



# Red Hat Ceph Storage 7

## ファイルシステムガイド

Ceph ファイルシステムの設定とマウント



# Red Hat Ceph Storage 7 ファイルシステムガイド

---

Ceph ファイルシステムの設定とマウント

## 法律上の通知

Copyright © 2024 Red Hat, Inc.

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

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

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

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

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

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

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

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

All other trademarks are the property of their respective owners.

## 概要

このガイドでは、Ceph Metadata Server (MDS) の設定方法と、Ceph File System (CephFS) の作成、マウント、作業方法について説明します。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。

## 目次

<b>第1章 CEPH ファイルシステムの紹介</b> .....	<b>4</b>
1.1. CEPH FILE SYSTEM の機能と強化点	4
1.2. CEPH FILE SYSTEM のコンポーネント	5
1.3. CEPH FILE SYSTEM と SELINUX	7
1.4. CEPH FILE SYSTEM の制限と POSIX 規格	7
<b>第2章 CEPH FILE SYSTEM METADATA SERVER</b> .....	<b>9</b>
2.1. METADATA SERVER デーモンの状態	9
2.2. メタデータサーバーのランク	9
2.3. メタデータサーバーのキャッシュサイズ制限	10
2.4. ファイルシステムアフィニティー	11
2.5. CEPH ORCHESTRATOR を使用した MDS サービスの管理	11
2.6. ファイルシステムのアフィニティーを設定する	18
2.7. 複数のアクティブな METADATA SERVER デーモンの設定	20
2.8. スタンバイデーモンの数の設定	21
2.9. STANDBY-REPLAY 用 METADATA SERVER の設定	22
2.10. エフェメラルピニングポリシー	22
2.11. ディレクトリーツリーを特定のランクに手動で固定する	24
2.12. アクティブな METADATA SERVER デーモンの数を減らす方法	25
<b>第3章 CEPH FILE SYSTEM のデプロイメント</b> .....	<b>28</b>
3.1. レイアウト、クォータ、スナップショット、およびネットワークの制限	28
3.2. CEPH ファイルシステムの作成	29
3.3. CEPH ファイルシステムへのイレイジャーコーディングされたプールの追加	33
3.4. CEPH ファイルシステム用のクライアントユーザーの作成	36
3.5. CEPH FILE SYSTEM のカーネルクライアントとしてのマウント	38
3.6. CEPH FILE SYSTEM の FUSE クライアントとしてのマウント	42
<b>第4章 CEPH ファイルシステムボリューム、サブボリュームグループ、およびサブボリュームの管理</b> .....	<b>48</b>
4.1. CEPH FILE SYSTEM ボリューム	48
4.2. CEPH FILE SYSTEM サブボリュームグループ	51
4.3. CEPH FILE SYSTEM サブボリューム	56
4.4. CEPH FILE SYSTEM サブボリュームのメタデータ情報	69
<b>第5章 CEPH FILE SYSTEM 管理</b> .....	<b>73</b>
5.1. CEPHFS-TOP ユーティリティの使用	73
5.2. MDS AUTOSCALER モジュールの使用	77
5.3. カーネルクライアントとしてマウントされた CEPH FILE SYSTEMS のアンマウント	78
5.4. FUSE クライアントとしてマウントされている CEPH FILE SYSTEMS のアンマウント	78
5.5. ディレクトリーツリーから METADATA SERVER デーモンのランクへのマッピング	79
5.6. METADATA SERVER デーモンのランクからディレクトリーツリーの解除	80
5.7. データプールの追加	80
5.8. CEPH FILE SYSTEM クラスターの停止	82
5.9. CEPH ファイルシステムの削除	84
5.10. CEPH MDS FAIL コマンドの使用	85
5.11. クライアント機能	86
5.12. CEPH FILE SYSTEM クライアントのエビクション	88
5.13. ブロックリスト CEPH FILE SYSTEM クライアント	88
5.14. CEPH FILE SYSTEM クライアントの手動エビクト	89
5.15. ブロックリストからの CEPH FILE SYSTEM クライアントの削除	90
<b>第6章 NFS クラスターとエクスポート管理</b> .....	<b>92</b>

6.1. NFS クラスターの作成	92
6.2. NFS 設定のカスタマイズ	93
6.3. NFS プロトコルを介した CEPH FILE SYSTEM 名前空間のエクスポート (可用性に制限あり)	95
6.4. CEPH ファイルシステムのエクスポートの変更	99
6.5. カスタム CEPH ファイルシステムエクスポートの作成	102
6.6. CEPH ファイルシステムのエクスポートの削除	104
6.7. NFS クラスターの削除	104
<b>第7章 CEPH FILE システムのクォータ</b>	<b>106</b>
7.1. CEPH FILE システムのクォータ	106
7.2. クォータの表示	106
7.3. クォータの設定	107
7.4. クォータの削除	108
<b>第8章 ファイルとディレクトリーのレイアウト</b>	<b>110</b>
8.1. ファイルとディレクトリーレイアウトの概要	110
8.2. ファイルとディレクトリーレイアウトフィールドの設定	110
8.3. ファイルとディレクトリーのレイアウトフィールドの表示	111
8.4. 個々のレイアウトフィールドの表示	112
8.5. ディレクトリーレイアウトの削除	113
<b>第9章 CEPH ファイルシステムのスナップショット</b>	<b>115</b>
9.1. CEPH ファイルシステムのスナップショット	115
9.2. CEPH ファイルシステムのスナップショットの作成	115
<b>第10章 CEPH FILE SYSTEM スナップショットのスケジューリング</b>	<b>118</b>
10.1. CEPH ファイルシステムのスナップショットスケジュール	118
10.2. CEPH ファイルシステムのスナップショットスケジュールの追加	119
10.3. CEPH FILE SYSTEM サブボリュームのスナップショットスケジュールの追加	121
10.4. CEPH ファイルシステムのスナップショットスケジュールのアクティブ化	123
10.5. CEPH ファイルシステムサブボリュームのスナップショットスケジュールのアクティブ化	124
10.6. CEPH ファイルシステムのスナップショットスケジュールの非アクティブ化	125
10.7. CEPH ファイルシステムサブボリュームのスナップショットスケジュールの非アクティブ化	125
10.8. CEPH ファイルシステムのスナップショットスケジュールの削除	126
10.9. CEPH ファイルシステムサブボリュームのスナップショットスケジュールの削除	127
10.10. CEPH ファイルシステムのスナップショットスケジュール保持ポリシーの削除	127
10.11. CEPH ファイルシステムサブボリュームのスナップショットスケジュール保持ポリシーの削除	128
<b>第11章 CEPH FILE SYSTEM のミラー</b>	<b>130</b>
11.1. CEPH FILE SYSTEM ミラーリング	130
11.2. CEPH ファイルシステムのスナップショットミラーの設定	131
11.3. CEPH FILE システムのミラーステータスの表示	136
<b>付録A CEPH FILE SYSTEM のヘルスメッセージ</b>	<b>140</b>
<b>付録B METADATA SERVER デーモン設定リファレンス</b>	<b>143</b>
<b>付録C ジャーナル設定の参照</b>	<b>158</b>
<b>付録D CEPH FILE SYSTEM ミラー設定リファレンス</b>	<b>160</b>



# 第1章 CEPH ファイルシステムの紹介

ストレージ管理者として、Ceph File System (CephFS) 環境を管理するための機能、システムコンポーネント、および制限事項について理解することができます。

## 1.1. CEPH FILE SYSTEM の機能と強化点

Ceph File System (CephFS) は、Ceph の RADOS (Reliable Autonomic Distributed Object Storage) と呼ばれる分散オブジェクトストアの上に構築された POSIX 規格と互換性のあるファイルシステムです。CephFS は、Red Hat Ceph Storage クラスタへのファイルアクセスを提供し、可能な限り POSIX セマンティクスを使用します。たとえば、NFS のような他の多くの一般的なネットワークファイルシステムとは対照的に、CephFS はクライアント間で強力なキャッシュコヒーレンシーを維持します。目標は、ファイルシステムを使用するプロセスが、異なるホストに存在するときも、同じホストにいるときも、同じように動作することです。ただし、CephFS は厳密な POSIX セマンティクスから乖離している場合もあります。

Ceph File System には、以下のような機能や強化があります。

### スケーラビリティ

Ceph File System は、メタデータサーバーの水平方向のスケーリングと、個々の OSD ノードでのクライアントの直接の読み書きにより、高いスケーラビリティを実現しています。

### 共有ファイルシステム

Ceph File System は共有ファイルシステムなので、複数のクライアントが同じファイルシステム上で同時に作業することができます。

### 複数のファイルシステム

1つのストレージクラスタで複数のファイルシステムをアクティブにできます。各 CephFS には、独自のプールのセットと、独自のメタデータサーバー (MDS) ランクのセットがあります。複数のファイルシステムをデプロイメントする場合は、MDS デーモンの数を増やす必要があります。これにより、メタデータのスループットを向上させることができますが、同時に運用コストも増加します。また、特定のファイルシステムへのクライアントのアクセスを制限することもできます。

### 高可用性

Ceph File System には、Ceph Metadata Server (MDS) のクラスタが用意されています。1つはアクティブで、他はスタンバイモードです。アクティブなデータシートが不意に終了した場合、スタンバイデータシートの1つがアクティブになります。その結果、サーバーが故障してもクライアントのマウントは継続して動作します。この動作により、Ceph File System は可用性が高くなります。さらに、複数のアクティブなメタデータサーバーを設定することも可能です。

### 設定可能なファイルおよびディレクトリーレイアウト

Ceph File System では、ファイルやディレクトリーのレイアウトを設定して、複数のプール、プールの名前空間、オブジェクト間のファイルストライピングモードを使用することができます。

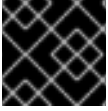
### POSIX アクセスコントロールリスト (ACL)

Ceph File System は POSIX Access Control Lists (ACL) をサポートしています。ACL は、カーネルバージョン **kernel-3.10.0-327.18.2.el7** 以降のカーネルクライアントとしてマウントされた Ceph File Systems でデフォルトで有効になります。FUSE クライアントとしてマウントされた Ceph File Systems で ACL を使用するには、ACL を有効にする必要があります。

### クライアントクォータ

Ceph File System は、システム内のあらゆるディレクトリーにクォータを設定することをサポートしています。クォータは、ディレクトリー階層のそのポイントの下に保存されているバイト数やファイル数を制限することができます。CephFS クライアントクォータはデフォルトで有効です。





## 重要

CephFS EC プールはアーカイブのみを目的としています。

### 関連情報

- Ceph Metadata サーバーをインストールするには、[オペレーションガイドの Ceph Orchestrator を使用した MDS サービスの管理](#) セクションを参照してください。
- Ceph File System を作成するには、[ファイルシステムガイドの Ceph File System の導入](#) セクションを参照してください。

## 1.2. CEPH FILE SYSTEM のコンポーネント

Ceph File System には 2 つの主要コンポーネントがあります。

### Clients

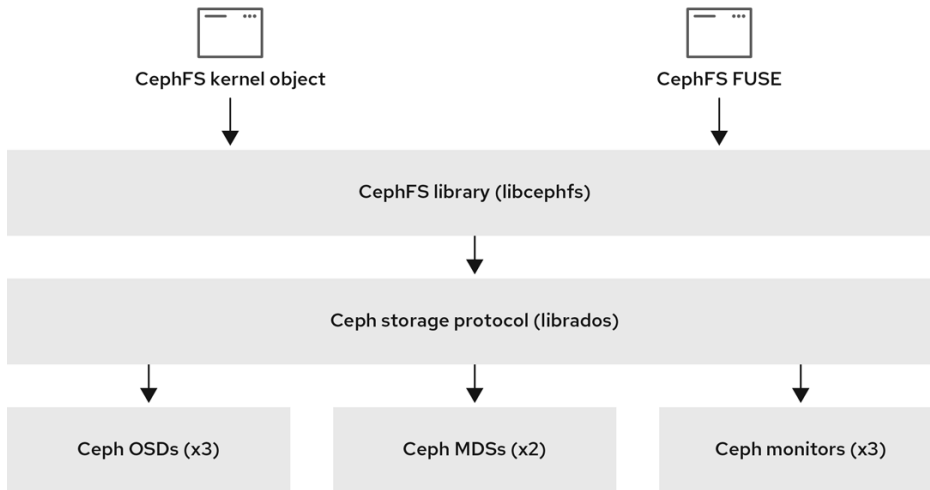
CephFS クライアントは、FUSE クライアントの **ceph-fuse** やカーネルクライアントの **kcephfs** など、CephFS を使用するアプリケーションの代わりに I/O 操作を行います。CephFS クライアントは、アクティブな Metadata Server にメタデータの要求を送信します。その代わりに、CephFS クライアントはファイルのメタデータ認識し、メタデータとファイルデータの両方を安全にキャッシュすることができます。

### メタデータサーバー (MDS)

MDS では以下のことを行います。

- CephFS クライアントにメタデータを提供します。
- Ceph File System に保存されているファイルに関連するメタデータを管理します。
- 共有されている Red Hat Ceph Storage クラスタへのアクセスを調整します。
- ホットなメタデータをキャッシュして、バックアップメタデータプールストアへのリクエストを減らします。
- CephFS クライアントのキャッシュを管理して、キャッシュコヒーレンスを維持します。
- アクティブなデータシート間でホットメタデータを複製します。
- メタデータミューテーションをコンパクトジャーナルにまとめて、バックメタデータプールに定期的にフラッシュします。
- CephFS では、少なくとも 1 つの Metadata Server デーモン (**ceph-mds**) の実行が必要です。

下図は、Ceph File System のコンポーネント層を示しています。



157\_Ceph\_1021

一番下の層は、基礎となるコアストレージクラスターコンポーネントを表しています。

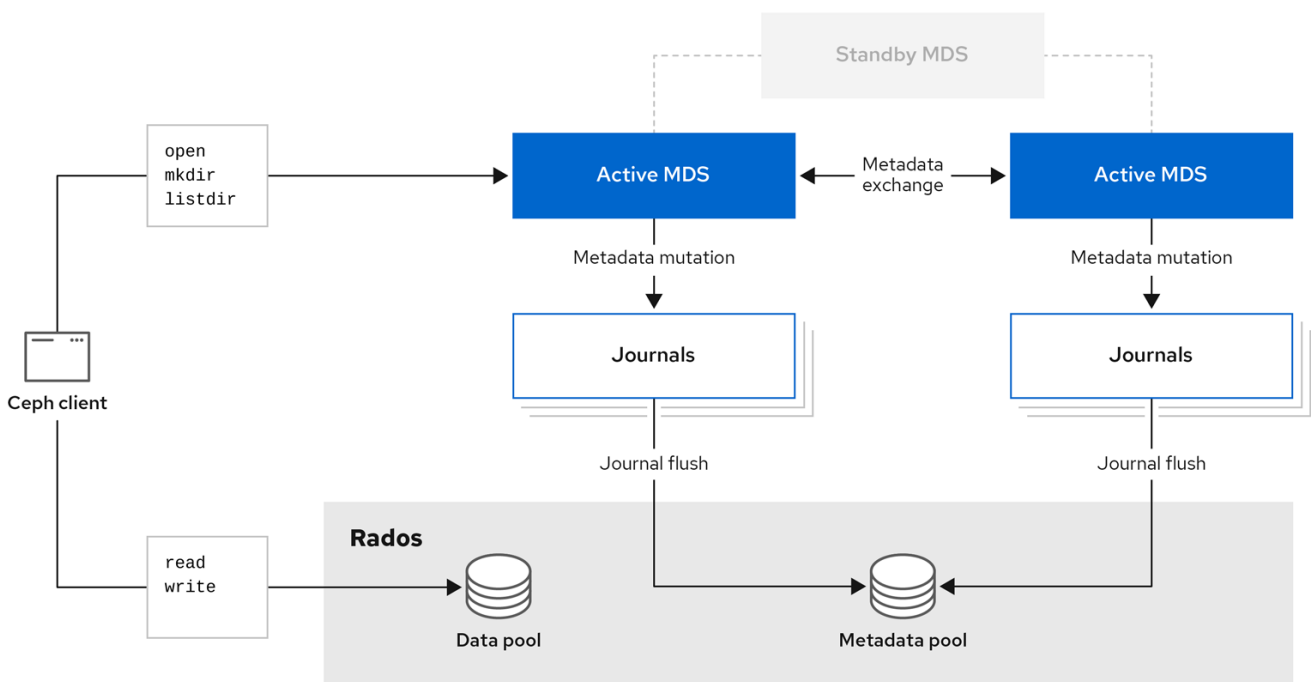
- Ceph OSD (**ceph-osd**) Ceph File System のデータとメタデータが格納されています。
- Ceph File System のメタデータを管理する Ceph Metadata Servers (**ceph-mds**)。
- クラスタマップのマスターコピーを管理する Ceph Monitors (**ceph-mon**)。

Ceph Storage プロトコル層は、コアストレージクラスターと対話するための Ceph ネイティブ **librados** ライブラリーを表します。

CephFS ライブラリー層には、**librados** の上で動作し、Ceph File System を表す CephFS **libcephfs** ライブラリーが含まれます。

一番上の層は、Ceph File Systems にアクセスできる 2 種類の Ceph クライアントを表しています。

下の図は、Ceph File System のコンポーネントがどのように相互に作用するかを詳しく示しています。



157\_Ceph\_1021

## 関連情報

- Ceph Metadata サーバーをインストールするには、ファイルシステムガイドの [Ceph Orchestrator を使用した MDS サービスの管理](#) セクションを参照してください。
- Ceph File System を作成するには、ファイルシステムガイドの [Red Hat Ceph File System の導入](#) セクションを参照してください。

## 1.3. CEPH FILE SYSTEM と SELINUX

Red Hat Enterprise Linux 8.3 および Red Hat Ceph Storage 4.2 より、Ceph File Systems (CephFS) 環境での Security-Enhanced Linux (SELinux) の使用をサポートしています。CephFS では、任意の SELinux ファイルタイプを設定できるようになったほか、個々のファイルに特定の SELinux タイプを割り当てすることもできます。このサポートは、Ceph File System Metadata Server (MDS)、CephFS File System in User Space (FUSE) クライアント、および CephFS カーネルクライアントに適用されます。

## 関連情報

- SELinux の詳細については、Red Hat Enterprise Linux 8 の [SELinux の使い方ガイド](#) を参照してください。

## 1.4. CEPH FILE SYSTEM の制限と POSIX 規格

Ceph File System は、以下の点で厳密な POSIX セマンティクスから乖離しています。

- クライアントがファイルの書き込みに失敗した場合、書き込み操作は必ずしも Atomic ではありません。例えば、**O\_SYNC** フラグで開かれた 8MB のバッファを持つファイルに対して、クライアントが **write()** システムコールを呼び出したところ、予期せぬ終了で、書き込み操作が部分的にしかできなくなってしまうことがあります。ローカルファイルシステムを含め、ほとんどのファイルシステムがこのような動作をします。
- 書き込み操作が同時に行われる状況では、オブジェクトの境界を超えた書き込み操作は必ずしも Atomic ではありません。例えば、ライター A が "aa|aa"、ライター B が "bb|bb" を同時に書いた場合、"|" はオブジェクトの境界であり、本来の "aa|aa" や "bb|bb" ではなく、"aa|bb" が書かれてしまいます。
- POSIX には **telldir()** や **seekdir()** というシステムコールがあり、カレントディレクトリーのオフセットを取得して、そこまでシークすることができます。CephFS はいつでもディレクトリーを断片化できるため、ディレクトリーの安定した整数オフセットを返すことは困難です。そのため、0 以外のオフセットで **seekdir()** システムコールを呼び出しても、動作する場合がありますが、動作を保証するものではありません。**seekdir()** をオフセット 0 で呼び出すと必ず動作します。これは、**rewinddir()** システムコールと同等のものです。
- スパースファイルは、**stat()** システムコールの **st\_blocks** フィールドに正しく伝わりませんでした。**st\_blocks** フィールドには、ファイルサイズをブロックサイズで割った商が常に入力されているため、CephFS では、割り当てられたり書き込まれたりしたファイルの一部を明示的に追跡しません。この動作により、**du** などのユーティリティーが使用スペースを過大評価してしまいます。
- **mmap()** システムコールでファイルを複数のホストのメモリーにマッピングした場合、書き込み操作が他のホストのキャッシュに一貫して伝わらない。つまり、あるページがホスト A でキャッシュされ、ホスト B で更新された場合、ホスト A のページはコヒーレントに無効にはなりません。
- CephFS クライアントには、スナップショットへのアクセス、作成、削除、名前の変更に使用

される隠れた `.snap` ディレクトリーがあります。このディレクトリーは `readdir()` システムコールから除外されていますが、同名のファイルやディレクトリーを作成しようとしたプロセスはエラーを返します。この隠しディレクトリーの名前は、マウント時に `-o snapdirname=<new_name>` オプションを使用するか、`client_snapdir` 設定オプションを使用して変更できません。

#### 関連情報

- Ceph Metadata サーバーをインストールするには、[ファイルシステムガイドの Ceph Orchestrator を使用した MDS サービスの管理](#) セクションを参照してください。
- Ceph File System を作成するには、[ファイルシステムガイドの Red Hat Ceph File System の導入](#) セクションを参照してください。

#### 関連情報

- Red Hat OpenStack Platform で Ceph File System へのインターフェイスとして NFS Ganesha を使用する場合、そのような環境をデプロイメントする方法は、[CephFS via NFS Back End Guide for Shared File System Service の CephFS with NFS-Ganesha](#) のデプロイメントセクションを参照してください。

## 第2章 CEPH FILE SYSTEM METADATA SERVER

ストレージ管理者として、Ceph File System (CephFS) Metadata Server (MDS) のさまざまな状態について学ぶとともに、CephFS MDS ランキングの仕組み、MDS スタンバイデーモンの設定、キャッシュサイズの制限についても学ぶことができます。これらの概念を知ることによって、ストレージ環境に合わせて MDS デーモンを設定することができます。

### 前提条件

- 実行中、および正常な Red Hat Ceph Storage クラスタ
- Ceph Metadata Server デーモン (**ceph-mds**) のインストール。MDS デーモンの設定の詳細は、[Red Hat Ceph Storage File System ガイドの Ceph Orchestrator を使用した MDS サービスの管理](#) セクションを参照してください。

### 2.1. METADATA SERVER デーモンの状態

Metadata Server (MDS) のデーモンは、2つの状態で動作します。

- **Active**: Ceph File System に保存されているファイルとディレクトリーのメタデータを管理します。
- **Standby**: バックアップとして機能し、アクティブな MDS デーモンが反応しなくなったときにアクティブになります。

デフォルトでは、Ceph File System はアクティブな MDS デーモンを1つだけ使用します。ただし、多くのクライアントがあるシステムでは複数のアクティブな MDS デーモンを使用する利点があります。

ファイルシステムでは、複数のアクティブな MDS デーモンを使用するように設定することで、大規模なワークロードに対してメタデータのパフォーマンスを拡張することができます。メタデータの負荷パターンが変化したときに、アクティブな MDS デーモンがメタデータのワークロードを動的に分担します。なお、複数のアクティブな MDS デーモンを持つシステムでは、高可用性を維持するためにスタンバイ MDS デーモンが必要となります。

#### Active MDS デーモンが停止したときの動作について

アクティブな MDS が応答しなくなると、Ceph Monitor デーモンは **mds\_beacon\_grace** オプションで指定された値に等しい秒数だけ待機します。指定した時間が経過してもアクティブな MDS が応答しない場合、Ceph Monitor は MDS デーモンを **laggy** としてマークします。設定に応じて、いずれかのスタンバイデーモンがアクティブになります。



#### 注記

**mds\_beacon\_grace** の値を変更するには、Ceph の設定ファイルにこのオプションを追加して、新しい値を指定します。

### 2.2. メタデータサーバーのランク

各 Ceph File System (CephFS) には、ランクの数があり、デフォルトでは1つで、ゼロから始まります。

ランクは、メタデータのワークロードを複数の Metadata Server (MDS) デーモン間で共有する方法を定義します。ランク数は、一度にアクティブにすることができる MDS デーモンの最大数です。各 MDS デーモンは、そのランクに割り当てられた CephFS メタデータのサブセットを処理します。

各 MDS デーモンは、最初はランクなしで起動します。Ceph Monitor は、デーモンにランクを割り当てます。MDS デーモンは一度に1つのランクしか保持できません。デーモンがランクを失うのは、停止したときだけです。

**max\_mds** の設定は、作成されるランクの数を制御します。

CephFS の実際のランク数は、新しいランクを受け入れるための予備のデーモンが利用できる場合にのみ増加します。

## ランクステート

ランクには、以下の状態があります。

- **Up:** MDS デーモンに割り当てられたランクです。
- **Failed:** どの MDS デーモンにも関連付けられていないランクです。
- **Damaged:** メタデータが破損していたり、欠落していたりと、ダメージを受けているランクです。オペレーターが問題を解決して、破損したランクに **ceph mds repaired** コマンドを使用するまで、破損したランクはどの MDS デーモンにも割り当てられません。

## 2.3. メタデータサーバーのキャッシュサイズ制限

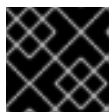
Ceph File System (CephFS) の Metadata Server (MDS) キャッシュのサイズを以下の方法で制限できます。

- **メモリーの制限:** **mds\_cache\_memory\_limit** オプションを使用します。Red Hat では、**mds\_cache\_memory\_limit** に 8 GB ~ 64 GB の値を推奨しています。より多くのキャッシュを設定すると、復元で問題が発生する可能性があります。この制限は、MDS の望ましい最大メモリー使用量の約 66% です。



### 注記

**mds\_cache\_memory\_limit** のデフォルト値は 4 GB です。デフォルト値は推奨範囲外であるため、Red Hat では前述の範囲内に値を設定することを推奨します。



### 重要

Red Hat は inode 数制限の代わりにメモリー制限を使用することを推奨します。

- **Inode 数:** **mds\_cache\_size** オプションを使用します。デフォルトでは、inode 数による MDS キャッシュの制限は無効になっています。

また、MDS の操作に **mds\_cache\_reservation** オプションを使用することで、キャッシュの予約を指定することができます。キャッシュ予約は、メモリーまたは inode の上限に対する割合で制限され、デフォルトでは 5% に設定されています。このパラメーターの目的は、データシートが新しいメタデータの操作に使用するために、キャッシュのメモリーを余分に確保することです。その結果、データシートは一般的にメモリー制限値以下で動作することになります。これは、データシートは、未使用のメタデータをキャッシュに落とすために、クライアントから古い状態を呼び出すためです。

**mds\_cache\_reservation** オプションは、MDS ノードがキャッシュが大きすぎることを示すヘルスアラートを Ceph Monitors に送信する場合を除き、すべての状況で **mds\_health\_cache\_threshold** オプションを置き換えます。デフォルトでは、**mds\_health\_cache\_threshold** は最大キャッシュサイズの 150% です。

キャッシュの制限はハードな制限ではないことに注意してください。CephFS クライアントや MDS のバグ、または誤動作するアプリケーションが原因で、MDS のキャッシュサイズが超過する可能性があります。`mds_health_cache_threshold` オプションは、ストレージクラスターの健全性に関する警告メッセージを設定し、データシートがキャッシュを縮小できない原因をオペレーターが調査できるようにします。

#### 関連情報

- 詳細は、Red Hat Ceph Storage File System ガイドの [Metadata Server daemon configuration reference](#) セクションで詳しく説明しています。

## 2.4. ファイルシステムアフィニティー

Ceph File System (CephFS) が特定の Ceph Metadata Server (MDS) を別の Ceph MDS よりも優先するように設定できます。例えば、新しく高速なハードウェアで動作している MDS を、古くて低速なハードウェアで動作しているスタンバイ MDS よりも優先的に使用したいとします。`mds_join_fs` オプションを設定することで、この優先順位を指定することができ、このファイルシステムのアフィニティーが実行されます。Ceph Monitor は、`mds_join_fs` が失敗したランクのファイルシステム名と同じである MDS スタンバイデーモンを優先します。standby-replay デーモンを選択してから、別のスタンバイデーモンを選択します。`mds_join_fs` オプションを指定したスタンバイデーモンが存在しない場合、Ceph Monitors は、最後の手段として、通常のスタンバイを交換用を選択するか、その他の利用可能なスタンバイを選択します。Ceph Monitor は、Ceph ファイルシステムを定期的に検査して、親和性の低い Ceph MDS の代わりに親和性の高いスタンバイが利用できるかどうかを確認します。

#### 関連情報

- 詳細は Red Hat Ceph Storage File System ガイドの [ファイルシステムのアフィニティーを設定する](#) のセクションを参照してください。

## 2.5. CEPH ORCHESTRATOR を使用した MDS サービスの管理

ストレージ管理者は、バックエンドにて Cephadm と Ceph Orchestrator を使用して MDS サービスをデプロイできます。デフォルトでは、Ceph File System (CephFS) はアクティブな MDS デーモンを1つだけ使用します。ただし、多くのクライアントがあるシステムでは複数のアクティブな MDS デーモンを使用する利点があります。

本セクションでは、以下の管理タスクを説明します。

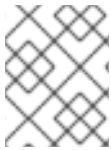
- [コマンドラインインターフェイスを使用した MDS サービスのデプロイ](#)
- [サービス指定を使用した MDS サービスのデプロイ](#)
- [Ceph Orchestrator を使用した MDS サービスの削除](#)

#### 前提条件

- 稼働中の Red Hat Ceph Storage クラスターがある。
- すべてのノードへの root レベルのアクセス。
- ホストがクラスターに追加されている。
- すべてのマネージャー、モニター、および OSD デーモンがデプロイされます。

## 2.5.1. コマンドラインインターフェイスを使用した MDS サービスのデプロイ

Ceph Orchestrator を使用すると、コマンドラインインターフェイスで **placement** 仕様を使用して、Metadata Server (MDS) サービスをデプロイできます。Ceph ファイルシステム (CephFS) には、1つ以上の MDS が必要です。



### 注記

最低でも、Ceph ファイルシステム (CephFS) データ用のプール1つと CephFS メタデータ用のプール1つの2つのプールがあるようにしてください。

### 前提条件

- 稼働中の Red Hat Ceph Storage クラスタがある。
- ホストがクラスタに追加されている。
- すべてのマネージャー、モニター、および OSD デーモンがデプロイされます。

### 手順

1. Cephadm シェルにログインします。

#### 例

```
[root@host01 ~]# cephadm shell
```

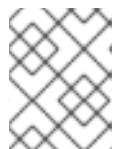
2. 配置仕様を使用して MDS デーモンをデプロイする方法は2つあります。

### 方法1

- **ceph fs volume** を使用して MDS デーモンを作成します。これにより、CephFS に関連付けられた CephFS ボリュームとプールが作成され、ホストで MDS サービスも開始されます。

#### 構文

```
ceph fs volume create FILESYSTEM_NAME --placement="NUMBER_OF_DAEMONS  
HOST_NAME_1 HOST_NAME_2 HOST_NAME_3"
```



### 注記

デフォルトでは、このコマンドに対してレプリケートされたプールが作成されません。

#### 例

```
[ceph: root@host01 /]# ceph fs volume create test --placement="2 host01 host02"
```

### 方法2

- プール、CephFS を作成してから、配置仕様を使用して MDS サービスをデプロイします。
  - a. CephFS のプールを作成します。



## 構文

```
ceph osd pool create DATA_POOL [PG_NUM]
ceph osd pool create METADATA_POOL [PG_NUM]
```

## 例

```
[ceph: root@host01 /]# ceph osd pool create cephfs_data 64
[ceph: root@host01 /]# ceph osd pool create cephfs_metadata 64
```

通常、メタデータプールは、データプールよりもオブジェクトがはるかに少ないため、控えめな数の配置グループ (PG) で開始できます。必要に応じて PG の数を増やすことができます。プールサイズの範囲は 64 PG ~ 512 PG です。データプールのサイズは、ファイルシステム内で予想されるファイルの数とサイズに比例します。



## 重要

メタデータプールでは、以下を使用することを検討してください。

- このプールへのデータ損失によりファイルシステム全体にアクセスできなくなる可能性があるため、レプリケーションレベルが高くなります。
- Solid-State Drive (SSD) ディスクなどのレイテンシーが低くなるストレージ。これは、クライアントで観察されるファイルシステム操作のレイテンシーに直接影響するためです。

- b. データプールおよびメタデータプールのファイルシステムを作成します。

## 構文

```
ceph fs new FILESYSTEM_NAME METADATA_POOL DATA_POOL
```

## 例

```
[ceph: root@host01 /]# ceph fs new test cephfs_metadata cephfs_data
```

- c. **ceph orch apply** コマンドを使用して MDS サービスをデプロイします。

## 構文

```
ceph orch apply mds FILESYSTEM_NAME --placement="NUMBER_OF_DAEMONS
HOST_NAME_1 HOST_NAME_2 HOST_NAME_3"
```

## 例

```
[ceph: root@host01 /]# ceph orch apply mds test --placement="2 host01 host02"
```

## 検証

- サービスをリスト表示します。

### 例

```
[ceph: root@host01 /]# ceph orch ls
```

- CephFS のステータスを確認します。

### 例

```
[ceph: root@host01 /]# ceph fs ls  
[ceph: root@host01 /]# ceph fs status
```

- ホスト、デーモン、およびプロセスをリスト表示します。

### 構文

```
ceph orch ps --daemon_type=DAEMON_NAME
```

### 例

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=mds
```

## 関連情報

- Ceph ファイルシステム (CephFS) の作成に関する詳細は、[Red Hat Ceph Storage ファイルシステムガイド](#) を参照してください。
- プール値の設定の詳細は、[プール内の配置グループ数の設定](#) を参照してください。

## 2.5.2. サービス仕様を使用した MDS サービスのデプロイ

Ceph Orchestrator を使用すると、サービス仕様を使用して MDS サービスをデプロイできます。



### 注記

少なくとも 2 つのプールがあることを確認してください。1 つは Ceph ファイルシステム (CephFS) データ用で、もう 1 つは CephFS メタデータ用です。

## 前提条件

- 稼働中の Red Hat Ceph Storage クラスタがある。
- ホストがクラスタに追加されている。
- すべてのマネージャー、モニター、および OSD デーモンがデプロイされます。

## 手順

1. **mds.yaml** ファイルを作成します。

### 例

```
[root@host01 ~]# touch mds.yaml
```

2. **mds.yaml** ファイルを編集し、以下の詳細を含めます。

### 構文

```
service_type: mds
service_id: FILESYSTEM_NAME
placement:
  hosts:
    - HOST_NAME_1
    - HOST_NAME_2
    - HOST_NAME_3
```

### 例

```
service_type: mds
service_id: fs_name
placement:
  hosts:
    - host01
    - host02
```

3. YAML ファイルをコンテナ内のディレクトリーにマウントします。

### 例

```
[root@host01 ~]# cephadm shell --mount mds.yaml:/var/lib/ceph/mds/mds.yaml
```

4. そのディレクトリーに移動します。

### 例

```
[ceph: root@host01 /]# cd /var/lib/ceph/mds/
```

5. Cephadm シェルにログインします。

### 例

```
[root@host01 ~]# cephadm shell
```

6. 以下のディレクトリーに移動します。

### 例

```
[ceph: root@host01 /]# cd /var/lib/ceph/mds/
```

7. サービス仕様を使用して MDS サービスをデプロイします。

### 構文

```
ceph orch apply -i FILE_NAME.yaml
```

### 例

```
[ceph: root@host01 mds]# ceph orch apply -i mds.yaml
```

8. MDS サービスがデプロイされ、機能したら、CephFS を作成します。

### 構文

```
ceph fs new CEPHFS_NAME METADATA_POOL DATA_POOL
```

### 例

```
[ceph: root@host01 /]# ceph fs new test metadata_pool data_pool
```

### 検証

- サービスをリスト表示します。

### 例

```
[ceph: root@host01 /]# ceph orch ls
```

- ホスト、デーモン、およびプロセスをリスト表示します。

### 構文

```
ceph orch ps --daemon_type=DAEMON_NAME
```

### 例

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=mds
```

### 関連情報

- Ceph ファイルシステム (CephFS) の作成に関する詳細は、[Red Hat Ceph Storage ファイルシステムガイド](#) を参照してください。

## 2.5.3. Ceph Orchestrator を使用した MDS サービスの削除

`ceph orch rm` コマンドを使用してサービスを削除できます。または、ファイルシステムおよび関連するプールを削除できます。

### 前提条件

- 稼働中の Red Hat Ceph Storage クラスタがある。
- すべてのノードへの root レベルのアクセス。
- ホストがクラスタに追加されている。
- ホストにデプロイされた MDS デーモン 1 つ以上。

### 手順

- MDS デーモンをクラスターから削除する方法は 2 つあります。

### 方法 1

- CephFS ボリューム、関連するプール、およびサービスを削除します。
  - a. Cephadm シェルにログインします。

#### 例

```
[root@host01 ~]# cephadm shell
```

- b. 設定パラメーター **mon\_allow\_pool\_delete** を **true** に設定します。

#### 例

```
[ceph: root@host01 /]# ceph config set mon mon_allow_pool_delete true
```

- c. ファイルシステムを削除します。

#### 構文

```
ceph fs volume rm FILESYSTEM_NAME --yes-i-really-mean-it
```

#### 例

```
[ceph: root@host01 /]# ceph fs volume rm cephfs-new --yes-i-really-mean-it
```

このコマンドは、ファイルシステム、そのデータ、メタデータプールを削除します。また、有効な **ceph-mgr** Orchestrator モジュールを使用して MDS を削除しようとします。

### 方法 2

- **ceph orch rm** コマンドを使用して、クラスター全体から MDS サービスを削除します。
  - a. サービスをリスト表示します。

#### 例

```
[ceph: root@host01 /]# ceph orch ls
```

- b. サービスの削除

#### 構文

```
ceph orch rm SERVICE_NAME
```

#### 例

```
[ceph: root@host01 /]# ceph orch rm mds.test
```

## 検証

- ホスト、デーモン、およびプロセスをリスト表示します。

## 構文

```
ceph orch ps
```

## 例

```
[ceph: root@host01 /]# ceph orch ps
```

## 関連情報

- 詳細は、Red Hat Ceph Storage オペレーションガイドの [Deploying the MDS service using the command line interface](#) セクションを参照してください。
- 詳細は、Red Hat Ceph Storage Operations ガイドの [サービス仕様を使用した MDS サービスのデプロイメント](#) セクションを参照してください。

## 2.6. ファイルシステムのアフィニティーを設定する

特定の Ceph Metadata Server (MDS) に対する Ceph File System (CephFS) のアフィニティーを設定します。

### 前提条件

- 正常で稼働している Ceph File System。
- Ceph Monitor ノードへの root レベルのアクセス。

### 手順

1. Ceph File System の現在の状態を確認します。

### 例

```
[root@mon ~]# ceph fs dump
dumped fsmap epoch 399
...
Filesystem 'cephfs01' (27)
...
e399
max_mds 1
in 0
up {0=20384}
failed
damaged
stopped
...
[mds.a{0:20384} state up:active seq 239 addr
[v2:127.0.0.1:6854/966242805,v1:127.0.0.1:6855/966242805]]

Standby daemons:
```

```
[mds.b{-1:10420} state up:standby seq 2 addr
[v2:127.0.0.1:6856/2745199145,v1:127.0.0.1:6857/2745199145]]
```

2. ファイルシステムのアフィニティーを設定します。

### 構文

```
ceph config set STANDBY_DAEMON mds_join_fs FILE_SYSTEM_NAME
```

### 例

```
[root@mon ~]# ceph config set mds.b mds_join_fs cephfs01
```

Ceph MDS のフェイルオーバーイベントが発生すると、ファイルシステムはアフィニティーが設定されているスタンバイデーモンを優先します。

### 例

```
[root@mon ~]# ceph fs dump
dumped fsmap epoch 405
e405
...
Filesystem 'cephfs01' (27)
...
max_mds 1
in 0
up {0=10420}
failed
damaged
stopped
...
[mds.b{0:10420} state up:active seq 274 join_fscid=27 addr
[v2:127.0.0.1:6856/2745199145,v1:127.0.0.1:6857/2745199145]] 1

Standby daemons:

[mds.a{-1:10720} state up:standby seq 2 addr
[v2:127.0.0.1:6854/1340357658,v1:127.0.0.1:6855/1340357658]]
```

- 1 **mds.b** デーモンのファイルシステムダンプの出力に **join\_fscid=27** が含まれるようになりました。



### 重要

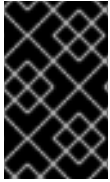
ファイルシステムがデグレードまたはアンダーサイズの状態である場合、ファイルシステムのアフィニティーを実施するためのフェイルオーバーは発生しません。

### 関連情報

- 詳細は、Red Hat Ceph Storage ファイルシステムガイドの [ファイルシステムのアフィニティ](#) セクションを参照してください。

## 2.7. 複数のアクティブな METADATA SERVER デーモンの設定

複数のアクティブなメタデータサーバー (MDS) デーモンを設定し、大規模システムのメタデータのパフォーマンスを拡張します。



### 重要

スタンバイ状態の MDS デーモンをすべてアクティブ状態に変換しないでください。Ceph File System (CephFS) は、高可用性を維持するために、少なくとも1つのスタンバイ MDS デーモンを必要とします。

### 前提条件

- MDS ノードでの Ceph 管理機能。
- Ceph Monitor ノードへの root レベルのアクセス。

### 手順

1. **max\_mds** パラメーターには、アクティブな MDS デーモンの数を設定してください。

#### 構文

```
ceph fs set NAME max_mds NUMBER
```

#### 例

```
[root@mon ~]# ceph fs set cephfs max_mds 2
```

この例では、**cephfs** という CephFS でアクティブな MDS デーモンの数を 2 つに増やしています。



### 注記

Ceph は、新しいランクを取るために予備の MDS デーモンが利用できる場合のみ、CephFS の実際のランク数を増やします。

2. アクティブな MDS デーモンの数を確認します。

#### 構文

```
ceph fs status NAME
```

#### 例

```
[root@mon ~]# ceph fs status cephfs
cephfs - 0 clients
=====
+-----+-----+-----+-----+-----+-----+-----+
```



```

| RANK | STATE | MDS | ACTIVITY | DNS | INOS | DIRS | CAPS |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 0 | active | node1 | Reqs: 0/s | 10 | 12 | 12 | 0 |
| 1 | active | node2 | Reqs: 0/s | 10 | 12 | 12 | 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
| POOL | TYPE | USED | AVAIL |
+-----+-----+-----+-----+
| cephfs_metadata | metadata | 4638 | 26.7G |
| cephfs_data | data | 0 | 26.7G |
+-----+-----+-----+-----+

+-----+
| STANDBY MDS |
+-----+
| node3 |
+-----+

```

## 関連情報

- 詳細は、Red Hat Ceph Storage ファイルシステムガイドの [Metadata Server の状態](#) セクションを参照してください。
- 詳細は、Red Hat Ceph Storage File System ガイドの [アクティブ MDS デーモンデーモンの数を減らす](#) セクションを参照してください。
- 詳細は、Red Hat Ceph Storage 管理ガイドの [Ceph ユーザーの管理](#) セクションを参照してください。

## 2.8. スタンバイデーモンの数の設定

各 Ceph File System (CephFS) では、健全であると判断するために必要なスタンバイデーモンの数を指定できます。この数には、ランク不具合を待っている standby-replay デーモンも含まれます。

### 前提条件

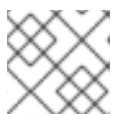
- Ceph Monitor ノードへの root レベルのアクセス。

### 手順

- 特定の CephFS のスタンバイデーモンの予想数を設定します。

### 構文

```
ceph fs set FS_NAME standby_count_wanted NUMBER
```



### 注記

**NUMBER** を 0 にすると、デーモンのヘルスチェックが無効になります。

### 例

```
[root@mon ~]# ceph fs set cephfs standby_count_wanted 2
```

この例では、予想されるスタンバイデーモンの数を 2 に設定しています。

## 2.9. STANDBY-REPLAY 用 METADATA SERVER の設定

各 Ceph File System (CephFS) を設定して、standby-replay の Metadata Server (MDS) デーモンを追加します。これにより、アクティブな MDS が利用できなくなった場合のフェイルオーバー時間を短縮することができます。

この特定の standby-replay デーモンは、アクティブなデータシートのメタデータジャーナルに従います。standby-replay デーモンは、同一ランクのアクティブなデータシートでのみ使用され、他のランクでは使用できません。



### 重要

standby-replay を使用する場合は、すべてのアクティブなデータシートに standby-replay デーモンが必要です。

### 前提条件

- Ceph Monitor ノードへの root レベルのアクセス。

### 手順

- 特定の CephFS の standby-replay を設定します。

### 構文

```
ceph fs set FS_NAME allow_standby_replay 1
```

### 例

```
[root@mon ~]# ceph fs set cephfs allow_standby_replay 1
```

この例では、ブール値が **1** であるため、standby-replay デーモンをアクティブな Ceph MDS デーモンに割り当てることができます。

### 関連情報

- 詳細は、Red Hat Ceph Storage File System ガイドの [ceph mds fail コマンドの使用](#) セクションを参照してください。

## 2.10. エフェメラルピンニングポリシー

エフェメラルピンは、サブツリーの静的なパーティションであり、拡張属性を使用したポリシーで設定できます。ポリシーでは、ディレクトリーにエフェメラルピンを自動的に設定することができます。ディレクトリーにエフェメラルピンを設定すると、そのピンは自動的に特定のランクに割り当てられ、すべての Ceph MDS ランクに均一に分散されるようになります。どのランクが割り当てられるかは、一貫したハッシュとディレクトリーの inode 番号によって決定されます。エフェメラルピンは、ディレクトリーの inode がファイルシステムのキャッシュから削除されても持続しません。Ceph Metadata Server (MDS) をフェイルオーバーする際には、エフェメラルピンがジャーナルに記録されるため、Ceph MDS スタンバイサーバーがこの情報を失うことはありません。エフェメラルピンを使用する際のポリシーは 2 種類あります。



## 注記

**attr** パッケージと **jq** パッケージのインストールは、エフェメラルピンングポリシーの前提条件です。

### Distributed (分散)

このポリシーでは、ディレクトリーの直接の子のすべてが、一時的にピン留めされなければならないことを強制します。たとえば、分散ポリシーを使用して、ユーザーのホームディレクトリーを Ceph File System クラスター全体に広げることができます。**ceph.dir.pin.distributed** 拡張属性を設定して、このポリシーを有効にします。

### 構文

```
setfattr -n ceph.dir.pin.distributed -v 1 DIRECTORY_PATH
```

### 例

```
[root@host01 mount]# setfattr -n ceph.dir.pin.distributed -v 1 dir1/
```

### ランダム

このポリシーでは、子孫のサブディレクトリーが一時的にピン留めされる可能性があります。エフェメラルピン留めが可能なディレクトリーの割合をカスタマイズできます。**ceph.dir.pin.random** を設定し、パーセンテージを設定することで、このポリシーを有効にします。Red Hat では、このパーセンテージを 1% (**0.01**) より小さい値に設定することを推奨します。サブツリーのパーティションの数が多すぎると、パフォーマンスが低下します。**mds\_export\_ephemeral\_random\_max** Ceph MDS 設定オプションを設定することで、最大の割合を設定できます。パラメーター **mds\_export\_ephemeral\_distributed** および **mds\_export\_ephemeral\_random** はすでに有効になっています。

### 構文

```
setfattr -n ceph.dir.pin.random -v PERCENTAGE_IN_DECIMAL DIRECTORY_PATH
```

### 例

```
[root@host01 mount]# setfattr -n ceph.dir.pin.random -v 0.01 dir1/
```

ピンングを有効にした後、次のいずれかのコマンドを実行して **確認** できます。

### 構文

```
getfattr -n ceph.dir.pin.random DIRECTORY_PATH
getfattr -n ceph.dir.pin.distributed DIRECTORY_PATH
```

### 例

```
[root@host01 mount]# getfattr -n ceph.dir.pin.distributed dir1/
# file: dir1/
ceph.dir.pin.distributed="1"
```

```
[root@host01 mount]# getfattr -n ceph.dir.pin.random dir1/
# file: dir1/
ceph.dir.pin.random="0.01"
```

## 例

```
[ceph: root@host01 /]# ceph tell mds.a get subtrees | jq '[] | [.dir.path, .auth_first, .export_pin]'
```

ディレクトリーが固定されている場合、**export\_pin** の値は、ランク **0** に固定されている場合は **0**、ランク **1** に固定されている場合は **1**、などとなります。ディレクトリーが固定されていない場合、値は **-1** です。

パーティション設定ポリシーを **削除** するには、拡張属性を削除するか、値を **0** に設定します。

## 構文

```
setfattr -n ceph.dir.pin.distributed -v 0 DIRECTORY_PATH
```

## 例

```
[root@host01 mount]# setfattr -n ceph.dir.pin.distributed -v 0 dir1/
```

次のコマンドのいずれかを実行して **確認** できます。構文

```
getfattr -n ceph.dir.pin.distributed DIRECTORY_PATH
```

## 例

```
[root@host01 mount]# getfattr -n ceph.dir.pin.distributed dir1/
```

エクスポートピンの場合は、拡張属性を削除するか、拡張属性を **-1** に設定します。

## 構文

```
setfattr -n ceph.dir.pin -v -1 DIRECTORY_PATH
```

## 例

```
[root@host01 mount]# setfattr -n ceph.dir.pin -v -1 dir1/
```

## 関連情報

- ピンの手動設定の詳細は、Red Hat Ceph Storage File System ガイドの [特定のランクに手動でディレクトリーツリーをピン設定する](#) セクションを参照してください。

## 2.11. ディレクトリーツリーを特定のランクに手動で固定する

メタデータを特定の Ceph Metadata Server (MDS) ランクに明示的にマッピングすることで、ダイナミックバランサーをオーバーライドしたい場合もあります。これを手動で行うことで、アプリケーションの負荷を均等に分散したり、ユーザーのメタデータ要求が Ceph File System クラスタに与える影響

を抑えることができます。ディレクトリーを手動で固定することは、**ceph.dir.pin** 拡張属性を設定することで、エクスポートピンとも呼ばれます。

ディレクトリーのエクスポートピンは、最も近い親ディレクトリーから継承されますが、そのディレクトリーにエクスポートピンを設定することで、上書きすることができます。例えば、あるディレクトリーにエクスポートピンを設定すると、そのディレクトリーのすべてのサブディレクトリーに影響します。

```
[root@client ~]# mkdir -p a/b ❶
[root@client ~]# setfattr -n ceph.dir.pin -v 1 a/ ❷
[root@client ~]# setfattr -n ceph.dir.pin -v 0 a/b ❸
```

- ❶ ディレクトリー **a/** と **a/b** はどちらも輸出用のピンセットがない状態でスタートします。
- ❷ ディレクトリー **a/** および **a/b** は、ランク **1** に固定されるようになりました。
- ❸ ディレクトリー **a/b** は、ランク **0** およびディレクトリー **a/** にピンングされ、そのサブディレクトリーの残りの部分は、引き続きランク **1** に固定されています。

### 前提条件

- 稼働中の Red Hat Ceph Storage クラスタがある。
- 実行中の Ceph File System
- CephFS クライアントへのルートレベルのアクセス。
- **attr** パッケージのインストール。

### 手順

- ディレクトリーにエクスポートピンを設定します。

### 構文

```
setfattr -n ceph.dir.pin -v RANK PATH_TO_DIRECTORY
```

### 例

```
[root@client ~]# setfattr -n ceph.dir.pin -v 2 cephfs/home
```

### 関連情報

- ピンの自動設定の詳細は、Red Hat Ceph Storage File System ガイドの [エフェメラルピンングポリシー](#) セクションを参照してください。

## 2.12. アクティブな METADATA SERVER デーモンの数を減らす方法

アクティブな Ceph File System (CephFS) メタデータサーバー (MDS) デーモンの数を減らす方法。

### 前提条件

- 削除するランクは最初にアクティブにする必要があります。つまり、**max\_mds** パラメーターで指定された MDS デーモンの数と同じである必要があります。
- Ceph Monitor ノードへの root レベルのアクセス。

## 手順

1. **max\_mds** パラメーターで指定された MDS デーモンの数を設定します。

### 構文

```
ceph fs status NAME
```

### 例

```
[root@mon ~]# ceph fs status cephfs
cephfs - 0 clients

+-----+-----+-----+-----+-----+-----+-----+
| RANK | STATE | MDS | ACTIVITY | DNS | INOS | DIRS | CAPS |
+-----+-----+-----+-----+-----+-----+-----+
| 0 | active | node1 | Reqs: 0/s | 10 | 12 | 12 | 0 |
| 1 | active | node2 | Reqs: 0/s | 10 | 12 | 12 | 0 |
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
| POOL | TYPE | USED | AVAIL |
+-----+-----+-----+-----+
| cephfs_metadata | metadata | 4638 | 26.7G |
| cephfs_data | data | 0 | 26.7G |
+-----+-----+-----+-----+

+-----+
| Standby MDS |
+-----+
| node3 |
+-----+
```

2. 管理機能を持つノードで、**max\_mds** パラメーターを必要なアクティブな MDS デーモンの数に変更します。

### 構文

```
ceph fs set NAME max_mds NUMBER
```

### 例

```
[root@mon ~]# ceph fs set cephfs max_mds 1
```

3. Ceph File System のステータスを監視して、ストレージクラスターが新しい **max\_mds** 値を安定させるのを待機します。
4. アクティブな MDS デーモンの数を確認します。

### 構文

```
ceph fs status NAME
```

## 例

```
[root@mon ~]# ceph fs status cephfs
cephfs - 0 clients

+-----+-----+-----+-----+-----+-----+-----+
| RANK | STATE | MDS | ACTIVITY | DNS | INOS | DIRS | CAPS |
+-----+-----+-----+-----+-----+-----+-----+
| 0 | active | node1 | Reqs: 0/s | 10 | 12 | 12 | 0 |
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
| POOL | TYPE | USED | AVAIL |
+-----+-----+-----+-----+
| cephfs_metadata | metadata | 4638 | 26.7G |
| cephfs_data | data | 0 | 26.7G |
+-----+-----+-----+-----+

+-----+
| Standby MDS |
+-----+
| node3 |
| node2 |
+-----+
```

## 関連情報

- Red Hat Ceph Storage ファイルシステムガイドの [Metadata Server デーモンの状態](#) セクションを参照してください。
- Red Hat Ceph Storage ファイルシステムガイドの [複数のアクティブな Metadata Server デーモンの設定](#) セクションを参照してください。
- Red Hat Ceph Storage クラスターのインストールの詳細は、[Red Hat Ceph Storage インストールガイド](#) を参照してください。

## 第3章 CEPH FILE SYSTEM のデプロイメント

ストレージ管理者は、ストレージ環境に Ceph File Systems (CephFS) をデプロイでき、ストレージのニーズを満たすためにクライアントがそれらの Ceph File System をマウントすることができます。

基本的に、デプロイメントワークフローは以下の3つのステップになります。

1. Ceph Monitor ノードで Ceph File Systems を作成します。
2. 適切な機能を持つ Ceph クライアントユーザーを作成し、Ceph File System がマウントされるノードでクライアントキーを利用できるようにします。
3. カーネルクライアントまたは File System in User Space (FUSE) クライアントで使用して、専用のノードに CephFS をマウントします。

### 前提条件

- 実行中、および正常な Red Hat Ceph Storage クラスター
- Ceph Metadata Server デーモン (**ceph-mds**) のインストールおよび設定

### 3.1. レイアウト、クォータ、スナップショット、およびネットワークの制限

これらのユーザー機能は、必要な要件に基づいて Ceph File System (CephFS) へのアクセスを制限するのに役立ちます。



#### 重要

**rw** を除くすべてのユーザーキーパビリティーフラグは、アルファベット順に指定する必要があります。

#### レイアウトとクォータ

レイアウトまたはクォータを使用する場合には、**rw** 機能に加えて、クライアントが **p** フラグが必要になります。**p** フラグを設定すると、特殊拡張属性 (**ceph.** 接頭辞が付いた属性) で設定されるすべての属性を制限します。また、これによりレイアウトを持つ **openc** 操作など、これらのフィールドを設定する他の方法が制限されます。

#### 例

```
client.0
key: AQAz7EVWYgILFRAAdlcuJ10opU/JKyfFmxhuaw==
caps: [mds] allow rwp
caps: [mon] allow r
caps: [osd] allow rw tag cephfs data=cephfs_a

client.1
key: AQAz7EVWYgILFRAAdlcuJ11opU/JKyfFmxhuaw==
caps: [mds] allow rw
caps: [mon] allow r
caps: [osd] allow rw tag cephfs data=cephfs_a
```

この例では、**client.0** はファイルシステムの **cephfs\_a** のレイアウトとクォータを修正できますが、**client.1** はできません。



## スナップショット

スナップショットの作成または削除時に、クライアントは **rw** 機能に加えて **s** フラグが必要になります。機能文字列に **p** フラグも含まれる場合は、**s** フラグがこれの後に表示される必要があります。

### 例

```
client.0
key: AQAz7EVWYyglLFRAAdlcuJ10opU/JKyfFmxhuaw==
caps: [mds] allow rw, allow rws path=/temp
caps: [mon] allow r
caps: [osd] allow rw tag cephfs data=cephfs_a
```

この例では、**client.0** はファイルシステムの **cephfs\_a** の **temp** ディレクトリーでスナップショットを作成または削除することができます。

## ネットワーク

特定のネットワークから接続するクライアントを制限します。

### 例

```
client.0
key: AQAz7EVWYyglLFRAAdlcuJ10opU/JKyfFmxhuaw==
caps: [mds] allow r network 10.0.0.0/8, allow rw path=/bar network 10.0.0.0/8
caps: [mon] allow r network 10.0.0.0/8
caps: [osd] allow rw tag cephfs data=cephfs_a network 10.0.0.0/8
```

オプションのネットワークおよび接頭辞長は CIDR 表記です (例: **10.3.0.0/16**)。

## 関連情報

- Ceph ユーザー機能の設定に関する詳細は、[Red Hat Ceph Storage File System ガイドの Ceph File System のクライアント](#) セクションを参照してください。

## 3.2. CEPH ファイルシステムの作成

Ceph Monitor ノードで複数の Ceph File Systems (CephFS) を作成することができます。

### 前提条件

- 実行中、および正常な Red Hat Ceph Storage クラスター
- Ceph Metadata Server デーモン (**ceph-mds**) のインストールおよび設定
- Ceph Monitor ノードへの root レベルのアクセス。
- Ceph クライアントノードへのルートレベルのアクセスがある。

### 手順

1. Ceph Storage クラスターを使用するようにクライアントノードを設定します。
  - a. Red Hat Ceph Storage Tools リポジトリを有効にします。

## Red Hat Enterprise Linux 8

```
[root@client01 ~]# subscription-manager repos --enable=rhceph-6-tools-for-rhel-8-x86_64-rpms
```

## Red Hat Enterprise Linux 9

```
[root@client01 ~]# subscription-manager repos --enable=rhceph-6-tools-for-rhel-9-x86_64-rpms
```

- b. **ceph-fuse** パッケージをインストールします。

```
[root@client ~]# dnf install ceph-fuse
```

- c. Ceph クライアントキーリングを Ceph Monitor ノードからクライアントノードにコピーします。

### 構文

```
scp root@MONITOR_NODE_NAME:/etc/ceph/KEYRING_FILE /etc/ceph/
```

**MONITOR\_NODE\_NAME** は、Ceph Monitor ホスト名または IP アドレスに置き換えます。

### 例

```
[root@client ~]# scp root@192.168.0.1:/etc/ceph/ceph.client.1.keyring /etc/ceph/
```

- d. Ceph 設定ファイルを Monitor ノードからクライアントノードにコピーします。

### 構文

```
scp root@MONITOR_NODE_NAME:/etc/ceph/ceph.conf /etc/ceph/ceph.conf
```

**MONITOR\_NODE\_NAME** は、Ceph Monitor ホスト名または IP アドレスに置き換えます。

### 例

```
[root@client ~]# scp root@192.168.0.1:/etc/ceph/ceph.conf /etc/ceph/ceph.conf
```

- e. 設定ファイルに適切なパーミッションを設定します。

```
[root@client ~]# chmod 644 /etc/ceph/ceph.conf
```

2. Ceph ファイルシステムを作成します。

### 構文

```
ceph fs volume create FILE_SYSTEM_NAME
```

**例**

```
[root@mon ~]# ceph fs volume create cephfs01
```

この手順を繰り返して、追加のファイルシステムを作成します。

**注記**

このコマンドを実行すると、Ceph は新しいプールを自動的に作成し、新しいファイルシステムをサポートする新たな Ceph Metadata Server (MDS) デーモンをデプロイします。また、これにより MDS アフィニティーを適宜設定します。

3. Ceph クライアントから新しい Ceph File System へのアクセスを確認します。
  - a. Ceph クライアントが新しいファイルシステムへのアクセスを承認します。

**構文**

```
ceph fs authorize FILE_SYSTEM_NAME CLIENT_NAME DIRECTORY PERMISSIONS
```

**例**

```
[root@mon ~]# ceph fs authorize cephfs01 client.1 / rw
[client.1]
  key = BQAmthpf81M+JhAAiHDYQkMiCq3x+J0n9e8REK==

[root@mon ~]# ceph auth get client.1
exported keyring for client.1
[client.1]
  key = BQAmthpf81M+JhAAiHDYQkMiCq3x+J0n9e8REK==
  caps mds = "allow rw fsname=cephfs01"
  caps mon = "allow r fsname=cephfs01"
  caps osd = "allow rw tag cephfs data=cephfs01"
```



## 注記

必要に応じて、**root\_squash** オプションを指定することで安全対策を追加できます。これにより、**uid=0** または **gid=0** のクライアントが書き込み操作を行うのを許可することで、誤って削除のシナリオは阻止されますが、読み取り操作は引き続き許可されます。

## 例

```
[root@mon ~]# ceph fs authorize cephfs01 client.1 / rw root_squash
/volumes rw
[client.1]
  key = BQAmthpf81M+JhAAiHDYQkMiCq3x+J0n9e8REK==

[root@mon ~]# ceph auth get client.1
[client.1]
  key = BQAmthpf81M+JhAAiHDYQkMiCq3x+J0n9e8REK==
  caps mds = "allow rw fsname=cephfs01 root_squash, allow rw
fsname=cephfs01 path=/volumes"
  caps mon = "allow r fsname=cephfs01"
  caps osd = "allow rw tag cephfs data=cephfs01"
```

この例では、**/volumes** ディレクトリツリー内ではファイルシステム **cephfs01** に対して **root\_squash** が有効になります。



## 重要

Ceph クライアントは、それが承認されている CephFS のみを認識することができます。

- b. Ceph ユーザーのキーリングを Ceph クライアントノードにコピーします。

### 構文

```
ceph auth get CLIENT_NAME > OUTPUT_FILE_NAME
scp OUTPUT_FILE_NAME TARGET_NODE_NAME:/etc/ceph
```

### 例

```
[root@mon ~]# ceph auth get client.1 > ceph.client.1.keyring
exported keyring for client.1
[root@mon ~]# scp ceph.client.1.keyring client:/etc/ceph
root@client's password:
ceph.client.1.keyring          100% 178  333.0KB/s  00:00
```

- c. Ceph クライアントノードで、新しいディレクトリを作成します。

### 構文

```
mkdir PATH_TO_NEW_DIRECTORY_NAME
```

### 例

```
[root@client ~]# mkdir /mnt/mycephfs
```

- d. Ceph クライアントノードで、新しい Ceph File System をマウントします。

### 構文

```
ceph-fuse PATH_TO_NEW_DIRECTORY_NAME -n CEPH_USER_NAME --client-  
fs=_FILE_SYSTEM_NAME
```

### 例

```
[root@client ~]# ceph-fuse /mnt/mycephfs/ -n client.1 --client-fs=cephfs01  
ceph-fuse[555001]: starting ceph client  
2022-05-09T07:33:27.158+0000 7f11feb81200 -1 init, newargv = 0x55fc4269d5d0  
newargc=15  
ceph-fuse[555001]: starting fuse
```

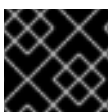
- e. Ceph クライアントノードで、新しいマウントポイントのディレクトリーコンテンツをリスト表示するか、新しいマウントポイントにファイルを作成します。

### 関連情報

- 詳細は、Red Hat Ceph Storage File System ガイドの [Ceph File System のクライアントユーザーの作成](#) セクションを参照してください。
- 詳細は、Red Hat Ceph Storage File System ガイドの [Ceph File System をカーネルクライアントとしてマウントする](#) セクションを参照してください。
- 詳細は、Red Hat Ceph Storage File System ガイドの [Ceph File System を FUSE クライアントとしてマウントする](#) セクションを参照してください。
- 詳細、Red Hat Ceph Storage File System ガイドの [Ceph File System 制限および POSIX 規格](#) セクションを参照してください。
- 詳細は、Red Hat Ceph Storage ストラテジーガイドの [プール](#) の章を参照してください。

## 3.3. CEPH ファイルシステムへのイレイジャーコーディングされたプールの追加

デフォルトでは、Ceph はデータプールにレプリケートされたプールを使用します。必要に応じて、Ceph File System に新たなイレイジャーコーディングデータプールを追加することもできます。イレイジャーコーディングプールが対応する Ceph File Systems (CephFS) は、複製されたプールでサポートされる Ceph File Systems と比較して、全体的なストレージの使用量を使用します。イレイジャーコーディングされたプールは、全体的なストレージを使用しますが、レプリケートされたプールよりも多くのメモリーおよびプロセッサリソースを使用します。



### 重要

CephFS EC プールはアーカイブのみを目的としています。



## 重要

実稼働環境では、CephFS にデフォルトのレプリケートデータプールを使用することを推奨します。CephFS で inode を作成すると、デフォルトのデータプールに少なくとも1つのオブジェクトが作成されます。デフォルトのデータにレプリケートされたプールを使用すると、小規模なオブジェクト書き込みパフォーマンスを向上し、バックトレースを更新する読み取りパフォーマンスが向上します。

### 前提条件

- 稼働中の Red Hat Ceph Storage クラスタがある。
- 既存の Ceph File System。
- BlueStore OSD を使用するプール。
- Ceph Monitor ノードへの root レベルのアクセス。
- **attr** パッケージのインストール。

### 手順

1. CephFS 用のイレイジャーコーディングデータプールを作成します。

#### 構文

```
ceph osd pool create DATA_POOL_NAME erasure
```

#### 例

```
[root@mon ~]# ceph osd pool create cephfs-data-ec01 erasure
pool 'cephfs-data-ec01' created
```

2. プールが追加されたことを確認します。

#### 例

```
[root@mon ~]# ceph osd lspools
```

3. 消去コード化されたプールでのオーバーライトを有効にします。

#### 構文

```
ceph osd pool set DATA_POOL_NAME allow_ec_overwrites true
```

#### 例

```
[root@mon ~]# ceph osd pool set cephfs-data-ec01 allow_ec_overwrites true
set pool 15 allow_ec_overwrites to true
```

4. Ceph File System のステータスを確認します。

#### 構文

```
ceph fs status FILE_SYSTEM_NAME
```

## 例

```
[root@mon ~]# ceph fs status cephfs-ec
cephfs-ec - 14 clients
=====
RANK STATE          MDS          ACTIVITY  DNS  INOS  DIRS  CAPS
0  active cephfs-ec.example.ooymyq Reqs: 0 /s 8231 8233 891 921
    POOL      TYPE  USED AVAIL
cephfs-metadata-ec metadata 787M 8274G
cephfs-data-ec    data 2360G 12.1T

    STANDBY MDS
cephfs-ec.example.irsrql
cephfs-ec.example.cauuaj
```

5. 既存の CephFS にイレイジャーコーディングのデータプールを追加します。

## 構文

```
ceph fs add_data_pool FILE_SYSTEM_NAME DATA_POOL_NAME
```

## 例

```
[root@mon ~]# ceph fs add_data_pool cephfs-ec cephfs-data-ec01
```

この例では、新しいデータプール **cephfs-data-ec01** を、既存のイレイジャーコーディングのファイルシステム **cephfs-ec** に追加します。

6. イレイジャーコーディングされたプールが Ceph File System に追加されていることを確認します。

## 構文

```
ceph fs status FILE_SYSTEM_NAME
```

## 例

```
[root@mon ~]# ceph fs status cephfs-ec
cephfs-ec - 14 clients
=====
RANK STATE          MDS          ACTIVITY  DNS  INOS  DIRS  CAPS
0  active cephfs-ec.example.ooymyq Reqs: 0 /s 8231 8233 891 921
    POOL      TYPE  USED AVAIL
cephfs-metadata-ec metadata 787M 8274G
cephfs-data-ec    data 2360G 12.1T
cephfs-data-ec01  data    0 12.1T

    STANDBY MDS
cephfs-ec.example.irsrql
cephfs-ec.example.cauuaj
```

7. 新しいディレクトリーにファイルレイアウトを設定します。

### 構文

```
mkdir PATH_TO_DIRECTORY  
setfattr -n ceph.dir.layout.pool -v DATA_POOL_NAME PATH_TO_DIRECTORY
```

### 例

```
[root@mon ~]# mkdir /mnt/cephfs/newdir  
[root@mon ~]# setfattr -n ceph.dir.layout.pool -v cephfs-data-ec01 /mnt/cephfs/newdir
```

この例では、`/mnt/cephfs/newdir` ディレクトリーで作成されるすべての新しいファイルは、ディレクトリーレイアウトを継承して、新たに追加したレイジャーコーディングプールにデータを配置します。

### 関連情報

- CephFS MDS の詳細は、[Red Hat Ceph Storage File System ガイドの TCeph File System Metadata Server](#) を参照してください。
- 詳細は、[Red Hat Ceph Storage File System ガイドの Ceph File Systems の作成](#) セクションを参照してください。
- 詳細は、[Red Hat Ceph Storage ストレージストラテジーガイドの イレイジャーコードプール](#) を参照してください。
- 詳細は、[Red Hat Ceph Storage ストレージストラテジーガイドの上書きによるイレイジャーコーディング](#) セクションを参照してください。

## 3.4. CEPH ファイルシステム用のクライアントユーザーの作成

Red Hat Ceph Storage は認証に `cephx` を使用します。これはデフォルトで有効になります。Ceph File System で `cephx` を使用するには、Ceph Monitor ノードで正しい承認機能を持つユーザーを作成し、そのキーを Ceph File System がマウントされるノードで利用できるようにします。

### 前提条件

- 稼働中の Red Hat Ceph Storage クラスタがある。
- Ceph Metadata Server デーモン (`ceph-mds`) のインストールおよび設定
- Ceph Monitor ノードへの root レベルのアクセス。
- Ceph クライアントノードへのルートレベルのアクセスがある。

### 手順

1. モニターノードの Cephadm シェルにログインします。

### 例

```
[root@host01 ~]# cephadm shell
```



2. Ceph Monitor ノードで、クライアントユーザーを作成します。

### 構文

```
ceph fs authorize FILE_SYSTEM_NAME client.CLIENT_NAME /DIRECTORY CAPABILITY
[/DIRECTORY CAPABILITY] PERMISSIONS ...
```

- クライアントを、ファイルシステム **cephfs\_a** の **temp** ディレクトリーでのみ書き込みするよう制限するには、以下を実行します。

### 例

```
[ceph: root@host01 /]# ceph fs authorize cephfs_a client.1 / r /temp rw

client.1
key = AQBSDfHcGZFUDRAAcKhG9CI2HPiDMMRv4DC43A==
```

- クライアントを **temp** ディレクトリーに完全に制限するには、root (/) ディレクトリーを削除します。

### 例

```
[ceph: root@host01 /]# ceph fs authorize cephfs_a client.1 /temp rw
```



### 注記

ファイルシステム名、**all** またはアスタリスク (\*) をファイルシステム名として指定することにより、すべてのファイルシステムへのアクセスが付与されます。通常、シェルから保護するには、アスタリスクを引用符で囲む必要があります。

3. 作成したキーを確認します。

### 構文

```
ceph auth get client.ID
```

### 例

```
[ceph: root@host01 /]# ceph auth get client.1

client.1
key = AQBSDfHcGZFUDRAAcKhG9CI2HPiDMMRv4DC43A==
caps mds = "allow r, allow rw path=/temp"
caps mon = "allow r"
caps osd = "allow rw tag cephfs data=cephfs_a"
```

4. キーリングをクライアントにコピーします。
  - a. Ceph Monitor ノードで、キーリングをファイルにエクスポートします。

### 構文

```
ceph auth get client.ID -o ceph.client.ID.keyring
```

### 例

```
[ceph: root@host01 /]# ceph auth get client.1 -o ceph.client.1.keyring
exported keyring for client.1
```

- b. Ceph Monitor ノードからクライアントノードの **/etc/ceph/** ディレクトリーに、クライアントキーリングをコピーします。

### 構文

```
scp /ceph.client.ID.keyring root@CLIENT_NODE_NAME:/etc/ceph/ceph.client.ID.keyring
```

**CLIENT\_NODE\_NAME** を Ceph クライアントのノード名または IP に置き換えます。

### 例

```
[ceph: root@host01 /]# scp /ceph.client.1.keyring
root@client01:/etc/ceph/ceph.client.1.keyring
```

5. クライアントノードから、キーリングファイルに適切なパーミッションを設定します。

### 構文

```
chmod 644 ceph.client.ID.keyring
```

### 例

```
[root@client01 ~]# chmod 644 /etc/ceph/ceph.client.1.keyring
```

## 関連情報

- 詳細は、Red Hat Ceph Storage 管理ガイドの [Ceph ユーザー管理](#) の章を参照してください。

## 3.5. CEPH FILE SYSTEM のカーネルクライアントとしてのマウント

Ceph File System (CephFS) は、システムの起動時に手動で、または自動でカーネルクライアントとしてマウントできます。



### 重要

Red Hat Enterprise Linux の他に、他の Linux ディストリビューションで実行しているクライアントは許可されますが、サポートされていません。これらのクライアントの使用時に、CephFS Metadata Server またはその他のストレージクラスターで問題が見つかる場合、Red Hat はそれらに対応します。原因がクライアント側にある場合は、Linux ディストリビューションのカーネルベンダーがこの問題に対応する必要があります。

## 前提条件

- Linux ベースのクライアントノードへのルートレベルのアクセス。

- Ceph Monitor ノードへの root レベルのアクセス。
- 既存の Ceph File System。

## 手順

1. Ceph Storage クラスターを使用するようにクライアントノードを設定します。
  - a. Red Hat Ceph Storage 7 Tools リポジトリを有効にします。

### Red Hat Enterprise Linux 9

```
[root@client01 ~]# subscription-manager repos --enable=rhceph-6-tools-for-rhel-9-x86_64-rpms
```

- b. **ceph-common** パッケージをインストールします。

```
[root@client01 ~]# dnf install ceph-common
```

- c. モニターノードの Cephadm シェルにログインします。

### 例

```
[root@host01 ~]# cephadm shell
```

- d. Ceph クライアントキーリングを Ceph Monitor ノードからクライアントノードにコピーします。

### 構文

```
scp /ceph.client.ID.keyring root@CLIENT_NODE_NAME:/etc/ceph/ceph.client.ID.keyring
```

**CLIENT\_NODE\_NAME** を Ceph クライアントのホスト名または IP アドレスに置き換えます。

### 例

```
[ceph: root@host01 /]# scp /ceph.client.1.keyring root@client01:/etc/ceph/ceph.client.1.keyring
```

- e. Ceph 設定ファイルを Monitor ノードからクライアントノードにコピーします。

### 構文

```
scp /etc/ceph/ceph.conf root@CLIENT_NODE_NAME:/etc/ceph/ceph.conf
```

**CLIENT\_NODE\_NAME** を Ceph クライアントのホスト名または IP アドレスに置き換えます。

### 例

```
[ceph: root@host01 /]# scp /etc/ceph/ceph.conf root@client01:/etc/ceph/ceph.conf
```

- f. クライアントノードから、設定ファイルに適切なパーミッションを設定します。

```
[root@client01 ~]# chmod 644 /etc/ceph/ceph.conf
```

- g. [automatically](#) または [manually](#) のいずれかを選択します。

## Manually Mounting

2. クライアントノードにマウントディレクトリーを作成します。

### 構文

```
mkdir -p MOUNT_POINT
```

### 例

```
[root@client01 ~]# mkdir -p /mnt/cephfs
```

3. Ceph ファイルシステムをマウントします。複数の Ceph Monitor アドレスを指定するには、**mount** コマンドでコンマで区切って、マウントポイントを指定し、クライアント名を設定します。



### 注記

Red Hat Ceph Storage 4.1 の時点で、**mount.ceph** はキーリングファイルを直接読み取りできます。そのため、シークレットファイルは不要になりました。**name=CLIENT\_ID** でクライアント ID を指定すると、**mount.ceph** は適切なキーリングファイルを検索します。

### 構文

```
mount -t ceph MONITOR-1_NAME:6789,MONITOR-2_NAME:6789,MONITOR-3_NAME:6789:/ MOUNT_POINT -o name=CLIENT_ID,fs=FILE_SYSTEM_NAME
```

### 例

```
[root@client01 ~]# mount -t ceph mon1:6789,mon2:6789,mon3:6789:/mnt/cephfs -o name=1,fs=cephfs01
```



### 注記

1つのホスト名が複数の IP アドレスに解決するように DNS サーバーを設定できます。次に、コンマ区切りリストを指定する代わりに、**mount** コマンドでその1つのホスト名を使用できます。



### 注記

また、Monitor ホスト名は **:/** に置き換えられ、**mount.ceph** は Ceph 設定ファイルを読み取り、どのモニターに接続するかを判断することもできます。



## 注記

**nowsync** オプションを設定して、Red Hat Ceph Storage クラスタでファイルの作成と削除を非同期的に実行できます。これにより、整合性に影響を与えずにこれらのシステムコールのラウンドトリップレイテンシーを回避することで、一部のワークロードのパフォーマンスが改善されます。**nowsync** オプションには、Red Hat Enterprise Linux 9.0 以降を搭載したカーネルクライアントが必要です。

## 例

```
[root@client01 ~]# mount -t ceph mon1:6789,mon2:6789,mon3:6789:/mnt/cephfs -o nowsync,name=1,fs=cephfs01
```

4. ファイルシステムが正常にマウントされていることを確認します。

## 構文

```
stat -f MOUNT_POINT
```

## 例

```
[root@client01 ~]# stat -f /mnt/cephfs
```

## 自動マウント

2. クライアントホストで、Ceph ファイルシステムをマウントする新しいディレクトリーを作成します。

## 構文

```
mkdir -p MOUNT_POINT
```

## 例

```
[root@client01 ~]# mkdir -p /mnt/cephfs
```

3. 以下のように **/etc/fstab** ファイルを編集します。

## 構文

```
#DEVICE                                PATH                                TYPE                                OPTIONS
MON_0_HOST:PORT,                        MOUNT_POINT                        ceph                                name=CLIENT_ID,
MON_1_HOST:PORT,
ceph.client_mountpoint=/VOL/SUB_VOL_GROUP/SUB_VOL/UID_SUB_VOL,
fs=FILE_SYSTEM_NAME,
MON_2_HOST:PORT:/q[_VOL_] /SUB_VOL/UID_SUB_VOL,
[ADDITIONAL_OPTIONS]
```

最初の列は、Ceph Monitor ホスト名とポート番号を設定します。

2 列目はマウントポイントを設定します。

3 列目は、ファイルシステムのタイプ (ここでは CephFS 用 **ceph**) を設定します。

4 番目のコラムは、**name** および **secretfile** オプションを使用してユーザー名やシークレットファイルなどのさまざまなオプションを設定します。**ceph.client\_mountpoint** オプションを使用して、特定のボリューム、サブボリューム、およびサブボリュームを設定できます。

ネットワークサブシステムの開始後にファイルシステムがマウントされ、ハングやネットワークの問題を回避するために、**\_netdev** オプションを設定します。アクセス時間情報がない場合は、**noatime** オプションを設定するとパフォーマンスが向上します。

5 番目のコラムと 6 番目のコラムをゼロに設定します。

## 例

```
#DEVICE      PATH          TYPE  OPTIONS      DUMP  FSCK
mon1:6789,   /mnt/cephfs   ceph  name=1,      0     0
mon2:6789,
ceph.client_mountpoint=/my_vol/my_sub_vol_group/my_sub_vol/0,
mon3:6789:/          fs=cephfs01,
                   _netdev,noatime
```

Ceph File System は、次のシステム起動時にマウントされます。



### 注記

Red Hat Ceph Storage 4.1 の時点で、**mount.ceph** はキーリングファイルを直接読み取りできます。そのため、シークレットファイルは不要になりました。**name=CLIENT\_ID** でクライアント ID を指定すると、**mount.ceph** は適切なキーリングファイルを検索します。



### 注記

また、Monitor ホスト名は **:/** に置き換えられ、**mount.ceph** は Ceph 設定ファイルを読み取り、どのモニターに接続するかを判断することもできます。

## 関連情報

- **mount(8)** man ページを参照してください。
- Ceph ユーザーの作成の詳細は、Red Hat Ceph Storage 管理ガイドの [Ceph ユーザー管理](#) の章を参照してください。
- 詳細は、Red Hat Ceph Storage File System ガイドの [Ceph File Systems の作成](#) セクションを参照してください。

## 3.6. CEPH FILE SYSTEM の FUSE クライアントとしてのマウント

Ceph File System (CephFS) は、システムの起動時に手動で、または自動で File System in User Space (FUSE) クライアントとしてマウントできます。

### 前提条件

- Linux ベースのクライアントノードへのルートレベルのアクセス。

- Ceph Monitor ノードへの root レベルのアクセス。
- 既存の Ceph File System。

## 手順

1. Ceph Storage クラスターを使用するようにクライアントノードを設定します。
  - a. Red Hat Ceph Storage 7 Tools リポジトリを有効にします。

### Red Hat Enterprise Linux 8

```
[root@client01 ~]# subscription-manager repos --enable=6-tools-for-rhel-8-x86_64-rpms
```

### Red Hat Enterprise Linux 9

```
[root@client01 ~]# subscription-manager repos --enable=6-tools-for-rhel-9-x86_64-rpms
```

- b. **ceph-fuse** パッケージをインストールします。

```
[root@client01 ~]# dnf install ceph-fuse
```

- c. モニターノードの Cephadm シェルにログインします。

### 例

```
[root@host01 ~]# cephadm shell
```

- d. Ceph クライアントキーリングを Ceph Monitor ノードからクライアントノードにコピーします。

### 構文

```
scp /ceph.client.ID.keyring root@CLIENT_NODE_NAME:/etc/ceph/ceph.client.ID.keyring
```

**CLIENT\_NODE\_NAME** を Ceph クライアントのホスト名または IP アドレスに置き換えます。

### 例

```
[ceph: root@host01 /]# scp /ceph.client.1.keyring  
root@client01:/etc/ceph/ceph.client.1.keyring
```

- e. Ceph 設定ファイルを Monitor ノードからクライアントノードにコピーします。

### 構文

```
scp /etc/ceph/ceph.conf root@CLIENT_NODE_NAME:/etc/ceph/ceph.conf
```

**CLIENT\_NODE\_NAME** を Ceph クライアントのホスト名または IP アドレスに置き換えます。

**例**

```
[ceph: root@host01 /]# scp /etc/ceph/ceph.conf root@client01:/etc/ceph/ceph.conf
```

- f. クライアントノードから、設定ファイルに適切なパーミッションを設定します。

```
[root@client01 ~]# chmod 644 /etc/ceph/ceph.conf
```

- g. [automatically](#) または [manually](#) のいずれかを選択します。

**Manually Mounting**

2. クライアントノードで、マウントポイントのディレクトリーを作成します。

**構文**

```
mkdir PATH_TO_MOUNT_POINT
```

**例**

```
[root@client01 ~]# mkdir /mnt/mycephfs
```

**注記**

MDS 機能で **path** オプションを使用した場合、マウントポイントは **path** で指定されたもの内になければなりません。

3. **ceph-fuse** ユーティリティーを使用して Ceph ファイルシステムをマウントします。

**構文**

```
ceph-fuse -n client.CLIENT_ID --client_fs FILE_SYSTEM_NAME MOUNT_POINT
```

**例**

```
[root@client01 ~]# ceph-fuse -n client.1 --client_fs cephfs01 /mnt/mycephfs
```

**注記**

**/etc/ceph/ceph.client.CLIENT\_ID.keyring** であるユーザーキーリングのデフォルト名と場所を使用しない場合は **--keyring** オプションを使用してユーザーキーリングへのパスを指定します。以下に例を示します。

**例**

```
[root@client01 ~]# ceph-fuse -n client.1 --keyring=/etc/ceph/client.1.keyring /mnt/mycephfs
```





### 注記

`-r` オプションを使用して、そのパスを `root` として処理するように指示します。

### 構文

```
ceph-fuse -n client.CLIENT_ID MOUNT_POINT -r PATH
```

### 例

```
[root@client01 ~]# ceph-fuse -n client.1 /mnt/cephfs -r /home/cephfs
```



### 注記

エビクトされた Ceph クライアントを自動的に再接続する場合は `--client_reconnect_stale=true` オプションを追加します。

### 例

```
[root@client01 ~]# ceph-fuse -n client.1 /mnt/cephfs --
client_reconnect_stale=true
```

4. ファイルシステムが正常にマウントされていることを確認します。

### 構文

```
stat -f MOUNT_POINT
```

### 例

```
[root@client01 ~]# stat -f /mnt/cephfs
```

## 自動マウント

2. クライアントノードで、マウントポイントのディレクトリを作成します。

### 構文

```
mkdir PATH_TO_MOUNT_POINT
```

### 例

```
[root@client01 ~]# mkdir /mnt/mycephfs
```



### 注記

MDS 機能で `path` オプションを使用した場合、マウントポイントは `path` で指定されたもの内になければなりません。

3. 以下のように `/etc/fstab` ファイルを編集します。

## 構文

```
#DEVICE          PATH          TYPE          OPTIONS          DUMP FSCK
HOST_NAME:PORT,  MOUNT_POINT fuse.ceph      ceph.id=CLIENT_ID,  0  0
HOST_NAME:PORT,
ceph.client_mountpoint=/VOL/SUB_VOL_GROUP/SUB_VOL/UID_SUB_VOL,
HOST_NAME:PORT:/
ceph.client_fs=FILE_SYSTEM_NAME,ceph.name=USERNAME,ceph.keyring=/etc/ceph/KEYRING_FILE,
[ADDITIONAL_OPTIONS]
```

最初の列は、Ceph Monitor ホスト名とポート番号を設定します。

2 列目は マウントポイントを設定します。

3 列目は、ファイルシステムのタイプ (ここでは CephFS 用 **fuse.ceph**) を設定します。

4 列目は、**ceph.name** および **ceph.keyring** オプションを使用して、ユーザー名やキーリングなどのさまざまなオプションを設定します。**ceph.client\_mountpoint** オプションを使用して、特定のボリューム、サブボリューム、およびサブボリュームを設定できます。アクセスする Ceph File System を指定するには、**ceph.client\_fs** オプションを使用します。ネットワークサブシステムの開始後にファイルシステムがマウントされ、ハングやネットワークの問題を回避するために、**\_netdev** オプションを設定します。アクセス時間情報が必要ない場合は、**noatime** オプションを設定するとパフォーマンスが向上します。エビクションの後に自動的に再接続する必要がある場合は、**client\_reconnect\_stale=true** オプションを設定します。

5 番目のコラムと 6 番目のコラム をゼロに設定します。

## 例

```
#DEVICE          PATH          TYPE          OPTIONS          DUMP FSCK
mon1:6789,       /mnt/mycephfs fuse.ceph      ceph.id=1,       0  0
mon2:6789,
ceph.client_mountpoint=/my_vol/my_sub_vol_group/my_sub_vol/0,
mon3:6789:/
ceph.client_fs=cephfs01,ceph.name=client.1,ceph.keyring=/etc/ceph/client1.keyring,
_netdev,defaults
```

Ceph File System は、次のシステム起動時にマウントされます。

## 関連情報

- **ceph-fuse(8)** man ページ
- Ceph ユーザーの作成の詳細は、Red Hat Ceph Storage 管理ガイドの [Ceph ユーザー管理](#) の章を参照してください。
- 詳細は、Red Hat Ceph Storage File System ガイドの [Ceph File Systems の作成](#) セクションを参照してください。

## 関連情報

- Ceph Metadata Server をインストールするには、「[Ceph Orchestrator を使用した MDS サービスの管理](#)」を参照してください。

- 詳細は、[「Ceph ファイルシステムの作成」](#) を参照してください。
- 詳細は、[「Ceph ファイルシステム用のクライアントユーザーの作成」](#) を参照してください。
- 詳細は、[「Ceph File System のカーネルクライアントとしてのマウント」](#) を参照してください。
- 詳細は、[「Ceph File System の FUSE クライアントとしてのマウント」](#) を参照してください。
- CephFS Metadata Server デーモンの設定に関する詳細は、[2章Ceph File System Metadata Server](#) を参照してください。

## 第4章 CEPH ファイルシステムボリューム、サブボリュームグループ、およびサブボリュームの管理

ストレージ管理者は、Red Hat の Ceph Container Storage Interface (CSI) を使用して Ceph File System (CephFS) エクスポートを管理できます。また、OpenStack のファイルシステムサービス (Manila) などの他のサービスは、一般的なコマンドラインインターフェイスを使用して対話できます。Ceph Manager デーモンの **volumes** モジュール (**ceph-mgr**) は、Ceph File Systems (CephFS) をエクスポートする機能を実装します。

Ceph Manager ボリュームモジュールは、以下のファイルシステムのエクスポートの抽象化を実装します。

- CephFS ボリューム
- CephFS サブボリュームグループ
- CephFS サブボリューム

### 4.1. CEPH FILE SYSTEM ボリューム

ストレージ管理者は、Ceph File System (CephFS) ボリュームの作成、リスト表示、および削除を行うことができます。CephFS ボリュームは、Ceph File Systems の抽象化です。

本セクションでは、以下を行う方法を説明します。

- [Ceph ファイルシステムボリュームを作成します。](#)
- [Ceph ファイルシステムボリュームを一覧表示します。](#)
- [Ceph ファイルシステムボリュームに関する情報を表示します。](#)
- [Ceph ファイルシステムボリュームを削除します。](#)

#### 4.1.1. Ceph ファイルシステムボリュームの作成

Ceph Orchestrator は、Ceph File System (CephFS) の Metadata Server (MDS) を作成する Ceph Manager のモジュールです。本項では、CephFS ボリュームを作成する方法を説明します。



#### 注記

これにより、データおよびメタデータプールと共に Ceph File System が作成されます。

#### 前提条件

- Ceph File System がデプロイされている稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。

#### 手順

- モニターノードに CephFS ボリュームを作成します。

## 構文

```
ceph fs volume create VOLUME_NAME
```

## 例

```
[ceph: root@host01 /]# ceph fs volume create cephfs
```

### 4.1.2. Ceph ファイルシステムボリュームの一覧表示

本項では、Ceph File System (CephFS) ボリュームをリスト表示する手順について説明します。

#### 前提条件

- Ceph File System がデプロイされている稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。
- CephFS ボリューム。

#### 手順

- CephFS ボリュームをリスト表示します。

## 例

```
[ceph: root@host01 /]# ceph fs volume ls
```

### 4.1.3. Ceph ファイルシステムボリュームに関する情報の表示

CephFS ボリュームのデータおよびメタデータプールの属性、保留中のサブボリュームの削除数など、Ceph ファイルシステム (CephFS) ボリュームに関する基本的な詳細を一覧表示できます。

#### 前提条件

- Ceph File System がデプロイされている稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。
- CephFS ボリュームが作成されている。

#### 手順

- CephFS ボリュームに関する情報を表示します。

## 構文

```
ceph fs volume info VOLUME_NAME
```

## 例

```
[ceph: root@host01 /]# ceph fs volume info cephfs
{
  "mon_addrs": [
    "192.168.1.7:40977",
  ],
  "pending_subvolume_deletions": 0,
  "pools": {
    "data": [
      {
        "avail": 106288709632,
        "name": "cephfs.cephfs.data",
        "used": 4096
      }
    ],
    "metadata": [
      {
        "avail": 106288709632,
        "name": "cephfs.cephfs.meta",
        "used": 155648
      }
    ]
  },
  "used_size": 0
}
```

**ceph fs volume info** コマンドの出力には以下が含まれます。

- **mon\_addrs**: モニターアドレスの一覧。
- **pending\_subvolume\_deletions**: 削除保留中のサブボリュームの数。
- **pools**: データおよびメタデータプールの属性。
  - **avail**: 利用可能な空き容量 (バイト単位)。
  - **name**: プールの名前。
  - **used**: 消費されたストレージの量 (バイト単位)。
- **used\_size**: CephFS ボリュームの現在の使用サイズ (バイト単位)。

#### 4.1.4. Ceph ファイルシステムボリュームの削除

Ceph Orchestrator は、Ceph File System (CephFS) の Metadata Server (MDS) を削除する Ceph Manager のモジュールです。本項では、Ceph File System (CephFS) ボリュームを削除する方法を説明します。

##### 前提条件

- Ceph File System がデプロイされている稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。

- CephFS ボリューム。

## 手順

1. **mon\_allow\_pool\_delete** オプションが **true** に設定されていない場合は、CephFS ボリュームを削除する前に **true** に設定します。

### 例

```
[ceph: root@host01 /]# ceph config set mon mon_allow_pool_delete true
```

2. CephFS ボリュームを削除します。

### 構文

```
ceph fs volume rm VOLUME_NAME [--yes-i-really-mean-it]
```

### 例

```
[ceph: root@host01 /]# ceph fs volume rm cephfs --yes-i-really-mean-it
```

## 4.2. CEPH FILE SYSTEM サブボリュームグループ

ストレージ管理者は、Ceph File System (CephFS) サブボリュームグループの作成、リスト表示、取得、および削除できます。CephFS サブボリュームグループは、サブボリュームのセット全体で、ファイルレイアウトなどのポリシーに影響を与えるディレクトリーレベルで抽象化されます。

Red Hat Ceph Storage 5.0 以降では、サブボリュームグループスナップショット機能はサポートされません。これらのサブボリュームグループの既存のスナップショットのみをリスト表示および削除できます。

本セクションでは、以下を行う方法を説明します。

- ファイルシステムのサブボリュームグループを作成します。
- ファイルシステムサブボリュームグループのクォータを設定および管理します。
- ファイルシステムのサブボリュームグループをリスト表示します。
- ファイルシステムのサブボリュームグループの絶対パスを取得します。
- ファイルシステムのサブボリュームグループのスナップショットをリスト表示します。
- ファイルシステムのサブボリュームグループのスナップショットを削除します。
- ファイルシステムのサブボリュームグループを削除します。

### 4.2.1. ファイルシステムのサブボリュームグループの作成

本項では、Ceph File System (CephFS) サブボリュームグループを作成する方法を説明します。



## 注記

subvolume グループを作成する場合は、そのデータプールのレイアウト (uid、gid、および file モード) を 8 進数で指定できます。デフォルトでは、サブボリュームグループは、親ディレクトリーの 8 進数ファイルモード '755'、uid '0'、gid '0'、およびデータプールレイアウトで作成されます。



## 注記

サブボリュームグループの作成中にクォータを設定するには、[ファイルシステムサブボリュームグループでのクォータの設定と管理](#) を参照してください。

## 前提条件

- Ceph File System がデプロイされている稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。

## 手順

- CephFS サブボリュームグループを作成します。

## 構文

```
ceph fs subvolumegroup create VOLUME_NAME GROUP_NAME [--pool_layout
DATA_POOL_NAME --uid UID --gid GID --mode OCTAL_MODE]
```

## 例

```
[ceph: root@host01 /]# ceph fs subvolumegroup create cephfs subgroup0
```

subvolume グループがすでに存在している場合でも、コマンドは成功します。

## 4.2.2. ファイルシステムサブボリュームグループのクォータの設定と管理

このセクションでは、Ceph ファイルシステム (CephFS) サブボリュームグループでクォータを設定および管理する方法について説明します。

## 前提条件

- Ceph File System がデプロイされている稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。

## 手順

1. サイズをバイト単位で指定して、サブボリュームグループの作成中にクォータを設定します。

## 構文

■



```
ceph fs subvolumegroup create VOLUME_NAME GROUP_NAME [--size SIZE_IN_BYTES]
[--pool_layout DATA_POOL_NAME] [--uid UID] [--gid GID] [--mode OCTAL_MODE]
```

## 例

```
[ceph: root@host01 /]# ceph fs subvolumegroup create cephfs subvolgroup_2 10737418240
```

- サブボリュームグループのサイズを変更します。

## 構文

```
ceph fs subvolumegroup resize VOLUME_NAME GROUP_NAME new_size [--no_shrink]
```

## 例

```
[ceph: root@host01 /]# ceph fs subvolumegroup resize cephfs subvolgroup_2 20737418240
[
  {
    "bytes_used": 10768679044
  },
  {
    "bytes_quota": 20737418240
  },
  {
    "bytes_pcent": "51.93"
  }
]
```

- サブボリュームグループのメタデータを取得します。

## 構文

```
ceph fs subvolumegroup info VOLUME_NAME GROUP_NAME
```

## 例

```
[ceph: root@host01 /]# ceph fs subvolumegroup info cephfs subvolgroup_2
{
  "atime": "2022-10-05 18:00:39",
  "bytes_pcent": "51.85",
  "bytes_quota": 20768679043,
  "bytes_used": 10768679044,
  "created_at": "2022-10-05 18:00:39",
  "ctime": "2022-10-05 18:21:26",
  "data_pool": "cephfs.cephfs.data",
  "gid": 0,
  "mode": 16877,
  "mon_addrs": [
    "60.221.178.236:1221",
    "205.64.75.112:1221",
    "20.209.241.242:1221"
  ],
}
```

```
"mtime": "2022-10-05 18:01:25",  
"uid": 0  
}
```

### 4.2.3. ファイルシステムのサブボリュームグループのリスト表示

本項では、Ceph File System (CephFS) サブボリュームグループをリスト表示する手順について説明します。

#### 前提条件

- Ceph File System がデプロイされている稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。
- CephFS サブボリュームグループ

#### 手順

- CephFS サブボリュームグループをリスト表示します。

#### 構文

```
ceph fs subvolumegroup ls VOLUME_NAME
```

#### 例

```
[ceph: root@host01 /]# ceph fs subvolumegroup ls cephfs
```

### 4.2.4. ファイルシステムのサブボリュームグループの絶対パスを取得中

本項では、Ceph File System (CephFS) サブボリュームの絶対パスを取得する方法を説明します。

#### 前提条件

- Ceph File System がデプロイされている稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。
- CephFS サブボリュームグループ

#### 手順

- CephFS サブボリュームの絶対パスを取得します。

#### 構文

```
ceph fs subvolumegroup getpath VOLUME_NAME GROUP_NAME
```

## 例

```
[ceph: root@host01 /]# ceph fs subvolume group getpath cephfs subgroup0
```

### 4.2.5. ファイルシステムのサブボリュームグループのスナップショットのリスト表示

本項では、Ceph File System (CephFS) サブボリュームグループのスナップショットをリスト表示する手順について説明します。

#### 前提条件

- Ceph File System がデプロイされている稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。
- CephFS サブボリュームグループ
- サブボリュームグループのスナップショット。

#### 手順

- CephFS サブボリュームグループのスナップショットをリスト表示します。

#### 構文

```
ceph fs subvolume group snapshot ls VOLUME_NAME GROUP_NAME
```

#### 例

```
[ceph: root@host01 /]# ceph fs subvolume group snapshot ls cephfs subgroup0
```

### 4.2.6. ファイルシステムのサブボリュームグループのスナップショットの削除

本セクションでは、Ceph File System (CephFS) サブボリュームグループのスナップショットを削除する手順について説明します。



#### 注記

**--force** フラグを使用すると、コマンドを正常に実行でき、スナップショットが存在しない場合には失敗します。

#### 前提条件

- Ceph File System がデプロイされている稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。
- Ceph File System ボリューム。

- サブボリュームグループのスナップショット。

## 手順

- CephFS サブボリュームグループのスナップショットを削除します。

## 構文

```
ceph fs subvolumegroup snapshot rm VOLUME_NAME GROUP_NAME SNAP_NAME [--force]
```

## 例

```
[ceph: root@host01 /]# ceph fs subvolumegroup snapshot rm cephfs subgroup0 snap0 --force
```

### 4.2.7. ファイルシステムのサブボリュームグループの削除

本項では、Ceph File System (CephFS) サブボリュームグループを削除する方法を説明します。



#### 注記

サブボリュームグループが空であるか、存在しないと、そのグループの削除に失敗します。**--force** フラグは、存在しないサブボリュームグループの削除を許可します。

## 前提条件

- Ceph File System がデプロイされている稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。
- CephFS サブボリュームグループ

## 手順

- CephFS サブボリュームグループを削除します。

## 構文

```
ceph fs subvolumegroup rm VOLUME_NAME GROUP_NAME [--force]
```

## 例

```
[ceph: root@host01 /]# ceph fs subvolumegroup rm cephfs subgroup0 --force
```

## 4.3. CEPH FILE SYSTEM サブボリューム

ストレージ管理者は、Ceph File System (CephFS) サブボリュームの作成、リスト表示、取得、メタデータの取得、削除が可能です。また、これらのサブボリュームのスナップショットの作成、リスト表示、および削除も可能です。CephFS サブボリュームは、独立した Ceph File Systems ディレクトリー

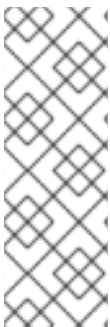
ツリーの抽象化です。

本セクションでは、以下を行う方法を説明します。

- ファイルシステムのサブボリュームの作成
- ファイルシステムのサブボリュームのリスト表示。
- ファイルシステムのサブボリュームのサイズ変更。
- ファイルシステムのサブボリュームの絶対パスの取得。
- ファイルシステムのサブボリュームのメタデータの取得。
- ファイルシステムのサブボリュームのスナップショットの作成。
- スナップショットからのサブボリュームのクローン作成
- ファイルシステムのサブボリュームのスナップショットのリスト表示。
- ファイルシステムサブボリュームのスナップショットのメタデータの取得。
- ファイルシステムのサブボリュームの削除。
- ファイルシステムのサブボリュームのスナップショットの削除。

#### 4.3.1. ファイルシステムのサブボリュームの作成

本項では、Ceph File System (CephFS) サブボリュームを作成する方法を説明します。



##### 注記

サブボリュームを作成する場合、そのサブボリュームグループ、データプールレイアウト、uid、gid、ファイルモード(8進数)、およびサイズ(バイト単位)を指定できます。サブボリュームは、**--namespace-isolated** オプションを指定することで、別の RADOS namespace に作成できます。デフォルトでは、サブボリュームはデフォルトの subvolume グループ内に作成され、サブボリュームグループの8進数ファイルモード '755'、サブボリュームグループの gid、親ディレクトリーのデータプールレイアウトとサイズ制限がありません。

##### 前提条件

- Ceph File System がデプロイされている稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。

##### 手順

- CephFS サブボリュームを作成します。

##### 構文

```
ceph fs subvolume create VOLUME_NAME SUBVOLUME_NAME [--size SIZE_IN_BYTES
--group_name SUBVOLUME_GROUP_NAME --pool_layout DATA_POOL_NAME --uid
_UID --gid GID --mode OCTAL_MODE] [--namespace-isolated]
```

### 例

```
[root@mon ~]# ceph fs subvolume create cephfs sub0 --group_name subgroup0 --
namespace-isolated
```

subvolume がすでに存在している場合でも、コマンドは成功します。

## 4.3.2. ファイルシステムのサブボリュームのリスト表示

本項では、Ceph File System (CephFS) サブボリュームをリスト表示する手順について説明します。

### 前提条件

- Ceph File System がデプロイされている稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。
- CephFS サブボリューム。

### 手順

- CephFS サブボリュームをリスト表示します。

### 構文

```
ceph fs subvolume ls VOLUME_NAME [--group_name SUBVOLUME_GROUP_NAME]
```

### 例

```
[root@mon ~]# ceph fs subvolume ls cephfs --group_name subgroup0
```

## 4.3.3. ファイルシステムのサブボリュームのサイズ変更

本項では、Ceph File System (CephFS) サブボリュームのサイズを変更する方法を説明します。



### 注記

**ceph fs subvolume resize** コマンドは、**new\_size** で指定されたサイズでサブボリュームのクォータのサイズを変更します。**--no\_shrink** フラグは、サブボリュームが現在使用されているサブボリュームのサイズを下回って縮小するのを防ぎます。サブボリュームは、**f** または **infinite** を **new\_size** として渡すと、無限にリサイズできます。

### 前提条件

- Ceph File System がデプロイされている稼働中の Red Hat Ceph Storage クラスタ。

- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。
- CephFS サブボリューム。

#### 手順

- CephFS サブボリュームのサイズを変更します。

#### 構文

```
ceph fs subvolume resize VOLUME_NAME SUBVOLUME_NAME NEW_SIZE [--group_name SUBVOLUME_GROUP_NAME] [--no_shrink]
```

#### 例

```
[root@mon ~]# ceph fs subvolume resize cephfs sub0 1024000000 --group_name subgroup0 --no_shrink
```

### 4.3.4. ファイルシステムのサブボリュームの絶対パスを取得中

本項では、Ceph File System (CephFS) サブボリュームの絶対パスを取得する方法を説明します。

#### 前提条件

- Ceph File System がデプロイされている稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。
- CephFS サブボリューム。

#### 手順

- CephFS サブボリュームの絶対パスを取得します。

#### 構文

```
ceph fs subvolume getpath VOLUME_NAME SUBVOLUME_NAME [--group_name _SUBVOLUME_GROUP_NAME]
```

#### 例

```
[root@mon ~]# ceph fs subvolume getpath cephfs sub0 --group_name subgroup0
```

### 4.3.5. ファイルシステムのサブボリュームのメタデータの取得

本項では、Ceph File System (CephFS) サブボリュームのメタデータを取得する方法について説明します。

## 前提条件

- Ceph File System がデプロイされている稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。
- CephFS サブボリューム。

## 手順

- CephFS サブボリュームのメタデータを取得します。

### 構文

```
ceph fs subvolume info VOLUME_NAME SUBVOLUME_NAME [--group_name SUBVOLUME_GROUP_NAME]
```

### 例

```
[root@mon ~]# ceph fs subvolume info cephfs sub0 --group_name subgroup0
```

### 出力例

```
# ceph fs subvolume info cephfs sub0
{
  "atime": "2023-07-14 08:52:46",
  "bytes_pcent": "0.00",
  "bytes_quota": 1024000000,
  "bytes_used": 0,
  "created_at": "2023-07-14 08:52:46",
  "ctime": "2023-07-14 08:53:54",
  "data_pool": "cephfs.cephfs.data",
  "features": [
    "snapshot-clone",
    "snapshot-autoprotect",
    "snapshot-retention"
  ],
  "flavor": "2",
  "gid": 0,
  "mode": 16877,
  "mon_addrs": [
    "10.0.208.172:6789",
    "10.0.211.197:6789",
    "10.0.209.212:6789"
  ],
  "mtime": "2023-07-14 08:52:46",
  "path": "/volumes/_nogroup/sub0/834c5cbc-f5db-4481-80a3-aca92ff0e7f3",
  "pool_namespace": "",
  "state": "complete",
  "type": "subvolume",
  "uid": 0
}
```



出力形式は JSON で、以下のフィールドが含まれます。

- **atime**: "YYYY-MM-DD HH:MM:SS" 形式のサブボリュームパスのアクセス時間。
- **bytes\_pcent**: クォータが設定されている場合に使用するクォータ。それ以外の場合は "undefined" を表示します。
- **bytes\_quota**: クォータが設定されている場合のクォータサイズ (バイト単位)。それ以外の場合は infinite を表示します。
- **bytes\_used**: サブボリュームの現在の使用サイズ (バイト単位)。
- **created\_at**: サブボリュームを作成する時間 ("YYYY-MM-DD HH:MM:SS" 形式)。
- **ctime**: サブボリュームの時間を "YYYY-MM-DD HH:MM:SS" 形式で変更します。
- **data\_pool**: サブボリュームが属するデータプール。
- **features**: "snapshot-clone"、"snapshot-autoprotect"、"snapshot-retention" などのサブボリュームでサポートされる機能。
- **flavor**: サブボリュームのバージョン。バージョン 1 の場合は **1**、バージョン 2 の場合は **2** です。
- **gid**: サブボリュームパスのグループ ID。
- **モード**: サブボリュームパスのモード。
- **mon\_addrs**: モニターアドレスのリスト。
- **mtime**: サブボリュームパスの変更時間 ("YYYY-MM-DD HH:MM:SS" 形式)。
- **path**: サブボリュームの絶対パス。
- **pool\_namespace**: サブボリュームの RADOS 名前空間。
- **state**: サブボリュームの現在の状態 (例:complete または snapshot-retained)
- **type**: クローンかサブボリュームかを示すサブボリュームタイプ。
- **uid**: サブボリュームパスのユーザー ID。

#### 4.3.6. ファイルシステムのサブボリュームのスナップショットの作成

本項では、Ceph File System (CephFS) サブボリュームのスナップショットを作成する方法を説明します。

##### 前提条件

- Ceph File System がデプロイされている稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。
- CephFS サブボリューム。

- クライアントの読み取り (**r**) および書き込み (**w**) 機能のほかに、クライアントはファイルシステム内のディレクトリパスに **s** フラグも必要になります。

## 手順

- s** フラグがディレクトリに設定されていることを確認します。

### 構文

```
ceph auth get CLIENT_NAME
```

### 例

```
[root@mon ~]# ceph auth get client.0
[client.0]
key = AQAz7EVWygILFRAAdlcuJ12opU/JKyfFmxhuaw==
caps mds = "allow rw, allow rws path=/bar" 1
caps mon = "allow r"
caps osd = "allow rw tag cephfs data=cephfs_a" 2
```

- 1 2** この例では、**client.0** はファイルシステムの **cephfs\_a** の **bar** ディレクトリにスナップショットを作成または削除することができます。

- Ceph File System サブボリュームのスナップショットを作成します。

### 構文

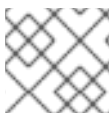
```
ceph fs subvolume snapshot create VOLUME_NAME SUBVOLUME_NAME SNAP_NAME
[--group_name GROUP_NAME]
```

### 例

```
[root@mon ~]# ceph fs subvolume snapshot create cephfs sub0 snap0 --group_name
subgroup0
```

## 4.3.7. スナップショットからのサブボリュームのクローン作成

サブボリュームスナップショットのクローン作成により、サブボリュームを作成できます。これは、スナップショットからサブボリュームにデータをコピーする非同期操作です。



### 注記

非常に大規模なデータセットの場合、クローン作成は非効率的です。

### 前提条件

- Ceph File System がデプロイされている稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。

- スナップショットの作成や書き込み機能の削除に加えて、クライアントはファイルシステム内のディレクトリーパスに **s** フラグが必要です。

## 構文

### CLIENT\_NAME

```
key = AQAz7EVWYgILFRAAdIcuJ12opU/JKyfFmxhuaw==
caps mds = allow rw, allow rws path=DIRECTORY_PATH
caps mon = allow r
caps osd = allow rw tag cephfs data=DIRECTORY_NAME
```

以下の例では、**client.0** はファイルシステム **cephfs\_a** の **bar** ディレクトリーにスナップショットを作成または削除することができます。

## 例

```
[client.0]
key = AQAz7EVWYgILFRAAdIcuJ12opU/JKyfFmxhuaw==
caps mds = "allow rw, allow rws path=/bar"
caps mon = "allow r"
caps osd = "allow rw tag cephfs data=cephfs_a"
```

## 手順

1. Ceph File System (CephFS) ボリュームを作成します。

## 構文

```
ceph fs volume create VOLUME_NAME
```

## 例

```
[root@mon ~]# ceph fs volume create cephfs
```

これにより、CephFS ファイルシステム、そのデータおよびメタデータプールが作成されます。

2. サブボリュームグループを作成します。デフォルトでは、サブボリュームグループは、モード '755' および、親ディレクトリーのデータプールレイアウトで作成されます。

## 構文

```
ceph fs subvolumegroup create VOLUME_NAME GROUP_NAME [--pool_layout
DATA_POOL_NAME --uid UID --gid GID --mode OCTAL_MODE]
```

## 例

```
[root@mon ~]# ceph fs subvolumegroup create cephfs subgroup0
```

3. サブボリュームを作成します。デフォルトでは、サブボリュームはデフォルトの subvolume グループ内に作成され、サブボリュームグループの 8 進数ファイルモード '755'、サブボリュームグループの gid、親ディレクトリーのデータプールレイアウトとサイズ制限がありません。

## 構文

```
ceph fs subvolume create VOLUME_NAME SUBVOLUME_NAME [--size SIZE_IN_BYTES
--group_name SUBVOLUME_GROUP_NAME --pool_layout DATA_POOL_NAME --uid
_UID --gid GID --mode OCTAL_MODE]
```

## 例

```
[root@mon ~]# ceph fs subvolume create cephfs sub0 --group_name subgroup0
```

- サブボリュームのスナップショットを作成します。

## 構文

```
ceph fs subvolume snapshot create VOLUME_NAME SUBVOLUME_NAME SNAP_NAME
[--group_name SUBVOLUME_GROUP_NAME]
```

## 例

```
[root@mon ~]# ceph fs subvolume snapshot create cephfs sub0 snap0 --group_name
subgroup0
```

- クローン操作を開始します。



## 注記

デフォルトでは、クローン作成されたサブボリュームがデフォルトのグループに作成されます。

- ソースサブボリュームとターゲットのクローンがデフォルトのグループにある場合は、以下のコマンドを実行します。

## 構文

```
ceph fs subvolume snapshot clone VOLUME_NAME SUBVOLUME_NAME
SNAP_NAME TARGET_CLONE_NAME
```

## 例

```
[root@mon ~]# ceph fs subvolume snapshot clone cephfs sub0 snap0 clone0
```

- ソースサブボリュームがデフォルト以外のグループにある場合は、以下のコマンドでソース subvolume グループを指定します。

## 構文

```
ceph fs subvolume snapshot clone VOLUME_NAME SUBVOLUME_NAME
SNAP_NAME TARGET_CLONE_NAME --group_name SUBVOLUME_GROUP_NAME
```

## 例

```
[root@mon ~]# ceph fs subvolume snapshot clone cephfs sub0 snap0 clone0 --
group_name subgroup0
```

- c. ターゲットのクローンがデフォルト以外のグループにある場合は、以下のコマンドでターゲットグループを指定します。

### 構文

```
ceph fs subvolume snapshot clone VOLUME_NAME SUBVOLUME_NAME
SNAP_NAME TARGET_CLONE_NAME --target_group_name
SUBVOLUME_GROUP_NAME
```

### 例

```
[root@mon ~]# ceph fs subvolume snapshot clone cephfs sub0 snap0 clone0 --
target_group_name subgroup1
```

6. clone 操作のステータスを確認します。

### 構文

```
ceph fs clone status VOLUME_NAME CLONE_NAME [--group_name
TARGET_GROUP_NAME]
```

### 例

```
[root@mon ~]# ceph fs clone status cephfs clone0 --group_name subgroup1
{
  "status": {
    "state": "complete"
  }
}
```

### 関連情報

- Red Hat Ceph Storage 管理ガイドの [Ceph ユーザーの管理](#) セクションを参照してください。

#### 4.3.8. ファイルシステムのサブボリュームのスナップショットのリスト表示

本項では、Ceph File システム (CephFS) サブボリュームのスナップショットをリスト表示する手順について説明します。

### 前提条件

- Ceph File System がデプロイされている稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。
- CephFS サブボリューム。

- サブボリュームのスナップショット。

## 手順

- CephFS サブボリュームのスナップショットをリスト表示します。

### 構文

```
ceph fs subvolume snapshot ls VOLUME_NAME SUBVOLUME_NAME [--group_name  
SUBVOLUME_GROUP_NAME]
```

### 例

```
[root@mon ~]# ceph fs subvolume snapshot ls cephfs sub0 --group_name subgroup0
```

## 4.3.9. ファイルシステムサブボリュームのスナップショットのメタデータの取得。

本項では、Ceph File System (CephFS) サブボリュームのスナップショットのメタデータを取得する手順について説明します。

### 前提条件

- Ceph FS がデプロイされている稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。
- CephFS サブボリューム。
- サブボリュームのスナップショット。

## 手順

1. CephFS サブボリュームのスナップショットのメタデータを取得します。

### 構文

```
ceph fs subvolume snapshot info VOLUME_NAME SUBVOLUME_NAME SNAP_NAME [--  
group_name SUBVOLUME_GROUP_NAME]
```

### 例

```
[root@mon ~]# ceph fs subvolume snapshot info cephfs sub0 snap0 --group_name  
subgroup0
```

### 出力例

```
{  
  "created_at": "2022-05-09 06:18:47.330682",  
  "data_pool": "cephfs_data",
```

```

    "has_pending_clones": "no",
    "size": 0
  }

```

出力形式は JSON で、以下のフィールドが含まれます。

- **created\_at**: スナップショットの作成時間 (YYYY-MM-DD HH:MM:SS:ffffff 形式)。
- **data\_pool**: スナップショットが属するデータプール。
- **has\_pending\_clones**: スナップショットの選択が進行中の場合は "yes" そうでなければ "no"。
- **サイズ**: スナップショットサイズ (バイト単位)。

#### 4.3.10. ファイルシステムのサブボリュームの削除

本項では、Ceph File System (CephFS) サブボリュームを削除する方法を説明します。



#### 注記

**ceph fs subvolume rm** コマンドは、2つのステップでサブボリュームとその内容を削除します。まず、サブボリュームをゴミ箱フォルダーに移動し、そのコンテンツを非同期的にパージします。

サブボリュームは、**--retain-snapshots** オプションを使用してサブボリュームの既存のスナップショットを保持できます。スナップショットが保持されると、そのサブボリュームは、保持済みスナップショットが関係するすべての操作に対して空であると見なされます。保持されるスナップショットは、サブボリュームを再作成するクローンソースとして使用するか、新しいサブボリュームにクローンを作成します。

#### 前提条件

- Ceph File System がデプロイされている稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。
- CephFS サブボリューム。

#### 手順

1. CephFS サブボリュームを削除します。

#### 構文

```

ceph fs subvolume rm VOLUME_NAME SUBVOLUME_NAME [--group_name
SUBVOLUME_GROUP_NAME] [--force] [--retain-snapshots]

```

#### 例

```

[root@mon ~]# ceph fs subvolume rm cephfs sub0 --group_name subgroup0 --retain-
snapshots

```

2. 保持されるスナップショットからサブボリュームを再作成するには、以下を実行します。

### 構文

```
ceph fs subvolume snapshot clone VOLUME_NAME DELETED_SUBVOLUME
RETAINED_SNAPSHOT NEW_SUBVOLUME --group_name
SUBVOLUME_GROUP_NAME --target_group_name
SUBVOLUME_TARGET_GROUP_NAME
```

- **NEW\_SUBVOLUME**は、以前に削除された同じサブボリュームにするか、新しいサブボリュームにクローンを作成します。

### 例

```
[root@mon ~]# ceph fs subvolume snapshot clone cephfs sub0 snap0 sub1 --group_name
subgroup0 --target_group_name subgroup0
```

#### 4.3.11. ファイルシステムのサブボリュームのスナップショットの削除

本セクションでは、Ceph File System (CephFS) サブボリュームグループのスナップショットを削除する手順について説明します。



#### 注記

**--force** フラグを使用すると、コマンドを正常に実行でき、スナップショットが存在しない場合には失敗します。

#### 前提条件

- Ceph File System がデプロイされている稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。
- Ceph File System ボリューム。
- サブボリュームグループのスナップショット。

#### 手順

- CephFS サブボリュームのスナップショットを削除します。

### 構文

```
ceph fs subvolume snapshot rm VOLUME_NAME SUBVOLUME_NAME SNAP_NAME [--
group_name GROUP_NAME --force]
```

### 例

```
[root@mon ~]# ceph fs subvolume snapshot rm cephfs sub0 snap0 --group_name
subgroup0 --force
```



## 関連情報

- Red Hat Ceph Storage 管理ガイドの [Ceph ユーザーの管理](#) セクションを参照してください。

## 4.4. CEPH FILE SYSTEM サブボリュームのメタデータ情報

ストレージ管理者は、Ceph File System (CephFS) サブボリュームのメタデータ情報を設定、取得、一覧表示、および削除できます。

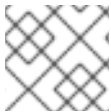
カスタムメタデータは、ユーザーがメタデータをサブボリュームに保存するためのものです。ユーザーは **xattr** と同様のキーと値のペアを Ceph ファイルシステムに保存できます。

本セクションでは、以下を行う方法を説明します。

- [ファイルシステムサブボリュームでのカスタムメタデータの設定](#)
- [ファイルシステムサブボリュームでのカスタムメタデータの取得](#)
- [ファイルシステムサブボリュームでのカスタムメタデータの一覧表示](#)
- [ファイルシステムサブボリュームからのカスタムメタデータの削除](#)

### 4.4.1. ファイルシステムサブボリュームでのカスタムメタデータの設定

ファイルシステムサブボリュームにカスタムメタデータをキーと値のペアとして設定できます。



#### 注記

**key\_name** がすでに存在する場合、古い値は新しい値に置き換えられます。



#### 注記

**KEY\_NAME** と **VALUE** は、python の **string.printable** で指定されている ASCII 文字の文字列である必要があります。**KEY\_NAME** は大文字と小文字が区別されず、常に小文字で保存されます。



#### 重要

サブボリュームのカスタムメタデータは、サブボリュームのスナップショット作成時に保持されないため、サブボリュームスナップショットのクローン作成時にも保持されません。

## 前提条件

- 稼働中の Red Hat Ceph Storage クラスタがある。
- Ceph ファイルシステム (CephFS)、CephFS ボリューム、サブボリュームグループ、およびサブボリュームが作成されました。

## 手順

1. CephFS サブボリュームにメタデータを設定します。

### 構文

```
ceph fs subvolume metadata set VOLUME_NAME SUBVOLUME_NAME KEY_NAME
VALUE [--group_name SUBVOLUME_GROUP_NAME]
```

**例**

```
[ceph: root@host01 /]# ceph fs subvolume metadata set cephfs sub0 test_meta cluster --
group_name subgroup0
```

- オプション: **KEY\_NAME** にスペースを含むカスタムメタデータを設定します。

**例**

```
[ceph: root@host01 /]# ceph fs subvolume metadata set cephfs sub0 "test meta" cluster --
group_name subgroup0
```

これにより、**KEY\_NAME** を持つ別のメタデータが **VALUE cluster** の **test meta** として作成されます。

- オプション: 同じメタデータに別の値を設定することもできます。

**例**

```
[ceph: root@host01 /]# ceph fs subvolume metadata set cephfs sub0 "test_meta" cluster2 --
group_name subgroup0
```

#### 4.4.2. ファイルシステムサブボリュームでのカスタムメタデータの取得

ボリューム内、およびオプションで特定のサブボリュームグループ内の Ceph ファイルシステム (CephFS) のカスタムメタデータ (キーと値のペア) を取得できます。

##### 前提条件

- 稼働中の Red Hat Ceph Storage クラスタがある。
- CephFS ボリューム、サブボリュームグループ、およびサブボリュームが作成されました。
- CephFS サブボリュームで作成されたカスタムメタデータ。

##### 手順

- CephFS サブボリュームのメタデータを取得します。

**構文**

```
ceph fs subvolume metadata get VOLUME_NAME SUBVOLUME_NAME KEY_NAME [--
group_name SUBVOLUME_GROUP_NAME]
```

**例**

```
[ceph: root@host01 /]# ceph fs subvolume metadata get cephfs sub0 test_meta --
group_name subgroup0
```

cluster

### 4.4.3. ファイルシステムサブボリュームでのカスタムメタデータの一覧表示

ボリューム内の Ceph ファイルシステム (CephFS) のキーに関連付けられたカスタムメタデータを一覧表示し、オプションで特定のサブボリュームグループ内に一覧表示することもできます。

#### 前提条件

- 稼働中の Red Hat Ceph Storage クラスタがある。
- CephFS ボリューム、サブボリュームグループ、およびサブボリュームが作成されました。
- CephFS サブボリュームで作成されたカスタムメタデータ。

#### 手順

- CephFS サブボリュームのメタデータを一覧表示します。

#### 構文

```
ceph fs subvolume metadata ls VOLUME_NAME SUBVOLUME_NAME [--group_name SUBVOLUME_GROUP_NAME]
```

#### 例

```
[ceph: root@host01 /]# ceph fs subvolume metadata ls cephfs sub0
{
  "test_meta": "cluster"
}
```

### 4.4.4. ファイルシステムサブボリュームからのカスタムメタデータの削除

ボリューム内、およびオプションで特定のサブボリュームグループ内の Ceph ファイルシステム (CephFS) のカスタムメタデータ (キーと値のペア) を削除できます。

#### 前提条件

- 稼働中の Red Hat Ceph Storage クラスタがある。
- CephFS ボリューム、サブボリュームグループ、およびサブボリュームが作成されました。
- CephFS サブボリュームで作成されたカスタムメタデータ。

#### 手順

1. CephFS サブボリュームのカスタムメタデータを削除します。

#### 構文

```
ceph fs subvolume metadata rm VOLUME_NAME SUBVOLUME_NAME KEY_NAME [--group_name SUBVOLUME_GROUP_NAME]
```

**例**

```
[ceph: root@host01 /]# ceph fs subvolume metadata rm cephfs sub0 test_meta --  
group_name subgroup0
```

2. メタデータを一覧表示します。

**例**

```
[ceph: root@host01 /]# ceph fs subvolume metadata ls cephfs sub0  
  
{}
```

## 第5章 CEPH FILE SYSTEM 管理

ストレージ管理者は、以下のような共通の Ceph File System (CephFS) の管理タスクを実行することができます。

- CephFS メトリックのリアルタイム監視については、[「cephfs-top ユーティリティーの使用」](#)を参照してください。
- 特定の MDS ランクにディレクトリーをマッピングする場合は、[「ディレクトリーツリーから Metadata Server デーモンのランクへのマッピング」](#)を参照してください。
- MDS ランクからディレクトリーの関連付けを解除する場合は、[「Metadata Server デーモンのランクからディレクトリーツリーの解除」](#)を参照してください。
- 新しいデータプールの追加は、[「データプールの追加」](#)を参照してください。
- クォータの使用は、[7章 Ceph File システムのクォータ](#)を参照してください。
- ファイルとディレクトリーレイアウトを使用する場合は、[8章 ファイルとディレクトリーのレイアウト](#)を参照してください。
- Ceph File System の削除中は [「Ceph ファイルシステムの削除」](#)を参照してください。
- クライアント機能は、[「クライアント機能」](#)を参照してください。
- `ceph mds fail` コマンドの使用は、[「ceph mds fail コマンドの使用」](#)を参照してください。
- CephFS クライアントを手動でエビクトします。詳細は [「Ceph File System クライアントの手動エビクト」](#)を参照してください。

### 前提条件

- 実行中、および正常な Red Hat Ceph Storage クラスタ
- Ceph Metadata Server デーモン (`ceph-mds`) のインストールおよび設定
- Ceph ファイルシステムを作成してマウントします。

### 5.1. CEPHFS-TOP ユーティリティーの使用

Ceph File System (CephFS) は、リアルタイムに Ceph File Systems でメトリックを表示する `top` のようなユーティリティーを提供します。`cephfs-top` ユーティリティーは、Ceph Manager の `stats` モジュールを使用してクライアントパフォーマンスメトリックを取得して表示する `curses` ベースの Python スクリプトです。

現在、`cephfs-top` ユーティリティーは約 10,000 のクライアントをサポートしています。



#### 注記

現在、Red Hat Enterprise Linux 9.2 カーネルで、パフォーマンス統計がすべて利用できるわけではありません。`cephfs-top` は Red Hat Enterprise Linux 9 以降でサポートされており、Red Hat Enterprise Linux の標準ターミナルの1つを使用します。



## 重要

**cephfs-top** ユーティリティーと互換性のある Python の最小バージョンは 3.6.0 です。

### 前提条件

- 正常かつ稼働中の Red Hat Ceph Storage クラスタ
- Ceph File System のデプロイメント
- Ceph クライアントノードへのルートレベルのアクセスがある。
- **cephfs-top** パッケージのインストール

### 手順

1. Red Hat Ceph Storage 6 ツールリポジトリがまだ有効になっていない場合は、有効にします。

#### Red Hat Enterprise Linux 9

```
[root@client ~]# subscription-manager repos --enable=rhceph-6-tools-for-rhel-9-x86_64-rpms
```

2. **cephfs-top** パッケージをインストールします。

#### 例

```
[root@client ~]# dnf install cephfs-top
```

3. Ceph Manager **stats** プラグインを有効にします。

#### 例

```
[root@client ~]# ceph mgr module enable stats
```

4. Ceph ユーザー **client.fstop** を作成します。

#### 例

```
[root@client ~]# ceph auth get-or-create client.fstop mon 'allow r' mds 'allow r' osd 'allow r' mgr 'allow r' > /etc/ceph/ceph.client.fstop.keyring
```



## 注記

必要に応じて、**--id** 引数を使用して、**client.fstop** 以外の別の Ceph ユーザーを指定します。

5. **cephfs-top** ユーティリティーを起動します。

#### 例

```
[root@client ~]# cephfs-top
```

```
cephfs-top - Wed Nov 30 15:26:05 2022
```

```
All Filesystem Info
```

```
Total Client(s): 4 - 3 FUSE, 1 kclient, 0 libcephfs
```

```
COMMANDS: m - select a filesystem | s - sort menu | l - limit number of clients | r - reset to default | q - quit
```

```
client_id mount_root chit(%) dlease(%) ofiles oicaps oinodes rtio(MB) raio(MB) rsp(MB/s)
wtio(MB) waio(MB) wsp(MB/s) rlatavg(ms) rlatsd(ms) wlatavg(ms) wlatsd(ms) mlatavg(ms)
mlatsd(ms) mount_point@host/addr
```

```
Filesystem: cephfs1 - 2 client(s)
```

```
4500 / 100.0 100.0 0 751 0 0.0 0.0 0.0 578.13 0.03 0.0
N/A N/A N/A N/A N/A N/A N/A@example/192.168.1.4
4501 / 100.0 0.0 0 1 0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.41 0.0 /mnt/cephfs2@example/192.168.1.4
```

```
Filesystem: cephfs2 - 2 client(s)
```

```
4512 / 100.0 0.0 0 1 0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.4 0.0 /mnt/cephfs3@example/192.168.1.4
4518 / 100.0 0.0 0 1 0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.52 0.0 /mnt/cephfs4@example/192.168.1.4
```

### 5.1.1. cephfs-top ユーティリティーの対話型コマンド

特定のファイルシステムを選択し、**cephfs-top** ユーティリティーの対話型コマンドを使用して、そのファイルシステムに関連するメトリックを表示します。

**m**

#### 説明

ファイルシステムの選択: 選択するファイルシステムのメニューを表示します。

**q**

#### 説明

終了: すべてのファイルシステム情報が表示されたホーム画面にいる場合は、ユーティリティーを終了します。ホーム画面にいない場合は、ホーム画面にリダイレクトされます。

**s**

#### 説明

ソートフィールドの選択: ソートフィールドを指定します。'cap\_hit' がデフォルトです。

**l**

#### 説明

クライアントの制限: 表示するクライアント数の制限を設定します。

**r**

#### 説明

リセット: ソートフィールドと制限値をデフォルトにリセットします。

メトリクス表示は、矢印キー、PgUp/PgDn、Home/End、およびマウスを使用してスクロールできます。

### ファイルシステム選択メニューの開始と終了の例

```
[root@client ~]# m

          Filesystems
Press "q" to go back to home (all filesystem info) screen

          cephfs01
          cephfs02
[root@client ~]# q

cephfs-top - Thu Oct 20 07:29:35 2022
Total Client(s): 3 - 2 FUSE, 1 kclient, 0 libcephfs
```

### 5.1.2. cephfs-top ユーティリティーのオプション

**cephfs-top** ユーティリティーコマンドをさまざまなオプションとともに使用できます。

#### 例

```
[root@client ~]# cephfs-top --selftest
selftest ok
```

#### **--cluster NAME\_OF\_THE\_CLUSTER**

##### 説明

このオプションを使用すると、デフォルト以外のクラスター名に接続できます。デフォルト名は **ceph** です。

#### **--id user**

##### 説明

これは、Ceph クラスターに接続するクライアントであり、デフォルトでは **fstop** です。

#### **--selftest**

##### 説明

このオプションを使用すると、セルフテストを実行できます。このモードは、**stats** モジュールの健全性チェックを実行します。

#### **--conffile PATH\_TO\_THE\_CONFIGURATION\_FILE**

##### 説明

このオプションを使用すると、Ceph クラスター設定ファイルへのパスを指定できます。

#### **-d/--delay INTERVAL\_IN\_SECONDS**



**説明**

**cephfs-top** ユーティリティーは、デフォルトで統計を毎秒更新します。このオプションを使用すると、更新間隔を変更できます。

**注記**

間隔は1秒以上にする必要があります。小数秒は尊重されます。

**--dump****説明**

このオプションを使用すると、`curses` 表示を作成せずにメトリクスを標準出力にダンプできます。

**--dumpfs FILE\_SYSTEM\_NAME****説明**

このオプションを使用すると、`curses` 表示を作成せずに、指定されたファイルシステムのメトリクスを標準出力にダンプできます。

## 5.2. MDS AUTOSCALER モジュールの使用

MDS Autoscaler モジュールは、Ceph File System (CephFS) を監視し、十分な MDS デーモンが利用可能であることを確認します。MDS サービスの Orchestrator バックエンドの配置仕様を調整することで機能します。

モジュールは、以下のファイルシステム設定をモニタリングして、配置数の調整を通知します。

- **max\_mds** ファイルシステムの設定
- **standby\_count\_wanted** ファイルシステムの設定

Ceph monitor デーモンは、これらの設定に応じて MDS をプロモートまたは停止します。**mds\_autoscaler** は、オーケストレーターによって起動する MDS の数を調整します。

**前提条件**

- 正常かつ稼働中の Red Hat Ceph Storage クラスタ
- Ceph File System のデプロイメント
- Ceph Monitor ノードへの root レベルのアクセス。

**手順**

- MDS Autoscaler モジュールを有効にします。

**例**

```
[ceph: root@host01 /]# ceph mgr module enable mds_autoscaler
```

### 5.3. カーネルクライアントとしてマウントされた CEPH FILE SYSTEMS のアンマウント

カーネルクライアントとしてマウントされている Ceph File System をアンマウントする方法。

#### 前提条件

- マウントを実行するノードへの Root レベルのアクセス。

#### 手順

- カーネルクライアントとしてマウントされている Ceph File System をアンマウントするには、以下を実行します。

#### 構文

```
umount MOUNT_POINT
```

#### 例

```
[root@client ~]# umount /mnt/cephfs
```

#### 関連情報

- **umount(8)** man ページ

### 5.4. FUSE クライアントとしてマウントされている CEPH FILE SYSTEMS のアンマウント

File System in User Space (FUSE) クライアントとしてマウントされている Ceph File System のアンマウント。

#### 前提条件

- FUSE クライアントノードへのルートレベルのアクセス。

#### 手順

- FUSE にマウントされた Ceph File System をアンマウントするには、以下を実行します。

#### 構文

```
fusermount -u MOUNT_POINT
```

#### 例

```
[root@client ~]# fusermount -u /mnt/cephfs
```

#### 関連情報

- **ceph-fuse(8)** man ページ

## 5.5. ディレクトリーツリーから METADATA SERVER デーモンのランクへのマッピング

ディレクトリーとそのサブディレクトリーを特定のアクティブな Metadata Server (MDS) ランクにマッピングするには、そのメタデータがランクを保持する MDS デーモンによってのみ管理されるようにします。このアプローチにより、アプリケーションの負荷や、ユーザーのメタデータ要求の影響をストレージクラスター全体に均等に分散させることができます。



### 重要

内部バランサーは、すでにアプリケーションの負荷を動的に分散します。そのため、特定の慎重に選択したアプリケーションに対して、ディレクトリーツリーのみをマップします。

さらに、ディレクトリーがランクにマップされると、バランサーはこれを分割できません。そのため、マップされたディレクトリー内の多数の操作を行うと、ランクおよびそれを管理する MDS デーモンをオーバーロードできます。

### 前提条件

- 少なくとも 2 つのアクティブな MDS デーモン。
- CephFS クライアントノードへのユーザーアクセス
- マウントされた Ceph File System を含む CephFS クライアントノードに **attr** パッケージがインストールされていることを確認します。

### 手順

1. Ceph ユーザーのケイパビリティーに **p** フラグを追加します。

#### 構文

```
ceph fs authorize FILE_SYSTEM_NAME client.CLIENT_NAME /DIRECTORY CAPABILITY
[/DIRECTORY CAPABILITY] ...
```

#### 例

```
[user@client ~]$ ceph fs authorize cephfs_a client.1 /temp rwp
client.1
key: AQBSdFhcGZFUDRAAcKhG9CI2HPiDMMRv4DC43A==
caps: [mds] allow r, allow rwp path=/temp
caps: [mon] allow r
caps: [osd] allow rw tag cephfs data=cephfs_a
```

2. ディレクトリーに **ceph.dir.pin** 拡張属性を設定します。

#### 構文

```
setfattr -n ceph.dir.pin -v RANK DIRECTORY
```

#### 例

```
[user@client ~]$ setfaattr -n ceph.dir.pin -v 2 /temp
```

この例では、**/temp** ディレクトリーとそのすべてのサブディレクトリーを rank 2 に割り当てます。

## 関連情報

- **p** フラグの詳細は、Red Hat Ceph Storage ファイルシステムガイドの [レイアウト、クォータ、スナップショット、ネットワークの制限](#) についてのセクションを参照してください。
- 詳細は、Red Hat Ceph Storage File System ガイドの [ディレクトリーツリーを特定のランクに手動で固定する](#) セクションを参照してください。
- 詳細は、Red Hat Ceph Storage ファイルシステムガイドの [複数のアクティブな Metadata Server デーモンの設定](#) セクションを参照してください。

## 5.6. METADATA SERVER デーモンのランクからディレクトリーツリーの解除

特定のアクティブなメタデータサーバー (MDS) ランクからディレクトリーの関連付けを解除します。

### 前提条件

- Ceph File System (CephFS) クライアントノードへのユーザーアクセス
- マウントされた CephFS を持つクライアントノードに **attr** パッケージがインストールされていることを確認します。

### 手順

- ディレクトリーの **ceph.dir.pin** 拡張属性を **-1** に設定します。

### 構文

```
setfaattr -n ceph.dir.pin -v -1 DIRECTORY
```

### 例

```
[user@client ~]$ setfaattr -n ceph.dir.pin -v -1 /home/ceph-user
```



### 注記

**/home/ceph-user/** の個別にマッピングされたサブディレクトリーは影響を受けません。

## 関連情報

- 詳細は、Red Hat Ceph Storage File System ガイドの [ディレクトリーツリーから Metadata Server デーモンのランクへのマッピング](#) セクションを参照してください。

## 5.7. データプールの追加

Ceph File System (CephFS) では、データの保存に使用する複数のプールの追加をサポートします。これは以下に役立ちます。

- ログデータの冗長性プールの削減。
- SSD または NVMe プールへのユーザーのホームディレクトリーの保存。
- 基本的なデータ分離。

Ceph File System で別のデータプールを使用する前に、本セクションで説明されているように追加する必要があります。

デフォルトでは、ファイルデータを保存するために、CephFS は作成中に指定された初期データプールを使用します。セカンダリーデータプールを使用するには、ファイルシステム階層の一部を設定して、そのプールにファイルデータを保存するか、必要に応じてそのプールの名前空間内にファイルデータを保存し、ファイルおよびディレクトリーのレイアウトを使用します。

### 前提条件

- Ceph Monitor ノードへのルートレベルのアクセス。

### 手順

1. 新しいデータプールを作成します。

#### 構文

```
ceph osd pool create POOL_NAME
```

以下を置き換えます。

- **POOL\_NAME** は、プールの名前に置き換えます。

#### 例

```
[ceph: root@host01 /]# ceph osd pool create cephfs_data_ssd  
pool 'cephfs_data_ssd' created
```

2. メタデータサーバーの制御の下に、新たに作成されたプールを追加します。

#### 構文

```
ceph fs add_data_pool FS_NAME POOL_NAME
```

以下を置き換えます。

- **FS\_NAME** は、ファイルシステムの名前に置き換えます。
- **POOL\_NAME** は、プールの名前に置き換えます。

たとえば、以下のようになります。

```
[ceph: root@host01 /]# ceph fs add_data_pool cephfs cephfs_data_ssd
added data pool 6 to fsmap
```

3. プールが正常に追加されたことを確認します。

#### 例

```
[ceph: root@host01 /]# ceph fs ls
name: cephfs, metadata pool: cephfs_metadata, data pools: [cephfs_data cephfs_data_ssd]
```

4. オプション: ファイルシステムからデータプールを削除します。

#### 構文

```
ceph fs rm_data_pool FS_NAME POOL_NAME
```

たとえば、以下のようになります。

```
[ceph: root@host01 /]# ceph fs rm_data_pool cephfs cephfs_data_ssd
removed data pool 6 from fsmap
```

- a. プールが正常に削除されたことを確認します。

#### 例

```
[ceph: root@host01 /]# ceph fs ls
name: cephfs, metadata pool: cephfs_metadata, data pools: [cephfs.cephfs.data]
```

5. **cephx** 認証を使用する場合は、クライアントが新しいプールにアクセスできることを確認してください。

#### 関連情報

- 詳細は、Red Hat Ceph Storage File System ガイドの [ファイルおよびディレクトリーレイアウト](#) セクションを参照してください。
- 詳細は、Red Hat Ceph Storage File System ガイドの [Ceph File System のクライアントユーザーの作成](#) セクションを参照してください。

## 5.8. CEPH FILE SYSTEM クラスターの停止

**down** フラグを **true** に設定すると、Ceph File System (CephFS) クラスターを停止することができます。これにより、ジャーナルをメタデータプールにフラッシュして Metadata Server (MDS) デーモンを正常にシャットダウンし、すべてのクライアント I/O が停止します。

また、CephFS クラスターを迅速に停止してファイルシステムの削除をテストし、メタデータサーバー (MDS) デーモン (障害復旧シナリオなど) を停止することもできます。これにより、MDS のスタンバイデーモンがファイルシステムをアクティベートしないように **jointable** フラグが設定されます。

## 前提条件

- Ceph Monitor ノードへの root レベルのアクセス。

## 手順

1. CephFS クラスタが停止しているには、以下を実行します。

### 構文

```
ceph fs set FS_NAME down true
```

### 例

```
[ceph: root@host01 /]# ceph fs set cephfs down true
```

- a. CephFS クラスタを起動するには、以下をバックアップします。

### 構文

```
ceph fs set FS_NAME down false
```

### 例

```
[ceph: root@host01 /]# ceph fs set cephfs down false
```

## または

1. CephFS クラスタを迅速に停止するには、以下を実行します。

### 構文

```
ceph fs fail FS_NAME
```

### 例

```
[ceph: root@host01 /]# ceph fs fail cephfs
```



### 注記

CephFS クラスターのバックアップを作成するには、**cephfs** を **joinable** に設定します。

### 構文

```
ceph fs set FS_NAME joinable true
```

### 例

```
[ceph: root@host01 /]# ceph fs set cephfs joinable true
cephfs marked joinable; MDS may join as newly active.
```

## 5.9. CEPH ファイルシステムの削除

Ceph File System (CephFS) を削除できます。その前に、すべてのデータのバックアップを作成し、すべてのクライアントがローカルにファイルシステムのマウントを解除していることを確認します。



### 警告

この操作は破壊的で、Ceph File System に保存されているデータが永続的にアクセスできないようにします。

### 前提条件

- データのバックアップを作成します。
- Ceph Monitor ノードへの root レベルのアクセス。

### 手順

1. ストレージクラスターに down のマークを付けます。

### 構文

```
ceph fs set FS_NAME down true
```

### 置き換え

- **FS\_NAME** は、削除する Ceph File System の名前に置き換えます。

### 例

```
[ceph: root@host01 /]# ceph fs set cephfs down true
cephfs marked down.
```



## 2. Ceph File System のステータス表示

```
ceph fs status
```

### 例

```
[ceph: root@host01 /]# ceph fs status

cephfs - 0 clients
=====
+-----+-----+-----+-----+
|  POOL   | TYPE | USED | AVAIL |
+-----+-----+-----+-----+
|cephfs.cephfs.meta | metadata | 31.5M | 52.6G|
|cephfs.cephfs.data | data    | 0    | 52.6G|
+-----+-----+-----+-----+
                STANDBY MDS
cephfs.ceph-host01
cephfs.ceph-host02
cephfs.ceph-host03
```

## 3. Ceph ファイルシステムを削除します。

### 構文

```
ceph fs rm FS_NAME --yes-i-really-mean-it
```

### 置き換え

- **FS\_NAME** は、削除する Ceph File System の名前に置き換えます。

### 例

```
[ceph: root@host01 /]# ceph fs rm cephfs --yes-i-really-mean-it
```

## 4. ファイルシステムが正常に削除されたことを確認します。

### 例

```
[ceph: root@host01 /]# ceph fs ls
```

## 5. オプションです。削除されたファイルシステムに関連付けられたデータおよびメタデータプールを削除します。

### 関連情報

- Red Hat Ceph Storage ストレージストラテジーガイドの [プールの削除](#) セクションを参照してください。

## 5.10. CEPH MDS FAIL コマンドの使用

`ceph mds fail` コマンドを使用して、以下を実行します。

- MDS デーモンを失敗としてマークします。デーモンがアクティブで適切なスタンバイデーモンが利用可能な場合で、**standby-replay** 設定を無効にした後にスタンバイデーモンがアクティブな場合には、このコマンドを使用すると standby デーモンへのフェイルオーバーを強制します。**standby-replay** デーモンを無効にすることで、新規の **standby-replay** デーモンが割り当てられないようにします。
- 実行中の MDS デーモンを再起動します。デーモンがアクティブで、適切なスタンバイデーモンが利用できる場合には、failed デーモンはスタンバイデーモンになります。

#### 前提条件

- Ceph MDS デーモンのインストールおよび設定

#### 手順

- デーモンが失敗するには、以下を実行します。

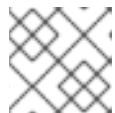
#### 構文

```
ceph mds fail MDS_NAME
```

**MDS\_NAME** は、MDS ノード **standby-replay** の名前です。

#### 例

```
[ceph: root@host01 /]# ceph mds fail example01
```



#### 注記

Ceph MDS 名は、**ceph fs status** コマンドで確認することができます。

#### 関連情報

- Red Hat Ceph Storage File System ガイドの [アクティブな Metadata Server デーモンの数を減らす](#) セクションを参照してください。
- Red Hat Ceph Storage File System ガイドの [スタンバイデーモンの数の設定](#) セクションを参照してください。
- Red Hat Ceph Storage File System ガイドの [Metadata Server ランク](#) セクションを参照してください。

## 5.11. クライアント機能

Ceph File System (CephFS) 機能を設定する際には、クライアントが Ceph File Systems を使用できるようにサポートしておく必要がある場合があります。これらの機能がないクライアントでは、他の CephFS クライアントを中断したり、予期せぬ動作をすることがあります。また、古いものや、クライアントが Ceph File System に接続しないように、新機能が必要になる場合があります。



## 重要

新たに追加された機能がない CephFS クライアントは自動的にエビクトされます。

**fs features ls** コマンドを使用して、すべての CephFS 機能をリスト表示できます。**fs required\_client\_features** コマンドを使用して要件を追加または削除できます。

## 構文

```
fs required_client_features FILE_SYSTEM_NAME add FEATURE_NAME
fs required_client_features FILE_SYSTEM_NAME rm FEATURE_NAME
```

## 機能の説明

### reply\_encoding

#### 説明

Ceph Metadata Server(MDS) は、クライアントがこの機能をサポートする場合は、拡張可能な形式で応答要求をエンコードします。

### reclaim\_client

#### 説明

Ceph MDS により、新しいクライアントがデッド、クライアントの状態が停止するなど、別のクライアントを回収できます。この機能は、NFS Ganesha により使用されます。

### lazy\_caps\_wanted

#### 説明

古いクライアントが再開すると、Ceph MDS は、クライアントがこの機能をサポートしている場合に限り、明示的に必要な機能を再発行する必要があります。

### multi\_reconnect

#### 説明

Ceph MDS フェイルオーバーのイベントの後に、クライアントは再接続メッセージを MDS に送信してキャッシュ状態を再確立します。クライアントは、大きな再接続メッセージを複数のメッセージに分割できます。

### deleg\_ino

#### 説明

Ceph MDS は、クライアントがこの機能に対応していると、inode 数をクライアントに委譲します。inode 番号を委譲することは、クライアントが非同期ファイルの作成を行うための前提条件です。

### metric\_collect

#### 説明

CephFS クライアントは、パフォーマンスメトリックを Ceph MDS に送信することができます。

### alternate\_name

#### 説明

CephFS クライアントは、ディレクトリーエントリーの代替名を設定して理解することができます。この機能により、暗号化されたファイル名を使用できます。

## 5.12. CEPH FILE SYSTEM クライアントのエビクション

Ceph File System (CephFS) クライアントが応答しない、または不正な動作をする場合、強制的に終了したり、CephFS にアクセスしないようにエビクトする必要がある場合があります。CephFS クライアントをエビクトすると、さらにメタデータサーバー (MDS) デーモンおよび Ceph OSD デーモンと通信できなくなります。CephFS クライアントがエビクション時に CephFS に I/O をバッファーする場合、フラッシュされていないデータはすべて失われます。CephFS クライアントのエビクションプロセスは、すべてのクライアントタイプに適用されます。FUSE マウント、カーネルマウント、NFS ゲートウェイ、および `libcephfs` API ライブラリーを使用するプロセス。

CephFS クライアントを自動的にエビクトできます。これにより、MDS デーモンと一時的に通信できない場合、または手動での通信を行うことができます。

### 自動クライアントエビクション

以下のシナリオにより、CephFS クライアントの自動エビクションが発生します。

- CephFS クライアントがデフォルトの 300 秒にわたってアクティブな MDS デーモンと通信していない場合、または `session_autoclose` オプションで設定した場合。
- `mds_cap_revoke_eviction_timeout` オプションが設定されている場合、CephFS クライアントは設定される期間 (秒単位) の制限メッセージに対応しない。 `mds_cap_revoke_eviction_timeout` オプションはデフォルトで無効にされています。
- MDS の起動またはフェイルオーバー時に、MDS デーモンは、すべての CephFS クライアントが新しい MDS デーモンに接続するのを待機する再接続フェーズを通過します。CephFS クライアントがデフォルトの時間枠内に 45 秒以内に再接続できない場合、または `mds_reconnect_timeout` オプションで設定した場合。

### 関連情報

- 詳細は、Red Hat Ceph Storage ファイルシステムガイドの [Ceph File System クライアントの手動エビクト](#) セクションを参照してください。

## 5.13. ブロックリスト CEPH FILE SYSTEM クライアント

Ceph File System (CephFS) クライアントのブロック機能は、デフォルトで有効になっています。エビクションコマンドを単一の Metadata Server (MDS) デーモンに送信すると、拒否リストが他の MDS デーモンに伝播されます。これは、CephFS クライアントがデータオブジェクトにアクセスできないようにするため、他の CephFS クライアントを更新し、拒否されたクライアントエントリーを含む最新の Ceph OSD マップと共に MDS デーモンを更新する必要があります。

Ceph OSD マップの更新時に内部の `osdmap epoch barrier` メカニズムが使用されます。バリアは、ENOSPC やエビクションからのブロックリスト化されたクライアントなど、同じ RADOS オブジェクトへのアクセスが許可される可能性のある機能が割り当てられている前に、CephFS クライアントが機能を受信するために十分な最近の Ceph OSD マップがあることを検証することです。

低速なノードまたは信頼できないネットワークが原因で CephFS クライアントのエビクションが頻繁に行われていて、根本的な問題を修正できない場合、MDS により厳格に見なっているよう依頼できません。MDS セッションを単純にドロップすることで、遅い CephFS クライアントに応答することができます。

ますが、CephFS クライアントがセッションを再度開いたことを許可し、Ceph OSD との通信を継続できます。`mds_session_blocklist_on_timeout` および `mds_session_blocklist_on_evict` オプションを `false` に設定すると、このモードが有効になります。



### 注記

ブロックリストが無効になっている場合、エビクトされた CephFS クライアントはコマンドの送信先となる MDS デーモンにのみ影響を与えます。複数のアクティブな MDS デーモンがあるシステムでは、エビクションコマンドを各アクティブなデーモンに送信する必要があります。

## 5.14. CEPH FILE SYSTEM クライアントの手動エビクト

Ceph File System (CephFS) クライアントを手動でエビクトして、クライアントの誤作動やクライアントノードへのアクセスがない場合や、クライアントの動作がタイムアウトするのを待たずに、クライアントセッションがタイムアウトするのを待たずに、Ceph File System (CephFS) クライアントを手動でエビクトしたい場合があります。

### 前提条件

- Ceph Monitor ノードへのルートレベルのアクセス。

### 手順

1. クライアントリストを確認します。

### 構文

```
ceph tell DAEMON_NAME client ls
```

### 例

```
[ceph: root@host01 /]# ceph tell mds.0 client ls
[
  {
    "id": 4305,
    "num_leases": 0,
    "num_caps": 3,
    "state": "open",
    "replay_requests": 0,
    "completed_requests": 0,
    "reconnecting": false,
    "inst": "client.4305 172.21.9.34:0/422650892",
    "client_metadata": {
      "ceph_sha1": "79f0367338897c8c6d9805eb8c9ad24af0dcd9c7",
      "ceph_version": "ceph version 16.2.8-65.el8cp
(79f0367338897c8c6d9805eb8c9ad24af0dcd9c7)",
      "entity_id": "0",
      "hostname": "senta04",
      "mount_point": "/tmp/tmpcMpF1b/mnt.0",
      "pid": "29377",
      "root": "/"
    }
  }
]
```

```

    }
  }
]

```

- 指定した CephFS クライアントをエビクトします。

### 構文

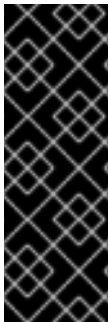
```
ceph tell DAEMON_NAME client evict id=ID_NUMBER
```

### 例

```
[ceph: root@host01 /]# ceph tell mds.0 client evict id=4305
```

## 5.15. ブロックリストからの CEPH FILE SYSTEM クライアントの削除

場合によっては、以前のブロックリストされた Ceph File System (CephFS) クライアントがストレージクラスターに再接続できるようにすることが役に立つ場合があります。



### 重要

blocklist から CephFS クライアントを削除すると、データの整合性はリスクに伴います。また、その結果、CephFS クライアントが完全に確実に機能することを保証することはありません。エビクション後に完全に正常な CephFS クライアントを取得するには、CephFS クライアントをアンマウントして新規マウントを行うのが最適です。その他の CephFS クライアントが、拒否リストされた CephFS クライアントによってバッファされた I/O を実行してデータの破損につながるファイルにアクセスしている場合は、データが破損する可能性があります。

### 前提条件

- Ceph Monitor ノードへのルートレベルのアクセス。

### 手順

- ブロックリストを確認します。

### 例

```
[ceph: root@host01 /]# ceph osd blocklist ls

listed 1 entries
127.0.0.1:0/3710147553 2022-05-09 11:32:24.716146
```

- ブロックリストから CephFS クライアントを削除します。

### 構文

```
ceph osd blocklist rm CLIENT_NAME_OR_IP_ADDR
```

### 例

```
[ceph: root@host01 /]# ceph osd blocklist rm 127.0.0.1:0/3710147553
```

```
un-blocklisting 127.0.0.1:0/3710147553
```

3. オプションで、ブロックリストからカーネルベースの CephFS クライアントを自動的に再接続できます。カーネルベースの CephFS クライアントで、手動マウントを行う場合は **clean** に以下のオプションを設定するか、**/etc/fstab** ファイルのエントリーで自動的にマウントします。

```
recover_session=clean
```

4. 任意で、ブロックリストから削除すると、FUSE ベースの CephFS クライアントを自動的に再接続できます。FUSE クライアントで、手動マウントを行う場合は以下のオプションを **true** に設定するか、**/etc/fstab** ファイルのエントリーで自動的にマウントします。

```
client_reconnect_stale=true
```

## 関連情報

- 詳細は、Red Hat Ceph Storage File System ガイドの [Ceph File System を FUSE クライアントとしてマウント](#) セクションを参照してください。

## 第6章 NFS クラスターとエクスポート管理

ストレージ管理者は、NFS クラスターを作成してカスタマイズし、NFS プロトコル経由で Ceph File System 名前空間をエクスポートできます。

### 前提条件

- 実行中、および正常な Red Hat Ceph Storage クラスター
- Ceph Metadata Server デーモン (**ceph-mds**) のインストールおよび設定
- Ceph ファイルシステムを作成してマウントします。

### 6.1. NFS クラスターの作成

**nfs cluster create** コマンドを使用して NFS クラスターを作成します。これにより、すべての NFS Ganesha デーモン、クラスター名に基づく新しいユーザー、および共通の NFS Ganesha 設定 RADOS オブジェクトに共通の回復プールが作成されます。

### 前提条件

- 実行中、および正常な Red Hat Ceph Storage クラスター
- 既存の Ceph File System。
- Ceph Monitor への root レベルのアクセス。
- Ceph Manager ホスト上の **nfs-ganesha**、**nfs-ganesha-ceph**、**nfs-ganesha-grace** および **nfs-ganesha-rados-urls** パッケージをインストールします。
- クライアントへの root レベルのアクセス。

### 手順

1. Cephadm シェルにログインします。

#### 例

```
[root@mds ~]# cephadm shell
```

2. Ceph Manager の NFS モジュールを有効にします。

#### 例

```
[ceph: root@host01 /]# ceph mgr module enable nfs
```

3. NFS Ganesha クラスターを作成します。

#### 構文

```
ceph nfs cluster create CLUSTER_NAME [PLACEMENT] [--ingress] [--virtual_ip  
IP_ADDRESS] [--ingress-mode {default|keepalive-only|haproxy-standard|haproxy-protocol}]  
[--port PORT]
```



**例**

```
[ceph: root@host01 /]# ceph nfs cluster create nfs-cephfs "host01 host02"
NFS Cluster Created Successfully
```

この例では、NFS Ganesha クラスタ名は **nfs-cephfs** で、デーモンコンテナーは **host01** および **host02** にデプロイされます。

**重要**

Red Hat は、1つのホストにつき1つの NFS Ganesha デーモンのみをサポートします。

4. NFS-Ganesha クラスタ情報を検証します。

**構文**

```
ceph nfs cluster info [CLUSTER_NAME]
```

**例**

```
[ceph: root@host01 /]# ceph nfs cluster info nfs-cephfs
{
  "nfs-cephfs": [
    {
      "hostname": "host01",
      "ip": "10.74.179.124",
      "port": 2049
    },
    {
      "hostname": "host02",
      "ip": "10.74.180.160",
      "port": 2049
    }
  ]
}
```

**注記**

**CLUSTER\_NAME** の指定は任意です。

## 6.2. NFS 設定のカスタマイズ

設定ファイルを使用して NFS クラスタをカスタマイズします。これにより、NFS クラスタは指定された設定を使用し、デフォルトの設定ブロックよりも優先されます。

**前提条件**

- 実行中、および正常な Red Hat Ceph Storage クラスタ
- Ceph Metadata Server (MDS) ノードへのルートレベルのアクセス。

- **ceph nfs cluster create** コマンドを使用して作成された NFS クラスタ。

## 手順

1. 設定ファイルを作成します。

### 例

```
[ceph: root@host01 /]# touch nfs-cephfs.conf
```

2. 次のブロックを使用して、設定ファイルでのログ記録を有効にします。

### 例

```
[ceph: root@host01 /]# vi nfs-cephfs.conf
```

```
LOG {  
  COMPONENTS {  
    ALL = FULL_DEBUG;  
  }  
}
```

3. 新しい設定を行います。

### 構文

```
ceph nfs cluster config set CLUSTER_NAME -i PATH_TO_CONFIG_FILE
```

### 例

```
[ceph: root@host01 /]# ceph nfs cluster config set nfs-cephfs -i nfs-cephfs.conf
```

```
NFS-Ganesha Config Set Successfully
```

4. カスタマイズされた NFS Ganesha 設定を表示します。

### 構文

```
ceph nfs cluster config get CLUSTER_NAME
```

### 例

```
[ceph: root@host01 /]# ceph nfs cluster config get nfs-cephfs
```

```
LOG {  
  COMPONENTS {  
    ALL = FULL_DEBUG;  
  }  
}
```

これにより、ユーザー定義の設定の出力が提供されます (存在する場合)。

5. オプション: ユーザー定義の設定を削除する場合は、次のコマンドを実行します。

### 構文

```
ceph nfs cluster config reset CLUSTER_NAME
```

### 例

```
[ceph: root@host01 /]# ceph nfs cluster config reset nfs-cephfs
```

```
NFS-Ganesha Config Reset Successfully
```

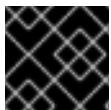
## 6.3. NFS プロトコルを介した CEPH FILE SYSTEM 名前空間のエクスポート (可用性に制限あり)

NFS Ganesha ファイルサーバーを使用して、Ceph File Systems (CephFS) 名前空間を NFS プロトコルでエクスポートすることができます。CephFS namespace をエクスポートするには、まず実行中の NFS Ganesha クラスターが必要です。



### 注記

この技術には、利用制限があります。詳細は、[Deprecated functionality](#) の章を参照してください。



### 重要

Red Hat は、NFS バージョン 4.0 以降のみをサポートします。



### 重要

NFS クライアントは、ネイティブ NFS マウント経由で CephFS スナップショットを作成できません。スナップショットの要件に合わせて、サーバー側のオペレーターツールを使用する必要があります。

### 前提条件

- 実行中、および正常な Red Hat Ceph Storage クラスター
- **ceph nfs cluster create** コマンドを使用して作成された NFS クラスター。

### 手順

1. CephFS エクスポートを作成します。

### 構文

```
ceph nfs export create cephfs CLUSTER_NAME BINDING FILE_SYSTEM_NAME [--readonly] [--path=PATH_WITHIN_CEPHFS]
```

### 例

```
[ceph: root@host01 /]# ceph nfs export create cephfs nfs-cephfs /ceph cephfs01 --path=
```

この例では、**BINDING (/ceph)** は pseudo root パスであり、これは一意のパスと絶対パスになります。



### 注記

**--readonly** オプションは、読み取り専用のパーミッションでパスをエクスポートし、デフォルトは read、および write のパーミッションになります。



### 注記

**PATH\_WITHIN\_CEPHFS** はサブボリュームになります。以下のコマンドを使用して、絶対サブボリュームパスを取得できます。

### 構文

```
ceph fs subvolume getpath VOLUME_NAME SUBVOLUME_NAME [--group_name SUBVOLUME_GROUP_NAME]
```

### 例

```
[ceph: root@host01 /]# ceph fs subvolume getpath cephfs sub0
```

- 疑似ルート名に基づいてエクスポートブロックを表示します。

### 構文

```
ceph nfs export get CLUSTER_NAME BINDING
```

### 例

```
[ceph: root@host01 /]# ceph nfs export get nfs-cephfs /ceph
{
  "export_id": 1,
  "path": "/",
  "cluster_id": "nfs-cephfs",
  "pseudo": "/ceph",
  "access_type": "RW",
  "squash": "no_root_squash",
  "security_label": true,
  "protocols": [
    4
  ],
  "transports": [
    "TCP"
  ],
  "fsal": {
    "name": "CEPH",
    "user_id": "cephnfs11",
    "fs_name": "cephfs",
    "sec_label_xattr": ""
  }
}
```

```

    },
    "clients": []
  }
}

```

3. NFS エクスポートをリスト表示します。

### 構文

```
ceph nfs export ls CLUSTER_NAME [--detailed]
```

### 例

```

[ceph: root@host01 /]# ceph nfs export ls nfs-cephfs
[
  "/ceph/"
]
[ceph: root@host01 /]# ceph nfs export ls nfs-cephfs --detailed
[
  {
    "export_id": 100,
    "path": "/",
    "cluster_id": "nfs-cephfs",
    "pseudo": "/ceph/",
    "access_type": "RW",
    "squash": "no_root_squash",
    "security_label": true,
    "protocols": [
      4
    ],
    "transports": [
      "TCP"
    ],
    "fsal": {
      "name": "CEPH",
      "user_id": "nfstest01",
      "fs_name": "cephfs",
      "sec_label_xattr": ""
    },
    "clients": []
  }
]

```

4. NFS エクスポートの情報を取得します。

### 構文

```
ceph nfs export info CLUSTER_NAME [PSEUDO_PATH]
```

### 例

```

[ceph: root@host01 /]# ceph nfs export info nfs-cephfs /ceph
{
  "export_id": 1,

```

```

"path": "/",
"cluster_id": "nfs-cephfs",
"pseudo": "/ceph",
"access_type": "RW",
"squash": "none",
"security_label": true,
"protocols": [
  4
],
"transports": [
  "TCP"
],
"fsal": {
  "name": "CEPH",
  "user_id": "nfs.nfs-cephfs.1",
  "fs_name": "cephfs"
},
"clients": []
}

```

- クライアントホストで、エクスポートされた Ceph File System をマウントします。

#### 構文

```
mount -t nfs -o port=GANESHA_PORT HOST_NAME:BINDING LOCAL_MOUNT_POINT
```

#### 例

```
[root@client01 ~]# mount -t nfs -o port=2049 host01:/ceph/ /mnt/nfs-cephfs
```

- システムの起動時に自動的にマウントするには、改行を追加して **/etc/fstab** ファイルを開いて編集します。

#### 構文

```
HOST_NAME:BINDING LOCAL_MOUNT_POINT nfs4
defaults,seclabel,vers=4.2,proto=tcp,port=2049 0 0
```

#### 例

```
host01:/ceph/ /mnt/nfs-cephfs nfs4 defaults,seclabel,vers=4.2,proto=tcp,port=2049 0
0
```

- クライアントホストで、**Ingress** サービスで作成されたエクスポートされた NFS Ceph File System をマウントするには、以下を実行します。

#### 構文

```
mount -t nfs VIRTUAL_IP_ADDRESS:BINDING LOCAL_MOUNT_POINT
```

- VIRTUAL\_IP\_ADDRESS** は、NFS クラスターの作成に使用される **--ingress --virtual-ip** IP アドレスに置き換えます。

- **BINDING** は、擬似ルートパスに置き換えます。
- **LOCAL\_MOUNT\_POINT** は、エクスポートをマウントするマウントポイントに置き換えます。

### 例

```
[root@client01 ~]# mount -t nfs 10.10.128.75:/nfs-cephfs /mnt
```

この例では、マウントポイント `/mnt` に `--ingress --virtual-ip 10.10.128.75` で作成した NFS クラスタに存在するエクスポート `nfs-cephfs` をマウントします。

## 6.4. CEPH ファイルシステムのエクスポートの変更

設定ファイルを使用してエクスポート内の次のパラメーターを変更できます。

- **access\_type**: **RW**、**RO**、または **NONE** のいずれかになります。
- **squash**: **No\_Root\_Squash**、**None**、または **Root\_Squash** のいずれかになります。
- **security\_label**: これは **true** または **false** です。

### 前提条件

- 実行中、および正常な Red Hat Ceph Storage クラスタ
- NFS エクスポートが作成されました。

### 手順

1. 疑似ルート名に基づいてエクスポートブロックを表示します。

### 構文

```
ceph nfs export get CLUSTER_NAME BINDING
```

### 例

```
[ceph: root@host01 /]# ceph nfs export get nfs-cephfs /ceph
{
  "export_id": 1,
  "path": "/",
  "cluster_id": "nfs-cephfs",
  "pseudo": "/ceph",
  "access_type": "RO",
  "squash": "none",
  "security_label": true,
  "protocols": [
    4
  ],
  "transports": [
    "TCP"
  ],
  "fsal": {
```

```

    "name": "CEPH",
    "user_id": "cephnfs11",
    "fs_name": "cephfs",
    "sec_label_xattr": ""
  },
  "clients": []
}

```

2. 設定ファイルをエクスポートします。

### 例

```
[ceph: root@host01 /]# ceph nfs export get nfs-cephfs /ceph > export.conf
```

3. エクスポート情報を編集します。

### 構文

```

{
  "export_id": EXPORT_ID,
  "path": "/",
  "cluster_id": "CLUSTER_NAME",
  "pseudo": "CLUSTER_PSEUDO_PATH",
  "access_type": "RW/RO",
  "squash": "SQUASH",
  "security_label": SECURITY_LABEL,
  "protocols": [
    PROTOCOL_ID_
  ],
  "transports": [
    "TCP"
  ],
  "fsal": {
    "name": "NAME",
    "user_id": "USER_ID",
    "fs_name": "FILE_SYSTEM_NAME",
    "sec_label_xattr": ""
  },
  "clients": []
}

```

### 例

```
[ceph: root@host01 /]# vi export.conf
```

```

{
  "export_id": 1,
  "path": "/",
  "cluster_id": "nfs-cephfs",
  "pseudo": "/ceph",
  "access_type": "RW",
  "squash": "none",
  "security_label": true,
  "protocols": [

```



```

    4
  ],
  "transports": [
    "TCP"
  ],
  "fsal": {
    "name": "CEPH",
    "user_id": "cephnfs11",
    "fs_name": "cephfs",
    "sec_label_xattr": ""
  },
  "clients": []
}

```

上記の例では、**access\_type** が **RO** から **RW** に変更されます。

- 仕様を適用します。

#### 構文

```
ceph nfs export apply CLUSTER_NAME PATH_TO_EXPORT_FILE
```

#### 例

```
[ceph: root@host01 /]# ceph nfs export apply nfs-cephfs -i export.conf
Added export /ceph
```

- 更新されたエクスポート情報を取得します。

#### 構文

```
ceph nfs export get CLUSTER_NAME BINDING
```

#### 例

```
[ceph: root@host01 /]# ceph nfs export get nfs-cephfs /ceph
{
  "export_id": 1,
  "path": "/",
  "cluster_id": "nfs-cephfs",
  "pseudo": "/ceph",
  "access_type": "RW",
  "squash": "none",
  "security_label": true,
  "protocols": [
    4
  ],
  "transports": [
    "TCP"
  ],
  "fsal": {
    "name": "CEPH",
    "user_id": "cephnfs11",
    "fs_name": "cephfs",

```

```
"sec_label_xattr": ""
},
"clients": []
}
```

## 6.5. カスタム CEPH ファイルシステムエクスポートの作成

Ceph File System (CepFS) エクスポートをカスタマイズし、設定を適用できます。

### 前提条件

- 実行中、および正常な Red Hat Ceph Storage クラスタ
- **ceph nfs cluster create** コマンドを使用して作成された NFS クラスタ。
- CephFS が作成されている。

### 手順

1. カスタムファイルを作成します。

#### 例

```
[ceph: root@host01 /]# touch export_new.conf
```

2. カスタムファイルを使用してエクスポートを作成します。

#### 構文

```
EXPORT {
  Export_Id = EXPORT_ID;
  Transports = TCP/UDP;
  Path = PATH;
  Pseudo = PSEUDO_PATH;
  Protocols = NFS_PROTOCOLS;
  Access_Type = ACCESS_TYPE;
  Attr_Expiration_Time = EXPIRATION_TIME;
  Squash = SQUASH;
  FSAL {
    Name = NAME;
    Filesystem = "CEPH_FILE_SYSTEM_NAME";
    User_Id = "USER_ID";
  }
}
```

#### 例

```
[ceph: root@host01 /]# cat export_new.conf
EXPORT {
  Export_Id = 2;
  Transports = TCP;
  Path = /;
  Pseudo = /ceph1/;
```

```

Protocols = 4;
Access_Type = RW;
Attr_Expiration_Time = 0;
Squash = None;
FSAL {
  Name = CEPH;
  Filesystem = "cephfs";
  User_Id = "nfs.nfs-cephfs.2";
}
}

```

- 仕様を適用します。

### 構文

```
ceph nfs export apply CLUSTER_NAME -i PATH_TO_EXPORT_FILE
```

### 例

```
[ceph: root@host01 /]# ceph nfs export apply nfs-cephfs -i new_export.conf
Added export /ceph1
```

- 更新されたエクスポート情報を取得します。

### 構文

```
ceph nfs export get CLUSTER_NAME BINDING
```

### 例

```
[ceph: root@host01 /]# ceph nfs export get nfs-cephfs /ceph1
{
  "export_id": 1,
  "path": "/",
  "cluster_id": "nfs-cephfs",
  "pseudo": "/ceph1",
  "access_type": "RW",
  "squash": "None",
  "security_label": true,
  "protocols": [
    4
  ],
  "transports": [
    "TCP"
  ],
  "fsal": {
    "name": "CEPH",
    "user_id": "nfs.nfs-cephfs.2",
    "fs_name": "cephfs",
    "sec_label_xattr": ""
  },
  "clients": []
}
```

## 6.6. CEPH ファイルシステムのエクスポートの削除

**Ceph Export rm** コマンドを使用して、Ceph File System (CephFS) NFS エクスポートを削除できます。

### 前提条件

- 実行中、および正常な Red Hat Ceph Storage クラスタ
- CephFS が作成されている。

### 手順

- CephFS エクスポートを削除します。

#### 構文

```
ceph nfs export rm CLUSTER_NAME BINDING
```

#### 例

```
[ceph: root@host01 /]# ceph nfs export rm nfs-cephfs /ceph
```

## 6.7. NFS クラスタの削除

**nfs cluster rm** コマンドを使用して、NFS クラスタを削除します。これにより、デプロイされたクラスタが削除されます。NFS デーモンと入力サービスの削除は非同期です。**ceph orch ls** コマンドで削除状況を確認します。

### 前提条件

- 実行中、および正常な Red Hat Ceph Storage クラスタ
- Ceph Metadata Server (MDS) ノードへのルートレベルのアクセス。
- **ceph nfs cluster create** コマンドを使用してデプロイされた NFS デーモン。

### 手順

1. Cephadm シェルにログインします。

#### 例

```
[root@mds ~]# cephadm shell
```

2. NFS Ganesha クラスタを削除します。

#### 構文

```
ceph nfs cluster rm CLUSTER_NAME
```

#### 例

```
[ceph: root@host01 /]# ceph nfs cluster rm nfs-cephfs
```

## 第7章 CEPH FILE システムのクォータ

ストレージ管理者は、ファイルシステム内の任意のディレクトリーでクォータを表示、設定、および削除できます。クォータの制限は、バイト数またはディレクトリー内のファイル数に配置できます。

### 前提条件

- 実行中、および正常な Red Hat Ceph Storage クラスター
- Ceph File System のデプロイメント
- **attr** パッケージがインストールされていることを確認します。

### 7.1. CEPH FILE システムのクォータ

Ceph File System (CephFS) のクォータにより、ディレクトリー構造に保存されたファイルの数またはバイト数を制限できます。Ceph File System のクォータは、FUSE クライアントまたはカーネルクライアント (バージョン 4.17 以降) を使用して完全にサポートされます。

#### 制限

- CephFS のクォータは、設定された制限に達するとデータの書き込みを停止するためにファイルシステムをマウントするクライアントとの協調に依存しています。ただし、クォータのみでは、信頼できないクライアントがファイルシステムを埋めないようにすることはできません。
- ファイルシステムにデータを書き込むプロセスが、設定された制限に到達したら、データ量がクォータ制限を超えるか、プロセスがデータの書き込みを停止するまでの短い期間が長くなります。通常、期間 (秒) は数十秒で測定されます。ただし、プロセスは、その期間中データの書き込みを続けます。プロセスが書き込む追加データ量は、停止前の経過時間によって異なります。
- パスベースのアクセス制限を使用する場合は、クライアントが制限されているディレクトリーのクォータを設定するか、その下でネスト化されたディレクトリーにクォータを設定してください。クライアントが MDS 機能に基づいて特定のパスへのアクセス制限があり、そのクォータがクライアントにアクセスできない上位ディレクトリーに設定されている場合、クライアントはクォータを強制しません。たとえば、クライアントが **/home/** ディレクトリーにアクセスできず、クォータが **/home/** で設定されている場合、クライアントは **/home/user/** ディレクトリーのクォータを強制できません。
- 削除または変更されたスナップショットファイルデータは、クォータに対してカウントされません。
- **setxattr** の使用時、NFS クライアントを使用したクォータのサポートはありません。NFS のファイルレベルのクォータはサポートされません。NFS 共有でクォータを使用するには、サブボリュームをエクスポートし、**--size** オプションを設定します。

### 7.2. クォータの表示

**getfattr** コマンドおよび **ceph.quota** 拡張属性を使用して、ディレクトリーのクォータ設定を表示します。



## 注記

属性が inode に表示されると、そのディレクトリーにクォータが設定されている必要があります。属性が inode に表示されない場合は、ディレクトリーにはクォータセットがありませんが、親ディレクトリーにはクォータが設定されている可能性があります。拡張属性の値が **0** の場合、クォータは設定されません。

## 前提条件

- Ceph クライアントノードへの root レベルのアクセス。
- **attr** パッケージがインストールされます。

## 手順

1. CephFS クォータを表示するには、以下を実行します。
  - a. バイト制限クォータの使用:

### 構文

```
getfattr -n ceph.quota.max_bytes DIRECTORY
```

### 例

```
[root@client ~]# getfattr -n ceph.quota.max_bytes /mnt/cephfs/
getfattr: Removing leading '/' from absolute path names
# file: mnt/cephfs/
ceph.quota.max_bytes="100000000"
```

この例では、**100000000** は 100 MB となります。

- b. ファイル制限クォータの使用:

### 構文

```
getfattr -n ceph.quota.max_files DIRECTORY
```

### 例

```
[root@client ~]# getfattr -n ceph.quota.max_files /mnt/cephfs/
getfattr: Removing leading '/' from absolute path names
# file: mnt/cephfs/
ceph.quota.max_files="10000"
```

この例では **10000** は 10,000 ファイルと等しくなります。

## 関連情報

- 詳細は、**getfattr(1)** man ページを参照してください。

## 7.3. クォータの設定

本セクションでは、**setfaattr** コマンドおよび **ceph.quota** 拡張属性を使用して、ディレクトリーのクォータを設定する方法を説明します。

### 前提条件

- Ceph クライアントノードへの root レベルのアクセス。
- **attr** パッケージがインストールされます。

### 手順

- バイト制限クォータを使用して、ディレクトリーのクォータを設定します。



#### 注記

バイト制限クォータとして値 K、Ki、M、Mi、G、Gi、T、および Ti がサポートされています。

### 構文

```
setfaattr -n ceph.quota.max_bytes -v LIMIT_VALUE DIRECTORY
```

### 例

```
[root@client ~]# setfaattr -n ceph.quota.max_bytes -v 2T /cephfs/
```

- ファイル制限クォータを使用して、ディレクトリーのクォータを設定します。

### 構文

```
setfaattr -n ceph.quota.max_files -v LIMIT_VALUE DIRECTORY
```

### 例

```
[root@client ~]# setfaattr -n ceph.quota.max_files -v 10000 /cephfs/
```

この例では **10000** は 10,000 ファイルと等しくなります。



#### 注記

ファイル **LIMIT\_VALUE** では数値のみがサポートされます。

### 関連情報

- 詳細は、**setfaattr(1)** man ページを参照してください。

## 7.4. クォータの削除

本セクションでは、**setfaattr** コマンドおよび **ceph.quota** 拡張属性を使用して、ディレクトリーからクォータを削除する方法を説明します。



## 別添木竹

- Ceph クライアントノードへの root レベルのアクセス。
- **attr** パッケージがインストールされていることを確認します。

## 手順

1. CephFS クォータを削除するには、以下のコマンドを実行します。

- a. バイト制限クォータの使用:

### 構文

```
setfattr -n ceph.quota.max_bytes -v 0 DIRECTORY
```

### 例

```
[root@client ~]# setfattr -n ceph.quota.max_bytes -v 0 /mnt/cephfs/
```

- b. ファイル制限クォータの使用:

### 構文

```
setfattr -n ceph.quota.max_files -v 0 DIRECTORY
```

### 例

```
[root@client ~]# setfattr -n ceph.quota.max_files -v 0 /mnt/cephfs/
```

## 関連情報

- 詳細は、**setfattr(1)** man ページを参照してください。

## 関連情報

- Red Hat Ceph Storage File System ガイドの [Ceph File System のデプロイメント](#) セクションを参照してください。
- 詳細は、**getfattr(1)** man ページを参照してください。
- 詳細は、**setfattr(1)** man ページを参照してください。

## 第8章 ファイルとディレクトリーのレイアウト

ストレージ管理者は、ファイルまたはディレクトリーのデータがオブジェクトにマップされる方法を制御できます。

本セクションでは、以下を行う方法を説明します。

- [ファイルおよびディレクトリーのレイアウトの理解](#)
- [ファイルおよびディレクトリーのレイアウトの設定](#)
- [ファイルとディレクトリーのレイアウトフィールドの表示](#)
- [個別のレイアウトフィールドの表示](#)
- [ディレクトリーレイアウト削除](#)

### 前提条件

- 実行中、および正常な Red Hat Ceph Storage クラスター
- Ceph File System のデプロイメント
- `attr` パッケージのインストール

### 8.1. ファイルとディレクトリーレイアウトの概要

本セクションでは、Ceph File System のコンテキストにおけるファイルおよびディレクトリーのレイアウトを説明します。

ファイルまたはディレクトリーのレイアウトは、そのコンテンツを Ceph RADOS オブジェクトにマップする方法を制御します。ディレクトリーレイアウトは、主にそのディレクトリー内の新しいファイルに継承されたレイアウトを設定します。

ファイルまたはディレクトリーのレイアウトを表示および設定するには、仮想拡張属性または拡張ファイル属性 (`xattrs`) を使用します。layout 属性の名前は、ファイルが通常のファイルかディレクトリーであるかによって異なります。

- 通常ファイルのレイアウト属性は `ceph.file.layout` という名前です。
- ディレクトリーのレイアウト属性は `ceph.dir.layout` と呼ばれます。

### レイアウトの継承

ファイルは、作成時に、親ディレクトリーのレイアウトを継承します。ただし、後続の親ディレクトリーのレイアウトは子には影響を及ぼしません。ディレクトリーにレイアウトが設定されていない場合、ファイルはディレクトリー構造のレイアウトで最も近いディレクトリーからレイアウトを継承します。

### 8.2. ファイルとディレクトリーレイアウトフィールドの設定

`setfattr` コマンドを使用して、ファイルまたはディレクトリーにレイアウトフィールドを設定します。



## 重要

ファイルのレイアウトフィールドを修正すると、ファイルは空にする必要があります。指定しない場合は、エラーが発生します。

### 前提条件

- ノードへのルートレベルのアクセス。

### 手順

- ファイルまたはディレクトリーのレイアウトフィールドを変更するには、次のコマンドを実行します。

### 構文

```
setfattr -n ceph.TYPE.layout.FIELD -v VALUE PATH
```

以下を置き換えます。

- **TYPE** を **file** または **dir** に変更。
- **FIELD** をフィールドの名前に変更。
- **VALUE** をフィールドの新しい値に変更。
- **PATH** をファイルまたはディレクトリーへのパスに変更。

### 例

```
[root@mon ~]# setfattr -n ceph.file.layout.stripe_unit -v 1048576 test
```

### 関連情報

- 詳細は、Red Hat Ceph Storage File System ガイドの [ファイルとディレクトリーレイアウトの概要](#) セクションを参照してください。
- **setfattr(1)** の man ページを参照してください。

## 8.3. ファイルとディレクトリーのレイアウトフィールドの表示

**getfattr** コマンドを使用して、ファイルまたはディレクトリーのレイアウトフィールドを表示します。

### 前提条件

- 稼働中の Red Hat Ceph Storage クラスタがある。
- ストレージクラスター内のすべてのノードへの root レベルのアクセス。

### 手順

- 1つの文字列としてファイルまたはディレクトリーのレイアウトフィールドを表示するには、次のコマンドを実行します。

## 構文

```
getfattr -n ceph.TYPE.layout PATH
```

## 置き換え

- **PATH** をファイルまたはディレクトリーへのパスに変更。
- **TYPE** を **file** または **dir** に変更。

## 例

```
[root@mon ~]# getfattr -n ceph.dir.layout /home/test
ceph.dir.layout="stripe_unit=4194304 stripe_count=2 object_size=4194304
pool=cephfs_data"
```



## 注記

ディレクトリーには、設定するまで明示的なレイアウトがありません。そのため、表示する変更がないため、最初に設定せずにレイアウトを表示しようとすると失敗します。

## 関連情報

- [getfattr\(1\) man ページ](#)
- 詳細は、Red Hat Ceph Storage File System ガイドの [ファイルおよびディレクトリーレイアウトフィールドの設定](#) セクションを参照してください。

## 8.4. 個々のレイアウトフィールドの表示

**getfattr** コマンドを使用して、ファイルまたはディレクトリーの個別のレイアウトフィールドを表示します。

## 前提条件

- 稼働中の Red Hat Ceph Storage クラスタがある。
- ストレージクラスター内のすべてのノードへの root レベルのアクセス。

## 手順

- ファイルまたはディレクトリーの個別のレイアウトフィールドを表示するには、次のコマンドを実行します。

## 構文

```
getfattr -n ceph.TYPE.layout.FIELD_PATH
```

## 置き換え

- **TYPE** を **file** または **dir** に変更。

- FIELD をフィールドの名前に変更。
- PATH をファイルまたはディレクトリーへのパスに変更。

### 例

```
[root@mon ~]# getfattr -n ceph.file.layout.pool test
ceph.file.layout.pool="cephfs_data"
```



### 注記

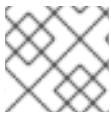
**pool** フィールドのプールは、名前で示されます。ただし、新規作成されたプールは ID で識別できます。

### 関連情報

- **getfattr(1)** man ページ

## 8.5. ディレクトリーレイアウトの削除

**setfattr** コマンドを使用して、ディレクトリーからレイアウトを削除します。



### 注記

ファイルレイアウトを設定する場合は、ファイルの変更や削除ができません。

### 前提条件

- レイアウトを含むディレクトリー。

### 手順

1. ディレクトリーからレイアウトを削除するには、以下のコマンドを実行します。

### 構文

```
setfattr -x ceph.dir.layout DIRECTORY_PATH
```

### 例

```
[user@client ~]$ setfattr -x ceph.dir.layout /home/cephfs
```

2. **pool\_namespace** フィールドを削除するには、以下を実行します。

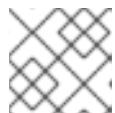
### 構文

```
setfattr -x ceph.dir.layout.pool_namespace DIRECTORY_PATH
```

### 例

```
[user@client ~]$ setfattr -x ceph.dir.layout.pool_namespace /home/cephfs
```

■



### 注記

**pool\_namespace** フィールドは、個別に削除できる唯一のフィールドです。

### 関連情報

- **setfattr(1)** man ページ

### 関連情報

- Red Hat Ceph Storage File System ガイドの [Ceph File System のデプロイメント](#) セクションを参照してください。
- 詳細は、**getfattr(1)** man ページを参照してください。
- 詳細は、**setfattr(1)** man ページを参照してください。

## 第9章 CEPH ファイルシステムのスナップショット

ストレージ管理者は、Ceph File System (CephFS) ディレクトリーの特定の時点のスナップショットを取得できます。CephFS スナップショットは非同期で、作成するディレクトリースナップショットを選択できます。

### 前提条件

- 実行中、および正常な Red Hat Ceph Storage クラスタ
- Ceph File System のデプロイメント

### 9.1. CEPH ファイルシステムのスナップショット

Ceph File System (CephFS) のスナップショット機能は、新しい Ceph File Systems でデフォルトで有効になっていますが、既存の Ceph File Systems で手動で有効にする必要があります。CephFS スナップショットは、Ceph File System のイミュータブルな、ポイントインタイムビューを作成します。CephFS スナップショットは非同期で、CephFS ディレクトリーの特別な非表示ディレクトリー (**.snap**) に保存されます。Ceph ファイルシステム内の任意のディレクトリーのスナップショット作成を指定できます。ディレクトリーを指定すると、スナップショットにはその中のすべてのサブディレクトリーも含まれます。



#### 警告

各 Ceph Metadata Server (MDS) クラスタは snap 識別子を別個に割り当てます。1つのプールを共有する複数の Ceph ファイルシステムのスナップショットを使用すると、スナップショットの競合が発生し、ファイルデータがありません。

### 関連情報

- 詳細は、Red Hat Ceph Storage File System ガイドの [Ceph File System のスナップショットの作成](#) セクションを参照してください。
- 詳細は、Red Hat Ceph Storage File System ガイドの [Ceph File System のスナップショットスケジュールの作成](#) セクションを参照してください。

### 9.2. CEPH ファイルシステムのスナップショットの作成

Ceph File System (CephFS) のイミュータブル (ポイントインタイムビュー) を作成するには、スナップショットを作成します。



#### 注記

新しい Ceph ファイルシステムの場合、スナップショットはデフォルトで有効になります。

### 前提条件

- 実行中、および正常な Red Hat Ceph Storage クラスタ

- Ceph File System のデプロイメント
- Ceph Metadata Server (MDS) ノードへのルートレベルのアクセス。

## 手順

1. Cephadm シェルにログインします。

### 例

```
[root@mds ~]# cephadm shell
```

2. 既存の Ceph ファイルシステムの場合は、スナップショット化機能を有効にします。

### 構文

```
ceph fs set FILE_SYSTEM_NAME allow_new_snaps true
```

### 例

```
[ceph: root@mds ~]# ceph fs set cephfs01 allow_new_snaps true
```

3. **.snap** ディレクトリーに新しい snapshot サブディレクトリーを作成します。

### 構文

```
mkdir NEW_DIRECTORY_PATH
```

### 例

```
[ceph: root@mds ~]# mkdir /.snap/new-snaps
```

この例では **new-snaps** サブディレクトリーを作成し、これにより Ceph Metadata Server (MDS) がスナップショットの作成を開始するように通知します。

- a. スナップショットを削除するには、以下のコマンドを実行します。

### 構文

```
rmdir NEW_DIRECTORY_PATH
```

### 例

```
[ceph: root@mds ~]# rmdir /.snap/new-snaps
```



### 重要

基礎となるスナップショットが含まれる可能性のある root レベルのスナップショットの削除を試みると、失敗します。

## 関連情報



- 詳細は、Red Hat Ceph Storage ファイルシステムガイドの [Ceph File System スナップショットスケジュール](#) セクションを参照してください。
- 詳細は、Red Hat Ceph Storage ファイルシステムガイドの [Ceph File System スナップショット](#) セクションを参照してください。
- Red Hat Ceph Storage File System ガイドの [Ceph File System のデプロイメント](#) セクションを参照してください。

## 第10章 CEPH FILE SYSTEM スナップショットのスケジューリング

ストレージ管理者は、Ceph File System (CephFS) ディレクトリーの特定の時点のスナップショットを取得できます。CephFS スナップショットは非同期で、作成するディレクトリースナップショットを選択できます。

### 前提条件

- 実行中、および正常な Red Hat Ceph Storage クラスター
- Ceph File System のデプロイメント

### 10.1. CEPH ファイルシステムのスナップショットスケジュール

Ceph File System (CephFS) は、ファイルシステムディレクトリーのスナップショットをスケジュールできます。スナップショットのスケジューリングは Ceph Manager によって管理され、Python タイマーに依存します。スナップショットスケジュールデータは CephFS メタデータプールにオブジェクトとして保存され、ランタイム時に、スケジュールデータはすべてシリアルライズされた SQLite データベースに置かれます。



#### 重要

スケジューラーは、ストレージクラスターが通常の負荷がかかる場合に、スナップショットを個別に維持するために指定された時間に基づいて正確です。Ceph Manager の負荷が高い場合は、スナップショットがすぐにスケジュールされない可能性があり、スナップショットが若干遅延させる可能性があります。この場合、次のスケジュールされたスナップショットは遅延がないかのように動作します。遅延しているスケジュールされたスナップショットでは、スケジュール全体にドリフトが発生しません。

### 使用法

Ceph File System (CephFS) のスナップショットスケジューリングは、**snap\_schedule** Ceph Manager モジュールにより管理されます。このモジュールは、スナップショットスケジュールを追加、クエリー、削除し、保持ポリシーを管理するインターフェイスを提供します。このモジュールは **ceph fs snap-schedule** コマンドも実装し、スケジュールを管理する複数のサブコマンドと保持ポリシーも実装します。すべてのサブコマンドは、CephFS ボリュームパスと subvolume パス引数を取り、複数の Ceph File Systems を使用する場合のファイルシステムパスを指定します。CephFS ボリュームパスを指定しない場合、引数は **fs\_map** にリスト表示されている最初のファイルシステムに対してデフォルトで設定され、subvolume パス引数は何も指定しません。

スナップショットスケジュールは、ファイルシステムパス、繰り返しの間隔、および開始時間で識別されます。繰り返し間隔は、2つの後続のスナップショットの間隔を定義します。間隔の形式は、数 + 時間指定 **h(our)**、**d(ay)**、**w(eek)** です。たとえば、間隔が **4h** であれば、4時間ごとに1つのスナップショットが必要になります。開始時間は ISO 形式の文字列の値 **%Y-%m-%dT%H:%M:%S** です。指定されていない場合、開始時間は最後の midnight のデフォルト値を使用します。たとえば、デフォルトの開始時間値を使用して、スナップショットを **14:45** にスケジュールすると、繰り返される間隔が **1h** になると、最初のスナップショットは 15:00 に作成されます。

保持ポリシーは、ファイルシステムパスと保持ポリシーの仕様で識別されます。保持ポリシーの定義は、**COUNT TIME\_PERIOD** の形式で、数字と時間指定のペア、または連結されたペアのいずれかで設定されます。このポリシーにより、スナップショットの数も保持され、スナップショットは少なくとも

特定の期間にわたって保持されます。期間指定とは、**h(our)**、**d(ay)**、**w(eek)**、**m(onth)**、**y(ear)**、および **n** です。**n** の時間指定は特別な修飾子であり、タイミングに関係なくスナップショットの最後の数を保持します。たとえば、**4d** は、1日以上経過した4つのスナップショットを保持します。

## 関連情報

- 詳細は、Red Hat Ceph Storage File System ガイドの [Ceph File System のスナップショットの作成](#) セクションを参照してください。
- 詳細は、Red Hat Ceph Storage File System ガイドの [Ceph File System のスナップショットスケジュールの作成](#) セクションを参照してください。

## 10.2. CEPH ファイルシステムのスナップショットスケジュールの追加

まだ存在しない CephFS パスのスナップショットスケジュールを追加します。1つのパスに対して1つ以上のスケジュールを作成できます。繰り返される間隔と開始時間が異なる場合、スケジュールは異なると見なされます。

CephFS パスには保持ポリシーを1つだけ指定できますが、保持ポリシーは複数のカウントタイムペアを持つことができます。



### 注記

スケジューラーモジュールが有効になると、**ceph fs snap-schedule** コマンドを実行すると、利用可能なサブコマンドと、その使用形式が表示されます。

## 前提条件

- 実行中、および正常な Red Hat Ceph Storage クラスタ
- Ceph File System のデプロイメント
- Ceph Manager および Metadata Server (MDS) ノードへのルートレベルのアクセス。
- ファイルシステムで CephFS スナップショットを有効にします。

## 手順

1. Ceph Manager ノードで Cephadm シェルにログインします。

### 例

```
[root@host01 ~]# cephadm shell
```

2. **snap\_schedule** モジュールを有効にします。

### 例

```
[ceph: root@host01 /]# ceph mgr module enable snap_schedule
```

3. クライアントノードにログインします。

### 例

■

```
[root@host02 ~]# cephadm shell
```

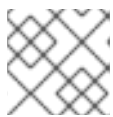
- Ceph File System の新しいスケジュールを追加します。

### 構文

```
ceph fs snap-schedule add FILE_SYSTEM_VOLUME_PATH REPEAT_INTERVAL  
[START_TIME] --fs CEPH_FILE_SYSTEM_NAME
```

### 例

```
[ceph: root@host02 /]# ceph fs snap-schedule add /cephfs_kernelf739cwtus2/pmo9axbws  
1h 2022-06-27T21:50:00 --fs mycephfs
```



### 注記

**START\_TIME** は、ISO8601 形式で表されます。

この例では、ファイルシステム **mycephfs** 内のパス **/cephfs** のスナップショットスケジュールを作成し、1時間ごとにスナップショットを作成し、2022年6月27日午後9時50分に開始します。

- CephFS ボリュームパスのスナップショット用に新たな保持ポリシーを追加します。

### 構文

```
ceph fs snap-schedule retention add FILE_SYSTEM_VOLUME_PATH  
[COUNT_TIME_PERIOD_PAIR] TIME_PERIOD COUNT
```

### 例

```
[ceph: root@host02 /]# ceph fs snap-schedule retention add /cephfs h 14 1  
[ceph: root@host02 /]# ceph fs snap-schedule retention add /cephfs d 4 2  
[ceph: root@host02 /]# ceph fs snap-schedule retention add /cephfs 14h4w 3
```

- この例では、14 スナップショットを1時間以上保持します。
- この例では、4つのスナップショットを1日以上保持します。
- この例では、14時間ごとに1週間、また4週間のスナップショットを保持します。

- スナップショットスケジュールを一覧表示して、新しいスケジュールが作成されたことを確認します。

### 構文

```
ceph fs snap-schedule list FILE_SYSTEM_VOLUME_PATH [--format=plain|json] [--  
recursive=true]
```

### 例

```
[ceph: root@host02 /]# ceph fs snap-schedule list /cephfs --recursive=true
```

この例では、ディレクトリーツリーのすべてのスケジュールをリスト表示しています。

7. スナップショットスケジュールのステータスを確認します。

### 構文

```
ceph fs snap-schedule status FILE_SYSTEM_VOLUME_PATH [--format=plain|json]
```

### 例

```
[ceph: root@host02 /]# ceph fs snap-schedule status /cephfs --format=json
```

以下の例では、CephFS **/cephfs** パスのスナップショットスケジュールのステータスを JSON 形式で表示しています。デフォルトの形式はプレーンテキストで、指定されていない場合はプレーンテキストになります。

### 関連情報

- 詳細は、Red Hat Ceph Storage ファイルシステムガイドの [Ceph File System スナップショットスケジュール](#) セクションを参照してください。
- 詳細は、Red Hat Ceph Storage ファイルシステムガイドの [Ceph File System スナップショット](#) セクションを参照してください。

## 10.3. CEPH FILE SYSTEM サブボリュームのスナップショットスケジュールの追加

Ceph File System (CephFS) サブボリュームスナップショットの保持ポリシーを管理するために、1つのパスに対して異なるスケジュールを設定できます。

繰り返される間隔と開始時間が異なる場合、スケジュールは異なると見なされます。

まだ存在しない CephFS ファイルパスのスナップショットスケジュールを追加します。CephFS パスには保持ポリシーを1つだけ指定できますが、保持ポリシーは複数のカウントタイムペアを持つことができます。



#### 注記

スケジューラーモジュールが有効になると、**ceph fs snap-schedule** コマンドを実行すると、利用可能なサブコマンドと、その使用形式が表示されます。



#### 注記

現在、スナップショットのスケジュールを設定できるのは、デフォルトのサブボリュームグループに属するサブボリュームのみです。

### 前提条件

- Ceph File System (CephFS) がデプロイされた稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。

- Ceph Manager ノードの読み取りおよび書き込み機能。
- CephFS サブボリュームとサブボリュームグループが作成されました。

## 手順

1. サブボリュームパスを取得します。

### 構文

```
ceph fs subvolume getpath VOLUME_NAME SUBVOLUME_NAME
SUBVOLUME_GROUP_NAME
```

### 例

```
[ceph: root@host02 /]# ceph fs subvolume getpath cephfs subvol_1 subvolgroup_1
```

2. Ceph File System サブボリュームパスの新しいスケジュールを追加します。

### 構文

```
ceph fs snap-schedule add ./ SNAP_SCHEDULE [START_TIME] --fs
CEPH_FILE_SYSTEM_NAME --subvol SUBVOLUME_NAME
```

### 例

```
[ceph: root@host02 /]# ceph fs snap-schedule add /cephfs_kernelf739cwtus2/pmo9axbws
1h 2022-06-27T21:50:00 --fs cephfs --subvol subvol_1
Schedule set for path ./
```



### 注記

**START\_TIME** は、ISO8601形式で表されます。

この例では、サブボリュームパスのスナップショットスケジュールを作成し、1時間ごとにスナップショットを作成します。2022年6月27日午後9時50分に開始します。

3. CephFS サブボリュームのスナップショットスケジュールの新しい保持ポリシーを追加します。

### 構文

```
ceph fs snap-schedule retention add SUBVOLUME_VOLUME_PATH
[COUNT_TIME_PERIOD_PAIR] TIME_PERIOD COUNT
```

### 例

```
[ceph: root@host02 /]# ceph fs snap-schedule retention add
/volumes/_nogroup/subvol_1/85a615da-e8fa-46c1-afc3-0eb8ae64a954/.. h 14 1
[ceph: root@host02 /]# ceph fs snap-schedule retention add
/volumes/_nogroup/subvol_1/85a615da-e8fa-46c1-afc3-0eb8ae64a954/.. d 4 2
[ceph: root@host02 /]# ceph fs snap-schedule retention add
```

```
/volumes/_nogroup/subvol_1/85a615da-e8fa-46c1-afc3-0eb8ae64a954/.. 14h4w 3
```

```
Retention added to path /volumes/_nogroup/subvol_1/85a615da-e8fa-46c1-afc3-0eb8ae64a954/..
```

- 1 この例では、14 スナップショットを1時間以上保持します。
- 2 この例では、4つのスナップショットを1日以上保持します。
- 3 この例では、14時間ごとに1週間、また4週間のスナップショットを保持します。

4. スナップショットスケジュールをリスト表示します。

#### 構文

```
ceph fs snap-schedule list SUBVOLUME_VOLUME_PATH [--format=plain|json] [--recursive=true]
```

#### 例

```
[ceph: root@host02 /]# ceph fs snap-schedule list / --recursive=true
/volumes/_nogroup/subv1/85a615da-e8fa-46c1-afc3-0eb8ae64a954/.. 4h
```

この例では、ディレクトリツリーのすべてのスケジュールをリスト表示しています。

5. スナップショットスケジュールのステータスを確認します。

#### 構文

```
ceph fs snap-schedule status SUBVOLUME_VOLUME_PATH [--format=plain|json]
```

#### 例

```
[ceph: root@host02 /]# ceph fs snap-schedule status /volumes/_nogroup/subv1/85a615da-e8fa-46c1-afc3-0eb8ae64a954/.. --format=json
{"fs": "cephfs", "subvol": "subvol_1", "path": "/volumes/_nogroup/subvol_1/85a615da-e8fa-46c1-afc3-0eb8ae64a954/..", "rel_path": "/..", "schedule": "4h", "retention": {"h": 14}, "start": "2022-05-16T14:00:00", "created": "2023-03-20T08:47:18", "first": null, "last": null, "last_pruned": null, "created_count": 0, "pruned_count": 0, "active": true}
```

この例では、`/volumes/_nogroup/subv1/85a615da-e8fa-46c1-afc3-0eb8ae64a954/..` パスのスナップショットスケジュールのステータスをJSON形式で表示します。デフォルトの形式はプレーンテキストで、指定されていない場合はプレーンテキストになります。

## 10.4. CEPH ファイルシステムのスナップショットスケジュールのアクティブ化

このセクションでは、Ceph ファイルシステム (CephFS) のスナップショットスケジュールを手動でアクティブに設定する手順を説明します。

## 前提条件

- Ceph File System (CephFS) がデプロイされた稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。

## 手順

- スナップショットスケジュールを有効にします。

### 構文

```
ceph fs snap-schedule activate FILE_SYSTEM_VOLUME_PATH [REPEAT_INTERVAL]
```

### 例

```
[ceph: root@host01 /]# ceph fs snap-schedule activate /cephfs
```

以下の例では、CephFS **/cephfs** パスのすべてのスケジュールを有効にします。

## 10.5. CEPH ファイルシステムサブボリュームのスナップショットスケジュールのアクティブ化

このセクションでは、Ceph ファイルシステム (CephFS) サブボリュームのスナップショットスケジュールを手動でアクティブに設定する手順を説明します。

## 前提条件

- Ceph File System (CephFS) がデプロイされた稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。

## 手順

- スナップショットスケジュールを有効にします。

### 構文

```
ceph fs snap-schedule activate SUB_VOLUME_PATH [REPEAT_INTERVAL]
```

### 例

```
[ceph: root@host01 /]# ceph fs snap-schedule activate  
/volumes/_nogroup/subvol_1/85a615da-e8fa-46c1-afc3-0eb8ae64a954/..
```

この例では、CephFS **/volumes/\_nogroup/subvol\_1/85a615da-e8fa-46c1-afc3-0eb8ae64a954/..** パスのすべてのスケジュールをアクティブ化します。



## 10.6. CEPH ファイルシステムのスナップショットスケジュールの非アクティブ化

このセクションでは、Ceph ファイルシステム (CephFS) のスナップショットスケジュールを手動で非アクティブに設定する手順を説明します。このアクションにより、スナップショットは、再度アクティブ化されるまで、スケジュールから除外されます。

### 前提条件

- Ceph File System (CephFS) がデプロイされた稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。
- スナップショットスケジュールが作成され、アクティブな状態になっています。

### 手順

- CephFS パスのスナップショットスケジュールを非アクティブにします。

### 構文

```
ceph fs snap-schedule deactivate FILE_SYSTEM_VOLUME_PATH [REPEAT_INTERVAL]
```

### 例

```
[ceph: root@host02 ~]# ceph fs snap-schedule deactivate /cephfs 1d
```

この例では、**/cephfs** パスの日次のスナップショットを非アクティブ化して、それ以降のスナップショットの作成を一時停止します。

## 10.7. CEPH ファイルシステムサブボリュームのスナップショットスケジュールの非アクティブ化

このセクションでは、Ceph ファイルシステム (CephFS) サブボリュームのスナップショットスケジュールを手動で非アクティブに設定する手順を説明します。このアクションにより、スナップショットは再びアクティブ化されるまでスケジュールから除外されます。

### 前提条件

- Ceph File System (CephFS) がデプロイされた稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。
- スナップショットスケジュールが作成され、アクティブな状態になっています。

### 手順

- CephFS サブボリュームパスのスナップショットスケジュールを非アクティブにします。

## 構文

```
ceph fs snap-schedule deactivate SUB_VOLUME_PATH [REPEAT_INTERVAL]
```

## 例

```
[ceph: root@host02 ~]# ceph fs snap-schedule deactivate  
/volumes/_nogroup/subvol_1/85a615da-e8fa-46c1-afc3-0eb8ae64a954/.. 1d
```

この例では、`/volumes/_nogroup/subvol_1/85a615da-e8fa-46c1-afc3-0eb8ae64a954/..` パスの日次スナップショットを非アクティブ化し、それ以降のスナップショットの作成を一時停止します。

## 10.8. CEPH ファイルシステムのスナップショットスケジュールの削除

このセクションでは、Ceph ファイルシステム (CephFS) のスナップショットスケジュールを削除する手順を説明します。

### 前提条件

- Ceph File System (CephFS) がデプロイされた稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。
- スナップショットスケジュールが作成されます。

### 手順

1. 特定のスナップショットスケジュールを削除します。

## 構文

```
ceph fs snap-schedule remove FILE_SYSTEM_VOLUME_PATH [REPEAT_INTERVAL]  
[START_TIME]
```

## 例

```
[ceph: root@host02 ~]# ceph fs snap-schedule remove /cephfs 4h 2022-05-16T14:00:00
```

この例では、4時間ごとにスナップショットを作成し、2022年5月16日午後2:00に開始する、`/cephfs` ボリュームの特定のスナップショットスケジュールを削除します。

2. 特定の CephFS ボリュームパスのすべてのスナップショットスケジュールを削除します。

## 構文

```
ceph fs snap-schedule remove FILE_SYSTEM_VOLUME_PATH
```

## 例

```
[ceph: root@host02 ~]# ceph fs snap-schedule remove /cephfs
```

この例では、**/cephfs** ボリュームパスのすべてのスナップショットスケジュールを削除します。

## 10.9. CEPH ファイルシステムサブボリュームのスナップショットスケジュールの削除

このセクションでは、Ceph ファイルシステム (CephFS) サブボリュームのスナップショットスケジュールを削除する手順を説明します。

### 前提条件

- Ceph File System (CephFS) がデプロイされた稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。
- スナップショットスケジュールが作成されます。

### 手順

- 特定のスナップショットスケジュールを削除します。

### 構文

```
ceph fs snap-schedule remove SUB_VOLUME_PATH [REPEAT_INTERVAL]  
[START_TIME]
```

### 例

```
[ceph: root@host02 ~]# ceph fs snap-schedule remove  
/volumes/_nogroup/subvol_1/85a615da-e8fa-46c1-afc3-0eb8ae64a954/.. 4h 2022-05-  
16T14:00:00
```

この例では、2022年5月16日午後2:00に開始する、4時間ごとにスナップショットを作成する **/volumes/\_nogroup/subvol\_1/85a615da-e8fa-46c1-afc3-0eb8ae64a954/..** ボリュームの特定のスナップショットスケジュールを削除します。

## 10.10. CEPH ファイルシステムのスナップショットスケジュール保持ポリシーの削除

このセクションでは、Ceph ファイルシステム (CephFS) のスケジュールされたスナップショットの保持ポリシーを削除する手順について説明します。

### 前提条件

- Ceph File System (CephFS) がデプロイされた稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。

- Ceph Manager ノードの読み取りおよび書き込み機能。
- CephFS ボリュームパス用に作成されたスナップショットスケジュール。

## 手順

- CephFS パスの保持ポリシーを削除します。

### 構文

```
ceph fs snap-schedule retention remove FILE_SYSTEM_VOLUME_PATH
[COUNT_TIME_PERIOD_PAIR] TIME_PERIOD COUNT
```

### 例

```
[ceph: root@host02 ~]# ceph fs snap-schedule retention remove /cephfs h 4 1
[ceph: root@host02 ~]# ceph fs snap-schedule retention remove /cephfs 14d4w 2
```

- 1** この例では、4 時間のスナップショットを削除します。
- 2** この例では、14 日次とスナップショット 4 週間を削除します。

## 10.11. CEPH ファイルシステムサブボリュームのスナップショットスケジュール保持ポリシーの削除

このセクションでは、Ceph ファイルシステム (CephFS) サブボリュームのスケジュールされたスナップショットの保持ポリシーを削除する手順について説明します。

### 前提条件

- Ceph File System (CephFS) がデプロイされた稼働中の Red Hat Ceph Storage クラスタ。
- Ceph Monitor での少なくとも読み取りアクセス。
- Ceph Manager ノードの読み取りおよび書き込み機能。
- CephFS サブボリュームパス用に作成されたスナップショットスケジュール。

## 手順

- CephFS サブボリュームパスの保持ポリシーを削除します。

### 構文

```
ceph fs snap-schedule retention remove SUB_VOLUME_PATH
[COUNT_TIME_PERIOD_PAIR] TIME_PERIOD COUNT
```

### 例

```
[ceph: root@host02 ~]# ceph fs snap-schedule retention remove
/volumes/_nogroup/subvol_1/85a615da-e8fa-46c1-afc3-0eb8ae64a954/.. h 4 1
```

```
[ceph: root@host02 ~]# ceph fs snap-schedule retention remove  
/volumes/_nogroup/subvol_1/85a615da-e8fa-46c1-afc3-0eb8ae64a954/.. 14d4w 2
```

- 1 この例では、4時間のスナップショットを削除します。
  - 2 この例では、14日次とスナップショット4週間を削除します。
- Red Hat Ceph Storage File System ガイドの [Ceph File System のデプロイメント](#) セクションを参照してください。

## 第11章 CEPH FILE SYSTEM のミラー

ストレージ管理者は、別の Red Hat Ceph Storage クラスターのリモート Ceph File System に Ceph File System (CephFS) を複製できます。

### 前提条件

- ソースおよびターゲットストレージクラスターは、Red Hat Ceph Storage 6.0 以降を実行している必要があります。

### 11.1. CEPH FILE SYSTEM ミラーリング

Ceph File System (CephFS) は、別の Red Hat Ceph Storage クラスター上のリモート Ceph FS へのスナップショットの非同期レプリケーションをサポートします。スナップショットの同期は、スナップショットデータをリモートの Ceph ファイルシステムにコピーし、同じ名前のリモートターゲットに新しいスナップショットを作成します。スナップショット同期用に特定のディレクトリーを設定できます。

CephFS ミラーの管理は、CephFS ミラーリングデーモン (**cephfs-mirror**) により実行されます。このスナップショットデータは、リモートの CephFS への一括コピーを実行することで同期されます。スナップショットのペアの同期順序は、**snap-id** を使用して作成時に決定されます。



#### 重要

ハードリンクの同期はサポートされていません。ハードリンクされたファイルは、通常のファイルとして同期されます。

CephFS ミラーリングには、スナップショットインカーネーションや高可用性などの機能が含まれます。これらは、推奨される制御インターフェイスである Ceph Manager **mirroring** モジュールを介して管理できます。

#### Ceph Manager モジュールとインターフェイス

Ceph Manager **mirroring** モジュールは、デフォルトでは無効になっています。ディレクトリースナップショットのミラーリングを管理するためのインターフェイスを提供します。Ceph Manager インターフェイスは、ほとんどの場合、CephFS ミラーリングを管理するためのモニターコマンドのラッパーです。これらは、推奨される制御インターフェイスです。

Ceph Manager **mirroring** モジュールは、Ceph Manager プラグインとして実装されます。これは、Synchronization のためにディレクトリーを **cephfs-mirror** デーモンに割り当てるロールを果たします。

Ceph Manager の **mirroring** モジュールは、ディレクトリースナップショットのミラーリングを制御するコマンドのファミリーも提供します。**mirroring** モジュールは、**cephfs-mirror** デーモンを管理しません。**cephfs-mirror** デーモンの停止、起動、再起動、および有効化は **systemctl** によって制御されますが、**cephadm** によって管理されます。



#### 注記

ミラーリングモジュールコマンドは、**fs mirror** 接頭辞を使用する監視コマンドと比較して、**fs snapshot mirror** 接頭辞を使用します。module コマンド接頭辞を使用して、ディレクトリースナップショットのミラーリングを制御していることを確認してください。

## スナップショットの再生成

スナップショットが削除され、同じ名前異なる内容で再作成される場合があります。ユーザーは以前に古いスナップショットを同期し、ミラーリングが無効になったときにスナップショットを再作成できました。スナップショット名を使用して継続点を推測すると、新しいスナップショットが生成され、同期のために取得されることはありません。

セカンダリーファイルシステムのスナップショットには、同期元のスナップショットのスナップ ID が格納されます。このメタデータは、Ceph メタデータサーバーの **SnapInfo** 構造に保存されます。

## 高可用性

複数の **cephfs-mirror** デーモンを2つ以上のノードにデプロイして、ディレクトリースナップショットの同期で同時実行を実現できます。**cephfs-mirror** デーモンがデプロイまたは終了されると、Ceph Manager **mirroring** モジュールは変更された **cephfs-mirror** デーモンのセットを検出し、新しいセット間でディレクトリーの割り当てを再調整して、高可用性を提供します。

**cephfs-mirror** デーモンは、単純な M/N ポリシーを使用して同期負荷を共有します。ここで、M はディレクトリーの数、N は **cephfs-mirror** デーモンの数です。

## Ceph File System ミラーピアの再追加

別のクラスター内の CephFS にピアを再追加または再割り当てする場合は、すべてのミラーデーモンがピアへの同期を停止していることを確認してください。これは、**fs mirror status** コマンドを使用して確認できます。ピア UUID はコマンド出力に表示されません。

別の CephFS に再追加する前に、同期されたディレクトリーをピアからパーズします。特に、新しいプライマリーファイルシステムに存在する可能性のあるディレクトリーを削除します。以前に同期した同じプライマリーファイルシステムにピアを再度追加する場合、これは必要ありません。

## 関連情報

- **fs mirror status** コマンドの詳細は、[Ceph ファイルシステムのミラーステータスの表示](#) を参照してください。

## 11.2. CEPH ファイルシステムのスナップショットミラーの設定

Ceph File System (CephFS) を設定して、リモートの Red Hat Ceph Storage クラスターの別の CephFS にスナップショットを複製するようにミラーリングできます。



### 注記

リモートストレージクラスターへの同期にかかる時間は、ファイルのサイズとミラーリングパス内のファイルの合計数によって異なります。

## 前提条件

- ソースおよびターゲットストレージクラスターは、Red Hat Ceph Storage 6.0 以降を実行していて正常である必要があります。
- ソースおよびターゲットストレージクラスターの Ceph Monitor ノードへのルートレベルのアクセス。
- ストレージクラスターにデプロイされた少なくとも1つの Ceph ファイルシステム。

## 手順

1. Cephadm シェルにログインします。

#### 例

```
[root@host01 ~]# cephadm shell
```

2. ソースストレージクラスターで、CephFS ミラーリングデーモンをデプロイします。

#### 構文

```
ceph orch apply cephfs-mirror ["NODE_NAME"]
```

#### 例

```
[ceph: root@host01 /]# ceph orch apply cephfs-mirror "node1.example.com"
Scheduled cephfs-mirror update...
```

このコマンドにより、**cephfs-mirror** という名前の Ceph ユーザーが作成され、特定のノードに **cephfs-mirror** デーモンがデプロイされます。

- a. オプション: 複数の CephFS ミラーリングデーモンをデプロイし、高可用性を実現します。

#### 構文

```
ceph orch apply cephfs-mirror --placement="PLACEMENT_SPECIFICATION"
```

#### 例

```
[ceph: root@host01 /]# ceph orch apply cephfs-mirror --placement="3 host1 host2 host3"
Scheduled cephfs-mirror update...
```

この例では、3つの **cephfs-mirror** デーモンを異なるホストにデプロイします。



#### 警告

次のエラーが発生するため、ホストをコンマで区切らないでください。

```
Error EINVAL: name component must include only a-z, 0-9, and -
```

3. ターゲットストレージクラスターで、それぞれの CephFS ピア用にユーザーを作成します。

#### 構文

```
ceph fs authorize FILE_SYSTEM_NAME CLIENT_NAME / rwps
```

#### 例



```
[ceph: root@host01 /]# ceph fs authorize cephfs client.mirror_remote / rwps
[client.mirror_remote]
  key = AQCjZ5Jg739AAxAAxdulKoTZbiFJ0lgose8luQ==
```

4. ソースストレージクラスターで、CephFS ミラーリングモジュールを有効にします。

#### 例

```
[ceph: root@host01 /]# ceph mgr module enable mirroring
```

5. ソースストレージクラスターで、Ceph File System でミラーリングを有効にします。

#### 構文

```
ceph fs snapshot mirror enable FILE_SYSTEM_NAME
```

#### 例

```
[ceph: root@host01 /]# ceph fs snapshot mirror enable cephfs
```

- a. オプション: スナップショットミラーリングを無効にします。

#### 構文

```
ceph fs snapshot mirror disable FILE_SYSTEM_NAME
```

#### 例

```
[ceph: root@host01 /]# ceph fs snapshot mirror disable cephfs
```



#### 警告

ファイルシステムでスナップショットミラーリングを無効にすると、設定されたピアが削除されます。ピアをブートストラップして再度インポートする必要があります。

6. ターゲットピアストレージクラスターを準備します。
  - a. ターゲットノードで、**mirroring** Ceph Manager モジュールを有効にします。

#### 例

```
[ceph: root@host01 /]# ceph mgr module enable mirroring
```

- b. 同じターゲットノードで、ピアブートストラップを作成します。

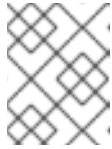
#### 構文



```
ceph fs snapshot mirror peer_remove FILE_SYSTEM_NAME PEER_UUID
```

### 例

```
[ceph: root@host01 /]# ceph fs snapshot mirror peer_remove cephfs e5ecb883-097d-492d-b026-a585d1d7da79
```



### 注記

ピア UUID 値を見つける方法は、[Ceph ファイルシステムのミラーステータスの表示](#) を参照してください。

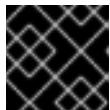
9. ソースストレージクラスターで、スナップショットミラーリングのディレクトリーを設定します。

### 構文

```
ceph fs snapshot mirror add FILE_SYSTEM_NAME PATH
```

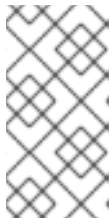
### 例

```
[ceph: root@host01 /]# ceph fs snapshot mirror add cephfs /volumes/_nogroup/subvol_1
```



### 重要

Ceph ファイルシステム内の絶対パスのみが有効です。



### 注記

Ceph Manager の **mirroring** モジュールは、パスを正規化します。たとえば、`/d1/d2/./dN` ディレクトリーは `/d1/d2` と同等です。ミラーリング用にディレクトリーが追加されると、その上位ディレクトリーおよびサブディレクトリーがミラーリング用に追加されなくなります。

- a. オプション: ディレクトリーのスナップショットミラーリングを停止します。

### 構文

```
ceph fs snapshot mirror remove FILE_SYSTEM_NAME PATH
```

### 例

```
[ceph: root@host01 /]# ceph fs snapshot mirror remove cephfs /home/user1
```

### 関連情報

- 詳細は、Red Hat Ceph Storage ファイルシステムガイドの [Ceph File System のミラーステータスの表示](#) セクションを参照してください。

- 詳細は、Red Hat Ceph Storage ファイルシステムガイドの [Ceph ファイルシステムのミラーリング](#) セクションを参照してください。

### 11.3. CEPH FILE システムのミラーステータスの表示

Ceph File System (CephFS) ミラーデーモン (**cephfs-mirror**) は、CephFS ミラーリングステータスの変更に関する非同期通知とピアの更新と共に行われます。CephFS ミラーリングモジュールは、ミラーデーモンのステータスを確認するためのミラーデーモンステータスインターフェイスを提供します。詳細については、コマンドを使用して **cephfs-mirror** 管理ソケットにクエリーを実行し、ミラーのステータスとピアのステータスを取得できます。

#### 前提条件

- 稼働中の Red Hat Ceph Storage クラスタがある。
- ミラーリングが有効にされている Ceph File System のデプロイメントを少なくとも1つ。
- CephFS ミラーリングデーモンを実行するノードへのルートレベルのアクセス。

#### 手順

1. Cephadm シェルにログインします。

#### 例

```
[root@host01 ~]# cephadm shell
```

2. **cephfs-mirror** デーモンのステータスを確認します。

#### 構文

```
ceph fs snapshot mirror daemon status
```

#### 例

```
[ceph: root@host01 /]# ceph fs snapshot mirror daemon status
[
  {
    "daemon_id": 15594,
    "filesystems": [
      {
        "filesystem_id": 1,
        "name": "cephfs",
        "directory_count": 1,
        "peers": [
          {
            "uuid": "e5ecb883-097d-492d-b026-a585d1d7da79",
            "remote": {
              "client_name": "client.mirror_remote",
              "cluster_name": "remote-site",
              "fs_name": "cephfs"
            },
            "stats": {
              "failure_count": 1,

```

```

        "recovery_count": 0
      }
    }
  ]
}
]

```

詳細については、以下で説明する管理ソケットインターフェイスを使用してください。

3. CephFS ミラーリングデーモンを実行しているノードで Ceph File System ID を見つけます。

### 構文

```
ceph --admin-daemon PATH_TO_THE_ASOK_FILE help
```

### 例

```

[ceph: root@host01 /]# ceph --admin-daemon /var/run/ceph/1011435c-9e30-4db6-b720-5bf482006e0e/ceph-client.cephfs-mirror.node1.bndvox.asok help
{
  ...
  "fs mirror peer status cephfs@11 1011435c-9e30-4db6-b720-5bf482006e0e": "get peer mirror status",
  "fs mirror status cephfs@11": "get filesystem mirror status",
  ...
}

```

この例の Ceph File System ID は **cephfs@11** です。



### 注記

ミラーリングが無効になっている場合、ファイルシステムのそれぞれの **fs mirror status** コマンドは、**help** コマンドに表示されません。

4. ミラーステータスを表示します。

### 構文

```
ceph --admin-daemon PATH_TO_THE_ASOK_FILE fs mirror status FILE_SYSTEM_NAME@_FILE_SYSTEM_ID
```

### 例

```

[ceph: root@host01 /]# ceph --admin-daemon /var/run/ceph/1011435c-9e30-4db6-b720-5bf482006e0e/ceph-client.cephfs-mirror.node1.bndvox.asok fs mirror status cephfs@11
{
  "rados_inst": "192.168.0.5:0/1476644347",
  "peers": {
    "1011435c-9e30-4db6-b720-5bf482006e0e": { 1
      "remote": {
        "client_name": "client.mirror_remote",

```

```

        "cluster_name": "remote-site",
        "fs_name": "cephfs"
    }
},
"snap_dirs": {
    "dir_count": 1
}
}

```

- ① これは、固有のピア UUID です。

5. ピアステータスを表示します。

### 構文

```
ceph --admin-daemon PATH_TO_ADMIN_SOCKET fs mirror status
FILE_SYSTEM_NAME@FILE_SYSTEM_ID PEER_UUID
```

### 例

```
[ceph: root@host01 /]# ceph --admin-daemon /var/run/ceph/cephfs-mirror.asok fs mirror peer
status cephfs@11 1011435c-9e30-4db6-b720-5bf482006e0e
{
  "/home/user1": {
    "state": "idle", ①
    "last_synced_snap": {
      "id": 120,
      "name": "snap1",
      "sync_duration": 0.079997898999999997,
      "sync_time_stamp": "274900.558797s"
    },
    "snaps_synced": 2, ②
    "snaps_deleted": 0, ③
    "snaps_renamed": 0
  }
}

```

**state** は、以下の 3 つの値のいずれかになります。

- ① **idle** は、ディレクトリーが現在同期していないことを意味します。
- ② **syncing** とは、ディレクトリーが現在同期中であることを意味します。
- ③ **failed** の原因は、ディレクトリーが連続した失敗の上限に達したことを意味します。

連続する失敗のデフォルト数は 10 で、デフォルトの再試行間隔は 60 秒です。

1. **cephfs-mirror** デーモンがマップされているディレクトリーを表示します。

### 構文

```
ceph fs snapshot mirror dirmap FILE_SYSTEM_NAME PATH
```

■  
例

```
[ceph: root@host01 /]# ceph fs snapshot mirror dirmap cephfs /volumes/_nogroup/subvol_1
{
  "instance_id": "25184", ❶
  "last_shuffled": 1661162007.012663,
  "state": "mapped"
}
```

- ❶ **instance\_id** は、**cephfs-mirror** デーモンに関連付けられた RADOS インスタンス ID です。

## 例

```
[ceph: root@host01 /]# ceph fs snapshot mirror dirmap cephfs /volumes/_nogroup/subvol_1
{
  "reason": "no mirror daemons running",
  "state": "stalled" ❶
}
```

- ❶ **stalled** 状態は、CephFS ミラーリングが停止していることを意味します。

2 番目の例は、ミラーデーモンが実行されていないときのコマンド出力を示しています。

## 関連情報

- 詳細は、Red Hat Ceph Storage ファイルシステムガイドの [Ceph ファイルシステムのミラーセクション](#)を参照してください。

## 関連情報

- 詳細は、Red Hat Ceph Storage File System ガイドの [Ceph File System のデプロイメントセクション](#)を参照してください。
- 詳細は、[Red Hat Ceph Storage インストールガイド](#)を参照してください。
- 詳細は、Red Hat Ceph Storage File System ガイドの [Ceph File System Metadata Server](#)を参照してください。
- 詳細は、Red Hat Ceph Storage ファイルシステムガイドの [Ceph File System ミラーセクション](#)を参照してください。

## 付録A CEPH FILE SYSTEM のヘルスメッセージ

### クラスターのヘルスチェック

Ceph Monitor デーモンは、メタデータサーバー (MDS) の特定の状態に応じてヘルスメッセージを生成します。以下は、ヘルスメッセージとその説明です。

#### mds rank(s) <ranks> have failed

現在、1つ以上の MDS ランクが MDS デーモンに割り当てられていません。ストレージクラスターは、適切な置き換えデーモンが開始するまで回復しません。

#### MDS rank(s)<ranks> is damaged

MDS ランク 1つまたは複数で、保存されたメタデータに重大な破損が生じ、メタデータが修復されるまで再度起動できません。

#### MDS クラスターが動作が低下しています。

現在、MDS のランク 1つ以上が稼働していないため、この状況が解決されるまで、クライアントはメタデータ I/O を一時停止する可能性があります。これには、実行に失敗したか、破損したランクが含まれます。また、MDS で実行していても **active** 状態でないランクも含まれます (例: **replay** 状態)。

#### mds <names> are laggy

MDS デーモンは、**mds\_beacon\_interval** オプションで指定した間隔で、監視にメッセージを送る必要があります。デフォルトは 4 秒です。MDS デーモンが、**mds\_beacon\_grace** オプションで指定された時間内にメッセージ送信に失敗した場合、デフォルトは 15 秒です。Ceph Monitor は MDS デーモンに **laggy** とマークし、利用可能な場合には自動的にスタンバイデーモンに置き換えます。

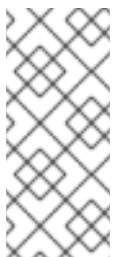
### デーモンでレポートされたヘルスチェック

MDS デーモンは、さまざまな不要な状況を特定し、それらを **ceph status** コマンドの出力で返すことができます。これらの条件には人が判読できるメッセージがあり、JSON の出力に表示される **MDS\_HEALTH** を開始するための一意のコードもあります。以下は、デーモンメッセージ、それらのコード、および説明のリストです。

#### "Behind on trimming..."

コード: MDS\_HEALTH\_TRIM

CephFS は、ログセグメントに分割されるメタデータジャーナルを維持します。ジャーナルの長さ (セグメント数) は、**mds\_log\_max\_segments** 設定で制御されます。セグメントの数が設定を超えた場合、MDS はメタデータの書き込みを開始し、最も古いセグメントを削除 (トリミング) できるようにします。このプロセスの速度が遅い場合や、ソフトウェアのバグがトリミングされると、この健全性メッセージが表示されます。このメッセージに表示されるしきい値は、セグメントの数が double **mds\_log\_max\_segments** となるものです。



#### 注記

トリム警告が発生した場合は、**mds\_log\_max\_segments** を増やすことを推奨します。ただし、クラスターの健全性が回復し、トリム警告が表示されなくなったら、必ずこの設定をデフォルトにリセットしてください。MDS がトリミングに追いつくことができるように、**mds\_log\_max\_segments** を 256 に設定することを推奨します。

#### "Client <name> failing to respond to capability release"

コード: MDS\_HEALTH\_CLIENT\_LATE\_RELEASE, MDS\_HEALTH\_CLIENT\_LATE\_RELEASE\_MANY

CephFS クライアントは、MDS により機能が発行されます。この機能はロックのように機能します。たとえば、別のクライアントがアクセスする必要がある場合、MDS はクライアントに対してそ



の機能を解放するよう要求します。クライアントが応答しない場合は、プロンプトが表示されたら、またはこれをまったく使用できない可能性があります。このメッセージは、クライアントが **mds\_revoke\_cap\_timeout** オプションで指定された時間 (デフォルトは 60 秒) に準拠するために時間がかかる場合に表示されます。

#### "Client <name> failing to respond to cache pressure"

コード: MDS\_HEALTH\_CLIENT\_RECALL, MDS\_HEALTH\_CLIENT\_RECALL\_MANY

クライアントはメタデータキャッシュを維持します。クライアントキャッシュ内の inode などの項目は、MDS キャッシュでも固定されます。MDS がキャッシュサイズの制限内に留まるように MDS を縮小する必要がある場合、MDS はメッセージをクライアントに送信してキャッシュを縮小します。クライアントが応答しない場合は、MDS がキャッシュサイズ内に適切に残らないようにすることができます。また、MDS は最終的にメモリーを使い果たし、予期せずに終了する可能性があります。このメッセージは、クライアントが **mds\_recall\_state\_timeout** オプションで指定された時間 (デフォルトは 60 秒) に準拠するために時間がかかる場合に表示されます。詳細は、[Metadata Server キャッシュサイズの制限](#) のセクションを参照してください。

#### "Client name failing to advance its oldest client/flush tid"

コード: MDS\_HEALTH\_CLIENT\_OLDEST\_TID, MDS\_HEALTH\_CLIENT\_OLDEST\_TID\_MANY

クライアントと MDS サーバー間で通信するための CephFS プロトコルは、**oldest tid** というフィールドを使用して、MDS が対応するためにクライアント要求が完全に完了している MDS に通知するものです。反応しないクライアントがこのフィールドを進めない場合、MDS はクライアント要求によって使用されるリソースを適切にクリーンアップできなくなる可能性があります。このメッセージは、クライアントが **max\_completed\_requests** オプション (デフォルトは 100000) で指定された数値よりも多くのリクエストがある場合に表示されます。これは、MDS 側では完全でも、クライアントの **最も古い tid 値** について考慮されていないことを示しています。

#### "Metadata damage detected"

コード: MDS\_HEALTH\_DAMAGE

メタデータプールから読み取り時に、破損したメタデータまたは欠落しているメタデータが見つかりました。このメッセージは、MDS が動作を継続するために十分な破損した分離されたことを示しています。ただし、クライアントが破損したサブツリーへのアクセスにより I/O エラーが返されることを示します。**damage ls** administration socket コマンドを使用して、破損の詳細を表示します。このメッセージは、破損が発生するとすぐに表示されます。

#### "MDS in read-only mode"

Code: MDS\_HEALTH\_READ\_ONLY

MDS は読み取り専用モードに入力されており、メタデータの変更を試みるクライアント操作に **EROFS** エラーコードを返します。MDS は読み取り専用モードに入ります。

- メタデータプールへの書き込み中に書き込みエラーが発生した場合
- **force\_readonly** 管理ソケットコマンドを使用して、管理者が MDS を読み取り専用モードに強制するとき。

#### "<N> slow requests are blocked"

コード: MDS\_HEALTH\_SLOW\_REQUEST

1つ以上のクライアント要求が完了しておらず、MDS が非常に遅いか、バグが発生したことを示しています。**ops** 管理ソケットコマンドを使用して、未処理のメタデータ操作をリスト表示します。このメッセージは、クライアントの要求が **mds\_op\_complaint\_time** オプションで指定した値よりも時間がかかる場合に表示されます (デフォルトは 30 秒)。

#### "Too many inodes in cache"

コード: MDS\_HEALTH\_CACHE\_OVERSIZED

MDS は、管理者が設定した制限に準拠するためにキャッシュをトリミングできませんでした。MDS キャッシュが大きすぎると、デーモンは利用可能なメモリーを使い切ったり、予期せずに終了する可能性があります。デフォルトでは、MDS キャッシュサイズが制限よりも 50% を超えると、このメッセージが表示されます。

#### 関連情報

- 詳細は、[Red Hat Ceph Storage ファイルシステムガイド](#)の [Metadata Server キャッシュサイズの制限](#) セクションを参照してください。

## 付録B METADATA SERVER デーモン設定リファレンス

メタデータサーバー (MDS) デーモンの設定に使用できる、このコマンドリストを参照してください。

### mon\_force\_standby\_active

#### 説明

**true** に設定した場合は、スタンバイ再生モードの MDS を強制的にアクティブにします。Ceph 設定ファイルの **[mon]** または **[global]** セクションで設定します。

#### 型

Boolean

#### デフォルト

**true**

### max\_mds

#### 説明

クラスター作成時にアクティブな MDS デーモンの数。Ceph 設定ファイルの **[mon]** または **[global]** セクションで設定します。

#### 型

32 ビット整数

#### デフォルト

**1**

### mds\_cache\_memory\_limit

#### 説明

MDS がキャッシュに強制するメモリ制限。Red Hat は、**mds cache size** パラメーターの代わりにこのパラメーターを使用することを推奨します。

#### 型

64 ビット整数未署名

#### デフォルト

**4294967296**

### mds\_cache\_reservation

#### 説明

MDS キャッシュが維持するキャッシュ予約、メモリ、または inode。この値は、設定された最大キャッシュの割合です。MDS が予約にデップを開始したら、キャッシュサイズが縮小して予約を復元するまで、クライアントの状態をやり直します。

#### 型

浮動小数点 (Float)

#### デフォルト

**0.05**

### mds\_cache\_size

#### 説明

キャッシュする inode の数。値が 0 の場合は、無制限の数字を示します。Red Hat は、MDS キャッシュが使用するメモリー量を制限するために `mds_cache_memory_limit` を使用することを推奨します。

**型**

32 ビット整数

**デフォルト**

0

**mds\_cache\_mid****説明**

キャッシュ LRU 内の新しい項目の挿入ポイント (トップ)

**型**

浮動小数点 (Float)

**デフォルト**

0.7

**mds\_dir\_commit\_ratio****説明**

部分的な更新ではなく、Ceph が完全な更新を使用してコミットする前に、ディレクトリーの一部に誤った情報が含まれています。

**型**

浮動小数点 (Float)

**デフォルト**

0.5

**mds\_dir\_max\_commit\_size****説明**

ディレクトリー更新の最大サイズ (MB 単位)。これを上回ると Ceph がディレクトリーを小規模なトランザクションに分割します。

**型**

32 ビット整数

**デフォルト**

90

**mds\_decay\_halfife****説明**

MDS キャッシュ温度の半減期。

**型**

浮動小数点 (Float)

**デフォルト**

5

**mds\_beacon\_interval****説明**

モニターに送信されるメッセージの頻度 (秒単位)。

**型**

浮動小数点 (Float)

**デフォルト**

**4**

**mds\_beacon\_grace****説明**

Ceph が MDS **laggy** を宣言する前に acons がなく、置き換えることができる間隔。

**型**

浮動小数点 (Float)

**デフォルト**

**15**

**mds\_blacklist\_interval****説明**

OSD マップの失敗した MDS デーモンのブラックリスト期間。

**型**

浮動小数点 (Float)

**デフォルト**

**24.0\*60.0**

**mds\_session\_timeout****説明**

Ceph の機能およびリースがタイムアウトするまでのクライアントの非アクティブの間隔 (秒単位)。

**型**

浮動小数点 (Float)

**デフォルト**

**60**

**mds\_session\_autoclose****説明**

Ceph が **laggy** クライアントセッションを閉じるまでの間隔 (秒単位)。

**型**

浮動小数点 (Float)

**デフォルト**

**300**

**mds\_reconnect\_timeout****説明**

MDS の再起動時にクライアントが再接続するまで待機する間隔 (秒単位)。

**型**

浮動小数点 (Float)

デフォルト

**45**

#### **mds\_tick\_interval**

**説明**

MDS が内部周期的タスクを実行する頻度。

**型**

浮動小数点 (Float)

デフォルト

**5**

#### **mds\_dirstat\_min\_interval**

**説明**

ツリーで再帰的な統計の伝播を回避する最小間隔 (秒単位)。

**型**

浮動小数点 (Float)

デフォルト

**1**

#### **mds\_scatter\_nudge\_interval**

**説明**

ディレクトリー統計の急速な変更が反映されます。

**型**

浮動小数点 (Float)

デフォルト

**5**

#### **mds\_client\_prealloc\_inos**

**説明**

クライアントセッションごとに事前割り当てする inode 番号の数。

**型**

32 ビット整数

デフォルト

**1000**

#### **mds\_early\_reply**

**説明**

MDS により、クライアントがジャーナルにコミットする前にリクエスト結果を確認できるかどうかを決定します。

**型**

Boolean

デフォルト

**true**

#### mds\_use\_tmap

##### 説明

ディレクトリーの更新には、**trivialmap** を使用します。

##### 型

Boolean

##### デフォルト

**true**

#### mds\_default\_dir\_hash

##### 説明

ディレクトリーフラグメント間でファイルをハッシュ化するために使用する関数。

##### 型

32 ビット整数

##### デフォルト

**2**、つまり **rjenkins**

#### mds\_log

##### 説明

MDS がジャーナルメタデータの更新を行う必要がある場合は、**true** に設定します。ベンチマークのみを無効にします。

##### 型

Boolean

##### デフォルト

**true**

#### mds\_log\_skip\_corrupt\_events

##### 説明

MDS がジャーナルの再生中に破損したジャーナルイベントをスキップするかどうかを決定します。

##### 型

Boolean

##### デフォルト

**false**

#### mds\_log\_max\_events

##### 説明

Ceph がトリミングを開始する前に、ジャーナルの最大イベント。制限を無効にするには **-1** に設定します。

##### 型

32 ビット整数

##### デフォルト

**-1**

**mds\_log\_max\_segments****説明**

Ceph がトリミングを開始する前に、ジャーナルのセグメントまたはオブジェクトの最大数。制限を無効にするには **-1** に設定します。

**型**

32 ビット整数

**デフォルト**

**30**

**mds\_log\_max\_expiring****説明**

並行して期限切れになるセグメントの最大数。

**型**

32 ビット整数

**デフォルト**

**20**

**mds\_log\_eopen\_size****説明**

**EOpen** イベントにおける inode の最大数。

**型**

32 ビット整数

**デフォルト**

**100**

**mds\_bal\_sample\_interval****説明**

断片化の決定を行うとき、ディレクトリー温度のサンプル頻度を決定します。

**型**

浮動小数点 (Float)

**デフォルト**

**3**

**mds\_bal\_replicate\_threshold****説明**

Ceph がメタデータを他のノードに複製するまでの最大温度。

**型**

浮動小数点 (Float)

**デフォルト**

**8000**

**mds\_bal\_unreplicate\_threshold****説明**



Ceph が他のノードへのメタデータの複製を停止する前の最小温度。

**型**

浮動小数点 (Float)

**デフォルト**

**0**

**mds\_bal\_frag****説明**

MDS がディレクトリーをフラグメント化するかどうかを決定します。

**型**

Boolean

**デフォルト**

**false**

**mds\_bal\_split\_size****説明**

MDS がディレクトリーのフラグメントを小規模なビットに分割する前にの最大ディレクトリーサイズ。root ディレクトリーには、デフォルトのフラグメントサイズが 10000 です。

**型**

32 ビット整数

**デフォルト**

**10000**

**mds\_bal\_split\_rd****説明**

Ceph がディレクトリーのフラグメントを分割するまでの最大ディレクトリー読み取り温度。

**型**

浮動小数点 (Float)

**デフォルト**

**25000**

**mds\_bal\_split\_wr****説明**

Ceph がディレクトリーのフラグメントを分割するまでの最大ディレクトリー書き込み温度。

**型**

浮動小数点 (Float)

**デフォルト**

**10000**

**mds\_bal\_split\_bits****説明**

ディレクトリーフラグメントを分割するビット数。

**型**

32 ビット整数

デフォルト

**3**

### **mds\_bal\_merge\_size**

**説明**

Ceph が隣接ディレクトリーフラグメントをマージしようとする前の最小ディレクトリーサイズ。

**型**

32 ビット整数

デフォルト

**50**

### **mds\_bal\_merge\_rd**

**説明**

Ceph が隣接するディレクトリーフラグメントのマージ前の最小限の読み取り温度。

**型**

浮動小数点 (Float)

デフォルト

**1000**

### **mds\_bal\_merge\_wr**

**説明**

Ceph が隣接するディレクトリーのフラグメントをマージする前に最小の書き込み温度。

**型**

浮動小数点 (Float)

デフォルト

**1000**

### **mds\_bal\_interval**

**説明**

MDS ノード間のワークロード交換の頻度 (秒単位)。

**型**

32 ビット整数

デフォルト

**10**

### **mds\_bal\_fragment\_interval**

**説明**

ディレクトリーの断片化を調整する頻度 (秒単位)。

**型**

32 ビット整数

デフォルト

## 5

**mds\_bal\_idle\_threshold****説明**

Ceph がサブツリーをその親に移行する前の最小温度。

**型**

浮動小数点 (Float)

**デフォルト**

0

**mds\_bal\_max****説明**

Ceph が停止する前に balancer を実行する反復数。テストの目的でのみ使用してください。

**型**

32 ビット整数

**デフォルト**

-1

**mds\_bal\_max\_until****説明**

Ceph が停止するまでの balancer を実行する秒数。テストの目的でのみ使用してください。

**型**

32 ビット整数

**デフォルト**

-1

**mds\_bal\_mode****説明**

MDS 負荷を計算する方法:

- **1** = ハイブリッド
- **2** = リクエストレートとレイテンシー。
- **3** = CPU 負荷

**型**

32 ビット整数

**デフォルト**

0

**mds\_bal\_min\_rebalance****説明**

Ceph の移行前の最小サブツリーの温度。

**型**

浮動小数点 (Float)

デフォルト

**0.1**

### mds\_bal\_min\_start

説明

Ceph がサブツリーを検索するまでの最小サブツリーの温度。

型

浮動小数点 (Float)

デフォルト

**0.2**

### mds\_bal\_need\_min

説明

許可するターゲットサブツリーの最小分数。

型

浮動小数点 (Float)

デフォルト

**0.8**

### mds\_bal\_need\_max

説明

許可するターゲットサブツリーサイズの最大分数。

型

浮動小数点 (Float)

デフォルト

**1.2**

### mds\_bal\_midchunk

説明

Ceph は、ターゲットサブツリーサイズのこの分を超えるサブツリーを移行します。

型

浮動小数点 (Float)

デフォルト

**0.3**

### mds\_bal\_minchunk

説明

Ceph は、ターゲットサブツリーサイズのこの分よりも小さいサブツリーを無視します。

型

浮動小数点 (Float)

デフォルト

**0.001**

**mds\_bal\_target\_removal\_min****説明**

Ceph が MDS マップから古い MDS ターゲットを削除する前に、 balancer の反復回数。

**型**

32 ビット整数

**デフォルト**

5

**mds\_bal\_target\_removal\_max****説明**

Ceph が MDS マップから古い MDS ターゲットを削除するまでの balancer 反復の最大数。

**型**

32 ビット整数

**デフォルト**

10

**mds\_replay\_interval****説明**

ジャーナルは、 **hot standby** の **standby-replay** モードの場合に ポーリングする間隔です。

**型**

浮動小数点 (Float)

**デフォルト**

1

**mds\_shutdown\_check****説明**

MDS のシャットダウン中にキャッシュをポーリングする間隔。

**型**

32 ビット整数

**デフォルト**

0

**mds\_thrash\_exports****説明**

Ceph はノード間でサブツリーをランダムエクスポートします。テストの目的でのみ使用してください。

**型**

32 ビット整数

**デフォルト**

0

**mds\_thrash\_fragments****説明**

Ceph の無作為に断片化したり、ディレクトリーをマージしたりします。

**型**

32 ビット整数

**デフォルト**

**0**

**mds\_dump\_cache\_on\_map****説明**

Ceph は MDS キャッシュの内容を各 MDS マップのファイルにダンプします。

**型**

Boolean

**デフォルト**

**false**

**mds\_dump\_cache\_after\_rejoin****説明**

Ceph は、リカバリー中にキャッシュを再度参加した後に MDS キャッシュの内容をファイルにダンプします。

**型**

Boolean

**デフォルト**

**false**

**mds\_verify\_scatter****説明**

Ceph は、さまざまな scatter/gather invariants が **true** であることをアサートします。開発者向けの使用のみ。

**型**

Boolean

**デフォルト**

**false**

**mds\_debug\_scatterstat****説明**

Ceph は、バリエーション内のさまざまな再帰統計が **true** であるアサートされます。開発者向けの使用のみ。

**型**

Boolean

**デフォルト**

**false**

**mds\_debug\_frag****説明**

Ceph は、使用時にディレクトリーの断片化を変えるように検証します。開発者向けの使用のみ。

**型**

Boolean

**デフォルト**

**false**

**mds\_debug\_auth\_pins****説明**

デバッグ認証のバリエーション。開発者向けの使用のみ。

**型**

Boolean

**デフォルト**

**false**

**mds\_debug\_subtrees****説明**

サブツリーのバリエーションのデバッグ開発者向けの使用のみ。

**型**

Boolean

**デフォルト**

**false**

**mds\_kill\_mdstable\_at****説明**

Ceph は、MDS テーブルコードに MDS 障害を挿入します。開発者向けの使用のみ。

**型**

32 ビット整数

**デフォルト**

**0**

**mds\_kill\_export\_at****説明**

Ceph は、サブツリーのエクスポートコードに MDS の失敗を注入します。開発者向けの使用のみ。

**型**

32 ビット整数

**デフォルト**

**0**

**mds\_kill\_import\_at****説明**

Ceph は、サブツリーのインポートコードに MDS の失敗を注入します。開発者向けの使用のみ。

**型**

32 ビット整数

**デフォルト**

0

**mds\_kill\_link\_at****説明**

Ceph は、ハードリンクコードに MDS 障害を挿入します。開発者向けの使用のみ。

**型**

32 ビット整数

**デフォルト**

0

**mds\_kill\_rename\_at****説明**

Ceph は、名前変更コードに MDS の失敗を注入します。開発者向けの使用のみ。

**型**

32 ビット整数

**デフォルト**

0

**mds\_wipe\_sessions****説明**

Ceph は、起動時にすべてのクライアントセッションを削除します。テストの目的でのみ使用してください。

**型**

Boolean

**デフォルト**

0

**mds\_wipe\_ino\_prealloc****説明**

Ceph は、起動時に inode 事前割り当てメタデータを削除します。テストの目的でのみ使用してください。

**型**

Boolean

**デフォルト**

0

**mds\_skip\_ino****説明**

起動時にスキップする inode 番号の数。テストの目的でのみ使用してください。

**型**

32 ビット整数



**デフォルト****0****mds\_standby\_for\_name****説明**

MDS デーモンは、この設定で指定された名前の別の MDS デーモンに対するスタンバイです。

**型**

String

**デフォルト**

該当なし

**mds\_standby\_for\_rank****説明**

MDS デーモンのインスタンスは、このランクの別の MDS デーモンインスタンスに対するスタンバイです。

**型**

32 ビット整数

**デフォルト****-1****mds\_standby\_replay****説明**

MDS デーモンが **hot standby** として使用する場合にアクティブな MDS のログをポーリングおよび再生するかどうかを決定します。

**型**

Boolean

**デフォルト****false**

## 付録C ジャーナル設定の参照

ジャーナルヤー設定に使用できる list コマンドのリファレンス

### journaler\_write\_head\_interval

**説明**

ジャーナルヘッドオブジェクトを更新する頻度。

**型**

整数

**必須**

いいえ

**デフォルト**

15

### journaler\_prefetch\_periods

**説明**

ジャーナル再生に先行するストライプ期間の数。

**型**

整数

**必須**

いいえ

**デフォルト**

10

### journaler\_prezero\_periods

**説明**

書き込み位置が 0 より進んだストライプ期間の数。

**型**

整数

**必須**

いいえ

**デフォルト**

10

### journaler\_batch\_interval

**説明**

人為的に発生する最大レイテンシー (秒単位)。

**型**

Double

**必須**

いいえ

**デフォルト**

.001

---

**journaler\_batch\_max****説明**

フラッシュを遅延させる最大バイト。

**型**

64 ビット未署名の整数

**必須**

いいえ

**デフォルト**

**0**

## 付録D CEPH FILE SYSTEM ミラー設定リファレンス

このセクションでは、Ceph ファイルシステム (CephFS) ミラーの設定オプションを一覧表示します。

### **cephfs\_mirror\_max\_concurrent\_directory\_syncs**

#### 説明

**cephfs-mirror** デーモンが同時に同期できるディレクトリースナップショットの最大数。同期スレッドの数を制御します。

#### 型

整数

#### デフォルト

3

#### Min

1

### **cephfs\_mirror\_action\_update\_interval**

#### 説明

保留中のミラー更新アクションを処理する間隔 (秒単位)。

#### 型

**secs**

#### デフォルト

2

#### Min

1

### **cephfs\_mirror\_restart\_mirror\_on\_blocklist\_interval**

#### 説明

ブロックリストに登録されたミラーインスタンスを再起動する間隔 (秒単位)。ゼロ (0) に設定すると、ブロックリストに登録されたインスタンスの再起動が無効になります。

#### 型

**secs**

#### デフォルト

30

#### Min

0

### **cephfs\_mirror\_max\_snapshot\_sync\_per\_cycle**

#### 説明

ディレクトリーがワーカースレッドによってミラーリングのために選択されたときに、ミラーリングするスナップショットの最大数。

#### 型

整数

#### デフォルト

3

Min

1

### **cephfs\_mirror\_directory\_scan\_interval**

説明

スナップショットミラーリング用に設定されたディレクトリーをスキャンする間隔 (秒単位)。

型

整数

デフォルト

10

Min

1

### **cephfs\_mirror\_max\_consecutive\_failures\_per\_directory**

説明

ディレクトリーを失敗としてマークする連続したスナップショット同期の失敗の数。失敗したディレクトリーは、同期のために再試行される頻度が低くなります。

型

整数

デフォルト

10

Min

0

### **cephfs\_mirror\_retry\_failed\_directories\_interval**

説明

失敗したディレクトリーの同期を再試行する間隔 (秒単位)。

型

整数

デフォルト

60

Min

1

### **cephfs\_mirror\_restart\_mirror\_on\_failure\_interval**

説明

失敗したミラーインスタンスを再起動する間隔 (秒単位)。ゼロ (0) に設定すると、失敗したミラーインスタンスの再起動が無効になります。

型

secs

デフォルト

20

Min

0

**cephfs\_mirror\_mount\_timeout****説明**

**cephfs-mirror** デーモンによるプライマリまたはセカンダリ CephFS のマウントのタイムアウト (秒単位)。これをより高い値に設定すると、クラスターに到達できない場合にファイルシステムをマウントするときに、ミラーデーモンが停止する可能性があります。このオプションは、通常の **client\_mount\_timeout** をオーバーライドするために使用されます。

**型****secs****デフォルト****10****Min****0**