# Red Hat Ceph Storage 7

## 設定ガイド

Red Hat Ceph Storage の設定

Last Updated: 2024-03-27

# Red Hat Ceph Storage 7 設定ガイド

Red Hat Ceph Storage の設定

## 法律上の通知



## 概要

このドキュメントでは、ブート時と実行時に Red Hat Ceph Storage を設定する手順を説明します。また、設定の参考情報も掲載しています。 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 設定の基本

ストレージ管理者としては、Ceph の設定を表示する方法と、Red Hat Ceph Storage クラスターの Ceph 設定オプションを設定する方法について、基本的な理解が必要です。実行時に Ceph の設定オプションを表示、設定することができます。

**前提条件**

- Red Hat Ceph Storage ソフトウェアのインストール

## 1.1. CEPH の設定

すべての Red Hat Ceph Storage クラスターには、以下の項目を定義する設定があります。

- クラスター ID
- 認証設定
- Ceph デーモン
- ネットワーク設定
- ノード名およびアドレス
- キーリングへのパス
- OSD ログファイルへのパス
- 他のランタイムオプション

**cephadm** などのデプロイメントツールは、通常、初期の Ceph 設定ファイルを作成します。ただし、デプロイメントツールを使用して Red Hat Ceph Storage クラスターをブートストラップする場合には、独自に作成することができます。

**関連情報**

- **cephadm** と Ceph Orchestrator の詳細は、 Red Hat Ceph Storage オペレーションガイド を参照してください。

## 1.2. CEPH 設定データベース

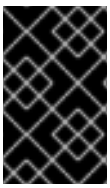
Ceph Monitor は、Ceph オプションの設定データベースを管理します。これにより、ストレージクラスター全体の設定オプションを格納することで、設定の管理を一元化します。Ceph の設定をデータベースに一元化することで、ストレージクラスターの管理を簡素化します。

Ceph がオプションの設定に使用する優先順位は以下のとおりです。

- コンパイルされたデフォルト値
- Ceph クラスター設定データベース
- ローカルの **ceph.conf** ファイル
- **ceph daemon DAEMON-NAME config set** または **ceph tell DAEMON-NAME injectargs** コマンドを使用したランタイムオーバーライド

ローカルの Ceph 設定ファイル (デフォルトでは **/etc/ceph/ceph.conf**) で定義できる Ceph オプションはまだいくつかあります。ただし、**ceph.conf** は Red Hat Ceph Storage 7 では非推奨となっています。

**cephadm** は、Ceph Monitor への接続、認証、および設定情報の取得のための最小限のオプションセットのみが含まれる基本的な **ceph.conf** ファイルを使用します。ほとんどの場合、**cephadm** は **mon_host** オプションのみを使用します。**mon_host** オプションのためだけに **ceph.conf** を使用するのを避けるために、DNS SRV レコードを使用して Monitor で操作を行います。



**重要**

Red Hat では、**assimilate-conf** 管理コマンドを使用して、有効なオプションを **ceph.conf** ファイルから設定データベースに移動することを推奨します。**assimilate-conf** の詳細については、管理用コマンドを参照してください。

Ceph では、実行時にデーモンの設定を変更することができます。この機能は、デバッグ設定の有効化/無効化によりログ出力を増減する場合に役立ちます。さらに、ランタイムの最適化にも使用できます。

**注記**

同じオプションが設定データベースと Ceph 設定ファイルに存在する場合、設定データベースのオプションの優先順位は Ceph 設定ファイルで設定されているものよりも低くなります。

### セクションおよびマスク

Ceph 設定ファイルで Ceph オプションをグローバルに、デーモンタイプごとに、または特定のデーモンごとに設定できるのと同様に、これらのセクションに従って設定データベースで Ceph オプションを設定することもできます。

| セクション | 説明 |
|---|---|
| **global** | すべてのデーモンおよびクライアントに影響します。 |
| **mon** | すべての Ceph Monitor に影響します。 |
| **mgr** | すべての Ceph Manager に影響します。 |
| **osd** | すべての Ceph OSD に影響します。 |
| **mds** | すべての Ceph Metadata Server に影響します。 |
| **client** | マウントされたファイルシステム、ブロックデバイス、RADOS Gateway など、すべての Ceph クライアントに影響します。 |

Ceph の設定オプションには、マスクを関連付けることができます。これらのマスクは、オプションを適用するデーモンやクライアントをさらに制限することができます。

マスクには 2 つの形式があります。

**type:location**

**type** は CRUSH プロパティーで、例えば **rack** や **host** などです。**location** は、プロパティータイプの値です。たとえば、**host:foo** は、**foo** ホストで実行しているデーモンまたはクライアントにのみオプションを制限します。

**例**

```
ceph config set osd/host:magna045 debug_osd 20
```

**class:device-class**

**device-class** は、**hdd** や **ssd** など、CRUSH デバイスクラスの名前です。たとえば、**class:ssd** は、ソリッドステートドライブ (SSD) ベースの Ceph OSD にのみオプションを制限します。このマスクは、クライアントの非 OSD デーモンには影響しません。

**例**

```
ceph config set osd/class:hdd osd_max_backfills 8
```

### 管理コマンド

Ceph 設定データベースは、サブコマンド **ceph config ACTION** で管理できます。実施できるアクションは以下のとおりです。

**ls**

利用可能な設定オプションを一覧表示します。

**dump**

ストレージクラスターのオプションの設定データベース全体をダンプします。

**get WHO**

特定のデーモンまたはクライアントの設定をダンプします。例えば、**WHO** は **mds.a** のようなデーモンになります。

**set WHO OPTION VALUE**

Ceph 設定データベースに設定オプションを設定します。**WHO** はターゲットデーモン、**OPTION** は設定するオプション、**VALUE** は必要な値です。

**show WHO**

実行中のデーモンについて、報告された実行中の設定を表示します。ローカル設定ファイルが使用されていたり、コマンドラインや実行時にオプションが上書きされていたりすると、これらのオプションは Ceph Monitor が保存するオプションとは異なる場合があります。また、オプション値のソースは出力の一部として報告されます。

**assimilate-conf -i INPUT_FILE -o OUTPUT_FILE**

**INPUT_FILE** から設定ファイルを同化し、有効なオプションを Ceph Monitor の設定データベースに移動します。認識できない、無効な、または Ceph Monitor で制御できないオプションは、**OUTPUT_FILE** に格納された省略された設定ファイルで返されます。このコマンドは、従来の設定ファイルから一元化された設定データベースに移行する際に便利です。設定を同化する際に、Monitor や他のデーモンが同じオプションセットに異なる値を設定している場合、最終的な結果はファイルを同化する順序に依存することに注意してください。

**help OPTION -f json-pretty**

特定の **OPTION** のヘルプを JSON 形式の出力で表示します。

### 関連情報

- コマンドの詳細は、**実行時における特定設定の定義** を参照してください。

## 1.3. CEPH メタ変数の使用

メタ変数は、Ceph ストレージクラスターの設定を大幅に簡素化します。メタ変数が設定値に設定されると、Ceph はそのメタ変数を具体的な値にデプロイメントします。

メタ変数は、Ceph 設定ファイルの **[global]** セクション、**[osd]** セクション、**[mon]** セクション、または **[client]** セクション内で使用すると非常に強力です。しかし、管理用ソケットでも使用可能です。Ceph メタ変数は、Bash のシェル拡張に似ています。

Ceph は以下のメタ変数をサポートしています。

**$cluster**

**説明**

Ceph ストレージクラスター名にデプロイメントします。同じハードウェアで複数の Ceph ストレージクラスターを実行する場合に便利です。

**例**

**/etc/ceph/$cluster.keyring**

**デフォルト**

**ceph**

**$type**

**説明**

インスタントデーモンのタイプに応じて、**osd** または **mon** のいずれかにデプロイメントします。

**例**

**/var/lib/ceph/$type**

**$id**

**説明**

デーモン識別子に拡張します。**osd.0** の場合、これは **0** になります。

**例**

**/var/lib/ceph/$type/$cluster-$id**

**$host**

**説明**

インスタントデーモンのホスト名にデプロイメントします。

**$name**

**説明**

**$type.$id** までデプロイメントします。

**例**

**/var/run/ceph/$cluster-$name.asok**

## 1.4. ランタイム時に CEPH 設定をリスト表示

Ceph 設定ファイルは、ブート時および実行時に表示することができます。

**前提条件**

- Ceph OSD ノードへのルートレベルのアクセス。
- 管理キーリングへのアクセス。

**手順**

1. ランタイム設定を表示するには、デーモンを実行している Ceph ノードにログインして以下を実行します。

   **構文**

   ```
   ceph daemon DAEMON_TYPE.ID config show
   ```

   **osd.0** の設定を確認するには、**osd.0** を含むノードにログインして以下のコマンドを実行します。

   **例**

   ```
   [root@osd ~]# ceph daemon osd.0 config show
   ```

2. 追加のオプションについては、デーモンと **help** を指定します。

   **例**

   ```
   [root@osd ~]# ceph daemon osd.0 help
   ```

## 1.5. 実行時における特定設定の表示

Red Hat Ceph Storage の設定は、Ceph Monitor ノードから実行時に確認することができます。

**前提条件**

- 稼働中の Red Hat Ceph Storage クラスターがある。
- Ceph Monitor ノードへの root レベルのアクセス。

**手順**

1. Ceph ノードにログインして以下を実行します。

   **構文**

   ```
   ceph daemon DAEMON_TYPE.ID config get PARAMETER
   ```

   **例**

   ```
   [root@mon ~]# ceph daemon osd.0 config get public_addr
   ```

## 1.6. 実行時における特定設定の定義

実行時に特定の Ceph 設定を定義するには、**ceph config set** コマンドを使用します。

**前提条件**

- 稼働中の Red Hat Ceph Storage クラスターがある。
- Ceph Monitor または OSD ノードへの root レベルのアクセス

**手順**

1. すべての Monitor または OSD デーモンの設定を定義します。

   **構文**

   ```
   ceph config set DAEMON CONFIG-OPTION VALUE
   ```

   **例**

   ```
   [root@mon ~]# ceph config set osd debug_osd 10
   ```

2. オプションと値が設定されていることを検証します。

   **例**

   ```
   [root@mon ~]# ceph config dump
   osd      advanced debug_osd  10/10
   ```

   - すべてのデーモンから設定オプションを削除するには、以下を実行します。

     **構文**

     ```
     ceph config rm DAEMON CONFIG-OPTION VALUE
     ```

     **例**

     ```
     [root@mon ~]# ceph config rm osd debug_osd
     ```

   - 特定のデーモンを設定するには、以下を実行します。

     **構文**

     ```
     ceph config set DAEMON.DAEMON-NUMBER CONFIG-OPTION VALUE
     ```

     **例**

     ```
     [root@mon ~]# ceph config set osd.0 debug_osd 10
     ```

   - 指定したデーモンに設定が定義されていることを確認するには、以下を実行します。

**例**

```
[root@mon ~]# ceph config dump
osd.0      advanced debug_osd     10/10
```

- 特定のデーモンの設定を削除するには、次のコマンドを実行します。

**構文**

```
ceph config rm DAEMON.DAEMON-NUMBER CONFIG-OPTION
```

**例**

```
[root@mon ~]# ceph config rm osd.0 debug_osd
```

**注記**

設定データベースからのオプションの読み取りをサポートしていないクライアントを使用している場合、または他の理由でクラスターの設定を変更するために **ceph.conf** を使用する必要がある場合は、次のコマンドを実行します。

```
ceph config set mgr mgr/cephadm/manage_etc_ceph_ceph_conf false
```

**ceph.conf** ファイルを維持してストレージクラスター全体に配布する必要があります。

## 1.7. OSD メモリーターゲット

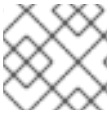
BlueStore は、**osd_memory_target** 設定オプションを使用して、OSD ヒープメモリーの使用を指定されたターゲットサイズで保持します。

**osd_memory_target** オプションは、システムで利用可能な RAM に基づいて OSD メモリーを設定します。TCMalloc がメモリーアロケーターとして設定されており、BlueStore の **bluestore_cache_autotune** オプションが **true** に設定されている場合、このオプションを使用します。

Ceph OSD のメモリーキャッシングは、ブロックデバイスが低速である場合に重要となります (例えば、従来のハードドライブの場合)。キャッシュヒットのメリットがソリッドステートドライブの場合よりもはるかに大きいからです。ただし、ハイパーコンバージドインフラストラクチャー (HCI) や他のアプリケーションなど、他のサービスと OSD を共存させる場合には、この点を考慮する必要があります。

### 1.7.1. OSD メモリーターゲットの設定

ストレージクラスター内のすべての OSD、または特定の OSD に最大メモリーしきい値を設定するには、**osd_memory_target** オプションを使用します。**osd_memory_target** オプションを 16 GB に設定した OSD は、最大 16 GB のメモリーを使用することができます。



**注記**

個々の OSD の設定オプションは、すべての OSD に対する設定よりも優先されます。

**前提条件**

- 稼働中の Red Hat Ceph Storage クラスターがある。
- ストレージクラスター内のすべてのホストへの root レベルのアクセス

**手順**

- ストレージクラスター内のすべての OSD に **osd_memory_target** を設定するには、以下を実行します。

  **構文**

  ```
  ceph config set osd osd_memory_target VALUE
  ```

  **VALUE** は、ストレージクラスター内の各 OSD に割り当てるメモリーのギガバイト数です。

- ストレージクラスター内の特定の OSD に **osd_memory_target** を設定するには、以下を実行します。

  **構文**

  ```
  ceph config set osd.id osd_memory_target VALUE
  ```

  **.id** は OSD の ID で、 **VALUE** は特定の OSD に割り当てるメモリーの GB 数です。たとえば、ID 8 の OSD が最大 16 ギガバイトのメモリーを使用するように設定するには、以下を実行します。

  **例**

  ```
  [ceph: root@host01 /]# ceph config set osd.8 osd_memory_target 16G
  ```

- ある個別の OSD がある最大量のメモリーを使用するように設定し、残りの OSD が別の量を使用するように設定するには、まず個別の OSD を指定します。

  **例**

  ```
  [ceph: root@host01 /]# ceph config set osd osd_memory_target 16G
  [ceph: root@host01 /]# ceph config set osd.8 osd_memory_target 8G
  ```

**関連情報**

- OSD のメモリー使用量を自動調整するように Red Hat Ceph Storage を設定するには、**オペレーションガイド** の **OSD メモリーの自動チューニング** を参照してください。

## 1.8. OSD メモリーの自動チューニング

OSD デーモンは、**osd_memory_target** 設定オプションに基づいてメモリー消費を調整します。**osd_memory_target** オプションは、システムで利用可能な RAM に基づいて OSD メモリーを設定します。

Red Hat Ceph Storage が他のサービスとメモリーを共有しない専用ノードにデプロイされている場合、**cephadm** は RAM の合計量とデプロイされた OSD の数に基づいて OSD ごとの消費を自動的に調整します。

**重要**

デフォルトでは、Red Hat Ceph Storage 5.1 で **osd_memory_target_autotune** パラメーターは **true** に設定されます。

**構文**

```
ceph config set osd osd_memory_target_autotune true
```

OSD の追加や OSD の置き換えなど、クラスターのメンテナンスのためにストレージクラスターを Red Hat Ceph Storage 5.0 にアップグレードした後、Red Hat は **osd_memory_target_autotune** パラメーターを **true** に設定し、システムメモリーごとに osd メモリーを自動調整することを推奨します。

Cephadm は、**mgr/cephadm/autotune_memory_target_ratio** の割合で始まります。これはデフォルトでは、システムの合計 RAM 容量の **0.7** になります。これから、非 OSDS や **osd_memory_target_autotune** が false の OSD などの自動調整されないデーモンによって消費されるメモリー分を引き、残りの OSD で割ります。

**osd_memory_target** パラメーターは、以下のように計算されます。

**構文**

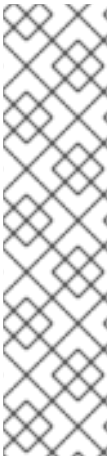
```
osd_memory_target = TOTAL_RAM_OF_THE_OSD * (1048576) * (autotune_memory_target_ratio) / NUMBER_OF_OSDS_IN_THE_OSD_NODE - (SPACE_ALLOCATED_FOR_OTHER_DAEMONS)
```

SPACE_ALLOCATED_FOR_OTHER_DAEMONS には、任意で以下のデーモン領域の割り当てを含めることができます。

- Alertmanager: 1 GB
- Grafana: 1 GB
- Ceph Manager: 4 GB
- Ceph Monitor: 2 GB
- Node-exporter: 1 GB
- Prometheus: 1 GB

たとえば、ノードに OSD が 24 個あり、251 GB の RAM 容量がある場合、 **osd_memory_target** は **7860684936** になります。

最後のターゲットは、オプションとともに設定データベースに反映されます。**MEM LIMIT** 列の **ceph orch ps** の出力で、制限と各デーモンによって消費される現在のメモリーを確認できます。



**注記**

Red Hat Ceph Storage 5.1 では、**osd_memory_target_autotune** のデフォルト設定 **true** は、コンピュートサービスと Ceph ストレージサービスが共存するハイパーコンバージドインフラストラクチャーでは適切ではありません。ハイパーコンバージドインフラストラクチャーでは、**autotune_memory_target_ratio** を **0.2** に設定して、Ceph のメモリー消費を減らすことができます。

**例**

```
[ceph: root@host01 /]# ceph config set mgr
mgr/cephadm/autotune_memory_target_ratio 0.2
```

ストレージクラスターで OSD の特定のメモリーターゲットを手動で設定できます。

**例**

```
[ceph: root@host01 /]# ceph config set osd.123 osd_memory_target 7860684936
```

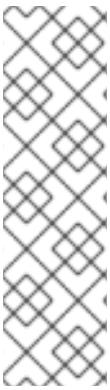
ストレージクラスターで OSD ホストの特定のメモリーターゲットを手動で設定できます。

**構文**

```
ceph config set osd/host:HOSTNAME osd_memory_target TARGET_BYTES
```

**例**

```
[ceph: root@host01 /]# ceph config set osd/host:host01 osd_memory_target 1000000000
```



**注記**

**osd_memory_target_autotune** を有効にすると、既存の手動の OSD メモリーターゲット設定が上書きされます。**osd_memory_target_autotune** オプションまたはその他の同様のオプションが有効になっている場合でもデーモンメモリーがチューニングされないようにするには、ホストに **_no_autotune_memory** ラベルを設定します。

**構文**

```
ceph orch host label add HOSTNAME _no_autotune_memory
```

自動チューニングオプションを無効にし、特定のメモリーターゲットを設定して、OSD をメモリー自動チューニングから除外できます。

**例**

```
[ceph: root@host01 /]# ceph config set osd.123 osd_memory_target_autotune false
[ceph: root@host01 /]# ceph config set osd.123 osd_memory_target 16G
```

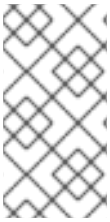
## 1.9. MDS メモリーキャッシュの制限

MDS サーバーは、そのメタデータを別のストレージプール (**cephfs_metadata**) に保持し、Ceph OSD

のユーザーです。Ceph File System の場合、MDS サーバーはストレージクラスター内の単一のストレージデバイスだけでなく、Red Hat Ceph Storage クラスター全体をサポートする必要があるため、特にワークロードが小/中サイズのファイルで設定されている場合 (データに対するメタデータの比率が高い)、メモリー要件が大きくなる可能性があります。

例: **mds_cache_memory_limit** を 2000000000 バイトに設定

```
ceph_conf_overrides:
  osd:
    mds_cache_memory_limit=2000000000
```



**注記**

メタデータを多用するワークロードを持つ大規模な Red Hat Ceph Storage クラスターでは、MDS サーバーを他のメモリーを多用するサービスと同じノードに置かないでください。そうすることで、より多くのメモリー (たとえば 100 GB を超えるサイズ) を MDS に割り当てることができます。

**関連情報**

- **Red Hat Ceph Storage ファイルシステムガイド**の **メタデータサーバーのキャッシュサイズ制限** を参照してください。
- 特定のオプションの詳細や使用方法は、**設定オプション** の一般的な Ceph 設定オプションを参照してください。

# 第2章 CEPH ネットワーク設定

ストレージ管理者は、Red Hat Ceph Storage クラスターが動作するネットワーク環境を理解し、それに応じて Red Hat Ceph Storage を設定する必要があります。Ceph のネットワークオプションを理解して設定することで、ストレージクラスター全体のパフォーマンスと信頼性を最適化することができます。

**前提条件**

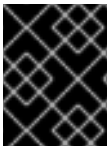
- ネットワーク接続
- Red Hat Ceph Storage ソフトウェアのインストール

## 2.1. CEPH のネットワーク設定

高性能な Red Hat Ceph Storage クラスターを構築するには、ネットワークの設定が重要です。Ceph ストレージクラスターは、Ceph クライアントに代わって要求のルーティングやディスパッチを実行しません。代わりに、Ceph クライアントは Ceph OSD デーモンに直接要求を出します。Ceph OSD は Ceph クライアントに代わってデータレプリケーションを実行するため、レプリケーションおよび他の要素によって Ceph ストレージクラスターのネットワークに追加の負荷がかかります。

Ceph には、すべてのデーモンに適用される 1 つのネットワーク設定要件があります。Ceph 設定ファイルは、各デーモンに **host** を指定する必要があります。

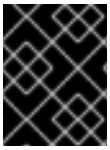
**cephadm** などの一部のデプロイメントユーティリティーは、設定ファイルを作成してくれます。デプロイメントユーティリティーがこれらの値を設定する場合は、設定しないでください。



**重要**

**host** オプションは、FQDN ではなく、ノードの短縮名です。IP アドレスではありません。

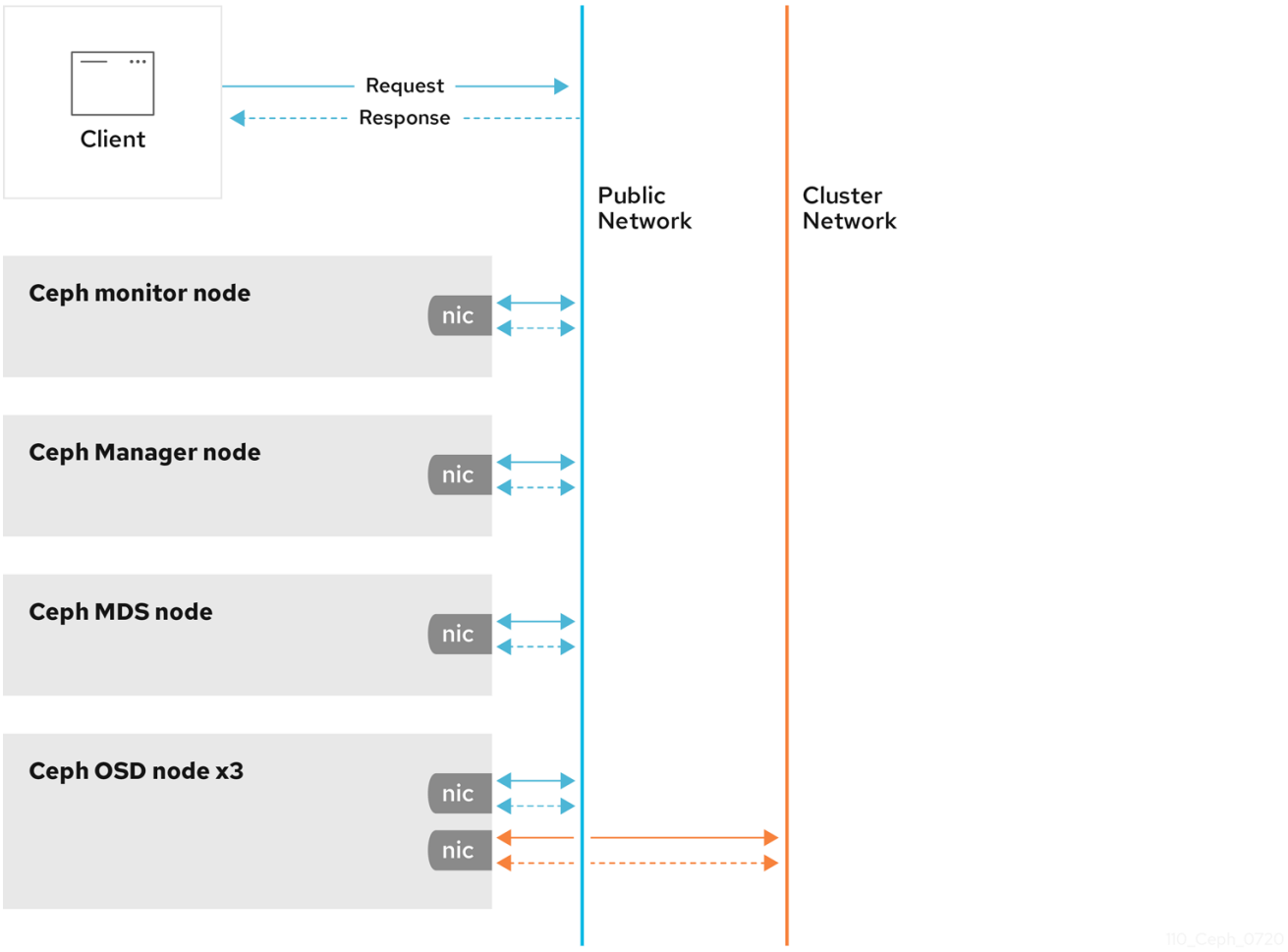
すべての Ceph ク ラスターは、パブリックネットワークを使用する必要があります。ただし、内部のクラスターネットワークを指定しない限り、Ceph は 1 つのパブリックネットワークを想定します。Ceph はパブリックネットワークでのみ機能しますが、大規模なストレージクラスターの場合は、クラスター関連のトラフィックのみを伝送する第 2 のプライベートネットワークを使用すると、パフォーマンスが大幅に向上します。



**重要**

Red Hat では、Ceph ストレージクラスターを 2 つのネットワークで運用することを推奨しています (1 つのパブリックネットワークと 1 つのプライベートネットワーク)。

2 つのネットワークをサポートするには、各 Ceph Node に複数のネットワークインターフェイスカード (NIC) が必要になります。




2 つの別々のネットワークを運用することを検討する理由はいくつかあります。

- **パフォーマンス:** Ceph OSD は Ceph クライアントのデータレプリケーションを処理します。Ceph OSD がデータを複数回複製すると、Ceph OSD 間のネットワーク負荷は、Ceph クライアントと Ceph ストレージクラスター間のネットワーク負荷をすぐに阻害してしまいます。これによりレイテンシーが発生し、パフォーマンスに問題が生じます。リカバリーやリバランシングを行うと、パブリックネットワーク上で大きなレイテンシーが発生します。

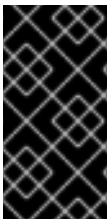
- **セキュリティー:** 通常、多くのユーザーはサービス拒否 (DoS) 攻撃と呼ばれる攻撃に関与します。Ceph OSD 間のトラフィックが中断されると、ピアリングが失敗し、配置グループが **active + clean** 状態を反映しなくなり、ユーザーがデータを読み書きできなくなる可能性があります。この種の攻撃に対抗するには、インターネットに直接接続しない、完全に独立したクラスターネットワークを維持することが有効です。

ネットワーク設定の定義は必要ありません。Ceph はパブリックネットワークでのみ機能するので、Ceph デーモンを実行するすべてのホストでパブリックネットワークが設定されている必要があります。しかし、Ceph では、複数の IP ネットワークやサブネットマスクなど、より具体的な条件をパブリックネットワークに設定することができます。また、OSD ハートビート、オブジェクトのレプリケーション、およびリカバリートラフィックを処理するために、別のクラスターネットワークを構築することもできます。

設定で定義する IP アドレスと、ネットワーククライアントがサービスにアクセスする際に使用する公開用の IP アドレスを混同しないようにしてください。通常、内部 IP ネットワークは **192.168.0.0** または **10.0.0.0** です。

**注記**

Ceph はサブネットに CIDR 表記を使用します (例: **10.0.0.0/24**)。



**重要**

パブリックネットワークまたはプライベートネットワークのいずれかに複数の IP アドレスとサブネットマスクを指定する場合、ネットワーク内のサブネットは相互にルーティング可能でなければなりません。さらに、各 IP アドレスとサブネットを IP テーブルに含め、必要に応じてポートを開くようにしてください。

ネットワークの設定が完了したら、クラスターの再起動や各デーモンの再起動を行います。Ceph デーモンは動的にバインドするので、ネットワーク設定を変更してもクラスター全体を一度に再起動する必要はありません。

**関連情報**

- 特定のオプションの説明や使用方法は、**Red Hat Ceph Storage 設定ガイド**の 付録 B の共通オプションを参照してください。

## 2.2. CEPH ネットワークメッセンジャー

メッセンジャーは Ceph ネットワーク層の実装です。Red Hat は 2 種類のメッセンジャーをサポートしています。

- **simple**
- **async**

Red Hat Ceph Storage 6 以降では、**async** がデフォルトのメッセンジャータイプです。messenger タイプを変更するには、Ceph 設定ファイルの **[global]** セクションに **ms_type** 設定を指定します。

**注記**

**async** mesenger では、Red Hat は **posix** トランスポートタイプをサポートしますが、現在 **rdma** または **dpdk** をサポートしていません。デフォルトでは、Red Hat Ceph Storage 6 以降の **ms_type** 設定は **async+posix** を反映します。ここで、**async** はメッセンジャータイプで、**posix** はトランスポートタイプです。

### SimpleMessenger

**SimpleMessenger** 実装は、1 ソケットあたり 2 つのスレッドを持つ TCP ソケットを使用します。Ceph は、各論理セッションを接続に関連付けます。パイプは、各メッセージの入力と出力を含む接続を処理します。**SimpleMessenger** は、**posix** トランスポートタイプに有効ですが、**rdma**、**dpdk** などの他のトランスポートタイプには有効ではありません。

### AsyncMessenger

したがって、**AsyncMessenger** は、Red Hat Ceph Storage 6 以降のデフォルトのメッセンジャータイプです。Red Hat Ceph Storage 6 以降では、**AsyncMessenger** 実装は、接続用に固定サイズのスレッドプールを持つ TCP ソケットを使用します。これは、レプリカまたはイレイジャーコードチャンクの最大数と同じでなければなりません。CPU 数が少なかったり、サーバーあたりの OSD 数が多かったりしてパフォーマンスが低下する場合は、スレッドカウントを低い値に設定することができます。

**注記**

現時点で、Red Hat は **rdma**、**dpdk** などの他のトランスポートタイプをサポートしていません。

**関連情報**

- 特定のオプションの説明や使用方法は、**Red Hat Ceph Storage 設定ガイド**の **付録 B** の AsyncMessenger オプションを参照してください。
- Ceph messenger バージョン 2 プロトコルでの **伝送時暗号化** の使用に関する詳細は、**Red Hat Ceph Storage アーキテクチャーガイド** を参照してください。

## 2.3. パブリックネットワークの設定

Ceph ネットワークを設定するには、**cephadm** シェル内で **config set** コマンドを使用します。ネットワーク設定で定義する IP アドレスは、ネットワーククライアントがサービスにアクセスする際に使用する公開用の IP アドレスと異なる点に注意してください。

Ceph は、パブリックネットワークとだけ完全に機能します。しかし、Ceph では、複数の IP ネットワークなど、より具体的な条件をパブリックネットワークに設定することができます。

また、OSD ハートビート、オブジェクトのレプリケーション、およびリカバリートラフィックを処理するために、別のプライベートクラスターネットワークを構築することもできます。プライベートネットワークの詳細は、**プライベートネットワークの設定** を参照してください。

**注記**

Ceph はサブネットに CIDR 表記を使用します (例: 10.0.0.0/24)。通常、内部 IP ネットワークは 192.168.0.0/24 または 10.0.0.0/24 です。

**注記**

パブリックネットワークまたはクラスターネットワークのいずれかに複数の IP アドレスを指定する場合、ネットワーク内のサブネットは相互にルーティング可能でなければなりません。さらに、各 IP アドレスを IP テーブルに含め、必要に応じてポートを開くようにしてください。

パブリックネットワークの設定では、特にパブリックネットワークの IP アドレスとサブネットを定義することができます。

**前提条件**

- Red Hat Ceph Storage ソフトウェアのインストール

**手順**

1. **cephadm** シェルにログインします。

   **例**

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

2. サブネットを使用してパブリックネットワークを設定します。

**構文**

```
ceph config set mon public_network IP_ADDRESS_WITH_SUBNET
```

**例**

```
[ceph: root@host01 /]# ceph config set mon public_network 192.168.0.0/24
```

3. ストレージクラスター内のサービスの一覧を取得します。

**例**

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

4. デーモンを再起動します。Ceph デーモンは動的にバインドするので、特定のデーモンのネットワーク設定を変更してもクラスター全体を一度に再起動する必要はありません。

**例**

```
[ceph: root@host01 /]# ceph orch restart mon
```

5. オプション: クラスターを再起動する場合は、root ユーザーとして管理ノードで **systemctl** コマンドを実行します。

**構文**

```
systemctl restart ceph-FSID_OF_CLUSTER.target
```

**例**

```
[root@host01 ~]# systemctl restart ceph-1ca9f6a8-d036-11ec-8263-fa163ee967ad.target
```

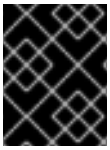
**関連情報**

- 特定のオプションの説明や使用方法は、**Red Hat Ceph Storage 設定ガイド**の 付録 B の共通オプションを参照してください。

## 2.4. プライベートネットワークの設定

ネットワーク設定の定義は必要ありません。Ceph では、クラスターネットワーク (**プライベートネットワーク** としても知られる) を特に設定しない限り、パブリックネットワーク上のすべてのホストが動作している必要があります。

クラスターネットワークを作成した場合、OSD はハートビート、オブジェクトのレプリケーション、およびリカバリートラフィックをクラスターネットワーク上でルーティングします。これにより、単一のネットワークを使用する場合と比較して、パフォーマンスが向上します。



**重要**

セキュリティー強化のためは、クラスターネットワークにはパブリックネットワークやインターネットからアクセスできないようにしてください。

クラスターネットワークを割り当てるには、**cephadm bootstrap** コマンドで **--cluster-network** オプションを使用します。指定するクラスターネットワークには、CIDR 表記のサブネットを定義する必要があります (例: 10.90.90.0/24 または fe80::/64)。

ブースター後に **cluster_network** を設定することもできます。

**前提条件**

- Ceph ソフトウェアリポジトリーへのアクセス。
- 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**

- ストレージクラスター内の Monitor ノードとして使用する最初のノードから、**cephadm bootstrap** コマンドを実行します。コマンドに **--cluster-network** オプションを追加します。

  **構文**

  ```
  cephadm bootstrap --mon-ip IP-ADDRESS --registry-url registry.redhat.io --registry-username USER_NAME --registry-password PASSWORD --cluster-network NETWORK-IP-ADDRESS
  ```

  **例**

  ```
  [root@host01 ~]# cephadm bootstrap --mon-ip 10.10.128.68 --registry-url registry.redhat.io --registry-username myuser1 --registry-password mypassword1 --cluster-network 10.10.0.0/24
  ```

- ブートストラップ後に **cluster_network** を設定するには、**config set** コマンドを実行し、デーモンを再デプロイします。

  1. **cephadm** シェルにログインします。

     **例**

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

  2. サブネットを使用してクラスターネットワークを設定します。

     **構文**

     ```
     ceph config set global cluster_network IP_ADDRESS_WITH_SUBNET
     ```

     **例**

     ```
     [ceph: root@host01 /]# ceph config set global cluster_network 10.10.0.0/24
     ```

  3. ストレージクラスター内のサービスの一覧を取得します。

     **例**

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

4. デーモンを再起動します。Ceph デーモンは動的にバインドするので、特定のデーモンのネットワーク設定を変更してもクラスター全体を一度に再起動する必要はありません。

   **例**

   ```
   [ceph: root@host01 /]# ceph orch restart mon
   ```

5. オプション: クラスターを再起動する場合は、root ユーザーとして管理ノードで **systemctl** コマンドを実行します。

   **構文**

   ```
   systemctl restart ceph-FSID_OF_CLUSTER.target
   ```

   **例**

   ```
   [root@host01 ~]# systemctl restart ceph-1ca9f6a8-d036-11ec-8263-fa163ee967ad.target
   ```

**関連情報**

- **cephadm bootstrap** の呼び出し方法の詳細は、**Red Hat Ceph Storage インストールガイド** の **新しいストレージクラスターのブートストラップ** セクションを参照してください。

## 2.5. 複数のパブリックネットワークをクラスターに設定する

ユーザーが複数のネットワークサブネットに属するホスト上に Ceph Monitor デーモンを配置したい場合は、クラスターに対して複数のパブリックネットワークを設定する必要があります。使用例としては、OpenShift Data Foundation の Metro DR の Advanced Cluster Management (ACM) に使用されるストレッチクラスターモードがあります。

ブートストラップ中およびブートストラップの完了後に、クラスターに対して複数のパブリックネットワークを設定できます。

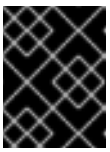
**前提条件**

- ホストを追加する前に、Red Hat Ceph Storage クラスターが実行されていることを確認してください。

**手順**

1. 複数のパブリックネットワークで設定された Ceph クラスターをブートストラップします。

   a. **mon** パブリックネットワークセクションを含む **ceph.conf** ファイルを準備します。

   **重要**

   ブートストラップに使用される現在のホスト上で、提供されたパブリックネットワークの少なくとも 1 つを設定する必要があります。

   **構文**

```
[mon]
public_network = PUBLIC_NETWORK1, PUBLIC_NETWORK2
```

**例**

```
[mon]
public_network = 10.40.0.0/24, 10.41.0.0/24, 10.42.0.0/24
```

これは、ブートストラップ用に 3 つのパブリックネットワークが提供される例です。

b. **ceph.conf** ファイルを入力として指定して、クラスターをブートストラップします。

**注記**

ブートストラップ中に、指定する他の引数を含めることができます。

**構文**

```
cephadm --image IMAGE_URL bootstrap --mon-ip MONITOR_IP -c
PATH_TO_CEPH_CONF
```

**注記**

あるいは、**IMAGE_URL** の代わりに **IMAGE_ID** (**13ea90216d0be03003d12d7869f72ad9de5cec9e54a27fd308e01e467c0d4a0a** など) を使用できます。

**例**

```
[root@host01 ~]# cephadm –image cp.icr.io/cp/ibm-ceph/ceph-5-rhel8:latest bootstrap –
mon-ip 10.40.0.0/24 -c /etc/ceph/ceph.conf
```

2. 新しいホストをサブネットに追加します。

**注記**

追加されるホストは、アクティブなマネージャーが実行されているホストからアクセス可能である必要があります。

a. クラスターの公開 SSH キーを新しいホストの root ユーザーの **authorized_keys** ファイルにインストールします。

**構文**

```
ssh-copy-id -f -i /etc/ceph/ceph.pub root@NEW_HOST
```

**例**

```
[root@host01 ~]# ssh-copy-id -f -i /etc/ceph/ceph.pub root@host02
[root@host01 ~]# ssh-copy-id -f -i /etc/ceph/ceph.pub root@host03
```

b. **cephadm** シェルにログインします。

**例**

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

c. 新しいホストを Ceph クラスターに追加します。

**構文**

```
ceph orch host add NEW_HOST IP [LABEL1 ...]
```

**例**

```
[root@host01 ~]# ceph orch host add host02 10.10.0.102 label1
[root@host01 ~]# ceph orch host add host03 10.10.0.103 label2
```

**注記**

- ホスト IP アドレスを明示的に指定することを推奨します。IP が指定されていない場合、ホスト名は DNS 経由ですぐに解決され、その IP が使用されます。
- 新しいホストにすぐにラベルを付けるために、1 つ以上のラベルを含めることもできます。たとえば、デフォルトでは、**_admin** ラベルにより、cephadm は **ceph.conf** ファイルと **client.admin** キーリングファイルのコピーを **/etc/ceph** ディレクトリーに保持します。

3. パブリックネットワークパラメーターのネットワーク設定を実行中のクラスターに追加します。サブネットがコンマで区切られていること、およびサブネットがサブネット/マスク形式でリストされていることを確認してください。

**構文**

```
ceph config set mon public_network "SUBNET_1,SUBNET_2, ..."
```

**例**

```
[root@host01 ~]# ceph config set mon public_network "192.168.0.0/24, 10.42.0.0/24, ..."
```

必要に応じて、**mon** 仕様を更新して、指定されたサブネット内のホストに **mon** デーモンを配置します。

**関連情報**

- ホストの追加の詳細は、**Red Hat Ceph Storage インストールガイド** の **ホストの追加** を参照してください。
- **ストレッチクラスターの** 詳細は、**Red Hat Ceph Storage 管理ガイド** の Ceph ストレージのストレッチクラスターを参照してください。

## 2.6. デフォルトの CEPH ポート用にファイアウォールルールが設定されていることの確認

デフォルトでは、Red Hat Ceph Storage デーモンは TCP ポート 6800-7100 を使用してクラスター内の他のホストと通信します。ホストのファイアウォールがこれらのポートで接続できることを確認できます。

**注記**

ネットワークに専用のファイアウォールがある場合は、この手順に加えてその設定を確認する必要がある場合があります。詳細は、ファイアウォールのドキュメントを参照してください。

詳細は、ファイアウォールのドキュメントを参照してください。

**前提条件**

- ホストへのルートレベルのアクセス。

**手順**

1. ホストの **iptables** 設定を確認します。
   a. アクティブなルールを一覧表示します。

   ```
   [root@host1 ~]# iptables -L
   ```

   b. TCP ポート 6800-7100 の接続を制限するルールが存在しないことを確認します。

   **例**

   ```
   REJECT all -- anywhere anywhere reject-with icmp-host-prohibited
   ```

2. ホストの **firewalld** 設定を確認します。
   a. ホストで開いているポートを一覧表示します。

   **構文**

   ```
   firewall-cmd --zone ZONE --list-ports
   ```

   **例**

   ```
   [root@host1 ~]# firewall-cmd --zone default --list-ports
   ```

   b. 範囲が TCP ポート 6800-7100 に含まれていることを確認します。

## 2.7. CEPH MONITOR ノードのファイアウォール設定

messenger バージョン 2 プロトコルの導入により、ネットワーク上のすべての Ceph トラフィックの暗号化を有効にすることができます。メッセンジャー v2 の **secure** モード設定は、Ceph デーモンと Ceph クライアント間の通信を暗号化し、エンドツーエンドの暗号化を提供します。

### メッセンジャー v2 プロトコル

Ceph の有線プロトコルの 2 つ目のバージョンである **msgr2** には、以下の新機能が含まれています。

- 安全なモードは、ネットワークを介したすべてのデータの移動を暗号化します。
- 認証ペイロードのカプセル化による改善。
- 機能のアドバタイズおよびネゴシエーションの改善。

Ceph デーモンは、レガシー、v1 互換、および新しい v2 互換の Ceph クライアントを同じストレージクラスターに接続することができるように、複数のポートにバインドします。Ceph Monitor デーモンに接続する Ceph クライアントまたはその他の Ceph デーモンは、まず **v2** プロトコルの使用を試みますが、可能でない場合は古い **v1** プロトコルが使用されます。デフォルトでは、メッセンジャープロトコル **v1** と **v2** の両方が有効です。新規の v2 ポートは 3300 で、レガシー v1 ポートはデフォルトで 6789 になります。

#### 前提条件

- 稼働中の Red Hat Ceph Storage クラスターがある。
- Ceph ソフトウェアリポジトリーへのアクセス。
- Ceph Monitor ノードへの root レベルのアクセス。

#### 手順

1. 以下の例を使用してルールを追加します。

```
[root@mon ~]# sudo iptables -A INPUT -i IFACE -p tcp -s IP-ADDRESS/NETMASK --dport 6789 -j ACCEPT
[root@mon ~]# sudo iptables -A INPUT -i IFACE -p tcp -s IP-ADDRESS/NETMASK --dport 3300 -j ACCEPT
```

   a. **IFACE** は、パブリックネットワークインターフェイス (例: **eth0**、**eth1** など) に置き換えます。

   b. **IP-ADDRESS** は、パブリックネットワークの IP アドレスに、 **NETMASK** は、パブリックネットワークのネットマスクに置き換えます。

2. **firewalld** デーモンの場合は、以下のコマンドを実行します。

```
[root@mon ~]# firewall-cmd --zone=public --add-port=6789/tcp
[root@mon ~]# firewall-cmd --zone=public --add-port=6789/tcp --permanent
[root@mon ~]# firewall-cmd --zone=public --add-port=3300/tcp
[root@mon ~]# firewall-cmd --zone=public --add-port=3300/tcp --permanent
```

#### 関連情報

- 特定のオプションの説明や使用方法は、**Ceph のネットワーク設定オプション** の **Red Hat Ceph Storage ネットワーク設定オプション** を参照してください。
- Ceph messenger バージョン 2 プロトコルでの **Ceph の伝送時暗号化** の使用に関する詳細は、**Red Hat Ceph Storage アーキテクチャーガイド** を参照してください。

# 第3章 CEPH MONITOR の設定

ストレージ管理者として、Ceph Monitor のデフォルト設定値を使用することも、目的のワークロードに応じてカスタマイズすることもできます。

**前提条件**

- Red Hat Ceph Storage ソフトウェアのインストール

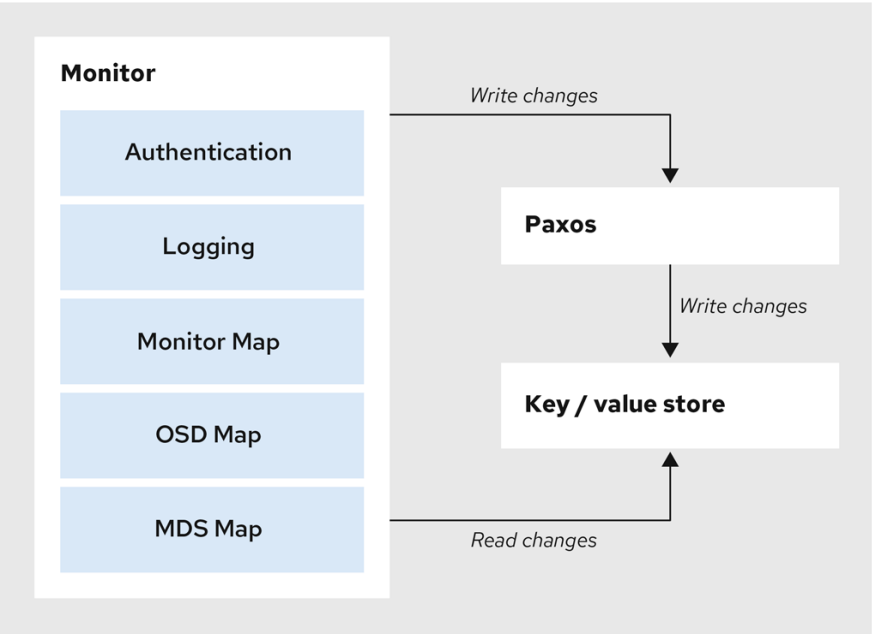
## 3.1. CEPH MONITOR の設定

Ceph Monitor の設定方法を理解することは、信頼性の高い Red Hat Ceph Storage クラスターを構築する上で重要なことです。すべてのストレージクラスターには少なくとも 1 つのモニターがあります。通常、Ceph Monitor の設定はほぼ一定のままですが、ストレージクラスター内の Ceph Monitor を追加、削除、または交換することができます。

Ceph モニターは、クラスターマップのマスターコピーを維持します。つまり、1 つの Ceph モニターに接続して最新のクラスターマップを取得するだけで、Ceph クライアントはすべての Ceph モニターと Ceph OSD の位置を把握することができます。

Ceph クライアントが Ceph OSD に対して読み取り/書き込みを行うには、まず Ceph Monitor に接続する必要があります。クラスターマップの現在のコピーと CRUSH アルゴリズムを使用して、Ceph クライアントは任意のオブジェクトの位置を計算できます。オブジェクトの位置を計算できることで、Ceph クライアントは Ceph OSD と直接対話できるます。このことは、Ceph の高いスケーラビリティーとパフォーマンスを実現する上で非常に重要な要素となります。

Ceph Monitor の主なロールは、クラスターマップのマスターコピーを維持することです。Ceph Monitor は、認証とログサービスも提供します。Ceph Monitor は、モニターサービスのすべての変更を 1 つの Paxos インスタンスに書き込み、Paxos はその変更をキー/値ストアに書き込んで強い一貫性を持たせます。Ceph Monitor は、同期操作中にクラスターマップの最新バージョンにクエリーを行うことができます。Ceph Monitor は、**rocksdb** データベースを使用したキー値ストアのスナップショットやイテレーターを使用して、ストア全体の同期を実行します。




## 3.2. CEPH MONITOR 設定データベースの表示

設定データベースで Ceph Monitor 設定を表示できます。

**注記**

Red Hat Ceph Storage の以前のリリースでは、**/etc/ceph/ceph.conf** で Ceph Monitor 設定を一元管理します。この設定ファイルは、Red Hat Ceph Storage 5 では非推奨になっています。

**前提条件**

- 稼働中の Red Hat Ceph Storage クラスターがある。
- Ceph Monitor ホストへの root レベルのアクセス。

**手順**

1. **Cephadm** シェルにログインします。

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

2. **ceph config** コマンドを使用して、設定データベースを表示します。

   **例**

```
[ceph: root@host01 /]# ceph config get mon
```

**関連情報**

- **ceph config** コマンドで使用できるオプションの詳細については、**ceph config -h** を使用します。

## 3.3. CEPH クラスターマップ

クラスターマップは、モニターマップ、OSD マップ、および配置グループマップなどのマップを合成したものです。クラスターマップは、多くの重要なイベントを追跡します。

- どのプロセスが Red Hat Ceph Storage クラスター内 (**in**) にあるか。
- Red Hat Ceph Storage クラスター内 **in** にあるプロセスが **up** で稼働しているか、**down** であるか。
- 配置グループが **active** または **inactive** で **clean** な、または他の一部の状態にあるかどうか。
- クラスターの現状を反映したその他の詳細情報。これには以下が含まれます。
  - ストレージ容量の合計、または
  - 使用されているストレージ容量の合計

例えば、Ceph OSD がダウンしたり、配置グループがデグレード状態に陥ったりするなど、クラスターの状態に大きな変化があった場合。クラスターマップが更新され、クラスターの現在の状態が反映されます。さらに、Ceph モニターはクラスターの以前の状態の履歴も保持します。モニターマップ、OSD マップ、および配置グループマップは、それぞれのマップバージョンの履歴を保持します。各バージョンは **エポック** と呼ばれます。

Red Hat Ceph Storage クラスターを運用する場合、これらの状態を追跡することはクラスター管理の重要な部分です。

## 3.4. CEPH MONITOR クォーラム

クラスターは、1 台のモニターで十分に動作します。しかし、1 台のモニターは単一故障点になります。本番環境の Ceph ストレージクラスターで高可用性を確保するには、複数のモニターで Ceph を実行し、1 つのモニターの故障がストレージクラスター全体の障害にならないようにします。

Ceph ストレージクラスターが高可用性のために複数の Ceph Monitor を実行している場合、Ceph Monitor は Paxos アルゴリズムを使用してマスタークラスターマップに関する合意を確立します。コンセンサスを得るには、大半のモニターが動作していて、クラスターマップに関するコンセンサスのためのクォーラムを確立する必要があります。例えば、1、3 つのうちの 2 つ、5 つのうちの 3 つ、6 つのうちの 4 つ等。

Red Hat では、高可用性を確保するために、少なくとも 3 つの Ceph Monitor で本番環境の Red Hat Ceph Storage クラスターを実行することを推奨しています。複数のモニターを実行する場合、クォーラムを確立するためにストレージクラスターのメンバーでなければならない初期モニターを指定することができます。これにより、ストレージクラスターがオンラインになるまでの時間が短縮される場合があります。

```
[mon]
mon_initial_members = a,b,c
```

**注記**

クォーラムを確立するには、ストレージクラスター内のモニターの **大半** が相互に到達できる必要があります。**mon_initial_members** オプションでクォーラムを確立するモニターの最初の数を減らすことができます。

## 3.5. CEPH MONITOR の一貫性

Ceph 設定ファイルにモニター設定を追加する場合、Ceph Monitor モニターのアーキテクチャー的な側面をいくつか知っておく必要があります。Ceph は、クラスター内で別の Ceph Monitor を検出する際に、Ceph Monitor に厳格な一貫性要件を課します。Ceph クライアントおよびその他の Ceph デーモンは、Ceph 設定ファイルを使用してモニターを検出しますが、モニターは Ceph 設定ファイルではなくモニターマップ (**monmap**) を使用して相互を検出します。

Ceph Monitor が Red Hat Ceph Storage クラスター内の他の Ceph Monitor を検出する場合、常にモニターマップのローカルコピーを参照します。Ceph 設定ファイルではなくモニターマップを使用することで、クラスターが壊れる可能性のあるエラーを回避できます。例えば、Ceph 設定ファイルでモニターのアドレスやポートを指定する際のタイプミスなどです。モニターは検出のためにモニターマップを使用し、クライアントや他の Ceph デーモンとモニターマップを共有するため、モニターマップは、モニターのコンセンサスが有効であることをモニターに対して厳格に保証します。

### モニターマップへの更新適用時の厳格な一貫性

Ceph Monitor の他の更新と同様に、モニターマップへの変更は常に Paxos と呼ばれる分散型コンセンサスアルゴリズムを介して行われます。Ceph Monitor は、Ceph Monitor の追加や削除など、モニターマップへの各更新について合意し、クォーラムの各モニターが同じバージョンのモニターマップを持つようにする必要があります。モニターマップへの更新はインクリメンタルに行われるため、Ceph Monitor は最新の合意バージョンと以前のバージョンのセットを持つことになります。

### 履歴の維持

履歴を維持することで、古いバージョンのモニターマップを持つ Ceph Monitor が、Red Hat Ceph Storage クラスターの現在の状態に追いつくことができます。

Ceph Monitor がモニターマップではなく Ceph 設定ファイルを介してお互いを検出する場合、Ceph 設定ファイルは自動的に更新および配布されないため、新たなリスクが発生する可能性があります。Ceph Monitor が誤って古い Ceph 設定ファイルを使用し、Ceph Monitor の識別に失敗し、クォーラムから外れたり、Paxos がシステムの現在の状態を正確に判断できなかったりする状況が発生する可能性があります。

## 3.6. CEPH MONITOR のブートストラップ

ほとんどの設定とデプロイメントの場合、**cephadm** などの Ceph をデプロイするツールは、モニターマップを生成して Ceph モニターのブートストラップを支援することがあります。

Ceph モニターには、いくつかの明示的な設定が必要です。

- **ファイルシステム ID**: **fsid** は、オブジェクトストアの一意識別子です。同じハードウェア上で複数のストレージクラスターを稼働させることができるため、モニターのブートストラップを行う場合には、オブジェクトストアの一意の ID を指定する必要があります。**cephadm** などのデプロイメントツールを使用すると、ファイルシステムの識別子が生成されますが、**fsid** も手動で指定できます。

- **モニター ID**: モニター ID は、クラスター内の各モニターに割り当てられる一意の ID です。通常、ID はモニターのホスト名に設定されます。このオプションは、デプロイメントツールを使用して、**ceph** コマンドまたは Ceph 設定ファイルで設定できます。Ceph 設定ファイルでは、セクションは以下のように形成されます。

  **例**

  ```
  [mon.host1]

  [mon.host2]
  ```

- **キー**: モニターには秘密鍵が必要です。

**関連情報**

- **cephadm** と Ceph Orchestrator の詳細は、 Red Hat Ceph Storage オペレーションガイド を参照してください。

## 3.7. CEPH MONITOR の最小設定

Ceph 設定ファイルの Ceph Monitor の最低限のモニター設定には、各モニターのホスト名 (DNS に設定されていない場合) とモニターアドレスが含まれます。Ceph Monitor はデフォルトで port **6789** および **3300** で実行されます。

**重要**

Ceph 設定ファイルは編集しないでください。

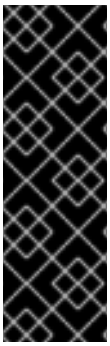
**注記**

このモニターの最小設定は、デプロイメントツールが **fsid** と **mon.** キーを生成することを前提としています。

以下のコマンドを使用して、ストレージクラスターの設定オプションを設定するか、読み取ることができます。

- **Ceph config dump**: ストレージクラスター全体の設定データベース全体をダンプします。
- **ceph config generate-minimal-conf**: 最小限の **ceph.conf** ファイルを生成します。
- **ceph config get WHO**: Ceph Monitor の設定データベースに保管されている特定のデーモンまたはクライアントの設定をダンプします。
- **ceph config set WHO OPTION VALUE**: Ceph Monitor の設定データベースの設定オプションを設定します。
- **ceph config show WHO**: 実行中のデーモンについて、報告された実行中の設定を表示します。
- **ceph config assimilate-conf -i INPUT_FILE -o OUTPUT_FILE**: 入力ファイルから設定ファイルを取得し、有効なオプションをすべて Ceph Monitor の設定データベースに移動します。

ここで、**WHO** パラメーターはセクションまたは Ceph デーモンの名前、 **OPTION** は設定ファイルで、**VALUE** は **true** または **false** のいずれかになります。



**重要**

Ceph デーモンが設定ストアからオプションを取得する前に設定オプションが必要な場合は、以下のコマンドを実行して設定を行うことができます。

```
ceph cephadm set-extra-ceph-conf
```

このコマンドにより、すべてのデーモンの **ceph.conf** ファイルにテキストが追加されます。これは回避策であり、推奨される操作ではありません。

## 3.8. CEPH の一意の識別子

各 Red Hat Ceph Storage クラスターには固有の ID (**fsid**) があります。指定した場合には、通常は設定ファイルの **[global]** セクションに表示されます。デプロイメントツールは通常、**fsid** を生成してモニターマップに保存するため、値は設定ファイルに表示されない可能性があります。**fsid** を使用すると、同じハードウェア上で複数のクラスターに対してデーモンを実行できます。

**注記**

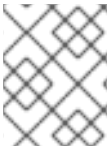
値を設定するデプロイメントツールを使用している場合は、この値を設定しないでください。

## 3.9. CEPH MONITOR のデータストア

Ceph では、Ceph モニターがデータを保存するデフォルトのパスが用意されています。

**重要**

Red Hat では、実稼働 Red Hat Ceph Storage クラスターで最適なパフォーマンスを得るために、Ceph OSD とは別のドライブで Ceph モニターを実行することを推奨します。



**注記**

MON データベースには、50 ～ 100 GB のサイズの専用の **/var/lib/ceph** パーティションを使用する必要があります。

Ceph モニターは **fsync()** 関数を頻繁に呼び出します。これは、Ceph OSD ワークロードに干渉する可能性があります。

Ceph モニターは、データをキー/値ペアとして保存します。データストアを使用すると、他のメリットに加えて、復旧中の Ceph モニターが Paxos と通じて破損したバージョンを実行することを防ぎ、1 つのアトミックバッチで複数の修正操作が可能になります。

**重要**

Red Hat はデフォルトのデータの場所を変更することを推奨しません。デフォルトの場所を変更する場合は、設定ファイルの **[mon]** セクションにそれを設定して、Ceph モニター全体で統一します。

## 3.10. CEPH ストレージの容量

Red Hat Ceph Storage クラスターが最大容量 (**mon_osd_full_ratio** パラメーターにより指定) に近くなると、データの損失を防ぐために安全対策として Ceph OSD への書き込みや読み取りができなくなります。そのため、本番環境の Red Hat Ceph Storage クラスターをそのフル比率に近づけてしまうことは、高可用性が犠牲になってしまうのでグッドプラクティスとは言えません。デフォルトのフル比率は、**.95** (容量の 95%) です。これは、OSD の数が少ないテストクラスター用の非常に厳しい設定です。

**ヒント**

クラスターをモニタリングする際に、**nearfull** な比率に関連する警告にアラートしてください。つまり、1 つまたは複数の OSD が故障した場合、一部の OSD の障害により一時的にサービスが中断される可能性があります。ストレージの容量を増やすために、OSD の増設を検討してください。

テストクラスターの一般的なシナリオでは、システム管理者が Red Hat Ceph Storage クラスターから Ceph OSD を削除してクラスターの再バランスを観察します。その後、別の Ceph OSD を削除し、Red Hat Ceph Storage クラスターが最終的にフル比率に達してロックアップするまでこれを繰り返します。

**重要**

Red Hat では、テストクラスターであっても、多少の容量計画を立てることを推奨しています。計画を立てることで、高可用性を維持するためにどれだけの予備容量が必要なのかを把握することができます。

理想的には、Ceph OSD を直ちに置き換えることなく、クラスターが **active + clean** な状態に復元できる Ceph OSD の一連の障害を計画する必要があります。クラスターを **active + degraded** の状態で実行できますが、これは通常の動作条件には理想的ではありません。

次の図は、33 台の Ceph Node が含まれる単純化した Red Hat Ceph Storage クラスターを示しています。ホストごとに 1 つの Ceph OSD があり、各 Ceph OSD デーモンは 3 TB のドライブに対して読み取りおよび書き込を行います。つまり、この例の Red Hat Ceph Storage クラスターの最大実容量は 99 TB です。**mon osd full ratio** が **0.95** の場合は、Red Hat Ceph Storage クラスターが空き容量が 5 TB になると、Ceph クライアントはデータの読み取りと書き込みを許可しません。そのため、Red Hat Ceph Storage クラスターの運用上の容量は 99 TB ではなく 95 TB となります。


このようなクラスターでは、1 つまたは 2 つの OSD が故障するのが普通です。頻度は低いが妥当なシナリオとしては、ラックのルーターや電源が故障し、複数の OSD が同時にダウンすることが挙げられます (例: OSD 7 - 12)。このようなシナリオでは、さらに OSD のあるホストを短い順序で追加する場合でも、動作し続け、**active + clean** な状態を実現するクラスターを試す必要があります。容量利用率が高すぎると、データを失うことはないかもしれませんが、クラスターの容量利用率がフル比率を超えた場合、障害ドメイン内の障害を解決している間データの可用性が犠牲になる可能性があります。このため、Red Hat では、少なくとも大まかな容量計画を立てることを推奨しています。

クラスターに関する 2 つの数字を把握します。

- OSD の数
- クラスターの総容量

クラスター内の OSD の平均容量を求めるには、クラスターの総容量をクラスター内の OSD の数で割ります。この数に、通常の運用で同時に故障すると予想される OSD の数 (比較的小さい数) を乗じます。最後に、クラスターの容量にフル比率を掛けて、運用上の最大容量を算出します。そして、失敗すると予想される OSD からデータ量を差し引いて、合理的なフル比率を算出します。前述のプロセスを、より多くの OSD 故障数 (例えば、OSD のラック) で繰り返し、ほぼフル比率のための妥当な数を算出します。

## 3.11. CEPH ハートビート

Ceph モニターは、各 OSD からのレポートを要求し、隣接する OSD の状態に関するレポートを OSD から受け取ることで、クラスターについて把握します。Ceph では、モニターと OSD の間の相互作用について妥当なデフォルト設定が用意されていますが、必要に応じて変更することができます。

## 3.12. CEPH MONITOR の同期ロール

複数のモニターを持つ本番環境用のクラスターを運用する場合 (推奨される設定)、各モニターは隣接するモニターがより新しいバージョンのクラスターマップを持っているかどうかを確認します。例えば、隣接するモニターのマップのエポックナンバーが、インスタントモニターのマップの最新のエポックよ

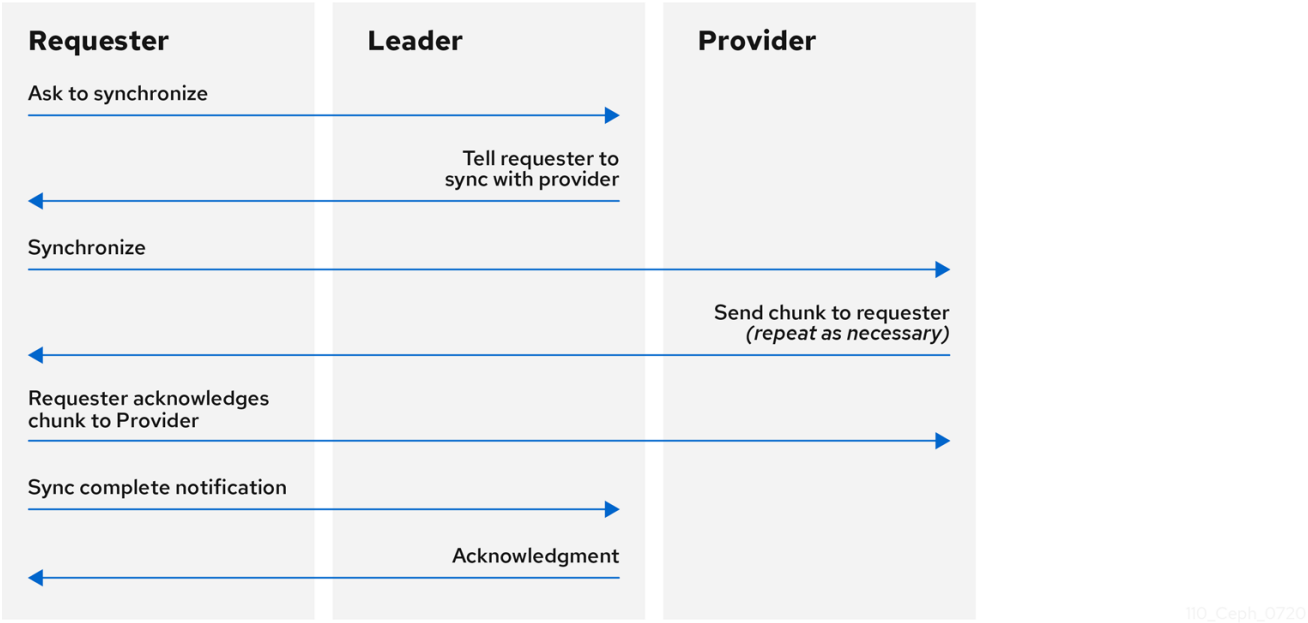
り 1 つ以上高い場合。定期的に、クラスター内のあるモニターが他のモニターから遅れをとることがあります。その場合、そのモニターはクォーラムから離脱し、同期をとってクラスターに関する最新の情報を取得した後、再びクォーラムに参加しなければなりません。

### 同期ロール

同期のために、モニターは以下の 3 つのロールのいずれかを取ります。

- **リーダー**: リーダーは、クラスターマップの最新の Paxos バージョンを実現する最初のモニターです。
- **プロバイダー**: プロバイダーは最新バージョンのクラスターマップを持つモニターですが、最新バージョンを最初に達成したわけではありません。
- **リクエスター**: リクエスターはリーダーに遅れをとっているモニターで、クォーラムに再参加する前にクラスターに関する最新情報を取得するために同期する必要があります。

これらのロールにより、リーダーは同期のタスクをプロバイダーに委譲しすることができ、同期の要求によりリーダーが過負荷になることを防ぎ、パフォーマンスが向上します。次の図では、リクエスターが他のモニターに遅れをとっていることを認識しています。リクエスターはリーダーに同期を依頼し、リーダーはリクエスターにプロバイダーとの同期を指示します。




### モニターの同期

新しいモニターがクラスターに参加すると、常に同期が行われます。実行時の運用において、モニターは異なるタイミングでクラスターマップへの更新を受け取る場合があります。つまり、リーダーとプロバイダーのロールが、モニター間で移動する可能性があるということです。例えば、同期中にこれが起こると、プロバイダーはリーダーから遅れてしまい、プロバイダーはリクエスターとの同期を終了することができます。

同期が完了すると、Ceph ではクラスター全体のトリミングが必要になります。トリミングを行うには、配置グループが **active + clean** である必要があります。

## 3.13. CEPH の時刻同期

Ceph デーモンは、クリティカルなメッセージを相互に渡します。このメッセージは、デーモンがタイムアウトのしきい値に達する前に処理する必要があります。Ceph モニターのクロックが同期していないと、さまざまな異常が発生する可能性があります。

以下に例を示します。

- デーモンが受信したメッセージを無視する (タイムスタンプが古いなど)。
- メッセージ受信のタイミングが適切でない場合、タイムアウトの発生が早すぎたり遅すぎたりする。

### ヒント

Ceph モニターホストに NTP をインストールして、モニタークラスターのクロックが同期した状態で動作するようにします。

NTP では、遅れによる悪影響が出ていなくても、クロックドリフトが目立つことがあります。NTP が適切なレベルの同期を維持していても、Ceph のクロックドリフトとクロックスキューの警告が発生することがあります。このような状況では、クロックドリフトを増やすことが許容できるかもしれません。しかし、ワークロード、ネットワークレイテンシー、デフォルトのタイムアウトに対するオーバーライド設定、およびその他の同期オプションなど、多くの要因が、Paxos の保証を損なうことなく許容できるクロックドリフトのレベルに影響を与えます。

### 関連情報

- 詳細は、**Ceph の時刻同期** セクションを参照してください。
- 特定のオプションの説明や使用方法は、**Ceph Monitor の設定オプション** のすべての Red Hat Ceph Storage Monitor 設定オプションを参照してください。

# 第4章 CEPH の認証設定

ストレージ管理者として、ユーザーとサービスを認証することは、Red Hat Ceph Storage クラスターのセキュリティーにとって重要です。Red Hat Ceph Storage には、デフォルトで暗号認証用の Cephx プロトコルと、ストレージクラスターで認証を管理するツールが含まれています。

Red Hat Ceph Storage には、デフォルトで暗号認証用の Cephx プロトコルと、ストレージクラスターで認証を管理するツールが含まれています。

Ceph 認証設定の一部として、セキュリティーを強化するために Ceph およびゲートウェイデーモンのキーのローテーションを検討してください。キーのローテーションは、コマンドラインの **cephadm** を介して行われます。詳細は、**キーローテーションの有効化** を参照してください。

**前提条件**

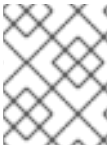
- Red Hat Ceph Storage ソフトウェアのインストール

## 4.1. CEPHX 認証

**cephx** プロトコルはデフォルトで有効になっています。暗号認証には多少の計算コストがかかりますが、一般的には非常に低いものです。クライアントとホストを結ぶネットワーク環境が安全と考えられ、認証の計算コストがかけられない場合は、無効にすることができます。Ceph Storage クラスターをデプロイする際に、デプロイメントツールは **client.admin** ユーザーおよびキーリングを作成します。

**重要**

Red Hat では認証の使用を推奨しています。



**注記**

認証を無効にすると、中間者攻撃によってクライアントとサーバーのメッセージが改ざんされる危険性があり、重大なセキュリティー問題に発展する可能性があります。

### Cephx の有効化と無効化

Cephx を有効にするには、Ceph Monitor と OSD 用のキーをデプロイする必要があります。Cephx 認証のオン/オフを切り替える場合は、デプロイメント手順を繰り返す必要はありません。

## 4.2. CEPHX の有効化

**cephx** が有効な場合には、Ceph はデフォルトの検索パス **/etc/ceph/$cluster.$name.keyring** を含む) でキーリングを探します。Ceph 設定ファイルの **[global]** セクションに **keyring** オプションを追加することで、この場所を上書きすることができますが、これは推奨されません。

認証が無効になっているクラスターで **cephx** を有効にするには、以下の手順を実行します。ご自身またはデプロイメントユーティリティーがすでにキーを生成している場合は、キーの生成に関する手順を省略できます。

**前提条件**

- 稼働中の Red Hat Ceph Storage クラスターがある。
- Ceph Monitor ノードへの root レベルのアクセス。

**手順**

1. **client.admin** キーを作成し、クライアントホストのキーのコピーを保存します。

```
[root@mon ~]# ceph auth get-or-create client.admin mon 'allow *' osd 'allow *' -o
/etc/ceph/ceph.client.admin.keyring
```

**警告**

これにより、既存の **/etc/ceph/client.admin.keyring** ファイルの内容が消去されます。すでにデプロイメントツールがこの作業を行っている場合は、この手順を実行しないでください。

2. モニタークラスター用のキーリングを作成し、モニターシークレットキーを生成します。

```
[root@mon ~]# ceph-authtool --create-keyring /tmp/ceph.mon.keyring --gen-key -n mon. --
cap mon 'allow *'
```

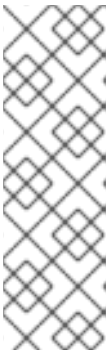
3. すべてのモニターの **mon data** ディレクトリーの **ceph.mon.keyring** ファイルにモニターキーリングをコピーします。たとえば、これをクラスター **ceph** の **mon.a** にコピーするには、以下のコマンドを使用します。

```
[root@mon ~]# cp /tmp/ceph.mon.keyring /var/lib/ceph/mon/ceph-a/keyring
```

4. すべての OSD に秘密鍵を生成します。ここで、**ID** は OSD 番号です。

```
ceph auth get-or-create osd.ID mon 'allow rwx' osd 'allow *' -o
/var/lib/ceph/osd/ceph-ID/keyring
```

5. デフォルトでは、**cephx** 認証プロトコルは有効になっています。

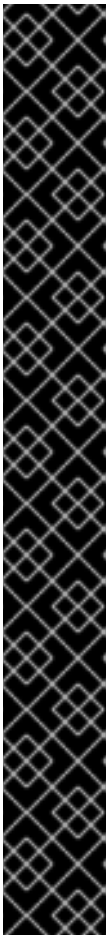


**注記**

認証オプションを **none** に設定して **cephx** 認証プロトコルが無効にされていた場合には、Ceph 設定ファイル (**/etc/ceph/ceph.conf**) の **[global]** セクションの下にある以下の行を削除して、**cephx** 認証プロトコルを再度有効にします。

```
auth_cluster_required = none
auth_service_required = none
auth_client_required = none
```

6. Ceph Storage クラスターを起動または再起動します。



**重要**

**cephx** を有効にするには、クラスターを完全に再起動する必要があるか、クライアントの I/O が無効になったときにシャットダウンしてから起動する必要があるため、ダウンタイムが必要です。

これらのフラグは、ストレージクラスターを再起動またはシャットダウンする前に設定する必要があります。

```
[root@mon ~]# ceph osd set noout
[root@mon ~]# ceph osd set norecover
[root@mon ~]# ceph osd set norebalance
[root@mon ~]# ceph osd set nobackfill
[root@mon ~]# ceph osd set nodown
[root@mon ~]# ceph osd set pause
```

**cephx** が有効になり、すべての PG がアクティブかつクリーンな状態になったら、フラグの設定を解除します。

```
[root@mon ~]# ceph osd unset noout
[root@mon ~]# ceph osd unset norecover
[root@mon ~]# ceph osd unset norebalance
[root@mon ~]# ceph osd unset nobackfill
[root@mon ~]# ceph osd unset nodown
[root@mon ~]# ceph osd unset pause
```

## 4.3. CEPHX の無効化

以下の手順では、Cephx を無効にする方法を説明します。クラスター環境が比較的安全であれば、認証を実行するための計算コストを相殺することができます。

**重要**

Red Hat では認証を有効にすることを推奨しています。

しかし、セットアップやトラブルシューティングの際には、一時的に認証を無効にした方が簡単な場合もあります。

**前提条件**

- 稼働中の Red Hat Ceph Storage クラスターがある。
- Ceph Monitor ノードへの root レベルのアクセス。

**手順**

1. Ceph 設定ファイルの **[global]** セクションに以下のオプションを設定して、**cephx** 認証を無効にします。

   **例**

```
auth_cluster_required = none
auth_service_required = none
auth_client_required = none
```

2. Ceph Storage クラスターを起動または再起動します。

## 4.4. CEPHX ユーザーキーリング

認証が有効な Ceph を実行する場合には、Ceph Storage クラスターにアクセスするために **ceph** 管理コマンドおよび Ceph クライアントに認証キーが必要です。

**ceph** 管理コマンドおよびクライアントにこれらの鍵を提供する最も一般的な方法は、**/etc/ceph/** ディレクトリーの下に Ceph キーリングを追加することです。ファイル名は通常 **ceph.client.admin.keyring** または **$cluster.client.admin.keyring** です。**/etc/ceph/** ディレクトリーにキーリングを含める場合は、Ceph 設定ファイルで **keyring** エントリーを指定する必要はありません。

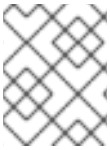
**重要**

Red Hat は、**client.admin** キーが含まれるため、Red Hat Ceph Storage クラスターのキーリングファイルを管理コマンドを実行するノードにコピーすることを推奨します。

それを行うには、以下のコマンドを実行します。

```
# scp USER@HOSTNAME:/etc/ceph/ceph.client.admin.keyring /etc/ceph/ceph.client.admin.keyring
```

**USER** を、ホストで使用されるユーザー名に **client.admin** キーを使用し、**HOSTNAME** をそのホストのホスト名に置き換えます。



**注記**

**ceph.keyring** ファイルに、クライアントマシンに適切なパーミッションが設定されていることを確認します。

推奨されていない **key** 設定を使用して、Ceph 設定ファイルにキー自体を指定したり、**keyfile** 設定を使用してキーファイルへのパスを指定することができます。

## 4.5. CEPHX デーモンのキーリング

管理ユーザーやデプロイメントツールは、ユーザーキーリングの生成と同じ方法で、デーモンキーリングを生成することがあります。デフォルトでは、Ceph はデーモンのキーリングをデータディレクトリー内に保存します。デフォルトのキーリングの場所や、デーモンが機能するために必要な機能など。

**注記**

モニターキーリングにはキーが含まれていますが、機能はなく、Ceph Storage クラスターの **auth** データベースの一部ではありません。

デーモンデータのディレクトリーの位置は、デフォルトでは以下の形式のディレクトリーになります。

```
/var/lib/ceph/$type/CLUSTER-ID
```

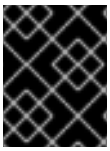
**例**

```
/var/lib/ceph/osd/ceph-12
```

これらの場所を上書きすることもできますが、推奨できません。

## 4.6. CEPHX イメージの署名

Ceph にはきめ細かな制御機能があり、クライアントと Ceph の間のサービスメッセージの署名を有効または無効にすることができます。Ceph デーモン間のメッセージに対する署名を有効または無効にすることができます。



**重要**

Red Hat では、最初の認証のために設定されたセッションキーを使用して、Ceph がエンティティー間のすべての進行中のメッセージを認証することを推奨しています。

**注記**

Ceph のカーネルモジュールは、まだ署名をサポートしていません。

# 第5章 プール、配置グループ、および CRUSH の設定

ストレージ管理者として、プール、配置グループ、および CRUSH アルゴリズムに Red Hat Ceph Storage のデフォルトオプションを使用するか、目的のワークロードに合わせてカスタマイズするかを選択することができます。

**前提条件**

- Red Hat Ceph Storage ソフトウェアのインストール

## 5.1. プール、配置グループ、および CRUSH

プールを作成し、プールの配置グループの数を設定するとき、特にデフォルトをオーバーライドしない場合、Ceph はデフォルト値を使用します。

**重要**

Red Hat では、いくつかのデフォルトを上書きすることを推奨します。具体的には、プールのレプリカサイズを設定し、デフォルトの配置グループ数を上書きします。

これらの値は、pool コマンドの実行時に設定できます。

デフォルトでは、Ceph はオブジェクトの 3 つのレプリカを作成します。オブジェクトの 4 つのコピーをデフォルト値、プライマリーコピー、および 3 つのレプリカコピーとして設定する場合は、**osd_pool_default_size** のようにデフォルト値をリセットします。Ceph が劣化状態でより少ない数のコピーを書き込めるようにする場合は、**osd_pool_default_min_size** を **osd_pool_default_size** 値よりも小さい数に設定します。

**例**

```
[ceph: root@host01 /]# ceph config set global osd_pool_default_size 4  # Write an object 4 times.
[ceph: root@host01 /]# ceph config set global osd_pool_default_min_size 1  # Allow writing one copy in a degraded state.
```

配置グループの数が正しいことを確認してください。Red Hat は、OSD ごとに約 100 を推奨します。たとえば、OSD の合計数を 100 で乗算して、レプリカ数で除算します (**osd pool default size**)。10 OSD および **osd_pool_default_size** = 4 の場合は、おおよそ(100 * 10)/ 4 = 250 を推奨します。

**例**

```
[ceph: root@host01 /]# ceph config set global osd_pool_default_pg_num 250
[ceph: root@host01 /]# ceph config set global osd_pool_default_pgp_num 250
```

**関連情報**

- 特定のオプションの説明や使用方法は、**付録 E** の Red Hat Ceph Storage プール、配置グループ、および CRUSH 設定オプションをすべて参照してください。

# 第6章 CEPH OBJECT STORAGE DAEMON (OSD) の設定

ストレージ管理者として、Ceph Object Storage Daemon (OSD) を設定して、意図するワークロードに基づいて冗長化と最適化を行うことができます。

**前提条件**

- Red Hat Ceph Storage ソフトウェアのインストール

## 6.1. CEPH OSD の設定

すべての Ceph クラスターには、以下の項目を定義する設定があります。

- クラスター ID
- 認証設定
- クラスター内の Ceph daemon のメンバーシップ
- ネットワーク設定
- ホスト名およびアドレス
- キーリングへのパス
- OSD ログファイルへのパス
- 他のランタイムオプション

**cephadm** などのデプロイメントツールは、通常、初期の Ceph 設定ファイルを作成します。ただし、デプロイメントツールを使用してクラスターをブートストラップする場合には、独自に作成することができます。

便利なように、各デーモンには一連のデフォルト値が用意されています。多くは **ceph/src/common/config_opts.h** スクリプトで設定されます。これらの設定は、Ceph 設定ファイル、またはランタイム時にモニターの **tell** コマンドを使用するか、Ceph ノード上のデーモンソケットに直接接続して上書きできます。

**重要**

Red Hat では、Ceph を後でトラブルシューティングするのが難しくなるため、デフォルトのパスを変更することは推奨していません。

**関連情報**

- **cephadm** と Ceph Orchestrator の詳細は、 Red Hat Ceph Storage オペレーションガイド を参照してください。

## 6.2. OSD のスクラブ

Ceph は、オブジェクトの複数のコピーを作成するだけでなく、配置グループをスクラビングすることでデータの整合性を確保します。Ceph のスクラブは、オブジェクトストレージ層の **fsck** コマンドに似ています。

各配置グループについて、Ceph はすべてのオブジェクトのカタログを生成し、各プライマリーオブジェクトとそのレプリカを比較して、オブジェクトの欠落や不一致がないことを確認します。

ライトスクラビング (毎日) では、オブジェクトのサイズや属性をチェックします。ディープスクラビング (毎週) は、データを読み込んでチェックサムでデータの整合性を確保します。

スクラビングはデータの整合性を保つために重要ですが、パフォーマンスを低下させる可能性があります。以下の設定を調整して、スクラブ動作を増減させます。

**関連情報**

- 詳細については、**Red Hat Ceph Storage 設定ガイド** の付録にある Ceph スクラビングオプション を参照してください。

## 6.3. OSD のバックフィル

Ceph OSD をクラスターに追加したり、クラスターから削除したりすると、CRUSH アルゴリズムは、配置グループを Ceph OSD に移動させたり、Ceph OSD から移動させたりしてバランスを回復させ、クラスターのバランスを取り戻します。配置グループとそれに含まれるオブジェクトを移行するプロセスは、クラスターの運用パフォーマンスを大幅に低下させます。運用パフォーマンスを維持するために、Ceph はこの移行をバックフィルプロセスで実行します。これにより、Ceph はバックフィル操作をデータの読み取りまたは書き込みの要求よりも低い優先度に設定できます。

## 6.4. OSD リカバリー

クラスターが起動したとき、または Ceph OSD が予期せず終了して再起動したとき、OSD は書き込み操作を行う前に他の Ceph OSD とのピアリングを開始します。

Ceph OSD がクラッシュしてオンラインに戻ると、通常、配置グループのオブジェクトのより新しいバージョンが含まれる他の Ceph OSD との同期が取れなくなります。このような場合、Ceph OSD はリカバリーモードに入り、データの最新コピーを取得してマップを最新の状態に戻そうとします。Ceph OSD が停止していた時間によっては、OSD のオブジェクトや配置グループが大幅に古くなっている可能性があります。また、障害ドメイン (例: ラックなど) ダウンした場合、複数の Ceph OSD が同時にオンラインに戻る可能性があります。そのため、復旧作業には時間とリソースが必要になります。

運用パフォーマンスを維持するために、Ceph はリカバリー要求数、スレッド数、およびオブジェクトチャンクサイズを制限してリカバリーを実行し、これにより Ceph は劣化した状態でも適切なパフォーマンスを発揮することができます。

**関連情報**

- 特定のオプションの説明や使用方法は、Object Storage Daemon (OSD) の設定オプションのすべての Red Hat Ceph Storage Ceph OSD 設定オプションを参照してください。

# 第7章 CEPH MONITOR と OSD の連動設定

ストレージ管理者としては、安定した動作環境を確保するために、Ceph Monitor と OSD の相互作用を適切に設定する必要があります。

**前提条件**

- Red Hat Ceph Storage ソフトウェアのインストール

## 7.1. CEPH MONITOR と OSD の連動

Ceph の初期設定が完了したら、Ceph をデプロイして実行することができます。**ceph health**、**ceph -s** などのコマンドを実行すると、Ceph Monitor は Ceph Storage クラスターの現在の状態を報告します。Ceph Monitor は、各 Ceph OSD デーモンからのレポートを要求し、隣接する Ceph OSD デーモンの状態に関するレポートを Ceph OSD デーモンから受け取ることで、Ceph ストレージクラスターについて把握します。Ceph Monitor がレポートを受信しない場合、または Ceph ストレージクラスターの変更のレポートを受信した場合、Ceph Monitor は Ceph クラスターマップのステータスを更新します。

Ceph では、Ceph Monitor と OSD の連携について妥当なデフォルト設定が用意されています。ただし、デフォルト値を上書きできます。以下のセクションでは、Ceph ストレージクラスターを監視する目的で、Ceph Monitor と Ceph OSD デーモンがどのように相互作用するかを説明します。

## 7.2. OSD ハートビート

各 Ceph OSD デーモンは、6 秒ごとに他の Ceph OSD デーモンのハートビートをチェックします。ハートビートの間隔を変更するには、ランタイム時に値を変更します。

**構文**

```
ceph config set osd osd_heartbeat_interval TIME_IN_SECONDS
```

**例**

```
[ceph: root@host01 /]# ceph config set osd osd_heartbeat_interval 60
```

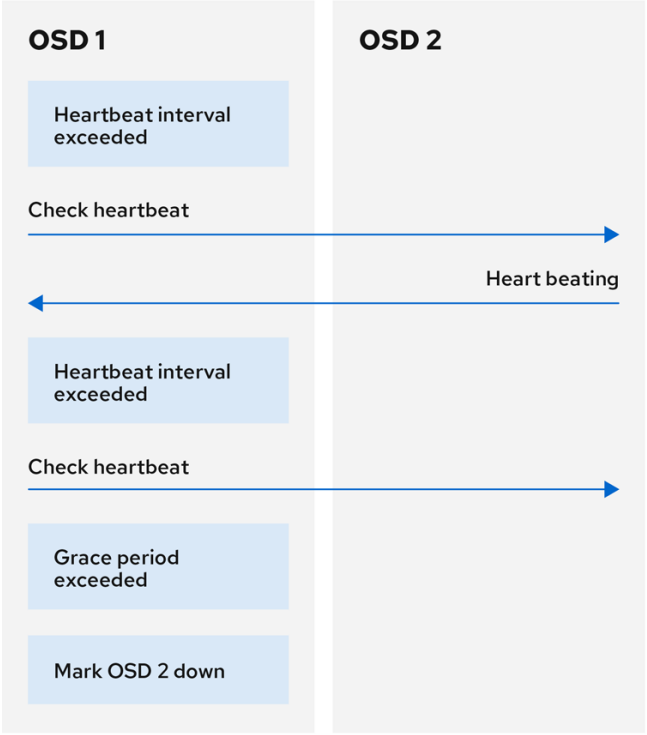
近傍の Ceph OSD デーモンが 20 秒の猶予期間内にハートビートパケットを送信しない場合、Ceph OSD デーモンは近傍の Ceph OSD デーモンが **down** であるとみなされる可能性があります。それを Ceph Monitor に報告して、Ceph クラスターマップを更新することができます。猶予期間を変更するには、ランタイム時に値を設定します。

**構文**

```
ceph config set osd osd_heartbeat_grace TIME_IN_SECONDS
```

**例**

```
[ceph: root@host01 /]# ceph config set osd osd_heartbeat_grace 30
```




## 7.3. OSD がダウンであることの報告

デフォルトでは、**異なるホスト** の 2 つの Ceph OSD デーモンは、報告された Ceph OSD デーモンが **down** していることを Ceph モニターが確認する前に、別の CephOSD デーモンが **down** していることを Ceph モニターに報告する必要があります。

しかし、障害を報告するすべての OSD が、ラック内の異なるホストに設置されており、スイッチ不良により OSD 間の接続に問題が生じる場合があります。

誤報を避けるために、Ceph は障害を報告したピアを、同様に遅延しているサブクラスターの代理として考えます。これは必ずしもそうとは限りませんが、管理者が、パフォーマンスの低下しているシステムのサブセットに局所的に適切な補正を適用するのに役立つ場合があります。

Ceph は **mon_osd_reporter_subtree_level** 設定を使用して、CRUSH マップの共通の先復元タイプでピアを subcluster にグループ化します。

デフォルトでは、**異なるサブツリー**からわずか 2 つのレポートは、他の Ceph OSD デーモン **down** を報告する必要があります。管理者は、**mon_osd_min_down_reporters** および **mon_osd_reporter_subtree_level** の値をランタイム時に設定することで、Ceph Monitor に Ceph OSD デーモンの **down** を報告するために必要な固有のサブツリーと共通の祖先型からレポーターの数を変更することができます。

**構文**

```
ceph config set mon mon_osd_min_down_reporters NUMBER
```

**例**

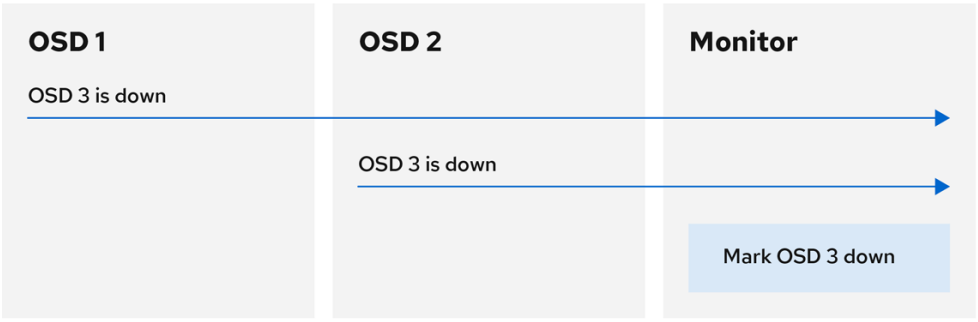
```
[ceph: root@host01 /]# ceph config set mon mon_osd_min_down_reporters 4
```

**構文**

```
ceph config set mon mon_osd_reporter_subtree_level CRUSH_ITEM
```

**例**

```
[ceph: root@host01 /]# ceph config set mon mon_osd_reporter_subtree_level host
[ceph: root@host01 /]# ceph config set mon mon_osd_reporter_subtree_level rack
[ceph: root@host01 /]# ceph config set mon mon_osd_reporter_subtree_level osd
```




## 7.4. ピアリングの失敗の報告

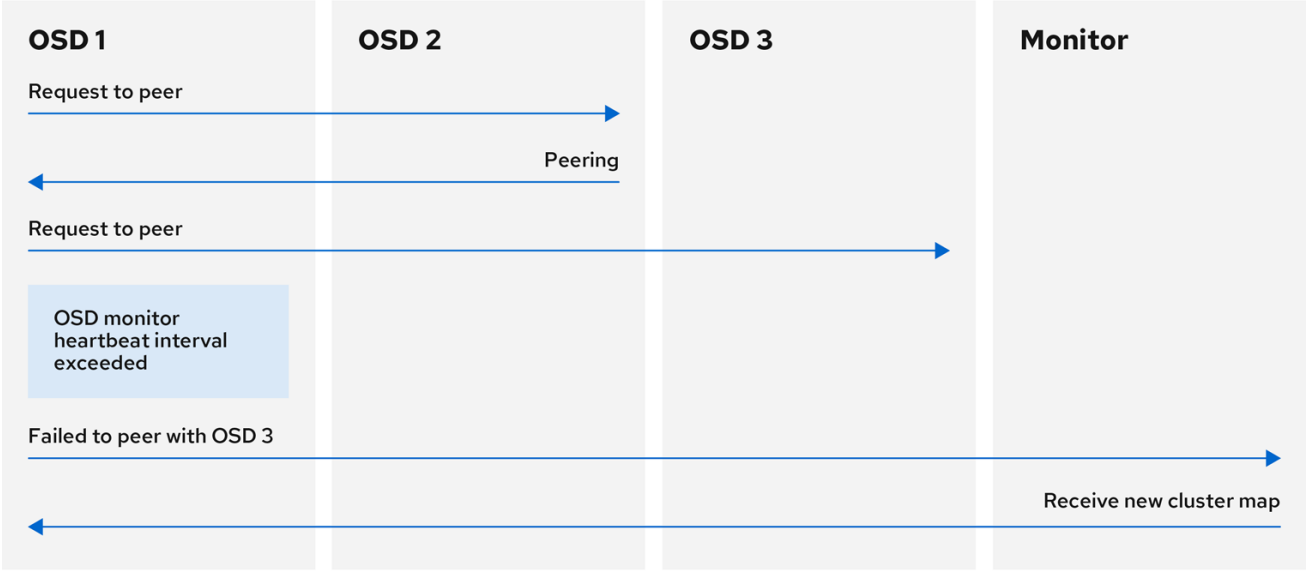
Ceph OSD デーモンが、その Ceph 設定ファイルまたはクラスターマップで定義された Ceph OSD デーモンのいずれともピアリングできない場合、30 秒ごとにクラスターマップの最新コピーを求めて Ceph Monitor に ping を実行します。Ceph Monitor ハートビートの間隔は、ランタイム時に値を設定することで変更できます。

**構文**

```
ceph config set osd osd_mon_heartbeat_interval TIME_IN_SECONDS
```

**例**

```
[ceph: root@host01 /]# ceph config set osd osd_mon_heartbeat_interval 60
```




## 7.5. OSD の報告状況

Ceph OSD デーモンが Ceph Monitor に報告しない場合、Ceph Monitor は **mon_osd_report_timeout**(900 秒) の経過後に Ceph OSD Daemon を **down** とマークします。Ceph OSD デーモンは、障害、配置グループ統計の変更、**up_thru** の変更、または 5 秒以内にブートするなどの報告可能なイベント時に、Ceph Monitor にレポートを送信します。

ランタイム時に **osd_mon_report_interval** の値を設定することで、Ceph OSD デーモンの最小レポート間隔を変更することができます。

**構文**

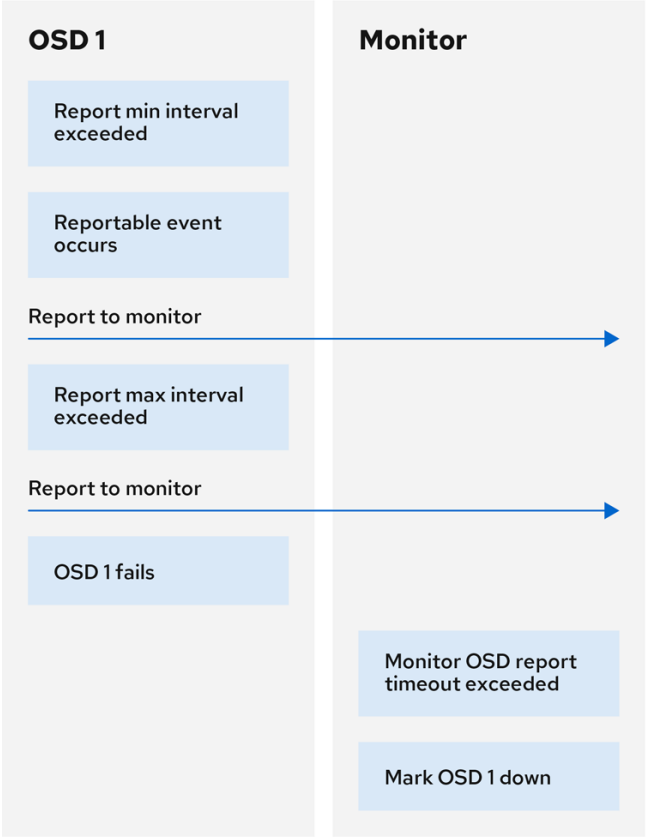
```
ceph config set osd osd_mon_report_interval TIME_IN_SECONDS
```

設定を取得、設定、および検証するには、以下の例を使用できます。

**例**

```
[ceph: root@host01 /]# ceph config get osd osd_mon_report_interval
5
[ceph: root@host01 /]# ceph config set osd osd_mon_report_interval 20
[ceph: root@host01 /]# ceph config dump | grep osd

global                advanced  osd_pool_default_crush_rule              -1
  osd                 basic     osd_memory_target                        4294967296
  osd                 advanced  osd_mon_report_interval                  20
```




**関連情報**

- 特定のオプションの説明や使用方法は、Ceph Monitor および OSD 設定オプションのすべての Red Hat Ceph Storage Ceph Monitor および OSD 設定オプションを参照してください。

# 第8章 CEPH のデバッグとロギングの設定

ストレージ管理者として、**cephadm** のデバッグとログ情報の量を増やして、Red Hat Ceph Storage の問題を診断するのに役立てることができます。

**前提条件**

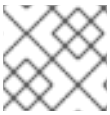
- Red Hat Ceph Storage ソフトウェアがインストールされている。

**関連情報**

- 特定のオプションの説明や使用方法は、**Ceph のデバッグとロギングの設定オプション** に記載されているすべての Red Hat Ceph Storage Ceph デバッグおよびロギング設定オプションを参照してください。
- **cephadm** のトラブルシューティングの詳細は、**Red Hat Ceph Storage 管理ガイド** の **Cephadm のトラブルシューティング** を参照してください。
- **cephadm** のロギングの詳細は、**Red Hat Ceph Storage 管理ガイド** の **Cephadm の操作** を参照してください。

# 付録A 一般的な設定オプション

Ceph の一般的な設定オプションを以下に示します。



**注記**

通常は、**cephadm** などのデプロイメントツールによって自動的に設定されます。

**fsid**

**説明**

ファイルシステム ID です。クラスターごとに 1 つになります。

**型**

UUID

**必須**

いいえ

**デフォルト**

該当なし。通常、デプロイメントツールによって生成されます。

**admin_socket**

**説明**

Ceph モニターがクォーラムを確立しているかどうかにかかわらず、デーモンの管理コマンドを実行するためのソケット

**型**

String

**必須**

いいえ

**デフォルト**

**/var/run/ceph/$cluster-$name.asok**

**pid_file**

**説明**

モニターや OSD が自分の PID を書き込むためのファイル。たとえば、**/var/run/$cluster/$type.$id.pid** は、**ceph** クラスターで実行している id **a** を持つ **mon** の /var/run/ceph/mon.a.pid を作成します。**pid file** は、デーモンが正常に停止すると削除されます。プロセスがデーモン化されていない場合 (つまり、**-f** オプションまたは **-d** オプションで実行)、**pid file** は作成されません。

**型**

String

**必須**

いいえ

**デフォルト**

いいえ

**chdir**

**説明**

Ceph デーモンが起動してから変更するディレクトリー。デフォルトの / ディレクトリーが推奨されます。

**型**

String

**必須**

いいえ

**デフォルト**

/

**max_open_files**

**説明**

これが設定されている場合には、Red Hat Ceph Storage クラスターが起動すると Ceph は OS レベルで **max_open_fds** を設定します (つまりファイル記述子の最大数 #)。これにより、Ceph OSD がファイル記述子を使い果たすのを防ぐことができます。

**型**

64 ビット整数

**必須**

いいえ

**デフォルト**

**0**

**fatal_signal_handlers**

**説明**

設定されていると、SEGV、ABRT、BUS、ILL、FPE、XCPU、XFSZ、SYS シグナルのシグナルハンドラーをインストールして、有用なログメッセージを生成します。

**型**

Boolean

**デフォルト**

**true**

# 付録B CEPH のネットワーク設定オプション

Ceph の共通的なネットワーク設定オプションを以下に示します。

**public_network**

**説明**

パブリック (フロントエンド) ネットワークの IP アドレスとネットマスク (例: **192.168.0.0/24**)。**[global]** に設定します。コンマ区切りのサブネットを指定できます。

**型**

**<ip-address>/<netmask> [, <ip-address>/<netmask>]**

**必須**

いいえ

**デフォルト**

該当なし

**public_addr**

**説明**

パブリック (フロントサイド) ネットワークの IP アドレスです。各デーモンのセット。

**型**

IP アドレス

**必須**

いいえ

**デフォルト**

該当なし

**cluster_network**

**説明**

クラスターネットワークの IP アドレスとネットマスク (例: **10.0.0.0/24**)。**[global]** に設定します。コンマ区切りのサブネットを指定できます。

**型**

**<ip-address>/<netmask> [, <ip-address>/<netmask>]**

**必須**

いいえ

**デフォルト**

該当なし

**cluster_addr**

**説明**

クラスターネットワークの IP アドレスです。各デーモンのセット。

**型**

アドレス

**必須**

いいえ

**デフォルト**

該当なし

**ms_type**

**説明**

ネットワークトランスポート層のメッセンジャータイプです。Red Hat は、**posix** セマンティクスを使用した、messenger タイプ **simple** および **async** をサポートします。

**型**

String.

**必須**

いいえ

**デフォルト**

**async+posix**

**ms_public_type**

**説明**

パブリックネットワークのネットワークトランスポート層のメッセンジャータイプです。これは **ms_type** と同じように動作しますが、パブリックネットワークまたはフロントエンドネットワークにのみ適用されます。この設定により、Ceph はパブリックまたはフロントエンドまたはバックサイドのネットワークに異なるメッセンジャータイプを使用できます。

**型**

String.

**必須**

いいえ

**デフォルト**

なし。

**ms_cluster_type**

**説明**

クラスターネットワークのネットワークトランスポート層のメッセンジャータイプです。これは **ms_type** と同じように動作しますが、クラスターまたはバックサイドネットワークにのみ適用されます。この設定により、Ceph はパブリックまたはフロントエンドまたはバックサイドのネットワークに異なるメッセンジャータイプを使用できます。

**型**

String.

**必須**

いいえ

**デフォルト**

なし。

### ホストオプション

宣言された各モニターの下に **mon addr** 設定を指定して、Ceph 設定ファイル内で少なくとも 1 つの Ceph Monitor を宣言する必要があります。Ceph では、Ceph 設定ファイルの宣言されたモニター、メタデータサーバー、および OSD の下に **host** の設定が必要です。

**重要**

**localhost** は使用しないでください。完全修飾ドメイン名 (FQDN) ではなく、ノードの短縮名を使用してください。ノード名を取得するサードパーティーのデプロイメントシステムを使用する場合は、**host** の値を指定しないでください。

**mon_addr**

**説明**

クライアントが Ceph モニターへの接続に使用できる **<hostname>:<port>** エントリーのリスト。設定していない場合には、Ceph は **[mon.*]** セクションを検索します。

**型**

String

**必須**

いいえ

**デフォルト**

該当なし

**host**

**説明**

ホスト名です。この設定は、特定のデーモンインスタンス (**[osd.0]** など) に使用します。

**型**

String

**必須**

デーモンインスタンスの場合は Yes。

**デフォルト**

**localhost**

### TCP オプション

Ceph はデフォルトで TCP バッファリングを無効にします。

**ms_tcp_nodelay**

**説明**

Ceph は **ms_tcp_nodelay** を有効化して、各リクエストが即時に送信されます (バッファーなし)。Nagle アルゴリズムを無効にすると、ネットワークのトラフィックが増加し、混雑の原因となります。小さいパケットが多数ある場合は、**ms_tcp_nodelay** を無効にしてみてください。ただし、通常はこれを無効にすると待ち時間が長くなることに注意してください。

**型**

Boolean

**必須**

いいえ

**デフォルト**

**true**

**ms_tcp_rcvbuf**

**説明**

ネットワーク接続の受信側のソケットバッファーのサイズです。デフォルトでは無効です。

**型**

32 ビット整数

**必須**

いいえ

**デフォルト**

**0**

**ms_tcp_read_timeout**

**説明**

クライアントまたはデーモンが別の Ceph デーモンへの要求を行い、未使用の接続を解除しない場合、**tcp read timeout** は、指定した秒数後に接続をアイドル状態として定義します。

**型**

未署名の 64 ビット整数

**必須**

いいえ

**デフォルト**

**900** 15 分。

### バインドオプション

バインオプションは、Ceph OSD デーモンのデフォルトのポート範囲を設定します。デフォルトの範囲は **6800:7100** です。また、Ceph デーモンが IPv6 アドレスにバインドするように設定することもできます。

**重要**

ファイアウォールの設定で、設定したポート範囲を使用できることを確認してください。

**ms_bind_port_min**

**説明**

OSD デーモンがバインドする最小のポート番号。

**型**

32 ビット整数

**デフォルト**

**6800**

**必須**

いいえ

**ms_bind_port_max**

**説明**

OSD デーモンがバインドする最大のポート番号。

**型**

32 ビット整数

**デフォルト**

**7300**

**必須**

いいえ

**ms_bind_ipv6**

**説明**

Ceph デーモンが IPv6 アドレスにバインドするように設定します。

**型**

Boolean

**デフォルト**

**false**

**必須**

いいえ

### 非同期型メッセンジャーオプション

これらの Ceph messenger オプションは、**AsyncMessenger** の動作を設定します。

**ms_async_transport_type**

**説明**

**AsyncMessenger** が使用するトランスポートタイプ。Red Hat は **posix** 設定をサポートしますが、現時点では **dpdk** 設定または **rdma** 設定をサポートしません。POSIX は標準的な TCP/IP ネットワークを使用しており、デフォルト値です。その他のトランスポートタイプは実験的なもので、サポートされて **いません**。

**型**

String

**必須**

いいえ

**デフォルト**

**posix**

**ms_async_op_threads**

**説明**

各 **AsyncMessenger** インスタンスによって使用されるワーカースレッドの初期数。この設定は、レプリカまたはイレイジャーコードチャンクの数に等しく **なければならない** が、CPU コア数が低い場合や、単一のサーバー上での OSD の数が高い場合には低く設定することもできます。

**型**

64 ビット未署名の整数

**必須**

いいえ

**デフォルト**

**3**

**ms_async_max_op_threads**

**説明**

各 **AsyncMessenger** インスタンスによって使用されるワーカースレッドの最大数。OSD ホストの CPU 数が制限されている場合は低い値に設定し、Ceph が CPU を十分に活用していない場合は高い値に設定します。

**型**

64 ビット未署名の整数

**必須**

いいえ

**デフォルト**

**5**

**ms_async_set_affinity**

**説明**

**AsyncMessenger** ワーカーを特定の CPU コアにバインドするには、**true** に設定します。

**型**

Boolean

**必須**

いいえ

**デフォルト**

**true**

**ms_async_affinity_cores**

**説明**

**ms_async_set_affinity** が **true** の場合、この文字列は **AsyncMessenger** ワーカーを CPU コアにバインドする方法を指定します。たとえば、**0,2** はそれぞれワーカー #1 と #2 を CPU コア #0 および #2 にバインドします。**注記:** アフィニティーを手動で設定する場合は、ハイパースレッディングや同様のテクノロジーが原因で作成された仮想 CPU にワーカーを割り当てないようにしてください。これは、物理 CPU コアよりも遅いためです。

**型**

String

**必須**

いいえ

**デフォルト**

**(empty)**

**ms_async_send_inline**

**説明**

キューイングや **AsyncMessenger** スレッドから送信せずに、生成したスレッドからメッセージを直接送信します。このオプションは、CPU コア数の多いシステムではパフォーマンスが低下することが知られているため、デフォルトでは無効になっています。

**型**

Boolean

**必須**

いいえ

**デフォルト**

**false**

### 接続モードの設定オプション

Red Hat Ceph Storage 6 以降では、ほとんどの接続に、暗号化および圧縮に使用されるモードを制御するオプションがあります。

**ms_cluster_mode**

**説明**

Ceph デーモン間のクラスター内の通信に使用される接続モード。複数のモードが一覧表示されている場合は、最初に表示されているモードが優先されます。

**型**

String

**デフォルト**

**crc secure**

**ms_service_mode**

**説明**

ストレージクラスターへの接続時にクライアントが使用できるモードのリスト。

**型**

String

**デフォルト**

**crc secure**

**ms_client_mode**

**説明**

クライアントが Ceph クラスターと対話するときに使用する接続モードのリスト (優先順)。

**型**

String

**デフォルト**

**crc secure**

**ms_mon_cluster_mode**

**説明**

Ceph モニター間で使用する接続モード。

**型**

String

**デフォルト**

**secure crc**

**ms_mon_service_mode**

**説明**

クライアントまたは他の Ceph デーモンがモニターに接続するときに使用できるモードのリスト。

**型**

String

**デフォルト**

**secure crc**

**ms_mon_client_mode**

**説明**

クライアントまたは非モニターデーモンが Ceph モニターへの接続時に使用する接続モードのリスト (優先順)。

**型**

String

**デフォルト**

**secure crc**

### 圧縮モードの設定オプション

Red Hat Ceph Storage 6 以降では、messenger v2 プロトコルを使用して、圧縮モードの設定オプションを使用できます。

**ms_compress_secure**

**説明**

暗号化と圧縮を組み合わせると、ピア間のメッセージのセキュリティーレベルが低下します。暗号化と圧縮の両方が有効な場合は、圧縮設定は無視され、メッセージは圧縮されません。この設定は、オプションで上書きします。キューイングや **AsyncMessenger** スレッドから送信せずに、生成したスレッドからメッセージを直接送信します。このオプションは、CPU コア数の多いシステムではパフォーマンスが低下することが知られているため、デフォルトでは無効になっています。

**型**

Boolean

**デフォルト**

**false**

**ms_osd_compress_mode**

**説明**

Ceph OSD との通信のためにメッセンジャーで使用する圧縮ポリシー。

**型**

String

**デフォルト**

**none**

**有効な選択肢**

**none** または **force**

**ms_osd_compress_min_size**

**説明**

伝送時圧縮の対象となる最小メッセージサイズ。

**型**

Integer

**デフォルト**

**1 Ki**

**ms_osd_compression_algorithm**

**説明**

OSD との接続の圧縮アルゴリズム (優先順)

**型**

String

**デフォルト**

**snappy**

**有効な選択肢**

**snappy**、**zstd**、**zlib**、または **lz4**

# 付録C CEPH MONITOR の設定オプション

デプロイメント時に設定可能な Ceph モニターの設定オプションを以下に示します。

これらの設定オプションは、**ceph config set mon CONFIGURATION_OPTION VALUE** コマンドを使用して設定できます。

**mon_initial_members**

**説明**

起動時のクラスター内の最初のモニターの ID です。指定すると、Ceph は最初のクォーラムを形成するための奇数の数のモニターを必要とします (たとえば、3)。

**型**

String

**デフォルト**

なし

**mon_force_quorum_join**

**説明**

過去にマップから削除されたモニターでも、強制的にクォーラムに参加させます。

**型**

Boolean

**デフォルト**

**False**

**mon_dns_srv_name**

**説明**

モニターのホスト/アドレスを DNS にクエリーする際に使用するサービス名です。

**型**

String

**デフォルト**

**ceph-mon**

**fsid**

**説明**

クラスター ID です。クラスターごとに 1 つになります。

**型**

UUID

**必須**

Yes

**デフォルト**

該当なし。指定されていない場合は、デプロイメントツールによって生成されます。

**mon_data**

**説明**

モニターのデータの場所です。

**型**

String

**デフォルト**

**/var/lib/ceph/mon/$cluster-$id**

**mon_data_size_warn**

**説明**

Ceph は、モニターのデータストアがこのしきい値に達すると、クラスターログで **HEALTH_WARN** ステータスを発行します。デフォルト値は 15GB です。

**型**

Integer

**デフォルト**

**15*1024*1024*1024***

**mon_data_avail_warn**

**説明**

Ceph は、モニターのデータストアで利用可能なディスク領域がこの割合以下になると、クラスターログに **HEALTH_WARN** ステータスを発行します。

**型**

Integer

**デフォルト**

**30**

**mon_data_avail_crit**

**説明**

Ceph は、モニターのデータストアで利用可能なディスク領域がこの割合以下になると、クラスターログに **HEALTH_ERR** ステータスを発行します。

**型**

Integer

**デフォルト**

**5**

**mon_warn_on_cache_pools_without_hit_sets**

**説明**

キャッシュプールに **hit_set_type** パラメーターが設定されていないと、Ceph はクラスターログで **HEALTH_WARN** ステータスを発行します。

**型**

Boolean

**デフォルト**

True

**mon_warn_on_crush_straw_calc_version_zero**

**説明**

CRUSH の **straw_calc_version** がゼロの場合、Ceph はクラスターログの **HEALTH_WARN** ステータスを発行します。詳細は、CRUSH の調整可能パラメーター を参照してください。

**型**

Boolean

**デフォルト**

True

**mon_warn_on_legacy_crush_tunables**

**説明**

CRUSH の調整可能なパラメーターが古くなり過ぎた場合 (**mon_min_crush_required_version** よりも古い場合)、Ceph はクラスターログで **HEALTH_WARN** ステータスを発行します。

**型**

Boolean

**デフォルト**

True

**mon_crush_min_required_version**

**説明**

この設定では、クラスターが必要とする最小のチューナブルプロファイルバージョンを定義します。

**型**

String

**デフォルト**

**hammer**

**mon_warn_on_osd_down_out_interval_zero**

**説明**

**mon_osd_down_out_interval** 設定がゼロの場合、Ceph はクラスターログで **HEALTH_WARN** ステータスを発行します。これは、**noout** フラグが設定されている場合にもリーダーと同様の動作をするためです。管理者は、**noout** フラグを設定してクラスターのトラブルシューティングが容易になります。Ceph は、管理者が設定がゼロであることを認識するために警告を発します。

**型**

Boolean

**デフォルト**

True

**mon_cache_target_full_warn_ratio**

**説明**

**cache_target_full** と **target_max_object** の比率で、Ceph により警告が表示されます。

**型**

浮動小数点 (Float)

**デフォルト**

**0.66**

**mon_health_data_update_interval**

**説明**

クォーラム内のモニターがピアとヘルスステータスを共有する頻度 (秒単位)。マイナスの数値を入力すると、ヘルス更新が無効になります。

**型**

浮動小数点 (Float)

**デフォルト**

**60**

**mon_health_to_clog**

**説明**

この設定により、Ceph が定期的にクラスターログにヘルスサマリーを送信することができます。

**型**

Boolean

**デフォルト**

True

**mon_health_detail_to_clog**

**説明**

この設定により、Ceph が定期的にクラスターログにヘルス詳細を送信することができます。

**型**

Boolean

**デフォルト**

True

**mon_op_complaint_time**

**説明**

更新が行われなかった後、Ceph Monitor 操作がブロックされたと見なされるまでの秒数。

**型**

Integer

**デフォルト**

30

**mon_health_to_clog_tick_interval**

**説明**

モニターが正常性の要約をクラスターログに送信する頻度 (秒単位)。正数以外の数値を指定すると、この設定は無効になります。現在のヘルスサマリーが空であったり、前回と同じであったりする場合、モニターはステータスをクラスターログに送信しません。

**型**

Integer

**デフォルト**

60.000000

**mon_health_to_clog_interval**

**説明**

モニターが正常性の要約をクラスターログに送信する頻度 (秒単位)。正数以外の数値を指定すると、この設定は無効になります。モニターは常にクラスターログにサマリーを送信します。

**型**

Integer

**デフォルト**

600

**mon_osd_full_ratio**

**説明**

OSD が **full** とみなされるまでのディスク領域のパーセンテージ。

**型**

浮動小数点

**デフォルト**

**.95**

**mon_osd_nearfull_ratio**

**説明**

OSD がほぼ **nearfull** とみなされるまでのディスク領域のパーセンテージ。

**型**

浮動小数点 (Float)

**デフォルト**

**.85**

**mon_sync_trim_timeout**

**説明, 型**

Double

**デフォルト**

**30.0**

**mon_sync_heartbeat_timeout**

**説明, 型**

Double

**デフォルト**

**30.0**

**mon_sync_heartbeat_interval**

**説明, 型**

Double

**デフォルト**

**5.0**

**mon_sync_backoff_timeout**

**説明, 型**

Double

**デフォルト**

**30.0**

**mon_sync_timeout**

**説明**

モニターが、更新メッセージをあきらめて再びブートストラップを行うまで、同期プロバイダーから次のメッセージを待つ秒数。

**型**

Double

**デフォルト**

**60.000000**

**mon_sync_max_retries**

**説明, 型**

Integer

**デフォルト**

**5**

**mon_sync_max_payload_size**

**説明**

同期ペイロードの最大サイズ (単位: バイト) です。

**型**

32 ビット整数

**デフォルト**

**1045676**

**paxos_max_join_drift**

**説明**

モニターデータストアを最初に同期させるまでの、Paxos 最大反復回数です。モニターは、ピアが自分よりも先に進んでいると判断すると、先に進む前にまずデータストアと同期します。

**型**

Integer

**デフォルト**

**10**

**paxos_stash_full_interval**

**説明**

PaxosService の状態のフルコピーを隠す頻度 (コミット数)。現在、この設定は **mds**、**mon**、**auth**、および **mgr** PaxosServices のみに影響します。

**型**

Integer

**デフォルト**

25

**paxos_propose_interval**

**説明**

この時間間隔で更新情報を集めてから、マップの更新を提案します。

**型**

Double

**デフォルト**

**1.0**

**paxos_min**

**説明**

維持する paxos の状態の最小数

**型**

Integer

**デフォルト**

500

**paxos_min_wait**

**説明**

活動していない期間の後に更新を収集するための最小時間。

**型**

Double

**デフォルト**

**0.05**

**paxos_trim_min**

**説明**

トリミング前に許容される追加提案の数

**型**

Integer

**デフォルト**

250

**paxos_trim_max**

**説明**

一度にトリミングする追加提案の最大数

**型**

Integer

**デフォルト**

500

**paxos_service_trim_min**

**説明**

トリムのトリガーとなる最小のバージョン数 (0 であれば無効)

**型**

Integer

**デフォルト**

250

**paxos_service_trim_max**

**説明**

1 回の提案中にトリミングするバージョン数の最大値 (0 であれば無効)

**型**

Integer

**デフォルト**

500

**mon_max_log_epochs**

**説明**

1 回の提案中にトリミングするログエポック数の最大値

**型**

Integer

**デフォルト**

500

**mon_max_pgmap_epochs**

**説明**

1 回の提案中にトリミングする pgmap エポック数の最大値

**型**

Integer

**デフォルト**

500

**mon_mds_force_trim_to**

**説明**

モニターがこのポイントまで mdsmaps をトリミングするのを強制します (0 は無効、危険なので使用には注意が必要)。

**型**

Integer

**デフォルト**

0

**mon_osd_force_trim_to**

**説明**

指定したエポックでクリーンではない PG があっても、モニターがこのポイントまで osdmaps をトリミングするのを強制します (0 は無効、危険なので使用には注意が必要)。

**型**

Integer

**デフォルト**

0

**mon_osd_cache_size**

**説明**

基礎となるストアのキャッシュに依存しない、osdmaps のキャッシュサイズ

**型**

Integer

**デフォルト**

500

**mon_election_timeout**

**説明**

選択の提案側で、すべての ACK を待つ最長の時間 (秒単位)

**型**

浮動小数点 (Float)

**デフォルト**

**5**

**mon_lease**

**説明**

モニターのバージョンのリース期間 (秒単位)

**型**

浮動小数点 (Float)

**デフォルト**

**5**

**mon_lease_renew_interval_factor**

**説明**

**mon lease** * **mon lease renew interval factor** は、リーダーが他のモニターのリースを更新する間隔になります。係数は **1.0** 未満でなければなりません。

**型**

浮動小数点 (Float)

**デフォルト**

**0.6**

**mon_lease_ack_timeout_factor**

**説明**

リーダーは、プロバイダーがリース拡張を承認するまで **mon lease** * **mon lease ack timeout factor** を待機します。

**型**

浮動小数点 (Float)

**デフォルト**

**2.0**

**mon_accept_timeout_factor**

**説明**

Leader は **mon lease** * **mon accept timeout factor** を待ち、リクエスターが Paxos の更新を受け入れるのを待機します。また、Paxos の回復期にも同様の目的で使用されます。

**型**

浮動小数点 (Float)

**デフォルト**

**2.0**

**mon_min_osdmap_epochs**

**説明**

常時保持する OSD マップエポックの最小数

**型**

32 ビット整数

**デフォルト**

**500**

**mon_max_pgmap_epochs**

**説明**

モニターが保持すべき PG マップエポックの最大数

**型**

32 ビット整数

**デフォルト**

**500**

**mon_max_log_epochs**

**説明**

モニターが保持すべきログエポックの最大数

**型**

32 ビット整数

**デフォルト**

**500**

**clock_offset**

**説明**

システムクロックをどれだけオフセットするか。詳細は、**Clock.cc** を参照してください。

**型**

Double

**デフォルト**

**0**

**mon_tick_interval**

**説明**

モニターの目盛りの間隔 (秒単位)

**型**

32 ビット整数

**デフォルト**

**5**

**mon_clock_drift_allowed**

**説明**

モニター間で許容されるクロックドリフト (秒単位)

**型**

浮動小数点 (Float)

**デフォルト**

**.050**

**mon_clock_drift_warn_backoff**

**説明**

クロックドリフト警告のための指数バックオフ

**型**

浮動小数点 (Float)

**デフォルト**

**5**

**mon_timecheck_interval**

**説明**

リーダーの時刻チェック (クロックドリフトチェック) 間隔 (秒単位)

**型**

浮動小数点 (Float)

**デフォルト**

**300.0**

**mon_timecheck_skew_interval**

**説明**

スキューがあった場合のリーダーの時刻チェック (クロックドリフトチェック) 間隔 (秒単位)

**型**

浮動小数点 (Float)

**デフォルト**

**30.0**

**mon_max_osd**

**説明**

クラスターで許容される OSD の最大数

**型**

32 ビット整数

**デフォルト**

**10000**

**mon_globalid_prealloc**

**説明**

クラスター内のクライアントおよびデーモンに事前に割り当てるグローバル ID の数

**型**

32 ビット整数

**デフォルト**

**10000**

**mon_sync_fs_threshold**

**説明**

指定された数のオブジェクトを書き込む際に、ファイルシステムと同期します。無効にするには **0** に設定します。

**型**

32 ビット整数

**デフォルト**

**5**

**mon_subscribe_interval**

**説明**

サブスクリプションの更新間隔 (秒単位)。サブスクリプションメカニズムにより、クラスターマップやログ情報を取得することができます。

**型**

Double

**デフォルト**

**86400.000000**

**mon_stat_smooth_intervals**

**説明**

最後の **N** PG マップに対する統計は、Ceph によりスムーズになります。

**型**

Integer

**デフォルト**

**6**

**mon_probe_timeout**

**説明**

モニターがブートストラップを行うまで、ピアを探すために待機する秒数

**型**

Double

**デフォルト**

**2.0**

**mon_daemon_bytes**

**説明**

メタデータサーバーおよび OSD メッセージのメッセージメモリー容量 (単位: バイト)

**型**

64 ビット整数未署名

**デフォルト**

**400ul << 20**

**mon_max_log_entries_per_event**

**説明**

1イベントあたりのログエントリーの最大数

**型**

Integer

**デフォルト**

**4096**

**mon_osd_prime_pg_temp**

**説明**

クラスター外の OSD がクラスターに戻ってきたときに、以前の OSD で PGMap のプライミングを行うことを有効または無効にします。**true** 設定では、クライアントは、PG のピア化として OSD で新たに実行するまで、以前の OSD を引き続き使用します。

**型**

Boolean

**デフォルト**

**true**

**mon_osd_prime_pg_temp_max_time**

**説明**

クラスター外の OSD がクラスターに戻ってきたときに、モニターが PGMAP のプライミングを試みる時間 (秒単位)

**型**

浮動小数点 (Float)

**デフォルト**

**0.5**

**mon_osd_prime_pg_temp_max_time_estimate**

**説明**

すべての PG を並行してプライミングするまでに、各 PG での時間の最大推定値

**型**

浮動小数点 (Float)

**デフォルト**

**0.25**

**mon_osd_allow_primary_affinity**

**説明**

osdmap で **primary_affinity** を設定できるようにします。

**型**

Boolean

**デフォルト**

False

**mon_osd_pool_ec_fast_read**

**説明**

プールでの高速読み込みオンにするかどうか。作成時に **fast_read** が指定されていない場合に、新たに作成されたイレイジャープールのデフォルト設定として使用します。

**型**

Boolean

**デフォルト**

False

**mon_mds_skip_sanity**

**説明**

バグ発生に関わらず続行したい際に、FSMap の安全アサーションをスキップします。FSMap のサニティーチェックに失敗すると Monitor は終了しますが、このオプションを有効にすることでそれを無効にすることができます。

**型**

Boolean

**デフォルト**

False

**mon_max_mdsmap_epochs**

**説明**

1 回の提案中にトリミングする mdsmap エポック数の最大値

**型**

Integer

**デフォルト**

500

**mon_config_key_max_entry_size**

**説明**

config-key エントリーの最大サイズ (単位: バイト)

**型**

Integer

**デフォルト**

65536

**mon_warn_pg_not_scrubbed_ratio**

**説明**

警告するスクラブ最大間隔を超えたスクラブ最大間隔の割合。

**型**

float

**デフォルト**

0.5

**mon_warn_pg_not_deep_scrubbed_ratio**

**説明**

警告するディープスクラブ間隔を超えたディープスクラブ間隔の割合

**型**

float

**デフォルト**

0.75

**mon_scrub_interval**

**説明**

保存されているチェックサムと、保存されているすべての鍵の計算されたチェックサムを比較して、モニターがストアをスクラブする頻度 (秒単位)

**型**

Integer

**デフォルト**

3600*24

**mon_scrub_timeout**

**説明**

mon クォーラム参加者のスクラブを再開するためのタイムアウトが最新のチャンクに応答しません。

**型**

Integer

**デフォルト**

5 min

**mon_scrub_max_keys**

**説明**

都度スクラブするキーの最大数

**型**

Integer

**デフォルト**

100

**mon_scrub_inject_crc_mismatch**

**説明**

Ceph Monitor スクラブに CRC 不一致を挿入する確率。

**型**

Integer

**デフォルト**

3600*24

**mon_scrub_inject_missing_keys**

**説明**

欠落しているキーを mon スクラブに挿入する確率。

**型**

float

**デフォルト**

0

**mon_compact_on_start**

**説明**

**ceph-mon** の起動時に Ceph Monitor ストアとして使用されるデータベースを圧縮します。手動コンパクションは、通常のコンパクションが機能しない場合に、モニターデータベースを縮小し、そのパフォーマンスを向上させるのに役立ちます。

**型**

Boolean

**デフォルト**

False

**mon_compact_on_bootstrap**

**説明**

ブートストラップ時に Ceph Monitor ストアとして使用されるデータベースを圧縮します。ブートストラップ後に、モニターはクォーラムを作るためにお互いにプロービングを開始します。クォーラムに参加する前にタイムアウトした場合は、やり直して、再びブートストラップを行います。

**型**

Boolean

**デフォルト**

False

**mon_compact_on_trim**

**説明**

古い状態をトリミングする際に、ある接頭辞 (paxos を含む) をコンパクト化します。

**型**

Boolean

**デフォルト**

True

**mon_cpu_threads**

**説明**

モニター上で CPU 負荷の高い作業を行うためのスレッドの数

**型**

Boolean

**デフォルト**

True

### mon_osd_mapping_pgs_per_chunk

**説明**

配置グループから OSD へのマッピングをチャンクで計算します。このオプションで、チャンクごとの配置グループ数を指定します。

**型**

Integer

**デフォルト**

4096

### mon_osd_max_split_count

**説明**

分割を作成させるための関係する OSD ごとの最大の PG 数。プールの **pg_num** を増やすと、配置グループは、そのプールを提供するすべての OSD で分割されます。PG を分割する際、極端な倍数は避けるべきです。

**型**

Integer

**デフォルト**

300

### rados_mon_op_timeout

**説明**

rados 操作からのエラーを返す前に、モニターからの応答を待つ時間 (秒数)。0 は制限、または待ち時間がないことを意味します。

**型**

Double

**デフォルト**

0

**関連情報**

- プール値
- CRUSH の調整可能パラメーター

# 付録D CEPHX の設定オプション

デプロイメント時に設定可能な Cephx の設定オプションを以下に示します。

auth_cluster_required

**説明**

これが有効な場合には、Red Hat Ceph Storage クラスターデーモン **ceph-mon** および **ceph-osd** は相互に認証する必要があります。有効な設定は **cephx** または **none** です。

**型**

String

**必須**

いいえ

**デフォルト**

**cephx**.

auth_service_required

**説明**

有効にすると、Red Hat Ceph Storage クラスターデーモンは、Ceph サービスにアクセスするために、Ceph クライアントが Red Hat Ceph Storage クラスターと認証することを要求します。有効な設定は **cephx** または **none** です。

**型**

String

**必須**

いいえ

**デフォルト**

**cephx**.

auth_client_required

**説明**

有効にすると、Ceph クライアントは、Red Hat Ceph Storage クラスターが Ceph クライアントと認証することを要求します。有効な設定は **cephx** または **none** です。

**型**

String

**必須**

いいえ

**デフォルト**

**cephx**.

keyring

**説明**

キーリングファイルのパス

**型**

String

**必須**

いいえ

**デフォルト**

**/etc/ceph/$cluster.$name.keyring,/etc/ceph/$cluster.keyring,/etc/ceph/keyring,/etc/ceph/keyring.bin**

keyfile

**説明**

キーファイル (つまり、キーのみを含むファイル) へのパス

**型**

String

**必須**

いいえ

**デフォルト**

なし

key

**説明**

キー (つまり、キーそのもののテキスト文字列)。推奨されません。

**型**

String

**必須**

いいえ

**デフォルト**

なし

ceph-mon

**場所**

**$mon_data/keyring**

**ケイパビリティー**

**mon 'allow *'**

ceph-osd

**場所**

**$osd_data/keyring**

**ケイパビリティー**

**mon 'allow profile osd' osd 'allow *'**

radosgw

**場所**

**$rgw_data/keyring**

**ケイパビリティー**

**mon 'allow rwx' osd 'allow rwx'**

cephx_require_signatures

**説明**

**true** に設定した場合には、Ceph クライアントと Red Hat Ceph Storage クラスター間の全メッセージトラフィック、および Red Hat Ceph Storage クラスターを設定するデーモン間での署名が必要です。

**型**

Boolean

**必須**

いいえ

**デフォルト**

**false**

cephx_cluster_require_signatures

**説明**

**true** に設定した場合には、Ceph では、Red Hat Ceph Storage クラスターを設定する Ceph デーモン間のすべてのメッセージトラフィックに対する署名が必要です。

**型**

Boolean

**必須**

いいえ

**デフォルト**

**false**

cephx_service_require_signatures

**説明**

**true** に設定した場合には、Ceph クライアントと Red Hat Ceph Storage クラスター間のすべてのメッセージトラフィックに対する署名が必要です。

**型**

Boolean

**必須**

いいえ

**デフォルト**

**false**

cephx_sign_messages

**説明**

Ceph のバージョンがメッセージ署名をサポートしている場合、Ceph はすべてのメッセージに署名し、メッセージが偽装されないようにします。

**型**

Boolean

**デフォルト**

**true**

auth_service_ticket_ttl

**説明**

Red Hat Ceph Storage クラスターが Ceph クライアントに認証用のチケットを送信すると、クラスターはそのチケットに生存時間を割り当てます。

**型**

Double

**デフォルト**

**60*60**

# 付録E プール、配置グループ、および CRUSH の設定オプション

プール、配置グループ、および CRUSH アルゴリズムを管理する Ceph のオプションです。

**mon_allow_pool_delete**

**説明**

モニターがプールを削除することができます。RHCS 3 以降のリリースでは、データ保護のための追加措置として、モニターはデフォルトでプールを削除できません。

**型**

Boolean

**デフォルト**

**false**

**mon_max_pool_pg_num**

**説明**

プールあたりの配置グループの最大数

**型**

Integer

**デフォルト**

**65536**

**mon_pg_create_interval**

**説明**

同じ Ceph OSD デーモンでの PG 作成の間の秒数

**型**

浮動小数点 (Float)

**デフォルト**

**30.0**

**mon_pg_stuck_threshold**

**説明**

PG がスタックしていると判断できるまでの秒数

**型**

32 ビット整数

**デフォルト**

**300**

**mon_pg_min_inactive**

**説明**

Ceph は、**mon_pg_stuck_threshold** より長く非アクティブのままの PG の数がこの設定を超える場合に、クラスターログに **HEALTH_ERR** ステータスを発行します。デフォルト設定は 1 つの PG です。正数以外の数値を指定すると、この設定は無効になります。

**型**

Integer

**デフォルト**

**1**

mon_pg_warn_min_per_osd

**説明**

Ceph は、クラスター内の OSD ごとの PG の平均数がこの設定よりも小さい場合に、クラスターログで **HEALTH_WARN** ステータスを発行します。正数以外の数値を指定すると、この設定は無効になります。

**型**

Integer

**デフォルト**

**30**

mon_pg_warn_max_per_osd

**説明**

Ceph は、クラスター内の OSD ごとの PG の平均数がこの設定よりも大きい場合に、クラスターログの **HEALTH_WARN** ステータスを発行します。正数以外の数値を指定すると、この設定は無効になります。

**型**

Integer

**デフォルト**

**300**

mon_pg_warn_min_objects

**説明**

クラスター内のオブジェクトの総数がこの数以下の場合は警告を発生しません。

**型**

Integer

**デフォルト**

**1000**

mon_pg_warn_min_pool_objects

**説明**

オブジェクト数がこの数以下のプールには警告を発生しません。

**型**

Integer

**デフォルト**

**1000**

mon_pg_check_down_all_threshold

**説明**

**down** OSD のしきい値 (パーセント) で、Ceph はすべての PG をチェックして、それらがスタックまたは古くなっていることを確認します。

**型**

浮動小数点 (Float)

**デフォルト**

**0.5**

mon_pg_warn_max_object_skew

**説明**

プール内のオブジェクトの平均数 **mon pg warn max object skew** を超える場合、Ceph はクラスターログで **HEALTH_WARN** ステータスを発行します。正数以外の数値を指定すると、この設定は無効になります。

**型**

浮動小数点 (Float)

**デフォルト**

**10**

mon_delta_reset_interval

**説明**

Ceph が PG デルタをゼロにリセットするまでの非アクティブ時の秒数。Ceph は、各プールの使用済み容量のデルタを追跡し、管理者がリカバリーの進捗状況やパフォーマンスを評価するのに役立てます。

**型**

Integer

**デフォルト**

**10**

mon_osd_max_op_age

**説明**

**HEALTH_WARN** ステータスを発行する前に操作が完了するまでの最大期間 (秒単位)。

**型**

浮動小数点 (Float)

**デフォルト**

**32.0**

osd_pg_bits

**説明**

Ceph OSD デーモンごとの配置グループのビット数

**型**

32 ビット整数

**デフォルト**

**6**

osd_pgp_bits

**説明**

配置目的の配置グループ (PGP) の Ceph OSD デーモンあたりのビット数

**型**

32 ビット整数

**デフォルト**

**6**

osd_crush_chooseleaf_type

**説明**

CRUSH ルールで **chooseleaf** に使用するバケットタイプ。名前ではなく従来のランクを使用します。

**型**

32 ビット整数

**デフォルト**

**1**.通常は、1 つまたは複数の Ceph OSD デーモンを含むホストです。

osd_pool_default_crush_replicated_ruleset

**説明**

レプリケートされたプールを作成する際に使用するデフォルトの CRUSH ルールセット

**型**

8 ビット整数

**デフォルト**

**0**

osd_pool_erasure_code_stripe_unit

**説明**

イレイジャーコード化されたプールのオブジェクトストライプのチャンクのデフォルトサイズをバイト単位で設定します。サイズ S のすべてのオブジェクトは N ストライプとして格納され、各データチャンクは **stripe unit** バイトを受け取ります。**N * stripe unit** バイトの各ストライプは、個別にエンコード/エンコードされます。このオプションは、イレイジャーコードプロファイルの **stripe_unit** 設定で上書きできます。

**型**

32 ビット符号なし整数

**デフォルト**

**4096**

osd_pool_default_size

**説明**

プール内のオブジェクトのレプリカ数を設定します。デフォルト値は、**ceph osd pool set {pool-name} size {size}** と同じです。

**型**

32 ビット整数

**デフォルト**

**3**

osd_pool_default_min_size

**説明**

プール内のオブジェクトに対して、クライアントへの書き込み操作を確認するための、書き込み

済みレプリカの最小数を設定します。最小値が満たされていない場合、Ceph はクライアントへの書き込みを確認しません。この設定により、**degraded** モードで動作している場合にレプリカの最小数を確保できます。

**型**

32 ビット整数

**デフォルト**

**0** (これは、特定の最小値がないことを意味します)**0** の場合、最小は **size - (size / 2)** になります。

osd_pool_default_pg_num

**説明**

プールの配置グループのデフォルト数。デフォルト値は、**mkpool** で **pg_num** と同じです。

**型**

32 ビット整数

**デフォルト**

**32**

osd_pool_default_pgp_num

**説明**

プールに対する配置の配置グループのデフォルト数です。デフォルト値は、**mkpool** で **pgp_num** と同じです。PG と PGP は同じであるべきです。

**型**

32 ビット整数

**デフォルト**

**0**

osd_pool_default_flags

**説明**

新しいプールのデフォルトフラグ

**型**

32 ビット整数

**デフォルト**

**0**

osd_max_pgls

**説明**

リストアップする配置グループの最大数。大きな数を要求するクライアントは、Ceph OSD デーモンを拘束できます。

**型**

未署名の 64 ビット整数

**デフォルト**

**1024**

**備考**

デフォルトで問題ありません。

osd_min_pg_log_entries

**説明**

ログファイルをトリミングする際に維持する配置グループログの最小数

**型**

32 ビット符号なし整数

**デフォルト**

**250**

osd_default_data_pool_replay_window

**説明**

クライアントが要求を再生するのを OSD が待つ時間 (秒)

**型**

32 ビット整数

**デフォルト**

**45**

# 付録F OBJECT STORAGE DAEMON (OSD) の設定オプション

デプロイメント時に設定可能な Ceph Object Storage Daemon (OSD) の設定オプションを以下に示します。

これらの設定オプションは、**ceph config set osd CONFIGURATION_OPTION VALUE** コマンドを使用して設定できます。

**osd_uuid**

**説明**

Ceph OSD の Universally Unique Identifier (UUID)

**型**

UUID

**デフォルト**

UUID

**備考**

**osd uuid** は単一の Ceph OSD に適用されます。 **fsid** はクラスター全体に適用されます。

**osd_data**

**説明**

OSD のデータへのパスCeph のデプロイ時にディレクトリーを作成する必要があります。OSD データ用のドライブをこのマウントポイントにマウントします。

> IMPORTANT: Red Hat does not recommend changing the default.

**型**

String

**デフォルト**

**/var/lib/ceph/osd/$cluster-$id**

**osd_max_write_size**

**説明**

書き込みの最大サイズ (メガバイト)

**型**

32 ビット整数

**デフォルト**

**90**

**osd_client_message_size_cap**

**説明**

メモリー上で許可される最大のクライアントデータメッセージ

**型**

64 ビット整数未署名

**デフォルト**

500 MB のデフォルト**500*1024L*1024L**

osd_class_dir

**説明**

RADOS クラスのプラグインのクラスパス

**型**

String

**デフォルト**

**$libdir/rados-classes**

osd_max_scrubs

**説明**

Ceph OSD ごとの同時スクラブ操作の最大数

**型**

32 ビット整数

**デフォルト**

**1**

osd_scrub_thread_timeout

**説明**

スクラブスレッドがタイムアウトするまでの最大時間 (秒単位)

**型**

32 ビット整数

**デフォルト**

**60**

osd_scrub_finalize_thread_timeout

**説明**

スクラブ最終スレッドがタイムアウトするまでの最大時間 (秒単位)

**型**

32 ビット整数

**デフォルト**

**60*10**

osd_scrub_begin_hour

**説明**

これにより、スクラブは 1 日のこの時間以降に制限されます。**osd_scrub_begin_hour = 0** および **osd_scrub_end_hour = 0** を使用して、1 日中スクラブできるようにします。**osd_scrub_end_hour** とともに、スクラブが発生する時間枠を定義できます。ただし、配置グループのスクラブ間隔が **osd_scrub_max_interval** を超えている限り、時間ウィンドウが許可するかどうかに関係なく、スクラブが実行されます。

**型**

Integer

**デフォルト**

**0**

**許容範囲**

**[0,23]**

osd_scrub_end_hour

**説明**

これにより、スクラブはこれより前の1時間に制限されます。**osd_scrub_begin_hour = 0** および **osd_scrub_end_hour = 0** を使用して、1日住スクラブできるようにします。**osd_scrub_begin_hour** とともに、スクラブが発生する時間枠を定義できます。ただし、配置グループのスクラブ間隔が **osd_scrub_max_interval** を超えている限り、時間ウィンドウが許可するかどうかに関係なく、スクラブが実行されます。

**型**

Integer

**デフォルト**

**0**

**許容範囲**

**[0,23]**

osd_scrub_load_threshold

**説明**

最大の負荷。(**getloadavg()** 関数で定義された) システムの負荷がこの数値よりも大きい場合、Ceph はスクラブを実行しません。デフォルトは **0.5** です。

**型**

浮動小数点 (Float)

**デフォルト**

**0.5**

osd_scrub_min_interval

**説明**

Red Hat Ceph Storage クラスターの負荷が低いときに、Ceph OSD をスクラブする最小の間隔 (秒単位)

**型**

浮動小数点 (Float)

**デフォルト**

1日1回。**60*60*24**

osd_scrub_max_interval

**説明**

クラスター負荷に関わらず Ceph OSD をスクラビングする最大の間隔 (秒単位)。

**型**

浮動小数点 (Float)

**デフォルト**

1週間に1回。**7*60*60*24**

osd_scrub_interval_randomize_ratio

**説明**

比率を取り、**osd scrub min interval** および **osd scrub max interval** の間隔の間でスケジュールされたスクラブをランダム化します。

**型**

浮動小数点 (Float)

**デフォルト**

**0.5**。

mon_warn_not_scrubbed

**説明**

スクラブされていない PG について警告する **osd_scrub_interval** からの秒数。

**型**

Integer

**デフォルト**

**0** (警告なし)。

osd_scrub_chunk_min

**説明**

オブジェクトストアは、ハッシュの境界で終わるチャンクに分割されています。チャンキースクラブの場合、Ceph はオブジェクトを 1 チャンクずつスクラブし、そのチャンクへの書き込みをブロックします。**osd scrub chunk min** 設定は、スクラビングするチャンクの最小数を表します。

**型**

32 ビット整数

**デフォルト**

**5**

osd_scrub_chunk_max

**説明**

スクラブするチャンクの最大数

**型**

32 ビット整数

**デフォルト**

**25**

osd_scrub_sleep

**説明**

ディープスクラブ操作の間のスリープ時間

**型**

浮動小数点 (Float)

**デフォルト**

**0** (またはオフ)

osd_scrub_during_recovery

**説明**

リカバリー時のスクラブを可能にします。

**型**

ブール (Bool)

**デフォルト**

**false**

osd_scrub_invalid_stats

**説明**

無効と判定された統計情報を修正するために、強制的に追加のスクラブを実行します。

**型**

ブール (Bool)

**デフォルト**

**true**

osd_scrub_priority

**説明**

クライアント I/O に対するスクラブ操作のキューの優先順位を制御します。

**型**

32 ビット符号なし整数

**デフォルト**

**5**

osd_requested_scrub_priority

**説明**

ワークキュー上のユーザー要求のスクラブに設定された優先順位。この値が **osd_client_op_priority** より小さい場合、スクラブがクライアント操作をブロックしているときに、この値を **osd_client_op_priority** の値まで上げることができます。

**型**

32 ビット符号なし整数

**デフォルト**

**120**

osd_scrub_cost

**説明**

キューのスケジューリングのために、スクラブ操作のコストをメガバイト単位で表したもの。

**型**

32 ビット符号なし整数

**デフォルト**

**52428800**

osd_deep_scrub_interval

**説明**

すべてのデータを完全に読み込むディープスクラビングのための間隔。**osd scrub load threshold** パラメーターは、この設定には影響を与えません。

**型**

浮動小数点 (Float)

**デフォルト**

1 週間に 1 回。**60*60*24*7**

osd_deep_scrub_stride

**説明**

デープスクラブを実施する際の読み取りサイズ

**型**

32 ビット整数

**デフォルト**

512 KB。**524288**

mon_warn_not_deep_scrubbed

**説明**

スクラビングされていない PG について警告する **osd_deep_scrub_interval** からの秒数。

**型**

Integer

**デフォルト**

**0** (警告なし)。

osd_deep_scrub_randomize_ratio

**説明**

スクラブが無作為にディープスクラビングになる変化 (**osd_deep_scrub_interval** が経過する可能性も)

**型**

浮動小数点 (Float)

**デフォルト**

**0.15** または 15%

osd_deep_scrub_update_digest_min_age

**説明**

スクラブがオブジェクト全体のダイジェストを更新するまでに、オブジェクトが何秒経過していなければならないか。

**型**

Integer

**デフォルト**

**7200** (120 時間)。

osd_deep_scrub_large_omap_object_key_threshold

**説明**

これより多くの OMAP キーを持つオブジェクトに遭遇した場合の警告。

**型**

Integer

**デフォルト**

**200000**

osd_deep_scrub_large_omap_object_value_sum_threshold

**説明**

これより多くの OMAP キーバイトを持つオブジェクトに遭遇した場合に警告が表示されます。

**型**

Integer

**デフォルト**

**1 G**

osd_delete_sleep

**説明**

次の削除トランザクションまでのスリープ時間 (秒)。これにより、配置グループの削除プロセスにスロットリングを適用されます。

**型**

浮動小数点 (Float)

**デフォルト**

**0.0**

osd_delete_sleep_hdd

**説明**

HDD の次の削除トランザクションまでのスリープ時間 (秒)。

**型**

浮動小数点 (Float)

**デフォルト**

**5.0**

osd_delete_sleep_ssd

**説明**

SSD の次の削除トランザクションまでのスリープ時間 (秒)。

**型**

浮動小数点 (Float)

**デフォルト**

**1.0**

osd_delete_sleep_hybrid

**説明**

Ceph OSD データが HDD にあり、OSD ジャーナルまたは WAL と DB が SSD にある場合の、次の削除トランザクションまでのスリープ時間 (秒)。

**型**

浮動小数点 (Float)

**デフォルト**

**1.0**

osd_op_num_shards

**説明**

クライアント操作のためのシャード数

**型**

32 ビット整数

**デフォルト**

**0**

osd_op_num_threads_per_shard

**説明**

クライアント操作のためのシャードあたりのスレッド数

**型**

32 ビット整数

**デフォルト**

**0**

osd_op_num_shards_hdd

**説明**

HDD 操作のためのシャード数

**型**

32 ビット整数

**デフォルト**

**5**

osd_op_num_threads_per_shard_hdd

**説明**

HDD 操作のためのシャードあたりのスレッド数

**型**

32 ビット整数

**デフォルト**

**1**

osd_op_num_shards_ssd

**説明**

SSD 操作のためのシャード数

**型**

32 ビット整数

**デフォルト**

**8**

osd_op_num_threads_per_shard_ssd

**説明**

SSD 操作のためのシャードあたりのスレッド数

**型**

32 ビット整数

**デフォルト**

**2**

osd_op_queue

**説明**

Ceph OSD 内での操作の優先順位付けに使用されるキューのタイプを設定します。OSD デーモンの再起動が必要です。

**型**

String

**デフォルト**

**wpq**

**有効な選択肢**

**wpq**、**mclock_scheduler**、**debug_random**

**重要**

mClock OSD スケジューラーは、テクノロジープレビュー機能としてのみご利用いただけます。テクノロジープレビュー機能は、実稼働環境での Red Hat サービスレベルアグリーメント (SLA) ではサポートされておらず、機能的に完全ではない可能性があるため、Red Hat では実稼働環境での使用を推奨していません。テクノロジープレビューの機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。詳細は、Red Hat テクノロジープレビュー機能のサポート範囲 を参照してください。

**osd_op_queue_cut_off**

**説明**

ストリクトキューと通常のキューに送信する操作の優先度を設定します。OSD デーモンの再起動が必要です。
low に設定すると、すべてのレプリケーション以上の操作はストリクトキューに送信され、high に設定するとレプリケーション確認操作以上の操作のみストリクトキューに送信されます。

高く設定した場合、特に **osd_op_queue** 設定の **wpq** オプションと組み合わせると、クラスター内の一部の Ceph OSD が非常にビジーな場合に役立ちます。レプリケーショントラフィックの処理で非常にビジーな Ceph OSD は、これらの設定がないと、OSD のプライマリークライアントトラフィックを枯渇させる可能性があります。

**型**

String

**デフォルト**

**高**

**有効な選択肢**

**low**、**high**、**debug_random**

osd_client_op_priority

**説明**

クライアントの操作に設定されている優先順位。これは、**osd recovery op priority** と相対的になります。

**型**

32 ビット整数

**デフォルト**

**63**

**有効な範囲**

1-63

osd_recovery_op_priority

**説明**

復元の操作に設定されている優先順位。これは、**osd client op priority** と相対的になります。

**型**

32 ビット整数

**デフォルト**

**3**

**有効な範囲**

1-63

osd_op_thread_timeout

**説明**

Ceph OSD 操作スレッドのタイムアウト (秒単位)

**型**

32 ビット整数

**デフォルト**

**15**

osd_op_complaint_time

**説明**

指定された秒数が経過すると、クレームに値する操作になります。

**型**

浮動小数点 (Float)

**デフォルト**

**30**

osd_disk_threads

**説明**

スクラビングやスナップトリミングなど、バックグラウンドでのディスクを多用する OSD 操作に使用されるディスクスレッドの数

**型**

32 ビット整数

**デフォルト**

**1**

osd_op_history_size

**説明**

追跡する完了した操作の最大数

**型**

32 ビット未署名の整数

**デフォルト**

**20**

osd_op_history_duration

**説明**

追跡する最も古い完了した操作

**型**

32 ビット未署名の整数

**デフォルト**

**600**

osd_op_log_threshold

**説明**

一度に表示する操作ログの数

**型**

32 ビット整数

**デフォルト**

**5**

osd_op_timeout

**説明**

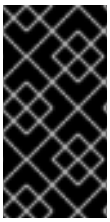
実行中の OSD 操作がタイムアウトするまでの時間 (秒)

**型**

Integer

**デフォルト**

**0**



**重要**

クライアントが結果に対応できない限り、**osd op timeout** オプションを設定しないでください。例えば、仮想マシン上で動作するクライアントにこのパラメーターを設定すると、仮想マシンがこのタイムアウトをハードウェアの故障と解釈するため、データの破損につながる可能性があります。

osd_max_backfills

**説明**

1 つの OSD に対して、または 1 つの OSD から許容されるバックフィル操作の最大数

**型**

64 ビット未署名の整数

**デフォルト**

**1**

osd_backfill_scan_min

**説明**

バックフィルスキャン 1 回あたりのオブジェクトの最小数

**型**

32 ビット整数

**デフォルト**

**64**

osd_backfill_scan_max

**説明**

バックフィルスキャン 1 回あたりのオブジェクトの最大数

**型**

32 ビット整数

**デフォルト**

**512**

osd_backfill_full_ratio

**説明**

Ceph OSD のフル比率がこの値以上の場合、バックフィル要求の受け入れを拒否します。

**型**

浮動小数点 (Float)

**デフォルト**

**0.85**

osd_backfill_retry_interval

**説明**

バックフィル要求を再試行するまでの待ち時間 (秒数)

**型**

Double

**デフォルト**

**30.000000**

osd_map_dedup

**説明**

OSD マップの重複の削除を有効にします。

**型**

Boolean

**デフォルト**

**true**

osd_map_cache_size

**説明**

OSD マップキャッシュのサイズ (メガバイト)

**型**

32 ビット整数

**デフォルト**

**50**

osd_map_cache_bl_size

**説明**

OSD デーモンのメモリー内 OSD マップキャッシュのサイズ

**型**

32 ビット整数

**デフォルト**

**50**

osd_map_cache_bl_inc_size

**説明**

OSD デーモンのメモリー内 OSD マップキャッシュの増分サイズ

**型**

32 ビット整数

**デフォルト**

**100**

osd_map_message_max

**説明**

MOSDMap メッセージごとに許容される最大のマップエントリー数

**型**

32 ビット整数

**デフォルト**

**40**

osd_snap_trim_thread_timeout

**説明**

スナップトリムスレッドがタイムアウトするまでの最大時間 (秒単位)

**型**

32 ビット整数

**デフォルト**

**60*60*1**

osd_pg_max_concurrent_snap_trims

**説明**

PG ごとの並列スナップトリムの最大数。PG ごとに何個のオブジェクトを一度にトリミングするかを制御します。

**型**

32 ビット整数

**デフォルト**

**2**

osd_snap_trim_sleep

**説明**

PG が発行する各トリム操作の間にスリープを挿入します。

**型**

32 ビット整数

**デフォルト**

**0**

osd_snap_trim_sleep_hdd

**説明**

HDD の次のスナップショットトリミングまでのスリープ時間 (秒)。

**型**

浮動小数点 (Float)

**デフォルト**

**5.0**

osd_snap_trim_sleep_ssd

**説明**

NVMe を含む SSD OSD の次のスナップショットトリミング操作までのスリープ時間 (秒単位)。

**型**

浮動小数点 (Float)

**デフォルト**

**0.0**

osd_snap_trim_sleep_hybrid

**説明**

OSD データが HDD 上にあり、OSD ジャーナルまたは WAL および DB が SSD 上にある場合の、次のスナップショットトリミング操作までのスリープ時間 (秒単位)。

**型**

浮動小数点 (Float)

**デフォルト**

**2.0**

osd_max_trimming_pgs

**説明**

トリミング PG の最大数

**型**

32 ビット整数

**デフォルト**

**2**

osd_backlog_thread_timeout

**説明**

バックログスレッドがタイムアウトするまでの最大時間 (秒単位)

**型**

32 ビット整数

**デフォルト**

**60*60*1**

osd_default_notify_timeout

**説明**

OSD デフォルト通知のタイムアウト (単位: 秒)

**型**

32 ビット符号なし整数

**デフォルト**

**30**

osd_check_for_log_corruption

**説明**

ログファイルが破損していないか確認します。計算量が多くなる可能性があります。

**型**

Boolean

**デフォルト**

**false**

osd_remove_thread_timeout

**説明**

OSD 削除スレッドがタイムアウトするまでの最大時間 (秒単位)

**型**

32 ビット整数

**デフォルト**

**60*60**

osd_command_thread_timeout

**説明**

コマンドスレッドがタイムアウトするまでの最大時間 (秒単位)

**型**

32 ビット整数

**デフォルト**

**10*60**

osd_command_max_records

**説明**

失ったオブジェクトを返す際の数を制限します。

**型**

32 ビット整数

**デフォルト**

**256**

osd_auto_upgrade_tmap

**説明**

古いオブジェクトの **omap** に **tmap** を使用します。

**型**

Boolean

**デフォルト**

**true**

osd_tmapput_sets_users_tmap

**説明**

デバッグにだけ **tmap** を使用します。

**型**

Boolean

**デフォルト**

**false**

osd_preserve_trimmed_log

**説明**

トリミングされたログファイルは保持されますが、より多くのディスク容量を使用します。

**型**

Boolean

**デフォルト**

**false**

osd_recovery_delay_start

**説明**

ピアリングが完了すると、Ceph は指定された秒数だけ遅延してからオブジェクトの回復を開始します。

**型**

浮動小数点 (Float)

**デフォルト**

**0**

osd_recovery_max_active

**説明**

OSD ごとに一度のアクティブな復旧要求の数。リクエストが増えれば復旧も早くなりますが、その分クラスターへの負荷も大きくなります。

**型**

32 ビット整数

**デフォルト**

**0**

osd_recovery_max_active_hdd

**説明**

プライマリーデバイスが HDD の場合に、同時に存在できる Ceph OSD ごとのアクティブリカバリーリクエスト数。

**型**

Integer

**デフォルト**

**3**

osd_recovery_max_active_ssd

**説明**

プライマリーデバイスが OSD の場合に、同時に存在できる Ceph OSD ごとのアクティブリカバリーリクエスト数。

**型**

Integer

**デフォルト**

**10**

osd_recovery_sleep

**説明**

次のリカバリーまたはバックフィル操作までのスリープ時間 (秒)。この値を大きくすると、リカバリー操作が遅くなりますが、クライアント操作への影響は少なくなります。

**型**

浮動小数点 (Float)

**デフォルト**

**0.0**

osd_recovery_sleep_hdd

**説明**

HDD の次のリカバリーまたはバックフィル操作までのスリープ時間 (秒)。

**型**

浮動小数点 (Float)

**デフォルト**

**0.1**

osd_recovery_sleep_ssd

**説明**

SSD の次のリカバリーまたはバックフィル操作までのスリープ時間 (秒)。

**型**

浮動小数点 (Float)

**デフォルト**

**0.0**

osd_recovery_sleep_hybrid

**説明**

Ceph OSD データが HDD にあり、OSD ジャーナルまたは WAL と DB が SSD にある場合の、次のリカバリーまたはバックフィル操作までのスリープ時間 (秒)。

**型**

浮動小数点 (Float)

**デフォルト**

**0.025**

osd_recovery_max_chunk

**説明**

復元したデータチャンクをプッシュする際の最大サイズ

**型**

64 ビット整数未署名

**デフォルト**

**8388608**

osd_recovery_threads

**説明**

データを復元するためのスレッド数

**型**

32 ビット整数

**デフォルト**

**1**

osd_recovery_thread_timeout

**説明**

復元スレッドがタイムアウトするまでの最大時間 (秒単位)

**型**

32 ビット整数

**デフォルト**

**30**

osd_recover_clone_overlap

**説明**

復元時のクローンのオーバーラップを保持します。常に **true** に設定する必要があります。

**型**

Boolean

**デフォルト**

**true**

**rados_osd_op_timeout**

**説明**

RADOS 操作からのエラーを返す前に、RADOS が OSD からの応答を待つ時間 (秒数)。値が 0 の場合は制限がないことを意味します。

**型**

Double

**デフォルト**

0

# 付録G CEPH MONITOR と OSD の設定オプション

ハートビート設定を変更する際には、Ceph 設定ファイルの **[global]** セクションにその設定を含めます。

**mon_osd_min_up_ratio**

**説明**

Ceph が Ceph OSD デーモンを **down** とマークする前に **up** となる Ceph OSD デーモンの最小比率。

**型**

Double

**デフォルト**

**.3**

**mon_osd_min_in_ratio**

**説明**

Ceph が Ceph OSD デーモンを **out** とマークを付ける前に **in** となる Ceph OSD デーモンの最小比率。

**型**

Double

**デフォルト**

**0.750000**

**mon_osd_laggy_halflife**

**説明**

**laggy** 予測の秒数が減ります。

**型**

Integer

**デフォルト**

**60*60**

**mon_osd_laggy_weight**

**説明**

**laggy** 予測の減少時の新しいサンプルの重み。

**型**

Double

**デフォルト**

**0.3**

**mon_osd_laggy_max_interval**

**説明**

ラグ推定値の **laggy_interval** の最大値 (秒単位)。モニターは適応アプローチを使用して特定の OSD の **laggy_interval** を評価します。この値は、その OSD の猶予時間を算出するために使用されます。

**型**

Integer

**デフォルト**

**300**

mon_osd_adjust_heartbeat_grace

**説明**

**true** に設定すると、Ceph は **laggy** 推定値に基づいてスケーリングします。

**型**

Boolean

**デフォルト**

**true**

mon_osd_adjust_down_out_interval

**説明**

**true** に設定すると、Ceph は **laggy** 推定値に基づいてスケーリングされます。

**型**

Boolean

**デフォルト**

**true**

mon_osd_auto_mark_in

**説明**

Ceph は、Ceph OSD デーモンのブートを、Ceph Storage Cluster の **in** とマークします。

**型**

Boolean

**デフォルト**

**false**

mon_osd_auto_mark_auto_out_in

**説明**

Ceph は、Ceph Storage クラスターから自動的に **out** とマーク付けされた Ceph OSD デーモンの起動が、クラスター内 **in** にあるとマークされます。

**型**

Boolean

**デフォルト**

**true**

mon_osd_auto_mark_new_in

**説明**

Ceph は、新しい Ceph OSD デーモンのブートを Ceph Storage Cluster の **in** とマークします。

**型**

Boolean

**デフォルト**

**true**

mon_osd_down_out_interval

**説明**

Ceph が Ceph OSD デーモンを **down** および **out** マークした後に応答しない場合には、Ceph が待機する秒数。

**型**

32 ビット整数

**デフォルト**

**600**

mon_osd_downout_subtree_limit

**説明**

Ceph が自動的に **out** とマークアウトする最大の CRUSH ユニットタイプ。

**型**

String

**デフォルト**

**rack**

mon_osd_reporter_subtree_level

**説明**

この設定は、報告する OSD の親 CRUSH ユニットタイプを定義します。OSD は、応答しないピアを見つけた場合、モニターに障害レポートを送信します。モニターは報告された OSD の数を **down** とマークし、猶予期間後に **out** になる可能性があります。

**型**

String

**デフォルト**

**host**

mon_osd_report_timeout

**説明**

応答しない Ceph OSD デーモンが **down** するまでの猶予期間 (秒単位)。

**型**

32 ビット整数

**デフォルト**

**900**

mon_osd_min_down_reporters

**説明**

**down** な Ceph OSD デーモンの報告に必要な Ceph OSD デーモンの最小数。

**型**

32 ビット整数

**デフォルト**

**2**

osd_heartbeat_address

**説明**

ハートビート用の Ceph OSD デーモンのネットワークアドレス

**型**

アドレス

**デフォルト**

ホストアドレス

osd_heartbeat_interval

**説明**

Ceph OSD デーモンがピアに ping を実行する頻度 (秒単位)

**型**

32 ビット整数

**デフォルト**

**6**

osd_heartbeat_grace

**説明**

Ceph OSD デーモンに Ceph Storage Cluster が **down** とみなすハートビートが表示されなかった場合の経過時間。

**型**

32 ビット整数

**デフォルト**

**20**

osd_mon_heartbeat_interval

**説明**

Ceph OSD デーモンピアがない場合に、Ceph OSD デーモンが Ceph Monitor に ping を実行する頻度

**型**

32 ビット整数

**デフォルト**

**30**

osd_mon_report_interval_max

**説明**

Ceph OSD デーモンが Ceph Monitor に報告しなければならなくなるまでに待機できる最大時間 (秒)

**型**

32 ビット整数

**デフォルト**

**120**

osd_mon_report_interval_min

**説明**

Ceph OSD デーモンが起動またはその他の報告可能なイベントから Ceph Monitor に報告するまでに待機する最小秒数

**型**

32 ビット整数

**デフォルト**

**5**

**有効な範囲**

**osd mon report interval max** 未満である必要があります。

osd_mon_ack_timeout

**説明**

Ceph Monitor が統計情報の要求を確認するまでの待ち時間 (秒数)

**型**

32 ビット整数

**デフォルト**

**30**

# 付録H CEPH のスクラブオプション

Ceph は配置グループをスクラブすることでデータの整合性を確保します。以下は、スクラブ操作を増減するために調整できる Ceph スクラビングオプションです。

これらの設定オプションは、**ceph config set global CONFIGURATION_OPTION VALUE** コマンドを使用して設定できます。

mds_max_scrub_ops_in_progress

**説明**

並行して実行されるスクラブ操作の最大数。この値は **ceph config set mds_max_scrub_ops_in_progress VALUE** コマンドを使用して設定できます。

**型**

integer

**デフォルト**

5

osd_max_scrubs

**説明**

Ceph OSD Deamon ごとの同時スクラブ操作の最大数

**型**

integer

**デフォルト**

1

osd_scrub_begin_hour

**説明**

スクラブが開始される特定の時間。**osd_scrub_end_hour** とともに、スクラブが発生する時間枠を定義できます。**osd_scrub_begin_hour = 0** および **osd_scrub_end_hour = 0** を使用して、1日中スクラブできるようにします。

**型**

integer

**デフォルト**

0

**許容範囲**

**[0, 23]**

osd_scrub_end_hour

**説明**

スクラブが終了する特定の時間。**osd_scrub_begin_hour** とともに、スクラブが発生する時間枠を定義できます。**osd_scrub_begin_hour = 0** および **osd_scrub_end_hour = 0** を使用して、1日住スクラブできるようにします。

**型**

integer

**デフォルト**

0

**許容範囲**

**[0, 23]**

osd_scrub_begin_week_day

**説明**

スクラブが開始される特定の日。0 = 日曜日、1 は月曜日などosd_scrub_end_week_day とともに、スクラブが発生する時間枠を定義できます。**osd_scrub_begin_week_day = 0** および **osd_scrub_end_week_day = 0** を使用して、週全体のスクラブを許可します。

**型**

integer

**デフォルト**

0

**許容範囲**

**[0, 6]**

osd_scrub_end_week_day

**説明**

これは、スクラビングが終了する日を定義します。0 = 日曜日、1 は月曜日など**osd_scrub_begin_hour** とともに、スクラブが発生する時間枠を定義できます。**osd_scrub_begin_week_day = 0** および **osd_scrub_end_week_day = 0** を使用して、週全体のスクラブを許可します。

**型**

integer

**デフォルト**

0

**許容範囲**

**[0, 6]**

osd_scrub_during_recovery

**説明**

復元中のスクラブを許可します。これを **false** に設定すると、復元中のものがあれば、新しいスクラブとディープスクラブのスケジューリングが無効になります。すでに実行中のスクラブは継続されます。これは、ビジー状態のストレージクラスターの負荷を減らすのに役立ちます。

**型**

boolean

**デフォルト**

false

osd_scrub_load_threshold

**説明**

正規化された最大負荷。getloadavg ()/オンライン CPU の数で定義されるシステム負荷が、この定義された数よりも高い場合、スクラブは行われません。

**型**

float

**デフォルト**

0.5

**osd_scrub_min_interval**

**説明**

Ceph Storage クラスターの負荷が低い場合に Ceph OSD デーモンをスクラブする最小間隔 (秒)。

**型**

float

**デフォルト**

1 day

**osd_scrub_max_interval**

**説明**

クラスターの負荷に関係なく、Ceph OSD デーモンをスクラブする最大間隔 (秒単位)。

**型**

float

**デフォルト**

7 日

**osd_scrub_chunk_min**

**説明**

1 回の操作でスクラブするオブジェクトストアチャンクの最小数。Ceph は、スクラブ中に単一のチャンクへの書き込みをブロックします。

**type**

integer

**デフォルト**

5

**osd_scrub_chunk_max**

**説明**

1 回の操作でスクラブするオブジェクトストアチャンクの最大数。

**type**

integer

**デフォルト**

25

**osd_scrub_sleep**

**説明**

次のチャンクをスクラブする前にスリープ状態になる時間。この値を増やすと、スクラブの全体的な速度が遅くなるため、クライアント操作の影響は低くなります。

**type**

float

**デフォルト**

0.0

osd_scrub_extended_sleep

**説明**

スクラビング時間または秒のうち、スクラビング中に遅延を挿入する期間。

**type**

float

**デフォルト**

0.0

osd_scrub_backoff_ratio

**説明**

スクラブをスケジュールするためのバックオフ率。これは、スクラブをスケジュールしないダニの割合であり、66% は、3 つのダニのうち 1 つがスクラブをスケジュールすることを意味します。

**type**

float

**デフォルト**

0.66

osd_deep_scrub_interval

**説明**

すべてのデータを完全に読み取る **deep** スクラビングの間隔。**osd_scrub_load_threshold** はこの設定には影響しません。

**type**

float

**デフォルト**

7 日

osd_debug_deep_scrub_sleep

**説明**

ディープスクラブ IO 中に高価なスリープを注入して、プリエンプションを誘発しやすくします。

**type**

float

**デフォルト**

0

osd_scrub_interval_randomize_ratio

**説明**

配置グループの次のスクラブジョブをスケジュールする際に、**osd_scrub_min_interval** に無作為に遅延を追加します。遅延は **osd_scrub_min_interval** * **osd_scrub_interval_randomized_ratio** 未満のランダムな値です。デフォルト設定では、**1、1.5** * **osd_scrub_min_interval** の許容時間枠でスクラブが分散されます。

**type**

float

**デフォルト**

0.5

osd_deep_scrub_stride

**説明**

デープスクラブを実施する際の読み取りサイズ

**type**

size

**デフォルト**

512 KB

osd_scrub_auto_repair_num_errors

**説明**

この多くのエラーが見つかると、自動修復は発生しません。

**type**

integer

**デフォルト**

5

osd_scrub_auto_repair

**説明**

これを **true** に設定すると、スクラブまたはディープスクラブによってエラーが見つかった場合に配置グループ (PG) の自動修復が有効になります。ただし、**osd_scrub_auto_repair_num_errors** を超えるエラーが見つかった場合、修復は実行されません。

**type**

boolean

**デフォルト**

false

osd_scrub_max_preemptions

**説明**

スクラブを完了するためにクライアント IO をブロックする前に、クライアント操作によるディープスクラブをプリエンプトする必要がある最大回数を設定します。

**type**

integer

**デフォルト**

5

osd_deep_scrub_keys

**説明**

ディープスクラブ中に一度にオブジェクトから読み取るキーの数。

**type**

integer

**デフォルト**

1024

# 付録I BLUESTORE の設定オプション

デプロイメント時に設定可能な Ceph BlueStore の設定オプションを以下に示します。

**注記**

このリストは完全ではありません。

**rocksdb_cache_size**

**説明**

RocksDB キャッシュのサイズ (単位: MB)

**型**

32 ビット整数

**デフォルト**

**512**

**bluestore_throttle_bytes**

**説明**

ユーザーが入力または出力 (I/O) の送信に対してスロットリングを適用するまでに使用できる最大バイト数。

**型**

サイズ

**デフォルト**

64 MB

**bluestore_throttle_deferred_bytes**

**説明**

ユーザーが I/O 送信に対してスロットリングを適用するまでのデファードライトの最大バイト数。

**型**

サイズ

**デフォルト**

128 MB

**bluestore_throttle_cost_per_io**

**説明**

各 I/O のトランザクションコスト (バイト単位) に追加されるオーバーヘッド。

**型**

サイズ

**デフォルト**

0 B

**bluestore_throttle_cost_per_io_hdd**

**説明**

HDD のデフォルトの **bluestore_throttle_cost_per_io** 値。

**型**

符号なしの整数

**デフォルト**

**67 000**

bluestore_throttle_cost_per_io_ssd

**説明**

SSD のデフォルトの **bluestore_throttle_cost_per_io** 値。

**型**

符号なしの整数

**デフォルト**

**4 000**

bluestore_debug_enforce_settings

**説明**

**hdd** は、回転ドライブ上の BlueStore 向けの設定を強制します。**ssd** は、ソリッドドライブ上の BlueStore 向けの設定を強制します

**型**

**default**、**hdd**、**ssd**

**デフォルト**

**default**

**注記**

**bluestore_debug_enforce_settings** オプションを変更した後、OSD を再起動します。