Red Hat Enterprise Linux 9

9.2 リリースノート

Red Hat Enterprise Linux 9.2 リリースノート

Last Updated: 2024-06-24

Red Hat Enterprise Linux 9 9.2 リリースノート

Red Hat Enterprise Linux 9.2 リリースノート

法律上の通知

概要

このリリースノートでは、Red Hat Enterprise Linux 9.2 での改良点および実装された追加機能の概要、このリリースにおける既知の問題などを説明します。また、重要なバグ修正、テクニカルプレビュー、非推奨機能などの詳細も説明します。 Red Hat Enterprise Linux のインストールは、Installation を参照してください。

目次

多様性を受け入れるオープンソースの強化

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

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに関するご意見やご感想をお寄せください。また、改善点があればお知らせください。

Jira からのフィードバック送信 (アカウントが必要)

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

第1章 概要

1.1. RHEL 9.2 における主な変更点

インストーラーおよびイメージの作成

Image Builder の主なハイライト:

- Image Builder On-Prem では、Image Builder Web コンソールでブループリントとイメージを作成するための新しく改良された方法が提供されるようになりました。
- **/etc** ディレクトリー内でのカスタマイズされたファイルとディレクトリーの作成がサポートされるようになりました。
- RHEL for Edge Simplified Installer イメージタイプが Image Builder Web コンソールで使用できるようになりました。

詳細については、新機能 - インストーラーとイメージの作成 を参照してください。

RHEL for Edge

RHEL for Edge の主なハイライト:

- **simplified-installer** イメージのブループリントでのユーザーの指定がサポートされるようになりました。
- Ignition プロビジョニングユーティリティーが、RHEL for Edge でサポートされるようになりました。
- 簡素化されたインストーラーイメージは、ブループリントの FDO カスタマイズセクションなしで作成できるようになりました。

詳細については、新機能 - RHEL for Edge を参照してください。

セキュリティー

セキュリティー関連の主なハイライト:

- **OpenSSL** セキュア通信ライブラリーがバージョン 3.0.7 にリベースされました。
- **SELinux ユーザー空間** パッケージがバージョン 3.5 に更新されました。
- **Keylime** がバージョン 6.5.2 にリベースされました。
- **OpenSCAP** がバージョン 1.3.7 にリベースされました。
- **SCAP セキュリティーガイド** がバージョン 0.1.66 にリベースされました。
- アイドルセッション終了に関する新しいルールが SCAP セキュリティーガイドに追加されました。
- **Clevis** が外部トークンを受け入れるようになりました。
- **Rsyslog** TLS 暗号化ログが複数の CA ファイルをサポートするようになりました。
- Rsyslog 権限は、セキュリティー上の危険を最小限に抑えるために制限されています。
- **fapolicyd** フレームワークは、RPM データベースのフィルタリングを提供するようになりました。

詳細は、新機能 - セキュリティー を参照してください。

動的プログラミング言語、Web サーバー、およびデータベースサーバー
次のアプリケーションストリームの新しいバージョンが利用可能になりました。

- Python 3.11
- nginx 1.22
- PostgreSQL 15

以下のコンポーネントがアップグレードされました。

- Git がバージョン 2.39.1 へ
- Git LFS がバージョン 3.2.0 へ

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

コンパイラーおよび開発ツール
更新されたシステムツールチェーン
RHEL 9.2 では、以下のシステムツールチェインコンポーネントが更新されました。

- GCC 11.3.1
- glibc 2.34
- binutils 2.35.2

パフォーマンスツールとデバッガーの更新
RHEL 9.2 では、以下のパフォーマンスツールおよびデバッガーが更新されました。

- GDB 10.2
- Valgrind 3.19
- SystemTap 4.8
- Dyninst 12.1.0
- elfutils 0.188

更新されたパフォーマンスモニタリングツール
RHEL 9.2 では、以下のパフォーマンス監視ツールが更新されました。

- PCP 6.0.1
- Grafana 9.0.9

更新されたコンパイラーツールセット
次のコンパイラーツールセットが RHEL 9.2 で更新されました。

- GCC Toolset 12
- LLVM Toolset 15.0.7
- Rust Toolset 1.66

- Go Toolset 1.19.6

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

RHEL 9 の Java 実装
RHEL 9 AppStream リポジトリーには、以下が含まれます。

- **java-17-openjdk** パッケージ。OpenJDK 17 Java Runtime Environment および OpenJDK 17 Java Software Development Kit を提供します。
- **java-11-openjdk** パッケージ。OpenJDK 11 Java Runtime Environment および OpenJDK 11 Java Software Development Kit を提供します。
- **java-1.8.0-openjdk** パッケージ。OpenJDK 8 Java Runtime Environment および OpenJDK 8 Java Software Development Kit を提供します。

OpenJDK パッケージの Red Hat ビルドは、ポータブル Linux リリースと RHEL 9.2 以降のリリースの間で単一のバイナリーセットを共有します。この更新により、ソース RPM から RHEL 上で OpenJDK パッケージを再構築するプロセスが変更されました。新しい再構築プロセスの詳細は、README.md ファイルを参照してください。これは Red Hat build of OpenJDK の SRPM パッケージで利用可能であるほか、**java-*-openjdk-headless** パッケージによっても **/usr/share/doc** ツリーの下にインストールされます。

詳細は、OpenJDK のドキュメント を参照してください。

Web コンソール
RHEL Web コンソールは、LUKS で暗号化されたルートボリュームを **NBDE** デプロイメントにバインドするための追加手順を実行するようになりました。

また、グラフィカルインターフェイスを介して、**DEFAULT:SHA1**、**LEGACY:AD-SUPPORT**、および **FIPS:OSPP** の **暗号化サブポリシー** を適用できるようになりました。

詳細については、新機能 - Web コンソール を参照してください。

コンテナー
主な変更点は、以下のとおりです。

- **podman** RHEL システムロールが利用できるようになりました。
- Fulcio および Rekor を使用した sigstore 署名のクライアントが利用できるようになりました。
- Skopeo は、sigstore キーペアの生成をサポートするようになりました。
- Podman は監査用のイベントをサポートするようになりました。
- Container Tools パッケージが更新されました。
- Aardvark および Netavark ネットワークスタックは、カスタム DNS サーバーの選択をサポートするようになりました。
- ツールボックスが利用可能になりました。
- Podman Quadlet は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。
- CNI ネットワークスタックは非推奨になりました。

詳細については、新機能 - コンテナー を参照してください。

1.2. インプレースアップグレード

RHEL 8 から RHEL 9 へのインプレースアップグレード
現在サポートされているインプレースアップグレードパスは次のとおりです。

- 次のアーキテクチャー上の RHEL 8.6 から RHEL 9.0 および RHEL 8.8 から RHEL 9.2:
  - 64 ビット Intel
  - 64 ビット AMD
  - 64-bit ARM
  - IBM POWER 9 (リトルエンディアン)
  - z13 を除く IBM Z アーキテクチャー
- SAP HANA を搭載したシステム上の RHEL 8.6 から RHEL 9.0 および RHEL 8.8 から RHEL 9.2

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

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

SAP HANA で RHEL 9.2 にアップグレードする場合は、アップグレード前に、システムが SAP に対して認定されていることを確認してください。SAP 環境があるシステムでインプレースアップグレードを実行する手順については、SAP 環境を RHEL 8 から RHEL 9 にインプレースアップグレードする方法 を参照してください。

主な機能拡張は、次のとおりです。

- RHEL のインプレースアップグレードパスストラテジーが変更されました。詳細は、Supported in-place upgrade paths for Red Hat Enterprise Linux を参照してください。
- RHEL 9.2 のリリースにより、RHEL 8 から RHEL 9 へのインプレースアップグレードに複数のアップグレードパスが利用できるようになりました。現在のリリースでは、RHEL 8.8 から RHEL 9.2 へ、または RHEL 8.6 から RHEL 9.0 へのインプレースアップグレードを実行できます。利用可能なアップグレードパスは、標準の RHEL システムと SAP HANA を備えた RHEL システムの間で異なることに注意してください。
- **leapp-upgrade-el8toel9** パッケージの最新リリースに、必要な leapp データファイルがすべて含まれるようになりました。これらのデータファイルを手動でダウンロードする必要がなくなりました。
- FIPS モードでの RHEL 8.8 システムの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。
- ターゲットバージョンを含む ISO イメージを使用したインプレースアップグレードが可能になりました。
- RPM 署名がインプレースアップグレード時に自動的にチェックされるようになりました。自動チェックを無効にするには、アップグレードの実行時に **--nogpgcheck** オプションを使用します。
- RHSM にサブスクライブしているシステムが、Red Hat Insights に自動的に登録されるようになりました。自動登録を無効にするには、**LEAPP_NO_INSIGHTS_REGISTER** 環境変数を **1** に設定します。

- Red Hat は、ユーティリティーの使用状況を分析するために、アップグレードの開始時間や終了時間などのアップグレード関連のデータを収集するようになりました。データ収集を無効にするには、**LEAPP_NO_RHSM_FACTS** 環境変数を **1** に設定します。

RHEL7 から RHEL 9 へのインプレースアップグレード
RHEL7 から RHEL 9 へのインプレースアップグレードを直接実行することはできません。ただし、RHEL 7 から RHEL 8 へのインプレースアップグレードを実行してから、RHEL 9 への 2 回目のインプレースアップグレードを実行することはできます。詳細は、RHEL7 から RHEL8 へのアップグレードを参照してください。

1.3. RED HAT CUSTOMER PORTAL LABS

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

- Registration Assistant
- Kickstart Generator
- Red Hat Product Certificates
- Red Hat CVE Checker
- Kernel Oops Analyzer
- Red Hat Code Browser
- VNC Configurator
- Red Hat OpenShift Container Platform Update Graph
- Red Hat Satellite Upgrade Helper
- JVM Options Configuration Tool
- Load Balancer Configuration Tool
- Red Hat OpenShift Data Foundation サポートおよび相互運用性チェッカー
- Ansible Automation Platform Upgrade Assistant
- Ceph Placement Groups (PGs) per Pool Calculator

1.4. 関連情報

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

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

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

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

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

RHEL 8 から RHEL 9 へのインプレースアップグレード を実行する方法は、Upgrading from RHEL 8 to RHEL 9を参照してください。

すべての RHEL サブスクリプションで、既知の技術問題の特定、検証、および解決をプロアクティブに行う **Red Hat Insights** サービスが利用できます。Red Hat Insights クライアントをインストールし、システムをサービスに登録する方法は、Red Hat Insights を使い始める ページを参照してください。

第2章 アーキテクチャー

Red Hat Enterprise Linux 9.2 ではカーネルバージョン 5.14.0-284.11.1 が使用されており、最低限必要なバージョンで以下のアーキテクチャーをサポートします。

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

各アーキテクチャーに適切なサブスクリプションを購入してください。詳細は Get Started with Red Hat Enterprise Linux - additional architectures を参照してください。

第3章 RHEL 9 のコンテンツの配布

3.1. インストール

Red Hat Enterprise Linux 9 は、ISO イメージを使用してインストールします。AMD64、Intel 64 ビット、64 ビット ARM、IBM Power Systems、IBM Z アーキテクチャーで、以下の 2 種類のインストールメディアが利用できます。

- インストール ISO: BaseOS リポジトリーおよび AppStream リポジトリーが含まれ、リポジトリーを追加しなくてもインストールを完了できる完全インストールイメージです。製品のダウンロード ページでは、**インストール ISO** は **バイナリー DVD** と呼ばれます。

注記

インストール用 ISO イメージのサイズは複数 GB であるため、光学メディア形式には適合しない場合があります。インストール ISO イメージを使用して起動可能なインストールメディアを作成する場合は、USB キーまたは USB ハードドライブを使用することが推奨されます。Image Builder ツールを使用すれば、RHEL イメージをカスタマイズできます。Image Builder の詳細は **Composing a customized RHEL system image** を参照してください。

- Boot ISO - インストールプログラムを起動するのに使用する最小限の ISO ブートイメージです。このオプションでは、ソフトウェアパッケージをインストールするのに、BaseOS リポジトリーおよび AppStream リポジトリーにアクセスする必要があります。リポジトリーは、Installation ISO イメージの一部です。インストール中に Red Hat CDN または Satellite に登録して、Red Hat CDN または Satellite から最新の BaseOS および AppStream コンテンツを使用することもできます。

ISO イメージのダウンロード、インストールメディアの作成、RHEL 9 インストールの完了の方法は、標準的な RHEL 9 インストールの実行 を参照してください。自動化したキックスタートインストールなどの高度なトピックは 高度な RHEL 9 インストールの実行 を参照してください。

3.2. リポジトリー

Red Hat Enterprise Linux 9 は、2 つのメインリポジトリー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ーも必要で、すべての RHEL サブスクリプションで利用できます。

BaseOS リポジトリーのコンテンツは、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。詳細は、対象範囲の詳細 を参照してください。

AppStream リポジトリーには、さまざまなワークロードとユースケースに対応するために、ユーザー空間アプリケーション、ランタイム言語、およびデータベースが同梱されます。

また、CodeReady Linux Builder リポジトリーは、すべての RHEL サブスクリプションで利用できます。このリポジトリーは、開発者向けの追加パッケージを提供します。CodeReady Linux Builder リポジトリーに含まれるパッケージは、サポート対象外です。

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

3.3. APPLICATION STREAMS (APPSTREAM)

複数のバージョンのユーザー空間コンポーネントが Application Streams として提供され、BaseOS リポジトリーよりも頻繁に更新されます。これにより、プラットフォームや特定のデプロイメントの基盤となる安定性に影響を及ぼさずに、RHEL をより柔軟にカスタマイズできます。

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

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

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

注記

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

追加の Application Streams バージョンの中には、将来のマイナー RHEL 9 リリースで、ライフサイクルが短いモジュールとして配布されるものがあります。モジュールは、論理ユニット (アプリケーション、言語スタック、データベース、またはツールセット) を表すパッケージの集まりです。これらのパッケージはまとめてビルドされ、テストされ、そしてリリースされます。

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

代替コンパイラーやコンテナーツールなど、迅速な更新を必要とするコンテンツは、代替バージョンを並行して提供しないローリングストリームで利用できます。ローリングストリームは、RPM またはモジュールとしてパッケージ化されることがあります。

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

3.4. YUM/DNF を使用したパッケージ管理

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

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

詳細は、DNF ツールを使用したソフトウェアの管理 を参照してください。

第4章 新機能

ここでは、Red Hat Enterprise Linux 9.2 に追加された新機能および主要な機能拡張を説明します。

4.1. インストーラーおよびイメージの作成

Image Builder Web コンソールでブループリントとイメージを作成するための新しく改良された方法

この機能強化により、イメージビルダーツールの統合バージョンにアクセスできるようになり、ユーザーエクスペリエンスが大幅に向上しました。

Image Builder ダッシュボード GUI の注目すべき機能強化は次のとおりです。

- カーネル、ファイルシステム、ファイアウォール、ロケール、その他のカスタマイズなど、これまで CLI のみでサポートされていたすべてのカスタマイズを使用してブループリントをカスタマイズできるようになりました。
- ブループリントを **.JSON** または **.TOML** 形式でアップロードまたはドラッグすることでブループリントをインポートし、インポートされたブループリントからイメージを作成できます。
- ブループリントを **.JSON** または **.TOML** 形式でエクスポートまたは保存することもできます。
- 並べ替え、フィルタリングが可能で、大文字と小文字が区別されるブループリントリストにアクセスします。
- Image Builder ダッシュボードを使用して、次のタブに移動してブループリント、イメージ、ソースにアクセスできるようになりました。
  - ブループリント - ブループリントタブで、ブループリントをインポート、エクスポート、または削除できるようになりました。
  - イメージ - イメージタブでは、次のことができます。
    - イメージをダウンロードします。
    - イメージログをダウンロードします。
    - イメージを削除します。
  - ソース - ソースタブでは、次のことができます。
    - イメージをダウンロードします。
    - イメージログをダウンロードします。
    - イメージのソースを作成します。
    - イメージを削除します。

Jira:RHELPLAN-139448

/etc ディレクトリーにカスタマイズされたファイルとディレクトリーを作成する機能

この機能強化により、2 つの新しいブループリントのカスタマイズが利用可能になりました。**[[customizations.files]]** および **[[customizations.directories]]** ブループリントのカスタマイズを使用すると、イメージの **/etc** ディレクトリーにカスタマイズされたファイルとディレクトリーを作成で

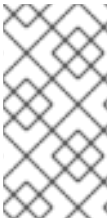
きます。現在、これらのカスタマイズは **/etc** ディレクトリーのみで使用できます。

customizations.directories を使用すると、次のことが可能になります。

- 新しいディレクトリーを作成する
- ディレクトリーのユーザーとグループの所有権を設定する
- モード許可を 8 進数形式で設定する

[[customizations.files]] ブループリントのカスタマイズを使用すると、次のことが可能になります。

- 親の / ディレクトリーの下に新しいファイルを作成する
- 既存のファイルの変更 - これにより、既存のコンテンツが上書きされます。
- 作成しているファイルのユーザーとグループの所有権を設定する
- モード許可を 8 進数形式で設定する



注記

新しいブループリントのカスタマイズは、**edge-container**、**edge-commit** などのすべてのイメージタイプでサポートされます。**edge-raw-image**、**edge-installer**、**edge-simplified-installer** などのインストーラーイメージの作成に使用されるブループリントではカスタマイズはサポートされません。

Jira:RHELPLAN-147428

simplified-installer イメージのブループリントでユーザーを指定する機能

以前は、簡易インストーラーイメージのブループリントを作成するときに、カスタマイズが使用されずに破棄されたため、ブループリントのカスタマイズでユーザーを指定できませんでした。今回の更新により、ブループリントからイメージを作成すると、インストール時にこのブループリントによって **/usr/lib/passwd** ディレクトリーにユーザーが作成され、**/usr/etc/shadow** ディレクトリーにパスワードが作成されます。ブループリント用に作成したユーザー名とパスワードを使用してデバイスにログインできます。システムにアクセスした後、**useradd** コマンドなどでユーザーを作成する必要があることに注意してください。

Jira:RHELPLAN-149091

イメージビルダーで構築された .vhd イメージの 64 ビット ARM のサポート

以前は、イメージビルダーツールで作成された Microsoft Azure **.vhd** イメージは、64 ビット ARM アーキテクチャーではサポートされていませんでした。この更新プログラムでは、64 ビット ARM Microsoft Azure **.vhd** イメージのサポートが追加され、イメージビルダーを使用して **.vhd** イメージを構築し、Microsoft Azure クラウドにアップロードできるようになりました。

Jira:RHELPLAN-139424

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

RHEL 8.4 以降では、**s390utils-base** パッケージは、**s390utils-core** パッケージと補助 **s390utils-base** パッケージに分割されています。そのため、RHEL インストールを **minimal-environment** に設定すると、必要な **s390utils-core** パッケージのみがインストールされ、補助 **s390utils-base** パッケージはイ

ンストールされません。最小限の RHEL インストールで **s390utils-base** パッケージを使用する場合は、RHEL インストールの完了後にパッケージを手動でインストールするか、キックスタートファイルを使用して **s390utils-base** を明示的にインストールする必要があります。

Bugzilla:1932480

4.2. RHEL FOR EDGE

RHEL for Edge 簡略化イメージでの Ignition サポート

この機能強化により、ブループリントをカスタマイズすることによって、Ignition ファイルを簡易インストーラーイメージに追加できるようになります。GUI と CLI の両方で Ignition カスタマイズがサポートされています。RHEL for Edge は、Ignition プロビジョニングユーティリティーを使用して、ブートプロセスの初期段階でユーザー設定をイメージに挿入します。最初の起動時に、Ignition はリモート URL または簡易インストーラーイメージに埋め込まれたファイルから設定を読み取り、その設定をイメージに適用します。

Jira:RHELPLAN-139659

簡素化されたインストーラーイメージは、ブループリントの FDO カスタマイズセクションなしで作成できるようになりました。

以前は、RHEL for Edge Simplified Installer イメージを構築するには、FIDO デバイスオンボーディング (FDO) カスタマイズセクションに詳細を追加する必要がありました。そうしないと、イメージのビルドが失敗します。この更新により、ブループリントでの FDO カスタマイズがオプションになり、エラーなしで RHEL for Edge Simplified Installer イメージをビルドできるようになりました。

Jira:RHELPLAN-139655

RHEL for Edge イメージ用の MicroShift 有効化の Red Hat ビルド

この機能強化により、RHEL for Edge システムで Red Hat build of MicroShift サービスを有効にすることができます。**customizations.firewalld.zones** ブループリントカスタマイズを使用すると、ブループリントカスタマイズに **firewalld** ソースのサポートを追加できます。そのためには、ゾーンの名前とその特定のゾーン内のソースのリストを指定します。ソースは、**source[/mask]|MAC|ipset:ipset** の形式にすることができます。

以下は、RHEL for Edge システムで Red Hat build of MicroShift サービスのサポートを設定およびカスタマイズする方法に関するブループリントの例です。

```
[[packages]]
name = "microshift"
version = "*"
[customizations.services]
enabled = ["microshift"]
[[customizations.firewall.zones]]
name = "trusted"
sources = ["10.42.0.0/16", "169.254.169.1"]
```

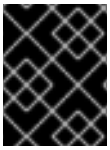
Red Hat build of MicroShift のインストール要件 (ファイアウォールポリシー、MicroShift RPM、**systemd** サービスなど) を使用すると、実稼働環境にすぐに使用できるデプロイメントを作成して、現場でデプロイされた最小限のエッジデバイスへのワークロードの移植性、およびデフォルトでの LVM デバイスマッパーの有効化を実現できます。

Jira:RHELPLAN-136489

4.3. ソフトウェア管理

RHEL でのオフライン更新のための新しい dnf offline-upgrade コマンド

この機能強化により、DNF **system-upgrade** プラグインの新しい **dnf offline-upgrade** コマンドを使用して、オフライン更新を RHEL に適用できるようになります。



重要

system-upgrade プラグインに含まれる **dnf system-upgrade** コマンドは、RHEL ではサポートされていません。

Bugzilla:2131288

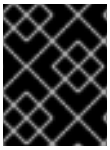
dnf offline-upgrade へのアドバイザリーセキュリティーフィルターの適用がサポートされるようになりました。

この機能強化により、アドバイザリーフィルタリングの新しい機能が追加されました。その結果、**dnf offline-upgrade** コマンドをアドバイザリーセキュリティーフィルター (**--advisory**、**--security**、**--bugfix**、およびその他のフィルター) とともに使用することにより、指定されたアドバイザリーのみからパッケージとその依存関係をダウンロードできるようになりました。

Bugzilla:2139326

unload_plugins 関数が DNF API で使用できるようになりました。

この機能強化により、プラグインのアンロードを可能にする新しい **unload_plugins** 関数が DNF API に追加されました。



重要

最初に **init_plugins** 関数を実行してから、**unload_plugins** 関数を実行する必要があることに注意してください。

Bugzilla:2121662

rpm2archive の新しい --nocompression オプション

この機能強化により、**--nocompression** オプションが **rpm2archive** ユーティリティーに追加されました。このオプションを使用すると、RPM パッケージを直接解凍するときに圧縮を回避できます。

Bugzilla:2150804

4.4. シェルおよびコマンドラインツール

ReaR は 64 ビット IBM Z アーキテクチャーでも完全にサポートされるようになりました。

Basic Relax and Recover (ReaR) 機能は、64 ビット IBM Z アーキテクチャーでテクノロジープレビューとして以前に利用可能でしたが、**rear** パッケージのバージョン 2.6-17.el9 以降で完全にサポートされています。ReaR レスキューイメージは、z/VM 環境の IBM Z アーキテクチャー上のみで作成できます。論理パーティション (LPAR) のバックアップとリカバリーは、現時点ではサポートされていません。ReaR は、Extended Count Key Data (ECKD) ダイレクトアクセスストレージデバイス (DASD) 上のみでディスクレイアウトの保存と復元をサポートします。ファイバーチャネルプロトコル (FCP) を介して接

続された固定ブロックアクセス (FBA) DASD および SCSI ディスクは、この目的ではサポートされていません。現在利用可能な唯一の出力方法は、初期プログラムロード (IPL) です。これは、**zIPL** ブートローダーと互換性のあるカーネルと初期 RAM ディスク (initrd) を生成します。

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

Bugzilla:2046653

systemd がバージョン 252 にリベース

systemd パッケージがバージョン 252 にアップグレードされました。主な変更点は、以下のとおりです。

- **system.conf** ファイルおよび **user.conf** ファイルの **DefaultDeviceTimeoutSec=** オプションを使用して、デバイスユニットのアクティブ化を待機する際にデフォルトのタイムアウトを指定できます。
- シャットダウン時に、**systemd** はファイルシステムのアンマウントをブロックするプロセスに関するログを記録するようになりました。
- 一時的なユニットにもドロップインを使用できるようになりました。
- **ConditionMemory=** オプションで、K、M、G、T などのサイズの接尾辞を使用できます。
- **systemctl list-automounts** コマンドを使用すると、自動マウントポイントを一覧表示できます。
- **systemd-logind** ユーティリティーを使用し、**StopIdleSessionSec=** オプションを使用して事前設定されたタイムアウト後にアイドルセッションを停止できます。
- **systemd-udev** ユーティリティーは、Infiniband verb デバイスの **infiniband by-path** および **infiniband by-ibdev** リンクを作成するようになりました。
- **systemd-tmpfiles** ユーティリティーは、存在しない **C** コピーのソースを適切に処理するようになりました。
- **systemd-repart** ユーティリティーは、署名を含む **dm-verity** パーティションを生成するようになりました。

Bugzilla:2217931

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

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

Renaming IPoIB devices の手順に従って、InfiniBand インターフェイスの名前にカスタム命名ルールを定義できます。

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

Bugzilla:2136937

4.5. インフラストラクチャーサービス

chrony がバージョン 4.3 にリベースされました

chrony スイートがバージョン 4.3 に更新されました。バージョン 4.2 からの主な機能強化は、以下のとおりです。

- ネットワークタイムプロトコル (NTP) 測定の長期分位数ベースのフィルタリングを追加しました。この機能を有効にするには、**maxlayquant** オプションを **pool**、**server**、または **peer** ディレクティブに追加します。
- ソースの **chronyd** 選択に関する詳細情報を提供するために、選択ログを追加しました。選択ログを有効にするには、**log** ディレクティブに **selection** オプションを追加します。
- ハードウェアタイムスタンプおよび Pulse-Per-Second ハードウェアクロック (PHC) 基準クロックを使用する場合の同期の安定性が向上しました。
- 温度補償水晶発振器 (TCXO)、オーブン制御水晶発振器 (OCXO)、原子時計などのフリーランニング安定クロックを使用したシステムクロック安定化のサポートが追加されました。
- 最大ポーリングレートが 1 秒あたり 128 メッセージに増えました。

Bugzilla:2133754

frr がバージョン 8.3.1 にリベースされました。

動的ルーティングスタックを管理するための **frr** パッケージがバージョン 8.3.1 に更新されました。バージョン 8.2.2 への主な変更点は、以下のとおりです。

- ボーダーゲートウェイプロトコル (BGP) とやりとりするための新しいコマンドセットを追加しました。
  - BGP ルートの Autonomous System (AS) パス属性を新しい値に置き換える **set as-path replace** コマンド。
  - BGP ルートマップの設定時に、特定の BGP ピアまたはグループを照合するには、**matchpeer** コマンドを使用します。
  - **ead-es-frag evi-limit** コマンドを使用して、EVPN で一定期間内に送信できる EVI フラグメントごとのイーサネット AD の数に制限を設定します。
  - **match evpn route-type** コマンドを使用すると、ルートターゲット、ルート識別子、または MAC/IP ルートなど、特定のタイプの EVPN ルートに対して特定のアクションを実行できます。
- FRR デーモンとやりとりするための **show thread timers** コマンドが VTYSH コマンドラインインターフェイスに追加されました。
- OSPF プロトコルを通じて現在到達可能なルーターのリストを表示する **show ip ospf reverseable-routers** コマンドが追加されました。
- Protocol Independent Multicast (PIM) デーモンとやりとりするための新しいコマンドが追加されました。
  - **debug igmp trace detail** コマンドを使用すると、詳細なトレースによるインターネットグループ管理プロトコル (IGMP) メッセージのデバッグが可能になります。
  - インターフェイスを passive と設定し、PIM メッセージを送信しないようにする **ip pim passive** コマンド

- ECMP、EVPN、MPLS ステータスなどの **show zebra** コマンドの新しい出力が追加されました。
- カーネルの **mroute** テーブルからマルチキャスト関連情報を表示するための **show ip nht mrib** コマンドが ZEBRA コンポーネントに追加されました。

Bugzilla:2129731

vsftpd がバージョン 3.0.5 にリベースされました。

Very Secure FTP Daemon (**vsftpd**) はホスト間でファイルを転送するセキュアな方法を提供します。**vsftpd** パッケージがバージョン 3.0.5 に更新されました。以下の SSL モダナイゼーションを含む変更および機能強化がおこなわれています。

- デフォルトでは、**vsftpd** ユーティリティーには、セキュアな接続のために、TLS バージョン 1.2 以降の使用が必要になりました。
- **vsftpd** ユーティリティーが最新の FileZilla クライアントと互換性を持つようになりました。

Bugzilla:2018284

frr パッケージには、ターゲットを絞った SELinux ポリシーが含まれるようになりました。

ダイナミックルーティングスタックを管理するための **frr** パッケージの急速な開発により、新機能とアクセスベクターキャッシュ (AVC) の問題が頻繁に発生しました。この機能強化により、SELinux ルールが FRR とともにパッケージ化され、問題に迅速に対処できるようになりました。SELinux は、必須のアクセス制御ポリシーを強制することで、パッケージに追加の保護レベルを追加します。

Bugzilla:2129743

powertop がバージョン 2.15 にリベースされました。

エネルギー効率を向上させる **powertop** パッケージがバージョン 2.15 に更新されました。注目すべき変更点と機能強化は次のとおりです。

- **powertop** ツールの安定性を向上させるために、いくつかの Valgrind エラーとバッファーオーバーランの可能性が修正されました。
- Ryzen プロセッサーおよび Kaby Lake プラットフォームとの互換性が向上しました。
- Lake Field、Alder Lake N、および Raptor Lake プラットフォームのサポートが可能になりました。
- Ice Lake NNPI および Meteor Lake のモバイルおよびデスクトップのサポートを有効にしました。

Bugzilla:2044132

systemd-sysusers ユーティリティーは、chrony、dhcp、radvd、および quid パッケージで使用できます。

systemd-sysusers ユーティリティーは、パッケージのインストール時にシステムユーザーとグループを作成し、パッケージの削除時にそれらを削除します。この機能強化により、次のパッケージのスクリプトレットに **systemd-sysusers** ユーティリティーが含まれるようになりました。

- **chrony**
- **dhcp**

- **radvd**
- **squid**

Jira:RHELPLAN-136485

周波数同期用の新しい synce4l パッケージが利用可能になりました。

SyncE (同期イーサネット) は、PTP クロックが物理層で周波数の正確な同期を達成できるようにするハードウェア機能です。SyncE は、特定のネットワークインターフェイスカード (NIC) およびネットワークスイッチでサポートされています。

この機能強化により、SyncE のサポートを提供する新しい **synce4l** パッケージが利用できるようになりました。その結果、Telco Radio Access Network (RAN) アプリケーションは、より正確な時刻同期により、より効率的な通信を実現できるようになりました。

Bugzilla:2143264

tuned がバージョン 2.20.0 にリベースされました。

アプリケーションとワークロードのパフォーマンスを最適化するための TuneD ユーティリティーがバージョン 2.20.0 に更新されました。バージョン 2.19.0 に対する主な変更点および機能強化は、以下のとおりです。

- API の拡張機能により、実行時にプラグインインスタンス間でデバイスを移動できるようになります。
- CPU 関連のパフォーマンス設定を微調整する **plugin_cpu** モジュールには、次の機能強化が導入されています。
  - **pm_qos_resume_latency_us** 機能を使用すると、各 CPU がアイドル状態からアクティブ状態に移行するまでに許可される最大時間を制限できます。
  - TuneD は、さまざまな使用シナリオに基づいてシステムの電源管理を調整するためのスケーリングアルゴリズムを提供する **intel_pstate** スケーリングドライバーのサポートを追加します。
- Unix ドメインソケットを通じて TuneD を制御するソケット API がテクノロジープレビューとして利用可能になりました。詳細については、テクノロジープレビューとして利用可能な TuneD 用の Socket API を参照してください。

Bugzilla:2133815、Bugzilla:2113925、Bugzilla:2118786、Bugzilla:2095829

4.6. セキュリティー

Libreswan が 4.9 にリベースされました。

libreswan パッケージがバージョン 4.9 にアップグレードされました。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。

- **addconn** および **whack** ユーティリティーに対する **{left,right}pubkey=** オプションのサポート
- KDF セルフテスト
- ホストの認証キーを表示 (**showhostkey**):
  - ECDSA 公開鍵のサポート

    - PEM エンコードされた公開鍵を出力するための新しい **--pem** オプション

- Internet Key Exchange プロトコルバージョン 2 (IKEv2):
    - 拡張認証プロトコル – トランスポート層セキュリティー (EAP-TLS) のサポート
    - EAP のみの認証のサポート
- **pluto** IKE デーモン:
    - **maxbytes** カウンターと **maxpacket** カウンターのサポート

Bugzilla:2128669

OpenSSL が 3.0.7 にリベースされました。

OpenSSL パッケージがバージョン 3.0.7 にリベースされ、さまざまなバグ修正と拡張機能が含まれています。最も注目すべき点は、デフォルトのプロバイダーに **RIPEMD160** ハッシュ関数が含まれるようになった点です。

Bugzilla:2129063

libssh がスマートカードをサポートするようになりました。

Public-Key Cryptography Standard (PKCS) #11 Uniform Resource Identifier (URI) を通じてスマートカードを使用できるようになりました。その結果、スマートカードを **libssh** SSH ライブラリーおよび **libssh** を使用するアプリケーションで使用できるようになります。

Bugzilla:2026449

libssh が 0.10.4 にリベースされました。

セキュアなリモートアクセスとマシン間のファイル転送のための SSH プロトコルを実装する **libssh** ライブラリーがバージョン 0.10.4 に更新されました。

新機能:

- OpenSSL 3.0 のサポートが追加されました。
- スマートカードのサポートが追加されました。
- 2 つの新しい設定オプション **IdentityAgent** と **ModuliFile** が追加されました。

その他の主な変更点は次の通りです。

- OpenSSL バージョン 1.0.1 より古いものはサポートされなくなりました。
- デフォルトでは、デジタル署名アルゴリズム (DSA) のサポートはビルド時に無効になっています。
- SCP API は非推奨になりました。
- **pubkey** API と **privatekey** API は非推奨になりました。

Bugzilla:2068475

SELinux ユーザー空間パッケージが 3.5 に更新されました。

SELinux ユーザー空間パッケージ **libselinux**、**libsepol**、**libsemanage**、**checkpolicy**、**mcstrans**、および **sepolicy** ユーティリティーを含む **policycoreutils** がバージョン 3.5 に更新されました。以下は、主な機能強化およびバグ修正です。

- **sepolicy** ユーティリティー:
  - 欠落していたブール値を man ページに追加しました。
  - Python と GTK の複数の更新
- **PCRE2** ライブラリーによるヒープメモリーの使用量を削減する回避策を **libselinux** に追加しました。
- **libsepol** パッケージ:
  - カーネルポリシーのタイプ AV ルールの属性を拒否します。
  - 空のクラス定義を書き込まなくなり、簡単なラウンドトリップテストが可能になりました。
  - より厳格なポリシー検証
- **fixfiles** スクリプトは、**SIGINT** シグナル上の一時的なバインドマウントをアンマウントします。
- 多くのコードとスペルのバグが修正されました。
- 非推奨の Python モジュール **distutils** と PIP を使用したインストールへの依存関係を削除しました。
- **semodule** オプション **--rebuild-if-modules-changed** の名前が **--refresh** に変更されました。
- 生成された説明の翻訳が更新され、サポートされていない言語の処理が改善されました。
- 多くの静的コード分析のバグ、ファザーの問題、コンパイラーの警告を修正しました。

Bugzilla:2145224、Bugzilla:2145228、Bugzilla:2145229、Bugzilla:2145226、Bugzilla:2145230、Bugzilla:214

OpenSCAP が 1.3.7 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 1.3.7 にリベースされました。このバージョンは、さまざまなバグ修正と機能拡張を提供します。特に、次のとおりです。

- OVAL フィルター処理時のエラーを修正しました (RHBZ#2126882)。
- XPath が一致しない場合、OpenSCAP は無効な空の **xmlfilecontent** 項目を出力しなくなりました (RHBZ#2139060)。
- **Failed to check available memory** エラーの出力を回避しました (RHBZ#2111040)。

Bugzilla:2159286

SCAP セキュリティーガイドが 0.1.66 にリベースされました。

SCAP セキュリティーガイド (SSG) パッケージがアップストリームバージョン 0.1.66 にリベースされました。このバージョンは、さまざまな拡張機能とバグ修正を提供します。特に、次のようなものがあります。

- 新しい CIS RHEL9 プロファイル

- ルール **account_passwords_pam_faillock_audit** の廃止、代わりに **accounts_passwords_pam_faillock_audit** の使用

Bugzilla:2158405

アイドルセッション終了の新しい SCAP ルール

新しい SCAP ルール **logind_session_timeout** が拡張レベルおよび高レベルの ANSSI-BP-028 プロファイルの **scap-security-guide** パッケージに追加されました。このルールは、**systemd** サービスマネージャーの新機能を使用し、一定時間が経過すると、アイドル状態のユーザーセッションを終了します。このルールは、複数のセキュリティーポリシーで必要とされる堅牢なアイドルセッション終了メカニズムの自動設定を提供します。その結果、OpenSCAP はアイドル状態のユーザーセッションの終了に関連するセキュリティー要件を自動的にチェックし、必要に応じて修正できます。

Bugzilla:2122325

Rsyslog ログファイルの scap-security-guide ルールは、RainerScript ログと互換性があります。

Rsyslog ログファイルの所有権、グループ所有権、およびアクセス許可を確認および修正するための **scap-security-guide** のルールは、RainerScript 構文とも互換性があるようになりました。最新のシステムはすでに Rsyslog 設定ファイルで RainerScript 構文を使用していますが、それぞれのルールはこの構文を認識できませんでした。その結果、**scap-security-guide** ルールは、使用可能な両方の構文で、Rsyslog ログファイルの所有権、グループ所有権、およびアクセス許可をチェックして修正できるようになりました。

Bugzilla:2169414

Keylime が 6.5.2 にリベースされました。

keylime パッケージがアップストリームバージョン keylime-6.5.2-5.el9 にリベースされました。このバージョンには、さまざまな機能強化とバグ修正が含まれていますが、特に注目すべき点は次のとおりです。

- 脆弱性 CVE-2022-3500 に対処しました。
- Keylime エージェントは、あるスクリプトが別のスクリプトのすぐ後に実行された場合でも、IMA 認証に失敗しなくなりました (RHBZ#2138167)。
- **/usr/share/keylime/create_mb_refstate** スクリプトのセグメンテーション違反を修正しました (RHBZ#2140670)。
- **require_ek_cert** オプションが有効になっている場合、EK 検証中にレジストラーがクラッシュしなくなりました (RHBZ#2142009)。

Bugzilla:2150830

Clevis は外部トークンを受け入れます。

Clevis 自動暗号化ツールに導入された新しい **-e** オプションを使用すると、外部トークン ID を指定して、**cryptsetup** 中にパスワードを入力する必要がなくなります。この機能により、設定プロセスがより自動化され便利になり、特に Clevis を使用する **stratis** などのパッケージに役立ちます。

Bugzilla:2126533

Rsyslog TLS 暗号化ログが複数の CA ファイルをサポートするようになりました。

新しい **NetstreamDriverCaExtraFiles** ディレクティブを使用すると、TLS 暗号化リモートログ用の追加の認証局 (CA) ファイルのリストを指定できます。新しいディレクティブは、**ossl** (OpenSSL) Rsyslog ネットワークストリームドライバーのみで使用できることに注意してください。

Bugzilla:2124849

Rsyslog 権限は制限されています。

Rsyslog ログ処理システムの権限は、Rsyslog によって明示的に必要な権限のみに制限されるようになりました。これにより、ネットワークプラグインなどの入力リソースに潜在的なエラーが発生した場合のセキュリティーの危険が最小限に抑えられます。その結果、Rsyslog は同じ機能を持ちますが、不必要な権限を持ちません。

Bugzilla:2127404

SELinux ポリシーにより、Rsyslog が起動時に権限を削除できるようになります。

Rsyslog ログ処理システムの権限は、セキュリティーの危険を最小限に抑えるためにさらに制限されているため (RHBZ#2127404)、SELinux ポリシーが更新され、**rsyslog** サービスが開始時に権限を削除できるようになりました。

Bugzilla:2151841

Tang は systemd-sysusers を使用するようになりました。

Tang ネットワークプレゼンスサーバーは、**useradd** コマンドを含むシェルスクリプトの代わりに、**systemd-sysusers** サービスを通じてシステムユーザーとグループを追加するようになりました。これにより、システムユーザーリストのチェックが簡素化され、**sysuser.d** ファイルに優先順位を付けてシステムユーザーの定義を上書きすることもできます。

Bugzilla:2095474

opencryptoki が 3.19.0 にリベースされました。

opencryptoki パッケージがバージョン 3.19.0 にリベースされ、多くの機能強化とバグ修正が提供されています。最も注目すべき点は、**opencryptoki** が次の機能をサポートするようになったということです。

- IBM 固有の Dilithium キー
- 二重機能暗号機能
- PKCS #11 暗号化トークンインターフェイスの基本仕様 v3.0 で説明されているように、新しい **C_SessionCancel** 関数を使用して、アクティブなセッションベースの操作をキャンセルする
- **CKM_IBM_ECDSA_OTHER** メカニズムによる Schnorr 署名
- **CKM_IBM_BTC_DERIVE** メカニズムによるビットコイン鍵の導出
- IBM z16 システムの EP11 トークン

Bugzilla:2110314

SELinux は mptcpd と udftools を制限するようになりました。

selinux-policy パッケージの今回の更新により、SELinux は次のサービスを制限します。

- **mptcpd**

- **udftools**

Bugzilla:1972222

fapolicyd は RPM データベースのフィルタリングを提供するようになりました。

新しい設定ファイル **/etc/fapolicyd/rpm-filter.conf** を使用すると、**fapolicyd** ソフトウェアフレームワークが信頼データベースに保存する RPM データベースファイルのリストをカスタマイズできます。これにより、RPM によってインストールされた特定のアプリケーションをブロックしたり、デフォルトの設定フィルターによって拒否されたアプリケーションを許可したりできます。

Jira:RHEL-192

GnuTLS は復号化および暗号化中にパディングを追加および削除できます。

特定のプロトコルの実装では、復号化および暗号化中に PKCS#7 パディングが必要です。パディングを透過的に処理するために、**gnutls_cipher_encrypt3** および **gnutls_cipher_decrypt3** ブロック暗号関数が GnuTLS に追加されました。その結果、これらの関数を **GNUTLS_CIPHER_PADDING_PKCS7** フラグと組み合わせて使用し、元のプレーンテキストの長さがブロックサイズの倍数でない場合にパディングを自動的に追加または削除できるようになりました。

Bugzilla:2084161

NSS が 1023 ビット未満の RSA 鍵に対応しなくなる

Network Security Services (NSS) ライブラリーの更新により、すべての RSA 操作の最小鍵サイズが 128 から 1023 ビットに変更されます。つまり、NSS は以下の機能を実行しなくなります。

- RSA 鍵の生成は 1023 ビット未満です。
- 1023 ビット未満の RSA 鍵で RSA に署名するか、署名を検証します。
- 1023 ビットより短い RSA キーで値を暗号化または復号化します。

Bugzilla:2091905

Extended Master Secret TLS エクステンションが FIPS 対応システムに適用されるようになりました。

RHSA-2023:3722 アドバイザリーのリリースにより、FIPS 対応 RHEL 9 システム上の TLS 1.2 接続に、TLS **Extended Master Secret** (EMS) エクステンション (RFC 7627) エクステンションが必須になりました。これは FIPS-140-3 要件に準拠しています。TLS 1.3 は影響を受けません。

EMS または TLS 1.3 をサポートしていないレガシークライアントは、RHEL 9 で実行されている FIPS サーバーに接続できなくなりました。同様に、FIPS モードの RHEL 9 クライアントは、EMS なしでは TLS 1.2 のみをサポートするサーバーに接続できません。これは実際には、これらのクライアントが RHEL 6、RHEL 7、および RHEL 以外のレガシーオペレーティングシステム上のサーバーに接続できないことを意味します。これは、OpenSSL のレガシー 1.0.x バージョンが EMS または TLS 1.3 をサポートしていないためです。

さらに、ハイパーバイザーが EMS なしで TLS 1.2 を使用する場合は、FIPS 対応 RHEL クライアントから VMWare ESX などのハイパーバイザーへの接続が **Provider routines::ems not enabled** エラーで失敗するようになりました。この問題を回避するには、EMS 拡張で TLS 1.3 または TLS 1.2 をサポートするようにハイパーバイザーを更新します。VMWare vSphere の場合、これはバージョン 8.0 以降を意味します。

詳細は、TLS Extension "Extended Master Secret" enforced with Red Hat Enterprise Linux 9.2 を参照してください。

Bugzilla:2188046、Bugzilla:2218721

4.7. ネットワーク

NetworkManager がバージョン 1.42.2 にリベースされました。

NetworkManager パッケージがアップストリームバージョン 1.42.2 にアップグレードされ、以前のバージョンに対するバグ修正や機能強化が数多く追加されました。

- イーサネットボンドはソースロードバランシングをサポートします。
- NetworkManager は、**loopback** デバイスで接続を管理できます。
- IPv4 Equal-Cost Multi-Path (ECMP) ルートのサポートが追加されました。
- 仮想ローカルエリアネットワーク (VLAN) 接続での **802.1ad** タグ付けのサポートが追加されました。
- **nmtui** アプリケーションは、Wi-Fi WPA-Enterprise、802.1X 認証を備えたイーサネット、および MACsec 接続プロファイルをサポートします。
- すべてのアドレスが IPv6 重複アドレス検出 (DAD) に失敗した場合、NetworkManager は DHCPv6 リースを拒否します。

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。

Bugzilla:2134897

NetworkManager を使用した ECMP ルーティングの weight プロパティーの導入

この更新により、RHEL 9 は、IPv4 等コストマルチパス (ECMP) ルートを定義する際の新しいプロパティーの **weight** をサポートします。NetworkManager を使用してマルチパスルーティングを設定し、ネットワークトラフィックの負荷分散と安定化を行うことができます。これにより、2 つのノード間のデータ送信に複数のパスを使用できるようになり、ネットワーク効率が向上し、リンク障害が発生した場合に冗長性が提供されます。**weight** プロパティーを使用するための条件は次のとおりです。

- 有効な値は、1 ～ 256 です。
- **weight** プロパティーを使用して、複数のネクストホップルートをシングルホップルートとして定義します。
- **weight** を設定しない場合、NetworkManager はルートを ECMP ルートにマージできません。

Bugzilla:2081302

NetworkManager の更新により、複数のネットワークにわたる DNS 設定の柔軟性が向上しました。

この更新により、**/etc/Networkmanager/NetworkManager.conf** ファイルの既存の **global-dns** セクションを使用して、**[global-dns-domain-*]** セクションで **nameserver** の値を指定せずに DNS オプションを設定できるようになります。これにより、実際の DNS 解決のためにネットワーク接続によって提供される DNS サーバーに依存しながら、**/etc/resolv.conf** ファイルで DNS オプションを設定できます。その結果、この機能により、異なる DNS サーバーを使用して異なるネットワークに接続する際の DNS 設定の管理がより簡単かつ柔軟になります。特に、**/etc/resolv.conf** ファイルを使用して DNS オプションを設定する場合はそうです。

Bugzilla:2019306

NetworkManager が新しい vlan.protocol プロパティーをサポートするようになりました。

今回の更新により、**vlan** インターフェイスタイプは、新しい **protocol** プロパティーを受け入れるようになりました。プロパティータイプは文字列です。受け入れられる値は **802.1Q** (デフォルト) または **802.1ad** のいずれかです。新しいプロパティーは、カプセル化のタグ識別子を制御する VLAN プロトコルを指定します。

Bugzilla:2128809

NetworkManager で、アンマネージドインターフェイスを介した VLAN 設定が可能になりました。

この機能拡張により、NetworkManager で仮想 LAN (VLAN) を設定するときに、アンマネージドネットワークインターフェイスをベースインターフェイスとして使用できるようになります。その結果、VLAN ベースインターフェイスは、**nmcli device set enp1s0 managed true** コマンドまたは NetworkManager の他の API を通じて明示的に変更されないかぎり、そのまま残ります。

Bugzilla:2110307

NetworkManager を使用したマルチパス TCP の設定が完全にサポートされるようになりました。

今回の更新で、NetworkManager ユーティリティーが Multipath TCP (MPTCP) 機能を提供するようになりました。**nmcli** コマンドを使用して MPTCP を制御し、その設定を永続化できます。

詳細は以下を参照してください。

- Understanding Multipath TCP: High availability for endpoints and the networking highway of the future
- RFC 8684: TCP Extensions for Multipath Operation with Multiple Addresses
- MPTCP アプリケーションの複数パスの永続的な設定

Bugzilla:2029636

NetworkManager ユーティリティーは、loopback インターフェイスでの接続のアクティブ化をサポートするようになりました。

管理者は、**loopback** インターフェイスを管理して、以下を行うことができます。

- 追加の IP アドレスを **loopback** インターフェイスに追加する
- DNS 設定を定義する
- インターフェイスにバインドしない特別なルートを定義する
- インターフェイスに関係しないルートルールを定義する
- **loopback** インターフェイスの最大伝送単位 (MTU) サイズを変更する

Bugzilla:2073512

balance-slb ボンディングモードがサポートされるようになりました。

新しい **balance-slb** ボンディングモードソースロードバランシングには、スイッチ設定は必要ありません。**balance-slb** は、**xmit_hash_policy=vlan+srcmac** を使用して、ソースイーサネットアドレスのトラフィックを分割し、NetworkManager はトラフィックフィルタリングに必要な **nftables** ルールを追

加します。その結果、NetworkManager を使用して、**balance-slb** オプションを有効にして結合プロファイルを作成できるようになりました。

Bugzilla:2128216

firewalld が バージョン 1.2 にリベースされました。

firewalld パッケージがバージョン 1.2 にアップグレードされ、複数の機能強化が提供されています。主な変更点は、以下のとおりです。

- 新しいサービスのサポート (netdata、IPFS など)
- フェールセーフモードにより、**firewalld** サービスの起動中にエラーが発生した場合でも、システムが保護された状態を維持し、ネットワーク通信が中断されないようにします。
- 一部の **firewalld** ポリシーコマンドのコマンドライン (CLI) でのタブ補完

Bugzilla:2125371

firewalld は起動時のフェールセーフメカニズムをサポートするようになりました。

この機能強化により、起動に失敗した場合、**firewalld** はフェイルセーフのデフォルトに戻ります。この機能は、無効な設定やその他の起動の問題が発生した場合にホストを保護します。その結果、ユーザー設定が無効であっても、**firewalld** を実行しているホストは起動時にフェイルセーフになるようになりました。

Bugzilla:2077512

conntrack-tools がバージョン 1.4.7 にリベースされました。

conntrack-tools パッケージがバージョン 1.4.7 にアップグレードされ、複数のバグ修正と拡張機能が提供されています。主な変更点は、以下のとおりです。

- **IPS_HW_OFFLOAD** フラグを追加。これは、**conntrack** エントリーのハードウェアへのオフロードを指定します。
- **clash_resolve** および **chaintoolong** 統計カウンターを追加
- IP アドレスファミリーによるイベントのフィルタリングをサポート
- **conntrackd.conf** ファイル内で、yes または no を on または off の同義語として受け入れる
- デーモン起動時のユーザー空間ヘルパーの自動読み込みをサポート。ユーザーは **nfct add helper** コマンドを手動で実行する必要はありません。
- **-o userspace** コマンドオプションを削除し、ユーザー空間でトリガーされたイベントに常にタグ付け
- 外部注入の問題を警告のみとしてログに記録
- キャッシュエントリーを検索するときに conntrack ID を無視し、スタックした古いエントリーを置き換えることができるようにする
- **ssdp cthelper** モジュールでの IPv6 **M-SEARCH** リクエストの解析の破損を修正
- **nfct** ライブラリーでの遅延バインディング手法の必要性を排除
- プロトコル値の解析をサニタイズし、無効な値をキャッチ

Bugzilla:2132398

nmstate API が IPv6 リンクローカルアドレスを DNS サーバーとしてサポートするようになりました。

この機能強化により、**nmstate** API を使用して、IPv6 link-local アドレスを DNS サーバーとして設定できるようになりました。たとえば、**<link-local_address>%<interface>** 形式を使用します。

```
dns-resolver:
  config:
    server:
    - fe80::deef:1%enp1s0
```

Bugzilla:2095207

nmstate API が MPTCP フラグをサポートするようになりました。

この更新では、MultiPath TCP (MPTCP) フラグのサポートにより **nmstate** API が強化されています。その結果、**nmstate** を使用して、静的または動的 IP アドレスを持つインターフェイスに MPTCP アドレスフラグを設定できます。

Bugzilla:2120473

すべてのインターフェイスの MTU に min-mtu および max-mtu プロパティーが追加されました。

以前は、例外メッセージが明確ではなく、サポートされている MTU 範囲を理解できませんでした。この更新では、すべてのインターフェイスに **min-mtu** および **max-mtu** プロパティーが導入されます。その結果、必要な MTU が範囲外の場合、**nmstate** はサポートされる MTU 範囲を示します。

Bugzilla:2044150

NetworkManager で、アンマネージドインターフェイスを介した VLAN 設定が可能になりました。

この機能拡張により、NetworkManager で仮想 LAN (VLAN) を設定するときに、アンマネージドネットワークインターフェイスをベースインターフェイスとして使用できるようになります。その結果、VLAN ベースインターフェイスは、**nmcli device set enp1s0 managed true** コマンドまたは NetworkManager の他の API を通じて明示的に変更されないかぎり、そのまま残ります。

Bugzilla:2058292

balance-slb ボンディングモードがサポートされるようになりました。

新しい **balance-slb** ボンディングモードソースロードバランシングには、スイッチ設定は必要ありません。**balance-slb** は、**xmit_hash_policy=vlan+srcmac** を使用して、送信元イーサネットアドレス上のトラフィックを分割し、NetworkManager はトラフィックフィルタリングに必要な **nftables** ルールを追加します。その結果、NetworkManager を使用して、**balance-slb** オプションを有効にして結合プロファイルを作成できるようになりました。

Bugzilla:2130240

Nmstate の新しい weight プロパティー

この更新では、Nmstate API およびツールスイートに **weight** プロパティーが導入されました。**weight** を使用して、Equal Cost Multi-Path (ECMP) ルートグループ内の各パスの相対的な weight を指定できます。weight は 1 - 256 の間の数値です。その結果、Nmstate の **weight** プロパティーにより、ECMP グループ内のトラフィック分散に対する柔軟性と制御が向上します。

Bugzilla:2162401

xdp-tools がバージョン 1.3.1 にリベースされました。

xdp-tools パッケージがアップストリームバージョン 1.3.1 にアップグレードされ、以前のバージョンに比べて多くの機能強化とバグ修正が行われました。

- 次のユーティリティーが追加されました。
  - **xdp-bench**: 受信側で XDP ベンチマークを実行します。
  - **xdp-monitor**: カーネルトレースポイントを使用して、XDP エラーと統計を監視します。
  - **xdp-trafficgen**: XDP ドライバーフックを介してトラフィックを生成および送信します。
- 以下の機能が **libxdp** ライブラリーに追加されました。
  - **xdp_multiprog__xdp_frags_support()**、**xdp_program__set_xdp_frags_support()**、および **xdp_program__xdp_frags_support()** の関数は、XDP **frags** サポート (**multibuffer XDP** とも呼ばれる機能) によるプログラムの読み込みをサポートするために追加されました。
  - ライブラリーは、プログラムを **AF_XDP** ソケットに接続するときに適切な参照カウントを実行します。その結果、アプリケーションはソケットを使用するときに XDP プログラムを手動で切り離す必要がなくなりました。**libxdp** ライブラリーは、プログラムが使用されなくなったときにプログラムを自動的に切り離すようになりました。
  - ライブラリーに以下の関数が追加されました。
    - **xdp_program__create()** (**xdp_program** オブジェクトの作成用)
    - **xdp_program__clone()** (**xdp_program** リファレンスのクローン作成用)
    - **xdp_program__test_run()** (XDP programs through the **BPF_PROG_TEST_RUN** カーネル API の実行用)
  - **LIBXDP_BPFFS_AUTOMOUNT** 環境変数が設定されている場合、**libxdp** ライブラリーは、**bpffs** 仮想ファイルシステムが見つからない場合の自動的なマウントをサポートするようになりました。ライブラリー機能のサブセットは、**bpffs** がマウントされていない場合にも機能できるようになりました。

このバージョンでは、ネットワークデバイスにロードされている XDP ディスパッチャープログラムのバージョン番号も変更されることに注意してください。これは、**libxdp** と **xdp-tools** の以前のバージョンと新しいバージョンを同時に使用できないことを意味します。**libxdp** 1.3 ライブラリーは古いバージョンのディスパッチャーを表示しますが、自動的にアップグレードしません。さらに、**libxdp** 1.3 でプログラムをロードすると、古いバージョンは新しいバージョンと相互運用できなくなります。

Bugzilla:2160066

iproute がバージョン 6.1.0 にリベースされました。

iproute パッケージがバージョン 6.1.0 にアップグレードされ、複数のバグ修正と拡張機能が提供されています。主な変更点は、以下のとおりです。

- **vdpa** デバイス統計の読み取りをサポート
  - インデックス 1 の **virtqueue** データ構造の統計読み取りの図:

```
# vdpa dev vstats show vdpa-a qidx 1
vdpa-a:
vdpa-a: queue_type tx received_desc 321812 completed_desc 321812
```

- インデックス 16 の **virtqueue** データ構造の統計読み取りの図:

```
# vdpa dev vstats show vdpa-a qidx 16
vdpa-a: queue_type control_vq received_desc 17 completed_desc 17
```

- 対応する man ページを更新

Bugzilla:2155604

カーネルは、SYN フラッドメッセージにリスニングアドレスを記録するようになりました。

この機能拡張により、リスニング IP アドレスが SYN フラッドメッセージに追加されます。

```
Possible SYN flooding on port <ip_address>:<port>.
```

その結果、多くのプロセスが異なる IP アドレスの同じポートにバインドされている場合、管理者は影響を受けるソケットを明確に特定できるようになりました。

Bugzilla:2143850

VLAN インターフェイスの新しい nmstate 属性の導入

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

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

YAML 設定ファイルは以下の例のようになります。

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

Jira:RHEL-19142

4.8. カーネル

RHEL 9.2 のカーネルバージョン

Red Hat Enterprise Linux 9.2 は、カーネルバージョン 5.14.0-284.11.1 で配布されます。

Bugzilla:2177782

64k ページサイズのカーネルが利用可能になりました。

4k ページをサポートする RHEL 9 for ARMカーネルに加えて、Red Hat は 64k ページをサポートするオプションのカーネルパッケージ **kernel-64k** を提供するようになりました。

64k ページサイズのカーネルは、ARM プラットフォーム上の大規模なデータセットに便利なオプションです。これにより、メモリーや CPU を大量に消費する一部の操作のパフォーマンスが向上します。

64 ビット ARM アーキテクチャーシステムでは、インストール時にページサイズを選択する必要があります。**kernel-64k** パッケージを **Kickstart** ファイルのパッケージリストに追加すると、Kickstart のみで **kernel-64k** をインストールできます。

kernel-64k のインストールの詳細については、高度な RHEL 9 インストールの実行 を参照してください。

Bugzilla:2153073

kexec-tools の virtiofs サポートが有効になりました。

この機能拡張では、新しいオプション **virtiofs myfs** を導入することにより、**kexec-tools** の **virtiofs** 機能が追加されます。ここで、**myfs** は、**qemu** コマンドラインで設定する変数タグ名です (**-device vhost-user-fs-pci,tag=myfs** など)。

virtiofs ファイルシステムには、ホスト上にエクスポートされたディレクトリーをゲストがマウントできるようにするドライバーが実装されています。この機能拡張を使用すると、仮想マシンの **vmcore** ダンプファイルを次の場所に保存できます。

- **virtiofs** 共有ディレクトリー。
- ルートファイルシステムが **virtiofs** 共有ディレクトリーである場合は、サブディレクトリー (**/var/crash** など)。
- 仮想マシンのルートファイルシステムが **virtiofs** 共有ディレクトリーである場合は、別の **virtiofs** 共有ディレクトリー。

Bugzilla:2085347

kexec-tools パッケージにリモート kdump ターゲットの機能強化が追加されました。

この機能拡張により、**kexec-tools** パッケージに重要なバグ修正と機能拡張が追加されました。以下は、主な変更点です。

- 必要なネットワークインターフェイスのみを有効にすることで、**kdump** のメモリー消費を最適化しました。
- 接続タイムアウト障害が発生した場合の **kdump** のネットワーク効率が向上しました。ネットワークが確立されるまでのデフォルトの待ち時間は最大 10 分です。これにより、キャリアを識別するための回避策として **rd.net.timeout.carrier** や **rd.net.timeout.dhcp** などの **dracut** パラメーターを渡す必要がなくなります。

Bugzilla:2076416

BPF がバージョン 6.0 にリベースされました。

Berkeley Packet Filter (BPF) 機能が複数の拡張機能を備えた Linux カーネルバージョン 6.0 にリベースされました。この更新により、カーネルモジュールの BPF Type Format (BTF) に依存するすべての BPF 機能が有効になります。このような機能には、トレース用の BPF トランポリンの使用、Compile Once - Run Everywhere (CO-RE) メカニズムの可用性、および複数のネットワーク関連機能が含まれます。さらに、カーネルモジュールにはデバッグ情報が含まれるようになりました。つまり、実行中のモジュールを検査するために **debuginfo** パッケージをインストールする必要がなくなりました。

実行中のカーネルで使用できる BPF 機能の完全なリストの詳細については、**bpftool feature** コマンドを使用してください。

Jira:RHELPLAN-133650

rtla メタツールは、トレース機能を向上させるために osnoise および timerlat トレーサーを追加します。

Real-Time Linux Analysis (**rtla**) は、Linux のリアルタイムプロパティーを分析する一連のコマンドを含むメタツールです。**rtla** は、カーネルトレース機能を利用して、予期しないシステム結果のプロパティーと根本原因に関する正確な情報を提供します。**rtla** は現在、**osnoise** および **timerlat** トレーサーコマンドのサポートを追加しています。

- **osnoise** トレーサーは、オペレーティングシステムのノイズに関する情報を報告します。
- **timerlat** トレーサーは、タイマー IRQ ハンドラーおよびスレッドハンドラーでのタイマーレイテンシーを定期的に出力します。

rtla の **timerlat** 機能を使用するには、**sysctl -w kernel.sched_rt_runtime_us=-1** スクリプトを使用して、アドミッションコントロールを無効にする必要があることに注意してください。

Bugzilla:2075216

Tuna のargparse モジュールが CPU ソケットの設定をサポートするようになりました。

この機能強化により、複数の CPU ソケットがある場合に特定の CPU ソケットを指定できるようになります。サブコマンドで **-h** を使用すると、ヘルプの使用法を表示できます (**tuna show_threads -h** など)。

特定の CPU ソケットを設定するには、CPU ソケットを使用する必要がある各 **tuna** コマンドで **-S** オプションを指定します。

```
tuna <command> [-S CPU_SOCKET_LIST]
```

たとえば、**tuna show_threads -S 2,3** を使用してスレッドを表示するか、**tuna show_irqs -S 2,3** を使用してアタッチされた割り込み要求 (IRQ) を表示します。

結果として、この機能強化により、各 CPU を個別に指定する必要がなく、CPU ソケットに基づいた CPU 使用が容易になります。

Bugzilla:2122781

Tuna の cgroups と irqs の出力形式が改善され、読みやすくなりました。

この機能強化により、**cgroup** ユーティリティーの **tuna show_threads** コマンド出力が端末のサイズに基づいて構造化されるようになりました。新しい **-z** または **--spaced** オプションを **show_threads** コマンドに追加することで、**cgroups** 出力に追加のスペースを設定することもできます。

その結果、**cgroups** 出力は、端末のサイズに合わせて読みやすい形式に改良されました。

Bugzilla:2121517

新しいコマンドラインインターフェイスがリアルタイムで tuna ツールに追加されました。

この機能拡張により、**argparse** 解析モジュールに基づく新しいコマンドラインインターフェイスが **tuna** ツールに追加されます。この更新により、次のタスクを実行できるようになりました。

- アプリケーションおよびカーネルスレッドの属性を変更します。
- 名前または番号によって割り込み要求 (IRQ) を操作します。
- プロセス識別子を使用してタスクまたはスレッドを操作します。
- CPU またはソケット番号を使用して、CPU と CPU セットを指定します。

tuna -h コマンドを使用すると、コマンドライン引数とそれに対応するオプションを出力できます。各コマンドにはオプションの引数があり、**tuna <command> -h** コマンドで表示できます。

その結果、**tuna** は、コマンドラインインターフェイスよりも使いやすく保守しやすい、より標準化されたコマンドとオプションのメニューを備えたインターフェイスを提供するようになりました。

Bugzilla:2062865

rteval コマンドの出力には、プログラムのロードと測定スレッドの情報が含まれるようになりました。

rteval コマンドは、プログラムのロード数、測定スレッド、およびこれらのスレッドを実行した対応する CPU を含むレポートの概要を表示するようになりました。この情報は、特定のハードウェアプラットフォームの負荷下でのリアルタイムカーネルのパフォーマンスを評価するのに役立ちます。

rteval レポートは、システムのブートログとともに XML ファイルに書き込まれ、**rteval-<date>-N-tar.bz2** 圧縮ファイルに保存されます。**date** はレポート生成日を指定し、**N** は N 回目の実行のカウンターです。

rteval レポートを生成するには、次のコマンドを入力します。

```
# rteval --summarize rteval-<date>-N.tar.bz2
```

Bugzilla:2081325

レイテンシーを測定するために、-W および --bucket-width オプションが oslat プログラムに追加されました。

この機能強化により、単一バケットのレイテンシー範囲をナノ秒の精度で指定できるようになりました。1000 ナノ秒の倍数ではない幅は、ナノ秒の精度を示します。新しいオプション **-W** または **--bucket-width** を使用すると、バケット間のレイテンシー間隔を変更して、マイクロ秒未満の遅延時間内のレイテンシーを測定できます。

たとえば、1 - 4 の CPU 範囲で実行するために 10 秒間にわたって 32 個のバケットのレイテンシーバケット幅を 100 ナノ秒に設定し、ゼロのバケットサイズを省略するには、次のコマンドを実行します。

```
# oslat -b 32 -D 10s -W 100 -z -c 1-4
```

このオプションを使用する前に、誤差測定に関してどのレベルの精度が重要であるかを判断する必要があることに注意してください。

Bugzilla:2041637

kdump ストレージターゲットとして有効になった NVMe/FC トランスポートプロトコル。

kdump メカニズムは、ダンプターゲットとしてファイバーチャネル (NVMe/FC) プロトコル上の Nonvolatile Memory Express (NVMe) のサポートを提供するようになりました。この更新により、カーネルクラッシュダンプファイルを NVMe/FC ストレージターゲットに保存するように **kdump** を設定できるようになります。

その結果、**kdump** は、カーネルクラッシュが発生した場合でも、**timeout** や **reconnect** エラーを発生させることなく、**vmcore** ファイルをキャプチャーして **NVMe/FC** に保存できます。

NVMe/FC 設定の詳細については、ストレージデバイスの管理 を参照してください。

Bugzilla:2080110

crash-utility ツールがバージョン 8.0.2 にリベースされました。

アクティブなシステム状態またはカーネルクラッシュ後の分析を行う **crash-utility** がバージョン 8.0.2 にリベースされました。注目すべき変更には、**multiqueue(blk-mq)** デバイスのサポートの追加が含まれます。**dev -d** または **dev -D** コマンドを使用すると、**multiqueue(blk-mq)** デバイスのディスク I/O 統計を表示できます。

Bugzilla:2119685

openssl-ibmca がバージョン 2.3.1 にリベースされました。

64 ビット IBM Z アーキテクチャー上の IBMCA の動的 OpenSSL エンジンとプロバイダーがアップストリームバージョン 2.3.1 にリベースされました。RHEL 9 のユーザーは、OpenSSL の今後の更新との互換性を確保するために、OpenSSL **プロバイダー** を使用することを推奨します。**エンジン** 機能は OpenSSL バージョン 3 で非推奨になりました。

Bugzilla:2110378

顧客キーを使用した Secure Execution ゲストダンプ暗号化

この新機能により、**kdump** ユーティリティーが機能しないシナリオで、Secure Execution ゲストのハイパーバイザー開始ダンプが KVM からカーネルクラッシュ情報を収集できるようになります。Secure Execution のハイパーバイザー開始ダンプは、IBM Z シリーズ z16 および LinuxONE Empire 4 ハードウェア向けに設計されていることに注意してください。

Bugzilla:2044204

リアルタイムの TSN プロトコルが ADL-S プラットフォームで有効になりました。

この機能強化により、IEEE Time Sensitive Networking (TSN) 仕様により、Intel Alder Lake S (ADL-S) プラットフォーム上のネットワーク上でのリアルタイムワークロードの時刻同期と確定的処理が可能になります。次のネットワークデバイスをサポートします。

- TSN サポートを備えたディスクリート 2.5GbE MAC-PHY コンボ: Intel® i225/i226
- 1GbE および 2.5Gbe 速度をカバーする Marvell、Maxlinear、TI のサードパーティー PHY チップを備えた SOC 内の統合 2.5GbE MAC は、一部の **SKU** および SOC で利用できます。

TSN プロトコルを使用すると、組み込み実装での決定論的なアプリケーションのスケジューリング、プリエンプション、および正確な時刻同期タイプのワークロードを管理できます。これらの実装には専用の特殊な独自のネットワークが必要ですが、ワークロードは標準のイーサネット、Wi-Fi、および 5G ネットワークで実行されます。

その結果、TSN は次の機能を向上させました。

- ハードウェア: IoT でのリアルタイムワークロードの実装に使用される Intel ベースのシステム

- 決定的で時間に敏感なアプリケーション

Bugzilla:2100606

Intel Ice ドライバーがバージョン 6.0.0 にリベースされました。

Intel **ice** ドライバーはアップストリームバージョン 6.0.0 にアップグレードされ、以前のバージョンに比べて多くの機能強化とバグ修正が行われました。注目すべき機能強化には次のものがあります。

- Point-to-Point Protocol over Ethernet (**PPPoE**) プロトコルのハードウェアオフロード
- Inter-Integrated Circuit (**I2C**) プロトコル書き込みコマンド
- イーサネットスイッチデバイスドライバーモデル (**switchdev**) の VLAN タグプロトコル識別子 (**TPID**) フィルター
- **switchdev** での二重 VLAN タグ付け

Bugzilla:2104468

GNSS モジュールのデータを書き込むオプションが利用可能になりました。

この更新では、**gnss** レシーバーにデータを書き込むオプションが提供されます。以前は、**gnss** は完全には設定できませんでした。この機能強化により、すべての **gnss** 機能が利用できるようになりました。

Bugzilla:2111048

IBM zSystems のセキュアブート証明書のホスティング

IBM z16 A02/AGZ および LinuxONE Rockhopper 4 LA2/AGL 以降、ハードウェア管理コンソール (HMC) でセキュアブートを有効にしてシステムを起動するときに、Linux カーネルの検証に使用される証明書を管理できるようになりました。以下に例を示します。

- DPM およびクラシックモードで HMC を使用し、HMC からアクセスできる FTP サーバーからシステム証明書ストアに証明書をロードできます。HMC に接続された USB デバイスから証明書をロードすることもできます。
- 証明書ストアに保存されている証明書を LPAR パーティションに関連付けることができます。複数の証明書を 1 つのパーティションに関連付けたり、1 つの証明書を複数のパーティションに関連付けたりできます。
- HMC インターフェイスを使用して、証明書ストア内の証明書の関連付けをパーティションから解除できます。
- 証明書ストアから証明書を削除できます。
- 最大 20 個の証明書を 1 つのパーティションに関連付けることができます。

ビルトインのファームウェア証明書は引き続き使用できます。ユーザー管理の証明書ストアを使用するとすぐに、ビルトインの証明書は使用できなくなります。

証明書ストアにロードする証明書ファイルは、次の要件を満たしている必要があります。

- **PEM** または **DER-encoded X.509v3** 形式で、ファイル名拡張子が **.pem**、**.cer**、**.crt**、または **.der** のいずれかである。
- 有効期限が切れていない。

- キー使用属性が **デジタル署名** である。
- 拡張キー使用属性に **コード署名** が含まれている。

ファームウェアインターフェイスを使用すると、論理パーティションで実行されている Linux カーネルが、このパーティションに関連付けられた証明書をロードできるようになります。Linux on IBM Z は、これらの証明書を **.platform** キーリングに保存して、Linux カーネルが **kexec** カーネルを検証し、そのパーティションに関連付けられた証明書を使用してサードパーティーのカーネルモジュールを検証できるようにします。

検証済みの証明書のみをアップロードし、失効した証明書を削除するのは、オペレーターの責任です。

注記

HMC に読み込む必要がある **Red Hat Secureboot 302** 証明書は、Product Signing Keys から入手できます。

Bugzilla:2190123

zipl が 64 ビット IBM Z でのセキュアブート IPL とダンプをサポート

この更新により、**zipl** ユーティリティーは、64 ビット IBM Z アーキテクチャー上の Extended Count Key Data (ECKD) Direct Access Storage Devices (DASD) からのList-Directed IPL および List-Directed ダンプをサポートします。その結果、IBM Z での RHEL のセキュアブートは、ECKD タイプの DASD でも動作します。

Bugzilla:2044200

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

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

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

Jira:RHEL-18359

4.9. ファイルシステムおよびストレージ

nvme-cli がバージョン 2.2.1 にリベースされました。

nvme-cli パッケージがバージョン 2.2.1 にアップグレードされ、複数のバグ修正と拡張機能が提供されています。主な変更点は、以下のとおりです。

- すべての NVMe サブシステムのトポロジーを表示する、新しい **nvme show-topology** コマンドが追加されました。
- **libuuid** 依存関係を削除しました。

- **uint128** データフィールドは正しく表示されます。
- **libnvme** 依存関係をバージョン 1.2 に更新しました。

Bugzilla:2139753

libnvme がバージョン 1.2 にリベースされました。

libnvme パッケージがバージョン 1.2 にアップグレードされ、複数のバグ修正と拡張機能が提供されています。最も注目すべき変更は、**libuuid** ライブラリーの依存関係が削除されたことです。

Bugzilla:2139752

Stratis はプール内で一貫したブロックサイズを強制します。

Stratis は、プール内でブロックサイズが混在しているデバイスが存在する場合に発生する可能性がある潜在的なエッジケースの問題に対処するために、プール内で一貫したブロックサイズを強制するようになりました。この機能強化により、ユーザーはプールを作成したり、プール内の既存のデバイスとは異なるブロックサイズを持つ新しいデバイスを追加したりできなくなります。その結果、プール障害のリスクが軽減されます。

Bugzilla:2039957

Stratis プール内の既存のディスク増加のサポート。

以前は、ユーザーが RAID アレイに新しいディスクを追加すると、通常、RAID アレイのサイズが増加していました。ただし、どの場合でも、Stratis はサイズの増加を無視し、最初にプールに追加されたときに RAID アレイ上で利用可能なスペースのみを使用し続けました。その結果、Stratis は新しいデバイスを識別できず、ユーザーはプールのサイズを増やすことができませんでした。

この機能強化により、Stratis はサイズが拡大したプールデバイスメンバーを識別できるようになりました。その結果、ユーザーは要件に基づいてプールを拡張するコマンドを発行できるようになりました。

Stratis は、新しいディスクを追加してプールを拡張する既存の機能に加えて、プール内の既存のディスクの拡張をサポートするようになりました。

Bugzilla:2039955

lvreduce コマンドの機能の改善。

この機能強化により、論理ボリューム (LV) がアクティブな場合、**lvreduce** コマンドは、LV サイズを縮小することでその上に存在するファイルシステムが損傷するかどうかをチェックします。LV 上のファイルシステムで縮小が必要であり、**lvreduce resizefs** オプションが有効になっていない場合、LV は縮小されません。

さらに、LV を削減しながらファイルシステムの処理を制御するための新しいオプションが利用できるようになりました。これらのオプションにより、ユーザーは **lvreduce** コマンドを使用する際の柔軟性と制御が向上します。

Bugzilla:1878893

statx のダイレクト I/O アライメント情報が追加されました。

この更新により、新しいマスク値 **STATX_DIOALIGN** が **statx(2)** 呼び出しに導入されました。この値が **stx_mask** フィールドに設定されている場合、**stx_dio_mem_align** および **stx_dio_offset_align** 値を要求します。これらの値は、それぞれ、ユーザーメモリーバッファーとファイルオフセットに必要なアライメント (バイト単位)、およびこのファイルのダイレクト I/O (O_DIRECT) の I/O セグメント長を

示します。ファイルで直接 I/O がサポートされていない場合、両方の値は 0 になります。このインターフェイスは、RHEL9 の xfs および ext4 ファイルシステム上のファイルだけでなく、ブロックデバイスにも実装されるようになりました。

Bugzilla:2150284

NFSv4.1 セッショントランキングの検出

この更新により、クライアントは同じサーバーおよびセッションに対して複数の接続を使用できるようになり、データ転送が高速化されます。NFS クライアントが異なる IP アドレスを持つマルチホーム NFS サーバーをマウントする場合、デフォルトでは 1 つの接続のみが使用され、残りは無視されます。パフォーマンスを向上させるために、この更新では、**trunkdiscovery** および **max_connect** マウントオプションのサポートが追加されています。これにより、クライアントは各接続をテストし、複数の接続を同じ NFSv4.1+ サーバーおよびセッションに関連付けることができます。

Bugzilla:2066372

NFS IO サイズを TCP および RDMA の PAGE_SIZE の倍数として設定できるようになりました。

この更新により、ユーザーは TCP および RDMA 接続の NFS IO サイズを **PAGE_SIZE** の倍数として設定できるようになります。これにより、一部のアーキテクチャーの NFS パフォーマンスを最適化する際の柔軟性が向上します。

Bugzilla:2107347

nfsrahead が RHEL 9 に追加されました。

nfsrahead ツールの導入により、それを使用して NFS マウントの **readahead** 値を変更できるため、NFS 読み取りパフォーマンスに影響を与えることができます。

Bugzilla:2143747

4.10. 高可用性およびクラスター

新しい enable-authfile ブース設定オプション

クラスター設定でブースチケットマネージャーを使用するブース設定を作成する場合、**pcs boost setup** コマンドにより、新しい **enable-authfile** ブース設定オプションがデフォルトで有効になるようになりました。**pcsboothenable-authfile** コマンドを使用して、既存のクラスターでこのオプションを有効にできます。さらに、**pcs status** および **pcs boost status** コマンドは、**enable-authfile** の設定ミスの可能性を検出したときに警告を表示するようになりました。

Bugzilla:2116295

pcs はリソースおよび stonith エージェントの validate-all アクションを実行できるようになりました。

リソースまたは STONITH デバイスを作成または更新するときに、**--agent-validation** オプションを指定できるようになりました。このオプションを使用すると、**pcs** は、エージェントのメタデータに基づいて **pcs** によって実行される検証に加えて、エージェントの **validate-all** アクション (利用可能な場合) を使用します。

Bugzilla:2112270、Bugzilla:2159454

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

Python 3.11 は RHEL 9 で利用可能

RHEL 9.2 では、新しいパッケージ **python3.11** とそのために構築された一連のパッケージ、および **ubi9/python-311** コンテナーイメージによって提供される Python 3.11 が導入されています。

以前にリリースされた Python 3.9 と比較して注目すべき機能強化は次のとおりです。

- パフォーマンスが大幅に向上しました。
- 新しい **match** キーワードを使用した構造パターンマッチング (他の言語の **switch** と同様)。
- たとえば、閉じられていない丸かっこや角かっこを示すエラーメッセージが改善されました。
- デバッグやその他の使用例のための正確な行番号。
- 定義を丸かっこで囲むことにより、複数行にわたるコンテキストマネージャーの定義をサポートします。
- 新しい **X | Y** 型ユニオン演算子、variadic generics、新しい **Self** 型など、タイプヒントと **typing** モジュールに関連するさまざまな新機能。
- エラーの原因となった式を示すトレースバック内の正確なエラー位置。
- TOML の解析をサポートする新しい **tomllib** 標準ライブラリーモジュール。
- 例外グループと新しい **except*** 構文を使用して、無関係な複数の例外を同時に発生させて処理する機能。

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

python3.11 スタックからパッケージをインストールするには、たとえば、次を使用します。

```
# dnf install python3.11
# dnf install python3.11-pip
```

インタープリターを実行するには、たとえば、以下を使用します。

```
$ python3.11
$ python3.11 -m pip --help
```

詳細については、Python のインストールと使用 を参照してください。

Python 3.11 のライフサイクルは、RHEL 9 のデフォルトの Python 実装である Python 3.9 よりも短いことに注意してください。Red Hat Enterprise Linux アプリケーションストリームのライフサイクル を参照してください。

Bugzilla:2127923

nodejs:18 がバージョン 18.14 にリベースされ、npm がバージョン 9 にリベースされました。

更新された **Node.js 18.14** には、**npm** のバージョン 8 からバージョン 9 への SemVer メジャーアップグレードが含まれています。この更新はメンテナンス上の理由から必要であり、**npm** 設定の調整が必要になる場合があります。

特に、特定のレジストリーに範囲を限定しない認証関連の設定はサポートされなくなりました。この変更はセキュリティー上の理由から行われました。スコープ指定されていない認証設定を使用した場合、指定されたトークンは **.npmrc** ファイルにリストされているすべてのレジストリーに送信されます。

スコープなしの認証トークンを使用する場合は、レジストリースコープ付きトークンを生成して **.npmrc** ファイルに指定します。

.npmrc ファイル内に **//registry.npmjs.org/:_auth** など、**_auth** を使用する設定行がある場合は、それらを **//registry.npmjs.org/:_authToken=${NPM_TOKEN}** に置き換え、生成したスコープ付きトークンを指定します。

変更の完全なリストについては、アップストリームの変更ログ を参照してください。

Bugzilla:2178088

git がバージョン 2.39.1 にリベースされました。

Git バージョン管理システムがバージョン 2.39.1 に更新され、以前にリリースされたバージョン 2.31 に比べて、バグ修正、拡張機能、およびパフォーマンスが向上しました。

主な機能拡張は、次のとおりです。

- **git log** コマンドは、**git describe** 出力のフォーマットプレースホルダーをサポートするようになりました。**git log --format=%(describe)**
- **git commit** コマンドで、ログメッセージを変更せずにコミットの内容を修正できる **--fixup<commit>** オプションがサポートされるようになりました。この更新により、以下も使用できるようになります。
  - **--fixup=amend:<commit>** オプションは、メッセージとコンテンツの両方を変更します。
  - **--fixup=reword:<commit>** オプションは、コミットメッセージのみを更新します。
- **git clone** コマンドで新しい **--reject-shallow** オプションを使用すると、浅いリポジトリーからのクローン作成を無効にすることができます。
- **git branch** コマンドで **--recurse-submodules** オプションがサポートされるようになりました。
- **git merge-tree** コマンドを使用して、次のことができるようになりました。
  - 2 つのブランチをマージできるかどうかをテストします。
  - ブランチがマージされた場合にマージコミットになるツリーを計算します。
- 新しい **safe.bareRepository** 設定変数を使用して、ベアリポジトリーをフィルタリングして除外できます。

Bugzilla:2139379

git-lfs がバージョン 3.2.0 にリベースされました。

Git Large File Storage (LFS) 拡張機能がバージョン 3.2.0 に更新され、以前にリリースされたバージョン 2.13 に比べて、バグ修正、拡張機能、およびパフォーマンスが向上しました。

主な変更点は、以下のとおりです。

- **Git LFS** は純粋な SSH ベースのトランスポートプロトコルを導入します。
- **Git LFS** はマージドライバーを提供するようになりました。
- **git lfs fsck** ユーティリティーは、ポインターが正規であること、および予期される LFS ファイルの形式が正しいことをさらにチェックするようになりました。
- NT LAN Manager (NTLM) 認証プロトコルのサポートは削除されました。代わりに Kerberos または Basic 認証を使用してください。

Bugzilla:2139383

新しいモジュールストリーム: nginx:1.22

nginx 1.22 Web およびプロキシーサーバーは、**nginx:1.22** モジュールストリームとして利用できるようになりました。この更新では、以前にリリースされたバージョン 1.20 に対して、多数のバグ修正、セキュリティー修正、新機能、機能強化が提供されます。

新機能:

- **nginx** は以下をサポートするようになりました。
  - OpenSSL 3.0、および OpenSSL 3.0 を使用する場合の **SSL_sendfile()** 関数。
  - PCRE2 ライブラリー。
  - **mail** プロキシーモジュールでの POP3 および IMAP パイプライン。
- **nginx** は、**Auth-SSL-Protocol** および **Auth-SSL-Cipher** ヘッダー行をメールプロキシー認証サーバーに渡すようになりました。

拡張されたディレクティブ:

- **ssl_conf_command**、**ssl_reject_handshake** など、新しいディレクティブが複数利用できるようになりました。
- **proxy_cookie_flags** ディレクティブが変数に対応するようになりました。
- **nginx** は、**proxy_ssl_certificate**、**proxy_ssl_certificate_key**、**grpc_ssl_certificate**、**grpc_ssl_certificate_key**、**uwsgi_ssl_certificate**、および **uwsgi_ssl_certificate_key** ディレクティブの変数をサポートするようになりました。
- ストリームモジュールの **listen** ディレクティブは、新しい **fastopen** パラメーターをサポートするようになりました。これにより、リスニングソケットの **TCP Fast Open** モードが有効になります。
- 新しい **max_errors** ディレクティブが **mail** プロキシーモジュールに追加されました。

その他の変更点:

- **nginx** は、次の場合、常にエラーを返すようになりました。
  - **CONNECT** メソッドが使用されます。
  - **Content-Length** と **Transfer-Encoding** の両方のヘッダーがリクエストに指定されます。

- リクエストヘッダー名にスペースまたは制御文字が含まれています。
- **Host** リクエストヘッダー行には、スペースまたは制御文字が含まれています。

- **nginx** は、**Transfer-Encoding** ヘッダーを含むすべての HTTP/1.0 リクエストをブロックするようになりました。
- **nginx** は、Application Layer Protocol Negotiation (ALPN) を使用して HTTP/2 接続を確立するようになり、Next Protocol Negotiation (NPN) プロトコルはサポートされなくなりました。

nginx:1.22 ストリームをインストールするには、次を使用します。

```
# dnf module install nginx:1.22
```

詳細については、NGINX のセットアップと設定 を参照してください。

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

Bugzilla:2096174

mod_security がバージョン 2.9.6 にリベースされました。

Apache HTTP サーバーの **mod_security** モジュールがバージョン 2.9.6 に更新され、以前に利用可能だったバージョン 2.9.3 に新機能、バグ修正、セキュリティー修正が追加されました。

主な機能拡張は、次のとおりです。

- **modsecurity.conf-recommended** ファイル内のパーサーのアクティブ化ルールを調整しました。
- **mod_security** が HTTP マルチパートリクエストを解析する方法が強化されました。
- 新しい **MULTIPART_PART_HEADERS** コレクションが追加されました。
- フォーマットされたログのタイムスタンプに microsec のタイムスタンプ解像度を追加しました。
- 欠落している地域の国を追加しました。

Bugzilla:2143211

新しいパッケージ: tomcat

RHEL 9.2 では、Apache Tomcat サーバーバージョン 9 が導入されています。Tomcat は、Java Servlet および JavaServer Pages テクノロジーの公式リファレンス実装で使用されるサーブレットコンテナーです。Java Servlet および JavaServer Pages の仕様は、Java Community Process に基づいて Sun によって開発されました。Tomcat はオープンな参加型環境で開発され、Apache ソフトウェアライセンスバージョン 2.0 に基づいてリリースされています。

Bugzilla:2160511

新しいモジュールストリーム: postgresql:15

RHEL 9.2 では、**PostgreSQL 15** が **postgresql:15** モジュールストリームとして導入されています。**PostgreSQL 15** は、バージョン 13 に比べて多くの新機能と拡張機能を提供します。主な変更点は、以下のとおりです。

- サブスクリプトを使用して **PostgreSQL** JSON データにアクセスできるようになりました。クエリーの例:

```
SELECT ('{ "postgres": { "release": 15 }}'::jsonb)['postgres']['release'];
```

- **PostgreSQL** は、複数範囲のデータ型をサポートし、**range_agg** 関数を拡張して複数範囲のデータ型を集約するようになりました。
- **PostgreSQL** は監視と可観測性を向上させます。
  - **COPY** コマンドとログ先行書き込み (WAL) アクティビティーの進行状況を追跡できるようになりました。
  - **PostgreSQL** はレプリケーションスロットに関する統計を提供するようになりました。
  - **compute_query_id** パラメーターを有効にすることで、**pg_stat_activity** や **EXPLAIN VERBOSE** など、複数の **PostgreSQL** 機能を通じてクエリーを独自に追跡できるようになりました。
- **PostgreSQL** では、次のようにクエリー並列処理のサポートが向上しています。
  - 並列順次スキャンのパフォーマンスが向上しました。
  - **RETURN QUERY** コマンドの使用時に並列クエリーを実行する SQL 手続き型言語 (**PL/pgSQL**) の機能。
  - **REFRESH MATERIALIZED VIEW** コマンドで並列処理を有効にしました。
- **PostgreSQL** には SQL 標準の **MERGE** コマンドが含まれるようになりました。**MERGE** を使用すると、**INSERT**、**UPDATE**、および **DELETE** アクションを 1 つのステートメントに含めることができる条件付き SQL ステートメントを作成できます。
- **PostgreSQL** では、正規表現を使用して文字列を検査するための新しい関数 **regexp_count()**、**regexp_instr()**、**regexp_like()**、および **regexp_substr()** を提供します。
- **PostgreSQL** には、**security_invoker** パラメーターが追加されており、これを使用すると、ビュー作成者ではなくビュー呼び出し元の権限でデータをクエリーすることができます。これは、ビューの呼び出し元が基になるデータを操作するための適切な権限を持っていることを確認するのに役立ちます。
- **PostgreSQL** は、アーカイブ機能とバックアップ機能のパフォーマンスを向上させます。
- **PostgreSQL** では、**LZ4** および **Zstandard** (**zstd**) 可逆圧縮アルゴリズムのサポートが追加されています。
- **PostgreSQL** は、メモリー内およびディスク上のソートアルゴリズムを改善します。
- 更新された **postgresql.service** systemd ユニットファイルにより、ネットワークが起動した後に **postgresql** サービスが確実に開始されるようになりました。

次の変更には下位互換性がありません。

- パブリックスキーマのデフォルトの権限が変更されました。新規に作成されたユーザーは、**GRANT ALL ON SCHEMA public TO myuser;** コマンドを使用して、権限を明示的に付与する必要があります。以下に例を示します。

```
postgres=# CREATE USER mydbuser;
```

```
postgres=# GRANT ALL ON SCHEMA public TO mydbuser;
postgres=# \c postgres mydbuser
postgres=$ CREATE TABLE mytable (id int);
```

- **libpq PQsendQuery()** 関数はパイプラインモードではサポートされなくなりました。影響を受けるアプリケーションを変更して、代わりに **PQsendQueryParams()** 関数を使用します。

PostgreSQL の使用 も参照してください。

postgresql:15 ストリームをインストールするには、次を使用します。

```
# dnf module install postgresql:15
```

RHEL 9 内の以前の **postgresql** ストリームからアップグレードする場合は、PostgreSQL の RHEL9 バージョンへの移行 の説明に従って **PostgreSQL** データを移行します。

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

Bugzilla:2128410

4.12. コンパイラーおよび開発ツール

openblas がバージョン 0.3.21 にリベースされました。

OpenBLAS ライブラリーがバージョン 0.3.21 に更新されました。この更新には、IBM POWER10 プラットフォームのパフォーマンス最適化パッチが含まれています。

Bugzilla:2112099

新しいモジュールストリーム: swig:4.1

RHEL 9.2 では、Simplified Wrapper and Interface Generator (SWIG) バージョン 4.1 が **swig:4.1** モジュールストリームとして導入され、CodeReady Linux Builder (CRB) リポジトリーで利用できます。CodeReady Linux Builder リポジトリーに含まれるパッケージは、サポート対象外であることに注意してください。

RHEL 9.0 でリリースされた **SWIG 4.0** と比較すると、**SWIG 4.1** は次のとおりです。

- **Node.js** バージョン 12 ～ 18 のサポートを追加し、**Node.js** バージョン 6 より前のサポートを削除します。
- **PHP 8** のサポートを追加します。
- **PHP** C API を通じて完全に **PHP** ラッピングを処理し、デフォルトでは **.php** ラッパーを生成しなくなりました。
- **Perl 5.8.0** 以降のバージョンのみをサポートします。
- **Python** バージョン 3.9 から 3.11 のサポートを追加します。
- **Python 3.3** 以降の **Python 3** バージョンと **Python 2.7** のみをサポートします。
- **Python** で生成されたコードにおけるさまざまなメモリーリークの修正を提供します。

- C99、C++11、C++14、および C++17 標準のサポートが向上し、C++20 標準の実装が開始されます。
- C++ **std::unique_ptr** ポインタークラスのサポートを追加します。
- C++ テンプレートの処理に複数の小さな改善が含まれています。
- さまざまなケースでの C++ 宣言の使用法を修正しました。

swig:4.1 モジュールストリームをインストールするには:

1. CodeReady Linux Builder (CRB) リポジトリー を有効にします。
2. モジュールストリームをインストールします。

```
# dnf module install swig:4.1
```

Bugzilla:2139101

新しいパッケージ: CRB リポジトリー内の jmc

RHEL 9.2 では、HotSpot JVM バージョン 8.2.0 用の JDK Mission Control (JMC) プロファイラーが導入されており、AMD および Intel 64 ビットアーキテクチャー用の CodeReady Linux Builder (CRB) リポジトリーの **jmc** パッケージとして利用できます。

JMC をインストールするには、まず CodeReady Linux Builder (CRB) リポジトリー を有効にする必要があります。

CRB リポジトリーに含まれるパッケージはサポートされていないことに注意してください。

Bugzilla:2122401

OpenJDK サービス属性が FIPS モードで利用可能。

以前は、FIPS モードの OpenJDK で利用可能な暗号化サービスとアルゴリズムが厳格にフィルターされ、サービス属性が利用できなくなっていました。この機能強化により、これらのサービス属性が FIPS モードで使用できるようになりました。

Bugzilla:2186803

Performance Co-Pilot がバージョン 6.0 にリベースされました。

Performance Co-Pilot (**PCP**) がバージョン 6.0 に更新されました。以下は、主な改善点です。

1. バージョン 3 PCP アーカイブのサポート:
   これには、ドメイン変更デルタ、2038 年対応タイムスタンプ、ナノ秒精度のタイムスタンプ、任意のタイムゾーンのサポート、およびより大きな (2GB を超える) 個々のボリューム全体で使用される 64 ビットファイルオフセットのサポートが含まれます。

   この機能は現在、**/etc/pcp.conf** ファイルの **PCP_ARCHIVE_VERSION** 設定によってオプトインされています。

   バージョン 2 アーカイブはデフォルトのままです。
2. PCP 全体では OpenSSL のみが使用されます。Mozilla NSS/NSPR の使用は廃止されました。これは、**libpcp**、**PMAPI** クライアント、および **PMCD** の暗号化の使用に影響します。これらの要素は、すでに OpenSSL を使用していた **pmproxy** HTTPS サポートおよび **redis-server** と一貫して設定および使用されるようになりました。

3. 新しいナノ秒精度のタイムスタンプ **PMAPI** は、タイムスタンプを利用する **PCP** ライブラリーインターフェイスを呼び出します。
これらはすべてオプションであり、既存のツールに対して完全な下位互換性が維持されます。

4. 次のツールとサービスが更新されました。

pcp2elasticsearch

認証サポートを実装しました。

pcp-dstat

top-alike プラグインのサポートを実装しました。

pcp-htop

最新の安定したアップストリームリリースに更新されました。

pmseries

sum、**avg**、**stdev**、**nth_percentile**、**max_inst**、**max_sample**、**min_inst**、**min_sample** 関数が追加されました。

pmdabpf

CO-RE (Compile Once - Run Everywhere) モジュールと、AMD64、Intel 64 ビット、64 ビット ARM、および IBM Power Systems のサポートが追加されました。

pmdabpftrace

自動起動スクリプトの例を **/usr/share** ディレクトリーに移動しました。

pmdadenki

複数のアクティブなバッテリーのサポートが追加されました。

pmdalinux

最新の **/proc/net/netstat** 変更の更新。

pmdaopenvswitch

インターフェイスとカバレッジ統計を追加しました。

pmproxy

リクエストパラメーターをリクエスト本文で送信できるようになりました。

pmieconf

Open vSwitch メトリック用の複数の **pmie** ルールを追加しました。

pmlogger_farm

ファームロガーのデフォルト設定ファイルを追加しました。

pmlogger_daily_report

いくつかの大幅な効率改善。

Bugzilla:2117074

grafana がバージョン 9.0.9 にリベースされました。

grafana パッケージがバージョン 9.0.9 にリベースされました。主な変更点は、以下のとおりです。

- 時系列パネルがデフォルトの視覚化オプションになり、グラフパネルに置き換わりました。
- 新しいヒートマップパネル
- 新しい Prometheus および Loki クエリービルダー
- Grafana アラートの更新

- 複数の UI/UX とパフォーマンスの改善
- ライセンスが Apache 2.0 から GNU Affero General Public License (AGPL) に変更されました。

以下はオプトインの実験的機能として提供されています。

- 新しい棒グラフパネル
- 新しい状態タイムラインパネル
- 新しいステータス履歴パネル
- 新しいヒストグラムパネル

詳細については、Grafana v9.0 の新機能 および Grafana v8.0 の新機能 を参照してください。

Bugzilla:2116847

grafana-pcp がバージョン 5.1.1 にリベースされました。

grafana-pcp パッケージがバージョン 5.1.1 にリベースされました。主な変更点は、以下のとおりです。

クエリーエディター
: レートの変換と時間使用率の変換を無効にするボタンを追加しました。

Redis
: 非推奨の **label_values(metric, label)** 関数を削除しました。

Redis
: 多くの系列を持つメトリックのネットワークエラーを修正しました (Performance Co-Pilot v6+ が必要)。

Redis
: **pmproxy** API タイムアウトを 1 分に設定します。

Bugzilla:2116848

GCC Toolset 12 の更新

GCC Toolset 12 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 Streams として利用できます。

RHEL 9.2 で導入された注目すべき変更点は次のとおりです。

- GCC コンパイラーがバージョン 12.2.1 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。
- **annobin** がバージョン 11.08 に更新されました。

以下のツールおよびバージョンは、GCC Toolset 12 で利用できます。

ツール	バージョン
GCC	12.2.1

ツール	バージョン
GDB	11.2
binutils	2.38
dwz	0.14
annobin	11.08

GCC Toolset 12 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。

```
# dnf install gcc-toolset-12
```

GCC Toolset 12 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-12 tool
```

GCC Toolset バージョン 12 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-12 bash
```

詳細については、GCC Toolset 12 を参照してください。

Bugzilla:2110583

更新された GCC コンパイラーが RHEL 9.2 で利用できるようになりました。

システム GCC コンパイラーバージョン 11.3.1 が更新され、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。

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

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

Bugzilla:2117632

LLVM Toolset がバージョン 15.0.7 にリベースされました。

LLVM Toolset がバージョン 15.0.7 に更新されました。主な変更点は、以下のとおりです。

- **-Wimplicit-function-declaration** および **-Wimplicit-int** 警告は、C99 以降ではデフォルトで有効になっています。これらの警告は、Clang 16 以降ではデフォルトでエラーになります。

Bugzilla:2118567

Rust Toolset がバージョン 1.66.1 にリベースされました。

Rust Toolset がバージョン 1.66.1 に更新されました。主な変更点は、以下のとおりです。

- **thread::scope** API は、新しく生成されたスレッドによってローカル変数を安全に借用できる字句スコープを作成します。また、それらのスレッドはスコープが終了する前にすべて終了することが保証されます。
- **hint::black_box** API はコンパイラーの最適化に障壁を追加します。これは、他の方法では最適化されてしまう可能性のあるベンチマークの動作を維持するのに役立ちます。
- **.await** キーワードは、**for** と **IntoIterator** の関係と同様に、**IntoFuture** 特性を使用して変換を行うようになりました。
- ジェネリック関連型 (GAT) を使用すると、特性にジェネリックパラメーターを持つ型エイリアスを含めることができ、型と有効期間の両方にわたる新しい抽象化が可能になります。
- 新しい **let-else** ステートメントでは、条件付きパターンマッチングでローカル変数をバインドし、パターンが一致しない場合に分岐 **else** ブロックを実行できます。
- ラベル付きブロックを使用すると、オプションで式の値を追加して、**break** ステートメントはブロックの末尾にジャンプできます。
- **rust-analyzer** は言語サーバープロトコルの新しい実装であり、多くのエディターで Rust のサポートを可能にします。これは以前の **rls** パッケージを置き換えますが、**rust-analyzer** に移行するにはエディターの設定を調整する必要がある場合があります。
- Cargo には、**Cargo.toml** から依存関係を削除するための新しい **cargo remove** サブコマンドがあります。

Bugzilla:2123900

Go Toolset がバージョン 1.19.6 にリベースされました。

Go Toolset がバージョン 1.19.6 に更新されました。主な変更点は、以下のとおりです。

- 次のパッケージに対するセキュリティー修正:
  - **crypto/tls**
  - **mime/multipart**
  - **net/http**
  - **path/filepath**
- バグ修正:
  - **go** コマンド
  - リンカー
  - ランタイム
  - **crypto/x509** パッケージ
  - **net/http** パッケージ
  - **time** パッケージ

Bugzilla:2175173

tzdata パッケージには /usr/share/zoneinfo/leap-seconds.list ファイルが含まれるようになりました。

以前は、**tzdata** パッケージには、**/usr/share/zoneinfo/leapseconds** ファイルのみが同梱されていました。一部のアプリケーションは、**/usr/share/zoneinfo/leap-seconds.list** ファイルによって提供される代替形式に依存しているため、エラーが発生する可能性があります。

今回の更新により、**tzdata** パッケージには両方のファイルが含まれるようになり、どちらの形式に依存するアプリケーションもサポートされるようになりました。

Bugzilla:2157982

4.13. IDENTITY MANAGEMENT

ホームディレクトリーを小文字に変換するための SSSD のサポート

この機能強化により、ユーザーのホームディレクトリーを小文字に変換するように SSSD を設定できるようになりました。これは、RHEL 環境の大文字と小文字を区別する性質とより適切に統合するのに役立ちます。**/etc/sssd/sssd.conf** ファイルの **[nss]** セクションの **override_homedir** オプションが **%h** テンプレート値を認識するようになりました。**override_homedir** 定義の一部として **%h** を使用すると、SSSD は **%h** をユーザーのホームディレクトリーの小文字に置き換えます。

Jira:RHELPLAN-139430

SSSD が shadow パスワードポリシーを使用した LDAP ユーザーパスワードの変更をサポートするようになりました。

この機能強化により、**/etc/sssd/sssd.conf** ファイルで **ldap_pwd_policy** を **shadow** に設定すると、LDAP ユーザーは LDAP に保存されているパスワードを変更できるようになります。以前は、**ldap_pwd_policy** が **shadow** に設定されている場合、対応する **shadow** LDAP 属性が更新されているかどうかが明確ではないため、パスワードの変更は拒否されました。

さらに、LDAP サーバーが **shadow** 属性を自動的に更新できない場合は、**/etc/sssd/sssd.conf** ファイルで **ldap_chpass_update_last_change** オプションを **True** に設定して、属性を更新するように SSSD に指示します。

Bugzilla:1507035

IdM は min_lifetime パラメーターをサポートするようになりました。

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

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

Bugzilla:2184333

ipapwpolicy ansible-freeipa モジュールが新しいパスワードポリシーオプションをサポートするようになりました。

この更新により、**ansible-freeipa** パッケージに含まれる **ipapwpolicy** モジュールは、追加の **libpwquality** ライブラリーオプションをサポートします。

maxrepeat

同じ文字の最大数を連続して指定します。

maxsequence

単調な文字シーケンスの最大長を指定します (**abcd**)。

dictcheck

パスワードが辞書の単語であるかどうかを確認します。

usercheck

パスワードにユーザー名が含まれるかどうかを確認します。

新しいパスワードポリシーオプションのいずれかが設定されている場合、パスワードの最小長は 6 文字です。新しいパスワードポリシー設定は、新しいパスワードのみに適用されます。

RHEL 7 サーバーと RHEL 8 サーバーが混在する環境では、新しいパスワードポリシー設定は、RHEL 8.4 以降で実行されているサーバーのみに適用されます。ユーザーが IdM クライアントにログインし、IdM クライアントが RHEL 8.3 以前で実行されている IdM サーバーと通信している場合、システム管理者によって設定された新しいパスワードポリシー要件は適用されません。一貫した動作を保証するには、すべてのサーバーを RHEL 8.4 以降にアップグレードします。

Jira:RHELPLAN-137416

IdM が ipanetgroup Ansible 管理モジュールをサポートするようになりました。

Identity Management (IdM) システム管理者は、IdM を NIS ドメインおよびネットグループと統合できます。**ipanetgroup ansible-freeipa** モジュールを使用すると、次のことを実現できます。

- 既存の IdM ネットグループに特定の IdM ユーザー、グループ、ホスト、ホストグループ、およびネストされた IdM ネットグループが含まれていることを確認できます。
- 特定の IdM ユーザー、グループ、ホスト、ホストグループ、およびネストされた IdM ネットグループが既存の IdM ネットグループに存在しないことを確認できます。
- 特定のネットグループが IdM に存在するか存在しないかを確認できます。

Jira:RHELPLAN-137411

クライアントの DNS リゾルバーを指定する新しい ipaclient_configure_dns_resolver および ipaclient_dns_servers Ansible ipaclient ロール変数

以前は、**ansible-freeipa ipaclient** ロールを使用して Identity Management (IdM) クライアントをインストールする場合、インストールプロセス中に DNS リゾルバーを指定できませんでした。インストール前に DNS リゾルバーを設定する必要がありました。

この機能強化により、**ipaclient** ロールを使用して IdM クライアントをインストールするときに、**ipaclient_configure_dns_resolver** 変数と **ipaclient_dns_servers** 変数を使用して DNS リゾルバーを指定できるようになりました。その結果、**ipaclient** ロールは、**ansible-freeipa ipaserver** ロールが IdM サーバー上で行うのと同様の方法で、**resolv.conf** ファイル、**NetworkManager** および **systemd-resolved** ユーティリティーを変更して、クライアント上で DNS リゾルバーを設定します。その結果、**ipaclient** ロールを使用して IdM クライアントをインストールする際の DNS の設定がより効率的になりました。

注記

ipa-client-install コマンドラインインストーラーを使用して IdM クライアントをインストールするには、インストール前に DNS リゾルバーを設定する必要があります。

Jira:RHELPLAN-137406

ipaclient ロールを使用して IdM クライアントを OTP とともにインストールするには、Ansible コントローラーを事前に変更する必要はありません。

以前は、Ansible コントローラーの **kinit** コマンドは、Identity Management (IdM) クライアントのデプロイメント用のワンタイムパスワード (OTP) を取得するための前提条件でした。Red Hat Ansible Automation Platform (AAP) では、コントローラーで OTP を取得する必要性が問題でした。AAP では、**krb5-workstation** パッケージがデフォルトでインストールされませんでした。

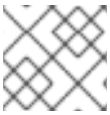
この更新により、管理者の TGT に対するリクエストは、最初に指定または検出された IdM サーバーに委任されるようになりました。その結果、Ansible コントローラーを追加変更することなく、OTP を使用して IdM クライアントのインストールを承認できるようになりました。これにより、AAP での **ipaclient** ロールの使用が簡素化されます。

Jira:RHELPLAN-137403

IdM は、Kerberos チケットに MS-PAC 構造体の存在を適用するようになりました。

RHEL 9.2 以降、セキュリティーを向上させるために、Identity Management (IdM) および MIT Kerberos は、RHEL IdM Kerberos Distribution Center (KDC) が発行する Kerberos チケットに 特権属性証明書 (MS-PAC) 構造体の存在を適用するようになりました。

2022 年 11 月、CVE-2022-37967 への対応として、Microsoft はサーバーのチェックサムではなく、MS-PAC 構造体全体で計算される拡張署名を導入しました。RHEL 9.2 以降、IdM KDC が発行する Kerberos チケットに拡張署名も含まれるようになりました。



注記

IdM では、拡張署名の存在はまだ強制されていません。

Jira:RHELPLAN-159146

FIPS 140-3 準拠のキー暗号化を有効にする KDC の新しいレルム設定テンプレート

今回の更新により、**/var/kerberos/krb5kdc/kdc.conf** ファイルに新しい **EXAMPLE.COM** レルム設定の例が提供されます。これにより、次の 2 つの変更が行われます。

- FIPS 140-3 準拠の **AES HMAC SHA-2** ファミリーが、キー暗号化でサポートされるタイプのリストに追加されました。
- KDC マスターキーの暗号化タイプが **AES 256 HMAC SHA-1** から **AES 256 HMAC SHA-384** に切り替えられます。

警告

この更新は、スタンドアロンの MIT レルムに関するものです。RHEL Identity Management で Kerberos Distribution Center (KDC) 設定を変更しないでください。

新しいレルムには、この設定テンプレートを使用することを推奨します。テンプレートは、すでにデプロイされているレルムには影響しません。テンプレートに従ってレルムの設定をアップグレードすることを計画している場合は、次の点を考慮してください。

マスターキーをアップグレードするには、KDC 設定の設定を変更するだけでは不十分です。MIT Kerberos ドキュメントに記載されているプロセスに従います。https://web.mit.edu/kerberos/krb5-1.20/doc/admin/database.html#updating-the-master-key

AES HMAC SHA-2 ファミリーをキー暗号化のサポートされているタイプに追加しても、KDC 内の既存のエントリーに影響しないため、常に安全です。キーは、新しいプリンシパルを作成するとき、または認証情報を更新するときのみに生成されます。この新しいタイプのキーは、既存のキーに基づいて生成できないことに注意してください。これらの新しい暗号化タイプを特定のプリンシパルで使用できるようにするには、認証情報を更新する必要があります。つまり、サービスプリンシパルのキータブも更新する必要があります。

プリンシパルが **AES HMAC SHA-2** キーを備えてはならない唯一のケースは、Active Directory (AD) クロスレルム ticket-granting ticket (TGT) のものです。AD は RFC8009 を実装していないため、**AES HMAC SHA-2** 暗号化タイプファミリーを使用しません。したがって、**AES HMAC SHA-2** で暗号化されたクロスレルム TGT を使用するクロスレルム TGS-REQ は失敗します。MIT Kerberos クライアントが AD に対して **AES HMAC SHA-2 を** 使用しないようにする最善の方法は、AD クロスレルムプリンシパルに **AES HMAC SHA-2** キーを提供しないことです。これを行うには、AD ですべてサポートされているキー暗号化タイプの明示的なリストを使用して、クロスレルム TGT エントリーを作成してください。

```
  kadmin.local <<EOF
  add_principal +requires_preauth -e aes256-cts-hmac-sha1-96,aes128-cts-hmac-sha1-96 -pw
[password] krbtgt/[MIT realm]@[AD realm]
  add_principal +requires_preauth -e aes256-cts-hmac-sha1-96,aes128-cts-hmac-sha1-96 -pw
[password] krbtgt/[AD realm]@[MIT realm]
  EOF
```

MIT Kerboros クライアントが **AES HMAC SHA-2** 暗号化タイプを使用するようにするには、これらの暗号化タイプをクライアントと KDC 設定の両方で **permitted** に設定する必要もあります。RHEL では、この設定は crypto-policy システムによって管理されます。たとえば、RHEL 9 では、**DEFAULT** 暗号化ポリシーを使用するホストは **AES HMAC SHA-2** および **AES HMAC SHA-1** で暗号化されたチケットを許可しますが、**FIPS** 暗号化ポリシーを使用するホストは **AES HMAC SHA-2** のみを受け入れます。

Bugzilla:2068535

設定ファイルを使用して pam_pwhistory を設定します。

この更新により、**/etc/security/pwhistory.conf** 設定ファイルで **pam_pwhistory** モジュールを設定できるようになりました。**pam_pwhistory** モジュールは、パスワード変更履歴を管理するために、各ユーザーの最後のパスワードを保存します。**authselect** にもサポートが追加され、**pam_pwhistory** モジュールを PAM スタックに追加できるようになりました。

Bugzilla:2126640、Bugzilla:2142805

IdM が新しい Active Directory 証明書マッピングテンプレートをサポートするようになりました

Active Directory (AD) ドメイン管理者は、**altSecurityIdentities** 属性を使用して、証明書を AD 内のユーザーに手動でマッピングできます。この属性には 6 つの値がサポートされていますが、3 つのマッピングは安全ではないと考えられています。2022 年 5 月 10 日のセキュリティー更新 の一部として、この更新プログラムがドメインコントローラーにインストールされると、すべてのデバイスが互換モードになります。証明書がユーザーに弱くマッピングされている場合、認証は期待どおりに行われます

が、完全強制モードと互換性のない証明書を示す警告メッセージがログに記録されます。2023 年 11 月 14 日以降、すべてのデバイスは完全強制モードに更新され、証明書が強力なマッピング基準を満たさない場合、認証は拒否されます。

IdM は新しいマッピングテンプレートをサポートするようになったため、AD 管理者は両方を維持することなく、新しいルールを使用できるようになりました。IdM は、次の新しいマッピングテンプレートをサポートするようになりました。

- シリアル番号: **LDAPU1:(altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<SR>{serial_number!hex_ur})**
- Subject Key Id: **LDAPU1:(altSecurityIdentities=X509:<SKI>{subject_key_id!hex_u})**
- User SID: **LDAPU1:(objectsid={sid})**

新しい SID 拡張子を使用して証明書を再発行したくない場合は、AD のユーザーの **altSecurityIdentities** 属性に適切なマッピング文字列を追加して、手動マッピングを作成できます。

Bugzilla:2087247

samba がバージョン 4.17.5 にリベースされました。

samba パッケージがアップストリームバージョン 4.17.5 にアップグレードされ、以前のバージョンに対するバグ修正と拡張機能が提供されています。最も注目すべき変更点:

- 以前のリリースでのセキュリティーの向上は、高メタデータワークロードのサーバーメッセージブロック (SMB) サーバーのパフォーマンスに影響を与えました。この更新により、このシナリオでのパフォーマンスが向上します。
- 詳細なステータス情報を JSON 形式で表示するために、**--json** オプションが **smbstatus** ユーティリティーに追加されました。
- **samba.smb.conf** モジュールと **samba.samba3.smb.conf** モジュールが **smbconf** Python API に追加されました。これらを Python プログラムで使用すると、Samba 設定をネイティブに読み取ったり、必要に応じて書き込むことができます。

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

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

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

重要な変更点の詳細については、更新する前に、アップストリームリリースノート をお読みください。

Bugzilla:2131993

ipa-client-install が PKINIT による認証をサポートするようになりました。

以前は、**ipa-client-install** はパスワードベースの認証のみをサポートしていました。この更新により、PKINIT による認証のための **ipa-client-install** のサポートが提供されます。

以下に例を示します。

```
ipa-client-install --pkinit-identity=FILE:/path/to/cert.pem,/path/to/key.pem --pkinit-
anchor=FILE:/path/to/cacerts.pem
```

PKINIT 認証を使用するには、IdM と PKINIT 証明書の CA チェーンの間に信頼を確立する必要があります。詳細については、**ipa-cacert-manage(1)** man ページを参照してください。また、証明書 ID マッピングルールは、ホストの PKINIT 証明書を、ホストレコードを追加または変更する権限を持つプリンシパルにマップする必要があります。詳細については、**ipa certmaprule-add** man ページを参照してください。

Bugzilla:2143224

Red Hat IdM と Certificate System が EST プロトコルをサポートするようになりました。

Enrollment over Secure Transport (EST) は、RFC 7030 で指定されている新しい Certificate System サブシステム機能で、認証局 (CA) から証明書をプロビジョニングするために使用されます。EST は、**/getcacerts**、**/simpleenroll**、**/simplereenroll** などのサーバー側の操作を実装します。

Red Hat は、Certificate System で EST とオリジナルの Simple Certificate Enrollment Protocol (SCEP) の両方をサポートしていることに注意してください。

Bugzilla:1849834

ネガティブキャッシュの使用を強化します。

この更新により、セキュリティー識別子 (SID) によるルックアップの SSSD パフォーマンスが向上します。存在しない SID を個々のドメインのネガティブキャッシュに保存し、SID が属するドメインを要求するようになりました。

Bugzilla:1766490

Directory Server が TLS の ECDSA 秘密キーをサポートするようになりました。

以前は、RSA より強力な暗号化アルゴリズムを使用して Directory Server 接続を保護することはできませんでした。この機能強化により、Directory Server は ECDSA キーと RSA キーの両方をサポートするようになりました。

Bugzilla:2096795

Directory Server が検索操作の拡張ログをサポートするようになりました。

以前は、アクセスログの記録には、一部の検索操作で **etime** 値が非常に大きい理由が示されませんでした。このリリースでは、インデックス検索 (データベース読み取り操作) の数や各検索操作ごとのインデックス検索の全体的な継続時間などの統計のログを有効にすることができます。これらの統計レコードは、**etime** 値がリソースを非常に高価にする理由を分析するのに役立ちます。

Bugzilla:1859271

NUNC_STANS エラーログレベルは、新しい 1048576 ログレベルに置き換えられました。

以前は、パスワードポリシーの問題を簡単にデバッグできませんでした。エラーログの新しいログレベル **1048576** を使用すると、次のパスワードポリシー情報を確認できるようになりました。

- どのローカルポリシーがパスワードの更新を拒否または許可するか。
- どのパスワードポリシーに違反したか。

Bugzilla:2057070

Directory Server ではセキュリティーログが導入されています。

問題を長期にわたって適切に追跡するために、Directory Server にはセキュリティーデータを維持する特殊なログが用意されるようになりました。セキュリティーログは、すべての情報が含まれるアクセスログと比較して、すぐには更新されず、消費するディスクリソースも少なくなりますが、セキュリティーデータを取得するために高価な解析が必要になります。

新しいサーバーログには、認証イベント、認可の問題、DoS/TCP 攻撃、その他のイベントなどのセキュリティーイベントが記録されます。

Directory Server は、セキュリティーログを他のログファイルとともに **/var/log/dirsrv/slapd-instance_name/** ディレクトリーに保存します。

Bugzilla:2093981

Directory Server でアーカイブされたログファイルを圧縮できるようになりました。

以前は、アーカイブされたログファイルは圧縮されていませんでした。このリリースでは、アクセス、エラー、監査、監査失敗ログ、セキュリティーログファイルの圧縮を有効にして、ディスク領域を節約できます。デフォルトではセキュリティーログファイルの圧縮のみが有効になっていることに注意してください。

圧縮を管理するには、**cn=config** エントリーで次の新しい設定属性を使用します。

- アクセスログ用の **nsslapd-accesslog-compress**
- エラーログ用の **nsslapd-errorlog-compress**
- 監査ログ用の **nsslapd-auditlog-compress**
- 監査失敗ログ用の **nsslapd-auditfaillog-compress**
- セキュリティーログ用の **nsslapd-securelog-compress**

Bugzilla:1132524

新しい pamModuleIsThreadSafe 設定オプションが利用可能になりました

PAM モジュールがスレッドセーフである場合、新しい **pamModuleIsThreadSafe** 設定オプションを **yes** に設定することで、その特定のモジュールの PAM 認証のスループットと応答時間を改善できます。

```
pamModuleIsThreadSafe: yes
```

この設定は、PAM モジュール設定エントリー (**cn=PAM Pass Through Auth,cn=plugins,cn=config** の子) に適用されます。

dse.ldif 設定ファイルまたは **ldapmodify** コマンドの **pamModuleIsThreadSafe** オプションを使用します。**ldapmodify** コマンドを使用するには、サーバーを再起動する必要があることに注意してください。

Bugzilla:2142639

Directory Server が証明書バンドルをインポートできるようになりました。

以前は、**dsconf** または **dsctl** ユーティリティーを使用して証明書バンドルを追加しようとすると、手順がエラーで失敗し、証明書バンドルはインポートされませんでした。このような動作は、一度に 1 つの証明書しかインポートできない **certutil** ユーティリティーが原因で発生しました。この更新により、

Directory Server は **certutil** の問題を回避し、証明書バンドルが正常に追加されました。

Bugzilla:1878808

デフォルトの動作の変更: Directory Server が、データベース追加時とまったく同じスペルの DN を返すようになりました。

cn=config エントリーの新しい **nsslapd-return-original-entrydn** パラメーターを使用すると、検索操作中に Directory Server がエントリーの識別名 (DN) を返す方法を管理できます。

デフォルトでは、**nsslapd-return-original-entrydn** パラメーターは **on** に設定されており、Directory Server は最初にデータベースに追加されたときとまったく同じ DN を返します。たとえば、設定がオンの状態で、エントリー **uid=User,ou=PEople,dc=ExaMPIE,DC=COM** を追加または変更すると、Directory Server はエントリーの DN のスペルと同じもの (**uid=User,ou=PEople,dc=ExaMPIE,DC=COM**) を返します。

nsslapd-return-original-entrydn パラメーターを **off** に設定すると、Directory Server は、エントリーの Relative DN (RDN) と、データベース接尾辞設定の **cn=userroot,cn=ldbm database,cn=plugins,cn=config** に格納されているベース DN を組み合わせてエントリー DN を生成します。ベース DN を **ou=people,dc=example,dc=com** に設定し、**nsslapd-return-original-entrydn** 設定を **off** をオフにした場合、Directory Server は、データベースにエントリーを追加したときの DN のスペルではなく、**uid=User,ou=people,dc=example,dc=com** を検索時に返します。

Bugzilla:2075017

MIT Kerberos はチケット署名および拡張 KDC MS-PAC 署名をサポートします

この更新により、Red Hat が使用する MIT Kerberos は、最近の CVE に対応して Microsoft が導入した、2 種類の特権属性証明書 (PAC) 署名のサポートを実装します。具体的には、次の署名がサポートされます。

- チケット署名
  - KB4598347 でリリース
  - "Bronze-Bit" 攻撃とも呼ばれる CVE-2020-17049 への対処
- 拡張 KDC 署名
  - KB5020805 でリリース
  - CVE-2022-37967 への対処

RHSA-2023:2570 および **krb5-1.20.1-6.el9** も参照してください。

Bugzilla:2165827

Directory Server 監査ログ用の新しい nsslapd-auditlog-display-attrs 設定パラメーター

以前は、監査ログイベント内のターゲットエントリーを識別する方法は、識別名 (DN) のみでした。新しい **nsslapd-auditlog-display-attrs** パラメーターを使用すると、監査ログに追加の属性を表示するように Directory Server を設定できます。これにより、変更されたエントリーに関する詳細情報が提供されます。

たとえば、**nsslapd-auditlog-display-attrs** パラメーターを **cn** に設定すると、監査ログの出力にはエントリー **cn** 属性が表示されます。変更されたエントリーのすべての属性を含めるには、パラメーター値としてアスタリスク (*) を使用します。

詳細は、nsslapd-auditlog-display-attrs を参照してください。

Bugzilla:2136610

4.14. デスクトップ

ワークスペースを切り替えるためのスワイプを無効にします。

以前は、3 本の指で上下にスワイプすると、常にタッチスクリーン上のワークスペースが切り替わっていました。このリリースでは、ワークスペースの切り替えを無効にすることができます。

詳細については、スワイプによるワークスペース切り替えの無効化 を参照してください。

Bugzilla:2154358

Wayland が Aspeed GPU で有効になりました。

以前は、Aspeed GPU ドライバーのパフォーマンスが十分ではなく、Wayland セッションを実行できませんでした。この問題を回避するために、Aspeed GPU に対して Wayland セッションが無効になりました。

このリリースでは、ドライバーのパフォーマンスが大幅に向上し、Wayland セッションの応答性が向上しました。その結果、Aspeed GPU では Wayland セッションがデフォルトで有効になるようになりました。

Bugzilla:2131203

デスクトップ上のカスタム右クリックメニュー

デスクトップの背景を右クリックしたときに開くメニューをカスタマイズできるようになりました。任意のコマンドを実行するカスタムエントリーをメニューに作成できます。

メニューをカスタマイズするには、デスクトップの右クリックメニューのカスタマイズ を参照してください。

Bugzilla:2160553

4.15. WEB コンソール

特定の暗号化サブポリシーが Web コンソールで使用できるようになりました。

RHEL Web コンソールの今回の更新により、**Change crypto policy** ダイアログのオプションが拡張されました。4 つのシステム全体の暗号化ポリシーに加えて、グラフィカルインターフェイスを介して次のサブポリシーも適用できるようになりました。

- **DEFAULT:SHA1** は、**SHA-1** アルゴリズムが有効になっている **DEFAULT** ポリシーです。
- **LEGACY:AD-SUPPORT** は、Active Directory サービスの相互運用性を向上させるセキュリティーの低い設定を持つ **LEGACY** ポリシーです。
- **FIPS:OSPP** は、情報技術セキュリティー評価標準の Common Criteria に触発されたさらなる制限を備えた **FIPS** ポリシーです。

Jira:RHELPLAN-137505

Web コンソールは、LUKS で暗号化されたルートボリュームを NBDE にバインドするための追加手順を実行するようになりました。

この更新により、RHEL Web コンソールは、LUKS で暗号化されたルートボリュームを Network-

Bound Disk Encryption (NBDE) デプロイメントにバインドするために必要な追加の手順を実行します。暗号化されたルートファイルシステムと Tang サーバーを選択した後、カーネルコマンドラインへの **rd.neednet=1** パラメーターの追加、**clevis-dracut** パッケージのインストール、および初期 RAM ディスク (**initrd**) の再生成をスキップできます。非ルートファイルシステムの場合、Web コンソールは、**remote-cryptsetup.target** および **clevis-luks-akspass.path systemd** ユニットを有効にし、**clevis-systemd** パッケージをインストールし、**_netdev** パラメーターを **fstab** および **crypttab** 設定ファイルに追加するようになりました。その結果、LUKS で暗号化されたルートボリュームの自動ロック解除のための NBDE デプロイメントを作成するときに、すべての Clevis クライアント設定手順でグラフィカルインターフェイスを使用できるようになりました。

Jira:RHELPLAN-139125

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

ルーティングルールは名前でルートテーブルを検索できます。

この更新により、**rhel-system-roles.network** RHEL システムロールが、ルーティングルールを定義するときに名前によるルートテーブルの検索をサポートするようになりました。この機能は、ネットワークセグメントごとに異なるルーティングルールが必要な複雑なネットワーク設定を迅速にナビゲートします。

Bugzilla:2131293

network システムロールが DNS 優先度値の設定をサポートするようになりました

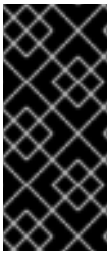
この機能拡張により、RHEL **network** システムロールに **dns_priority** パラメーターが追加されました。このパラメーターは、**-2147483648** から **2147483647** までの値に設定できます。デフォルト値は **0** です。値が小さいほど優先順位が高くなります。負の値を指定すると、システムロールによって、より大きな数値の優先度を持つ他の設定が除外されることに注意してください。したがって、少なくとも 1 つの負の優先度値が存在する場合、システムロールは優先度値が最も低い接続プロファイルの DNS サーバーのみを使用します。

その結果、**network** システムロールを使用して、さまざまな接続プロファイル内の DNS サーバーの順序を定義できるようになりました。

Bugzilla:2133858

vpn RHEL システムロールの新しい IPsec カスタマイズパラメーター

特定のネットワークデバイスを正しく動作させるには、IPsec のカスタマイズが必要なため、**vpn** RHEL システムロールに次のパラメーターが追加されました。



重要

高度な知識がないかぎり、次のパラメーターを変更しないでください。ほとんどのシナリオでは、カスタマイズする必要はありません。

さらに、セキュリティー上の理由から、Ansible Vault を使用して **shared_key_content** パラメーターの値を暗号化します。

- トンネルパラメーター:
  - **shared_key_content**
  - **ike**

- **esp**
- **ikelifetime**
- **salifetime**
- **retransmit_timeout**
- **dpddelay**
- **dpdtimeout**
- **dpdaction**
- **leftupdown**

- ホストごとのパラメーター:
  - **leftid**
  - **rightid**

その結果、**vpn** ロールを使用して、幅広いネットワークデバイスへの IPsec 接続を設定できます。

Bugzilla:2119102

selinux RHEL システムロールが local パラメーターをサポートするようになりました

今回の **selinux** RHEL システムロールの更新により、**local** パラメーターのサポートが導入されました。このパラメーターを使用すると、ローカルポリシーの変更のみを削除し、組み込みの SELinux ポリシーを保持できます。

Bugzilla:2128843

ha_cluster システムロールが、firewall、selinux、certificate システムロールの自動実行をサポートするようになりました

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

firewall および selinux システムロールを使用してポートアクセスを管理する

firewalld および **selinux** サービスを実行するようにクラスターのポートを設定するには、新しいロール変数 **ha_cluster_manage_firewall** および **ha_cluster_manage_selinux** を **true** に設定します。これにより、**firewall** および **selinux** システムロールを使用するようにクラスターが設定され、**ha_cluster** システムロール内で該当する操作が自動化および実行されます。これらの変数がデフォルト値の **false** に設定されている場合、ロールは実行されません。このリリースでは、ファイアウォールはデフォルトで設定されなくなりました。これは、ファイアウォールで **ha_cluster_manage_firewall** が **true** に設定されている場合のみ、設定されるためです。

certificate システムロールを使用して pcsd 秘密鍵と証明書のペアを作成する

ha_cluster システムロールは、**ha_cluster_pcsd_certificates** ロール変数をサポートするようになりました。この変数を設定すると、その値が **certificate** システムロールの **certificate_requests** 変数に渡されます。これは、**pcsd** の秘密鍵と証明書のペアを作成するための代替方法を提供します。

Bugzilla:2130010

postfix RHEL システムロールが、firewall および selinux RHEL システムロールを使用してポートアクセスを管理できるようになりました

この機能強化により、新しいロール変数 **postfix_manage_firewall** および **postfix_manage_selinux** を使用してポートアクセスの管理を自動化できます。

- これらが **true** に設定されている場合、各ロールはポートアクセスの管理に使用されます。
- これらが **false** (デフォルト) に設定されている場合、ロールは関与しません。

Bugzilla:2130329

vpn RHEL システムロールが、firewall および selinux ロールを使用してポートアクセスを管理できるようになりました

この機能拡張により、**firewall** および **selinux** ロールを介した **vpn** RHEL システムロールでのポートアクセスの管理を自動化できます。新しいロール変数 **vpn_manage_firewall** および **vpn_manage_selinux** を **true** に設定すると、ロールはポートアクセスを管理します。

Bugzilla:2130344

logging RHEL システムロールが、ポートアクセスと証明書の生成をサポートするようになりました

この機能強化により、logging ロールを使用してポートアクセスを管理し、新しいロール変数で証明書を生成できるようになります。新しいロール変数 **logging_manage_firewall** および **logging_manage_selinux** を **true** に設定すると、ロールはポートアクセスを管理します。証明書を生成するための新しいロール変数は、**logging_certificates** です。タイプと使用法は、**certificate** ロール **certificate_requests** と同じです。**logging** ロールを使用して、これらの操作を直接自動化できるようになりました。

Bugzilla:2130357

metrics RHEL システムロールが、firewall および selinux ロールを使用してポートアクセスを管理できるようになりました

この機能強化により、ポートへのアクセスを制御できるようになります。新しいロール変数 **metrics_manage_firewall** および **metrics_manage_firewall** を **true** に設定すると、ロールはポートアクセスを管理します。**metrics** ロールを使用して、これらの操作を自動化し、直接実行できるようになりました。

Bugzilla:2133528

nbde_server RHEL システムロールが、firewall および selinux ロールを使用してポートアクセスを管理できるようになりました

この機能強化により、**firewall** および **selinux** ロールを使用してポートアクセスを管理できるようになります。新しいロール変数 **nbde_server_manage_firewall** および **nbde_server_manage_selinux** を **true** に設定すると、ロールはポートアクセスを管理します。**nbde_server** ロールを使用して、これらの操作を直接自動化できるようになりました。

Bugzilla:2133930

initscripts ネットワークプロバイダーが、デフォルトゲートウェイのルートメトリクス設定をサポートするようになりました

この更新により、**rhel-system-roles.network** RHEL システムロールの **initscripts** ネットワークプロバイダーを使用して、デフォルトゲートウェイのルートメトリクスを設定できるようになりました。

このような設定の理由としては、次のことが考えられます。

- トラフィック負荷をさまざまなパスに分散する
- プライマリールートとバックアップルートを指定する
- ルーティングポリシーを利用して、特定のパスを介して特定の宛先にトラフィックを送信する

Bugzilla:2134202

cockpit RHEL システムロールと firewall、selinux、および certificate ロールの統合

この機能拡張により、**cockpit** ロールを **firewall** ロール、ポートアクセスを管理するための **selinux** ロール、および証明書を生成するための **証明書** ロールと統合できるようになります。

ポートアクセスを制御するには、新しい **cockpit_manage_firewall** 変数と **cockpit_manage_selinux** 変数を使用します。どちらの変数もデフォルトでは **false** に設定されており、実行されません。これらを **true** に設定すると、**firewall** および **selinux** ロールが RHEL Web コンソールサービスポートアクセスを管理できるようになります。その後、操作は **cockpit** ロール内で実行されます。

ファイアウォールと SELinux のポートアクセスを管理する責任があることに注意してください。

証明書を生成するには、新しい **cockpit_certificates** 変数を使用します。この変数はデフォルトで **false** に設定されており、実行されません。この変数は、**certificate** ロールで **certificate_request** 変数を使用するのと同じ方法で使用できます。その後、**cockpit** ロールは **certificate** ロールを使用して RHEL Web コンソール証明書を管理します。

Bugzilla:2137663

Active Directory と直接統合するための新しい RHEL システムロール

新しい **rhel-system-roles.ad_integration** RHEL システムロールが **rhel-system-roles** パッケージに追加されました。その結果、管理者は RHEL システムと Active Directory ドメインの直接統合を自動化できるようになりました。

Bugzilla:2140795

Red Hat Insights と Subscription Management のための新しい Ansible ロール

rhel-system-roles パッケージには、リモートホスト設定 (**rhc**) システムロールが含まれるようになりました。このロールにより、管理者は RHEL システムを Red Hat Subscription Management (RHSM) および Satellite サーバーに簡単に登録できるようになります。デフォルトでは、**rhc** システムロールを使用してシステムを登録すると、システムは Red Hat Insights に接続します。新しい **rhc** システムロールを使用すると、管理者はマネージドノードで次のタスクを自動化できるようになりました。

- システムの自動更新、修復、タグなど、Red Hat Insights への接続を設定します。
- リポジトリーを有効または無効にします。
- 接続に使用するプロキシーを設定します。
- システムのリリースを設定します。

これらのタスクを自動化する方法の詳細については、RHC システムロールを使用したシステムの登録を参照してください。

Bugzilla:2141330

クローン MAC アドレスのサポートを追加しました。

クローンされた MAC アドレスは、マシンの MAC アドレスと同じデバイスの WAN ポートの MAC アド

レスです。この更新により、ユーザーは MAC アドレスを使用してボンディングインターフェイスまたはブリッジインターフェイスを指定したり、ボンディングインターフェイスまたはブリッジインターフェイスのデフォルトの MAC アドレスを取得するために **random** または **preserve** などの戦略を指定したりできるようになります。

Bugzilla:2143768

Microsoft SQL Server Ansible ロールは非同期高可用性レプリカをサポートします。

以前は、Microsoft SQL Server Ansible ロールは、プライマリー、同期、監視の高可用性レプリカのみをサポートしていました。**mssql_ha_replica_type** 変数を **asynchronous** に設定して、新規または既存のレプリカに対して非同期レプリカタイプを設定できるようになりました。

Bugzilla:2151282

Microsoft SQL Server Ansible ロールは読み取りスケールクラスタータイプをサポートします。

以前は、Microsoft SQL Ansible ロールは外部クラスタータイプのみをサポートしていました。これで、新しい変数 **mssql_ha_ag_cluster_type** を使用してロールを設定できるようになりました。デフォルト値は **external** です。これを使用して Pacemaker でクラスターを設定します。Pacemaker を使用せずにクラスターを設定するには、その変数に値 **none** を使用します。

Bugzilla:2151283

Microsoft SQL Server Ansible ロールは TLS 証明書を生成できます。

以前は、Microsoft SQL Ansible ロールを設定する前に、ノード上で TLS 証明書と秘密キーを手動で生成する必要がありました。この更新により、Microsoft SQL Server Ansible ロールは、その目的で **redhat.rhel_system_roles.certificate** ロールを使用できるようになりました。これで、**mssql_tls_certificates** 変数を **certificate** ロールの **certificate_requests** 変数の形式で設定して、ノード上に TLS 証明書と秘密鍵を生成できるようになりました。

Bugzilla:2151284

Microsoft SQL Server Ansible ロールは SQL Server バージョン 2022 の設定をサポートします。

以前は、Microsoft SQL Ansible ロールは SQL Server バージョン 2017 とバージョン 2019 の設定のみをサポートしていました。この更新プログラムでは、Microsoft SQL Ansible ロールの SQL Server バージョン 2022 のサポートが提供されます。新しい SQL Server 2022 を設定するか、SQL Server をバージョン 2019 からバージョン 2022 にアップグレードするために、**mssql_version** 値を **2022** に設定できるようになりました。SQL Server をバージョン 2017 からバージョン 2022 にアップグレードすることはできないことに注意してください。

Bugzilla:2153428

Microsoft SQL Server Ansible ロールは、Active Directory 認証の設定をサポートします。

この更新により、Microsoft SQL Ansible ロールは SQL Server の Active Directory 認証の設定をサポートします。これで、**mssql_ad_** 接頭辞を使用して変数を設定することで、Active Directory 認証を設定できるようになりました。

Bugzilla:2163709

journald RHEL システムロールが利用可能になりました

journald サービスは、ログデータを収集し、一元化されたデータベースに保存します。この機能強化により、**journald** システムロール変数を使用して **systemd** ジャーナルの設定を自動化し、Red Hat Ansible Automation Platform を使用して永続的なログを設定できるようになりました。

Bugzilla:2165175

ha_cluster システムロールがクォーラムデバイス設定をサポートするようになりました

クォーラムデバイスは、クラスターのサードパーティー調停デバイスとして機能します。クォーラムデバイスは、偶数のノードを持つクラスターに推奨されます。2 ノードクラスターでクォーラムデバイスを使用すると、スプリットブレインの状況で存続するノードをより適切に判別できます。**ha_cluster** システムロール (クラスター用の **qdevice** と調停ノード用の **qnetd** の両方) を使用してクォーラムデバイスを設定できるようになりました。

Bugzilla:2140804

4.17. 仮想化

ハードウェア暗号化デバイスを自動的にホットプラグできるようになりました。

以前は、仲介デバイスが開始される前に暗号化デバイスがホスト上に存在していた場合のみ、パススルー用の暗号化デバイスを定義できました。これで、仮想マシン (VM) にパススルーするすべての暗号化デバイスをリストする仲介デバイスマトリックスを定義できるようになりました。その結果、指定された暗号化デバイスは、後で使用可能になった場合、実行中の VM に自動的にパススルーされます。また、デバイスが使用できなくなると、デバイスは VM から削除されますが、ゲストオペレーティングシステムは正常に動作し続けます。

Bugzilla:1871126

IBM Z 上の PCI パススルーデバイスのパフォーマンスの向上

この更新では、I/O 処理に対する複数の改善により、IBM Z ハードウェアでの PCI パススルー実装が強化されました。その結果、IBM Z ホスト上の KVM 仮想マシン (VM) にパススルーされる PCI デバイスのパフォーマンスが大幅に向上しました。

さらに、ISM デバイスを IBM Z ホスト上の VM に割り当てることができるようになりました。

Bugzilla:1871143

新しいパッケージ: passt

この更新では、**passt** パッケージが追加され、仮想マシンで **passt** ユーザーモードネットワークバックエンドを使用できるようになります。

passt の使用に関する詳細は、passt ユーザー空間の接続設定 を参照してください。

Bugzilla:2131015

zPCI デバイスの割り当て

IBM Z ハードウェア上で実行される RHEL でホストされる仮想マシン (VM) に、zPCI デバイスをパススルーデバイスとして接続できるようになりました。たとえば、これを使用すると、仮想マシンで NVMe フラッシュドライブを使用できます。

Jira:RHELPLAN-59528

新しいパッケージ: python-virt-firmware

この更新では、Open Virtual Machine Firmware (OVMF) ファームウェアイメージを処理するためのツールが含まれる **python-virt-firmware** パッケージが追加されます。これらのツールは、たとえば次の目的で使用できます。

- ファームウェアイメージのコンテンツの出力
- **edk2** 変数ストアの更新
- QEMU で仮想マシンを起動せずにセキュアブートキーの登録を処理する

結果として、これらにより OVMF イメージのビルドが容易になります。

Bugzilla:2089785

4.18. サポート性

sos ユーティリティーは 4 週間ごとの更新頻度に移行しています。

RHEL マイナーリリースで **sos** 更新をリリースする代わりに、**sos** ユーティリティーのリリース頻度が 6 か月から 4 週間に変更されます。**sos** パッケージの更新の詳細については、RPM 変更ログで 4 週間ごとに確認できます。また、**sos** 更新の概要は RHEL リリースノートで 6 か月ごとに確認できます。

Bugzilla:2164987

sos clean コマンドで IPv6 アドレスが難読化されるようになりました。

以前は、**sos clean** コマンドは IPv6 アドレスを難読化せず、収集された **sos** レポートに顧客の機密データの一部が残っていました。この更新により、**sos clean** は期待どおりに IPv6 アドレスを検出し、難読化します。

Bugzilla:2134906

4.19. コンテナー

新しい podman RHEL システムロールが利用可能になりました。

Podman 4.2 以降では、**podman** システムロールを使用して、Podman 設定、コンテナー、および Podman コンテナーを実行する **systemd** サービスを管理できるようになりました。

Jira:RHELPLAN-118705

Podman は監査用のイベントをサポートするようになりました。

Podman v4.4 以降、コンテナーに関するすべての関連情報を 1 つのイベントと **journald** エントリーから直接収集できるようになりました。Podman 監査を有効にするには、**container.conf** 設定ファイルを変更し、**events_container_create_inspect_data=true** オプションを **engine** セクションに追加します。データは JSON 形式であり、**podman container inspect** コマンドからのものと同じです。詳細については、Podman 4.4 の新しいコンテナーイベントと監査機能の使用方法 を参照してください。

Jira:RHELPLAN-136602

container-tools メタパッケージが更新された

Podman、Buildah、Skopeo、crun、runc ツールを含む **container-tools** RPM メタパッケージが利用可能になりました。この更新では、以前のバージョンに対する一連のバグ修正と機能強化が適用されます。

Podman v4.4 の注目すべき変更点は次のとおりです。

- Podman を使用して systemd サービスを簡単に作成および保守できる新しい systemd ジェネレーターである Quadlet を紹介します。
- 新しいコマンド **podman network update** が追加されました。これは、コンテナーと Pod のネットワークを更新します。
- buildah のバージョンを表示する新しいコマンド **podman buildx version** が追加されました。
- コンテナーに起動ヘルスチェックを設定できるようになり、通常のヘルスチェックがアクティブになる前にコマンドを実行してコンテナーが完全に起動していることを確認できるようになりました。
- **podman --dns** コマンドを使用して、カスタム DNS サーバーの選択をサポートします。
- Fulcio と Rekor を使用した sigstore 署名の作成と検証が利用できるようになりました。
- Docker との互換性が向上しました (新しいオプションとエイリアス)。
- Podman の Kubernetes 統合の改善 - コマンド **podman kubegenerate** および **podman kube play** が利用可能になり、**podmangenerate kube** および **podman play kube** コマンドに置き換わりました。**podman generated kube** および **podman play kube** コマンドは引き続き使用できますが、新しい **podman kube** コマンドを使用することを推奨します。
- **podman kube play** コマンドによって作成された Systemd 管理の Pod は、**io.containers.sdnotify** アノテーション (または特定のコンテナーの場合は **io.containers.sdnotify/$name**) を使用して sd-notify と統合されるようになりました。
- **podman kube play** によって作成された Systemd 管理の Pod は、 **io.containers.auto-update** アノテーション (または特定のコンテナーの場合は **io.containers.auto-update/$name**) を使用して自動更新できるようになりました。

Podman がバージョン 4.4 にアップグレードされました。注目すべき変更点の詳細については、アップストリームリリースノート を参照してください。

Jira:RHELPLAN-136607

Aardvark と Netavark がカスタム DNS サーバーの選択をサポートするようになりました。

Aardvark および Netavark ネットワークスタックは、ホスト上のデフォルトの DNS サーバーの代わりに、コンテナーのカスタム DNS サーバーの選択をサポートするようになりました。カスタム DNS サーバーを指定するには、次の 2 つのオプションがあります。

- **containers.conf** 設定ファイルに **dns_servers** フィールドを追加します。
- 新しい **--dns** Podman オプションを使用して、DNS サーバーの IP アドレスを指定します。

--dns オプションは、**container.conf** ファイル内の値をオーバーライドします。

Jira:RHELPLAN-138024

Skopeo は、sigstore キーペアの生成をサポートするようになりました。

skopeogenerate-sigstore-key コマンドを使用して、sigstore 公開キー/秘密キーのペアを生成できます。詳細については、**skopeo-generate-sigstore-key** man ページを参照してください。

Jira:RHELPLAN-151481

ツールボックスが利用可能になりました。

toolbox ユーティリティーを使用すると、トラブルシューティングツールをシステムに直接インストールしなくても、コンテナー化されたコマンドライン環境を使用できます。Toolbox は、Podman および OCI のその他の標準コンテナーテクノロジーを基盤として構築されています。詳細については、toolbx を参照してください。

Jira:RHELPLAN-150266

コンテナーイメージに 2 桁のタグが付けられるようになりました。

RHEL 9.0 および RHEL 9.1 では、コンテナーイメージには 3 桁のタグがありました。RHEL 9.2 以降、コンテナーイメージには 2 桁のタグが付けられるようになりました。

Jira:RHELPLAN-147982

イメージに署名するための複数の信頼できる GPG キーの機能が利用可能です。

/etc/containers/policy.json ファイルは、信頼できるキーを含むファイルのリストを受け入れる新しい **keyPaths** フィールドをサポートします。このため、Red Hat の一般公開キーとベータ GPG キーで署名されたコンテナーイメージがデフォルト設定で受け入れられるようになりました。

以下に例を示します。

```
"registry.redhat.io": [
        {
            "type": "signedBy",
            "keyType": "GPGKeys",
            "keyPaths": ["/etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release", "/etc/pki/rpm-gpg/RPM-GPG-
KEY-redhat-beta"]
        }
]
```

Jira:RHELPLAN-129327

Podman が実行前フックをサポートするようになりました。

/usr/libexec/podman/pre-exec-hooks ディレクトリーと **/etc/containers/pre-exec-hooks** ディレクトリーにある root 所有のプラグインスクリプトは、コンテナー操作の詳細な制御、特に無許可のアクションのブロックを定義します。

/etc/containers/podman_preexec_hooks.txt ファイルは管理者が作成する必要があり、空でもかまいません。**/etc/containers/podman_preexec_hooks.txt** が存在しない場合、プラグインスクリプトは実行されません。すべてのプラグインスクリプトがゼロ値を返す場合、**podman** コマンドが実行されます。それ以外の場合、**podman** コマンドは継承された終了コードで終了します。

Red Hat では、スクリプトを正しい順序で実行するために、**DDD-plugin_name.lang** (例: **010-check-group.py**) という命名規則を使用することを推奨しています。プラグインスクリプトは作成時点で有効であることに注意してください。プラグインスクリプトの前に作成されたコンテナーは影響を受けません。

Bugzilla:2119200

sigstore 署名が利用可能になりました。

Podman 4.2 以降では、コンテナーイメージ署名の sigstore 形式を使用できます。sigstore 署名はコンテナーイメージと共にコンテナーレジストリーに格納されるため、イメージ署名を格納するために別の署名サーバーを用意する必要はありません。

Jira:RHELPLAN-74672

Toolbox は RHEL 9 コンテナーを作成できます。

以前は、Toolbox ユーティリティーは RHEL UBI 8 イメージのみをサポートしていました。このリリースでは、Toolbox は RHEL UBI 9 もサポートするようになりました。その結果、RHEL 8 または 9 に基づいて Toolbox コンテナーを作成できます。

次のコマンドは、ホストシステムと同じ RHEL リリースに基づいて RHEL コンテナーを作成します。

```
$ toolbox create
```

あるいは、特定の RHEL リリースを使用してコンテナーを作成することもできます。たとえば、RHEL 9.2 に基づいてコンテナーを作成するには、次のコマンドを使用します。

```
$ toolbox create --distro rhel --release 9.2
```

Bugzilla:2163752

新しいパッケージ: passt

この更新では、**passt** パッケージが追加され、コンテナーで **passt** ルートレスネットワークバックエンドを使用できるようになります。

現在、Podman による非特権ネットワーキングのデフォルトとして使用されている **Slirp** 接続と比較すると、**pasta** は次の拡張機能を提供します。

- スループットの向上、および近隣探索プロトコル (NDP) および DHCPv6 のサポートを含む IPv6 のサポートの向上
- IPv6 で TCP および UDP ポートのポート転送を設定する機能

pasta を使用して Podman コンテナーに接続するには、**--network pasta** コマンドラインオプションを使用します。

Bugzilla:2209419

第5章 外部カーネルパラメーターへの重要な変更

この章では、システム管理者向けに、Red Hat Enterprise Linux 9.2 で配布されるカーネルの重要な変更点の概要を提供します。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

nomodeset

このカーネルパラメーターを使用すると、カーネルモード設定を無効にすることができます。DRM ドライバーは、表示モードの変更や高速レンダリングを実行しません。システムフレームバッファーがファームウェアまたはブートローダーによって設定されている場合、システムフレームバッファーのみが使用可能になります。
nomodeset は、フォールバックとして、またはテストとデバッグに役立ちます。

printk.console_no_auto_verbose

このカーネルパラメーターを使用すると、oops、パニック、または lockdep で検出された問題 (ロックデバッグがオンの場合のみ) でのコンソールログレベルの上昇を無効にすることができます。シリアルコンソールでボーレートが低いセットアップを除き、より多くのデバッグ情報を提供するには、このパラメーターを **0** に設定します。

- 形式: **<bool>**
- デフォルトは **0** (**auto_verbose** が有効)

rcupdate.rcu_exp_cpu_stall_timeout=[KNL]

このカーネルパラメーターを使用すると、RCU CPU ストールの緊急警告メッセージのタイムアウトを設定できます。値はミリ秒単位で、最大許容値は 21000 ミリ秒です。
この値は、アーキテクチャーのタイマーティック解像度に合わせて調整されることに注意してください。これをゼロに設定すると、**rcupdate.rcu_cpu_stall_timeout** の値が使用されます (秒からミリ秒への変換後)。

rcupdate.rcu_task_stall_info=[KNL]

このパラメーターを使用すると、RCU タスクストール情報メッセージの初期タイムアウトを数秒で設定できます。これは、10 分間待つのに十分な忍耐力がない人に問題の兆候を示すものです。情報メッセージは、特定の猶予期間中、ストール警告メッセージの前のみに出力されます。ゼロ以下の値を指定すると無効になります。

- デフォルトは **10** 秒です。
- 値の変更は、次の猶予期間が始まるまで有効になりません。

rcupdate.rcu_task_stall_info_mult=[KNL]

このパラメーターは、特定の RCU タスク猶予期間における連続する RCU タスクストール情報メッセージ間の時間間隔の乗数です。この値は 1 〜 10 に固定されます。
デフォルト値は 3 です。そのため、最初の情報メッセージは猶予期間に入って 10 秒で出力され、2 番目は 40 秒で、3 番目は 160 秒で出力され、600 秒でのストール警告によって 640 秒で 4 番目のメッセージが出力されなくなります。

smp.csd_lock_timeout=[KNL]

このパラメーターを使用すると、**smp_call_function()** とその仲間が CPU が CSD ロックを解放するまで待機する時間をミリ秒単位で指定できます。これは、長時間にわたって割り込みを無効にする CPU に関連するバグを診断するときに役立ちます。

- デフォルトは **5,000** ミリ秒です。
- 値をゼロに設定すると、この機能が無効になります。
- この機能は、**csdlock_debugkernel** パラメーターを使用してより効率的に無効にすることができます。

srcutree.big_cpu_lim=[KNL]

このパラメーターを使用すると、**srcu_struct** 構造体が **srcu_node** 配列を即座に割り当てるように、大規模システムを設定する CPU の数を指定できます。

- デフォルトは **128** です。
- **srcutree.convert_to_big** の下位 4 ビットが **3** に等しい場合のみ、有効になります (ブート時に決定)。

srcutree.convert_to_big=[KNL]

このパラメーターを使用すると、SRCU ツリー **srcu_struct** 構造体を大きな形式、つまり **rcu_node** ツリーに変換する条件を指定できます。

- 0: まったくなし。
- 1: **init_srcu_struct()** 時。
- 2: **rcutorture** で決定されたとき。
- 3: 起動時に決定 (デフォルト)。
- 0x1X: 競合が多い場合は、上記に加えます。
  いずれの場合も、**srcu_node** ツリーのサイズは、コンパイル時の **CONFIG_NR_CPUS** ではなく、実際の実行時の CPU 数 (**nr_cpu_ids**) に基づいて設定されます。

srcutree.srcu_max_nodelay=[KNL]

このパラメーターを使用すると、SRCU 猶予期間ワーカースレッドが遅延ゼロで再スケジュールされる、1 秒あたりの遅延なしインスタンスの数を指定できます。この制限を超えると、ワーカースレッドは 1 時間のスリープ遅延を伴って再スケジュールされます。

srcutree.srcu_max_nodelay_phase=[KNL]

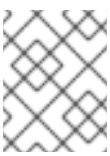
このパラメーターを使用すると、猶予期間ごとのフェーズ、リーダーの非スリープポーリングの数を指定できます。この制限を超えると、猶予期間ワーカースレッドは、猶予期間フェーズの間、リーダーの各再スキャンの間に 1 ジフイのスリープ遅延を設けて再スケジュールされます。

srcutree.srcu_retry_check_delay=[KNL]

このパラメーターを使用すると、リーダーの各非スリープポーリング間の非スリープ遅延のマイクロ秒数を指定できます。

srcutree.small_contention_lim=[KNL]

このパラメーターを使用すると、**srcu_struct** 構造体の大きな形式への変換を開始する前に許容される 1 ジフィーあたりの更新側競合イベントの数を指定できます。



注記

競合ベースの変換を行うには、**srcutree.convert_to_big** の値に 0x10 ビットが設定されている必要があります。

更新されたカーネルパラメーター

crashkernel=size[KMG][@offset[KMG]]

[KNL] **kexec** を使用すると、Linux はパニック時にクラッシュカーネルに切り替えることができます。このパラメーターは、そのカーネルイメージの物理メモリー領域 [offset, offset + size] を予約します。**@offset** を省略すると、適切なオフセットが自動的に選択されます。
[KNL, X86-64, ARM64] 最初に 4G 未満の領域を選択し、**@offset** が指定されていない場合は 4G を超える予約領域にフォールバックします。

詳細については、**Documentation/admin-guide/kdump/kdump.rst** を参照してください。

crashkernel=size[KMG],low

- [KNL, X86-64, ARM64] このパラメーターを使用すると、2 番目のカーネルに 4G 未満の低範囲を指定できます。**crashkernel=X,high** が渡されると、ある程度の低メモリーが必要になります。たとえば、**swiotlb** には少なくとも 64M+32K の低メモリーが必要です。また、32 ビットデバイスの DMA バッファーが不足しないようにするために十分な追加の低メモリーも必要です。カーネルは、4G 未満のデフォルトサイズのメモリーを自動的に割り当てようとします。デフォルトサイズはプラットフォームによって異なります。
  - x86: max(swiotlb_size_or_default() + 8MiB, 256MiB)
  - arm64: 128MiB
    0: 低割り当てを無効にします。

    crashkernel=X,high が使用されない場合、または予約されたメモリーが 4G 未満の場合、このパラメーターは無視されます。
- [KNL, ARM64] このパラメーターを使用すると、クラッシュダンプカーネルの DMA ゾーンの下限範囲を指定できます。
  crashkernel=X,high が使用されていない場合、このパラメーターは無視されます。

deferred_probe_timeout=[KNL]

このパラメーターを使用すると、遅延プローブのタイムアウトを秒単位で設定して、依存関係のプローブの待機を放棄できます。オプトインされている特定の依存関係 (サブシステムまたはドライバー) のみは無視されます。
タイムアウトを **0** にすると、initcall の終了時にタイムアウトになります。タイムアウトが経過していない場合、ドライバーの登録が成功するたびにオプションが再起動されます。このオプションは、再試行後に遅延プローブリストに残っているデバイスもダンプします。

driver_async_probe=[KNL]

このパラメーターを使用すると、非同期的にプローブされるドライバー名のリストを作成できます。* (アスタリスク) はすべてのドライバー名に一致します。

- * が指定されている場合、リストされている残りのドライバー名は * に一致しないものになります。
  形式: **<driver_name1>,<driver_name2>…**

hugetlb_cma=[HW,CMA]

このパラメーターを使用すると、巨大なヒュージページの割り当てに使用される CMA 領域のサイズを指定できます。または、ノード形式を使用して、ノードごとの CMA エリアのサイズを指定します。
形式: **nn[KMGTPE] or (node format) <node>:nn[KMGTPE][,<node>:nn[KMGTPE]]**

指定されたサイズの CMA 領域を予約し、CMA アロケーターを使用して巨大なヒュージページを割り当てます。有効にすると、起動時の巨大な hugepage の割り当てがスキップされます。

hugepages=[HW]

このパラメーターを使用すると、起動時に割り当てる HugeTLB ページの数を指定できます。

- hugepagesz に続く場合は、割り当てる hugepagesz のページ数を指定します。
- これがコマンドラインの最初の HugeTLB パラメーターである場合、デフォルトの巨大ページサイズに割り当てるページ数を指定します。
- ノード形式を使用する場合、ノードごとに割り当てるページ数を指定できます。**Documentation/admin-guide/mm/hugetlbpage.rst** も参照してください。

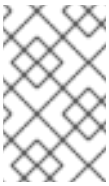
  形式: **<integer> or (node format) <node>:<integer>[,<node>:<integer>]**

hugetlb_free_vmemmap=[KNL]

このパラメーターを使用するには、**CONFIG_HUGETLB_PAGE_OPTIMIZE_VMEMMAP** を有効にする必要があります。hugetlb のヘビーユーザーがさらにメモリーを解放できるようにします (2MB hugetlb ページごとに 7 * PAGE_SIZE)。

- 形式: **{ [oO][Nn]/Y/y/1 | [oO][Ff]/N/n/0 (default) }**
- [oO][Nn]/Y/y/1: 機能を有効にする
- [oO][Ff]/N/n/0: 機能を無効にする **CONFIG_HUGETLB_PAGE_OPTIMIZE_VMEMMAP_DEFAULT_ON=y** で構築されており、

  デフォルトは **on** です。



注記

このパラメーターは、**memory_hotplug.memmap_on_memory** と互換性がありません。両方のパラメーターが有効な場合、**hugetlb_free_vmemmap** が **memory_hotplug.memmap_on_memory** より優先されます。

ivrs_ioapic=[HW,X86-64]

このパラメーターは、IVRS ACPI テーブルで提供される IOAPIC-ID <-> DEVICE-ID マッピングをオーバーライドします。
デフォルトでは、PCI セグメントは **0** であり、省略できます。以下に例を示します。

- IOAPIC-ID 10 進数 10 を PCI デバイス 00:14.0 にマッピングするには、パラメーターを次のように記述します。

```
ivrs_ioapic[10]=00:14.0
```

- IOAPIC-ID 10 進数 10 を PCI セグメント 0x1 および PCI デバイス 00:14.0 にマッピングするには、パラメーターを次のように記述します。

```
ivrs_ioapic[10]=0001:00:14.0
```

ivrs_hpet=[HW,X86-64]

このパラメーターは、IVRS ACPI テーブルで提供される HPET-ID <-> DEVICE-ID マッピングをオーバーライドします。
デフォルトでは、PCI セグメントは **0** であり、省略できます。以下に例を示します。

- HPET-ID 10 進数 0 を PCI デバイス 00:14.0 にマッピングするには、パラメーターを次のように記述します。

```
ivrs_hpet[0]=00:14.0
```

- HPET-ID 10 進数 10 を PCI セグメント 0x1 および PCI デバイス 00:14.0 にマッピングするには、パラメーターを次のように記述します。

```
ivrs_ioapic[10]=0001:00:14.0
```

ivrs_acpihid=[HW,X86-64]

このパラメーターは、IVRS ACPI テーブルで提供される ACPI-HID:UID <-> DEVICE-ID マッピングをオーバーライドします。
たとえば、**UART-HID:UID AMD0020:0** を PCI セグメント **0x1** および PCI デバイス ID **00:14.5** にマッピングするには、パラメーターを次のように記述します。

```
ivrs_acpihid[0001:00:14.5]=AMD0020:0
```

デフォルトでは、PCI セグメントは **0** であり、省略できます。たとえば、PCI デバイス **00:14.5** の場合、パラメーターを次のように記述します。

```
ivrs_acpihid[00:14.5]=AMD0020:0
```

kvm.eager_page_split=[KVM,X86]

このパラメーターを使用すると、KVM がダーティーロギング中にすべての巨大ページを積極的に分割しようとするかどうかを制御できます。
積極的なページ分割により、巨大なページを遅延分割するために必要となる書き込み保護エラーと MMU ロック競合が排除され、vCPU 実行の中断が軽減されます。書き込みをほとんど実行しない VM ワークロード、または VM メモリーの小さな領域のみに書き込みを行う VM ワークロードでは、Eager Page Splitting を無効にして、巨大なページを引き続き読み取りに使用できるようにすると、メリットが得られる場合があります。

積極的なページ分割の動作は、**KVM_DIRTY_LOG_INITIALLY_SET** が有効か無効かによって異なります。

- 無効にすると、その mems スロットでダーティーロギングが有効になったときに、mems スロット内のすべての巨大ページが積極的に分割されます。

- 有効にすると、**KVM_CLEAR_DIRTY** ioctl 中に、クリアされるページのみに、積極的なページ分割が実行されます。
  積極的なページ分割は、**kvm.tdp_mmu=Y** の場合のみ、サポートされます。

  デフォルトは **Y** (オン) です。

kvm-arm.mode=[KVM,ARM]

このパラメーターを使用すると、KVM/arm64 の動作モードの 1 つを選択できます。

- none: KVM を強制的に無効にします。

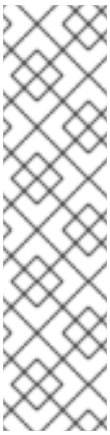
- nvhe: 標準の nVHE ベースのモード。保護されたゲストはサポートされません。
- protected: 状態がホストからプライベートに保たれるゲストをサポートする nVHE ベースのモード。
  ハードウェアのサポートに基づいて、デフォルトは **VHE/nVHE** になります。

nosmep=[X86,PPC64s]

このパラメーターを使用すると、プロセッサーでサポートされている場合でも SMEP (スーパーバイザーモード実行防止) を無効にすることができます。
形式: **pci=option[,option…] [PCI] various_PCI_subsystem_options**

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>]
```



注記

- 最初の形式は、新しいハードウェアが挿入された場合、マザーボードのファームウェアが変更された場合、または他のカーネルパラメーターによる変更により変更される可能性がある PCI バス/デバイス/機能アドレスを指定します。ドメインを指定しないと、ゼロになります。オプションで、複数のデバイスおよび機能アドレスを介したデバイスへのパスをベースアドレスの後に指定できます (これにより、再番号付けの問題に対してより堅牢になります)。
- 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。

- earlydump: カーネルが何かを変更する前に PCI 設定スペースをダンプします。
- off: [X86] PCI バスをプローブしません。
- bios: [X86-32] PCI BIOS の使用を強制します。ハードウェアに直接アクセスしません。マシンに非標準の PCI ホストブリッジがある場合は、これを使用します。
- nobios: [X86-32] PCI BIOS の使用を禁止し、ハードウェアへの直接アクセス方法のみが許可されます。起動時にクラッシュが発生し、BIOS が原因であると思われる場合は、これを使用します。
- conf1: [X86] PCI コンフィギュレーションアクセスメカニズム 1 (IO ポート 0xCF8 のコンフィギュレーションアドレス、IO ポート 0xCFC のデータ、両方とも 32 ビット) の使用を強制します。
- conf2: [X86] PCI コンフィギュレーションアクセスメカニズム 2 の使用を強制します (IO ポート 0xCF8 はこの機能用の 8 ビットポートであり、IO ポート 0xCFA も 8 ビットで、バス番号を設定します。その後、設定スペースにはポート **0xC000-0xCFFF** を通じてアクセスされます。
  - 設定アクセスメカニズムの詳細については、http://wiki.osdev.org/PCI を参照してください。
- noaer: [PCIE] PCIEAER カーネル設定パラメーターが有効な場合、このカーネルブートオプションを使用して、PCIE 高度なエラーレポートの使用を無効にすることができます。

- nodomains: [PCI] 複数の PCI ルートドメイン (ACPI 用語では PCI セグメントとも呼ばれます) のサポートを無効にします。
- nommconf: [X86] PCI 設定での MMCONFIG の使用を無効にします。
- check_enable_amd_mmconf: [X86] AMD ファミリー 10h CPU 上の PCI 設定スペースへの適切に設定された MMIO アクセスをチェックして有効にします。
- nomsi: [MSI] **PCI_MSI** カーネル設定パラメーターが有効な場合、このカーネルブートオプションを使用して、システム全体で MSI 割り込みの使用を無効にすることができます。
- noioapicquirk: [APIC] すべてのブート割り込みの Quirk を無効にします。ブート IRQ を有効にしておくための安全オプション。これは決して必要ではありません。
- ioapicreroute: [APIC] ブート IRQ を無効にできないブリッジのプライマリー IO-APIC へのブート IRQ の再ルーティングを有効にします。これにより、システムが IRQ をマスクする場合のスプリアス IRQ の原因が修正されます。
- noioapicreroute [APIC] ブート IRQ を無効にできないチップセットに接続する IRQ と同等のブート IRQ を使用する回避策を無効にします。ioapicreroute の逆。
- biosirq: [X86-32] PCI BIOS 呼び出しを使用して割り込みルーティングテーブルを取得します。これらの呼び出しはいくつかのマシンでバグがあることが知られており、使用するとマシンがハングしますが、他のコンピューターではこれが割り込みルーティングテーブルを取得する唯一の方法です。カーネルが IRQ を割り当てられない場合、またはマザーボード上のセカンダリー PCI バスを検出できない場合は、このオプションを試してください。
- rom: [X86] 拡張 ROM にアドレス空間を割り当てます。特定のデバイスは ROM と他のリソース間でアドレスデコーダを共有するため、使用には注意してください。
- norom: [X86] BIOS にアドレス範囲が割り当てられていない拡張 ROM にはアドレス空間を割り当てないでください。
- nobar: [X86] BIOS によって割り当てられていない BAR にアドレス空間を割り当てないでください。
- irqmask=0xMMMM: [X86] PCI デバイスへの自動割り当てを許可する IRQ のビットマスクを設定します。この方法で、カーネルに ISA カードの IRQ を除外させることができます。
- pirqaddr=0xAAAAA: [X86] PIRQ テーブル (通常 BIOS によって生成される) の物理アドレスが **F0000h ~ 100000h の** 範囲外の場合に指定します。
- lastbus=N: [X86] バス #N を介してすべてのバスをスキャンします。カーネルがセカンダリーバスを見つけられず、どのバスであるかをカーネルに明示的に伝えたい場合に便利です。
- assign-busses: [X86] すべての PCI バス番号を常に自分で割り当て、ファームウェアが行ったものを上書きします。
- usepirqmask: [X86] BIOS $PIR テーブルに保存されている可能な IRQ マスクを尊重します。これは、BIOS が壊れている一部のシステム、特に一部の HP Pavilion N5400 および Omnibook XE3 ノートブックで必要です。ACPI IRQ ルーティングが有効な場合、これは効果がありません。
- noacpi: [X86] IRQ ルーティングまたは PCI スキャンに ACPI を使用しないでください。

- use_crs: [X86] ACPI からの PCI ホストブリッジウィンドウ情報を使用します。2008 年以降の BIOS では、これはデフォルトで有効になっています。これを使用する必要がある場合は、バグを報告してください。

- nocrs: [X86] ACPI からの PCI ホストブリッジウィンドウを無視します。これを使用する必要がある場合は、バグを報告してください。

- use_e820: [X86] E820 予約を使用して、PCI ホストブリッジウィンドウの一部を除外します。これは、ホストブリッジ _CRS メソッドにおける BIOS の欠陥に対する回避策です。これを使用する必要がある場合は、バグを **linux-pci@vger.kernel.org** に報告してください。

- no_e820: [X86] PCI ホストブリッジウィンドウの E820 予約を無視します。これは最新のハードウェアのデフォルトです。これを使用する必要がある場合は、バグを **linux-pci@vger.kernel.org** に報告してください。

- routeirq: すべての PCI デバイスに対して IRQ ルーティングを実行します。これは通常、**pci_enable_device()** で行われるため、このオプションは、それを呼び出さない壊れたドライバーに対する一時的な回避策です。

- Skip_isa_align: [X86] IO 開始アドレスを調整しないため、より多くの PCI カードを処理できます。

- oearly: [X86] 初期のタイプ 1 スキャンを実行しません。これは、一部のデバイスの設定スペースが読み取られるときにマシンがチェックする壊れたボードで役立つ可能性があります。ただし、さまざまな回避策が無効になっており、一部の IOMMU ドライバーは機能しません。

- bfsort: PCI デバイスを幅優先順にソートします。この並べ替えは、古い (⇐ 2.4) カーネルに対応するデバイスの順番となります。

- nobfsort: PCI デバイスを幅優先順にソートしません。

- pcie_bus_tune_off: PCIe MPS (最大ペイロードサイズ) チューニングを無効にし、BIOS で設定された MPS のデフォルトを使用します。

- pcie_bus_safe: すべてのデバイスの MPS をルートコンプレックス以下のすべてのデバイスでサポートされる最大値に設定します。

- pcie_bus_perf: デバイスの MPS をその親バスに基づいて最大許容 MPS に設定します。また、最高のパフォーマンスを得るために、MRRS (最大読み取り要求サイズ) をサポートされている最大の値 (デバイスまたはバスがサポートできる MPS を超えない値) に設定します。

- pcie_bus_peer2peer: すべてのデバイスの MPS を 128B に設定します。これはすべてのデバイスがサポートすることが保証されています。この設定により、デバイスの任意のペア間でピアツーピア DMA が可能になりますが、パフォーマンスは低下する可能性があります。これにより、ホットアドされたデバイスが動作することも保証されます。

- cbiosize=nn[KMG]: CardBus ブリッジの IO ウィンドウ用に予約される固定量のバススペース。デフォルト値は **256 バイト** です。

- cbmemsize=nn[KMG]: CardBus ブリッジのメモリーウィンドウ用に予約される固定量のバススペース。デフォルト値は **64 メガバイト** です。

- resource_alignment=

  - 形式: **[<order of align>@]<pci_dev>[; ...]**

  - アライメント、およびアライメントされたメモリーリソースを再割り当てするデバイス

を指定します。デバイスの指定方法は上記で説明しました。**<order of align>** が指定されていない場合は、**PAGE_SIZE** がアライメントとして使用されます。リソースウィンドウを拡張する必要がある場合は、PCI-PCI ブリッジを指定できます。デバイスの複数のインスタンスのアライメントを指定するには、PCI ベンダー、デバイス、サブベンダー、およびサブデバイスを指定できます (4096 バイトアライメントの場合は、**12@pci:8086:9c22:103c:198f** など)。

- ecrc=: PCIe ECRC (トランザクション層のエンドツーエンド CRC チェック) を有効/無効にします。
  - bios: BIOS/ファームウェア設定を使用します。これはデフォルトになります。
  - off: ECRC をオフにします。
  - on: ECRC をオンにします。
- hpiosize=nn[KMG]: ホットプラグブリッジの IO ウィンドウ用に予約される固定量のバススペース。デフォルトサイズは **256 バイト** です。
- hpmmiosize=nn[KMG]: ホットプラグブリッジの MMIO ウィンドウ用に予約される固定量のバススペース。デフォルトサイズは **2 メガバイト** です。
- hpmmioprefsize=nn[KMG]: 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固定量のバススペース。デフォルトサイズは **2 メガバイト** です。
- hpmemsize=nn[KMG]: ホットプラグブリッジの MMIO および MMIO_PREF ウィンドウ用に予約される固定量のバススペース。デフォルトサイズは **2 メガバイト** です。
- hpbussize=nn: ホットプラグブリッジの下のバス用に予約される追加バス番号の最小数。デフォルトは **1** です。
- realloc=: BIOS による割り当てが小さすぎてすべての子デバイスに必要なリソースを収容できない場合に、PCI ブリッジリソースの再割り当てを有効または無効にします。
  - off: リアロックをオフにします。
  - on: リアロックをオンにします。
- realloc: realloc=on と同じです。
- noari: PCIe ARI を使用しません。
- noats: [PCIE, Intel-IOMMU, AMD-IOMMU] PCIe ATS (および IOMMU デバイス IOTLB) を使用しません。
- pcie_scan_all: 考えられるすべての PCIe デバイスをスキャンします。それ以外の場合は、PCIe ダウンストリームポートの下にある 1 つのデバイスのみが検索されます。
- big_root_window: AMD CPU 上の PCIe ルートコンプレックスに大きな 64 ビットメモリーウィンドウを追加してみます。一部の GFX ハードウェアは、すべての VRAM にアクセスできるように BAR のサイズを変更できます。ウィンドウの追加には若干の危険が伴います (報告されていないデバイスと競合する可能性があります)。そのため、これによりカーネルがテイントされます。
- disable_acs_redir=<pci_dev>[; ...]: 1 つ以上の PCI デバイスを (上で指定した形式で) セミコロンで区切って指定します。指定された各デバイスでは、PCI ACS リダイレクト機能が強制的にオフになり、アップストリームを強制することなくブリッジを介したデバイス間の P2P

トラフィックが許可されます。注記: これにより、デバイス間の分離が解除され、より多くのデバイスが IOMMU グループに追加される可能性があります。

- force_floating: [S390] フローティング割り込みの使用を強制します。
- nomio: [S390] MIO 命令を使用しません。
- norid: S390 RID フィールドを無視し、PCI 機能ごとに 1 つの PCI ドメインの使用を強制します。

rcupdate.rcu_cpu_stall_timeout=[KNL]

RCU CPU ストール警告メッセージのタイムアウトを設定します。値は秒単位で、最大許容値は 300 秒です。

rcupdate.rcu_task_stall_timeout=[KNL]

このパラメーターを使用すると、RCU タスクストール警告メッセージのタイムアウトを数秒で設定できます。ゼロ以下の値を指定すると無効になります。
デフォルトは **10** 分です。

値の変更は、次の猶予期間が始まるまで有効になりません。

retbleed=[X86]

このパラメーターを使用すると、RETBleed (リターン命令による任意の投機的コード実行) 脆弱性の緩和策を制御できます。
AMD ベースの UNRET および IBPB の緩和策だけでは、兄弟スレッドが他の兄弟スレッドの予測に影響を与えることを防ぐことはできません。そのため、STIBP はそれをサポートするプロセッサーで使用され、STIBP をサポートしないプロセッサーでは SMT を緩和します。

- オフ - 緩和策なし
- auto - 緩和策を自動的に選択します。
- auto,nosmt - 緩和策を自動的に選択し、完全な緩和策に必要な場合は SMT を無効にします (STIBP のない Zen1 以前のみ)。
- ibpb - AMD では、基本ブロック境界での短い推測ウィンドウも緩和します。安全で最高のパフォーマンスの衝撃。STIBP が存在する場合は、それも有効になります。Intel には適していません。
- ibpb,nosmt - 上記の **ibpb** と似ていますが、STIBP が使用できない場合は SMT を無効にします。これは、STIBP を持たないシステムの代替手段です。
- unret - トレーニングされていないリターンサンクを強制的に有効にします。AMD f15h-f17h ベースのシステムのみで有効です。
- unret,nosmt - unret と似ていますが、STIBP が利用できない場合は SMT を無効にします。これは、STIBP を持たないシステムの代替手段です。
  auto を選択すると、CPU に応じて実行時に緩和策が選択されます。

  このオプションを指定しないことは、**retbleed=auto** と同等です。

swiotlb=[ARM,IA-64,PPC,MIPS,X86]

形式: **{ <int> [,<int>] | force | noforce }**

- <int> - I/O TLB スラブの数。

- <int> - コンマの後の 2 番目の整数。独自のロックを持つ **swiotlb** エリアの数。2 の累乗に切り上げられます。
- force - カーネルによって自動的に使用されない場合でも、バウンスバッファーの使用を強制します。
- noforce - バウンスバッファーを使用しません (デバッグ用)。

新しい sysctl パラメーター

kernel.nmi_wd_lpm_factor (PPC only)

この係数は、LPM 中に NMI ウォッチドッグタイムアウトを計算するときに **watchdog_thresh** に追加されるパーセンテージを表します。ソフトロックアップのタイムアウトは影響を受けません。この係数を使用して、NMI ウォッチドッグタイムアウトに適用します (**nmi_watchdog** が 1 に設定されている場合のみ)。

- 値 **0** は変化がないことを意味します。
- デフォルトは **200** です。これは、NMI ウォッチドッグが 30 **秒** に設定されていることを意味します (**watchdog_thresh** が 10 に等しいことに基づく)。

net.core.txrehash

このパラメーターを使用すると、**SO_TXREHASH** オプションが **SOCK_TXREHASH_DEFAULT** に設定されている (つまり、**setsockopt** によってオーバーライドされていない) 場合に、リスニングソケットでのデフォルトのハッシュ再考動作を制御できます。

- **1** (デフォルト) に設定すると、リスニングソケットでハッシュの再考が実行されます。
- **0** に設定すると、ハッシュの再考は実行されません。

net.sctp.reconf_enable - BOOLEAN

この拡張機能を使用すると、RFC6525 で規定されている Stream Reconfiguration 機能の拡張機能を有効または無効にすることができます。この拡張機能はストリームをリセットする機能を提供し、**Outcoming/Incoming SSN Reset**、**SSN/TSN Reset**、および **Add Outcoming/Incoming Streams** のパラメーターを含みます。

- 1: 拡張機能を有効にします。
- 0: 拡張機能を無効にします。
- デフォルトは **0** です。

net.sctp.intl_enable - BOOLEAN

この拡張機能を使用すると、RFC8260 で規定されている User Message Interleaving 機能の拡張機能を有効または無効にすることができます。この拡張機能により、異なるストリームで送信されるユーザーメッセージのインターリーブが可能になります。この機能を有効にすると、ピアでもサポートされている場合、I-DATA チャンクが DATA チャンクを置き換えてユーザーメッセージを伝送します。この機能を使用するには、このオプションを **1** に設定し、ソケットオプション **SCTP_FRAGMENT_INTERLEAVE** を **2** に、**SCTP_INTERLEAVING_SUPPORTED** を **1** に設定する必要があることに注意してください。

- 1: 拡張機能を有効にします。
- 0: 拡張機能を無効にします。

- デフォルトは **0** です。

net.sctp.ecn_enable - BOOLEAN

この拡張機能を使用すると、SCTP による明示的輻輳通知 (ECN) の使用を制御できます。TCP と同様、ECN は、SCTP 接続の両端がサポートを示している場合のみ、使用されます。この機能は、サポートしているルーターがパケットをドロップする前に輻輳を通知できるようにすることで、輻輳による損失を回避するのに役立ちます。

- 1: ECN を有効にします。
- 0: ECN を無効にします。
- デフォルトは **1** です。

vm.hugetlb_optimize_vmemmap

このノブは、**memory_hotplug.memmap_on_memory** カーネルパラメーターが設定されている場合、または **構造体ページ** (**include/linux/mm_types.h** で定義されている構造体) のサイズが 2 のべき乗ではない場合には使用できません (異常なシステム設定によりこの結果が生じる可能性があります)。
各 HugeTLB ページに関連付けられた **vmemmap** ページを最適化する機能を有効 (1 に設定) または無効 (0 に設定) できます。

- 有効にすると、バディアロケーターからの後続の HugeTLB ページ割り当ての **vmemmap** ページが最適化されます (2MB HugeTLB ページあたり 7 ページ、1GB HugeTLB ページあたり 4095 ページ)。一方、すでに割り当てられている HugeTLB ページは最適化されません。これらの最適化された HugeTLB ページが HugeTLB プールからバディアロケーターに解放されると、その範囲を表す **vmemmap** ページを再度マッピングし、以前に破棄された **vmemmap** ページを再度再配置する必要があります。
- HugeTLB ページが即興で割り当てられるユースケースの場合 (たとえば、**nr_hugepages** で HugeTLB ページを明示的に割り当てるのではなく、**nr_overcommit_hugepages** のみを設定すると、オーバーコミットされた HugeTLB ページは即興で割り当てられます)、HugeTLB プールから取得するのではなく、HugeTLB プールとバディアロケーター間の HugeTLB ページの割り当てまたは解放によるオーバーヘッドの増加 (以前よりも約 2 倍遅い) に対するメモリーの節約という利点を比較検討する必要があります。注意すべきもう 1 つの動作は、システムのメモリー負荷が高い場合、**vmemmap** ページの割り当てが失敗する可能性があるため、ユーザーが HugeTLB ページを HugeTLB プールからバディアロケータに解放できなくなる可能性があることです。システムがこの状況に遭遇する場合は、後で再試行する必要があります。
- 無効にすると、バディアロケータからの後続の HugeTLB ページの割り当ての **vmemmap** ページは最適化されなくなり、バディアロケータからの割り当て時の余分なオーバーヘッドがなくなりますが、すでに最適化された HugeTLB ページは影響を受けません。最適化された HugeTLB ページがないことを確認したい場合は、まず **nr_hugepages** を **0** に設定してから、これを無効にします。**nr_hugepages** に **0** を書き込むと、**使用中** の HugeTLB ページが余剰ページになることに注意してください。したがって、これらの余剰ページは、使用されなくなるまで最適化され続けます。システム内に最適化されたページがなくなる前に、余剰ページが解放されるまで待つ必要があります。

net.core.rps_default_mask

新しく作成されたネットワークデバイスで使用されるデフォルトの RPS CPU マスク。空のマスクは、デフォルトで RPS が無効になっていることを意味します。

変更された sysctl パラメーター

kernel.numa_balancing

kernel.numa_balancing

このパラメーターを使用すると、自動ページフォールトベースの NUMA メモリーバランシングを有効、無効、および設定できます。メモリーは、頻繁にアクセスするノードに自動的に移動されます。設定する値は、次の論理和をとった結果になります。

```
= ================================
0 NUMA_BALANCING_DISABLED
1 NUMA_BALANCING_NORMAL
2 NUMA_BALANCING_MEMORY_TIERING
= ================================
```

または、**NUMA_BLANCING_NORMAL** を使用して、異なる NUMA ノード間でのページ配置を最適化し、リモートアクセスを削減します。NUMA マシンでは、CPU がリモートメモリーにアクセスするかどうかについてパフォーマンスのペナルティーがあります。この機能が有効になっている場合、カーネルは定期的にページのマッピングを解除し、後でページフォールトをトラップすることによって、どのタスクスレッドがメモリーにアクセスしているかをサンプリングします。ページフォールトの発生時に、アクセス中のデータをローカルメモリーノードに移行する必要があるかどうかが決定されます。

または、**NUMA_BLANCING_MEMORY_TIERING** を使用して、異なるタイプのメモリー (異なる NUMA ノードとして表される) 間でページ配置を最適化し、高速メモリーにホットページを配置します。これもアンマッピングとページフォールトに基づいて実装されています。

net.ipv6.route.max_size

ガベージコレクションがキャッシュされたルートエントリーを管理するため、これは ipv6 では非推奨になりました。

net.sctp.sctp_wmem

この調整パラメーターは以前、何の効果もないことが文書化されていました。現在は、最初の値 (**min**) のみが使用され、**default** と **max** は無視されます。

- min: SCTP ソケットで使用できる送信バッファーの最小サイズ。適度なメモリー負荷下でも、各 SCTP ソケットに対して保証されます (関連付けは保証されません)。
- デフォルトは **4K** です。

第6章 デバイスドライバー

6.1. 新しいドライバー

- ACPI ビデオドライバー (**video**)、64 ビット ARM アーキテクチャーのみ
- CXL メモリーエンドポイントデバイスおよびメモリー拡張用スイッチ用の CXL ドライバー (**cxl_mem**)
- GNSS 受信機コア (**gnss**)
- GPIO シミュレーターモジュール (**gpio-sim**)、64 ビット ARM アーキテクチャーのみ
- VirtIO GPIO ドライバー (**gpio-virtio**)、64 ビット ARM アーキテクチャーのみ
- NVIDIA Tegra HTE (ハードウェアタイムスタンプエンジン) ドライバー (**hte-tegra194**)、64 ビット ARM アーキテクチャーのみ
- LPI2C バス用の I2C アダプタードライバー (**i2c-imx-lpi2c**)、64 ビット ARM アーキテクチャーのみ
- Virtio i2c バスドライバー (**i2c-virtio**)、64 ビット ARM アーキテクチャーのみ
- 入力サブシステム (**uinput**) のユーザーレベルドライバーのサポート (64 ビット ARM アーキテクチャーのみ)
- nvme ホストまたはターゲットドライバーで使用できる共通機能を実装するモジュール (**nvme-common**)
- AMD PMC ドライバー (**amd-pmc**)、AMD および Intel 64 ビットアーキテクチャーのみ
- Nvidia sn2201 プラットフォームドライバー (**nvsw-sn2201**)、AMD および Intel 64 ビットアーキテクチャーのみ
- シリアルマルチインスタンス化擬似デバイスドライバー (**serial-multi-instantiate**)、AMD および Intel 64 ビットアーキテクチャーのみ
- Micro Crystal RV8803 RTC ドライバー (**rtc-rv8803**)、64 ビット ARM アーキテクチャーおよび AMD および Intel 64 ビットアーキテクチャーのみ
- NVIDIA Tegra QSPI コントローラードライバー (**spi-tegra210-quad**)、64 ビット ARM アーキテクチャーのみ
- Cypress CCGx Type-C コントローラー用の UCSI ドライバー (**ucsi_ccg**)、64 ビット ARM アーキテクチャーのみ
- Confidential Computing EFI 秘密領域アクセス (**efi_secret**)、AMD および Intel 64 ビットアーキテクチャーのみ
- TDX ゲストドライバー (**tdx-guest**)、AMD および Intel 64 ビットアーキテクチャーのみ
- HPE ウォッチドッグドライバー (**hpwdt**)、64 ビット ARM アーキテクチャーのみ
- POWER アーキテクチャープラットフォームウォッチドッグドライバー (**pseries-wdt**)、IBM Power Systems のみ、リトルエンディアン

ネットワークドライバー

- VXLAN カプセル化トラフィック (**vxlan**) 用ドライバー
- Marvell OcteonTX2 RVU Admin Function Driver (**rvu_af**)、64 ビット ARM アーキテクチャーのみ
- Marvell RVU NIC 物理関数ドライバー (**rvu_nicpf**)、64 ビット ARM アーキテクチャーのみ
- Marvell RVU NIC PTP ドライバー (**otx2_ptp**)、64 ビット ARM アーキテクチャーのみ
- Marvell RVU NIC 仮想機能ドライバー (**rvu_nicvf**)、64 ビット ARM アーキテクチャーのみ
- NVIDIA Tegra MGBE ドライバー (**dwmac-tegra**)、64 ビット ARM アーキテクチャーのみ
- シリアルライン CAN インターフェイス (**slcan**)、64 ビット ARM アーキテクチャーのみ
- Solarflare Siena ネットワークドライバー (**sfc-siena**)、IBM Power Systems、Little Endian、AMD および Intel 64 ビットアーキテクチャーのみ

グラフィックドライバーとその他のドライバー

- DRM Buddy Allocator (**drm_buddy**)、64 ビット ARM アーキテクチャーおよび IBM Power Systems のみ、リトルエンディアン
- DRM ディスプレイアダプターヘルパー (**drm_display_helper**)、64 ビット ARM アーキテクチャー、IBM Power Systems、リトルエンディアン、AMD および Intel 64 ビットアーキテクチャーのみ
- DRM DisplayPort AUX バス (**drm_dp_aux_bus**)、64 ビット ARM アーキテクチャーのみ
- Tegra 製品用の Host1x ドライバー (**host1x**)、64 ビット ARM アーキテクチャーのみ
- NVIDIA Tegra DRM ドライバー (**tegra-drm**)、64 ビット ARM アーキテクチャーのみ
- Intel® GVT-g for KVM (**kvmgt**)、AMD および Intel 64 ビットアーキテクチャーのみ
- HP® iLO/iLO2 管理プロセッサー (**hpilo**)、64 ビット ARM アーキテクチャーのみ
- GSC デバイス用インテル® 補助ドライバー (**mei-gsc**) (AMD およびインテル 64 ビットアーキテクチャーのみ)

6.2. 更新されたドライバー

ストレージドライバーの更新

- Microchip Smart Family Controller (**smartpqi**) のドライバーがバージョン 2.1.20-035 に更新されました (64 ビット ARM アーキテクチャー、IBM Power Systems、リトルエンディアン、AMD および Intel 64 ビットアーキテクチャーのみ)。
- Emulex LightPulse ファイバーチャネル SCSI ドライバー (**lpfc**) がバージョン 14.2.0.8 に更新されました (64 ビット ARM アーキテクチャー、IBM Power Systems、リトルエンディアン、AMD および Intel 64 ビットアーキテクチャーのみ)。
- MPI3 Storage Controller Device Driver (**mpi3mr**) がバージョン 8.2.0.3.0 に更新されました。
- CSI デバッグアダプタードライバー (**scsi_debug**) がバージョン 0191 に更新されました。

- LSI MPT Fusion SAS 3.0 デバイスドライバー (**mpt3sas**) がバージョン 43.100.00.00 に更新されました (64 ビット ARM アーキテクチャー、IBM Power Systems、リトルエンディアン、AMD および Intel 64 ビットアーキテクチャーのみ)。

第7章 利用可能な BPF 機能

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

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

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

表7.1 システム設定とその他のオプション

オプション	値
unprivileged_bpf_disabled	2 (特権ユーザーに限定された bpf() syscall、管理者は変更可能)
JIT コンパイラー	1 (有効)
JIT コンパイラーの強化	1 (権限のないユーザーに対して有効)
JIT コンパイラー kallsyms エクスポート	1 (ルートで有効)
非特権ユーザーの JIT のメモリー制限	264241152
CONFIG_BPF	y
CONFIG_BPF_SYSCALL	y
CONFIG_HAVE_EBPF_JIT	y
CONFIG_BPF_JIT	y
CONFIG_BPF_JIT_ALWAYS_ON	y
CONFIG_DEBUG_INFO_BTF	y
CONFIG_DEBUG_INFO_BTF_MODULES	y
CONFIG_CGROUPS	y
CONFIG_CGROUP_BPF	y
CONFIG_CGROUP_NET_CLASSID	y
CONFIG_SOCK_CGROUP_DATA	y

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

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

表7.2 利用可能なプログラムの種類とサポートされているヘルパー

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

表7.3 利用可能なマップの種類

マップの種類	Available
ハッシュ	はい
array	はい
prog_array	はい
perf_event_array	はい
percpu_hash	はい
percpu_array	はい
stack_trace	はい
cgroup_array	はい
lru_hash	はい
lru_percpu_hash	はい
lpm_trie	はい
array_of_maps	はい
hash_of_maps	はい
devmap	はい
sockmap	はい
cpumap	はい
xskmap	はい
sockhash	はい
cgroup_storage	はい
reuseport_sockarray	はい
percpu_cgroup_storage	はい
queue	はい

マップの種類	Available
stack	はい
sk_storage	はい
devmap_hash	はい
struct_ops	はい
ringbuf	はい
inode_storage	はい
task_storage	はい
bloom_filter	はい

第8章 バグ修正

このパートでは、Red Hat Enterprise Linux 9.2 で修正された、ユーザーに重大な影響を与えるバグについて説明します。

8.1. インストーラーおよびイメージの作成

インストーラーは、マルチパスまたは DDF RAID デバイスを使用したカスタムパーティショニングで正しい合計ディスク容量を表示するようになりました。

以前は、マルチパスまたは DDF RAID デバイスを備えたシステム上のインストーラーでカスタムパーティション分割が選択されている場合、合計ディスク容量が正しく報告されず、メンバーディスクデバイスがパーティション分割に使用できるものとしてリストされていました。

この更新により、インストーラーのカスタムパーティション分割により、合計ディスク容量の正しい値が報告され、DDF RAID またはマルチパスデバイス全体の使用のみが許可されるようになりました。

Bugzilla:2052938

インストーラーは、設定オプションを yum リポジトリーファイルに正しく追加するようになりました。

以前は、追加のインストールリポジトリーからパッケージを含めたり除外したりするときに、インストーラは yum リポジトリーファイルに設定オプションを正しく追加しませんでした。この更新により、yum リポジトリーファイルが正しく作成されるようになりました。その結果、**repo** キックスタートコマンドで **--excludepkgs=** または **--includepkgs=** オプションを使用すると、インストール中に指定されたパッケージが期待どおりに除外または組み込まれるようになりました。

Bugzilla:2158210

filename DHCP オプションを使用しても、インストール用の kickstart ファイルのダウンロードがブロックされなくなりました。

以前は、NFS サーバーからキックスタートファイルを取得するためのパスを構築するときに、インストーラーは **filename** DHCP オプションを考慮しませんでした。その結果、インストーラーはキックスタートファイルをダウンロードせず、インストールプロセスをブロックしていました。この更新により、**filename** DHCP オプションはキックスタートファイルへのパスを正しく構築します。その結果、キックスタートファイルが適切にダウンロードされ、インストールプロセスが正しく開始されます。

Bugzilla:1991843

インストーラーはカスタムパーティショニング中に新しい GPT ディスクレイアウトを作成するようになりました。

以前は、カーネルコマンドラインで **inst.gpt** が指定されている場合、インストーラはディスクレイアウトを GPT に変更せず、ユーザーはカスタムパーティショニングスポーク上の MBR ディスクレイアウトを持つディスクからすべてのパーティションを削除していました。その結果、MBR ディスクレイアウトがディスク上に残りました。

この更新により、カーネルコマンドラインで **inst.gpt** が指定されている場合、インストーラーはディスク上に新しい GPT ディスクレイアウトを作成し、カスタムパーティションスポーク上のディスクからすべてのパーティションが削除されます。

Bugzilla:2127100

インストーラーは、カスタムパーティショニング中にすべての PPC PreP Boot または BIOS Boot パーティションをリストするようになりました。

以前は、カスタムパーティショニング中に複数の **PPC PreP Boot** または **BIOS Boot** パーティションを追加すると、Custom Partitioning 画面には関連するタイプのパーティションが 1 つだけ表示されていました。その結果、カスタムパーティショニング画面には意図したパーティショニングレイアウトの実際の状態が反映されず、パーティショニングプロセスが困難かつ不透明になっていました。

この更新により、カスタムパーティショニング画面のパーティションリストにすべての **PPC PreP Boot** または **BIOS Boot** パーティションが正しく表示されるようになりました。その結果、ユーザーは意図したパーティショニングレイアウトをよりよく理解し、管理できるようになりました。

Bugzilla:2093793

Anaconda が、FIPS 要件の LUKS パスフレーズを検証するようになりました。

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

この更新により、パスフレーズの最小長を検証して強制するようにインストーラーが改善されました。その結果、インストーラーは、LUKS パスフレーズが FIPS モードで使用するには短すぎるかどうかを通知し、予期しない終了を阻止します。

Bugzilla:2163497

8.2. サブスクリプションの管理

サブスクリプションマネージャーが Red Hat コンテンツの登録と取得を拒否しなくなりました

以前は、RHEL 9 でコンテナー検出ロジックが改善されたため、OpenShift Container Platform (OCP) で実行される場合、**subscription-manager** はコンテナーモードで動作していました。その結果、システムは提供されたサブスクリプション認証情報を使用できず、Red Hat コンテンツを取得できませんでした。

この更新により、OCP で実行されている **subscription-manager** がシステム (つまり、実行中の Pod) をコンテナーとして検出しないように、コンテナー検出ロジックが修正されました。その結果、提供されたサブスクリプション認証情報を使用したり、独自の認証情報を使用して登録したりして、OpenShift コンテナーから Red Hat コンテンツをフェッチできるようになりました。

Bugzilla:2108549

subscription-manager は端末に不要なテキストを保持しなくなりました。

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

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

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

Bugzilla:2136694

8.3. ソフトウェア管理

fapolicyd サービスの再起動を伴うトランザクション中に RPM がハングしなくなりました

以前は、**fapolicyd** サービスの再起動を引き起こすパッケージ (**systemd** など) を更新しようとすると、**fapolicyd** プラグインが **fapolicyd** デーモンとの通信に失敗したため、RPM トランザクションが応答を停止していました。

この更新により、**fapolicyd** プラグインは **fapolicyd** デーモンと正しく通信できるようになりました。その結果、**fapolicyd** サービスの再起動を伴うトランザクション中に RPM がハングしなくなりました。

Bugzilla:2111251

パッケージグループまたは環境に対して DNF アップグレードトランザクションを元に戻せるようになりました。

以前は、パッケージグループまたは環境のアップグレードトランザクションを元に戻そうとすると、**dnfhistory rollback** コマンドが失敗していました。

今回の更新により、この問題は修正され、パッケージグループまたは環境の DNF アップグレードトランザクションを元に戻せるようになりました。

Bugzilla:2122626

アップグレードによってアーキテクチャーが変更されるパッケージに対してセキュリティー DNF アップグレードが可能になりました。

RHBA-2022:8295 とともに導入された BZ#2108969 のパッチにより、セキュリティーフィルターを使用した DNF アップグレードにより、アップグレードを通じてアーキテクチャーが noarch からまたは **noarch** に変更されたパッケージがスキップされるというリグレッションが発生しました。したがって、これらのパッケージのセキュリティーアップグレードが不足していると、システムが脆弱な状態になる可能性があります。

今回の更新により、この問題は修正され、セキュリティー DNF アップグレードで、アーキテクチャーを noarch から変更するパッケージ、または **noarch** に変更するパッケージがスキップされなくなりました。

Bugzilla:2124480

RPM パッケージのビルドまたは再ビルド時に、3 文字の名前を持つ Qt メッセージ QM ファイルが、パッケージ化されるようになりました。

以前は、**find-lang.sh** スクリプトは、名前が 3 文字で設定される Qt メッセージ QM ファイル (**.qm**) を見つけることができませんでした。したがって、これらのファイルは RPM パッケージに追加されませんでした。

今回の更新により、この問題は修正され、RPM のビルドまたはリビルド時に 3 文字の Qt メッセージ QM ファイルをパッケージ化できるようになりました。

Bugzilla:2144005

8.4. シェルおよびコマンドラインツール

ReaR は、IBM Z アーキテクチャー上で除外された DASD を正しく処理します。

以前の IBM Z アーキテクチャーでは、ReaR は、ユーザーが保存されたレイアウトから除外し、コンテンツを復元するつもりがなかった DASD を含め、接続されているすべてのダイレクトアクセスストレージデバイス (DASD) をリカバリープロセス中に再フォーマットしていました。その結果、保存されたレイアウトから一部の DASD を除外すると、システムの回復中にそれらのデータが失われます。この更新により、ReaR は、(zIPL ブートローダーを使用して) ReaR レスキューシステムがブートされたデバイ

スを含む、システムリカバリー中に除外された DASD をフォーマットしなくなりました。ReaR が DASD を再フォーマットする前に、DASD フォーマットスクリプトを確認するように求められます。これにより、除外された DASD 上のデータはシステム回復後も確実に残ります。

Bugzilla:2172589

ReaR は非 LVM XFS ファイルシステムの復元に失敗しなくなりました。

以前は、ReaR を使用して特定の設定とディスクマッピングを使用して非 LVM XFS ファイルシステムを復元すると、ReaR は指定された設定ではなくデフォルト設定でファイルシステムを作成していました。たとえば、ファイルシステムの **sunit** パラメーターと **swidth** パラメーターが 0 以外の値に設定されており、ディスクマッピングを備えた ReaR を使用してファイルシステムを復元した場合、ファイルシステムは、指定された値を無視してデフォルトの **sunit** パラメーターと **swidth** パラメーターで作成されます。その結果、ReaR は特定の XFS オプションを使用してファイルシステムをマウントするときに失敗しました。この更新により、ReaR は指定された設定でファイルシステムを正しく復元します。

Bugzilla:2160748

wsmancli は HTTP 401 Unauthorized ステータスを正しく処理します。

Web サービス管理プロトコルを使用してシステムを管理するための **wsmancli** ユーティリティーは、RFC 2616 への準拠を強化するために認証を処理するようになりました。

以前は、認証が必要なサービスに接続すると、**wsmancli** コマンドは、認証情報が不完全であるなどの理由で、HTTP 401 Unauthorized 応答を受信した直後、エラーメッセージ **Authentication failed, please retry** を返していました。続行するには、認証情報の一部をすでに提供している場合でも、**wsmancli** はユーザー名とパスワードの両方を提供するように求めます。

この更新により、**wsmancli** は、以前に提供されていなかった認証情報のみを必要とするようになりました。その結果、最初の認証試行ではエラーメッセージが表示されません。エラーメッセージは、完全な認証情報を入力して認証が失敗した場合のみ、表示されます。

Bugzilla:2127416

8.5. セキュリティー

USBGuard は、RuleFile が定義されていない場合でもルールを保存します。

以前は、USBGuard の **RuleFolder** 設定ディレクティブが設定されていても、**RuleFile** が設定されていない場合、ルールセットを変更できませんでした。今回の更新により、RuleFile が設定されていなくても、RuleFolder が設定されている場合は、ルールセットを変更できるようになりました。その結果、USBGuard の永続ポリシーを変更して、新しく追加されたルールを永続的に保存できます。

Bugzilla:2155910

python-sqlalchemy が 1.4.45 にリベースされました。

python-sqlalchemy パッケージがバージョン 1.4.45 にリベースされ、バージョン 1.4.37 に比べて多くのバグ修正が行われています。最も注目すべき点は、このバージョンにはキャッシュキー生成における重大なメモリーバグの修正が含まれているということです。

Bugzilla:2152649

crypto-policies はバインドの NSEC3DSA を無効にするようになりました。

以前は、システム全体の暗号化ポリシーは、バインド設定の NSEC3DSA アルゴリズムを制御していませんでした。その結果、現在のセキュリティー要件を満たしていない NSEC3DSA は、DNS サーバーで無効になりませんでした。この更新により、すべての暗号化ポリシーはデフォルトでバインド設定の

NSEC3DSA を無効にします。

Bugzilla:2152635

SECLEVEL=3 の OpenSSL が PSK 暗号スイートで動作するようになりました。

以前は、事前共有キー (PSK) 暗号スイートは、PFS (Perfect Forward Secrecy) キー交換方式を実行すると認識されませんでした。その結果、**ECDHE-PSK** および **DHE-PSK** 暗号スイートは、たとえば、システム全体の暗号化ポリシーが **FUTURE** に設定されている場合、**SECLEVEL=3** に設定された OpenSSL では機能しませんでした。**openssl** パッケージの新しいバージョンでは、この問題が解決されています。

Bugzilla:2060044

Clevis は、crypttab でコメントアウトされたデバイスを正しくスキップするようになりました。

以前は、Clevis が **crypttab** ファイル内のコメントアウトされたデバイスのロックを解除しようとしたため、デバイスが有効でない場合でも **clevis-luks-askpass** サービスが実行されてしまいました。これにより、不必要なサービスが実行され、トラブルシューティングが困難になりました。

この修正により、Clevis はコメントアウトされたデバイスを無視します。現在、無効なデバイスがコメントアウトされている場合、Clevis はそのデバイスのロックを解除しようとせず、**clevis-luks-askpass** が適切に終了します。これにより、トラブルシューティングが容易になり、不必要なサービスの実行が削減されます。

Bugzilla:2159728

Clevis は pwmake に過剰なエントロピーを要求しなくなりました。

以前は、Clevis が **pwmake** を使用してデータを **LUKS** メタデータに保存するためのパスワードを作成するときに、**pwmake** パスワード生成ユーティリティーによって不要な警告が表示され、Clevis が使用するエントロピーが低下していました。この更新により、Clevis は **pwmake** に提供されるエントロピービットが 256 に制限され、不要な警告が排除され、正しい量のエントロピーが使用されます。

Bugzilla:2159735

USBGuard で紛らわしい警告が表示されなくなりました。

以前は、親プロセスが最初の子プロセスよりも早く終了すると、USBGuard で競合状態が発生することがありました。その結果、**systemd** は、誤って識別された親 PID (PPID) を持つプロセスが存在すると報告しました。この更新により、親プロセスは最初の子プロセスが作業モードで終了するまで待機します。その結果、**systemd** は、そのような警告を報告しなくなります。

Bugzilla:2042345

OOM キラーが usbguard を途中で終了させなくなりました。

以前は、**usbguard.service** ファイルには、**systemd** サービスの **OOMScoreAdjust** オプションの定義が含まれていませんでした。その結果、システムのリソースが不足すると、**usbguard-daemon** プロセスが他の権限のないプロセスよりも前に終了する可能性がありました。今回の更新により、**usbguard.service** ファイルに **OOMScoreAdjust** 設定が含まれるようになり、メモリー不足 (OOM) キラーが **usbguard-daemon** プロセスを途中で終了するのを防ぎます。

Bugzilla:2097419

logrotate はログローテーションで Rsyslog に誤って通知しなくなりました。

以前は、**logrotate** スクリプトで引数の順序が誤って設定されており、構文エラーが発生していました。これにより、**logrotate** がログローテーション中に Rsyslog に正しく信号を送信できなくなりました。

今回の更新により、**logrotate** の引数の順序が修正され、**POSIXLY_CORRECT** 環境変数が設定されている場合でも、**logrotate** はログローテーション後に Rsyslog に正しく通知するようになりました。

Bugzilla:2124488

imklog は失われたオブジェクトに対して free() を呼び出さなくなりました。

以前は、**imklog** モジュールは、すでに解放されたオブジェクトに対して **free()** 関数を呼び出していました。その結果、**imklog** によってセグメンテーション違反が発生する可能性があります。今回の更新により、オブジェクトが 2 回解放されることがなくなりました。

Bugzilla:2157659

fagenrules --load が正常に動作するようになりました

以前は、**fapolicyd** サービスはシグナルのハングアップ (SIGHUP) を正しく処理しませんでした。そのため、**fapolicyd** は SIGHUP を受信した後に終了し、**fagenrules --load** コマンドが正しく機能しませんでした。この更新では、その問題が修正されました。その結果、**fagenrules --load** が正常に機能し、ルール更新時に **fapolicyd** を手動で再起動する必要がなくなりました。

Bugzilla:2070655

スキャンと修復は SCAP 監査ルールを正しく無視します。

以前は、監査キー (**-k** または **-F** キー) を使用せずに定義された監査監視ルールでは、次の問題が発生しました。

- ルールの他の部分が正しい場合でも、ルールは非準拠としてマークされていました。
- Bash 修復により監視ルールのパスと権限が修正されましたが、監査キーが正しく追加されませんでした。
- 修復によって欠落したキーが修正されず、**fixed** 値の代わりに **error** が返されることがありました。

これは次のルールに影響を与えました。

- **audit_rules_login_events**
- **audit_rules_login_events_faillock**
- **audit_rules_login_events_lastlog**
- **audit_rules_login_events_tallylog**
- **audit_rules_usergroup_modification**
- **audit_rules_usergroup_modification_group**
- **audit_rules_usergroup_modification_gshadow**
- **audit_rules_usergroup_modification_opasswd**
- **audit_rules_usergroup_modification_passwd**

- **audit_rules_usergroup_modification_shadow**
- **audit_rules_time_watch_localtime**
- **audit_rules_mac_modification**
- **audit_rules_networkconfig_modification**
- **audit_rules_sysadmin_actions**
- **audit_rules_session_events**
- **audit_rules_sudoers**
- **audit_rules_sudoers_d**

この更新により、Audit キーがチェック、Bash および Ansible の修復から削除されました。その結果、チェックおよび修復中にキーフィールドによって引き起こされる不一致は発生しなくなり、監査人はこれらのキーを任意に選択して監査ログの検索を容易にすることができます。

Bugzilla:2120978

Keylime は、複数の IMA で測定されたファイルにアクセスするシステムの認証に失敗しなくなりました。

以前は、Keylime エージェントを実行するシステムが、Integrity Measurement Architecture (IMA) によって測定された複数のファイルに連続してアクセスした場合、Keylime ベリファイアは IMA ログの追加を誤って処理していました。その結果、実行中のハッシュが正しいプラットフォーム設定レジスター (PCR) の状態と一致せず、システムは認証に失敗しました。この更新により問題が修正され、複数の測定ファイルに迅速にアクセスするシステムが認証に失敗することがなくなりました。

Bugzilla:2138167

Keylime ポリシー生成スクリプトでセグメンテーションフォールトとコアダンプが発生しなくなりました。

create_mb_refstate スクリプトは、Keylime で測定されたブート認証用のポリシーを生成します。以前は、**create_mb_refstate** が **DevicePath** フィールドのデータ長を誤って計算していました。その結果、スクリプトは誤って計算された長さを使用して無効なメモリーにアクセスしようとし、セグメンテーションフォールトとコアダンプが発生しました。

アドバイザリー RHBA-2023:0309 で公開されたこの更新プログラムは、測定されたブートイベントログを処理する際のセグメンテーションフォールトを防止します。その結果、測定されたブートポリシーを生成できます。

Bugzilla:2140670

TPM 証明書によって Keylime レジストラがクラッシュすることはなくなりました。

以前は、Keylime TPM 証明書ストア内の一部の証明書は不正な形式の x509 証明書であり、Keylime レジストラーのクラッシュを引き起こしていました。この更新により問題が修正され、不正な形式の証明書が原因で Keylime レジストラーがクラッシュすることはなくなりました。

Bugzilla:2142009

8.6. ネットワーク

NetworkManager は、新しい DHCP リースを取得する前の再適用中に IP アドレスを保持するようになりました。

以前は、接続設定を変更してから **nmcli device reapply** コマンドを使用した後、NetworkManager は DHCP リースを保存しませんでした。その結果、IP アドレスが一時的に削除されました。この修正により、NetworkManager は DHCP リースを保存し、リースの有効期限が切れるか、クライアントが新しいリースを要求するまでそれを使用します。その結果、**nmcli device reapply** コマンドが DHCP クライアントを再起動しても、IP アドレスは一時的に削除されません。

Bugzilla:2117352

firewalld サービスは、直接ルールを使用する場合のみ、ipset 非推奨の警告をトリガーするようになりました。

以前は、**firewalld** サービスは、不要な場合に非推奨の **ipset** カーネルモジュールを使用していました。その結果、RHEL はモジュールの非推奨警告をログに記録しました。これは、**firewalld** の **ipset** 機能が非推奨ではないため、誤解を招く可能性があります。この更新により、**firewalld** は、ユーザーが **--direct** オプションを指定して明示的に **ipset** を使用した場合にのみ、非推奨の **ipset** モジュールを使用し、警告をログに記録します。

Bugzilla:2122678

再起動後に HNV インターフェイスにオプションが表示されるようになりました

以前は、**nmcli** ユーティリティーは NetworkManager API を使用してハイブリッドネットワーク仮想化 (HNV) ボンドを作成していました。その結果、再起動後、HNV ボンドはプライマリーポート設定を失いました。この修正により、**nmcli** は **hcnmgr** を使用してプライマリーポートのボンディングオプションを設定するようになりました。**hcnmgr** ユーティリティーは、ハイブリッドネットワークのシングルルート入出力仮想化 (SR-IOV) を使用したライブパーティションの移行をサポートします。その結果、再起動後に HNV ボンドインターフェイスに **active slave/primary_reselect** オプションが表示されます。

Bugzilla:2125152

8.7. カーネル

セキュアブートで有効になっている FADump は正しく動作します。

以前は、セキュアブート環境でファームウェアアシストダンプ (FADump) が有効になっており、ブートコンポーネントのいずれかが割り当てられたメモリー領域を超えた場合、システムの再起動により GRUB メモリー不足 (OOM) 状態が発生していました。この更新では、**kexec-tools** に修正が加えられ、セキュアブートと FADump が正しく連携できるようになりました。

Bugzilla:2139000

8.8. ブートローダー

grubby は引数を新しいカーネルに正しく渡すようになりました。

grubby ツールを使用して新しいカーネルを追加し、引数を指定しないか、引数を空白のままにすると、**grubby** は新しいカーネルに引数を渡さず、**root** は設定されません。**--args** および **--copy-default** オプションを使用すると、新しい引数がデフォルトの引数に追加されます。

Bugzilla:2127453

PReP のサイズが 4 MiB または 8 MiB でない場合でも、RHEL のインストールが成功するようになりました。

以前は、4 kiB セクターを使用するディスク上の PowerPC Reference Platform (PReP) パーティションのサイズが 4 MiB または 8 MiB とは異なる場合、RHEL インストーラーがブートローダーをインストールできませんでした。その結果、ディスクに RHEL をインストールできませんでした。

このリリースでは、この問題は修正されています。そのため、インストーラーは期待どおりに RHEL をディスクにインストールできます。

Bugzilla:2026579

8.9. ファイルシステムおよびストレージ

セクターサイズが 512 バイトの LUKSv2 デバイスを作成するインストーラー。

以前は、ディスクに 4096 バイトの物理セクターがある場合、RHEL インストーラーは 4096 バイトのセクターを持つ LUKSv2 デバイスを作成しました。この更新により、インストーラーはセクターサイズが 512 バイトの LUKSv2 デバイスを作成するようになり、LVM 物理ボリュームが暗号化されている場合でも、1 つの LVM ボリュームグループで一緒に使用されるさまざまな物理セクターサイズとのディスク互換性が向上しました。

Bugzilla:2103800

supported_speeds sysfs 属性は正しい速度値を報告します。

以前は、**qla2xxx** ドライバーの定義が間違っていたため、HBA の **supported_speeds sysfs** 属性は、予想される 64 Gb/s の速度ではなく、20 Gb/s の速度を報告していました。その結果、HBA が 64 Gb/s リンク速度をサポートしている場合、**supported_speeds sysfs** 値が正しくなくなり、報告された速度値に影響を及ぼしました。

この更新により、HBA の **supported_speeds sysfs** 属性は、正しい速度値 (16 Gb/s、32 Gb/s、および 64 Gb/s) を報告するようになりました。**cat /sys/class/fc_host/host*/supported_speeds** コマンドを実行すると、速度値を表示できます。

Bugzilla:2069758

lpfc ドライバーは、D_ID ポートスワップ中に有効な状態にあります。

以前は、SAN ブートホストが NetApp ギブバックオペレーションを発行した後、LVM ハングタスクの警告とストールした I/O を引き起こしていました。この問題は、ファイバーチャネル **D_ID** ポートスワップが原因で、DM-Multipath 環境で代替パスが利用可能であった場合でも発生しました。競合状態の結果として、**D_ID** ポートスワップにより **lpfc** ドライバーで不整合な状態が発生し、I/O の発行が妨げられました。

今回の修正により、**lpfc** ドライバーは、**D_ID** ポートスワップが発生すると、必ず有効な状態になるようになりました。その結果、ファイバーチャネル **D_ID** ポートスワップにより、ハングした I/O が発生しなくなりました。

Bugzilla:2173947

8.10. 高可用性およびクラスター

pcs では、変更すべきではないクラスターのプロパティーを変更できなくなりました。

以前は、**pcs** コマンドラインインターフェイスを使用して、変更すべきでないクラスタープロパティーや、変更が有効にならないクラスタープロパティーを変更できました。この修正により、**pcs** では、クラスタープロパティー **cluster-infrastructor**、**cluster-name**、**dc-version**、**have-watchdog**、および **last-lrm-refresh** を変更できなくなりました。

Bugzilla:1620043

pcs は、明示的に設定されていないクラスターのプロパティーを表示するようになりました。

以前は、特定のクラスタープロパティーの値を表示する **pcs** コマンドでは、CIB で明示的に設定されていない値がリストされませんでした。この修正により、クラスタープロパティーが設定されていない場合、**pcs** はプロパティーのデフォルト値を表示します。

Bugzilla:1796827

crm_mon を呼び出すクラスターリソースがシャットダウン時に正常に停止するようになりました。

以前は、Pacemaker のシャットダウン中に **crm_mon** ユーティリティーがゼロ以外の終了ステータスを返していました。**ocf:heartbeat:pqsql** などのモニターアクションで **crm_mon** を呼び出したリソースエージェントが、クラスターのシャットダウン時に誤って失敗を返す可能性がありました。この修正により、クラスターがシャットダウン中であっても **crm_mon** は成功を返すようになりました。**crm_mon** を呼び出すリソースは、クラスターのシャットダウン時に正常に停止するようになりました。

Bugzilla:2133546

OCF リソースエージェントのメタデータアクションが、予期しないフェンシングを引き起こすことなく crm_node を呼び出せるようになりました。

RHEL 8.5 以降、OCF リソースエージェントのメタデータアクションはコントローラーをブロックし、**crm_node** クエリーはコントローラー要求を実行しました。その結果、エージェントのメタデータアクションが **crm_node** を呼び出した場合、アクションがタイムアウトになるまで 30 秒間コントローラーがブロックされました。これにより、他のアクションが失敗し、ノードが隔離される可能性があります。

この修正により、コントローラーはメタデータアクションを非同期で実行するようになりました。OCF リソースエージェントのメタデータアクションは問題なく **crm_node** を呼び出せるようになりました。

Bugzilla:2125344

Pacemaker は、リソースの順序が変更されたときにすぐにリソースの割り当てを再チェックするようになりました。

RHEL 8.7 以降、リソース定義を変更せずに CIB 内のリソースの順序が変更された場合、Pacemaker はリソース割り当てを再チェックしませんでした。設定の並べ替えによりリソースが移動する場合、次の自然な移行 (**cluster-recheck-interval-property** の値まで) まで移動は行われません。これにより、リソースの固定性がリソースに対して設定されていない場合に問題が発生する可能性があります。

この変更により、Pacemaker は、以前の Pacemaker リリースと同様に、CIB 内のリソースの順序が変更されたときにリソース割り当てを再チェックします。クラスターは、必要に応じてこれらの変更に即座に応答するようになりました。

Bugzilla:2125337

単一のリソースと監視操作を有効にしても、リソースグループ内のすべてのリソースの監視操作は有効になりません。

以前は、リソースグループ内のすべてのリソースの管理を解除し、操作を監視した後、そのグループ内のリソースの 1 つをその監視操作とともに管理すると、リソースグループ内のすべてのリソースの監視操作が再び有効になりました。これにより、クラスターの予期しない動作が引き起こされる可能性があります。

この修正により、リソースを管理し、その監視操作を再度有効にすると、そのリソースに対してのみ監視操作が再度有効になり、リソースグループ内の他のリソースに対しては無効になります。

Bugzilla:2092950

8.11. コンパイラーおよび開発ツール

一部の CNAME レコードが無効な場合でも、DNS ルックアップが成功するようになりました。

以前は、**glibc** DNS スタブリゾルバーは、ホスト名ではない所有者名を持つ CNAME レコードを DNS パケットエラーとして処理していました。その結果、DNS パケットエラーにより DNS クエリーが失敗しました。今回の更新により、**glibc** スタブリゾルバーは無効な CNAME レコードをスキップし、対応するエイリアス情報は抽出されなくなりました。したがって、サーバー応答にホスト名ではないドメイン名を含む CNAME チェーンが含まれている場合でも、DNS ルックアップが成功するようになりました。

Bugzilla:2129005

golang が x509 FIPS モードで 4096 ビットキーをサポートするようになりました。

以前は、**golang** は x509 FIPS モードの 4096 ビットキーをサポートしていませんでした。その結果、ユーザーが 4096 ビットのキーを使用すると、プログラムがクラッシュしました。この更新により、**golang** は x509 FIPS モードで 4096 ビットキーをサポートするようになりました。

Bugzilla:2133019

すべてのアーキテクチャーで pip を使用して SciPy をインストールできます。

以前は、**openblas-devel** パッケージには OpenBLAS ライブラリーの pkg-config ファイルが含まれていませんでした。その結果、特定のシナリオでは、OpenBLAS でコンパイル中に **pkgconf** ユーティリティーを使用してコンパイラーとリンカーのフラグを決定することができませんでした。たとえば、これにより、64 ビット IBM Z および IBM Power Systems のリトルエンディアンアーキテクチャー上で **pip install scipy** コマンドが失敗します。

この更新により、サポートされているすべてのアーキテクチャーの **openblas-devel** パッケージに **openblas.pc** ファイルが追加されます。その結果、**pip** パッケージインストーラーを使用して SciPy ライブラリーをインストールできます。

RHEL 9 では、**flexiblas-devel** パッケージに対してアプリケーションを構築し、プロジェクトを FlexiBLAS ラッパーライブラリーにリンクすることが推奨されることに注意してください。

Bugzilla:2115737

TZ データに DST ルールがある場合、glibc の tzset 関数は夏時間変数をゼロ以外の値に設定するようになりました。

以前は、タイムゾーンデータファイル内の最後の DST 移行によって、標準時間オフセットの同時変更が原因でクロックが変更されなかった場合、**glibc** の **tzset** 関数は夏時間変数を 0 に設定していました。したがって、アプリケーションが夏時間変数を使用して DST がアクティブだったかどうかを確認すると、正しい結果が得られず、この情報に基づいて誤ったアクションが実行されます。これを修正するために、**tzset** 関数は、オフセットに関係なく、タイムゾーンデータに DST ルールがある場合、夏時間変数をゼロ以外の値に設定するようになりました。その結果、アプリケーションはオフセットの変更に関係なく、DST ルールの存在を監視するようになりました。

Bugzilla:2155352

OpenJDK RSAPSSSignature 実装では、RSA キーを使用する前に検証するようになりました。

以前は、OpenJDK の RSAPSSSignature 実装は、指定された RSA キーを使用する前に、SunRSASign プロバイダーがその RSA キーを使用できるかどうかを完全にチェックしなかったため、カスタムセキュリティープロバイダーの使用時にエラーが発生していました。このバグは修正され、その結果、RSAPSSSignature 実装は RSA キーを検証し、他のプロバイダーがこれらのキーを処理できない場合に処理できるようになりました。

Bugzilla:2188023

OpenJDK XML 署名プロバイダーが FIPS モードで機能するようになりました。

以前は、OpenJDK XML 署名プロバイダーは FIPS モードで動作できませんでした。FIPS モードのサポートが強化された結果、OpenJDK XML 署名プロバイダーが FIPS モードで有効になりました。

Bugzilla:2186810

FIPS モードの OpenJDK で、特定の PKCS#11 トークンで予期しないエラーが発生しなくなりました。

以前は、一部の PKCS#11 トークンは、OpenJDK による FIPS モードでの使用前に完全に初期化されておらず、予期しないエラーが発生していました。今回のアップグレードにより、これらのエラーは予期され、FIPS サポートコードによって処理されるようになりました。

Bugzilla:2186806

8.12. IDENTITY MANAGEMENT

クライアントシークレットを必要とする外部 IdP への認証が可能になりました。

以前は、SSSD はクライアントシークレットを外部 ID プロバイダー (IdP) に適切に渡しませんでした。その結果、クライアントシークレットを要求するように **ipa idp-add --secret** コマンドで以前に設定した外部 IdP に対する認証が失敗しました。この更新により、SSSD はクライアントシークレットを IdP に渡し、ユーザーは認証できるようになります。

Jira:RHELPLAN-148303

IdM は、Ansible を使用した sudo ルールのホストマスクの設定をサポートするようになりました。

以前は、**ipa sudorule-add-host** コマンドでは、**sudo** ルールで使用されるホストマスクを設定できましたが、このオプションは **ansible-freeipa** パッケージには存在していませんでした。この更新により、**ansible-freeipa hostmask** 変数を使用して、Identity Management (IdM) で定義された特定の **sudo** ルールが適用されるホストマスクのリストを定義できるようになりました。

その結果、Ansible を使用して IdM **sudo** ルールのホストマスクの設定を自動化できるようになりました。

Bugzilla:2127913

db_dir パラメーターでカスタムパスを使用すると、dscreate ユーティリティーが正しく動作するようになる

以前は、カスタムディレクトリーの SELinux ラベルが間違っていたため、カスタムディレクトリーパスを使用するインスタンスは起動に失敗していました。その結果、SELinux はこれらのディレクトリーへのアクセスを拒否し、インスタンスは作成されませんでした。このリリースでは、**dscreate** ユーティリティーはカスタムインスタンスディレクトリーに正しい SELinux ラベルを設定します。

Bugzilla:1924569

Directory Server レプリケーションマネージャーアカウントのパスワード変更が正しく機能するようになりました

以前は、パスワード変更後、Directory Server はレプリケーションアグリーメントのパスワードキャッシュを適切に更新していませんでした。その結果、レプリケーションマネージャーアカウントのパスワードを変更すると、レプリケーションが失敗しました。この更新により、Directory Server はキャッシュを適切に更新し、その結果、レプリケーションが期待どおりに機能するようになりました。

Bugzilla:1956987

IdM クライアントインストーラーは、ldap.conf ファイルで TLS CA 設定を指定しなくなる

以前は、IdM クライアントインストーラーは **ldap.conf** ファイルで TLS CA 設定を指定していました。この更新により、OpenLDAP はデフォルトのトラストストアを使用し、IdM クライアントインストーラーは **ldap.conf** ファイルに TLS CA 設定をセットアップしません。

Bugzilla:2094673

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

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

RHBA-2023:4359 のリリースにより、大文字と小文字を区別する比較は、大文字と小文字を区別しない比較に置き換えられました。その結果、IdM クライアントは、ユーザー名に大文字と小文字が混在しており、IdM でオーバーライドが設定されている場合でも、AD の信頼済みドメインのユーザーを検索できるようになりました。

Jira:SSSD-6096

8.13. グラフィックインフラストラクチャー

Matrox G200e が VGA ディスプレイに出力を表示するようになりました。

以前は、次のシステム設定を使用している場合、ディスプレイにグラフィカル出力が表示されないことがありました。

- Matrox G200e GPU
- VGA コントローラーで接続されたディスプレイ

したがって、この設定で RHEL を使用またはインストールできませんでした。

このリリースでは、この問題は修正されています。そのため、期待どおりに RHEL が起動し、グラフィック出力が表示されます。

Bugzilla:1960467

8.14. WEB コンソール

Web コンソールの NBDE バインディング手順が、ルートファイルシステムを持つボリュームグループで機能するようになる

RHEL 9.2.0 では、ユーザーがルートファイルシステムに Tang キーを追加したかどうかを判断するコードのバグが原因で、LUKS コンテナー上にファイルシステムがまったくない場合に、Web コンソールのバインディングプロセスがクラッシュしていました。**Verify key** ダイアログの **Trust key** ボタンをクリックした後、Web コンソールにエラーメッセージ **TypeError: Qe(...) is undefined** が表示されたため、説明されているシナリオのコマンドラインインターフェイスで必要な手順をすべて実行する必要がありました。

RHBA-2023:4346 アドバイザリーのリリースにより、Web コンソールはルートファイルシステムへの Tang キーの追加を正しく処理できるようになりました。その結果、Web コンソールは、さまざまなシナリオで Network-Bound Disk Encryption (NBDE) を使用した LUKS 暗号化ボリュームの自動ロック解除に必要なバインド手順をすべて完了します。

Bugzilla:2207498

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

nbde_client システムロールが、clevis-luks-askpass のさまざまな名前を正しく処理するようになりました

nbde_client システムロールは、**clevis-luks-askpass systemd** ユニットの名前が異なるシステムを処理できるように更新されました。このロールは、マネージドノード上のさまざまな名前の **clevis-luks-askpass** で正しく動作するようになりました。これには、ブートプロセスの後半でマウントされる LUKS 暗号化ボリュームのロックも解除する必要があります。

Bugzilla:2126959

ha_cluster システムロールログに、暗号化されていないパスワードや機密情報が表示されなくなりました

ha_cluster システムロールは、パスワードなどの機密情報であるパラメーターを受け入れます。以前は、一部のタスクは入力と出力をログに記録していました。その結果、ロールログに暗号化されていないパスワードなどの機密情報が含まれる可能性がありました。

この更新により、タスクは Ansible **no_log: true** ディレクティブを使用するように変更され、タスクの出力はロールログに表示されなくなりました。**ha_cluster** システムロールログに、パスワードなどの機密情報が含まれなくなりました。この更新によりセキュアな情報は保護されますが、ロールログで提供される情報は設定のデバッグ時に使用できる情報が少なくなります。

Bugzilla:2143816

ha_cluster システムロールで SBD を使用し、ブート時に起動しないように設定されたクラスターが正しく動作するようになりました

以前は、ユーザーが **ha_cluster** システムロールで SBD を使用し、ブート時に起動しないようにクラスターを設定した場合、SBD サービスが無効になり、SBD は起動しませんでした。この修正により、クラスターがブート時に起動するように設定されているかどうかに関係なく、クラスターが SBD を使用するように設定されている場合、SBD サービスが常に有効になります。

Bugzilla:2153030

cockpit-session-recording SSSD 設定を修正するための暗黙的ファイルプロバイダーの有効化。

SSSD 暗黙的ファイルプロバイダーが無効になっているため、**cockpit-session-recording** モジュールが無効な System Security Services Daemon (SSSD) 設定を作成しました。この更新により、ファイルプロバイダーが無条件で有効になり、その結果、**cockpit-session-recording** によって作成された SSSD 設定が期待どおりに機能するようになりました。

Bugzilla:2153043

nbde_client_clevis ロールはユーザーにトレースバックを報告しなくなりました。

以前は、**nbde_client_clevis** ロールが例外で失敗することがあり、トレースバックが発生し、**encryption_password** フィールドなどの機密データがユーザーに報告されていました。今回の更新により、ロールは機密データを報告しなくなり、適切なエラーメッセージのみが報告されるようになりました。

Bugzilla:2162782

ha_cluster システムロールを使用した stonith-watchdog-timeout プロパティーの設定が、停止したクラスターでも機能するようになりました

以前は、停止したクラスターで **ha_cluster** システムロールを使用して **stonith-watchdog-timeout** プロパティーを設定すると、プロパティーが以前の値に戻り、ロールが失敗していました。この修正により、**ha_cluster** システムロールを使用した **stonith-watchdog-timeout** プロパティーの設定が正しく機能するようになりました。

Bugzilla:2167528

networking RHEL システムロールで initscripts を使用する場合、ネットワークトラフィックが目的のネットワークインターフェイス経由で送信されるようになりました

以前は、**initscripts** プロバイダーを使用する場合、ネットワーク接続のルーティング設定で、トラフィックが通過する出力デバイスが指定されませんでした。その結果、カーネルはユーザーが意図したものとは異なる出力デバイスを使用する可能性があります。現在、接続用の Playbook でネットワークインターフェイス名が指定されている場合、その名前がルート設定ファイルの出力デバイスとして使用されます。これにより、デバイス上でプロファイルをアクティブ化するときにルート内の出力デバイスを設定する NetworkManager と動作が調整されます。その結果、ユーザーはトラフィックが意図したネットワークインターフェイスを介して確実に送信されるようになります。

Bugzilla:2168735

selinux ロールがポリシーモジュールをべき等性で管理するようになりました。

以前は、**selinux** ロールは毎回既存のモジュールをマネージドノードにコピーし、モジュールがすでに存在する場合でも変更を報告していました。この更新により、**selinux** ロールはモジュールがマネージドノードにインストールされているかどうかを確認し、モジュールがすでにインストールされている場合はコピーしてインストールしようとしません。

Bugzilla:2160152

rhc_auth にアクティベーションキーが含まれている場合、rhc システムロールが登録済みシステムで失敗しなくなる

以前は、**rhc_auth** パラメーターで指定されたアクティベーションキーを使用して登録済みシステムで Playbook ファイルを実行すると、エラーが発生していました。この問題は解決されています。**rhc_auth** パラメーターにアクティベーションキーが提供されていても、すでに登録されているシステムで Playbook ファイルを実行できるようになりました。

Bugzilla:2186218

8.16. 仮想化

ネストされた VM 上のシステム時刻が確実に動作するようになりました。

以前は、ネストされた仮想マシン (VM) 上のシステム時刻がレベル 0 およびレベル 1 のホストから非同期になる場合がありました。これにより、ネストされた VM が応答しなくなったり、予期せず終了したりすることがありました。

この更新により、KVM ホストカーネルコードの時刻処理コードが修正され、上記のエラーの発生が防止されました。

Bugzilla:2140899

memfd メモリーバッキングを使用しているときに IBM Z 上の VM が起動に失敗しなくなりました。

以前は、IBM Z ホスト上で、次のように **memfd** タイプの hugepage メモリーバッキングを使用するように設定されている場合、仮想マシン (VM) が起動できませんでした。

```
<memoryBacking>
  <hugepages/>
  <source type='memfd'/>
</memoryBacking>
```

この更新により、根本的な原因が修正され、影響を受ける VM が正しく起動するようになりました。

Bugzilla:2116496

移行後に VNC が UEFI VM に確実に接続できるようになりました。

以前は、仮想マシン (VM) の移行中にメッセージキューを有効または無効にすると、移行の完了後に仮想ネットワークコンピューティング (VNC) クライアントが VM に接続できませんでした。

この問題は、Open Virtual Machine Firmware (OVMF) を使用する UEFI ベースの VM のみに影響します。

この問題は修正され、移行の完了後、VNC クライアントは確実に UEFI VM に接続できるようになりました。

Jira:RHELPLAN-135600

インストーラーには、VM に RHEL をインストールするために予期されるシステムディスクが表示されます。

以前は、**virtio-scsi** デバイスを使用して VM に RHEL をインストールする場合、 **device-mapper-multipath** のバグにより、これらのデバイスがインストーラーに表示されない可能性がありました。したがって、インストール中に、シリアルセットを持つデバイスとシリアルセットを持たないデバイスがある場合、**multipath** コマンドはすべてのデバイスがシリアルを持つと主張していました。このため、インストーラーは、VM に RHEL をインストールするための予期されたシステムディスクを見つけることができませんでした。

今回の更新により、**multipath** はシリアルのないデバイスを World Wide Identifier (WWID) を持たないものとして正しく設定し、無視します。インストール時に、**multipath** は **multipathd** がマルチパスデバイスのバインドに使用するデバイスのみを要求し、インストーラーは VM に RHEL をインストールするために予期されるシステムディスクを表示します。

Bugzilla:1926147

第9章 テクノロジープレビュー

ここでは、Red Hat Enterprise Linux 9 で利用可能なテクノロジープレビューのリストを提示します。

テクノロジープレビューに対する Red Hat のサポート範囲の詳細は、テクノロジープレビューのサポート範囲 を参照してください。

9.1. インストーラーおよびイメージの作成

NVMe over Fibre Channel デバイスが RHEL インストーラーでテクノロジープレビューとして利用可能になる

NVMe over Fibre Channel デバイスをテクノロジープレビューとして RHEL インストールに追加できるようになりました。RHEL インストーラーでは、インストール先 画面でディスクを追加するときに、NVMe ファブリックデバイスセクションでこれらのデバイスを選択できます。

Bugzilla:2107346

9.2. シェルおよびコマンドラインツール

RHEL 9 でテクノロジープレビューとして利用可能な GIMP

GNU Image Manipulation Program (GIMP) 2.99.8 が、テクノロジープレビューとして RHEL 9 で利用できるようになりました。**gimp** パッケージバージョン 2.99.8 は、改善された一連の改良を含むリリース前のバージョンですが、機能のセットが制限され、安定性の保証は保証されません。公式の GIMP 3 のリリース後すぐに、今回のリリース前のバージョンの更新として RHEL 9 に導入されます。

RHEL 9 では、RPM パッケージとして **gimp** を簡単にインストールできます。

Bugzilla:2047161

9.3. インフラストラクチャーサービス

TuneD 用のソケット API がテクノロジープレビューとして利用可能になる

Unix ドメインソケットを通じて TuneD を制御するためのソケット API がテクノロジープレビューとして利用可能になりました。ソケット API は D-Bus API と 1 対 1 でマッピングされ、D-Bus が利用できない場合に代替通信方法を提供します。ソケット API を使用すると、TuneD デーモンを制御してパフォーマンスを最適化したり、さまざまなチューニングパラメーターの値を変更したりできます。ソケット API はデフォルトでは無効になっていますが、**tuned-main.conf** ファイルで有効にできます。

Bugzilla:2113900

9.4. セキュリティー

gnutls がテクノロジープレビューとして KTLS を使用するようになる

更新された **gnutls** パッケージは、テクノロジープレビューとして、暗号化チャネルでのデータ転送を加速するためにカーネル TLS (KTLS)を使用できます。KTLS を有効にするには、**modprobe** コマンドを使用して **tls.ko** カーネルモジュールを追加し、以下の内容でシステム全体の暗号化ポリシー用の新しい設定ファイル **/etc/crypto-policies/local.d/gnutls-ktls.txt** を作成します。

```
[global]
ktls = true
```

現在のバージョンは、TLS **KeyUpdate** メッセージによるトラフィックキーの更新をサポートしていません。これは、AES-GCM 暗号スイートのセキュリティーに影響を与えることに注意してください。詳細は、RFC 7841 - TLS 1.3 ドキュメントを参照してください。

Bugzilla:2042009

9.5. ネットワーク

WireGuard VPN はテクノロジープレビューとして利用可能になる

Red Hat がサポートしていないテクノロジープレビューとして提供している WireGuard は、Linux カーネルで実行する高パフォーマンスの VPN ソリューションです。最新の暗号を使用し、その他の VPN ソリューションよりも簡単に設定できます。さらに、WireGuard のコードベースが小さくなり、攻撃の影響が減るため、セキュリティーが向上します。

詳細は Setting up a WireGuard VPN を参照してください。

Bugzilla:1613522

KTLS がテクノロジープレビューとして利用可能になる

RHEL は、テクノロジープレビューとして KTLS (Kernel Transport Layer Security) を提供します。KTLS は、AES-GCM 暗号化のカーネルで対称暗号化アルゴリズムまたは複号アルゴリズムを使用して TLS レコードを処理します。KTLS には、この機能を提供するネットワークインターフェイスコントローラー (NIC) に TLS レコード暗号化をオフロードするインターフェイスも含まれています。

Bugzilla:1570255

systemd-resolved サービスがテクノロジープレビューとして利用可能になる

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

systemd-resolved は、サポートされていないテクノロジープレビューであることに注意してください。

Bugzilla:2020529

9.6. カーネル

SGX がテクノロジープレビューとして利用可能

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

Bugzilla:1874182

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

カーネルの Intel データストリーミングアクセラレータードライバー (IDXD) は、現在テクノロジープレビューとして利用できます。これは IntelCPU 統合アクセラレータであり、プロセスアドレス空間 ID (pasid) 送信と共有仮想メモリー (SVM) を備えた共有ワークキューが含まれています。

Bugzilla:2030412

Soft-iWARP ドライバーがテクノロジープレビューとして利用可能になる

Soft-iWARP(siw) は、Linux 用のソフトウェア、インターネットワイドエリア RDMA プロトコル (iWARP)、カーネルドライバーです。soft-iWARP は、TCP/IP ネットワークスタックで iWARP プロトコルスイートを実装します。このプロトコルスイートはソフトウェアで完全に実装されており、特定のリモートダイレクトメモリーアクセス (RDMA) ハードウェアを必要としません。soft-iWARP を使用すると、標準のイーサネットアダプターを備えたシステムが iWARP アダプターまたは他のシステムに接続でき、すでに Soft-iWARP がインストールされている別のシステムに接続できます。

Bugzilla:2023416

SGX がテクノロジープレビューとして利用可能

Software Guard Extensions(SGX) は、ソフトウェアコードおよび公開および修正からのデータを保護する Intel® テクノロジーです。RHEL カーネルは、SGX v1 および v1.5 の機能を部分的に提供します。バージョン 1 では、**Flexible Launch Control** メカニズムを使用するプラットフォームで SGX テクノロジーを使用できるようになります。バージョン 2 では、**Enclave Dynamic Memory Management** (EDMM) が追加されています。主な変更には以下のものがあります。

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

Bugzilla:1660337

rvu_af、rvu_nicpf、および rvu_nicvf がテクノロジープレビューとして利用可能

次のカーネルモジュールは、Marvell OCTEON TX2 インフラストラクチャープロセッサーファミリーのテクノロジープレビューとして利用できます。

- **rvu_nicpf** - Marvell OcteonTX2 NIC 物理機能ドライバー
- **rvu_nicvf** - Marvell OcteonTX2 NIC 仮想機能ドライバー
- **rvu_nicvf** - Marvell OcteonTX2 RVU 管理機能ドライバー

Bugzilla:2040643

9.7. ファイルシステムおよびストレージ

DAX がテクノロジープレビューとして ext4 および XFS で利用可能になる

RHEL 9 では、DAX ファイルシステムがテクノロジープレビューとして提供されています。DAX は、アプリケーションが永続メモリーをそのアドレス空間に直接マップするための手段を提供します。DAX を使用するには、システムに何らかの形式の永続メモリー (通常は 1 つ以上の不揮発性デュアルインラインメモリーモジュール (NVDIMM) の形式) が必要であり、DAX 互換ファイルシステムを NVDIMM 上に作成する必要があります。)。また、ファイルシステムは **dax** マウントオプションでマウントする必要があります。これにより、dax をマウントしたファイルシステムのファイルの **mmap** が、アプリケーションのアドレス空間にストレージを直接マッピングされます。

Bugzilla:1995338

Stratis はテクノロジープレビューとして利用可能です

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

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

Stratis ストレージを管理するには、バックグランドサービス **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。

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

Bugzilla:2041558

NVMe-oF Discovery Service 機能がテクノロジープレビューとして利用可能になる

NVMexpress.org Technical Proposals (TP) 8013 および 8014 で定義されている NVMe-oF Discovery Service の機能は、テクノロジープレビューとして利用できます。これらの機能をプレビューするには、**nvme-cli 2.0** パッケージを使用して、TP-8013 または TP-8014 を実装する NVMe-oF ターゲットデバイスにホストを割り当てます。TP-8013 および TP-8014 の詳細については、https://nvmexpress.org/specations/ Web サイトから NVM Express 2.0 認定 TP を参照してください。

Bugzilla:2021672

NVMe-stas パッケージがテクノロジープレビューとして利用可能になる

Linux の Central Discovery Controller (CDC) クライアントである **nvme-stas** パッケージがテクノロジープレビューとして利用できるようになりました。これは、非同期イベント通知 (AEN)、自動化された NVMe サブシステム接続制御、エラー処理とレポート、および Automatic (**zeroconf**) 手動設定を処理します。

このパッケージは、Storage Appliance Finder (**stafd**)と Storage Appliance Connector (**stacd**)の 2 つのデーモンで構成されています。

Bugzilla:1893841

NVMe TP 8006 インバンド認証がテクノロジープレビューとして利用可能

Non-Volatile Memory Express (NVMe) の実装 NVMe over Fabrics (NVMe-oF) のインバンド認証である TP 8006 は、サポートされていないテクノロジープレビューとして利用できるようになりました。NVMe Technical Proposal 8006 で、この機能強化で提供される NVMe-oF の **DH-HMAC-CHAP** インバンド認証プロトコルが定義されています。

詳細は、**nvme-connect(1)** の man ページの **dhchap-secret** および **dhchap-ctrl-secret** オプションの説明を参照してください。

Bugzilla:2027304

9.8. コンパイラーおよび開発ツール

jmc-core および owasp-java-encoder がテクノロジープレビューとして利用可能

RHEL 9 は、AMD および Intel 64 ビットアーキテクチャー用のテクノロジープレビューとして、**jmc-core** および **owasp-java-encoder** パッケージとともに配布されます。

jmc-core は、Java Development Kit (JDK) Mission Control のコア API を提供するライブラリーです。これには、JDK Flight Recording ファイルの解析および書き込み用のライブラリーや、Java Discovery Protocol (JDP) による Java Virtual Machine (JVM) 検出のライブラリーが含まれます。

owasp-java-encoder パッケージは、Java の高パフォーマンスな低オーバーヘッドコンテキストエンコーダーのコレクションを提供します。

RHEL 9.2 以降、**jmc-core** および **owasp-java-encoder** は CodeReady Linux Builder (CRB) リポジトリーで使用できるため、明示的に有効にする必要があることに注意してください。詳細は、CodeReady Linux Builder 内でコンテンツを有効にして利用する方法 を参照してください。

Bugzilla:1980981

9.9. IDENTITY MANAGEMENT

DNSSEC が IdM でテクノロジープレビューとして利用可能

統合 DNS のある Identity Management (IdM) サーバーは、DNS プロトコルのセキュリティーを強化する DNS に対する拡張セットである DNS Security Extensions (DNSSEC) を実装するようになりました。IdM サーバーでホストされる DNS ゾーンは、DNSSEC を使用して自動的に署名できます。暗号鍵は、自動的に生成およびローテートされます。

DNSSEC で DNS ゾーンを保護する場合は、以下のドキュメントを参照することが推奨されます。

- DNSSEC Operational Practices, Version 2
- Secure Domain Name System (DNS) Deployment Guide
- DNSSEC Key Rollover Timing Considerations

統合 DNS のある IdM サーバーは、DNSSEC を使用して、他の DNS サーバーから取得した DNS 回答を検証することに注意してください。これが、推奨される命名方法に従って設定されていない DNS ゾーンの可用性に影響を与える可能性があります。

Bugzilla:2084180

Identity Management JSON-RPC API がテクノロジープレビューとして利用可能

Identity Management (IdM) では API が利用できます。API を表示するために、IdM は、テクノロジープレビューとして API ブラウザーも提供します。

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

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

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

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。

Bugzilla:2084166

sssd-idp サブパッケージがテクノロジープレビューとして利用可能

SSSD の **sssd-idp** サブパッケージには、Identity Management (IdM) サーバーに対して OAuth2 認証を実行するクライアント側のコンポーネントである **oidc_child** プラグインおよび krb5 **idp** プラグインが含まれます。この機能は、RHEL 9.1 以降の IdM サーバーのみで使用できます。

Bugzilla:2065693

SSSD の内部 krb5 idp プラグインがテクノロジープレビューとして利用可能

SSSD krb5 **idp** プラグインを使用すると、OAuth2 プロトコルを使用して外部アイデンティティープロバイダー (IdP) に対して認証できます。この機能は、RHEL 9.1 以降の IdM サーバーのみで使用できます。

Bugzilla:2056482

RHEL IdM では、ユーザー認証をテクノロジープレビューとして外部 ID プロバイダーに委任できる

RHEL IdM では、OAuth 2 デバイス認証フローをサポートする外部 ID プロバイダー (IdP) にユーザーを関連付けることができるようになりました。これらのユーザーが RHEL 9.1 以降で利用可能な SSSD バージョンで認証すると、外部 IdP で認証と認可を実行した後、Kerberos チケットを使用した RHEL IdMSingle Sign-On 機能を受け取ります。

主な変更には以下のものがあります。

- **ipa idp-*** コマンドによる外部 IdP への参照の追加、変更、および削除
- **ipa user-mod --user-auth-type=idp** コマンドを使用したユーザーの IdP 認証の有効化

追加情報については、外部 ID プロバイダーを使用した IdM への認証 を参照してください。

Bugzilla:2069202

テクノロジープレビューとして ACME が期限切れの証明書の自動削除をサポート

Identity Management (IdM) の自動証明書管理環境 (ACME) サービスは、期限切れの証明書を認証局 (CA) からパージする自動メカニズムをテクノロジープレビューとして追加します。その結果、ACME は指定された間隔で期限切れの証明書を自動的に削除できるようになりました。期限切れの証明書の削除はデフォルトでは無効になっています。有効にするには、次のように入力します。

この機能強化により、ACME は指定された間隔で期限切れの証明書を自動的に削除できるようになりました。

期限切れの証明書の削除はデフォルトでは無効になっています。有効にするには、次のように入力します。

```
# ipa-acme-manage pruning --enable --cron "0 0 1 * *"
```

これにより、期限切れの証明書が毎月 1 日の午前 0 時に削除されます。

注記

期限切れの証明書は、保持期間が経過すると削除されます。デフォルトでは、これは有効期限切れから 30 日後です。

詳細については、**ipa-acme-manage(1)** man ページを参照してください。

Bugzilla:2162677

9.10. デスクトップ

64 ビット ARM アーキテクチャーの GNOME がテクノロジープレビューとして利用できるようになりました。

GNOME デスクトップ環境は、テクノロジープレビューとして 64 ビット ARM アーキテクチャーで利用できます。

VNC を使用して 64 ビット ARM サーバーのデスクトップセッションに接続できるようになりました。その結果、グラフィカルアプリケーションを使用してサーバーを管理できます。

64 ビット ARM では、限定されたグラフィカルアプリケーションのセットを使用できます。以下に例を示します。

- Firefox Web ブラウザー
- Red Hat Subscription マネージャー (**subscription-manager-cockpit**)
- ファイアウォール設定 (**firewall-config**)
- ディスク使用状況アナライザー (**baobab**)

Firefox を使用して、サーバー上の Cockpit サービスに接続できます。

LibreOffice などの特定のアプリケーションは、コマンドラインインターフェイスのみを提供し、グラフィカルインターフェイスは無効になっています。

Jira:RHELPLAN-27394

テクノロジープレビューとして利用可能な IBM Z アーキテクチャー用の GNOME

GNOME デスクトップ環境は、テクノロジープレビューとして IBM Z アーキテクチャーで利用できます。

VNC を使用して IBM Z サーバーのデスクトップセッションに接続できるようになりました。その結果、グラフィカルアプリケーションを使用してサーバーを管理できます。

IBM Z では、限定されたグラフィカルアプリケーションのセットを使用できます。たとえば、次のようになります。

- Firefox Web ブラウザー
- Red Hat Subscription マネージャー (**subscription-manager-cockpit**)
- ファイアウォール設定 (**firewall-config**)

- ディスク使用状況アナライザー (**baobab**)

Firefox を使用して、サーバー上の Cockpit サービスに接続できます。

LibreOffice などの特定のアプリケーションは、コマンドラインインターフェイスのみを提供し、グラフィカルインターフェイスは無効になっています。

Jira:RHELPLAN-27737

9.11. グラフィックインフラストラクチャー

Intel Arc A シリーズグラフィックスがテクノロジープレビューとして利用可能。

Alchemist または DG2 としても知られる Intel Arc A シリーズグラフィック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。

Intel Arc A シリーズグラフィックス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次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** を次のいずれかに置き換えます。

- Intel GPU の PCI ID。
- * 文字は、すべてのアルファ品質のハードウェアで i915 ドライバーを有効にします。

Bugzilla:2041690

9.12. WEB コンソール

Stratis が RHEL Web コンソールでテクノロジープレビューとして利用可能

今回の更新で、Red Hat Enterprise Linux Web コンソールは、Stratis ストレージをテクノロジープレビューとして管理できるようになりました。

Stratis の詳細は、Stratis とは を参照してください。

Jira:RHELPLAN-122345

9.13. 仮想化

入れ子仮想マシンの作成

入れ子 KVM 仮想化は、RHEL 9 で Intel、AMD64、および IBM Z ホストで実行している KVM 仮想マシン用のテクノロジープレビューとして提供されます。この機能を使用すると、物理 RHEL 9 ホストで実行中の RHEL 7、RHEL 8、または RHEL 9 仮想マシンがハイパーバイザーとして機能し、独自の仮想マシンをホストできます。

Jira:RHELDOCS-17040

Intel SGX がテクノロジープレビューとして VM で利用可能。

テクノロジープレビューとして、RHEL 9 でホストされる仮想マシン (VM) 用に Intel Software Guard Extensions (SGX) を設定できるようになりました。SGX は、Intel ハードウェア上の特定のプロセスのデータの整合性と機密性を保護するのに役立ちます。ホスト上で SGX をセットアップすると、その機

能はその VM に渡され、ゲストオペレーティングシステム (OS) で使用できるようになります。

ゲスト OS で SGX を使用するには、まずその特定の OS 用の SGX ドライバーをインストールする必要があることに注意してください。さらに、ホスト上の SGX は VM をメモリー暗号化できません。

Jira:RHELPLAN-69761

KVM 仮想マシンの AMD SEV および SEV-ES

RHEL 9 は、テクノロジープレビューとして、KVM ハイパーバイザーを使用する AMD EPYC ホストマシンに、セキュア暗号化仮想化 (SEV) 機能を提供します。仮想マシンで有効になっている場合は、SEV が仮想マシンのメモリーを暗号化して、ホストから仮想マシンへのアクセスを防ぎます。これにより、仮想マシンのセキュリティーが向上します。

さらに、強化された SEV (Encrypted State) バージョンの SEV (SEV-ES) もテクノロジープレビューとして提供されます。SEV-ES は、仮想マシンの実行が停止すると、すべての CPU レジスターの内容を暗号化します。これにより、ホストが仮想マシンの CPU レジスターを変更したり、そこから情報を読み取ったりできなくなります。

SEV および SEV-ES は、第 2 世代の AMD EPYC CPU (コードネーム Rome) 以降のみで動作することに注意してください。また、RHEL 9 には SEV および SEV-ES の暗号化が含まれますが、SEV および SEV-ES のセキュリティー証明は含まれません。

Jira:RHELPLAN-65217

ARM 64 で仮想化が利用可能になる

テクノロジープレビューとして、ARM 64 CPU を使用してシステムに KVM 仮想マシンを作成できるようになりました。

Jira:RHELPLAN-103993

AMD64、Intel 64、および ARM 64 で virtio-mem が利用可能になる

RHEL 9 では、テクノロジープレビューとして、AMD64、Intel 64、および ARM 64 システムに **virtio-mem** 機能が追加されました。**virtio-mem** を使用すると、仮想マシンでホストメモリーを動的に追加または削除できます。

virtio-mem を使用するには、仮想マシンの XML 設定で **virtio-mem** メモリーデバイスを定義し、**virsh update-memory-device** コマンドを使用して、仮想マシンの実行中にメモリーデバイスのサイズ変更を要求します。このようなメモリーデバイスが実行中の仮想マシンに公開される現在のメモリーサイズを表示するには、仮想マシンの XML 設定を表示します。

Bugzilla:2014487、Bugzilla:2044172、Bugzilla:2044162

RHEL ゲストのインテル TDX

テクノロジープレビューとして、Intel Trust Domain Extension (TDX) 機能が RHEL 9.2 ゲストオペレーティングシステムで使用できるようになりました。ホストシステムが TDX をサポートしている場合は、トラストドメイン (TD) と呼ばれる、ハードウェアから分離された RHEL 9 仮想マシン (VM) をデプロイできます。ただし、TDX は現在 **kdump** では機能せず、TDX を有効にすると VM 上で **kdump** が失敗することに注意してください。

Bugzilla:1955275

RHEL の統合カーネルイメージがテクノロジープレビューとして利用可能になる

テクノロジープレビューとして、RHEL カーネルを仮想マシン (VM) の Unified Kernel Image (UKI) として入手できるようになりました。Unified Kernel Image は、カーネル、initramfs、およびカーネルコマンドラインを単一の署名付きバイナリーファイルに結合します。

UKI は、仮想化環境やクラウド環境、特に強力なセキュアブート機能が必要な機密 VM で使用できます。UKI は、RHEL 9 リポジトリーの **kernel-uki-virt** パッケージとして利用できます。

現在、RHEL UKI は、UEFI ブート設定のみで使用できます。

Bugzilla:2142102

Intel vGPU がテクノロジープレビューとして利用可能になる

テクノロジープレビューとして、物理 Intel GPU デバイスを、**mediated devices** と呼ばれる複数の仮想デバイスに分割できるようになりました。この仲介デバイスは、仮想 GPU として複数の仮想マシンに割り当てることができます。これにより、この仮想マシンが、1 つの物理 Intel GPU のパフォーマンスを共有します。

この機能は非推奨であり、今後の RHEL リリースでは完全に削除される予定であることに注意してください。

Jira:RHELDOCS-17050

9.14. クラウド環境の RHEL

RHEL がテクノロジープレビューとして Azure Confidential VM で利用可能になる

更新された RHEL カーネルを使用すると、RHEL 機密仮想マシン (VM) を Microsoft Azure 上でテクノロジープレビューとして作成して実行できるようになりました。新しく追加された Unified Kernel Image (UKI) により、暗号化された機密 VM イメージを Azure 上で起動できるようになりました。UKI は、RHEL 9 リポジトリーの **kernel-uki-virt** パッケージとして利用できます。

現在、RHEL UKI は、UEFI ブート設定のみで使用できます。

Jira:RHELPLAN-139800

9.15. コンテナー

Podman の Quadlet 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。

Podman v4.4 以降では、Quadlet を使用して、コンテナーの説明から **systemd** サービスファイルをテクノロジープレビューとして自動的に生成できます。コンテナーの説明は **systemd** ユニットファイル形式です。この説明では、関連するコンテナーの詳細に焦点を当てており、**systemd** でコンテナーを実行する際の技術的な複雑さは隠しています。Quadlet は、**systemd** ユニットファイルよりも作成と保守が簡単です。

詳細については、アップストリームのドキュメント と Make systemd better for Podman with Quadlet を参照してください。

Jira:RHELPLAN-148394

Fulcio と Rekor を使用した sigstore 署名のクライアントがテクノロジープレビューとして利用可能になりました。

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

ました。この追加機能はクライアント側のサポートのみであり、Fulcio サーバーや Rekor サーバーは含まれません。

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

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

Jira:RHELPLAN-136611

podman-machine コマンドはサポート対象外です。

仮想マシンを管理するための **podman-machine** コマンドは、テクノロジープレビューとしてのみ利用可能です。代わりに、コマンドラインから直接 Podman を実行してください。

Jira:RHELDOCS-16861

第10章 非推奨になった機能

ここでは、Red Hat Enterprise Linux 9 で **非推奨** となった機能の概要を説明します。

非推奨の機能は、本製品の今後のメジャーリリースではサポートされない可能性が高く、新たに実装することは推奨されません。特定のメジャーリリースにおける非推奨機能の最新情報は、そのメジャーリリースの最新版のリリースノートを参照してください。

非推奨の機能のサポートステータスは、Red Hat Enterprise Linux 9 では変更されていません。サポート期間については、Red Hat Enterprise Linux のライフサイクル および Red Hat Enterprise Linux アプリケーションストリームのライフサイクル を参照してください。

現行および今後のメジャーリリースでは、非推奨のハードウェアコンポーネントの新規実装は推奨されません。ハードウェアドライバーの更新は、セキュリティーと重大な修正のみに行われます。Red Hat では、このようなハードウェアの早期交換を推奨します。

パッケージが非推奨となり、使用の継続が推奨されない場合があります。製品からパッケージが削除されることもあります。その場合には、製品のドキュメントで、非推奨となったパッケージと同様、同一、またはより高度な機能を提供する最近のパッケージが指定され、詳しい推奨事項が記載されます。

RHEL 8 には存在するが RHEL 9 では **削除** された機能については、RHEL 9 を導入する際の考慮事項 を参照してください。

10.1. インストーラーおよびイメージの作成

非推奨のキックスタートコマンド

以下のキックスタートコマンドが非推奨になりました。

- **timezone --ntpservers**
- **timezone --nontp**
- **logging --level**
- **%packages --excludeWeakdeps**
- **%packages --instLangs**
- **%Anaconda**
- **pwpolicy**

特定のオプションだけがリスト表示されている場合は、基本コマンドおよびその他のオプションは引き続き利用でき、非推奨ではないことに注意してください。キックスタートファイルで非推奨のコマンドを使用すると、ログに警告が出力されます。**inst.ksstrict** 起動オプションを使用して、非推奨のコマンド警告をエラーにすることもできます。

Bugzilla:1899167

edge-commit および edge-container ブループリントのユーザーとグループのカスタマイズは非推奨になる

ブループリントでユーザーまたはグループのカスタマイズを指定することは、**edge-commit** および **edge-container** イメージタイプでは非推奨になりました。これは、イメージをアップグレードし、ブループリントでユーザーを再度指定しないと、ユーザーのカスタマイズが失われるためです。したがっ

て、既存の OSTree コミット (**edge-raw-image**、**edge-installer**、**edge-simplified-installer** など) のデプロイに使用されるエッジイメージタイプのブループリントでユーザーとグループを直接指定する必要があります。

ブループリントでのユーザーまたはグループのカスタマイズの指定は引き続きサポートされていますが、このサポートは最終的に削除される予定であることに注意してください。

Bugzilla:2173928

10.2. サブスクリプションの管理

subscription-manager コマンドの --token オプションは非推奨になりました。

subscription-manager register コマンドの **--token=<TOKEN>** オプションは、システムを Red Hat に登録するのに役立つ認証方法です。このオプションは、エンタイトルメントサーバーが提供する機能に応じて異なります。デフォルトのエンタイトルメントサーバー **subscription.rhsm.redhat.com** は、この機能をオフにする予定です。その結果、**subscription-manager register --token=<TOKEN>** を使用しようとすると、次のエラーメッセージが表示されて失敗する可能性があります。

> Token authentication not supported by the entitlement server

subscription-manager register コマンドのペアのオプション **--username** / **--password** および **--org** / **--activationkey** を含めるなど、他の認証方法を使用してシステムの登録を続けることができます。

Bugzilla:2163716

10.3. シェルおよびコマンドラインツール

ReaR 設定ファイルでの TMPDIR 変数の設定が非推奨になる

export TMPDIR=... などのステートメントを使用して、**/etc/rear/local.conf** または **/etc/rear/site.conf** ReaR 設定ファイルで **TMPDIR** 環境変数を設定することは、機能せず非推奨です。

ReaR 一時ファイルのカスタムディレクトリーを指定するには、ReaR を実行する前にシェル環境で変数をエクスポートします。たとえば、**export TMPDIR=...** ステートメントを実行してから、同じシェルセッションまたはスクリプトで **rear** コマンドを実行します。

Jira:RHELDOCS-18049

dump からの dump ユーティリティーが非推奨になりました。

ファイルシステムのバックアップに使用される **dump** ユーティリティーが非推奨になり、RHEL 9 では使用できなくなります。

RHEL 9 では、使用方法に基づいて、**tar**、**dd**、または **bacula** のバックアップユーティリティーを使用することが推奨されています。これにより、ext2、ext3、および ext4 のファイルシステムで完全で安全なバックアップが提供されます。

dump パッケージの **restore** ユーティリティーは、RHEL 9 で引き続き利用可能で、サポートされており、**restore** パッケージとして利用できます。

Bugzilla:1997366

Bacula の SQLite データベースバックエンドは廃止されました

Bacula バックアップシステムは、複数のデータベースバックエンド (PostgreSQL、MySQL、および

SQLite) をサポートしていました。SQLite バックエンドは廃止され、RHEL の今後のリリースではサポートされなくなります。代わりに、他のバックエンド (PostgreSQL または MySQL) のいずれかに移行し、新しい展開では SQLite バックエンドを使用しないでください。

Bugzilla:2089395

10.4. セキュリティー

SHA-1 は暗号化の目的で非推奨になる

暗号化を目的とした SHA-1 メッセージダイジェストの使用は、RHEL 9 では非推奨になりました。SHA-1 によって生成されたダイジェストは、ハッシュ衝突の検出に基づく多くの攻撃の成功例が記録化されているため、セキュアであるとは見なされません。RHEL コア暗号コンポーネントは、デフォルトで SHA-1 を使用して署名を作成しなくなりました。RHEL 9 のアプリケーションが更新され、セキュリティー関連のユースケースで SHA-1 が使用されないようになりました。

例外の中でも、HMAC-SHA1 メッセージ認証コードと Universal Unique Identifier (UUID) 値は、SHA-1 を使用して作成できます。これは、これらのユースケースが現在セキュリティーリスクをもたらさないためです。SHA-1 は、Kerberos や WPA-2 など、相互運用性および互換性に関する重要な懸念事項に関連する限られたケースでも使用できます。詳細は、RHEL 9 セキュリティーの強化ドキュメント の FIPS 140-3 に準拠していない暗号化を使用する RHEL アプリケーションのリスト を参照してください。

既存またはサードパーティーの暗号署名を検証するために SHA-1 を使用する必要がある場合は、次のコマンドを入力して有効にできます。

```
# update-crypto-policies --set DEFAULT:SHA1
```

または、システム全体の暗号化ポリシーを **LEGACY** ポリシーに切り替えることもできます。**LEGACY** は、セキュアではない他の多くのアルゴリズムも有効にすることに注意してください。

Jira:RHELPLAN-110763

fapolicyd.rules が非推奨になる

実行ルールの許可と拒否を含むファイルの **/etc/fapolicyd/rules.d/** ディレクトリーは、**/etc/fapolicyd/fapolicyd.rules** ファイルを置き換えます。**fagenrules** スクリプトは、このディレクトリー内のすべてのコンポーネントルールファイルを **/etc/fapolicyd/compiled.rules** ファイルにマージするようになりました。**/etc/fapolicyd/fapolicyd.trust** のルールは引き続き **fapolicyd** フレームワークによって処理されますが、下位互換性を確保するためのみに使用されます。

Bugzilla:2054740

RHEL 9 で SCP が非推奨になる

SCP (Secure Copy Protocol) には既知のセキュリティー脆弱性があるため、非推奨となりました。SCP API は RHEL 9 ライフサイクルで引き続き利用できますが、システムセキュリティーが低下します。

- **scp** ユーティリティーでは、SCP はデフォルトで SSH ファイル転送プロトコル (SFTP) に置き換えられます。
- OpenSSH スイートは、RHEL 9 では SCP を使用しません。
- **libssh** ライブラリーで SCP が非推奨になりました。

Jira:RHELPLAN-99136

Digest-MD5 SASL では非推奨となりました。

SASL (Simple Authentication Security Layer) フレームワークの Digest-MD5 認証メカニズムは非推奨になり、将来バージョンのメジャーリリースでは **cyrus-sasl** パッケージから削除される可能性あり

Bugzilla:1995600

OpenSSL は、MD2、MD4、MDC2、Whirlpool、Blowfish、CAST、DES、IDEA、RC2、RC4、RC5、SEED、および PBKDF1 を非推奨にします。

OpenSSL プロジェクトは、セキュアではない、一般的ではない、またはその両方であるという理由で、一連の暗号アルゴリズムを非推奨にしました。Red Hat もそれらのアルゴリズムの使用を推奨せず、RHEL 9 では、新しいアルゴリズムを使用するために暗号化されたデータを移行するためにそれらを提供しています。ユーザーは、自分のシステムのセキュリティーのためにこれらのアルゴリズムに依存してはいけません。

アルゴリズム MD2、MD4、MDC2、Whirlpool、Blowfish、CAST、DES、IDEA、RC2、RC4、RC5、SEED、および PBKDF1 の実装は、OpenSSL のレガシープロバイダーに移行されました。

レガシープロバイダーをロードし、非推奨のアルゴリズムのサポートを有効にする方法については、**/etc/pki/tls/openssl.cnf** 設定ファイルを参照してください。

Bugzilla:1975836

/etc/system-fips が非推奨に

/etc/system-fips ファイルで FIPS モードが削除されることを示すサポートにより、ファイルは今後の RHEL バージョンに含まれなくなります。FIPS モードで RHEL をインストールするには、システムのインストール時に **fips=1** パラメーターをカーネルコマンドラインに追加します。**fips-mode-setup --check** コマンドを使用して、RHEL が FIPS モードで動作しているかどうかを確認できます。

Jira:RHELPLAN-103232

libcrypt.so.1 が非推奨に

libcrypt.so.1 ライブラリーは現在非推奨であり、RHEL の将来のバージョンで削除される可能性があります。

Bugzilla:2034569

OpenSSL では、FIPS モードでの RSA 暗号化にパディングが必要です。

OpenSSL は、FIPS モードでのパディングなしの RSA 暗号化をサポートしなくなりました。パディングを使用しない RSA 暗号化は一般的ではないため、ほとんど使用されません。RSA (RSASVE) によるキーのカプセル化はパディングを使用しませんが、引き続きサポートされていることに注意してください。

Bugzilla:2168665

10.5. ネットワーク

RHEL 9 でネットワークチームが非推奨になりました

teamd サービスおよび **libteam** ライブラリーは、Red Hat Enterprise Linux 9 では非推奨になり、次回のメジャーリリースでは削除される予定です。代替として、ネットワークチームの代わりにボンディングを設定します。

Red Hat は、機能が類似するボンディングとチームの機能を 2 つ管理しなくてもいいように、カーネルベースのボンディングに注力しています。ボンディングコードは、顧客の採用率が高く、堅牢で、活発なコミュニティー開発が行われています。その結果、ボンディングコードは拡張、更新されます。

ボンディングにチームを移行する方法は、Migrating a network team configuration to network bond を参照してください。

Bugzilla:1935544

ifcfg 形式の NetworkManager 接続プロファイルが非推奨に

RHEL 9.0 以降では、**ifcfg** 形式の接続プロファイルは非推奨になりました。次の RHEL メジャーリリースでは、この形式のサポートが削除されます。ただし、RHEL 9 では、既存のプロファイルを変更すると、NetworkManager は引き続きこの形式で既存のプロファイル処理および更新します。

デフォルトでは、NetworkManager は接続プロファイルをキーファイル形式で **/etc/NetworkManager/system-connections/** ディレクトリーに保存するようになりました。**ifcfg** 形式とは異なり、キーファイル形式は、NetworkManager が提供するすべての接続設定をサポートします。キーファイル形式とプロファイルの移行方法の詳細は、NetworkManager connection profiles in keyfile format を参照してください。

Bugzilla:1894877

firewalld の iptables バックエンドが非推奨に

RHEL 9 では、**iptables** フレームワークは非推奨になりました。結果として、**iptables** バックエンドと、**firewalld** の **直接インターフェイス** も非推奨になりました。**直接インターフェイス** の代わりに、**firewalld** のネイティブ機能を使用して、必要なルールを設定できます。

Bugzilla:2089200

10.6. カーネル

RHEL 9 で ATM カプセル化が非推奨になりました

非同期転送モード (ATM) カプセル化により、ATM アダプテーションレイヤー 5(AAL-5) のレイヤー 2(ポイントツーポイントプロトコル、イーサネット) またはレイヤー 3(IP) 接続が可能になります。Red Hat は、RHEL7 以降 ATMNIC ドライバーのサポートを提供していません。ATM 実装のサポートは RHEL 9 で廃止されています。これらのプロトコルは現在、ADSL テクノロジーをサポートし、メーカーによって段階的に廃止されているチップセットのみで使用されています。したがって、ATM カプセル化は Red Hat Enterprise Linux 9 では非推奨です。

詳細は、PPP Over AAL5、Multiprotocol Encapsulation over ATM Adaptation Layer 5 、および Classical IP and ARPoverATM を参照してください。

Bugzilla:2058153

kexec-tools の kexec_load システムコールが非推奨になりました

2 番目のカーネルをロードする **kexec_load** システムコールは、将来の RHEL リリースではサポートされなくなります。**kexec_file_load** システムコールは **kexec_load** に代わるもので、現在はすべてのアーキテクチャーのデフォルトのシステムコールです。

Bugzilla:2113873

RHEL 9 でネットワークチームが非推奨になりました

teamd サービスおよび **libteam** ライブラリーは、Red Hat Enterprise Linux 9 では非推奨になり、次回のメジャーリリースでは削除される予定です。代替として、ネットワークチームの代わりにボンディングを設定します。

Red Hat は、機能が類似するボンディングとチームの機能を 2 つ管理しなくてもいいように、カーネルベースのボンディングに注力しています。ボンディングコードは、顧客の採用率が高く、堅牢で、活発なコミュニティー開発が行われています。その結果、ボンディングコードは拡張、更新されます。

ボンディングにチームを移行する方法は、Migrating a network team configuration to network bond を参照してください。

Bugzilla:2013884

10.7. ファイルシステムおよびストレージ

lvm2-activation-generator およびその生成されたサービスが RHEL 9.0 で削除される

lvm2-activation-generator プログラムとその生成されたサービス **lvm2-activation**、**lvm2-activation-early**、および **lvm2-activation-net** は、RHEL 9.0 で削除されています。サービスをアクティベートするために使用される **lvm.conf event_activation** 設定は機能しなくなりました。ボリュームグループを自動アクティブ化する唯一の方法は、イベントベースのアクティブ化です。

Bugzilla:2038183

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

libdb が非推奨になりました

RHEL 8 および RHEL 9 は、現在、LGPLv2 ライセンスで配布される Berkeley DB (**libdb**) バージョン 5.3.28 を提供しています。アップストリームの Berkeley DB バージョン 6 は、より厳しい AGPLv3 ライセンスで利用できます。

libdb パッケージは、RHEL 9 で非推奨になり、将来バージョンの RHEL では利用できない可能性があります。

また、RHEL 9 では、**libdb** から暗号アルゴリズムが削除され、RHEL 9 では複数の **libdb** 依存関係が削除されています。

libdb のユーザーは、別の鍵値データベースに移行することが推奨されます。詳細は、ナレッジベースの記事 Available replacements for the deprecated Berkeley DB (libdb) in RHEL を参照してください。

Bugzilla:1927780、Jira:RHELPLAN-80695、Bugzilla:1974657

10.9. コンパイラーおよび開発ツール

2048 より小さいサイズのキーは、openssl 3.0 で廃止されました。

2048 ビットより小さい鍵サイズは **openssl** 3.0 で廃止され、Go の FIPS モードでは機能しなくなりました。

Bugzilla:2111072

一部の PKCS1 v1.5 モードが非推奨になりました

一部の **PKCS1** v1.5 モードは、 **FIPS-140-3** で暗号化が承認されておらず、無効になっています。Go の FIPS モードでは機能しなくなります。

Bugzilla:2092016

10.10. IDENTITY MANAGEMENT

OpenDNSSec の SHA-1 が非推奨になりました

OpenDNSSec は、**SHA-1** アルゴリズムを使用したデジタル署名および認証レコードのエクスポートに対応しています。**SHA-1** アルゴリズムの使用に対応しなくなりました。RHEL 9 リリースでは、OpenDNSSec の **SHA-1** が非推奨になり、今後のマイナーリリースで削除される可能性があります。また、OpenDNSSec のサポートは、Red Hat Identity Management との統合に限定されます。OpenDNSSec はスタンドアロンでは対応していません。

Bugzilla:1979521

SSSD 暗黙的なファイルプロバイダードメインは、デフォルトで無効になっています。

/etc/shadow などのローカルファイルからユーザー情報を取得する SSSD 暗黙的な **ファイル** プロバイダードメイン、および **/etc/group** からグループ情報を取得する SSSD 暗黙的な <g id="1">ファイル</g> プロバイダードメインは、デフォルトで無効になりました。

SSSD を使用してローカルファイルからユーザーおよびグループ情報を取得するには、次のコマンドを実行します。

1. SSSD を設定します。以下のいずれかのオプションを選択します。
   a. **sssd.conf** 設定ファイルで **id_provider=files** を使用して、ローカルドメインを明示的に設定します。

   ```
   [domain/local]
   id_provider=files
   ...
   ```

   b. **sssd.conf** 設定ファイルで **enable_files_domain=true** を設定して、**ファイル** プロバイダーを有効にします。

   ```
   [sssd]
   enable_files_domain = true
   ```

2. ネームサービススイッチを設定します。

   ```
   # authselect enable-feature with-files-provider
   ```

Jira:RHELPLAN-100639

-h および -p オプションは、OpenLDAP クライアントユーティリティーで廃止されました。

アップストリームの OpenLDAP プロジェクトは、そのユーティリティーで **-h** および **-p** オプションを廃止し、代わりに **-H** オプションを使用して LDAP URI を指定することを推奨しています。その結果、RHEL 9 では、すべての OpenLDAP クライアントユーティリティーでこれら 2 つのオプションが廃止されました。**-h** および **-p** オプションは、将来のリリースで RHEL 製品から削除される予定です。

Jira:RHELPLAN-137660

SSSD files プロバイダーは非推奨になりました。

SSSD **files** プロバイダーは Red Hat Enterprise Linux (RHEL) 9 で非推奨になりました。 **files** プロバイダーは、RHEL の将来のリリースから削除される可能性があります。

Jira:RHELPLAN-139805

nsslapd-idlistscanlimit パラメーターは非推奨となり、デフォルト値が変更されました

新しいフィルターの並べ替えの最適化により、**nsslapd-idlistscanlimit** 属性が検索パフォーマンスに与える影響は、役に立つというよりも有害になります。その結果、この属性は非推奨になりました。さらに、デフォルト値が **2147483646** (無制限) に変更されました。

Bugzilla:1952241

SMB1 プロトコルは Samba では非推奨に

Samba 4.11 以降、安全でない Server Message Block バージョン 1 (SMB1) プロトコルは非推奨となり、今後のリリースでは削除される予定です。

セキュリティーを向上させるために、デフォルトでは、Samba サーバーおよびクライアントユーティリティーで SMB1 が無効になっています。

Jira:RHELDOCS-16612

10.11. デスクトップ

GTK 2 が非推奨になりました

レガシー GTK 2 ツールキットと、以下の関連パッケージが非推奨になりました。

- **adwaita-gtk2-theme**
- **gnome-common**
- **gtk2**
- **gtk2-immodules**
- **hexchat**

現在、他にも複数のパッケージが GTK 2 に依存しています。今後の RHEL メジャーリリースで非推奨パッケージへの依存が発生しないよう、これらは変更されます。

GTK 2 を使用するアプリケーションを維持する場合、Red Hat は、アプリケーションを GTK 4 に移植することを推奨します。

Jira:RHELPLAN-131882

LibreOffice が非推奨になりました。

LibreOffice RPM パッケージは非推奨となり、今後の RHEL メジャーリリースで削除される予定です。LibreOffice は、RHEL 7、8、および 9 のライフサイクル全体を通じて引き続き完全にサポートされます。

Red Hat は、RPM パッケージの代わりに、The Document Foundation が提供する次のいずれかのソースから LibreOffice をインストールすることを推奨します。

- Flathub リポジトリーの公式 Flatpak パッケージ: https://flathub.org/apps/org.libreoffice.LibreOffice
- 公式 RPM パッケージ: https://www.libreoffice.org/download/download-libreoffice/

Jira:RHELDOCS-16300

10.12. グラフィックインフラストラクチャー

Motif が非推奨になりました

アップストリームの Motif コミュニティーでの開発は非アクティブであるため、Motif ウィジェットツールキットは RHEL で非推奨になりました。

開発バリアントおよびデバッグバリアントを含む、以下の Motif パッケージが非推奨になりました。

- **motif**
- **openmotif**
- **openmotif21**
- **openmotif22**

さらに、**motif-static** パッケージが削除されました。

Red Hat は、GTK ツールキットを代替として使用することを推奨します。GTK は Motif と比較してメンテナンス性が高く、新機能を提供します。

Jira:RHELPLAN-98983

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

RHEL 9 ノードでチームを設定すると、network システムロールが非推奨の警告を表示します

ネットワークチーミング機能は、RHEL 9 では非推奨になりました。その結果、RHEL 8 コントロールノードの **network** RHEL システムロールを使用して RHEL 9 ノードでネットワークチームを設定すると、非推奨に関する警告が表示されます。

Bugzilla:1999770

10.14. 仮想化

SHA1 ベースの署名を使用した SecureBoot イメージ検証が非推奨になりました

UEFI (PE/COFF) 実行ファイルでの SHA1 ベースの署名を使用した SecureBoot イメージ検証の実行は非推奨になりました。代わりに、Red Hat は、SHA2 アルゴリズムまたはそれ以降に基づく署名を使用することを推奨します。

Bugzilla:1935497

仮想マシンスナップショットのサポートが限定されました

仮想マシンのスナップショットの作成は、現在、UEFI ファームウェアを使用していない仮想マシンのみでサポートされています。さらに、スナップショット操作中に QEMU モニターがブロックされる可能性があり、これは特定のワークロードのハイパーバイザーのパフォーマンスに悪影響を及ぼします。

また、現在の仮想マシンスナップショットの作成メカニズムは非推奨となり、Red Hat は実稼働環境での仮想マシンスナップショットの使用を推奨していないことにも注意してください。ただし、新しい VM スナップショットメカニズムは開発中であり、RHEL 9 の将来のマイナーリリースで完全に実装される予定です。

Jira:RHELPLAN-15509、Bugzilla:1621944

仮想フロッピードライバーが非推奨に

仮想フロッピーディスクデバイスを制御する **isa-fdc** ドライバーが非推奨になり、今後の RHEL ではサポートされなくなります。そのため、移行した仮想マシンとの前方互換性を確保するため、Red Hat では、RHEL 9 でホストされている仮想マシンでのフロッピーディスクデバイスの使用を推奨しません。

Bugzilla:1965079

qcow2-v2 イメージ形式が非推奨になりました

RHEL 9 では、仮想ディスクイメージの qcow2-v2 形式が非推奨になり、将来バージョンの RHEL ではサポートされなくなります。また、RHEL 9 Image Builder は、qcow2-v2 形式のディスクイメージを作成できません。

Red Hat では、qcow2-v2 の代わりに、qcow2-v3 の使用を推奨しています。qcow2-v2 イメージを、それ以降の形式に変換する場合は、**qemu-img amend** コマンドを使用します。

Bugzilla:1951814

virt-manager が非推奨になりました

Virtual Machine Manager アプリケーション (**virt-manager**) は非推奨になっています。RHEL Web コンソール (**Cockpit**) は、後続のリリースで置き換えられる予定です。したがって、GUI で仮想化を管理する場合は、Web コンソールを使用することが推奨されます。ただし、**virt-manager** で利用可能な機能によっては、RHEL Web コンソールで利用できない場合があります。

Jira:RHELPLAN-10304

libvirtd が非推奨に

モノリシック **libvirt** デーモン **libvirtd** は、RHEL 9 で非推奨になり、RHEL の将来のメジャーリリースで削除される予定です。ハイパーバイザーで仮想化を管理するために **libvirtd** を引き続き使用できることに注意してください。ただし、Red Hat では、新しく導入されたモジュラー **libvirt** デーモンに切り替えることを推奨します。手順と詳細は、RHEL 9 の仮想化の設定と管理 に関するドキュメントを参照してください。

Jira:RHELPLAN-113995

レガシー CPU モデルは非推奨になりました

かなりの数の CPU モデルが非推奨になり、RHEL の将来のメジャーリリースで仮想マシン (VM) での使用がサポートされなくなります。非推奨のモデルは次のとおりです。

- Intel の場合: Intel Xeon 55xx および 75xx プロセッサーファミリー (Nehalem とも呼ばれます) より前のモデル
- AMD の場合: AMD Opteron G4 より前のモデル
- IBM Z の場合: IBM z14 より前のモデル

VM が非推奨の CPU モデルを使用しているかどうかを確認するには、**virsh dominfo** ユーティリティーを使用し、**Message** セクションで次のような行を探します。

```
tainted: use of deprecated configuration settings
deprecated configuration: CPU model 'i486'
```

Bugzilla:2060839

RDMA ベースのライブマイグレーションが非推奨になりました

この更新により、リモートダイレクトメモリーアクセス (RDMA) を使用した実行中の仮想マシンの移行は非推奨になりました。その結果、**rdma://** 移行 URI を使用して RDMA 経由の移行を要求することは可能ですが、この機能は RHEL の将来のメジャーリリースではサポートされなくなります。

Jira:RHELPLAN-153267

10.15. コンテナー

RHEL 7 ホストでの RHEL 9 コンテナーの実行がサポート対象外

RHEL 7 ホストでは、RHEL 9 コンテナーの実行に対応していません。正常に動作するかもしれませんが、保証されません。

詳細は、Red Hat Enterprise Linux Container Compatibility Matrix を参照してください。

Jira:RHELPLAN-100087

Podman 内の SHA1 ハッシュアルゴリズムが非推奨になる

ルートレスネットワーク namespace のファイル名を生成するために使用される SHA1 アルゴリズムは Podman ではサポートされなくなりました。したがって、Podman 4.1.1 以降に更新する前に起動されたルートレスコンテナーは、ネットワークに参加している場合は (**slirp4netns** を使用するだけでなく) 再起動して、アップグレード後に起動したコンテナーに接続できるようにする必要があります。

Bugzilla:2069279

rhel9/pause が非推奨になる

rhel9/pause コンテナーイメージが非推奨になりました

Bugzilla:2106816

CNI ネットワークスタックが非推奨に

Container Network Interface (CNI) ネットワークスタックは、将来のマイナーバージョンで非推奨になる予定です。以前は、コンテナーは DNS 経由のみで単一の Container Network Interface (CNI) プラグインに接続していました。Podman v.4.0 では、新しい Netavark ネットワークスタックが導入されました。Netavark ネットワークスタックは、Podman およびその他の Open Container Initiative (OCI) コンテナー管理アプリケーションとともに使用できます。Podman 用の Netavark ネットワークスタックは、高度な Docker 機能とも互換性があります。複数のネットワーク内のコンテナーは、それらのネットワークのいずれかにあるコンテナーにアクセスできます。

詳細については、CNI から Netavark へのネットワークスタックの切り替え を参照してください。

Jira:RHELPLAN-147725

10.16. 非推奨のパッケージ

このセクションでは、非推奨となり、将来バージョンの Red Hat Enterprise Linux には含まれない可能性があるパッケージのリストを示します。

RHEL 8 と RHEL 9 との間でパッケージを変更する場合は、**RHEL 9 の導入における考慮事項** ドキュメントの パッケージの変更 を参照してください。

重要

非推奨パッケージのサポート状況は、RHEL 9 内でも変更されません。サポート期間の詳細は、Red Hat Enterprise Linux のライフサイクル および Red Hat Enterprise Linux アプリケーションストリームのライフサイクル を参照してください。

次のパッケージは RHEL 9 で非推奨になりました。

- iptables-devel
- iptables-libs
- iptables-nft
- iptables-nft-services
- iptables-utils
- libdb
- mcpp
- mod_auth_mellon
- motif
- motif-devel
- python3-pytz
- xorg-x11-server-Xorg

第11章 既知の問題

このパートでは、Red Hat Enterprise Linux 9.2 の既知の問題について説明します。

11.1. インストーラーおよびイメージの作成

キックスタートコマンドの auth および authconfig で AppStream リポジトリーが必要になる

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。

Bugzilla: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。

Bugzilla:1697896

Anaconda がアプリケーションとして実行されているシステムでの予期しない SELinux ポリシー

Anaconda がすでにインストールされているシステムでアプリケーションとして実行されている場合 (たとえば、**–image** anaconda オプションを使用してイメージファイルに別のインストールを実行する場合)、システムはインストール中に SELinux のタイプと属性を変更することを禁止されていません。そのため、SELinux ポリシーの特定の要素は、Anaconda が実行されているシステムで変更される可能性があります。この問題を回避するには、実稼働システムで Anaconda を実行せず、一時的な仮想マシンで実行します。そうすることで、実稼働システムの SELinux ポリシーは変更されません。**boot.iso** や **dvd.iso** からのインストールなど、システムインストールプロセスの一部として anaconda を実行しても、この問題の影響は受けません。

Bugzilla:2050140

サードパーティーのツールを使用して作成した USB からインストールを起動する際に、Local Media のインストールソースが検出されない

サードパーティーツールを使用して作成した USB から RHEL インストールを起動すると、インストーラーは **Local Media** インストールソースを検出できません (**Red Hat CDN**のみが検出されます)。

この問題は、デフォルトの起動オプション **int.stage2=** が **iso9660** イメージ形式の検索を試みるためです。ただし、サードパーティーツールは、別の形式の ISO イメージを作成する可能性があります。

回避策として、以下のソリューションのいずれかを使用します。

- インストールの起動時に **Tab** キーをクリックしてカーネルコマンドラインを編集し、起動オプション **inst.stage2=** を **inst.repo=** に変更します。
- Windows で起動可能な USB デバイスを作成するには、Fedora Media Writer を使用します。
- Rufus などのサードパーティーツールを使用して起動可能な USB デバイスを作成し、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。

Bugzilla:1877697

USB CD-ROM ドライブが Anaconda のインストールソースとして利用できない

USB CD-ROM ドライブがソースで、キックスタート **ignoredisk --only-use=** コマンドを指定すると、インストールに失敗します。この場合、Anaconda はこのソースディスクを見つけ、使用できません。

この問題を回避するには、**harddrive --partition=sdX --dir=/** コマンドを使用して USB CD-ROM ドライブからインストールします。その結果、インストールは失敗しなくなりました。

Bugzilla:1914955

ドライバーディスクメニューがコンソールでユーザー入力を表示できない

ドライバーディスクを使用したカーネルコマンドラインで **inst.dd** オプションを使用して RHEL インストールを開始すると、コンソールはユーザー入力を表示できません。その結果、アプリケーションがユーザー入力に応答せずにフリーズしているように見えますが、出力が表示されるため、ユーザーにとってわかりにくいます。ただし、この動作は機能に影響を与えず、**Enter** を押すとユーザー入力が登録されます。

回避策として、予想される結果を確認するには、コンソールでユーザー入力が存在しないことを無視し、入力の追加が終了したら **Enter** を押します。

Bugzilla:2109231

iso9660 ファイルシステムで、ハードドライブがパーティション分割されたインストールが失敗する

ハードドライブが **iso9660** ファイルシステムでパーティションが設定されているシステムには、RHEL をインストールできません。これは、**iso9660** ファイルシステムパーティションを含むハードディスクを無視するように設定されている、更新されたインストールコードが原因です。これは、RHEL が DVD を使用せずにインストールされている場合でも発生します。

この問題を回避するには、インストールの開始前に、キックスタートファイルに次のスクリプトを追加して、ディスクをフォーマットします。

メモ: 回避策を実行する前に、ディスクで利用可能なデータのバックアップを作成します。**wipefs** は、ディスク内の全データをフォーマットします。

```
%pre
wipefs -a /dev/sda
%end
```

その結果、インストールでエラーが発生することなく、想定どおりに機能します。

Bugzilla:1929105

Anaconda が管理者ユーザーアカウントの存在の確認に失敗する

グラフィカルユーザーインターフェイスを使用して RHEL をインストールしている場合に、管理者アカウントが作成されていると、Anaconda が確認に失敗します。その結果、管理者ユーザーアカウントがなくても、システムをインストールできてしまう可能性があります。

この問題を回避するには、管理者ユーザーアカウントを設定するか、root パスワードを設定して、root アカウントのロックを解除します。その結果、インストール済みシステムで管理タスクを実行できます。

Bugzilla:2047713

新しい XFS 機能により、バージョン 5.10 よりも古いファームウェアを持つ PowerNV IBM POWER システムが起動しなくなる

PowerNV IBM POWER システムは、ファームウェアに Linux カーネルを使用し、GRUB の代わりに Petitboot を使用します。これにより、ファームウェアカーネルのマウント **/boot** が発生し、Petitboot が GRUB 設定を読み取り、RHEL を起動します。

RHEL 9 カーネルでは、XFS ファイルシステムに **bigtime=1** 機能および **inobtcount=1** 機能が導入されています。これは、バージョン 5.10 よりも古いファームウェアのカーネルが理解できません。

この問題を回避するには、**/boot** に別のファイルシステム (ext4 など) を使用できます。

Bugzilla:1997832

rpm-ostree ペイロードをインストールすると、RHEL for Edge インストーラーイメージがマウントポイントの作成に失敗する

RHEL for Edge インストーラーイメージなどで使用される **rpm-ostree** ペイロードをデプロイする場合、インストーラーはカスタムパーティションの一部のマウントポイントを適切に作成しません。その結果、インストールは以下のエラーで中止されます。

```
The command 'mount --bind /mnt/sysimage/data /mnt/sysroot/data' exited with the code 32.
```

この問題を回避するには、以下を実行します。

- 自動パーティション設定スキームを使用し、手動でマウントポイントを追加しないでください。
- マウントポイントは、**/var** ディレクトリー内のみに手動で割り当てます。たとえば、**/var/my-mount-point** や、/、**/boot**、**/var** などの標準ディレクトリーです。

その結果、インストールプロセスは正常に終了します。

Bugzilla:2125542

ネットワークに接続されているが、DHCP または静的 IP アドレスが設定されていない場合、NetworkManager はインストール後に起動に失敗する

RHEL 9.0 以降、特定の **ip=** または kickstart ネットワーク設定が設定されていない場合、Anaconda はネットワークデバイスを自動的にアクティブ化します。Anaconda は、イーサネットデバイスごとにデフォルトの永続的な設定ファイルを作成します。接続プロファイルには、**ONBOOT** と **autoconnect** の値が **true** に設定されています。その結果、インストールされたシステムの起動中に、RHEL がネットワークデバイスをアクティブ化し、**networkManager-wait-online** サービスが失敗します。

回避策として、以下のいずれかを実行します。

- 使用する 1 つの接続を除いて、**nmcli** ユーティリティーを使用してすべての接続を削除します。以下に例を示します。

  a. すべての接続プロファイルを一覧表示します。

  ```
  # nmcli connection show
  ```

  b. 不要な接続プロファイルを削除します。

  ```
  # nmcli connection delete <connection_name>
  ```

  <connection_name> を、削除する接続の名前に置き換えます。

- 特定の **ip=** またはキックスタートネットワーク設定が設定されていない場合は、Anaconda の自動接続ネットワーク機能を無効にします。

  a. Anaconda GUI で、**Network & Host Name** に移動します。

  b. 無効にするネットワークデバイスを選択します。

  c. **Configure** をクリックします。

  d. **General** タブで、**Connect automatically with priority** の選択を解除します。

  e. **Save** をクリックします。

Bugzilla:2115783

インストール環境でドライバー更新ディスクから更新されたドライバーをロードできない

インストールの初期 RAM ディスクから同じドライバーがすでにロードされている場合、ドライバー更新ディスクからの新しいバージョンのドライバーがロードされない場合があります。そのため、ドライバーの最新バージョンをインストール環境に適用できません。

回避策として、**modprobe.blacklist=** カーネルコマンドラインオプションを **inst.dd** オプションと一緒に使用します。たとえば、ドライバー更新ディスクから **virtio_blk** ドライバーの更新バージョンが確実にロードされるようにするには、**modprobe.blacklist=virtio_blk** を使用し、通常の手順を続行してドライバー更新ディスクからドライバーを適用します。その結果、システムはドライバーの更新バージョンをロードし、それをインストール環境で使用できるようになります。

Bugzilla:2164216

キックスタートインストールでネットワーク接続の設定に失敗する

Anaconda は、NetworkManager API を通じてのみキックスタートネットワーク設定を実行します。Anaconda は、**%pre** キックスタートセクションの後にネットワーク設定を処理します。その結果、キックスタート **%pre** セクションの一部のタスクがブロックされます。たとえば、**%pre** セクションからのパッケージのダウンロードは、ネットワーク設定が利用できないため失敗します。

この問題を回避するには、以下を実行します。

- たとえば、**%pre** スクリプトの一部として **nmcli** ツールを使用して、ネットワークを設定します。
- インストーラーの起動オプションを使用して、**%pre** スクリプト用にネットワークを設定します。

その結果、**%pre** セクションのタスクにネットワークを使用できるようになり、キックスタートインストールプロセスが完了します。

Bugzilla:2173992

11.2. ソフトウェア管理

インストールプロセスが応答しなくなることがある

RHEL をインストールすると、インストールプロセスが応答しなくなることがあります。**/tmp/packaging.log** ファイルは、最後に以下のメッセージを表示します。

```
10:20:56,416 DDEBUG dnf: RPM transaction over.
```

この問題を回避するには、インストールプロセスを再起動します。

Bugzilla:2073510

11.3. シェルおよびコマンドラインツール

設定ファイルで TMPDIR 変数が設定されている場合、ReaR がリカバリー中に失敗する

/etc/rear/local.conf または **/etc/rear/site.conf** ReaR 設定ファイルで **TMPDIR** を設定してエクスポートすることは、機能せず非推奨です。

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

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

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

そのため、**/etc/rear/local.conf** ファイルで **TMPDIR** を設定してエクスポートすると、レスキューイメージの起動時に次のエラーが発生します。

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

または、**rear recover** の実行中に次のエラーが発生し、その後中断していました。

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

この問題を回避するには、カスタム一時ディレクトリーが必要な場合、ReaR を実行する前にシェル環境で変数をエクスポートして、ReaR 一時ファイル用のカスタムディレクトリーを指定します。たとえば、**export TMPDIR=...** ステートメントを実行してから、同じシェルセッションまたはスクリプトで **rear** コマンドを実行します。その結果、説明した設定でリカバリーが成功します。

Jira:RHEL-24847

ifcfg ファイルを使用したネットワークインターフェイスの名前変更に失敗する

RHEL 9 では、**initscripts** はデフォルトでインストールされません。その結果、**ifcfg** ファイルを使用したネットワークインターフェイスの名前変更に失敗します。この問題を解決するには、**udev** ルールを使用するか、ファイルをリンクしてインターフェイスの名前を変更することが推奨されます。詳細は、一貫したネットワークインターフェイスデバイスの命名 および **systemd.link(5)** の man ページを参照してください。

推奨される方法のいずれも使用できない場合は、**initscripts** パッケージをインストールします。

Bugzilla:2018112

RHEL 9 では、 chkconfig パッケージがデフォルトでインストールされない

システムサービス用のランレベル情報を更新およびクエリーする **chkconfig** パッケージは、RHEL 9 ではデフォルトでインストールされません。

サービスを管理するには、**systemctl** コマンドを使用するか、**chkconfig** パッケージを手動でインストールします。

systemd の詳細は、systemd の管理 を参照してください。**systemctl** ユーティリティーの使用方法については、systemctl を使用したシステムサービスの管理 を参照してください。

Bugzilla:2053598

Service Location Protocol (SLP) は UDP を介した攻撃に対して脆弱である

OpenSLP は、プリンターやファイルサーバーなどのローカルエリアネットワーク内のアプリケーションに動的設定メカニズムを提供します。ただし、SLP は、インターネットに接続されたシステムで UDP を介した反射型/増幅型サービス拒否攻撃に対して脆弱です。SLP を使用すると、認証されていない攻撃者は、SLP 実装によって設定された制限なしで新しいサービスを登録できます。攻撃者は UDP を使用し、送信元アドレスをスプーフィングすることで、サービス一覧を要求し、スプーフィングされたアドレスにサービス拒否を作成できます。

外部の攻撃者が SLP サービスにアクセスできないようにするには、インターネットに直接接続されているなど、信頼できないネットワークで実行されているすべてのシステムで SLP を無効にします。または、この問題を回避するには、UDP および TCP ポート 427 でトラフィックをブロックまたはフィルタリングするようにファイアウォールを設定します。

Bugzilla:2184570

11.4. インフラストラクチャーサービス

bind および unbound の両方が SHA-1- ベースの署名の検証を無効化する

bind および **unbound** コンポーネントは、すべての RSA/SHA1 (アルゴリズム番号 5) および RSASHA1-NSEC3-SHA1 (アルゴリズム番号 7) 署名の検証サポートを無効にし、署名の SHA-1 使用は DEFAULT システム全体の暗号化ポリシーで制限されます。

その結果、SHA-1、RSA/SHA1、および RSASHA1-NSEC3-SHA1 ダイジェストアルゴリズムで署名された特定の DNSSEC レコードは、Red Hat Enterprise Linux 9 で検証できず、影響を受けるドメイン名が脆弱になります。

この問題を回避するには、RSA/SHA-256 や楕円曲線キーなどの別の署名アルゴリズムにアップグレードします。

影響を受け脆弱なトップレベルドメインの詳細とリストについては、RSASHA1 で署名された DNSSEC レコードがソリューションを検証できない を参照してください。

Bugzilla:2070495

同じ書き込み可能ゾーンファイルが複数のゾーンで使用されていると、named が起動しない

BIND では、複数のゾーンに同じ書き込み可能ゾーンファイルを使用することができません。そのため、**named** で変更可能なファイルへのパスを共有するゾーンが複数存在すると、**named** が起動できなくなります。この問題を回避するには、**in-view** 節を使用して、複数のビュー間で 1 つのゾーンを共有し、異なるゾーンに異なるパスを使用するようにします。たとえば、パスにビュー名を含めます。

書き込み可能なゾーンファイルは通常、動的更新が許可されたゾーン、スレーブゾーン、または DNSSEC が管理するゾーンで使用されることに注意してください。

Bugzilla:1984982

libotr は FIPS に準拠していない

libotr ライブラリーとオフザレコード (OTR) メッセージング用のツールキットは、インスタントメッセージングの会話にエンドツーエンドの暗号化を提供します。ただし、**libotr** ライブラリーは **gcry_pk_sign()** および **gcry_pk_verify()** 関数を使用しているため、連邦情報処理標準 (FIPS) に準拠していません。その結果、FIPS モードでは **libotr** ライブラリーを使用できません。

Bugzilla:2086562

コンソール keymap を設定するには、最小限のインストールで libxkbcommon ライブラリーが必要

RHEL 9 では、特定の **systemd** ライブラリーの依存関係が動的リンクから動的ロードに変換され、システムが実行時にライブラリーを開いて使用できるようになりました。今回の変更により、必要なライブラリーをインストールしない限り、このようなライブラリーに依存する機能は使用できなくなります。これは、最小限のインストール設定を使用するシステムにおけるキーボードレイアウトの設定にも影響します。その結果、**localectl --no-convert set-x11-keymap gb** コマンドに失敗します。

この問題を回避するには、**libxkbcommon** ライブラリーをインストールします。

```
# dnf install libxkbcommon
```

Bugzilla:2214130

sysstat パッケージの %vmeff メトリックに誤った値が表示される

sysstat パッケージは、ページ再利用効率を測定するための **%vmeff** メトリックを提供します。**sysstat** は、新しいカーネルバージョンで提供されるすべての関連する **/proc/vmstat** 値を解析しないため、**sar -B** コマンドによって返される **%vmeff** 列の値は正しくありません。この問題を回避するには、**/proc/vmstat** ファイルから **%vmeff** 値を手動で計算します。詳細は、Why the **sar(1)** tool reports **%vmeff** values beyond 100 % in RHEL 8 and RHEL 9? を参照してください。

Bugzilla:2230431

11.5. セキュリティー

tangd-keygen は デフォルト以外の umask を 正しく処理しません。

tangd-keygen スクリプトは、生成されたキーファイルのファイル権限を変更しません。その結果、他のユーザーへのキーの読み取りを防止するデフォルトのユーザーファイル作成モードマスク (**umask**) が設定されているシステムでは、**tang-show-keys** コマンドはキーを表示する代わりにエラーメッセージ **Internal Error 500** を返します。

この問題を回避するには、**chmod o+r *.jwk** コマンドを使用して、**/var/db/tang** ディレクトリー内のファイルのアクセス許可を変更します。

Bugzilla:2188743

OpenSSL は、PKCS #11 トークンが生の RSA または RSA-PSS 署名の作成をサポートしているかどうかを検出しません。

TLS 1.3 プロトコルには、RSA-PSS 署名のサポートが必要です。PKCS #11 トークンが生の RSA または RSA-PSS 署名をサポートしていない場合、キーが PKCS #11 トークンによって保持されている場合、OpenSSL ライブラリーを使用するサーバーアプリケーションは RSA キーを操作できません。これにより、上記のシナリオで TLS 通信に失敗します。

この問題を回避するには、利用可能な最高の TLS プロトコルバージョンとして TLS バージョン 1.2 を使用するようにサーバーとクライアントを設定します。

Bugzilla:1681178

OpenSSL が、生の RSA または RSA-PSS の署名に対応していない PKCS #11 トークンを誤って処理する

OpenSSL ライブラリーは、PKCS #11 トークンの鍵関連の機能を検出しません。したがって、生の RSA または RSA-PSS の署名に対応しないトークンで署名が作成されると、TLS 接続の確立に失敗します。

この問題を回避するには、**/etc/pki/tls/openssl.cnf** ファイルの **crypto_policy** セクションの末尾にある **.include** 行の後に、以下の行を追加します。

```
SignatureAlgorithms =
RSA+SHA256:RSA+SHA512:RSA+SHA384:ECDSA+SHA256:ECDSA+SHA512:ECDSA+SHA384
MaxProtocol = TLSv1.2
```

これにより、このシナリオで TLS 接続を確立できます。

Bugzilla:1685470

特定の構文の使用時に scp はコピーされたファイルを空にします

scp ユーティリティーが Secure copy protocol (SCP) からよりセキュアな SSH ファイル転送プロトコル (SFTP) に変更されました。したがって、ある場所からファイルを同じ場所にコピーすると、ファイルの内容が消去されます。この問題は以下の構文に影響します。

scp localhost:/myfile localhost:/myfile

この問題を回避するには、この構文を使用して、ソースの場所と同じ宛先にファイルをコピーしないでください。

この問題は、以下の構文に対して修正されました。

- **scp /myfile localhost:/myfile**
- **scp localhost:~/myfile ~/myfile**

Bugzilla:2056884

OSCAP Anaconda アドオンは、グラフィカルインストールで調整されたプロファイルをフェッチしない

OSCAP Anaconda アドオンには、RHEL グラフィカルインストールでセキュリティープロファイルの調整を選択または選択解除するオプションがありません。RHEL 8.8 以降、アドオンはアーカイブまたは RPM パッケージからインストールするときにデフォルトで調整を考慮しません。その結果、インストールでは、OSCAP に合わせたプロファイルを取得する代わりに、次のエラーメッセージが表示されます。

```
There was an unexpected problem with the supplied content.
```

この問題を回避するには、キックスタートファイルの **%addon org_fedora_oscap** セクションにパスを指定する必要があります。次に例を示します。

```
xccdf-path = /usr/share/xml/scap/sc_tailoring/ds-combined.xml
tailoring-path = /usr/share/xml/scap/sc_tailoring/tailoring-xccdf.xml
```

その結果、OSCAP 調整プロファイルのグラフィカルインストールは、対応するキックスタート仕様のみで使用できます。

Bugzilla:2165920

Ansible 修復には追加のコレクションが必要

ansible-core パッケージによる Ansible Engine の置き換えにより、RHEL サブスクリプションで提供される Ansible モジュールのリストが削減されました。これにより、**scap-security-guide** パッケージに含まれる Ansible コンテンツを使用する修復を実行するには、**rhc-worker-playbook** パッケージからのコレクションが必要です。

Ansible 修復の場合は、以下の手順を実行します。

1. 必要なパッケージをインストールします。

   ```
   # dnf install -y ansible-core scap-security-guide rhc-worker-playbook
   ```

2. **/usr/share/scap-security-guide/ansible** ディレクトリーに移動します。

   ```
   # cd /usr/share/scap-security-guide/ansible
   ```

3. 追加の Ansible コレクションへのパスを定義する環境変数を使用して、関連する Ansible Playbook を実行します。

   ```
   # ANSIBLE_COLLECTIONS_PATH=/usr/share/rhc-worker-playbook/ansible/collections/ansible_collections/ ansible-playbook -c local -i localhost, rhel9-playbook-cis_server_l1.yml
   ```

   cis_server_l1 を、システムを修正するプロファイルの ID に置き換えます。

これにより、Ansible コンテンツは正しく処理されます。

注記

rhc-worker-playbook で提供されるコレクションのサポートは、**scap-security-guide** から取得する Ansible コンテンツの有効化だけに限定されます。

Bugzilla:2105162

oscap-anaconda-addon では、Network Servers パッケージグループを使用したシステムの CIS 強化が許可されません。

ネットワークサーバーパッケージグループが選択されているシステムに、CIS セキュリティープロファイル (**cis**、**cis_server_l1**、**cis_workstation_l1**、または **cis_workstation_l2**) を使用して RHEL ネットワークサーバーをインストールすると、**oscap-anaconda-addon** によってエラーメッセージ **package tftp has been added to the list of excluded packages, but it can't be removed from the current software selection without breaking the install** が送信されます。インストールを続行するには、ソフトウェアの選択に戻り、追加ソフトウェア **Network Servers** のチェックを外して、インストールとハードニングを終了します。次に、必要なパッケージをインストールします。

Bugzilla:2172264

Keylime は連結された PEM 証明書を受け入れません。

Keylime が単一のファイルに連結された PEM 形式の複数の証明書として証明書チェーンを受信すると、**keylime-agent-rust** Keylime コンポーネントは署名検証中に提供されたすべての証明書を正しく使用せず、TLS ハンドシェイクが失敗します。その結果、クライアントコンポーネント (**keylime_verifier** および **keylime_tenant**) は Keylime エージェントに接続できません。この問題を回避するには、複数の証明書ではなく 1 つの証明書だけを使用します。

Jira:RHELPLAN-157225

Keylime には tls_dir = default の特定のファイルが必要

Keylime verifier または registrar 設定で **tls_dir** 変数が **default** に設定されている場合、Keylime は **/var/lib/keylime/cv_ca** ディレクトリーに **cacert.crt** ファイルが存在するか確認します。ファイルが存在しない場合、**keylime_verifier** または **keylime_registrar** サービスは開始に失敗し、メッセージ **Exception: It appears that the verifier has not yet created a CA and certificates, please run the verifier first** がログに記録されます。その結果、Keylime は、ファイル名が異なるカスタム認証局 (CA) 証明書が **/var/lib/keylime/ca_cv** ディレクトリーに配置されている場合でも拒否します。

この問題を回避してカスタム CA 証明書を使用するには、**tls_dir = default** を使用する代わりに **tls_dir =/var/lib/keylime/ca_cv** を手動で指定します。

Jira:RHELPLAN-157337

デフォルトの SELinux ポリシーにより、制限のない実行ファイルがスタックを実行可能にする

SELinux ポリシーの **selinuxuser_execstack** ブール値のデフォルトの状態は on です。これは、制限のない実行ファイルがスタックを実行可能にすることを意味します。実行可能ファイルはこのオプションを使用しないでください。また、ハードコーディングされていない実行ファイルや攻撃の可能性を示している可能性があります。ただし、他のツール、パッケージ、およびサードパーティー製品との互換性のため、Red Hat はデフォルトポリシーのブール値を変更できません。そのような互換性の側面に依存しない場合は、コマンド **setsebool -P selinuxuser_execstack off** を入力して、ローカルポリシーでブール値をオフにすることができます。

Bugzilla:2064274

STIG プロファイルの SSH タイムアウトルールが誤ったオプションを設定する

OpenSSH の更新は、次の米国国防情報システム局のセキュリティー技術実装ガイド (DISA STIG) プロファイルのルールに影響を与えました。

- RHEL 9 用 DISA STIG (**xccdf_org.ssgproject.content_profile_stig**)
- RHEL 9 用、GUI の DISA STIG (**xccdf_org.ssgproject.content_profile_stig_gui**)

これらの各プロファイルでは、次の 2 つのルールが影響を受けます。

> Title: Set SSH Client Alive Count Max to zero
> CCE Identifier: CCE-90271-8
> Rule ID: xccdf_org.ssgproject.content_rule_sshd_set_keepalive_0
>
> Title: Set SSH Idle Timeout Interval
> CCE Identifier: CCE-90811-1
> Rule ID: xccdf_org.ssgproject.content_rule_sshd_set_idle_timeout

SSH サーバーに適用すると、これらの各ルールは、以前のように動作しなくなったオプション (**ClientAliveCountMax** および **ClientAliveInterval**) を設定します。その結果、OpenSSH は、これらのルールで設定されたタイムアウトに達したときに、アイドル状態の SSH ユーザーを切断しなくなりました。回避策として、これらのルールは、ソリューションが開発されるまで、DISA STIG for RHEL 9 および DISA STIG with GUI for RHEL 9 プロファイルから一時的に削除されました。

Bugzilla:2038978

GnuPG は crypto-policies によって許可されていない場合でも、SHA-1 署名の使用を誤って許可する

GNU Privacy Guard (GnuPG) 暗号化ソフトウェアは、システム全体の暗号化ポリシーで定義されている設定に関係なく、SHA-1 アルゴリズムを使用する署名を作成および検証できます。したがって、**DEFAULT** の暗号化ポリシーで暗号化の目的で SHA-1 を使用できます。これは、署名に対するこのセキュアではないアルゴリズムのシステム全体での非推奨とは一致しません。

この問題を回避するには、SHA-1 を含む GnuPG オプションを使用しないでください。これにより、セキュアでない SHA-1 署名を使用して GnuPG がデフォルトのシステムセキュリティーを下げるのを防ぎます。

Bugzilla:2070722

GPG-agent が FIPS モードで SSH エージェントとして動作しない

gpg-agent ツールは、FIPS モードが MD5 ダイジェストが無効であっても **ssh-agent** プログラムにキーを追加する際に MD5 フィンガープリントを作成します。その結果、**ssh-add** ユーティリティーは認証エージェントへのキーの追加に失敗します。

この問題を回避するには、**~/.gnupg/sshcontrol** ファイルを **gpg-agent --daemon --enable-ssh-support** コマンドを使用せずに作成します。たとえば、**gpg --list-keys** コマンドの出力を **<FINGERPRINT> 0** 形式で **~/.gnupg/sshcontrol** に貼り付けることができます。これにより、**gpg-agent** は SSH 認証エージェントとして機能します。

Bugzilla:2073567

OpenSCAP のメモリー消費の問題

メモリーが限られているシステムでは、OpenSCAP スキャナが途中で終了するか、結果ファイルが生成されない可能性があります。この問題を回避するには、スキャンプロファイルをカスタマイズして、/ ファイルシステム全体の再帰を含むルールの選択を解除します。

- **rpm_verify_hashes**
- **rpm_verify_permissions**
- **rpm_verify_ownership**
- **file_permissions_unauthorized_world_writable**
- **no_files_unowned_by_user**
- **dir_perms_world_writable_system_owned**
- **file_permissions_unauthorized_suid**
- **file_permissions_unauthorized_sgid**
- **file_permissions_ungroupowned**
- **dir_perms_world_writable_sticky_bits**

詳細とその他の回避策については、関連する ナレッジベースの記事 を参照してください。

Bugzilla:2161499

キックスタートインストール時のサービス関連のルールの修正が失敗する場合があります。

キックスタートのインストール時に、OpenSCAP ユーティリティーで、サービス **enable** または **disable** 状態の修正が必要でないことが誤って表示されることがあります。これにより、OpenSCAP が、インストール済みシステムのサービスを非準拠状態に設定する可能性があります。回避策として、キックスタートインストール後にシステムをスキャンして修復できます。これにより、サービス関連の問題が修正されます。

BZ#1834716

11.6. ネットワーク

nm-cloud-setup サービスは、手動で設定されたセカンダリー IP アドレスをインターフェイスから削除する

クラウド環境から受け取った情報に基づいて、**nm-cloud-setup** サービスがネットワークインターフェイスを設定します。インターフェイスを手動で設定するには、**nm-cloud-setup** を無効にします。ただし、場合によっては、ホスト上の他のサービスもインターフェイスを設定できます。たとえば、これらのサービスはセカンダリー IP アドレスを追加できます。**nm-cloud-setup** がセカンダリー IP アドレスを削除しないようにするには、

1. **nm-cloud-setup** サービスおよびタイマーを停止して無効にします。

```
# systemctl disable --now nm-cloud-setup.service nm-cloud-setup.timer
```

2. 使用可能な接続プロファイルを表示します。

```
# nmcli connection show
```

3. 影響を受ける接続プロファイルを再アクティブ化します。

```
# nmcli connection up "<profile_name>"
```

その結果、このサービスは、手動で設定されたセカンダリー IP アドレスをインターフェイスから削除しなくなりました。

Bugzilla:2151040

セッションキーの更新に失敗すると、接続が切断される

カーネルトランスポートレイヤーセキュリティー (kTLS) プロトコルは、対称暗号で使用されるセッションキーの更新をサポートしていません。その結果、ユーザーはキーを更新することができず、接続が切断されてしまいます。この問題を回避するには、kTLS を無効にしてください。その結果、この回避策により、セッションキーを正常に更新できます。

Bugzilla:2013650

initscripts パッケージがデフォルトでインストールされない

デフォルトでは、**initscripts** パッケージはインストールされません。これにより、**ifup** ユーティリティーおよび **ifdown** ユーティリティーが利用できません。別の方法として、**nmcli connection up** コマンドおよび **nmcli connection down** コマンドを使用して、接続を有効および無効にします。提案された代替案がうまくいかない場合は、問題を報告し、**NetworkManager-initscripts-updown** パッケージをインストールしてください。これは、**ifup** および **ifdown** ユーティリティー用の NetworkManager ソリューションを提供します。

Bugzilla:2082303

mlx5 ドライバーを使用し、MTU が 3498 バイトを超えた状態で XDP マルチバッファーモードを使用するには、RX Striding RQ を無効にする必要がある

次の条件にすべて一致するホストにおいて、マルチバッファーモードを使用した eXpress Data Path (XDP) スクリプトの実行が失敗します。

- ホストは **mlx5** ドライバーを使用しています。
- 最大伝送単位 (MTU) の値が 3498 バイトを超えています。
- Mellanox インターフェイスで、受信ストライディング受信キュー (RX Striding RQ) 機能が有効になっています。

すべての条件が当てはまる場合、スクリプトは **link set xdp fd failed** エラーで失敗します。MTU が高いホストで XDP スクリプトを実行するには、Mellanox インターフェイスで RX Striding RQ を無効にします。

```
# ethtool --set-priv-flags <interface_name> rx_striding_rq off
```

これにより、**mlx5** ドライバーを使用し、MTU 値が 3498 バイトを超えるインターフェイスで XDP マルチバッファーモードを使用できます。

Jira:RHEL-6496

11.7. カーネル

カーネルの kdump メカニズムにより、64K カーネルで OOM エラーが発生します。

64 ビット ARM アーキテクチャー上の 64K カーネルページサイズは、4KB カーネルよりも多くのメモリーを使用します。その結果、**kdump** はカーネルパニックを引き起こし、メモリー不足 (OOM) エラーでメモリー割り当てが失敗します。回避策として、**crashkernel** 値を手動で 640 MB に設定します。たとえば、**crashkernel=** パラメーターを **crashkernel=2G- :640M** として設定します。

結果として、説明されているシナリオでは、**kdump** メカニズムは 64K カーネルで失敗しません。

Bugzilla:2160676

カーネルページサイズに依存する顧客アプリケーションは、ページサイズカーネルを 4k から 64k に移行するときに更新が必要になる場合があります。

RHEL は、4k と 64k の両方のページサイズのカーネルと互換性があります。4K カーネルページサイズに依存する顧客アプリケーションは、4K から 64K ページサイズカーネルに移行するときに更新が必要になる場合があります。この既知の例には、**jemalloc** および依存アプリケーションが含まれます。

jemalloc メモリーアロケータライブラリーは、システムのランタイム環境で使用されるページサイズの影響を受けます。このライブラリーは、たとえば、**--with-lg-page=16** または **env JEMALLOC_SYS_WITH_LG_PAGE=16** (**jemallocator** Rust クレートの場合) で設定されている場合、4k および 64k ページサイズのカーネルと互換性があるように構築できます。その結果、ランタイム環境のページサイズと、**jemalloc** に依存するバイナリーのコンパイル時に存在したページサイズとの間に不一致が発生する可能性があります。その結果、**jemalloc** ベースのアプリケーションを使用すると、次のエラーが発生します。

```
<jemalloc>: Unsupported system page size
```

この問題を回避するには、次のいずれかの方法を使用します。

- 適切なビルド設定または環境オプションを使用して、4k および 64k ページサイズと互換性のあるバイナリーを作成します。
- 最終的な 64k カーネルおよびランタイム環境で起動した後、**jemalloc** を使用するユーザー空間パッケージをビルドします。

たとえば、同じく **jemalloc** を使用する **fd-find** ツールを、**cargo** Rust パッケージマネージャーを使用して構築できます。最後の 64k 環境では、**cargo** コマンドを入力して、すべての依存関係の新しいビルドをトリガーし、ページサイズの不一致を解決します。

```
# cargo install fd-find --force
```

Bugzilla:2167783

kdump サービスが IBM Z システムで initrd ファイルの構築に失敗する

64 ビットの IBM Z システムでは、**s390-** subchannels などの **znet** 関連の設定情報が非アクティブな **NetworkManager** 接続プロファイルに存在する場合、**kdump** サービスは初期 RAM ディスク (**initrd**) のロードに失敗します。その結果、**kdump** メカニズムは次のエラーで失敗します。

```
dracut: Failed to set up znet
kdump: mkdumprd: failed to make kdump initrd
```

回避策として、次のいずれかの解決策を使用してください。

- **znet** 設定情報を持つ接続プロファイルを再利用して、ネットワークボンディングまたはブリッジを設定します。

  ```
  $ nmcli connection modify enc600 master bond0 slave-type bond
  ```

- 非アクティブな接続プロファイルからアクティブな接続プロファイルに **znet** 設定情報をコピーします。

a. **nmcli** コマンドを実行して、**NetworkManager** 接続プロファイルを照会します。

```
# nmcli connection show

NAME                    UUID              TYPE   Device

bridge-br0           ed391a43-bdea-4170-b8a2 bridge   br0
bridge-slave-enc600  caf7f770-1e55-4126-a2f4 ethernet enc600
enc600               bc293b8d-ef1e-45f6-bad1 ethernet --
```

b. 非アクティブな接続からの設定情報でアクティブなプロファイルを更新します。

```
#!/bin/bash
 inactive_connection=enc600
 active_connection=bridge-slave-enc600
 for name in nettype subchannels options; do
 field=802-3-ethernet.s390-$name
 val=$(nmcli --get-values "$field"connection show "$inactive_connection")
 nmcli connection modify "$active_connection" "$field" $val"
 done
```

c. 変更を有効にするために **kdump** サービスを再起動します。

```
# kdumpctl restart
```

Bugzilla:2064708

kTLS は、TLS 1.3 の NIC へのオフロードをサポートしない

Kernel Transport Layer Security(kTLS) は、TLS 1.3 の NIC へのオフロードをサポートしていません。そのため、NIC が TLS オフロードをサポートしていても、TLS 1.3 によるソフトウェア暗号化が使用されます。この問題を回避するには、オフロードが必要な場合は TLS 1.3 を無効にしてください。その結果、TLS 1.2 のみをオフロードすることができます。TLS 1.3 が使用されている場合、TLS 1.3 をオフロードすることができないため、パフォーマンスが低下します。

Bugzilla:2000616

デフォルトでは、Delay Accounting 機能は SWAPIN および IO% 統計列を表示しない

初期のバージョンとは異なり、**Delayed Accounting** 機能はデフォルトで無効になっています。その結果、**iotop** アプリケーションは **SWAPIN** および **IO%** 統計列を表示せず、次の警告を表示します。

```
CONFIG_TASK_DELAY_ACCT not enabled in kernel, cannot determine SWAPIN and IO%
```

taskstats インターフェイスを使用する **Delay Accounting** 機能は、スレッドグループに属するすべてのタスクまたはスレッドの遅延統計を提供します。タスク実行の遅延は、カーネルリソースが利用可能になるのを待つときに発生します。たとえば、空き CPU が実行されるのを待っているタスクです。統計は、タスクの CPU 優先度、I/O 優先度、および **rss** 制限値を適切に設定するのに役立ちます。

回避策として、実行時または起動時に、**delayacct** ブートオプションを有効にすることができます。

- 実行時に **delayacct** を有効にするには、次のように入力します。

```
echo 1 > /proc/sys/kernel/task_delayacct
```

このコマンドはシステム全体で機能を有効にしますが、このコマンドの実行後に開始したタスクに対してのみ有効であることに注意してください。

- 起動時に **delayacct** を永続的に有効にするには、次のいずれかの手順を使用します。
  - **/etc/sysctl.conf** ファイルを編集して、デフォルトのパラメーターをオーバーライドします。
    a. 次のエントリーを **/etc/sysctl.conf** ファイルに追加します。

    ```
    kernel.task_delayacct = 1
    ```

    詳細は、Red Hat Enterprise Linux で sysctl 変数を設定する方法 を参照してください。
    b. システムを再起動して、変更を反映させます。
  - カーネルコマンドラインに **delayacct** オプションを追加します。
    詳細は、カーネルコマンドラインパラメーターの設定 を参照してください。

その結果、**iotop** アプリケーションは **SWAPIN** および **IO%** 統計列を表示します。

Bugzilla:2132480

kdump メカニズムは、LUKS 暗号化ターゲットで vmcore ファイルをキャプチャーできない

Linux Unified Key Setup (LUKS) で暗号化されたパーティションを使用するシステムで **kdump** を実行する場合、システムには一定量の使用可能なメモリーが必要です。使用可能なメモリーが必要なメモリー量より少ない場合、**systemd-cryptsetup** サービスはパーティションのマウントに失敗します。その結果、2 番目のカーネルは LUKS 暗号化ターゲット上のクラッシュダンプファイル (**vmcore**) のキャプチャに失敗します。

kdumpctl Estimate コマンドを使用すると、**kdump** に必要な推奨メモリーサイズである **推奨クラッシュカーネル値** を照会できます。

この問題を回避するには、次の手順を使用して、LUKS 暗号化ターゲットで **kdump** に必要なメモリーを設定します。

1. 推定 **crashkernel** 値を出力します。

   ```
   # kdumpctl estimate
   ```

2. **crashkernel** の値を増やして、必要なメモリー量を設定します。

   ```
   # grubby --args=crashkernel=652M --update-kernel=ALL
   ```

3. システムを再起動して、変更を反映させます。

   ```
   # reboot
   ```

これにより、LUKS で暗号化したパーティションがあるシステムで **kdump** が正常に機能します。

Bugzilla:2017401

起動時にクラッシュカーネルメモリーの割り当てに失敗する

特定の Ampere Altra システムでは、利用可能なメモリーが 1 GB 未満の場合に、起動中に **kdump** の使用に対してクラッシュカーネルメモリーの割り当てに失敗します。その結果、**kdumpctl** コマンドは **kdump** サービスの起動に失敗します。

この問題を回避するには、以下のいずれかを実行します。

- **crashkernel** パラメーターの値を 240 MB 以上減らしてサイズ要件に合わせます (例: **crashkernel=240M**)。
- **crashkernel=x,high** オプションを使用して、**kdump** 用に 4 GB を超えるクラッシュカーネルメモリーを予約します。

その結果、Ampere Altra システムで **kdump** のクラッシュカーネルメモリー割り当てが失敗しなくなりました。

Bugzilla:2065013

VMD が有効になっている場合、RHEL が NVMe ディスクを認識できません。

ドライバーをリセットまたは再接続しても、ボリューム管理デバイス (VMD) ドメインは現在ソフトリセットされません。その結果、ハードウェアはデバイスを適切に検出して列挙できなくなります。その結果、VMD が有効になっているオペレーティングシステムは、特にサーバーをリセットするときや VM マシンを操作するときに、NVMe ディスクを認識しません。

Bugzilla:2128610

iwl7260-firmware により、Intel Wi-Fi 6 AX200、AX210、および Lenovo ThinkPad P1 Gen 4 で Wi-Fi が切断される

iwl7260-firmware または **iwl7260-wifi** ドライバーを RHEL 9.1 以降で提供されるバージョンに更新すると、ハードウェアが不正な内部状態になります。その状態を誤って報告します。その結果、Intel Wifi 6 カードが機能せず、次のエラーメッセージが表示される場合があります。

```
kernel: iwlwifi 0000:09:00.0: Failed to start RT ucode: -110
kernel: iwlwifi 0000:09:00.0: WRT: Collecting data: ini trigger 13 fired (delay=0ms)
kernel: iwlwifi 0000:09:00.0: Failed to run INIT ucode: -110
```

未確認の回避策は、システムの電源をオフにしてから再度オンにすることです。再起動しないでください。

Bugzilla:2129288

kmod の weak-modules がモジュールの相互依存関係で機能しない

kmod パッケージによって提供される **weak-modules** スクリプトは、どのモジュールがインストールされたカーネルと kABI 互換であるかを判別します。ただし、モジュールのカーネル互換性をチェックしている間、**weak-modules は** モジュールシンボルの依存関係を、それらがビルドされたカーネルの上位リリースから下位リリースへと処理します。結果として、異なるカーネルリリースに対して構築された相互依存関係を持つモジュールは互換性がないと解釈される可能性があるため、**weak-modules** はこのシナリオでは機能しません。

この問題を回避するには、新しいカーネルをインストールする前に、最新のストックカーネルに対して追加のモジュールをビルドまたは配置します。

Bugzilla:2103605

Mellanox ConnectX-5 アダプターの使用中に mlx5 ドライバーが失敗します。

イーサネットスイッチデバイスドライバーモデル (**switchdev**) モードでは、デバイス管理フローステアリング (DMFS) パラメーターと **ConnectX-5** アダプターがサポートするハードウェアを使用して設定されていると、**mlx5** ドライバーが失敗します。その結果、次のエラーメッセージが表示されることがあります。

```
BUG: Bad page cache in process umount pfn:142b4b
```

この問題を回避するには、DMFS の代わりにソフトウェア管理フローステアリング (SMFS) パラメーターを使用します。

Bugzilla:2180665

コア数が大きいシステムのリアルタイムカーネルのハードウェア認定では、ロックの競合を回避するために skew-tick=1 ブートパラメーターを渡す必要がある場合があります。

多数のソケットとコアカウントが大きい大規模なシステムまたは中規模のシステムでは、タイムキーピングシステムで使用される **xtime_lock** のロック競合により、レイテンシーの急増が発生する可能性があります。その結果、レイテンシーの急増およびハードウェア認証のレイテンシーは、マルチプロセッシングシステムで発生する可能性があります。回避策として、**skew_tick=1** ブートパラメーターを追加することで、CPU ごとにタイマーティックをオフセットし、別のタイミングで開始できます。

ロックの競合を回避するには、**skew_tick=1** を有効にします。

1. **grubby** で **skew_tick=1** パラメーターを有効にします。

   ```
   # grubby --update-kernel=ALL --args="skew_tick=1"
   ```

2. 変更を有効にするために再起動します。

3. **cat /proc/cmdline** コマンドを実行して、新しい設定を確認します。

skew_tick=1 を有効にすると、消費電力が大幅に増加するため、レイテンシーの影響を受けるリアルタイムワークロードを実行している場合にのみ有効にする必要があります。

Bugzilla:2214508

64 ビット ARM CPU で正しくコンパイルされたドライバーでのプログラム失敗に関して dkms が誤った警告を出す

Dynamic Kernel Module Support (**dkms**) ユーティリティーは、64 ビット ARM CPU のカーネルヘッダーが、ページサイズが 4 キロバイトのカーネルと 64 キロバイトのカーネルの両方で動作することを認識しません。その結果、**dkms** は、カーネルの更新時に **kernel-64k-devel** パッケージがインストールされていない場合、正しくコンパイルされたドライバーでプログラムが失敗した理由に関して誤った警告を出します。この問題を回避するには、**kernel-headers** パッケージをインストールします。このパッケージは、両タイプの ARM CPU アーキテクチャー用のヘッダーファイルを含むもので、**dkms** とその要件に特化したものではありません。

Jira:RHEL-25967

11.8. ファイルシステムおよびストレージ

CHAP 認証の試行に失敗した後、no authentication メソッドを使用して iSCSI サーバーにログインできない

CHAP 認証を使用して iSCSI ディスクを追加し、間違った認証情報によりログイン試行に失敗した場合は、**no authentication** 方式でのディスクへの再ログインに失敗します。この問題を回避するには、現行セッションを閉じて、**no authentication** メソッドを使用してログインします。

Bugzilla:1983602

デバイスマッパーマルチパスは NVMe/TCP ではサポートされない

nvme-tcp ドライバーで Device Mapper Multipath を使用すると、コールトレースの警告とシステムの不安定性が発生する可能性があります。この問題を回避するには、NVMe/TCP ユーザーはネイティブ NVMe マルチパスを有効にする必要があり、NVMe で **device-mapper-multipath** ツールを使用しないでください。

デフォルトでは、ネイティブ NVMe マルチパスは RHEL 9 で有効になっています。詳細は、Enabling multipathing on NVMe devices を参照してください。

Bugzilla:2033080

blk-availability systemd サービスは、複雑なデバイススタックを非アクティブ化する

systemd では、デフォルトのブロック非アクティブ化コードは、仮想ブロックデバイスの複雑なスタックを常に正しく処理するとは限りません。一部の設定では、シャットダウン中に仮想デバイスが削除されない場合があり、エラーメッセージがログに記録されます。この問題を回避するには、次のコマンドを実行して、複雑なブロックデバイススタックを非アクティブ化します。

```
# systemctl enable --now blk-availability.service
```

その結果、複雑な仮想デバイススタックはシャットダウン中に正しく非アクティブ化され、エラーメッセージは生成されません。

Bugzilla:2011699

クォータを有効にしてマウントされた XFS ファイルシステムでは、クォータアカウンティングを無効にすることはできなくなりました。

RHEL 9.2 以降、クォータを有効にしてマウントされた XFS ファイルシステムでクォータアカウンティングを無効にすることはできなくなりました。

この問題を回避するには、クォータオプションを削除してファイルシステムを再マウントし、クォータアカウンティングを無効にします。

Bugzilla:2160619

/etc/fstab にマウントポイントとして NVMe-FC デバイスを追加すると、システムの起動に失敗する

/etc/fstab ファイルを介してマウントされた Non-volatile Memory Express over Fibre Channel (NVMe-FC) デバイスは起動時にマウントに失敗し、システムは緊急モードに入ります。これは、**nvme-cli nvmf-autoconnect systemd** サービスの既知のバグが原因です。

Bugzilla:2168603

NVMe デバイスの udev ルールの変更

NVMe デバイスの udev ルールに変更があり、**OPTIONS="string_escape=replace"** パラメーターが追加されました。これにより、デバイスのシリアル番号の先頭に空白がある場合、一部のベンダーではディスク ID による名前が変更されます。

Bugzilla:2185048

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

python3.11-lxml は lxml.isoschematron サブモジュールを提供しない

python3.11-lxml パッケージは、オープンソースライセンスの下にないため、**lxml.isoschematron** サブモジュールなしで配布されます。サブモジュールは ISO Schematron サポートを実装します。代わりに、ISO-Schematron 前の検証を **lxml.etree.Schematron** クラスで利用できます。**python3.11-lxml** パッケージの残りのコンテンツは影響を受けません。

Bugzilla:2157708

MySQL および MariaDB の --ssl-fips-mode オプションでは FIPS モードが変更されない

MySQL の **--ssl-fips-mode** オプションと RHEL の **MariaDB** は、アップストリームとは異なる動作をします。

RHEL 9 では、**--ssl-fips-mode** を **mysqld** デーモンまたは **mariadbd** デーモンの引数として使用する場合や、**MySQL** または **MariaDB** サーバー設定ファイルに **ssl-fips-mode** を使用すると、**--ssl-fips-mode** はこれらのデータベースサーバーの FIPS モードを変更しません。

代わりに、以下のようになります。

- **--ssl-fips-mode** を **ON** に設定すると、**mysqld** サーバーデーモンまたは **mariadbd** サーバーデーモンは起動しません。
- FIPS が有効なシステムで **--ssl-fips-mode** を **OFF** に設定すると、**mysqld** サーバーデーモンまたは **mariadbd** サーバーデーモンは FIPS モードで稼働します。

これは、特定のコンポーネントではなく、RHEL システム全体で FIPS モードを有効または無効にする必要があるためです。

したがって、RHEL の **MySQL** または **MariaDB** では **--ssl-fips-mode** オプションを使用しないでください。代わりに、FIPS モードが RHEL システム全体で有効になっていることを確認します。

- FIPS モードが有効な RHEL をインストールすることが推奨されます。インストール時に FIPS モードを有効にすると、システムは FIPS で承認されるアルゴリズムと継続的な監視テストですべての鍵を生成するようになります。FIPS モードで RHEL をインストールする方法は、FIPS モードでのシステムのインストール を参照してください。
- または、FIPS モードへのシステムの切り替え の手順に従って、RHEL システム全体の FIPS モードを切り替えることができます。

Bugzilla:1991500

11.10. コンパイラーおよび開発ツール

64 ビット ARM アーキテクチャーの SystemTap で一部のシンボルベースのプローブが動作しない

カーネル設定は、**SystemTap** に必要な特定の機能を無効にします。したがって、一部のシンボルベースのプローブは、64 ビット ARM アーキテクチャーでは機能しません。その結果、影響を受ける **SystemTap** スクリプトが実行されないか、目的のプローブポイントでヒットが収集されない可能性があります。

このバグは、RHBA-2022:5259 アドバイザリーのリリースにより、残りのアーキテクチャーで修正されていることに注意してください。

Bugzilla:2083727

GCC Toolset 12 の GCC: Intel Sapphire Rapids プロセッサーで CPU 検出が失敗する場合がある

Intel Sapphire Rapids プロセッサー上の CPU 検出は、**AVX512_VP2INTERSECT** 機能の存在に依存しています。この機能は GCC の GCC Toolset 12 バージョンから削除されたため、Intel Sapphire Rapids プロセッサーでは CPU 検出が失敗する可能性があります。

Bugzilla:2141718

11.11. IDENTITY MANAGEMENT

Directory Server で接尾辞の紹介の設定に失敗する

Directory Server でバックエンド参照を設定すると、**dsconf <instance_name> backend suffix set --state referral** コマンドを使用したバックエンドの状態設定に失敗し、次のエラーが表示されます。

```
Error: 103 - 9 - 53 - Server is unwilling to perform - [] - need to set nsslapd-referral before moving to referral state
```

これにより、接尾辞の参照の設定に失敗します。この問題を回避するには、以下のコマンドを実行します。

1. **nsslapd-referral** パラメーターを手動で設定します。

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com

dn: cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping tree,cn=config
changetype: modify
add: nsslapd-referral
nsslapd-referral: ldap://remote_server:389/dc=example,dc=com
```

2. バックエンド状態を設定します。

```
# dsconf <instance_name> backend suffix set --state referral
```

その結果、回避策により、接尾辞の参照を設定できます。

Bugzilla:2063140

dsconf ユーティリティーには、entryUUID プラグインの修正タスクを作成するオプションがない

dsconf ユーティリティーは、**entryUUID** プラグインの修正タスクを作成するオプションを提供しません。その結果、管理者は **dsconf** を使用して、既存のエントリーに **entryUUID** 属性を自動的に追加するタスクを作成することはできません。回避策として、タスクを手動で作成します。

```
# ldapadd -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: cn=entryuuid_fixup_<time_stamp>,cn=entryuuid task,cn=tasks,cn=config
objectClass: top
```

```
objectClass: extensibleObject
basedn: <fixup base tree>
cn: entryuuid_fixup_<time_stamp>
filter: <filtered_entry>
```

タスクが作成された後、Directory Server は **entryUUID** 属性が欠落しているか無効であるエントリーを修正します。

Bugzilla:2047175

MIT Kerberos は PKINIT の ECC 証明書をサポートしない

MIT Kerberos は、初期認証 (PKINIT) の公開鍵暗号化における楕円曲線暗号化 (ECC) サポートの設計を説明するコメントドキュメントに、RFC5349 要求を実装していません。したがって、RHEL で使用される MIT **krb5-pkinit** パッケージは ECC 証明書に対応していません。詳細は、 Elliptic Curve Cryptography (ECC) Support for Public Key Cryptography for Initial Authentication in Kerberos (PKINIT) を参照してください。

Bugzilla:2106043

PKINIT が AD KDC に対して機能するように、DEFAULT:SHA1 サブポリシーを RHEL 9 クライアントに設定する必要がある

SHA-1 ダイジェストアルゴリズムは RHEL 9 で非推奨になり、初期認証 (PKINIT) の公開鍵暗号化の CMS メッセージは、より強力な SHA-256 アルゴリズムで署名されるようになりました。

しかし、Active Directory (AD) Kerberos Distribution Center (KDC) は引き続き SHA-1 ダイジェストアルゴリズムを使用して CMS メッセージに署名します。その結果、RHEL 9 Kerberos クライアントは、AD KDC に対して PKINIT を使用してユーザーを認証できません。

この問題を回避するには、次のコマンドを使用して、RHEL 9 システムで SHA-1 アルゴリズムのサポートを有効にします。

```
# update-crypto-policies --set DEFAULT:SHA1
```

Bugzilla:2060798

RHEL 9 Kerberos エージェントが RHEL-9 以外および AD 以外の Kerberos エージェントと通信すると、ユーザーの PKINIT 認証に失敗する

クライアントまたは Kerberos Distribution Center (KDC) のいずれかの RHEL 9 Kerberos エージェントが、Active Directory (AD) エージェントではない RHEL-9 Kerberos エージェントとやりとりすると、ユーザーの PKINIT 認証に失敗します。この問題を回避するには、以下のいずれかのアクションを実行します。

- RHEL 9 エージェントの crypto-policy を **DEFAULT:SHA1** に設定して、SHA-1 署名の検証を許可します。

  ```
  # update-crypto-policies --set DEFAULT:SHA1
  ```

- RHEL 9 以外および AD 以外のエージェントを更新して、SHA-1 アルゴリズムを使用して CMS データを署名しないようにします。そのためには、Kerberos パッケージを SHA-1 の代わりに SHA-256 を使用するバージョンに更新します。
  - CentOS 9 Stream: krb5-1.19.1-15
  - RHEL 8.7: krb5-1.18.2-17

- RHEL 7.9: krb5-1.15.1-53
- Fedora Rawhide/36: krb5-1.19.2-7
- Fedora 35/34: krb5-1.19.2-3

その結果、ユーザーの PKINIT 認証が正しく機能します。

他のオペレーティングシステムでは、エージェントが SHA-1 ではなく SHA-256 で CMS データを署名するように krb5-1.20 リリースであることに注意してください。

PKINIT が AD KDC に対して機能するように、DEFAULT:SHA1 サブポリシーを RHEL 9 クライアントに設定する必要があるも併せて参照してください。

Bugzilla:2077450

AD 信頼の FIPS サポートには、AD-SUPPORT 暗号サブポリシーが必要

Active Directory (AD) は、AES SHA-1 HMAC 暗号化タイプを使用します。これは、デフォルトで RHEL 9 の FIPS モードでは許可されていません。AD トラストで RHEL 9 ホストを使用する場合は、IdM ソフトウェアをインストールする前に、AESSHA-1HMAC 暗号化タイプのサポートを有効にしてください。

FIPS 準拠は技術的合意と組織的合意の両方を伴うプロセスであるため、**AD-SUPPORT** サブポリシーを有効にして技術的手段が AES SHA-1 HMAC 暗号化タイプをサポートできるようにする前に、FIPS 監査人に相談してから、RHEL IdM をインストールしてください。

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

Bugzilla:2057471

Heimdal クライアントは、RHEL 9 KDC に対して PKINIT を使用してユーザーを認証できない

デフォルトでは、Heimdal Kerberos クライアントは、Internet Key Exchange (IKE) に Modular Exponential (MODP) Diffie-Hellman Group 2 を使用して、IdM ユーザーの PKINIT 認証を開始します。ただし、RHEL 9 の MIT Kerberos Distribution Center (KDC) は、MODP Group 14 および 16 のみに対応しています。

したがって、Heimdal クライアントで **krb5_get_init_creds: PREAUTH_FAILED** エラーが発生し、RHEL MIT KDC では **Key parameters not accepted** が発生します。

この問題を回避するには、Heimdal クライアントが MODP Group 14 を使用していることを確認してください。クライアント設定ファイルの **libdefaults** セクションで **pkinit_dh_min_bits** パラメーターを 1759 に設定します。

```
[libdefaults]
pkinit_dh_min_bits = 1759
```

その結果、Heimdal クライアントは、RHEL MIT KDC に対する PKINIT 事前認証を完了します。

Bugzilla:2106296

FIPS モードの IdM は、双方向のフォレスト間信頼を確立するための NTLMSSP プロトコルの使用をサポートしない

FIPS モードが有効な Active Directory (AD)と Identity Management (IdM) との間で双方向のフォレスト間の信頼を確立すると、New Technology LAN Manager Security Support Provider (NTLMSSP) 認証が FIPS に準拠していないため、失敗します。FIPS モードの IdM は、認証の試行時に AD ドメインコント

ローラーが使用する RC4 NTLM ハッシュを受け入れません。

Bugzilla:2124243

IdM から AD へのレルム間の TGS 要求が失敗します

IdM Kerberos チケットの 特権属性証明書 (PAC) 情報は、Active Directory (AD) でサポートされていない AES SHA-2 HMAC 暗号化で署名されるようになりました。

その結果、IdM から AD へのレルム間 TGS 要求 (双方向の信頼の設定) は、以下のエラーを出して失敗します。

```
Generic error (see e-text) while getting credentials for <service principal>
```

Bugzilla:2060421

FIPS モードで IdM Vault 暗号化および復号化に失敗する

FIPS モードが有効な場合は、OpenSSL RSA-PKCS1v15 パディング暗号化がブロックされます。その結果、現在は IdM が PKCS1v15 パディングを使用してセッションキーをトランスポート証明書でラップするため、Identity Management (IdM) Vault が正しく機能しません。

Bugzilla:2089907

SID のないユーザーは、アップグレード後に IdM にログインできません

IdM レプリカを RHEL 9.2 にアップグレードした後、IdM Kerberos Distribution Center (KDC) は、アカウントにセキュリティー識別子 (SID) が割り当てられていないユーザーに Ticket-Granting Ticket (TGT) を発行できない場合があります。その結果、ユーザーは自分のアカウントにログインできなくなります。

この問題を回避するには、トポロジー内の別の IdM レプリカで IdM 管理者として次のコマンドを実行して SID を生成します。

```
# ipa config-mod --enable-sid --add-sids
```

その後もユーザーがログインできない場合は、Directory Server のエラーログを調べてください。ユーザーの POSIX ID を含めるように ID 範囲を調整する必要がある場合があります。

詳細は、ナレッジベースのソリューション記事 When upgrading to RHEL9, IDM users are not able to login anymore を参照してください。

Jira:RHELPLAN-157939

ドメイン SID の不一致により、移行した IdM ユーザーがログインできない可能性がある

ipa migrate-ds スクリプトを使用して IdM デプロイメントから別のデプロイメントにユーザーを移行する場合、そのユーザーの以前のセキュリティー識別子 (SID) には現在の IdM 環境のドメイン SID がないため、ユーザーが IdM サービスを使用する際に問題が発生する可能性があります。たとえば、これらのユーザーは **kinit** ユーティリティーを使用して Kerberos チケットを取得できますが、ログインできません。この問題を回避するには、ナレッジベースの記事 Migrated IdM users unable to log in due to mismatching domain SIDs を参照してください。

Jira:RHELPLAN-109613

ユーザー PAC を生成する暗号化タイプに互換性がないため、MIT krb5 ユーザーは AD TGT の取得に失敗する

MIT **krb5 1.20** 以降のパッケージでは、デフォルトですべての Kerberos チケットに特権属性証明書 (PAC) が含まれています。MIT Kerberos Distribution Center (KDC) は、PAC で KDC チェックサムを生成するために使用できる最も強力な暗号化タイプを選択します。これは現在、RFC8009 で定義されている **AES HMAC-SHA2** 暗号化タイプです。ただし、Active Directory (AD) はこの RFC をサポートしていません。その結果、AD-MIT クロスレルム設定では、MIT KDC によって生成されたクロスレルム TGT に互換性のない KDC チェックサムタイプが PAC に含まれているため、MIT **krb5** ユーザーは AD チケット認可チケット (TGT) を取得できません。

この問題を回避するには、**/var/kerberos/krb5kdc/kdc.conf** 設定ファイルの **[realms]** セクションで MIT レルムの **disable_pac** パラメーターを **true** に設定します。その結果、MIT KDC は PAC なしでチケットを生成します。これは、AD が失敗したチェックサム検証をスキップし、MIT **krb5** ユーザーが AD TGT を取得できることを意味します。

Bugzilla:2016312

ldap_id_use_start_tls オプションのデフォルト値を使用する場合の潜在的なリスク

ID ルックアップに TLS を使用せずに **ldap://** を使用すると、攻撃ベクトルのリスクが生じる可能性があります。特に、中間者 (MITM) 攻撃は、攻撃者が、たとえば、LDAP 検索で返されたオブジェクトの UID または GID を変更することによってユーザーになりすますことを可能にする可能性があります。

現在、TLS を強制する SSSD 設定オプション **ldap_id_use_start_tls** は、デフォルトで **false** に設定されています。セットアップが信頼できる環境で動作していることを確認し、**id_provider = ldap** に暗号化されていない通信を使用しても安全かどうかを判断してください。注記: **id_provider = ad** および **id_provider = ipa** は、SASL および GSSAPI によって保護された暗号化接続を使用するため、影響を受けません。

暗号化されていない通信を使用することが安全ではない場合は、**/etc/sssd/sssd.conf** ファイルで **ldap_id_use_start_tls** オプションを **true** に設定して TLS を強制します。デフォルトの動作は、RHEL の将来のリリースで変更される予定です。

Jira:RHELPLAN-155168

RHEL 8.6 以前で初期化された FIPS モードの IdM デプロイメントに FIPS モードの RHEL 9 レプリカを追加すると失敗する

FIPS 140-3 への準拠を目的としたデフォルトの RHEL 9 FIPS 暗号化ポリシーでは、RFC3961 のセクション 5.1 で定義されている AES HMAC-SHA1 暗号化タイプのキー派生関数の使用が許可されていません。

この制約は、最初のサーバーが RHEL 8.6 システム以前にインストールされている FIPS モードの RHEL 8 IdM 環境に、FIPS モードの RHEL 9 Identity Management (IdM) レプリカを追加する際の障害となります。これは、AES HMAC-SHA1 暗号化タイプを一般的に使用し、AES HMAC-SHA2 暗号化タイプを使用しない、RHEL 9 と以前の RHEL バージョンの間に共通の暗号化タイプがないためです。

サーバーで次のコマンドを入力すると、IdM マスターキーの暗号化タイプを表示できます。

```
# kadmin.local getprinc K/M | grep -E '^Key:'
```

この問題を回避するには、RHEL 9 レプリカで AES HMAC-SHA1 の使用を有効にします。

```
update-crypto-policies --set FIPS:AD-SUPPORT
```

WARNING

この回避策は FIPS 準拠に違反する可能性があります。

その結果、RHEL 9 レプリカの IdM デプロイメントへの追加が正しく進行します。

RHEL 7 および RHEL 8 サーバー上で欠落している AES HMAC-SHA2 暗号化 Kerberos キーを生成する手順を提供する作業が進行中であることに注意してください。これにより、RHEL 9 レプリカで FIPS 140-3 準拠が達成されます。ただし、Kerberos キー暗号化の設計により、既存のキーを別の暗号化タイプに変換することができないため、このプロセスは完全には自動化されません。唯一の方法は、ユーザーにパスワードの更新を求めることです。

Bugzilla:2103327

SSSD は DNS 名を適切に登録します。

以前は、DNS が正しく設定されていない場合、SSSD は DNS 名の登録の最初の試行で常に失敗していました。この問題を回避するために、この更新では新しいパラメーター **dns_resolver_use_search_list** が提供されます。DNS 検索リストの使用を回避するには、**dns_resolver_use_search_list = false** を設定します。

Bugzilla:1608496

referral mode で起動すると、Directory Server が予期せず終了する

バグにより、Directory Server ではグローバル参照モードが動作しません。**dirsrv** ユーザーとして **refer** オプションを指定して **ns-slapd** プロセスを開始すると、Directory Server はポート設定を無視し、予期せず終了します。**root** ユーザーが SELinux ラベルを変更し、サービスが将来通常モードで開始されないようにプロセスを実行しようとしています。回避策はありません。

Bugzilla:2053204

Directory Server は、/var/lib/dirsrv/slapd-instance_name/ldif/ からのみ LDIF ファイルをインポート可能

RHEL 8.3 以降、Red Hat Directory Server (RHDS) は独自のプライベートディレクトリーを使用し、LDAP サービスに対して **PrivateTmp** systemd ディレクティブがデフォルトで有効になっています。その結果、RHDS は、**/var/lib/dirsrv/slapd-instance_name/ldif/** ディレクトリーからのみ LDIF ファイルをインポートできます。LDIF ファイルが **/var/tmp**、**/tmp**、**/root** などの別のディレクトリーに保存されている場合、インポートは次のようなエラーで失敗します。

```
Could not open LDIF file "/tmp/example.ldif", errno 2 (No such file or directory)
```

この問題を回避するには、以下の手順を実行します。

1. LDIF ファイルを **/var/lib/dirsrv/slapd-instance_name/ldif/** ディレクトリーに移動します。

```
# mv /tmp/example.ldif /var/lib/dirsrv/slapd-instance_name__/ldif/
```

2. **dirsrv** ユーザーがファイルを読み取れるようにする権限を設定します。

```
# chown dirsrv /var/lib/dirsrv/slapd-instance_name/ldif/example.ldif
```

3. SELinux コンテキストを復元します。

```
# restorecon -Rv /var/lib/dirsrv/slapd-instance_name/ldif/
```

詳細については、ソリューション記事 LDAP サービスがホストの /tmp および /var/tmp ディレクトリーにあるファイルにアクセスできない を参照してください。

Bugzilla:2075525

EMS 強制により、FIPS モードで RHEL 9.2+ IdM サーバーを使用した RHEL 7 IdM クライアントのインストールが失敗する

TLS **Extended Master Secret** (EMS) 拡張機能 (RFC 7627) は、FIPS 対応の RHEL 9.2 以降のシステムでの TLS 1.2 接続に必須になりました。これは FIPS-140-3 要件に準拠しています。ただし、RHEL 7.9 以前で利用可能な **openssl** バージョンは EMS をサポートしていません。その結果、RHEL 9.2 以降で実行されている FIPS 対応の IdM サーバーに RHEL 7 Identity Management (IdM) クライアントをインストールすると失敗します。

IdM クライアントをインストールする前にホストを RHEL 8 にアップグレードできない場合は、FIPS 暗号化ポリシーに加えて NO-ENFORCE-EMS サブポリシーを適用して、RHEL 9 サーバーでの EMS 使用の要件を削除することで問題を回避します。

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

この削除は FIPS 140-3 要件に反することに注意してください。その結果、EMS を使用しない TLS 1.2 接続を確立して受け入れることができ、RHEL 7 IdM クライアントのインストールは成功します。

Bugzilla:2220915

11.12. デスクトップ

RHEL 9 にアップグレードすると、Firefox アドオンが無効になります

RHEL8 から RHEL 9 にアップグレードすると、Firefox で以前に有効にしたすべてのアドオンが無効になります。

この問題を回避するには、アドオンを手動で再インストールまたは更新します。その結果、アドオンは予想通りに有効になります。

Bugzilla:2013247

RHEL 9 へのアップグレード後に VNC が実行されていない

RHEL8 から RHEL 9 にアップグレードした後、以前に有効にされていたとしても、VNC サーバーは起動に失敗します。

この問題を回避するには、システムのアップグレード後に **vncserver** サービスを手動で有効にします。

```
# systemctl enable --now vncserver@:port-number
```

その結果、VNC が有効になり、システムが起動するたびに期待どおりに起動します。

Bugzilla:2060308

User Creation 画面が応答しない

グラフィカルユーザーインターフェイスを使用して RHEL をインストールすると、User Creation の画面が応答しなくなります。そのため、インストール中にユーザーを作成するのが困難です。

この問題を回避するには、以下のソリューションのいずれかを使用してユーザーを作成します。

- VNC モードでインストールを実行し、VNC ウィンドウのサイズを変更します。
- インストールプロセスの完了後にユーザーを作成します。

BZ#2122636

11.13. グラフィックインフラストラクチャー

NVIDIA ドライバーが X.org に戻る可能性がある

特定の条件下では、プロプライエタリー NVIDIA ドライバーは Wayland ディスプレイプロトコルを無効にし、X.org ディスプレイサーバーに戻ります。

- NVIDIA ドライバーのバージョンが 470 未満の場合。
- システムがハイブリッドグラフィックスを使用するラップトップの場合。
- 必要な NVIDIA ドライバーオプションを有効にしていない場合。

また、Wayland は有効になっていますが、NVIDIA ドライバーのバージョンが 510 未満の場合には、デスクトップセッションはデフォルトで X.org を使用します。

Jira:RHELPLAN-119001

ナイトライトは、NVIDIA の Wayland では利用できない

システムで独自の NVIDIA ドライバーが有効になっている場合、GNOME の **ナイトライト** 機能は Wayland セッションでは使用できません。NVIDIA ドライバーは、現在 **Night Light** をサポートしていません。

Jira:RHELPLAN-119852

Wayland では X.org 設定ユーティリティーが動作しない

画面を操作するための X.org ユーティリティーは、Wayland セッションでは機能しません。特に、**xrandr** ユーティリティーは、処理、解像度、回転、およびレイアウトへのアプローチが異なるため、Wayland では機能しません。

Jira:RHELPLAN-121049

11.14. WEB コンソール

VNC コンソールが特定の解像度で正しく動作しない

特定のディスプレイ解像度で Virtual Network Computing (VNC) コンソールを使用すると、マウスオフセットの問題が発生したり、インターフェイスの一部しか表示されない場合があります。そのため、VNC コンソールを使用できない場合があります。この問題を回避するには、VNC コンソールのサイズを拡大するか、代わりにコンソールタブのデスクトップビューアーを使用してリモートビューアーを起動します。

Bugzilla:2030836

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

metrics システムロールが、ファクト収集が無効になっていると機能しない

Ansible ファクト収集は、パフォーマンスまたはその他の理由により、環境内で無効になっている場合があります。このような設定では、現時点では **metrics** システムロールを使用することはできません。この問題を回避するには、ファクトキャッシングを有効にしてください。ファクト収集を使用できない場合は、**metrics** システムロールを使用しないでください。

Bugzilla:2078999

firewalld.service がマスクされている場合、firewall RHEL システムロールの使用が失敗する

RHEL システム上で **firewalld.service** がマスクされている場合、**firewall** RHEL システムロールは失敗します。この問題を回避するには、**firewalld.service** のマスクを解除します。

```
systemctl unmask firewalld.service
```

Bugzilla:2123859

環境名でシステムを登録できない

rhc_environment に環境名を指定すると、**rhc** システムロールはシステムの登録に失敗します。回避策として、登録時に環境名の代わりに環境 ID を使用します。

Bugzilla:2187539

11.16. 仮想化

https または ssh 経由での仮想マシンのインストールに失敗する場合がある

現在、**virt-install** ユーティリティーは、https または ssh 接続を介して ISO ソースからゲストオペレーティングシステム (OS)をインストールしようとすると失敗します。たとえば、**virt-install--cdromhttps://example/path/to/image.iso** を使用します。仮想マシンを作成する代わりに、上述の操作は**internal error: process exited while connecting to monitor**(監視への接続中にプロセスが終了しました) というメッセージで予想外に終了します。

同様に、RHEL 9 Web コンソールを使用してゲスト OS をインストールすると失敗し、https または ssh URL を使用すると **Unknown driver 'https'** エラーが発生するか、**Download OS** 機能が表示されます。

この問題を回避するには、ホストに **qemu-kvm-block-curl** および **qemu-kvm-block-ssh** をインストールして、https および ssh プロトコルのサポートをそれぞれ有効にします。別の接続プロトコルまたは別のインストールソースを使用することもできます。

Bugzilla:2014229

仮想マシンで NVIDIA ドライバーを使用すると Wayland が無効になる

現在、NVIDIA ドライバーは Wayland グラフィカルセッションと互換性がありません。これにより、NVIDIA ドライバーを使用する RHEL ゲストオペレーティングシステムは、Wayland を自動的に無効にし、代わりに Xorg セッションを読み込みます。これは主に以下のシナリオで生じます。

- NVIDIA GPU デバイスを RHEL 仮想マシンに渡す場合
- NVIDIA vGPU 仲介デバイスを RHEL 仮想マシンに割り当てる場合

Jira:RHELPLAN-117234

Milan 仮想マシンの CPU タイプは、AMD Milan システムで利用できないことがある

一部の AMD Milan システムでは、Enhanced REP MOVSB (**erms**) および Fast Short REP MOVSB (**fsrm**) 機能フラグがデフォルトで BIOS で無効になっています。したがって、**Milan** CPU タイプは、これらのシステムで利用できない可能性があります。さらに、機能フラグ設定が異なる Milan ホスト間の仮想マシンのライブマイグレーションが失敗する可能性があります。これらの問題を回避するには、ホストの BIOS で **erms** および **fsrm** を手動で有効にします。

Bugzilla:2077767

フェイルオーバー設定のある hostdev インターフェイスは、ホットアンプラグされた後にホットプラグすることはできない

フェイルオーバー設定の **hostdev** ネットワークインターフェイスを実行中の仮想マシン (VM) から削除した後、現在、インターフェイスを同じ実行中の VM に再接続することはできません。

Bugzilla:2052424

フェイルオーバー VF を使用した VM のコピー後のライブマイグレーションが失敗する

現在、VM が仮想機能 (VF) フェイルオーバー機能が有効になっているデバイスを使用している場合、実行中の仮想マシン (VM) のコピー後移行の試行は失敗します。この問題を回避するには、コピー後の移行ではなく、標準の移行タイプを使用します。

Bugzilla:1817965

ライブマイグレーション中にホストネットワークが VF と VM に ping できない

設定済みの仮想機能 (VF) で仮想マシン (仮想 SR-IOV ソフトウェアを使用する仮想マシンなど) のライブマイグレーションを行う場合、仮想マシンのネットワークは他のデバイスに表示されず、**ping** などのコマンドで仮想マシンに到達できません。ただし、移行が終了すると、問題は発生しなくなります。

Bugzilla:1789206

フェイルオーバー virtio NIC には、Windows 仮想マシンで IP アドレスが割り当てられていない

現在、フェイルオーバー virtio NIC のみで Windows 仮想マシンを起動すると、仮想マシンは NIC に IP アドレスを割り当てることができません。したがって、NIC はネットワーク接続を設定できません。現在、回避策はありません。

Bugzilla:1969724

AVX を無効にすると、仮想マシンが起動できなくなる

Advanced Vector Extensions (AVX) をサポートする CPU を使用するホストマシンで、現在、AVX を明示的に無効にして VM を起動しようとすると失敗し、代わりに VM でカーネルパニックが発生します。

Bugzilla:2005173

ネットワークインターフェイスのリセット後に Windows VM が IP アドレスの取得に失敗する

ネットワークインターフェイスの自動リセット後に、Windows 仮想マシンが IP アドレスの取得に失敗することがあります。その結果、VM はネットワークに接続できません。この問題を回避するには、Windows デバイスマネージャーでネットワークアダプタードライバーを無効にしてから再度有効にします。

Bugzilla:2084003

Broadcom ネットワークアダプターが、ライブマイグレーション後に Windows VM で正しく動作しない

現在、Broadcom、Qlogic、Marvell などの Broadcom デバイスファミリーのネットワークアダプターは、Windows 仮想マシン (VM) のライブマイグレーション中にホットアンプラグできません。その結果、移行が完了した後、アダプターが正しく動作しません。

この問題は、Single-root I/O virtualization (SR-IOV) を使用して Windows VM に接続されているアダプターのみに影響します。

Bugzilla:2090712、Bugzilla:2091528、Bugzilla:2111319

vCPU をホットプラグした後、Windows Server 2016 VM が動作を停止することがあります。

現在、Windows Server 2016 ゲストオペレーティングシステムで実行中の仮想マシン (VM) に vCPU を割り当てると、VM が予期せず終了したり、応答しなくなったり、再起動したりするなど、さまざまな問題が発生する可能性があります。

Bugzilla:1915715

多数のキューを使用すると、Windows 仮想マシンで障害が発生することがある

仮想 Trusted Platform Module (vTPM) デバイスが有効で、**マルチキュー virtio-net** 機能が 250 を超えるキューを使用するように設定されている場合、Windows 仮想マシン (VM) が失敗することがあります。

この問題は、vTPM デバイスの制限が原因で発生します。vTPM デバイスには、開いているファイル記述子の最大数に関するハードコーディングされた制限があります。新しいキューごとに複数のファイル記述子が開かれるため、内部の vTPM 制限を超えて VM が失敗する可能性があります。

この問題を回避するには、次の 2 つのオプションのいずれかを選択します。

- vTPM デバイスを有効のままにしますが、使用するキューは 250 未満にします。
- 250 を超えるキューを使用するには、vTPM デバイスを無効にします。

Bugzilla:2020146

NVIDIA パススルーデバイスを備えた VM での冗長エラーメッセージ。

RHEL 9.2 オペレーティングシステムで Intel ホストマシンを使用している場合、パススルー NVDIA GPU デバイスを備えた仮想マシン (VM) で、次のエラーメッセージが頻繁に記録されます。

> Spurious APIC interrupt (vector 0xFF) on CPU#2, should never happen.

ただし、このエラーメッセージは VM の機能には影響しないため、無視してかまいません。詳細については、Red Hat KnoweldgeBase を参照してください。

Bugzilla:2149989

AMD EPYC CPU を搭載したホストで v2v 変換後に一部の Windows ゲストが起動に失敗する

virt-v2v ユーティリティーを使用して、Windows 11 または Windows Server 2022 をゲスト OS として使用する仮想マシン (VM) を変換した後、現在 VM は起動に失敗します。これは、AMD EPYC シリーズ CPU を使用するホストで発生します。

Bugzilla:2168082

ホストで OVS サービスを再起動すると、実行中の VM でネットワーク接続がブロックされることがある

ホストで Open vSwitch (OVS) サービスが再起動またはクラッシュすると、このホストで実行されている仮想マシン (VM) はネットワークデバイスの状態を回復できません。その結果、仮想マシンがパケットを完全に受信できなくなる可能性があります。

この問題は、**virtio** ネットワークスタックで圧縮された virtqueue 形式を使用するシステムのみに影響します。

この問題を回避するには、**virtio** ネットワークデバイス定義で **packed=off** パラメーターを使用して、圧縮された virtqueue を無効にします。圧縮された virtqueue を無効にすると、状況によっては、ネットワークデバイスの状態を RAM から回復できます。

Bugzilla:1947422

VM のシャットダウン後に Nvidia GPU ドライバーが動作を停止します。

RHEL カーネルは、デバイスの電源遷移遅延を PCIe 仕様で要求される遅延にさらに近づけるアップストリーム Linux の変更を採用しました。その結果、GPU のオーディオ機能が原因で、VM のシャットダウン後に一部の Nvidia GPU が動作を停止する可能性があります。

この問題を回避するには、VM から GPU のオーディオ機能の割り当てを解除します。さらに、デバイス割り当て (つまり、IOMMU グループ化) の DMA 分離要件により、バインドはオーディオ機能を **vfio-pci** ドライバーに割り当てます。これにより、GPU 機能が引き続き割り当てられ、正常に機能することが可能になります。

Bugzilla:2178956

nodedev-dumpxml が特定の仲介デバイスの属性を正しく一覧表示しない

現在、**nodedev-dumpxml** は、**nodedev-create** コマンドを使用して作成された仲介デバイスの属性を正しく一覧表示していません。この問題を回避するには、代わりに **nodedev-define** コマンドおよび **nodedev-start** コマンドを使用します。

Bugzilla:2143158

中断されたコピー後の VM 移行の回復が失敗することがある

仮想マシン (VM) のコピー後の移行が中断された後、同じ受信ポートですぐに再開されると、移行は **Address already in use** のエラーで失敗する可能性があります。

この問題を回避するには、コピー後の移行を再開する前に少なくとも 10 秒待つか、移行の回復のために別のポートに切り替えます。

Bugzilla:2178376

virtqemud または libvirtd を再起動した後、virtiofs デバイスをアタッチできない

現在、**virtqemud** サービスまたは **libvirtd** サービスを再起動すると、**virtiofs** ストレージデバイスがホスト上の仮想マシンにアタッチされなくなります。

Bugzilla:2078693

virsh blkiotune --weight コマンドが正しい cgroup I/O コントローラー値を設定できない

現在、**virsh blkiotune --weight** コマンドを使用して VM weight を設定しても、期待どおりに機能しません。このコマンドは、cgroup I/O コントローラーインターフェイスファイルに正しい **io.bfq.weight** 値を設定できません。現時点では回避策はありません。

Jira:RHELPLAN-83423

仮想マシンへの Watchdog カードのホットプラグが失敗する

現在、使用可能な PCI スロットがない場合、実行中の仮想マシン (VM) に Watchdog カードを追加すると、次のエラーが発生して失敗します。

```
Failed to configure watchdog
ERROR Error attempting device hotplug: internal error: No more available PCI slots
```

この問題を回避するには、Watchdog カードを追加する前に VM をシャットダウンします。

Bugzilla:2173584

AMD EPYC CPU で NUMA ノードマッピングが正しく機能しない

QEMU は、AMD EPYC CPU の NUMA ノードマッピングを正しく処理しません。これにより、NUMA ノード設定を使用する場合、これらの CPU を持つ仮想マシン (VM) のパフォーマンスに悪影響が及ぶ可能性があります。さらに、VM は起動時に以下のような警告を表示します。

```
sched: CPU #4's llc-sibling CPU #3 is not on the same node! [node: 1 != 0]. Ignoring dependency.
WARNING: CPU: 4 PID: 0 at arch/x86/kernel/smpboot.c:415 topology_sane.isra.0+0x6b/0x80
```

この問題を回避するには、NUMA ノード設定に AMD EPYC CPU を使用しないでください。

Bugzilla:2176010

VM 移行中の NFS 障害により、移行が失敗してソース仮想マシンのコアダンプが発生する

現在、仮想マシン (VM) の移行中に NFS サービスまたはサーバーがシャットダウンした場合、ソース VM の QEMU は、実行を再開したときに NFS サーバーに再接続できません。その結果、移行に失敗し、ソース VM でコアダンプが開始されます。現在、使用可能な回避策はありません。

Bugzilla:2058982

PCIe ATS デバイスが Windows 仮想マシンで動作しない

Windows ゲストオペレーティングシステムを使用して仮想マシン (VM) の XML 設定で PCIe アドレス変換サービス (ATS) デバイスを設定しても、ゲストが仮想マシンの起動後に ATS デバイスを有効にしません。これは、Windows が現在 **virtio** デバイス上の ATS をサポートしていないためです。

Bugzilla:2073872

AMD SEV-SNP を搭載した仮想マシンで Kdump が失敗する

現在、Secure Nested Paging (SNP) 機能を備えた AMD Secure Encrypted Virtualization (SEV) を使用する RHEL 9 仮想マシン (VM) では kdump が失敗します。

Jira:RHEL-10019

11.17. クラウド環境の RHEL

Nutanix AHV で LVM を使用する RHEL 9 仮想マシンのクローンを作成または復元すると、ルート以外のパーティションが表示されなくなる

Nutanix AHV ハイパーバイザーをホストとする仮想マシン (VM) で RHEL 9 ゲストオペレーティングシステムを実行する場合、スナップショットから VM を復元するか VM をクローンすると、ゲストが論理ボリューム管理 (LVM) を使用している場合は VM 内の非ルートパーティションを消失させることがあります。これにより、以下の問題が発生します。

- スナップショットから仮想マシンを復元すると、仮想マシンは起動できず、緊急モードに入ります。
- クローンを作成して作成した仮想マシンは起動できず、緊急モードに入ります。

これらの問題を回避するには、仮想マシンの緊急モードで以下を行います。

1. 以下の LVM システムデバイスファイルを削除します: **rm/etc/lvm/devices/system.devices**
2. LVM デバイス設定を再作成します。**vgimportdevices -a**
3. 仮想マシンを再起動します。

これにより、クローン化または復元された VM を正しく起動できます。

または、問題が発生しないようにするには、VM のクローンを作成する前、または VM のスナップショットを作成する前に、次の手順を実行します。

1. **/etc/lvm/lvm.conf** ファイルの **use_devicesfile = 0** 行のコメントを外します
2. 仮想マシンを再起動します。

Bugzilla:2059545

ESXi で RHEL 9 ゲストをカスタマイズすると、ネットワークの問題が発生することがある

現在、VMware ESXi ハイパーバイザーでの RHEL 9 ゲストオペレーティングシステムのカスタマイズは、NetworkManager キーファイルでは正しく機能しません。その結果、ゲストがそのようなキーファイルを使用している場合、IP アドレスやゲートウェイなどのネットワーク設定が正しくなくなります。

詳細と回避策は、VMware ナレッジベース を参照してください。

Bugzilla:2037657

cloud-init によってプロビジョニングされ、NFSv3 マウントエントリーで設定された場合、Azure で RHEL インスタンスが起動しない

現在、仮想マシンが **cloud-init** ツールによってプロビジョニングされ、仮想マシンのゲストオペレーティングシステムで **/etc/fstab** ファイルに NFSv3 マウントエントリーがある場合、Microsoft Azure クラウドプラットフォームで RHEL 仮想マシンの起動に失敗します。

Bugzilla:2081114

VMware ホストの RHEL 仮想マシンで静的 IP を設定できない

現在、VMware ホストで RHEL を仮想マシンのゲストオペレーティングシステムとして使用すると、DatasourceOVF 機能は正しく機能しません。これにより、**cloud-init** ユーティリティーを使用して、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。

この問題を回避するには、VMware ナレッジベース を参照してください。

Bugzilla:1750862

11.18. サポート性

IBM Power Systems (Little Endian) で sos report を実行するとタイムアウトする

数百または数千の CPU を搭載した IBM Power Systems (Little Endian) で **sos report** コマンドを実行すると、**/sys/devices/system/cpu** ディレクトリーの膨大なコンテンツを収集する際のプロセッサープラグインはデフォルトのタイムアウトである 300 秒に達します。回避策として、それに応じてプラグインのタイムアウトを増やします。

- 1回限りの設定の場合は、次を実行します。

```
# sos report -k processor.timeout=1800
```

- 永続的な変更を行うには、**/etc/sos/sos.conf** ファイルの **[plugin_options]** セクションを編集します。

```
[plugin_options]
# Specify any plugin options and their values here. These options take the form
# plugin_name.option_name = value
#rpm.rpmva = off
processor.timeout = 1800
```

値の例は 1800 に設定されています。特定のタイムアウト値は、特定のシステムに大きく依存します。プラグインのタイムアウトを適切に設定するには、次のコマンドを実行して、タイムアウトなしで1つのプラグインを収集するために必要な時間を最初に見積もることができます。

```
# time sos report -o processor -k processor.timeout=0 --batch --build
```

Bugzilla:1869561

11.19. コンテナー

古いコンテナーイメージ内で systemd を実行すると動作しない

古いコンテナーイメージ (例:**centos:7**) で systemd を実行しても動作しません。

```
$ podman run --rm -ti centos:7 /usr/lib/systemd/systemd
 Storing signatures
 Failed to mount cgroup at /sys/fs/cgroup/systemd: Operation not permitted
 [!!!!!!] Failed to mount API filesystems, freezing.
```

この問題を回避するには、以下のコマンドを使用します。

```
# mkdir /sys/fs/cgroup/systemd
# mount none -t cgroup -o none,name=systemd /sys/fs/cgroup/systemd
# podman run --runtime /usr/bin/crun --annotation=run.oci.systemd.force_cgroup_v1=/sys/fs/cgroup -
-rm -ti centos:7 /usr/lib/systemd/systemd
```

Jira:RHELPLAN-96940

付録A コンポーネント別のチケットリスト

参考のために、Bugzilla および JIRA チケットのリストをこのドキュメントに記載します。リンクをクリックすると、チケットについて説明したこのドキュメントのリリースノートにアクセスできます。

コンポーネント	チケット
389-ds-base	Bugzilla:2096795、Bugzilla:1859271、Bugzilla:2057070、Bugzilla:2093981、Bugzilla:1132524、Bugzilla:2136610、Bugzilla:2142639、Bugzilla:1878808、Bugzilla:1924569、Bugzilla:1956987、Bugzilla:1952241、Bugzilla:2063140、Bugzilla:2047175、Bugzilla:2053204
Doc-administration-guide	Bugzilla:2075525
NetworkManager	Bugzilla:2134897、Bugzilla:2081302、Bugzilla:2019306、Bugzilla:2128809、Bugzilla:2110307、Bugzilla:2117352、Bugzilla:2029636、Bugzilla:2073512、Bugzilla:2128216、Bugzilla:1894877、Bugzilla:2151040
aardvark-dns	Jira:RHELPLAN-138024
anaconda	Bugzilla:2052938、Bugzilla:2158210、Bugzilla:1991843、Bugzilla:2127100、Bugzilla:2093793、Bugzilla:2107346、Bugzilla:2050140、Bugzilla:1877697、Bugzilla:1914955、Bugzilla:1929105、Bugzilla:1997832、Bugzilla:2125542、Bugzilla:2115783、Bugzilla:2164216、Bugzilla:2163497
ansible-collection-microsoft-sql	Bugzilla:2151282、Bugzilla:2151283、Bugzilla:2151284、Bugzilla:2153428、Bugzilla:2163709
ansible-freeipa	Bugzilla:2127913
bacula	Bugzilla:2089395
bind	Bugzilla:1984982
chrony	Bugzilla:2133754
clevis	Bugzilla:2126533、Bugzilla:2159728、Bugzilla:2159735
cloud-init	Bugzilla:1750862
cockpit	Bugzilla:2207498
cockpit-appstream	Bugzilla:2030836
cockpit-machines	Bugzilla:2173584
conntrack-tools	Bugzilla:2132398

コンポーネント	チケット
crash	Bugzilla:2119685
crypto-policies	Bugzilla:2152635
cyrus-sasl	Bugzilla:1995600
device-mapper-multipath	Bugzilla:2033080、Bugzilla:2011699、Bugzilla:1926147
dnf	Bugzilla:2131288、Bugzilla:2121662、Bugzilla:2122626、Bugzilla:2073510
dnf-plugins-core	Bugzilla:2139326
edk2	Bugzilla:1935497
fapolicyd	Jira:RHEL-192、Bugzilla:2054740、Bugzilla:2070655
firefox	Bugzilla:2013247
firewalld	Bugzilla:2125371、Bugzilla:2077512、Bugzilla:2122678
frr	Bugzilla:2129731、Bugzilla:2129743
gcc	Bugzilla:2110583、Bugzilla:2117632、Bugzilla:2141718
gdm	Bugzilla:2131203
gimp	Bugzilla:2047161
git	Bugzilla:2139379
git-lfs	Bugzilla:2139383
glibc	Bugzilla:2129005、Bugzilla:2155352
gnome-shell-extensions	Bugzilla:2154358、Bugzilla:2160553
gnupg2	Bugzilla:2070722、Bugzilla:2073567
gnutls	Bugzilla:2084161、Bugzilla:2042009
golang	Bugzilla:2133019、Bugzilla:2175173、Bugzilla:2111072、Bugzilla:2092016
grafana	Bugzilla:2116847

コンポーネント	チケット
grafana-pcp	Bugzilla:2116848
grub2	Bugzilla:2026579
grubby	Bugzilla:2127453
gssproxy	Bugzilla:2184333
ipa	Bugzilla:2143224、Bugzilla:2162677、Bugzilla:2084180、Bugzilla:2084166、Bugzilla:2069202、Bugzilla:2094673 、 Bugzilla:2057471、Bugzilla:2124243、Bugzilla:2089907
iproute	Bugzilla:2155604
java-1.8.0-openjdk	Bugzilla:2188023
java-17-openjdk	Bugzilla:2186803、Bugzilla:2186810、Bugzilla:2186806
jmc	Bugzilla:2122401
jmc-core	Bugzilla:1980981
kdump-anaconda-addon	Bugzilla:2017401
kernel	Bugzilla:2153073、Bugzilla:2143850、Bugzilla:1871126、Bugzilla:1871143 、Bugzilla:2075216、Bugzilla:2100606、Bugzilla:2104468、Bugzilla:2111048、Bugzilla:2150284、Bugzilla:2066372、Bugzilla:2107347、Bugzilla:2140899、Bugzilla:2069758、Bugzilla:1613522、Bugzilla:1874182、Bugzilla:1995338、Bugzilla:1570255、Bugzilla:2023416、Bugzilla:2021672、Bugzilla:2027304、Bugzilla:1660337、Bugzilla:1955275、Bugzilla:2142102、Bugzilla:2041690、Bugzilla:2040643、Bugzilla:2167783、Bugzilla:2000616、Bugzilla:2013650、Bugzilla:2132480、Bugzilla:2059545、Bugzilla:1960467、Bugzilla:2005173、Bugzilla:2128610、Bugzilla:2129288 、Bugzilla:2013884、Bugzilla:2149989、Bugzilla:2168603、Bugzilla:2173947、Bugzilla:2178956、Bugzilla:2180665、Jira:RHEL-6496
kexec-tools	Bugzilla:2085347、Bugzilla:2076416 、 Bugzilla:2160676、Bugzilla:2080110、Bugzilla:2139000、Bugzilla:2113873、Bugzilla:2064708、Bugzilla:2065013
keylime	Bugzilla:2150830、Bugzilla:2138167、Bugzilla:2140670、Bugzilla:2142009
kmod	Bugzilla:2103605

コンポーネント	チケット
krb5	Bugzilla:2068535、Bugzilla:2106043、Bugzilla:2060798、Bugzilla:2077450、Bugzilla:2106296、Bugzilla:2060421、Bugzilla:2016312、Bugzilla:2103327
libdnf	Bugzilla:2124480
libnvme	Bugzilla:2139752
libotr	Bugzilla:2086562
libreswan	Bugzilla:2128669
libsepol	Bugzilla:2145224
libssh	Bugzilla:2026449、Bugzilla:2068475
libvirt	Bugzilla:2014487、Bugzilla:2143158、Bugzilla:2078693
libxcrypt	Bugzilla:2034569
llvm-toolset	Bugzilla:2118567
lvm2	Bugzilla:1878893、Bugzilla:2038183
mod_security	Bugzilla:2143211
mysql	Bugzilla:1991500
nfs-utils	Bugzilla:2143747、Bugzilla:2081114
nginx	Bugzilla:2096174
nmstate	Bugzilla:2095207、Bugzilla:2120473、Bugzilla:2044150、Bugzilla:2058292、Bugzilla:2130240、Bugzilla:2162401
nodejs	Bugzilla:2178088
nss	Bugzilla:2091905
nvme-cli	Bugzilla:2139753
nvme-stas	Bugzilla:1893841
open-vm-tools	Bugzilla:2037657

コンポーネント	チケット
openblas	Bugzilla:2112099、Bugzilla:2115737
opencryptoki	Bugzilla:2110314
openscap	Bugzilla:2159286、Bugzilla:2161499
openslp	Bugzilla:2184570
openssh	Bugzilla:2056884
openssl	Bugzilla:2129063、Bugzilla:2188046、Bugzilla:2060044、Bugzilla:1975836、Bugzilla:2168665、Bugzilla:1681178、Bugzilla:1685470
openssl-ibmca	Bugzilla:2110378
osbuild-composer	Bugzilla:2173928
oscap-anaconda-addon	Bugzilla:2165920、Bugzilla:2172264
pacemaker	Bugzilla:2133546、Bugzilla:2125344、Bugzilla:2125337
pam	Bugzilla:2126640
passt	Bugzilla:2131015
pause-container	Bugzilla:2106816
pcp	Bugzilla:2117074
pcs	Bugzilla:2116295、Bugzilla:2112270、Bugzilla:1620043、Bugzilla:1796827、Bugzilla:2092950
pki-core	Bugzilla:1849834
podman	Jira:RHELPLAN-136602、Jira:RHELPLAN-136607、Bugzilla:2119200、Jira:RHELPLAN-136611、Bugzilla:2069279
postgresql	Bugzilla:2128410
powerpc-utils	Bugzilla:2125152
powertop	Bugzilla:2044132
python-blivet	Bugzilla:2103800

コンポーネント	チケット
python-sqlalchemy	Bugzilla:2152649
python3.11	Bugzilla:2127923
python3.11-lxml	Bugzilla:2157708
qemu-kvm	Bugzilla:2116496、Bugzilla:1965079、Bugzilla:1951814、Bugzilla:2060839、Bugzilla:2014229、Bugzilla:2052424、Bugzilla:1817965、Bugzilla:1789206、Bugzilla:2090712、Bugzilla:1915715、Bugzilla:2020146、Bugzilla:1947422、Bugzilla:2178376、Bugzilla:2176010、Bugzilla:2058982
realtime-tests	Bugzilla:2041637
rear	Bugzilla:2172589、Bugzilla:2160748
restore	Bugzilla:1997366
rhel-system-roles	Bugzilla:2131293、Bugzilla:2133858、Bugzilla:2078999、Bugzilla:2119102、Bugzilla:2128843、Bugzilla:2130010、Bugzilla:2130329、Bugzilla:2130344、Bugzilla:2130357、Bugzilla:2133528、Bugzilla:2133930、Bugzilla:2134202、Bugzilla:2137663、Bugzilla:2140795、Bugzilla:2141330、Bugzilla:2143768、Bugzilla:2165175、Bugzilla:2140804、Bugzilla:2126959、Bugzilla:2143816、Bugzilla:2153030、Bugzilla:2153043、Bugzilla:2162782、Bugzilla:2167528、Bugzilla:2168735、Bugzilla:2160152、Bugzilla:1999770、Bugzilla:2123859、Bugzilla:2187539、Bugzilla:2186218
rpm	Bugzilla:2150804、Bugzilla:2111251、Bugzilla:2144005
rsyslog	Bugzilla:2124849、Bugzilla:2127404、Bugzilla:2124488、Bugzilla:2157659
rteval	Bugzilla:2081325
rust	Bugzilla:2123900
s390utils	Bugzilla:2044204、Bugzilla:1932480
samba	Bugzilla:2131993、Jira:RHELDOCS-16612
scap-security-guide	Bugzilla:2158405、Bugzilla:2122325、Bugzilla:2169414、Bugzilla:2105162、Bugzilla:2120978、Bugzilla:2038978
selinux-policy	Bugzilla:2151841、Bugzilla:1972222、Bugzilla:2064274
sos	Bugzilla:2164987、Bugzilla:2134906、Bugzilla:1869561

コンポーネント	チケット
sssd	Bugzilla:1507035、Bugzilla:2087247、Bugzilla:1766490、Bugzilla:2065693、Bugzilla:2056482、Bugzilla:1608496
stratisd	Bugzilla:2039957、Bugzilla:2039955、Bugzilla:2041558
subscription-manager	Bugzilla:2108549、Bugzilla:2163716、Bugzilla:2136694
swig	Bugzilla:2139101
synce4l	Bugzilla:2143264
systemd	Bugzilla:2217931、Bugzilla:2018112
systemtap	Bugzilla:2083727
tang	Bugzilla:2095474、Bugzilla:2188743
tigervnc	Bugzilla:2060308
tomcat	Bugzilla:2160511
toolbox	Bugzilla:2163752
tuna	Bugzilla:2122781、Bugzilla:2121517、Bugzilla:2062865
tuned	Bugzilla:2133815、Bugzilla:2113900
tzdata	Bugzilla:2157982
udisks2	Bugzilla:1983602
unbound	Bugzilla:2070495
usbguard	Bugzilla:2155910、Bugzilla:2042345、Bugzilla:2097419
virt-v2v	Bugzilla:2168082
virtio-win	Bugzilla:1969724、Bugzilla:2084003
vsftpd	Bugzilla:2018284
wsmancli	Bugzilla:2127416
xdp-tools	Bugzilla:2160066

コンポーネント	チケット
その他	Bugzilla:2177782、Jira:RHELPLAN-137505、Jira:RHELPLAN-139125、Bugzilla:2046653、Jira:RHELPLAN-133650、Jira:RHELPLAN-139430、Jira:RHELPLAN-137416、Jira:RHELPLAN-137411、Jira:RHELPLAN-137406、Jira:RHELPLAN-137403、Jira:RHELPLAN-159146、Jira:RHELPLAN-139448、Jira:RHELPLAN-151481、Jira:RHELPLAN-150266、Jira:RHELPLAN-147982、Jira:RHELPLAN-147428、Jira:RHELPLAN-139659、Jira:RHELPLAN-149091、Jira:RHELPLAN-139655、Jira:RHELPLAN-139424、Jira:RHELPLAN-136489、Jira:RHELPLAN-59528、Bugzilla:2209419、Bugzilla:2190123、Jira:RHELPLAN-135600、Jira:RHELPLAN-148303、Bugzilla:2020529、Bugzilla:2030412、Jira:RHELPLAN-103993、Jira:RHELPLAN-122345、Jira:RHELPLAN-27394、Jira:RHELPLAN-27737、Jira:RHELPLAN-148394、Bugzilla:1927780、Jira:RHELPLAN-110763、Bugzilla:1935544、Bugzilla:2089200、Jira:RHELPLAN-15509、Jira:RHELPLAN-99136、Jira:RHELPLAN-103232、Bugzilla:1899167、Bugzilla:1979521、Jira:RHELPLAN-100087、Jira:RHELPLAN-100639、Bugzilla:2058153、Jira:RHELPLAN-113995、Jira:RHELPLAN-98983、Jira:RHELPLAN-131882、Jira:RHELPLAN-137660、Jira:RHELPLAN-139805、Jira:RHELPLAN-147725、Jira:RHELPLAN-153267、Jira:RHELDOCS-16300、Jira:RHELPLAN-157225、Jira:RHELPLAN-157337、Bugzilla:1640697、Bugzilla:1697896、Bugzilla:2047713、Jira:RHELPLAN-96940、Jira:RHELPLAN-117234、Jira:RHELPLAN-119001、Jira:RHELPLAN-119852、Bugzilla:2077767、Bugzilla:2053598、Bugzilla:2082303、Jira:RHELPLAN-121049、Jira:RHELPLAN-157939、Jira:RHELPLAN-109613、Bugzilla:2160619、Bugzilla:2173992、Bugzilla:2185048、Jira:RHELPLAN-83423

付録B 改訂履歴

0.3-8

2024 年 6 月 11 日火曜日、Brian Angelica (bangelic@redhat.com)

- 非推奨の機能 RHELDOCS-18049 (シェルとコマンドラインツール) を追加しました。

0.3-7

2024 年 6 月 11 日火曜日、Brian Angelica (bangelic@redhat.com)

- 既知の問題 RHEL-24847 (シェルとコマンドラインツール) を追加しました。

0.3-6

2024 年 5 月 16 日 (木)、Gabriela Fialová (gfialova@redhat.com)

- 既知の問題 RHEL-10019 (仮想化) を追加しました。

0.3-5

2024 年 4 月 25 日 (木)、Gabriela Fialová (gfialova@redhat.com)

- 機能拡張 BZ#2136610 (Identity Management) を追加しました。

0.3-4

2024 年 4 月 18 日 (木)、Gabriela Fialová (gfialova@redhat.com)

- 機能拡張 RHEL-19142 (ネットワーク) を追加しました。

0.3-3

2024 年 3 月 14 日 (木)、Gabriela Fialová (gfialova@redhat.com)

- 機能拡張 RHEL-18359 (カーネル) を追加しました。
- 既知の問題 RHEL-25967 (カーネル) を追加しました。

0.3-2

2024 年 3 月 4 日 (月)、Gabriela Fialová (gfialova@redhat.com)

- バグ修正 Jira:SSSD-6096 (Identity Management) を追加しました

0.3.1

2024 年 2 月 1 日木曜日、Gabriela Fialová (gfialova@redhat.com)

- 既知の問題 BZ#1834716 (セキュリティー) を追加しました。

0.3-0

2024 年 1 月 12 日金曜日、Marc Muehlfeld (mmuehlfeld@redhat.com)

- 既知の問題 Jira:RHEL-6496 (ネットワーキング) を追加しました。

0.2-9

2023 年 12 月 12 日火曜日、Gabriela Fialová (gfialova@redhat.com)

- テクノロジープレビュー BZ#2162677 (IdM) を追加しました。

0.2-8

2023 年 12 月 7 日木曜日、Lucie Vařáková (lvarakova@redhat.com)

- 新機能 BZ#2044200 (カーネル) を追加しました。

0.2-7

2023 年 11 月 20 日月曜日、Gabriela Fialová (gfialova@redhat.com)

- 機能拡張 BZ#2165827 (Identity Management) を追加しました。

0.2-6

2023 年 11 月 13 日月曜日、Gabriela Fialová (gfialova@redhat.com)

- テクノロジープレビュー JIRA:RHELDOCS-17040 (仮想化) を追加しました。

0.2-5

2023 年 11 月 10 日金曜日、Gabriela Fialová (gfialova@redhat.com)

- RHEL ドキュメントへのフィードバックの提供に関するモジュールを更新しました。

0.2-4

2023 年 11 月 10 日金曜日、Gabriela Fialová (gfialova@redhat.com)

- テクノロジープレビュー JIRA:RHELDOCS-17050 (仮想化) を追加しました。

0.2-3

2023 年 11 月 2 日木曜日、ガブリエラフィアロヴァ (gfialova@redhat.com)

- BZ#2125371 (ネットワーキング) のドキュメントテキストを更新しました。

0.2-2

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 (コンテナー) を追加しました。

0.2-1

2023 年 9 月 25 日、Gabriela Fialová (gfialova@redhat.com)

- 既知の問題 BZ#2122636 (デスクトップ) を追加しました。

0.2-0

2023 年 9 月 13 日、Lenka Špačková (lspackova@redhat.com)

- BZ#2220915 のコマンド形式を修正しました。

0.1-9

2023 年 9 月 8 日、Marc Muehlfeld (mmuehlfeld@redhat.com)

- 非推奨機能のリリースノート JIRA:RHELDOCS-16612 (Samba) を追加しました。

- Red Hat ドキュメントへのフィードバック セクションを更新しました。

0.1-8

2023 年 9 月 5 日、Gabriela Fialová (gfialova@redhat.com)

- 機能拡張 BZ#2075017 (idm_ds) を追加しました。

0.1-7

2023 年 8 月 31 日、Gabriela Fialová (gfialova@redhat.com)

- 既知の問題 BZ#2230431 (plumbers) を追加しました。

0.1-6

2023 年 8 月 29 日、Gabriela Fialová (gfialova@redhat.com)

- 既知の問題 BZ#2220915 (IdM) を追加しました。

0.1-5

2023 年 8 月 25 日、Lucie Vařáková (lvarakova@redhat.com)

- 既知の問題 BZ#2214508 (カーネル) を追加しました。

0.1.4

2023 年 8 月 17 日、Gabriela Fialová (gfialova@redhat.com)

- BZ#2136937 (Plumbers)の機能拡張を追加します。

0.1.3

2023 年 8 月 14 日、Lenka Špačková (lspackova@redhat.com)

- BZ#2128410 の誤字が修正されました。

0.1.2

2023 年 8 月 9 日、Gabriela Fialová (gfialova@redhat.com)

- セキュリティーバグ修正 BZ#2155910 (CS) を更新しました。

0.1.1

2023 年 8 月 7 日、Gabriela Fialová (gfialova@redhat.com)

- 非推奨機能のリリースノート BZ#2214130 (CS) を更新しました。

0.1.0

2023 年 8 月 3 日、Lenka Špačková (lspackova@redhat.com)

- BZ#2142639 および BZ#2119102 の書式を修正しました。
- 概要の改善

0.0.9

2023 年 8 月 2 日、Marc Muehlfeld (mmuehlfeld@redhat.com)

- 非推奨機能のリリースノート BZ#1894877 (NetworkManager) を更新しました

- 非推奨機能のリリースノート BZ#1894877 (NetworkManager) を更新しました。

0.0.8

2023 年 8 月 1 日、Mirek Jahoda (mjahoda@redhat.com)

- バグ修正 BZ#2207498 (RHEL Web コンソール) により、Web コンソールの既知の問題を NBDE に置き換えました。

0.0.7

2023 年 7 月 27 日、Gabriela Fialová (gfialova@redhat.com)

- DDF フィードバックに従って、カーネルの 3 つの機能強化と、コンパイラーと開発ツールに関する 1 つの機能強化について改正しました。

0.0.6

2023 年 7 月 25 日、Gabriela Fialová (gfialova@redhat.com)

- 既知の問題 BZ#2109231 (インストーラー) を追加しました。

0.0.5

2023 年 6 月 22 日、Gabriela Fialová (gfialova@redhat.com)

- 拡張機能 BZ#2087247 (IdM) を追加しました。

0.0.4

2023 年 6 月 8 日、Gabriela Fialová (gfialova@redhat.com)

- 拡張機能 BZ#2190123 (カーネル) を追加しました。

0.0.3

2023 年 6 月 6 日、Gabriela Fialová (gfialova@redhat.com)

- RHELPLAN-159146 (IdM) を追加しました。

0.0.2

2023 年 6 月 5 日、Gabriela Fialová (gfialova@redhat.com)

- KI BZ#2176010 (virt) を追加しました。

0.0.1

2023 年 5 月 10 日、Gabriela Fialová (gfialova@redhat.com)

- Red Hat Enterprise Linux 9.2 リリースノートのリリース。

0.0.0

2023 年 3 月 29 日、Gabriela Fialová (gfialova@redhat.com)

- Red Hat Enterprise Linux 9.2 Beta リリースノートのリリース。