



# Red Hat Enterprise Linux 9

## 장치 매퍼 다중 경로 구성

장치 매퍼 다중 경로 기능 구성 및 관리



# Red Hat Enterprise Linux 9 장치 맵퍼 다중 경로 구성

---

장치 맵퍼 다중 경로 기능 구성 및 관리

## 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<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

장치 매퍼 다중 경로(DM Multipath)를 사용하면 서버 노드와 스토리지 어레이 간의 여러 I/O 경로를 단일 장치로 구성할 수 있습니다. 이러한 I/O 경로는 별도의 케이블, 스위치 및 컨트롤러를 포함할 수 있는 물리적 SAN(Storage Area Network) 연결입니다. 다중 경로에서는 I/O 경로를 집계하고 집계된 경로로 구성된 새 장치를 생성합니다.

## 차례

RED HAT 문서에 관한 피드백 제공 .....	4
<b>1장. 장치 매핑 다중 경로 개요 .....</b>	<b>5</b>
1.1. 하나의 RAID 장치가 있는 ACTIVE/PASSIVE 다중 경로 구성	5
1.2. 두 개의 RAID 장치가 있는 ACTIVE/PASSIVE 다중 경로 구성	6
1.3. 하나의 RAID 장치가 있는 ACTIVE/ACTIVE 다중 경로 구성	7
1.4. DM MULTIPATH 구성 요소	8
1.5. 다중 경로 명령	9
1.6. 다중 경로 토폴로지 표시	10
1.7. 경로 상태	11
1.8. 추가 리소스	12
<b>2장. 다중 경로 장치 .....</b>	<b>13</b>
2.1. 다중 경로 장치 식별자	13
2.2. 논리 볼륨의 다중 경로 장치	14
<b>3장. DM MULTIPATH 구성 .....</b>	<b>16</b>
3.1. DEVICE-MAPPER-MULTIPATH 패키지 확인	16
3.2. DM MULTIPATH를 사용하여 기본 장애 조치 구성 설정	16
3.3. 다중 경로 장치를 생성할 때 로컬 디스크 무시	18
3.4. DM MULTIPATH를 사용하여 추가 스토리지 구성	20
3.5. INITRAMFS에서 다중 경로 구성	21
<b>4장. NVME 장치에서 멀티패스 활성화 .....</b>	<b>24</b>
4.1. 네이티브 NVME 다중 경로 및 DM MULTIPATH	24
4.2. NVME 장치에서 DM MULTIPATH 활성화	24
4.3. 기본 NVME 다중 경로 활성화	27
<b>5장. DM MULTIPATH 구성 파일 수정 .....</b>	<b>31</b>
5.1. 구성 파일 개요	31
5.2. 구성 파일 기본값	32
5.3.	40
5.4.	43
5.5.	46
5.6. 장치 시간 초과 DM MULTIPATH 덮어쓰기	50
5.7. 다중 경로 구성 파일 기본값 수정	50
5.8. 특정 장치의 다중 경로 설정 수정	52
5.9. 프로토콜을 사용하여 특정 장치의 다중 경로 구성 수정	53
5.10. 스토리지 컨트롤러의 다중 경로 설정 수정	55
5.11. 모든 장치에 대한 다중 경로 값 설정	57
<b>6장. 장치가 다중 경로 방지 .....</b>	<b>59</b>
6.1. DM MULTIPATH가 경로에 대한 다중 경로 장치를 생성하는 경우의 조건	59
6.2. 특정 장치에서 멀티패스를 비활성화하는 기준	61
6.3. WWID를 통한 다중 경로 비활성화	62
6.4. 장치 이름으로 다중 경로 비활성화	63
6.5. 장치 유형별 다중 경로 비활성화	64
6.6. UDEV 속성의 다중 경로 비활성화	65
6.7. 장치 프로토콜별 다중 경로 비활성화	66
6.8. 다중 경로가 비활성화된 장치에 대한 예외 추가	69
<b>7장. 다중 경로 볼륨 관리 .....</b>	<b>72</b>
7.1. 온라인 다중 경로 장치 크기 조정	72

---

7.2. 루트 파일 시스템을 단일 경로 장치에서 다중 경로 장치로 이동	73
7.3. 스왑 파일 시스템을 단일 경로 장치에서 다중 경로 장치로 이동	74
7.4. DMSETUP 명령을 사용하여 장치 매핑 항목 확인	77
7.5. 다중 경로 데몬 관리	77
<b>8장. 스토리지 장치 제거</b> .....	<b>79</b>
8.1. 안전한 스토리지 장치 제거	79
8.2. 블록 장치 및 관련 메타데이터 제거	80
<b>9장. DM MULTIPATH 문제 해결</b> .....	<b>85</b>
9.1. QUEUE_IF_NO_PATH 기능 문제 해결	85
9.2. 다중 경로 대화식 콘솔을 사용하여 문제 해결	86
<b>10장. EH_DEADLINE으로 스토리지 오류 복구의 최대 시간 구성</b> .....	<b>88</b>
10.1. EH_DEADLINE 매개변수	88
10.2. EH_DEADLINE 매개변수 설정	89



## 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. 장치 매핑 다중 경로 개요

DM Multipath가 다음을 제공합니다.

### redundancy

DM Multipath는 활성/수동 구성에서 장애 조치를 제공할 수 있습니다. 활성/수동 구성에서 경로의 하위 집합만 I/O에 대해 언제든지 사용됩니다. 케이블, 스위치 또는 컨트롤러와 같은 I/O 경로의 요소가 실패하면 DM Multipath가 대체 경로로 전환됩니다.



### 참고

경로 수는 설정에 따라 다릅니다. 일반적으로 DM Multipath 설정에는 스토리지에 대한 2, 4 또는 8 경로가 있지만 이는 일반적인 설정이며 경로에 다른 숫자가 가능합니다.

### 성능 개선

DM Multipath는 라운드 로빈 방식으로 경로를 통해 I/O가 분산되는 활성/활성 모드로 구성할 수 있습니다. 일부 구성에서 DM Multipath는 I/O 경로에서 로드를 감지하고 부하를 동적으로 리밸런싱할 수 있습니다.

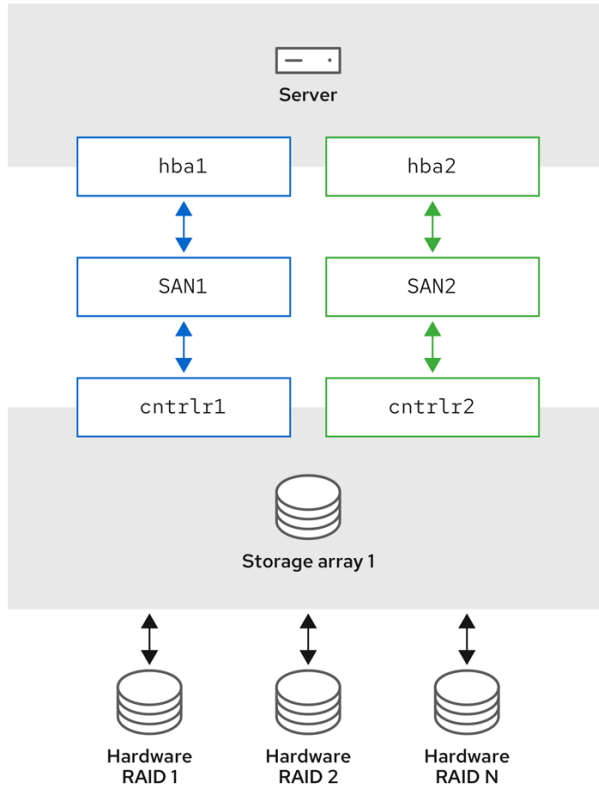
### 1.1. 하나의 RAID 장치가 있는 ACTIVE/PASSIVE 다중 경로 구성

이 구성에는 서버에 두 개의 호스트 버스 어댑터(HBA)와 두 개의 RAID 컨트롤러가 있습니다. 다음은 이 구성에서 발생할 수 있는 오류입니다.

- HBA 실패
- 파이버 채널 케이블 오류
- SAN 전환 실패
- 배열 컨트롤러 포트 실패

DM Multipath가 구성된 경우 이러한 포인트 중 하나라도 실패하면 DM Multipath가 대체 I/O 경로로 전환합니다. 다음 이미지는 서버에서 RAID 장치로의 I/O 경로가 두 개 있는 구성을 설명합니다. 여기에 **hba1,SAN1** 및 **cntrlr1** 을 통과하는 하나의 I/O 경로가 있으며 **hba2,SAN2** 및 **cntrlr2** 를 통과하는 두 번째 I/O 경로가 있습니다.

그림 1.1. 하나의 RAID 장치가 있는 Active/Passive 다중 경로 구성

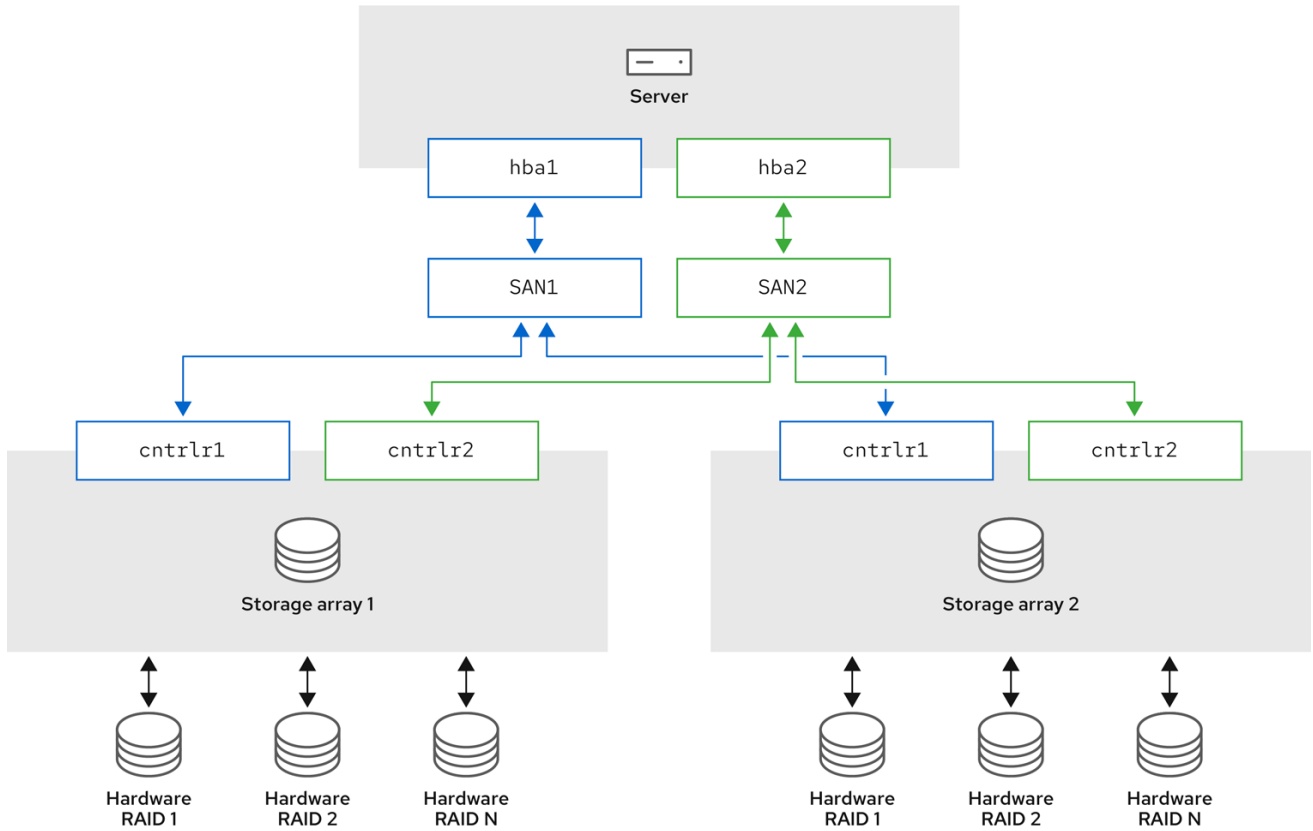


315\_RHEL\_0323

## 1.2. 두 개의 RAID 장치가 있는 ACTIVE/PASSIVE 다중 경로 구성

이 구성에는 서버에 두 개의 HBA, SAN 스위치 두 개, RAID 컨트롤러가 각각 두 개의 RAID 장치가 있습니다. DM Multipath가 구성된 경우 RAID 장치에 대한 I/O 경로 중 하나에서 오류가 발생하면 DM Multipath가 해당 장치의 대체 I/O 경로로 전환합니다. 다음 이미지는 각 RAID 장치에 대한 두 개의 I/O 경로가 있는 구성을 설명합니다. 여기에서 각 RAID 장치에 대한 두 개의 I/O 경로가 있습니다.

그림 1.2. 두 개의 RAID 장치가 있는 Active/Passive 다중 경로 구성

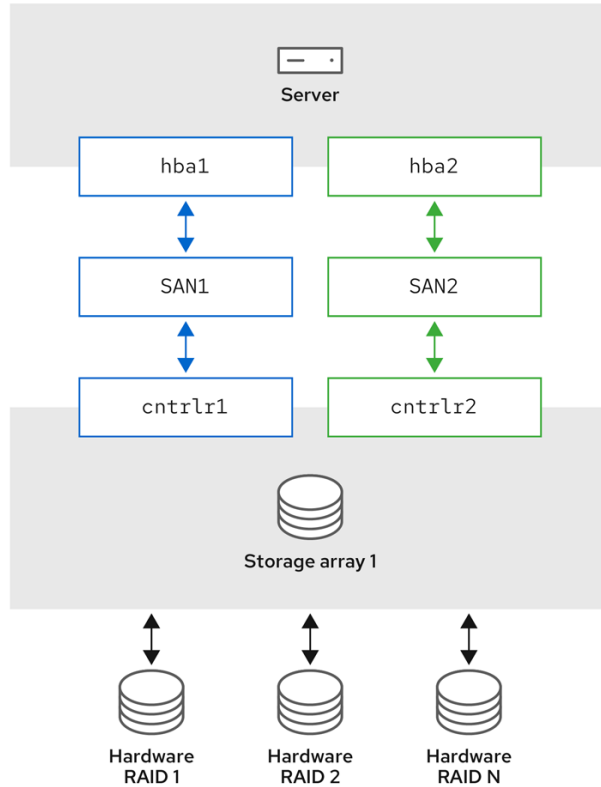


315\_RHEL\_0323

### 1.3. 하나의 RAID 장치가 있는 ACTIVE/ACTIVE 다중 경로 구성

이 구성에는 서버에 두 개의 HBA, SAN 스위치 2개, RAID 컨트롤러 두 개가 있습니다. 다음 이미지는 서버에서 스토리지 장치로의 두 개의 I/O 경로가 있는 구성을 설명합니다. 여기에서 I/O를 이 두 경로에 분산할 수 있습니다.

그림 1.3. 하나의 RAID 장치가 있는 Active/Active 다중 경로 구성



315\_RHEL\_0323

### 1.4. DM MULTIPATH 구성 요소

다음 표에서는 DM Multipath 구성 요소에 대해 설명합니다.

표 1.1. DM Multipath 구성 요소

구성 요소	설명
<b>dm_multipath</b> 커널 모듈	경로 및 경로 그룹에 대해 I/O를 다시 라우팅하고 장애 조치를 지원합니다.
<b>mpathconf</b> 유틸리티	장치 매핑 다중 경로를 구성하고 활성화합니다.
다중 경로 명령	다중 경로 장치를 나열하고 구성합니다. 또한 블록 장치를 추가할 때마다 <b>udev</b> 에 의해 실행되어 장치가 다중 경로 장치의 일부인지 아닌지를 확인합니다.
<b>multipathd</b> 데몬	다중 경로 장치를 자동으로 생성 및 제거하고 경로를 모니터링합니다. 경로가 실패하고 돌아올 때 다중 경로 장치를 업데이트할 수 있습니다. 다중 경로 장치에 대한 대화형 변경 가능. <b>/etc/multipath.conf</b> 파일에 변경 사항이 있는 경우 서비스를 다시 로드합니다.

<b>kpartx</b> 명령	장치에 있는 파티션에 대한 장치 매핑 장치를 생성합니다. 이 명령은 다중 경로 장치를 생성하기 위해 다중 경로 장치를 생성할 때 <b>udev</b> 에 의해 자동으로 실행됩니다. <b>kpartx</b> 명령은 자체 패키지로 제공되지만 <b>device-mapper-multipath</b> 패키지는 이 패키지에 따라 다릅니다.
<b>multipathpersist</b>	다중 경로 장치에 <b>SCSI-3</b> 영구 예약을 설정합니다. 이 명령은 <b>sg_persist</b> 가 다중 경로가 아닌 SCSI 장치에서 작동하는 방식과 유사하게 작동하지만 다중 경로 장치의 모든 경로에서 영구 예약을 처리합니다. <b>multipathd</b> 와 조정하여 나중에 추가된 경로에 예약이 올바르게 설정되었는지 확인합니다. 이 기능을 사용하려면 <b>reservation_key</b> 특성을 <b>/etc/multipath.conf</b> 파일에 정의해야 합니다. 그렇지 않으면 <b>multipathd</b> 데몬은 새로 검색된 경로 또는 복원 경로에 대한 영구 예약을 확인하지 않습니다.

## 1.5. 다중 경로 명령

**multipath** 명령은 여러 경로를 감지하여 장치로 결합하는 데 사용됩니다. 다중 경로 장치를 관리하는 데 사용할 수 있는 다양한 옵션을 제공합니다.

다음 표에서는 유용할 수 있는 **다중 경로** 명령의 몇 가지 옵션에 대해 설명합니다.

표 1.2. 유용한 다중 경로 명령 옵션

옵션	설명
<b>-l</b>	<b>sysfs</b> 및 장치 매핑에서 수집된 현재 다중 경로 토폴로지를 표시합니다.
<b>-ll</b>	<b>sysfs</b> , 장치 매핑 및 시스템에서 사용 가능한 기타 모든 구성 요소에서 수집한 현재 다중 경로 토폴로지를 표시합니다.
<b>-F 장치</b>	이름이 지정된 다중 경로 장치를 제거합니다.
<b>-F</b>	사용되지 않는 모든 다중 경로 장치를 제거합니다.
<b>-W 장치</b>	지정된 장치의 <b>wwid</b> 를 <b>wwids</b> 파일에서 제거합니다.
<b>-W</b>	현재 다중 경로 장치만 포함하도록 <b>wwids</b> 파일을 재설정합니다.
<b>-r</b>	다중 경로 장치를 강제로 다시 로드합니다.

## 1.6. 다중 경로 토폴로지 표시

경로를 효과적으로 모니터링하거나 다중 경로 문제를 해결하거나 다중 경로 구성이 올바르게 설정되었는지 확인하려면 다중 경로 토폴로지를 표시할 수 있습니다.

### 절차

1. 다중 경로 장치 토폴로지를 표시합니다.

```
# multipath -ll
mpatha (3600d0230000000000e13954ed5f89300) dm-4 WINSYS,SF2372
size=233G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
  `-- 6:0:0:0 sdf 8:80 active ready running
```

출력은 세 부분으로 나눌 수 있습니다. 각 부분은 다음 그룹에 대한 정보를 표시합니다.

- 다중 경로 장치 정보:
  - **mpatha (3600d0230000000000e13954ed5f89300)**: 별칭 (별명과 다른 경우)
  - **DM-4**: dm 장치 이름
  - **CryostatSYS,SF2372**: 공급 업체, 제품
  - **size=233G**: size
  - **features='1 queue\_if\_no\_path'**: features
  - **hwhandler='0'**: hardware handler
  - **Cryostat=rw**: 쓰기 권한
- 경로 그룹 정보:
  - **policy='service-time 0'**: 스케줄링 정책
  - **Prio=1**: 경로 그룹 우선순위
  - **status=active**: 경로 그룹 상태
- 경로 정보:
  - **6:0:0:0**: host:channel:id:lun
  - **sdf**: devnode
  - **8:80**: major:minor numbers
  - **active**: dm status
  - **Ready**: 경로 상태
  - **실행 중**: 온라인 상태  
dm, path 및 online status에 대한 자세한 내용은 [경로 상태를 참조하십시오](#).

다중 경로 장치를 나열, 생성 또는 다시 로드하는 데 사용되는 기타 다중 경로 명령도 장치 토폴로지를 표시합니다. 그러나 일부 정보는 알 수 없고 출력에 **undef** 로 표시될 수 있습니다. 이는 정상적인 동작입니다. **multipath -ll** 명령을 사용하여 올바른 상태를 확인합니다.



## 참고

다중 경로 장치 생성과 같은 특정 경우에는 다중 경로 토폴로지에 매개 변수를 표시하여 작업이 수행되었는지 여부를 나타냅니다. 예를 들어 다음 명령 출력은 다중 경로 장치가 생성되었음을 나타내는 **create:** 매개 변수를 보여줍니다.

```
create: mpatha (3600d0230000000000e13954ed5f89300) undef WINSYS,SF2372
size=233G features='1 queue_if_no_path' hwhandler='0' wp=undef
`-+- policy='service-time 0' prio=1 status=undef
  `-- 6:0:0:0 sdf 8:80 undef ready running
```

## 1.7. 경로 상태

경로 상태는 **/etc/multipath.conf** 파일에 정의된 폴링 간격을 기반으로 **multipathd** 데몬을 통해 주기적으로 업데이트됩니다. 커널 측면에서 **dm** 상태는 경로 상태와 유사합니다. **dm** 상태는 경로 검사기가 완료될 때까지 현재 상태를 유지합니다.

### 경로 상태

#### 준비됨

I/O 경로가 준비되었습니다.

#### faulty, shaky

경로가 다운되어 있습니다.

#### I/O 보류 중

검사기에서 이 경로를 적극적으로 확인하고 있으며 곧 상태가 업데이트됩니다.

#### I/O 시간 제한

시간 초과 전에 검사기에서 **성공/실패** 를 반환하지 않았습니다. 이는 **결함이 있는** 것과 동일하게 취급됩니다.

#### 제거됨

경로는 시스템에서 제거되었으며 곧 다중 경로 장치에서 제거됩니다. 이는 **결함이 있는** 것과 동일하게 취급됩니다.

#### Wild

내부 오류 또는 구성 문제로 인해 **multipathd** 에서 경로 검사를 실행할 수 없었습니다. 이는 **잘못된** 것으로 취급됩니다. 다중 경로는 경로의 많은 작업을 건너뛵니다.

#### 선택되지 않음

경로 검사기가 방금 발견되었으므로 경로 검사기에서 이 경로에서 실행되지 않았으며, 할당된 경로 검사기 또는 경로 검사기에 오류가 발생했습니다. 이 문제는 **Wild** 와 동일하게 취급됩니다.

#### 지연

경로 검사기에서는 경로가 가동 중임을 반환하지만 최근에 경로가 여러 번 실패했으며 이 경우 경로를 지연하도록 다중 경로가 경로를 지연하고 있습니다. 이는 **결함이 있는** 것과 동일하게 취급됩니다.

### DM 상태

#### 활성 상태

준비 및 **유령** 경로 상태로 매핑됩니다.

### Failed

동일한 **dm** 상태가 없는 **i/o pending** 을 제외하고 다른 모든 경로 상태에 매핑됩니다.

### 온라인 상태

#### Running

장치가 활성화되어 있습니다.

#### 오프라인

장치가 비활성화되었습니다.

## 1.8. 추가 리소스

- **multipath(8)** 및 **multipathd(8)** 매뉴얼 페이지
- **/etc/multipath.conf** 파일

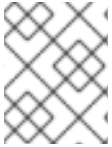


## 2장. 다중 경로 장치

DM Multipath는 기본 장치 상단에 단일 다중 경로 장치를 생성하여 I/O 경로를 논리적으로 구성하는 방법을 제공합니다. DM Multipath가 없으면 I/O 경로가 동일한 스토리지 컨트롤러에 동일한 서버 노드를 연결하는 경우에도 서버 노드의 각 경로를 별도의 장치로 처리합니다.

### 2.1. 다중 경로 장치 식별자

새 장치가 DM Multipath가 제어되면 이러한 장치는 **/dev/mapper/** 및 **/dev/** 디렉토리에 생성됩니다.



#### 참고

**/dev/dm-X** 형식의 모든 장치는 내부용으로만 사용되며 관리자가 직접 사용해서는 안 됩니다.

다음은 다중 경로 장치 이름을 설명합니다.

- **user\_friendly\_names** 구성 옵션이 **no** 로 설정되어 있으면 다중 경로 장치의 이름은 WWID(WWID)로 설정됩니다. 기본적으로 다중 경로 장치의 이름은 WWID로 설정됩니다. 장치 이름은 **/dev/mapper/WWID** 입니다. 또한 **/dev/** 디렉토리에 **/dev/dm-X** 라는 이름으로 생성됩니다.
- 또는 **/etc/multipath.conf** 파일에서 **user\_friendly\_names** 옵션을 **yes** 로 설정할 수 있습니다. 이를 통해 다중 경로 섹션의 별칭을 **mpathN** 형식의 노드 고유 이름으로 설정합니다. 장치 이름은 **/dev/mapper/mpathN** 및 **/dev/dm-X** 입니다. 그러나 다중 경로 장치를 사용하는 모든 노드에서 장치 이름이 같을 수는 없습니다. 마찬가지로 **/etc/multipath.conf** 파일에서 **alias** 옵션을 설정하면 클러스터의 모든 노드에서 이름이 자동으로 일치하지 않습니다.



#### 참고

이 경우 LVM을 사용하여 다중 경로 장치에서 논리 장치를 생성하는 경우 문제가 발생하지 않습니다. 다중 경로 장치 이름을 모든 노드에서 일관되게 유지하려면 **user\_friendly\_names** 옵션을 비활성화하는 것이 좋습니다.

예를 들어 단일 영역의 FC 스위치를 통해 두 개의 포트가 있는 스토리지 컨트롤러에 연결된 노드는 **/dev/sda,/dev/sdb,/dev/sdc,/dev/sddd** 의 네 가지 장치를 확인할 수 있습니다. **DM Multipath**는 다중 경로 구성에 따라 I/O를 기본 장치 4개로 라우팅하는 고유한 WWID를 사용하여 단일 장치를 생성합니다.

다중 경로 장치에는 **user\_friendly\_names** 및 별칭 옵션 외에 다른 속성도 있습니다. **/etc/multipath.conf** 파일의 **multipaths** 섹션에 해당 장치의 항목을 생성하여 특정 다중 경로 장치에 대한 이러한 속성을 수정할 수 있습니다.

#### 추가 리소스

- **multipath(8)** 및 **multipath.conf(8)** 매뉴얼 페이지
- **/etc/multipath.conf** 파일

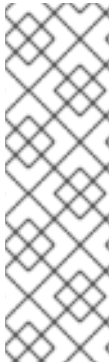
## DM Multipath 구성 요소

### 2.2. 논리 볼륨의 다중 경로 장치

다중 경로 장치를 생성한 후 논리 볼륨 관리자(LVM) 물리 볼륨을 생성할 때 물리 장치 이름을 사용하는 것처럼 다중 경로 장치 이름을 사용할 수 있습니다. 예를 들어 `/dev/mapper/mpatha` 가 다중 경로 장치의 이름인 경우 `pvcreate /dev/mapper/mpatha` 명령은 `/dev/mapper/mpatha` 를 물리 볼륨으로 표시합니다.

다른 LVM 물리 장치를 사용하는 것처럼 LVM 볼륨 그룹을 만들 때 결과 LVM 물리 장치를 사용할 수 있습니다.

`/etc/lvm/lvm.conf` 파일의 모든 `sd` 장치를 필터링하려면 파일의 `devices` 섹션에 `필터 = [ "r/block/", "r/disk/", "r/sd./", "a/./" ]` 필터를 추가하십시오.



#### 참고

파티션을 구성한 전체 장치에 LVM 물리 볼륨을 생성하려고 하면 `pvcreate` 명령이 실패합니다. 모든 블록 장치에 대해 달리 지정하지 않는 경우 `Anaconda` 및 `Kickstart` 설치 프로그램은 빈 파티션 테이블을 만듭니다. 파티션을 생성하는 대신 전체 장치를 사용하려면 장치에서 기존 파티션을 제거합니다. `kpartx -d device` 명령 및 `metadata` 유틸리티를 사용하여 기존 파티션을 제거할 수 있습니다. 시스템에 2Tb보다 큰 블록 장치가 있는 경우 `parted` 유틸리티를 사용하여 파티션을 제거합니다.

활성/수동 다중 경로 배열을 기본 물리 장치로 사용하는 LVM 논리 볼륨을 생성할 때 선택적으로 `/etc/lvm/lvm.conf` 파일에 필터를 포함하여 다중 경로 장치가 있는 디스크를 제외할 수 있습니다. 배열이 I/O를 수신할 때 수동 경로를 자동으로 변경하면 이러한 장치가 필터링되지 않는 경우 LVM에서 수동 경로를 스캔할 때마다 다중 경로 장애 조치 및 `failback`이기 때문입니다.

커널은 사용할 올바른 하드웨어 핸들러를 자동으로 탐지하여 활성/수동 상태를 변경합니다. 상태를 변경하기 위해 개입이 필요한 활성/수동 경로의 경우 다중 경로는 필요에 따라 이 하드웨어 처리기를 자동으로 사용합니다. 커널이 사용할 올바른 하드웨어 핸들러를 자동으로 감지하지 않으면 `"hardware_handler"` 옵션으로 `multipath.conf` 파일에 사용할 하드웨어 처리기를 구성할 수 있습니다. 패시브 경로를 활성화하는 명령이 필요한 활성/수동 배열의 경우 LVM은 이 문제가 발생할 때 경고 메시지를 출력합니다.

구성에 따라 LVM에서 다음 메시지를 출력할 수 있습니다.

- **LUN이 준비되지 않음:**  
  
end\_request: I/O error, dev sdc, sector 0  
sd 0:0:0:3: Device not ready: <6>: Current: sense key: Not Ready  
Add. Sense: Logical unit not ready, manual intervention required

- **읽기 실패:**  
  
/dev/sde: read failed after 0 of 4096 at 0: Input/output error

언급된 오류의 원인은 다음과 같습니다.

- 다중 경로는 시스템에 활성/수동 경로를 제공하는 스토리지 장치에 설정되지 않습니다.
- 경로는 다중 경로 장치를 통해 직접 액세스할 수 있습니다.

추가 리소스

- **LVM.conf** 도움말 페이지
- **DM Multipath** 구성 요소

### 3장. DM MULTIPATH 구성

**mpathconf** 유틸리티를 사용하여 **DM Multipath**를 설정할 수 있습니다. 이 유틸리티는 다음 시나리오에 따라 **/etc/multipath.conf** 다중 경로 구성 파일을 생성하거나 편집합니다.

- **/etc/multipath.conf** 파일이 이미 있으면 **mpathconf** 유틸리티에서 편집합니다.
- **/etc/multipath.conf** 파일이 없으면 **mpathconf** 유틸리티에서 **/etc/multipath.conf** 파일을 처음부터 생성합니다.

#### 3.1. DEVICE-MAPPER-MULTIPATH 패키지 확인

시스템에서 **DM Multipath**를 설정하기 전에 시스템이 최신 버전이며 **device-mapper-multipath** 패키지가 포함되어 있는지 확인합니다.

##### 절차

1. 시스템에 **device-mapper-multipath** 패키지가 포함되어 있는지 확인합니다.

```
# rpm -q device-mapper-multipath
device-mapper-multipath-current-package-version
```

시스템에 패키지가 포함되어 있지 않으면 다음이 출력됩니다.

```
package device-mapper-multipath is not installed
```

2. 시스템에 패키지가 포함되어 있지 않으면 다음 명령을 실행하여 설치합니다.

```
# dnf install device-mapper-multipath
```

#### 3.2. DM MULTIPATH를 사용하여 기본 장애 조치 구성 설정

기본 페일오버 구성에 대해 **DM Multipath**를 설정하고 **multipathd** 데몬을 시작하기 전에 **/etc/multipath.conf** 파일을 편집할 수 있습니다.

##### 사전 요구 사항

- 관리 액세스.

## 절차

1. 다중 경로 구성 파일을 활성화하고 초기화합니다.

```
# mpathconf --enable
```

2. 선택 사항: `/etc/multipath.conf` 파일을 편집합니다.

대부분의 기본 설정은 페일오버 로 설정된 `path_grouping_policy` 를 포함하여 이미 구성되어 있습니다.

3. 선택 사항: 다중 경로 장치의 기본 이름 지정 형식은 `/dev/mapper/mpathn` 형식으로 설정됩니다. 다른 이름 지정 형식을 선호하는 경우:
  - a. `mpath_n_` 사용자 친화적인 이름 체계 대신 다중 경로 장치 **WWID**를 이름으로 사용하도록 **DM Multipath**를 구성합니다.

```
# mpathconf --enable --user_friendly_names n
```

- b. **DM Multipath** 데몬 설정을 다시 로드합니다.

```
# systemctl reload multipathd.service
```

4. **DM Multipath** 데몬을 시작합니다.

```
# systemctl start multipathd.service
```

## 검증

- **DM Multipath** 데몬이 문제 없이 실행되고 있는지 확인합니다.

```
# systemctl status multipathd.service
```

- 다중 경로 장치의 이름 지정 형식을 확인합니다.

```
# ls /dev/mapper/
```

### 3.3. 다중 경로 장치를 생성할 때 로컬 디스크 무시

일부 시스템에는 내부 디스크에 대한 로컬 **SCSI** 카드가 있으며 이러한 장치에는 **DM Multipath**를 사용하지 않는 것이 좋습니다. **find\_multipaths** 구성 매개변수를 **yes**로 설정하면 이러한 장치에서 다중 경로를 비활성화할 필요가 없습니다.

**find\_multipaths** 구성 매개변수를 **yes**로 설정하지 않으면 다음 절차를 사용하여 다중 경로를 구성할 때 로컬 디스크를 무시하도록 **DM Multipath** 구성 파일을 수정할 수 있습니다.

#### 절차

1.

장치의 모델, 경로 또는 공급 업체와 같은 알려진 매개 변수를 사용하여 내부 디스크를 식별하고 다음 옵션 중 하나를 사용하여 **WWID**를 결정합니다.

- 기존 다중 경로 장치를 표시합니다.

```
# multipath -v2 -l

mpatha (WDC_WD800JD-75MSA3_WD-WMAM9FU71040) dm-2 ATA,WDC WD800JD-75MS
size=33 GB features="0" hwhandler="0" wp=rw
`-+- policy='round-robin 0' prio=0 status=active
|- 0:0:0:0 sda 8:0 active undef running
```

- **DM Multipath**가 생성할 수 있는 추가 다중 경로 장치를 표시합니다.

```
# multipath -v2 -d

: mpatha (WDC_WD800JD-75MSA3_WD-WMAM9FU71040) dm-2 ATA,WDC WD800JD-75MS
size=33 GB features="0" hwhandler="0" wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
|- 0:0:0:0 sda 8:0 undef ready running
```

- 장치 정보를 표시합니다.

```
# multipathd show paths raw format "%d %w" | grep sda
sda WDC_WD800JD-75MSA3_WD-WMAM9FU71040
```

이 예에서 `/dev/sda` 는 내부 디스크이고 `WWID`는 `WDC_WD800JD-75MSA3_WD-WMAM9FU71040` 입니다.

2.

`WWID` 특성을 사용하여 이 장치를 무시하도록 `/etc/multipath.conf` 파일의 `blacklist` 섹션을 편집합니다.

```
blacklist {
    wwid WDC_WD800JD-75MSA3_WD-WMAM9FU71040
}
```



주의

`sda`와 같은 `devnode` 매개변수를 사용하여 장치를 식별할 수는 있지만 `/dev/sda` 가 재부팅 시 동일한 장치를 참조하는 것을 보장하지 않기 때문에 안전한 절차가 아닙니다.

3.

`/etc/multipath.conf` 파일에서 구성 오류가 있는지 확인합니다.

```
# multipath -t > /dev/null
```

전체 보고서를 보려면 명령 출력을 삭제하지 마십시오.

```
# multipath -t
```

4.

디스크가 `initramfs` 에 포함된 경우 `initramfs`를 다시 생성합니다. 자세한 내용은 [initramfs에 서 다중 경로 구성을 참조하십시오](#).

5.

`multipathd` 데몬을 재구성하여 `/etc/multipath.conf` 파일을 다시 로드합니다.

```
# systemctl reload multipathd
```



## 참고

사용 중인 경우 로컬 디스크 상단에 있는 다중 경로 장치를 제거할 수 없습니다. 이러한 장치를 무시하려면 장치의 모든 사용자를 중지합니다. 예를 들어 위에 있는 파일 시스템을 마운트 해제하고 이를 사용하여 논리 볼륨을 비활성화하면 됩니다. 이 옵션을 사용할 수 없는 경우 시스템을 재부팅하여 다중 경로 장치를 제거할 수 있습니다.

## 검증

1.

내부 디스크가 무시되고 다중 경로 출력에 표시되지 않는지 확인합니다.

- 

다중 경로 장치를 나열합니다.

```
# multipath -v2 -l
```

- 

DM Multipath에서 생성할 수 있는 추가 장치를 나열합니다.

```
# multipath -v2 -d
```

## 추가 리소스

- 

[multipath.conf\(5\) 매뉴얼 페이지](#)

## 3.4. DM MULTIPATH를 사용하여 추가 스토리지 구성

기본적으로 DM Multipath에는 DM Multipath를 지원하는 가장 일반적인 스토리지 어레이에 대한 기본 제공 구성이 포함되어 있습니다. 스토리지 배열에 구성이 없는 경우 `/etc/multipath.conf` 파일을 편집하여 추가할 수 있습니다.

## 참고

초기 구성 중에 추가 스토리지 장치를 추가하여 설정을 예상 요구 사항에 맞춥니다. DM Multipath는 나중에 확장성 또는 업그레이드를 위해 장치를 추가할 수 있지만 이 접근 방식에서는 호환성을 보장하기 위해 구성을 조정해야 할 수 있습니다.

## 사전 요구 사항

- 

관리 액세스.



## 절차

1. 기본 구성 값 및 지원되는 장치를 확인합니다.

```
# multipathd show config
```

- 2.

**/etc/multipath.conf** 파일을 편집하여 다중 경로 구성을 설정합니다.

### 예 3.1. HP OPEN-V 스토리지 장치에 대한 DM 다중 경로 구성

```
# Set default configurations for all devices managed by DM Multipath

defaults {
    # Enable user-friendly names for devices
    user_friendly_names yes
}

devices {
    # Define configuration for HP OPEN-V storage
    device {
        vendor "HP"
        pproduct "OPEN-V"
        no_path_retry 18
    }
}
```

3. 변경 사항을 저장하고 편집기를 종료합니다.
4. 새 장치를 스캔하여 다중 경로 장치 목록을 업데이트합니다.

```
# multipath -r
```

## 검증

- 다중 경로 장치가 올바르게 인식되었는지 확인합니다.

```
# multipath -ll
```

### 3.5. INITRAMFS에서 다중 경로 구성

**initramfs** 파일 시스템에서 다중 경로를 설정하는 것은 특히 중복성 및 로드 밸런싱이 필요한 시나리오에서 원활한 스토리지 기능에 필요합니다. 이 설정을 통해 부팅 프로세스 초기에 다중 경로 장치를 사용할

수 있습니다. 이는 스토리지 설정의 무결성을 유지하고 잠재적인 문제를 방지하는 데 중요합니다.

#### 사전 요구 사항

- 관리 액세스.
- 시스템에 **DM** 다중 경로를 구성합니다.

#### 절차

1. 다중 경로 구성 파일을 사용하여 **initramfs** 파일 시스템을 다시 빌드합니다.

```
# dracut --force --add multipath
```

#### 참고

**initramfs** 에서 다중 경로를 사용하고 구성 파일을 수정하는 경우 **tale effect**를 변경하려면 **initramfs** 를 다시 빌드해야 합니다. 루트 장치가 다중 경로를 사용하는 경우 **dracut** 명령은 **initramfs** 에 **multipath** 모듈을 자동으로 포함합니다.

2. 선택 사항: **initramfs** 의 다중 경로가 더 이상 필요하지 않은 경우 다음을 수행합니다.

- a. 다중 경로 구성 파일을 제거합니다.

```
# rm /etc/dracut.conf.d/multipath.conf
```

- b. 추가된 다중 경로 구성을 사용하여 **initramfs** 를 다시 빌드합니다.

```
# dracut --force --omit multipath
```

#### 검증

- 다중 경로 관련 파일 및 구성이 있는지 확인합니다.

```
# lsinitrd /path/to/initramfs.img -m | grep multipath
```

#### 참고

제공되는 **verification** 단계는 성공 표시를 제공할 수 있지만 구성이 예상대로 작동하는지 확인하려면 최종 테스트 부팅을 사용하는 것이 좋습니다.

- 재부팅 후 다중 경로 장치가 올바르게 인식되었는지 확인합니다.

```
# multipath -ll
```

## 4장. NVMe 장치에서 멀티패스 활성화

FCP(Fibre Channel)와 같은 패브릭 전송을 통해 시스템에 연결된 NVMe™(Non-volatile Memory Express™) 장치를 멀티패스할 수 있습니다. 다중 경로 솔루션 중에서 선택할 수 있습니다.

### 4.1. 네이티브 NVMe 다중 경로 및 DM MULTIPATH

NVMe(Non-Volatile Memory Express™) 장치는 기본 다중 경로 기능을 지원합니다. NVMe에서 멀티패스를 구성할 때 표준 DM Multipath 프레임워크와 기본 NVMe 다중 경로 중에서 선택할 수 있습니다.

DM Multipath 및 기본 NVMe 다중 경로 모두 NVMe 장치의 ANA(Asymmetric Namespace Access) 다중 경로 체계를 지원합니다. ANA는 컨트롤러와 호스트 간의 최적화된 경로를 식별하고 성능을 향상시킵니다.

네이티브 NVMe 멀티패스가 활성화되면 모든 NVMe 장치에 전역적으로 적용됩니다. 향상된 성능을 제공할 수 있지만 DM Multipath에서 제공하는 모든 기능이 포함되어 있지는 않습니다. 예를 들어 기본 NVMe 다중 경로에서는 numa 및 round-robin 경로 선택 방법만 지원합니다.

기본적으로 NVMe 멀티패스는 Red Hat Enterprise Linux 9에서 활성화되며 권장되는 다중 경로 솔루션입니다.

### 4.2. NVMe 장치에서 DM MULTIPATH 활성화

nvme\_core.multipath 옵션의 기본 커널 설정은 Y로 설정되어 있습니다. 즉, 기본 NVMe(Non-volatile Memory Express™) 다중 경로가 활성화됩니다. 네이티브 NVMe 멀티패스를 비활성화하여 연결된 NVMe 장치에서 DM Multipath를 활성화할 수 있습니다.

#### 사전 요구 사항

- NVMe 장치가 시스템에 연결됩니다. 자세한 내용은 [패브릭 장치를 통한 NVMe 개요](#) 를 참조하십시오.

#### 절차

1. 네이티브 NVMe 멀티패스가 활성화되어 있는지 확인합니다.

```
# cat /sys/module/nvme_core/parameters/multipath
```

명령은 다음 중 하나를 표시합니다.

**N**

네이티브 NVMe 멀티패스가 비활성화되어 있습니다.

**Y**

네이티브 NVMe 다중 경로가 활성화되어 있습니다.

2.

네이티브 NVMe 멀티패스를 활성화하면 다음 방법 중 하나를 사용하여 비활성화합니다.

•

커널 옵션 사용:

a.

**nvme\_core.multipath=N** 옵션을 명령줄에 추가합니다.

```
# grubby --update-kernel=ALL --args="nvme_core.multipath=N"
```

b.

**64비트 IBM Z** 아키텍처에서 부팅 메뉴를 업데이트합니다.

```
# zipl
```

c.

시스템을 재부팅합니다.

•

커널 모듈 구성 파일 사용:

a.

다음 콘텐츠를 사용하여 **/etc/modprobe.d/nvme\_core.conf** 구성 파일을 만듭니다.

```
options nvme_core multipath=N
```

b.

**initramfs** 파일을 백업합니다.

```
# cp /boot/initramfs-$(uname -r).img /boot/initramfs-$(uname r).bak.$(date +%m%d-%H%M%S).img
```

c.

**initramfs** 를 다시 빌드합니다.

```
# cp /boot/initramfs-$(uname -r).img /boot/initramfs-$(uname -r).bak.$(date
+%m-%d-%H%M%S).img
# dracut --force --verbose
```

d.

시스템을 재부팅합니다.

3.

**DM Multipath**를 활성화합니다.

```
# systemctl enable --now multipathd.service
```

4.

모든 사용 가능한 경로에 I/O 배포. **/etc/multipath.conf** 파일에 다음 내용을 추가합니다.

```
devices {
  device {
    vendor "NVME"
    product ".*"
    path_grouping_policy group_by_prio
  }
}
```



참고

**DM Multipath**가 **NVMe** 장치를 관리할 때 **/sys/class/nvme-subsys0/iopolicy** 구성 파일은 I/O 배포에 영향을 미치지 않습니다.

5.

**multipathd** 서비스를 다시 로드하여 구성 변경 사항을 적용합니다.

```
# multipath -r
```

검증

•

네이티브 **NVMe** 멀티패스가 비활성화되어 있는지 확인합니다.

```
# cat /sys/module/nvme_core/parameters/multipath
N
```

- **DM** 다중 경로가 **nvme** 장치를 인식하는지 확인합니다.

```
# multipath -l

eui.00007a8962ab241100a0980000d851c8 dm-6 NVME,NetApp E-Series
size=20G features='0' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=0 status=active
  |- 0:10:2:2 nvme0n2 259:3 active undef running
`-+- policy='service-time 0' prio=0 status=enabled
  |- 4:11:2:2 nvme4n2 259:28 active undef running
`-+- policy='service-time 0' prio=0 status=enabled
  |- 5:32778:2:2 nvme5n2 259:38 active undef running
`-+- policy='service-time 0' prio=0 status=enabled
  |- 6:32779:2:2 nvme6n2 259:44 active undef running
```

#### 추가 리소스

- [커널 명령줄 매개변수 구성](#)
- [DM 다중 경로 구성](#)

### 4.3. 기본 NVME 다중 경로 활성화

네이티브 NVMe 멀티패스가 비활성화된 경우 다음 솔루션을 사용하여 활성화할 수 있습니다.

#### 사전 요구 사항

- NVMe 장치가 시스템에 연결됩니다. 자세한 내용은 [패브릭 장치를 통한 NVMe 개요](#) 를 참조 하십시오.

#### 절차

1. 커널에서 네이티브 NVMe 멀티패스가 활성화되어 있는지 확인합니다.

```
# cat /sys/module/nvme_core/parameters/multipath
```

명령은 다음 중 하나를 표시합니다.

N

네이티브 NVMe 멀티패스가 비활성화되어 있습니다.

Y

네이티브 NVMe 다중 경로가 활성화되어 있습니다.

2.

네이티브 NVMe 멀티패스가 비활성화된 경우 다음 방법 중 하나를 사용하여 활성화합니다.

- 

커널 옵션 사용:

a.

커널 명령 줄에서 `nvme_core.multipath=N` 옵션을 제거합니다.

```
# grubby --update-kernel=ALL --remove-args="nvme_core.multipath=N"
```

b.

64비트 IBM Z 아키텍처에서 부팅 메뉴를 업데이트합니다.

```
# zipl
```

c.

시스템을 재부팅합니다.

- 

커널 모듈 구성 파일 사용:

a.

`/etc/modprobe.d/nvme_core.conf` 설정 파일을 삭제합니다.

```
# rm /etc/modprobe.d/nvme_core.conf
```

b.

`initramfs` 파일을 백업합니다.

```
# cp /boot/initramfs-$(uname -r).img /boot/initramfs-$(uname -r).bak.$(date +%m-%d-%H%M%S).img
```

c.

`initramfs` 를 다시 빌드합니다.

```
# dracut --force --verbose
```



d.

시스템을 재부팅합니다.

3.

선택 사항: 실행 중인 시스템에서 NVMe 장치에서 I/O 정책을 변경하여 사용 가능한 모든 경로에 I/O를 배포합니다.

```
# echo "round-robin" > /sys/class/nvme-subsystem/nvme-subsys0/iopolicy
```

4.

선택 사항: udev 규칙을 사용하여 I/O 정책을 영구적으로 설정합니다. 다음 콘텐츠를 사용하여 /etc/udev/rules.d/71-nvme-io-policy.rules 파일을 만듭니다.

```
ACTION=="add|change", SUBSYSTEM=="nvme-subsystem", ATTR{iopolicy}="round-robin"
```

## 검증

1.

시스템이 NVMe 장치를 인식하는지 확인합니다. 다음 예제에서는 두 개의 NVMe 네임스페이스가 있는 페브리크 스토리지 하위 시스템에 연결된 NVMe가 있다고 가정합니다.

```
# nvme list
```

Node Format	SN FW Rev	Model	Namespace Usage
/dev/nvme0n1	a34c4f3a0d6f5cec	Linux	1 250.06 GB /
250.06 GB	512 B + 0 B	4.18.0-2	
/dev/nvme0n2	a34c4f3a0d6f5cec	Linux	2 250.06 GB /
250.06 GB	512 B + 0 B	4.18.0-2	

2.

연결된 NVMe 하위 시스템을 모두 나열합니다.

```
# nvme list-subsys
```

```
nvme-subsys0 - NQN=testnqn
```

```
\
```

```
+ - nvme0 fc traddr=nn-0x20000090fadd597a:pn-0x10000090fadd597a host_traddr=nn-0x20000090fac7e1dd:pn-0x10000090fac7e1dd live
+ - nvme1 fc traddr=nn-0x20000090fadd5979:pn-0x10000090fadd5979 host_traddr=nn-0x20000090fac7e1dd:pn-0x10000090fac7e1dd live
+ - nvme2 fc traddr=nn-0x20000090fadd5979:pn-0x10000090fadd5979 host_traddr=nn-0x20000090fac7e1de:pn-0x10000090fac7e1de live
+ - nvme3 fc traddr=nn-0x20000090fadd597a:pn-0x10000090fadd597a host_traddr=nn-0x20000090fac7e1de:pn-0x10000090fac7e1de live
```

활성 전송 유형을 확인합니다. 예를 들어 **nvme0 fc** 은 장치가 파이버 채널 전송을 통해 연결되고 **nvme tcp** 는 장치가 **TCP**를 통해 연결됨을 나타냅니다.

3.

커널 옵션을 편집한 경우 커널 명령 줄에서 네이티브 **NVMe** 멀티패스가 활성화되어 있는지 확인합니다.

```
# cat /proc/cmdline  
BOOT_IMAGE=[...] nvme_core.multipath=Y
```

4.

**I/O** 정책을 변경한 경우 **NVMe** 장치에서 라운드 로빈 이 활성 **I/O** 정책인지 확인합니다.

```
# cat /sys/class/nvme-subsystem/nvme-subsys0/iopolicy  
round-robin
```

#### 추가 리소스

•

[커널 명령줄 매개변수 구성](#)

## 5장. DM MULTIPATH 구성 파일 수정

기본적으로 **DM Multipath**는 다중 경로의 가장 일반적인 사용에 대한 구성 값을 제공합니다. 또한 **DM Multipath**에는 **DM Multipath**를 자체적으로 지원하는 가장 일반적인 스토리지 어레이 지원이 포함됩니다. `/etc/multipath.conf` 구성 파일을 편집하여 **DM Multipath**의 기본 구성 값을 덮어쓸 수 있습니다. 필요한 경우 기본 스토리지 어레이에서 지원되지 않는 스토리지를 구성 파일에 추가할 수도 있습니다.

지원되는 장치를 포함하여 기본 구성 값에 대한 정보는 다음 명령 중 하나를 실행합니다.

```
# multipathd show config
# multipath -t
```



### 참고

**initramfs** 파일 시스템에서 다중 경로를 실행하고 다중 경로 구성 파일을 변경한 경우 **initramfs** 파일 시스템을 다시 빌드하여 변경 사항을 적용해야 합니다.

다중 경로 구성 파일에서 구성에 필요한 섹션만 지정하거나 기본값에서 변경해야 합니다. 환경과 관련이 없거나 기본값을 재정의할 필요가 없는 파일의 섹션이 있는 경우 초기 파일에 있으므로 주석 처리된 상태로 둘 수 있습니다.

구성 파일은 정규식 설명 구문을 허용합니다.

### 5.1. 구성 파일 개요

다중 경로 구성 파일은 다음 섹션으로 나뉩니다.

#### **blacklist**

다중 경로로 간주되지 않는 특정 장치 목록입니다.

#### **blacklist\_exceptions**

**blacklist** 섹션의 매개변수에 따라 무시되는 다중 경로 장치 목록입니다.

#### **defaults**

**DM Multipath**의 일반적인 기본 설정입니다.

## 다중 경로

개별 다중 경로 장치의 특성에 대한 설정입니다. 이러한 값은 구성 파일의 덮어쓰기, 장치 및 **defaults** 섹션에 지정된 항목을 덮어씁니다.

## 장치

개별 스토리지 컨트롤러에 대한 설정입니다. 이러한 값은 구성 파일의 **defaults** 섹션에 지정된 항목을 덮어씁니다. 기본적으로 지원되지 않는 스토리지 어레이를 사용하는 경우 배열의 **devices** 하위 섹션을 생성해야 할 수 있습니다.

## 덮어쓰기

모든 장치에 적용되는 설정입니다. 이러한 값은 장치 및 구성 파일의 **defaults** 섹션에 지정된 항목을 덮어씁니다.

시스템에서 다중 경로 장치의 속성을 결정할 때 다음 순서로 **multipath.conf** 파일의 별도의 섹션 설정을 확인합니다.

1. **multipaths** 섹션
2. **overrides** 섹션
3. **devices** 섹션
4. **defaults** 섹션

## 5.2. 구성 파일 기본값

`/etc/multipath.conf` 설정 파일에는 **defaults** 섹션이 포함되어 있습니다. 이 섹션에는 **DM(Device Mapper) Multipath**의 기본 구성이 포함되어 있습니다. 기본값은 초기 장치 설정에 따라 다를 수 있습니다.

기본 구성을 보는 방법은 다음과 같습니다.

- 다중 경로 장치에 시스템을 설치하면 기본 다중 경로 구성이 자동으로 적용됩니다. 기본 구성에는 다음이 포함됩니다.

- 기본 구성 값의 전체 목록의 경우 `multipath -t` 또는 `multipathd show config` 명령을 실행합니다.
- 설명이 포함된 구성 옵션 목록은 `multipath.conf` 도움말 페이지를 참조하십시오.
- 설치 중에 다중 경로를 설정하지 않은 경우 `multipathconf --enable` 명령을 실행하여 기본 구성을 가져옵니다.

다음 표에서는 `multipath.conf` 구성 파일의 `defaults` 섹션에 설정된 속성을 설명합니다. `multipaths` 섹션에 지정된 속성은 `devices` 섹션의 값보다 우선 순위가 높습니다. `devices` 섹션에 지정된 속성은 기본값보다 우선 순위가 높습니다. `overrides` 섹션을 사용하여 `devices` 섹션에 해당 장치 유형에 기본 제공 구성 항목이 있는 경우에도 모든 장치 유형에 대한 속성 값을 설정합니다. `overrides` 섹션에는 필수 속성이 없습니다. 그러나 이 섹션에 설정된 모든 속성은 `devices` 또는 `defaults` 섹션의 값보다 우선합니다.

표 5.1. 다중 경로 구성 기본값

속성	설명
<code>polling_interval</code>	두 경로 점검(초) 사이의 간격을 지정합니다. 제대로 작동하는 경로의 경우 검사 간격이 <code>max_polling_interval</code> 로 점차 증가합니다. 기본값은 <b>5</b> 입니다.
<code>max_polling_interval</code>	두 경로 점검(초) 사이의 간격의 최대 길이를 지정합니다. 기본값은 <b>4 * poll_interval</b> 입니다.
<code>find_multipaths</code>	다중 경로 장치 설정 모드를 정의합니다. 사용 가능한 값은 다음과 같습니다.  <b>아니요:</b> <code>find_multipaths</code> 가 .로 설정되어 있으면 다중 경로는 엄격한 값과 마찬가지로 규칙을 적용하고 <code>multipathd</code> 데몬은 <b>greedy</b> 값과 마찬가지로 규칙을 적용합니다.  <b>예:</b> 동일한 WWID(world Wide Identifier)를 사용하는 <b>블랙리스트</b> 에 없는 장치가 있거나 다중 경로 장치가 있기 전에 장치 WWID를 사용하여 다중 경로 장치를 생성한 경우 장치는 다중 경로 장치 경로로 취급됩니다.  <b>greedy:</b> <code>multipath d</code> 와 다중 경로 모두 차단 목록이 없는 모든 장치를 다중 경로 장치 경로로 처리합니다.

속성	설명
	<p><b>smart:</b> Multipath는 차단 목록이 없는 모든 장치가 다중 경로 장치 경로임을 자동으로 고려합니다. 두 번째 경로인 동일한 WWID가 <b>find_multipaths_timeout</b>에 설정된 시간 내에 나타나지 않으면 다중 경로에서 장치를 해제하고 나머지 시스템에서 사용할 수 있습니다. <b>multipathd</b> 데몬은 <b>yes</b> 값을 사용하여 로 규칙을 적용합니다.</p> <p><b>Strict:</b> WWID 장치가 있는 다중 경로 장치를 생성하는 경우 이 값은 장치를 다중 경로 경로로만 처리합니다.</p> <p>기본값은 <b>OFF</b>입니다. 기본 <b>multipath.conf</b> 파일은 <b>find_multipaths</b>를 <b>yes</b>로 설정합니다.</p>
<b>find_multipaths_timeout</b>	<p><b>find_multipaths</b> 스마트가 설정된 경우 첫 번째 경로를 감지한 후 추가 경로를 대기하는 시간(초)을 나타냅니다. 가능한 값은 다음과 같습니다.</p> <p><b>positive 값:</b> 양수를 사용하여 설정하면 차단되지 않은 모든 장치에 시간 초과가 적용됩니다.</p> <p><b>음수 값으로</b> 설정하면 시간 제한이 기본 제공 테이블 또는 장치 섹션에 있는 다중 경로 하드웨어 테이블에 항목이 있는 알려진 <b>장치에</b>만 적용됩니다. 다른 알 수 없는 장치는 부팅 지연을 방지하기 위해 1초의 시간 초과를 사용합니다.</p> <p><b>0:</b> 시스템이 이 속성에 내장된 기본값을 적용합니다.</p> <p>알려진 하드웨어의 기본값은 <b>-10</b>입니다. 즉, 알려진 장치에는 10초의 시간 초과가 있습니다. 알 수 없는 장치에는 1초의 타임아웃이 있습니다. <b>find_multipaths</b> 속성에 <b>smart</b>가 아닌 값이 있는 경우 이 속성이 적용되지 않습니다.</p>
<b>uxsock_timeout</b>	<p><b>다중 경로</b> 대화형 명령의 시간 초과를 밀리초 단위로 설정합니다.</p> <p>많은 장치가 있는 시스템의 경우 <b>multipathd</b> 대화형 명령이 시간 초과되어 실패할 수 있습니다. 이 문제가 발생하면 이 시간 제한을 늘려 문제를 해결합니다.</p> <p>기본값은 <b>4000</b>입니다.</p>
<b>reassign_maps</b>	<p>장치-매핑 맵 재 할당을 활성화합니다. 이 옵션을 사용하면 <b>multipathd</b> 데몬이 기본 블록 장치가 아닌 항상 다중 경로 장치를 가리키도록 기존 장치 매핑 맵을 다시 매핑합니다. 가능한 값은 <b>yes</b> 및 <b>no</b>입니다. 기본값은 <b>no</b>입니다.</p>
<b>상세 정보 표시</b>	<p>기본 세부 정보 표시 값은 <b>2</b>입니다. 값이 클수록 상세 수준을 높입니다. 유효한 수준은 <b>0</b>에서 <b>4</b> 사이입니다.</p>

속성	설명
<b>path_selector</b>	다음 I/O 작업에 사용할 경로를 결정하는 데 사용할 기본 알고리즘을 지정합니다. 가능한 값은 다음과 같습니다.
	<b>라운드 로빈 0</b> : 경로 그룹의 모든 경로를 통해 <code>rr_min_io</code> 또는 <code>rr_min_io_rq</code> 로 결정되는 동일한 수의 I/O 요청을 보냅니다.
	<b>queue-length 0</b> : 다음 그룹인 I/O 요청 그룹을 미해결 I/O 요청 수와 함께 경로 아래로 보냅니다.
	<b>Service-time 0</b> : 다음 그룹 I/O 요청을 예상 서비스 시간으로 경로 아래로 보냅니다. 이는 각 경로에서 상대 처리량에 따라 미지정 I/O의 총 크기를 나눠 결정됩니다.
	기본값은 <b>service-time 0</b> 입니다.
<b>path_grouping_policy</b>	지정되지 않은 다중 경로에 적용할 기본 경로 그룹화 정책을 지정합니다. 가능한 값은 다음과 같습니다.
	장애 조치( <b>failover</b> ): 우선순위 그룹당 경로 1개
	<b>Multibus</b> : 하나의 우선 순위 그룹의 모든 유효한 경로입니다.
	<b>group_by_serial</b> : 감지된 일련 번호당 1개의 우선 순위 그룹
	<b>group_by_prio</b> : 경로 우선순위 값당 1개의 우선순위 그룹 우선순위는 <b>prio</b> 특성에 따라 결정됩니다.
	<b>group_by_node_name</b> : 대상 노드 이름당 1개의 우선순위 그룹 <code>/sys/class/fc_transport/target*/node_name</code> 디렉터리에는 대상 노드 이름이 포함됩니다.
	기본값은 장애 조치( <b>failover</b> )입니다.
<b>uid_attrs</b>	WWID로 <b>uevents</b> 병합을 활성화하려면 이 옵션을 설정합니다. 이 작업은 uevent 처리 효율성을 향상시킬 수 있습니다. 또한 WWID(고유 경로 식별자)를 결정하는 데 사용할 udev 속성을 구성하는 대체 방법입니다.
	이 옵션의 값은 <b>type:ATTR</b> 과 같은 공백으로 구분된 레코드 목록입니다. 여기서 <b>type</b> 은 장치 노드 이름의 시작 부분과 일치하며, <b>ATTR</b> 은 일치하는 장치에 사용할 udev 속성의 이름입니다.
	이 옵션을 구성하고 장치의 장치 노드 이름과 일치하는 경우 이 장치의 WWID를 결정하는 기타 구성된 방법을 덮어씁니다.
	이 값을 <b>sd:ID_SERIAL dasd:ID_UID nvme:ID_WWN</b> 로 설정하여 <b>uevent</b> 병합을 활성화할 수 있습니다.

속성	설명
	기본값은 <b>unset</b> 입니다.
<b>Prio</b>	<p>경로 우선 순위 값을 얻기 위해 호출할 기본 함수를 지정합니다. 예를 들어, SPC-3의 ALUA 비트는 악용 가능한 <b>prio</b> 값을 제공합니다. 가능한 값은 다음과 같습니다.</p> <p><b>const:</b> 우선 순위를 모든 경로로 1로 설정합니다.</p> <p><b>EMC:</b> EMC 어레이의 경로 우선 순위를 생성합니다.</p> <p><b>sysfs: sysfs</b> 에서 경로 우선 순위를 생성합니다. 이 우선순위는 선택적 <b>prio_arg</b> 값 <b>exclusive_pref_bit</b> 을 허용합니다. <b>sysfs</b> 값은 <b>sysfs</b> 속성 <b>access_state</b> 및 <b>preferred_path</b> 를 사용합니다.</p> <p><b>ALU A:</b> SCSI-3 ALUA 설정에 따라 경로 우선 순위를 생성합니다. 장치 구성에 <b>prio alua</b> 및 <b>prio_args exclusive_pref_bit</b> 을 지정하는 경우 다중 경로는 <b>exclusive_pref_bit</b> 이 설정된 경로만 포함하는 경로 그룹을 생성하고 해당 경로 그룹을 가장 높은 우선 순위로 할당합니다. 이러한 유형의 사례에 대한 자세한 내용은 <b>multipath.conf(5)</b> 매뉴얼 페이지를 참조하십시오.</p> <p><b>ONTAP :</b> NetApp 배열의 경로 우선 순위를 생성합니다.</p> <p><b>R DAC:</b> LSI/Engenio RDAC 컨트롤러에 대한 경로 우선 순위를 생성합니다.</p> <p><b>hp_sw:</b> active/standby 모드에서 Compaq/HP 컨트롤러에 대한 경로 우선 순위를 생성합니다.</p> <p><b>IRQ S :</b> HitachigradleS 모듈형 스토리지 어레이의 경로 우선 순위를 생성합니다.</p> <p><b>random:</b> 1에서 10 사이의 임의의 우선 순위를 생성합니다.</p> <p><b>weightedpath:</b> 정규식과 지정된 우선 순위를 인수로 기반으로 경로 우선 순위를 생성합니다. <b>prio_args</b> 키워드가 필요합니다.</p> <p><b>path_latency:</b> 대기 시간 알고리즘을 기반으로 경로 우선 순위를 생성합니다. <b>prio_args</b> 키워드가 필요합니다.</p> <p><b>Ana:</b> NVMe ANA 설정을 기반으로 경로 우선 순위를 생성합니다. 이 우선 순위 루틴은 하드웨어에 따라 다릅니다.</p> <p><b>datacore:</b> 일부 DataCore 스토리지 어레이의 경로 우선 순위를 생성합니다. <b>prio_args</b> 키워드가 필요합니다. 이 우선 순위 루틴은 하드웨어에 따라 다릅니다.</p>



속성	설명
	<p><b>I ET:</b> IP 주소를 기반으로 iSCSI 대상의 경로 우선 순위를 생성합니다.  <b>prio_args</b> 키워드가 필요합니다. 이 우선순위 루틴은 iSCSI에서만 사용할 수 있습니다.</p> <p>기본값은 <b>detect_prio</b> 설정에 따라 다릅니다. <b>detect_prio</b> 가 <b>yes</b> 로 설정된 경우 기본 우선순위 알고리즘은 <b>sysfs</b> 입니다. 유일한 예외는 NetAPP E-Series에 대한 것입니다. 여기서 기본값은 <b>alua</b> 입니다. <b>detect_prio</b> 가 <b>no</b> 로 설정되어 있으면 기본 우선순위 알고리즘은 <b>const</b> 입니다.</p>
<b>prio_args</b>	<p><b>prio</b> 함수에 전달할 인수입니다. 이는 다음 우선순위에 대해서만 적용됩니다.</p> <p>가중치: &lt; hctl,devname,serial,wwn&gt; &lt;regex1&gt; &lt;prex1&gt;  &lt;regex2&gt; &lt;prex2&gt; 형식의 값이 필요합니다.</p>
<b>features</b>	
<b>path_checker</b>	

속성	설명
<b>alias_prefix</b>	
<b>failback</b>	
<b>rr_min_io</b>	
<b>rr_min_io_rq</b>	기본값은 <b>1</b> 입니다.
<b>no_path_retry</b>	
<b>user_friendly_names</b>	
<b>queue_without_daemon</b>	
<b>flush_on_last_del</b>	
<b>max_fds</b>	
<b>checker_timeout</b>	
<b>fast_io_fail_tmo</b>	
<b>dev_loss_tmo</b>	
<b>eh_deadline</b>	

속성	설명
<b>detect_prio</b>	
<b>uid_attribute</b>	
<b>force_sync</b>	
<b>strict_timing</b>	
<b>retrigger_tries, retrigger_delay</b>	
<b>missing_uev_wait_timeout</b>	
<b>deferred_remove</b>	
<b>san_path_err_threshold, san_path_err_forget_rate, san_path_err_recovery_time</b>	
<b>marginal_pathgroups</b>	
<b>log_checker_err</b>	
<b>skip_kpartx</b>	
<b>max_sectors_kb</b>	
<b>ghost_delay</b>	
<b>enable_foreign</b>	

속성	설명
<b>recheck_wwid</b>	
<b>remove_retries</b>	
<b>detect_checker</b>	
<b>reservation_key</b>	
<b>all_tg_pt</b>	

추가 리소스

- [multipath.conf\(5\) 매뉴얼 페이지](#)

### 5.3.

표 5.2.

속성	설명
<b>wwid</b>	

- **path\_grouping\_policy**
- **path\_selector**

- **prio**
- **prio\_args**
- **failback**
- **no\_path\_retry**
- **rr\_min\_io**
- **rr\_min\_io\_rq**
- **flush\_on\_last\_del**
- **features**
- **reservation\_key**
- **user\_friendly\_names**
- **deferred\_remove**
- **san\_path\_err\_threshold**
- **san\_path\_err\_forget\_rate**
- **san\_path\_err\_recovery\_time**

- `marginal_path_err_sample_time`
- `marginal_path_err_rate_threshold`
- `marginal_path_err_recheck_gap_time`
- `marginal_path_double_failed_time`
- `delay_watch_checks`
- `delay_wait_checks`
- `skip_kpartx`
- `max_sectors_kb`
- `ghost_delay`

첫 번째 장치에는 WWID가 `3600508b4000156d70001200000b0000` 이고 노란색 이라는 심볼릭 이름이 있습니다.

#### 예 5.1.

```
multipaths {
  multipath {
    wwid          3600508b4000156d70001200000b0000
    alias         yellow
    path_grouping_policy multibus
    path_selector "round-robin 0"
    failback      manual
    no_path_retry 5
  }
  multipath {
```



속성	설명
<b>product_blacklist</b>	
<b>vpd_vendor</b>	
<b>hardware_handler</b>	



중요

- **path\_grouping\_policy**
- **uid\_attribute**
- **getuid\_callout**
- **path\_selector**
- **path\_checker**
- **prio**
- **prio\_args**



- **failback**
- **alias\_prefix**
- **no\_path\_retry**
- **rr\_min\_io**
- **rr\_min\_io\_rq**
- **flush\_on\_last\_del**
- **features**
- **reservation\_key**
- **user\_friendly\_names**
- **deferred\_remove**
- **san\_path\_err\_threshold**
- **san\_path\_err\_forget\_rate**
- **san\_path\_err\_recovery\_time**
- **marginal\_path\_err\_sample\_time**

- **marginal\_path\_err\_rate\_threshold**
- **marginal\_path\_err\_recheck\_gap\_time**
- **marginal\_path\_double\_failed\_time**
- **delay\_watch\_checks**
- **delay\_wait\_checks**
- **skip\_kpartx**
- **max\_sectors\_kb**
- **ghost\_delay**
- **all\_tg\_pt**

#### 추가 리소스

- **multipath.conf(5) 매뉴얼 페이지**
- 

#### 5.5.

- `path_grouping_policy`
- `uid_attribute`
- `getuid_callout`
- `path_selector`
- `path_checker`
- `alias_prefix`
- `features`
- `prio`
- `prio_args`
- `failback`
- `no_path_retry`
- `rr_min_io`
- `rr_min_io_rq`
- `flush_on_last_del`

- **fast\_io\_fail\_tmo**
- **dev\_loss\_tmo**
- **eh\_deadline**
- **user\_friendly\_names**
- **retain\_attached\_hw\_handler**
- **detect\_prio**
- **detect\_checker**
- **deferred\_remove**
- **san\_path\_err\_threshold**
- **san\_path\_err\_forget\_rate**
- **san\_path\_err\_recovery\_time**
- **marginal\_path\_err\_sample\_time**
- **marginal\_path\_err\_rate\_threshold**
- **marginal\_path\_err\_recheck\_gap\_time**

- `marginal_path_double_failed_time`
- `delay_watch_checks`
- `delay_wait_checks`
- `skip_kpartx`
- `max_sectors_kb`
- `ghost_delay`
- `all_tg_pt`

표 5.4.

속성	설명
type	

- `fast_io_fail_tmo`
- `dev_loss_tmo`
- `eh_deadline`

## 추가 리소스

- [multipath.conf\(5\) 매뉴얼 페이지](#)

- 

## 5.6. 장치 시간 초과 DM MULTIPATH 덮어쓰기

**recovery\_tmo sysfs** 옵션은 특정 **iSCSI** 장치의 시간 제한을 제어합니다. 다음 옵션은 **recovery\_tmo** 값을 전역적으로 덮어씁니다.

- **replacement\_timeout** 구성 옵션은 모든 **iSCSI** 장치의 **recovery\_tmo** 값을 전역적으로 덮어 씁니다.

- **DM Multipath**에서 관리하는 모든 **iSCSI** 장치의 경우 **DM Multipath**의 **fast\_io\_fail\_tmo** 옵션은 **recovery\_tmo** 값을 전체적으로 덮어씁니다.

**DM Multipath**의 **fast\_io\_fail\_tmo** 옵션은 파이버 채널 장치의 **fast\_io\_fail\_tmo** 옵션도 덮어 씁니다.

**DM Multipath fast\_io\_fail\_tmo** 옵션이 **replacement\_timeout** 보다 우선합니다. **DM Multipath**는 **multipathd** 서비스가 다시 로드될 때 항상 **recovery\_tmo** 를 재설정하므로 **replacement\_timeout** 을 사용하여 **DM Multipath**에서 관리하는 장치에서 **recovery\_tmo** 를 덮어쓰는 것을 권장하지 않습니다.

## 5.7. 다중 경로 구성 파일 기본값 수정

**/etc/multipath.conf** 구성 파일에는 다음과 같이 **user\_friendly\_names** 매개변수를 **yes** 로 설정하는 **defaults** 섹션이 포함되어 있습니다.

```
defaults {
    user_friendly_names yes
}
```

이는 **user\_friendly\_names** 매개변수의 기본값을 덮어씁니다. **multipath.conf** 파일의 **defaults** 섹션에 설정된 기본값은 **DM Multipath**에서 장치, **multipath** 또는 **overrides** 섹션에 지정된 속성으로 덮어 쓰지 않는 한 사용합니다.

## 절차

1. 구성 기본값 템플릿을 포함하는 `/etc/multipath.conf` 구성 파일을 확인합니다.

```
#defaults {
#   polling_interval    10
#   path_selector       "round-robin 0"
#   path_grouping_policy multibus
#   uid_attribute       ID_SERIAL
#   prio                alua
#   path_checker        readsector0
#   rr_min_io           100
#   max_fds             8192
#   rr_weight           priorities
#   failback            immediate
#   no_path_retry       fail
#   user_friendly_names yes
#}
```

2. 구성 매개변수의 기본값을 덮어씁니다. 이 템플릿의 관련 행을 **defaults** 섹션에 복사하고 주석 처리를 해제할 수 있습니다.

예를 들어, `path_grouping_policy` 매개변수를 기본 장애 조치 값 대신 **multibus** 로 덮어쓰려면 템플릿에서 구성 파일의 초기 **defaults** 섹션으로 적절한 행을 복사하고 다음과 같이 주석을 제거합니다.

```
defaults {
    user_friendly_names yes
    path_grouping_policy multibus
}
```

3. 다음 명령 중 하나를 실행하여 다중 경로 구성 파일을 수정한 후 `/etc/multipath.conf` 파일을 확인합니다.

- 구성 오류를 표시하려면 다음을 실행합니다.

```
# multipath -t > /dev/null
```

- 변경 사항이 추가된 새 구성을 표시하려면 다음을 실행합니다.

```
# multipath -t
```

4. `/etc/multipath.conf` 파일을 다시 로드하고 변경 사항을 적용하려면 **multipathd** 데몬을 재구

성하십시오.

```
# service multipathd reload
```

추가 리소스

- [multipath.conf\(5\) 및 multipathd\(8\) 매뉴얼 페이지](#)

## 5.8. 특정 장치의 다중 경로 설정 수정

`multipath.conf` 구성 파일의 `multipaths` 섹션에서 필수 `WWID` 매개변수에서 참조하는 개별 다중 경로 장치에 고유한 구성을 추가할 수 있습니다.

이러한 기본값은 `multipath.conf` 파일의 `overrides,defaults` 및 `devices` 섹션에 설정된 `DM Multipath` 및 `override` 속성에서 사용합니다. `multipaths` 섹션에는 여러 `multipath` 하위 섹션이 있을 수 있습니다.

절차

1.

특정 다중 경로 장치의 `multipaths` 섹션을 수정합니다. 다음 예제에서는 두 개의 특정 다중 경로 장치에 대한 구성 파일에 지정된 다중 경로 속성을 보여줍니다.

- 첫 번째 장치에는 `WWID`가 `3600508b4000156d70001200000b0000` 이고 노란색 이라는 심볼릭 이름이 있습니다.
- 예제의 두 번째 다중 경로 장치에는 `WWID`가 `1DEC_321816758474` 이고 심볼릭 이름은 빨간색입니다.

이 예제에서는 `rr_weight` 특성이 우선순위로 설정됩니다.

```
multipaths {
  multipath {
    wwid          3600508b4000156d70001200000b0000
    alias         yellow
    path_grouping_policy multibus
    path_selector "round-robin 0"
    failback      manual
    rr_weight     priorities
    no_path_retry 5
  }
  multipath {
```



```

wwid          1DEC_321816758474
alias         red
rr_weight     priorities
}
}

```

2.

다음 명령 중 하나를 실행하여 다중 경로 구성 파일을 수정한 후 `/etc/multipath.conf` 파일을 확인합니다.

- 구성 오류를 표시하려면 다음을 실행합니다.

```
# multipath -t > /dev/null
```

- 변경 사항이 추가된 새 구성을 표시하려면 다음을 실행합니다.

```
# multipath -t
```

3.

`/etc/multipath.conf` 파일을 다시 로드하고 변경 사항을 적용하려면 `multipathd` 데몬을 재구성하십시오.

```
# service multipathd reload
```

추가 리소스

- [multipath.conf\(5\) 매뉴얼 페이지](#)

## 5.9. 프로토콜을 사용하여 특정 장치의 다중 경로 구성 수정

전송 프로토콜을 기반으로 다중 경로 장치 경로를 구성할 수 있습니다. `/etc/multipath.conf` 파일에서 **overrides** 섹션의 **protocol** 하위 섹션을 사용하면 특정 경로에서 다중 경로 구성 설정을 재정의할 수 있습니다. 이를 통해 **파이버 채널(FC)** 또는 **iSCSI(Internet Small Computer Systems Interface)**와 같은 여러 전송 프로토콜을 통해 다중 경로 장치에 액세스할 수 있습니다.

프로토콜 하위 섹션에 설정된 옵션은 덮어쓰기, 장치 및 기본값 섹션의 값을 재정의합니다. 이러한 옵션은 하위 섹션의 **type** 매개 변수와 일치하는 전송 프로토콜을 사용하는 장치에만 적용됩니다.

사전 요구 사항

- 시스템에서 장치 매핑(DM) 다중 경로를 구성했습니다.
- 모든 경로가 동일한 전송 프로토콜을 사용하는 것은 아닌 다중 경로 장치가 있습니다.

절차

1. 다음을 실행하여 특정 경로 프로토콜을 확인합니다.

```
# multipathd show paths format "%d %P"  
dev protocol  
sda scsi:ata  
sdb scsi:fc  
sdc scsi:fc
```

2. 각 다중 경로 유형에 **protocol** 하위 섹션을 추가하여 **/etc/multipath.conf** 파일의 **overrides** 섹션을 편집합니다.

- **scsi:fc** 프로토콜을 사용하는 경로 장치에 대한 설정:

```
overrides {  
    dev_loss_tmo 60  
    fast_io_fail_tmo 8  
    protocol {  
        type "scsi:fc"  
        dev_loss_tmo 70  
        fast_io_fail_tmo 10  
        eh_deadline 360  
    }  
}
```

- **scsi:iscsi** 프로토콜을 사용하는 경로 장치에 대한 설정:

```
overrides {  
    dev_loss_tmo 60  
    fast_io_fail_tmo 8  
    protocol {  
        type "scsi:iscsi"  
        dev_loss_tmo 60  
        fast_io_fail_tmo 120  
    }  
}
```

- 다른 모든 프로토콜을 사용하는 경로 장치의 설정:

```
overrides {
    dev_loss_tmo 60
    fast_io_fail_tmo 8
    protocol {
        type "<type of protocol>"
        dev_loss_tmo 60
        fast_io_fail_tmo 8
    }
}
```

**overrides** 섹션에는 여러 프로토콜 하위 섹션이 포함될 수 있습니다.



중요

**protocol** 하위 섹션에는 **type** 매개 변수가 포함되어야 합니다. 그러면 일치하는 **type** 매개 변수가 있는 모든 경로의 구성이 **protocol** 하위 섹션에 나열된 나머지 매개 변수로 업데이트됩니다.

추가 리소스

- [multipath.conf\(5\) 매뉴얼 페이지](#)

## 5.10. 스토리지 컨트롤러의 다중 경로 설정 수정

**multipath.conf** 구성 파일의 **devices** 섹션에서는 개별 스토리지 장치에 대한 속성을 설정합니다. **DM Multipath**는 해당 장치가 장치가 포함된 경로에 대해 **multipath.conf** 파일의 **multipath.conf** 파일에 지정된 속성으로 덮어쓰거나 재정의하지 않는 한 이러한 속성을 사용합니다. 이러한 속성은 **multipath.conf** 파일의 **defaults** 섹션에 설정된 속성을 재정의합니다.

절차

1. 지원되는 장치를 포함하여 기본 구성 값에 대한 정보를 확인합니다.

```
# multipathd show config
# multipath -t
```

다중 경로를 지원하는 대부분의 장치는 기본적으로 다중 경로 구성에 포함됩니다.

2.

선택 사항: 기본 구성 값을 수정해야 하는 경우 해당 값을 덮어쓰는 장치의 구성 파일에 항목을 포함하여 기본값을 덮어쓸 수 있습니다. `multipathd show config` 명령에서 표시하는 장치의 장치 구성 기본값을 복사하고 변경하려는 값을 덮어쓸 수 있습니다.

3.

`vendor` 및 `product` 매개 변수를 설정하여 기본적으로 구성 파일의 `devices` 섹션에 자동으로 구성되지 않은 장치를 추가합니다. 다음 예와 같이 `/sys/block/ device_name /device/vendor` 및 `/sys/block/device_name/device/model` 파일을 열어 이러한 값을 찾습니다. 여기서 `device_name`은 다중 경로 지정 장치입니다.

```
# cat /sys/block/sda/device/vendor
WINSYS
# cat /sys/block/sda/device/model
SF2372
```

4.

선택 사항: 특정 장치에 따라 추가 매개변수를 지정합니다.

#### 활성/활성 장치

일반적으로 이 경우 추가 매개변수를 설정할 필요가 없습니다. 필요한 경우 `path_grouping_policy` 를 `multibus` 로 설정할 수 있습니다. 설정해야 할 다른 매개변수는 `no_path_retry` 및 `rr_min_io` 입니다.

#### 활성/수동 장치

I/O의 경로를 수동 경로로 자동 전환하면 `Checker` 함수를 작동 중인지 테스트할 경로에 I/O를 보내지 않고, 그렇지 않으면 장치가 계속 장애를 유지합니다. 즉, 테스트 단위 `Ready` 명령을 지원하는 모든 `SCSI` 장치에 대해 `path_checker` 를 `tur` 로 설정해야 합니다.

장치에 경로를 전환하기 위해 특수 명령이 필요한 경우 다중 경로를 위해 이 장치를 구성하려면 하드웨어 처리기 커널 모듈이 필요합니다. 사용 가능한 현재 하드웨어 처리기가 `emc` 입니다. 장치에 충분하지 않은 경우 다중 경로의 장치를 구성하지 못할 수 있습니다.

다음 예는 다중 경로 구성 파일의 장치 항목을 보여줍니다.

```
# }
# device {
# vendor "COMPAQ "
# product "MSA1000 "
# path_grouping_policy multibus
# path_checker tur
# rr_weight priorities
# }
# }
```

5.

다음 명령 중 하나를 실행하여 다중 경로 구성 파일을 수정한 후 `/etc/multipath.conf` 파일을 확인합니다.

•

구성 오류를 표시하려면 다음을 실행합니다.

```
# multipath -t > /dev/null
```

•

변경 사항이 추가된 새 구성을 표시하려면 다음을 실행합니다.

```
# multipath -t
```

6.

`/etc/multipath.conf` 파일을 다시 로드하고 변경 사항을 적용하려면 `multipathd` 데몬을 재구성하십시오.

```
# service multipathd reload
```

추가 리소스

•

[multipath.conf\(5\)](#) 및 [multipathd\(8\)](#) 매뉴얼 페이지

### 5.11. 모든 장치에 대한 다중 경로 값 설정

`multipath.conf` 구성 파일의 `overrides` 섹션을 사용하여 모든 장치에 대한 구성 값을 설정할 수 있습니다. 이 섹션에서는 `multipath.conf` 구성 파일의 `devices` 및 `defaults` 섹션에서 모두 지원하는 모든 속성을 지원합니다. 이 속성은 `vendor,product,revision` 을 제외한 모든 `devices` 섹션 특성에 해당합니다.

DM Multipath는 장치를 포함하는 경로에 대해 `multipath.conf` 파일의 `multipaths` 섹션에 지정된 속성을 덮어쓰지 않는 한 모든 장치에 이러한 속성을 사용합니다. 이러한 속성은 `devices` 에 설정된 속성과 `multipath.conf` 파일의 `defaults` 섹션을 재정의합니다.

절차

1.

장치별 설정을 재정의합니다. 예를 들어 모든 장치가 `no_path_retry` 를 실패 하도록 설정할 수 있습니다. 모든 경로가 실패한 경우 다음 명령을 사용하여 큐를 끄십시오. 이렇게 하면 모든 장치별 설정이 재정의됩니다.

```
overrides {  
    no_path_retry fail  
}
```

2.

다음 명령 중 하나를 실행하여 다중 경로 구성 파일을 수정한 후 **/etc/multipath.conf** 파일을 확인합니다.

- 구성 오류를 표시하려면 다음을 실행합니다.

```
# multipath -t > /dev/null
```

- 변경 사항이 추가된 새 구성을 표시하려면 다음을 실행합니다.

```
# multipath -t
```

3.

**/etc/multipath.conf** 파일을 다시 로드하고 변경 사항을 적용하려면 **multipathd** 데몬을 재구성하십시오.

```
# service multipathd reload
```

추가 리소스

- **multipath.conf(5)** 매뉴얼 페이지

## 6장. 장치가 다중 경로 방지

다중 경로 장치를 구성할 때 선택한 장치를 무시하도록 **DM Multipath**를 구성할 수 있습니다. **DM Multipath**는 이러한 무시된 장치를 다중 경로 장치로 그룹화하지 않습니다.

### 6.1. DM MULTIPATH가 경로에 대한 다중 경로 장치를 생성하는 경우의 조건

**DM Multipath**에는 경로에 대한 다중 경로 장치를 생성할지 또는 경로를 무시할지 여부를 결정하는 기본 규칙 세트가 있습니다. 동작을 구성할 수 있습니다.

**find\_multipaths** 구성 매개변수를 **off**로 설정하면 다중 경로가 명시적으로 비활성화되지 않은 모든 경로에 대해 다중 경로 장치를 생성하려고 합니다. **find\_multipaths** 구성 매개 변수가 **on**의 로 설정된 경우 다음 조건 중 하나가 충족되는 경우에만 다중 경로가 장치를 생성합니다.

- 비활성화되지 않은 **WWID(WWide Identification)**가 동일한 두 개 이상의 경로가 있습니다.
- 다중 경로 명령으로 장치를 지정하여 장치 생성을 수동으로 강제 적용합니다.
- 경로(**WWID**)는 해당 다중 경로 장치가 현재 존재하지 않는 경우에도 이전에 생성된 다중 경로 장치와 동일한 **WWID**입니다. 다중 경로 장치가 생성될 때마다 다중 경로가 장치의 **WWID**를 기억하여 장치의 **WWID**가 자동으로 생성되므로 해당 **WWID**가 있는 경로가 표시되는 즉시 장치를 다시 생성합니다. 이를 통해 다른 장치에서 멀티패스를 비활성화하지 않고도 다중 경로 장치에 올바른 경로를 자동으로 선택할 수 있습니다.

이전에 **find\_multipaths** 매개변수를 사용하지 않고 다중 경로 장치를 생성한 다음 나중에 매개 변수를 설정하면 **/etc/multipath/wwids** 파일에서 다중 경로 장치로 생성되지 않는 모든 장치의 **WWID**를 제거해야 할 수 있습니다. 다음 예제는 샘플 **/etc/multipath/wwids** 파일을 보여줍니다. **WWID**는 슬래시(/)로 묶어야 합니다.

```
# Multipath wwids, Version : 1.0
# NOTE: This file is automatically maintained by multipath and multipathd.
# You should not need to edit this file in normal circumstances.
#
# Valid WWIDs:
/3600d0230000000000e13955cc3757802/
/3600d0230000000000e13955cc3757801/
/3600d0230000000000e13955cc3757800/
/3600d02300069c9ce09d41c31f29d4c00/
/SWINSYS SF2372 0E13955CC3757802/
/3600d0230000000000e13955cc3757803/
```

**on** 및 **off** 뿐만 아니라 **find\_multipaths** 를 다음 값으로 설정할 수도 있습니다.

### strict

다중 경로는 이전에 다중 경로가 아닌 경로를 허용하지 않으므로 **/etc/multipath/wwids** 파일에 없습니다.

### smart

다중 경로에서는 항상 **udev** 에서 비활성화되지 않은 장치를 표시 즉시 허용합니다. **multipathd** 가 **find\_multipaths\_timeout** 매개변수로 설정된 시간 내에 장치를 생성하지 않으면 장치에서 해당 클레이를 해제합니다.

**find\_multipaths** 의 기본값은 해제 되어 있습니다. 그러나 **mpathconf** 에서 생성한 기본 **multipath.conf** 파일은 **find\_multipaths** 값을 의 로 설정합니다.

**find\_multipaths** 매개변수가 **on;**의 로 설정된 경우 다중 경로인 여러 경로가 있는 장치에서만 다중 경로를 비활성화합니다. 이로 인해 일반적으로 장치에서 멀티패스를 비활성화할 필요가 없습니다.

이전에 생성된 다중 경로 장치를 추가하여 블랙리스트에 추가된 경우 **-w** 옵션을 사용하여 **/etc/multipath/wwids** 파일에서 해당 장치의 **WWID**를 제거하면 다른 프로그램의 문제가 발생하지 않을 수 있습니다. 예를 들어, **/etc /multipath/ wwids** 파일에서 **WWID 3600d0230000000000e13954ed5f89300 ed5f89300** 장치를 제거하려면 다음 방법 중 하나를 사용할 수 있습니다.

- 장치 이름을 사용하여 다중 경로 장치 제거.

```
# multipath -w /dev/sdb
wwid '3600d0230000000000e13954ed5f89300' removed
```

- 장치의 **WWID**를 사용하여 다중 경로 장치 제거.

```
# multipath -w 3600d0230000000000e13954ed5f89300
wwid '3600d0230000000000e13954ed5f89300' removed
```

또한 **-W** 옵션을 사용하여 **/etc/multipath/wwids** 파일을 업데이트할 수도 있습니다. 이렇게 하면 현재 다중 경로 장치의 **WWID**만 포함하도록 **/etc/multipath/wwids** 파일이 재설정됩니다. 파일을 재설정하려면 다음을 실행합니다.



```
# multipath -W
successfully reset wwid
```

추가 리소스

- **multipath.conf(5)** 매뉴얼 페이지

## 6.2. 특정 장치에서 멀티패스를 비활성화하는 기준

다음 기준 중 하나를 사용하여 장치에서 다중 경로를 비활성화할 수 있습니다.

- **WWID**
- 장치 이름
- 장치 유형
- 속성
- 프로토콜

모든 장치에 대해 **DM Multipath**는 다음 순서로 이러한 기준을 평가합니다.

1. 속성
2. **devnode**
3. **device**
4. 프로토콜

5. **wwid**

장치가 언급된 기준에 따라 비활성화되는 경우 **DM Multipath**는 다중 경로 의 처리에서 해당 장치를 제외하고 이후 기준을 평가하지 않습니다. 각 조건에 대해 장치가 둘 다 일치하면 비활성화된 장치 목록보다 예외 목록이 우선합니다.



참고

구성 파일의 초기 블랙리스트 섹션을 주석 처리한 후에도 다양한 장치 유형이 비활성화되어 있습니다.

추가 리소스

- [다중 경로가 비활성화된 장치에 대한 예외 추가](#)

6.3. WWID를 통한 다중 경로 비활성화

**WWID(WWide Identification)**를 통해 개별 장치에서 다중 경로를 비활성화할 수 있습니다.

절차

1. 장치의 **WWID**를 찾습니다.

```
# multipathd show paths raw format "%d %w" | grep sdb
sdb 3600508b4001080520001e00011700000
```

2. **wwid** 항목을 사용하여 **/etc/multipath.conf** 구성 파일에서 장치를 비활성화합니다.

다음 예제에서는 **WWID**가 **3600508b4001080520001e00011700000** 인 장치를 비활성화하는 **DM Multipath** 구성 파일의 행을 보여줍니다.

```
blacklist {
    wwid 3600508b4001080520001e00011700000
}
```

3. 다음 명령 중 하나를 실행하여 다중 경로 구성 파일을 수정한 후 **/etc/multipath.conf** 파일을 확인합니다.

- 구성 오류를 표시하려면 다음을 실행합니다.

```
# multipath -t > /dev/null
```

- 변경 사항이 추가된 새 구성을 표시하려면 다음을 실행합니다.

```
# multipath -t
```

4.

`/etc/multipath.conf` 파일을 다시 로드하고 변경 사항을 적용하려면 `multipathd` 데몬을 재구성하십시오.

```
# service multipathd reload
```

#### 6.4. 장치 이름으로 다중 경로 비활성화

**DM Multipath**가 다중 경로 장치로 그룹화하지 않도록 장치 유형별로 다중 경로를 비활성화할 수 있습니다.

##### 절차

1. 장치 정보를 표시합니다.

```
# udevadm info --query=all -n /dev/mapper/sd*
```

2.

`devnode` 항목을 사용하여 `/etc/multipath.conf` 구성 파일에서 장치를 비활성화합니다.

다음 예제에서는 모든 `sd*` 장치를 비활성화하기 때문에 모든 **SCSI** 장치를 비활성화하는 **DM Multipath** 구성 파일의 행을 보여줍니다.

```
blacklist {
    devnode "^sd[a-z]"
}
```

`devnode` 항목을 사용하여 특정 유형의 모든 장치가 아닌 개별 장치를 비활성화할 수 있습니다. 그러나 `udev` 규칙에 의해 정적으로 매핑되지 않는 한 특정 장치의 이름이 재부팅 시 동일한 것이라는 보장이 없기 때문에 이 방법은 사용하지 않는 것이 좋습니다. 예를 들어 장치 이름이 재부팅 시 `/dev/sda` 에서 `/dev/sdb` 로 변경될 수 있습니다.

기본적으로 **DM Multipath**는 다음 **devnode** 항목을 사용하여 **SCSI**, **NVMe** 또는 **DASD**가 아닌 모든 장치를 비활성화합니다.

```
blacklist {
    devnode "!^(sd[a-z]|dasd[a-z]|nvme[0-9])"
}
```

이 항목을 비활성화하는 장치는 일반적으로 **DM Multipath**를 지원하지 않습니다.

3.

다음 명령 중 하나를 실행하여 다중 경로 구성 파일을 수정한 후 **/etc/multipath.conf** 파일을 확인합니다.

- 구성 오류를 표시하려면 다음을 실행합니다.

```
# multipath -t > /dev/null
```

- 변경 사항이 추가된 새 구성을 표시하려면 다음을 실행합니다.

```
# multipath -t
```

4.

**/etc/multipath.conf** 파일을 다시 로드하고 변경 사항을 적용하려면 **multipathd** 데몬을 재구성하십시오.

```
# service multipathd reload
```

#### 추가 리소스

- [다중 경로가 비활성화된 장치에 대한 예외 추가](#)

## 6.5. 장치 유형별 다중 경로 비활성화

**device** 섹션을 사용하여 장치에서 멀티패스를 비활성화할 수 있습니다.

#### 절차

1. 표시 장치 유형:

```
# multipathd show paths raw format "%d %s" | grep sdb
sdb HP,HSV210
```

2.

**device** 섹션을 사용하여 **/etc/multipath.conf** 구성 파일에서 장치를 비활성화합니다.

다음 예제에서는 모든 **IBM DS4200** 및 **HP** 장치에서 멀티패스를 비활성화합니다.

```
blacklist {
  device {
    vendor "IBM"
    product "3S42"    #DS4200 Product 10
  }
  device {
    vendor "HP"
    product ".*"
  }
}
```

3.

다음 명령 중 하나를 실행하여 다중 경로 구성 파일을 수정한 후 **/etc/multipath.conf** 파일을 확인합니다.

- 구성 오류를 표시하려면 다음을 실행합니다.

```
# multipath -t > /dev/null
```

- 변경 사항이 추가된 새 구성을 표시하려면 다음을 실행합니다.

```
# multipath -t
```

4.

**/etc/multipath.conf** 파일을 다시 로드하고 변경 사항을 적용하려면 **multipathd** 데몬을 재구성하십시오.

```
# service multipathd reload
```

## 6.6. UDEV 속성의 다중 경로 비활성화

**udev property** 매개변수를 통해 장치에서 멀티패스를 비활성화할 수 있습니다.

절차

1. 장치의 **udev** 변수를 표시합니다.

```
# udevadm info --query=all -n /dev/sdb
```

2. **property** 매개 변수를 사용하여 **/etc/multipath.conf** 구성 파일에서 장치를 비활성화합니다. 이 매개변수는 장치의 **udev** 환경 변수 이름과 일치하는 정규식 문자열입니다.

다음 예제에서는 **udev** 속성 **ID\_ATA** 를 사용하여 모든 장치에서 다중 경로를 비활성화합니다.

```
blacklist {
    property "ID_ATA"
}
```

3. 다음 명령 중 하나를 실행하여 다중 경로 구성 파일을 수정한 후 **/etc/multipath.conf** 파일을 확인합니다.

- 구성 오류를 표시하려면 다음을 실행합니다.

```
# multipath -t > /dev/null
```

- 변경 사항이 추가된 새 구성을 표시하려면 다음을 실행합니다.

```
# multipath -t
```

4. **/etc/multipath.conf** 파일을 다시 로드하고 변경 사항을 적용하려면 **multipathd** 데몬을 재구성하십시오.

```
# service multipathd reload
```

## 6.7. 장치 프로토콜별 다중 경로 비활성화

장치 프로토콜을 사용하여 장치에서 다중 경로를 비활성화할 수 있습니다.

절차

1. 선택 사항: 경로가 사용 중인 프로토콜을 확인합니다.

```
# multipathd show paths raw format "%d %P" | grep sdb
sdb scsi:fc
```

2. **protocol** 매개변수를 사용하여 **/etc/multipath.conf** 구성 파일에서 장치를 비활성화합니다.

**protocol** 매개변수는 정규식을 사용하고 프로토콜 문자열이 일치하는 모든 장치를 블랙리스트로 지정합니다. 예를 들어 모든 **nvme** 장치에서 다중 경로를 비활성화하려면 다음을 사용합니다.

```
blacklist {
    protocol "nvme"
}
```

**DM Multipath**는 다음 프로토콜 문자열을 인식합니다.

- **scsi:fc**
- **scsi:spi**
- **scsi:ssa**
- **scsi:sbp**
- **scsi:srp**
- **scsi:iscsi**
- **scsi:sas**
- **scsi:adt**

- **scsi:ata**
- **scsi:unspec**
- **ccw**
- **cciss**
- **nvme:pcie**
- **nvme:rdma**
- **nvme:fc**
- **nvme:tcp**
- **nvme:loop**
- **nvme:apple-nvme**
- **nvme:unspec**
- **undef**

3.

다음 명령 중 하나를 실행하여 다중 경로 구성 파일을 수정한 후 **/etc/multipath.conf** 파일을 확인합니다.

- 구성 오류를 표시하려면 다음을 실행합니다.



```
# multipath -t > /dev/null
```

- 변경 사항이 추가된 새 구성을 표시하려면 다음을 실행합니다.

```
# multipath -t
```

4.

`/etc/multipath.conf` 파일을 다시 로드하고 변경 사항을 적용하려면 `multipathd` 데몬을 재구성하십시오.

```
# service multipathd reload
```

## 6.8. 다중 경로가 비활성화된 장치에 대한 예외 추가

현재 다중 경로가 비활성화된 장치에 예외를 추가하여 다중 경로를 활성화할 수 있습니다.

### 사전 요구 사항

- 특정 장치에서 멀티패스가 비활성화됩니다.

### 절차

1.

`/etc/multipath.conf` 구성 파일의 `blacklist_exceptions` 섹션을 사용하여 장치에서 다중 경로를 활성화합니다.

구성 파일의 `blacklist_exceptions` 섹션에 장치를 지정할 때 블랙리스트 섹션에 지정된 것과 동일한 기준을 사용하여 예외를 지정해야 합니다. 예를 들어, `WWID` 예외는 비활성화 된 장치가 해당 `WWID`와 연결된 경우에도 `devnode` 항목에 의해 비활성화 된 장치에 적용되지 않습니다. 마찬가지로 `devnode` 예외는 `devnode` 항목에만 적용되며 장치 예외는 장치 항목에만 적용됩니다.

#### 예 6.1. WWID의 예외

많은 장치가 있으며 `WWID`가 `3600d0230000000000e13955cc3757803` 을 사용하여 하나의 다중 경로만 배치하려는 경우 원하는 모든 장치를 개별적으로 비활성화하는 대신 모든 장치를 비활성화한 다음 `/etc/multipath.conf` 파일에 다음 행만 추가하여 활성화할 수 있습니다.

```
blacklist {
    wwid ".*"
}
```

```
blacklist_exceptions {
    wwid "3600d0230000000000e13955cc3757803"
}
```

또는 느낌표(!)를 사용하여 블랙리스트 항목을 반전하여 지정된 WWID를 제외한 모든 장치를 비활성화할 수 있습니다.

```
blacklist {
    wwid "!3600d0230000000000e13955cc3757803"
}
```

## 예 6.2. udev 속성의 예외

**property** 매개 변수는 다른 **blacklist\_exception** 매개 변수와 다르게 작동합니다. **property** 매개 변수 값은 **udev** 데이터베이스의 변수 이름과 일치해야 합니다. 그렇지 않으면 장치가 비활성화되어 있습니다. 이 매개 변수를 사용하면 **USB** 고정 및 로컬 하드 드라이브와 같은 특정 **SCSI** 장치에서 다중 경로를 비활성화할 수 있습니다.

다중 경로 지정될 수 있는 **SCSI** 장치에서만 멀티패스를 활성화하려면 다음 예와 같이 이 매개 변수를 (**SCSI\_IDENT\_ID\_WWN**) 로 설정합니다.

```
blacklist_exceptions {
    property "(SCSI_IDENT_ID_WWN)"
}
```

2.

다음 명령 중 하나를 실행하여 다중 경로 구성 파일을 수정한 후 **/etc/multipath.conf** 파일을 확인합니다.

- 구성 오류를 표시하려면 다음을 실행합니다.

```
# multipath -t > /dev/null
```

- 변경 사항이 추가된 새 구성을 표시하려면 다음을 실행합니다.

```
# multipath -t
```

3.

**/etc/multipath.conf** 파일을 다시 로드하고 변경 사항을 적용하려면 **multipathd** 데몬을 재구성하십시오.

█ # service multipathd reload

## 7장. 다중 경로 볼륨 관리

다음은 다중 경로 볼륨을 관리하는 데 사용할 수 있는 **DM Multipath**에서 제공하는 몇 가지 명령입니다.

- **multipath**
- **dmsetup**
- **multipathd**

### 7.1. 온라인 다중 경로 장치 크기 조정

온라인 다중 경로 장치의 크기를 조정해야 하는 경우 다음 절차를 사용하십시오.

#### 절차

1. 물리적 장치의 크기를 조정합니다.
2. 다음 명령을 실행하여 논리 장치 번호(LUN)의 경로를 찾습니다.

```
# multipath -l
```

3. 경로 크기 조정. SCSI 장치의 경우 장치의 다시 스캔 파일에 1을 저장하면 다음 명령과 같이 SCSI 드라이버가 다시 스캔됩니다.

```
# echo 1 > /sys/block/path_device/device/rescan
```

각 경로 장치에 대해 이 명령을 실행해야 합니다. 예를 들어 경로 장치가 **sda,sdb,sde** 및 **sdf** 인 경우 다음 명령을 실행합니다.

```
# echo 1 > /sys/block/sda/device/rescan
# echo 1 > /sys/block/sdb/device/rescan
# echo 1 > /sys/block/sde/device/rescan
# echo 1 > /sys/block/sdf/device/rescan
```

4. 다중 경로 장치 크기 조정:

```
# multipathd resize map multipath_device
```

5. 파일 시스템의 크기 조정 (LVM 또는 DOS 파티션이 사용되지 않는 것으로 가정).

```
# resize2fs /dev/mapper/mpatha
```

## 7.2. 루트 파일 시스템을 단일 경로 장치에서 다중 경로 장치로 이동

시스템을 단일 경로 장치에 설치하고 나중에 루트 파일 시스템에 다른 경로를 추가하는 경우 루트 파일 시스템을 다중 경로 장치로 이동해야 합니다. 단일 경로에서 다중 경로 장치로 이동하려면 다음 절차를 참조하십시오.

### 사전 요구 사항

- **device-mapper-multipath** 패키지가 설치되어 있습니다.

### 절차

1. **/etc/multipath.conf** 구성 파일을 생성하고 **multipath** 모듈을 로드하고, **multipathd systemd** 서비스를 활성화합니다.

```
# dnf install device-mapper-multipath
```

2. 다음 명령을 실행하여 **/etc/multipath.conf** 구성 파일을 생성하고, **multipath** 모듈을 로드하고, **multipathd**의 **chkconfig**를 **on:**으로 설정합니다.

```
# mpathconf --enable
```

3. **find\_multipaths** 구성 매개변수가 **yes**로 설정되지 않은 경우 멀티패스에서 장치 방지에 설정된 대로 **/etc/multipath.conf** 파일의 **blacklist** 및 **blacklist\_exceptions** 섹션을 편집합니다.

4. 루트 장치가 검색되면 다중 경로 장치를 빌드하려면 루트 장치 상단에 다중 경로 장치를 빌드하려면 다음 명령을 입력합니다. 또한 이 명령은 **path**만 있어도 **find\_multipaths**에서 장치를 허용합니다.

```
# multipath -a root_devname
```

예를 들어 루트 장치가 **/dev/sdb** 인 경우 다음 명령을 입력합니다.

```
# multipath -a /dev/sdb
wwid '3600d02300069c9ce09d41c4ac9c53200' added
```

5.

다중 경로 명령을 실행하여 구성 파일이 올바르게 설정되었는지 확인하고 다음 형식의 한 줄에 대한 출력을 검색합니다. 이는 명령이 다중 경로 장치를 생성하지 못했음을 나타냅니다.

```
date wwid: ignoring map
```

예를 들어 장치의 WWID가 **3600d02300069c9ce09d41c4ac9c53200** 이면 다음과 같은 출력이 표시됩니다.

```
# multipath
Oct 21 09:37:19 | 3600d02300069c9ce09d41c4ac9c53200: ignoring map
```

6.

다중 경로를 사용하여 **initramfs** 파일 시스템을 다시 빌드합니다.

```
# dracut --force -H --add multipath
```

7.

머신을 종료합니다.

8.

시스템을 부팅합니다.

9.

다른 경로를 시스템에서 볼 수 있도록 합니다.

#### 검증 단계

- 

다음 명령을 실행하여 다중 경로 장치가 생성되었는지 확인합니다.

```
# multipath -l | grep 3600d02300069c9ce09d41c4ac9c53200
mpatha (3600d02300069c9ce09d41c4ac9c53200) dm-0 3PARdata,VV
```

### 7.3. 스왑 파일 시스템을 단일 경로 장치에서 다중 경로 장치로 이동

기본적으로 스왑 장치는 논리 볼륨으로 설정됩니다. 이 경우 논리 볼륨 그룹을 구성하는 물리 볼륨에 다중 경로를 설정하는 한 다중 경로 장치로 구성하는 특수 프로시저가 필요하지 않습니다. 그러나 스왑 장치가 LVM 볼륨이 아니며 장치 이름으로 마운트된 경우 `/etc/fstab` 파일을 편집하여 적절한 다중 경로 장치 이름으로 전환해야 할 수 있습니다.

## 절차

1. 장치의 WWID를 `/etc/multipath/wwids` 파일에 추가합니다.

```
# multipath -a swap_devname
```

예를 들어 루트 장치가 `/dev/sdb` 인 경우 다음 명령을 입력합니다.

```
# multipath -a /dev/sdb
wwid '3600d02300069c9ce09d41c4ac9c53200' added
```

2. 다중 경로 명령을 실행하여 구성 파일이 올바르게 설정되었는지 확인하고 다음 형식의 행을 검색합니다.

```
date wwid: ignoring map
```

이는 명령이 다중 경로 장치를 생성하지 못했음을 나타냅니다.

예를 들어 장치의 WWID가 `3600d02300069c9ce09d41c4ac9c53200`인 경우 출력에 다음과 같은 행이 표시됩니다.

```
# multipath
Oct 21 09:37:19 | 3600d02300069c9ce09d41c4ac9c53200: ignoring map
```

3. `/etc/multipath.conf` 파일에서 스왑 장치의 별칭을 설정합니다.

```
multipaths {
  multipath {
    wwid WWID_of_swap_device
    alias swapdev
  }
}
```

4.

**/etc/fstab** 파일을 편집하고 루트 장치의 이전 장치 경로를 다중 경로 장치로 교체합니다.

예를 들어 **/etc/fstab** 파일에 다음 항목이 있는 경우 다음을 수행합니다.

```
/dev/sdb2 swap          swap defaults    0 0
```

항목을 다음으로 변경합니다.

```
/dev/mapper/swapdev swap      swap defaults    0 0
```

5.

다중 경로를 사용하여 **initramfs** 파일 시스템을 다시 빌드합니다.

```
# dracut --force -H --add multipath
```

6.

머신을 종료합니다.

7.

시스템을 부팅합니다.

8.

다른 경로를 시스템에서 볼 수 있도록 합니다.

#### 검증 단계

•

스왑 장치가 다중 경로 장치에 있는지 확인합니다.

```
# swapon -s
```

예를 들어 다음과 같습니다.

```
# swapon -s
```

<b>Filename</b>	<b>Type</b>	<b>Size Used</b>	<b>Priority</b>
<b>/dev/dm-3</b>	<b>partition</b>	<b>4169724 0</b>	<b>-2</b>

파일 이름은 다중 경로 스왑 장치와 일치해야 합니다.



```
# readlink -f /dev/mapper/swapdev
/dev/dm-3
```

#### 7.4. DMSETUP 명령을 사용하여 장치 매핑 항목 확인

**dmsetup** 명령을 사용하여 다중 경로 장치와 일치하는 장치 매핑 항목을 확인할 수 있습니다.

##### 절차

- 모든 장치 매핑 장치와 해당 주요 번호와 마이너 번호를 표시합니다. 마이너 번호는 **dm** 장치의 이름을 결정합니다. 예를 들어 마이너 수 **3**은 다중 경로 장치 **/dev/dm-3**에 해당합니다.

```
# dmsetup ls
mpathd (253:4)
mpathp1 (253:12)
mpathfp1 (253:11)
mpathb (253:3)
mpathgp1 (253:14)
mpathhp1 (253:13)
mpatha (253:2)
mpathh (253:9)
mpathg (253:8)
VolGroup00-LogVol01 (253:1)
mpathf (253:7)
VolGroup00-LogVol00 (253:0)
mpathe (253:6)
mpathbp1 (253:10)
mpathd (253:5)
```

#### 7.5. 다중 경로 데몬 관리

**multipathd** 명령을 사용하여 **multipathd** 데몬을 관리할 수 있습니다.

##### 절차

- **multipathd show maps** 명령 출력의 기본 형식을 확인합니다.

```
# multipathd show maps
name sysfs uuid
mpathc dm-0 360a98000324669436c2b45666c567942
```

- 일부 **multipathd** 명령에는 **format** 옵션과 와일드카드가 포함됩니다. 다음 명령을 사용하여 사용 가능한 와일드카드 목록을 표시합니다.

```
# multipathd show wildcards
multipath format wildcards:
%n name
%w uuid
%d sysfs
...
```

- **multipathd** 가 모니터링하는 다중 경로 장치를 표시합니다. 와일드카드를 사용하여 표시된 필드를 지정합니다.

```
# multipathd show maps format "%n %w %d %s"
name uuid sysfs vend/prod/rev
mpathc 360a98000324669436c2b45666c567942 dm-0 NETAPP,LUN
```

- **multipathd** 가 모니터링하는 경로를 표시합니다. 와일드카드를 사용하여 표시된 필드를 지정합니다.

```
# multipathd show paths format "%n %w %d %s"
target WWNN uuid dev vend/prod/rev
0x50001fe1500d2250 3600508b4001080520001e00011700000 sdb HP,HSV210
```

- 데이터를 원시 형식으로 표시합니다.

```
# multipathd show maps raw format "%n %w %d %s"
mpathc 360a98000324669436c2b45666c567942 dm-0 NETAPP,LUN
```

원시 형식에서 헤더가 인쇄되지 않으며 필드는 헤더와 열을 정렬하지 않습니다. 이 출력은 스크립팅에 더 쉽게 사용할 수 있습니다.

## 추가 리소스

- **multipathd(8)** 매뉴얼 페이지

## 8장. 스토리지 장치 제거

실행 중인 시스템에서 스토리지 장치를 안전하게 제거하여 시스템 메모리 과부하 및 데이터 손실을 방지할 수 있습니다.

### 사전 요구 사항

- 스토리지 장치를 제거하기 전에 **I/O 플러시** 중에 시스템 메모리가 증가하여 사용 가능한 시스템 메모리가 충분히 있는지 확인해야 합니다. 다음 명령을 사용하여 시스템의 현재 메모리 부하 및 사용 가능한 메모리를 확인합니다.

```
# vmstat 1 100
# free
```

- **Red Hat**은 다음과 같은 경우 시스템에서 스토리지 장치를 제거하는 것을 권장하지 않습니다.
  - 사용 가능한 메모리는 **100개당 10개 이상의 샘플**에서 총 메모리의 **5%** 미만입니다.
  - 스와핑이 활성화 상태입니다(**vmstat** 명령 출력의 **0**이 아닌 **si** 및 **so** 열).

### 8.1. 안전한 스토리지 장치 제거

실행 중인 시스템에서 안전하게 스토리지 장치를 제거하려면 **top-to-bottom** 접근 방식이 필요합니다. 일반적으로 애플리케이션 또는 파일 시스템인 최상위 계층에서 시작하여 물리적 장치인 맨 아래 계층으로 시작합니다.

스토리지 장치를 여러 가지 방법으로 사용할 수 있으며 물리적 장치에 따라 다양한 가상 구성을 사용할 수 있습니다. 예를 들어 장치의 여러 인스턴스를 다중 경로 장치로 그룹화하거나 **RAID**의 일부로 설정하거나 **LVM** 그룹의 일부로 만들 수 있습니다. 또한 파일 시스템을 통해 장치에 액세스하거나 **"raw"** 장치와 같이 직접 액세스할 수 있습니다.

**top-to-bottom** 접근 방식을 사용하는 동안 다음을 확인해야 합니다.

- 삭제하려는 장치를 사용하지 않음

- 장치에 대한 모든 보류 중인 I/O가 플러시됩니다.
- 운영 체제가 스토리지 장치를 참조하지 않음

## 8.2. 블록 장치 및 관련 메타데이터 제거

실행 중인 시스템에서 블록 장치를 안전하게 제거하려면 시스템 메모리 과부하 및 데이터 손실을 방지하려면 먼저 메타데이터에서 메타데이터를 제거해야 합니다. 파일 시스템부터 시작하여 스택의 각 계층을 처리하고 디스크를 진행합니다. 이러한 작업으로 인해 시스템이 일관되지 않은 상태가 되지 않습니다.

제거할 장치 유형에 따라 다를 수 있는 특정 명령을 사용합니다.

- `lvremove`, `remove` 및 `pvremove` 는 LVM에 따라 다릅니다.
- 소프트웨어 RAID의 경우 `mdadm` 을 실행하여 배열을 제거합니다. 자세한 내용은 [RAID 관리](#) 를 참조하십시오.
- LUKS를 사용하여 암호화된 블록 장치의 경우 특정 추가 단계가 있습니다. LUKS를 사용하여 암호화된 블록 장치에는 다음 절차가 작동하지 않습니다. 자세한 내용은 [LUKS를 사용하여 블록 장치 암호화](#) 를 참조하십시오.



### 주의

여기에 설명된 절차에 따라 I/O 시간 초과, 장치가 예기치 않게 제거되거나 데이터 손실로 인한 지연을 초래하지 않고 SCSI 버스를 다시 스캔하거나 운영 체제의 상태를 변경하는 다른 작업을 수행할 수 있습니다.

### 사전 요구 사항

- 파일 시스템, 논리 볼륨 및 볼륨 그룹이 포함된 기존 블록 장치 스택이 있습니다.

- 다른 애플리케이션이나 서비스가 제거하려는 장치를 사용하지 않는 것을 확인했습니다.
- 제거하려는 장치에서 데이터를 백업했습니다.
- 선택 사항: 다중 경로 장치를 제거하고 해당 경로 장치에 액세스할 수 없는 경우 다음 명령을 실행하여 다중 경로 장치의 대기열을 비활성화합니다.

```
# multipathd disablequeueing map multipath-device
```

이를 통해 장치의 I/O가 실패하여 장치를 사용하는 애플리케이션을 종료할 수 있습니다.



#### 참고

한 번에 메타데이터 하나씩 레이어로 장치를 제거하면 오래된 서명이 디스크에 남아 있지 않습니다.

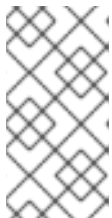
#### 절차

1. 파일 시스템을 마운트 해제합니다.

```
# umount /mnt/mount-point
```

2. 파일 시스템을 제거합니다.

```
# wipefs -a /dev/vg0/myvol
```



#### 참고

파일 시스템과 마운트 지점 간의 지속적인 연결을 위해 **/etc/fstab** 파일에 항목을 추가한 경우 이 시점에서 **/etc/fstab** 도 편집하여 해당 항목을 제거해야 합니다.

제거하려는 장치 유형에 따라 다음 단계를 계속합니다.

3. 파일 시스템이 포함된 LV(Logical Volume)를 제거합니다.

```
# lvremove vg0/myvol
```

4.

볼륨 그룹(VG)에 남아 있는 다른 논리 볼륨이 없는 경우 장치가 포함된 VG를 안전하게 제거할 수 있습니다.

```
# vgrename vg0
```

5.

PV 장치에서 PV(물리 볼륨) 메타데이터를 제거합니다.

```
# pvremove /dev/sdc1
```

```
# wipefs -a /dev/sdc1
```

6.

PV가 포함된 파티션을 제거합니다.

```
# parted /dev/sdc rm 1
```



참고

장치를 완전히 초기화하려는 경우에만 다음 단계를 수행하십시오.

7.

파티션 테이블을 제거합니다.

```
# wipefs -a /dev/sdc
```



참고

장치를 물리적으로 제거하려는 경우에만 다음 단계를 수행하십시오.

•

다중 경로 장치를 제거하는 경우 다음 명령을 실행합니다.

a.

장치에 대한 모든 경로를 표시합니다.

```
# multipath -l
```

이 명령의 출력은 이후 단계에서 필요합니다.

- i. I/O를 플러시하고 다중 경로 장치를 제거합니다.

```
# multipath -f multipath-device
```

- 장치가 다중 경로 장치로 구성되지 않았거나 장치가 다중 경로 장치로 구성되어 있고 이전에 I/O를 개별 경로에 전달한 경우 사용되는 모든 장치 경로에 뛰어난 I/O를 플러시합니다.

```
# blockdev --flushbufs device
```

이는 **NoExecute** 또는 **vgreduce** 명령이 I/O를 플러시하지 않는 경우 직접 액세스하는 장치에 중요합니다.

- SCSI 장치를 제거하는 경우 다음 명령을 실행합니다.

- a. `/dev/sd`, `/dev/disk/by-path`, 또는 시스템의 애플리케이션, 스크립트 또는 유틸리티에서 장치의 경로 기반 이름에 대한 참조를 제거합니다. 이렇게 하면 나중에 추가된 다른 장치가 현재 장치에 대해 잘못되지 않습니다.

- b. SCSI 하위 시스템에서 장치로의 각 경로를 제거합니다.

```
# echo 1 > /sys/block/device-name/device/delete
```

여기에서 **장치 이름**이 이전에 다중 경로 장치로 사용된 경우 **multipath -l** 명령의 출력에서 검색됩니다.

8. 실행 중인 시스템에서 물리적 장치를 제거합니다. 다른 장치의 I/O는 이 장치를 제거할 때 중지되지 않습니다.

## 검증

- 제거하려는 장치가 **lsblk** 명령의 출력에 표시되지 않는지 확인합니다. 다음은 출력 예제입니다.

```
# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda   8:0  0  5G  0 disk
sr0   11:0  1 1024M 0 rom
vda   252:0  0  10G  0 disk
|-vda1 252:1  0   1M  0 part
|-vda2 252:2  0 100M  0 part /boot/efi
`-vda3 252:3  0  9.9G  0 part /
```

#### 추가 리소스

- **multipath(8), pvremove(8), ECDHE remove(8), lvremove(8), delete fs(8), parted(8), blockdev(8) 및 umount(8) 매뉴얼 페이지.**



## 9장. DM MULTIPATH 문제 해결

다중 경로 구성을 구현하는 데 문제가 있는 경우 확인할 수 있는 다양한 문제가 있습니다. 다음 문제로 인해 다중 경로 구성이 느려지거나 작동하지 않을 수 있습니다.

다중 경로 데몬이 실행되지 않음

다중 경로 구성을 구현하는 데 문제가 있는 경우 **DM Multipath** 구성에 설명된 대로 **multipathd** 데몬이 실행 중인지 확인합니다. 다중 경로 장치를 사용하려면 **multipathd** 데몬이 실행 중이어야 합니다.

**queue\_if\_no\_path** 기능 문제

"1 queue\_if\_no\_path" 기능으로 다중 경로 장치가 구성된 경우 하나 이상의 경로가 복원될 때까지 I/O가 중단되는 프로세스를 실행합니다.

### 9.1. QUEUE\_IF\_NO\_PATH 기능 문제 해결

"1 queue\_if\_no\_path" 기능으로 다중 경로 장치가 구성된 경우 하나 이상의 경로가 복원될 때까지 I/O가 중단되는 프로세스를 실행합니다. 이를 방지하려면 `/etc/multipath.conf` 파일에서 `no_path_retry N` 매개변수를 설정합니다. 여기서 *N*은 시스템이 경로를 다시 시도해야 하는 횟수입니다.

설명된 문제 없이 "1 queue\_if\_no\_path" 옵션을 사용하려면 모든 경로를 사용할 수 없는 특정 LUN의 런타임에 큐링 정책을 비활성화할 수 있습니다.

절차

1. 대기열을 비활성화합니다.

- 특정 장치의 경우:

```
# multipathd disablequeueing map device
```

- 모든 장치의 경우:

```
# multipathd disablequeueing maps
```

대기열을 비활성화한 후에는 다중 경로를 다시 시작하거나 다시 로드할 때까지 비활성화된 상태로 유지됩니다.

2. 큐링을 이전 값으로 재설정합니다.

- 특정 장치의 경우:

```
# multipathd restorequeueing map device
```

- 모든 장치의 경우:

```
# multipathd restorequeueing maps
```

## 9.2. 다중 경로 대화식 콘솔을 사용하여 문제 해결

**multipathd -k** 명령은 다중 경로 데몬의 대화식 인터페이스입니다. 이 명령을 입력하면 대화식 다중 경로 콘솔이 표시됩니다. 이 명령을 실행한 후 도움말 을 입력하여 사용 가능한 명령 목록을 가져오고 **Ctrl+D** 종료할 수 있습니다.

**multipathd** 대화형 콘솔을 사용하여 시스템과 관련된 문제를 해결합니다.

### 절차

1. 콘솔을 종료하기 전에 기본값을 포함하여 다중 경로 구성을 표시합니다.

```
# multipathd -k
multipathd> show config
multipathd> Ctrl+D
```

2. 다중 경로가 **multipath.conf** 파일의 모든 변경 사항을 선택했는지 확인합니다.

```
# multipathd -k
multipathd> reconfigure
multipathd> Ctrl+D
```

3. 경로 검사기가 제대로 작동하는지 확인합니다.

```
# multipathd -k  
multipathd> show paths  
multipathd> Ctrl+D
```

4.

대화형 콘솔을 시작하지 않고 명령줄에서 직접 단일 **multipathd** 대화형 명령을 실행할 수도 있습니다. 예를 들어 다중 경로가 **multipath.conf** 파일에 대한 모든 변경 사항을 선택했는지 확인하려면 다음 명령을 실행합니다.

```
# multipathd reconfigure
```

## 10장. EH\_DEADLINE으로 스토리지 오류 복구의 최대 시간 구성

실패한 **SCSI** 장치를 복구할 수 있는 최대 허용 시간을 구성할 수 있습니다. 이 구성을 사용하면 오류로 인해 스토리지 하드웨어가 응답하지 않는 경우에도 **I/O** 응답 시간을 보장합니다.

### 10.1. EH\_DEADLINE 매개변수

**SCSI** 오류 처리(**EH**) 메커니즘은 실패한 **SCSI** 장치에서 오류 복구를 수행하려고 합니다. **SCSI** 호스트 오브젝트 **eh\_deadline** 매개변수를 사용하면 복구 시간을 최대로 구성할 수 있습니다. 구성된 시간이 만료되면 **SCSI EH**가 중지되고 전체 **HBA**(호스트 버스 어댑터)를 재설정합니다.

**eh\_deadline** 을 사용하면 시간을 줄일 수 있습니다.

- 실패한 경로를 종료하려면,
- 경로를 전환하려면 또는
- **RAID** 슬라이스를 비활성화하려면 다음을 수행합니다.



#### 주의

**eh\_deadline** 이 만료되면 **SCSI EH**는 **HBA**를 재설정하여 해당 **HBA**의 모든 대상 경로뿐만 아니라 실패한 대상 경로에 영향을 미칩니다. 일부 중복 경로를 다른 이유로 사용할 수 없는 경우 **I/O** 오류가 발생할 수 있습니다. 모든 대상에 다중 경로가 구성된 경우에만 **eh\_deadline** 을 활성화합니다. 또한 다중 경로 장치가 완전히 중복되지 않은 경우 **no\_path\_retry** 가 경로를 복구할 수 있을 만큼 충분히 크게 설정되어 있는지 확인해야 합니다.

**eh\_deadline** 매개변수 값은 초 단위로 지정됩니다. 기본 설정은 해제 되어 시간 제한을 비활성화하고 모든 오류 복구를 수행할 수 있습니다.

#### **eh\_deadline**이 유용한 시나리오

대부분의 시나리오에서는 **eh\_deadline** 을 활성화할 필요가 없습니다. **eh\_deadline** 사용은 특정 시나

리오에서 유용할 수 있습니다. 예를 들어 파이버 채널(FC) 스위치와 대상 포트 간에 링크 손실이 발생하고 HBA에서 등록된 **State Change Notifications(RSCN)**를 수신하지 않는 경우. 이러한 경우 오류가 발생하지 않고 항상 I/O 요청 및 오류 복구 명령이 실행됩니다. 이 환경에서 **eh\_deadline** 을 설정하면 복구 시간에 상한이 됩니다. 따라서 실패한 I/O가 **DM Multipath**에서 사용 가능한 다른 경로에서 재시도할 수 있습니다.

다음 조건에서 **eh\_deadline** 매개변수는 I/O 및 오류 복구 명령이 즉시 실패하여 **DM Multipath**를 다시 시도할 수 있기 때문에 추가적인 이점이 없습니다.

- RSCN이 활성화된 경우
- HBA에서 링크를 등록할 수 없는 경우

## 10.2. EH\_DEADLINE 매개변수 설정

이 절차에서는 최대 SCSI 복구 시간을 제한하도록 **eh\_deadline** 매개변수 값을 구성합니다.

### 절차

- 다음 방법 중 하나를 사용하여 **eh\_deadline** 을 구성할 수 있습니다.

- **multipath.conf** 파일의 **defaults** 섹션

**multipath.conf** 파일의 **defaults** 섹션에서 **eh\_deadline** 매개변수를 필요한 초로 설정합니다.

```
# eh_deadline 300
```



참고

RHEL 8.4에서 **multipath.conf** 파일의 **defaults** 섹션을 사용하여 **eh\_deadline** 매개변수를 설정하는 것이 좋습니다.

이 방법을 사용하여 **eh\_deadline** 매개변수를 끄려면 **eh\_deadline** 을 **off** 로 설정합니다.

○

**sysfs**

`/sys/class/scsi_host/host<host-number>/eh_deadline` 파일에 시간 (초)을 작성합니다. 예를 들어 SCSI 호스트 6의 **sysfs** 를 통해 **eh\_deadline** 매개변수를 설정하려면 다음을 수행합니다.

```
# echo 300 > /sys/class/scsi_host/host6/eh_deadline
```

이 방법으로 **eh\_deadline** 매개변수를 끄려면 **echo off** 를 사용합니다.

○

**커널 매개변수**

**scsi\_mod.eh\_deadline** 커널 매개변수를 사용하여 모든 SCSI HBA의 기본값을 설정합니다.

```
# echo 300 > /sys/module/scsi_mod/parameters/eh_deadline
```

이 방법으로 **eh\_deadline** 매개변수를 끄려면 **echo -1** 을 사용합니다.

**추가 리소스**

●

[udev 규칙을 사용하여 eh\\_deadline 및 eh\\_timeout을 영구적으로 설정하는 방법](#)