



Red Hat Enterprise Linux 9

RHEL 설치 프로그램의 부팅 옵션

부팅 옵션을 지정하여 설치 프로그램의 동작 사용자 정의

Red Hat Enterprise Linux 9 RHEL 설치 프로그램의 부팅 옵션

부팅 옵션을 지정하여 설치 프로그램의 동작 사용자 정의

법적 공지

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

초록

RHEL 설치 시스템에는 관리자를 위한 다양한 부팅 옵션이 포함되어 있습니다. 이러한 옵션을 사용하면 설치 프로그램을 사용자 지정하도록 특정 기능을 활성화하거나 비활성화하여 설치 프로그램의 기본 동작을 수정할 수 있습니다.

차례

| | |
|-----------------------------|----|
| RED HAT 문서에 관한 피드백 제공 | 3 |
| 1장. 부팅 옵션 | 4 |
| 1.1. 부팅 옵션 유형 | 4 |
| 1.2. 부팅 옵션 편집 | 4 |
| 1.3. 설치 소스 부팅 옵션 | 6 |
| 1.4. 네트워크 부팅 옵션 | 12 |
| 1.5. 콘솔 부팅 옵션 | 16 |
| 1.6. 디버그 부팅 옵션 | 19 |
| 1.7. 스토리지 부팅 옵션 | 21 |
| 1.8. KICKSTART 부팅 옵션 | 22 |
| 1.9. 고급 설치 부팅 옵션 | 23 |
| 1.10. 제거된 부팅 옵션 | 25 |

RED HAT 문서에 관한 피드백 제공

문서에 대한 피드백에 감사드립니다. 어떻게 개선할 수 있는지 알려주십시오.

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. 부팅 옵션

이 섹션에는 설치 프로그램의 기본 동작을 수정하는 데 사용할 수 있는 일부 부팅 옵션에 대한 정보가 포함되어 있습니다. 전체 부팅 옵션 목록은 [업스트림 부팅 옵션](#) 내용을 참조하십시오.

1.1. 부팅 옵션 유형

두 가지 유형의 부팅 옵션은 "=" 기호와 동일한 "sign"이 없는 부팅 옵션입니다. 부팅 옵션은 부팅 명령줄에 추가되고 공백으로 구분된 여러 옵션을 추가할 수 있습니다. 설치 프로그램과 관련된 부팅 옵션은 항상 **inst**로 시작합니다.

등호 "=" 기호가 있는 옵션

= 기호를 사용하는 부팅 옵션의 값을 지정해야 합니다. 예를 들어 **inst.vncpassword=** 옵션에는 값이 포함되어야 합니다(이 예에서는 암호). 이 예제의 올바른 구문은 **inst.vncpassword=password**입니다.

동등하지 않은 옵션 "=" sign

이 부팅 옵션은 값 또는 매개변수를 허용하지 않습니다. 예를 들어, **rd.live.check** 옵션은 설치를 시작하기 전에 설치 미디어를 강제로 설치 프로그램에 강제 적용합니다. 이 부팅 옵션이 있는 경우 설치 프로그램은 확인을 수행하고 부팅 옵션이 없으면 확인을 건너뛸 것입니다.

1.2. 부팅 옵션 편집

이 섹션에서는 부팅 메뉴에서 부팅 옵션을 편집하는 다양한 방법에 대해 설명합니다. 설치 미디어를 부팅하면 부팅 메뉴가 열립니다.

1.2.1. BIOS에서 boot: prompt 편집

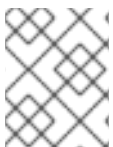
boot: 프롬프트를 사용하는 경우 첫 번째 옵션은 로드할 설치 프로그램 이미지 파일을 항상 지정해야 합니다. 대부분의 경우 키워드를 사용하여 이미지를 지정할 수 있습니다. 요구 사항에 따라 추가 옵션을 지정할 수 있습니다.

사전 요구 사항

- 부팅 가능한 설치 미디어(USB, CD 또는 DVD)를 생성했습니다.
- 미디어에서 설치를 부팅했으며 설치 부팅 메뉴가 열립니다.

절차

1. 부팅 메뉴를 열고 키보드의 **Esc** 키를 누릅니다.
2. 이제 **boot:** 프롬프트에 액세스할 수 있습니다.
3. 키보드에서 **Tab** 키를 눌러 도움말 명령을 표시합니다.
4. 키보드에서 **Enter** 키를 눌러 옵션으로 설치를 시작합니다. **boot:** 프롬프트에서 부팅 메뉴로 돌아가려면 시스템을 재시작하고 설치 미디어에서 다시 부팅합니다.



참고

boot: prompt는 **dracut** 커널 옵션도 허용합니다. 옵션 목록은 **dracut.cmdline(7)** 매뉴얼 페이지에서 사용할 수 있습니다.

1.2.2. > 프롬프트를 사용하여 사전 정의된 부팅 옵션 편집

BIOS 기반 AMD64 및 Intel 64 시스템에서는 > 프롬프트를 사용하여 사전 정의된 부팅 옵션을 편집할 수 있습니다. 전체 옵션 세트를 표시하려면 이 미디어 테스트를 선택하고 부팅 메뉴에서 **RHEL 9**를 설치합니다.

사전 요구 사항

- 부팅 가능한 설치 미디어(**USB, CD 또는 DVD**)를 생성했습니다.
- 미디어에서 설치를 부팅했으며 설치 부팅 메뉴가 열립니다.

절차

1. 부팅 메뉴에서 옵션을 선택하고 키보드의 **Tab** 키를 누릅니다. **>** 프롬프트에 액세스할 수 있으며 사용 가능한 옵션을 표시합니다.
2. 필요한 옵션을 **>** 프롬프트에 추가합니다.
3. **Enter**를 눌러 설치를 시작합니다.
4. **Esc** 를 눌러 편집을 취소하고 부팅 메뉴로 돌아갑니다.

1.2.3. UEFI 기반 시스템의 GRUB2 메뉴 편집

GRUB2 메뉴는 **UEFI** 기반 **AMD64, Intel 64** 및 **64비트 ARM** 시스템에서 사용할 수 있습니다.

사전 요구 사항

- 부팅 가능한 설치 미디어(**USB, CD 또는 DVD**)를 생성했습니다.
- 미디어에서 설치를 부팅했으며 설치 부팅 메뉴가 열립니다.

절차

1. 부팅 메뉴 창에서 필요한 옵션을 선택하고 **e** 를 누릅니다.
2. UEFI 시스템에서 커널 명령줄은 **linuxefi**로 시작합니다. 커서를 **linuxefi** 커널 명령줄의 끝으로 이동합니다.
3. 필요에 따라 매개 변수를 편집합니다. 예를 들어 하나 이상의 네트워크 인터페이스를 구성하려면 **linuxefi** 커널 명령줄 끝에 **ip=** 매개 변수를 추가한 다음 필수 값을 추가합니다.
4. 편집을 마치면 **Ctrl+X** 를 눌러 지정된 옵션을 사용하여 설치를 시작합니다.

1.3. 설치 소스 부팅 옵션

이 섹션에서는 다양한 설치 소스 부팅 옵션에 대해 설명합니다.

inst.repo=

inst.repo= boot 옵션은 설치 소스, 즉 패키지 리포지토리를 제공하는 위치와 이를 설명하는 유효한 **.treeinfo** 파일을 지정합니다. 예: **inst.repo=cdrom**. **inst.repo=** 옵션의 대상은 다음 설치 미디어 중 하나여야 합니다.

- 설치 프로그램 이미지, 패키지 및 리포지토리 데이터와 유효한 **.treeinfo** 파일을 포함하는 디렉터리 구조인 설치 가능 트리
- DVD (시스템 DVD 드라이브에 물리적 디스크가 있음)
- 전체 Red Hat Enterprise Linux 설치 DVD의 ISO 이미지는 디스크 또는 시스템에서 액세스할 수 있는 네트워크 위치에 배치됩니다.

inst.repo= 부트 옵션을 사용하여 다른 형식을 사용하여 다양한 설치 방법을 구성합니다. 다음 표에는 **inst.repo=** 부트 옵션 구문에 대한 세부 정보가 포함되어 있습니다.

표 1.1. inst.repo= 부팅 옵션 및 설치 소스의 유형 및 형식

| 소스 유형 | 부팅 옵션 형식 | 소스 형식 |
|-------------|---------------------------------------|----------------------|
| CD/DVD 드라이브 | inst.repo=cdrom:<device> | DVD를 물리적 디스크로 설치 [a] |

| 소스 유형 | 부팅 옵션 형식 | 소스 형식 |
|---|--|--|
| 마운트 가능한 장치 (HDD 및 USB 고착) | inst.repo=hd:<device>:/<path> | 설치 DVD의 이미지 파일입니다. |
| NFS 서버 | inst.repo=nfs: [options:]<server>:/<path> | 설치 DVD의 이미지 파일 또는 설치 DVD에 있는 디렉터리 및 파일의 전체 사본인 설치 트리입니다. [b] |
| HTTP 서버 | inst.repo=http://<host>/<path> | 설치 DVD에서 디렉터리 및 파일의 전체 사본인 설치 트리. |
| HTTPS 서버 | inst.repo=https://<host>/<path> | |
| FTP 서버 | inst.repo=ftp://<username>:<password>@<host>/<path> | |
| HMC | inst.repo=hmc | |
| <p>[a] 장치가 꺼져 있으면 설치 프로그램은 설치 DVD가 포함된 드라이브를 자동으로 검색합니다.</p> <p>[b] NFS Server 옵션은 기본적으로 NFS 프로토콜 버전 3을 사용합니다. 다른 버전을 사용하려면 nfsvers=X를 <i>options</i>에 추가하고 X를 사용하려는 버전 번호로 대체합니다.</p> | | |

다음과 같은 형식으로 디스크 장치 이름을 설정합니다.

- 커널 장치 이름 (예: /dev/sda1 또는 sdb2)
- 파일 시스템 레이블 (예: LABEL= skopeo 또는 LABEL=RHEL8)
- 파일 시스템 UUID (예: UUID=8176c7bf-04ff-403a-a832-9557f94e61db)

영숫자가 아닌 문자는 \xNN으로 표시되어야 합니다. 여기서 NN은 문자의 16진수 표현입니다. 예를 들어 \x20은 공백 (" ")입니다.

inst.addrepo=

inst.addrepo= boot 옵션을 사용하여 기본 리포지토리(**inst.repo=**)와 함께 다른 설치 소스로 사용할 수 있는 추가 리포지토리를 추가합니다. 부팅 중에 **inst.addrepo=** 부트 옵션을 여러 번 사용할 수

있습니다. 다음 표에는 `inst.addrepo=` 부팅 옵션 구문에 대한 세부 정보가 포함되어 있습니다.



참고

REPO_NAME은 리포지토리의 이름이며 설치 프로세스에 필요합니다. 이러한 리포지토리는 설치 프로세스 중에만 사용되며 설치된 시스템에 설치되지 않습니다.

통합 ISO에 대한 자세한 내용은 [통합 ISO를 참조하십시오.](#)

표 1.2. 설치 소스 및 부팅 옵션 형식

| 설치 소스 | 부팅 옵션 형식 | 추가 정보 |
|-------------------|---|---|
| URL에 설치 가능한 트리 | <code>inst.addrepo=REPO_NAME, [http,https,ftp]://<host>/<path></code> | 지정된 URL에서 설치 가능한 트리를 찾습니다. |
| NFS 경로에 설치 가능한 트리 | <code>inst.addrepo=REPO_NAME,nfs://<server>:/<path></code> | 지정된 NFS 경로에서 설치 가능한 트리를 찾습니다. 호스트 다음에 콜론이 필요합니다. 설치 프로그램은 RFC 2224에 따른 URL 구문 분석 대신 <code>nfs://</code> 이후의 모든 항목을 마운트 명령에 직접 전달합니다. |
| 설치 환경에 설치 가능한 트리 | <code>inst.addrepo=REPO_NAME,file://<path></code> | 설치 환경의 지정된 위치에서 설치할 수 있는 트리를 찾습니다. 이 옵션을 사용하려면 설치 프로그램에서 사용 가능한 소프트웨어 그룹을 로드하기 전에 리포지토리를 마운트해야 합니다. 이 옵션을 사용하면 하나의 부팅 가능 ISO에 여러 리포지토리를 사용할 수 있으며, ISO에서 기본 리포지토리와 추가 리포지토리를 모두 설치할 수 있습니다. 추가 리포지토리의 경로는 <code>/run/install/source/REPO_ISO_PATH</code> 입니다. 또한 Kickstart 파일의 <code>%pre</code> 섹션에 리포지토리 디렉터리를 마운트할 수 있습니다. 경로는 절대적이어야 하며 /로 시작해야 합니다. (예: <code>inst.addrepo=REPO_NAME,file:///<path></code>) |

| 설치 소스 | 부팅 옵션 형식 | 추가 정보 |
|-------|--|--|
| 디스크 | inst.addrepo=REPO_NAME,hd:<device>:<path> | 지정된 <device> 파티션을 마운트하고 <path>에 지정된 ISO에서 설치합니다. <path>를 지정하지 않으면 설치 프로그램은 <device>에서 유효한 설치 ISO를 찾습니다. 이 설치 방법을 사용하려면 설치 가능한 유효한 트리가 있는 ISO가 필요합니다. |

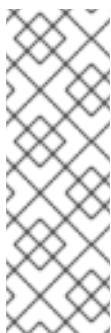
inst.stage2=

inst.stage2= 부트 옵션은 설치 프로그램의 런타임 이미지의 위치를 지정합니다. 이 옵션은 유효한 **.treeinfo** 파일이 포함된 디렉터리의 경로를 예상하고 **.treeinfo** 파일에서 런타임 이미지 위치를 읽습니다. **.treeinfo** 파일을 사용할 수 없는 경우 설치 프로그램은 **images/install.img** 에서 이미지를 로드하려고 합니다.

inst.stage2 옵션을 지정하지 않으면 설치 프로그램에서 **inst.repo** 옵션으로 지정된 위치를 사용하려고 합니다.

나중에 설치 프로그램에서 설치 소스를 수동으로 지정하려면 이 옵션을 사용합니다. 예를 들어, **CDN(Content Delivery Network)**을 설치 소스로 선택하려는 경우입니다. 설치 DVD 및 부팅 ISO에는 각 ISO에서 설치 프로그램을 부팅하는 데 적합한 **inst.stage2** 옵션이 이미 포함되어 있습니다.

설치 소스를 지정하려면 대신 **inst.repo=** 옵션을 사용합니다.



참고

기본적으로 **inst.stage2=** 부트 옵션은 설치 미디어에서 사용되며 특정 레이블로 설정됩니다(예: **inst.stage2=hd:LABEL=RHEL-x-0-BaseOS-x86_64**). 런타임 이미지가 포함된 파일 시스템의 기본 레이블을 수정하거나 사용자 지정 프로시저를 사용하여 설치 시스템을 부팅하는 경우 **inst.stage2=** 부팅 옵션이 올바른 값으로 설정되어 있는지 확인합니다.

inst.noverifyssl

inst.noverifyssl 부팅 옵션을 사용하여 설치 프로그램이 추가 Kickstart 리포지토리를 제외하고 모든 HTTPS 연결에 대한 SSL 인증서를 확인하지 않도록 합니다. 여기서 **--noverifyssl**을 리포지토리별로 설정할 수 있습니다.

예를 들어 원격 설치 소스가 자체 서명된 SSL 인증서를 사용하는 경우 **inst.noverifyssl** 부팅 옵션을 사용하면 설치 프로그램이 SSL 인증서를 확인하지 않고 설치를 완료할 수 있습니다.

inst.stage2=를 사용하여 소스를 지정할 때의 예

```
inst.stage2=https://hostname/path_to_install_image/ inst.noverifyssl
```

inst.repo=를 사용하여 소스를 지정할 때의 예

```
inst.repo=https://hostname/path_to_install_repository/ inst.noverifyssl
```

inst.stage2.all

inst.stage2.all 부팅 옵션을 사용하여 여러 **HTTP**, **HTTPS** 또는 **FTP** 소스를 지정합니다. **inst.stage2=** 부팅 옵션을 **inst.stage2.all** 옵션과 함께 여러 번 사용하여 성공할 때까지 소스에서 이미지를 순차적으로 가져올 수 있습니다. 예를 들어 다음과 같습니다.

```
inst.stage2.all
inst.stage2=http://hostname1/path_to_install_tree/
inst.stage2=http://hostname2/path_to_install_tree/
inst.stage2=http://hostname3/path_to_install_tree/
```

inst.dd=

inst.dd= 부팅 옵션은 설치 중에 드라이버 업데이트를 수행하는 데 사용됩니다. 설치 중에 드라이버를 업데이트하는 방법에 대한 자세한 내용은 [RHEL 자동 설치를](#) 참조하십시오.

inst.repo=hmc

이 옵션은 외부 네트워크 설정 요구 사항을 제거하고 설치 옵션을 확장합니다. 바이너리 DVD에서 부팅할 때 설치 프로그램에서 추가 커널 매개 변수를 입력하라는 메시지를 표시합니다. DVD를 설치 소스로 설정하려면 **inst.repo=hmc** 옵션을 커널 매개 변수에 추가합니다. 그러면 설치 프로그램에서 지원 요소(**SE**) 및 하드웨어 관리 콘솔(**HMC**) 파일 액세스를 활성화하고, DVD에서 **stage2** 이미지를 가져온 다음 소프트웨어 선택을 위해 DVD의 패키지에 액세스할 수 있습니다.

inst.proxy=

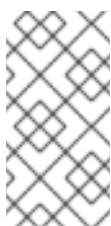
inst.proxy= 부팅 옵션은 **HTTP**, **HTTPS** 및 **FTP** 프로토콜에서 설치를 수행할 때 사용됩니다. 예를 들어 다음과 같습니다.

[PROTOCOL://][USERNAME[:PASSWORD]@]HOST[:PORT]

inst.nosave=

inst.nosave= boot 옵션을 사용하여 설치된 시스템에 저장되지 않은 설치 로그 및 관련 파일(예: **input_ks,output_ks,all_ks,logs** 및 **all**)을 제어합니다. 여러 값을 쉼표로 구분하여 결합할 수 있습니다. 예를 들면 다음과 같습니다.

inst.nosave=Input_ks,logs



참고

inst.nosave 부팅 옵션은 로그 및 입력/출력 **Kickstart** 결과와 같은 **Kickstart %post** 스크립트에서 제거할 수 없는 설치된 시스템의 파일을 제외하는 데 사용됩니다.

input_ks

입력 **Kickstart** 결과를 저장하는 기능을 비활성화합니다.

output_ks

설치 프로그램에서 생성된 출력 **Kickstart** 결과를 저장하는 기능을 비활성화합니다.

all_ks

입력 및 출력 **Kickstart** 결과를 저장하는 기능을 비활성화합니다.

logs

모든 설치 로그를 저장하는 기능을 비활성화합니다.

all

모든 **Kickstart** 결과 및 모든 로그를 저장하는 기능을 비활성화합니다.

inst.multilib

inst.multilib 부팅 옵션을 사용하여 **DNF**의 **multilib_policy**를 **best**가 아닌 **all**로 설정합니다.

inst.memcheck

inst.memcheck 부팅 옵션은 설치를 완료하는 데 충분한 **RAM**이 있는지 확인하는 검사를 수행합니다. **RAM**이 충분하지 않으면 설치 프로세스가 중지됩니다. 설치 중에 시스템 점검은 대략적이고 메

모리 사용은 패키지 선택, 사용자 인터페이스(예: 그래픽 또는 텍스트) 및 기타 매개 변수에 따라 달라집니다.

inst.nomemcheck

inst.nomemcheck 부팅 옵션은 설치를 완료하기에 충분한 **RAM**이 있는지 확인하기 위해 검사를 수행하지 않습니다. 최소 메모리 용량이 권장되지 않는 메모리보다 적은 설치를 시도하면 설치 프로세스가 실패할 수 있습니다.

1.4. 네트워크 부팅 옵션

로컬 이미지에서 부팅하지 않고 네트워크를 통해 이미지를 부팅해야 하는 경우 다음 옵션을 사용하여 네트워크 부팅을 사용자 지정할 수 있습니다.



참고

dracut 툴을 사용하여 네트워크를 초기화합니다. **dracut** 옵션 전체 목록은 **dracut.cmdline(7)** 매뉴얼 페이지를 참조하십시오.

ip=

ip= boot 옵션을 사용하여 하나 이상의 네트워크 인터페이스를 구성합니다. 여러 인터페이스를 구성하려면 다음 방법 중 하나를 사용합니다.

- 각 인터페이스에 한 번 **ip** 옵션을 여러 번 사용합니다. 이렇게 하려면 **rd.neednet=1** 옵션을 사용하고 **bootdev** 옵션을 사용하여 기본 부팅 인터페이스를 지정합니다.
- **ip** 옵션을 한 번 사용한 다음 **Kickstart**를 사용하여 추가 인터페이스를 설정합니다. 이 옵션에는 여러 다른 형식을 사용할 수 있습니다. 다음 테이블에는 가장 일반적인 옵션에 대한 정보가 포함되어 있습니다.

다음 표에서는 다음을 수행합니다.

- **ip** 매개 변수는 클라이언트 **IP** 주소를 지정하고 **IPv6** 에는 **192.0.2.1** 또는 **[2001:db8::99]**와 같이 대괄호가 필요합니다.
- **gateway** 매개 변수는 기본 게이트웨이입니다. **IPv6** 에는 대괄호가 필요합니다.

- **netmask** 매개 변수는 사용할 넷마스크입니다. 전체 넷마스크(예: 255.255.255.0) 또는 접두사(예: 64)일 수 있습니다.
- **hostname** 매개 변수는 클라이언트 시스템의 호스트 이름입니다. 이 매개 변수는 선택 사항입니다.

표 1.3. 네트워크 인터페이스를 구성하기 위한 부팅 옵션 형식

| 부팅 옵션 형식 | 구성 방법 |
|---|---|
| ip=method | 인터페이스 자동 설정 |
| ip=interface:method | 특정 인터페이스의 자동 설정 |
| ip=ip::gateway:netmask:hostname:interface:none | 정적 구성(예: IPv4: ip=192.0.2.1::192.0.2.254:255.255.0:server.example.com:enp1s0:none) IPv6: ip=[2001:db8::1]::[2001:db8::ffe]:64:server.example.com:enp1s0:none |
| ip=ip::gateway:netmask:hostname:interface:method:mtu | 덮어쓰기를 사용하는 특정 인터페이스의 자동 구성 |

자동 인터페이스에 대한 설정 방법

재정의를 사용하여 특정 인터페이스의 자동 구성은 **dhcp** 와 같은 지정된 자동 구성 방법을 사용하여 인터페이스를 엽니다. 그러나 자동으로 가져온 **IP** 주소, 게이트웨이, 넷마스크, 호스트 이름 또는 기타 지정된 매개 변수를 덮어씁니다. 모든 매개 변수는 선택 사항이므로 재정의할 매개 변수만 지정합니다.

method 매개 변수는 다음 중 하나일 수 있습니다.

DHCP

dhcp

IPv6 DHCP

dhcp6

IPv6 자동 구성

auto6

iSCSI 부팅 펌웨어 테이블(ibft)

ibft

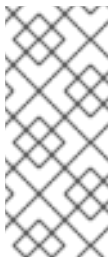


참고

- **ip** 옵션을 지정하지 않고 **inst.ks=http://host/path** 와 같은 네트워크 액세스 권한이 필요한 부팅 옵션을 사용하는 경우 **ip =dhcp..**
- **iSCSI** 대상에 자동으로 연결하려면 **ip=ibft** 부팅 옵션을 사용하여 대상에 액세스하기 위해 네트워크 장치를 활성화합니다.

nameserver=

nameserver= 옵션은 이름 서버의 주소를 지정합니다. 이 옵션을 여러 번 사용할 수 있습니다.



참고

ip= 매개 변수에는 대괄호가 필요합니다. 그러나 **IPv6** 주소는 대괄호로 묶지 않습니다. **IPv6** 주소에 사용할 올바른 구문의 예는 **nameserver=2001:db8::1**입니다.

bootdev=

bootdev= 옵션은 부팅 인터페이스를 지정합니다. **ip** 옵션을 두 개 이상 사용하는 경우 이 옵션이 필요합니다.

ifname=

ifname= 옵션은 지정된 **MAC** 주소가 있는 네트워크 장치에 인터페이스 이름을 할당합니다. 이 옵션을 여러 번 사용할 수 있습니다. 구문은 **ifname=interface:MAC**입니다. 예를 들어 다음과 같습니다.

```
ifname=eth0:01:23:45:67:89:ab
```



참고

ifname= 옵션은 설치 중에 사용자 지정 네트워크 인터페이스 이름을 설정하는 유일한 방법입니다.

inst.dhcpclass=

inst.dhcpclass= 옵션은 DHCP 공급업체 클래스 식별자를 지정합니다. **dhcpd** 서비스는 이 값을 **vendor-class-identifier** 로 인식합니다. 기본값은 **anaconda-\$(uname -srm)**입니다. **inst.dhcpclass=** 옵션이 올바르게 적용되도록 하려면 **ip** 옵션도 추가하여 설치 초기 단계에서 네트워크 활성화를 요청합니다.

inst.waitfornet=

inst.waitfornet=SECONDS 부팅 옵션을 사용하면 설치 시스템이 설치 전에 네트워크 연결을 기다릴 수 있습니다. **SECONDS** 인수에서 제공되는 값은 시간 초과 전에 네트워크 연결을 대기하고 네트워크 연결이 존재하지 않는 경우에도 설치 프로세스를 계속할 때까지 대기하는 최대 시간을 지정합니다.

vlan=

vlan= 옵션을 사용하여 지정된 이름의 지정된 인터페이스에서 **VLAN(Virtual LAN)** 장치를 구성합니다. 구문은 **vlan=name:interface**입니다. 예를 들어 다음과 같습니다.

```
vlan=vlan5:enp0s1
```

이렇게 하면 **enp0s1** 인터페이스에서 **vlan5** 라는 **VLAN** 장치가 구성됩니다. 이름은 다음 형식을 사용할 수 있습니다.

- **VLAN_PLUS_VID: vlan0005**
- **VLAN_PLUS_VID_NO_PAD: vlan5**
- **DEV_PLUS_VID: enp0s1.0005**
- **DEV_PLUS_VID_NO_PAD: enp0s1.5**

bond=

bond= 옵션을 사용하여 **bond=name[:interfaces][:options]** 구문을 사용하여 본딩 장

치를 구성합니다. **name**을 본딩 장치 이름으로, 인터페이스를 쉼표로 구분된 물리(Ethernet) 인터페이스로, 옵션을 쉼표로 구분된 본딩 옵션 목록으로 교체합니다. 예를 들어 다음과 같습니다.

```
bond=bond0:enp0s1,enp0s2:mode=active-backup,tx_queues=32,downdelay=5000
```

사용 가능한 옵션 목록을 보려면 **modinfo** 본딩 명령을 실행합니다.

team=

team= 옵션을 사용하여 **team=name:interfaces** 구문으로 팀 장치를 구성합니다. **name**을 팀 장치의 원하는 이름으로 바꾸고, 팀 장치에서 기본 인터페이스로 사용할 이름을 쉼표로 구분된 물리적(Ethernet) 장치로 바꿉니다. 예를 들어 다음과 같습니다.

```
team=team0:enp0s1,enp0s2
```



중요

Red Hat Enterprise Linux 9에서는 네트워크 타이밍이 더 이상 사용되지 않습니다. 대안으로 네트워크 본딩 드라이버를 사용하는 것이 좋습니다. 자세한 내용은 [네트워크 본딩 구성](#)을 참조하십시오.

bridge=

bridge= 옵션을 사용하여 **bridge=name:interfaces** 구문을 사용하여 브리지 장치를 구성합니다. **name**을 브리지 장치 및 **interfaces**의 원하는 이름으로 교체하고, 브릿지 장치에서 기본 인터페이스로 사용할 물리적(Ethernet) 장치 목록을 쉼표로 구분한 목록으로 바꿉니다. 예를 들어 다음과 같습니다.

```
bridge=bridge0:enp0s1,enp0s2
```

추가 리소스

- [네트워킹 구성 및 관리](#)

1.5. 콘솔 부팅 옵션

이 섹션에서는 콘솔의 부팅 옵션, 모니터 디스플레이 및 키보드를 구성하는 방법을 설명합니다.

console=

console= 옵션을 사용하여 기본 콘솔로 사용할 장치를 지정합니다. 예를 들어 첫 번째 직렬 포트에서 콘솔을 사용하려면 **console=ttyS0**을 사용합니다. **console=** 인수를 사용하면 텍스트 UI로 설치가 시작됩니다. **console=** 옵션을 여러 번 사용해야 하는 경우 지정된 모든 콘솔에 부팅 메시지가 표시됩니다. 그러나 설치 프로그램은 마지막 지정된 콘솔만 사용합니다. 예를 들어 **console=ttyS0 console=ttyS1**을 지정하면 설치 프로그램에서 **ttyS1**을 사용합니다.

inst.lang=

설치 중에 사용할 언어를 설정하려면 **inst.lang=** 옵션을 사용합니다. 로케일 목록을 보려면 **locale -a | grep _** 또는 **localectl list-locales | grep _** 명령을 입력합니다.

inst.geoloc=

설치 프로그램에서 **inst.geoloc=** 옵션을 사용하여 위치 사용을 구성합니다. **Geolocation**은 언어 및 시간대를 사전 설정하는 데 사용되며 **inst.geoloc=value** 구문을 사용합니다. **value**는 다음 매개변수 중 하나일 수 있습니다.

- **geolocation: inst.geoloc=0**을 비활성화합니다.
- **Fedora GeolP API 사용: inst.geoloc=provider_fedora_geoip.** 이 옵션은 더 이상 사용되지 않습니다.
- **Hostip.info GeolP API: inst.geoloc=provider_hostip** 를 사용합니다. 이 옵션은 더 이상 사용되지 않습니다.

inst.keymap=

inst.keymap= 옵션을 사용하여 설치에 사용할 키보드 레이아웃을 지정합니다.

inst.cmdline

inst.cmdline 옵션을 사용하여 설치 프로그램이 명령줄 모드에서 실행되도록 합니다. 이 모드는 상호 작용을 허용하지 않으며 **Kickstart** 파일 또는 명령줄의 모든 옵션을 지정해야 합니다.

inst.graphical

inst.graphical 옵션을 사용하여 설치 프로그램이 그래픽 모드에서 실행되도록 합니다. 그래픽 모드가 기본값입니다.

inst.text

inst.text 옵션을 사용하여 설치 프로그램이 그래픽 모드 대신 텍스트 모드에서 실행되도록 합니다.

inst.noninteractive

비대화형 모드에서 설치 프로그램을 실행하려면 **inst.noninteractive** 부팅 옵션을 사용합니다. 비대화형 모드에서는 사용자 상호 작용이 허용되지 않으며, **inst.noninteractive** 옵션을 그래픽 또는 텍스트 설치와 함께 사용할 수 있습니다. 텍스트 모드에서 **inst.noninteractive** 옵션을 사용하면 **inst.cmdline** 옵션과 동일하게 작동합니다.

inst.resolution=

inst.resolution= 옵션을 사용하여 그래픽 모드에서 화면 해상도를 지정합니다. 형식은 **NxM**입니다. 여기서 **N**은 화면 너비이고 **M**은 화면 높이(px)입니다. 권장 해상도는 **1024x768**입니다.

inst.vnc

inst.vnc 옵션을 사용하여 **VNC(Virtual Network Computing)**를 사용하여 그래픽 설치를 실행합니다. 설치 프로그램과 상호 작용하려면 **VNC 클라이언트 애플리케이션**을 사용해야 합니다. **VNC** 공유가 활성화되면 여러 클라이언트가 연결할 수 있습니다. **VNC**를 사용하여 설치된 시스템은 텍스트 모드에서 시작됩니다.

inst.vncpassword=

설치 프로그램에서 사용하는 **VNC 서버**에서 암호를 설정하려면 **inst.vncpassword=** 옵션을 사용합니다.

inst.vncconnect=

inst.vncconnect= 옵션을 사용하여 지정된 호스트 위치에서 수신 대기 중인 **VNC 클라이언트**에 연결합니다(예: **inst.vncconnect=<host>[:<port>]** 기본 포트는 **5900**입니다. **vncviewer -listen** 명령을 입력하여 이 옵션을 사용할 수 있습니다.

inst.xdriver=

inst.xdriver= 옵션을 사용하여 설치 중 및 설치된 시스템에서 모두 사용할 **X 드라이버 이름**을 지정합니다.

inst.usefbx

inst.usefbx 옵션을 사용하여 하드웨어별 드라이버 대신 프레임 버퍼 **X 드라이버**를 사용하도록 설치 프로그램에 메시지를 표시합니다. 이 옵션은 **inst.xdriver=fbdev** 옵션과 동일합니다.

modprobe.blacklist=

modprobe.blacklist= 옵션을 사용하여 하나 이상의 드라이버를 차단 목록에 표시하거나 완전히 비활성화합니다. 이 옵션을 사용하지 않도록 설정하는 드라이버(mods)는 설치가 시작될 때 로드할 수 없습니다. 설치가 완료되면 설치된 시스템에서 이러한 설정을 유지합니다. **blacklisted** 드라이버 목록은 **/etc/modprobe.d/** 디렉토리에서 찾을 수 있습니다. 쉘표로 구분된 목록을 사용하여 여러 드라이버를 비활성화합니다. 예를 들어 다음과 같습니다.

```
modprobe.blacklist=ahci,firewire_ohci
```



참고

modprobe.blacklist 를 다른 명령행 옵션과 함께 사용할 수 있습니다. 예를 들어 **inst.dd** 옵션과 함께 사용하여 기존 드라이버의 업데이트된 버전이 드라이버 업데이트 디스크에서 로드되었는지 확인합니다.

```
modprobe.blacklist=virtio_blk
```

inst.xtimeout=

inst.xtimeout= 옵션을 사용하여 X 서버를 시작하는 데 시간 초과를 초 단위로 지정합니다.

inst.sshd

SSH를 사용하여 시스템에 연결하고 설치 진행 상황을 모니터링할 수 있도록 **inst.sshd** 옵션을 사용하여 설치 중에 **sshd** 서비스를 시작합니다. SSH에 대한 자세한 내용은 **ssh(1)** 도움말 페이지를 참조하십시오. 기본적으로 **sshd** 옵션은 64비트 IBM Z 아키텍처에서만 자동으로 시작됩니다. 다른 아키텍처에서 **inst.sshd** 옵션을 사용하지 않는 경우 **sshd**가 시작되지 않습니다.



참고

설치하는 동안 **root** 계정에는 기본적으로 암호가 없습니다. **sshpw Kickstart** 명령을 사용하여 설치 중에 **root** 암호를 설정할 수 있습니다.

inst.kdump_addon=

inst.kdump_addon= 옵션을 사용하여 설치 프로그램에서 **Kdump** 구성 화면(add-on)을 활성화하거나 비활성화합니다. 이 화면은 기본적으로 활성화되어 있습니다. **inst.kdump_addon=off**를 사용하여 비활성화합니다. 애드온을 비활성화하면 그래픽 및 텍스트 기반 인터페이스의 **Kdump** 화면과 **%addon com_redhat_kdump Kickstart** 명령이 비활성화됩니다.

1.6. 디버그 부팅 옵션

이 섹션에서는 문제를 디버깅할 때 사용할 수 있는 옵션에 대해 설명합니다.

inst.rescue

inst.rescue 옵션을 사용하여 시스템 진단 및 수정에 필요한 복구 환경을 실행합니다. 예를 들어 복구 모드에서 파일 시스템을 복구할 수 있습니다.

inst.updates=

inst.updates= 옵션을 사용하여 설치 중에 적용할 **updates.img** 파일의 위치를 지정합니다. **updates.img** 파일은 여러 소스 중 하나에서 파생될 수 있습니다.

표 1.4. **updates.img** 파일 소스

| 소스 | 설명 | 예제 |
|----------------|--|--|
| 네트워크에서 업데이트 | updates.img 의 네트워크 위치를 지정합니다. 설치 트리를 수정할 필요가 없습니다. 이 방법을 사용하려면 inst.updates 를 포함하도록 커널 명령행을 편집합니다. | inst.updates=http://website.com/path/to/updates.img. |
| 디스크 이미지에서 업데이트 | 플로피 드라이브 또는 USB 키에 updates.img 를 저장합니다. 이는 ext2 파일 시스템 유형의 updates.img 에서만 수행할 수 있습니다. 이미지 내용을 플로피 드라이브에 저장하려면 플로피 디스크를 삽입하고 명령을 실행합니다. | DD if=updates.img of=/dev/fd0 bs=72k count=20. USB 키 또는 플래시 미디어를 사용하려면 /dev/fd0 을 USB 플래시 드라이브의 장치 이름으로 교체합니다. |
| 설치 트리에서 업데이트 | CD, 디스크, HTTP 또는 FTP 설치를 사용하는 경우 모든 설치에서 .img 파일을 감지할 수 있도록 설치 트리에 updates.img 를 저장합니다. 파일 이름은 updates.img 여야 합니다. | NFS 설치의 경우 파일을 images/ 디렉터리에 저장하거나 RHupdates/ 디렉터리에 저장합니다. |

inst.syslog=

설치가 시작될 때 지정된 호스트의 **syslog** 프로세스로 로그 메시지를 보냅니다. 들어오는 연결을 허용하도록 원격 **syslog** 프로세스가 구성된 경우에만 **inst.syslog=**를 사용할 수 있습니다.

inst.virtolog=

inst.virtolog= 옵션을 사용하여 로그 전달에 사용할 **virtio** 포트(**/dev/virtio-ports/name**에서 문자 장치)를 지정합니다. 기본값은 **org.fedoraproject.anaconda.log.0**입니다.

rd.live.ram

images/install.img의 2단계 이미지를 **RAM**에 복사합니다. 이렇게 하면 이미지 크기에 필요한 메모리가 증가하며 일반적으로 **400~800MB** 사이입니다.

inst.nokill

치명적인 오류가 발생하거나 설치 프로세스가 끝나면 설치 프로그램이 재부팅되지 않도록 합니다. 재부팅 시 손실되는 설치 로그를 캡처하여 사용합니다.

inst.noshell

설치하는 동안 터미널 세션 2(**tty2**)의 셸을 방지합니다.

inst.notmux

설치 중에 **tmux**를 사용하지 않도록 합니다. 출력은 터미널 제어 문자 없이 생성되며 대화형이 아닌 용도로 사용됩니다.

inst.remotelog=

TCP 연결을 사용하여 모든 로그를 원격 호스트:port 로 보냅니다. 리스너가 없고 설치가 정상적으로 진행된다면 연결이 중단됩니다.

1.7. 스토리지 부팅 옵션

이 섹션에서는 스토리지 장치에서 부팅을 사용자 지정하기 위해 지정할 수 있는 옵션에 대해 설명합니다.

inst.nodmraid

dmraid 지원을 비활성화합니다.

**주의**

이 옵션을 주의해서 사용하십시오. 펌웨어 **RAID** 배열의 일부로 잘못 식별되는 디스크가 있는 경우 **dmraid** 또는 **wipefs** 와 같은 적절한 도구를 사용하여 제거해야 하는 오래된 **RAID** 메타데이터가 있을 수 있습니다.

inst.nompath

다중 경로 장치 지원을 비활성화합니다. 일반 블록 장치를 다중 경로 장치로 잘못 식별하는 **false-positive**가 있는 경우에만 이 옵션을 사용합니다.



주의

이 옵션을 주의해서 사용하십시오. 다중 경로 하드웨어에 이 옵션을 사용하지 마십시오. 이 옵션을 사용하여 다중 경로 장치의 단일 경로에 설치하는 것은 지원되지 않습니다.

`inst.gpt`

설치 프로그램이 **MBR(Master Boot Record)**이 아닌 **GUID 파티션 테이블(GPT)**에 파티션 정보를 설치하도록 강제 적용합니다. 이 옵션은 **BIOS 호환성 모드**에 있지 않는 한 **UEFI 기반 시스템**에서 유효하지 않습니다. 일반적으로 **BIOS 기반 시스템** 및 **BIOS 호환성 모드**의 **UEFI 기반 시스템**은 디스크 크기가 **2³²** 섹터이거나 큰 경우 파티션 정보를 저장하기 위해 **DASD 스키마**를 사용하려고 합니다. 디스크 섹터는 일반적으로 크기가 **512바이트**이므로 일반적으로 **2TiB**와 동일합니다. `inst.gpt` 부팅 옵션을 사용하면 **GPT**를 더 작은 디스크에 쓸 수 있습니다.

`inst.wait_for_disks=`

`inst.wait_for_disks=` 옵션을 사용하여 설치 시작 시 디스크 장치가 표시될 때까지 대기할 설치 프로그램 초를 지정합니다. **OEMDRV** 레이블이 지정된 장치를 사용하여 **Kickstart** 파일 또는 커널 드라이버를 자동으로 로드하지만 부팅 프로세스 중에 장치가 표시되는 데 시간이 오래 걸리는 경우 이 옵션을 사용합니다. 기본적으로 설치 프로그램은 **5 초** 동안 기다립니다. 지연을 최소화하기 위해 **0 초**를 사용합니다.

1.8. KICKSTART 부팅 옵션

이 섹션에서는 **Kickstart** 파일에 추가하여 설치를 자동화할 수 있는 부팅 옵션에 대해 설명합니다.

`inst.ks=`

설치를 자동화하는 데 사용할 **Kickstart** 파일의 위치를 정의합니다. `inst.repo` 형식을 사용하여 위치를 지정할 수 있습니다. 장치가 아니라 경로를 지정하면 설치 프로그램은 지정된 장치에서 `/ks.cfg`에서 **Kickstart** 파일을 찾습니다.

장치를 지정하지 않고 이 옵션을 사용하는 경우 설치 프로그램은 옵션에 다음 값을 사용합니다.

```
inst.ks=nfs:next-server:/filename
```

이전 예에서 `next-server`는 **DHCP next-server** 옵션 또는 **DHCP** 서버 자체의 **IP 주소**이며 `filename`은 **DHCP** 파일 이름 옵션 또는 `/kickstart/`입니다. 지정된 파일 이름이 / 문자로 종료되면 `ip-kickstart`가 추가

됩니다. 다음 표에는 예제가 포함되어 있습니다.

표 1.5. 기본 Kickstart 파일 위치

| DHCP 서버 주소 | 클라이언트 주소 | Kickstart 파일 위치 |
|---------------|-----------------|--|
| 192.168.122.1 | 192.168.122.100 | 192.168.122.1:/kickstart/192.168.122.100-kickstart |

OEMDRV 레이블이 있는 볼륨이 있는 경우 설치 프로그램은 **ks.cfg** 라는 Kickstart 파일을 로드하려고 합니다. Kickstart 파일이 이 위치에 있는 경우 **inst.ks=** 부팅 옵션을 사용할 필요가 없습니다.

inst.ks.all

inst.ks.all 옵션을 지정하여 여러 **inst.ks** 옵션에서 제공하는 여러 Kickstart 파일 위치를 순차적으로 시도합니다. 첫 번째 위치가 사용됩니다. 이는 유형 **http,https** 또는 **ftp**의 위치에만 적용되며 다른 위치는 무시됩니다.

inst.ks.sendmac

inst.ks.sendmac 옵션을 사용하여 모든 네트워크 인터페이스의 **MAC** 주소가 포함된 발신 **HTTP** 요청에 헤더를 추가합니다. 예를 들어 다음과 같습니다.

```
X-RHN-Provisioning-MAC-0: eth0 01:23:45:67:89:ab
```

이 기능은 **inst.ks=http**를 사용하여 시스템을 프로비저닝할 때 유용할 수 있습니다.

inst.ks.sendsn

inst.ks.sendsn 옵션을 사용하여 발신 **HTTP** 요청에 헤더를 추가합니다. 이 헤더에는 **/sys/class/dmi/id/product_serial**에서 읽은 시스템 일련 번호가 포함되어 있습니다. 헤더에는 다음 구문이 있습니다.

```
X-System-Serial-Number: R8VA23D
```

추가 리소스

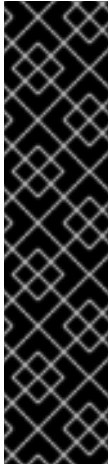
- [전체 부팅 옵션 목록](#)

1.9. 고급 설치 부팅 옵션

이 섹션에는 고급 설치 부팅 옵션에 대한 정보가 포함되어 있습니다.

inst.kexec

재부팅을 수행하는 대신 설치 종료 시 **kexec** 시스템 호출을 실행합니다. **inst.kexec** 옵션은 새 시스템을 즉시 로드하고 BIOS 또는 펌웨어에서 일반적으로 수행하는 하드웨어 초기화를 바이패스합니다.



중요

이 옵션은 더 이상 사용되지 않으며 기술 프리뷰로만 제공됩니다. 기술 프리뷰 기능에 대한 Red Hat 지원 범위 정보는 기술 [프리뷰 기능 지원 범위](#) 문서를 참조하십시오.

kexec 를 사용하면 전체 시스템 재부팅 중에 일반적으로 지워지는 장치 레지스터가 데이터로 채워질 수 있습니다. 이로 인해 특정 장치 드라이버에 문제가 발생할 수 있습니다.

inst.multilib

64비트 AMD64 또는 Intel 64 시스템에 32비트 패키지를 설치할 수 있도록 다중lib 패키지에 시스템을 구성합니다. 일반적으로 AMD64 또는 Intel 64 시스템에서는 이 아키텍처의 패키지만, x86_64로 표시되고 모든 아키텍처의 패키지는 noarch로 표시됩니다. **inst.multilib** 부팅 옵션을 사용하면 i686으로 표시된 32비트 AMD 또는 Intel 시스템용 패키지가 자동으로 설치됩니다.

이는 %packages 섹션에 직접 지정된 패키지에만 적용됩니다. 패키지가 종속성으로 설치된 경우 정확히 지정된 종속성만 설치됩니다. 예를 들어 glibc 패키지에 따라 bash 패키지를 설치하는 경우 bash 패키지는 여러 변형으로 설치되지만 glibc 패키지는 bash 패키지에 필요한 변형으로만 설치됩니다.

selinux=0

설치 프로그램 및 설치된 시스템에서 SELinux 사용을 비활성화합니다. 기본적으로 SELinux는 설치 프로그램에서 허용 모드로 작동하며 설치된 시스템의 강제 모드에서 작동합니다.



참고

inst.selinux=0 및 **selinux=0** 옵션은 동일하지 않습니다. * **inst.selinux=0**: 설치 프로그램에서 SELinux만 비활성화합니다. * **selinux=0**: 설치 프로그램과 설치된 시스템에서 SELinux 사용을 비활성화합니다. SELinux를 비활성화하면 이벤트가 기록되지 않습니다.

inst.nonibftiscsiboot

iSCSI 부팅 펌웨어 테이블(iBFT)에 구성되지 않은 iSCSI 장치에 부트 로더를 배치합니다.

1.10. 제거된 부팅 옵션

다음 부팅 옵션이 Red Hat Enterprise Linux에서 제거되었습니다.

inst.zram

zram.service는 더 이상 실행할 수 없습니다. 자세한 내용은 **zram-generator**를 참조하십시오.

inst.singlelang

단일 언어 모드는 더 이상 지원되지 않습니다.

inst.loglevel

로그 수준은 항상 **debug**로 설정됩니다.