



# Red Hat Enterprise Linux 9

## Device Mapper Multipath の設定

Device Mapper Multipath 機能の使用



# Red Hat Enterprise Linux 9 Device Mapper Multipath の設定

---

## Device Mapper Multipath 機能の使用

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.

## 法律上の通知

Copyright © 2022 | 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<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.

## 概要

本書は、Red Hat Enterprise Linux 9 で Device Mapper Multipath (DM-Multipath) 機能を設定および管理する方法を説明します。

## 目次

多様性を受け入れるオープンソースの強化 .....	4
RED HAT ドキュメントへのフィードバックの提供 .....	5
<b>第1章 デバイスマAPPERマルチパスの概要 .....</b>	<b>6</b>
1.1. 1つの RAID デバイスを使用したアクティブ/パッシブのマルチパス構成	6
1.2. 2つの RAID デバイスを使用したアクティブ/パッシブのマルチパス構成	7
1.3. 1つの RAID デバイスを使用したアクティブ/アクティブのマルチパス構成	8
1.4. DM MULTIPATH コンポーネント	9
1.5. MULTIPATH コマンド	10
1.6. MULTIPATH コマンドの出力	11
1.7. マルチパス設定の表示	12
1.8. 関連情報	13
<b>第2章 マルチパスデバイス .....</b>	<b>14</b>
2.1. マルチパスデバイス識別子	14
2.2. 論理ボリューム内のマルチパスデバイス	15
<b>第3章 DM MULTIPATH の設定 .....</b>	<b>17</b>
3.1. DEVICE-MAPPER-MULTIPATH パッケージの確認	17
3.2. 基本的なフェイルオーバー設定用の DM MULTIPATH のセットアップ	17
3.3. マルチパスデバイスの作成時にローカルのディスクを無視	18
3.4. 追加のストレージデバイスの設定	19
3.5. INITRAMFS ファイルシステムでマルチパスの設定	20
<b>第4章 NVME デバイスでのマルチパスの有効化 .....</b>	<b>21</b>
4.1. ネイティブ NVME マルチパスと DM MULTIPATH	21
4.2. ネイティブ NVME マルチパスの実現	21
4.3. NVME デバイスでの DM MULTIPATH の有効化	23
<b>第5章 DM MULTIPATH 設定ファイルの編集 .....</b>	<b>27</b>
5.1. 設定ファイルの概要	27
5.2. DM MULTIPATH がデバイスタイムアウトの上書き	28
5.3. マルチパス設定ファイルのデフォルトの編集	28
5.4. 特定デバイスのマルチパス設定の編集	29
5.5. プロトコルを使用した特定デバイスのマルチパス設定の変更	30
5.6. ストレージコントローラーのマルチパス設定の編集	32
5.7. すべてのデバイスへのマルチパス値の設定	33
<b>第6章 デバイスのマルチパスの防止 .....</b>	<b>35</b>
6.1. DM MULTIPATH がパスのマルチパスデバイスを作成する際の条件	35
6.2. 特定のデバイスでマルチパスを無効にする基準	36
6.3. WWID によるマルチパスの無効化	37
6.4. デバイス名によるマルチパスの無効化	37
6.5. デバイスの種別によるマルチパスの無効化	38
6.6. UDEV プロパティによるマルチパスの無効化	39
6.7. デバイスプロトコルによるマルチパスの無効化	40
6.8. マルチパスを無効にしたデバイスに対する例外の追加	41
<b>第7章 マルチパス化されたボリュームの管理 .....</b>	<b>43</b>
7.1. オンラインのマルチパスデバイスのサイズ変更	43
7.2. ROOT ファイルシステムをシングルパスデバイスからマルチパスデバイスへ移動	43
7.3. SWAP ファイルシステムをシングルパスデバイスからマルチパスデバイスへ移動	45

---

7.4. DMSETUP コマンドでデバイスマッパーエントリーの特定	46
7.5. MULTIPATHD デーモンの管理	47
<b>第8章 ストレージデバイスの削除</b> .....	<b>49</b>
8.1. ストレージデバイスの安全な削除	49
8.2. ブロックデバイスの削除	49
<b>第9章 DM MULTIPATH のトラブルシューティング</b> .....	<b>52</b>
9.1. QUEUE_IF_NO_PATH 機能に関する問題のトラブルシューティング	52
9.2. MULTIPATHD 対話式コンソールでトラブルシューティング	52
<b>第10章 EH_DEADLINE を使用したストレージエラーからの回復における最大時間の設定</b> .....	<b>54</b>
10.1. EH_DEADLINE パラメーター	54
eh_deadline が便利なシナリオ	54
10.2. EH_DEADLINE パラメーターの設定	54



## 多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[弊社の CTO、Chris Wright のメッセージ](#) を参照してください。



## RED HAT ドキュメントへのフィードバックの提供

ご意見ご要望をお聞かせください。ドキュメントの改善点はございませんか。

- 特定の部分についての簡単なコメントをお寄せいただく場合は、以下をご確認ください。
  1. ドキュメントの表示が **Multi-page HTML** 形式になっていて、ドキュメントの右上隅に **Feedback** ボタンがあることを確認してください。
  2. マウスカーソルで、コメントを追加する部分を強調表示します。
  3. そのテキストの下に表示される **Add Feedback** ポップアップをクリックします。
  4. 表示される手順に従ってください。
- Bugzilla を介してフィードバックを送信するには、新しいチケットを作成します。
  1. [Bugzilla](#) の Web サイトに移動します。
  2. Component で **Documentation** を選択します。
  3. **Description** フィールドに、ドキュメントの改善に関するご意見を記入してください。ドキュメントの該当部分へのリンクも記入してください。
  4. **Submit Bug** をクリックします。

## 第1章 デバイスマッパーマルチパスの概要

Device Mapper のマルチパス (DM Multipath) を使用すると、サーバーノードとストレージレイとの間の複数の I/O パスを 1 つのデバイスに設定できます。これらの I/O パスは、個別のケーブル、スイッチ、コントローラーを含むことができる物理的なストレージエリアネットワーク (SAN) 接続です。マルチパスは I/O パスを集約し、集約されたパスで構成される新しいデバイスを作成します。

DM Multipath は以下を提供します。

### 冗長性

DM Multipath は、アクティブ/パッシブ構成でフェイルオーバーを提供できます。アクティブ/パッシブ構成では、パスは、I/O には常に半分しか使用されません。ケーブル、スイッチ、コントローラーなどの I/O パスの要素に障害が発生した場合、DM Multipath は代替パスに切り替わります。

### パフォーマンスの向上

DM Multipath は、アクティブ/アクティブモードで構成できます。このモードでは、I/O はラウンドロビン方式でパスに分散されます。一部の構成では、DM Multipath は I/O パスの負荷を検出し、負荷を動的に再調整できます。

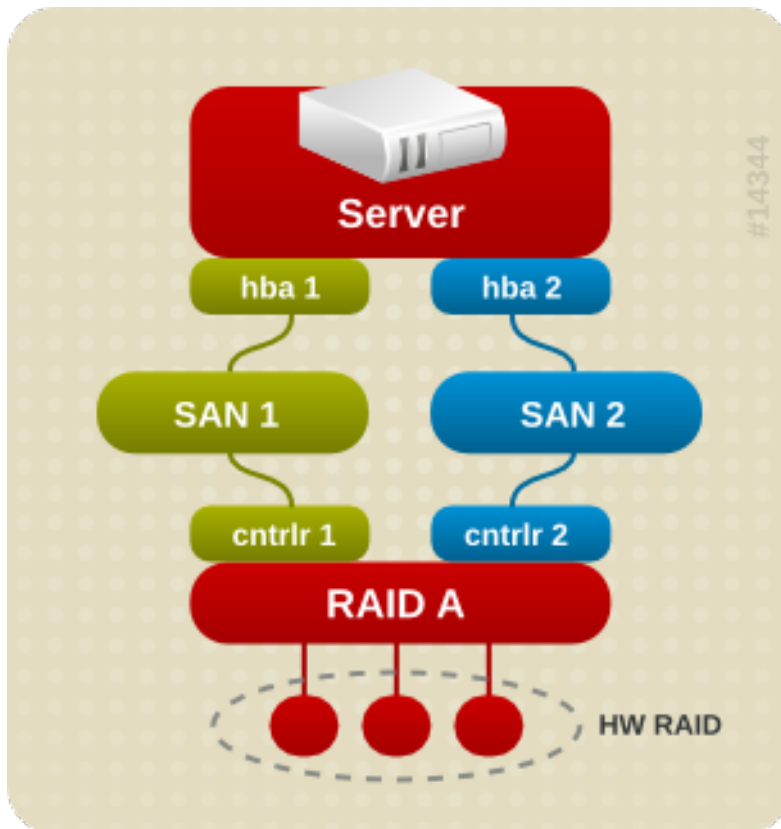
### 1.1.1 つの RAID デバイスを使用したアクティブ/パッシブのマルチパス構成

この構成では、サーバー上に 2 つのホストバスアダプター (HBA)、2 つの SAN スイッチ、および 2 つの RAID コントローラーがあります。この構成では、次のような障害が発生する可能性があります。

- HBA の障害
- ファイバーチャネルケーブルの障害
- SAN スイッチの障害
- アレイコントローラーポートの障害

DM Multipath を設定すると、これらのポイントのいずれかで障害が発生すると、DM Multipath は代替の I/O パスに切り替えます。以下の図は、サーバーから RAID デバイスへの 2 つの I/O パスを使用した設定を説明します。ここでは、**hba1**、**SAN1**、および **cntrlr1** を通る 1 つの I/O パスと、**hba2**、**SAN2**、および **cntrlr2** を通る別の I/O パスがあります。

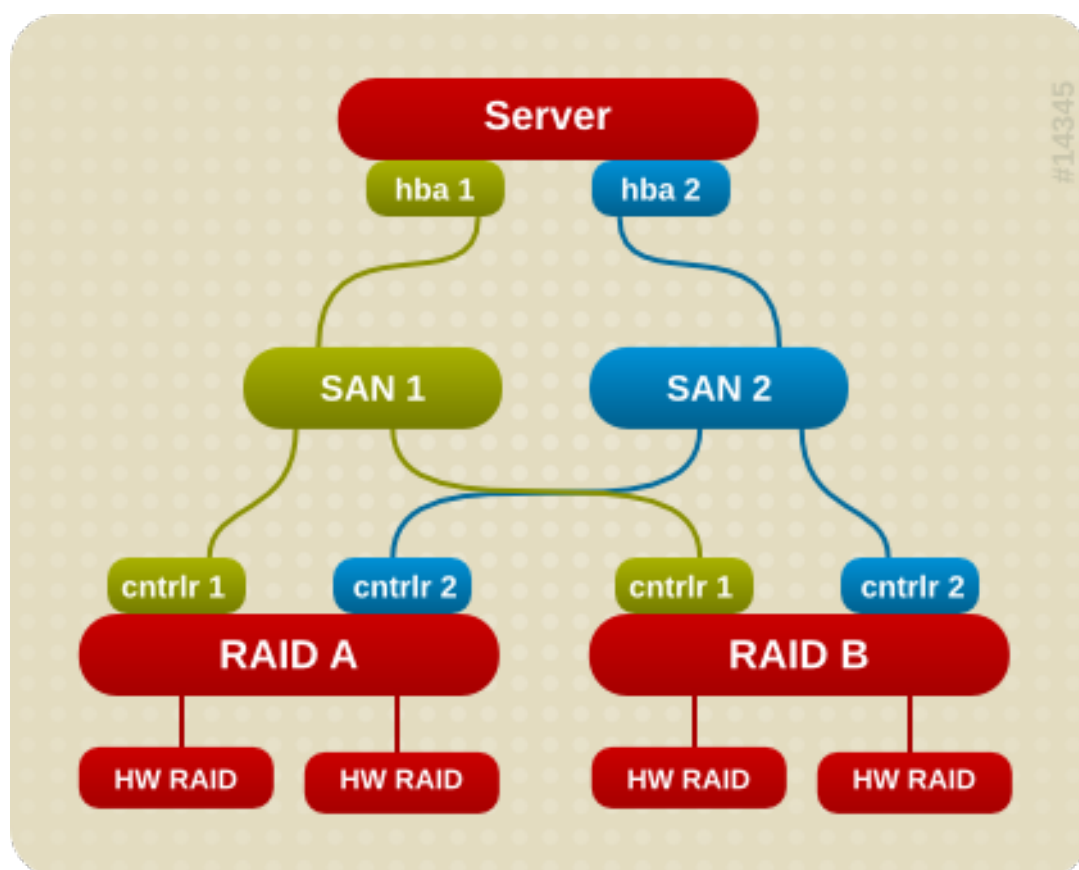
図1.11つの RAID デバイスを使用したアクティブ/パッシブのマルチパス構成



## 1.2. 2つの RAID デバイスを使用したアクティブ/パッシブのマルチパス構成

この構成では、サーバー上に2つのHBA、2つのSANスイッチ、およびそれぞれ2つのRAIDコントローラーを備えた2つのRAIDデバイスがあります。DM Multipathを設定すると、いずれかのRAIDデバイスへのI/Oパスのいずれかのポイントで障害が発生すると、DM Multipathはそのデバイスの代替I/Oパスに切り替えます。以下の図は、各RAIDデバイスへの2つのI/Oパスを使用した設定を説明します。ここでは、各RAIDデバイスへの2つのI/Oパスがあります。

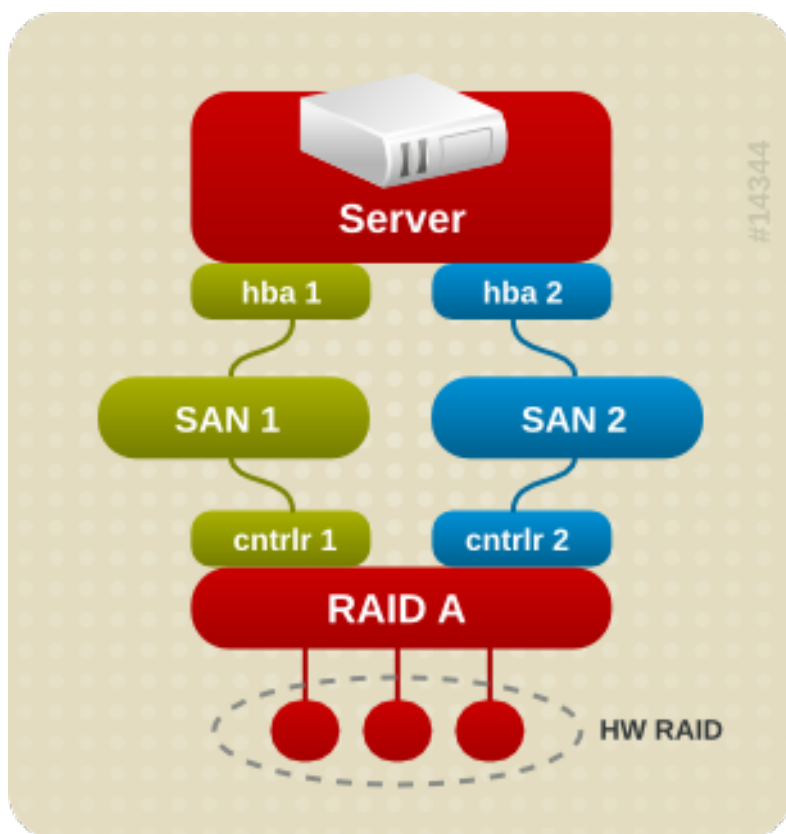
図1.2 2つの RAID デバイスを使用したアクティブ/パッシブのマルチパス構成



### 1.3.1つの RAID デバイスを使用したアクティブ/アクティブのマルチパス構成

この設定では、サーバー上に HBA、2つの SAN スイッチ、および 2つの RAID コントローラーがあります。以下の図は、サーバーからストレージデバイスへの 2つの I/O パスを使用した設定について説明します。ここで、I/O は、これら 2つのパスに分散できます。

図1.3 1つの RAID デバイスを使用したアクティブ/アクティブのマルチパス構成



## 1.4. DM MULTIPATH コンポーネント

以下の表は、DM Multipath コンポーネントを示しています。

表1.1 DM Multipath のコンポーネント

コンポーネント	説明
<b>dm_multipath</b> カーネルモジュール	I/O を再ルーティングし、パスとパスグループのフェールオーバーに対応します。
<b>mpathconf</b> ユーティリティ	デバイスマッパーマルチパスを構成して有効にします。
<b>multipath</b> コマンド	マルチパスデバイスを一覧表示して構成します。これは、ブロックデバイスが追加されるたびに <b>udev</b> により実行され、デバイスがマルチパスデバイスの一部であるかどうかを判断します。
<b>multipathd</b> デモン	マルチパスデバイスを自動的に作成および削除し、パスを監視します。パスが失敗して戻ってくると、マルチパスデバイスが更新される場合があります。マルチパスデバイスへのインタラクティブな変更を許可します。 <b>/etc/multipath.conf</b> ファイルに変更がある場合は、サービスを再読み込みします。

<p><b>kpartx</b> コマンド</p>	<p>デバイス上のパーティションのデバイスマッパーデバイスを作成します。このコマンドは、マルチパスデバイスが作成され、その上にパーティションデバイスが作成されると、<b>udev</b> により自動的に実行されます。<b>kpartx</b> コマンドは独自のパッケージで提供されますが、<b>device-mapper-multipath</b> パッケージはそれに依存しています。</p>
<p><b>mpathpersist</b></p>	<p>マルチパスデバイスに <b>SCSI-3</b> 永続予約を設定します。このコマンドは、<b>sg_persist</b> が SCSI デバイスに対して行う方法と似ていますが、マルチパスデバイスのすべてのパスで永続的な予約の設定を処理します。<b>multipathd</b> と調整して、後で追加されるパスに予約が正しく設定されるようにします。この機能を使用するには、<b>reservation_key</b> 属性を <b>/etc/multipath.conf</b> ファイルで定義する必要があります。定義しないと、<b>multipathd</b> デーモンは、新しく検出されたパスまたは復元されたパスの永続的な予約を確認しません。</p>

## 1.5. MULTIPATH コマンド

**multipath** コマンドは、デバイスの複数のパスを検出および結合するために使用されます。マルチパス化されたデバイスの管理に使用できるさまざまなオプションを提供します。

次の表に、役立つと思われる **multipath** コマンドのいくつかのオプションを示します。

表1.2 便利な **multipath** コマンドのオプション

オプション	説明
-l	<b>sysfs</b> およびデバイスマッパーから取得した現在のマルチパス設定を表示します。
-ll	<b>sysfs</b> およびデバイスマッパーと、システムで利用可能なその他の全コンポーネントから取得した現在のマルチパス設定を表示します。
-f device	指定したマルチパスデバイスを削除します。
-F	不要なマルチパスデバイスをすべて削除します。
-w device	<b>wwids</b> ファイルから指定デバイスの <b>wwid</b> を削除します。
-W	現在のマルチパスデバイスのみが含まれるように、 <b>wwids</b> ファイルをリセットします。

## 1.6. MULTIPATH コマンドの出力

マルチパスデバイスの作成、修正、一覧表示を行うと、現在のデバイス設定が表示されます。形式を以下に示します。

- マルチパスデバイス

```
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
```

- パスグループ

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

- パス

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

マルチパスコマンドの出力例を以下に示します。

```
3600d023000000000000e13955cc3757800 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
```

パスが有効になり、I/O が可能になると、パスのステータスが **ready** または **ghost** になります。パスが停止している場合は、ステータスが **faulty** または **shaky** になります。パスの状態は、`/etc/multipath.conf` ファイルに定義されているポーリング間隔に応じ、`multipathd` デーモンによって定期的に更新されます。

その他の可能なパスのステータス値は以下のとおりです。

- **i/o pending** - チェッカーがアクティブにこのパスを確認し、ステータスはまもなく更新されません。
- **i/o timeout - faulty** と同じです。タイムアウトする前にチェッカーが成功または失敗のいずれかを返さなかったことをユーザーに通知します。
- **removed** - システムからパスが削除され、マルチパスデバイスからまもなく削除されます。**faulty** と同様に処理されます。
- **wild** - 内部エラーまたは設定の問題によって、`multipathd` はパスチェッカーを実行できませんでした。マルチパスがパス上の多くのアクションをスキップすることを除いて、**faulty** とほぼ同じです。
- **unchecked** - パスが今検出された、割り当てられたパスチェッカーがない、パスチェッカーにエラーが発生したなどの理由で、このパスでパスチェッカーが実行していません。これは **wild** と同じように処理されます。
- **delayed** - パスチェッカーはパスが有効であると返すにも関わらず、パスが複数回失敗し、パスを遅延するようにマルチパスが設定されているため、マルチパスがパスの復元を遅らせています。

カーネルに関しては、dm ステータスはパスステータスに似ています。dm のステータスが **active** の場合は、パスのステータスが **ready** および **ghost** になります。**pending** パスのステータスと同等の dm ステータスはありません。その他のパスのステータスはすべて、dm のステータスでは **failed** となります。dm のステータスは、パスチェッカーが完了するまで現在のステータスを保持します。

**online\_status** に使用できる値は **running** と **offline** です。**offline** ステータスは、この SCSI デバイスが無効になっていることを表します。



### 注記

マルチパスデバイスを作成または変更すると、マルチパスはデバイス設定を出力します。ただし、書き込み権限などの一部の機能や、その他の機能情報は不明な場合があります。出力と、作成または変更中に選択した機能との間に違いがある場合があります。これは通常の動作です。正しい状態を表示するには、作成後にデバイスを一覧表示してください。

## 1.7. マルチパス設定の表示

**-l** および **multipath** コマンドを使用すると、現在のマルチパス設定を表示できます。**-l** オプションは、**sysfs** の情報から収集したマルチパストポロジと、デバイスマッパーを表示します。**-ll** オプションは、**-l** オプションが表示する情報のほかに、使用できる他のすべてのシステムコンポーネントを表示します。

マルチパス設定を表示するとき、**multipath** コマンドの **-v** オプションを使用すると詳細レベルを指定できます。**-v0** を指定すると何も出力されません。**-v1** を指定すると、作成または更新されたマルチパス名のみが出力され、**kpartx** などの別のツールに提供できます。**-v2** を指定すると、検出されたパス、マルチパス、およびデバイスマップがすべて出力されます。より詳細な情報を出力する場合は、**-v3**、**-v4**、または **-v5** を指定することもできます。

**multipath -l** コマンドの出力を以下に示します。

```
# 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
```

**multipath -ll** コマンドの出力を以下に示します。

```
# 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. 関連情報

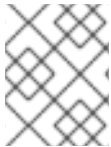
- man ページの **multipath(8)** および **multipathd(8)**
- **/etc/multipath.conf** ファイル

## 第2章 マルチパスデバイス

DM Multipath は、基礎となるデバイスの上に1つのマルチパスデバイスを作成することにより、I/O パスを論理的に整理する方法を提供します。DM Multipath を使用しない場合は、I/O パスが同じサーバーノードを同じストレージコントローラーに接続している場合でも、システムはサーバーノードからストレージコントローラーへの各パスを個別のデバイスとして扱います。

### 2.1. マルチパスデバイス識別子

新しいデバイスが DM Multipath の制御下にある場合に、これらのデバイスは `/dev/mapper/` ディレクトリーおよび `/dev/` ディレクトリーに作成されます。



#### 注記

`/dev/dm-X` という形式のデバイスは内部使用専用であるため、管理者が直接使用するものではありません。

以下は、マルチパスデバイス名について説明しています。

- **user\_friendly\_names** 構成オプションが **no** に設定されている場合は、マルチパスデバイスの名前が World Wide Identifier (WWID) に設定されます。デフォルトでは、マルチパスデバイスの名前は WWID に設定されます。デバイス名は `/dev/mapper/WWID` になります。また、`/dev/dm-X` という名前の `/dev/` ディレクトリーにも作成されます。
- または、`/etc/multipath.conf` ファイルで、**user\_friendly\_names** オプションを **yes** に設定できます。これにより、**multipath** セクションの **alias** が、**mpathN** 形式のノード固有の名前に設定されます。デバイス名は、`/dev/mapper/mpathN` および `/dev/dm-X` になります。ただし、マルチパスデバイスを使用するすべてのノードでデバイス名が同じであるとは限りません。同様に、`/etc/multipath.conf` ファイルで **alias** オプションを設定した場合は、クラスター内のすべてのノードで自動的に名前が一致しません。



#### 注記

LVM を使用してマルチパスデバイスから論理デバイスを作成する場合は、これにより問題が発生することはありません。すべてのノードでマルチパスデバイス名の一貫性を保つために、Red Hat は、**user\_friendly\_names** オプションを無効にすることを推奨します。

たとえば、ゾーンに分けられていない1つの FC スイッチにより、2つのポートを持つストレージコントローラーに接続された2つの HBA を持つノードは、`/dev/sda`、`/dev/sdb`、`/dev/sdc`、および `/dev/sdd` の4つのデバイスを認識します。DM Multipath は、マルチパス構成に従って、I/O を基本となるこれらの4つのデバイスにルーティングしなおす一意の WWID を持つシングルデバイスを作成します。

**user\_friendly\_names** オプションおよび **alias** オプションの他に、マルチパスデバイスには他の属性もあります。`/etc/multipath.conf` ファイルの **multipaths** セクションに、デバイスのエントリーを作成することにより、特定のマルチパスデバイスのこれらの属性を変更できます。

#### 関連情報

- man ページの **multipath(8)** および **multipath.conf(8)**
- `/etc/multipath.conf` ファイル

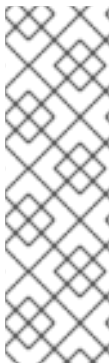
- DM Multipath コンポーネント

## 2.2. 論理ボリューム内のマルチパスデバイス

マルチパスデバイスを作成したら、論理ボリュームマネージャー(LVM)物理ボリュームを作成する際に物理デバイス名を使用するのと同様に、マルチパスデバイス名を使用できます。たとえば、`/dev/mapper/mpatha` がマルチパスデバイスの名前である場合、`pvcreate /dev/mapper/mpatha` コマンドは、`/dev/mapper/mpatha` を物理ボリュームとしてマークします。

他の LVM 物理デバイスを使用するのと同じように、LVM ボリュームグループを作成するときに、作成された LVM 物理デバイスを使用できます。

`/etc/lvm/lvm.conf` ファイル内のすべての `sd` デバイスをフィルタリングするには、そのファイルの `devices` セクションに `filter = [ "r/block/", "r/disk/", "r/sd./", "a./" ]` を追加します。



### 注記

パーティションを設定したデバイス全体に LVM 物理ボリュームを作成しようとする、`pvcreate` コマンドは失敗します。Anaconda および Kickstart のインストールプログラムは、すべてのブロックデバイスに特に指定しない限り、空のパーティションテーブルを作成します。パーティションを作成する代わりにデバイス全体を使用する場合は、デバイスから既存のパーティションを削除します。`kpartx -d` デバイスコマンドと `fdisk` ユーティリティーを使用して、既存のパーティションを削除できます。システムに 2Tb を超えるブロックデバイスがある場合は、`parted` ユーティリティーを使用してパーティションを削除します。

**active/passive** マルチパスアレイを基礎となる物理デバイスとして使用する LVM 論理ボリュームを作成する場合は、必要に応じて、`/etc/lvm/lvm.conf` ファイルにフィルターを追加して、マルチパスデバイスの基礎となるディスクを除外できます。これは、I/O の受信時にアレイがアクティブパスをパッシブパスに自動的に変更する場合に、このようなデバイスにフィルターが設定されていないと、LVM がパッシブパスをスキャンするたびにマルチパスがフェールオーバーおよびフェールバックするためです。

カーネルは、使用する正しいハードウェアハンドラーを自動的に検出してアクティブ/パッシブ状態を変更します。状態を変更するために介入を必要とするアクティブ/パッシブパスの場合、マルチパスは自動的にこのハードウェアハンドラーを使用して、必要に応じて介入します。カーネルが使用する正しいハードウェアハンドラーを自動的に検出しない場合は、`multipath.conf` ファイルで `"hardware_handler"` オプションを使用して、使用するハードウェアハンドラーを設定できます。パッシブパスをアクティブにするコマンドを必要とする **アクティブ/パッシブ** アレイでこの問題が発生すると、LVM が警告メッセージを出力します。

設定によっては、LVM が以下のいずれかのメッセージを出力することがあります。

- LUN の準備ができていません:

```
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
```

- 読み取りに失敗しました:

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

以下は、上記のエラーの理由です。

- マルチパスは、マシンにアクティブ/パッシブパスを提供するストレージデバイスでは設定されません。
- パスはマルチパスデバイスではなく、直接アクセスします。

#### 関連情報

- [lvm.conf](#) の man ページ
- [DM Multipath コンポーネント](#)

## 第3章 DM MULTIPATH の設定

**mpathconf** ユーティリティーを使用して DM Multipath を設定できます。このユーティリティーは、次のシナリオに基づいて **/etc/multipath.conf** マルチパス設定ファイルを作成または編集します。

- **/etc/multipath.conf** ファイルが既に存在する場合は、**mpathconf** ユーティリティーでファイルを編集します。
- **/etc/multipath.conf** ファイルが存在しない場合は、**mpathconf** ユーティリティーにより **/etc/multipath.conf** ファイルが新たに作成されます。

### 3.1. DEVICE-MAPPER-MULTIPATH パッケージの確認

システムに DM-Multipath をセットアップする前に、システムが最新であり、**device-mapper-multipath** パッケージがインストールされていることを確認してください。

#### 手順

1. システムに **device-mapper-multipath** パッケージが含まれているか確認します。

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

システムにパッケージが含まれていない場合は、次のよう出力されます。

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

2. システムにパッケージが含まれていない場合は、次のコマンドを実行してパッケージをインストールします。

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

### 3.2. 基本的なフェイルオーバー設定用の DM MULTIPATH のセットアップ

**multipathd** デーモンを起動する前に **/etc/multipath.conf** ファイルを編集する必要がある場合は、以下の手順に従い基本的なフェイルオーバー設定用の DM Multipath をセットアップします。

#### 手順

1. マルチパス設定ファイルを有効にします。

```
# mpathconf --enable
```

2. 必要に応じて **/etc/multipath.conf** ファイルを編集します。DM-Multipath 用のデフォルト設定はシステムにコンパイルされているため、**/etc/multipath.conf** ファイルで明示的に指定する必要はありません。

**path\_grouping\_policy** のデフォルト値は **failover** に設定されているため、この例では **/etc/multipath.conf** ファイルを編集する必要はありません。

設定ファイル内で初めてのデフォルトセクションはシステムを設定するため、マルチパスデバイスの名前の形式は **/dev/mapper/mpathn;** になります。この設定がないと、デバイスの WWID がマルチパスデバイスの名前になります。ユーザーフレンドリー名を使用しない場合は、次のコマンドを実行します。

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

マルチパスデーモンを起動してからマルチパス設定ファイルの編集が必要になった場合は、**systemctl reload multipathd.service** コマンドを実行して変更を反映する必要があります。

- 必要に応じて設定ファイルを保存し、エディターを終了します。
- マルチパスデーモンを起動し、マルチパスデバイスを作成します。

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



### 注記

**device-mapper-multipath** パッケージを削除しても、**/etc/multipath.conf** ファイル、または **/etc/multipath** ディレクトリー内のファイルは削除されません。これは、そのディレクトリーに、現在リストされているファイル以外のものが含まれている可能性があるためです。このファイルは、**device-mapper-multipath** パッケージの後続のインストール時に手動で削除する必要があります。

## 3.3. マルチパスデバイスの作成時にローカルのディスクを無視

一部のマシンには内部ディスク用のローカル SCSI カードがあり、DM Multipath をこのようなデバイスで使用することは推奨されません。**find\_multipaths** 設定パラメーターを **on** に設定すると、このようなデバイスでマルチパスを無効にする必要はありません。

**find\_multipaths** 設定パラメーターを **on** に設定しない場合は、以下の手順に従って DM Multipath 設定ファイルを変更して、マルチパスの設定時にローカルのディスクを無視できます。

### 手順

- 内部ディスクであるディスクを確認します。以下の例では、**/dev/sda** が内部ディスクになります。
  - 既存のマルチパスデバイスを表示します。

```
# 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 Multipath が作成できるその他のマルチパスデバイスを表示します。

```
# 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
```

- /etc/multipath.conf** ファイルの **blacklist** セクションを編集して、このデバイスを追加します。

WWID 属性を使用してデバイスを特定します。 **devnode** タイプで **devnode** デバイスをブラックリストに指定することもできますが、 **/dev/sda** が再起動後に同じになる保証はないため安全な方法ではありません。

上記の例では、 **/dev/sda** デバイスの WWID は **SIBM-ESXSST336732LC\_\_\_\_F3ET0EP0Q000072428BX1** です。このデバイスをブラックリストに指定する場合は、 **/etc/multipath.conf** ファイルに以下の記述を追加します。

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

- 次のいずれかのコマンドを実行してマルチパス設定ファイルを変更した後、 **/etc/multipath.conf** ファイルを検証します。

- 設定エラーを表示するには、以下のコマンドを実行します。

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

- 変更が追加された新しい設定を表示するには、以下のコマンドを実行します。

```
# multipath -t
```

- /etc/multipath.conf** ファイルを再読み込みし、 **multipathd** デーモンを再設定して変更を反映します。

```
# service multipathd reload
```

## 関連情報

- man ページの **multipath.conf(5)**

## 3.4. 追加のストレージデバイスの設定

デフォルトでは、DM Multipath をサポートする最も一般的なストレージアレイをサポートしています。

### 手順

- 対応デバイスを含む初期設定値を表示します。

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

- オプション: 既知のマルチパスデバイスとして、デフォルトではサポートされていないストレージデバイスを追加するには、 **/etc/multipath.conf** ファイルを編集して適切なデバイス情報を記入します。

次の例は、HP Open-V シリーズに関する情報を追加する方法を示しています。これは、すべてのパスに障害が発生した場合には、1分間 (または5秒ごとの再試行で12回の再試行を行う) のキュー待ちを行うようにデバイスを設定します。

```
devices {
    device {
```

```

    vendor "HP"
    product "OPEN-V"
    no_path_retry 12
  }
}

```

### 3.5. INITRAMFS ファイルシステムでマルチパスの設定

**initramfs** ファイルシステムでマルチパスを設定できます。マルチパス化したいデバイスを使用していない場合は、ブートが **initramfs** ファイルシステムから出た後まで、設定する必要はありません。

#### 前提条件

- DM マルチパスが設定されています。

#### 手順

- 以下のコマンドを実行して、マルチパス設定ファイルを含む **initramfs** ファイルシステムを再構築します。

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

**initramfs** ファイルシステムからマルチパスを実行してマルチパスの設定ファイルを変更した場合は、**initramfs** ファイルシステムを再構築して変更を有効にする必要があります。root デバイスがマルチパスを使用する場合は、**dracut** コマンドを実行すると、multipath モジュールが自動的に **initramfs** に追加されます。

- オプション：initramfs でマルチパスを実行する必要がある場合に、マルチパスを設定した root デバイスを設定していない場合は、次のコマンドを実行します。

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

#### 注記

**dracut** コマンドには、マルチパスが不要であっても **initramfs** にマルチパスが含まれます。マルチパスなどを停止するには、以下のコマンドを実行します。

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



## 第4章 NVMe デバイスでのマルチパスの有効化

ファイバーチャネル (FC) などのファブリックトランスポートを介して、システムに接続されている NVMe デバイスをマルチパスすることができます。複数のマルチパスソリューションを選択することができます。

### 4.1. ネイティブ NVMe マルチパスと DM MULTIPATH

NVMe デバイスは、ネイティブなマルチパス機能をサポートしています。NVMe にマルチパスを設定する場合、標準の DM Multipath フレームワークと NVMe のネイティブなマルチパスのどちらかを選択できます。

DM Multipath と NVMe のネイティブマルチパスは、どちらも NVMe デバイスのマルチパス方式である ANA (Asymmetric Namespace Access) に対応しています。ANA は、ターゲットとイニシエーター間の最適なパスを特定し、パフォーマンスを向上させます。

ネイティブ NVMe マルチパスを有効にすると、すべての NVMe デバイスにグローバルに適用されます。より高いパフォーマンスを提供できますが、DM Multipath が提供するすべての機能は含まれていません。例えば、ネイティブの NVMe マルチパスは、**failover** と **round-robin** のパス選択方法のみをサポートしています。

デフォルトでは、NVMe マルチパスは Red Hat Enterprise Linux 9 で有効になっており、これは推奨されるマルチパスソリューションです。

### 4.2. ネイティブ NVMe マルチパスの実現

この手順では、ネイティブの NVMe マルチパスソリューションを使用して、接続された NVMe デバイスでマルチパスを有効にします。

#### 前提条件

- NVMe デバイスがシステムに接続されていることを確認します。  
ファブリックトランスポートを介した NVMe の接続の詳細については、[Overview of NVMe over fabric devices](#) を参照してください。

#### 手順

1. カーネルでネイティブ NVMe マルチパスが有効になっているかどうかを確認します。

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

コマンドは以下のいずれかを表示します。

**N**

ネイティブ NVMe マルチパスは無効です。

**Y**

ネイティブ NVMe マルチパスは有効です。

2. ネイティブ NVMe マルチパスが無効になっている場合は、以下のいずれかの方法で有効にしてください。
  - カーネルオプションの使用
    - i. カーネルのコマンドラインに **nvme\_core.multipath=Y** オプションを追加します。

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

- ii. 64 ビットの IBM Z アーキテクチャでは、ブートメニューを更新します。

```
# zipl
```

- iii. システムを再起動します。

- カーネルモジュール設定ファイルの使用

- i. 以下の内容で `/etc/modprobe.d/nvme_core.conf` 設定ファイルを作成します。

```
options nvme_core multipath=Y
```

- ii. **initramfs** ファイルシステムをバックアップします。

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

- iii. **initramfs** ファイルシステムを再構築します。

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

- iv. システムを再起動します。

3. オプション: 実行中のシステムで、NVMe デバイスの I/O ポリシーを変更して、利用可能なすべてのパスに I/O を分散させます。

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

4. オプション: **udev** ルールを使用して I/O ポリシーを永続的に設定します。以下の内容で `/etc/udev/rules.d/71-nvme-io-policy.rules` ファイルを作成します。

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

## 検証

1. システムが NVMe デバイスを認識していることを確認します。

```
# 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. 接続されているすべての NVMe サブシステムをリストアップします。

```
# 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
```

アクティブトランスポートタイプを確認します。例えば、**nvme0 fc** はファイバーチャネルトランスポートで接続されていることを示し、**nvme tcp** は TCP で接続されていることを示しています。

3. カーネルオプションを編集した場合は、カーネルのコマンドラインでネイティブ NVMe マルチパスが有効になっていることを確認してください。

```
# cat /proc/cmdline

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

4. DM Multipath が NVMe 名前空間を、たとえば **nvme0n1** から **nvme3n1** としてではなく、たとえば **nvme0c0n1** から **nvme0c3n1** として報告することを確認します。

```
# 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 ポリシーを変更した場合は、NVMe デバイスで **round-robin** がアクティブな I/O ポリシーになっていることを確認してください。

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

round-robin
```

## 関連情報

- [Configuring kernel command-line parameters](#)

## 4.3. NVME デバイスでの DM MULTIPATH の有効化

この手順では、DM Multipath ソリューションを使用して、接続された NVMe デバイスでマルチパスを有効にします。

## 前提条件

- NVMe デバイスがシステムに接続されていることを確認します。  
ファブリックトランスポートを介した NVMe の接続の詳細については、[Overview of NVMe over fabric devices](#) を参照してください。

## 手順

1. ネイティブ NVMe マルチパスが無効になっていることを確認します。

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

コマンドは以下のいずれかを表示します。

**N**

ネイティブ NVMe マルチパスは無効です。

**Y**

ネイティブ NVMe マルチパスは有効です。

2. ネイティブ NVMe マルチパスが有効な場合は、無効にします。
  - a. カーネルのコマンドラインから **nvme\_core.multipath=Y** オプションを削除しました。

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

- b. 64 ビットの IBM Z アーキテクチャでは、ブートメニューを更新します。

```
# zipl
```

- c. **options nvme\_core multipath=Y** 行が存在する場合は、**/etc/modprobe.d/nvme\_core.conf** ファイルから削除します。

- d. システムを再起動します。

3. DM Multipath が有効になっていることを確認してください。

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

4. 利用可能なすべてのパスに I/O を分配します。**/etc/multipath.conf** ファイルに以下の内容を追加します。

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



## 注記

DM Multipath が NVMe デバイスを管理する場合、`/sys/class/nvme-subsystem/nvme-subsys0/iopolicy` 設定ファイルは I/O ディストリビューションには影響を与えません。

5. 設定の変更を適用するために、**multipathd** サービスをリロードします。

```
# multipath -r
```

6. **initramfs** ファイルシステムをバックアップします。

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

7. **initramfs** ファイルシステムを再構築します。

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

## 検証

1. システムが NVMe デバイスを認識していることを確認します。

```
# 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. 接続されているすべての NVMe サブシステムを一覧表示します。コマンドがそれらを、たとえば、**nvme0c0n1** から **nvme0c3n1** としてではなく、たとえば、**nvme0n1** から **nvme3n2** として報告することを確認します。

```
# 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
```

## 関連情報

- [Configuring kernel command-line parameters](#)
- [DM Multipath の設定](#)

## 第5章 DM MULTIPATH 設定ファイルの編集

DM Multipath では、マルチパスで最も一般的に使用する設定値がデフォルトで提供されています。また、DM Multipath に対応する最も一般的なストレージレイへのサポートも DM Multipath に含まれています。DM-Multipath のデフォルトの設定値は、`/etc/multipath.conf` 設定ファイルを編集するとオーバーライドできます。必要に応じて、サポートされていないデフォルトのストレージレイを設定ファイルに追加することもできます。

対応デバイスなど、デフォルトの設定値の詳細は、次のいずれかのコマンドを実行して確認してください。

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



### 注記

**initramfs** ファイルシステムからマルチパスを実行してマルチパス設定ファイルに変更を加える場合、変更を有効にするには **initramfs** ファイルシステムを再構築する必要があります。

マルチパス設定ファイルで指定するのは、使用状況に応じて必要とされるセクションや、デフォルト値の変更が必要なセクションのみです。使用環境には無関係なファイルのセクションや、デフォルト値を無効にする必要がないファイルのセクションでは、初期ファイルに指定されているコメントアウトを削除する必要はありません。

設定ファイルでは、正規表現の記述構文を使用できます。

### 5.1. 設定ファイルの概要

マルチパス設定ファイルは以下のセクションに分かれています。

#### blacklist

マルチパス設定の対象として考慮しないデバイスの一覧。

#### blacklist\_exceptions

指定されていなければ、**blacklist** セクションのパラメーターに従って無視されるマルチパスのデバイスの一覧。

#### defaults

DM Multipath のデフォルトの全般設定。

#### multipaths

マルチパスデバイスの特性に関する個別設定。ここで指定する値は、設定ファイルの **overrides**、**devices**、および **defaults** のセクションで指定されている値より優先されます。

#### devices

ストレージコントローラーの個別設定。ここで指定する値は、設定ファイル内の **defaults** セクションで指定されている値より優先されます。デフォルトでは対応していないストレージレイを使用している場合は、そのアレイ用の **devices** サブセクションを作成する必要があります。

#### overrides

すべてのデバイスに適用される設定。ここで指定する値は、設定ファイルの **devices** セクションおよび **defaults** セクションで指定されている値より優先されます。

システムがマルチパスデバイスの属性を決定するとき、次の順序で **multipath.conf** ファイルから個別のセクションの設定をチェックします。

1. **multipaths** セクション
2. **overrides** セクション
3. **devices** セクション
4. **defaults** セクション

## 5.2. DM MULTIPATH がデバイスタイムアウトの上書き

**recovery\_tmo sysfs** オプションは、特定の iSCSI デバイスのタイムアウトを制御します。次のオプションは、システム全体の **recovery\_tmo** 値を上書きします。

- **replacement\_timeout** 設定オプションは、システム全体で全 iSCSI デバイスの **recovery\_tmo** 値を上書きします。
- DM Multipath が管理するすべての iSCSI デバイスで、DM Multipath の **fast\_io\_fail\_tmo** オプションは、システム全体の **recovery\_tmo** 値を上書きします。  
DM Multipath の **fast\_io\_fail\_tmo** オプションは、ファイバーチャネルデバイスの **fast\_io\_fail\_tmo** オプションを上書きします。

DM Multipath の **fast\_io\_fail\_tmo** オプションは **replacement\_timeout** よりも優先します。Red Hat では、**replacement\_timeout** を使用して、DM Multipath が管理するデバイスの **recovery\_tmo** を上書きすることは推奨しません。これは、**multipathd** サービスが再読み込みを行うと、DM Multipath が常に **recovery\_tmo** をリセットするためです。

## 5.3. マルチパス設定ファイルのデフォルトの編集

**/etc/multipath.conf** 設定ファイルには **defaults** セクションがあり、以下のように **user\_friendly\_names** パラメーターを **yes** に設定できます。

```
defaults {
    user_friendly_names yes
}
```

上記は、**user\_friendly\_names** パラメーターのデフォルト値を上書きします。**multipath.conf file** の **defaults** セクションで設定されたデフォルト値は、**devices**、**multipath** で指定された属性によって上書きされない限り、または **multipath.conf** ファイルのセクションをオーバーライドしない限り、DM Multipath によって使用されます。

### 手順

1. **/etc/multipath.conf** 設定ファイルを表示します。このファイルには、設定のデフォルトのテンプレートが含まれています。

```
#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
# fallback      immediate
# no_path_retry fail
# user_friendly_names yes
#}
```

- 設定パラメーターのデフォルト値を上書きします。このテンプレートから **defaults** のセクションに関連する行をコピーして、コメントを外すことができます。  
たとえば、**path\_grouping\_policy** パラメーターを、デフォルト値の **failover** ではなく **multibus** に上書きするには、以下のように、テンプレートで該当行を見つけて設定ファイルの defaults セクションにコピーし、そのコメントを外します。

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

- 次のいずれかのコマンドを実行してマルチパス設定ファイルを変更した後、**/etc/multipath.conf** ファイルを検証します。

- 設定エラーを表示するには、以下のコマンドを実行します。

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

- 変更が追加された新しい設定を表示するには、以下のコマンドを実行します。

```
# multipath -t
```

- /etc/multipath.conf** ファイルを再読み込みし、**multipathd** デーモンを再設定して変更を反映します。

```
# service multipathd reload
```

## 関連情報

- multipath.conf(5)** and **multipathd(8)** man pages

## 5.4. 特定デバイスのマルチパス設定の編集

**multipath.conf** 設定ファイルの **multipaths** セクションで、必須の WWID パラメーターによって参照される個々のマルチパスデバイスに固有の設定を追加できます。

このデフォルトは DM Multipath により使用され、**multipath.conf** ファイルの **overrides** セクション、**defaults** セクション、および **devices** セクションに設定された属性を上書きします。**multipaths** セクションには、任意の数のマルチパスサブセクションを含めることができます。

## 手順

- 特定のマルチパスデバイスの **multipaths** セクションを変更します。設定ファイルで、2つの特定マルチパスデバイスに対して **multipath** 属性を指定している例を以下に示します。
  - 1つ目のデバイスの WWID は **3600508b4000156d70001200000b0000** で、シンボリック名

は **yellow** です。

- 2番目のマルチパスデバイスの WWID は **1DEC\_321816758474** で、シンボリック名 **red** があります。

**rr\_weight** 属性は **priorities** に設定されています。

```

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. 次のいずれかのコマンドを実行してマルチパス設定ファイルを変更した後、**/etc/multipath.conf** ファイルを検証します。

- 設定エラーを表示するには、以下のコマンドを実行します。

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

- 変更が追加された新しい設定を表示するには、以下のコマンドを実行します。

```
# multipath -t
```

3. **/etc/multipath.conf** ファイルを再読み込みし、**multipathd** デーモンを再設定して変更を反映します。

```
# service multipathd reload
```

## 関連情報

- man ページの **multipath.conf(5)**

## 5.5. プロトコルを使用した特定デバイスのマルチパス設定の変更

トランスポートプロトコルに基づいてマルチパスデバイスパスを設定できます。**/etc/multipath.conf** ファイルの **overrides** セクションの **protocol** サブセクションを使用すると、特定のパスでマルチパス設定を上書きできます。これにより、ファイバーチャネル(FC)や Internet Small Computer Systems Interface(iSCSI)などの複数のトランスポートプロトコルでマルチパスデバイスにアクセスすることができます。

**protocol** サブセクションで設定したオプションは、**overrides**、**devices**、および **defaults** セクションの値を上書きします。これらのオプションは、サブセクションの **type** パラメーターに一致するトランスポートプロトコルを使用するデバイスにのみ適用されます。

## 前提条件

- システムに Device Mapper(DM)マルチパスを設定している。
- すべてのパスが同じトランスポートプロトコルを使用するのではなく、マルチパスデバイスがある。

## 手順

1. 以下のコマンドを実行して、特定のパスプロトコルを表示します。

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

2. 各マルチパスタイプに **protocol** サブセクションを追加して、**/etc/multipath.conf** ファイルの **overrides** セクションを編集します。

- **scsi:fc** プロトコルを使用するパスデバイスの設定。

```
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** プロトコルを使用するパスデバイスの設定

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

- 他のすべてのプロトコルを使用するパスデバイスの設定 :

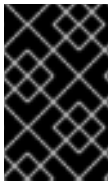
```
overrides {
    dev_loss_tmo 60
    fast_io_fail_tmo 8
    protocol {
        type "<type of protocol>"
    }
}
```

```

dev_loss_tmo 60
fast_io_fail_tmo 8
}

```

**overrides** セクションには、複数の **protocol** サブセクションを含めることができます。



### 重要

**protocol** サブセクションには **type** パラメーターが含まれている必要があります。一致する **type** パラメーターを持つすべてのパスの設定は、**protocol** サブセクションに記載されている残りのパラメーターで更新されます。

### 関連情報

- man ページの **multipath.conf(5)**

## 5.6. ストレージコントローラーのマルチパス設定の編集

**multipath.conf** 設定ファイルの **devices** セクションは、各ストレージデバイスの属性を設定します。デバイスを含むパスの **multipath.conf** ファイルの **multipaths** セクションまたは **overrides** セクションに指定された属性により上書きされた場合を除き、これらの値は DM Multipath により使用されます。これらの属性は、**multipath.conf** ファイルの **defaults** セクションに設定された属性を上書きします。

### 手順

1. サポートされているデバイスを含む、デフォルトの設定値に関する情報を表示します。

```

# multipathd show config
# multipath -t

```

マルチパスに対応しているデバイスの多くは、デフォルトでマルチパスの設定に含まれていません。

2. オプション: デフォルトの設定値を変更する必要がある場合は、それらの値を上書きするデバイスの設定ファイルにエントリーを含めることで、デフォルト値を上書きできます。**multipathd show config** コマンドが表示する、目的のデバイスのデフォルト値をコピーして、変更したい値に書き換えることができます。
3. **vendor** と **product** のパラメーターを設定して、デフォルトで自動的に設定されないデバイスを設定ファイルの **devices** セクションに追加します。次の例に示すように、**/sys/block/device\_name/device/vendor** および **/sys/block/device\_name/device/model** ファイルを開いてこれらの値を見つけます。ここで、**device\_name** はマルチパスされるデバイスです。

```

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

```

4. オプション: 特定のデバイスに応じて、追加のパラメーターを指定します。

**active/active** デバイス

通常、この場合、追加のパラメーターを設定する必要はありません。必要に応じて、**path\_grouping\_policy** を **multibus** に設定できます。この他に、設定が必要となる可能性があるパラメーターは **no\_path\_retry** と **rr\_min\_io** です。

### active/passive デバイス

I/O を持つパスが自動的に **passive** パスに切り替えられる場合は、チェッカーの関数を、I/O をパスに送信しない関数に変更し、それが正しく動作するかどうかを検証する必要があります (これを行わないとデバイスはフェイルオーバーし続けます)。これは、**path\_checker** を **tur** に設定したことを意味します。これは、ほとんどの場合、Test Unit Ready コマンドをサポートするすべての SCSI デバイスで機能します。

パスの切り替えに特殊なコマンドを必要とするデバイスにマルチパスを設定するには、ハードウェアハンドラーカーネルモジュールが必要になります。現在、利用可能なハードウェアハンドラーは **emc** です。このハンドラーが目的のデバイスに使用できない場合は、そのデバイスにマルチパスを設定できない可能性があります。

マルチパス設定ファイルの **device** エントリーの例を以下に示します。

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

- 次のいずれかのコマンドを実行してマルチパス設定ファイルを変更した後、**/etc/multipath.conf** ファイルを検証します。

- 設定エラーを表示するには、以下のコマンドを実行します。

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

- 変更が追加された新しい設定を表示するには、以下のコマンドを実行します。

```
# multipath -t
```

- /etc/multipath.conf** ファイルを再読み込みし、**multipathd** デーモンを再設定して変更を反映します。

```
# service multipathd reload
```

### 関連情報

- multipath.conf(5)** and **multipathd(8)** man pages

## 5.7. すべてのデバイスへのマルチパス値の設定

**multipath.conf** 設定ファイルの **overrides** セクションを使用すると、すべてのデバイスの設定値を設定できます。このセクションは、**multipath.conf** 設定ファイルの **devices** セクションおよび **defaults** セクションの両方で対応しているすべての属性に対応します。これは、**vendor**、**product**、および **revision** 以外のすべての **devices** セクション属性になります。

このデバイスを含むパスの **multipath.conf** ファイルで、**multipaths** セクションに指定された属性により上書きされた場合を除き、この属性はすべてのデバイスの DM Multipath により使用されます。この属性は、**multipath.conf** ファイルの **devices** セクションおよび **defaults** セクションに設定された属性を上書きします。

## 手順

1. デバイス固有の設定を上書きします。たとえば、すべてのデバイスで **no\_path\_retry** を **fail** に設定できます。すべてのパスに障害が発生した場合は、次のコマンドを使用してキューイングをオフにします。これにより、デバイス固有の設定が上書きされます。

```
overrides {
    no_path_retry fail
}
```

2. 次のいずれかのコマンドを実行してマルチパス設定ファイルを変更した後、**/etc/multipath.conf** ファイルを検証します。

- 設定エラーを表示するには、以下のコマンドを実行します。

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

- 変更が追加された新しい設定を表示するには、以下のコマンドを実行します。

```
# multipath -t
```

3. **/etc/multipath.conf** ファイルを再読み込みし、**multipathd** デーモンを再設定して変更を反映します。

```
# service multipathd reload
```

## 関連情報

- man ページの **multipath.conf(5)**



`find_multipaths` の組み込みデフォルト値は **off** です。ただし、`mpathconf` により作成されたデフォルトの `multipath.conf` ファイルは、`find_multipaths` の値を **on** に設定します。

`find_multipaths` パラメーターが **on** に設定されている場合、マルチパスを使用したくない複数のパスを持つデバイスでのみマルチパスを無効にします。このため、通常、デバイスでマルチパスを無効にする必要はありません。

以前に作成したマルチパスデバイスを **blacklist** に追加する場合は、`-w` オプションを使用して `/etc/multipath/wwids` ファイルからそのデバイスの WWID を削除すると、他のプログラムとの問題を回避するのに役立つことがあります。たとえば、WWID `3600d0230000000000e13954ed5f89300` のデバイス `/dev/sdb` を `/etc/multipath/wwids` ファイルから削除する場合は、以下のいずれかの方法を使用することができます。

- デバイス名でマルチパスデバイスを削除する。

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

- マルチパスデバイスの WWID を使用して削除する。

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

また、`-W` オプションで `/etc/multipath/wwids` ファイルを更新することも可能です。これにより、`/etc/multipath/wwids` ファイルがリセットされ、現在のマルチパスデバイスの WWID のみが含まれるようになります。リセットする場合は、以下を実行してください。

```
#multipath -W
successfully reset wwids
```

## 関連情報

- man ページの `multipath.conf(5)`

## 6.2. 特定のデバイスでマルチパスを無効にする基準

以下のいずれかの基準により、デバイスでマルチパスを無効にできます。

- WWID
- デバイス名
- デバイスの種別
- プロパティ
- プロトコル

すべてのデバイスに対して、DM Multipath は以下の順番でこの基準を評価します。

1. プロパティ
2. `devnode`
3. `device`



#### 4. プロトコル

#### 5. wwid

上記の基準のいずれかによってデバイスが無効になっていることが判明した場合、DM Multipath はそのデバイスを **multipathd** による処理から除外し、後の基準を評価しません。各基準において、デバイスが例外リストと無効なデバイスリストの両方に一致する場合は、例外リストが優先されます。



#### 注記

デフォルトでは、設定ファイルの初期 **blacklist** セクションをコメントアウトした後でも、さまざまなデバイス種別が無効化されます。

#### 関連情報

- [マルチパスを無効にしたデバイスに対する例外の追加](#)

### 6.3. WWID によるマルチパスの無効化

WWID (World-Wide Identification) を使用して、個々のデバイスでマルチパスを無効にできます。

#### 手順

1. **wwid** エントリを使用して、**/etc/multipath.conf** 設定ファイルのデバイスを無効にします。以下の例は、**26353900f02796769** が WWID であるデバイスを無効にする DM Multipath 設定ファイル内の行を示しています。

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

2. 次のいずれかのコマンドを実行してマルチパス設定ファイルを変更した後、**/etc/multipath.conf** ファイルを検証します。

- 設定エラーを表示するには、以下のコマンドを実行します。

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

- 変更が追加された新しい設定を表示するには、以下のコマンドを実行します。

```
# multipath -t
```

3. **/etc/multipath.conf** ファイルを再読み込みし、**multipathd** デーモンを再設定して変更を反映します。

```
# service multipathd reload
```

### 6.4. デバイス名によるマルチパスの無効化

DM Multipath がマルチパスデバイスにグループ化しないように、デバイス名でデバイスタイプのマルチパスを無効にできます。

## 手順

1. **devnode** エントリーを使用して、**/etc/multipath.conf** 設定ファイルのデバイスを無効にします。  
以下の例は、すべての **sd\*** デバイスを無効にするため、すべての SCSI デバイスを無効にする DM Multipath 設定ファイル内の行を示しています。

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

**devnode** エントリーを使用すると、特定タイプのすべてのデバイスではなく、個別のデバイスを無効にできます。ただし、**udev** ルールで静的にマッピングされていない限り、再起動時に特定のデバイス名が同じ名前になる保証がないため、この方法は推奨されません。たとえば、システムが再起動するとデバイス名が **/dev/sda** から **/dev/sdb** に変わる可能性があります。

デフォルトでは、DM Multipath は以下の **devnode** エントリーを使用して、SCSI、NVMe、または DASD 以外のすべてのデバイスを無効にします。

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

このエントリーを無効にするデバイスは、通常、DM Multipath に対応していません。

2. 次のいずれかのコマンドを実行してマルチパス設定ファイルを変更した後、**/etc/multipath.conf** ファイルを検証します。

- 設定エラーを表示するには、以下のコマンドを実行します。

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

- 変更が追加された新しい設定を表示するには、以下のコマンドを実行します。

```
# multipath -t
```

3. **/etc/multipath.conf** ファイルを再読み込みし、**multipathd** デーモンを再設定して変更を反映します。

```
# service multipathd reload
```

## 関連情報

- [マルチパスを無効にしたデバイスに対する例外の追加](#)

## 6.5. デバイスの種別によるマルチパスの無効化

デバイスセクションを使用して、デバイスのマルチパスを無効にできます。

## 手順

1. **device** セクションを使用して、**/etc/multipath.conf** 設定ファイルのデバイスを無効にします。  
以下の例は、IBM DS4200 および HP のすべてのデバイスでのマルチパスを無効にします。

```

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

```

- 次のいずれかのコマンドを実行してマルチパス設定ファイルを変更した後、**/etc/multipath.conf** ファイルを検証します。

- 設定エラーを表示するには、以下のコマンドを実行します。

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

- 変更が追加された新しい設定を表示するには、以下のコマンドを実行します。

```
# multipath -t
```

- /etc/multipath.conf** ファイルを再読み込みし、**multipathd** デーモンを再設定して変更を反映します。

```
# service multipathd reload
```

## 6.6. UDEV プロパティによるマルチパスの無効化

**udev** プロパティパラメーターを使用して、デバイスのマルチパスを無効にできます。

### 手順

- property** パラメーターを使用して、**/etc/multipath.conf** 設定ファイルのデバイスを無効にします。このパラメーターは、デバイスの **udev** 環境変数名と一致する正規表現の文字列です。以下の例は、**udev** プロパティ **ID\_ATA** を持つすべてのデバイスでマルチパスを無効にします。

```

blacklist {
    property "ID_ATA"
}

```

- 次のいずれかのコマンドを実行してマルチパス設定ファイルを変更した後、**/etc/multipath.conf** ファイルを検証します。

- 設定エラーを表示するには、以下のコマンドを実行します。

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

- 変更が追加された新しい設定を表示するには、以下のコマンドを実行します。

```
# multipath -t
```

3. `/etc/multipath.conf` ファイルを再読み込みし、`multipathd` デーモンを再設定して変更を反映します。

```
# service multipathd reload
```

## 6.7. デバイスプロトコルによるマルチパスの無効化

デバイスプロトコルを使用して、デバイスのマルチパスを無効にすることができます。

### 手順

1. オプション: パスが使用するプロトコルを表示します。

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

2. `protocol` セクションを使用して、`/etc/multipath.conf` 設定ファイルのデバイスを無効にします。

以下の例は、未定義のプロトコルまたは未知の SCSI トランスポートタイプを持つすべてのデバイスでマルチパスを無効にします。

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

DM Multipath は以下のプロトコル文字列を認識します。

- `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. 次のいずれかのコマンドを実行してマルチパス設定ファイルを変更した後、`/etc/multipath.conf` ファイルを検証します。

- 設定エラーを表示するには、以下のコマンドを実行します。

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

- 変更が追加された新しい設定を表示するには、以下のコマンドを実行します。

```
# multipath -t
```

4. `/etc/multipath.conf` ファイルを再読み込みし、`multipathd` デーモンを再設定して変更を反映します。

```
# service multipathd reload
```

## 6.8. マルチパスを無効にしたデバイスに対する例外の追加

マルチパスが現在無効になっているデバイスに例外を追加することで、マルチパスを有効にできます。

### 前提条件

- 特定のデバイスでマルチパスが無効になっている。

### 手順

1. `/etc/multipath.conf` 設定ファイルの `blacklist_exceptions` セクションを使用して、デバイスでマルチパスを有効にします。

設定ファイルの `blacklist_exceptions` セクションでデバイスを指定する場合は、`blacklist` セクションで指定したものと同一基準を使用して例外を指定する必要があります。たとえば、無効にしたデバイスがその WWID に関連付けられている場合でも、WWID 例外は `devnode` エントリーで無効になっているデバイスには適用されません。同様に、`devnode` 例外は `devnode` エントリーにしか適用されず、`device` 例外はデバイスエントリーにしか適用されません。

#### 例6.1 WWID による例外

たとえば、デバイスが多数あり、その中の1つのデバイス (以下の例では WWID が `3600d0230000000000e13955cc3757803` のデバイス) でのみマルチパスを有効にする場合は、有効にするデバイス以外のものを1つ1つ無効にするのではなく、一旦すべてのデバイスを無効にしてから、`/etc/multipath.conf` ファイルに以下の行を追加し、必要なデバイスのみを有効にします。

```
blacklist {
    wwid ".*"
}

blacklist_exceptions {
    wwid "3600d0230000000000e13955cc3757803"
}
```

または、感嘆符 (!) を使用して `blacklist` エントリーを反転することもできます。これにより、指定した WWID を除くすべてのデバイスを無効にできます。

```
blacklist {
    wwid "!3600d02300000000000e13955cc3757803"
}
```

### 例6.2 udev プロパティによる例外

**property** パラメーターの挙動は、他の **blacklist\_exception** パラメーターとは異なります。 **property** パラメーターの値は、 **udev** データベース内の変数の名前と一致する必要があります。それ以外の場合は、デバイスは無効になります。このパラメーターを使用すると、USB スティックやローカルハードドライブなどの特定の SCSI デバイスでマルチパスを無効にできます。

合理的にマルチパス化できる SCSI デバイスでのみマルチパスを有効にするには、以下の例のようにこのパラメーターを **SCSI\_IDENT\_ID\_WWN** に設定します。

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

- 次のいずれかのコマンドを実行してマルチパス設定ファイルを変更した後、 **/etc/multipath.conf** ファイルを検証します。

- 設定エラーを表示するには、以下のコマンドを実行します。

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

- 変更が追加された新しい設定を表示するには、以下のコマンドを実行します。

```
# multipath -t
```

- /etc/multipath.conf** ファイルを再読み込みし、 **multipathd** デーモンを再設定して変更を反映します。

```
# service multipathd reload
```

## 第7章 マルチパス化されたボリュームの管理

DM Multipath が提供するコマンドのうち、マルチパスボリュームを管理するコマンドを以下に示します。

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

### 7.1. オンラインのマルチパスデバイスのサイズ変更

オンラインのマルチパスデバイスのサイズを変更する必要がある場合は、以下の手順に従ってください。

#### 手順

1. 物理デバイスのサイズを変更します。
2. 以下のコマンドを実行し、論理ユニット番号 (LUN) までのパスを検索します。

```
# multipath -l
```

3. パスのサイズを変更します。SCSI デバイスの場合は、デバイスの **rescan** ファイルに 1 と書き込むと、SCSI ドライバーによる再スキャンが行われます。以下にコマンド例を示します。

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

各パスデバイスに対してこのコマンドを実行します。たとえば、パスデバイスが **sda**、**sdb**、**sde**、および **sdf** の場合は、次のコマンドを実行します。

```
# 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. マルチパスデバイスのサイズを変更します。

```
# multipathd resize map multipath_device
```

5. ファイルシステムのサイズを変更します (LVM または DOS のパーティションが使用されていないことを前提とします)。

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

### 7.2. ROOT ファイルシステムをシングルパスデバイスからマルチパスデバイスへ移動

シングルパスのデバイスにシステムをインストールしてから、別のパスを root ファイルシステムに追加する場合は、root ファイルシステムをマルチパスのデバイスに移行する必要があります。シングルパスからマルチパスへの移行については、以下の手順を参照してください。

## 前提条件

- **device-mapper-multipath** パッケージがインストールされている。

## 手順

1. **/etc/multipath.conf** 設定ファイルを作成し、multipath モジュールをロードし、**multipathd systemd** サービスを有効にします。

```
# yum install device-mapper-multipath
```

## 手順

1. 以下のコマンドを実行して **/etc/multipath.conf** 設定ファイルを作成、マルチパスモジュールをロードして **multipathd** の **chkconfig** を **on** に設定します。
2. **find\_multipaths** 設定パラメーターが **yes** に設定されていない場合は、[Preventing devices from multipathing](#) の説明にしたがって **/etc/multipath.conf** の **blacklist** および **blacklist\_exceptions** セクションを編集します。
3. 検出され次第、root デバイスにマルチパスデバイスを構築させるため、次のコマンドを実行します。また、このコマンドを実行すると、パスが1つしかない場合でも必ず **find\_multipaths** がデバイスを許可するようになります。

```
# multipath -a root_devname
```

たとえば、root デバイスが **/dev/sdb** の場合は、次のコマンドを実行します。

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

4. **multipath** コマンドを実行して設定ファイルの設定が正しく行われたことを確認し、次のような行が出力されていることを検索します。これは、コマンドがマルチパスデバイスの作成に失敗したことを示しています。

```
date wwid: ignoring map
```

たとえば、デバイスの WWID が **3600d02300069c9ce09d41c4ac9c53200** の場合は、次のような行が出力に表示されます。

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

5. **multipath** を使用して **initramfs** ファイルシステムを再構築します。

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

6. マシンをシャットダウンします。
7. マシンを起動します。
8. 他のパスがマシンから見えるようにする。



## 検証手順

- 以下のコマンドを実行して、マルチパスデバイスが作成されているかどうかを確認します。

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

## 7.3. SWAP ファイルシステムをシングルパスデバイスからマルチパスデバイスへ移動

デフォルトでは、swap デバイスは論理ボリュームとして設定されます。論理ボリュームグループを構成する物理ボリュームでマルチパスを設定している限り、このようなデバイスをマルチパスデバイスとして設定する特別な手順は必要ありません。ただし、swap デバイスが LVM ボリュームではなく、デバイス名でマウントする場合には、**/etc/fstab** ファイルに、適切なマルチパスデバイス名を設定しないといけない場合があります。

### 手順

1. **/etc/multipath/wwids** ファイルにデバイスの WWID を追加します。

```
# multipath -a swap_devname
```

たとえば、root デバイスが **/dev/sdb** の場合は、次のコマンドを実行します。

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

2. 設定ファイルの設定が正しく行われたことを確認するため、**multipath** コマンドを実行して、次のような行が出力されていることを検索します。

```
date wwid: ignoring map
```

これは、コマンドがマルチパスデバイスの作成に失敗したことを示しています。

たとえば、デバイスの WWID が 3600d02300069c9ce09d41c4ac9c53200 の場合は、次のような行が出力に表示されます。

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

3. **/etc/multipath.conf** ファイルに、swap デバイスのエイリアスを設定します。

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

4. **/etc/fstab** ファイルで、root デバイスへの古いデバイスパスを、マルチパスデバイスに置き換えます。

たとえば、**/etc/fstab** ファイルに、以下のようなエントリーがあるとします。

-

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

エントリーを以下のように変更します。

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

5. multipath を使用して initramfs ファイルシステムを再構築します。

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

6. マシンをシャットダウンします。
7. マシンを起動します。
8. 他のパスがマシンから見えるようにする。

## 検証手順

- スワップデバイスがマルチパスデバイス上にあるかどうかを確認します。

```
# swapon -s
```

以下に例を示します。

```
# swapon -s

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

ファイル名は、マルチパススワップデバイスと一致させる必要があります。

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

## 7.4. DMSETUP コマンドでデバイスマッパーエントリーの特定

**dmsetup** コマンドを使って、マルチパスを設定したデバイスに一致するデバイスマッパーのエントリーを検索できます。

### 手順

- デバイスマッパーの全デバイスとそのメジャー番号、マイナー番号を表示します。dm デバイス名は、マイナー番号で特定できます。たとえば、マイナー番号 3 は、マルチパスを設定したデバイス **/dev/dm-3** に対応します。

```
# 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 デーモンの管理

**multipathd** コマンドを使用して、**multipathd** デーモンを管理できます。

### 手順

- **multipathd show maps** コマンドの出力の標準的なデフォルトフォーマットを表示します。

```

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

```

- 一部の **multipathd** コマンドには、後にワイルドカードが付いた **format** オプションを含むものがあります。次のコマンドを実行すると、使用できるワイルドカードの一覧を表示できます。

```

# multipathd show wildcards

```

- **multipathd** が監視しているマルチパスデバイスを、マルチパスワイルドカードを使用したフォーマット文字列で、regular および raw フォーマットで表示します。

```

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

```

**multipathd** コマンドは、「raw」形式のバージョンでマルチパスデバイスとパスのステータスを表示する **format** コマンドに対応しています。raw 形式ではヘッダーは出力されず、フィールドがパディングされていないため、列とヘッダーが調整されていません。代わりに、フィールドは **format** 文字列の指定どおりに出力します。このため、出力はスクリプトで使いやすくなります。**multipathd show wildcards** コマンドを使用すると、書式設定文字列に使用されたワイルドカードを表示することができます。

- **multipathd** が監視しているパスを、マルチパスワイルドカードを含むフォーマット文字列で、regular および raw フォーマットで表示します。

```

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

```

- **multipathd show maps** の場合は、非 Raw 形式と Raw 形式の違いを表示します。**raw** 形式ではヘッダーがなく、単一のスペースのみが列の間にあることに注意してください。

```

# 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
```

#### 関連情報

- [multipathd\(8\) の man ページ](#)

## 第8章 ストレージデバイスの削除

実行中のシステムからストレージデバイスを安全に削除できるので、システムメモリのオーバーロードやデータ損失を防ぐことができます。

### 前提条件

- I/O フラッシュ中にシステムメモリの読み込みが増加するため、ストレージデバイスを削除する前に、システムのメモリが十分であることを確認する必要があります。次のコマンドを使用して、システムの現在のメモリ負荷および空きメモリを表示します。

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

- Red Hat では、以下のシステムでのストレージデバイスの削除は推奨していません。
  - 空きメモリが合計メモリの 5% 未満 (サンプル 100 件の内 10 件以上)。
  - スワップが有効になっている (`vmstat` コマンドの出力で **si** と **so** のコラムが 0 以外の値)。

### 8.1. ストレージデバイスの安全な削除

稼働中のシステムからストレージデバイスを安全に取り外すには、上から下へのアプローチが必要です。アプリケーションやファイルシステムなどの最上位層から始め、物理デバイスなどの最下位層に向かって作業を進めます。

ストレージデバイスは複数の方法で使用でき、物理デバイスの上層に別の仮想設定を指定できます。例えば、デバイスの複数のインスタンスをマルチパスデバイスにグループ化したり、RAID の一部にしたり、LVM グループの一部にしたりすることが可能です。さらに、デバイスはファイルシステムを介してアクセスすることもできるし、raw デバイスのように直接アクセスすることもできます。

上から下へのアプローチを用いながら、次のことを確認する必要があります。

- 削除したいデバイスが使用中でないこと
- デバイスへの保留中の I/O がすべてフラッシュされる
- オペレーティングシステムがストレージデバイスを参照していない

### 8.2. ブロックデバイスの削除

実行中のシステムからブロックデバイスを安全に削除することで、システムメモリのオーバーロードやデータ損失を防ぐことができます。



#### 警告

SCSI バスを再びスキャンしたり、ここで説明されている手順に従わずにオペレーティングシステムを変更する別のアクションを実行すると、I/O タイムアウトが原因で遅延が発生したり、デバイスやデータが予期せず削除されたりする可能性があります。

## 前提条件

- マルチパスデバイスを削除してそのパスデバイスにアクセスできない場合は、マルチパスデバイスのキューを無効にしておく。

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

無効にすることで、デバイスの I/O が失敗し、デバイスを使用しているアプリケーションがシャットダウンできるようになります。

- 他のアプリケーションやサービスで、削除予定のデバイスが使用されていないことを確認します。
- 削除するデバイスのデータを確実にバックアップしてください。

## 手順

1. **umount** コマンドを使用して、デバイスにマウントされているファイルシステムをすべてアンマウントします。
2. MD RAID アレイまたは、デバイスが所属する LVM ボリュームからデバイスを削除します。デバイスの種類に応じて、以下のいずれかの手順を実行します。
  - たとえば、デバイスが LVM グループのメンバーで、マルチパスデバイスの場合は、次のコマンドを実行します。

- a. データを別のデバイスに移動します。

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

- b. ボリュームグループからデバイスを削除します。

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

- c. 必要に応じて、物理デバイスから LVM メタデータを削除します。

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

- マルチパスデバイスを削除する場合は、次のコマンドを実行します。

- a. デバイスへの全パスを表示します。

```
# multipath -l
```

このコマンドの出力は、後のステップで必要になります。

- b. I/O をフラッシュして、マルチパスデバイスを削除します。

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

- デバイスがマルチパスデバイスとして設定されていない場合や、デバイスがマルチパスデバイスとして設定されていて、過去に I/O を個別のパスに渡している場合は、未処理の I/O を、使用されている全デバイスパスにフラッシュします。

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

■

この操作は、**umount** コマンドまたは **vgreduce** コマンドで I/O がフラッシュされないデバイスに直接アクセスする場合に重要になります。

- SCSI デバイスを取り外す場合は、以下のコマンドを実行します。
  - a. システム上のアプリケーション、スクリプト、またはユーティリティーで、**/dev/sd**、**/dev/disk/by-path**、または **major:minor** 番号など、デバイスのパスベースの名前への参照をすべて削除します。参照を削除することで、今後追加される別のデバイスが現在のデバイスと混同されないようにします。
  - b. SCSI サブシステムからデバイスへの各パスを削除します。

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

デバイスが以前にマルチパスデバイスとして使用されていた場合、**device-name** は、**multipath -l** コマンドの出力からの内容に置き換えます。

3. 稼働中のシステムから物理デバイスを削除します。このデバイスを削除しても、他のデバイスへの I/O は停止しないことに注意してください。

## 関連情報

- **multipath(8)**、**pvmove(8)**、**vgreduce(8)**、**blockdev(8)**、および **umount(8)** man ページ。

## 第9章 DM MULTIPATH のトラブルシューティング

マルチパス設定の実装に問題がある場合は、さまざまな項目を確認できます。マルチパス設定の反応が遅かったり、機能しない場合は、以下の問題が原因として考えられます。

### マルチパスデーモンが実行していない

マルチパス設定の実装に問題がある場合は、[DM Multipath の設定](#) で説明するように、**multipathd** デーモンが実行していることを確認してください。マルチパスが設定されているデバイスを使用するには、**multipathd** デーモンを実行しておく必要があります。

### queue\_if\_no\_path 機能に関する問題

マルチパスデバイスが **features "1 queue\_if\_no\_path"** オプションで設定されている場合は、1つ以上のパスが復元されるまで、I/O を発行するすべてのプロセスがハングアップします。

## 9.1. QUEUE\_IF\_NO\_PATH 機能に関する問題のトラブルシューティング

マルチパスデバイスが **features "1 queue\_if\_no\_path"** オプションで設定されている場合は、1つ以上のパスが復元されるまで、I/O を発行するすべてのプロセスがハングアップします。これを回避するには、**/etc/multipath.conf** ファイルに **no\_path\_retry N** パラメーターを設定します (N はシステムでパスを再試行する回数に置き換えます)。

**features "1 queue\_if\_no\_path"** オプションを使用する必要がある、ここで説明する問題が発生した場合は、特定の LUN に対して起動時にキューイングポリシーを無効にできます (すべてのパスが利用できません)。

### 手順

- 特定のデバイスのキューイングを無効にします。

```
# multipathd disablequeueing map device
```

- すべてのデバイスのキューイングを無効にします。

```
# multipathd disablequeueing maps
```

デバイスのキューイングを無効にしたら、**multipathd** が再起動または再読み込みされるか、以下のコマンドのいずれかを実行するまで、そのデバイスは無効になります。

- 特定のデバイスのキューイングを以前の値に戻します。

```
# multipathd restorequeueing map device
```

- 全デバイスのキューイングを以前の値に戻します。

```
# multipathd restorequeueing maps
```

## 9.2. MULTIPATHD 対話式コンソールでトラブルシューティング

**multipathd -k** コマンドは、**multipathd** デーモンに対する対話式のインターフェースです。このコマンドを実行すると対話式のマルチパスコンソールが立ち上がります。このコマンドを実行した後、**help** を入力すると、利用可能なコマンドの一覧が表示され、**Ctrl+D** を押すと終了できます。



**multipathd** 対話型コンソールを使用して、システムで発生した問題のトラブルシューティングを行います。

## 手順

- コンソールを終了する前に、デフォルトを含むマルチパスの設定を表示します。

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

- **multipath.conf** ファイルへの変更が multipath に反映されていることを確認します。

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

- パスチェッカーが正常に動作していることを確認します。

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

- また、対話型コンソールを起動せずに、コマンドラインから直接1つの **multipathd** 対話型コマンドを実行することも可能です。例えば、multipath が **multipath.conf** ファイルへの変更を取り込んだかどうかを確認するには、次のように実行します。

```
# multipathd reconfigure
```

## 第10章 EH\_DEADLINE を使用したストレージエラーからの回復における最大時間の設定

障害が発生した SCSI デバイスを復旧するのに許容できる最大時間を設定できます。この設定は、ストレージハードウェアが不具合により応答しなくなっても、I/O 応答時間を保証します。

### 10.1. EH\_DEADLINE パラメーター

SCSI エラー処理 (EH) メカニズムは、障害が発生した SCSI デバイスでエラーからの復旧の実行を試行します。SCSI ホストオブジェクト **eh\_deadline** パラメーターでは、復旧時間の最大量を設定できます。設定した時間が過ぎると、SCSI EH は、ホストバスアダプター (HBA) 全体を停止してリセットします。

**eh\_deadline** を使用すると、以下のいずれかの時間を短縮できます。

- エラーのあるパスのシャットオフ
- パスの切り替え
- RAID スライスの無効化



#### 警告

**eh\_deadline** が過ぎると、SCSI EH は HBA をリセットします。これは、エラーが発生しているものだけでなく、HBA 上のすべてのターゲットパスに影響します。一部の冗長パスがその他の理由により利用できない場合は、I/O エラーが発生する可能性があります。**eh\_deadline** は、すべてのターゲットに完全な冗長マルチパス設定がある場合にのみ有効にします。

**eh\_deadline** パラメーターの値は秒単位で指定されます。デフォルト設定は **off** で、時間制限が無効になり、すべてのエラー復旧が行われるようになります。

#### **eh\_deadline** が便利なシナリオ

多くの場合、**eh\_deadline** を有効にする必要はありません。**eh\_deadline** を使用すると、特定のシナリオで役立つ場合があります。たとえば、ファイバーチャネル (FC) スイッチとターゲットポート間でリンクが失われ、HBA が Registered State Change Notifications (RSCN) を受信しない場合などです。このような場合、I/O 要求やエラーからの復旧コマンドは、エラーに遭遇することなく、すべてタイムアウトになります。この環境で **eh\_deadline** を設定すると、リカバリー時間に上限が課せられます。これにより、DM Multipath により、利用できる別のパスで不具合の発生した I/O の再試行が可能になります。

以下の条件下では、**eh\_deadline** パラメーターは、これ以上のメリットをもたらしません。その理由は、DM Multipath の再試行を可能にする I/O とエラー復旧コマンドがすぐに失敗するためです。

- RSCN が有効になっている場合
- HBA が利用できなくなっているリンクを登録しない場合

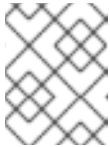
### 10.2. EH\_DEADLINE パラメーターの設定

この手順では、SCSI を復旧する最大時間を制限する **eh-daedline** パラメーターの値を設定します。

## 手順

- **eh\_deadline** は、以下のいずれかの方法で設定できます。
  - **multipath.conf** ファイルの **defaults** セクション  
**multipath.conf** ファイルの defaults セクションから、**eh\_deadline** パラメーターを必要な秒数に設定します。

```
# eh_deadline 300
```



### 注記

RHEL 8.4 以降、**multipath.conf** ファイルの defaults セクションを使用して **eh\_deadline** パラメーターを設定することが推奨されます。

このメソッドで **eh\_deadline** パラメーターをオフにするには、**eh\_deadline** を **off** に設定します。

- **sysfs**  
**/sys/class/scsi\_host/host<host-number>/eh\_deadline** ファイルに秒数を書き込みます。たとえば、SCSI ホスト 6 の **sysfs** を介して **eh\_deadline** パラメーターを設定するには、次のようにします。

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

このメソッドで **eh\_deadline** パラメーターをオフにするには、**echo off** を使用します。

- カーネルパラメーター  
すべての SCSI HBA のデフォルト値は **scsi\_mod.eh\_deadline** カーネルパラメーターを使用して設定します。

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

このメソッドで **eh\_deadline** パラメーターをオフにするには、**echo -1** を使用します。

## 関連情報

- [How to set eh\\_deadline and eh\\_timeout persistently, using a udev rule](#)