



# Red Hat Enterprise Linux 8

## 사용자 공간 구성 요소 설치, 관리 및 제거

YUM 소프트웨어 관리 도구를 사용하여 BaseOS 및 AppStream 리포지토리의 콘텐츠 관리



# Red Hat Enterprise Linux 8 사용자 공간 구성 요소 설치, 관리 및 제거

---

YUM 소프트웨어 관리 도구를 사용하여 BaseOS 및 AppStream 리포지토리의 콘텐츠 관리

## 법적 공지

Copyright © 2023 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.

## 초록

YUM 툴을 사용하여 BaseOS 및 AppStream 리포지토리를 통해 배포된 콘텐츠를 찾고, 설치하고, 활용합니다. 패키지, 모듈, 스트림 및 프로필 작업을 수행하는 방법을 알아봅니다.

## 차례

보다 포괄적 수용을 위한 오픈 소스 용어 교체 .....	3
RED HAT 문서에 관한 피드백 제공 .....	4
<b>1장. APPSTREAM 사용</b> .....	<b>5</b>
1.1. RHEL 8의 콘텐츠 배포	5
1.2. APPLICATION STREAMS	5
1.3. RHEL 8의 패키징 방법	6
1.4. RHEL 8의 YUM을 사용하는 패키지 관리	6
<b>2장. 모듈 소개</b> .....	<b>7</b>
2.1. 모듈 스트림	7
2.2. 모듈 프로필	8
<b>3장. RHEL 8 콘텐츠 검색</b> .....	<b>9</b>
3.1. 패키지 검색	9
3.2. 사용 가능한 모듈 및 콘텐츠 나열	9
3.3. 콘텐츠를 나열하는 명령	13
<b>4장. RHEL 8 콘텐츠 설치</b> .....	<b>14</b>
4.1. 패키지 설치	14
4.2. 패키지를 설치하기 전에 스트림 선택	14
4.3. 모듈식 콘텐츠 설치	15
4.4. 설치된 콘텐츠 실행	17
4.5. RHEL 8 콘텐츠 설치를 위한 명령	18
4.6. 추가 리소스	18
<b>5장. RHEL 8 콘텐츠 제거</b> .....	<b>19</b>
5.1. 설치된 패키지 제거	19
5.2. 설치된 모듈식 콘텐츠 제거	19
5.3. 콘텐츠 제거를 위한 명령	26
<b>6장. 애플리케이션 스트림 콘텐츠 버전 관리</b> .....	<b>27</b>
6.1. 모듈식 종속성 및 스트림 변경 사항	27
6.2. 모듈식 및 비모형 종속성의 상호 작용	28
6.3. 모듈 스트림 재설정	28
6.4. 모듈의 모든 스트림 비활성화	28
6.5. 이후 스트림으로 전환	28
6.6. 모듈 기본 스트림 덮어쓰기	30



## 보다 포괄적 수용을 위한 오픈 소스 용어 교체

Red Hat은 코드, 문서 및 웹 속성에서 문제가 있는 언어를 교체하기 위해 최선을 다하고 있습니다. 먼저 마스터(master), 슬레이브(slave), 블랙리스트(blacklist), 화이트리스트(whitelist) 등 네 가지 용어를 교체하고 있습니다. 이러한 변경 작업은 작업 범위가 크므로 향후 여러 릴리스에 걸쳐 점차 구현할 예정입니다. 자세한 내용은 [CTO Chris Wright의 메시지](#)를 참조하십시오.

## RED HAT 문서에 관한 피드백 제공

문서에 대한 피드백에 감사드립니다. 어떻게 개선할 수 있는지 알려주십시오.

### 특정 문구에 대한 의견 제출

1. **Multi-page HTML** 형식으로 설명서를 보고 페이지가 완전히 로드된 후 오른쪽 상단 모서리에 **피드백** 버튼이 표시되는지 확인합니다.
2. 커서를 사용하여 주석 처리할 텍스트 부분을 강조 표시합니다.
3. 강조 표시된 텍스트 옆에 표시되는 **피드백 추가** 버튼을 클릭합니다.
4. 의견을 추가하고 **제출** 을 클릭합니다.

### Jira를 통해 피드백 제출 (등록 필요)

1. [Jira](#) 웹 사이트에 로그인합니다.
2. 상단 탐색 모음에서 **생성** 을 클릭합니다.
3. **Summary** (요약) 필드에 설명 제목을 입력합니다.
4. **Description** (설명) 필드에 개선을 위한 제안을 입력합니다. 문서의 관련 부분에 대한 링크를 포함합니다.
5. 대화 상자 하단에서 **생성** 을 클릭합니다.



# 1장. APPSTREAM 사용

다음 섹션에서는 Red Hat Enterprise Linux 8의 AppStream 리포지토리 및 관련된 개념을 설명합니다.

- [RHEL 8의 콘텐츠 배포](#).
- [애플리케이션 스트림](#).
- [RHEL 8의 패키징 방법](#).
- [RHEL 8에서 YUM을 사용한 패키지 관리](#).

## 1.1. RHEL 8의 콘텐츠 배포

Red Hat Enterprise Linux 8 콘텐츠는 다음 두 가지 주요 리포지토리를 통해 배포됩니다. **BaseOS** 및 **AppStream**.

### BaseOS

BaseOS 리포지토리에는 모든 설치의 기반을 제공하는 기본 운영 체제 기능의 코어 세트가 포함되어 있습니다. 이 콘텐츠는 RPM 패키지 형태로 제공되며 이전 Red Hat Enterprise Linux 릴리스와 유사한 지원 조건이 적용됩니다.

### AppStream

AppStream 리포지토리에는 다양한 워크로드 및 사용 사례를 지원하는 추가 사용자 공간 애플리케이션, 런타임 언어 및 데이터베이스가 포함되어 있습니다. AppStream의 콘텐츠는 RPM 패키지 및 *모듈*이라는 RPM 형식의 확장이라는 두 가지 형식 중 하나로 사용할 수 있습니다.



#### 중요

BaseOS 및 AppStream 콘텐츠 세트는 기본 RHEL 설치에 필요하며 모든 RHEL 서브스크립션에서 사용할 수 있습니다. 설치 지침은 [표준 RHEL 8 설치](#) 문서를 참조하십시오.

## 1.2. APPLICATION STREAMS

Red Hat Enterprise Linux 8에는 Application Streams의 개념 - 사용자 공간 구성 요소 버전이 도입되었습니다. 이제 이러한 구성 요소의 여러 버전이 핵심 운영 체제 패키지보다 더 자주 제공되고 업데이트됩니다. 따라서 플랫폼 또는 특정 배포의 기본 안정성에 영향을 주지 않고 Red Hat Enterprise Linux를 사용자 정의할 수 있는 유연성이 향상됩니다.

Application Streams로 제공되는 구성 요소는 모듈 또는 RPM 패키지로 패키징할 수 있으며 Red Hat Enterprise Linux 8의 AppStream 리포지토리를 통해 제공됩니다. 각 Application Stream에는 RHEL 8과 같거나 더 짧은 기간 동안 특정 애플리케이션에 더 적합한 라이프 사이클이 있습니다. 라이프 사이클이 짧은 Application Streams는 [Red Hat Enterprise Linux 8 Application Streams 라이프 사이클](#) 페이지에 나열되어 있습니다.



#### 참고

모든 모듈이 Application Streams인 것은 아닙니다. 다른 모듈의 종속 항목은 Application Streams로 간주되지 않습니다.

### 추가 리소스

- [Red Hat Enterprise Linux 라이프 사이클](#)

- [Red Hat Enterprise Linux 8 Application Streams 라이프 사이클](#)

### 1.3. RHEL 8의 패키징 방법

AppStream 리포지토리의 콘텐츠는 다음 두 가지 방법으로 패키징됩니다.

- **개별 RPM 패키지**  
즉시 설치할 수 있는 기존 RPM 패키지
- **모듈**  
모듈은 논리 단위, 애플리케이션, 언어 스택, 데이터베이스 또는 툴 세트를 나타내는 패키지 컬렉션입니다. 이러한 패키지는 함께 빌드, 테스트, 릴리스됩니다.

### 1.4. RHEL 8의 YUM을 사용하는 패키지 관리

이제 **YUM** 패키지 관리 도구가 DNF 기술을 기반으로 하며 새로운 모듈식 기능에 대한 지원이 추가되었습니다.

개별 RPM 패키지를 처리할 때 **YUM**의 사용은 변경되지 않았습니다. 모듈식 콘텐츠를 처리하기 위해 **yum module** 명령이 추가되었습니다. 자세한 내용은 [RHEL 8 콘텐츠 설치](#)를 참조하십시오.

필요한 경우 모듈식 기능은 자동으로 적절한 모듈과 스트림 조합을 선택하여 편리한 사용을 위해 논리 패키지 세트를 설치할 수 있습니다.

## 2장. 모듈 소개

개별 RPM 패키지 외에도 AppStream 리포지토리에는 모듈이 포함되어 있습니다. 모듈은 구성 요소를 나타내는 RPM 패키지 세트이며 일반적으로 함께 설치됩니다. 일반적인 모듈에는 애플리케이션이 있는 패키지, 애플리케이션별 종속성 라이브러리가 있는 패키지, 애플리케이션에 대한 문서가 있는 패키지, 도우미 유틸리티가 있는 패키지가 포함되어 있습니다.

다음 섹션에서는 조직 및 모듈 내의 콘텐츠 처리 기능에 대해 알아봅니다.

- [모듈 스트림](#) - 버전별 콘텐츠 구성.
- [모듈 프로필](#) - 용도별 콘텐츠 구성입니다.

### 2.1. 모듈 스트림

모듈 스트림은 AppStream 물리적 리포지토리에서 가상 리포지토리로 간주할 수 있는 필터입니다. 모듈 스트림은 AppStream 구성 요소의 버전을 나타냅니다. 각 스트림은 독립적으로 업데이트를 수신합니다.

모듈 스트림은 활성 또는 비활성 상태일 수 있습니다. 활성 스트림은 특정 모듈 스트림 내의 RPM 패키지에 대한 시스템에 액세스할 수 있으므로 해당 구성 요소 버전을 설치할 수 있습니다. 스트림은 기본적으로 표시되거나 사용자 작업에 의해 명시적으로 활성화된 경우 활성화됩니다.

특정 모듈의 한 스트림만 지정된 시점에서 활성화할 수 있습니다. 따라서 하나의 구성 요소 버전만 시스템에 설치할 수 있습니다. 개별 컨테이너에서 서로 다른 버전을 사용할 수 있습니다.

각 모듈에는 기본 스트림이 있을 수 있습니다. 기본 스트림을 사용하면 모듈에 대해 배울 필요 없이 RHEL 패키지를 일반적인 방식으로 쉽게 사용할 수 있습니다. 전체 모듈이 비활성화되거나 해당 모듈의 다른 스트림이 활성화된 경우가 아니면 기본 스트림이 활성화됩니다.



#### 중요

기본 스트림은 RHEL 주요 릴리스 전체에서 변경되지 않습니다. 각 스트림의 [라이프 사이클](#)을 항상 고려하십시오. RHEL 주요 릴리스가 종료되기 전에 기본 스트림이 Life 상태에 도달하는 인스턴스에 대해 기본 스트림을 사용하지 마십시오.

특정 모듈 스트림은 다른 모듈 스트림에 따라 다릅니다. 예를 들어 `perl-App-cpanminus`, `perl-DBD-MySQL`, `perl-DBD-Pg`, `perl-DBD-SQLite`, `perl-DBI`, `perl-YAML`, `freeradius` 모듈 스트림은 특정 `perl` 모듈 스트림에 따라 달라집니다.

런타임 사용자 애플리케이션 또는 개발자 애플리케이션에 대한 특정 스트림을 선택하려면 다음을 고려하십시오.

- 필수 기능 및 이를 지원하는 구성 요소 버전
- 호환성
- [라이프 사이클](#) 길이 및 업데이트 계획

사용 가능한 모든 모듈 및 스트림 목록은 [패키지 매니페스트](#)를 참조하십시오. 구성 요소별 변경 사항은 [릴리스 노트](#)를 참조하십시오.

#### 예 2.1. PostgreSQL 모듈 스트림

`postgresql` 모듈은 각 스트림 9.6, 10, 12, 13 및 15에 PostgreSQL 데이터베이스 버전 9.6, 10, 12 및 13을 제공합니다. 스트림 10은 기본값입니다. 즉, `postgresql`에 대한 요청 시 `postgresql-10.6` 패키지를 설치하려고 합니다.

항상 사용할 모듈 스트림을 결정하고 버전을 명시적으로 설치합니다.

### 추가 리소스

- [모듈식 종속 항목 및 스트림 변경](#)
- [이후 스트림으로 전환](#)
- [패키지 매니페스트](#)
- [릴리스 노트](#)

## 2.2. 모듈 프로필

프로필은 서버, 클라이언트, 개발, 최소 설치 또는 기타와 같은 특정 사용 사례에 대해 함께 설치되는 권장 패키지 목록입니다. 이러한 패키지 목록에는 일반적으로 BaseOS 리포지토리 또는 스트림의 종속 항목에서 모듈 스트림 외부의 패키지가 포함될 수 있습니다.

프로필을 사용하여 패키지를 설치하는 것은 사용자의 편의를 위해 제공되는 일회성 작업입니다. 모듈에서 제공하는 패키지를 설치하거나 제거하지는 않습니다. 추가 준비 단계 없이 동일한 모듈 스트림의 여러 프로필을 사용하여 패키지를 설치할 수도 있습니다.

각 모듈 스트림은 `none`을 포함하여 여러 프로필을 가질 수 있습니다. 지정된 모듈 스트림에 대해 일부 프로필을 기본값으로 표시할 수 있으며, 명시적으로 지정되지 않은 경우 프로필 설치 작업에 사용할 수 있습니다. 그러나 모듈 스트림에 대한 기본 프로필이 있을 필요는 없습니다.

### 예 2.2. httpd 모듈 프로필

Apache 웹 서버를 제공하는 `httpd` 모듈은 설치를 위해 다음 프로필을 제공합니다.

- **common** - 강화된 프로덕션 지원 배포인 default 프로필입니다.
- **devel** - `httpd`를 수정하는 데 필요한 패키지입니다.
- **최소** - 실행 중인 웹 서버를 제공하는 최소 패키지 세트입니다.

## 3장. RHEL 8 콘텐츠 검색

다음 섹션에서는 **YUM** 을 사용하여 Red Hat Enterprise Linux 8의 AppStream 및 BaseOS 리포지토리에  
서 콘텐츠를 찾아서 검사하는 방법을 알아봅니다.

- 원하는 콘텐츠를 제공하는 패키지를 검색합니다.
- 사용 가능한 모듈을 나열하고 이에 대한 세부 정보를 확인하십시오 .
- RHEL 8 콘텐츠를 검사하는 데 유용한 명령을 검사합니다.

### 3.1. 패키지 검색

특정 애플리케이션 또는 기타 콘텐츠를 제공하는 패키지를 찾으려면 다음 단계를 완료합니다.

#### 절차

1. 애플리케이션 이름과 같은 텍스트 문자열이 있는 패키지를 검색합니다.

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

2. 패키지에 대한 세부 정보를 확인합니다.

```
$ yum info package
```

### 3.2. 사용 가능한 모듈 및 콘텐츠 나열

사용 가능한 모듈과 세부 정보를 확인하려면 다음 단계를 완료합니다.

#### 절차

- 시스템에서 사용 가능한 모듈 스트림을 나열하려면 다음을 사용합니다.

```
$ yum module list
```

이 명령의 출력은 별도의 행에 이름, 스트림, 프로필 및 요약이 포함된 모듈 스트림을 나열합니다.

- 설명, 모든 프로필 목록 및 제공된 모든 패키지 목록을 포함하여 모듈에 대한 세부 정보를 표시하려면 다음을 사용합니다.

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

- 각 모듈 프로필에서 설치하는 패키지를 나열하려면 다음을 사용합니다.

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

- 활성화된 스트림 및 설치된 프로필을 포함하여 모듈의 현재 상태를 표시하려면 다음을 사용합니다.

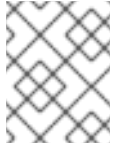
```
$ yum module list module-name
```

## 추가 리소스

- [모듈 소개](#)

### 예 3.1. 모듈에 대한 세부 정보 검색

다음은 AppStream 리포지토리에서 사용 가능한 모듈을 나열하는 방법 및 **postgresql** 모듈 콘텐츠에 대한 정보를 가져오는 방법의 예입니다.



#### 참고

이 예제의 출력은 간결성을 위해 편집되었습니다. 실제 출력에는 여기에 표시된 것보다 더 많은 정보가 포함될 수 있습니다.

1. 사용 가능한 모듈을 나열합니다.

```
$ 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. **postgresql** 모듈에 대한 세부 정보를 검사합니다.

```
$ 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

스트림을 지정하지 않으면 **yum** 은 사용 가능한 모든 스트림을 나열합니다.

### 3. **postgresql** 모듈의 스트림 **10** 에서 사용 가능한 프로필을 검사합니다.

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

각 프로필은 종속 항목을 포함하여 다른 패키지 세트를 설치합니다.

4. 기본 스트림 **10** 및 기본 프로필 **서버**를 사용하여 **postgresql** 모듈을 설치합니다.

```
# 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. **postgresql** 모듈의 현재 상태를 검사합니다.

```
$ 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

출력에서 기본 스트림 **10** 이 활성화되고 해당 프로필 **서버**가 설치되어 있음을 보여줍니다.

### 3.3. 콘텐츠를 나열하는 명령

다음은 Red Hat Enterprise Linux 8에서 콘텐츠와 세부 정보를 찾는 데 일반적으로 사용되는 명령입니다.

명령	설명
<b>yum list</b> 사용 가능	사용 가능한 패키지를 나열합니다.
<b>yum repoquery</b> 패키지	선택한 패키지에 대해 사용 가능한 YUM 리포지토리를 검색합니다.
<b>yum search</b> "text string"	임의의 텍스트 문자열을 사용하여 패키지를 검색합니다.
<b>yum info</b> 패키지	패키지의 세부 정보를 표시합니다.
<b>yum module</b> 에서 패키지제공	패키지를 제공하는 모듈을 표시합니다.  모듈 외부에서 패키지를 사용할 수 있는 경우 이 명령의 출력은 비어 있습니다.
<b>yum module list</b>	사용 가능한 모듈을 나열합니다.
<b>yum module info</b> module-name	모듈의 세부 정보를 표시합니다.
<b>yum module info --profile</b> module-name	기본 스트림을 사용하여 모듈 프로필에서 설치한 패키지를 나열합니다.
<b>yum module info --profile</b> module-name:stream	지정된 스트림을 사용하여 모듈 프로필로 설치한 패키지를 표시합니다.
<b>yum module list</b> module-name	모듈의 현재 상태를 표시합니다.

## 4장. RHEL 8 콘텐츠 설치

다음 섹션에서는 Red Hat Enterprise Linux 8에 콘텐츠를 설치하는 방법을 설명합니다.

- 패키지를 설치합니다.
- 패키지 설치 스트림을 선택합니다.
- 모듈, 스트림 및 프로필에서 제공하는 패키지 세트를 설치합니다.
- RHEL 8 설치된 콘텐츠를 실행합니다.
- RHEL 8 콘텐츠를 설치하는 데 유용한 명령을 검사합니다.

### 4.1. 패키지 설치

패키지를 설치하려면 다음 단계를 완료합니다.

#### 절차

- 패키지를 설치합니다.

```
# yum install package
```

패키지를 패키지 이름으로 바꿉니다.

- 모듈 스트림에서 패키지를 제공하지 않는 경우 이 절차는 이전 버전의 Red Hat Enterprise Linux에서 사용된 절차와 동일합니다.
- 패키지가 활성화된 모듈 스트림에 의해 제공되는 경우 추가 조작 없이 패키지가 설치됩니다.
- 패키지가 기본값으로 표시된 모듈 스트림에서 제공되는 경우 **yum** 은 이 패키지를 설치하기 전에 해당 모듈 스트림을 자동으로 활성화합니다.



#### 중요

기본 스트림을 사용하는 대신 설치할 특정 모듈 스트림을 항상 선택하는 것이 좋습니다. 특정 기본 모듈 스트림은 RHEL 주요 릴리스가 종료되기 전에 라이프 사이클 종료 상태에 도달합니다. 항상 각 스트림의 [라이프](#) 사이클을 고려하십시오.

- 활성화 상태가 아닌 모듈 스트림에서 패키지를 제공하는 경우(위의 경우 모두) 각 모듈 스트림을 수동으로 활성화할 때까지 인식되지 않습니다.

#### 추가 리소스

- [모듈식 콘텐츠 설치](#)
- [RHEL 8의 YUM을 사용하는 패키지 관리](#)
- [Red Hat Enterprise Linux Application Streams Life Cycle](#)

### 4.2. 패키지를 설치하기 전에 스트림 선택

설치에 대해 항상 특정 모듈 스트림을 선택하는 것이 좋습니다. 각 스트림의 [라이프 사이클](#)을 항상 고려하십시오.



### 중요

특정 기본 모듈 스트림은 RHEL 주요 릴리스가 종료되기 전에 라이프사이클 종료 상태에 도달합니다.

기본이 아닌 스트림에서 패키지를 설치하려면 먼저 스트림을 활성화합니다.

### 사전 요구 사항

- [활성 모듈 스트림의 개념](#)을 이해합니다.

### 절차

- 모듈 스트림을 활성화합니다.

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

*module-name* 및 *stream* 을 모듈 및 스트림 이름으로 교체합니다.

**yum** 은 확인을 요청하고 스트림이 활성화되고 활성화됩니다.



### 참고

모듈의 다른 스트림이 기본값이 되었기 때문에 이전에 활성화된 경우 더 이상 활성화 상태가 아닙니다.

### 추가 리소스

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

## 4.3. 모듈식 콘텐츠 설치

모듈 스트림 또는 프로필에서 제공하는 모듈식 콘텐츠를 설치하려면 다음 단계를 완료합니다.

### 사전 요구 사항

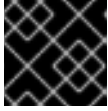
- [활성 모듈 스트림의 개념](#)을 이해합니다.
- 동일한 모듈의 다른 스트림에서 패키지를 설치할 수 없습니다.

### 절차

- 선택한 모듈 스트림을 설치하려면 다음을 사용합니다.

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

이 명령을 실행하면 선택한 스트림을 자동으로 활성화합니다. 기본 프로필이 스트림에 대해 정의되면 이 프로필이 자동으로 설치됩니다.



**중요**

모듈 스트림의 [라이프 사이클](#)을 항상 고려하십시오.

- 모듈 스트림의 선택한 프로필을 설치하려면 다음을 사용합니다.

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

이 명령을 실행하면 스트림을 활성화하고 모듈의 지정된 스트림(버전) 및 프로필(purpose)에 권장되는 패키지 세트를 설치합니다.

**추가 리소스**

- [모듈 소개](#)
- [RHEL 8 콘텐츠 설치를 위한 명령](#)
- [Red Hat Enterprise Linux Application Streams Life Cycle](#)

**예 4.1. 애플리케이션의 기본이 아닌 스트림 설치**

다음은 버전 이외의 스트림(버전), 즉, 버전 **13**의 PostgreSQL 서버(**postgresql-server** 패키지)에서 애플리케이션을 설치하는 방법의 예입니다. 기본 스트림은 버전 **10**을 제공합니다.

**절차**

1. **postgresql-server** 패키지를 제공하는 모듈을 나열하여 사용 가능한 스트림을 확인합니다.

```
$ 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

출력에서 **postgresql** 모듈을 스트림 **9.6,10,12,12,13** 및 **15**와 함께 사용할 수 있음을 보여줍니다. 기본 스트림은 **10**입니다.

2. 스트림 **13**의 **postgresql** 모듈에서 제공하는 패키지를 설치합니다.

```
# 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:
```

```

libicu          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!

설치 프로필이 지정되지 않았기 때문에 기본 프로필 **서버**가 사용되었습니다.

3. 설치된 **PostgreSQL** 버전을 확인합니다.

```

$ postgres --version
postgres (PostgreSQL) 13.10

```

## 4.4. 설치된 콘텐츠 실행

일반적으로 RHEL 8 리포지토리에서 콘텐츠를 설치한 후 새 명령이 활성화됩니다. 모듈에서 활성화한 RPM 패키지에서 명령이 시작된 경우 이러한 명령 사용 환경은 다를 수 없습니다.

### 절차

- 새 명령을 실행하려면 직접 입력합니다.

```

$ command

```

실행할 명령의 이름으로 명령을 바꿉니다.



**참고**

RHEL 8에서 GCC Toolset은 소프트웨어 컬렉션으로 패키징됩니다. 소프트웨어 컬렉션으로 패키징된 구성 요소에서 명령을 실행하려면 다음을 사용합니다.

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

컬렉션을 소프트웨어 컬렉션의 이름으로 교체합니다.

자세한 내용은 [GCC Toolset 사용](#)을 참조하십시오.

### 4.5. RHEL 8 콘텐츠 설치를 위한 명령

다음은 Red Hat Enterprise Linux 8 콘텐츠를 설치하는 데 일반적으로 사용되는 명령입니다.

명령	설명
<code>yum install 패키지</code>	패키지를 설치합니다.  모듈 스트림에서 패키지를 제공하는 경우 <b>yum</b> 은 필요한 모듈 스트림을 확인하고 이 패키지를 설치하는 동안 자동으로 활성화합니다. 이는 모든 패키지 종속 항목에 대해 반복적으로 수행됩니다. 더 많은 모듈 스트림이 요구 사항을 충족하는 경우 기본 모듈이 사용됩니다.
<code>yum module enable module-name:stream</code>	특정 스트림을 사용하여 모듈을 활성화합니다.  모듈 스트림의 <a href="#">라이프 사이클</a> 을 항상 고려하십시오.
<code>yum module install module-name:stream</code> <code>yum install @module-name:stream</code>	특정 스트림 및 기본 프로필을 사용하여 모듈 설치
<code>yum module install module-name:stream/profile</code> <code>yum install @module-name:stream/profile</code>	특정 스트림 및 프로필을 사용하여 모듈을 설치합니다.

### 4.6. 추가 리소스

- [yum](#)을 사용하여 소프트웨어 패키지 설치
- [yum\(8\)](#) man page

## 5장. RHEL 8 콘텐츠 제거

다음 섹션에서는 Red Hat Enterprise Linux 8에서 콘텐츠를 제거하는 방법을 설명합니다.

- 패키지를 제거합니다.
- 모듈 스트림 또는 프로필에서 설치된 콘텐츠를 제거합니다.
- RHEL 8 콘텐츠를 제거하는 명령을 검사합니다.

### 5.1. 설치된 패키지 제거

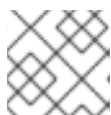
시스템에 설치된 패키지를 제거하려면 다음 단계를 완료합니다.

#### 절차

- 특정 패키지를 제거하려면 다음을 사용합니다.

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

*package-name* 을 제거할 패키지 이름으로 바꿉니다.



#### 참고

**yum** 명령은 다른 종속 패키지와 함께 패키지를 제거합니다.

### 5.2. 설치된 모듈식 콘텐츠 제거

설치된 모듈식 콘텐츠를 제거할 때 선택한 프로필 또는 전체 스트림에서 패키지를 제거할 수 있습니다.



#### 중요

**YUM** 은 종속 패키지를 포함하여 프로필 또는 스트림과 함께 설치된 패키지에 해당하는 이름을 가진 모든 패키지를 제거합니다. 특히 시스템에서 사용자 지정 리포지토리를 활성화한 경우, 진행하기 전에 제거할 패키지 목록을 항상 확인하십시오.

#### 5.2.1. 모듈 스트림에서 모든 패키지 제거

모듈 스트림으로 설치된 패키지를 제거하면 **yum** 은 스트림에서 설치한 패키지에 해당하는 이름을 가진 모든 패키지를 제거합니다. 여기에는 다른 모듈에 필요한 패키지를 제외하고 패키지의 종속성이 포함됩니다.

#### 사전 요구 사항

- 모듈 스트림이 활성화되었으며 스트림의 일부 패키지가 설치되었습니다.
- 모듈식 종속성 확인을 이해할 수 있습니다.

#### 절차

1. 선택한 스트림에서 모든 패키지를 제거합니다.

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

- 
- module-name* 및 *stream* 을 설치 제거할 모듈 및 스트림으로 바꿉니다.
- 2. **제거 중인 패키지 목록을 확인하고 사용하지 않는 종속 항목 제거:** 제거 트랜잭션을 진행하기 전에 확인하십시오.
- 3. 선택적으로 스트림을 재설정하거나 비활성화합니다.

선택한 프로필에서 패키지만 제거하려면 [설치된 프로필에서 패키지 제거](#) 의 지침을 따르십시오.

### 예 5.1. 전체 스트림에서 패키지 제거

다음은 **php:7.3** 모듈 스트림에서 모든 패키지를 제거하는 방법의 예입니다.

#### 절차

1. 사용 가능한 모든 프로필을 포함하여 **php:7.3** 모듈 스트림을 설치합니다.

```
# 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. php 모듈을 검사합니다.

```
$ 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. php:7.3 모듈 스트림에서 모든 패키지를 제거합니다.

```
# 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. 제거 후 **php** 모듈을 검사합니다.

```

$ 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** 모듈의 **7.3** 스트림은 현재 활성화되어 있지만 이 스트림의 패키지는 설치되지 않습니다.

### 5.2.2. 설치된 프로파일에서 패키지 제거

프로파일로 설치된 패키지를 제거하면 **yum** 은 프로파일에서 설치한 패키지에 해당하는 이름이 있는 모든 패키지를 제거합니다. 여기에는 다른 프로파일에 필요한 패키지를 제외하고 패키지 종속성이 포함됩니다.

#### 사전 요구 사항

- **yum module install *module-name:stream/profile*** 명령 또는 **yum install *module-name:stream*** 명령을 사용하여 선택한 프로파일이 설치되었습니다.

- 모듈식 종속성 확인을 이해할 수 있습니다.

## 절차

1. 선택한 프로필에 속하는 패키지 설치 제거:

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

*module-name, stream, profile* 을 설치 제거할 모듈, 스트림, 프로필로 바꿉니다.

또는 스트림 내 설치된 모든 프로필의 패키지를 설치 제거합니다.

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

이러한 작업은 프로필에 속하지 않는 스트림에서 패키지를 제거하지 않습니다.

2. 제거 중인 패키지 목록을 확인하고 **사용하지 않는 종속 항목 제거**: 제거 트랜잭션을 진행하기 전에 확인하십시오.

선택한 스트림에서 모든 패키지를 제거하려면 **모듈 스트림에서 모든 패키지를 제거하는** 지침을 따르십시오.

### 예 5.2. 선택한 프로필에서 패키지 제거

다음은 **php:7.3** 모듈 스트림의 **devel** 프로필에 속하는 패키지 및 해당 종속성을 제거하는 방법의 예입니다.



#### 참고

이 예제의 출력은 간결성을 위해 편집되었습니다. 실제 출력에는 여기에 표시된 것보다 더 많은 정보가 포함될 수 있습니다.

## 절차

1. 사용 가능한 모든 프로필을 포함하여 **php:7.3** 모듈 스트림을 설치합니다.

```
# 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. 설치된 프로필을 검사합니다.

```

$ 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

```

모든 프로파일은 출력에 표시된 대로 설치됩니다.

3. **devel** 프로파일에서 패키지 제거:

```
# 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. 제거 후 설치된 프로필을 검사합니다.

```
$ 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
```

현재 **devel** 을 제외한 모든 프로파일이 설치되어 있습니다.

### 5.3. 콘텐츠 제거를 위한 명령

다음은 Red Hat Enterprise Linux 8에서 콘텐츠를 제거하는 데 일반적으로 사용되는 명령입니다.

명령	설명
<b>yum remove</b> <i>패키지</i>	패키지를 제거합니다.
<b>yum module remove</b> <i>module-name:stream/profile</i>	설치된 프로필에서 패키지를 제거합니다.
<b>yum module remove --all</b> <i>module-name:stream</i>	활성 스트림에서 모든 패키지를 제거합니다.
<b>yum module reset</b> <i>module-name</i>	모듈을 초기 상태로 재설정합니다.
<b>yum module disable</b> <i>module-name</i>	모듈 및 모든 스트림을 비활성화합니다.

## 6장. 애플리케이션 스트림 콘텐츠 버전 관리

AppStream 리포지토리의 콘텐츠는 모듈 스트림에 해당하는 여러 버전에서 사용할 수 있습니다.

다음 섹션에서는 기존 활성화된 모듈 스트림을 변경할 때 수행해야 하는 작업을 설명합니다.

- 모듈식 종속 항목 및 스트림 변경 사항.
- 모듈식 종속성 및 수정되지 않은 종속성의 상호 작용.
- 모듈을 초기 상태로 재설정.
- 모듈 및 모든 스트림을 완전히 비활성화합니다.
- 모듈 의 이후 스트림 으로 전환합니다.
- 모듈 기본 스트림을 재정의합니다.

### 6.1. 모듈식 종속성 및 스트림 변경 사항

일반적으로 콘텐츠를 제공하는 패키지는 추가 패키지에 따라 다르며 일반적으로 원하는 종속성 버전을 지정합니다. 모듈에 포함된 패키지의 경우 이 메커니즘도 적용되지만 패키지 및 특정 버전을 모듈로 그룹화하고 스트림으로 더 많은 제약 조건을 제공합니다. 또한 모듈 스트림은 포함된 패키지에 관계없이 다른 모듈 스트림에 대한 종속성을 선언할 수 있습니다.

패키지 또는 모듈이 적용된 작업 후에는 기본 설치된 모든 패키지의 전체 종속성 트리가 패키지가 선언한 모든 조건을 충족해야 합니다. 또한 모든 모듈 스트림 종속성을 충족해야 합니다.

결과는 다음과 같습니다.

- 모듈 스트림을 활성화하려면 추가 모듈 스트림을 활성화해야 할 수 있습니다.
- 모듈 스트림 프로필을 설치하거나 스트림에서 패키지를 설치하려면 추가 모듈 스트림을 활성화하고 추가 패키지를 설치해야 할 수 있습니다.
- 모듈 스트림을 비활성화하면 다른 모듈 스트림을 비활성화해야 할 수 있습니다. 패키지가 자동으로 제거되지 않습니다.
- 패키지를 제거하려면 추가 패키지를 제거해야 할 수 있습니다. 이러한 패키지가 모듈에서 제공하는 경우 더 이상 이러한 스트림에서 설치된 패키지가 없는 경우에도 추가 설치를 위해 모듈 스트림이 활성화된 상태로 유지됩니다. 이렇게 하면 사용되지 않는 YUM 저장소의 동작이 미러링됩니다.



#### 중요

동일한 모듈의 다른 스트림이 이미 활성화된 경우 모듈 스트림을 활성화할 수 없습니다. 스트림을 전환하려면 다음 **스트림으로 전환**의 절차를 따르십시오. 또는 모듈을 재설정 한 다음 새 스트림을 활성화합니다.

다른 스트림으로 전환하기 전에 스트림에서 설치된 모든 패키지를 제거하면 시스템이 해당 패키지를 제공하지 않고 패키지를 설치할 수 있는 상태에 도달할 수 없습니다.

기술적으로 재설정 모듈은 설치된 패키지를 자동으로 변경하지 않습니다. 이전 스트림에서 제공한 패키지 및 해당 패키지에 종속된 패키지를 제거하는 작업은 명시적 수동 작업입니다.

## 6.2. 모듈식 및 비모형 종속성의 상호 작용

**모듈식 종속성**은 일반 RPM 종속 항목 위에 있는 추가 계층입니다. 모듈식 종속성은 리포지토리 간 가상 종속성과 유사하게 작동합니다. 즉, 다른 패키지를 설치하려면 RPM 종속성을 확인해야 할 뿐만 아니라 모듈식 종속성도 사전에 해결해야 합니다.

시스템은 명시적으로 변경하지 않는 한 모듈 및 스트림 선택을 유지합니다. 모듈식 패키지는 이 패키지를 제공하는 현재 활성화된 모듈 스트림에 포함된 업데이트를 받지만 다른 스트림에 포함된 버전으로 업그레이드하지는 않습니다.

## 6.3. 모듈 스트림 재설정

모듈을 재설정하는 것은 모든 스트림을 초기 상태로 반환하는 작업입니다(활성화 또는 비활성화됨). 모듈에 기본 스트림이 있는 경우 모듈을 재설정 후 이 스트림이 활성화됩니다.

### 절차

- 모듈 상태를 재설정합니다.

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

*module-name* 을 재설정할 모듈의 이름으로 바꿉니다.

이 모듈은 초기 상태로 반환됩니다. 활성화된 스트림 및 설치된 프로필에 대한 정보는 삭제되지만 설치되지 않은 콘텐츠는 삭제되지 않습니다.

## 6.4. 모듈의 모든 스트림 비활성화

기본 스트림이 있는 모듈에는 항상 하나의 스트림이 활성화됩니다. 모든 모듈 스트림의 콘텐츠에 액세스할 수 없는 경우 전체 모듈을 비활성화할 수 있습니다.

### 사전 요구 사항

- [활성 모듈 스트림의 개념](#)을 이해합니다.

### 절차

- 모듈을 비활성화합니다.

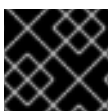
```
# yum module disable module-name
```

*module-name* 을 비활성화하려는 모듈의 이름으로 바꿉니다.

**yum** 명령은 확인을 요청한 다음 모든 스트림을 사용하여 모듈을 비활성화합니다. 모든 모듈 스트림이 비활성화됩니다. 설치된 콘텐츠가 제거되지 않습니다.

## 6.5. 이후 스트림으로 전환

이후 모듈 스트림으로 전환하면 모듈의 모든 패키지가 최신 버전으로 교체됩니다.



### 중요

이 절차는 사전 요구 사항 섹션에 설명된 조건에서만 가능합니다.



## 사전 요구 사항

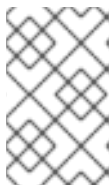
- 시스템이 완전히 업데이트됩니다.
- 시스템에 설치된 패키지가 리포지토리에서 사용할 수 있는 패키지보다 최신 상태가 아닙니다.

## 절차

1. 시스템에서 이후 스트림으로 전환할 준비가 되었는지 확인합니다.

```
# yum distro-sync
```

이 명령은 *필요한 작업 없음* 메시지로 끝나야 합니다. *완료되었습니다!* 이 대신 변경 사항을 제안하고 확인을 요청하면 이러한 변경 사항을 신중하게 검토하고 진행할지 여부를 고려하십시오. 필요한 경우 **yum distro-sync** 명령을 반복적으로 실행합니다. 또는 제안된 변경을 거부하고 시스템이 *Nothing*을 반환하는 상태로 시스템을 수동으로 수정할 수 있습니다. *완료되었습니다!*



### 참고

스트림을 전환하기 전에 **yum distro-sync** 결과를 확인하면 이 절차의 마지막 단계와 동일한 명령이 필요하므로 스트림 전환과 관련이 없는 시스템을 변경하지 않습니다.

2. 활성 스트림을 다음 스트림으로 변경합니다.

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

3. 설치된 패키지를 동기화하여 스트림 간 변경을 수행합니다.

```
# yum distro-sync
```

이 조치에서 스트림 외부의 콘텐츠에 대한 변경을 제안하는 경우 신중하게 검토하십시오.



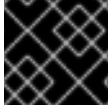
### 참고

- 설치된 특정 패키지가 이전 스트림에 종속되고 이후 스트림에 호환되는 버전이 없는 경우 **yum** 은 종속성 충돌을 보고합니다. 이 경우 **--allowrasing** 옵션을 사용하여 이러한 패키지를 제거할 수 있습니다. 이러한 패키지는 누락된 종속성으로 인해 이후 스트림과 함께 설치할 수 없기 때문입니다.
- **Perl** 모듈을 전환할 때 기본 RHEL 8 설치의 특정 패키지가 **Perl 5.26** 에 의존하기 때문에 항상 **--allowrasing** 옵션을 사용해야 합니다.
- 해석된 언어의 바이너리 확장(일반적으로 C 또는 C++로 작성된)은 새 스트림이 활성화된 후 다시 설치해야 합니다. 예를 들어 **ruby** 모듈에서 **gem** 명령에 의해 설치된 특정 패키지, **nodejs** 모듈의 **npm** 명령, **perl** 모듈의 **cpan** 명령 또는 **php** 모듈의 **pecl** 명령을 다시 설치해야 합니다. 자세한 내용은 [RHEL 8에서 Ruby 스트림을 전환하는 방법을 참조하십시오](#).

또는 현재 스트림에서 설치된 모든 모듈 콘텐츠를 제거하고 모듈을 재설정 한 다음 새 스트림을 설치합니다.

## 6.6. 모듈 기본 스트림 덮어쓰기

기본적으로 **YUM** 유틸리티는 모듈이 포함된 리포지토리에 정의된 모듈 기본 스트림을 사용합니다. **/etc/dnf/modules.defaults.d/** 디렉터리에서 기본 스트림을 재정의할 수 있습니다.



### 중요

모듈 스트림의 **라이프 사이클**을 항상 고려하십시오.

### 사전 요구 사항

- **활성 모듈 스트림의 개념을 이해하고 있습니다.**

### 절차

1. **/etc/dnf/modules.defaults.d/** drop-in 디렉터리에 YAML 구성 파일을 만듭니다.

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

앞의 출력은 이 작성 시 **postgresql** 모듈에 대한 기본 정의를 나타냅니다.

### 예 6.1. 원래 기본값이 있는 postgresql 모듈의 예

다음은 **postgresql** 모듈의 스트림 **13** 을 기본 스트림으로 구성하는 방법의 예입니다.

1. **postgresql** 모듈을 검사합니다.

```
# 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. 기본 스트림을 **13** 으로 설정하려면 **/etc/dnf/modules.defaults.d/postgresql.yaml** 파일에서 다음 YAML 파일 구성을 구현합니다.

```

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

```

3. **postgresql** 모듈을 다시 검사합니다.

```

# 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

```