Blueprint が、Red Hat Enterprise Linux 8.3 への更新後に表示されなくなる](#) 参照してください。

(BZ#1897383)

Kickstart インストールでは、自己署名 HTTPS サーバーを使用できません。

現在では、インストールソースが Kickstart ファイルで指定され、`--noverifyssl` オプションが指定されると、インストーラーは自己署名の https サーバーからのインストールに失敗します。

```
url --url=https://SERVER/PATH --noverifyssl
```

この問題を回避するには、Kickstart インストールの開始時に、`inst.noverifyssl` パラメーターをカーネルコマンドラインに追加します。

以下に例を示します。

```
inst.ks=<URL> inst.noverifyssl
```

(BZ#1745064)

リポジリーの更新が完了する前に CDN を使用した登録解除を試みると、GUI インストールが失敗することがあります。

RHEL 8.2 以降、システムを登録し、コンテンツ配信ネットワーク (CDN) を使用してサブスクリプションを割り当てると、GUI インストールプログラムにより、リポジリーメタデータの更新が開始されます。更新プロセスは、登録およびサブスクリプションプロセスの一部ではないため、Red Hat への接続ウィンドウで **登録解除** ボタンが有効になります。ネットワーク接続によっては、更新プロセスが完了するのに1分以上かかることがあります。更新プロセスが完了する前に **登録解除** ボタンをクリックすると、登録解除プロセスで、インストールプログラムが CDN との通信に必要とする証明書と CDN リポジリーファイルが削除されるため、GUI インストールが失敗する可能性があります。

この問題を回避するには、Red Hat への接続ウィンドウで **登録** ボタンをクリックした後に、GUI インストールで次の手順を実行します。

1. Red Hat への接続画面から **完了** をクリックして、インストールの概要画面に戻ります。
2. インストールの概要 ウィンドウで、インストールソース および ソフトウェアの選択 の斜体のステータスメッセージに処理情報が表示されていないことを確認します。
3. インストールソースとソフトウェアの選択のカテゴリが準備できたら、Red Hat への接続 をクリックします。
4. **登録解除** ボタンをクリックします。

これらの手順を完了したら、GUI のインストール時にシステムの登録を安全に解除できます。

(BZ#1821192)

複数の組織に属するユーザーアカウントの登録に失敗していました

現在、複数の組織に属するユーザーアカウントでシステムを登録しようとする時、登録プロセスが失敗し、**You must specify an organization for new units**(新しいユニットの組織を指定する必要があります)。というメッセージが表示されます。

この問題を回避するには、以下のいずれかを行います。

- 複数の組織に属さない別のユーザーアカウントを使用します。
- GUI および Kickstart インストールの Connect to Red Hat 機能で利用できる **アクティベーションキー** 認証方法を使用します。
- Connect to Red Hat の登録手順を省略し、Subscription Manager を使用してインストール後にシステムを登録します。

(BZ#1822880)

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

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

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

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

この問題を回避するには、Lorax ツールを使用して、更新された Anaconda パッケージおよび NetworkManager パッケージを含む新しいインストールメディアを作成します。

Lorax ツールを使用して、更新された Anaconda パッケージおよび NetworkManager パッケージを含む新しいインストールメディアを作成する方法は、[Unable to install Red Hat Enterprise Linux 8.3.0 with InfiniBand network interfaces](#) を参照してください。

(BZ#1890261)

NVDIMM デバイス名前空間が devdax モードに設定されていると、Anaconda のインストールに失敗します。

GUI インストールの前に、NVDIMM デバイス名前空間が **devdax** モードに設定した状態で起動すると、Anaconda のインストールに失敗します。

この問題を回避するには、インストールを開始する前に、NVDIMM デバイスを再設定し、名前空間を **devdax** モードとは異なるモードに設定します。これにより、インストールを続行できます。

(BZ#1891827)

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 **inst.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 デバイスを作成し、最初に Linux システムで RHEL ISO イメージを再生成すると、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、[RHEL 8.3 のインストール時にインストールメディアは自動検出されない](#) を参照してください。

(BZ#1877697)

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

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

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

(BZ#1874394)

グラフィカルインストーラーの使用時に、Red Hat Insights クライアントがオペレーティングシステムの登録に失敗する

現在、Insights クライアントを示すエラーでインストールに失敗します。

この問題を回避するには、インストーラーでシステムを登録する前に、**Connect to Red Hat** ステップで、**Connect to Red Hat Insights** プシジョンの選択を解除します。

その結果、以下のコマンドを使用してインストールを完了し、その後 Insights に登録できます。

```
# insights-client --register
```

(BZ#1931069)

5.7.2. サブスクリプション管理

syspurpose addons は **subscription-manager attach --auto** 出力に影響しません。

Red Hat Enterprise Linux 8 では、**syspurpose** コマンドラインツールの 4 つの属性 (**role**、**usage**、**service_level_agreement**、および **addons**) が追加されました。現在、**role**、**usage**、および **service_level_agreement** のみが、**subscription-manager attach --auto** コマンドの実行の出力に影響します。**addons** 引数に値を設定しても、自動登録されたサブスクリプションには影響がありません。

(BZ#1687900)

5.7.3. インフラストラクチャーサービス

dnf update の実行時に **libmaxminddb-devel-debuginfo.rpm** が削除される

dnf update コマンドを実行すると、バイナリー **mmdblookup** ツールは **libmaxminddb-devel** サブパッケージからメインの **libmaxminddb** パッケージに移動します。これにより、**libmaxminddb-devel-debuginfo.rpm** が削除され、このパッケージに破損した更新パスが作成される可能性があります。この問題を回避するには、**dnf update** コマンドの実行前に **libmaxminddb-devel-debuginfo** を削除します。

注記: **libmaxminddb-debuginfo** は新しい **debuginfo** パッケージです。

(BZ#1642001)

5.7.4. セキュリティー

ユーザーは、ロックされたユーザーとして **sudo** コマンドを実行できます。

ALL キーワードで **sudoers** パーミッションが定義されているシステムでは、パーミッションを持つ **sudo** ユーザーは、アカウントがロックされているユーザーとして **sudo** コマンドを実行できます。そのため、ロックされたアカウントと期限切れのアカウントを使用して、コマンドを実行し続けることができます。

この問題を回避するには、**/etc/shells** 内の有効なシェルの適切な設定と併せて、新たに実装した **runas_check_shell** オプションを有効にします。これにより、攻撃者が **bin** などのシステムアカウントでコマンドを実行するのを防ぎます。

(BZ#1786990)

GnuTLS が NSS サーバーとの現行セッションを再開できません。

TLS (Transport Layer Security) 1.3 セッションを再開するとき、**GnuTLS** クライアントは 60 ミリ秒に加えて、サーバーがセッション再開データを送信するために推定されるラウンドトリップタイムの分だけ待機します。サーバーがこの時間内に再開データを送信しないと、クライアントは現行セッションを再開せずに新しいセッションを作成します。これは、通常のセッションネゴシエーションのパフォーマンスに若干影響することを除いて、深刻な悪影響を与えることはありません。

(BZ#1677754)

libselinux-python は、そのモジュールからのみ利用可能

libselinux-python パッケージには、SELinux アプリケーション開発用の Python 2 バインディングのみが含まれ、後方互換性に使用されます。このため、**libselinux-python** コマンドを使用して、デフォルトの RHEL 8 リポジトリで **dnf install libselinux-python** コマンドが利用できなくなりました。

この問題を回避するには、**libselinux-python** モジュールおよび **python27** モジュールの両方を有効にし、以下のコマンドで **libselinux-python** パッケージとその依存関係をインストールします。

```
# dnf module enable libselinux-python
# dnf install libselinux-python
```

または、1つのコマンドでインストールプロファイルを使用して **libselinux-python** をインストールします。

```
# dnf module install libselinux-python:2.8/common
```

これにより、各モジュールを使用して **libselinux-python** をインストールできます。

(BZ#1666328)

udica は、--env container=podman で開始したときにのみ UBI 8 コンテナを処理します。

Red Hat Universal Base Image 8 (UBI 8) コンテナは、**podman** の値ではなく、コンテナ 環境変数を **oci** 値に設定します。これにより、**udica** ツールがコンテナ JavaScript Object Notation (JSON) ファイルを分析しなくなります。

この問題を回避するには、**--env container=podman** パラメーターを指定して、**podman** コマンドで UBI 8 コンテナを起動します。そのため、**udica** は、上記の回避策を使用している場合に限り、UBI 8 コンテナの SELinux ポリシーを生成することができます。

(BZ#1763210)

デフォルトのロギング設定がパフォーマンスに与える悪影響

デフォルトのログ環境設定は、メモリーを 4 GB 以上使用する可能性があり、**rsyslog** で **systemd-journald** を実行している場合は、速度制限値の調整が複雑になります。

詳細は、ナレッジベースの記事 [Negative effects of the RHEL default logging setup on performance and their mitigations](#) を参照してください。

(JIRA:RHELPLAN-10431)

/etc/passwd- のファイル権限が CIS RHEL 8 Benchmark 1.0.0 と合致しない

CIS Benchmark の問題により、**/etc/passwd-** バックアップファイルの権限を保証する SCAP ルールの修正によって、権限が **0644** に設定されます。ただし、**CIS Red Hat Enterprise Linux 8 Benchmark 1.0.0** では、そのファイルに対するファイルパーミッション **0600** が必要です。そのため、修正後、**/etc/passwd-** のファイル権限はベンチマークに合うように設定されません。

([BZ#1858866](#))

/etc/selinux/config の SELINUX=disabled が正常に動作しません。

/etc/selinux/config で **SELINUX=disabled** オプションを使用して SELinux を無効にすると、カーネルが SELinux を有効にして起動し、その後のブートプロセスで無効化モードに切り替わります。これにより、メモリーリークが生じる可能性があります。

この問題を回避するには、SELinux を完全に無効にする必要がある場合に [SELinux の使用のシステムの起動時に SELinux モードの変更](#) で説明されているように、**selinux=0** パラメーターをカーネルコマンドラインに追加して SELinux を無効にすることが推奨されます。

(JIRA:RHELPLAN-34199)

ssh-keyscan が、FIPS モードでサーバーの RSA 鍵を取得できません。

FIPS モードで RSA 署名の **SHA-1** アルゴリズムが無効になっています。これにより、**ssh-keyscan** ユーティリティーがそのモードで稼働しているサーバーの RSA 鍵を取得できなくなります。

この問題を回避するには、代わりに ECDSA 鍵を使用するか、サーバーの **/etc/ssh/ssh_host_rsa_key.pub** ファイルから鍵をローカルに取得します。

([BZ#1744108](#))

OpenSSL が、生の RSA または RSA-PSS の署名に対応していない PKCS #11 トークンを誤って処理します。

OpenSSL ライブラリーは、PKCS #11 トークンの鍵関連の機能を検出しません。したがって、生の RSA または RSA-PSS の署名に対応しないトークンで署名が作成されると、TLS 接続の確立に失敗します。

この問題を回避するには、**/etc/pki/tls/openssl.cnf** ファイルの **crypto_policy** セクションの末尾にある **.include** 行の後に、以下の行を追加します。

```
SignatureAlgorithms =
RSA+SHA256:RSA+SHA512:RSA+SHA384:ECDSA+SHA256:ECDSA+SHA512:ECDSA+SHA384
MaxProtocol = TLSv1.2
```

これにより、このシナリオで TLS 接続を確立できます。

([BZ#1685470](#))

FIPS モードの OpenSSL が、特定の D-H パラメーターのみを受け入れます。

FIPS モードでは、OpenSSL を使用するトランスポートレイヤーセキュリティ (TLS) クライアントが **bad dh value** エラーを返し、手動で生成されたパラメーターを使用したサーバーへの TLS 接続を中止します。これは、FIPS 140-2 に準拠するよう設定されている場合、OpenSSL が NIST SP 800-56A rev3 付録 D (RFC 3526 で定義されたグループ 14、15、16、17、18、および RFC 7919 で定義されたグループ) に準拠した D-H パラメーターでのみ機能するためです。また、OpenSSL を使用するサーバーは、その他のパラメーターをすべて無視し、代わりに同様のサイズの既知のパラメーターを選択します。この問題を回避するには、準拠するグループのみを使用します。

(BZ#1810911)

rpm-plugin-selinux パッケージを削除すると、システムからすべての **selinux-policy** パッケージが削除されます。

rpm-plugin-selinux パッケージを削除すると、マシン上で SELinux が無効になります。また、システムからすべての **selinux-policy** パッケージも削除されます。**rpm-plugin-selinux** パッケージを繰り返しインストールしてから、**selinux-policy-targeted** ポリシーが以前に存在していても、**selinux-policy-minimum** SELinux ポリシーをインストールします。ただし、繰り返しインストールしても、ポリシーの変更のために SELinux 設定ファイルが更新されることはありません。このため、**rpm-plugin-selinux** パッケージを再インストールしても、SELinux が無効になります。

この問題を回避するには、以下を実行します。

1. **umount /sys/fs/selinux/** コマンドを実行します。
2. 足りない **selinux-policy-targeted** パッケージを手動でインストールします。
3. ポリシーが **SELINUX=enforcing** と同等になるように **/etc/selinux/config** ファイルを編集します。
4. コマンド **load_policy -i** を実行します。

これにより、SELinux が有効になり、以前と同じポリシーが実行されている状態になります。

(BZ#1641631)

systemd サービスが任意のパスからコマンドを実行できない

SELinux ポリシーパッケージにはこのようなルールが含まれていないため、**systemd** サービスは **/home/user/bin** の任意のパスからコマンドを実行できません。そのため、システム以外のパスで実行されるカスタムサービスは失敗し、SELinux がアクセスを拒否すると、AVC (アクセスベクターキャシュ) 監査メッセージをログに記録します。この問題を回避するには、以下のいずれかを実行します。

- **-c** オプションを指定し、シェル スクリプトを使用してコマンドを実行します。以下に例を示します。

```
bash -c command
```

- **/bin**、**/sbin**、**/usr/sbin**、**/usr/local/bin**、**/usr/local/sbin** の共通のディレクトリーを使用して共通のパスからコマンドを実行します。

(BZ#1860443)

CIS プロファイルで **rpm_verify_permissions** が失敗する

rpm_verify_permissions ルールでは、ファイルパーミッションがパッケージのデフォルトパーミッションと比較されます。ただし、**scap-security-guide** パッケージで提供される Center for Internet Security (CIS) プロファイルでは、一部のファイルパーミッションがデフォルトよりも厳格なものに変

更されます。その結果、**rpm_verify_permissions** を使用した特定ファイルの検証が失敗します。

この問題を回避するには、これらのファイルに以下のパーミッションがあることを手作業で確認します。

- **/etc/cron.d** (0700)
- **/etc/cron.hourly** (0700)
- **/etc/cron.monthly** (0700)
- **/etc/crontab** (0600)
- **/etc/cron.weekly** (0700)
- **/etc/cron.daily** (0700)

(BZ#1843913)

RHEL 8 のキックスタートが、**com_redhat_oscaps** の代わりに **org_fedora_oscaps** を使用

キックスタートは、**com_redhat_oscaps** ではなく、**org_fedora_oscaps** として Open Security Content Automation Protocol (OSCAP) Anaconda アドオンを参照します。これが、混乱を招く可能性があります。これは、Red Hat Enterprise Linux 7 との後方互換性を維持するために行われます。

(BZ#1665082)

SSG における相互依存ルールの特定のセットが失敗する可能性があります。

ルールとその依存関係の順序付けを定義しないため、ベンチマークの **SCAP Security Guide (SSG)** ルールの修正が失敗する可能性があります。たとえば、特定の順番で複数のルールを実行する必要がある場合、あるルールがコンポーネントをインストールし、別のルールが同じコンポーネントを設定した場合すると、それらは正しくない順序で実行される可能性があり、修正によってエラーが報告されます。この問題を回避するには、修正を再実行して、番目の実行で依存ルールを修正します。

(BZ#1750755)

OSCAP Anaconda Addon がすべてのパッケージをテキストモードでインストールしません。

OSCAP Anaconda Addon プラグインは、インストールがテキストモードで実行している場合、システムインストーラーによってインストールに選択されているパッケージのリストを変更することはできません。これにより、キックスタートを使用してセキュリティーポリシープロファイルが指定され、インストールがテキストモードで実行している場合に、インストール中にセキュリティーポリシーに必要な追加パッケージがインストールされません。

この問題を回避するには、グラフィカルモードでインストールを実行するか、キックスタートファイルの **%packages** セクションにあるセキュリティーポリシーで、セキュリティーポリシープロファイルに必要なパッケージをすべて指定します。

これにより、セキュリティーポリシープロファイルで必要となるパッケージは、上記の回避策のいずれかを行わなければ RHEL インストールインストール時にインストールされません。また、インストール後のシステムは、指定のセキュリティーポリシープロファイルと互換性がありません。

(BZ#1674001)

oscaps Anaconda Addon がカスタムプロファイルを正しく処理しません。

OSCAP Anaconda Addon プラグインは、個別のファイルでカスタマイズを使用したセキュリティープロファイルを適切に処理しません。これにより、対応する Kickstart セクションで適切に指定しても、RHEL グラフィカルインストールでカスタマイズしたプロファイルは利用できません。

この問題を回避するには、ナレッジベースの記事 [Creating a single SCAP data stream from an original DS and a tailoring file](#) を参照してください。この回避策により、RHEL グラフィカルインストールでカスタマイズした SCAP プロファイルを使用できます。

(BZ#1691305)

OSPP ベースのプロファイルに、GUI パッケージグループとの互換性がありません。

Server with GUI パッケージグループでインストールされる **GNOME** パッケージには、Operating System Protection Profile (OSPP) に準拠しない **nfs-utils** パッケージが必要です。これにより、OSPP または OSPP ベースのプロファイル (Security Technical Implementation Guide (STIG) など) を使用したシステムのインストール時に **Server with GUI** パッケージグループを選択すると、OpenSCAP により選択したパッケージグループがセキュリティーポリシーと互換性がありませんといった警告が表示されます。OSPP ベースのプロファイルがインストール後に適用される場合、システムは起動できません。この問題を回避するには、**Server with GUI** パッケージグループや、または OSPP プロファイルと OSPP ベースのプロファイルを使用する際に GUI をインストールするその他のグループをインストールしないでください。代わりに **Server** または **Minimal Install** パッケージグループを使用すると、システムは問題なくインストールされ、正常に機能します。

(BZ#1787156)

Server with GUI または Workstation ソフトウェアの選択と CIS セキュリティープロファイルを使用したインストールはできません。

CIS セキュリティープロファイルは、**Server with GUI** および **Workstation** ソフトウェアの選択と互換性がありません。そのため、**Server with GUI** ソフトウェアの選択と CIS プロファイルを使用した RHEL 8 のインストールはできません。CIS プロファイルと、これらのソフトウェアの選択のいずれかを使用したインストール試行では、エラーメッセージが生成されます。

```
package xorg-x11-server-common has been added to the list of excluded packages, but it can't be removed from the current software selection without breaking the installation.
```

この問題を回避するには、**Server with GUI** または **Workstation** ソフトウェアの選択で CIS セキュリティープロファイルを使用しないでください。

(BZ#1843932)

キックスタートインストール時のサービス関連のルールの修正が失敗する場合があります。

キックスタートのインストール時に、OpenSCAP ユーティリティーで、サービス **enable** または **disable** 状態の修正が必要でないことが誤って表示されることがあります。これにより、OpenSCAP が、インストール済みシステムのサービスを非準拠状態に設定する可能性があります。回避策として、キックスタートインストール後にシステムをスキャンして修復できます。これにより、サービス関連の問題が修正されます。

(BZ#1834716)

特定の rsyslog 優先度の文字列が正常に動作しません。

imtcp に **GnuTLS** 優先度文字列を設定して、完成していない暗号化をきめ細かく制御できるようになりました。したがって、**rsyslog** では、以下の優先文字列が正常に動作しません。


```
NONE:+VERS-ALL:-VERS-TLS1.3:+MAC-ALL:+DHE-RSA:+AES-256-GCM:+SIGN-RSA-
SHA384:+COMP-ALL:+GROUP-ALL
```

この問題を回避するには、正しく機能する優先度文字列のみを使用します。

```
NONE:+VERS-ALL:-VERS-TLS1.3:+MAC-ALL:+ECDHE-RSA:+AES-128-CBC:+SIGN-RSA-
SHA1:+COMP-ALL:+GROUP-ALL
```

したがって、現在の設定は、正しく機能する文字列に限定する必要があります。

([BZ#1679512](#))

crypto-policies が Camellia 暗号を誤って許可する。

RHEL 8 システム全体の暗号化ポリシーでは、製品ドキュメントで説明されているように、すべてのポリシーレベルで Camellia 暗号を無効にする必要があります。ただし、Kerberos プロトコルでは、デフォルトでこの Camellia 暗号が有効になります。

この問題を回避するには、**NO-CAMELLIA** サブポリシーを適用します。

```
# update-crypto-policies --set DEFAULT:NO-CAMELLIA
```

これまでに上記のコマンドで、**DEFAULT** から切り替えたことがある場合は、**DEFAULT** を暗号化レベルの名前に置き換えます。

その結果、この回避策を使用して Camellia 暗号を無効にしている場合に限り、システム全体の暗号化ポリシーを使用する全ポリシーで、この暗号化を適切に拒否できます。(BZ#1919155)

5.7.5. ネットワーク

iptables ユーティリティーは、**NLM_F_CREATE** フラグに関係なく、チェーンを更新するコマンドに対してモジュールの読み込みを要求するようになりました。

以前では、チェーンのポリシーを設定する際に、**iptables-nft** ユーティリティーは **NEWCHAIN** メッセージを生成しましたが、**NLM_F_CREATE** フラグを設定しませんでした。これにより、RHEL 8 カーネルはモジュールを読み込みず、関連付けられたカーネルモジュールが手動で読み込まれなかった場合に、生成される更新チェーンコマンドに失敗していました。今回の更新で、**iptables-nft** ユーティリティーは、チェーンを更新するすべてのコマンドに対してモジュールの読み込みを要求するようになり、関連するモジュールを手動で読み込むことなく、**iptables-nft** ユーティリティーを使用してチェーンのポリシーを設定できるようになりました。

(BZ#1812666)

カーネルでの **packet/byte** カウンターの更新に対するサポートが、RHEL 7 と RHEL 8 の間で誤って変更されました。

iptables ルールからの有効なカウンターを持つ **ipset** コマンドを参照する場合、一致する **ipset** エントリに追加の制約を指定する場合には、**ipset** カウンターはすべて追加の制約が一致する場合にのみ更新されます。**--packets-gt** または **--bytes-gt** の制約でも問題があります。

したがって、**iptables** ルールセットを RHEL 7 から RHEL 8 に移行すると、**ipset** ルックアップに関連するルールが機能しなくなり、調整する必要がある場合があります。この問題を回避するには、**--packets-gt** オプションまたは **--bytes-gt** オプションの使用を回避し、それらを **--packets-lt** オプションまたは **--bytes-lt** オプションに置き換えてください。

(BZ#1806882)

nfp ドライバーを使用する Netronome ネットワークカードで XDP プログラムのアンロードに失敗する

Netronome ネットワークカードの **nfp** ドライバーにはバグがあります。したがって、このようなカードを使用し、**XDP_FLAGS_REPLACE** フラグとともに **IFLA_XDP_EXPECTED_FD** 機能を使用して XDP プログラムを読み込むと、eXpress Data Path (XDP) プログラムのアンロードに失敗します。たとえば、このバグは、**libxdp** ライブラリーを使用して読み込まれた XDP プログラムに影響します。現在、この問題に対する回避策はありません。

(BZ#1880268)

ip 起動オプションで DHCP を使用する場合に Anaconda にネットワークアクセスがない

初期 RAM ディスク (**initrd**) は、NetworkManager を使用してネットワークを管理します。RHEL 8.3 ISO ファイルが提供する **dracut** NetworkManager モジュールは誤って、Anaconda 起動オプションの **ip** オプションの最初のフィールドが常に設定されていると仮定していました。そのため、DHCP を使用して **ip=:::;<host_name>::dhcp** を設定すると、NetworkManager は IP アドレスを取得せず、Anaconda でネットワークを利用できませんでした。

この問題を回避するには、以下のオプションを使用できます。

1. **ip`option to `.** (ピリオド) の最初のフィールドを定めます。

```
ip=.:::;<host_name>::dhcp
```

この回避策は、問題が修正された場合に RHEL の今後のバージョンでは機能しないことに注意してください。

2. バグの修正が含まれる BaseOS リポジトリから最新のパッケージを使用して、**boot.iso** ファイルを再作成します。

```
# lorax '--product=Red Hat Enterprise Linux' --version=8.3 --release=8.3 \
--source=<URL_to_BaseOS_repository> \
--source=<URL_to_AppStream_repository> \
--nomacboot --buildarch=x86_64 '--volid=RHEL 8.3' <output_directory>
```

.Red Hat は、自己作成された ISO ファイルをサポートしていないことに注意してください。

これにより、RHEL は DHCP サーバーから IP アドレスを取得し、Anaconda でネットワークアクセスを利用することができます。

(BZ#1902791)

5.7.6. カーネル

tboot-1.9.12-2 ユーティリティーを使用すると、RHEL 8 でシステムが起動できない

バージョン 1.9.12-2 の **tboot** ユーティリティーにより、TPM(Trusted Platform Module)2.0 の一部のシステムがレガシーモードでの起動に失敗します。これにより、tboot Grand Unified Bootloader(GRUB) エントリーから起動を試みると、システムは停止します。この問題を回避するには、バージョン 1.9.10 の **tboot** にダウングレードします。

(BZ#1947839)

カーネルが IBM Z システムで誤検出の警告を返す

RHEL 8 では、IBM Z システムで **ZONE_DMA** メモリーゾーンのホワイトリストエントリーが含まれており、ユーザーアクセスが許可されます。したがって、カーネルは以下のような誤検出の警告を返します。

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

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

この問題を回避するには、**hardened_usercopy=off** パラメーターをカーネルコマンドラインに追加します。

これにより、上記のシナリオで警告メッセージが表示されなくなります。

(BZ#1660290)

rngd サービスのビジー待機により、FIPS モードで CPU が完全に消費される

バージョン 4.18.0-193.10 で始まるカーネル用に、FIPS モードの新しいカーネルエントロピーソースが追加されました。その結果、FIPS モードでは、**rngd** サービスのビジー状態が **/dev/random** デバイスの **poll()** システムコールを待つため、CPU 時間が 100% 消費していました。この問題を回避するには、以下を実行して **rngd** を停止して無効にします。

```
# systemctl stop rngd
# systemctl disable rngd
```

その結果、**rngd** は上記のシナリオで **poll()** でビジー待機しなくなりました。

(BZ#1884857)

softirq の変更により、負荷が大きい場合に localhost インターフェイスが UDP パケットをドロップできるようになります

Linux カーネルのソフトウェア割り込み (**softirq**) 処理の変更を行い、サービス拒否 (DOS) の影響を軽減します。その結果、これにより、localhost インターフェイスが負荷が大きいときに localhost インターフェイスが UDP (User Datagram Protocol) パケットをドロップする状況が生じます。

この問題を回避するには、ネットワークデバイスのバックログバッファのサイズを 6000: の値に増やします。

```
echo 6000 > /proc/sys/net/core/netdev_max_backlog
```

Red Hat テストでは、パケットロスを防ぐために十分な値でした。より負荷の高いシステムには、より大きなバックログ値が必要になる場合があります。バックログが増えると、ローカルホストインターフェイスのレイテンシーが増加する可能性があります。

その結果、バッファを増やし、より多くのパケットを処理まで待機できるため、ローカルホストパケットをドロップする可能性を減らすことができます。

(BZ#1779337)

vmcore キャプチャーはメモリーのホットプラグまたはアンプラグの操作を実行した後に失敗します。

メモリーのホットプラグまたはホットアンプラグ操作の実行後に、メモリーのレイアウト情報を含むデバイスツリーを更新するとイベントが発生します。これにより、**makedumpfile** ユーティリティーは存在しない物理アドレスにアクセスしようとします。以下の条件を満たすと問題が発生します。

- IBM Power System (little endian) で RHEL 8 を実行する。
- システムで **kdump** サービスまたは **fadump** サービスが有効になっている。

このような場合に、メモリーホットプラグまたはホットアンプラグの操作後にカーネルクラッシュが発生すると、カーネルのキャプチャーで **vmcore** の保存に失敗します。

この問題を回避するには、ホットプラグまたはホットアンプラグ後に **kdump** サービスを再起動します。

```
# systemctl restart kdump.service
```

これにより、上記のシナリオで **vmcore** が正常に保存されます。

(BZ#1793389)

irqpoll を使用すると **vmcore** の生成に失敗します。

Amazon Web Services (AWS) クラウドプラットフォームで実行している 64 ビット ARM アーキテクチャー上には **nvme** ドライバーの既存の問題があります。この問題により、最初のカーネルに **irqpoll** カーネルコマンドラインパラメーターを指定すると **vmcore** の生成に失敗します。したがって、カーネルクラッシュ後に **vmcore** が **/var/crash/** ディレクトリーにダンプされません。この問題を回避するには、以下を実行します。

1. **/etc/sysconfig/kdump** ファイルの **KDUMP_COMMANDLINE_REMOVE** キーに **irqpoll** を追加します。
2. **systemctl restart kdump** コマンドを実行して、**kdump** サービスを再起動します。

その結果、最初のカーネルが正常に起動し、カーネルクラッシュ時に **vmcore** がキャプチャーされることが予想されます。

kdump サービスは、大量のクラッシュカーネルメモリーを使用して **vmcore** ファイルをダンプできることに注意してください。キャプチャーカーネルには、**kdump** サービス用のメモリーが十分あることを確認します。

(BZ#1654962)

RHEL 8 で、**デバッグカーネルがクラッシュキャプチャー環境で起動に失敗します。**

デバッグカーネルのメモリー要求の性質により、デバッグカーネルが使用中で、カーネルパニックが発生すると、問題が発生します。その結果、デバッグカーネルはキャプチャーカーネルとして起動できず、代わりにスタックトレースが生成されます。この問題を回避するには、クラッシュカーネルメモリーを適宜増やします。これにより、デバッグカーネルが、クラッシュキャプチャー環境で正常に起動します。

(BZ#1659609)

zlib は、一部の圧縮機能で **vmcore** キャプチャーの速度を低下させる可能性があります。

kdump 設定ファイルはデフォルトで、**lzo** 圧縮形式 (**makedumpfile -l**) を使用します。**zlib** 圧縮形式

(**makedumpfile -c**) を使用して設定ファイルを変更すると、**vmcore** のキャプチャープロセスの速度を低下させる代わりに、圧縮の因子が改善される可能性が高くなっています。これにより、**lzo** と比較して、**kdump** が **zlib** で **vmcore** をキャプチャーするのに最大 4 倍の時間がかかります。

このように、Red Hat は、速度が主要な要因である場合に、デフォルトの **lzo** を使用することを推奨します。ただし、ターゲットマシンで利用可能な領域が少ない場合は、**zlib** の方が適しています。

(BZ#1790635)

HP NMI ウォッチドッグが常にクラッシュダンプを生成しない

特定の場合において、HP NMI ウォッチドッグの **hpwdt** ドライバーは、マスク不可割り込み (NMI) が **perfmon** ドライバーにより使用されたため、HPE ウォッチドッグタイマーが生成した NMI を要求できません。

欠落している NMI は、以下の 2 つの条件のいずれかによって開始されます。

1. Integrated Lights-Out (iLO) サーバー管理ソフトウェアの **NMI 生成** ボタン。このボタンはユーザーがトリガーします。
2. **hpwdt** ウォッチドッグ。デフォルトでは、有効期限により NMI がサーバーに送信されます。

通常、両方のシーケンスは、システムが応答しない場合に発生します。通常、これらの状況の NMI ハンドラーは **kernel panic()** 関数を呼び出します。また、設定されていれば、**kdump** サービスが **vmcore** ファイルを生成します。

ただし、NMI が見つからないため、**kernel panic()** は呼び出されず、**vmcore** が収集されません。

最初のケース (1.) でシステムが応答しない場合は、その状態のままになります。このシナリオを回避するには、仮想 **電源** ボタンを使用してサーバーをリセットするか、電源を切って入れ直します。

2 つ目のケース (2.) では、欠落している NMI が Automated System Recovery (ASR) からのリセットの後 9 秒後に続きます。

HPE Gen9 Server ラインでは、1 桁台の割合でこの問題が発生します。Gen10 の周波数がさらに小さくなる。

(BZ#1602962)

tuned-adm profile powersave コマンドを使用すると、システムが応答しなくなります。

tuned-adm profile powersave コマンドを実行すると、古い Thunderx (CN88xx) プロセッサを持つ Penguin Valkyrie 2000 2 ソケットシステムが応答しなくなります。これにより、作業を再開するためシステムを再起動することになります。この問題を回避するには、システムが上記の仕様と一致する場合には **powersave** プロファイルの使用を避けてください。

(BZ#1609288)

デフォルトの 7417 printk 値により、一時的なシステムが応答しなくなることがあります。

デフォルトの 7417 **printk** 値を使用することで、カーネルアクティビティのデバッグを改善できます。ただし、シリアルコンソールと組み合わせると、この **printk** 設定により、RHEL システムが一時的に応答しなくなるような激しい I/O がバーストする可能性があります。この問題を回避するには、新しい **optimize-serial-console** TuneD プロファイルを追加し、デフォルトの **printk** 値を 4417 に減らします。ユーザーは、以下のようにシステムをインストルメント化できます。

```
# tuned-adm profile throughput-performance optimize-serial-console
```

再起動後も **printk** 値を短くすると、システムがハングする可能性が低くなります。

この設定変更は、余分なデバッグ情報が失われる代償を伴うことに注意してください。

新たに追加された機能の詳細は、[printk の値を減らして I/O をシリアルコンソールに減らす新しい optimize-serial-console TuneD プロファイル](#) を参照してください。

(JIRA:RHELPLAN-28940)

カーネル ACPI ドライバーは、PCIe ECAM メモリーリージョンにアクセスできないことを報告します。

ファームウェアが提供する Advanced Configuration and Power Interface (ACPI) テーブルは、PCI バスデバイスの現在のリソース設定 (_CRS) メソッドにおいて PCI バス上のメモリーリージョンを定義しません。したがって、システムの起動時に以下の警告メッセージが表示されます。

```
[ 2.817152] acpi PNP0A08:00: [Firmware Bug]: ECAM area [mem 0x30000000-0x31ffffff] not reserved in ACPI namespace
[ 2.827911] acpi PNP0A08:00: ECAM at [mem 0x30000000-0x31ffffff] for [bus 00-1f]
```

ただし、カーネルは依然として **0x30000000-0x31ffffff** メモリーリージョンにアクセスできます。また、そのメモリーリージョンを PCI Enhanced Configuration Access Mechanism (ECAM) に適切に割り当てることができます。以下の出力で 256 バイトオフセットで PCIe 設定領域にアクセスして、PCI ECAM が正常に機能することを確認できます。

```
03:00.0 Non-Volatile memory controller: Sandisk Corp WD Black 2018/PC SN720 NVMe SSD (prog-if 02 [NVM Express])
...
Capabilities: [900 v1] L1 PM Substates
L1SubCap: PCI-PM_L1.2- PCI-PM_L1.1- ASPM_L1.2+ ASPM_L1.1- L1_PM_Substates+
PortCommonModeRestoreTime=255us PortTPowerOnTime=10us
L1SubCtl1: PCI-PM_L1.2- PCI-PM_L1.1- ASPM_L1.2- ASPM_L1.1-
T_CommonMode=0us LTR1.2_Threshold=0ns
L1SubCtl2: T_PwrOn=10us
```

これにより、警告メッセージを無視します。

問題の詳細は、[Firmware Bug: ECAM area mem 0x30000000-0x31ffffff not reserved in ACPI namespace" appears during system boot](#) を参照してください。

(BZ#1868526)

OPEN MPI ライブラリーは、デフォルトの PML でランタイムが失敗する可能性があります。

OPEN Message Passing Interface (OPEN MPI) 実装 4.0.x シリーズでは、UCX (Unified Communication X) がデフォルトの PPL (ポイントツーポイント) です。OPEN MPI 4.0.x シリーズの新しいバージョンでは、**openib** Byte Transfer Layer (BTL) が非推奨になりました。

ただし、OPEN MPI は **同種** クラスタ (同じハードウェアおよびソフトウェア設定) で実行される場合も、UCX は MPI **openlib** の一方向操作に BTL を使用します。これにより、実行エラーが発生する可能性があります。この問題を回避するには、以下を実行します。

- 以下のパラメーターを使用して **mpirun** コマンドを実行します。

```
-mca btl openib -mca pml ucx -x UCX_NET_DEVICES=mlx5_ib0
```

詳細は以下のようになります。

- **-mca btl openib** パラメーターは **openib** BTL を無効にします。
- **-mca pml ucx** パラメーターは、**ucx** PML を使用するように OPEN MPI を設定します。
- **x UCX_NET_DEVICES=** パラメーターは、指定したデバイスを使用するように UCX を制限します。

OPEN MPI は、**異種** クラスタ (ハードウェアおよびソフトウェア設定に異なる) を実行する場合は、デフォルトの PML として UCX を使用します。これにより、OPEN MPI ジョブが不安定なパフォーマンス、応答しない動作で実行されたり、またはクラッシュによる不具合とともに実行される可能性があります。この問題を回避するには、UCX の優先度を以下のように設定します。

- 以下のパラメーターを使用して **mpirun** コマンドを実行します。

```
-mca pml_ucx_priority 5
```

これにより、OPEN MPI ライブラリーは、UCX を介して利用可能な別のトランスポート層を選択することができます。

(BZ#1866402)

5.7.7. ファイルシステムおよびストレージ

/boot ファイルシステムを LVM に配置することができません。

/boot ファイルシステムを LVM 論理ボリュームに配置することはできません。この制限は、以下の理由により存在します。

- EFI システムでは、**EFI システムパーティション** が従来の **/boot** ファイルシステムとして機能します。UEFI 標準では、特定の GPT パーティションタイプと、このパーティションの特定のファイルシステムタイプが必要です。
- RHEL 8 は、システムブートエントリーに **Boot Loader Specification (BLS)** を使用します。この仕様では、プラットフォームのファームウェアが **/boot** ファイルシステムを読み込める必要があります。EFI システムでは、プラットフォームファームウェアは UEFI 標準で定義された **/boot** 設定のみを読み取ることができます。
- GRUB 2 ブートローダーでの LVM 論理ボリュームに対するサポートは完全ではありません。Red Hat は、UEFI や BLS などの標準があるので、この機能のユースケース数が減少しているため、サポートを改善する予定はありません。

Red Hat では、LVM での **/boot** のサポートを提供する予定はありません。代わりに、Red Hat は、**/boot** ファイルシステムを LVM 論理ボリュームに配置する必要がないシステムスナップショットおよびロールバックを管理するツールを提供します。

(BZ#1496229)

LVM で、複数のブロックサイズを持つボリュームグループが作成できない

vgcreate または **vgextend** などの LVM ユーティリティーでは、物理ボリューム (PV) の論理ブロックサイズが異なるボリュームグループ (VG) を作成できなくなりました。別のブロックサイズの PV で基礎となる論理ボリューム (LV) を拡張するとファイルシステムがマウントに失敗するため、LVM はこの変更を採用しました。

ブロックサイズが混在する VG の作成を再度有効にするには、**lvm.conf** ファイルの **allow_mixed_block_sizes=1** オプションを設定します。

(BZ#1768536)

LVM writecache の制限

writecache LVM キャッシュメソッドには以下の制限がありますが、**cache** メソッドには存在しません。

- **pvmove** コマンドを使用すると、**writecache** 論理ボリュームに名前を付けることはできません。
- **writecache** を指定した論理ボリュームは、シンプールまたは VDO と組み合わせて使用できません。

以下の制限は、**cache** メソッドにも適用されます。

- **cache** または **writecache** がアタッチされている間は、論理ボリュームのサイズを変更することはできません。

(JIRA:RHELPLAN-27987, BZ#1798631, BZ#1808012)

LUKS ボリュームを格納する LVM mirror デバイスが応答しなくなることがあります。

セグメントタイプが **mirror** のミラーリング LVM デバイスで LUKS ボリュームを格納すると、特定の条件下で応答しなくなる可能性があります。デバイスが応答しなくなると、すべての I/O 操作を拒否します。

耐障害性のソフトウェア定義ストレージに、LUKS ボリュームをスタックする必要がある場合に、この問題を回避するには、Red Hat はセグメントタイプが **mirror** ではなく **raid1** の LVM RAID 1 デバイスを使用することを推奨します。

raid1 のセグメントタイプは、デフォルトの RAID 設定タイプで、**mirror** の代わりに、推奨のソリューションとしてこのタイプが使用されます。

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

(BZ#1730502)

NFS 4.0 パッチにより、オープンな高ワークロードでパフォーマンスが低下する可能性があります。

以前、場合によっては NFS のオープン操作で、サーバー上のファイルが削除されたり、名前が変更されたりするという事実を見落とすというバグが修正されています。ただし、この修正により、多くのオープンな操作が必要とするワークロードのパフォーマンスが遅くなる可能性があります。この問題を回避するには、NFS バージョン 4.1 以降を使用します。これは、多くの場合においてクライアントに委譲を付与するように改善されています。このため、クライアントがローカルに素早く安全にオープン操作を実行できます。

(BZ#1748451)

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

32 ビットアプリケーションで呼び出されると `getpwnam()` が失敗する可能性がある

この問題の修正は、Red Hat Enterprise Linux 8.3 のリリースノートに記載されています。

NIS のユーザーが `getpwnam()` 関数を呼び出す 32 ビットアプリケーションを使用する場合は、`nss_nis.i686` パッケージがないと呼び出しに失敗します。この問題を回避するには、`yum install nss_nis.i686` コマンドを使用して、不足しているパッケージを手動でインストールします。

(BZ#1803161)

OpenLDAP ライブラリー間のシンボルの競合により、`httpd` でクラッシュが発生することがある

OpenLDAP が提供する `libldap` ライブラリーと `libldap_r` ライブラリーの両方が、単一のプロセス内にロードされ、使用されると、これらのライブラリー間でシンボルの競合が発生する可能性があります。そのため、`httpd` 設定によって `mod_security` または `mod_auth_openidc` モジュールもロードされると、PHP `ldap` 拡張機能を使用する Apache `httpd` 子プロセスが突然終了する可能性があります。

Apache Portable Runtime (APR) ライブラリーに対する今回の更新では、`APR_DEEPBIND` 環境変数を設定することでこの問題を回避できます。これにより、`httpd` モジュールのロード時に `RTLD_DEEPBIND` 動的リンカーオプションを使用できるようになります。`APR_DEEPBIND` 環境変数を有効にすると、競合するライブラリーをロードする `httpd` 設定でクラッシュが発生しなくなります。

(BZ#1819607)

MariaDB で PAM プラグインが機能しない

MariaDB 10.3 は、PAM (Pluggable Authentication Modules) プラグインバージョン 1.0 を提供します。RHEL 8 では、**MariaDB** PAM プラグインバージョン 1.0 は機能しません。この問題を回避するには、`mariadb:10.5` モジュールストリームによって提供される PAM プラグインバージョン 2.0 を使用します。これは、RHEL 8.4 で利用できます。

(BZ#1942330)

5.7.9. ID 管理

すべての KRA メンバーが非表示レプリカの場合は、KRA のインストールに失敗します。

最初の KRA インスタンスが非表示レプリカにインストールされている場合、Key Recovery Authority (KRA) がすでに存在するクラスターでは `ipa-kra-install` ユーティリティーで問題が発生します。そのため、これ以上、追加の KRA インスタンスをクラスターに追加することはできません。

この問題を回避するには、新しい KRA インスタンスを追加する前に、KRA ロールが割り当てられた非表示レプリカを解除します。`ipa-kra-install` が正常に終了してから、レプリカを再度非表示にできません。

(BZ#1816784)

`--agent-uid pkidbuser` オプションを指定して `cert-fix` ユーティリティーを使用すると、証明書システムが破損します。

`--agent-uid pkidbuser` オプションを指定して `cert-fix` ユーティリティーを使用すると、証明書システムの LDAP 設定が破損します。したがって、証明書システムは不安定になり、システムの復元に手動の操作が必要になる可能性があります。

(BZ#1729215)

PKI CA に接続する PKI ACME Responder が発行する証明書により、OCSP の検証が失敗する可能性があります。

PKI CA が提供するデフォルトの ACME 証明書プロファイルには、実際の OCSP サービスを参照しないサンプル OCSP URL が含まれています。これにより、PKI ACME Responder が PKI CA 発行者を使用す

るように設定されている場合、レスポンスが発行する証明書は OCSP 検証に失敗する可能性があります。

この問題を回避するには、`/usr/share/pki/ca/profiles/ca/acmeServerCert.cfg` 設定ファイルの `policyset.serverCertSet.5.default.params.authInfoAccessADLocation_0` プロパティを空の値に設定する必要があります。

1. ACME Responder 設定ファイルで `policyset.serverCertSet.5.default.params.authInfoAccessADLocation_0=http://ocsp.example.com` を `policyset.serverCertSet.5.default.params.authInfoAccessADLocation_0=` に変更します。
2. サービスを再起動して、証明書を再生成します。

これにより、PKI CA は、実際の OCSP サービスを参照する自動生成 OCSP URL で ACME 証明書を生成します。

(BZ#1868233)

FreeRADIUS が 249 文字を超える Tunnel-Passwords を断りなく切り捨てます。

Tunnel-Password が 249 文字を超える場合、FreeRADIUS サービスはそのパスワードを断りなく切り捨てます。これにより、他のシステムと矛盾する想定外のパスワードになる可能性があります。

この問題を回避するには、249 文字以下のパスワードを選択します。

(BZ#1723362)

IdM ホストの `/var/log/lastlog` 分析ファイルが、パフォーマンスの問題を引き起こす可能性があります。

IdM のインストール時に、利用できる合計 10,000 の範囲からの 200,000 の UID の範囲が無作為に選択され、割り当てられます。このようにランダムな範囲を選択すると、今後別の 2 つの IdM ドメインを統合する場合に、ID の競合が発生する可能性を大幅に削減できます。

ただし、UID が多いと、`/var/log/lastlog` ファイルで問題が発生する可能性があります。たとえば、1280000008 の UID を持つユーザーが IdM クライアントにログインすると、ローカルの `/var/log/lastlog` ファイルサイズは、約 400 GB に増えます。実際のファイルはスパースで、その領域をすべて使用しません。ただし、一部のアプリケーションはデフォルトではスパースファイルを識別するように設計されています。そのため、それら进行处理する特定のオプションが必要になる場合があります。たとえば、設定が複雑でバックアップ、コピーアプリケーションがスパースファイルを正しく処理しない場合、ファイルはサイズが 400 GB であるかのようにコピーされます。この動作により、パフォーマンスの問題が発生する可能性があります。

この問題を回避するには、以下を実行します。

- 標準パッケージの場合は、そのドキュメントを参照して、スパースファイル进行处理するオプションを特定します。
- カスタムアプリケーションの場合、`/var/log/lastlog` などのスパースファイルを正しく管理できることを確認してください。

(JIRA:RHELPLAN-59111)

`ldap_id_use_start_tls` オプションのデフォルト値を使用する場合の潜在的なリスク

ID ルックアップに TLS を使用せずに `ldap://` を使用すると、攻撃ベクトルのリスクが生じる可能性があります。特に、中間者 (MITM) 攻撃は、攻撃者が、たとえば、LDAP 検索で返されたオブジェクトの UID または GID を変更することによってユーザーになりすますことを可能にする可能性があります。

現在、TLS を強制する SSSD 設定オプション `ldap_id_use_start_tls` は、デフォルトで `false` に設定されています。セットアップが信頼できる環境で動作していることを確認し、`id_provider = ldap` に暗号化されていない通信を使用しても安全かどうかを判断してください。注記: `id_provider = ad` および `id_provider = ipa` は、SASL および GSSAPI によって保護された暗号化接続を使用するため、影響を受けません。

暗号化されていない通信を使用することが安全ではない場合は、`/etc/sss/sss.conf` ファイルで `ldap_id_use_start_tls` オプションを `true` に設定して TLS を強制します。デフォルトの動作は、RHEL の将来のリリースで変更される予定です。

(JIRA:RHELPLAN-155168)

5.7.10. デスクトップ

ソフトウェアリポジトリからの flatpak リポジトリの無効化ができません。

現時点で、GNOME Software ユーティリティーの Software Repositories ツールで `flatpak` リポジトリを無効化または削除することはできません。

(BZ#1668760)

ドラッグアンドドロップが、デスクトップとアプリケーション間で機能しません。

`gnome-shell-extensions` パッケージのバグにより、ドラッグアンドドロップ機能は現在、デスクトップとアプリケーションの間では機能しません。この機能のサポートは、今後のリリースで追加される予定です。

(BZ#1717947)

Generation 2 の RHEL 8 仮想マシンが Hyper-V Server 2016 ホストで起動できない場合があります。

Microsoft Hyper-V Server 2016 ホストで実行している仮想マシンで RHEL 8 をゲストオペレーティングシステムとして使用すると、仮想マシンが起動しなくなり、GRUB ブートメニューに戻る場合があります。さらに、以下のエラーが Hyper-V イベントログに記録されます。

The guest operating system reported that it failed with the following error code: 0x1E

このエラーは、Hyper-V ホストの UEFI ファームウェアバグが原因で発生します。この問題を回避するには、Hyper-V Server 2019 をホストとして使用します。

(BZ#1583445)

5.7.11. グラフィックインフラストラクチャー

radeon がハードウェアを適切なハードウェアリセットに失敗します。

現在、`radeon` カーネルドライバーは、`kexec` コンテキストでハードウェアを正しくリセットしません。代わりに `radeon` がフェイルオーバーします。これにより、`kdump` サービスの残りの部分が失敗します。

この問題を回避するには、`/etc/kdump.conf` ファイルに以下の行を追加して、`kdump` で `radeon` を無効にします。

```
dracut_args --omit-drivers "radeon"  
force_rebuild 1
```

マシンと **kdump** を再起動します。**kdump**の起動後、設定ファイルから **force_rebuild 1** 行が削除される可能性があります。

このシナリオでは、**kdump** 中にグラフィックは利用できませんが、**kdump** は正常に動作します。

(BZ#1694705)

1つの MST トポロジーで複数の HDR ディスプレイを使用すると、電源が入らないことがあります。

nouveau ドライバーの NVIDIA Turing GPUs を使用するシステムで、**DisplayPort** ハブ (ラップトップのドックなど) を使用して HDR プラグインのサポートがあるモニターを複数接続すると、電源が入らないことがあります。これは、全ディスプレイをサポートする帯域幅がハブ上にないと、システムが誤って判断してしまうことが原因で発生します。

(BZ#1812577)

sudo コマンドを使用してグラフィカルアプリケーションを実行できません。

権限が昇格されたユーザーで、グラフィカルアプリケーションを実行しようとする時、エラーメッセージが表示され、アプリケーションを開くことができません。この障害は、**Xauthority** ファイルで、通常ユーザーの認証情報を使用して認証するように、**Xwayland** に制限が加えられているため発生します。

この問題を回避するには、**sudo -E** コマンドを使用して、**root** ユーザーとしてグラフィカルアプリケーションを実行します。

(BZ#1673073)

VNC Viewer が、IBM Z で 16 ビットのカラーデプスで誤った色を表示

VNC Viewer アプリケーションは、16 ビットのカラーデプスで IBM Z サーバーの VNC セッションに接続すると、誤った色を表示します。

この問題を回避するには、VNC サーバーで 24 ビットのカラーデプスを設定します。**Xvnc** サーバーの場合は、**Xvnc** 設定で **-depth 16** オプションを **-depth 24** に置き換えます。

その結果、VNC クライアントで色が正しく表示されますが、サーバーでは、より多くのネットワーク帯域幅が使用されます。

(BZ#1886147)

ARM でハードウェアアクセラレーションがサポートされない

組み込みグラフィックドライバーは、64 ビット ARM アーキテクチャー上のハードウェアアクセラレーションまたは Vulkan API に対応していません。

ARM でハードウェアアクセラレーションまたは Vulkan を有効にするには、プロプライエタリーの Nvidia ドライバーをインストールします。

(JIRA:RHELPLAN-57914)

NVIDIA Ampere で RHEL インストーラーが応答しなくなる

RHEL 8.3.0 は NVIDIA Ampere GPU に対応していません。NVIDIA Ampere GPU を持つシステムで RHEL インストールを開始すると、インストーラーが応答しなくなります。したがって、インストールは正常に完了できません。

NVIDIA Ampere ファミリーには、以下の GPU モデルが含まれます。

- GeForce RTX 3060 Ti
- GeForce RTX 3070
- GeForce RTX 3080
- GeForce RTX 3090
- RTX A6000
- NVIDIA A40
- NVIDIA A100
- NVIDIA A100 80GB

この問題を回避するには、**nouveau** グラフィックスドライバーを無効にして、テキストモードで RHEL をインストールします。

1. インストーラーの起動メニューで起動します。
2. カーネルコマンドラインに **nouveau.modeset=0** オプションを追加します。
詳細は [起動オプションの編集](#) を参照してください。
3. システムに RHEL をインストールします。
4. 新たにインストールした RHEL で起動起動メニューで、カーネルコマンドラインに **nouveau.modeset=0** オプションを追加します。
5. **nouveau** ドライバーを永続的に無効にします。

```
# echo 'blacklist nouveau' >> /etc/modprobe.d/blacklist.conf
```

これにより、インストールが正常に完了し、RHEL がテキストモードで実行されるようになりました。

オプションとして、プロプライエタリー NVIDIA GPU ドライバーをインストールして、グラフィックスを有効にできます。手順は、[RHEL 8 に NVIDIA プロプライエタリードライバーをインストールする方法](#) を参照してください。

(BZ#1903890)

5.7.12. Web コンソール

非特権ユーザーがサブスクリプションページにアクセスできます。

管理者以外のユーザーが Web コンソールの [サブスクリプションページ](#) に移動すると、Web コンソールでは、**Cockpit had an unexpected internal error** (Cockpit に予期しない内部エラーが発生しました) という一般的なエラーメッセージが表示されます。

この問題を回避するには、権限のあるユーザーで Web コンソールにサインインし、**Reuse my password for privileged tasks** チェックボックスをチェックします。

(BZ#1674337)

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

システムロールロギングでは、**oVirt** 入力および **elasticsearch** 出力機能に対応していません。

システムロールロギングでは、**oVirt** 入力と **elasticsearch** の出力はサポートされていませんが、README ファイルで説明されています。現在利用できる回避策はありません。

(BZ#1889468)

5.7.14. 仮想化

QXL で、**Wayland** を使用する仮想マシンの複数のモニターを表示できません。

remote-viewer ユーティリティを使用して、**Wayland** ディスプレイサーバーを使用している仮想マシンのモニターを複数表示すると、仮想マシンが応答なくなり、**Waiting for display** というステータスメッセージが永久に表示されます。

この問題を回避するには、**Wayland** を使用する仮想マシンの GPU デバイスとして **qxl** の代わりに **virtio-gpu** を使用します。

(BZ#1642887)

virsh iface-* コマンドが一貫して動作しません。

現在、**virsh iface-*** コマンド (**virsh iface-start**、**virsh iface-destroy** など) は、設定の依存関係が原因で頻繁に失敗します。したがって、ホストネットワーク接続の設定および管理には **virsh iface-*** コマンドを使用しないことが推奨されます。代わりに、**NetworkManager** プログラムとその関連管理アプリケーションを使用します。

(BZ#1664592)

多数の **virtio-blk** ディスクを使用すると、仮想マシンが起動しないことがあります。

多数の **virtio-blk** デバイスを仮想マシンに追加すると、プラットフォームで利用可能な割り込みベクトルの数が使い切られる可能性があります。これが発生すると、仮想マシンのゲスト OS は起動できず、**dracut-initqueue[392]: Warning: Could not boot** エラーが表示されます。

(BZ#1719687)

virtio-blk を使用して仮想マシンに **LUN** デバイスを割り当てると機能しません。

q35 マシンタイプは、移行用の **virtio 1.0** デバイスをサポートしないため、**RHEL 8** では **virtio 1.0** で非推奨となった機能はサポートされません。特に、**RHEL 8** ホストで **virtio-blk** デバイスから **SCSI** コマンドを送信することはできません。したがって、**virtio-blk** コントローラーを使用する場合は、物理ディスクを **LUN** デバイスとして仮想マシンに割り当てると失敗します。

物理ディスクをゲストオペレーティングシステムを通して渡すことは引き続き可能ですが、**device='lun'** オプションではなく、**device='disk'** オプションで設定する必要があることに留意してください。

(BZ#1777138)

ホストで **TSX** が無効になっていると、**Cooperlake** を使用する仮想マシンを起動できません。

現在、ホストで **TSX** CPU フラグが無効化されている場合、**Cooperlake** CPU モデルを使用する仮想マシンは起動に失敗します。代わりに、ホストに以下のエラーメッセージが表示されます。

```
the CPU is incompatible with host CPU: Host CPU does not provide required features: hle, rtm
```

このようなホストで仮想マシンが **Cooperlake** を使用できるようにするには、VM の XML 設定の VM 設定で HLE、RTM、および TAA_NO のフラグを無効化します。

```
<feature policy='disable' name='hle'/>
<feature policy='disable' name='rtm'/>
<feature policy='disable' name='taa-no'/>
```

(BZ#1860743)

仮想マシンが、Witherspoon ホストで起動できないことがあります。

仮想マシン (VM) で **pseries-rhel7.6.0-sxxm** マシンタイプを使用すると、DD 2.2 または DD2.3 CPU を使用する **Power9 S922LC for HPC** ホストで起動できない場合があります。

代わりに仮想マシンを起動しようとするすると、以下のエラーメッセージが出力されます。

```
qemu-kvm: Requested safe indirect branch capability level not supported by kvm
```

この問題を回避するには、仮想マシンの XML 設定を次のように設定します。

```
<domain type='qemu' xmlns:qemu='http://libvirt.org/schemas/domain/qemu/1.0'>
  <qemu:commandline>
    <qemu:arg value='-machine'/>
    <qemu:arg value='cap-ibs=workaround'/>
  </qemu:commandline>
```

(BZ#1732726)

5.7.15. クラウド環境の RHEL

Azure NV6 インスタンスの GPU の問題

Microsoft Azure NV6 インスタンスで RHEL 8 をゲストオペレーティングシステムとして実行すると、仮想マシンの休止から仮想マシンを再開したときに、仮想マシンの GPU が不適切に動作することがあります。これが発生すると、カーネルは以下のメッセージをログに記録します。

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

(BZ#1846838)

Azure および Hyper-V で kdump が起動しないことがあります。

Microsoft Azure または Hyper-V ハイパーバイザーでホストされている RHEL 8 ゲストオペレーティングシステムでは、実行後通知が有効な場合に **kdump** カーネルの起動が失敗することがあります。

この問題を回避するには、`crash kexec post notifiers` を無効にします。

```
# echo N > /sys/module/kernel/parameters/crash_kexec_post_notifiers
```

(BZ#1865745)

VMWare ホストの RHEL 8 仮想マシンで静的 IP を設定できませんでした。

現在、VMWare ホストで RHEL 8 を仮想マシンのゲストオペレーティングシステムとして使用すると、DatasourceOVF 機能は正しく機能しません。これにより、**cloud-init** ユーティリティーを使用して、仮想マシンのネットワークを静的 IP に設定し、仮想マシンを再起動すると、仮想マシンのネットワークが DHCP に変更されます。

(BZ#1750862)

特定の NIC を搭載した RHEL 8 仮想マシンの Azure のリモートマシンへのコアダンプには、予想よりも長い時間がかかっていました。

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

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

(BZ#1854037)

仮想マシンが休止状態から再開した後 TX/RX パケットカウンターが増加しない

TX/RX パケットカウンターは、CX4 VF NIC を使用して RHEL 8 仮想マシン (VM) が Microsoft Azure のハイバネートから再開したときに、カウントを停止します。カウンターの動作を維持するには、仮想マシンを再起動します。これを実行すると、カウンターがリセットされることに注意してください。

(BZ#1876527)

RHEL 8 仮想マシンが Azure のハイバネートから再開できない

SR-IOV が有効になっている RHEL 8 仮想マシンが Microsoft Azure でハイバネートし、割り当て解除すると、Virtual Function (VF)、**vmbus** デバイスの GUID が変更になります。その結果、仮想マシンが再起動すると、再開とクラッシュに失敗します。回避策として、Azure シリアルコンソールを使用して仮想マシンのハードリセットを実行します。

(BZ#1876519)

RHEL 7-ALT ホストから RHEL 8 への POWER9 ゲストの移行に失敗する

現在のリリースでは、RHEL 7-ALT ホストシステムから RHEL 8 に POWER9 仮想マシンを移行すると、Migration status: active のステータスで応答がなくなります。

この問題を回避するには、RHEL 7-ALT ホストで Transparent Huge Pages (THP) を無効にすることで、移行が正常に完了します。

(BZ#1741436)

5.7.16. サポート関連

redhat-support-tool が FUTURE 暗号化ポリシーを使用すると機能しません。

カスタマーポータル API の証明書が使用する暗号化キーは **FUTURE** のシステム全体の暗号化ポリシーが定義する要件を満たさないため、現時点で **redhat-support-tool** ユーティリティーは、このポリシーレベルでは機能しません。

この問題を回避するには、カスタマーポータル API への接続中に **DEFAULT** 暗号化ポリシーを使用します。

(BZ#1802026)

5.7.17. コンテナ

UDICA が 1.0 の安定したストリームと連携するように想定されていません。

UDICA (コンテナの SELinux ポリシーを生成するツール) は、**container-tools:1.0** モジュールストリームの podman 1.0.x で実行されるコンテナで機能するように想定されていません。

(JIRA:RHELPLAN-25571)

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

podman system connection add コマンドは、最初の接続をデフォルト接続に自動的に設定しません。デフォルトの接続を設定するには、**podman system connection default <connection_name>** コマンドを手動で実行する必要があります。

(BZ#1881894)

第6章 国際化

6.1. RED HAT ENTERPRISE LINUX 8 の多言語

Red Hat Enterprise Linux 8 は、複数の言語のインストールと、要件に応じた言語の変更に対応します。

- 東アジア言語 - 日本語、韓国語、簡体字中国語、および繁体字中国語。
- ヨーロッパ言語 - 英語、ドイツ語、スペイン語、フランス語、イタリア語、ポルトガル語、およびロシア語。

次の表は、さまざまな主要言語に提供されるフォントと入力方法を示しています。

言語	デフォルトフォント (フォントパッケージ)	入力メソッド
英語	dejavu-sans-fonts	
フランス語	dejavu-sans-fonts	
ドイツ語	dejavu-sans-fonts	
イタリア語	dejavu-sans-fonts	
ロシア語	dejavu-sans-fonts	
スペイン語	dejavu-sans-fonts	
ポルトガル語	dejavu-sans-fonts	
簡体字中国語	google-noto-sans-cjk-ttc-fonts、 google-noto-serif-cjk-ttc-fonts	ibus-libpinyin、libpinyin
繁体字中国語	google-noto-sans-cjk-ttc-fonts、 google-noto-serif-cjk-ttc-fonts	ibus-libzhuyin、libzhuyin
日本語	google-noto-sans-cjk-ttc-fonts、 google-noto-serif-cjk-ttc-fonts	ibus-kkc、libkkc
韓国語	google-noto-sans-cjk-ttc-fonts、 google-noto-serif-cjk-ttc-fonts	ibus-hangul、libhangul

6.2. RHEL 8 における国際化の主な変更点

RHEL 8 では、RHEL 7 の国際化に以下の変更が加えられています。

- Unicode 11 コンピューティングの業界標準のサポートが追加されました。

- 国際化は複数のパッケージで配布され、より小さなフットプリントのインストールを可能にします。詳細は、[Using langpacks](#) を参照してください。
- 多くの **glibc** ロケールが Unicode Common Locale Data Repository (CLDR) と同期されています。

付録A コンポーネント別のチケットリスト

本書には Bugzilla と JIRA ID が記載されています。一般にアクセス可能な Bugzilla バグには、チケットへのリンクが含まれます。

コンポーネント	チケット
389-ds-base	BZ#1816862 、 BZ#1638875 、 BZ#1728943
NetworkManager	BZ#1814746 、 BZ#1626348
anaconda	BZ#1665428 、 BZ#1775975 、 BZ#1630299 、 BZ#1823578 、 BZ#1672405 、 BZ#1644662 、 BZ#1745064 、 BZ#1821192 、 BZ#1822880 、 BZ#1862116 、 BZ#1890261 、 BZ#1891827 、 BZ#1691319 、 BZ#1931069
apr	BZ#1819607
authselect	BZ#1654018
bcc	BZ#1837906
bind	BZ#1818785
buildah-container	BZ#1627898
buildah	BZ#1806044
clevis	BZ#1716040 、 BZ#1818780 、 BZ#1436735 、 BZ#1819767
cloud-init	BZ#1750862
cloud-utils-growpart	BZ#1846246
cockpit-session-recording	BZ#1826516
cockpit	BZ#1710731 、 BZ#1666722
corosync-qdevice	BZ#1784200
crun	BZ#1841438
crypto-policies	BZ#1832743 、 BZ#1660839
cyrus-sasl	BZ#1817054
distribution	BZ#1815402 、 BZ#1657927

コンポーネント	チケット
dnf	BZ#1793298 , BZ#1832869 , BZ#1842285
elfutils	BZ#1804321
fapolicyd	BZ#1897090 , BZ#1817413 , BZ#1714529
fence-agents	BZ#1830776 、 BZ#1775847
firewalld	BZ#1790948 , BZ#1682913 , BZ#1809225 , BZ#1817205 , BZ#1809636
freeradius	BZ#1672285 , BZ#1859527 , BZ#1723362
gcc-toolset-10-gdb	BZ#1838777
gcc	BZ#1784758
gdb	BZ#1659535
git	BZ#1825114
glibc	BZ#1812756 、 BZ#1743445 、 BZ#1783303 、 BZ#1642150 、 BZ#1810146 、 BZ#1748197 、 BZ#1774115 、 BZ#1807824 、 BZ#1757354 、 BZ#1836867 、 BZ#1780204 、 BZ#1821531 、 BZ#1784525
gnome-session	BZ#1739556
gnome-shell-extensions	BZ#1717947
gnome-shell	BZ#1724302
gnome-software	BZ#1668760
gnutls	BZ#1677754 、 BZ#1789392 、 BZ#1849079 、 BZ#1855803
go-toolset	BZ#1820596
gpgme	BZ#1829822
grafana-container	BZ#1823834
grafana-pcp	BZ#1807099
grafana	BZ#1807323

コンポーネント	チケット
grub2	BZ#1583445
httpd	BZ#1209162
initial-setup	BZ#1676439
ipa-healthcheck	BZ#1852244
ipa	BZ#1816784 、 BZ#1810154 、 BZ#913799 、 BZ#1651577 、 BZ#1851139 、 BZ#1664719 、 BZ#1664718
iperf3	BZ#1665142、 BZ#1700497
jss	BZ#1821851
kernel-rt	BZ#1818138
kernel	BZ#1758323、BZ#1812666、BZ#1793389、BZ#1694705、BZ#1748451、BZ#1654962、BZ#1792125、BZ#1708456、BZ#1812577、BZ#1757933、BZ#1847837、BZ#1791664、BZ#1666538、BZ#1602962、BZ#1609288、BZ#1730502、BZ#1806882、BZ#1660290、BZ#1846838、BZ#1865745、BZ#1868526、BZ#1884857、BZ#1854037、BZ#1876527、BZ#1876519、 BZ#1823764 、BZ#1822085、BZ#1735611、BZ#1281843、BZ#1828642、BZ#1825414、BZ#1761928、BZ#1791041、BZ#1796565、BZ#1834769、BZ#1785660、BZ#1683394、BZ#1817752、BZ#1782831、BZ#1821646、BZ#1519039、BZ#1627455、BZ#1501618、BZ#1495358、BZ#1633143、BZ#1503672、BZ#1570255、BZ#1696451、BZ#1348508、 BZ#1778762 、BZ#1839311、BZ#1783396、BZ#1665295、BZ#1658840、BZ#1660627、BZ#1569610
krb5	BZ#1791062、 BZ#1784655 、 BZ#1820311 、BZ#1802334、 BZ#1877991
libbpf	BZ#1759154
libcap	BZ#1487388
libdb	BZ#1670768
libffi	BZ#1723951
libgnome-keyring	BZ#1607766
libkcapi	BZ#1683123
libmaxminddb	BZ#1642001

コンポーネント	チケット
libpcap	BZ#1806422
libreswan	BZ#1544463 , BZ#1820206
libseccomp	BZ#1770693
libselinux-python-2.8-module	BZ#1666328
libssh	BZ#1804797
libvirt	BZ#1664592 , BZ#1528684
lldb	BZ#1841073
llvm-toolset	BZ#1820587
llvm	BZ#1820319
lshw	BZ#1794049
lvm2	BZ#1496229 , BZ#1768536 , BZ#1598199 , BZ#1541165 , JIRA:RHELPLAN-39320
mariadb	BZ#1942330
Memcached CRD	BZ#1809536
mesa	BZ#1886147
microdnf	BZ#1781126
mod_http2	BZ#1814236
nfs-utils	BZ#1817756 , BZ#1592011
nginx	BZ#1668717 , BZ#1826632
nmstate	BZ#1674456
nss_nis	BZ#1803161
nss	BZ#1817533 , BZ#1645153
opencryptoki	BZ#1780293

コンポーネント	チケット
openmpi	BZ#1866402
opencs	BZ#1810660
openscap	BZ#1803116 , BZ#1870087 , BZ#1795563 , BZ#1824152 , BZ#1829761
openssh	BZ#1744108
openssl	BZ#1685470, BZ#1810911
oscap-anaconda-addon	BZ#1816199, BZ#1665082, BZ#1674001 , BZ#1691305, BZ#1787156 , BZ#1843932 , BZ#1834716
pacemaker	BZ#1828488 , BZ#1784601 , BZ#1837747, BZ#1718324
papi	BZ#1807346, BZ#1664056 , BZ#1726070
pcp-container	BZ#1497296
pcp	BZ#1792971
pcs	BZ#1817547, BZ#1684676 , BZ#1839637 , BZ#1619620
perl-5.30-module	BZ#1713592
perl-IO-Socket-SSL	BZ#1824222
perl-libwww-perl	BZ#1781177
php	BZ#1797661
pki-core	BZ#1729215 , BZ#1868233 , BZ#1770322, BZ#1824948
podman	BZ#1804193 , BZ#1881894 , BZ#1627899
powertop	BZ#1783110
pykickstart	BZ#1637872
python38	BZ#1847416
qemu-kvm	BZ#1719687 , BZ#1860743 , JIRA:RHELPLAN-45901, BZ#1651994
rear	BZ#1843809 , BZ#1729502 , BZ#1743303

コンポーネント	チケット
redhat-support-tool	BZ#1802026
resource-agents	BZ#1814896
rhel-system-roles-sap	BZ#1844190 、 BZ#1660832
rhel-system-roles	BZ#1889468 、 BZ#1822158 、 BZ#1677739
rpm	BZ#1688849
rsyslog	BZ#1659383 、 JIRA:RHELPLAN-10431 、 BZ#1679512 、 BZ#1713427
ruby-2.7-module	BZ#1817135
ruby	BZ#1846113
rust-toolset	BZ#1820593
samba	BZ#1817557 、 JIRA:RHELPLAN-13195
scap-security-guide	BZ#1843913 、 BZ#1858866 、 BZ#1750755 、 BZ#1760734 、 BZ#1832760 、 BZ#1815007
scap-workbench	BZ#1640715
selinux-policy	BZ#1826788 、 BZ#1746398 、 BZ#1776873 、 BZ#1772852 、 BZ#1641631 、 BZ#1860443
setools	BZ#1820079
skopeo-container	BZ#1627900
smartmontools	BZ#1671154
spice	BZ#1849563
squid	BZ#1829467
sssd	BZ#1827615 、 BZ#1793727
stratis-cli	BZ#1734496
stunnel	BZ#1808365
subscription-manager	BZ#1674337

コンポーネント	チケット
sudo	BZ#1786990
systemtap	BZ#1804319
tang	BZ#1716039
tcpdump	BZ#1804063
tigervnc	BZ#1806992
tpm2-tools	BZ#1789682
tuned	BZ#1792264 、 BZ#1840689 、 BZ#1746957
udica	BZ#1763210
usbguard	BZ#1738590 、 BZ#1667395 、 BZ#1683567
valgrind	BZ#1804324
wayland	BZ#1673073
xdp-tools	BZ#1880268 、 BZ#1820670
xorg-x11-drv-qxl	BZ#1642887
xorg-x11-server	BZ#1698565
yum	BZ#1788154

コンポーネント	チケット
その他	JIRA:RHELPLAN-45950、 JIRA:RHELPLAN-57572、 BZ#1640697、 BZ#1659609、 BZ#1687900 、 BZ#1697896、 BZ#1790635、 BZ#1823398、 BZ#1757877、 JIRA:RHELPLAN-25571、 BZ#1777138、 JIRA:RHELPLAN-27987、 JIRA:RHELPLAN-28940、 JIRA:RHELPLAN-34199、 JIRA:RHELPLAN-57914、 BZ#1897383 、 BZ#1900019 、 BZ#1839151 、 BZ#1780124 、 JIRA:RHELPLAN-42395、 BZ#1889736 、 BZ#1842656、 JIRA:RHELPLAN-45959、 JIRA:RHELPLAN-45958、 JIRA:RHELPLAN-45957、 JIRA:RHELPLAN-45956、 JIRA:RHELPLAN-45952、 JIRA:RHELPLAN-45945、 JIRA:RHELPLAN-45939、 JIRA:RHELPLAN-45937、 JIRA:RHELPLAN-45936、 JIRA:RHELPLAN-45930、 JIRA:RHELPLAN-45926、 JIRA:RHELPLAN-45922、 JIRA:RHELPLAN-45920、 JIRA:RHELPLAN-45918、 JIRA:RHELPLAN-45916、 JIRA:RHELPLAN-45915、 JIRA:RHELPLAN-45911、 JIRA:RHELPLAN-45910、 JIRA:RHELPLAN-45909、 JIRA:RHELPLAN-45908、 JIRA:RHELPLAN-45906、 JIRA:RHELPLAN-45904、 JIRA:RHELPLAN-45900、 JIRA:RHELPLAN-45899、 JIRA:RHELPLAN-45884、 JIRA:RHELPLAN-37573、 JIRA:RHELPLAN-37570、 JIRA:RHELPLAN-49954、 JIRA:RHELPLAN-50002、 JIRA:RHELPLAN-43531、 JIRA:RHELPLAN-48838、 BZ#1873567 、 BZ#1866695 、 JIRA:RHELPLAN-14068、 JIRA:RHELPLAN-7788、 JIRA:RHELPLAN-40469、 JIRA:RHELPLAN-42617、 JIRA:RHELPLAN-30878、 JIRA:RHELPLAN-37517、 JIRA:RHELPLAN-55009、 JIRA:RHELPLAN-42396、 BZ#1836211、 JIRA:RHELPLAN-57564、 JIRA:RHELPLAN-57567、 BZ#1890499 、 JIRA:RHELPLAN-40234、 JIRA:RHELPLAN-56676、 JIRA:RHELPLAN-14754、 JIRA:RHELPLAN-51289、 BZ#1893174 、 BZ#1690207、 JIRA:RHELPLAN-1212、 BZ#1559616、 BZ#1889737 、 BZ#1812552 、 JIRA:RHELPLAN-14047、 BZ#1769727 、 JIRA:RHELPLAN-27394、 JIRA:RHELPLAN-27737、 JIRA:RHELPLAN-41549、 BZ#1642765、 JIRA:RHELPLAN-10304、 BZ#1646541、 BZ#1647725、 BZ#1686057 、 BZ#1748980 、 BZ#1827628、 BZ#1871025 、 BZ#1871953 、 BZ#1874892、 BZ#1893767 、 JIRA:RHELPLAN-60226

付録B 改訂履歴

0.4-1

2024年5月15日(水) Brian Angelica (bangelic@redhat.com)

- [BZ#1690207](#) でテクニカルプレビューを更新しました。

0.4-0

2024年5月9日(木)、Gabriela Fialova (gfialova@redhat.com)

- 既知の問題 [BZ#1730502](#) (ストレージ) を更新しました。

0.3-9

2023年12月7日木曜日、Lucie Vařáková (lvarakova@redhat.com)

- 非推奨の機能 [JIRA:RHELDOCS-16612](#) (Identity Management) を追加しました。

0.3-8

2023年11月10日金曜日、Gabriela Fialová (gfialova@redhat.com)

- RHEL ドキュメントへのフィードバックの提供に関するモジュールを更新しました。

0.3-7

2022年11月15日(火) Gabriela Fialová (gfialova@redhat.com)

- 壊れたリンクを修正しました。

0.3-6

2023年10月13日(金) Gabriela Fialová (gfialova@redhat.com)

- テクノロジープレビュー [JIRA:RHELDOCS-16861](#) (コンテナ) を追加。

0.3-5

2023年4月27日(木) Gabriela Fialová (gfialova@redhat.com)

- 既知の問題 [JIRA:RHELPLAN-155168](#) (アイデンティティ管理) を追加

0.3-4

2022年4月29日、金曜日、Lenka Špačková (lspackova@redhat.com)

- 非推奨になった機能の概要を更新。
- [BZ#1605216](#) のタイプミスを修正。
- 無効になっていたリンクを修正。

0.3-3

2022年4月5日(火) Jaroslav Klech (jklech@redhat.com)

- バグ修正 [BZ#1666538](#) (カーネル) を追加しました。

0.3-2

2022年3月17日(木) Jaroslav Klech (jklech@redhat.com)

- 既知の問題を追加 (Kernel)

0.3-1

2022年2月4日(金) Lucie Maňásková (Imanasko@redhat.com)

- 非推奨の機能 [BZ#1794513](#) (ファイルシステムおよびストレージ) が追加されました。

0.3-0

2021年12月23日(木) Lenka Špačková (lspackova@redhat.com)

- Soft-RoCE ドライバー `rdma_rxe` に関する情報が、テクノロジープレビューの [BZ#1605216](#) および非推奨の機能 [BZ#1878207](#) (カーネル) に追加されました。

0.2-9

2021年10月7日(木) Lenka Špačková (lspackova@redhat.com)

- 既知の問題 [BZ#1942330](#) を更新 (動的プログラミング言語、Web サーバー、およびデータベースサーバー)。

0.2-8

2021年10月5日(火) Lucie Maňásková (Imanasko@redhat.com)

- 非推奨の機能 [BZ#1999620](#) (シェルおよびコマンドラインツール) が追加されました。

0.2-7

2021年8月19日(木) Lucie Maňásková (Imanasko@redhat.com)

- [YUM/DNF を使用したパッケージ管理](#) を、ディストリビューションの章に追加。
- [BZ#1708456](#) (Kernel) のテキストを更新。

0.2-6

2021年7月9日(金) Lucie Maňásková (Imanasko@redhat.com)

- [JIRA:RHELPLAN-34199](#) (セキュリティー) のテキストを更新。

0.2-5

2021年6月23日(水) Lucie Maňásková (Imanasko@redhat.com)

- [BZ#1922488](#) の AlternateTab の削除に関する情報を追加 (デスクトップ)。

0.2-4

2021年5月21日(金) Lenka Špačková (lspackova@redhat.com)

- [概要](#) で OS 変換に関する情報を更新。

0.2-3

2021年5月20日(木) Lenka Špačková (lspackova@redhat.com)

- 既知の問題 [BZ#1942330](#) に回避策を追加 (動的プログラミング言語、Web サーバー、およびデータベースサーバー)。

0.2-2

2021年5月14日(金) Lucie Maňásková (Imanasko@redhat.com)

- .NET 5 サポートに関する新機能 [BZ#1944677](#) が追加 (コンパイラおよび開発ツール)。
- 新機能 `xfer:BZ-1959289[BZ#1959289]` (RHEL システムロール) を追加。
- [BZ#1820670](#) (ネットワーク) および [BZ#1780124](#) (カーネル) の **xdp-tools** パッケージに関する情報を更新。

0.2-1

2021年4月19日(月) Lenka Špačková (lspackova@redhat.com)

- 既知の問題 [BZ#1942330](#) を追加 (動的プログラミング言語、Web サーバー、およびデータベースサーバー)。

0.2-0

2021年4月13日(火) Lenka Špačková (lspackova@redhat.com)

- 既知の問題の追加 (インストーラーおよびイメージの作成)

0.1-9

2021年4月6日(火) Lenka Špačková (lspackova@redhat.com)

- 対応しているアーキテクチャーのリストを改善

0.1-8

2021年3月31日(水) Lenka Špačková (lspackova@redhat.com)

- サポートされている **Convert2RHEL** ユーティリティの可用性で OS 変換に関する情報を更新。

0.1-7

2021年3月29日(月) Lucie Maňásková (Imanasko@redhat.com)

- 新機能セクションを更新 (Kernel)。

0.1-6

2021年2月25日(木) Lenka Špačková (lspackova@redhat.com)

- CentOS Linux 名を修正。

0.1-5

2021年2月23日(火) Lucie Maňásková (Imanasko@redhat.com)

- 既知の問題を追加 (Identity Management)。
- **podman** ユーティリティのリベースを RHEL 8.3.1 セクションに追加。

0.1-4

2021年2月18日(木) Jaroslav Klech (jklech@redhat.com)

- 既知の問題を追加 (Kernel)

- 機能強化のリンクを修正 (Kernel)。

0-1-3

2021年2月16日(火) Lenka Špačková (lspackova@redhat.com)

- Red Hat Enterprise Linux 8.3.1 リリースノートも併せて参照してください。
- [RHBA-2021:0569](#) アドバイザリーのリリースにより、概要のインプレースアップグレードセクションを更新。

0-1-2

2021年2月12日(金) Lucie Maňásková (lmanasko@redhat.com)

- 既知の問題 (セキュリティー、インストーラー) の2つを追加。

0-1-1

2021年2月10日(水) Lucie Maňásková (lmanasko@redhat.com)

- 既知の問題 (仮想化) を追加。

0-1-0

2021年2月3日(水) Lenka Špačková (lspackova@redhat.com)

- **ip** パラメーター (ネットワーク) のカーネルコマンドラインでネットワーク設定の集約に関する注意書きを追加。
- 非推奨のパッケージに **mercurial** を追加。
- Witherspoon ホスト (仮想化) に関連する既知の問題を追加。

0-0-9

2021年1月29日(金) Lucie Maňásková (lmanasko@redhat.com)

- 新しいバグ修正の説明を追加 (セキュリティー)。
- **mailman** パッケージ (ソフトウェア管理) の非推奨に関する注記を追加。
- 新機能セクションを更新 (セキュリティー、Identity Management)
- **systemd-resolved** サービスに関するテクノロジープレビューの注記を追加。
- その他の若干の更新。

0.0-8

2020年12月14日(月) Lucie Maňásková (lmanasko@redhat.com)

- 既知の問題セクションとバグ修正のセクションを更新。

0.0-7

2020年11月27日(金) Lucie Maňásková (lmanasko@redhat.com)

- **fapolicyd** (セキュリティー) の問題に対するバグ修正が追加されました。
- バグ修正セクションへのその他の更新。

- Podman varlink ベースの REST API V1 の非推奨に関する注記を追加 (コンテナ)
- 新機能セクションを更新。
- **lorax-composer** バックエンドから新しい **osbuild-composer** バックエンドへの Blueprint の複製に関する新しい既知の問題を追加 (Image Builder)。

0.0-6

2020 年 11 月 20 日 (金) Lucie Maňásková (Imanasko@redhat.com)

- OpenSCAP バグ修正の説明を追加 (セキュリティー)。
- 新機能セクションを更新 (ソフトウェア管理)

0.0-5

2020 年 11 月 18 日 (水) Lenka Špačková (lspackova@redhat.com)

- Oracle Linux または CentOS Linux から RHEL に変換するための情報を追加 (概要)。

0.0-4

2020 年 11 月 12 日 (木) Lenka Špačková (lspackova@redhat.com)

- [RHEA-2020:5101](#) アドバイザリーでリリースされた **Node.js 14.15.0** に関する情報が追加。

0.0-3

2020 年 11 月 11 日 (水) Lucie Maňásková (Imanasko@redhat.com)

- Omni-Path Architecture (OPA) ホストソフトウェアのサポートに関する説明を新機能に追加。

0.0-2

2020 年 11 月 9 日 (月) Lenka Špačková (lspackova@redhat.com)

- Intel Tiger Lake グラフィックをテクノロジープレビューとして追加 (Graphics インフラストラクチャー)。

0.0-1

2020 年 11 月 04 日 (水) Lucie Maňásková (Imanasko@redhat.com)

- Red Hat Enterprise Linux 8.3 リリースノートも併せて参照してください。

0.0-0

2020 年 7 月 28 日 (火) Lucie Maňásková (Imanasko@redhat.com)

- Red Hat Enterprise Linux 8.3 Beta リリースノートをリリースしました。