Red Hat Ceph Storage 5

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

Last Updated: 2022-01-12

Red Hat Ceph Storage 5 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

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律上の通知

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 プロパティー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

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。

スケーラビリティー

Ceph File Systemは、メタデータサーバ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ーを実現しています。

共有ファイルシステム

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

複数のファイルシステム

Red Hat Ceph Storage 5 からは、1つのストレージクラスタで複数のファイルシステムをアクティブにすることができます。各 CephFS には、独自のプールのセットと、独自のメタデータサーバー (MDS) ランクのセットがあります。複数のファイルシステムを展開する場合は、MDS デーモンの数を増やす必要があります。これにより、メタデータのスループットを向上させることができますが、同時に運用コストも増加します。また、特定のファイルシステムへのクライアントのアクセスを制限することもできます。

高可用性

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

クライアントクオータ

Ceph File System は、システム内のあらゆるディレクトリにクォータを設定することをサポートしています。クオータは、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 クライアントクオータはデフォルトで有効です。

関連情報

- Ceph Metadata Server をインストールするには、『オペレーションガイド』 の「Ceph Orchestrator を使用した MDS サービスの管理 」セクションを参照してください。
- Ceph File System を作成するには、**ファイルシステムガイド**のCepth File System の導入セクションを参照してください。

1.2. CEPH FILE SYSTEM のコンポーネント

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

Clients

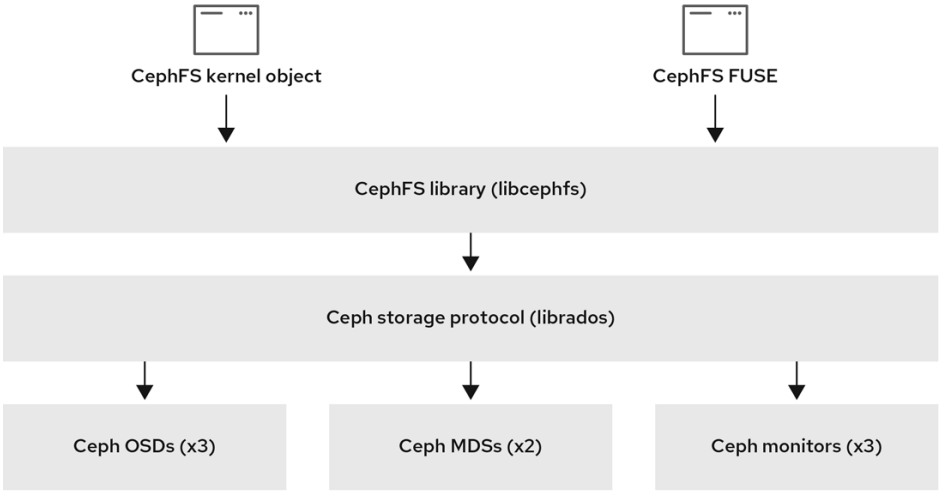
CephFS クライアントは、FUSE クライアントの **ceph-fuse** やカーネルクライアントの **kcephfs** など、CephFS を使用するアプリケーションの代わりに I/O 操作を行います。CephFS クライアントは、アクティブな Metadata Server にメタデータの要求を送信します。その代わり、CephFS クライアントはファイルのメタデータ認識し、メタデータとファイルデータの両方を安全にキャッシュすることができます。

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、バッキング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。

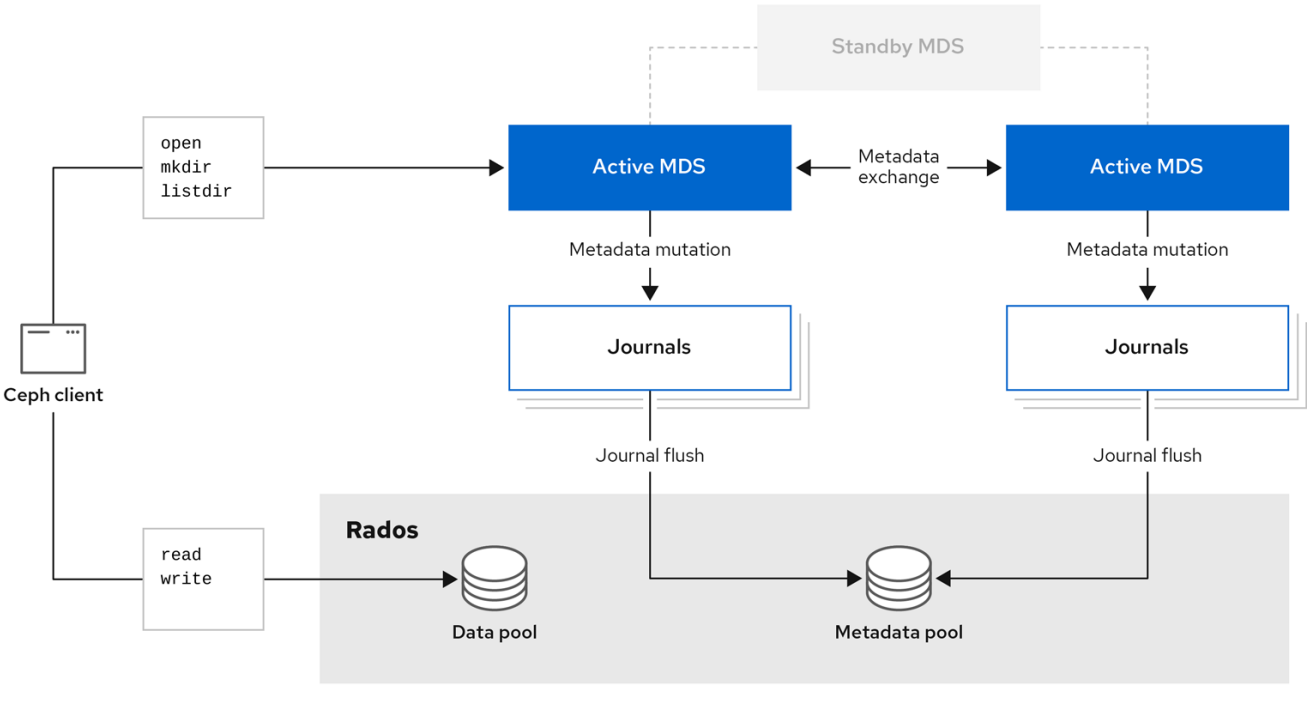
- Ceph OSD (**ceph-osd**) Ceph File System のデータとメタデータが格納されています。
- Ceph File System のメタデータを管理する Ceph Metadata Servers (**ceph-mds**)。
- クラスタマップのマスターコピーを管理する Ceph Monitors (**ceph-mon**)。

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サーバをインストールするには、**『ファイルシステムガイド』**の**「Ceph Orchestrator を使用した MDS サービスの管理」**セクションを参照してください。
- Ceph File System を作成するには、**ファイルシステムガイド**の**Red Hat Ceph File System の導入**セクションを参照してください。

1.3. CEPH FILE SYSTEM と SELINUX

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

関連情報

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

1.4. CEPH FILE SYSTEM の制限と POSIX 規格

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

- クライアントがファイルの書き込みに失敗した場合、書き込み操作は必ずしも Atomic ではありません。例えば、**O_SYNC** フラグで開かれた 8MB のバッファを持つファイルに対して、クライアントが **write()** システムコールを呼び出したところ、予期せぬ終了で、書き込み操作が部分的にしかできなくなってしまうことがあります。ローカルファイルシステムを含め、ほとんどのファイルシステムがこのような動作をします。
- 書き込み操作が同時に行われる状況では、オブジェクトの境界を超えた書き込み操作は必ずしも Atomic ではありません。例えば、ライター **A** が **"aa|aa"**、ライター **B** が **"bb|bb"** を同時に書いた場合、**"|"** はオブジェクトの境界であり、本来の**"aa|aa"** や **"bb|bb"** ではなく、**"aa|bb"** が書かれてしまいます。
- POSIX には **telldir()** や **seekdir()** というシステムコールがあり、カレントディレクトリのオフセットを取得して、そこまでシークすることができます。CephFS はいつでもディレクトリーを断片化できるため、ディレクトリーの安定した整数オフセットを返すことは困難です。そのため、0 以外のオフセットで **seekdir()** システムコールを呼び出しても、動作する場合がありますが、動作を保証するものではありません。**seekdir()** をオフセット 0 で呼び出すと必ず動作します。これは、**rewinddir()** システムコールと同等のものです。
- スパースファイルは、**stat()** システムコールの **st_blocks** フィールドに正しく伝わりませんでした。**st_blocks** フィールドには、ファイルサイズをブロックサイズで割った商が常に入力されているため、CephFS では、割り当てられたり書き込まれたりしたファイルの一部を明示的に追跡しません。この動作により、**du** などのユーティリティーが使用スペースを過大評価してしまいます。
- **mmap()** システムコールでファイルを複数のホストのメモリ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サーバをインストールするには、**『ファイルシステムガイド』**の**「Ceph Orchestrator を使用した MDS サービスの管理」**セクションを参照してください。
- Ceph File System を作成するには、**ファイルシステムガイド**の**Red Hat Ceph File System の導入**セクションを参照してください。

1.5. 関連情報

- Ceph Metadataサーバをインストールするには、**『ファイルシステムガイド』**の**「Ceph Orchestrator を使用した MDS サービスの管理」**セクションを参照してください。
- Red Hat OpenStack Platform で Ceph File System へのインターフェースとして NFS Ganesha を使用する場合、そのような環境を展開する方法については、**CephFS via NFS Back End Guide for Shared File System Service** の CephFS with NFS-Ganesha の展開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で、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 クラスター
- 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。

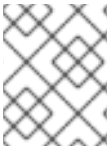
- Active: Ceph File System に保存されているファイルとディレクトリーのメタデータを管理します。
- Stanby: バックアップとして機能し、アクティブな MDS デーモンが反応しなくなったときにアクティブになります。

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 の構成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

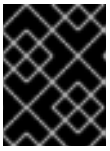
ランクには、以下の状態があります。

- **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 は、mds **_cache_memory_limit** に 8 GB から 64 GB までの値を推奨しています。キャッシュをより多く設定すると、リカバリーで問題が発生する可能性があります。この制限は、MDS の必要な最大メモリー使用の約 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. ファイルシステムアフィニティー

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

関連情報

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

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

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

本セクションでは、以下の管理タスクを説明します。

- **コマンドラインインターフェースを使用した MDS サービスのデプロイ**
- **サービス指定を使用した MDS サービスのデプロイ**
- **Ceph Orchestrator を使用した MDS サービスの削除**

2.6.1. 前提条件

- 実行中の Red Hat Ceph Storage クラスター。
- すべてのノードへの root レベルのアクセス。
- ホストはクラスターに追加される。
- すべてのマネージャー、モニター、および OSD デーモンがデプロイされます。

2.6.2. コマンドラインインターフェースを使用した MDS サービスのデプロイ

Ceph Orchestrator を使用すると、コマンドラインインターフェースで **placement** 指定を使用して Metadata Server (MDS) サービスをデプロイできます。Ceph File System (CephFS) には 1 つ以上の MDS が必要です。

注記

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

前提条件

- 実行中の Red Hat Ceph Storage クラスター。
- ホストはクラスターに追加される。
- すべてのマネージャー、モニター、および OSD デーモンがデプロイされます。

手順

1. Cephadm シェルにログインします。

   例

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

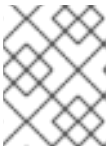
2. 配置指定を使用して MDS デーモンをデプロイする方法は 2 つあります。

方法 1

- **ceph fs volume** を使用して MDS デーモンを作成します。これにより、CephFS ボリューム、CephFS に関連付けられたプールが作成され、ホスト上で MDS サービスも起動します。

  構文

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

  注記

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

  例

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

方法 2

- プール、CephFS を作成してから、配置指定を使用して MDS サービスをデプロイします。

  a. CephFS のプールを作成します。

  構文

  ```
  ceph osd pool create DATA_POOL
  ceph osd pool create METADATA_POOL
  ```

例

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

b. データプールおよびメタデータプールのファイルシステムを作成します。

構文

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

例

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

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

構文

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

例

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

検証

- サービスを一覧表示します。

  例

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

- CephFS のステータスを確認します。

  例

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

- ホスト、デーモン、およびプロセスを一覧表示します。

  構文

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

  例

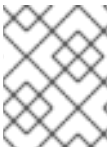
  ```
  [ceph: root@host01 /]# ceph orch ps --daemon_type=mds
  ```

関連情報

- Ceph ファイルシステム (CephFS) の作成に関する詳細は、『Red Hat Ceph Storage File System guide』を参照してください。

2.6.3. サービス指定を使用した MDS サービスのデプロイ

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



注記

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

前提条件

- 実行中の Red Hat Ceph Storage クラスター。
- ホストはクラスターに追加される。
- すべてのマネージャー、モニター、および OSD デーモンがデプロイされます。

手順

1. Cephadm シェルにログインします。

   例

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

2. 以下のディレクトリーに移動します。

   構文

   ```
   cd /var/lib/ceph/DAEMON_PATH/
   ```

   例

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

   注記

   ディレクトリー **mds** が存在しない場合は、作成します。

3. **mds.yml** ファイルを作成します。

   例

   ```
   [ceph: root@host01 mds/]# touch mds.yml
   ```

4. **mds.yml** ファイルを編集し、以下の詳細が含まれるようにします。

構文

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

例

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

5. サービス指定を使用して MDS サービスをデプロイします。

構文

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

例

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

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

構文

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

例

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

検証

- サービスを一覧表示します。

例

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

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

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

例

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

関連情報

- Ceph ファイルシステム (CephFS) の作成に関する詳細は、『Red Hat Ceph Storage File System guide』を参照してください。

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

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

前提条件

- 実行中の Red Hat Ceph Storage クラスター。
- すべてのノードへの root レベルのアクセス。
- ホストはクラスターに追加される。
- ホストにデプロイされた MDS デーモン 1 つ以上。

手順

1. MDS デーモンをクラスターから削除する方法は 2 つあります。

方法 1

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

  例

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

  b. 設定パラメーター **mon_allow_pool_delete** を **true** に設定します。

  例

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

  c. ファイルシステムを削除します。

  構文

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

例

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

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

方法 2

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

  a. サービスを一覧表示します。

  例

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

  b. サービスの削除

  構文

  ```
  ceph orch rm SERVICE_NAME
  ```

  例

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

検証

- ホスト、デーモン、およびプロセスを一覧表示します。

  構文

  ```
  ceph orch ps
  ```

  例

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

関連情報

- 詳細は、『Red Hat Ceph Storage Operations Guide』の「Deploying the MDS service using the command line interface」セクションを参照してください。
- 詳細は、『Red Hat Ceph Storage Operations Guide』の「Deploying the MDS service using the service specification」セクションを参照してください。

2.7. ファイルシステムのアフィニティーを設定する

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

前提条件

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

手順

1. Ceph File System の現在の状態を確認します。

   例

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

   Standby daemons:

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

2. ファイルシステムのアフィニティを設定します。

   構文

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

   例

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

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

   例

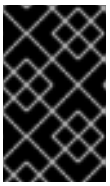
   ```
   [root@mon ~]# ceph fs dump
   dumped fsmap epoch 405
   e405
   ...
   Filesystem 'cephfs01' (27)
   ...
   ```

```
max_mds 1
in      0
up      {0=10420}
failed
damaged
stopped
...
[mds.b{0:10420} state up:active seq 274 join_fscid=27 addr
[v2:127.0.0.1:6856/2745199145,v1:127.0.0.1:6857/2745199145]] 1

Standby daemons:

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

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



重要

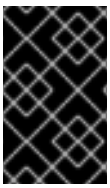
ファイルシステムがデグレードまたはアンダーサイズの状態である場合、ファイルシステムのアフィニティーを実施するためのフェイルオーバーは発生しません。

関連情報

- 詳細は、『**Red Hat Ceph Storage ファイルシステムガイド**』の「ファイルシステムのアフィニティー」セクションを参照してください。

2.8. 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 (MDS) デーモンを設定し、大規模システムのメタデータのパフォーマンスを拡張します。



重要

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

前提条件

- MDS ノードでの Ceph 管理機能。

手順

1. **max_mds** パラメータには、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にのみ、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。

2.9. 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。

2.10. STANDBY-REPLAY 用 METADATA SERVER の設定

各Ceph File System (CephFS) を構成して、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。

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。

2.11. エフェメラルピニングポリシー

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

Distributed (分散)

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

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

ランダム

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

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

デフォルトでは、これらのエフェメラルピンのポリシーはどちらも無効になっています。この機能は、**mds_export_ephemeral_distributed** または **mds_export_ephemeral_random** Ceph MDS 設定オプションのいずれかを設定することで有効になります。

関連情報

- ピンの手動設定の詳細については、**Red Hat Ceph Storage File System ガイド**のManually pinning directory trees to a particular rankセクションを参照してください。

2.12. ディレクトリツリーを特定のランクに手動で固定する

メタデータを特定の Ceph Metadata Server (MDS) ランクに明示的にマッピングすることで、ダイナミックバランサーをオーバーライドしたい場合もあります。これを手動で行うことで、アプリケーションの負荷を均等に分散したり、ユーザーのメタデータ要求が Ceph File System クラスターに与える影響を抑えることができます。ディレクトリを手動で固定することは、**ceph.dir.pin** 拡張属性を設定することで、エクスポートピンとも呼ばれます。

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

```
[root@client ~]# mkdir -p a/b 1
[root@client ~]# setfattr -n ceph.dir.pin -v 1 a/ 2
[root@client ~]# setfattr -n ceph.dir.pin -v 0 a/b 3
```

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

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- 実行中の Ceph File System
- CephFS クライアントへのルートレベルのアクセス。

手順

1. ディレクトリーにエクスポートピンを設定します。

   構文

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

   例

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

関連情報

- ピンの設定は、『Red Hat Ceph Storage File System Guide』の「エフェメラルピニングポリシー」セクションを参照してください。

2.13. 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

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

   構文

   ```
   ceph fs status NAME
   ```

   例

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

   +------+--------+-------+---------------+-------+-------+
   | Rank | State  |  MDS  |    Activity   |  dns  |  inos |
   +------+--------+-------+---------------+-------+-------+
   |  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    |
   +-------------+
   ```

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

   構文

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

   例

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

3. Ceph File System のステータスを監視して、ストレージクラスターが新しい **max_mds** 値を安定させるのを待機します。

4. アクティブな MDS デーモンの数を確認します。

構文

```
ceph fs status NAME
```

例

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

+------+--------+-------+---------------+-------+-------+
| Rank | State  |  MDS  |    Activity   |  dns  |  inos |
+------+--------+-------+---------------+-------+-------+
|  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。

2.14. 関連情報

- Red Hat Ceph Storage クラスターのインストールの詳細は、**『Red Hat Ceph Storage インストールガイド』**を参照してください。

第3章 CEPH FILE SYSTEM のデプロイメント

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

基本的に、デプロイメントワークフローは以下の 3 つのステップになります。

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

3.1. 前提条件

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

3.2. 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ケイパビリティーフラグは、アルファベット順に指定する必要があります。

レイアウトとクォータ

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

例

```
client.0
    key: AQAz7EVWygILFRAAdIcuJ10opU/JKyfFmxhuaw==
    caps: [mds] allow rwp
    caps: [mon] allow r
    caps: [osd] allow rw tag cephfs data=cephfs_a

client.1
    key: AQAz7EVWygILFRAAdIcuJ11opU/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: AQAz7EVWygILFRAAdIcuJ10opU/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: AQAz7EVWygILFRAAdIcuJ10opU/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プレフィックス長は 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 ノードで複数の Ceph File Systems (CephFS) を作成することができます。

前提条件

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

手順

1. Ceph ファイルシステムを作成します。

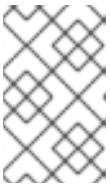
構文

```
ceph fs volume create FILE_SYSTEM_NAME
```

例

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

この手順を繰り返して、追加のファイルシステムを作成します。



注記

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

2. Ceph クライアントから新しい Ceph File System へのアクセスを確認します。

   a. Ceph クライアントが新しいファイルシステムへのアクセスを承認します。

      構文

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

      例

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

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

注記

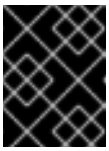
必要に応じて、**root_squash** オプションを指定することで安全対策を追加できます。これにより、**uid=0** または **gid=0** のクライアントが書き込み操作を行うのを許可することで、誤って削除のシナリオは阻止されますが、読み取り操作は引き続き許可されます。

例

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

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

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



重要

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

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

構文

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

例

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

c. Ceph クライアントノードで、新しいディレクトリーを作成します。

構文

```
mkdir PATH_TO_NEW_DIRECTORY_NAME
```

例

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

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

構文

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

例

```
[root@client ~]# ceph-fuse /mnt/cephfs/ -n client.1 --client-fs=cephfs01
ceph-fuse[555001]: starting ceph client
2021-04-07T07:33:27.158+0000 7f11feb81200 -1 init, newargv = 0x55fc4269d5d0
newargc=15
ceph-fuse[555001]: starting fuse
```

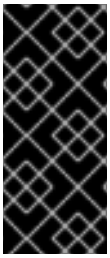
e. Ceph クライアントノードで、新しいマウントポイントのディレクトリーコンテンツを一覧表示するか、新しいマウントポイントにファイルを作成します。

関連情報

- 詳細は、『**Red Hat Ceph Storage File System ガイド**』の「Creating client users for a Ceph File System」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「Mounting the Ceph File System as a kernel client」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「Mounting the Ceph File System as a FUSE client」セクション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。
- 詳細は、『**Red Hat Ceph Storage ストラテジーガイド**』の「プール」の章を参照してください。

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

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



重要

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

前提条件

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

手順

1. CephFS 用のイレイジャーコーディングデータプールを作成します。

   構文

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

   例

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

2. プールが追加されたことを確認します。

   例

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

3. 消去コード化されたプールでのオーバーライトを有効にします。

   構文

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

   例

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

4. Ceph File System のステータスを確認します。

   構文

   ```
   ceph fs status FILE_SYSTEM_NAME
   ```

–

例

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

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

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

構文

```
ceph fs add_data_pool FILE_SYSTEM_NAME DATA_POOL_NAME
```

例

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

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

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

構文

```
ceph fs status FILE_SYSTEM_NAME
```

例

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

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

7. 新しいディレクトリーにファイルレイアウトを設定します。

構文

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

例

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

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

関連情報

- CephFS MDS 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。
- 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。
- 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

- Red Hat Ceph Storage クラスターが実行中である。
- 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
      ```

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

構文

```
scp 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
```

4. 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「Ceph 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 5 Tools リポジトリーを有効にします。

```
[root@client ~]# subscription-manager repos --enable=rhceph-5-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。

```
[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
```

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

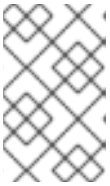
Red Hat Ceph Storage 4.1 の時点で、**mount.ceph** はキーリングファイルを直接読み取りできます。そのため、シークレットファイルは不要になりました。**name=CLIENT_ID** でクライアント ID を指定すると、**mount.ceph** は適切なキーリングファイルを検索します。

構文

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

例

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



注記

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

注記

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

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
```

自動マウント

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

構文

```
mkdir -p MOUNT_POINT
```

例

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

3. 以下のように **/etc/fstab** ファイルを編集します。

構文

```
#DEVICE              PATH        TYPE     OPTIONS              DUMP  FSCK
HOST_NAME:PORT,    MOUNT_POINT ceph     name=CLIENT_ID,       0     0
HOST_NAME:PORT,
ceph.client_mountpoint=/VOL/SUB_VOL_GROUP/SUB_VOL/UID_SUB_VOL,
HOST_NAME:PORT:/                    fs=FILE_SYSTEM_NAME,
                                [ADDITIONAL_OPTIONS]
```

最初の列は、Ceph Monitor ホスト名とポート番号を設定します。

2 列目は マウントポイントを設定します。

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

4 番目のコラム は、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。

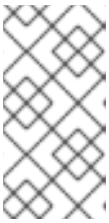
ネットワークサブシステムの開始後にファイルシステムがマウントされ、ハングやネットワークの問題を回避するために、**_netdev** オプションを設定します。アクセス時間情報が必要ない場合は、**noatime** オプションを設定するとパフォーマンスが向上します。

5 番目のコラムと 6 番目のコラム をゼロに設定します。

例

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

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



注記

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

注記

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

関連情報

- **mount(8)** man ページを参照してください。
- Ceph ユーザーの作成の詳細は、『**Red Hat Ceph Storage 管理ガイド**』の「Ceph ユーザー管理」の章を参照してください。
- 詳細は、『**Red Hat Ceph Storage 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 5 Tools リポジトリーを有効にします。

   ```
   [root@client ~]# subscription-manager repos --enable=rhceph-5-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。

   ```
   [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。

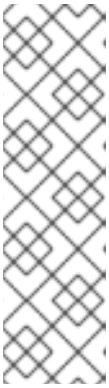
3. **ceph-fuse** ユーティリティーを使用して Ceph ファイルシステムをマウントします。

構文

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

例

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



注記

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

例

```
[root@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
```

注記

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

例

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

4. ファイルシステムが正常にマウントされていることを確認します。

構文

```
stat -f MOUNT_POINT
```

例

```
[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:/                                ceph.client_fs=FILE_SYSTEM_NAME,
                                        [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。アクセスする Ceph File System を指定するには、**ceph.client_fs** オプションを使用します。ネットワークサブシステムの開始後にファイルシステムがマウントされ、ハングやネットワークの問題を回避するために、**_netdev** オプションを設定します。アクセス時間情報が必要ない場合は、**noatime** オプションを設定するとパフォーマンスが向上します。エビクションの後に自動的に再接続する必要がある場合は、**client_reconnect_stale=true** オプションを設定します。

5 番目のコラムと 6 番目のコラム をゼロに設定します。

例

```
#DEVICE        PATH             TYPE         OPTIONS         DUMP  FSCK
mon1:6789,     /mnt/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:/                                  ceph.client_fs=cephfs01,
                                    _netdev,defaults
```

Ceph File System は、次回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

- Ceph Metadata Server をインストールするには、「Ceph Orchestrator を使用した MDS サービスの管理」 を参照してください。
- 詳細は、「Ceph ファイルシステムの作成」 を参照してください。
- 詳細は、「Ceph ファイルシステム用のクライアントユーザーの作成」 を参照してください。
- 詳細は、「Ceph File System のカーネルクライアントとしてのマウント」 を参照してください。
- 詳細は、「Ceph File System の FUSE クライアントとしてのマウント」 を参照してください。
- CephFS Metadata Server デーモンの設定に関する詳細は、2章*Ceph File System Metadata Server* を参照してください。

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

ストレージ管理者は、Red Hat の Ceph Container Storage Interface (CSI) を使用して Ceph File System (CephFS) エクスポートを管理できます。また、OpenStack のファイルシステムサービス (Manila) などの他のサービスは、一般的なコマンドラインインターフェー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

4.1. CEPH FILE SYSTEM ボリューム

ストレージ管理者は、Ceph File System (CephFS) ボリュームの作成、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。

4.1.1. 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。

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
   ```

4.1.2. 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。

手順

1. CephFS ボリュームを一覧表示します。

   例

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

4.1.3. 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。

手順

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

   例

   ```
   [root@mon ~]# ceph config set mon mon_allow_pool_delete true
   ```

2. CephFS ボリュームを削除します。

   構文

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

   例

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

4.2. CEPH FILE SYSTEM サブボリュームグループ

ストレージ管理者は、Ceph File System (CephFS) サブボリュームグループの作成、一覧表示、取得、および削除できます。CephFS サブボリュームグループは、サブボリュームのセット全体で、ファイルレイアウトなどのポリシーに影響を与えるディレクトリーレベルで抽象化されます。

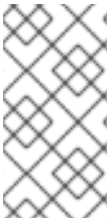
Red Hat Ceph Storage 5.0 以降では、サブボリュームグループスナップショット機能はサポートされません。これらのサブボリュームグループの既存のスナップショットのみを一覧表示および削除できます。

本セクションでは、以下を行う方法を説明します。

- ファイルシステムのサブボリュームグループを作成します。
- ファイルシステム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。

4.2.1. ファイルシステムのサブボリュームグループの作成

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



注記

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

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。

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。

4.2.2. 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 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
   ```

4.2.3. 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 システム (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 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
   ```

4.2.4. 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 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

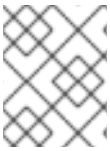
   ```
   ceph fs subvolumegroup snapshot ls VOLUME_NAME GROUP_NAME
   ```

   例

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

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

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。
- 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
   ```

4.2.6. 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、または存在しないと、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 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
```

4.3. CEPH FILE SYSTEM サブボリューム

ストレージ管理者は、Ceph File System (CephFS) サブボリュームの作成、一覧表示、取得、メタデータの取得、削除が可能です。また、これらのサブボリュームのスナップショットの作成、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。
- 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。

4.3.1. ファイルシステムのサブボリュームの作成

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

注記

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

前提条件

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

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。

4.3.2. 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 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
   ```

4.3.3. ファイルシステムのサブボリュームのサイズ変更

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

注記

ceph fs subvolume resize コマンドは、**new_size** で指定されたサイズでサブボリュームのクォータのサイズを変更します。**--no_shrink** フラグは、サブボリュームが現在使用されているサブボリュームのサイズ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 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
   ```

4.3.4. ファイルシステムのサブボリュームの絶対パスを取得中

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

前提条件

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

4.3.5. ファイルシステムのサブボリュームのメタデータの取得

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

前提条件

- Ceph File System がデプロイされている稼働中の Red Hat Ceph Storage クラスター。
- 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 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かを示します。
- **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」)

4.3.6. ファイルシステムのサブボリュームのスナップショットの作成

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

前提条件

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

- 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 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 _SNAP_NAME
   [--group_name GROUP_NAME]
   ```

   例

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

4.3.7. スナップショットからのサブボリュームのクローン作成

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

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。

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

  構文

  ```
  CLIENT_NAME
    key: AQAz7EVWygILFRAAdIcuJ12opU/JKyfFmxhuaw==
    caps: [mds] allow rw, allow rws path=DIRECTORY_PATH
    caps: [mon] allow r
    caps: [osd] allow rw tag cephfs data=DIRECTORY_NAME
  ```

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

  例

  ```
  client.0
    key: AQAz7EVWygILFRAAdIcuJ12opU/JKyfFmxhuaw==
    caps: [mds] allow rw, allow rws path=/bar
    caps: [mon] allow r
    caps: [osd] allow rw tag cephfs data=cephfs_a
  ```

手順

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

   構文

   ```
   ceph fs volume create VOLUME_NAME
   ```

   例

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

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

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

   構文

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

   例

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

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

構文

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

例

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

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。

4.3.8. ファイルシステムのサブボリュームのスナップショットの一覧表示

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 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
   ```

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

本項では、Ceph File 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 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",
   ```

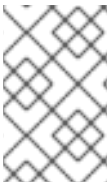
```
    "has_pending_clones": "no",
    "size": 0
}
```

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)。

4.3.10. ファイルシステムのサブボリュームの削除

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



注記

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

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

前提条件

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

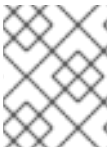
*NEW_SUBVOLUME: 以前に削除された同じサブボリュームにするか、新しいサブボリュームにクローンを作成します。

例

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

4.3.11. ファイルシステムのサブボリュームのスナップショットの削除

本セクションでは、Ceph File 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。
- 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
```

4.4. 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。

第5章 CEPH FILE SYSTEM 管理

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

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

5.1. 前提条件

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

5.2. CEPHFS-TOP ユーティリティーの使用

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

注記

現在、Red Hat Enterprise Linux 8 カーネルで、パフォーマンス統計がすべて利用できるわけではありません。

前提条件

- 正常かつ稼働中の Red Hat Ceph Storage クラスター
- Ceph File System のデプロイメント

- Ceph Monitor ノードへの root レベルのアクセス。

手順

1. Red Hat Ceph Storage 5 ツールリポジトリーが有効になっていない場合は、有効にします。

   例

   ```
   [root@mon ~]# subscription-manager repos --enable=rhceph-5-tools-for-rhel-8-x86_64-rpms
   ```

2. **cephfs-top** パッケージをインストールします。

   例

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

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

   例

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

4. Ceph ユーザー **client.fstop** を作成します。

   例

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

   注記

   オプションで 、**--id** 引数を使用して、**client.fstop** とは異なる Ceph ユーザーを指定します。

5. **cephfs-top** ユーティリティーを起動します。

   例

   ```
   [root@mon ~]# cephfs-top
   cephfs-top 0.0.1 - Tue Feb 17 16:54:04 2021
   Client(s): 2 - 1 FUSE, 0 kclient, 1 libcephfs

     CLIENT_ID  MOUNT_POINT  CAP_HIT(%)  READ_LATENCY(s)  WRITE_LATENCY(s)
   METADATA_LATENCY(s)  DENTRY_LEASE(%)  MOUNT_POINT@HOST/ADDR
     1244133    /          100.0       0.0              0.0              0.0                  0.0
   /mnt/cephfs@example/192.168.1.4
     1244143    /          100.0       0.0              0.0              0.01                 0.0
   N/A@example/192.168.1.4
   ```

注記

デフォルトでは、**cephfs-top** はストレージクラスター名 **ceph** に接続します。デフォルト以外のストレージクラスター名を使用するには、**cephfs-top** ユーティリティーで **--cluster NAME** オプションを使用できます。

5.3. MDS AUTOSCALER モジュールの使用

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

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

- **max_mds** ファイルシステムの設定
- **standby_count_wanted** ファイルシステムの設定

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

前提条件

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

手順

- MDS Autoscaler モジュールを有効にします。

  例

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

5.4. 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

5.5. 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

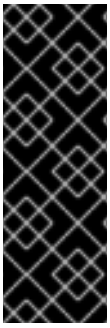
   ```
   [root@client ~]# fusermount -u /mnt/cephfs
   ```

関連情報

- **ceph-fuse(8)** man ページ

5.6. 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: AQBSdFhcGZFUDRAAcKhG9Cl2HPiDMMRv4DC43A==
     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 ガイド**の**Manually pinning directory trees to a particular rank**セクションを参照してください。
- 詳細は、『**Red Hat Ceph Storage ファイルシステムガイド**』の「**複数のアクティブな Metadata Server デーモンの設定**」セクションを参照してください。

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

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

前提条件

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

手順

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

  構文

  ```
  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。

5.8. データプールの追加

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

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

- 基本的なデータ分離。

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

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

前提条件

- Ceph Monitor ノードへのルートレベルのアクセス。

手順

1. 新しいデータプールを作成します。

   構文

   ```
   ceph osd pool create POOL_NAME
   ```

   以下を置き換えます。

   - **POOL_NAME** は、プールの名前に置き換えます。

   例

   ```
   [root@mon ~]# ceph osd pool create cephfs_data_ssd
   pool 'cephfs_data_ssd' created
   ```

2. メタデータサーバーの制御の下に、新たに作成されたプールを追加します。

   構文

   ```
   ceph fs add_data_pool FS_NAME POOL_NAME
   ```

   以下を置き換えます。

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

   たとえば、以下のようになります。

   ```
   [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、「**ファイルおよびディレクトリーのレイアウト**」を参照してください。
- 詳しくは、「**Ceph File System 用クライアントユーザーの作成**」を参照してください。

5.9. 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 クラスターを起動するには、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 クラスター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
```

注記

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

構文

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

例

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

5.10. CEPH ファイルシステムの削除

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

警告

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

前提条件

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

手順

1. ストレージクラスターに down のマークを付けます。

構文

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

置き換え

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

例

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

2. Ceph File System のステータス表示

```
ceph fs status
```

例

```
[root@mon]# ceph fs status
cephfs - 0 clients
======
+------------------+----------+-------+-------+
|       Pool       |   type   | used | avail |
+------------------+----------+-------+-------+
|cephfs.cephfs.meta | metadata | 31.5M |  52.6G|
|cephfs.cephfs.data |   data   |    0  |  52.6G|
+------------------+----------+-------+---------+
              STANDBY MDS
cephfs.ceph-host01
cephfs.ceph-host02
cephfs.ceph-host03
```

3. Ceph ファイルシステムを削除します。

構文

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

置き換え

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

例

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

4. ファイルシステムが正常に削除されたことを確認します。

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

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

関連情報

- 『Red Hat Ceph Storage 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。

5.11. 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 ファイルシステムガイド』の「アクティブなメタデータサーバーデーモンの数の低減」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「スタンバイデーモンの数の設定」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「メタデータサーバーのランク」を参照してください。

5.12. クライアント機能

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

重要

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

fs features ls コマンドを使用して、すべての CephFS 機能を一覧表示できます。**fs required_client_features** コマンドを使用して要件を追加または削除できます。

構文

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

機能の説明

reply_encoding

詳細

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

reclaim_client

詳細

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

lazy_caps_wanted

詳細

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

multi_reconnect

詳細

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

deleg_ino

詳細

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

metric_collect

詳細

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

alternate_name

詳細

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

5.13. CEPH FILE SYSTEM クライアントのエビクション

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

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

自動エビクション

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。

5.14. ブロックリスト CEPH FILE SYSTEM クライアント

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

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

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

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

注記

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

5.15. 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
```

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

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

重要

blocklist から CephFS クライアントを削除すると、データの整合性はリスクに伴います。また、その結果、CephFS クライアントが完全に確実に機能することを保証することはありません。エビクション後に完全に正常な CephFS クライアントを取得するには、CephFS クライアントをアンマウントして新規マウントを行うのが最適です。その他の CephFS クライアントが、拒否リストされた CephFS クライアントによってバッファーされた 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. ブロックリストを確認します。

例

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

2. ブロックリストから CephFS クライアントを削除します。

構文

```
ceph osd blocklist rm CLIENT_NAME_OR_IP_ADDR
```

例

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

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

```
recover_session=clean
```

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

```
client_reconnect_stale=true
```

関連情報

- 詳細は、『**Red Hat Ceph Storage File System ガイド**』の「Mounting the Ceph File System as a FUSE client」を参照してください。

5.17. NFS プロトコルを介した CEPH FILE SYSTEM 名前空間のエクスポート

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

重要

Red Hat は、NFS バージョン 4.0 以降のみをサポートします。

重要

Red Hat は、NFS エクスポートでの CephFS スナップショットの作成をサポートしていません。

前提条件

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

手順

1. Ceph Manager の NFS モジュールを有効にします。

例

```
[root@mon ~]# ceph mgr module enable nfs
```

2. NFS Ganesha クラスターを作成します。

構文

```
ceph nfs cluster create CLUSTER_NAME "NODE_PLACEMENT_LIST"
```

例

```
[root@mon ~]# ceph nfs cluster create nfs-cephfs "node1.example.com,node2.example.com"
NFS Cluster Created Successfully
```

この例では、NFS Ganesha クラスター名は **nfs-cephfs** で、デーモンコンテナーは **node1.example.com** および **node2.example.com** にデプロイされます。

重要

Red Hat は、1 つのノードにつき 1 つの NFS Ganesha デーモンのみをサポートします。

a. NFS Ganesha クラスターを削除するには、以下を実行します。

構文

```
ceph nfs cluster delete CLUSTER_NAME
```

例

```
[root@mon ~]# ceph nfs cluster delete nfs-cephfs
NFS Cluster Deleted Successfully
```

3. NFS-Ganesha クラスター情報を検証します。

構文

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

例

```
[root@mon ~]# ceph nfs cluster info nfs-cephfs
{
   "nfs-cephfs": [
      {
         "hostname": "node1.example.com",
         "ip": [
            "2620:52:0:880:225:90ff:fefc:2a2e",
            "10.8.128.21"
         ],
```

```
            "port": 2049
        },
        {
            "hostname": "node2.example.com",
            "ip": [
                "2620:52:0:880:225:90ff:fefc:2c3c",
                "10.8.128.22"
            ],
            "port": 2049
        }
    ]
}
```

注記

CLUSTER_NAME の指定 は任意です。

4. CephFS エクスポートを作成します。

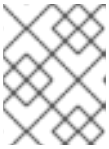
構文

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

例

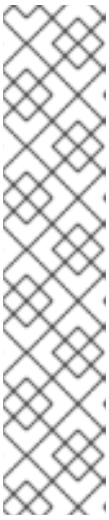
```
[root@mon ~]# ceph nfs export create cephfs cephfs01 nfs-cephfs /ceph --path=/
```

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



注記

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



注記

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

構文

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

例

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

a. 擬似 root 名に基づいてエクスポートブロックを表示するには、次のコマンドを実行します。

構文

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

例

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

b. CephFS エクスポートを削除するには、以下を実行します。

構文

```
ceph nfs export delete CLUSTER_NAME BINDING
```

例

```
[root@mon ~]# ceph nfs export delete nfs-cephfs /ceph
```

5. NFS Ganesha 設定をカスタマイズします。

a. 新しい Ceph ユーザーを作成します。

構文

```
ceph auth get-or-create client.USER_NAME mon 'allow r' osd 'allow rw pool=nfs-ganesha namespace=CLUSTER_NAME, allow rw tag cephfs data=FILE_SYSTEM_NAME' mds 'allow rw path=PATH_WITHIN_CEPHFS'
```

例

```
[root@mon ~]# ceph auth get-or-create client.nfstest01 mon 'allow r' osd 'allow rw
pool=nfs-ganesha namespace=nfs-cephfs, allow rw tag cephfs data=cephfs01' mds
'allow rw path=/'
```

b. 新しい設定ファイルを作成します。

構文

```
touch PATH_TO_CONFIG_FILE
```

例

```
[root@mon ~]# touch /root/nfs-cephfs.conf
```

c. カスタムエクスポートブロックを追加して新しい設定ファイルを開いて編集します。

構文

```
EXPORT {
  Export_Id = NUMERICAL_ID;
  Transports = TCP;
  Path = PATH_WITHIN_CEPHFS;
  Pseudo = BINDING;
  Protocols = 4;
  Access_Type = PERMISSIONS;
  Attr_Expiration_Time = 0;
  Squash = None;
  FSAL {
    Name = CEPH;
    Filesystem = "FILE_SYSTEM_NAME";
    User_Id = "USER_NAME";
    Secret_Access_Key = "USER_SECRET_KEY";
  }
}
```

例

```
EXPORT {
  Export_Id = 100;
  Transports = TCP;
  Path = /;
  Pseudo = /ceph/;
  Protocols = 4;
  Access_Type = RW;
  Attr_Expiration_Time = 0;
  Squash = None;
  FSAL {
    Name = CEPH;
    Filesystem = "cephfs01";
    User_Id = "nfstest01";
```

```
    Secret_Access_Key = "AQD9aW1g1UWmCxAAxZTW8YKeVi4sF5X+5ehH4Q==";
  }
}
```

注記

必要に応じて、以下のブロックを使用して、設定ファイルでロギングを有効にすることもできます。

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

d. 新しい設定を行います。

構文

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

例

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

e. NFS エクスポートを一覧表示します。

構文

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

例

```
[root@mon ~]# ceph nfs export ls nfs-cephfs
[
  "/ceph/"
]
[root@mon ~]#
[root@mon ~]# ceph nfs export ls nfs-cephfs --detailed
[
  {
    "export_id": 100,
    "path": "/",
    "cluster_id": "nfs-cephfs",
    "pseudo": "/ceph/",
    "access_type": "RW",
    "squash": "no_root_squash",
    "security_label": true,
    "protocols": [
      4
    ],
    "transports": [
```

```
      "TCP"
    ],
    "fsal": {
      "name": "CEPH",
      "user_id": "nfstest01",
      "fs_name": "cephfs",
      "sec_label_xattr": ""
    },
    "clients": []
  }
]
```

f. カスタム設定を削除するには、以下を実行します。

構文

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

例

```
[root@mon ~]# ceph nfs cluster config reset nfs-cephfs -i /root/nfs-cephfs.conf
```

6. クライアントノードで、エクスポートした Ceph File System をマウントします。

構文

```
mount -t nfs -o port=GANESHA_PORT NODE_NAME:_BINDING_ LOCAL_MOUNT_POINT
```

例

```
[root@client ~]# mount -t nfs -o port=2049 node1:/ceph/ /mnt/nfs-cephfs
```

a. システムの起動時に自動的にマウントするには、改行を追加して **/etc/fstab** ファイルを開いて編集します。

構文

```
NODE_NAME:_BINDING_   LOCAL_MOUNT_POINT   nfs4
defaults,seclabel,vers=4.2,proto=tcp,port=2049  0  0
```

例

```
node1.example.com:/ceph/   /mnt/nfs-cephfs   nfs4
defaults,seclabel,vers=4.2,proto=tcp,port=2049  0  0
```

関連情報

- 詳細は、『Red Hat Ceph Storage File System ガイド』の「Mounting the Ceph File System as a kernel client」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「Mounting the Ceph File System as a FUSE client」セクションを参照してください。

5.18. CEPH FILE システムのクォータ

ストレージ管理者は、ファイルシステム内の任意のディレクトリーでクォータを表示、設定、および削除できます。クォータの制限は、バイト数またはディレクトリー内のファイル数に配置できます。

5.18.1. 前提条件

- 実行中、および正常な Red Hat Ceph Storage クラスター
- Ceph File System のデプロイメント
- **attr** パッケージがインストールされていることを確認します。

5.18.2. Ceph File システムのクォータ

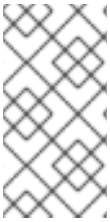
Ceph File System (CephFS) のクォータにより、ディレクトリー構造に保存されたファイルの数またはバイト数を制限できます。Ceph File System のクォータは、FUSE クライアントまたはカーネルクライアント (バージョン 4.17 以降) を使用して完全にサポートされます。

制限

- CephFS のクォータは、設定された制限に達するとデータの書き込みを停止するためにファイルシステムをマウントするクライアントとの協調に依存しています。ただし、クォータのみでは、信頼できないクライアントがファイルシステムを埋めないようにすることはできません。
- ファイルシステムにデータを書き込むプロセスが、設定された制限に到達したら、データ量がクォータ制限を超えるか、プロセスがデータの書き込みを停止するまでの短い期間が長くなります。通常、期間 (秒) は数十秒で測定されます。ただし、プロセスは、その期間中データの書き込みを続けます。プロセスが書き込む追加データ量は、停止前の経過時間によって異なります。
- パスベースのアクセス制限を使用する場合は、クライアントが制限されているディレクトリーのクォータを設定するか、その下でネスト化されたディレクトリーにクォータを設定してください。クライアントが MDS 機能に基づいて特定のパスへのアクセス制限があり、そのクォータがクライアントにアクセスできない上位ディレクトリーに設定されている場合、クライアントはクォータを強制しません。たとえば、クライアントが **/home/** ディレクトリーにアクセスできず、クォータが **/home/** で設定されている場合、クライアントは **/home/user/** ディレクトリーのクォータを強制できません。
- 削除または変更されたスナップショットファイルデータは、クォータに対してカウントされません。
- **setxattr** の使用時、NFS クライアントを使用したクォータのサポートはありません。NFS のファイルレベルのクォータはサポートされません。NFS 共有でクォータを使用するには、サブボリュームをエクスポートし、**--size** オプションを設定します。

5.18.3. 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

- Ceph クライアントノードへの root レベルのアクセス。
- **attr** パッケージがインストールされていることを確認します。

手順

1. CephFS クォータを表示するには、以下を実行します。
   a. バイト制限クォータの使用:

      構文

      ```
      getfattr -n ceph.quota.max_bytes DIRECTORY
      ```

      例

      ```
      [root@client ~]# getfattr -n ceph.quota.max_bytes /mnt/cephfs/
      getfattr: Removing leading '/' from absolute path names
      # file: mnt/cephfs/
      ceph.quota.max_bytes="100000000"
      ```

      この例では、**100000000** は 100 MB となります。

   b. ファイル制限クォータの使用:

      構文

      ```
      getfattr -n ceph.quota.max_files DIRECTORY
      ```

      例

      ```
      [root@client ~]# getfattr -n ceph.quota.max_files /mnt/cephfs/
      getfattr: Removing leading '/' from absolute path names
      # file: mnt/cephfs/
      ceph.quota.max_files="10000"
      ```

      この例では **10000** は 10,000 ファイルと等しくなります。

関連情報

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

5.18.4. 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

- Ceph クライアントノードへの root レベルのアクセス。
- **attr** パッケージがインストールされていることを確認します。

手順

1. CephFS クォータを設定します。
   a. バイト制限クォータの使用:

      構文

      ```
      setfattr -n ceph.quota.max_bytes -v 100000000 DIRECTORY
      ```

      例

      ```
      [root@client ~]# 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 DIRECTORY
      ```

      例

      ```
      [root@client ~]# 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。

5.18.5. 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

- Ceph クライアントノードへの root レベルのアクセス。
- **attr** パッケージがインストールされていることを確認します。

手順

1. CephFS クォータを削除するには、以下のコマンドを実行します。

   a. バイト制限クォータの使用:

      構文

      ```
      setfattr -n ceph.quota.max_bytes -v 0 DIRECTORY
      ```

      例

      ```
      [root@client ~]# setfattr -n ceph.quota.max_bytes -v 0 /mnt/cephfs/
      ```

   b. ファイル制限クォータの使用:

      構文

      ```
      setfattr -n ceph.quota.max_files -v 0 DIRECTORY
      ```

      例

      ```
      [root@client ~]# setfattr -n ceph.quota.max_files -v 0 /mnt/cephfs/
      ```

関連情報

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

5.18.6. 関連情報

- **ファイルシステムガイド**の**Red Hat Ceph File System の導入**セクションを参照してください。
- 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。

5.19. ファイルとディレクトリーのレイアウト

ストレージ管理者は、ファイルまたはディレクトリーのデータがオブジェクトにマップされる方法を制御できます。

本セクションでは、以下を行う方法を説明します。

- ファイルおよびディレクトリーのレイアウトの理解
- ファイルおよびディレクトリーのレイアウトの設定
- ファイルとディレクトリーのレイアウトフィールドの表示
- 個別のレイアウトフィールドの表示
- ディレクトリーレイアウト削除

5.19.1. 前提条件

- 実行中、および正常な Red Hat Ceph Storage クラスター
- Ceph File System のデプロイメント
- **attr** パッケージのインストール

5.19.2. 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。

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。

5.19.3. 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。

5.19.4. 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

- Red Hat Ceph Storage クラスターが実行中である。
- 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

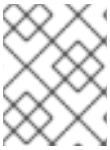
   ```
   getfattr -n ceph.TYPE.layout PATH
   ```

   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。

5.19.5. 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

- Red Hat Ceph Storage クラスターが実行中である。
- 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

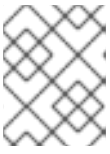
   ```
   getfattr -n ceph.TYPE.layout.FIELD _PATH
   ```

   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

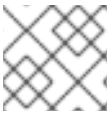
   pool フィールドのプールは、名前で示されます。ただし、新規作成されたプールは ID で識別できます。

関連情報

- **getfattr(1)** man ページ
- 詳細は、『**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**」セクションを参照してください。

5.19.6. 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

5.19.7. 関連情報

- **ファイルシステムガイド**の**Red Hat Ceph File System の導入**セクションを参照してください。
- 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。

5.20. CEPH ファイルシステムのスナップショット

ストレージ管理者は、Ceph File System (CephFS) ディレクトリーの特定の時点のスナップショットを取得できます。CephFS スナップショットは非同期で、作成するディレクトリースナップショットを選択できます。

5.20.1. 前提条件

- 実行中、および正常な Red Hat Ceph Storage クラスター
- Ceph File System のデプロイメント

5.20.2. Ceph ファイルシステムのスナップショット

Ceph File System (CephFS) のスナップショット機能は、新しい Ceph File Systems でデフォルトで有効になっていますが、既存の Ceph File Systems で手動で有効にする必要があります。CephFS スナップショットは、Ceph File System のイミュータブルな、ポイントインタイムビューを作成します。CephFS スナップショットは非同期で、CephFS ディレクトリーの特別な非表示ディレクトリー (**.snap**) に保存されます。Ceph ファイルシステム内の任意のディレクトリーのスナップショット作成を指定できます。ディレクトリーを指定すると、スナップショットにはその中のすべてのサブディレクトリーも含まれます。

警告

各 Ceph Metadata Server (MDS) クラスターは snap 識別子を別個に割り当てます。1 つのプールを共有する複数の Ceph ファイルシステムのスナップショットを使用すると、スナップショットの競合が発生し、ファイルデータがありません。

関連情報

- 詳細は、『**Red Hat Ceph Storage File System ガイド**』の「**Ceph File System のスナップショットの作成**」セクションを参照してください。
- 詳細は、『**Red Hat Ceph Storage File System ガイド**』の「**Cepth File System のスナップショットスケジュールの作成**」セクションを参照してください。

5.20.3. Ceph ファイルシステムのスナップショットの作成

Ceph File System (CephFS) のイミュータブル (ポイントインタイムビュー) を作成するには、スナップショットを作成します。

注記

新しい Ceph ファイルシステムの場合、スナップショットはデフォルトで有効になります。

前提条件

- 実行中、および正常な Red Hat Ceph Storage クラスター
- Ceph File System のデプロイメント

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

手順

1. 既存の Ceph ファイルシステムの場合は、スナップショット化機能を有効にします。

   構文

   ```
   ceph fs set FILE_SYSTEM_NAME allow_new_snaps true
   ```

   例

   ```
   [ceph: root@mds ~]# ceph fs set cephfs01 allow_new_snaps true
   ```

2. **.snap** ディレクトリーに新しい snapshot サブディレクトリーを作成します。

   構文

   ```
   mkdir NEW_DIRECTORY_PATH
   ```

   例

   ```
   [ceph: root@mds ~]# mkdir /.snap/new-snaps
   ```

   この例では **new-snaps** サブディレクトリーを作成し、これにより Ceph Metadata Server (MDS) がスナップショットの作成を開始するように通知します。

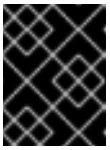
   a. スナップショットを削除するには、以下のコマンドを実行します。

      構文

      ```
      rmdir NEW_DIRECTORY_PATH
      ```

      例

      ```
      [ceph: root@mds ~]# rmdir /.snap/new-snaps
      ```

      重要

      基礎となるスナップショットが含まれる可能性のある root レベルのスナップショットの削除を試みると、失敗します。

関連情報

- 詳細は、『Red Hat Ceph Storage ファイルシステムガイド』の「Ceph File System スナップショットスケジュール」セクションを参照してください。
- 詳細は、『Red Hat Ceph Storage ファイルシステムガイド』の「Ceph File System スナップショット」セクションを参照してください。

5.20.4. Ceph ファイルシステムのスナップショットスケジュール

Ceph File System (CephFS) は、ファイルシステムディレクトリーのスナップショットをスケジュールできます。スナップショットのスケジューリングは Ceph Manager によって管理され、Python タイマーに依存します。スナップショットスケジュールデータは CephFS メタデータプールにオブジェクトとして保存され、ランタイム時に、スケジュールデータはすべてシリアライズされた SQLite データベースに置かれます。

重要

スケジューラーは、ストレージクラスターが通常の負荷がかかる場合に、スナップショットを個別に維持するために指定された時間に基づいて正確です。Ceph Manager の負荷が高い場合は、スナップショットがすぐにスケジュールされない可能性があり、スナップショットが若干遅延させる可能性があります。この場合、次のスケジュールされたスナップショットは遅延がないかのように動作します。遅延しているスケジュールされたスナップショットでは、スケジュール全体にドリフトが発生しません。

使用法

Ceph File System (CephFS) のスナップショットスケジューリングは、**snap_schedule** Ceph Manager モジュールにより管理されます。このモジュールは、スナップショットスケジュールを追加、クエリー、削除し、保持ポリシーを管理するインターフェースを提供します。このモジュールは **ceph fs snap-schedule** コマンドも 実装し、スケジュールを管理する複数のサブコマンドと保持ポリシーも実装します。すべてのサブコマンドは、CephFS ボリュームパスと subvolume パス引数を取り、複数の Ceph File Systems を使用する場合のファイルシステムパスを指定します。CephFS ボリュームパスを指定しない場合、引数は **fs_map** に一覧表示されている最初のファイルシステムに対してデフォルトで設定され、subvolume パス引数は何も指定しません。

スナップショットスケジュールは、ファイルシステムパス、繰り返しの間隔、および開始時間で識別されます。繰り返し間隔は、2 つの後続のスナップショットの間隔を定義します。間隔の形式は、数 + 時間指定 **h**(our)、**d**(ay)、**w**(eek) です。たとえば、間隔が **4h** であれば、4 時間ごとに 1 つのスナップショットが必要になります。開始時間は ISO 形式の文字列の値 **%Y-%m-%dT%H:%M:%S** です。指定されていない場合、開始時間は最後の midnight のデフォルト値を使用します。たとえば、デフォルトの開始時間値を使用して、スナップショットを **14:45** にスケジュールすると、繰り返される間隔が **1h** になると、最初のスナップショットは 15:00 に作成されます。

保持ポリシーは、ファイルシステムパスと保持ポリシーの仕様で識別されます。保持ポリシーの定義は、**COUNT TIME_PERIOD** の形式で、数字と時間指定のペア、または連結されたペアのいずれかで構成されます。このポリシーにより、スナップショットの数も保持され、スナップショットは少なくとも特定の期間にわたって保持されます。期間指定とは、**h(our)**、**d(ay)**、**w(eek)**、**m(onth)**、**y(ear)**、および **n** です。**n** の時間指定は特別な修飾子であり、タイミングに関係なくスナップショットの最後の数を保持します。たとえば、**4d** は、1 日以上経過した 4 つのスナップショットを保持します。

関連情報

- 詳細は、『**Red Hat Ceph Storage File System ガイド**』の「**Ceph File System のスナップショットの作成**」セクションを参照してください。
- 詳細は、『**Red Hat Ceph Storage File System ガイド**』の「**Ceph File System のスナップショットスケジュールの作成**」セクションを参照してください。

5.20.5. Ceph ファイルシステムのスナップショットスケジュールの作成

スナップショットスケジュールを追加、クエリー、および削除し、Ceph File System (CephFS) スナップショットの保持ポリシーを管理できます。単一パスについて異なるスケジュールを設定できます。繰り返される間隔と開始時間が異なる場合、スケジュールは異なると見なされます。スナップショットス

ケジュールは、存在しない CephFS パスに追加できます。CephFS パスには保持ポリシーを 1 つだけ指定できますが、保持ポリシーは複数のカウントタイムペアを持つことができます。

注記

スケジューラーモジュールが有効になると、**ceph fs snap-schedule** コマンドを実行すると、利用可能なサブコマンドと、その使用形式が表示されます。

前提条件

- 実行中、および正常な Red Hat Ceph Storage クラスター
- Ceph File System のデプロイメント
- Ceph Manager および Metadata Server (MDS) ノードへのルートレベルのアクセス。
- ファイルシステムで CephFS スナップショットを有効にします。

手順

1. Ceph Manager ノードで **snap_schedule** モジュールを有効にします。

   例

   ```
   [ceph: root@mon ~]# ceph mgr module enable snap_schedule
   ```

2. Ceph File System の新しいスケジュールを追加します。

   構文

   ```
   ceph fs snap-schedule add FILE_SYSTEM_VOLUME_PATH REPEAT_INTERVAL
   [START_TIME]
   ```

   例

   ```
   [ceph: root@mds ~]# ceph fs snap-schedule add /cephfs 4h 14:00
   ```

   この例では、CephFS /**cephfs** ボリュームのスナップショットスケジュールを作成し、4 時間ごとにスナップショットを作成して、**14:00** から開始します。

   a. 特定のスナップショットスケジュールを削除するには、以下を実行します。

      構文

      ```
      ceph fs snap-schedule remove FILE_SYSTEM_VOLUME_PATH [REPEAT_INTERVAL]
      [START_TIME]
      ```

      例

      ```
      [ceph: root@mds ~]# ceph fs snap-schedule remove /cephfs 4h 14:00
      ```

      この例では、4 時間ごとにスナップショットされ、**14:00** から始まる CephFS /**cephfs** ボリュームの特定のスナップショットスケジュールを削除します。

b. 特定の CephFS ボリュームのスナップショットスケジュールすべてを削除するには、以下を実行します。

構文

```
ceph fs snap-schedule remove FILE_SYSTEM_VOLUME_PATH
```

例

```
[ceph: root@mds ~]# ceph fs snap-schedule remove /cephfs
```

以下の例では、CephFS /**cephfs** ボリュームのすべてのスナップショットスケジュールを削除します。

3. CephFS ボリュームパスのスナップショット用に新たな保持ポリシーを追加します。

構文

```
ceph fs snap-schedule retention add FILE_SYSTEM_VOLUME_PATH
[COUNT_TIME_PERIOD_PAIR] TIME_PERIOD COUNT
```

例

```
[ceph: root@mds ~]# ceph fs snap-schedule retention add /cephfs h 14 (1)
[ceph: root@mds ~]# ceph fs snap-schedule retention add /cephfs d 4 (2)
[ceph: root@mds ~]# ceph fs snap-schedule retention add /cephfs 14h4w (3)
```

(1) この例では、14 スナップショットを 1 時間以上保持します。

(2) この例では、4 つのスナップショットを 1 日以上保持します。

(3) この例では、14 時間ごとに 1 週間、また 4 週間のスナップショットを保持します。

a. CephFS パスで保持ポリシーを削除するには、以下を実行します。

構文

```
ceph fs snap-schedule retention remove FILE_SYSTEM_VOLUME_PATH
[COUNT_TIME_PERIOD_PAIR] TIME_PERIOD COUNT
```

例

```
[ceph: root@mds ~]# ceph fs snap-schedule retention remove /cephfs h 4 (1)
[ceph: root@mds ~]# ceph fs snap-schedule retention remove /cephfs 14d4w (2)
```

この例では、4 時間のスナップショットを削除します。

この例では、14 日次とスナップショット 4 週間を削除します。

4. CephFS パスの新規スナップショットスケジュールをアクティブにします。

構文

```
ceph fs snap-schedule activate FILE_SYSTEM_VOLUME_PATH [REPEAT_INTERVAL]
```

例

```
[ceph: root@mds ~]# ceph fs snap-schedule activate /cephfs
```

以下の例では、CephFS /**cephfs** パスのすべてのスケジュールを有効にします。

5. CephFS パスのスナップショットスケジュールを無効にするには、以下を実行します。

構文

```
ceph fs snap-schedule deactivate FILE_SYSTEM_VOLUME_PATH [REPEAT_INTERVAL]
```

例

```
[ceph: root@mds ~]# ceph fs snap-schedule deactivate /cephfs 1d
```

この例では、CephFS /**cephfs** パスの最初のスナップショットを非アクティブにすることで、スナップショットの作成が一時停止します。

6. スナップショットスケジュールのステータスを確認します。

   a. スナップショットスケジュールを一覧表示します。

   構文

   ```
   ceph fs snap-schedule list FILE_SYSTEM_VOLUME_PATH [--format=plain|json] [--recursive=true]
   ```

   例

   ```
   [ceph: root@mds ~]# ceph fs snap-schedule list /cephfs --recursive=true
   ```

   この例では、ディレクトリーツリーのすべてのスケジュールを一覧表示しています。

   b. スナップショットスケジュールのステータスを確認します。

   構文

   ```
   ceph fs snap-schedule status FILE_SYSTEM_VOLUME_PATH [--format=plain|json]
   ```

   例

   ```
   [ceph: root@mds ~]# ceph fs snap-schedule status /cephfs --format=json
   ```

   以下の例では、CephFS /**cephfs** パスのスナップショットスケジュールのステータスを JSON 形式で表示しています。デフォルトの形式はプレーンテキストで、指定されていない場合はプレーンテキストになります。

関連情報

- 詳細は、**『Red Hat Ceph Storage ファイルシステムガイド』**の**「Ceph File System スナップショットスケジュール」**セクションを参照してください。
- 詳細は、**『Red Hat Ceph Storage ファイルシステムガイド』**の**「Ceph File System スナップショット」**セクションを参照してください。

5.20.6. 関連情報

- **ファイルシステムガイド**の**Red Hat Cepth File System の導入**セクションを参照してください。

5.21. CEPH FILE SYSTEM のミラー

ストレージ管理者は、別の Red Hat Ceph Storage クラスターのリモート Ceph File System に Ceph File System (CephFS) を複製できます。Ceph File System は、スナップショットディレクトリーの非同期レプリケーションをサポートします。

5.21.1. 前提条件

- ソースおよびターゲットストレージクラスターは、Red Hat Ceph Storage 5.0 以降を実行している必要があります。

5.21.2. Ceph File System ミラーリング

Ceph File System (CephFS) は、別の Red Hat Ceph Storage クラスター上のリモート Ceph File System へのスナップショットの非同期レプリケーションをサポートします。スナップショットの同期は、スナップショットデータをリモートの Ceph ファイルシステムにコピーし、同じ名前のリモートターゲットに新しいスナップショットを作成します。スナップショット同期用に特定のディレクトリーを設定できます。

CephFS ミラーの管理は、CephFS ミラーリングデーモン (**cephfs-mirror**) により実行されます。このスナップショットデータは、リモートの CephFS への一括コピーを実行することで同期されます。スナップショットのペアの同期順序は、**snap-id** を使用して作成時に決定されます。

重要

ハードリンクされたファイルは、別のファイルとして同期されます。

重要

Red Hat は、ストレージクラスター 1 つにつき 1 つの **cephfs-mirror** デーモンのみの実行をサポートします。

Ceph Manager モジュール

Ceph Manager **mirroring** モジュールは、デフォルトでは無効になっています。ディレクトリースナップショットミラーリングを管理するインターフェースを提供し、**cephfs-mirror** デーモンにディレクトリーを同期させる役割を果たします。Ceph Manager の **mirroring** モジュールは、ディレクトリースナップショットのミラーリングを制御するコマンドのファミリーも提供します。**mirroring** モジュールは、**cephfs-mirror** デーモンを管理しません。**cephfs-mirror** デーモンの停止、起動、再起動、および有効化は **systemctl** によって制御されますが、**cephadm** によって管理されます。

5.21.3. Ceph ファイルシステムのスナップショットミラーの設定

Ceph File System (CephFS) を設定して、リモートの Red Hat Ceph Storage クラスターの別の CephFS にスナップショットを複製するようにミラーリングできます。

注記

リモートストレージクラスターへの同期にかかる時間は、ファイルのサイズとミラーリングパス内のファイルの合計数によって異なります。

前提条件

- ソースおよびターゲットストレージクラスターは、Red Hat Ceph Storage 5.0 以降を実行していて正常である必要があります。
- ソースおよびターゲットストレージクラスターの Ceph Monitor ノードへのルートレベルのアクセス。
- 少なくとも 1 つの Ceph File System のデプロイメント

手順

1. ソースストレージクラスターで、CephFS ミラーリングデーモンをデプロイします。

   構文

   ```
   ceph orch apply cephfs-mirror ["NODE_NAME"]
   ```

   例

   ```
   [root@mon ~]# ceph orch apply cephfs-mirror "node1.example.com"
   Scheduled cephfs-mirror update...
   ```

   このコマンドにより、**cephfs-mirror** という名前の Ceph ユーザーが作成され、特定のノードに **cephfs-mirror** デーモンがデプロイされます。

2. ターゲットストレージクラスターで、それぞれの CephFS ピア用にユーザーを作成します。

   構文

   ```
   ceph fs authorize FILE_SYSTEM_NAME CLIENT_NAME / rwps
   ```

   例

   ```
   [root@mon ~]# ceph fs authorize cephfs client.mirror_remote / rwps
   [client.mirror_remote]
         key = AQCjZ5Jg739AAxAAxduIKoTZbiFJ0lgose8luQ==
   ```

3. ソースストレージクラスターで、CephFS ミラーリングモジュールを有効にします。

   例

   ```
   [root@mon ~]# ceph mgr module enable mirroring
   ```

4. ソースストレージクラスターで、Ceph File System でミラーリングを有効にします。

構文

```
ceph fs snapshot mirror enable FILE_SYSTEM_NAME
```

例

```
[root@mon ~]# ceph fs snapshot mirror enable cephfs
```

a. 任意。スナップショットミラーリングを無効にするには、以下のコマンドを使用します。

構文

```
ceph fs snapshot mirror disable FILE_SYSTEM_NAME
```

例

```
[root@mon ~]# ceph fs snapshot mirror disable cephfs
```

警告

ファイルシステムでスナップショットミラーリングを無効にすると、設定されたピアが削除されます。ピアをブートストラップして再度インポートする必要があります。

5. ターゲットピアストレージクラスターを準備します。

a. ターゲットノードで、**mirroring** Ceph Manager モジュールを有効にします。

例

```
[root@mon ~]# ceph mgr module enable mirroring
```

b. 同じターゲットノードで、ピアブートストラップを作成します。

構文

```
ceph fs snapshot mirror peer_bootstrap create FILE_SYSTEM_NAME CLIENT_NAME SITE_NAME
```

SITE_NAME は、ターゲットのストレージクラスターを識別するユーザー定義の文字列です。

例

```
[root@mon ~]# ceph fs snapshot mirror peer_bootstrap create cephfs
```

```
client.mirror_remote remote-site
{"token":
"eyJmc2lkIjogIjBkZjE3MjE3LWRmY2QtNDAzMC05MDc5LTM2Nzk4NTVkNDJlZiIsICJmaW
xlc3lzdGVtIjogImJhY2t1cF9mcyIsICJ1c2VyIjogImNsaWVudC5taXJyb3JfcGVlcl9ib290c3Ry
XAiLCAic2l0ZV9uYW1lIjogInNpdGUtcmVtb3RlIiwgImtleSI6ICJBUUFhcDBCZ0xtRmpOeEF
BVnNyZXozai9YYUV0T2UrbUJEZlJDZz09IiwgIm1vbl9ob3N0IjogIlt2MjoxOTIuMTY4LjAuN
To0MDkxOCx2MToxOTIuMTY4LjAuNTo0MDkxOV0ifQ=="}
```

次の手順で使用する二重引用符の間のトークン文字列をコピーします。

6. ソースストレージクラスターで、ターゲットストレージクラスターからブートストラップトークンをインポートします。

構文

```
ceph fs snapshot mirror peer_bootstrap import FILE_SYSTEM_NAME TOKEN
```

例

```
[root@mon ~]# ceph fs snapshot mirror peer_bootstrap import cephfs
eyJmc2lkIjogIjBkZjE3MjE3LWRmY2QtNDAzMC05MDc5LTM2Nzk4NTVkNDJlZiIsICJmaWxlc3lz
dGVtIjogImJhY2t1cF9mcyIsICJ1c2VyIjogImNsaWVudC5taXJyb3JfcGVlcl9ib290c3RyYXAiLCAi
2l0ZV9uYW1lIjogInNpdGUtcmVtb3RlIiwgImtleSI6ICJBUUFhcDBCZ0xtRmpOeEFBVnNyZXoza
i9YYUV0T2UrbUJEZlJDZz09IiwgIm1vbl9ob3N0IjogIlt2MjoxOTIuMTY4LjAuNTo0MDkxOCx2M
ToxOTIuMTY4LjAuNTo0MDkxOV0ifQ==
```

7. ソースストレージクラスターで、CephFS ミラーピアを一覧表示します。

構文

```
ceph fs snapshot mirror peer_list FILE_SYSTEM_NAME
```

例

```
[root@mon ~]# ceph fs snapshot mirror peer_list cephfs
```

a. 任意。スナップショットピアを削除するには、以下のコマンドを使用します。

構文

```
ceph fs snapshot mirror peer_remove FILE_SYSTEM_NAME PEER_UUID
```

例

```
[root@mon ~]# ceph fs snapshot mirror peer_remove cephfs a2dc7784-e7a1-4723-
b103-03ee8d8768f8
```

注記

この手順の「**追加リソース**」セクションの「ピア UUID 値の検索方法」の「**Ceph File System リンクのミラーステータスの表示**」を参照してください。

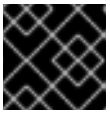
8. ソースストレージクラスターで、スナップショットミラーリングのディレクトリーを設定します。

構文

```
ceph fs snapshot mirror add FILE_SYSTEM_NAME PATH
```

例

```
[root@mon ~]# ceph fs snapshot mirror add cephfs /home/user1
```



重要

絶対パスのみが有効になります。

注記

Ceph Manager の **mirroring** モジュールは、パスを正規化します。たとえば、**/d1/d2/../dN** ディレクトリーは **/d1/d2** と同等です。ミラーリング用にディレクトリーが追加されると、その上位ディレクトリーおよびサブディレクトリーがミラーリング用に追加されなくなります。

a. 任意。ディレクトリーのスナップショットミラーリングを停止するには、以下のコマンドを使用します。

構文

```
ceph fs snapshot mirror remove FILE_SYSTEM_NAME PATH
```

例

```
[root@mon ~]# ceph fs snapshot mirror remove cephfs /home/user1
```

関連情報

- 詳細は、『**Red Hat Ceph Storage ファイルシステムガイド**』の「**Ceph File Systemのミラーステータスの表示**」セクションを参照してください。
- 詳細は、『**Red Hat Ceph Storage ファイルシステムガイド**』の「**Ceph ファイルシステムのミラーリング**」セクションを参照してください。

5.21.4. Ceph File システムのミラーステータスの表示

Ceph File System (CephFS) ミラーデーモン (**cephfs-mirror**) は、CephFS ミラーリングステータスの変更に関する非同期通知とピアの更新と共に行われます。コマンドとともに **cephfs-mirror** admin ソケットをクエリーして、ミラーステータスとピアステータスを取得できます。

前提条件

- ミラーリングが有効にされている Ceph File System のデプロイメントを少なくとも 1 つ。
- CephFS ミラーリングデーモンを実行するノードへのルートレベルのアクセス。

手順

1. Ceph File System ID を検索します。

   構文

   ```
   ceph --admin-daemon PATH_TO_THE_ASOK_FILE help
   ```

   例

   ```
   [root@mon ~]# ceph --admin-daemon /var/run/ceph/ceph-client.cephfs-
   mirror.node1.bndvox.asok help
   {
       ...
       "fs mirror peer status cephfs@11 1011435c-9e30-4db6-b720-5bf482006e0e": "get peer
   mirror status",
       "fs mirror status cephfs@11": "get filesystem mirror status",
       ...
   }
   ```

   この例の Ceph File System ID は **cephfs@11** です。

2. ミラーステータスを表示するには、以下を実行します。

   構文

   ```
   ceph --admin-daemon PATH_TO_THE_ASOK_FILE fs mirror status
   FILE_SYSTEM_NAME@_FILE_SYSTEM_ID
   ```

   例

   ```
   [root@mon ~]# ceph --admin-daemon /var/run/ceph/ceph-client.cephfs-
   mirror.node1.bndvox.asok fs mirror status cephfs@11
   {
     "rados_inst": "192.168.0.5:0/1476644347",
     "peers": {
       "1011435c-9e30-4db6-b720-5bf482006e0e": { (1)
         "remote": {
           "client_name": "client.mirror_remote",
           "cluster_name": "remote-site",
           "fs_name": "cephfs"
         }
       }
     },
     "snap_dirs": {
       "dir_count": 1
     }
   }
   ```

   (1) これは、固有のピア UUID です。

3. ピアステータスを表示するには、以下を実行します。

   構文

構文

```
ceph --admin-daemon PATH_TO_ADMIN_SOCKET fs mirror status
FILE_SYSTEM_NAME@FILE_SYSTEM_ID _PEER_UUID
```

例

```
[root@mon ~]# ceph --admin-daemon /var/run/ceph/cephfs-mirror.asok fs mirror peer status
cephfs@11 1011435c-9e30-4db6-b720-5bf482006e0e
{
  "/home/user1": {
      "state": "idle", 1
      "last_synced_snap": {
          "id": 120,
          "name": "snap1",
          "sync_duration": 0.079997898999999997,
          "sync_time_stamp": "274900.558797s"
      },
      "snaps_synced": 2, 2
      "snaps_deleted": 0, 3
      "snaps_renamed": 0 4
  }
}
```

1 **state** は、以下の 3 つの値のいずれかになります。

- **idle** は、ディレクトリーが現在同期していないことを意味します。
- **syncing** とは、ディレクトリーが現在同期中であることを意味します。
- **failed** の原因は、ディレクトリーが連続した失敗の上限に達したことを意味します。連続する失敗のデフォルト数は 10 で、デフォルトの再試行間隔は 60 秒です。

2 3 4 同期統計: **snaps_synced**、**snaps_deleted**、および **snaps_renamed** は、**cephfs-mirror** デーモンの再起動時にリセットされます。

関連情報

- 詳細は、『**Red Hat Ceph Storage ファイルシステムガイド**』の「Ceph ファイルシステムのミラー」セクションを参照してください。

5.22. 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 ファイルシステムガイド』の「**Ceph File System Metadata Server**」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、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、「メタデータサーバーキャッシュサイズの制限」のセクションを参照してください。

"Client name failing to advance its oldest client/flush tid"

コード: MDS_HEALTH_CLIENT_OLDEST_TID, MDS_HEALTH_CLIENT_OLDEST_TID_MANY
クライアントと MDS サーバー間で通信するための CephFS プロトコルは、**oldest tid** というフィールドを使用して、MDS が対応するためにクライアント要求が完全に完了している MDS に通知するものです。反応しないクライアントがこのフィールドを進めない場合、MDS はクライアント要求によって使用されるリソースを適切にクリーンアップできなくなる可能性があります。このメッセージは、クライアントが **max_completed_requests** オプション (デフォルトは 100000) で指定された数値よりも多くのリクエストがある場合に表示されます。これは、MDS 側では完全でも、クライアントの **最も古い tid** 値について考慮されていないことを示しています。

"Metadata damage detected"

コード: MDS_HEALTH_DAMAGE
メタデータプールから読み取り時に、破損したメタデータまたは欠落しているメタデータが見つかりました。このメッセージは、MDS が動作を継続するために十分な破損した分離されたことを示しています。ただし、クライアントが破損したサブツリーへのアクセスにより I/O エラーが返されることを示します。**damage ls** administration socket コマンドを使用して、破損の詳細を表示します。このメッセージは、破損が発生するとすぐに表示されます。

"MDS in read-only mode"

Code: MDS_HEALTH_READ_ONLY
MDS は読み取り専用モードに入力されており、メタデータの変更を試みるクライアント操作に **EROFS** エラーコードを返します。MDS は読み取り専用モードに入ります。

- メタデータプールへの書き込み中に書き込みエラーが発生した場合
- **force_readonly** 管理ソケットコマンドを使用して、管理者が MDS を読み取り専用モードに強制するとき。

"<N> slow requests are blocked"

コード: MDS_HEALTH_SLOW_REQUEST
1 つ以上のクライアント要求が完了しておらず、MDS が非常に遅いか、バグが発生したことを示しています。**ops** 管理ソケットコマンドを使用して、未処理のメタデータ操作を一覧表示します。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。

関連情報

- 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ー制限。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、メモリー、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_halflife

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 が停止する前にバランサーを実行する反復数。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 が停止するまでのバランサーを実行する秒数。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、バランサーの反復回数。

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バランサー反復の最大数。

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、バリアント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アント。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

サブツリーのバリアント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

ジャーナルャ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

No

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

No

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

No

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

No

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

No

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カ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数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ー名を設定します。

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に設計されています。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)