



Red Hat Ceph Storage 3

Ceph ファイルシステムガイド

Ceph ファイルシステムの設定とマウント

Red Hat Ceph Storage 3 Ceph 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

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Ceph_File_System_Guide.ent file |.

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

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

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

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

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

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

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

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

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

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

All other trademarks are the property of their respective owners.

概要

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

目次

第1章 CEPH ファイルシステムの概要	4
1.1. CEPH ファイルシステムについて	4
Ceph ファイルシステムコンポーネント	4
1.2. CEPHFS の主な機能	5
1.3. CEPHFS の制限	6
1.4. CEPH ファイルシステムの POSIX コンプライアンスとの相違点	7
1.5. その他のリソース	7
第2章 メタデータサーバーデーモンの設定	8
2.1. 前提条件	8
2.2. その他のリソース	8
第3章 CEPH ファイルシステムのデプロイ	9
3.1. 前提条件	9
3.2. CEPH ファイルシステムの作成	9
前提条件	9
手順	10
その他のリソース	11
3.3. CEPH FILE SYSTEM クライアントユーザーの作成	11
手順	11
その他のリソース	13
3.4. CEPH FILE SYSTEM のカーネルクライアントとしてのマウント	13
3.4.1. 前提条件	13
3.4.2. Ceph ファイルシステムをカーネルクライアントとして手動でマウントする手順	14
その他のリソース	15
3.4.3. Ceph ファイルシステムをカーネルクライアントとして自動的にマウント	15
前提条件	15
手順	15
3.5. CEPH ファイルシステムを FUSE クライアントとしてマウントする	16
3.5.1. 前提条件	16
3.5.2. Ceph ファイルシステムを FUSE クライアントとして手動でマウントする手順	17
前提条件	17
手順	17
その他のリソース	18
3.5.3. Ceph ファイルシステムを FUSE クライアントとして自動的にマウント	18
前提条件	18
手順	18
3.6. イレイジャーコーディングを使用した CEPH ファイルシステムの作成	19
第4章 CEPH ファイルシステムの管理	23
4.1. 前提条件	23
4.2. ディレクトリーツリーの MDS ランクへのマッピング	23
前提条件	23
手順	23
その他のリソース	24
4.3. MDS RANKS からのディレクトリーツリーの関連付けを解除	24
前提条件	24
手順	24
その他のリソース	24
4.4. ファイルとディレクトリーのレイアウトでの作業	24
4.4.1. 前提条件	24
4.5. データプールの追加	25

手順	25
4.6. CEPH FILE SYSTEM クォータの使用	26
4.6.1. 前提条件	26
4.6.2. Ceph File システムのクォータ	26
4.6.3. クォータの表示	26
4.6.4. クォータの設定	27
4.6.5. クォータの削除	28
4.6.6. その他のリソース	29
4.7. CEPH ファイルシステムの削除	29
第5章 CEPH ファイルシステムのマウント解除	31
5.1. カーネルクライアントとしてマウントされた CEPH ファイルシステムのマウント解除	31
手順	31
その他のリソース	31
5.2. FUSE クライアントとしてマウントされた CEPH ファイルシステムのマウント解除	31
手順	31
その他のリソース	31
付録A トラブルシューティング	32
A.1. CEPHFS のヘルスメッセージ	32
付録B 設定リファレンス	34
B.1. MDS 設定リファレンス	34
B.2. ジャーナル設定リファレンス	48
B.3. FUSE クライアント設定のリファレンス	50
開発者オプション	55

第1章 CEPH ファイルシステムの概要

本章では、Ceph File System(CephFS)の概要と、その仕組みについて説明します。

1.1. CEPH ファイルシステムについて

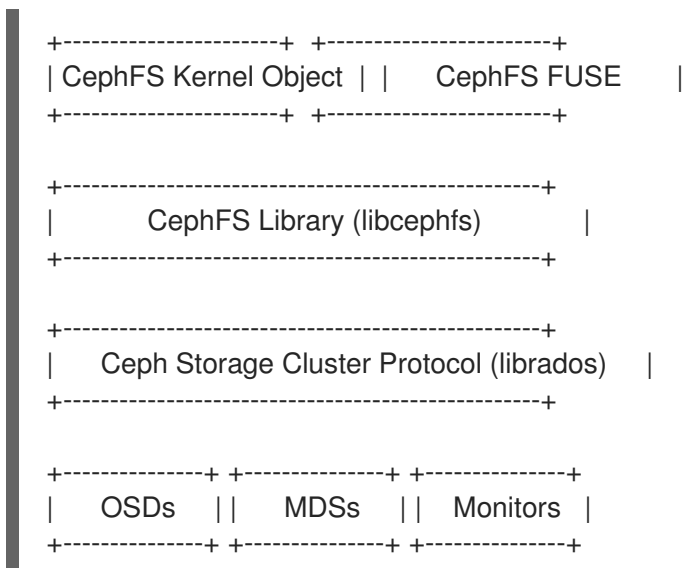
Ceph File System(CephFS)は、POSIX 標準と互換性のあるファイルシステムで、Ceph Storage クラスターへのファイルアクセスを提供します。

CephFS を実行するには、少なくとも1つのメタデータサーバー(MDS)デーモン(**ceph-mds**)が必要です。MDS デーモンは、Ceph ファイルシステムに保管されているファイルに関するメタデータを管理し、共有 Ceph Storage クラスターへのアクセスも調整します。

CephFS は、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。詳細は、「[Ceph ファイルシステムの POSIX コンプライアンスとの相違点](#)」を参照してください。

Ceph ファイルシステムコンポーネント

この図は、Ceph File System のさまざまなレイヤーを示しています。



下層は、以下を含む基礎となるコアクラスターを表します。

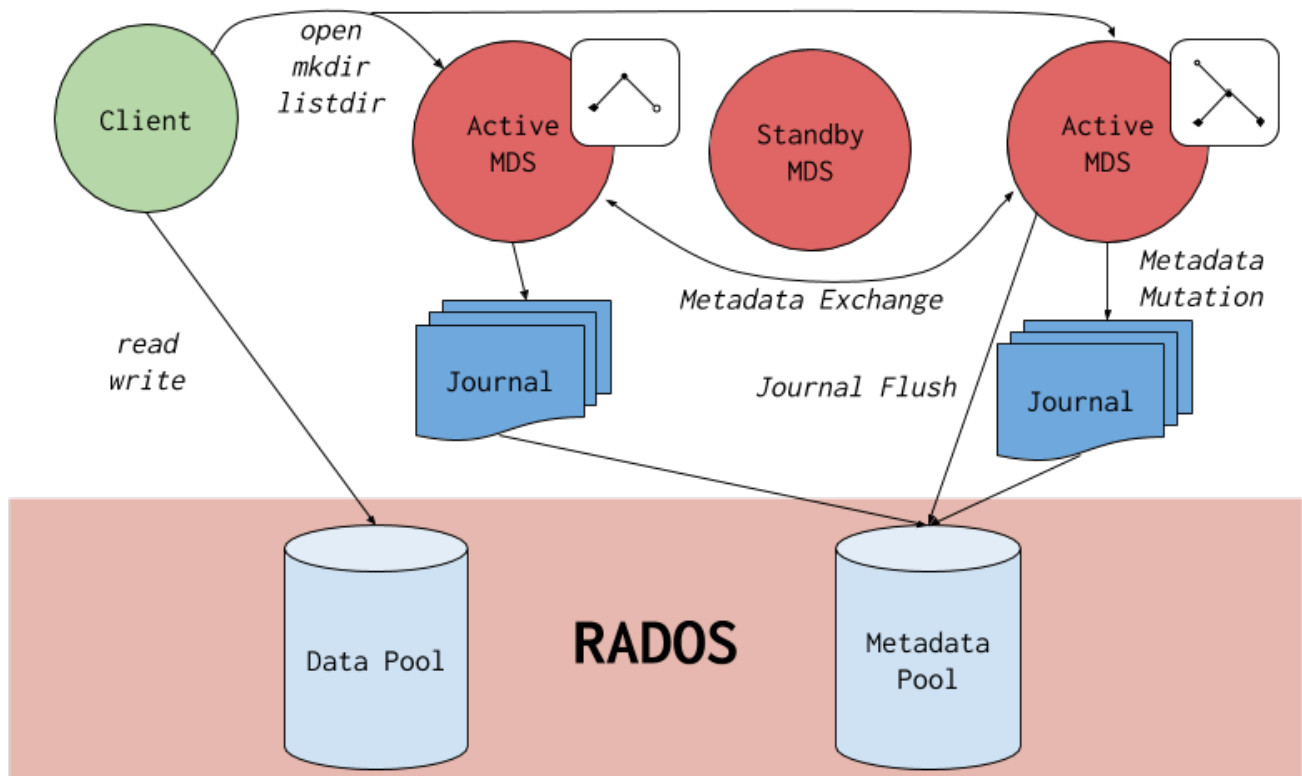
- Ceph File System データとメタデータが保存される OSD(**ceph-osd**)
- Ceph ファイルシステムのメタデータを管理するメタデータサーバー(**ceph-mds**)
- クラスタマップのマスターコピーを管理するモニター(**ceph-mon**)

Ceph Storage Cluster Protocol レイヤーは、コア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。

上層は、Ceph ファイルシステムにアクセスできる2種類のクライアントを表します。

この図では、Ceph File System コンポーネントが相互に対話する方法を説明します。



Ceph File System には、以下の主要コンポーネントがあります。

- **クライアント** は CephFS を使用したアプリケーションの代わりに I/O 操作を実行するエンティティを表します (FUSE クライアントの場合は **ceph-fuse**、カーネルクライアントの場合は **kcephfs**)。クライアントはメタデータ要求をアクティブな MDS に送信します。戻りに、クライアントはファイルメタデータを学習し、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** はクライアントにメタデータを提供し、バックアップメタデータプールストアへの要求を減らし、キャッシュの一貫性を維持するためにクライアントキャッシュを管理し、アクティブな MDS 間でホットメタデータを複製し、メタデータの変更をバックアップメタデータプールに通常のフラッシュでコンパクトなジャーナルに結合します。

1.2. CEPHFS の主な機能

Ceph File System には、以下の機能および機能拡張が追加されました。

スケーラビリティ

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を実現しています。

共有ファイルシステム

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

高可用性

Ceph File System には、Ceph Metadata Server (MDS) のクラスターが用意されています。1つはアクティブで、他はスタンバイモードです。アクティブなデータシートが不意に終了した場合、スタンバイデータシートの1つがアクティブになります。その結果、サーバーが故障してもクライアント

のマウントは継続して動作します。この動作により、Ceph File System は可用性が高くなります。さらに、複数のアクティブなメタデータサーバーを設定することも可能です。詳しくは、[???](#) を参照してください。

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。詳しくは、「[ファイルとディレクトリーのレイアウトでの作業](#)」を参照してください。

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** でカーネルクライアントとしてマウントされている Ceph File Systems でデフォルトで有効になっています。

Ceph File Systems で FUSE クライアントとしてマウントされている ACL を使用するには、その ACL を有効にする必要があります。詳しくは、「[CephFS の制限](#)」を参照してください。

クライアントクォータ

Ceph File System FUSE クライアントは、システム内の任意のディレクトリーにクォータの設定をサポートします。クォータは、ディレクトリー階層のそのポイントの下に保存されているバイト数やファイル数を制限することができます。クライアントクォータはデフォルトで有効になっていません。

1.3. CEPHFS の制限

FUSE クライアントのアクセス制御リスト (ACL) のサポート

FUSE クライアントとしてマウントされた Ceph File System で ACL 機能を使用するには、これを有効にする必要があります。そのためには、以下のオプションを Ceph 設定ファイルに追加します。

```
[client]
client_acl_type=posix_acl
```

次に、Ceph クライアントを再起動します。

スナップショット

この機能は実験的な機能で、MDS またはクライアントノードが予期せず終了するため、スナップショットの作成はデフォルトでは有効になっていません。

リスクを理解し、スナップショットを有効にする必要がある場合は、以下を使用します。

```
ceph mds set allow_new_snaps true --yes-i-really-mean-it
```

複数の Ceph ファイルシステム

デフォルトでは、1つのクラスターで複数の Ceph ファイルシステムの作成は無効になっています。追加の Ceph File System の作成を試みると、以下のエラーで失敗します。

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

1つのクラスターで複数の Ceph ファイルシステムを作成することは完全にサポートされていないため、MDS またはクライアントノードが予期せず終了する可能性があります。

リスクを理解し、複数の Ceph ファイルシステムを有効にする必要がある場合は、以下を使用します。

■

```
ceph fs flag set enable_multiple true --yes-i-really-mean-it
```

1.4. CEPH ファイルシステムの POSIX コンプライアンスとの相違点

本セクションでは、Ceph File System(CephFS)が厳格な POSIX セマンティクスと異なる状況を一覧表示します。

- 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。たとえば、ライター A は "aa|aa" と writer B を同時に書き込みます ("|" はオブジェクト境界)。また、「aa|bb」は適切な「aa|aa」または「bb|bb」ではなく、書き込まれます。
- 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。CephFS はファイルのどの部分が割り当てまたは書き込まれるのかを明示的に追跡しないため、**st_blocks** フィールドには、ブロックサイズで分けられたファイルサイズが常に設定されます。この動作により、**du** などのユーティリティーが消費した領域を過剰に予測します。
- **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。マウント時のこの非表示ディレクトリーの名前は、**-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。

1.5. その他のリソース

- 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 メタデータサーバーデーモンの設定

本章では、Ceph Metadata Server(MDS)デーモンを設定する方法について説明します。

- MDS デーモンのさまざまな状態を把握するには、[???](#)を参照してください。
- MDS 設定の「ランク」の意味を把握するには、[???](#)を参照してください。
- さまざまな設定タイプのスタンバイ MDS デーモンの情報は、[???](#)を参照してください。
- スタンバイ MDS デーモンを設定するには、[???](#)を参照してください。
- 複数のアクティブな MDS デーモンを設定するには、[???](#)を参照してください。
- アクティブな MDS デーモンの数を減らすには、[???](#)を参照してください。
- MDS キャッシュサイズの制限については、[???](#)を参照してください。



注記

Red Hat Ceph Storage 3.2 以降では、`ceph-mds` デーモンおよび `ceph-fuse` デーモンを Enforcing モードで SELinux で実行できます。

2.1. 前提条件

- Ceph Storage クラスターがない場合はデプロイします。詳細は、『[Installation Guide for Red Hat Enterprise Linux or Ubuntu](#)』を参照してください。
- Ceph Metadata Server デーモン(`ceph-mds`)をインストールします。詳細は、『[Installation Guide for Red Hat Enterprise Linux or Ubuntu](#)』を参照してください。

2.2. その他のリソース

- [Red Hat Enterprise Linux のインストールガイド](#)
- [Ubuntu のインストールガイド](#)

第3章 CEPH ファイルシステムのデプロイ

本章では、Ceph ファイルシステムを作成およびマウントする方法を説明します。

Ceph File System をデプロイするには、以下を実行します。

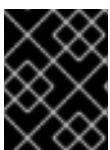
1. Monitor ノードに Ceph ファイルシステムを作成します。詳しくは、[「Ceph ファイルシステムの作成」](#) を参照してください。
2. 適切なアクセス権限とパーミッションでクライアントユーザーを作成し、その鍵を Ceph File System がマウントされるノードで利用できるようにします。詳しくは、[「Ceph File System クライアントユーザーの作成」](#) を参照してください。
3. 専用ノードに CephFS をマウントします。以下の方法のいずれかを選択します。
 - a. CephFS をカーネルクライアントとしてマウントする。[「Ceph File System のカーネルクライアントとしてのマウント」](#) を参照
 - b. CephFS を FUSE クライアントとしてマウントする。[「Ceph ファイルシステムを FUSE クライアントとしてマウントする」](#) を参照

3.1. 前提条件

- Ceph Storage クラスタがない場合はデプロイします。詳細は、[『Installation Guide for Red Hat Enterprise Linux or Ubuntu』](#) を参照してください。
- Ceph Metadata Server デーモン(ceph-mds)をインストールして設定します。詳細は、[『Installation Guide for Red Hat Enterprise Linux or Ubuntu and 2章メタデータサーバーデーモンの設定』](#) を参照してください。

3.2. CEPH ファイルシステムの作成

本セクションでは、モニターノードで Ceph File System を作成する方法を説明します。



重要

デフォルトでは、Ceph Storage クラスタに Ceph File System を1つだけ作成できません。詳しくは、[「CephFS の制限」](#) を参照してください。

前提条件

- Ceph Storage クラスタがない場合はデプロイします。詳細は、[『Installation Guide for Red Hat Enterprise Linux』](#) または [『Installation Guide for Ubuntu』](#) を参照してください。
- Ceph Metadata Server デーモン(ceph-mds)をインストールして設定します。詳細は、[『Installation Guide for Red Hat Enterprise Linux』](#) の「Installing Metadata Servers」または [「Installation Guide for Ubuntu」](#) を参照してください。
- ceph-common パッケージをインストールします。

- On Red Hat Enterprise Linux:

```
# yum install ceph-common
```

- Ubuntu の場合 :

```
$ sudo apt-get install ceph-common
```



注記

リポジトリを有効にし、定義したクライアントノードに **ceph-common** パッケージをインストールするには、『[Installation Guide for Red Hat Enterprise Linux](#)』の「[Installing the Ceph Client Role](#)」または「[Installation Guide for Ubuntu](#)」を参照してください。

手順

Monitor ホストと **root** ユーザーで、以下のコマンドを使用します。

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

```
ceph osd pool create <name> <pg_num>
```

以下のように、プール名と配置グループ(PG)の数を指定します。

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

通常、メタデータプールはデータプールよりもはるかに少ない数の PG で始めることができます。必要に応じて PG の数を増やすことができます。64 PG から 512 PG に推奨されるメタデータプールサイズの範囲。データプールのサイズは、ファイルシステムで予想されるファイルの数およびサイズに比例します。



重要

メタデータプールについては、[このリンク](#)の使用を検討してください。

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

2. Ceph File System を作成します。

```
ceph fs new <name> <metadata-pool> <data-pool>
```

Ceph File System の名前、メタデータおよびデータプールの名前を指定します。以下に例を示します。

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

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

```
ceph fs status <name>
```

Ceph File System の名前を指定します。以下に例を示します。

```
[root@monitor ~]# 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 |
+-----+-----
```

その他のリソース

- Red Hat Enterprise Linux の『[Red Hat Ceph Storage 3 インストールガイド](#)』の「[Red Hat Ceph Storage リポジトリの有効化](#)」セクション
- Ubuntu の [Red Hat Ceph Storage リポジトリの有効化](#) Red Hat Ceph Storage 3 インストールガイド
- Red Hat Ceph Storage 3 の『[ストレージ戦略](#)』ガイドの「[プール](#)」の章

3.3. CEPH FILE SYSTEM クライアントユーザーの作成

Red Hat Ceph Storage 3 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、Monitor ノードで正しい承認機能を備えたユーザーを作成し、その鍵を 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。

カーネルクライアントで使用するために鍵を使用できるようにするには、そのキーでクライアントノードでシークレットファイルを作成します。ユーザー空間(FUSE)クライアントのファイルシステムで鍵を利用できるようにするには、キーリングをクライアントノードにコピーします。

手順

1. Monitor ホストで、クライアントユーザーを作成します。

```
ceph auth get-or-create client.<id> <capabilities>
```

クライアント ID と必要な機能を指定します。

- クライアントをクラスター内の特定のプールからのみ書き込み/読み取りに制限するには、以下を実行します。

```
ceph auth get-or-create client.1 mon 'allow r' mds 'allow rw' osd 'allow rw pool=<pool>'
```

たとえば、クライアントがデータ プールからのみ書き込み/読み取りするよう制限するには、次のコマンドを実行します。

```
[root@monitor ~]# ceph auth get-or-create client.1 mon 'allow r' mds 'allow rw' osd 'allow rw pool=data'
```

- クライアントがファイルおよびディレクトリーに使用されるプールを変更しないようにするには、以下を実行します。

```
[root@monitor ~]# ceph auth get-or-create client.1 mon 'allow r' mds 'allow r' osd 'allow r pool=<pool>'
```

たとえば、クライアントがデータ プールを変更しないようにするには、次のコマンドを実行します。

```
[root@monitor ~]# ceph auth get-or-create client.1 mon 'allow r' mds 'allow r' osd 'allow r pool=data'
```



注記

Ceph File System クライアントはアクセスできないので、メタデータ プールの機能は作成しないでください。

2. 作成したキーを確認します。

```
ceph auth get client.<id>
```

以下に例を示します。

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

3. カーネルクライアントを使用する予定がある場合は、前の手順で取得したキーを使用してシークレットファイルを作成します。
クライアントノードで、`key =` の後に文字列を `/etc/ceph/ceph.client.<id>.secret` にコピーします。

たとえば、クライアント ID が 1 の場合は、キーを使用して `/etc/ceph/ceph.client.1.secret` に 1 つの行を追加します。

```
[root@client ~]# cat /etc/ceph/ceph.client.1.secret
AQBSdFhcGZFUDRAAcKhG9CI2HPiDMMRv4DC43A==
```



重要

`key =` と文字列の間にスペースを含めないでください。そうでないとマウントが動作しません。

4. User Space(FUSE)クライアントで File System を使用する予定がある場合は、キーリングをクライアントにコピーします。
 - a. 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。


```
# ceph auth get client.<id> -o ceph.client.<id>.keyring
```

たとえば、クライアント ID が 1 の場合は、以下のようになります。

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

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

```
scp root@<monitor>:/root/ceph.client.1.keyring /etc/ceph/ceph.client.1.keyring
```

<monitor> を Monitor ホスト名または IP に置き換えます。以下に例を示します。

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

5. 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。

```
chmod 644 <keyring>
```

キーリングへのパスを指定します。以下に例を示します。

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

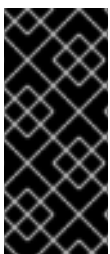
その他のリソース

- Red Hat Ceph Storage 3 の『管理ガイド』の「[ユーザー管理](#)」の章

3.4. 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 をカーネルクライアントとしてマウントすることができます。

- [mount コマンドラインユーティリティーを使用した手動](#)
- [/etc/fstab ファイルにエントリーを追加して自動的に設定する](#)



重要

Red Hat Enterprise Linux 以外の Linux ディストリビューション上の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を使用する際に MDS またはその他のクラスターの他の部分で問題が検出される場合は、Red Hat はこれらに対応しますが、原因がクライアント側にある場合は、カーネルのベンダーが問題に対処する必要があります。

3.4.1. 前提条件

- クライアントノードで、Red Hat Ceph Storage 3 Tools リポジトリーを有効にします。
 - Red Hat Enterprise Linux の場合は、以下を使用します。

```
[root@client ~]# subscription-manager repos --enable=rhel-7-server-rhceph-3-tools-rpms
```

- Ubuntu の場合は、以下を使用します。

```
[user@client ~]$ sudo bash -c 'umask 0077; echo deb
https://customername:customerpasswd@rhcs.download.redhat.com/3-updates/Tools
$(lsb_release -sc) main | tee /etc/apt/sources.list.d/Tools.list'
[user@client ~]$ sudo bash -c 'wget -O - https://www.redhat.com/security/fd431d51.txt |
apt-key add -'
[user@client ~]$ sudo apt-get update
```

- 宛先クライアントノードで、新しい `etc/ceph` ディレクトリーを作成します。

```
[root@client ~]# mkdir /etc/ceph
```

- Ceph 設定ファイルを Monitor ノードから移行先クライアントノードにコピーします。

```
scp root@<monitor>:/etc/ceph/ceph.conf /etc/ceph/ceph.conf
```

< ;monitor> を Monitor ホスト名または IP アドレスに置き換えます。以下に例を示します。

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

- `ceph.conf` ファイルに正しい所有者およびグループを設定します。

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

- 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
```

3.4.2. Ceph ファイルシステムをカーネルクライアントとして手動でマウントする手順

Ceph File System をカーネルクライアントとして手動でマウントするには、`mount` ユーティリティーを使用します。

前提条件

- Ceph File System が作成されている。
- `ceph-common` パッケージがインストールされている。

手順

1. マウントディレクトリーを作成します。

```
mkdir -p <mount-point>
```

以下に例を示します。

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

2. Ceph ファイルシステムをマウントします。複数の Monitor アドレスを指定するには、`mount`

コマンドにコンマを使用して区切るか、または1つのホスト名が複数のIPアドレスに解決されるようにDNSサーバーを設定し、そのホスト名を `mount` コマンドに渡します。ユーザー名およびシークレットファイルへのパスを設定します。

```
mount -t ceph <monitor1-host-name>:6789,<monitor2-host-name>:6789,<monitor3-host-name>:6789:/ <mount-point> -o name=<user-name>,secretfile=<path>
```

以下に例を示します。

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

3. 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。

```
stat -f <mount-point>
```

以下に例を示します。

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

その他のリソース

- man ページの `mount(8)`
- Red Hat Enterprise Linux 7 の『ネットワークガイド』の「[DNS サーバー](#)」の章
- Red Hat Ceph Storage 3 の『管理ガイド』の「[ユーザー管理](#)」の章

3.4.3. Ceph ファイルシステムをカーネルクライアントとして自動的にマウント

起動時に Ceph File System を自動的にマウントするには、`/etc/fstab` ファイルを編集します。

前提条件

- 最初にファイルシステムを手動でマウントすることを検討してください。詳しくは、「[Ceph ファイルシステムをカーネルクライアントとして手動でマウントする手順](#)」を参照してください。
- `secretfile=` マウントオプションを使用する場合は、`ceph-common` パッケージをインストールします。

手順

1. 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。

```
mkdir -p <mount-point>
```

以下に例を示します。

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

2. 以下のように `/etc/fstab` ファイルを編集します。

```
#DEVICE          PATH          TYPE  OPTIONS
<host-name>:<port>:/, <mount-point> ceph  _netdev,
<host-name>:<port>:/,          [name=<user-name>,
<host-name>:<port>:/,          secret=<key>|
                                secretfile=<file>,
                                [<mount-options>]
```

最初の列で、Monitor ホスト名とそのポートを設定します。複数のモニターアドレスを指定する別の方法として、1つのホスト名が複数の IP アドレスに解決されるように DNS サーバーを設定することができます。

2 番目のコラムにマウントポイントを設定し、3 番目のコラムで種別を `ceph` に設定します。

`name` と `secretfile` オプションを使用して、4 列にユーザー名とシークレットファイルをそれぞれ設定します。

ネットワークの問題を防ぐために、ネットワークサブシステムの後にはファイルシステムがマウントされるように `_netdev` オプションを設定します。時間情報にアクセスする必要がない場合は、パフォーマンスを向上させるために `noatime` を設定します。

以下に例を示します。

```
#DEVICE          PATH          TYPE  OPTIONS
mon1:6789:/,     /mnt/cephfs   ceph  _netdev, name=admin,
mon2:6789:/,     secretfile=
mon3:6789:/     /home/secret.key,
                                noatime 00
```

ファイルシステムが次の起動時にマウントされます。

3.5. CEPH ファイルシステムを FUSE クライアントとしてマウントする

Ceph File System は、FUSE(User Space)クライアントでファイルシステムとしてマウントできます。

- [ceph-fuse コマンドラインユーティリティーを使用した手動](#)
- [/etc/fstab ファイルにエントリーを追加して自動的に設定する](#)

3.5.1. 前提条件

- クライアントノードで、Red Hat Ceph Storage 3 Tools リポジトリを有効にします。
 - Red Hat Enterprise Linux の場合は、以下を使用します。

```
[root@client ~]# subscription-manager repos --enable=rhel-7-server-rhceph-3-tools-rpms
```

- Ubuntu の場合は、以下を使用します。

```
[user@client ~]$ sudo bash -c 'umask 0077; echo deb
https://customername:customerpasswd@rhcs.download.redhat.com/3-updates/Tools
$(lsb_release -sc) main | tee /etc/apt/sources.list.d/Tools.list'
[user@client ~]$ sudo bash -c 'wget -O - https://www.redhat.com/security/fd431d51.txt |
apt-key add -'
[user@client ~]$ sudo apt-get update
```

- クライアントキーリングをクライアントノードにコピーします。詳しくは、「[Ceph File System クライアントユーザーの作成](#)」を参照してください。
- 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。

```
scp root@<monitor>:/etc/ceph/ceph.conf /etc/ceph/ceph.conf
```

<monitor> を Monitor ホスト名または IP に置き換えます。以下に例を示します。

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

- 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
```

3.5.2. Ceph ファイルシステムを FUSE クライアントとして手動でマウントする手順

Ceph File System をユーザー空間(FUSE)クライアントでファイルシステムとしてマウントするには、**ceph-fuse** ユーティリティを使用します。

前提条件

- Ceph File System がマウントされるノードに、**ceph-fuse** パッケージをインストールします。
 - Red Hat Enterprise Linux の場合は、以下を使用します。

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

- Ubuntu の場合は、以下を使用します。

```
[user@client ~]$ sudo apt-get install ceph-fuse
```

手順

1. マウントポイントとして使用するディレクトリーを作成します。MDS 機能と共に **path オプション**を使用した場合、マウントポイントは **パス** で指定されたものである必要があります。

```
mkdir <mount-point>
```

以下に例を示します。

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

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

```
ceph-fuse -n client.<client-name> <mount-point>
```

以下に例を示します。

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

- 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(`/etc/ceph/ceph.client.<client-name/id>.keyring`), 以下のように `--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。

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

- クライアントを特定のディレクトリー内のマウントのみに制限し、作業する場合は、`-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クライアントに指示します。

```
ceph-fuse -n client.<client-name/id> <mount-point> -r <path>
```

たとえば、`/home/cephfs/` ディレクトリーを `root` として処理するようにクライアントに対して ID 1 に指示するには、次のコマンドを実行します。

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

3. 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。

```
stat -f <mount-point>
```

以下に例を示します。

```
[user@client ~]$ stat -f /mnt/cephfs
```

その他のリソース

- `ceph-fuse(8)` man ページ *
- Red Hat Ceph Storage 3 の『管理ガイド』の「[ユーザー管理](#)」の章

3.5.3. Ceph ファイルシステムを FUSE クライアントとして自動的にマウント

起動時に Ceph File System を自動的にマウントするには、`/etc/fstab` ファイルを編集します。

前提条件

- 最初にファイルシステムを手動でマウントすることを検討してください。詳しくは、「[Ceph ファイルシステムをカーネルクライアントとして手動でマウントする手順](#)」を参照してください。

手順

1. 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。

```
mkdir -p <mount-point>
```

以下に例を示します。

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

2. 以下のように `etc/fstab` ファイルを編集します。

```
#DEVICE    PATH          TYPE    OPTIONS
```

```
none <mount-point> fuse.ceph _netdev
      ceph.id=<user-id>
      [,ceph.conf=<path>],
      defaults 0 0
```

client-admin ではなく、admin やマウントポイントなどの使用 ID を指定します。Ceph 設定ファイルをデフォルトの場所以外に保存する場合は、conf オプションを使用します。さらに、必要なマウントオプションを指定します。ネットワークの問題を防ぐために、ネットワークサブシステムの後にはファイルシステムがマウントされるように _netdev オプションを使用することを検討してください。以下に例を示します。

```
#DEVICE PATH TYPE OPTIONS
none /mnt/ceph fuse.ceph _netdev
      ceph.id=admin,
      ceph.conf=/etc/ceph/cluster.conf,
      defaults 0 0
```

ファイルシステムが次の起動時にマウントされます。

3.6. イレイジャーコーディングを使用した CEPH ファイルシステムの作成

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



重要

イレイジャーコーディングされたプールの Ceph ファイルシステムはテクノロジープレビュー機能です。詳細は、「[Erasure Coding with Overwrites \(テクノロジープレビュー\)](#)」を参照してください。



重要

イレイジャーコーディングされたプールの Ceph ファイルシステムには、BlueStore オブジェクトストアを使用するプールが必要です。詳細は、「[Erasure Coding with Overwrites \(テクノロジープレビュー\)](#)」を参照してください。



重要

Red Hat は、レプリケートされたプールをデフォルトのデータプールとして使用することを推奨します。

前提条件

- 稼働中の Red Hat Ceph Storage クラスタ
- BlueStore OSD を使用するプール。

手順

1. Ceph File System 用のイレイジャーコーディングされたデータプールを作成します。

```
ceph osd pool create $DATA_POOL $PG_NUM erasure
```

たとえば、64 個の配置グループを持つ `cephfs-data-ec` という名前のイレイジャーコーディングされたプールを作成するには、以下を実行します。

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

2. Ceph File System 用に複製されたメタデータプールを作成します。

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

たとえば、64 個の配置グループを持つ `cephfs-metadata` という名前のプールを作成するには、次のコマンドを実行します。

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

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。

```
ceph osd pool set $DATA_POOL allow_ec_overwrites true
```

たとえば、`cephfs-data-ec` という名前のイレイジャーコーディングされたプールで上書きを有効にするには、以下を実行します。

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

4. Ceph File System を作成します。

```
ceph fs new $FS_EC $METADATA_POOL $DATA_POOL
```



注記

デフォルトのデータプールにイレイジャーコーディングされたプールを使用することは推奨されませんが、`--force` を使用してこのデフォルトを上書きできます。Ceph File System の名前と、メタデータとデータプールの名前を指定します。以下に例を示します。

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

5. 設定に応じて、1つ以上の MDS が active 状態にあることを確認します。

```
ceph fs status $FS_EC
```

Ceph File System の名前を指定します。以下に例を示します。

```
[root@monitor ~]# ceph fs status cephfs-ec
cephfs-ec - 0 clients
=====
+-----+-----+-----+-----+-----+-----+
| Rank | State | MDS | Activity | dns | inos |
+-----+-----+-----+-----+-----+-----+
```



```

| 0 | active | node1 | Reqs: 0/s | 10 | 12 |
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
| Pool | type | used | avail |
+-----+-----+-----+-----+-----+
| cephfs-metadata | metadata | 4638 | 26.7G |
| cephfs-data-ec | data | 0 | 26.7G |
+-----+-----+-----+-----+-----+

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

```

6. イレイジャーコーディングされたプールをデータプールとして既存のファイルシステムに追加する場合は、以下を行います。

- a. Ceph File System 用のイレイジャーコーディングされたデータプールを作成します。

```
ceph osd pool create $DATA_POOL $PG_NUM erasure
```

たとえば、64 個の配置グループを持つ **cephfs-data-ec1** という名前のイレイジャーコーディングされたプールを作成するには、以下を実行します。

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

- b. 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。

```
ceph osd pool set $DATA_POOL allow_ec_overwrites true
```

たとえば、**cephfs-data-ec1** という名前のイレイジャーコーディングされたプールで上書きを有効にするには、以下を実行します。

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

- c. 新規作成されたプールを既存の Ceph File System に追加します。

```
ceph fs add_data_pool $FS_EC $DATA_POOL
```

たとえば、**cephfs-data-ec1** という名前のイレイジャーコーディングされたプールを追加するには、以下を実行します。

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

- d. 設定に応じて、1つ以上の MDS が active 状態にあることを確認します。

```
ceph fs status $FS_EC
```

Ceph File System の名前を指定します。以下に例を示します。

```
[root@monitor ~]# ceph fs status cephfs-ec
```

```

cephfs-ec - 0 clients
=====
+-----+-----+-----+-----+-----+
| Rank | State | MDS | Activity | dns | inos |
+-----+-----+-----+-----+-----+
| 0 | active | node1 | Reqs: 0/s | 10 | 12 |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| Pool | type | used | avail |
+-----+-----+-----+-----+
| cephfs-metadata | metadata | 4638 | 26.7G |
| cephfs-data-ec | data | 0 | 26.7G |
| cephfs-data-ec1 | data | 0 | 26.7G |
+-----+-----+-----+-----+

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

```

その他のリソース

- 詳細は、『Red Hat Ceph Storage ストレージ戦略ガイド』の「[イレイジャーコードプール](#)」セクションを参照してください。
- 詳細は、『Red Hat Ceph Storage ストレージ戦略ガイド』の「[上書きによるイレイジャーコード](#)」セクションを参照してください。

第4章 CEPH ファイルシステムの管理

本章では、一般的な Ceph File System の管理タスクについて説明します。

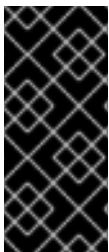
- ディレクトリーを特定の MDS ランクにマッピングするには、「[ディレクトリーツリーの MDS ランクへのマッピング](#)」を参照してください。
- MDS ランクからディレクトリーの関連付けを解除するには、「[MDS Ranks からのディレクトリーツリーの関連付けを解除](#)」を参照してください。
- ファイルとディレクトリーのレイアウトを使用する際には、「[ファイルとディレクトリーのレイアウトでの作業](#)」を参照してください。
- 新しいデータプールを追加するには、「[データプールの追加](#)」を参照してください。
- クォータを使用するには、「[Ceph File System クォータの使用](#)」を参照してください。
- Ceph File System を削除するには、「[Ceph ファイルシステムの削除](#)」を参照してください。

4.1. 前提条件

- Ceph Storage クラスタがない場合はデプロイします。詳細は、『[Installation Guide for Red Hat Enterprise Linux or Ubuntu](#)』を参照してください。
- Ceph Metadata Server デーモン(ceph-mds)をインストールして設定します。詳細は、『[Installation Guide for Red Hat Enterprise Linux or Ubuntu 2章メタデータサーバーデーモンの設定](#)』を参照してください。
- Ceph File System を作成し、マウントします。詳細は、[3章Ceph ファイルシステムのデプロイ](#)を参照してください。

4.2. ディレクトリーツリーの MDS ランクへのマッピング

本セクションでは、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メタデータサーバー(MDS)のランクにマッピングして、そのメタデータがそのランクを保持する MDS デーモンによってのみ管理されるようにする方法を説明します。この方法では、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の影響をクラスタ全体に制限したりすることができます。



重要

内部 balancer はすでにアプリケーションの負荷を分散することに注意してください。したがって、ディレクトリーツリーを、慎重に選択した特定のアプリケーションに対してのみランク付けします。さらに、ディレクトリーがランクにマップされると、balancer 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

- 複数のアクティブな MDS デーモンを設定します。詳しくは、[???](#)を参照してください。
- attr パッケージが、マウントされた Ceph File System のクライアントノードにインストールされていることを確認します。

手順

- [ディレクトリーを特定の MDS ランクにマッピングする](#)

- [ディレクトリーに `ceph.dir.pin` 拡張属性を設定します。](#)

```
setfattr -n ceph.dir.pin -v <rank> <directory>
```

たとえば、`/home/ceph-user/` ディレクトリーをすべてランク 2 に割り当てるには、そのサブディレクトリーをすべてランク 2 に割り当てます。

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

その他のリソース

- [「MDS Ranks からのディレクトリーツリーの関連付けを解除」](#)

4.3. MDS RANKS からのディレクトリーツリーの関連付けを解除

本セクションでは、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する方法を説明します。

前提条件

- `attr` パッケージが、マウントされた Ceph File System のクライアントノードにインストールされていることを確認します。

手順

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

```
setfattr -n ceph.dir.pin -v -1 <directory>
```

たとえば、MDS ランクから `/home/ceph-user/` ディレクトリーの関連付けを解除するには、次のコマンドを実行します。

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

`/home/ceph-user/` の個別マップされたサブディレクトリーには影響はありません。

その他のリソース

- [「ディレクトリーツリーの MDS ランクへのマッピング」](#)

4.4. ファイルとディレクトリーのレイアウトでの作業

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](#)
- [レイアウトの設定](#)
- [レイアウトの表示](#)
- [ディレクトリーレイアウト削除](#)

4.4.1. 前提条件

- `attr` パッケージがインストールされていることを確認します。

4.5. 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、ファイルとディレクトリーレイアウトを使用して、そのプール（およびオプションでそのプールの名前空間内）にファイルデータを保存するように、ファイルシステム階層の一部も設定する必要があります。詳しくは、「[ファイルとディレクトリーのレイアウトでの作業](#)」を参照してください。

手順

Monitor ホストと root ユーザーで、以下のコマンドを使用します。

1. 新しいデータプールを作成します。

```
ceph osd pool create <name> <pg_num>
```

以下を置き換えます。

- `<name>` は、プールの名前に置き換えます。
- `<pg_num>` は、配置グループ(PG)の数に置き換えます。

以下に例を示します。

```
[root@monitor]# ceph osd pool create cephfs_data_ssd 64
pool 'cephfs_data_ssd' created
```

2. メタデータサーバーの制御下に新たに作成されたプールを追加します。

```
ceph mds add_data_pool <name>
```

以下を置き換えます。

- `<name>` は、プールの名前に置き換えます。

以下に例を示します。

```
[root@monitor]# ceph mds add_data_pool 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。

```
[root@monitor]# 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。詳しくは、「[Ceph File System クライアントユーザーの作成](#)」を参照してください。

4.6. CEPH FILE SYSTEM クォータの使用

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

4.6.1. 前提条件

- `attr` パッケージがインストールされていることを確認します。

4.6.2. Ceph File システムのクォータ

本項では、CephFS におけるクォータのプロパティとその制限について説明します。

クォータの制限について

- 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。
- Linux カーネルクライアントバージョン 4.17 以降では、ユーザー空間クライアント `libcephfs` および `ceph-fuse` を使用して CephFS クォータをサポートします。ただし、これらのカーネルクライアントは `mimic+` クラスターのクォータのみをサポートします。最近のバージョンでも、カーネルクライアントはクォータの拡張属性を設定できる場合でも、古いストレージクラスターでクォータを管理できません。
- 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。<http://tracker.ceph.com/issues/24284> も参照してください。

4.6.3. クォータの表示

本項では、`getfaattr` 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する方法を説明します。



注記

属性が inode に表示されると、そのディレクトリーにクォータが設定されている必要があります。属性が inode に表示されない場合は、ディレクトリーにはクォータセットがありませんが、親ディレクトリーにはクォータが設定されている可能性があります。拡張属性の値が 0 の場合、クォータは設定されません。

前提条件

- `attr` パッケージがインストールされていることを確認します。

手順

1. CephFS クォータを表示するには、以下を実行します。

- a. バイト制限クォータの使用:

構文

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

例

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

- b. ファイル制限クォータの使用:

構文

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

例

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

その他のリソース

- 詳細は、`getfattr(1)` man ページを参照してください。

4.6.4. クォータの設定

本セクションでは、`setfattr` コマンドおよび `ceph.quota` 拡張属性を使用して、ディレクトリーのクォータを設定する方法を説明します。

前提条件

- `attr` パッケージがインストールされていることを確認します。

手順

1. CephFS クォータを設定します。

- a. バイト制限クォータの使用:

構文

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

例

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

この例では、100000000 バイトは 100 MB となります。

b. ファイル制限クォータの使用:

構文

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

例

```
[root@fs ~]# setfattr -n ceph.quota.max_files -v 10000 /cephfs/
```

この例では 10000 は 10,000 ファイルと等しくなります。

その他のリソース

- 詳細は、`setfattr(1)` man ページを参照してください。

4.6.5. クォータの削除

本セクションでは、`setfattr` コマンドおよび `ceph.quota` 拡張属性を使用して、ディレクトリーからクォータを削除する方法を説明します。

前提条件

- `attr` パッケージがインストールされていることを確認します。

手順

1. CephFS クォータを削除するには、以下のコマンドを実行します。

a. バイト制限クォータの使用:

構文

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

例

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

b. ファイル制限クォータの使用:

構文


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

例

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

関連情報

- 詳細は、`setfattr(1)` man ページを参照してください。

4.6.6. その他のリソース

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

4.7. CEPH ファイルシステムの削除

ストレージ管理者は、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 ノードに `root` ユーザーとしてログインします。

手順

1. クラスタに `down` のマークを付けます。

```
ceph fs set name cluster_down true
```

以下を置き換えます。

- `name` を、削除する Ceph File System の名前に置き換えます。

以下に例を示します。

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

2. Ceph File System のステータスを表示します。

```
ceph fs status
```

以下に例を示します。

```
[root@monitor]# ceph fs status
cephfs - 0 clients
=====
+-----+-----+-----+-----+-----+-----+
| Rank | State | MDS | Activity | dns | inos |
+-----+-----+-----+-----+-----+-----+
| 0 | active | ceph4 | Reqs: 0 /s | 10 | 12 |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
| Pool | type | used | avail |
+-----+-----+-----+-----+
| cephfs_metadata | metadata | 2246 | 975G |
| cephfs_data | data | 0 | 975G |
+-----+-----+-----+-----+
```

3. ステータスに表示されるすべての MDS ランクが失敗します。

```
# ceph mds fail rank
```

以下を置き換えます。

- MDS デーモンの ランク 付きが失敗する

以下に例を示します。

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

4. Ceph File System を削除します。

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

以下を置き換えます。

- name を、削除する Ceph File System の名前に置き換えます。

以下に例を示します。

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

5. 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@monitor]# ceph fs ls
```

6. オプション: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。『Red Hat Ceph Storage 3 ストレージストラテジーガイド』の「[プールの削除](#)」セクションを参照してください。

第5章 CEPH ファイルシステムのマウント解除

本章では、FUSE(User Space)クライアントでカーネルまたはファイルシステムとしてマウントされた Ceph ファイルシステムのマウントを解除する方法を説明します。

5.1. カーネルクライアントとしてマウントされた CEPH ファイルシステムのマウント解除

本項では、カーネルクライアントとしてマウントされている Ceph ファイルシステムのマウントを解除する方法を説明します。

手順

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

```
umount <mount-point>
```

ファイルシステムがマウントされるマウントポイントを指定します。

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

その他のリソース

- `umount(8)` man ページ

5.2. FUSE クライアントとしてマウントされた CEPH ファイルシステムのマウント解除

本セクションでは、ユーザー空間(FUSE)クライアントのファイルシステムとしてマウントされている Ceph ファイルシステムのマウントを解除する方法を説明します。

手順

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

```
fusermount -u <mount-point>
```

ファイルシステムがマウントされるマウントポイントを指定します。

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

その他のリソース

- `ceph-fuse(8)` man ページ

付録A トラブルシューティング

A.1. CEPHFS 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、MDS クラスターの特定の状態に応じて正常性メッセージを生成します。以下は、クラスターのヘルスメッセージとその説明の一覧です。

`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。クラスターは、適切な交換デーモンが起動するまで復元されません。

`MDS rank(s)<ranks> is damaged`

MDS ランク 1つまたは複数で、保存されたメタデータに重大な破損が生じ、メタデータが修復されるまで再度起動できません。

`MDS クラスターが動作が低下しています。`

現在、MDS のランク 1つ以上が稼働していないため、この状況が解決されるまで、クライアントはメタデータ I/O を一時停止する可能性があります。これには、失敗または破損のランクが含まれます。これには MDS で実行されているランクも含まれますが、再生状態のランクなど、アクティブな状態にないランクも含まれます。

`mds <names> are laggy`

MDS デーモンは、`mds_beacon_interval` オプションで指定された間隔で、モニターに beacon メッセージを送信します (デフォルトは 4 秒)。MDS デーモンが `mds_beacon_grace` オプションで指定した時間内にメッセージを送信できない場合 (デフォルトは 15 秒)、Ceph モニターは MDS デーモンを `laggy` とマークし、利用可能な場合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。この状態には人間が判読できるメッセージや、JSON 出力に表示される `MDS_HEALTH` を開始する固有のコードがあります。以下は、デーモンメッセージ、そのコード、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、すぐに実行できないか、またはまったく実行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くなると、MDS がキャッシュサイズ内に適切に維持されなくなり、最終的に MDS のメモリーが不足し、予期せず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。詳しくは、[???](#) 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 MDS 側で完了したが、クライアントの最も古い tid 値でまだ対応していない `max_completed_requests` オプション (デフォルトは 100000) よりも多くの要求がある場合に表示されます。

"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。クライアントリクエストに `mds_op_complaint_time` オプションで指定された値よりも長い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、予期せず終了する可能性があります。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 (デフォルトでは)。詳しくは、[???](#) を参照してください。

付録B 設定リファレンス

B.1. MDS 設定リファレンス

mon force standby active

説明

`true` に設定した場合は、スタンバイ再生モードの MDS を強制的にアクティブにします。Ceph 設定ファイルの `[mon]` または `[global]` セクションで設定します。

タイプ

ブール値

デフォルト

`true`

最大 mds

説明

クラスター作成時にアクティブな MDS デーモンの数。Ceph 設定ファイルの `[mon]` または `[global]` セクションで設定します。

タイプ

32 ビット整数

デフォルト

1

MDS の最大ファイルサイズ

説明

新規ファイルシステムの作成時に許可される最大ファイルサイズ。

タイプ

64 ビット整数未署名

デフォルト

`1ULL << 40`

MDS キャッシュのメモリー制限

説明

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 キャッシュの予約

説明

維持する MDS キャッシュのキャッシュ予約 (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 キャッシュサイズ**説明**

キャッシュする inode の数。値が 0 の場合は、無制限の数字を示します。Red Hat は、MDS キャッシュが使用するメモリー量を制限するために `mds_cache_memory_limit` を使用することを推奨します。

タイプ

32 ビット整数

デフォルト

0

mds cache mid**説明**

キャッシュ LRU（上から）の新しいアイテムの挿入ポイント。

タイプ

浮動小数点 (Float)

デフォルト

0.7

MDS のディレクトリーコミットの比率**説明**

ディレクトリーの一部には、（部分的な更新ではなく）Ceph が完全な更新を使用して Ceph がコミットする前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 ディレクトリーの最大コミットサイズ**説明**

Ceph がディレクトリーを小さなトランザクションに分割する前にディレクトリー更新の最大サイズ (MB 単位)。

タイプ

32 ビット整数

デフォルト

90

MDS decay halfife**説明**

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**説明**

モニターに送信される beacon 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**説明**

Ceph が MDS laggy を宣言する前に通知なしの間隔 (また置き換える可能性あり)

タイプ

浮動小数点 (Float)

デフォルト

15

MDS ブラックリストの間隔**説明**

OSD マップの失敗した MDS デーモンのブラックリスト期間。

タイプ

浮動小数点 (Float)

デフォルト

24.0*60.0

MDS セッションのタイムアウト**説明**

Ceph が機能およびリースを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 セッションの自動クローズ**説明**

Ceph がラグクライアントの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 の再接続タイムアウト

説明

MDS の再起動時にクライアントが再接続され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 ティック間隔

説明

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 分間隔

説明

ツリー上での再帰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

説明

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 クライアントの prealloc inos

説明

クライアントセッションごとに事前割り当てする inode 番号の数。

タイプ

32 ビット整数

デフォルト

1000

mds early reply

説明

MDS により、クライアントがジャーナルにコミットする前にリクエスト結果を確認できるかどうかを決定します。

タイプ

ブール値

デフォルト

true

MDS が tmap を使用する

説明

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 のデフォルトのディレクトリーハッシュ

説明

ディレクトリーフラグメント間でファイルをハッシュ化するために使用する関数。

タイプ

32 ビット整数

デフォルト

2 (つまり rjenkins)

mds log

説明

MDS がジャーナルメタデータを更新する場合は true に設定します (ベンチマークのみでは無効)。

タイプ

ブール値

デフォルト

true

mds log skip corrupted events (mds ログが破損しているイベントをスキップする)

説明

MDS がジャーナルの再生中に破損したジャーナルイベントをスキップするかどうかを決定します。

タイプ

ブール値

デフォルト

false

MDS ログの最大イベント

説明

Cephがトリミングを開始する前に、ジャーナルの最大イベント。制限を無効にするには -1 に設定します。

タイプ

32 ビット整数

デフォルト

-1

MDS ログの最大セグメント**説明**

Cephがトリミングを開始する前に、ジャーナル内のセグメント（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 ログの最大有効期限**説明**

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**説明**

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**説明**

ディレクトリー温度をサンプリングする頻度（断片化的な決定）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**説明**

Cephがメタデータを他のノードに複製するまでの最大温度。

タイプ

浮動小数点 (Float)

デフォルト**8000****MDS bal unreplicate threshold****説明**

Ceph が他のノードへのメタデータの複製を停止する前の最小温度。

タイプ

浮動小数点 (Float)

デフォルト**0****mds bal frag****説明**

MDS がディレクトリーを断片化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 bal split size****説明**

MDS がディレクトリーフラグメントを小さなビットに分割する前の最大のディレクトリーサイズ。

タイプ

32 ビット整数

デフォルト**10000****mds bal split rd****説明**

Ceph がディレクトリーのフラグメントを分割するまでの最大ディレクトリー読み取り温度。

タイプ

浮動小数点 (Float)

デフォルト**25000****mds bal split wr****説明**

Ceph がディレクトリーのフラグメントを分割するまでの最大ディレクトリー書き込み温度。

タイプ

浮動小数点 (Float)

デフォルト

10000

mds bal split bits

説明

ディレクトリーフラグメントを分割するビット数。

タイプ

32 ビット整数

デフォルト

3

mds bal merge size

説明

Ceph が隣接ディレクトリーフラグメントをマージしようとする前の最小ディレクトリーサイズ。

タイプ

32 ビット整数

デフォルト

50

MDS bal merge rd

説明

Ceph が隣接するディレクトリーフラグメントのマージ前の最小限の読み取り温度。

タイプ

浮動小数点 (Float)

デフォルト

1000

mds bal merge wr

説明

Ceph が隣接するディレクトリーのフラグメントをマージする前に最小の書き込み温度。

タイプ

浮動小数点 (Float)

デフォルト

1000

mds bal interval

説明

MDS ノード間のワークロードエクスチェンジの頻度 (秒単位)。

タイプ

32 ビット整数

デフォルト

10

mds bal fragment interval

説明

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

説明

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

説明

Ceph が停止する前に balancer を実行する反復数。テストの目的でのみ使用されます。

タイプ

32 ビット整数

デフォルト

-1

mds bal max until

説明

Ceph が停止するまでの balancer を実行する秒数。テストの目的でのみ使用されます。

タイプ

32 ビット整数

デフォルト

-1

mds bal mode

説明

MDS 負荷を計算する方法:

- 1 = ハイブリッド
- 2 = リクエストレートとレイテンシー。
- 3 = CPU 負荷

タイプ

32 ビット整数

デフォルト

0

mds bal min rebalance

説明

Ceph の移行前の最小サブツリーの温度。

タイプ

浮動小数点 (Float)

デフォルト

0.1

mds bal min start

説明

Ceph がサブツリーを検索するまでの最小サブツリーの温度。

タイプ

浮動小数点 (Float)

デフォルト

0.2

mds bal need min

説明

許可するターゲットサブツリーの最小分数。

タイプ

浮動小数点 (Float)

デフォルト

0.8

mds bal need max

説明

許可するターゲットサブツリーサイズの最大分数。

タイプ

浮動小数点 (Float)

デフォルト

1.2

mds bal midchunk

説明

Ceph は、ターゲットサブツリーサイズよりも大きな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

説明

Ceph は、ターゲットサブツリーサイズ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**説明**

Ceph が MDS マップから古い MDS ターゲットを削除する前に、 balancer の反復回数。

タイプ

32 ビット整数

デフォルト

5

mds bal target removal max**説明**

Ceph が MDS マップから古い MDS ターゲットを削除するまでの balancer 反復の最大数。

タイプ

32 ビット整数

デフォルト

10

MDS の再生間隔**説明**

standby-replay モード (ホットスタンバイ) モードの場合のジャーナルポーリング間隔。

タイプ

浮動小数点 (Float)

デフォルト

1

MDS のシャットダウンチェック**説明**

MDS のシャットダウン中にキャッシュをポーリングする間隔。

タイプ

32 ビット整数

デフォルト

0

MDS sash エクスポート**説明**

Ceph はノード間でサブツリーを無作為にエクスポートします (テストのみ)。

タイプ

32 ビット整数

デフォルト

0**MDS ｼrash フラグメント**

説明

Ceph は、無作為に断片化したり、ディレクトリーをマージします。

タイプ

32 ビット整数

デフォルト

0**マップでの MDS ダンプキャッシュ**

説明

Ceph は MDS キャッシュコンテンツを各 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**再結合後の MDS ダンプキャッシュ**

説明

Ceph は、リカバリー中にキャッシュを再度結合した後に 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 で散在(scatter)の検証**

説明

Ceph は、さまざまな scatter/gather invariants が true であることをアサートします（開発者用のみ）。

タイプ

ブール値

デフォルト

false**MDS デバッグスキッターstat**

説明

Ceph は、バリエントのさまざまな再帰的な統計が true であることをアサートします（開発者用のみ）。

タイプ

ブール値

デフォルト**false****mds debug frag****説明**

Ceph は、便利な場合にディレクトリーの断片化をバリエーション（開発者専用）で確認します。

タイプ**ブール値****デフォルト****false****MDS デバッグ認証のピン****説明**

デバッグ認証のピン（開発者の場合のみ）。

タイプ**ブール値****デフォルト****false****MDS デバッグサブツリー****説明**

debug サブツリーのvariants（開発者の場合のみ）。

タイプ**ブール値****デフォルト****false****MDS による mdstable の強制終了****説明**

Ceph は MDS テーブルコードに MDS の失敗を注入します（開発者用のみ）。

タイプ**32 ビット整数****デフォルト****0****MDS kill export at****説明**

Ceph は MDS の失敗をサブツリーエクスポートコードに挿入します（開発者用のみ）。

タイプ**32 ビット整数****デフォルト****0**

MDS kill import at

説明

Ceph は MDS の失敗をサブツリーのインポートコードに挿入します（開発者用のみ）。

タイプ

32 ビット整数

デフォルト

0

MDS の kill リンク

説明

Ceph は MDS の失敗をハードリンクコードに挿入します（開発者用のみ）。

タイプ

32 ビット整数

デフォルト

0

MDS の kill rename at

説明

Ceph は MDS の失敗を名前コードに挿入します（開発者用のみ）。

タイプ

32 ビット整数

デフォルト

0

mds wipe sessions

説明

Ceph は、起動時にすべてのクライアントセッションを削除します（テストのみ）。

タイプ

ブール値

デフォルト

0

MDS wipe ino prealloc

説明

Ceph は起動時に inode 事前割り当てメタデータを削除します（テストのみ）。

タイプ

ブール値

デフォルト

0

MDS skip ino

説明

起動時にスキップする inode 番号の数（テストのみ）。

タイプ

32 ビット整数

デフォルト

0

mds standby の名前**説明**

MDS デーモンは、この設定で指定された名前の別の MDS デーモンにスタンバイになります。

タイプ

文字列

デフォルト

該当なし

ランク用の MDS スタンバイ**説明**

MDS デーモンのインスタンスは、このランクの別の MDS デーモンインスタンスに対してスタンバイになります。

タイプ

32 ビット整数

デフォルト

-1

MDS のスタンバイ再生**説明**

MDS デーモンがアクティブな MDS (ホットスタンバイ) のログをポーリングして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

B.2. ジャーナル設定リファレンス

journaler でエントリーを分割可能**説明**

エントリーがストライプ境界にまたがることを許可します。

タイプ

ブール値

必須

いいえ

デフォルト

true

ジャーナル書き込みヘッド間隔

説明

ジャーナルヘッドオブジェクトを更新する頻度。

タイプ

整数

必須

いいえ

デフォルト

15

ジャーナルの事前フェッチ期間

説明

ジャーナル再生に先行するストライプ期間の数。

タイプ

整数

必須

いいえ

デフォルト

10

ジャーナルの先行ゼロ期間

説明

書き込み位置が 0 より進んだストライプ期間の数。

タイプ

整数

必須

いいえ

デフォルト

10

ジャーナルバッチ間隔

説明

人為的に発生する最大レイテンシー (秒単位)。

タイプ

double

必須

いいえ

デフォルト

.001

ジャーナルバッチの最大値

説明

フラッシュを遅延させる最大バイト。

タイプ

64 ビット未署名の整数

必須

いいえ

デフォルト

0

B.3. FUSE クライアント設定のリファレンス

本セクションでは、CephFS FUSE クライアントの設定オプションを紹介します。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 midpoint 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機能が必要な場合に、クライアントが機能のリリースを待つ秒数を設定します。

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。ストレージクラスターが CephFS を使用していない場合、この値は、Ceph Monitor ノードがストレージクラスター内の他の Ceph Monitor ノードと通信しようとする秒数を指します。Ceph Monitor ノードが他のノードに到達できない場合は、定義した秒数後にタイムアウトします。また、アクティブな `libvirt/librados` 操作が同時にタイムアウトします。タイムアウト値を設定すると、他の操作をブロックせずに、指定された時間間隔後にアプリケーションが失敗します。

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。以下を参照してください。

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

statsf 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

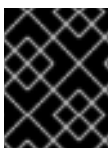
false に設定すると、ceph-fuse ユーティリティーは FUSE のパーミッションの適用に依存せずに独自のパーミッションチェックを行います。**false** に、クライアントの **acl type=posix_acl** オプションと共に設定し、POSIX ACL を有効にします。

タイプ

ブール値

デフォルト

true

開発者オプション**重要**

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

client_debug_getattr_caps

詳細

MDS からの応答に必要な機能が含まれているかどうかを確認します。

タイプ

ブール値

デフォルト

false

client_debug_inject_tick_delay

詳細

クライアントティックの間に人為的な遅延を追加します。

タイプ

整数

デフォルト

0

client_inject_fixed_oldest_tid**詳細, タイプ**

ブール値

デフォルト

false

client_inject_release_failure**詳細, タイプ**

ブール値

デフォルト

false

client_trace**詳細**

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

タイプ

文字列

デフォルト

"" (無効)