



Red Hat Software Collections 3

3.4 发行注记

Red Hat Software Collections 3.4 发行注记

Red Hat Software Collections 3 3.4 发行注记

Red Hat Software Collections 3.4 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 © 2019-2020 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

摘要

Red Hat Software Collections 3.4 发行注记记录了主要功能，并包含有关 Red Hat Software Collections 3.4 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

第 1 章 RED HAT SOFTWARE COLLECTIONS 3.4	4
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 3.4 中的更改	14
1.3.1. 概述	14
构架	14
新 Software Collections	14
更新了 Software Collections	15
Red Hat Software Collections 容器镜像	15
1.3.2. Red Hat Developer Toolset 中的更改	15
1.3.3. Node.js 中的更改	16
1.3.4. PHP 中的更改	16
1.3.5. nginx 中的更改	17
1.3.6. PostgreSQL 中的更改	18
1.3.7. Maven 中的更改	19
1.3.8. Apache httpd 中的更改	19
1.4. 兼容性信息	19
1.5. 已知问题	19
其他备注	23
1.6. 弃用的功能	25
第 2 章 安装	26
2.1. 获取 RED HAT SOFTWARE COLLECTIONS	26
2.1.1. 使用 Red Hat Subscription Management	26
2.1.2. 可选频道的软件包	28
2.2. 安装 RED HAT SOFTWARE COLLECTIONS	30
2.2.1. 安装单个 Software Collections	30
2.2.2. 安装可选软件包	31
2.2.3. 安装调试信息	31
2.3. 卸载 RED HAT SOFTWARE COLLECTIONS	31
2.4. 重建 RED HAT SOFTWARE COLLECTIONS	32
第 3 章 使用方法	33
3.1. 使用 RED HAT SOFTWARE COLLECTIONS	33
3.1.1. 从 Software Collection 运行可执行文件	33
3.1.2. 使用 Software Collection 作为默认运行 Shell 会话	33
3.1.3. 从 Software Collection 运行系统服务	34
从 Red Hat Enterprise Linux 6 中的 Software Collection 运行系统服务	34
从 Red Hat Enterprise Linux 7 中的 Software Collection 运行系统服务	34
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	35
3.4. RED HAT SOFTWARE COLLECTIONS 容器镜像	36
第 4 章 独立 SOFTWARE COLLECTIONS 的具体信息	43
4.1. RED HAT DEVELOPER TOOLSET	43
4.2. RUBY ON RAILS 5.0	43
4.3. MONGODB 3.6	44
4.4. MONGODB 3.4	45
MongoDB 3.4 on Red Hat Enterprise Linux 6	45
MongoDB 3.4 on Red Hat Enterprise Linux 7	46
4.5. MAVEN	47

4.6. PASSENGER	47
4.7. 数据库连接器	48
第 5 章 MIGRATION (迁移)	50
5.1. 迁移到 MARIADB 10.3	50
5.1.1. rh-mariadb102 和 rh-mariadb103 Software Collections 之间的显著差异	50
5.1.2. 从 rh-mariadb102 升级到 rh-mariadb103 Software Collection	51
5.2. 迁移到 MARIADB 10.2	52
5.2.1. rh-mariadb101 和 rh-mariadb102 Software Collections 之间的显著差异	53
5.2.2. 从 rh-mariadb101 升级到 rh-mariadb102 Software Collection	53
5.3. 迁移到 MYSQL 8.0	55
5.3.1. MySQL 5.7 和 MySQL 8.0 之间的显著差异	55
特定于 rh-mysql80 Software Collection 的不同	55
MySQL 8.0 中的常规更改	56
5.3.2. 升级到 rh-mysql80 Software Collection	57
5.4. 迁移到 MONGODB 3.6	58
5.4.1. MongoDB 3.4 和 MongoDB 3.6 之间的显著差异	58
常规更改	58
兼容性更改	58
后向兼容功能	59
5.4.2. 从 rh-mongodb34 升级到 rh-mongodb36 Software Collection	59
5.5. 迁移到 MONGODB 3.4	61
5.5.1. 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	62
5.5.2. 从 rh-mongodb32 升级到 rh-mongodb34 Software Collection	62
5.6. 迁移到 POSTGRESQL 12	64
5.6.1. 从 Red Hat Enterprise Linux System 版本的 PostgreSQL 迁移到 PostgreSQL 12 Software Collection	65
5.6.2. 从 PostgreSQL 10 Software Collection 迁移到 PostgreSQL 12 Software Collection	69
5.7. 迁移到 POSTGRESQL 9.6	72
5.7.1. PostgreSQL 9.5 和 PostgreSQL 9.6 之间的显著差异	72
5.7.2. 从 Red Hat Enterprise Linux System 版本的 PostgreSQL 迁移到 PostgreSQL 9.6 Software Collection	74
5.7.3. 从 PostgreSQL 9.5 Software Collection 迁移到 PostgreSQL 9.6 Software Collection	77
5.8. 迁移到 NGINX 1.16	81
5.9. 迁移到 REDIS 5	81
兼容性备注	82
第 6 章 其它资源	83
6.1. 红帽产品文档	83
6.2. RED HAT DEVELOPERS	84
附录 A. 修订历史记录	85

第 1 章 RED HAT SOFTWARE COLLECTIONS 3.4

本章作为 Red Hat Software Collections 3.4 内容集的概述。它提供组件及其描述列表，以及此版本的更改，记录了相关兼容性信息，以及列出已知问题。

1.1. 关于 RED HAT SOFTWARE COLLECTIONS

对于某些应用程序，通常需要较新版本的部分软件组件，以便使用其最新的新功能。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 3.4 适用于 Red Hat Enterprise Linux 7；为 Red Hat Enterprise Linux 6 选择了之前发布的组件。有关作为 Red Hat Software Collections 一部分分发的组件的完整列表及其功能的简要概述，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，默认 Perl 或 PostgreSQL 版本保留由基本 Red Hat Enterprise Linux 系统提供的版本。



注意

在 Red Hat Enterprise Linux 8 中，类似组件作为 [Application Streams](#) 提供。

所有 Red Hat Software Collections 组件在 Red Hat Enterprise Linux 订阅级别协议中被完全支持，其功能已完成，并适用于生产环境。在 Red Hat Software Collections 订阅者以类似 Red Hat Enterprise Linux 的方式向 Red Hat Software Collections 订阅者发布重要的程序错误修复和安全勘误，自每个主要版本发布后至少两年。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。有关独立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 3.4 组件”](#) 列出 Red Hat Software Collections 3.4 发行版本时支持的组件。

表 1.1. Red Hat Software Collections 3.4 组件

组件	Software Collection	描述
Red Hat Developer Toolset 9.0	devtoolset-9	Red Hat Developer Toolset 是为在 Red Hat Enterprise Linux 平台上工作的开发人员设计的。它提供了 GNU Compiler Collection 、 GNU Debugger 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 Red Hat Developer Toolset 用户指南中的 Red Hat Developer Toolset 组件表 。

组件	Software Collection	描述
Perl 5.26.3 ^[a]	rh-perl526	Perl 的一个高级编程语言，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 DateTime Perl 模块和 mod_perl Apache httpd 模块，该模块仅支持 httpd24 软件集合。另外，它还轻松安装 CPAN 模块提供 cpanm 工具。rh-perl526 打包与上游一致；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 7.2.24 ^[a]	rh-php72	使用 PEAR 1.10.5、APCu 5.1.12 和增强语言功能的 PHP 7.2 发行版本。
PHP 7.3.11 ^[a]	rh-php73	带有 PEAR 1.10.9、APCu 5.1.17 和 Xdebug 扩展的 PHP 7.3 发行版本。
Python 2.7.16	python27	带有其他工具的 Python 2.7 发行版本。这个 Python 版本提供各种功能和增强功能，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 解释器</i> ，一组用于编程 Web 应用程序和 mod_wsgi 的扩展库（只在 httpd24 Software Collection 中被支持）、MySQL 和 PostgreSQL 数据库连接器，以及 numpy 和 scipy 。
Python 3.6.9	rh-python36	rh-python36 Software Collection 包含 Python 3.6.9，它引入了多个新功能，如 <i>f-strings</i> 、 <i>变量注解语法</i> 以及 <i>异步生成器</i> 和 <i>理解</i> 。此外，还包括一组适用于编程 Web 应用的扩展库， mod_wsgi （仅支持 httpd24 软件集合）、PostgreSQL 数据库连接器、 numpy 和 scipy 。
Ruby 2.4.6	rh-ruby24	Ruby 2.4 发布。这个版本提供多个性能改进和增强，例如 <i>改进的哈希表</i> 、 <i>新的调试功能</i> 、 <i>对 Unicode 案例映射的支持</i> ，以及 <i>OpenSSL 1.1.0 的支持</i> 。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。
Ruby 2.5.5 ^[a]	rh-ruby25	Ruby 2.5 发布。此版本提供多个性能改进和新功能，例如， <i>简化了带有 rescue、else 和 ensure 关键字的块的使用</i> ，一个新的 yield_self 方法， <i>支持分支覆盖和方法覆盖测量</i> 、 <i>新的 Hash#slice 和 Hash#transform_keys 方法</i> 。Ruby 2.5.0 维护与 Ruby 2.4 的源级向后兼容性。
Ruby 2.6.2 ^[a]	rh-ruby26	Ruby 2.6 发行版本。此版本提供多个性能改进和新功能，如 <i>无终端 范围、Binding#source_location 方法和 \$SAFE 进程全局状态</i> 。Ruby 2.6.0 维护与 Ruby 2.5 的源级向后兼容性。

组件	Software Collection	描述
Ruby on Rails 5.0.1	rh-ror50	Ruby on Rails 5.0 发布，这是使用 Ruby 语言编写的 Web 应用程序框架的最新版本。主要的新功能包括 <i>Action Cable</i> 、 <i>API mode</i> 、 <i>exclusive using rails CLI over Rake</i> 、和 <i>ActionRecord</i> 属性。这个 Software Collection 与 rh-ruby24 Collection 一起被支持。
Scala 2.10.6 [a]	rh-scala210	Scala 是 Java 平台的通用编程语言版本，集成了面向对象和功能语言的功能。
MariaDB 10.2.22	rh-mariadb102	MariaDB 发布，这是 Red Hat Enterprise Linux 用户的替代选择。就所有实际目的而言，MySQL 与 MariaDB 兼容，可以在没有任何数据转换的情况下替换为它。此版本添加了 <i>MariaDB 备份</i> 、 <i>Flashback</i> 、支持 <i>Recursive Common Table Expressions</i> 、 <i>窗口功能</i> 和 <i>JSON 功能</i> 。
MariaDB 10.3.13 [a]	rh-mariadb103	MariaDB 发布，这是 Red Hat Enterprise Linux 用户的替代选择。就所有实际目的而言，MySQL 与 MariaDB 兼容，可以在没有任何数据转换的情况下替换为它。此版本引入了 <i>系统版本的表</i> 、 <i>不可见的列</i> 、 InnoDB 的 <i>新即时 ADD COLUMN 操作</i> ，以及适用于 <i>MariaDB</i> 和 <i>MySQL</i> 的 <i>JDBC 连接器</i> 。
MongoDB 3.4.9	rh-mongodb34	MongoDB 的一个发行版本，是一个面向跨平台文档的数据库系统，被分类为一个 NoSQL 数据库。此发行版本引入了对 <i>新架构的支持</i> ， <i>增加了对十进制28 类型的消息压缩和支持</i> ， <i>增强了联合功能</i> 等。
MongoDB 3.6.3 [a]	rh-mongodb36	MongoDB 的一个发行版本，是一个面向跨平台文档的数据库系统，被分类为一个 NoSQL 数据库。此发行版本引入了 <i>更改流</i> 、 <i>可重试的写入</i> 和 <i>JSON Schema</i> ，以及其他功能。
MySQL 8.0.17 [a]	rh-mysql80	MySQL 服务器的发行版本，它引入了多个新的 <i>安全性和帐户管理功能</i> 及 <i>增强功能</i> 。
PostgreSQL 9.6.10	rh-postgresql96	PostgreSQL 的一个发行版本，它引入了 <i>后续扫描</i> 、 <i>加入和聚合的并行执行</i> ，并提供 <i>同步复制</i> 、 <i>全文本搜索</i> 、 <i>重复数据删除驱动程序</i> 、 <i>postgres_fdw</i> 以及 <i>性能改进</i> 的改进。
PostgreSQL 10.6 [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 publish 和 subscribe 关键字的 <i>逻辑复制制</i> ，或者基于 SCRAM-SHA-256 机制的 <i>更强大的密码身份验证</i> 。
PostgreSQL 12.1 [a]	rh-postgresql12	PostgreSQL 的一个发行版本，它提供了 pgaudit 扩展， <i>对分区和并行性的各种增强</i> ， <i>支持 SQL/JSON 路径语言</i> 和 <i>性能改进</i> 。

组件	Software Collection	描述
Node.js 10.16.3 [a]	rh-nodejs10	Node.js 发行版本提供多个 API 增强功能和新功能，包括 V8 引擎版本 6.6、完整的 N-API 支持以及稳定性改进。
Node.js 12.10.0 [a]	rh-nodejs12	一个 Node.js 发行版本，带有 V8 引擎版本 7.6，支持 ES6 模块，并改进了对原生模块的支持。
nginx 1.10.2	rh-nginx10	nginx 版本（一个 Web 和代理服务器），专注于高并发性、性能和低内存用量。此版本引入了许多新功能，包括动态模块支持、HTTP/2 支持、Perl 集成以及大量性能改进。
nginx 1.14.1 [a]	rh-nginx14	nginx 版本（一个 Web 和代理服务器），专注于高并发性、性能和低内存用量。此版本提供了很多功能，如 mirror 模块、HTTP/2 服务器推送、gRPC 代理模块以及大量性能改进。
nginx 1.16.1 [a]	rh-nginx16	nginx 版本（一个 Web 和代理服务器），专注于高并发性、性能和低内存用量。此版本引入了多个与 SSL 相关的更新、几个新指令和参数以及各种改进。
Apache httpd 2.4.34	httpd24	发布 Apache HTTP 服务器(httpd)，包括高性能事件处理模型，增强的 SSL 模块和 FastCGI 支持。也会包含 mod_auth_kerb、mod_auth_mellon 和 ModSecurity 模块。
Varnish Cache 5.2.1 [a]	rh-varnish5	Varnish 缓存发行版本，它是一个高性能 HTTP 反向代理。此版本包括分片 director、实验性 HTTP/2 支持以及通过单独的 VCL 文件和 VCL 标签对 Varnish 配置的改进。
Varnish Cache 6.0.2 [a]	rh-varnish6	Varnish 缓存发行版本，它是一个高性能 HTTP 反向代理。此版本包括对 Unix 域套接字的支持（包括客户端和后端服务器）、新级别的 VCL 语言(vcl 4.1)，以及改进的 HTTP/2 支持。
Maven 3.5.0 [a]	rh-maven35	Maven 发布，这是一个软件项目管理和理解工具。此发行版本引进了对新架构和许多新功能的支持，包括颜色的日志记录。
Maven 3.6.1 [a]	rh-maven36	Maven 发布，这是一个软件项目管理和理解工具。此发行版本提供各种改进和程序错误修复。
Git 2.18.1 [a]	rh-git218	Git 的一个发行版本，即具有分散架构的分布式修订控制系统。与使用客户端-服务器模型的集中版本控制系统外，Git 确保 Git 存储库的每个工作副本都与完整的修订历史记录完全相同。此版本包括大文件存储(LFS)扩展。
Redis 5.0.5 [a]	rh-redis5	Redis 5.0 发布，它是一个持久的键值数据库。Redis 现在提供 redis-trib，它是一个集群管理工具。

组件	Software Collection	描述
HAProxy 1.8.17 [a]	rh-haproxy18	HAProxy 1.8 版本，这是一个可靠、高性能的 <i>网络负载均衡器</i> ，适用于 TCP 和基于 HTTP 的应用程序。
常见 Java 软件包	rh-java-common	此 Software Collection 提供了由 <i>其他集合使用的通用 Java 库和工具</i> 。rh-java-common Software Collection 需要 rh-maven35 和 rh-scala210 组件，用户不应直接安装它。
JDK Mission Control [a]	rh-jmc	Software Collection 包括 <i>JDK Mission Control (JMC)</i> ，这是 HotSpot JVM 的强大配置集器。JMC 提供了一组高级工具，用于对 JDK Flight Recorder 收集的大量数据进行高效和详细分析。JMC 需要运行 JDK 版本 8 或更高版本。目标 Java 应用程序必须至少使用 OpenJDK 版本 11 运行，以便 JMC 可以访问 JDK Flight Recorder 功能。rh-jmc Software Collection 需要 rh-maven35 Software Collection。

[\[a\]](#) 这个 Software Collection 仅适用于 Red Hat Enterprise Linux 7

以前发布的 Software Collections 在相同的发行版本频道中仍然可用。所有 Software Collections（包括已停用组件）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。有关之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 支持的架构
Red Hat Software Collections 3.4 中的组件			
Red Hat Developer Toolset 9.0	devtoolset-9	RHEL7	x86_64, s390x, aarch64, ppc64, ppc64le
Node.js 12.10.0	rh-nodejs12	RHEL7	x86_64, s390x, aarch64, ppc64le
PHP 7.3.11	rh-php73	RHEL7	x86_64, s390x, aarch64, ppc64le
nginx 1.16.1	rh-nginx116	RHEL7	x86_64, s390x, aarch64, ppc64le
PostgreSQL 12.1	rh-postgresql12	RHEL7	x86_64, s390x, aarch64, ppc64le

组件	Software Collection	可用性	RHEL7 支持的架构
Red Hat Software Collections 3.4 中的组件			
Maven 3.6.1	rh-maven36	RHEL7	x86_64, s390x, aarch64, ppc64le

Red Hat Software Collections 3.4 中的组件更新			
Apache httpd 2.4.34	httpd24	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64le

在 Red Hat Software Collections 3.3 中更新的组件			
Red Hat Developer Toolset 8.1	devtoolset-8	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64, ppc64le
MariaDB 10.3.13	rh-mariadb103	RHEL7	x86_64, s390x, aarch64, ppc64le
Redis 5.0.5	rh-redis5	RHEL7	x86_64, s390x, aarch64, ppc64le
Ruby 2.6.2	rh-ruby26	RHEL7	x86_64, s390x, aarch64, ppc64le
HAProxy 1.8.17	rh-haproxy18	RHEL7	x86_64
Varnish Cache 6.0.2	rh-varnish6	RHEL7	x86_64, s390x, aarch64, ppc64le

在 Red Hat Software Collections 3.2 中最后更新的组件			
PHP 7.2.24	rh-php72	RHEL7	x86_64, s390x, aarch64, ppc64le
MySQL 8.0.17	rh-mysql80	RHEL7	x86_64, s390x, aarch64, ppc64le
Node.js 10.16.3	rh-nodejs10	RHEL7	x86_64, s390x, aarch64, ppc64le
nginx 1.14.1	rh-nginx114	RHEL7	x86_64, s390x, aarch64, ppc64le

在 Red Hat Software Collections 3.2 中最后更新的组件

Git 2.18.1	rh-git218	RHEL7	x86_64, s390x, aarch64, ppc64le
JDK Mission Control	rh-jmc	RHEL7	x86_64

在 Red Hat Software Collections 3.1 中更新的组件

Red Hat Developer Toolset 7.1	devtoolset-7*	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64, ppc64le
Perl 5.26.3	rh-perl526	RHEL7	x86_64, s390x, aarch64, ppc64le
Ruby 2.5.5	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 10.6	rh-postgresql10	RHEL7	x86_64, s390x, aarch64, ppc64le
PHP 7.0.27	rh-php70*	RHEL6, RHEL7	x86_64
MySQL 5.7.24	rh-mysql57*	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64le

在 Red Hat Software Collections 3.0 中更新的组件

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 3.6.9	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 中更新的组件

MariaDB 10.2.22	rh-mariadb102	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64le
PostgreSQL 9.6.10	rh-postgresql96	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64le
MongoDB 3.4.9	rh-mongodb34	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64le
Node.js 8.11.4	rh-nodejs8*	RHEL7	x86_64, s390x, aarch64, ppc64le

在 Red Hat Software Collections 2.4 中更新的组件

Red Hat Developer Toolset 6.1	devtoolset-6*	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64, ppc64le
scala 2.10.6	rh-scala210	RHEL7	x86_64
nginx 1.10.2	rh-nginx110	RHEL6, RHEL7	x86_64
Node.js 6.11.3	rh-nodejs6*	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64le
Ruby 2.4.6	rh-ruby24	RHEL6, RHEL7	x86_64
Ruby on Rails 5.0.1	rh-ror50	RHEL6, RHEL7	x86_64
Eclipse 4.6.3	rh-eclipse46*	RHEL7	x86_64
Python 2.7.16	python27	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64le
Thermostat 1.6.6	rh-thermostat16*	RHEL6, RHEL7	x86_64
Maven 3.3.9	rh-maven33*	RHEL6, RHEL7	x86_64
常见 Java 软件包	rh-java-common	RHEL6, RHEL7	x86_64

在 Red Hat Software Collections 2.3 中更新的组件

Git 2.9.3	rh-git29*	RHEL6, RHEL7	x86_64, s390x, aarch64, ppc64le
-----------	-----------	--------------	---------------------------------

在 Red Hat Software Collections 2.3 中更新的组件

Redis 3.2.4	rh-redis32*	RHEL6, RHEL7	x86_64
Perl 5.24.0	rh-perl524*	RHEL6, RHEL7	x86_64
Python 3.5.1	rh-python35*	RHEL6, RHEL7	x86_64
MongoDB 3.2.10	rh-mongodb32*	RHEL6, RHEL7	x86_64
Ruby 2.3.8	rh-ruby23*	RHEL6, RHEL7	x86_64
PHP 5.6.25	rh-php56*	RHEL6, RHEL7	x86_64

在 Red Hat Software Collections 2.2 中更新的组件

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 9.5.14	rh-postgresql95*	RHEL6, RHEL7	x86_64
Ruby on Rails 4.2.6	rh-ror42*	RHEL6, RHEL7	x86_64
MongoDB 2.6.9	rh-mongodb26*	RHEL6, RHEL7	x86_64
Thermostat 1.4.4	thermostat1*	RHEL6, RHEL7	x86_64

在 Red Hat Software Collections 2.1 中更新的组件

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

在 Red Hat Software Collections 2.0 中更新的组件

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 中更新的组件

Git 1.9.4	git19*	RHEL6, RHEL7	x86_64
Perl 5.16.3	perl1516*	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 中更新的组件

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

图例：

- RHEL6 - Red Hat Enterprise Linux 6
- RHEL7 - Red Hat Enterprise Linux 7
- x86_64 - AMD64 和 Intel 64 架构
- s390x - IBM Z
- aarch64 - 64 位 ARM 架构
- ppc64 - IBM POWER, big endian
- ppc64le - IBM POWER, little endian
- * - 停用的组件；此 Software Collection 不再被支持

以上表格列出了通过异步更新提供的最新版本。

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。

Eclipse 作为 [Red Hat Developer Tools](#) 产品的一部分提供。

1.3. RED HAT SOFTWARE COLLECTIONS 3.4 中的更改

1.3.1. 概述

构架

Red Hat Software Collections 产品包含在 AMD64 和 Intel 64 架构上运行的 Red Hat Enterprise Linux 7 的软件包；某些以前的 Software Collections 也可用于 Red Hat Enterprise Linux 6。

另外，Red Hat Software Collections 3.4 在 Red Hat Enterprise Linux 7 中支持以下架构：

- 64 位 ARM 架构
- IBM Z
- 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 3.4 添加以下新 Software Collections：

- devtoolset-9 – see [第 1.3.2 节 “Red Hat Developer Toolset 中的更改”](#)

- rh-nodejs12 – see [第 1.3.3 节 “Node.js 中的更改”](#)
- rh-php73 – see [第 1.3.4 节 “PHP 中的更改”](#)
- rh-nginx116 – see [第 1.3.5 节 “nginx 中的更改”](#)
- rh-postgresql12 – see [第 1.3.6 节 “PostgreSQL 中的更改”](#)
- rh-maven36 – see [第 1.3.7 节 “Maven 中的更改”](#)

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

更新了 **Software Collections**

在 Red Hat Software Collections 3.4 中更新了以下组件：

- httpd24 – see [第 1.3.8 节 “Apache httpd 中的更改”](#)

Red Hat Software Collections 容器镜像

以下容器镜像在 Red Hat Software Collections 3.4 中是新的：

- rhsc/devtoolset-9-toolchain-rhel7
- rhsc/devtoolset-9-perftools-rhel7
- rhsc/nodejs-12-rhel7
- rhsc/php-73-rhel7
- rhsc/nginx-116-rhel7
- rhsc/postgresql-12-rhel7

Red Hat Software Collections 3.4 中更新了以下容器镜像：

- rhsc/httpd-24-rhel7

有关 Red Hat Software Collections 容器镜像的详细信息，请参阅 [第 3.4 节 “Red Hat Software Collections 容器镜像”](#)。

1.3.2. Red Hat Developer Toolset 中的更改

与之前的 Red Hat Developer Toolset 发行版本相比，在 Red Hat Developer Toolset 中已升级了以下组件：

- **GCC** 到版本 9.1.1
- **binutils** 到版本 2.32
- **GDB** 升级到版本 8.3
- **strace** 到版本 5.1
- **SystemTap** 到版本 4.1
- **Valgrind** 到版本 3.15.0
- **Dyninst** 到版本 10.1.0

有关 9.0 中更改的详细信息，请参阅 [Red Hat Developer Toolset 用户指南](#)。

1.3.3. Node.js 中的更改

新的 rh-nodejs12 Software Collection 提供 **Node.js 12.10.0**。这个版本中的显著改进包括：

- V8 引擎升级至版本 7.6
- 新的默认 HTTP 解析器 **llhttp**（不再实验性）
- 堆转储生成的集成功能
- 支持 ECMAScript 2015 (ES6) 模块
- 改进了原生模块的支持
- worker 线程不再需要一个标记
- 新的实验性诊断报告功能
- 提高的性能

有关 Node.js 12.10.0 的详细信息，请参阅[上游发行注记](#)和[上游文档](#)。

1.3.4. PHP 中的更改

使用 **PHP 7.3.11** 的新 rh-php73 Software Collection 比 rh-php72 引入了以下显著改进：

- 用于开发的 Xdebug 扩展
- 增强且更灵活的 **heredoc** 和 **nowdoc** 语法
- PCRE 扩展升级到 PCRE2
- 改进了多字节字符串处理
- 支持 LDAP 控制
- 改进了 FastCGI 进程管理器(FPM)日志记录
- 在功能调用中添加结尾逗号的功能
- 提高的性能
- 几个弃用和后不兼容的更改

如需更多信息，请参阅[从 PHP 7.2.x 迁移到 PHP 7.3.x](#)。

请注意，以下行为与上游不同：

- rh-php73 Software Collection 不支持 **Argon2** 密码哈希算法。
- X (PCRE_EXTENDED) 模式修饰符始终在 rh-php73 软件集中启用。**因此，无效的转义序列不会被解释为字面。**

1.3.5. nginx 中的更改

新的 rh-nginx116 Software Collection 提供 nginx 1.16.1，它引入了一些新功能和增强。例如：

- 与 SSL 相关的大量更新（从变量加载 SSL 证书和密钥，在 `ssl_certificate` 和 `ssl_certificate_key` 指令中支持，一个新的 `ssl_early_data` 指令）
- 新的 `keepalive`- 相关指令
- 用于分布式负载均衡的新 `random` 指令
- 对现有指令的新参数和改进（`listen` 指令的端口范围，用于 `limit_req` 指令的新 `delay` 参数，它可启用双阶段速率限制）
- 新的 `$upstream_bytes_sent` 变量
- 用户数据报协议(UDP)代理的改进

其他显著变化包括：

- `ssl` 指令已弃用；改为使用 `ssl` 参数作为 `listen` 指令。
- nginx 现在会在配置测试过程中检测缺少的 SSL 证书。
- 当在 `listen` 指令中使用主机名时，nginx 现在为主机名解析到的所有地址创建侦听套接字。

有关 nginx 中更改的更多信息，[请参阅上游发行注记](#)。

有关迁移说明，[请参阅第 5.8 节“迁移到 nginx 1.16”](#)。

1.3.6. PostgreSQL 中的更改

新的 rh-postgresql12 Software Collection 包括 PostgreSQL 12.1。与之前的 Software Collection 版本中发布的版本 10 相比，这个版本引入了各种改进，例如：

- PostgreSQL Audit 扩展 pgaudit，它通过标准的 PostgreSQL 日志记录功能提供详细的会话和对象审计日志记录。
- 分区功能的改进，例如，对 hash 分区的支持
- 查询并行性的改进
- 启用事务管理存储的 SQL 流程
- 各种性能改进
- 管理功能的改进
- 支持 SQL/JSON 路径语言
- 存储生成的列
- Nondeterministic collations
- 新的身份验证功能，包括在使用 GSSAPI 身份验证或多因素身份验证时对 TCP/IP 连接进行加密。

有关详细更改，请参阅 [PostgreSQL 11](#) 和 [PostgreSQL 12](#) 的上游发行注记。

请注意，自 PostgreSQL 11 起，对 Just-In-Time (JIT)编译的支持没有由 rh-postgresql12 软件集合提供。

rh-postgresql12 Software Collection 包括 **rh-postgresql12-syspaths** 软件包，它安装为二进制文件、脚本、手册页等提供系统范围打包程序的软件包。安装 **rh-postgresql12*-syspaths** 软件包后，用户不需要使用 **scl enable** 命令对 **rh-postgresql12*** 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。要了解更多有关 **syspath** 的信息，请参阅 [Red Hat Software Collections Packaging Guide](#)。

有关迁移的详情，请参考 [第 5.6 节“迁移到 PostgreSQL 12”](#)。

1.3.7. Maven 中的更改

带有 **Maven 3.6.1** 的新 **rh-maven36 Software Collection** 包括大量程序错误修复和增强。有关详细更改，请参阅 [上游发行注记](#)。

1.3.8. Apache httpd 中的更改

httpd24 Software Collection 已更新，以提供几个安全和程序错误修复。

1.4. 兼容性信息

Red Hat Software Collections 3.4 适用于所有在 **AMD64** 和 **Intel 64** 构架中的 **Red Hat Enterprise Linux 7** 版本、**64 位 ARM 架构**、**IBM Z** 和 **IBM POWER, little endian**。

某些组件也可用于 **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

多个组件, [BZ#1716378](#)

Software Collections debuginfo 软件包提供的某些文件可能与来自基本 **Red Hat Enterprise Linux** 系统或其他版本的 **Red Hat Software Collections** 组件的对应 **debuginfo** 软件包文件冲突。例如，**python27-python-debuginfo** 软件包文件可能与核心系统上安装的 **python-debuginfo** 软件包中的相应文件冲突。同样，**httpd24-mod_auth_mellon-debuginfo** 软件包中的文件可能与基本系统 **mod_auth_mellon-debuginfo** 软件包提供的类似文件冲突。要临时解决这个问题，请在安装 **Software Collection debuginfo** 软件包前卸载基本系统 **debuginfo** 软件包。

rh-mysql80 component, BZ#1646363

`mysql-connector-java` 数据库连接器无法用于 MySQL 8.0 服务器。要临时解决这个问题，请使用 `rh-mariadb103 Software Collection` 中的 `mariadb-java-client` 数据库连接器。

rh-mysql80 component, BZ#1646158

MySQL 8.0 中默认字符集已改为 `utf8mb4`，但 `php-mysqlnd` 数据库连接器不支持此字符集。因此，`php-mysqlnd` 无法在默认配置中连接。要临时解决这个问题，请指定已知字符集作为 MySQL 服务器配置参数。例如，将 `/etc/opt/rh/rh-mysql80/my.cnf.d/mysql-server.cnf` 文件修改为：

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

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 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，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，请在 CGI 脚本中根据需要设置 `LD_LIBRARY_PATH`。

httpd24 组件

不支持根据 httpd24 Software Collection 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该变量。

rh-python35,rh-python36 组件, BZ#1499990

Babel 用于时区支持的 pytz 模块不包含在 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 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，这个版本与之前提供的版本 2.5.2 并不完全兼容。

在

在 Red Hat Enterprise Linux 7.5 及更早版本中，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 架构以及 IBM POWER, little endian 上没有正确软件包所有权。因此，当卸载 Software Collection 时，这个目录不会被删除。要临时解决这个问题，在删除 Software Collection 时手动删除 `<collection>/root/usr/lib64/`。

Ruby 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`，构建将失败。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`。

Maven 组件

当用户同时安装了 maven-local 软件包的 Red Hat Enterprise Linux 系统版本以及 rh-maven*-maven-local 软件包 (XMvn) 时，用于构建 Java RPM 软件包的工具（从 Maven Software

Collection 运行) 会尝试从基础系统中读取配置文件, 并失败。要临时解决这个问题, 请从基本 Red Hat Enterprise Linux 系统卸载 `maven-local` 软件包。

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。

PostgreSQL 组件

Red Hat Enterprise Linux 6 的 `rh-postgresql9*` 软件包不提供 `sepgsql` 模块, 因为这个功能需要安装 `libseline` 版本 2.0.99, 这在 Red Hat Enterprise Linux 6 中不可用。

`httpd,mariadb,mongodb,mysql,nodejs,perl,php,python,ruby,ruby`, 和 `ror` 组件, [BZ#1072319](#)

在卸载 `httpd24,rh-mariadb*,rh-mongodb*,rh-mysql*,rh-nodejs*,rh-perl*,rh-php*,python27,rh-python*,rh-ruby*,` 或 `rh-ror*` 软件包时, 卸载的顺序可能是相关的, 因为依赖软件包的所有权相关。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。

MariaDB,mysql 组件, [BZ#1194611](#)

自 MariaDB 10 和 MySQL 5.6 起, `rh-mariadb*-mariadb-server` 和 `rh-mysql*-mysql-server` 软件包不再提供 `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 `test` 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。

MariaDB,mysql,postgresql,mongodb 组件

Red Hat Software Collections 包括 MySQL 5.7,MySQL 8.0,MariaDB 10.2,MariaDB 10.3,PostgreSQL 9.6,PostgreSQL 10,PostgreSQL 12,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。用户需要使用随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。

其他备注

rh-ruby*, rh-python*, rh-php* 组件

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

- 当 rh-ruby* Software Collection 位于只读 NFS 中时，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，配置文件、变量数据和独立集合的运行时数据存储在与之前 Red Hat Software Collections 版本不同的目录中。

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
```

Python 组件

当用户尝试从 python27 和 rh-python* Software Collections 安装多个 scl-devel 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 `rh-php*` 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 `rh-ruby*` 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 `rh-perl*` 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_perl`、`%scl_prefix_perl`)。

Nginx 组件

当用户尝试从 `rh-nginx*` Software Collections 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_nginx`、`%scl_prefix_nginx`)。

1.6. 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

本章详细介绍了如何访问内容集，在系统上安装 Red Hat Software Collections 3.4，并重建 Red Hat Software Collections。

2.1. 获取 RED HAT SOFTWARE COLLECTIONS

Red Hat Software Collections 内容集可供客户使用 Red Hat Enterprise Linux 6 和 Red Hat Enterprise Linux 7 订阅，网址为 <https://access.redhat.com/solutions/472793>。有关如何在 Red Hat Subscription Management (RHSM) 中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，Red Hat Software Collections 和 Red Hat Developer Toolset 内容也以 ISO 格式(<https://access.redhat.com/downloads> 特别是 [Server](#) 和 [Workstation](#)) 提供。请注意，需要 Optional 频道的软件包（在[第 2.1.2 节“可选频道的软件包”](#)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“可选频道的软件包”](#)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

如果您的系统注册到 Red Hat Subscription Management，请完成以下步骤以附加提供对 Red Hat Software Collections 仓库的访问权限并启用存储库的订阅：

1. 显示可用于您的系统的所有订阅的列表，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、到期日期以及与之相关的其他详细信息。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
```

使用您在上一步中确定的池 ID 替换 `pool_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，并确定红帽软件集合存储库的确切名称。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，并采用以下格式：

```
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
```

当订阅附加到系统后，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[使用和配置红帽订阅管理器](#)。



注意

通过 RHN 的订阅不再可用。

2.1.2. 可选频道的软件包

有些 Red Hat Software Collections 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，请参考<https://access.redhat.com/solutions/392003>。

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

表 2.1. 在 Red Hat Enterprise Linux 7 中启用可选频道的软件包

Software Collection 中的软件包	可选频道的所需软件包
devtoolset-8-build	scl-utils-build
devtoolset-8-dyninst-testsuite	glibc-static
devtoolset-8-gcc-plugin-devel	libmpc-devel
devtoolset-9-build	scl-utils-build
devtoolset-9-dyninst-testsuite	glibc-static
devtoolset-9-gcc-plugin-devel	libmpc-devel
devtoolset-9-gdb	source-highlight
httpd24-mod_ldap	apr-util-ldap

Software Collection 中的软件包	可选频道的所需软件包
httpd24-mod_session	apr-util-openssl
python27-python-debug	tix
python27-python-devel	scl-utils-build
python27-tkinter	tix
rh-git218-git-cvs	cvsp
rh-git218-git-svn	perl-Git-SVN, subversion
rh-git218-perl-Git-SVN	subversion-perl
rh-java-common-ant-apache-bsf	rhino
rh-java-common-batik	rhino
rh-maven35-xpp3-javadoc	java-1.7.0-openjdk-javadoc, java-1.8.0-openjdk-javadoc, java-1.8.0-openjdk-javadoc-zip, java-11-openjdk-javadoc, java-11-openjdk-javadoc-zip
rh-php72-php-pspell	aspell
rh-php73-php-devel	pcre2-devel
rh-php73-php-pspell	aspell
rh-python36-python-devel	scl-utils-build
rh-python36-python-sphinx	texlive-framed, texlive-threeparttable, texlive-titlesec, texlive-wrapfig

表 2.2. 在 Red Hat Enterprise Linux 6 中启用可选频道的软件包

Software Collection 中的软件包	可选频道的所需软件包
devtoolset-8-dyninst-testsuite	glibc-static
devtoolset-8-elfutils-devel	xz-devel
devtoolset-8-gcc-plugin-devel	gmp-devel, mpfr-devel
devtoolset-8-libatomic-devel	libatomic
devtoolset-8-libgccjit	mpfr

Software Collection 中的软件包	可选频道的所需软件包
python27-python-devel	scl-utils-build
rh-mariadb102-boost-devel	libcicu-devel
rh-mariadb102-mariadb-bench	perl-GD
rh-mongodb34-boost-devel	libcicu-devel
rh-perl524-perl-devel	gdbm-devel, systemtap-sdt-devel
rh-python36-python-devel	scl-utils-build

2.2. 安装 RED HAT SOFTWARE COLLECTIONS

Red Hat Software Collections 是 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进行安装、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红帽软件集合。有关如何将您的系统与适当的订阅相关联的详情，并可以访问 Red Hat Software Collections，请参阅 [第 2.1 节“获取 Red Hat Software Collections”](#)。

使用 Red Hat Software Collections 3.4 需要删除所有更早的版本，包括 Beta 版本。如果您已安装任何以前的 Red Hat Software Collections 3.4 版本，请从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，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，强烈建议您删除所有 Red Hat Software Collections 软件包，执行原位升级，更新 Red Hat Software Collections 存储库，然后再次安装 Software Collections 软件包。建议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 3.4 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...
```

将 *software_collection* 替换为您要安装的以空格分隔的 Software Collections 列表。例如，要安装 php54 和 rh-mariadb100，请以 root 用户身份键入：

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

这会为所选 **Software Collection** 安装主 **meta** 软件包，以及一组必要的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 的每个组件都附带多个可选软件包，它们默认没有安装。要列出部分特定 **Software Collection** 但不安装在您的系统中的所有软件包，在 **shell** 提示符后输入以下内容：

```
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-perl526-perl-CPAN** 和 **rh-perl526-perl-Archive-Tar**，请输入：

```
~]# yum install rh-perl526-perl-CPAN rh-perl526-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，要为 **rh-ruby25-ruby** 软件包安装调试信息，请输入：

```
~]# debuginfo-install rh-ruby25-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\*
```

将 *software_collection* 替换为您要卸载的 **Software Collection** 组件。

请注意，卸载由 **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，不应使用不同内容重建现有集合。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。原始集合必须在不更改的情况下使用。

有关构建 **Software Collections** 的详细信息，请参阅 [Red Hat Software Collections Packaging Guide](#)。

第 3 章 使用方法

本章介绍了重建和使用 Red Hat Software Collections 3.4 以及部署使用 Red Hat Software Collections 的应用程序所需的步骤。

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...
```

将 `software_collection` 替换为您要使用的空间分隔的 Software Collections 列表，并使用您要运行的命令替换 `software_collection`。例如，要执行存储在名为 `hello.pl` 的 Perl 程序以及 `perl526 Software Collection` 中的 Perl 解释器，请输入：

```
~]# scl enable rh-perl526 '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。有关随 Red Hat Software Collections 一起发布的 Software Collections 的完整列表，请参阅 [表 1.1 “Red Hat Software Collections 3.4 组件”](#)。

3.1.2. 使用 Software Collection 作为默认运行 Shell 会话

要在首选使用所选 Software Collection 中启动带有可执行文件的新 shell 会话，在 shell 提示符后输入以下内容：

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

将 `software_collection` 替换为您要使用的以空格分隔的 Software Collections 列表。例如，要启动一个带有 `python27` 和 `rh-postgresql10 Software Collections` 的新 shell 会话作为默认值，请输入：

```
~]# scl enable python27 rh-postgresql10 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-postgresql10
```

有关随 **Red Hat Software Collections** 一起发布的 **Software Collections** 的完整列表，请参阅 [表 1.1 “Red Hat Software Collections 3.4 组件”](#)。

3.1.3. 从 Software Collection 运行系统服务

从 **Red Hat Enterprise Linux 6** 中的 **Software Collection** 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，作为您要启动的服务的名称。

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 96` 软件集合启动 `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-postgresql96-postgresql start
Starting rh-postgresql96-postgresql service:          [ OK ]
~]# chkconfig rh-postgresql96-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](#)。有关随 **Red Hat Software Collections** 一起发布的 **Software Collections** 的完整列表，请参阅 [表 1.1 “Red Hat Software Collections 3.4 组件”](#)。

从 **Red Hat Enterprise Linux 7** 中的 **Software Collection** 运行系统服务

在 **Red Hat Enterprise Linux 7** 中，初始化脚本已被 `systemd` 服务单元文件替代，该文件以 `.service`

文件扩展结尾，并作为初始化脚本提供类似的目的。要在当前会话中启动服务，以 root 用户身份执行以下命令：

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

将 `software_collection` 替换为 `Software Collection` 和 `service_name` 的名称，作为您要启动的服务的名称。

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：

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

例如，要从 `rh-postgresql 10 Software Collection` 启动 `postgresql` 服务并在引导时启用它，以 root 用户身份键入：

```
~]# systemctl start rh-postgresql10-postgresql.service  
~]# systemctl enable rh-postgresql10-postgresql.service
```

有关如何在 Red Hat Enterprise Linux 7 中管理系统服务的更多信息，请参阅 [Red Hat Enterprise Linux 7 系统管理员指南](#)。有关随 Red Hat Software Collections 一起发布的 Software Collections 的完整列表，请参阅 [表 1.1 “Red Hat Software Collections 3.4 组件”](#)。

3.2. 从 SOFTWARE COLLECTION 访问 MAN PAGE

每个 Software Collection 均包含一个常规手册页，用于描述此组件的内容。每个手册页的名称都与组件相同，它位于 `/opt/rh` 目录中。

要读取 Software Collection 的 man page，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-mariadb102` 的手册页，请输入：

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

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之一来部署依赖于生产环境中的 Red Hat Software Collections 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。

有关如何手动安装单独的 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，请阅读 [Red Hat Software Collections Packaging Guide](#)。

3.4. RED HAT SOFTWARE COLLECTIONS 容器镜像

基于 Red Hat Software Collections 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 3.4:
提供了以下容器镜像

- `rhsc/devtoolset-9-toolchain-rhel7`
- `rhsc/devtoolset-9-perftools-rhel7`
- `rhsc/nodejs-12-rhel7`
- `rhsc/php-73-rhel7`
- `rhsc/nginx-116-rhel7`

- *rhsc/postgresql-12-rhel7*

- *rhsc/httpd-24-rhel7*

以下容器镜像基于 Red Hat Software Collections 3.3:

- *rhsc/mariadb-103-rhel7*

- *rhsc/redis-5-rhel7*

- *rhsc/ruby-26-rhel7*

- *rhsc/devtoolset-8-toolchain-rhel7*

- *rhsc/devtoolset-8-perftools-rhel7*

- *rhsc/varnish-6-rhel7*

以下容器镜像基于 Red Hat Software Collections 3.2 :

- *rhsc/mysql-80-rhel7*

- *rhsc/nginx-114-rhel7*

- *rhsc/php-72-rhel7*

- *rhsc/nodejs-10-rhel7*

以下容器镜像基于 Red Hat Software Collections 3.1:

- ***rhsc/devtoolset-7-toolchain-rhel7 (EOL)***
- ***rhsc/devtoolset-7-perftools-rhel7 (EOL)***
- ***rhsc/mongodb-36-rhel7***
- ***rhsc/perl-526-rhel7***
- ***rhsc/php-70-rhel7 (EOL)***
- ***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 (EOL)***
- ***rhsc/nodejs-8-rhel7 (EOL)***

- *rhsc1/php-71-rhel7 (EOL)*
- *rhsc1/postgresql-96-rhel7*
- *rhsc1/python-36-rhel7*

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

- *rhsc1/devtoolset-6-toolchain-rhel7 (EOL)*
- *rhsc1/devtoolset-6-perftools-rhel7 (EOL)*
- *rhsc1/nginx-110-rhel7*
- *rhsc1/nodejs-6-rhel7 (EOL)*
- *rhsc1/python-27-rhel7*
- *rhsc1/ruby-24-rhel7*
- *rhsc1/ror-50-rhel7*
- *rhsc1/thermostat-16-agent-rhel7 (EOL)*
- *rhsc1/thermostat-16-storage-rhel7 (EOL)*

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

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

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

- *rhsc/devtoolset-4-toolchain-rhel7 (EOL)*
- *rhsc/devtoolset-4-perftools-rhel7 (EOL)*
- *rhsc/mariadb-101-rhel7 (EOL)*
- *rhsc/nginx-18-rhel7 (EOL)*
- *rhsc/nodejs-4-rhel7 (EOL)*
- *rhsc/postgresql-95-rhel7 (EOL)*

- *rhsc/ror-42-rhel7 (EOL)*
- *rhsc/thermostat-1-agent-rhel7 (EOL)*
- *rhsc/varnish-4-rhel7 (EOL)*

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

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

- ***rhsc1/s2i-base-rhel7***

标记为"生命周期结束(EOL)"的镜像不再被支持。

第 4 章 独立 SOFTWARE COLLECTIONS 的具体信息

本章着重介绍了特定的 **Software Collections** 细节，并提供了有关这些组件的更多详情。

4.1. RED HAT DEVELOPER TOOLSET

Red Hat Developer Toolset 是为在 **Red Hat Enterprise Linux** 平台上工作的开发人员设计的。**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。

有关功能概述，请参阅 **Red Hat Developer Toolset** 发行注记的 **Features** 部分。

4.2. RUBY ON RAILS 5.0

Red Hat Software Collections 3.4 提供 **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` 组件一起被支持。

4.3. 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 4.4 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。将共享库用于使用 C++11 (或更新版本) 功能的 MongoDB C++ 驱动程序的二进制文件, 还必须使用 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。

4.4. 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。使用使用 C++11（或更新版本）功能的 MongoDB C++ 驱动程序的共享库进行二进制文件，还必须使用 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](#)。

MongoDB 3.4 on Red Hat Enterprise Linux 6

如果您使用 Red Hat Enterprise Linux 6，以下指令适用于您的系统。

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。

4.5. MAVEN

rh-maven35 Software Collection 仅适用于 Red Hat Enterprise Linux 7，提供了一个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-maven36 Collection**，请以 root 用户身份输入以下命令：

```
yum install rh-maven36
```

要启用此集合，在 shell 提示符后输入以下命令：

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

全局 Maven 设置（如远程存储库或镜像）可以通过编辑 `/opt/rh/rh-maven36/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>

4.6. 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。

4.7. 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，这是到数据库服务器的本地或远程连接所必需的。表 4.1 “语言和数据库间的互操作性” 使用包括特定数据库服务器连接器的语言运行时列出 `Software Collections`

- 支持 - 支持组合
- 否 - 不支持组合

表 4.1. 语言和数据库间的互操作性

语言（软件集合）	数据库				
	MariaDB	MongoDB	MySQL	PostgreSQL	Redis
rh-nodejs4	否	否	否	否	否
rh-nodejs6	否	否	否	否	否
rh-nodejs8	否	否	否	否	否
rh-nodejs10	否	否	否	否	否
rh-nodejs12	否	否	否	否	否
rh-perl520	是	否	是	是	否
rh-perl524	是	否	是	是	否
rh-perl526	是	否	是	是	否
rh-php56	是	是	是	是	否

语言 (软件集合)	数据库				
	MariaDB	MongoDB	MySQL	PostgreSQL	Redis
rh-php70	是	否	是	是	否
rh-php71	是	否	是	是	否
rh-php72	是	否	是	是	否
rh-php73	是	否	是	是	否
python27	是	是	是	是	否
rh-python34	否	是	否	是	否
rh-python35	是	是	是	是	否
rh-python36	是	是	是	是	否
rh-ror41	是	是	是	是	否
rh-ror42	是	是	是	是	否
rh-ror50	是	是	是	是	否
rh-ruby25	是	是	是	是	否
rh-ruby26	是	是	是	是	否

第 5 章 MIGRATION (迁移)

本章提供有关迁移到 Red Hat Software Collections 3.4 中包含的组件版本的信息。

5.1. 迁移到 MARIADB 10.3

`rh-mariadb103 Software Collection` 可用于 Red Hat Enterprise Linux 7, 其中包括 MariaDB 5.5 作为默认的 MySQL 实施。

`rh-mariadb103 Software Collection` 与来自核心系统的 `mysql` 或 `mariadb` 软件包没有冲突。除非安装了 `*-syspaths` 软件包 (请参见如下), 否则可以将 `rh-mariadb103 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。此外, 还可以安装 `rh-mariadb103 Software Collection`, 而 `rh-mariadb102 Collection` 仍已安装甚至正在运行。

`rh-mariadb103 Software Collection` 包括 `rh-mariadb103-syspaths` 软件包, 它安装为二进制文件、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-mariadb103*-syspaths` 软件包后, 用户不需要使用 `scl enable` 命令对 `rh-mariadb103*` 软件包提供的二进制文件和脚本正常工作。请注意, `*-syspaths` 软件包与基础 Red Hat Enterprise Linux 系统以及 `rh-mariadb102` 和 `rh-mysql80 Software Collections` 的对应软件包冲突。要了解更多信息, 请参阅 [Red Hat Software Collections Packaging Guide](#)。

推荐的从 MariaDB 5.5 到 MariaDB 10.3 的迁移路径是首先升级到 MariaDB 10.0, 然后使用一个版本连续升级。详情请参阅早期的 Red Hat Software Collections 发行注记: [Migrating to MariaDB 10.0](#), [Migrating to MariaDB 10.1](#), 和 [Migrating to MariaDB 10.2](#)。



注意

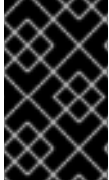
`rh-mariadb103 Software Collection` 支持使用 `scl register` 命令通过 NFS 挂载, 也不支持使用 `scl register` 命令进行动态注册。

5.1.1. `rh-mariadb102` 和 `rh-mariadb103 Software Collections` 之间的显著差异

- `mariadb-bench` 子软件包已被删除。
- 默认允许插件成熟度等级已改为比服务器成熟度低一个等级。因此, 在之前使用的, 但成熟度较低的插件将不再加载。

有关 MariaDB 10.3 的更多信息，请参阅上游文档有关 [更改](#) 和有关 [升级](#) 的信息。

5.1.2. 从 rh-mariadb102 升级到 rh-mariadb103 Software Collection



重要

在升级前，备份所有数据，包括任何 MariaDB 数据库。

1.

如果 rh-mariadb102 数据库服务器仍在运行，请停止它。

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。在 [上游文档](#) 中了解更多有关此选项的信息。这个操作的时间可能比正常关闭的时间要长。

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

停止 rh-mariadb102 服务器。

```
systemctl stop rh-mariadb102-mariadb.service
```

2.

安装 rh-mariadb103 Software Collection，包括提供 `mysql_upgrade` 工具的子软件包。

```
yum install rh-mariadb103-mariadb-server rh-mariadb103-mariadb-server-utils
```

请注意，可以在仍然安装 rh-mariadb102 Software Collection 时安装 rh-mariadb103 Software Collection，因为这些集合不会冲突。

3.

检查 rh-mariadb103 的配置，它存储在 `/etc/opt/rh/rh-mariadb103/my.cnf` 文件中，以及 `/etc/opt/rh/rh-mariadb103/my.cnf.d/` 目录中。将其与存储在 `/etc/opt/rh/rh-mariadb102/my.cnf` 和 `/etc/opt/rh/rh-mariadb102/my.cnf.d/` 中的 rh-mariadb102 配置进行比较，并根据需要进行调整。

4.

rh-mariadb102 Software Collection 的所有数据存储在 `/var/opt/rh/rh-`

`mariadb102/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-mariadb103/lib/mysql/`。您可以移动内容，但要记住在继续升级前备份数据。确保数据归 `mysql` 用户和 SELinux 上下文所有。

5. 启动 `rh-mariadb103` 数据库服务器。

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

6. 执行数据迁移。请注意，由于在 [MDEV-14637](#) 中引入上游更改，需要运行 `mysql_upgrade` 命令。

```
scl enable rh-mariadb103 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-mariadb103 -- mysql_upgrade -p
```

请注意，当安装 `rh-mariadb103*-syspaths` 软件包时，不需要 `scl enable` 命令。但是，`*-syspaths` 软件包与基础 Red Hat Enterprise Linux 系统以及 `rh-mariadb102` 和 `rh-mysql80` Software Collections 的对应软件包冲突。

5.2. 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 3.4 带有 MariaDB 10.2。

`rh-mariadb102` Software Collection 用于 Red Hat Enterprise Linux 6 和 Red Hat Enterprise Linux 7，与来自核心系统的 `mysql` 或 `mariadb` 软件包不会冲突。除非安装了 `*-syspaths` 软件包（请参阅以下），否则可以将 `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。此外，还可以安装 `rh-mariadb102` Software Collection，而 `rh-mariadb101` Collection 仍已安装甚至正在运行。

推荐的从 MariaDB 5.5 到 MariaDB 10.3 的迁移路径是首先升级到 MariaDB 10.0，然后使用一个版本连续升级。详情请参阅早期的 Red Hat Software Collections 发行注记：[Migrating to MariaDB 10.0](#) 和

[Migrating to MariaDB 10.1](#) 中的说明。

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 `scl register` 命令通过 NFS 挂载，也不支持使用 `scl register` 命令进行动态注册。

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

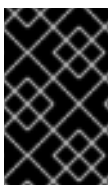
MariaDB 10.2 的主要变化在 [Red Hat Software Collections 3.0 发行注记](#) 中进行了描述。

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](#)。

多个选项更改了其默认值，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，它将安装为二进制文件、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** 软件包与来自基础 Red Hat Enterprise Linux 系统和 **rh-mysql80 Software Collection** 的对应软件包冲突。要了解更多有关 **syspaths** 的信息，请参阅 [Red Hat Software Collections Packaging Guide](#)。

5.2.2. 从 rh-mariadb101 升级到 rh-mariadb102 Software Collection



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，请停止它。

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。在 [上游文档](#) 中了解更多有关此选项的信息。这个操作的时间可

能比正常关闭的时间要长。

```
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，可以在仍然安装 `rh-mariadb101 Software Collection` 时安装 `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/`。您可以移动内容，但要记住在继续升级前备份数据。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
```

■

请注意，当安装 `rh-mariadb102*-syspaths` 软件包时，不需要 `scl enable` 命令。但是，`*-syspaths` 软件包与来自基础 Red Hat Enterprise Linux 系统和 `rh-mysql80 Software Collection` 的对应软件包冲突。

5.3. 迁移到 MySQL 8.0

`rh-mysql80 Software Collection` 适用于 Red Hat Enterprise Linux 7，其中包括 MariaDB 5.5 作为默认的 MySQL 实施。

`rh-mysql80 Software Collection` 不与来自核心系统的 `mysql` 或 `mariadb` 软件包以及 `rh-mysql*` 或 `rh-mariadb*` Software Collections 冲突，除非安装了 `*-syspaths` 软件包（请参阅以下）。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 5.7 升级到 MySQL 8.0。如果您需要从早期版本升级，请首先升级到 MySQL 5.7。具体步骤请参阅 [迁移到 MySQL 5.7](#)。

5.3.1. MySQL 5.7 和 MySQL 8.0 之间的显著差异

特定于 `rh-mysql80 Software Collection` 的不同

- `rh-mysql80 Software Collection` 提供的 MySQL 8.0 服务器被配置为使用 `mysql_native_password` 作为默认身份验证插件，因为 Red Hat Enterprise Linux 7 中的客户端工具和库与 `caching_sha2_password` 方法不兼容，后者在上游 MySQL 8.0 版本中默认使用。

要将默认身份验证插件更改为 `caching_sha2_password`，请编辑 `/etc/opt/rh/rh-mysql80/my.cnf.d/mysql-default-authentication-plugin.cnf` 文件：

```
[mysqld]
default_authentication_plugin=caching_sha2_password
```

有关 `caching_sha2_password` 身份验证插件的更多信息，[请参阅上游文档](#)。

- `rh-mysql80 Software Collection` 包含 `rh-mysql80-syspaths` 软件包，它将安装 `rh-mysql80-mysql-config-syspaths`、`rh-mysql80-mysql-syspaths`，和 `rh-mysql80-mysql-syspaths` 软件包。这些子软件包为二进制文件、脚本、手册页和其他提供系统范围的打包程序。安装 `rh-mysql80*-syspaths` 软件包后，用户不需要使用 `scl enable` 命令对 `rh-mysql80*` 软件包

提供的二进制文件和脚本正常工作。请注意，*-syspaths 软件包与基础 Red Hat Enterprise Linux 系统以及 rh-mariadb102 和 rh-mariadb103 Software Collections 的对应软件包冲突。要了解更多有关 syspaths 的信息，请参阅 [Red Hat Software Collections Packaging Guide](#)。

MySQL 8.0中的常规更改

- 在服务器启动期间，默认启用二进制日志记录。现在，log_bin 系统变量默认被设置为 ON，即使没有指定 --log-bin 选项。要禁用二进制日志记录，请在启动时指定 --skip-log-bin 或 --disable-log-bin 选项。
- 要接受 CREATE FUNCTION 语句，必须至少指定 DETERMINISTIC、NO SQL 或 READS SQL DATA 关键字，否则必须明确指定错误。
- 与帐户管理相关的某些功能已被删除。即，使用 GRANT 语句修改特权分配以外的帐户属性，如身份验证、SSL 和 resource-limit 等。要在创建 account-creation 时建立上述属性，请使用 CREATE USER 语句。要修改这些属性，请使用 ALTER USER 语句。
- 客户端中删除了某些与 SSL 相关的选项。使用 --ssl-mode=REQUIRED 选项，而不是 --ssl=1 或 --enable-ssl。使用 --ssl-mode=DISABLED 选项，而不是 --ssl=0、--skip-ssl 或 --disable-ssl。使用 --ssl-mode=VERIFY_IDENTITY 选项而不是 --ssl-verify-server-cert 选项。请注意，这些选项在服务器端保持不变。
- 默认字符集已从 latin1 改为 utf8mb4。
- utf8 字符集目前是 utf8mb3 的别名，但将来它将成为 utf8mb4 的引用。要防止不确定性，请为字符集引用（而不是 utf8）明确指定 utf8 mb4。
- 在 SET 以外的声明中设置用户变量已被弃用。
- log_syslog 变量之前配置了到系统日志的错误日志记录，它已被删除。
- 引入了对空间数据支持的某些不兼容的更改。
- GROUP BY 子句的已弃用的 ASC 或 DESC 限定符已被删除。要生成指定的排序顺序，请提供 ORDER BY 子句。

有关 MySQL 8.0 与早期版本相比的详细更改，请参阅上游文档：[MySQL 8.0 中的新内容](#)，以及对 [MySQL 8.0 的升级更改](#)。

5.3.2. 升级到 rh-mysql80 Software Collection



重要

在升级前，备份所有数据，包括任何 MySQL 数据库。

1. 安装 rh-mysql80 Software Collection。

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

2. 检查 rh-mysql80 的配置，它存储在 `/etc/opt/rh/rh-mysql80/my.cnf` 文件中，以及 `/etc/opt/rh/rh-mysql80/my.cnf.d/` 目录中。将它与存储在 `/etc/opt/rh/rh-mysql57/my.cnf` 和 `/etc/opt/rh/rh-mysql57/my.cnf.d/` 中的 `rh-mysql57/my.cnf.d/` 的配置进行比较，并在需要时进行调整。

3. 如果 rh-mysql57 数据库服务器仍在运行，请停止它。

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

4. rh-mysql57 Software Collection 的所有数据存储在 `/var/opt/rh/rh-mysql57/lib/mysql/` 目录中。将该目录的整个内容复制到 `/var/opt/rh/rh-mysql80/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-mysql80 数据库服务器。

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

6. 执行数据迁移。

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

请注意，当安装 `rh-mysql80*-syspaths` 软件包时，不需要 `scl enable` 命令。但是，`*-syspaths` 软件包与基础 Red Hat Enterprise Linux 系统以及 `rh-mariadb102` 和 `rh-mariadb103 Software Collections` 的对应软件包冲突。

5.4. 迁移到 MONGODB 3.6

Red Hat Software Collections 3.4 带有 MongoDB 3.6，由 `rh-mongodb36 Software Collection` 提供，仅适用于 Red Hat Enterprise Linux 7。

`rh-mongodb36 Software Collection` 包括 `rh-mongodb36-syspaths` 软件包，它安装为二进制文件、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。要了解更多信息，请参阅 [Red Hat Software Collections Packaging Guide](#)。

5.4.1. 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 复制已被弃用**

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**
- **块了解第二个工具**
- **查看使用版本 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](#)。

5.4.2. 从 rh-mongodb34 升级到 rh-mongodb36 Software Collection



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，建议首先运行部署，而不为刻录的周期启用后不兼容的功能，以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 `Sharded` 集群的详情，请参考上游 [MongoDB Manual](#)。

5.5. 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。

5.5.1. MongoDB 3.2 和 MongoDB 3.4 之间的显著差异

常规更改

`rh-mongodb34 Software Collection` 引进了各种常规更改。知识库文章 [Migrating from MongoDB 3.2 to MongoDB 3.4](#) 一书中列出了主要变化。有关详细更改，请参阅 [上游发行注记](#)。

此外，这个 `Software Collection` 中包含 `rh-mongodb34-syspaths` 软件包，它安装为二进制文件、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。要了解更多信息有关 `syspaths` 的信息，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/>

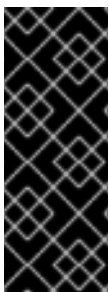
值得注意的是，以下 MongoDB 3.4 功能会向后兼容，需要使用 `featureCompatibilityVersion` 命令将版本设置为 3.4：

- 支持从现有集合或其他视图中创建只读视图
- 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](#)。

5.5.2. 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。您只能降级为 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, 而不先启用这些功能。

有关升级的详情, 请查看 [上游发行注记](#)。

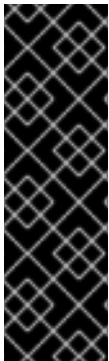
有关升级 **Replica Set** 的详情，请参考上游 [MongoDB Manual](#)。

有关升级 **Sharded 集群** 的详情，请参考上游 [MongoDB Manual](#)。

5.6. 迁移到 PostgreSQL 12

Red Hat Software Collections 3.4 带有 PostgreSQL 12，仅适用于 Red Hat Enterprise Linux 7。 `rh-postgresql12` Software Collection 可以与 PostgreSQL 的基本或任何 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。有关如何迁移到较早版本或使用 Red Hat Enterprise Linux 6 时，请参阅第 5.7 节“迁移到 PostgreSQL 9.6”。

`rh-postgresql12` Software Collection 包括 `rh-postgresql12-syspaths` 软件包，它安装为二进制文件、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-postgresql12*-syspaths` 软件包后，用户不需要使用 `scl enable` 命令对 `rh-postgresql12*` 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。要了解更多有关 `syspaths` 的信息，请参阅 [Red Hat Software Collections Packaging Guide](#)。



重要

在迁移到 PostgreSQL 12 之前，请查看 [PostgreSQL 11](#) 和 [PostgreSQL 12](#) 的上游兼容性备注。

如果在容器中升级 PostgreSQL 数据库，请参阅 [特定于容器的说明](#)。

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 10` 和 `rh-postgresql12` Software Collections 中的不同路径的概述。

表 5.1. PostgreSQL 路径中的 Differences

内容	postgresql	rh-postgresql10	rh-postgresql12
可执行文件	/usr/bin/	/opt/rh/rh-postgresql10/root/usr/bin/ /	/opt/rh/rh-postgresql12/root/usr/bin/
libraries	/usr/lib64/	/opt/rh/rh-postgresql10/root/usr/lib64/	/opt/rh/rh-postgresql12/root/usr/lib64/

内容	postgresql	rh-postgresql10	rh-postgresql12
Documentation	/usr/share/doc/postgresql/html/	/opt/rh/rh-postgresql10/root/usr/share/doc/postgresql/html/	/opt/rh/rh-postgresql12/root/usr/share/doc/postgresql/html/
PDF 文档	/usr/share/doc/postgresql-docs/	/opt/rh/rh-postgresql10/root/usr/share/doc/postgresql-docs/	/opt/rh/rh-postgresql12/root/usr/share/doc/postgresql-docs/
contrib 文档	/usr/share/doc/postgresql-contrib/	/opt/rh/rh-postgresql10/root/usr/share/doc/postgresql-contrib/	/opt/rh/rh-postgresql12/root/usr/share/doc/postgresql-contrib/
源	未安装	未安装	未安装
data	/var/lib/pgsql/data/	/var/opt/rh/rh-postgresql10/lib/pgsql/data/	/var/opt/rh/rh-postgresql12/lib/pgsql/data/
备份区域	/var/lib/pgsql/backups/	/var/opt/rh/rh-postgresql10/lib/pgsql/backups/	/var/opt/rh/rh-postgresql12/lib/pgsql/backups/
模板	/usr/share/pgsql/	/opt/rh/rh-postgresql10/root/usr/share/pgsql/	/opt/rh/rh-postgresql12/root/usr/share/pgsql/
流程语言	/usr/lib64/pgsql/	/opt/rh/rh-postgresql10/root/usr/lib64/pgsql/	/opt/rh/rh-postgresql12/root/usr/lib64/pgsql/
开发标头	/usr/include/pgsql/	/opt/rh/rh-postgresql10/root/usr/include/pgsql/	/opt/rh/rh-postgresql12/root/usr/include/pgsql/
其他共享数据	/usr/share/pgsql/	/opt/rh/rh-postgresql10/root/usr/share/pgsql/	/opt/rh/rh-postgresql12/root/usr/share/pgsql/
回归测试	/usr/lib64/pgsql/test/regress/ (in the -test package)	/opt/rh/rh-postgresql10/root/usr/lib64/pgsql/test/regress/ (in the -test package)	/opt/rh/rh-postgresql12/root/usr/lib64/pgsql/test/regress/ (in the -test package)

5.6.1. 从 Red Hat Enterprise Linux System 版本的 PostgreSQL 迁移到 PostgreSQL 12 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-postgresql12 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，第二个方法通常非常慢，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 12 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

要执行 PostgreSQL 服务器的快速升级，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
```

要验证服务器是否正在运行，请键入：

```
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-postgresql12/lib/pgsql/data/` 不存在：

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

如果您正在运行一个全新的 PostgreSQL 12 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-postgresql12/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-postgresql12 -- postgresql-setup --upgrade
```

或者，您可以使用 `/opt/rh/rh-postgresql12/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` 选项获得可能的升级场景列表。

建议您读取生成的 `/var/lib/pgsql/upgrade_rh-postgresql12-postgresql.log` 日志文件，以找出升级过程中出现任何问题。

5.

以 root 用户身份启动新服务器：

```
systemctl start rh-postgresql12-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-postgresql12 ~/analyze_new_cluster.sh'
```

6.

另外，您可以将 PostgreSQL 12 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 12 服务器，以 root 用户身份键入：

```
chkconfig rh-postgresql12-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-postgresql12/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 5.2. 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 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-postgresql12 -- postgresql-setup initdb
```

5. 以 root 用户身份启动新服务器：

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

6. 从之前创建的 SQL 文件中导入数据：

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

7. 另外，您可以将 PostgreSQL 12 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 12 服务器，以 root 用户身份键入：

-

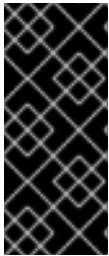

```
chkconfig rh-postgresql12-postgresql on
```

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-postgresql12/lib/pgsql/data/pg_hba.conf` 配置文件。否则，只有 `postgres` 用户才能访问数据库。

5.6.2. 从 PostgreSQL 10 Software Collection 迁移到 PostgreSQL 12 Software Collection

要将数据从 `rh-postgresql10 Software Collection` 迁移到 `rh-postgresql12 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，第二个方法通常非常慢，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 10 迁移到 PostgreSQL 12 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-postgresql10/lib/pgsql/data/` 目录中。

过程 5.3. 使用 `pg_upgrade` 工具快速升级

要执行 PostgreSQL 服务器的快速升级，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-postgresql10-postgresql.service
```

要验证服务器是否正在运行，请键入：

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

2.

验证旧目录 `/var/opt/rh/rh-postgresql10/lib/pgsql/data/` 是否存在：

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

并备份您的数据。

3.

验证新数据目录 `/var/opt/rh/rh-postgresql12/lib/pgsql/data/` 不存在：

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

如果您正在运行一个全新的 PostgreSQL 12 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-postgresql12/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-postgresql12 -- postgresql-setup --upgrade --upgrade-from=rh-postgresql10-postgresql
```

另外，您可以使用 `/opt/rh/rh-postgresql12/root/usr/bin/postgresql-setup --upgrade --upgrade-from=rh-postgresql10-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` 选项获得可能的升级场景列表。

建议您读取生成的 `/var/lib/pgsql/upgrade_rh-postgresql12-postgresql.log` 日志文件，以找出升级过程中出现任何问题。

5.

以 `root` 用户身份启动新服务器：

```
systemctl start rh-postgresql12-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-postgresql12 ~/analyze_new_cluster.sh'
```

6.

另外，您可以将 PostgreSQL 12 服务器配置为在引导时自动启动。要禁用旧的 PostgreSQL 10 服务器，请以 `root` 用户身份输入以下命令：

```
chkconfig rh-postgresql10-postgresql off
```

要启用 PostgreSQL 12 服务器，以 root 用户身份键入：

```
chkconfig rh-postgresql12-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-postgresql12/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-postgresql10-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-postgresql10 "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-postgresql10-postgresql.service
```

4.

以 root 用户身份初始化新服务器的数据目录：

```
scl enable rh-postgresql12 -- postgresql-setup initdb
```

5.

以 root 用户身份启动新服务器：

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

6.

从之前创建的 SQL 文件中导入数据：

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

7.

另外，您可以将 PostgreSQL 12 服务器配置为在引导时自动启动。要禁用旧的 PostgreSQL 10 服务器，请以 root 用户身份输入以下命令：

```
chkconfig rh-postgresql10-postgresql off
```

要启用 PostgreSQL 12 服务器，以 root 用户身份键入：

```
chkconfig rh-postgresql12-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-postgresql12/lib/pgsql/data/pg_hba.conf` 配置文件。否则，只有 postgres 用户才能访问数据库。

5.7. 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。



重要

如果在容器中升级 PostgreSQL 数据库，请参阅 [特定于容器的说明](#)。请注意，目前无法在 OpenShift 环境中通过 Gluster 文件卷将 PostgreSQL 从 9.5 升级到 9.6。

5.7.1. 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，它安装为二进制文件、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。要了解更多有关 `syspaths` 的信息，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/
libraries	/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/
源	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/

内容	postgresql	postgresql92	rh-postgresql95	rh-postgresql96
开发标头	/usr/include/postgresql/	/opt/rh/postgresql92/root/usr/include/postgresql/	/opt/rh/rh-postgresql95/root/usr/include/postgresql/	/opt/rh/rh-postgresql96/root/usr/include/postgresql/
其他共享数据	/usr/share/postgresql/	/opt/rh/postgresql92/root/usr/share/postgresql/	/opt/rh/rh-postgresql95/root/usr/share/postgresql/	/opt/rh/rh-postgresql96/root/usr/share/postgresql/
回归测试	/usr/lib64/postgresql/test/regress/ (in the -test package)	/opt/rh/postgresql92/root/usr/lib64/postgresql/test/regress/ (in the -test package)	/opt/rh/rh-postgresql95/root/usr/lib64/postgresql/test/regress/ (in the -test package)	/opt/rh/rh-postgresql96/root/usr/lib64/postgresql/test/regress/ (in the -test package)

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](#)。

5.7.2. 从 Red Hat Enterprise Linux System 版本的 PostgreSQL 迁移到 PostgreSQL 9.6 Software Collection

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，第二个方法通常非常慢，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/pgsql/data/` 目录中。

过程 5.5. 使用 `pg_upgrade` 工具快速升级

要执行 PostgreSQL 服务器的快速升级，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
```

要验证服务器是否正在运行，请键入：

```
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` 选项获得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 文件中导入数据：

```
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
```

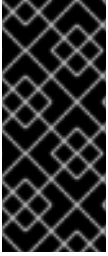
要启用 PostgreSQL 9.6 服务器，以 root 用户身份输入：

```
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。

5.7.3. 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，第二个方法通常非常慢，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

要执行 PostgreSQL 服务器的快速升级，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
```

要验证服务器是否正在运行，请键入：

```
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` 选项获得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 文件中导入数据：

```
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。

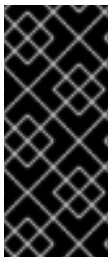
如果您需要从 postgresql92 Software Collection 迁移，请参阅 [Red Hat Software Collections 2.0 发行注记](#)。该流程是相同的，您只需要调整新集合的版本。同样适用于从 rh-postgresql94 Software Collection 迁移，这在 [Red Hat Software Collections 2.2 发行注记](#) 中进行了描述。

5.8. 迁移到 NGINX 1.16

rh-nginx116 Software Collection 的根目录位于 `/opt/rh/rh-nginx116/root/` 中。默认情况下，错误日志存储在 `/var/opt/rh/rh-nginx116/log/nginx` 中。

配置文件存储在 `/etc/opt/rh/rh-nginx116/nginx/` 目录中。nginx 1.16 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-nginx116/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 1.14 升级到 nginx 1.16 之前，请备份您的所有数据，包括位于 `/opt/rh/nginx114/root/ tree` 中的 web 页面，以及位于 `/etc/opt/rh/nginx114/nginx/` 树的配置文件。

如果您已进行任何特定的更改，如更改配置文件或设置 Web 应用程序（在 `/opt/rh/nginx114/root/` 树中），在新的 `/opt/rh/rh-nginx116/root/` 目录中复制这些更改，以及 `/etc/opt/rh/rh-nginx116/nginx/` 目录中。

您可以使用此流程直接从 nginx 1.8、nginx 1.10、nginx 1.12 或 nginx 1.14 升级到 nginx 1.16。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/>。

5.9. 迁移到 REDIS 5

Redis 3.2 由 rh-redis32 Software Collection 提供，主要是 Redis 4.0 的严格子集，这主要是 Redis 5.0 的一个严格子集。因此，从 3.2 升级到 5.0 时不应发生重大问题。

要将 Redis Cluster 升级到 5.0 版本，需要重启所有实例。

兼容性备注

- **RDB 文件格式已改变。Redis 5 能够读取所有早期版本的格式，但较早的版本能够读取 Redis 5 格式。**
- **自版本 4.0 起，Red Hat Redis Cluster 总线协议不再与 Redis 3.2 兼容。**
- **有关次版本的非后向性更改，请参阅上游版本 [4.0](#) 和版本 [5.0](#)。**

第 6 章 其它资源

本章提供了关于 Red Hat Software Collections 3.4 和 Red Hat Enterprise Linux 的其他相关信息参考。

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 3.4 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 9.0 发行注记](#) - Red Hat Developer Toolset 发行注记 包括了已知问题、可能的问题、更改以及有关这个 Software Collection 的其他重要信息。
- [Red Hat Developer Toolset 9.0 用户指南](#) - Red Hat Developer Toolset 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](#) - 本指南包含有关在 Red Hat Enterprise Linux 7 和 Red Hat Enterprise Linux Atomic Host 上构建和使用容器镜像的全面概述。
- [使用和配置红帽订阅管理器](#) - 使用和配置 红帽订阅管理器 书提供了有关如何注册 Red Hat Enterprise Linux 系统、管理订阅和查看已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该系统的部署、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 的系统管理员指南 提供了有关该系统的部署、配置和管理的信息。

6.2. RED HAT DEVELOPERS

- [Red Hat Developer Program](#) - 红帽开发人员社区 门户。
- [Red Hat Developers 上的 Red Hat Software Collections 概述](#) - Red Hat Developers Portal 提供了多个教程来帮助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 Developer Blog](#) - Red Hat Developer Blog 包含最新的信息、最佳实践、建议、产品和程序公告，以及基于红帽技术设计和开发应用程序的指针。

附录 A. 修订历史记录

修订 3.4-4 修复了有问题的链接和定位符。	Wed Dec 20 2023	Lenka Špačková
修订 3.4-3 更新了 第 4.7 节“数据库连接器” 。	Fri Nov 12 2021	Lenka Špačková
修订 3.4-2 添加了对 PostgreSQL 的特定容器升级说明的引用。	Tue Mar 17 2020	Lenka Špačková
修订 3.4-1 发布 Red Hat Software Collections 3.4 发行注记。	Tue Dec 10 2019	Lenka Špačková
修订 3.4-0 发布 Red Hat Software Collections 3.4 Beta 发行注记。	Thu Nov 07 2019	Lenka Špačková