



# Red Hat Enterprise Linux 8

## 8.0 リリースノート

Red Hat Enterprise Linux 8.0 リリースノート



# Red Hat Enterprise Linux 8 8.0 リリースノート

---

Red Hat Enterprise Linux 8.0 リリースノート

## 法律上の通知

Copyright © 2019 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

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

All other trademarks are the property of their respective owners.

## 概要

本リリースノートでは、Red Hat Enterprise Linux 8.0 での改良点や実装された追加機能の概要、本リリースにおける既知の問題を説明します。また、重要なバグ修正、テクノロジープレビュー、使用されなくなった機能などの詳細も説明します。

## 目次

RED HAT ドキュメントへのフィードバック .....	4
<b>第1章 概要</b> .....	<b>5</b>
ディストリビューション	5
ソフトウェア管理	5
Web サーバー、データベース、動的言語	5
デスクトップ	5
インストーラーおよびイメージの作成	5
ファイルシステムおよびストレージ	6
セキュリティ	6
ネットワーク	6
仮想化	6
コンパイラーおよび開発ツール	6
高可用性およびクラスター	7
関連資料	7
<b>第2章 アーキテクチャー</b> .....	<b>8</b>
<b>第3章 RHEL 8 のコンテンツの配布</b> .....	<b>9</b>
3.1. インストール	9
3.2. リポジトリ	9
3.3. アプリケーションストリーム	9
<b>第4章 新機能</b> .....	<b>11</b>
4.1. WEB コンソール	11
4.2. インストーラーおよびイメージの作成	12
4.3. カーネル	13
4.4. ソフトウェア管理	15
4.5. インフラストラクチャーサービス	16
4.6. シェルおよびコマンドラインツール	17
4.7. 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	18
4.8. デスクトップ	24
4.9. ID 管理	26
4.10. コンパイラーおよび開発ツール	30
4.11. ファイルシステムおよびストレージ	39
4.12. 高可用性およびクラスター	43
4.13. ネットワーク	45
4.14. セキュリティー	49
4.15. 仮想化	54
4.16. サポート関連	56
<b>第5章 テクノロジープレビュー</b> .....	<b>57</b>
5.1. カーネル	57
5.2. ファイルシステムおよびストレージ	58
5.3. 高可用性およびクラスター	59
5.4. セキュリティー	59
5.5. 仮想化	59
<b>第6章 非推奨の機能</b> .....	<b>61</b>
6.1. インストーラーおよびイメージの作成	61
6.2. ファイルシステムおよびストレージ	61
6.3. ネットワーク	61
6.4. セキュリティー	62

---

6.5. 仮想化	62
<b>第7章 既知の問題</b> .....	<b>64</b>
7.1. インストーラーおよびイメージの作成	64
7.2. カーネル	64
7.3. ソフトウェア管理	65
7.4. インフラストラクチャーサービス	66
7.5. シェルおよびコマンドラインツール	66
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	66
7.7. ID 管理	67
7.8. コンパイラーおよび開発ツール	68
7.9. ファイルシステムおよびストレージ	69
7.10. ネットワーク	70
7.11. セキュリティー	70
<b>第8章 コンテナに関する主な変更点</b> .....	<b>72</b>
<b>付録A コンポーネント別のチケットリスト</b> .....	<b>73</b>



## RED HAT ドキュメントへのフィードバック

ドキュメントの改善に関するご意見やご要望をお聞かせください。

- 特定の文章に簡単なコメントを記入する場合は、ドキュメントが Multi-page HTML 形式になっているのを確認してください。コメントを追加する部分を強調表示し、そのテキストの下に表示される **Add Feedback** ポップアップをクリックし、表示された手順に従ってください。
- より詳細なフィードバックを行う場合は、Bugzilla のチケットを作成します。
  1. [Bugzilla](#) の Web サイトにアクセスします。
  2. Component で **Documentation** を選択します。
  3. **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記入してください。
  4. **Submit Bug** をクリックします。



## 第1章 概要

Red Hat Enterprise Linux 8.0 は、Fedora 28 およびアップストリームカーネル 4.18 を基に、従来および新しいワードロードに対応するのに必要なツールを備え、ハイブリッドクラウドデプロイメント全体に、安定した、安全で、一貫した基盤を提供します。このリリースの主な機能は以下のようになります。

### ディストリビューション

- コンテンツは、**BaseOS** リポジトリおよびアプリケーションストリーム (**AppStream**) リポジトリで利用できます。
- **AppStream** リポジトリは、従来の RPM 形式の新しい拡張機能 (**モジュール**) に対応します。これにより、同じコンポーネントのメジャーバージョンを複数インストールできるようになります。

詳細は[3章 RHEL 8 のコンテンツの配布](#)を参照してください。

### ソフトウェア管理

- **YUM** パッケージマネージャーは **DNF** テクノロジーに基づいており、モジュール式コンテンツ、パフォーマンス向上、およびツールングとの統合用への、安定し、優れた API を提供します。

詳細は「[ソフトウェア管理](#)」を参照してください。

### Web サーバー、データベース、動的言語

- RHEL 8 におけるデフォルトの Python 実装は **Python 3.6** で、Python 2.7 への対応は限定されます。デフォルトでは、いずれのバージョンもインストールされていません。
- RHEL 8 で提供しているデータベースサーバーは、MariaDB 10.3、MySQL 8.0、PostgreSQL 10、PostgreSQL 9.6、および Redis 4.0 です。

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。

### デスクトップ

- **GNOME Shell** がバージョン 3.28 にリベースしました。
- GNOME セッションおよび GNOME Display Manager では、デフォルトのディスプレイサーバーとして **Wayland** を使用します。RHEL 7 でデフォルトのディスプレイサーバーとして使用されていた **X.Org** サーバーも利用できます。

詳細は「[デスクトップ](#)」を参照してください。

### インストーラーおよびイメージの作成

- **Anaconda** インストーラーは、**LUKS2** ディスク暗号化を使用し、**NVDIMM** デバイスへのインストールに対応するようになりました。
- 新しい **Composer** ツールでは、さまざまな形式でカスタマイズしたシステムイメージを作成できます。Composer は **テクノロジープレビュー** として利用できます。
- **IBM Z** では、Hardware Management Console (**HMC**) および Support Element (**SE**) を使用した、DVD からのインストールに対応します。

詳細は「[インストーラーおよびイメージの作成](#)」を参照してください。

## ファイルシステムおよびストレージ

- ローカルストレージマネージャー **Stratis** が導入されました。Stratis により、統合インターフェースを使用して複雑なストレージタスクを簡単に実行し、ストレージスタックを管理できます。
- 従来の LUKS (LUKS1) 形式が、LUKS バージョン 2 (**LUKS2**) 形式に置き換わりました。**dm-crypt** サブシステムおよび **cryptsetup** ツールで、暗号化ボリュームのデフォルト形式として LUKS2 が使用されるようになりました。

詳細は「[ファイルシステムおよびストレージ](#)」を参照してください。

## セキュリティ

- システム全体の **暗号化ポリシー** により、コア暗号化サブシステムにポリシーを設定します。デフォルトでは、TLS、IPSec、SSH、DNSSec、および Kerberos のプロトコルが対象です。管理者は、新しい **update-crypto-policies** コマンドを使用して、モード間 (default、legacy、future、および fips) を簡単に切り替えることができます。
- PKCS #11** を使用した **スマートカード** および Hardware Security Modules ( **HSM**) のサポート手順が、システム全体で統一されています。

詳細は「[セキュリティ](#)」を参照してください。

## ネットワーク

- iptables** に代わり、**nftables** フレームワークが、デフォルトのネットワークパケットフィルタリングフレームワークになりました。
- firewalld** デーモンでは、デフォルトのバックエンドとして **nftables** が使用されるようになりました。
- 複数のコンテナに対してネットワーク接続を有効にする仮想ネットワークドライバー **IPVLAN** に対応します。

詳細は「[ネットワーク](#)」を参照してください。

## 仮想化

- RHEL 8 の仮想マシンでより近代的な PCI Express ベースのマシントイプ (**Q35**) に対応するようになり、新規に作成すると自動的に設定されるようになりました。これにより、さまざまな機能改善と、仮想デバイスの互換性が追加されました。
- RHEL 8 Web コンソール (**Cockpit** としても知られています) から、仮想マシンを作成して管理できるようになりました。
- QEMU** エミュレーターに **sandboxing** 機能が導入されました。これにより、QEMU が実行するシステムコールに設定可能な制限が課され、仮想マシンがより安全になります。

詳細は「[仮想化](#)」を参照してください。

## コンパイラーおよび開発ツール

- バージョン 8.2 に基づいた **GCC** コンパイラーは、新しい C++ 言語の標準仕様バージョンに対応します。最適化の改善と、新しいコードの堅牢化技術、警告の改善、そして新しいハードウェア機能に対応します。

- コード生成、操作、およびデバッグ用のさまざまなツールが、**DWARF5** デバッグの情報形式を実験的に処理できるようになりました。
- **eBPF** トレースへのカーネルサポートが、**BCC**、**PCP**、**SystemTap** などのツールで利用できます。
- バージョン 2.28 に基づいた **glibc** ライブラリーでは、Unicode 11 のサポート、新しい Linux システムコール、DNS スタブリゾルバーにおける重要な改善、セキュリティーの強化、パフォーマンスの改善が追加されました。

詳細は「[コンパイラおよび開発ツール](#)」を参照してください。

## 高可用性およびクラスター

- **Pacemaker** クラスターリソースマネージャーがアップストリームのバージョン 2.0.0 にアップグレードしており、バグ修正および機能強化が数多く追加されました。
- RHEL 8 の **pcs** 設定システムで、Corosync 3、**knet**、およびノード名に完全に対応するようになりました。

詳細は「[高可用性およびクラスター](#)」を参照してください。

## 関連資料

- 他のバージョンと比較した Red Hat Enterprise Linux 8.0 の機能および制限は、Red Hat ナレッジベースアトicle「[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。
- RHEL 8 パッケージの一覧は『[Package manifest](#)』を参照してください。

## 第2章 アーキテクチャー

Red Hat Enterprise Linux 8.0 ではカーネルバージョン 4.18 が使用されており、以下のアーキテクチャーに対応します。

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

## 第3章 RHEL 8 のコンテンツの配布

### 3.1. インストール

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

- Binary DVD ISO: BaseOS リポジトリおよび AppStream リポジトリが含まれ、リポジトリを追加しなくてもインストールを完了できる完全インストールイメージです。
- Boot ISO: インストールプログラムを起動するのに使用する最小限の ISO ブートイメージです。このオプションでは、ソフトウェアパッケージをインストールするのに、BaseOS リポジトリおよび AppStream リポジトリにアクセスする必要があります。リポジトリは、Binary DVD ISO イメージに含まれます。

ISO イメージのダウンロード、インストールメディアの作成、RHEL インストールの完了の方法は、『[標準的な RHEL インストールの実行](#)』を参照してください。自動化した Kickstart インストールなどの高度なトピックは『[高度な RHEL インストールの実行](#)』を参照してください。

### 3.2. リポジトリ

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

- BaseOS
- AppStream

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

BaseOS リポジトリのコンテンツは、すべてのインストールの基盤になる OS 機能のコアセットを提供することを目的としています。このコンテンツは RPM 形式で提供されており、RHEL の以前のリリースと同様のサポート条件が適用されます。BaseOS から配布されるパッケージの一覧は『[Package manifest](#)』を参照してください。

Application Stream (アプリケーションストリーム) リポジトリのコンテンツには、さまざまなワークロードとユースケースに対応するために、ユーザー領域アプリケーション、ランタイム言語、およびデータベースが追加されています。AppStream のコンテンツ形式には、従来の RPM 形式と、RPM 形式の拡張 (**モジュール** と呼ばれます) の 2 つが利用できます。AppStream で利用可能なパッケージの一覧は『[Package manifest](#)』を参照してください。

また、CodeReady Linux Builder リポジトリは、すべての RHEL サブスクリプションで利用できます。このリポジトリは、開発者向けの追加パッケージを提供します。CodeReady Linux Builder リポジトリに含まれるパッケージは、本番環境での使用はサポートされていません。

RHEL 8 リポジトリの詳細は『[Package manifest](#)』を参照してください。

### 3.3. アプリケーションストリーム

Red Hat Enterprise Linux 8.0 では、アプリケーションストリームの概念が導入されました。ユーザー領域コンポーネントのバージョンは複数配信され、コアオペレーティングシステムパッケージよりも頻繁に更新されるようになりました。これによりプラットフォームや特定のデプロイメントの基本的な安定性に影響を及ぼすことなく、Red Hat Enterprise Linux をカスタマイズする柔軟性が向上します。

Application Streams として使用できるコンポーネントは、モジュールまたは RPM パッケージとしてパッケージ化され、RHEL 8 の AppStream リポジトリを介して配信されます。Application Stream のコンポーネントには、それぞれ特定のライフサイクルがあります。詳細は「[Red Hat Enterprise Linux のライフサイクル](#)」を参照してください。

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

モジュールストリームは、アプリケーションストリームコンポーネントのバージョンを表します。たとえば、`postgresql` モジュールでは 2 つのストリーム (バージョン) の PostgreSQL データベースサーバー、つまり PostgreSQL 10 (デフォルトストリーム) および PostgreSQL 9.6 が利用できます。システムにインストールできるモジュールストリームは 1 つだけです。複数のコンテナで異なるバージョンを使用できます。

詳細なモジュールコマンドは『[Application Stream の使用](#)』を参照してください。AppStream で利用可能なモジュールの一覧は『[Package manifest](#)』を参照してください。

## 第4章 新機能

ここでは、Red Hat Enterprise Linux 8 で導入される新機能および主要な機能拡張を説明します。

### 4.1. WEB コンソール



#### 注記

Web コンソールのサブスクリプションページは、新しい **subscription-manager-cockpit** パッケージで提供されるようになりました。

#### Web コンソールに firewall インターフェースが追加

RHEL 8 Web コンソールの **Networking** ページに **Firewall** セクションが追加されました。このセクションでは、ファイアウォールの有効化または無効化、およびファイアウォールルールの追加、削除、変更を行います。

(BZ#1647110)

#### Web コンソールがデフォルトで利用可能

RHEL 8 Web コンソールのパッケージ (Cockpit と呼ばれます) は、Red Hat Enterprise Linux のデフォルトリポジトリに同梱されるようになったため、登録済みの RHEL 8 システムにすぐにインストールできます。

さらに、RHEL 8 の最小インストール以外のインストールでは、Web コンソールが自動的にインストールされ、コンソールに必要なファイアウォールポートが自動的に開くようになりました。ログイン前に、Web コンソールを有効にしたり、Web コンソールにアクセスする方法を示すシステムメッセージも追加されました。

(JIRA:RHELPLAN-10355)

#### Web コンソール用の IdM 統合が改善

システムが Identity Management (IdM) ドメインに登録されていると、RHEL 8 Web コンソールはデフォルトで、ドメインで集中管理されている IdM リソースを使用するようになりました。これには、以下の利点があります。

- IdM ドメインの管理者は、Web コンソールを使用して、ローカルマシンを管理できます。
- コンソールの Web サーバーでは、IdM 認証局 (CA) が発行した証明書に自動的に切り替わり、ブラウザーに許可されます。
- IdM ドメインに Kerberos チケットがあると、Web コンソールにアクセスする際にログイン認証情報を指定する必要がなくなりました。
- IdM ドメインで認識されている SSH ホストは、手動で SSH 接続を追加しなくても Web コンソールにアクセスできます。

IdM と Web コンソールの統合が適切に機能するようにするには、IdM マスターシステムの **enable-admins-sudo** オプションを使用して、**ipa-advise** コーティリティーを実行する必要があります。

(JIRA:RHELPLAN-3010)

#### Web コンソールにモバイルブラウザーとの互換性が追加

今回の更新で、モバイルブラウザのバリエーションで、Web コンソールメニューとページを移動できるようになりました。これにより、モバイルデバイスから RHEL 8 Web コンソールを使用してシステムを管理できます。

(JIRA:RHELPLAN-10352)

### Web コンソールのフロントページに、不足している更新およびサブスクリプションを表示

RHEL 8 Web コンソールが管理するシステムに古いパッケージまたは失効したサブスクリプションがある場合は、システムの Web コンソールのフロントページに警告が表示されるようになりました。

(JIRA:RHELPLAN-10353)

### Web コンソールが PBD 登録に対応

今回の更新で、RHEL 8 Web コンソールインターフェースを使用して、管理システムのディスクに PBD (Policy-Based Decryption) ルールを適用できるようになりました。これは、Clevis 複号クライアントを使用して、LUKS で暗号化されたディスクパーティションの自動ロック解除など、Web コンソールのさまざまなセキュリティー管理機能を容易にします。

(JIRA:RHELPLAN-10354)

### Web コンソールを使用して仮想マシンが管理可能に

RHEL 8 Web コンソールインターフェースに、**Virtual Machines** ページを追加できるようになりました。これにより、libvirt ベースの仮想マシンを作成および管理できるようになりました。

(JIRA:RHELPLAN-2896)

## 4.2. インストーラーおよびイメージの作成

### IBM Z で、SE および HMC を使用して DVD から RHEL をインストールする方法を完全にサポート

IBM Z ハードウェアで、**Support Element (SE)** および **Hardware Management Console (HMC)** を使用して DVD から Red Hat Enterprise Linux 8 をインストールすることに完全に対応するようになりました。この追加機能により、SE および HMC を使用した IBM Z のインストールプロセスが簡単になります。

Binary DVD から起動すると、インストーラーにより、追加のカーネルパラメーターを入力するように求められます。DVD をインストールソースとして設定する場合は、カーネルパラメーターに **inst.repo=hmc** を追加すると、インストーラーにより、SE ファイルおよび HMC ファイルへのアクセスが有効になり、DVD から stage2 のイメージを取得して、DVD 上のパッケージにアクセスしてソフトウェアを選択できるようになります。

この新機能により、外部ネットワーク設定の必要がなくなるため、インストールのオプションが増えます。

(BZ#1500792)

### インストーラーが LUKS2 ディスク暗号化フォーマットをサポート

Red Hat Enterprise Linux 8 インストーラーは、デフォルトで LUKS2 フォーマットを使用しますが、**Anaconda** の Custom Partitioning ウィンドウ、Kickstart の **autopart** コマンド、**logvol** コマンド、**part** コマンド、および **RAID** コマンドで新しいオプションを使用して、LUKS バージョンを選択できます。



LUKS2 は改良や機能を数多く提供します。たとえば、ディスク上のフォーマット機能を拡張し、メタデータを格納する柔軟な方法を提供します。

(BZ#1547908)

### Anaconda が RHEL 8 でシステムの目的をサポート

以前は、**Anaconda** から、**Subscription Manager** にシステムの目的の情報を提供していませんでした。Red Hat Enterprise Linux 8.0 では、**Anaconda** の **システムの目的** ウィンドウまたは Kickstart の **syspurpose** コマンドを使用して、システムの使用目的を設定できるようになりました。そのシステムの目的情報は、インストールが完了し、システムをサブスクライブする際に **Subscription Manager** が使用します。

(BZ#1612060)

### Pykickstart が RHEL 8 でシステムの目的をサポート

以前は、**pykickstart** ライブラリーで、**Subscription Manager** にシステムの目的の情報が提供されていました。Red Hat Enterprise Linux 8.0 では、**pykickstart** が、新しい **syspurpose** コマンドを解析し、完全または一部自動化しているインストール時に、システムの使用目的を記録します。その後、その情報は **Anaconda** に渡され、新しくインストールしたシステムに保存され、システムのサブスクライブ時に **Subscription Manager** で利用できるようになります。

(BZ#1612061)

### Anaconda が RHEL 8 で新しいカーネル起動パラメーターをサポート

以前は、カーネル起動パラメーターからベースリポジトリだけを指定できました。Red Hat Enterprise Linux 8 では、新しいカーネルパラメーター **inst.addrepo=<name>,<url>** を使用して、インストール時に追加リポジトリを指定できるようになりました。

このパラメーターでは、リポジトリの名前と、リポジトリを指定する URL の 2 つの値が必要になります。詳細は <https://anaconda-installer.readthedocs.io/en/latest/boot-options.html#inst-addrepo> を参照してください。

(BZ#1595415)

### Anaconda が RHEL 8 で統合 ISO をサポート

Red Hat Enterprise Linux 8.0 では、統合 ISO が、インストールソースリポジトリ BaseOS および AppStream を自動的に読み込みます。

この機能は、インストール時に読み込まれる最初のベースリポジトリに対して有効です。たとえば、リポジトリを設定せずにインストールを起動し、ベースリポジトリとして統合 ISO が GUI にある場合、または、統合 ISO を指定する **inst.repo=** オプションを使用してインストールを起動する場合などです。その結果、AppStream リポジトリは、**Installation Source** GUI ウィンドウの **Additional Repositories** セクションで有効になります。AppStream リポジトリを削除したり、その設定を変更することはできませんが、**インストールソース** でそれを無効にできます。別のベースリポジトリを使用してインストールを起動し、そのあとで統合 ISO に変えた場合にはこの機能は無効です。この場合は、ベースリポジトリが置き換えられますが、AppStream リポジトリは置き換えられず、元のファイルを指定します。

(BZ#1610806)

## 4.3. カーネル

### ARM 52 ビット物理アドレス指定が可能に

今回の更新で、64 ビット ARM アーキテクチャー用の 52 ビット物理アドレッシング (PA) のサポートが利用可能になりました。これにより、以前の 48 ビットの PA よりも大きなアドレス空間が提供されます。

(BZ#1643522)

## IOMMU コードが RHEL 8 で 5 レベルのページテーブルをサポート

Linux カーネルの I/O メモリ管理ユニット (IOMMU) コードは、Red Hat Enterprise Linux 8 で 5 レベルのページテーブルをサポートするように更新されました。

(BZ#1485546)

## 5 レベルのページングをサポート

Red Hat Enterprise Linux 8 で 5 レベルのページングをサポートするために、新しい **P4d\_t** ソフトウェアページのテーブルタイプが Linux カーネルに追加されました。

(BZ#1485532)

## メモリー管理が 5 レベルのページテーブルをサポート

Red Hat Enterprise Linux 7 では、既存のメモリーバスに、48/46 ビットの仮想または物理のメモリーアドレッシング機能があり、Linux カーネルは、4 つのレベルのページテーブルを実装して、物理アドレスへの仮想アドレスを管理します。物理バスのアドレス線は、物理メモリーの容量上限を 64 TB までに制限します。

この制限は、57/52 ビットの仮想および物理のメモリーアドレスにより、128 PiB の仮想アドレス空間と、4 PB の物理メモリーの容量まで拡張されました。

拡張アドレス範囲では、Red Hat Enterprise Linux 8 のメモリー管理は、拡張アドレス範囲を処理できるように、5 レベルのページテーブルの実装に対応するようになりました。

(BZ#1485525)

## RHEL 8 で、**kernel-signing-ca.cer** が **kernel-core** に移動

Red Hat Enterprise Linux 7 では、すべてのバージョンで、**kernel-signing-ca.cer** 公開鍵が **kernel-doc** パッケージにありましたが、Red Hat Enterprise Linux 8 では、すべてのアーキテクチャーで、**kernel-signing-ca.cer** が **kernel-core** パッケージに移動しました。

(BZ#1638465)

## eBPF ベースのプログラムおよびマップの調査および操作に使用する **bpftool** が追加

eBPF (extended Berkeley Packet Filtering) に基づくプログラムとマップを利用した、インスペクションと簡単な操作に役立つ **bpftool** ユーティリティーが Linux カーネルに追加されました。**bpftool** はカーネルソースツリーの一部で、**bpftool** パッケージにより提供されます。これは、**kernel** パッケージのサブパッケージとして同梱されます。

(BZ#1559607)

## **kernel-rt** ソースが更新

**kernel-rt** ソースが更新され、最新の RHEL カーネルソースツリーを使用するようになりました。最新のカーネルソースツリーは、現在、アップストリーム v4.18 のリアルタイムパッチセットを使用しています。これにより、以前のバージョンにバグ修正および機能強化が数多く追加されました。

(BZ#1592977)

## 4.4. ソフトウェア管理

### YUM パフォーマンスの改善、およびモジュール式コンテンツのサポート

Red Hat Enterprise Linux 8 へのソフトウェアのインストールは、新しいバージョンの YUM ツールで実施されます。これは、DNF テクノロジーに基づいています。

DNF に基づいている YUM は、RHEL 7 で使用されていた以前の YUM v3 に比べて、以下の点が優れています。

- パフォーマンスの向上
- モジュール化コンテンツのサポート
- ツーリングと統合するために適切に設計され、安定した API

新しい YUM ツールと、以前のバージョンである RHEL 7 の YUM v3 との間における相違点の詳細は、[http://dnf.readthedocs.io/en/latest/cli\\_vs\\_yum.html](http://dnf.readthedocs.io/en/latest/cli_vs_yum.html) を参照してください。

DNF ベースの YUM は、コマンドラインでの利用、および設定ファイルの編集または作成について、YUM v3 と互換性があります。

ソフトウェアをインストールする際は、RHEL 7 と同じ方法で **yum** コマンドと特定のオプションを使用できます。パッケージは、**Provides** を使用して、以前の名前でインストールできます。このパッケージは互換性を持ったシンボリックリンクを提供するため、バイナリー、設定ファイル、ディレクトリーは通常の方法で確認できます。

YUM v3 が提供するレガシーの Python API、および Libdnf C API は不安定で、Red Hat Enterprise Linux 8 のライフサイクル期間内に変更する可能性があります。プラグインおよびスクリプトは、新しい DNF Python API に移行することが推奨されます。DNF Python API は <https://dnf.readthedocs.io/en/latest/api.html> から利用できます。

YUM v3 機能の一部は、DNF をベースにした YUM の挙動と異なる可能性があります。この変更がワークフローに影響を与える場合は、「[How do I open and manage a support case on the Customer Portal?](#)」に従って、Red Hat サポートでケースを作成してください。

(BZ#1581198)

### RHEL 8 における主な RPM 機能

Red Hat Enterprise Linux 8 は RPM 4.14 で配布されています。このバージョンでは、RHEL 7 で利用できた RPM 4.11 に対する機能拡張が数多く追加されました。以下は、主な機能です。

- **debuginfo** パッケージの複数バージョンを同時にインストール可能
- 弱い依存関係のサポート
- 豊富なブール型依存関係のサポート
- 4 GB を超えるファイルのパッケージ化サポート
- ファイルトリガーのサポート

以下は、主な変更点です。

- 厳密なスペックパーサー
- 非冗長モードで出力を確認する簡易署名

- マクロの追加および廃止

(BZ#1581990)

## RPM が、インストールを開始する前にパッケージ全体のコンテンツを検証

Red Hat Enterprise Linux 7 では、RPM ユーティリティーにより、解凍時に各ファイルのペイロードの内容が検証されていました。以下の理由により、この方法は十分ではありません。

- ペイロードが破損していても、スクリプトアクションを実行した後にしか通知されず、元に戻せません。
- ペイロードが破損していると、以前のバージョンのファイルの一部置き換えた後でパッケージのアップグレードが中止するため、正常に機能しません。
- 個々のファイルのハッシュは、圧縮されていないデータに対して実行するため、RPM は、デコンプレッサーの脆弱性に対して脆弱です。

Red Hat Enterprise Linux 8 では、インストール前に、利用可能な最善のハッシュを使用して、パッケージ全体が個別の手順で検証されます。

Red Hat Enterprise Linux 8 に構築したパッケージでは、圧縮ペイロードに新しい **SHA256** ハッシュを使用しています。署名付きパッケージでは、署名によりペイロードハッシュの保護が強化されるため、パッケージヘッダーでは、署名やその他のハッシュを壊さずにペイロードハッシュを変更することはできません。以前のパッケージでは、FIPS モードなどの設定で無効にされていない限り、ヘッダーとペイロードの **MD5** ハッシュを使用します。

さらに、`%\_pkgverify\_level` マクロを使用して、インストール前に署名検証を強制することを有効にしたり、ペイロード検証を完全に無効にしたりできます。さらに、`%\_pkgverify\_flags` マクロを使用して、どのハッシュおよび署名を許可するかを制限できます。たとえば、弱い **MD5** ハッシュの使用を無効にすることもできますが、これにより、古いパッケージとの互換性がなくなります。

(JIRA:RHELPLAN-1499)

## 4.5. インフラストラクチャーサービス

### RHEL 8 で推奨される Tuned プロファイルの主な変更点

今回の更新で、(**tuned-adm recommend** コマンドにより報告される) 推奨される Tuned プロファイルが、次のルールに基づいて選択されるようになりました。一致する最初の規則が有効になります。

- (**syspurpose** コマンドにより報告される) **syspurpose** ロールには「**atomic**」が含まれ、以下のようになります。
  - Tuned がベアメタルで実行している場合は、「**atomic-host**」プロファイルが選択されません。
  - Tuned が仮想マシンで実行している場合は、「**atomic-guest**」プロファイルが選択されません。
- Tuned が仮想マシンで実行している場合は、「**virtual-guest**」プロファイルが選択されます。
- **syspurpose** ロールに **desktop** または **workstation** が含まれ、シャーシタイプ (**dmidecode** が報告) が **Notebook**、**Laptop**、または **Portable** の場合は、**balanced** プロファイルが選択されます。

- 上記のどのルールにも一致しない場合は「**throughput-performance**」プロファイルが選択されます。

(BZ#1565598)

### named が作業ディレクトリーに生成するファイルを書き込み可能に

以前は、**named** デーモンが、作業ディレクトリーにデータを保存していましたが、Red Hat Enterprise Linux では、このディレクトリーは読み取り専用でした。今回の更新で、選択したファイルのパスが、書き込みが許可されているサブディレクトリーに変更になりました。これにより、Unix および SELinux のパーミッションでは、デフォルトのディレクトリーへの書き込みが許可されています。ただし、ディレクトリー内に配布されているファイルは引き続き **named** に対して読み取り専用になります。

(BZ#1588592)

### Geolite Databases が Geolite2 Databases に置き換え

Red Hat Enterprise Linux 7 で提供していた Geolite Databases が、Red Hat Enterprise Linux 8 で Geolite2 Databases に置き換えられました。

Geolite Databases は **GeoIP** パッケージで提供されていました。アップストリームでは、従来のデータベースが同梱されていたこのパッケージがサポートされなくなりました。

Geolite2 Databases は、複数のパッケージで提供されます。**libmaxminddb** パッケージには、ライブラリーと、アドレスの手動検索を可能にする **mmdblookup** コマンドラインツールが同梱されています。従来の **GeoIP** パッケージの **geoipupdate** バイナリーは **geoipupdate** パッケージで提供されているため、従来のデータベースと新しい Geolite2 データベースの両方をダウンロードできるようになりました。

(JIRA:RHELPLAN-6746)

### journald が CUPS ログに対応

RHEL 8 では、RHEL 7 で使用されていた **/var/log/cups** ディレクトリーの特定ファイルに CUPS ログが保存されなくなりました。RHEL 8 では、どのタイプの CUPS ログも、他のプログラムとともに **systemd** の **journald** デーモンが集中管理します。したがって、CUPS ログにアクセスするには、**journalctl -u cups** コマンドを使用します。詳細は「[Working with CUPS logs](#)」を参照してください。

(JIRA:RHELPLAN-12764)

## 4.6. シェルおよびコマンドラインツール

### nobody ユーザーが nfsnobody に置き換え

Red Hat Enterprise Linux 7 では、以下のように設定されていました。

- ID が 99 の **nobody** のユーザーおよびグループのペア
- ID が 65534 の **nfsnobody** ユーザーとグループのペア。この ID は、デフォルトのカーネルオーバーフロー ID です。

これはいずれも、Red Hat Enterprise Linux 8 では、**nobody** ユーザーおよびグループのペア (ID 65534) に統合され、新しいインストールでは、**nfsnobody** ペアは作成されなくなりました。

この変更により、**nobody** が所有していて、NFS とは無関係のファイルに関する混乱が軽減されます。

(BZ#1591969)

## RHEL 8 のバージョン管理システム

RHEL 8 は、次のバージョン管理システムを提供します。

- **Git 2.18** は、分散アーキテクチャーを持つ分散型リビジョン管理システムです。
- **Mercurial 4.8** は、大規模プロジェクトを効率的に処理するために設計された、軽量の分散バージョン管理システムです。
- **Subversion 1.10** は、集中バージョン管理システムです。詳細は「[Notable changes in Subversion 1.10](#)」を参照してください。

RHEL 7 で利用できた Concurrent Versions System (CVS) は、RHEL 8 では配布されていません。

(BZ#1693775)

### Subversion 1.10 への主な変更点

**Subversion 1.10** には、RHEL 7 で配布されたバージョン 1.7 以降に追加された新機能と、次の互換性の変更が含まれています。

- 言語バインディングに対応するために使用される **Subversion** ライブラリーにおける非互換性のため、**Subversion 1.10** の **Python 3** バインディングは利用できません。その結果、**Subversion** に **Python** バインディングを必要とするアプリケーションには対応していません。
- **Berkeley DB** に基づくリポジトリには対応しなくなりました。移行する前に、**svnadmin dump** コマンドを使用して、**Subversion 1.7** で作成したリポジトリをバックアップします。RHEL 8 をインストールした後、**svnadmin load** コマンドを使用してリポジトリを復元します。
- RHEL 7 の **Subversion 1.7** クライアントがチェックアウトした既存の作業コピーは、**Subversion 1.10** から使用する前に新しい形式にアップグレードする必要があります。RHEL 8 をインストールしたら、各作業コピーで **svn upgrade** コマンドを実行します。
- **https://** を使用してリポジトリにアクセスするスマートカード認証はサポートされなくなりました。

(BZ#1571415)

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

### Python 3 が RHEL 8 におけるデフォルトの Python 実装に

Red Hat Enterprise Linux 8 には、**Python 3.6** が同梱されています。このパッケージはデフォルトではインストールされません。**Python 3.6** をインストールする場合は、**yum install python3** コマンドを使用します。

**Python 2.7** は、**python2** パッケージから入手できますが、**Python 3** への移行をより円滑に進められるように、**Python 2** のライフサイクルは短くなっています。

デフォルトの **python** パッケージも、バージョン管理外の **/usr/bin/python** 実行可能ファイルも、RHEL 8 には同梱されていません。**python3** または **python2** を直接使用することが推奨されます。もしくは、管理者が、**alternatives** コマンドを使用して、バージョン管理外の **python** コマンドを設定できます。

詳細は「[Using Python in Red Hat Enterprise Linux 8](#)」を参照してください。

(BZ#1580387)

## RPM ビルド時に Python スクリプトでハッシュタグにメジャーバージョンを指定することが必要

RHEL 8 で実行可能な Python スクリプトでは、主な Python バージョンを明示的に指定する hashbang (shebang) を使用します。

`/usr/lib/rpm/redhat/brp-mangle-shebangs` BRP (buildroot policy) スクリプトは、RPM パッケージを構築する際に自動的に実行し、実行可能なすべてのファイルで hashbangs を修正します。スクリプトで、Python のメジャーバージョンで解決できないあいまいな hashbangs に遭遇すると、エラーが作成されます。そのようなあいまいな hashbang の例は以下のようになります。

- `#!/usr/bin/python`
- `#!/usr/bin/env python`

Python スクリプトで、RPM ビルド時にエラーが発生する hashbang を修正するには、`platform-python-devel` パッケージの `pathfix.py` スクリプトを使用します。

```
pathfix.py -pn -i %[__python3] PATH ...
```

複数の `PATH` を指定できます。 `PATH` がディレクトリーの場合、`pathfix.py` は Python スクリプトを再帰的にスキャンして、`^[a-zA-Z0-9_]+\.$` パターンに一致するものを探します。これは、あいまいな hashbang があるものだけではありません。このコマンドを `%prep` セクション、または `%install` セクションに追加します。

詳細は「[Handling hashbangs in Python scripts](#)」を参照してください。

(BZ#1583620)

## PHP への主な変更点

Red Hat Enterprise Linux 8 には **PHP 7.2** が同梱されています。このバージョンには、RHEL 7 で利用できた **PHP 5.4** に対する重要な変更が導入されています。

- **PHP** はデフォルトで FastCGI Process Manager (FPM) を使用します (スレッド化された **httpd** で安全に使用できます)。
- `php_value` 変数および `php-flag` 変数は **httpd** 設定ファイルで使用されなくなり、代わりにプール設定の `/etc/php-fpm.d/*.conf` で設定する必要があります。
- **PHP** スクリプトのエラーと警告のログは、`/var/log/httpd/error.log` ではなく `/var/log/php-fpm/www-error.log` ファイルに記録されます。
- **PHP** の `max_execution_time` 設定変数を変更するときは、変更した値に合わせて **httpd ProxyTimeout** 設定を増やす必要があります。
- **PHP** スクリプトを実行するユーザーが、FPM プール設定 (`/etc/php-fpm.d/www.conf` ファイル、**apache** ユーザーがデフォルト) に設定されるようになりました。
- 設定を変更した場合、または新しい拡張機能をインストールした場合は、**php-fpm** サービスを再起動する必要があります。

以下の拡張機能が削除されました。

- **aspell**
- **mysql** (**mysqli** および **pdo\_mysql** 拡張機能は、**php-mysqldb** パッケージで引き続き利用できます)
- **zip**
- **memcache**

(BZ#1580430, [BZ#1691688](#))

## Ruby への主な変更点

RHEL 8 では、RHEL 7 に同梱されていた **Ruby 2.0.0** に新機能および機能強化を追加した **Ruby 2.5** が提供されます。以下は、主な変更点です。

- インクリメンタルガベージコレクターが追加されました。
- **Refinements** 構文が追加されました。
- シンボルは、ガベージコレクションが行われるようになりました。
- 安全レベルの **\$SAFE=2** および **\$SAFE=3** が廃止されました。
- **Fixnum** クラスと **Bignum** クラスが、**Integer** クラスに統合されました。
- **Hash** クラスの最適化、インスタンス変数へのアクセスの向上、**Mutex** クラスの小型化と高速化により、パフォーマンスが向上しました。
- 古い API が非推奨になりました。
- **RubyGems**、**Rake**、**RDoc**、**Psych**、**Minitest**、**test-unit** などのバンドルされたライブラリーが更新されました。
- **Ruby** とともに配布されていた、**mathn**、**DL**、**ext/tk**、および **XMLRPC** などのライブラリーは非推奨になり、同梱されなくなりました。
- **SemVer** バージョン管理スキームが、**Ruby** バージョン管理に使用されるようになりました。

(BZ#1648843)

## Perl への主な変更点

RHEL 8 で提供される **Perl 5.26** では、RHEL 7 で提供されていたバージョンに以下のような変更が追加されました。

- **Unicode 9.0** がサポートされるようになりました。
- 新しい **SystemTap** のプロブ **op-entry**、**loading-file**、および **loaded-file** が提供されるようになりました。
- パフォーマンスを向上させるために、スカラーを割り当てる際に、コピーオンライトメカニズムが使用されます。
- IPv4 ソケットおよび IPv6 ソケットを透過的に処理するために **IO::Socket::IP** モジュールが追加されました。



- 構造化された方法で **perl -V** データにアクセスするために、**Config::Perl::V** モジュールが追加されました。
- Comprehensive Perl Archive Network (CPAN) リポジトリからモジュールを取得、抽出、ビルド、およびインストールする **cpanm** ユーティリティーが同梱される **perl-App-cpanminus** パッケージが追加されました。
- セキュリティー上の理由により、**@INC** モジュールの検索パスから、現在のディレクトリ (.) が削除されました。
- 上記の動作上の変更によりファイルの読み込みに失敗した時に、**do** ステートメントが非推奨の警告を返すようになりました。
- **do subroutine(LIST)** 呼び出しはサポートされなくなり、構文エラーが発生するようになりました。
- ハッシュがデフォルトでランダム化されるようになりました。ハッシュから鍵と値が返される順序は、**perl** の実行ごとに変わります。ランダム化を無効にするには、**PERL\_PERTURB\_KEYS** 環境変数を **0** に設定します。
- 正規表現のパターンで、エスケープされていないリテラルの { 文字が使用できなくなりました。
- **\$\_** 変数に対する語彙的なスコープへの対応が削除されました。
- 配列またはハッシュに **defined** 演算子を使用すると、致命的なエラーが発生します。
- **UNIVERSAL** モジュールから関数をインポートすると、致命的なエラーが発生します。
- ツールの **find2perl**、**s2p**、**a2p**、**c2ph**、および **pstruct** が削除されました。
- **\${^ENCODING}** 機能が削除されました。**encoding** プラグマのデフォルトモードはサポートされなくなりました。**UTF-8** 以外のエンコーディングでソースコードを記述する場合は、エンコーディングの **Filter** オプションを使用します。
- アップストリームに合わせて、**perl** パッケージが変更になりました。**perl** パッケージは、コアモジュール群をインストールし、**/usr/bin/perl** インタープリターは **perl-interpreter** パッケージで提供されます。以前のリリースでは、**perl** パッケージには最小限のインタープリターだけが含まれていて、**perl-core** パッケージには、インタープリターとコアモジュールの両方が含まれていました。

(BZ#1511131)

## Node.js が RHEL 8 に新登場

JavaScript プログラミング言語で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ためのソフトウェア開発プラットフォームである **Node.js** が RHEL で初めて提供されます。以前は、ソフトウェアコレクションとしてのみ利用できました。RHEL 8 では **Node.js 10** が提供されます。

(BZ#1622118)

## SWIG への主な変更点

RHEL 8 には、SWIG (Simplified Wrapper and Interface Generator) バージョン 3.0 が含まれています。RHEL 7 に同梱されていたバージョン 2.0 に新機能、機能拡張、およびバグ修正が数多く追加されました。特に注目すべきは、C++11 標準仕様サポートの実装です。SWIG は、Go 1.6、PHP 7、Octave 4.2、および Python 3.5 もサポートしています。

(BZ#1660051)

## Apache httpd への主な変更点

RHEL 8 には、Apache HTTP Server 2.4.37 が同梱されています。このバージョンでは、RHEL 7 で提供されていた **httpd** に、次の変更点が追加されました。

- **httpd** モジュールに同梱される **mod\_http2** パッケージにより、HTTP/2 サポートが提供されるようになりました。
- (Let's Encrypt などの証明書プロバイダーで使用するため) 自動証明書管理環境 (ACME) プロトコルを使用した、TLS 証明書の自動プロビジョニングおよび更新が、**mod\_md** パッケージでサポートされるようになりました。
- Apache HTTP Server が、**PKCS#11** モジュールを利用して、ハードウェアのセキュリティトークンから、TLS 証明書および秘密鍵を直接読み込むようになりました。これにより、**mod\_ssl** 設定で、**SSLCertificateKeyFile** ディレクティブおよび **SSLCertificateFile** ディレクティブに、**PKCS#11** URL を使用して、TLS 秘密鍵と、必要に応じて TLS 証明書をそれぞれ指定できるようになりました。
- Apache HTTP Server でデフォルト設定されているマルチプロセッシングモジュール (MPM) は、(**prefork** として知られる) マルチプロセスのフォークモデルから、高パフォーマンスのマルチスレッドモデル **event** へ変更しました。スレッドセーフではないサードパーティーのモジュールは、交換または削除する必要があります。設定した MPM を変更するには、**/etc/httpd/conf.modules.d/00-mpm.conf** ファイルを編集します。詳細は、man ページの **httpd.conf(5)** を参照してください。

**httpd** の詳細は「[Setting up the Apache HTTP web server](#)」を参照してください。

(BZ#1632754, BZ#1527084, BZ#1581178)

## nginx Web サーバーが RHEL 8 に新登場

RHEL 8 では、HTTP などのプロトコルに対応する Web サーバーおよびプロキシサーバーである **nginx 1.14** が導入されました。**nginx 1.14** は、実行性およびパフォーマンスの高さ、ならびにメモリー使用量の低さに焦点を当てています。**nginx** は、以前はソフトウェアコレクションでしか入手できませんでした。

**nginx** の Web サーバーは、**PKCS#11** モジュールを利用してハードウェアセキュリティトークンから直接 TLS 秘密鍵を読み込むようになりました。これにより、**nginx** 設定で **ssl\_certificate\_key** ディレクティブに、**PKCS#11** URL を使用して TLS 秘密鍵を指定できるようになりました。

(BZ#1545526)

## RHEL 8 のデータベースサーバー

RHEL 8 は、次のデータベースサーバーを提供します。

- **MySQL 8.0**。マルチユーザー、マルチスレッドの SQL データベースサーバーです。**MySQL** サーバーデーモンである **mysqld** と、多数のクライアントプログラムで構成されます。
- **MariaDB 10.3**。マルチユーザー、マルチスレッドの SQL データベースサーバーです。あらゆる実用的な目的において、**MariaDB** には、**MySQL** とバイナリー互換性があります。
- **PostgreSQL 10** および **PostgreSQL 9.6**。高度なオブジェクトリレーショナルデータベース管理システム (DBMS) です。

- **Redis 5**。高度な鍵と値のストアです、鍵には、文字列、ハッシュ、リスト、セット、およびソートセットを含めることができるため、データ構造サーバーと呼ばれています。RHEL では、**Redis** が初めて提供されます。

NoSQL **MongoDB** データベースサーバーは、Server Side Public License (SSPL) を使用するため、RHEL 8.0 には同梱されていません。

(BZ#1647908)

## MySQL 8.0 への主な変更点

RHEL 8 に同梱される **MySQL 8.0** には、たとえば次のような機能強化が含まれます。

- **MySQL** に、データベースオブジェクトに関する情報を格納するトランザクショナルデータディクショナリーが組み込まれました。
- **MySQL** がロールに対応するようになりました。ロールは特権の集まりです。
- デフォルトの文字セットが、**latin1** から **utf8mb4** に変更しました。
- 非再帰的および再帰的な Common Table Expression のサポートが追加されました。
- **MySQL** クエリーで関連する行の集合に対して、行ごとに計算を実行する window 関数に対応するようになりました。
- **InnoDB** が、locking readステートメントにより、**NOWAIT** オプションおよび **SKIP LOCKED** オプションをサポートするようになりました。
- GIS 関連の関数が改善しました。
- JSON 関数が強化されました。
- 新しい **mariadb-connector-c** パッケージは、**MySQL** と **MariaDB** に共通のクライアントライブラリーを提供します。このライブラリーは、データベースサーバー **MySQL** および **MariaDB** のすべてのバージョンで使用できます。これにより、アプリケーションの同一のビルドから、RHEL 8 にディストリビュートされる **MySQL** サーバーおよび **MariaDB** サーバーのいずれにも接続できます。

さらに、RHEL 8 に同梱されている **MySQL 8.0** サーバーでは、デフォルトの認証プラグインに **mysql\_native\_password** を使用するように設定されています。RHEL 8 のクライアントツールおよびライブラリーは、アップストリームの **MySQL 8.0** バージョンでデフォルトで使用されている **caching\_sha2\_password** メソッドと互換性がないためです。

デフォルトの認証プラグインを **caching\_sha2\_password** に変更するには、**/etc/my.cnf.d/mysql-default-authentication-plugin.cnf** ファイルを次のように変更します。

```
[mysqld]
default_authentication_plugin=caching_sha2_password
```

(BZ#1649891, BZ#1519450, BZ#1631400)

## MariaDB 10.3 への主な変更点

**MariaDB 10.3** では、RHEL 7 に同梱されていたバージョン 5.5 に新機能が数多く追加されました。以下は、主な変更点です。

- 同期マルチクラスターでもある **MariaDB Galera Cluster** が、**MariaDB** の標準に含まれるようになりました。
- **InnoDB** は、**XtraDB** の代わりに、デフォルトのストレージエンジンとして使用されます。
- Common Table Expression
- System-versioned tables
- **FOR** loops
- 非表示の列
- シーケンス
- **InnoDB** のインスタント **ADD COLUMN**
- ストレージエンジンに依存しないカラム圧縮
- 並列レプリケーション
- マルチソースのレプリケーション

さらに、新しい **mariadb-connector-c** パッケージは、**MySQL** と **MariaDB** に共通のクライアントライブラリーを提供します。このライブラリーは、データベースサーバー **MySQL** および **MariaDB** のすべてのバージョンで使用できます。その結果、RHEL 8 にディストリビュートされる **MySQL** サーバーおよび **MariaDB** サーバーのいずれかに構築されるアプリケーションの1つに接続できます。

[「Using MariaDB on Red Hat Enterprise Linux 8」](#) も併せて参照してください。

(BZ#1637034, BZ#1519450)

## 4.8. デスクトップ

### RHEL 8 の GNOME シェル (バージョン 3.28)

Red Hat Enterprise Linux (RHEL) 8 では、GNOME シェルのバージョン 3.28 が利用できます。以下は、主な変更点です。

- GNOME Boxes の新機能
- 新しいオンスクリーンキーボード
- デバイスサポートの拡張 (最も大きな統合は Thunderbolt 3 インターフェース)
- GNOME ソフトウェア、dconf-editor、および GNOME ターミナルの改善

(BZ#1649404)

### Wayland がデフォルトのディスプレイサーバーに

Red Hat Enterprise Linux 8 では、GNOME セッションおよび GNOME Display Manager (GDM) のデフォルトのディスプレイサーバーに、以前の RHEL メジャーバージョンで使用されていた **X.org** に代わり、**Wayland** を使用します。

**X.org** と比べると、**Wayland** では、主に以下の利点と改善点が提供されます。

- より強力なセキュリティーモデル

- 改善されたマルチモニター処理
- 改善されたユーザーインターフェース (UI) スケーリング
- デスクトップでウィンドウ処理を直接制御

以下の機能は、現在利用できない、または期待通りに機能しない状態です。

- **Wayland** では、マルチ GPU の設定が利用できません。
- **Wayland** では、**NVIDIA** バイナリードライバーが有効ではありません。
- **xrandr** ユーティリティーは、解像度、ローテーション、およびレイアウトの処理方法が異なるため、**Wayland** では有効ではありません。画面を操作するその他の **X.org** ユーティリティーも、**Wayland** では機能しません。
- 画面の録画、リモートデスクトップ、およびアクセシビリティは、**Wayland** では正常に機能しない場合があります。
- クリップボードマネージャーは利用できません。
- **Wayland** は、仮想マシンビューアーなどの X11 アプリケーションのキーボードグラブを無視します。
- ゲスト仮想マシン (VM) の **Wayland** には安定性とパフォーマンスに問題があるため、仮想環境では X11 セッションを使用することが推奨されます。

**X.org** GNOME システムを使用していた RHEL 7 システムから RHEL 8 にアップグレードすると、システムでは引き続き **X.org** が使用されます。また、次のグラフィックドライバが使用されている場合は、自動的に **X.org** にフォールバックします。

- **NVIDIA** バイナリードライバー
- **cirrus** ドライバー
- **mga** ドライバー
- **aspeed** ドライバー

**Wayland** の使用は手動で無効にできます。

- **GDM** の **Wayland** を無効にするには、`/etc/gdm/custom.conf` ファイルに **WaylandEnable=false** オプションを設定します。
- GNOME セッションで **Wayland** を無効にするには、ログイン名を入力してから、ログイン画面の歯車メニューで legacy X11 オプションを選択します。

**Wayland** の詳細は <https://wayland.freedesktop.org/> を参照してください。

(BZ#1589678)

### デフォルトでは有効でないリポジトリにある RPM パッケージの検索

デスクトップ用の追加のリポジトリは、デフォルトでは有効になりません。無効化は、対応する **.repo** ファイルの **enabled=0** 行で指定されています。PackageKit を使用して、そのようリポジトリからパッケージをインストールしようとする、アプリケーションが利用できないことを知らせるエラーメッセージが PackageKit により示されます。パッケージを利用できるようにするには、各 **.repo** ファイルの **enabled=0** の行を **enabled=1** に変更します。

(JIRA:RHELPLAN-2878)

## パッケージ管理用の GNOME Software

Red Hat Enterprise Linux 7 のグラフィック環境でパッケージを管理するツール群を提供していた **gnome-packagekit** パッケージを利用できなくなりました。Red Hat Enterprise Linux 8 では、アプリケーションと gnome-shell 拡張機能のインストールと更新を可能にする **GNOME Software** ユーティリティーにより同様の機能が提供されます。**GNOME Software** は、**gnome-software** パッケージで配布されます。

(JIRA:RHELPLAN-3001)

## 4.9. ID 管理

### Directory Server で新しいパスワード構文チェック

この機能拡張により、Directory Server に新しいパスワード構文チェック機能が追加されました。管理者は、たとえば、辞書チェックを有効にし、文字のシーケンスや回文を使用して許可または拒否を行います。したがって、これを有効にすると、Directory Server のパスワードポリシー構文チェックにより、より安全なパスワードが強制されます。

(BZ#1334254)

### Directory Server が提供する内部操作ログサポートが改善

サーバーとクライアントが開始した Directory Server の操作により、バックグラウンドで追加の操作が発生します。以前は、サーバーが、内部操作の **Internal** 接続キーワードのみを記録し、操作 ID は常に **1** に設定されていました。この機能拡張により、Directory Server は実際の接続と操作 ID のログを記録できるようになったため、この操作の原因となったサーバーまたはクライアントの操作に対する内部操作を追跡できるようになりました。

内部操作ログ記録の詳細は、[https://access.redhat.com/documentation/en-us/red\\_hat\\_directory\\_server/11/html-single/administration\\_guide/#logging\\_internal\\_operations](https://access.redhat.com/documentation/en-us/red_hat_directory_server/11/html-single/administration_guide/#logging_internal_operations) を参照してください。

(BZ#1358706)

### tomcatjss ライブラリーが、AIA 拡張のレスポnderを使用した OCSP 確認のサポート

この機能拡張により、**tomcatjss** ライブラリーは、証明書の Authority Information Access (AIA) 拡張のレスポnderを使用した Online Certificate Status Protocol (OCSP) チェックをサポートします。これにより、Red Hat Certificate System の管理者は、AIA 拡張の URL を使用して OCSP チェックを設定できるようになりました。

(BZ#1636564)

### Directory Server に、インスタンスを管理する新しいコマンドラインユーティリティーの導入

Red Hat Directory Server 11.0 では、**dscreate** ユーティリティー、**dsconf** ユーティリティー、および **dsctl** ユーティリティーが導入されました。このユーティリティーは、コマンドラインを使用した Directory Server を簡単に管理します。たとえば、複雑な LDIF ステートメントをサーバーに送信する代わりに、パラメーターを付けてコマンドを使用して機能を設定できるようになりました。

以下は、各ユーティリティーの目的の概要です。

- **dscreate** ユーティリティーを使用して、対話モードまたは INF ファイルを使用して Directory Server インスタンスを新規作成します。INF ファイル形式は、以前のバージョンの Directory Server で使用されていたインストーラーのものとは異なる点に注意してください。

- **dsconf** ユーティリティを使用して、実行時に、Directory Server インスタンスを管理します。たとえば、**dsconf** で以下のことを行います。
  - **cn=config** エントリーを設定します。
  - プラグインを設定します。
  - レプリケーションを設定します。
  - インスタンスをバックアップおよび復元します。
- オフライン時に Directory Server インスタンスを管理する **dsctl** ユーティリティを使用します。たとえば、**dsctl** で以下のことを行います。
  - インスタンスを起動して停止します。
  - サーバーデータベースのインデックスを再作成します。
  - インスタンスをバックアップおよび復元します。

このユーティリティは、Directory Server 10 で非推奨とマークされた Perl およびシェルのスクリプトに代わるものです。このスクリプトは未サポートの **389-ds-base-legacy-tools** パッケージで引き続き利用できます。ただし、Red Hat がサポートするのは、新しいユーティリティを使用した Directory Server の管理に限られます。

LDIF ステートメントを使用した Directory Server の設定には引き続き対応しますが、Red Hat は、ユーティリティの使用を推奨します。

ユーティリティの使用に関する詳細は、[Red Hat Directory Server 11 のドキュメント](#) を参照してください。

(BZ#1693159)

### **pki subsystem-cert-find** コマンドおよび **pki subsystem-cert-show** コマンドが証明書のシリアル番号を表示

この機能強化により、証明書システムの **pki subsystem-cert-find** コマンドおよび **pki subsystem-cert-show** コマンドでは、出力の証明書のシリアル番号が表示されるようになりました。シリアル番号は重要な情報であり、他の複数のコマンドで必要になることがよくあります。これにより、証明書のシリアル番号を識別するのが簡単になりました。

(BZ#1566360)

### 証明書システムで **pki user** コマンドおよび **pki group** コマンドが非推奨に

今回の更新で、証明書システムの **pki user** コマンドおよび **pki group** コマンド、新しい **pki <subsystem>-user** コマンドおよび **pki <subsystem>-group** コマンドに置き換えられました。以前のコマンドも引き続き使用できますが、コマンドが非推奨であるというメッセージと、代替となる新しいコマンドが表示されます。

(BZ#1394069)

### 証明書システムが、システム証明書のオフライン更新をサポート

この機能強化により、管理者はオフライン更新機能を使用して、証明書システムで構成されているシステム証明書を更新できます。システム証明書の期限が切れると、証明書システムが起動できなくなりました。この機能強化により、管理者が期限切れのシステム証明書を置き換える必要なくなりました。

(BZ#1669257)

## 証明書システムが、外部 CA 署名用に SKI 拡張を使用して CSR を作成できるようになる

この機能強化により、証明書システムは、外部認証局 (CA) 署名用の SKI (Subject Key Identifier) 拡張子を持つ証明書署名要求 (CSR) の作成をサポートします。特定の CA は、特定の値で、または CA 公開鍵から派生したこの拡張を必要とします。これにより、管理者は **pkispawn** ユーティリティに渡される設定ファイルの **pki\_req\_ski** パラメーターを使用して、SKI 拡張子を持つ CSR を作成できるようになりました。

(BZ#1656856)

## SSSD によりローカルユーザーがキャッシュに格納され、nss\_sss モジュールにより処理される

RHEL 8 の System Security Services Daemon (SSSD) は、デフォルトで **/etc/passwd** ファイルおよび **/etc/groups** ファイルから、ユーザーとグループにサービスを提供します。**sss** の **nsswitch** モジュールは、**/etc/nsswitch.conf** ファイルのファイルに先行します。

SSSD を通して、ローカルユーザーにサービスを提供する利点は、**nss\_sss** モジュールに **memory-mapped cache** があるため、NSS (Name Service Switch) への各要求でディスクにアクセスしてファイルを開くのと比較して、Name Service Switch (NSS) ルックアップが高速になることでした。これまで、ディスクへのアクセスの高速化は、Name サービスキャッシュデーモン (**nscd**) により行われていました。ただし、SSSD と **nscd** の両方が独自の独立したキャッシュを使用するため、**nscd** と SSSD を並行して使用するのには容易ではありません。そのため、SSSD が、リモートドメイン (LDAP または Active Directory など) のユーザーにもサービスを提供している環境で **nscd** を使用すると、予期しない動作が発生する可能性があります。

今回の更新で、RHEL 8 ではローカルユーザーおよびグループの解決が高速になりました。**root** ユーザーは、SSSD が処理することはないため、SSSD の潜在的なバグが **root** の解決に影響を及ぼすことはありません。また、SSSD が実行していない場合は、問題を回避するために、**nss\_sss** モジュールが **nss\_files** にフォールバックすることでこの状況が適切に処理されます。SSSD を設定する必要はありません。files ドメインは自動的に追加されます。

(JIRA:RHELPLAN-10439)

## KCM が、KEYRING をデフォルトの認証情報キャッシュストレージとして置き換え

RHEL 8 でデフォルトの認証情報キャッシュストレージは、**sssd-kcm** デーモンにより強化されている Kerberos Credential Manager (KCM) です。KCM では、以前使用されていた KEYRING の制限 (名前空間がないためにコンテナ化された環境での使用や、クォータの表示および管理が難しいなど) が解消されています。

今回の更新で、RHEL 8 には、コンテナ環境により適した認証情報キャッシュが含まれ、将来のリリースでより多くの機能を構築するための基盤が提供されます。

(JIRA:RHELPLAN-10440)

## Active Directory ユーザーが Identity Management を管理可能

今回の更新で、RHEL 8 は、Identity Management (IdM) グループのメンバーとして、Active Directory (AD) ユーザーのユーザー ID オーバーライドを追加できるようになりました。ID オーバーライドは、特定の AD ユーザーまたはグループのプロパティが、特定の ID ビュー (この場合はデフォルトの信頼ビュー) 内でどのように見えるべきかを記述するレコードです。

AD ユーザーは、IdM UI のセルフサービス機能 (SSH 鍵のアップロード、個人データの変更など) を使用できるようになりました。AD 管理者は、2 つの異なるアカウントとパスワードがなくても、IdM を完全に管理できます。現在、AD ユーザーは、IdM で選択されている機能を利用できない場合があります。



(JIRA:RHELPLAN-10442)

### sssctl が、IdM ドメインの HBAC ルールレポートを出力

今回の更新で、System Security Services Daemon (SSSD) の **sssctl** ユーティリティーは、Identity Management (IdM) ドメインのアクセス制御レポートを出力できるようになりました。この機能は、規制上の理由から、特定のクライアントマシンにアクセスできるユーザーとグループの一覧を表示するニーズを満たします。IdM クライアントで **sssctl access-report domain\_name** を実行すると、クライアントマシンに適用する IdM ドメインで解析されたホストベースのアクセス制御 (HBAC) ルールのサブセットを表示します。

IdM サポート以外のプロバイダーは、この機能に対応していません。

(JIRA:RHELPLAN-10443)

### Identity Management パッケージがモジュールとして利用可能

RHEL 8 では、Identity Management (IdM) サーバーとクライアントのインストールに必要なパッケージがモジュールとして公開されています。**client** ストリームは、**idm** モジュールのデフォルトのストリームなので、このストリームを有効にしなくてもクライアントのインストールに必要なパッケージをダウンロードできます。

IdM サーバーのモジュールストリームは **DL1** ストリームと呼ばれています。このストリームには、さまざまなタイプの IdM サーバーに対応する複数のプロファイル (server、dns、adtrust、client、および default) が含まれています。**DL1** ストリームの特定のプロファイルにあるパッケージをダウンロードする場合は、ストリームを有効にして、ストリームを介して配信された RPM に切り替え、**yum module install idm:DL1/profile\_name** コマンドを実行します。

(JIRA:RHELPLAN-10438)

### RHEL 8 のセッション記録ソリューションが追加

Red Hat Enterprise Linux 8 (RHEL 8) にセッション記録ソリューションが追加されました。新しい **tlog** パッケージと、それに関連付けられた Web コンソールセッションプレイヤーにより、ユーザー端末セッションを記録および再生できます。記録は、システムセキュリティサービスデーモン (SSSD) サービスを介して、ユーザー別またはユーザーグループ別に設定できます。端末への入出力はすべてキャプチャーされ、テキストベースの形式でシステムジャーナルに保存されます。セキュリティ上の理由から、未加工のパスワードやその他の機密情報を傍受されないように、入力デフォルトでは非アクティブになっています。

このソリューションを、セキュリティが重要なシステムでユーザーセッションの監査に使用できます。セキュリティ違反が発生した場合に、記録されたセッションをフォレンジック分析として見直すことができます。システム管理者は、セッション記録をローカルに設定し、**tlog-play** ユーティリティーを使用して、RHEL 8 Web コンソールインターフェースまたはコマンドラインインターフェースから、結果を表示できるようになりました。

(JIRA:RHELPLAN-1473)

### authselect がユーザー認証の設定を簡略化

今回の更新で、**authconfig** ユーティリティーに代わり、RHEL 8 ホストの設定を簡略化する **authselect** ユーティリティーが導入されました。**authselect** には、システム管理者が PAM 設定変更を簡単に行える、より安全な PAM スタック管理アプローチが含まれています。**authselect** は、パスワード、証明書、スマートカード、フィンガープリントなどの認証方法を設定するのに使用できます。**authselect** では、リモートドメインに参加するのに必要なサービスを設定しないことに注意してください。このタスクは、**realmd**、**ipa-client-install** のような専門のツールにより実行されます。

(JIRA:RHELPLAN-10445)

## SSSD で、複数のスマートカード認証デバイスの中から1つを選択可能

この更新では、スマートカード認証デバイスを選択するために、PKCS#11 URI を設定できるようになりました。

デフォルトでは、SSSD が、スマートカード認証用デバイスを自動的に検出しようとします。複数のデバイスを接続している場合は、最初に見つかったデバイスを SSSD が選択するため、特定のデバイスを選択することはできません。失敗する可能性もあります。

そのため、**sssd.conf** の **[pam]** セクションに、新しい **p11\_uri** オプションを設定できるようになりました。このオプションを使用すると、スマートカード認証に使用するデバイスを定義できます。

たとえば、OpenSC PKCS#11 モジュールにより検出されたスロット ID (2) のリーダーを選択するには、以下を、

```
p11_uri = library-description=OpenSC%20smartcard%20framework;slot-id=2
```

**sssd.conf** の **[pam]** セクションに追加します。

詳細は **man sssd-conf** コマンドで確認してください。

(BZ#1620123)

## 4.10. コンパイラーおよび開発ツール

### Boost がバージョン 1.66 へ更新

Boost C++ ライブラリーが、アップストリームのバージョン 1.66 に更新されました。Red Hat Enterprise Linux 7 に同梱される Boost のバージョンは 1.53 です。詳細は、アップストリームの更新履歴ログ <https://www.boost.org/users/history/> を参照してください。

今回の更新で、以前のバージョンとの互換性を損なう以下の変更が行われています。

- splay コンテナの **bs\_set\_hook()** 関数、**splay\_set\_hook()** 関数と、Intrusive ライブラリーの **splaytree\_algorithms()** 関数の追加パラメーター **bool splay = true** が削除されました。
- JSON ファイル内のコメントまたは文字列の連結が、Property Tree ライブラリーのパーサーでサポートされなくなりました。
- Math ライブラリーのディストリビューションや特別な関数の一部を、ドキュメントに記載されているとおりに動作し、最大の有限値を返さずに **overflow\_error** を送出するように修正しました。
- Math ライブラリーの一部のヘッダーを、**libs/math/include\_private** ディレクトリーに移動しました。
- Regex ライブラリーの **basic\_regex<>::mark\_count()** 関数および **basic\_regex<>::subexpression(n)** 関数の挙動が、ドキュメントに合わせて変更になりました。
- Variant ライブラリーで可変テンプレートを使用すると、メタプログラミング機能が壊れる可能性があります。
- **boost::python::numeric** API が削除され、代わりに **boost::python::numpy** が使用できるようになりました。

- 非オブジェクト型へのポインターに対する算術演算が、Atomic ライブラリーでは提供されなくなりました。

(BZ#1494495)

### Unicode 11.0.0 のサポート

Red Hat Enterprise Linux コア C ライブラリー **glibc** が、Unicode 標準バージョン 11.0.0 に対応するように更新されたため、文字セット間の書き直しおよび変換を含むワイド文字およびマルチバイト文字の API のすべてが、この規格に準拠した正確な情報を提供します。

(BZ#1512004)

### boost が Python から独立

今回の更新で、**boost** パッケージをインストールしても、**Boost.Python** ライブラリーが依存関係としてインストールされなくなりました。**Boost.Python** を使用するには、**boost-python3** パッケージまたは **boost-python3-devel** パッケージをインストールする必要があります。

(BZ#1616244)

### 新しい **compat-libgfortran-48** パッケージが利用可能に

Fortran ライブラリーを使用する Red Hat Enterprise Linux 6 および 7 のアプリケーションとの互換性のために、**libgfortran.so.3** ライブラリーを提供する **compat-libgfortran-48** 互換性パッケージが新たに利用できるようになりました。

(BZ#1607227)

### GCC での Retpoline サポート

今回の更新で、GCC への retpoline のサポートが追加されました。retpoline は、CVE-2017-5715 の説明にあるように、Spectre Variant 2 攻撃を軽減するためのオーバーヘッドを減らすために、カーネルで使用されるソフトウェア構造です。

(BZ#1535774)

### toolchain コンポーネントにおける 64 ビットの ARM アーキテクチャーのサポートの強化

ツールチェーンコンポーネントの **GCC** および **binutils** は、64 ビットの ARM アーキテクチャーの拡張サポートを提供します。以下に例を示します。

- **GCC** と **binutils** が Scalable Vector Extension (SVE) をサポートするようになりました。
- ARM v8.2 で提供されていた **FP16** データ型のサポートが **GCC** に追加されました。**FP16** データ型は特定のアルゴリズムのパフォーマンスを改善します。
- **binutils** のツールは、ポインター認証を含む ARM v8.3 アーキテクチャーの定義をサポートするようになりました。ポインター認証機能により、悪意のあるコードが、独自の関数ポインターを作成してプログラムまたはカーネルの通常の実行を妨害するのを防ぎます。その結果、コード内のさまざまな場所に分岐する際に信頼できるアドレスだけが使用され、セキュリティが向上します。

(BZ#1504980, BZ#1550501, BZ#1504995, BZ#1504993, BZ#1504994)

### IBM POWER システム用 **glibc** の最適化

今回の更新で、IBM POWER 8 と IBM POWER 9 の両アーキテクチャー向けに最適化された新しいバージョンの **glibc** が提供されるようになりました。これにより、IBM POWER 8 システムおよび IBM

POWER 9 システムが、実行時に最適化された適切な **glibc** バリエーションに自動的に切り替わるようになりました。

(BZ#1376834)

## GNU C ライブラリーがバージョン 2.28 に更新

Red Hat Enterprise Linux 8 には、バージョン 2.28 の GNU C ライブラリー (glibc) が同梱されます。以下は、主な改善点です。

- セキュリティーの強化機能:
  - **AT\_SECURE** フラグが付いた安全なバイナリーファイルは、**LD\_LIBRARY\_PATH** 環境変数を無視します。
  - シャットダウンのスピードを上げ、危険にさらされた環境でより多くのコードが実行されることを回避するため、スタックチェックの失敗に対してバックトレースが表示されなくなりました。
- パフォーマンスの向上:
  - スレッドローカルキャッシュにより、**malloc()** 関数のパフォーマンスが向上しました。
  - ライブラリーパフォーマンス特性を変更する **GLIBC\_TUNABLES** 環境変数が追加されました。
  - スレッドセマフォの実装が改善され、スケラブルな **pthread\_rwlock\_xxx()** 関数が新たに追加されました。
  - **math** ライブラリーのパフォーマンスが向上しました。
- Unicode 11.0.0 のサポートが追加されました。
- 標準規格の ISO/IEC/IEEE 60559:2011、IEEE 754-2008、および ISO/IEC TS 18661-3:2015 で定義されている 128 ビット浮動小数点のサポートが向上しました。
- **/etc/resolv.conf** 設定ファイルに関連するドメインネームサービス (DNS) スタブリゾルバーが改善しました。
  - ファイルを変更すると、設定が自動的に再読み込みされるようになりました。
  - 任意の検索ドメインに対応するようになりました。
  - **rotate** オプションの適切なランダム選択が追加されました。
- 次のような、開発用の機能が新たに追加されました。
  - カーネルコールの **preadv2** および **pwritev2** で使用される Linux のラッパー関数
  - 新しい関数 (**reallocarray()**、**explicit\_bzero()** など)
  - **posix\_spawnattr\_setflags()** 関数の新しいフラグ (**POSIX\_SPAWN\_SETSID** など)

(BZ#1512010, BZ#1504125, BZ#506398)

## RHEL で CMake が利用可能に

CMake ビルドシステムバージョン 3.11 は、Red Hat Enterprise Linux 8 の **cmake** パッケージで入手できます。

(BZ#1590139, BZ#1502802)

## make バージョン 4.2.1

Red Hat Enterprise Linux 8 には、**make** ビルドツールバージョン 4.2.1 が同梱されています。以下は、主な変更点です。

- レシピに失敗すると、makefile のファイルとレシピの行番号が表示されます。
- ターゲットの追跡を可能にするために、**--trace** オプションが追加されました。このオプションを使用すると、起動時に、非表示になっているものも含むすべてのレシピが、レシピが置かれているファイル名、行番号、そしてレシピを呼び出すための前提条件とともに出力されます。
- 明示的ルールと暗黙的ルールを混在させても、**make** が実行を終了することはなくなりました。代わりに警告が表示されます。この構文は非推奨になっており、将来、完全に削除される可能性があります。
- テキストをファイルに書き込む **\$(file ...)** 関数が追加されました。テキスト引数なしで呼び出しを行うと、ファイルを開くだけですぐに閉じます。
- 新しいオプション **--output-sync** または **-O** を使用すると、複数のジョブからの出力をジョブごとにグループ化するため、並列ビルドのデバッグを容易にできます。
- **--debug** オプションは、現在有効なデバッグ設定を無効にする **n** (none) フラグも受け入れるようになりました。
- BSD makefile との互換性を高めるために、**\$(shell ...)** 関数の代わりにシェル代入演算子 **!=** が追加されました。詳細と、演算子と関数の違いは、GNU make のマニュアルを参照してください。

これにより、**variable!=value** のように、感嘆符で終わり、その後に代入が続く名前の変数は、新しい構文として解釈されるようになりました。以前の動作に戻すには、**variable! =value** のように、感嘆符の後に空白を追加します。

+

- POSIX 規格で定義されている代入演算子 **::=** が追加されました。
- **.POSIX** 変数を指定している場合、**make** は、バックスラッシュと改行を処理する POSIX 標準規格の要件に従います。このモードでは、バックスラッシュ前の後続スペースが保持され、各バックスラッシュとそれに続く改行文字と空白文字が、空白文字1つに変換されます。
- **MAKEFLAGS** 変数および **MFLAGS** 変数の動作が、より正確に定義されました。
- 新しい変数 **GNUMAKEFLAGS** は、**MAKEFLAGS** と同じように **make** フラグとして解釈されるため、GNU の **make** 固有のフラグを **MAKEFLAGS** 以外に保存でき、makefile の移植性が向上します。
- ホストアーキテクチャーを含む新しい変数 **MAKE\_HOST** が追加されました。
- 新しい変数 **MAKE\_TERMOUT** と **MAKE\_TERMERR** には、それぞれ **make** が標準出力とエラーを端末に書き込んでいいるかどうかを示されます。
- makefile 内の **MAKEFLAGS** 変数に **-r** オプションおよび **-R** オプションを設定すると正しく機能し、すべての組み込み規則と変数がそれぞれ適切に削除されるようになりました。
- **.RECIPEPREFIX** 設定はレシピごとに記録されるようになりました。さらに、そのレシピで展開された変数も、そのレシピの接頭辞設定を使用します。

- **.RECIPEPREFIX** 設定と、ターゲット固有の変数はすべて、コメントとしてではなく、makefile 内にあるように **-p** オプションの出力に表示されます。

(BZ#1641015)

### Go Toolset でビルドされた Go プログラムが FIPS に準拠

ホストシステムを FIPS モードで設定している場合は、**Go Toolset** で利用可能な暗号化ライブラリーが **OpenSSL** ライブラリーバージョン 1.1.0 を使用するように変更されたため、このバージョンの **Go Toolset** でビルドしたプログラムが FIPS に準拠するようになりました。

Go プログラムで、未認証の標準暗号化ルーチンのみが使用されるようにするには、ビルド時に、**go** コンパイラーで **-tags no\_openssl** オプションを使用します。

(BZ#1512570)

### SystemTap バージョン 4.0

Red Hat Enterprise Linux 8 には、**SystemTap** 計測ツールのバージョン 4.0 が同梱されています。以下は、主な変更点です。

- extended Berkeley Packet Filter (eBPF) バックエンド (特に文字列と関数) が改良されました。このバックエンドを使用するには、**SystemTap** を起動する際に **--runtime=bpf** オプションを追加します。
- Prometheus モニタリングシステムで使用するエクスポートネットワークサービスが新たに追加されました。
- システムコールプロービングの実装は、必要に応じてカーネルトレースポイントを使用するように改善されました。

(BZ#1641032)

### binutils バージョン 2.30 の改良

Red Hat Enterprise Linux 8 には、バージョン 2.30 の **binutils** パッケージが同梱されています。以下は、主な変更点です。

- 新しい s390x アーキテクチャー拡張のサポートが改善しました。

アセンブラー:

- WebAssembly ファイル形式のサポート、および WebAssembly から wasm32 ELF ファイル形式への変換サポートが追加されました。
- ARMv8-R アーキテクチャーと、Cortex-R52、Cortex-M23、および Cortex-M33 のプロセッサに対応するようになりました。
- RISC-V アーキテクチャーに対応するようになりました。

リンカー:

- リンカーは、デフォルトで、コードと、読み取り専用データを別々のセグメントに配置するようになりました。その結果、動的ローダーが、読み取り専用データを含むメモリーページの実行を無効にするため、作成される実行可能ファイルは大きくなり、実行の安全性が向上しました。

- バイナリーファイルに関する動的ローダーへのヒントを提供する GNU Property ノートに対応するようになりました。
- 以前は、リンカーが、Indirect Branch Tracking (IBT) テクノロジーに無効な実行可能コードを生成していたため、生成された実行可能ファイルを開始できませんでした。このバグは修正されています。
- 以前は、**gold** リンカーが、プロパティノートを不適切にマージしていたため、生成されたコードで誤ったハードウェア機能が有効になり、コードが突然終了することがありました。このバグは修正されています。
- 以前は、**gold** リンカーが、アーキテクチャーに合わせた調整を行うために、最後にパディングバイトを持つノートセクションを作成していましたが、動的ローダーではパディングが予想されていなかったため、読み込んでいるプログラムが突然終了していました。このバグは修正されています。

その他のツール:

- **readelf** ツールおよび **objdump** ツールに、リンクをたどって別ファイル上のデバッグ情報に移動し、それらの情報を表示するオプションが追加されました。
- 新しい **--inlines** オプションは、**objdump** ツールの既存の **--line-numbers** オプションを拡張したもので、インライン関数のネスト情報を表示します。
- **nm** ツールに、シンボルの名前の後にバージョン情報を表示するオプション **--with-version-strings** が新たに追加されました。

(BZ#1641004, BZ#1637072, BZ#1501420, BZ#1504114, BZ#1614908, BZ#1614920)

### Performance co-pilot バージョン 4.1.3

Red Hat Enterprise Linux 8 には、**Performance co-pilot** (pcp) バージョン 4.1.3 が同梱されています。以下は、主な変更点です。

- **pcp-dstat** ツールに、履歴分析とコンマ区切り値 (CSV) 形式の出力が追加されました。
- ログユーティリティーで、メトリックラベルとヘルプテキストレコードを使用できます。
- **pmdaperfevent** ツールが、低レベルの同時マルチスレッド (SMT) で正しい CPU 番号を報告するようになりました。
- **pmdapostgresql** ツールが PostgreSQL シリーズ 10.x に対応するようになりました。
- **pmdaredis** ツールが Redis シリーズ 5.x に対応するようになりました。
- 動的プロセスフィルタリングと、各プロセスのシステムコール、ucall、および ustat により、**pmdabcc** ツールが強化されました。
- **pmdammv** ツールが、メトリックラベルをエクスポートするようになり、フォーマットのバージョンが 3 に増えました。
- **pmdagfs2** ツールが、追加で glock および glock ホストダーの計測をサポートするようになりました。
- SELinux ポリシーにいくつかの変更が加えられました。

(BZ#1641034)

## メモリー保護キー

今回の更新で、スレッドごとにページ保護フラグの変更を可能にするハードウェア機能が有効になりました。 `pkey_alloc()` 関数、 `pkey_free()` 関数、および `pkey_mprotect()` 関数に、 `glibc` システムコールラッパーが新たに追加されました。さらに、各スレッドの保護フラグへのアクセスを許可するために、 `pkey_set()` 関数および `pkey_get()` 関数が追加されました。

(BZ#1304448)

## elfutils がバージョン 0.174 への更新

Red Hat Enterprise Linux 8 では、 `elfutils` パッケージのバージョン 0.174 が利用できます。以下は、主な変更点です。

- `eu-readelf` ツールでは、変数を示す際に、負の値を、符号がない大きな値のように使用したり、符号がない大きな値を、負の値として使用していたのが修正され、 `eu-readelf` が、定数値型のサイズおよび定数値型のサイズと符号属性を調べて、正しく表示するようになりました。
- CU がない `.debug_line` データを読み込む新しい関数 `dwarf_next_lines()` が `libdw` ライブラリーに追加されました。この関数は、 `dwarf_getsrclines()` 関数および `dwarf_getsrcfiles()` 関数の代わりに使用できます。
- 以前は、セクションが 65280 を超えるファイルは、 `libelf` ライブラリーおよび `libdw` ライブラリー、ならびにそのライブラリーを使用するツールでエラーを発生させる可能性がありました。このバグは修正されています。これにより、ELF ファイルヘッダーで拡張された `shnum` 値および `shstrndx` 値は正しく処理されます。

(BZ#1641007)

## Valgrind がバージョン 3.14 へ更新

Red Hat Enterprise Linux 8 には、Valgrind 実行可能コード分析ツールバージョン 3.14 が同梱されています。以下は、主な変更点です。

- `--keep-debuginfo` オプションが追加され、アンロードされたコードのデバッグ情報を保持できるようになりました。これにより、保存されるスタックトレースには、メモリーからなくなったコードのファイルおよび行の情報が含まれる可能性があります。
- ソースファイルと行番号に基づいた抑制が追加されました。
- `Helgrind` ツールは、 `--delta-stacktrace` オプションにより、完全な履歴のスタックトレースの計算を指定するようになりました。特に、このオプションを `--history-level=full` とともに使用すると、 `Helgrind` のパフォーマンスを最大 25% 向上させることができます。
- Intel および AMD の 64 ビットアーキテクチャーで最適化されたコードで、 `Memcheck` ツールの誤検出の割合が減少しました。定義チェック処理を制御する `--expensive-definedness-checks` は使用できませんが、パフォーマンスに影響を及ぼします。
- Valgrind では、IBM Power Systems (リトルエンディアン) の命令で認識できるものが増えました。
- Valgrind が、IBM Z アーキテクチャーの z13 プロセッサーで、整数および文字列の `vector` 命令を部分的に処理するようになりました。

新しいオプションと既知の制限の詳細は、man ページの `valgrind(1)` を参照してください。

(BZ#1641029, BZ#1501419)



## GDB バージョン 8.2

Red Hat Enterprise Linux 8 には、GDB デバッガーのバージョン 8.2 が同梱されています。以下は、主な変更点です。

- IPv6 プロトコルは、GDB および **gdbserver** によるリモートデバッグに対応しています。
- デバッグ情報なしのデバッグが改善されました。
- GDB ユーザーインターフェースのシンボル補完が、ABI タグや名前空間などの、より構文的な構成を使用することで、より良い提案を提供するように改善されました。
- コマンドが、バックグラウンドで実行できるようになりました。
- Rust プログラミング言語で作成されたプログラムのデバッグが可能になりました。
- C 言語および C++ 言語のデバッグが、**\_Alignof** 演算子、**alignof** 演算子、C++ rvalue 参照、および C99 変数長自動配列のパarser サポートにより改善されました。
- GDB 拡張スクリプトが、Guile スクリプト言語を使用できるようになりました。
- 拡張機能用の Python スクリプト言語インターフェースが改善され、API 機能、フレームデコレーター、フィルター、およびアンワインダーが新たに追加されました。また、GDB 設定の **.debug\_gdb\_scripts** セクションにあるスクリプトも自動的に読み込まれます。
- GDB は、Python バージョン 3 を使用して、プリンター、フレームデコレーター、フィルター、アンワイダーなどのスクリプトを実行するようになりました。
- ARM アーキテクチャーおよび 64 ビットの ARM アーキテクチャーが改善され、Thumb 32 ビット命令およびシステムコール命令を含む、プロセスの実行記録および再生が追加されました。
- Intel MPX レジスターと境界違反、PKU レジスター、および Intel プロセッサトレースのサポートが追加されました。
- 記録と再生の機能で、Intel ベースのシステムの **rdrand** および **rdseed** の命令を含むように拡張されました。
- IBM Z アーキテクチャー上の GDB の機能は、トレースポイントと高速トレースポイント、ベクトルレジスターと ABI、および **Catch** システムコールに対応するように拡張されました。また、GDB が、アーキテクチャーの新しい命令に対応するようになりました。
- GDB は、64 ビット ARM アーキテクチャーで、SystemTap の静的ユーザー領域プローブ (SDT) を使用できるようになりました。

(BZ#1641022, BZ#1497096, BZ#1505346, BZ#1592332)

## RHEL のローライゼーションが複数のパッケージで配布

RHEL 8.0 では、1つの **glibc-common** パッケージでロケールと翻訳が提供されなくなり、すべてのロケールと言語が **glibc-langpack-CODE** パッケージで利用できるようになりました。さらに、デフォルトですべてのロケールがインストールされるわけではなく、インストーラーで選択した言語だけがインストールされます。その他の言語は、必要なロケールパッケージを個別にインストールする必要があります。

言語パッケージの使用方法は「[言語パックのインストールおよび使用](#)」を参照してください。

(BZ#1512009)

## strace がバージョン 4.24 に更新

Red Hat Enterprise Linux 8 には、**strace** ツールバージョン 4.24 が同梱されています。以下は、主な変更点です。

- **-e inject=** オプションで、システムコールの改ざん機能が追加されました。これには、エラー、戻り値、遅延、およびシグナルの挿入が含まれます。
- システムコール修飾が改善されました。
  - 正規表現でシステムコールをフィルタリングするために、**-e trace=/regex** オプションが追加されました。
  - **-e trace=** オプションのシステムコール修飾の前に疑問符を付けると、その修飾がいずれのシステムコールにも一致しなくても **strace** を続行できます。
  - **-e trace** オプションのシステムコール修飾に、パーソナリティーの指定が追加されました。
- 終了理由の **kvm vcpu** デコードが追加されました。これには、**-e kvm=vcpu** オプションを使用します。
- **-k** オプションを使用している場合は、**libdw** ライブラリーがスタックのバックトレースに使用されるようになりました。さらに、**libiberty** ライブラリーがシステムにインストールされていると、シンボルのデマングルが可能になります。
- 以前は、**-r** オプションを指定すると、**strace** が **-t** オプションを無視していました。この問題が修正され、干渉を受けなくなりました。
- 追加モードで出力ファイルを開くための **-A** オプションが追加されました。
- **-X** オプションに **xlat** 出力フォーマットを設定する引数が追加されました。
- **-yy** オプションを使用したソケットアドレスのデコードが改善されました。さらに、**-yy** モードに、ブロックデバイスとキャラクターデバイスの番号出力が追加されました。

さらに、以下の項目のデコードが追加、改善、または更新されました。

- **netlink** プロトコル、メッセージ、および属性
- **arch\_prctl**、**bpf**、**getsockopt**、**io\_pgetevent**、**kern\_features**、**keyctl**、**prctl**、**pkey\_alloc**、**pkey\_free**、**pkey\_mprotect**、**ptrace**、**rseq**、**setsockopt**、**socket**、**statx**、その他のシステムコール
- **ioctl** システムコールの多数のコマンド
- さまざまな型の定数
- システムコールの **execveat**、**inotify\_add\_watch**、**inotify\_init**、**select**、**symlink**、**symlinkat**、および間接引数が付いた **mmap** システムコールのパス追跡
- ARM アーキテクチャー **\_\_ARM\_NR\_\*** 固有のシステムコール
- シグナルコードの一覧

(BZ#1641014)

## 4.11. ファイルシステムおよびストレージ

### XFS が、共有コピーオンライトのデータエクステントをサポート

XFS ファイルシステムは、共有コピーオンライトのデータエクステント機能に対応します。この機能により、2つ以上のファイルで共通のデータブロックセットを共有できます。共通ブロックを共有しているファイルのいずれかが変更すると、XFS は共通ブロックへのリンクを解除して、新しいファイルを作成します。これは、その他のファイルシステムに見られるコピーオンライト (COW) 機能と類似します。

共有コピーオンライトのデータエクステントは次の特徴があります。

#### Fast

共有コピーを作成しても、ディスク I/O を使用しません。

#### Space-efficient

共有ブロックは、追加のディスク容量を使用しません。

#### Transparent

共通ブロックを共有するファイルは、通常のファイルと同じように機能します。

ユーザー領域ユーティリティーは、次の目的で共有コピーオンライトのデータエクステントを使用できます。

- **cp --reflink** コマンドなどを使用した効果的なファイルのクローン作成
- ファイルごとのスナップショット

この機能は、Overlayfs、NFS などのカーネルサブシステムで、より効率的な操作に使用されます。

共有コピーオンライトエクステントは、**xfsprogs** パッケージのバージョン **4.17.0-2.el8** 以降で、XFS ファイルシステムの作成時にデフォルトで有効になります。

直接アクセス (DAX) デバイスは、現在、共有コピーオンライトデータエクステントを使用した XFS をサポートしていません。この機能なしで XFS ファイルシステムを作成するには、次のコマンドを使用します。

```
# mkfs.xfs -m reflink=0 block-device
```

Red Hat Enterprise Linux 7 は、読み取り専用モードでのみ、共有コピーオンラインとデータエクステントを持つ XFS ファイルシステムをマウントできます。

(BZ#1494028)

### XFS ファイルシステムの最大サイズが 1024 TiB に

XFS ファイルシステムで対応可能な最大サイズが 500 TiB から 1024 TiB に増えました。

500 TiB を超えるファイルシステムでは、以下が必要です。

- メタデータの CRC 機能と、フリー inode btree 機能の両方が、ファイルシステムフォーマットで有効になっています。
- 割り当てグループのサイズは、最低 512 GiB です。

RHEL 8 では、**mkfs.xfs** ユーティリティーが、デフォルトでこの要件を満たすファイルシステムを作成します。

この要件を満たしていない小規模なファイルシステムを、500 TiB を超えるサイズに拡張することはサポートしていません。

(BZ#1563617)

## VDO がすべてのアーキテクチャーをサポート

Virtual Data Optimizer (VDO) が、RHEL 8 でサポートされるすべてのアーキテクチャーで利用可能になりました。

対応しているアーキテクチャーの一覧は、[2章 アーキテクチャー](#)を参照してください。

(BZ#1534087)

## BOOM ブートマネージャーが、ブートエントリーを作成するプロセスを簡素化

BOOM は、Linux システム用のブートマネージャーで、ブートエントリー設定の BootLoader 仕様に対応するブートローダを使用します。柔軟なブート設定が可能になり、ブートエントリーの新規作成や変更が容易になります。たとえば、LVM を使用して作成したシステムのスナップショットイメージを起動できます。

BOOM は、既存のブートローダ設定を変更せず、追加エントリーを挿入するだけです。既存の設定は維持され、カーネルのインストールや更新のスクリプトなどのディストリビューションの統合は、以前と同じように機能し続けます。

BOOM には、ブートエントリーを作成するタスクを容易にする、単純化されたコマンドラインインターフェース (CLI) および API があります。

(BZ#1649582)

## LUKS2 が、ボリューム暗号化のデフォルトフォーマットに

RHEL 8 では、LUKS バージョン 2 (LUKS2) 形式が、レガシーの LUKS (LUKS1) 形式に代わり使用されます。**dm-crypt** サブシステムおよび **cryptsetup** ツールでは、暗号化ボリュームのデフォルト形式として LUKS2 が使用されるようになりました。LUKS2 は、部分的なメタデータ破損イベントが発生した場合に備えて、暗号化されたボリュームにメタデータの冗長性と自動回復を提供します。

内部の柔軟なレイアウトにより、LUKS2 は将来の機能も実現します。これは、**libcryptsetup** に組み込まれた一般的なカーネルキーリングトークンにより自動ロック解除に対応し、カーネルキーリング保持サービスに保存されているパズルピースを使用して LUKS2 ボリュームのロックを解除します。

以下は、その他の主な機能強化です。

- ラップ鍵暗号方式を使用した保護鍵の設定
- Policy-Based Decryption (Clevis) とのより簡単な統合
- 最大 32 個の鍵スロット (LUKS1 は鍵スロットを 8 個だけ提供します)

詳細は、man ページの **cryptsetup(8)** および **cryptsetup-reencrypt(8)** を参照してください。

(BZ#1564540)

## Broadcom Emulex ファイバーチャネルのアダプターで NVMe/FC に完全対応

Broadcom Emulex Fibre Channel 32Gbit アダプターとともに使用すると、イニシエーターモードで NVMe/FC (NVMe over Fibre Channel) トランスポートタイプに完全に対応するようになりました。

Red Hat Enterprise Linux に同梱されていた RDMA (Remote Direct Memory Access) プロトコルに加えて、NVMe over Fibre Channel が、NVMe (Nonvolatile Memory Express) プロトコルのファブリックトランスポートタイプとして追加されました。

**lpfc** ドライバーで NVMe/FC を有効にするには、`/etc/modprobe.d/lpfc.conf` ファイルに以下のオプションを追加します。

```
lpfc_enable_fc4_type=3
```

**lpfc** 以外のドライバーはテクノロジープレビューのままになります。

その他の制限:

- マルチパスは、NVMe/FC ではサポートされません。
- NVMe クラスタリングは、NVMe/FC ではサポートされません。
- 現在、Red Hat Enterprise Linux は、イニシエーターモードで NVMe/FC および SCSI/FC の同時使用をサポートしていません。
- **kernel-alt** パッケージは、NVMe/FC をサポートしません。
- **kdump** は、NVMe/FC ではサポートされません。
- SAN (Storage Area Network) の NVMe/FC からのシステム起動はサポートされません。

(BZ#1649497)

### DM Multipath 設定ファイルの **overrides** セクションが追加

`/etc/multipath.conf` ファイルには、全デバイスの設定値を設定できる **overrides** セクションが含まれています。この属性は、デバイスを含むパスの `/etc/multipath.conf` ファイルの **multipaths** セクションに指定された属性により上書きされた場合を除き、DM Multipath によりすべてのデバイスに使用されます。この機能は、設定ファイルの **devices** セクションの **all\_devs** パラメーターに代わるものです。

(BZ#1643294)

### NVDIMM デバイスからのインストールおよび起動に対応

以前は、どのモードでも、NVDIMM (Nonvolatile Dual Inline Memory Module) デバイスがインストーラーから無視されていました。

今回の更新で、NVDIMM デバイスをサポートするカーネル実装により、システムパフォーマンスの機能が改善し、データベースまたは解析のワークロードなど、書き込みが集中するアプリケーションへのファイルシステムのアクセスが強化され、CPU オーバーヘッドが削減しました。

今回の更新で、以下のサポートが導入されました。

- Kickstart の **nvdimm** コマンドおよび GUI を使用したインストールに NVDIMM デバイスを使用。セクターモードの NVDIMM デバイスからインストールおよび起動して、インストール時にセクターモードに NVDIMM デバイスに再設定できます。
- NVDIMM デバイスを扱うコマンドで、Anaconda の **Kickstart** スクリプトを拡張。
- システムコンポーネント **grub2**、**efibootmgr**、および **efivar** が、NVDIMM デバイスを処理して起動する能力。

(BZ#1499442)

## DM Multipath のマージナルパスの検出が改善

**multipathd** サービスでは、マージナルパスの検出が改善しました。これにより、マルチパスデバイスが、繰り返し失敗する可能性があるパスを回避して、パフォーマンスを向上します。マージナルパスは、永続的で断続的な I/O エラーがあるパスです。

マージナルパスの動作は、`/etc/multipath.conf` ファイルの以下のオプションで制御します。

- **marginal\_path\_double\_failed\_time**
- **marginal\_path\_err\_sample\_time**
- **marginal\_path\_err\_rate\_threshold**
- **marginal\_path\_err\_recheck\_gap\_time**

以下の場合、DM Multipath はパスを無効にし、サンプル期間中に I/O を繰り返しテストします。

- **multipath.conf** オプションが設定されている
- 設定した期間内にパスが 2 回失敗する
- その他のパスが利用できる

このテスト中に、パスのエラーが設定されたエラー率よりも大きい場合に、設定したギャップ時間の間、DM Multipath はパスを無視してから、それが復旧できるぐらいに適切に機能しているかどうかを再確認します。

詳細は、man ページの **multipath.conf** を参照してください。

(BZ#1643550)

## マルチキューのデフォルト動作

Red Hat Enterprise Linux 8 では、ブロックデバイスがマルチキュースケジューリングを使用するようになりました。これにより、高速ソリッドステートドライブ (SSD) およびマルチコアシステムでの拡張が向上します。

SCSI マルチキュー (**scsi-mq**) ドライバーがデフォルトで有効になり、カーネルが **scsi\_mod.use\_blk\_mq=Y** オプションで起動します。この変更は、アップストリームの Linux カーネルと同じです。

デバイスマッパーマルチパス (DM Multipath) を使用するには、**scsi-mq** ドライバーがアクティブになっている必要があります。

(BZ#1647612)

## Stratis が利用可能に

Stratis は、ユーザーへの追加機能を備えたストレージプールに、管理されるファイルシステムを提供する新しいローカルストレージマネージャーです。

Stratis を使用すると、次のようなストレージタスクをより簡単に実行できます。

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

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

詳細は、Stratis のドキュメント「[Managing layered local storage with Stratis](#)」を参照してください。

(JIRA:RHELPLAN-1212)

## 4.12. 高可用性およびクラスター

### 新しい **pcs** コマンドで、利用可能なウォッチドッグデバイスの一覧を表示し、ウォッチドッグデバイスをテスト

Pacemaker を使用して SBD を設定するために、動作するウォッチドッグデバイスが必要です。このリリースでは、ローカルノードで利用可能なウォッチドッグデバイスを一覧表示する **pcs stonith sbd watchdog list** コマンドと、ウォッチドッグデバイスをテストする **pcs stonith sbd watchdog test** コマンドがサポートされます。**sbd** コマンドラインツールの詳細は、man ページの **sbd(8)** を参照してください。

(BZ#1578891)

### **pcs** コマンドで、オペレーションと、その間隔によるリソースのフィルタリングの不具合をサポート

Pacemaker は、リソース名およびノードのほかに、リソース操作別にリソースの不具合を追跡するようになりました。したがって、**pcs resource failcount show** コマンドでは、不具合のフィルタリングに、リソース、ノード、操作、および間隔を選択できます。これにより、リソースおよびノード別、もしくはリソース、ノード、操作、および間隔別に細かく集計した不具合を表示するオプションを提供します。また、**pcs resource failcount reset** コマンドを使用すれば、リソース、ノード、操作、および間隔別に不具合をフィルタリングできます。

(BZ#1591308)

### **corosync** ログでタイムスタンプが有効になっている

以前は、**corosync** ログにタイムスタンプが含まれていなかったため、他のノードやデーモンからのログに関連付けることは困難でした。このリリースでは、タイムスタンプが **corosync** ログに記録されます。

(BZ#1615420)

### コマンド **pcs cluster setup**、**pcs cluster node add**、**pcs cluster node remove** の新しい形式

Red Hat Enterprise Linux 8 の **pcs** は、Corosync 3、**knet**、およびノード名を完全にサポートします。ノード名は必須となり、ノード識別子となるノードアドレスを置き換えます。ノードアドレスは任意となりました。

- **pcs host auth** コマンドで、ノードアドレスがデフォルトでノード名となります。
- **pcs cluster setup** コマンドおよび **pcs cluster node add** コマンドでは、ノードアドレスが、**pcs host auth** コマンドで指定したノードアドレスにデフォルト設定されます。

この変更により、クラスターのセットアップ、クラスターへのノードの追加、およびクラスターからのノードの削除を行うコマンドの形式が変更になりました。新しいコマンド形式の詳細は、**pcs cluster setup** コマンド、**pcs cluster node add** コマンド、および **pcs cluster node remove** コマンドのヘルプ表示を参照してください。

(BZ#1158816)

## RHEL 8 の Pacemaker 2.0.0

**pacemaker** パッケージがアップストリームバージョンの Pacemaker 2.0.0 にアップグレードし、以前のバージョンに対するバグ修正および機能拡張が数多く追加されました。

- Pacemaker の詳細ログが、(**/var/log** に直接置かれるのではなく、**/var/log/cluster** にある **corosync** ログを組み合わせる) デフォルトで **/var/log/pacemaker/pacemaker.log** になりました。
- ログの読み取りをより直観的にするために、Pacemaker デーモンプロセスの名前が変更になりました。たとえば、**pengine** の名前は **pacemaker-schedulerd** に変更になりました。
- 非推奨のクラスタープロパティ **default-resource-stickiness** および **is-managed-default** のサポートが終了しました。代わりに、**resource-stickiness** プロパティおよび **is-managed** プロパティをリソースのデフォルトに設定する必要があります。非推奨の構文を持つ既存の構成 (新たに作成されたものではありません) は自動的に更新され、対応している構文を使用するようになります。
- 変更点の詳細は「[Pacemaker 2.0 upgrade in Red Hat Enterprise Linux 8](#)」を参照してください。

Red Hat Enterprise Linux 7 以前を使用して既存のクラスターをアップグレードするユーザーは、すべてのクラスターノードで RHEL をアップグレードする前と後に、任意のクラスターノードで **pcs cluster cib-upgrade** を実行することが推奨されます。

(BZ#1543494)

### マスターリソースから、昇格可能なクローンリソースに名前が変更

Red Hat Enterprise Linux (RHEL) 8 は Pacemaker 2.0 に対応しています。ここでは、マスターとスレーブのリソースが、別の種類のリソースではなくなり、**Promotable** メタ属性が **true** に設定されている標準クローンリソースになりました。今回の更新に対応するために、以下の変更が行われました。

- **pcs** コマンドでマスターリソースを作成することができなくなりました。代わりに、**昇格可能なクローンリソース**を作成できます。関連キーワードおよびコマンドが、**master** から **promotable** に変わりました。
- 既存のすべてのマスターリソースが昇格可能なクローンリソースとして表示されます。
- Web UI で RHEL7 クラスターを管理すると、RHEL7 クラスターは昇格可能なクローンをサポートしないため、マスターリソースは引き続きマスターと呼ばれます。

(BZ#1542288)

### クラスターのノードを認証する新しいコマンド

Red Hat Enterprise Linux (RHEL) 8 では、クラスターでノードの認証に使用されるコマンドに以下の変更が加えられています。

- 認証用の新しいコマンドは **pcs host auth** です。このコマンドにより、ユーザーは、ホスト名、アドレス、および **pcsd** ポートを指定できます。
- **pcs cluster auth** コマンドは、ローカルクラスターのノードのみを認証し、ノードリストは許可しません、
- 各ノードにアドレスを指定できるようになりました。**pcs/pcsd** は、指定したアドレスを使用して各ノードと通信します。このアドレスは、**corosync** が内部で使用しているアドレスとは異なる場合があります。



- **pcs pcsd clear-auth** コマンドは、**pcs pcsd deauth** コマンドおよび **pcs host deauth** コマンドに置き換えられました。この新しいコマンドを使用すると、ユーザーはすべてのホストの認証を解除するだけでなく、1台のホストでも認証解除できます。
- 以前は、ノード認証が双方向で、**pcs cluster auth** コマンドを実行すると、指定したすべてのノードが互いに認証されていましたが、**pcs host auth** コマンドを使用すると、指定したノードに対してローカルホストだけが認証されます。これにより、このコマンドを実行する際に、どのノードが他のどのノードに対して認証されているかをより適切に制御できます。クラスターを設定する場合や、ノードを追加する場合に、**pcs** がクラスターのトークンを自動的に同期するため、クラスター内のすべてのノードが以前と同じように自動的に認証され、クラスターノードは互いに通信できます。

これらの変更は後方互換性がないことに注意してください。RHEL 7 システムで認証されたノードは、再認証する必要があります。

(BZ#1549535)

### pcs コマンドで、フェンシング履歴の表示、クリーンアップ、および同期がサポート

Pacemaker の fence デーモンは、実行したすべての fence アクションの履歴 (保留、成功、および失敗) を追跡します。このリリースでは、**pcs** コマンドを使用して、ユーザーは次の方法でフェンシング履歴にアクセスできます。

- **pcs status** コマンドは、失敗した保留中のフェンシングアクションを表示します。
- **pcs status --full** コマンドは、フェンシング履歴全体を表示します。
- **pcs stonith history** コマンドは、フェンシング履歴を表示してクリーンアップするオプションを提供します。
- フェンシング履歴は自動的に同期されますが、**pcs stonith history** コマンドで **update** オプションがサポートされるようになり、必要に応じてフェンシング履歴を手動で同期できるようになりました。

(BZ#1620190, BZ#1615891)

## 4.13. ネットワーク

**nftables** が、**iptables** を、デフォルトのネットワークパケットフィルタリングのフレームワークとして置き換え

**nftables** フレームワークは、パケットの分離機能を提供し、**iptables** ツール、**ip6tables** ツール、**arptables** ツール、および **ebtables** ツールの後継となります。利便性、機能、パフォーマンスにおいて、以前のネットワークフィルタリングツールに多くの改良が追加されました。以下に例を示します。

- 線形処理の代わりにルックアップテーブル
- **IPv4** プロトコルおよび **IPv6** プロトコルの両方に対する単一のフレームワーク
- 完全ルールセットのフェッチ、更新、および保存を行わず、すべてアトミックに適用されるルール
- ルールセットにおけるデバッグおよびトレースのサポート (**nfttrace**) およびトレースイベントの監視 (**nft** ツール)
- より一貫性のあるコンパクトな構文、プロトコル固有の拡張なし

- サードパーティーのアプリケーション用 Netlink API

**iptables** と同様、**nftables** は、チェーンを保存するテーブルを使用します。このチェーンには、アクションを実行する個々のルールが含まれます。**nft** ツールは、以前のパケットフィルタリングフレームワークのツールをすべて置き換えます。**libnftables** ライブラリーは、**libmnl** ライブラリーの **nftables** Netlink API で、低レベルの対話のために使用できます。

**iptables** ツール、**ip6tables** ツール、**ebtables** ツール、および **arptables** ツールは、**nftables** ベースの同じ名前のドロップインツールに置き換えられました。外部の挙動は従来のもと同じですが、内部的には必要に応じて互換インターフェースを通して、従来の **netfilter** カーネルモジュールを使用した **nftables** を使用します。

**nftables** ルールセットに対するモジュールの効果は、**nft list ruleset** コマンドを使用して確認できます。これらのツールは、テーブル、チェーン、およびルールを **nftables** ルールセットに追加するため、**nft flush ruleset** などの **nftables** ルールセット操作は、先に別々のコマンドを使用してインストールしたルールセットに影響を及ぼす可能性があることに注意してください。

どの種類のツールが存在するかをすばやく識別するために、バックエンド名を追加するようにバージョン情報が更新されました。RHEL 8 では、**nftables** ベースの **iptables** ツールで、次のバージョン文字列が出力されます。

```
$ iptables --version
iptables v1.8.0 (nf_tables)
```

一方、従来の **iptables** ツールが存在する場合は、次のバージョン情報が出力されます。

```
$ iptables --version
iptables v1.8.0 (legacy)
```

(BZ#1644030)

## RHEL 8 における TCP の新機能

Red Hat Enterprise Linux 8 には、TCP ネットワーキングスタックバージョン 4.16 が同梱され、より高いパフォーマンスおよび安定性と、より優れたスケラビリティが提供されます。特に、入力接続率が高い、ビジー状態の TCP サーバーのパフォーマンスが向上します。

また、2つの新しい TCP 輻輳制御アルゴリズム (**BBR** および **NV**) が利用でき、ほとんどのシナリオで、キュービクよりも待ち時間が少なく、スループットも良くなります。

(BZ#1562998)

## firewalld がデフォルトで nftables を使用

今回の更新で、**nftables** フィルタリングサブシステムが、**firewalld** デーモンのデフォルトのファイアウォールバックエンドになります。バックエンドを変更するには、**/etc/firewalld.conf** ファイルの **FirewallBackend** オプションを使用します。

この変更により、**nftables** を使用した時の動作に以下のような差異が生じます。

1. **iptables** ルールの実行は、常に **firewalld** ルールの前に行われます。
  - **iptables** の **DROP** は、**firewalld** からパケットを隠すことを意味します。
  - **iptables** の **ACCEPT** は、パケットが引き続き **firewalld** ルールに準拠することを意味しません。

2. 他の **firewalld** 機能は **nftables** を使用しますが、**firewalld** ダイレクトルールは、**iptables** を介して実装されます。
3. 直接ルールの実行は、確立された接続の **firewalld** 一般的な承認の前に行われます。

(BZ#1509026)

### RHEL 8 の **wpa\_supplicant** への主な変更点

Red Hat Enterprise Linux (RHEL) 8 の **wpa\_supplicant** パッケージは、**CONFIG\_DEBUG\_SYSLOG** が有効になった状態で構築されています。これにより、**/var/log/wpa\_supplicant.log** ファイルの内容を確認する代わりに、**journalctl** ユーティリティを使用して **wpa\_supplicant** ログを読み取ることができます。

(BZ#1582538)

### NetworkManager が SR-IOV 仮想ファクションのサポート

Red Hat Enterprise Linux 8.0 では、**NetworkManager** で、SR-IOV (Single Root I/O virtualization) に対応するインターフェースに対する仮想ファクション (VF) の数を設定できます。また、**NetworkManager** では、MAC アドレス、VLAN、**spoof checking** 設定、許可されるビットレートなどの、VF の属性の一部を設定できます。SR-IOV に関連するすべてのプロパティーは、**sriov** 接続設定で利用できます。詳細は、man ページの **nm-settings(5)** を参照してください。

(BZ#1555013)

### IPVLAN 仮想ネットワークドライバーをサポート

Red Hat Enterprise Linux 8.0 のカーネルには、IPVLAN 仮想ネットワークドライバーのサポートが含まれます。今回の更新で、IPVLAN の仮想ネットワークインターフェースカード (NIC) により、ローカルネットワークに1つの MAC アドレスを公開する複数のコンテナに対するネットワーク接続が可能になります。これにより、ピアネットワーク機器でサポートされている MAC アドレス数の制限を改善するために、1つのホストで多数のコンテナを使用できるようになります。

(BZ#1261167)

### NetworkManager が、接続に対するワイルドカードインターフェースの名前一致をサポート

以前は、インターフェース名の完全一致のみを使用して、特定のインターフェースへの接続を制限していました。今回の更新で、接続には、ワイルドカードに対応する新しい **match.interface-name** プロパティーが含まれるようになり、ワイルドカードパターンを使用した柔軟な方法で、接続用インターフェースを選択できるようになります。

(BZ#1555012)

### Networking Stack 4.18 の改良

Red Hat Enterprise Linux 8.0 は、アップストリームバージョン 4.18 にアップグレードされたネットワークスタックが含まれています。いくつかのバグ修正と機能拡張が提供されます。以下は、主な変更点です。

- **UDP\_GSO** などの新しいオフロード機能、および一部のデバイスドライバーに **GRO\_HW** が導入されました。
- ユーザーデータグラムプロトコル (UDP) のスケーラビリティが大幅に向上しました。
- 汎用ビジーポーリングコードを改善しました。
- IPv6 プロトコルのスケーラビリティが改善しました。

- ルーティングコードのスケラビリティが改善しました。
- 新しいデフォルト送信キュースケジューリングアルゴリズム **fq\_codel** が追加されました。転送遅延が改善されます。
- 一部の送信キュースケジューリングアルゴリズムのスケラビリティが向上しました。たとえば、**pfifo\_fast** がロックされなくなりました。

(BZ#1562987)

### iptables を nftables に変換する新しいツール

今回の更新で、既存の **iptables** ルールまたは **ip6tables** ルールを、**nftables** で同等のルールに変換する **iptables-translate** ツールおよび **ip6tables-translate** ツールが追加されました。拡張機能によっては対応する機能がない場合もあります。対応する機能がない拡張機能が存在する場合は、ツールにより、その前に **#** 記号が付いた未変換ルールが出力されます。以下に例を示します。

```
| % iptables-translate -A INPUT -j CHECKSUM --checksum-fill
| nft # -A INPUT -j CHECKSUM --checksum-fill
```

さらに、ユーザーは、**iptables-restore-translate** ツールおよび **ip6tables-restore-translate** ツールを使用して、ルールのダンプを変換できます。その前に、**iptables-save** コマンドまたは **ip6tables-save** コマンドを使用して、たとえば、現在のルールのダンプを出力できます。

```
| % sudo iptables-save >/tmp/iptables.dump
| % iptables-restore-translate -f /tmp/iptables.dump
| # Translated by iptables-restore-translate v1.8.0 on Wed Oct 17 17:00:13 2018
| add table ip nat
| ...
```

(BZ#1564596)

### NetworkManager による VPN に追加された新機能

Red Hat Enterprise Linux 8.0 の **NetworkManager** は、VPN に以下の新機能を提供します。

- インターネット鍵交換バージョン 2 (IKEv2) プロトコルのサポート
- **rightid**、**leftcert**、**narrowing**、**rekey**、**fragmentation** などの **Libreswan** オプションが追加されました。サポートされているオプションの詳細は、man ページの **nm-settings-libreswan** を参照してください。
- デフォルトの暗号を更新しました。つまり、暗号化方式を指定しないと、**NetworkManager-libreswan** プラグインにより、**Libreswan** アプリケーションがシステムのデフォルト暗号を選択します。ただし、ユーザーが IKEv1 アグレッシブモード設定を選択した場合は異なります。この場合は、**ike = aes256-sha1;modp1536** 値および **eps = aes256-sha1** 値が **Libreswan** に渡されます。

(BZ#1557035)

### 新しいデータチャンクタイプ I-DATA が SCTP に追加

今回の更新で、新しいデータチャンクタイプ (**I-DATA**)、およびストリームスケジューラーが SCTP (Stream Control Transmission Protocol) に追加されました。以前は、ユーザーが送信する順序で SCTP がユーザーメッセージを送信していたため、SCTP ユーザーメッセージが大きくなると、送信が完了するまで、ストリーム内の他のすべてのメッセージがブロックされていました。**I-DATA** チャンクを使用すると、TSN (Transmission Sequence Number) フィールドがオーバーロードされなくなるため、SCTP

ではさまざまな方法でストリームをスケジュールできるようになり、**I-DATA** ではユーザーメッセージのインターリーブ (RFC 8260) が可能になりました。両方のピアが **I-DATA** チャンクタイプをサポートしなければならないことに注意してください。

(BZ#1273139)

## 4.14. セキュリティー

### SCAP セキュリティーガイド PCI-DSS プロファイルがバージョン 3.2.1 に準拠

**SCAP セキュリティーガイド** プロジェクトが、Red Hat Enterprise Linux 8 の PCI-DSS (Payment Card Industry Data Security Standard) プロファイルを提供し、最新の PCI-DSS バージョン (3.2.1) に合わせて更新されました。

(BZ#1618528)

### OpenSSH がバージョン 7.8p1 にリベース

**openssh** パッケージが、アップストリームバージョン 7.8p1 にアップグレードされました。以下は、主な変更点です。

- **SSH version 1** プロトコルのサポートがなくなりました。
- **hmac-ripemd160** メッセージ認証コードのサポートがなくなりました。
- **RC4 (arcfour)** 暗号のサポートがなくなりました。
- **Blowfish** 暗号のサポートがなくなりました。
- **CAST** 暗号のサポートがなくなりました。
- **UseDNS** オプションのデフォルト値を **no** に変更しました。
- **DSA** 公開鍵アルゴリズムをデフォルトで無効にしました。
- **Diffie-Hellman** パラメーターの最小モジュールサイズを 2048 ビットに変更しました。
- **ExposeAuthInfo** 設定オプションのセマンティクスを変更しました。
- **UsePrivilegeSeparation=sandbox** オプションは必須になったため、無効にできなくなりました。
- 許可される最小の **RSA** キーサイズを 1024 ビットに設定します。

(BZ#1622511)

### RSA-PSS が OpenSC でサポート

今回の更新で、**OpenSC** スマートカードドライバーに、RSA-PSS 暗号化署名スキームのサポートが追加されました。新しいスキームでは、クライアントソフトウェアの TLS 1.3 サポートに必要な安全な暗号化アルゴリズムが有効になります。

(BZ#1595626)

### RHEL 8 の rsyslog への主な変更点

**rsyslog** パッケージがアップストリームバージョン 8.37.0 にアップグレードし、以前のバージョンに対するバグ修正および機能拡張が数多く追加されました。以下は、主な変更点です。

- **rsyslog** の内部メッセージの処理が強化されました。レートを制限できるようになり、デッドロックの可能性が修正されました。
- 一般的に強化されたレート制限。実際の **スパムの送信元** がログに記録されました。
- 特大サイズのメッセージ処理が改善されました。ユーザーは、コアと特定モジュールの両方で、別々の操作でメッセージを処理する方法を設定できるようになりました。
- 個別のファイルを作成する代わりに、**mmnormalize** ルールベースが **config** ファイルに組み込むことができるようになりました。
- ユーザーは、**imtcp** に **GnuTLS** 優先度文字列を設定して、暗号化をきめ細かく制御できるようになりました。
- JSON の変数を含むすべての **config** 変数は、大文字と小文字を区別しません。
- PostgreSQL への出力にさまざまな改善が追加されました。
- 追加の設定ファイルの条件付き読み込み、ステートメントの実行、**config** へのテキストの挿入など、**config** 処理を制御するシェル変数を使用できるようになりました。この機能を過度に使用すると、**rsyslog** での問題のデバッグが非常に困難になる可能性があることに注意してください。
- **config** で、4 桁のファイル作成モードを指定できるようになりました。
- RELP (Reliable Event Logging Protocol) の入力は、指定したアドレスにのみバインドできるようになりました。
- メール出力の **enable.body** オプションのデフォルト値が、ドキュメントに合わせて変更になりました。
- ユーザーは、**MongoDB** 出力で無視する必要がある挿入エラーコードを指定できるようになりました。
- Parallel TCP (pTCP) 入力には、より良い負荷分散に、設定できないバックログがあります。

(BZ#1613880)

### 新しい **rsyslog** モジュール: **omkafka**

**kafka** 集中型データストレージシナリオを有効にするには、新しい **omkafka** モジュールを使用して、**kafka** インフラストラクチャーにログを転送できるようになりました。

(BZ#1542497)

### **libssh** が SSH をコア暗号化コンポーネントとして実装

この変更により、Red Hat Enterprise Linux 8 のコア暗号化コンポーネントとして **libssh** が導入されました。**libssh** ライブラリーは、Secure SHell (SSH) プロトコルを実装しています。

**libssh** は、システム全体の暗号化ポリシーに準拠していません。

(BZ#1485241)

### スマートカードおよび HSM の PKCS #11 がシステム全体で統一

今回の更新で、PKCS #11 暗号化トークンインターフェースを使用するスマートカードおよび HSM (Hardware Security Modules) の使用方法が統一されました。つまり、ユーザーおよび管理者は、システムに関連するすべてのツールに同じ構文を使用できます。以下は、主な機能強化です。

- 管理者とアプリケーション作成者に向けた、RHEL サーバーで、トークンの簡略化された使用可能性を保証する PKCS #11 Uniform Resource Identifier (URI) スキームのサポート。
- システム全体で、スマートカードと HSM の登録に **pkcs11.conf** `pkcs11.conf` を使用します。
- NSS、GnuTLS、および OpenSSL (**openssl-pkcs11** エンジンを使用) のアプリケーションで利用できる、HSM およびスマートカードに対する一貫したサポート。
- Apache HTTP サーバー (**httpd**) が、HSM をシームレスに対応。

詳細は、man ページの **pkcs11.conf(5)** を参照してください。

(BZ#1516741)

### システム全体の暗号化ポリシーがデフォルトで適用

暗号ポリシーは、Red Hat Enterprise Linux 8 でコア暗号化サブシステムを構成するコンポーネントで、TLS、IPSec、SSH、DNSSec、および Kerberos の各プロトコルに対応します。これにより、管理者が **update-crypto-policies** コマンドで選択できる小規模セットのポリシーを提供します。

**デフォルト** のシステム全体の暗号化ポリシーは、現在の脅威モデルに対して安全な設定を提供します。これは、TLS 1.2 と 1.3 のプロトコルだけでなく、IKEv2 プロトコルと SSH2 プロトコルを可能にします。RSA 鍵と Diffie-Hellman パラメーターの長さが 2047 ビット以上であれば許可されます。

詳細は、Red Hat ブログの記事「[Consistent security by crypto policies in Red Hat Enterprise Linux 8](#)」と、man ページの **update-crypto-policies(8)** を参照してください。

(BZ#1591620)

### SCAP セキュリティーガイドが OSPP 4.2 に対応

**SCAP セキュリティーガイド**では、Red Hat Enterprise Linux 8 用の OSPP (Protection Profile for General Purpose Operating Systems) プロファイルバージョン 4.2 のドラフトが提供されています。このプロファイルは、汎用オペレーティングシステムの保護プロファイル (プロテクションプロファイルバージョン 4.2) の NIAP 設定付録に記載されている必須の設定管理を反映しています。SCAP セキュリティーガイドは、OSPP に定義したユーザーが満たすことを可能にする自動化されたチェックとスクリプトを提供します。

(BZ#1618518)

### OpenSCAP コマンドラインインターフェースが改善

**oscap** モジュールおよびサブモジュールのすべてで詳細モードが利用できるようになりました。ツールの出力で形式が改善しました。

コマンドラインインターフェースの使いやすさを向上させるために、廃止予定のオプションが削除されました。

次のオプションは使用できなくなりました。

- **oscap xccdf generate report** の **--show** は完全に削除されました。
- **oscap oval eval** の **--probe-root** は削除されました。環境変数 **OSCAP\_PROBE\_ROOT** を設定することで置き換えることができます。

- **oscap xccdf eval** の **--sce-results** は、**--check-engine-results** に置き換えられました。
- CPE、OVAL、XCCDF の各モジュールから、**validate-xml** サブモジュールが削除されました。代わりに、XML スキーマおよび XSD スキマトロンに対して SCAP コンテンツを検証するのに **validate** サブモジュールを使用できます。
- **oscap oval list-probes** コマンドが削除されました。代わりに、**oscap --version** を使用して、利用可能なプローブの一覧を表示できます。

OpenSCAP では、**--profile '(all)'** を使用して、プロファイルに関係なく、特定の XCCDF ベンチマーク上のすべてのルールを評価できます。

(BZ#1618484)

### mmap システムコールに対する新しいマップパーミッションチェックのサポート

SELinux の **map** パーミッションは、ファイル、ディレクトリー、ソケットなどへのメモリーマップアクセスを制御するために追加されました。これにより、SELinux ポリシーが、さまざまなファイルシステムオブジェクトへの直接メモリーアクセスを防ぎ、そのようなすべてのアクセスが再検証されるようになります。

(BZ#1592244)

### SELinux が systemd No New Privileges をサポート

今回の更新で、新旧のコンテキスト間で **nnp\_nosuid\_transition** が許可されている場合に、**No New Privileges** (NNP) または **nosuid** 下で SELinux ドメインの移行を可能にする **nnp\_nosuid\_transition** ポリシー機能が導入されました。**selinux-policy** パッケージに、**NNP** セキュリティー機能を使用する **systemd** サービスのポリシーが含まれるようになりました。

次のルールは、サービスにこの機能を許可することを示しています。

```
allow source_domain target_type:process2 { nnp_transition nosuid_transition };
```

以下に例を示します。

```
allow init_t fprintd_t:process2 { nnp_transition nosuid_transition };
```

ディストリビューションポリシーには、m4 マクロインターフェースも含まれています。これは、**init\_nnp\_daemon\_domain()** 関数を使用するサービスの SELinux セキュリティーポリシーで使用できます。

(BZ#1594111)

### SELinux が process クラスの getrlimit アクセス権をサポート

今回の更新で、新しい SELinux アクセス制御チェック (**process:getrlimit**) が導入されました。これは、**prlimit()** 関数に追加されました。これにより、SELinux ポリシー開発者は、**process:setrlimit** パーミッションを使用して、あるプロセスが、別のプロセスのリソース制限を読み込み、変更するかどうかを制御できます。SELinux は、プロセスが **prlimit()** を介して、自身のリソース制限を操作することは制限しないことに注意してください。詳細は man ページの **prlimit(2)** および **getrlimit(2)** を参照してください。

(BZ#1549772)

### 暗号化ライブラリーにおける TLS 1.3 のサポート



今回の更新で、主要なすべてのバックエンド暗号ライブラリーで、TLS (Transport Layer Security) 1.3 が有効になっています。これにより、オペレーティングシステムの通信層での待ち時間を短縮し、RSA-PSS、X25519 などの新しいアルゴリズムを使用して、アプリケーションのプライバシーとセキュリティを強化します。

(BZ#1516728)

## RHEL 8 の OpenSCAP における新機能

**OpenSCAP** スイートがアップストリームバージョン 1.3.0 にアップグレードし、以前のバージョンに対する機能拡張が数多く追加されました。以下は、主な変更点です。

- API と ABI が統合され、更新、非推奨、未使用のシンボルが削除されています。
- プローブは、独立したプロセスとしてではなく、**oscap** プロセス内のスレッドとして実行します。
- コマンドラインインターフェースが更新しました。
- **Python 2** バインディングが、**Python 3** バインディングに置き換えられました。

(BZ#1614273)

## Audit 3.0 が auditd を auditd に置き換え

今回の更新で、**auditd** 機能が **auditd** に移動したため、**auditd** 設定オプションが **auditd.conf** に追加されました。さらに、**plugins.d** ディレクトリーが **/etc/audit** に移動しました。**auditd** と、そのプラグインの現在のステータスは、**service auditd state** コマンドを実行して確認できます。

(BZ#1616428)

## rsyslog imfile がシンボリックリンクをサポート

今回の更新で、**rsyslog imfile** モジュールのパフォーマンスが向上し、設定オプションが増えました。これにより、より複雑なファイルにこのモジュールを使用してユースケースを監視できるようになりました。たとえば、設定されたパスのどこにでも、グロブパターンを使用したファイルのモニターを使用し、データのスループットが増えたときに、シンボリックリンクのターゲットをローテートできます。

(BZ#1614179)

## OpenSSH サーバーキーの自動生成を sshd-keygen@.service が処理

サーバーホストキーの RSA、ECDSA、および ED25519 がない場合は、**OpenSSH** が自動的に作成します。RHEL 8 でホストキーの作成を設定するには、インスタンス化したサービス **sshd-keygen@.service** を使用します。

たとえば、RSA キータイプの自動作成を無効にするには、次のコマンドを実行します。

```
# systemctl mask sshd-keygen@rsa.service
```

詳細は **/etc/sysconfig/ssh** ファイルを参照してください。

(BZ#1228088)

## 現在、デフォルトの rsyslog 設定ファイルのフォーマットがレガシーではない

**rsyslog** パッケージの設定ファイルが、デフォルトで非レガシーフォーマットを使用するようになりました。レガシーフォーマットも使用できますが、現在の設定ステートメントとレガシーの設定ステート

メントを混在する場合は制約がいくつかあります。以前の RHEL リリースから引き継がれた設定は修正する必要があります。詳細は man ページの **rsyslog.conf(5)** を参照してください。

(BZ#1619645)

### 新しい SELinux ブール値

今回の SELinux システムポリシーの更新により、以下のブール値が導入されました。

- colord\_use\_nfs
- mysql\_connect\_http
- pdns\_can\_network\_connect\_db
- ssh\_use\_tcpd
- sslh\_can\_bind\_any\_port
- sslh\_can\_connect\_any\_port
- virt\_use\_pcscd

詳細は、以下のコマンドの出力を参照してください。

```
# semanage boolean -l
```

(JIRA:RHELPLAN-10347)

## 4.15. 仮想化

### KVM が、5 レベルのページングをサポート

Red Hat Enterprise Linux 8 では、KVM 仮想化は 5 レベルのページング機能をサポートします。これにより、ホストおよびゲストのシステムが使用できる物理アドレス空間と、仮想アドレス空間が大幅に増加します。

(BZ#1485229)

### KVM は RHEL 8 で UMIP をサポート

KVM 仮想化システムが、ユーザー領域のアプリケーションが、システム全体の設定にアクセスしないようにする UMIP (User-Mode Instruction Prevention) 機能をサポートするようになりました。これにより、権限昇格攻撃の潜在的なベクトルが減少するため、KVM ハイパーバイザーとそのゲストマシンがより安全になります。

(BZ#1494651)

### KVM ゲストクラッシュレポートの追加情報

ゲストが予期せず終了したり、応答しなくなった場合に、KVM ハイパーバイザーが生成するクラッシュ情報が拡張されました。これにより、KVM 仮想化のデプロイメントにおける問題の診断と修正が容易になります。

(BZ#1508139)

### RHEL 8 の qemu-kvm 2.12

Red Hat Enterprise Linux 8 は、**qemu-kvm** 2.12 で配布されています。このバージョンでは複数のバグが修正され、Red Hat Enterprise Linux 7 で利用可能なバージョン 1.5.3 に機能拡張が数多く追加されました。

具体的には、以下の機能が導入されました。

- Q35 ゲストマシンのタイプ
- UEFI ゲストブート
- ゲストの NUMA チューニングとピンニング
- vCPU のホットプラグとホットアンプラグ
- ゲストの I/O スレッド

**qemu-kvm** 2.12 で利用できる機能の一部は、Red Hat Enterprise Linux 8 ではサポートされていないことに注意してください。詳細情報は、Red Hat カスタマーポータル「Feature support and limitations in RHEL 8 virtualization」を参照してください。

(BZ#1559240)

### NVIDIA vGPU が VNC コンソールと互換性がある

NVIDIA 仮想 GPU (vGPU) 機能を使用するとき、VNC コンソールを使用して、ゲストの視覚的な出力を表示できるようになりました。

(BZ#1497911)

### 仮想化システムで Ceph をサポート

今回の更新で、Red Hat がサポートする全 CPU アーキテクチャーの KVM 仮想システムで、Ceph ストレージがサポートされるようになりました。

(BZ#1578855)

### 仮想化システムで Q35 マシンタイプをサポート

Red hat Enterprise Linux 8 では、より現代的な PCI Express ベースのマシンタイプである **Q35** のサポートが導入されました。これにより、仮想デバイスの機能とパフォーマンスにさまざまな改善が行われ、最新の広範囲なデバイスへの互換性が保証されます。さらに、Red Hat Enterprise Linux 8 で作成された仮想マシンは、デフォルトで **Q35** を使用するよう設定されています。

また、以前のデフォルトの **PC** マシンタイプは、非推奨となったため、Q35 に対応しない以前のオペレーティングシステムを仮想化する場合にのみ使用してください。

(BZ#1599777)

### IBM Z の仮想マシン用の対話型ブートローダー

IBM Z ホストで仮想マシン (VM) を起動するとき、QEMU ブートローダファームウェアが対話型コンソールインタフェースを表示するようになりました。これにより、ホスト環境にアクセスせずにゲスト OS の起動問題をトラブルシューティングすることができます。

(BZ#1508137)

### QEMU サンドボックス化が追加

Red Hat Enterprise Linux 8 では、QEMU エミュレーターでサンドボックス機能が導入されました。これにより、QEMU が実行するシステムコールに制限が設定でき、仮想マシンの安全性が高まりました。この機能はデフォルトで有効になっており、設定されています。

(JIRA:RHELPLAN-10628)

### Intel Xeon SnowRidge で GFNI および CLDEMOT の命令セットが有効化

Intel Xeon SnowRidge システムの RHEL 8 ホストで稼働している仮想マシン (VM) で、GFNI および CLDEMOT 命令セットを使用できるようになりました。これにより、特定のシナリオでこのような VM のパフォーマンスが大幅に向上する可能性があります。

(BZ#1494705)

### NVMe デバイス用の VFIO ベースのブロックドライバーが追加

QEMU エミュレーターは、NVMe (non-volatile memory express) デバイスで、VFIO (Virtual Function I/O) ベースのドライバーを導入しています。ドライバーは、仮想マシン (VM) に接続されている NVMe デバイスと直接通信し、カーネルシステムレイヤーと、その NVMe ドライバーを使用しないようにします。これにより、仮想マシン内の NVMe デバイスのパフォーマンスが向上します。

(BZ#1519004)

### Huge page の対応が改善

RHEL 8 を仮想ホストとして使用する場合、ユーザーは仮想マシン (VM) のメモリーに対応するページのサイズを、CPU が対応する任意のサイズに変更できます。これにより、仮想マシンのパフォーマンスが大幅に改善します。

仮想マシンのメモリーページのサイズを設定するには、仮想マシンの XML 設定を編集して、`<hugepages>` 要素を `<memoryBacking>` セクションに追加します。

(JIRA:RHELPLAN-14607)

## 4.16. サポート関連

### sosreport が eBPF ベースのプログラムとマップを報告

`sosreport` ツールが拡張され、Red Hat Enterprise Linux 8 に読み込まれた eBPF (extended Berkeley Packet Filtering) プログラムおよびマップをすべて報告できるようになりました。

(BZ#1559836)

## 第5章 テクノロジープレビュー

本パートでは、Red Hat Enterprise Linux 8.0 で利用可能なテクノロジープレビュー機能を説明します。

テクノロジープレビュー機能に対する Red Hat のサポート範囲の詳細は、「[テクノロジープレビュー機能のサポート範囲](#)」を参照してください。

### 5.1. カーネル

#### XDP がテクノロジープレビューとして利用可能に

テクノロジープレビューとして提供されている XDP (eXpress Data Path)機能により、カーネルで高性能なパケット処理を行うために、BPF (Berkeley Packet Filter) プログラムをアップロードする手段が提供されます。これにより、カーネルのネットワークデータパスをプログラムで制御できます。

(BZ#1503672)

#### eBPF がテクノロジープレビューとして利用可能に

eBPF (extended Berkeley Packet Filtering)機能は、ネットワーキングおよび追跡の両方で、テクノロジープレビューとして利用できます。eBPF を使用すると、ユーザー領域でカスタムプログラムをさまざまなポイント (ソケット、トレースポイント、パケット受信) に接続し、データを受信して処理できます。この機能には、新しいシステムコール `bpf()` が含まれます。これにより、さまざまな種類のマップを作成したり、さまざまな種類のプログラムをカーネルに挿入できます。詳細は、man ページの `bpf(2)` を参照してください。

(BZ#1559616)

#### BCC がテクノロジープレビューとして利用可能に

**BCC (BPF Compiler Collection)** は、Red Hat Enterprise Linux 8 でテクノロジープレビューとして入手可能な、効率的なカーネルトレースおよび操作のプログラムを作成するためのユーザー領域ツールキットです。**BCC** は、**eBPF (extended Berkeley Packet Filtering)** を使用して、Linux オペレーティングシステムの I/O 解析、ネットワーキング、およびモニタリング用のツールを提供します。

(BZ#1548302)

#### RHEL 8 では、Control Group v2 がテクノロジープレビューとして利用可能に

統一された階層制御グループである **Control Group v2** メカニズムに対応します。**Control Group v2** は、プロセスを階層的に編成し、制御された設定可能な方法で、階層に従ってシステムリソースを分配します。

以前のバージョンとは異なり、**Control Group v2** には階層が1つしかありません。このように階層が単純であるため、Linux カーネルでは次のことが可能になります。

- 所有者の役割に基づいたプロセスの分類
- 複数の階層でポリシーが競合する問題の解消

**Control Group v2** では、多くのコントローラーがサポートされています。

- CPU コントローラーにより、CPU サイクルの配分が調整されます。このコントローラーには以下が実装されています。
  - 通常のスケジューリングポリシーに対する重みと、絶対帯域幅制限モデル
  - 実時間スケジューリングポリシーに対する絶対帯域幅割り当てモデル

- メモリーコントローラーは、メモリー配分を調整します。現在、次の種類のメモリー使用量が追跡されます。
  - ユーザー側のメモリー (ページキャッシュと匿名メモリー)
  - dentry、inode などのカーネルデータ構造
  - TCP ソケットバッファ
- I/O コントローラーは、I/O リソースの配分を制限します。
- ライトバックコントローラーは、メモリーコントローラーと I/O コントローラーの両方と対話し、Control Group v2 に固有のものであります。

上記の情報は、<https://www.kernel.org/doc/Documentation/cgroup-v2.txt> に基づいています。ここでは、個別の Control Group v2 コントローラーに関する情報を取得できます。

(BZ#1401552)

### early kdump が Red Hat Enterprise Linux 8 でテクノロジープレビューとして利用可能

early kdump 機能により、早い段階でクラッシュしても vmcore 情報を取得するのに十分な早さでクラッシュカーネルと initramfs の読み込みが行われるようになりました。early kdump の詳細は </usr/share/doc/kexec-tools/early-kdump-howto.txt> ファイルを参照してください。

(BZ#1520209)

## 5.2. ファイルシステムおよびストレージ

### LVM 論理ボリュームタイプとして VDO が利用可能

LVM を使用して、Virtual Data Optimizer (VDO) タイプの論理ボリュームを作成できます。VDO は、データの圧縮および重複排除を行う機能を備えた仮想ブロックデバイスです。

この機能はテクノロジープレビューとなります。

(BZ#1643553)

### DIF/DIX (Data Integrity Field/Data Integrity Extension) のサポート

DIF/DIX は SCSI 規格への追加分です。サポートされていると明記されている場合を除き、引き続き HBA およびストレージアレイのテクノロジープレビューとなります。

DIF/DIX により DIF (Data Integrity Field) が追加され、一般的に使用される 512 バイトのディスクブロックのサイズが 512 から 520 バイトに増加します。DIF は、書き込みの発生時に HBA (Host Bus Adapter) により算出されるデータブロックのチェックサム値を保存します。その後、受信時にストレージデバイスがチェックサムを確認し、データとチェックサムの両方を保存します。読み取りが発生すると、チェックサムが、ストレージデバイスおよび受信する HBA により検証されます。

(BZ#1649493)

### qla2xxx を使用する Qlogic アダプターで、NVMe/FC がテクノロジープレビューとして利用可能に

qla2xxx ドライバーを使用する Qlogic アダプターで、トランスポートタイプ NVMe/FC (The NVMe over Fibre Channel) がテクノロジープレビューとして利用できます。

(BZ#1649922)

## 5.3. 高可用性およびクラスター

### Pacemaker podman バンドルがテクノロジープレビューとして利用可能に

Pacemaker コンテナバンドルは、**podman** コンテナプラットフォーム上で動作するようになりました。コンテナバンドル機能はテクノロジープレビューとして利用できますが、例外が1つありません。Red Hat は、Red Hat Openstack 用の Pacemaker バンドルの使用を完全にサポートします。

(BZ#1619620)

## 5.4. セキュリティー

### RHEL 8.0 リリースの SWID タグ

ISO/IEC 19770-2:2015 メカニズムを使用して RHEL 8.0 インストールの識別を可能にするために、`/usr/lib/swidtag/redhat.com/com.redhat.RHEL-8-<architecture>.swidtag` ファイルおよび `/usr/lib/swidtag/redhat.com/com.redhat.RHEL-8.0-<architecture>.swidtag` ファイルに、ソフト識別 (SWID) タグがインストールされます。このタグの親ディレクトリーも、`/etc/swid/swidtags.d/redhat.com` シンボリックリンクからたどることができます。

SWID タグファイルの XML 署名は、`xmlsec1 verify` コマンドを使用して確認できます。以下に例を示します。

```
xmlsec1 verify --trusted-pem /etc/pki/swid/CA/redhat.com/redhatcodesignca.cert
/usr/share/redhat.com/com.redhat.RHEL-8-x86_64.swidtag
```

コード署名認証局の証明書は、カスタマーポータル[の「製品署名キー」](#) ページからも入手できます。

(BZ#1636338)

## 5.5. 仮想化

### KVM 仮想マシンの AMD SEV

テクノロジープレビューとして、RHEL 8 に、KVM ハイパーバイザーを使用する AMD EPYC ホストマシン用のセキュア暗号化仮想化 (SEV) 機能が同梱されます。仮想マシン (VM) で有効になっている場合は、ホストが仮想マシン上のデータにアクセスできないように、SEV が VM メモリーを暗号化します。ホストがマルウェアに感染した場合は、これにより仮想マシンのセキュリティーが向上します。

1台のホストでこの機能を同時に使用できる仮想マシンの数は、ホストのハードウェアによって決まります。現在の AMD EPYC プロセッサは、SEV を使用して 15 台以下の稼働中の仮想マシンをサポートします。

(BZ#1501618, BZ#1501607)

IBM POWER 9 でネストされた仮想化システムが利用可能になりました。

テクノロジープレビューとして、IBM POWER 9 システムで稼働する RHEL 8 ホストマシンでネストされた仮想化機能を使用することが可能になりました。ネストされた仮想化により、KVM 仮想マシンをハイパーバイザーとして機能させることができます。これにより、仮想マシン内で仮想マシンを実行できます。

ネストされた仮想環境では、IBM POWER 9、ホスト、ゲスト、ネストされたゲストのすべてで、以下の OS のいずれかが実行している必要があります。

- RHEL 8

- RHEL 7 for POWER 9

(BZ#1505999)



## 第6章 非推奨の機能

ここでは、Red Hat Enterprise Linux 8.0 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、Red Hat Enterprise Linux 8 のライフサイクルが終了するまでサポートされます。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。

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の更新は、セキュリティと重大な修正にのみ行われます。Red Hat は、このようなハードウェアはできるだけ早い機会に取り替えることをお勧めします。

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、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。

### 6.1. インストーラーおよびイメージの作成

#### ignoredisk Kickstart コマンドの `--interactive` オプションが非推奨に

Red Hat Enterprise Linux の将来のリリースで `--interactive` オプションを使用すると、致命的なインストールエラーが発生します。このオプションを削除するには、Kickstart ファイルを変更することが推奨されます。

(BZ#1637872)

### 6.2. ファイルシステムおよびストレージ

#### NFSv3 over UDP が無効に

NFS サーバーは、デフォルトで UDP (User Datagram Protocol) ソケットを開いたり、リッスンしなくなりました。バージョン 4 では TCP (Transmission Control Protocol) が必要なため、この変更は NFS バージョン 3 にのみ影響を及ぼします。

RHEL 8 では、NFS over UDP がサポートされなくなりました。

(BZ#1592011)

#### NVMe/FC ターゲットモードが非推奨に

NVMe/FC (Nonvolatile Memory Express over Fibre Channel) トランスポートプロトコルのターゲットモードは、RHEL 7 ではテクノロジープレビューとして利用できました。RHEL 8 では NVMe/FC ターゲットモードが非推奨になっています。

NVMe ターゲットモードで FC ホストバスアダプター (HBA) ポートを有効にすると、以下のエラーメッセージが発生します。

**警告:** Red Hat は、このリリースでは、NVMe over FC ターゲットのドライバーに十分なテストを行っていないため、実稼働システムでは使用できません。

(BZ#1664838)

### 6.3. ネットワーク

## RHEL 8 でネットワークスクリプトが非推奨に

Red Hat Enterprise Linux 8 では、ネットワークスクリプトが非推奨になっており、デフォルトでは提供されなくなりました。基本的なインストールでは、`nmcli` ツール経由で、**NetworkManager** サービスを呼び出す **ifup** スクリプトおよび **ifdown** スクリプトの新しいバージョンを提供します。Red Hat Enterprise Linux 8 では、**ifup** スクリプトおよび **ifdown** スクリプトを実行するのに、**NetworkManager** の実行が必要になります。

`/sbin/ifup-local`、`ifdown-pre-local`、および `ifdown-local` の各スクリプトでは、カスタムコマンドが実行されません。

このスクリプトが必要な場合は、次のコマンドを使用すればシステム内に非推奨のネットワークスクリプトをインストールできます。

```
~]# yum install network-scripts
```

**ifup** スクリプトと **ifdown** スクリプトが、インストールされている従来のネットワークスクリプトにリンクされます。

従来のネットワークスクリプトを呼び出すと、それらが非推奨であることを示す警告が表示されます。

(BZ#1647725)

## 6.4. セキュリティー

### DSA が Red Hat Enterprise Linux 8 で非推奨に

Red Hat Enterprise Linux 8 では、DSA (Digital Signature Algorithm) は非推奨であるとみなれされます。DSA キーに依存する認証メカニズムはデフォルト設定では機能しません。**OpenSSH** クライアントは、LEGACY ポリシーでも DSA ホストキーを許可しません。

(BZ#1646541)

### SSL2 Client Hello が NSS で非推奨に

TLS (Transport Layer Security) プロトコルバージョン 1.2 以前は、**SSL** (Secure Sockets Layer) プロトコルバージョン 2 と後方互換性がある形式の **Client Hello** メッセージを使用してネゴシエーションを開始できます。**NSS** (Network Security Services) ライブラリーにおけるこの機能のサポートは非推奨になっており、デフォルトで無効になっています。

この機能のサポートが必要なアプリケーションを有効にするには、新しい **SSL\_ENABLE\_V2\_COMPATIBLE\_HELLO** API を使用する必要があります。この機能のサポートは、Red Hat Enterprise Linux 8 の将来のリリースから完全に削除される可能性があります。

(BZ#1645153)

## 6.5. 仮想化

### RHEL 8 では、仮想マシンのスナップショットのサポートが適切ではない

仮想マシン (VM) スナップショットを作成する現在のメカニズムが適切に機能していないため、推奨されなくなりました。これにより、RHEL 8 では、VM スナップショットを使用しないことをお勧めします。

新しい VM スナップショットメカニズムは開発中で、RHEL 8 の将来のマイナーリリースで完全に実装される予定です。

---

(BZ#1686057)

### **Cirrus VGA 仮想 GPU タイプが非推奨に**

Red Hat Enterprise Linux の将来のメジャーアップデートでは、KVM 仮想マシンで **Cirrus VGA** GPU デバイスがサポートされなくなります。したがって、Red Hat は、Cirrus VGA の代わりに **stdvga** デバイス、**virtio-vga** デバイス、または **qxl** デバイスを使用することを推奨します。

(BZ#1651994)

### **virt-manager が非推奨に**

Virtual Machine Manager アプリケーション (**virt-manager**) は非推奨になっています。RHEL 8 Web コンソール (**Cockpit**) は、後続のリリースで置き換えられる予定です。したがって、GUI で仮想化を管理する場合は、Web コンソールを使用することが推奨されます。ただし、Red Hat Enterprise Linux 8.0 では、機能によっては、**virt-manager** またはコマンドラインからしかアクセスできません。

(JIRA:RHELPLAN-10304)

## 第7章 既知の問題

このパートでは Red Hat Enterprise Linux 8 の既知の問題を説明します。

### 7.1. インストーラーおよびイメージの作成

#### Kickstart コマンドの `auth` および `authconfig` で AppStream リポジトリが必要になる

インストール中に、Kickstart コマンドの `auth` および `authconfig` で `authselect-compat` パッケージが必要になります。`auth` または `authconfig` を使用したときに、このパッケージがないとインストールに失敗します。ただし、設計上、`authselect-compat` パッケージは AppStream リポジトリでしか利用できません。

この問題を回避するには、BaseOS リポジトリおよび AppStream リポジトリがインストーラーで利用できることを確認するか、インストール中に Kickstart コマンドの `authselect` コマンドを使用します。

(BZ#1640697)

#### Binary DVD.iso ファイルの内容をパーティションにコピーしても、`.treeinfo` ファイルおよび `.discinfo` ファイルがコピーされない

RHEL 8.0 Binary DVD.iso イメージファイルの内容を、ローカルインストール用のパーティションにコピーする場合に、`cp <path>/* <mounted partition>/dir` コマンドの `*` で、`.treeinfo` ファイルおよび `.discinfo` ファイルがコピーされず、インストールに失敗します。これにより、BaseOS リポジトリおよび AppStream のリポジトリが読み込まれず、`anaconda.log` ファイルのデバッグログメッセージでしか問題を確認できません。

この問題を回避するには、不足している `.treeinfo` ファイルおよび `.discinfo` ファイルをパーティションにコピーします。

(BZ#1692746)

### 7.2. カーネル

#### i40iw モジュールがシステムの起動時に自動的に読み込まれない

多くの i40e NIC で iWarp がサポートされておらず、i40iw モジュールがサスペンド/レジュームを完全にサポートしていないため、このモジュールがデフォルトで自動的に読み込まれてサスペンド/レジューム正しく機能させることができません。この問題を回避するには、`/lib/udev/rules.d/90-rdma-hw-modules.rules` ファイルを手動で編集して、i40iw の自動ロードを有効にします。

また、同じマシンにある i40e デバイスに、別の RDMA デバイスがインストールされている場合に、i40e 以外の RDMA デバイスで、i40iw モジュールを含む、有効なすべての RDMA スタックモジュールを読み込む `rdma` サービスが起動します。

(BZ#1623712)

#### システムにデバイスが多数接続されていると、システムが応答しなくなることがある

Red Hat Enterprise Linux 8 に多数のデバイスを設定すると、システムコンソールにコンソールメッセージが多数表示されます。たとえば、論理ユニット番号 (LUN) が多数あり、その各 LUN へのパスが複数あると、このような状況が発生します。カーネルが行っている他の作業に加えて、コンソールメッセージがあふれるとカーネルがハングし、カーネルのウォッチドッグによりカーネルパニック発生する場合があります。

スキャンはブートサイクルの早い段階で行われるため、接続されているデバイスが多くなるとシステムが応答しなくなります。これは通常、起動時に発生します。

システム起動後のデバイススキャンイベント中に、マシンで **kdump** が有効になっていると、ハードロックアップにより **vmcore** イメージがキャプチャーされます。

この問題を回避するには、ウォッチドッグロックアップタイマーを増やします。これを行うには、カーネルのコマンドラインに **watchdog\_thresh=N** オプションを追加します。**N** を秒数に置き換えます。

- デバイスが 1000 台未満の場合は、**30** を使用してください。
- デバイスが 1000 台以上の場合は、**60** を使用してください。

ストレージの場合、デバイスの数はすべての LUN へのパスの数です。通常、これは **/dev/sd\*** デバイスの数です。

回避策を適用した後、大量のデバイスを構成してもシステムが応答しなくなることはなくなりました。

(BZ#1598448)

### KSM が、NUMA メモリーポリシーを無視することがある

"merge\_across\_nodes=1" パラメーターで、カーネル共有メモリー (KSM) 機能を有効にすると、KSM は、mbind() 関数が設定したメモリーポリシーを無視し、一部のメモリーから、ポリシーに一致しない Non-Uniform Memory Access (NUMA) ノードにページをマージすることができない場合があります。

この問題を回避するには、KSM を無効にするか、QEMU で NUMA メモリーバインディングを使用する場合は merge\_across\_nodes パラメーターを「0」に設定すると、KVM VM 用に設定した NUMA メモリーポリシーが期待どおりに機能します。

(BZ#1153521)

## 7.3. ソフトウェア管理

### root 以外のユーザーで yum list を実行すると、YUM がクラッシュする

**libdnf** パッケージを更新してから、root 以外のユーザーで **yum list** コマンドを実行すると、YUM が突然終了することがあります。このバグが発生した場合は、問題を解決するために root で **yum list** を実行してください。これにより、root 以外のユーザーで **yum list** を実行しようとしても、YUM クラッシュが発生しなくなります。

(BZ#1642458)

### man ページの yum(8) に、yum module profile コマンドが誤って記載されている

man ページの **yum(8)** に、YUM パッケージ管理ツールに、モジュールプロファイルに関する詳細を提供する **yum module profile** コマンドが含まれていると誤って記載されていますが、このコマンドは使用できなくなったため、使用すると、YUM により、コマンドが無効であることを示すエラーメッセージが表示されます。モジュールプロファイルの詳細は、代わりに **yum module info --profile** コマンドを使用してください。

(BZ#1622580)

### yum-plugin-aliases が現在利用できない

カスタムの yum エイリアスを追加する **alias** を提供する **yum-plugin-aliases** パッケージが現在利用できないため、現在エイリアスを使用することはできません。

(BZ#1647760)

### yum-plugin-changelog が現在利用できない

**yum-plugin-changelog** パッケージを使用すると、パッケージ更新前後で変更したログを確認できますが、現在は利用できません。

(BZ#1581191)

## 7.4. インフラストラクチャーサービス

### Tuned が、カーネル起動コマンドラインパラメーターを設定しない

**Tuned** ツールは、Boot Loader Specification (BLS) に対応していません。これはデフォルトで有効になっています。これにより、**Tuned** は、特定のカーネルブートコマンドラインパラメーターを設定しないため、パフォーマンスの低下や CPU コアの分離ができないなどの問題が発生します。この問題を回避するために、BLS を無効にして、**Tuned** を再起動します。

1. **grubby** パッケージをインストールします。
2. **/etc/default/grub** ファイルから次の行を削除します。

```
GRUB_ENABLE_BLSCFG=true
```

3. EFI 以外のシステムでは以下を実行して、**grub2.cfg** ファイルを再生成します。

```
grub2-mkconfig -o /etc/grub2.cfg
```

または EFI システムの場合:

```
grub2-mkconfig -o /etc/grub2-efi.cfg
```

4. 次のコマンドを実行して、**Tuned** を再起動します。

```
systemctl restart tuned
```

これにより、**Tuned** は、カーネルのブートパラメーターを期待通りに設定します。

(BZ#1576435)

## 7.5. シェルおよびコマンドラインツール

### net-snmp パッケージの Python バインディングが利用できない

**Net-SNMP** ツールスイートは、RHEL 8 のデフォルトの **Python** の実装である **Python 3** にバインディングを提供しません。これにより、**python-net-snmp** パッケージ、**python2-net-snmp** パッケージ、または **python3-net-snmp** パッケージが RHEL 8 では使用できません。

(BZ#1584510)

## 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

## データベースサーバーを並行してインストールできない

RPM パッケージが競合しているため、RHEL 8.0 では、**mariadb** モジュールと **mysql** モジュールを同時にインストールできません。

設計上、同じモジュールの複数のバージョン (ストリーム) を並行してインストールすることはできません。たとえば、**postgresql** モジュールから利用可能なストリーム ( **10** (デフォルト) または **9.6** ) の中から1つ選択する必要があります。RHEL 6 および RHEL 7 用の Red Hat Software Collections では、コンポーネントの並列インストールが可能です。RHEL 8 では、コンテナ内で異なるバージョンのデータベースサーバーを使用できます。

(BZ#1566048)

## mod\_cgid ログイングの問題

**mod\_cgid** Apache httpd モジュールを、スレッド化マルチプロセッシングモジュール (MPM) で使用する場合 (RHEL 8 のデフォルト) は、以下のログイング問題が発生します。

- CGI スクリプトの **stderr** 出力には、標準のタイムスタンプ情報が付いていません。
- **VirtualHost** に固有のログファイルに CGI スクリプトの **stderr** 出力が設定されていても、正しくリダイレクトされません。

(BZ#1633224)

## Perl モジュール **IO::Socket::SSL** が TLS 1.3 をサポートしない

セッション再開、ハンドシェイク後認証などの TLS 1.3 プロトコルの新機能は RHEL 8 の **OpenSSL** ライブラリーには実装されますが、Perl モジュール **Net::SSL** には実装されないため、**IO::Socket::SSL** Perl モジュールでは利用できません。クライアント証明書認証が失敗し、TLS 1.2 プロトコルを使用した場合よりもセッションの再確立が遅くなる可能性があります。

この問題を回避するには、**IO::Socket::SSL** オブジェクトを作成する場合に、**SSL\_version** オプションを **!TLSv1\_3** 値に設定して、TLS 1.3 の使用を無効にします。

(BZ#1632600)

## 7.7. ID 管理

### KCM 認証情報キャッシュは、1つの認証情報キャッシュ内で多数の認証情報には適していない

認証情報キャッシュに含まれる認証情報が多くなりすぎると、**kinit** などの Kerberos 操作には、**sssd-kcm** コンポーネントと、基本的なデータベースとの間でデータを転送するのに使用するバッファにハードコード化された制限があるため、失敗します。

この問題を回避するには、**/etc/sss/sss.conf** ファイルの **kcm** セクションに **ccache\_storage = memory** オプションを追加します。これにより、**kcm** レスポンダーが、永続的ではなく、認証情報キャッシュをメモリー内に格納するようになります。これを行った場合は、システムを再起動するか、**sssd-kcm** で認証情報キャッシュを消去します。KCM は、最大 64 KB の **ccache** サイズを処理できます。

(BZ#1448094)

### タイムアウト値が競合しているため、SSSD がサーバーに接続できない

System Security Services Daemon (SSSD) が使用するフェイルオーバー操作に関連するデフォルトのタイムアウト値の一部が競合しています。これにより、全体のタイムアウトとなる接続操作になる前に、SSSD が1台のサーバーと通信するために予約されているタイムアウト値により、SSSD がその他の

サーバーに試行しなくなります。この問題を回避するために、`ldap_opt_timeout` タイムアウトパラメーターの値を `dns_resolver_timeout` パラメーターの値より大きく設定し、`dns_resolver_timeout` パラメーターの値を `dns_resolver_op_timeout` パラメーターの値よりも高く設定します。

(BZ#1382750)

### IdM Web UI へのログインにスマートカードを使用できない

ユーザーが、自分のスマートカードに保存されている証明書を使用して Identity Management (IdM) の Web UI にログインしようとする、システムセキュリティーサービスデーモン (SSSD) の D-Bus インターフェースコードが、誤ったコールバックを使用してユーザーを調べ、ルックアップがクラッシュします。この問題を回避するには、他の認証方法を使用してください。

(BZ#1642508)

### IdM サーバーが FIPS で機能しない

Tomcat 用の SSL コネクタの実装が不完全なため、証明書サーバーがインストールされた IdM サーバーが、FIPS モードが有効になっているマシンでは機能しません。

(BZ#1673296)

### nuxwdog サービスが HSM 環境では失敗するため、非 HSM 環境に keyutils パッケージをインストールする必要がある

`nuxwdog` ウォッチドッグサービスが証明書システムに統合されました。その結果、`nuxwdog` は別パッケージとして提供されなくなりました。ウォッチドッグサービスを使用するには、`pki-server` パッケージをインストールしてください。

`nuxwdog` サービスには、以下の既知の問題があります。

- ハードウェアストレージモジュール (HSM) を使用していると、`nuxwdog` サービスは機能しません。この問題の回避策はありません。
- 非 HSM 環境では、Red Hat Enterprise Linux 8.0 は、`keyutils` パッケージを依存関係として自動的にインストールしません。このパッケージを手動でインストールするには、`dnf install keyutils` コマンドを使用します。

(BZ#1652269)

## 7.8. コンパイラーおよび開発ツール

### GCC により生成された合成関数により SystemTap が混乱する

GCC 最適化により、その他の関数を部分的にインラインにコピーした合成関数を生成する場合があります。SystemTap や GDB などのツールは、これらの合成関数と実際の関数を区別できません。これにより、SystemTap は、合成関数と実関数の両方のエントリーポイントにプローブを置くことができるため、1つの実関数呼び出しに対して、複数のプローブを数えあげます。

この問題を回避するには、SystemTap スクリプトを、再帰の検出、インライン化された部分関数に関連するプローブの非表示などの方法で調整する必要があります。たとえば、スクリプトは、以下のようになります。

```
probe kernel.function("can_nice").call { }
```

説明されている問題は、次のように回避を試行できます。



```
global in_can_nice%

probe kernel.function("can_nice").call {
    in_can_nice[tid()] ++;
    if (in_can_nice[tid()] > 1) { next }
    /* code for real probe handler */
}

probe kernel.function("can_nice").return {
    in_can_nice[tid()] --;
}

```

このスクリプト例では、不明な kprobes や kretprobes、または、本物の意図的な再帰など、考えられるすべてのシナリオが考慮されているわけではありません。

(BZ#1169184)

### Itrace ツールが、関数呼び出しを報告しない

すべての RHEL コンポーネントに適用されるバイナリ強化の改善により、**Itrace** ツールが、RHEL コンポーネントからのバイナリファイルの関数呼び出しを検出できなくなりました。これにより、**Itrace** の出力では、このようなバイナリファイルで使用されたときに検出される呼び出しが報告されなくなるため、空になります。現在利用できる回避策はありません。

ただし、**Itrace** では、各強化フラグを使用せずに構築されたカスタムバイナリファイルの呼び出しは報告されます。

(BZ#1618748, BZ#1655368)

## 7.9. ファイルシステムおよびストレージ

### iscsiuio パッケージを使用して iSCSI ターゲットを検出できない

Red Hat Enterprise Linux 8 では、PCI レジスタ領域への同時アクセスが許可されていません。そのため、**could not set host net params (err 29)** エラーが設定され、ディスカバリーポータルへの接続が失敗します。この問題を回避するには、iSCSI オフロードに、カーネルコマンドラインでカーネルパラメーター **iomem=relaxed** を設定します。これは特に、**bnx2i** ドライバーを使用したオフロードに関係します。その結果、ディスカバリーポータルへの接続は成功し、**iscsiuio** パッケージが正しく機能するようになりました。

(BZ#1626629)

### XFS DAX マウントオプションは、共有コピーオンライトデータエクステンツと互換性がない

共有コピーオンライトのデータエクステンツ機能でフォーマットされた XFS ファイルシステムは、**-o dax** マウントオプションと互換性がありません。これにより、そのようなファイルシステムを **-o dax** でマウントすると失敗します。

この問題を回避するには、**reflink=0** メタデータオプションを使用してファイルシステムをフォーマットし、共有コピーオンライトデータエクステンツを無効にします。

```
# mkfs.xfs -m reflink=0 block-device

```

このため、**-o dax** でファイルシステムをマウントすることに成功しました。

(BZ#1620330)

## 7.10. ネットワーク

### nftables が多次元の IP セットタイプをサポートしない

**nftables** パケットフィルタリングフレームワークは、連結と区間を持つセット型をサポートしません。これにより、**hash:net,port** などの多次元 IP セットタイプを、**nftables** と共に使用することができません。

この問題を回避するには、多次元 IP セットタイプが必要な場合に、**iptables** フレームワークを **ipset** ツールと共に使用してください。

(BZ#1593711)

## 7.11. セキュリティー

### OpenSCAP の rpmverifypackage が正常に動作しない

**rpmverifypackage** プロブにより、システムコール **chdir** および **chroot** が 2 回呼び出されます。これにより、カスタムの Open Vulnerability and Assessment Language (OVAL) コンテンツを使用した **OpenSCAP** をスキャンする際にそのプルーフを使用していると、エラー発生します。

この問題を回避するには、コンテンツで OVAL テスト **rpmverifypackage\_test** を使用しないか、**rpmverifypackage\_test** が使用されていない **scap-security-guide** パッケージのコンテンツのみを使用します。

(BZ#1646197)

### libssh がシステム全体の暗号化ポリシーに準拠しない

**libssh** ライブラリーは、システム全体の暗号化ポリシー設定には従いません。これにより、管理者が、**update-crypto-policies** コマンドを使用して暗号ポリシーレベルを変更しても、サポートされているアルゴリズムのセットは変更しません。

この問題を回避するには、**libssh** を使用するアプリケーションごとに、公開された一連のアルゴリズムを個別に設定する必要があります。これにより、システムがポリシーレベルの LEGACY または FUTURE に設定されていると、**OpenSSH** と比較したときに、**libssh** を使用するアプリケーションの動作が矛盾します。

(BZ#1646563)

### SCAP Workbench が、カスタムプロファイルから結果ベースの修正を生成することができない

**SCAP Workbench** ツールを使用してカスタムプロファイルから結果ベースの修正ロールを生成しようとすると、以下のエラーが発生します。

```
Error generating remediation role .../remediation.sh: Exit code of oscap was 1: [output truncated]
```

この問題を回避するには、**oscap** コマンドを、**--tailoring-file** オプションとともに使用します。

(BZ#1640715)

### OpenSCAP の rpmverifyfile が機能しない

**OpenSCAP** スキャナーは、オフラインモードで現在の作業ディレクトリーを正しく変更せず、**fchdir** 関数が、**OpenSCAP rpmverifyfile** プロブで正しい引数で呼び出さないようにします。そのため、SCAP コンテンツで **rpmverifyfile\_test** を使用すると、**oscap-chroot** コマンドを使用した任意のファ

---

イルシステムのスキャンに失敗します。したがって、上記のシナリオでは、**oscap-chroot** が中断します。

(BZ#1636431)

### コンテナのセキュリティおよびコンプライアンススキャンを行うユーティリティーが利用できない

Red Hat Enterprise Linux 7 では、Atomic テクノロジーに基づいた Docker コンテナのスキャンに、**oscap-docker** ユーティリティーを使用できました。Red Hat Enterprise Linux 8 では、Docker 関連、および Atomic 関連の **OpenSCAP** コマンドが利用できません。そのため、コンテナのセキュリティおよびコンプライアンススキャンには、**oscap-docker** または同等のユーティリティーを使用できません。

(BZ#1642373)

## 第8章 コンテナに関する主な変更点

Red Hat Enterprise Linux (RHEL) 8.0 用の一連のコンテナイメージが利用できます。以下は、主な変更点です。

- RHEL 8.0 には、Docker が同梱されていません。コンテナを使用するには、ツールの **podman**、**buildah**、**skopeo**、および **runc** が必要です。RHEL 8 でこのツールとコンテナを使用する方法は『[Building, running, and managing containers](#)』を参照してください。
- **podman** ツールが、完全にサポートされるようになりました。**podman** ツールでは、1つのノードにある Pod、コンテナイメージ、およびコンテナが管理されます。これは、**libpod** ライブラリーでビルドされます。これは、コンテナおよびコンテナのグループ (Pod と呼ばれています) の管理を有効にします。

**podman** の使用方法は『[Building, running, and managing containers](#)』を参照してください。

- RHEL 8 GA では、Red Hat Universal Base Images (UBI) が新たに利用できるようになりました。UBI は、標準および最小の RHEL ベースイメージなど、以前に提供された Red Hat イメージの一部を置き換えるものです。以前の Red Hat イメージとは異なり、UBI は自由に再配布可能です。つまり、あらゆる環境で使用でき、どこでも共有できます。Red Hat のお客様でなくても使用できます。

UBI の詳細は『[Building, running, and managing containers](#)』を参照してください。

- RHEL 8 GA では、AppStream コンポーネントを提供する追加のコンテナイメージが用意されています。このコンテナイメージは、RHEL 7 の **Red Hat Software Collections** で配布されます。このような RHEL 8 イメージは、すべて **ubi8** ベースイメージに基づいています。
- 64 ビットの ARM アーキテクチャーに対するコンテナイメージの ARM は、RHEL 8 で完全にサポートされます。
- RHEL 8 では、**rhel-tools** コンテナが削除されました。**sos** ツールおよび **redhat-support-tool** ツールは **support-tools** コンテナで提供されています。システム管理者は、システムツールコンテナイメージをビルドするためのベースとしてこのイメージを使用することもできます。
- ルートレスコンテナのサポートは、RHEL 8 ではテクノロジープレビューとして利用できません。ルートレスコンテナは、管理者権限なしで通常のシステムユーザーにより作成および管理されるコンテナです。

## 付録A コンポーネント別のチケットリスト

コンポーネント	チケット
<b>389-ds-base</b>	BZ#1334254, BZ#1358706, <a href="#">BZ#1693159</a>
<b>NetworkManager</b>	BZ#1555013, BZ#1555012, BZ#1557035
<b>PackageKit</b>	BZ#1559414
<b>anaconda</b>	BZ#1499442, BZ#1500792, BZ#1547908, BZ#1612060, BZ#1595415, BZ#1610806, BZ#1533904
<b>audit</b>	BZ#1616428
<b>bcc</b>	BZ#1548302
<b>bind</b>	BZ#1588592
<b>boom-boot</b>	BZ#1649582
<b>boost</b>	BZ#1494495, BZ#1616244
<b>cloud-init</b>	BZ#1615599
<b>cmake</b>	BZ#1590139
<b>cockpit</b>	BZ#1619993
<b>crypto-policies</b>	BZ#1591620
<b>cryptsetup</b>	BZ#1564540
<b>device-mapper-multipath</b>	BZ#1643550
<b>distribution</b>	BZ#1566048, BZ#1516741, BZ#1516728
<b>dnf</b>	BZ#1622580, BZ#1647760, BZ#1581191
<b>esc</b>	BZ#1538645
<b>firewalld</b>	BZ#1509026
<b>gcc</b>	BZ#1169184, BZ#1607227, BZ#1535774, BZ#1504980, BZ#1246444
<b>gdm</b>	BZ#1589678

コンポーネント	チケット
<b>glibc</b>	BZ#1512004, BZ#1376834, BZ#1512010, BZ#1304448, BZ#1512009, BZ#1512006, BZ#1514839, BZ#1533608
<b>go-toolset</b>	BZ#1512570
<b>httpd</b>	BZ#1633224, BZ#1632754
<b>iproute</b>	BZ#1640991
<b>iptables</b>	BZ#1644030, BZ#1564596
<b>iscsi-initiator-utils</b>	BZ#1626629, BZ#1582099
<b>kernel-rt</b>	BZ#1592977
<b>kernel</b>	BZ#1598448, BZ#1643522, BZ#1485546, BZ#1562998, BZ#1485229, BZ#1494651, BZ#1485532, BZ#1494028, BZ#1563617, BZ#1485525, BZ#1261167, BZ#1562987, BZ#1273139, BZ#1559607, BZ#1401552, BZ#1638465, BZ#1598776, BZ#1503672, BZ#1664838, BZ#1596240, BZ#1534870, BZ#1501618, BZ#1153521, BZ#1494705, BZ#1620330, BZ#1505999
<b>kexec-tools</b>	BZ#1520209
<b>kmod-kvdo</b>	BZ#1534087, BZ#1639512
<b>libdnf</b>	BZ#1642458
<b>libreswan</b>	<a href="#">BZ#1657854</a>
<b>libssh</b>	BZ#1485241
<b>ltrace</b>	BZ#1618748, BZ#1584322
<b>lvm2</b>	BZ#1643553, BZ#1643543, BZ#1643545, BZ#1643547, BZ#1643549, BZ#1643562, BZ#1643576
<b>mariadb</b>	BZ#1637034
<b>net-snmp</b>	BZ#1584510
<b>nfs-utils</b>	BZ#1592011, BZ#1639432
<b>nftables</b>	BZ#1593711

コンポーネント	チケット
<b>nginx</b>	BZ#1545526
<b>nodejs-10-module</b>	BZ#1622118
<b>nss</b>	BZ#1645153
<b>nuxwdog</b>	<a href="#">BZ#1652269</a>
<b>openldap</b>	BZ#1570056
<b>opensc</b>	BZ#1595626
<b>openscap</b>	BZ#1646197, BZ#1636431, BZ#1642373, BZ#1618484, BZ#1614273, BZ#1618464
<b>openssh</b>	BZ#1622511, BZ#1228088
<b>pacemaker</b>	<a href="#">BZ#1543494</a>
<b>pcs</b>	BZ#1578891, BZ#1591308, BZ#1615420, BZ#1158816, BZ#1542288, BZ#1549535, BZ#1620190, BZ#1566430, BZ#1595829, BZ#1436217, BZ#1578955, BZ#1596050, BZ#1554310, BZ#1638852, BZ#1640477, BZ#1619620
<b>perl-IO-Socket-SSL</b>	BZ#1632600
<b>perl</b>	BZ#1511131
<b>pki-core</b>	BZ#1565073, BZ#1623444, BZ#1566360, BZ#1394069, <a href="#">BZ#1669257</a> , BZ#1656856, <a href="#">BZ#1673296</a>
<b>pykickstart</b>	BZ#1637872, BZ#1612061
<b>qemu-kvm</b>	BZ#1508139, BZ#1559240, BZ#1497911, BZ#1578855, BZ#1651994, BZ#1621817, BZ#1508137, BZ#1519004
<b>redhat-release</b>	BZ#1636338
<b>rsyslog</b>	BZ#1613880, BZ#1542497, BZ#1614179, BZ#1619645
<b>scap-security-guide</b>	BZ#1618528, BZ#1618518
<b>scap-workbench</b>	BZ#1640715
<b>selinux-policy</b>	BZ#1592244, BZ#1594111, BZ#1549772, BZ#1626446

コンポーネント	チケット
<b>setup</b>	BZ#1591969
<b>sos</b>	BZ#1559836
<b>sssd</b>	BZ#1448094, BZ#1382750, BZ#1642508, BZ#1620123
<b>subversion</b>	BZ#1571415
<b>swig-3.0-module</b>	BZ#1660051
<b>tomcatjss</b>	BZ#1424966, BZ#1636564
<b>tuned</b>	BZ#1576435, BZ#1565598
<b>valgrind</b>	BZ#1500481, BZ#1538009
<b>virt-manager</b>	BZ#1599777, BZ#1643609
<b>wpa_supplicant</b>	BZ#1582538
その他	BZ#1646563, BZ#1640697, BZ#1623712, BZ#1649404, BZ#1581198, BZ#1581990, BZ#1649497, BZ#1643294, BZ#1647612, BZ#1641015, BZ#1641032, BZ#1641004, BZ#1641034, BZ#1647110, BZ#1641007, BZ#1641029, BZ#1641022, BZ#1649493, BZ#1649922, BZ#1559616, BZ#1646541, BZ#1647725, <a href="#">BZ#1686057</a> , BZ#1582530, BZ#1581496, BZ#1650618, BZ#1650675, <a href="#">BZ#1650701</a> , JIRA:RHELPLAN-10347, JIRA:RHELPLAN-10439, JIRA:RHELPLAN-10440, JIRA:RHELPLAN-10442, JIRA:RHELPLAN-10443, JIRA:RHELPLAN-10438, JIRA:RHELPLAN-2878, JIRA:RHELPLAN-10355, JIRA:RHELPLAN-3010, JIRA:RHELPLAN-10352, JIRA:RHELPLAN-10353, JIRA:RHELPLAN-1212, JIRA:RHELPLAN-1473, JIRA:RHELPLAN-10445, JIRA:RHELPLAN-1499, JIRA:RHELPLAN-3001, JIRA:RHELPLAN-6746, JIRA:RHELPLAN-10354, JIRA:RHELPLAN-2896, JIRA:RHELPLAN-10304, JIRA:RHELPLAN-10628, JIRA:RHELPLAN-10441, JIRA:RHELPLAN-10444, JIRA:RHELPLAN-1842, JIRA:RHELPLAN-12764, JIRA:RHELPLAN-14607, BZ#1641014, BZ#1692746, BZ#1693775, BZ#1580387, BZ#1583620, BZ#1580430, BZ#1648843, BZ#1647908, BZ#1649891