



# Red Hat Enterprise Linux 9

## DNF 툴을 사용하여 소프트웨어 관리

Red Hat Enterprise Linux 9에서 DNF를 사용한 소프트웨어 관리 가이드



# Red Hat Enterprise Linux 9 DNF 툴을 사용하여 소프트웨어 관리

---

Red Hat Enterprise Linux 9에서 DNF를 사용한 소프트웨어 관리 가이드

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

## 법적 공지

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Managing\_software\_with\_the\_DNF\_tool.ent file |.

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

이 문서에서는 Red Hat Enterprise Linux 9의 DNF 툴을 사용하여 AppStream 및 BaseOS 리포지토리의 검색, 검색, 설치, 사용에 대해 설명합니다. 여기에는 모듈, 애플리케이션 스트림 및 프로필을 사용하는 방법에 대한 설명이 포함됩니다.

## 차례

보다 포괄적 수용을 위한 오픈 소스 용어 교체 .....	4
RED HAT 문서에 관한 피드백 제공 .....	5
1장. RED HAT ENTERPRISE LINUX 9의 소프트웨어 관리 툴 .....	6
2장. RHEL 9의 콘텐츠 배포 .....	7
2.1. 리포지토리 .....	7
2.2. APPLICATION STREAMS .....	7
2.3. 모듈 .....	8
2.4. 모듈 스트림 .....	8
2.5. 모듈 프로필 .....	9
3장. DNF 구성 .....	10
3.1. 현재 DNF 구성 보기 .....	10
3.2. DNF 기본 옵션 설정 .....	10
3.3. DNF 플러그인 사용 .....	10
3.3.1. DNF 플러그인 관리 .....	10
3.3.2. DNF 플러그인 활성화 및 비활성화 .....	10
4장. RHEL 9 콘텐츠 검색 .....	12
4.1. 소프트웨어 패키지 검색 .....	12
4.2. 소프트웨어 패키지 나열 .....	12
4.3. 리포지토리 나열 .....	13
4.4. 패키지 정보 표시 .....	13
4.5. 패키지 그룹 나열 .....	14
4.6. 사용 가능한 모듈 나열 .....	14
4.7. DNF 입력에서 전역 표현식 지정 .....	15
4.8. 추가 리소스 .....	15
5장. RHEL 9 콘텐츠 설치 .....	16
5.1. 패키지 설치 .....	16
5.2. 패키지 그룹 설치 .....	16
5.3. 설치된 콘텐츠 실행 .....	17
5.4. 추가 리소스 .....	17
6장. RHEL 9 콘텐츠 업데이트 .....	18
6.1. 업데이트 확인 .....	18
6.2. 패키지 업데이트 .....	18
6.3. 보안 관련 패키지 업데이트 .....	19
7장. RHEL 9에서 소프트웨어 업데이트 자동화 .....	20
7.1. DNF 자동 설치 .....	20
7.2. DNF 자동 구성 파일 .....	20
7.3. DNF 자동 활성화 .....	21
7.4. DNF-AUTOMATIC 패키지에 포함된 SYSTEMD 타이머 장치 개요 .....	22
8장. RHEL 9 콘텐츠 제거 .....	24
8.1. 설치된 패키지 제거 .....	24
8.2. 패키지 그룹 제거 .....	24
8.3. 추가 리소스 .....	24
9장. 패키지 관리 내역 처리 .....	26
9.1. 트랜잭션 나열 .....	26

9.2. 트랜잭션 복원	26
9.3. 트랜잭션 반복	27
<b>10장. 사용자 정의 소프트웨어 리포지토리 관리</b> .....	<b>28</b>
10.1. DNF 리포지토리 옵션 설정	28
10.2. DNF 리포지토리 추가	28
10.3. DNF 리포지토리 활성화	29
10.4. DNF 리포지토리 비활성화	29
<b>부록 A. DNF 명령 목록</b> .....	<b>31</b>
A.1. RHEL 9의 콘텐츠를 나열하는 명령	31
A.2. RHEL 9에서 콘텐츠를 설치하기 위한 명령	32
A.3. RHEL 9에서 콘텐츠를 제거하는 명령	33



## 보다 포괄적 수용을 위한 오픈 소스 용어 교체

Red Hat은 코드, 문서, 웹 속성에서 문제가 있는 용어를 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

문서 개선을 위한 의견을 보내 주십시오. 문서를 개선할 수 있는 방법에 대해 알려주십시오.

- 특정 문구에 대한 간단한 의견 작성 방법은 다음과 같습니다.
  1. 문서가 *Multi-page HTML* 형식으로 표시되는지 확인합니다. 또한 문서 오른쪽 상단에 **피드백** 버튼이 있는지 확인합니다.
  2. 마우스 커서를 사용하여 주석 처리하려는 텍스트 부분을 강조 표시합니다.
  3. 강조 표시된 텍스트 아래에 표시되는 **피드백** 추가 팝업을 클릭합니다.
  4. 표시된 지침을 따릅니다.
- Bugzilla를 통해 피드백을 제출하려면 새 티켓을 생성하십시오.
  1. [Bugzilla](#) 웹 사이트로 이동하십시오.
  2. 구성 요소로 **문서**를 사용합니다.
  3. **설명** 필드에 문서 개선을 위한 제안 사항을 기입하십시오. 관련된 문서의 해당 부분 링크를 알려주십시오.
  4. **버그** 제출을 클릭합니다.

## 1장. RED HAT ENTERPRISE LINUX 9의 소프트웨어 관리 툴

Red Hat Enterprise Linux 9에서 소프트웨어 설치에 **DNF** 도구에 의해 보장됩니다. Red Hat은 RHEL의 이전 주요 버전과의 일관성 유지를 위해 **yum** 용어 사용을 계속 지원합니다. **dnf** 대신 **yum** 을 입력하면 둘 다 호환성을 위한 별칭이므로 명령이 예상대로 작동합니다.



### 참고

RHEL 8 및 RHEL 9는 **DNF** 를 기반으로 하지만 RHEL 7에서 사용되는 **YUM** 과 호환됩니다.

## 2장. RHEL 9의 콘텐츠 배포

다음 섹션에서는 Red Hat Enterprise Linux 9의 소프트웨어 배포에 대한 개요를 설명합니다.

- [섹션 2.1. "repositories"](#) 는 Red Hat Enterprise Linux 9의 콘텐츠가 BaseOS 및 AppStream으로 분할되는 방법을 설명합니다.
- [섹션 2.2. "application Streams"](#) 는 Application Streams의 개념을 설명합니다.
- [섹션 2.3. "modules"](#) 는 모듈 개념을 설명합니다.
- [섹션 2.4. "모듈 스트림"](#) 는 버전별 콘텐츠 조직을 설명합니다.
- [섹션 2.5. "module profiles"](#) 는 목적별로 콘텐츠 조직을 설명합니다.

### 2.1. 리포지토리

RHEL 9 콘텐츠는 다음 두 가지 주요 리포지토리를 통해 배포됩니다. **BaseOS** 및 **AppStream**. BaseOS 및 AppStream 콘텐츠 세트는 기본 RHEL 설치에 필요하며 모든 RHEL 서브스크립션에서 사용할 수 있습니다. 설치 지침은 [표준 RHEL 9 설치](#) 문서를 참조하십시오.

#### BaseOS

BaseOS 리포지토리의 콘텐츠는 모든 설치의 기반이 되는 기본 OS 기능의 코어 세트를 제공하는 데 사용됩니다. 이 콘텐츠는 RPM 형식으로 제공되며 이전 Red Hat Enterprise Linux 릴리스와 유사한 지원 조건이 적용됩니다.

#### AppStream

AppStream 리포지토리의 콘텐츠에는 다양한 워크로드 및 사용 사례를 지원하는 추가 사용자 공간 애플리케이션, 런타임 언어 및 데이터베이스가 포함됩니다.

#### CodeReady Linux Builder

CodeReady Linux Builder 리포지토리는 모든 RHEL 서브스크립션과 함께 사용할 수 있습니다. 이는 개발자가 사용할 수 있는 추가 패키지를 제공합니다. CodeReady Linux Builder 리포지토리에 포함된 패키지는 지원되지 않습니다.

#### 추가 리소스

- [표준 RHEL 9 설치 수행](#)
- [패키지 매니페스트](#)

### 2.2. APPLICATION STREAMS

여러 사용자 공간 구성 요소의 버전은 Application Streams로 제공되며 핵심 운영 체제 패키지보다 자주 업데이트됩니다. 따라서 플랫폼 또는 특정 배포의 기본 안정성에 영향을 주지 않고 RHEL을 사용자 정의할 수 있는 유연성이 향상됩니다.

각 Application Stream 구성 요소에는 RHEL 9와 동일하거나 특정 애플리케이션에 더 적합한 라이프사이클이 있습니다. RHEL 라이프 사이클 정보는 [Red Hat Enterprise Linux 라이프 사이클](#) 및 [Red Hat Enterprise Linux Application Streams 라이프 사이클](#)에서 참조하십시오.

Application Streams는 다음 형식으로 제공됩니다.

- 친숙한 RPM 형식

- RPM 형식 확장 모듈
- 소프트웨어 컬렉션으로
- 예를 들어 Flatpaks입니다.

RHEL 9는 기존 **dnf install** 명령을 사용하여 **RPM 패키지로 간단히 설치**할 수 있는 초기 Application Stream 버전을 제공하여 애플리케이션 스트림 환경을 개선합니다.



**참고**

RPM 형식의 특정 초기 Application Streams는 Red Hat Enterprise Linux 9보다 라이프사이클이 짧습니다.

일부 추가 Application Stream 버전은 향후 마이너 RHEL 9 릴리스에서 라이프사이클이 짧은 모듈로 배포됩니다.

설치하려는 Application Stream 버전을 항상 결정하고 먼저 [Red Hat Enterprise Linux Application Stream Lifecycle](#) 을 검토하십시오.



**참고**

일부 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 Application Streams 라이프 사이클](#)
- [Red Hat Enterprise Linux 9: 애플리케이션 호환성 가이드](#)
- [패키지 매니페스트](#)

**2.3. 모듈**

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.



**참고**

모듈은 향후 RHEL 9 릴리스에서 제공될 예정입니다.

**2.4. 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. 사용자 작업에서 명시적으로 활성화하면 스트림이 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.

특정 모듈 스트림은 다른 모듈 스트림에 따라 달라질 수 있습니다.

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

- 필수 기능 및 해당 기능을 지원하는 구성 요소 버전
- 호환성
- 라이프 사이클 기간 및 업데이트 계획

구성 요소별 변경 사항은 [릴리스 노트](#) 를 참조하십시오.

## 2.5. 모듈 프로필

프로필은 서버, 클라이언트, 개발, 최소 설치 또는 기타와 같은 특정 사용 사례에 대해 함께 설치할 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 none을 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.

## 3장. DNF 구성

DNF 및 관련 유틸리티의 구성 정보는 `/etc/dnf/dnf.conf` 파일에 저장됩니다. 이 파일에는 필수 `[main]` 섹션이 포함되어 있으며 이는 글로벌 효과가 있는 DNF 옵션을 설정할 수 있습니다.

다음 섹션에서는 다음 방법을 설명합니다.

- 현재 DNF 구성을 확인합니다.
- DNF `[main]` 옵션을 설정합니다.
- DNF 플러그인을 사용합니다.

### 3.1. 현재 DNF 구성 보기

다음 절차에서는 현재 DNF 구성을 표시하는 방법을 설명합니다.

절차

- `/etc/dnf/dnf.conf` 파일의 `[main]` 섹션에 지정된 글로벌 DNF 옵션의 현재 값을 표시하려면 다음을 사용합니다.

```
# dnf config-manager --dump
```

### 3.2. DNF 기본 옵션 설정

`/etc/dnf/dnf.conf` 구성 파일에는 하나의 `[main]` 섹션이 포함되어 있습니다. 이 섹션의 키-값 쌍은 DNF 가리표지토리를 작동하고 처리하는 방식에 영향을 미칩니다.

`[main]` 섹션 아래에 `/etc/dnf/dnf.conf` 의 추가 옵션을 추가할 수 있습니다.

사용 가능한 `[main]` 옵션의 전체 목록은 `dnf.conf(5)` 매뉴얼 페이지의 `[main] OPTIONS` 섹션을 참조하십시오.

### 3.3. DNF 플러그인 사용

DNF 는 작업을 확장하고 개선하는 플러그인을 제공합니다. 기본적으로 특정 플러그인이 설치되어 있습니다.

다음 절차에서는 DNF 플러그인을 활성화, 구성 및 비활성화하는 방법을 설명합니다.

#### 3.3.1. DNF 플러그인 관리

플러그인 구성 파일에는 항상 `enabled=` 옵션이 `dnf` 명령을 실행할 때 플러그인이 활성화되어 있는지 여부를 제어하는 `[main]` 섹션이 포함되어 있습니다. 이 옵션이 없는 경우 파일에 수동으로 추가할 수 있습니다.

설치된 모든 플러그인에는 `/etc/dnf/plugins/` 디렉터리에 고유한 구성 파일이 있을 수 있습니다. 이러한 파일에서 플러그인 특정 옵션을 활성화하거나 비활성화할 수 있습니다.

#### 3.3.2. DNF 플러그인 활성화 및 비활성화

DNF 도구에서는 기본적으로 플러그인이 로드됩니다.

다음 절차에서는 DNF 플러그인 로드를 수정하고 특정 DNF 플러그인을 활성화 또는 비활성화하는 방법을 설명합니다.

## 절차

- DNF 플러그인 로드를 비활성화하거나 활성화하려면 **plugins=** 로 시작하는 행이 **/etc/dnf/dnf.conf** 파일의 **[main]** 섹션에 있는지 확인하십시오.

1. DNF 플러그인 로드를 비활성화하려면 **plugins=** 값을 **0** 으로 설정합니다.



### 중요

모든 플러그인을 비활성화하는 것은 권장되지 않습니다. 특정 플러그인은 중요한 DNF 서비스 및 명령을 제공합니다. 특히 **product-id** 및 **subscription-manager** 플러그인은 인증서 기반 CDN( **Content Delivery Network** )을 지원합니다. 전역적으로 플러그인을 비활성화하면 편리한 옵션으로 제공되며 DNF 에서 잠재적인 문제를 진단하는 것이 좋습니다.

2. DNF 플러그인 로드를 활성화하려면 **plugins=** 값을 **1** 로 설정합니다.

- 특정 명령에 대해 모든 DNF 플러그인을 비활성화하려면 **--noplugins** 옵션을 명령에 추가합니다. 예를 들어 **update** 명령에 대해 DNF 플러그인을 비활성화하려면 다음을 실행합니다.

```
# dnf --noplugins update
```

- 단일 명령에 대해 특정 DNF 플러그인을 비활성화하려면 **--disableplugin=plugin-name** 옵션을 명령에 추가합니다. 예를 들어 **update** 명령에 대해 특정 DNF 플러그인을 비활성화하려면 다음을 수행합니다.

```
# dnf update --disableplugin=plugin-name
```

*plugin-name* 을 플러그인 이름으로 교체합니다.

- 단일 명령에 특정 DNF 플러그인을 활성화하려면 **--enableplugin=plugin-name** 옵션을 명령에 추가합니다. 예를 들어 **update** 명령에 대해 특정 DNF 플러그인을 비활성화하려면 다음을 수행합니다.

```
# dnf update --enableplugin=plugin-name
```

*plugin-name* 을 플러그인 이름으로 교체합니다.

## 4장. RHEL 9 콘텐츠 검색

다음 섹션에서는 Red Hat Enterprise Linux 9의 AppStream 및 BaseOS 리포지토리에서 콘텐츠를 찾고 검사하는 방법을 설명합니다.

- [섹션 4.1. "소프트웨어 패키지 검색"](#) 원하는 콘텐츠를 제공하는 패키지를 검색하는 방법을 설명합니다.
- [섹션 4.2. "소프트웨어 패키지 목록"](#) 는 설치 및 사용 가능한 패키지를 나열하는 방법을 설명합니다.
- [섹션 4.3. "리포지토리 목록"](#) 는 활성화된 리포지토리 및 비활성화 리포지토리를 나열하는 방법을 설명합니다.
- [섹션 4.4. "패키지 정보 표시"](#) 는 사용 가능한 패키지에 대한 정보를 표시하는 방법을 설명합니다.
- [섹션 4.5. "패키지 그룹 나열"](#) 는 설치 및 사용 가능한 패키지 그룹을 나열하는 방법을 설명합니다.
- [섹션 4.6. "사용 가능한 모듈 목록"](#) 는 사용 가능한 모듈을 나열하고 이에 대한 세부 정보를 찾는 방법을 설명합니다.
- [섹션 4.7. "dnf 입력에서 글로벌 표현식 확인"](#) 는 전역 표현식이 의도한 대로 **dnf** 에 전달되도록 하는 방법을 설명합니다.

### 4.1. 소프트웨어 패키지 검색

이 섹션에서는 특정 애플리케이션 또는 기타 콘텐츠를 제공하는 패키지를 찾는 데 필요한 단계에 대해 설명합니다.

#### 절차

- 패키지를 검색하려면 다음을 사용합니다.

```
# dnf search term
```

*term* 을 패키지와 관련된 용어로 바꿉니다.

**dnf search** 명령은 패키지의 이름 및 요약 내에서 용어를 반환합니다. 이렇게 하면 검색 속도가 빨라지고 이름을 모르는 패키지를 검색할 수 있지만 관련 용어를 알고 있습니다.

- 패키지 설명 내에 용어 일치를 포함하려면 다음을 사용합니다.

```
# dnf search --all term
```

패키지 이름, 요약 또는 설명에서 검색할 용어로 용어를 바꿉니다.

**dnf search --all** 명령을 사용하면 더 포괄적이지만 느린 검색을 수행할 수 있습니다.

### 4.2. 소프트웨어 패키지 나열

다음 절차에서는 **dnf** 를 사용하여 사용 가능한 패키지를 나열하는 방법을 설명합니다.

#### 절차



- 설치 및 사용 가능한 모든 패키지에 대한 정보를 나열하려면 다음을 사용합니다.

```
# dnf list --all
```

- 시스템에 설치된 모든 패키지를 나열하려면 다음을 사용합니다.

```
# dnf list --installed
```

또는 다음을 수행합니다.

```
# dnf repoquery --installed
```

- 설치할 수 있는 활성화된 모든 리포지토리의 모든 패키지를 나열하려면 다음을 사용합니다.

```
# dnf list --available
```

또는 다음을 수행합니다.

```
# dnf repoquery
```

전역 식을 인수로 추가하여 결과를 필터링할 수 있습니다. 자세한 내용은 [dnf 입력에서 전역 식 지정](#)을 참조하십시오.

### 4.3. 리포지토리 나열

다음 절차에서는 **dnf** 를 사용하여 리포지토리를 나열하는 방법을 설명합니다.

#### 절차

- 시스템에서 활성화된 모든 리포지토리를 나열하려면 다음을 사용합니다.

```
# dnf repolist
```

- 시스템의 비활성화 리포지토리를 모두 나열하려면 다음을 사용합니다.

```
# dnf repolist --disabled
```

- 활성화된 리포지토리와 비활성화된 리포지토리를 모두 나열하려면 다음을 사용합니다.

```
# dnf repolist --all
```

- 리포지토리에 대한 추가 정보를 나열하려면 다음을 사용합니다.

```
# dnf repoinfo
```

ID 또는 리포지토리 이름을 인수로 전달하거나 전역 식을 추가하여 결과를 필터링할 수 있습니다. 자세한 내용은 [dnf 입력에서 전역 식 지정](#)을 참조하십시오.

### 4.4. 패키지 정보 표시

다음 절차에서는 **dnf** 를 사용하여 패키지 정보를 표시하는 방법을 설명합니다.

## 절차

- 사용 가능한 패키지에 대한 정보를 표시하려면 다음을 사용합니다.

```
# dnf info package-name
```

*package-name* 을 패키지 이름으로 바꿉니다.

또는 다음을 수행합니다.

```
# dnf repoquery --info package-name
```

*package-name* 을 패키지 이름으로 바꿉니다.

- 시스템에 설치된 하나 이상의 패키지에 대한 정보를 표시하려면 다음을 사용합니다.

```
# dnf repoquery --info --installed package-name
```

*package-name* 을 패키지 이름으로 바꿉니다.

전역 식을 인수로 추가하여 결과를 필터링할 수 있습니다. 자세한 내용은 [dnf 입력에서 전역 식 지정을 참조하십시오](#).

## 4.5. 패키지 그룹 나열

다음 절차에서는 **dnf** 를 사용하여 패키지 그룹을 나열하는 방법을 설명합니다.

- 설치 및 사용 가능한 그룹의 수를 보려면 다음을 사용하십시오.

```
# dnf group summary
```

- 설치 및 사용 가능한 모든 그룹을 나열하려면 다음을 사용하십시오.

```
# dnf group list
```

**dnf group list** 명령(**--hidden**, **--available**)에 대한 명령줄 옵션을 추가하여 결과를 필터링할 수 있습니다. 사용 가능한 자세한 옵션은 도움말 페이지를 참조하십시오.

- 특정 그룹에 포함된 필수 및 선택적 패키지를 나열하려면 다음을 사용합니다.

```
# dnf group info group-name
```

*group-name* 을 그룹 이름으로 바꿉니다.

전역 식을 인수로 추가하여 결과를 필터링할 수 있습니다. 자세한 내용은 [dnf 입력에서 전역 식 지정을 참조하십시오](#).

## 4.6. 사용 가능한 모듈 나열

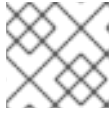
다음 절차에서는 사용 가능한 모듈과 **dnf** 를 사용하는 세부 정보를 찾는 방법을 설명합니다.

## 절차

- 시스템에서 사용할 수 있는 모듈 스트림을 나열하려면 다음을 수행합니다.

```
# dnf module list
```

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



#### 참고

모듈은 향후 RHEL 9 릴리스에서 제공될 예정입니다.

#### 추가 리소스

- [모듈](#)
- [모듈 스트림](#)
- [모듈 프로필](#)

## 4.7. DNF 입력에서 전역 표현식 지정

**dnf** 명령을 사용하면 하나 이상의 **전역 표현식**을 인수로 추가하여 결과를 필터링할 수 있습니다. 전역 표현식은 **dnf** 명령에 인수로 전달할 때 이스케이프해야 합니다.

다음 절차에서는 전역 표현식이 의도한 대로 **dnf**에 전달되도록 하는 두 가지 방법을 설명합니다.

#### 절차

- 전체 전역 표현식을 두 번 따옴표로 묶거나 한 번 인용하십시오.

```
# dnf provides "*/file-name"
```

*file-name* 을 파일 이름으로 바꿉니다.

*파일* 이름은 원하는 결과를 제공하기 위해 / 또는 \* 문자 시퀀스로 선행되어야 합니다.

- 뒤에 백슬래시(\) 문자를 사용하여 와일드카드 문자를 이스케이프합니다.

```
# dnf provides \*/file-name
```

*file-name* 을 파일 이름으로 바꿉니다.

## 4.8. 추가 리소스

- [RHEL 9의 콘텐츠를 나열하는 명령](#)

## 5장. RHEL 9 콘텐츠 설치

다음 섹션에서는 Red Hat Enterprise Linux 9의 콘텐츠를 설치하는 방법을 설명합니다.

- [섹션 5.1. "패키지 설치"](#)에는 패키지 설치 단계가 포함되어 있습니다.
- [섹션 5.2. "패키지 그룹 설치"](#) 패키지 그룹을 설치하는 방법을 설명합니다.
- [섹션 5.3. "설치된 콘텐츠 실행"](#)은 RHEL 9 설치된 콘텐츠를 실행하는 데 필요한 세부 정보를 제공합니다.

### 5.1. 패키지 설치

다음 절차에서는 **dnf** 를 사용하여 패키지를 설치하는 방법을 설명합니다.

#### 절차

- 패키지를 설치합니다.

```
# dnf install package-name
```

*package-name* 을 패키지 이름으로 바꿉니다.

- 여러 패키지와 해당 종속 항목을 동시에 설치하려면 다음을 사용합니다.

```
# dnf install package-name-1 package-name-2
```

*package-name-1* 및 *package-name-2* 를 패키지 이름으로 교체합니다.

- *다중lib* 시스템(AMD64, Intel 64 시스템)에 패키지를 설치할 때 패키지 이름에 추가하여 패키지의 아키텍처를 지정할 수 있습니다.

```
# dnf install package-name.arch
```

*package-name.arch* 를 패키지의 이름 및 아키텍처로 바꿉니다.

- 설치하려는 바이너리의 이름을 알고 있지만 패키지 이름은 아닌 경우 바이너리의 경로를 인수로 사용할 수 있습니다.

```
# dnf install /usr/sbin/binary-file
```

*/usr/sbin/binary-file* 을 바이너리 파일의 경로로 바꿉니다.

**DNF** 는 패키지 목록을 검색하고 */usr/sbin/binary-file* 을 제공하는 패키지를 찾은 다음 설치 여부를 묻는 메시지를 표시합니다.

- 로컬 디렉터리에서 이전에 다운로드한 패키지를 설치하려면 다음을 사용합니다.

```
# dnf install /path/
```

*/path/* 를 패키지 경로로 바꿉니다.

### 5.2. 패키지 그룹 설치

다음 절차에서는 그룹 이름 또는 **dnf** 를 사용하여 groupID로 패키지 그룹을 설치하는 방법을 설명합니다.

#### 절차

- 그룹 이름으로 패키지 그룹을 설치하려면 다음을 사용합니다.

```
# dnf group install group-name
```

*group-name* 을 그룹 또는 환경 그룹의 전체 이름으로 바꿉니다.

- groupID로 패키지 그룹을 설치하려면 다음을 사용합니다.

```
# dnf group install groupID
```

*groupID* 를 그룹의 ID로 바꿉니다.

### 5.3. 설치된 콘텐츠 실행

일반적으로 RHEL 9 리포지토리에서 콘텐츠를 설치한 후 새 명령이 활성화됩니다. RPM 패키지 또는 RPM 패키지에서 명령을 시작한 경우, 명령을 사용한 경험이 다를 수 없습니다.

#### 절차

- 새 명령을 직접 실행하려면 다음을 수행하십시오.

```
$ command
```

### 5.4. 추가 리소스

- **dnf(8)** 도움말 페이지
- [RHEL 9에서 콘텐츠를 설치하기 위한 명령](#)

## 6장. RHEL 9 콘텐츠 업데이트

DNF 를 사용하면 시스템에 보류 중인 업데이트가 있는지 확인할 수 있습니다. 업데이트가 필요한 패키지를 나열하고 단일 패키지, 여러 패키지 또는 모든 패키지를 한 번에 업데이트하도록 선택할 수 있습니다. 업데이트하는 패키지에 종속성이 있는 경우 해당 패키지도 업데이트됩니다.

다음 섹션에서는 DNF 를 사용하여 Red Hat Enterprise Linux 9의 콘텐츠를 업데이트하는 방법을 설명합니다.

- [섹션 6.1. "업데이트를 확인"](#) 사용 가능한 업데이트를 확인하는 방법을 설명합니다.
- [섹션 6.2. "패키지 업데이트"](#) 는 단일 패키지, 패키지 그룹 또는 모든 패키지 및 해당 종속성을 업데이트하는 방법을 설명합니다.
- [섹션 6.3. "보안 관련 패키지 업데이트"](#) 보안 업데이트 적용 방법을 설명합니다.

### 6.1. 업데이트 확인

다음 절차에서는 **dnf** 를 사용하여 시스템에 설치된 패키지에 대해 사용 가능한 업데이트를 확인하는 방법을 설명합니다.

#### 절차

- 다음 명령을 실행하여 시스템에 설치된 패키지가 사용 가능한 업데이트가 있는지 확인합니다.

```
# dnf check-update
```

출력에는 사용 가능한 업데이트가 있는 패키지 목록 및 해당 종속 항목이 반환됩니다.

### 6.2. 패키지 업데이트

다음 절차에서는 **dnf** 를 사용하여 단일 패키지, 패키지 그룹 또는 모든 패키지와 해당 종속 항목을 업데이트하는 방법을 설명합니다.

#### 절차

- 모든 패키지 및 종속 항목을 업데이트하려면 다음을 사용합니다.

```
# dnf upgrade
```

- 단일 패키지를 업데이트하려면 다음을 사용합니다.

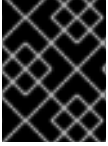
```
# dnf upgrade package-name
```

*package-name* 을 패키지 이름으로 바꿉니다.

- 패키지 그룹을 업데이트하려면 다음을 사용합니다.

```
# dnf group upgrade group-name
```

*group-name* 을 패키지 그룹의 이름으로 바꿉니다.



### 중요

업데이트를 커널에 적용할 때 **dnf upgrade** 또는 **dnf install** 명령을 사용 중인지 여부에 관계없이 항상 새 커널을 설치합니다.

## 6.3. 보안 관련 패키지 업데이트

다음 절차에서는 **dnf** 를 사용하여 보안 관련 패키지를 업데이트하는 방법을 설명합니다.

### 절차

- 보안 에라타가 있는 사용 가능한 최신 패키지로 업그레이드하려면 다음을 사용하십시오.

```
# dnf upgrade --security
```

- 마지막 보안 에라타 패키지로 업그레이드하려면 다음을 사용하십시오.

```
# dnf upgrade-minimal --security
```

## 7장. RHEL 9에서 소프트웨어 업데이트 자동화

패키지 업데이트를 자동으로 확인하고 다운로드하려면 **dnf-automatic** 패키지에서 제공하는 **DNF** 자동 도구를 사용할 수 있습니다.

**DNF Automatic** 은 `systemd` 타이머, `cron` 작업 및 기타 이러한 도구를 사용하여 자동 및 정기적으로 실행하는 데 적합한 **DNF** 에 대한 대체 명령줄 인터페이스입니다.

**DNF**는 필요에 따라 패키지 메타데이터를 자동으로 동기화하고, 사용 가능한 업데이트를 확인한 다음 툴 구성 방법에 따라 다음 작업 중 하나를 수행합니다.

- `exit`
- 업데이트된 패키지 다운로드
- 업데이트 다운로드 및 적용

그런 다음 작업 결과는 표준 출력 또는 이메일과 같은 선택한 메커니즘에 의해 보고됩니다.

다음 섹션에서는 Red Hat Enterprise Linux 9에서 소프트웨어 업데이트를 자동화하는 방법을 설명합니다.

- [섹션 7.1. "DNF 자동 설치"](#) **DNF** 자동 툴을 설치하는 방법을 설명합니다.
- [섹션 7.2. "DNF 자동 구성 파일"](#) **DNF** 자동 구성 파일 및 해당 섹션을 설명합니다.
- [섹션 7.3. "enabling DNF Automatic"](#) **DNF** 자동 툴을 활성화하는 방법을 설명합니다.
- [섹션 7.4. "dnf-automatic 패키지에 포함된 systemd 타이머 장치 개요"](#) 는 **dnf-automatic** `systemd` 타이머 장치를 나열합니다.

### 7.1. DNF 자동 설치

다음 절차에서는 **DNF** 자동 도구를 설치하는 방법을 설명합니다.

절차

- **dnf-automatic** 패키지를 설치합니다.

```
# dnf install dnf-automatic
```

검증

- **dnf-automatic** 패키지가 있는지 확인하여 설치에 성공했는지 확인합니다.

```
# rpm -qi dnf-automatic
```

### 7.2. DNF 자동 구성 파일

기본적으로 **DNF Automatic** 은 `/etc/dnf/automatic.conf` 를 구성 파일로 사용하여 동작을 정의합니다.

구성 파일은 다음 주제 섹션으로 구분됩니다.

- **[명령]** 섹션  
DNF 자동 작동 모드를 설정합니다.



- **[emitters]** 섹션  
DNF 자동 결과를 보고하는 방법을 정의합니다.
- **[command\_email]** 섹션  
이메일을 보내는 데 사용되는 외부 명령의 이메일 발송기 구성을 제공합니다.
- **[email]** 섹션  
이메일 발송기 구성을 제공합니다.
- **[base]** 섹션  
DNF의 기본 구성 파일의 설정을 재정의합니다.

`/etc/dnf/automatic.conf` 파일의 기본 설정에서 DNF는 사용 가능한 업데이트를 자동으로 점검하고, 해당 업데이트를 다운로드하고, 결과를 표준 출력으로 보고합니다.



#### 주의

**[commands]** 섹션의 작업 모드 설정은 **dnf-automatic.timer**를 제외한 모든 타이머 장치에 대해 `systemd` 타이머 장치에서 사용하는 설정에 의해 재정의됩니다.

#### 추가 리소스

- [dnf 자동 문서](#)
- `man dnf-automatic` man 페이지
- [dnf-automatic 패키지에 포함된 systemd 타이머 장치 개요](#)

## 7.3. DNF 자동 활성화

DNF 자동 실행하려면 항상 특정 `systemd` 타이머 장치를 활성화하고 시작해야 합니다. **dnf-automatic** 패키지에 제공된 타이머 장치 중 하나를 사용하거나 필요에 따라 자체 타이머 장치를 작성할 수 있습니다.

다음 절차에서는 DNF 자동 활성화 방법을 설명합니다.

#### 사전 요구 사항

- `/etc/dnf/automatic.conf` 구성 파일을 수정하여 DNF 자동 동작을 지정했습니다.

#### 절차

- 사용 가능한 업데이트를 다운로드하는 `systemd` 타이머 장치를 선택, 활성화 및 시작하려면 다음을 사용합니다.

```
# systemctl enable dnf-automatic-download.timer
```

```
# systemctl start dnf-automatic-download.timer
```

- 사용 가능한 업데이트를 다운로드하고 설치하는 `systemd` 타이머 장치를 선택, 활성화 및 시작하려면 다음을 사용합니다.

```
# systemctl enable dnf-automatic-install.timer
```

```
# systemctl start dnf-automatic-install.timer
```

- 사용 가능한 업데이트를 보고하는 `systemd` 타이머 장치를 선택, 활성화 및 시작하려면 다음을 사용합니다.

```
# systemctl enable dnf-automatic-notifyonly.timer
```

```
# systemctl start dnf-automatic-notifyonly.timer
```

- 사용 가능한 업데이트를 다운로드, 다운로드 및 설치 또는 보고서를 다운로드하는 `systemd` 타이머 장치를 선택, 활성화, 시작하려면 다음을 사용하십시오.

```
# systemctl enable dnf-automatic.timer
```

```
# systemctl start dnf-automatic.timer
```

- 선택적으로 `--now` 옵션을 사용하여 하나의 명령에서 `systemd` 타이머 장치를 선택, 활성화, 시작합니다. 예를 들면 다음과 같습니다.

```
# systemctl enable --now dnf-automatic-download.timer
```



#### 참고

명령줄에서 직접 또는 사용자 지정 스크립트에서 `/usr/bin/dnf-automatic` 파일을 실행하여 DNF 자동 실행을 실행할 수도 있습니다.

#### 검증

- 타이머가 활성화되어 있는지 확인합니다.

```
# systemctl status <systemd timer unit>
```

#### 추가 리소스

- [man dnf-automatic man 페이지](#)
- [dnf-automatic 패키지에 포함된 systemd 타이머 장치 개요](#)
- [dnf 자동 구성 파일](#)

## 7.4. DNF-AUTOMATIC 패키지에 포함된 SYSTEMD 타이머 장치 개요

`systemd` 타이머 단위는 우선 순위를 두고 업데이트를 다운로드하고 적용할 때 `/etc/dnf/automatic.conf` 구성 파일의 설정을 재정의합니다.

예를 들어 다음을 설정하는 경우:

```
download_updates = yes
```

`/etc/dnf/automatic.conf` 구성 파일에서 `dnf-automatic-notifyonly.timer` 장치를 활성화했으므로 패키지가 다운로드되지 않습니다.

`dnf-automatic` 패키지에는 다음과 같은 `systemd` 타이머 장치가 포함됩니다.

표 7.1. `dnf-automatic` 패키지에 포함된 `systemd` 타이머

타이머 장치	함수	<code>/etc/dnf/automatic.conf</code> 파일의 설정을 덮어씁니다.
<code>dnf-automatic-download.timer</code>	패키지를 캐시로 다운로드하여 업데이트할 수 있도록 합니다.  참고: 이 타이머 장치는 업데이트된 패키지를 설치하지 않습니다. 설치를 수행하려면 <b>dnf update</b> 명령을 실행해야 합니다.	있음
<code>dnf-automatic-install.timer</code>	업데이트된 패키지를 다운로드하여 설치합니다.	있음
<code>dnf-automatic-notifyonly.timer</code>	리포지토리 캐시를 최신 상태로 유지하고 사용 가능한 업데이트에 대해 알리는 리포지토리 데이터만 다운로드합니다.  참고: 이 타이머 장치는 업데이트된 패키지를 다운로드하거나 설치하지 않습니다.	있음
<code>dnf-automatic.timer</code>	업데이트를 다운로드하고 적용할 때 이 타이머의 동작은 <code>/etc/dnf/automatic.conf</code> 구성 파일의 설정에 의해 지정됩니다.  기본 동작은 <code>dnf-automatic-download.timer</code> 단위와 동일합니다. 패키지를 다운로드하지만 설치하지는 않습니다.	없음

#### 추가 리소스

- `man dnf-automatic` `man` 페이지
- [dnf 자동 구성 파일](#)

## 8장. RHEL 9 콘텐츠 제거

다음 섹션에서는 Red Hat Enterprise Linux 9의 내용을 제거하는 방법을 설명합니다.

- [섹션 8.1. "설치된 패키지 제거"](#)는 패키지 제거를 설명합니다.
- [섹션 8.2. "패키지 그룹 제거"](#) 패키지 그룹 제거를 설명합니다.

### 8.1. 설치된 패키지 제거

다음 절차에서는 `dnf` 를 사용하여 패키지를 제거하는 방법을 설명합니다.

절차

- 특정 패키지 및 사용되지 않는 종속 패키지를 모두 제거하려면 다음을 사용합니다.

```
# dnf remove package-name
```

`package-name` 을 패키지 이름으로 바꿉니다.

패키지는 다른 종속 패키지와 함께 제거됩니다.

- 여러 패키지와 사용되지 않는 종속 항목을 동시에 제거하려면 다음을 사용하십시오.

```
# dnf remove package-name-1 package-name-2
```

`package-name-1` 및 `package-name-2` 를 패키지 이름으로 교체합니다.



참고

DNF 는 종속 패키지를 제거하지 않고 패키지를 제거할 수 없습니다.

### 8.2. 패키지 그룹 제거

다음 절차에서는 그룹 이름 또는 `groupID`로 패키지를 제거하는 방법을 설명합니다.

절차

- 그룹 이름으로 패키지 그룹을 제거하려면 다음을 사용합니다.

```
# dnf group remove group-name
```

`group-name` 을 그룹의 전체 이름으로 바꿉니다.

- `groupID`로 패키지 그룹을 제거하려면 다음을 사용합니다.

```
# dnf group remove groupID
```

`groupID` 를 그룹의 ID로 바꿉니다.

### 8.3. 추가 리소스

- RHEL 9에서 콘텐츠를 제거하는 명령

## 9장. 패키지 관리 내역 처리

`dnf history` 명령을 사용하면 다음 정보를 확인할 수 있습니다.

- DNF 트랜잭션 타임라인
- 트랜잭션이 발생한 날짜 및 시간
- 트랜잭션의 영향을 받는 패키지 수
- 트랜잭션이 성공했는지 또는 중단되었는지 여부
- 트랜잭션 간에 RPM 데이터베이스가 변경된 경우

`dnf history` 명령을 사용하여 트랜잭션을 취소하거나 다시 실행할 수도 있습니다.

다음 섹션에서는 `dnf` 를 사용하여 Red Hat Enterprise Linux 9의 패키지 관리 이력을 처리하는 방법을 설명합니다.

- [섹션 9.1. "패키지 목록"](#)은 최신 트랜잭션, 선택한 패키지에 대한 최신 작업, 특정 트랜잭션의 세부 정보를 나열하는 방법을 설명합니다.
- [섹션 9.2. "Reverting transactions"](#)은 선택한 트랜잭션 또는 마지막 트랜잭션을 되돌리는 방법을 설명합니다.
- [섹션 9.3. "repeating transactions"](#)은 선택한 트랜잭션 또는 마지막 트랜잭션을 반복하는 방법을 설명합니다.

### 9.1. 트랜잭션 나열

다음 절차에서는 최신 DNF 트랜잭션, 선택한 패키지의 최신 작업 및 특정 트랜잭션의 세부 정보를 나열하는 방법을 설명합니다.

절차

- 모든 최신 DNF 트랜잭션 목록을 표시하려면 다음을 사용합니다.

```
# dnf history
```

- 선택한 패키지에 대한 모든 최신 작업 목록을 표시하려면 다음을 사용합니다.

```
# dnf history list package-name
```

*package-name* 을 패키지 이름으로 바꿉니다. 전역 표현식을 추가하여 명령 출력을 필터링할 수 있습니다. 자세한 내용은 [dnf 입력에서 전역 식 지정](#)을 참조하십시오.

- 특정 트랜잭션의 세부 정보를 표시하려면 다음을 사용합니다.

```
# dnf history info transactionID
```

*transactionID* 를 트랜잭션의 ID로 바꿉니다.

### 9.2. 트랜잭션 복원

다음 절차에서는 **dnf** 를 사용하여 선택한 트랜잭션 또는 마지막 트랜잭션을 되돌리는 방법을 설명합니다.

절차

- 특정 트랜잭션을 되돌리려면 다음을 사용합니다.

```
# dnf history undo transactionID
```

*transactionID* 를 트랜잭션의 ID로 바꿉니다.

- 마지막 트랜잭션을 되돌리려면 다음을 사용합니다.

```
# dnf history undo last
```

**dnf history undo** 명령은 트랜잭션 중에 수행한 단계만 되돌립니다. 트랜잭션이 새 패키지를 설치한 경우 **dnf** 기록이 이를 제거 취소합니다. 트랜잭션이 패키지를 제거하면 **dnf** 기록에서 이를 다시 설치합니다. **dnf history undo** 명령은 이전 패키지를 계속 사용할 수 있는 경우 업데이트된 모든 패키지를 이전 버전으로 다운그레이드하려고 시도합니다.

### 9.3. 트랜잭션 반복

다음 절차에서는 **dnf** 를 사용하여 선택한 트랜잭션 또는 마지막 트랜잭션을 반복하는 방법을 설명합니다.

절차

- 특정 트랜잭션을 반복하려면 다음을 사용합니다.

```
# dnf history redo transactionID
```

*transactionID* 를 트랜잭션의 ID로 바꿉니다.

- 마지막 트랜잭션을 반복하려면 다음을 사용합니다.

```
# dnf history redo last
```

**dnf** 기록 재실행 명령은 트랜잭션 중에 수행된 단계만 반복합니다.

## 10장. 사용자 정의 소프트웨어 리포지토리 관리

DNF 및 관련 유틸리티의 구성 정보는 `/etc/dnf/dnf.conf` 파일에 저장됩니다. 이 파일에는 리포지토리별 옵션을 설정할 수 있는 하나 이상의 `[repository]` 섹션이 포함되어 있습니다.

`/etc/yum .repo s.d/` 디렉토리에 개별 리포지토리 또는 기존 `.repo` 파일을 정의하는 것이 좋습니다.

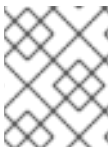
`/etc/dnf/dnf.conf` 파일의 개별 `[repository]` 섹션에 정의된 값은 `[main]` 섹션에 설정된 값을 재정의합니다.

다음 섹션에서는 Red Hat Enterprise Linux 9에서 사용자 지정 소프트웨어 리포지토리를 관리하는 방법을 설명합니다.

- [섹션 10.1. "setting DNF repository options"](#)는 `[repository]` 옵션을 설정하는 방법을 설명합니다.
- [섹션 10.2. "DNF 리포지토리 추가"](#)는 새 DNF 리포지토리를 정의하는 방법을 설명합니다.
- [섹션 10.3. "DNF 리포지토리 사용"](#) 시스템에 추가된 DNF 리포지토리를 활성화하는 방법을 설명합니다.
- [섹션 10.4. "DNF 리포지토리 사용"](#) 시스템에 추가된 DNF 리포지토리를 비활성화하는 방법을 설명합니다.

### 10.1. DNF 리포지토리 옵션 설정

`/etc/dnf/dnf.conf` 구성 파일에는 `[repository]` 섹션이 포함되어 있습니다. 여기서 `repository` 는 고유한 리포지토리 ID입니다. `[repository]` 섹션을 사용하면 개별 DNF 리포지토리를 정의할 수 있습니다.



#### 참고

충돌을 피하기 위해 Red Hat 리포지토리에서 사용하는 사용자 정의 리포지토리 이름을 지정하지 마십시오.

사용 가능한 `[리포지토리]` 옵션의 전체 목록은 `dnf.conf(5)` 매뉴얼 페이지의 `[repository] OPTIONS` 섹션을 참조하십시오.

### 10.2. DNF 리포지토리 추가

새 리포지토리를 정의하려면 다음을 수행할 수 있습니다.

- `[repository]` 섹션을 `/etc/dnf/dnf.conf` 파일에 추가합니다.
- `/etc/yum .repo s.d/` 디렉토리의 `.repo` 파일에 `[repository]` 섹션을 추가합니다. 설치된 RPM 또는 소프트웨어 관리 도구(예: Subscription Manager)는 자체 `.repo` 파일을 제공할 수 있습니다.



#### 참고

이 디렉토리의 `.repo` 파일 확장자가 `.nf`인 모든 파일로 `/etc/dnf/dnf.conf` 대신 `.repo` 파일에 리포지토리를 정의합니다.

다음 절차에서는 시스템에 DNF 리포지토리를 추가하는 방법을 설명합니다.



## 절차

- 시스템에 리포지토리를 추가합니다.

```
# dnf config-manager --add-repo repository_URL
```

*repository\_url* 을 리포지토리를 가리키는 URL로 바꿉니다.



### 주의

Red Hat 인증서 기반CDN( Content Delivery Network ) 이외의 확인되지 않았거나 신뢰할 수 없는 소스에서 소프트웨어 패키지를 취득 및 설치하면 잠재적인 보안 위험이 있으며 보안, 안정성, 호환성 및 유지 관리 문제가 발생할 수 있습니다.

## 10.3. DNF 리포지토리 활성화

다음 절차에서는 시스템에 DNF 리포지토리를 추가하는 방법을 설명합니다.

## 절차

- 리포지토리를 활성화합니다.

```
# dnf config-manager --enable repositoryID
```

*repositoryID* 를 고유한 리포지토리 ID로 바꿉니다.

## 추가 리소스

[리포지토리 나열](#)

## 10.4. DNF 리포지토리 비활성화

다음 절차에서는 시스템에 추가된 DNF 리포지토리를 비활성화하는 방법을 설명합니다.

## 절차

- 리포지토리를 비활성화합니다.

```
# dnf config-manager --disable repositoryID
```

*repositoryID* 를 고유한 리포지토리 ID로 바꿉니다.

## 추가 리소스

[리포지토리 나열](#)

## 부록 A. DNF 명령 목록

이 장에는 Red Hat Enterprise Linux 9의 콘텐츠 나열, 설치 및 제거를 위한 DNF 명령이 나열되어 있습니다.

## A.1. RHEL 9의 콘텐츠를 나열하는 명령

다음 표에는 RHEL 9에서 콘텐츠 및 세부 정보를 찾는 데 일반적으로 사용되는 DNF 명령이 나열되어 있습니다.

명령	설명
<b>DNF 검색어</b>	패키지와 관련된 용어를 사용하여 패키지 검색
<b>DNF repoquery 패키지</b>	선택한 패키지에서 사용 가능한 DNF 리포지토리 검색
<b>dnf 목록</b>	설치 및 사용 가능한 모든 패키지에 대한 정보 나열
<b>DNF 목록 -- installed</b> <b>DNF repoquery -- installed</b>	시스템에 설치된 모든 패키지 나열
<b>dnf list --available</b> <b>DNF repoquery</b>	활성화된 모든 리포지토리의 패키지를 설치할 수 있는 모든 패키지 나열
<b>DNF 리포지터리</b>	시스템에서 활성화된 모든 리포지토리 나열
<b>DNF repolist --disabled</b>	시스템의 비활성화된 모든 리포지토리 나열
<b>DNF repolist --all</b>	활성화된 리포지토리와 비활성화된 리포지토리 모두 나열
<b>DNF repoinfo</b>	리포지토리에 대한 추가 정보 나열
<b>dnf info package-name</b> <b>dnf repoquery --info package_name</b>	사용 가능한 패키지의 세부 정보 표시
<b>dnf repoquery --info --installed package_name</b>	시스템에 설치된 패키지의 세부 정보 표시
<b>DNF 모듈 목록</b>	모듈 및 현재 상태 나열  모듈 외부에서 패키지를 사용할 수 있는 경우 이 명령의 출력이 비어 있습니다.

명령	설명
<b>DNF 그룹 요약</b>	설치 및 사용 가능한 그룹의 수 보기
<b>DNF 그룹 목록</b>	설치 및 사용 가능한 모든 그룹 나열
<b>DNF 그룹 정보 <i>그룹 이름</i></b>	특정 그룹에 포함된 필수 및 선택적 패키지 나열

### A.2. RHEL 9에서 콘텐츠를 설치하기 위한 명령

다음 표에는 RHEL 9에서 콘텐츠를 설치하는 데 일반적으로 사용되는 DNF 명령이 나열되어 있습니다.

명령	설명
<b>dnf install <i>package-name</i></b>	패키지를 설치합니다.  패키지가 모듈 스트림에서 제공하는 경우 <b>dnf</b> 는 필수 모듈 스트림을 확인하고 이 패키지를 설치하는 동안 자동으로 활성화합니다. 이는 모든 패키지 종속 항목에 대해 재귀적으로 발생합니다. 더 많은 모듈 스트림이 요구 사항을 충족하는 경우 기본 모듈이 사용됩니다.
<b>dnf install <i>package-name-1 package-name-2</i></b>	여러 패키지 및 종속 항목을 동시에 설치
<b>dnf install <i>package-name.arch</i></b>	<i>다중lib</i> 시스템(AMD64, Intel 64 머신)에 패키지를 설치할 때 패키지 이름에 추가하여 패키지의 아키텍처를 지정합니다.
<b>DNF에서 <i>/usr/sbin/binary-file</i> 설치</b>	바이너리에 대한 경로를 인수로 사용하여 바이너리 설치
<b>DNF 설치 <i>/path/</i></b>	로컬 디렉터리에서 이전에 다운로드한 패키지 설치

명령	설명
<b>dnf install <i>package-url</i></b>	<p>패키지 URL을 사용하여 원격 패키지 설치</p> <p>시스템에서 패키지를 사용할 수 있도록 하려면 모듈을 활성화하지만 현재 이러한 패키지를 설치하지 않으려는 경우 모듈을 활성화합니다.</p> <p>일부 모듈은 기본 스트림을 정의하지 않을 수 있습니다. 이러한 경우 스트림을 명시적으로 지정해야 합니다.</p> <p>모듈에서 기본 스트림을 정의하는 경우 스트림과 콜론을 생략할 수 있습니다.</p> <p>이 명령을 실행하면 RPM 패키지가 설치되지 않습니다.</p> <p>일부 모듈은 기본 스트림을 정의하지 않습니다.</p> <p>이 명령을 실행하면 지정된 스트림도 활성화됩니다.</p>
<b>DNF 그룹 설치 <i>그룹 이름</i></b>	그룹 이름으로 패키지 그룹 설치
<b>DNF 그룹 설치 <i>groupID</i></b>	groupID로 패키지 그룹 설치

### A.3. RHEL 9에서 콘텐츠를 제거하는 명령

다음 표에는 RHEL 9에서 콘텐츠를 제거하는 데 일반적으로 사용되는 DNF 명령이 나열되어 있습니다.

명령	설명
<b>dnf remove <i>package-name</i></b>	특정 패키지 및 모든 종속 패키지 제거
<b>dnf remove <i>package-name-1 package-name-2</i></b>	여러 패키지 및 사용되지 않는 종속 항목을 동시에 제거
<b>DNF group remove <i>group-name</i></b>	그룹 이름으로 패키지 그룹 제거
<b>DNF 그룹 삭제 <i>groupID</i></b>	<p>groupID로 패키지 그룹 제거</p> <p>이 명령을 실행하면 시스템에서 중요한 패키지를 제거할 수 있습니다.</p> <p>이 명령을 실행하면 지정된 모듈에서 패키지를 제거하지 않습니다.</p> <p>이 명령을 실행하면 지정된 모듈에서 패키지를 제거하지 않습니다.</p>

