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 Hradilek
Red Hat Customer Content Services
jhradilek@redhat.com

Eliška Slobodová
Red Hat Customer Content Services
Legal Notice

Copyright © 2021 Red Hat, Inc.

This document is licensed by Red Hat under the Creative Commons Attribution-ShareAlike 3.0 Unported License. If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Abstract

# Table of Contents

**MAKING OPEN SOURCE MORE INCLUSIVE** ................................................................. 3

**CHAPTER 1. RED HAT SOFTWARE COLLECTIONS 3.7** .................................................. 4
  1.1. ABOUT RED HAT SOFTWARE COLLECTIONS ................................................. 4
    1.1.1. Red Hat Developer Toolset ............................................................... 4
  1.2. MAIN FEATURES ....................................................................................... 4
  1.3. CHANGES IN RED HAT SOFTWARE COLLECTIONS 3.7 .......................... 15
    1.3.1. Overview .......................................................................................... 15
      Architectures ......................................................................................... 15
      New Software Collections ................................................................ 15
      Updated Software Collections ......................................................... 16
      Red Hat Software Collections Container Images .......................... 16
    1.3.2. Changes in Red Hat Developer Toolset ........................................... 16
    1.3.3. Changes in MariaDB ....................................................................... 17
    1.3.4. Changes in PostgreSQL ................................................................. 18
    1.3.5. Changes in Ruby ........................................................................... 19
    1.3.6. Changes in JDK Mission Control ................................................. 20
  1.4. COMPATIBILITY INFORMATION .............................................................. 20
  1.5. KNOWN ISSUES ..................................................................................... 20
  1.6. OTHER NOTES ....................................................................................... 23
  1.7. DEPRECATED FUNCTIONALITY ............................................................... 25

**CHAPTER 2. INSTALLATION** .................................................................................... 27
  2.1. GETTING ACCESS TO RED HAT SOFTWARE COLLECTIONS .............. 27
    2.1.1. Using Red Hat Subscription Management ...................................... 27
    2.1.2. Packages from the Optional Repository ........................................ 28
  2.2. INSTALLING RED HAT SOFTWARE COLLECTIONS .................................. 30
    2.2.1. Installing Individual Software Collections .................................. 30
    2.2.2. Installing Optional Packages ....................................................... 30
    2.2.3. Installing Debugging Information ................................................ 31
  2.3. UNINSTALLING RED HAT SOFTWARE COLLECTIONS ....................... 31
  2.4. REBUILDING RED HAT SOFTWARE COLLECTIONS .............................. 31

**CHAPTER 3. USAGE** .............................................................................................. 33
  3.1. USING RED HAT SOFTWARE COLLECTIONS ......................................... 33
    3.1.1. Running an Executable from a Software Collection .................... 33
    3.1.2. Running a Shell Session with a Software Collection as Default ... 33
    3.1.3. Running a System Service from a Software Collection ................ 34
  3.2. ACCESSING A MANUAL PAGE FROM A SOFTWARE COLLECTION ....... 34
  3.3. DEPLOYING APPLICATIONS THAT USE RED HAT SOFTWARE COLLECTIONS 34
  3.4. RED HAT SOFTWARE COLLECTIONS CONTAINER IMAGES .................. 35

**CHAPTER 4. SPECIFICS OF INDIVIDUAL SOFTWARE COLLECTIONS** ......................... 37
  4.1. RED HAT DEVELOPER TOOLSET ............................................................. 37
  4.2. MAVEN ..................................................................................................... 37
  4.3. DATABASE CONNECTORS ...................................................................... 37

**CHAPTER 5. MIGRATION** ....................................................................................... 40
  5.1. MIGRATING TO MARIADB 10.5 ............................................................... 40
    5.1.1. Notable Differences Between the rh-mariadb103 and rh-mariadb105 Software Collections 40
    5.1.2. Upgrading from the rh-mariadb103 to the rh-mariadb105 Software Collection 42
  5.2. MIGRATING TO MYSQL 8.0 .................................................................... 43
5.2.1. Notable Differences Between MySQL 5.7 and MySQL 8.0 43
   Differences Specific to the rh-mysql80 Software Collection 43
   General Changes in MySQL 8.0 44
5.2.2. Upgrading to the rh-mysql80 Software Collection 44
5.3. MIGRATING TO POSTGRESQL 13 45
   5.3.1. Migrating from a Red Hat Enterprise Linux System Version of PostgreSQL to the PostgreSQL 13 Software Collection 47
   5.3.2. Migrating from the PostgreSQL 12 Software Collection to the PostgreSQL 13 Software Collection 49
5.4. MIGRATING TO NGINX 1.18 51
5.5. MIGRATING TO REDIS 5 52
   Compatibility Notes 52

CHAPTER 6. ADDITIONAL RESOURCES ................................................................. 53
   6.1. RED HAT PRODUCT DOCUMENTATION 53
   6.2. RED HAT DEVELOPERS 53

APPENDIX A. REVISION HISTORY ................................................................. 54
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
<table>
<thead>
<tr>
<th>Component</th>
<th>Software Collection</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Hat Developer Toolset 10.1</td>
<td>devtoolset-10</td>
<td>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.</td>
</tr>
<tr>
<td>Perl 5.30.1</td>
<td>rh-perl530</td>
<td>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.</td>
</tr>
<tr>
<td>PHP 7.3.20</td>
<td>rh-php73</td>
<td>A release of PHP 7.3 with PEAR 1.10.9, APCu 5.1.17, and the Xdebug extension.</td>
</tr>
<tr>
<td>Python 2.7.18</td>
<td>python27</td>
<td>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.</td>
</tr>
<tr>
<td>Python 3.8.6</td>
<td>rh-python38</td>
<td>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.</td>
</tr>
<tr>
<td>Component</td>
<td>Software Collection</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Ruby 2.6.7</td>
<td>rh-ruby26</td>
<td>A release of Ruby 2.6. This version provides multiple performance improvements and new features, such as <strong>endless ranges</strong>, the <strong>Binding#source_location</strong> method, and the <strong>$SAFE process global state</strong>. Ruby 2.6.0 maintains source-level backward compatibility with Ruby 2.5.</td>
</tr>
<tr>
<td>Ruby 2.7.3</td>
<td>rh-ruby27</td>
<td>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.</td>
</tr>
<tr>
<td>Ruby 3.0.1</td>
<td>rh-ruby30</td>
<td>A release of Ruby 3.0. This version provides multiple performance improvements and new features, such as <strong>Ractor</strong>, <strong>Fiber Scheduler</strong> and the <strong>RBS language</strong>. Ruby 3.0 maintains source-level backward compatibility with Ruby 2.7.</td>
</tr>
<tr>
<td>MariaDB 10.3.27</td>
<td>rh-mariadb103</td>
<td>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 <strong>system-versioned tables</strong>, <strong>invisible columns</strong>, a <strong>new instant ADD COLUMN operation for InnoDB</strong>, and a <strong>JDBC connector for MariaDB and MySQL</strong>.</td>
</tr>
<tr>
<td>MariaDB 10.5.9</td>
<td>rh-mariadb105</td>
<td>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.</td>
</tr>
<tr>
<td>MySQL 8.0.21</td>
<td>rh-mysql80</td>
<td>A release of the MySQL server, which introduces a number of new security and account management features and enhancements.</td>
</tr>
<tr>
<td>PostgreSQL 10.15</td>
<td>rh-postgresql10</td>
<td>A release of PostgreSQL, which includes a significant performance improvement and a number of new features, such as <strong>logical replication using the publish and subscribe keywords</strong>, or stronger password authentication based on the <strong>SCRAM-SHA-256 mechanism</strong>.</td>
</tr>
<tr>
<td>PostgreSQL 12.5</td>
<td>rh-postgresql12</td>
<td>A release of PostgreSQL, which provides the <strong>pgaudit extension</strong>, various enhancements to partitioning and parallelism, support for the SQL/JSON path language, and performance improvements.</td>
</tr>
<tr>
<td>Component</td>
<td>Software Collection</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PostgreSQL 13.2</td>
<td>rh-postgresql13</td>
<td>A release of PostgreSQL, which enables improved query planning and introduces various performance improvements and two new packages, pg_repack and plpython3.</td>
</tr>
<tr>
<td>Node.js 12.21.0</td>
<td>rh-nodejs12</td>
<td>A release of Node.js with V8 engine version 7.6, support for ES6 modules, and improved support for native modules.</td>
</tr>
<tr>
<td>Node.js 14.16.0</td>
<td>rh-nodejs14</td>
<td>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.</td>
</tr>
<tr>
<td>nginx 1.16.1</td>
<td>rh-nginx116</td>
<td>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.</td>
</tr>
<tr>
<td>nginx 1.18.0</td>
<td>rh-nginx118</td>
<td>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.</td>
</tr>
<tr>
<td>Apache httpd 2.4.34</td>
<td>httpd24</td>
<td>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.</td>
</tr>
<tr>
<td>Varnish Cache 6.0.6</td>
<td>rh-varnish6</td>
<td>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.</td>
</tr>
<tr>
<td>Maven 3.6.1</td>
<td>rh-maven36</td>
<td>A release of Maven, a software project management and comprehension tool. This release provides various enhancements and bug fixes.</td>
</tr>
</tbody>
</table>
Git 2.27.0

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

A release of Redis 5.0, a persistent key-value database. Redis now provides `redis-trib`, a cluster management tool.

HAProx 1.8.24

A release of HAProx 1.8, a reliable, high-performance network load balancer for TCP and HTTP-based applications.

JDK Mission Control 8.0.0

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

<table>
<thead>
<tr>
<th>Component</th>
<th>Software Collection</th>
<th>Availability</th>
<th>Architectures supported on RHEL7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Components New in Red Hat Software Collections 3.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MariaDB 10.5.9</td>
<td>rh-mariadb105</td>
<td>RHEL7</td>
<td>x86_64, s390x, ppc64le</td>
</tr>
<tr>
<td>PostgreSQL 13.2</td>
<td>rh-postgresql13</td>
<td>RHEL7</td>
<td>x86_64, s390x, ppc64le</td>
</tr>
<tr>
<td>Ruby 3.0.1</td>
<td>rh-ruby30</td>
<td>RHEL7</td>
<td>x86_64, s390x, ppc64le</td>
</tr>
<tr>
<td>Components Updated in Red Hat Software Collections 3.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------------------------------------------------</td>
<td>------------------</td>
<td>--------</td>
<td>------------------------</td>
</tr>
<tr>
<td>Red Hat Developer Toolset 10.1</td>
<td>devtoolset-10</td>
<td>RHEL7</td>
<td>x86_64, s390x, ppc64,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JDK Mission Control 8.0.0</td>
<td>rh-jmc</td>
<td>RHEL7</td>
<td>x86_64</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ruby 2.7.3</td>
<td>rh-ruby27</td>
<td>RHEL7</td>
<td>x86_64, s390x, aarch64,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ruby 2.6.7</td>
<td>rh-ruby26</td>
<td>RHEL7</td>
<td>x86_64, s390x, aarch64,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Components Last Updated in Red Hat Software Collections 3.5</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Hat Developer Toolset 9.1</td>
<td>devtoolset-9</td>
<td>RHEL7</td>
<td>x86_64, s390x, aarch64,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Python 3.8.6</td>
<td>rh-python38</td>
<td>RHEL7</td>
<td>x86_64, s390x, aarch64,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Varnish Cache 6.0.6</td>
<td>rh-varnish6</td>
<td>RHEL7</td>
<td>x86_64, s390x, aarch64,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Components Last Updated in Red Hat Software Collections 3.5

<table>
<thead>
<tr>
<th>Component</th>
<th>Container Name</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache httpd 2.4.34 (the last update for RHEL6)</td>
<td>httpd24 (RHEL6)*</td>
<td>RHEL6, x86_64</td>
</tr>
</tbody>
</table>

### Components Last Updated in Red Hat Software Collections 3.4

<table>
<thead>
<tr>
<th>Component</th>
<th>Container Name</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node.js 12.21.0</td>
<td>rh-nodejs12</td>
<td>RHEL7, x86_64, s390x, aarch64, ppc64le</td>
</tr>
<tr>
<td>nginx 1.16.1</td>
<td>rh-nginx116</td>
<td>RHEL7, x86_64, s390x, aarch64, ppc64le</td>
</tr>
<tr>
<td>PostgreSQL 12.5</td>
<td>rh-postgresql12</td>
<td>RHEL7, x86_64, s390x, aarch64, ppc64le</td>
</tr>
<tr>
<td>Maven 3.6.1</td>
<td>rh-maven36</td>
<td>RHEL7, x86_64, s390x, aarch64, ppc64le</td>
</tr>
</tbody>
</table>

### Components Last Updated in Red Hat Software Collections 3.3

<table>
<thead>
<tr>
<th>Component</th>
<th>Container Name</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Hat Developer Toolset 8.1</td>
<td>devtoolset-8*</td>
<td>RHEL6, RHEL7, x86_64, s390x, aarch64, ppc64, ppc64le</td>
</tr>
<tr>
<td>MariaDB 10.3.27</td>
<td>rh-mariadb103</td>
<td>RHEL7, x86_64, s390x, aarch64, ppc64le</td>
</tr>
<tr>
<td>Redis 5.0.5</td>
<td>rh-redis5</td>
<td>RHEL7, x86_64, s390x, aarch64, ppc64le</td>
</tr>
</tbody>
</table>

### Components Last Updated in Red Hat Software Collections 3.2

<table>
<thead>
<tr>
<th>Component</th>
<th>Container Name</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHP 7.2.24</td>
<td>rh-php72*</td>
<td>RHEL7, x86_64, s390x, aarch64, ppc64le</td>
</tr>
<tr>
<td>MySQL 8.0.21</td>
<td>rh-mysql80</td>
<td>RHEL7, x86_64, s390x, aarch64, ppc64le</td>
</tr>
<tr>
<td>Node.js 10.21.0</td>
<td>rh-nodejs10*</td>
<td>RHEL7, x86_64, s390x, aarch64, ppc64le</td>
</tr>
<tr>
<td>nginx 1.14.1</td>
<td>rh-nginx114*</td>
<td>RHEL7, x86_64, s390x, aarch64, ppc64le</td>
</tr>
</tbody>
</table>
### Components Last Updated in Red Hat Software Collections 3.2

<table>
<thead>
<tr>
<th>Component</th>
<th>RHN ID</th>
<th>OS Support</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Git 2.18.4</td>
<td>rh-git218*</td>
<td>RHEL7</td>
<td>x86_64, s390x, aarch64, ppc64le</td>
</tr>
</tbody>
</table>

### Components Last Updated in Red Hat Software Collections 3.1

<table>
<thead>
<tr>
<th>Component</th>
<th>RHN ID</th>
<th>OS Support</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Hat Developer Toolset 7.1</td>
<td>devtoolset-7*</td>
<td>RHEL6, RHEL7</td>
<td>x86_64, s390x, aarch64, ppc64, ppc64le</td>
</tr>
<tr>
<td>Perl 5.26.3</td>
<td>rh-perl526*</td>
<td>RHEL7</td>
<td>x86_64, s390x, aarch64, ppc64le</td>
</tr>
<tr>
<td>MongoDB 3.6.3</td>
<td>rh-mongodb36*</td>
<td>RHEL7</td>
<td>x86_64, s390x, aarch64, ppc64le</td>
</tr>
<tr>
<td>Varnish Cache 5.2.1</td>
<td>rh-varnish5*</td>
<td>RHEL7</td>
<td>x86_64, s390x, aarch64, ppc64le</td>
</tr>
<tr>
<td>PostgreSQL 10.15</td>
<td>rh-postgresql10</td>
<td>RHEL7</td>
<td>x86_64, s390x, aarch64, ppc64le</td>
</tr>
<tr>
<td>PHP 7.0.27</td>
<td>rh-php70*</td>
<td>RHEL6, RHEL7</td>
<td>x86_64</td>
</tr>
<tr>
<td>MySQL 5.7.24</td>
<td>rh-mysql57*</td>
<td>RHEL6, RHEL7</td>
<td>x86_64, s390x, aarch64, ppc64le</td>
</tr>
</tbody>
</table>

### Components Last Updated in Red Hat Software Collections 3.0

<table>
<thead>
<tr>
<th>Component</th>
<th>RHN ID</th>
<th>OS Support</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHP 7.1.8</td>
<td>rh-php71*</td>
<td>RHEL7</td>
<td>x86_64, s390x, aarch64, ppc64le</td>
</tr>
<tr>
<td>nginx 1.12.1</td>
<td>rh-nginx12*</td>
<td>RHEL7</td>
<td>x86_64, s390x, aarch64, ppc64le</td>
</tr>
<tr>
<td>Python 3.6.12</td>
<td>rh-python36*</td>
<td>RHEL6, RHEL7</td>
<td>x86_64, s390x, aarch64, ppc64le</td>
</tr>
<tr>
<td>Maven 3.5.0</td>
<td>rh-maven35*</td>
<td>RHEL7</td>
<td>x86_64, s390x, aarch64, ppc64le</td>
</tr>
<tr>
<td>MariaDB 10.2.22</td>
<td>rh-mariadb102*</td>
<td>RHEL6, RHEL7</td>
<td>x86_64, s390x, aarch64, ppc64le</td>
</tr>
<tr>
<td>PostgreSQL 9.6.19</td>
<td>rh-postgresql96*</td>
<td>RHEL6, RHEL7</td>
<td>x86_64, s390x, aarch64, ppc64le</td>
</tr>
</tbody>
</table>
## Components Last Updated in Red Hat Software Collections 3.0

<table>
<thead>
<tr>
<th>Component</th>
<th>Version</th>
<th>Red Hat Distributions</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>MongoDB 3.4.9</td>
<td>rh-mongodb34*</td>
<td>RHEL6, RHEL7</td>
<td>x86_64, s390x, aarch64, ppc64le</td>
</tr>
<tr>
<td>Node.js 8.11.4</td>
<td>rh-nodejs8*</td>
<td>RHEL7</td>
<td>x86_64, s390x, aarch64, ppc64le</td>
</tr>
</tbody>
</table>

## Components Last Updated in Red Hat Software Collections 2.4

<table>
<thead>
<tr>
<th>Component</th>
<th>Version</th>
<th>Red Hat Distributions</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Hat Developer Toolset 6.1</td>
<td>devtoolset-6*</td>
<td>RHEL6, RHEL7</td>
<td>x86_64, s390x, aarch64, ppc64, ppc64le</td>
</tr>
<tr>
<td>Scala 2.10.6</td>
<td>rh-scala210*</td>
<td>RHEL7</td>
<td>x86_64</td>
</tr>
<tr>
<td>nginx 1.10.2</td>
<td>rh-nginx110*</td>
<td>RHEL6, RHEL7</td>
<td>x86_64</td>
</tr>
<tr>
<td>Node.js 6.11.3</td>
<td>rh-nodejs6*</td>
<td>RHEL6, RHEL7</td>
<td>x86_64, s390x, aarch64, ppc64le</td>
</tr>
<tr>
<td>Ruby 2.4.6</td>
<td>rh-ruby24*</td>
<td>RHEL6, RHEL7</td>
<td>x86_64</td>
</tr>
<tr>
<td>Ruby on Rails 5.0.1</td>
<td>rh-ror50*</td>
<td>RHEL6, RHEL7</td>
<td>x86_64</td>
</tr>
<tr>
<td>Eclipse 4.6.3</td>
<td>rh-eclipse46*</td>
<td>RHEL7</td>
<td>x86_64</td>
</tr>
<tr>
<td>Python 2.7.18</td>
<td>python27</td>
<td>RHEL6*, RHEL7</td>
<td>x86_64, s390x, aarch64, ppc64le</td>
</tr>
<tr>
<td>Thermostat 1.6.6</td>
<td>rh-thermostat16*</td>
<td>RHEL6, RHEL7</td>
<td>x86_64</td>
</tr>
<tr>
<td>Maven 3.3.9</td>
<td>rh-maven33*</td>
<td>RHEL6, RHEL7</td>
<td>x86_64</td>
</tr>
<tr>
<td>Common Java Packages</td>
<td>rh-java-common*</td>
<td>RHEL6, RHEL7</td>
<td>x86_64</td>
</tr>
</tbody>
</table>

## Components Last Updated in Red Hat Software Collections 2.3

<table>
<thead>
<tr>
<th>Component</th>
<th>Version</th>
<th>Red Hat Distributions</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Git 2.9.3</td>
<td>rh-git29*</td>
<td>RHEL6, RHEL7</td>
<td>x86_64, s390x, aarch64, ppc64le</td>
</tr>
<tr>
<td>Redis 3.2.4</td>
<td>rh-redis32*</td>
<td>RHEL6, RHEL7</td>
<td>x86_64</td>
</tr>
<tr>
<td>Perl 5.24.0</td>
<td>rh-perl524*</td>
<td>RHEL6, RHEL7</td>
<td>x86_64</td>
</tr>
<tr>
<td>Python 3.5.1</td>
<td>rh-python35*</td>
<td>RHEL6, RHEL7</td>
<td>x86_64</td>
</tr>
<tr>
<td>Components Last Updated in Red Hat Software Collections 2.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>MongoDB 3.2.10</strong></td>
<td>rh-mongodb32*</td>
<td>RHEL6, RHEL7</td>
<td>x86_64</td>
</tr>
<tr>
<td><strong>Ruby 2.3.8</strong></td>
<td>rh-ruby23*</td>
<td>RHEL6, RHEL7</td>
<td>x86_64</td>
</tr>
<tr>
<td><strong>PHP 5.6.25</strong></td>
<td>rh-php56*</td>
<td>RHEL6, RHEL7</td>
<td>x86_64</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Components Last Updated in Red Hat Software Collections 2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Red Hat Developer Toolset 4.1</strong></td>
</tr>
<tr>
<td><strong>MariaDB 10.1.29</strong></td>
</tr>
<tr>
<td><strong>MongoDB 3.0.11 upgrade collection</strong></td>
</tr>
<tr>
<td><strong>Node.js 4.6.2</strong></td>
</tr>
<tr>
<td><strong>PostgreSQL 9.5.14</strong></td>
</tr>
<tr>
<td><strong>Ruby on Rails 4.2.6</strong></td>
</tr>
<tr>
<td><strong>MongoDB 2.6.9</strong></td>
</tr>
<tr>
<td><strong>Thermostat 1.4.4</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Components Last Updated in Red Hat Software Collections 2.1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Varnish Cache 4.0.3</strong></td>
</tr>
<tr>
<td><strong>nginx 1.8.1</strong></td>
</tr>
<tr>
<td><strong>Node.js 0.10</strong></td>
</tr>
<tr>
<td><strong>Maven 3.0.5</strong></td>
</tr>
<tr>
<td><strong>V8 3.14.5.10</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Components Last Updated in Red Hat Software Collections 2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Red Hat Developer Toolset 3.1</strong></td>
</tr>
<tr>
<td>Component</td>
</tr>
<tr>
<td>----------------------------</td>
</tr>
<tr>
<td>Perl 5.20.1</td>
</tr>
<tr>
<td>Python 3.4.2</td>
</tr>
<tr>
<td>Ruby 2.2.9</td>
</tr>
<tr>
<td>Ruby on Rails 4.1.5</td>
</tr>
<tr>
<td>MariaDB 10.0.33</td>
</tr>
<tr>
<td>MySQL 5.6.40</td>
</tr>
<tr>
<td>PostgreSQL 9.4.14</td>
</tr>
<tr>
<td>Passenger 4.0.50</td>
</tr>
<tr>
<td>PHP 5.4.40</td>
</tr>
<tr>
<td>PHP 5.5.21</td>
</tr>
<tr>
<td>nginx 1.6.2</td>
</tr>
<tr>
<td>DevAssistant 0.9.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Component</th>
<th>Collection</th>
<th>架构</th>
<th>操作系统</th>
<th>位宽</th>
</tr>
</thead>
<tbody>
<tr>
<td>Git 1.9.4</td>
<td>git19</td>
<td></td>
<td>RHEL6, RHEL7</td>
<td>x86_64</td>
</tr>
<tr>
<td>Perl 5.16.3</td>
<td>perl516</td>
<td></td>
<td>RHEL6, RHEL7</td>
<td>x86_64</td>
</tr>
<tr>
<td>Python 3.3.2</td>
<td>python33</td>
<td></td>
<td>RHEL6, RHEL7</td>
<td>x86_64</td>
</tr>
<tr>
<td>Ruby 1.9.3</td>
<td>ruby193</td>
<td></td>
<td>RHEL6, RHEL7</td>
<td>x86_64</td>
</tr>
<tr>
<td>Ruby 2.0.0</td>
<td>ruby200</td>
<td></td>
<td>RHEL6, RHEL7</td>
<td>x86_64</td>
</tr>
<tr>
<td>Ruby on Rails 4.0.2</td>
<td>ror40</td>
<td></td>
<td>RHEL6, RHEL7</td>
<td>x86_64</td>
</tr>
<tr>
<td>MariaDB 5.5.53</td>
<td>mariadb55</td>
<td></td>
<td>RHEL6, RHEL7</td>
<td>x86_64</td>
</tr>
<tr>
<td>MongoDB 2.4.9</td>
<td>mongodb24</td>
<td></td>
<td>RHEL6, RHEL7</td>
<td>x86_64</td>
</tr>
<tr>
<td>MySQL 5.5.52</td>
<td>mysql55</td>
<td></td>
<td>RHEL6, RHEL7</td>
<td>x86_64</td>
</tr>
</tbody>
</table>
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 have 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-* binaries. 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-mysqli` database connector. Consequently, `php-mysqli` 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:

```python
>>> 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:

```python
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-rhsc1-6-rpms
rhl-variant-rhsc1-6-debug-rpms
rhl-variant-rhsc1-6-source-rpms
rhl-server-rhsc1-6-eus-rpms
rhl-server-rhsc1-6-eus-source-rpms
rhl-server-rhsc1-6-eus-debug-rpms
rhl-variant-rhsc1-7-rpms
rhl-variant-rhsc1-7-debug-rpms
rhl-variant-rhsc1-7-source-rpms
rhl-server-rhsc1-7-eus-rpms
rhl-server-rhsc1-7-eus-source-rpms
rhl-server-rhsc1-7-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. For information how to migrate to RHSM, see [https://access.redhat.com/products/red-hat-subscription-management/#migration](https://access.redhat.com/products/red-hat-subscription-management/#migration).

### 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

<table>
<thead>
<tr>
<th>Package from a Software Collection</th>
<th>Required Package from the Optional Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td>devtoolset-10-build</td>
<td>scl-utils-build</td>
</tr>
<tr>
<td>devtoolset-10-dyninst-testsuite</td>
<td>glibc-static</td>
</tr>
<tr>
<td>devtoolset-10-elfutils-debuginfod</td>
<td>bsdtar</td>
</tr>
<tr>
<td>devtoolset-10-gcc-plugin-devel</td>
<td>libmpc-devel</td>
</tr>
<tr>
<td>devtoolset-10-gdb</td>
<td>source-highlight</td>
</tr>
<tr>
<td>devtoolset-9-build</td>
<td>scl-utils-build</td>
</tr>
<tr>
<td>devtoolset-9-dyninst-testsuite</td>
<td>glibc-static</td>
</tr>
<tr>
<td>devtoolset-9-gcc-plugin-devel</td>
<td>libmpc-devel</td>
</tr>
<tr>
<td>devtoolset-9-gdb</td>
<td>source-highlight</td>
</tr>
<tr>
<td>httpd24-mod_ldap</td>
<td>apr-util-ldap</td>
</tr>
<tr>
<td>httpd24-mod_session</td>
<td>apr-util-openssl</td>
</tr>
<tr>
<td>python27-python-debug</td>
<td>tix</td>
</tr>
<tr>
<td>python27-python-devel</td>
<td>scl-utils-build</td>
</tr>
<tr>
<td>python27-tkinter</td>
<td>tix</td>
</tr>
<tr>
<td>rh-git227-git-cvs</td>
<td>cvspsp</td>
</tr>
<tr>
<td>rh-git227-git-svn</td>
<td>perl-Git-SVN, subversion</td>
</tr>
<tr>
<td>rh-git227-perl-Git-SVN</td>
<td>subversion-perl</td>
</tr>
<tr>
<td>rh-java-common-ant-apache-bsf</td>
<td>rhino</td>
</tr>
<tr>
<td>rh-java-common-batik</td>
<td>rhino</td>
</tr>
<tr>
<td>rh-maven35-build</td>
<td>scl-utils-build</td>
</tr>
</tbody>
</table>
### 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:

```bash
yum list available software_collection\*
```

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

```bash
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` packages, type:

```bash
~\# 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`:

```bash
debuginfo-install package_name
```

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

```bash
~\# 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`:

```bash
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.

Table 4.1. Interoperability Between Languages and Databases
<table>
<thead>
<tr>
<th>Language (Software Collection)</th>
<th>MariaDB</th>
<th>MongoDB</th>
<th>MySQL</th>
<th>PostgreSQL</th>
<th>Redis</th>
<th>SQLite3</th>
</tr>
</thead>
<tbody>
<tr>
<td>rh-nodejs4</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>rh-nodejs6</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>rh-nodejs8</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>rh-nodejs10</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>rh-nodejs12</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>rh-nodejs14</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>rh-perl520</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>rh-perl524</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>rh-perl526</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>rh-perl530</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>rh-php56</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>rh-php70</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>rh-php71</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>rh-php72</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>
### CHAPTER 4. SPECIFICS OF INDIVIDUAL SOFTWARE COLLECTIONS

<table>
<thead>
<tr>
<th>Language (Software Collection)</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MariaDB</td>
</tr>
<tr>
<td>rh-php73</td>
<td>✓</td>
</tr>
<tr>
<td>python27</td>
<td>✓</td>
</tr>
<tr>
<td>rh-python34</td>
<td>×</td>
</tr>
<tr>
<td>rh-python35</td>
<td>✓</td>
</tr>
<tr>
<td>rh-python36</td>
<td>✓</td>
</tr>
<tr>
<td>rh-python38</td>
<td>✓</td>
</tr>
<tr>
<td>rh-ror41</td>
<td>✓</td>
</tr>
<tr>
<td>rh-ror42</td>
<td>✓</td>
</tr>
<tr>
<td>rh-ror50</td>
<td>✓</td>
</tr>
<tr>
<td>rh-ruby25</td>
<td>✓</td>
</tr>
<tr>
<td>rh-ruby26</td>
<td>✓</td>
</tr>
<tr>
<td>rh-ruby27</td>
<td>✓</td>
</tr>
<tr>
<td>rh-ruby30</td>
<td>✓</td>
</tr>
</tbody>
</table>

- Supported
- Unsupported
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-* binaries. 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.


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-mysqld-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: Differences in the PostgreSQL paths

<table>
<thead>
<tr>
<th>Content</th>
<th>postgresql</th>
<th>rh-postgresql12</th>
<th>rh-postgresql13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executables</td>
<td>/usr/bin/</td>
<td>/opt/rh/rh-postgresql12/root/usr/bin/</td>
<td>/opt/rh/rh-postgresql13/root/usr/bin/</td>
</tr>
<tr>
<td>Source</td>
<td>not installed</td>
<td>not installed</td>
<td>not installed</td>
</tr>
<tr>
<td>Development Headers</td>
<td>/usr/include/pgsql/</td>
<td>/opt/rh/rh-postgresql12/root/usr/include/pgsql/</td>
<td>/opt/rh/rh-postgresql13/root/usr/include/pgsql/</td>
</tr>
</tbody>
</table>
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`:

   ```bash
   systemctl stop postgresql.service
   ```

   To verify that the server is not running, type:

   ```bash
   systemctl status postgresql.service
   ```

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

   ```bash
   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:

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

---

<table>
<thead>
<tr>
<th>Content</th>
<th>postgresql</th>
<th>rh-postgresql12</th>
<th>rh-postgresql13</th>
</tr>
</thead>
</table>
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:

```bash
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:

```bash
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:

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

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

```bash
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:

```bash
chkconfig postgresql off
```

To enable the PostgreSQL 13 server, type as root:

```bash
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:

```bash
systemctl start postgresql.service
```

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

```bash
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-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 5.4. 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:

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

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

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

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

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

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

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

5. Start the new server as root:

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

6. Import data from the previously created SQL file:

   ```bash
   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:

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

   To enable the PostgreSQL 13 server, type as root:

   ```bash
   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.


### 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.


- **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

<table>
<thead>
<tr>
<th>Revision 3.7-4</th>
<th>Fri Nov 12 2021</th>
<th>Lenka Špačková</th>
</tr>
</thead>
<tbody>
<tr>
<td>Updated Section 4.3, “Database Connectors”.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Revision 3.7-3</th>
<th>Tue Aug 24 2021</th>
<th>Lenka Špačková</th>
</tr>
</thead>
<tbody>
<tr>
<td>Added two notes related to asynchronous changes in Python due to CVE fixes (Section 1.6, “Other Notes”).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extended Section 1.3.5, “Changes in Ruby”.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Revision 3.7-2</th>
<th>Mon Jul 12 2021</th>
<th>Lenka Špačková</th>
</tr>
</thead>
<tbody>
<tr>
<td>Added a note about the <code>.mysql_history</code> file.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Revision 3.7-1</th>
<th>Thu Jun 03 2021</th>
<th>Lenka Špačková</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Revision 3.7-0</th>
<th>Mon May 03 2021</th>
<th>Lenka Špačková</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release of Red Hat Software Collections 3.7 Beta Release Notes.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>