



# Red Hat Software Collections 3

## 3.1 发行注记

Red Hat Software Collections 3.1 发行注记



# Red Hat Software Collections 3 3.1 发行注记

---

## Red Hat Software Collections 3.1 发行注记

Lenka Špačková

Red Hat Customer Content Services

lspackova@redhat.com

Jaromír Hradílek

Red Hat Customer Content Services

jhradilek@redhat.com

Eliška Slobodová

Red Hat Customer Content Services

## 法律通告

Copyright © 2018 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

## 摘要

Red Hat Software Collections 3.1 发行注记记录了主要功能，并包含有关 Red Hat Software Collections 3.1 中已知问题的重要信息。Red Hat Developer Toolset 集合记录在 Red Hat Developer Toolset 发行注记和 Red Hat Developer Toolset 用户指南中。

## 目录

<b>第 1 章 RED HAT SOFTWARE COLLECTIONS 3.1</b> .....	<b>4</b>
1.1. 关于 RED HAT SOFTWARE COLLECTIONS	4
1.1.1. Red Hat Developer Toolset	4
1.2. 主要功能	4
1.3. RED HAT SOFTWARE COLLECTIONS 3.1 的变化	13
1.3.1. 概述	13
构架	13
新 Software Collections	13
更新了 Software Collections	14
Red Hat Software Collections 容器镜像	14
1.3.2. Red Hat Developer Toolset 的变化	14
1.3.3. Perl 中的更改	15
1.3.4. Ruby 的变化	15
1.3.5. MongoDB 的变化	16
1.3.6. PostgreSQL 中的更改	16
1.3.7. Varnish Cache 中的更改	17
1.3.8. HAProxy 的更改	17
1.3.9. PHP 中的更改	17
1.3.10. MySQL 的更改	18
1.3.11. Apache httpd 中的更改	18
1.4. 兼容性信息	18
1.5. 已知问题	18
其他备注	23
1.6. 已弃用的功能	26
<b>第 2 章 安装</b> .....	<b>27</b>
2.1. 访问 RED HAT SOFTWARE COLLECTIONS	27
2.1.1. 使用 Red Hat Subscription Management	27
2.1.2. Optional 频道中的软件包	29
2.2. 安装 RED HAT SOFTWARE COLLECTIONS	31
2.2.1. 安装单个 Software Collections	32
2.2.2. 安装可选软件包	32
2.2.3. 安装调试信息	32
2.3. 卸载 RED HAT SOFTWARE COLLECTIONS	33
2.4. 重建 RED HAT SOFTWARE COLLECTIONS	33
<b>第 3 章 使用方法</b> .....	<b>34</b>
3.1. 使用 RED HAT SOFTWARE COLLECTIONS	34
3.1.1. 从 Software Collection 运行可执行文件	34
3.1.2. 使用 Software Collection 作为默认值运行 Shell 会话	34
3.1.3. 从 Software Collection 中运行系统服务	35
3.2. 从 SOFTWARE COLLECTION 访问 MAN PAGE	35
3.3. 部署使用 RED HAT SOFTWARE COLLECTIONS 的应用程序	36
3.4. RED HAT SOFTWARE COLLECTIONS 容器镜像	36
<b>第 4 章 单个软件集合的具体信息</b> .....	<b>42</b>
4.1. RED HAT DEVELOPER TOOLSET	42
4.2. ECLIPSE 4.6.3	42
4.2.1. 安装 Eclipse	44
4.2.2. 使用 Eclipse	45
4.2.2.1. 使用 Red Hat Developer Toolset Toolchain	45
4.2.2.2. 使用 Red Hat Enterprise Linux Toolchain	46

4.2.3. 其它资源	47
安装的文档	47
另请参阅	47
4.3. RUBY ON RAILS 5.0	48
4.4. MONGODB 3.6	48
4.5. MONGODB 3.4	50
Red Hat Enterprise Linux 6 上的 MongoDB 3.4	50
MongoDB 3.4 on Red Hat Enterprise Linux 7	51
4.6. GIT	51
4.7. MAVEN	51
4.8. PASSENGER	52
4.9. 数据库连接器	53
<b>第 5 章 MIGRATION (迁移)</b>	<b>55</b>
5.1. 迁移到 MARIADB 10.2	55
5.1.1. rh-mariadb101 和 rh-mariadb102 Software Collections 之间的显著差异	55
5.1.2. 从 rh-mariadb101 升级到 rh-mariadb102 软件集合	56
5.2. 迁移到 MONGODB 3.6	57
5.2.1. MongoDB 3.4 和 MongoDB 3.6 之间的显著差异	57
常规更改	57
兼容性更改	58
后向兼容的功能	58
5.2.2. 从 rh-mongodb34 升级到 rh-mongodb36 软件集合	59
5.3. 迁移到 MONGODB 3.4	60
5.3.1. MongoDB 3.2 和 MongoDB 3.4 之间的显著差异	61
常规更改	61
兼容性更改	61
5.3.2. 从 rh-mongodb32 升级到 rh-mongodb34 Software Collection	61
5.4. 迁移到 MYSQL 5.7	63
5.4.1. MySQL 5.6 和 MySQL 5.7 之间的显著差异	63
5.4.2. 升级到 rh-mysql57 Software Collection	64
5.5. 迁移到 POSTGRESQL 10	65
5.5.1. 从 Red Hat Enterprise Linux 系统版本 PostgreSQL 迁移到 PostgreSQL 10 Software Collection	66
5.5.2. 从 PostgreSQL 9.6 Software Collection 迁移到 PostgreSQL 10 Software Collection	70
5.6. 迁移到 POSTGRESQL 9.6	73
5.6.1. PostgreSQL 9.5 和 PostgreSQL 9.6 之间的显著差异	73
5.6.2. 从 Red Hat Enterprise Linux 系统版本 PostgreSQL 迁移到 PostgreSQL 9.6 软件集合	75
5.6.3. 从 PostgreSQL 9.5 Software Collection 迁移到 PostgreSQL 9.6 Software Collection	78
5.7. 迁移到 NGINX 1.12	81
<b>第 6 章 其它资源</b>	<b>83</b>
6.1. 红帽产品文档	83
6.2. RED HAT DEVELOPERS	84
<b>附录 A. 修订历史记录</b>	<b>85</b>



# 第 1 章 RED HAT SOFTWARE COLLECTIONS 3.1

本章作为 Red Hat Software Collections 3.1 内容集的概述。它提供了组件及其描述列表，总结此版本的更改、记录相关兼容性信息以及列出已知问题。

## 1.1. 关于 RED HAT SOFTWARE COLLECTIONS

对于某些应用程序，通常需要最新版本的一些软件组件才能使用它们的最新新功能。**Red Hat Software Collections** 是一个红帽产品，它提供一组动态编程语言、数据库服务器和各种相关软件包，它们比基本 Red Hat Enterprise Linux 系统中包含的对等版本更最新的，或者首次对此系统可用。

Red Hat Software Collections 3.1 可用于 Red Hat Enterprise Linux 7；为 Red Hat Enterprise Linux 6 选择了新组件和以前发布的组件。有关作为 Red Hat Software Collections 一部分分发的组件的完整列表，以及其功能的简短概述，请参阅 [第 1.2 节“主要功能”](#)。

Red Hat Software Collections 不会替换 Red Hat Enterprise Linux 6 或 Red Hat Enterprise Linux 7 提供的默认系统工具。相反，在 `/opt/` 目录中安装了一组并行工具，并可以选择使用提供的 `scl` 工具为每个应用程序启用。例如，Perl 或 PostgreSQL 的默认版本会保留由基本 Red Hat Enterprise Linux 系统提供的。

Red Hat Enterprise Linux 订阅级别协议完全支持所有 Red Hat Software Collections 组件，它们的功能完整，并可用于生产环境。重要的程序错误修复和安全勘误会以类似 Red Hat Enterprise Linux 的方式向 Red Hat Software Collections 发布程序发布，从每个主版本发行之日起至少两年。在每个主发行版本流中，所选组件的每个版本都保持向后兼容性。有关各个组件支持长度的详细信息，请参阅 [Red Hat Software Collections 产品生命周期](#) 文档。

### 1.1.1. Red Hat Developer Toolset

**Red Hat Developer Toolset** 是 Red Hat Software Collections 的一部分，作为一个单独的 Software Collection 提供。有关 Red Hat Developer Toolset 的更多信息，请参阅 [Red Hat Developer Toolset 发行注记](#) 和 [Red Hat Developer Toolset 用户指南](#)。

## 1.2. 主要功能

[表 1.1“Red Hat Software Collections 3.1 组件”](#) 列出 Red Hat Software Collections 3.1 版本时支持的所有组件。

表 1.1. Red Hat Software Collections 3.1 组件

组件	Software Collection	描述
Red Hat Developer Toolset 7.1	devtoolset-7	Red Hat Developer Toolset 专为使用 Red Hat Enterprise Linux 平台的开发人员设计。它提供了 <b>GNU Compiler Collection</b> 、 <b>GNU Debugger</b> 和其他开发、调试和性能监控工具的当前版本。有关组件的完整列表，请参阅 <a href="#">Red Hat Developer Toolset 用户指南中的 Red Hat Developer Toolset 组件表</a> 。
Eclipse 4.6.3 <sup>[a]</sup>	rh-eclipse46	Eclipse <b>集成开发环境</b> 的发布，它基于 Eclipse Foundation 的 Neon 版本培训。 <b>Eclipse</b> 之前作为 Red Hat Developer Toolset 组件提供。此 Software Collection 依赖于 rh-java-common 组件。



组件	Software Collection	描述
Perl 5.24.0	rh-perl524	Perl 版本，一种高级编程语言，通常用于系统管理实用程序和 Web 编程。rh-perl524 Software Collection 为 MySQL 和 PostgreSQL 提供了额外的实用程序、脚本和数据库连接器。它包括 <b>DateTime</b> Perl 模块和 <b>mod_perl</b> Apache httpd 模块，该模块仅支持 httpd24 软件集合。另外，它还为轻松安装 CPAN 模块提供 <b>cpanm</b> 工具。
Perl 5.26.1 <sup>[a]</sup>	rh-perl526	Perl 版本，一种高级编程语言，通常用于系统管理实用程序和 Web 编程。rh-perl526 Software Collection 为 MySQL 和 PostgreSQL 提供了额外的实用程序、脚本和数据库连接器。它包括 <b>DateTime</b> Perl 模块和 <b>mod_perl</b> Apache httpd 模块，该模块仅支持 httpd24 软件集合。另外，它还为轻松安装 CPAN 模块提供 <b>cpanm</b> 工具。rh-perl526 打包与上游社区一致；perl526-perl 软件包也会安装核心模块，而解释器则由 perl-interpreter 软件包提供。
PHP 7.0.27	rh-php70	带有 PEAR 1.10 的 PHP 7.0 发行版本，改进了语言 <b>功能和性能</b> 。
PHP 7.1.8 <sup>[a]</sup>	rh-php71	使用 PEAR 1.10、APCu 5.1.8 和增强功能的 PHP 7.1 发布。
Python 2.7.13	python27	带有多个附加工具的 Python 2.7 发行版本。这个 Python 版本提供各种功能和增强，包括排序的字典类型、更快的 I/O 操作，并改进了与 Python 3 的正向兼容性。python27 Software Collections 包含 <i>Python 2.7.13</i> 解释器，一组用于编程 Web 应用程序和 <b>mod_wsgi</b> 的扩展库（只在 httpd24 Software Collection 中被支持）、MySQL 和 PostgreSQL 数据库连接器，以及 <b>numpy</b> 和 <b>scipy</b> 。
Python 3.5.1	rh-python35	rh-python35 Software Collection 包含 <i>Python 3.5.1</i> 解释器，一组用于编程 Web 应用和 <b>mod_wsgi</b> 的扩展库（只在 httpd24 Software Collection 中受支持）、PostgreSQL 数据库连接器和 <b>numpy</b> 和 <b>scipy</b> 。
Python 3.6.3	rh-python36	rh-python36 Software Collection 包含 Python 3.6.3，它引入了一些新功能，如 <i>f-strings</i> 、 <b>变量注解的语法</b> 以及 <b>异步生成器和理解</b> 。此外，还包括一组适用于编程 Web 应用的扩展库， <b>mod_wsgi</b> （仅支持 httpd24 软件集合）、PostgreSQL 数据库连接器、 <b>numpy</b> 和 <b>scipy</b> 。
Ruby 2.3.6	rh-ruby23	Ruby 2.3 发布。此版本 <b>引入了一个命令行选项，用于冻结源文件中的所有字符串文字、安全导航运算符和多个性能增强</b> ，同时维护与 Ruby 2.2、Ruby 2.0.0 和 Ruby 1.9.3 的源级向后兼容性。
Ruby 2.4.3	rh-ruby24	Ruby 2.4 的版本。此版本提供多个性能改进和增强，例如 <b>改进哈希表、新的调试功能、对 Unicode 案例映射的支持以及 OpenSSL 1.1.0 的支持</b> 。Ruby 2.4.0 维护与 Ruby 2.3、Ruby 2.2、Ruby 2.0.0 和 Ruby 1.9.3 的源级向后兼容性。

组件	Software Collection	描述
Ruby 2.5.0 <a href="#">[a]</a>	rh-ruby25	Ruby 2.5 发行之日。此版本提供多个性能改进和新功能，例如，简化了带有 <b>rescue</b> 、 <b>else</b> 和 <b>ensure</b> 关键字的块的使用，一个新的 <b>yield_self</b> 方法，支持分支覆盖和方法覆盖测量、新的 <b>Hash#slice</b> 和 <b>Hash#transform_keys</b> 方法。Ruby 2.5.0 保持与 Ruby 2.4 的源级向后兼容性。
Ruby on Rails 4.2.6	rh-ror42	Ruby on Rails 4.2 的一个发行版本，它是一个使用 Ruby 语言编写的 web 应用程序框架。本发行版本中的亮点包括 <i>Active Job</i> 、 <i>异步邮件</i> 、 <i>Adequate Record</i> 、 <i>Web 控制台</i> 和 <i>外键支持</i> 。此 Software Collection 与 rh-ruby23 和 rh-nodejs4 Collections 一起支持。
Ruby on Rails 5.0.1	rh-ror50	Ruby on Rails 5.0 版本，这是使用 Ruby 语言编写的 Web 应用程序框架的最新版本。主要新功能包括 <i>Action Cable</i> 、 <i>API mode</i> 、 <i>exclusive use of rails CLI over Rake</i> 、和 <i>ActionRecord 属性</i> 。此 Software Collection 与 rh-ruby24 和 rh-nodejs6 Collections 一起支持。
Scala 2.10.6 <a href="#">[a]</a>	rh-scala210	Scala 的发布是 Java 平台的通用编程语言，集成了面向对象和功能语言的功能。
MariaDB 10.1.29	rh-mariadb101	MariaDB 版本，这是 Red Hat Enterprise Linux 用户的 MySQL 的替代方案。对于所有实际目的，MySQL 与 MariaDB 兼容的二进制文件，可以在没有任何数据转换的情况下被替换为它。此版本添加了 <i>Galera 集群支持</i> 。
MariaDB 10.2.8	rh-mariadb102	MariaDB 版本，这是 Red Hat Enterprise Linux 用户的 MySQL 的替代方案。对于所有实际目的，MySQL 与 MariaDB 兼容的二进制文件，可以在没有任何数据转换的情况下被替换为它。此版本添加了 <i>MariaDB 备份</i> 、 <i>Flashback</i> 、 <i>对递归通用表表达式</i> 、 <i>窗口功能</i> 和 <i>JSON 功能的支持</i> 。
MongoDB 3.2.10	rh-mongodb32	MongoDB 的发布，是一个跨平台 <i>文档型数据库系统</i> ，被归类为 <i>NoSQL 数据库</i> 。此 Software Collection 包括 <i>mongo-java-driver</i> 软件包版本 3.2.1。
MongoDB 3.4.9	rh-mongodb34	MongoDB 的发布，一个面向平台的跨平台文档数据库系统被归类为 <i>NoSQL 数据库</i> 。此发行版本引进了对新架构的支持，增加了对 <i>十进制28 类型的消息压缩</i> 和支持，增强了 <i>协调功能</i> 等等。
MongoDB 3.6.3 <a href="#">[a]</a>	rh-mongodb36	MongoDB 的发布，一个面向平台的跨平台文档数据库系统被归类为 <i>NoSQL 数据库</i> 。此发行版本引入了 <i>更改流</i> 、 <i>可重试写入</i> 和 <i>JSON 架构</i> ，以及其他功能。
MySQL 5.7.21	rh-mysql57	MySQL 的发布，它提供很多新功能和增强，包括提高性能。

组件	Software Collection	描述
PostgreSQL 9.5.9	rh-postgresql95	PostgreSQL 的发行版本，它提供了很多改进，包括 <i>行级安全控制</i> 、引入复制进度跟踪、改进处理大量列的大型表，并提高了排序和多 CPU 机器的性能。
PostgreSQL 9.6.5	rh-postgresql96	PostgreSQL 的一个发行版本，它引入了后续扫描、加入和聚合的并行执行，并提供同步复制、全文本搜索、解码驱动程序、 <code>postgres_fdw</code> 以及性能改进。
PostgreSQL 10.3 <a href="#">[a]</a>	rh-postgresql10	PostgreSQL 发行版本，它包括显著的性能改进和一些新功能，如使用 <i>publish</i> 和 <i>subscribe</i> 关键字的 <i>逻辑复制</i> ，或者基于 <i>SCRAM-SHA-256</i> 机制的更强大的密码身份验证。
Node.js 4.6.2	rh-nodejs4	Node.js 的一个发行版本，它提供了一个基于 Chrome 的 V8 JavaScript 引擎和 <code>npm 2.15.1</code> 构建的 JavaScript 运行时，这是 JavaScript 的软件包管理器。这个版本包括增强的 API、多个安全和程序错误修复，以及对 <i>SPDY 协议版本 3.1</i> 的支持
Node.js 6.11.3	rh-nodejs6	一个 Node.js 版本，提供多个 API 增强、性能和安全改进， <i>ECMA Script 2015</i> 支持， <code>npm 3.10.9</code> 。
Node.js 8.9.4 <a href="#">[a]</a>	rh-nodejs8	Node.js 的一个发行版本，提供多个 API 增强和新功能，包括 <i>V8 引擎版本 6.0</i> 、 <code>npm 5.6.0</code> 和 <code>npx</code> 、增强的安全性、 <i>实验性 N-API</i> 支持和性能改进。
nginx 1.8.1	rh-nginx18	nginx 版本、一个 web 和代理服务器，专注于高并发、性能和低内存使用。此版本引入了一些新功能，包括 <i>后端 SSL 证书验证</i> 、 <i>记录到 syslog</i> 、 <i>用于卸载 I/O 请求或哈希负载均衡方法的线程池</i> 。
nginx 1.10.2	rh-nginx110	nginx 版本、一个 web 和代理服务器，专注于高并发、性能和低内存使用。这个版本引入了很多新功能，包括 <i>动态模块支持</i> 、 <i>HTTP/2</i> 支持、 <i>Perl 集成</i> 和大量性能改进。
nginx 1.12.1 <a href="#">[a]</a>	rh-nginx112	nginx 版本、一个 web 和代理服务器，专注于高并发、性能和低内存使用。这个版本引入了很多新功能，包括 <i>IP Transparency</i> 、 <i>改进了 TCP/UDP 负载均衡</i> 、 <i>增强的缓存性能</i> 以及大量性能改进。
Apache httpd 2.4.27	httpd24	Apache HTTP 服务器(httpd)发行版本， <i>包括基于性能事件的处理模型</i> 、 <i>增强的 SSL 模块</i> 和 <i>FastCGI</i> 支持。还包括 <code>mod_auth_kerb</code> 模块。
Varnish Cache 4.0.3	rh-varnish4	Varnish Cache 的一个发行版本，它是一个 <i>高性能 HTTP 反向代理</i> 。 <b>Varnish 缓存</b> 将文件或文件片段存储在内存中，用于减少未来对等请求的响应时间和网络带宽消耗。

组件	Software Collection	描述
Varnish Cache 5.2.1 <a href="#">[a]</a>	rh-varnish5	Varnish Cache 的一个发行版本，它是一个高性能 HTTP 反向代理。此版本包括 <b>分片 director</b> 、 <b>实验性 HTTP/2 支持</b> 以及 <b>通过单独的 VCL 文件和 VCL 标签对 Varnish 配置的改进</b> 。
Maven 3.3.9	rh-maven33	Maven 版本，一种 <b>软件项目管理和理解工具</b> ，主要用于 Java 项目。此版本提供各种改进，例如， <b>改进了内核扩展机制</b> 。
Maven 3.5.0 <a href="#">[a]</a>	rh-maven35	Maven 版本，一种软件项目管理和理解工具。此发行版本引进了对新架构和多个新功能的支持，包括 <b>颜色日志记录</b> 。
Git 2.9.3	rh-git29	Git 版本，一种具有分散架构的分布式修订控制系统。与使用客户端-服务器模型的集中式版本控制系统不同，Git 确保 Git 存储库的每个工作副本都是其完整修订历史记录的确切副本。
Redis 3.2.4	rh-redis32	Redis 3.2 的一个发行版本，它是一个 <b>持久的键值数据库</b> 。
HAProxy 1.8.4 <a href="#">[a]</a>	rh-haproxy18	HAProxy 1.8 版本，它是一个可靠、高性能的 <b>网络负载均衡器</b> ，用于基于 TCP 和 HTTP 的应用程序。
常见 Java 软件包	rh-java-common	此 Software Collection 提供由 <b>其他集合使用的通用 Java 库和工具</b> 。devtoolset-4、devtoolset-3、dev-maven33、maven30、rh-mongodb32、rh-mongodb26、rh-mongodb26、rmostat1、rh-thermostat16 和 rh-eclipse46 组件需要 rh-java-common Software Collection，用户不应直接安装它。
<a href="#">[a]</a> 这个 Software Collection 仅适用于 Red Hat Enterprise Linux 7		

以前发布的 Software Collections 在同一个发行版本频道中仍然可用。所有 Software Collections（包括已弃用组件）都列在 [表 1.2 “所有可用的 Software Collections”](#) 中。不再支持的软件集合标有星号(\*)。

有关各个组件支持长度的信息，请参阅 [Red Hat Software Collections 产品生命周期](#) 文档。有关以前发布的组件的详细信息，请参阅早期版本的 Red Hat Software Collections [发行注记](#)。

表 1.2. 所有可用的 Software Collections

组件	Software Collection	可用性	RHEL7 支持的构架
Red Hat Software Collections 3.1 中的组件新功能			
Perl 5.26.1	rh-perl526	RHEL7	x86_64, s390x, aarch64, ppc64le

组件	Software Collection	可用性	RHEL7 支持的构架
<b>Red Hat Software Collections 3.1 中的组件新功能</b>			
Ruby 2.5.0	rh-ruby25	RHEL7	x86_64, s390x, aarch64, ppc64le
MongoDB 3.6.3	rh-mongodb36	RHEL7	x86_64, s390x, aarch64, ppc64le
Varnish Cache 5.2.1	rh-varnish5	RHEL7	x86_64, s390x, aarch64, ppc64le
PostgreSQL 10.3	rh-postgresql10	RHEL7	x86_64, s390x, aarch64, ppc64le
HAProxy 1.8.4	rh-haproxy18	RHEL7	x86_64

<b>Red Hat Software Collections 3.1 中更新的组件</b>			
Red Hat Developer Toolset 7.1	devtoolset-7	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64, ppc64le
PHP 7.0.27	rh-php70	RHEL6, RHEL7	x86_64
MySQL 5.7.21	rh-mysql57	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64le
Apache httpd 2.4.27	httpd24	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64le

<b>在 Red Hat Software Collections 3.0 中更新组件</b>			
PHP 7.1.8	rh-php71	RHEL7	x86_64, s390x, aarch64, ppc64le
nginx 1.12.1	rh-nginx112	RHEL7	x86_64, s390x, aarch64, ppc64le
Python 3.6.3	rh-python36	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64le
Maven 3.5.0	rh-maven35	RHEL7	x86_64, s390x, aarch64, ppc64le

## 在 Red Hat Software Collections 3.0 中更新组件

MariaDB 10.2.8	rh-mariadb102	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64le
PostgreSQL 9.6.5	rh-postgresql96	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64le
MongoDB 3.4.9	rh-mongodb34	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64le
Node.js 8.9.4	rh-nodejs8	RHEL7	x86_64, s390x, aarch64, ppc64le

## Red Hat Software Collections 2.4 中的组件最后更新

Red Hat Developer Toolset 6.1	devtoolset-6	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64, ppc64le
scala 2.10.6	rh-scala210	RHEL7	x86_64
nginx 1.10.2	rh-nginx110	RHEL6, RHEL7	x86_64
Node.js 6.11.3	rh-nodejs6	RHEL6, RHEL7	x86_64
Ruby 2.4.3	rh-ruby24	RHEL6, RHEL7	x86_64
Ruby on Rails 5.0.1	rh-ror50	RHEL6, RHEL7	x86_64
Eclipse 4.6.3	rh-eclipse46	RHEL7	x86_64
Python 2.7.13	python27	RHEL6, RHEL7	x86_64
Thermostat 1.6.6	rh-thermostat16*	RHEL6, RHEL7	x86_64
Maven 3.3.9	rh-maven33	RHEL6, RHEL7	x86_64
常见 Java 软件包	rh-java-common	RHEL6, RHEL7	x86_64

## 在 Red Hat Software Collections 2.3 中更新组件

Git 2.9.3	rh-git29	RHEL6, RHEL7	x86_64
Redis 3.2.4	rh-redis32	RHEL6, RHEL7	x86_64

## 在 Red Hat Software Collections 2.3 中更新组件

Perl 5.24.0	rh-perl524	RHEL6, RHEL7	x86_64
Python 3.5.1	rh-python35	RHEL6, RHEL7	x86_64
MongoDB 3.2.10	rh-mongodb32	RHEL6, RHEL7	x86_64
Ruby 2.3.6	rh-ruby23	RHEL6, RHEL7	x86_64
PHP 5.6.25	rh-php56*	RHEL6, RHEL7	x86_64

## 在 Red Hat Software Collections 2.2 中更新组件

Red Hat Developer Toolset 4.1	devtoolset-4*	RHEL6, RHEL7	x86_64
MariaDB 10.1.29	rh-mariadb101	RHEL6, RHEL7	x86_64
MongoDB 3.0.11 升级集合	rh-mongodb30upg*	RHEL6, RHEL7	x86_64
Node.js 4.6.2	rh-nodejs4	RHEL6, RHEL7	x86_64
PostgreSQL 9.5.9	rh-postgresql95	RHEL6, RHEL7	x86_64
Ruby on Rails 4.2.6	rh-ror42	RHEL6, RHEL7	x86_64
MongoDB 2.6.9	rh-mongodb26*	RHEL6, RHEL7	x86_64
Thermostat 1.4.4	Thermostat1*	RHEL6, RHEL7	x86_64

## Red Hat Software Collections 2.1 中的组件最后更新

Varnish Cache 4.0.3	rh-varnish4	RHEL6, RHEL7	x86_64
nginx 1.8.1	rh-nginx18	RHEL6, RHEL7	x86_64
Node.js 0.10	nodejs010*	RHEL6, RHEL7	x86_64
Maven 3.0.5	maven30*	RHEL6, RHEL7	x86_64
V8 3.14.5.10	v8314*	RHEL6, RHEL7	x86_64

## 组件最后在 Red Hat Software Collections 2.0 中更新

Red Hat Developer Toolset 3.1	devtoolset-3*	RHEL6, RHEL7	x86_64
Perl 5.20.1	rh-perl520*	RHEL6, RHEL7	x86_64
Python 3.4.2	rh-python34*	RHEL6, RHEL7	x86_64
Ruby 2.2.9	rh-ruby22*	RHEL6, RHEL7	x86_64
Ruby on Rails 4.1.5	rh-ror41*	RHEL6, RHEL7	x86_64
MariaDB 10.0.33	rh-mariadb100*	RHEL6, RHEL7	x86_64
MySQL 5.6.40	rh-mysql56*	RHEL6, RHEL7	x86_64
PostgreSQL 9.4.14	rh-postgresql94*	RHEL6, RHEL7	x86_64
passenger 4.0.50	rh-passenger40*	RHEL6, RHEL7	x86_64
PHP 5.4.40	php54*	RHEL6, RHEL7	x86_64
PHP 5.5.21	php55*	RHEL6, RHEL7	x86_64
nginx 1.6.2	nginx16*	RHEL6, RHEL7	x86_64
DevAssistant 0.9.3	devassist09*	RHEL6, RHEL7	x86_64

## 在 Red Hat Software Collections 1 中更新组件

Git 1.9.4	git19*	RHEL6, RHEL7	x86_64
Perl 5.16.3	perl516*	RHEL6, RHEL7	x86_64
Python 3.3.2	python33*	RHEL6, RHEL7	x86_64
Ruby 1.9.3	ruby193*	RHEL6, RHEL7	x86_64
Ruby 2.0.0	ruby200*	RHEL6, RHEL7	x86_64
Ruby on Rails 4.0.2	ror40*	RHEL6, RHEL7	x86_64
MariaDB 5.5.53	mariadb55*	RHEL6, RHEL7	x86_64
MongoDB 2.4.9	mongodb24*	RHEL6, RHEL7	x86_64



## 在 Red Hat Software Collections 1 中更新组件

MySQL 5.5.52	mysql55*	RHEL6, RHEL7	x86_64
PostgreSQL 9.2.18	postgresql92*	RHEL6, RHEL7	x86_64

RHEL6 – Red Hat Enterprise Linux 6

RHEL7 – Red Hat Enterprise Linux 7

x86\_64 - AMD64 和 Intel 64 架构

s390x – IBM z Systems

aarch64 - 64 位 ARM 架构

ppc64 - IBM POWER, big endian

ppc64le - IBM POWER, little endian

\* 弃用的组件 - 此 Software Collection 不再被支持

上面的表通过异步更新列出最新版本。

请注意，Red Hat Software Collections 2.0 及之后的版本中发布的 Software Collections 在其名称中包含 **rh-** 前缀。

## 1.3. RED HAT SOFTWARE COLLECTIONS 3.1 的变化

### 1.3.1. 概述

构架

Red Hat Software Collections 产品包括在 AMD64 和 Intel 64 构架上运行的 Red Hat Enterprise Linux 7 的软件包；某些 Software Collections 还可用于 Red Hat Enterprise Linux 6。

另外，Red Hat Software Collections 3.1 在 Red Hat Enterprise Linux 7 中支持以下构架：

- 64 位 ARM 架构
- IBM z 系统
- IBM POWER, little endian

有关组件及其可用性的完整列表，请参阅 [表 1.2 “所有可用的 Software Collections”](#)。

### 新 Software Collections

Red Hat Software Collections 3.1 添加以下新 Software Collections：

- rh-perl526 – see [第 1.3.3 节 “Perl 中的更改”](#)
- rh-ruby25 – see [第 1.3.4 节 “Ruby 的变化”](#)
- rh-mongodb36 – see [第 1.3.5 节 “MongoDB 的变化”](#)

- rh-postgresql10 – see [第 1.3.6 节 “PostgreSQL 中的更改”](#)
- rh-varnish5 – see [第 1.3.7 节 “Varnish Cache 中的更改”](#)
- rh-haproxy18 – see [第 1.3.8 节 “HAProxy 的更改”](#)

所有新的 Software Collections 仅适用于 Red Hat Enterprise Linux 7。

### 更新了 Software Collections

Red Hat Software Collections 3.1 中更新了以下组件 hav :

- devtoolset-7 – see [第 1.3.2 节 “Red Hat Developer Toolset 的变化”](#)
- rh-php70 – see [第 1.3.9 节 “PHP 中的更改”](#)
- rh-mysql57 – see [第 1.3.10 节 “MySQL 的更改”](#)
- httpd24 – see [第 1.3.11 节 “Apache httpd 中的更改”](#)

### Red Hat Software Collections 容器镜像

以下容器镜像是 Red Hat Software Collections 3.1 中的新容器镜像 :

- rhsc/perl-526-rhel7
- rhsc/ruby-25-rhel7
- rhsc/mongodb-36-rhel7
- rhsc/varnish-5-rhel7
- rhsc/postgresql-10-rhel7

Red Hat Software Collections 3.1 中更新了以下容器镜像 :

- rhsc/devtoolset-7-toolchain-rhel7
- rhsc/devtoolset-7-perftools-rhel7
- rhsc/php-70-rhel7
- rhsc/httpd-24-rhel7

有关 Red Hat Software Collections 容器镜像的详情, 请参考 [第 3.4 节 “Red Hat Software Collections 容器镜像”](#)。

### 1.3.2. Red Hat Developer Toolset 的变化

与之前的 Red Hat Developer Toolset 7.1 版本相比, Red Hat Developer Toolset 7.1 中已经升级了以下组件 :

- **GCC** 到版本 7.3.1

另外, 以下组件提供了程序错误修正更新 :

- **GDB**
- **Valgrind**

- `elfutils`
- `binutils`
- `dwz`
- `memstomp`
- `make`
- `ltrace`
- `strace`
- `oprofile`
- `SystemTap`
- `Dyninst`

有关 7.1 中更改的详细信息，请参阅 [Red Hat Developer Toolset 用户指南](#)。

### 1.3.3. Perl 中的更改

新的 `rh-perl526` Software Collection 包括 **Perl 5.26.1**，它提供很多程序错误修复和增强。使用 `widespread` 效果的一些显著更改有：

- 由于安全原因，当前目录 `.` 已从 `@INC` 模块搜索路径中删除
- 现在，当因为上述行为更改而加载文件时，`do` 语句现在会返回弃用警告
- 在正则表达式模式中未转义的字面 `{` 字符不再被允许
- Unicode 9.0 现在被支持

有关 **Perl 5.26** 的详细更改，请参阅 [5.26.0](#) 和 [5.26.1](#) 版本的上游更改日志。

另外，`rh-perl526` 打包与上游一致。`rh-perl526-perl` 软件包也会安装核心模块，而 `/usr/bin/perl` 解释器则由 `rh-perl526-perl-interpreter` 软件包提供。在以前的版本中，`perl` 软件包仅包含一个最小解释器，而 `perl-core` 软件包同时包含解释器和核心模块。

### 1.3.4. Ruby 的变化

新的 `rh-ruby25` Software Collection 提供 **Ruby 2.5.0**。此版本引入了性能改进以及一些新功能和更改，例如：

- **rescue**、**其他** 和 **确保** 关键字现在可以直接在由 `do` 和 `end` 分隔的块内使用，而无需添加 **begin** 和 **end** 关键字
- 添加了一个新的 `yield_self` 方法，以返回块的结果
- 添加了对分支覆盖和方法覆盖测量的支持
- 实施了新的 `Hash#slice` 和 `Hash#transform_keys` 方法
- **struct** 子类构造器可以接受关键字参数

- 顶级常量查找不再可用

Ruby 2.5.0 维护与 Ruby 2.4 的源级向后兼容性。

有关 Ruby 2.5.0 的详细更改，请参阅 [上游发行注记](#)。

### 1.3.5. MongoDB 的变化

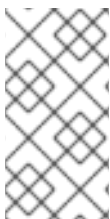
新的 rh-mongodb36 Software Collection 包括 **MongoDB 3.6.3**，它提供很多程序错误修复和增强。最显著的新功能包括：

- 更改流，使应用程序能够访问实时数据更改并立即对它们做出反应
- 可重试写入，使 MongoDB 驱动程序自动重试某些写入操作，例如在出现网络错误时
- JSON 架构，允许用户自行定义验证 JSON 文档的条件

以下子软件包也已被升级：

- mongo-c-driver 更新到版本 1.9.2
- mongo-cxx-driver 更新到版本 3.2.0
- mongo-tools 更新到版本 3.6.3
- mongo-java-driver 更新到版本 3.6.3

有关 MongoDB 3.6 的详细信息，请参阅 [上游发行注记](#)。



#### 注意

rh-mongodb36-mongo-cxx-driver 软件包已使用 Red Hat Developer Toolset 6 中的 **GCC** 使用 **-std=gnu++14** 选项构建。对使用 C++11（或更新版本）功能的 MongoDB C++ 驱动程序使用共享库的二进制文件必须使用 Red Hat Developer Toolset 6 或更高版本构建。请参阅 [Red Hat Developer Toolset 6 用户指南中的 C++ 兼容性详情](#)。

rh-mongodb36 Software Collection 包括 rh-mongodb36-syspaths 软件包，该软件包安装为二进制文件、脚本、手册页和其他提供系统范围打包程序的软件包。安装 rh-mongodb36\*-syspaths 软件包后，用户不需要使用 **scl enable** 命令对 rh-mongodb36\* 软件包提供的二进制文件和脚本正常工作。要了解有关 syspaths 的更多信息，请参阅 [Red Hat Software Collections Packaging Guide](#)。

有关迁移的步骤，请参阅 [第 5.2 节“迁移到 MongoDB 3.6”](#)。

### 1.3.6. PostgreSQL 中的更改

新的 rh-postgresql10 Software Collection 提供 **PostgreSQL 10.3**。这个版本的主要改进包括：

- 使用 **publish** 和 **subscribe** 关键字的逻辑复制
- 基于 **SCRAM-SHA-256** 机制的更强大的密码身份验证
- 声明性表分区
- 改进了查询并行性
- 显著的常规性能改进

- 改进了监控和控制

有关 PostgreSQL 10.3 的详细信息，请查看 [上游 documentation](#)。

rh-postgresql10 Software Collection 包括 rh-postgresql10-syspaths 软件包，该软件包会安装为二进制文件、脚本、手册页和其他提供系统范围打包程序的软件包。安装 rh-postgresql10\*-syspaths 软件包后，用户不需要使用 **scl enable** 命令对 rh-postgresql10\* 软件包提供的二进制文件和脚本正常工作。请注意，\*-syspaths 软件包与基本 Red Hat Enterprise Linux 系统中的对应软件包冲突。要了解有关 syspaths 的更多信息，请参阅 [Red Hat Software Collections Packaging Guide](#)。

有关迁移的信息，请参考 [第 5.5 节“迁移到 PostgreSQL 10”](#)。

### 1.3.7. Varnish Cache 中的更改

新的 rh-varnish5 Software Collection 包括 **Varnish Cache 5.2.1**，它提供很多程序错误修复和增强。值得注意的是：

- 通过单独的 Varnish 配置语言(VCL)文件和 VCL 标签对配置的改进，这可能会降低配置的复杂性
- 实验性的 HTTP/2 支持
- **分片** director
- 提高了稳定性

有关 Varnish Cache 5.2.1 中的详细 [更改](#)，请参阅 [上游更改日志](#)。另请参阅 [上游文档和升级 备注](#)。

### 1.3.8. HAProxy 的更改

新的 rh-haproxy18 Software Collection 提供 **HAProxy 1.8.4**。这个版本中最显著的变化包括：

- HTTP 请求和响应捕获
- Multiprocess peers 和 stick-tables
- 在运行时使用 DNS 的服务器 IP 解析
- 电子邮件警报
- 无缝服务器状态
- HTTP/2 支持（实验性）
- 多线程（实验性）
- [无缝重新加载](#)
- [用于服务发现的 DNS](#)
- [改进了 Runtime API](#)

有关 HAProxy 自版本 1.5 起（包括在 Red Hat Enterprise Linux 7 中）的详情，请查看 [HAProxy 配置手册](#)。

### 1.3.9. PHP 中的更改

rh-php70 Software Collection 已升级至版本 7.0.27，与 Red Hat Software Collections 2.3 中发布的版本相比，它提供了很多程序错误修复和增强。

有关本发行版本中 bug 修复和增强的详细信息，请参阅 [上游更改日志 7.0.27](#) 及更早版本。

### 1.3.10. MySQL 的更改

在这个版本中，在 rh-mysql57 Software Collection 中添加了对以下架构的支持：

- 64 位 ARM 架构
- IBM z 系统
- IBM POWER, little endian

### 1.3.11. Apache httpd 中的更改

在这个版本中，httpd24-mod\_auth\_mellon 软件包添加到 httpd24 Software Collection 中。Apache HTTP 服务器的 **mod\_auth\_mellon** 模块是实现 SAML 2.0 联邦协议的身份验证服务。该模块根据身份提供程序(IdP)服务器生成的断言中收到的属性授予访问权限。

## 1.4. 兼容性信息

Red Hat Software Collections 3.1 可用于 AMD64 和 Intel 64 架构、64 位 ARM 架构、IBM z Systems 和 IBM POWER, little endian 的所有支持的 Red Hat Enterprise Linux 7 版本。

某些组件还可用于 AMD64 和 Intel 64 构架中所有支持的 Red Hat Enterprise Linux 6 版本。

有关可用组件的完整列表，请参阅 [表 1.2 “所有可用的 Software Collections”](#)。

## 1.5. 已知问题

### httpd24 component, BZ#1429006

从 httpd 2.4.27 开始，默认的预处理模块(MPM)不再支持 **mod\_http2** 模块。要启用 HTTP/2 支持，请编辑位于 `/opt/rh/httpd24/root/etc/httpd/conf.modules.d/00-mpm.conf` 的配置文件，并切换到 **事件** 或 **worker MPM**。

请注意，HTTP/2 server-push 功能不适用于 64 位 ARM 架构、IBM z Systems 和 IBM POWER, little endian。

### httpd24 component, BZ#1327548

**mod\_ssl** 模块不支持 Red Hat Enterprise Linux 6 或 Red Hat Enterprise Linux 7.3 及更早版本上的 ALPN 协议。因此，只支持使用 ALPN 将 TLS 连接升级到 HTTP/2 的客户端仅限于 HTTP/1.1 支持。

### httpd24 component, BZ#1224763

当在 FastCGI Process Manager (PHP-FPM)中使用 **mod\_proxy\_fcgi** 模块时，httpd 默认使用

端口 8000 作为 FastCGI 协议，而不是正确的端口 9000。要临时解决这个问题，请在配置中明确指定正确的端口。

#### httpd24 组件, BZ#1382706

启用 SELinux 后，LD\_LIBRARY\_PATH 环境变量不会传递给 httpd 调用的 CGI 脚本。因此，在某些情况下，无法从 httpd 运行的 CGI 脚本中的 /opt/rh/httpd24/service-environment 文件中调用启用的 Software Collections 的可执行文件。要临时解决这个问题，请从 CGI 脚本中设置 LD\_LIBRARY\_PATH。

#### httpd24 组件

不支持针对 httpd24 软件集中的 Apache Portable Runtime (APR) 和 APR-util 库编译外部应用程序。httpd24 中没有设置 LD\_LIBRARY\_PATH 环境变量，因此此 Software Collection 中的任何应用程序都不需要它。

#### rh-python34, rh-python35, rh-python36 components, BZ#1499990

Babel 用于时区支持的 pytz 模块不包含在 rh-python34、rh-python35 和 rh-python36 Software Collections 中。因此，当用户尝试从 Babel 导入日期模块时，会返回回溯。要临时解决这个问题，请使用 pip install pytz 命令从 pypi 公共存储库安装 pytz。

#### rh-python36 组件

numpy 提供的某些复杂的 trigonometric 功能可能会返回 64 位 ARM 架构、IBM z Systems 和 IBM POWER, little endian 上的不正确的值。AMD64 和 Intel 64 架构不受此问题的影响。

#### python27 component, BZ#1330489

python27-python-pymongo 软件包已更新至版本 3.2.1。请注意，此版本与之前附带的版本 2.5.2 并不完全兼容。

#### python27 组件

在 Red Hat Enterprise Linux 7 中，当用户试图安装 python27-python-debuginfo 软件包时，/usr/src/debug/Python-2.7.5/Modules/socketmodule.c 文件与核心系统上安装的 python-debuginfo 软件包的对应文件冲突。因此，安装 python27-python-debuginfo 会失败。要临时解决这个问题，请卸载 python-debuginfo 软件包，然后安装 python27-python-debuginfo 软件包。

#### Scl-utils 组件

由于 `scl-utils` 软件包中的特定于架构的宏错误，`<collection>/root/usr/lib64/` 目录在 64 位 ARM 架构和 IBM POWER, little endian 上没有正确的软件包所有权。因此，当卸载 Software Collection 时，此目录不会被删除。要临时解决这个问题，在删除 Software Collection 时手动删除 `<collection>/root/usr/lib64/`。

### rh-ruby24,rh-ruby23 组件

确定 RubyGem 安装路径取决于启用多个 Software Collections 的顺序。自 Red Hat Software Collections 2.3 中提供的 Ruby 2.3.1 开始，所需的顺序已被修改，以支持依赖集合。因此，当以不正确的顺序提供 Software Collections 时，用于 gem 安装的 RubyGem 路径无效。例如，如果 RPM spec 文件包含 `scl enable rh-ror50 rh-nodejs6`，构建现在会失败。要临时解决这个问题，请启用 `rh-ror50` Software Collection 最后，例如 `scl enable rh-nodejs6 rh-ror50`。

### rh-maven35,rh-maven33 组件

当用户同时安装了 `maven-local` 软件包的 Red Hat Enterprise Linux 系统版本，并且 `rh-maven35-maven-local` 软件包或 `rh-maven33-maven-local` 软件包时，`XMvn` 用于构建 Java RPM 软件包的工具，从 `rh-maven35` 或 `rh-maven33` Software Collection 运行。要临时解决这个问题，请从基本 Red Hat Enterprise Linux 系统中卸载 `maven-local` 软件包。

### rh-nodejs4 组件, BZ#[1316626](#)

`/opt/rh/rh-nodejs4/root/usr/share/licenses/` 目录都归任何软件包所有。因此，当卸载 `rh-nodejs4` 集合时，此目录不会被删除。要临时解决这个问题，请在卸载 `rh-nodejs4` 后手动删除目录。

### Perl 组件

无法安装多个 `mod_perl.so` 库。因此，无法使用多个 Perl Software Collection 中的 `mod_perl` 模块。

### nodejs010 组件

`nodejs010` Software Collection 提供的共享库（即 `libcares`、`libhttp_parser` 和 `libuv`）没有以集合名称正确前缀。因此，可能会出现与对应的系统库冲突。

### nodejs-hawk 组件

`nodejs-hawk` 软件包使用了来自 `CryptoJS` 项目的 SHA-1 和 SHA-256 算法的实现。在本发行版本中，客户端 JavaScript 会模糊处理。将来的修复将涉及直接从 `CryptoJS` 库使用加密功能。



## PostgreSQL 组件

Red Hat Enterprise Linux 6 的 postgresql92、rh-postgresql94 和 rh-postgresql95 软件包不提供 sepgsql 模块，因为此功能需要安装 libselinux 版本 2.0.99，这在 Red Hat Enterprise Linux 6 中不可用。

httpd,mariadb,mongodb,mysql,nodejs, nodejs ,perl,php55,rh-php56,python , ruby, ror ,thermostat, 和 v8314 组件, BZ#1072319

卸载 httpd 24,mariadb55,rh-mariadb100,mongodb24,rh-mongodb26,mysql55,rh-mysql56,nodejs010,perl516,rh-perl520,php55,rh-php56, python27,python33,rh-python34,ruby193, ruby193 ,ruby200,rh-ruby22 ,ror40,rh-ror41,rmostat 1, 或 v8314 软件包可能会因为依赖软件包的所有权而相关。因此，一些目录和文件可能无法被正确删除，并可能在系统中保留。

rh-mysql57,rh-mysql56,rh-mariadb100,rh-mariadb101 组件, BZ#1194611

rh-mysql57-mysql-server、rh-mysql56-mysql-server、rh-mariadb100- mariadb-server 和 rh-mariadb.1.1.mariadb-server 软件包不再默认提供 test 数据库。虽然在初始化过程中不会创建此数据库，但授权表会预先填充与默认创建测试时相同的值。因此，在稍后创建 test 或 testzFCP 数据库时，这些数据库的访问权限比新数据库的默认权限要小。

另外，在运行基准时，run-all-tests 脚本不再可用于示例参数。您需要在运行测试前创建一个测试数据库，并在 --database 参数中指定数据库名称。如果未指定该参数，则默认执行测试，但需要确保 test 数据库存在。

## mongodb24 组件

Red Hat Software Collections 1.2 的 mongodb24 Software Collection 无法通过 Red Hat Software Collections 3.1 提供的 rh-java-common 和 maven30 Software Collections 重新构建。另外，由于 maven30-javapackages-tools 和 maven30-maven- local 软件包需要不满意，所以 mongodb24-build 和 mongodb24-scldevel 软件包无法被安装。安装 mongodb24-scldevel 软件包时，会报告有问题的依赖项，yum --skip-broken 命令会跳过太多的软件包。建议用户更新到 rh-mongodb26 Software Collection。

## MariaDB,mysql,postgresql,mongodb 组件

Red Hat Software Collections 3.1 包含 MySQL 5.6,MySQL 5.7,MariaDB 10.0,MariaDB 10.1,MariaDB 10.2,PostgreSQL 9.4,PostgreSQL 9.5,PostgreSQL 9.6,PostgreSQL 10,MongoDB 2.6,MongoDB 3.2,MongoDB 3.4, 和 MongoDB 3.6 数据库。Red Hat Enterprise Linux 6 的核心版本提供了 MySQL 和 PostgreSQL 数据库的早期版本（客户端库和守护进程）。Red Hat Enterprise

Linux 7 的核心提供 MariaDB 和 PostgreSQL 数据库的早期版本（客户端库和守护进程）。客户端库也用于动态语言、库等的数据库连接器。

PostgreSQL 组件中打包的 Red Hat Software Collections 数据库软件包的客户端库不应被使用，因为它仅用于服务器实用程序和守护进程。用户需要使用由核心系统提供的系统库和数据库连接器。

客户端库和守护进程之间使用的协议在数据库版本之间保持稳定，例如，使用带有 PostgreSQL 9.4 或 9.5 守护进程的 PostgreSQL 9.2 客户端库可以正常工作。

Red Hat Enterprise Linux 6 和 Red Hat Enterprise Linux 7 的核心不包括 MongoDB 的客户端库。要将这个客户端库用于应用程序，您应该使用 Red Hat Software Collections 中的客户端库，并总是使用 `scl enable ...` 调用，每次运行链接到此 MongoDB 客户端库的应用程序时。

### MariaDB,mysql,mongodb 组件

在创建日志文件时，MariaDB、MySQL 和 MongoDB 不使用 `/opt/provider/collection/root` 前缀。请注意，日志文件保存在 `/var/opt/provider/collection/log/` 目录中，而不是在 `/opt/provider/collection/root/var/log/` 中。

### rh-eclipse46 组件

当安装第三方更新站点的插件时，Eclipse 有时无法在工作区日志文件中以 `NullPointerException` 开始。要临时解决这个问题，请使用 `-clean` 选项重启 Eclipse。例如：

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

### rh-eclipse46 组件

Eclipse Docker 工具引入了一个 Dockerfile 编辑器，其语法突出显示和基本的命令自动完成功能。打开 Build Image 向导并按下 Edit Dockerfile 按钮时，Dockerfile 编辑器会在分离的编辑器窗口中打开该文件。但是，此窗口不包含 Cancel 和 Save 按钮。要临时解决这个问题，请按 `Ctrl+S` 保存您的更改，或者右键单击编辑器以启动上下文菜单，该菜单提供 Save 选项。要取消您的更改，请关闭窗口。

### rh-eclipse46 组件

在 Red Hat Enterprise Linux 7.2 中，perf 工具中的一个错误，用于在 Eclipse 中填充 Perf Profile View，这会导致视图中的一些项目没有正确链接到 Eclipse Editor 中的相应位置。虽然性能

分析按预期工作，但无法通过单击 **Perl Profile View** 的部分来导航到编辑器中的相关位置。

## rh-thermostat16 组件

由于桌面应用程序文件中的拼写错误，用户无法使用桌面图标启动 **Thermostat**。要临时解决这个问题，请从以下修改 `/usr/share/applications/rh-thermostat16-thermostat.desktop` 文件：

```
[Desktop Entry]
Version=1.0
Type=Application
Name=%{thermostat_desktop_app_name}
Comment=A monitoring and serviceability tool for OpenJDK
Exec=/opt/rh/rh-thermostat16/root/usr/share/thermostat/bin/thermostat local
Icon=thermostat
```

至：

```
[Desktop Entry]
Version=1.0
Type=Application
Name=Thermostat-1.6
Comment=A monitoring and serviceability tool for OpenJDK
Exec=scl enable rh-thermostat16 "thermostat local"
Icon=rh-thermostat16-thermostat
```

或者，从命令行运行 **Thermostat**：

```
$ scl enable rh-thermostat16 "thermostat local"
```

## 其他备注

### rh-ruby22,rh-ruby23,rh-python34,rh-python35,rh-php56,rh-php70 组件

在只读 **NFS** 中使用 **Software Collections** 有一些限制。

- 

当 **rh-ruby22** 或 **rh-ruby23** **Software Collection** 位于只读 **NFS** 时，无法安装 **Ruby gems**。因此，当用户尝试使用 `gem install ab` 命令安装 **ab gem** 时，会显示出错信息，例如：

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

当用户试图通过运行捆绑包更新或捆绑包 `install` 命令从外部源更新或安装 `gems` 时，会出现相同的问题。

- 当使用 Python Package Index (PyPI) 在只读 NFS 中安装 Python 软件包时，运行 `pip` 命令会失败并显示类似如下的错误消息：

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

- 使用 `pear` 命令在只读 NFS 上从 PHP 扩展和应用程序存储库(PEAR)安装软件包会失败，并显示出错信息：

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

这是预期的行为。

## httpd 组件

Apache 的语言模块仅支持 Apache httpd 的红帽软件集合版本，而不支持使用 Red Hat Enterprise Linux 系统版本的 httpd。例如，`rh-python35` Collection 中的 `mod_wsgi` 模块只能用于 `httpd24` Collection。

## 所有组件

从 Red Hat Software Collections 2.0 开始，单个集合的配置文件、变量数据和运行时数据存储在与之前版本的 Red Hat Software Collections 不同的目录中。

## coreutils,util-linux,屏幕 组件

有些实用程序（如 `su`、`login` 或 `screen`）不会在所有情况下都导出环境设置，这可能会导致意外的结果。因此，建议您使用 `sudo` 而不是 `su`，并在 `/etc/sudoers` 文件中设置 `env_keep` 环境变量。或者，您可以按相反的顺序运行命令，例如：

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

而不是

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

当使用 屏幕 或 登录 等工具时，您可以使用以下命令保留环境设置：

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

## php54 组件

请注意，Red Hat Software Collections 中的替代 PHP Cache (APC) 只提供用户数据缓存。对于 opcode 缓存，提供了 Zend OPcache。

## Python 组件

当用户尝试从 python27、python33、rh-python34 和 rh-python35 Software Collections 中安装多个 `scldevel` 软件包时，会返回一个事务检查错误消息。这是一个预期的行为，因为用户只能安装由软件包提供的一组宏文件(`%scl_python`、`%scl_prefix_python`)。

## PHP 组件

当用户尝试从 php54、php55、rh-php56 和 rh-php70 Software Collections 中安装多个 `scldevel` 软件包时，会返回一个事务检查错误消息。这是一个预期的行为，因为用户只能安装由软件包提供的一组宏文件(`%scl_php`、`%scl_prefix_php`)。

## Ruby 组件

当用户尝试从 ruby193、ruby200、rh-ruby22 和 rh-ruby23 Software Collections 中安装多个 `scldevel` 软件包时，会返回一个事务检查错误消息。这是一个预期的行为，因为用户只能安装由软件包提供的一组宏文件(`%scl_ruby`、`%scl_prefix_ruby`)。

## Perl 组件

当用户尝试从 perl516、rh-perl520 和 rh-perl524 Software Collections 中安装多个 `scldevel` 软件包时，会返回事务检查错误消息。这是一个预期的行为，因为用户只能安装由软件包提供的一组宏文件(`%scl_perl`、`%scl_prefix_perl`)。

## Nginx 组件

当用户尝试从 nginx16 和 rh-nginx18 Software Collections 中安装多个 `scldevel` 软件包时，会返回事务检查错误消息。这是一个预期的行为，因为用户只能安装由软件包提供的一组宏文件(`%scl_nginx`、`%scl_prefix_nginx`)。

## nodejs 组件

安装 `nodejs010 Software Collection` 时，`nodejs010` 在基本 Red Hat Enterprise Linux 系统中安装 GCC 作为依赖项，除非已安装了 `gcc` 软件包。

## rh-eclipse46 component

Red Hat Enterprise Linux 7 上的 Eclipse SWT 图形库使用 GTK 3.x。在 GTK 3.x 上，Eclipse Dark Theme 还没有完全稳定，因此这个主题被视为技术预览且不被支持。有关红帽技术预览的更多信息，请参阅 <https://access.redhat.com/support/offerings/techpreview/>。

## 1.6. 已弃用的功能

### httpd24 component, BZ#1434053

在以前的版本中，在需要基于名称的 SSL 虚拟主机选择的 SSL/TLS 配置中，`mod_ssl` 模块会拒绝带有 400 Bad Request 错误的请求，如果 `Host:` 标头中提供的主机名与 `Server Name Indication (SNI)` 标头中提供的主机名不匹配。如果在所选虚拟主机之间配置的 SSL/TLS 安全参数与上游 `mod_ssl` 的行为相同，则此类请求不再被拒绝。

## 第 2 章 安装

本章论述了如何访问内容集、在系统中安装 Red Hat Software Collections 3.1 并重新构建 Red Hat Software Collections。

### 2.1. 访问 RED HAT SOFTWARE COLLECTIONS

Red Hat Software Collections 内容集可供使用 Red Hat Enterprise Linux 6 和 Red Hat Enterprise Linux 7 订阅的客户，网址为：<https://access.redhat.com/solutions/472793>有关如何使用 Red Hat Subscription Management (RHSM)注册您的系统的详情，请参考 [使用和配置 Red Hat Subscription Manager](#)。有关如何使用 RHSM 启用 Red Hat Software Collections 的详情，请参考 [第 2.1.1 节“使用 Red Hat Subscription Management”](#)。

从 Red Hat Software Collections 2.2 开始 <https://access.redhat.com/downloads>，Red Hat Software Collections 和 Red Hat Developer Toolset 内容也以 ISO 格式提供，特别是用于 [服务器](#) 和 [Workstation](#)。请注意，需要 Optional 频道的软件包（在 [第 2.1.2 节“Optional 频道中的软件包”](#) 中列出的）无法从 ISO 镜像安装。



#### 注意

需要 Optional 频道的软件包无法从 ISO 镜像安装。[第 2.1.2 节“Optional 频道中的软件包”](#) 中提供了需要启用 Optional 频道的软件包列表。

Beta 内容在 ISO 格式不可用。

#### 2.1.1. 使用 Red Hat Subscription Management

如果您的系统使用 Red Hat Subscription Management 注册，请完成以下步骤附加订阅，该订阅提供对 Red Hat Software Collections 的访问权限并启用存储库：

1. 显示您的系统所有可用订阅的列表，并确定提供 Red Hat Software Collections 的订阅池 ID。要做到这一点，以 root 用户身份在 shell 提示符后输入以下内容：

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

对于每个可用订阅，这个命令会显示其名称、唯一标识符、过期日期和其他与它相关的详细信息。池 ID 在以 **Pool Id** 开头的行上列出。

2.

以 **root** 用户身份运行以下命令，为您的系统附加适当的订阅：

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

将 *pool\_id* 替换为您在上一步中确定的池 ID。要验证系统当前附加的订阅列表，以 **root** 用户身份输入：

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

3.

显示可用的 Yum 列表存储库列表，以检索存储库元数据并确定 Red Hat Software Collections 存储库的确切名称。以 **root** 用户身份，输入：

```
subscription-manager repos --list
```

或者，对简短列表运行 `yum repolist all`。

软件仓库名称取决于您使用的 Red Hat Enterprise Linux 的特定版本，采用以下格式：

```
rhel-variant-rhsc1-6-rpms  
rhel-variant-rhsc1-6-debug-rpms  
rhel-variant-rhsc1-6-source-rpms  
  
rhel-server-rhsc1-6-eus-rpms  
rhel-server-rhsc1-6-eus-source-rpms  
rhel-server-rhsc1-6-eus-debug-rpms  
  
rhel-variant-rhsc1-7-rpms  
rhel-variant-rhsc1-7-debug-rpms  
rhel-variant-rhsc1-7-source-rpms  
  
rhel-server-rhsc1-7-eus-rpms  
rhel-server-rhsc1-7-eus-source-rpms  
rhel-server-rhsc1-7-eus-debug-rpms
```

使用 Red Hat Enterprise Linux 系统变体（即 **server** 或 **workstation**）替换 *variant*。请注意，在客户端或 **ComputeNode** 变体上都不支持 Red Hat Software Collections。



4.

以 root 用户身份运行以下命令来启用适当的存储库：

```
subscription-manager repos --enable repository
```

订阅附加到系统后，您可以安装 Red Hat Software Collections，如第 2.2 节“安装 Red Hat Software Collections”所述。有关如何使用红帽订阅管理注册您的系统并将其与订阅相关联的更多信息，请参阅使用和配置 Red Hat Subscription Manager。



注意

通过 RHN 订阅不再可用。

### 2.1.2. Optional 频道中的软件包

有些 Red Hat Software Collections 3.1 软件包需要启用 Optional 频道才能完成这些软件包的完整安装。有关如何向这个频道订阅您的系统的详情，请查看 Red Hat Subscription Management 的相关知识库文章 <https://access.redhat.com/solutions/392003>，或者 <https://access.redhat.com/solutions/70019> 您的系统使用 RHN Classic 注册。

下表中列出了需要启用 Optional 频道的 Software Collections for Red Hat Enterprise Linux 6 中的软件包。

表 2.1. 在 Red Hat Enterprise Linux 6 中，需要启用 Optional Channel 的软件包

Software Collection 中的软件包	Optional 频道中所需的软件包
devtoolset-6-dyninst-testsuite	glibc-static
devtoolset-7-dyninst-testsuite	glibc-static
rh-git29-git-all	cvsp, perl-Net-SMTP-SSL
rh-git29-git-cvs	cvsp
rh-git29-git-email	perl-Net-SMTP-SSL
rh-git29-perl-Git-SVN	perl-YAML, subversion-perl
rh-mariadb101-boost-devel	libicu-devel
rh-mariadb101-boost-examples	libicu-devel

Software Collection 中的软件包	Optional 频道中所需的软件包
rh-mariadb101-boost-static	libcicu-devel
rh-mongodb30upg-boost-devel	libcicu-devel
rh-mongodb30upg-boost-examples	libcicu-devel
rh-mongodb30upg-boost-static	libcicu-devel
rh-mongodb30upg-yaml-cpp-devel	libcicu-devel
rh-mongodb32-boost-devel	libcicu-devel
rh-mongodb32-boost-examples	libcicu-devel
rh-mongodb32-boost-static	libcicu-devel
rh-mongodb32-yaml-cpp-devel	libcicu-devel
rh-mongodb34-boost-devel	libcicu-devel
rh-mongodb34-boost-examples	libcicu-devel
rh-mongodb34-boost-static	libcicu-devel
rh-mongodb34-yaml-cpp-devel	libcicu-devel
rh-php56-php-imap	libc-client
rh-php56-php-recode	recode
rh-php70-php-imap	libc-client
rh-php70-php-recode	recode

下表列出了需要 Red Hat Enterprise Linux 7 中 Optional 频道的 Software Collections 软件包。

表 2.2. 在 Red Hat Enterprise Linux 7 中，需要启用 Optional Channel 的软件包

Software Collection 中的软件包	Optional 频道中所需的软件包
devtoolset-7-dyninst-testsuite	glibc-static

Software Collection 中的软件包	Optional 频道中所需的软件包
devtoolset-7-gcc-plugin-devel	libmpc-devel
httpd24-mod_ldap	apr-util-ldap
rh-eclipse46	ruby-doc
rh-eclipse46-eclipse-dltk-ruby	ruby-doc
rh-eclipse46-eclipse-dltk-sdk	ruby-doc
rh-eclipse46-eclipse-dltk-tests	ruby-doc
rh-git29-git-all	cvsp
rh-git29-git-cvs	cvsp
rh-git29-perl-Git-SVN	subversion-perl
rh-perl520-perl-Pod-Perldoc	groff

请注意，不支持 Optional 频道中的软件包。详情请查看 知识库文章。 <https://access.redhat.com/articles/1150793>

## 2.2. 安装 RED HAT SOFTWARE COLLECTIONS

Red Hat Software Collections 作为 RPM 软件包的集合，可以使用 Red Hat Enterprise Linux 中包含的标准软件包管理工具安装、更新和卸载。请注意，需要一个有效的订阅才能在您的系统上安装 Red Hat Software Collections。有关如何将您的系统与适当的订阅关联并可以访问 Red Hat Software Collections 的详细信息，请参阅 第 2.1 节“访问 Red Hat Software Collections”。

使用 Red Hat Software Collections 3.1 需要删除任何早期的预发布版本，包括 Beta 版本。如果您已安装任何版本的 Red Hat Software Collections 3.1，请从您的系统中卸载它并安装新版本，如 第 2.3 节“卸载 Red Hat Software Collections”和 第 2.2.1 节“安装单个 Software Collections”部分所述。

Red Hat Software Collections 不支持从 Red Hat Enterprise Linux 6 原位升级到 Red Hat Enterprise Linux 7。因此，在升级后，安装的 Software Collections 可能无法正常工作。如果要从 Red Hat Enterprise Linux 6 升级到 Red Hat Enterprise Linux 7，强烈建议删除所有 Red Hat Software Collections 软件包，执行原位升级，更新 Red Hat Software Collections 软件仓库，并再次安装 Software Collections 软件包。建议您在升级前备份所有数据。

### 2.2.1. 安装单个 Software Collections

要安装 [表 1.1 “Red Hat Software Collections 3.1 组件”](#) 中列出的任何 Software Collections，请以 root 用户身份在 shell 提示符后输入以下内容来安装对应的 meta 软件包：

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

使用您要安装的 Software Collections 列表替换 *software\_collection*。例如，要安装 php54 和 rh-mariadb100，请以 root 用户身份键入：

```
~]# yum install rh-php56 rh-mariadb100
```

这会为所选 Software Collection 安装主 meta 软件包，以及一组所需软件包作为其依赖项。有关如何安装其他软件包（如附加模块）的详情，请参考 [第 2.2.2 节 “安装可选软件包”](#)。

### 2.2.2. 安装可选软件包

Red Hat Software Collections 的每个组件都带有很多默认未安装的可选软件包。要列出属于特定 Software Collection 但未在您的系统中安装的所有软件包，在 shell 提示符下输入以下内容：

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

要安装这些可选软件包，以 root 用户身份键入：

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

使用您要安装的软件包列表替换 *package\_name*。例如，要安装 rh-perl520-perl-CPAN 和 rh-perl520-perl-Archive-Tar，请输入：

```
~]# yum install rh-perl524-perl-CPAN rh-perl524-perl-Archive-Tar
```

### 2.2.3. 安装调试信息

要为任何 Red Hat Software Collections 软件包安装调试信息，请确保已安装 yum-utils 软件包，并以 root 用户身份运行以下命令：

```
debuginfo-install package_name
```

例如，要安装 `rh-ruby22-ruby` 软件包的调试信息，请输入：

```
~]# debuginfo-install rh-ruby22-ruby
```

请注意，您需要使用这些软件包访问存储库。如果您的系统通过红帽订阅管理注册，请启用 `rhel-variant-rhsc1-6-debug-rpms` 或 `rhel-variant-rhsc1-7-debug-rpms` 存储库，如第 2.1.1 节“使用 Red Hat Subscription Management”所述。有关如何访问 `debuginfo` 软件包的更多信息，请参阅 <https://access.redhat.com/solutions/9907>。

### 2.3. 卸载 RED HAT SOFTWARE COLLECTIONS

要卸载任何 Software Collections 组件，以 root 用户身份在 shell 提示符后输入以下内容：

```
yum remove software_collection*
```

使用您要卸载的 Software Collection 组件替换 `software_collection`。

请注意，卸载 Red Hat Software Collections 提供的软件包不会影响这些工具的 Red Hat Enterprise Linux 系统版本。

### 2.4. 重建 RED HAT SOFTWARE COLLECTIONS

`<collection>-build` 软件包默认不提供。如果要重建集合，且不想使用 `rpmbuild --define 'scl foo'` 命令，您首先需要重新构建 `metapackage`，它提供 `<collection>-build` 软件包。

请注意，现有集合不应使用不同的内容重建。要将新软件包添加到现有集合中，您需要创建一个包含新软件包的新集合，并使其依赖于原始集合中的软件包。必须在不更改的情况下使用原始集合。

有关构建软件集合的详细信息，请参阅 [Red Hat Software Collections 打包指南](#)。

## 第 3 章 使用方法

本章论述了重建和使用 Red Hat Software Collections 3.1 所需的步骤，以及部署使用 Red Hat Software Collections 的应用程序。

### 3.1. 使用 RED HAT SOFTWARE COLLECTIONS

#### 3.1.1. 从 Software Collection 运行可执行文件

要从特定的 Software Collection 运行可执行文件，在 shell 提示符下输入以下命令：

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

或者，使用以下命令：

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

使用您要使用的 Software Collections 的空格分隔列表替换 *software\_collection*，使用您要运行的命令替换 *command*。例如，要执行存储在名为 `hello.pl` 的 Perl 程序以及 `perl516` 软件集中的 Perl 解释器，请输入：

```
~]$ scl enable rh-perl524 'perl hello.pl'  
Hello, World!
```

您可以使用 `scl` 工具执行任何命令，从而导致它使用所选 Software Collection 的可执行文件运行，并优先选择其对应的 Red Hat Enterprise Linux 系统。有关 Red Hat Software Collections 附带的 Software Collections 的完整列表，请参阅 [表 1.1 “Red Hat Software Collections 3.1 组件”](#)。

#### 3.1.2. 使用 Software Collection 作为默认值运行 Shell 会话

要启动一个新的 shell 会话，它带有来自所选 Software Collection 的可执行文件，首选使用 Red Hat Enterprise Linux 等效的，在 shell 提示符后输入以下内容：

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

使用您要使用的 Software Collections 列表替换 *software\_collection*。例如，要启动带有 `python27` 和 `rh-postgresql95` Software Collections 的新 shell 会话，请输入：

```
~]$ scl enable python27 rh-postgresql95 bash
```

在当前会话中启用的 **Software Collections** 列表存储在 `$X_SCLS` 环境变量中，例如：

```
~]$ echo $X_SCLS
python27 rh-postgresql95
```

有关 Red Hat Software Collections 附带的 Software Collections 的完整列表，请参阅 [表 1.1 “Red Hat Software Collections 3.1 组件”](#)。

### 3.1.3. 从 Software Collection 中运行系统服务

包括系统服务的软件集合在 `/etc/rc.d/init.d/` 目录中安装相应的初始化脚本。要在当前会话中启动该服务，以 `root` 用户身份在 `shell` 提示符后输入以下内容：

```
service software_collection-service_name start
```

将 `software_collection` 替换为 Software Collection 和 `service_name` 的名称，使用您要启动的服务名称。要将此服务配置为在引导时自动启动，请以 `root` 用户身份输入以下命令：

```
chkconfig software_collection-service_name on
```

例如，要从 `rh-postgresql 95 Software Collection` 启动 `postgresql` 服务，并在运行级别 2、3、4 和 5 中启用它，以 `root` 用户身份：

```
~]# service rh-postgresql95-postgresql start
Starting rh-postgresql95-postgresql service:           [ OK ]
~]# chkconfig rh-postgresql95-postgresql on
```

有关如何在 Red Hat Enterprise Linux 6 中管理系统服务的更多信息，请参阅 [Red Hat Enterprise Linux 6 部署指南](#)。有关 Red Hat Software Collections 附带的 Software Collections 的完整列表，请参阅 [表 1.1 “Red Hat Software Collections 3.1 组件”](#)。

### 3.2. 从 SOFTWARE COLLECTION 访问 MAN PAGE

每个 Software Collection 都包含描述此组件内容的一般手册页。每个手册页的名称都与组件相同，它位于 `/opt/rh` 目录中。

要阅读 **Software Collection** 的手册页，请输入以下命令：

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

将 **software\_collection** 替换为特定的 Red Hat Software Collections 组件。例如，要显示 **rh-mariadb101** 的手册页，请输入：

```
~]$ scl enable rh-mariadb101 "man rh-mariadb101"
```

### 3.3. 部署使用 RED HAT SOFTWARE COLLECTIONS 的应用程序

通常，您可以使用以下两种方法部署依赖于生产中红帽软件集合的组件的应用程序：

- 手动安装所有必需的 **Software Collections** 和软件包，然后部署应用程序，或者
- 为您的应用程序创建一个新的 **Software Collection**，并将所有必需的 **Software Collections** 和其他软件包指定为依赖项。

有关如何手动安装独立的 Red Hat Software Collections 组件的详情，请参考 [第 2.2 节“安装 Red Hat Software Collections”](#)。有关如何使用 Red Hat Software Collections 的详情，请参考 [第 3.1 节“使用 Red Hat Software Collections”](#)。有关如何创建自定义 **Software Collection** 或扩展现有软件集合的详细信息，请参阅 [Red Hat Software Collections Packaging Guide](#)。

### 3.4. RED HAT SOFTWARE COLLECTIONS 容器镜像

基于红帽软件集合的容器镜像包括应用程序、守护进程和数据库。镜像可以在 Red Hat Enterprise Linux 7 服务器和 Red Hat Enterprise Linux Atomic Host 上运行。有关其用法的详情，请参考 [使用 Red Hat Software Collections 3 容器镜像](#)。有关基于 Red Hat Software Collections 版本 2.4 及更早版本的容器镜像的详情，请参阅 [使用 Red Hat Software Collections 2 容器镜像](#)。

Red Hat Software Collections 3.1 提供了以下容器镜像：

- `rhsc/devtoolset-7-toolchain-rhel7`



- `rhsc/devtoolset-7-perftools-rhel7`
- `rhsc/httpd-24-rhel7`
- `rhsc/mongodb-36-rhel7`
- `rhsc/perl-526-rhel7`
- `rhsc/php-70-rhel7`
- `rhsc/postgresql-10-rhel7`
- `rhsc/ruby-25-rhel7`
- `rhsc/varnish-5-rhel7`

以下容器镜像基于 Red Hat Software Collections 3.0 :

- `rhsc/mariadb-102-rhel7`
- `rhsc/mongodb-34-rhel7`
- `rhsc/nginx-112-rhel7`
- `rhsc/nodejs-8-rhel7`
- `rhsc/php-71-rhel7`

- **rhsc/postgresql-96-rhel7**
- **rhsc/python-36-rhel7**

以下容器镜像基于 Red Hat Software Collections 2.4 :

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

以下容器镜像基于 Red Hat Software Collections 2.3 :

- **rhsc/mysql-57-rhel7**

- `rhsc1/perl-524-rhel7`
- `rhsc1/redis-32-rhel7`
- `rhsc1/mongodb-32-rhel7`
- `rhsc1/php-56-rhel7`
- `rhsc1/python-35-rhel7`
- `rhsc1/ruby-23-rhel7`

以下容器镜像基于 Red Hat Software Collections 2.2 :

- `rhsc1/devtoolset-4-toolchain-rhel7`
- `rhsc1/devtoolset-4-perftools-rhel7`
- `rhsc1/mariadb-101-rhel7`
- `rhsc1/nginx-18-rhel7`
- `rhsc1/nodejs-4-rhel7`
- `rhsc1/postgresql-95-rhel7`
- `rhsc1/ror-42-rhel7`

- **rhsc/thermostat-1-agent-rhel7 (EOL)**
- **rhsc/varnish-4-rhel7**

以下容器镜像基于 Red Hat Software Collections 2.0 :

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

不再支持标记为 **Life (EOL)**的镜像。

## 第 4 章 单个软件集合的具体信息

本章重点介绍了某些 **Software Collections** 的具体内容，并提供了有关这些组件的额外详情。

### 4.1. RED HAT DEVELOPER TOOLSET

**Red Hat Developer Toolset** 专为使用 **Red Hat Enterprise Linux** 平台的开发人员设计。**Red Hat Developer Toolset** 提供 **GNU Compiler Collection**、**GNU Debugger** 和其他开发、调试和性能监控工具的当前版本。与其他 **Software Collections** 类似，还有一组额外的工具安装到 `/opt/` 目录中。这些工具由用户使用提供的 `scl` 工具根据需要启用。与其他 **Software Collections** 类似，它们不会替换这些工具的 **Red Hat Enterprise Linux** 系统版本，除非使用 `scl` 程序显式调用这些系统版本，它们不会被使用。

有关功能的概述，请参阅 *Red Hat Developer Toolset 发行注记* 中的 [主要功能](#) 部分。

有关组件的完整列表，请参阅 *Red Hat Developer Toolset 用户指南* 中的 [Red Hat Developer Toolset 组件](#) 表。

请注意，自 **Red Hat Developer Toolset 3.1** 开始，**Red Hat Developer Toolset** 需要 **rh-java-common Software Collection**。

### 4.2. ECLIPSE 4.6.3

适用于 **Red Hat Enterprise Linux 7** 的 **rh-eclipse46 Software Collection** 包括 **Eclipse 4.6.3**，它基于 **Eclipse Foundation** 的 **Neon** 发行培训。之前，这个集成开发环境作为 **Red Hat Developer Toolset** 的一部分提供。请注意，**rh-eclipse46 Software Collection** 需要 **rh-java-common Collection**。

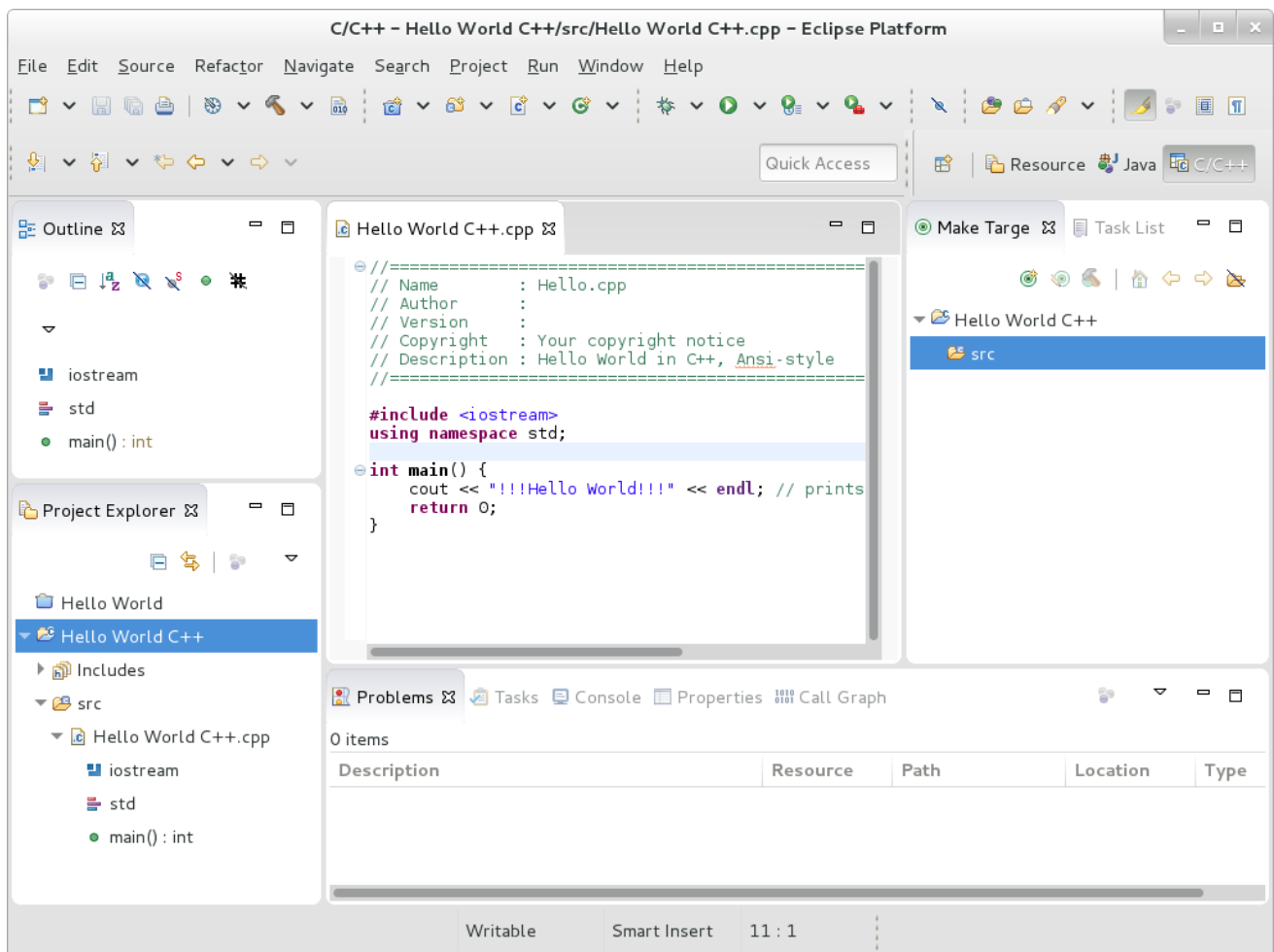


#### 注意

现在，作为 **Red Hat Developer Tools** 产品的 **rh-eclipse47** 组件提供了新版本的 **Eclipse**。如需更多信息，请参阅 [使用 Eclipse](#)。

**Eclipse** 是一个强大的开发环境，可为开发流程的每个阶段提供工具。它将各种不同的工具集成到统一环境中，以创建丰富的开发体验，提供完全可配置的用户界面，并提供可插拔式架构，允许以多种方式进行扩展。例如，**Valgrind** 插件允许编程人员通过 **Eclipse** 用户界面在命令行中执行内存性能分析。

图 4.1. Eclipse 会话示例



[D]

**Eclipse** 提供了与命令行工具的传统交互的替代图形开发环境，因此，对于不想使用命令行界面的开发人员来说，这是一个欢迎使用。传统、主要基于命令行的 Linux 工具套件（如 `gcc` 或 `gdb`）为编程提供了两种不同的编程方法。

请注意，如果您打算为红帽 JBoss 中间件开发应用程序或需要支持 OpenShift 工具，建议您使用 [红帽 JBoss Developer Studio](#)。

表 4.1. rh-eclipse46 软件集中包含的 Eclipse 组件

软件包	描述
rh-eclipse46-eclipse-cdt	C/C++ 开发工具(CDT)，它为使用 C 和 C++ 开发提供了功能和插件。
rh-eclipse46-eclipse-changelog	ChangeLog 插件，允许您创建和维护更改日志文件。
rh-eclipse46-eclipse-egit	EGit，EGit 是 Eclipse 的团队供应商，提供用于与 Git 存储库交互的功能和插件。

软件包	描述
rh-eclipse46-eclipse-emf	<b>Eclipse Modeling Framework (EMF)</b> , 允许您根据结构化数据模型构建应用程序。
rh-eclipse46-eclipse-epp-logging	Eclipse 错误报告工具。
rh-eclipse46-eclipse-gcov	GCov 插件将 <b>GCov</b> 测试覆盖程序与 <b>Eclipse</b> 集成。
rh-eclipse46-eclipse-gef	图形编辑框架(GEF), 允许您从现有应用程序模型创建丰富的图形编辑器。
rh-eclipse46-eclipse-gprof	Gprof 插件将 <b>Gprof</b> 性能分析实用程序与 <b>Eclipse</b> 集成。
rh-eclipse46-eclipse-jdt	<b>Eclipse Java</b> 开发工具(JDT)插件。
rh-eclipse46-eclipse-jgit	jgit, <b>Git</b> 修订控制系统的 Java 实施。
rh-eclipse46-eclipse-manpage	Man Page 插件, 允许您查看 <b>Eclipse</b> 中的手册页。
rh-eclipse46-eclipse-mpc	<b>Eclipse Marketplace</b> 客户端。
rh-eclipse46-eclipse-mylyn	mylyn 是 <b>Eclipse</b> 的任务管理系统。
rh-eclipse46-eclipse-oprofile	OProfile 插件, 它将 <b>OProfile</b> 与 <b>Eclipse</b> 集成。
rh-eclipse46-eclipse-pde	用于开发 <b>Eclipse</b> 插件的插件开发环境。
rh-eclipse46-eclipse-perf	Perf 插件, 它将 <b>perf</b> 工具与 <b>Eclipse</b> 集成。
rh-eclipse46-eclipse-ptp	为同步项目 <a href="#">提供支持的 PTP 项目</a> 的子集。
rh-eclipse46-eclipse-pydev	<b>Eclipse</b> 的完整功能 Python IDE。
rh-eclipse46-eclipse-remote	Remote Services 插件, 它提供可扩展的远程服务框架。
rh-eclipse46-eclipse-rpm-editor	<b>Eclipse Spec File Editor</b> , 允许您维护 RPM 规格文件。
rh-eclipse46-eclipse-rse	远程 System Explorer (RSE)框架, 允许您通过 <b>Eclipse</b> 使用远程系统。
rh-eclipse46-eclipse-systemtap	SystemTap 插件, 它将 <b>SystemTap</b> 与 <b>Eclipse</b> 集成。
rh-eclipse46-eclipse-valgrind	Valgrind 插件, 它将 <b>Valgrind</b> 与 <b>Eclipse</b> 集成。
rh-eclipse46-eclipse-webtools	<b>Eclipse Webtools</b> 插件。

#### 4.2.1. 安装 Eclipse



Eclipse 开发环境作为 RPM 软件包的集合提供。要安装 rh-eclipse46 软件集合，请以 root 用户身份运行以下命令：

```
yum install rh-eclipse46
```

有关可用组件列表，请参阅 [表 4.1 “rh-eclipse46 软件集合中包含的 Eclipse 组件”](#)。



#### 注意

rh-eclipse46 Software Collection 完全支持 C、C++ 和 Java 开发，但 *不提供*对 Fortran 编程语言的支持。

### 4.2.2. 使用 Eclipse

要启动 rh-eclipse46 Software Collection，可以从面板中选择 **Applications** → **Programming** → **Red Hat Eclipse**，或者在 shell 提示符后输入以下内容：

```
scl enable rh-eclipse46 eclipse
```

在启动过程中，Eclipse 会提示您选择一个 **工作区**，即您要在其中存储项目的目录。您可以使用 `~/workspace/`，这是默认选项，或者点击 **Browse** 按钮浏览您的文件系统并选择自定义目录。另外，您可以选择 **Use this as the default**，且不要再次询问复选框，以防止 Eclipse 在下次运行此开发环境时显示此对话框。完成后，单击 **OK** 按钮以确认选择并继续启动。

#### 4.2.2.1. 使用 Red Hat Developer Toolset Toolchain

要使用 rh-eclipse46 Software Collection 支持 Red Hat Developer Toolset 中的 GNU Compiler Collection 和 binutils，请确保安装了 devtoolset-7-toolchain 软件包并运行应用程序，如 [第 4.2.2 节 “使用 Eclipse”](#) 所述。rh-eclipse46 Collection 默认使用 Red Hat Developer Toolset 工具链。

有关如何在您的系统上安装 devtoolset-7-toolchain 软件包的详细信息，请参阅 [Red Hat Developer Toolset 用户指南](#)。



#### 重要

如果您正在处理之前使用 GNU Compiler Collection 的 Red Hat Enterprise Linux 版本构建的项目，请确保丢弃所有以前的构建结果。为此，请在 Eclipse 中打开项目，然后从菜单中选择 **Project** → **Clean**。

#### 4.2.2.2. 使用 Red Hat Enterprise Linux Toolchain

要使用 `rh-eclipse46 Software Collection`，其支持随 Red Hat Enterprise Linux 一起分发的工具链，请将项目的配置更改为使用 `gcc`、`g++` 和 的绝对路径。

要将 Eclipse 配置为明确为当前项目使用工具的 Red Hat Enterprise Linux 系统版本，请完成以下步骤：

1. 在 C/C++ 透视图，从主菜单栏中选择 **Project** → **Properties** 以打开项目属性。

2. 在对话框左侧的菜单中，单击 **C/C++ Build** → **Settings**。

3. 选择 **Tool Settings** 选项卡。

4. 如果您在使用 C 项目：

1. 选择 **GCC C Compiler** 或 **Cross GCC Compiler**，并将 **Command** 字段的值更改为：

```
/usr/bin/gcc
```

2. 选择 **GCC C Linker** 或 **Cross GCC Linker**，并将 **Command** 字段的值更改为：

```
/usr/bin/gcc
```

3. 选择 **GCC Assembler** 或 **Cross GCC Assembler**，并将 **Command** 字段的值更改为：

```
/usr/bin/as
```

如果您在使用 C++ 项目：

1. 选择 **GCC C++ Compiler** 或 **Cross G++ Compiler**，并将 **Command** 字段的值更改为：

```
/usr/bin/g++
```

2. 选择 **GCC C Compiler** 或 **Cross GCC Compiler**，并将 **Command** 字段的值更改为：

```
/usr/bin/gcc
```

3. 选择 **GCC C++ Linker** 或 **Cross G++ Linker**，并将 **Command** 字段的值更改为：

```
/usr/bin/g++
```

4. 选择 **GCC Assembler** 或 **Cross GCC Assembler**，并将 **Command** 字段的值更改为：

```
/usr/bin/as
```

5. 单击 **OK** 按钮以保存配置更改。

### 4.2.3. 其它资源

有关 **Eclipse** 及其所有功能的详细描述超出了本书的讨论范围。如需更多信息，请参阅以下列出的资源。

#### 安装的文档

- **Eclipse** 包括一个内置的 **Help** 系统，它为每个集成功能和工具提供了广泛的文档。这大大减少了新开发人员所需的初始时间投资，以便在其使用方面变得流畅。在以下链接的 *Red Hat Enterprise Linux 开发人员指南* 中详细介绍了使用此帮助部分。

#### 另请参阅

- [使用 Eclipse](#) 描述红帽开发人员的 `rh-eclipse47` 组件的使用。
- [Red Hat Developer Toolset 用户指南中的 Red Hat Developer Toolset 章节](#) 概述了 Red Hat Developer Toolset，以及如何在您的系统上安装它。
- [Red Hat Developer Toolset 用户指南中的 GNU Compiler Collection \(GCC\) 章节](#) 提供了有

关如何在命令行中编译使用 C、C++ 和 Fortran 编写的程序的信息。

### 4.3. RUBY ON RAILS 5.0

Red Hat Software Collections 3.1 提供了 `rh-ruby24` Software Collection 和 `rh-ror50` Collection。

要安装 Ruby on Rails 5.0, 以 `root` 用户身份运行以下命令：

```
yum install rh-ror50
```

从 `rh-ror50` Software Collection 安装任何软件包会自动拉取 `rh-ruby24` 和 `rh-nodejs6` 作为依赖项。

`rh-nodejs6` Collection 由资产管道中的某些 gems 用于处理后 web 资源, 如 `sass` 或 `coffee-script` 源文件。此外, `Action Cable` 框架使用 `rh-nodejs6` 来处理 Rails 中的 `WebSocket`。

要运行 `rails s` 命令而无需 `rh-nodejs6`, 请在 `Gemfile` 中禁用 `coffee-rails` 和 `uglifier` gems。

要在没有 Node.js 的情况下运行 Ruby on Rails, 请运行以下命令, 它将自动启用 `rh-ruby24`：

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

要使用所有功能运行 Ruby on Rails, 请同时启用 `rh-nodejs6` Software Collection：

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

`rh-ror50` Software Collection 与 `rh-ruby24` 和 `rh-nodejs6` 组件一起支持。

### 4.4. MONGODB 3.6

`rh-mongodb36` Software Collection 仅适用于 Red Hat Enterprise Linux 7。有关如何在 Red Hat Enterprise Linux 6 中使用 MongoDB 3.4 的说明, 请参阅 [第 4.5 节 “MongoDB 3.4”](#)。

要安装 `rh-mongodb36` 集合，请以 `root` 用户身份输入以下命令：

```
yum install rh-mongodb36
```

要运行 MongoDB shell 工具，请输入以下命令：

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



#### 注意

`rh-mongodb36-mongo-cxx-driver` 软件包已使用 Red Hat Developer Toolset 6 中的 GCC 使用 `-std=gnu++14` 选项构建。对使用 C++11（或更新版本）功能的 MongoDB C++ 驱动程序使用共享库的二进制文件必须使用 Red Hat Developer Toolset 6 或更高版本构建。请参阅 [Red Hat Developer Toolset 6 用户指南中的 C++ 兼容性详情](#)。

要启动 MongoDB 守护进程，请以 `root` 用户身份输入以下命令：

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

要在引导时启动 MongoDB 守护进程，请以 `root` 身份输入这个命令：

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

要启动 MongoDB 分片服务器，请以 `root` 用户身份输入以下命令：

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

要在引导时启动 MongoDB 分片服务器，以 `root` 身份输入这个命令：

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

请注意，除非用户至少启动一个配置服务器并在 `mongos.conf` 文件中指定它，否则 MongoDB 分片服务器无法正常工作。

## 4.5. MONGODB 3.4

要安装 `rh-mongodb34` 集合，请以 `root` 用户身份输入以下命令：

```
yum install rh-mongodb34
```

要运行 MongoDB shell 工具，请输入以下命令：

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



### 注意

`rh-mongodb34-mongo-cxx-driver` 软件包已使用 Red Hat Developer Toolset 6 中的 GCC 使用 `-std=gnu++14` 选项构建。使用 C++11（或更新版本）功能的 MongoDB C++ 驱动程序使用共享库的二进制文件必须使用 Red Hat Developer Toolset 6 构建。请参阅 [Red Hat Developer Toolset 6 用户指南中的 C++ 兼容性详情](#)。

## Red Hat Enterprise Linux 6 上的 MongoDB 3.4

如果您使用 Red Hat Enterprise Linux 6，以下说明适用于您的系统。

要启动 MongoDB 守护进程，请以 `root` 用户身份输入以下命令：

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

要在引导时启动 MongoDB 守护进程，请以 `root` 身份输入这个命令：

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

要启动 MongoDB 分片服务器，以 `root` 身份输入这个命令：

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

要在引导时启动 MongoDB 分片服务器，请以 `root` 用户身份输入以下命令：

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

请注意，除非用户至少启动一个配置服务器并在 `mongos.conf` 文件中指定它，否则 MongoDB 分片服务器无法正常工作。

## MongoDB 3.4 on Red Hat Enterprise Linux 7

使用 Red Hat Enterprise Linux 7 时，可以使用以下命令：

要启动 MongoDB 守护进程，请以 root 用户身份输入以下命令：

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

要在引导时启动 MongoDB 守护进程，请以 root 身份输入这个命令：

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

要启动 MongoDB 分片服务器，请以 root 用户身份输入以下命令：

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

要在引导时启动 MongoDB 分片服务器，以 root 身份输入这个命令：

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

请注意，除非用户至少启动一个配置服务器并在 `mongos.conf` 文件中指定它，否则 MongoDB 分片服务器无法正常工作。

## 4.6. GIT

Git 是一个分布式修订控制系统，具有分散的架构。与使用客户端-服务器模型的集中式版本控制系统不同，Git 确保 Git 存储库的每个工作副本都是带有完整修订历史记录的精副本。这不仅允许您处理项目，无需有权将更改推送到其官方存储库，还可以让您使用任何网络连接。如需更多信息，请参阅 *Red Hat Enterprise Linux 7 开发人员指南* 中的 [Git 章节](#)。

## 4.7. MAVEN

`rh-maven35 Software Collection` 仅适用于 Red Hat Enterprise Linux 7，提供软件项目管理和理解工具。根据项目对象模型(POM)的概念，Maven 可以在中央信息中管理项目的构建、报告和文档。

要安装 `rh-maven35` Collection，请以 `root` 用户身份运行以下命令：

```
yum install rh-maven35
```

要启用这个集合，在 `shell` 提示符下输入以下命令：

```
scl enable rh-maven35 bash
```

全局 `Maven` 设置（如远程存储库或镜像）可以通过编辑 `/opt/rh/rh-maven35/root/etc/maven/settings.xml` 文件进行自定义。

有关使用 `Maven` 的更多信息，请参阅 [Maven 文档](#)。本节描述了插件的使用；要查找有关各个插件的文档，请参阅 [插件的索引](#)。<https://maven.apache.org/guides/getting-started/index.html#how-do-i-use-plugins>

## 4.8. PASSENGER

`rh-passenger40` Software Collection 提供 `Phusion Passenger`，它是一个旨在快速、强大且轻量级的 `Web` 和应用服务器。

`rh-passenger40` Collection 支持 `Ruby` 的多个版本，特别是 `ruby193`、`ruby200` 和 `rh-ruby22` Software Collections，以及使用 `ror40` 或 `rh-ror41` Collections 的 `Ruby on Rails`。在将 `Passenger` 与任何 `Ruby` Software Collections 搭配使用前，请先从 `rh-passenger40` Collection 安装对应的软件包：`rh-passenger-ruby193`、`rh-passenger-ruby200` 或 `rh-passenger-ruby22` 软件包。

`rh-passenger40` Software Collection 也可以与 `httpd24` Software Collection 中的 `Apache httpd` 一起使用。为此，请安装 `rh-passenger40-mod_passenger` 软件包。对于 `Apache httpd` 配置示例，请参考默认配置文件 `/opt/rh/httpd24/root/etc/httpd/conf.d/passenger.conf`，它演示了如何在单个 `Apache httpd` 实例中使用多个 `Ruby` 版本。

此外，`rh-passenger40` Software Collection 可以和 `nginx16` 软件集中的 `nginx 1.6` `web` 服务器一起使用。要将 `nginx 1.6` 与 `rh-passenger40` 搭配使用，您可以在 `web` 应用请求的目录中以 `Standalone` 模式运行 `Passenger`：

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



或者，按照上游 [Passenger](#) 文档中所述编辑 `nginx16` 配置文件。

#### 4.9. 数据库连接器

数据库连接器软件包提供数据库客户端功能，这是本地或远程与数据库服务器的连接所必需的。[表 4.2](#) “语言和数据库之间的互操作性” 使用包含特定数据库服务器的语言运行时列出 **Software Collections**。

表 4.2. 语言和数据库之间的互操作性

语言 (软件集合)	数据库				
	MariaDB	MongoDB	MySQL	PostgreSQL	Redis
rh-nodejs4	✗	✗	✗	✗	✗
rh-nodejs6	✗	✗	✗	✗	✗
rh-nodejs8	✗	✗	✗	✗	✗
rh-perl520	✓	✗	✓	✓	✗
rh-perl524	✓	✗	✓	✓	✗
rh-perl526	✓	✗	✓	✓	✗
rh-php56	✓	✓	✓	✓	✗
rh-php70	✓	✗	✓	✓	✗
rh-php71	✓	✗	✓	✓	✗
python27	✓	✓	✓	✓	✗
rh-python34	✗	✓	✗	✓	✗

语言 (软件集合)	数据库				
	MariaDB	MongoDB	MySQL	PostgreSQL	Redis
rh-python35	✓	✓	✓	✓	✗
rh-python36	✓	✓	✓	✓	✗
rh-ror41	✓	✓	✓	✓	✗
rh-ror42	✓	✓	✓	✓	✗
rh-ror50	✓	✓	✓	✓	✗
rh-ruby25	✓	✓	✓	✓	✗
✓	支持		✗	不支持	

## 第 5 章 MIGRATION (迁移)

本章介绍了迁移到 Red Hat Software Collections 3.1 中包含的组件版本的信息。

### 5.1. 迁移到 MARIADB 10.2

Red Hat Enterprise Linux 6 包含 MySQL 5.1, 作为默认的 MySQL 实现。Red Hat Enterprise Linux 7 包括 MariaDB 5.5 作为默认的 MySQL 实现。MariaDB 是社区开发的 MySQL 的简易替代品。自 Red Hat Software Collections 2.2 开始, MariaDB 10.1 作为一个 Software Collection 提供; Red Hat Software Collections 3.1 提供了 MariaDB 10.2。

rh-mariadb102 Software Collection, 适用于 Red Hat Enterprise Linux 6 和 Red Hat Enterprise Linux 7, 不会与核心系统的 mysql 或 mariadb 软件包冲突, 因此可以将 rh-mariadb102 Software Collection 与 mysql 或 mariadb 软件包一起安装。也可以同时运行这两个版本, 但 my.cnf 文件中的端口号和套接字需要更改, 以防止这些特定资源冲突。另外, 可以在 rh-mariadb101 Collection 仍然安装 rh-mariadb102 Software Collection 时安装 rh-mariadb102 Software Collection, 甚至在运行。

请注意, 如果您使用 MariaDB 5.5 或 MariaDB 10.0, 首先需要升级到 rh-mariadb101 Software Collection, 这在 [Red Hat Software Collections 2.4 发行注记](#) 中进行了描述。

有关 MariaDB 10.2 的更多信息, 请参阅上游文档( [版本 10.2](#) )和有关 [升级](#) 的信息。



#### 注意

rh-mariadb102 Software Collection 支持使用 scli register 命令通过 NFS 挂载, 也不支持使用 scli register 命令进行动态注册。

#### 5.1.1. rh-mariadb101 和 rh-mariadb102 Software Collections 之间的显著差异

MariaDB 10.2 的主要变化在 Red Hat Software Collections 3.0 [发行注记](#) 中进行了描述。

从 MariaDB 10.2 开始, 更改了 SQL\_MODE 变量的行为; 详情请查看 [上游文档](#)。

多个选项更改了默认值, 或者已弃用或删除。详情请查看 [从 MariaDB 10.1 迁移到 MariaDB 10.2 Software Collection](#) 的知识库文章。

**rh-mariadb102 Software Collection** 包括 **rh-mariadb102-syspaths** 软件包，它会安装为二进制文件、脚本、手册页和其他提供系统范围打包程序的软件包。安装 **rh-mariadb102\*-syspaths** 软件包后，用户不需要使用 `scl enable` 命令对 **rh-mariadb102\*** 软件包提供的二进制文件和脚本正常工作。请注意，**\*-syspaths** 软件包与基本 Red Hat Enterprise Linux 系统中的对应软件包冲突。

### 5.1.2. 从 rh-mariadb101 升级到 rh-mariadb102 软件集合



#### 重要

在升级前，备份所有数据，包括任何 MariaDB 数据库。

1.

如果 **rh-mariadb101** 数据库服务器仍在运行，则停止它。

在停止服务器前，将 `innodb_fast_shutdown` 选项设置为 `0`，以便 InnoDB 执行缓慢关闭，包括完整清除和插入缓冲区合并。请参阅[上游文档](#)中的有关此选项的更多信息。如果正常关闭时，此操作可能需要更长的时间。

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

停止 **rh-mariadb101** 服务器。

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

2.

安装 **rh-mariadb102 Software Collection**。

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

请注意，在 **rh-mariadb101 Software Collection** 仍然被安装时，可以安装 **rh-mariadb102 Software Collection**，因为这些集合不会冲突。

3.

检查 **rh-mariadb102** 的配置，它存储在 `/etc/opt/rh/rh-mariadb102/my.cnf` 文件和 `/etc/opt/rh/rh-mariadb102/my.cnf.d/` 目录中。将其与存储在 `/etc/opt/rh/rh-mariadb101/my.cnf` 和 `/etc/opt/rh/rh-mariadb101/my.cnf.d/` 中的 **rh-mariadb101** 配置进行比较，并根据需要进行调整。

4.

**rh-mariadb101 Software Collection** 的所有数据存储在 `/var/opt/rh/rh-`

`mariadb101/lib/mysql/` 目录中，除非进行了不同。将该目录的整个内容复制到 `/var/opt/rh/rh-mariadb102/lib/mysql/`。在继续升级前，您可以移动内容但记住备份数据。确保数据归 `mysql` 用户和 SELinux 上下文所有。

5. 启动 `rh-mariadb102` 数据库服务器。

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

6. 执行数据迁移。

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

如果 `root` 用户定义了非空密码（应该定义了密码），则需要使用 `-p` 选项调用 `mysql_upgrade` 工具并指定密码。

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

## 5.2. 迁移到 MONGODB 3.6

Red Hat Software Collections 3.1 带有 MongoDB 3.6，由 `rh-mongodb36` Software Collection 提供，并仅适用于 Red Hat Enterprise Linux 7。

`rh-mongodb36` Software Collection 包括 `rh-mongodb36-syspaths` 软件包，该软件包安装为二进制文件、脚本、手册页和其他提供系统范围打包程序的软件包。安装 `rh-mongodb36*-syspaths` 软件包后，用户不需要使用 `scl enable` 命令对 `rh-mongodb36*` 软件包提供的二进制文件和脚本正常工作。要了解有关 `syspaths` 的更多信息，请参阅 [Red Hat Software Collections Packaging Guide](#)。

### 5.2.1. MongoDB 3.4 和 MongoDB 3.6 之间的显著差异

#### 常规更改

`rh-mongodb36` Software Collection 包括以下显著的常规更改：

- 在 Non-Uniform Access Memory (NUMA) 硬件上，可以将 `systemd` 服务配置为使用 `numactl` 命令启动；请参阅 [上游建议](#)。要将 MongoDB 与 `numactl` 命令搭配使用，您需要安装 `numactl` RPM 软件包，并相应地更改 `/etc/opt/rh/rh-mongodb36/sysconfig/mongod` 和 `/etc/opt/rh/rh-mongodb36/sysconfig/mongos` 配置文件。

## 兼容性更改

MongoDB 3.6 包括各种小更改，可能会影响与以前版本的 MongoDB 的兼容性：

- **MongoDB 二进制文件现在默认绑定到 localhost，因此需要明确启用侦听不同的 IP 地址。请注意，这已经是 MongoDB Software Collections 发布的 systemd 服务的默认行为。**
- **MONGODB-CR 身份验证机制已被弃用。对于由 3.0 之前的 MongoDB 版本创建的用户的数据库，请将身份验证模式升级到 [SCRAM](#)。**
- **HTTP 接口和 REST API 已被删除**
- **副本集中的仲裁程序具有优先级 0**
- **master-slave 复制已弃用**

有关 MongoDB 3.6 的详细兼容性更改，请参阅 [上游发行注记](#)。

## 后向兼容的功能

以下 MongoDB 3.6 功能向后兼容，需要使用 `featureCompatibilityVersion` 命令将版本设置为 3.6：

- **集合的 UUID**
- **`$jsonschema` 文档验证**
- **更改流**
- **chunk aware secondaries**
- **查看使用 3.6 查询功能的定义、文档验证器和部分索引过滤器**

- 会话和可重试写入
- 带有 authenticationRestrictions 的用户和角色

有关 MongoDB 3.6 中向后兼容更改的详情，请查看 [上游发行注记](#)。

### 5.2.2. 从 rh-mongodb34 升级到 rh-mongodb36 软件集合



#### 重要

在从 rh-mongodb34 迁移到 rh-mongodb36 Software Collection 之前，请备份所有数据，包括任何 MongoDB 数据库，这些数据库默认存储在 `/var/opt/rh/rh-mongodb34/lib/mongodb/` 目录中。另外，请参阅 [兼容性更改](#) 以确保应用程序和部署与 MongoDB 3.6 兼容。

要升级到 rh-mongodb36 Software Collection，请执行以下步骤。

1. 要进行升级，rh-mongodb34 实例必须将 `featureCompatibilityVersion` 设置为 3.4。检查 `featureCompatibilityVersion`：

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

如果 mongod 服务器配置有启用的访问控制，请在 mongo 命令中添加 `--username` 和 `--password` 选项。

2. 从 rh-mongodb36 Software Collections 安装 MongoDB 服务器和 shell：

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

3. 停止 MongoDB 3.4 服务器：

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

4.

将数据复制到新位置：

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

5.

在 `/etc/opt/rh/rh-mongodb36/mongod.conf` 文件中配置 `rh-mongodb36-mongod` 守护进程。

6.

启动 MongoDB 3.6 服务器：

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

7.

启用向后不兼容的功能：

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

如果 `mongod` 服务器配置有启用的访问控制，请在 `mongo` 命令中添加 `--username` 和 `--password` 选项。



#### 注意

升级后，建议先运行部署，而不为刻录期间启用向后兼容的功能，从而最大程度降低降级的可能性。

有关升级的详情，请查看 [上游发行注记](#)。

有关升级 Replica Set 的详情，请参考上游 [MongoDB Manual](#)。

有关升级分片集群的详情，请参考上游 [MongoDB Manual](#)。

### 5.3. 迁移到 MONGODB 3.4

`rh-mongodb34` Software Collection 适用于 Red Hat Enterprise Linux 6 和 Red Hat Enterprise



Linux 7, 提供 MongoDB 3.4。

### 5.3.1. MongoDB 3.2 和 MongoDB 3.4 之间的显著差异

#### 常规更改

`rh-mongodb34` Software Collection 引入了各种常规更改。从 [MongoDB 3.2 迁移到 MongoDB 3.4](#) 的知识库文章中列出了主要变化。有关详细更改, 请参阅 [上游发行注记](#)。

另外, 此 Software Collection 包括 `rh-mongodb34-syspaths` 软件包, 该软件包会安装为二进制文件、脚本、手册页和其他提供系统范围打包程序的软件包。安装 `rh-mongodb34*-syspaths` 软件包后, 用户不需要使用 `scl enable` 命令对 `rh-mongodb34*` 软件包提供的二进制文件和脚本正常工作。要了解有关 `syspaths` 的更多信息, 请参阅 [Red Hat Software Collections Packaging Guide](#)。

#### 兼容性更改

MongoDB 3.4 包括各种小更改, 可能会影响与以前版本的 MongoDB 的兼容性。详情请查看从 [MongoDB 3.2 迁移到 MongoDB 3.4 和上游文档](#) 的知识库文章。 <https://docs.mongodb.com/manual/release-notes/3.4-compatibility/>

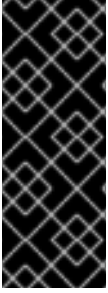
值得注意的是, 以下 MongoDB 3.4 功能会向后兼容, 需要使用 `featureCompatibilityVersion` 命令将版本设置为 3.4 :

- 支持从现有集合或其他视图创建只读视图
- 索引版本 v : 2, 它添加了对协调、十进制数据和区分大小写的索引的支持
- 支持带有新 `十进制数据类型` 的十进制 128 格式

有关 MongoDB 3.4 中向后兼容更改的详情, 请查看 [上游发行注记](#)。

### 5.3.2. 从 rh-mongodb32 升级到 rh-mongodb34 Software Collection

请注意, 在升级到 MongoDB 3.4 并使用新功能启动后, 无法降级到 3.2.7 或更早版本。您只能降级到 3.2.8 或更高版本。



## 重要

在从 rh-mongodb32 迁移到 rh-mongodb34 Software Collection 之前，请备份所有数据，包括任何 MongoDB 数据库，这些数据库默认存储在 /var/opt/rh/rh-mongodb32/lib/mongodb/ 目录中。另外，请参阅兼容性更改，以确保应用程序和部署与 MongoDB 3.4 兼容。

要升级到 rh-mongodb34 Software Collection，请执行以下步骤：

1. 从 rh-mongodb34 Software Collections 安装 MongoDB 服务器和 shell：

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

2. 停止 MongoDB 3.2 服务器：

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

在 Red Hat Enterprise Linux 6 系统上使用服务 rh-mongodb32-mongod stop 命令。

3. 将数据复制到新位置：

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

4. 在 /etc/opt/rh/rh-mongodb34/mongod.conf 文件中配置 rh-mongodb34-mongod 守护进程。

5. 启动 MongoDB 3.4 服务器：

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

在 Red Hat Enterprise Linux 6 上，改为使用服务 rh-mongodb34-mongod start 命令。

6. 启用向后兼容的功能：

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

如果 mongod 服务器配置有启用的访问控制，请将 `--username` 和 `--password` 选项添加到 `mongo` 命令。

请注意，建议您在升级后运行部署，而无需首先启用这些功能。

有关升级的详情，请查看 [上游发行注记](#)。

有关升级 Replica Set 的详情，请参考上游 [MongoDB Manual](#)。

有关升级分片集群的详情，请参考上游 [MongoDB Manual](#)。

## 5.4. 迁移到 MYSQL 5.7

Red Hat Enterprise Linux 6 包含 MySQL 5.1 作为默认的 MySQL 实现。Red Hat Enterprise Linux 7 包括 MariaDB 5.5 作为默认的 MySQL 实现。除了这些基本版本外，自 Red Hat Software Collections 2.0 开始，MySQL 5.6 也可作为 Red Hat Enterprise Linux 6 和 Red Hat Enterprise Linux 7 的软件集合提供。

`rh-mysql 57` Software Collection（适用于 Red Hat Enterprise Linux 6 和 Red Hat Enterprise Linux 7）都与核心系统的 `mysql` 或 `mariadb` 软件包冲突，因此可以和 `rh-mysql56` 软件集合一起安装 `rh-mysql57` 软件集合。也可以同时运行多个版本；但是，需要更改 `my.cnf` 文件中的端口号和套接字，以防止这些特定资源冲突。

请注意，只能从 MySQL 5.6 升级到 MySQL 5.7。如果您需要从早期版本升级，请首先升级到 MySQL 5.6。[Red Hat Software Collections 2.2 发行注记中](#) 提供了有关如何升级到 MySQL 5.6 的说明。

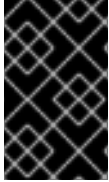
### 5.4.1. MySQL 5.6 和 MySQL 5.7 之间的显著差异

- `mysql-bench` 子软件包不包含在 `rh-mysql57` 软件集合中。
- 自 MySQL 5.7.7 起，默认的 SQL 模式包含 `NO_AUTO_CREATE_USER`。因此，需要使用 `CREATE USER` 语句创建 MySQL 帐户，因为 `GRANT` 语句不再默认创建用户。详情请查看 [上](#)

[游文档](#)。

要了解有关 MySQL 5.7 中与早期版本相比的详细更改，请参阅上游文档：[MySQL 5.7 中的新内容](#)，并将 [升级更改](#) 为 [MySQL 5.7](#)。

#### 5.4.2. 升级到 rh-mysql57 Software Collection



##### 重要

在升级前，备份您的所有数据，包括任何 MySQL 数据库。

1. 安装 rh-mysql57 Software Collection。

```
yum install rh-mysql57-mysql-server
```

2. 检查 rh-mysql57 的配置，它存储在 `/etc/opt/rh/rh-mysql57/my.cnf` 文件中，以及 `/etc/opt/rh/rh-mysql57/my.cnf.d/` 目录中。将它与存储在 `/etc/opt/rh/rh-mysql56/my.cnf` 和 `/etc/opt/rh/rh-mysql56/my.cnf.d/` 中的 `rh-mysql56/my.cnf.d/` 的配置进行比较，并根据需要进行调整。

3. 如果仍然在运行，则停止 rh-mysql56 数据库服务器。

```
service rh-mysql56-mysqld stop
```

4. rh-mysql56 Software Collection 的所有数据存储在 `/var/opt/rh/rh-mysql56/lib/mysql/` 目录中。将该目录的整个内容复制到 `/var/opt/rh/rh-mysql57/lib/mysql/`。您还可以移动内容，但记得在继续升级前备份数据。

5. 启动 rh-mysql57 数据库服务器。

```
service rh-mysql57-mysqld start
```

6. 执行数据迁移。

```
scl enable rh-mysql57 mysql_upgrade
```

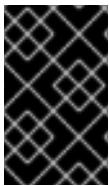
如果 root 用户定义了非空密码（应该定义了密码），则需要使用 `-p` 选项调用 `mysql_upgrade` 工具并指定密码。

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

## 5.5. 迁移到 PostgreSQL 10

Red Hat Software Collections 3.1 带有 PostgreSQL 10，仅适用于 Red Hat Enterprise Linux 7。rh-postgresql10 Software Collection 可以与 PostgreSQL 基本或任何 PostgreSQL Software Collection 并行安装在同一台机器上。也可以同时在一台机器上运行多个 PostgreSQL 版本，但您需要使用不同的端口或 IP 地址并调整 SELinux 策略。有关如何迁移到更早的版本或使用 Red Hat Enterprise Linux 6 的说明，请查看 [第 5.6 节“迁移到 PostgreSQL 9.6”](#)。

rh-postgresql10 Software Collection 包括 rh-postgresql10-syspaths 软件包，该软件包会安装为二进制文件、脚本、手册页和其他提供系统范围打包程序的软件包。安装 rh-postgresql10\*-syspaths 软件包后，用户不需要使用 `scl enable` 命令对 rh-postgresql10\* 软件包提供的二进制文件和脚本正常工作。请注意，\*-syspaths 软件包与基本 Red Hat Enterprise Linux 系统中的对应软件包冲突。要了解有关 syspaths 的更多信息，请参阅 [Red Hat Software Collections Packaging Guide](#)。



### 重要

在迁移到 PostgreSQL 10 之前，[请查看上游兼容性备注](#)。

下表提供了 postgresql 软件包提供的 PostgreSQL 的 Red Hat Enterprise Linux 7 系统版本以及 rh-postgresql 96 和 rh-postgresql10 Software Collections 中的不同路径的概述。

表 5.1. PostgreSQL 路径中的 Differences

内容	postgresql	rh-postgresql96	rh-postgresql10
可执行文件	/usr/bin/	/opt/rh/rh-postgresql96/root/usr/bin/ /	/opt/rh/rh-postgresql10/root/usr/bin/ /
库	/usr/lib64/	/opt/rh/rh-postgresql96/root/usr/lib64/	/opt/rh/rh-postgresql10/root/usr/lib64/
Documentation	/usr/share/doc/postgresql/html/	/opt/rh/rh-postgresql96/root/usr/share/doc/postgresql/html/	/opt/rh/rh-postgresql10/root/usr/share/doc/postgresql/html/

内容	postgresql	rh-postgresql96	rh-postgresql10
PDF 文档	/usr/share/doc/postgresql-docs/	/opt/rh/rh-postgresql96/root/usr/share/doc/postgresql-docs/	/opt/rh/rh-postgresql10/root/usr/share/doc/postgresql-docs/
contrib 文档	/usr/share/doc/postgresql-contrib/	/opt/rh/rh-postgresql96/root/usr/share/doc/postgresql-contrib/	/opt/rh/rh-postgresql10/root/usr/share/doc/postgresql-contrib/
Source	未安装	未安装	未安装
data	/var/lib/pgsql/data/	/var/opt/rh/rh-postgresql96/lib/pgsql/data/	/var/opt/rh/rh-postgresql10/lib/pgsql/data/
备份区域	/var/lib/pgsql/backups/	/var/opt/rh/rh-postgresql96/lib/pgsql/backups/	/var/opt/rh/rh-postgresql10/lib/pgsql/backups/
模板	/usr/share/pgsql/	/opt/rh/rh-postgresql96/root/usr/share/pgsql/	/opt/rh/rh-postgresql10/root/usr/share/pgsql/
流程语言	/usr/lib64/pgsql/	/opt/rh/rh-postgresql96/root/usr/lib64/pgsql/	/opt/rh/rh-postgresql10/root/usr/lib64/pgsql/
开发标头	/usr/include/pgsql/	/opt/rh/rh-postgresql96/root/usr/include/pgsql/	/opt/rh/rh-postgresql10/root/usr/include/pgsql/
其他共享数据	/usr/share/pgsql/	/opt/rh/rh-postgresql96/root/usr/share/pgsql/	/opt/rh/rh-postgresql10/root/usr/share/pgsql/
回归测试	/usr/lib64/pgsql/test/regress/ (在 -test 软件包中)	/opt/rh/rh-postgresql96/root/usr/lib64/pgsql/test/regress/ (在 -test 软件包中)	/opt/rh/rh-postgresql10/root/usr/lib64/pgsql/test/regress/ (在 -test 软件包中)

### 5.5.1. 从 Red Hat Enterprise Linux 系统版本 PostgreSQL 迁移到 PostgreSQL 10 Software Collection

Red Hat Enterprise Linux 7 带有 PostgreSQL 9.2。要将数据从 PostgreSQL 的 Red Hat Enterprise Linux 系统版本迁移到 rh-postgresql10 Software Collection，您可以使用 `pg_upgrade` 工具（推荐）执行快速升级，或使用 `SQL` 命令将数据库数据转储到文本文件中，并在新数据库中导入它。

请注意，第二个方法通常慢得多，可能需要手动修复；有关此升级方法的更多信息，请参阅 [PostgreSQL 文档](#)。



### 重要

在将数据从 PostgreSQL 的 Red Hat Enterprise Linux 系统版本迁移到 PostgreSQL 10 之前，请确保备份所有数据，包括 PostgreSQL 数据库文件，这些文件默认位于 `/var/lib/pgsql/data/` 目录中。

#### 过程 5.1. 使用 `pg_upgrade` 工具快速升级

要快速升级 PostgreSQL 服务器，请完成以下步骤：

1. 停止旧的 PostgreSQL 服务器，以确保数据不处于不一致的状态。要做到这一点，以 `root` 用户身份在 `shell` 提示符后输入以下内容：

```
systemctl stop postgresql.service
```

要验证服务器是否没有运行，请输入：

```
systemctl status postgresql.service
```

2. 验证旧目录 `/var/lib/pgsql/data/` 是否存在：

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

并备份您的数据。

3. 验证新数据目录 `/var/opt/rh/rh-postgresql10/lib/pgsql/data/` 不存在：

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

如果您正在运行一个全新的 PostgreSQL 10 安装，则系统中不存在此目录。如果是，请以 `root` 用户身份运行以下命令备份它：

```
mv /var/opt/rh/rh-postgresql10/lib/pgsql/data{-, -scl-backup}
```

4.

以 `root` 用户身份运行以下命令来升级新服务器的数据库数据：

```
scl enable rh-postgresql10 -- postgresql-setup --upgrade
```

或者，您可以使用

```
/opt/rh/rh-postgresql10/root/usr/bin/postgresql-setup --upgrade
```

 命令。

请注意，您可以使用 `--upgrade-from` 选项从不同的 PostgreSQL 版本升级。可以使用 `--upgrade-ids` 选项进行可能的升级场景列表。

建议您读取生成的 `/var/lib/pgsql/upgrade_rh-postgresql10-postgresql.log` 日志文件，以了解升级过程中是否存在任何问题。

5.

以 `root` 用户身份启动新服务器：

```
systemctl start rh-postgresql10-postgresql.service
```

另外，建议您运行 `analyze_new_cluster.sh` 脚本，如下所示：

```
su - postgres -c 'scl enable rh-postgresql10 ~/analyze_new_cluster.sh'
```

6.

另外，您可以将 PostgreSQL 10 服务器配置为在引导时自动启动。要禁用旧的系统 PostgreSQL 服务器，以 `root` 用户身份运行以下命令：

```
chkconfig postgresql off
```

要启用 PostgreSQL 10 服务器，以 `root` 用户身份键入：

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

7.

如果您的配置与默认文件不同，请确保更新配置文件，特别是 `/var/opt/rh/rh-postgresql10/lib/pgsql/data/pg_hba.conf` 配置文件。否则，只有 `postgres` 用户才能访问数据库。



要执行 PostgreSQL 服务器的转储和恢复升级，请完成以下步骤：

1. 以 root 用户身份在 shell 提示符后输入以下内容来确保旧的 PostgreSQL 服务器正在运行：

```
systemctl start postgresql.service
```

2. 将 PostgreSQL 数据库中的所有数据转储到脚本文件中。以 root 用户身份，输入：

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

3. 以 root 用户身份运行以下命令来停止旧服务器：

```
systemctl stop postgresql.service
```

4. 以 root 用户身份初始化新服务器的数据目录：

```
scl enable rh-postgresql10-postgresql -- postgresql-setup --initdb
```

5. 以 root 用户身份启动新服务器：

```
systemctl start rh-postgresql10-postgresql.service
```

6. 从之前创建的 SQL 文件导入数据：

```
su - postgres -c 'scl enable rh-postgresql10 "psql -f ~/pgdump_file.sql postgres"'
```

7. 另外，您可以将 PostgreSQL 10 服务器配置为在引导时自动启动。要禁用旧的系统 PostgreSQL 服务器，以 root 用户身份运行以下命令：

```
chkconfig postgresql off
```

要启用 PostgreSQL 10 服务器，以 root 用户身份键入：

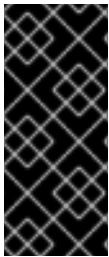
```
chkconfig rh-postgresql10-postgresql on
```

8.

如果您的配置与默认文件不同，请确保更新配置文件，特别是 `/var/opt/rh/rh-postgresql10/lib/pgsql/data/pg_hba.conf` 配置文件。否则，只有 `postgres` 用户才能访问数据库。

### 5.5.2. 从 PostgreSQL 9.6 Software Collection 迁移到 PostgreSQL 10 Software Collection

要将数据从 `rh-postgresql96 Software Collection` 迁移到 `rh-postgresql10 Collection`，您可以使用 `pg_upgrade` 工具（推荐）执行快速升级，或使用 `SQL` 命令将数据库数据转储到文本文件中，并在新数据库中导入它。请注意，第二个方法通常慢得多，可能需要手动修复；有关此升级方法的更多信息，请参阅 [PostgreSQL 文档](#)。



#### 重要

在将数据从 PostgreSQL 9.6 迁移到 PostgreSQL 10 之前，请确保备份所有数据，包括 PostgreSQL 数据库文件，这些文件默认位于 `/var/opt/rh/rh-postgresql96/lib/pgsql/data/` 目录中。

#### 过程 5.3. 使用 `pg_upgrade` 工具快速升级

要快速升级 PostgreSQL 服务器，请完成以下步骤：

1.

停止旧的 PostgreSQL 服务器，以确保数据不处于不一致的状态。要做到这一点，以 `root` 用户身份在 `shell` 提示符后输入以下内容：

```
systemctl stop rh-postgresql96-postgresql.service
```

要验证服务器是否没有运行，请输入：

```
systemctl status rh-postgresql96-postgresql.service
```

2.

验证旧目录 `/var/opt/rh/rh-postgresql96/lib/pgsql/data/` 是否存在：

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

并备份您的数据。

3.

验证新数据目录 `/var/opt/rh/rh-postgresql10/lib/pgsql/data/` 不存在：

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

如果您正在运行一个全新的 PostgreSQL 10 安装，则系统中不存在此目录。如果是，请以 `root` 用户身份运行以下命令备份它：

```
mv /var/opt/rh/rh-postgresql10/lib/pgsql/data{,-scl-backup}
```

4.

以 `root` 用户身份运行以下命令来升级新服务器的数据库数据：

```
scl enable rh-postgresql10 -- postgresql-setup --upgrade --upgrade-from=rh-postgresql96-postgresql
```

或者，您可以使用

`/opt/rh/rh-postgresql10/root/usr/bin/postgresql-setup --upgrade --upgrade-from=rh-postgresql96-postgresql` 命令。

请注意，您可以使用 `--upgrade-from` 选项来从不同版本的 PostgreSQL 升级。可以使用 `--upgrade-ids` 选项进行可能的升级场景列表。

建议您读取生成的 `/var/lib/pgsql/upgrade_rh-postgresql10-postgresql.log` 日志文件，以了解升级过程中是否存在任何问题。

5.

以 `root` 用户身份启动新服务器：

```
systemctl start rh-postgresql10-postgresql.service
```

另外，建议您运行 `analyze_new_cluster.sh` 脚本，如下所示：

```
su - postgres -c 'scl enable rh-postgresql10 ~/analyze_new_cluster.sh'
```

6.

另外，您可以将 PostgreSQL 10 服务器配置为在引导时自动启动。要禁用旧的 PostgreSQL 9.6 服务器，请以 `root` 用户身份输入以下命令：

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

要启用 PostgreSQL 10 服务器，以 root 用户身份键入：

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

7.

如果您的配置与默认文件不同，请确保更新配置文件，特别是 `/var/opt/rh/rh-postgresql10/lib/pgsql/data/pg_hba.conf` 配置文件。否则，只有 postgres 用户才能访问数据库。

#### 过程 5.4. 执行转储和恢复升级

要执行 PostgreSQL 服务器的转储和恢复升级，请完成以下步骤：

1.

以 root 用户身份在 shell 提示符后输入以下内容来确保旧的 PostgreSQL 服务器正在运行：

```
systemctl start rh-postgresql96-postgresql.service
```

2.

将 PostgreSQL 数据库中的所有数据转储到脚本文件中。以 root 用户身份，输入：

```
su - postgres -c 'scl enable rh-postgresql96 "pg_dumpall > ~/pgdump_file.sql"'
```

3.

以 root 用户身份运行以下命令来停止旧服务器：

```
systemctl stop rh-postgresql96-postgresql.service
```

4.

以 root 用户身份初始化新服务器的数据目录：

```
scl enable rh-postgresql10-postgresql -- postgresql-setup --initdb
```

5.

以 root 用户身份启动新服务器：

```
systemctl start rh-postgresql10-postgresql.service
```

6.

从之前创建的 SQL 文件导入数据：

```
su - postgres -c 'scl enable rh-postgresql10 "psql -f ~/pgdump_file.sql postgres"'
```

7.

另外，您可以将 PostgreSQL 10 服务器配置为在引导时自动启动。要禁用旧的 PostgreSQL 9.6 服务器，请以 root 用户身份输入以下命令：

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

要启用 PostgreSQL 10 服务器，以 root 用户身份键入：

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

8.

如果您的配置与默认文件不同，请确保更新配置文件，特别是 `/var/opt/rh/rh-postgresql10/lib/pgsql/data/pg_hba.conf` 配置文件。否则，只有 postgres 用户才能访问数据库。

## 5.6. 迁移到 POSTGRESQL 9.6

PostgreSQL 9.6 可用于 Red Hat Enterprise Linux 6 和 Red Hat Enterprise Linux 7，它可以与 Red Hat Enterprise Linux 6 中的 PostgreSQL 8.4 并行安装同一机器上，或者 Red Hat Enterprise Linux 7 中的 PostgreSQL 9.2 或之前版本的 Red Hat Software Collections 中发布的任何 PostgreSQL 版本。也可以同时在一台机器上运行多个 PostgreSQL 版本，但您需要使用不同的端口或 IP 地址并调整 SELinux 策略。

### 5.6.1. PostgreSQL 9.5 和 PostgreSQL 9.6 之间的显著差异

[上游发行注记](#) 中描述了 PostgreSQL 9.5 和 PostgreSQL 9.6 之间的显著变化。

rh-postgresql96 Software Collection 包括 rh-postgresql96-syspaths 软件包，该软件包会安装为二进制文件、脚本、手册页和其他提供系统范围打包程序的软件包。安装 rh-postgresql96\*-syspaths 软件包后，用户不需要使用 `scl enable` 命令对 rh-postgresql96\* 软件包提供的二进制文件和脚本正常工作。请注意，\*-syspaths 软件包与基本 Red Hat Enterprise Linux 系统中的对应软件包冲突。要了解有关 syspaths 的更多信息，请参阅 [Red Hat Software Collections Packaging Guide](#)。

下表提供了 Red Hat Enterprise Linux 系统版本 PostgreSQL (postgresql) 和 postgresql92、rh-postgresql95 和 rh-postgresql96 Software Collections 中的不同路径的概述。请注意，Red Hat Enterprise Linux 6 发布的 PostgreSQL 8.4 的路径以及 Red Hat Enterprise Linux 7 提供的 PostgreSQL 9.2 系统版本相同；rh-postgresql94 Software Collection 的路径与 rh-postgresql95 类似。

表 5.2. PostgreSQL 路径中的 Differences

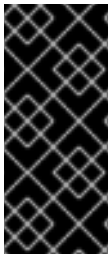
内容	postgresql	postgresql92	rh-postgresql95	rh-postgresql96
可执行文件	/usr/bin/	/opt/rh/postgresql92 /root/usr/bin/	/opt/rh/rh- postgresql95/root/us r/bin/	/opt/rh/rh- postgresql96/root/us r/bin/
库	/usr/lib64/	/opt/rh/postgresql92 /root/usr/lib64/	/opt/rh/rh- postgresql95/root/us r/lib64/	/opt/rh/rh- postgresql96/root/us r/lib64/
Documenta tion	/usr/share/doc /postgresql/ht ml/	/opt/rh/postgresql92 /root/usr/share/doc/ postgresql/html/	/opt/rh/rh- postgresql95/root/us r/share/doc/postgres ql/html/	/opt/rh/rh- postgresql96/root/us r/share/doc/postgres ql/html/
PDF 文档	/usr/share/doc /postgresql- docs/	/opt/rh/postgresql92 /root/usr/share/doc/ postgresql-docs/	/opt/rh/rh- postgresql95/root/us r/share/doc/postgres ql-docs/	/opt/rh/rh- postgresql96/root/us r/share/doc/postgres ql-docs/
contrib 文档	/usr/share/doc /postgresql- contrib/	/opt/rh/postgresql92 /root/usr/share/doc/ postgresql-contrib/	/opt/rh/rh- postgresql95/root/us r/share/doc/postgres ql-contrib/	/opt/rh/rh- postgresql96/root/us r/share/doc/postgres ql-contrib/
Source	未安装	未安装	未安装	未安装
data	/var/lib/pgsql/ data/	/opt/rh/postgresql92 /root/var/lib/pgsql/d ata/	/var/opt/rh/rh- postgresql95/lib/pgs ql/data/	/var/opt/rh/rh- postgresql96/lib/pgs ql/data/
备份区域	/var/lib/pgsql/ backups/	/opt/rh/postgresql92 /root/var/lib/pgsql/b ackups/	/var/opt/rh/rh- postgresql95/lib/pgs ql/backups/	/var/opt/rh/rh- postgresql96/lib/pgs ql/backups/
模板	/usr/share/pgs ql/	/opt/rh/postgresql92 /root/usr/share/pgs ql/ /	/opt/rh/rh- postgresql95/root/us r/share/pgs ql/	/opt/rh/rh- postgresql96/root/us r/share/pgs ql/
流程语言	/usr/lib64/pgs ql/	/opt/rh/postgresql92 /root/usr/lib64/pgs ql/ /	/opt/rh/rh- postgresql95/root/us r/lib64/pgs ql/	/opt/rh/rh- postgresql96/root/us r/lib64/pgs ql/
开发标头	/usr/include/p gs ql/	/opt/rh/postgresql92 /root/usr/include/pgs ql/	/opt/rh/rh- postgresql95/root/us r/include/pgs ql/	/opt/rh/rh- postgresql96/root/us r/include/pgs ql/
其他共享数 据	/usr/share/pgs ql/	/opt/rh/postgresql92 /root/usr/share/pgs ql/ /	/opt/rh/rh- postgresql95/root/us r/share/pgs ql/	/opt/rh/rh- postgresql96/root/us r/share/pgs ql/

内容	postgresql	postgresql92	rh-postgresql95	rh-postgresql96
回归测试	/usr/lib64/pgs ql/test/regress / (在 -test 软 件包中)	/opt/rh/postgresql92 /root/usr/lib64/pgs ql/test/regress/ (在 - test 软件包中)	/opt/rh/rh- postgresql95/root/us r/lib64/pgs ql/test/re gress/ (在 -test 软件 包中)	/opt/rh/rh- postgresql96/root/us r/lib64/pgs ql/test/re gress/ (在 -test 软件 包中)

有关 PostgreSQL 8.4 和 PostgreSQL 9.2 之间的更改，请参阅 [Red Hat Software Collections 1.2 发行注记](#)。Red Hat Software Collections 2.0 发行注记中描述了 PostgreSQL 9.2 和 PostgreSQL 9.4 之间的显著变化。有关 PostgreSQL 9.4 和 PostgreSQL 9.5 之间的区别，请参阅 [Red Hat Software Collections 2.2 发行注记](#)。

### 5.6.2. 从 Red Hat Enterprise Linux 系统版本 PostgreSQL 迁移到 PostgreSQL 9.6 软件集合

Red Hat Enterprise Linux 6 包括 PostgreSQL 8.4，Red Hat Enterprise Linux 7 带有 PostgreSQL 9.2。要将数据从 PostgreSQL 的 Red Hat Enterprise Linux 系统版本迁移到 rh-postgresql96 Software Collection，您可以使用 `pg_upgrade` 工具（推荐）执行快速升级，或使用 SQL 命令将数据库数据转储到文本文件中，并在新数据库中导入它。请注意，第二个方法通常慢得多，可能需要手动修复；有关此升级方法的更多信息，请参阅 [PostgreSQL 文档](#)。以下流程适用于 Red Hat Enterprise Linux 6 和 Red Hat Enterprise Linux 7 系统版本。



#### 重要

在将数据从 PostgreSQL 的 Red Hat Enterprise Linux 系统版本迁移到 PostgreSQL 9.6 之前，请确保备份所有数据，包括 PostgreSQL 数据库文件，这些文件默认位于 `/var/lib/pgs  
ql/data/` 目录中。

### 过程 5.5. 使用 `pg_upgrade` 工具快速升级

要快速升级 PostgreSQL 服务器，请完成以下步骤：

1.

停止旧的 PostgreSQL 服务器，以确保数据不处于不一致的状态。要做到这一点，以 root 用户身份在 shell 提示符后输入以下内容：

```
service postgresql stop
```

要验证服务器是否没有运行，请输入：

```
service postgresql status
```

2. 验证旧目录 `/var/lib/pgsql/data/` 是否存在：

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

并备份您的数据。

3. 验证新数据目录 `/var/opt/rh/rh-postgresql96/lib/pgsql/data/` 不存在：

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

如果您正在运行一个全新的 PostgreSQL 9.6 安装，则系统中不存在此目录。如果是，请以 `root` 用户身份运行以下命令备份它：

```
mv /var/opt/rh/rh-postgresql96/lib/pgsql/data{,-scl-backup}
```

4. 以 `root` 用户身份运行以下命令来升级新服务器的数据库数据：

```
scl enable rh-postgresql96 -- postgresql-setup --upgrade
```

或者，您可以使用

```
/opt/rh/rh-postgresql96/root/usr/bin/postgresql-setup --upgrade 命令。
```

请注意，您可以使用 `--upgrade-from` 选项从不同的 PostgreSQL 版本升级。可以使用 `--upgrade-ids` 选项进行可能的升级场景列表。

建议您读取生成的 `/var/lib/pgsql/upgrade_rh-postgresql96-postgresql.log` 日志文件，以了解升级过程中是否存在任何问题。

5. 以 `root` 用户身份启动新服务器：

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

另外，建议您运行 `analyze_new_cluster.sh` 脚本，如下所示：



```
su - postgres -c 'scl enable rh-postgresql96 ~/analyze_new_cluster.sh'
```

6.

另外，您可以将 PostgreSQL 9.6 服务器配置为在引导时自动启动。要禁用旧的系统 PostgreSQL 服务器，以 root 用户身份运行以下命令：

```
chkconfig postgresql off
```

要启用 PostgreSQL 9.6 服务器，以 root 用户身份输入：

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

7.

如果您的配置与默认文件不同，请确保更新配置文件，特别是 `/var/opt/rh/rh-postgresql96/lib/pgsql/data/pg_hba.conf` 配置文件。否则，只有 postgres 用户才能访问数据库。

### 过程 5.6. 执行转储和恢复升级

要执行 PostgreSQL 服务器的转储和恢复升级，请完成以下步骤：

1.

以 root 用户身份在 shell 提示符后输入以下内容来确保旧的 PostgreSQL 服务器正在运行：

```
service postgresql start
```

2.

将 PostgreSQL 数据库中的所有数据转储到脚本文件中。以 root 用户身份，输入：

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

3.

以 root 用户身份运行以下命令来停止旧服务器：

```
service postgresql stop
```

4.

以 root 用户身份初始化新服务器的数据目录：

```
scl enable rh-postgresql96-postgresql -- postgresql-setup --initdb
```

-

5. 以 root 用户身份启动新服务器：

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

6. 从之前创建的 SQL 文件导入数据：

```
su - postgres -c 'scl enable rh-postgresql96 "psql -f ~/pgdump_file.sql postgres"'
```

7. 另外，您可以将 PostgreSQL 9.6 服务器配置为在引导时自动启动。要禁用旧的系统 PostgreSQL 服务器，以 root 用户身份运行以下命令：

```
chkconfig postgresql off
```

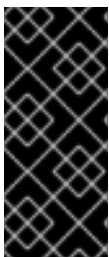
要启用 PostgreSQL 9.6 服务器，以 root 用户身份输入：

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

8. 如果您的配置与默认文件不同，请确保更新配置文件，特别是 `/var/opt/rh/rh-postgresql96/lib/pgsql/data/pg_hba.conf` 配置文件。否则，只有 postgres 用户才能访问数据库。

### 5.6.3. 从 PostgreSQL 9.5 Software Collection 迁移到 PostgreSQL 9.6 Software Collection

要将数据从 rh-postgresql95 Software Collection 迁移到 rh-postgresql96 Collection，您可以使用 `pg_upgrade` 工具（推荐）执行快速升级，或使用 SQL 命令将数据库数据转储到文本文件中，并在新数据库中导入它。请注意，第二个方法通常慢得多，可能需要手动修复；有关此升级方法的更多信息，请参阅 [PostgreSQL 文档](#)。



#### 重要

在将数据从 PostgreSQL 9.5 迁移到 PostgreSQL 9.6 之前，请确保备份所有数据，包括 PostgreSQL 数据库文件，这些文件默认位于 `/var/opt/rh/rh-postgresql95/lib/pgsql/data/` 目录中。

#### 过程 5.7. 使用 `pg_upgrade` 工具快速升级

要快速升级 PostgreSQL 服务器，请完成以下步骤：

1.

停止旧的 PostgreSQL 服务器，以确保数据不处于不一致的状态。要做到这一点，以 root 用户身份在 shell 提示符后输入以下内容：

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

要验证服务器是否没有运行，请输入：

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

2.

验证旧目录 `/var/opt/rh/rh-postgresql95/lib/pgsql/data/` 是否存在：

```
file /var/opt/rh/rh-postgresql95/lib/pgsql/data/
```

并备份您的数据。

3.

验证新数据目录 `/var/opt/rh/rh-postgresql96/lib/pgsql/data/` 不存在：

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

如果您正在运行一个全新的 PostgreSQL 9.6 安装，则系统中不存在此目录。如果是，请以 root 用户身份运行以下命令备份它：

```
mv /var/opt/rh/rh-postgresql96/lib/pgsql/data{,-scl-backup}
```

4.

以 root 用户身份运行以下命令来升级新服务器的数据库数据：

```
scl enable rh-postgresql96 -- postgresql-setup --upgrade --upgrade-from=rh-postgresql95-postgresql
```

或者，您可以使用

`/opt/rh/rh-postgresql96/root/usr/bin/postgresql-setup --upgrade --upgrade-from=rh-postgresql95-postgresql` 命令。

请注意，您可以使用 `--upgrade-from` 选项来从不同版本的 PostgreSQL 升级。可以使用 `--upgrade-ids` 选项进行可能的升级场景列表。

建议您读取生成的 `/var/lib/pgsql/upgrade_rh-postgresql96-postgresql.log` 日志文件，以了解升级过程中是否存在任何问题。

5.

以 `root` 用户身份启动新服务器：

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

另外，建议您运行 `analyze_new_cluster.sh` 脚本，如下所示：

```
su - postgres -c 'scl enable rh-postgresql96 ~/analyze_new_cluster.sh'
```

6.

另外，您可以将 PostgreSQL 9.6 服务器配置为在引导时自动启动。要禁用旧的 PostgreSQL 9.5 服务器，请以 `root` 用户身份输入以下命令：

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

要启用 PostgreSQL 9.6 服务器，以 `root` 用户身份输入：

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

7.

如果您的配置与默认文件不同，请确保更新配置文件，特别是 `/var/opt/rh/rh-postgresql96/lib/pgsql/data/pg_hba.conf` 配置文件。否则，只有 `postgres` 用户才能访问数据库。

### 过程 5.8. 执行转储和恢复升级

要执行 PostgreSQL 服务器的转储和恢复升级，请完成以下步骤：

1.

以 `root` 用户身份在 `shell` 提示符后输入以下内容来确保旧的 PostgreSQL 服务器正在运行：

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

2.

将 PostgreSQL 数据库中的所有数据转储到脚本文件中。以 `root` 用户身份，输入：

```
su - postgres -c 'scl enable rh-postgresql95 "pg_dumpall > ~/pgdump_file.sql"'
```

3. 以 root 用户身份运行以下命令来停止旧服务器：

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

4. 以 root 用户身份初始化新服务器的数据目录：

```
scl enable rh-postgresql96-postgresql -- postgresql-setup --initdb
```

5. 以 root 用户身份启动新服务器：

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

6. 从之前创建的 SQL 文件导入数据：

```
su - postgres -c 'scl enable rh-postgresql96 "psql -f ~/pgdump_file.sql postgres"'
```

7. 另外，您可以将 PostgreSQL 9.6 服务器配置为在引导时自动启动。要禁用旧的 PostgreSQL 9.5 服务器，请以 root 用户身份输入以下命令：

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

要启用 PostgreSQL 9.6 服务器，以 root 用户身份输入：

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

8. 如果您的配置与默认文件不同，请确保更新配置文件，特别是 `/var/opt/rh/rh-postgresql96/lib/pgsql/data/pg_hba.conf` 配置文件。否则，只有 postgres 用户才能访问数据库。

如果您需要从 postgresql92 软件集合进行迁移，请参阅 [Red Hat Software Collections 2.0 发行注记](#)；这个过程相同，您只需要调整新集合的版本。这同样适用于从 rh-postgresql94 Software Collection 迁移，这在 [Red Hat Software Collections 2.2 发行注记](#) 中进行了描述。

## 5.7. 迁移到 NGINX 1.12

**rh-nginx112 Software Collection** 仅适用于 Red Hat Enterprise Linux 7.4 及更新的版本。

**rh-nginx112 Software Collection** 的根目录位于 `/opt/rh/rh-nginx112/root/` 中。默认情况下，错误日志存储在 `/var/opt/rh/rh-nginx112/log/nginx` 中。

配置文件存储在 `/etc/opt/rh/rh-nginx112/nginx/` 目录中。nginx 1.12 中的配置文件的语法相同，其格式与之前的 nginx Software Collections 相同。

`/etc/opt/rh/rh-nginx112/nginx/default.d/` 目录中的配置文件（带有 `.conf` 扩展名）包含在端口 80 的默认服务器块配置中。



### 重要

在从 nginx 1.10 升级到 nginx 1.12 之前，请备份您的所有数据，包括位于 `/opt/rh/nginx110/root/tree` 中的 web 页面，以及位于 `/etc/opt/rh/nginx110/nginx/` 树中的配置文件。

如果您进行了任何特定的更改，如更改配置文件或设置 Web 应用程序（在 `/opt/rh/nginx110/root/` 树中），在新的 `/opt/rh/rh-nginx112/root/` 目录中复制这些更改，以及 `/etc/opt/rh/rh-nginx112/nginx/` 目录中。

您可以使用此流程直接从 nginx 1.4、nginx 1.6、nginx 1.8 或 nginx 1.10 升级到 nginx 1.12。在这种情况下，使用适当的路径。

有关官方 nginx 文档，请参阅 <http://nginx.org/en/docs/>。

## 第 6 章 其它资源

本章介绍了有关 Red Hat Software Collections 3.1 和 Red Hat Enterprise Linux 的其他相关信息。

### 6.1. 红帽产品文档

以下文档与本书直接或间接相关：

- [Red Hat Software Collections 3.1 打包指南](#) - Red Hat Software Collections 的打包指南解释了 Software Collections 的概念，记录 scl 实用程序，并提供了有关如何创建自定义 Software Collection 或扩展现有软件集合的详细说明。
- [Red Hat Developer Toolset 7.1 发行注记](#) - Red Hat Developer Toolset 的发行注记 记录了已知问题、更改以及有关此 Software Collection 的其他重要信息。
- [Red Hat Developer Toolset 7.1 用户指南](#) - Red Hat Developer Toolset 的用户指南 包含有关安装和使用此 Software Collection 的更多信息。
- [使用 Red Hat Software Collections 容器镜像](#) - 本书提供了有关如何使用基于 Red Hat Software Collections 的容器镜像的信息。可用的容器镜像包括应用程序、守护进程、数据库和 Red Hat Developer Toolset 容器镜像。镜像可以在 Red Hat Enterprise Linux 7 服务器和 Red Hat Enterprise Linux Atomic Host 上运行。
- [容器入门](#) - 本指南概述了在 Red Hat Enterprise Linux 7 和 Red Hat Enterprise Linux Atomic Host 上构建和使用 docker 格式的容器镜像的信息。
- [使用和配置 Red Hat Subscription Manager](#) - 使用和配置 Red Hat Subscription Manager 一书提供了有关如何注册 Red Hat Enterprise Linux 系统、管理订阅和查看注册系统通知的详细信息。
- [Red Hat Enterprise Linux 6 部署指南](#) - Red Hat Enterprise Linux 6 的部署指南提供了有关此系统的部署、配置和管理的信息。
- [Red Hat Enterprise Linux 7 系统管理员指南](#) - Red Hat Enterprise Linux 7 系统管理员指南提供了有关此系统部署、配置和管理的信息。

## 6.2. RED HAT DEVELOPERS

- [红帽开发人员计划](#) - 红帽开发人员社区 门户。
- [Red Hat Developers 上的 Red Hat Software Collections 概述](#) - Red Hat Developers 门户 提供了很多教程，供您使用不同的开发技术开发代码。这包括 Node.js、Perl、PHP、Python 和 Ruby Software Collections。
- [Red Hat Enterprise Linux 开发人员计划](#) - Red Hat Enterprise Linux 开发人员计划 提供业界领先的开发人工具、指导资源和专家生态系统，以帮助编程人员最大限度地提高构建 Linux 应用程序的生产力。
- [Red Hat Developer Blog](#) - 红帽开发人员博客包含最新的信息、最佳实践、建议、产品和计划公告，以及有关基于红帽技术设计和开发应用程序的资源的指针。



## 附录 A. 修订历史记录

<b>修订 3.1-4</b> 修复了有问题的链接和定位符。	Wed Dec 20 2023	Lenka Špačková
<b>修订 3.1-3</b> 更新了 <a href="#">第 4.9 节 “数据库连接器”</a> 。	Fri Nov 12 2021	Lenka Špačková
<b>修订 3.1-2</b> 修复了 PostgreSQL 迁移章节中命令的语法。	Tue Dec 18 2018	Lenka Špačková
<b>修订 3.1-1</b> 红帽软件集合 3.1 发行注记发布。	Thu May 03 2018	Lenka Špačková
<b>修订 3.1-0</b> 红帽软件集合 3.1 Beta 发行注记发布。	Wed Mar 04 2018	Lenka Špačková