



# Red Hat Ceph Storage 4

## 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 4 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
1.5. 関連情報	8
<b>第2章 CEPH FILE SYSTEM METADATA SERVER</b> .....	<b>10</b>
2.1. 前提条件	10
2.2. METADATA SERVER デーモンの状態	10
2.3. メタデータサーバーのランク	10
2.4. メタデータサーバーのキャッシュサイズ制限	11
2.5. 複数のアクティブな METADATA SERVER デーモンの設定	12
2.6. スタンバイデーモンの数の設定	13
2.7. STANDBY-REPLAY 用 METADATA SERVER の設定	14
2.8. アクティブな METADATA SERVER デーモンの数を減らす方法	14
2.9. 関連情報	16
<b>第3章 CEPH FILE SYSTEM のデプロイメント</b> .....	<b>17</b>
3.1. 前提条件	17
3.2. 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	17
3.3. CEPH ファイルシステムの作成	18
3.4. イレイジャーコーディングを使用した CEPH ファイルシステムの作成 (テクノロジープレビュー)	20
3.5. CEPH ファイルシステム用のクライアントユーザーの作成	24
3.6. CEPH FILE SYSTEM のカーネルクライアントとしてのマウント	26
3.7. CEPH FILE SYSTEM の FUSE クライアントとしてのマウント	29
3.8. 関連情報	32
<b>第4章 CEPH FILE SYSTEM 管理</b> .....	<b>34</b>
4.1. 前提条件	34
4.2. 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	34
4.3. 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	35
4.4. 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	35
4.5. 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	37
4.6. データプールの追加	37
4.7. CEPH FILE SYSTEM クォータの使用	39
4.8. ファイルとディレクトリーのレイアウトでの作業	42
4.9. CEPH FILE SYSTEM スナップショットに関する留意事項	46
4.10. CEPH FILE SYSTEM スナップショットの管理	48
4.11. CEPH FILE SYSTEM クラスターの停止	52
4.12. コマンドラインインターフェイスを使用した CEPH FILE SYSTEM の削除	53
4.13. ANSIBLE を使用した CEPH FILE SYSTEM の削除	57
4.14. 最小クライアントバージョンの設定	61
4.15. CEPH MDS FAIL コマンドの使用	62
4.16. CEPH FILE SYSTEM クライアントのエビクション	63
4.17. ブラックリスト CEPH FILE SYSTEM クライアント	63
4.18. CEPH FILE SYSTEM クライアントの手動エビクト	64
4.19. ブラックリストからの CEPH FILE SYSTEM クライアントの削除	65
4.20. 関連情報	66
<b>第5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>67</b>
5.1. CEPH FILE SYSTEM ボリューム	67

5.2. CEPH FILE SYSTEM サブボリューム	69
5.3. CEPH FILE SYSTEM サブボリュームグループ	84
5.4. 関連情報	89
<b>付録A CEPH FILE SYSTEM のヘルスメッセージ</b> .....	<b>90</b>
<b>付録B METADATA SERVER デーモン設定リファレンス</b> .....	<b>93</b>
<b>付録C ジャーナル設定の参照</b> .....	<b>108</b>
<b>付録D CEPH FILE SYSTEM クライアント設定の参照</b> .....	<b>110</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。

### 高可用性

Ceph File System には、Ceph Metadata Server (MDS) のクラスター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。

### サイズ変更

Ceph File System サイズは、そのデータプールにサービスを提供する OSD の容量でのみバインドされます。容量を増やすには、CephFS データプールに OSD をさらに追加します。容量を減らすには、クライアントクォータまたはプールクォータのいずれかを使用します。

### Snapshots

Ceph File System は読み取り専用のスナップショットをサポートしますが、書き込み可能なクローンはサポートしません。

### POSIX ファイルシステムの操作



Ceph File System は、以下のアクセスパターンを含む、標準かつ一貫性のある POSIX ファイルシステム操作をサポートします。

- Linux ページキャッシュによるバッファー書き込み操作。
- Linux ページキャッシュによるキャッシュ読み取り操作。
- ページキャッシュをバイパスするダイレクト I/O 非同期または同期読み取り/書き込み操作。
- メモリーマップされた I/O。

## 関連情報

- Ceph メタデータサーバーをインストールするには、[Installation Guide](#)の [Installing Metadata servers](#) セクションを参照してください。
- Ceph ファイルシステムを作成するには、[File System Guide](#)の [Deploying Ceph File Systems](#) 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

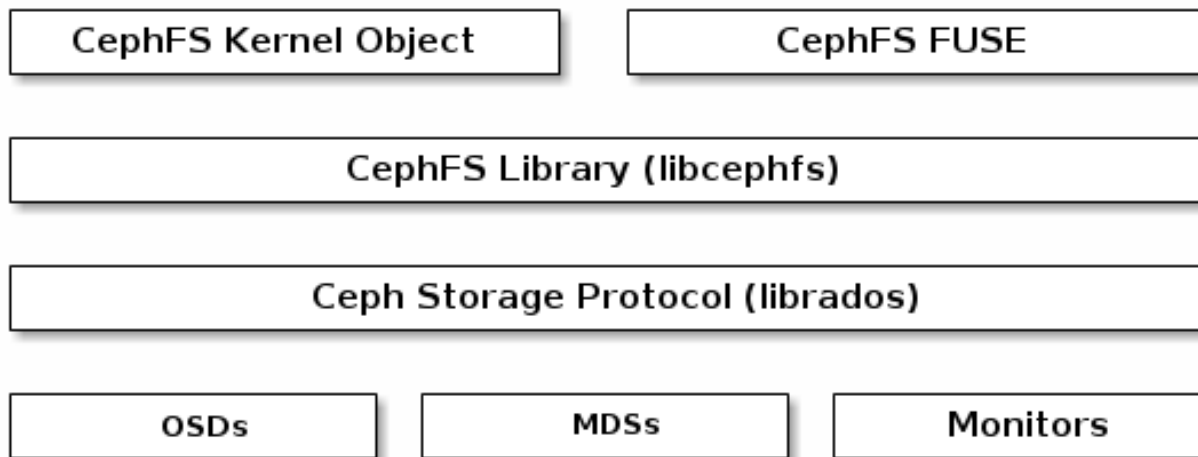
CephFS クライアントは、FUSE クライアントの **ceph-fuse** やカーネルクライアントの **kcephfs** など、CephFS を使用するアプリケーションの代わりに I/O 操作を行います。CephFS クライアントは、アクティブな Metadata Server にメタデータの要求を送信します。その代わりに、CephFS クライアントはファイルのメタデータ認識し、メタデータとファイルデータの両方を安全にキャッシュすることができます。

### 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 クラスター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。



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。

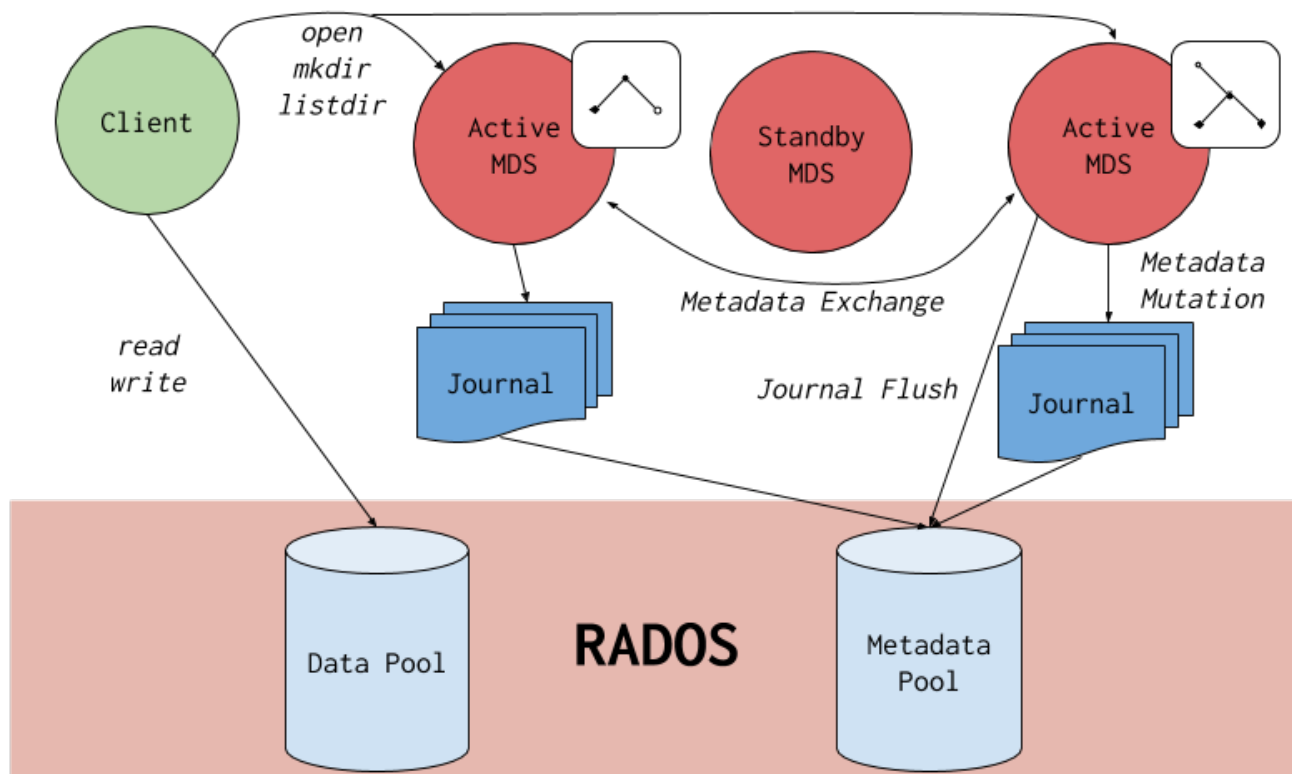
- Ceph OSD (**ceph-osd**) Ceph File System のデータとメタデータが格納されています。
- Ceph File System のメタデータを管理する Ceph Metadata Servers (**ceph-mds**)。
- クラスターマップのマスターコピーを管理する Ceph Monitors (**ceph-mon**)。

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。



#### 関連情報

- Ceph Metadata サーバーをインストールするには、[Red Hat Ceph Storage Installation Guide](#) の [Installing Metadata servers](#) セクションを参照してください。
- Ceph File System を作成するには、[Red Hat Ceph Storage File System Guide](#) の [Deploying Ceph File Systems](#) 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。

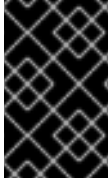
#### 関連情報

- 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

1つの Red Hat Ceph Storage クラスターでの複数の Ceph File Systems の作成はデフォルトで無効にされています。Ceph File System の追加作成を試みると、以下のエラーメッセージが出て失敗します。

```
Error EINVAL: Creation of multiple filesystems is disabled.
```



## 重要

技術的には可能ですが、Red Hat では複数の Ceph File Systems を1つの Red Hat Ceph Storage クラスタ上に提供していません。これを実行すると、MDS または CephFS クライアントノードが予期せずに終了する可能性があります。

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、
- 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す。

## 関連情報

- Ceph Metadata サーバーをインストールするには、Red Hat Ceph Storage Installation Guide の [Installing Metadata servers](#) セクションを参照してください。
- Ceph File System を作成するには、Red Hat Ceph Storage File System Guide の [Deploying Ceph File Systems](#) セクションを参照してください。

## 1.5. 関連情報

- 詳細は、Red Hat Ceph Storage Installation Guide の [Installing Metadata Servers](#) を参照してください。

- 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する方法については、**Deploying the Shared File Systems service with CephFS through NFS**の [CephFS with NFS-Ganesha deployment](#) 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。

### 2.1. 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。

### 2.2. 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。

### 2.3. 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

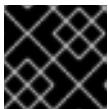
ランクには、以下の状態があります。

- **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。

## 2.4. 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。



### 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。

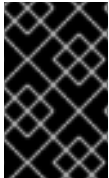
キャッシュの制限はハードな制限ではないことに注意してください。CephFS クライアントや MDS のバグ、または誤動作するアプリケーションが原因で、MDS のキャッシュサイズが超過する可能性があります。**mds\_health\_cache\_threshold** オプションは、ストレージクラスターの健全性に関する警告メッセージを設定し、データシートがキャッシュを縮小できない原因をオペレーターが調査できるようにします。

## 関連情報

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

## 2.5. 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。



### 重要

スクラブプロセスは、アクティブな MDS デーモンが複数設定されている場合は、現時点でサポートされていません。

## 前提条件

- MDS ノードでの Ceph 管理機能。

## 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 |
+-----+-----+-----+-----+-----+
| 0 | active | node1 | Reqs: 0/s | 10 | 12 |
| 1 | active | node2 | Reqs: 0/s | 10 | 12 |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
| 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 [Decreasing the Number of Active MDS Daemons](#) 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。

## 2.6. 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 ノードへのユーザーアクセス。

### 手順

1. 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。

## 2.7. 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 ノードへのユーザーアクセス。

### 手順

1. 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。



### 注記

ブール値 **allow\_standby\_replay** を **0** に戻すと、新しい standby-replay デーモンが割り当てられなくなるだけです。実行中のデーモンも停止するには、**ceph mds fail** コマンドで **failed** とマークします。

### 関連情報

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

## 2.8. 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。

## 手順

- 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 |
+-----+-----+-----+-----+-----+
| 0 | active | node1 | Reqs: 0/s | 10 | 12 |
| 1 | active | node2 | Reqs: 0/s | 10 | 12 |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
| Pool | type | used | avail |
+-----+-----+-----+-----+
| cephfs_metadata | metadata | 4638 | 26.7G |
| cephfs_data | data | 0 | 26.7G |
+-----+-----+-----+-----+

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

- 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
```

- 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 |
+-----+-----+-----+-----+-----+-----+
| 0 | active | node1 | Reqs: 0/s | 10 | 12 |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
| 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 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。

## 2.9. 関連情報

- 詳細は、Red Hat Ceph Storage Installation Guideの [Installing Metadata servers](#) 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。

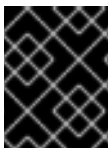
1. Ceph Monitor ノードに Ceph File System を作成します。
2. 適切な機能を持つ Ceph クライアントユーザーを作成し、Ceph File System がマウントされるノードでクライアントキーを利用できるようにします。
3. カーネルクライアントまたは File System in User Space (FUSE) クライアントで使用して、専用のノードに CephFS をマウントします。

### 3.1. 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

### 3.2. 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ービリティフラグは、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 Guideの Creating client users for a Ceph File System](#) セクションを参照してください。

## 3.3. CEPH ファイルシステムの作成

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



### 重要

デフォルトでは、Ceph Storage クラスターごとに1つの CephFS のみを作成できます。

## 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。

## 手順

1. プールを2つ作成します。1つはデータの保存用で、もう1つはメタデータの保存用です。

## 構文

```
ceph osd pool create NAME _PG_NUM
```

## 例

```
[root@mon ~]# ceph osd pool create cephfs_data 64
[root@mon ~]# 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。64 PG から 512 PG に推奨されるメタデータプールサイズの範囲。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。

2. CephFS を作成します。

## 構文

```
ceph fs new NAME METADATA_POOL DATA_POOL
```

## 例

```
[root@mon ~]# ceph fs new cephfs cephfs_metadata cephfs_data
```

3. 設定に応じて、1つ以上の MDS が active の状態に入力されていることを確認します。

## 構文

```
ceph fs status NAME
```

## 例

```
[root@mon ~]# ceph fs status cephfs
cephfs - 0 clients
=====
+-----+-----+-----+-----+-----+-----+
| Rank | State | MDS | Activity | dns | inos |
+-----+-----+-----+-----+-----+
| 0 | active | node1 | Reqs: 0/s | 10 | 12 |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

```

| Pool | type | used | avail |
+-----+-----+-----+-----+
| cephfs_metadata | metadata | 4638 | 26.7G |
| cephfs_data | data | 0 | 26.7G |
+-----+-----+-----+-----+

+-----+
| Standby MDS |
+-----+

| node3 |
| node2 |
+-----+-----

```

## 関連情報

- 詳細は、Red Hat Ceph Storage インストールガイドの [Red Hat Ceph Storage リポジトリの有効化](#) 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。
- Ceph File System 制限に関する詳しい情報は、Red Hat Ceph Storage File System Guideの [The Ceph File System](#) セクションを参照してください。
- Red Hat Ceph Storage 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。
- 詳細は、Red Hat Ceph Storage Installation Guideの [Installing Metadata Servers](#) を参照してください。

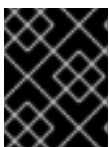
## 3.4. イレイジャーコーディングを使用した CEPH ファイルシステムの作成 (テクノロジープレビュー)

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、イレイジャーコーディングの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

イレイジャーコーディングプールを使用する Ceph File System は、テクノロジープレビュー機能です。テクノロジープレビュー機能は、Red Hat の実稼働環境でのサービスレベルアグリーメント (SLA) ではサポートされていないため、Red Hat では実稼働環境での使用を推奨していません。テクノロジープレビューの機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。詳細は、[Red Hat テクノロジープレビュー機能のサポート範囲](#) を参照してください。



### 重要

実稼働環境では、Red Hat は、複製されたプールをデフォルトのデータプールとして使用することを推奨します。



## 前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- CephFS 環境が実行中である。
- BlueStore OSD を使用するプール。
- Ceph Monitor ノードへのユーザーレベルのアクセス。

## 手順

1. CephFS メタデータ用にレプリケートされたメタデータプールを作成します。

### 構文

```
ceph osd pool create METADATA_POOL PG_NUM
```

### 例

```
[root@mon ~]# ceph osd pool create cephfs-metadata 64
```

この例では、64 個の配置グループを持つ **cephfs-metadata** という名前のプールを作成します。

2. CephFS のデフォルトの複製データプールを作成します。

### 構文

```
ceph osd pool create DATA_POOL PG_NUM
```

### 例

```
[root@mon ~]# ceph osd pool create cephfs-data 64
```

この例では、64 個の配置グループを持つ **cephfs-metadata** という名前のレプリケートされたプールを作成します。

3. 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 PG_NUM erasure
```

### 例

```
[root@mon ~]# ceph osd pool create cephfs-data-ec 64 erasure
```

この例では、64 個の配置グループを持つ **cephfs-metadata** という名前のイレイジャーコーディングプールを作成します。

4. 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 allow_ec_overwrites true
```

**例**

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

この例では、**cephfs-data-ec** という名前のイレイジャーコーディングプールで上書きを有効にします。

5. イレイジャーコーディングされたデータプールを CephFS Metadata Server (MDS) に追加します。

**構文**

```
ceph fs add_data_pool cephfs-ec DATA_POOL
```

**例**

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

- a. 必要に応じて、データプールが追加されたことを確認します。

```
[root@mon ~]# ceph fs ls
```

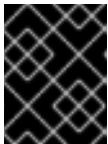
6. CephFS を作成します。

**構文**

```
ceph fs new cephfs METADATA_POOL DATA_POOL
```

**例**

```
[root@mon ~]# ceph fs new cephfs cephfs-metadata cephfs-data
```

**重要**

デフォルトのデータプールにイレイジャーコーディングプールを使用することは推奨されていません。

7. イレイジャーコーディングを使用して CephFS を作成します。

**構文**

```
ceph fs new cephfs-ec METADATA_POOL DATA_POOL
```

**例**

```
[root@mon ~]# ceph fs new cephfs-ec cephfs-metadata cephfs-data-ec
```

8. Ceph FS Metadata Servers (MDS) がアクティブな状態になっていることを確認します。

### 構文

```
ceph fs status FS_EC
```

### 例

```
[root@mon ~]# ceph fs status cephfs-ec
cephfs-ec - 0 clients
=====
+-----+-----+-----+-----+-----+-----+
| Rank | State | MDS | Activity | dns | inos |
+-----+-----+-----+-----+-----+-----+
| 0 | active | node1 | Reqs: 0/s | 10 | 12 |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
| Pool | type | used | avail |
+-----+-----+-----+-----+
| cephfs-metadata | metadata | 4638 | 26.7G |
| cephfs-data | data | 0 | 26.7G |
| cephfs-data-ec | data | 0 | 26.7G |
+-----+-----+-----+-----+

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

9. 新しいイレイジャーコーディングされたデータプールを既存のファイルシステムに追加します。
- 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 PG_NUM erasure
```

### 例

```
[root@mon ~]# ceph osd pool create cephfs-data-ec1 64 erasure
```

- 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 allow_ec_overwrites true
```

### 例

```
[root@mon ~]# ceph osd pool set cephfs-data-ec1 allow_ec_overwrites true
```

- c. イレイジャーコーディングされたデータプールを CephFS Metadata Server (MDS) に追加します。

#### 構文

```
ceph fs add_data_pool cephfs-ec DATA_POOL
```

#### 例

```
[root@mon ~]# ceph fs add_data_pool cephfs-ec cephfs-data-ec1
```

10. イレイジャーコーディングを使用して CephFS を作成します。

#### 構文

```
ceph fs new cephfs-ec METADATA_POOL DATA_POOL
```

#### 例

```
[root@mon ~]# ceph fs new cephfs-ec cephfs-metadata cephfs-data-ec1
```

#### 関連情報

- CephFS MDS に関する詳しい情報は、[Red Hat Ceph Storage File System Guide](#)の [The Ceph File System Metadata Server](#) の章を参照してください。
- CephFS のインストールに関する詳細は、[Red Hat Ceph Storage Installation Guide](#)の [Installing Metadata Servers](#) セクションを参照してください。
- 詳細は、[Red Hat Ceph Storage Storage Strategies Guide](#)の [Erasure-Coded Pools](#) セクションを参照してください。
- 詳細は、[Red Hat Ceph Storage Storage Strategies Guide](#)の [Erasure Coding with Overwrites](#) セクションを参照してください。

### 3.5. 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。

#### 手順

1. 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] ...
```

- 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

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

client.1
key: AQBSdFhcGZFUDRAAcKhG9Cl2HPiDMMRv4DC43A==
caps: [mds] allow r, allow rw path=/temp
caps: [mon] allow r
caps: [osd] allow rw tag cephfs data=cephfs_a
```

- 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

```
[root@mon ~]# 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。

2. 作成したキーを確認します。

### 構文

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

### 例

```
[root@mon ~]# ceph auth get client.1
```

3. 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

```
[root@mon ~]# ceph auth get client.1 -o ceph.client.1.keyring
exported keyring for client.1
```

- 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 root@MONITOR_NODE_NAME:/root/ceph.client.1.keyring /etc/ceph/
```

Replace `MONITOR_NODE_NAME` を Ceph Monitor ノード名または IP に置き換えます。

### 例

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

- 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 KEYRING
```

### 例

```
[root@client ~]# 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 [ユーザー管理](#) の章を参照してください。

## 3.6. 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 ノードへのユーザーレベルのアクセス。
- 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 4 Tools リポジトリを有効にします。

### Red Hat Enterprise Linux 7

```
[root@client ~]# subscription-manager repos --enable=rhel-7-server-rhceph-4-tools-rpms
```

### Red Hat Enterprise Linux 8

```
[root@client ~]# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

- b. **ceph-common** パッケージをインストールします。

### Red Hat Enterprise Linux 7

```
[root@client ~]# yum install ceph-common
```

### Red Hat Enterprise Linux 8

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

- 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. 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@client]# 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
```

### 例

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



### 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。

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@client ~]# stat -f /mnt/cephfs
```

## 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 ファイルシステムガイドの [Creating a Ceph File System](#) セクションを参照してください。

## 3.7. 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 ノードへのユーザーレベルのアクセス。
- 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 4 Tools リポジトリを有効にします。

### Red Hat Enterprise Linux 7

```
[root@client ~]# subscription-manager repos --enable=rhel-7-server-rhceph-4-tools-rpms
```

### Red Hat Enterprise Linux 8

```
[root@client ~]# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

- b. **ceph-fuse** パッケージをインストールします。

### Red Hat Enterprise Linux 7

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

### Red Hat Enterprise Linux 8

```
[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
```

- f. [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@client ~]# 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。

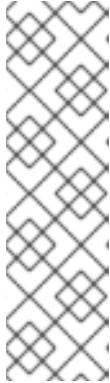
3. **ceph-fuse** ユーティリティーを使用して Ceph ファイルシステムをマウントします。

## 構文

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

## 例

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

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

```
[user@client ~]$ 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ーを作成します。

## 構文

```
mkdir PATH_TO_MOUNT_POINT
```

## 例

```
[root@client ~]# 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_:/ [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 番目のコラムは、それぞれ **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 fuse.ceph ceph.id=1, 0 0
mon2:6789,
ceph.client_mountpoint=/my_vol/my_sub_vol_group/my_sub_vol/0,
mon3:6789:/ _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 ファイルシステムガイドの [Creating a Ceph File System](#) セクションを参照してください。

## 3.8. 関連情報

- 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 のインストールに関する詳細は、[Red Hat Ceph Storage Installation Guide](#) 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 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。

- 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 のランクからディレクトリーの関連付けを解除するには、「[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。
- ファイルとディレクトリーのレイアウトを使用する際には、「[ファイルとディレクトリーのレイアウトでの作業](#)」を参照してください。
- 新しいデータプールを追加するには、「[データプールの追加](#)」を参照してください。
- クォータを使用するには、「[Ceph File System クォータの使用](#)」を参照してください。
- コマンドラインインターフェイスを使用して Ceph File System を削除するには、「[コマンドラインインターフェイスを使用した Ceph File System の削除](#)」を参照してください。
- Ansible を使用して Ceph File System を削除するには、「[Ansible を使用した Ceph File System の削除](#)」を参照してください。
- 最小クライアントのバージョンを設定するには、「[最小クライアントバージョンの設定](#)」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。

### 4.1. 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 File System を作成し、マウントします。

### 4.2. 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

1. 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

## 4.3. 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

1. 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

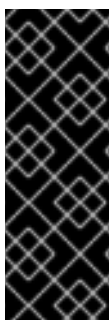
```
[root@client ~]# fusermount -u /mnt/cephfs
```

### 関連情報

- **ceph-fuse(8)** man ページ

## 4.4. 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: AQBSdFhcGZFUDRAAAcKhG9CI2HPiDMMRv4DC43A==  
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 ~]$ setfattr -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** フラグの詳細については、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 Guideの [Disassociating directory trees from Metadata Server daemon ranks](#) 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。



## 4.5. 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

1. 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

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

### 例

```
[user@client ~]$ setfattr -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 ファイルシステムガイドの [ディレクトリーツリーから MDS Ranks へのマッピング](#) セクションを参照してください。

## 4.6. 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。

## 別添条件

- 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 PG_NUMBER
```

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。
- **PG\_NUMBER** は配置グループ (PG) の数に置き換えます。

### 例

```
[root@mon ~]# ceph osd pool create cephfs_data_ssd 64  
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。

```
[root@mon ~]# 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

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

4. **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、[Working with File and Directory Layouts](#) を参照してください。

- 詳細は、[Creating Ceph File System Client Users](#)を参照してください。

## 4.7. CEPH FILE SYSTEM クォータの使用

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。

### 4.7.1. 前提条件

- `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。

### 4.7.2. 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。

#### 制限事項

- 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。
- 以前のバージョンでは、クォータはユーザー空間 FUSE クライアントでのみサポートされていました。Linux カーネルバージョン 4.17 以降では、CephFS カーネルクライアントは Ceph mimc またはそれ以降のクラスターに対するクォータに対応します。これらのバージョン要件は、Red Hat Enterprise Linux 8 と Red Hat Ceph Storage 4 がそれぞれ対応しています。ユーザー空間 FUSE クライアントは、古い OS およびクラスターのバージョンで使用できます。FUSE クライアントは `ceph-fuse` パッケージで提供されます。
- 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。

### 4.7.3. 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

- `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@fs ~]# getfattr -n ceph.quota.max_bytes /cephfs/
```

- 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@fs ~]# getfattr -n ceph.quota.max_files /cephfs/
```

## 関連情報

- 詳細は、**getfattr(1)** man ページを参照してください。

## 4.7.4. クォータの設定

本セクションでは、`setfattr` 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

- `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 クォータを設定します。

- a. バイト制限クォータの使用:

### 構文

```
setfattr -n ceph.quota.max_bytes -v 100000000 /some/dir
```

### 例

```
[root@fs ~]# setfattr -n ceph.quota.max_bytes -v 100000000 /cephfs/
```

この例では、100000000 バイトは 100 MB となります。

- b. ファイル制限クォータの使用:

### 構文

```
setfattr -n ceph.quota.max_files -v 10000 /some/dir
```

### 例

```
[root@fs ~]# setfattr -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。

## 関連情報

- 詳細は、**setfattr(1)** man ページを参照してください。

### 4.7.5. クォータの削除

本セクションでは、**setfattr** 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。

#### 前提条件

- **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@fs ~]# setfattr -n ceph.quota.max_bytes -v 0 /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@fs ~]# setfattr -n ceph.quota.max_files -v 0 /cephfs/
```

### 関連情報

- 詳細は、**setfattr(1)** man ページを参照してください。

### 4.7.6. 関連情報

- 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。

## 4.8. ファイルとディレクトリーのレイアウトでの作業

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](#)

### 4.8.1. 前提条件

- **attr** パッケージのインストール

### 4.8.2. 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ピング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。

[File and Directory Layout Fields](#)の表には、ファイルやディレクトリーに設定できるレイアウトフィールドが記載されています。

## 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。

## 関連情報

- 詳細は、[Layouts Inheritance](#) を参照してください。

### 4.8.3. 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

1. 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@fs ~]# setfattr -n ceph.file.layout.stripe_unit -v 1048576 test
```

## 関連情報

- 詳細は、[ファイルとディレクトリーのレイアウト](#) のファイルシステムガイド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。

#### 4.8.4. 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. 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 ファイルシステムガイドの [ファイルおよびディレクトリーのレイアウトの設定](#) セクションを参照してください。

#### 4.8.5. 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

1. 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

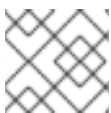
**pool** フィールドのプールは、名前で示されます。ただし、新規作成されたプールは ID で識別できます。

## 関連情報

- **getfattr(1)** man ページ
- 詳細は、[File and directory layout fields](#) を参照してください。

### 4.8.6. 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

## 4.9. CEPH FILE SYSTEM スナップショットに関する留意事項

ストレージ管理者として、Ceph File System (CephFS) スナップショットを管理するためのデータ構造、システムコンポーネント、および留意事項を理解できます。

スナップショットは、作成時にファイルシステムのイミュータブルなビューを作成します。スナップショットはディレクトリー内に作成でき、そのディレクトリー下のファイルシステムにあるすべてのデータがカバーされます。

### 4.9.1. Ceph File System のスナップショットメタデータの保存

スナップショットディレクトリーエントリーとその inode のストレージは、スナップショットの時点でそれらがあったディレクトリーの一部としてインラインで実行されます。すべてのディレクトリーエントリーには、有効な最初と最後の **snapid** が含まれています。

### 4.9.2. Ceph File System スナップショットのライトバック

Ceph スナップショットは、どの操作がスナップショットに適用されるかを決定し、スナップショットデータとメタデータを OSD および MDS クラスタにフラッシュバックするために、クライアントに依存します。スナップショットはファイル階層のサブツリーに適用され、スナップショットの作成はいつでも発生する可能性があるため、スナップショットのライトバックの処理は複雑なプロセスです。

同じスナップショットセットに属するファイル階層の一部は、単一の **SnapRealm** によって参照されます。各スナップショットは、ディレクトリーの下にネストされたサブディレクトリーに適用され、ファイル階層を複数のレルムに分割します。レルムに含まれるすべてのファイルは、同じスナップショットのセットを共有します。

Ceph Metadata Server (MDS) は、各 inode の機能 (caps) を発行して、inode のメタデータおよびファイルデータへのクライアントアクセスを制御します。スナップショットの作成中に、クライアントは、その時点でのファイルの状態を記述する機能を備えた inode 上のダーティメタデータを取得します。クライアントが **ClientSnap** メッセージを受信すると、ローカルの **SnapRealm** と特定の inode へのリンクが更新され、inode の **CapSnap** が生成されます。機能のライトバックは **CapSnap** をフラッシュし、ダーティデータが存在する場合は、**CapSnap** を使用して、スナップショットが OSD にフラッシュされるまで新規データの書き込みをブロックします。

MDS は、それらをフラッシュするためのルーチンプロセスの一部として、スナップショットを表すディレクトリーエントリーを生成します。MDS は、ライトバックプロセスがそれらをフラッシュするまで、メモリーとジャーナルに固定された未処理の **CapSnap** データを含むディレクトリーエントリーを保持します。

#### 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 Guide](#)の [Creating client users for a Ceph File System](#) セクションを参照してください。

#### 4.9.3. Ceph File System のスナップショットとハードリンク

Ceph は、複数のハードリンクを持つ inode をダミーのグローバル **SnapRealm** に移動します。このダミー **SnapRealm** は、ファイルシステム内のすべてのスナップショットに対応します。新しいスナップショットは inode のデータを保持します。この保存されたデータは、inode の任意のリンケージのスナップショットをカバーします。

#### 4.9.4. Ceph File System のスナップショットの作成

スナップショットを更新するプロセスは、スナップショットを削除するプロセスと似ています。

親 **SnapRealm** から inode を削除すると、**SnapRealm** がまだ存在しない場合、Ceph は名前が変更された inode の新しい **SnapRealm** を生成します。Ceph は、元の親 **SnapRealm** で有効なスナップショットの ID を、新しい **SnapRealm** の **past\_parent\_snaps** データ構造に保存してから、スナップショットの作成と同様のプロセスに従います。

#### 関連情報

- スナップショットのデータ構造について、詳細は [Red Hat Ceph Storage File System Guide](#) の [Ceph File System snapshot data structures](#) を参照してください。

#### 4.9.5. Ceph File System のスナップショットと複数のファイルシステム

スナップショットは、複数のファイルシステムでは正常に機能しないことが知られています。

名前空間を持つ単一の Ceph プールを共有する複数のファイルシステムがある場合、それらのスナップショットは競合し、1つのスナップショットを削除すると、同じ Ceph プールを共有する他のスナップショットのファイルデータが失われます。

#### 4.9.6. Ceph File System の\sナップショットのデータ構造

Ceph File System (CephFS) は、以下のスナップショットデータ構造を使用してデータを効率的に保存します。

##### SnapRealm

**SnapRealm** は、ファイル階層の新しいポイントでスナップショットを作成するとき、またはスナッ

プショットされた inode を親スナップショットの外に移動するときに作成されます。単一の **SnapRealm** は、同じスナップショットのセットに属するファイル階層の一部を表します。 **SnapRealm** には、スナップショットの一部である **sr\_t\_srnode** と **inodes\_with\_caps** が含まれます。

### sr\_t

**sr\_t** はディスク上のスナップショットメタデータです。これには、シーケンスカウンター、タイムスタンプ、および関連するスナップショット ID と **past\_parent\_snaps** の一覧が含まれます。

### SnapServer

**SnapServer** は、スナップショット ID の割り当て、スナップショットの削除、およびファイルシステムの累積スナップショット一覧の維持を管理します。ファイルシステムには、**SnapServer** のインスタンスが1つだけ含まれます。

### SnapContext

**SnapContext** はスナップショットシーケンス ID (snapid) と、オブジェクトに現在定義されているすべてのスナップショット ID で設定されます。書き込み操作が発生すると、Ceph クライアントは **SnapContext** を提供し、オブジェクトに存在するスナップショットのセットを指定します。 **SnapContext** 一覧を生成するには、Ceph は **SnapRealm** に関連付けられた snapid と、 **past\_parent\_snaps** データ構造の有効な snapid を組み合わせます。ファイルデータは RADOS の自己管理のスナップショットを使用して保存されます。自己管理されたスナップショットでは、クライアントは書き込みごとに現在の **SnapContext** を提供する必要があります。クライアントは、Ceph OSD にファイルデータを書き込む際に、正しい **SnapContext** を慎重に使用します。

**SnapClient** がキャッシュした有効なスナップショットは、古い snapid を除外します。

### SnapClient

**SnapClient** は **SnapServer** と通信し、累積スナップショットをローカルでキャッシュするために使用されます。各メタデータサーバー (MDS) のランクには、**SnapClient** インスタンスがあります。

## 4.10. CEPH FILE SYSTEM スナップショットの管理

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。

### 4.10.1. 前提条件

- 実行中で正常な Red Hat Ceph Storage クラスタ
- Ceph File System のデプロイメント

### 4.10.2. Ceph ファイルシステムのスナップショット

Ceph File System (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。

### 4.10.3. Ceph File System のスナップショットの有効化

新しい Ceph File System はデフォルトでスナップショット機能を有効にしますが、既存の Ceph File Systems ではこの機能を手動で有効にする必要があります。

### 前提条件

- 実行中で正常な 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

- 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

```
[root@mds ~]# ceph fs set cephfs allow_new_snaps true  
enabled new snapshots
```

### 関連情報

- スナップショットの作成について、詳細は [Red Hat Ceph Storage File System Guide の Creating a snapshot for a Ceph File System](#) セクションを参照してください。
- スナップショットの削除について、詳細は [Red Hat Ceph Storage File System Guide の Deleting a snapshot for a Ceph File System](#) セクションを参照してください。
- スナップショットの復元について、詳細は [Red Hat Ceph Storage File System Guide の Restoring a snapshot for a Ceph File System](#) セクションを参照してください。

### 4.10.4. Ceph ファイルシステムのスナップショットの作成

Ceph File System 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。スナップショットは、ディレクトリーにある非表示ディレクトリーを使用して、スナップショットを作成します。このディレクトリーの名前は、デフォルトで **.snap** です。

### 前提条件

- 実行中で正常な 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

- スナップショットを作成するには、**.snap** ディレクトリーに新しいサブディレクトリーを作成します。スナップショット名は新しいサブディレクトリー名です。

### 構文

```
mkdir NEW_DIRECTORY_PATH
```

### 例

```
[root@mds cephfs]# mkdir .snap/new-snaps
```

以下の例では、**/mnt/cephfs** にマウントされている Ceph File System に **new-snaps** サブディレクトリーを作成し、スナップショットの作成を開始するように Ceph Metadata Server (MDS) に通知します。

### 検証

- 新規スナップショットディレクトリーを一覧表示します。

### 構文

```
ls -l .snap/
```

**new-snaps** サブディレクトリーが **.snap** ディレクトリーの下に表示されます。

### 関連情報

- スナップショットの削除について、詳細は [Red Hat Ceph Storage File System Guide の Deleting a snapshot for a Ceph File System](#) セクションを参照してください。
- スナップショットの復元について、詳細は [Red Hat Ceph Storage File System Guide の Restoring a snapshot for a Ceph File System](#) セクションを参照してください。

## 4.10.5. Ceph File System のスナップショットの削除

**.snap** ディレクトリーの対応するディレクトリーを削除して、スナップショットを削除できます。

### 前提条件

- 実行中で正常な Red Hat Ceph Storage クラスタ
- Ceph File System のデプロイメント
- Ceph File System スナップショットの作成。
- Ceph Metadata Server (MDS) ノードへのルートレベルのアクセス。

### 手順

- スナップショットを削除するには、対応するディレクトリーを削除します。

### 構文

```
rmdir DIRECTORY_PATH
```

### 例

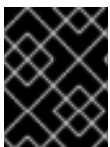
```
[root@mds cephfs]# rmdir .snap/new-snaps
```

以下の例では、`/mnt/cephfs` にマウントされている Ceph File System の `new-snaps` サブディレクトリーを削除します。



### 注記

通常のディレクトリーとは異なり、ディレクトリーが空でなくても `rmdir` コマンドは成功するため、再帰的な `rm` コマンドを使用する必要はありません。



### 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

- スナップショットの復元について、詳細は [Red Hat Ceph Storage File System Guide の Restoring a snapshot for a Ceph File System](#) セクションを参照してください。
- スナップショットの作成について、詳細は [Red Hat Ceph Storage File System Guide の Creating a snapshot for a Ceph File System](#) セクションを参照してください。

## 4.10.6. Ceph File System スナップショットの作成

スナップショットからファイルを復元したり、Ceph File System (CephFS) の完全なスナップショットを完全に復元したりできます。

### 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
- Ceph Metadata Server (MDS) ノードへのルートレベルのアクセス。

## 手順

- スナップショットからファイルを復元するには、スナップショットディレクトリーから通常のツリーにコピーします。

### 構文

```
cp -a .snap/SNAP_DIRECTORY/FILENAME
```

### 例

```
[root@mds dir1]# cp .snap/new-snaps/file1 .
```

この例では、**file1** を現在のディレクトリーに復元します。

- **.snap** ディレクトリーツリーからスナップショットを完全に復元することもできます。現在のエントリーを必要なスナップショットのコピーに置き換えます。

### 構文

```
[root@mds dir1]# rm -rf *  
[root@mds dir1]# cp -a .snap/SNAP_DIRECTORY/* .
```

### 例

```
[root@mds dir1]# rm -rf *  
[root@mds dir1]# cp -a .snap/new-snaps/* .
```

この例では、**dir1** 下のファイルおよびディレクトリーをすべて削除し、そのファイルを **new-snaps** スナップショットから現在のディレクトリーである **dir1** に復元します。

## 4.10.7. 関連情報

- Red Hat Ceph Storage File System Guide の [Deployment of the Ceph File System](#) セクションを参照してください。

## 4.11. CEPH FILE SYSTEM クラスターの停止

単に **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) を下させることもできます。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 ノードへのユーザーアクセス。

### 手順

1. CephFS クラスター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

```
[root@mon]# 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

```
[root@mon]# 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

```
[root@mon]# ceph fs fail cephfs
```

## 4.12. コマンドラインインターフェイスを使用した CEPH FILE SYSTEM の削除

コマンドラインインターフェイスを使用して、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

- データをバックアップしている。
- すべてのクライアントが Ceph File System (CephFS) をアンマウントしている。

- Ceph Monitor ノードへの root レベルのアクセス。

## 手順

1. CephFS ステータスを表示して MDS ランクを確認します。

### 構文

```
ceph fs status
```

### 例

```
[root@mon ~]# ceph fs status
cephfs - 0 clients
=====
+-----+-----+-----+-----+-----+
| Rank | State | MDS | Activity | dns | inos |
+-----+-----+-----+-----+-----+
| 0 | active | cluster1-node6 | Reqs: 0/s | 10 | 13 |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
| Pool | type | used | avail |
+-----+-----+-----+-----+
| cephfs_metadata | metadata | 2688k | 15.0G |
| cephfs_data | data | 0 | 15.0G |
+-----+-----+-----+-----+
+-----+
| Standby MDS |
+-----+
| cluster1-node5 |
+-----+
```

上記の例では、ランクは 0 です。

2. CephFS をダウンとしてマークします。

### 構文

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

**FS\_NAME** を、削除する CephFS の名前に置き換えます。

### 例

```
[root@mon]# ceph fs set cephfs down true
marked down
```

3. CephFS のステータスを表示して、停止しているかどうかを判断します。

### 構文

```
ceph fs status
```

## 例

```
[root@mon ~]# ceph fs status
cephfs - 0 clients
=====
+-----+-----+-----+-----+-----+
| Rank | State | MDS | Activity | dns | inos |
+-----+-----+-----+-----+-----+
| 0 | stopping | cluster1-node6 | | 10 | 12 |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
| Pool | type | used | avail |
+-----+-----+-----+-----+
| cephfs_metadata | metadata | 2688k | 15.0G |
| cephfs_data | data | 0 | 15.0G |
+-----+-----+-----+-----+
+-----+
| Standby MDS |
+-----+
| cluster1-node5 |
+-----+
```

しばらくすると、MDS は一覧表示されなくなります。

## 例

```
[root@mon ~]# ceph fs status
cephfs - 0 clients
=====
+-----+-----+-----+-----+-----+
| Rank | State | MDS | Activity | dns | inos |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
| Pool | type | used | avail |
+-----+-----+-----+-----+-----+
| cephfs_metadata | metadata | 2688k | 15.0G |
| cephfs_data | data | 0 | 15.0G |
+-----+-----+-----+-----+-----+
+-----+
| Standby MDS |
+-----+
| cluster1-node5 |
+-----+
```

- ステップ1のステータスで、MDS のランクがすべて表示されます。

## 構文

```
ceph mds fail RANK
```

**RANK** を、MDS デーモンのランクに置き換え、失敗します。

## 例

```
[root@mon]# ceph mds fail 0
```

5. CephFS を削除します。

#### 構文

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

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

#### 例

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

6. ファイルシステムが削除されていることを確認します。

#### 構文

```
ceph fs ls
```

#### 例

```
[root@mon ~]# ceph fs ls  
No filesystems enabled
```

7. オプション: CephFS が使用したプールを削除します。

- a. Ceph Monitor ノードで、プールを一覧表示します。

#### 構文

```
ceph osd pool ls
```

#### 例

```
[root@mon ~]# ceph osd pool ls  
rbd  
cephfs_data  
cephfs_metadata
```

この出力例では、**cephfs\_metadata** および **cephfs\_data** は CephFS によって使用されたプールです。

- b. メタデータプールを削除します。

#### 構文

```
ceph osd pool delete CEPH_METADATA_POOL CEPH_METADATA_POOL --yes-i-really-really-mean-it
```

プール名を 2 回追加することで、**CEPH\_METADATA\_POOL** をメタデータストレージに使用されるプール CephFS に置き換えます。

**例**

```
[root@mon ~]# ceph osd pool delete cephfs_metadata cephfs_metadata --yes-i-really-mean-it
pool 'cephfs_metadata' removed
```

- c. データプールを削除します。

**構文**

```
ceph osd pool delete CEPH_DATA_POOL CEPH_DATA_POOL --yes-i-really-mean-it
```

プール名を 2 回追加することで、**CEPH\_DATA\_POOL** をデータストレージに使用するプール CephFS に置き換えます。

**例**

```
[root@mon ~]# ceph osd pool delete cephfs_data cephfs_data --yes-i-really-mean-it
pool 'cephfs_data' removed
```

**関連情報**

- Red Hat Ceph Storage File System Guide の [Removing a Ceph File System Using Ansible](#) を参照してください。
- Red Hat Ceph Storage 戦略ガイドの [プールの削除](#) セクションを参照してください。

**4.13. ANSIBLE を使用した CEPH FILE SYSTEM の削除**

**ceph-ansible** を使用して Ceph File System( CephFS) を削除することができます。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**

- 稼働中の Red Hat Ceph Storage クラスタがある。
- データの適切なバックアップ。
- すべてのクライアントが Ceph File System をアンマウントしている。
- Ansible 管理ノードへのアクセス

- Ceph Monitor ノードへの root レベルのアクセス。

## 手順

1. `/usr/share/ceph-ansible` ディレクトリーに移動します。

```
[admin@admin ~]$ cd /usr/share/ceph-ansible
```

2. Ansible インベントリーファイルの **[mdss]** セクションを確認して、Ceph Metadata Server (MDS) ノードを特定します。Ansible 管理ノードで `/usr/share/ceph-ansible/hosts` を開きます。

### 例

```
[mdss]
cluster1-node5
cluster1-node6
```

この例では、**cluster1-node5** および **cluster1-node6** は MDS ノードです。

3. **max\_mds** パラメーターを **1** に設定します。

### 構文

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

### 例

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

4. 削除するメタデータサーバー (MDS) を指定して、Playbook **shrink-mds.yml** を実行します。

### 構文

```
ansible-playbook infrastructure-playbooks/shrink-mds.yml -e mds_to_kill=MDS_NODE -i
hosts
```

**MDS\_NODE** を、削除するメタデータサーバーノードに置き換えます。Ansible Playbook から、クラスターを縮小するかどうかを尋ねられます。**yes** と入力して、Enter キーを押します。

### 例

```
[admin@admin ceph-ansible]$ ansible-playbook infrastructure-playbooks/shrink-mds.yml -e
mds_to_kill=cluster1-node6 -i hosts
```

5. オプション: 追加の MDS ノードのプロセスを繰り返します。

### 構文

```
ansible-playbook infrastructure-playbooks/shrink-mds.yml -e mds_to_kill=MDS_NODE -i
hosts
```

**MDS\_NODE** を、削除するメタデータサーバーノードに置き換えます。Ansible Playbook から、クラスターを縮小するかどうかを尋ねられます。**yes** と入力して、Enter キーを押します。

## 例

```
[admin@admin ceph-ansible]$ ansible-playbook infrastructure-playbooks/shrink-mds.yml -e mds_to_kill=cluster1-node5 -i hosts
```

6. CephFS のステータスを確認します。

## 構文

```
ceph fs status
```

## 例

```
[root@mon ~]# ceph fs status
cephfs - 0 clients
=====
+-----+-----+-----+-----+-----+-----+
| Rank | State | MDS | Activity | dns | inos |
+-----+-----+-----+-----+-----+
| 0 | failed | cluster1-node6 | Reqs: 0/s | 10 | 13 |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
| Pool | type | used | avail |
+-----+-----+-----+-----+
| cephfs_metadata | metadata | 2688k | 15.0G |
| cephfs_data | data | 0 | 15.0G |
+-----+-----+-----+-----+
+-----+
| Standby MDS |
+-----+
| cluster1-node5 |
+-----+
```

7. **mdss** セクションとその中のノードを Ansible インベントリーファイルから削除して、**site.yml** または **site-container.yml** Playbook の今後の実行でメタデータサーバーとして再プロビジョニングされないようにします。Ansible インベントリーファイル `/usr/share/ceph-ansible/hosts` を編集するために開きます。

## 例

```
[mdss]
cluster1-node5
cluster1-node6
```

**[mdss]** セクションと、その下のすべてのノードを削除します。

8. CephFS を削除します。

## 構文

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

■

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

### 例

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

9. オプション: CephFS が使用したプールを削除します。

a. Ceph Monitor ノードで、プールを一覧表示します。

### 構文

```
ceph osd pool ls
```

CephFS が使用したプールを見つけます。

### 例

```
[root@mon ~]# ceph osd pool ls
rbd
cephfs_data
cephfs_metadata
```

この出力例では、**cephfs\_metadata** および **cephfs\_data** は CephFS によって使用されたプールです。

b. メタデータプールを削除します。

### 構文

```
ceph osd pool delete CEPH_METADATA_POOL CEPH_METADATA_POOL --yes-i-really-really-mean-it
```

プール名を 2 回追加することで、**CEPH\_METADATA\_POOL** をメタデータストレージに使用されるプール CephFS に置き換えます。

### 例

```
[root@mon ~]# ceph osd pool delete cephfs_metadata cephfs_metadata --yes-i-really-really-mean-it
pool 'cephfs_metadata' removed
```

c. データプールを削除します。

### 構文

```
ceph osd pool delete CEPH_DATA_POOL CEPH_DATA_POOL --yes-i-really-really-mean-it
```

プール名を 2 回追加することで、**CEPH\_METADATA\_POOL** をメタデータストレージに使用されるプール CephFS に置き換えます。



## 例

```
[root@mon ~]# ceph osd pool delete cephfs_data cephfs_data --yes-i-really-really-mean-it
pool 'cephfs_data' removed
```

- d. そのプールが存在しないことを確認します。

## 例

```
[root@mon ~]# ceph osd pool ls
rbd
```

**cephfs\_metadata** プールおよび **cephfs\_data** プールは一覧表示されなくなりました。

## 関連情報

- Red Hat Ceph Storage File System Guide の [Removing a Ceph File System Manually](#) を参照してください。
- Red Hat Ceph Storage 戦略ガイドの [プールの削除](#) セクションを参照してください。

## 4.14. 最小クライアントバージョンの設定

サードパーティーのクライアントを実行している必要のある最低限の Ceph を設定して、Red Hat Ceph Storage Ceph File System (CephFS) に接続することができます。古いクライアントがファイルシステムをマウントしないように **min\_compat\_client** パラメーターを設定します。また、CephFS は、**min\_compat\_client** で設定されたバージョンよりも古いバージョンを使用する現在接続されているクライアントも自動的に削除します。

この設定の理由は、バグが含まれている可能性がある、または機能の互換性が不完全な古いクライアントがクラスターに接続して他のクライアントを中断しないようにすることです。たとえば、CephFS クライアントの古いバージョンは、機能を適切にリリースせず、他のクライアント要求の処理速度が遅くなる可能性があります。

**min\_compat\_client** の値は、アップストリームの Ceph バージョンに基づいています。Red Hat は、サードパーティークライアントが Red Hat Ceph Storage クラスターベースであるのと同じメジャーバージョンのアップストリームバージョンを使用することを推奨します。アップストリームバージョンおよび対応する Red Hat Ceph Storage バージョンを確認するには、以下の表を参照してください。

表4.1 min\_compat\_client 値

値	アップストリームの Ceph バージョン	Red Hat Ceph Storage のバージョン
luminous	12.2	Red Hat Ceph Storage 3
mimic	13.2	該当せず
nautilus	14.2	Red Hat Ceph Storage 4



## 重要

Red Hat Enterprise Linux 7 を使用している場合は、**min\_compat\_client** を、**luminous** よりも新しいバージョンに設定しないでください。これは、Red Hat Enterprise Linux 7 は luminous のクライアントと見なされており、それ以降のバージョンを使用する場合は、CephFS でマウントポイントにアクセスできなくなります。

### 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

### 手順

1. クライアントの最小バージョンを設定します。

```
ceph fs set name min_compat_client release
```

**name** を Ceph File System の名前に置き換え、**release** を最小クライアントバージョンに置き換えます。たとえば、クライアントが最低限の **cephfs** Ceph File System で **nautilus** のアップストリームバージョンを使用するようにクライアントを制限する場合は、以下を実行します。

```
$ ceph fs set cephfs min_compat_client nautilus
```

使用可能な値の完全なリストと、その値が Red Hat CephStorage のバージョンとどのように対応するかについては、[表4.1 「min\\_compat\\_client 値」](#) を参照してください。

## 4.15. 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 デーモンに failed 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

1. デーモンが失敗するには、以下を実行します。

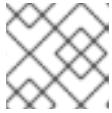
#### 構文

```
ceph mds fail MDS_NAME
```

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

#### 例

```
[root@mds ~]# 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 Guideの [Decreasing the Number of Active MDS Daemons](#) を参照してください。
- Red Hat Ceph Storage File System Guideの [Configuring Standby Metadata Server Daemons](#) を参照してください。
- Red Hat Ceph Storage File System Guideの [Explanation of Ranks in Metadata Server Configuration](#) を参照してください。

## 4.16. 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 をバッファリングする場合、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は、

### 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。

## 4.17. ブラックリスト CEPH FILE SYSTEM クライアント

Ceph File System クライアントのブラックリスト機能は、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、ブラックリストが他の 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、ブラックリストに登録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ブラックリストに登録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せんが、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_blacklist_on_timeout` および `mds_session_blacklist_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

ブラ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。

## 4.18. 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 ノードへのユーザーアクセス。

### 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

```
[root@mon]# 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": "ae81e49d369875ac8b569ff3e3c456a31b8f3af5",
      "ceph_version": "ceph version 12.0.0-1934-gae81e49
(ae81e49d369875ac8b569ff3e3c456a31b8f3af5)",
      "entity_id": "0",
      "hostname": "senta04",
      "mount_point": "/tmp/tmpcMpF1b/mnt.0",
      "pid": "29377",
      "root": "/"
    }
  }
}
]

```

2. 指定した CephFS クライアントをエビクトします。

### 構文

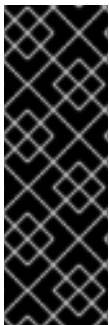
```
ceph tell DAEMON_NAME client evict id=ID_NUMBER
```

### 例

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

## 4.19. ブラックリストからの CEPH FILE SYSTEM クライアントの削除

場合によっては、以前のブラックリストされた 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

ブラックリストから 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、ブラックリストに登録された 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 ノードへのユーザーアクセス。

### 手順

1. ブラックリストを確認します。

### 例

```

[root@mon]# ceph osd blacklist ls
listed 1 entries
127.0.0.1:0/3710147553 2020-03-19 11:32:24.716146

```

2. ブラックリストから CephFS クライアントを削除します。

### 構文

```
ceph osd blacklist rm CLIENT_NAME_OR_IP_ADDR
```

### 例

```
[root@mon]# ceph osd blacklist rm 127.0.0.1:0/3710147553  
un-blacklisting 127.0.0.1:0/3710147553
```

3. 必要に応じて、ブラックリストから FUSE ベースの CephFS クライアントを削除するときに自動的に再接続を試みるようにするには、以下を実行します。FUSE クライアントで、以下のオプションを **true** に設定します。

```
client_reconnect_stale = true
```

## 4.20. 関連情報

- 詳細は [3章 Ceph File System のデプロイメント](#) 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 Guide](#) の [Configuring Metadata Server Daemons](#) を参照してください。

## 第5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

本章では、以下を使用する方法を説明します。

- [Ceph File System ボリューム](#)
- [Ceph File System サブボリュームグループ](#)
- [Ceph File System サブボリューム](#)

### 5.1. CEPH FILE SYSTEM ボリューム

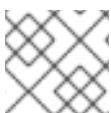
ストレージ管理者は、Ceph File System (CephFS) ボリュームの作成、一覧表示、および削除を行うことができます。CephFS ボリュームは、Ceph File Systems の抽象化です。

本セクションでは、以下を行う方法を説明します。

- [ファイルシステムボリュームを作成します。](#)
- [ファイルシステムボリュームを一覧表示します。](#)
- [ファイルシステムボリュームを削除します。](#)

#### 5.1.1. ファイルシステムボリュームの作成

Ceph Manager のオーケストレーターモジュールは、Ceph File System (CephFS) 用にメタデータデータサーバー (MDS) を作成します。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

1. 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
```

## 5.1.2. 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) ボリュームを一覧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

1. CephFS ボリュームを一覧表示します。

### 例

```
[root@mon ~]# ceph fs volume ls
```

## 5.1.3. ファイルシステムボリュームの削除

Ceph Manager のオーケストレーターモジュールは、Ceph File System (CephFS) の Meta Data Server (MDS) を削除します。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. 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

```
[root@mon ~]# ceph fs volume rm cephfs --yes-i-really-mean-it
```

## 5.2. 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、一覧表示、取得、メタデータの取得、削除が可能です。

CephFS サブボリュームに対して Ceph クライアントユーザーを承認することもできます。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、一覧表示、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一覧表示。](#)
- [File System サブボリュームの Ceph クライアントユーザーの認証。](#)
- [File System サブボリュームの Ceph クライアントユーザーの認証解除。](#)
- [File System サブボリュームの Ceph クライアントユーザーの一覧表示。](#)
- [File System のサブボリュームからの Ceph クライアントユーザーのエビクト。](#)
- [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。](#)
- [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一覧表示。](#)
- [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。](#)

### 5.2.1. 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

1. 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。

## 5.2.2. ファイルシステムのサブボリュームの一覧表示

本項では、Ceph File System (CephFS) サブボリュームを一覧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

1. CephFS サブボリュームを一覧表示します。

### 構文

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

### 例

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

## 5.2.3. File System サブボリューム用の Ceph クライアントユーザーの認証

Red Hat Ceph Storage クラスターは認証に **cephx** を使用します。これはデフォルトで有効になっています。Ceph File System (CephFS) サブボリュームで **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 がマウントされているノードで利用できるようにします。**authorize** コマンドを使用して、CephFS サブボリュームにアクセスするユーザーを承認できます。

### 前提条件

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

1. 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。

2. CephFS サブボリュームへの読み書きアクセスのいずれかで、Ceph クライアントユーザーを承認します。

#### 構文

```
ceph fs subvolume authorize VOLUME_NAME SUBVOLUME_NAME AUTH_ID [--  
group_name=GROUP_NAME] [--access_level=ACCESS_LEVEL]
```

**ACCESS\_LEVEL** は **r** または **rw** で、**AUTH\_ID** は Ceph クライアントユーザー (文字列) です。

### 例

-

```
[root@mon ~]# ceph fs subvolume authorize cephfs sub0 guest --group_name=subgroup0 --access_level=rw
```

この例では、client.guest は、サブボリュームグループ **subgroup0** のサブボリューム **sub0** にアクセスすることを承認されています。

## 関連情報

- Red Hat Ceph Storage Configuration Guide の [Ceph authentication configuration](#) セクションを参照してください。
- Red Hat Ceph Storage Ceph File System Guide の [Creating a file system volume](#) セクションを参照してください。

## 5.2.4. File System サブボリュームの Ceph クライアントユーザーの認証解除

**deauthorize** コマンドを使用して、Ceph File System (CephFS) サブボリュームにアクセスするユーザーを承認できます。

### 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 ボリュームおよびサブボリュームが作成されている。
- CephFS サブボリュームへのアクセスが承認されている Ceph クライアントユーザー。

### 手順

- Ceph クライアントユーザーの CephFS サブボリュームへのアクセス許可を解除します。

### 構文

```
ceph fs subvolume deauthorize VOLUME_NAME SUBVOLUME_NAME AUTH_ID [--group_name=GROUP_NAME]
```

**AUTH\_ID** は Ceph クライアントユーザーであり、文字列です。

### 例

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

この例では、client.guest は、サブボリュームグループ **subgroup0** のサブボリューム **sub0** へのアクセス許可を解除されています。

## 関連情報

- Red Hat Ceph Storage Ceph File System Guide の [Authorizing Ceph client users for File System subvolumes](#) を参照してください。

### 5.2.5. File System サブボリュームの Ceph クライアントユーザーの一覧表示

`authorized_list` コマンドを使用して、Ceph File System (CephFS) サブボリュームへのユーザーアクセスを一覧表示することができます。

#### 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 ボリュームおよびサブボリュームが作成されている。
- CephFS サブボリュームへのアクセスが承認されている Ceph クライアントユーザー。

#### 手順

- Ceph クライアントユーザーの CephFS サブボリュームへのアクセスを一覧表示します。

#### 構文

```
ceph fs subvolume authorized_list VOLUME_NAME SUBVOLUME_NAME [--group_name=GROUP_NAME]
```

#### 例

```
[root@mon ~]# ceph fs subvolume authorized_list cephfs sub0 --group_name=subgroup0
[
  {
    "guest": "rw"
  }
]
```

#### 関連情報

- Red Hat Ceph Storage Ceph File System Guide の [Authorizing Ceph client users for File System subvolumes](#) を参照してください。

### 5.2.6. File System サブボリュームからの Ceph クライアントユーザーのエビクト

`_AUTH_ID` とマウントされたサブボリュームに基づく `evict` コマンドを使用して、Ceph File System (CephFS) サブボリュームから Ceph クライアントユーザーをエビクトできます。

#### 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 ボリュームおよびサブボリュームが作成されている。

- CephFS サブボリュームへのアクセスが承認されている Ceph クライアントユーザー。

## 手順

- CephFS サブボリュームから Ceph クライアントユーザーをエビクトします。

## 構文

```
ceph fs subvolume evict VOLUME_NAME SUBVOLUME_NAME AUTH_ID [--group_name=GROUP_NAME]
```

**AUTH\_ID** は Ceph クライアントユーザーであり、文字列です。

## 例

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

この例では、client.guest はサブボリュームグループ **subgroup0** からエビクトされています。

## 関連情報

- Red Hat Ceph Storage Ceph File System Guide の [Authorizing Ceph client users for File System subvolumes](#) を参照してください。

## 5.2.7. 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の下に縮小されないようにします。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

1. 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
```

### 5.2.8. 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

1. 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  
  
/volumes/subgroup0/sub0/c10cc8b8-851d-477f-99f2-1139d944f691
```

### 5.2.9. 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

1. 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

```
{  
  "atime": "2020-09-08 09:27:15",  
  "bytes_pcent": "undefined",  
  "bytes_quota": "infinite",  
  "bytes_used": 0,  
  "created_at": "2020-09-08 09:27:15",  
  "ctime": "2020-09-08 09:27:15",  
  "data_pool": "cephfs_data",  
  "features": [  
    "snapshot-clone",  
    "snapshot-autoprotect",  
    "snapshot-retention"  
  ],  
  "gid": 0,  
  "mode": 16877,  
  "mon_addrs": [  
    "10.8.128.22:6789",  
    "10.8.128.23:6789",  
    "10.8.128.24:6789"  
  ],  
  "mtime": "2020-09-08 09:27:15",  
  "path": "/volumes/subgroup0/sub0/6d01a68a-e981-4ebe-84ca-96b660879173",  
  "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。
- **mtime**: サブボリュームパスの変更時間 ("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。
- **UID**: サブボリュームの UID。
- **GID**: サブボリュームパスの GID。
- **モード**: サブボリュームのモード。
- **mon\_addrs**: モニターアドレスの一覧。



- **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)。
- **data\_pool**: サブボリュームが属するデータプール。
- **path**: サブボリュームの絶対パス。
- **type**: subvolume タイプは、そのクローンまたはサブボリュームであることを示します。
- **pool\_namespace**: サブボリュームの RADOS 名前空間。
- **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。
- **state**: サブボリュームの現在の状態 (例:complete または snapshot-retained)

### 5.2.10. 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ーパスに **s** フラグも必要になります。

#### 手順

1. **s** フラグがディレクトリーに設定されていることを確認します。

#### 構文

```
ceph auth get CLIENT_NAME
```

#### 例

```
client.0
key: AQAz7EVWygILFRAAdIcuJ12opU/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。

2. 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
_SSNAP_NAME [--group_name GROUP_NAME]
```

### 例

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

## 5.2.11. 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。

### 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: AQAz7EVWYygILFRAdIcuJ12opU/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: AQAz7EVWYygILFRAdIcuJ12opU/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
```

4. サブボリュームのスナップショットを作成します。

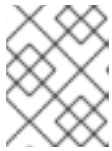
### 構文

```
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
```

5. 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。

- a. 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_SUBVOLUME_NAME
```

#### 例

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

- b. 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_SUBVOLUME_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_SUBVOLUME_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。

## 5.2.12. 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一覧表示

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一覧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

1. CephFS サブボリュームのスナップショットを一覧表示します。

### 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
```

## 5.2.13. 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 システム (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": "2021-09-08 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)。

## 5.2.14. 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思なされます。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**: 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

```
ceph fs subvolume snapshot clone cephfs sub0 snap0 sub1 --group_name subgroup0 --
target_group_name subgroup0
```

## 5.2.15. 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 システム (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

1. 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
```

## 5.3. 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、一覧表示、取得、および削除できます。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、一覧表示、および削除も可能です。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。

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。](#)
- [ファイルシステムのサブボリュームグループを一覧表示します。](#)
- [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作成します。](#)
- [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一覧表示します。](#)
- [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。](#)



### 5.3.1. 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。

#### 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. 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

```
[root@mon ~]# 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。

### 5.3.2. ファイルシステムのサブボリュームグループの一覧表示

本項では、Ceph File System (CephFS) サブボリュームグループを一覧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

1. CephFS サブボリュームグループを一覧表示します。

#### 構文

```
ceph fs subvolumegroup ls VOLUME_NAME
```

#### 例

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

### 5.3.3. 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

1. 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

```
[root@mon ~]# ceph fs subvolumegroup getpath cephfs subgroup0  
  
/volumes/subgroup0
```

### 5.3.4. 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作成

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 サブボリュームグループ
- 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ーパスに **s** フラグも必要になります。

## 手順

1. **s** フラグがディレクトリーに設定されていることを確認します。

### 構文

```
ceph auth get CLIENT_NAME
```

### 例

```
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。

2. CephFS サブボリュームグループのスナップショットを作成します。

### 構文

```
ceph fs subvolumegroup snapshot create VOLUME_NAME _GROUP_NAME SNAP_NAME
```

### 例

```
[root@mon ~]# ceph fs subvolumegroup snapshot create cephfs subgroup0 snap0
```

このコマンドは、subvolume グループ下のすべてのサブボリュームを暗黙的にスナップショットします。

## 5.3.5. 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一覧表示

本項では、Ceph File システム (CephFS) サブボリュームグループのスナップショットを一覧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

1. CephFS サブボリュームグループのスナップショットを一覧表示します。

## 構文

```
ceph fs subvolumegroup snapshot ls VOLUME_NAME GROUP_NAME
```

## 例

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

### 5.3.6. 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 システム (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

1. 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

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

### 5.3.7. 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

1. 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

```
[root@mon ~]# ceph fs subvolumegroup rm cephfs subgroup0 --force
```

## 5.4. 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。

## 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、および説明の一覧です。

#### "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。

#### "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、[Understanding MDS Cache Size Limits](#) 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一覧表示します。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。

- 詳しくは、[Red Hat Ceph Storage ファイルシステムガイド](#)の**メタデータサーバー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

Metadata Server (MDS) デーモン設定に使用できるリストコマンドを参照してください。

### mon\_force\_standby\_active

#### 詳細

**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。

#### タイプ

ブール値

#### デフォルト

**true**

### max\_mds

#### 詳細

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。

#### タイプ

32 ビット整数

#### デフォルト

**1**

### mds\_cache\_memory\_limit

#### 詳細

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。

#### タイプ

64 ビット整数未署名

#### デフォルト

**1073741824**

### mds\_cache\_reservation

#### 詳細

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。

#### タイプ

浮動小数点 (Float)

#### デフォルト

**0.05**

### mds\_cache\_size

#### 詳細

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。

**タイプ**

32 ビット整数

**デフォルト****0****mds\_cache\_mid****詳細**

キャッシュ LRU 内の新しい項目の挿入ポイント (トップ)

**タイプ**

浮動小数点 (Float)

**デフォルト****0.7****mds\_dir\_commit\_ratio****詳細**

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。

**タイプ**

浮動小数点 (Float)

**デフォルト****0.5****mds\_dir\_max\_commit\_size****詳細**

Ceph がディレクトリーが小規模なトランザクションに分割される前にディレクトリー更新の最大サイズ (MB 単位)。

**タイプ**

32 ビット整数

**デフォルト****90****mds\_decay\_halfife****詳細**

MDS キャッシュ温度の半期

**タイプ**

浮動小数点 (Float)

**デフォルト****5****mds\_beacon\_interval****詳細**

モニターに送信されるメッセージの頻度 (秒単位)。

**タイプ**

浮動小数点 (Float)

**デフォルト**

**4**

**mds\_beacon\_grace****詳細**

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。

**タイプ**

浮動小数点 (Float)

**デフォルト**

**15**

**mds\_blacklist\_interval****詳細**

OSD マップの失敗した MDS デーモンのブラックリスト期間。

**タイプ**

浮動小数点 (Float)

**デフォルト**

**24.0\*60.0**

**mds\_session\_timeout****詳細**

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)。

**タイプ**

浮動小数点 (Float)

**デフォルト**

**60**

**mds\_session\_autoclose****詳細**

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)。

**タイプ**

浮動小数点 (Float)

**デフォルト**

**300**

**mds\_reconnect\_timeout****詳細**

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)。

**タイプ**

浮動小数点 (Float)

**デフォルト**

**45**

#### **mds\_tick\_interval**

**詳細**

MDS が内部周期的タスクを実行する頻度。

**タイプ**

浮動小数点 (Float)

**デフォルト**

**5**

#### **mds\_dirstat\_min\_interval**

**詳細**

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)。

**タイプ**

浮動小数点 (Float)

**デフォルト**

**1**

#### **mds\_scatter\_nudge\_interval**

**詳細**

ディレクトリー統計の急速な変更が反映されます。

**タイプ**

浮動小数点 (Float)

**デフォルト**

**5**

#### **mds\_client\_prealloc\_inos**

**詳細**

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。

**タイプ**

32 ビット整数

**デフォルト**

**1000**

#### **mds\_early\_reply**

**詳細**

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。

**タイプ**

ブール値

**デフォルト**

**true**

## mds\_use\_tmap

### 詳細

ディレクトリーの更新には、**trivialmap** を使用します。

### タイプ

ブール値

### デフォルト

**true**

## mds\_default\_dir\_hash

### 詳細

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。

### タイプ

32 ビット整数

### デフォルト

**2**、つまり **rjenkins**

## mds\_log

### 詳細

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。

### タイプ

ブール値

### デフォルト

**true**

## mds\_log\_skip\_corrupt\_events

### 詳細

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。

### タイプ

ブール値

### デフォルト

**false**

## mds\_log\_max\_events

### 詳細

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。

### タイプ

32 ビット整数

### デフォルト

**-1**

## mds\_log\_max\_segments

### 詳細

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。

### タイプ

32 ビット整数

### デフォルト

**30**

## mds\_log\_max\_expiring

### 詳細

並行して期限切れになるセグメントの最大数。

### タイプ

32 ビット整数

### デフォルト

**20**

## mds\_log\_eopen\_size

### 詳細

**EOpen** イベントにおける inode の最大数。

### タイプ

32 ビット整数

### デフォルト

**100**

## mds\_bal\_sample\_interval

### 詳細

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。

### タイプ

浮動小数点 (Float)

### デフォルト

**3**

## mds\_bal\_replicate\_threshold

### 詳細

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。

### タイプ

浮動小数点 (Float)

### デフォルト

**8000**

## mds\_bal\_unreplicate\_threshold

### 詳細

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。

**タイプ**

浮動小数点 (Float)

**デフォルト**

**0**

**mds\_bal\_frag****詳細**

MDS フラグメントのディレクトリーを決定します。

**タイプ**

ブール値

**デフォルト**

**false**

**mds\_bal\_split\_size****詳細**

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。

**タイプ**

32 ビット整数

**デフォルト**

**10000**

**mds\_bal\_split\_rd****詳細**

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。

**タイプ**

浮動小数点 (Float)

**デフォルト**

**25000**

**mds\_bal\_split\_wr****詳細**

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。

**タイプ**

浮動小数点 (Float)

**デフォルト**

**10000**

**mds\_bal\_split\_bits****詳細**

ディレクトリーフラグメントを分割するビット数。

**タイプ**

32 ビット整数

デフォルト

**3**

### **mds\_bal\_merge\_size**

詳細

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。

タイプ

32 ビット整数

デフォルト

**50**

### **mds\_bal\_merge\_rd**

詳細

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。

タイプ

浮動小数点 (Float)

デフォルト

**1000**

### **mds\_bal\_merge\_wr**

詳細

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。

タイプ

浮動小数点 (Float)

デフォルト

**1000**

### **mds\_bal\_interval**

詳細

MDS ノード間のワークロード交換の頻度 (秒単位)。

タイプ

32 ビット整数

デフォルト

**10**

### **mds\_bal\_fragment\_interval**

詳細

ディレクトリーの断片化を調整する頻度 (秒単位)。

タイプ

32 ビット整数

デフォルト



## 5

**mds\_bal\_idle\_threshold****詳細**

Ceph がサブツリーをその親に移行する前の最小温度。

**タイプ**

浮動小数点 (Float)

**デフォルト**

0

**mds\_bal\_max****詳細**

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。

**タイプ**

32 ビット整数

**デフォルト**

-1

**mds\_bal\_max\_until****詳細**

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。

**タイプ**

32 ビット整数

**デフォルト**

-1

**mds\_bal\_mode****詳細**

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

**タイプ**

32 ビット整数

**デフォルト**

0

**mds\_bal\_min\_rebalance****詳細**

Ceph の移行前の最小サブツリーの温度。

**タイプ**

浮動小数点 (Float)

**デフォルト**

**0.1**

**mds\_bal\_min\_start**

**詳細**

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。

**タイプ**

浮動小数点 (Float)

**デフォルト**

**0.2**

**mds\_bal\_need\_min**

**詳細**

許可するターゲットサブツリーの最小分数。

**タイプ**

浮動小数点 (Float)

**デフォルト**

**0.8**

**mds\_bal\_need\_max**

**詳細**

許可するターゲットサブツリーサイズの最大分数。

**タイプ**

浮動小数点 (Float)

**デフォルト**

**1.2**

**mds\_bal\_midchunk**

**詳細**

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。

**タイプ**

浮動小数点 (Float)

**デフォルト**

**0.3**

**mds\_bal\_minchunk**

**詳細**

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。

**タイプ**

浮動小数点 (Float)

**デフォルト**

**0.001**

## mds\_bal\_target\_removal\_min

### 詳細

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。

### タイプ

32 ビット整数

### デフォルト

5

## mds\_bal\_target\_removal\_max

### 詳細

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。

### タイプ

32 ビット整数

### デフォルト

10

## mds\_replay\_interval

### 詳細

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。

### タイプ

浮動小数点 (Float)

### デフォルト

1

## mds\_shutdown\_check

### 詳細

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。

### タイプ

32 ビット整数

### デフォルト

0

## mds\_thrash\_exports

### 詳細

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。

### タイプ

32 ビット整数

### デフォルト

0

## mds\_thrash\_fragments

### 詳細

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。

**タイプ**

32 ビット整数

**デフォルト**

0

**mds\_dump\_cache\_on\_map****詳細**

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。

**タイプ**

ブール値

**デフォルト**

false

**mds\_dump\_cache\_after\_rejoin****詳細**

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。

**タイプ**

ブール値

**デフォルト**

false

**mds\_verify\_scatter****詳細**

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。

**タイプ**

ブール値

**デフォルト**

false

**mds\_debug\_scatterstat****詳細**

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。

**タイプ**

ブール値

**デフォルト**

false

**mds\_debug\_frag****詳細**

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。

**タイプ**

ブール値

**デフォルト**

**false**

**mds\_debug\_auth\_pins****詳細**

デバッグ認証のバリエーション。開発者向けの使用のみ。

**タイプ**

ブール値

**デフォルト**

**false**

**mds\_debug\_subtrees****詳細**

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。

**タイプ**

ブール値

**デフォルト**

**false**

**mds\_kill\_mdstable\_at****詳細**

Ceph の MDS 表のコードで MDS の失敗を注入します。開発者向けの使用のみ。

**タイプ**

32 ビット整数

**デフォルト**

**0**

**mds\_kill\_export\_at****詳細**

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。

**タイプ**

32 ビット整数

**デフォルト**

**0**

**mds\_kill\_import\_at****詳細**

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。

**タイプ**

32 ビット整数

**デフォルト**

0

**mds\_kill\_link\_at****詳細**

Ceph は、MDS をハードリンクコードに注入します。開発者向けの使用のみ。

**タイプ**

32 ビット整数

**デフォルト**

0

**mds\_kill\_rename\_at****詳細**

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。

**タイプ**

32 ビット整数

**デフォルト**

0

**mds\_wipe\_sessions****詳細**

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。

**タイプ**

ブール値

**デフォルト**

0

**mds\_wipe\_ino\_prealloc****詳細**

Ceph deletea inode の事前割り当てメタデータテストの目的でのみ使用してください。

**タイプ**

ブール値

**デフォルト**

0

**mds\_skip\_ino****詳細**

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。

**タイプ**

32 ビット整数

**デフォルト****0****mds\_standby\_for\_name****詳細**

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。

**タイプ**

文字列

**デフォルト**

該当なし

**mds\_standby\_for\_rank****詳細**

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。

**タイプ**

32 ビット整数

**デフォルト****-1****mds\_standby\_replay****詳細**

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。

**タイプ**

ブール値

**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

#### 詳細

ジャーナルヘッドオブジェクトを更新する頻度。

#### タイプ

整数

#### 必須

いいえ

#### デフォルト

15

### journaler\_prefetch\_periods

#### 詳細

ジャーナル再生に先行するストライプ期間の数。

#### タイプ

整数

#### 必須

いいえ

#### デフォルト

10

### journal\_prezero\_periods

#### 詳細

書き込み位置が 0 より進んだストライプ期間の数。

#### タイプ

整数

#### 必須

いいえ

#### デフォルト

10

### journaler\_batch\_interval

#### 詳細

人為的に発生する最大レイテンシー (秒単位)。

#### タイプ

double

#### 必須

いいえ

#### デフォルト

.001



---

**journaler\_batch\_max****詳細**

フラッシュを遅延させる最大バイト。

**タイプ**

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 クライアント設定の参照

本項では、Ceph File System (CephFS) FUSE クライアントの設定オプションについて説明します。Ceph 設定ファイルの **[client]** セクションで設定します。

### client\_acl\_type

#### 詳細

ACL タイプを設定します。現在、POSIX ACL を有効にする場合は **posix\_acl** または空の文字列のみが許可されます。このオプションは、**fuse\_default\_permissions** が **false** に設定されている場合にのみ有効になります。

#### タイプ

文字列

#### デフォルト

"" (ACL 実施なし)

### client\_cache\_mid

#### 詳細

クライアントキャッシュ mid ポイントを設定します。midpoint は、最も新しいリストをホットリストと warm リストに分割します。

#### タイプ

浮動小数点 (Float)

#### デフォルト

**0.75**

### client\_cache サイズ

#### 詳細

クライアントがメタデータキャッシュに保持する inode の数を設定します。

#### タイプ

整数

#### デフォルト

**16384** (16 MB)

### client\_caps\_release\_delay

#### 詳細

機能リリース間の遅延を秒単位で設定します。遅延は、別のユーザー空間操作に性能が必要な場合に、クライアントが機能に待機する秒数を設定します。

#### タイプ

整数

#### デフォルト

**5** (秒)

### client\_debug\_force\_sync\_read

#### 詳細

**true** に設定すると、クライアントはローカルページキャッシュを使用する代わりに OSD から直接データを読み取ります。

**タイプ**

ブール値

**デフォルト****false****client\_dirsize\_rbytes****詳細**

**true** に設定した場合は、ディレクトリーの再帰的サイズ (つまりすべての上位の合計) を使用します。

**タイプ**

ブール値

**デフォルト****true****client\_max\_inline\_size****詳細**

RADOS の別のデータオブジェクトではなく、ファイル inode に保存されるインラインデータの最大サイズを設定します。この設定は、**inline\_data** フラグが MDS マップに設定されている場合にのみ該当します。

**タイプ**

整数

**デフォルト****4096****client\_metadata****詳細**

自動生成されたバージョン、ホスト名、およびその他のメタデータに加えて、各 MDS に送信されるクライアントメタデータのコンマ区切りの文字列。

**タイプ**

文字列

**デフォルト**

"" (追加のメタデータなし)

**client\_mount\_gid****詳細**

CephFS マウントのグループ ID を設定します。

**タイプ**

整数

**デフォルト****-1****client\_mount\_timeout****詳細**

CephFS マウントのタイムアウトを秒単位で設定します。

**タイプ**

浮動小数点 (Float)

**デフォルト****300.0****client\_mount\_uid****詳細**

CephFS マウントのユーザー ID を設定します。

**タイプ**

整数

**デフォルト****-1****client\_mountpoint****詳細****ceph-fuse** コマンドの **-r** オプションの代替手段です。**タイプ**

文字列

**デフォルト****/****client\_oc****詳細**

オブジェクトのキャッシュを有効にします。

**タイプ**

ブール値

**デフォルト****true****client\_oc\_max\_dirty****詳細**

オブジェクトキャッシュのダーティーバイトの最大数を設定します。

**タイプ**

整数

**デフォルト****104857600** (100MB)**client\_oc\_max\_dirty\_age****詳細**

ライトバックの前に、オブジェクトキャッシュ内のダーティーデータの最大期間を秒単位で設定します。

**タイプ**

浮動小数点 (Float)

**デフォルト****5.0 (秒)****client\_oc\_max\_objects****詳細**

オブジェクトキャッシュ内のオブジェクトの最大数を設定します。

**タイプ**

整数

**デフォルト****1000****client\_oc\_size****詳細**

クライアントキャッシュがデータのバイト数を設定します。

**タイプ**

整数

**デフォルト****209715200** (200 MB)**client\_oc\_target\_dirty****詳細**

ダーティーデータのターゲットサイズを設定します。Red Hat は、この数を少ない状態に維持することを推奨します。

**タイプ**

整数

**デフォルト****8388608** (8MB)**client\_permissions****詳細**

すべての I/O 操作でクライアントパーミッションを確認します。

**タイプ**

ブール値

**デフォルト****true****client\_quota\_df****詳細**

**statfs** 操作のルートディレクトリーのクォータを報告します。

**タイプ**

ブール値

**デフォルト****true**

## client\_readahead\_max\_bytes

### 詳細

カーネルが将来の読み取り操作のために読み取る最大バイト数を設定します。**client\_readahead\_max\_periods** 設定で上書きされます。

### タイプ

整数

### デフォルト

0 (無制限)

## client\_readahead\_max\_periods

### 詳細

カーネルが読み取るファイルレイアウト期間 (オブジェクトサイズ \* ストライプの数) を設定します。**client\_readahead\_max\_bytes** 設定を上書きします。

### タイプ

整数

### デフォルト

4

## client\_readahead\_min

### 詳細

カーネルが読み取る最小バイトを設定します。

### タイプ

整数

### デフォルト

131072 (128KB)

## client\_snapdir

### 詳細

スナップショットディレクトリ名を設定します。

### タイプ

文字列

### デフォルト

".snap"

## client\_tick\_interval

### 詳細

機能の更新とその他の upkeep の間隔を秒単位で設定します。

### タイプ

浮動小数点 (Float)

### デフォルト

1.0

## client\_use\_random\_mds

### 詳細

各リクエストにランダムな MDS を選択します。

#### タイプ

ブール値

#### デフォルト

**false**

### fuse\_default\_permissions

#### 詳細

**false** に設定すると、**ceph-fuse** ユーティリティーは FUSE のパーミッションの適用に依存せずに独自のパーミッションチェックを行います。**client acl type=posix\_acl** オプションとともに **false** に設定して POSIX ACL を有効にします。

#### タイプ

ブール値

#### デフォルト

**true**



### 開発者オプション

これらのオプションは内部です。これは、オプションのリストを完了するためだけにリストされています。

### client\_debug\_getattr\_caps

#### 詳細

MDS からの応答に必要な機能が含まれているかどうかを確認します。

#### タイプ

ブール値

#### デフォルト

**false**

### client\_debug\_inject\_tick\_delay

#### 詳細

クライアントティックの間に人為的な遅延を追加します。

#### タイプ

整数

#### デフォルト

**0**

### client\_inject\_fixed\_oldest\_tid

#### 詳細, タイプ

ブール値

#### デフォルト

**false**

### client\_inject\_release\_failure

**詳細, タイプ**

ブール値

**デフォルト****false****client\_trace****詳細**

すべてのファイル操作のトレースファイルへのパス。出力は、Ceph の合成クライアントが使用するよう設計されています。詳細は、**ceph-syn(8)** man ページを参照してください。

**タイプ**

文字列

**デフォルト**

"" (無効)