



Red Hat Software Collections 3

3.8 リリースノート

Red Hat Software Collections 3.8 のリリースノート

Red Hat Software Collections 3.8 リリースノート

Red Hat Software Collections 3.8 のリリースノート

Lenka Špačková

Red Hat Customer Content Services

lspackova@redhat.com

Jaromír Hradílek

Red Hat Customer Content Services

Eliška Slobodová

Red Hat Customer Content Services

法律上の通知

Copyright © 2023 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

Red Hat Software Collections 3.8 リリースノートには、Red Hat Software Collections 3.8 の主な機能が記載されており、既知の問題に関する重要な情報が含まれています。Red Hat Developer Toolset コレクションは、Red Hat Developer Toolset リリースノート および Red Hat Developer Toolset ユーザーガイドに記載されています。

目次

多様性を受け入れるオープンソースの強化	4
第1章 RED HAT SOFTWARE COLLECTIONS 3.8	5
1.1. RED HAT SOFTWARE COLLECTIONS について	5
1.1.1. Red Hat Developer Toolset	5
1.2. 主な特長	5
1.3. RED HAT SOFTWARE COLLECTIONS 3.8 における変更点	15
1.3.1. 概要	15
アーキテクチャー	15
新しい Software Collections	16
Software Collections の更新	16
Red Hat Software Collections コンテナイメージ	16
1.3.2. Red Hat Developer Toolset 12 の変更点	16
1.3.2.1. Red Hat Developer Toolset 12.0 の変更点	16
1.3.2.2. Red Hat Developer Toolset 12.1 の変更点	17
1.3.3. Red Hat Developer Toolset 11 の変更点	17
1.3.4. nginx の変更点	18
1.3.5. Redis の変更点	18
1.3.6. JDK Mission Control の変更点	19
1.3.7. Apache httpd の変更 (非同期更新)	19
1.4. 互換性情報	19
1.5. 既知の問題	19
1.6. その他の注意事項	22
1.7. 非推奨の機能	24
第2章 インストール	26
2.1. RED HAT SOFTWARE COLLECTIONS へのアクセス	26
2.1.1. Red Hat Subscription Management の使用	26
2.1.2. Optional リポジトリからのパッケージ	27
2.2. RED HAT SOFTWARE COLLECTIONS のインストール	28
2.2.1. Software Collection の個別インストール	29
2.2.2. オプションパッケージのインストール	29
2.2.3. デバッグ情報のインストール	30
2.3. RED HAT SOFTWARE COLLECTIONS のアンインストール	30
2.4. RED HAT SOFTWARE COLLECTIONS の再構築	30
第3章 使用方法	31
3.1. RED HAT SOFTWARE COLLECTIONS の使用	31
3.1.1. Software Collection からの実行可能ファイルの実行	31
3.1.2. デフォルトで Software Collection を使用したシェルセッションの実行	31
3.1.3. Software Collection からのシステムサービスの実行	32
3.2. SOFTWARE COLLECTION からの手動ページへのアクセス	32
3.3. RED HAT SOFTWARE COLLECTIONS を使用するアプリケーションのデプロイ	32
3.4. RED HAT SOFTWARE COLLECTIONS コンテナイメージ	33
第4章 個別の SOFTWARE COLLECTIONS の詳細	35
4.1. RED HAT DEVELOPER TOOLSET	35
4.2. MAVEN	35
4.3. データベースコネクター	35
第5章 移行	38
5.1. MARIADB 10.5 への移行	38

5.1.1. rh-mariadb103 および rh-mariadb105 Software Collections 間の主な相違点	38
5.1.2. rh-mariadb103 から rh-mariadb105 Software Collection へのアップグレード	40
5.2. MYSQL 8.0 への移行	41
5.2.1. MySQL 5.7 と MySQL 8.0 間の主な相違点	41
rh-mysql80 ソフトウェアの収集に固有の相違点	41
MySQL 8.0 の一般的な変更点	42
5.2.2. rh-mysql80 Software Collection へのアップグレード	43
5.3. POSTGRESQL 13 への移行	43
5.3.1. Red Hat Enterprise Linux システムバージョンの PostgreSQL から PostgreSQL 13 Software Collection への移行	45
5.3.2. PostgreSQL 12 Software Collection から PostgreSQL 13 Software Collection への移行	48
5.4. NGINX 1.20 への移行	50
5.5. REDIS 6 への移行	50
互換性に関する注意事項	51
第6章 関連情報	52
6.1. RED HAT 製品ドキュメント	52
6.2. RED HAT 開発者	52
付録A 更新履歴	53

多様性を受け入れるオープンソースの強化

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

第1章 RED HAT SOFTWARE COLLECTIONS 3.8

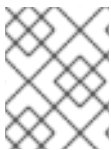
この章では、Red Hat Software Collections 3.8 のコンテンツセットの概要について説明しています。これは、コンポーネントとその説明、このバージョンの変更の概要、関連互換性情報のドキュメント、および既知の問題を一覧表示します。

1.1. RED HAT SOFTWARE COLLECTIONS について

アプリケーションによっては、最新の新機能を使用するために、より新しいバージョンのソフトウェアコンポーネントが必要になることがあります。**Red Hat Software Collections**は、基本的な Red Hat Enterprise Linux システムに含まれる同等のバージョンよりも新しい、またはこのシステムで最初に利用可能になった動的プログラミング言語、データベースサーバー、およびさまざまな関連パッケージのセットを提供する Red Hat 製品です。

Red Hat Software Collections 3.8 は、Red Hat Enterprise Linux 7 で利用できます。Red Hat Software Collections の一部として配布されるコンポーネントの一覧と、その機能の概要は「[主な特長](#)」を参照してください。

Red Hat Software Collections では、Red Hat Enterprise Linux 7 で提供されるデフォルトのシステムツールは置き換えられません。その代わりに、並列のツールセットが `/opt/` ディレクトリーにインストールされ、提供された **scl** ユーティリティーを使用してユーザーがアプリケーションごとにオプションで有効にできます。たとえば、Perl または PostgreSQL のデフォルトのバージョンは、ベース Red Hat Enterprise Linux システムが提供するバージョンのままになります。



注記

Red Hat Enterprise Linux 8 および Red Hat Enterprise Linux 9 では、同様のコンポーネントは [Application Streams](#) として提供されます。

すべての Red Hat Software Collections コンポーネントは、Red Hat Enterprise Linux サブスクリプション契約で完全にサポートされ、機能的に完全で、実稼働環境での使用を目的としています。重要なバグ修正とセキュリティエラーは、Red Hat Enterprise Linux と同様に、各メジャーバージョンのリリースから少なくとも 2 年間は Red Hat Software Collections サブスクライバーに発行されます。各メジャーリリースストリームでは、選択したコンポーネントの各バージョンは後方互換性を維持します。個別コンポーネントのサポート期間の詳細は、[Red Hat Software Collections Product Life Cycle](#) を参照してください。

1.1.1. Red Hat Developer Toolset

Red Hat Developer Toolsetは、個別の Software Collection として同梱される Red Hat Software Collections の一部です。Red Hat Developer Toolset の詳細は、[Red Hat Developer Toolset リリースノート](#) および [Red Hat Developer Toolset ユーザーガイド](#) を参照してください。

1.2. 主な特長

[表1.1 「Red Hat Software Collections のコンポーネント」](#) は、Red Hat Software Collections 3.8 のリリース時点でサポートされるコンポーネントのリストを示しています。すべての Software Collections は、Red Hat Enterprise Linux 7 でのみサポートされています。

表1.1 Red Hat Software Collections のコンポーネント

コンポーネント	Software Collection	説明
Red Hat Developer Toolset 12.1	devtoolset-12	Red Hat Developer Toolset は、Red Hat Enterprise Linux プラットフォームで作業する開発者向けに設計されています。 GNU Compiler Collection 、 GNU Debugger 、その他の開発用ツールやデバッグ用ツール、およびパフォーマンス監視ツールの現行バージョンを提供します。コンポーネントの完全なリストは、『Red Hat Developer Toolset ユーザーガイド』の Red Hat Developer Toolset コンポーネント の表を参照してください。
Red Hat Developer Toolset 12.1	devtoolset-12	Red Hat Developer Toolset は、Red Hat Enterprise Linux プラットフォームで作業する開発者向けに設計されています。 GNU Compiler Collection 、 GNU Debugger 、その他の開発用ツールやデバッグ用ツール、およびパフォーマンス監視ツールの現行バージョンを提供します。コンポーネントの完全なリストは、『Red Hat Developer Toolset ユーザーガイド』の Red Hat Developer Toolset コンポーネント の表を参照してください。
Perl 5.30.1	rh-perl530	Perl のリリース。これは、システム管理ユーティリティと Web プログラミングに一般的に使用される高レベルのプログラミング言語です。rh-perl530 Software Collection は、 MySQL 、 PostgreSQL 、および SQLite 用の追加のユーティリティ、スクリプト、およびデータベースコネクタを提供します。これには、httpd24 Software Collection でのみサポートされる Perl モジュール DateTime と Apache httpd モジュール mod_perl が含まれます。さらに、CPAN モジュールを簡単にインストールできる cpanm ユーティリティ、HTTP サーバーと通信するための LWP::UserAgent モジュール、通信のセキュリティを保護する LWP::Protocol::https モジュールが提供されます。パッケージは rh-perl530 アップストリームに合わせて調整されます。インタープリターは perl530-perl パッケージにより提供されますが、perl-interpreter パッケージはコアモジュールもインストールします。
PHP 7.3.33	rh-php73	PEAR 1.10.9、APCu 5.1.17、および Xdebug 拡張が含まれる PHP 7.3 のリリース
Python 3.8.14	rh-python38	rh-python38 Software Collection には Python 3.8 が含まれ、 新しい Python モジュール (contextvars、dataclasses、または importlib.resources、新しい言語機能、開発者エクスペリエンスの向上、パフォーマンスの向上) が導入されています。さらに、 mod_wsgi (httpd24 Software Collection とのみサポート)、 numpy 、 scipy 、 psycopg2 PostgreSQL データベースコネクタなど、一般的な拡張ライブラリーのセットが提供されます。

コンポーネント	Software Collection	説明
Ruby 3.0.4	rh-ruby30	Ruby 3.0. のリリースこのバージョンでは、複数のパフォーマンスの向上と新機能が提供されます (例: Ractor 、 Fiber Scheduler 、および RBS 言語)。Ruby 3.0 は、Ruby 2.7 とソースレベルの後方互換性を維持しています。
MariaDB 10.5.16	rh-mariadb105	Red Hat Enterprise Linux ユーザー向けの MySQL の代替となる MariaDB のリリース。あらゆる実用的な目的で、MySQL は MariaDB とバイナリー互換性があり、データ変換なしで MySQL と置き換えることができます。このバージョンには、さまざまな新機能が含まれ、バージョン 4 にアップグレードされた MariaDB Galera Cluster 、および PAM プラグインバージョン 2.0 が含まれます。
MySQL 8.0.32	rh-mysql80	MySQL サーバーのリリース。 セキュリティーおよびアカウント管理機能 および 新規機能 および 機能強化 が数多く追加されました。
PostgreSQL 10.23	rh-postgresql10	PostgreSQL のリリースには、パフォーマンスが大幅に向上し、 publish および subscribe キーワードを使用した 論理レプリケーション 、 SCRAM-SHA-256 メカニズムに基づく 強力なパスワード認証 などの新機能が多数含まれています。
PostgreSQL 12.11	rh-postgresql12	PostgreSQL のリリースは、 pgaudit 拡張 、 パーティション および 並列処理のさまざまな機能強化 、 SQL/JSON パス言語のサポート 、および パフォーマンスの向上 を提供します。
PostgreSQL 13.7	rh-postgresql13	PostgreSQL のリリースにより、 クエリー計画が向上 し、さまざまなパフォーマンス改善 と、 pg_repack および ppython3 の 2 つの 新しいパッケージ が導入されました。
Node.js 14.21.3	rh-nodejs14	Node.js の Node.js リリース (V8 バージョン 8.3 、実験的な新しい WebAssembly System Interface(WASI) 、および実験的な新しい Async Local Storage API)。
nginx 1.20.1	rh-nginx120	nginx のリリース、Web サーバーおよびプロキシサーバーは、高い同時実行性、パフォーマンス、およびメモリー使用量の少ない部分に重点を置いています。このバージョンは、 OCSP によるクライアント SSL 証明書の検証をサポート し、 HTTP/2 のサポートを改善 します。
Apache httpd 2.4.34	httpd24	Apache HTTP Server (httpd) のリリース。これには、高パフォーマンスの イベントベースの処理モデル 、 強化された SSL モジュール 、および FastCGI サポート が含まれます。 mod_auth_kerb 、 mod_auth_mellon 、および ModSecurity モジュールも含まれています。

コンポーネント	Software Collection	説明
Varnish Cache 6.0.8	rh-varnish6	高パフォーマンスの HTTP リバースプロキシである Varnish Cache のリリース。このバージョンには、 Unix ドメインソケットのサポート (クライアントおよびバックエンドサーバーの両方) 、 VCL 言語の新規レベル (vcl4.1) 、および改善した HTTP/2 サポート が含まれます。
Maven 3.6.1	rh-maven36	Maven のリリース (ソフトウェアプロジェクト管理および内包表記ツール)。本リリースでは、さまざまな機能強化およびバグ修正が提供されます。
Git 2.27.0	rh-git227	分散アーキテクチャーを備えた分散リビジョン管理システムである Git のリリース。クライアントサーバーモデルを使用する集中型バージョン管理システムとは対照的に、Git は Git リポジトリの各作業コピーが完全なリビジョン履歴で正確なコピーになるようにします。このバージョンでは、 git checkout コマンドを git switch や git restore に分割し、git rebase コマンドの挙動を変更するなど、多くの機能拡張が追加されました 。さらに、Git Large File Storage (LFS) がバージョン 2.11.0 に更新されました。
Redis 6.0.16	rh-redis6	Redis 6.0 のリリース (永続キーと値のデータベース) です。Redis が、 すべてのチャンネル、アクセス制御リスト、および Redis Serialization Protocol version 3 で SSL をサポートするようになり ました。
HAProxy 1.8.24	rh-haproxy18	HAProxy 1.8 のリリース (TCP および HTTP ベースのアプリケーションの信頼できる高パフォーマンスな ネットワークロードバランサー) です。
JDK Mission Control 8.0.1	rh-jmc	この Software Collection には、HotSpot JVM の強力なプロファイラーである JDK Mission Control (JMC) が含まれています。JMC では、JDK Flight Recorder が収集した豊富なデータを効率的かつ詳細に分析するための高度なツールセットを利用できます。JMC を実行するには、JDK バージョン 11 以降が必要であることに注意してください。ターゲット Java アプリケーションは、最低でも OpenJDK バージョン 8 で実行する必要があります。これにより、JMC が JDK Flight Recorder 機能にアクセスできるようになります。rh-jmc Software Collection には、rh-maven36 Software Collection が必要です。

これまでリリースされた Software Collections は同じディストリビューションチャンネルで引き続き利用できます。終了したコンポーネントを含む Software Collections はすべて、[表1.2 「利用可能なすべての Software Collections」](#) に記載されています。サポートされなくなった Software Collections にはアスタリスク (*) が付いています。

個々のコンポーネントのサポート期間の詳細は、[Red Hat Software Collections Product Life Cycle](#) を参照してください。以前にリリースされたコンポーネントの詳細は、Red Hat Software Collections の以前のバージョンの[Release Notes](#)を参照してください。

表1.2 利用可能なすべての Software Collections

コンポーネント	Software Collection	可用性	RHEL7でサポートされるアーキテクチャー
Red Hat Developer Toolset 12 リリース			
Red Hat Developer Toolset 12.1	devtoolset-12	RHEL7	x86_64、s390x、ppc64、ppc64le
Red Hat Software Collections 3.8 の新しいコンポーネント			
Red Hat Developer Toolset 11.0	devtoolset-11	RHEL7	x86_64、s390x、ppc64、ppc64le
nginx 1.20.1	rh-nginx120	RHEL7	x86_64、s390x、ppc64le
Redis 6.0.16	rh-redis6	RHEL7	x86_64、s390x、ppc64le
Red Hat Software Collections 3.8 で更新されたコンポーネント			
JDK Mission Control 8.0.1	rh-jmc	RHEL7	x86_64
Red Hat Software Collections 3.7 で最後に更新されたコンポーネント			
Red Hat Developer Toolset 10.1	devtoolset-10*	RHEL7	x86_64、s390x、ppc64、ppc64le
MariaDB 10.5.16	rh-mariadb105	RHEL7	x86_64、s390x、ppc64le
PostgreSQL 13.7	rh-postgresql13	RHEL7	x86_64、s390x、ppc64le
Ruby 3.0.4	rh-ruby30	RHEL7	x86_64、s390x、ppc64le
Ruby 2.7.8	rh-ruby27*	RHEL7	x86_64、s390x、aarch64、ppc64le
Ruby 2.6.10	rh-ruby26*	RHEL7	x86_64、s390x、aarch64、ppc64le
Red Hat Software Collections 3.6 で最後に更新されたコンポーネント			
Git 2.27.0	rh-git227	RHEL7	x86_64、s390x、ppc64le
nginx 1.18.0	rh-nginx118*	RHEL7	x86_64、s390x、ppc64le
Node.js 14.21.3	rh-nodejs14	RHEL7	x86_64、s390x、ppc64le

コンポーネント	Software Collection	可用性	RHEL7でサポートされるアーキテクチャー
Apache httpd 2.4.34	httpd24	RHEL7	x86_64, s390x, aarch64, ppc64le
PHP 7.3.33	rh-php73	RHEL7	x86_64, s390x, aarch64, ppc64le
HAProxy 1.8.24	rh-haproxy18	RHEL7	x86_64
Perl 5.30.1	rh-perl530	RHEL7	x86_64, s390x, aarch64, ppc64le
Ruby 2.5.9	rh-ruby25*	RHEL7	x86_64, s390x, aarch64, ppc64le
Red Hat Software Collections 3.5 で最後に更新されたコンポーネント			
Red Hat Developer Toolset 9.1	devtoolset-9*	RHEL7	x86_64, s390x, aarch64, ppc64, ppc64le
Python 3.8.14	rh-python38	RHEL7	x86_64, s390x, aarch64, ppc64le
Varnish Cache 6.0.8	rh-varnish6	RHEL7	x86_64, s390x, aarch64, ppc64le
Apache httpd 2.4.34 (RHEL6の最終更新)	httpd24 (RHEL6)*	RHEL6	x86_64
Red Hat Software Collections 3.4 で最後に更新されたコンポーネント			
Node.js 12.22.12	rh-nodejs12*	RHEL7	x86_64, s390x, aarch64, ppc64le
nginx 1.16.1	rh-nginx116*	RHEL7	x86_64, s390x, aarch64, ppc64le
PostgreSQL 12.11	rh-postgresql12	RHEL7	x86_64, s390x, aarch64, ppc64le
Maven 3.6.1	rh-maven36	RHEL7	x86_64, s390x, aarch64, ppc64le
Red Hat Software Collections 3.3 で最後に更新されたコンポーネント			
Red Hat Developer Toolset 8.1	devtoolset-8*	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64, ppc64le

コンポーネント	Software Collection	可用性	RHEL7でサポートされるアーキテクチャー
MariaDB 10.3.35	rh-mariadb103*	RHEL7	x86_64、s390x、aarch64、ppc64le
Redis 5.0.5	rh-redis5*	RHEL7	x86_64、s390x、aarch64、ppc64le
Red Hat Software Collections 3.2 で最後に更新されたコンポーネント			
PHP 7.2.24	rh-php72*	RHEL7	x86_64、s390x、aarch64、ppc64le
MySQL 8.0.32	rh-mysql80	RHEL7	x86_64、s390x、aarch64、ppc64le
Node.js 10.21.0	rh-nodejs10*	RHEL7	x86_64、s390x、aarch64、ppc64le
nginx 1.14.1	rh-nginx114*	RHEL7	x86_64、s390x、aarch64、ppc64le
Git 2.18.4	rh-git218*	RHEL7	x86_64、s390x、aarch64、ppc64le
Red Hat Software Collections 3.1 で最後に更新されたコンポーネント			
Red Hat Developer Toolset 7.1	devtoolset-7*	RHEL6、RHEL7	x86_64、s390x、aarch64、ppc64、ppc64le
Perl 5.26.3	rh-perl526*	RHEL7	x86_64、s390x、aarch64、ppc64le
MongoDB 3.6.3	rh-mongodb36*	RHEL7	x86_64、s390x、aarch64、ppc64le
Varnish Cache 5.2.1	rh-varnish5*	RHEL7	x86_64、s390x、aarch64、ppc64le
PostgreSQL 10.23	rh-postgresql10	RHEL7	x86_64、s390x、aarch64、ppc64le
PHP 7.0.27	rh-php70*	RHEL6、RHEL7	x86_64
MySQL 5.7.24	rh-mysql57*	RHEL6、RHEL7	x86_64、s390x、aarch64、ppc64le
Red Hat Software Collections 3.0 で最後に更新されたコンポーネント			

コンポーネント	Software Collection	可用性	RHEL7でサポートされるアーキテクチャー
PHP 7.1.8	rh-php71*	RHEL7	x86_64、s390x、aarch64、ppc64le
nginx 1.12.1	rh-nginx112*	RHEL7	x86_64、s390x、aarch64、ppc64le
Python 3.6.12	rh-python36*	RHEL6、RHEL7	x86_64、s390x、aarch64、ppc64le
Maven 3.5.0	rh-maven35*	RHEL7	x86_64、s390x、aarch64、ppc64le
MariaDB 10.2.22	rh-mariadb102*	RHEL6、RHEL7	x86_64、s390x、aarch64、ppc64le
PostgreSQL 9.6.19	rh-postgresql96*	RHEL6、RHEL7	x86_64、s390x、aarch64、ppc64le
MongoDB 3.4.9	rh-mongodb34*	RHEL6、RHEL7	x86_64、s390x、aarch64、ppc64le
Node.js 8.17.0	rh-nodejs8*	RHEL7	x86_64、s390x、aarch64、ppc64le
Red Hat Software Collections 2.4 で最後に更新されたコンポーネント			
Red Hat Developer Toolset 6.1	devtoolset-6*	RHEL6、RHEL7	x86_64、s390x、aarch64、ppc64、ppc64le
Scala 2.10.6	rh-scala210*	RHEL7	x86_64
nginx 1.10.2	rh-nginx110*	RHEL6、RHEL7	x86_64
Node.js 6.11.3	rh-nodejs6*	RHEL6、RHEL7	x86_64、s390x、aarch64、ppc64le
Ruby 2.4.6	rh-ruby24*	RHEL6、RHEL7	x86_64
Ruby on Rails 5.0.1	rh-ror50*	RHEL6、RHEL7	x86_64
Eclipse 4.6.3	rh-eclipse46*	RHEL7	x86_64
Python 2.7.18	python27*	RHEL6*、RHEL7	x86_64、s390x、aarch64、ppc64le

コンポーネント	Software Collection	可用性	RHEL7でサポートされるアーキテクチャー
Thermostat 1.6.6	rh-thermostat16*	RHEL6、RHEL7	x86_64
Maven 3.3.9	rh-maven33*	RHEL6、RHEL7	x86_64
一般的な Java パッケージ	rh-java-common*	RHEL6、RHEL7	x86_64
Red Hat Software Collections 2.3 で最後に更新されたコンポーネント			
Git 2.9.3	rh-git29*	RHEL6、RHEL7	x86_64、s390x、aarch64、ppc64le
Redis 3.2.4	rh-redis32*	RHEL6、RHEL7	x86_64
Perl 5.24.0	rh-perl524*	RHEL6、RHEL7	x86_64
Python 3.5.1	rh-python35*	RHEL6、RHEL7	x86_64
MongoDB 3.2.10	rh-mongodb32*	RHEL6、RHEL7	x86_64
Ruby 2.3.8	rh-ruby23*	RHEL6、RHEL7	x86_64
PHP 5.6.25	rh-php56*	RHEL6、RHEL7	x86_64
Red Hat Software Collections 2.2 で最後に更新されたコンポーネント			
Red Hat Developer Toolset 4.1	devtoolset-4*	RHEL6、RHEL7	x86_64
MariaDB 10.1.29	rh-mariadb101*	RHEL6、RHEL7	x86_64
MongoDB 3.0.11 アップグレードコレクション	rh-mongodb30upg*	RHEL6、RHEL7	x86_64
Node.js 4.6.2	rh-nodejs4*	RHEL6、RHEL7	x86_64
PostgreSQL 9.5.14	rh-postgresql95*	RHEL6、RHEL7	x86_64
Ruby on Rails 4.2.6	rh-ror42*	RHEL6、RHEL7	x86_64
MongoDB 2.6.9	rh-mongodb26*	RHEL6、RHEL7	x86_64
Thermostat 1.4.4	thermostat1*	RHEL6、RHEL7	x86_64
Red Hat Software Collections 2.1 で最後に更新されたコンポーネント			

コンポーネント	Software Collection	可用性	RHEL7でサポートされるアーキテクチャー
Varnish Cache 4.0.3	rh-varnish4*	RHEL6、RHEL7	x86_64
nginx 1.8.1	rh-nginx18*	RHEL6、RHEL7	x86_64
Node.js 0.10	nodejs010*	RHEL6、RHEL7	x86_64
Maven 3.0.5	maven30*	RHEL6、RHEL7	x86_64
V8 3.14.5.10	v8314*	RHEL6、RHEL7	x86_64
Red Hat Software Collections 2.0 で最後に更新されたコンポーネント			
Red Hat Developer Toolset 3.1	devtoolset-3*	RHEL6、RHEL7	x86_64
Perl 5.20.1	rh-perl520*	RHEL6、RHEL7	x86_64
Python 3.4.2	rh-python34*	RHEL6、RHEL7	x86_64
Ruby 2.2.9	rh-ruby22*	RHEL6、RHEL7	x86_64
Ruby on Rails 4.1.5	rh-ror41*	RHEL6、RHEL7	x86_64
MariaDB 10.0.33	rh-mariadb100*	RHEL6、RHEL7	x86_64
MySQL 5.6.40	rh-mysql56*	RHEL6、RHEL7	x86_64
PostgreSQL 9.4.14	rh-postgresql94*	RHEL6、RHEL7	x86_64
Passenger 4.0.50	rh-passenger40*	RHEL6、RHEL7	x86_64
PHP 5.4.40	php54*	RHEL6、RHEL7	x86_64
PHP 5.5.21	php55*	RHEL6、RHEL7	x86_64
nginx 1.6.2	nginx16*	RHEL6、RHEL7	x86_64
DevAssistant 0.9.3	devassist09*	RHEL6、RHEL7	x86_64
Red Hat Software Collections 1 で最後に更新されたコンポーネント			
Git 1.9.4	git19*	RHEL6、RHEL7	x86_64
Perl 5.16.3	perl516*	RHEL6、RHEL7	x86_64

コンポーネント	Software Collection	可用性	RHEL7でサポートされるアーキテクチャー
Python 3.3.2	python33*	RHEL6、RHEL7	x86_64
Ruby 1.9.3	ruby193*	RHEL6、RHEL7	x86_64
Ruby 2.0.0	ruby200*	RHEL6、RHEL7	x86_64
Ruby on Rails 4.0.2	ror40*	RHEL6、RHEL7	x86_64
MariaDB 5.5.53	mariadb55*	RHEL6、RHEL7	x86_64
MongoDB 2.4.9	mongodb24*	RHEL6、RHEL7	x86_64
MySQL 5.5.52	mysql55*	RHEL6、RHEL7	x86_64
PostgreSQL 9.2.18	postgresql92*	RHEL6、RHEL7	x86_64

説明:

- RHEL6 – Red Hat Enterprise Linux 6
- RHEL7 – Red Hat Enterprise Linux 7
- x86_64: AMD および Intel 64 ビットアーキテクチャー
- s390x: 64 ビット IBM Z アーキテクチャー
- aarch64 - 64 ビット ARM アーキテクチャー
- ppc64 - IBM POWER、ビッグエンディアン
- ppc64le - IBM POWER、リトルエンディアン
- * - 終了したコンポーネント (この Software Collection はサポート対象外になりました)

上記の表には、非同期更新で利用できる最新バージョンがリスト表示されます。

Red Hat Software Collections 2.0 以降でリリースされた Software Collections には、その名前に **rh-** 接頭辞が含まれていることに注意してください。

Eclipse は、[Red Hat Developer Tools](#) 製品から入手できます。

1.3. RED HAT SOFTWARE COLLECTIONS 3.8 における変更点

1.3.1. 概要

アーキテクチャー

Red Hat Software Collections 製品には、以下のアーキテクチャーで実行している Red Hat Enterprise Linux 7 のパッケージが含まれます。

- AMD アーキテクチャーおよび Intel 64 ビットアーキテクチャー
- 64 ビット IBM Z
- IBM POWER、リトルエンディアン

コンポーネントの完全なリストとそれらの可用性は、[表1.2「利用可能なすべての Software Collections」](#) を参照してください。

新しい Software Collections

Red Hat Software Collections 3.8 では、以下の新しい Software Collections を追加しています。

- devtoolset-12 (2022 年 11 月以降に利用可能) – [「Red Hat Developer Toolset 12 の変更点」](#) を参照
- devtoolset-11 – [「Red Hat Developer Toolset 11 の変更点」](#) を参照してください。
- rh-nginx120 – [「nginx の変更点」](#) を参照してください。
- rh-redis6 – [「Redis の変更点」](#) を参照してください。

すべての新しい Software Collections は、Red Hat Enterprise Linux 7 でのみ利用できます。

Software Collections の更新

Red Hat Software Collections 3.8 では、以下のコンポーネントが更新されました。

- rh-jmc – [「JDK Mission Control の変更点」](#) を参照してください。
- httpd24 (非同期更新) – [「Apache httpd の変更 \(非同期更新\)」](#) を参照してください。

Red Hat Software Collections コンテナイメージ

以下のコンテナイメージは、Red Hat Software Collections 3.8 の新機能です。

- rhsc/devtoolset-12-toolchain-rhel7 (2022 年 11 月以降に利用可能)
- rhsc/devtoolset-12-perftools-rhel7 (2022 年 11 月以降に利用可能)
- rhsc/nginx-120-rhel7
- rhsc/redis-6-rhel7

Red Hat Software Collections コンテナイメージの詳細は、[「Red Hat Software Collections コンテナイメージ」](#) を参照してください。

1.3.2. Red Hat Developer Toolset 12 の変更点

1.3.2.1. Red Hat Developer Toolset 12.0 の変更点

以前のリリースと比較すると、以下のコンポーネントが Red Hat Developer Toolset 12.0 でアップグレードされました。

- GCC: バージョンを 12.1.1 へ
- annobin: バージョン 10.76 へ
- elfutils: バージョン 0.187 へ

- **GDB**: バージョン 11.2 へ
- **strace**: バージョン 5.18 へ
- **SystemTap**: バージョン 4.7 へ
- **Valgrind**: バージョン 3.19.0 へ
- **Dyninst**: バージョン 12.1.0 へ

また、**binutils** にバグ修正の更新を利用できます。

Red Hat Developer Toolset 12.0 の詳細な変更は、[Red Hat Developer Toolset ユーザーガイド](#) を参照してください。

1.3.2.2. Red Hat Developer Toolset 12.1 の変更点

Red Hat Developer Toolset 12.1 では、以下のコンポーネントがアップグレードされました。

- **GCC**: バージョンを 12.2.1 へ
- **annobin**: バージョン 11.08 へ

さらに、**binutils** のセキュリティー更新プログラムも利用可能です。

Red Hat Developer Toolset 12.1 の詳細は、[Red Hat Developer Toolset User Guide](#) を参照してください。

1.3.3. Red Hat Developer Toolset 11 の変更点

以前のリリースと比較すると、以下のコンポーネントが Red Hat Developer Toolset 12.1 でアップグレードされました。

- **GCC**: バージョンを 11.2.1 へ
- **binutils**: バージョンを 2.36.1 へ
- **elfutils**: バージョンを 0.185 へ
- **dwz**: バージョンを 0.14 へ
- **GDB**: バージョンを 10.2 へ
- **strace**: バージョンを 5.13 へ
- **SystemTap**: バージョンを 4.5 へ
- **Valgrind**: バージョンを 3.17.0 へ
- **Dyninst**: バージョンを 11.0.0 へ
- **make**: バージョンを 4.3 へ
- **annobin**: バージョンを 9.82 へ

12.1 の変更に関する詳細は、[Red Hat Developer Toolset ユーザーガイド](#) を参照してください。

1.3.4. nginx の変更点

新しい rh-nginx120 Software Collection では **nginx 1.20.1** が導入され、バージョン 1.18 に対するバグ修正およびセキュリティ修正が数多く追加されました。

新機能:

- **nginx** が、OCSP (Online Certificate Status Protocol) を使用したクライアント SSL 証明書の検証に対応するようになりました。
- **nginx** が、最小限の空き領域に基づくキャッシュクリアに対応するようになりました。これに対応するのは、**proxy_cache_path** ディレクティブの **min_free** パラメーターとして実装されています。
- 新しい **ngx_stream_set_module** モジュールが追加されました。これにより、変数の値を設定できるようになりました。

拡張されたディレクティブ:

- **ssl_conf_command**、**ssl_reject_handshake** など、新しいディレクティブが複数利用できるようになりました。
- **proxy_cookie_flags** ディレクティブが変数に対応するようになりました。

HTTP/2 のサポートが改善されました。

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

nginx の変更に関する詳細は、[upstream release notes](#) を参照してください。

移行手順は、「[nginx 1.20 への移行](#)」を参照してください。

1.3.5. Redis の変更点

新しい rh-redis6 Software Collection には **Redis 6.0.16** が含まれています。このバージョンでは、以前のバージョンの Red Hat Software Collections リリースとともに配布されるバージョン 5.0.5 に対する機能拡張およびバグ修正が複数追加されました。**Redis 5** への主な変更点は、以下の通りです。

- **Redis** がすべてのチャンネルで SSL をサポートするようになりました。
- **Redis** が、コマンド呼び出しおよびキーパターンアクセスのためのユーザーパーミッションを定義するアクセス制御リスト (ACL) をサポートするようになりました。
- **Redis** が REdis Serialization Protocol バージョン 3 (RESP3) をサポートするようになりました。これにより、より多くのセマンティック応答が返されます。
- **Redis** がオプションでスレッドを使用して I/O を処理できるようになりました。
- **Redis** は、クライアント側のキー値のキャッシュにサーバー側のサポートを提供するようになりました。

- **Redis** のアクティブな期限切れサイクルが改善され、期限切れの鍵のエビクションが速くなりました。

移行および互換性の注意事項については、「[Redis 6 への移行](#)」を参照してください。

Redis の詳細な変更は、[アップストリームのリリースノート](#) を参照してください。

1.3.6. JDK Mission Control の変更点

rh-jmc Software Collection が提供する JDK Mission Control (JMC) がバージョン 8.0.1 にアップグレードされ、バグ修正およびセキュリティ修正が提供されます。

1.3.7. Apache httpd の変更 (非同期更新)

[CVE-2022-29404](#) を修正するために、[RHSA-2022:6753](#) アドバイザリーのリリースにより、Apache HTTP Server の **LimitRequestBody** ディレクティブのデフォルト値が **0** (無制限) から 1 GiB に変更されました。

LimitRequestBody の値が **httpd** 設定ファイルで明示的に指定されていないシステムでは、**httpd** パッケージを更新すると、**LimitRequestBody** がデフォルト値の 1 GiB に設定されます。その結果、HTTP 要求本文の合計サイズがデフォルトの制限である 1 GiB を超える場合、**httpd** は **413 Request Entity Too Large** エラーコードを返します。

HTTP 要求メッセージ本文の新しいデフォルトの許容サイズがユースケースに不十分な場合は、それぞれのコンテキスト (サーバー、ディレクトリーごと、ファイルごと、または場所ごと) 内で **httpd** 設定ファイルを更新し、適切な制限をバイト単位で設定します。たとえば、新しい制限を 2 GiB に設定するには、次を使用します。

```
LimitRequestBody 2147483648
```

すでに **LimitRequestBody** ディレクティブで明示的な値を使用するように設定されているシステムは、この変更の影響を受けません。

1.4. 互換性情報

Red Hat Software Collections 3.8 は、AMD および Intel 64 ビットアーキテクチャー、64 ビット IBM Z および IBM POWER (リトルエンディアン) 上の Red Hat Enterprise Linux 7 のサポートされるすべてのリリースで利用できます。

64 ビット ARM アーキテクチャーで、以前にリリースされた一部のコンポーネントも利用できます。

利用可能なコンポーネントのリストは、[表1.2 「利用可能なすべての Software Collections」](#) を参照してください。

1.5. 既知の問題

rh-mariadb component

rh-mariadb103 Software Collection は、PAM (Pluggable Authentication Modules) プラグインバージョン 1.0 を提供します。rh-mariadb105 Software Collection は、プラグインバージョン 1.0 および 2.0 を提供します。バージョン 2.0 がデフォルトです。**MariaDB** の PAM プラグインバージョン 1.0 は機能しません。この問題を回避するには、rh-mariadb105 が提供する PAM プラグインバージョン 2.0 を使用します。

rh-ruby27 コンポーネント (BZ#1836201)

カスタムスクリプトに Psych YAML パーサーを必要とし、その後に **Gem.load_yaml** メソッドを使用すると、スクリプトの実行に失敗し、以下のエラーメッセージが表示されます。

```
superclass mismatch for class Mark (TypeError)
```

この問題を回避するには、スクリプトの **require 'psych'** 行の上のどこかに、**gem 'psych'** 行を追加します。

```
...
gem 'psych'
...
require 'psych'
Gem.load_yaml
```

複数のコンポーネント (BZ#1716378)

debuginfo Software Collections パッケージが提供する特定のファイルは、ベースの Red Hat Enterprise Linux システムまたはその他のバージョンの Red Hat Software Collections コンポーネントに対応する debuginfo パッケージファイルと競合する可能性があります。たとえば、python27-python-debuginfo パッケージファイルは、コアシステムにインストールされている python-debuginfo パッケージの対応するファイルと競合する可能性があります。同様に、httpd24-mod_auth_mellon-debuginfo パッケージのファイルは、ベースシステム mod_auth_mellon-debuginfo パッケージが提供する同様のファイルと競合する可能性があります。この問題を回避するには、Software Collection debuginfo パッケージをインストールする前に、ベースシステムの debuginfo パッケージをアンインストールします。

rh-mysql80 コンポーネント (BZ#1646363)

mysql-connector-java データベースコネクタは MySQL 8.0 サーバーと動作しません。この問題を回避するには、rh-mariadb103 Software Collection からの **mariadb-java-client** データベースコネクタを使用します。

rh-mysql80 コンポーネント (BZ#1646158)

MySQL 8.0 ではデフォルトの文字セットが **utf8mb4** に変更になりましたが、この文字セットは **php-mysqldb** データベースコネクタではサポートされません。これにより、**php-mysqldb** がデフォルト設定で接続できません。この問題を回避するには、既知の文字セットを MySQL サーバー設定のパラメーターとして指定します。たとえば、**/etc/opt/rh/rh-mysql80/my.cnf.d/mysql-server.cnf** ファイルを read に変更します。

```
[mysqld]
character-set-server=utf8
```

httpd24 コンポーネント (BZ#1429006)

httpd 2.4.27 以降、**mod_http2** モジュールはデフォルトの **prefork** Multi-Processing Module (MPM) でサポートされなくなりました。HTTP/2 サポートを有効にするには、**/opt/rh/httpd24/root/etc/httpd/conf.modules.d/00-mpm.conf** で設定ファイルを編集し、**event** または **worker** MPM に切り替えます。

HTTP/2 の server-push 機能は、64 ビット ARM アーキテクチャー、64 ビット IBM Z、および IBM POWER (リトルエンディアン) では機能しないことに注意してください。

httpd24 コンポーネント、BZ#1224763

FastCGI Process Manager (PHP-FPM) で **mod_proxy_fcgi** モジュールを使用する場合は、**httpd** は正しいポート **9000** ではなく、デフォルトで FastCGI プロトコルのポート **8000** を使用します。この問題を回避するには、正しいポートを明示的に指定します。

httpd24 コンポーネント、BZ#1382706

SELinux が有効になっている場合、**LD_LIBRARY_PATH** 環境変数は、**httpd** によって呼び出される CGI スクリプトには渡されません。そのため、**httpd** が実行する CGI スクリプトの **/opt/rh/httpd24/service-environment** ファイルで有効にした Software Collections から実行ファイルを読み出す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、CGI スクリプト内から希望どおりに **LD_LIBRARY_PATH** を設定します。

httpd24 コンポーネント

httpd24 Software Collection からの Apache Portable Runtime (APR) および APR-util ライブラリーに対する外部アプリケーションのコンパイルはサポートされていません。**LD_LIBRARY_PATH** 環境変数は、この Software Collection のいずれのアプリケーションでも不要であるため、httpd24 に設定されていません。

scl-utils コンポーネント (

Red Hat Enterprise Linux 7.5 以前では、scl-utils パッケージのアーキテクチャー固有のマクロバグにより、**<collection>/root/usr/lib64/** ディレクトリーには 64 ビット ARM アーキテクチャーおよび IBM POWER (リトルエンディアン) のパッケージの所有権が正しくありません。したがって、Software Collection がアンインストールされると、このディレクトリーは削除されません。この問題を回避するには、Software Collection を削除する際に **<collection>/root/usr/lib64/** を手動で削除します。

maven コンポーネント)

ユーザーが Red Hat Enterprise Linux システムバージョンの maven-local パッケージおよび rh-maven*-maven-local パッケージの両方をインストールすると、Java RPM パッケージを構築するツールである **XMvn** を Maven Software Collection から実行すると、ベースシステムから設定ファイルを読み取ろうとし、失敗します。この問題を回避するには、ベースの Red Hat Enterprise Linux システムから maven-local パッケージをアンインストールします。

perl コンポーネント

複数の **mod_perl.so** ライブラリーをインストールすることはできません。したがって、複数の Perl Software Collection から **mod_perl** モジュールを使用することはできません。

httpd, mariadb, mysql, nodejs, perl, php, python, and ruby components, BZ#1072319

httpd24、rh-mariadb*、rh-mysql*、rh-nodejs*、rh-perl*、rh-php*、python27、rh-python* または rh-ruby* パッケージをアンインストールする際は、依存パッケージの所有権により、アンインストールの順序には関連性があります。そのため、一部のディレクトリーおよびファイルはシステム上に残される可能性があるため、削除されない可能性があります。

mariadb、mysql コンポーネント (BZ#1194611)

MariaDB 10 および MySQL 5.6 以降、rh-mariadb*-mariadb-server および rh-mysql*-mysql-server パッケージでは、デフォルトで **test** データベースが提供されなくなりました。このデータベースは初期化中には作成されませんが、付与テーブルは、**test** がデフォルトで作成されたものと同じ値で事前に入力されます。その結果、**test** または **test_*** データベースが後で作成されると、これらのデータベースへのアクセス権限は、新規データベースのデフォルトよりも制限されません。

また、ベンチマークを実行している場合、**run-all-tests** スクリプトは、サンプルパラメーターを使用してもそのままでは機能しません。テストを実行する前にテストデータベースを作成し、**--database** パラメーターにデータベース名を指定する必要があります。パラメーターが指定されてい

ない場合、**test** はデフォルトで取得されますが、**test** データベースが存在することを確認する必要があります。

mariadb, mysql, postgresql コンポーネント

Red Hat Software Collections には、**MySQL 8.0**、**MariaDB 10.3**、**MariaDB 10.5**、**PostgreSQL 10**、**PostgreSQL 12** および **PostgreSQL 13** のデータベースサーバーが含まれます。コア Red Hat Enterprise Linux 7 は、**MariaDB** および **PostgreSQL** データベース (クライアントライブラリーおよびデーモン) の以前のバージョンを提供します。クライアントライブラリーは、動的言語、ライブラリーなどのデータベースコネクタにも使用されます。

PostgreSQL コンポーネントの Red Hat Software Collections データベースパッケージにパッケージ化されたクライアントライブラリーは、サーバーユーティリティーおよびデーモンの目的にのみ含まれているため、使用する予定はありません。代わりに、ユーザーはコアシステムで提供されるシステムライブラリーとデータベースコネクタを使用することが想定されます。

クライアントライブラリーとデーモンの間で使用されるプロトコルは、データベースのバージョンが変わっても安定しているため、たとえば、**PostgreSQL 12** または **13** デーモンで **PostgreSQL 10** クライアントライブラリーを使用すると、期待どおりに動作します。

mariadb, mysql コンポーネント

MariaDB および MySQL は、ログファイルの作成時に **/opt/provider/collection/root** 接頭辞を利用しません。ログファイルは、**/opt/provider/collection/root/var/log/** ではなく、**/var/opt/provider/collection/log/** ディレクトリーに保存されることに注意してください。

1.6. その他の注意事項

rh-ruby*、rh-python*、rh-php* コンポーネント ()

読み取り専用 NFS で Software Collections を使用すると、いくつかの制限があります。

- **rh-ruby*** Software Collection が読み取り専用 NFS にある間は、Ruby gems をインストールできません。したがって、たとえば、**gem install ab** コマンドを使用して **ab gem** をインストールしようとする、以下のようなエラーメッセージが表示されます。

```
ERROR: While executing gem ... (Errno::EROFS)
  Read-only file system @ dir_s_mkdir - /opt/rh/rh-ruby22/root/usr/local/share/gems
```

または **bundle update** または **bundle install** コマンドを実行して、ユーザーが外部ソースから gem を更新またはインストールしようとする、同じ問題が発生します。

- Python Package Index (PyPI) を使用して読み取り専用 NFS に Python パッケージをインストールすると、この **pip** コマンドが失敗し、以下のようなエラーメッセージが表示されます。

```
Read-only file system: '/opt/rh/rh-python34/root/usr/lib/python3.4/site-packages/ipython-3.1.0.dist-info'
```

- **pear** コマンドを使用した読み取り専用 NFS への PEAR (PHP Extension and Application Repository) からパッケージのインストールに失敗し、エラーメッセージが表示されます。

```
Cannot install, php_dir for channel "pear.php.net" is not writeable by the current user
```

これは想定される動作です。

httpd コンポーネント

Apache の言語モジュールは、**Apache httpd** の Red Hat Software Collections バージョンでのみサポートされ、Red Hat Enterprise Linux のシステムバージョン **httpd** では対応していません。たとえば、rh-python35 コレクションの **mod_wsgi** モジュールは httpd24 コレクションでのみ使用できます。

すべてのコンポーネント

Red Hat Software Collections 2.0 以降、設定ファイル、変数データ、および各 Collections のランタイムデータは、以前のバージョンの Red Hat Software Collections とは異なるディレクトリーに保存されます。

coreutils、util-linux、screen コンポーネント

su、**login**、**screen** などの一部のユーティリティーは、すべてのケースで環境設定をエクスポートせず、予期せぬ結果になる可能性があります。そのため、**su** の代わりに **sudo** を使用し、**/etc/sudoers** ファイルに **env_keep** 環境変数を設定することを推奨します。下の順序でコマンドを実行できます。例を以下に示します。

```
su -l postgres -c "scl enable rh-postgresql94 psql"
```

以下の代わりとなります。

```
scl enable rh-postgresql94 bash
su -l postgres -c psql
```

screen、**login** などのツールを使用する場合は、以下のコマンドを使用して環境設定を保存できます。

```
source /opt/rh/<collection_name>/enable
```

python コンポーネント

ユーザーが **python27** および **rh-python*** Software Collections から複数の **scldevel** パッケージをインストールしようとする、トランザクションチェックのエラーメッセージが返されます。パッケージ (**%scl_python**, **%scl_prefix_python**) が提供するマクロファイルのセットをユーザーが1つだけインストールできるため、これは想定される動作です。

PHP コンポーネント)

ユーザーが **rh-php*** Software Collections から複数の **scldevel** パッケージをインストールしようすると、トランザクションチェックのエラーメッセージが返されます。パッケージ (**%scl_php**, **%scl_prefix_php**) が提供するマクロファイルのセットをユーザーが1つしかインストールできないため、これは想定される動作です。

ruby コンポーネント

ユーザーが **rh-ruby*** Software Collections から複数の **scldevel** パッケージをインストールしようすると、トランザクションチェックのエラーメッセージが返されます。パッケージ (**%scl_ruby**, **%scl_prefix_ruby**) が提供するマクロファイルのセットをユーザーが1つだけインストールできるため、これは想定される動作です。

perl コンポーネント

ユーザーが **rh-perl*** Software Collections から複数の **scldevel** パッケージをインストールしようすると、トランザクションチェックのエラーメッセージが返されます。パッケージ (**%scl_perl**, **%scl_prefix_perl**) が提供するマクロファイルのセットをユーザーが1つだけインストールできるた

め、これは想定される動作です。

nginx コンポーネント

ユーザーが `rh-nginx*` Software Collections から複数の `scl-devel` パッケージをインストールしようとすると、トランザクションチェックのエラーメッセージが返されます。パッケージ (`%scl_nginx`, `%scl_prefix_nginx`) が提供するマクロファイルのセットをユーザーが1つだけインストールできるため、これは想定される動作です。

python コンポーネント

Python `urllib` ライブラリーの [Web Cache Poisoning CVE-2021-23336](#) を緩和するため、`urllib.parse.parse_qs` および `urllib.parse.parse_qs` 関数のデフォルト区切り文字が、アンパサンド (&) とセミコロン (;) の両方が単一のアンパサンドに変更されます。

この変更は、python27 および rh-python38 Software Collections に、[RHSA -2021:3252](#) および [RHSA-2021: 3254](#) アドバイザリーのリリースで実装されています。

デフォルトの区切り文字の変更は後方互換性がない可能性があるため、Red Hat はデフォルトの区切り文字が変更された Python パッケージの動作を設定する手段を提供します。さらに、影響を受ける `urllib` 解析関数は、お客様のアプリケーションが変更の影響を受けることを検知すると警告を発行します。

詳細は、[Mitigation of Web Cache Poisoning in the Python urllib library \(CVE-2021-23336\)](#) ナレッジ記事を参照してください。

python コンポーネント

[RHSA-2021:3254](#) アドバイザリーのリリースでは、rh-python38 Software Collection で以下の変更が導入されています。

[CVE-2021-29921](#) を緩和するために、Python `ipaddress` モジュールが、**AddressValueError: Leading zeros** の先頭のゼロを持つ IPv4 アドレスを拒否するようになりました。

以前の動作に依存するお客様は、IPv4 アドレス入力を事前に処理して、先頭のゼロオフを削除します。以下に例を示します。

```
>>> def reformat_ip(address): return '.'.join(part.lstrip('0') if part != '0' else part for part in
address.split('.'))
>>> reformat_ip('0127.0.0.1')
'127.0.0.1'
```

読みやすのために明示的なループで先頭のゼロを取り除くには、以下を使用します。

```
def reformat_ip(address):
    parts = []
    for part in address.split('.'):
        if part != "0":
            part = part.lstrip('0')
            parts.append(part)
    return '.'.join(parts)
```

1.7. 非推奨の機能

httpd24 コンポーネント (BZ#1434053)

以前では、名前ベースの SSL 仮想ホスト選択が必要な SSL/TLS 設定で、**Host:** ヘッダーで提供されるホスト名が Server Name Indication (SNI) ヘッダーで提供されるホスト名と一致していなければ、**mod_ssl** モジュールは **400 Bad Request** エラーのあるリクエストを拒否していました。選択されたバーチャルホスト間で設定された SSL/TLS セキュリティーパラメーターが同じであれば、アップストリーム **mod_ssl** の動作に合わせて、そのようなリクエストは拒否されなくなりました。

第2章 インストール

この章では、コンテンツセットへのアクセス方法、システムへの Red Hat Software Collections 3.8 のインストール方法、および Red Hat Software Collections の再構築方法について詳しく説明します。

2.1. RED HAT SOFTWARE COLLECTIONS へのアクセス

Red Hat Software Collections コンテンツセットは、ナレッジベースの記事 [How to use Red Hat Software Collections \(RHSC\) or Red Hat Developer Toolset \(DTS\)?](#) に記載されている Red Hat Enterprise Linux サブスクリプションを利用することが可能です。Red Hat Subscription Management (RHSM) でシステムを登録する方法は、[Red Hat Subscription Manager の使用](#) を参照してください。RHSM を使用して Red Hat Software Collections を有効にする方法は、「[Red Hat Subscription Management の使用](#)」を参照してください。

Red Hat Software Collections 2.2 以降、Red Hat Software Collections および Red Hat Developer Toolset のコンテンツは、(特に [Server](#) および [Workstation](#) 向けに) <https://access.redhat.com/downloads> にて ISO 形式でも提供されています。「[Optional リポジトリからのパッケージ](#)」に記載されている **Optional** リポジトリを必要とするパッケージは、ISO イメージからインストールできないことに注意してください。



注記

Optional リポジトリを必要とするパッケージを ISO イメージからインストールすることはできません。**Optional** リポジトリの有効化が必要なパッケージの一覧は、「[Optional リポジトリからのパッケージ](#)」で提供されています。

ベータコンテンツは ISO 形式では使用できません。

2.1.1. Red Hat Subscription Management の使用

システムが Red Hat Subscription Management に登録されている場合は、以下の手順を実施して、Red Hat Software Collections のリポジトリへのアクセスを提供するサブスクリプションを割り当て、リポジトリを有効にします。

1. システムで利用可能なサブスクリプションのリストを表示し、Red Hat Software Collections を提供するサブスクリプションのプール ID を判別します。これを行うには、**root** で次のコマンドを実行します。

```
subscription-manager list --available
```

このコマンドは、使用可能なサブスクリプションごとに、その名前、一意の識別子、有効期限、およびそれに関連するその他の詳細を表示します。プール ID は、**Pool Id** で始まる行にリスト表示されます。

2. **root** で以下のコマンドを実行して、適切なサブスクリプションをシステムに割り当てます。

```
subscription-manager attach --pool=pool_id
```

pool_id を、直前のステップで確認したプール ID に置き換えます。システムに割り当てているサブスクリプションのリストを随時確認するには、**root** で以下を入力します。

```
subscription-manager list --consumed
```

- 利用可能な Yum list リポジトリのリストを表示して、リポジトリメタデータを取得し、Red Hat Software Collections リポジトリの正確な名前を決定します。**root** で以下のコマンドを実行します。

subscription-manager repos --list

または、**yum repolist all** を簡単なリストに対して実行します。

リポジトリ名は、使用している Red Hat Enterprise Linux のバージョンによって異なり、以下のフォーマットに基づいています。

```
rhel-variant-rhsc1-7-rpms
rhel-variant-rhsc1-7-debug-rpms
rhel-variant-rhsc1-7-source-rpms
```

variant を、Red Hat Enterprise Linux システムのバリエーション (つまり **server** または **workstation**) に置き換えます。Red Hat Software Collections は、**Client** または **ComputeNode** バリエーションではサポートされないことに注意してください。

- root** で以下のコマンドを実行して、適切なリポジトリを有効にします。

subscription-manager repos --enable repository

サブスクリプションがシステムに割り当てられたら、「[Red Hat Software Collections のインストール](#)」の説明に従って Red Hat Software Collections をインストールできます。Red Hat Subscription Management を使用してシステムを登録し、サブスクリプションに関連付ける方法は、[Red Hat Subscription Management の使用](#) を参照してください。



注記

RHN によるサブスクリプションは利用できなくなりました。

2.1.2. Optional リポジトリからのパッケージ

Red Hat Software Collections パッケージの一部では、このパッケージの完全なインストールを完了するために、**Optional** リポジトリを有効にする必要があります。このリポジトリにシステムをサブスクライブする方法の詳細は、ナレッジベースの記事 [How to access Optional and Supplementary channels, and -devel packages using Red Hat Subscription Management \(RHSM\)?](#) を参照してください。

Optional リポジトリを有効にする必要がある Red Hat Enterprise Linux の Software Collections パッケージを以下の表に示します。**Optional** リポジトリのパッケージはサポートされないことに注意してください。詳細は、ナレッジベース記事の [Support policy of the optional and supplementary channels in Red Hat Enterprise Linux](#) を参照してください。

表2.1 Red Hat Enterprise Linux 7 で Optional リポジトリを有効にする必要があるパッケージ

Software Collection からのパッケージ	Optional リポジトリから必要なパッケージ
devtoolset-12-build	scl-utils-build
devtoolset-12-dyninst-testsuite	glibc-static

Software Collection からのパッケージ	Optional リポジトリから必要なパッケージ
devtoolset-12-elfutils-debuginfod	bsdtar
devtoolset-12-gcc-plugin-devel	libmpc-devel
devtoolset-12-gdb	source-highlight
devtoolset-11-build	scl-utils-build
devtoolset-11-dyninst-testsuite	glibc-static
devtoolset-11-elfutils-debuginfod	bsdtar
devtoolset-11-gcc-plugin-devel	libmpc-devel
devtoolset-11-gdb	source-highlight
devtoolset-10-build	scl-utils-build
devtoolset-10-dyninst-testsuite	glibc-static
devtoolset-10-elfutils-debuginfod	bsdtar
devtoolset-10-gcc-plugin-devel	libmpc-devel
devtoolset-10-gdb	source-highlight
httpd24-mod_ldap	apr-util-ldap
rh-git227-git-cvs	cvsp
rh-git227-git-svn	perl-Git-SVN
rh-git227-perl-Git-SVN	subversion-perl
rh-php73-php-devel	pcre2-devel
rh-php73-php-pspell	aspell
rh-python38-python-devel	scl-utils-build

2.2. RED HAT SOFTWARE COLLECTIONS のインストール

Red Hat Software Collections は、Red Hat Enterprise Linux に含まれる標準のパッケージ管理ツールを使用して、インストール、更新、アンインストールが可能な RPM パッケージのコレクションとして配

布されます。Red Hat Software Collections をシステムにインストールするには、有効なサブスクリプションが必要です。システムを適切なサブスクリプションに関連付け、Red Hat Software Collections にアクセスする方法は、「[Red Hat Software Collections へのアクセス](#)」を参照してください。

Red Hat Software Collections 3.8 を使用するには、ベータリリースを含む以前のプレリリースバージョンを削除する必要があります。Red Hat Software Collections 3.8 の以前のバージョンをインストールしていた場合は、システムからアンインストールし、「[Red Hat Software Collections のアンインストール](#)」および「[Software Collection の個別インストール](#)」セクションで説明されているように、新しいバージョンをインストールしてください。

Red Hat Enterprise Linux 6 から Red Hat Enterprise Linux 7 へのインプレースアップグレードは、Red Hat Software Collections ではサポートされていません。したがって、アップグレード後にインストールされた Software Collections が正しく動作しない可能性があります。Red Hat Enterprise Linux 6 から Red Hat Enterprise Linux 7 にアップグレードする場合は、すべての Red Hat Software Collections パッケージを削除し、インプレースアップグレードを実行し、Red Hat Software Collections リポジトリを更新し、再度 Software Collections パッケージをインストールすることが強く推奨されます。アップグレードする前に、すべてのデータのバックアップを作成することが推奨されます。

2.2.1. Software Collection の個別インストール

表1.1「[Red Hat Software Collections のコンポーネント](#)」に記載されている Software Collection をインストールするには、シェルプロンプトで **root** として次のように入力して、対応するメタパッケージをインストールします。

```
yum install software_collection...
```

`software_collection` を、インストールする Software Collections のスペース区切りリストに置き換えます。たとえば、`rh-php73` および `rh-mariadb105` をインストールし、**root** として以下を入力します。

```
~]# yum install rh-php73 rh-mariadb105
```

これにより、選択した Software Collection のメインメタパッケージと、必要なパッケージの依存関係がインストールされます。追加モジュールなどの追加パッケージをインストールする方法は、「[オプションパッケージのインストール](#)」を参照してください。

2.2.2. オプションパッケージのインストール

Red Hat Software Collections の各コンポーネントは、デフォルトでインストールされていない複数のオプションパッケージとともに配布されます。特定の Software Collection に含まれていて、システムにインストールされていないパッケージのリストを表示するには、シェルプロンプトで次のコマンドを実行します。

```
yum list available software_collection-*
```

これらのオプションのパッケージのいずれかをインストールするには、**root** で以下を入力します。

```
yum install package_name...
```

`package_name` を、インストールするパッケージのリストに置き換えます。たとえば、`rh-perl530-perl-CPAN` および `rh-perl530-perl-Archive-Tar` をインストールするには、以下を入力します。

```
~]# yum install rh-perl530-perl-CPAN rh-perl530-perl-Archive-Tar
```

2.2.3. デバッグ情報のインストール

Red Hat Software Collections パッケージのデバッグ情報をインストールするには、`yum-utils` パッケージがインストールされていることを確認し、`root` で以下のコマンドを入力します。

```
debuginfo-install package_name
```

たとえば、`rh-ruby27-ruby` パッケージのデバッグ情報をインストールするには、以下を入力します。

```
~]# debuginfo-install rh-ruby27-ruby
```

これらのパッケージを含むリポジリーにアクセスできる必要があることに注意してください。システムが Red Hat Subscription Management に登録されている場合は、「[Red Hat Subscription Management の使用](#)」で説明しているように、`rhel-variant-rhsc1-6-debug-rpms` または `rhel-variant-rhsc1-7-debug-rpms` リポジリーを有効化します。`debuginfo` パッケージにアクセスする方法は、[RHEL システムで debuginfo パッケージをダウンロードまたはインストールする](#)を参照してください。

2.3. RED HAT SOFTWARE COLLECTIONS のアンインストール

Software Collections コンポーネントをアンインストールするには、`root` で次のコマンドを実行します。

```
yum remove software_collection*
```

`software_collection` を、アンインストールする Software Collection コンポーネントに置き換えます。

Red Hat Software Collections が提供するパッケージをアンインストールしても、これらのツールの Red Hat Enterprise Linux システムバージョンには影響がないことに注意してください。

2.4. RED HAT SOFTWARE COLLECTIONS の再構築

`<collection>-build` パッケージはデフォルトでは提供されません。コレクションを再構築して、`rpmbuild --define 'scl foo'` コマンドを使用しない場合には、最初にメタパッケージを再構築する必要があります。これにより、`<collection>-build` パッケージが提供されます。

既存のコレクションは、異なる内容で再構築しないでください。既存のコレクションに新しいパッケージを追加するには、新しいパッケージを含む新しいコレクションを作成し、元のコレクションからのパッケージに依存する必要があります。元のコレクションは、変更せずに使用する必要があります。

Software Collections のビルドに関する詳細は、[Red Hat Software Collections パッケージガイド](#)を参照してください。

第3章 使用方法

この章では、Red Hat Software Collections 3.8 の使用および Red Hat Software Collections を使用するアプリケーションのデプロイに必要な手順について説明します。

3.1. RED HAT SOFTWARE COLLECTIONS の使用

3.1.1. Software Collection からの実行可能ファイルの実行

特定の Software Collection から実行ファイルを実行するには、シェルプロンプトで以下のコマンドを入力します。

```
scl enable software_collection... 'command...'
```

または、以下のコマンドを使用します。

```
scl enable software_collection... -- command...
```

`software_collection` を、使用する Software Collections のスペース区切りのリストに置き換え、`command` を、実行するコマンドに置き換えます。たとえば、`perl526` Software Collection から Perl インタープリターで `hello.pl` という名前が付けられたファイルに保存されている Perl プログラムを実行するには、以下を入力します。

```
~]$ scl enable rh-perl526 'perl hello.pl'  
Hello, World!
```

この `scl` ユーティリティーを使用してコマンドを実行すると、同等の Red Hat Enterprise Linux システムの代わりに、選択した Software Collection から実行可能なものを使用して実行できます。Red Hat Software Collections で配布される Software Collections の完全リストは、[表1.1 「Red Hat Software Collections のコンポーネント」](#) を参照してください。

3.1.2. デフォルトで Software Collection を使用したシェルセッションの実行

Red Hat Enterprise Linux の同等のものよりも選択した Software Collection の実行可能ファイルで新しいシェルセッションを開始するには、シェルプロンプトで次のように入力します。

```
scl enable software_collection... bash
```

`software_collection` を、使用する Software Collections のスペース区切りリストに置き換えます。たとえば、デフォルトとして `python27` および `rh-postgresql12` Software Collections で新しいシェルセッションを開始するには、以下を入力します。

```
~]$ scl enable python27 rh-postgresql12 bash
```

現行セッションで有効になっている Software Collections のリストは、`$X_SCLS` 環境変数に保存されます。以下に例を示します。

```
~]$ echo $X_SCLS  
python27 rh-postgresql12
```

Red Hat Software Collections で配布される Software Collections の完全リストは、[表1.1 「Red Hat Software Collections のコンポーネント」](#) を参照してください。

3.1.3. Software Collection からのシステムサービスの実行

Red Hat Enterprise Linux 7 では、init スクリプトが **systemd** サービスのユニットファイルに置き換えられました。このファイルは、**.service** ファイル拡張子で終わり、init スクリプトとして同様の目的で使用されています。現行セッションでサービスを起動するには、**root** で以下のコマンドを実行します。

```
systemctl start software_collection-service_name.service
```

`software_collection` を、Software Collection および `service_name` を、開始するサービスの名前に置き換えます。

システムの起動時にこのサービスが自動的に開始するように設定するには、**root** で以下のコマンドを入力します。

```
systemctl enable software_collection-service_name.service
```

たとえば、rh-postgresql12 Software Collection から **postgresql** サービスを起動し、システムの起動時に有効にするには、**root** で以下を入力します。

```
~]# systemctl start rh-postgresql12-postgresql.service  
~]# systemctl enable rh-postgresql12-postgresql.service
```

Red Hat Enterprise Linux 7 でシステムサービスを管理する方法は、[Red Hat Enterprise Linux 7 System Administrator's Guide](#) を参照してください。Red Hat Software Collections で配布される Software Collections の完全リストは、[表1.1 「Red Hat Software Collections のコンポーネント」](#) を参照してください。

3.2. SOFTWARE COLLECTION からの手動ページへのアクセス

すべての Software Collection には、このコンポーネントの内容を説明する一般的な man ページが含まれています。各 man ページにはコンポーネントと同じ名前が付いており、**/opt/rh** ディレクトリーにあります。

Software Collection の man ページを確認するには、以下のコマンドを入力します。

```
scl enable software_collection 'man software_collection'
```

`software_collection` を、特定の Red Hat Software Collections コンポーネントに置き換えます。たとえば、rh-mariadb105 の man ページを表示するには、以下を入力します。

```
~]$ scl enable rh-mariadb105 "man rh-mariadb105"
```

3.3. RED HAT SOFTWARE COLLECTIONS を使用するアプリケーションのデプロイ

通常、以下の2つの方法のいずれかを使用して、実稼働環境の Red Hat Software Collections のコンポーネントに依存するアプリケーションをデプロイすることができます。

- 必要な Software Collections およびパッケージをすべて手動でインストールしてから、アプリケーションをデプロイする、または
- アプリケーション用の新しい Software Collection を作成し、必要な Software Collections およびその他のパッケージをすべて依存関係として指定する

個々の Red Hat Software Collections コンポーネントを手動でインストールする方法は、「[Red Hat Software Collections のインストール](#)」を参照してください。Red Hat Software Collections の使用方法に関する詳細は、「[Red Hat Software Collections の使用](#)」を参照してください。カスタム Software Collection を作成する方法や、既存のものを拡張する方法については、[Red Hat Software Collections パッケージガイド](#)を参照してください。

3.4. RED HAT SOFTWARE COLLECTIONS コンテナイメージ

Red Hat Software Collections に基づくコンテナイメージには、アプリケーション、デーモン、およびデータベースが含まれます。イメージは、Red Hat Enterprise Linux 7 Server および Red Hat Enterprise Linux Atomic Host で実行できます。使用方法は、[Using Red Hat Software Collections 3 Container Images](#)を参照してください。Red Hat Software Collections バージョン 2.4 以前の Red Hat Software Collections バージョン 2.4 をベースとしたコンテナイメージの詳細は、[Red Hat Software Collections 2 コンテナイメージ](#)を参照してください。

各コンテナイメージの最新バージョンのみがサポートされることに注意してください。

以下のコンテナイメージは、Red Hat Software Collections 3.8 で利用できます。

- `rhsc/devtoolset-12-toolchain-rhel7` (2022 年 11 月以降に利用可能)
- `rhsc/devtoolset-12-perftools-rhel7` (2022 年 11 月以降に利用可能)
- `rhsc/nginx-120-rhel7`
- `rhsc/redis-6-rhel7`

以下のコンテナイメージは Red Hat Software Collections 3.7: をベースとしています。

- `rhsc/mariadb-105-rhel7`
- `rhsc/postgresql-13-rhel7`
- `rhsc/ruby-30-rhel7`

以下のコンテナイメージは Red Hat Software Collections 3.6: をベースとしています。

- `rhsc/httpd-24-rhel7`
- `rhsc/nodej-14-rhel7`
- `rhsc/perl-530-rhel7`
- `rhsc/php-73-rhel7`

以下のコンテナイメージは Red Hat Software Collections 3.5 をベースとしています。

- `rhsc/python-38-rhel7`
- `rhsc/varnish-6-rhel7`

以下のコンテナイメージは、Red Hat Software Collections 3.4 をベースとしています。

- `rhsc/postgresql-12-rhel7`

以下のコンテナイメージは、Red Hat Software Collections 3.2 をベースとしています。

- `rhsc/mysql-80-rhel7`

以下のコンテナイメージは、Red Hat Software Collections 3.1 をベースとしています。

- `rhsc/postgresql-10-rhel7`

以下のコンテナイメージは、Red Hat Software Collections 3.2 をベースとしています。

- `rhsc/s2i-base-rhel7`

第4章 個別の SOFTWARE COLLECTIONS の詳細

本章では、特定の Software Collections の詳細に重点を置き、これらのコンポーネントに関する追加情報を提供します。

4.1. RED HAT DEVELOPER TOOLSET

Red Hat Developer Toolset は、Red Hat Enterprise Linux プラットフォームで作業する開発者向けに設計されています。Red Hat Developer Toolset は、現在のバージョンの **GNU Compiler Collection**、**GNU Debugger**、およびその他の開発、デバッグ、パフォーマンス監視ツールを提供します。他の Software Collections と同様に、追加のツールセットが `/opt/` ディレクトリーにインストールされます。これらのツールは、提供された `scl` ユーティリティーを使用してオンデマンドでユーザーが有効にします。他の Software Collections と同様に、これらのツールの Red Hat Enterprise Linux システムバージョンを置き換えることはありません。また、`scl` ユーティリティーを使用して明示的に呼び出されない限り、これらのシステムバージョンを優先して使用することもできます。

機能の概要については、『Red Hat Developer Toolset リリースノート』の [Features](#) セクションを参照してください。12.1における使用方法および変更に関する詳細は、[Red Hat Developer Toolset ユーザーガイド](#) を参照してください。

4.2. MAVEN

`rh-maven36` Software Collection は、Red Hat Enterprise Linux 7 でのみ利用可能です。ソフトウェアプロジェクトの管理および解釈を行うツールを提供します。**Maven** はプロジェクトオブジェクトモデル (POM) の概念に基づいて、プロジェクトのビルド、レポート、およびドキュメントを一元的な情報から管理できます。

`rh-maven36` Collection をインストールするには、**root** で次のコマンドを実行します。

```
yum install rh-maven36
```

このコレクションを有効にするには、シェルプロンプトで以下のコマンドを入力します。

```
scl enable rh-maven36 bash
```

リモートリポジトリやミラーなどのグローバル Maven 設定は、`/opt/rh/rh-maven36/root/etc/maven/settings.xml` ファイルを編集してカスタマイズできます。

Maven の使用に関する詳細は、[Maven ドキュメント](#) を参照してください。プラグインの用法については、[こちらのセクション](#) で説明します。個々のプラグインに関するドキュメントをお探しの場合は、[プラグインのインデックス](#) を参照してください。

4.3. データベースコネクタ

データベースコネクタパッケージは、データベースサーバーへのローカルまたはリモート接続に必要なデータベースクライアント機能を提供します。[表4.1「言語とデータベース間の相互運用性」](#) 特定のデータベースサーバーのコネクタを含む言語ランタイムを含む Software Collections を一覧表示します。()

- **はい** - この組み合わせはサポートされています
- **いいえ** - この組み合わせはサポートされていません

表4.1 言語とデータベース間の相互運用性

言語 (Software Collection)	Database					
	MariaDB	MongoDB	MySQL	PostgreSQL	Redis	SQLite3
rh-nodejs4	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
rh-nodejs6	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
rh-nodejs8	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
rh-nodejs10	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
rh-nodejs12	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
rh-nodejs14	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
rh-perl520	はい	いいえ	はい	はい	いいえ	いいえ
rh-perl524	はい	いいえ	はい	はい	いいえ	いいえ
rh-perl526	はい	いいえ	はい	はい	いいえ	いいえ
rh-perl530	はい	いいえ	はい	はい	いいえ	はい
rh-php56	はい	はい	はい	はい	いいえ	はい
rh-php70	はい	いいえ	はい	はい	いいえ	はい
rh-php71	はい	いいえ	はい	はい	いいえ	はい
rh-php72	はい	いいえ	はい	はい	いいえ	はい
rh-php73	はい	いいえ	はい	はい	いいえ	はい
python27	はい	はい	はい	はい	いいえ	はい
rh-python34	いいえ	はい	いいえ	はい	いいえ	はい
rh-python35	はい	はい	はい	はい	いいえ	はい
rh-python36	はい	はい	はい	はい	いいえ	はい
rh-python38	はい	いいえ	はい	はい	いいえ	はい
rh-ror41	はい	はい	はい	はい	いいえ	はい

言語 (Software Collection)	Database					
	MariaDB	MongoDB	MySQL	PostgreSQL	Redis	SQLite3
rh-ror42	はい	はい	はい	はい	いいえ	はい
rh-ror50	はい	はい	はい	はい	いいえ	はい
rh-ruby25	はい	はい	はい	はい	いいえ	いいえ
rh-ruby26	はい	はい	はい	はい	いいえ	いいえ
rh-ruby27	はい	はい	はい	はい	いいえ	いいえ
rh-ruby30	はい	いいえ	はい	はい	いいえ	はい

第5章 移行

この章では、Red Hat Software Collections 3.8 に含まれるコンポーネントのバージョンへの移行に関する情報を提供します。

5.1. MARIADB 10.5 への移行

rh-mariadb105 Software Collection は、デフォルトの MySQL 実装として **MariaDB 5.5** を含む Red Hat Enterprise Linux 7 で利用できます。

rh-mariadb105 Software Collection は、コアシステムの mysql または mariadb パッケージと競合しません。*-syspaths パッケージがインストールされていない限り (下記参照)、mysql パッケージまたは mariadb パッケージとともに rh-mariadb105 Software Collection をインストールできます。特定のリソースが競合しないようにするため、両方のバージョンを同時に実行することもできますが、ポート番号と **my.cnf** ファイルのソケットを変更する必要があります。さらに、rh-mariadb103 Collection がインストールされ、実行中でも、rh-mariadb105 Software Collection をインストールすることもできます。

rh-mariadb105 Software Collection には、バイナリー、スクリプト、man ページなどのシステム全体のラッパーを提供するパッケージをインストールする rh-mariadb105-syspaths パッケージが含まれます。rh-mariadb105*-syspaths パッケージのインストール後に、rh-mariadb105* パッケージによって提供されるバイナリーおよびスクリプトが正しく動作するかを **scl enable** コマンドを使用して確認する必要はありません。*-syspaths パッケージは、ベースの Red Hat Enterprise Linux システムと、rh-mariadb103 および rh-mysql80 Software Collections からの対応するパッケージと競合することに注意してください。syspaths の詳細は、[Red Hat Software Collections パッケージガイド](#) を参照してください。

MariaDB 5.5 から **MariaDB 10.5** への推奨される移行パスは、最初に **MariaDB 10.0** にアップグレードしてから、1バージョンずつ順次アップグレードすることです。詳細は、以前の Red Hat Software Collections リリースノートの手順 ([Migrating to MariaDB 10.0](#)、[Migrating to MariaDB 10.1](#)、[Migrating to MariaDB 10.2](#)、および [Migrating to MariaDB 10.3](#)) を参照してください。**MariaDB 10.4** は Software Collection として利用できないため、rh-mariadb103 から rh-mariadb105 に直接移行する必要があります。



注記

rh-mariadb105 Software Collection では、NFS によるマウントや **scl register** コマンドによる動的登録はサポートしていません。

5.1.1. rh-mariadb103 および rh-mariadb105 Software Collections 間の主な相違点

MariaDB 10.3 と **MariaDB 10.5** の間の重要な変更点には以下が含まれます。

- **MariaDB** はデフォルトで **unix_socket** 認証プラグインを使用するようになりました。このプラグインを使用すると、ローカルの UNIX ソケットファイルを介して **MariaDB** に接続する際に、オペレーティングシステムの資格情報を使用できます。
- **MariaDB** は、**mariadb-*** という名前のバイナリーと、**mariadb-*** バイナリーを指す **mysql*** シンボリックリンクを追加します。たとえば、**mysqladmin**、**mysqlaccess**、および **mysqlshow** のシンボリックリンクは、**mariadb-admin**、**mariadb-access**、および **mariadb-show** のバイナリーをそれぞれポイントします。
- 各ユーザーロールに合わせて、**SUPER** 特権が複数の特権に分割されました。その結果、一部のステートメントが必要な特権が変更されました。

- 並列レプリケーションでは、**slave_parallel_mode** が **optimistic** にデフォルト設定されるようになりました。
- InnoDB ストレージエンジンでは、以下の変数のデフォルトが変更されました。**innodb_adaptive_hash_index** は **OFF**、**innodb_checksum_algorithm** は **full_crc32**。
- MariaDB は、以前使用された **readline** ライブラリーではなく、MariaDB コマンド履歴 (**.mysql_history** ファイル) を管理する基盤となるソフトウェアの **libedit** 実装を使用するようになりました。この変更は、**.mysql_history** ファイルを直接使用しているユーザーに影響します。**.mysql_history** は MariaDB または MySQL アプリケーションによって管理されるファイルであるため、ユーザーは直接ファイルでは機能しないことに注意してください。人間が判読可能な外観は偶然です。



注記

セキュリティを強化するために、履歴ファイルの維持を考慮することができます。コマンド履歴の記録を無効にするには、以下を実行します。

1. 存在する場合は、**.mysql_history** ファイルを削除します。
2. 以下のいずれかの方法を使用します。
 - **MYSQL_HISTFILE** 変数を **/dev/null** に設定し、これをシェルの起動ファイルに追加します。
 - **.mysql_history** ファイルを **/dev/null** へのシンボリックリンクに変更します。

```
In -s /dev/null $HOME/.mysql_history
```

MariaDB Galera クラスタがバージョン 4 にアップグレードされ、以下の主な変更点が加えられました。

- Galera は、サイズ制限なしのトランザクションの複製をサポートする、新しいストリーミングレプリケーション機能を追加します。ストリーミングレプリケーションの実行時に、クラスタは小さなフラグメントでトランザクションを複製します。
- Galera がグローバルトランザクション ID (GTID) に完全に対応するようになりました。
- **/etc/my.cnf.d/galera.cnf** ファイルの **wsrep_on** オプションのデフォルト値が **1** から **0** に変更され、エンドユーザーが必要な追加オプションを設定せずに **wsrep** レプリケーションを開始できないようにします。

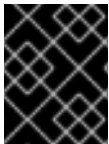
MariaDB 10.5 の PAM プラグインへの変更には、以下が含まれます。

- MariaDB 10.5 は、PAM (Pluggable Authentication Modules) プラグインの新バージョンを追加します。PAM プラグインバージョン 2.0 は、個別の **setuid root** ヘルパーバイナリーを使用して PAM 認証を実行します。これにより MariaDB が追加の PAM モジュールを使用できるようになります。
- ヘルパーバイナリーは、**mysql** グループのユーザーによってのみ実行できます。デフォルトでは、グループには **mysql** ユーザーのみが含まれます。Red Hat では、このヘルパーユーティリティを介してスロットルまたはログの記録をせずにパスワード推測攻撃を防ぐために、管理者が **mysql** グループにユーザーをさらに追加しないことを推奨しています。

- **MariaDB 10.5** では、PAM (Pluggable Authentication Modules) プラグインとその関連ファイルが新しいサブパッケージ `mariadb-pam` に移動しました。したがって、**MariaDB** に PAM 認証を使用しないシステムに、新しい **setuid root** バイナリーが導入されることはありません。
- `rh-mariadb105-mariadb-pam` パッケージには PAM プラグインバージョンの両方が含まれています。バージョン 2.0 はデフォルトで、バージョン 1.0 は **auth_pam_v1** 共有オブジェクトライブラリーとして利用できます。
- **MariaDB** サーバーでは、デフォルトで `rh-mariadb105-mariadb-pam` パッケージがインストールされないことに注意してください。**MariaDB 10.5** で PAM 認証プラグインを利用できるようにするには、`rh-mariadb105-mariadb-pam` パッケージを手動でインストールします。

詳細は、[MariaDB 10.4 の変更](#) および [MariaDB 10.5 の変更](#) に関するアップストリームドキュメントを参照してください。[MariaDB 10.4 へのアップグレード](#) および [MariaDB 10.5 へのアップグレード](#) に関するアップストリーム情報も参照してください。

5.1.2. rh-mariadb103 から rh-mariadb105 Software Collection へのアップグレード



重要

アップグレードする前に、MariaDB データベースを含むすべてのデータのバックアップを作成します。

1. `rh-mariadb103` データベースサーバーが実行している場合は停止します。

サーバーを停止する前に、**innodb_fast_shutdown** オプションを **0** に設定し、**InnoDB** が完全なページや挿入バッファーマージを含む低速なシャットダウンを実行します。[アップストリームのドキュメント](#) で、このオプションの詳細を参照してください。この操作は、通常のシャットダウンの場合よりも長い時間がかかる可能性があります。

```
mysql -uroot -p -e "SET GLOBAL innodb_fast_shutdown = 0"
```

`rh-mariadb103` サーバーを停止します。

```
systemctl stop rh-mariadb103-mariadb.service
```

2. **mysql_upgrade** ユーティリティーを提供するサブパッケージを含む `rh-mariadb105` Software Collection をインストールします。

```
yum install rh-mariadb105-mariadb-server rh-mariadb105-mariadb-server-utils
```

これらの Collections が競合しないため、`rh-mariadb105` Software Collection のインストール中は `rh-mariadb103` Software Collection をインストールすることができることに注意してください。

3. `/etc/opt/rh/rh-mariadb105/my.cnf` ファイルおよび `/etc/opt/rh/rh-mariadb105/my.cnf.d/` ディレクトリーに保存されている `rh-mariadb105` の設定を検証します。これを `/etc/opt/rh/rh-mariadb103/my.cnf` および `/etc/opt/rh/rh-mariadb103/my.cnf.d/` に保存されている `rh-mariadb103` の設定と比較して、必要に応じて調整します。
4. `rh-mariadb103` Software Collection のすべてのデータは、異なる設定しない限り、`/var/opt/rh/rh-mariadb103/lib/mysql/` ディレクトリーに保存されます。このディレクトリーのすべての内容を `/var/opt/rh/rh-mariadb105/lib/mysql/` にコピーします。コンテンツを移

動することはできますが、アップグレードを続行する前にデータをバックアップすることを忘れないようにしてください。データが **mysql** ユーザーによって所有され、SELinux コンテキストが正しいことを確認します。

5. rh-mariadb105 データベースサーバーを起動します。

```
systemctl start rh-mariadb105-mariadb.service
```

6. データ移行を実行します。MDEV-14637 で導入されたアップストリームの変更により、**mysql_upgrade** コマンドを実行する必要があることに注意してください。

```
scl enable rh-mariadb105 mysql_upgrade
```

root ユーザーに空ではないパスワードが定義されている場合 (パスワードを定義しておく必要あり) は、**-p** オプションを指定して **mysql_upgrade** ユーティリティを呼び出してパスワードを指定する必要があります。

```
scl enable rh-mariadb105 -- mysql_upgrade -p
```

rh-mariadb105*-syspaths パッケージのインストール時には、**scl enable** コマンドは必要ありません。ただし、*-syspaths パッケージは、ベースの Red Hat Enterprise Linux システムと、rh-mariadb103 および rh-mysql80 Software Collections からの対応するパッケージと競合します。

5.2. MYSQL 8.0 への移行

rh-mysql80 Software Collection は、デフォルトの **MySQL** 実装として **MariaDB 5.5** を含む Red Hat Enterprise Linux 7 で利用できます。

rh-mysql80 Software Collection は、*-syspaths パッケージがインストールされていない限り、コアシステムの **mysql** パッケージまたは **mariadb** パッケージ、rh-mysql* または rh-mariadb* Software Collections と競合しません (以下を参照)。また、複数のバージョンを同時に実行することも可能ですが、特定リソースが競合しないようにするため、ポート番号と **my.cnf** ファイルのソケットを変更する必要があります。

MySQL 5.7 からのみ **MySQL 8.0** にアップグレードできることに注意してください。以前のバージョンからアップグレードする必要がある場合は、最初に **MySQL 5.7** にアップグレードしてください。手順は、[Migration to MySQL 5.7](#) を参照してください。

5.2.1. MySQL 5.7 と MySQL 8.0 間の主な相違点

rh-mysql80 ソフトウェアの収集に固有の相違点

- Red Hat Enterprise Linux 7 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、rh-mysql80 Software Collection が提供する **MySQL 8.0** サーバーは、**mysql_native_password** をデフォルトの認証プラグインとして使用します。

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/opt/rh/rh-mysql80/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
```

cached_sha2_password 認証プラグインの詳細は、[アップストリームのドキュメント](#) を参照してください。

- rh-mysql80 Software Collection には、rh-mysql80-mysql-config-syspaths パッケージ、rh-mysql80-mysql-server-syspaths パッケージ、および rh-mysql80-mysql-syspaths パッケージをインストールする rh-mysql80-syspaths パッケージが含まれます。これらのサブパッケージは、バイナリー、スクリプト、man ページなどのシステム全体のラッパーを提供します。rh-mysql80*-syspaths パッケージのインストール後に、rh-mysql80* パッケージによって提供されるバイナリーおよびスクリプトが正しく動作するかを **scl enable** コマンドを使用して確認する必要はありません。*-syspaths パッケージは、ベースの Red Hat Enterprise Linux システムと、rh-mariadb103 および rh-mariadb105 Software Collections からの対応するパッケージと競合することに注意してください。syspaths の詳細は、[Red Hat Software Collections Packaging Guide](#) を参照してください。

MySQL 8.0 の一般的な変更点

- バイナリーロギングは、サーバーの起動時にデフォルトで有効になります。**--log-bin** オプションが指定されていない場合でも、**log_bin** システム変数はデフォルトで **ON** に設定されます。バイナリーロギングを無効にするには、起動時に **--skip-log-bin** または **--disable-log-bin** オプションを指定します。
- **CREATE FUNCTION** ステートメントを許可するには、**DETERMINISTIC**、**NO SQL**、または **READS SQL DATA** のキーワードの中から少なくとも1つを明示的に指定する必要があります。指定しない場合は、エラーが発生します。
- アカウント管理に関連する特定の機能が削除されました。つまり、**GRANT** ステートメントを使用して、認証、SSL、リソース制限などの特権割り当て以外のアカウントプロパティーを変更できなくなりました。アカウント作成時に上記のプロパティーを確立するには、**CREATE USER** ステートメントを使用します。これらのプロパティーを変更するには、**ALTER USER** ステートメントを使用します。
- クライアント側で一部の SSL 関連のオプションが削除されました。**--ssl=1** または **--enable-ssl** の代わりに **--ssl-mode=REQUIRED** オプションを使用します。**--ssl=0**、**--skip-ssl**、または **--disable-ssl** の代わりに **--ssl-mode=DISABLED** オプションを使用してください。**--ssl-verify-server-cert** オプションの代わりに **--ssl-mode=VERIFY_IDENTITY** オプションを使用してください。このオプションは、サーバー側では変更されないままになることに注意してください。
- デフォルトの文字セットが、**latin1** から **utf8mb4** に変更しました。
- **utf8** 文字セットは現時点で **utf8mb3** のエイリアスになりますが、今後は **utf8mb4** への参照となります。曖昧さを防ぐには、**utf8** ではなく文字セット参照に **utf8mb4** を明示的に指定します。
- **SET** 以外のステートメントでユーザー変数を設定することが非推奨になりました。
- 以前エラーログをシステムログに記録した **log_syslog** 変数は削除されました。
- 空間データのサポートに対する互換性のない変更がいくつか導入されています。
- **GROUP BY** 句の非推奨の **ASC** または **DESC** 修飾子が削除されました。特定のソート順序を生成するには、**ORDER BY** 句を指定します。

以前のバージョンと比較した MySQL 8.0 における詳細な変更については、アップストリームのドキュメント [What Is New in MySQL 8.0](#) および [Changes Affecting Upgrades to MySQL 8.0](#) を参照してください。

5.2.2. rh-mysql80 Software Collection へのアップグレード



重要

アップグレードする前に、MySQL データベースを含むすべてのデータのバックアップを作成します。

1. rh-mysql80 Software Collection をインストールします。

```
yum install rh-mysql80-mysql-server
```

2. `/etc/opt/rh/rh-mysql80/my.cnf` ファイルおよび `/etc/opt/rh/rh-mysql80/my.cnf.d/` ディレクトリーに保存されている rh-mysql80 の設定を検証します。これを `/etc/opt/rh/rh-mysql57/my.cnf` と `/etc/opt/rh/rh-mysql57/my.cnf.d/` に保存されている rh-mysql57 の設定と比較して、必要に応じて調整します。
3. rh-mysql57 データベースサーバーが実行している場合は停止します。

```
systemctl stop rh-mysql57-mysqld.service
```

4. rh-mysql57 Software Collection のすべてのデータは `/var/opt/rh/rh-mysql57/lib/mysql/` ディレクトリーに保存されます。このディレクトリーのすべての内容を `/var/opt/rh/rh-mysql80/lib/mysql/` にコピーします。コンテンツを移動することもできますが、アップグレードを続行する前にデータのバックアップを作成することを忘れないようにしてください。
5. rh-mysql80 データベースサーバーを起動します。

```
systemctl start rh-mysql80-mysqld.service
```

6. データ移行を実行します。

```
scl enable rh-mysql80 mysql_upgrade
```

root ユーザーに空ではないパスワードが定義されている場合 (パスワードを定義しておく必要あり) は、**-p** オプションを指定して `mysql_upgrade` ユーティリティーを呼び出してパスワードを指定する必要があります。

```
scl enable rh-mysql80 -- mysql_upgrade -p
```

rh-mysql80*-syspaths パッケージのインストール時には、**scl enable** コマンドは必要ありません。ただし、*-syspaths パッケージは、ベースの Red Hat Enterprise Linux システムと、rh-mariadb103 および rh-mariadb105 Software Collections からの対応するパッケージと競合しません。

5.3. POSTGRESQL 13 への移行

Red Hat Software Collections 3.8 には **PostgreSQL 13** が同梱されており、Red Hat Enterprise Linux 7 でのみ利用できます。rh-postgresql13 Software Collection は、**PostgreSQL** または **PostgreSQL** Software Collection のベースの Red Hat Enterprise Linux システムバージョンと並行して、同じマシンに安全にインストールできます。複数のバージョンの **PostgreSQL** を同時にマシン上で実行することもできますが、別のポートまたは IP アドレスを使用し、SELinux ポリシーを調整する必要があります。

rh-postgresql13 Software Collection には、バイナリー、スクリプト、man ページなどのシステム全体のラッパーを提供するパッケージをインストールする rh-postgresql13-syspaths パッケージが含まれます。rh-postgresql13*-syspaths パッケージのインストール後に、rh-postgresql13* パッケージによって提供されるバイナリーおよびスクリプトが正しく動作するかを **scl enable** コマンドを使用して確認する必要はありません。*-syspaths パッケージは、ベースの Red Hat Enterprise Linux システムと対応するパッケージと競合することに注意してください。syspaths の詳細は、[Red Hat Software Collections パッケージガイド](#)を参照してください。



重要

PostgreSQL 13 に移行する前に、アップストリームによる [PostgreSQL 13](#) の互換性に関する注意事項を参照してください。

コンテナで PostgreSQL データベースをアップグレードする場合は、[コンテナ固有の手順](#)を参照してください。

以下の表は、postgresql パッケージが提供する PostgreSQL の Red Hat Enterprise Linux 7 システムバージョン、ならびに rh-postgresql12 および rh-postgresql13 Software Collection の異なるパスの概要を示しています。

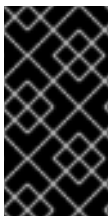
表5.1 PostgreSQL パスの相違点

コンテンツ	postgresql	rh-postgresql12	rh-postgresql13
実行ファイル	/usr/bin/	/opt/rh/rh-postgresql12/root/usr/bin/	/opt/rh/rh-postgresql13/root/usr/bin/
ライブラリー	/usr/lib64/	/opt/rh/rh-postgresql12/root/usr/lib64/	/opt/rh/rh-postgresql13/root/usr/lib64/
資料	/usr/share/doc/postgresql/html/	/opt/rh/rh-postgresql12/root/usr/share/doc/postgresql/html/	/opt/rh/rh-postgresql13/root/usr/share/doc/postgresql/html/
PDF ドキュメント	/usr/share/doc/postgresql-docs/	/opt/rh/rh-postgresql12/root/usr/share/doc/postgresql-docs/	/opt/rh/rh-postgresql13/root/usr/share/doc/postgresql-docs/
Contrib ドキュメント	/usr/share/doc/postgresql-contrib/	/opt/rh/rh-postgresql12/root/usr/share/doc/postgresql-contrib/	/opt/rh/rh-postgresql13/root/usr/share/doc/postgresql-contrib/
ソース	未インストール	未インストール	未インストール
データ	/var/lib/pgsql/data/	/var/opt/rh/rh-postgresql12/lib/pgsql/data/	/var/opt/rh/rh-postgresql13/lib/pgsql/data/

コンテンツ	postgresql	rh-postgresql12	rh-postgresql13
バックアップエリア	/var/lib/pgsql/backups/	/var/opt/rh/rh-postgresql12/lib/pgsql/backups/	/var/opt/rh/rh-postgresql13/lib/pgsql/backups/
テンプレート	/usr/share/pgsql/	/opt/rh/rh-postgresql12/root/usr/share/pgsql/	/opt/rh/rh-postgresql13/root/usr/share/pgsql/
手順言語	/usr/lib64/pgsql/	/opt/rh/rh-postgresql12/root/usr/lib64/pgsql/	/opt/rh/rh-postgresql13/root/usr/lib64/pgsql/
開発ヘッダー	/usr/include/pgsql/	/opt/rh/rh-postgresql12/root/usr/include/pgsql/	/opt/rh/rh-postgresql13/root/usr/include/pgsql/
他の共有データ	/usr/share/pgsql/	/opt/rh/rh-postgresql12/root/usr/share/pgsql/	/opt/rh/rh-postgresql13/root/usr/share/pgsql/
リグレッションテスト	/usr/lib64/pgsql/test/regress/ (-test パッケージ内)	/opt/rh/rh-postgresql12/root/usr/lib64/pgsql/test/regress/ (-test パッケージ内)	/opt/rh/rh-postgresql13/root/usr/lib64/pgsql/test/regress/ (-test パッケージ内)

5.3.1. Red Hat Enterprise Linux システムバージョンの PostgreSQL から PostgreSQL 13 Software Collection への移行

Red Hat Enterprise Linux 7 には PostgreSQL 9.2 が同梱されています。PostgreSQL の Red Hat Enterprise Linux システムバージョンから rh-postgresql13 Software Collection にデータを移行するには、**pg_upgrade** ツール (推奨) を使用して高速アップグレードを実行するか、SQL コマンドでデータベースデータをテキストファイルにダンプし、新しいデータベースにインポートします。2 つ目の方法は、通常大幅に遅くなるため、手動で修正が必要になる場合があります。このアップグレード方法の詳細は、[PostgreSQL ドキュメント](#) を参照してください。



重要

Red Hat Enterprise Linux システムバージョンの PostgreSQL から PostgreSQL 13 にデータを移行する前に、デフォルトで **/var/lib/pgsql/data/** ディレクトリーに格納される PostgreSQL データベースファイルを含む、すべてのデータをバックアップしてください。

手順5.1 pg_upgrade ツールを使用した高速アップグレード

PostgreSQL サーバーの高速アップグレードを実行するには、以下の手順を実行します。

1. 古い PostgreSQL サーバーを停止し、データが一貫性のない状態にあることを確認します。これを行うには、**root** で次のコマンドを実行します。

```
systemctl stop postgresql.service
```

サーバーが起動していないことを確認するには、以下を入力します。

```
systemctl status postgresql.service
```

- 古いディレクトリー `/var/lib/pgsql/data/` が存在することを確認します。

```
file /var/lib/pgsql/data/
```

データのバックアップを作成します。

- 新しいデータディレクトリー `/var/opt/rh/rh-postgresql13/lib/pgsql/data/` が存在しないことを確認します。

```
file /var/opt/rh/rh-postgresql13/lib/pgsql/data/
```

PostgreSQL 13 の新規インストールを実行している場合は、このディレクトリーはシステムに存在しないはずですが、その場合は、**root** で以下のコマンドを実行してバックアップを作成します。

```
mv /var/opt/rh/rh-postgresql13/lib/pgsql/data{,-scl-backup}
```

- root** で以下のコマンドを実行して、新しいサーバーのデータベースデータをアップグレードします。

```
scl enable rh-postgresql13 -- postgresql-setup --upgrade
```

または、`/opt/rh/rh-postgresql13/root/usr/bin/postgresql-setup --upgrade` コマンドを使用できます。

別のバージョンの PostgreSQL からアップグレードする場合には、`--upgrade-from` オプションを使用できます。可能なアップグレードシナリオのリストは、`--upgrade-ids` オプションを使用して利用できます。

作成された `/var/lib/pgsql/upgrade_rh-postgresql13-postgresql.log` ログファイルを読み取り、アップグレード中に問題が発生したかどうかを確認することが推奨されます。

- root** で新しいサーバーを起動します。

```
systemctl start rh-postgresql13-postgresql.service
```

また、以下のように `analyze_new_cluster.sh` スクリプトを実行することが推奨されます。

```
su - postgres -c 'scl enable rh-postgresql13 ~/analyze_new_cluster.sh'
```

- 必要に応じて、システムの起動時に PostgreSQL 13 サーバーが自動的に起動するように設定できます。古いシステム PostgreSQL サーバーを無効にするには、**root** で以下のコマンドを入力します。

```
chkconfig postgresql off
```

PostgreSQL 13 サーバーを有効にするには、**root** で以下を入力します。

```
chkconfig rh-postgresql13-postgresql on
```

7. デフォルトの設定と異なる場合は、設定ファイル (特に `/var/opt/rh/rh-postgresql13/lib/pgsql/data/pg_hba.conf` 設定ファイル) を必ず更新してください。それ以外の場合は、**postgres** ユーザーのみがデータベースにアクセスできます。

手順5.2 ダンプおよびリストアアップグレードの実行

PostgreSQL サーバーのダンプおよび復元アップグレードを実行するには、以下の手順を実行します。

1. シェルプロンプトで、**root** で以下を入力し、古い PostgreSQL サーバーが実行中であることを確認します。

```
systemctl start postgresql.service
```

2. PostgreSQL データベースの全データをスクリプトファイルにダンプします。**root** で以下のコマンドを実行します。

```
su - postgres -c 'pg_dumpall > ~/pgdump_file.sql'
```

3. **root** で以下のコマンドを実行して、古いサーバーを停止します。

```
systemctl stop postgresql.service
```

4. 新規サーバーのデータディレクトリーを **root** として初期化します。

```
scl enable rh-postgresql13 -- postgresql-setup initdb
```

5. **root** で新しいサーバーを起動します。

```
systemctl start rh-postgresql13-postgresql.service
```

6. 以前に作成した SQL ファイルからデータをインポートします。

```
su - postgres -c 'scl enable rh-postgresql13 "psql -f ~/pgdump_file.sql postgres"'
```

7. 必要に応じて、システムの起動時に PostgreSQL 13 サーバーが自動的に起動するように設定できます。古いシステム PostgreSQL サーバーを無効にするには、**root** で以下のコマンドを入力します。

```
chkconfig postgresql off
```

PostgreSQL 13 サーバーを有効にするには、**root** で以下を入力します。

```
chkconfig rh-postgresql13-postgresql on
```

8. デフォルトの設定と異なる場合は、設定ファイル (特に `/var/opt/rh/rh-postgresql13/lib/pgsql/data/pg_hba.conf` 設定ファイル) を必ず更新してください。それ以外の場合は、**postgres** ユーザーのみがデータベースにアクセスできます。

5.3.2. PostgreSQL 12 Software Collection から PostgreSQL 13 Software Collection への移行

データを rh-postgresql12 Software Collection から rh-postgresql13 コレクションに移行するには、**pg_upgrade** ツール (推奨) を使用して高速アップグレードを実行するか、SQL コマンドでデータベースデータをテキストファイルにダンプして新しいデータベースにインポートします。2つ目の方法は、通常大幅に遅くなるため、手動で修正が必要になる場合があります。このアップグレード方法の詳細は、[PostgreSQL ドキュメント](#) を参照してください。



重要

データを PostgreSQL 12 から PostgreSQL 13 に移行する前に、デフォルトで `/var/opt/rh/rh-postgresql12/lib/pgsql/data/` ディレクトリーにある PostgreSQL データベースファイルを含む、すべてのデータのバックアップを作成してください。

手順5.3 pg_upgrade ツールを使用した高速アップグレード

PostgreSQL サーバーの高速アップグレードを実行するには、以下の手順を実行します。

1. 古い PostgreSQL サーバーを停止し、データが一貫性のない状態にあることを確認します。これを行うには、**root** で次のコマンドを実行します。

```
systemctl stop rh-postgresql12-postgresql.service
```

サーバーが起動していないことを確認するには、以下を入力します。

```
systemctl status rh-postgresql12-postgresql.service
```

2. 古いディレクトリー `/var/opt/rh/rh-postgresql12/lib/pgsql/data/` が存在することを確認します。

```
file /var/opt/rh/rh-postgresql12/lib/pgsql/data/
```

データのバックアップを作成します。

3. 新しいデータディレクトリー `/var/opt/rh/rh-postgresql13/lib/pgsql/data/` が存在しないことを確認します。

```
file /var/opt/rh/rh-postgresql13/lib/pgsql/data/
```

PostgreSQL 13 の新規インストールを実行している場合は、このディレクトリーはシステムに存在しないはずですが、その場合は、**root** で以下のコマンドを実行してバックアップを作成します。

```
mv /var/opt/rh/rh-postgresql13/lib/pgsql/data{-,scl-backup}
```

4. **root** で以下のコマンドを実行して、新しいサーバーのデータベースデータをアップグレードします。

```
scl enable rh-postgresql13 -- postgresql-setup --upgrade --upgrade-from=rh-postgresql12-postgresql
```

または、`/opt/rh/rh-postgresql13/root/usr/bin/postgresql-setup --upgrade --upgrade-from=rh-postgresql12-postgresql` コマンドを使用できます。

別のバージョンの PostgreSQL からアップグレードする場合には、`--upgrade-from` オプションを使用できます。可能なアップグレードシナリオのリストは、`--upgrade-ids` オプションを使用して利用できます。

作成された `/var/lib/pgsql/upgrade_rh-postgresql13-postgresql.log` ログファイルを読み取り、アップグレード中に問題が発生したかどうかを確認することが推奨されます。

5. `root` で新しいサーバーを起動します。

```
systemctl start rh-postgresql13-postgresql.service
```

また、以下のように `analyze_new_cluster.sh` スクリプトを実行することが推奨されます。

```
su - postgres -c 'scl enable rh-postgresql13 ~/analyze_new_cluster.sh'
```

6. 必要に応じて、システムの起動時に PostgreSQL 13 サーバーが自動的に起動するように設定できます。古い PostgreSQL 12 サーバーを無効にするには、`root` で以下のコマンドを入力します。

```
chkconfig rh-postgresql12-postgresql off
```

PostgreSQL 13 サーバーを有効にするには、`root` で以下を入力します。

```
chkconfig rh-postgresql13-postgresql on
```

7. デフォルトの設定と異なる場合は、設定ファイル (特に `/var/opt/rh/rh-postgresql13/lib/pgsql/data/pg_hba.conf` 設定ファイル) を必ず更新してください。それ以外の場合は、`postgres` ユーザーのみがデータベースにアクセスできます。

手順5.4 ダンプおよびリストアアップグレードの実行

PostgreSQL サーバーのダンプおよび復元アップグレードを実行するには、以下の手順を実行します。

1. シェルプロンプトで、`root` で以下を入力し、古い PostgreSQL サーバーが実行中であることを確認します。

```
systemctl start rh-postgresql12-postgresql.service
```

2. PostgreSQL データベースの全データをスクリプトファイルにダンプします。`root` で以下のコマンドを実行します。

```
su - postgres -c 'scl enable rh-postgresql12 "pg_dumpall > ~/pgdump_file.sql"'
```

3. `root` で以下のコマンドを実行して、古いサーバーを停止します。

```
systemctl stop rh-postgresql12-postgresql.service
```

4. 新規サーバーのデータディレクトリーを `root` として初期化します。

```
scl enable rh-postgresql13 -- postgresql-setup initdb
```

5. **root** で新しいサーバーを起動します。

```
systemctl start rh-postgresql13-postgresql.service
```

6. 以前に作成した SQL ファイルからデータをインポートします。

```
su - postgres -c 'scl enable rh-postgresql13 "psql -f ~/pgdump_file.sql postgres"'
```

7. 必要に応じて、システムの起動時に PostgreSQL 13 サーバーが自動的に起動するように設定できます。古い PostgreSQL 12 サーバーを無効にするには、**root** で以下のコマンドを入力します。

```
chkconfig rh-postgresql12-postgresql off
```

PostgreSQL 13 サーバーを有効にするには、**root** で以下を入力します。

```
chkconfig rh-postgresql13-postgresql on
```

8. デフォルトの設定と異なる場合は、設定ファイル (特に `/var/opt/rh/rh-postgresql13/lib/pgsql/data/pg_hba.conf` 設定ファイル) を必ず更新してください。それ以外の場合は、**postgres** ユーザーのみがデータベースにアクセスできます。

5.4. NGINX 1.20 への移行

rh-nginx120 Software Collection のルートディレクトリーは `/opt/rh/rh-nginx120/root/` にあります。エラーログは、デフォルトで `/var/opt/rh/rh-nginx120/log/nginx` に保存されます。

設定ファイルは `/etc/opt/rh/rh-nginx120/nginx/` ディレクトリーに保存されます。nginx 1.20 の設定ファイルは、これまでの nginx Software Collections と同じ構文で、ほぼ同じフォーマットになっています。

`/etc/opt/rh/rh-nginx120/nginx/default.d/` ディレクトリーの設定ファイル (`.conf` 拡張子) は、ポート 80 のデフォルトのサーバーブロック設定に含まれます。



重要

nginx 1.18 から nginx 1.20 にアップグレードする前に、`/opt/rh/nginx118/root/` ツリーにある Web ページ、および `/etc/opt/rh/nginx118/nginx/` ツリーにある設定ファイルを含むすべてのデータのバックアップを作成します。

設定ファイルの変更や Web アプリケーションのセットアップなど、特定の変更を加えた場合、`/opt/rh/nginx118/root/` ツリーでこれらの変更を新しい `/opt/rh/rh-nginx120/root/` ディレクトリーおよび `/etc/opt/rh/rh-nginx120/nginx/` ディレクトリーに対して複製します。

この手順に従って、nginx 1.16 から nginx 1.20 へ直接アップグレードできます。この場合は、適切なパスを使用してください。

nginx の公式ドキュメントは、<http://nginx.org/en/docs/> を参照してください。

5.5. REDIS 6 への移行

rh-redis5 Software Collection で提供される **Redis 5.0** は、主に **Redis 6.0** の狭義のサブセットです。したがって、バージョン 5.0 からバージョン 6.0 へのアップグレード時には、主要な問題が発生しません。

Redis Cluster をバージョン 6.0 にアップグレードするには、全インスタンスを大量に再起動する必要があります。

互換性に関する注意事項

- セットキーが存在しない場合は、**SPOP <count>** コマンドは null を返しなくなりました。**Redis 6** では、コマンドは、このシナリオでは空のセットを返します。これは、引数 **0** で呼び出された場合と同様です。
- 後方互換性のないマイナーな変更は、[アップストリームのリリースノート](#) を参照してください。

第6章 関連情報

この章では、Red Hat Software Collections 3.8 および Red Hat Enterprise Linux に関するその他の関連情報ソースへの参照を提供します。

6.1. RED HAT 製品ドキュメント

以下のドキュメントは直接的または間接的にこのガイドに関連しています。

- [Red Hat Software Collections 3.8 パッケージガイド](#) : Red Hat Software Collections の『パッケージガイド』は、Software Collections の概念や **scl** ユーティリティーを説明し、カスタム Software Collection の作成方法や既存 Software Collection の拡張方法を詳細に説明しています。
- [Red Hat Developer Toolset 12.1 リリースノート](#) - Red Hat Developer Toolset の『リリースノート』には、既知の問題、考えられる問題、変更点、およびこの Software Collection に関するその他の重要な情報が記載されています。
- [Red Hat Developer Toolset 12 ユーザーガイド](#) - Red Hat Developer Toolset の『ユーザーガイド』には、この Software Collection のインストールおよび使用に関する詳細情報が記載されています。
- [Using Red Hat Software Collections Container Images](#) : 本書は、Red Hat Software Collections に基づくコンテナイメージの使用方法に関する情報を提供します。利用可能なコンテナイメージには、アプリケーション、デーモン、データベース、および Red Hat Developer Toolset コンテナイメージが含まれます。イメージは、Red Hat Enterprise Linux 7 Server および Red Hat Enterprise Linux Atomic Host で実行できます。
- [Getting Containers](#) では、Red Hat Enterprise Linux 7 および Red Hat Enterprise Linux Atomic Host でのコンテナイメージのビルドおよび使用に関する包括的な概要を説明します。
- [Red Hat Subscription Management の使用](#) - 『Red Hat Subscription Management の使用』ガイドでは、Red Hat Enterprise Linux システムの登録、サブスクリプションの管理、および登録済みシステムの通知の表示方法の詳細情報を提供します。
- [Red Hat Enterprise Linux 7 システム管理者のガイド](#) - Red Hat Enterprise Linux 7 の『システム管理者のガイド』は、このシステムのデプロイメント、設定、および管理に関する情報を提供します。

6.2. RED HAT 開発者

- [Red Hat Developer Program](#): 『Red Hat Developers』 コミュニティポータル
- [Overview of Red Hat Software Collections on Red Hat Developers](#) 『Red Hat Developers』ポータルでは、さまざまな開発技術を使用してコードを開発するためのチュートリアルがいくつか紹介されています。これには、Node.js、Perl、PHP、Python、Ruby Software Collections が含まれます。
- [Red Hat Developer Blog](#) - 『Red Hat Developer Blog』には、最新の情報、ベストプラクティス、意見、製品およびプログラムアナウンス、ならびに Red Hat の技術に基づくアプリケーションを設計および開発するユーザー向けのサンプルコードやその他のリソースへのポインターが含まれます。

付録A 更新履歴

改訂 3.8-7 壊れたリンクとアンカーを修正しました。	Wed Dec 20 2023	Lenka Špačková
改訂 3.8-6 のrh-mariadb103 Software Collection は EOL です。	Mon Jul 03 2023	Lenka Špačková
改訂 3.8-5 Red Hat Developer Toolset 12.1 に関する情報を追加しました。	Tue May 23 2023	Lenka Špačková
改訂 3.8-4 Red Hat Developer Toolset 12.0 に関する情報を追加しました。	Tue Nov 22 2022	Lenka Špačková
改訂 3.8-3 廃止されたコンポーネントを更新しました。 非同期更新で利用可能なコンポーネントバージョンを更新しました。	Fri Nov 18 2022	Lenka Špačková
改訂 3.8-2 「 Apache httpd の変更 (非同期更新) 」を追加しました。	Thu Sep 29 2022	Lenka Špačková
改訂 3.8-1 Red Hat Software Collections 3.8 リリースノートをリリースしました。	Mon Nov 15 2021	Lenka Špačková
改訂 3.8-0 Red Hat Software Collections 3.8 Beta リリースノートも併せて参照してください。	Mon Oct 11 2021	Lenka Špačková