



Red Hat Software Collections 3

3.2 リリースノート

Red Hat Software Collections 3.2 のリリースノート

Red Hat Software Collections 3 3.2 リリースノート

Red Hat Software Collections 3.2 のリリースノート

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

法律上の通知

Copyright © 2022 | You need to change the HOLDER entity in the en-US/3.2_Release_Notes.ent file |.

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 3.2 リリースノートには、Red Hat Software Collections 3.2 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

第1章 RED HAT SOFTWARE COLLECTIONS 3.2	4
1.1. RED HAT SOFTWARE COLLECTIONS について	4
1.1.1. Red Hat Developer Toolset	4
1.2. 主な特長	4
1.3. RED HAT SOFTWARE COLLECTIONS 3.2 での変更点	16
1.3.1. 概要	16
アーキテクチャー	16
新しい Software Collections	16
Software Collections の更新	17
Red Hat Software Collections コンテナイメージ	17
1.3.2. Red Hat Developer Toolset の変更点	17
1.3.3. PHP の変更点	18
1.3.4. MySQL の変更点	18
アップストリームの MySQL 8.0 と rh-mysql80の主な違い。	19
1.3.5. Node.js の変更点	19
1.3.6. nginx の変更点	19
1.3.7. Varnish キャッシュの変更点	20
1.3.8. Git の変更点	20
1.3.9. Apache httpd の変更点	20
1.4. 互換性情報	21
1.5. 既知の問題	21
その他の注意事項	25
1.6. 非推奨の機能	26
第2章 インストールシステム	27
2.1. RED HAT SOFTWARE COLLECTIONS へのアクセス	27
2.1.1. Red Hat Subscription Management の使用	27
2.1.2. オプションチャンネルからのパッケージ	28
2.2. RED HAT SOFTWARE COLLECTIONS のインストール	31
2.2.1. Software Collection の個別インストール	31
2.2.2. オプションパッケージのインストール	31
2.2.3. デバッグ情報のインストール	32
2.3. RED HAT SOFTWARE COLLECTIONS のアンインストール	32
2.4. RED HAT SOFTWARE COLLECTIONS の再構築	32
第3章 使用法	33
3.1. RED HAT SOFTWARE COLLECTIONS の使用	33
3.1.1. Software Collection からの実行可能ファイルの実行	33
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	33
3.1.3. Software Collection からのシステムサービスの実行	34
Red Hat Enterprise Linux 6 の Software Collection からシステムサービスの実行	34
Red Hat Enterprise Linux 7 の Software Collection からシステムサービスの実行	34
3.2. SOFTWARE COLLECTION からの手動ページへのアクセス	35
3.3. RED HAT SOFTWARE COLLECTIONS を使用するアプリケーションのデプロイ	35
3.4. RED HAT SOFTWARE COLLECTIONS コンテナイメージ	35
第4章 個別の SOFTWARE COLLECTIONS の詳細	39
4.1. RED HAT DEVELOPER TOOLSET	39
4.2. RUBY ON RAILS 5.0	39
4.3. MONGODB 3.6	40
4.4. MONGODB 3.4	40
Red Hat Enterprise Linux 6 上の MongoDB 3.4	41

Red Hat Enterprise Linux 7 の MongoDB 3.4	41
4.5. MAVEN	42
4.6. パッケージャー	42
4.7. データベースコネクター	42
第5章 マイグレーション	45
5.1. MARIADB 10.2 への移行	45
5.1.1. rh-mariadb101 および rh-mariadb102 Software Collections 間の主な相違点	45
5.1.2. rh-mariadb101 から rh-mariadb102 Software Collection へのアップグレード	45
5.2. MYSQL 8.0 への移行	47
5.2.1. MySQL 5.7 と MySQL 8.0 間の主な相違点	47
rh-mysql80 ソフトウェアの収集に固有の相違点	47
MySQL 8.0 の一般的な変更点	47
5.2.2. rh-mysql80 Software Collection へのアップグレード	48
5.3. MYSQL 5.7 への移行	49
5.3.1. MySQL 5.6 と MySQL 5.7 の注目すべき相違点	49
5.3.2. rh-mysql57 Software Collection へのアップグレード	49
5.4. MONGODB 3.6 への移行	50
5.4.1. MongoDB 3.4 と MongoDB 3.6 の主な相違点	50
一般的な変更点	50
互換性の変更点	51
後方互換性のない機能	51
5.4.2. rh-mongodb34 から rh-mongodb36 Software Collection へのアップグレード	51
5.5. MONGODB 3.4 への移行	53
5.5.1. MongoDB 3.2 と MongoDB 3.4 との間の主な相違点	53
一般的な変更点	53
互換性の変更点	53
5.5.2. rh-mongodb32 から rh-mongodb34 Software Collection へのアップグレード	53
5.6. POSTGRESQL 10 への移行	55
5.6.1. Red Hat Enterprise Linux システムバージョンの PostgreSQL から PostgreSQL 10 Software Collection への移行	56
5.6.2. PostgreSQL 9.6 Software Collection から PostgreSQL 10 Software Collection への移行	59
5.7. POSTGRESQL 9.6 への移行	61
5.7.1. PostgreSQL 9.5 および PostgreSQL 9.6 間の主な違い	61
5.7.2. Red Hat Enterprise Linux システムバージョンの PostgreSQL から PostgreSQL 9.6 Software Collection への移行	63
5.7.3. PostgreSQL 9.5 Software Collection から PostgreSQL 9.6 Software Collection への移行	65
5.8. NGINX 1.14 への移行について	68
第6章 関連情報	69
6.1. RED HAT 製品ドキュメント	69
6.2. RED HAT 開発者	69
付録A 改訂履歴	71

第1章 RED HAT SOFTWARE COLLECTIONS 3.2

この章では、Red Hat Software Collections 3.2 のコンテンツセットの概要を説明します。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 3.2 は Red Hat Enterprise Linux 7 で利用可能であり、選択された新しいコンポーネントおよび以前にリリースされたコンポーネントは Red Hat Enterprise Linux 6 でも利用可能です。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 6 または 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/` ディレクトリにインストールされ、提供された `scl` ユーティリティ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。

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 3.2 コンポーネント](#)」には、Red Hat Software Collections 3.2 リリース時にサポートされているすべてのコンポーネントが記載されています。

表1.1 Red Hat Software Collections 3.2 コンポーネント

コンポーネント	Software Collection	説明
---------	---------------------	----

コンポーネント	Software Collection	説明
Red Hat Developer Toolset 8.0	devtoolset-8	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については、『Red Hat Developer Toolset User Guide』の Red Hat Developer Toolset Components 表を参照してください。
Perl 5.24.0	rh-perl524	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-perl524 Software Collection は、追加のユーティリティ、スクリプト、そして MySQL および PostgreSQL の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するための cpanm ユーティリティが提供されます。
Perl 5.26.1 ^[a]	rh-perl526	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-perl526 Software Collection は、追加のユーティリティ、スクリプト、そして MySQL および PostgreSQL の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するための cpanm ユーティリティが提供されます。rh-perl526 パッケージは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 perl526-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 7.0.27	rh-php70	PEAR 1.10 を搭載した PHP 7.0 のリリースで、言語機能の強化とパフォーマンスの向上が図られています。
PHP 7.1.8 ^[a]	rh-php71	PEAR 1.10、APCu 5.1.8 および強化された言語機能を備えた PHP 7.1 のリリース。
PHP 7.2.10 ^[a]	rh-php72	PEAR 1.10.5、APCu 5.1.12、および強化された言語機能を備えた PHP 7.2 のリリース。

コンポーネント	Software Collection	説明
Python 2.7.13	python27	多くの追加ユーティリティを備えた Python 2.7 のリリース。この Python バージョンでは、順序付けされたディクショナリータイプ、高速な I/O 操作、Python 3 との前方互換性など、さまざまな機能および機能拡張が提供されます。python27 Software Collections には Python 2.7.13 インタープリター が含まれています。これは、Web アプリケーションおよび mod_wsgi (httpd24 Software Collection でのみサポート)、MySQL データベースコネクタおよび PostgreSQL データベースコネクタ、ならびに numpy および scipy のプログラミングに役立つ拡張ライブラリーセットです。
Python 3.5.1	rh-python35	rh-python35 Software Collection には、 Python 3.5.1 インタプリタ 、Web アプリケーションのプログラミングに便利な拡張ライブラリー群、 mod_wsgi (httpd24 Software Collection でのみサポート)、PostgreSQL データベースコネクタ、 numpy および scipy が含まれています。
Python 3.6.3	rh-python36	rh-python36 Software Collection には Python 3.6.3 が含まれ、 f-strings 、 変数アノテーションの構文 、 非同期ジェネレーター および 内包表記 など多くの新機能が導入されています。さらに、Web アプリケーションのプログラミングに便利な拡張ライブラリーのセットが含まれており、 mod_wsgi (httpd24 Software Collection との併用のみ対応)、PostgreSQL データベースコネクタ、 numpy と scipy が含まれています。
Ruby 2.3.6	rh-ruby23	Ruby 2.3 のリリースです。このバージョンでは、Ruby 2.2、Ruby 2.0.0、Ruby 1.9.3 とのソースレベルでの後方互換性を維持しつつ ソースファイル内のすべての文字列リテラルをフリーズするコマンドラインオプション 、 安全なナビゲーション演算子 、および 複数のパフォーマンスの向上 が導入されています。
Ruby 2.4.3	rh-ruby24	Ruby 2.4 のリリースです。このバージョンでは、 ハッシュテーブルの改良 、 新しいデバッグ機能 、 Unicode の大文字小文字のマッピングのサポート 、 OpenSSL 1.1.0 のサポート など、複数のパフォーマンスの改善と強化が行われています。Ruby 2.4.0 は、Ruby 2.3、Ruby 2.2、Ruby 2.0.0、Ruby 1.9.3 とのソースレベルでの後方互換性を維持しています。
Ruby 2.5.0 ^[a]	rh-ruby25	Ruby 2.5 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。たとえば、 rescue 、 else 、および ensure キーワードによる ブロックの使用の簡素化 、新しい yield_self メソッド、 ブランチ および メソッド範囲測定 のサポート、新しい Hash#slice および Hash#transform_keys メソッド。Ruby 2.5.0 は、Ruby 2.4 とソースレベルの後方互換性を維持します。

コンポーネント	Software Collection	説明
Ruby on Rails 4.2.6	rh-ror42	Ruby 言語で書かれた Web アプリケーションフレームワークである Ruby on Rails 4.2 のリリース。今回のリリースでは、 Active Job 、 非同期メール 、 Adequate Record 、 Web Console 、 外国語キーのサポート などの機能が追加されています。Software Collection は、rh-ruby23 および rh-nodejs4 コレクションとともにサポートされます。
Ruby on Rails 5.0.1	rh-ror50	Ruby 言語で書かれた Web アプリケーションフレームワークの最新版である Ruby on Rails 5.0 をリリースしました。注目すべき新機能としては、 Action Cable 、 API モード 、 Rake ではなく rails CLI の独占使用 、 ActionRecord の属性 などがあります。Software Collection は、rh-ruby24 および rh-nodejs6 コレクションとともにサポートされます。
Scala 2.10.6 [a]	rh-scala210	オブジェクト指向言語と関数型言語の機能を統合した、Java プラットフォーム用の汎用プログラミング言語 Scala をリリースしました。
MariaDB 10.1.29	rh-mariadb101	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。このバージョンでは、 Galera Cluster のサポート が追加されています。
MariaDB 10.2.8	rh-mariadb102	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。このバージョンでは、 MariaDB バックアップ 、 フラッシュバック 、 再帰的共通テーブル式のサポート 、 ウィンドウ関数 、 JSON 関数 が追加されています。
MongoDB 3.2.10	rh-mongodb32	NoSQL データベースに分類されるクロスプラットフォームのドキュメント指向データベースシステム である MongoDB のリリースです。Software Collection には、mongo-java-driver パッケージバージョン 3.2.1 が含まれています。
MongoDB 3.4.9	rh-mongodb34	MongoDB のリリース。これは、NoSQL データベースとして分類されるクロスプラットフォームのドキュメント指向のデータベースシステムです。本リリースでは、新しいアーキテクチャーのサポートが導入され、 メッセージ圧縮 と decimal128 タイプのサポート 、 照合機能の向上 などが追加されます。

コンポーネント	Software Collection	説明
MongoDB 3.6.3 [a]	rh-mongodb36	MongoDB のリリース。これは、NoSQL データベースとして分類されるクロスプラットフォームのドキュメント指向のデータベースシステムです。本リリースでは、 変更ストリーム 、 再試行可能な書き込み 、および JSON スキーマ および その他の機能が導入されました。
MySQL 5.7.24	rh-mysql57	パフォーマンスの向上を含む多くの新機能と機能強化を提供する、MySQL のリリース。
MySQL 8.0.13 [a]	rh-mysql80	MySQL サーバーのリリース。 セキュリティ 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 9.5.14	rh-postgresql95	PostgreSQL のリリースで、 行レベルのセキュリティ制御 を含む多くの機能強化が行われ、レプリケーションの進捗状況の追跡が導入され、カラム数の多い大規模テーブルの扱いが改善され、ソートマシンやマルチ CPU マシンのパフォーマンスが向上しました。
PostgreSQL 9.6.10	rh-postgresql96	PostgreSQL のリリース。順次スキャン、結合、および集計の並列実行が導入され、同期レプリケーション、全文検索、ディレクションドライバー、postgres_fdw の機能が強化され、パフォーマンスが改善されています。
PostgreSQL 10.5 [a]	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。
Node.js 6.11.3	rh-nodejs6	複数の API の強化、パフォーマンスとセキュリティの改善、ECMAScript 2015 のサポート、npm 3.10.9 を提供する Node.js のリリースです。
Node.js 8.11.4 [a]	rh-nodejs8	V8 エンジンバージョン 6.0 、 npm 5.6.0 、 npx 、 セキュリティの強化 、 実験的な N-API のサポート 、 パフォーマンスの向上 など、複数の API の強化と新機能を提供する Node.js のリリースです。
Node.js 10.10.0 [a]	rh-nodejs10	Node.js のリリースでは、複数の API の機能拡張および新機能が提供されます。これには、 V8 エンジンバージョン 6.6 、 完全な N-API サポート 、および 安定性の改善 が含まれます。

コンポーネント	Software Collection	説明
nginx 1.10.2	rh-nginx110	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。このバージョンでは、 ダイナミックモジュールのサポート、HTTP/2 のサポート、Perl の統合、多数のパフォーマンスの向上 など、多くの新機能が導入されています。
nginx 1.12.1 ^[a]	rh-nginx112	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。このバージョンでは、 IP トランスペアレンシー、TCP/UDP ロードバランシングの改善、キャッシングパフォーマンスの向上 など、数多くの新機能が導入されています。
nginx 1.14.0 ^[a]	rh-nginx114	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。このバージョンでは、 ミラーモジュール、HTTP/2 サーバープッシュ、gRPC プロキシモジュール、多くのパフォーマンス改善 などの多くの機能を利用できます。
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 モジュールおよび mod_auth_mellon モジュールも含まれています。
Varnish Cache 5.2.1 ^[a]	rh-varnish5	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、個別の VCL ファイルと VCL ラベルを使用した Varnish 設定の shard ディレクター、実験的な HTTP/2 サポート、および改善 が含まれています。
Varnish Cache 6.0.0 ^[a]	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 3.3.9	rh-maven33	主に Java プロジェクトで使用される ソフトウェアのプロジェクトの管理および状況把握ツール である Maven のリリースです。このバージョンでは、 コア拡張メカニズムの改善 など、さまざまな機能強化が行われています。
Maven 3.5.0 ^[a]	rh-maven35	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)。このリリースでは、 新しいアーキテクチャーのサポートと、カラー化されたログ を含む多くの新機能が導入されています。

コンポーネント	Software Collection	説明
Git 2.18.1 ^[a]	rh-git218	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。このバージョンには、 Large File Storage (LFS) 拡張 が含まれます。
Redis 3.2.4	rh-redis32	永続的なキーバリュデータベース である Redis 3.2 のリリースです。
HAProxy 1.8.4 ^[a]	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。
Common Java Packages	rh-java-common	Software Collection は、他のコレクションで使用されている 共通の Java ライブラリーとツール を提供します。rh-java-common Software Collection は、devtoolset-4、devtoolset-3、rh-maven33、maven30、rh-mongodb32、rh-mongodb26、thermostat1、rh-thermostat16、rh-eclipse46 コンポーネントで必要とされるものであり、ユーザーが直接インストールすることは想定されていません。
JDK Mission Control ^[a]	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 8 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 11 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-maven35 Software Collection が必要です。
[a] この Software Collection は、Red Hat Enterprise Linux 7 でのみ利用できます。		

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 Software Collections 3.2 の新コンポーネント			
Red Hat Developer Toolset 8.0	devtoolset-8	RHEL6、RHEL7	x86_64、s390x、aarch64、ppc64、ppc64le
PHP 7.2.10	rh-php72	RHEL7	x86_64、s390x、aarch64、ppc64le
MySQL 8.0.13	rh-mysql80	RHEL7	x86_64、s390x、aarch64、ppc64le
Node.js 10.10.0	rh-nodejs10	RHEL7	x86_64、s390x、aarch64、ppc64le
nginx 1.14.0	rh-nginx114	RHEL7	x86_64、s390x、aarch64、ppc64le
Varnish Cache 6.0.0	rh-varnish6	RHEL7	x86_64、s390x、aarch64、ppc64le
Git 2.18.1	rh-git218	RHEL7	x86_64、s390x、aarch64、ppc64le
JDK Mission Control	rh-jmc	RHEL7	x86_64

Red Hat Software Collections 3.2 で更新されたコンポーネント

Apache httpd 2.4.34	httpd24	RHEL6、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 5.26.1	rh-perl526	RHEL7	x86_64、s390x、aarch64、ppc64le
Ruby 2.5.0	rh-ruby25	RHEL7	x86_64、s390x、aarch64、ppc64le
MongoDB 3.6.3	rh-mongodb36	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

Varnish Cache 5.2.1	rh-varnish5	RHEL7	x86_64, s390x, aarch64, ppc64le
PostgreSQL 10.5	rh-postgresql10	RHEL7	x86_64, s390x, aarch64, ppc64le
HAProxy 1.8.4	rh-haproxy18	RHEL7	x86_64
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

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 3.6.3	rh-python36	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64le
Maven 3.5.0	rh-maven35	RHEL7	x86_64, s390x, aarch64, ppc64le
MariaDB 10.2.8	rh-mariadb102	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64le
PostgreSQL 9.6.10	rh-postgresql96	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64le
MongoDB 3.4.9	rh-mongodb34	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64le
Node.js 8.11.4	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
-------------------------------	---------------	--------------	--

Red Hat Software Collections 2.4 で最後に更新されたコンポーネント

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 2.4.3	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 2.7.13	python27	RHEL6、RHEL7	x86_64、s390x、aarch64、ppc64le
Thermostat 1.6.6	rh-thermostat16*	RHEL6、RHEL7	x86_64
Maven 3.3.9	rh-maven33	RHEL6、RHEL7	x86_64
Common Java Packages	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 2.3.6	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
-------------------------------	---------------	-------------	--------

Red Hat Software Collections 2.2 で最後に更新されたコンポーネント

MariaDB 10.1.29	rh-mariadb101	RHEL6、RHEL7	x86_64
MongoDB 3.0.11 upgrade collection	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

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

Red Hat Software Collections 2.0 で最後に更新されたコンポーネント

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
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: AMD64 および Intel 64 アーキテクチャー
- s390x – IBM Z
- 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一覧表示されます。

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 3.2 での変更点

1.3.1. 概要

アーキテクチャー

Red Hat Software Collections には、AMD64 および Intel 64 アーキテクチャー上で動作する Red Hat Enterprise Linux 7 用のパッケージが含まれていますが、一部の Software Collections は Red Hat Enterprise Linux 6 でもご利用いただけます。

さらに、Red Hat Software Collections 3.2 は、Red Hat Enterprise Linux 7 の以下のアーキテクチャーをサポートします。

- 64 ビット ARM アーキテクチャー
- 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 3.2 では、以下の新しい Software Collection が追加されました。

- devtoolset-8: [「Red Hat Developer Toolset の変更点」](#) を参照してください。
- rh-php72: [「PHP の変更点」](#) を参照してください。
- rh-mysql80: [「MySQL の変更点」](#) を参照してください。
- rh-nodejs10: [「Node.js の変更点」](#) を参照してください。
- rh-nginx114: [「Nginx の変更点」](#) を参照してください。
- rh-varnish6: [「Varnish キャッシュの変更点」](#) を参照してください。
- rh-git218: [「Git の変更点」](#) を参照してください。
- rh-jmc: JDK Mission Control(JMC) は、HotSpot JVM の強力なプロファイラーです。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。ツールチェーンを使用すると、開発者および管理者は、実稼働環境でローカルに実行しているまたはデプロイした Java アプリケーションのデータを収集および分析できます。

JMC を実行するには、JDK バージョン 8 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 11 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 は、[RHEA-2019:0543](#) アドバイザリーで利用できます。

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 3.2 で以下のコンポーネントが更新されました。

- httpd24: [「Apache httpd の変更点」](#) 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 3.2 で新しく加われました。

- rhsc/devtoolset-8-toolchain-rhel7
- rhsc/devtoolset-8-perftools-rhel7
- rhsc/mysql-80-rhel7
- rhsc/nginx-114-rhel7
- rhsc/php-72-rhel7
- rhsc/varnish-6-rhel7

Red Hat Software Collections 3.2 で以下のコンテナイメージが更新されました。

- rhsc/httpd-24-rhel7

Red Hat Software Collections のコンテナイメージに関する詳細は、[「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 の変更点

Red Hat Developer Toolset 8.0 では、以下のコンポーネントが Red Hat Developer Toolset の旧リリースと比較してアップグレードされています。

- GCC のバージョンを 8.2.1 に変更
- GDB をバージョン 8.2 に更新
- Valgrind をバージョン 3.14.0 に更新
- elfutils をバージョン 0.174 に更新
- binutils のバージョンを 2.30 に変更
- strace のバージョンを 4.24 に更新
- OProfile をバージョン 1.3.0 に更新
- SystemTap をバージョン 3.3 に更新

さらに、以下のコンポーネントに対するバグ修正の更新を利用できます。

- `dwz`
- `ltrace`
- `Dyninst`

8.0 での変更点の詳細については、[Red Hat Developer Toolset User Guide](#) を参照してください。

1.3.3. PHP の変更点

新しい `rh-php2` Software Collection には、**PEAR 1.10.5**、**APCu 5.1.12**、および改善された言語機能を備えた **PHP 7.2.10** が含まれます。このバージョンでは、以下の機能拡張が追加されました。

- `object-to-array` および `array-to-object` キャスト内の数値キーの変換
- 非対応オブジェクトの数
- 新しい **object** タイプヒント
- **HashContext** がリソースからオブジェクトに変更
- TLS 定数の改善
- パフォーマンスの向上

`rh-php72` で提供されるバグフィックスや機能強化の詳細については、[change log](#) をご覧ください。PHP 7.1 から PHP 7.2 への移行に関する情報は、[upstream migration guide](#) を参照してください。

1.3.4. MySQL の変更点

新しい `rh-mysql80` Software Collection には **MySQL 8.0.13** が含まれ、多くの新しいセキュリティーおよびアカウント管理機能および機能強化が追加されました。以下は、主な変更点です。

- **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 [transactional data dictionary](#) 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。ロールは特権のコレクションです。
- 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 expressions](#) に対応するようになりました。
- **MySQL** が、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 functions](#) に対応するようになりました。
- `locking read` ステートメントにより、**InnoDB** で、**NOWAIT** および **SKIP LOCKED** オプションに対応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。

詳細な変更は、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 in MySQL 8.0](#) を参照してください。

移行手順については、「[MySQL 8.0 への移行](#)」を参照してください。

アップストリームの MySQL 8.0 と rh-mysql80 の主な違い。

- 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。

```
[mysql]
default_authentication_plugin=caching_sha2_password
```

caching_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は、rh-mysql80* パッケージが提供するバイナリーおよびスクリプトの正しい機能に **scl enable** コマンドを使用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-mariadb102 Software Collection から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。

1.3.5. Node.js の変更点

新しい rh-nodejs10 Software Collection は、 **npm 6.4.1** で **Node.js 10.10.0** を提供します。本リリースでは、主な機能は次のとおりです。

- V8 engine** バージョン 6.6
- N-API** のサポートは実験的なものではなくなりました。
- 安定性の改良
- 強化されたセキュリティー機能

Node.js 10.10.0 の詳細な変更は、[upstream release notes](#) および [upstream documentation](#) を参照してください。

1.3.6. nginx の変更点

新しい rh-nginx114 Software Collection には **nginx 1.14.0** が含まれ、以下のような多くのパフォーマンス改善、バグ修正、および新機能が提供されます。

- mirror** モジュール
- gRPC** プロキシモジュール
- HTTP/2 サーバーのプッシュ
- Vim** 構文強調表示スクリプトの改善

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 1.14 への移行について](#)」を参照してください。

1.3.7. Varnish キャッシュの変更点

新しい rh-varnish6 Software Collection に含まれる **Varnish Cache 6.0.0** は、以前にリリースされたバージョンにバグ修正および機能拡張が数多く追加されました。例:

- クライアントおよびバックエンドサーバーの両方に対する Unix Domain Sockets (UDS) のサポート
- Varnish Configuration Language(VCL)、**vcl 4.1**の新規レベル
- HTTP/2 サポートの改善
- 新規および改善された Varnish Modules(VMODs)
 - **vmod_directors**
 - **vmod_proxy**
 - **vmod_unix**
 - **vmod_vtc**

Varnish Cache 6.0.0 の詳細な変更は、[upstream change log](#) を参照してください。[upstream documentation](#) および [upgrading notes](#) も参照してください。

1.3.8. Git の変更点

新しい rh-git218 Software Collection には **Git 2.18.1**が含まれています。これは、Red Hat Software Collections 2.3 でリリースされた rh-git29 Collection と比較して、バグ修正および新機能が数多く追加されました。rh-git218 Software Collection に固有の主な変更点は、以下のとおりです。

- **lfs** 拡張機能が追加され、デフォルトで rh-git218 によりインストールされます。**Git Large File Storage (LFS)** は、大規模なファイルを **Git** 内のテキストポインターに置き換え、ファイルの内容をリモートサーバーに保存します。
- 新しい rh-git218-git-instaweb サブパッケージが利用できます。これは、Apache HTTP サーバーのベース Red Hat Enterprise Linux のバージョンによって異なります。rh-git218-git-instaweb パッケージをインストールすると、**git instaweb** コマンドは追加設定なしで Web サーバーで機能します。

Git 2.18.1に関するその他の機能拡張、バグ修正、および後方互換性に関する注意事項の詳細は、[upstream release notes](#)を参照してください。バージョン 2.18.1 の [Git manual page](#) を参照してください。

1.3.9. Apache httpd の変更点

httpd24 Software Collection が提供する **Apache HTTP Server** がアップストリームバージョン 2.4.34 に更新されました。以下は、主な変更点です。

- HTTP/2 のサポートが改善しました。
- **OpenSSL 1.0.2** が提供する追加機能が実装されました。
- 今回の更新で、**mod_md** モジュールが httpd24 Software Collection に追加されました。モ

ジュールにより、自動証明書管理環境 (ACME) プロトコルを使用して、仮想ホストおよび証明書のプロビジョニング全体でドメインを管理できます。**mod_md** モジュールは、Red Hat Enterprise Linux 7 でのみ利用できます。

- **Apache HTTP Server** での TLS Server Name Indication(SNI) ヒントの処理が変更されました。
 - TLS ハンドシェイクで指定された SNI ヒントが HTTP リクエストの **Host:** ヘッダーと一致しない場合、HTTP **421 Misdirected Request** エラー応答は、以前の **400 Bad Request** エラー応答ではなくサーバーによって送信されるようになりました。
 - SNI ヒントが設定済みの **VirtualHost** のサーバー名と一致しない場合は、通常の **VirtualHost** マッチングルールの後に、最初に設定されたホストと一致します。以前のバージョンでは、**400 Bad Request** エラーの応答が送信されました。

Apache httpd 2.4.34 の変更点の詳細については、[upstream release notes](#) を参照してください。

1.4. 互換性情報

Red Hat Software Collections 3.2 は、AMD64 および Intel 64 アーキテクチャー、64 ビット ARM アーキテクチャー、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。

また、特定のコンポーネントが、AMD64 および Intel 64 アーキテクチャー上の Red Hat Enterprise Linux 6 の全サポートリリースに向けて提供されています。

利用可能なコンポーネントの一覧は、[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-mysql80, BZ#1646363

mysql-connector-java データベースコネクタは MySQL 8.0 サーバーでは動作しません。

rh-mysql80, 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-mysqlnd** データベースコネクタではサポートされません。そのため、**php-mysqlnd** 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

httpd24 Software Collection に含まれる **cURL** ツールの更新バージョンは、HTTP/2 をサポートしません。そのため、このバージョンの **cURL** の HTTP/2 サポートに依存するスクリプトは失敗するか、HTTP/1.1 にフォールバックします。

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、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#1327548

`mod_ssl` モジュールは、Red Hat Enterprise Linux 6 または Red Hat Enterprise Linux 7.3 以前で ALPN プロトコルに対応していません。そのため、ALPN を使用した TLS 接続の HTTP/2 へのアップグレードをサポートするクライアントは、HTTP/1.1 のサポートに制限されます。

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 を有効にすると、`LD_LIBRARY_PATH` 環境変数は `httpd` が呼び出す CGI スクリプトに渡されません。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。

rh-python35、rh-python36 コンポーネント、BZ#1499990

`Babel` がタイムゾーンのサポートに使用している `pytz` モジュールは、rh-python35、および rh-python36 Software Collections には含まれていません。その結果、ユーザーが `Babel` から `dates` モジュールをインポートしようとすると、トレースバックが返されます。この問題を回避するには、`pip install pytz` コマンドを使用して、`pypi` 公開リポジトリから `pip` パッケージマネージャーを介して `pytz` をインストールします。

rh-python36 コンポーネント ()

`numpy` が提供する特定の複雑な三角関数は、64 ビット ARM アーキテクチャー、IBM Z、および IBM POWER (リトルエンディアン) で誤った値を返す可能性があります。AMD64 および Intel 64 のアーキテクチャーは、この問題の影響を受けません。

python27 コンポーネント、BZ#1330489

python27-python-pymongo パッケージがバージョン 3.2.1 に更新されました。このバージョンは、これまでに同梱されているバージョン 2.5.2 と完全に互換性がないことに注意してください。詳細は <https://api.mongodb.org/python/current/changelog.html> を参照してください。

python27 コンポーネント

Red Hat Enterprise Linux 7 では、ユーザーが `python27-python-debuginfo` パッケージをインストールしようとすると、`/usr/src/debug/Python-2.7.5/Modules/socketmodule.c` ファイルが、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衝突します。その結果、`python27-python-debuginfo` のインストールに失敗します。この問題を回避するには、`python-debuginfo` パッケージをアンインストールしてから、`python27-python-debuginfo` パッケージをインストールしてください。

scl-utils コンポーネント で修正済み

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。

rh-ruby24、rh-ruby23 コンポーネント

RubyGem のインストールパスは、複数の Software Collection が有効になっている順番に応じて決定されます。依存するコレクションをサポートするために、Red Hat Software Collections 2.3 で出荷された **Ruby 2.3.1** から必要な順序が変更されました。その結果、RPM ビルド時に **gem** のインストールに使用される **RubyGem** のパスは、Software Collection が誤った順序で提供されている場合には無効になります。例えば、RPM のスペックファイルに **scl enable rh-ror50 rh-nodejs6** が含まれている場合、ビルドに失敗するようになりました。この問題を回避するには、rh-ror50 Software Collection を最後に有効にします。例えば、**scl enable rh-nodejs6 rh-ror50** のようにします。

rh-maven35、rh-maven33 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 パッケージと rh-maven35-maven-local package または rh-maven33-maven-local package の両方をインストールしている場合、rh-maven35 または rh-maven33 Software Collection から実行される Java RPM パッケージの構築に使用されるツール **XMvn** がベースシステムから設定ファイルを読み取ろうとして失敗します。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` モジュールを使用できません。

postgresql コンポーネント

Red Hat Enterprise Linux 6 の rh-postgresql9* パッケージは、**sepgsql** モジュールを提供しません。この機能には、Red Hat Enterprise Linux 6 では利用できない libselinux バージョン 2.0.99 のインストールが必要であるためです。

httpd、mariadb、mongodb、mysql、nodejs、perl、php、python、ruby、および ror コンポーネント、BZ#1072319

httpd24、rh-mariadb*、rh-mongodb*、rh-mysql*、rh-nodejs*、rh-perl*、rh-php*、python27、rh-python*、rh-ruby*、または rh-ror* パッケージをアンインストールする際に、依存するパッケージの所有権により、アンインストールの順番が関係することがあります。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、mongodb コンポーネント)

Red Hat Software Collections 3.2 に

は、MySQL 5.7、MySQL 8.0、MariaDB 10.0、MariaDB 10.1、MariaDB 10.2、PostgreSQL 9.5、およびMongoDB 3.6 のデータベースが含まれます。Red Hat Enterprise Linux 6 のコアは、以前のバージョンの MySQL 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) を提供します。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向けとしてのみ含まれているので、使用しないでください。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複数のデータベースバージョン間で安定するため、たとえば PostgreSQL 9.4 デーモンまたは 9.5 デーモンで PostgreSQL 9.2 クライアントライブラリーを使用すると期待どおりに機能します。

コアの Red Hat Enterprise Linux 6 および Red Hat Enterprise Linux 7 には MongoDB のクライアントライブラリーが含まれていません。アプリケーションにこのクライアントライブラリーを使用するには、Red Hat Software Collections からのクライアントライブラリーを使用し、この MongoDB クライアントライブラリーにリンクされたアプリケーションを実行するたびに **scl enable ...** 呼び出しを常に使用する必要があります。

mariadb、mysql、mongodb コンポーネント ()

ログファイルの作成時に、MariaDB、MySQL、および MongoDB は **/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。

rh-eclipse46 コンポーネント

サードパーティーの更新サイトからプラグインをインストールされていると、Eclipse が起動に失敗し、ワークスペースのログファイルに **NullPointerException** が発生することがあります。この問題を回避するには、**-clean** オプションを付けて Eclipse を再起動してください。例:

```
~]$ scl enable rh-eclipse46 "eclipse -clean"
```

rh-eclipse46 コンポーネント

Eclipse Docker Tooling では、シンタックスハイライトや基本的なコマンドの自動補完機能を備えた Dockerfile editor を導入しています。Build Image Wizard を開いた状態で、Edit Dockerfile ボタンを押すと、Dockerfile editor がファイルを切り離れたエディターウィンドウで開きます。ただし、このウィンドウには Cancel と Save のボタンはありません。この問題を回避するには **Ctrl+S** を押して変更を保存するか、エディター内で右クリックしてコンテキストメニューを表示し、Save オプションを選択します。変更をキャンセルするには、ウィンドウを閉じます。

rh-eclipse46 コンポーネント

Red Hat Enterprise Linux 7.2 では、Eclipse の **Perf Profile View** の入力に使用される perf ツールにバグがあり、ビュー内の一部のアイテムが Eclipse エディター内のそれぞれの位置に正しくリンクされませんでした。プロファイリングは期待通りに行われますが、**Perl Profile View** の一部をクリックしても、エディター内の関連するポジションに移動することはできません。

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、Red Hat Software Collections バージョンの **Apache httpd** でのみサポートされ、Red Hat Enterprise Linux システムバージョンの **httpd** に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せん。

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、全ケースで環境設定がエクスポートされるわけではなく、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。

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 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_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。

1.6. 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 インストールシステム

本章では、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 3.2 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、<https://access.redhat.com/solutions/472793> に記載されている Red Hat Enterprise Linux 6 および Red Hat Enterprise Linux 7 サブスクリプションをご利用いただけます。Red Hat Subscription Management (RHSM) でシステムを登録する方法は、[Using and Configuring Red Hat Subscription Manager](#) 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。なお、「[オプションチャンネルからのパッケージ](#)」に掲載されている **Optional** チャンネルを必要とするパッケージは、ISO イメージからはインストールできません。



注記

Optional チャンネルを必要とするパッケージは、ISO イメージからはインストールできません。**Optional** チャンネルの有効化を必要とするパッケージのリストは、「[オプションチャンネルからのパッケージ](#)」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一覧を表示し、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** で始まる行に一覧表示されます。

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一覧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 リポジトリーの一覧を表示して、リポジトリーメタデータを取得し、Red Hat Software Collections リポジトリー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。

リポジトリー名は、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-rhsc1-6-rpms
rhel-variant-rhsc1-6-debug-rpms
rhel-variant-rhsc1-6-source-rpms

rhel-server-rhsc1-6-eus-rpms
rhel-server-rhsc1-6-eus-source-rpms
rhel-server-rhsc1-6-eus-debug-rpms

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

rhel-server-rhsc1-7-eus-rpms
rhel-server-rhsc1-7-eus-source-rpms
rhel-server-rhsc1-7-eus-debug-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、適切なリポジトリーを有効にします。

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、[Using and Configuring Red Hat Subscription Manager](#) 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。RHSM への移行方法は <https://access.redhat.com/products/red-hat-subscription-management/#migration> を参照してください。

2.1.2. オプションチャンネルからのパッケージ

Red Hat Software Collections 3.2 パッケージの中には、これら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** チャンネルを有効にする必要があるものがあります。このチャンネル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 <https://access.redhat.com/solutions/392003> を参照してください。

Software Collections for Red Hat Enterprise Linux 6 のパッケージのうち、**Optional** チャンネルを有効にする必要があるものを以下の表に示します。

表2.1 Red Hat Enterprise Linux 6 のオプションチャンネルの有効化を必要とするパッケージ

Software Collection からのパッケージ	オプションチャンネルからの必須パッケージ
devtoolset-6-dyninst-testsuite	glibc-static
devtoolset-7-dyninst-testsuite	glibc-static
devtoolset-8-dyninst-testsuite	glibc-static
rh-git29-git-all	cvsp, perl-Net-SMTP-SSL
rh-git29-git-cvs	cvsp
rh-git29-git-email	perl-Net-SMTP-SSL
rh-git29-perl-Git-SVN	perl-YAML、 subversion-perl
rh-mariadb101-boost-devel	libc-devel
rh-mariadb101-boost-examples	libc-devel
rh-mariadb101-boost-static	libc-devel
rh-mongodb32-boost-devel	libc-devel
rh-mongodb32-boost-examples	libc-devel
rh-mongodb32-boost-static	libc-devel
rh-mongodb32-yaml-cpp-devel	libc-devel
rh-mongodb34-boost-devel	libc-devel
rh-mongodb34-boost-examples	libc-devel
rh-mongodb34-boost-static	libc-devel
rh-mongodb34-yaml-cpp-devel	libc-devel
rh-php70-php-imap	libc-client
rh-php70-php-recode	recode

Red Hat Enterprise Linux 7 で **Optional** チャンネルを必要とする Software Collection パッケージを以下の表に示します。

表2.2 Red Hat Enterprise Linux 7 のオプションチャンネルの有効化を必要とするパッケージ

Software Collection からのパッケージ	オプションチャンネルからの必須パッケージ
devtoolset-7-dyninst-testsuite	glibc-static
devtoolset-7-gcc-plugin-devel	libmpc-devel
devtoolset-8-dyninst-testsuite	glibc-static
devtoolset-8-gcc-plugin-devel	libmpc-devel
httpd24-mod_ldap	apr-util-ldap
rh-eclipse46	ruby-doc
rh-eclipse46-eclipse-dltk-ruby	ruby-doc
rh-eclipse46-eclipse-dltk-sdk	ruby-doc
rh-eclipse46-eclipse-dltk-tests	ruby-doc
rh-eclipse46-icu4j-javadoc	java-1.8.0-openjdk-javadoc
rh-eclipse46-stringtemplate-javadoc	java-1.8.0-openjdk-javadoc
rh-git218-git-all	cvsp, subversion-perl
rh-git218-git-cvs	cvsp
rh-git218-git-svn	subversion-perl
rh-git218-perl-Git-SVN	subversion-perl
rh-git29-git-all	cvsp
rh-git29-git-cvs	cvsp
rh-git29-perl-Git-SVN	subversion-perl

なお、**Optional** チャンネルのパッケージはサポートされていません。詳細は、ナレッジベースの記事 <https://access.redhat.com/articles/1150793> を参照してください。

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 3.2 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。以前のバージョンの Red Hat Software Collections 3.2 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 3.2 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、php54 および rh-mariadb100 をインストールし、**root** として以下を入力します。

```
~]# yum install rh-php72 rh-mariadb102
```

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一覧を表示するには、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一覧に置き換えます。たとえば、rh-perl526-perl-CPAN および rh-perl526-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-perl526-perl-CPAN rh-perl526-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-ruby25-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-ruby25-ruby
```

これらのパッケージを含みリポジトリ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** リポジトリを有効化します。debuginfo パッケージへのアクセス方法は<https://access.redhat.com/solutions/9907>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 の構築に関する詳細は、[Red Hat Software Collections Packaging Guide](#) を参照してください。

第3章 使用法

この章では、Red Hat Software Collections 3.2 を再構築して使用し、Red Hat Software Collections を使用するアプリケーションをデプロイするために必要な手順を説明します。

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一覧に置き換え、`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 3.2 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-postgresql10` 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-postgresql10 bash
```

現行セッションで有効になっている Software Collections の一覧は、`$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-postgresql10
```

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 3.2 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 6 の Software Collection からシステムサービスの実行

システムサービスを含む Software Collections は、対応する init スクリプトを `/etc/rc.d/init.d/` ディレクトリにインストールします。現行のセッションでそのようなサービスを起動するには、シェルプロンプトで **root** として以下を入力します。

```
service software_collection-service_name start
```

`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。

```
chkconfig software_collection-service_name on
```

たとえば、`rh-postgresql96` Software Collection から **postgresql** サービスを起動し、ランレベル 2、3、4、および 5 で **root** で有効にするには、**root** で次のコマンドを実行します。

```
~]# service rh-postgresql96-postgresql start
Starting rh-postgresql96-postgresql service:          [ OK ]
~]# chkconfig rh-postgresql96-postgresql on
```

Red Hat Enterprise Linux 6 でシステムサービスを管理する方法は、[Red Hat Enterprise Linux 6 デプロイメントガイド](#)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 3.2 コンポーネント」](#)を参照してください。

Red Hat Enterprise Linux 7 の Software Collection から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-postgresql10` 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-postgresql10-postgresql.service
~]# systemctl enable rh-postgresql10-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 3.2 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 ページはコンポーネントと同じ名前を持ち、`/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-mariadb102` の man ページを表示するには、以下を入力します。

```
~]$ scl enable rh-mariadb102 "man rh-mariadb102"
```

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、既存のソフトウェアを拡張する方法については、[Red Hat Software Collections Packaging Guide](#) 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、[Using Red Hat Software Collections 2 Container Images](#) を参照してください。

以下のコンテナイメージは Red Hat Software Collections 3.2 で利用可能です。

- `rhsc/devtoolset-8-toolchain-rhel7`
- `rhsc/devtoolset-8-perftools-rhel7`
- `rhsc/httpd-24-rhel7`
- `rhsc/mysql-80-rhel7`
- `rhsc/nginx-114-rhel7`
- `rhsc/php-72-rhel7`

- rhsc/varnish-6-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/devtoolset-7-toolchain-rhel7
- rhsc/devtoolset-7-perftools-rhel7
- rhsc/mongodb-36-rhel7
- rhsc/perl-526-rhel7
- rhsc/php-70-rhel7
- rhsc/postgresql-10-rhel7
- rhsc/ruby-25-rhel7
- rhsc/varnish-5-rhel7

以下のコンテナイメージは Red Hat Software Collections 3.0 をベースとしています。

- rhsc/mariadb-102-rhel7
- rhsc/mongodb-34-rhel7
- rhsc/nginx-112-rhel7
- rhsc/nodejs-8-rhel7
- rhsc/php-71-rhel7
- rhsc/postgresql-96-rhel7
- rhsc/python-36-rhel7

以下のコンテナイメージは、Red Hat Software Collections 2.4 に基づいています。

- rhsc/devtoolset-6-toolchain-rhel7 (EOL)
- rhsc/devtoolset-6-perftools-rhel7 (EOL)
- rhsc/nginx-110-rhel7
- rhsc/nodejs-6-rhel7
- rhsc/python-27-rhel7
- rhsc/ruby-24-rhel7
- rhsc/ror-50-rhel7
- rhsc/thermostat-16-agent-rhel7 (EOL)
- rhsc/thermostat-16-storage-rhel7 (EOL)

以下のコンテナイメージは、Red Hat Software Collections 2.3 に基づいています。

- rhsc/mysql-57-rhel7
- rhsc/perl-524-rhel7
- rhsc/redis-32-rhel7
- rhsc/mongodb-32-rhel7
- rhsc/php-56-rhel7 (EOL)
- rhsc/python-35-rhel7
- rhsc/ruby-23-rhel7

以下のコンテナイメージは、Red Hat Software Collections 2.2 に基づいています。

- rhsc/devtoolset-4-toolchain-rhel7 (EOL)
- rhsc/devtoolset-4-perftools-rhel7 (EOL)
- rhsc/mariadb-101-rhel7
- rhsc/nginx-18-rhel7 (EOL)
- rhsc/nodejs-4-rhel7 (EOL)
- rhsc/postgresql-95-rhel7
- rhsc/ror-42-rhel7
- rhsc/thermostat-1-agent-rhel7 (EOL)
- rhsc/varnish-4-rhel7 (EOL)

以下のコンテナイメージは、Red Hat Software Collections 2.0 に基づいています。

- rhsc/mariadb-100-rhel7 (EOL)
- rhsc/mongodb-26-rhel7 (EOL)
- rhsc/mysql-56-rhel7 (EOL)
- rhsc/nginx-16-rhel7 (EOL)
- rhsc/passenger-40-rhel7 (EOL)
- rhsc/perl-520-rhel7 (EOL)
- rhsc/postgresql-94-rhel7 (EOL)
- rhsc/python-34-rhel7 (EOL)
- rhsc/ror-41-rhel7 (EOL)
- rhsc/ruby-22-rhel7 (EOL)
- rhsc/s2i-base-rhel7

EOL(End of Life) と表示されているイメージはサポート対象外となります。

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、同梱の `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 Release Notes』の [Main Features](#) セクションを参照してください。

コンポーネントの完全なリストについては、『Red Hat Developer Toolset User Guide』の [Red Hat Developer Toolset Components](#) 表を参照してください。

なお、Red Hat Developer Toolset 3.1 以降、Red Hat Developer Toolset には `rh-java-common` Software Collection が必要です。

4.2. RUBY ON RAILS 5.0

Red Hat Software Collections 3.2 は `rh-ruby24` Software Collection を `rh-ror50` Collection と共に提供します。

Ruby on Rails 5.0 をインストールするには、`root` で以下のコマンドを入力します。

```
yum install rh-ror50
```

`rh-ror50` Software Collection のパッケージをインストールすると、依存関係にある `rh-ruby24` と `rh-nodejs6` が自動的に取り込まれます。

`rh-nodejs6` Collection は、アセットパイプライン内の特定の `gem` が、`sass` や `coffee-script` のソースファイルなどの Web リソースを後処理するために使用します。さらに、`Action Cable` フレームワークでは、Rails で **WebSockets** を扱うために `rh-nodejs6` を使用します。

`rh-nodejs6` を必要とせずに `rails s` コマンドを実行するには、`Gemfile` で `coffee-rails` と `uglifier` のジェムを無効にします。

Node.js を使わずに Ruby on Rails を実行するには、以下のコマンドを実行すると、自動的に `rh-ruby24` が有効になります。

```
scl enable rh-ror50 bash
```

Ruby on Rails をすべての機能で動作させるには、`rh-nodejs6` Software Collection も有効にしてください。

```
scl enable rh-ror50 rh-nodejs6 bash
```

rh-ror50 Software Collection は、rh-ruby24 と rh-nodejs6 のコンポーネントと共にサポートされています。

4.3. MONGODB 3.6

rh-mongodb36 Software Collection は、Red Hat Enterprise Linux 7 でのみ利用できます。Red Hat Enterprise Linux 6 で MongoDB 3.4 を使用方法は、「[MongoDB 3.4](#)」を参照してください。

rh-mongodb36 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-mongodb36
```

MongoDB シェルユーティリティを実行するには、以下のコマンドを入力します。

```
scl enable rh-mongodb36 'mongo'
```



注記

rh-mongodb36-mongo-cxx-driver パッケージは、Red Hat Developer Toolset 6 の GCC を使用して **-std=gnu++14** オプションで構築されています。C++11 (以降) の機能を使用する MongoDB C++ ドライバーの共有ライブラリーを使用するバイナリーは、Red Hat Developer Toolset 6 以降でもビルドする必要があります。[Red Hat Developer Toolset 6 User Guide](#) の C++ compatibility を参照してください。

MongoDB デーモンを起動するには、**root** で以下のコマンドを入力します。

```
systemctl start rh-mongodb36-mongod.service
```

システムの起動時に MongoDB デーモンを起動するには、**root** でこのコマンドを入力します。

```
systemctl enable rh-mongodb36-mongod.service
```

MongoDB シャードサーバーを起動するには、**root** で以下のコマンドを入力します。

```
systemctl start rh-mongodb36-mongos.service
```

システムの起動時に MongoDB シャードサーバーを起動するには、**root** でこのコマンドを入力します。

```
systemctl enable rh-mongodb36-mongos.service
```

少なくとも1つの設定サーバーを起動し、**mongos.conf** ファイルで指定しない限り、MongoDB シャーディングサーバーは機能しないことに注意してください。

4.4. MONGODB 3.4

rh-mongodb34 コレクションをインストールするには、**root** で次のコマンドを入力します。

```
yum install rh-mongodb34
```

MongoDB シェルユーティリティを実行するには、以下のコマンドを入力します。

```
scl enable rh-mongodb34 'mongo'
```



注記

rh-mongodb34-mongo-cxx-driver パッケージは、Red Hat Developer Toolset 6 のGCCを使用して**-std=gnu++14** オプションでビルドされています。C++11 (以降) の機能を使用する MongoDB C++ ドライバーの共有ライブラリーを使用するバイナリーは、Red Hat Developer Toolset 6 でもビルドする必要があります。[Red Hat Developer Toolset 6 User Guide](#) の C++ compatibility を参照してください。

Red Hat Enterprise Linux 6 上の MongoDB 3.4

Red Hat Enterprise Linux 6 を使用している場合は、以下の手順がシステムに適用されます。

MongoDB デーモンを起動するには、**root** で以下のコマンドを入力します。

```
service rh-mongodb34-mongod start
```

システムの起動時に MongoDB デーモンを起動するには、**root** でこのコマンドを入力します。

```
chkconfig rh-mongodb34-mongod on
```

MongoDB シャード化サーバーを起動するには、**root** でこのコマンドを入力します。

```
service rh-mongodb34-mongos start
```

起動時に MongoDB シャード化サーバーを起動するには、**root** で次のコマンドを入力します。

```
chkconfig rh-mongodb34-mongos on
```

少なくとも1つの設定サーバーを起動し、**mongos.conf** ファイルで指定しない限り、MongoDB シャーディングサーバーは機能しないことに注意してください。

Red Hat Enterprise Linux 7 の MongoDB 3.4

Red Hat Enterprise Linux 7 を使用する場合は、以下のコマンドを使用できます。

MongoDB デーモンを起動するには、**root** で以下のコマンドを入力します。

```
systemctl start rh-mongodb34-mongod.service
```

システムの起動時に MongoDB デーモンを起動するには、**root** でこのコマンドを入力します。

```
systemctl enable rh-mongodb34-mongod.service
```

MongoDB シャードサーバーを起動するには、**root** で以下のコマンドを入力します。

```
systemctl start rh-mongodb34-mongos.service
```

システムの起動時に MongoDB シャードサーバーを起動するには、**root** でこのコマンドを入力します。

```
systemctl enable rh-mongodb34-mongos.service
```

少なくとも1つの設定サーバーを起動し、**mongos.conf** ファイルで指定しない限り、MongoDB シャーディングサーバーは機能しないことに注意してください。

4.5. MAVEN

Red Hat Enterprise Linux 7 でのみ利用可能な rh-maven35 Software Collection は、ソフトウェアのプロジェクト管理と理解を深めるツールを提供します。プロジェクトオブジェクトモデル (POM) の概念に基づく **Maven** は、一元的な情報をもとにプロジェクトのビルド、レポート、およびドキュメントを管理できます。

rh-maven35 Collection をインストールするには、**root**で次のコマンドを入力します。

```
yum install rh-maven35
```

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-maven35 bash
```

リモートリポジトリやミラーなど、Maven のグローバル設定は、**/opt/rh/rh-maven35/root/etc/maven/settings.xml** ファイルを編集することで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。プラグインの使用方法については、[本セクション](#) 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確認するには、[プラグインのインデックス](#) を参照してください。

4.6. パッセンジャー

rh-passenger40 Software Collection では、高速、堅牢、軽量に設計された Web/アプリケーションサーバーである **Phusion Passenger** を提供しています。

rh-passenger40 コレクションは、特に ruby193、ruby200、rh-ruby22 Software Collection、および ror40 または rh-ror41 コレクションを使用した **Ruby on Rails** など、複数のバージョンの **Ruby** をサポートしています。**Passenger** をいずれかの Ruby Software Collections と一緒に使用する前に、rh-passenger40 Collection から対応するパッケージ (rh-passenger-ruby193、または rh-passenger-ruby200、rh-passenger-ruby22 パッケージ) をインストールしてください。

rh-passenger40 Software Collection は、httpd24 Software Collection の **Apache httpd** と一緒に使うこともできます。そのためには、rh-passenger40-mod_passenger パッケージをインストールしてください。**Apache httpd** の設定例として、デフォルトの設定ファイル **/opt/rh/httpd24/root/etc/httpd/conf.d/passenger.conf** を参照すると、1つの **Apache httpd** インスタンスで複数の **Ruby** バージョンを使用する方法が示されています。

さらに、rh-passenger40 Software Collection は、nginx16 Software Collection のウェブサーバー **nginx 1.6** と組み合わせて使用することができます。rh-passenger40 で **nginx 1.6** を使用するには、Web アプリケーションのディレクトリーで次のコマンドを使用して **Passenger** をスタンドアロンモードで実行します。

```
scl enable nginx16 rh-passenger40 'passenger start'
```

または、アップストリームの [Passenger documentation](#) に記載されているように、nginx16 設定ファイルを編集してください。

4.7. データベースコネクター

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
rh-nodejs4	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
rh-nodejs8	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
rh-perl520	はい	いいえ	はい	はい	いいえ
rh-perl524	はい	いいえ	はい	はい	いいえ
rh-perl526	はい	いいえ	はい	はい	いいえ
rh-php56	はい	はい	はい	はい	いいえ
rh-php70	はい	いいえ	はい	はい	いいえ
rh-php71	はい	いいえ	はい	はい	いいえ
rh-php72	はい	いいえ	はい	はい	いいえ
python27	はい	はい	はい	はい	いいえ
rh-python34	いいえ	はい	いいえ	はい	いいえ
rh-python35	はい	はい	はい	はい	いいえ
rh-python36	はい	はい	はい	はい	いいえ
rh-ror41	はい	はい	はい	はい	いいえ
rh-ror42	はい	はい	はい	はい	いいえ

言語 (Software Collection)	Database				
	MariaDB	MongoDB	MySQL	PostgreSQL	Redis
rh-ror50	はい	はい	はい	はい	いいえ
rh-ruby25	はい	はい	はい	はい	いいえ

第5章 マイグレーション

この章では、Red Hat Software Collections 3.2 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 10.2 への移行

Red Hat Enterprise Linux 6 には、デフォルトのMySQL実装としてMySQL 5.1が含まれています。Red Hat Enterprise Linux 7 には、デフォルトのMySQL実装としてMariaDB 5.5 が搭載されています。MariaDBは、コミュニティによって開発された、MySQLを置き換えるドロップイン方式のソフトウェアです。MariaDB 10.1はRed Hat Software Collections 2.2 から Software Collection として提供されており、Red Hat Software Collections 3.2 はMariaDB 10.2 とともに配布されています。

Red Hat Enterprise Linux 6 と Red Hat Enterprise Linux 7 の両方で利用可能なrh-mariadb102 Software Collection は、コアシステムのmysql またはmariadb パッケージと競合しないため、rh-mariadb102 Software Collection をmysql またはmariadb パッケージと一緒にインストールすることが可能です。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-mariadb101 Collection がインストールされ、実行中でも、rh-mariadb102 Software Collection をインストールすることもできます。

なお、MariaDB 5.5 またはMariaDB 10.0 を使用している場合は、[Red Hat Software Collections 2.4 Release Notes](#) に記載されているrh-mariadb101 Software Collection に、先にアップグレードする必要があります。

MariaDB 10.2 の詳細については、[changes in version 10.2](#) や[upgrading](#) についてのアップストリームドキュメントをご覧ください。



注記

rh-mariadb102 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-mariadb101 および rh-mariadb102 Software Collections 間の主な相違点

MariaDB 10.2 における主な変更は、[Red Hat Software Collections 3.0 Release Notes](#) を参照してください。

MariaDB 10.2 以降、**SQL_MODE** 変数の動作が変更されました。詳細は[upstream documentation](#) を参照してください。

複数のオプションがデフォルト値を変更しているか、非推奨または削除されました。詳細は、ナレッジベースの記事[Migrating from MariaDB 10.1 to the MariaDB 10.2 Software Collection](#) を参照してください。

rh-mariadb102 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-mariadb102-syspaths パッケージが含まれます。rh-mariadb102*-syspaths パッケージをインストールした後、rh-mariadb102* パッケージで提供されるバイナリーやスクリプトが正しく機能するために、ユーザーが**scl enable** コマンドを使用する必要があります。*-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。

5.1.2. rh-mariadb101 から rh-mariadb102 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-mariadb101 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-mariadb101 サーバーを停止します。

```
service rh-mariadb101-mariadb stop
```

2. rh-mariadb102 Software Collection をインストールします。

```
yum install rh-mariadb102-mariadb-server
```

これらの Collections が競合しないため、rh-mariadb102 Software Collection のインストール中は rh-mariadb101 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-mariadb102/my.cnf` ファイルと `/etc/opt/rh/rh-mariadb102/my.cnf.d/` ディレクトリーに格納されている rh-mariadb102 の設定を検査します。`/etc/opt/rh/rh-mariadb101/my.cnf` および `/etc/opt/rh/rh-mariadb101/my.cnf.d/` に保存されている rh-mariadb101 の設定と比較して、必要に応じて調整してください。
4. rh-mariadb101 Software Collection のすべてのデータは、特に設定されていない限り、`/var/opt/rh/rh-mariadb101/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内容をすべて `/var/opt/rh/rh-mariadb102/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-mariadb102 データベースサーバーを起動します。

```
service rh-mariadb102-mariadb start
```

6. データ移行を実行します。

```
scl enable rh-mariadb102 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-mariadb102 -- mysql_upgrade -p
```

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、コアシステムの `mysql` パッケージ、`mariadb` パッケージ、`rh-mysql*` または `rh-mariadb*` Software Collections と競合しません。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 8.0 には、**MySQL 5.7** からしかアップグレードできない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。**MySQL 5.7** へのアップグレード方法「[MySQL 5.7 への移行](#)」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
```

`caching_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は、rh-mysql80* パッケージが提供するバイナリーおよびスクリプトの正しい機能に `scl enable` コマンドを使用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-mariadb102 Software Collection から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** 節を指定します。

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-mariadb102 Software Collection から対応するパッケージと競合します。

5.3. MYSQL 5.7 への移行

Red Hat Enterprise Linux 6 には、デフォルトの MySQL 実装として MySQL 5.1 が含まれています。Red Hat Enterprise Linux 7 には、デフォルトの MySQL 実装として MariaDB 5.5 が搭載されています。これらの基本バージョンに加えて、MySQL 5.6 が、Red Hat Software Collections 2.0 以降の Red Hat Enterprise Linux 6 および Red Hat Enterprise Linux 7 にて、Software Collection として利用できるようになりました。

Red Hat Enterprise Linux 6 と Red Hat Enterprise Linux 7 の両方で利用可能な rh-mysql57 Software Collection は、コアシステムの mysql または mariadb パッケージや rh-mysql56 Software Collection とは競合しないため、rh-mysql57 Software Collection を mysql、mariadb、rh-mysql56 パッケージと一緒にインストールすることができます。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。

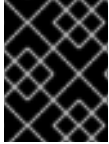
なお、MySQL 5.6 からは MySQL 5.7 にしかアップグレードできません。以前のバージョンからアップグレードする必要がある場合は、まず MySQL 5.6 にアップグレードしてください。MySQL 5.6 へのアップグレード方法は [Red Hat Software Collections 2.2 Release Notes](#) に記載されています。

5.3.1. MySQL 5.6 と MySQL 5.7 の注目すべき相違点

- mysql-bench サブパッケージは、rh-mysql57 Software Collection には含まれていません。
- MySQL 5.7.7 以降、デフォルトの SQL モードには **NO_AUTO_CREATE_USER** が含まれています。**GRANT** ステートメントではデフォルトでユーザーが作成されないため、**CREATE USER** ステートメントを使用して MySQL アカウントを作成する必要があります。詳細は、[upstream documentation](#) を参照してください。

以前のバージョンと比較した MySQL 5.7 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 5.7](#) および [Changes Affecting Upgrades to MySQL 5.7](#) を参照してください。

5.3.2. rh-mysql57 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-mysql57 Software Collection をインストールします。

yum install rh-mysql57-mysql-server

2. `/etc/opt/rh/rh-mysql57/my.cnf` ファイルと `/etc/opt/rh/rh-mysql57/my.cnf.d/` ディレクトリーに格納されている rh-mysql57 の設定を検査します。`/etc/opt/rh/rh-mysql56/my.cnf` および `/etc/opt/rh/rh-mysql56/my.cnf.d/` に格納されている rh-mysql56 の設定と比較して、必要に応じて調整してください。
3. rh-mysql56 データベースサーバーが稼働している場合は停止します。

service rh-mysql56-mysqld stop

4. rh-mysql56 Software Collection のデータはすべて `/var/opt/rh/rh-mysql56/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内容を全て `/var/opt/rh/rh-mysql57/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-mysql57 データベースサーバーを起動します。

service rh-mysql57-mysqld start

6. データ移行を実行します。

scl enable rh-mysql57 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-mysql57 -- mysql_upgrade -p

5.4. MONGODB 3.6 への移行

Red Hat Software Collections 3.2 は、rh-mongodb36 Software Collection から提供される **MongoDB 3.6** でリリースされ、Red Hat Enterprise Linux 7 でのみ利用できます。

rh-mongodb36 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-mongodb36-syspaths パッケージが含まれます。rh-mongodb36*-syspaths パッケージのインストール後に、rh-mongodb36* 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 の詳細は、[Red Hat Software Collections Packaging Guide](#) を参照してください。

5.4.1. MongoDB 3.4 と MongoDB 3.6 の主な相違点

一般的な変更点

rh-mongodb36 Software Collection では、以下のような重要な変更点を加えられています。

- NUMA (Non-Uniform Access Memory) ハードウェアでは、**numactl** コマンドを使用して起動するように **systemd** サービスを設定することができます。[アップストリームの推奨事項](#) を参照してください。**numactl** コマンドで MongoDB を使用するには、numactl RPM パッケージをインストールし、`/etc/opt/rh/rh-mongodb36/sysconfig/mongod` および `/etc/opt/rh/rh-mongodb36/sysconfig/mongos` 設定ファイルを変更する必要があります。

互換性の変更点

MongoDB 3.6 には、さまざまなマイナーな変更が含まれており、MongoDB の以前のバージョンとの互換性に影響を与える可能性があります。

- MongoDB バイナリーはデフォルトで **localhost** にバインドされるため、異なる IP アドレスでリッスンするように明示的に有効にする必要があります。これは、MongoDB Software Collections で配布される **systemd** サービスのデフォルト動作であることに注意してください。
- MONGODB-CR 認証メカニズムが非推奨になりました。3.0 よりも前のバージョンの MongoDB で作成したユーザーの場合は、認証スキーマを **SCRAM** にアップグレードします。
- HTTP インターフェイスおよび REST API が削除される
- レプリカセットの Arbiter の優先順位は **0** に設定します。
- master-slave レプリケーションが非推奨に

MongoDB 3.6 での詳細な互換性の変更点は、[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。

後方互換性のない機能

以下の MongoDB 3.6 機能は後方互換性がなく、**featureCompatibilityVersion** コマンドを使用してバージョンを 3.6 に設定する必要があります。

- コレクションの UUID
- **\$jsonSchema** ドキュメント検証
- ストリームの変更
- チャンク対応のセカンダリー
- バージョン 3.6 のクエリー機能を使用する定義、ドキュメントバリデーター、部分インデックスフィルターを表示する
- セッションおよび再試行可能な書き込み
- **authenticationRestrictions** のあるユーザーおよびロール

MongoDB 3.6 での後方互換性のない変更の詳細は、[アップストリームのリリースノート](#) を参照してください。

5.4.2. rh-mongodb34 から rh-mongodb36 Software Collection へのアップグレード



重要

rh-mongodb34 から rh-mongodb36 Software Collection に移行する前に、デフォルトで `/var/opt/rh/rh-mongodb34/lib/mongodb/` ディレクトリーに保存される MongoDB データベースを含むすべてのデータをバックアップします。また、[互換性の変更点](#) を参照して、アプリケーションおよびデプロイメントが MongoDB 3.6 と互換性があることを確認してください。

rh-mongodb36 Software Collection にアップグレードするには、以下の手順を実行します。

1. アップグレードできるようにするには、rh-mongodb34 インスタンスの **featureCompatibilityVersion** が 3.4 に設定されている必要があります。 **featureCompatibilityVersion** を確認します。

```
~]$ scli enable rh-mongodb34 'mongo --host localhost --port 27017 admin' --eval
'db.adminCommand({getParameter: 1, featureCompatibilityVersion: 1})'
```

mongod サーバーがアクセス制御が有効な状態で設定されている場合は、**--username** および **-password** オプションを **mongo** コマンドに追加します。

2. rh-mongodb36 Software Collections から MongoDB サーバーおよびシェルをインストールします。

```
~]# yum install rh-mongodb36
```

3. MongoDB 3.4 サーバーを停止します。

```
~]# systemctl stop rh-mongodb34-mongod.service
```

4. データを新しい場所にコピーします。

```
~]# cp -a /var/opt/rh/rh-mongodb34/lib/mongodb/* /var/opt/rh/rh-
mongodb36/lib/mongodb/
```

5. `/etc/opt/rh/rh-mongodb36/mongod.conf` ファイルに **rh-mongodb36-mongod** デーモンを設定します。

6. MongoDB 3.6 サーバーを起動します。

```
~]# systemctl start rh-mongodb36-mongod.service
```

7. 後方互換性のない機能を有効にします。

```
~]$ scli enable rh-mongodb36 'mongo --host localhost --port 27017 admin' --eval
'db.adminCommand( { setFeatureCompatibilityVersion: "3.6" } )'
```

mongod サーバーがアクセス制御が有効な状態で設定されている場合は、**--username** および **-password** オプションを **mongo** コマンドに追加します。



注記

アップグレード後、ダウングレードの可能性を最小限に抑えるために、焼き付き期間の間、下位互換性のない機能を有効にせずに、最初にデプロイメントを実行することが推奨されます。

アップグレードの詳細は、[アップストリームのリリースノート](#) を参照してください。

レプリカセットのアップグレードに関する詳細は、アップストリームの [MongoDB Manual](#) を参照してください。

Sharded Cluster のアップグレードに関する詳細は、アップストリームの [MongoDB Manual](#) を参照してください。

5.5. MONGODB 3.4 への移行

Red Hat Enterprise Linux 6 および Red Hat Enterprise Linux 7 で利用可能な rh-mongodb34 Software Collection は MongoDB 3.4 を提供します。

5.5.1. MongoDB 3.2 と MongoDB 3.4 との間の主な相違点

一般的な変更点

rh-mongodb34 Software Collection では、一般的な変更点を加えられています。ナレッジベースの記事 [Migrating from MongoDB 3.2 to MongoDB 3.4](#) に主な変更点が記載されています。詳細な変更については、[upstream release notes](#) を参照してください。

また、この Software Collection には rh-mongodb34-syspaths パッケージが含まれており、バイナリ、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がインストールされます。rh-mongodb34*-syspaths パッケージのインストール後に、rh-mongodb34* パッケージによって提供されるバイナリ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 の詳細は、[Red Hat Software Collections Packaging Guide](#) を参照してください。

互換性の変更点

MongoDB 3.4 には様々なマイナーチェンジが含まれており、以前のバージョンの MongoDB との互換性に影響を与える可能性があります。詳細は、ナレッジベースの記事 [Migrating from MongoDB 3.2 to MongoDB 3.4](#) および [upstream documentation](#) を参照してください。

注目すべきは、以下の MongoDB 3.4 の機能は後方互換性がないため、**featureCompatibilityVersion** コマンドでバージョンを **3.4** に設定する必要があります。

- 既存のコレクションまたはその他のビューから読み取り専用ビューを作成するためのサポート
- インデックスバージョン **v: 2** によって、照合、10 進数データ、大文字と小文字を区別しないインデックスのサポートが追加されます。
- 新しい **decimal** データ型による **decimal128** フォーマットのサポート。

MongoDB 3.4 での後方互換性のない変更点の詳細については、[upstream release notes](#) を参照してください。

5.5.2. rh-mongodb32 から rh-mongodb34 Software Collection へのアップグレード

なお、MongoDB 3.4 にアップグレードして新機能を使い始めたら、3.2.7 以前のバージョンにダウングレードすることはできません。バージョン 3.2.8 以降にダウングレードできます。



重要

rh-mongodb32 から rh-mongodb34 Software Collection に移行する前に、デフォルトで `/var/opt/rh/rh-mongodb32/lib/mongodb/` ディレクトリーに保存されている MongoDB データベースを含め、すべてのデータをバックアップしてください。また、互換性の変更点を参照して、お使いのアプリケーションやデプロイメントが MongoDB 3.4 と互換性があることを確認してください。

rh-mongodb34 Software Collection にアップグレードするには、以下の手順を実行します。

1. rh-mongodb34 Software Collections から MongoDB サーバーとシェルをインストールします。

```
~]# yum install rh-mongodb34
```

2. MongoDB 3.2 サーバーを停止します。

```
~]# systemctl stop rh-mongodb32-mongod.service
```

Red Hat Enterprise Linux 6 システムで `service rh-mongodb32-mongod stop` コマンドを使用します。

3. データを新しい場所にコピーします。

```
~]# cp -a /var/opt/rh/rh-mongodb32/lib/mongodb/* /var/opt/rh/rh-mongodb34/lib/mongodb/
```

4. `/etc/opt/rh/rh-mongodb34/mongod.conf` ファイルで `rh-mongodb34-mongod` デーモンを設定します。
5. MongoDB 3.4 のサーバーを起動します。

```
~]# systemctl start rh-mongodb34-mongod.service
```

Red Hat Enterprise Linux 6 では、代わりに `service rh-mongodb34-mongod start` コマンドを使用します。

6. 後方互換性機能を有効にします。

```
~]$ scl enable rh-mongodb34 'mongo --host localhost --port 27017 admin' --eval 'db.adminCommand( { setFeatureCompatibilityVersion: "3.4" } )'
```

`mongod` サーバーでアクセスコントロールが有効に設定されている場合は、`mongo` コマンドに `-username` と `--password` のオプションを追加します。

アップグレード後に、この機能を最初に有効にせずにデプロイメントを実行することが推奨されます。

アップグレードの詳細は、[アップストリームのリリースノート](#) を参照してください。

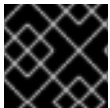
レプリカセットのアップグレードに関する詳細は、アップストリームの [MongoDB Manual](#) を参照してください。

Sharded Cluster のアップグレードに関する詳細は、アップストリームの [MongoDB マニュアル](#) を参照してください。

5.6. POSTGRESQL 10 への移行

Red Hat Software Collections 3.2 には **PostgreSQL 10** が同梱されており、Red Hat Enterprise Linux 7 でのみ利用できます。rh-postgresql10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。以前のバージョンに移行する方法、または Red Hat Enterprise Linux 6 を使用する場合は、「[PostgreSQL 9.6 への移行](#)」を参照してください。

rh-postgresql10 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-postgresql10-syspaths パッケージが含まれます。rh-postgresql10*-syspaths パッケージをインストールした後、rh-postgresql10* パッケージで提供されるバイナリーやスクリプトが正しく機能するために、ユーザーが **scl enable** コマンドを使用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 Packaging Guide](#) を参照してください。



重要

PostgreSQL 10 に移行する前に、[upstream compatibility notes](#) 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-postgresql96 および rh-postgresql10 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-postgresql96	rh-postgresql10
実行ファイル	/usr/bin/	/opt/rh/rh-postgresql96/root/usr/bin/ /	/opt/rh/rh-postgresql10/root/usr/bin/ /
ライブラリー	/usr/lib64/	/opt/rh/rh-postgresql96/root/usr/lib64/	/opt/rh/rh-postgresql10/root/usr/lib64/
ドキュメント	/usr/share/doc/postgresql/html/	/opt/rh/rh-postgresql96/root/usr/share/doc/postgresql/html/	/opt/rh/rh-postgresql10/root/usr/share/doc/postgresql/html/
PDF ドキュメント	/usr/share/doc/postgresql-docs/	/opt/rh/rh-postgresql96/root/usr/share/doc/postgresql-docs/	/opt/rh/rh-postgresql10/root/usr/share/doc/postgresql-docs/
Contrib ドキュメント	/usr/share/doc/postgresql-contrib/	/opt/rh/rh-postgresql96/root/usr/share/doc/postgresql-contrib/	/opt/rh/rh-postgresql10/root/usr/share/doc/postgresql-contrib/

コンテンツ	postgresql	rh-postgresql96	rh-postgresql10
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-postgresql96/lib/pgsql/data/	/var/opt/rh/rh-postgresql10/lib/pgsql/data/
バックアップエリア	/var/lib/pgsql/backups/	/var/opt/rh/rh-postgresql96/lib/pgsql/backups/	/var/opt/rh/rh-postgresql10/lib/pgsql/backups/
テンプレート	/usr/share/pgsql/	/opt/rh/rh-postgresql96/root/usr/share/pgsql/	/opt/rh/rh-postgresql10/root/usr/share/pgsql/
手順言語	/usr/lib64/pgsql/	/opt/rh/rh-postgresql96/root/usr/lib64/pgsql/	/opt/rh/rh-postgresql10/root/usr/lib64/pgsql/
開発ヘッダー	/usr/include/pgsql/	/opt/rh/rh-postgresql96/root/usr/include/pgsql/	/opt/rh/rh-postgresql10/root/usr/include/pgsql/
他の共有データ	/usr/share/pgsql/	/opt/rh/rh-postgresql96/root/usr/share/pgsql/	/opt/rh/rh-postgresql10/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-postgresql96/root/usr/lib64/pgsql/test/regress/ (-test パッケージ内)	/opt/rh/rh-postgresql10/root/usr/lib64/pgsql/test/regress/ (-test パッケージ内)

5.6.1. Red Hat Enterprise Linux システムバージョンの PostgreSQL から PostgreSQL 10 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。Red Hat Enterprise Linux システムバージョンの **PostgreSQL** から **rh-postgresql10 Software Collection** にデータを移行するには、**pg_upgrade** ツールを使用して高速アップグレードを行う (推奨) か、データベースデータを SQL コマンドで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する方法があります。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 documentation](#) 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 10 にデータを移行する前に、デフォルトで **/var/lib/pgsql/data/** ディレクトリーにある 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
```

2. 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。

3. 新しいデータディレクトリー `/var/opt/rh/rh-postgresql10/lib/pgsql/data/` が存在しないことを確認します。

```
file /var/opt/rh/rh-postgresql10/lib/pgsql/data/
```

PostgreSQL 10 を新規にインストールした場合、このディレクトリーはシステム内に存在しないはずですが、存在する場合は、**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-postgresql10/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-postgresql10 -- postgresql-setup --upgrade
```

または、`/opt/rh/rh-postgresql10/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 からのアップグレードには `--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一覧は、`--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-postgresql10-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-postgresql10-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-postgresql10 ~/analyze_new_cluster.sh'
```

6. 必要に応じて、システムの起動時に PostgreSQL 10 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 10 サーバーを有効にするには、**root**で入力します。

chkconfig rh-postgresql10-postgresql on

7. 設定がデフォルトと異なる場合は、設定ファイル (特に `/var/opt/rh/rh-postgresql10/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-postgresql10-postgresql -- postgresql-setup --initdb

5. **root** で新しいサーバーを起動します。

systemctl start rh-postgresql10-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-postgresql10 "psql -f ~/pgdump_file.sql postgres"'

7. 必要に応じて、システムの起動時に PostgreSQL 10 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 10 サーバーを有効にするには、**root**で入力します。

chkconfig rh-postgresql10-postgresql on

- 設定がデフォルトと異なる場合は、設定ファイル (特に `/var/opt/rh/rh-postgresql10/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。

5.6.2. PostgreSQL 9.6 Software Collection から PostgreSQL 10 Software Collection への移行

rh-postgresql96 Software Collection から rh-postgresql10 Collection にデータを移行するには、`pg_upgrade` ツールを使って高速アップグレードを行うか (推奨)、データベースのデータを SQL コマンドで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する方法があります。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 documentation](#) を参照してください。



重要

PostgreSQL 9.6 から PostgreSQL 10 にデータを移行する前に、デフォルトでは `/var/opt/rh/rh-postgresql96/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め、すべてのデータをバックアップしてください。

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。

- 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-postgresql96-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-postgresql96-postgresql.service
```

- 古いディレクトリー `/var/opt/rh/rh-postgresql96/lib/pgsql/data/` が存在することを確認します。

```
file /var/opt/rh/rh-postgresql96/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-postgresql10/lib/pgsql/data/` が存在しないことを確認します。

```
file /var/opt/rh/rh-postgresql10/lib/pgsql/data/
```

PostgreSQL 10 を新規にインストールした場合、このディレクトリーはシステム内に存在しないはずですが。存在する場合は、`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-postgresql10/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-postgresql10 -- postgresql-setup --upgrade --upgrade-from=rh-postgresql96-postgresql
```

■

また、`/opt/rh/rh-postgresql10/root/usr/bin/postgresql-setup --upgrade --upgrade-from=rh-postgresql96-postgresql` コマンドを使用することもできます。

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一覧は、`--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-postgresql10-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-postgresql10-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-postgresql10 ~/analyze_new_cluster.sh'
```

6. 必要に応じて、システムの起動時に PostgreSQL 10 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 9.6 サーバーを無効にするには、**root**で次のコマンドを入力します。

```
chkconfig rh-postgresql96-postgresql off
```

PostgreSQL 10 サーバーを有効にするには、**root**で入力します。

```
chkconfig rh-postgresql10-postgresql on
```

7. 設定がデフォルトと異なる場合は、設定ファイル (特に `/var/opt/rh/rh-postgresql10/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-postgresql96-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-postgresql96 "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-postgresql96-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-postgresql10-postgresql -- postgresql-setup --initdb
```

5. **root** で新しいサーバーを起動します。

```
systemctl start rh-postgresql10-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-postgresql10 "psql -f ~/pgdump_file.sql postgres"'
```

7. 必要に応じて、システムの起動時に PostgreSQL 10 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 9.6 サーバーを無効にするには、**root**で次のコマンドを入力します。

```
chkconfig rh-postgresql96-postgresql off
```

PostgreSQL 10 サーバーを有効にするには、**root**で入力します。

```
chkconfig rh-postgresql10-postgresql on
```

8. 設定がデフォルトと異なる場合は、設定ファイル (特に `/var/opt/rh/rh-postgresql10/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。

5.7. POSTGRESQL 9.6 への移行

PostgreSQL 9.6 は、Red Hat Enterprise Linux 6 と Red Hat Enterprise Linux 7 の両方で利用できます。また、Red Hat Enterprise Linux 6 の PostgreSQL 8.4、Red Hat Enterprise Linux 7 の PostgreSQL 9.2、または以前のバージョンの Red Hat Software Collections でリリースされた任意のバージョンの PostgreSQL 9.2 と同時に、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。

5.7.1. PostgreSQL 9.5 および PostgreSQL 9.6 間の主な違い

PostgreSQL 9.5 と PostgreSQL 9.6 の間の最も注目すべき変更点は、[upstream release notes](#) に記載されています。

rh-postgresql96 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-postgresql96-syspaths パッケージが含まれます。rh-postgresql96*-syspaths パッケージをインストールした後、rh-postgresql96* パッケージで提供されるバイナリーやスクリプトが正しく機能するために、ユーザーが **scl enable** コマンドを使用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 Packaging Guide](#) を参照してください。

次の表は、Red Hat Enterprise Linux システムバージョンの PostgreSQL (postgresql) と postgresql92、rh-postgresql95、rh-postgresql96 Software Collections の異なるパスの概要を示しています。Red Hat Enterprise Linux 6 で配布されている PostgreSQL 8.4 と Red Hat Enterprise Linux 7 で出荷されているシステムバージョンの PostgreSQL 9.2 のパスは同じであることに注意してください。rh-postgresql94 Software Collection のパスは rh-postgresql95 に類似しています。

表5.2 PostgreSQL パスの相違点

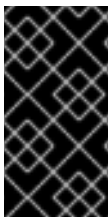
コンテンツ	postgresql	postgresql92	rh-postgresql95	rh-postgresql96
実行ファイル	/usr/bin/	/opt/rh/postgresql92 /root/usr/bin/	/opt/rh/rh- postgresql95/root/us r/bin/	/opt/rh/rh- postgresql96/root/us r/bin/
ライブラリー	/usr/lib64/	/opt/rh/postgresql92 /root/usr/lib64/	/opt/rh/rh- postgresql95/root/us r/lib64/	/opt/rh/rh- postgresql96/root/us r/lib64/
資料	/usr/share/doc /postgresql/ht ml/	/opt/rh/postgresql92 /root/usr/share/doc/ postgresql/html/	/opt/rh/rh- postgresql95/root/us r/share/doc/postgres ql/html/	/opt/rh/rh- postgresql96/root/us r/share/doc/postgres ql/html/
PDF ドキュメント	/usr/share/doc /postgresql- docs/	/opt/rh/postgresql92 /root/usr/share/doc/ postgresql-docs/	/opt/rh/rh- postgresql95/root/us r/share/doc/postgres ql-docs/	/opt/rh/rh- postgresql96/root/us r/share/doc/postgres ql-docs/
Contrib ドキュメント	/usr/share/doc /postgresql- contrib/	/opt/rh/postgresql92 /root/usr/share/doc/ postgresql-contrib/	/opt/rh/rh- postgresql95/root/us r/share/doc/postgres ql-contrib/	/opt/rh/rh- postgresql96/root/us r/share/doc/postgres ql-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	未インストール
データ	/var/lib/pgsql/ data/	/opt/rh/postgresql92 /root/var/lib/pgsql/d ata/	/var/opt/rh/rh- postgresql95/lib/pgs ql/data/	/var/opt/rh/rh- postgresql96/lib/pgs ql/data/
バックアップエリア	/var/lib/pgsql/ backups/	/opt/rh/postgresql92 /root/var/lib/pgsql/b ackups/	/var/opt/rh/rh- postgresql95/lib/pgs ql/backups/	/var/opt/rh/rh- postgresql96/lib/pgs ql/backups/
テンプレート	/usr/share/pgs ql/	/opt/rh/postgresql92 /root/usr/share/pgs ql/ /	/opt/rh/rh- postgresql95/root/us r/share/pgs ql/	/opt/rh/rh- postgresql96/root/us r/share/pgs ql/
手順言語	/usr/lib64/pgs ql/	/opt/rh/postgresql92 /root/usr/lib64/pgs ql/ /	/opt/rh/rh- postgresql95/root/us r/lib64/pgs ql/	/opt/rh/rh- postgresql96/root/us r/lib64/pgs ql/
開発ヘッダー	/usr/include/p gs ql/	/opt/rh/postgresql92 /root/usr/include/pgs ql/	/opt/rh/rh- postgresql95/root/us r/include/pgs ql/	/opt/rh/rh- postgresql96/root/us r/include/pgs ql/
他の共有データ	/usr/share/pgs ql/	/opt/rh/postgresql92 /root/usr/share/pgs ql/ /	/opt/rh/rh- postgresql95/root/us r/share/pgs ql/	/opt/rh/rh- postgresql96/root/us r/share/pgs ql/

コンテンツ	postgresql	postgresql92	rh-postgresql95	rh-postgresql96
リグレーションテスト	/usr/lib64/pgsql/test/regress / (-test パッケージ内)	/opt/rh/postgresql92 /root/usr/lib64/pgsql/test/regress/ (-test パッケージ内)	/opt/rh/rh-postgresql95/root/usr/lib64/pgsql/test/regress/ (-test パッケージ内)	/opt/rh/rh-postgresql96/root/usr/lib64/pgsql/test/regress/ (-test パッケージ内)

PostgreSQL 8.4 と PostgreSQL 9.2 との間の変更点については、[Red Hat Software Collections 1.2 Release Notes](#) を参照してください。PostgreSQL 9.2 と PostgreSQL 9.4 の間の重要な変更点は、[Red Hat Software Collections 2.0 Release Notes](#) に記載されています。PostgreSQL 9.4 と PostgreSQL 9.5 の違いについては、[Red Hat Software Collections 2.2 Release Notes](#) を参照してください。

5.7.2. Red Hat Enterprise Linux システムバージョンの PostgreSQL から PostgreSQL 9.6 Software Collection への移行

Red Hat Enterprise Linux 6 には PostgreSQL 8.4 が、Red Hat Enterprise Linux 7 には PostgreSQL 9.2 が配布されています。Red Hat Enterprise Linux システムバージョンの PostgreSQL から rh-postgresql96 Software Collection にデータを移行するには、**pg_upgrade** ツールを使用して高速アップグレードを行う (推奨) か、データベースデータを SQL コマンドで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する方法があります。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。以下の手順は、Red Hat Enterprise Linux 6 と Red Hat Enterprise Linux 7 システムバージョンの PostgreSQL に適用されます。



重要

Red Hat Enterprise Linux システムバージョンの PostgreSQL から PostgreSQL 9.6 にデータを移行する前に、**デフォルトで /var/lib/pgsql/data/** ディレクトリーにある 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5.5 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。

```
service postgresql stop
```

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。

```
service postgresql status
```

2. 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。

3. 新しいデータディレクトリー `/var/opt/rh/rh-postgresql96/lib/pgsql/data/` が存在していないことを確認します。

```
file /var/opt/rh/rh-postgresql96/lib/pgsql/data/
```

PostgreSQL 9.6 を新規にインストールした場合、このディレクトリーはシステム内に存在しないはずですが、存在する場合は、**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-postgresql96/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-postgresql96 -- postgresql-setup --upgrade
```

また、`/opt/rh/rh-postgresql96/root/usr/bin/postgresql-setup --upgrade` コマンドを使用することもできます。

別のバージョンの PostgreSQL からのアップグレードには `--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一覧は、`--upgrade-ids` オプションを使用して利用できます。

アップグレード中に何か問題が発生したかどうかを確認するために、出来上がった `/var/lib/pgsql/upgrade_rh-postgresql96-postgresql.log` ログファイルを読むことをお勧めします。

5. **root** で新しいサーバーを起動します。

```
service rh-postgresql96-postgresql start
```

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-postgresql96 ~/analyze_new_cluster.sh'
```

6. 必要に応じて、システムの起動時に PostgreSQL 9.6 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 9.6 サーバーを有効にするには、**root** で入力します。

```
chkconfig rh-postgresql96-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-postgresql96/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5.6 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。

```
service postgresql start
```

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。

```
service postgresql stop
```

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-postgresql96-postgresql -- postgresql-setup --initdb
```

5. **root** で新しいサーバーを起動します。

```
service rh-postgresql96-postgresql start
```

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-postgresql96 "psql -f ~/pgdump_file.sql postgres"'
```

7. 必要に応じて、システムの起動時に PostgreSQL 9.6 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 9.6 サーバーを有効にするには、**root**で入力します。

```
chkconfig rh-postgresql96-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-postgresql96/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。

5.7.3. PostgreSQL 9.5 Software Collection から PostgreSQL 9.6 Software Collection への移行

rh-postgresql95 Software Collection から rh-postgresql96 Collection にデータを移行するには、**pg_upgrade** ツールを使って高速アップグレードを行うか (推奨)、データベースのデータを SQL コマンドで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する方法があります。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

PostgreSQL 9.5 から PostgreSQL 9.6 にデータを移行する前に、デフォルトでは `/var/opt/rh/rh-postgresql95/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め、すべてのデータをバックアップしてください。

手順5.7 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。

```
service rh-postgresql95-postgresql stop
```

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。

```
service rh-postgresql95-postgresql status
```

2. 古いディレクトリー `/var/opt/rh/postgresql95/lib/pgsql/data/` が存在することを確認します。

```
file /var/opt/rh/rh-postgresql95/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-postgresql96/lib/pgsql/data/` が存在していないことを確認します。

```
file /var/opt/rh/rh-postgresql96/lib/pgsql/data/
```

PostgreSQL 9.6 を新規にインストールした場合、このディレクトリーはシステム内に存在しないはずですが、存在する場合は、**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-postgresql96/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-postgresql96 -- postgresql-setup --upgrade --upgrade-from=rh-postgresql95-postgresql
```

また、`/opt/rh/rh-postgresql96/root/usr/bin/postgresql-setup --upgrade --upgrade-from=rh-postgresql95-postgresql` コマンドを使用することもできます。

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一覧は、`--upgrade-ids` オプションを使用して利用できます。

アップグレード中に何か問題が発生したかどうかを確認するために、出来上がった `/var/lib/pgsql/upgrade_rh-postgresql96-postgresql.log` ログファイルを読むことをお勧めします。

5. **root** で新しいサーバーを起動します。

```
service rh-postgresql96-postgresql start
```

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-postgresql96 ~/analyze_new_cluster.sh'
```

- 必要に応じて、システムの起動時に PostgreSQL 9.6 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 9.5 サーバーを無効にするには、**root**で次のコマンドを入力します。

```
chkconfig rh-postgresql95-postgresql off
```

PostgreSQL 9.6 サーバーを有効にするには、**root**で入力します。

```
chkconfig rh-postgresql96-postgresql on
```

- 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-postgresql96/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5.8 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。

- 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。

```
service rh-postgresql95-postgresql start
```

- 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-postgresql95 "pg_dumpall > ~/pgdump_file.sql"'
```

- 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。

```
service rh-postgresql95-postgresql stop
```

- 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-postgresql96-postgresql -- postgresql-setup --initdb
```

- root** で新しいサーバーを起動します。

```
service rh-postgresql96-postgresql start
```

- 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-postgresql96 "psql -f ~/pgdump_file.sql postgres"'
```

- 必要に応じて、システムの起動時に PostgreSQL 9.6 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 9.5 サーバーを無効にするには、**root**で次のコマンドを入力します。

```
chkconfig rh-postgresql95-postgresql off
```

PostgreSQL 9.6 サーバーを有効にするには、**root**で入力します。

```
chkconfig rh-postgresql96-postgresql on
```

- 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-postgresql96/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。

postgresql92 Software Collection から移行する必要がある場合は、[Red Hat Software Collections 2.0 Release Notes](#) を参照してください。この手順は同じですが、新しい Collection のバージョンを調整する必要があります。これは、[Red Hat Software Collections 2.2 Release Notes](#) で説明したように、rh-postgresql94 Software Collection からの移行を適用します。

5.8. NGINX 1.14 への移行について

rh-nginx114 Software Collection のルートディレクトリーは、`/opt/rh/rh-nginx114/root/`にあります。エラーログは、デフォルトでは `/var/opt/rh/rh-nginx114/log/nginx` に保存されます。

設定ファイルは、`/etc/opt/rh/rh-nginx114/nginx/`ディレクトリーに格納されます。nginx 1.14 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-nginx114/nginx/default.d/`ディレクトリーにある設定ファイル (拡張子が `.conf`) は、80 番ポートのデフォルトのサーバブロック設定に含まれます。



重要

nginx 1.12 から nginx 1.14 にアップグレードする前に、`/opt/rh/nginx112/root/`ツリーにある Web ページや、`/etc/opt/rh/nginx112/nginx/`ツリーにある設定ファイルなど、すべてのデータをバックアップしてください。

設定ファイルの変更や Web アプリケーションの設定など、特定の変更を `/opt/rh/nginx112/root/`ツリーで行った場合は、その変更を新しい `/opt/rh/rh-nginx114/root/`と `/etc/opt/rh/rh-nginx114/nginx/`ディレクトリーにも反映させてください。

この手順では、nginx 1.8、nginx 1.10、または nginx 1.12 から nginx 1.14 へ直接アップグレードすることができます。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。

第6章 関連情報

この章では、Red Hat Software Collections 3.2 および Red Hat Enterprise Linux に関するその他の関連情報源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 3.2 Packaging Guide](#) : Red Hat Software Collections の『Packaging Guide』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 8.0 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 8.0 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 Manager の使用および設定](#) : 『Red Hat Subscription Manager の使用および設定』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、登録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 6 Deployment Guide](#) : Red Hat Enterprise Linux 6 の『Deployment Guide』では、Red Hat Enterprise Linux 6 のデプロイメント、設定、および管理に関する関連情報を提供します。
- [Red Hat Enterprise Linux 7 System Administrator's Guide](#) : The 『System Administrator's Guide』 for Red Hat Enterprise Linux 7 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ト

ターが含まれます。

付録A 改訂履歴

改訂 3.2-6 「データベースコネクタ」を更新	Fri Nov 12 2021	Lenka Špačková
改訂 3.2-5 誤字が修正されました。	Wed May 01 2019	Lenka Špačková
改訂 3.2-4 RHEA-2019:0543 で利用可能な新しい rh-jmc Software Collection に関する情報が追加されました。	Wed Mar 13 2019	Lenka Špačková
改訂 3.2-3 EOL コンポーネントを更新。	Thu Dec 20 2018	Lenka Špačková
改訂 3.2-2 PostgreSQL の移行の章で、コマンドの構文を修正しました。	Tue Dec 18 2018	Lenka Špačková
改訂 3.2-1 Red Hat Software Collections 3.2 リリースノートの公開。	Tue Nov 13 2018	Lenka Špačková
改訂 3.2-0 Red Hat Software Collections 3.2 Beta リリースノートの公開。	Tue Oct 23 2018	Lenka Špačková