



Hewlett Packard
Enterprise

インテル® Virtual RAID on CPU for HPE Gen10 Plusユーザーガイド、Linuxエディション

部品番号: 30-24B8ADE7-001_ja-JP

発行: 2021年6月

版数: 1

インテル® Virtual RAID on CPU for HPE Gen10 Plusユーザーガイド、Linuxエディション

摘要

このドキュメントでは、機能、インストール、および構成に関する情報について説明します。このガイドは、サーバーおよびストレージシステムのインストール、管理、トラブルシューティングを行う担当者を対象としています。コンピューター機器の保守の資格があり、高電圧製品の危険性について理解していることを前提としています。

部品番号: 30-24B8ADE7-001_ja-JP

発行: 2021年6月

版数: 1

© Copyright 2021 Hewlett Packard Enterprise Development LP

ご注意

本書の内容は、将来予告なしに変更されることがあります。Hewlett Packard Enterprise製品およびサービスに対する保証については、当該製品およびサービスの保証規定書に記載されています。本書のいかなる内容も、新たな保証を追加するものではありません。本書の内容につきましては万全を期しておりますが、本書中の技術的あるいは校正上の誤り、脱落に対して、責任を負いかねますのでご了承ください。

本書で取り扱っているコンピューターソフトウェアは秘密情報であり、その保有、使用、または複製には、Hewlett Packard Enterprise から使用許諾を得る必要があります。FAR 12.211 および 12.212 に従って、商業用コンピューターソフトウェア、コンピューターソフトウェアドキュメンテーション、および商業用製品の技術データ (Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items) は、ベンダー標準の商業用使用許諾のもとで、米国政府に使用許諾が付与されます。

他社の Web サイトへのリンクは、Hewlett Packard Enterprise の Web サイトの外に移動します。Hewlett Packard Enterprise は、Hewlett Packard Enterprise の Web サイト以外の情報を管理する権限を持たず、また責任を負いません。

商標

Intel®、Itanium®、Optane™、Pentium®、Xeon®、Intel Inside®、インテル®VMD、およびIntel Insideロゴは、インテルコーポレーションまたはその子会社のアメリカ合衆国およびその他の国/地域における商標または登録商標です。

AMDおよびAMD EPYC™、ならびにこれらの組み合わせは、Advanced Micro Devices, Incの商標です。

Microsoft®およびWindows®は、米国および/またはその他の国/地域におけるMicrosoft Corporationの登録商標または商標です。

Adobe®およびAcrobat®は、米国Adobe Systems Incorporatedの登録商標です。

Java®およびOracle®は、Oracleおよび/またはその関連会社の登録商標です。

UNIX®は、The Open Groupの登録商標です。

VMware® ESXi™とVMware vSphere®は、VMware, Inc. およびその子会社の米国および各国での登録商標または商標です。

Linux ®は、Linus Torvaldsの米国およびその他の国/地域における登録商標です。

すべてのサードパーティのマークは、それぞれの所有者に帰属します。

目次

1 はじめに

- 1.1 サポートされているRAIDボリューム
 - 1.1.1 インテルマトリクスストレージマネージャー
- 1.2 インテルVROC RAID Write Hole Closure

2 BIOS HIIでインテルVMDを有効にしてRAIDを作成/削除する

- 2.1 BIOS/プラットフォーム構成 (RBSU) でSATAまたはsSATAのインテルVROC (SATA RAID) を有効にする
- 2.2 BIOS/プラットフォーム構成 (RBSU) でインテルVMDおよびVROCを有効にする
- 2.3 RBSUでインテルVROC (SATA RAID) 用のRAIDボリュームを作成する
- 2.4 RBSUでインテルVROC (SATA RAID) 用のRAIDボリュームを削除するI
- 2.5 RBSUでインテルVROC (VMD NVMe RAID) 用のRAIDボリュームを作成する
- 2.6 RBSUでインテルVROC (VMD NVMe RAID) 用のRAIDボリュームを削除する
- 2.7 インテル® Virtual RAID on CPUドライバーのダウンロード

3 インテルVROC SEDマネージャー

- 3.1 セキュリティと暗号化の構成をセットアップする
- 3.2 システムキーを交換する
- 3.3 セキュリティと暗号化の構成をクリーンアップする
- 3.4 ドライブを管理する

4 Linux用インテルVROCのインストール

- 4.1 サポートされているLinux OSディストリビューション
- 4.2 インストールファイル

5 mdadmクイックスタートガイド

6 mdadmを使用したRAIDボリュームの作成/削除

- 6.1 システムBIOSからのRAIDの詳細のレポート
- 6.2 RAIDボリュームの作成
- 6.3 追加のRAIDコンテナとボリューム作成の例
- 6.4 スペアの追加
- 6.5 RAID構成ファイルの作成
- 6.6 RAIDボリュームの初期化/再同期
 - 6.6.1 初期化/再同期速度の調整
- 6.7 mdadmを使用したRAIDボリュームの削除
- 6.8 マトリクス内のアレイの削除
- 6.9 RAIDメタデータの消去

7 RAIDのレポートと監視

- 7.1 RAID情報のレポート
- 7.2 ログイン
- 7.3 mdmon
- 7.4 mdadmとsystemctlを使用した監視
 - 7.4.1 mdadm監視デーモン
 - 7.4.2 RAID監視用のsystemctl
- 7.5 監視用構成ファイル
- 7.6 syslogで監視されるイベントの例

8 mdadmによる追加のボリューム操作

- 8.1 ボリュームアセンブリ
- 8.2 RAIDボリュームでのファイルシステムの作成
- 8.3 オンライン容量拡張
- 8.4 ドライブを障害として削除する
- 8.5 RAIDレベルの移行

- 8.6 再形成のフリーズ
- 8.7 RAID書き込みホールの保護の例
- 8.8 RAIDボリュームのリカバリ
 - 8.8.1 障害ドライブの取り外し
 - 8.8.2 再構築

9 Ubuntu LinuxでのVROC

10 MDRAID sysfsコンポーネント

11 Webサイト


12 サポートと他のリソース

- 12.1 Hewlett Packard Enterpriseサポートへのアクセス
- 12.2 アップデートへのアクセス
- 12.3 リモートサポート (HPE通報サービス)
- 12.4 保証情報
- 12.5 規定に関する情報
- 12.6 ドキュメントに関するご意見、ご指摘


はじめに

このドキュメントの目的は、インテルボリューム管理デバイス（インテルVMD）コントローラーによって管理される NVMeドライブ上のインテル® Virtual RAID on CPU (Intel VROC) RAIDのほか、Linuxオペレーティングシステム用の SATAまたはsSATAコントローラーに接続されたSATAドライブ上のインテルVROC RAIDボリュームをユーザーが適切にセットアップ、構成、および管理できるようにすることです。Linux OS内で、インテルVROC RAIDを管理するための主要な構成ソフトウェアは、Linux上のインテルVROCとともに使用するネイティブLinuxツールであるmdadmアプリケーションです。

このドキュメントでは、このアプリケーションの使用方法和、インテルVROC RAIDをセットアップ、管理、および監視するために提供されているさまざまなオプションについて説明します。

 注記: このドキュメントの情報は、サポートされているオペレーティングシステムによるサポートされているインテルチップセットを搭載したシステムにのみ関連しています。お使いのシステムとオペレーティングシステムがインテルVROCをサポートするように適切に構成されていることを確認してください。

 注記: このドキュメントのほとんどの情報は、ソフトウェア構成に関連しています。

 注記: このユーザーガイドはmdadmユーザーガイドではなく、インテルVROCをサポートするために使用されるコマンドにのみ焦点を当てています。

お客様は、特定のハードウェアまたはソフトウェア構成に関するサポートについて質問がある場合、常に購入場所またはシステム/ソフトウェアの製造元に問い合わせる必要があります。

インテルVROCは、エンタープライズサーバー用のNVMe SSDとSATAデバイスの両方にエンタープライズRAIDサポートを提供します。

1. インテルVROC (VMD NVMe RAID) - この製品は、インテルVMDテクノロジーをサポートするプラットフォームでエンタープライズRAIDソリューションを提供します。
2. インテルVROC (SATA RAID) - この製品は、RAIDモード用に構成されたインテルPlatform Control Hub (PCH) であるSATA/sSATAに接続されたSATAデバイスに、エンタープライズRAIDソリューションを提供します。

サポートされているRAIDボリューム

Linux用Intel VROCは、次のRAIDレベルをサポートしています。

- RAID 0
- RAID 1
- RAID 5
- RAID 10

インテルマトリクスストレージマネージャー

インテルVROCパッケージは、高性能NVMe RAID機能を提供し、RAIDアレイとスパンボリュームを作成できます。マトリクスアレイは、単一のRAIDコンテナに最大2つのRAIDボリュームがあることを示しています。

たとえば、インテルC620-Aチップセットを搭載したシステムでは、インテルマトリクスストレージマネージャー (IMSM) により、RAID 0ボリュームのほか、4台のドライブにまたがるRAID 5ボリュームの両方を作成できます。マトリクスRAIDの重要な要件は、コンテナ内のボリュームが同じメンバードライブセットにまたがることです。IMSMメタデータによってコンテナを作成する場合、`mdadm "-e imsm"`オプションを使用します。

IMSMメタデータのもう1つの機能は、LinuxとMicrosoft Windows*ホストシステム間でアレイをローミングする機能です。

インテルVROC RAID Write Hole Closure

インテルVROC製品ファミリーは、RAID 5構成でRAID書き込みホールのシナリオを解決する機能をサポートします。これは、インテルXeonスケーラブルプラットフォーム上のインテルVROCに適用されます。

RAID書き込みホール (RWH) は、パリティベースのRAIDに関連する障害シナリオです。これは、電源障害/クラッシュとドライブ障害 (ストリップ書き込みや完全なドライブのクラッシュなど) が同時に、または互いにきわめて近くで発生した場合に発生します。

残念ながら、これらのシステムクラッシュとディスク障害は関連イベントです。これは、パリティベースのRAIDのメンバーディスク間における書き込み操作の原子性の欠如により、サイレントデータ破壊や回復不能データにつながる可能性があります。原子性の欠如により、電源障害時のアクティブストライプのパリティが不正で、残りのストリップデータと一致しなくなる可能性があります。このような整合性のないストライプ上のデータには必要な保護がなく、さらに悪いことに、誤った修正 (サイレントデータエラー) につながる可能性があります。

RAID書き込みホール状況に対処するために実装された以前のインテルVROCメカニズムには、ダーティストライプジャーナリングと部分パリティログの組み合わせが含まれていました。この実装では、RAID書き込みホールを部分的にしか解決していませんでした。インテルVROC製品ファミリーでは、含まれているRWHソリューションが、この状況を完全に解決します (RWHクローズメカニズムが有効にされている場合)。

BIOS HIIでインテルVMDを有効にしてRAIDを作成/削除する

BIOS/プラットフォーム構成 (RBSU) でSATAまたはsSATAのインテルVROC (SATA RAID) を有効にする

手順

1. POSTの直後に、BIOSセットアップメニューにアクセスするオプションを選択します。この例ではF9キーを使用します。
2. 矢印キーを使用してカーソルを移動してメニューオプションシステム構成を強調表示し、Enterキーを押します。
3. System Configurationメニューで、矢印キーを使用してカーソルを移動してメニューオプションBIOS/Platform Configuration (RBSU)を強調表示し、Enterキーを押します。
4. BIOS/Platform Configuration (RBSU)メニューで、矢印キーを使用してカーソルを移動してメニューオプションStorage Optionsを強調表示し、Enterキーを押します。
5. Storage Optionsメニューで、矢印キーを使用してカーソルを移動してメニューオプションSATA Controller Optionsを強調表示し、Enterキーを押します。
6. SATA controller optionsメニューで、矢印キーを使用してカーソルを移動してメニューオプションEmbedded SATA configurationを強調表示し、Enterキーを押します。
7. ドロップダウンリストから、Intel VROC SATA supportを選択します。
8. F12キーを押して、変更を保存してシステムを再起動します。

BIOS/プラットフォーム構成（RBSU）でインテルVMDおよびVROCを有効にする

手順

1. F9キーを押してセットアップメニューに入ります。
2. 矢印キーを使用してカーソルを移動してメニューオプションSystem Configurationを強調表示し、Enterキーを押します。
3. System Configurationメニューで、矢印キーを使用してカーソルを移動してメニューオプションBIOS/Platform Configuration (RBSU)を強調表示し、Enterキーを押します。
4. BIOS/Platform Configuration (RBSU)メニューで、矢印キーを使用してカーソルを移動してメニューオプションStorage Optionsを強調表示し、Enterキーを押します。
5. Storage Optionsメニューで、矢印キーを使用してカーソルを移動してメニューオプションNVM express optionを強調表示し、Enterキーを押します。
6. NVM express Optionsメニューで、矢印キーを使用してカーソルを移動してメニューオプションIntel (R) NVMe optionを強調表示し、Enterキーを押します。
7. Intel (R) NVMe optionメニューで、矢印キーを使用してカーソルを移動してメニューオプションIntel (R) CPU VMD supportを強調表示します。
8. ドロップダウンリストから、すべてのCPU NVMeルートポートが有効を選択します。
9. 矢印キーを使用してカーソルを移動してメニューオプションIntel (R) VROC supportを強調表示します。ドロップダウンリストから、HPE VROC For HPE NVMsを選択します。
10. F12キーを押して、変更を保存してシステムを再起動します。

RBSUでインテルVROC (SATA RAID) 用のRAIDボリュームを作成する

インテルVROCテクノロジーと互換性のある機器を製造しているベンダーは数多くあります。適切な構成については、付属の資料と次のガイド資料を参照してください。

前提条件

- RAIDボリュームを作成するための適切なサイズとタイプのドライブが十分にあることを確認する。
- ベンダーの仕様に従ってドライブがシステムに正しく接続されていることを確認する。
- プレOSですべてのドライブを認識できることを確認する。

手順

1. セットアップメニューに入るには、起動中に画面に示される適切なキーを押します。この例ではF9キーを使用します。
2. 矢印キーを使用してカーソルを移動してメニューオプションSystem Configurationを強調表示し、Enterキーを押します。
3. Intel (R) VROC SATA/sSATA Controllerに移動し、Enterキーを押します。
4. Create RAID Volumeに移動し、Enterキーを押します。
5. ボリューム名を変更する場合は、[Name]を強調表示してEnterキーを押します。デフォルトを維持する場合は、次の項目に移動します。
6. [RAID Level] (デフォルトではRAID 0 (Stripe)に設定されています) を強調表示し、Enterキーを押します。
7. アレイに含めるドライブを選択するには、識別されたドライブの横にある空の角カッコを強調表示し、Enterキーを押し、Xを強調表示します。Enterキーを押し、選択内容をアレイの一部として確定します。
8. アレイに含める必要があるドライブごとに手順7を繰り返します。RAID 1を除いて、Strip Sizeでアレイのストリップサイズを変更できます。
9. 容量の値はメガバイト単位です。現在の値を強調表示してEnterキーを押し、それをメガバイト単位の希望する量に置き換えてから、Enterキーを押し保存します。
10. Create Volumeに移動し、Enterキーを押します。
11. Yesに移動し、Enterキーを押します。

RBSUでインテルVROC (SATA RAID) 用のRAIDボリュームを削除するI

手順

1. システムの起動中に、適切なキーを押してセットアップメニューに入ります。この例ではF9キーを使用します。
2. System Configurationに移動し、Enterキーを押します。
3. Intel (R) VROC SATA/sSATA Controllerに移動し、Enterキーを押します。
4. リストされたRAID Volumesに移動し、Enterキーを押します。
5. Deleteに移動し、Enterキーを押します。
6. Yesに移動し、Enterキーを押してRAIDの削除を確認します。

RBSUでインテルVROC (VMD NVMe RAID) 用のRAIDボリュームを作成する

インテルVROCテクノロジーと互換性のある機器を製造しているベンダーは数多くあります。適切な構成については、付属の資料と次のガイド資料を参照してください。

前提条件

- システムBIOS内でインテルVMDを有効にする。
- RAIDボリュームを作成するための適切なサイズとタイプのドライブが十分にあることを確認する。
- ベンダーの仕様に従ってドライブがシステムに正しく接続されていることを確認する。
- プレOSですべてのドライブを認識できることを確認する。

手順

1. セットアップメニューに入るには、起動中に画面に示される適切なキーを押します。この例ではF9キーを使用します。
2. 矢印キーを使用してカーソルを移動してメニューオプションSystem Configurationを強調表示し、Enterキーを押します。
3. Intel (R) Virtual RAID on CPUに移動し、Enterキーを押します。
4. 矢印キーを使用してカーソルを移動してメニューオプションAll Intel VMD Controllersを強調表示し、Enterキーを押します。
5. 矢印キーを使用してカーソルを移動してメニューオプションCreate RAID Volumeを強調表示し、Enterキーを押します。
6. ボリューム名を変更する場合は、[Name]を強調表示してEnterキーを押します。デフォルトを維持する場合は、次の項目に移動します。
7. RAID Level (デフォルトではRAID 0 (Stripe)に設定されています) を強調表示し、Enterキーを押します。
8. RAID spanned over VMD Controllerを有効にするには、空の角カッコを強調表示し、Enterキーを押します。Xに移動し、Enterキーを押して、ステータスをEnabledに設定します。
9. アレイに含めるドライブを選択するには、識別されたドライブの横にある空の角カッコを強調表示し、Enterキーを押し、Xを強調表示します。Enterキーを押して、選択内容をアレイの一部として確定します。
10. アレイに含める必要があるドライブごとに手順9を繰り返します。RAID 1を除いて、Strip Sizeでアレイのストリップサイズを変更できます。
11. 容量の値はメガバイト単位です。現在の値を強調表示してEnterキーを押し、それをメガバイト単位の希望する量に置き換えてから、Enterキーを押して保存します。
12. Create Volumeに移動し、Enterキーを押します。
13. Yesに移動し、Enterキーを押します。

RBSUでインテルVROC (VMD NVMe RAID) 用のRAIDボリュームを削除する

手順

1. システムの起動中に、適切なキーを押してセットアップメニューに入ります。この例ではF9キーを使用します。
2. System Configurationに移動し、Enterキーを押します。
3. Intel(R) Virtual RAID on CPUに移動し、Enterキーを押します。
4. リストされたRAID Volumesに移動し、Enterキーを押します。
5. Deleteに移動し、Enterキーを押します。
6. Yesに移動し、Enterキーを押してRAIDの削除を確認します。

インテル® Virtual RAID on CPUドライバーのダウンロード

表1: RHEL7およびSLES12用インテル® Virtual RAID on CPU Gen10 Plus SW RAIDドライバーSmartコンポーネント

ドライバー	直接ダウンロードリンク
RHEL7、バージョン1.0.0.1467 rev1	https://www.hpe.com/global/swpublishing/MTX-f69e55353f1842ecaa0550881d
SLES12、バージョン1.0.0.1488 rev1	https://www.hpe.com/global/swpublishing/MTX-2472c6e0eea34820a32233c85f

インテルVROC SEDマネージャー

インテルVROCはSEDリモートキー管理をサポートしており、外部キーマネージャーを使用してiLOからドライブごとにシステムキーを生成できます。次の手順を実行して、システムをセットアップし、システムキーを交換し、システムを元に戻します。

セキュリティと暗号化の構成をセットアップする

前提条件

- RBSUでインテルVMDとVROCを有効にする。
- OPALを備えたNVMeドライブが少なくとも1つ、有効なインテルVMDコントローラーに接続されていることを確認する。
- iLOキーマネージャーページでリモートキー管理を構成する。詳しくは、次のWebサイトにあるHPE iLOユーザーガイドを参照してください：<https://support.hpe.com/hpesc>。

手順

1. POSTの直後に、BIOSセットアップメニューにアクセスするオプションを選択します。この例ではF9キーを使用します。
2. 矢印キーを使用してカーソルを移動してメニューオプションシステム構成を強調表示し、Enterキーを押します。
3. System Configurationメニューで、矢印キーを使用してカーソルを移動してメニューオプションIntel (R) VROC SED Managerを強調表示し、Enterキーを押します。
4. Intel (R) VROC SED - Dashboard Viewメニューで、矢印キーを使用してカーソルを移動してメニューオプションSystem Setupを強調表示し、Enterキーを押します。
5. Execute System Setupオプションに移動し、Enterキーを押します。

システムキーを交換する

前提条件

- RBSUでインテルVMDとVROCを有効にする。
- OPALを備えたNVMeドライブが少なくとも1つ、有効なインテルVMDコントローラーに接続されている。
- iLOキーマネージャーページでリモートキー管理を構成する。詳しくは、次のWebサイトにあるHPE iLOユーザーガイドを参照してください：<https://support.hpe.com/hpesc>。

手順

1. POSTの直後に、BIOSセットアップメニューにアクセスするオプションを選択します。この例ではF9キーを使用します。
2. 矢印キーを使用してカーソルを移動してメニューオプションSystem Configurationを強調表示し、Enterキーを押します。
3. System Configurationメニューで、矢印キーを使用してカーソルを移動してメニューオプションIntel (R) VROC SED Managerを強調表示し、Enterキーを押します。
4. Intel (R) VROC SED - Dashboard Viewメニューで、矢印キーを使用してカーソルを移動してメニューオプションDrive Managementを強調表示し、Enterキーを押します。
5. Execute System Re-Keyオプションに移動し、Enterキーを押します。

セキュリティと暗号化の構成をクリーンアップする

前提条件

- RBSUでインテルVMDとVROCを有効にする。
- OPALを備えたNVMeドライブが少なくとも1つ、有効なインテルVMDコントローラーに接続されている。
- iLOキーマネージャーページでリモートキー管理を構成する。詳しくは、次のWebサイトにあるHPE iLOユーザーガイドを参照してください：<https://support.hpe.com/hpesc>。

手順

1. POSTの直後に、BIOSセットアップメニューにアクセスするオプションを選択します。この例ではF9キーを使用します。
2. 矢印キーを使用してカーソルを移動してメニューオプションSystem Configurationを強調表示し、Enterキーを押します。
3. System Configurationメニューで、矢印キーを使用してカーソルを移動してメニューオプションIntel (R) VROC SED Managerを強調表示し、Enterキーを押します。
4. Intel (R) VROC SED - Dashboard Viewメニューで、矢印キーを使用してカーソルを移動してメニューオプションDrive Managementを強調表示し、Enterキーを押します。
5. Confirmをクリックし、Execute System Revertオプションに移動して、Enterキーを押します。

ドライブを管理する

前提条件

- RBSUでインテルVMDとVROCを有効にする。
- OPALを備えたNVMeドライブが少なくとも1つ、有効なインテルVMDコントローラーに接続されている。
- iLOキーマネージャーページでリモートキー管理を構成する。詳しくは、次のWebサイトにあるHPE iLOユーザーガイドを参照してください：<https://support.hpe.com/hpesc>。

手順

1. POSTの直後に、BIOSセットアップメニューにアクセスするオプションを選択します。この例ではF9キーを使用します。
2. 矢印キーを使用してカーソルを移動してメニューオプションSystem Configurationを強調表示し、Enterキーを押します。
3. System Configurationメニューで、矢印キーを使用してカーソルを移動してメニューオプションIntel (R) VROC SED Managerを強調表示し、Enterキーを押します。
4. Intel (R) VROC SED - Dashboard Viewメニューで、矢印キーを使用してカーソルを移動してメニューオプションDrive Managementを強調表示し、Enterキーを押します。
5. ドライブを選択し、Enterキーを押します。
ドライブの詳細とステータスが表示されます。

Linux用インテルVROCのインストール

次の手順は、インテルXeonスケーラブルプラットフォームサーバークラスのプロセッサの一部であるインテルVMDを組み込むインテルVROC 7.x機能を備えた、カスタムコンポーネントを含むアーカイブをインストールする機会をユーザーに提供します。

サポートされているLinux OSディストリビューション

VROC Linuxでサポートされている構成については、

<https://www.intel.com/content/www/us/en/support/articles/000030310/memory-and-storage/ssd-software.html>を参照してください。

インストールファイル

Intel VROC Linuxリリース戦略は、Linuxコミュニティのアップストリームであり、OSVパートナーと緊密に連携して、可能であればリリースパッケージをインボックスにします。VROCサポートインボックス情報の詳細については、Intel VROC MoWを参照してください。Intel Xeonプラットフォーム用のIntel VROC 6.x Linuxリリースパッケージは、サポートされている最新のOSVディストリビューションに含まれています。Intel VROC 7.xには、一部のサポートされている古いOSディストリビューション用のアップデートLinuxパッケージがある場合があります。

mdadmクイックスタートガイド

使用法: #mdadm [mode] <raiddevice> [options] <component-devices>

Assemble - 以前に作成したアレイをアクティブなアレイにアセンブルします

```
# mdadm -A -s構成ファイルに基づいて、すべてのRAIDボリュームをスキャンし、アセンブルします
# mdadm -A /dev/md/ims0 -e imsm /dev/<member drives> 構成ファイルがない場合、手動でims0コンテナをアセンブルします
# mdadm -A /dev/md/md0 /dev/md/ims0構成ファイルがない場合、md0アレイを手動でアセンブルします
```

Create/grow/misc - メタデータで新しいアレイを作成します/アクティブなデバイスのアクティブなサイズまたは数を変更します

```
# mdadm -S -s# mdadm -C /dev/md/ims0 /dev/nvme[0-3]n1 -n 4 -e imsm 4台のnvme0-3ドライブを含むims0コンテナを作成します
# mdadm -C /dev/md/md0 /dev/md/ims0 -n 4 -l 5 imsm0コンテナにmd0を4DR5として作成します
# mdadm -C /dev/md/md0 /dev/md/ims0 -n 4 -l 5 --consistency-policy=pp1 RWHを有効にしてims0コンテナにmd0を4DR5として作成します
# mdadm -C /dev/md/md0 /dev/md/ims0 -n 4 -l 5 -z $(100*1024*1024) 各ドライブに100GBのサイズの4DR5を作成します。このコマンド例では300GBのサイズが作成されます
# mkfs.ext4 /dev/md/md0 md0 RAIDボリュームにext4ファイルシステムを作成します
# mount /dev/md/md0 /mnt/<mountpoint> FSにmd0 RAIDボリュームをマウントします
# mdadm -a /dev/md/ims0 /dev/<nvmeXn1> imsm0コンテナにホットスペアドライブを追加します
# mdadm -Ebs > /etc/mdadm.confデバイスメタデータをmdadm.confファイルにエクスポートします
# mdadm -S /dev/md/md0 md0アレイを停止して非アクティブ化します
# mdadm -S /dev/md/ims0 imsm0コンテナを停止して非アクティブ化します
# mdadm -S -sすべてを停止して非アクティブ化します
# mdadm -f /dev/md/md0 /dev/nvme0n1 md0アレイのnvme0n1ドライブを障害とします
# mdadm -r /dev/md/ims0 /dev/nvme0n1 imsm0コンテナのnvme0n1ドライブをデタッチします
# mdadm -zero-superblock /dev/nvme0n1 nvme0n1ドライブのスーパーブロックをクリアしてゼロ化します
# mdadm -G /dev/md/md0 -l 0既存の2DR1を1DR0に拡張します
# mdadm -G /dev/md/ims0 -n 2 2DR0をコンテナに拡張します
# mdadm -a /dev/md/ims0 /dev/nvme2n1コンテナにドライブをもう1つ追加します
# mdadm -G /dev/md/md0 -l 5 --layout=left-asymmetric 2DR0を3DR5に拡張します
# mdadm -G /dev/md/md0 --size=128Gコンポーネントサイズmd0を128GBに拡張します
# mdadm -G /dev/md/md0 --consistency-policy=resync実行中のmd0でRWHクローズポリシーを再同期に切り替えます
# mdadm -G /dev/md/md0 --consistency-policy=pp1実行中のmd0でRWHクローズメカニズムを有効にします
```

ボリューム、メンバー、ハードウェア情報

```
# mdadm --detail-platformプラットフォームのRAID機能の詳細を出力します
# mdadm -D /dev/md/md0 md0デバイスの詳細を出力します
# mdadm -E /dev/nvme0n1 nvme0n1のメタデータストアの内容を出力します
# cat /proc/mdstatすべてのアクティブなmdデバイスを情報とともにリストします
```

Follow/monitor/misc - 1つ以上のデバイスの状態の変化を監視します

```
# systemctl status mdmonitor.service mdmonitorサービスをチェックします
# systemctl start mdmonitor.service mdmonitorサービスを開始します
# systemctl stop mdmonitor.service mdmonitorサービスを停止します
# systemctl enable mdmonitor.service mdmonitorサービスの次の起動時の自動開始を有効にします
```

mdadmを使用したRAIDボリュームの作成/削除

RAIDボリュームは、インテルVMDを有効にした後に、Intel VROC統合UEFI HII機能を使用して、または-e imsmメタデータオプションで指定した場合にインテルマトリクスストレージマネージャー (IMSM) メタデータ形式をサポートするmdadmコマンドラインユーティリティを使用して、BIOS内で作成できます。

システムBIOSからのRAIDの詳細のレポート

mdadmを使用してRAIDを作成する前に、ユーザーは“detail-platform”を使用して、BIOSによって提供されているインテルVROC RAIDのサポートをチェックできます。

```
# mdadm --detail-platform
Platform : Intel(R) Virtual RAID on CPU
Version : 6.0.0.1024
RAID Levels : raid0 raid1 raid10 raid5
Chunk Sizes : 4k 8k 16k 32k 64k 128k
2TB volumes : supported
2TB disks : supported
Max Disks : 8
Max Volumes : 2 per array, 8 per controller
I/O Controller : /sys/devices/pci0000:00/0000:00:11.5 (SATA)
Port0 : - no device attached -
Port1 : - no device attached -
Port2 : - no device attached -
Port3 : - no device attached -
Port4 : - no device attached -
Port5 : - no device attached -

Platform : Intel(R) Virtual RAID on CPU
Version : 6.0.0.1024
RAID Levels : raid0 raid1 raid10 raid5
Chunk Sizes : 4k 8k 16k 32k 64k 128k
2TB volumes : supported
2TB disks : supported
Max Disks : 8
Max Volumes : 2 per array, 8 per controller
I/O Controller : /sys/devices/pci0000:00/0000:17.0 (SATA)
Port4 : /dev/sda (W629KMP2)
Port0 : - no device attached -
Port1 : - no device attached -
Port2 : - no device attached -
Port3 : - no device attached -
Port5 : - no device attached -
Port6 : - no device attached -
Port7 : - no device attached -

Platform : Intel(R) Virtual RAID on CPU
Version : 6.0.0.1024
RAID Levels : raid0 raid1 raid10 raid5
Chunk Sizes : 4k 8k 16k 32k 64k 128k
2TB volumes : supported
2TB disks : supported
Max Disks : 24
Max Volumes : 2 per array, 24 per controller
3rd Party NVMe : supported
I/O Controller : /sys/devices/pci0000:85/0000:85:05.5 (VMD)
I/O Controller : /sys/devices/pci0000:17/0000:17:05.5 (VMD)
I/O Controller : /sys/devices/pci0000:5d/0000:5d:05.5 (VMD)
NVMe under VMD : /sys/devices/pci0000:5d/0000:5d:05.5/pci10001:00/10001:00:02.0/10001:01:00.0
NVMe under VMD : /sys/devices/pci0000:5d/0000:5d:05.5/pci10001:5d.05.5/pci10001:00:03.0/10001:02:00.0
I/O Controller : /sys/devices/pci0000:d7/0000:d7:05.5 (VMD)
NVMe under VMD : /sys/devices/pci0000:d7/0000:d7:05.5/pci10004:00/10004:00:00.0/10004:01:00.0
NVMe under VMD : /sys/devices/pci0000:d7/0000:d7:05.5/pci10004:00:01.0/10004:02:00.0
I/O Controller : /sys/devices/pci0000:ae/05.5 (VMD)
```



注記:

この出力に示されているバージョンは、プラットフォームのインテルVROC PreOSのバージョンであり、使用されているmdadmのバージョンと混同しないでください。

RAIDボリュームの作成

⚠ 警告: RAIDアレイを作成すると、選択したドライブ上の既存のデータが完全に削除されます。これらのステップを開始する前に、すべての重要なデータをバックアップしてください。

以下は、4台のインテルNVMe SSDでRAID5アレイを作成する方法の例を示しています。

1. まず、インテルIMSMメタデータを確立するコンテナを作成する必要があります。

```
# mdadm -C /dev/md/ims0 /dev/nvme[0-3]n1 -n 4 -e imsm
Continue creating array? Y
mdadm: container /dev/md/ims0 prepared.
```

このコマンドは、インテルマトリクスストレージマネージャーメタデータ形式でRAIDコンテナを作成します。コンテナのデバイスノードは `/dev/md/ims0` になります。この例では、ドライブ `nvme0n1`、`nvme1n1`、`nvme2n1`、および `nvme3n1` がこのRAIDコンテナに使用され、ドライブの合計数は4です。ワイルドカード表現 `/dev/nvme[0-3]n1` を使用して、ドライブの範囲を指定できます。個別のドライブも使用できます。ワイルドカード表現はシェルによって解析されています。複数の単一の数字で構成される位置パラメーターの場合は、中括弧で囲む必要があります。たとえば、`/dev/nvme{9,10}n1` などです。

2. 次に、RAID 5ボリュームが作成されます。

```
# mdadm -C /dev/md/md0 /dev/md/ims0 -n 4 -l 5
```

このコマンドは、`/dev/md/ims0` コンテナ内にRAID 5ボリューム `/dev/md0` を作成します。

次のコマンドパラメーターを使用して、RAIDボリュームの作成をより細かく制御することもできます。

`-n` ボリュームで使用されるアクティブなRAIDデバイスの数。

`-c` チャンク（ストリップ）サイズをキロバイト単位で指定します。デフォルトは128KiBです。この値は4KiBの倍数でなければなりません。予想されるワークロードプロファイルに応じて、最適なチャンクサイズを検討する必要があります。

`-l` RAIDレベルを指定します。サポートされるオプションは0、1、5、10です。

`-z` 各ディスク上のRAIDボリューム専用にするスペースのサイズ（キロバイト単位）を指定します。サイズはチャンクサイズの倍数でなければなりません。例：

```
# mdadm -C /dev/md/md0 /dev/md/ims0 -n 4 -l 5 -z $((100*1024*1024))
```

上記のコマンドは、各ドライブメンバーで使用される100GiBのディスク領域がある `/dev/md/ims0` コンテナ内にRAIDボリュームを作成します。

接尾辞 'M' または 'G' を指定して、それぞれMiBiバイトまたはGiBiバイトを示すことができます。これは `-c` パラメーターにも当てはまります。そのため、上記のコマンドはこれと同等になります。

```
# mdadm -C /dev/md/md0 /dev/md/ims0 -n 4 -l 5 -z 100G
```

追加のRAIDコンテナとボリューム作成の例

2ドライブコンテナとRAID 0ボリュームを作成します。

```
# mdadm -C /dev/md/ism0 /dev/nvme[0-1]n1 -n 2 -e imsm
# mdadm -C /dev/md/md0 /dev/md/ism0 -n 2 -l 0
```

2ドライブコンテナとRAID 1ボリュームを作成します。

```
# mdadm -C /dev/md/ism0 /dev/nvme[0-1]n1 -n 2 -e imsm
# mdadm -C /dev/md/md0 /dev/md/ism0 -n 2 -l 1
```

3ドライブコンテナとRAID 5ボリュームを作成します。

```
# mdadm -C /dev/md/ism0 /dev/nvme[0-2]n1 -n 3 -e imsm
# mdadm -C /dev/md/md0 /dev/md/ism0 -n 3 -l 5
```

4ドライブコンテナとRAID 10ボリュームを作成します。

```
# mdadm -C /dev/md/ism0 /dev/nvme[0-3]n1 -n 4 -e imsm
# mdadm -C /dev/md/md0 /dev/md/ism0 -n 4 -l 10
```

注記: 同じコンテナ内に複数のRAIDボリュームを作成するには、それらが同じ数のドライブにまたがる必要があります。たとえば、同じコンテナにRAID 0ボリュームとRAID 5ボリュームを含めるには、両方のボリュームで4台のドライブを使用します。

スペアの追加

スペアドライブを追加すると、デバイスの障害が検出されたときにRAIDボリュームをすぐに再構築できます。mdadmは、障害が発生したデバイスを「不良」とマークし、最初に使用可能なスペアドライブで再構築を開始します。スペアドライブは、RAIDボリュームの拡張にも使用できます。スペアドライブは、通常の操作中はアイドル状態です。ismmメタデータでmdadmを使用すると、コンテナに追加されたスペアドライブはその特定のコンテナ専用になります。次のコマンドは、指定されたコンテナ/dev/md/ismm0にスペアドライブを追加します。

```
# mdadm -a /dev/md/ismm0 /dev/<nvmeXn1>
```

注記:

スペアとしてのコンテナ内のメンバードライブのメタデータUUIDは、ドライブがRAIDボリュームの一部として再割り当てされるまで、すべてゼロのまま維持されます。システムの再起動後に、UUIDがゼロのメンバードライブが別のコンテナに割り当てられる場合があります。

RAID構成ファイルの作成

構成ファイルを作成して、既存のRAIDボリュームを記録できます。既存のRAIDセットアップから情報を抽出できます。構成ファイルは通常、`/etc/mdadm.conf`のデフォルトの場所に保存されます。これにより、プラットフォームの起動後に、`/dev/md`内の整合性のあるアセンブルリンクが可能になります。この信頼できる整合性のあるリンクをシステム（自動マウント用の`/etc/fstab`など）で使用できます。`/dev`内のアセンブルノードは、システムの列挙に応じて変わる場合があります。

```
# mdadm -Ebs > /etc/mdadm.conf
```

OSディストリビューションによっては、構成ファイルの別の保存手段がある場合があります。お使いのディストリビューションに適用するものを判断するには、`mdadm.conf`マンページを参照することが重要です。

RAIDボリュームの初期化/再同期

RAIDレベルが1、10、または5の場合、RAIDボリュームが作成された直後に初期化（または再同期）が開始されます。この期間に、RAIDレベル5ボリュームに保存されたデータは、障害が発生した場合に安全であることが保証されません。初期化期間中にドライブ障害が発生した場合、リカバリができません。このシナリオは、RAIDボリュームの再構築時にも当てはまります。

初期化/再同期速度の調整

デフォルトで、Linuxでは、ボリュームアセンブリと同期の速度として200MBの速度を使用しています。これは、回転プラットフォームベースのハードドライブに基づいて設定されたコミュニティ標準でした。テクノロジーの進歩に伴った標準の移動先に関して、合意に達していません。


システムでRAIDアレイの初期化や再構築に過度の時間がかかるのを防ぐために、この値を変更することができます。以下の一連のコマンドは、デフォルト値に設定されているシステムと、その変更方法を示しています。

システムが使用している現在の速度を確認するには：

```
# cat /sys/block/md0/md/sync_speed_max
200000 (system)
```

これを、NVMeドライブに対して5GBなど、高い値に変更するには、次のようにします。

```
# echo 5000000 > /sys/block/md0/md/sync_speed_max
```

 注記：ドライブタイプによって、適切な値は異なる場合があります。データセンターのインテルNVMe SSDモデルには5GBが妥当です。

mdadmを使用したRAIDボリュームの削除

mdadmを使用してRAIDアレイを構築できますが、mdadmによってそれらを削除することもできます。

まず、アレイを停止します。すべてのアクティブなRAIDボリュームを停止するには、次のコマンドを使用できます。mdadmは、実行中のすべてのRAIDボリュームとコンテナをスキャンして停止します。

```
# mdadm -vS /dev/md/md0
mdadm: stopped /dev/md/md0
```

```
# mdadm -S -s
```

さらにコンテナを停止するには、次のコマンドを使用できます。

```
# mdadm -S /dev/md/ism0
```

ここでアレイのステータスをチェックします。

```
# mdadm -D /dev/md/md0
mdadm cannot open /dev/md/md0: No such file or directory
```

次に、次のコマンドで、関連するすべてのディスクからスーパーブロックを削除します。

```
# mdadm --zero-superblock /dev/nvmeXn1 /dev/nvmeYn1 /dev/nvmeZn1
```

マトリクス内のアレイの削除

コンテナ内に複数のアレイがある場合、これはマトリクスRAIDと呼ばれます。これは、削除の実装で、もう少し注意する必要があります。

まず、両方のボリュームを停止する必要があります。

```
# mdadm -S /dev/md/md{numberForFirstArray}{number}
# mdadm -S /dev/md/md{numberForSecondArray}
```

次に、ボリュームメタデータを削除します。

```
# mdadm --kill-subarray={index} /dev/md{container_number}
```

ここで、`{index}` を0または1のいずれかにして、1番目または2番目のボリュームを表す必要があります。

RAIDメタデータの消去

不正または不良メタデータがあると、RAIDボリュームが正しくアSEMBルされない可能性があります。次のコマンドを使用して、潜在的な各メンバードライブでメタデータを消去し、ドライブを確実にクリーンにすることができます。この操作では、既存のユーザーデータは消去されませんが、現在のメンバードライブまたはスペアドライブである場合にアレイが削除されます。消去操作を実行するには、RAIDボリュームを停止して非アクティブ化する必要があります。

```
# mdadm --zero-superblock /dev/<nvmeXn1>
```

複数のドライブを指定して、同時にスーパーブロックをクリアできます。

RAID情報のレポート

RAIDコンテナまたはボリュームに関する詳細を出力するには、次のコマンドを使用します。

```
# mdadm -D /dev/md/ims0
/dev/md/ims0:
    Version : imsm
    Raid Level : container
    Total Devices : 4
Working Devices : 4
    UUID : b559b502:b199f86f:ee9fbd40:cd10e91d
Member Arrays :
    Number   Major   Minor   Raid Device
    0         8       32      -           /dev/nvme0n1
    1         8       48      -           /dev/nvme1n1
    2         8       80      -           /dev/nvme2n1
    3         8       96      -           /dev/nvme3n1
```

コマンド-Dは、現在の状態のアレイについてカーネルが認識しているすべてを出力します。

```
# mdadm -D /dev/md/md0
/dev/md0:
    Container : /dev/md/ims0, member 0
    Raid Level : raid1
    Array Size : 781408256 (745.21 GiB 800.16 GB)
    Used Dev Size : 781409024 (745.21 GiB 800.16 GB)
    Raid Devices : 2
    Total Devices : 2

    State : clean, resyncing
Active Devices : 2
Working Devices : 2
Failed Devices : 0
Spare Devices : 0

Resync Status : 38% complete
    UUID : 084d2b20:09897744:36757c5b:77e0e945
    Number   Major   Minor   Raid Device State
    0         259     2       0       active sync /dev/nvme1n1
    1         259     1       1       active sync /dev/nvme0n1
```

メンバードライブに関するメタデータ状態の詳細を出力するには:

```
# mdadm -E /dev/nvme0n1
/dev/nvme1n1:

Magic : Intel Raid ISM Cfg Sig.
Version : 1.1.00
Orig Family : f957b05e
Family : f957b05e
Generation : 00000015
Attributes : All supported
UUID : 5f85c04a:be746f31:67d16085:cff460f1
Checksum : abb8c53b correct
MPB Sectors : 1
Disks : 2
RAID Devices : 1
Disk00 Serial : LJ7423090T1P0FGN
State : active
Id : 00000000
Usable Size : 1953514766 (931.51 GiB 1000.20 GB)
[1]:
UUID : f196d4f7:81aea65f:f7391cae:ddfd67a7
RAID Level : 1
```

```
Members : 2
Slots : [UU]
Failed disk : none
This Slot : 0
Sector Size : 512
Array Size : 33554432 (16.00 GiB 17.18 GB)
Per Dev Size : 33556480 (16.00 GiB 17.18 GB)
Sector Offset : 0
Num Stripes : 131072
Chunk Size : 64 KiB
Reserved : 0
Migrate State : idle
Map State : normal
Dirty State : clean
RWH Policy : off
Disk01 Serial : FT5462000H1P6JGN
State : active
Id : 00000000
Usable Size : 3125617166 (1490.41 GiB 1600.32 GB)
```

すべてのRAIDボリュームの最新のステータスを取得するには、ファイル `/proc/mdstat` を調べることができます。このファイルは、継続的にアップデートされ、すべてのコンテナとRAIDボリュームのステータスを示す特殊なファイルです。以下の例では、ステータスに現在使用可能なサポートされているRAIDレベルが0、1、5、および10であることが示されています。Md0は、RAIDレベル5および128kストリップサイズのアクティブなRAIDボリュームです。RAIDボリュームには、すべて正常（UP）ステータスの4台のドライブが含まれています。“md127”は、RAIDボリュームのIMSMコンテナです。

```
# cat /proc/mdstat
Personalities : [raid1] [raid6] [raid5] [raid4]
md0 : active raid5 nvme0n1[3] nvme3n1[2] nvme2n1[1] nvme1n1[0]
      2930270208 blocks super external:/md127/0 level 5, 128k chunk, algorithm 0 [4/4] [UUUU]
md127 : inactive nvme3n1[3](S) nvme2n1[2](S) nvme1n1[1](S) nvme0n1[0](S)
      4420 blocks super external:imsm
unused devices: <none>
```

注記:

コンテナとボリュームを作成するときに、`/proc/mdstat` で、デバイス名が正確に一致しないことに気付くでしょう。たとえば、`/dev/md/imsm0`を作成すると、`/proc/mdstat` とその他の詳細にもmd127が表示されます。`/dev/md/imsm0` は `/dev/md127` デバイスノードのエイリアスとして作成されます。`/dev/md` ディレクトリを調べると、`/dev/md/imsm0` が `/dev/md127` へのソフトリンクであることに気付きます。

ロギング

カーネル内のMDRAIDサブシステムから送られるさまざまなメッセージがログに記録されます。通常、人気のあるLinuxディストリビューションで、メッセージは、その他のカーネルステータス、警告、およびエラー出力とともにログファイル `/var/log/messages` に保存されます。

以下は、ログがどのように見えるかのスニペットの例です。

```
Dec 11 15:07:28 localhost kernel: md/raid:md0: not clean -- starting background reconstruction
Dec 11 15:07:28 localhost kernel: md/raid:md0: device nvme0n1 operational as raid disk 3
Dec 11 15:07:28 localhost kernel: md/raid:md0: device nvme3n1 operational as raid disk 2
Dec 11 15:07:28 localhost kernel: md/raid:md0: device nvme2n1 operational as raid disk 1
Dec 11 15:07:28 localhost kernel: md/raid:md0: device nvme1n1 operational as raid disk 0
Dec 11 15:07:28 localhost kernel: md/raid:md0: raid level 5 active with 4 out of 4 devices, algorithm 0
Dec 11 15:07:28 localhost kernel: md/raid456: discard support disabled due to uncertainty.
Dec 11 15:07:28 localhost kernel: Set raid456.devices_handle_discard_safely=Y to override.
Dec 11 15:07:28 localhost kernel: md0: detected capacity change from 0 to 3000596692992
Dec 11 15:07:28 localhost systemd: Starting MD Metadata Monitor on /dev/md127...
Dec 11 15:07:28 localhost systemd: Started MD Metadata Monitor on /dev/md127.
Dec 11 15:07:28 localhost kernel: md: resync of RAID array md0
```

mdadmツール内には、RAIDボリュームのイベントを監視するための2つのコンポーネントがあります。Mdadm (-monitorパラメーターを付けて) を使用して、一般的なRAIDイベントを監視でき、mdmonは、外部メタデータベースのRAIDボリュームのドライブ障害、クリーンからダーティへの遷移など、「メタデータイベント」の発生を監視する機能を提供します。カーネルは、sysfsを介してそのようなアクションをユーザー空間に報告する機能を提供し、mdmonは監視機能によって、それに応じてアクションを実行します。mdmonは、sysfsをポーリングして、array_state、sync_action、およびドライブごとの状態属性ファイルのエントリ値の変更を探します。

mdmon

mdadmモニターmdmonは、mdadmで作成またはアSEMBルによってMDRAIDボリュームがアクティブ化されると、自動的に開始されます。mdmonは、メタデータマネージャーおよびアップデーターです。デーモンは、次のように手動で開始できます。

次のコマンドを実行します。

```
# mdmon /dev/md0
```

--allパラメーターをコンテナ名の代わりに使用して、すべてのアクティブなコンテナのモニターを開始できます。

注記:

外部メタデータRAIDアレイをルートファイルシステムとしてサポートするには、mdmonをinitramfsで起動する必要があります。最後のルートファイルシステムがマウントされたら、新しい名前空間でmdmonを再起動する必要があります。デーモンを手動で起動または再起動することはお勧めしません。

```
# mdmon --takeover --all
```

mdadmとsystemctlを使用した監視

RHEL 7.Xの新規インストールで設計されたすべての監視サービスは、RAIDボリュームの作成時に初期化されますが、これらの機能が正しく初期化されたことを確認する方法があります。これらのツールにより、Linux管理者は、CLI環境内でRAID監視を支援するために使用するツールを、システムでチェックして有効にすることができます。

mdadm監視デーモン

mdadmの監視は、次のコマンドラインで開始できます。

```
# mdadm --monitor --scan --daemonise --syslog
```

上記のコマンドは、mdadmをデーモンとして実行して、すべてのmdデバイスを監視します。すべてのイベントはsyslogに報告されます。ユーザーは、syslogを監視し、生成された特定のmdadmイベントをフィルター処理できます。監視を開始する前に、mdadm.confファイルにメールアドレスを定義することが必須です。

RAID監視用のsystemctl

mdmonitorサービスが期待通りに実行しているかどうかを確認するには、次のコマンドを使用してステータスをチェックできます。

```
# systemctl status mdmonitor.service
```

サービスが実行していないことが分かった場合は、以下のコマンドを使用して手動でサービスを開始できます。

```
# systemctl start mdmonitor.service
```

または、各再起動後にすぐに開始するようにサービスを有効にすることができます。次のコマンドを使用します。

```
# systemctl enable mdmonitor.service
```

サービスを停止するには、このコマンドを使用します。

```
# systemctl stop mdmonitor.service
```

次の表は、起動時にmdadmに渡すことができる追加のコマンドラインパラメーターを示しています。

表2: モニターモードのmdadmのパラメーター

長形式	短形式	説明
<code>--mail</code>	<code>-m</code>	アラートまたは障害のメール送信先のメールアドレスを指定します。
<code>--program</code> または <code>--alert</code>	<code>-p</code>	イベントが検出されたときに実行するプログラムを指定します。
<code>--delay</code>	<code>-d</code>	状態のポーリング間の遅延の秒数。デフォルトは60秒です。
<code>--config</code>	<code>-c</code>	別の構成ファイルを指定します。
<code>--scan</code>	<code>-s</code>	構成ファイルでメールアドレス/プログラム設定を見つけます。
<code>--oneshot</code>	<code>-l</code>	縮退したアレイをチェックしてから終了します。
<code>--test</code>	<code>-t</code>	起動時に各アレイに対してテストメッセージイベントを生成します。
<code>--syslog</code>	<code>-y</code>	'syslog'を通じてすべてのイベントを報告させます。メッセージには、「デーモン」の機能とさまざまな優先順位があります。
<code>--increment</code>	<code>-r</code>	パーセントの増分を指定します。mdadmはRebuildNNイベントを生成します。NNは再構築イベントが発生したパーセンテージを示します。
<code>--daemonise</code>	<code>-f</code>	監視中の場合、バックグラウンドデーモンとして実行します。
<code>--pid-file</code>	<code>-i</code>	デーモンプロセスのpidを指定したファイルに書き込みます。
<code>--no-sharing</code>	該当なし	これは、アレイ間でスペアドライブを移動する機能を抑制します。

注記:

mdmonitor.serviceまたはmdadm.confファイルが変更された場合、mdmonitorサービスを再起動する必要があります。

次の表は、mdadmモニターによって報告されるすべてのイベントを示しています。

表3: イベント

イベント名	説明
DeviceDisappeared	以前に構成されたMDアレイが存在しません。
RebuildStarted	MDアレイが再構築を開始しました。

イベント名	説明
RebuildNN	NNは2桁の数字で、再構築が全体の何パーセントを超えたことを示します。たとえば、Rebuild50は、再構築の50%が完了したときにイベントをトリガーします。
RebuildFinished	MDアレイの再構築が完了しました。
Fail *	アレイのアクティブなコンポーネントが障害とマークされました。
FailSpare *	障害のあるデバイスを交換するために再構築されていたスペアドライブに障害が発生しました。
SpareActive	障害のあるデバイスを交換するために再構築されていたスペアドライブが再構築され、アクティブです。
NewArray	/proc/mdstatで新しいMDアレイが検出されました。
DegradedArray *	新しく検出されたアレイが縮退しているように見えます。
MoveSpare	障害ドライブを交換するために、スペアドライブがスペアグループ内のあるアレイから別のアレイに移動されました。両方のアレイに、同じスペアグループのラベルが付けられます。
SparesMissing *	MDアレイが最初に検出されたときに、構成ファイルと比較して、スペアデバイスが存在しません。
TestMessage *	--testフラグが使用されているときに、新しいアレイが検出されました。



注記:

示されたイベントにより、メールが送信されます。これらはsyslogに送信することもできます。

監視用構成ファイル

mdadmは、mdadm.conf構成ファイルをチェックして、監視対象の適切なエントリを抽出します。mdmonに渡すために設定できるエントリ:

- MAILADDR : この構成エントリにより、アラートにメールアドレスを使用できます。使用できるメールアドレスは1つだけです。
- MAILFROM : この構成エントリは、アラートメールの送信元に表示されるメールアドレスを設定します。デフォルトの送信元は、ドメインのない「ルート」ユーザーです。このエントリはデフォルトを上書きします。
- PROGRAM : この構成エントリは、mdmonが監視しているアレイのいずれかで関心が高そうなイベントを検出したときに、プログラムが実行するように設定します。構成ファイルにはPROGRAM行を1つだけ含めることができます。

syslogで監視されるイベントの例

```
Personalities : [raid5]

md0 : active raid5 nvme2n1[2] nvme1n1[1] nvme0n1[0]
      204800 blocks super external:/md127/0 level 5, 128k chunk, algorithm 0 [3/3] [UUU]

md127 : inactive - nvme2n1[2](S) nvme1n1[1](S) nvme0n1[0](S)
       3315 blocks super external:imsm
unused devices: <none>
```

すべてのRAIDコンテナを監視するには、次のコマンドを使用してmdadmデーモンを起動できます。

```
# mdadm --monitor --scan --daemonise -syslog
```

注記:

同時に実行できる監視デーモンは1つだけです。ユーザーは、新しいデーモンを実行する前に、既存のデーモンを強制終了する必要があります。

注記:

RHELベースのシステムでは、mdmonitorサービスを実行するために、ユーザーがmdadm構成ファイルにMAILADDRを指定する必要があることを確認します（簡単な例: #echo "MAILADDR root" >> /etc/mdadm.conf）。

すべてのイベントがsyslogに書き込まれるようになります。mdadmデーモンが起動されると、/var/log/messagesまたはディストリビューションで指定された対応するsyslogファイルに次のメッセージが表示されます。

```
May 15 09:58:40 myhost mdadm[9863]: NewArray event detected on md device /dev/md0
May 15 09:58:40 myhost mdadm[9863]: NewArray event detected on md device /dev/md127
```

スペアドライブが追加されている場合:

```
May 15 09:59:07 myhost mdadm[9863]: SpareActive event detected on md device /dev/md0, component device /dev/<nvmeXn1>
```

OLCEコマンドが終了したとき:

```
May 15 09:59:16 myhost mdadm[9863]: RebuildFinished event detected on md device /dev/md0
```

ドライブに障害が発生した場合:

```
May 15 10:01:04 myhost mdadm[9863]: Fail event detected on md device /dev/md0, component device /dev/<nvmeXn1>
```

再構築が完了したとき:

```
May 15 10:02:22 myhost mdadm[9863]: RebuildFinished event detected on md device /dev/md0
May 15 10:02:22 myhost mdadm[9863]: SpareActive event detected on md device /dev/md0
```

すべてのMDデバイスが停止している場合:

```
May 15 10:03:27 myhost mdadm[9863]: DeviceDisappeared event detected on md device /dev/md0
May 15 10:03:27 myhost mdadm[9863]: DeviceDisappeared event detected on md device /dev/md127
```


ボリュームアセンブリ

RAIDボリュームは、mdadmアプリケーションを使用して、サポートされているオペレーティングシステムを備えたサポートされているシステムでアセンブルすることができます。すべての非アクティブなRAIDボリュームは、mdadmとアセンブルオプションを使用してアクティブ化できます。

次のコマンドは、RAIDボリュームをアセンブルするために、`/etc/mdadm.conf`にあるmdadm構成ファイルをスキャンします。構成ファイルが見つからない場合、RAIDメンバーに使用可能なすべてのドライブをスキャンし、すべてのRAIDボリュームをアセンブルします。さらに、`/dev/md`リンクは、各ボリュームにサフィックスとして“_0”を追加します。

```
# mdadm -A -s
```

構成ファイルを使用せずにRAIDボリュームを手動でアセンブルしてアクティブ化するには、次の例を使用できます。

```
# mdadm -A /dev/md/md0 -e imsm /dev/<member drives>
```

このコマンドは、指定されたドライブのリストを使用して、名前 `/dev/md/md0` でコンテナをアセンブルします。

```
# mdadm -A /dev/md/md0 /dev/md/imsm0
```

このコマンドは、コンテナ `/dev/md/imsm0` を使用して、RAIDボリューム `/dev/md/md0` をアセンブルします。

mdadm.confの詳細については、Linuxのマニュアルを参照してください。

RAIDボリュームでのファイルシステムの作成

RAIDボリュームを作成した後、RAIDボリュームをマウントできるようにするためにファイルシステムを作成できます。

```
# mkfs.ext4 /dev/md/md0
```

ファイルシステムが作成されたら、それを選択した場所にマウントできます。

```
# mount /dev/md/md0 /mnt/<mountpoint>
```

オンライン容量拡張

オンライン容量拡張 (OLCE) 機能により、RAIDボリュームの容量を拡張できます。「オンライン」機能により、RAIDボリューム上にファイルシステムをマウントした状態で操作を実行できます。これにより、サービスのためにRAIDボリュームをオフラインにすることからのダウンタイムやデータの損失が避けられます。

RAIDボリュームのサイズを増やすには、RAIDコンテナにドライブを追加するか、または（それがコンテナ内の最後のボリュームである場合のみ）RAIDボリュームに使用可能な既存の未使用のドライブ領域でそれを拡張します。最初のケースでは、2つのボリュームが同じコンテナに存在する場合、すべてのボリュームがIMSMメタデータの同じディスクセットにまたがる必要があるため、両方のボリューム（1つずつ）に対してOLCEが自動的に実行されます。

次のコマンドを発行して、RAIDボリュームを拡張できます。1つ目は、それがコンテナ内の最後のボリュームであり、拡張する追加スペースがあることを前提としています。2つ目は、追加ドライブがIMSMコンテナに追加されていることを前提としています。

⚠ 警告:

データが変更されていなくても、拡張操作を実行する前にデータを常にバックアップする必要があります。

コンテナの最後のボリュームに追加スペースがある場合は、ボリュームを使用可能な最大容量まで拡張できます。

注記: この機能は、mdadm v3.2.5以降でのみ使用できます。これはMDADM_EXPERIMENTAL環境変数によって保護され、GROWコマンドを成功させるために、それを"1"に設定する必要があります。新しいmdadmリリースでは、この変数は必要ありません。詳細については、mdadm githubを参照してください。

```
# mdadm -G /dev/md/md0 --size=max
```

📄 注記:

RAID 0のサイズ拡張は現在サポートされていません。

以下の例では、RAIDコンテナに単一ドライブを追加してから、ボリュームを拡張しています。コンテナ内のIMSMボリュームは同じ数のドライブにまたがる必要があるため、すべてのボリュームが拡張されます。MDRAIDがバックアップスーパーブロックを格納するバックアップファイルが指定されます。このファイルは、操作中のアクティブなRAIDボリュームに存在してはなりません。

```
# mdadm -a /dev/md/ims0 /dev/<nvmeXn1>
# mdadm -G /dev/md/md0 -n 4 --backup-file=/tmp/backup
```

📄 注記:

この操作は、バリエーションメンバードライブの異なるRAIDレベルに依存する場合があります。

ドライブを障害として削除する

アクティブなドライブを手動で“障害”としてマークする（または障害として設定する）には、基本的に2つのステップを使用して、障害ドライブをボリュームとコンテナから削除します。

最初のステップは、ボリュームからドライブを障害とすることです。障害ドライブはまだコンテナ内に留まりますが、RAIDボリューム内にありません。

```
# mdadm -f /dev/md/md0 /dev/<nvmeXn1>
```

2番目のステップは、障害ドライブをコンテナから削除することです。次のコマンドを実行する必要があります。これは、コンテナベースのRAIDボリュームでのみ機能します。

```
# mdadm -r /dev/md/md127 /dev/<nvmeXn1>
```

注記:

上記のコマンドで論理ドライブがボリュームから削除されたら、ユーザーはシステムから物理NVMe SSDを削除できます。

RAIDレベルの移行

RAIDレベルの移行機能により、ボリュームに保存されているデータを失うことなく、RAIDボリュームレベルを変更できます。オペレーティングシステムの再インストールは必要ありません。すべてのアプリケーションとデータはそのまま残ります。

⚠ 警告:

データが変更されていなくても、移行操作を実行する前にデータを常にバックアップする必要があります。

次の表は、Intel IMSMメタデータで利用可能な移行サポートを示しています。スペアドライブとして変換先のレベルに必要な適切な数のドライブが必要です。

表4: IMSMによる移行機能

移行先/移行元のレベル	RAID 0	RAID 1	RAID 10	RAID 5
RAID 0	該当なし	なし	あり	あり
RAID 1	あり	該当なし	なし	あり*
RAID 10	あり	なし	該当なし	あり*
RAID 5	なし	なし	なし	該当なし

* RAID 1からRAID 5へ、またはRAID 10からRAID 5への移行は、2ステップで行う必要があります。RAID 5に変換する前に、最初にRAID 0に変換する必要があります。2番目のステップ (RAID 0からRAID 5への移行) で、スペアドライブの追加が必要になる場合があります。

何らかの拡張操作を実行する前に (まだ実行されていない場合)、1つの環境変数を設定する必要があります。

```
# export MDADM_EXPERIMENTAL=1
```

📝 注記:

この機能は、mdadm v3.2.5以降でのみ使用できます。この設定は大文字と小文字が区別されます。新しいmdadmリリースでは、この変数は必要ありません。詳細についてはmdadm githubを参照してください。

以下は、2ディスクRAID 1から3ディスクRAID 5への移行の例です。

```
# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4] [raid1]
md0 : active raid1 nvme0n1[1] nvme1n1[0]
      16777216 blocks super external:/md127/0 [2/2] [UU]
md127 : inactive nvme1n1[1] (S) nvme0n1[0] (S)
      2210 blocks super external:imsm
unused devices: <none>
```

最初のステップ: 2ディスクRAID 1から1ディスクRAID 0に移行します。

```
# mdadm -G /dev/md0 -l 0
md/raid1:md0: Disk failure on nvme1n1, disabling device.
md/raid1:md0: Operation continuing on 1devices.
mdadm: level of /dev/md0 changed to raid0
Personalities : [raid1] [raid0] [raid6] [raid5] [raid4]
md0 : active raid0 nvme0n1[1]
      16777216 blocks super external:/md127/0 64k chunks
md127 : inactive nvme1n1[1] (S) nvme0n1[0] (S)
      2210 blocks super external:imsm
unused devices: <none>
```

2番目のステップ: オンライン容量拡張を使用して、1ディスクRAID 0から2ディスクRAID 0に移行します。

```
# mdadm -G /dev/md127 -n 2
# cat /proc/mdstat
Personalities : [raid1] [raid0] [raid6] [raid5] [raid4]
```

```
md0 : active raid4 nvme1n1[2] nvme0n1[1]
      16777216 blocks super external:/md127/0 level 4, 64k chunks algorithm 0 [3/2] [U__]

md127 : inactive nvme1n1[1](S) nvme0n1[0](S)
        2210 blocks super external:imsm

unused devices: <none>
```

📖 注記:

これは中間ステップであり、LinuxはRAID 4をサポートしていますが、Intel VROCによる最終段階のソリューションとしてサポートされていません。次のステップに進んで、データを確実に保存します。

3番目のステップ: コンテナmd127にスペアディスクを追加します。

```
# mdadm -a /dev/md127 /dev/nvme2n1
# cat /proc/mdstat
Personalities : [raid0] [raid1] [raid6] [raid5] [raid4]
md0 : active raid0 nvme1n1[2] nvme0n1[1]
      16777216 blocks super external:/md127/0 64k chunks

md127 : inactive - nvme2n1[2](S) nvme1n1[1](S) nvme0n1[0](S)
        3315 blocks super external:imsm

unused devices: <none>
```

4番目のステップ: 2ディスクRAID 0から3ディスクRAID 5に移行します。

```
# mdadm -G /dev/md0 -l 5 --layout=left-asymmetric
# cat /proc/mdstat
Personalities : [raid0] [raid1] [raid6] [raid5] [raid4]
md0 : active raid5 nvme2n1[3] nvme1n1[2] nvme0n1[1]
      16777216 blocks super external:/md127/0 level 5, 64k chunk, algorithm 0 [3/3] [UUU]

md127 : inactive - nvme2n1[2](S) nvme1n1[1](S) nvme0n1[0](S)
        3315 blocks super external:imsm

unused devices: <none>
```

⚠ 警告:

IMSMメタデータは、RAID 5の左非対称レイアウトのみをサポートします。デフォルトのレイアウトは左対称であるため、移行時にIMSMメタデータのレイアウトを明示的に指定する必要があります。左非対称は、ローテートパリティ0+データ継続の使用です。

📖 注記:

再形成プロセスの進行中に、整合性ポリシーを調整するコマンドは失敗します。メッセージ“mdadm: PPL cannot be enabled when reshape is in progress”が表示されます。

これは、4ディスクRAID 10から4ディスクRAID 5に移行する別の例です。

```
[root@localhost ~]# cat /proc/mdstat
Personalities : [raid1] [raid10]
md1 : active raid10 nvme2n1[3] nvme3n1[2] nvme0n1[1] nvme1n1[0]
      20971520 blocks super external:/md127/0 128K chunks 2 near-copies [4/4] [UUUU]
md127 : inactive nvme3n1[3](S) nvme2n1[2](S) nvme1n1[1](S) nvme0n1[0](S)
        4420 blocks super external:imsm

unused devices: <none>
```

最初のステップは、4ディスクRAID 10を2ディスクRAID 0に移行することです。

```
[root@localhost ~]# mdadm -G /dev/md1 -l 0
mdadm: level of /dev/md1 changed to raid0
[root@localhost ~]# cat /proc/mdstat
Personalities : [raid1] [raid10] [raid0]
```

```
md1 : active raid0 nvme2n1[3] nvme1n1[0]
      20971520 blocks super external:/md127/0 128k chunks
md127 : inactive nvme3n1[3] (S) nvme2n1[2] (S) nvme1n1[1] (S) nvme0n1[0] (S)
      4420 blocks super external:imsm
unused devices: <none>
```

2番目のステップは、2ディスクRAID 0から3ディスクRAID 5に移行することです。

```
[root@localhost ~]# mdadm -G /dev/md1 -l 5 --layout=left-asymmetric
mdadm: level of /dev/md1 changed to raid5
[root@localhost ~]# cat /proc/mdstat
Personalities : [raid1] [raid10] [raid0] [raid6] [raid5] [raid4]
md1 : active raid5 nvme3n1[4] nvme2n1[3] nvme1n1[0]
      20971520 blocks super external:-md127/0 level 5, 128k chunk, algorithm 5 [3/2] [UU_]
      [=>.....] reshape = 13.9% (1458176/10485760) finish=13.9min speed=10768K/sec
md127 : inactive nvme3n1[3] (S) nvme2n1[2] (S) nvme1n1[1] (S) nvme0n1[0] (S)
      4420 blocks super external:imsm
unused devices: <none>
```

3番目のステップは、3ディスクRAID5から4ディスクRAID 5に拡張することです。再同期が再形成の前に完了している必要があります。

```
[root@localhost ~]# mdadm -G /dev/md127 -n 4
[root@localhost ~]# cat /proc/mdstat
Personalities : [raid1] [raid10] [raid0] [raid6] [raid5] [raid4]
md1 : active raid5 nvme0n1[5] nvme3n1[4] nvme2n1[3] nvme1n1[0]
      20971520 blocks super external:-md127/0 level 5, 128k chunk, algorithm 0 [4/4] [UUUU]
      [>.....] reshape = 3.9% (416256/10485760) finish=1.2min speed=138752K/sec
md127 : inactive nvme3n1[3] (S) nvme2n1[2] (S) nvme1n1[1] (S) nvme0n1[0] (S)
      4420 blocks super external:imsm
unused devices: <none>
```


再形成のフリーズ

RAIDボリュームが再形成プロセス中の場合、再形成プロセスはinitramfs起動フェーズでフリーズし、システムが稼働すると再開します。mdadm 3.2.5以降、これらの機能がサポートされています。オペレーティングシステムベンダーからのディストリビューションでは、initスクリプトセットアップユーティリティでこれに配慮する必要がありますが、独自のディストリビューションを構築しているお客様向けに詳細を以下に説明します。

⚠ 警告:

データが変更されていなくても、再形成操作を実行する前にデータを常にバックアップする必要があります。

システムの起動initramfsフェーズ中に再形成操作を一時停止するには、パラメーター `--freeze-reshape` を使用します。例:

```
# mdadm -As --freeze-reshape
```

再形成がフリーズされると、`/proc/mdstat`によって提供されるステータスに、“super external:/md127/0”ではなく“super external:-md127/0”などのハイフン付きで状態が示されます。

```
Personalities : [raid5]
md0 : active raid5 nvme2n1[2] nvme1n1[1] nvme0n1[0]
      204800 blocks super external:-md127/0 level 5, 128k chunk, algorithm 0 [3/3] [UUU]
      [>.....] reshape = 2.2% (116736/5242880) finish=501934.9min speed=0K/sec

md127 : inactive -
      9459 blocks super external:imsm

unused devices: <none>
```

システムが起動したら、`--continue` パラメーターを付けた次の例を使用して、再形成プロセスを再開できます。

```
# mdadm -G /dev/md/md0 --continue
```

またはボリュームを付けて:

```
# mdadm -G /dev/md/Volume0 --continue
```

RAID書き込みホールの保護の例

RAID書き込みホール (RWH) は、RAID 5書き込みホールの略です。システムクラッシュまたはその他の書き込み操作の中断により、書き込みプロセスの非原子性のため、パリティがデータと不整合になる状態が発生することがあります。そのため、ディスク障害が発生した場合にパリティをリカバリに使用できなくなります（いわゆるRAID 5書き込みホール）。RAID書き込みホールは、ディスクへの書き込みのデステージの中断によって発生する、古いローエンドRAIDの既知のデータ破損問題です。

部分パリティログ (PPL) は、RAID 5アレイで使用できる機能です。書き込み操作の部分パリティは、この書き込みによって変更されていないストライプデータチャンクのXORです。書き込みホールからのリカバリに必要なデータはこれだけで十分です。変更されたチャンクと部分パリティのXORにより、どのチャンク書き込みが完了したかに関係なく、書き込み操作前の状態と整合性のあるストライプのパリティが生成されます。このストライプの変更されていないデータディスクの1つが見つからない場合、このアップデートされたパリティを使用してその内容をリカバリできます。PPLリカバリは、正常にシャットダウンされなかった後にアレイを起動し、すべてのディスクが使用可能である場合にも実行され、アレイを再同期する必要がありません。

VROCは、PPLによってRWHクローズをサポートします。アクティブアレイの整合性ポリシーを変更するには、作成モードまたは拡張モードで“consistency-policy”オプションを使用します。現在のmdadmでは、RAID 5部分パリティログを有効または無効にするpplおよびresyncポリシーをサポートしています。

RWHポリシーを有効にしてRAID5ボリュームを作成するには：

```
# mdadm -C /dev/md/ims0 /dev/nvme[0-3]n1 -n 4 -e imsm
Continue creating array? Y
mdadm: container /dev/md/ims0 prepared.
# mdadm -C /dev/md0 /dev/md/ims0 -l5 -n4 --consistency-policy=ppl
```

現在のRWHポリシーをチェックするには：

```
# mdadm -D /dev/md0
/dev/md0:
    Container : /dev/md/ims0, member 0
    Raid Level : raid5
    Array Size : 117212672 (1117.83 GiB 1200.26 GB)
    Used Dev Size : 390708224 (372.61 GiB 400.09 GB)
    Raid Devices : 4
    Total Devices : 4

    State : clean, resyncing
    Active Devices : 4
    Working Devices : 40
    Failed Devices : 0
    Spare Devices : 0

    Layout : left-asymmetric
    Chunk Size: 128K
    Consistency Policy : ppl

    UUID : a0470058:714d8834:f9daaa0:f20a1a1aec
    NumberMajorMinorRaidDeviceState
    025910active sync/dev/nvme2n1
    025901active sync/dev/nvme3n1
    025932active sync/dev/nvme4n1
    025923active sync/dev/nvme1n1
```

RWHクローズポリシーを実行中のアレイの再同期に切り替えるには：

```
# mdadm -G /dev/md/md0 --consistency-policy=resync
```

RWHクローズポリシーを実行中のアレイのpplに切り替えるには：（RWHクローズメカニズム有効）

```
# mdadm -G /dev/md/md0 --consistency-policy=ppl
```

RAIDボリュームのリカバリ

リカバリは、RAIDを使用する上で最も重要な側面の1つです。ドライブ障害が発生したときに、システムに冗長RAIDボリュームを再構築できます。リカバリは、次のRAIDレベル1、5、および10の場合にのみ可能です。一般的なリカバリは、障害が発生したドライブが1台だけの場合に可能です。ただし、RAID 10の場合、4台のドライブのうち2台で障害が発生した場合でも、2台の障害ドライブが2つの異なるミラーリングペアのメンバーである場合に、リカバリが可能です。1つのミラーの両方のドライブに障害が発生した場合は、リカバリできません。

注記:

RAIDレベル0は、リカバリ用に設計されていません。RAIDレベル0の縮退後、mdadmによってメタデータステータスのアップデートが継続されない場合があります。

障害ドライブの取り外し

障害ドライブは、次のコマンドでRAIDボリュームから削除できます。

```
# mdadm -f /dev/md/md0 /dev/nvme0n1
```

次のコマンドでコンテナから削除します。

```
# mdadm -r /dev/md/ism0 /dev/nvme2n1
```

再構築

この時点で、このRAIDボリュームは縮退モードで実行されています。ただし、まだ動作しています。コンテナに使用可能なスペアドライブがある場合、RAIDボリュームの再構築が自動的に開始されます。スペアドライブを手動で追加して、再構築プロセスを開始することもできます。

自動再構築により、ドライブに障害が発生したときにRAIDボリュームが自動的に再構築されます。これが発生する可能性のある3つの異なるシナリオがあります。

1. コンテナ内に1台以上のスペアドライブがある再構築対応RAIDボリュームがあります。ボリューム内のいずれかのドライブに障害が発生した場合、スペアドライブが自動的に取り込まれ、再構築が開始されます。
2. コンテナ内にスペアドライブがない再構築対応RAIDボリュームがあります。ボリューム内のいずれかのドライブに障害が発生した場合、それは縮退モードになります。スペアドライブをコンテナに手動で追加すると、再構築が自動的に開始されます。
3. 2つ以上のコンテナがあります。1つのコンテナにはスペアドライブがあり、もう1つのコンテナにはありません。mdadmが監視モードで実行されており、mdadm.confファイルに適切なポリシーが構成されている場合、RAIDボリュームが縮退し、再構築にスペアドライブが必要になった場合に、スペアドライブがコンテナ間で自動的に移動されます。

シナリオ1：同じコンテナ内のスペアドライブで自動再構築します。

1. 3台のディスクでコンテナ“`imsm0`”を作成します。

```
# mdadm -C /dev/md/imsm0 -n3 -e imsm /dev/nvme[1-3]n1
```

2. コンテナ“`imsm0`”内にRAID1ボリューム“`vol1`”を作成します。ドライブ/`dev/nvme1n1`はスペアのままです。

```
# mdadm -C /dev/md/vol1 /dev/md/imsm0 -l1 -n2
# cat /proc/mdstat
Personalities : [raid1]
md125 : active raid1 nvme3n1[1] nvme2n1[0]
        31457280 blocks super external:/md127/0 [2/2] [UU]
md127 : inactive nvme3n1[2] (S) nvme2n1[1] (S) nvme1n1[0] (S)
        3315 blocks super external:imsm
```

3. ボリューム“`vol1`”内のいずれかのRAIDメンバードライブで障害が発生し、スペアドライブで自動的に再構築が開始されます。

```
# mdadm -f /dev/md/vol1 /dev/nvme3n1
# cat /proc/mdstat
Personalities : [raid1]
md125 : active raid1 nvme1n1[2] nvme2n1[0]
        31457280 blocks super external:/md127/0 [2/1] [U_]
        [=====>.....] recovery = 42.8% (13483520/31457280) finish=0.1min speed=2247253K/sec
md127 : inactive nvme3n1[2] (S) nvme2n1[1] (S) nvme1n1[0] (S)
        3315 blocks super external:imsm
```

シナリオ2：同じコンテナに新しいスペースを追加した後、自動再構築します。

4ディスクRAID 5ボリュームmd0で障害が発生したときに、`imsm0`コンテナにnvme4n1を追加します。nvme4n1が追加されると、自動的に再構築がトリガーされます。

```
# mdadm -add /dev/md/imsm0 /dev/<nvme4n1>

Personalities : [raid1] [raid0] [raid6] [raid5] [raid4]
md0 : active raid5 nvme0n1[4] nvme3n1[2] nvme2n1[1] nvme1n1[0]
        50331648 blocks super external:/md127/0 level 5, 128k chunk, algorithm 0 [4/3] [UUU_]
        [=====>.....] recovery = 26.6% (4471296/16777216) finish=0.6min speed=319378K/sec
md127 : inactive nvme0n1[0] (S) nvme3n1[3] (S) nvme2n1[2] (S) nvme1n1[1] (S)
        8516 blocks super external:imsm
unused devices: <none>Auto Rebuild
```

シナリオ3：他のコンテナ内のスペアドライブで自動再構築します。

1. 3台のディスクのあるコンテナ“`imsm1`”を作成します。

```
# mdadm -C /dev/md/imsm1 -n3 -e imsm /dev/nvme[0-2]n1
```

2. 1. コンテナ“`imsm1`”内にRAID1ボリューム“`md1`”を作成します。ドライブ/`dev/nvme2n1`はスペアのままです。

```
# mdadm -C /dev/md/md1 /dev/md/imsm1 -l1 -n2
```

3. 2台の追加ドライブがあるコンテナ“`imsm2`”を作成します。

```
# mdadm -C /dev/md/imsm2 -n2 -e imsm /dev/nvme[3-4]n1
```

4. スペアドライブなしで、コンテナ“`imsm2`”内に、RAID1ボリューム“`md2`”を作成します。

```
# mdadm -C /dev/md/md2 /dev/md/imsm2 -l1 -n2
Personalities : [raid1] [raid0] [raid6] [raid5] [raid4]
md2 : active raid1 nvme3n1[1] nvme4n1[0]
      976756736 blocks super external:/md127/0 [2/2] [UU]
md127 : inactive nvme4n1[1] (S) nvme3n1[0] (S)
      2210 blocks super external:imsm
md1 : active raid1 nvme2n1[1] nvme1n1[0]
      976756736 blocks super external:/md126/0 [2/2] [UU]
md126 : inactive nvme2n1[2] (S) nvme1n1[1] (S) nvme0n1[0] (S)
      3315 blocks super external:imsm
Unused devices: <none>
```

5. 構成ファイルを保存します。

```
# mdadm -E -s > /etc/mdadm.conf
```

6. すべてのドライブに対して同じドメインと同じアクションのポリシーを構成ファイルに追加します。これにより、スペアをコンテナ間で移動して再構築できます。

```
# echo "POLICY domain=DOMAIN path=* metadata=imsm action=spare-same-slot" >> /etc/mdadm.conf
```

注記:

RHELベースのシステムでは、`mdmonitor`サービスを実行するために、ユーザーが`mdadm`構成ファイルに`MAILADDR`を指定する必要があることを確認してください（簡単な例: `#echo "MAILADDR root" >> /etc/mdadm.conf`）

`/etc/mdadm.conf`内の構成ファイルは次のようになります。

```
ARRAY metadata=imsm UUID=60582f12:51325766:88172a8a:8424eb16 spares=1
ARRAY /dev/md/1 container=60582f12:51325766:88172a8a:8424eb16 member=0 UUID=0467835f:e74cb807:0c7b1fe2:84ccf80c
ARRAY metadata=imsm UUID=a4a1b4a1:23f34684:de07ba5b:328edbee
ARRAY /dev/md/2 container=a4a1b4a1:23f34684:de07ba5b:328edbee member=0 UUID=39964136:f3fd21ae:e3cf676c:8f73b3fa
POLICY domain=DOMAIN path=* metadata=imsm action=spare-same-slot
```

7. `mdadm`が監視モードであることを確認します。

```
# mdadm --monitor --scan --daemonise
```

8. スペアのないボリュームであるボリューム“`md2`”のいずれかのドライブを障害とします。

```
# mdadm -f /dev/md/md2 /dev/nvme3n1
```

スペアドライブ`/dev/nvme0n1`がコンテナ“`md126`”からコンテナ“`md127`”に自動的に移動されるはずであり、“`md2`”の再構築が自動的に開始します。

```
Personalities : [raid1]
md2 : active raid1 nvme0n1[2] nvme4n1[0]
      16777216 blocks super external:/md127/0 [2/1] [U_]
      [=====>.....] recovery = 59.5% (9991424/16777216) finish=0.2min speed=525864K/sec
md127 : inactive nvme0n1[2] (S) nvme4n1[1] (S) nvme3n1[0] (S)
      7411 blocks super external:imsm
md1 : active raid1 nvme2n1[1] nvme1n1[0]
      16777216 blocks super external:/md126/0 [2/2] [UU]
md126 : inactive nvme2n1[2] (S) nvme1n1[1] (S)
      2210 blocks super external:imsm
unused devices: <none>
```

再構築が完了した場合：


```
Personalities : [raid1]
md2 : active raid1 nvme0n1[2] nvme4n1[0]
      16777216 blocks super external:/md127/0 [2/2] [UU]
md127 : inactive nvme0n1[2] (S) nvme4n1[1] (S) nvme3n1[0] (S)
      7411 blocks super external:imsm
md1 : active raid1 nvme2n1[1] nvme1n1[0]
      16777216 blocks super external:/md126/0 [2/2] [UU]
md126 : inactive nvme2n1[2] (S) nvme1n1[1] (S)
      2210 blocks super external:imsm
unused devices: <none>
```

`mdadm`は、`/etc/mdadm.conf`に基づいて適切な`udev`ルールを生成できます。次に例を示します。


```
# mdadm --udev-rules > /etc/udev/rules.d/65-md-bare.rules
#udevadm control --reload
```

カスタムudevルールが追加され、次のようになります。

```
# ACTION=="add", SUBSYSTEM=="block", ENV{ID_PATH}=="*", RUN+="/sbin/mdadm --incremental $env{DEVNAME}"
```

 注記:

このセクション内で説明しているように、正しいポリシー設定を含む構成ファイル（/etc/mdadm.conf）が存在する必要があります。

 注記:

新しいディスクは、少なくとも最初と最後の4KBが空白である「ベア」である必要があります。

Ubuntu LinuxでのVROC

このユーザーガイドは、Red Hat Linux (RHEL) の構成と設定から始めます。このセクションでは、Ubuntu Server LTS でVROCを使用する場合のいくつかの注意点について説明します。

- VROC Linuxは、Ice Lakeプラットフォーム用のUbuntu Server版インボックスをサポートします。ライブISOイメージの代わりにレガシーインストーラーを使用します。レガシーサーバーISOイメージは、次のリンクのUbuntu Webサイトからダウンロードできます。インストーラーリビジョンの詳細については、canonicalで確認してください。

<https://old-releases.ubuntu.com/releases/>

<https://cdimage.ubuntu.com/ubuntu-legacy-server/releases/20.04/release/>

MDRAID sysfsコンポーネント

MDRAIDサブシステムには、情報を提供したり、動作やパフォーマンスを調整するために使用できるsysfsコンポーネントがあります。システムに存在するすべてのMDRAIDデバイスは、次の場所に表示されます。

```
/sys/block/
```

例：

```
# ls -l /sys/block/md*
lrwxrwxrwx 1 root root 0 May 17 13:26 /sys/block/md126 -> ../devices/virtual/block/md126
lrwxrwxrwx 1 root root 0 May 17 13:26 /sys/block/md127 -> ../devices/virtual/block/md127
```

デバイス番号とその名前とのマッピングは次のように参照できます。

```
# ls -l /dev/md/
total 0
lrwxrwxrwx 1 root root 8 May 17 13:26 imsm0 -> ../md127
lrwxrwxrwx 1 root root 8 May 17 13:26 raid1 -> ../md126
```

md127はimsm0で、md126はraid1です。

/sys/block内のMDデバイスは、/sys/devices/virtual/blockを指しているシンボリックリンクです。すべてのMDデバイスは、/sys/devices/virtual/block/mdXYZディレクトリの'md'サブディレクトリにあります。mdディレクトリには、次の内容がある可能性があります。

```
# ls -l /sys/devices/virtual/block/md127/md
total 0
-rw-r--r-- 1 root root 4096 May 18 13:18 array_size
-rw-r--r-- 1 root root 4096 May 17 13:26 array_state
drwxr-xr-x 2 root root 0 May 18 13:18 bitmap
-rw-r--r-- 1 root root 4096 May 18 13:18 chunk_size
-rw-r--r-- 1 root root 4096 May 18 13:18 component_size
drwxr-xr-x 2 root root 0 May 17 13:26 dev-nvme1n1
drwxr-xr-x 2 root root 0 May 17 13:26 dev-nvme2n1
-rw-r--r-- 1 root root 4096 May 18 13:18 layout
-rw-r--r-- 1 root root 4096 May 17 13:26 level
-rw-r--r-- 1 root root 4096 May 18 13:18 max_read_errors
-rw-r--r-- 1 root root 4096 May 17 13:26 metadata_version
--w----- 1 root root 4096 May 17 13:26 new_dev
-rw-r--r-- 1 root root 4096 May 17 13:26 raid_disks
-rw-r--r-- 1 root root 4096 May 18 13:18 reshape_position
-rw-r--r-- 1 root root 4096 May 18 13:18 resync_start
-rw-r--r-- 1 root root 4096 May 18 13:18 safe_mode_delay
```

MDデバイスはコンテナであるため、メタデータバージョンファイルに次のように表示されます。

```
# cat /sys/devices/virtual/block/md127/md/metadata_version
external:imsm
```

このディレクトリには、コンテナのアセンブル元のディスクを指定するサブディレクトリdev-nvme1n1およびdev-nvme2n1が含まれます。

MDボリュームの内容：

```
# ls -l /sys/devices/virtual/block/md126/md/
total 0
-rw-r--r-- 1 root root 4096 May 17 13:26 array_size
-rw-r--r-- 1 root root 4096 May 17 13:26 array_state
drwxr-xr-x 2 root root 0 May 18 13:10 bitmap
--w----- 1 root root 4096 May 18 13:10 bitmap_set_bits
-rw-r--r-- 1 root root 4096 May 17 13:26 chunk_size
-rw-r--r-- 1 root root 4096 May 17 13:26 component_size
-r--r--r-- 1 root root 4096 May 17 13:26 degraded
drwxr-xr-x 2 root root 0 May 17 13:26 dev-nvme1n1
drwxr-xr-x 2 root root 0 May 17 13:26 dev-nvme2n1
-rw-r--r-- 1 root root 4096 May 17 13:26 layout
-rw-r--r-- 1 root root 4096 May 17 13:26 level
```

```

-rw-r--r-- 1 root root 4096 May 18 13:10 max_read_errors
-rw-r--r-- 1 root root 4096 May 17 13:26 metadata_version
-r--r--r-- 1 root root 4096 May 18 13:10 mismatch_cnt
--w----- 1 root root 4096 May 17 13:26 new_dev
-rw-r--r-- 1 root root 4096 May 17 13:26 raid_disks
lrwxrwxrwx 1 root root 0 May 17 13:26 rd0 -> dev-nvme1n1
lrwxrwxrwx 1 root root 0 May 17 13:26 rd1 -> dev-nvme2n1
-rw-r--r-- 1 root root 4096 May 18 13:10 reshape_position
-rw-r--r-- 1 root root 4096 May 17 13:26 resync_start
-rw-r--r-- 1 root root 4096 May 17 13:26 safe_mode_delay
-rw-r--r-- 1 root root 4096 May 18 13:10 suspend_hi
-rw-r--r-- 1 root root 4096 May 18 13:10 suspend_lo
-rw-r--r-- 1 root root 4096 May 17 13:26 sync_action
-r--r--r-- 1 root root 4096 May 17 13:26 sync_completed
-rw-r--r-- 1 root root 4096 May 18 13:10 sync_force_parallel
-rw-r--r-- 1 root root 4096 May 18 13:10 sync_max
-rw-r--r-- 1 root root 4096 May 18 13:10 sync_min
-r--r--r-- 1 root root 4096 May 17 13:26 sync_speed
-rw-r--r-- 1 root root 4096 May 18 13:10 sync_speed_max
-rw-r--r-- 1 root root 4096 May 18 13:10 sync_speed_min

```

いくつかの新しいファイルが存在し、それらはRAIDボリュームのプロパティに関連しています。基本情報をファイルから読み取ることができます。

- **アレイサイズ**

```

# cat /sys/devices/virtual/block/md126/md/array_size
1048576

```

- **アレイの状態**

```

# cat /sys/devices/virtual/block/md126/md/array_state
clean

```

- **Raidレベル**

```

# cat /sys/devices/virtual/block/md126/md/level
raid1

```

- **ストリップサイズ**

```

# cat /sys/devices/virtual/block/md126/md/chunk_size
65536

```

- **メタデータ**

```

# cat /sys/devices/virtual/block/md126/md/metadata_version
external:/md127/0

```

これは、RAID情報の例としてmdstatに表示されるものです。

```

# cat /proc/mdstat
Personalities : [raid1]
md127 : active raid1 nvme0n1[1] nvme1n1[0]
      78148256 blocks super external:/md0/0 [2/2] [UU]

md0 : inactive nvme1n1[1](S) nvme0n1 [0](S)
      2210 blocks super external:imsm

unused devices: <none>

```

Webサイト

一般的なWeb サイト

Single Point of Connectivity Knowledge (SPOCK) のStorage compatibility matrix

<https://www.hpe.com/storage/spock>

ストレージのホワイトペーパーおよび分析レポート

<https://www.hpe.com/storage/whitepapers>

上記以外のWebサイトについては、[サポートと他のリソース](#)を参照してください。

Hewlett Packard Enterpriseサポートへのアクセス

- ライブアシスタンスについては、Contact Hewlett Packard Enterprise WorldwideのWebサイトにアクセスします。

<https://www.hpe.com/info/assistance>

- ドキュメントとサポートサービスにアクセスするには、Hewlett Packard EnterpriseサポートセンターのWebサイトにアクセスします。

<https://www.hpe.com/support/hpesc>

ご用意いただく情報

- テクニカルサポートの登録番号（該当する場合）
- 製品名、モデルまたはバージョン、シリアル番号
- オペレーティングシステム名およびバージョン
- ファームウェアバージョン
- エラーメッセージ
- 製品固有のレポートおよびログ
- アドオン製品またはコンポーネント
- 他社製品またはコンポーネント

アップデートへのアクセス

- 一部のソフトウェア製品では、その製品のインターフェイスを介してソフトウェアアップデートにアクセスするためのメカニズムが提供されます。ご使用の製品のドキュメントで、ソフトウェアの推奨されるソフトウェアアップデート方法を確認してください。
- 製品のアップデートをダウンロードするには、以下のいずれかにアクセスします。

Hewlett Packard Enterpriseサポートセンター

<https://www.hpe.com/support/hpesc>

Hewlett Packard Enterpriseサポートセンター：ソフトウェアのダウンロード

<https://www.hpe.com/support/downloads>

マイHPEソフトウェアセンター

<https://www.hpe.com/software/hpesoftwarecenter>

- eNewslettersおよびアラートをサブスクライブするには、以下にアクセスします。

<https://www.hpe.com/support/e-updates-ja>

- お客様の資格を表示、アップデート、または契約や保証をお客様のプロファイルにリンクするには、Hewlett Packard EnterpriseサポートセンターのMore Information on Access to Support Materialsページに移動します。

<https://www.hpe.com/support/AccessToSupportMaterials>

① 重要:

一部のアップデートにアクセスするには、Hewlett Packard Enterpriseサポートセンターからアクセスするときに製品資格が必要になる場合があります。関連する資格を使ってHPEパスポートをセットアップしておく必要があります。

リモートサポート（HPE通報サービス）

リモートサポートは、お客様の保証またはサポート契約の一部として、サポートされているデバイスで使用できます。これは優れたイベント診断、Hewlett Packard Enterpriseへのハードウェアイベント通知の自動かつ安全な送信を提供します。また、お使いの製品のサービスレベルで高速かつ正確な解決方法を開始します。Hewlett Packard Enterpriseでは、ご使用のデバイスをリモートサポートに登録することを強くお勧めします。

ご使用の製品にリモートサポートの追加詳細情報が含まれる場合は、検索を使用してその情報を見つけてください。

HPE通報サービス

<http://www.hpe.com/jp/hpalert>

HPE Pointnext Tech Care

<https://www.hpe.com/jp/ja/services/tech-care.html>

HPEデータセンターケア

<https://www.hpe.com/jp/ja/services/datacenter-hybrid-services.html>

保証情報

ご使用の製品の保証情報を確認するには、以下のリンクを参照してください。

HPE ProLiantとIA-32サーバーおよびオプション

<https://www.hpe.com/support/ProLiantServers-Warranties>

HPE EnterpriseおよびCloudlineサーバー

<https://www.hpe.com/support/EnterpriseServers-Warranties>

HPEストレージ製品

<https://www.hpe.com/support/Storage-Warranties>

HPEネットワーク製品

<https://www.hpe.com/support/Networking-Warranties>

規定に関する情報

安全、環境、および規定に関する情報については、Hewlett Packard Enterpriseサポートセンターからサーバー、ストレージ、電源、ネットワーク、およびラック製品の安全と準拠に関する情報を参照してください。

<https://www.hpe.com/support/Safety-Compliance-EnterpriseProducts>

規定に関する追加情報

Hewlett Packard Enterpriseは、REACH（欧州議会と欧州理事会の規則EC No 1907/2006）のような法的な要求事項に準拠する必要に応じて、弊社製品の含有化学物質に関する情報をお客様に提供することに全力で取り組んでいます。この製品の含有化学物質情報レポートは、次を参照してください。

<https://www.hpe.com/info/reach>

RoHS、REACHを含むHewlett Packard Enterprise製品の環境と安全に関する情報と準拠のデータについては、次を参照してください。

<https://www.hpe.com/info/ecodata>

社内プログラム、製品のリサイクル、エネルギー効率などのHewlett Packard Enterpriseの環境に関する情報については、次を参照してください。

<https://www.hpe.com/info/environment>

ドキュメントに関するご意見、ご指摘

Hewlett Packard Enterpriseでは、お客様により良いドキュメントを提供するように努めています。ドキュメントの改善に役立てるために、Hewlett Packard Enterpriseサポートセンターポータル (<https://www.hpe.com/support/hpesc>) にあるフィードバックボタンとアイコン（開いているドキュメントの下部にあります）から、エラー、提案、またはコメントを送信いただけます。すべてのドキュメント情報は、プロセスによってキャプチャーされます。