Red Hat Enterprise Linux 9

9.0 リリースノート

Red Hat Enterprise Linux 9.0 リリースノート

Last Updated: 2024-04-19

Red Hat Enterprise Linux 9 9.0 リリースノート

Red Hat Enterprise Linux 9.0 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 9.0 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 をインストールする方法の詳細は、Installation セクション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、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、Red Hat CTO である Chris Wright のメッセージ をご覧ください。

RED HAT ドキュメントへのフィードバック (英語のみ)

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

Jira からのフィードバック送信 (アカウントが必要)

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

第1章 概要

1.1. RHEL 9.0 における主な変更点

セキュリティー
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、FIPS140-3 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 セクションを参照してください。

まだ SHA-1 を必要とするシステムとの互換性の問題の解決策については、次の KCS の記事を参照してください。

- RHEL 9 から RHEL 6 システムへの SSH が機能しない
- SHA-1 で署名されたパッケージはインストールまたはアップグレードできない
- 'server-sig-algs' 拡張機能をサポートしていない SSH サーバーおよびクライアントとの接続に失敗しました

OpenSSLが、バージョン 3.0.1 で提供されるようになりました。このバージョンでは、プロバイダーのコンセプト、新しいバージョニングスキーム、改良された HTTP(S) クライアント、新しいプロトコル、フォーマット、アルゴリズムのサポートが追加されている他、その他多くの改良が加えられています。

システム全体の**暗号化ポリシー**が、最新のセキュアデフォルトを提供するように調整されました。

OpenSSHがバージョン 8.7p1 で配布されており、RHEL 8.5 で配布されているバージョン 8.0p1 と比較して、多くの機能強化、バグ修正、およびセキュリティーの向上が図られています。

SFTP プロトコルは、**OpenSSH** で以前に使用されていた SCP/RCP プロトコルを置き換えます。SFTP は、より予測可能なファイル名の処理を提供するため、リモート側のシェルで **glob(3)** パターンを拡張する必要はありません。

SELinuxのパフォーマンスは、SELinux ポリシーをカーネルにロードする時間、メモリーオーバーヘッド、その他のパラメーターを含め、大幅に改善されました。詳細は、SELinux のパフォーマンスと領域の効率を向上させる ブログ投稿を参照してください。

RHEL 9 は、アップストリームバージョン 1.1 で **fapolicyed** フレームワークを提供します。その他の改善点の中でも、新しい **rules.d/** および **trust.d/** ディレクトリー、**fagenrules** スクリプト、および **fapolicyd-cli** コマンドの新しいオプションを使用できるようになりました。

SCAP セキュリティーガイド (SSG) パッケージはバージョン 0.1.60 で提供されており、デルタ調整、更新されたセキュリティープロファイル、およびその他の改善が導入されています。

詳細は、「セキュリティー」 を参照してください。

署名に SHA-1 を使用することは、DEFAULT 暗号化ポリシーで制限されています。HMAC を除いて、SHA-1 は TLS、DTLS、SSH、IKEv2、DNSSEC、および Kerberos プロトコルでは許可されなくなりました。

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。

Cyrus SASL が Berkeley DB ではなく GDBM を使用するようになり、NSS (Network Security Services) ライブラリーが、信頼データベースの DBM ファイル形式に対応しなくなりました。

/etc/selinux/config ファイルの **SELINUX=disabled** オプションを使用した SELinux の無効化に対応する機能がカーネルから削除されました。**/etc/selinux/config** でのみ SELinux を無効にすると、システムは SELinux が有効化されますが、ポリシーが読み込まれずに開始します。SELinux を無効にする必要がある場合は、**selinux=0** パラメーターをカーネルコマンドラインに追加します。

RHEL 9 と RHEL 8 のセキュリティーに関する主な違いについては、**Considerations in adopting RHEL 9** ドキュメントのSecurityの項を参照してください。

ネットワーク

新しい MultiPath TCP デーモン (mptcpd) を使用して、**iproute2** ユーティリティーを使用せずに、MultiPath TCP (MPTCP) エンドポイントを設定できます。MPTCP サブフローおよびエンドポイントを永続化するには、NetworkManager の dispatcher スクリプトを使用します。

デフォルトで、NetworkManager は鍵ファイルを使用して新しい接続プロファイルを保存するようになりました。**ifcfg** 形式には依然として対応していることに注意してください。

このリリースで導入された機能と既存の機能の変更の詳細については、新機能 - ネットワーキング を参照してください。

WireGuard VPN テクノロジーが、サポートされていないテクノロジープレビューとして利用できるようになりました。詳細は、テクノロジープレビュー - ネットワーキング を参照してください。

teamd サービスおよび **libteam** ライブラリーが非推奨になりました。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。

iptables-nft および **ipset** が非推奨になりました。このようなパッケージには、**iptable**、**ip6tables**、**ebtables**、**arptables** などのユーティリティーが含まれます。**nftable** フレームワークを使用して、ファイアウォールルールを設定します。

非推奨の機能の詳細については、非推奨の機能 - ネットワーキング を参照してください。

network-scripts が削除されました。NetworkManager を使用して、ネットワーク接続を設定します。RHEL の唯一の部分ではない機能の詳細については、**RHEL 9 ドキュメントの採用に関する考慮事項** のネットワーキング セクション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

RHEL 9.0 では、以下の動的プログラミング言語が提供されます。

- Node.js 16
- Perl 5.32

- PHP 8.0
- Python 3.9
- Ruby 3.0

RHEL 9.0 には、以下のバージョン制御システムが同梱されています。

- Git 2.31
- subversion 1.14

以下の Web サーバーには、RHEL 9.0 が同梱されています。

- Apache HTTP サーバー 2.4.51
- nginx 1.20

以下のプロキシーキャッシュサーバーを使用できます。

- Varnish Cache 6.6
- Squid 5.2

RHEL 9.0 は、以下のデータベースサーバーを提供します。

- MariaDB 10.5
- MySQL 8.0
- PostgreSQL 13
- Redis 6.2

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

コンパイラーおよび開発ツール
システムツールチェーン
RHEL 9.0 では、以下のシステムツールチェインコンポーネントを利用できます。

- GCC 11.2.1
- glibc 2.34
- binutils 2.35.2

RHEL 9 システムツールチェーンコンポーネントには、POWER10 のサポートが含まれています。

パフォーマンスツールおよびデバッガー
RHEL 9.0 では、以下のパフォーマンスツールおよびデバッガーが利用できます。

- GDB 10.2
- Valgrind 3.18.1
- SystemTap 4.6

- Dyninst 11.0.0
- elfutils 0.186

パフォーマンス監視ツール
RHEL 9.0 では、以下のパフォーマンス監視ツールが利用できます。

- PCP 5.3.5
- Grafana 7.5.11

コンパイラーツールセット
RHEL 9.0 では、以下のコンパイラーツールセットを使用できます。

- LLVM Toolset 13.0.1
- Rust Toolset 1.58.1
- Go Toolset 1.17.7

詳細な変更は、「コンパイラーおよび開発ツール」 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。

詳細は、OpenJDK のドキュメント を参照してください。

Java ツール
RHEL 9.0 では、以下の Java ツールが利用できます。

- Maven 3.6
- Ant 1.10

詳細は、「コンパイラーおよび開発ツール」 を参照してください。

デスクトップ
GNOME 環境は、GNOME 3.28 から GNOME 40 に更新され、多くの新機能が追加されました。

X.org ディスプレイサーバーは非推奨になり、今後の RHEL のメジャーリリースで削除される予定です。ほとんどの場合、デフォルトのデスクトップセッションは **Wayland** セッションになりました。

NVIDIA ドライバーを使用する場合は、ドライバー設定が Wayland に対応していると、デスクトップセッションはデフォルトで Wayland ディスプレイプロトコルを選択するようになりました。以前の RHEL リリースでは、NVIDIA ドライバーが常に Wayland を無効にしていました。

PipeWireサービスは、すべてのオーディオ出力と入力を管理するようになりました。**PipeWire** は、一般的な使用例では **PulseAudio**サービスを、専門的な使用例では **JACK** サービスを置き換えます。

詳細は、「デスクトップ」 を参照してください。

仮想化
RHEL 9 では、**libvirt** ライブラリーは、ホスト上の個々の仮想化ドライバーセットを処理するモジュラーデーモンを使用します。これにより、リソース負荷の最適化や監視など、仮想化ドライバーに関連するさまざまなタスクをきめ細かくすることができます。

QEMU エミュレーターが、Clang コンパイラーを使用して構築されるようになりました。これにより、RHEL 9 KVM ハイパーバイザーで、多くの高度なセキュリティー機能およびデバッグ機能を使用できるようになります。この機能の 1 つが SafeStack です。これにより、RHEL 9 でホストされている仮想マシンが、ROP (Return-Oriented Programming) に基づく攻撃に対して大幅にセキュアになります。

さらに、Virtual Trusted Platform Module (vTPM) が完全にサポートされるようになりました。vTPM を使用すると、TPM 仮想暗号化プロセッサーを VM に追加できます。これは、暗号化キーの生成、保存、および管理に使用できます。

最後に、**virtiofs** 機能を実装し、RHEL 9 ホストと仮想マシン間でファイルをより効率的に共有するために使用できます。

このリリースで導入された仮想化機能の詳細については、「仮想化」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

- 以下のアーキテクチャーで、RHEL 8.6 から RHEL 9.0 へ:
  - 64 ビット Intel
  - 64 ビット AMD
  - 64 ビット 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 を使用するシステムの RHEL 8.6 から RHEL 9.0 へ

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 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。

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:アプリケーションの互換性ガイド を参照してください。

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 9.0 は、カーネルバージョン 5.14.0 とともに配布されます。これは、最低限必要なバージョンで次のアーキテクチャーのサポートを提供します。

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

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

第3章 RHEL 9 のコンテンツの配布

3.1. インストール

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

- インストール ISO:BaseOS リポジトリーおよび AppStream リポジトリーが含まれ、リポジトリーを追加せずにインストールを完了できる、フル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 で使用可能なアプリケーションストリーム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:アプリケーションの互換性ガイド を参照してください。

3.4. YUM/DNF を使用したパッケージ管理

Red Hat Enterprise Linux 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 9.0 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

Anaconda は、Satellite のキックスタートインストールでマシンのプロビジョニングの rhsm をサポートします。

以前は、マシンのプロビジョニングは、Red Hat Satellite へのキックスタートインストール用のカスタム **%post** スクリプトに依存していました。この **%post** スクリプトは、カスタム satellite の自己署名証明書をインポートし、マシンを登録し、サブスクリプションを割り当て、リポジトリーに存在するパッケージをインストールします。

RHEL 9 では、**rhsm** コマンドを使用してマシンのプロビジョニングに、Satellite サポートが追加されました。**rhsm** コマンドは、システムの登録、RHEL サブスクリプションの割り当て、Satellite インスタンスからのインストールなど、すべてのプロビジョニングタスクに利用できるようになりました。

(BZ#1951709)

RHEL は静的ホスト名として localhost をサポートします

RHEL 9 以降、**/etc/hostname** で静的ホスト名として **localhost** を設定することが有効になります。この場合、NetworkManager は DHCP または逆引き DNS ルックアップを通じて一時的なホスト名を取得しようとしません。

(BZ#2190045)

ライセンス、システム、およびユーザー設定の画面が、標準インストール後に無効になりました。

RHEL ユーザーは、**gnome-initial-setup** 画面およびログイン画面に先立ち、ライセンス、システム (サブスクリプションマネージャー)、および ユーザー設定を設定していました。今回の更新で、ユーザーエクスペリエンスを向上させるために、デフォルトで初期セットアップ画面が無効になりました。

ユーザー作成またはライセンス表示の初期セットアップを実行する必要がある場合は、要件に基づいて以下のパッケージをインストールします。

1. 初期セットアップパッケージをインストールします。

```
# dnf install initial-setup initial-setup-gui
```

2. システムの次回の再起動時に、初期セットアップを有効にします。

```
# systemctl enable initial-setup
```

3. システムを再起動して、初期設定を表示します。

キックスタートを使用したインストールでは、**initial-setup-gui** を packages セクションに追加し、**initial-setup** サービスを有効にします。

```
firstboot --enable
%packages
@^graphical-server-environment
```

```
initial-setup-gui
%end
```

(BZ#1878583)

Anaconda は、対話型インストールでネットワークを自動的にアクティブにする

以前のバージョンは、キックスタートまたは起動オプションでネットワークをアクティブにせずに対話型インストールを実行すると、ユーザーがネットワークスポークでネットワークを手動でアクティブにする必要がありました。今回の更新で、Anaconda が、ユーザーがネットワークスポークにアクセスして手動でアクティベートする必要なく、ネットワークを自動的にアクティベートするようになりました。

注記

この更新では、キックスタートインストールおよび **ip=** 起動オプションを使用したインストールのインストールエクスペリエンスは変更されません。

(BZ#1978264)

Image Builder がファイルシステム設定に対応しました。

今回の機能拡張により、Blueprint でカスタムファイルシステム設定を指定でき、必要なディスクレイアウトでイメージを作成できるようになりました。したがって、デフォルト以外のレイアウトを持つことで、セキュリティーベンチマーク、既存設定との一貫性、パフォーマンス、およびディスク不足エラーに対する保護に関してメリットが得られます。

Blueprint でファイルシステム設定をカスタマイズするには、以下のカスタマイズを設定します。

```
[[customizations.filesystem]]
mountpoint = "MOUNTPOINT"
size = MINIMUM-PARTITION-SIZE
```

注記

ブループリントにファイルシステムのカスタマイズを追加すると、ファイルシステムは LVM パーティションに変換されます。

(BZ#2011448)

root アカウントのロック および パスワードを使用した root の SSH ログインを許可 への新しいオプション

RHEL グラフィカルインストールの root パスワード設定画面に以下の新しいオプションが追加されました。

- root アカウントをロックします。このオプションを使用して、マシンへの root アクセスをロックします。
- root の SSH ログインを password で許可します。このオプションを使用して、パスワードベースの SSH root ログインを有効にします。

パスワードベースの SSH root ログイン を有効にするには、インストールプロセスを開始する前に、キックスタートファイルに以下の行を追加します。

```
%post
echo "PermitRootLogin yes" > /etc/ssh/sshd_config.d/01-permitrootlogin.conf
%end
```

(BZ#1940653)

Image Builder が起動可能なインストーライメージの作成に対応

この機能拡張により、Image Builder を使用して、ルートファイルシステムを含む **tarball** ファイルで設定されるブート可能な ISO イメージを作成することができます。その結果、起動可能な ISO イメージを使用して、**tarball** ファイルシステムをベアメタルシステムにインストールすることができます。

(BZ#2019318)

4.2. RHEL FOR EDGE

RHEL for Edge は、デフォルトで Greenboot ビルトインヘルスチェックをサポートするようになりました

この更新により、RHEL for Edge **Greenboot** には、再起動中にハードウェアがハングアップまたはフリーズしないようにする **watchdog** 機能を備えたビルトインヘルスチェックが含まれるようになりました。これにより、次の機能を利用できます。

- **watchdogs** ハードウェアユーザーがビルトインヘルスチェックを簡単に導入できるようになります
- ビルトイン OS コンポーネントに価値を提供するデフォルトのヘルスチェックのセット
- **watchdog** がデフォルトのプリセットとして表示されるようになるため、この機能を簡単に有効または無効にできます。
- すでに利用可能なヘルスチェックに基づいてカスタムヘルスチェックを作成する機能。

(BZ#2083036)

RHEL 9 は rpm-ostree v2022.2 を提供します。

RHEL 9 は、**rpm-ostree** バージョン v2022.2 とともに配布され、複数のバグ修正と機能強化が提供されます。主な変更点は、以下のとおりです。

- 新しい **--append-if-missing** フラグおよび **--delete-if-present** kargs フラグを使用することで、カーネル引数を idempotent 形式で更新できるようになりました。
- DNF の **Count Me** 機能が、すべてのリポジトリークエリーでデフォルトで完全に無効になり、対応する **rpm-ostree-countme.timer** ユニットおよび **rpm-ostree-countme.service** ユニットによってのみトリガーされるようになりました。countme を参照してください。
- 後処理ロジックで、**user.ima** IMA 拡張アトリビュートを処理できるようになりました。**xattr** 拡張属性が見つかると、最終的な **OSTree** パッケージコンテンツの **security.ima** に自動的に変換されます。
- **treefile** には、新しい **repo-packages** 項目があります。これを使用して、一連のパッケージを特定のリポジトリーに固定できます。

(BZ#1961324)

RHEL 9 は OSTree v2021.2 を提供します

RHEL 9 は、**OSTree** パッケージバージョン v2021.2 とともに配布されており、複数のバグ修正と機能拡張が提供されています。主な変更点は、以下のとおりです。

- 新しい ostree-rs-ext プロジェクトで使用されるファイルを書き込む新しい API で、tarball からのインポートが改善されました。
- **rofiles-fuse** は、**xattrs** 拡張属性を処理するようになりました。注記:**rofiles-fuse** は非推奨とみなされています。#2281 を参照してください。
- **introspection** API およびテストの改善

(BZ#1961254)

rpm-ostree rebase ツールは、RHEL8 から RHEL 9 へのアップグレードをサポートします

この機能拡張により、**rpm-ostree rebase** ツールを使用して RHEL8 システムを RHEL 9 にアップグレードできます。RHEL8 の最新の更新から RHEL 9 の最新の更新までの Edge アップグレード用の RHEL のデフォルトパッケージセットを完全にサポートします。

(BZ#2082306)

4.3. サブスクリプションの管理

subscription-manager syspurpose で統合されたシステム目的のコマンド

以前は、システムの目的属性を設定するコマンドとして、**syspurpose** と **subscription-manager** がありました。1 つのモジュールですべてのシステムの目的属性を統一するため、subscription-manager の **addons** コマンド、**role** コマンド、**service-level** コマンド、および **usage** コマンドはすべて、新しいサブモジュール **subscription-manager syspurpose** に移動しました。

新規サブモジュール外に存在する **subscription-manager** コマンドは非推奨になります。RHEL 9 では、**syspurpose** コマンドラインツールを提供する別のパッケージ (**python3-syspurpose**) が削除されました。

この更新により、subscription-manager の 1 つのコマンドを使用して、すべてのシステムの目的属性を表示、設定、および更新する方法が統一されました。これにより、既存のシステムの目的のコマンドがすべて、新しいサブコマンドとして利用可能な同等のバージョンに置き換わります。たとえば、**subscription-manager role --set SystemRole** は **subscription-manager syspurpose role --set SystemRole** になります。

新しいコマンド、オプション、およびその他の属性の詳細は、**subscription-manager** の man ページの **SYSPURPOSE OPTIONS** セクションを参照してください。

(BZ#1898563)

4.4. ソフトウェア管理

RHEL 9 が RPM 4.16 を提供

RHEL 9 には、RPM バージョン 4.16 が同梱されています。バージョン 4.14 への主なバグ修正および機能強化は、以下のとおりです。

- 以下の主要機能を含む新しい SPEC の機能
  - 高速なマクロベースの依存関係ジェネレータ
  - 動的なビルド依存関係を生成できる **%generate_buildrequires** セクション

    - メタ (順不同) な依存関係
    - パッケージビルドの並列性向上
    - 式でのネイティブなバージョン比較
    - チルダとは異なるカレットバージョンの演算子
    - **%elif**、**%elifos**、および **%elifarch** ステートメント
    - オプションの自動パッチとソースのナンバリング
    - **%autopatch** がパッチの範囲を受け入れる
    - **%patchlist** および **%sourcelist** セクション
    - ビルド時のヘッダーデータの UTF-8 検証の強制
- rpm のデータベースは、**sqlite** ライブラリーをベースになりました。**BerkeleyDB** データベースに対する読み取り専用のサポートは、移行および照会の目的で保持されています。
- トランザクションに関する監査ログイベントを発行するための新しい **rpm-plugin-audit** プラグイン (以前は RPM 自体に組み込まれていた)

(JIRA:RHELPLAN-80734)

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

rpm パッケージの今回の更新で、**fapolicyd** フレームワークと RPM データベースを統合する新しい RPM プラグインが導入されました。プラグインは、RPM トランザクション中にインストール済みおよび変更されたファイルについて **fapolicyd** に通知します。これにより、**fapolicyd** が整合性チェックに対応するようになりました。

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

(BZ#1942549)

RPM が、EdDSA 公開鍵アルゴリズムに対応するようになりました。

この機能強化により、**rpm** コマンドは、EdDSA 公開鍵アルゴリズムを使用した署名鍵に対応します。その結果、EdDSA を使用して生成された署名鍵が、パッケージの署名および検証に使用できるようになりました。

ただし、EdDSA を使用した署名鍵がサポートされるようになりました。RSA は、GnuPG では引き続きデフォルトの公開鍵アルゴリズムになります。

(BZ#1962234)

RPM は、Zstandard (zstd) 圧縮アルゴリズムをサポートするようになりました

この機能拡張により、デフォルトの RPM 圧縮アルゴリズムが Zstandard (**zstd**) に切り替わりました。その結果、ユーザーはより高速なパッケージインストールの恩恵を受けることができます。これは、大規模なトランザクション中に特に顕著になる可能性があります。

(JIRA:RHELPLAN-117903)

新しい DNF オプション exclude_from_weak_autodetect および exclude_from_weak

この機能拡張により、デフォルトの DNF 動作は、不要な弱い依存関係をインストールしません。この動作を変更するには、次の新しいオプションを使用します。

- **exclude_from_weak_autodetect**
  有効にすると、**exclude_from_weak_autodetect** オプションは、システムにインストールされているパッケージの満たされていない弱い依存関係 (推奨: または補足:) を自動検出します。その結果、これらの弱い依存関係のプロバイダーは弱い依存関係としてインストールされませんが、プルされると、通常の依存関係としてインストールされます。デフォルト値は **true** です。

- **exclude_from_weak**
  有効にすると、**exclude_from_weak** オプションは、弱い依存関係としてパッケージをインストールすることを防ぎます (推奨: または補足:)。パッケージは、パッケージ名または glob のいずれかで指定し、コンマで区切ることができます。デフォルト値は **[]** です。

(BZ#2005305)

RHEL 9 は libmodulemd2.13.0 を提供

RHEL 9 には、**libmodulemd** パッケージのバージョン 2.13.0 が同梱されています。バージョン 2.9.4 への主なバグ修正および機能強化は、以下のとおりです。

- モジュールからモジュール化解除されたパッケージをリストから除外するサポートを追加
- **modulemd-validator** ツールの新しい **--type** オプションを使用して、**modulemd-packager- v3** ドキュメントを検証するためのサポートを追加。
- 整数値を解析する機能強化
- 各種の **modulemd-validator** の問題を修正

(BZ#1984403)

4.5. シェルおよびコマンドラインツール

ブラケットの貼り付けが、デフォルトで bash で有効になりました。

bash の **readline** ライブラリーバージョン 8.1 が利用できるようになりました。これにより、デフォルトでブラケットの貼り付けモードが有効になります。ターミナルにテキストを貼り付けると、**bash** でテキストが強調表示されます。貼り付けたコマンドを実行するには、**Enter** キーを押す必要があります。ブラケットの貼り付けモードは、悪意のあるコマンドを誤って実行しないようにデフォルト設定です。

特定ユーザーの括弧付きペーストモードを無効にするには、以下の行を ~/**.inputrc** に追加します。

```
set enable-bracketed-paste off
```

すべてのユーザーに対して括弧で囲まれた貼り付けモードを無効にするには、次の行を /**etc/inputrc** に追加します。

```
set enable-bracketed-paste off
```

括弧で囲まれた貼り付けモードを無効にすると、コマンドは貼り付け時に直接実行されるため、**Enter** キーを押してコマンドを確認する必要はありません。

(BZ#2079078)

RHEL 9 には powerpc-utils1.3.9 が含まれています

RHEL 9 は、**powerpc-utils** パッケージバージョン 1.3.9 を提供します。バージョン 1.3.8 への主なバグ修正および機能強化は、以下のとおりです。

- **drmgr** でログサイズが 1MB に増えました。
- システムの起動時に **HCIND** アレイサイズが修正されました。
- **hcngr** で HNV 接続に **autoconnect-slaves** を実装しました。
- **hcngr** で、HNV ボンディングリスト接続が改善されました。
- **hcngr** で **util-linux** の **hexdump** を使用します。
- NetworkManager で **hcn-init.service** を起動します。
- **ofpathname** でマルチパスの論理 FC ルックアップを修正しました。
- **ofpathname** でパーティションを持つ論理ルックアップに対する OF を修正しました。
- 5 パスを超えるマルチパスデバイスのブートリストを修正しました。
- **ofpathname** の l2of_vd() に、 **devpart** の部分文字列が欠落した抽出を追加しました。
- **lpamumascore** が導入されました。
- **drmgr** で **index operation** による削除を修正しました。
- **ofpathname** で、**SYS_PATH** の定義を **l2of_vs()** から **l2of_scsi()** に移動しました。
- partstat でセキュリティーを強化する **-x** オプションが追加されました。
- **lparstat** man ページの **nroff** の警告およびエラーを修正しました。
- **drmgr** で NUMA ベースの LMB 削除を実装しました。
- **hcngr** で **udev** 名が変更された **ofpathname** 競合を修正しました。
- **hcngr** でボンディングインターフェイスの状態を確認する場合は、**NetworkManager nmcli** を使用します。
- **NetworkManager nmcli** を使用して、システムの起動時に HNV が存在しない場合にボンドインターフェイスを削除します。

(BZ#1873868)

RHEL 9 は opal-prd6.7.1 が同梱されています

opal-prd パッケージバージョン 6.7.1 は、以前に利用可能だったバージョン 6.6.3 に比べて、次の注目すべきバグ修正と機能拡張を提供します。

- **xscom OPAL** 呼び出しによる **xscom** エラーログの問題を修正しました。
- **DEBUG** ビルドでデッドロックが修正されました。
- **fast-reboot** が **core/platform** で失敗した場合は、**full_reboot** にフォールバックします。

- **core/cpu** の **next_ungarded_primary** が修正されました。
- Self-Boot Engine (SBE) におけるレートリミットタイマーの要求とタイマーの状態が改善されました。

(BZ#1869560)

RHEL 9 は lsvpd 1.7.12 を提供します

RHEL 9 には、**lsvpd** パッケージバージョン 1.7.12 が同梱されています。バージョン 1.7.11 への主なバグ修正および機能強化は、以下のとおりです。

- **sysvpd** の UUID プロパティーが追加されました。
- **NVMe** ファームウェアバージョンが改善されました。
- PCI デバイスメーカー解析ロジックを修正
- **lsvpd** 設定ファイルに **recommends clause** を追加

(BZ#1869564)

ppc64-diag バージョン 2.7.7 が利用可能

ppc64-diag パッケージバージョン 2.7.7 は、RHEL 9 で提供されます。バージョン 2.7.6 への主なバグ修正および機能強化は、以下のとおりです。

- ユニットテストケースが改善されました。
- **sysvpd** の UUID プロパティーが追加されました。
- **rtas_errd** サービスは、Linux コンテナーでは実行されません。
- 廃止されたロギングオプションは、**systemd** サービスファイルでは利用できなくなりました。

(BZ#1869567)

RHEL 9 には Fetchmail6.4.24 が含まれています

RHEL 9 には、**fetchmail** パッケージのバージョン 6.4.24 が同梱されています。 **Fetchmail** は、リモートメールの取得および転送ユーティリティーです。

詳細は以下を参照してください。

- **/usr/share/doc/fetchmail/NEWS** ファイル
- **fetchmail(1)** man ページ
- 設定を変更する必要がある場合は、SSL 関連の情報の **/usr/share/doc/fetchmail/README.SSL** ファイル。

(BZ#1999276)

RHEL 9 には Eigen3.4 が含まれています

RHEL 9 は、**eigen3** パッケージバージョン 3.4 とともに配布されます。 **eigen 3.4** は、リニア algebra 用の C++ テンプレートライブラリーで、POWER10 マトリックスの多重化サポート手順に対応するようになりました。

これにより、**Eigen 3.4** のユーザーは、POWER10 システムで最適化された線形化ルーバ計算を実行できます。

(BZ#2032423)

RHEL 9 で cdrskin パッケージが追加されました。

RHEL 9 では、CD、DVD、または BD メディアにデータを書き込みするための **cdrskin** パッケージが導入されました。**cdrskin** パッケージは、RHEL 9 で利用できない **wodim** パッケージから **cdrecord** 実行可能ファイルの代替を提供します。

cdrskin パッケージには、以下が含まれます。

- 光学メディア上のデータの空白、フォーマット、および書き込み。
- CD のマルチセッション。
- 上書き可能な DVD+RW、DVD-RW、DVD-RAM、BD-RE の ISO-9660 マルチセッション。

cdrskin パッケージは、**cdrskin** バイナリーへのシンボリックリンクとして **cdrecord** コマンドを提供するため、ユーザースクリプトを変更する必要はありません。機能の全セットは、**cdrskin(1)** man ページを参照してください。

(BZ#2015861)

redhat.rhel_mgmt Ansible コレクションは、RHEL 9 リリースでサポートされます。

この更新により、Intelligent Platform Management Interface (**IPMI**) Ansible モジュールがサポートされます。**IPMI** は、ベースボード管理コントローラー (BMC) デバイスと通信するための一連の管理インターフェイスの仕様です。**IPMI** モジュール (**ipmi_power** および **ipmi_boot**) は、**redhat.rhel_mgmt** コレクションで利用できます。このコレクションには、**ansible-collection-redhat-rhel_mgmt** パッケージをインストールすることでアクセスできます。

(BZ#2023381)

RHEL 9 で util-linux-core パッケージが追加されました。

util-linux パッケージに加えて、RHEL 9 は、インストールされたパッケージのサイズが重要な機能であるシナリオ (buildroots、特定のコンテナー、ブートイメージなど) 用の **util-linux-core** サブパッケージを提供します。

util-linux-core サブパッケージには、**util-linux** ユーティリティーのサブセットが含まれています。これは、**mount** ユーティリティーなどの Linux システムを起動するために必要です。

util-linux-core サブパッケージには、外部の依存関係は含まれません。たとえば、PAM ライブラリーに依存するため、ログインユーティリティーは利用できません。

インストールなどの標準のユースケースでは、標準の **util-linux** パッケージを使用します。**util-linux** パッケージは **util-linux-core** に依存しています。つまり、**util-linux** をインストールすると、**util-linux-core** が自動的にインストールされます。

(BZ#2079313)

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。

(BZ#2136937)

4.6. インフラストラクチャーサービス

s-nail が mailx を置き換え

s-nail メール処理システムが、**mailx** ユーティリティーに置き換わりました。**s-nail** ユーティリティーは **mailx** と互換性があり、新機能が数多く追加されました。**mailx** パッケージはアップストリームで維持されなくなりました。

(BZ#1940863)

TuneD2.18 が利用可能です

RHEL 9 には TuneD バージョン 2.18 が同梱されています。バージョン 2.16 への主な変更点は、以下のとおりです。

- **net** プラグイン: **txqueuelen** チューニングのサポートが追加されました。
- **disk** プラグイン: NVMe ディスクチューニングのサポートが追加されました。
- **tuned-gui** バグ修正。

(BZ#2003838)

RHEL 9 では mod_security_crs 3.3 が提供されます。

RHEL 9 には、**mod_security_crs** パッケージバージョン 3.3 が同梱されています。主なバグ修正と機能拡張は、以下のとおりです。

- **libinjection** が導入されました。
- ファイル名が ~ で終わるブロックされたバックアップファイル。
- 新しい **LDAP** インジェクションおよび **HTTP** 分割ルールが追加されました。
- 制限付きの拡張機能に **.swp** が追加されました。
- 攻撃分類に、CAPEC (Common Attack Pattern Enumeration and Classification) タグを追加しました。
- **Nuclei**、**WFuzz**、および **ffuf** の脆弱性スキャナーの検出サポートが追加されました。
- 変数が小文字 (**modsec3 behavior fix**) に改善されました。
- 初期化されていない変数、文字列の連結、および globing パターンを介した Unix RCE バイパス技術の検出に対応しました。

- 古いルールタグを削除:**WASCTC**、**OWASP_TOP_10**、**OWASP_AppSensor/RE1**、および **OWASP_CRS/FOO/BAR**。**OWASP_CRS** および **attack-type** は、引き続き **mod_security_crs** パッケージに同梱されています。

- **crs-setup.conf** 変数の形式 **tx.allowed_request_content_type** が、他の変数と同じになるように変更になりました。変数が上書きされた場合は、**crs-setup.conf** ファイルの新しい区切り文字の例を参照してください。

(BZ#1947962)

RHEL 9 が chrony 4.1 を提供

RHEL 9 には、**chrony** バージョン 4.1 が同梱されます。バージョン 3.5 への主なバグ修正および機能強化は、以下のとおりです。

- ネットワークタイムセキュリティー (NTS) 認証のサポートが追加されました。詳細は、chrony における Network Time Security (NTS) の概要を参照してください。

- デフォルトでは、認証されたネットワークタイムプロトコル (NTP) ソースは、非認証の NTP ソースで信頼されます。元の動作を復元するには、**chrony.conf** ファイルに **autselectmode ignore** 引数を追加します。

- **RIPEMD** キー (**RMD128**、**RMD160**、**RMD256**、**RMD320**) による認証のサポートは利用できなくなりました。

- NTPv4 パケットにおける長い非標準 MAC のサポートが利用できなくなりました。**chrony 2.x** (**MD5/SHA1 以外の** 鍵) を使用している場合は、**バージョン 3** オプションで **chrony** を設定する必要があります。

また、以下は、RHEL 8 バージョンの **chrony** とは異なります。

- **seccomp** フィルターはデフォルトで有効になっています (**-F2** は **/etc/sysconfig/chronyd** に設定されています)。**seccomp** フィルターは **mailonchange** ディレクティブと競合します。**/etc/chrony.conf** に **mailonchange** ディレクティブがある場合は、**/etc/sysconfig/chronyd** から **-F 2** の設定を削除します。

(BZ#1961131)

4.7. セキュリティー

システム全体の crypto-policies のセキュリティーが強化されました。

今回の更新で、システム全体の暗号化ポリシーが、最新のセキュアデフォルトを提供するように調整されました。

- すべてのポリシーで、TLS 1.0、TLS 1.1、DTLS 1.0、RC4、Camellia、DSA、3DES、および FFDHE-1024 が無効になりました。

- LEGACY で、RSA キーの最小サイズと Diffie-Hellman パラメーターの最小サイズが増加しました。

- HMAC (Hash-based Message Authentication Code) での SHA-1 の使用を除く、SHA-1 を使用した無効な TLS アルゴリズムおよび SSH アルゴリズム

シナリオにおいて、無効化されたアルゴリズムと暗号の一部を有効にする必要がある場合は、カスタムポリシーまたはサブポリシーを使用します。

(BZ#1937651)

RHEL 9 は OpenSSL 3.0.1 を提供します。

RHEL 9 は、アップストリームバージョン 3.0.1 で **openssl** パッケージを提供します。これには、以前のバージョンに改善されたバグ修正が数多く含まれます。以下は、主な変更点です。

- 新しい Provider 概念が追加されました。プロバイダーは一連のアルゴリズムで、異なるアプリケーションに異なるプロバイダーを選択できます。
- 新しいバージョン管理スキームが、**<major>.<minor>.<patch>** の形式で導入されました。
- Certificate Management Protocol (CMP、RFC 4210)、Certificate Request Message Format (CRMF)、および HTTP transfer (RFC 6712) へのサポートが追加されました。
- GET および POST、リダイレクト、プレーンエンコードおよび ASN.1 エンコードのコンテンツ、プロキシー、およびタイムアウトに対応する HTTP(S) クライアントが導入されました。
- 新しい鍵派生関数 API (EVP_KDF) およびメッセージ認証コード API (EVP_MAC) が追加されました。
- **enable-ktls** 設定オプションを使用したコンパイルによる Linux カーネル TLS (KTLS) のサポートが追加されました。
- CAdES-BES 署名検証のサポートが追加されました。
- CAdES-BES 署名スキームおよび属性のサポート (RFC 5126) が CMS API に追加されました。
- 新しいアルゴリズムのサポートが追加されました。以下に例を示します。
  - KDF アルゴリズムの SINGLE STEP および SSH
  - MAC アルゴリズム GMAC および KMAC。
  - KEM アルゴリズム "RSASVE"
  - 暗号アルゴリズム AES-SIV
- AES_GCM を使用した AuthEnvelopedData コンテンツタイプ構造 (RFC 5083) を追加しました。
- **PKCS12_create()** 機能を使用した PKCS #12 作成用のデフォルトアルゴリズムが、より最新の PBKDF2 および AES ベースのアルゴリズムに変更されました。
- 新しい汎用トレース API を追加しました。

(BZ#1990814)

OpenSSL にプロバイダーが含まれるようになりました。

RHEL 9 に含まれるバージョン 3.0.1 の OpenSSL ツールキットに、プロバイダーの概念が追加されました。プロバイダーは一連のアルゴリズムで、異なるアプリケーションに異なるプロバイダーを選択できます。OpenSSL には現在、**base**、**default**、**fips**、**legacy**、および **null** のプロバイダーが含まれています。

デフォルトでは、OpenSSL は、RSA、DSA、DH、CAMELLIA、SHA-1、SHA-2 などの一般的に使用されるアルゴリズムを含む **default** プロバイダーをロードしてアクティブ化します。

カーネルで FIPS フラグが設定されていると、OpenSSL は FIPS プロバイダーを自動的に読み込み、FIPS が承認したアルゴリズムのみを使用します。そのため、OpenSSL を FIPS モードに手動で切り替える必要がありません。

システムレベルで別のプロバイダーに変更するには、**openssl.cnf** 設定ファイルを編集します。たとえば、シナリオで **レガシー** プロバイダーの使用が必要な場合は、対応するセクションのコメントを外します。

警告

プロバイダーを明示的にアクティブにすると、デフォルトプロバイダーの暗黙的なアクティブ化が上書きされ、OpenSSH スイートなどにより、システムにリモートでアクセスできない場合があります。

各プロバイダーに含まれるアルゴリズムの詳細は、関連する man ページを参照してください。たとえば、**legacy** プロバイダーの **OSSL_PROVIDER-legacy(7)** の man ページなどです。

(BZ#2010291)

OpenSSL のランダムビットジェネレータが CPACF に対応

今回のリリースの **openssl** パッケージでは、OpenSSL NIST SP800-90A 準拠の AES ベースの Deterministic Random Bit Generator (DRBG) において、CP Assist for Cryptographic Functions (CPACF) のサポートが導入されました。

(BZ#1871147)

openssl-spkac が SHA-1 および SHA-256 で署名された SPKAC ファイルを作成できるようになりました。

openssl-spkac ユーティリティーは、MD5 とは異なるハッシュで署名された Netscapesigned public key and challenge (SPKAC) ファイルを作成できるようになりました。また、SHA-1 および SHA-256 ハッシュで署名された SPKAC ファイルも作成して検証できるようになりました。

(BZ#1970388)

RHEL 9 は openCryptoki3.17.0 を提供します

RHEL 9 には、**openCryptoki** バージョン 3.17.0 が同梱されています。バージョン 3.16.0 への主なバグ修正および機能強化は、以下のとおりです。

- **p11sak** ユーティリティーは、キーをリスト表示する新しい機能を追加します。
- **openCryptoki** は以下をサポートするようになりました。
  - OpenSSL 3.0.
  - イベント通知。
  - ICA トークンのソフトウェアのフォールバック。
- ハードウェアクリプトアダプターが使用可能になっている場合でも、WebSphereApplicationServer の始動に失敗することはなくなりました。

RHEL 9 には、RHEL 固有の追加のパッチが含まれる OpenSSL が含まれています。システムが FIPS (Federal Information Processing Standards) モードになっている場合、OpenSSL は FIPS プロバイダーとベースプロバイダーを自動的に読み込み、アプリケーションが FIPS プロバイダーを使用するように強制します。したがって、RHEL 9 の **openCryptoki** の動作はアップストリームとは異なります。

- OpenSSL の暗号化操作の実装に依存するトークン (ソフトトークンおよび ICA トークンソフトウェアフォールバック) は、未承認のメカニズムがまだ使用可能としてリストされている場合でも、FIPS 承認済みのメカニズムのみをサポートするようになりました。
- **openCryptoki** は、2 つの異なるトークンデータ形式をサポートしています。これらは、FIPS 承認されていないアルゴリズム (DES や SHA1 など) を使用する古いデータ形式と、FIPS 承認されたアルゴリズムのみを使用する新しいデータ形式です。
  FIPS プロバイダーは FIPS 承認のアルゴリズムのみの使用を許可しているため、古いデータ形式は機能しなくなりました。

重要

openCryptoki を RHEL 9 で機能させるには、システムで FIPS モードを有効にする前に、トークンを移行して新しいデータ形式を使用します。**openCryptoki 3.17** では古いデータ形式がデフォルトのままであるため、これが必要です。システムが FIPS 対応に変更されると、古いトークンデータ形式を使用する既存の **openCryptoki** インストールは機能しなくなります。

openCryptoki で提供される **pkcstok_migrate** ユーティリティーを使用して、トークンを新しいデータ形式に移行できます。移行中は、**pkcstok_migrate** は FIPS で承認されていないアルゴリズムを使用することに注意してください。したがって、システムで FIPS モードを有効にする前に、このツールを使用します。詳細は、FIPS 準拠への移行 -pkcstok_migrate ユーティリティーを 参照してください。

(BZ#1869533)

バージョン 3.7.3 で提供される GnuTLS

RHEL 9 では、**gnutls** パッケージはアップストリームバージョン 3.7.3 で提供されています。これにより、以前のバージョンの改善とバグ修正 (特に以下) が数多く追加されました。

- FIPS 140-3 明示的なインジケーターの API を導入。
- PKCS#12 ファイルのエクスポートに強化されたデフォルト。
- 初期データ (ゼロラウンドトリップデータ、0-RTT) 交換のタイミングを修正しました。
- **certutil** ツールは、証明書署名要求 (CSR) の署名時に、認証局 (CA) から CRL (Certificate Revocation List) 配布点を継承しなくなりました。

(BZ#2033220)

RHEL 9 は NSS3.71 を提供します

RHEL 9 は、Network Security Services (NSS) ライブラリーバージョン 3.71 とともに配布されます。主な変更点は、以下のとおりです。

- 従来の DBM データベース形式のサポートは完全に削除されました。NSS は、RHEL 9 の SQLite データベース形式のみをサポートします。

- PKCS#12 暗号化暗号は、PBE-SHA1-RC2-40 および PBE-SHA1-2DES の代わりに、PBKDF2 および SHA-256 アルゴリズムで AES-128-CBC を使用するようになりました。

(BZ#2008320)

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。

(BZ#2099438)

OpenSSH の最小 RSA 鍵ビット長オプション

誤って短い RSA 鍵を使用すると、システムが攻撃に対してより脆弱になる可能性があります。今回の更新により、OpenSSH サーバーおよびクライアントの RSA キーの最小ビット長を設定できるようになりました。最小の RSA 鍵の長さを定義するには、OpenSSH サーバーの場合は **/etc/ssh/sshd_config** ファイルで、OpenSSH クライアントの場合は **/etc/ssh/ssh_config** ファイルで新しい **RSAMinSize** オプションを使用します。

(BZ#2119694)

8.7p1 で配布された OpenSSH

RHEL 9 には、バージョン 8.7p1 の **OpenSSH** が含まれています。このバージョンでは、**OpenSSH** バージョン 8.0p1 で多くの機能拡張とバグ修正が行われました。これは、RHEL 8.5 で配布されており、以下が重要な変更となります。

新機能

- 以前に使用されていた SCP/RCP プロトコルの代わりに SFTP プロトコルを使用した転送のサポート。SFTP は、より予測可能なファイル名の処理を提供するため、リモート側のシェルで glob(3) パターンを拡張する必要はありません。
  SFTP のサポートはデフォルトで有効になっています。使用しているシナリオで SFTP が利用できない場合や互換性がない場合は、**-O** フラグを使用して、元の SCP/RCP プロトコルを強制的に使用できます。
- ファイル/関数/行パターンリストで最大デバッグロギングを強制できるようにする **LogVerbose** 設定ディレクティブ。
- 新しい **sshd_config** の **PerSourceMaxStartups** ディレクティブおよび **PerSourceNetBlockSize** ディレクティブを使用した、クライアントのアドレスベースのレートリミット。これにより、全体の **MaxStartups** 制限よりも詳細な制御が可能になります。
- **HostbasedAcceptedAlgorithms** キーワードが、キータイプによるフィルタリングではなく、署名アルゴリズムに基づいてフィルタリングされるようになりました。
- **glob** パターンを使用して追加の設定ファイルを含めることができる、**sshd** デーモンの **Include sshd_config** キーワード。
- FIDO Alliance で規定されている Universal 2nd Factor (U2F) ハードウェアオーセンティケーターに対応します。U2F/FIDO は、Web サイトの認証に広く使用されている、安価な 2 要素認

証ハードウェア用のオープンスタンダードです。**OpenSSH** では、FIDO デバイスは、新しい公開鍵タイプの **ecdsa-sk** および **ed25519-sk** と、対応する証明書タイプで対応しています。

- 使用するたびに PIN を必要とする FIDO キーに対応します。このような鍵は、新しい **verify-required** を指定して **ssh-keygen** を使用して生成できます。PIN が必要な鍵を使用すると、署名の操作を完了するための PIN を求めるプロンプトが表示されます。

- **authorized_keys** ファイルが、新しい **verify-required** オプションに対応するようになりました。このオプションでは、署名を行う前に FIDO 署名がユーザーの存在のトークン検証を表明する必要があります。FIDO プロトコルは、ユーザー検証に複数の方法をサポートしています。OpenSSH は、現在 PIN 検証のみをサポートしています。

- FIDO **webauthn** 署名の検証に対応しました。**webauthn** は、Web ブラウザーで FIDO 鍵を使用するための規格です。このような署名は、プレーンの FIDO 署名とは形式が若干異なるため、明示的なサポートが必要になります。

バグ修正

- **ClientAliveCountMax=0** キーワードのセマンティクスを明確にしました。現在では、最初の Liveness テストの成功に関係なく、そのテストの後に接続を即座に強制終了するという以前の動作ではなく、接続の強制終了を完全に無効にしています。

セキュリティー

- XMSS キータイプの秘密鍵解析コードで、悪用可能な整数オーバーフローのバグを修正しました。この鍵タイプは依然として試験的なもので、デフォルトではコンパイルされていません。ポータブルの OpenSSH には、有効にするユーザー向けの autoconf オプションは存在しません。

- Spectre、Meltdown、Rambled などの投機やメモリーサイドチャネル攻撃に対して、RAM にある秘密鍵に対する保護を追加しました。このリリースでは、ランダムデータ (現在 16KB) で設定される比較的大きなプレキーから派生した対称鍵で秘密鍵が使用されていない場合に、秘密鍵を暗号化します。

(BZ#1952957)

OpenSSH ではデフォルトでロケール転送が無効になっています

コンテナーや仮想マシンなどの小さなイメージで **C.UTF-8** ロケールを使用すると、従来の **en_US.UTF-8** ロケールを使用するよりもサイズが小さくなり、パフォーマンスが向上します。

ほとんどのディストリビューションは、デフォルトでロケール環境変数を送信し、サーバー側でそれらを受け入れます。ただし、これは、**C** または **C.UTF-8** 以外のロケールを使用するクライアントから **glibc-langpack-en** または **glibc-all-langpacks** パッケージがインストールされていないサーバーに SSH 経由でログインすると、ユーザーエクスペリエンスが低下することを意味します。具体的には、UTF-8 形式の出力が壊れており、一部のツールが機能しないか、頻繁に警告メッセージを送信していました。

この更新により、OpenSSH ではロケール転送がデフォルトでオフになります。これにより、クライアントが少数のロケールのみをサポートする最小限のインストールでサーバーに接続する場合でも、ロケールを実行可能に保つことができます。

(BZ#2002734)

OpenSSH は U2F/FIDO セキュリティーキーをサポートします

以前は、ハードウェアに格納された OpenSSH キーは、SSH での他のセキュリティーキーの使用を制限する PKCS#11 標準を介してのみサポートされていました。U2F/FIDO セキュリティーキーのサポート

はアップストリームで開発され、現在 RHEL 9 に実装されています。これにより、PKCS#11 インターフェイスに関係なく SSH 内のセキュリティーキーの使いやすさが向上します。

(BZ#1821501)

バージョン 4.6 で提供される Libreswan

RHEL 9 では、Libreswan はアップストリームバージョン 4.6 で提供されています。このバージョンは、多くのバグ修正と機能拡張を提供します。特に、インターネットキーエクスチェンジバージョン 2 (IKEv2) で使用されるラベル付き IPsec の改善です。

(BZ#2017355)

Libreswan はデフォルトで IKEv1 パッケージを受け入れません

Internet Key Exchange v2(IKEv2) プロトコルが広くデプロイメントされているため、Libreswan はデフォルトで IKEv1 パケットをサポートしなくなりました。IKEv2 では、より安全な環境と攻撃に対する回復力が実現されています。シナリオで IKEv1 を使用する必要がある場合は、**ikev1-policy=accept** オプションを **/etc/ipsec.conf** 設定ファイルに追加することで有効にできます。

(BZ#2039877)

RHEL 9 は stunnel5.62 を提供します

RHEL 9 は、**stunnel** パッケージバージョン 5.62 とともに配布されます。主なバグ修正と機能拡張は、以下のとおりです。

- FIPS モードのシステムでは、**stunnel** は常に FIPS モードを使用するようになりました。
- **NO_TLSv1.1**、**NO_TLSv1.2**、および **NO_TLSv1.3** オプションは、それぞれ **NO_TLSv1_1**、**NO_TLSv1_2**、および **NO_TLSv1_3** に名前が変更されました。
- 新しいサービスレベルの **sessionResume** オプションは、セッションの再開を有効または無効にします。
- LDAP が **protocol** オプションを使用して **stunnel** クライアントでサポートされるようになりました。
- Bash-completion スクリプトが利用できるようになりました。

(BZ#2039299)

RHEL 9 は nettle 3.7.3 を提供します。

RHEL 9 は、**nettle** パッケージ 3.7.3 バージョンに、バグ修正および機能強化を複数提供します。主な変更は以下のとおりです。

- 新しいアルゴリズムとモード (**Ed448**、**SHAKE256**、**AES-XTS**、**SIV-CMAC** など) に対応します。
- 既存のアルゴリズムにアーキテクチャー固有の最適化を追加します。

(BZ#1986712)

RHEL 9 が p11-kit 0.24 を提供

RHEL 9 では、**p11-kit** パッケージに 0.24 バージョンが提供されます。このバージョンでは、バグ修正および機能強化が複数追加されました。特に、信頼できない認証局を保存するサブディレクトリーの名前が **blocklist** に変更されました。

(BZ#1966680)

cyrus-sasl は Berkeley DB の代わりに GDBM を使用

cyrus-sasl パッケージは、**libdb** 依存関係なしで構築されるようになりました。**sasldb** プラグインは、Berkeley DB ではなく GDBM データベース形式を使用します。古い Berkeley DB 形式で保存されている既存の Simple Authentication and Security Layer (SASL) データベースを移行するには、**cyrusbdb2current** を使用します。以下の構文を使用します。

```
cyrusbdb2current <sasldb_path> <new_path>
```

(BZ#1947971)

RHEL 9 の SELinux ポリシーは、現在のカーネルで最新のものになる

SELinux ポリシーに、カーネルの一部でもある新しいパーミッション、クラス、およびケイパビリティが含まれるようになりました。そのため、SELinux はカーネルの持つポテンシャルを最大限に活用することができます。特に、SELinux ではパーミッション付与の粒度が改善され、セキュリティー上の利点が得られました。また、MLS SELinux ポリシーは、システムにポリシーに対して不明なパーミッションが含まれていた場合に、一部のシステムを起動させなくしていたため、これにより、MLS SELinux ポリシーでシステムを起動できるようになります。

(BZ#1941810、BZ#1954145)

デフォルトの SELinux ポリシーにより、テキスト再配置ライブラリーのコマンドが禁止されます。

インストール済みシステムのセキュリティーフットプリントを向上させるために、**selinuxuser_execmod** ブール値がデフォルトでオフになりました。そのため、ライブラリーファイルに **textrel_shlib_t** ラベルがない場合は、SELinux ユーザーは、テキストの再配置を必要とするライブラリーを使用してコマンドを入力できません。

(BZ#2055822)

OpenSCAP はバージョン 1.3.6 で提供

RHEL 9 にはバージョン 1.3.6 に OpenSCAP が含まれており、バグ修正および改善点が提供されます。以下に例を示します。

- **--local-files** オプションを使用してスキャン中にダウンロードするのではなく、リモート SCAP ソースデータストリームコンポーネントのローカルコピーを指定できます。
- OpenSCAP は、複数の **--rule** 引数を受け入れて、コマンドラインで複数のルールを選択します。
- **--skip-rule** オプションを使用して、一部のルールの評価を省略できます。
- **OSCAP_PROBE_MEMORY_USAGE_RATIO** 環境変数を使用して、OpenSCAP プローブによって消費されるメモリーを制限できます。
- OpenSCAP は、修復タイプとして OSBuild ブループリントをサポートするようになりました。

(BZ#2041782)

OSCAP Anaconda Add-on が、新しいアドオン名に対応しました。

この改善により、**OSCAP Anaconda アドオン** プラグインのキックスタートファイルにある従来の **org_fedora_oscap** アドオン名とは異なり、新しい **com_redhat_oscap** アドオン名を使用できるようになりました。キックスタートセクションの設定は、以下のようになります。

```
%addon com_redhat_oscap
  content-type = scap-security-guide
%end
```

OSCAP Anaconda Add-on は、現在、従来のアドオン名と互換性がありますが、今後のメジャーバージョンの RHEL では、従来のアドオン名に対するサポートが削除されます。

(BZ#1893753)

CVE OVAL フィードが圧縮される

今回の更新で、Red Hat は CVE OVAL フィードを圧縮形式で提供するようになりました。これらは XML ファイルとしては利用できなくなりますが、代わりに **bzip2** 形式になります。RHEL9 のフィードの場所も、この変更を反映するように更新されています。圧縮されたコンテンツの参照は標準化されていないため、サードパーティーの SCAP スキャナーでは、圧縮されたフィードを使用するスキャンルールで問題が発生する可能性があることに注意してください。

(BZ#2028435)

バージョン 0.1.60 で提供される SCAP セキュリティーガイド

RHEL 9 には、バージョン 0.1.60 の **scap-security-guide** パッケージが含まれています。このバージョンでは、主なバグ修正および機能強化が数多く追加されました。

- PAM スタックを強化するルールは、設定ツールとして**authselect**を使用するようになりました。
- SCAP セキュリティーガイドは、STIG プロファイルのデルタ調整ファイルを提供するようになりました。この調整ファイルは、DISA の自動化された STIG と SSG の自動化されたコンテンツの違いを表すプロファイルを定義します。

(BZ#2014561)

RHEL 9.0 で対応する SCAP セキュリティーガイドプロファイル

RHEL 9.0 に含まれている SCAP セキュリティーガイドコンプライアンスプロファイルを使用すると、発行組織からの推奨事項に合わせてシステムを強化できます。その結果、関連する修復と SCAP プロファイルを使用して、必要な強化レベルに応じて RHEL 9 システムのコンプライアンスを設定および自動化できます。

プロファイル名	プロファイル ID	ポリシーバージョン
Security of Information Systems (ANSSI) BP-028 Enhanced Level	**xccdf_org.ssgproject.content_profile_anssi_bp28_enhanced**	1.2
French National Agency for the Security of Information Systems (ANSSI) BP-028 High Level	**xccdf_org.ssgproject.content_profile_anssi_bp28_high**	1.2

プロファイル名	プロファイル ID	ポリシーバージョン
French National Agency for the Security of Information Systems (ANSSI) BP-028 Intermediary Level	**xccdf_org.ssgproject.content_profile_anssi_bp28_intermediary**	1.2
French National Agency for the Security of Information Systems (ANSSI) BP-028 Minimal Level	**xccdf_org.ssgproject.content_profile_anssi_bp28_minimal**	1.2
[ドラフト] CIS Red Hat Enterprise Linux 9 Benchmark for Level 2 - Server	**xccdf_org.ssgproject.content_profile_cis**	ドラフト[a]
[ドラフト] CIS Red Hat Enterprise Linux 9 Benchmark for Level 1 - Server	**xccdf_org.ssgproject.content_profile_cis_server_l1**	ドラフト[a]
[ドラフト] CIS Red Hat Enterprise Linux 9 Benchmark for Level 1 - Workstation	**xccdf_org.ssgproject.content_profile_cis_workstation_l1**	ドラフト[a]
[ドラフト] CIS Red Hat Enterprise Linux 9 Benchmark for Level 2 - Workstation	**xccdf_org.ssgproject.content_profile_cis_workstation_l2**	ドラフト[a]
[ドラフト] Unclassified Information in Non-federal Information Systems and Organizations (NIST 800-171)	**xccdf_org.ssgproject.content_profile_cui**	r2
Australian Cyber Security Centre (ACSC) Essential Eight	**xccdf_org.ssgproject.content_profile_e8**	バージョン付けなし
Health Insurance Portability and Accountability Act (HIPAA)	**xccdf_org.ssgproject.content_profile_hipaa**	バージョン付けなし
Australian Cyber Security Centre (ACSC) ISM Official	**xccdf_org.ssgproject.content_profile_ism_o**	バージョン付けなし
[DRAFT] Protection Profile for General Purpose Operating Systems	**xccdf_org.ssgproject.content_profile_ospp**	4.2.1
PCI-DSS v3.2.1 Control Baseline for Red Hat Enterprise Linux 9	**xccdf_org.ssgproject.content_profile_pci-dss**	3.2.1
[ドラフト] DISA STIG for Red Hat Enterprise Linux 9	**xccdf_org.ssgproject.content_profile_stig**	ドラフト[b]

プロファイル名	プロファイル ID	ポリシーバージョン
[ドラフト] DISA STIG with GUI for Red Hat Enterprise Linux 9	**xccdf_org.ssgproject.content_profile_stig_gui**	ドラフト[b]

[a] CIS は RHEL 9 の公式ベンチマークを公開していません。

[b] DISA は RHEL 9 の公式ベンチマークを公開していません。

警告

自動修正によりシステムが機能しなくなる場合があります。まずテスト環境で修復を実行してください。

(BZ#2045341, BZ#2045349, BZ#2045361, BZ#2045368, BZ#2045374, BZ#2045381, BZ#2045386, BZ#2045393, BZ#2045403)

RHEL 9 が fapolicyd 1.1 を提供

RHEL 9 には、**fapolicyd** パッケージバージョン 1.1 が同梱されています。以下は、主な変更点です。

- 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。詳細については、新しい **fagenrules (8)** の man ページを参照してください。

- RPM データベース外のファイルを信頼できるものとしてマークするための **/etc/fapolicyd/fapolicyd.trust** ファイルに加えて、信頼できるファイルのリストをより多くのファイルに分割することをサポートする新しい **/etc/fapolicyd/trust.d** ディレクトリーを使用できるようになりました。これらのファイルに **--trust-file** ディレクティブを指定して **fapolicyd-cli-f** サブコマンドを使用して、ファイルのエントリーを追加することもできます。詳細については、**fapolicyd-cli(1)** および **fapolicyd.trust(13)** の man ページを参照してください。

- **fapolicyd** trust データベースは、ファイル名の空白をサポートするようになりました。

- **fapolicyd** は、ファイルを信頼データベースに追加するときに、実行可能ファイルへの正しいパスを格納するようになりました。

(BZ#2032408)

Rsyslog には、より高性能な操作と CEF のための mmfields モジュールが含まれます

Rsyslog には、**mmfields** モジュールを提供する **rsyslog-mmfields** サブパッケージが含まれるようになりました。これは、プロパティー置き換えフィールド抽出を使用する代わりの方法ですが、プロパティー置き換えとは対照的に、すべてのフィールドが一度に抽出され、構造化データ部分の内部に格納されます。その結果、特に Common Event Format (CEF) などのログ形式を処理する場合や、多数のフィールドが必要であったり特定のフィールドを再使用したりする場合などに、**mmfields** を使用できます。このような場合の **mmfields** のパフォーマンスは、既存の Rsyslog 機能よりも優れています。

(BZ#2027971)

logrotate が別の rsyslog-logrotate に同梱されている

logrotate 設定は、メインの **rsyslog** パッケージから新しい **rsyslog-logrotate** パッケージに分離されました。これは、ログローテーションが必要ないなど、特定の最小環境で役立ち、不要な依存関係のインストールを防ぎます。

(BZ#1992155)

sudo が Python プラグインをサポート

RHEL 9 に含まれる **sudo** プログラムバージョン 1.9 では、Python で **sudo** プラグインを作成できます。これにより、特定のシナリオに合わせて **sudo** をより正確に改良することが容易になります。

詳細は、**sudo_plugin_python(8)** man ページを参照してください。

(BZ#1981278)

バージョン 2.5.2 で提供される libseccomp

RHEL 9.0 は、アップストリームバージョン 2.5.2 で **libseccomp** パッケージを提供します。このバージョンでは、以前のバージョンに比べて多くのバグ修正と機能拡張を提供します。

- Linux の syscall テーブルがバージョン **v5.14-rc7** に更新されました。
- 通知ファイル記述子を取得するために、**get_notify_fd()** 関数が Python バインディングに追加されました。
- すべてのアーキテクチャーの多重化されたシステムコール処理が 1 つの場所に統合されました。
- 多重化されたシステムコールのサポートが、PowerPC (PPC) および MIPS アーキテクチャーに追加されました。
- カーネル内で **SECCOMP_IOCTL_NOTIF_ID_VALID** 操作の意味が変更されました。
- **libseccomp** ファイル記述子通知ロジックは、カーネルの以前および新しい **SECCOMP_IOCTL_NOTIF_ID_VALID** の使用をサポートするように変更されました。
- **seccomp_load ()** を 1 回しか呼び出せなかったバグを修正しました。
- フィルターに **_NOTIFY** アクションがある場合にのみ、通知 **fd** を要求するように通知 **fd** 処理を変更しました。
- **SCMP_ACT_NOTIFY** に関するドキュメントを **seccomp_add_rule(3)** のマンページに追加しました。
- メンテナーの GPG キーを明確にしました。

(BZ#2019887)

Clevis が SHA-256 に対応しました。

この改善により、Clevis フレームワークは、**RFC 7638** が推奨する JSON Web 鍵 (JWK) サムプリントのデフォルトハッシュとして **SHA-256** アルゴリズムに対応します。古いサムプリント (SHA-1) にも対応しているため、以前に暗号化したデータは復号できます。

(BZ#1956760)

4.8. ネットワーク

diag モジュールがカーネルで利用可能に

diag モジュールがカーネルイメージに含まれるようになりました。今回の更新で、**ss** コマンドの使用時に、**diag** モジュールを動的に読み込む必要がなくなりました。これにより、カーネルモジュールに関するカスタマーポリシーに関係なく、ネットワーク問題のデバッグが容易になります。カーネルに含まれるモジュール:

```
CONFIG_INET_DIAG
CONFIG_INET_RAW_DIAG
CONFIG_INET_TCP_DIAG
CONFIG_INET_UDP_DIAG
CONFIG_INET_MPTCP_DIAG
CONFIG_NETLINK_DIAG
CONFIG_PACKET_DIAG
CONFIG_UNIX_DIAG
```

(BZ#1948340)

新しいコアおよび IPv4 関連のネットワーキング sysctl カーネルパラメーター

RHEL 9.0 カーネルは、以前の RHEL バージョンと比較して、次の新しいコアおよび IPv4 ネットワーキング **sysctl** パラメーターを提供します。

- **net.core.devconf_inherit_init_net**
- **net.core.gro_normal_batch**
- **net.core.high_order_alloc_disable**
- **net.core.netdev_unregister_timeout_secs**
- **net.ipv4.fib_multipath_hash_fields**
- **net.ipv4.fib_notify_on_flag_change**
- **net.ipv4.fib_sync_mem**
- **net.ipv4.icmp_echo_enable_probe**
- **net.ipv4.ip_autobind_reuse**
- **net.ipv4.nexthop_compat_mode**
- **net.ipv4.raw_l3mdev_accept**
- **net.ipv4.tcp_comp_sack_slack_ns**
- **net.ipv4.tcp_migrate_req**
- **net.ipv4.tcp_mtu_probe_floor**
- **net.ipv4.tcp_no_ssthresh_metrics_save**
- **net.ipv4.tcp_reflect_tos**

これらのパラメーターの詳細は、**kernel-doc** パッケージをインストールし、以下のファイルを参照してください。

- **/usr/share/doc/kernel-doc-<version>/Documentation/admin-guide/sysctl/net.rst**
- **/usr/share/doc/kernel-doc-<version>/Documentation/networking/ip-sysctl.rst**

(BZ#2068532)

ゾーン間でパケットを送信する際の firewalld での挙動の変更

ゾーンベースのファイアウォールでは、パケットは 1 つのゾーンにしか入りません。暗黙的なパケット送信は概念違反であり、トラフィックやサービスを予想外に許可する可能性があります。Red Hat Enterprise Linux 9 では、**firewalld** で 2 つのゾーン間での暗黙的なパケット送信が許可されなくなりました。

この変更の詳細は、ゾーン間でパケットを送信する際の **firewalld** での挙動の変更 ナレッジアーティクルを参照してください。

(BZ#2029211)

intra-zone forwarding がデフォルトで有効に

firewalld のゾーン転送機能により、firewalld ゾーン内のインターフェイスまたはソース間でトラフィックを転送できます。RHEL 9.0 以降、この機能はデフォルトで有効になっています。**firewall-cmd** ユーティリティーの **--add-forward** オプションを使用して、特定ゾーンのゾーン転送を有効にします。**firewall-cmd --list-all** コマンドは、ゾーンに対して intra-zone 転送が有効または無効であるかを表示します。

```
# firewall-cmd --list-all
public (active)
  ...
  forward: no
```

(BZ#2089193)

多様性により配慮した Nmstate

Red Hat では、意識的な言語の使用に取り組んでいます。これについては、多様性を受け入れるオープンソースの強化 で詳細を参照してください。そのため、**nmstate**API の **slave** という用語は、**port** という用語に置き換えられています。

(BZ#1969941)

NetworkManager は、IBM Z の rd.znet_ifname カーネルオプションで設定されたインターフェイス名をサポートします

この機能拡張により、IBM Z プラットフォームで、NetworkManager は、ネットワークから Red Hat Enterprise Linux をインストールまたは起動するときに、**rd.znet** および **rd.znet_ifname** カーネルコマンドラインオプションを解釈するようになりました。その結果、デフォルトのサブチャネルの代わりに、サブチャネルによって識別されるネットワークインターフェイスの名前を指定することができます。

(BZ#1980387)

hostapd パッケージが RHEL 9.0 に追加されました。

今回のリリースで、RHEL は **hostapd** パッケージを提供します。ただし、Red Hat が **hostapd** に対応

するのは、イーサネットネットワークで RHEL ホストを 802.1X 認証子として設定することのみです。Wi-Fi アクセスポイントや Wi-Fi ネットワークのオーセンティケーターなど、その他のシナリオには対応していません。

FreeRADIUS バックエンドを備えた 802.1X オーセンティケーターとして RHEL を設定する方法の詳細については、Setting up an 802.1x network authentication service for LAN clients using hostapd with FreeRADIUS backend を参照してください。

(BZ#2019830)

バージョン 1.18.2 で提供される ModemManager

RHEL 9.0 は、アップストリームバージョン 1.18.2 で **ModemManager** パッケージを提供します。このバージョンには、以前のバージョンに対するバグ修正と機能拡張が含まれています。特に、次のとおりです。

- 5G 機能を使用するデバイスの機能およびモード処理を向上
- その他のデバイスのサポート

(BZ#1996716)

NetworkManager では、ボンディングポートの queue_id を変更できます。

ボンディングの NetworkManager ポートが、**queue_id** パラメーターに対応するようになりました。**eth1** がボンディングインターフェイスのポートである場合は、以下のコマンドでボンディングポートの **queue_id** を有効にできます。

```
# nmcli connection modify eth1 bond-port.queue-id 1
# nmcli connection up eth1
```

このオプションを使用する必要があるネットワークインターフェイスは、すべてのインターフェイスに適切な優先度が設定されるまで、複数の呼び出しで設定する必要があります。詳細は、**kernel-doc** により提供される **/usr/share/docs/kernel-doc-_<version>/Documentation/networking/bonding.rst** を参照してください。

(BZ#1949127)

最新の NetworkManager による blackhole、prohibit、unreachable ルートタイプの設定のサポート

カーネルは、共通の **unicast**、**broadcast**、および **local** ルートタイプ以外の複数のルートタイプをサポートします。さらに、ユーザーは NetworkManager の接続プロファイルで **blackhole**、**prohibit**、**unreachable** な静的ルートタイプを設定できるようになりました。プロファイルがアクティブになると、NetworkManager によりプロファイルが追加されます。

(BZ#2060013)

RoCE Express アダプターが、改善されたインターフェイス命名スキームを使用するようになりました。

この機能拡張により、RDMA over Converged Ethernet (RoCE) Express アダプターは、予測可能なインターフェイス命名スキームと z-system (zPCI) コネクターの Peripheral Communication Interface を使用します。この命名スキームでは、RHEL はユーザー識別子 (UID) または機能識別子 (FID) を使用して一意の名前を生成します。一意の UID が使用できない場合、RHEL は FID を使用して命名スキームを設定します。

(BZ#2091653)

4.9. カーネル

RHEL 9.0 のカーネルバージョン

Red Hat Enterprise Linux 9.0 には、カーネルバージョン 5.14.0-70 が同梱されています。

(BZ#2077836)

Red Hat は、デフォルトで、特権ユーザーに対してのみ、すべての RHEL バージョンで eBPF を有効にします。

Extended Berkeley Packet Filter (**eBPF**) は、ユーザーが Linux カーネル内でカスタムコードを実行できるようにする複雑なテクノロジーです。その性質上、**eBPF** コードはベリファイアやその他のセキュリティーメカニズムを通過する必要があります。Common Vulnerabilities and Exposures (CVE) インスタンスがあり、このコードのバグが不正な操作に悪用される可能性がありました。このリスクを軽減するため、Red Hat は、特権ユーザーの場合にのみ、すべての RHEL バージョンで **eBPF** を有効にしています。kernel.command-line パラメーター **unprivileged_bpf_disabled=0** を使用して、非特権ユーザーに対して **eBPF** を有効にすることができます。

ただし、

- **unprivileged_bpf_disabled=0** を適用すると、カーネルが Red Hat サポートの資格を失い、システムがセキュリティーリスクにさらされます。
- Red Hat は、**CAP_BPF** 機能を持つプロセスを、その機能が **CAP_SYS_ADMIN** と同等であるかのように扱うことを推奨します。
- **unprivileged_bpf_disabled=0** を設定しても、非特権ユーザーが多くの BPF プログラムを実行するには不十分です。これは、ほとんどの BPF プログラムタイプのロードには追加の機能 (通常は **CAP_SYS_ADMIN** または **CAP_PERFMON**) が必要になるためです。

カーネルコマンドラインパラメーターの設定方法は、カーネルコマンドラインパラメーターの設定 を参照してください。

(BZ#2091643)

Red Hat は、マイナーリリースに対してのみカーネルシンボルを保護します。

Red Hat は、保護されたカーネルシンボルを使用してカーネルモジュールをコンパイルする場合にのみ、カーネルモジュールが Extended Update Support (EUS) リリース内の将来のすべての更新でロードされ続けることを保証します。RHEL 9 のマイナーリリース間では、カーネルアプリケーションバイナリーインターフェイス (ABI) の保証はありません。

(BZ#2059183)

信頼できる SecureBoot 証明書で署名された RHEL 9 Beta カーネル

これまでの RHEL ベータ版では、ユーザーがマシンオーナーキー (MOK) 機能を使用して別のベータ版公開鍵を登録する必要がありました。RHEL 9 ベータ版以降、カーネルが信頼できる SecureBoot 証明書で署名されているため、UEFI セキュアブートが有効なシステムでベータ版を使用する際に、ベータ版用の公開鍵を別途登録する必要がなくなりました。

(BZ#2002499)

RHEL 9 では、デフォルトで有効になっている cgroup-v2

コントロールグループバージョン 2 (**cgroup-v2**) 機能は、制御グループの管理を簡素化する 1 つの階層モデルを実装します。また、プロセスが、一度に 1 つのコントロールグループのメンバーにのみなれる

ようにします。**systemd** との深い統合により、RHEL システムでリソース制御を設定する際のエンドユーザーエクスペリエンスが改善されます。

新機能の開発は、主に **cgroup-v2** 向けに行われます。これには、**cgroup-v1** に欠けている機能がいくつかあります。同様に、**cgroup-v1** には、**cgroup-v2** に欠けている従来の機能がいくつか含まれています。また、制御インターフェイスも異なります。したがって、**cgroup-v1** に直接依存するサードパーティーソフトウェアは、**cgroup-v2** では適切に実行されない可能性があります。

cgroup-v1 を使用するには、以下のパラメーターをカーネルコマンドラインに追加する必要があります。

```
systemd.unified_cgroup_hierarchy=0
systemd.legacy_systemd_cgroup_controller
```

注記

cgroup-v1 と **cgroup-v2** の両方がカーネルで完全に有効になっている。カーネルから見た場合、デフォルトのコントロールグループバージョンはありません。また、システムの起動時にマウントするかどうかは、**systemd** により決定します。

(BZ#1953515)

サードパーティーのカーネルモジュールに影響を与える可能性のあるカーネル変更

5.9 以前のカーネルバージョンを持つ Linux ディストリビューションは、GPL 以外の機能としての GPL 機能のエクスポートに対応していました。これにより、ユーザーは **shim** メカニズムを介して、独自の機能を GPL カーネル機能にリンクできます。今回のリリースで、RHEL カーネルにアップストリームの変更が組み込まれました。これにより、RHEL の機能が強化され、**shim** の再バフィングにより GPL が適用されるようになりました。

重要

パートナーおよび独立したソフトウェアベンダー (ISV) は、初期バージョンの RHEL 9 でカーネルモジュールをテストして、GPL への準拠を確認する必要があります。

(BZ#1960556)

RHEL 9 の 64 ビット ARM アーキテクチャーのページサイズが 4KB

Red Hat Enterprise Linux 9 の 64 ビット ARM アーキテクチャーでは、4KB ページサイズの物理メモリーが選択されています。このサイズペアは、ARM ベースのシステムの大半に存在するワークロードおよびメモリー量と十分に一致します。大きなページサイズを効率的に使用するには、huge pages オプションを使用して、大量のメモリーや、大規模なデータセットのワークロードに対処します。

huge pages の詳細は、システムのステータスおよびパフォーマンスの監視および管理 を参照してください。

(BZ#1978382)

strace ユーティリティーで SELinux コンテキストの不一致が正しく表示されるようになりました。

strace の既存の **--secontext** オプションは、**mismatch** パラメーターで拡張されました。このパラメーターを使用すると、不一致の場合にのみ、実際のコンテキストとともに期待されるコンテキストを出力できます。出力は、2 つの感嘆符 (**!!**) で区切られます。最初は実際のコンテキスト、次に期待されるコ

ンテキストです。以下の例では、コンテキストのユーザー部分が不一致であるため、**full,mismatch** パラメーターは、実際のコンテキストとともに期待される完全なコンテキストを出力します。ただし、単独の **mismatch** を使用する場合は、コンテキストのタイプ部分のみをチェックします。コンテキストのタイプ部分が一致するため、予期されるコンテキストは出力されません。

```
[...]
$ strace --secontext=full,mismatch -e statx stat /home/user/file
statx(AT_FDCWD, "/home/user/file"
[system_u:object_r:user_home_t:s0!!unconfined_u:object_r:user_home_t:s0], ...

$ strace --secontext=mismatch -e statx stat /home/user/file
statx(AT_FDCWD, "/home/user/file" [user_home_t:s0], ...
```

SELinux コンテキストの不一致は、SELinux に関連するアクセス制御の問題を引き起こすことがよくあります。システムコールトレースに出力された不一致により、SELinux コンテキストの正確性のチェックが大幅に迅速化されます。システムコールトレースは、アクセス制御チェックに関する特定のカーネルの動作を説明することもできます。

(BZ#2038965)

perf-top が特定の列でソートできるようになりました。

今回のシステムプロファイリングツール **perf-top** の更新により、任意のイベントカラムでサンプルをソートできるようになりました。これまでは、1 つのグループで複数のイベントをサンプリングした場合、イベントは最初の列でソートしていました。サンプルをソートするには、**--group-sort-idx** コマンドラインオプションを使用し、数字キーを押すと、一致するデータ列でテーブルをソートすることができます。なお、列番号は **0** から始まります。

(BZ#1851933)

新規パッケージ: jigawatts

CRIU (Checkpoint/Restore In Userspace) は、プロセスのチェックポイントと復元を可能にする Linux ユーティリティーです。**jigawatts** パッケージには、Java ライブラリーが含まれています。これは、Java アプリケーションから CRIU メカニズムの利便性を向上させることを目的としています。

(BZ#1972029)

trace-cmd reset コマンドの挙動の変更

以前は、**trace-cmd reset** コマンドで **traceing_on** 設定が 0 にリセットされていました。**trace-cmd reset** の新しい動作では、**traceing_on** をデフォルト値 1 にリセットします。

(BZ#1933980)

RHEL 9 で Extended Berkeley Packet Filter に対応

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

eBPF バイトコードが最初にカーネルに読み込みます。その後、バイトコードは検証され、すぐにコンパイルしてネイティブマシンコードに変換されます。最後に、仮想マシンはコードを実行します。

Red Hat は、**eBPF** 仮想マシンを使用するコンポーネントを数多く提供しています。RHEL 9 では、これらのコンポーネントには以下が含まれます。

- eBPF を使用して Linux オペレーティングシステムの I/O 分析、ネットワーク、およびモニタリングを行う **BPF コンパイラーコレクション (BCC)** パッケージ。
- BCC ライブラリー。これを使用すると、BCC ツールパッケージで提供されるツールと同様のツールを開発できます。
- **bpftrace** トレース言語。
- **libbpf** パッケージ。これは、**bpf** 開発および **bpftrace** などの **bpf** 関連アプリケーションにとって重要です。
  - **libbpf** ライブラリーの **XDP** および **AF_XDP** API 部分はサポートされていません。今後のリリースで削除される可能性があります。
- eBPF for Traffic Control (tc)機能。これにより、カーネルネットワークデータパスでのプログラミング可能なパケット処理が可能になります。
- eXpress Data Path (XDP)機能。カーネルネットワークスタックがパケットを処理する前に、受信したパケットへのアクセスを提供します。Red Hat は、**libxdp** ライブラリーを介して使用されている場合にのみ **XDP** をサポートします。
- **xdp-tools** パッケージには、XDP 機能のユーザースペースサポートユーティリティーが含まれており、AMD64 および Intel64 CPU アーキテクチャーでサポートされています。**xdp-tools** パッケージには、次のものが含まれます。
  - **libxdp** ライブラリー。
  - XDP プログラムをロードするための **xdp-loader** ユーティリティー。
  - パケットのフィルタリング用の **xdp-filter** のサンプルプログラム。
  - **XDP** が有効になっているネットワークインターフェイスからパケットをキャプチャするための **xdpdump** ユーティリティー。**xdpdump** ユーティリティーは、現在、AMD64 および Intel64 CPU アーキテクチャーでのみサポートされています。これは、テクノロジープレビューとして他のアーキテクチャーで利用できます。
- eXpress Data Path (XDP)パスをユーザー空間に接続する **AF_XDP** ソケット。

(BZ#2070506)

RHEL 9 は、crash ユーティリティーバージョン 8.0.0 を提供します。

RHEL 9 は、**crash** ユーティリティーバージョン 8.0.0 とともに配布されます。バグ修正および主な機能強化は、以下のとおりです。

- **add-symbol-file** コマンドで新しい **offset** パラメーターを追加します。このサポートは、**kaslr_offset** を **gdb** に設定するのに役立ちます。
- **gdb-7.6** を **gdb-10.2** にアップグレードします。

(BZ#1896647)

makedumpfile は、改善された zstd 圧縮機能をサポートするようになりました

この機能拡張により、**makedumpfile** には、高い圧縮率を提供する Zstandard (**zstd**) 圧縮機能が含まれるようになりました。この改善は、特に大量のメモリーシステムに役立ちます。

zstd 圧縮機能は、以前の圧縮率と比較して、**vmcore** ダンプサイズと圧縮時間の消費量のバランスが取れています。その結果、圧縮メカニズムが改善されたため、許容範囲で大きな **vmcore** ファイルが作成されるようになりました。

適切な圧縮率は、システムの使用方法と RAM に格納されているデータ型にも依存することに注意してください。

(BZ#1988894)

IntelXeon スケーラブルサーバープロセッサーで有効になっている numatop

numatop は、NUMA システムで実行されているプロセスとスレッドの動作を追跡および分析し、NUMA 関連のパフォーマンスのボトルネックを特定できるメトリックを表示するツールです。

numatop は、インテルのパフォーマンスカウンターサンプリングテクノロジーを使用し、パフォーマンスデータを Linux システムの **runtime** 情報に関連付けて、本番システムでの分析を提供します。

(BZ#1874125)

RHEL 9 のデフォルトオプションとして kexec_file_load が追加されました。

この更新により、64 ビット ARM アーキテクチャーの **kexec_file_load** システムコールが追加されます。**kdump** 用のカーネル内 **kexec** ローダーを提供します。以前は、カーネルは、セキュア起動オプションが有効な場合に署名されていないカーネルイメージを読み込むことができませんでした。**kdump** メカニズムは、最初にセキュアブートが有効になっているかどうかを検出しようとし、次に実行するブートインターフェイスを選択します。その結果、署名されていないカーネルは、セキュアブートが有効で、**kexec_file_load ()** が指定されているとロードできませんでした。

この更新により問題が修正され、説明されているシナリオで署名されていないカーネルが正しく機能します。

(BZ#1895232)

makedumpfile に、推定 vmcore サイズを取得するための改善されたオプションが含まれるようになりました

この実装により、**makedumpfile** ユーティリティーには、現在実行中のカーネルのダンプサイズの見積もりを出力するのに役立つ次のオプションが含まれるようになりました。

- **--dry-run** は、他のオプションで指定されたすべての操作を実行しますが、出力ファイルは書き込みません。
- **--show-stats** は、レポートメッセージを出力します。これは、**--message-level** オプションに提供されたレベルでビット 4 を有効にする代わりの方法です。

次の例は、**--dry-run** と **--show-stats の** 使用法を示しています。

```
$ makedumpfile --dry-run --show-stats -l --message-level 7 -d 31 /proc/kcore dump.dummy
```

ダンプファイルのサイズは、パニック時のシステム状態によって異なる場合があり、オプションによって提供される見積もりが実際の状態と異なる場合があることに注意してください。

(BZ#1958452)

kexec-tools パッケージは、RHEL 9 のデフォルトの crashkernel メモリー予約値をサポートするようになりました。

kexec-tools パッケージは、デフォルトの **クラッシュカーネル** メモリー予約値を維持するようになりま

した。**kdump** サービスはデフォルト値を使用して、カーネルごとに **crashkernel** メモリーを確保します。また、この実装により、システムの利用可能なメモリーが 4GB 未満になると、**kdump** のメモリー割り当てが改善されます。

デフォルトの crashkernel 値をクエリーするには、以下を実行します。

```
$ kdumpctl get-default-crashkernel
```

デフォルトの **crashkernel** 値で予約されているメモリーがシステムでは不十分な場合は、**crashkernel** パラメーターを増やします。

起動コマンドラインの **crashkernel=auto** オプションは、RHEL 9 以降のリリースではサポートされなくなりました。

詳細は、**/usr/share/doc/kexec-tools/crashkernel-howto.txt** ファイルを参照してください。

(BZ#2034490)

コアスケジューリングは RHEL 9 でサポートされている

コアスケジューリング機能を使用すると、相互に信頼できないタスクが同じ CPU コアを共有するのを防ぐことができます。同様に、ユーザーは CPU コアを共有できるタスクのグループを定義できます。

以下のグループを指定できます。

- SMT (Cross-Symmetric Multithreading) 攻撃を軽減することでセキュリティーを改善するには、以下の手順を行います。
- コア全体を必要とするタスクを分離するには、以下を行います。たとえば、リアルタイム環境のタスク、または SIMD (Multiple Data) 処理や Single Instruction などの特定のプロセッサー機能に依存するタスクなど。

詳細は コアスケジューリング を参照してください。

(JIRA:RHELPLAN-100497)

非制限の iommu モードをデフォルトとして使用し、64 ビット ARM アーキテクチャーのパフォーマンスを強化

今回の更新で、64 ビットの ARM アーキテクチャーは、システムメモリー管理ユニット (SMMU) の Lazy Direct Memory Access (DMA) ドメインの使用にデフォルト設定されています。パフォーマンスが大幅に向上しますが、アドレスアンマップと、SMMU でのトランスレーションルックアサイドバッファー (TLB: Translation Lookaside Buffer) 間のウィンドウを追加することができます。以前のバージョンでは、64 ビットの ARM アーキテクチャーは厳格な DMA ドメインをデフォルトとして設定していたため、ページサイズが 4KB が原因でパフォーマンスが低下することがありました。

厳密な DMA ドメインモードを使用する必要がある場合は、カーネルコマンドラインを使用して **iommu.strict=1** モードを指定します。厳密な DMA ドメインを使用すると、64 ビット ARM アーキテクチャーでパフォーマンスが低下する可能性があることに注意してください。

(BZ#2050415)

kernel-rt ソースツリーが RHEL 9.0 ツリーに更新されました。

kernel-rt ソースが更新され、最新の Red Hat Enterprise Linux カーネルソースツリーを使用するようになりました。リアルタイムパッチセットも、最新のアップストリームバージョン v5.15-rt19 に更新されました。これらの更新は、バグ修正および機能強化を多数提供します。

(BZ#2002474)

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

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

(BZ#1844416)

POWERPC hv_24x7 ネストイベントのメトリックが利用可能に

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

(BZ#1780258)

IRDMA ドライバーが RHEL 9 に導入

IRDMA ドライバーは、RDMA 対応の Intel® ネットワークデバイスで RDMA 機能を有効にします。このドライバーが対応しているデバイスは、以下のとおりです。

- Intel® Ethernet Controller E810
- Intel® Ethernet Network Adapter X722

RHEL 9 は、iWARP (X722 Internet Wide-area RDMA Protocol) デバイス向けに、最新の IRDMA (Intel® Ethernet Protocol Driver for RDMA) を提供します。RHEL 9 には、iWARP および RDMA over Converged Ethernet (RoCEv2) に対応する新しい E810 デバイスが追加されました。IRDMA モジュールは、X722 用の従来の i40iw モジュールに代わるもので、i40iw 用に定義された Application Binary Interface (ABI) を拡張します。この変更は、従来の X722 RDMA-Core プロバイダー (libi40iw) と後方互換性があります。

- X722 デバイスは、iWARP およびより限定的な設定パラメーターセットにのみ対応しています。
- E810 デバイスは、次の RDMA および輻輳管理機能セットに対応しています。
  - iWARP および RoCEv2 RDMA のトランスポート
  - Priority Flow Control (PFC)
  - Explicit Congestion Notification (ECN)

(BZ#1874195)

カーネル bonding モジュール lacp_active の新しいパラメーター

RHEL 9 では、**bonding** カーネルモジュールの **lacp_active** パラメーターが導入されました。このパラメーターは、指定の間隔で Link Aggregation Control Protocol Data Unit (LACPDU) フレームを送信するかどうかを指定します。オプションは次のとおりです。

- **on** (デフォルト): 設定済みの **lacp_rate** パラメーターと共に LACPDU フレームを送信できるようにします。
- **off**: LACPDU フレームは speak when spoken to として機能します。

ポートの初期化またはバインド解除時に、LACPDU 状態フレームは引き続き送信されます。

(BZ#1951951)

4.10. ブートローダー

ブートローダーの設定ファイルを CPU アーキテクチャーを通じて統一

GRUB ブートローダーの設定ファイルが、サポートされているすべての CPU アーキテクチャーにおいて、**/boot/grub2/** ディレクトリーに格納されるようになりました。GRUB が UEFI システムのメイン設定ファイルとして使用されていた **/boot/efi/EFI/redhat/grub.cfg** ファイルは、**/boot/grub2/grub.cfg** ファイルを読み込むだけになりました。

この変更により、GRUB 設定ファイルのレイアウトが簡素化され、ユーザーの操作性が向上するとともに、以下のような注目すべきメリットが得られます。

- EFI とレガシー BIOS のどちらでも、同じインストールを起動できます。
- すべてのアーキテクチャーに同じドキュメントとコマンドを使用することができます。
- GRUB 設定ツールは、シンボリックリンクに依存しなくなり、プラットフォーム固有のケースを処理する必要がないため、より堅牢になっています。
- GRUB 設定ファイルの使い方は、CoreOS Assembler(COSA) や OSBuild で生成されたイメージと一致しています。
- GRUB の設定ファイルの使い方は、他の Linux ディストリビューションと同じです。

(JIRA:RHELPLAN-101246)

4.11. ファイルシステムおよびストレージ

一貫したユーザーエクスペリエンスのために、Samba ユーティリティーのオプションの名前が変更され、削除されました。

Samba ユーティリティーが改善され、一貫したコマンドラインインターフェイスが提供されるようになりました。この改善には、オプションの名前変更や削除が含まれます。そのため、更新後の問題を回避するには、Samba ユーティリティーを使用するスクリプトを確認し、必要に応じてスクリプトを更新します。

Samba 4.15 では、Samba ユーティリティーに以下の変更が加えられました。

- Samba コマンドラインユーティリティーは、不明なオプションを暗黙的に無視していました。予期しない動作を防ぐために、ユーティリティーが、不明なオプションを常に拒否するようになりました。
- いくつかのコマンドラインオプションには、デフォルト値を制御するのに対応する **smb.conf** が追加されました。コマンドラインオプションに **smb.conf** 変数名があるかどうかを確認するには、ユーティリティーの man ページを参照してください。
- デフォルトで、Samba ユーティリティーが標準エラー (**stderr**) にログを記録するようになりました。この挙動を変更するには、**--debug-stdout** を使用します。
- 一般的なパーサーに **--client-protection=off|sign|encrypt** が追加されました。
- 以下のオプションは、すべてのユーティリティーで名前が変更されています。
  - **--kerberos** から **--use-kerberos=required|desired|off** へ

    - **--krb5-ccache** から **--use-krb5-ccache=CCACHE** へ
    - **--scope** から **--netbios-scope=SCOPE** へ
    - **--use-ccache** から **--use-winbind-ccache** へ
- 以下のオプションがすべてのユーティリティーから削除されました。
    - **-e** および **--encrypt**
    - **--use-winbind-ccache** から削除された **-C**
    - **--netbios-scope** から削除された **-i**
    - **-S** および **--signing**
- オプションの重複を防ぐため、次のユーティリティーから特定のオプションが削除されたり、名前が変更されたりしています。
    - **ndrdump**: **-l** は、**--load-dso** では使用できなくなりました。
    - **net**: **-l** は、**--long** では使用できなくなりました。
    - **sharesec**: **-V** は、**--viewsddl** では使用できなくなりました。
    - **smbcquotas**: **--user** の名前が **--quota-user** に変更になりました。
    - **nmbd**: **--log-stdout** の名前が **--debug-stdout** に変更になりました。
    - **smbd**: **--log-stdout** の名前が **--debug-stdout** に変更になりました。
    - **winbindd**: **--log-stdout** の名前が **--debug-stdout** に変更になりました。

(BZ#2065646)

RHEL 9 の NFS クライアントとサーバーの変更点

- RHEL 9.0 NFS サーバーおよびクライアントは、セキュアでない GSS Kerberos 5 暗号化タイプ **des-cbc-crc** に対応しなくなりました。
- NFS クライアントは、UDP トランスポートを使用したファイルシステムのマウントに対応しなくなりました。

(BZ#1952863)

GFS2 ファイルシステムが、フォーマットバージョン 1802 で作成されるようになる

RHEL 9 の GFS2 ファイルシステムは、フォーマットバージョン 1802 で作成されます。これにより、以下の機能が有効になります。

- **trusted** 名前空間の拡張属性 ("trusted.* xattrs") は、**gfs2** と **gfs2-utils** で認識されます。
- **rprlvb** は、デフォルトで有効になっています。これにより、**gfs2** が更新したリソースグループデータを DLM ロック要求に割り当てることができるため、ロックを取得しているノードは、ディスクからリソースグループ情報を更新する必要がありません。これにより、場合によってはパフォーマンスが改善されます。

新しいフォーマットバージョンで作成されたファイルシステムは、以前のバージョンの RHEL にマウントできなくなり、古いバージョンの **fsck.gfs2** ユーティリティーではこれらをチェックできなくなります。

オプション **-o format=1801** で **mkfs.gfs2** コマンドを実行すると、古い形式バージョンのファイルシステムを作成できます。

マウントを解除したファイルシステムで、**tunegfs2 -r 1802 device** を実行している古いファイルシステムのフォーマットバージョンをアップグレードできます。フォーマットバージョンのダウングレードには対応していません。

(BZ#1616432)

RHEL 9 では nvml パッケージバージョン 1.10.1 が提供されます。

RHEL 9.0 は、**nvml** パッケージをバージョン 1.10.1 に更新します。今回の更新で機能が追加され、電源損失時のデータ破損のバグが修正されました。

(BZ#1874208)

exFAT ファイルシステムのサポートが追加されました。

RHEL 9.0 は、Extensible File Allocation Table (exFAT) のファイルシステムに対応します。ファイルシステムをマウント、フォーマット、および一般的に使用できるようになりました。これは通常、フラッシュメモリーでデフォルトで使用されます。

(BZ#1943423)

rpcctl コマンドが SunRPC 接続情報を表示するようになりました。

今回の更新で、**rpcctl** コマンドを使用して、システムの SunRPC オブジェクトに関する SunRPC **sysfs** ファイルで収集された情報を表示するようになりました。**sysfs** ファイルシステムを介して、SunRPC ネットワークレイヤーのオブジェクトを表示、削除、および設定できます。

(BZ#2059245)

LVM のデバイスセットの制限

デフォルトでは、RHEL 9 の LVM は、明示的に選択するデバイスのみを使用します。新しいコマンド **lvmdevices** および **vgimportdevices** を使用して、特定のデバイスを選択します。**pvcreate** コマンド、**vgcreate** コマンド、および **vgextend** コマンドを使用して、**lvm** に新しいデバイスをまだ選択していない場合は間接的に選択します。LVM は、このコマンドのいずれかを使用して選択するまで、システムに接続されているデバイスを無視します。**lvm** コマンドは、選択したデバイスのリストを、デバイスファイル **/etc/lvm/devices/system.devices** に保存します。新しいデバイスファイル機能を有効にすると、**lvm.conf** フィルターまたはその他のコマンドライン設定フィルターは機能しません。デバイスファイルを削除または無効にすると、LVM は該当するすべてのデバイスにフィルターを適用します。この機能の詳細は、**lvmdevices(8)** man ページを参照してください。

(BZ#1749513)

nvme_tcp.ko を使用する NVMe/TCP ホストが完全にサポートされるようになりました。

nvme_tcp.ko カーネルモジュールを使用した TCP/IP ネットワーク (NVMe/TCP) での NVMe (Nonvolatile Memory Express) ストレージが完全にサポートされるようになりました。**nvmet_tcp.ko** モジュールの NVMe/TCP ターゲットは、RHEL 9.0 で Unmaintained ステータスで利用できます。

(BZ#2054441)

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

marginal_pathgroups 設定オプションに新しい値 **fpin** を追加すると、**multipathd** を有効化して Link Integrity Fabric Performance Impact Notification (PFIN-Li) イベントを監視し、リンク整合性の問題があるパスをマージナルパスグループに移動します。**fpin** 値を設定すると、**multipathd** は既存のマージナルパス検出方法をオーバーライドし、ファイバーチャネルファブリックに依存してリンクの整合性の問題を特定します。

この機能強化により、**multipathd** メソッドは、PFIN-Li イベントを発行できるファイバーチャネルファブリック上のマージナルパスの検出においてより堅牢になります。

(BZ#2053642)

4.12. 高可用性およびクラスター

新しく作成されたクラスターでは、resource-stickiness リソースのメタ属性が 0 ではなく 1 にデフォルトが設定されるようになる

以前では、**resource-stickiness** リソースのメタ属性のデフォルト値は、新しく作成したクラスターではデフォルト値 0 でした。このメタ属性のデフォルトは 1 になりました。

stickiness を 0 にすると、クラスターは、必要に応じてリソースを移動して、ノード間でリソースのバランスを調整できます。これにより、関連のないリソースが起動または停止したときにリソースが移動する可能性があります。stickiness が高くなると、リソースは現在の場所に留まり、その他の状況が stickiness を上回る場合に限り移動するようになります。これにより、新しく追加したノードに割り当てられたリソースは、管理者の介入なしには利用できなくなる可能性があります。どちらのアプローチも予想外の動作を起こす可能性がありますが、ほとんどのユーザーは、ある程度の stickiness を使用することが好みます。このメタ属性のデフォルト値が、この設定を反映して 1 に変更されました。

この変更の影響を受けるのは、新しく作成されたクラスターのみであるため、既存のクラスターの動作は変更しません。クラスターの古い挙動を使用することを好むユーザーは、リソースのデフォルトから **resource-stickiness** エントリーを削除できます。

(BZ#1850145)

自動アクティブ化を制御する新しい LVM ボリュームグループフラグ

LVM ボリュームグループは、ボリュームグループから作成した論理ボリュームを起動時に自動的にアクティブにするかどうかを制御する **setautoactivation** フラグに対応するようになりました。クラスターで Pacemaker が管理するボリュームグループを作成する場合は、データの破損を防ぐために、ボリュームグループで **vgcreate --setautoactivation n** コマンドを実行して、このフラグを **n** に設定します。Pacemaker クラスターで使用される既存のボリュームグループがある場合は、**vgchange --setautoactivation n** でフラグを設定します。

(BZ#1899214)

新しい pcs resource status 表示コマンド

pcs resource status コマンドおよび **pcs stonith status** コマンドで、以下のオプションが使用できるようになりました。

- **pcs resource status node=node_id** コマンドおよび **pcs stonith status node=node_id** コマンドを使用すると、特定ノードに設定したリソースの状態を表示できます。これらのコマンドを使用すると、クラスターとリモートノードの両方でリソースのステータスを表示できます。
- **pcs resource status resource_id** コマンドおよび **pcs stonith status resource_id** コマンドを使用すると、1 つのリソースの状況を表示できます。

- **pcs resource status tag_id** コマンドおよび **pcs stonith status tag_id** コマンドを使用すると、指定したタグで、すべてのリソースの状態を表示できます。

(BZ#1290830, BZ#1285269)

pcs resource safe-disable コマンド用の新しい縮小出力表示オプション

pcs resource safe-disable コマンドおよび **pcs resource disable --safe** コマンドは、エラーレポートの後に長いシミュレーション結果を出力します。これらのコマンドに、エラーのみを出力する **--brief** を指定できるようになりました。エラーレポートには、影響を受けるリソースのリソース ID が常に含まれるようになりました。

(BZ#1909901)

他のすべてのリソースを再起動せずに SCSI フェンシングデバイスを更新する新しい pcs コマンド

pcs stonith update コマンドを使用して SCSI フェンスデバイスを更新すると、stonith リソースが実行されているのと同じノードで実行中の全リソースを再起動することになります。新しい **pcs stonith update-scsi-devices** コマンドを使用すると、他のクラスターリソースを再起動せずに SCSI デバイスを更新できます。

(BZ#1872378)

クラスターノードのサブセットでフェンシング用にウォッチドッグのみの SBD を設定する機能

以前のバージョンでは、ウォッチドッグのみの SBD 設定を使用するには、クラスター内のすべてのノードで SBD を使用する必要がありました。一部のノードはサポートしているが、他のノード (リモートノード) では他のフェンシングが必要なクラスターで SBD が使用できませんでした。ユーザーは、新しい **fence_watchdog** エージェントを使用して、ウォッチドッグのみの SBD 設定を設定できるようになりました。これにより、一部のノードのみがフェンシングにウォッチドッグのみの SBD を使用し、その他のノードが他のフェンシングタイプを使用するクラスター設定が可能になります。クラスターはこのようなデバイスを 1 つしか持たず、これは **watchdog** という名前にする必要があります。

(BZ#1443666)

内部エラーの詳細なペースメーカーステータス表示

エージェントがインストールされていない、内部タイムアウトが発生したなど、何らかの理由で Pacemaker がリソースまたはフェンスエージェントを実行できない場合は、Pacemaker ステータス表示に内部エラーの詳細な終了理由が表示されるようになりました。

(BZ#1470834)

pcmk_delay_base パラメーターは、ノードごとに異なる値を取る可能性があります

フェンスデバイスを設定するときに、**pcmk_delay_base parameter** を使用してノードごとに異なる値を指定できるようになりました。これにより、ノードごとに異なる遅延を使用して、単一のフェンスデバイスを 2 ノードクラスターで使用できます。これは、各ノードが同時に他のノードをフェンスしようとする状況を防ぐのに役立ちます。ノードごとに異なる値を指定するには、pcmk_host_map と同様の構文を使用して、ホスト名をそのノードの遅延値にマップします。たとえば、node1:0;node2:10s は、node1 をフェンシングするときに遅延を使用せず、node2 をフェンシングするときに 10 秒の遅延を使用します。

(BZ#1082146)

pcmk_host_map 値内の特殊文字のサポート

pcmk_host_map プロパティーは、値の前に円記号 (\) を使用して、**pcmk_host_map** 値内の特殊文字をサポートするようになりました。たとえば、**pcmk_host_map="node3:plug\ 1"** を指定して、ホストエイリアスにスペースを含めることができます。

(BZ#1376538)

OpenShift 用の新しいフェンシングエージェント

現在、**fence_kubevirt** フェンシングエージェントは、Red Hat OpenShift Virtualization の RHEL High Availability で使用できます。**fence_kubevirt** エージェントの詳細については、**fence_kubevirt**(8) man ページを参照してください。

(BZ#1977588)

pcs cluster setup コマンドのローカルモードバージョンが完全にサポート

デフォルトでは、**pcs cluster setup** コマンドは、すべての設定ファイルをクラスターノードに自動的に同期します。**pcs cluster setup** コマンドが、**--corosync-conf** オプションに完全に対応するようになりました。このオプションを指定すると、コマンドが **local** モードに切り替わります。このモードでは、**pcs** コマンドラインインターフェイスは他のノードと通信せずに **corosync.conf** ファイルを作成し、ローカルノード上の指定されたファイルに保存します。これにより、スクリプトで **corosync.conf** ファイルを作成し、スクリプトでそのファイルを処理できます。

(BZ#2008558)

リソースの移動に伴う場所の制約の自動削除

pcs resource move コマンドを実行すると、現在実行しているノードでそれが実行されないように、制約がリソースに追加されます。デフォルトでは、リソースを移動すると、コマンドが作成する場所の制約が自動的に削除されます。このコマンドを実行しても、リソースが必ずしも元のノードに戻る訳ではありません。この時点でリソースが実行できる場所は、リソースの最初の設定方法によって異なります。リソースを移動し、その制約を適用したままにする場合は、**pcs resource move-with-contraint** を使用します。

(BZ#2008575)

OCF Resource Agent API 1.1 標準の pcs サポート

pcs コマンドラインインターフェイスは、OCF 1.1 リソースと STONITH エージェントをサポートするようになりました。このサポートの実装の一環として、エージェントのメタデータは OCF スキーマ (エージェントが OCF 1.0 または OCF 1.1 エージェントであるかに関係なく) に準拠する必要があります。エージェントのメタデータが OCF スキーマに準拠していない場合、**pcs** はエージェントが無効であると仮定し、**--force** オプションが指定されていない場合にエージェントのリソースを作成または更新しません。エージェントをリスト表示する **pcsd** Web UI および **pcs** コマンドは、リスト表示で無効なメタデータを持つエージェントを削除するようになりました。

(BZ#2018969)

pcs が、Promoted および Unpromoted をロール名として受け入れるようになる

Pacemaker 設定でロールが設定される場合、**pcs** コマンドラインインターフェイスで **Promoted** および **Unpromoted** を受け入れるようになりました。これらのロール名は、以前の RHEL リリースの Pacemaker ロール **Master** ロールおよび **Slave** と機能的に同等で、設定ディスプレイおよびヘルプページで確認できるロール名です。

(BZ#2009455)

pcsd Web UI のバージョンの更新

Pacemaker/Corosync クラスターを作成および設定するグラフィカルユーザーインターフェイスである **pcsd** Web UI が更新されました。更新された Web UI は、ユーザーエクスペリエンスの向上と、他の Red Hat Web アプリケーションで使用される PatternFly フレームワークで構築された標準化されたインターフェイスを提供します。

(BZ#1996067)

4.13. 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

RHEL 9 の Python

Python 3.9 が RHEL 9 におけるデフォルトの **Python** 実装に**Python 3.9** は、BaseOS リポジトリーにあるモジュール以外の **python3** RPM パッケージで配布され、通常はデフォルトでインストールされます。**Python 3.9** は、RHEL 9 のライフサイクル全体で対応します。

追加バージョンの **Python 3** は、AppStream リポジトリーを介してより短いライフサイクルで RPM パッケージとして配布され、並行してインストールできます。

python コマンド (**/usr/bin/python**) や、**pip** などの **Python** 関連コマンドは、バージョンを指定せずに使用でき、デフォルトの **Python 3.9** を指定します。

Python 2 は RHEL 9 に同梱されていません。

RHEL 9 の **Python** の詳細は、Python の概要 を参照してください。

(BZ#1941595, JIRA:RHELPLAN-80598)

RHEL 9 で利用可能な Node.js 16

RHEL 9 は、Long Term Support (LTS) バージョン 16 の **Node.js** を提供します。これは、JavaScript プログラミング言語を使用して高速でスケーラブルなネットワークアプリケーションを構築するソフトウェア開発プラットフォームです。

Node.js 14 に対する **Node.js 16** の主な変更点は、以下のとおりです。

- **V8** エンジンがバージョン 9.4 にアップグレードされました。
- **npm** パッケージマネージャーがバージョン 8.3.1 にアップグレードされました。
- 新しい **Timers Promises** API は、**Promise** オブジェクトを返すタイマー関数の代替セットを提供します。
- **Node.js** が **OpenSSL 3.0** と互換性があるようになりました。
- **Node.js** は、実験的な新しい **Web Streams** API と実験的な ECMAScript モジュール (ESM) ローダーフック API を提供するようになりました。

Node.js 16 は、このアプリケーションストリームの初版で、RPM パッケージとして簡単にインストールできます。**Node.js 16** のライフサイクルは RHEL 9 よりも短くなります。詳細は、Red Hat Enterprise Linux Application Streams ライフサイクル を参照してください。また、追加の **Node.js** バージョンは、RHEL 9 の今後のマイナーリリースで、ライフサイクルが短いモジュールとして提供されます。

(BZ#1953491)

RHEL 9 が Ruby 3.0 を提供

RHEL 9 には **Ruby 3.0.3** が同梱されており、**Ruby 2.7** とともにパフォーマンスの改善、バグおよびセキュリティーの修正、ならびに新しい機能が数多く提供されます。

主な機能拡張は、次のとおりです。

- 同時実行機能および並列処理機能:
  - **Ractor** が実験的な機能として提供されています。この機能はスレッドセーフな並列実行を提供するアクターモデルの抽象化です。
  - **Fiber Scheduler** が実験的な機能として導入されました。**Fiber Scheduler** はブロッキング操作をインターセプトし、既存のコードを変更せずに軽量の同時実行を可能にします。
- 静的な分析機能:
  - **Ruby** プログラムの設定を説明する **RBS** が導入されました。**RBS** で書かれた型定義を解析するために **rbs** gem が追加されました。
  - **Ruby** 符号の型解析ツールである **TypeProf** ユーティリティーが導入されました。
- **case/in** 式を使用したパターンの照合は実験的なものではありません。
- 実験的な機能である 1 行パターンの一致が再設計されました。
- 検索パターンが実験的な機能として追加されました。

以下のパフォーマンスの向上が実装されています。

- **Interactive Ruby Shell (IRB)** への長いコードの貼り付けが大幅に高速になりました。
- **IRB** に時間計測用の **measure** コマンドが追加されました。

その他の主な変更点は次の通りです。

- キーワード引数とその他の引数が切り替わりました。
- **$HOME/.gem/** ディレクトリーがすでに存在しない限り、ユーザーがインストールした gems のデフォルトディレクトリーが **$HOME/.local/share /gem/** になります。

Ruby 3.0 は、このアプリケーションストリームの初版で、RPM パッケージとして簡単にインストールできます。追加の **Ruby** バージョンは、RHEL 9 の今後のマイナーリリースで、ライフサイクルが短いモジュールとして提供されます。

(JIRA:RHELPLAN-80758)

RHEL 9 に Perl 5.32 が導入されました。

RHEL 9 には **Perl 5.32** が含まれており、バージョン 5.30 に対するバグ修正や機能強化が数多く追加されました。

主な機能強化は、次のとおりです。

- **Perl** は Unicode バージョン 13.0 に対応するようになりました。
- **qr** 引用符のような演算子が強化されました。

- **POSIX::mblen()**、**mbtowc**、**wctomb** 関数がシフト状態のロケールで動作するようになり、ロケールのスレッドセーフ機能を持つプラットフォームで実行した場合に、C99 以上のコンパイラーではスレッドセーフになりました。長さのパラメーターはオプションになりました。
- 新しい実験的な **isa**infix 演算子は、与えられたオブジェクトが、与えられたクラスのインスタンスであるか、そこから派生したクラスであるかをテストします。
- アルファアサーションはもはや実験的なものではありません。
- スクリプトの実行は実験的なものではなくなりました。
- 機能チェックが速くなりました。
- **Perl** は最適化の前にコンパイルされたパターンをダンプできるようになりました。

Perl 5.32 は、このアプリケーションストリームの初版で、RPM パッケージとして簡単にインストールできます。追加の **Perl** バージョンは、RHEL 9 の今後のマイナーリリースで、ライフサイクルが短いモジュールとして提供されます。

(JIRA:RHELPLAN-80759)

RHEL 9 には PHP 8.0 が含まれるようになりました。

RHEL 9 には **PHP 8.0** が使用されており、バージョン 7.4 に対するバグ修正や機能強化が数多く追加されました。

主な機能拡張は、次のとおりです。

- 新しい名前付き引数は順序に依存せず、自己ドキュメント化されており、必要なパラメーターのみを指定できるようになりました。
- 新しい属性により、PHP のネイティブ構文で構造化メタデータを使用できます。
- 新しいユニオンタイプにより、複数のタイプの組み合わせに対して、PHPDoc アノテーションの代わりに実行時に検証されるネイティブのユニオンタイプ宣言を使用できます。
- 内部関数は、パラメーターの検証に失敗した場合に警告ではなく、Error 例外を常に発生させるようになりました。
- 新しい Just-In-Time コンパイルエンジンにより、アプリケーションのパフォーマンスが大幅に向上します。
- PHP の **Xdebug** デバッグおよび生産性拡張機能がバージョン 3 に更新されました。このバージョンでは、**Xdebug 2** と比較した機能および設定に大きな変更が加えられました。

PHP 8.0 は、このアプリケーションストリームの初版で、RPM パッケージとして簡単にインストールできます。追加の **PHP** バージョンは、RHEL 9 の今後のマイナーリリースで、ライフサイクルが短いモジュールとして提供されます。

詳細は PHP スクリプト言語の使用 を参照してください。

(BZ#1949319)

RHEL 9 は、 Git 2.31 および Git LFS 2.13 を提供します

RHEL 9 には、RHEL 8 で利用可能なバージョン 2.27 よりも多くの機能拡張とパフォーマンス改善を行う **Git 2.31** が同梱されています。主な変更点は、以下のとおりです。

- **git status** コマンドが、スパースチェックアウトの状態を報告するようになりました。
- **git archive** で **--add-file** を使用できるようになりました。これで、トラッキングされていないファイルを、tree-ish 識別子からスナップショットに組み込むことができるようになりました。
- **clone.defaultremotename** 設定変数を使用すると、ソースリモートリポジトリーのニックネームをカスタマイズできます。
- **git format-patch** で作成する出力ファイル名の上限を設定できます。以前は、長さの制限は 64 バイトでした。
- 非推奨の PCRE1 ライブラリーのサポートが削除されました。

また、**Git Large File Storage (LFS)** 拡張機能バージョン 2.13 が利用できるようになりました。RHEL 8 に同梱されるバージョン 2.11 以降の機能拡張には、以下が含まれます。

- **Git LFS** が SHA-256 リポジトリーに対応するようになりました。
- **Git LFS** が、**socks5h** プロトコルに対応するようになりました。
- **git lfs install** コマンドおよび **git lfs uninstall** コマンドには、新しい **--worktree** オプションが用意されています。
- **git lfs migrate import** では、新しい **--above** パラメーターを使用できます。

(BZ#1956345, BZ#1952517)

RHEL 9 の Subversion 1.14

RHEL 9 には **Subversion 1.14** が同梱されています。**Subversion 1.14** は、このアプリケーションストリームの初版で、RPM パッケージとして簡単にインストールできます。追加の **Subversion** バージョンは、RHEL 9 の今後のマイナーリリースで、ライフサイクルが短いモジュールとして提供されます。

(JIRA:RHELPLAN-82578)

Apache HTTP Server への主な変更点

RHEL 9.0 は、Apache HTTP Server のバージョン 2.4.51 を提供します。バージョン 2.4.37 への主な変更点は、以下のとおりです。

- Apache HTTP Server Control Interface (**apachectl**):
  - **apachectl status** 出力では、**systemctl** ページャーが無効になりました。
  - 追加の引数を渡すと警告が表示される代わりに、**apachectl** コマンドが失敗するようになりました。
  - **apachectl graceful-stop** がすぐに戻るようになりました。
  - **apachectl configtest** コマンドが、SELinux コンテキストを変更せずに、**httpd -t** コマンドを実行するようになりました。
  - RHEL の **apachectl(8)** man ページで、アップストリームの **apachectl** との相違点が完全に説明されるようになりました。
- Apache eXtenSion ツール (**apxs**):
  - **/usr/bin/apxs** コマンドは、**httpd** パッケージのビルド時に適用されたコンパイラーの最適

化フラグを使用または公開しなくなりました。**/usr/lib64/httpd/build/vendor-apxs** コマンドを使用して、**httpd** のビルドに使用されるのと同じコンパイラーフラグを適用できるようになりました。**vendor-apxs** コマンドを使用するには、最初に **redhat-rpm-config** パッケージをインストールする必要があります。

- Apache モジュール:
  - **mod_lua** モジュールが、別のパッケージで提供されるようになりました。
  - Apache HTTP Server 用の新しい **mod_jk** コネクターは、Apache JServ Protocol (AJP) を使用して Web サーバーを Apache Tomcat およびその他のバックエンドに接続するモジュールです。
  - 新しい **mod_proxy_cluster** モジュールは、通信チャネルを使用してリクエストをロードバランサーからアプリケーションサーバーノードの 1 つに転送する httpd ベースのロードバランサーを提供します。アプリケーションサーバーノードは、この接続を使用してサーバー側の負荷分散係数およびライフサイクルイベントを MCMP(Mod-Cluster Management Protocol) と呼ばれる HTTP メソッドのカスタムセットを使用してロードバランサーに戻します。この追加のフィードバックチャネルを使用すると、**mod_proxy_cluster** は他の負荷分散ソリューションで見つからない情報および粒度のレベルを提供できます。このモジュールでは、**ModCluster** クライアントをバックエンドサーバーにインストールして正常に通信する必要があります。
- 設定構文の変更
  - **mod_access_compat** が提供する非推奨の **Allow** ディレクティブでは、コメント (**#** 文字) が暗黙的に無視される代わりにシンタックスエラーを発生するようになりました。
- その他の変更:
  - カーネルスレッド ID は、エラーログメッセージで直接使用されるようになり、精度と簡潔性が向上しました。
  - 多くのマイナーな機能強化とバグ修正
  - モジュールの作成者には、利用可能な新しいインターフェイスが多数用意されています。

RHEL 8 以降、**httpd** モジュール API に後方互換性のない変更はありません。

Apache HTTP Server 2.4 は、この Apache HTTP Server 2.4 の初期バージョンです。これは、RPM パッケージとして簡単にインストールできます。

詳細は Apache HTTP Web サーバーの設定 を参照してください。

(JIRA:RHELPLAN-68364, BZ#1931976, JIRA:RHELPLAN-80725)

RHEL 9 で利用可能な nginx 1.20

RHEL 9 には、**nginx 1.20** Web およびプロキシーサーバーが同梱されています。このリリースでは、バージョン 1.18 に対するバグ修正、セキュリティー修正、新機能、および機能拡張が数多く提供されます。

新機能:

- **nginx** が、OCSP (Online Certificate Status Protocol) を使用したクライアント SSL 証明書の検証に対応するようになりました。

- **nginx** が、最小限の空き領域に基づくキャッシュクリアに対応するようになりました。これに対応するのは、**proxy_cache_path** ディレクティブの **min_free** パラメーターとして実装されています。
- 新しい **ngx_stream_set_module** モジュールが追加されました。これにより、変数の値を設定できるようになりました。
- **nginx** 用の外部動的モジュールを構築するための、RPM マクロや **nginx** ソースコードを含む、必要なすべてのファイルを提供する **nginx-mod-devel** パッケージが追加されました。

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。

HTTP/2 のサポートが改善されました。

- **ngx_http_v2** モジュールに、**lingering_close** ディレクティブ、**lingering_time** ディレクティブ、**lingering_timeout** ディレクティブが含まれるようになりました。
- HTTP/2 での接続の処理は、HTTP/1.x に合わせて行われました。**nginx 1.20** では、削除した **http2_recv_timeout** ディレクティブ、**http2_idle_timeout** ディレクティブ、および **http2_max_requests** ディレクティブの代わりに、**keepalive_timeout** ディレクティブおよび **keepalive_requests** ディレクティブを使用します。

nginx 1.20 は、このアプリケーションストリームの初版で、RPM パッケージとして簡単にインストールできます。追加の **nginx** バージョンは、RHEL 9 の今後のマイナーリリースで、ライフサイクルが短いモジュールとして提供されます。

詳細は、NGINX のセットアップと設定 を参照してください。

(BZ#1953639, BZ#1991720)

RHEL 9 の Varnish Cache 6.6

RHEL 9 には、高パフォーマンスの HTTP リバースプロキシーである **Varnish Cache 6.6** が含まれます。

バージョン 6.0 以降の主な変更点は、以下のとおりです。

- **varnishlog** などのログ処理ツールのパフォーマンスが改善
- 統計の精度の向上
- キャッシュ検索における多くの最適化
- さまざまな設定変更
- 数多くの機能強化およびバグ修正

Varnish Cache 6 は、このアプリケーションストリームの初版で、RPM パッケージとして簡単にインストールできます。

(BZ#1984185)

RHEL 9 に Squid 5 が導入されました。

RHEL 9 には、Web クライアント、対応する FTP、Gopher、および HTTP データオブジェクト向けの高パフォーマンスのプロキシーキャッシュサーバーである **Squid 5.2** が同梱されています。本リリースでは、バージョン 4 に対するバグ修正、セキュリティー修正、新機能、および機能強化が数多く追加されました。

新機能:

- **Squid** は、ハッピーアイボールアルゴリズムを使用することで機能を改善します。
  - **Squid** は、潜在的な転送先がすべて解決されるのを待つ代わりに、リクエスト転送が必要になるとすぐに、受信した IP アドレスを使用するようになりました。
  - 新しいディレクティブ (**happy_eyeballs_connect_gap** ディレクティブ、**happy_eyeballs_connect_limit** ディレクティブ、および **happy_eyeballs_connect_timeout** ディレクティブ) が利用できるようになりました。
  - **dns_v4_first** ディレクティブが削除されました。
- **Squid** は、コンテンツ配信ネットワーク (CDN) でループ検出のソースとして **CDN-Loop** ヘッダーを使用するようになりました。
- **Squid** では、SSL バンプに対するピアサポートが導入されました。
- 新しい Internet Content Adaptation Protocol (ICAP) トレイラー機能が利用可能になりました。これにより、ICAP エージェントはメッセージボディー後にメッセージメタデータを確実に送信できるようになりました。

設定オプションの変更:

- **clientside_mark** の代わりに、**mark_client_packet** 設定が使用されました。
- **collapsed_forwarding_shared_entries_limit** の代わりに、**shared_transient_entries_limit** 設定が使用されました。

Squid 5 は、このアプリケーションストリームの初版で、RPM パッケージとして簡単にインストールできます。

詳細は、Configuring the Squid caching proxy server を参照してください。

(BZ#1990517)

RHEL 9 の MariaDB 10.5

RHEL 9 は、**MariaDB 10.5** を提供します。**MariaDB 10.5** は、このアプリケーションストリームの初版で、RPM パッケージとして簡単にインストールできます。追加の **MariaDB** バージョンは、RHEL 9 の今後のマイナーリリースで、ライフサイクルが短いモジュールとして提供されます。

詳しくは、Using MariaDB をご覧ください。

(BZ#1971248)

RHEL 9 に MySQL 8.0 が含まれる

RHEL 9 には **MySQL 8.0** が同梱されています。**MySQL 8.0** は、このアプリケーションストリームの初版で、RPM パッケージとして簡単にインストールできます。**MySQL 8.0** のライフサイクルは RHEL 9 よりも短くなります。詳細は、Red Hat Enterprise Linux Application Streams ライフサイクル を参照してください。

使用方法は、MySQL の使用 を参照してください。

(JIRA:RHELPLAN-78673)

RHEL 9 が提供する PostgreSQL 13

PostgreSQL 13 が RHEL 9 で利用可能 **PostgreSQL 13** は、このアプリケーションストリームの初版で、RPM パッケージとして簡単にインストールできます。追加の **PostgreSQL** バージョンは、RHEL 9 の今後のマイナーリリースで、ライフサイクルが短いモジュールとして提供されます。

詳しくは、Using PostgreSQL をご覧ください。

(JIRA:RHELPLAN-78675)

RHEL 9 の Redis 6.2

RHEL 9 には **Redis 6.2** が同梱されており、RHEL 8 で利用可能なバージョン 6.0 以降でバグ修正およびセキュリティー機能拡張が数多く追加されています。

特に、**Redis** サーバーの設定ファイルは、専用のディレクトリー (**/etc/redis/redis.conf** および **/etc/redis/sentinel.conf**) に置かれています。RHEL 8 では、このようなファイルの場所は、それぞれ **/etc/redis.conf** と **/etc/redis-sentinel.conf** になりました。

Redis 6 は、このアプリケーションストリームの初版で、RPM パッケージとして簡単にインストールできます。

(BZ#1959756)

新しいパッケージ: perl-Module-Signature

RHEL 9 では、Perl モジュール **perl-Module-Signature** が導入されました。この新しいモジュールを使用すると、**cpan** の署名チェックを有効にして CVE-2020-16156 を軽減できます。詳細は、 How to mitigate CVE-2020-16154 in perl-App-cpanminus and CVE-2020-16156 in perl-CPAN を参照してください。

(BZ#2039361)

4.14. コンパイラーおよび開発ツール

RHEL 9 は、IBM POWER10 プロセッサーへの対応を提供します。

Linux カーネルからシステムツールチェーン (GCC、binutils、glibc) を介して、IBM の最新の POWER プロセッサーである POWER10 のサポートが含まれるように Red Hat Enterprise Linux 9 が更新されました。RHEL 9 は、POWER10 でワークロードに対応するため、今後のリリースで機能拡張が提供されています。

(BZ#2027596)

GCC 11.2.1 が利用できる。

RHEL 9 には、GCC バージョン 11.2.1 が同梱されています。主なバグ修正と機能拡張は、以下のとおりです。

一般的な改善

- GCC は、DWARF バージョン 5 のデバッグフォーマットをデフォルトで使用するようになりました。

- 診断で表示される列番号は、デフォルトでは実際の列番号を表し、複数列の文字を尊重します。
- 直線コードベクターライザは、機能全体を考慮してベクターリングを行います。
- 同じ変数を比較する一連の条件式は、それぞれに比較式が含まれていれば、switch ステートメントに変換することができます。
- プロシージャー間の最適化の改善:
  - **fipa-modref** オプションで制御される新しい IPA-modref パスは、関数呼び出しの副作用を追跡し、ポイントツー分析の精度を向上させます。
  - **fipa-icf** オプションで制御される同一コードのフォールディングパスが大幅に改善され、統一された関数の数が増え、コンパイル時のメモリー使用量が削減されました。
- リンクタイム最適化の改善:
  - Link-time optimization (LTO) を使用すると、コンパイラーは、リンク時に中間表現を使用して、プログラムのすべての変換ユニットでさまざまな最適化を実行できます。詳しくは Link time optimization をご覧ください。
  - リンク時のメモリー割り当てを改善し、ピークのメモリー使用量を削減しました。
- IDE で新しい **GCC_EXTRA_DIAGNOSTIC_OUTPUT** 環境変数を使用すると、ビルドフラグを調整することなく、機械的に読める修正のヒントを要求することができます。
- **-fanalyzer** オプションで実行されるスタティックアナライザーが大幅に改善され、多数のバグフィックスと機能強化が行われました。

言語固有の改善

C ファミリー

- C および C++ コンパイラーは、OpenMP 5.0 仕様の OpenMP コンストラクトおよびアロケータルーチンにおいて、非矩形のループネストをサポートしています。
- 属性:
  - 新しい **no_stack_protector** 属性は、スタック保護 **(-fstack-protector)** をかけてはいけない関数を示します。
  - 改良された **malloc** 属性は、アロケータとデアロケータの API ペアを識別するために使用することができます。
- 新しい警告:
  - **-Wsizeof-array-div**(**-Wall** オプションで有効) は、2 つの **sizeof** 演算子の除算について、最初の演算子が配列に適用され、除算値が配列要素のサイズと一致しない場合に警告を発します。
  - **-Wstringop-overread** は、デフォルトで有効になっており、引数として渡された配列の最後を超えて読み取ろうとする文字列関数の呼び出しについて警告します。
- 警告の強化:
  - **-Wfree-nonheap-object** は、動的メモリー割り当て関数から返されていないポインターを使用した割り当て解除関数の呼び出しのインスタンスをより多く検出します。

- **-Wmaybe-uninitialized** は、初期化されていないメモリーへのポインターや参照が、**const**-qualified 引数を取る関数に渡すことを診断します。
- **-Wuninitialized** は、初期化されていない動的に割り当てられたメモリーからの読み取りを検出します。

C

- **-std=c2x** および **-std=gnu2x** オプションにより、ISO C 規格の次期 C2X 改訂版の新機能がサポートされています。以下に例を示します。
  - 標準属性がサポートされています。
  - **__has_c_attribute** プリプロセッサー演算子がサポートされています。
  - ラベルは、宣言の前や複合ステートメントの最後に表示されることがあります。

C++

- デフォルトのモードは **-std=gnu++17** に変更されます。
- C++ライブラリーの **libstdc++** では、C++17 のサポートが改善されました。
- C++20 の新機能がいくつか実装されています。なお、C++20 のサポートは実験的なものです。各機能の詳細については、C++20 Language Features を参照してください。
- C++フロントエンドは、今後予定されている C++23 ドラフト機能の一部を実験的にサポートしています。
- 新しい警告:
  - **-Wctad-maybe-unsupported** はデフォルトでは無効で、控除ガイドのない型でクラステンプレート引数の控除を行うことについて警告します。
  - **-Wrange-loop-construct** は、**-Wall** で有効になり、範囲ベースの for ループが不必要でリソース効率の悪いコピーを作成している場合に警告を発します。
  - **-Wmismatched-new-delete** は **-Wall** で有効になり、不一致な形式の new 演算子や他の不一致な割り当て関数から返されたポインターを持つ delete 演算子の呼び出しについて警告します。
  - **-Wvexing-parse** はデフォルトで有効になっており、最も厄介な構文解析ルールを警告します。つまり、宣言が変数定義のように見えても、C++言語では関数宣言として解釈される必要がある場合です。

アーキテクチャー固有の改善

64 ビット ARM アーキテクチャー

- Armv8-R アーキテクチャーは、**-march=armv8-r** オプションでサポートされています。
- GCC は、加算、減算、乗算、および複素数の累積と減算を行う演算を自動ベクトル化することができます。

AMD アーキテクチャーおよび Intel 64 ビットアーキテクチャー

- Intel AVX-VNNI の新しい ISA 拡張サポートが追加されました。**-mavxvnni** コンパイラースイッチは、AVX-VNNI の組込みを制御します。

- znver3 コアを搭載した AMD CPU は、新たな **-march=znver3** オプションによりサポートされます。
- x86-64 psABI サプリメント で定義されている 3 つのマイクロアーキテクチャーレベルは、新しい **-march=x86-64-v2**、**-march=x86-64-v3**、および **-march=x86-64-v4** オプションでサポートされています。

IBM Z アーキテクチャー

- GCC 11.2.1 はデフォルトで IBM z14 プロセッサーに設定されます。

IBM Power Systems

- GCC 11.2.1 はデフォルトで IBM POWER9 プロセッサーに設定されます。
- GCC コンパイラーが、新しい **-mcpu=power10** コマンドラインオプションを使用して POWER10 命令をサポートするようになりました。

(BZ#1986836, BZ#1870016, BZ#1870025, BZ#1870028, BZ#2019811, BZ#2047296)

glibc 最適化データをキャプチャーするための新しいコマンド

新しい **ld.so --list-diagnostics** コマンドは、IFUNC の選択や **glibc-hwcaps** の設定など、**glibc** の最適化の決定に影響を与えるデータを単一の機械可読ファイルにキャプチャーします。

(BZ#2023422)

binutils への主な変更点

RHEL 9 では、**binutils** に以下の変更が加えられています。

- **binutils** は Intel の AMX/TMUL 命令セットに対応するようになり、この新機能を活用できるアプリケーションのパフォーマンスが向上しました。
- アセンブラー、リンカー、およびその他のバイナリーユーティリティーが、POWER10 命令をサポートするようになりました。

(BZ#2030554, BZ#1870021)

sched_getcpu の実装では、オプションで rseq (再起動可能なシーケンス) を使用して、64 ビット ARM アーキテクチャーおよびその他のアーキテクチャーでのパフォーマンスを向上できるようになりました。

64 ビット ARM アーキテクチャーでの **sched_getcpu** の以前の実装では、**getcpu** システムコールを使用しますが、これはほとんどの並列アルゴリズムで効率的に使用するには遅すぎます。他のアーキテクチャーでは、vDSO (仮想動的共有オブジェクト) アクセラレーションを使用してこれを回避します。**rseq** を使用して **sched_getcpu** を実装すると、64 ビット ARM アーキテクチャーのパフォーマンスが大幅に改善されます。他のアーキテクチャーではわずかな改善が見られます。

sched_getcpu が **rseq** を使用するように設定するには、**GLIBC_TUNABLES=glibc.pthread.rseq=1** 環境変数を設定します。

```
# GLIBC_TUNABLES=glibc.pthread.rseq=1
# export GLIBC_TUNABLES
```

(BZ#2024347)

パフォーマンスツールとデバッガーの更新

RHEL 9.0 では、以下のパフォーマンスツールおよびデバッガーが利用できます。

- GDB 10.2
- Valgrind 3.18.1
- SystemTap 4.6
- Dyninst 11.0.0
- elfutils 0.186

(BZ#2019806)

IBM POWER10 の GDB で DAWR 機能が改善

RHEL 9 には、改善された DAWR 機能を提供する GDB 10.2 が同梱されています。IBM POWER10 プロセッサーでは、新しいハードウェアウォッチポイント機能が GDB で有効になりました。たとえば、DAWR/DAWRX レジスターの新しいセットが追加されました。

(BZ#1870029)

GDB が、IBM POWER10 で接頭辞が付けられた新しい命令に対応

GDB 10.2 は、POWER10 で Power ISA 3.1 の接頭辞が付いた命令に完全に対応しています。これには、8 バイトの接頭辞が付いた命令が含まれます。RHEL 8.4 では、GDB は 4 バイトの命令のみに対応していました。

(BZ#1870031)

RHEL 9 は ブースト 1.75.0 を提供します

RHEL 9 には、**boost** パッケージのバージョン 1.75.0 が同梱されています。バージョン 1.67.0 への主なバグ修正および機能強化は、以下のとおりです。

- **Boost.Signals** ライブラリーが削除され、header-only **Boost.Signals2** コンポーネントに置き換えられました。
- **boost-jam** パッケージの **bjam** ツールは、**boost-b2** パッケージの **b2** に置き換えられました。
- 新しいライブラリー:
  - **Boost.Contracts**
  - **Boost.HOF**
  - **Boost.YAP**
  - **boost.Safe Numerics**
  - **Boost.Outcome**
  - **Boost.Histogram**
  - **Boost.Variant2**
  - **Boost.Nowide**

- **Boost.StaticString**
- **Boost.STL_Interfaces**
- **boost.JSON**
- **Boost.LEAF**
- **Boost.PFR**

(BZ#1957950)

RHEL 9 は LLVM Toolset 13.0.1 を提供します。

RHEL 9 には、LLVM Toolset バージョン 13.0.1 が同梱されています。バージョン 12.0.1 への主なバグ修正および機能強化は、以下のとおりです。

- Clang は、C++ のステートメント属性 **[[clang::musttail]]** および C の **__attribute__((musttail))** をサポートするようになりました。
- Clang は、**-Wreserved-identifier** 警告をサポートするようになりました。これは、コードで予約済み識別子を使用するときに開発者に警告します。
- Clang の **-Wshadow** フラグは、シャドウされた構造化バインディングもチェックするようになりました。
- Clang の **-Wextra** は、**-Wnull-pointer-subtraction** も意味するようになりました。
- Clang は、C++ のステートメント属性 **[[clang::musttail]]** および C の **__attribute__((musttail))** をサポートするようになりました。

RHEL 9 では、**llvm-toolset** を RPM パッケージとして簡単にインストールできます。

(BZ#2001107)

CMake 3.20.2 における主な変更点

RHEL 9 には CMake 3.20.2 が同梱されています。バージョン 3.20.2 以前を必要とするプロジェクトで CMake を使用するには、**cmake_minimum_required**(バージョン 3.20.2) コマンドを使用します。

主な変更点は、以下のとおりです。

- C++23 コンパイラーモードは、ターゲットプロパティー **CXX_STANDARD**、**CUDA_STANDARD**、**OBJCXX_STANDARD**、またはコンパイル機能の **cxx_std_23** メタ機能を使用して指定できるようになりました。
- CUDA 言語サポートにより、NVIDIA CUDA コンパイラーをシンボリックリンクにすることができるようになりました。
- Intel oneAPI NextGen LLVM コンパイラー ID が、**IntelLLVM** コンパイラー ID で対応されるようになりました。
- Cmake は、Android NDK のツールチェーンファイルとマージすることで、Android のクロスコンパイルを容易にします。
- **cmake(1)** を実行してプロジェクトビルドシステムを生成すると、ハイフンで始まる不明なコマンドライン引数が拒否されるようになりました。

新機能および非推奨になった機能の詳細は、CMake Release Notes を参照してください。

(BZ#1957948)

RHEL 9 では Go 1.17.7 が提供されます。

RHEL 9 には、Go Toolset バージョン 1.17.7 が同梱されています。バージョン 1.16.7 への主なバグ修正および機能強化は、以下のとおりです。

- スライスを配列ポインターに変換するオプションを追加しました。
- //go:build 行のサポートを追加しました。
- amd64 での関数呼び出しのパフォーマンスを改善しました。
- 関数引数を、スタックトレースでより明確にフォーマットしました。
- クロージャーを含む関数をインライン化できるようにしました。
- x509 証明書の解析におけるリソース消費を削減しました。

RHEL 9 では、**go-toolset** を RPM パッケージとして簡単にインストールできます。

(BZ#2014087)

Go FIPS モードは OpenSSL 3 でサポートされています。

Go FIPS モードの場合に OpenSSL 3 ライブラリーを使用できるようになりました。

(BZ#1984110)

RHEL 9 は Rust Toolset 1.58.1 を提供

RHEL 9 には、Rust Toolset バージョン 1.58.1 が同梱されています。バージョン 1.54.0 への主なバグ修正および機能強化は、以下のとおりです。

- Rust コンパイラーは、クロージャーの disjoint キャプチャー、配列の **IntoIterator**、新しい Cargo 機能リゾルバーなどを備えた 2021 年版の言語をサポートするようになりました。
- 新しいカスタムプロファイルの Cargo サポートが追加されました。
- Cargo はコンパイラーエラーを重複排除します。
- 新しいオープンレンジパターンが追加されました。
- フォーマット文字列にキャプチャーされた識別子を追加しました。

詳細は、Rust 1.55 Rust 1.56 Rust 1.57 Rust 1.58 を参照してください。

RHEL 9 では、RPM パッケージとして **rust-toolset** を簡単にインストールできます。

(BZ#2002885)

RHEL 9 は、 pcp パッケージのバージョン 5.3.5 を提供します。

RHEL 9 には、Performance Co-Pilot (**pcp**) パッケージバージョン 5.3.5 が同梱されています。バージョン 5.3.1 以降、新しい **pcp-pmda-bpf** サブパッケージが追加され、BPF CO-RE (**libbpf** および **BTF**) を利用する **eBPF** プログラムからのパフォーマンスデータを提供します。

(BZ#1991764)

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

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

(BZ#1847808)

新しい pcp-ss PCP ユーティリティーが利用可能に

pcp-ss PCP ユーティリティーは、**pmdasockets(1)** PMDA が収集したソケット統計を報告します。このコマンドは、多くの **ss** コマンドラインオプションとレポート形式と互換性があります。また、ライブモードのローカルまたはリモート監視と、以前に記録された PCP アーカイブからの過去の再生の利点も提供します。

(BZ#1981223)

RHEL 9 は grafana 7.5.11 を提供します。

RHEL 9 には、**grafana** パッケージバージョン 7.5.11 が同梱されています。バージョン 7.5.9 への主な変更点は、以下のとおりです。

- 新しいデータフレーム形式をサポートしないパネルの下位互換性のために、新しい **prepare time series** 変換が追加されました。
- パスワードリセットトークンを生成するために、SHA-1 の代わりに HMAC-SHA-256 を使用するようにパスワードリカバリー機能を更新。

(BZ#1993215)

RHEL 9 は grafana-pcp 3.2.0 を提供します。

RHEL 9 には、**grafana-pcp** パッケージバージョン 3.2.0 が同梱されています。バージョン 3.1.0 への主なバグ修正および機能強化は、以下のとおりです。

- PCP Redis 用の新しい MS SQL サーバーダッシュボードに追加
- PCP Vector eBPF/BCC 概要ダッシュボードに空のヒストグラムバケットの可視性を追加しました。
- PCP Redis の **metric()** 関数がすべてのメトリック名を返さないバグを修正しました。

(BZ#1993156)

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

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

(BZ#1845592)

新規パッケージ: ansible-pcp

ansible-pcp パッケージには、PCP (Performance Co-Pilot) および **metrics** RHEL システムロールの実装に使用される Redis や Grafana などの関連ソフトウェアロールが含まれます。

(BZ#1957566)

RHEL 9 は python-jsonpointer 2.0 を提供します。

RHEL 9 には、**python-jsonpointer** パッケージバージョン 2.0 が同梱されています。

バージョン 1.9 への主な変更点は、以下のとおりです。

- Python バージョン 2.6 および 3.3 は非推奨になりました。
- **python-jsonpointer** モジュールは、無効なエスケープシーケンスのポインターを自動的に確認するようになりました。
- コマンドラインでポインターを引数として記述できるようになりました。
- URL エンコード形式でポインターを送信することはできなくなりました。

(BZ#1980256)

.NET 6.0 が利用可能です。

RHEL 9 には **.NET** バージョン 6.0 が同梱されています。以下は、主な改善点です。

- 64 ビット Arm (aarch64) に対応
- IBM Z および LinuxONE (s390x) に対応

詳細は、.NET 6.0 RPM パッケージリリースノート および .NET 6.0 コンテナーリリースノート を参照してください。

.NET 6.0 は、この Application Stream の初期バージョンであり、RPM パッケージとして簡単にインストールできます。**.NET 6.0** では、RHEL 9 よりもライフサイクルが短くなります。詳細は、 Red Hat Enterprise Linux Application Streams ライフサイクル を参照してください。

(BZ#1986211)

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。

詳細は、OpenJDK のドキュメント を参照してください。

(BZ#2021262)

RHEL 9 の Java ツール

RHEL 9 AppStream リポジトリーには、以下の Java ツールが同梱されています。

- ソフトウェアプロジェクトマネジメントおよび理解ツールである **Maven 3.6.3**

- Java アプリケーションのコンパイル、アセンブル、テスト、および実行を行う Java ライブラリーおよびコマンドラインツールである **Ant 1.10.9**

Maven3.6 と **Ant1.10** は、これらのアプリケーションストリームの初期バージョンであり、非モジュラー RPM パッケージとして簡単にインストールできます。

(BZ#1951482)

CRB リポジトリーで利用可能な SWIG 4.0

Simplified Wrapper and Interface Generator (SWIG) バージョン 4.0 は、CodeReady Linux Builder (CRB) リポジトリーで利用できます。本リリースでは、**PHP 8** のサポートが追加されました。

RHEL 9 では、**SWIG** を RPM パッケージとして簡単にインストールできます。

CodeReady Linux Builder リポジトリーに含まれるパッケージには対応しないことに注意してください。

(BZ#1943580)

4.15. IDENTITY MANAGEMENT

Directory Server はグローバル changelog を使用しなくなりました

この機能強化により、Directory Server changelog がメインのデータベースに統合されました。以前は、Directory Server はグローバル changelog を使用していました。ただし、ディレクトリーが複数のデータベースを使用した場合は、問題が発生する可能性がありました。その結果、各接尾辞には、通常のデータベースファイルと同じディレクトリーに独自の changelog が含まれるようになりました。

(BZ#1805717)

すべての依存関係を持つ AppStream リポジトリーで ansible-freeipa が利用できるようになりました。

以前の RHEL 8 では、**ansible-freeipa** パッケージをインストールする前に、まず Ansible リポジトリーを有効にして **ansible** パッケージをインストールする必要がありました。RHEL 8.6 および RHEL 9 では、準備手順なしで **ansible-freeipa** をインストールできます。**ansible-freeipa** をインストールすると、依存関係として、**ansible** のより基本的なバージョンである **ansible-core** パッケージが自動的にインストールされます。**ansible-freeipa** と **ansible-core** の両方が、**rhel-9-for-x86_64-appstream-rpms** リポジトリーで利用できます。

RHEL 8.6 および RHEL 9 の **ansible-freeipa** には、RHEL 8 で含まれていたモジュールがすべて含まれています。

(JIRA:RHELPLAN-100359)

IdM は、automountlocation、automountmap、および automountkeyAnsible モジュールをサポートするようになりました。

この更新では、**ansible-freeipa** パッケージに、**ipaautomountlocation**、**ipaautomountmap**、および **ipaautomountkey** モジュールが含まれています。これらのモジュールを使用して、IdM の場所にある IdM クライアントにログインした IdM ユーザーが自動的にマウントされるようにディレクトリーを設定できます。現在サポートされているのはダイレクトマップのみであることに注意してください。

(JIRA:RHELPLAN-79161)

サブ ID 範囲の管理は、shadow-utils で可能です。

以前では、**shadow-utils** が設定したサブ ID は、**/etc/subuid** ファイルおよび **/etc/subgid** ファイルから自動的に範囲を設定していました。今回の更新で、**subid** フィールドに値を設定することで、**/etc/nsswitch.conf** ファイルで subID 範囲の設定を利用できるようになりました。詳細は **man subuid** および **man subgid** を参照してください。また、今回の更新で、IPA サーバーからのサブ ID 範囲を提供する、**shadow-utils** プラグインの SSSD 実装が利用可能になりました。この機能を使用するには、**subid: sss** を **/etc/nsswitch.conf** に追加します。このソリューションは、コンテナー化した環境でルートレスコンテナーを容易にするために役立ちます。

/etc/nsswitch.conf ファイルが **authselect** ツールで設定されている場合は、**authselect** のドキュメントに記載されている手順に従う必要があります。そうでない場合は、**/etc/nsswitch.conf** を手動で修正できます。

(BZ#1859252)

subID 範囲の管理は、IdM でサポートされています。

この更新により、Identity Management でユーザーの ID サブ範囲を管理できるようになりました。**ipa**CLI ツールまたは IdM Web UI インターフェイスを使用して、自動的に設定されたサブ ID 範囲をユーザーに割り当てることができますが、これはコンテナー化された環境では便利かもしれません。

(BZ#1952028)

Identity Management インストールパッケージがモジュール解除されました

RHEL 8 以前では、IdM パッケージはモジュールとして配布されていたため、ストリームを有効にして、目的のインストールに対応するプロファイルをインストールする必要がありました。IdM インストールパッケージは、RHEL 9 でモジュール解除されているため、次の **dnf** コマンドを使用して IdM サーバーをインストールできます。

統合 DNS サービスがないサーバーの場合は、次のコマンドを実行します。

```
# dnf install ipa-server
```

統合 DNS サービスがあるサーバーの場合は、次のコマンドを実行します。

```
# dnf install ipa-server ipa-server-dns
```

(BZ#2080875)

従来の RHEL ansible-freeipa リポジトリーの代替Ansible Automation Hub

この更新では、標準の RHEL リポジトリーからダウンロードする代わりに、Ansible Automation Hub (AAH) から **ansible-freeipa** モジュールをダウンロードできます。AAH を使用することで、このリポジトリーで利用可能な **ansible-freeipa** モジュールのより高速な更新の恩恵を受けることができます。

AAH では、**ansible-freeipa** のロールとモジュールがコレクション形式で配布されます。AAH ポータルのコンテンツにアクセスするには、Ansible Automation Platform (AAP) サブスクリプションが必要であることに注意してください。また、**ansible** バージョン 2.9 以降も必要です。

redhat.rhel_idm コレクションには、従来の **ansible-freeipa** パッケージと同じコンテンツが含まれています。ただし、コレクション形式では、名前空間とコレクション名で設定される完全修飾コレクション名 (FQCN) が使用されます。たとえば、**redhat.rhel_idm.ipadnsconfig** モジュールは、RHEL リポジトリーによって提供される **ansible-freeipa** の **ipadnsconfig** モジュールに対応します。名前空間とコレクション名の組み合わせにより、オブジェクトが一意になり、競合することなく共有できるようになります。

(JIRA:RHELPLAN-103147)

ansible-freeipa モジュールを IdM クライアントでリモートで実行できるようになりました

以前は、**ansible-freeipa** モジュールは IdM サーバーでのみ実行できました。これには、Ansible 管理者が IdM サーバーへの **SSH** アクセスを持っている必要があり、潜在的なセキュリティーの脅威を引き起こしていました。この更新により、IdM クライアントであるシステム上で **ansible-freeipa** モジュールをリモートで実行できるようになります。その結果、IdM の設定とエンティティーをより安全な方法で管理できます。

IdM クライアントで **ansible-freeipa** モジュールを実行するには、次のいずれかのオプションを選択します。

- Playbook の **hosts** 変数を IdM クライアントホストに設定します。
- **ansible-freeipa** モジュールを使用する Playbook タスクに **ipa_context: client** 行を追加します。

ipa_context 変数を IdM サーバー上の **client** に設定することもできます。ただし、通常、サーバーコンテキストの方がパフォーマンスが向上します。**ipa_context** が設定されていない場合、**ansible-freeipa** はサーバーまたはクライアントで実行されているかどうかを確認し、それに応じてコンテキストを設定します。IdM クライアントホスト上の **server** に **context** が設定された **ansible-freeipa** モジュールを実行すると、**missing libraries** エラーが発生することに注意してください。

(JIRA:RHELPLAN-103146)

ipadnsconfig モジュールには、グローバルフォワーダーを除外するための action: member が必要になりました。

今回の更新で、**ansible-freeipa ipadnsconfig** モジュールを使用して Identity Management (IdM) のグローバルフォワーダーを除外するには、**state: absent** オプションの他に **action: member** オプションを使用する必要があります。Playbook で **action: member** を使用せずに **state: absent** だけを使用すると、その Playbook は失敗します。そのため、すべてのグローバルフォワーダーを削除するには、Playbook でこれらをすべて個別に指定する必要があります。一方、**state: present** オプションに **action: member** は必要ありません。

(BZ#2046325)

AD ユーザー向けの自動プライベートグループが、一元管理された設定をサポート

IdM クライアントの SSSD の互換バージョンで、信頼された Active Directory ドメインのユーザーのプライベートグループを管理する方法を一元的に定義できるようになりました。この改善により、AD ユーザーを処理する ID 範囲に対して、SSSD の **auto_private_groups** オプションの値を明示的に設定できるようになりました。

auto_private_groups オプションが明示的に設定されていない場合は、デフォルト値が使用されます。

- **ipa-ad-trust-posix** ID の範囲では、デフォルト値は **false** です。SSSD は、AD エントリーの **uidNumber** と **gidNumber** を常に使用します。**gidNumber** を持つグループが AD に存在している必要があります。
- **ipa-ad-trust** ID の範囲では、デフォルト値は **真** です。SSSD は、エントリー SID からの **uidNumber** をマッピングします。**gidNumber** は常に同じ値に設定され、プライベートグループは常にマッピングされます。

auto_private_groups を 3 番目の設定 (**ハイブリッド**) に設定することもできます。この設定では、ユーザーエントリーの GID が UID と同じであるにもかかわらず、この GID を持つグループがない場合に、SSSD がプライベートグループをマッピングします。UID と GID が異なる場合は、この GID 番号のグループが存在する必要があります。

この機能は、ユーザープライベートグループ用に別のグループオブジェクトの保持を停止しながら、既存のユーザープライベートグループを保持する管理者に役立ちます。

(BZ#1957736)

BIND のカスタマイズ可能なロギング設定

この改善により、**/etc/named/ipa-logging-ext.conf** 設定ファイルで、Identity Management サーバーの BIND DNS サーバーコンポーネントのロギング設定を設定できるようになりました。

(BZ#1966101)

IdM キータブの取得時の IdM サーバーの自動検出

この改善により、**ipa-getkeytab** コマンドで Kerberos キータブを取得する際に、IdM サーバーのホスト名を指定する必要がなくなりました。サーバーのホスト名を指定しない場合は、DNS 検出が使用されて IdM サーバーが検出されます。サーバーが見つからない場合は、**/etc/ipa/default.conf** 設定ファイルで指定された **host** に戻ります。

(BZ#1988383)

RHEL 9 が Samba 4.15.5 を提供する

RHEL 9 には Samba 4.15.5 が使用されており、バージョン 4.14 に対するバグ修正および機能拡張が提供されます。

- 一貫したユーザーエクスペリエンスのために、Samba ユーティリティーのオプションの名前が変更され、削除されました。
- サーバーのマルチチャンネルサポートがデフォルトで有効になりました。
- **SMB2_22**、**SMB2_24**、および **SMB3_10** のダイアレクトは、Windows のテクニカルプレビューでのみ使用されていましたが、削除されました。

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ないことに留意してください。

Samba を更新したら、**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。

(BZ#2013578)

ログアナライザーツールを使用したクライアント要求の追跡

SSSD(System Security Services Daemon) には、複数の SSSD コンポーネントからのログファイル全体で開始からの要求を追跡するログ解析ツールが追加されました。

ログアナライザーツールを使用すると、SSSD のデバッグログをより簡単に確認でき、SSSD の問題のトラブルシューティングに役立ちます。たとえば、SSSD プロセス全体で特定のクライアント要求のみに関連する SSSD ログを抽出および出力できます。アナライザーツールを実行するには、**sssctl analyze** コマンドを使用します。

(JIRA:RHELPLAN-97899)

SSSD がデフォルトでバックトレースをログするようになりました。

この改善により、SSSD は詳細なデバッグログをメモリー内のバッファーに保存し、障害発生時にログファイルに追加できるようになりました。デフォルトでは、以下のエラーレベルが原因でバックトレースが発生します。

- レベル 0: 致命的な障害
- レベル 1: 重大な障害
- レベル 2: 重大な障害

この動作は、**sssd.conf** 設定ファイルの対応するセクションにある **debug_level** オプションを設定することで、SSSD プロセスごとに変更できます。

- デバッグレベルを 0 に設定すると、レベル 0 のイベントのみがバックトレースをトリガーします。
- デバッグレベルを 1 に設定すると、レベル 0 と 1 でバックトレースが発生します。
- デバッグレベルを 2 以上に設定すると、レベル 0 から 2 のイベントでバックトレースが発生します。

sssd.conf の対応するセクションで **debug_backtrace_enabled** オプションを **false** に設定することで、SSSD プロセスごとにこの機能を無効にできます。

```
[sssd]
debug_backtrace_enabled = true
debug_level=0
...

[nss]
debug_backtrace_enabled = false
...

[domain/idm.example.com]
debug_backtrace_enabled = true
debug_level=2
...


...
```

(BZ#1949149)

SSSD のデフォルトの SSH ハッシュ値が OpenSSH 設定と一致するようになりました

ssh_hash_known_hosts のデフォルト値が false に変更になりました。これは、デフォルトでホスト名をハッシュしない OpenSSH 設定と一致するようになりました。

ただし、引き続きホスト名をハッシュする必要がある場合は、**/etc/sssd/sssd.conf** 設定ファイルの **[ssh]** セクションに **ssh_hash_known_hosts = True** を追加します。

(BZ#2014249)

Directory Server 12.0 は、アップストリームバージョン 2.0.14 をベースとする

Directory Server 12.0 は、アップストリームバージョン 2.0.14 をベースとしており、以前のバージョンに比べて多くのバグ修正と機能拡張が提供されています。主な変更点の一覧については、更新前にアップストリームのリリースノートを参照してください。

- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-14.html
- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-13.html
- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-12.html
- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-11.html
- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-10.html
- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-9.html
- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-8.html
- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-7.html
- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-6.html
- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-5.html
- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-4.html
- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-3.html
- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-2.html
- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-1.html

(BZ#2024693)

Directory Server が、tmpfs ファイルシステムのデータベースのメモリーマッピングされたファイルを保存するようになりました。

Directory Server の **nsslapd-db-home-directory** パラメーターは、データベースのメモリーマッピングファイルの場所を定義します。この改善により、パラメーターのデフォルト値が **/var/lib/dirsrv/slapd-instance_name/db/** から **/dev/shm/** に変更になりました。その結果、**tmpfs** ファイルシステムに保存されている内部データベースがあると、Directory Server のパフォーマンスが向上します。

(BZ#2088414)

FreeRADIUS サポートが再設計されました。

RHEL 9 では、既存の FreeRADIUS オファリングが合理化され、Identity Management (IdM)の戦略的な方向に密接に連携するようになりました。Red Hat は、IdM のお客様に最高のサポートを提供するために、FreeRADIUS を使用してこれらの外部認証モジュールのサポートを強化しています。

- **krb5** および LDAP に基づく認証
- **Python 3** 認証

次のモジュールはサポート対象外になりました。

- MySQL、PostgreSQL、SQlite、および unixODBC データベースコネクター
- **Perl** 言語モジュール
- REST API モジュール

注記

ベースパッケージの一部として提供される PAM 認証モジュールおよびその他の認証モジュールは影響を受けません。

削除されたモジュールの代替は、Fedora プロジェクトなどのコミュニティーでサポートされているパッケージで見つけることができます。

さらに、**freeradius** パッケージのサポート範囲は、次のユースケースに限定されています。

- FreeRADIUS をワイヤレス認証プロバイダーとして使用し、IdM を認証のバックエンドソースとして使用します。認証は、**krb5** および LDAP 認証パッケージを使用して、またはメインの FreeRADIUS パッケージの PAM 認証として行われます。
- FreeRADIUS を使用して、**Python 3** 認証パッケージで IdM の認証用に信頼できる情報源を提供します。

(JIRA:RHELDOCS-17553)

4.16. デスクトップ

GNOME がバージョン 40 に更新されました。

GNOME 環境は、GNOME 3.28 から GNOME 40 に更新され、多くの新機能が追加されました。

GNOME 40 には、新しく改良された**Activities Overview**のデザインが含まれています。これにより、概要がよりまとまりのあるものとなり、システムのナビゲーションやアプリケーションの起動などの操作性が向上しました。ワークスペースは水平に配置され、ウィンドウの概要とアプリケーショングリッドには垂直にアクセスできるようになりました。

その他の GNOME の改良点は以下の通りです。

- GNOME のパフォーマンスとリソースの使い方が大幅に改善されました。
- ユーザーインターフェイス、アイコン、デスクトップなどのビジュアルスタイルが一新されました。
- GNOME アプリケーションでは、トップパネルから利用可能だったアプリケーションメニューを使用しなくなりました。この機能は、アプリケーションウィンドウ内の主要メニューに配置されています。
- **Settings**アプリケーションのデザインが変更されました。
- 画面共有やリモートデスクトップセッションが改善されました。
- 独自の NVIDIA ドライバーを使用している場合は、ディスクリート GPU を使用したアプリケーションを起動できるようになりました。
  a. 概要を開きます。
  b. ダッシュ内のアプリケーションアイコンを右クリックします。
  c. メニューから**Launch on Discrete GPU**項目を選択します。
- **Power Off / Log Out**メニューには、**Suspend**オプションと、**Alt**キーを押した場合にシステムをブートローダーメニューに再起動できる**Restart**オプションが新たに加わりました。

- Flatpak アプリケーションが自動的に更新されるようになりました。
- 概要に表示されるアプリケーションのアイコンを、ドラッグアンドドロップでフォルダーにまとめることができるようになりました。
- **Terminal**アプリケーションでは、右から左への文字入力や双方向の文字入力が可能になりました。
- **Pointer Location**アクセシビリティ機能が Wayland セッションで動作するようになりました。この機能が有効な場合、**Ctrl**を押すと画面上のポインターの位置がハイライトされます。
- GNOME シェルの拡張機能は、**Software**ではなく、**Extensions**アプリケーションで管理されるようになりました。**Extensions**アプリケーションは、エクステンションの更新、拡張機能の更新、拡張機能設定の設定、拡張機能の削除や無効化を行います。
- 通知のポップオーバーに**Do Not Disturb**ボタンが追加されました。ボタンを有効にすると、画面に通知が表示されなくなります。
- パスワードを必要とするシステムダイアログで、目 (▮) のアイコンをクリックしてパスワードテキストを表示するオプションが追加されました。
- **Software**アプリケーションは、モバイルデータネットワークなどの従量制ネットワークを自動的に検出するようになりました。現在のネットワークが従量制の場合、**Software**はデータ使用量を削減するために更新を一時停止します。
- 接続されたディスプレイごとに、Wayland セッションで異なるリフレッシュレートを使用できるようになりました。
- フラクショナルディスプレイスケーリングは、実験的なオプションとして用意されています。あらかじめ設定されたいくつかの分数比が含まれています。
  実験的なフラクショナルスケーリングを有効にするには、有効な実験的機能のリストに **scale-monitor-framebuffer** の値を追加します。

  ```
  $ dconf write \
        /org/gnome/mutter/experimental-features \
        "['scale-monitor-framebuffer']"
  ```

  その結果、**Settings**の**Display**パネルで、フラクショナルスケーリングオプションにアクセスできるようになります。

GNOME の変更点の詳細については、リリースノート のバージョン 3.30 から 40.0 を参照してください。

(JIRA:RHELPLAN-101240)

PipeWire がデフォルトのオーディオサービスに

Pipewireサービスは、すべてのオーディオ出力と入力を管理するようになりました。**Pipewire**は、一般的な使用例では**PulseAudio**サービスを、専門的な使用例では**JACK**サービスを置き換えます。システムは、**PulseAudio**、**JACK**、または**ALSA**フレームワークを使用するアプリケーションからのオーディオを**Pipewire**にリダイレクトするようになりました。

従来のソリューションに対する**Pipewire**のメリットは以下のとおりです。

- コンシューマーとプロフェッショナルユーザーのための統一されたソリューション
- フレキシブルなモジュール式アーキテクチャー

- JACKサービスと同様の高いパフォーマンスと低いレイテンシー
- オーディオクライアント間の隔離によるセキュリティーの向上

JACK サービスを使用するアプリケーションのためにJACKサービスを設定する必要はありません。すべてのJACKアプリケーションはデフォルトの RHEL 設定で動作するようになりました。

RHEL で **pulseaudio** は依然として利用でき、**PipeWire** の代わりに有効にすることができます。詳細はPipeWire から PulseAudio への切り替え wo 参照してください。

(JIRA:RHELPLAN-101241)

電源プロファイルが GNOME で利用可能に

GNOME 環境のSettingsのPowerパネルで、複数の電源プロファイルを切り替えられるようになりました。電源プロファイルは、選択した目標に対してシステムの各種設定を最適化します。

利用できる電源プロファイルは以下のとおりです。

パフォーマンス
: 高いシステムパフォーマンスに最適化され、バッテリー寿命が短くなります。このプロファイルは、特定のシステム設定でのみ利用可能です。

Balanced
: 標準的なシステム性能と消費電力を提供します。これはデフォルトのプロファイルです。

Power Saver
: バッテリー駆動時間が長くなり、システムのパフォーマンスが低下します。このプロファイルは、バッテリー残量が少なくなると自動的に起動します。

電源プロファイルの設定は、システムが再起動しても保持されます。

電源プロファイルの機能は、デフォルトでインストールされている **power-profiles-daemon** パッケージから利用できます。

(JIRA:RHELPLAN-101242)

langpacks が、言語サポートを提供するようになりました。

様々な言語をサポートするために、**langpacks** パッケージが用意されました。インストールする言語サポートのレベルをカスタマイズするには、次のようなパッケージ名を使用します。ここで、**code** は言語の短い ISO コードで、例えばスペイン語は **es** となります。

langpacks-core-code
: 以下のような基本的な言語サポートを提供します。

  - **glibc** のロケール
  - デフォルトのフォント
  - 言語で要求されている場合は、デフォルトの入力方法

langpacks-core-font-code
: その言語のデフォルトフォントのみを提供します。

langpacks-code
: 基本的な言語サポートに加えて、以下を含む完全な言語サポートを提供します。

- 翻訳
- スペルチェッカーの辞書
- 追加フォント

(JIRA:RHELPLAN-101247)

軽量で単一アプリケーションの環境

1 つのアプリケーションのみを表示するグラフィカルユースケースでは、軽量のユーザーインターフェイス (UI) が利用できるようになりました。

GNOME は、単一アプリケーションセッション (kiosk モード とも呼ばれる) で起動できます。このセッションでは、GNOME は、設定したアプリケーションのフルスクリーンウィンドウのみを表示します。

単一アプリケーションセッションのリソース集中度は、標準の GNOME セッションよりも大幅に低くなります。

詳細は、Restricting the session to a single application を参照してください。

(JIRA:RHELPLAN-102552)

ログイン時およびデスクトップセッション時のセキュリティー分類バナー

分類バナーを設定して、システムの全体的なセキュリティー分類レベルを示すことができるようになりました。これは、ログインしているシステムのセキュリティー分類レベルをユーザーが認識している必要があるデプロイメントに役立ちます。

分類バナーは、設定に応じて、次のコンテキストで表示されます。

- 実行中のセッション内
- ロック画面
- ログイン画面

分類バナーは、無視できる通知または永続的なバナーのいずれかの形式をとることができます。

詳細については、Displaying the system security classification を参照してください。

(BZ#2031186)

デフォルトのウォールパパーによって Red Hat ロゴが追加されます。

デフォルトの RHEL ウォールパパーに、Red Hat ロゴが表示されるようになりました。ロゴは画面の左上隅にあります。

ロゴを無効にするには、**Background Logo** GNOME Shell 拡張機能を無効にします。

(BZ#2057150)

Firefox が PKCS#12 ファイルで強力な暗号化を使用するようになりました。

Firefox Web ブラウザーは、PKCS#12 ファイルを使用してクライアント認証証明書を確立します。以前は、Firefox は、レガシーアルゴリズムを使用してこれらのファイルを暗号化していました。

- PKCS#12 ファイルの証明書を暗号化する PBE-SHA1-RC2-40

- PKCS#12 ファイルの鍵を暗号化する PBE-SHA1-3DES

今回のリリースにより、Firefox はデフォルトで強力なアルゴリズムを使用してファイルを暗号化します。

- PKCS#12 ファイルの証明書を暗号化する PBKDF2 を使用した AES-256-CBC
- PKCS#12 ファイルの鍵を暗号化する PBKDF2 を使用した AES-128-CBC

この変更により、PKCS#12 ファイルは FIPS (Federal Information Processing Standard) と互換性があります。

レガシー暗号化アルゴリズムは、Firefox でデフォルト以外のオプションとして引き続きサポートされます。

(BZ#1764205)

4.17. グラフィックインフラストラクチャー

Wayland セッションが NVIDIA ドライバーのデフォルトになりました。

NVIDIA ドライバーを使用する場合は、ドライバー設定が Wayland に対応していると、デスクトップセッションはデフォルトで Wayland ディスプレイプロトコルを選択するようになりました。以前の RHEL リリースでは、NVIDIA ドライバーが常に Wayland を無効にしていました。

お使いのシステムで NVIDIA ドライバーを使用して Wayland 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。

- **nvidia-drm.modeset=1**
- **NVreg_PreserveVideoMemoryAllocations=1**

RHEL 8.0 以降、Wayland は、その他のグラフィックドライバーでデフォルトのディスプレイプロトコルでした。

現在、NVIDIA ドライバーを使用した Wayland セッションは引き続き完了せず、特定の既知の問題を表示します。Red Hat は NVIDIA をアクティブにし、GPU スタック全体でこのギャップや問題に対処しています。

NVIDIA ドライバーでの Wayland の制限のいくつかについては、**既知の問題** のセクションを参照してください。

(JIRA:RHELPLAN-119000)

4.18. WEB コンソール

sudo および Web コンソールからの SSH 用のスマートカード認証

これまでは、スマートカード認証を使用して sudo 権限を取得したり、Web コンソールで SSH を使用したりすることはできませんでした。この更新により、Identity Management のユーザーがスマートカードを使用して sudo 権限を取得したり、SSH で別のホストに接続したりできるようになります。

注記

1 枚のスマートカードを使用して、認証を行い sudo 権限を得ることしかできません。sudo に別のスマートカードを使用することはサポートされていません。

(JIRA:RHELPLAN-95126)

Web コンソールで再起動せずにカーネルセキュリティーパッチ

この Web コンソール更新により、ユーザーは **kpatch** フレームワークを使用して再起動を強制せずに、カーネルセキュリティーパッチを適用できます。管理者は、今後使用するカーネルをライブパッチストリームに自動的にサブスクライブすることもできます。

(JIRA:RHELPLAN-95056)

RHEL Web コンソールがデフォルトで Insights 登録を提供

この更新では、Red Hat Enterprise Linux Web コンソールを使用して RHEL システムを登録するときに、**このシステムを Red Hat Insights に接続します。**チェックボックスはデフォルトでチェックされています。Insights サービスに接続しない場合は、チェックボックスの選択を解除します。

(BZ#2049441)

コックピットで、既存の TLS 証明書の使用をサポートするようになりました

この機能拡張により、証明書には厳密なファイルパーミッション要件 (**root:cockpit-ws 0640** など) がなくなり、他のサービスと共有できるようになりました。

(JIRA:RHELPLAN-103855)

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

Networking システムロールが、SAE に対応するようになりました。

Wi-Fi Protected Access バージョン 3 (WPA3) ネットワークでは、SAE (authentication of equals) 方法により、暗号鍵が送信されないようになります。この機能強化により、Networking RHEL システムロールは SAE に対応します。これにより、管理者は Networking RHEL System Role を使用して、WPA-SAE を使用する Wi-Fi ネットワークへの接続を設定できるようになりました。

(BZ#1993304)

Networking システムロールが owe をサポートするようになりました。

Networking RHEL System Role は、Opportunistic Wireless Encryption (owe) をサポートするようになりました。**owe** は、Wi-Fi クライアントとアクセスポイント間の暗号化を使用し、Wi-Fi クライアントを盗聴攻撃から保護するワイヤレス認証キー管理タイプです。owe を使用するには、ワイヤレス認証キー管理タイプの **key_mgmt** フィールドを **owe** に設定します。

(BZ#1993377)

Firewall システムロールが、ファイアウォールデフォルトゾーンの設定に対応

ゾーンは、着信トラフィックをより透過的に管理する概念を表しています。ゾーンはネットワークインターフェイスに接続されているか、ソースアドレスの範囲に割り当てられます。各ゾーンのファイアウォールルールは個別に管理されるため、管理者は複雑なファイアウォール設定を定義してトラフィックに適用できます。この機能を使用すると、**firewall-cmd --set-default-zone zone-name** と同じように、インターフェイスを割り当てるデフォルトゾーンとして使用されるデフォルトゾーンを設定できます。

(BZ#2022461)

Storage RHEL システムロールが LVM VDO ボリュームに対応

この機能強化により、Storage システムロールを使用して、論理マネージャーボリューム (LVM) 仮想

データオプティマイザー (VDO) ボリュームを管理できるようになりました。LVM ファイルシステムは VDO ボリュームを管理し、この機能を使用して、LVM ボリュームを圧縮して重複排除できるようになりました。その結果、VDO はストレージボリュームの使用を最適化するのに役立ちます。

(BZ#1978488)

パーセンテージで表現されたボリュームサイズのサポートを、Storage システムロールで利用可能

この改善により、Storage RHEL システムロールに、LVM ボリュームのサイズをプールの合計サイズのパーセンテージで表現するサポートが追加されました。LVM ボリュームのサイズをプール/VG サイズの割合として指定できます。以下に例を示します。ファイルシステムの人間が読めるサイズに加えて 50%、たとえば 10g、50GiB。

(BZ#1984583)

Storage システムロールでキャッシュされたボリュームのサポートを使用できます

この機能拡張により、キャッシュされた LVM 論理ボリュームを作成および管理するためのストレージ RHEL システムロールのサポートが追加されます。LVM キャッシュを使用すると、SSD などの小規模なデバイスに論理ボリュームのサブセットを一時的に格納することにより、低速な論理ボリュームのパフォーマンスを向上させることができます。

(BZ#2016517)

ファイアウォールロールにソースを追加または削除する機能

この更新により、**source** パラメーターを使用してファイアウォール設定でソースを追加または削除できます。

(BZ#2021667)

Microsoft SQL Server Management 用の新しい Ansible ロール

新しい **microsoft.sql.server** ロールは、IT およびデータベース管理者が、Red Hat Enterprise Linux で SQL Server の設定、設定、およびパフォーマンスチューニングに関連するプロセスを自動化するのに役立ちます。

(BZ#2013853)

Microsoft SQL システムのロールは、切断されたサブスクリプションまたは Satellite サブスクリプション用にカスタマイズされたリポジトリーをサポートするようになりました

以前は、カスタムサーバーからパッケージをプルする必要がある切断された環境のユーザー、または Satellite または Capsule を指す必要がある Satellite ユーザーは、**microsoft.sql.server** ロールからのサポートがありませんでした。この更新プログラムは、パッケージをダウンロードするリポジトリーをカスタマイズするために使用できる **mssql_rpm_key**、**mssql_server_repository**、および **mssql_client_repository** 変数を提供することで修正されています。URL が指定されていない場合、**mssql** ロールは公式の Microsoft サーバーを使用して RPM をダウンロードします。

(BZ#2064648)

MSSQL ロールは、マネージド設定ファイルで一貫して Ansible_managed コメントを使用します

MSSQL ロールは、**/var/opt/mssql/mssql.conf** 設定ファイルを生成します。この更新により、MSSQL ロールは、Ansible 標準の **ansible_managed** 変数を使用して、設定ファイルに Ansiblemanaged コメントを挿入します。コメントは、MSSQL ロールによってファイルが上書きされるため、設定ファイル

を直接編集してはならないことを示します。その結果、設定ファイルには、設定ファイルが Ansible によって管理されていることを示す宣言が含まれています。

(BZ#2064690)

RHEL System Roles の Ansible Core サポート

RHEL 9 GA リリース時点で、RHEL がサポートする自動化のユースケースを可能にするために、Ansible Core はサポート範囲が制限されています。Ansible Core は、別のリポジトリーで以前のバージョンの RHEL で提供されていた Ansible Engine に代わるものです。Ansible Core は、RHEL の AppStream リポジトリーで利用できます。サポートされているユースケースの詳細については、RHEL 9 および RHEL 8.6 以降の AppStream リポジトリーに含まれる Ansible Core パッケージのサポート対象範囲 を参照してください。

Ansible Engine のサポートが必要な場合、または RHEL 以外の自動化のユースケースのサポートが必要な場合は、Red Hat Support で Red Hat を作成してください。

(JIRA:RHELPLAN-103540)

1 つの elasticsearch 出力ディクショナリーで複数の elasticsearch ホストの設定に対応

以前は、**server_host** パラメーターは、1 台のホストで文字列を取得するのに使用されていました。この改善により、基本となる **rsyslog omelasticsearch** の仕様に合わせて調整されたため、複数のホストに対応するために、文字列のリストが取得されるようになりました。その結果、基本的な **rsyslog omelasticsearch** の指定に従って、ホストに調整されます。その結果、1 つの **elasticsearch** 出力ディクショナリーで複数の **elasticsearch** ホストを設定できます。

(BZ#1986460)

RHEL システムロールが VPN 管理に対応

以前のリリースでは、Linux で安全で適切な IPsec トンネリングおよび仮想プライベートネットワーク (VPN) ソリューションを設定するのが困難でした。今回の機能拡張により、VPN RHEL システムロールを使用して、多数のホストにおいて、ホスト間およびメッシュ接続の VPN トンネルを簡単に設定できるようになりました。これにより、RHEL システムロールプロジェクト内で、VPN および IPsec トンネリング設定用の一貫した安定した設定インターフェイスが得られます。

(BZ#2019341)

SSHD RHEL システムロールが、非排他的設定スニペットに対応するようになりました。

この機能を使用すると、名前空間を使用して以前の設定を書き換えることなく、さまざまなロールや Playbook で SSHD を設定できます。名前空間はドロップインディレクトリーと似ており、SSHD 用に非排他設定スニペットを定義します。その結果、設定ファイル全体ではなく、設定のごく一部のみを設定する必要がある場合は、別のロールの SSHD RHEL システムロールを使用できます。

(BZ#1978752)

timesync RHEL システムロールに追加された NTS (Network Time Security) オプション

クライアントサーバーで **NTS** を有効にするために、Timesync RHEL システムロールに **NTS** オプションが追加されました。NTS は、Network Time Protocol (NTP) に指定されている新しいセキュリティーメカニズムです。NTS は、クライアント固有の設定がなくても NTP クライアントの同期をセキュアにでき、大量のクライアントにスケーリングできます。**NTS** オプションは、バージョン 4.0 以降の **chrony** NTP プロバイダーでのみ対応しています。

(BZ#1978753)

HA クラスターの RHEL システムロールのサポート

高可用性クラスター (HA クラスター) のロールが完全にサポートされるようになりました。次の注目すべき設定が利用可能です。

- フェンスデバイス、リソース、リソースグループ、およびリソースクローン (メタ属性およびリソース操作を含む) の設定
- リソースの場所の制約、リソースのコロケーションの制約、リソースの順序の制約、およびリソースチケットの制約の設定
- クラスタープロパティーの設定
- クラスターノード、カスタムクラスター名およびノード名の設定
- マルチリンククラスターの設定
- システムの起動時にクラスターが自動的に起動するかどうかの設定

ロールを実行すると、ロールでサポートされていない設定、またはロールの実行時に指定されていない設定が削除されます。

現在、HA Cluster システムロールは SBD をサポートしていません。

(BZ#2054401)

Elasticsearch への Rsyslog ユーザー名およびパスワード認証のサポート

今回の更新により、Elasticsearch のユーザー名とパスワードパラメーターが Logging システムロールに追加されました。その結果、Rsyslog がユーザー名とパスワードを使用して Elasticsearch に対して認証できるようにします。

(BZ#1990490)

NBDE クライアントシステムのロールは静的 IP アドレスをサポートします

以前のバージョンの RHEL では、静的 IP アドレスを使用してシステムを再起動し、ネットワークバウンドディスク暗号化 (NBDE) クライアントシステムロールを使用して設定すると、システムの IP アドレスが変更されていました。この変更により、NBDE クライアントシステムロールにより静的 IP アドレスを持つシステムがサポートされ、再起動後にその IP アドレスは変更されません。

デフォルトでは、NBDE ロールは起動時に DHCP を使用し、システムの起動時に設定済みの静的 IP に切り替えることに注意してください。

(BZ#2031555)

LVM に raid_level を指定するサポートが追加されました。

RHEL 9.0 は、**lvmraid** 機能を使用して、論理ボリューム管理 (LVM) ボリュームの RAID へのグループ化に対応しています。

(BZ#2016518)

Certificate ロールは、フックスクリプトで一貫して "Ansible_managed" コメントを使用します

この機能拡張により、Certificate ロールは、プロバイダーをサポートするためのプレスクリプトとポストスクリプトを生成します。ロールはこれに、Ansible 標準の "ansible_managed" 変数を使用して "Ansible managed" のコメントを挿入します。

- **/etc/certmonger/pre-scripts/script_name.sh**

- **/etc/certmonger/post-scripts/script_name.sh**

コメントは、Certificate ロールがファイルを上書きする可能性があるため、スクリプトファイルを直接編集してはならないことを示しています。その結果、設定ファイルには、設定ファイルが Ansible によって管理されていることを示す宣言が含まれています。

(BZ#2054364)

新しいオプション auto_gateway は、デフォルトルートの動作を制御します

以前は、**DEFROUTE** パラメーターは設定ファイルで設定できませんでしたが、すべてのルートに名前を付けることによって手動で設定することしかできませんでした。この更新により、接続の **ip** 設定セクションに新しい **auto_gateway** オプションが追加されます。これを使用して、デフォルトルートの動作を制御できます。**auto_gateway** は、次の方法で設定できます。

- **true** に設定すると、デフォルトゲートウェイ設定がデフォルトルートに適用されます。
- **false** に設定すると、デフォルトルートが削除されます。
- 指定しない場合、**network** ロールは選択した **network_provider** のデフォルトの動作を使用します。

(BZ#1978773)

network System Role に追加されたすべてのボンディングオプションのサポート

この更新は、**network** RHEL System Role へのすべてのボンディングオプションをサポートします。その結果、ボンディングされたインターフェイスを介したネットワーク伝送を柔軟に制御できます。その結果、そのインターフェイスにいくつかのオプションを指定することにより、ボンディングされたインターフェイスを介したネットワーク伝送を制御できます。

(BZ#2054435)

NetworkManager は、PCI アドレスを使用したネットワークカードの指定をサポートしています

以前は、接続プロファイルの設定中に、NetworkManager は名前または MAC アドレスのいずれかを使用してネットワークカードを指定することしか許可されていませんでした。この場合、デバイス名は安定しておらず、MAC アドレスには、使用された MAC アドレスの記録を維持するためのインベントリーが必要です。これで、接続プロファイルの PCI アドレスに基づいてネットワークカードを指定できます。

(BZ#1999162)

Network システムロールが Ansible の設定ファイルを直接管理

今回の機能拡張により、**network** ロールにより、**/etc/sysconfig/network-scripts** に **ifcfg** ファイルが生成されるようになりました。次に、標準の **ansible_managed** 変数を使用してコメント Ansible managed を挿入します。このコメントは、**network** ロールによって上書きされる可能性があるため、**ifcfg** ファイルを直接編集できないことを示します。**ifcfg** ファイルを処理して Ansible managed コメントを追加する際の重要な相違点は、**network** ロールでは **initscripts** パッケージを使用し、NetworkManager は **nm** パッケージを使用するためです。

(BZ#2057657)

RHEL システムロールの Ansible コアサポート

RHEL 9.0 では、RHEL 対応の自動化ユースケースを有効にするために、Ansible Core はサポート範囲が制限されています。Ansible Core は、以前は別のリポジトリーで提供されていた Ansible Engine を置

き換えます。Ansible Core は、RHEL の AppStream リポジトリーで利用できます。サポートされているユースケースの詳細については RHEL 9 および RHEL 8.6 移行の AppStream リポジトリーに含まれている Ansible Core パッケージのサポート範囲 を参照してください。ユーザーは、システムを Ansible Engine から Ansible Core に手動で移行する必要があります。

(BZ#2012298)

Cockpit システムロールのサポート

この機能拡張により、システムに Web コンソールをインストールして設定できます。そのため、Web コンソールを自動化された方法で管理できます。

(BZ#2021028)

Terminal Session Recoring システムロールは、マネージド設定ファイルで "Ansible managed" コメントを使用します

Terminal Session Recording ロールは、以下の 2 つの設定ファイルを生成します。

- **/etc/sssd/conf.d/sssd-session-recording.conf**
- **/etc/tlog/tlog-rec-session.conf**

今回の更新により、ターミナルセッションの録画ロールで、標準の Ansible 変数 **ansible_managed** を使用して、Ansible managed コメントが設定ファイルに挿入されるようになりました。コメントは、Terminal Session Recorging ロールがファイルを上書きする可能性があるため、設定ファイルを直接編集してはならないことを示しています。その結果、設定ファイルには、設定ファイルが Ansible によって管理されていることを示す宣言が含まれています。

(BZ#2054367)

VPN ロールは、管理された設定ファイルで Ansible_managed コメントを一貫して使用します。

VPN ロールは、次の設定ファイルを生成します。

- **/etc/ipsec.d/mesh.conf**
- **/etc/ipsec.d/policies/clear**
- **/etc/ipsec.d/policies/private**
- **/etc/ipsec.d/policies/private-or-clear**

この更新により、VPN ロールは、Ansible 標準の **ansible_managed** 変数を使用して、設定ファイルに Ansiblemanaged コメントを挿入します。コメントは、VPN ロールがファイルを上書きする可能性があるため、設定ファイルを直接編集してはならないことを示しています。その結果、設定ファイルには、設定ファイルが Ansible によって管理されていることを示す宣言が含まれています。

(BZ#2054369)

Postfix ロールは、管理設定ファイル内で常に Ansible_managed コメントを使用します。

Postfix ロールは **/etc/postfix/main.cf** 設定ファイルを生成します。今回の更新で、Postfix ロールは、Ansible 標準の **ansible_managed** 変数を使用して、設定ファイルに Ansible managed コメントを挿入します。コメントは、Postfixrole がファイルを上書きする可能性があるため、設定ファイルを直接編集しないことを示しています。その結果、設定ファイルには、設定ファイルが Ansible によって管理されていることを示す宣言が含まれています。

(BZ#2057662)

Firewall RHEL システムロールが RHEL 9 に追加されました。

今回の機能強化により、RHEL システムロール **rhel-system-roles.firewall** が **rhel-system-roles** パッケージに追加されました。その結果、管理者はマネージドノードのファイアウォール設定を自動化できます。

(BZ#2021665)

SSH クライアントの RHEL システムロールが、OpenSSH 8.7 の新しい設定オプションに対応するようになりました。

今回の機能強化により、OpenSSH が最新バージョンに更新され、新規ホストを設定するための SSH クライアントロールで利用可能な新しい設定オプションが提供されます。

(BZ#2029427)

4.20. 仮想化

RHEL Web コンソールの新しい仮想化機能

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

- 仮想マシンの名前変更
- クラウドイメージ認証を使用した仮想マシンの作成
- USB デバイスおよび PCI デバイスを仮想マシンに追加して削除
- ネットワークインターフェイスモデルの指定
- ホストと仮想マシンとの間で共有と共有解除

(JIRA:RHELPLAN-102009)

QEMU が Clang を使用する

QEMU エミュレーターが、Clang コンパイラーを使用して構築されるようになりました。これにより、RHEL 9 KVM ハイパーバイザーで、多くの高度なセキュリティー機能およびデバッグ機能を使用できるようになり、今後の機能開発がより効率的になります。

(BZ#1940132)

仮想マシン用の SafeStack

AMD64 および Intel 64 ハードウェア (x86_64) 上の RHEL 9 では、QEMU エミュレーターが、高度なコンパイラーベースのスタック保護機能である SafeStack を使用できるようになりました。SafeStack は、スタックベースのバッファーオーバーフローを悪用してスタック内のリターンポインターを変更し、Return-Oriented Programming (ROP) 攻撃を行う攻撃者の能力を低減します。その結果、RHEL 9 でホストされている仮想マシンは、ROP ベースの脆弱性に対する安全性が大幅に向上しています。

(BZ#1939509)

Intel 64、AMD64、および IBM Z での virtiofs の完全サポート

virtio ファイルシステム (**virtiofs**) は、Intel 64、AMD64、および IBM Z アーキテクチャーで完全にサポートされるようになりました。**virtiofs** を使用すると、ホストシステムとその仮想マシン間でファイルを効率的に共有できます。

(JIRA:RHELPLAN-64576)

KVM ゲストでサポートされる AMD EPYC 7003 シリーズプロセッサー

AMD EPYC 7003 シリーズプロセッサー (**AMD Milan** とも呼ばれます) のサポートが、KVM ハイパーバイザーとカーネルコード、および libvirtAPI に追加されました。これにより、KVM 仮想マシンが AMD EPYC 7003 シリーズプロセッサーを使用できるようになります。

(JIRA:RHELPLAN-65223)

qemu-kvm が追加のマシンタイプに対応しました。

仮想マシン (VM) で使用するために、RHEL 9 に基づく一連の新しいマシンタイプが追加されました。ホストで現在サポートされているすべてのマシンタイプを取得するには、**/usr/libexec/qemu-kvm -M help** コマンドを使用します。

さらに、RHEL 7.5.0 以前に基づくすべてのマシンタイプがサポートされなくなりました。これらには、**pc-i440fx-rhel7.5.0** 以前のマシンタイプも含まれます。これらは、RHEL の以前のメジャーバージョンでデフォルトでした。その結果、RHEL 9 でそのようなマシンタイプの VM を起動しようとすると、**unsupported configuration** エラーで失敗します。ホストを RHEL 9 にアップグレードした後にこの問題が発生した場合は、Red Hat KnowledgeBase を参照してください。

(JIRA:RHELPLAN-75866)

仲介されたデバイスが、IBM Z の仮想化 CLI で対応するようになりました。

virt-install または **virt-xml** を使用して、仲介されたデバイス (vfio-ap や vfio-ccw など) を VM に接続できるようになりました。たとえば、これにより、IBM Z ホストで DASD ストレージデバイスおよび暗号化コプロセッサーをより柔軟に管理できます。また、**virt-install** を使用して、既存の DASD 仲介デバイスをプライマリーディスクとして使用する仮想マシンを作成できます。手順は、RHEL 9 での仮想化の設定および管理ガイドを参照してください。

(BZ#1995131)

モジュラーの libvirt デーモン

RHEL 9 では、**libvirt** ライブラリーは、ホスト上の個々の仮想化ドライバーセットを処理するモジュラーデーモンを使用します。たとえば、**virtqemud** デーモンは QEMU ドライバーを処理します。これにより、リソース負荷の最適化や監視など、仮想化ドライバーに関連するさまざまなタスクをきめ細かくすることができます。

さらに、モノリシック libvirt デーモン **libvirtd** は非推奨になりました。ただし、RHEL 8 から RHEL 9 にアップグレードした場合でも、ホストは **libvirtd** を使用します。これは、RHEL 9 でも引き続き使用できます。ただし、Red Hat は、代わりにモジュラー **libvirt** デーモンに切り替えることを推奨します。

(JIRA:RHELPLAN-113994)

Windows 11 および Windows Server 2022 ゲストがサポートされるようになりました

RHEL 9 は、KVM 仮想マシンのゲストオペレーティングシステムとして Windows 11 および Windows Server 2022 の使用に対応します。

(BZ#2036856, BZ#2004161)

ksmtuned が qemu-kvmとは別に配布されるようになりました。

KVM ハイパーバイザーのフットプリントを減らすために、**ksmtuned** ユーティリティーは **qemu-kvm** に依存しなくなりました。したがって、kernel same-page merging(KSM) を設定する必要がある場合は、**ksmtuned** パッケージを手動でインストールする必要があります。

(BZ#2069501, BZ#1971678, BZ#1972158)

新機能: vTPM

RHEL 9 では、Virtual Trusted Platform Module (vTPM) に完全に対応しています。vTPM を使用して、RHEL 9 KVM ハイパーバイザーで実行している仮想マシンに TPM 仮想暗号プロセッサーを追加できます。これにより、仮想マシンを使用して暗号鍵を生成、保存、および管理できます。

(JIRA:RHELPLAN-98617)

Intel Atom P59 シリーズプロセッサーの仮想化サポート

今回の更新で、RHEL 9 の仮想化で、以前の Snow Ridge として知られる Intel Atom P59 シリーズプロセッサーのサポートが追加されました。その結果、RHEL 9 でホストされる仮想マシンは、**Snowridge** CPU モデルを使用し、プロセッサーが提供する新機能を活用できるようになりました。

(BZ#1874187)

4.21. クラウド環境の RHEL

RHEL 9 が WALinuxAgent 2.3.0.2 を提供

RHEL 9 は、Windows Azure Linux Agent (**WALinuxAgent**) パッケージバージョン 2.3.0.2 とともに配布されます。バージョン 2.2.49 への主なバグ修正および機能強化は、以下のとおりです。

- RequiredFeatures および GoalStateAggregateStatus API のサポートが追加されました。
- 拡張マニフェストのフォールバック先が追加されました。
- 例外を作成する際に str.format() の呼び出しが欠落していたのを修正しました。

(BZ#1972101)

Azure の RHEL が MANA に対応するようになりました。

Microsoft Azure で実行している RHEL 9 仮想マシンが、Microsoft Azure ネットワークアダプター (MANA) を使用できるようになりました。

(BZ#1957818)

cloud-init が VMware GuestInfo データソースに対応

今回の更新で、**cloud-init** ユーティリティーが VMware guestinfo データのデータソースを読み取ることができるようになりました。その結果、**cloud-init** を使用 s した VMware vSphere に RHEL 9 仮想マシンのセットアップが、より効率的で信頼性が高くなりました。

(BZ#2040090)

RHEL 9 仮想マシンが Azure の特定の ARM64 ホストでサポートされるようになりました。

ゲストオペレーティングシステムとして RHEL9 を使用する仮想マシンは、Ampere Altra ARM ベースのプロセッサーで実行されている Microsoft Azure ハイパーバイザーでサポートされるようになりました。

(BZ#1949613)

cloud-init は、MicrosoftAzure 上のユーザーデータをサポートする

--user-data オプションが **cloud-init** ユーティリティーに導入されました。このオプションを使用すると、Azure で RHEL 9 仮想マシンをセットアップするときに、 Azure Instance Metadata Service (IMDS) からスクリプトとメタデータを渡すことができます。

(BZ#2042351)

cloud-init の新しい SSH モジュール

今回の更新で、**cloud-init** ユーティリティーに SSH モジュールが追加され、インスタンスの作成時にホストキーが自動的に生成されるようになりました。

この変更により、デフォルトの **cloud-init** 設定が更新されました。したがって、ローカルの変更があった場合は、/etc/cloud/cloud.cfg に "ssh_genkeytypes: ['rsa', 'ecdsa', 'ed25519']" 行が含まれていることを確認してください。

そうしないと、**sshd** サービスを起動できないイメージが **cloud-init** によって作成されます。この問題が発生した場合は、次の手順に従って問題を回避してください。

1. **/etc/cloud/cloud.cfg** ファイルに次の行が含まれていることを確認します。

```
ssh_genkeytypes:  ['rsa', 'ecdsa', 'ed25519']
```

2. **/etc/ssh/ssh_host_*** ファイルがインスタンスに存在するかどうかを確認します。
3. **/etc/ssh/ssh_host_*** ファイルが存在しない場合は、次のコマンドを使用してホストキーを生成します。

```
cloud-init single --name cc_ssh
```

4. sshd サービスを再起動します。

```
systemctl restart sshd
```

(BZ#2115791)

4.22. サポート性

sos report は推定モードの実行を提供するようになりました

この **sos report** の更新により、RHEL サーバーから **sos** レポートを収集するために必要なディスク容量を概算できる **--estimate-only** オプションが追加されます。**sos report --estimate-only** コマンドの実行:

- **sos report** のドライランを実行します
- すべてのプラグインを連続して模倣し、それらのディスクサイズを推定します。

最終的なディスクスペースの見積もりは非常に概算であることに注意してください。したがって、推定値を 2 倍にすることを推奨します。

(BZ#2011537)

4.23. コンテナー

Podman が、セキュアな短縮名に対応

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

- Enforcing:イメージのプル中に一致するエイリアスが見つからない場合、Podman はユーザーが非修飾レジストリーのいずれかを選択するよう求めます。選択したイメージを正常に取得すると、Podman は、**$HOME/.cache/containers/short-name-aliases.conf** ファイル (ルートレスユーザー) および **/var/cache/containers/short-name-aliases.conf** (root ユーザー) に新しい短縮名のエイリアスを自動的に記録します。ユーザーを要求できない場合 (stdin や stdout など) が TTY ではない場合は、Podman は失敗します。**short-name-aliases.conf** ファイルは、両方が同じエイリアスを指定する場合、**registries.conf** ファイルよりも優先されることに注意してください。

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

例:

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

[aliases]

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

(JIRA:RHELPLAN-74542)

container-tools モジュールの変更

container-tools モジュールには、Podman、Buildah、Skopeo、runc の各ツールが含まれています。RHEL8 では **container-tools:rhel8** というストリームで表現されるローリングストリームは、RHEL9 では **container-tools:latest** という名前になっています。RHEL 8 と同様に、コンテナーツールの安定版は、番号付きのストリームで提供される予定です (例えば、3.0)。

Container Tools Application Stream については、Container Tools AppStream - Content Availability を参照してください。

(JIRA:RHELPLAN-73678)

containers-common が利用可能に

containers-common パッケージが **container-tools:latest** に追加されました。**containers-common** パッケージには、Podman、Buildah、Skopeo などのコンテナーツールのエコシステムに関する一般的な設定ファイルおよびドキュメントが含まれています。

(JIRA:RHELPLAN-77549)

新しいパッケージでコンテナーイメージの更新

たとえば、**registry.access.redhat.com/rhel9** コンテナーイメージを最新のパッケージで更新するには、以下のコマンドを使用します。

```
# podman run -it registry.access.redhat.com/rhel9
# dnf update -y && rm -rf /var/cache/dnf
```

特定の **<package>** をインストールするには、以下のコマンドを実行します

```
# dnf install <package>
```

詳細は、Adding software to a running UBI container を参照してください。

RHEL 9 では、イメージに新しいパッケージを更新またはインストールするには、使用する権利があるホストで実行している必要があります。Red Hat Enterprise Linux Developer Subscription for Individuals を使用すると、使用する権利のあるリポジトリーに無料でアクセスできます。

詳細は、No-cost Red Hat Enterprise Linux Individual Developer Subscription:FAQs を参照してください。

(JIRA:RHELPLAN-84168)

container-tools メタパッケージが更新された

Podman、Buildah、Skopeo、および runc ツールを含む **container-tools** RPM メタパッケージが利用できるようになりました。今回の更新で、以前のバージョンに対するバグ修正および機能拡張のリストが追加されました。

(JIRA:RHELPLAN-118914)

これで、podman-py パッケージが利用可能に

podman-py パッケージが、**container-tools:3.0** 安定モジュールストリームおよび **container-tools:latest** モジュールに追加されました。**podman-py** パッケージは、Podman の RESTful API を使用するバインディングのライブラリーです。

(BZ#1975462)

コントロールグループバージョン 2 が利用可能に

以前のバージョンのコントロールグループ cgroups バージョン 1 (cgroups v1) では、さまざまなアプリケーションでパフォーマンスの問題が発生しました。コントロールグループの最新リリースである cgroups バージョン 2 (cgroups v2) により、システム管理者はパフォーマンスの問題を発生させずに、どのアプリケーションのリソースも制限できます。

この新しいバージョンのコントロールグループ cgroups v2 は RHEL 8 で有効にでき、RHEL 9 のデフォルトで有効になっています。

(JIRA:RHELPLAN-73697)

container-tools メタパッケージが利用可能に

container-tools RPM メタパッケージには、Podman、Buildah、Skopeo、CRIU、Udica、および必要なすべてのライブラリーが含まれており、RHEL 9 で利用できます。安定したストリームは RHEL 9 では利用できません。Podman、Buildah、Skopeo などへの安定したアクセスを受けるには、RHEL EUS サブスクリプションを使用します。

container-tools meta-package をインストールするには、以下を入力します。

```
# dnf install container-tools
```

(BZ#2000871)

カーネルにおけるオーバーレイファイルシステムのネイティブサポートが利用可能

オーバーレイファイルシステムのサポートがカーネル 5.11 から利用可能になりました。非ルートユーザーは、ルートレスで (ユーザーとして) 実行しても、ネイティブなオーバーレイ性能を発揮します。

従って、この機能強化により、バインドマウントを必要とせずにオーバーレイファイルシステムを使用したい非 root ユーザーに、より良いパフォーマンスを提供します。

(JIRA:RHELPLAN-99892)

NFS ストレージが利用可能になりました

ファイルシステムで xattr がサポートされている場合は、NFS ファイルシステムをコンテナーとイメージのバックエンドストレージとして使用できるようになりました。

(JIRA:RHELPLAN-74543)

container-tools メタパッケージが更新された

container-tools メタパッケージには、Podman、Buildah、Skopeo、CRIU、Udica、および必要なすべてのライブラリーが含まれています。今回の更新で、以前のバージョンに対するバグ修正および機能拡張のリストが追加されました。

主な変更点は、以下のとおりです。

- ネットワークスタックの変更により、Podman v3 以前で作成されたコンテナーは Podman v4.0 では使用できなくなります。
- ネイティブオーバーレイファイルシステムがルートレスユーザーとして使用できる
- NFS ストレージがコンテナー内でサポートされるようになった
- コントロールグループバージョン 2 (cgroup v2) はデフォルトで有効になっています
- Podman v4 から v3 へのダウングレードは、すべてのコンテナーが破棄されて再作成されない限りサポートされません。

Podman の注目すべき変更点の詳細は、アップストリームのリリースノート を参照してください。

(JIRA:RHELPLAN-99889)

crun コンテナーランタイムがデフォルトになりました。

crun コンテナーランタイムがデフォルトのランタイムになりました。**crun** コンテナーランタイムは、コンテナーがルートレスユーザーの追加グループにアクセスできるようにするアノテーションをサポートします。これは、setgid が設定されたディレクトリーまたはユーザーがグループアクセスのみを持つディレクトリーにおけるボリュームマウントに役立ちます。**crun** および **runc** ランタイムは、どちらも **cgroup v2** を完全にサポートします。

(JIRA:RHELPLAN-99890)

コントロールグループバージョン 2 が利用できるようになりました。

以前のバージョンのコントロールグループ cgroups バージョン 1 (cgroup v1) では、さまざまなアプリケーションでパフォーマンスの問題が発生しました。コントロールグループの最新リリースである cgroup バージョン 2 (cgroup v2) により、システム管理者はパフォーマンスの問題を発生させずに、どのアプリケーションのリソースも制限できます。

RHEL 9 では、cgroup v2 がデフォルトで有効になっています。

(JIRA:RHELPLAN-75322)

ユニバーサルベースイメージが Docker Hub で利用可能に

これまでユニバーサルベースイメージは、Red Hat コンテナーカタログからしか入手できませんでした。この機能拡張により、ユニバーサルベースイメージも Docker Hub から 確認済みパブリッシャーイメージ として利用できます。

(JIRA:RHELPLAN-100032)

openssl コンテナーイメージが利用可能になりました

openssl イメージは、OpenSSL 暗号化ライブラリーのさまざまな機能を使用するための **openssl** コマンドラインツールを提供します。OpenSSL ライブラリーを使用すると、秘密鍵の生成、証明書署名要求 (CSR) の作成、および証明書情報の表示を行うことができます。

openssl コンテナーイメージは、次のリポジトリーで利用できます。

- registry.redhat.io/rhel9/openssl
- registry.access.redhat.com/ubi9/openssl

(JIRA:RHELPLAN-100034)

Netavark ネットワークスタックが利用可能になりました。

Netavark スタックは、コンテナーのネットワーク設定ツールです。RHEL 9 では、Netavark スタックは完全にサポートされ、デフォルトで有効になっています。

このネットワークスタックには、次の機能があります。

- ブリッジおよび MACVLAN インターフェイスを含むネットワークインターフェイスの作成、管理、および削除
- ネットワークアドレス変換 (NAT) やポートマッピングルールなどのファイアウォールの設定
- IPv4 および IPv6 (IPv4 and IPv6)
- 複数ネットワークのコンテナー機能の向上

(JIRA:RHELPLAN-101141)

Podman は、YAML ファイルを使用した Pod の自動ビルドと自動実行をサポートするようになりました

podman play kube コマンドは、YAML ファイルを使用して、Pod 内に複数のコンテナーを持つ複数の Pod を自動的にビルドして実行します。

(JIRA:RHELPLAN-108830)

Podman は、IdM から subUID および subGID の範囲を取得できるようになりました。

subUID と subGID の範囲を IdM で管理できるようになりました。同じ **/etc/subuid** ファイルおよび**/etc/subgid** ファイルをすべてのホストにデプロイする代わりに、単一の中央ストレージで範囲を定義できるようになりました。**/etc/nsswitch.conf** ファイルを変更し、**services: files sss** のようにサービスマップ行に **sss** を追加する必要があります。

詳細は、IdM ドキュメントの subID 範囲の手動管理 セクションを参照してください。

(JIRA:RHELPLAN-100020)

第5章 バグ修正

ここでは、ユーザーに大きな影響を及ぼしていた Red Hat Enterprise Linux 9.0 のバグで修正されたものを説明します。

5.1. インストーラーおよびイメージの作成

--leavebootorder が起動順序を変更しなくなりました。

以前では、ブートローダーのキックスタートコマンドに **--leavebootorder** を使用しても、UEFI システムでは正しく機能せず、起動順序が変更されていました。これにより、インストーラーは、UEFI 起動メニューのインストール済みシステムのリストの最上位に RHEL を追加しました。

今回の更新で問題が修正され、**--leavebootorder** を使用してもブートローダーの起動順序が変更しなくなりました。**--leavebootorder** は、UEFI システムの RHEL でサポートされるようになりました。

(BZ#2025953)

Anaconda は、静的ホスト名を設定してから %post スクリプトを実行します。

以前は、Anaconda がインストーラー環境のホスト名を Kickstart 設定 (**network --hostname**) からの値に設定する際に、一時的なホスト名を設定するのに使用されていました。ネットワークデバイスのアクティベーションなど、**%post** スクリプトの実行中に実行したアクションの一部で、リバース **dns** で取得した値にホスト名がリセットされた場合。

今回の更新で、Anaconda は、キックスタート **%post** スクリプトの実行時に、インストーラー環境の静的ホスト名を安定させるように設定するようになりました。

(BZ#2009403)

ユーザーは、RHEL for Edge インストーラーブループリントでユーザーアカウントを指定できるようになりました

以前は、rpm パッケージの追加など、アップグレードのedge-commitで定義されたユーザーアカウントなしでブループリントの更新を実行すると、アップグレードが適用された後、ユーザーがシステムからロックアウトされていました。これにより、ユーザーは既存のシステムをアップグレードするときにユーザーアカウントを再定義していました。この問題は、ユーザーが RHEL for Edge Installer ブループリントでユーザーアカウントを指定できるように修正されました。これにより、ユーザーを **ostree** コミットの一部。

(BZ#2060575)

basic graphics モードがブートメニューから削除されました

以前は、**basic graphics** モードを使用して、サポートされていないグラフィックカードを搭載したハードウェアに RHEL をインストールしたり、グラフィカルインターフェイスの起動を妨げるグラフィックドライバーの問題を回避したりしていました。この更新により、**basic graphics** モードでインストールするオプションがインストーラーのブートメニューから削除されました。サポートされないハードウェアでのグラフィカルインストールやドライバーバグの回避には、VNC インストールオプションを使用します。

VNC を使用したインストールの詳細については、Performing a remote RHEL installation using VNC セクションを参照してください。

(BZ#1961092)

5.2. サブスクリプションの管理

virt-who が Hyper-V ホストで正常に機能するようになる

以前は、**virt-who** を使用して Hyper-V ハイパーバイザーに RHEL 9 仮想マシンを設定すると、 **virt-who** がハイパーバイザーと正しく通信せず、設定に失敗していました。これは、**openssl** で暗号化メソッドが非推奨になっているためです。

今回の更新で、Hyper-V の **virt-who** 認証モードが修正され、**virt-who** を使用して Hyper-V に RHEL 9 仮想マシンを設定できるようになりました。これには、ハイパーバイザーが基本認証モードを使用することも必要であることに注意してください。このモードを有効にするには、以下のコマンドを使用します。

```
winrm set winrm/config/service/auth '@{Basic="true"}'
winrm set winrm/config/service '@{AllowUnencrypted="true"}'
```

(BZ#2008215)

5.3. ソフトウェア管理

モジュラーリポジトリーで createrepo_c --update を実行すると、モジュラーメタデータが保持されるようになりました

以前は、モジュラーメタデータの元のソースが存在しない状態で既存のモジュラーリポジトリーで **createrepo_c --update** コマンドを実行すると、デフォルトのポリシーでは、モジュラーメタデータを含むすべての追加メタデータがこのリポジトリーから削除され、その結果、リポジトリーが破損していました。メタデータを保持するには、追加の **--keep-all-metadata** オプションを指定して **createrepo_c --update** コマンドを実行する必要がありました。

この更新では、追加オプションなしで **createrepo_c --update** を実行することにより、モジュラーリポジトリーにモジュラーメタデータを保持できます。

追加のメタデータを削除するには、新しい **--discard-additional-metadata** オプションを使用できます。

(BZ#2055032)

5.4. シェルおよびコマンドラインツール

RHEL 9 は libservicelog1.1.19 を提供します

RHEL 9 には、**libservicelog** バージョン 1.1.19 が同梱されています。以下は、主なバグ修正です。

- 出力アライメントの問題を修正
- **servicelog_open()** の失敗時の **segfault** が修正されました。

(BZ#1869568)

5.5. セキュリティー

FIPS モードの場合に libgcrypt でハードウェアの最適化が有効化

これまでの Federal Information Processing Standard(FIPS 140-2) では、ハードウェアの最適化を使用することは認められていませんでした。したがって、RHEL の以前のバージョンでは、FIPS モードの

場合は **libgcrypt** パッケージで操作が無効でした。RHEL 9 は、FIPS モードでハードウェアの最適化を有効にします。その結果、すべての暗号化操作が速く実行されます。

(BZ#1990059)

crypto-policies が ChaCha20 暗号化の使用を無効にできるようになりました。

以前では、**crypto-policies** パッケージは誤ったキーワードを使用して OpenSSL で **ChaCha20** 暗号を無効にしていました。したがって、**crypto-policies** で OpenSSL の TLS 1.2 プロトコルの **ChaCha20** を無効にできませんでした。今回の更新では、**-Chacha20-poly1305** の代わりに **-CHACHA20** キーワードが使用されます。その結果、暗号化ポリシーを使用して、TLS1.2 および **TLS1.3** の OpenSSL で ChaCha20 暗号の使用を無効にできるようになりました。

(BZ#2004207)

FIPS モードでインストールするときに 64 ビット IBM Z システムが起動できなくなることはなくなりました

以前は、**--no-bootcfg** オプションを指定した **fips-mode-setup** コマンドは **zipl** ツールを実行しませんでした。**fips-mode-setup** は初期 RAM ディスク (**initrd**) を再生成し、その結果であるシステムを起動するには **zipl** 内部状態を更新する必要があるため、FIPS モードでインストールした後、64 ビット IBM Z システムは起動できない状態になります。この更新により、**fips-mode-setup** は、**-no-bootcfg** を指定して呼び出された場合でも、64 ビット IBM Z システムで **zipl** を実行するようになり、その結果、新しくインストールされたシステムが正常に起動します。

(BZ#2013195)

GNUTLS_NO_EXPLICIT_INIT が暗黙的なライブラリーの初期化を無効にしなくなりました。

以前は、**GNUTLS_NO_EXPLICIT_INIT** 環境変数が暗黙的なライブラリーの初期化を無効にしていました。RHEL 9 では、**GNUTLS_NO_IMPLICIT_INIT** 変数は、代わりに暗黙的なライブラリーの初期化を無効にします。

(BZ#1999639)

OpenSSL ベースのアプリケーションが、Turkish ロケールで正しく動作するようになりました。

OpenSSL ライブラリーは大文字と小文字を区別しない文字列比較関数を使用するため、OpenSSL ベースのアプリケーションはトルコ語ロケールで正しく機能せず、チェックを省略すると、このロケールを使用するアプリケーションがクラッシュしました。この更新プログラムは、大文字と小文字を区別しない文字列比較のために Portable Operating System Interface (POSIX) ロケールを使用するためのパッチを提供します。その結果、curl などの OpenSSL ベースのアプリケーションはトルコ語のロケールで正しく機能します。

(BZ#2071631)

SELinux のパーミッションが原因で kdump がクラッシュすることがなくなりました。

kdump クラッシュリカバリーサービスを正しく起動するには、追加の SELinux 権限が必要です。そのため、以前のバージョンでは、SELinux が **kdump** の動作を妨げ、**kdump** が動作していないことを報告したり、Access Vector Cache(AVC) の拒否が監査されたりしていました。このバージョンでは、必要なパーミッションが **selinux-policy** に追加され、その結果、**kdump** が正しく動作し、AVC の拒否が監査されません。

(BZ#1932752)

usbguard-selinux パッケージが usbguard に依存しなくなりました。

usbguard-selinux パッケージは、以前は **usbguard** パッケージに依存していました。これを、このパッケージの他の依存関係と組み合わせると、**usbguard** のインストール時にファイル競合が発生しました。そのため、特定システムに **usbguard** がインストールされなくなりました。このバージョンでは、**usbguard-selinux** は **usbguard** に依存しなくなり、その結果、**dnf** は **usbguard** を正しくインストールできます。

(BZ#1986785)

dnf install および dnf update が SELinux の fapolicyd で動作するようになりました。

fapolicyd の SELinux ルールを含む **fapolicyd-selinux** パッケージには、すべてのファイルとディレクトリーを監視するためのパーミッションが含まれていませんでした。これにより、**fapolicyd-dnf-plugin** が正常に動作せず、**dnf install** コマンドや **dnf update** コマンドにより、システムが無期限に応答しなくなりました。このバージョンでは、**fapolicyd-selinux** に任意のファイルタイプを見ることができる権限が追加されました。その結果、**fapolicyd-dnf-plugin** が正しく動作し、**dnf install** と **dnf update** のコマンドが動作するようになりました。

(BZ#1932225)

アンビエント機能が root 以外のユーザーに正しく適用されるようになりました

安全対策として、UID (ユーザー識別子) をルートから非ルートに変更すると、許可された有効な一連のアンビエント機能セットが無効になっていました。

しかし、アンビエントセットに含まれる機能は許可されたセットと継承可能なセットの両方に含まれている必要があるため、**pam_cap.so** モジュールはアンビエント機能を設定できません。さらに、たとえば **setuid** ユーティリティーを使用して) UID を変更すると許可されたセットが無効になるため、アンビエント機能を設定できません。

この問題を修正するために、**pam_cap.so** モジュールは **keepcaps** オプションをサポートするようになりました。これにより、プロセスは、UID をルートから非ルートに変更した後も許可された機能を保持できます。**pam_cap.so** モジュールは、**defer** オプションもサポートするようになりました。これにより、**pam_cap.so** は、**pam_end()** へのコールバック内でアンビエント機能を再適用します。このコールバックは、UID を変更した後、他のアプリケーションで使用できます。

そのため、**su** ユーティリティーおよび **login** ユーティリティーが更新済みで PAM に準拠している場合は、**keepcaps** オプションおよび **defer** オプションを指定して **pam_cap.so** を使用し、root 以外のユーザーにアンビエント機能を設定できるようになりました。

(BZ#2037215)

usbguard-notifier が Journal に記録するエラーメッセージの数が適正になりました

以前は、**usbguard-notifier** サービスに **usbguard-daemon** IPC インターフェイスに接続するためのプロセス間通信 (IPC) のパーミッションがありませんでした。したがって、**usbguard-notifier** はインターフェイスへの接続に失敗し、対応するエラーメッセージがジャーナルに書き込まれていました。**usbguard-notifier** は **--wait** オプションで始まるため、デフォルトでは接続障害後に毎秒 **usbguard-notifier** が IPC インターフェイスへの接続を試みるため、ログにはこれらのメッセージが過剰に含まれていました。

今回の更新により、**usbguard-notifier** はデフォルトで **--wait** で開始されなくなりました。サービスは、1 秒間隔で 3 回だけデーモンへの接続を試みます。その結果、ログには最大で 3 つのエラーメッセージが含まれます。

(BZ#2009226)

5.6. ネットワーク

Wifi および 802.1x のイーサネット接続プロファイルが適切に接続されるようになる

以前は、多くの Wifi および 802.1x のイーサネット接続プロファイルでは接続できませんでした。このバグが修正されました。すべてのプロファイルが適切に接続されるようになりました。従来の暗号アルゴリズムを使用するプロファイルも引き続き機能しますが、OpenSSL レガシープロバイダーを手動で有効にする必要があります。これは、MS-CHAPv2 で DES を使用したり、TKIP で RC4 を使用する場合などに必要です。

(BZ#1975718)

afterburn が /etc/hostnameに長いホスト名を設定しない

RHEL ホスト名の最大長は 64 文字です。ただし、特定のクラウドプロバイダーは、ホスト名として完全修飾ドメイン名 (FQDN) を使用します。これは最大 255 文字です。以前は、**afterburn-hostname** サービスはそのようなホスト名を直接 **/etc/hostname** ファイルに書き込みました。**systemd** サービスはホスト名を 64 文字に切り捨てられ、NetworkManager は省略された値から誤った DNS 検索ドメインを取得しています。今回の修正により、**afterburn-hostname** は最初のドットまたは 64 文字でホスト名を切り捨てます。これが最初に表示されるようになりました。これにより、NetworkManager は、**/etc/resolv.conf** に無効な DNS 検索ドメインを設定しなくなりました。

(BZ#2008521)

5.7. カーネル

modprobe は、out-of-tree カーネルモジュールを期待どおりに読み込みます。

/etc/depmod.d/dist.conf 設定ファイルは、**depmod** ユーティリティーの検索順序を提供します。検索順序に基づいて、**depmod** は **modules.dep.bin** ファイルを作成します。このファイルは、**modprobe** ユーティリティーがカーネルモジュールの読み込みとアンロード、およびモジュールの依存関係の解決に同時に使用するモジュールの依存関係のリストを表示します。以前は、**/etc/depmod.d/dist.conf** がありませんでした。その結果、**modprobe** は一部の out-of-tree カーネルモジュールを読み込むことができませんでした。この更新には、検索順序を修正する **/etc/depmod.d/dist.conf** 設定ファイルが含まれています。これにより、**modprobe** は、想定通りに out-of-tree カーネルモジュールを読み込みます。

(BZ#1985100)

alsa-lib が UCM を使用するオーディオデバイスを正しく処理するようになりました。

alsa-lib パッケージのバグにより、内部ユースケースマネージャー (UCM) 識別子が誤って解析されていました。そのため、UCM 設定を使用する一部の音声デバイスは検出されないか、正常に機能しませんでした。この問題は、システムが **pipewire** サウンドサービスを使用すると、より頻繁に発生しました。RHEL 9 の新しいリリースでは、**alsa-lib** ライブラリーを更新することで問題が修正されました。

(BZ#2015863)

5.8. ファイルシステムおよびストレージ

保護イベントにより、マルチパスデバイスのリロードエラーが発生しなくなりました

以前は、**read-only** パスデバイスが再スキャンされると、カーネルは 2 つの書き込み保護イベントを送信しました。1 つはデバイスが **read/write** に設定され、次はデバイスが **read-only** に設定されています。その結果、パスデバイスで **read/write** イベントが検出されると、**multipathd** はマルチパスデバイスをリロードしようとしました。これにより、リロードエラーメッセージが表示されました。この更新により、**multipathd** は、デバイスの読み取り/書き込みをリロードする前に、すべてのパスが **read/write** に設定されていることを確認するようになりました。その結果、**multipathd** は、**read-only** デバイスが再スキャンされるたびに **read/write** をリロードしようとしなくなりました。

(BZ#2017979)

device-mapper-multipath がバージョン 0.8.7 にリベースされました。

device-mapper-multipath パッケージがバージョン 0.8.7 にアップグレードされ、バグ修正および機能強化が複数追加されました。主な変更点は、以下のとおりです。

- **multipath** コマンドおよび **kpartx** コマンドのメモリーリークを修正しました。
- **multipathd.socket** ユニットファイルから繰り返し発生するエラーを修正
- DELL SC シリーズアレイ、EMC Invista、Symmetrix アレイ (とりわけ) など、より多くのデバイスの自動設定が改善されました。

(BZ#2017592)

5.9. 高可用性およびクラスター

Pacemaker 属性マネージャーがリモートノード属性を正しく判断し、ループのフェンシングが解除されないようにする

以前では、ノードの Pacemaker のコントローラーは、属性マネージャーが、already-active リモートノードがリモートであることを認識する前に、Designated Controller (DC) に選択されていました。これが発生すると、ノードのスケジューラーは、リモートノードのノード属性を認識しません。クラスターがフェンシング解除を使用すると、フェンシング解除ループが発生する可能性があります。この修正により、属性マネージャーは、起動時の初期属性同期などの追加イベントにより、リモートノードがリモートであることを学習できるようになりました。その結果、どのノードが DC に選択されているかに関係なく、フェンシング解除ループが発生しません。

(BZ#1975388)

5.10. コンパイラーおよび開発ツール

-Wsequence-point 警戒行動修正

以前のリリースでは、GCC で C++ プログラムをコンパイルする際に、**-Wsequence-point** 警告オプションが非常に長い式について警告を試みたため、クリアルな動作が発生し、コンパイル時間が大幅に長くなる可能性がありました。この更新により、**-Wsequence-point** は極端に大きな式について警告しようとせず、その結果、コンパイル時間が増加しません。

(BZ#1481850)

5.11. IDENTITY MANAGEMENT

OpenSSL レガシープロバイダーを使用した MS-CHAP 認証

以前は、MS-CHAP を使用する FreeRADIUS 認証機構は MD4 ハッシュ関数に依存していたため失敗していましたが、RHEL 9 では MD4 は非推奨となりました。今回の更新で、OpenSSL レガシープロバイダーを有効にすると、MS-CHAP または MS-CHAPv2 で FreeRADIUS ユーザーを認証できるようになりました。

デフォルトの OpenSSL プロバイダーを使用する場合は、MS-CHAP および MS-CHAPv2 認証が失敗し、修正を示す以下のエラーメッセージが表示されます。

```
Couldn't init MD4 algorithm. Enable OpenSSL legacy provider.
```

(BZ#1978216)

sudo コマンドを実行しても、KRB5CCNAME 環境変数をエクスポートしなくなりました。

以前のバージョンでは、**sudo** コマンドの実行後に、環境変数 **KRB5CCNAME** は、元のユーザーの Kerberos 認証情報キャッシュを参照していましたが、ターゲットユーザーがアクセスできない場合がありました。そのため、このキャッシュにアクセスできないため、Kerberos 関連の操作が失敗する可能性がありました。今回の更新で、**sudo** コマンドを実行しても **KRB5CCNAME** 環境変数が設定されなくなり、ターゲットユーザーがデフォルトの Kerberos 認証情報キャッシュを使用できるようになりました。

(BZ#1879869)

SSSD が、/etc/krb5.conf の Kerberos キータブ名のデフォルト設定を正しく評価

以前は、**krb5.keytab** ファイルの標準以外の場所を定義した場合は、SSSD はこの場所を使用せず、代わりにデフォルトの **/etc/krb5.keytab** の場所を使用していました。したがって、システムへのログイン試行時に、**/etc/krb5.keytab** にエントリーが含まれていないため、ログインに失敗していました。

今回の更新で、SSSD は **/etc/krb5.conf** の **default_keytab_name** 変数を評価し、この変数で指定された場所を使用するようになりました。**default_keytab_name** 変数が設定されていない場合にのみ、SSSD はデフォルトの **/etc/krb5.keytab** の場所を使用します。

(BZ#1737489)

PBKDF2 アルゴリズムでハッシュされたパスワードを使用した FIPS モードでの Directory Server への認証が期待どおりに機能するようになりました

Directory Server が FIPS (Federal Information Processing Standard) モードで実行している場合は、**PK11_ExtractKeyValue()** 機能を使用できません。その結果、この更新の前は、パスワードベースの鍵導出関数 2 (PBKDF2) アルゴリズムでハッシュされたパスワードを持つユーザーは、FIPS モードが有効になっているときにサーバーに対して認証できませんでした。今回の更新で、Directory Server が **PK11_Decrypt()** 機能を使用してパスワードハッシュデータを取得するようになりました。その結果、PBKDF2 アルゴリズムでハッシュされたパスワードによる認証が期待どおりに機能するようになりました。

(BZ#1779685)

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

IPv6 が無効になると、Networking システムロールが DNS 検索ドメインの設定に失敗しなくなりました。

以前では、IPv6 プロトコルが無効になっている場合、**libnm** ライブラリーの **nm_connection_verify()** 機能は DNS 検索ドメインを無視していました。そのため、Networking RHEL システムロールを使用し、**dns_search** を **ipv6_disabled: true** と一緒に設定すると、システムロールに障害が発生し、以下のエラーが表示されます。

```
nm-connection-error-quark: ipv6.dns-search: this property is not allowed for 'method=ignore' (7)
```

今回の更新で、IPv6 が無効になっている場合、**nm_connection_verify()** 機能は DNS 検索ドメインを無視します。これにより、IPv6 が無効になっていても、期待どおりに **dns_search** を使用できます。

(BZ#2004899)

postfix ロール README がプレーンロール名を使用しなくなる

以前は、**/usr/share/ansible/roles/rhel-system-roles.postfix/README.md** で提供される例では、**rhel-system-roles.postfix** ではなく、プレーンバージョンのロール名 **postfix** を使用していました。そのため、ユーザーはドキュメントを参照し、完全修飾ロール名 (FQRN) ではなくプレーンロール名を誤って使用していました。今回の更新で問題が修正され、ドキュメントでは FQRN、**rhel-system-roles.postfix** の例が含まれ、ユーザーは Playbook を正しく作成できるようになりました。

(BZ#1958964)

Postfix RHEL システムロールの README.md の Role Variables セクションで、変数が欠落しなくなりました

以前は、**postfix_check**、**postfix_backup**、**postfix_backup_multiple** などの Postfix RHEL システムロール変数は、Role Variables セクションでは使用できません。そのため、Postfix ロールのドキュメントを参照できませんでした。今回の更新で、Postfix README セクションに、ロール変数のドキュメントが追加されました。ロール変数は、**rhel-system-roles** が提供する **doc/usr/share/doc/rhel-system-roles/postfix/README.md** ドキュメントで文書化され、ユーザーが利用できるようになりました。

(BZ#1978734)

同じ出力を実行する際にロールタスクが変わらない

以前のバージョンでは、同じ入力をもう一度実行しても、変更がない場合でも、ロールタスクの一部は **CHANGED** として報告されていました。そのため、ロールはべき等性を持ちませんでした。この問題を修正するには、以下のアクションを実行します。

- 設定変数の変更を確認してから、それらを適用します。この検証には **--check** オプションを使用できます。
- **Last Modified: $date** ヘッダーを設定ファイルに追加しないでください。

その結果、ロールタスクはべき等になります。

(BZ#1978760)

logging_purge_confs オプションは不要な設定ファイルを正しく削除

logging_purge_confs オプションを **true** に設定すると、不要なログ設定ファイルが削除されます。ただし、以前は、**logging_purge_confs** が **true** に設定されていても、不要な設定ファイルは設定ディレクトリーから削除されませんでした。この問題は修正され、オプションは次のように再定義されました。**logging_purge_confs** が **true** に設定されている場合、Rsyslog は **rsyslog.d** ディレクトリーから rpm パッケージに属していないファイルを削除します。これには、以前の実行で生成される Logging ロールが含まれます。**logging_purge_confs** のデフォルト値は **false** です。

(BZ#2039106)

Grafana admin パスワードが変更された場合でも、Metrics ロールを使用する Playbook は複数回の実行で正常に完了します

以前は、**metrics_graph_service: yes** ブール値で Metrics ロールを実行した後に Grafana **admin** ユーザーパスワードを変更すると、Metrics ロールの後続の実行が失敗していました。これにより、Metrics ロールを使用した Playbook が失敗し、影響を受けるシステムはパフォーマンス分析用にのみ部分的に設定されました。現在、Metrics ロールは、Grafana **deployment** API が利用可能であり、必要な設定アクションを実行するためにユーザー名またはパスワードの知識を必要としない場合に、その API を使用します。その結果、管理者が Grafana **admin** パスワードを変更した場合でも、Metrics ロールを使用する Playbook は複数回の実行で正常に完了します。

(BZ#2041632)

Metrics ロールによる設定がシンボリックリンクを正しくたどるようになりました

mssql pcp パッケージがインストールされると、**mssql.conf** ファイルは **/etc/pcp/mssql/** に配置され、シンボリックリンク **/var/lib/pcp/pmdas/mssql/mssql.conf** のターゲットになります。ただし、以前の Metrics ロールはシンボリックリンクをたどって、**mssql.conf** を設定する代わりにシンボリックリンクを上書きしていました。その結果、Metrics ロールを実行すると、シンボリックリンクが通常のファイルに変更され、**/var/lib/pcp/pmdas/mssql/mssql.conf** ファイルのみ設定の影響を受けました。これによりシンボリックリンクが失敗し、メインの設定ファイル **/etc/pcp/mssql/mssql.conf** は設定の影響を受けませんでした。この問題は修正され、シンボリックリンクをたどる **follow: yes** オプションが Metrics ロールに追加されました。その結果、Metrics ロールはシンボリックリンクを保持し、メイン設定ファイルを正しく設定します。

(BZ#2058777)

timesync ロールは、要求されたサービス ptp4lの検索に失敗しなくなりました。

以前は、RHEL の一部バージョンでは、Ansible の **service_facts** モジュールでサービスのファクトが誤って報告されていました。そのため、**timesync** ロールは **ptp4l** サービスの起動を試行するエラーを報告していました。今回の修正により、Ansible **service_facts** モジュールはタスクの戻り値をチェックして **timesync** サービスを停止します。返された値が **failed** したものの、エラーメッセージが **Could not find the requested service NAME:** である場合、モジュールは成功を想定します。その結果、**timesync** ロールは **Could not find the requested service ptp4l** などのエラーなしで実行されるようになりました。

(BZ#2058645)

kernel_settings configobj はマネージドホストで利用できます。

以前は、**kernel_settings** ロールはマネージドホストに **python3-configobj** パッケージをインストールしませんでした。そのため、**configobj** Python モジュールが見つからないことを示すエラーが返されました。この修正により、ロールは **python3-configobj** パッケージがマネージドホストに存在し、**kernel_settings** ロールが期待どおりに機能することを保証します。

(BZ#2058756)

Terminal Session Recording ロール tlog-rec-session が SSSD によって正しくオーバーレイされるようになりました

以前は、Terminal Session Recording RHEL System Role は、システムセキュリティーサービスデーモン (SSSD) ファイルプロバイダーと有効な **authselect** オプション **with-files-domain** に依存して、**nsswitch.conf** ファイルに正しい **passwd** エントリーを設定していました。RHEL 9.0 では、SSSD はデフォルトではファイルプロバイダーを暗黙的に有効にせず、SSSD による **tlog-rec-session** シェルオーバーレイは機能しませんでした。今回の修正により、Terminal Session Recording ロールが更新され、**nsswitch.conf** が更新され、SSSD が **tlog-rec-session** が正しくオーバーロードするようになりました。

(BZ#2071804)

SSHD システムロールは FIPS モードでシステムを管理できます。

以前は、SSHD システムロールは、呼び出し時に **not allowed** HostKey タイプを作成できませんでした。これにより、SSHD システムロールは FIPS (Federal Information Processing Standard) モードで RHEL 8 および古いシステムを管理できませんでした。今回の更新で、SSHD システムロールが FIPS モードを検出し、デフォルトの HostKey リストを正しく調整するようになりました。その結果、システムのロールは、デフォルトの HostKey 設定を使用して FIPS モードで RHEL システムを管理できます。

(BZ#2029634)

SSHD システムロールは正しいテンプレートファイルを使用する

以前は、SSHD システムロールが間違ったテンプレートファイルを使用していました。そのため、生成された **sshd_config** ファイルには **ansible_managed** コメントが含まれていませんでした。この更新では、システムロールが正しいテンプレートファイルを使用し、**sshd_config** に正しい **ansible_managed** コメントが含まれています。

(BZ#2044408)

Kdump RHEL システムロールは再起動できるか、再起動が必要であることを示します。

以前では、Kdump RHEL システムロールは、クラッシュカーネル用に予約メモリーのないマネージドノードを無視していました。そのため、システムが正しく設定されていない場合でも、ロールは Success ステータスで終了しました。RHEL 9 の今回の更新で、この問題が修正されました。管理ノードでクラッシュカーネル用にメモリーが予約されていない場合には、Kdump RHEL システムロールが失敗し、ユーザーが **kdump_reboot_ok** 変数を **true** に設定して管理ノードで **kdump** サービスを適切に設定することが示唆されます。

(BZ#2029602)

Networking システムロールの nm プロバイダーがブリッジを正しく管理するようになりました

以前は、**initscripts** プロバイダーを使用した場合、Networking システムロールは、ブリッジインターフェイスをマネージド外としてマークするように NetworkManager を設定する **ifcfg** ファイルを作成しました。また、NetworkManager はフォローアップ **initscript** アクションを検出できませんでした。たとえば、initscript プロバイダーの **down** および **absent** アクションは、**down** および **absent** アクションの後に接続をリロードしない場合は、このインターフェイスのアンマネージ状態に関する NetworkManager の理解を変更しません。この修正により、Netwoking システムロールは **NM.Client.reload_connections_async()** 関数を使用して、NetworkManager 1.18 を使用してマネージドホストに NetworkManager をリロードします。その結果、NetworkManager は、プロバイダーを **initscript** から **nm** に切り替えるときにブリッジインターフェイスを管理します。

(BZ#2038957)

正しいボンディングモードの active-backup をサポートするようにタイプミスを修正する

以前は、**active-backup** ボンディングモードを指定する際に InfiniBand ポートをサポートする際に、タイプミス (**active_backup**) がありました。このタイプミスが原因で、接続は InfiniBand ボンディングポートの正しいボンディングモードをサポートできませんでした。この更新では、ボンディングモードを **active-backup** に変更することで、タイプミスを修正しています。これで、接続は InfiniBand ボンディングポートを正常にサポートします。

(BZ#2064391)

Logging システムロールロールがタスクを複数回呼び出さなくなりました

以前は、Logging ロールは 1 回だけ呼び出すべきタスクを複数回呼び出していました。そのため、余分なタスク呼び出しによりロールの実行速度が低下していました。この修正では、Logging ロールが変更され、タスクを 1 回だけ呼び出すようになり、Logging ロールのパフォーマンスが向上しました。

(BZ#2004303)

RHEL システムロールは、生成されたファイル内の複数行の ansible_managed コメントを処理するようになりました

以前は、一部の RHEL システムロールは **#{{ansible_managed}}** を使用して一部のファイルを生成していました。そのため、顧客が複数行のカスタム **ansible_managed** 設定を持っている場合、ファイルは正しく生成されませんでした。今回の修正により、すべてのシステムロールでファイルの生成時に **{{**

ansible_managed | comment }} と同等のものが使用されるようになり、複数行の **ansible_managed** 値を含め、**ansible_managed** 文字列は常に正しくコメントされるようになりました。その結果、生成されたファイルには正しい複数行の **ansible_managed** 値が含まれます。

(BZ#2006230)

Firewall システムロールは、target が変更されるとすぐにファイアウォールをリロードするようになりました

以前は、**target** パラメーターが変更されても Firewall システムロールはファイアウォールを再ロードしませんでした。この修正により、**target** が変更されると Firewall ロールがファイアウォールを再ロードするようになりました。その結果、**tatrget** の変更は即座に行われ、後続の操作で使用できるようになりました。

(BZ#2057164)

Certificate System ロールの group オプションでは、証明書がグループからアクセスできなくなるようになりました。

以前は、証明書のグループを設定する際に、**mode** はグループの読み取り権限を許可するように設定されませんでした。そのため、グループメンバーは、Certificate ロールが発行した証明書を読み取ることができませんでした。今回の修正で、グループ設定により、ファイルモードにグループ読み取り権限が確実に含まれるようになりました。その結果、グループの Certificate ロールにより発行された証明書に、グループメンバーがアクセスできます。

(BZ#2021025)

Logging ロールは、immark モジュールの間隔値の引用を見逃さなくなりました

以前のバージョンでは、**immark** モジュールが適切に設定されていないため、**immark** モジュールの **interval** フィールド値が適切に引用されませんでした。今回の修正により、**interval** の値が適切に引用符で囲まれるようになりました。これで、**immark** モジュールは期待どおりに機能します。

(BZ#2021676)

/etc/tuned/kernel_settings/tuned.conf ファイルには適切な ansible_managed ヘッダーがあります

以前では、**kernel_settings** RHEL システムロールは、**/etc/tuned/kernel_settings/tuned.conf** ファイルの **ansible_managed** ヘッダーにハードコーディングされた値がありました。その結果、ユーザーはカスタムの **ansible_managed** ヘッダーを提供できませんでした。この更新で問題が修正され、**kernel_settings** が **/etc/tuned/kernel_settings/tuned.conf** のヘッダーをユーザーの **ansible_managed** 設定で更新するようになりました。つまり、**/etc/tuned/kernel_settings/tuned.conf** には適切な **ansible_managed** ヘッダーがあります。

(BZ#2047506)

VPN システムロールフィルタープラグイン vpn_ipaddr が FQCN (完全修飾コレクション名) に変換するようになりました。

以前は、レガシーロールフォーマットからコレクションフォーマットへの変換で、フィルタープラグインの **vpn_ipaddr** が FQCN (Fully Qualified Collection Name) **redhat.rhel_system_roles.vpn_ipaddr** に変換されないことがありました。これにより、VPN ロールは短縮名でプラグインを見つけなくなり、エラーを報告していました。今回の修正により、変換スクリプトが変更され、フィルターがコレクションの FQCN 形式に変換されるようになりました。また、VPN ロールはエラーを発行せずに実行されるようになりました。

(BZ#2050341)

kdump.service のジョブに失敗しなくなりました。

以前は、カーネルクラッシュサイズを設定するための Kdump ロールコードは RHEL9 用に更新されていませんでした。これには、**kdumpctlreset-crashkernel** を使用する必要があります。これにより、**kdump.service** が起動しず、エラーが発生していました。今回の更新で、**kdump.service** ロールは **kdumpctl reset-crashkernel** を使用してクラッシュカーネルサイズを設定できるようになりました。**kdump.service** ロールは kdump サービスを正常に起動し、カーネルクラッシュサイズが正しく設定されるようになりました。

(BZ#2050419)

5.13. 仮想化

マウントされた仮想ディスクをホットプラグすると、ゲストカーネルが IBM Z でクラッシュしなくなりました。

以前は、マウントされたディスクを IBM Z ハードウェア上で実行中の仮想マシン (VM) から切り離すと、VM カーネルが次の条件下でクラッシュしました。

- ディスクはターゲットバスタイプ **scsi** に割り当てられ、ゲスト内にマウントされています。
- ディスクデバイスのホットプラグ後、対応する SCSI コントローラーもホットプラグされました。

今回の更新で、基礎となるコードが修正され、上記のクラッシュが発生しなくなりました。

(BZ#1997541)

5.14. コンテナー

UBI 9-Beta コンテナーは、RHEL 7 および 8 ホストで実行できます。

UBI 9-Beta コンテナーイメージでは、**containers-common** パッケージに間違った seccomp プロファイルが設定されていました。そのため、コンテナーが特定のシステムコールに対応できず、障害が発生しました。今回の更新で、この問題が修正されています。

(BZ#2019901)

第6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。

6.1. RHEL FOR EDGE

FDO プロセスがテクノロジープレビューとして利用可能に

RHEL for Edge イメージの自動プロビジョニングおよびオンボード用の FDO プロセスは、テクノロジープレビューとして利用できます。これにより、RHEL for Edge Simplified Installer イメージを構築し、それを RHEL for Edge イメージにプロビジョニングし、FDO(FIDO デバイスオンボーディング) プロセスを使用して、Edge デバイスを自動的にプロビジョニングおよびオンボーディングし、ネットワークに接続されている他のデバイスやシステムとデータを交換できます。その結果、FIDO デバイスのオンボーディングプロトコルは、製造段階でデバイスの初期化を実行し、次に実際にデバイスを使用するための遅延バインディングを実行します。

(BZ#1989930)

6.2. 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テクノロジープレビューとして利用できます。

Basic Relax and Recover (ReaR) 機能が、64 ビットの IBM Z アーキテクチャーでテクノロジープレビューとして利用できるようになりました。IBM Z では、z/VM 環境でのみ ReaR レスキューイメージを作成できます。論理パーティション (LPAR) のバックアップおよび復元はテストされていません。

現在利用できる出力方法は、Initial Program Load (IPL) のみです。IPL は、**zIPL** ブートローダーで使用できるカーネルと初期 ramdisk (initrd) を生成します。

警告

現在、レスキュープロセスは、システムに接続したすべての DASD (Direct Attached Storage Devices) を再フォーマットします。システムのストレージデバイスに貴重なデータが存在する場合は、システムの復旧を行わないでください。これには、レスキュー環境で起動するのに使用された **zIPL** ブートローダー、ReaR カーネル、および initrd で準備されたデバイスも含まれます。必ずコピーを保管してください。

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。

(BZ#2046653)

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。

(BZ#2047161)

6.3. 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。

(BZ#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。

(BZ#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です。

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。

(BZ#2020529)

6.4. カーネル

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りました。

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。

(BZ#2030412)

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。

(BZ#1874182)

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

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。

(BZ#2023416)

6.5. 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。

(BZ#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 のドキュメント を参照してください。Stratis ファイルシステムの設定

(BZ#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りました。

NVMexpress.org Technical Proposals (TP) 8013 および 8014 で定義されている NVMe-oF Discovery Service の機能は、テクノロジープレビューとして利用できます。これらの機能をプレビューするには、**nvme-cli 2.0** パッケージを使用して、TP-8013 または TP-8014 を実装する NVMe-oF ターゲットデバイスにホストを割り当てます。TP-8013 および TP-8014 の詳細は、https://nvmexpress.org/developers/nvme-specification/ の Web サイトの NVM Express 2.0 Ratified TPs を参照してください。

(BZ#2021672)

6.6. 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に

RHEL 9 には、**jmc-core** パッケージおよび **owasp-java-encoder** パッケージでテクノロジープレビューとして配布されます。

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。

(BZ#1980981)

6.7. 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。

(BZ#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。

(BZ#2084166)

ACME がテクノロジープレビューとして利用可能

Automated Certificate Management Environment (ACME) サービスが、テクノロジープレビューとして Identity Management (IdM) で利用可能になりました。ACME は、自動化識別子の検証および証明書の発行に使用するプロトコルです。この目的は、証明書の有効期間を短縮し、証明書のライフサイクル管理での手動プロセスを回避することにより、セキュリティーを向上させることです。

RHEL では、ACME サービスは Red Hat Certificate System (RHCS) PKI ACME レスポンダーを使用します。RHCS ACME サブシステムは、IdM デプロイメントのすべての認証局 (CA) サーバーに自動的にデプロイされますが、管理者が有効にするまでリクエストに対応しません。RHCS は、ACME 証明書を発行する際に **acmeIPAServerCert** プロファイルを使用します。発行された証明書の有効期間は 90 日です。ACME サービスの有効化または無効化は、IdM デプロイメント全体に影響します。

重要

ACME は、すべてのサーバーが RHEL 8.4 以降を実行している IdM デプロイメントでのみ有効にすることが推奨されます。以前の RHEL バージョンには ACME サービスが含まれていないため、バージョンが混在するデプロイメントで問題が発生する可能性があります。たとえば、ACME のない CA サーバーは、異なる DNS サブジェクト代替名 (SAN) を使用しているため、クライアント接続が失敗する可能性があります。

警告

現在、RHCS は期限切れの証明書を削除しません。ACME 証明書は 90 日後に期限切れになるため、期限切れの証明書が蓄積され、パフォーマンスに影響を及ぼす可能性があります。

- IdM デプロイメント全体で ACME を有効にするには、**ipa-acme-manage enable** コマンドを使用します。

```
# ipa-acme-manage enable
The ipa-acme-manage command was successful
```

- IdM デプロイメント全体で ACME を無効にするには、**ipa-acme-manage disable** コマンドを使用します。

```
# ipa-acme-manage disable
The ipa-acme-manage command was successful
```

- ACME サービスがインストールされ、有効または無効であるかを確認するには、**ipa-acme-manage status** コマンドを使用します。

```
```

```
# ipa-acme-manage status
ACME is enabled
The ipa-acme-manage command was successful
```

(BZ#2084181)

6.8. 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)

6.9. 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)

6.10. 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)

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

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 および Intel 64 で virtio-mem が利用できるようになりました。

テクノロジープレビューとして、RHEL 9 では、AMD64 および Intel 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。

(BZ#2014487)

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)

6.11. コンテナー

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7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。サポート期間の詳細は、Red Hat Enterprise Linux Life Cycle および Red Hat Enterprise Linux Application Streams Life Cycle 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 で使用され、RHEL 9 で **削除された** 機能の詳細はRHEL 9 の導入における検討事項 を参照してください。

7.1. インストーラーおよびイメージの作成

非推奨の Kickstart 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。Kickstart 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。

(BZ#1899167)

7.2. セキュリティー

SHA-1 は暗号化の目的で非推奨になる

暗号化を目的とした SHA-1 メッセージダイジェストの使用は、RHEL 9 では非推奨になりました。SHA-1 によって生成されたダイジェストは、ハッシュ衝突の検出に基づく多くの攻撃の成功例が記録化

されているため、セキュアであるとは見なされません。RHEL コア暗号コンポーネントは、デフォルトで SHA-1 を使用して署名を作成しなくなりました。RHEL 9 のアプリケーションが更新され、セキュリティー関連のユースケースで SHA-1 が使用されないようになりました。

例外の中でも、HMAC-SHA1 メッセージ認証コードと Universal Unique Identifier (UUID) 値は、SHA-1 を使用して作成できます。これは、これらのユースケースが現在セキュリティーリスクをもたらさないためです。SHA-1 は、Kerberos や WPA-2 など、相互運用性および互換性に関する重要な懸念事項に関連する限られたケースでも使用できます。詳細は、RHEL 9 セキュリティーの強化ドキュメント の FIPS 140-3 に準拠していない暗号化を使用する RHEL アプリケーションのリスト を参照してください。

既存またはサードパーティーの暗号署名を検証するために SHA-1 を使用する必要がある場合は、次のコマンドを入力して有効にできます。

```
# update-crypto-policies --set DEFAULT:SHA1
```

または、システム全体の暗号化ポリシーを **LEGACY** ポリシーに切り替えることもできます。**LEGACY** は、セキュアではない他の多くのアルゴリズムも有効にすることに注意してください。

(JIRA:RHELPLAN-110763)

RHEL 9 で SCP が非推奨に

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

(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

(BZ#1995600)

OpenSSL が MD2、MD4、MDC2、Whirlpool、RIPEMD160、Blowfish、CAST、DES、IDEA、RC2、RC4、RC5、SEED、および PBKDF1 を非推奨化

OpenSSL プロジェクトは、安全でない、一般的でない、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。

以下のアルゴリズムの実装は、OpenSSL では従来のプロバイダーに移動しています。MD2、MD4、MD2、Whirlpool、RIPEMD160、Blowfish、CAST、DES、IDEA、RC2、RC4、RC5、SEED、および PBKDF1。

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。

(BZ#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。

(BZ#2034569)

fapolicyd.rules が非推奨に

実行ルールの許可と拒否を含むファイルの **/etc/fapolicyd/rules.d/** ディレクトリーは、**/etc/fapolicyd/fapolicyd.rules** ファイルを置き換えます。**fagenrules** スクリプトは、このディレクトリー内のすべてのコンポーネントルールファイルを **/etc/fapolicyd/compiled.rules** ファイルにマー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。

(BZ#2054740)

7.3. ネットワーク

ipset と iptables-nft を非推奨化

RHEL では、**ipset** パッケージおよび **iptables-nft** パッケージが非推奨になりました。**iptables-nft** には、**iptable**、**ip6tables**、**ebtables**、**arptables** などのさまざまなツールが同梱されています。このようなツールには新しい機能がなくなり、新しいデプロイメントに使用することは推奨されません。代わりに、**nftables** パッケージが提供する **nft** コマンドラインツールを使用することが推奨されます。既存の設定は、可能であれば **nft** に移行する必要があります。

iptable、**ip6tables**、**ebtables**、**arptables**、**nft_compat**、または **ipset** モジュールを読み込むと、**/var/log/messages** ファイルに以下の警告がログに記録されます。

> Warning: <module_name> - this driver is not recommended for new deployments. It continues to be supported in this RHEL release, but it is likely to be removed in the next major release. Driver updates and fixes will be limited to critical issues. Please contact Red Hat Support for additional information.

nftables への移行の詳細は、Migrating from iptables to nftables と、**iptables-translate(8)**および**ip6tables-translate(8)** の man ページを参照してください。

(BZ#1945151)

RHEL 9 でネットワークチームが非推奨に

teamd サービスおよび **libteam** ライブラリーは、Red Hat Enterprise Linux 9 では非推奨になり、次回のメジャーリリースでは削除される予定です。代替として、ネットワークチームの代わりにボンディングを設定します。

Red Hat は、機能が類似するボンディングとチームの機能を 2 つ管理しなくてもいいように、カーネルベースのボンディングに注力しています。ボンディングコードは、顧客の採用率が高く、堅牢で、活発なコミュニティ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。

(BZ#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。

(BZ#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。

(BZ#2089200)

7.4. カーネル

ATM カプセル化は RHEL 9 で非推奨になりました

非同期転送モード (ATM) カプセル化により、ATM アダプテーションレイヤー 5(AAL-5) のレイヤー 2(ポイントツーポイントプロトコル、イーサネット) またはレイヤー 3(IP) 接続が可能になります。Red Hat は、RHEL7 以降 ATMNIC ドライバーのサポートを提供していません。ATM 実装のサポートは RHEL 9 で廃止されています。これらのプロトコルは現在、ADSL テクノロジーをサポートし、メーカーによって段階的に廃止されているチップセットのみで使用されています。したがって、ATM カプセル化は Red Hat Enterprise Linux 9 では非推奨です。

詳細については、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。

(BZ#2058153)

v4l/dvb テレビおよびビデオキャプチャーデバイスはサポート対象外になりました

Red Hat は RHEL 9 で **Video4Linux** (**v4l**) および **Linux DVB** (**DVB**) はサポート対象外となりました。これらのデバイスは、さまざまなテレビチューナーカードとその他のビデオキャプチャーカードで構成されており、Red Hat では関連ドライバーを提供しなくなりました。

(BZ#2074598)

7.5. ファイルシステムおよびストレージ

lvm2-activation-generator およびその生成されたサービスが RHEL 9.0 で削除されました。

lvm2-activation-generator プログラムとその生成されたサービス **lvm2-activation**、**lvm2-activation-early**、および **lvm2-activation-net** は、RHEL 9.0 で削除されています。サービスをアクティベートするために使用される **lvm.conf event_activation** 設定は機能しなくなりました。ボリュームグループを自動アクティブ化する唯一の方法は、イベントベースのアクティブ化です。

(BZ#2038183)

7.6. 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。

(BZ#1927780, BZ#1974657, JIRA:RHELPLAN-80695)

7.7. IDENTITY MANAGEMENT

OpenDNSSec の SHA-1 を非推奨化

OpenDNSSec は、**SHA-1** アルゴリズムを使用したデジタル署名および認証レコードのエクスポートに対応しています。**SHA-1** アルゴリズムの使用に対応しなくなりました。RHEL 9 リリースでは、OpenDNSSec の **SHA-1** が非推奨になり、今後のマイナーリリースで削除される可能性があります。また、OpenDNSSec のサポートは、Red Hat Identity Management との統合に限定されます。OpenDNSSec はスタンドアロンでは対応していません。

(BZ#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)

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

7.8. グラフィックインフラストラクチャー

X.org Server が非推奨に

X.org ディスプレイサーバーは非推奨になり、今後の RHEL のメジャーリリースで削除される予定です。ほとんどの場合、デフォルトのデスクトップセッションは **Wayland** セッションになりました。

X11 プロトコルは、**XWayland** バックエンドを使用して完全にサポートされたままです。その結果、**X11** を必要とするアプリケーションは **Wayland** セッションで実行できます。

Red Hat は、**Wayland** セッションの残りの問題、改善点の解決に取り組んでいます。**Wayland** の未解決の問題については、既知の問題 セクションを参照してください。

ユーザーセッションは **X.org** バックエンドに戻すことができます。詳細は、GNOME 環境と表示プロトコルの選択 を参照してください。

(JIRA:RHELPLAN-121048)

Motif は非推奨になりました

アップストリームの Motif コミュニティーでの開発は非アクティブであるため、Motif ウィジェットツールキットは RHEL で非推奨になりました。

開発バリ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)

7.9. 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、 networking 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 コントローラーの **networking** RHEL システムロールを使用して RHEL 9 ノードに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。

(BZ#1999770)

7.10. 仮想化

SHA1 ベースの署名を使用した SecureBoot イメージ検証が非推奨に

UEFI (PE/COFF) 実行ファイルでの SHA1 ベースの署名を使用した SecureBoot イメージ検証の実行は非推奨になりました。代わりに、Red Hat は、SHA2 アルゴリズムまたはそれ以降に基づく署名を使用することを推奨します。

(BZ#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, BZ#1621944)

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)

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。

(BZ#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。

(BZ#1951814)

7.11. 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

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、RHBA-2022:5951 アドバイザリーから Podman 4.1.1 に更新する前に開始された rootless コンテナーは、(**slirp4netns** を使用するだけでなく) ネットワークに参加している場合は、アップグレード後に開始されたコンテナーに確実に接続できるように再起動する必要があります。

(BZ#2069279)

rhel9/pause が非推奨に

rhel9/pause コンテナーイメージが非推奨になりました。

(BZ#2106816)

7.12. 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
- python3-pytz

第8章 既知の問題

このパートでは、Red Hat EnterpriseLinux9.0 の既知の問題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

reboot --kexec コマンドおよび inst.kexec コマンドが、予測可能なシステム状態を提供しない

キックスタートコマンド **reboot --kexec** またはカーネル起動パラメーター **inst.kexec** で RHEL インストールを実行しても、システムの状態が完全な再起動と同じになるわけではありません。これにより、システムを再起動せずにインストール済みのシステムに切り替えると、予期しない結果が発生することがあります。

kexec 機能は非推奨になり、Red Hat Enterprise Linux の今後のリリースで削除されることに注意してください。

(BZ#1697896)

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。

(BZ#1877697)

キックスタートコマンドの auth および authconfig で AppStream リポジトリーが必要になる

インストール中に、キックスタートコマンドの **auth** および **authconfig** で **authselect-compat** パッケージが必要になります。**auth** または **authconfig** を使用したときに、このパッケージがないとインストールに失敗します。ただし、設計上、 **authselect-compat** パッケージは AppStream リポジトリーでしか利用できません。

この問題を回避するには、BaseOS リポジトリーおよび AppStream リポジトリーがインストーラーで利用できることを確認するか、インストール中にキックスタートコマンドの **authselect** コマンドを使用します。

(BZ#1640697)

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。

(BZ#2050140)

USB CD-ROM ドライブが Anaconda のインストールソースとして利用できない

USB CD-ROM ドライブがソースで、キックスタート **ignoredisk --only-use=** コマンドを指定すると、インストールに失敗します。この場合、Anaconda はこのソースディスクを見つけ、使用できません。

この問題を回避するには、**harddrive --partition=sdX --dir=/** コマンドを使用して USB CD-ROM ドライブからインストールします。その結果、インストールは失敗しなくなりました。

(BZ#1914955)

最小 RHEL インストールに、s390utils-base パッケージが含まれなくなる

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。この問題を回避するには、RHEL インストールの完了後に **s390utils-base** 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します。

(BZ#1932480)

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。

注記: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。

(BZ#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。

(BZ#2047713)

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。

(BZ#1983602)

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りました。

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。

(BZ#1997832)

PReP のサイズが 4 または 8 MiB でない場合、RHEL をインストールできません

PowerPC Reference Platform (PReP) パーティションのサイズが 4kiB セクターを使用するディスク上の 4MiB または 8MiB と異なる場合、RHEL インストーラーはブートローダーをインストールできません。その結果、RHEL をディスクにインストールすることはできません。

この問題を回避するには、PReP パーティションのサイズが正確に 4 MiB または 8 MiB であり、サイズが別の値に丸められていないことを確認してください。これにより、インストーラーはディスクに RHEL をインストールできるようになりました。

(BZ#2026579)

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カーネルの PowerNV IBM POWER システムが起動しなくなりました。

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 カーネルは **bigtime=1** および **inobtcount=1** 機能を XFS ファイルシステムに導入しますが、バージョン 5.10 よりも古いカーネルのファームウェアはこれを認識しません。これにより、Anaconda では、以下のエラーメッセージでインストールできなくなります。

ファームウェアは、**/boot** ファイルシステムの XFS ファイルシステム機能に対応していません。システムは起動できません。ファームウェアをアップグレードするか、ファイルシステムタイプを変更してください。

回避策として、**ext4** など、**/boot** に別のファイルシステムを使用します。

(BZ#2008792)

8.2. サブスクリプションの管理

FIPS モードでは virt-who が ESX サーバーに接続できない

FIPS モードで RHEL 9 システムで **virt-who** ユーティリティーを使用すると、**virt-who** は ESX サーバーに接続できません。したがって、**virt-who** は、設定されていても ESX サーバーを報告せず、以下のエラーメッセージをログに記録します。

```
ValueError: [digital envelope routines] unsupported
```

この問題を回避するには、以下のいずれかを実行します。

- **virt-who** の実行に使用する RHEL 9 システムを FIPS モードに設定しないでください。
- **virt-who** の実行に使用する RHEL システムをバージョン 9.0 にアップグレードしないでください。

(BZ#2054504)

8.3. 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。

(BZ#2073510)

8.4. シェルおよびコマンドラインツール

ifcfg ファイルを使用したネットワークインターフェイスの名前変更に失敗する

RHEL 9 では、**initscripts** はデフォルトでインストールされません。その結果、**ifcfg** ファイルを使用したネットワークインターフェイスの名前変更に失敗します。この問題を解決するには、**udev** ルールを使用するか、ファイルをリンクしてインターフェイスの名前を変更することが推奨されます。詳細は、一貫したネットワークインターフェイスデバイスの命名 および **systemd.link(5)** の man ページを参照してください。

推奨される方法のいずれも使用できない場合は、**initscripts** パッケージをインストールします。

(BZ#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。

(BZ#2053598)

8.5. 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。

(BZ#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。

(BZ#1984982)

コンソール keymap を設定するには、最小限のインストールで libxkbcommon ライブラリーが必要です。

RHEL 9 では、特定の **systemd** ライブラリーの依存関係が動的リンクから動的ロードに変換され、システムが実行時にライブラリーを開いて使用できるようになりました。今回の変更により、必要なライブラリーをインストールしない限り、このようなライブラリーに依存する機能は使用できなくなります。これは、最小限のインストール設定を使用するシステムにおけるキーボードレイアウトの設定にも影響します。その結果、**localectl --no-convert set-x11-keymap gb** コマンドに失敗します。

この問題を回避するには、**libxkbcommon** ライブラリーをインストールします。

```
# dnf install libxkbcommon
```

(BZ#2214130)

8.6. セキュリティー

OpenSSL は、PKCS #11 トークンが、生の RSA 署名または RSA-PSS 署名の作成に対応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と、**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** キーを処理できません。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。

(BZ#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。

(BZ#1685470)

FIPS で承認されていない暗号化は、FIPS モードの OpenSSL で機能します

FIPS で承認されていない暗号化は、システム設定に関係なく、OpenSSL ツールキットで機能します。したがって、システムが FIPS モードで実行されているときに無効にする必要がある暗号化アルゴリズムと暗号を使用できます。たとえば、次のようになります。

- RSA 鍵交換を使用する TLS 暗号スイートが機能します。
- 公開鍵の暗号化と復号化のための RSA ベースのアルゴリズムは、PKCS#1 と SSLv23 のパディングを使用したり、2048 ビットより短い鍵を使用したりしても機能します。

(BZ#2053289)

OpenSSL が FIPS モードでエンジンを使用できない

エンジン API は OpenSSL 3.0 で非推奨となり、OpenSSL Federal Information Processing Standards (FIPS) 実装およびその他の FIPS 互換実装と互換性がありません。そのため、OpenSSL は FIPS モードでエンジンを実行できません。この問題に対する回避策はありません。

(BZ#2087253)

PSK 暗号スイートは FUTURE 暗号ポリシーでは機能しません

事前共有キー (PSK) 暗号スイートは、完全転送秘密 (PFS) キー交換方式を実行しているとは認識されません。結果として、**ECDHE-PSK** および **DHE-PSK** 暗号スイートは、**SECLEVEL=3** に設定された

OpenSSL、たとえば **FUTURE** 暗号化ポリシーでは機能しません。回避策として、PSK 暗号スイートを使用するアプリケーションに対して、制限の少ない暗号化ポリシーを設定するか、セキュリティーレベル (**SECLEVEL**) を低く設定することができます。

(BZ#2060044)

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します

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。

(BZ#2070722)

一部の OpenSSH 操作では FIPS で承認されるインターフェイスが使用されていない

OpenSSH で使用される OpenSSL 暗号化ライブラリーは、レガシーとモダンの 2 つのインターフェイスを提供します。OpenSSL の内部構造が変更されたため、最新のインターフェイスのみが FIPS 認定の暗号化アルゴリズムの実装を使用しています。OpenSSH は一部の操作にレガシーインターフェイスを使用するため、FIPS 要件に準拠していません。

(BZ#2087121)

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。

(BZ#2073567)

SELinux staff_u ユーザーが unconfined_r に間違って切り替える可能性がある

secure_mode ブール値が有効になっていると、**staff_u** ユーザーが **unconfined_r** ロールに間違って切り替える可能性があります。これにより、**staff_u** ユーザーは、システムのセキュリティーに影響する特権操作を実行できました。

(BZ#2021529)

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。シナリオが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。

(BZ#2064274)

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)

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。

(BZ#2038978)

fagenrules --load が正しく動作しない

fapolicyd サービスは、信号のハングアップ (SIGHUP) を正しく処理しません。その結果、**fapolicyd** は SIGHUP シグナルを受信した後に終了します。したがって、**fagenrules --load** コマンドは適切に機能せず、ルールの更新には **fapolicyd** を手動で再起動する必要があります。この問題を回避するには、ルールの変更後に **fapolicyd** サービスを再起動すると、**fagenrules --load** が正常に機能します。

(BZ#2070655)

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
```

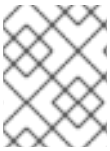
2. **/usr/share/scap-security-guide/ansible** ディレクトリーに移動します (# cd /usr/share/scap-security-guide/ansible)。

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。

(BZ#2105162)

8.7. ネットワーク

nm-cloud-setup サービスは、手動で設定されたセカンダリー IP アドレスをインターフェイスから削除する

クラウド環境から受け取った情報に基づいて、**nm-cloud-setup** サービスがネットワークインターフェイスを設定します。インターフェイスを手動で設定するには、**nm-cloud-setup** を無効にします。ただし、場合によっては、ホスト上の他のサービスもインターフェイスを設定できます。たとえば、これらのサービスはセカンダリー IP アドレスを追加できます。**nm-cloud-setup** がセカンダリー IP アドレスを削除しないようにするには、

1. **nm-cloud-setup** サービスおよびタイマーを停止して無効にします。

```
# systemctl disable --now nm-cloud-setup.service nm-cloud-setup.timer
```

2. 使用可能な接続プロファイルを表示します。

```
# nmcli connection show
```

3. 影響を受ける接続プロファイルを再アクティブ化します。

```
# nmcli connection up "<profile_name>"
```

その結果、このサービスは、手動で設定されたセカンダリー IP アドレスをインターフェイスから削除しなくなりました。

(BZ#2151040)

カーネルコマンドラインに空の rd.znet オプションがあると、ネットワークコンフィギュレーションに失敗します。

ネットタイプやサブチャネルなど、引数を指定しないと、カーネルの **rd.znet** オプションがネットワークの設定に失敗します。この問題を回避するには、コマンドラインから **rd.znet** オプションを完全に削除するか、関連するネットタイプ、サブチャンネル、およびその他の関連オプションを指定します。これらのオプションの詳細は、man ページの **dracut.cmdline(7)** を参照してください。

(BZ#1931284)

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。

(BZ#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。

(BZ#2082303)

インスタンスのプライマリー IP アドレスは、Alibaba Cloud で nm-cloud-setup サービスを開始した後に変更されます

Alibaba Cloud でインスタンスを起動した後、**nm-cloud-setup** サービスはプライマリー IP アドレスをインスタンスに割り当てます。ただし、インスタンスに複数のセカンダリー IP アドレスをインスタンスに割り当てて **nm-cloud-setup** サービスを開始すると、以前のプライマリー IP アドレスはすでに割り当てられているセカンダリー IP アドレスの 1 つに置き換えられます。返されたメタデータのリストは、同じことを確認します。この問題を回避するには、プライマリー IP アドレスが変更されないように、セカンダリー IP アドレスを手動で設定します。その結果、インスタンスは両方の IP アドレスを保持し、プライマリー IP アドレスは変更されません。

(BZ#2079849)

8.8. カーネル

RHEL 9 カーネルで kdump サービスが起動しない

RHEL 9 カーネルには、**crashkernel=auto** パラメーターがデフォルトとして設定されていません。そのため、デフォルトでは **kdump** が起動できません。

この問題を回避するには、**crashkernel=** オプションを必要な値に設定します。

たとえば、**grubby** ユーティリティーを使用して 256 MB のメモリーを確保するには、以下のコマンドを入力します。

```
# grubby --args crashkernel=256M --update-kernel ALL
```

その結果、RHEL 9 カーネルは **kdump** を開始し、設定されたメモリーサイズの値を使用して **vmcore** ファイルをダンプします。

(BZ#1894783)

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。

(BZ#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、必要なメモリーが使用可能なメモリーサイズを超えているため、**kdumpctl** コマンドは **kdump** サービスの開始に失敗します。

回避策として、**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**)。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。

(BZ#2065013)

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。

(BZ#2000616)

Secure Boot で fadump を有効にすると、GRUB Out of Memory (OOM) が発生する可能性があります。

Secure Boot 環境では、GRUB と PowerVM は、ブートメモリー用に、RMA (Real Mode Area)と呼ばれる 512 MB のメモリー領域を割り当てます。リージョンはブートコンポーネントに分割され、いずれかのコンポーネントが割り当てを超えると、メモリー不足の問題が発生します。

通常、デフォルトでインストールされている **initramfs** ファイルシステムと **vmlinux** シンボルテーブルは、このような障害を回避するために割り当て制限内に抑えられています。ただし、システムで Firmware Assisted Dump (FADump) が有効になっている場合は、デフォルトの **initramfs** サイズが増加して 95 MB を超える可能性があります。これにより、システムを再起動するたびに GRUB OOM 状態になります。

この問題を回避するには、Secure Boot と FADump を一緒に使用しないでください。この問題の回避方法は、link:https://www.ibm.com/support/pages/node/6846531 を参照してください。

(BZ#2149172)

Secure Boot のシステムは、動的 LPAR 操作を実行できません。

以下の条件のいずれかが満たされた場合、ユーザーはハードウェア管理コンソール (HMC) から動的論理パーティション (DLPAR) 操作を実行できません。

- Secure Boot 機能は、整合性モードでカーネル **ロックダウン** メカニズムを暗黙的にオンにする機能が有効になっている。
- カーネル **ロックダウン** メカニズムが、整合性モードまたは機密性モードで手動で有効にされている。

RHEL 9 では、カーネルの **lockdown** により、**/dev/mem** 文字デバイスファイルからアクセスできるシステムメモリーへの RunTimeAbstraction Services (RTAS) アクセスが完全にブロックされています。正しく機能させるには、いくつかの RTAS 呼び出しで **/dev/mem** への書き込みアクセスが必要です。したがって、RTAS 呼び出しが正しく実行されず、ユーザーには以下のエラーメッセージが表示されます。

> HSCL2957 Either there is currently no RMC connection between the management console and the partition <LPAR name> or the partition does not support dynamic partitioning operations. Verify the network setup on the management console and the partition and ensure that any firewall authentication between the management console and the partition has occurred. Run the management console diagrmc command to identify problems that might be causing no RMC connection.

(BZ#2083106)

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)

8.9. ブートローダー

新しいカーネルが以前のコマンドラインオプションを失います。

GRUB ブートローダーは、以前に設定されたカスタムのカーネルコマンドラインオプションを新しいカーネルに適用しません。したがって、カーネルパッケージをアップグレードすると、オプションが不足しているため、再起動後にシステムの動作が変わる可能性があります。

この問題を回避するには、カーネルをアップグレードするたびに、すべてのカスタムカーネルコマンドラインオプションを手動で追加します。その結果、カーネルは、次のカーネルアップグレードまで、期待どおりにカスタムオプションを適用します。

(BZ#1969362)

8.10. ファイルシステムおよびストレージ

デバイスマッパーマルチパスは NVMe/TCP ではサポートされていません

nvme-tcp ドライバーで Device Mapper Multipath を使用すると、コールトレースの警告とシステムの不安定性が発生する可能性があります。この問題を回避するには、NVMe/TCP ユーザーはネイティブ NVMe マルチパスを有効にする必要があり、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。

(BZ#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します

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。

(BZ#2011699)

supported_speeds の無効な sysfs 値

qla2xxx ドライバーは、**sys fssupported_speeds** 属性で サポート されているポート速度の 1 つとして、予想される 64Gb/s ではなく 20Gb/s を報告します。

```
$ cat /sys/class/fc_host/host12/supported_speeds
16 Gbit, 32 Gbit, 20 Gbit
```

これにより、HBA が 64Gb/s リンク速度に対応している場合は、**sysfs supported_speeds** 値が正しくありません。これは **sysfs** の **supported_speeds** 値にのみ影響し、ポートは予想されるネゴシエートされたリンクレートで動作します。

(BZ#2069758)

AMD EPYC システムの Broadcom イニシエーターから NVMe 名前空間に接続できない

デフォルトでは、RHEL カーネルは AMD ベースのプラットフォームで IOMMU を有効にします。その結果、AMD プロセッサーを搭載したサーバーで IOMMU 対応プラットフォームを使用すると、転送長の不一致が原因で I/O が失敗するなどの NVMeI/O の問題が発生する可能性があります。

この問題を回避するには、カーネルコマンドラインオプション **iommu=pt** を使用して、パススルーモードで IOMMU を追加します。その結果、AMDEPYC システムの Broadcom イニシエーターから NVMe 名前空間に接続できるようになりました。

(BZ#2073541)

8.11. 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

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。

(BZ#1991500)

8.12. 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。

(BZ#2083727)

8.13. IDENTITY MANAGEMENT

RHEL 9 クライアントが Heimdal KDC に対して PKINIT を使用してユーザーを認証できません

RHEL 9 Kerberos クライアントでの IdM ユーザーの PKINIT 認証中に、Kerberos クライアントが **supportedCMSTypes** フィールドをサポートしていないため、RHEL 9 以前の Heimdal Kerberos Distribution Center (KDC) は SHA-1 バックアップ署名アルゴリズムを使用します。ただし、SHA-1 アルゴリズムは RHEL 9 で非推奨になっているため、ユーザー認証は失敗します。

この問題を回避するには、次のコマンドを使用して、RHEL 9 クライアントで 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
```

その結果、PKINIT 認証は Kerberos クライアントと Heimdal KDC の間で機能します。

サポートされているバックアップ署名アルゴリズムの詳細は、CMS アルゴリズム識別子に対して定義された Kerberos 暗号化タイプ を参照してください。

RHEL 9 Kerberos エージェントが RHEL 9 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 も併せて参照してください。

(BZ#2068935)

RHEL 9 Kerberos エージェントが RHEL 9 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

RHEL 9 の Kerberos エージェントが環境内の別の RHEL 9 Kerberos エージェントと相互作用すると、ユーザーの初期認証 (PKINIT) 認証の公開鍵暗号化に失敗します。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。このため、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

パッチが適用されていないエージェントが Kerberos クライアントか Kerberos Distribution Center (KDC) であるかに関わらず、これらのアクションのいずれかを実行する必要があります。

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 が古い RHEL KDC および AD KDC に対して機能するには、RHEL 9 クライアントで DEFAULT:SHA1 サブポリシーを設定する必要があります。 も併せて参照してください。

(BZ#2077450)

PKINIT が古い RHEL KDC および AD KDC に対して機能するには、RHEL 9 クライアントで DEFAULT:SHA1 サブポリシーを設定する必要があります。

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。

RHEL 7.9 および RHEL 8.7 以降では、SHA-256 がデフォルトで使用されますが、RHEL 7.8 および RHEL 8.6 の古い Kerberos Key Distribution Centers (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 メッセージに署名するために使用されます。Active Directory (AD) KDC も同様です。

その結果、RHEL 9 Kerberos クライアントは、以下に対して PKINIT を使用してユーザーを認証できません。

- RHEL 7.8 以前で実行されている KDC
- RHEL 8.6 以前で実行されている KDC
- AD KDC

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
```

RHEL 9 クライアントが Heimdal KDC に対して PKINIT を使用してユーザーを認証できません も併せて参照してください。

(BZ#2060798)

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
```

(BZ#2057471)

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。

(BZ#2053204)

Directory Server で接尾辞の referral 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。

(BZ#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ありません。

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。

(BZ#2047175)

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)

8.14. 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。

(BZ#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。

(BZ#2060308)

8.15. 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しない

以下のシステム設定を使用すると、ディスプレイにグラフィカル出力が表示されない場合があります。

- 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することはできません。

この問題を回避するには、以下の手順に従います。

1. ブートローダーメニューにシステムを起動します。
2. **module_blacklist=mgag200** オプションをカーネルコマンドラインに追加します。

これにより、RHEL が起動し、予想どおりにグラフィカル出力が表示されますが、最大解像度は 16 ビットの色深度で 1024x768 に制限されます。

(BZ#1960467)

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)

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 では利用できません

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)

8.16. WEB コンソール

Web コンソールを使用した USB ホストデバイスの削除が期待どおりに機能しない

USB デバイスを仮想マシン (VM) に接続すると、USB デバイスのデバイス番号とバス番号が VM に渡された後に変更される場合があります。結果として、Web コンソールを使用してそのようなデバイスを削除すると、デバイスとバス番号の相関が正しくないために失敗します。この問題を回避するには、VM の XML 設定から USB デバイスの **<hostdev>** 部分を削除します。

(JIRA:RHELPLAN-109067)

Web コンソールを使用した複数のホストデバイスの接続は機能しない

Web コンソールを使用して仮想マシン (VM) に接続する複数のデバイスを選択すると、1 つのデバイスのみが接続され、残りは無視されます。この問題を回避するには、一度に 1 つのデバイスのみを接続します。

(JIRA:RHELPLAN-115603)

8.17. 仮想化

https 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ります。

現在、**virt-install** ユーティリティーは、https 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。

この問題を回避するには、**qemu-kvm-block-curl** をホストにインストールして、https プロトコルサポートを有効にします。別の接続プロトコルまたは別のインストールソースを使用することもできます。

(BZ#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ります。

一部の AMD Milan システムでは、Enhanced REP MOVSB (**erms**) および Fast Short REP MOVSB (**fsrm**) 機能フラグがデフォルトで BIOS で無効になっています。したがって、Milan CPU タイプは、こ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。

(BZ#2077767)

仮想マシンのネットワークトラフィックのパフォーマンスが低下する可能性があります

場合によっては、RHEL 9.0 ゲスト仮想マシン (VM) は、高レベルのネットワークトラフィックを処理するときにパフォーマンスをいくらか低下させます。

(BZ#1945040)

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。

(BZ#2005173)

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。

(BZ#1969724)

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ません

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。

(BZ#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。

(BZ#1817965, BZ#1789206)

8.18. クラウド環境の RHEL

SR-IOV は、Azure 上の ARM 64 RHEL 9 仮想マシンで最適に動作しません

現在、SR-IOV ネットワーキングデバイスは、Microsoft Azure プラットフォームで実行されている ARM 64 RHEL 9 仮想マシンで想定されるよりも、全体でははるかに低くなっており、レイテンシーは高くなっています。

(BZ#2068432)

コンソールプロキシーを使用する XenServer 7 上の RHEL 9 仮想マシンでは、マウスを使用できません。

コンソールプロキシーを使用して XenServer 7 プラットフォームで RHEL 9 仮想マシンを実行している場合は、仮想マシンの GUI でマウスを使用することはできません。この問題を回避するには、次のように仮想マシンで Wayland コンポジタープロトコルを無効にします。

1. **/etc/gdm/custom.conf** を開きます。
2. **WaylandEnable=false** 行のコメントを解除します。
3. ファイルを保存します。

また、Red Hat は、RHEL 仮想マシンを実行するプラットフォームとして XenServer に対応していないため、実稼働環境での RHEL での XenServer の使用を推奨しません。

(BZ#2019593)

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ります

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。

(BZ#2059545)

Hyper-V 仮想マシンに接続されたネットワークアダプターの SR-IOV 機能が機能しない場合があります

現在、シングルルート I/O 仮想化 (SR-IOV) が有効になっているネットワークアダプターを MicrosoftHyper-V ハイパーバイザーで実行されている RHEL 9 仮想マシン (VM) に接続すると、SR-IOV 機能が正しく機能しない場合があります。

この問題を回避するには、仮想マシン設定で SR-IOV を無効にしてから、再度有効にします。

1. Hyper-V Manager ウィンドウで、仮想マシンを右クリックします。
2. コンテキストメニューで、**Settings/Network Adapter/Hardware Acceleration** に移動します。
3. **Enable SR-IOV** の選択を解除します。
4. **Apply** をクリックします。
5. 手順 1 と 2 を繰り返して、**SR-IOV** を有効にするオプションに再度移動します。
6. **Enable SR-IOV** をオンにします。
7. **Apply** をクリックします。

(BZ#2030922)

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ります

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。

(BZ#2037657)

8.19. 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
```

(BZ#1869561)

8.20. コンテナー

ベータ版 GPG キーで署名されたコンテナーイメージがプルできない

現在、RHEL 9 Beta コンテナーイメージをプルしようとすると、**podman** が終了し、エラーメッセージ**Error:Source image rejected:None of the signatures were accepted** が表示されます。現在のビルドでは、RHEL ベータ版の GPG キーをデフォルトで信頼しないように設定されているため、イメージのプルに失敗します。

回避策としては、Red Hat Beta GPG キーがローカルシステムに保存されていることを確認し、**podman image trust set**コマンドで適切な beta 名前空間の既存の信頼範囲を更新します。

ベータ版の GPG キーがローカルに保存されていない場合は、以下のコマンドを実行することで、そのキーをプルすることができます。

```
sudo wget -O /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-beta
https://www.redhat.com/security/data/f21541eb.txt
```

Beta GPG キーを信頼済みとしてネームスペースに追加するには、次のいずれかのコマンドを使用します。

```
$ sudo podman image trust set -f /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-beta
registry.access.redhat.com/namespace
```

および

```
$ sudo podman image trust set -f /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-beta
registry.redhat.io/namespace
```

namespaceを**ubi9-beta**または**rhel9-beta**に置き換えてください。

(BZ#2020026)

"X509: certificate signed by unknown authority" のエラーで Podman はコンテナーをプルできない

独自の CA 証明書で署名された独自の内部レジストリーがある場合は、証明書をホストマシンにインポートする必要があります。そうでない場合は、エラーが発生します。

```
x509: certificate signed by unknown authority
```

ホストで CA 証明書をインポートします。

```
# cd /etc/pki/ca-trust/source/anchors/
[anchors]# curl -O <your_certificate>.crt

[anchors]# update-ca-trust
```

これで、内部レジストリーからコンテナーイメージをプルできます。

(BZ#2027576)

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)

podman system connection add および podman image scp が失敗する

Podman は、RSA 鍵交換に SHA-1 ハッシュを使用します。RHEL 9 では SHA-1 ハッシュが鍵交換に受け入れられないため、RSA キーを使用するマシン間の通常の SSH 接続は機能します。ただし、**podman system connection add** および **podman image scp** コマンドは同じ RSA 鍵を使用して機能しません。

```
$ podman system connection add --identity ~/.ssh/id_rsa test_connection
$REMOTE_SSH_MACHINE
Error: failed to connect: ssh: handshake failed: ssh: unable to authenticate, attempted methods [none
publickey], no supported methods remain
```

この問題を回避するには、ED25519 鍵を使用します。

1. リモートマシンに接続します。

```
$ ssh -i ~/.ssh/id_ed25519 $REMOTE_SSH_MACHINE
```

2. Podman サービスの ssh 宛先を記録します。

```
$ podman system connection add --identity ~/.ssh/id_ed25519 test_connection
$REMOTE_SSH_MACHINE
```

3. ssh の宛先が記録されたことを確認します。

```
$ podman system connection list
```

RHBA-2022:5951 アドバイザリーリリースでは、この問題が修正されています。

(JIRA:RHELPLAN-121180)

付録A コンポーネント別のチケットリスト

本書には Bugzilla と JIRA ID が記載されています。一般にアクセス可能な Bugzilla バグには、チケットへのリンクが含まれます。

コンポーネント	チケット
389-ds-base	BZ#2024693, BZ#1805717, BZ#1779685, BZ#2053204, BZ#2063140, BZ#2047175
ModemManager	BZ#1996716
NetworkManager	BZ#1980387, BZ#1949127, BZ#2060013, BZ#1931284, BZ#1894877, BZ#2079849
RHCOS	BZ#2008521
WALinuxAgent	BZ#1972101
alsa-lib	BZ#2015863
anaconda	BZ#1951709, BZ#1978264, BZ#2025953, BZ#2009403, BZ#2050140, BZ#1877697, BZ#1914955, BZ#1929105, BZ#1983602, BZ#1997832, BZ#2008792
ansible-collection-microsoft-sql	BZ#2064648, BZ#2064690
ansible-collection-redhat-rhel_mgmt	BZ#2023381
ansible-pcp	BZ#1957566
bash	BZ#2079078
bind	BZ#1984982
binutils	BZ#2030554
boost	BZ#1957950
chrony	BZ#1961131
clevis	BZ#1956760
cloud-init	BZ#2040090, BZ#2042351
cmake	BZ#1957948

コンポーネント	チケット
container-tools	BZ#2000871
containers-common	BZ#2019901
crash	BZ#1896647
createrepo_c	BZ#2055032
crypto-policies	BZ#2004207, BZ#2013195
cyrus-sasl	BZ#1947971, BZ#1995600
device-mapper-multipath	BZ#2017979, BZ#2017592, BZ#2011699
distribution	BZ#1878583
dnf	BZ#2005305, BZ#2073510
dotnet6.0	BZ#1986211
edk2	BZ#1935497
eigen3	BZ#2032423
fapolicyd	BZ#2032408, BZ#1932225, BZ#2054740, BZ#2070655
fence-agents	BZ#1977588
fetchmail	BZ#1999276
fido-device-onboard	BZ#1989930
firefox	BZ#1764205, BZ#2013247
firewalld	BZ#2029211
freeradius	BZ#1978216
gcc	BZ#1986836, BZ#1481850
gdb	BZ#1870029, BZ#1870031
gfs2-utils	BZ#1616432
gimp	BZ#2047161

コンポーネント	チケット
git	BZ#1956345
glibc	BZ#2023422, BZ#2024347
gnome-shell-extension-background-logo	BZ#2057150
gnome-shell-extensions	BZ#2031186
gnupg2	BZ#2070722, BZ#2073567
gnutls	BZ#2033220、BZ#1999639
golang	BZ#2014087, BZ#1984110
grafana-pcp	BZ#1993156, BZ#1845592
grafana	BZ#1993215
grub2	BZ#2026579
grubby	BZ#1969362
hostapd	BZ#2019830
ipa	BZ#1952028, BZ#1957736, BZ#1966101, BZ#1988383, BZ#2084180, BZ#2084166, BZ#2057471
iptables	BZ#1945151
javapackages-tools	BZ#1951482
jigawatts	BZ#1972029
jmc-core	BZ#1980981
kdump-anaconda-addon	BZ#1894783、BZ#2017401
kernel-rt	BZ#2002474

コンポーネント	チケット
kernel	BZ#1844416, BZ#1851933, BZ#1780258, BZ#1874195, BZ#1953515, BZ#1960556, BZ#1948340, BZ#1952863, BZ#1978382, BZ#1957818, BZ#2002499, BZ#2050415, BZ#1951951, BZ#1949613, BZ#2036856, BZ#2034490, BZ#1943423, BZ#2054441, BZ#2046472, BZ#2068432, BZ#1997541, BZ#1613522, BZ#1874182, BZ#1995338, BZ#1570255, BZ#2023416, BZ#2021672, BZ#2019593, BZ#2000616, BZ#2013650, BZ#2033080, BZ#2069758, BZ#2059545, BZ#2030922, BZ#1945040, BZ#2073541, BZ#1960467, BZ#2005173
kexec-tools	BZ#1988894, BZ#1895232, BZ#1958452, BZ#2065013
kmod	BZ#1985100
krb5	BZ#2060798, BZ#2068935, BZ#2077450
libburn	BZ#2015861
libcap	BZ#2037215
libgcrypt	BZ#1990059
libmodulemd	BZ#1984403
libreswan	BZ#2017355, BZ#2039877
libseccomp	BZ#2019887
libservicelog	BZ#1869568
libvirt	BZ#2014487
libxcrypt	BZ#2034569
llvm-toolset	BZ#2001107
lorax-templates-rhel	BZ#1961092
lsvpd	BZ#1869564
lvm2	BZ#1899214, BZ#1749513, BZ#2038183
mariadb	BZ#1971248
mod_security_crs	BZ#1947962

コンポーネント	チケット
nettle	BZ#1986712
nfs-utils	BZ#2059245
nginx	BZ#1953639
nmstate	BZ#1969941
nodejs	BZ#1953491
nss	BZ#2008320, BZ#2099438
numatop	BZ#1874125
nvml	BZ#1874208
opal-prd	BZ#1869560
open-vm-tools	BZ#2037657
opencryptoki	BZ#1869533
openscap	BZ#2041782
openssh	BZ#1952957, BZ#2002734, BZ#1821501, BZ#2087121
openssl	BZ#1990814, BZ#1871147, BZ#1970388, BZ#1975836, BZ#1681178, BZ#1685470, BZ#2053289, BZ#2087253, BZ#2060044, BZ#2071631
osbuild-composer	BZ#2060575
oscap-anaconda-addon	BZ#1893753
ostree	BZ#1961254
p11-kit	BZ#1966680
pacemaker	BZ#1850145, BZ#1443666, BZ#1470834, BZ#1082146, BZ#1376538, BZ#1975388
pcp	BZ#1991764, BZ#1847808, BZ#1981223
pcs	BZ#1290830, BZ#1909901, BZ#1872378, BZ#2018969, BZ#1996067

コンポーネント	チケット
perl-Module-Signature	BZ#2039361
php	BZ#1949319
pki-core	BZ#2084181
podman	JIRA:RHELPLAN-77549, JIRA:RHELPLAN-75322, JIRA:RHELPLAN-108830, BZ#2027576
powerpc-utils	BZ#1873868
ppc64-diag	BZ#1869567
python-jsonpointer	BZ#1980256
python-podman	BZ#1975462
qemu-kvm	BZ#1940132, BZ#1939509, JIRA:RHELPLAN-75866, BZ#1874187, BZ#1965079, BZ#1951814, BZ#2014229, BZ#2052424, BZ#1817965
redis	BZ#1959756
rhel-system-roles	BZ#1993304, BZ#1993377, BZ#2022461, BZ#1978488, BZ#1984583, BZ#2016517, BZ#2021667, BZ#1986460, BZ#1978752, BZ#1978753, BZ#1990490, BZ#2031555, BZ#2016518, BZ#2054364, BZ#1978773, BZ#2054435, BZ#1999162, BZ#2057657, BZ#2012298, BZ#2021028, BZ#2054367, BZ#2054369, BZ#2057662, BZ#2021665, BZ#2029427, BZ#2004899, BZ#1958964, BZ#1978734, BZ#1978760, BZ#2039106, BZ#2041632, BZ#2058777, BZ#2058645, BZ#2058756, BZ#2071804, BZ#2029634, BZ#2044408, BZ#2029602, BZ#2038957, BZ#2064391, BZ#2004303, BZ#2006230, BZ#2057164, BZ#2021025, BZ#2021676, BZ#2047506, BZ#2050341, BZ#2050419, BZ#1999770
rpm-ostree	BZ#1961324
rpm	BZ#1942549、BZ#1962234
rsyslog	BZ#2027971, BZ#1992155
rust-toolset	BZ#2002885
s390utils	BZ#1932480
samba	BZ#2013578、Jira:RHELDOCS-16612
scap-security-guide	BZ#2028435, BZ#2014561, BZ#2045341, BZ#2038978

コンポーネント	チケット
selinux-policy	BZ#2055822, BZ#1932752, BZ#2021529, BZ#2064274
shadow-utils	BZ#1859252
sos	BZ#2011537, BZ#1869561
squid	BZ#1990517
sssd	BZ#1949149, BZ#2014249, BZ#1879869, BZ#1737489
strace	BZ#2038965
stratisd	BZ#2041558
stunnel	BZ#2039299
subscription-manager	BZ#1898563, BZ#2049441
sudo	BZ#1981278
swig	BZ#1943580
systemd	BZ#2018112
systemtap	BZ#2083727
tigervnc	BZ#2060308
trace-cmd	BZ#1933980
tuned	BZ#2003838
unbound	BZ#2070495
usbguard	BZ#1986785, BZ#2009226
varnish	BZ#1984185
virt-manager	BZ#1995131
virt-who	BZ#2008215, BZ#2054504
virtio-win	BZ#1969724
wpa_supplicant	BZ#1975718

コンポーネント	チケット
その他	BZ#2077836, BZ#2019806, BZ#1937651,BZ#2010291, BZ#1941810, BZ#2091643, BZ#1941595, JIRA:RHELPLAN-80758, JIRA:RHELPLAN-80759, JIRA:RHELPLAN-82578, JIRA:RHELPLAN-68364, JIRA:RHELPLAN-78673, JIRA:RHELPLAN-78675, BZ#1940863, BZ#2079313, JIRA:RHELPLAN-100497, BZ#2068532,BZ#2089193, JIRA:RHELPLAN-102009, BZ#2065646, BZ#2088414, JIRA:RHELPLAN-80734, BZ#2013853, JIRA:RHELPLAN-103540, BZ#2019341, BZ#2008558, BZ#2008575, BZ#2009455, JIRA:RHELPLAN-74542, JIRA:RHELPLAN-73678, JIRA:RHELPLAN-84168, JIRA:RHELPLAN-73697, JIRA:RHELPLAN-95126, BZ#2080875, JIRA:RHELPLAN-97899, JIRA:RHELPLAN-100359, JIRA:RHELPLAN-103147, JIRA:RHELPLAN-103146, JIRA:RHELPLAN-79161, BZ#2046325, BZ#2021262, JIRA:RHELPLAN-64576, JIRA:RHELPLAN-65223, BZ#2083036, BZ#2011448, BZ#2019318, JIRA:RHELPLAN-101240, JIRA:RHELPLAN-101241, JIRA:RHELPLAN-101242, JIRA:RHELPLAN-101246, JIRA:RHELPLAN-101247, JIRA:RHELPLAN-102552, JIRA:RHELPLAN-99892, BZ#2027596, JIRA:RHELPLAN-119000, BZ#1940653, JIRA:RHELPLAN-95056, BZ#2054401, JIRA:RHELPLAN-113994, BZ#2059183, JIRA:RHELPLAN-74543, JIRA:RHELPLAN-99889, JIRA:RHELPLAN-99890, JIRA:RHELPLAN-100032, JIRA:RHELPLAN-100034, JIRA:RHELPLAN-101141, JIRA:RHELPLAN-100020, BZ#2069501, BZ#2070506, JIRA:RHELPLAN-117903, JIRA:RHELPLAN-98617, JIRA:RHELPLAN-103855, BZ#2091653, BZ#2082306, JIRA:RHELPLAN-65217, BZ#2020529, BZ#2030412, BZ#2046653, JIRA:RHELPLAN-103993, JIRA:RHELPLAN-122345, BZ#1927780, JIRA:RHELPLAN-110763, BZ#1935544, BZ#2089200, JIRA:RHELPLAN-15509, JIRA:RHELPLAN-99136, JIRA:RHELPLAN-103232, BZ#1899167, BZ#1979521, JIRA:RHELPLAN-100087, JIRA:RHELPLAN-100639, JIRA:RHELPLAN-10304, BZ#2058153, JIRA:RHELPLAN-113995, JIRA:RHELPLAN-121048, JIRA:RHELPLAN-98983, BZ#1640697, BZ#1697896, BZ#2020026, BZ#2047713, JIRA:RHELPLAN-109067, JIRA:RHELPLAN-115603, JIRA:RHELPLAN-96940, JIRA:RHELPLAN-117234, JIRA:RHELPLAN-119001, JIRA:RHELPLAN-119852, BZ#2077767, BZ#2053598, JIRA:RHELPLAN-121180, BZ#2082303, JIRA:RHELPLAN-121049

付録B 謝辞

RHEL 9 Readiness Challenge の一環としてフィードバックを提供している、以下の Red Hat Associates にご連絡ください。

- Buland Singh
- Pradeep Jagtap
- Omkar Andhekar
- Ju Ke
- Suresh Jagtap
- Prijesh Patel
- Nikhil Suryawanshi
- Amit Yadav
- Pranav Lawate
- John Pittman

付録C 改訂履歴

0.1-28

Thu Mar 14 2024, Gabriela Fialovpropagate (gfialova@redhat.com)

- 既知の問題 JIRA:RHEL-25967 （カーネル）を追加しました。

0.1-27

2024 年 2 月 14 日（水）Gabriela Fialovpropagate (gfialova@redhat.com)

- 拡張 JIRA:RHELDOCS-17553 (Identity Management)を追加しました。

0.1-26

2024 年 2 月 1 日木曜日、Gabi Fialova (gfialova@redhat.com)

- KI BZ#1834716 (セキュリティー) を追加しました

0.1-25

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。

0.1-24

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。

0.1-23

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。

0.1-22

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。

0.1-21

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。
- JIRA の RHEL を反映して「Red Hat ドキュメントへのフィードバック」を更新しました。

0.1-20

2023 年 8 月 17 日、Gabi Fialova (gfialova@redhat.com)

- Enh BZ#2136937 (Plumbers) を追加しました。

0.1-19

2023 年 8 月 7 日、Gabi Fialova (gfialova@redhat.com)

- KI BZ#2214130 (CS)を追加しました

- KI BZ#2214130 (CS)を追加しました。

0.1-18

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。

0.1-17

2023 年 6 月 19 日 (月) Gabi Fialova (gfialova@redhat.com)

- KI BZ#2068935 (IdM) のタイプミスを修正しました。

0.1-16

2023 年 5 月 18 日 (木) Gabi Fialova (gfialova@redhat.com)

- 機能拡張 BZ#2053642 (ファイルシステムとストレージ) を追加

0.1-15

2023 年 5 月 17 日 (水)、Gabi Fialova (gfialova@redhat.com)

- EOL に関する情報を追加して deprecated-packages.adoc を更新

0.1-14

2023 年 5 月 11 日 (木) Gabi Fialova (gfialova@redhat.com)

- 機能拡張 BZ#2190045 (インストーラー) を追加

0.1-13

2023 年 4 月 27 日 (木)、Gabi Fialova (gfialova@redhat.com)

- 既知の問題 JIRA:RHELPLAN-155168 (アイデンティティー管理) を追加

0.1-12

2023 年 4 月 13 日 (木)、Gabi Fialova (gfialova@redhat.com)

- 新機能 JIRA:RHELPLAN-84168 (コンテナー) の壊れたリンクを修正しています。

0.1-11

2023 年 3 月 1 日水曜日、Gabi Fialova (gfialova@redhat.com)

- BZ#2091643 (カーネル) のドキュメントテキストを変更しました。

0.1-10

2023 年 2 月 20 日 (月) Gabi Fialova (gfialova@redhat.com)

- SAP 環境に関する情報を「RHEL 8 から RHEL 9 へのインプレースアップグレード」に追加

0.1-9

2023 年 1 月 18 日 (水) Gabi Fialova (gfialova@redhat.com)

- 既知の問題ドキュメントテキスト BZ#2083106 (カーネル) を追加

0.1-8

2023 年 1 月 17 日 (火) Gabi Fialova (gfialova@redhat.com)

- テクノロジープレビューのドキュメントテキスト BZ#2084181 (アイデンティティー管理) を更新

0.1-7

2023 年 1 月 16 日 (月) Gabi Fialova (gfialova@redhat.com)

- 既知の問題ドキュメントテキストBZ#2149172 (カーネル) を追加

0.1-6

2022 年 12 月 22 日 (木) Gabi Fialova (gfialova@redhat.com)

- 既知の問題のドキュメントテキスト BZ#1960467 (グラフィックスインフラストラクチャー) を更新。

0.1-5

2022 年 12 月 8 日 (木) Marc Muehlfeld (mmuehlfeld@redhat.com)

- 既知の問題 BZ#2151040 (ネットワーキング) を追加。

0.1-4

2022 年 11 月 15 日 (火) Gabriela Fialová (gfialova@redhat.com)

- インプレースアップグレード セクションを更新

0.1-3

2022 年 9 月 23 日 (金) Gabriela Fialová (gfialova@redhat.com)

- 非推奨機能 BZ#2074598 (カーネル) を追加

0.1-2

2022 年 9 月 21 日 (水) Gabriela Fialová (gfialova@redhat.com)

- 既知の問題 BZ#2060798 (アイデンティティー管理) を追加
- バグ修正 BZ#2060798 (アイデンティティー管理) を追加

0.1-1

2022 年 9 月 12 日 (月) Gabriela Fialová (gfialova@redhat.com))

- proc_providing-feedback-on-red-hat-documentation.adoc を更新
- 拡張機能 BZ#2119694 (セキュリティー) を追加

0.1-0

2022 年 8 月 22 日 (火) Lenka Špačková (lspackova@redhat.com)

- 非推奨の機能 BZ#2069279 および BZ#2106816 (コンテナー) を追加
- z-stream の修正 (コンテナー) に関する情報で JIRA-RHELPLAN-121180 を更新

0.0-9

2022 年 8 月 10 日 (水) Lenka Špačková (lspackova@redhat.com)

- 既知の問題 BZ#1991500 を追加 (動的プログラミング言語、Web サーバー、およびデータベースサーバー)。

0.0-8

2022 年 8 月 4 日 (木) Gabriela Fialová (gfialova@redhat.com)

- 拡張機能 JIRA-RHELPLAN-118914 (コンテナー) を追加
- 既知の問題 BZ#2105162 (セキュリティー) を追加。
- 既知の問題 BZ#1960467 (グラフィックスインフラストラクチャー) を追加

0.0-7

2022 年 7 月 28 日 (木) Lenka Špačková (lspackova@redhat.com)

- BZ#2099438 (Security) の拡張機能を追加
- 既知の問題 BZ#2087253 (セキュリティー) を追加
- Distribution の Application Streams の拡張情報

0.0-6

2022 年 7 月 11 日 (木) Lenka Špačková (lspackova@redhat.com)

- 既知の問題 BZ#2077450 を追加
- BZ#2091653 の機能拡張を追加
- バグ修正 BZ#2006230 を追加

0.0-5

2022 年 6 月 29 日 (水) Lenka Špačková (lspackova@redhat.com)

- 既知の問題 BZ#2087121、BZ#2073567、BZ#2083727、および BZ#2005173 を追加
- BZ#2091643 の機能拡張を追加

0.0-4

2022 年 6 月 1 日 (水) Gabriela Fialová (gfialova@redhat.com)

- 既知の問題 BZ#2027576 を追加

0.0-3

2022 年 5 月 24 日 (水) Gabriela Fialová (gfialova@redhat.com)

- 人気のあるカスタマーポータルラボのトップ 10 リストを更新
- 非推奨の機能 BZ#2089200 (ネットワーキング) を追加し、再公開しました。

0.0-2

2022 年 5 月 18 日 (水) Gabriela Fialová (gfialova@redhat.com)

- Red Hat Enterprise Linux 9.0 リリースノートのリリース

0.0-1

2021 年 11 月 3 日 (水) Lenka Špačková (lspackova@redhat.com)

- Red Hat Enterprise Linux 9.0 Beta リリースノート