



# Red Hat Enterprise Linux 8

## 安装、管理和删除用户空间组件

使用 YUM 软件管理工具管理 BaseOS 和 AppStream 存储库中的内容



# Red Hat Enterprise Linux 8 安装、管理和删除用户空间组件

---

使用 YUM 软件管理工具管理 BaseOS 和 AppStream 存储库中的内容

## 法律通告

Copyright © 2024 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

使用 YUM 工具查找、安装和使用通过 BaseOS 和 AppStream 存储库分发的内容。了解如何使用软件包、模块、流和配置文件。

---

# 目录

对红帽文档提供反馈 .....	3
<b>第 1 章 使用 APPSTREAM .....</b>	<b>4</b>
1.1. RHEL 8 中的内容发布 .....	4
1.2. 应用程序流 .....	4
1.3. RHEL 8 中的打包方法 .....	4
1.4. 在 RHEL 8 中使用 YUM 进行软件包管理 .....	5
<b>第 2 章 模块简介 .....</b>	<b>6</b>
2.1. 模块流 .....	6
2.2. 模块配置集 .....	7
<b>第 3 章 查找 RHEL 8 内容 .....</b>	<b>8</b>
3.1. 搜索软件包 .....	8
3.2. 列出可用的模块及其内容 .....	8
3.3. 列出内容的命令 .....	12
<b>第 4 章 安装 RHEL 8 内容 .....</b>	<b>13</b>
4.1. 安装软件包 .....	13
4.2. 安装软件包前选择流 .....	13
4.3. 安装模块化内容 .....	14
4.4. 运行安装的内容 .....	16
4.5. 安装 RHEL 8 内容的命令 .....	16
4.6. 其他资源 .....	17
<b>第 5 章 删除 RHEL 8 内容 .....</b>	<b>18</b>
5.1. 删除安装的软件包 .....	18
5.2. 删除安装的模块内容 .....	18
5.3. 删除内容的命令 .....	24
<b>第 6 章 管理 APPLICATION STREAM 内容的版本 .....</b>	<b>26</b>
6.1. 模块依赖关系和流更改 .....	26
6.2. 模块化和非模块化依赖项的交互 .....	26
6.3. 重置模块流 .....	27
6.4. 禁用一个模块的所有流 .....	27
6.5. 切换到更新的流 .....	27
6.6. 覆盖模块默认流 .....	28



---

## 对红帽文档提供反馈

我们感谢您对我们文档的反馈。让我们了解如何改进它。

### 通过 Jira 提交反馈（需要帐户）

1. 登录到 [Jira](#) 网站。
2. 单击顶部导航栏中的 **Create**。
3. 在 **Summary** 字段中输入描述性标题。
4. 在 **Description** 字段中输入您的建议以改进。包括文档相关部分的链接。
5. 点对话框底部的 **Create**。

# 第 1 章 使用 APPSTREAM

在以下部分中，了解与 Red Hat Enterprise Linux 8 中 AppStream 存储库相关的概念：

- [RHEL 8 中内容的分发](#)。
- [应用程序流](#)。
- [RHEL 8 中的打包方法](#)。
- [在 RHEL 8 中使用 YUM 的软件包管理](#)。

## 1.1. RHEL 8 中的内容发布

Red Hat Enterprise Linux 8 内容是通过两个主要存储库分发的：**BaseOS** 和 **AppStream**。

### BaseOS

BaseOS 存储库包含为所有安装提供基础的底层操作系统功能的核心集合。此内容以 RPM 软件包的形式提供，遵守与早期 Red Hat Enterprise Linux 版本中相似的支持条款。

### AppStream

AppStream 存储库包含额外的用户空间应用程序、运行时语言和数据库，以支持各种工作负载和用例。AppStream 的内容以两种格式提供 - RPM 软件包和名为 *模块* 的 RPM 格式的扩展。



#### 重要

基本 RHEL 安装需要 BaseOS 和 AppStream 内容集，所有 RHEL 订阅都可使用它。有关安装说明，请参阅 [从安装介质主动安装 RHEL](#) 文档。

## 1.2. 应用程序流

Red Hat Enterprise Linux 8 引进了 Application Streams - 用户空间组件的版本。现在，会比核心操作系统软件包频繁交付和更新这些组件的多个版本。这为自定义 Red Hat Enterprise Linux 提供了更大的灵活性，不会影响平台或特定部署的基本稳定性。

作为 Application Streams 提供的组件可打包为模块 (module) 或 RPM 软件包，并通过 RHEL 8 中的 AppStream 软件仓库提供。每个 Application Stream 都有一个生命周期，它与 RHEL 8 的生命周期相同或更短。[Red Hat Enterprise Linux 8 Application Streams 生命周期页](#) 中列出了较短的生命周期。



#### 注意

不是所有模块都是 Application Streams。其他模块的相依性不被视为 Application Streams。

### 其它资源

- [Red Hat Enterprise Linux 生命周期](#)
- [Red Hat Enterprise Linux 8 Application Streams 生命周期](#)

## 1.3. RHEL 8 中的打包方法

AppStream 存储库中的内容以两种方式打包：



- **单个 RPM 软件包**  
可以用来立即安装的传统 RPM 软件包。
- **模块**  
模块是代表逻辑单元的软件包集合：应用程序、语言堆栈、数据库或一组工具。这些软件包被一同构建、测试并发布。

## 1.4. 在 RHEL 8 中使用 YUM 进行软件包管理

YUM 软件包管理工具现在基于 DNF 技术，它增加了对新模块化特性的支持。

在处理单个 RPM 软件包时，YUM 的使用并没有改变。为处理模块内容，添加了 **yum module** 命令。如需了解更多详细信息，请参阅 [安装 RHEL 8 内容](#)。

如果需要，模块化功能会自动选择模块和流组合，以启用逻辑软件包集合以方便使用。

## 第 2 章 模块简介

除了单独的 RPM 软件包，AppStream 存储库还包含模块。模块是一组代表组件且通常一起安装的 RPM 软件包。典型的模块包含应用的软件包、应用特定依赖项库的软件包、应用文档的软件包，以及帮助程序实用程序的软件包。

在以下部分中，了解组织和处理模块中内容的功能：

- [模块流](#) - 按版本组织内容。
- [模块配置文件](#) - 按目的组织内容。

### 2.1. 模块流

模块流是可以作为 AppStream 物理存储库中的虚拟存储库的过滤器。模块流代表 AppStream 组件的版本。每个流都单独接收更新。

模块流可以是活跃的或者不活跃的。有效流可让系统访问特定模块流中的 RPM 软件包，以便安装相应的组件版本。如果流标记为默认值，或者用户操作被显式启用，则流处于活跃状态。

给定时间点上只能激活一个特定模块的流。因此，系统中只能安装一个组件版本。不同的容器可以使用不同的版本。

每个模块都可以具有默认流。默认流可让您轻松使用 RHEL 软件包而无需了解模块。默认流是活跃的，除非禁用了整个模块或启用了该模块的另一个流。



#### 重要

默认流在整个 RHEL 主发行版本中不会改变。始终考虑每个流的 [生命周期](#)。不要依赖默认流，在 RHEL 主发行版本结束前，默认流达到生命周期结束状态。

某些模块流依赖于其他模块流。例如，`perl-App-cpanminus`、`perl-DBD-MySQL`、`perl-DBD-Pg`、`perl-DBD-SQLite`、`perl-DBI`、`perl-YAML` 和 `freeradius` 模块流取决于特定的 `perl` 模块流。

要为运行时用户应用程序或开发人员应用程序选择特定的流，请考虑以下几点：

- 所需的功能以及支持哪个组件版本
- 兼容性
- [生命周期](#) 长度和更新计划

有关所有可用模块和流的列表，请查看 [软件包清单](#)。有关每个组件的更改，请查看 [发行注记](#)。

#### 例 2.1. PostgreSQL 模块流

`postgresql` 模块在相应的流 9.6、10、12、13 和 15 中提供 PostgreSQL 数据库版本 9.6、10、12 和 13。流 10 是默认值。这意味着，如果要求 `postgresql`，系统会尝试安装 `postgresql-10.6` 软件包。

始终决定您要使用哪个模块流，并明确安装相应的版本。

#### 其他资源

- [模块依赖关系和流更改](#)

- [切换到更新的流](#)
- [软件包清单](#)
- [发行注记](#)

## 2.2. 模块配置集

配置文件是为特定用例一起安装的推荐的软件包列表，如服务器、客户端、开发、最小安装或其他。这些软件包列表可以包含模块流以外的软件包，通常是来自 BaseOS 存储库或流依赖项的软件包。

使用配置集安装软件包是为方便用户提供的一次性操作。它不会阻止安装或卸载该模块提供的任何软件包。您还可以使用同一模块流的多个配置集安装软件包，而无需进一步准备步骤。

每个模块流可以有任意数量的配置集，包括没有。对于任何给定模块流，其某些配置集可以被标记为 *default*，并在没有显式指定配置集时用于配置集安装操作。但是，不需要为模块流存在默认配置集。

### 例 2.2. httpd 模块配置集

提供 Apache web 服务器的 `httpd` 模块为安装提供以下配置集：

- **common** - 一个强化的生产环境就绪部署，它是默认的配置集。
- **devel** - 对 `httpd` 进行修改所需的软件包。
- **minimal** - 提供一个运行的 Web 服务器的最小软件包集合。

## 第 3 章 查找 RHEL 8 内容

在以下部分中，了解如何使用 YUM 在 Red Hat Enterprise Linux 8 中查找和检查 AppStream 和 BaseOS 存储库中的内容：

- [搜索提供所需内容的软件包。](#)
- [列出可用的模块并查找有关它们的详细信息。](#)
- [检查 RHEL 8 内容的有用命令。](#)

### 3.1. 搜索软件包

要查找提供特定应用程序或其他内容的软件包，请完成以下步骤。

#### 步骤

1. 使用文本字符串搜索软件包，如应用程序名称：

```
$ yum search "text string"
```

2. 查看软件包详情：

```
$ yum info package
```

### 3.2. 列出可用的模块及其内容

要找出哪些模块可用及其详情，请完成以下步骤。

#### 步骤

- 要列出您系统可用的模块流，请使用：

```
$ yum module list
```

此命令的输出列出了单独行中的名称、流、配置集和概述的模块流。

- 要显示模块详情，包括描述、所有配置文件的列表以及所有提供的软件包的列表，请使用：

```
$ yum module info module-name
```

- 要列出每个模块配置文件安装了这些软件包的哪一个，请使用：

```
$ yum module info --profile module-name
```

- 要显示模块的当前状态，包括启用的流和安装的配置文件，请使用：

```
$ yum module list module-name
```

#### 其他资源

- [模块简介](#)

### 例 3.1. 查找模块详情

以下是如何列出 AppStream 存储库中可用模块以及如何获取有关 **postgresql** 模块内容的信息的示例。



#### 注意

本例中的输出已编辑为 brevity。实际输出可能包含比这里显示的更多的信息。

#### 1. 列出可用模块：

```
$ yum module list
Name      Stream Profiles Summary
(...)
postgresql 9.6  client, PostgreSQL server and client module
          server [
          d]
postgresql 10 [d] client, PostgreSQL server and client module
          server [
          d]
postgresql 12  client, PostgreSQL server and client module
          server [
          d]
postgresql 13  client, PostgreSQL server and client module
          server [
          d]
postgresql 15  client, PostgreSQL server and client module
          server [
          d]
(...)

Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
```

#### 2. 检查 **postgresql** 模块的详情：

```
$ yum module info postgresql
...
Name      : postgresql
Stream    : 10 [d][a]
Version   : 8070020221124143148
Context   : bd1311ed
Architecture : x86_64
Profiles  : client, server [d]
Default profiles : server
Repo      : rhel-AppStream
Summary   : PostgreSQL server and client module
...

Name      : postgresql
Stream    : 12
Version   : 8060020221003080350
Context   : ad008a3a
Architecture : x86_64
```

```

Profiles      : client, server [d]
Default profiles : server
Repo          : rhel-AppStream
Summary       : PostgreSQL server and client module
...

Name          : postgresql
Stream        : 13
Version       : 8070020230227142544
Context       : bd1311ed
Architecture  : x86_64
Profiles      : client, server [d]
Default profiles : server
Repo          : rhel-AppStream
Summary       : PostgreSQL server and client module
...

Name          : postgresql
Stream        : 15
Version       : 8080020230212204728
Context       : fd72936b
Architecture  : x86_64
Profiles      : client, server [d]
Default profiles : server
Repo          : rhel-AppStream
Summary       : PostgreSQL server and client module
...

Name          : postgresql
Stream        : 9.6
Version       : 8040020210602182503
Context       : 522a0ee4
Architecture  : x86_64
Profiles      : client, server [d]
Default profiles : server
Repo          : rhel-AppStream
Summary       : PostgreSQL server and client module
...

Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled, [a]ctive

```

如果没有指定任何流，**yum** 会列出所有可用的流。

### 3. 检查 **postgresql** 模块流 **10** 中的配置集：

```

$ yum module info --profile postgresql:10
(...)
Name   : postgresql:10:8070020221124143148:bd1311ed:x86_64
client : postgresql
server : postgresql-server

```

请注意，每个配置文件会安装不同的软件包集合，包括它们的依赖项。

### 4. 使用默认流 **10** 和默认配置文件 **server** 安装 **postgresql** 模块：

```
# yum module install postgresql
```

```

...
Dependencies resolved.
=====
=====
Package           Architecture Version           Repository        Size
=====
Installing group/module packages:
postgresql-server x86_64          10.23-1.module+el8.7.0+17280+3a452e1f  rhel-
AppStream         5.1 M
Installing dependencies:
libpq             x86_64          13.5-1.el8                rhel-AppStream  198 k
postgresql        x86_64          10.23-1.module+el8.7.0+17280+3a452e1f  rhel-
AppStream         1.5 M
Installing module profiles:
  postgresql/server
Enabling module streams:
  postgresql          10

Transaction Summary
=====
=====
Install 3 Packages

Total download size: 6.7 M
Installed size: 26 M
Is this ok [y/N]: y
...

Installed:
  libpq-13.5-1.el8.x86_64
  postgresql-10.23-1.module+el8.7.0+17280+3a452e1f.x86_64
  postgresql-server-10.23-1.module+el8.7.0+17280+3a452e1f.x86_64

Complete!

```

##### 5. 检查 `postgresql` 模块的当前状态：

```

$ yum module list postgresql
rhel-AppStream
Name           Stream      Profiles          Summary
postgresql    9.6         client, server [d] PostgreSQL server and client
module
postgresql    10 [d][e]   client, server [d] [i] PostgreSQL server and client
module
postgresql    12          client, server [d] PostgreSQL server and client
module
postgresql    13          client, server [d] PostgreSQL server and client
module
postgresql    15          client, server [d] PostgreSQL server and client
module

Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled

```

输出显示默认流 **10** 已启用并安装了其配置集 **server**。

### 3.3. 列出内容的命令

以下是在 Red Hat Enterprise Linux 8 中查找内容及其详情的常用命令。

命令	描述
<b>yum list available</b>	列出可用的软件包。
<b>yum repoquery <i>package</i></b>	搜索所选软件包的可用的 YUM 存储库。
<b>yum search "<i>text string</i>"</b>	使用任意文本字符串搜索一个软件包。
<b>yum info <i>package</i></b>	显示软件包的详情。
<b>yum module provides <i>package</i></b>	显示哪个模块提供了软件包。 如果软件包不在任何模块之外，这个命令的输出就为空。
<b>yum module list</b>	列出可用的模块。
<b>yum module info <i>module-name</i></b>	显示模块的详细信息。
<b>yum module info --profile <i>module-name</i></b>	使用默认流列出由模块的配置文件安装的软件包。
<b>yum module info --profile <i>module-name:stream</i></b>	显示使用指定流由模块的配置文件安装的软件包。
<b>yum module list <i>module-name</i></b>	显示模块的当前状态。



## 第 4 章 安装 RHEL 8 内容

在以下部分中，了解如何在 Red Hat Enterprise Linux 8 中安装内容：

- [安装软件包](#)。
- [为软件包安装选择一个流](#)。
- [安装由模块、流和配置文件提供的软件包集合](#)。
- [运行 RHEL 8 安装的内容](#)。
- [检查安装 RHEL 8 内容的有用命令](#)。

### 4.1. 安装软件包

要安装软件包，请完成以下步骤。

#### 步骤

- 安装软件包：

```
# yum install package
```

使用软件包的名称替换 *package*。

- 如果任何模块流都没有提供软件包，这个流程与在早期版本的 Red Hat Enterprise Linux 上使用的流程一样。
- 如果软件包由启用的模块流提供，则安装该软件包时无需进一步操作。
- 如果软件包是由标记为默认模块流提供的，则 **yum** 会在安装此软件包前自动启用该模块流。



#### 重要

建议您始终选择特定的模块流进行安装，而不是依赖于默认流。在 RHEL 主发行版本结束前，某些默认模块流会到达生命周期结束。始终考虑每个流的 [生命周期](#)。

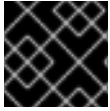
- 如果软件包由没有激活的模块流（不是上述情况）提供，则在手动启用相应的模块流前无法识别。

#### 其他资源

- [安装模块化内容](#)
- [在 RHEL 8 中使用 YUM 进行软件包管理](#)
- [Red Hat Enterprise Linux Application Streams Life Cycle](#)

### 4.2. 安装软件包前选择流

建议您始终选择特定的模块流进行安装。始终考虑每个流的 [生命周期](#)。



### 重要

在 RHEL 主发行版本结束前，某些默认模块流会到达生命周期结束。

要从非默认流中安装软件包，请首先启用流。

#### 先决条件

- 您了解了 [活动模块流的概念](#)。

#### 步骤

- 启用模块流：

```
# yum module enable module-name:stream
```

将 *module-name* 和 *stream* 替换为模块和流的名称。

**yum** 请求确认，且流已启用并激活。



### 注意

如果之前因为默认而激活的该模块的另一个流将不再活跃。

#### 其他资源

- [Red Hat Enterprise Linux Application Streams Life Cycle](#)

## 4.3. 安装模块化内容

要安装由模块流或配置文件提供的模块内容，请完成以下步骤。

#### 先决条件

- 您了解了 [活动模块流的概念](#)。
- 您没有任何从同一模块的另一个流安装的软件包。

#### 步骤

- 要安装所选模块流，请使用：

```
# yum module install module-name:stream
```

通过运行这个命令，您可以自动启用所选的流。请注意，如果已为流定义了默认配置文件，则此配置文件会被自动安装。



### 重要

始终考虑模块流的 [生命周期](#)。

- 要安装所选模块流的配置集，请使用：

■

```
# yum module install module-name:stream/profile
```

通过运行这个命令，您可以启用流，并为给定的流（版本）和模块的配置文件（用途）安装推荐的软件包组。

## 其他资源

- [模块简介](#)
- [安装 RHEL 8 内容的命令](#)
- [Red Hat Enterprise Linux Application Streams Life Cycle](#)

### 例 4.1. 安装应用程序的非默认流

以下是一个如何从非默认流（版本），即版本 **13** 中的 PostgreSQL 服务器(`postgresql-server` 软件包)安装应用程序的示例。默认流提供版本 **10**。

#### 步骤

1. 列出提供 `postgresql-server` 软件包的模块，以查看哪些流可用：

```
$ yum module list postgresql
Name      Stream Profiles      Summary
postgresql 9.6  client, server [d] PostgreSQL server and client module
postgresql 10 [d] client, server [d] PostgreSQL server and client module
postgresql 12  client, server [d] PostgreSQL server and client module
postgresql 13  client, server [d] PostgreSQL server and client module
postgresql 15  client, server [d] PostgreSQL server and client module

Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
```

输出显示 `postgresql` 模块与流 **9.6**、**10**、**12**、**13** 和 **15** 一起提供。默认流为 **10**。

2. 在流 **13** 中安装 `postgresql` 模块提供的软件包：

```
# yum module install postgresql:13
...
Dependencies resolved.
=====
Package                Architecture Version                Repository            Size
=====
Installing group/module packages:
postgresql-server      x86_64      13.10-1.module+el8.7.0+18279+1ca8cf12  rhel-AppStream      5.6 M
Installing dependencies:
libcicu                x86_64      60.3-2.el8_1          rhel                  8.8 M
libpq                  x86_64      13.5-1.el8            rhel-AppStream        198 k
postgresql             x86_64      13.10-1.module+el8.7.0+18279+1ca8cf12  rhel-AppStream        1.5 M
Installing module profiles:
  postgresql/server
Enabling module streams:
  postgresql          13
```

## Transaction Summary

```
=====
Install 4 Packages
```

```
Total download size: 16 M
```

```
Installed size: 61 M
```

```
Is this ok [y/N]: y
```

```
...
```

```
Installed:
```

```
libicu-60.3-2.el8_1.x86_64
```

```
libpq-13.5-1.el8.x86_64
```

```
postgresql-13.10-1.module+el8.7.0+18279+1ca8cf12.x86_64
```

```
postgresql-server-13.10-1.module+el8.7.0+18279+1ca8cf12.x86_64
```

```
Complete!
```

因为没有指定安装配置集，所以使用默认配置集 **server**。

### 3. 验证已安装的 PostgreSQL 版本：

```
$ postgres --version
postgres (PostgreSQL) 13.10
```

## 4.4. 运行安装的内容

从 RHEL 8 存储库安装内容后，通常会启用新命令。如果命令源自模块启用的 RPM 软件包，则使用这些命令的体验应该没有不同。

### 步骤

- 要运行新命令，直接输入它们：

```
$ command
```

使用您要运行的命令的名称替换 *command*。

### 注意

在 RHEL 8 中，GCC Toolset 被打包为 Software Collection。要从打包为 Software Collection 的组件运行命令，请使用：

```
$ scl enable collection 'command'
```

使用 Software Collection 的名称替换 *collection*。

如需更多信息，请参阅 [使用 GCC Toolset](#)。

## 4.5. 安装 RHEL 8 内容的命令

以下是安装 Red Hat Enterprise Linux 8 内容的常用命令。

命令	描述
<b>yum install <i>package</i></b>	安装软件包。  如果软件包由模块流提供，则 <b>yum</b> 会解析所需的模块流，并在安装此软件包时自动启用它。这也会以递归方式对所有软件包依赖项进行。如果更多模块流满足要求，则使用默认模块流。
<b>yum module enable <i>module-name:stream</i></b>	使用特定流启用模块。  始终考虑模块流的 <a href="#">生命周期</a> 。
<b>yum module install <i>module-name:stream</i></b> <b>yum install @<i>module-name:stream</i></b>	使用特定的流和默认配置文件安装模块
<b>yum module install <i>module-name:stream/profile</i></b> <b>yum install @<i>module-name:stream/profile</i></b>	使用特定的流和配置文件安装模块。

## 4.6. 其他资源

- [使用 yum 安装软件包](#)
- [yum\(8\) 手册页](#)

## 第 5 章 删除 RHEL 8 内容

在以下部分中，了解如何删除 Red Hat Enterprise Linux 8 中的内容：

- [删除软件包](#)。
- [从模块流或配置文件中删除安装的内容](#)。
- [检查删除 RHEL 8 内容的命令](#)。

### 5.1. 删除安装的软件包

要删除安装在您系统上的软件包，请完成以下步骤。

#### 流程

- 要删除特定的软件包，请使用：

```
# yum remove package-name
```

使用您要删除的软件包的名称替换 *package-name*。



#### 注意

`yum` 命令将软件包与其他依赖软件包一起删除。

### 5.2. 删除安装的模块内容

当删除安装的模块内容时，您可以从所选配置集或整个流中删除软件包。



#### 重要

YUM 删除名称与使用配置文件或流安装的软件包相对应的所有软件包，包括其依赖软件包。在进行操作前，务必检查要删除的软件包列表，特别是您是否在系统中启用了自定义软件仓库。

#### 5.2.1. 从模块流中删除所有软件包

当您删除使用模块流安装的软件包时，`yum` 会删除名称与流安装的软件包相对应的所有软件包。这包括软件包的依赖项，但其他模块所需的软件包除外。

#### 先决条件

- 已启用模块流，并至少已安装了流中的一些软件包。
- 您了解[模块化依赖关系的解析](#)。

#### 流程

1. 从所选流中删除所有软件包：

```
# yum module remove --all module-name:stream
```

使用您要卸载的模块和流替换 *module-name* 和 *stream*。

2. 在进行删除事务前，请检查 **Removing:** 和 **Removing unused dependencies:** 中的软件包列表。
3. (可选) 重置或禁用流。

如果您只想从所选配置文件中删除软件包，请按照 [从安装的配置文件中删除软件包](#) 中的说明操作。

### 例 5.1. 从整个流中删除软件包

以下是如何从 **php:7.3** 模块流中删除所有软件包的一个示例。

#### 流程

1. 安装 **php:7.3** 模块流，包括所有可用的配置集：

```
# yum module install php:7.3/*
Updating Subscription Management repositories.
Last metadata expiration check: 0:20:19 ago on Tue Mar 3 11:32:05 2020.
Dependencies resolved.
=====
=
Package      Arch Version                               Repository                               Size
=====
=
Installing group/module packages:
libzip       x86_64 1.5.2-1.module+el8.1.0+3189+a1bff096 rhel-8-for-x86_64-
appstream-rpms 63 k
php-cli      x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 3.0 M
php-common   x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 663 k
php-devel    x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 735 k
php-fpm      x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 1.6 M
php-json     x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 73 k
php-mbstring x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 610 k
php-pear     noarch 1:1.10.9-1.module+el8.1.0+3189+a1bff096
rhel-8-for-x86_64-appstream-rpms 359 k
php-pecl-zip x86_64 1.15.4-1.module+el8.1.0+3189+a1bff096
rhel-8-for-x86_64-appstream-rpms 51 k
php-process  x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 84 k
php-xml     x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 188 k
Installing dependencies:
autoconf    noarch 2.69-27.el8                               rhel-8-for-x86_64-appstream-rpms 710
k
...
Installing weak dependencies:
perl-IO-Socket-IP
noarch 0.39-5.el8                               rhel-8-for-x86_64-appstream-rpms 47 k
```

```
...
Installing module profiles:
php/common
php/devel
php/minimal
Enabling module streams:
httpd          2.4
nginx          1.14
php            7.3
```

#### Transaction Summary

```
=====
=
Install 73 Packages

Total download size: 76 M
Installed size: 220 M
Is this ok [y/N]: y
```

### 2. 检查 **php** 模块：

```
$ yum module info php
...
Name       : php
Stream     : 7.3 [e] [a]
Version    : 8020020200715124551
Context    : ceb1cf90
Architecture : x86_64
Profiles   : common [d] [i], devel [i], minimal [i]
Default profiles : common
...
Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled, [a]ctive
```

### 3. 从 **php:7.3** 模块流中删除所有软件包：

```
# yum module remove --all php:7.3
Updating Subscription Management repositories.
Last metadata expiration check: 0:21:26 ago on Tue Mar 3 11:32:05 2020.
Dependencies resolved.
=====
=
Package           Arch Version                               Repository                               Size
=====
=
Removing:
libzip            x86_64 1.5.2-1.module+el8.1.0+3189+a1bff096
                                     @rhel-8-for-x86_64-appstream-rpms 313 k
php-cli           x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6
                                     @rhel-8-for-x86_64-appstream-rpms 11 M
php-common        x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6
                                     @rhel-8-for-x86_64-appstream-rpms 6.5 M
php-devel         x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6
                                     @rhel-8-for-x86_64-appstream-rpms 5.3 M
php-fpm           x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6
                                     @rhel-8-for-x86_64-appstream-rpms 5.6 M
```



```

php-json          x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6
                  @rhel-8-for-x86_64-appstream-rpms 53 k
php-mbstring      x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6
                  @rhel-8-for-x86_64-appstream-rpms 1.9 M
php-pear          noarch 1:1.10.9-1.module+el8.1.0+3189+a1bff096
                  @rhel-8-for-x86_64-appstream-rpms 2.1 M
php-pecl-zip      x86_64 1.15.4-1.module+el8.1.0+3189+a1bff096
                  @rhel-8-for-x86_64-appstream-rpms 119 k
php-process       x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6
                  @rhel-8-for-x86_64-appstream-rpms 117 k
php-xml           x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6
                  @rhel-8-for-x86_64-appstream-rpms 557 k

Removing unused dependencies:
autoconf         noarch 2.69-27.el8           @rhel-8-for-x86_64-appstream-rpms 2.2
M
...
Disabling module profiles:
php/common
php/devel
php/minimal

Transaction Summary
=====
=
Remove 73 Packages

Freed space: 220 M
Is this ok [y/N]: y

```

#### 4. 在移除后检查 **php** 模块：

```

$ yum module info php
...
Name       : php
Stream     : 7.3 [e] [a]
Version    : 8020020200715124551
Context    : ceb1cf90
Architecture : x86_64
Profiles   : common [d], devel, minimal
Default profiles : common
...
Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled, [a]ctive

```

**php** 模块的 7.3 流当前已启用，但没有安装此流中的软件包。

### 5.2.2. 从安装的配置集中删除软件包

当您删除配置文件安装的软件包时，**yum** 会删除名称与配置集安装的软件包相对应的所有软件包。这包括软件包依赖项，但不同配置文件所需的软件包除外。

#### 先决条件

- 所选配置文件已使用 **yum module install *module-name:stream/profile*** 命令安装，或使用 **yum install *module-name:stream*** 命令安装为默认配置文件。

- 您了解[模块化依赖关系的解析](#)。

## 流程

1. 卸载属于所选配置集的软件包：

```
# yum module remove module-name:stream/profile
```

将 `module-name`、`stream` 和 `profile` 替换为您要卸载的模块、流和配置集。

或者，从流中的所有安装配置集中卸载软件包：

```
# yum module remove module-name:stream
```

这些操作不会从不属于任何配置集的流中删除软件包。

2. 在进行删除事务前，请检查 **Removing:** 和 **Removing unused dependencies:** 中的软件包列表。

要从所选流中删除所有软件包，请按照 [从模块流中删除所有软件包](#) 中的说明操作。

### 例 5.2. 从所选配置集中删除软件包

以下是一个如何删除属于 **php:7.3** 模块流的 **devel** 配置文件的软件包及其依赖项的示例。



#### 注意

本例中的输出已编辑为 brevity。实际输出可能包含比这里显示的更多的信息。

## 流程

1. 安装 **php:7.3** 模块流，包括所有可用的配置集：

```
# yum module install php:7.3/*
Updating Subscription Management repositories.
Last metadata expiration check: 0:08:41 ago on Tue Mar 3 11:32:05 2020.
Dependencies resolved.
=====
=
Package      Arch Version                               Repository                               Size
=====
=
Installing group/module packages:
libzip      x86_64 1.5.2-1.module+el8.1.0+3189+a1bff096 rhel-8-for-x86_64-
appstream-rpms 63 k
php-cli     x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 3.0 M
php-common  x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 663 k
php-devel   x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 735 k
php-fpm     x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 1.6 M
php-json    x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 73 k
```

```

php-mbstring x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 610 k
php-pear noarch 1:1.10.9-1.module+el8.1.0+3189+a1bff096
rhel-8-for-x86_64-appstream-rpms 359 k
php-pecl-zip x86_64 1.15.4-1.module+el8.1.0+3189+a1bff096
rhel-8-for-x86_64-appstream-rpms 51 k
php-process x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 84 k
php-xml x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6 rhel-8-for-x86_64-
appstream-rpms 188 k
Installing dependencies:
autoconf noarch 2.69-27.el8 rhel-8-for-x86_64-appstream-rpms 710
k
...
Installing weak dependencies:
perl-IO-Socket-IP
noarch 0.39-5.el8 rhel-8-for-x86_64-appstream-rpms 47 k
...
Installing module profiles:
php/common
php/devel
php/minimal
Enabling module streams:
httpd 2.4
nginx 1.14
php 7.3

Transaction Summary
=====
=
Install 73 Packages

Total download size: 76 M
Installed size: 220 M
Is this ok [y/N]: y

```

## 2. 检查安装的配置集：

```

$ yum module info php
...
Name       : php
Stream     : 7.3 [e] [a]
Version    : 8020020200715124551
Context    : ceb1cf90
Architecture : x86_64
Profiles   : common [d] [i], devel [i], minimal [i]
Default profiles : common
Repo       : rhel-AppStream
....
Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled, [a]ctive

```

所有配置集都如输出中显示。

## 3. 从 **devel** 配置集中删除软件包：

```
# yum module remove php:7.3/devel
```

```

Updating Subscription Management repositories.
Last metadata expiration check: 0:09:40 ago on Tue Mar 3 11:32:05 2020.
Dependencies resolved.
=====
=
Package             Arch Version             Repository             Size
=====
=
Removing:
libzip              x86_64 1.5.2-1.module+el8.1.0+3189+a1bff096
                    @rhel-8-for-x86_64-appstream-rpms 313 k
php-devel           x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6
                    @rhel-8-for-x86_64-appstream-rpms 5.3 M
php-pear            noarch 1:1.10.9-1.module+el8.1.0+3189+a1bff096
                    @rhel-8-for-x86_64-appstream-rpms 2.1 M
php-pecl-zip        x86_64 1.15.4-1.module+el8.1.0+3189+a1bff096
                    @rhel-8-for-x86_64-appstream-rpms 119 k
php-process         x86_64 7.3.5-5.module+el8.1.0+4560+e0eee7d6
                    @rhel-8-for-x86_64-appstream-rpms 117 k

Removing unused dependencies:
autoconf            noarch 2.69-27.el8           @rhel-8-for-x86_64-appstream-rpms 2.2
M
...
Disabling module profiles:
php/devel

Transaction Summary
=====
=
Remove 64 Packages

Freed space: 193 M
Is this ok [y/N]: y

```

#### 4. 在移除后检查安装的配置集：

```

$ yum module info php
...
Name           : php
Stream         : 7.3 [e] [a]
Version        : 8020020200715124551
Context        : ceb1cf90
Architecture   : x86_64
Profiles       : common [d] [i], devel, minimal [i]
Default profiles : common
Repo           : rhel-AppStream
...
Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled, [a]ctive

```

除 **devel** 外的所有配置文件当前都已安装。

### 5.3. 删除内容的命令

以下是删除 Red Hat Enterprise Linux 8 中内容的常用命令。

命令	描述
<b>yum remove <i>package</i></b>	删除软件包。
<b>yum module remove <i>module-name:stream/profile</i></b>	从安装的配置文件删除软件包。
<b>yum module remove --all <i>module-name:stream</i></b>	从一个活动的流中删除所有软件包。
<b>yum module reset <i>module-name</i></b>	将模块重置为初始状态。
<b>yum module disable <i>module-name</i></b>	禁用一个模块及其所有流。

## 第 6 章 管理 APPLICATION STREAM 内容的版本

AppStream 仓库的内容可以在多个版本中提供，对应于模块流。

在以下部分中，了解更改现有启用的模块流时必须执行的操作：

- [模块依赖项和流更改](#)。
- [模块和非模块化依赖关系的交互](#)。
- [将模块重置为其初始状态](#)。
- [完全禁用模块及其所有流](#)。
- [切换到模块的更新的流](#)。
- [覆盖模块默认流](#)。

### 6.1. 模块依赖关系和流更改

传统上，提供内容的软件包依赖于其他软件包，并且通常指定所需的依赖项版本。对于模块中包含的软件包，此机制也会应用这个机制，但将软件包及其特定版本分组到模块和流中可以进一步的限制。另外，模块流可以声明与其他模块流的依赖关系，独立于其包含的软件包并提供它们。

在使用软件包或模块操作后，所有底层已安装软件包的所有依赖项树都必须满足软件包声明的所有条件。另外，必须满足所有模块流依赖项。

因此：

- 启用模块流可能需要启用更多模块的流。
- 安装模块流配置集或从流安装软件包需要启用更多模块的流并安装更多软件包。
- 禁用一个模块的流可能需要禁用其他模块流。不会自动删除任何软件包。
- 删除软件包可能需要删除其他软件包。如果这些软件包由模块提供，则模块流保持启用，来为进一步安装做准备，即使不再有从这些流安装的软件包。这反映了未使用的 YUM 存储库的行为。



#### 重要

当同一模块的另一个流已启用时，您不能启用模块的流。要切换流，请按照 [切换到更新的流](#) 中的流程操作。另外，还可重置模块，然后启用新流。

在切换到不同流之前从流中删除安装的所有软件包可防止系统达到这样一种状态，即软件包可以在没有提供它们的存储库或流的情况下安装。

从技术上讲，重置模块不会自动更改任何已安装的软件包。删除以前流提供的软件包以及依赖于它们的任何软件包都是明确的手动操作。

### 6.2. 模块化和非模块化依赖项的交互

[模块依赖关系](#) 是常规 RPM 依赖项之上的额外层。模块依赖关系与存储库间可能存在的依赖关系的行为相似。这意味着安装不同的软件包不仅需要解析 RPM 的依赖关系，还需要事先解决模块依赖关系。

系统总是保留模块和流选择，除非明确指示更改它们。模块软件包接收提供此软件包的当前启用的模块流中包含的更新，但不会升级到其他流中包含的版本。

### 6.3. 重置模块流

重置模块是一种操作，它会将所有流返回至初始状态（不启用或禁用）。如果模块有默认流，则此流会因重置模块而变为活动状态。

#### 流程

- 重置模块状态：

```
# yum module reset module-name
```

使用您要重置的模块的名称替换 *module-name*。

模块返回到初始状态。已启用的流和安装的配置集的信息会被清除，但没有删除安装的内容。

### 6.4. 禁用一个模块的所有流

有默认流的模块总是有一个流处于活动状态。当无法访问所有模块流中的内容时，可以禁用整个模块。

#### 先决条件

- 您了解了 [活动模块流的概念](#)。

#### 流程

- 禁用模块：

```
# yum module disable module-name
```

使用您要禁用的模块的名称替换 *module-name*。

**yum** 命令要求确认，然后禁用模块及其所有流。所有模块流都不再活跃。没有安装的内容被删除。

### 6.5. 切换到更新的流

当您切换到更新的模块流时，该模块中的所有软件包都会被以后的版本替代。



#### 重要

这个流程只适用于先决条件部分中描述的情况。

#### 先决条件

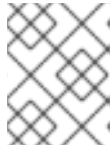
- 这个系统已被完全更新。
- 在系统中安装的软件包都比该程序库中可用软件包更新。

#### 流程

1. 确定您的系统是否已准备好切换到更新的流：

```
# yum distro-sync
```

这个命令必须以以下信息结束：*Nothing to do Complete!* 如果它没有提出更改而要求确认，请仔细检查这些更改，并考虑是否要继续。如果需要，重复运行 **yum distro-sync** 命令。另外，您可以拒绝推荐的修改，然后手动将您的系统改为命令返回 *Nothing to do. Complete!*



### 注意

通过在切换流前检查 **yum distro-sync** 结果，您可以防止更改与流交换相关的系统，因为这个过程的最后一步需要同样的命令。

2. 将活动流更改为后续流：

```
# yum module reset module-name
# yum module enable module-name:new-stream
```

3. 同步安装的软件包以在流间进行更改：

```
# yum distro-sync
```

如果此操作建议更改流外的内容，请仔细检查它们。



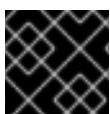
### 注意

- 如果某些安装的软件包依赖于以前的流，且后续流中没有兼容版本，**yum** 会报告依赖项冲突。在这种情况下，使用 **--allowerase** 选项删除这些软件包，因为它们因为缺少依赖项而无法与后续流一起安装。
- 在切换 **Perl** 模块时，您必须始终使用 **--allowerase** 选项，因为基本 RHEL 8 安装中的某些软件包依赖于 **Perl 5.26**。
- 在启用了新流后，需要重新安装解释语言的二进制扩展（通常使用 **C** 或 **C++**）；例如，由 **ruby** 模块安装的 **gem** 命令、**nodejs** 模块的 **npm** 命令、**perl** 模块中的 **cpan** 命令或 **php** 模块的 **pecl** 命令。如需更多信息，请参阅 [如何在 RHEL 8 中切换 Ruby 流](#)。

或者，[删除从当前流安装的所有模块内容](#)，[重置模块](#)，并 [安装新的流](#)。

## 6.6. 覆盖模块默认流

默认情况下，**YUM** 工具使用包含模块的存储库中定义的模块默认流。您可以覆盖 `/etc/dnf/modules.defaults.d/` 目录中的默认流。



### 重要

始终考虑模块流的 [生命周期](#)。

### 先决条件

- 您理解 [活动模块流的概念](#)。



## 流程

1. 在 `/etc/dnf/modules.defaults.d/` drop-in 目录中创建 YAML 配置文件。

```
---
document: modulemd-defaults
version: 1
data:
  module: postgresql
  stream: "10"
  profiles:
    10: [server]
    12: [server]
    13: [server]
    15: [server]
    9.6: [server]
...
```

前面的输出表示在编写本文时 `postgresql` 模块的默认定义。

### 例 6.1. 带有原始默认值的 postgresql 模块示例

以下一个是如何将 `postgresql` 模块的流 `13` 配置为默认流的示例。

1. 检查 `postgresql` 模块：

```
# yum module list postgresql
(...)
Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)
Name           Stream   Profiles      Summary
postgresql    9.6      client, server [d]  PostgreSQL server and client module
postgresql    10 [d]   client, server [d]  PostgreSQL server and client module
postgresql    12       client, server [d]  PostgreSQL server and client module
postgresql    13       client, server [d]  PostgreSQL server and client module
postgresql    15       client, server [d]  PostgreSQL server and client module
...
Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
```

2. 要将默认流设置为 `13`，请在 `/etc/dnf/modules.defaults.d/postgresql.yaml` 文件中实现以下 YAML 文件配置。

```
---
document: modulemd-defaults
version: 1
data:
  module: postgresql
  stream: "13"
  profiles:
    10: [server]
    12: [server]
    13: [server]
    15: [server]
    9.6: [server]
...
```

3. 再次检查 `postgresql` 模块：

```
# yum module list postgresql
(...)
Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)
Name           Stream   Profiles   Summary
postgresql    9.6     client, server [d]  PostgreSQL server and client module
postgresql    10     client, server [d]  PostgreSQL server and client module
postgresql    12     client, server [d]  PostgreSQL server and client module
postgresql    13 [d]   client, server [d]  PostgreSQL server and client module
postgresql    15     client, server [d]  PostgreSQL server and client module
...
Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
```