



Red Hat Enterprise Linux 8

장치 매핑 다중 경로 구성

장치 매핑 멀티패스 기능 사용

Red Hat Enterprise Linux 8 장치 매핑 다중 경로 구성

장치 매핑 멀티패스 기능 사용

Enter your first name here. Enter your surname here.

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

Enter your email address here.

법적 공지

Copyright © 2023 | You need to change the HOLDER entity in the en-US/Configuring_device_mapper_multipath.ent file |.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] 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

이 문서 컬렉션에서는 Red Hat Enterprise Linux 8에서 장치 매퍼 멀티패스(DM-Multipath) 기능을 구성하고 관리하는 방법에 대한 지침을 제공합니다.

차례

보다 포괄적 수용을 위한 오픈 소스 용어 교체	4
RED HAT 문서에 관한 피드백 제공	5
1장. 장치 매핑 다중 경로 개요	6
1.1. RAID 장치가 1개인 활성화/수동 다중 경로 구성	6
1.2. 두 개의 RAID 장치가 있는 활성화/수동 다중 경로 구성	7
1.3. 하나의 RAID 장치가 있는 활성화/활성 다중 경로 구성	8
1.4. DM MULTIPATH 구성 요소	9
1.5. 다중 경로 명령	10
1.6. 다중 경로 명령 출력	11
1.7. 다중 경로 구성 표시	12
1.8. 추가 리소스	13
2장. 다중 경로 장치	14
2.1. 다중 경로 장치 식별자	14
2.2. 논리 볼륨에 있는 다중 경로 장치	15
3장. DM MULTIPATH 구성	17
3.1. DEVICE-MAPPER-MULTIPATH 패키지 확인	17
3.2. 기본 장애 조치 구성에 DM MULTIPATH 설정	17
3.3. 다중 경로 장치를 생성할 때 로컬 디스크 무시	18
3.4. 추가 스토리지 장치 구성	20
3.5. INITRAMFS 파일 시스템에서 다중 경로 설정	21
4장. NVME 장치에서 다중 경로 활성화	22
4.1. 네이티브 NVME 멀티패스 및 DM MULTIPATH	22
4.2. 네이티브 NVME 다중 경로 활성화	22
4.3. NVME 장치에서 DM MULTIPATH 활성화	26
5장. DM-MULTIPATH 구성 파일 수정	30
5.1. 설정 파일 개요	30
5.2. DM MULTIPATH가 장치 시간 초과를 덮어씁니다	31
5.3. 다중 경로 구성 파일 기본값 수정	32
5.4. 특정 장치의 다중 경로 설정 수정	33
5.5. 프로토콜을 사용하여 특정 장치의 다중 경로 구성 수정	35
5.6. 스토리지 컨트롤러의 다중 경로 설정 수정	37
5.7. 모든 장치의 다중 경로 값 설정	39
6장. 장치가 다중 경로 방지	41
6.1. DM MULTIPATH가 경로에 대한 다중 경로 장치를 생성할 때 조건	41
6.2. 특정 장치에서 다중 경로 비활성화 기준	43
6.3. WWID로 다중 경로 비활성화	44
6.4. 장치 이름으로 다중 경로 비활성화	45
6.5. 장치 유형별 다중 경로 비활성화	46
6.6. UDEV 속성으로 다중 경로 비활성화	47
6.7. 장치 프로토콜로 다중 경로 비활성화	48
6.8. 다중 경로가 비활성화된 장치에 대한 예외 추가	50
7장. 다중 경로 볼륨 관리	53
7.1. 온라인 다중 경로 장치 크기 조정	53
7.2. 단일 경로 장치에서 다중 경로 장치로 루트 파일 시스템 이동	54
7.3. 단일 경로 장치에서 다중 경로 장치로 스왑 파일 시스템 이동	55

7.4. DMSETUP 명령으로 장치 매핑 항목 확인	58
7.5. MULTIPATHD 데몬 관리	58
8장. 스토리지 장치 제거	60
8.1. 스토리지 장치의 안전한 제거	60
8.2. 블록 장치 제거	61
9장. DM MULTIPATH 문제 해결	64
9.1. QUEUE_IF_NO_PATH 기능 문제 해결	64
9.2. MULTIPATHD 대화식 콘솔 문제 해결	65
10장. EH_DEADLINE을 사용하여 스토리지 오류 복구를 위한 최대 시간 구성	67
10.1. EH_DEADLINE 매개변수	67
eh_deadline이 유용한 시나리오	67
10.2. EH_DEADLINE 매개변수 설정	68

보다 포괄적 수용을 위한 오픈 소스 용어 교체

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

RED HAT 문서에 관한 피드백 제공

문서에 대한 피드백에 감사드립니다. 어떻게 개선할 수 있는지 알려주십시오.

특정 문구에 대한 의견 제출

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

Bugzilla를 통해 피드백 제출(등록 필요)

1. [Bugzilla](#) 웹 사이트에 로그인합니다.
2. **버전** 메뉴에서 올바른 버전을 선택합니다.
3. **Summary** (요약) 필드에 설명 제목을 입력합니다.
4. **Description** (설명) 필드에 개선을 위한 제안을 입력합니다. 문서의 관련 부분에 대한 링크를 포함합니다.
5. **Submit Bug**를 클릭하십시오.

1장. 장치 매핑 다중 경로 개요

DM Multipath(장치 매핑 멀티패스)를 사용하면 서버 노드와 스토리지 어레이 간에 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 경로는 I/O 경로를 집계하고 집계된 경로로 구성된 새 장치를 생성합니다.

DM Multipath는 다음을 제공합니다.

중복

DM Multipath는 액티브/패시브 구성으로 장애 조치(failover)를 제공할 수 있습니다. 액티브/패시브 구성에서는 언제든지 경로의 절반만 I/O에 사용됩니다. 연결, 스위치 또는 컨트롤러와 같은 I/O 경로의 요소가 실패하면 DM Multipath가 대체 경로로 전환됩니다.

성능 개선

DM Multipath는 I/O가 라운드 로빈 방식으로 경로에 분산되는 활성/활성 모드로 구성할 수 있습니다. 일부 구성에서 DM Multipath는 I/O 경로의 로드를 감지하고 동적으로 부하를 재조정할 수 있습니다.

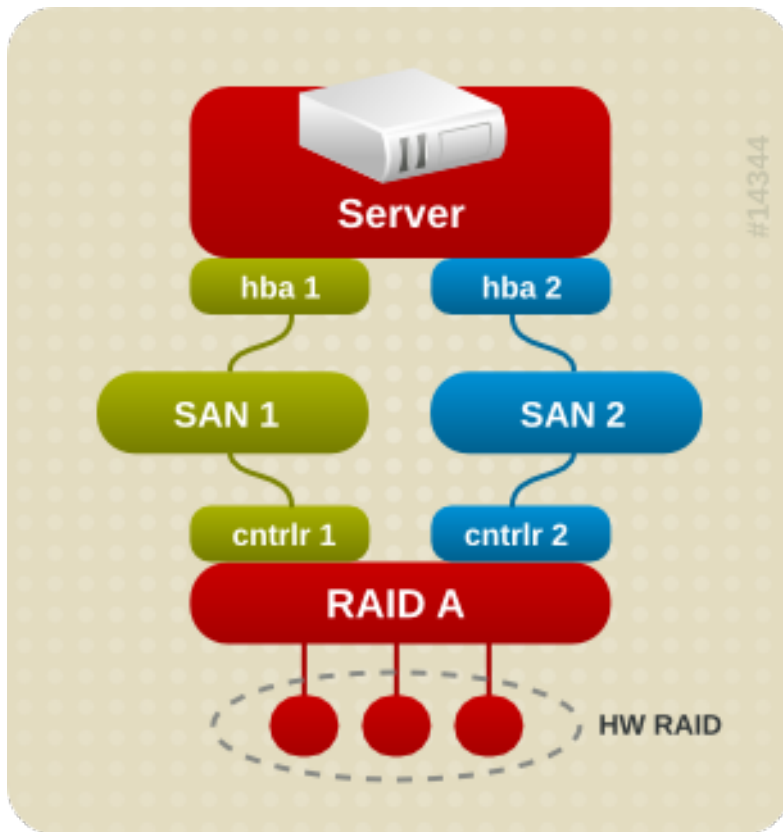
1.1. RAID 장치가 1개인 활성/수동 다중 경로 구성

이 구성에는 서버에 두 개의 HBA(호스트 버스 어댑터)와 두 개의 SAN 스위치와 두 개의 RAID 컨트롤러가 있습니다. 다음은 이 설정에서 오류가 발생할 수 있습니다.

- HBA 실패
- 파이버 채널 케이블 실패
- SAN 스위치 실패
- 배열 컨트롤러 포트 실패

DM Multipath가 구성된 경우 이러한 시점에서 오류가 발생하면 DM Multipath가 대체 I/O 경로로 전환됩니다. 다음 이미지는 서버에서 RAID 장치로의 두 개의 I/O 경로가 있는 구성을 설명합니다. 여기에는 **hba1, SAN1** 및 **cntrlr1**을 통과하는 하나의 I/O 경로와 **hba 2, SAN2** 및 **cntrlr 2**를 통과하는 두 번째 I/O 경로가 있습니다.

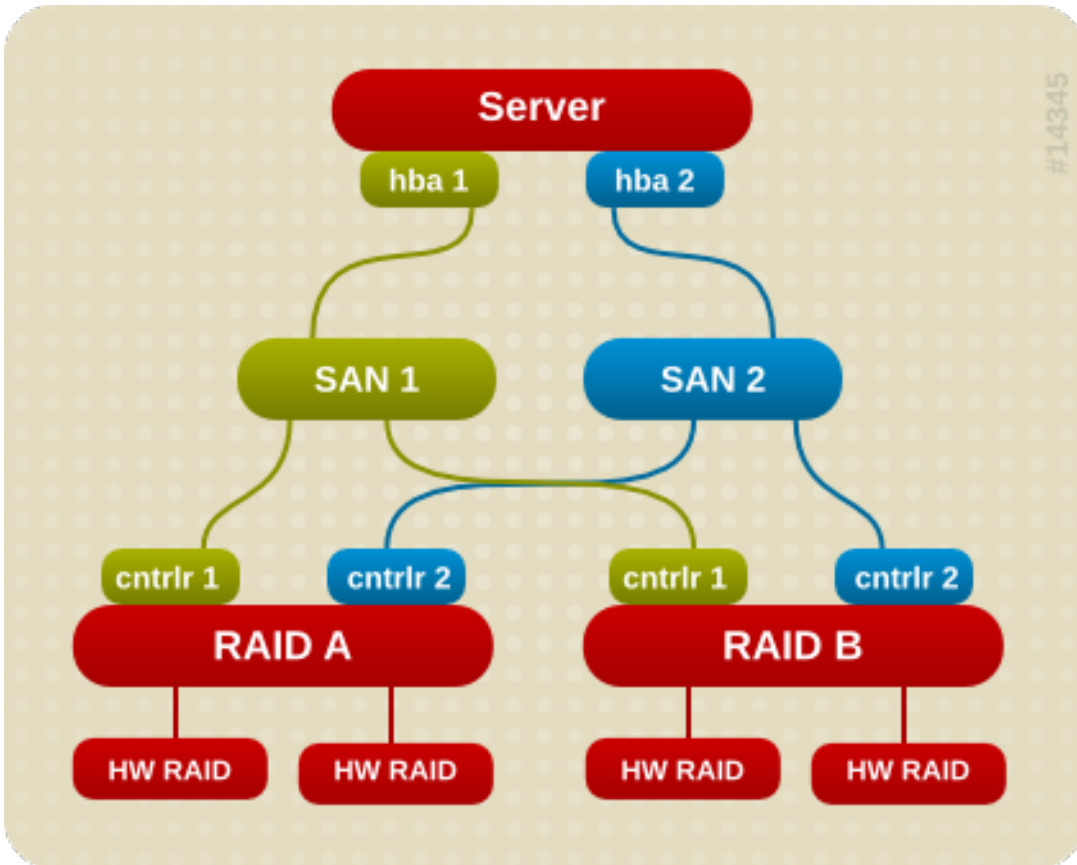
그림 1.1. RAID 장치가 1개인 활성/수동 다중 경로 구성



1.2. 두 개의 RAID 장치가 있는 활성/수동 다중 경로 구성

이 구성에는 서버에 두 개의 HBA, 두 개의 SAN 스위치와 각각 두 개의 RAID 컨트롤러가 있는 두 개의 RAID 장치가 있습니다. DM Multipath가 구성된 경우 I/O 경로 중 하나의 RAID 장치로의 장애로 인해 DM Multipath가 해당 장치의 대체 I/O 경로로 전환됩니다. 다음 이미지는 각 RAID 장치에 대한 두 개의 I/O 경로가 있는 구성을 설명합니다. 여기에 각 RAID 장치에 대한 두 개의 I/O 경로가 있습니다.

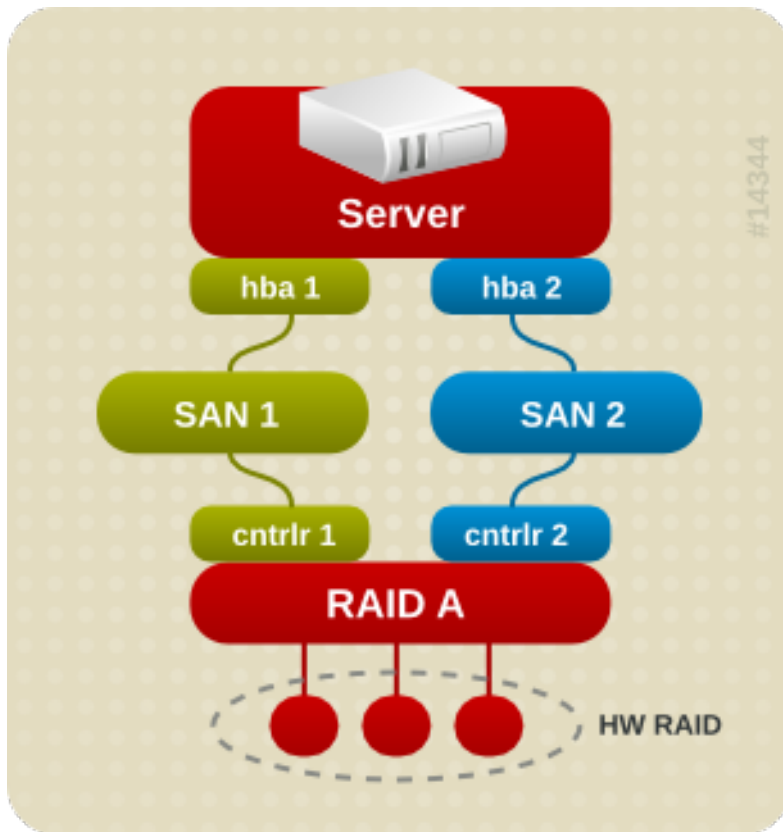
그림 1.2. 두 개의 RAID 장치가 있는 활성/수동 다중 경로 구성



1.3. 하나의 RAID 장치가 있는 활성/수동 다중 경로 구성

이 구성에는 서버에 HBA 두 개, SAN 스위치 두 개 및 두 개의 RAID 컨트롤러가 있습니다. 다음 이미지는 서버에서 스토리지 장치로의 두 개의 I/O 경로가 있는 구성을 설명합니다. 여기서는 이 두 경로에 I/O를 배포할 수 있습니다.

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



1.4. DM MULTIPATH 구성 요소

다음 표에서는 DM Multipath 구성 요소에 대해 설명합니다.

표 1.1. DM Multipath의 구성 요소

구성 요소	설명
dm_multipath 커널 모듈	I/O를 다시 라우팅하고 경로 및 경로 그룹에 대한 페일 오버를 지원합니다.
mpathconf 유틸리티	장치 매핑 다중 경로를 구성하고 활성화합니다.
다중 경로 명령	다중 경로 장치를 나열하고 구성합니다. 또한 블록 장치를 추가할 때마다 udev 가 실행하여 장치가 다중 경로 장치의 일부여야 하는지 여부를 확인합니다.
multipathd 데몬	다중 경로 장치 및 모니터링 경로를 자동으로 생성 및 제거합니다. 경로가 실패하고 되돌아오면 다중 경로 장치를 업데이트할 수 있습니다. 다중 경로 장치에 대한 대화형 변경을 허용합니다. /etc/multipath.conf 파일을 변경하는 경우 서비스를 다시 로드합니다.

<p>kpartx 명령</p>	<p>장치에서 파티션의 장치 매핑 장치를 만듭니다. 이 명령은 위에 파티션 장치를 생성하기 위해 다중 경로 장치를 만들 때 udev에 의해 자동으로 실행됩니다. kpartx 명령은 자체 패키지로 제공되지만 device-mapper-multipath 패키지는 이에 따라 다릅니다.</p>
<p>mpathpersist</p>	<p>다중 경로 장치에서 SCSI-3 영구 예약을 설정합니다. 이 명령은 sg_persist가 다중 경로가 없는 SCSI 장치에 대해 작동하는 방식과 유사하게 작동하지만 다중 경로 장치의 모든 경로에 대한 영구 예약을 처리합니다. 나중에 추가되는 경로에 예약이 올바르게 설정되도록 multipathd와 조정합니다. 이 기능을 사용하려면 reservation_key 특성을 /etc/multipath.conf 파일에 정의해야 합니다. 그렇지 않은 경우 multipathd 데몬은 새로 검색된 경로 또는 복원 경로에 대한 영구 예약을 확인하지 않습니다.</p>

1.5. 다중 경로 명령

multipath 명령은 장치에 대한 여러 경로를 감지하고 결합하는 데 사용됩니다. 다중 경로 장치를 관리하는 데 사용할 수 있는 다양한 옵션을 제공합니다.

다음 표에서는 유용할 수 있는 **multipath** 명령의 몇 가지 옵션을 설명합니다.

표 1.2. 유용한 다중 경로 명령 옵션

옵션	설명
-l	sysfs 및 장치 매핑에서 수집된 현재 다중 경로 구성을 표시합니다.
-ll	sysfs , 장치 매핑 및 시스템에서 사용 가능한 기타 모든 구성 요소에서 수집된 현재 다중 경로 구성을 표시합니다.
-F 장치	명명된 다중 경로 장치를 제거합니다.
-F	사용되지 않은 모든 다중 경로 장치를 제거합니다.
-W 장치	wwid s 파일에서 지정된 장치의 wwid 를 제거합니다.
-W	현재 다중 경로 장치만 포함하도록 wwids 파일을 재설정합니다.

1.6. 다중 경로 명령 출력

다중 경로 장치를 생성, 수정 또는 나열할 때 현재 장치 설정이 표시됩니다. 형식은 다음과 같습니다.

- 다중 경로 장치마다 다음을 수행합니다.

```
action_if_any: alias (wwid_if_different_from_alias) dm_device_name_if_known
vendor,product size=size features='features' hwhandler='hardware_handler'
wp=write_permission_if_known
```

- 각 경로 그룹에 대해 다음을 수행합니다.

```
-- policy='scheduling_policy' prio=prio_if_known status=path_group_status_if_known
```

- 각 경로에 대해 다음을 수행합니다.

```
`- host:channel:id:lun devnode major:minor dm_status_if_known path_status online_status
```

예를 들어 다중 경로 명령의 출력은 다음과 같을 수 있습니다.

```
3600d0230000000000e13955cc3757800 dm-1 WINSYS,SF2372
size=269G features='0' hwhandler='0' wp=rw
|-- policy='round-robin 0' prio=1 status=active
|`- 6:0:0:0 sdb 8:16 active ready running
`-- policy='round-robin 0' prio=1 status=enabled
   `- 7:0:0:0 sdf 8:80 active ready running
```

경로가 시작되어 I/O로 준비되면 경로 상태가 **ready** 또는 **ghost** 입니다. 경로가 다운되면 상태가 결함 또는 **shaky** 입니다. 경로 상태는 `/etc/multipath.conf` 파일에 정의된 폴링 간격에 따라 `multipathd` 데몬에 의해 주기적으로 업데이트됩니다.

가능한 추가 경로 상태 값은 다음과 같습니다.

- I/O 보류 중: 검사기가 이 경로를 적극적으로 확인하고 있으며 상태는 곧 업데이트됩니다.
- I/O 시간 제한: 이것은 결함 과 동일합니다. 사용자는 검사기가 시간 초과 기간 전에 성공 또는 실패를 반환하지 않았음을 알 수 있습니다.

- removed:** 경로는 시스템에서 제거되었으며 곧 다중 경로 장치에서 제거됩니다. 이는 결함 과 동일하게 취급됩니다.
- wild: multipathd** 에서 내부 오류 또는 구성 문제로 인해 경로 검사기를 실행할 수 없었습니다. 이는 오류와 거의 동일합니다. 단, 다중 경로는 경로의 여러 작업을 건너뛸니다.
- 선택되지 않음:** 경로 검사기가 검색되었거나 경로 검사기가 할당되지 않았거나 경로 검사기에 오류가 발생했으므로 이 경로에서 실행되지 않았습니다. 이는 와일드 와 동일하게 처리됩니다.
- 지연:** 경로 검사기에서 경로가 **up**임을 반환하지만, 경로가 최근에 여러 번 실패하고 다중 경로가 이 경우 경로를 지연하도록 다중 경로가 구성되었으므로 경로의 재상태를 늦추고 있습니다.

커널 측면에서 **dm** 상태는 경로 상태와 유사합니다. **active dm** 상태는 **ready** 및 **host** 경로 상태를 다룹니다. 보류 중인 경로 상태는 동일한 **dm** 상태가 아닙니다. 다른 모든 경로 상태는 실패한 **dm** 상태에 매핑됩니다. **dm** 상태는 경로 검사기가 완료될 때까지 현재 상태를 유지합니다.

online_status 에 가능한 값이 실행 중이고 오프라인 상태입니다. 오프라인 상태는 이 **SCSI** 장치가 비활성화되었음을 나타냅니다.



참고

다중 경로 장치를 생성하거나 수정할 때 다중 경로에서는 장치 구성을 출력합니다. 그러나 일부 기능(예: 쓰기 권한 및 기타 기능 정보는 알 수 없음)입니다. 출력을 만들거나 수정하는 동안 선택한 기능 사이에 차이가 있을 수 있습니다. 이것은 정상적인 동작입니다. 생성 후 장치를 나열하여 올바른 상태를 확인합니다.

1.7. 다중 경로 구성 표시

-l 및 **multipath** 명령을 사용하여 현재 다중 경로 구성을 표시할 수 있습니다. **l** 옵션은 **sysfs** 및 장치 매핑의 정보로 수집된 다중 경로 토폴로지를 표시합니다. **ll** 옵션은 시스템의 다른 모든 구성 요소 외에도 **-l** 옵션이 표시되는 정보를 표시합니다.

다중 경로 구성을 표시할 때 **multipath** 명령의 **-v** 옵션을 사용하여 상세 정보 표시 수준을 지정할 수 있습니다. **-v0** 을 지정하면 출력이 발생하지 않습니다. **-v1** 을 지정하면 생성된 경로 또는 업데이트된 다중

경로 이름만 출력하므로 **kpartx** 와 같은 다른 툴에 사용할 수 있습니다. **-v2** 를 지정하면 감지된 모든 경로, 다중 경로 및 장치 맵이 출력됩니다. 더 자세한 정보는 **-v3**, **-v4** 또는 **-v5** 를 지정할 수도 있습니다.

다음 예제는 **multipath -l** 명령의 출력을 보여줍니다.

```
# multipath -l
3600d0230000000000e13955cc3757800 dm-1 WINSYS,SF2372
size=269G features='0' hwhandler='0' wp=rw
|-- policy='round-robin 0' prio=1 status=active
|`- 6:0:0:0 sdb 8:16 active ready running
`-- policy='round-robin 0' prio=1 status=enabled
   `- 7:0:0:0 sdf 8:80 active ready running
```

다음 예제는 **multipath -ll** 명령의 출력을 보여줍니다.

```
# multipath -ll
3600d0230000000000e13955cc3757801 dm-10 WINSYS,SF2372
size=269G features='0' hwhandler='0' wp=rw
|-- policy='round-robin 0' prio=1 status=enabled
|`- 19:0:0:1 sdc 8:32 active ready running
`-- policy='round-robin 0' prio=1 status=enabled
   `- 18:0:0:1 sdh 8:112 active ready running
3600d0230000000000e13955cc3757803 dm-2 WINSYS,SF2372
size=125G features='0' hwhandler='0' wp=rw
`+- policy='round-robin 0' prio=1 status=active
  |- 19:0:0:3 sde 8:64 active ready running
  `- 18:0:0:3 sdj 8:144 active ready running
```

1.8. 추가 리소스

- [multipath\(8\) 및 multipathd\(8\) 매뉴얼 페이지](#)
- [/etc/multipath.conf](#) 파일

2장. 다중 경로 장치

DM Multipath는 기본 장치 위에 단일 다중 경로 장치를 생성하여 I/O 경로를 논리적으로 구성하는 방법을 제공합니다. **DM Multipath**가 없으면 시스템은 I/O 경로가 동일한 서버 노드를 동일한 스토리지 컨트롤러에 연결하는 경우에도 서버 노드에서 스토리지 컨트롤러로 각 경로를 별도의 장치로 처리합니다.

2.1. 다중 경로 장치 식별자

새 장치가 **DM Multipath**를 제어하는 경우 해당 장치는 `/dev/mapper/` 및 `/dev/` 디렉토리에 생성됩니다.



참고

`/dev/dm-X` 형식의 모든 장치는 내부 용도로만 사용되며 관리자가 직접 사용해서는 안 됩니다.

다음은 다중 경로 장치 이름을 설명합니다.

- **user_friendly_names** 구성 옵션을 **no** 로 설정하면 다중 경로 장치의 이름이 **WWID(World Wide Identifier)**로 설정됩니다. 기본적으로 다중 경로 장치의 이름은 **WWID**로 설정됩니다. 장치 이름은 `/dev/mapper/WWID` 입니다. 또한 `/dev/dm-X`라는 이름의 `/dev/` 디렉토리에 생성됩니다.
- 또는 `/etc/multipath.conf` 파일에서 **user_friendly_names** 옵션을 **yes** 로 설정할 수 있습니다. 이는 **multipath** 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 동일하게 보장되지는 않습니다. 마찬가지로 `/etc/multipath.conf` 파일에서 **alias** 옵션을 설정하면 클러스터의 모든 노드에서 이름이 자동으로 일관되지 않습니다.



참고

LVM을 사용하여 다중 경로 장치에서 논리적 장치를 생성하는 경우 문제가 발생하지 않습니다. 모든 노드에서 다중 경로 장치 이름을 일관되게 유지하려면 **user_friendly_names** 옵션을 비활성화하는 것이 좋습니다.

예를 들어, 영역이 없는 단일 **FC** 스위치를 통해 2개의 포트가 있는 스토리지 컨트롤러에 2개의 **HBA**가 연결된 노드는 `/dev/sda`, `/dev/sdb`, `/dev/sdc`, `/dev/sdd` 를 확인합니다. **DM Multipath**는 다중 경로 구성에 따라 I/O를 4개의 기본 장치로 다시 라우팅하는 고유한 **WWID**를 사용하여 단일 장치를 생성합니다.

`user_friendly_names` 및 `alias` 옵션 외에도 다중 경로 장치에 다른 특성도 있습니다. `/etc/multipath.conf` 파일의 `multipaths` 섹션에서 해당 장치에 대한 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 볼륨에 있는 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 이름이면 `pvcreate /dev/mapper/mpatha` 명령은 `/dev/mapper/mpatha` 를 물리 볼륨으로 표시합니다.

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

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



참고

파티션을 구성한 전체 장치에서 LVM 물리 볼륨을 생성하려고 하면 `pvcreate` 명령이 실패합니다. 모든 블록 장치에 대해 달리 지정하지 않는 경우 `Anaconda` 및 `Kickstart` 설치 프로그램은 빈 파티션 테이블을 생성합니다. 파티션을 만드는 대신 전체 장치를 사용하려면 장치에서 기존 파티션을 제거합니다. `kpartx -d device` 명령 및 `fdisk` 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 물리적 장치로 사용하는 LVM 논리 볼륨을 생성할 때 선택적으로 `/etc/lvm/lvm.conf` 파일에 필터를 포함하여 다중 경로 장치의 디스크를 제외할 수 있습니다. 이는 배열이 I/O를 수신할 때 수동 경로의 활성 경로를 자동으로 변경하는 경우 이러한 장치가 필터링되지 않으면 LVM

에서 패시브 경로를 검사할 때마다 다중 경로가 장애 조치되고 실패할 때마다 다중 경로에서 장애 조치를 수행하기 때문입니다.

커널은 사용할 올바른 하드웨어 처리기를 자동으로 감지하여 활성/수동 상태를 변경합니다. 상태를 변경하기 위해 개입해야 하는 활성/수동 경로의 경우 다중 경로에서는 필요에 따라 이 하드웨어 핸들러를 자동으로 사용합니다. 커널이 사용할 올바른 하드웨어 처리기를 자동으로 탐지하지 않으면 **"hardware_handler"** 옵션과 함께 **multipath.conf** 파일에서 사용할 하드웨어 처리기를 구성할 수 있습니다. 수동 경로를 활성화하기 위해 명령이 필요한 활성/수동 배열의 경우 **LVM**에서 이 문제가 발생할 때 경고 메시지를 출력합니다.

구성에 따라 **LVM**에서 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
```
- 실패를 읽습니다.

```
/dev/sde: read failed after 0 of 4096 at 0: Input/output error
```

다음은 언급된 오류의 이유입니다.

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

추가 리소스

- **LVM.conf** 매뉴얼 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 존재하는 경우 **mpathconf** 유틸리티가 이를 편집합니다.
- **/etc/multipath.conf** 파일이 없으면 **mpathconf** 유틸리티에서 **/etc/multipath.conf** 파일을 처음부터 새로 생성합니다.

3.1. DEVICE-MAPPER-MULTIPATH 패키지 확인

시스템에서 **DM Multipath**를 설정하기 전에 시스템이 최신 상태이며 **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 않은 경우 다음 명령을 실행하여 설치합니다.

```
# {PackageManager} install device-mapper-multipath
```

3.2. 기본 장애 조치 구성에 DM MULTIPATH 설정

multipathd 데몬을 시작하기 전에 **/etc/multipath.conf** 파일을 편집해야 하는 경우 기본 장애 조치 구성에 대해 **DM Multipath**를 설정하려면 다음 절차를 사용하십시오.

절차

1. 다중 경로 구성 파일을 활성화합니다.

```
# mpathconf --enable
```

2. 필요한 경우 `/etc/multipath.conf` 파일을 편집합니다. **DM Multipath**의 기본 설정은 시스템으로 컴파일되며 `/etc/multipath.conf` 파일에 명시적으로 설정할 필요가 없습니다.

`path_grouping_policy`의 기본값은 `failover`로 설정되어 있으므로 이 예제에서는 `/etc/multipath.conf` 파일을 편집할 필요가 없습니다.

구성 파일의 초기 `defaults` 섹션은 다중 경로 장치의 이름이 `/dev/mapper/mpathn` 형식으로 되어 있도록 시스템을 구성합니다. 이 설정없이 다중 경로 장치의 이름은 장치의 **WWID**에 별칭됩니다. 사용자 친숙한 이름을 사용하지 않으려면 다음 명령을 입력합니다.

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

다중 경로 데몬을 시작한 후 다중 경로 구성 파일을 편집해야 하는 경우, 변경 사항을 적용하려면 `systemctl reload multipathd.service` 명령을 실행해야 합니다.

3. 구성 파일을 저장하고 필요한 경우 편집기를 종료합니다.
4. 다중 경로 데몬을 시작하고 다중 경로 장치를 생성합니다.

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

참고

`device-mapper-multipath` 패키지를 제거하는 경우 `/etc/multipath.conf` 파일 또는 `/etc/multipath` 디렉터리의 모든 파일을 제거하지 않습니다. 해당 디렉터리에는 현재 나열된 파일만 포함될 수 있으므로 해당 디렉터리에는 더 많은 파일이 포함될 수 있습니다. 나중에 `device-mapper-multipath` 패키지를 설치할 때 해당 파일을 수동으로 제거해야 할 수 있습니다.

3.3. 다중 경로 장치를 생성할 때 로컬 디스크 무시

일부 시스템에는 내부 디스크에 대한 로컬 **SCSI** 카드가 있고 **DM Multipath**가 이러한 장치에는 권장되지 않습니다. `find_multipaths` 구성 매개 변수를 의로 설정하면 이러한 장치에서 다중 경로를 비활성화할

필요가 없습니다.

`find_multipaths` 구성 매개변수를 의 로 설정하지 않으면 다음 절차를 사용하여 다중 경로를 구성할 때 로컬 디스크를 무시하도록 **DM Multipath** 구성 파일을 수정할 수 있습니다.

절차

1.

내부 디스크인 디스크를 확인합니다. 이 예에서 `/dev/sda` 는 내부 디스크입니다.

-

기존 다중 경로 장치를 표시합니다.

```
# multipath -v2 -l

SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1 dm-2 WINSYS,SF2372
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 Multipath에서 생성할 수 있는 추가 다중 경로 장치를 표시합니다.

```
# multipath -v2 -d

: SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1 undef WINSYS,SF2372
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
```

2.

이 장치를 포함하도록 `/etc/multipath.conf` 파일의 **blacklist** 섹션을 편집합니다.

WWID 특성을 사용하여 장치를 식별합니다. **devnode** 유형을 사용하여 **the sda** 장치를 식별할 수 있지만, 재부팅 시 `/dev/sda` 가 동일하게 보장되지 않기 때문에 안전한 절차는 아닙니다.

이전 예에서 `/dev/sda` 장치의 **WWID**는 **SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1** 입니다. 이 장치를 무시하려면 `/etc/multipath.conf` 파일에 다음을 포함합니다.

```
blacklist {
    wwid SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1
}
```

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

- [multipath.conf\(5\) 매뉴얼 페이지](#)

3.4. 추가 스토리지 장치 구성

기본적으로 **DM Multipath**는 **DM Multipath**를 지원하는 가장 일반적인 스토리지 어레이를 지원합니다.

절차

- 지원되는 장치를 포함하여 기본 구성 값을 확인합니다.

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

- 선택 사항: 기본적으로 알려진 다중 경로 장치로 지원되지 않는 추가 스토리지 장치를 추가하려면 `/etc/multipath.conf` 파일을 편집하고 적절한 장치 정보를 삽입합니다.

다음 예제에서는 **HP Open-V** 시리즈에 대한 정보를 추가하는 방법을 설명합니다. 이렇게 하면 모든 경로가 실패한 후 재시도당 1분 또는 12번 재시도에 대해 장치가 대기열로 설정됩니다.


```

devices {
    device {
        vendor "HP"
        product "OPEN-V"
        no_path_retry 12
    }
}

```

3.5. INITRAMFS 파일 시스템에서 다중 경로 설정

initramfs 파일 시스템에서 다중 경로를 설정할 수 있습니다. 부팅 후 **initramfs** 파일 시스템이 종료될 때까지 다중 경로를 원하는 장치를 사용하지 않는 경우 이를 설정할 필요가 없습니다.

사전 요구 사항

- 시스템에 **DM** 다중 경로를 구성했습니다.

절차

- 다음 명령을 실행하여 다중 경로 구성 파일을 사용하여 **initramfs** 파일 시스템을 다시 빌드합니다.

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

initramfs 파일 시스템에서 다중 경로를 실행하고 다중 경로 구성 파일을 변경하는 경우 변경 사항을 적용하려면 **initramfs** 파일 시스템을 다시 빌드해야 합니다. 루트 장치에서 다중 경로를 사용하는 경우 **dracut** 명령을 실행하면 **initramfs**에 **multipath** 모듈이 자동으로 추가됩니다.

- 선택 사항: **initramfs**에서 실행 중 다중 경로가 필요하지만 다중 경로 루트 장치를 설정하지 않은 경우 다음을 실행합니다.

```
# echo add_dracutmodules+="multipath" > /etc/dracut.conf.d/multipath.conf
# dracut --force
```

참고

dracut 명령에는 다중 경로가 더 이상 필요하지 않은 경우에도 **initramfs**에 다중 경로가 포함됩니다. 다중 경로 포함을 중지하려면 다음을 실행합니다.

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

4장. NVMe 장치에서 다중 경로 활성화

파이버 채널(FC)과 같이 패브릭 전송을 통해 시스템에 연결된 NVMe 장치를 다중 경로로 지정할 수 있습니다. 여러 다중 경로 솔루션 중에서 선택할 수 있습니다.

4.1. 네이티브 NVMe 멀티패스 및 DM MULTIPATH

NVMe 장치는 기본 다중 경로 기능을 지원합니다. NVMe에서 다중 경로를 구성할 때 표준 DM Multipath 프레임워크와 네이티브 NVMe 다중 경로 중 하나를 선택할 수 있습니다.

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

네이티브 NVMe 다중 경로가 활성화되면 모든 NVMe 장치에 전역적으로 적용됩니다. 더 높은 성능을 제공할 수 있지만 DM Multipath에서 제공하는 모든 기능을 포함하지는 않습니다. 예를 들어 기본 NVMe 다중 경로에서는 numa 및 round-robin 경로 선택 방법만 지원합니다.

Red Hat Enterprise Linux 8에서 DM Multipath를 기본 다중 경로 솔루션으로 사용하는 것이 좋습니다.

4.2. 네이티브 NVMe 다중 경로 활성화

이 절차에서는 네이티브 NVMe 다중 경로 솔루션을 사용하여 연결된 NVMe 장치에서 멀티패스를 활성화합니다.

사전 요구 사항

- NVMe 장치가 시스템에 연결되어 있습니다.

패브릭 전송에 NVMe를 연결하는 방법에 대한 자세한 내용은 [패브릭 장치보다 NVMe 개요](#)를 참조하십시오.

절차

1. 커널에서 기본 NVMe 다중 경로가 활성화되어 있는지 확인합니다.

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

명령은 다음 중 하나를 표시합니다.

N

기본 **NVMe** 다중 경로가 비활성화되어 있습니다.

Y

네이티브 **NVMe** 다중 경로가 활성화되어 있습니다.

2.

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

-

커널 옵션 사용:

- i.

커널 명령줄에 **nvme_core.multipath=Y** 옵션을 추가합니다.

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

- ii.

64비트 IBM Z 아키텍처에서 부팅 메뉴를 업데이트합니다.

```
# zipl
```

- iii.

시스템을 재부팅합니다.

-

커널 모듈 구성 파일 사용:

- i.

다음 콘텐츠를 사용하여 **/etc/modprobe.d/nvme_core.conf** 구성 파일을 만듭니다.

```
options nvme_core multipath=Y
```

- ii.

initramfs 파일 시스템을 백업합니다.

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

iii.

initramfs 파일 시스템을 다시 빌드합니다.

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

iv.

시스템을 재부팅합니다.

3.

선택 사항: 실행 중인 시스템에서 **NVMe** 장치의 **I/O** 정책을 변경하여 사용 가능한 모든 경로에서 **I/O**를 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 생성합니다.

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

검증

1.

시스템이 **NVMe** 장치를 인식하는지 확인합니다.

```
# 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 모든 **NVMe** 하위 시스템을 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** 는 장치가 파이버 채널 전송을 통해 연결되어 있음을 나타내며 **nvme tcp** 는 장치가 **TCP**를 통해 연결되어 있음을 나타냅니다.

3.

커널 옵션을 편집한 경우 커널 명령줄에서 네이티브 **NVMe** 다중 경로가 활성화되어 있는지 확인합니다.

```
# cat /proc/cmdline

BOOT_IMAGE=[...] nvme_core.multipath=Y
```

4.

DM Multipath가 **NVMe** 네임스페이스(예: **nvme0c0n1 ~ nvme0c3n1**)를 **nvme 0c3n1 ~ nvme3n1** 로 보고하는 지 확인합니다.

```
# multipath -e -ll | grep -i nvme

uuid.8ef20f70-f7d3-4f67-8d84-1bb16b2bfe03 [nvme]:nvme0n1 NVMe,Linux,4.18.0-2
|`- 0:0:1 nvme0c0n1 0:0 n/a optimized live
|`- 0:1:1 nvme0c1n1 0:0 n/a optimized live
|`- 0:2:1 nvme0c2n1 0:0 n/a optimized live
|`- 0:3:1 nvme0c3n1 0:0 n/a optimized live

uuid.44c782b4-4e72-4d9e-bc39-c7be0a409f22 [nvme]:nvme0n2 NVMe,Linux,4.18.0-2
|`- 0:0:1 nvme0c0n1 0:0 n/a optimized live
|`- 0:1:1 nvme0c1n1 0:0 n/a optimized live
|`- 0:2:1 nvme0c2n1 0:0 n/a optimized live
|`- 0:3:1 nvme0c3n1 0:0 n/a optimized live
```

5.

I/O 정책을 변경한 경우 **round-robin** 이 **NVMe** 장치의 활성 **I/O** 정책인지 확인합니다.

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

round-robin
```

추가 리소스

•

[커널 명령줄 매개변수 구성](#)

4.3. NVMe 장치에서 DM MULTIPATH 활성화

이 절차에서는 **DM Multipath** 솔루션을 사용하여 연결된 **NVMe** 장치에서 멀티패스를 활성화합니다.

사전 요구 사항

- **NVMe** 장치가 시스템에 연결되어 있습니다.

패브릭 전송에 **NVMe**를 연결하는 방법에 대한 자세한 내용은 [패브릭 장치보다 NVMe 개요](#)를 참조하십시오.

절차

1. 네이티브 **NVMe** 다중 경로가 비활성화되었는지 확인합니다.

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

명령은 다음 중 하나를 표시합니다.

N

기본 **NVMe** 다중 경로가 비활성화되어 있습니다.

Y

네이티브 **NVMe** 다중 경로가 활성화되어 있습니다.

2. 네이티브 **NVMe** 멀티패스가 활성화된 경우 이를 비활성화합니다.

- a. 커널 명령줄에서 **nvme_core.multipath=Y** 옵션을 제거합니다.

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

- b. **64비트 IBM Z** 아키텍처에서 부팅 메뉴를 업데이트합니다.

```
# zipl
```

- c. `/etc/modprobe.d/nvme_core.conf` 파일에서 `nvme_core=Y` 줄이 있는 경우 해당 옵션을 제거합니다.
- d. 시스템을 재부팅합니다.

3. **DM Multipath**가 활성화되어 있는지 확인합니다.

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

4. 사용 가능한 모든 경로에 I/O를 배포합니다. `/etc/multipath.conf` 파일에 다음 내용을 추가합니다.

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



참고

DM Multipath가 NVMe 장치를 관리하는 경우 `/sys/class/nvme-subsys0/iopolicy` 구성 파일은 I/O 배포에 영향을 미치지 않습니다.

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

```
# multipath -r
```

6. **initramfs** 파일 시스템을 백업합니다.

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

7. **initramfs** 파일 시스템을 다시 빌드합니다.

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

1. 시스템이 **NVMe** 장치를 인식하는지 확인합니다.

```
# 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	
/dev/nvme1n1	a34c4f3a0d6f5cec	Linux	1 250.06 GB /
250.06 GB	512 B + 0 B	4.18.0-2	
/dev/nvme1n2	a34c4f3a0d6f5cec	Linux	2 250.06 GB /
250.06 GB	512 B + 0 B	4.18.0-2	
/dev/nvme2n1	a34c4f3a0d6f5cec	Linux	1 250.06 GB /
250.06 GB	512 B + 0 B	4.18.0-2	
/dev/nvme2n2	a34c4f3a0d6f5cec	Linux	2 250.06 GB /
250.06 GB	512 B + 0 B	4.18.0-2	
/dev/nvme3n1	a34c4f3a0d6f5cec	Linux	1 250.06 GB /
250.06 GB	512 B + 0 B	4.18.0-2	
/dev/nvme3n2	a34c4f3a0d6f5cec	Linux	2 250.06 GB /
250.06 GB	512 B + 0 B	4.18.0-2	

2. 연결된 모든 **NVMe** 하위 시스템을 나열합니다. 명령이 **nvme0n1~nvme 3n 2** 로 보고하는 지 확인합니다(예: **nvme0c0n1 ~ nvme0c3n1**) :

```
# nvme list-subsys
```

```
nvme-subsys0 - NQN=testnqn
\
+- nvme0 fc traddr=nn-0x20000090fadd5979:pn-0x10000090fadd5979 host_traddr=nn-0x20000090fac7e1dd:pn-0x10000090fac7e1dd live
+- nvme1 fc traddr=nn-0x20000090fadd597a:pn-0x10000090fadd597a 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
```

```
# multipath -ll
```

```
mpathae (uuid.8ef20f70-f7d3-4f67-8d84-1bb16b2bfe03) dm-36 NVME, Linux
size=233G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=50 status=active
  |- 0:1:1:1 nvme0n1 259:0 active ready running
  |- 1:2:1:1 nvme1n1 259:2 active ready running
  |- 2:3:1:1 nvme2n1 259:4 active ready running
  `-- 3:4:1:1 nvme3n1 259:6 active ready running
```



```
mpathaf (uuid.44c782b4-4e72-4d9e-bc39-c7be0a409f22) dm-39 NVME,Linux
size=233G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=50 status=active
  |- 0:1:2:2 nvme0n2 259:1 active ready running
  |- 1:2:2:2 nvme1n2 259:3 active ready running
  |- 2:3:2:2 nvme2n2 259:5 active ready running
  ` - 3:4:2:2 nvme3n2 259:7 active ready running
```

추가 리소스

- [커널 명령줄 매개변수 구성](#)
- [DM Multipath 구성.](#)

5장. DM-MULTIPATH 구성 파일 수정

기본적으로 **DM Multipath**는 다중 경로 사용을 위한 구성 값을 제공합니다. 또한 **DM Multipath**에는 **DM Multipath**를 지원하는 가장 일반적인 스토리지 어레이에 대한 지원이 포함되어 있습니다. `/etc/multipath.conf` 구성 파일을 편집하여 **DM Multipath**의 기본 구성 값을 재정의할 수 있습니다. 필요한 경우 기본 스토리지 어레이에서 지원되지 않는 스토리지를 구성 파일에 추가할 수도 있습니다.

지원되는 장치를 포함한 기본 구성 값에 대한 자세한 내용은 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 섹션 또는 기본값에서 변경하려는 섹션만 지정해야 합니다. 사용자 환경과 관련이 없거나 기본값을 재정의할 필요가 없는 파일의 섹션이 있는 경우, 초기 파일에 있기 때문에 주석 처리되어 있는 것입니다.

구성 파일에서는 정규 표현식 설명 구문을 사용할 수 있습니다.

5.1. 설정 파일 개요

다중 경로 구성 파일은 다음 섹션으로 나뉩니다.

블랙리스트

다중 경로로 고려되지 않는 특정 장치 목록입니다.

blacklist_exceptions

blacklist 섹션의 매개변수에 따라 무시되는 다중 경로 장치 목록입니다.

defaults

DM Multipath의 일반적인 기본 설정.

다중 경로

개별 다중 경로 장치의 특성 설정. 이러한 값은 구성 파일의 **overrides**, **devices** 및 **defaults** 섹션에 지정된 내용을 덮어씁니다.

장치

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

덮어쓰기

모든 장치에 적용되는 설정입니다. 이러한 값은 구성 파일의 **devices** 및 **defaults** 섹션에 지정된 내용을 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. DM MULTIPATH가 장치 시간 초과를 덮어씁니다

recovery_tmo sysfs 옵션은 특정 **iSCSI** 장치에 대한 시간 초과를 제어합니다. 다음 옵션은 **recovery_tmo** 값을 전역적으로 덮어씁니다.

- **replacement_timeout** 구성 옵션은 모든 **iSCSI** 장치의 **recovery_tmo** 값을 전역적으로 덮어씁니다.
- **DM Multipath**에서 관리하는 모든 **iSCSI** 장치의 경우 **DM Multipath**의 **fast_io_fail_tmo** 옵션은 **recovery_tmo** 값을 전역적으로 덮어씁니다.

DM Multipath의 `fast_io_fail_tmo` 옵션은 파이버 채널 장치의 `fast_io_fail_tmo` 옵션도 재정의합니다.

DM Multipath `fast_io_fail_tmo` 옵션이 `replacement_timeout` 보다 우선합니다. DM Multipath는 `multipathd` 서비스가 다시 로드될 때 항상 `recovery_tmo` 를 재설정하므로 `replacement_timeout` 을 사용하여 DM Multipath에서 관리하는 장치에서 `recovery_tmo` 를 덮어쓰는 것을 권장하지 않습니다.

5.3. 다중 경로 구성 파일 기본값 수정

`/etc/multipath.conf` 구성 파일에는 다음과 같이 `user_friendly_names` 매개 변수를 `yes` 로 설정하는 `defaults` 섹션이 포함되어 있습니다.

```
defaults {
    user_friendly_names yes
}
```

그러면 `user_friendly_names` 매개 변수의 기본값을 덮어씁니다. `multipath.conf` 파일의 `defaults` 섹션에 설정된 기본값은 DM Multipath에서 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 재구성하십시오.

```
# service multipathd reload
```

추가 리소스

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

5.4. 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 사용됩니다. `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 WWID가 **3600508b156d70001200000b0000** 이고 노란색 이라는 심볼릭 이름이 있습니다.
- 예제의 두 번째 다중 경로 장치에는 WWID가 **1DEC_321816758474** 이고 심볼릭 이름은 빨간색 입니다.

이 예제에서는 **rr_weight** 특성이 우선순위 로 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 재구

성하십시오.

```
# service multipathd reload
```

추가 리소스

- **multipath.conf(5) 매뉴얼 페이지**

5.5. 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
  }
}
```

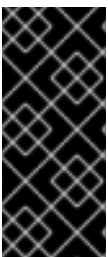
- **scsi:iscsi** 프로토콜을 사용하는 경로 장치에 대한 설정:

```
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**

5.6. 스토리지 컨트롤러의 다중 경로 설정 수정

multipath.conf 구성 파일의 **devices** 섹션에서는 개별 스토리지 장치의 속성을 설정합니다. **DM Multipath**는 다중 경로에 지정된 속성으로 덮어쓰거나 장치가 포함된 경로 의 **multipath.conf** 파일의 섹션을 덮어쓰지 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 많은 장치는 기본적으로 다중 경로 구성에 포함되어 있습니다.

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 매개 변수는 **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 핸들러 커널 모듈이 필요합니다. 현재 사용 가능한 하드웨어 핸들러는 **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**

5.7. 모든 장치의 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 장치가 포함된 경로에 대해 **multipath.conf** 파일의 **multipaths** 섹션에 지정된 속성으로 덮어쓰지 않는 한 모든 장치에 대해 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 생성할 때 조건

DM Multipath에는 경로에 대한 다중 경로 장치를 만들지 또는 경로를 무시할지 여부를 결정하는 기본 규칙 세트가 있습니다. 동작을 구성할 수 있습니다.

find_multipaths 구성 매개 변수를 **off** 로 설정하면 다중 경로에서 명시적으로 비활성화되지 않은 모든 경로에 대해 다중 경로 장치를 항상 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 묶여 있습니다.

```
# 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** 외에도 **find_multipaths** 를 다음 값으로 설정할 수도 있습니다.

strict

다중 경로는 이전에 다중 경로가 수행되지 않았으며 **/etc/multipath/wwids** 파일에 없는 경로를 허용하지 않습니다.

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 기본 제공 기본값이 꺼져 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 다중 경로를 비활성화할 필요가 없습니다.

이전에 생성된 다중 경로 장치를 차단 목록에 추가하면 **-w** 옵션을 사용하여 해당 장치의 **WWID**를 **/etc/multipath/wwids** 파일에서 제거하면 다른 프로그램과 관련된 문제를 방지할 수 있습니다. 예를 들어, **/etc /multipath/ wwids** 파일에서 **WWID 3600d0230000000000e13954f89300** 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 wwids
```

추가 리소스

- **multipath.conf(5) 매뉴얼 페이지**

6.2. 특정 장치에서 다중 경로 비활성화 기준

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

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

DM Multipath는 모든 장치에 대해 이러한 기준을 다음 순서로 평가합니다.

1. **property**
2. **devnode**
3. **device**
4. **protocol**

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 기본적으로 다양한 장치 유형이 비활성화됩니다.

추가 리소스

- [다중 경로가 비활성화된 장치에 대한 예외 추가](#)

6.3. WWID로 다중 경로 비활성화

WWID(World-Wide Identification)를 통해 개별 장치에서 다중 경로를 비활성화할 수 있습니다.

절차

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

다음 예제에서는 **WWID**가 **26353900f02796769** 인 장치를 비활성화하는 **DM Multipath** 구성 파일의 행을 보여줍니다.

```
blacklist {
    wwid 26353900f02796769
}
```

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
```

6.4. 장치 이름으로 다중 경로 비활성화

DM Multipath가 다중 경로 장치로 그룹화하지 않도록 장치 유형별로 다중 경로를 비활성화할 수 있습니다.

절차

1.

`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 이름은 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.

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

•

[다중 경로가 비활성화된 장치에 대한 예외 추가](#)

6.5. 장치 유형별 다중 경로 비활성화

`device` 섹션을 사용하여 장치에서 멀티패스를 비활성화할 수 있습니다.

절차

1.

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

다음 예제에서는 모든 **IBM DS4200** 및 **HP** 장치에서 다중 경로를 비활성화합니다.

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

```

    product "*"
  }
}

```

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
```

6.6. UDEV 속성으로 다중 경로 비활성화

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

절차

1.

`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` 가 있는 모든 장치에서 다중 경로를 비활성화합니다.

```

blacklist {
    property "ID_ATA"
}

```

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
```

6.7. 장치 프로토콜로 다중 경로 비활성화

장치 프로토콜을 사용하여 장치에서 다중 경로를 비활성화할 수 있습니다.

절차

1.

선택 사항: 경로에서 사용 중인 프로토콜을 확인합니다.

```
# multipathd show paths format "%d %P"
```

2.

`protocol` 섹션을 사용하여 `/etc/multipath.conf` 구성 파일에서 장치를 비활성화합니다.

다음 예제에서는 정의되지 않은 프로토콜 또는 알 수 없는 **SCSI** 전송 유형이 있는 모든 장치에서 다중 경로를 비활성화합니다.

```
blacklist {
    protocol "scsi:unspec"
    protocol "undef"
}
```

DM Multipath는 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**
- **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 다중 경로가 비활성화되어 있습니다.

절차

1.

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

구성 파일의 `blacklist_exceptions` 섹션에 장치를 지정할 때 `blacklist` 섹션에 지정된 것과 동일한 기준을 사용하여 예외를 지정해야 합니다. 예를 들어 비활성화된 장치가 `WWID`와 연결되어 있는 경우에도 `WWID` 예외는 `devnode` 항목에서 비활성화된 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 지정된 **WWID**를 제외한 모든 장치를 비활성화하는 블랙리스트 항목을 전환할 수 있습니다.

```

blacklist {
    wwid "!3600d0230000000000e13955cc3757803"
}

```

예 6.2. udev 속성의 예외

속성 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

DM Multipath에서 제공하는 몇 가지 명령은 다중 경로 볼륨을 관리하는 데 사용할 수 있습니다.

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

7.1. 온라인 다중 경로 장치 크기 조정

온라인 다중 경로 장치의 크기를 조정하려면 다음 절차를 사용하십시오.

절차

1. 물리적 장치의 크기를 조정합니다.
2. 다음 명령을 실행하여 논리 단위 번호(LUN)에 대한 경로를 찾습니다.

```
# multipath -l
```

3. 경로의 크기를 조정합니다. SCSI 장치의 경우 다음 명령과 같이 장치에 대한 재검사 파일에 1을 추가하면 SCSI 드라이버가 다시 스캔됩니다.

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

각 경로 장치에 대해 이 명령을 실행해야 합니다. 예를 들어 경로 장치가 **sdb**, **sde**, **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 장치의 크기를 조정합니다.

```
# multipathd resize map multipath_device
```

5. 파일 시스템의 크기를 조정합니다(LVM 또는 DOS 파티션이 사용되지 않는다고 가정).

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

7.2. 단일 경로 장치에서 다중 경로 장치로 루트 파일 시스템 이동

단일 경로 장치에 시스템을 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** 패키지를 설치했습니다.

절차

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

```
# yum 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 방지**에 설명된 대로 **/etc/multipath.conf** 파일의 **blacklist** 및 **blacklist_exceptions** 섹션을 편집합니다.

4. 다중 경로를 통해 **root** 장치 상단에 다중 경로 장치를 빌드하려면 다음 명령을 입력합니다. 이 명령은 또한 **find_multipaths**가 경로가 하나뿐인 경우에도 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.

`multipath` 명령을 실행하고 다음 형식의 행에 대해 출력을 검색하여 구성 파일이 올바르게 설정되었는지 확인합니다. 이는 명령이 다중 경로 장치를 생성하지 못했음을 나타냅니다.

```
date wwid: ignoring map
```

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

시스템을 종료합니다.

8.

시스템을 부팅합니다.

9.

다른 경로를 머신에 표시합니다.

검증 단계

-

다음 명령을 실행하여 다중 경로 장치가 생성되었는지 확인합니다.

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

7.3. 단일 경로 장치에서 다중 경로 장치로 스왑 파일 시스템 이동

기본적으로 스왑 장치는 논리 볼륨으로 설정됩니다. 논리 볼륨 그룹을 구성하는 물리 볼륨에 다중 경로

를 설정하는 한 다중 경로 장치로 구성하는 특별한 절차는 필요하지 않습니다. 그러나 스왑 장치가 **LVM** 볼륨이 아닌 경우 장치 이름으로 마운트되는 경우 **/etc/fstab** 파일을 편집하여 적절한 다중 경로 장치 이름으로 전환해야 할 수 있습니다.

절차

1. 장치의 **WWID**를 **/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. **multipath** 명령을 실행하고 다음 형식의 행에 대해 출력을 검색하여 구성 파일이 올바르게 설정되었는지 확인합니다.

```
date wwid: ignoring map
```

이는 명령이 다중 경로 장치를 생성하지 못했음을 나타냅니다.

예를 들어 장치의 **WWID**가 **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 이전 장치 경로를 다중 경로 장치로 바꿉니다.

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.

시스템을 종료합니다.

7.

시스템을 부팅합니다.

8.

다른 경로를 머신에 표시합니다.

검증 단계

-

스왑 장치가 다중 경로 장치에 있는지 확인합니다.

```
# swapon -s
```

예를 들어 다음과 같습니다.

```
# swapon -s
```

```
Filename          Type      Size Used  Priority
/dev/dm-3        partition 4169724 0    -2
```

파일 이름은 다중 경로 스왑 장치와 일치해야 합니다.

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

7.4. DMSETUP 명령으로 장치 매핑 항목 확인

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

절차

- 모든 장치 매핑 장치와 주요 및 마이너 번호를 표시합니다. 부 번호는 **dm** 장치의 이름을 결정합니다. 예를 들어, **3**의 부 수는 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. MULTIPATHD 데몬 관리

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

절차

- `multipathd show map` 명령의 출력에 대한 표준 기본 형식을 확인합니다.

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

- 일부 다중 경로 명령에는 형식 옵션 뒤에 와일드카드가 포함됩니다. 다음 명령을 사용하여 사용 가능한 와일드카드 목록을 표시합니다.

```
# multipathd show wildcards
```

- 다중 경로 와일드카드가 있는 형식 문자열을 일반 및 원시 형식으로 사용하여 다중 경로에서 모니터링 중인 다중 경로 장치를 표시합니다.

```
list|show maps|multipaths format $format
list|show maps|multipaths raw format $format
```

multipathd 명령은 다중 경로 장치 및 경로의 상태를 "raw" 형식 버전으로 표시하는 형식 명령을 지원합니다. 원시 형식으로는 헤더가 출력되지 않으며 열과 헤더를 맞추기 위해 필드가 채워지지 않습니다. 대신, 필드가 형식 문자열에 지정된 대로 정확하게 출력됩니다. 그러면 이 출력을 스크립팅에 더 쉽게 사용할 수 있습니다. **multipathd show wildcards** 명령을 사용하여 형식 문자열에 사용된 와일드카드를 표시할 수 있습니다.

- 다중 경로 와일드카드가 있는 형식 문자열을 일반 및 원시 형식으로 사용하여 다중 경로 모니터링 중인 경로를 표시합니다.

```
list|show paths format $format
list|show paths raw format $format
```

- 다중 경로 표시 맵의 비raw 형식과 원시 형식 간의 차이를 표시합니다. 원시 형식에는 헤더가 없으며 열 사이에 단일 공간만 있습니다.

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

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

추가 리소스

- [multipathd\(8\) 도움말 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 메모리 부하가 증가하여 사용 가능한 시스템 메모리가 충분한지 확인해야 합니다. 다음 명령을 사용하여 시스템의 현재 메모리 로드 및 사용 가능한 메모리를 확인합니다.

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

- 시스템에서 스토리지 장치를 제거하는 것을 권장하지 않습니다.
 - 사용 가능한 메모리는 **100개 샘플당 총 메모리의 5% 미만**입니다.
 - 스왑이 활성화되어 있습니다(**0**이 아닌 **si** 및 **vmstat** 명령 출력의 열).

8.1. 스토리지 장치의 안전한 제거

실행 중인 시스템에서 스토리지 장치를 안전하게 제거하려면 최상위 접근 방식이 필요합니다. 일반적으로 애플리케이션 또는 파일 시스템인 최상위 계층에서 시작하여 물리적 장치인 하단 계층으로 작업합니다.

스토리지 장치를 여러 가지 방법으로 사용할 수 있으며 물리적 장치 위에 다양한 가상 구성이 있을 수 있습니다. 예를 들어 장치의 여러 인스턴스를 다중 경로 장치로 그룹화하거나 **RAID**의 일부로 만들거나 **LVM** 그룹의 일부로 만들 수 있습니다. 또한 장치는 파일 시스템을 통해 액세스하거나 "**raw**" 장치와 같이 직접 액세스할 수 있습니다.

최상위 보안 접근 방식을 사용하는 동안 다음 조건을 충족해야 합니다.

- 삭제하려는 장치가 사용되지 않습니다.

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

8.2. 블록 장치 제거

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



주의

다음에 설명된 절차 없이 **SCSI** 버스를 다시 스캔하거나 운영 체제 상태를 변경하는 다른 작업을 수행하면 I/O 시간 초과, 장치가 예기치 않게 제거되거나 데이터 손실로 인한 지연이 발생할 수 있습니다.

사전 요구 사항

- 다중 경로 장치를 제거하려면 경로 장치에 액세스할 수 없는 경우 다중 경로 장치의 대기열을 비활성화합니다.

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

그러면 장치의 I/O가 실패할 수 있으므로 장치를 사용하는 애플리케이션이 종료될 수 있습니다.

- 제거하려는 장치를 사용하는 다른 애플리케이션 또는 서비스가 없는지 확인합니다.
- 제거할 장치에서 데이터를 백업했는지 확인합니다.

절차

1. **ArgoCD** 명령을 사용하여 장치에 마운트된 모든 파일 시스템을 마운트 해제합니다.

2.

모든 **MD RAID** 배열 또는 해당 장치가 속하는 **LVM** 볼륨에서 장치를 제거합니다. 장치 유형에 따라 다음 단계 중 하나를 실행합니다.

•

장치가 **LVM** 그룹의 멤버이고 다중 경로 장치인 경우 다음을 수행합니다.

a.

데이터를 다른 장치로 이동합니다.

```
# pvmove -b /dev/mapper/from-multipath-device /dev/mapper/to-multipath-device
```

b.

볼륨 그룹에서 장치를 제거합니다.

```
# vgreduce volume-group /dev/mapper/from-multipath-device
```

c.

선택 사항: 물리 장치에서 **LVM** 메타데이터를 제거합니다.

```
# pvremove /dev/mapper/from-multipath-device
```

•

다중 경로 장치를 제거하는 경우 다음 명령을 실행합니다.

a.

장치에 대한 모든 경로를 표시합니다.

```
# multipath -l
```

이 명령의 출력은 이후 단계에서 필요합니다.

b.

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/O**를 개별 경로로 전달한 경우 사용되는 모든 장치 경로로 미해결 **I/O**를 플러시합니다.

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

umount 또는 **vgreduce** 명령이 I/O를 플러시하지 않는 위치에서 직접 액세스하는 장치에 중요합니다.

-

SCSI 장치를 제거하는 경우 다음 명령을 실행합니다.

- a.

시스템의 애플리케이션, 스크립트 또는 유틸리티에서 **/dev/sd**, **/dev/disk/by-path** 또는 **major:minor** 번호와 같은 장치의 경로 기반 이름에 대한 참조를 제거합니다. 이렇게 하면 향후 추가되는 다른 장치가 현재 장치에 대해 실수를 하지 않습니다.

- b.

SCSI 하위 시스템에서 장치에 대한 각 경로를 제거합니다.

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

여기서 **device-name** 은 이전에 다중 경로 장치로 사용된 경우 **multipath -l** 명령의 출력에서 검색됩니다.

- 3.

실행 중인 시스템에서 물리적 장치를 제거합니다. 이 장치를 제거해도 다른 장치의 I/O는 중지되지 않습니다.

추가 리소스

-

multipath(8), **pv move(8)**, **arrow reduce(8)**, **blockdev(8)** 및 **ArgoCD (8)** 도움말 페이지.

9장. DM MULTIPATH 문제 해결

다중 경로 구성을 구현하는 데 문제가 있는 경우 다양한 문제를 확인할 수 있습니다. 다음 문제로 인해 느린 또는 작동하지 않는 다중 경로 구성이 발생할 수 있습니다.

다중 경로 데몬이 실행되고 있지 않습니다

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

queue_if_no_path 기능 관련 문제

"1 queue_if_no_path" 옵션을 사용하여 다중 경로 장치를 구성하면 하나 이상의 경로가 복원될 때까지 I/O가 중단되는 모든 프로세스가 중단됩니다.

9.1. QUEUE_IF_NO_PATH 기능 문제 해결

"1 queue_if_no_path" 옵션을 사용하여 다중 경로 장치를 구성하면 하나 이상의 경로가 복원될 때까지 I/O가 중단되는 모든 프로세스가 중단됩니다. 이를 방지하려면 **/etc/multipath.conf** 파일의 **no_path_retry N** 매개 변수를 설정합니다. 여기서 **N**은 시스템이 경로를 다시 시도해야 하는 횟수입니다.

"1 queue_if_no_path" 옵션을 사용해야 하며 여기에 언급된 문제가 발생하는 경우 모든 경로를 사용할 수 없는 특정 LUN에 대해 런타임 시 큐링 정책을 비활성화할 수 있습니다.

절차

- 특정 장치에 대한 큐링을 비활성화합니다.


```
# multipathd disablequeueing map device
```
- 모든 장치에 대한 큐링을 비활성화합니다.


```
# multipathd disablequeueing maps
```

장치의 대기열을 비활성화한 후에는 **multipathd** 를 다시 시작하거나 다시 로드하거나 다음 명령 중 하나를 실행할 때까지 계속 비활성화됩니다.

- 특정 장치의 이전 값으로 큐링을 재설정합니다.

```
# multipathd restorequeueing map device
```

- 모든 장치에 대한 이전 값으로 큐링을 재설정합니다.

```
# multipathd restorequeueing maps
```

9.2. MULTIPATHD 대화식 콘솔 문제 해결

multipathd -k 명령은 다중 경로 데몬에 대한 대화형 인터페이스입니다. 이 명령을 입력하면 대화형 다중 경로 콘솔이 나타납니다. 이 명령을 실행한 후에는 도움말 을 입력하여 사용 가능한 명령 목록과 **Ctrl+D** 종료하도록 입력할 수 있습니다.

다중 경로 대화형 콘솔을 사용하여 시스템에서 가질 수 있는 문제를 해결합니다.

절차

- 콘솔을 종료하기 전에 기본값을 포함하여 다중 경로 구성을 표시합니다.

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

- **multipath.conf** 파일의 변경 사항을 선택했는지 확인합니다.

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

- 경로 검사기가 제대로 작동하는지 확인합니다.

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

- 대화형 콘솔을 시작하지 않고 명령줄에서 직접 단일 다중 경로 대화형 명령을 실행할 수도 있습니다. 예를 들어 **multipath.conf** 파일의 변경 사항을 선택했는지 확인하려면 다음을 실행합니

다.

```
█ # multipathd reconfigure
```

10장. EH_DEADLINE을 사용하여 스토리지 오류 복구를 위한 최대 시간 구성

실패한 **SCSI** 장치를 복구하는 데 허용되는 최대 시간을 구성할 수 있습니다. 이 구성은 스토리지 하드웨어가 실패로 인해 응답하지 않게 되는 경우에도 **I/O** 응답 시간을 보장합니다.

10.1. EH_DEADLINE 매개변수

SCSI 오류 처리(**EH**) 메커니즘은 실패한 **SCSI** 장치에서 오류 복구를 시도합니다. **SCSI** 호스트 오브젝트 **eh_deadline** 매개변수를 사용하면 복구에 대한 최대 시간을 구성할 수 있습니다. 구성된 시간이 만료되면 **SCSI EH**는 전체 **HBA**(호스트 버스 어댑터)를 중지하고 재설정합니다.

eh_deadline 을 사용하면 시간을 줄일 수 있습니다.

- 오류가 발생한 경로를 종료하려면 다음을 수행합니다.
- 경로를 전환하거나
- **RAID** 슬라이스를 비활성화하려면 다음을 수행합니다.



주의

eh_deadline 이 만료되면 **SCSI EH**는 **HBA**를 재설정하여 실패한 경로뿐만 아니라 해당 **HBA**의 모든 대상 경로에 영향을 미칩니다. 중복 경로 중 일부를 다른 이유로 사용할 수 없는 경우 **I/O** 오류가 발생할 수 있습니다. 모든 대상에 완전히 중복된 다중 경로 구성이 있는 경우에만 **eh_deadline** 을 활성화합니다.

eh_deadline 매개변수의 값은 초 단위로 지정됩니다. 기본 설정은 **off** 로, 시간 제한을 비활성화하고 모든 오류 복구를 허용합니다.

eh_deadline이 유용한 시나리오

대부분의 시나리오에서는 **eh_deadline** 을 활성화할 필요가 없습니다. **eh_deadline** 사용은 특정 시나리오에서 유용할 수 있습니다. 예를 들어 파이버 채널(**FC**) 스위치와 대상 포트 간에 링크 손실이 발생하고 **HBA**에 등록된 상태 변경 알림(**RSCN**)이 수신되지 않습니다. 이러한 경우 오류가 발생하는 대신 **I/O** 요청

및 오류 복구 명령이 항상 시간 초과됩니다. 이 환경에서 **eh_deadline** 을 설정하면 복구 시간에 상한이 적용됩니다. **DM Multipath**를 통해 사용 가능한 다른 경로에서 실패한 I/O를 재시도할 수 있습니다.

다음 조건에서 **eh_deadline** 매개 변수는 I/O 및 오류 복구 명령이 즉시 실패하므로 추가 이점을 제공하지 않으므로 **DM Multipath**를 다시 시도할 수 있습니다.

- **RSCN**이 활성화된 경우
- **HBA**가 링크를 등록할 수 없는 경우

10.2. EH_DEADLINE 매개변수 설정

이 절차에서는 최대 **SCSI** 복구 시간을 제한하도록 **eh_deadline** 매개 변수 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** 매개 변수를 필요한 초 단위로 설정합니다.

```
# eh_deadline 300
```



참고

RHEL 8.4에서 **multipath.conf** 파일의 **defaults** 섹션을 사용하여 **eh_deadline** 매개 변수를 설정하는 것이 기본 방법입니다.

이 메시지를 사용하여 **eh_deadline** 매개 변수를 해제하려면 **eh_deadline** 을 **off** 로 설정합니다.

- **sysfs**

`/sys/class/scsi_host/host<host-number>/eh_deadline` 파일에 초 수를 작성합니다. 예를 들어 SCSI 호스트 6에서 `sysfs` 를 통해 `eh_deadline` 매개 변수를 설정하려면 다음을 수행합니다.

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

이 메시지를 사용하여 `eh_deadline` 매개 변수를 해제하려면 `echo off` 를 사용합니다.

○

커널 매개변수

`scsi_mod._deadline` 커널 매개 변수를 사용하여 모든 SCSI HBA의 기본값을 설정합니다.

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

이 메시지를 사용하여 `eh_deadline` 매개 변수를 해제하려면 `echo -1` 을 사용합니다.

추가 리소스

●

[udev](#) 규칙을 사용하여 `eh_deadline` 및 `eh_timeout`을 영구적으로 설정하는 방법