Red Hat Software Collections 3

3.7 Release Notes

Release Notes for Red Hat Software Collections 3.7

Red Hat Software Collections 3 3.7 Release Notes

Release Notes for Red Hat Software Collections 3.7

Lenka Špačková
Red Hat Customer Content Services
lspackova@redhat.com

Jaromír Hradílek
Red Hat Customer Content Services
jhradilek@redhat.com

Eliška Slobodová
Red Hat Customer Content Services

Legal Notice

Abstract

The Red Hat Software Collections 3.7 Release Notes document the major features and contain important information about known problems in Red Hat Software Collections 3.7. The Red Hat Developer Toolset collection is documented in the Red Hat Developer Toolset Release Notes and the Red Hat Developer Toolset User Guide .

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message.

CHAPTER 1. RED HAT SOFTWARE COLLECTIONS 3.7

This chapter serves as an overview of the Red Hat Software Collections 3.7 content set. It provides a list of components and their descriptions, sums up changes in this version, documents relevant compatibility information, and lists known issues.

1.1. ABOUT RED HAT SOFTWARE COLLECTIONS

For certain applications, more recent versions of some software components are often needed in order to use their latest new features. **Red Hat Software Collections** is a Red Hat offering that provides a set of dynamic programming languages, database servers, and various related packages that are either more recent than their equivalent versions included in the base Red Hat Enterprise Linux system, or are available for this system for the first time.

Red Hat Software Collections 3.7 is available for Red Hat Enterprise Linux 7. For a complete list of components that are distributed as part of Red Hat Software Collections and a brief summary of their features, see Section 1.2, "Main Features".

Red Hat Software Collections does not replace the default system tools provided with Red Hat Enterprise Linux 7. Instead, a parallel set of tools is installed in the **/opt/** directory and can be optionally enabled per application by the user using the supplied **scl** utility. The default versions of Perl or PostgreSQL, for example, remain those provided by the base Red Hat Enterprise Linux system.

> **NOTE**
>
> In Red Hat Enterprise Linux 8, similar components are provided as Application Streams.

All Red Hat Software Collections components are fully supported under Red Hat Enterprise Linux Subscription Level Agreements, are functionally complete, and are intended for production use. Important bug fix and security errata are issued to Red Hat Software Collections subscribers in a similar manner to Red Hat Enterprise Linux for at least two years from the release of each major version. In each major release stream, each version of a selected component remains backward compatible. For detailed information about length of support for individual components, refer to the Red Hat Software Collections Product Life Cycle document.

1.1.1. Red Hat Developer Toolset

Red Hat Developer Toolset is a part of Red Hat Software Collections, included as a separate Software Collection. For more information about Red Hat Developer Toolset, refer to the Red Hat Developer Toolset Release Notes and the Red Hat Developer Toolset User Guide .

1.2. MAIN FEATURES

Table 1.1, "Red Hat Software Collections Components" lists components that are supported at the time of the Red Hat Software Collections 3.7 release. All Software Collections are currently supported only on Red Hat Enterprise Linux 7.

Table 1.1. Red Hat Software Collections Components

Component	Software Collection	Description
Red Hat Developer Toolset 10.1	devtoolset-10	Red Hat Developer Toolset is designed for developers working on the Red Hat Enterprise Linux platform. It provides current versions of the **GNU Compiler Collection**, **GNU Debugger**, and other development, debugging, and performance monitoring tools. For a complete list of components, see the Red Hat Developer Toolset Components table in the *Red Hat Developer Toolset User Guide*.
Perl 5.30.1	rh-perl530	A release of Perl, a high-level programming language that is commonly used for system administration utilities and web programming. The rh-perl530 Software Collection provides additional utilities, scripts, and database connectors for **MySQL**, **PostgreSQL**, and **SQLite**. It includes the **DateTime** Perl module and the **mod_perl** Apache httpd module, which is supported only with the httpd24 Software Collection. Additionally, it provides the **cpanm** utility for easy installation of CPAN modules, the **LWP::UserAgent** module for communicating with the HTTP servers, and the **LWP::Protocol::https** module for securing the communication. The rh-perl530 packaging is aligned with upstream; the perl530-perl package installs also core modules, while the interpreter is provided by the perl-interpreter package.
PHP 7.3.20	rh-php73	A release of PHP 7.3 with PEAR 1.10.9, APCu 5.1.17, and the Xdebug extension.
Python 2.7.18	python27	A release of Python 2.7 with a number of additional utilities. This Python version provides various features and enhancements, including an ordered dictionary type, faster I/O operations, and improved forward compatibility with Python 3. The python27 Software Collections contains the *Python 2.7.13 interpreter*, a set of extension libraries useful for programming web applications and **mod_wsgi** (only supported with the httpd24 Software Collection), MySQL and PostgreSQL database connectors, and **numpy** and **scipy**.
Python 3.8.6	rh-python38	The rh-python38 Software Collection contains Python 3.8, which introduces *new Python modules, such as* **contextvars**, **dataclasses**, *or* **importlib.resources**, *new language features, improved developer experience, and performance improvements*. In addition, a set of popular extension libraries is provided, including **mod_wsgi** (supported only together with the httpd24 Software Collection), **numpy**, **scipy**, and the **psycopg2** PostgreSQL database connector.

Component	Software Collection	Description
Ruby 2.6.7	rh-ruby26	A release of Ruby 2.6. This version provides multiple performance improvements and new features, such as *endless ranges, the **Binding#source_location** method, and the **$SAFE** process global state*. Ruby 2.6.0 maintains source-level backward compatibility with Ruby 2.5.
Ruby 2.7.3	rh-ruby27	A release of Ruby 2.7. This version provides multiple performance improvements and new features, such as *Compaction GC or command-line interface for the LALR(1) parser generator,* and an enhancement to REPL. Ruby 2.7 maintains source-level backward compatibility with Ruby 2.6.
Ruby 3.0.1	rh-ruby30	A release of Ruby 3.0. This version provides multiple performance improvements and new features, such as ***Ractor**, **Fiber Scheduler** and the **RBS** language*. Ruby 3.0 maintains source-level backward compatibility with Ruby 2.7.
MariaDB 10.3.27	rh-mariadb103	A release of MariaDB, an alternative to MySQL for users of Red Hat Enterprise Linux. For all practical purposes, MySQL is binary compatible with MariaDB and can be replaced with it without any data conversions. This version introduces *system-versioned tables, invisible columns, a new instant **ADD COLUMN** operation for **InnoDB***, and a *JDBC connector for MariaDB and MySQL.*
MariaDB 10.5.9	rh-mariadb105	A release of MariaDB, an alternative to MySQL for users of Red Hat Enterprise Linux. For all practical purposes, MySQL is binary compatible with MariaDB and can be replaced with it without any data conversions. This version includes various new features, *MariaDB Galera Cluster upgraded to version 4, and PAM plug-in version 2.0*.
MySQL 8.0.21	rh-mysql80	A release of the MySQL server, which introduces a number of new *security and account management features* and enhancements.
PostgreSQL 10.15	rh-postgresql10	A release of PostgreSQL, which includes a significant performance improvement and a number of new features, such as *logical replication using the **publish** and **subscribe** keywords, or stronger password authentication based on the **SCRAM-SHA-256** mechanism*.
PostgreSQL 12.5	rh-postgresql12	A release of PostgreSQL, which provides *the **pgaudit** extension, various enhancements to partitioning and parallelism, support for the SQL/JSON path language,* and performance improvements.

Component	Software Collection	Description
PostgreSQL 13.2	rh-postgresql13	A release of PostgreSQL, which enables *improved query planning* and introduces various performance improvements and *two new packages, pg_repack and plpython3*.
Node.js 12.21.0	rh-nodejs12	A release of Node.js with *V8 engine version 7.6, support for ES6 modules,* and improved support for native modules.
Node.js 14.16.0	rh-nodejs14	A release of Node.js with *V8 version 8.3,* a new experimental WebAssembly System Interface (WASI), and a new experimental Async Local Storage API.
nginx 1.16.1	rh-nginx116	A release of nginx, a web and proxy server with a focus on high concurrency, performance, and low memory usage. This version introduces numerous *updates related to SSL, several new directives and parameters,* and various enhancements.
nginx 1.18.0	rh-nginx118	A release of nginx, a web and proxy server with a focus on high concurrency, performance, and low memory usage. This version introduces *enhancements to HTTP request rate and connection limiting, and a new* **auth_delay** *directive*. In addition, support for new variables has been added to multiple directives.
Apache httpd 2.4.34	httpd24	A release of the Apache HTTP Server (httpd), including a high performance *event-based processing model, enhanced SSL module and FastCGI support*. The **mod_auth_kerb**, **mod_auth_mellon**, and **ModSecurity** modules are also included.
Varnish Cache 6.0.6	rh-varnish6	A release of Varnish Cache, a high-performance HTTP reverse proxy. This version includes *support for Unix Domain Sockets (both for clients and for back-end servers), new level of the VCL language (* **vcl 4.1** *), and improved HTTP/2 support*.
Maven 3.6.1	rh-maven36	A release of Maven, a software project management and comprehension tool. This release provides various enhancements and bug fixes.

Component	Software Collection	Description
Git 2.27.0	rh-git227	A release of Git, a distributed revision control system with a decentralized architecture. As opposed to centralized version control systems with a client-server model, Git ensures that each working copy of a Git repository is its exact copy with complete revision history. This version introduces numerous enhancements; for example, *the **git checkout** command split into **git switch** and **git restore**, and changed behavior of the **git rebase** command.* In addition, Git Large File Storage (LFS) has been updated to version 2.11.0.
Redis 5.0.5	rh-redis5	A release of Redis 5.0, a *persistent key-value database.* Redis now provides **redis-trib**, *a cluster management tool.*
HAProxy 1.8.24	rh-haproxy18	A release of HAProxy 1.8, a reliable, high-performance *network load balancer* for TCP and HTTP-based applications.
JDK Mission Control 8.0.0	rh-jmc	This Software Collection includes *JDK Mission Control (JMC)*, a powerful profiler for HotSpot JVMs. JMC provides an advanced set of tools for efficient and detailed analysis of extensive data collected by the JDK Flight Recorder. JMC requires JDK version 11 or later to run. Target Java applications must run with at least OpenJDK version 8 so that JMC can access JDK Flight Recorder features. The rh-jmc Software Collection requires the rh-maven36 Software Collection.

Previously released Software Collections remain available in the same distribution channels. All Software Collections, including retired components, are listed in the Table 1.2, "All Available Software Collections". Software Collections that are no longer supported are marked with an asterisk (*****).

See the Red Hat Software Collections Product Life Cycle document for information on the length of support for individual components. For detailed information regarding previously released components, refer to the Release Notes for earlier versions of Red Hat Software Collections.

Table 1.2. All Available Software Collections

Component	Software Collection	Availability	Architectures supported on RHEL7
Components New in Red Hat Software Collections 3.7			
MariaDB 10.5.9	rh-mariadb105	RHEL7	x86_64, s390x, ppc64le
PostgreSQL 13.2	rh-postgresql13	RHEL7	x86_64, s390x, ppc64le
Ruby 3.0.1	rh-ruby30	RHEL7	x86_64, s390x, ppc64le

Components Updated in Red Hat Software Collections 3.7			
Red Hat Developer Toolset 10.1	devtoolset-10	RHEL7	x86_64, s390x, ppc64, ppc64le
JDK Mission Control 8.0.0	rh-jmc	RHEL7	x86_64
Ruby 2.7.3	rh-ruby27	RHEL7	x86_64, s390x, aarch64, ppc64le
Ruby 2.6.7	rh-ruby26	RHEL7	x86_64, s390x, aarch64, ppc64le

Components Last Updated in Red Hat Software Collections 3.6			
Git 2.27.0	rh-git227	RHEL7	x86_64, s390x, ppc64le
nginx 1.18.0	rh-nginx118	RHEL7	x86_64, s390x, ppc64le
Node.js 14.16.0	rh-nodejs14	RHEL7	x86_64, s390x, ppc64le
Apache httpd 2.4.34	httpd24	RHEL7	x86_64, s390x, aarch64, ppc64le
PHP 7.3.20	rh-php73	RHEL7	x86_64, s390x, aarch64, ppc64le
HAProxy 1.8.24	rh-haproxy18	RHEL7	x86_64
Perl 5.30.1	rh-perl530	RHEL7	x86_64, s390x, aarch64, ppc64le
Ruby 2.5.9	rh-ruby25*	RHEL7	x86_64, s390x, aarch64, ppc64le

Components Last Updated in Red Hat Software Collections 3.5			
Red Hat Developer Toolset 9.1	devtoolset-9	RHEL7	x86_64, s390x, aarch64, ppc64, ppc64le
Python 3.8.6	rh-python38	RHEL7	x86_64, s390x, aarch64, ppc64le
Varnish Cache 6.0.6	rh-varnish6	RHEL7	x86_64, s390x, aarch64, ppc64le

Components Last Updated in Red Hat Software Collections 3.5			
Apache httpd 2.4.34 (the last update for RHEL6)	httpd24 (RHEL6)*	RHEL6	x86_64

Components Last Updated in Red Hat Software Collections 3.4			
Node.js 12.21.0	rh-nodejs12	RHEL7	x86_64, s390x, aarch64, ppc64le
nginx 1.16.1	rh-nginx116	RHEL7	x86_64, s390x, aarch64, ppc64le
PostgreSQL 12.5	rh-postgresql12	RHEL7	x86_64, s390x, aarch64, ppc64le
Maven 3.6.1	rh-maven36	RHEL7	x86_64, s390x, aarch64, ppc64le

Components Last Updated in Red Hat Software Collections 3.3			
Red Hat Developer Toolset 8.1	devtoolset-8*	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64, ppc64le
MariaDB 10.3.27	rh-mariadb103	RHEL7	x86_64, s390x, aarch64, ppc64le
Redis 5.0.5	rh-redis5	RHEL7	x86_64, s390x, aarch64, ppc64le

Components Last Updated in Red Hat Software Collections 3.2			
PHP 7.2.24	rh-php72*	RHEL7	x86_64, s390x, aarch64, ppc64le
MySQL 8.0.21	rh-mysql80	RHEL7	x86_64, s390x, aarch64, ppc64le
Node.js 10.21.0	rh-nodejs10*	RHEL7	x86_64, s390x, aarch64, ppc64le
nginx 1.14.1	rh-nginx114*	RHEL7	x86_64, s390x, aarch64, ppc64le

Components Last Updated in Red Hat Software Collections 3.2			
Git 2.18.4	rh-git218*	RHEL7	x86_64, s390x, aarch64, ppc64le

Components Last Updated in Red Hat Software Collections 3.1			
Red Hat Developer Toolset 7.1	devtoolset-7*	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64, ppc64le
Perl 5.26.3	rh-perl526*	RHEL7	x86_64, s390x, aarch64, ppc64le
MongoDB 3.6.3	rh-mongodb36*	RHEL7	x86_64, s390x, aarch64, ppc64le
Varnish Cache 5.2.1	rh-varnish5*	RHEL7	x86_64, s390x, aarch64, ppc64le
PostgreSQL 10.15	rh-postgresql10	RHEL7	x86_64, s390x, aarch64, ppc64le
PHP 7.0.27	rh-php70*	RHEL6, RHEL7	x86_64
MySQL 5.7.24	rh-mysql57*	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64le

Components Last Updated in Red Hat Software Collections 3.0			
PHP 7.1.8	rh-php71*	RHEL7	x86_64, s390x, aarch64, ppc64le
nginx 1.12.1	rh-nginx112*	RHEL7	x86_64, s390x, aarch64, ppc64le
Python 3.6.12	rh-python36*	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64le
Maven 3.5.0	rh-maven35*	RHEL7	x86_64, s390x, aarch64, ppc64le
MariaDB 10.2.22	rh-mariadb102*	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64le
PostgreSQL 9.6.19	rh-postgresql96*	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64le

Components Last Updated in Red Hat Software Collections 3.0			
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

Components Last Updated in Red Hat Software Collections 2.4			
Red Hat Developer Toolset 6.1	devtoolset-6*	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64, ppc64le
Scala 2.10.6	rh-scala210*	RHEL7	x86_64
nginx 1.10.2	rh-nginx110*	RHEL6, RHEL7	x86_64
Node.js 6.11.3	rh-nodejs6*	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64le
Ruby 2.4.6	rh-ruby24*	RHEL6, RHEL7	x86_64
Ruby on Rails 5.0.1	rh-ror50*	RHEL6, RHEL7	x86_64
Eclipse 4.6.3	rh-eclipse46*	RHEL7	x86_64
Python 2.7.18	python27	RHEL6*, RHEL7	x86_64, s390x, aarch64, ppc64le
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

Components Last Updated in Red Hat Software Collections 2.3			
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

Components Last Updated in Red Hat Software Collections 2.3			
MongoDB 3.2.10	rh-mongodb32*	RHEL6, RHEL7	x86_64
Ruby 2.3.8	rh-ruby23*	RHEL6, RHEL7	x86_64
PHP 5.6.25	rh-php56*	RHEL6, RHEL7	x86_64

Components Last Updated in Red Hat Software Collections 2.2			
Red Hat Developer Toolset 4.1	devtoolset-4*	RHEL6, RHEL7	x86_64
MariaDB 10.1.29	rh-mariadb101*	RHEL6, RHEL7	x86_64
MongoDB 3.0.11 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

Components Last Updated in Red Hat Software Collections 2.1			
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

Components Last Updated in Red Hat Software Collections 2.0			
Red Hat Developer Toolset 3.1	devtoolset-3*	RHEL6, RHEL7	x86_64

Components Last Updated in Red Hat Software Collections 2.0			
Perl 5.20.1	rh-perl520*	RHEL6, RHEL7	x86_64
Python 3.4.2	rh-python34*	RHEL6, RHEL7	x86_64
Ruby 2.2.9	rh-ruby22*	RHEL6, RHEL7	x86_64
Ruby on Rails 4.1.5	rh-ror41*	RHEL6, RHEL7	x86_64
MariaDB 10.0.33	rh-mariadb100*	RHEL6, RHEL7	x86_64
MySQL 5.6.40	rh-mysql56*	RHEL6, RHEL7	x86_64
PostgreSQL 9.4.14	rh-postgresql94*	RHEL6, RHEL7	x86_64
Passenger 4.0.50	rh-passenger40*	RHEL6, RHEL7	x86_64
PHP 5.4.40	php54*	RHEL6, RHEL7	x86_64
PHP 5.5.21	php55*	RHEL6, RHEL7	x86_64
nginx 1.6.2	nginx16*	RHEL6, RHEL7	x86_64
DevAssistant 0.9.3	devassist09*	RHEL6, RHEL7	x86_64

Components Last Updated in Red Hat Software Collections 1			
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

Components Last Updated in Red Hat Software Collections 1			
PostgreSQL 9.2.18	postgresql92*	RHEL6, RHEL7	x86_64

Legend:

- RHEL6 – Red Hat Enterprise Linux 6

- RHEL7 – Red Hat Enterprise Linux 7

- x86_64 – AMD and Intel 64-bit architectures

- s390x – The 64-bit IBM Z architecture

- aarch64 – The 64-bit ARM architecture

- ppc64 – IBM POWER, big endian

- ppc64le – IBM POWER, little endian

- * – Retired component; this Software Collection is no longer supported

The tables above list the latest versions available through asynchronous updates.

Note that Software Collections released in Red Hat Software Collections 2.0 and later include a **rh-**prefix in their names.

Eclipse is available as a part of the Red Hat Developer Tools offering.

1.3. CHANGES IN RED HAT SOFTWARE COLLECTIONS 3.7

1.3.1. Overview

Architectures
The Red Hat Software Collections offering contains packages for Red Hat Enterprise Linux 7 running on the following architectures:

- AMD and Intel 64-bit architectures

- 64-bit IBM Z

- IBM POWER, little endian

For a full list of components and their availability, see Table 1.2, "All Available Software Collections".

New Software Collections
Red Hat Software Collections 3.7 adds the following new Software Collections:

- rh-mariadb105 — see Section 1.3.3, "Changes in MariaDB"

- rh-postgresql13 — see Section 1.3.4, "Changes in PostgreSQL"

- rh-ruby30 — see Section 1.3.5, "Changes in Ruby"

All new Software Collections are available only for Red Hat Enterprise Linux 7.

Updated Software Collections
The following components has been updated in Red Hat Software Collections 3.7:

- devtoolset-10 — see Section 1.3.2, "Changes in Red Hat Developer Toolset"

- rh-jmc — see Section 1.3.6, "Changes in JDK Mission Control"

- rh-ruby27 — see Section 1.3.5, "Changes in Ruby"

- rh-ruby26 — see Section 1.3.5, "Changes in Ruby"

In addition, a new package, rh-postgresql12-pg_repack is now available for **PostgreSQL 12.**

Red Hat Software Collections Container Images
The following container images are new in Red Hat Software Collections 3.7:

- rhscl/mariadb-105-rhel7

- rhscl/postgresql-13-rhel7

- rhscl/ruby-30-rhel7

The following container images have been updated in Red Hat Software Collections 3.7

- rhscl/devtoolset-10-toolchain-rhel7

- rhscl/devtoolset-10-perftools-rhel7

- rhscl/ruby-27-rhel7

- rhscl/ruby-26-rhel7

For more information about Red Hat Software Collections container images, see Section 3.4, "Red Hat Software Collections Container Images".

1.3.2. Changes in Red Hat Developer Toolset

The following components have been upgraded in Red Hat Developer Toolset 10.1 compared to the previous release:

- **SystemTap** to version 4.4

- **Dyninst** to version 10.2.1

- **elfutils** to version 0.182

In addition, bug fix updates are available for the following components:

- **GCC**

- **GDB**

- **binutils**

- **annobin**

For detailed information on changes in 10.1, see the Red Hat Developer Toolset User Guide .

1.3.3. Changes in MariaDB

The new rh-mariadb105 Software Collection provides **MariaDB 10.5.9**.

Notable enhancements over the previously available version 10.3 include:

- **MariaDB** now uses the **unix_socket** authentication plug-in by default. The plug-in enables users to use operating system credentials when connecting to **MariaDB** through the local Unix socket file.

- **MariaDB** supports a new **FLUSH SSL** command to reload SSL certificates without a server restart.

- **MariaDB** adds **mariadb-*** named binaries and **mysql*** symbolic links pointing to the **mariadb-*** binaires. For example, the **mysqladmin**, **mysqlaccess**, and **mysqlshow** symlinks point to the **mariadb-admin**, **mariadb-access**, and **mariadb-show** binaries, respectively.

- **MariaDB** supports a new **INET6** data type for storing IPv6 addresses.

- **MariaDB** now uses the Perl Compatible Regular Expressions (PCRE) library version 2.

- The **SUPER** privilege has been split into several privileges to better align with each user role. As a result, certain statements have changed required privileges.

- **MariaDB** adds a new global variable, **binlog_row_metadata**, as well as system variables and status variables to control the amount of metadata logged.

- The default value of the **eq_range_index_dive_limit** variable has been changed from **0** to **200**.

- A new **SHUTDOWN WAIT FOR ALL SLAVES** server command and a new **mysqladmin shutdown --wait-for-all-slaves** option have been added to instruct the server to shut down only after the last binlog event has been sent to all connected replicas.

- In parallel replication, the **slave_parallel_mode** variable now defaults to **optimistic**.

The **InnoDB** storage engine introduces the following changes:

- **InnoDB** now supports an instant **DROP COLUMN** operation and enables users to change the column order.

- Defaults of the following variables have been changed: **innodb_adaptive_hash_index** to **OFF** and **innodb_checksum_algorithm** to **full_crc32**.

- Several **InnoDB** variables have been removed or deprecated.

MariaDB Galera Cluster has been upgraded to version 4 with the following notable changes:

- **Galera** adds a new streaming replication feature, which supports replicating transactions of unlimited size. During an execution of streaming replication, a cluster replicates a transaction in small fragments.

- **Galera** now fully supports Global Transaction ID (GTID).

- The default value for the **wsrep_on** option in the **/etc/my.cnf.d/galera.cnf** file has changed from **1** to **0** to prevent end users from starting **wsrep** replication without configuring required additional options.

Changes to the PAM plug-in in **MariaDB 10.5** include:

- **MariaDB 10.5** adds a new version of the Pluggable Authentication Modules (PAM) plug-in. The PAM plug-in version 2.0 performs PAM authentication using a separate **setuid root** helper binary, which enables **MariaDB** to utilize additional PAM modules.

- In **MariaDB 10.5**, the Pluggable Authentication Modules (PAM) plug-in and its related files have been moved to a new subpackage, mariadb-pam. This subpackage contains both PAM plug-in versions: version 2.0 is the default, and version 1.0 is available as the **auth_pam_v1** shared object library.

- The rh-mariadb105-mariadb-pam package is not installed by default with the **MariaDB** server. To make the PAM authentication plug-in available in **MariaDB 10.5**, install the rh-mariadb105-mariadb-pam package manually.

The rh-mariadb105 Software Collection includes the rh-mariadb105-syspaths package, which installs packages that provide system-wide wrappers for binaries, scripts, manual pages, and other. After installing the rh-mariadb105*-syspaths packages, users are not required to use the **scl enable** command for correct functioning of the binaries and scripts provided by the rh-mariadb105* packages. Note that the *-syspaths packages conflict with the corresponding packages from the base Red Hat Enterprise Linux system. To find out more about syspaths, see the Red Hat Software Collections Packaging Guide.

For compatibility notes and migration instructions, see Section 5.1, "Migrating to MariaDB 10.5" .

For detailed changes in **MariaDB 10.5**, see the upstream documentation .

1.3.4. Changes in PostgreSQL

The new rh-postgresql13 Software Collection includes **PostgreSQL 13.2**. This release introduces various enhancements over version 12, such as:

- Performance improvements resulting from de-duplication of B-tree index entries

- Improved performance for queries that use aggregates or partitioned tables

- Improved query planning when using extended statistics

- Parallelized vacuuming of indexes

- Incremental sorting

For detailed changes, see the upstream release notes for PostgreSQL 13.

The following new subpackages are available with the rh-postgresql13 Software Collection:

- The pg_repack package provides a **PostgreSQL** extension that lets you remove bloat from tables and indexes, and optionally restore the physical order of clustered indexes. For details, see the upstream documentation regarding usage and examples.

  The pg_repack subpackage is now available also for the rh-postgresql12 Software Collection.

- The plpython3 package provides the **PL/Python** procedural language extension based on Python 3. **PL/Python** enables you to write **PostgreSQL** functions in the **Python** programming language. For details, see the upstream documentation.

  Previously released **PostgreSQL** Software Collections include only the plpython package based on **Python 2**. Red Hat Enterprise Linux 8 provides only plpython3. The rh-postgresql13 Software Collection includes both plpython and plpython3, so that you can migrate to plpython3 before upgrading to Red Hat Enterprise Linux 8.

In addition, the rh-postgresql13 Software Collection includes the rh-postgresql13-syspaths package, which installs packages that provide system-wide wrappers for binaries, scripts, manual pages, and others. After installing the rh-postgresql13*-syspaths packages, users are not required to use the **scl enable** command for correct functioning of the binaries and scripts provided by the rh-postgresql13* packages. Note that the *-syspaths packages conflict with the corresponding packages from the base Red Hat Enterprise Linux system. To find out more about syspaths, see the Red Hat Software Collections Packaging Guide.

Note that support for Just-In-Time (JIT) compilation, available in upstream since **PostgreSQL 11**, is not provided by the rh-postgresql13 Software Collection.

For information on migration, see Section 5.3, "Migrating to PostgreSQL 13".

1.3.5. Changes in Ruby

The new rh-ruby30 Software Collection provides **Ruby 3.0.1**, which introduces a number of performance improvements, bug fixes, and new features.

Notable enhancements include:

- Concurrency and parallelism features:

  - **Ractor**, an Actor-model abstraction that provides thread-safe parallel execution, is provided as an experimental feature.

  - **Fiber Scheduler** has been introduced as an experimental feature. **Fiber Scheduler** intercepts blocking operations, which enables light-weight concurrency without changing existing code.

- Static analysis features:

  - The **RBS** language has been introduced, which describes the structure of **Ruby** programs. The **rbs** gem has been added to parse type definitions written in **RBS**.

  - The **TypeProf** utility has been introduced, which is a type analysis tool for **Ruby** code.

- Pattern matching with the **case**/**in** expression is no longer experimental.

- One-line pattern matching has been redesigned as an experimental feature.

- Find pattern has been added as an experimental feature.

The following performance improvements have been implemented:

- Pasting long code to the **Interactive Ruby Shell (IRB)** is now significantly faster.

- The **measure** command has been added to **IRB** for time measurement.

Other notable changes include:

- Keyword arguments have been separated from other arguments, see the upstream documentation for details.

- The default directory for user-installed gems is now **$HOME/.local/share/gem/** unless the **$HOME/.gem/** directory is already present.

For more information about changes in **Ruby 3.0**, see the upstream announcement for version 3.0.0 and 3.0.1.

The rh-ruby27 and rh-ruby26 Software Collections have been updated with security and bug fixes.

1.3.6. Changes in JDK Mission Control

JDK Mission Control (JMC), provided by the rh-jmc Software Collection, has been upgraded from version 7.1.1 to version 8.0.0. Notable enhancements include:

- The **Treemap** viewer has been added to the **JOverflow** plug-in for visualizing memory usage by classes.

- The **Threads** graph has been enhanced with more filtering and zoom options.

- JDK Mission Control now provides support for opening JDK Flight Recorder recordings compressed with the LZ4 algorithm.

- New columns have been added to the **Memory** and **TLAB** views to help you identify areas of allocation pressure.

- **Graph** view has been added to improve visualization of stack traces.

- The **Percentage** column has been added to histogram tables.

For more information, see the upstream release notes.

1.4. COMPATIBILITY INFORMATION

Red Hat Software Collections 3.7 is available for all supported releases of Red Hat Enterprise Linux 7 on AMD and Intel 64-bit architectures, 64-bit IBM Z, and IBM POWER, little endian.

Certain previously released components are available also for the 64-bit ARM architecture.

For a full list of available components, see Table 1.2, "All Available Software Collections".

1.5. KNOWN ISSUES

rh-mariadb105 component, BZ#1942526

When the **OQGraph** storage engine plug-in is loaded to the **MariaDB 10.5** server, **MariaDB** does not warn about dropping a non-existent table. In particular, when the user attempts to drop a non-existent table using the **DROP TABLE** or **DROP TABLE IF EXISTS** SQL commands, **MariaDB** neither returns an error message nor logs a warning.

Note that the **OQGraph** plug-in is provided by the mariadb-oqgraph-engine package, which is not installed by default.

rh-mariadb component

The rh-mariadb103 Software Collection provides the Pluggable Authentication Modules (PAM) plug-in version 1.0. The rh-mariadb105 Software Collection provides the plug-in versions 1.0 and 2.0, version 2.0 is the default. The PAM plug-in version 1.0 in **MariaDB** does not work. To work around this problem, use the PAM plug-in version 2.0 provided by rh-mariadb105.

rh-ruby27 component, BZ#1836201

When a custom script requires the Psych YAML parser and afterwards uses the **Gem.load_yaml** method, running the script fails with the following error message:

> superclass mismatch for class Mark (TypeError)

To work around this problem, add the **gem 'psych'** line to the script somewhere above the **require 'psych'** line:

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

multiple components, BZ#1716378

Certain files provided by the Software Collections debuginfo packages might conflict with the corresponding debuginfo package files from the base Red Hat Enterprise Linux system or from other versions of Red Hat Software Collections components. For example, the python27-python-debuginfo package files might conflict with the corresponding files from the python-debuginfo package installed on the core system. Similarly, files from the httpd24-mod_auth_mellon-debuginfo package might conflict with similar files provided by the base system mod_auth_mellon-debuginfo package. To work around this problem, uninstall the base system debuginfo package prior to installing the Software Collection debuginfo package.

rh-mysql80 component, BZ#1646363

The **mysql-connector-java** database connector does not work with the **MySQL 8.0** server. To work around this problem, use the **mariadb-java-client** database connector from the rh-mariadb103 Software Collection.

rh-mysql80 component, BZ#1646158

The default character set has been changed to **utf8mb4** in **MySQL 8.0** but this character set is unsupported by the **php-mysqlnd** database connector. Consequently, **php-mysqlnd** fails to connect in the default configuration. To work around this problem, specify a known character set as a parameter of the MySQL server configuration. For example, modify the **/etc/opt/rh/rh-mysql80/my.cnf.d/mysql-server.cnf** file to read:

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

httpd24 component, BZ#1429006

Since **httpd 2.4.27**, the **mod_http2** module is no longer supported with the default **prefork** Multi-Processing Module (MPM). To enable HTTP/2 support, edit the configuration file at **/opt/rh/httpd24/root/etc/httpd/conf.modules.d/00-mpm.conf** and switch to the **event** or **worker** MPM.

Note that the HTTP/2 server-push feature does not work on the 64-bit ARM architecture, 64-bit IBM Z, and IBM POWER, little endian.

httpd24 component, BZ#1224763

When using the **mod_proxy_fcgi** module with FastCGI Process Manager (PHP-FPM), **httpd** uses port **8000** for the FastCGI protocol by default instead of the correct port **9000**. To work around this problem, specify the correct port explicitly in configuration.

httpd24 component, BZ#1382706

When SELinux is enabled, the **LD_LIBRARY_PATH** environment variable is not passed through to CGI scripts invoked by **httpd**. As a consequence, in some cases it is impossible to invoke executables from Software Collections enabled in the **/opt/rh/httpd24/service-environment** file from CGI scripts run by **httpd**. To work around this problem, set **LD_LIBRARY_PATH** as desired from within the CGI script.

httpd24 component

Compiling external applications against the Apache Portable Runtime (APR) and APR-util libraries from the httpd24 Software Collection is not supported. The LD_LIBRARY_PATH environment variable is not set in httpd24 because it is not required by any application in this Software Collection.

scl-utils component

In Red Hat Enterprise Linux 7.5 and earlier, due to an architecture-specific macro bug in the scl-utils package, the **<collection>/root/usr/lib64/** directory does not have the correct package ownership on the 64-bit ARM architecture and on IBM POWER, little endian. As a consequence, this directory is not removed when a Software Collection is uninstalled. To work around this problem, manually delete **<collection>/root/usr/lib64/** when removing a Software Collection.

maven component

When the user has installed both the Red Hat Enterprise Linux system version of maven-local package and the rh-maven*-maven-local package, **XMvn**, a tool used for building Java RPM packages, run from the Maven Software Collection tries to read the configuration file from the base system and fails. To work around this problem, uninstall the maven-local package from the base Red Hat Enterprise Linux system.

perl component

It is impossible to install more than one **mod_perl.so** library. As a consequence, it is not possible to use the **mod_perl** module from more than one **Perl** Software Collection.

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

When uninstalling the httpd24, rh-mariadb*, rh-mysql*, rh-nodejs*, rh-perl*, rh-php*, python27, rh-python*, or rh-ruby* packages, the order of uninstalling can be relevant due to ownership of dependent packages. As a consequence, some directories and files might not be removed properly and might remain on the system.

mariadb, mysql components, BZ#1194611

Since **MariaDB 10** and **MySQL 5.6**, the rh-mariadb*-mariadb-server and rh-mysql*-mysql-server packages no longer provide the **test** database by default. Although this database is not created during initialization, the grant tables are prefilled with the same values as when **test** was created by default. As a consequence, upon a later creation of the **test** or **test_*** databases, these databases have less restricted access rights than is default for new databases.

Additionally, when running benchmarks, the **run-all-tests** script no longer works out of the box with example parameters. You need to create a test database before running the tests and specify the database name in the **--database** parameter. If the parameter is not specified, **test** is taken by default but you need to make sure the **test** database exist.

mariadb, mysql, postgresql components

Red Hat Software Collections contains the **MySQL 8.0**, **MariaDB 10.3**, **MariaDB 10.5**, **PostgreSQL 10**, **PostgreSQL 12**, and **PostgreSQL 13** database servers. The core Red Hat Enterprise Linux 7 provides earlier versions of the **MariaDB** and **PostgreSQL** databases (client library and daemon). Client libraries are also used in database connectors for dynamic languages, libraries, and so on.

The client library packaged in the Red Hat Software Collections database packages in the **PostgreSQL** component is not supposed to be used, as it is included only for purposes of server utilities and the daemon. Users are instead expected to use the system library and the database connectors provided with the core system.

A protocol, which is used between the client library and the daemon, is stable across database versions, so, for example, using the **PostgreSQL 10** client library with the **PostgreSQL 12** or **13** daemon works as expected.

mariadb, mysql components

MariaDB and MySQL do not make use of the **/opt/***provider***/***collection***/root** prefix when creating log files. Note that log files are saved in the **/var/opt/***provider***/***collection***/log/** directory, not in **/opt/***provider***/***collection***/root/var/log/**.

1.6. OTHER NOTES

rh-ruby*, rh-python*, rh-php* components

Using Software Collections on a read-only NFS has several limitations.

- Ruby gems cannot be installed while the rh-ruby* Software Collection is on a read-only NFS. Consequently, for example, when the user tries to install the ab gem using the **gem install ab** command, an error message is displayed, for example:

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

  The same problem occurs when the user tries to update or install gems from an external source by running the **bundle update** or **bundle install** commands.

- When installing Python packages on a read-only NFS using the Python Package Index (PyPI), running the **pip** command fails with an error message similar to this:

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

- Installing packages from PHP Extension and Application Repository (PEAR) on a read-only NFS using the **pear** command fails with the error message:

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

This is an expected behavior.

httpd component

Language modules for Apache are supported only with the Red Hat Software Collections version of **Apache httpd** and not with the Red Hat Enterprise Linux system versions of **httpd**. For example, the **mod_wsgi** module from the rh-python35 Collection can be used only with the httpd24 Collection.

all components

Since Red Hat Software Collections 2.0, configuration files, variable data, and runtime data of individual Collections are stored in different directories than in previous versions of Red Hat Software Collections.

coreutils, **util-linux**, **screen** components

Some utilities, for example, **su**, **login**, or **screen**, do not export environment settings in all cases, which can lead to unexpected results. It is therefore recommended to use **sudo** instead of **su** and set the **env_keep** environment variable in the **/etc/sudoers** file. Alternatively, you can run commands in a reverse order; for example:

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

instead of

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

When using tools like **screen** or **login**, you can use the following command to preserve the environment settings:

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

python component

When the user tries to install more than one scldevel package from the python27 and rh-python* Software Collections, a transaction check error message is returned. This is an expected behavior because the user can install only one set of the macro files provided by the packages (**%scl_python**, **%scl_*prefix*_python**).

php component

When the user tries to install more than one scldevel package from the rh-php* Software Collections, a transaction check error message is returned. This is an expected behavior because the user can install only one set of the macro files provided by the packages (**%scl_php**, **%scl_*prefix*_php**).

ruby component

When the user tries to install more than one scldevel package from the rh-ruby* Software Collections, a transaction check error message is returned. This is an expected behavior because the user can install only one set of the macro files provided by the packages (**%scl_ruby**, **%scl_*prefix*_ruby**).

perl component

When the user tries to install more than one scldevel package from the rh-perl*

Software Collections, a transaction check error message is returned. This is an expected behavior because the user can install only one set of the macro files provided by the packages (**%scl_perl**, **%scl_*prefix*_perl**).

nginx component

When the user tries to install more than one scldevel package from the rh-nginx* Software Collections, a transaction check error message is returned. This is an expected behavior because the user can install only one set of the macro files provided by the packages (**%scl_nginx**, **%scl_*prefix*_nginx**).

python component

To mitigate the Web Cache Poisoning CVE-2021-23336 in the Python **urllib** library, the default separator for the **urllib.parse.parse_qsl** and **urllib.parse.parse_qs** functions is being changed from both ampersand (**&**) and semicolon (**;**) to only an ampersand.

This change has been implemented in the python27 and rh-python38 Software Collections with the release of the RHSA-2021:3252 and RHSA-2021:3254 advisories.

The change of the default separator is potentially backwards incompatible, therefore Red Hat provides a way to configure the behavior in Python packages where the default separator has been changed. In addition, the affected **urllib** parsing functions issue a warning if they detect that a customer's application has been affected by the change.

For more information, see the Mitigation of Web Cache Poisoning in the Python urllib library (CVE-2021-23336) Knowledgebase article.

python component

The release of the RHSA-2021:3254 advisory introduces the following change in the rh-python38 Software Collection:

To mitigate CVE-2021-29921, the Python **ipaddress** module now rejects IPv4 addresses with leading zeros with an **AddressValueError: Leading zeros are not permitted** error.

Customers who rely on the previous behavior can pre-process their IPv4 address inputs to strip the leading zeros off. For example:

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

To strip the leading zeros off with an explicit loop for readability, use:

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

1.7. DEPRECATED FUNCTIONALITY

httpd24 component, BZ#1434053

Previously, in an SSL/TLS configuration requiring name-based SSL virtual host selection, the **mod_ssl** module rejected requests with a **400 Bad Request** error, if the host name provided in the **Host:** header did not match the host name provided in a Server Name Indication (SNI) header. Such requests are no longer rejected if the configured SSL/TLS security parameters are identical between the selected virtual hosts, in-line with the behavior of upstream **mod_ssl**.

CHAPTER 2. INSTALLATION

This chapter describes in detail how to get access to the content set, install Red Hat Software Collections 3.7 on the system, and rebuild Red Hat Software Collections.

2.1. GETTING ACCESS TO RED HAT SOFTWARE COLLECTIONS

The Red Hat Software Collections content set is available to customers with Red Hat Enterprise Linux subscriptions listed in the Knowledgebase article How to use Red Hat Software Collections (RHSCL) or Red Hat Developer Toolset (DTS)?. For information on how to register your system with Red Hat Subscription Management (RHSM), see Using and Configuring Red Hat Subscription Manager . For detailed instructions on how to enable Red Hat Software Collections using RHSM, see Section 2.1.1, "Using Red Hat Subscription Management".

Since Red Hat Software Collections 2.2, the Red Hat Software Collections and Red Hat Developer Toolset content is available also in the ISO format at https://access.redhat.com/downloads, specifically for Server and Workstation. Note that packages that require the **Optional** repository, which are listed in Section 2.1.2, "Packages from the Optional Repository" , cannot be installed from the ISO image.

> **NOTE**
>
> Packages that require the **Optional** repository cannot be installed from the ISO image. A list of packages that require enabling of the **Optional** repository is provided in Section 2.1.2, "Packages from the Optional Repository" .
>
> Beta content is unavailable in the ISO format.

2.1.1. Using Red Hat Subscription Management

If your system is registered with Red Hat Subscription Management, complete the following steps to attach the subscription that provides access to the repository for Red Hat Software Collections and enable the repository:

1. Display a list of all subscriptions that are available for your system and determine the pool ID of a subscription that provides Red Hat Software Collections. To do so, type the following at a shell prompt as **root**:

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

   For each available subscription, this command displays its name, unique identifier, expiration date, and other details related to it. The pool ID is listed on a line beginning with **Pool Id**.

2. Attach the appropriate subscription to your system by running the following command as **root**:

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

   Replace *pool_id* with the pool ID you determined in the previous step. To verify the list of subscriptions your system has currently attached, type as **root**:

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

3. Display the list of available Yum list repositories to retrieve repository metadata and determine the exact name of the Red Hat Software Collections repositories. As **root**, type:

> **subscription-manager repos --list**

Or alternatively, run **yum repolist all** for a brief list.

The repository names depend on the specific version of Red Hat Enterprise Linux you are using and are in the following format:

> rhel-*variant*-rhscl-6-rpms
> rhel-*variant*-rhscl-6-debug-rpms
> rhel-*variant*-rhscl-6-source-rpms
>
> rhel-server-rhscl-6-eus-rpms
> rhel-server-rhscl-6-eus-source-rpms
> rhel-server-rhscl-6-eus-debug-rpms
>
> rhel-*variant*-rhscl-7-rpms
> rhel-*variant*-rhscl-7-debug-rpms
> rhel-*variant*-rhscl-7-source-rpms
>
> rhel-server-rhscl-7-eus-rpms
> rhel-server-rhscl-7-eus-source-rpms
> rhel-server-rhscl-7-eus-debug-rpms

Replace *variant* with the Red Hat Enterprise Linux system variant, that is, **server** or **workstation**. Note that Red Hat Software Collections is supported neither on the **Client** nor on the **ComputeNode** variant.

4. Enable the appropriate repository by running the following command as **root**:

> **subscription-manager repos --enable** *repository*

Once the subscription is attached to the system, you can install Red Hat Software Collections as described in Section 2.2, "Installing Red Hat Software Collections" . For more information on how to register your system using Red Hat Subscription Management and associate it with subscriptions, see Using and Configuring Red Hat Subscription Manager .

> **NOTE**
>
> Subscription through RHN is no longer available.

2.1.2. Packages from the Optional Repository

Some of the Red Hat Software Collections packages require the **Optional** repository to be enabled in order to complete the full installation of these packages. For detailed instructions on how to subscribe your system to this repository, see the relevant Knowledgebase article How to access Optional and Supplementary channels, and –devel packages using Red Hat Subscription Management (RHSM)?.

Packages from Software Collections for Red Hat Enterprise Linux that require the **Optional** repository to be enabled are listed in the tables below. Note that packages from the **Optional** repository are unsupported. For details, see the Knowledgebase article Support policy of the optional and supplementary channels in Red Hat Enterprise Linux.

Table 2.1. Packages That Require Enabling of the Optional Repository in Red Hat Enterprise Linux 7

Package from a Software Collection	Required Package from the Optional Repository
devtoolset-10-build	scl-utils-build
devtoolset-10-dyninst-testsuite	glibc-static
devtoolset-10-elfutils-debuginfod	bsdtar
devtoolset-10-gcc-plugin-devel	libmpc-devel
devtoolset-10-gdb	source-highlight
devtoolset-9-build	scl-utils-build
devtoolset-9-dyninst-testsuite	glibc-static
devtoolset-9-gcc-plugin-devel	libmpc-devel
devtoolset-9-gdb	source-highlight
httpd24-mod_ldap	apr-util-ldap
httpd24-mod_session	apr-util-openssl
python27-python-debug	tix
python27-python-devel	scl-utils-build
python27-tkinter	tix
rh-git227-git-cvs	cvsps
rh-git227-git-svn	perl-Git-SVN, subversion
rh-git227-perl-Git-SVN	subversion-perl
rh-java-common-ant-apache-bsf	rhino
rh-java-common-batik	rhino
rh-maven35-build	scl-utils-build
rh-maven35-xpp3-javadoc	java-1.8.0-openjdk-javadoc-zip, java-11-openjdk-javadoc, java-1.7.0-openjdk-javadoc, java-11-openjdk-javadoc-zip, java-1.8.0-openjdk-javadoc

Package from a Software Collection	Required Package from the Optional Repository
rh-php73-php-devel	pcre2-devel
rh-php73-php-pspell	aspell
rh-python38-python-devel	scl-utils-build

2.2. INSTALLING RED HAT SOFTWARE COLLECTIONS

Red Hat Software Collections is distributed as a collection of RPM packages that can be installed, updated, and uninstalled by using the standard package management tools included in Red Hat Enterprise Linux. Note that a valid subscription is required to install Red Hat Software Collections on your system. For detailed instructions on how to associate your system with an appropriate subscription and get access to Red Hat Software Collections, see Section 2.1, "Getting Access to Red Hat Software Collections".

Use of Red Hat Software Collections 3.7 requires the removal of any earlier pre-release versions. If you have installed any previous version of Red Hat Software Collections 2.1 component, uninstall it from your system and install the new version as described in the Section 2.3, "Uninstalling Red Hat Software Collections" and Section 2.2.1, "Installing Individual Software Collections" sections.

The in-place upgrade from Red Hat Enterprise Linux 6 to Red Hat Enterprise Linux 7 is not supported by Red Hat Software Collections. As a consequence, the installed Software Collections might not work correctly after the upgrade. If you want to upgrade from Red Hat Enterprise Linux 6 to Red Hat Enterprise Linux 7, it is strongly recommended to remove all Red Hat Software Collections packages, perform the in-place upgrade, update the Red Hat Software Collections repository, and install the Software Collections packages again. It is advisable to back up all data before upgrading.

2.2.1. Installing Individual Software Collections

To install any of the Software Collections that are listed in Table 1.1, "Red Hat Software Collections Components", install the corresponding meta package by typing the following at a shell prompt as **root**:

> **yum install *software_collection*...**

Replace *software_collection* with a space-separated list of Software Collections you want to install. For example, to install rh-php73 and rh-mariadb105, type as **root**:

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

This installs the main meta package for the selected Software Collection and a set of required packages as its dependencies. For information on how to install additional packages such as additional modules, see Section 2.2.2, "Installing Optional Packages".

2.2.2. Installing Optional Packages

Each component of Red Hat Software Collections is distributed with a number of optional packages that are not installed by default. To list all packages that are part of a certain Software Collection but are not installed on your system, type the following at a shell prompt:

> **yum list available *software_collection*-***

To install any of these optional packages, type as **root**:

> **yum install *package_name*...**

Replace *package_name* with a space-separated list of packages that you want to install. For example, to install the rh-perl530-perl-CPAN and rh-perl530-perl-Archive-Tar, type:

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

2.2.3. Installing Debugging Information

To install debugging information for any of the Red Hat Software Collections packages, make sure that the yum-utils package is installed and type the following command as **root**:

> **debuginfo-install *package_name***

For example, to install debugging information for the rh-ruby27-ruby package, type:

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

Note that you need to have access to the repository with these packages. If your system is registered with Red Hat Subscription Management, enable the **rhel-*variant*-rhscl-6-debug-rpms** or **rhel-*variant*-rhscl-7-debug-rpms** repository as described in Section 2.1.1, "Using Red Hat Subscription Management". For more information on how to get access to debuginfo packages, see How can I download or install debuginfo packages for RHEL systems? .

2.3. UNINSTALLING RED HAT SOFTWARE COLLECTIONS

To uninstall any of the Software Collections components, type the following at a shell prompt as **root**:

> **yum remove *software_collection****

Replace *software_collection* with the Software Collection component you want to uninstall.

Note that uninstallation of the packages provided by Red Hat Software Collections does not affect the Red Hat Enterprise Linux system versions of these tools.

2.4. REBUILDING RED HAT SOFTWARE COLLECTIONS

<collection>-build packages are not provided by default. If you wish to rebuild a collection and do not want or cannot use the **rpmbuild --define 'scl foo'** command, you first need to rebuild the metapackage, which provides the <collection>-build package.

Note that existing collections should not be rebuilt with different content. To add new packages into an existing collection, you need to create a new collection containing the new packages and make it dependent on packages from the original collection. The original collection has to be used without changes.

For detailed information on building Software Collections, refer to the Red Hat Software Collections Packaging Guide.

CHAPTER 3. USAGE

This chapter describes the necessary steps for using Red Hat Software Collections 3.7, and deploying applications that use Red Hat Software Collections.

3.1. USING RED HAT SOFTWARE COLLECTIONS

3.1.1. Running an Executable from a Software Collection

To run an executable from a particular Software Collection, type the following command at a shell prompt:

> **scl enable *software_collection*... '*command*...'**

Or, alternatively, use the following command:

> **scl enable *software_collection*... -- *command*...**

Replace *software_collection* with a space-separated list of Software Collections you want to use and *command* with the command you want to run. For example, to execute a Perl program stored in a file named **hello.pl** with the Perl interpreter from the perl526 Software Collection, type:

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

You can execute any command using the **scl** utility, causing it to be run with the executables from a selected Software Collection in preference to their possible Red Hat Enterprise Linux system equivalents. For a complete list of Software Collections that are distributed with Red Hat Software Collections, see Table 1.1, "Red Hat Software Collections Components".

3.1.2. Running a Shell Session with a Software Collection as Default

To start a new shell session with executables from a selected Software Collection in preference to their Red Hat Enterprise Linux equivalents, type the following at a shell prompt:

> **scl enable *software_collection*... bash**

Replace *software_collection* with a space-separated list of Software Collections you want to use. For example, to start a new shell session with the python27 and rh-postgresql12 Software Collections as default, type:

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

The list of Software Collections that are enabled in the current session is stored in the **$X_SCLS** environment variable, for instance:

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

For a complete list of Software Collections that are distributed with Red Hat Software Collections, see Table 1.1, "Red Hat Software Collections Components".

3.1.3. Running a System Service from a Software Collection

In Red Hat Enterprise Linux 7, init scripts have been replaced by **systemd** service unit files, which end with the **.service** file extension and serve a similar purpose as init scripts. To start a service in the current session, execute the following command as **root**:

> **systemctl start** *software_collection-service_name***.service**

Replace *software_collection* with the name of the Software Collection and *service_name* with the name of the service you want to start.

To configure this service to start automatically at boot time, type the following command as **root**:

> **systemctl enable** *software_collection-service_name***.service**

For example, to start the **postgresql** service from the rh-postgresql12 Software Collection and enable it at boot time, type as **root**:

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

For more information on how to manage system services in Red Hat Enterprise Linux 7, refer to the Red Hat Enterprise Linux 7 System Administrator's Guide . For a complete list of Software Collections that are distributed with Red Hat Software Collections, see Table 1.1, "Red Hat Software Collections Components".

3.2. ACCESSING A MANUAL PAGE FROM A SOFTWARE COLLECTION

Every Software Collection contains a general manual page that describes the content of this component. Each manual page has the same name as the component and it is located in the**/opt/rh** directory.

To read a manual page for a Software Collection, type the following command:

> **scl enable** *software_collection* **'man** *software_collection***'**

Replace *software_collection* with the particular Red Hat Software Collections component. For example, to display the manual page for rh-mariadb105, type:

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

3.3. DEPLOYING APPLICATIONS THAT USE RED HAT SOFTWARE COLLECTIONS

In general, you can use one of the following two approaches to deploy an application that depends on a component from Red Hat Software Collections in production:

- Install all required Software Collections and packages manually and then deploy your application, or

- Create a new Software Collection for your application and specify all required Software Collections and other packages as dependencies.

For more information on how to manually install individual Red Hat Software Collections components, see Section 2.2, "Installing Red Hat Software Collections" . For further details on how to use Red Hat Software Collections, see Section 3.1, "Using Red Hat Software Collections" . For a detailed explanation of how to create a custom Software Collection or extend an existing one, read the Red Hat Software Collections Packaging Guide.

3.4. RED HAT SOFTWARE COLLECTIONS CONTAINER IMAGES

Container images based on Red Hat Software Collections include applications, daemons, and databases. The images can be run on Red Hat Enterprise Linux 7 Server and Red Hat Enterprise Linux Atomic Host. For information about their usage, see Using Red Hat Software Collections 3 Container Images. For details regarding container images based on Red Hat Software Collections versions 2.4 and earlier, see Using Red Hat Software Collections 2 Container Images .

Note that only the latest version of each container image is supported.

The following container images are available with Red Hat Software Collections 3.7:

- rhscl/mariadb-105-rhel7

- rhscl/postgresql-13-rhel7

- rhscl/ruby-30-rhel7

- rhscl/devtoolset-10-toolchain-rhel7

- rhscl/devtoolset-10-perftools-rhel7

- rhscl/ruby-27-rhel7

- rhscl/ruby-26-rhel7

The following container images are based on Red Hat Software Collections 3.6:

- rhscl/httpd-24-rhel7

- rhscl/nginx-118-rhel7

- rhscl/nodej-14-rhel7

- rhscl/perl-530-rhel7

- rhscl/php-73-rhel7

The following container images are based on Red Hat Software Collections 3.5:

- rhscl/python-38-rhel7

- rhscl/varnish-6-rhel7

The following container images are based on Red Hat Software Collections 3.4:

- rhscl/nginx-116-rhel7

- rhscl/nodejs-12-rhel7

- rhscl/postgresql-12-rhel7

The following container images are based on Red Hat Software Collections 3.3:

- rhscl/mariadb-103-rhel7

- rhscl/redis-5-rhel7

The following container image is based on Red Hat Software Collections 3.2:

- rhscl/mysql-80-rhel7

The following container image is based on Red Hat Software Collections 3.1:

- rhscl/postgresql-10-rhel7

The following container image is based on Red Hat Software Collections 2:

- rhscl/python-27-rhel7

- rhscl/s2i-base-rhel7

CHAPTER 4. SPECIFICS OF INDIVIDUAL SOFTWARE COLLECTIONS

This chapter is focused on the specifics of certain Software Collections and provides additional details concerning these components.

4.1. RED HAT DEVELOPER TOOLSET

Red Hat Developer Toolset is designed for developers working on the Red Hat Enterprise Linux platform. Red Hat Developer Toolset provides current versions of the **GNU Compiler Collection**, **GNU Debugger**, and other development, debugging, and performance monitoring tools. Similarly to other Software Collections, an additional set of tools is installed into the **/opt/** directory. These tools are enabled by the user on demand using the supplied **scl** utility. Similarly to other Software Collections, these do not replace the Red Hat Enterprise Linux system versions of these tools, nor will they be used in preference to those system versions unless explicitly invoked using the **scl** utility.

For an overview of features, refer to the Features section of the *Red Hat Developer Toolset Release Notes*. For detailed information regarding usage and changes in 10.1, see the Red Hat Developer Toolset User Guide.

4.2. MAVEN

The rh-maven36 Software Collection, available only for Red Hat Enterprise Linux 7, provides a software project management and comprehension tool. Based on the concept of a project object model (POM), **Maven** can manage a project's build, reporting, and documentation from a central piece of information.

To install the rh-maven36 Collection, type the following command as **root**:

```
yum install rh-maven36
```

To enable this collection, type the following command at a shell prompt:

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

Global Maven settings, such as remote repositories or mirrors, can be customized by editing the **/opt/rh/rh-maven36/root/etc/maven/settings.xml** file.

For more information about using Maven, refer to the Maven documentation. Usage of plug-ins is described in this section; to find documentation regarding individual plug-ins, see the index of plug-ins.

4.3. DATABASE CONNECTORS

Database connector packages provide the database client functionality, which is necessary for local or remote connection to a database server. Table 4.1, "Interoperability Between Languages and Databases" lists Software Collections with language runtimes that include connectors for certain database servers:

- **yes** – the combination is supported

- **no** – the combination is not supported

Table 4.1. Interoperability Between Languages and Databases

Language (Software Collection)	Database					
	MariaDB	MongoDB	MySQL	PostgreSQL	Redis	SQLite3
rh-nodejs4	no	no	no	no	no	no
rh-nodejs6	no	no	no	no	no	no
rh-nodejs8	no	no	no	no	no	no
rh-nodejs10	no	no	no	no	no	no
rh-nodejs12	no	no	no	no	no	no
rh-nodejs14	no	no	no	no	no	no
rh-perl520	yes	no	yes	yes	no	no
rh-perl524	yes	no	yes	yes	no	no
rh-perl526	yes	no	yes	yes	no	no
rh-perl530	yes	no	yes	yes	no	yes
rh-php56	yes	yes	yes	yes	no	yes
rh-php70	yes	no	yes	yes	no	yes
rh-php71	yes	no	yes	yes	no	yes
rh-php72	yes	no	yes	yes	no	yes
rh-php73	yes	no	yes	yes	no	yes
python27	yes	yes	yes	yes	no	yes
rh-python34	no	yes	no	yes	no	yes
rh-python35	yes	yes	yes	yes	no	yes
rh-python36	yes	yes	yes	yes	no	yes
rh-python38	yes	no	yes	yes	no	yes
rh-ror41	yes	yes	yes	yes	yes	yes

Language (Software Collection)	Database					
	MariaDB	MongoDB	MySQL	PostgreSQL	Redis	SQLite3
rh-ror42	yes	yes	yes	yes	no	yes
rh-ror50	yes	yes	yes	yes	no	yes
rh-ruby25	yes	yes	yes	yes	no	no
rh-ruby26	yes	yes	yes	yes	no	no
rh-ruby27	yes	yes	yes	yes	no	no
rh-ruby30	yes	no	yes	yes	no	yes

CHAPTER 5. MIGRATION

This chapter provides information on migrating to versions of components included in Red Hat Software Collections 3.7.

5.1. MIGRATING TO MARIADB 10.5

The rh-mariadb105 Software Collection is available for Red Hat Enterprise Linux 7, which includes **MariaDB 5.5** as the default MySQL implementation.

The rh-mariadb105 Software Collection does not conflict with the mysql or mariadb packages from the core systems. Unless the *-syspaths packages are installed (see below), it is possible to install the rh-mariadb105 Software Collection together with the mysql or mariadb packages. It is also possible to run both versions at the same time, however, the port number and the socket in the **my.cnf** files need to be changed to prevent these specific resources from conflicting. Additionally, it is possible to install the rh-mariadb105 Software Collection while the rh-mariadb103 Collection is still installed and even running.

The rh-mariadb105 Software Collection includes the rh-mariadb105-syspaths package, which installs packages that provide system-wide wrappers for binaries, scripts, manual pages, and other. After installing the rh-mariadb105*-syspaths packages, users are not required to use the **scl enable** command for correct functioning of the binaries and scripts provided by the rh-mariadb105* packages. Note that the *-syspaths packages conflict with the corresponding packages from the base Red Hat Enterprise Linux system and from the rh-mariadb103 and rh-mysql80 Software Collections. To find out more about syspaths, see the Red Hat Software Collections Packaging Guide.

The recommended migration path from **MariaDB 5.5** to **MariaDB 10.5** is to upgrade to **MariaDB 10.0** first, and then upgrade by one version successively. For details, see instructions in earlier Red Hat Software Collections Release Notes: Migrating to MariaDB 10.0, Migrating to MariaDB 10.1, Migrating to MariaDB 10.2, and Migrating to MariaDB 10.3. Note that **MariaDB 10.4** is not available as a Software Collection, so you must migrate directly from rh-mariadb103 to rh-mariadb105.
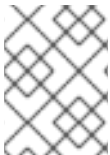
> **NOTE**
>
> The rh-mariadb105 Software Collection supports neither mounting over NFS nor dynamical registering using the **scl register** command.

5.1.1. Notable Differences Between the rh-mariadb103 and rh-mariadb105 Software Collections

Significant changes between **MariaDB 10.3** and **MariaDB 10.5** include:

- **MariaDB** now uses the **unix_socket** authentication plug-in by default. The plug-in enables users to use operating system credentials when connecting to **MariaDB** through the local Unix socket file.

- **MariaDB** adds **mariadb-*** named binaries and **mysql*** symbolic links pointing to the **mariadb-*** binaires. For example, the **mysqladmin**, **mysqlaccess**, and **mysqlshow** symlinks point to the **mariadb-admin**, **mariadb-access**, and **mariadb-show** binaries, respectively.

- The **SUPER** privilege has been split into several privileges to better align with each user role. As a result, certain statements have changed required privileges.

- In parallel replication, the **slave_parallel_mode** now defaults to **optimistic**.

- In the **InnoDB** storage engine, defaults of the following variables have been changed: **innodb_adaptive_hash_index** to **OFF** and **innodb_checksum_algorithm** to **full_crc32**.

- **MariaDB** now uses the **libedit** implementation of the underlying software managing the MariaDB command history (the **.mysql_history** file) instead of the previously used **readline** library. This change impacts users working directly with the **.mysql_history** file. Note that **.mysql_history** is a file managed by the **MariaDB** or **MySQL** applications, and users should not work with the file directly. The human-readable appearance is coincidental.

> **NOTE**
>
> To increase security, you can consider not maintaining a history file. To disable the command history recording:
>
> 1. Remove the **.mysql_history** file if it exists.
>
> 2. Use either of the following approaches:
>
>   - Set the **MYSQL_HISTFILE** variable to **/dev/null** and include this setting in any of your shell's startup files.
>
>   - Change the **.mysql_history** file to a symbolic link to **/dev/null**:
>
>   > **ln -s /dev/null $HOME/.mysql_history**

MariaDB Galera Cluster has been upgraded to version 4 with the following notable changes:

- **Galera** adds a new streaming replication feature, which supports replicating transactions of unlimited size. During an execution of streaming replication, a cluster replicates a transaction in small fragments.

- **Galera** now fully supports Global Transaction ID (GTID).

- The default value for the **wsrep_on** option in the **/etc/my.cnf.d/galera.cnf** file has changed from **1** to **0** to prevent end users from starting **wsrep** replication without configuring required additional options.

Changes to the PAM plug-in in **MariaDB 10.5** include:

- **MariaDB 10.5** adds a new version of the Pluggable Authentication Modules (PAM) plug-in. The PAM plug-in version 2.0 performs PAM authentication using a separate **setuid root** helper binary, which enables **MariaDB** to utilize additional PAM modules.

- The helper binary can be executed only by users in the **mysql** group. By default, the group contains only the **mysql** user. Red Hat recommends that administrators do not add more users to the **mysql** group to prevent password-guessing attacks without throttling or logging through this helper utility.

- In **MariaDB 10.5**, the Pluggable Authentication Modules (PAM) plug-in and its related files have been moved to a new subpackage, mariadb-pam. As a result, no new **setuid root** binary is introduced on systems that do not use PAM authentication for **MariaDB**.

- The rh-mariadb105-mariadb-pam package contains both PAM plug-in versions: version 2.0 is the default, and version 1.0 is available as the **auth_pam_v1** shared object library.

- The rh-mariadb105-mariadb-pam package is not installed by default with the **MariaDB** server. To make the PAM authentication plug-in available in **MariaDB 10.5**, install the rh-mariadb105-mariadb-pam package manually.

For more information, see the upstream documentation about changes in MariaDB 10.4 and changes in MariaDB 10.5. See also upstream information about upgrading to MariaDB 10.4 and upgrading to MariaDB 10.5.

5.1.2. Upgrading from the rh-mariadb103 to the rh-mariadb105 Software Collection
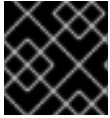
> **IMPORTANT**
>
> Prior to upgrading, back up all your data, including any MariaDB databases.

1. Stop the rh-mariadb103 database server if it is still running.

   Before stopping the server, set the **innodb_fast_shutdown** option to **0**, so that **InnoDB** performs a slow shutdown, including a full purge and insert buffer merge. Read more about this option in the upstream documentation. This operation can take a longer time than in case of a normal shutdown.

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

   Stop the rh-mariadb103 server:

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

2. Install the rh-mariadb105 Software Collection, including the subpackage providing the **mysql_upgrade** utility:

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

   Note that it is possible to install the rh-mariadb105 Software Collection while the rh-mariadb103 Software Collection is still installed because these Collections do not conflict.

3. Inspect configuration of rh-mariadb105, which is stored in the **/etc/opt/rh/rh-mariadb105/my.cnf** file and the **/etc/opt/rh/rh-mariadb105/my.cnf.d/** directory. Compare it with configuration of rh-mariadb103 stored in **/etc/opt/rh/rh-mariadb103/my.cnf** and **/etc/opt/rh/rh-mariadb103/my.cnf.d/** and adjust it if necessary.

4. All data of the rh-mariadb103 Software Collection is stored in the **/var/opt/rh/rh-mariadb103/lib/mysql/** directory unless configured differently. Copy the whole content of this directory to **/var/opt/rh/rh-mariadb105/lib/mysql/**. You can move the content but remember to back up your data before you continue to upgrade. Make sure the data is owned by the **mysql** user and SELinux context is correct.

5. Start the rh-mariadb105 database server:

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

6. Perform the data migration. Note that running the **mysql_upgrade** command is required due to upstream changes introduced in MDEV-14637.

> scl enable rh-mariadb105 mysql_upgrade

If the **root** user has a non-empty password defined (it should have a password defined), it is necessary to call the **mysql_upgrade** utility with the **-p** option and specify the password:

> scl enable rh-mariadb105 -- mysql_upgrade -p

Note that when the rh-mariadb105*-syspaths packages are installed, the **scl enable** command is not required. However, the *-syspaths packages conflict with the corresponding packages from the base Red Hat Enterprise Linux system and from the rh-mariadb103 and rh-mysql80 Software Collections.

5.2. MIGRATING TO MYSQL 8.0

The rh-mysql80 Software Collection is available for Red Hat Enterprise Linux 7, which includes **MariaDB 5.5** as the default **MySQL** implementation.

The rh-mysql80 Software Collection conflicts neither with the mysql or mariadb packages from the core systems nor with the rh-mysql* or rh-mariadb* Software Collections, unless the *-syspaths packages are installed (see below). It is also possible to run multiple versions at the same time; however, the port number and the socket in the **my.cnf** files need to be changed to prevent these specific resources from conflicting.

Note that it is possible to upgrade to **MySQL 8.0** only from **MySQL 5.7**. If you need to upgrade from an earlier version, upgrade to **MySQL 5.7** first. For instructions, see Migration to MySQL 5.7 .

5.2.1. Notable Differences Between MySQL 5.7 and MySQL 8.0

Differences Specific to the rh-mysql80 Software Collection

- The **MySQL 8.0** server provided by the rh-mysql80 Software Collection is configured to use **mysql_native_password** as the default authentication plug-in because client tools and libraries in Red Hat Enterprise Linux 7 are incompatible with the **caching_sha2_password** method, which is used by default in the upstream **MySQL 8.0** version.

  To change the default authentication plug-in to **caching_sha2_password**, edit the **/etc/opt/rh/rh-mysql80/my.cnf.d/mysql-default-authentication-plugin.cnf** file as follows:

  > [mysqld]
  > default_authentication_plugin=caching_sha2_password

 For more information about the **caching_sha2_password** authentication plug-in, see the upstream documentation .

- The rh-mysql80 Software Collection includes the rh-mysql80-syspaths package, which installs the rh-mysql80-mysql-config-syspaths, rh-mysql80-mysql-server-syspaths, and rh-mysql80-mysql-syspaths packages. These subpackages provide system-wide wrappers for binaries, scripts, manual pages, and other. After installing the rh-mysql80*-syspaths packages, users are not required to use the **scl enable** command for correct functioning of the binaries and scripts provided by the rh-mysql80* packages. Note that the *-syspaths packages conflict with the corresponding packages from the base Red Hat Enterprise Linux system and from the rh-mariadb103 and rh-mariadb105 Software Collections. To find out more about syspaths, see the Red Hat Software Collections Packaging Guide .

General Changes in MySQL 8.0

- Binary logging is enabled by default during the server startup. The **log_bin** system variable is now set to **ON** by default even if the **--log-bin** option has not been specified. To disable binary logging, specify the **--skip-log-bin** or **--disable-log-bin** option at startup.

- For a **CREATE FUNCTION** statement to be accepted, at least one of the **DETERMINISTIC**, **NO SQL**, or **READS SQL DATA** keywords must be specified explicitly, otherwise an error occurs.

- Certain features related to account management have been removed. Namely, using the **GRANT** statement to modify account properties other than privilege assignments, such as authentication, SSL, and resource-limit, is no longer possible. To establish the mentioned properties at account-creation time, use the **CREATE USER** statement. To modify these properties, use the **ALTER USER** statement.

- Certain SSL-related options have been removed on the client-side. Use the **--ssl-mode=REQUIRED** option instead of **--ssl=1** or **--enable-ssl**. Use the **--ssl-mode=DISABLED** option instead of **--ssl=0**, **--skip-ssl**, or **--disable-ssl**. Use the **--ssl-mode=VERIFY_IDENTITY** option instead of **--ssl-verify-server-cert** options. Note that these option remains unchanged on the server side.

- The default character set has been changed from **latin1** to **utf8mb4**.

- The **utf8** character set is currently an alias for **utf8mb3** but in the future, it will become a reference to **utf8mb4**. To prevent ambiguity, specify **utf8mb4** explicitly for character set references instead of **utf8**.

- Setting user variables in statements other than **SET** has been deprecated.

- The **log_syslog** variable, which previously configured error logging to the system logs, has been removed.

- Certain incompatible changes to spatial data support have been introduced.

- The deprecated **ASC** or **DESC** qualifiers for **GROUP BY** clauses have been removed. To produce a given sort order, provide an **ORDER BY** clause.

For detailed changes in **MySQL 8.0** compared to earlier versions, see the upstream documentation: What Is New in MySQL 8.0 and Changes Affecting Upgrades to MySQL 8.0 .

5.2.2. Upgrading to the rh-mysql80 Software Collection
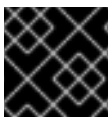
> **IMPORTANT**
>
> Prior to upgrading, back-up all your data, including any MySQL databases.

1. Install the rh-mysql80 Software Collection.

   yum install rh-mysql80-mysql-server

2. Inspect the configuration of rh-mysql80, which is stored in the **/etc/opt/rh/rh-mysql80/my.cnf** file and the **/etc/opt/rh/rh-mysql80/my.cnf.d/** directory. Compare it with the configuration of rh-mysql57 stored in **/etc/opt/rh/rh-mysql57/my.cnf** and **/etc/opt/rh/rh-mysql57/my.cnf.d/** and adjust it if necessary.

3. Stop the rh-mysql57 database server, if it is still running.

> **systemctl stop rh-mysql57-mysqld.service**

4. All data of the rh-mysql57 Software Collection is stored in the **/var/opt/rh/rh-mysql57/lib/mysql/** directory. Copy the whole content of this directory to **/var/opt/rh/rh-mysql80/lib/mysql/**. You can also move the content but remember to back up your data before you continue to upgrade.

5. Start the rh-mysql80 database server.

> **systemctl start rh-mysql80-mysqld.service**

6. Perform the data migration.

> **scl enable rh-mysql80 mysql_upgrade**

If the **root** user has a non-empty password defined (it should have a password defined), it is necessary to call the **mysql_upgrade** utility with the **-p** option and specify the password.

> **scl enable rh-mysql80 -- mysql_upgrade -p**

Note that when the rh-mysql80*-syspaths packages are installed, the **scl enable** command is not required. However, the *-syspaths packages conflict with the corresponding packages from the base Red Hat Enterprise Linux system and from the rh-mariadb103 and rh-mariadb105 Software Collections.

5.3. MIGRATING TO POSTGRESQL 13

Red Hat Software Collections 3.7 is distributed with **PostgreSQL 13**, available only for Red Hat Enterprise Linux 7. The rh-postgresql13 Software Collection can be safely installed on the same machine in parallel with the base Red Hat Enterprise Linux system version of **PostgreSQL** or any **PostgreSQL** Software Collection. It is also possible to run more than one version of **PostgreSQL** on a machine at the same time, but you need to use different ports or IP addresses and adjust SELinux policy.

The rh-postgresql13 Software Collection includes the rh-postgresql13-syspaths package, which installs packages that provide system-wide wrappers for binaries, scripts, manual pages, and other. After installing the rh-postgresql13*-syspaths packages, users are not required to use the **scl enable** command for correct functioning of the binaries and scripts provided by the rh-postgresql13* packages. Note that the *-syspaths packages conflict with the corresponding packages from the base Red Hat Enterprise Linux system. To find out more about syspaths, see the Red Hat Software Collections Packaging Guide.

> IMPORTANT
>
> Before migrating to **PostgreSQL 13**, see the upstream compatibility notes for PostgreSQL 13.
>
> In case of upgrading the **PostgreSQL** database in a container, see the container-specific instructions.

The following table provides an overview of different paths in a Red Hat Enterprise Linux 7 system version of **PostgreSQL** provided by the postgresql package, and in the rh-postgresql12 and rh-postgresql13 Software Collections.

Table 5.1. Diferences in thePostgreSQL paths

Content	postgresql	rh-postgresql12	rh-postgresql13
Executables	/usr/bin/	/opt/rh/rh-postgresql12/root/usr/bin/	/opt/rh/rh-postgresql13/root/usr/bin/
Libraries	/usr/lib64/	/opt/rh/rh-postgresql12/root/usr/lib64/	/opt/rh/rh-postgresql13/root/usr/lib64/
Documentation	/usr/share/doc/postgresql/html/	/opt/rh/rh-postgresql12/root/usr/share/doc/postgresql/html/	/opt/rh/rh-postgresql13/root/usr/share/doc/postgresql/html/
PDF documentation	/usr/share/doc/postgresql-docs/	/opt/rh/rh-postgresql12/root/usr/share/doc/postgresql-docs/	/opt/rh/rh-postgresql13/root/usr/share/doc/postgresql-docs/
Contrib documentation	/usr/share/doc/postgresql-contrib/	/opt/rh/rh-postgresql12/root/usr/share/doc/postgresql-contrib/	/opt/rh/rh-postgresql13/root/usr/share/doc/postgresql-contrib/
Source	not installed	not installed	not installed
Data	/var/lib/pgsql/data/	/var/opt/rh/rh-postgresql12/lib/pgsql/data/	/var/opt/rh/rh-postgresql13/lib/pgsql/data/
Backup area	/var/lib/pgsql/backups/	/var/opt/rh/rh-postgresql12/lib/pgsql/backups/	/var/opt/rh/rh-postgresql13/lib/pgsql/backups/
Templates	/usr/share/pgsql/	/opt/rh/rh-postgresql12/root/usr/share/pgsql/	/opt/rh/rh-postgresql13/root/usr/share/pgsql/
Procedural Languages	/usr/lib64/pgsql/	/opt/rh/rh-postgresql12/root/usr/lib64/pgsql/	/opt/rh/rh-postgresql13/root/usr/lib64/pgsql/
Development Headers	/usr/include/pgsql/	/opt/rh/rh-postgresql12/root/usr/include/pgsql/	/opt/rh/rh-postgresql13/root/usr/include/pgsql/

Content	postgresql	rh-postgresql12	rh-postgresql13
Other shared data	/usr/share/pgsql/	/opt/rh/rh-postgresql12/root/usr/share/pgsql/	/opt/rh/rh-postgresql13/root/usr/share/pgsql/
Regression tests	/usr/lib64/pgsql/test/regress/ (in the –test package)	/opt/rh/rh-postgresql12/root/usr/lib64/pgsql/test/regress/ (in the -test package)	/opt/rh/rh-postgresql13/root/usr/lib64/pgsql/test/regress/ (in the -test package)

5.3.1. Migrating from a Red Hat Enterprise Linux System Version of PostgreSQL to the PostgreSQL 13 Software Collection

Red Hat Enterprise Linux 7 is distributed with **PostgreSQL 9.2**. To migrate your data from a Red Hat Enterprise Linux system version of **PostgreSQL** to the rh-postgresql13 Software Collection, you can either perform a fast upgrade using the **pg_upgrade** tool (recommended), or dump the database data into a text file with SQL commands and import it in the new database. Note that the second method is usually significantly slower and may require manual fixes; see the PostgreSQL documentation for more information about this upgrade method.
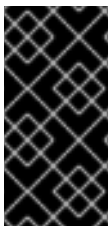
IMPORTANT

Before migrating your data from a Red Hat Enterprise Linux system version of PostgreSQL to PostgreSQL 13, make sure that you back up all your data, including the PostgreSQL database files, which are *by default* located in the **/var/lib/pgsql/data/** directory.

Procedure 5.1. Fast Upgrade Using the **pg_upgrade** Tool

To perform a fast upgrade of your PostgreSQL server, complete the following steps:

1. Stop the old PostgreSQL server to ensure that the data is not in an inconsistent state. To do so, type the following at a shell prompt as **root**:

   systemctl stop postgresql.service

   To verify that the server is not running, type:

   systemctl status postgresql.service

2. Verify that the old directory **/var/lib/pgsql/data/** exists:

   file /var/lib/pgsql/data/

   and back up your data.

3. Verify that the new data directory **/var/opt/rh/rh-postgresql13/lib/pgsql/data/** does not exist:

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

If you are running a fresh installation of **PostgreSQL 13**, this directory should not be present in your system. If it is, back it up by running the following command as **root**:

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

4. Upgrade the database data for the new server by running the following command as **root**:

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

Alternatively, you can use the **/opt/rh/rh-postgresql13/root/usr/bin/postgresql-setup --upgrade** command.

Note that you can use the **--upgrade-from** option for upgrading from different versions of **PostgreSQL**. The list of possible upgrade scenarios is available using the **--upgrade-ids** option.

It is recommended that you read the resulting **/var/lib/pgsql/upgrade_rh-postgresql13-postgresql.log** log file to find out if any problems occurred during the upgrade.

5. Start the new server as **root**:

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

It is also advised that you run the **analyze_new_cluster.sh** script as follows:

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

6. Optionally, you can configure the PostgreSQL 13 server to start automatically at boot time. To disable the old system PostgreSQL server, type the following command as **root**:

```
chkconfig postgresql off
```

To enable the PostgreSQL 13 server, type as **root**:

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

7. If your configuration differs from the default one, make sure to update configuration files, especially the **/var/opt/rh/rh-postgresql13/lib/pgsql/data/pg_hba.conf** configuration file. Otherwise only the **postgres** user will be allowed to access the database.

Procedure 5.2. Performing a Dump and Restore Upgrade

To perform a dump and restore upgrade of your PostgreSQL server, complete the following steps:

1. Ensure that the old PostgreSQL server is running by typing the following at a shell prompt as **root**:

```
systemctl start postgresql.service
```

2. Dump all data in the PostgreSQL database into a script file. As **root**, type:

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

3. Stop the old server by running the following command as **root**:

   ```
   systemctl stop postgresql.service
   ```

4. Initialize the data directory for the new server as **root**:

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

5. Start the new server as **root**:

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

6. Import data from the previously created SQL file:

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

7. Optionally, you can configure the PostgreSQL 13 server to start automatically at boot time. To disable the old system PostgreSQL server, type the following command as **root**:

   ```
   chkconfig postgresql off
   ```

   To enable the PostgreSQL 13 server, type as **root**:

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

8. If your configuration differs from the default one, make sure to update configuration files, especially the **/var/opt/rh/rh-postgresql13/lib/pgsql/data/pg_hba.conf** configuration file. Otherwise only the **postgres** user will be allowed to access the database.

5.3.2. Migrating from the PostgreSQL 12 Software Collection to the PostgreSQL 13 Software Collection

To migrate your data from the rh-postgresql12 Software Collection to the rh-postgresql13 Collection, you can either perform a fast upgrade using the **pg_upgrade** tool (recommended), or dump the database data into a text file with SQL commands and import it in the new database. Note that the second method is usually significantly slower and may require manual fixes; see the PostgreSQL documentation for more information about this upgrade method.

IMPORTANT

Before migrating your data from **PostgreSQL 12** to **PostgreSQL 13**, make sure that you back up all your data, including the PostgreSQL database files, which are by default located in the **/var/opt/rh/rh-postgresql12/lib/pgsql/data/** directory.

Procedure 5.3. Fast Upgrade Using the **pg_upgrade** Tool

To perform a fast upgrade of your PostgreSQL server, complete the following steps:

1. Stop the old PostgreSQL server to ensure that the data is not in an inconsistent state. To do so, type the following at a shell prompt as **root**:

> **systemctl stop rh-postgresql12-postgresql.service**

To verify that the server is not running, type:

> **systemctl status rh-postgresql12-postgresql.service**

2. Verify that the old directory **/var/opt/rh/rh-postgresql12/lib/pgsql/data/** exists:

> **file /var/opt/rh/rh-postgresql12/lib/pgsql/data/**

and back up your data.

3. Verify that the new data directory **/var/opt/rh/rh-postgresql13/lib/pgsql/data/** does not exist:

> **file /var/opt/rh/rh-postgresql13/lib/pgsql/data/**

If you are running a fresh installation of **PostgreSQL 13**, this directory should not be present in your system. If it is, back it up by running the following command as **root**:

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

4. Upgrade the database data for the new server by running the following command as **root**:

> **scl enable rh-postgresql13 -- postgresql-setup --upgrade --upgrade-from=rh-postgresql12-postgresql**

Alternatively, you can use the **/opt/rh/rh-postgresql13/root/usr/bin/postgresql-setup --upgrade --upgrade-from=rh-postgresql12-postgresql** command.

Note that you can use the **--upgrade-from** option for upgrading from different versions of PostgreSQL. The list of possible upgrade scenarios is available using the **--upgrade-ids** option.

It is recommended that you read the resulting **/var/lib/pgsql/upgrade_rh-postgresql13-postgresql.log** log file to find out if any problems occurred during the upgrade.

5. Start the new server as **root**:

> **systemctl start rh-postgresql13-postgresql.service**

It is also advised that you run the **analyze_new_cluster.sh** script as follows:

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

6. Optionally, you can configure the PostgreSQL 13 server to start automatically at boot time. To disable the old PostgreSQL 12 server, type the following command as **root**:

> **chkconfig rh-postgresql12-postgreqsql off**

To enable the PostgreSQL 13 server, type as **root**:

> **chkconfig rh-postgresql13-postgresql on**

7. If your configuration differs from the default one, make sure to update configuration files, especially the **/var/opt/rh/rh-postgresql13/lib/pgsql/data/pg_hba.conf** configuration file. Otherwise only the **postgres** user will be allowed to access the database.

Procedure 5.4. Performing a Dump and Restore Upgrade

To perform a dump and restore upgrade of your PostgreSQL server, complete the following steps:

1. Ensure that the old PostgreSQL server is running by typing the following at a shell prompt as **root**:

   > **systemctl start rh-postgresql12-postgresql.service**

2. Dump all data in the PostgreSQL database into a script file. As **root**, type:

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

3. Stop the old server by running the following command as **root**:

   > **systemctl stop rh-postgresql12-postgresql.service**

4. Initialize the data directory for the new server as **root**:

   > **scl enable rh-postgresql13 -- postgresql-setup initdb**

5. Start the new server as **root**:

   > **systemctl start rh-postgresql13-postgresql.service**

6. Import data from the previously created SQL file:

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

7. Optionally, you can configure the PostgreSQL 13 server to start automatically at boot time. To disable the old PostgreSQL 12 server, type the following command as **root**:

   > **chkconfig rh-postgresql12-postgresql off**

   To enable the PostgreSQL 13 server, type as **root**:

   > **chkconfig rh-postgresql13-postgresql on**

8. If your configuration differs from the default one, make sure to update configuration files, especially the **/var/opt/rh/rh-postgresql13/lib/pgsql/data/pg_hba.conf** configuration file. Otherwise only the **postgres** user will be allowed to access the database.

5.4. MIGRATING TO NGINX 1.18

The root directory for the rh-nginx118 Software Collection is located in **/opt/rh/rh-nginx118/root/**. The error log is stored in **/var/opt/rh/rh-nginx118/log/nginx** by default.

Configuration files are stored in the **/etc/opt/rh/rh-nginx118/nginx/** directory. Configuration files in **nginx 1.18** have the same syntax and largely the same format as previous **nginx** Software Collections.

Configuration files (with a **.conf** extension) in the **/etc/opt/rh/rh-nginx118/nginx/default.d/** directory are included in the default server block configuration for port **80**.
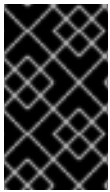
> **IMPORTANT**
>
> Before upgrading from **nginx 1.16** to **nginx 1.18**, back up all your data, including web pages located in the **/opt/rh/nginx116/root/** tree and configuration files located in the **/etc/opt/rh/nginx116/nginx/** tree.

If you have made any specific changes, such as changing configuration files or setting up web applications, in the **/opt/rh/nginx116/root/** tree, replicate those changes in the new **/opt/rh/rh-nginx118/root/** and **/etc/opt/rh/rh-nginx118/nginx/** directories, too.

You can use this procedure to upgrade directly from **nginx 1.12** or **nginx 1.14** to **nginx 1.18**. Use the appropriate paths in this case.

For the official **nginx** documentation, refer to http://nginx.org/en/docs/.

5.5. MIGRATING TO REDIS 5

Redis 3.2, provided by the rh-redis32 Software Collection, is mostly a strict subset of **Redis 4.0**, which is mostly a strict subset of **Redis 5.0**. Therefore, no major issues should occur when upgrading from version 3.2 to version 5.0.

To upgrade a **Redis** Cluster to version 5.0, a mass restart of all the instances is needed.

Compatibility Notes

- The format of RDB files has been changed. **Redis 5** is able to read formats of all the earlier versions, but earlier versions are incapable of reading the **Redis 5** format.

- Since version 4.0, the **Redis** Cluster bus protocol is no longer compatible with **Redis 3.2**.

- For minor non-backward compatible changes, see the upstream release notes for version 4.0 and version 5.0.

CHAPTER 6. ADDITIONAL RESOURCES

This chapter provides references to other relevant sources of information about Red Hat Software Collections 3.7 and Red Hat Enterprise Linux.

6.1. RED HAT PRODUCT DOCUMENTATION

The following documents are directly or indirectly relevant to this book:

- Red Hat Software Collections 3.7 Packaging Guide — The *Packaging Guide* for Red Hat Software Collections explains the concept of Software Collections, documents the **scl** utility, and provides a detailed explanation of how to create a custom Software Collection or extend an existing one.

- Red Hat Developer Toolset 10.1 Release Notes — The *Release Notes* for Red Hat Developer Toolset document known problems, possible issues, changes, and other important information about this Software Collection.

- Red Hat Developer Toolset 10.1 User Guide — The *User Guide* for Red Hat Developer Toolset contains more information about installing and using this Software Collection.

- Using Red Hat Software Collections Container Images — This book provides information on how to use container images based on Red Hat Software Collections. The available container images include applications, daemons, databases, as well as the Red Hat Developer Toolset container images. The images can be run on Red Hat Enterprise Linux 7 Server and Red Hat Enterprise Linux Atomic Host.

- Getting Started with Containers — This guide contains a comprehensive overview of information about building and using container images on Red Hat Enterprise Linux 7 and Red Hat Enterprise Linux Atomic Host.

- Using and Configuring Red Hat Subscription Manager — The *Using and Configuring Red Hat Subscription Manager* book provides detailed information on how to register Red Hat Enterprise Linux systems, manage subscriptions, and view notifications for the registered systems.

- Red Hat Enterprise Linux 7 System Administrator's Guide — The *System Administrator's Guide* for Red Hat Enterprise Linux 7 provides information on deployment, configuration, and administration of this system.

6.2. RED HAT DEVELOPERS

- Red Hat Developer Program — The *Red Hat Developers* community portal.

- Overview of Red Hat Software Collections on Red Hat Developers — The *Red Hat Developers* portal provides a number of tutorials to get you started with developing code using different development technologies. This includes the Node.js, Perl, PHP, Python, and Ruby Software Collections.

- Red Hat Developer Blog — The *Red Hat Developer Blog* contains up-to-date information, best practices, opinion, product and program announcements as well as pointers to sample code and other resources for those who are designing and developing applications based on Red Hat technologies.

APPENDIX A. REVISION HISTORY

Revision 3.7-5	**Wed Dec 20 2023**	**Lenka Špačková**

Fixed broken links and anchors.

Revision 3.7-4	**Fri Nov 12 2021**	**Lenka Špačková**

Updated Section 4.3, "Database Connectors".

Revision 3.7-3	**Tue Aug 24 2021**	**Lenka Špačková**

Added two notes related to asynchronous changes in Python due to CVE fixes (Section 1.6, "Other Notes").
Extended Section 1.3.5, "Changes in Ruby".

Revision 3.7-2	**Mon Jul 12 2021**	**Lenka Špačková**

Added a note about the **.mysql_history** file.

Revision 3.7-1	**Thu Jun 03 2021**	**Lenka Špačková**

Release of Red Hat Software Collections 3.7 Release Notes.

Revision 3.7-0	**Mon May 03 2021**	**Lenka Špačková**

Release of Red Hat Software Collections 3.7 Beta Release Notes.