Red Hat Ceph Storage 4

トラブルシューティングガイド

Red Hat Ceph Storage のトラブルシューティング

Last Updated: 2021-11-14

Red Hat Ceph Storage 4 トラブルシューティングガイド

Red Hat Ceph Storage のトラブルシューティング

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

法律上の通知

概要

本書では、Red Hat Ceph Storage で一般的な問題を解決する方法について説明します。 Red Hat では、コード、ドキュメント、Web プロパティーにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、弊社の CTO、Chris Wright のメッセージを参照してください。

目次

第1章 初期トラブルシューティング

本章には、以下の情報が含まれます。

- Ceph エラーのトラブルシューティングを開始する方法 (「問題の特定」)
- 最も一般的な **ceph health** エラーメッセージ (「Ceph の健常性の理解」)
- 最も一般的な Ceph ログのエラーメッセージ (「Ceph ログの理解」)

1.1. 前提条件

- Red Hat Ceph Storage クラスターが実行中である。

1.2. 問題の特定

Red Hat Ceph Storage クラスターで考えられる原因を特定するには、手順 セクションの質問に回答します。

前提条件

- 稼働中の Red Hat Ceph Storage クラスターがある。

手順

1. サポート対象外の設定を使用すると、特定の問題が発生する可能性があります。設定がサポートされていることを確認します。
2. Ceph コンポーネントが問題を引き起こすのかを把握しているか？
   a. いいえ。**『Red Hat Ceph Storage トラブルシューティングガイド』**の**の「Ceph Storage クラスターの健全性の診断」**の手順に従います。
   b. Ceph 監視**『Red Hat Ceph Storage トラブルシューティングガイド』**の**「Ceph モニターのトラブルシューティング」**セクションを参照してください。
   c. Ceph OSD**『Red Hat Ceph Storage トラブルシューティングガイド』**の**「Ceph OSD のトラブルシューティング」**セクションを参照してください。
   d. Ceph の配置グループ**『Red Hat Ceph Storage トラブルシューティングガイド』**の**「Ceph 配置グループのトラブルシューティング」**セクションを参照してください。
   e. マルチサイトの Ceph Object Gateway**『Red Hat Ceph Storage トラブルシューティングガイド』**の**「マルチサイトの Ceph Object Gateway のトラブルシューティング」**セクションを参照してください。

関連情報

- 詳細は、「Red Hat Ceph Storage でサポートされる構成」 を参照してください。

1.2.1. ストレージクラスターの健全性の診断

この手順では、Red Hat Ceph Storage クラスターの健全性を診断するための基本的な手順を説明します。

前提条件

- 稼働中の Red Hat Ceph Storage クラスターがある。

手順

1. ストレージクラスターの全体的なステータスを確認します。

```
[root@mon ~]# ceph health detail
```

コマンドが **HEALTH_WARN** または **HEALTH_ERR** を返す場合は、**「Ceph の正常性の理解」**を参照してください。

2. Ceph ログで**「Ceph ログの概要」**に記載されているエラーメッセージの有無を確認します。ログは、デフォルトで **/var/log/ceph/** ディレクトリーにあります。

3. ログに十分な情報が含まれていない場合は、デバッグレベルを上げて、失敗したアクションを再現してみてください。詳細は、**「ログの設定」**を参照してください。

1.3. CEPH の正常性の理解

ceph health コマンドは、Red Hat Ceph Storage クラスターのステータスについての情報を返します。

- **HEALTH_OK** はクラスターが正常であることを示します。
- **HEALTH_WARN** は警告を示します。場合によっては、Ceph のステータスは自動的に **HEALTH_OK** に戻ります。たとえば、Red Hat Ceph Storage クラスターがリバランスプロセスを終了する場合。ただし、クラスターが **HEALTH_WARN** の状態であればさらにトラブルシューティングを行うことを検討してください。
- **HEALTH_ERR** は、早急な対応が必要なより深刻な問題を示します。

ceph health detail および **ceph -s** コマンドを使用して、より詳細な出力を取得します。

関連情報

- **『Red Hat Ceph Storage トラブルシューティングガイド』**の**「Ceph Monitor エラーメッセージ」**の表を参照してください。
- **『Red Hat Ceph Storage トラブルシューティングガイド』**の**「Ceph OSD エラーメッセージ」**を参照してください。
- **『Red Hat Ceph Storage トラブルシューティングガイド』**の**「配置グループのエラーメッセージ」**の表を参照してください。

1.4. CEPH ログについて

デフォルトでは、Ceph はログを **/var/log/ceph/** ディレクトリーに保存します。

CLUSTER_NAME.log は、グローバルイベントを含むメインストレージクラスターのログファイルです。デフォルトでは、ログファイル名は **ceph.log** です。Ceph Monitorノードのみにメインストレージクラスターのログが含まれます。

Ceph OSD および Monitor の各ログファイルには、**CLUSTER_NAME-osd.NUMBER.log** と **CLUSTER_NAME-mon.HOSTNAME.log** という名前の独自のログファイルがあります。

Ceph サブシステムのデバッグレベルを上げると、Ceph はそれらのサブシステムにも新しいログファイルを生成します。

関連情報

- ログの詳細は、**『Red Hat Ceph Storage トラブルシューティングガイド』**の**「ログの設定」**を参照してください。
- **『Red Hat Ceph Storage トラブルシューティングガイド』**の Red Hat **「Ceph ログにおける一般的な Ceph Monitor エラーメッセージ」**を参照してください
- **『Red Hat Ceph Storage トラブルシューティングガイド』「Ceph ログにおける一般的な Ceph OSD エラーメッセージ」**を参照してください。

第2章 ロギングの設定

本章では、さまざまな Ceph サブシステムのロギングを設定する方法について説明します。

重要

ロギングはリソース集約型です。また、詳細ロギングは、比較的短い時間で大量のデータを生成できます。クラスターの特定のサブシステムで問題が発生した場合は、そのサブシステムのロギングのみを有効にします。詳細は、「Ceph サブシステム」 を参照してください。

さらに、ログファイルのローテーションを設定することも検討してください。詳しくは、「ログローテーションの頻度を上げる」 を参照してください。

発生した問題を解決したら、サブシステムのログとメモーリのレベルをデフォルトの値に変更します。すべての Ceph サブシステムのリストおよびそのデフォルト値については、付録A *Ceph サブシステムのデフォルトログレベルの値* を参照してください。

以下を行って Ceph ロギングを設定できます。

- ランタイム時に **ceph** コマンドを使用します。これは最も一般的な方法です。詳しくは、「実行時のロギング設定」 を参照してください。
- Ceph 設定ファイルの更新クラスターの起動時に問題が発生した場合は、このアプローチを使用します。詳しくは、「設定ファイルでのロギングの設定」 を参照してください。

2.1. 前提条件

- Red Hat Ceph Storage クラスターが実行中である。

2.2. CEPH サブシステム

本項では、Ceph サブシステムとそれらのログレベルについて説明します。

Ceph サブシステムおよびログレベルの理解

Ceph は複数のサブシステムで構成されます。

各サブシステムには、以下のログレベルがあります。

- デフォルトで **/var/log/ceph/** ディレクトリー (ログレベル) に保存されている出力ログ
- メモリーキャッシュ (メモリーレベル) に保存されるログ

通常、Ceph は以下でない限り、メモリーに保存されているログを出力ログに送信しません。

- 致命的なシグナルが発生した
- ソースコードの assert がトリガーされた
- ユーザーがリクエストした

これらのサブシステムごとに異なる値を設定できます。Ceph のロギングレベルは、**1** から **20** の範囲で動作します。**1** は簡潔で、**20** は詳細です。

ログレベルおよびメモリーレベルに単一の値を使用して、両方の値を同じ値に設定します。たとえば、**debug_osd = 5** の場合には、**ceph-osd** デーモンのデバッグレベルを **5** に設定します。

出力ログレベルとメモリーレベルで異なる値を使用するには、値をスラッシュ (/) で区切ります。たとえば、**debug_mon = 1/5** の場合は、**ceph-mon** デーモンのデバッグログレベルを **1** に設定し、そのメモリーログレベルを **5** に設定します。

表2.1 Ceph サブシステムとロギングのデフォルト値

サブシステム	ログレベル	メモリーレベル	詳細
asok	1	5	管理ソケット
auth	1	5	認証
client	0	5	クラスターに接続するために **librados** を使用するアプリケーションまたはライブラリー
bluestore	1	5	BlueStore OSD バックエンド
journal	1	5	OSD ジャーナル
mds	1	5	メタデータサーバー
monc	0	5	Monitor クライアントは、ほとんどの Ceph デーモンとモニター間の通信を処理します。
mon	1	5	モニター
ms	0	5	Ceph コンポーネント間のメッセージングシステム
osd	0	5	OSD デーモン
paxos	0	5	Monitor がコンセンサスを得るために使用するアルゴリズム
rados	0	5	Ceph のコアコンポーネントである、信頼できる Autonomic Distributed Object Store
rbd	0	5	Ceph ブロックデバイス
rgw	1	5	Ceph Object Gateway

ログ出力の例

以下の例は、Monitor および OSD の詳細度を上げた場合の、ログのメッセージタイプを示しています。

Monitor デバッグ設定

```
debug_ms = 5
debug_mon = 20
debug_paxos = 20
debug_auth = 20
```

Monitor デバッグ設定のログ出力の例

```
2016-02-12 12:37:04.278761 7f45a9afc700 10 mon.cephn2@0(leader).osd e322 e322: 2 osds: 2 up,
2 in
2016-02-12 12:37:04.278792 7f45a9afc700 10 mon.cephn2@0(leader).osd e322
min_last_epoch_clean 322
2016-02-12 12:37:04.278795 7f45a9afc700 10 mon.cephn2@0(leader).log v1010106 log
2016-02-12 12:37:04.278799 7f45a9afc700 10 mon.cephn2@0(leader).auth v2877 auth
2016-02-12 12:37:04.278811 7f45a9afc700 20 mon.cephn2@0(leader) e1 sync_trim_providers
2016-02-12 12:37:09.278914 7f45a9afc700 11 mon.cephn2@0(leader) e1 tick
2016-02-12 12:37:09.278949 7f45a9afc700 10 mon.cephn2@0(leader).pg v8126 v8126: 64 pgs: 64
active+clean; 60168 kB data, 172 MB used, 20285 MB / 20457 MB avail
2016-02-12 12:37:09.278975 7f45a9afc700 10 mon.cephn2@0(leader).paxosservice(pgmap
7511..8126) maybe_trim trim_to 7626 would only trim 115 < paxos_service_trim_min 250
2016-02-12 12:37:09.278982 7f45a9afc700 10 mon.cephn2@0(leader).osd e322 e322: 2 osds: 2 up,
2 in
2016-02-12 12:37:09.278989 7f45a9afc700  5 mon.cephn2@0(leader).paxos(paxos active c
1028850..1029466) is_readable = 1 - now=2016-02-12 12:37:09.278990 lease_expire=0.000000 has
v0 lc 1029466
....
2016-02-12 12:59:18.769963 7f45a92fb700  1 -- 192.168.0.112:6789/0 <== osd.1
192.168.0.114:6800/2801 5724 ==== pg_stats(0 pgs tid 3045 v 0) v1 ==== 124+0+0 (2380105412 0
0) 0x5d96300 con 0x4d5bf40
2016-02-12 12:59:18.770053 7f45a92fb700  1 -- 192.168.0.112:6789/0 --> 192.168.0.114:6800/2801
-- pg_stats_ack(0 pgs tid 3045) v1 -- ?+0 0x550ae00 con 0x4d5bf40
2016-02-12 12:59:32.916397 7f45a9afc700  0 mon.cephn2@0(leader).data_health(1) update_stats
avail 53% total 1951 MB, used 780 MB, avail 1053 MB
....
2016-02-12 13:01:05.256263 7f45a92fb700  1 -- 192.168.0.112:6789/0 --> 192.168.0.113:6800/2410
-- mon_subscribe_ack(300s) v1 -- ?+0 0x4f283c0 con 0x4d5b440
```

OSD デバッグ設定

```
debug_ms = 5
debug_osd = 20
```

OSD デバッグ設定のログ出力の例

```
2016-02-12 11:27:53.869151 7f5d55d84700  1 -- 192.168.17.3:0/2410 --> 192.168.17.4:6801/2801 --
osd_ping(ping e322 stamp 2016-02-12 11:27:53.869147) v2 -- ?+0 0x63baa00 con 0x578dee0
2016-02-12 11:27:53.869214 7f5d55d84700  1 -- 192.168.17.3:0/2410 --> 192.168.0.114:6801/2801
-- osd_ping(ping e322 stamp 2016-02-12 11:27:53.869147) v2 -- ?+0 0x638f200 con 0x578e040
2016-02-12 11:27:53.870215 7f5d6359f700  1 -- 192.168.17.3:0/2410 <== osd.1
192.168.0.114:6801/2801 109210 ==== osd_ping(ping_reply e322 stamp 2016-02-12
11:27:53.869147) v2 ==== 47+0+0 (261193640 0 0) 0x63c1a00 con 0x578e040
2016-02-12 11:27:53.870698 7f5d6359f700  1 -- 192.168.17.3:0/2410 <== osd.1
192.168.17.4:6801/2801 109210 ==== osd_ping(ping_reply e322 stamp 2016-02-12
```

```
11:27:53.869147) v2 ==== 47+0+0 (261193640 0 0) 0x6313200 con 0x578dee0
....
2016-02-12 11:28:10.432313 7f5d6e71f700  5 osd.0 322 tick
2016-02-12 11:28:10.432375 7f5d6e71f700 20 osd.0 322 scrub_random_backoff lost coin flip,
randomly backing off
2016-02-12 11:28:10.432381 7f5d6e71f700 10 osd.0 322 do_waiters -- start
2016-02-12 11:28:10.432383 7f5d6e71f700 10 osd.0 322 do_waiters -- finish
```

関連情報

- **ランタイム時のログ設定**
- **設定ファイルでのログの設定**

2.3. 実行時のロギング設定

システムの実行時に Ceph サブシステムのログを設定して、発生する可能性のある問題のトラブルシューティングに役立てることができます。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- Ceph デバッガーへのアクセス。

手順

1. ランタイム時に Ceph デバッグ出力である **dout()** をアクティベートするには、以下を実行します。

   ```
   ceph tell TYPE.ID injectargs --debug-SUBSYSTEM VALUE [--NAME VALUE]
   ```

2. 以下を置き換えます。
   - **TYPE** を、Ceph デーモンのタイプ (**osd**、**mon**、または **mds**) に置き換えます。
   - **ID** を、Ceph デーモンの特定の ID に。特定タイプのすべてのデーモンにランタイム設定を適用するには、* を使用します。
   - **SUBSYSTEM**を、特定のサブシステムに。
   - **VALUE** を、**1** から **20** までの数字。**1** は簡潔で、**20** は詳細です。
     たとえば、**osd.0** という名前の OSD サブシステムのログレベルを 0 に設定し、メモリーレベルを 5 に設定するには、以下を実行します。

     ```
     # ceph tell osd.0 injectargs --debug-osd 0/5
     ```

実行時に設定を表示するには、以下を実行します。

1. 実行中の Ceph デーモン (例: **ceph-osd** または **ceph-mon**) でホストにログインします。
2. 設定を表示します。

   ```
   ceph daemon NAME config show | less
   ```

例

```
# ceph daemon osd.0 config show | less
```

関連情報

- 詳細は、**「Ceph サブシステム」**を参照してください。
- 詳細は、**「設定ファイルの設定ログ」**を参照してください。
- Red Hat Ceph Storage 4 の **『構成ガイド』**の**「Ceph のデバッグおよびログ設定リファレンス」**の章。

2.4. 設定ファイルでのロギングの設定

Ceph サブシステムを設定して、情報、警告、およびエラーメッセージをログファイルに記録します。Ceph 設定ファイルでデバッグレベルを指定することができます (デフォルトでは **/etc/ceph/ceph.conf**)。

前提条件

- 稼働中の Red Hat Ceph Storage クラスターがある。

手順

1. 起動時に Ceph のデバッグ出力を有効にするには、システムの起動時に **dout()** のデバッグ設定を Ceph 設定ファイルに追加します。
   a. 各デーモンに共通するサブシステムの場合は、**[global]** セクションに設定を追加します。
   b. 特定のデーモンのサブシステムについては、**[mon]**、**[osd]**、**[mds]** などのデーモンセクションに設定を追加します。

   例

```
[global]
        debug_ms = 1/5

[mon]
        debug_mon = 20
        debug_paxos = 1/5
        debug_auth = 2

[osd]
        debug_osd = 1/5
        debug_monc = 5/20

[mds]
        debug_mds = 1
```

関連情報

- Ceph サブシステム

- ランタイム時のログ設定
- Red Hat Ceph Storage 4 の **『構成ガイド』**の「Ceph のデバッグおよびログ設定リファレンス」の章

2.5. ログローテーションの頻度を上げる

Ceph コンポーネントのデバッグレベルを上げると、大量のデータが生成される可能性があります。ディスクがほぼ満杯になると、**/etc/logrotate.d/ceph** にある Ceph ログローテーションファイルを変更することで、ログローテーションを迅速化することができます。Cron ジョブスケジューラーはこのファイルを使用してログローテーションをスケジュールします。

前提条件

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

手順

1. ログローテーションファイルで、ローテーション頻度の後にサイズの設定を追加します。

```
rotate 7
weekly
size SIZE
compress
sharedscripts
```

   たとえば、ログファイルが 500 MB に達した時点でローテーションを行います。

```
rotate 7
weekly
size 500 MB
compress
sharedscripts
size 500M
```

2. **crontab** エディターを開きます。

```
[root@mon ~]# crontab -e
```

3. エントリーを追加して、**/etc/logrotate.d/ceph** ファイルを確認します。たとえば、Cron に 30 分ごとに **/etc/logrotate.d/ceph** をチェックするように指示するには、以下を実行します。

```
30 * * * * /usr/sbin/logrotate /etc/logrotate.d/ceph >/dev/null 2>&1
```

関連情報

- Red Hat Enterprise Linux 7 の『システム管理者のガイド』の「cron を使用した繰り返しジョブのスケジュール設定」セクション

第3章 ネットワークの問題のトラブルシューティング

本章では、ネットワークおよび Network Time Protocol (NTP) に接続するトラブルシューティング手順を説明します。

3.1. 前提条件

- Red Hat Ceph Storage クラスターが実行中である。

3.2. 基本的なネットワークのトラブルシューティング

Red Hat Ceph Storage は、信頼できるネットワーク接続に大きく依存しています。Red Hat Ceph Storage ノードは、ネットワークを使用して相互に通信します。ネットワークの問題は、動作が不安定になったり、**down** していると誤って報告されたりするなど、Ceph OSD で多くの問題を引き起こす可能性があります。ネットワークの問題は、Ceph Monitor のクロックスキューエラーの原因にもなります。さらに、パケットロス、高レイテンシー、帯域幅の制限は、クラスターのパフォーマンスと安定性に影響を与えます。

前提条件

- ノードへのルートレベルのアクセス。

手順

1. **net-tools** および **telnet** パッケージをインストールすると、Ceph Storage クラスターで発生する可能性のあるネットワーク問題のトラブルシューティングに役立ちます。

   Red Hat Enterprise Linux 7

   ```
   [root@mon ~]# yum install net-tools
   [root@mon ~]# yum install telnet
   ```

   Red Hat Enterprise Linux 8

   ```
   [root@mon ~]# dnf install net-tools
   [root@mon ~]# dnf install telnet
   ```

2. Ceph 設定ファイルの **cluster_network** パラメーターおよび **public_network** パラメーターに正しい値が含まれることを確認します。

   例

   ```
   [root@mon ~]# cat /etc/ceph/ceph.conf | grep net
   cluster_network = 192.168.1.0/24
   public_network = 192.168.0.0/24
   ```

3. ネットワークインターフェースが起動していることを確認します。

   例

   ```
   [root@mon ~]# ip link list
   1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode
   ```

```
DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp22s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
mode DEFAULT group default qlen 1000
    link/ether 40:f2:e9:b8:a0:48 brd ff:ff:ff:ff:ff:ff
```

4. Ceph ノードは、短縮ホスト名を使用して相互に通信できることを確認します。ストレージクラスターの各ノードでこれを確認します。

   構文

   ```
   ping SHORT_HOST_NAME
   ```

   例

   ```
   [root@mon ~]# ping osd01
   ```

5. ファイアウォールを使用する場合、Ceph ノードが適切なポートで他のノードにアクセスできることを確認します。**firewall-cmd** ツールと **telnet** ツールは、ポートの状態を検証し、ポートが開いているかどうかを確認できます。

   構文

   ```
   firewall-cmd --info-zone=ZONE
   telnet IP_ADDRESS PORT
   ```

   例

   ```
   [root@mon ~]# firewall-cmd --info-zone=public
   public (active)
     target: default
     icmp-block-inversion: no
     interfaces: enp1s0
     sources: 192.168.0.0/24
     services: ceph ceph-mon cockpit dhcpv6-client ssh
     ports: 9100/tcp 8443/tcp 9283/tcp 3000/tcp 9092/tcp 9093/tcp 9094/tcp 9094/udp
     protocols:
     masquerade: no
     forward-ports:
     source-ports:
     icmp-blocks:
     rich rules:

   [root@mon ~]# telnet 192.168.0.22 9100
   ```

6. インターフェースカウンターにエラーがないことを確認します。ノード間のネットワーク接続で遅延が予想され、パケットロスがないことを確認します。

   a. **ethtool** コマンドの使用:

      構文

      ```
      ethtool -S INTERFACE
      ```

例

```
[root@mon ~]# ethtool -S enp22s0f0 | grep errors
NIC statistics:
     rx_fcs_errors: 0
     rx_align_errors: 0
     rx_frame_too_long_errors: 0
     rx_in_length_errors: 0
     rx_out_length_errors: 0
     tx_mac_errors: 0
     tx_carrier_sense_errors: 0
     tx_errors: 0
     rx_errors: 0
```

b. **ifconfig** コマンドの使用:

例

```
[root@mon ~]# ifconfig
enp22s0f0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
inet 10.8.222.13  netmask 255.255.254.0  broadcast 10.8.223.255
inet6 2620:52:0:8de:42f2:e9ff:feb8:a048  prefixlen 64  scopeid 0x0<global>
inet6 fe80::42f2:e9ff:feb8:a048  prefixlen 64  scopeid 0x20<link>
ether 40:f2:e9:b8:a0:48  txqueuelen 1000  (Ethernet)
RX packets 4219130  bytes 2704255777 (2.5 GiB)
RX errors 0  dropped 0  overruns 0  frame 0 (1)
TX packets 1418329  bytes 738664259 (704.4 MiB)
TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0 (2)
device interrupt 16
```

c. **netstat** コマンドの使用:

例

```
[root@mon ~]# netstat -ai
Kernel Interface table
Iface       MTU   RX-OK RX-ERR RX-DRP RX-OVR  TX-OK TX-ERR TX-DRP TX-OVR
Flg
docker0      1500      0      0      0 0           0      0      0      0 BMU
eno2         1500      0      0      0 0           0      0      0      0 BMU
eno3         1500      0      0      0 0           0      0      0      0 BMU
eno4         1500      0      0      0 0           0      0      0      0 BMU
enp0s20u13u5  1500  253277      0      0 0           0      0      0      0 BMRU
enp22s0f0     9000  234160      0      0 0      432326      0      0      0 BMRU (1)
lo           65536   10366      0      0 0       10366      0      0      0 LRU
```

7. パフォーマンスの問題では、レイテンシーの確認の他に、ストレージクラスターのすべてのノード間のネットワーク帯域幅を検証するため、**iperf3** ツールを使用します。**iperf3** ツールは、サーバーとクライアント間のシンプルなポイントツーポイントネットワーク帯域幅テストを実行します。

a. 帯域幅を確認する Red Hat Ceph Storage ノードに **iperf3** パッケージをインストールします。

Red Hat Enterprise Linux 7

```
[root@mon ~]# yum install iperf3
```

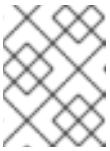
Red Hat Enterprise Linux 8

```
[root@mon ~]# dnf install iperf3
```

b. Red Hat Ceph Storageノードで、**iperf3** サーバを起動します。

例

```
[root@mon ~]# iperf3 -s
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```



注記

デフォルトのポートは 5201 ですが、**-P** コマンド引数を使用して設定できます。

c. 別の Red Hat Ceph Storage ノードで、**iperf3** クライアントを起動します。

例

```
[root@osd ~]# iperf3 -c mon
Connecting to host mon, port 5201
[  4] local xx.x.xxx.xx port 52270 connected to xx.x.xxx.xx port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec   114 MBytes   954 Mbits/sec    0    409 KBytes
[  4]   1.00-2.00   sec   113 MBytes   945 Mbits/sec    0    409 KBytes
[  4]   2.00-3.00   sec   112 MBytes   943 Mbits/sec    0    454 KBytes
[  4]   3.00-4.00   sec   112 MBytes   941 Mbits/sec    0    471 KBytes
[  4]   4.00-5.00   sec   112 MBytes   940 Mbits/sec    0    471 KBytes
[  4]   5.00-6.00   sec   113 MBytes   945 Mbits/sec    0    471 KBytes
[  4]   6.00-7.00   sec   112 MBytes   937 Mbits/sec    0    488 KBytes
[  4]   7.00-8.00   sec   113 MBytes   947 Mbits/sec    0    520 KBytes
[  4]   8.00-9.00   sec   112 MBytes   939 Mbits/sec    0    520 KBytes
[  4]   9.00-10.00  sec   112 MBytes   939 Mbits/sec    0    520 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec  1.10 GBytes   943 Mbits/sec    0             sender
[  4]   0.00-10.00  sec  1.10 GBytes   941 Mbits/sec                  receiver

iperf Done.
```

この出力では、Red Hat Ceph Storage ノード間のネットワーク帯域幅が 1.1Gbits/秒であることと、テスト中に再送 (**Retr**) がないことが示されています。

Red Hat は、ストレージクラスター内のすべてのノード間のネットワーク帯域幅を検証することを推奨します。

8. すべてのノードでネットワークの相互接続速度が同じであることを確認します。接続されているノードの速度が遅いと、アタッチされたノードの速度が遅くなることがあります。また、スイッチ間リンクが、アタッチされたノードの集約された帯域幅を処理できることを確認してください。

構文

```
ethtool INTERFACE
```

例

```
[root@mon ~]# ethtool enp22s0f0
Settings for enp22s0f0:
Supported ports: [ TP ]
Supported link modes:   10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
                        1000baseT/Half 1000baseT/Full
Supported pause frame use: No
Supports auto-negotiation: Yes
Supported FEC modes: Not reported
Advertised link modes:  10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
                        1000baseT/Half 1000baseT/Full
Advertised pause frame use: Symmetric
Advertised auto-negotiation: Yes
Advertised FEC modes: Not reported
Link partner advertised link modes:  10baseT/Half 10baseT/Full
                                     100baseT/Half 100baseT/Full
                                     1000baseT/Full
Link partner advertised pause frame use: Symmetric
Link partner advertised auto-negotiation: Yes
Link partner advertised FEC modes: Not reported
Speed: 1000Mb/s (1)
Duplex: Full (2)
Port: Twisted Pair
PHYAD: 1
Transceiver: internal
Auto-negotiation: on
MDI-X: off
Supports Wake-on: g
Wake-on: d
Current message level: 0x000000ff (255)
       drv probe link timer ifdown ifup rx_err tx_err
Link detected: yes (3)
```

関連情報

- 詳細は、カスタマーポータルの「Basic Network troubleshooting」を参照してください。
- 『Red Hat Ceph Storage Configuration Guide』の「Verifying and configuring the MTU value」のセクションを参照してください。
- 『Red Hat Ceph Storage Installation Guide』の「Configuring Firewall」セクションを参照してください。

- 詳細は、「"ethtool" コマンドは何ですか? このコマンドを使用して、ネットワークデバイスおよびインターフェイスの情報を取得する方法は?」を参照してください。
- 詳細は、カスタマーポータルの「RHEL ネットワークインターフェイスがパケットを破棄する」を参照してください。
- 詳細は、カスタマーポータルの「What are the performance benchmarking tools available for Red Hat Ceph Storage?」を参照してください。
- Red Hat Enterprise Linux 7 の「ネットワークガイド」
- 詳細は、カスタマーポータルのネットワーク問題のトラブルシューティングに関する ナレッジベースの記事およびソリューションを参照してください。

3.3. 基本的な NTP のトラブルシューティング

本セクションでは、基本的な NTP トラブルシューティング手順を説明します。

前提条件

- Ceph Monitor ノードへのルートレベルのアクセス。

手順

1. **ntpd** デーモンが Monitor ホストで実行されていることを確認します。

```
# systemctl status ntpd
```

2. **ntpd** が実行していない場合は、有効にして起動します。

```
# systemctl enable ntpd
# systemctl start ntpd
```

3. **ntpd** がクロックを正しく同期していることを確認します。

```
$ ntpq -p
```

関連情報

- 高度な NTP トラブルシューティング手順は、Red Hat カスタマーポータルの「NTP 問題のトラブルシューティング」を参照してください。
- 詳細は、『Red Hat Ceph Storage トラブルシューティングガイド』の「クロックスキュー」セクションを参照してください。

第4章 CEPH MONITOR のトラブルシューティング

本章では、Ceph Monitor に関連する最も一般的なエラーを修正する方法を説明します。

4.1. 前提条件

- ネットワーク接続の検証。

4.2. 最も一般的な CEPH MONITOR エラー

以下の表には、**ceph health detail** コマンドで返される、または Ceph ログに含まれる最も一般的なエラーメッセージを一覧表示しています。この表には、エラーを説明し、問題を修正するための特定の手順を示す、対応セクションへのリンクがあります。

4.2.1. 前提条件

- Red Hat Ceph Storage クラスターが実行中である。

4.2.2. Ceph Monitor エラーメッセージ

一般的な Ceph Monitor エラーメッセージの表およびその修正方法の表。

エラーメッセージ	参照
HEALTH_WARN	
mon.X is down (out of quorum)	Ceph Monitor がクォーラムを超えている
clock skew	クロックスキュー
store is getting too big!	Ceph Monitor ストアが大きすぎる

4.2.3. Ceph ログの共通の Ceph Monitor エラーメッセージ

Ceph ログにある一般的な Ceph Monitor エラーメッセージと、修正方法へのリンクが含まれる表。

エラーメッセージ	ログファイル	参照
clock skew	主なクラスターのログ	クロックスキュー
clocks not synchronized	主なクラスターのログ	クロックスキュー
Corruption: error in middle of record	監視ログ	Ceph Monitor がクォーラムを超えている Ceph Monitor ストアのリカバリー

エラーメッセージ	ログファイル	参照
Corruption: 1 missing files	監視ログ	Ceph Monitor がクォーラムを超えている Ceph Monitor ストアのリカバリー
Caught signal (Bus error)	監視ログ	Ceph Monitor がクォーラムを超えている

4.2.4. Ceph Monitor がクォーラムを超えている

1 つ以上の Ceph Monitor は **down** とマークされていますが、他の Ceph Monitor は引き続きクォーラムを形成することができます。さらに、**ceph health detail** コマンドは、以下のようなエラーメッセージを返します。

```
HEALTH_WARN 1 mons down, quorum 1,2 mon.b,mon.c
mon.a (rank 0) addr 127.0.0.1:6789/0 is down (out of quorum)
```

エラー内容:

Ceph では、さまざまな理由で Ceph Monitor が **down** とマークされます。

ceph-mon デーモンが実行していない場合は、ストアが破損しているか、その他のエラーによりデーモンを起動できません。また、**/var/** パーティションが満杯になっている可能性もあります。これにより、**ceph-mon** は **/var/lib/ceph/mon-SHORT_HOST_NAME/store.db** にデフォルトで配置されたストアに対する操作を実行できず、終了します。

ceph-mon デーモンが実行中で、Ceph Monitor がクォーラムを超えており、 **down** としてマークされている場合、問題の原因は Ceph Monitor 状態によって異なります。

- Ceph Monitor が予想よりも長く **プロービング** の場合は、他の Ceph Monitor を見つけることができません。この問題は、ネットワークの問題が原因で発生するか、または Ceph Monitor に古い Ceph Monitor マップ (**monmap**) があり、誤った IP アドレスで他の Ceph Monitor に到達しようとする可能性があります。**monmap** が最新の状態であれば、Ceph Monitor のクロックが同期されない可能性があります。

- Ceph Monitor が予想よりも長く **electing** 状態にある場合、Ceph Monitor のクロックが同期されていない可能性があります。

- Ceph Monitor の状態が **synchronizing** から **electing** に変更になり、元に戻る場合は、クラスターの状態が進行中です。これは、同期プロセスが処理できる以上の速さで新しいマップを生成していることを意味します。

- Ceph Monitor が自身を **leader** または **peon** としてマークしている場合、クォーラムにあると見なされますが、残りのクラスターはそうではないと確信しています。この問題は、クロック同期の失敗によって引き起こされる可能性があります。

この問題を解決するには、以下を行います。

1. **ceph-mon** デーモンが実行していることを確認します。そうでない場合は、起動します。

```
[root@mon ~]# systemctl status ceph-mon@HOST_NAME
[root@mon ~]# systemctl start ceph-mon@HOST_NAME
```

HOST_NAME を、デーモンが実行されているホストの短縮名に置き換えます。不明な場合は **hostname -s** コマンドを使用します。

2. **ceph-mon** を起動できない場合は、「**ceph-mon デーモンが起動できない**」の手順を行ってください。

3. **ceph-mon** デーモンを起動できるものの、**down** とマークされている場合は、**ceph-mon デーモンが実行しているが、`down` としてマークされている** の手順に従います。

ceph-mon デーモンを起動できない

1. デフォルトでは **/var/log/ceph/ceph-mon.HOST_NAME.log** にある対応する Ceph Monitor ログを確認します。

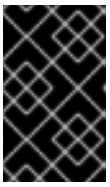
2. ログに以下のようなエラーメッセージが含まれる場合、Ceph Monitor のストアが破損している可能性があります。

```
Corruption: error in middle of record
Corruption: 1 missing files; e.g.: /var/lib/ceph/mon/mon.0/store.db/1234567.ldb
```

この問題を修正するには、Ceph Monitor を置き換えます。**「失敗したモニターの置き換え」**を参照してください。

3. ログに以下のようなエラーメッセージが含まれる場合は、**/var/** パーティションが満杯になっている可能性があります。**/var/** から不要なデータを削除します。

```
Caught signal (Bus error)
```



重要

Monitor ディレクトリーからデータを手動で削除しないでください。代わりに、**ceph-monstore-tool** を使用して圧縮します。詳細は、**「Ceph Monitor ストアの圧縮」**を参照してください。

4. 他のエラーメッセージが表示された場合は、サポートチケットを作成します。。詳細は、**「サービスについて Red Hat サポートへの問い合わせ」**を参照してください。

ceph-mon デーモンが実行しているが、down としてマークされている

1. クォーラムに達していない Ceph Monitor ホストから、**mon_status** コマンドを使用してその状態を確認します。

```
[root@mon ~]# ceph daemon ID mon_status
```

ID を、Ceph Monitor の ID に置き換えてください。以下に例を示します。

```
[root@mon ~]# ceph daemon mon.a mon_status
```

2. ステータスが **probing** の場合は、**mon_status** 出力内の他の Ceph Monitor の場所を確認します。

a. アドレスが正しくない場合は、Ceph Monitor の誤った Ceph Monitor マップ (**monmap**) が検出されます。この問題を修正するには、**「Ceph Monitor マップの注入」**を参照してください。

b. アドレスが正しい場合は、Ceph Monitor クロックが同期されていることを確認します。詳細は、**「クロックスキュー」**を参照してください。また、ネットワークの問題のトラブルシューティングについては、**「ネットワーク問題のトラブルシューティング」**を参照してください。

3. ステータスが **選択中** の場合は、Ceph Monitor クロックが同期されていることを確認します。詳細は、**「クロックスキュー」**を参照してください。

4. 状態が **選択中** から **同期中** に変わる場合は、サポートチケットを作成してください。詳細は、**「サービスについて Red Hat サポートへの問い合わせ」**を参照してください。

5. Ceph Monitor が **leader** または **peon** である場合は、Ceph Monitor クロックが同期されていることを確認します。詳細は、**「クロックスキュー」**を参照してください。クロックを同期させても問題が解決しない場合は、サポートチケットを作成します。。詳細は、**「サービスについて Red Hat サポートへの問い合わせ」**を参照してください。

関連情報

- **「Ceph Monitor ステータスの理解」**を参照してください。
- Red Hat Ceph Storage 4 の**『管理ガイド』**の**「インスタンスによる Ceph デーモンの起動、停止、再起動」**セクション
- Red Hat Ceph Storage 4 の**『管理ガイド』**の**「Ceph 管理ソケットの使用」**セクション

4.2.5. クロックスキュー

Ceph Monitor がクォーラムを超えており、**ceph health detail** コマンドの出力は、次のようなエラーメッセージが含まれています。

```
mon.a (rank 0) addr 127.0.0.1:6789/0 is down (out of quorum)
mon.a addr 127.0.0.1:6789/0 clock skew 0.08235s > max 0.05s (latency 0.0045s)
```

また、Ceph ログには以下のようなエラーメッセージが含まれます。

```
2015-06-04 07:28:32.035795 7f806062e700 0 log [WRN] : mon.a 127.0.0.1:6789/0 clock skew 0.14s
> max 0.05s
2015-06-04 04:31:25.773235 7f4997663700 0 log [WRN] : message from mon.1 was stamped
0.186257s in the future, clocks not synchronized
```

エラー内容:

clock skew エラーメッセージは、Ceph Monitor のクロックが同期されていないことを示します。Ceph Monitor は時間の精度に依存し、クロックが同期されていない場合に予測できない動作をするため、クロックの同期が重要になります。

mon_clock_drift_allowed パラメーターは、クロック間のどのような不一致を許容するかを決定します。デフォルトでは、このパラメーターは 0.05 秒に設定されています。

重要

以前のテストを行わずに **mon_clock_drift_allowed** のデフォルト値を変更しないでください。この値を変更すると、Ceph Monitor および Ceph Storage Cluster 全般の安定性に影響を与える可能性があります。

clock skew エラーの原因として、ネットワークの問題や Network Time Protocol (NTP) 同期の問題などがあります (設定されている場合)。また、仮想マシンにデプロイされた Ceph Monitor では、時間の同期が適切に機能しません。

この問題を解決するには、以下を行います。

1. ネットワークが正しく機能することを確認します。詳細は、**「ネットワーク問題のトラブルシューティング」**を参照してください。NTP を使用している場合は、特に NTP クライアントに関する問題をトラブルシューティングします。詳細は、**「基本的な NTP トラブルシューティング」**を参照してください。

2. リモートの NTP サーバーを使用する場合は、ネットワーク上に独自の NTP サーバーをデプロイすることを検討してください。詳細は、Red Hat Enterprise Linux 7 の**『システム管理者ガイド』**の**ntpd を使用した NTP 設定**」の章を参照してください。

3. NTP クライアントを使用していない場合は、NTP クライアントを設定してください。詳細は、Red Hat Ceph Storage 4 **『インストールガイド』**の**「Red Hat Ceph Storage 用ネットワークタイムプロトコルの設定」**セクションを参照してください。

4. Ceph Monitor をホストするために仮想マシンを使用する場合は、ベアメタルホストに移動します。Ceph Monitor をホストするために仮想マシンを使用することはサポートされていません。詳細は、Red Hat カスタマーポータルの**「Red Hat Ceph Storage でサポートされる構成」**の記事を参照してください。

注記

Ceph は 5 分ごとに時刻同期を評価するため、問題を修正してから **clock skew** メッセージを消去するまでに遅延が生じます。

関連情報

- **Ceph Monitor のステータスの理解**
- **Ceph Monitor がクォーラムを超えている**

4.2.6. Ceph Monitor ストアが大きすぎる

ceph health コマンドは、以下のようなエラーメッセージを返します。

```
mon.ceph1 store is getting too big! 48031 MB >= 15360 MB -- 62% avail
```

エラー内容:

Ceph Monitors ストアは、エントリーをキーと値のペアとして保存する LevelDB データベースです。データベースにはクラスターマップが含まれ、デフォルトでは **/var/lib/ceph/mon/CLUSTER_NAME-SHORT_HOST_NAME/store.db** に配置されます。

大規模な Monitor ストアのクエリーには時間がかかる場合があります。そのため、Ceph Monitor はクライアントクエリーへの応答が遅れることがあります。

また、**/var/** パーティションが満杯になると、Ceph Monitor はストアに対して書き込み操作を実行できず、終了します。この問題のトラブルシューティングについては、**「Ceph Monitor がクォーラムを超えている」**を参照してください。

この問題を解決するには、以下を行います。

1. データベースのサイズを確認します。

```
du -sch /var/lib/ceph/mon/CLUSTER_NAME-SHORT_HOST_NAME/store.db
```

クラスターの名前と、**ceph-mon** が実行しているホストの短縮ホスト名を指定します。

例

```
# du -sch /var/lib/ceph/mon/ceph-host1/store.db
47G     /var/lib/ceph/mon/ceph-ceph1/store.db/
47G     total
```

2. Ceph Monitor ストアを圧縮します。詳細は、**「Ceph Monitor ストアの圧縮」**を参照してください。

関連情報

- Ceph Monitor がクォーラムを超えている

4.2.7. Ceph Monitor のステータスの理解

mon_status コマンドは、以下のような Ceph Monitor についての情報を返します。

- 状態
- ランク
- 選出のエポック
- 監視マップ (**monmap**)

Ceph Monitor がクォーラムを形成できる場合は、**ceph** コマンドラインユーティリティーで **mon_status** を使用します。

Ceph Monitors がクォーラム (定足数) を形成できず、**ceph-mon** デーモンが実行中の場合は、管理ソケットを使用して **mon_status** を実行します。

mon_status の出力例

```
{
    "name": "mon.3",
    "rank": 2,
    "state": "peon",
    "election_epoch": 96,
    "quorum": [
        1,
        2
    ],
    "outside_quorum": [],
```

```
    "extra_probe_peers": [],
    "sync_provider": [],
    "monmap": {
        "epoch": 1,
        "fsid": "d5552d32-9d1d-436c-8db1-ab5fc2c63cd0",
        "modified": "0.000000",
        "created": "0.000000",
        "mons": [
            {
                "rank": 0,
                "name": "mon.1",
                "addr": "172.25.1.10:6789\/0"
            },
            {
                "rank": 1,
                "name": "mon.2",
                "addr": "172.25.1.12:6789\/0"
            },
            {
                "rank": 2,
                "name": "mon.3",
                "addr": "172.25.1.13:6789\/0"
            }
        ]
    }
}
```


Ceph Monitor の状態

Leader

選出フェーズ中に、Ceph Monitor はリーダーを選出します。リーダーは、最高ランクの Ceph Monitor で、つまり値が最も小さいランクです。上記の例では、リーダーは **mon.1** です。

Peon

Peons は、リーダーではないクォーラムの Ceph Monitor です。リーダーが失敗すると、一番ランクの高い peon が新しいリーダーになります。

Probing

Ceph Monitor が他の Ceph Monitor を検索する場合は、プロービング状態にあります。たとえば、Ceph Monitor を起動すると、Ceph Monitor マップ (**monmap**) に指定された十分な Ceph Monitor がクォーラムとなるまで **プローブ** が行われます。

Electing

Ceph Monitor がリーダーの選出中であれば、選出状態になります。通常、このステータスはすぐに変わります。

Synchronizing

Ceph Monitor が、他の Ceph Monitor と同期してクォーラムに参加する場合は、同期状態になります。Ceph Monitor ストアが小さいほど、同期処理は速くなります。したがって、ストアが大きい場合は、同期に時間がかかります。

関連情報

- 詳細は、Red Hat Ceph Storage 4 の **『管理ガイド』**の**「Ceph 管理ソケット」**を参照してください。

4.2.8. 関連情報

- 『**Red Hat Ceph Storage Troubleshooting Guide**』の 「Ceph Monitor エラーメッセージ」 を参照してください。
- 『**Red Hat Ceph Storage Troubleshooting Guide**』の 「Ceph ログの共通の Ceph Monitor エラーメッセージ」 を参照してください。

4.3. MONMAP の注入

Ceph Monitor に古いまたは破損した Ceph Monitor マップ (**mtte**) がある場合は、誤った IP アドレスで他の Ceph Monitor に到達しようとしているため、クォーラムに参加できません。

この問題の最も安全な方法は、他の Ceph Monitor から実際の Ceph Monitor マップを取得して注入することです。

注記

このアクションにより、Ceph Monitor によって保持される既存の Ceph Monitor マップが上書きされます。

この手順では、他の Ceph Monitor がクォーラムを形成できている場合、または少なくとも 1 つの Ceph Monitor が正しい Ceph Monitor マップを持っている場合に、Ceph Monitorマップを注入する方法を示します。すべての Ceph Monitor でストアが破損しているため、Ceph Monitor マップも破損している場合は、**「Ceph Monitor ストアの回復」**を参照してください。

前提条件

- Ceph Monitor マップへのアクセス。
- Ceph Monitor ノードへのルートレベルのアクセス。

手順

1. 残りの Ceph Monitor がクォーラムを形成できる場合には、**ceph mon getmap** コマンドを使用して Ceph Monitor マップを取得します。

```
[root@mon ~]# ceph mon getmap -o /tmp/monmap
```

2. 残りの Ceph Monitor がクォーラムを形成できず、正しい Ceph Monitor マップを持つ Ceph Monitor が少なくとも 1 つある場合は、その Ceph Monitor からコピーします。
   a. Ceph Monitor マップのコピー元の Ceph Monitor マップを停止します。

```
[root@mon ~]# systemctl stop ceph-mon@<host-name>
```

   たとえば、ホスト名 **host1** でホストで実行している Ceph Monitor を停止するには、以下のコマンドを実行します。

```
[root@mon ~]# systemctl stop ceph-mon@host1
```

   b. Ceph Monitor マップをコピーします。

```
[root@mon ~]# ceph-mon -i ID --extract-monmap /tmp/monmap
```

ID を、Ceph Monitor マップをコピーする Ceph Monitor の ID に置き換えます。

```
[root@mon ~]# ceph-mon -i mon.a  --extract-monmap /tmp/monmap
```

3. 破損したまたは古くなった Ceph Monitor マップを持つ Ceph Monitor を停止します。

```
[root@mon ~]# systemctl stop ceph-mon@HOST_NAME
```

たとえば、ホスト名が **host2** のホストで実行されている Ceph Monitor を停止するには、以下のコマンドを実行します。

```
[root@mon ~]# systemctl stop ceph-mon@host2
```

4. Ceph Monitor マップを注入します。

```
[root@mon ~]# ceph-mon -i ID --inject-monmap /tmp/monmap
```

ID を、破損した Ceph Monitor マップまたは古くなった Ceph Monitor マップに置き換えます。

```
[root@mon ~]# ceph-mon -i mon.c --inject-monmap /tmp/monmap
```

5. Ceph Monitor を起動します。

```
[root@mon ~]# systemctl start ceph-mon@host2
```

別の Ceph Monitor から Ceph Monitor マップをコピーした場合は、その Ceph Monitor も起動します。

```
[root@mon ~]# systemctl start ceph-mon@host1
```

関連情報

- Ceph Monitor がクォーラムを超えている
- Ceph Monitor ストアのリカバリー

4.4. 失敗したモニターの置き換え

Monitor に破損したストアがある場合、この問題を修正するには、Ansible 自動化アプリケーションを使用して Monitor を交換することをお勧めします。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- クォーラムを形成できる。
- Ceph Monitor ノードへの root レベルのアクセス。

手順

1. Monitor ホストから、デフォルトで **/var/lib/ceph/mon/CLUSTER_NAME-SHORT_HOST_NAME** にある Monitor ストアを削除します。

```
rm -rf /var/lib/ceph/mon/CLUSTER_NAME-SHORT_HOST_NAME
```

Monitor ホストの短縮ホスト名とクラスター名を指定します。たとえば、**host1** で実行している Monitor の Monitor ストアを、**remote** という名前のクラスターから削除するには、以下を実行します。

```
[root@mon ~]# rm -rf /var/lib/ceph/mon/remote-host1
```

2. Monitor マップ (**monmap**) から Monitor を削除します。

```
ceph mon remove SHORT_HOST_NAME --cluster CLUSTER_NAME
```

Monitor ホストの短縮ホスト名とクラスター名を指定します。たとえば、**host1** で実行しているモニターを **remote** というクラスターから削除するには、以下を実行します。

```
[root@mon ~]# ceph mon remove host1 --cluster remote
```

3. 基盤のファイルシステムまたは Monitor ホストのハードウェアに関連する問題をトラブルシューティングおよび修正します。

4. Ansible 管理ノードから、Playbook **ceph-ansible** を実行してモニターを再デプロイします。

```
$ /usr/share/ceph-ansible/ansible-playbook site.yml
```

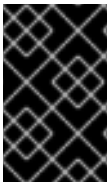
関連情報

- 詳細は、「Ceph Monitor がクォーラムを超えている」を参照してください。
- 『Red Hat Ceph Storage Operations Guide』の「Managing the storage cluster size」の章を参照してください。
- 『Red Hat Ceph Storage 4 インストールガイド』の「Red Hat Ceph Storage のデプロイ」の章を参照してください。

4.5. モニターストアの圧縮

モニターストアのサイズが大きくなってきたら、圧縮することができます。

- **ceph tell** コマンドを使用して、動的にこれを使用します。
- **ceph-mon** デーモンの起動時
- **ceph-mon** デーモンが稼働していない場合に **ceph-monstore-tool** を使用前述の方法が Monitor ストアを圧縮できない場合、または Monitor がクォーラムを超えていない状態で、そのログに **Caught signal (Bus error)** エラーメッセージが含まれる場合は、この方法を使用してください。



重要

クラスターが **active+clean** 状態ではない場合やリバランスプロセスでストアサイズの変更を監視します。このため、リバランスの完了時に Monitor ストアを圧縮します。また、配置グループが **active+clean** の状態であることを確認します。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- Ceph Monitor ノードへのルートレベルのアクセス。

手順

1. **ceph-mon** デーモンの実行中に Monitor ストアを圧縮するには、以下を実行します。

```
ceph tell mon.HOST_NAME compact
```

2. **HOST_NAME** を、**ceph-mon** を実行しているホストの短いホスト名に置き換えます。不明な場合は **hostname -s** コマンドを使用します。

```
# ceph tell mon.host1 compact
```

3. **[mon]** セクションの Ceph 設定に以下のパラメーターを追加します。

```
[mon]
mon_compact_on_start = true
```

4. **ceph-mon** デーモンを再起動します。

```
[root@mon ~]# systemctl restart ceph-mon@_HOST_NAME_
```

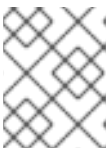
HOST_NAME を、デーモンが実行されているホストの短縮名に置き換えます。不明な場合は **hostname -s** コマンドを使用します。

```
[root@mon ~]# systemctl restart ceph-mon@host1
```

5. Monitor がクォーラムを形成することを確認します。

```
[root@mon ~]# ceph mon stat
```

6. 必要に応じて、他の Monitor でこの手順を繰り返します。



注記

開始する前に、**ceph-test** パッケージがインストールされていることを確認します。

7. 大型ストアを使用する **ceph-mon** デーモンが実行していないことを確認します。必要に応じてデーモンを停止します。

```
[root@mon ~]# systemctl status ceph-mon@HOST_NAME
[root@mon ~]# systemctl stop ceph-mon@HOST_NAME
```

HOST_NAME を、デーモンが実行されているホストの短縮名に置き換えます。不明な場合は **hostname -s** コマンドを使用します。

```
[root@mon ~]# systemctl status ceph-mon@host1
[root@mon ~]# systemctl stop ceph-mon@host1
```

8. Monitor ストアを圧縮します。

```
ceph-monstore-tool /var/lib/ceph/mon/mon.HOST_NAME compact
```

HOST_NAME は、Monitor ホストの短縮ホスト名に置き換えます。

```
# ceph-monstore-tool /var/lib/ceph/mon/mon.node1 compact
```

9. **ceph-mon** を再度起動します。

```
[root@mon ~]# systemctl start ceph-mon@HOST_NAME
```

例

```
[root@mon ~]# systemctl start ceph-mon@host1
```

関連情報

- Ceph Monitor ストアが大きすぎる
- Ceph Monitor がクォーラムを超えている

4.6. CEPH MANAGER のポート解放

ceph-mgr デーモンは、**ceph-osd** デーモンと同じ範囲のポート範囲の OSD から配置グループ情報を受け取ります。これらのポートが開かない場合、クラスターは **HEALTH_OK** から **HEALTH_WARN** に展開し、PG が不明なパーセンテージで PG が **unknown** なことを示します。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- Ceph マネージャーへのルートレベルのアクセス。

手順

1. この状況を解決するには、**ceph-mgr** デーモンを実行している各ホストでポート **6800:7300** を開きます。

例

```
[root@ceph-mgr] # firewall-cmd --add-port 6800:7300/tcp
[root@ceph-mgr] # firewall-cmd --add-port 6800:7300/tcp --permanent
```

1. **ceph-mgr** デーモンを再起動します。

4.7. CEPH MONITOR ストアのリカバリー

Ceph Monitor は、クラスターマップを LevelDB などのキーバリューストアに保存します。Monitor 上でストアが破損した場合、Monitor は異常終了し、再起動できなくなります。Ceph ログには以下のエラーが含まれる場合があります。

```
Corruption: error in middle of record
Corruption: 1 missing files; e.g.: /var/lib/ceph/mon/mon.0/store.db/1234567.ldb
```

実稼働の Red Hat Ceph Storage クラスターは少なくとも 3 つの Ceph Monitor を使用しており、1 つが故障しても別のものと交換できます。ただし、特定の状況では、すべての Ceph Monitor のストアが破損する可能性があります。たとえば、Ceph Monitor ノードのディスクやファイルシステムの設定が正しくない場合、停電によって基礎となるファイルシステムが破損する可能性があります。

すべての Ceph Monitor で破損がある場合には、**ceph-monstore-tool** および **ceph-objectstore-tool** と呼ばれるユーティリティーを使用して、OSD ノードに保管された情報で復元することができます。

重要

これらの手順は、以下の情報を復元できません。

- Metadata Daemon Server (MDS)キーリングおよびマップ
- 配置グループの設定:
  - **ceph pg set_full_ratio** コマンドを使用して設定する **full ratio**
  - **ceph pg set_nearfull_ratio** コマンドを使用して設定するほぼ **nearfull ratio**

重要

古いバックアップから Ceph Monitor ストアを復元しないでください。以下の手順に従って、現在のクラスター状態から Ceph Monitor ストアを再構築し、そこから復元します。

4.7.1. BlueStore の使用時の Ceph Monitor ストアのリカバリー

Ceph Monitor ストアがすべての Ceph Monitor で破損し、BlueStore バックエンドを使用する場合には、以下の手順に従います。

コンテナー化環境でこの方法を使用する場合、Ceph リポジトリーをアタッチし、最初にコンテナー化されていない Ceph Monitor に復元する必要があります。

警告

この手順では、データが失われる可能性があります。この手順で不明な点がある場合は、Red Hat テクニカルサポートに連絡して、リカバリープロセスの支援を受けてください。

前提条件

- **ベアメタルデプロイメント**
  - **rsync** パッケージおよび **ceph-test** パッケージがインストールされています。
- **コンテナーデプロイメント**
  - すべての OSD コンテナーが停止します。
  - ロールに基づいて Ceph ノードで Ceph リポジトリーを有効にします。
  - **ceph-test** パッケージおよび **rsync** パッケージが OSD および Monitor ノードにインストールされている。
  - **ceph-mon** パッケージが Monitor ノードにインストールされている。
  - **ceph-osd** パッケージが OSD ノードにインストールされている。

手順

1. **コンテナー** で Ceph を使用する場合は、Ceph データを含むすべてのディスクを一時的な場所にマウントします。すべての OSD ノードに対してこの手順を繰り返します。

   a. データパーティションを一覧表示します。デバイスの設定に使用したユーティリティーに応じて、**ceph-volume** または **ceph-disk** を使用します。

   ```
   [root@osd ~]# ceph-volume lvm list
   ```

   または

   ```
   [root@osd ~]# ceph-disk list
   ```

   b. データパーティションを一時的な場所にマウントします。

   ```
   for i in {OSD_ID}; do mount -t tmpfs /var/lib/ceph/osd/ceph-$i; done
   ```

   OSD_ID を、OSD ノード上の Ceph OSD ID の数値のスペース区切りリストに置き換えます。

   c. SELinux コンテキストを復元します。

   ```
   for i in {OSD_ID}; do restorecon /var/lib/ceph/osd/ceph-$i; done
   ```

   OSD_ID を、OSD ノード上の Ceph OSD ID の数値のスペース区切りリストに置き換えます。

   d. 所有者とグループを **ceph:ceph** に変更します。

   ```
   for i in {OSD_ID}; do chown -R ceph:ceph /var/lib/ceph/osd/ceph-$i; done
   ```

   OSD_ID を、OSD ノード上の Ceph OSD ID の数値のスペース区切りリストに置き換えます。

重要

update-mon-db コマンドが Monitor データベースに追加の **db** ディレクトリーおよび **db.slow** ディレクトリーを使用するバグにより、このディレクトリーもコピーする必要があります。改善点を報告する場合は、以下のように行います。

1. コンテナー外部の一時的な場所を準備して、OSD データベースをマウントしてアクセスし、Ceph Monitor を復元するために必要な OSD マップを展開します。

```
ceph-bluestore-tool --cluster=ceph prime-osd-dir --dev OSD-DATA --path /var/lib/ceph/osd/ceph-OSD-ID
```

OSD-DATA は OSD データへのボリュームグループ (VG) または論理ボリューム (LV) パスに、**OSD-ID** は OSD の ID に置き換えます。

2. BlueStore データベースと **block.db** との間のシンボリックリンクを作成します。

```
ln -snf BLUESTORE DATABASE /var/lib/ceph/osd/ceph-OSD-ID/block.db
```

BLUESTORE-DATABASE を BlueStore データベースへのボリュームグループ (VG) または論理ボリューム (LV) パスに置き換え、**OSD-ID** を OSD の ID に置き換えます。

2. 破損したストアのある Ceph Monitor ノードから次のコマンドを使用します。すべてのノードのすべての OSD に対してこれを繰り返します。

   a. すべての OSD ノードからクラスターマップを収集します。

```
[root@ mon~]# cd /root/
[root@mon ~]# ms=/tmp/monstore/
[root@mon ~]# db=/root/db/
[root@mon ~]# db_slow=/root/db.slow/

[root@mon ~]# mkdir $ms
[root@mon ~]# for host in $osd_nodes; do
                echo "$host"
                rsync -avz $ms $host:$ms
                rsync -avz $db $host:$db
                rsync -avz $db_slow $host:$db_slow

                rm -rf $ms
                rm -rf $db
                rm -rf $db_slow

                sh -t $host <<EOF
                  for osd in /var/lib/ceph/osd/ceph-*; do
                    ceph-objectstore-tool --type bluestore --data-path \$osd --op update-mon-db
--mon-store-path $ms

                  done
                EOF
```

```
        rsync -avz $host:$ms $ms
        rsync -avz $host:$db $db
        rsync -avz $host:$db_slow $db_slow
    done
```

b. 適切なケーパビリティーを設定します。

```
[root@mon ~]# ceph-authtool /etc/ceph/ceph.client.admin.keyring -n mon. --cap mon
'allow *' --gen-key
[root@mon ~]# cat /etc/ceph/ceph.client.admin.keyring
  [mon.]
    key = AQCleqldWqm5IhAAgZQbEzoShkZV42RiQVffnA==
    caps mon = "allow *"
  [client.admin]
    key = AQCmAKld8J05KxAArOWeRAw63gAwwZO5o75ZNQ==
    auid = 0
    caps mds = "allow *"
    caps mgr = "allow *"
    caps mon = "allow *"
    caps osd = "allow *"
```

c. **db** ディレクトリーおよび **db.slow** ディレクトリーから、すべての **sst** ファイルを一時的な場所に移動します。

```
[root@mon ~]# mv /root/db/*.sst /root/db.slow/*.sst /tmp/monstore/store.db
```

d. 収集したマップから Monitor ストアを再構築します。

```
[root@mon ~]# ceph-monstore-tool /tmp/monstore rebuild -- --keyring
/etc/ceph/ceph.client.admin
```

注記

このコマンドを実行後に、OSD から抽出したキーリングと、**ceph-monstore-tool** コマンドラインで指定されたキーリングのみが Ceph の認証データベースにあります。クライアント、Ceph Manager、Ceph Object Gateway などの他のすべてのキーリングを再作成またはインポートし、それらのクライアントがクラスターにアクセスできるようにする必要があります。

e. 破損したストアをバックアップします。すべての Ceph Monitor ノードでこの手順を繰り返します。

```
mv /var/lib/ceph/mon/ceph-HOSTNAME/store.db
/var/lib/ceph/mon/ceph-HOSTNAME/store.db.corrupted
```

HOSTNAME は、Ceph Monitor ノードのホスト名に置き換えます。

f. 破損したストアを交換します。すべての Ceph Monitor ノードでこの手順を繰り返します。

```
scp -r /tmp/monstore/store.db HOSTNAME:/var/lib/ceph/mon/ceph-HOSTNAME/
```

HOSTNAME は、Monitor ノードのホスト名に置き換えます。

g. 新しいストアの所有者を変更します。すべての Ceph Monitor ノードでこの手順を繰り返します。

```
chown -R ceph:ceph /var/lib/ceph/mon/ceph-HOSTNAME/store.db
```

HOSTNAME は、Ceph Monitor ノードのホスト名に置き換えます。

3. **コンテナーで** Ceph を使用する場合は、すべてのノードで一時的にマウントされた OSD をアンマウントします。

```
[root@osd ~]# umount /var/lib/ceph/osd/ceph-*
```

4. すべての Ceph Monitor デーモンを起動します。

```
[root@mon ~]# systemctl start ceph-mon *
```

5. Monitor がクォーラムを形成できることを確認します。

- **ベアメタルデプロイメント**

```
[root@mon ~]# ceph -s
```

- **コンテナー**

```
[user@admin ~]$ docker exec ceph-mon-_HOSTNAME_ ceph -s
```

HOSTNAME は、Ceph Monitor ノードのホスト名に置き換えます。

6. Ceph Manager キーリングをインポートして、すべての Ceph Manager プロセスを起動します。

```
ceph auth import -i /etc/ceph/ceph.mgr.HOSTNAME.keyring
systemctl start ceph-mgr@HOSTNAME
```

HOSTNAME は、Ceph Manager ノードのホスト名に置き換えてください。

7. すべての OSD ノード全体ですべての OSD プロセスを起動します。

```
[root@osd ~]# systemctl start ceph-osd *
```

8. OSD がサービスに返されることを確認します。

- **ベアメタルデプロイメント**

```
[root@mon ~]# ceph -s
```

- **コンテナー**

```
[user@admin ~]$ docker exec ceph-mon-_HOSTNAME_ ceph -s
```

HOSTNAME は、Ceph Monitor ノードのホスト名に置き換えます。

関連情報

- Ceph ノードをコンテンツ配信ネットワーク (CDN) に登録する方法の詳細は、**『Red Hat Ceph Storage インストールガイド』**の**「Red Hat Ceph Storage ノードの CDN への登録およびサブスクリプションの割り当て」**セクションを参照してください。
- リポジトリーの有効化に関する詳細は、**『Red Hat Ceph Storage インストールガイド』**の**「Red Hat Ceph Storage リポジトリーの有効化」**セクションを参照してください。

4.8. 関連情報

- ネットワーク関連の問題については、**『Red Hat Ceph Storage Troubleshooting Guide』**の 3 章 *ネットワークの問題のトラブルシューティング*を参照してください。

第5章 CEPH OSD のトラブルシューティング

本章では、Ceph OSD に関連する最も一般的なエラーを修正する方法を説明します。

5.1. 前提条件

- ネットワーク接続を確認します。詳しくは、**「ネットワーク問題のトラブルシューティング」**を参照してください。
- **ceph health** コマンドを使用して、Monitors にクォーラムがあることを確認します。コマンドがヘルスステータス (**HEALTH_OK**、**HEALTH_WARN**、**HEALTH_ERR**) を返すと、モニターはクォーラムを形成できます。そうでない場合は、最初に Monitor の問題に対応します。詳しくは、**「Ceph Monitors のトラブルシューティング」**を参照してください。**ceph health** に関する詳細は、**「Ceph の健常性」**を参照してください。
- 必要に応じて、リバランスプロセスを停止して、時間とリソースを節約します。詳細は、**「リバランスの停止および開始」**を参照してください。

5.2. 最も一般的な CEPH OSD エラー

以下の表には、**ceph health detail** コマンドで返される、または Ceph ログに含まれる最も一般的なエラーメッセージを一覧表示しています。この表には、エラーを説明し、問題を修正するための特定の手順を示す、対応セクションへのリンクがあります。

5.2.1. 前提条件

- Ceph OSD ノードへのルートレベルのアクセス。

5.2.2. Ceph OSD のエラーメッセージ

一般的な Ceph OSD エラーメッセージの表およびその修正方法。

エラーメッセージ	参照
HEALTH_ERR	
full osds	完全な OSDS
HEALTH_WARN	
nearfull osds	Nearfull OSDS
osds are down	OSDS がダウンしている OSDS のフラップ
requests are blocked	低速な要求がブロックされている
slow requests	低速な要求がブロックされている

5.2.3. Ceph ログの共通の Ceph OSD エラーメッセージ

Ceph ログにある一般的な Ceph OSD エラーメッセージと、修正方法へのリンクが含まれる表。

エラーメッセージ	ログファイル	参照
heartbeat_check: no reply from osd.X	主なクラスターのログ	OSDS のフラップ
wrongly marked me down	主なクラスターのログ	OSDS のフラップ
osds have slow requests	主なクラスターのログ	低速な要求がブロックされている
FAILED assert(0 == "hit suicide timeout")	OSD ログ	Down OSD

5.2.4. Full OSD

ceph health detail コマンドは、以下のようなエラーメッセージを返します。

```
HEALTH_ERR 1 full osds
osd.3 is full at 95%
```

エラー内容:

Ceph は、クライアントが完全な OSD ノードで I/O 操作を実行しないようにし、データの損失を防ぎます。クラスターが **mon_osd_full_ratio** パラメーターで設定された容量に達すると、**HEALTH_ERR full osds** メッセージを返します。デフォルトでは、このパラメーターは **0.95** に設定されています。これはクラスター容量の 95% を意味します。

この問題を解決するには、以下を行います。

Raw ストレージのパーセント数 (**%RAW USED**) を決定します。

```
# ceph df
```

%RAW USED が 70-75% を超える場合は、以下を行うことができます。

- 不要なデータを削除します。これは、実稼働環境のダウンタイムを回避するための短期的なソリューションです。
- 新しい OSD ノードを追加してクラスターをスケーリングします。これは、Red Hat が推奨する長期的なソリューションです。

関連情報

- 『Red Hat Ceph Storage トラブルシューティングガイド』の「Nearfull OSDS」
- 詳細は、「フルストレージクラスターからデータの削除」を参照してください。

5.2.5. Nearfull OSD

ceph health detail コマンドは、以下のようなエラーメッセージを返します。

```
HEALTH_WARN 1 nearfull osds
osd.2 is near full at 85%
```

エラー内容:

クラスターが **mon osd nearfull ratio defaults** パラメーターで設定されている容量に到達すると、Ceph はほぼ **nearfull osds** メッセージを返します。デフォルトでは、このパラメーターは **0.85** に設定されています。これはクラスター容量の 85% を意味します。

Ceph は、可能な限り最適な方法で CRUSH 階層に基づいてデータを分散しますが、均等な分散を保証することはできません。**不均等なデータ分散**と **nearfull osds** メッセージの主な原因は次のとおりです。

- OSD がクラスターの OSD ノード間で分散されていない。つまり、一部の OSD ノードが他のノードよりも大幅に多くの OSD をホストしていたり、CRUSH マップの一部の OSD の重みがその容量に対して十分でない。
- 配置グループ (PG) 数が、OSD の数、ユースケース、OSD ごとのターゲット PG 数、および OSD 使用率に応じて適切でない。
- クラスターが不適切な CRUSH 設定を使用する。
- OSD のバックエンドストレージがほぼ満杯である。

この問題を解決するには、以下を行います。

1. PG 数が十分であることを確認し、必要に応じてこれを増やします。
2. クラスタのバージョンに最適な CRUSH tunable を使用していることを確認し、そうでない場合は調整します。
3. 使用率別に OSD の重みを変更します。
4. バランスの取れた分散を実現するために OSD 間で配置グループ (PG) の配置を最適化する Ceph Manager バランサーモジュールを有効にします。

   例

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

5. OSD によって使用されるディスクの残りの容量を確認します。
   a. OSD が一般的に使用する容量を表示します。

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

   b. 特定のノードで OSD が使用する容量を表示します。**nearful** OSD が含まれるノードから以下のコマンドを使用します。

      ```
      $ df
      ```

   c. 必要な場合は、新規 OSD ノードを追加します。

関連情報

- 完全な OSD
- 『Red Hat Ceph Storage Operations Guide』の「Using the Ceph Manager balancer module」を参照してください。
- Red Hat Ceph Storage 4 の『ストレージストラテジー』の「使用率による OSD の重みの設定」セクションを参照してください。
- 詳細は、Red Hat Ceph Storage 4 の『ストレージストラテジー』ガイドの「CRUSH の調整可能なパラメーター」のセクションおよび「How can I test the impact CRUSH map tunable modifications will have on my PG distribution across OSDs in Red Hat Ceph Storage?」を参照してください。
- 詳細は、「配置グループの増加」を参照してください。

5.2.6. Down OSD

ceph health コマンドは、以下のようなエラーを返します。

```
HEALTH_WARN 1/3 in osds are down
```

エラー内容:

サービスの失敗やその他の OSD との通信に問題があるため、**ceph-osd** プロセスの 1 つを利用することはできません。そのため、残りの **ceph-osd** デーモンはこの失敗をモニターに報告していました。

ceph-osd デーモンが実行していない場合は、基礎となる OSD ドライブまたはファイルシステムが破損しているか、またはキーリングが見つからないなどのその他のエラーにより、デーモンが起動しません。

ほとんどの場合、ネットワークの問題により、**ceph-osd** デーモンが実行中にも **down** とマークされている場合に状況が生じます。

この問題を解決するには、以下を行います。

1. **down** になっている OSD を特定します。

```
[root@mon ~]# ceph health detail
HEALTH_WARN 1/3 in osds are down
osd.0 is down since epoch 23, last address 192.168.106.220:6800/11080
```

2. **ceph-osd** デーモンの再起動を試行します。

```
[root@mon ~]# systemctl restart ceph-osd@OSD_NUMBER
```

OSD_NUMBER を、**down** している OSD の ID に置き換えます。以下に例を示します。

```
[root@mon ~]# systemctl restart ceph-osd@0
```

   a. **ceph-osd** を起動できない場合は、**「ceph-osd デーモンが起動しない」**の手順を行ってください。

b. **ceph-osd** デーモンを起動できるものの、**down** とマークされている場合には、「**ceph-osd デーモンが実行しているが、`down` としてマークされている**」の手順に従ってください。

ceph-osd デーモンを起動できない

1. 複数の OSD (通常は 13 以上) が含まれる場合には、デフォルトの最大スレッド数 (PID 数) が十分であることを確認します。詳細は、「**PID 数の増加**」を参照してください。

2. OSD データおよびジャーナルパーティションが正しくマウントされていることを確認します。**ceph-volume lvm list** コマンドを使用して、Ceph Storage Cluster に関連付けられたデバイスおよびボリュームを一覧表示してから、適切にマウントされているかどうかを確認することができます。詳細は、man ページの **mount(8)** を参照してください。

3. **ERROR: missing keyring, cannot use cephx for authentication** が返された場合、OSD にはキーリングがありません。

4. **ERROR: unable to open OSD superblock on /var/lib/ceph/osd/ceph-1** エラーメッセージが出力されると、**ceph-osd** デーモンは基礎となるファイルシステムを読み込むことができません。このエラーをトラブルシューティングおよび修正する方法については、以下の手順を参照してください。

注記

OSD ホストのブート時にこのエラーメッセージが返される場合は、サポートチケットを開きます。これは、Red Hat Bugzilla 1439210 で追跡される既知の問題を示すためです。

5. 対応するログファイルを確認して、障害の原因を特定します。デフォルトでは、Ceph はログファイルを **/var/log/ceph/** ディレクトリーに保存します。

   a. 以下の **EIO** エラーメッセージのような EIO エラーメッセージは、基礎となるディスクの失敗を示しています。

   ```
   FAILED assert(!m_filestore_fail_eio || r != -5)
   ```

   この問題を修正するには、基礎となる OSD ディスクを交換します。詳細は、「**OSD ドライブを置き換え**」を参照してください。

   b. ログに、以下のような他の **FAILED assert** エラーが含まれる場合は、サポートチケットを作成してください。詳細は、「**サービスについて Red Hat サポートへの問い合わせ**」を参照してください。

   ```
   FAILED assert(0 == "hit suicide timeout")
   ```

6. **dmesg** 出力で、基礎となるファイルシステムまたはディスクのエラーを確認します。

```
$ dmesg
```

   a. **error -5** エラーメッセージは、ベースとなる XFS ファイルシステムの破損を示しています。この問題を修正する方法は、Red Hat カスタマーポータルの「**xfs_log_force: error 5 returned は何を示していますか?**」を参照してください。

   ```
   xfs_log_force: error -5 returned
   ```

b. **dmesg** 出力に **SCSI error** エラーメッセージが含まれる場合は、Red Hat カスタマーポータルの SCSI Error Codes Solution Finder ソリューションを参照して、問題を修正する最適な方法を判断してください。

c. または、基礎となるファイルシステムを修正できない場合は、OSD ドライブを交換します。詳細は、**「OSD ドライブを置き換え」**を参照してください。

7. OSD が以下のようなセグメンテーション違反で失敗した場合には、必要な情報を収集してサポートチケットを作成します。詳細は、**「サービスについて Red Hat サポートへの問い合わせ」**を参照してください。

```
Caught signal (Segmentation fault)
```

ceph-osd が実行中だが、down とマークされている。

1. 対応するログファイルを確認して、障害の原因を特定します。デフォルトでは、Ceph はログファイルを **/var/log/ceph/** ディレクトリーに保存します。

a. ログに以下のようなエラーメッセージが含まれる場合は、**「OSD のフラッピング」**を参照してください。

```
wrongly marked me down
heartbeat_check: no reply from osd.2 since back
```

b. 他のエラーが表示される場合は、サポートチケットを作成します。詳細は、**「サービスについて Red Hat サポートへの問い合わせ」**を参照してください。

関連情報

- **OSDS のフラップ**
- **古い配置グループ**
- **『Red Hat Ceph Storage 管理ガイド』**の**「インスタンスによる Ceph デーモンの開始、停止、再起動」**セクションを参照してください。
- **『Red Hat Ceph Storage 管理ガイド』**の**「Ceph キーリングの管理」**セクションを参照してください。

5.2.7. OSDS のフラップ

ceph -w | grep osds コマンドは、OSD を **down** として繰り返し示し、短期間に再び **up** します。

```
# ceph -w | grep osds
2021-04-05 06:27:20.810535 mon.0 [INF] osdmap e609: 9 osds: 8 up, 9 in
2021-04-05 06:27:24.120611 mon.0 [INF] osdmap e611: 9 osds: 7 up, 9 in
2021-04-05 06:27:25.975622 mon.0 [INF] HEALTH_WARN; 118 pgs stale; 2/9 in osds are down
2021-04-05 06:27:27.489790 mon.0 [INF] osdmap e614: 9 osds: 6 up, 9 in
2021-04-05 06:27:36.540000 mon.0 [INF] osdmap e616: 9 osds: 7 up, 9 in
2021-04-05 06:27:39.681913 mon.0 [INF] osdmap e618: 9 osds: 8 up, 9 in
2021-04-05 06:27:43.269401 mon.0 [INF] osdmap e620: 9 osds: 9 up, 9 in
2021-04-05 06:27:54.884426 mon.0 [INF] osdmap e622: 9 osds: 8 up, 9 in
2021-04-05 06:27:57.398706 mon.0 [INF] osdmap e624: 9 osds: 7 up, 9 in
2021-04-05 06:27:59.669841 mon.0 [INF] osdmap e625: 9 osds: 6 up, 9 in
```

```
2021-04-05 06:28:07.043677 mon.0 [INF] osdmap e628: 9 osds: 7 up, 9 in
2021-04-05 06:28:10.512331 mon.0 [INF] osdmap e630: 9 osds: 8 up, 9 in
2021-04-05 06:28:12.670923 mon.0 [INF] osdmap e631: 9 osds: 9 up, 9 in
```

また、Ceph ログには以下のようなエラーメッセージが含まれます。

```
2021-07-25 03:44:06.510583 osd.50 127.0.0.1:6801/149046 18992 : cluster [WRN] map e600547
wrongly marked me down
```

```
2021-07-25 19:00:08.906864 7fa2a0033700 -1 osd.254 609110 heartbeat_check: no reply from
osd.2 since back 2021-07-25 19:00:07.444113 front 2021-07-25 18:59:48.311935 (cutoff 2021-07-25
18:59:48.906862)
```

エラー内容:

OSD のフラップの主な原因は以下のとおりです。

- スクラビングやリカバリーなどの一部のストレージクラスター操作は、大きなインデックスや大きな配置グループを持つオブジェクトに対してこれらの操作を実行する場合などで、時間が異常にかかります。通常、これらの操作が完了すると、OSD のフラップ問題が解決されます。
- 基礎となる物理ハードウェアに関する問題。この場合、**ceph health details** コマンドも **slow requests** エラーメッセージを返します。
- ネットワークに関する問題。

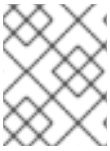
Ceph OSD は、ストレージクラスターのプライベートネットワークに障害が発生したり、クライアント向けのパブリックネットワークに大きな遅延が発生したりする状況を管理できません。

Ceph OSDは、**up** および **in** であることを示すために、プライベートネットワークを使用して、相互にハートビートパケットを送信します。プライベートストレージのクラスターネットワークが適切に機能しない場合、OSD はハートビートパケットを送受信できません。その結果、**up** であるとマークする一方で、Ceph Monitor に **down** であることを相互に報告します。

この動作は、Ceph 設定ファイルの以下のパラメーターの影響を受けます。

パラメーター	詳細	デフォルト値
osd_heartbeat_grace_time	OSD が **down** であると Ceph Monitor に報告する前に、ハートビートパケットが戻るまで OSD が待つ時間。	20 秒
mon_osd_min_down_reporters	Ceph Monitor が OSD を **down** とするまでに、他の OSD を **down** と報告する OSD の数。	2

この表は、デフォルト設定では、1つの OSD のみが最初の OSD が **down** していることについて 3 つの異なるレポートを作成した場合、Ceph Monitor が **down** としてマークすることを示しています。場合によっては、1つのホストにネットワークの問題が発生すると、クラスター全体で OSD のフラップが発生することもあります。これは、ホスト上に存在する OSD が、クラスター内の他の OSD を **down** として報告するためです。



注記

この OSD のフラップのシナリオには、OSD プロセスが起動された直後に強制終了される状況は含まれていません。

この問題を解決するには、以下を行います。

1. **ceph health detail** コマンドの出力を再度確認します。**slow requests** エラーメッセージが含まれる場合は、この問題のトラブルシューティング方法の詳細を参照してください。

```
# ceph health detail
HEALTH_WARN 30 requests are blocked > 32 sec; 3 osds have slow requests
30 ops are blocked > 268435 sec
1 ops are blocked > 268435 sec on osd.11
1 ops are blocked > 268435 sec on osd.18
28 ops are blocked > 268435 sec on osd.39
3 osds have slow requests
```

2. **down** としてマークされている OSD と、その OSD が置かれているノードを判別します。

```
# ceph osd tree | grep down
```

3. フラッピング OSD が含まれるノードで、ネットワークの問題をトラブルシューティングおよび修正します。詳細は、**「ネットワーク問題のトラブルシューティング」**を参照してください。

4. **noup** フラグおよび **nodown** フラグを設定して、OSD を **down** および **up** としてマークするのを停止するための一時的な強制モニターを実行できます。

```
# ceph osd set noup
# ceph osd set nodown
```

重要

noup フラグおよび **nodown** フラグを使用しても、問題の根本的な原因は修正されず、OSD がフラッピングしないようにします。サポートチケットを作成するには、**「Red Hat Suppor サービスの問い合わせ」**セクションを参照してください。

重要

OSD のフラッピングは、Ceph OSD ノードでの MTU 誤設定、ネットワークスイッチレベルでの MTU 誤設定、またはその両方が原因です。この問題を解決するには、計画的にダウンタイムを設けて、コアおよびアクセスネットワークスイッチを含むすべてのストレージクラスターノードで MTU を均一なサイズに設定します。**osd heartbeat min size** は調整しないでください。この設定を変更すると、ネットワーク内の問題が分からなくなり、実際のネットワークの不整合を解決できません。

関連情報

- 詳細は、**『Red Hat Ceph Storage インストールガイド』**の**「Red Hat Ceph Storage のネットワーク設定の確認」**セクションを参照してください。
- 詳細は、**『Red Hat Ceph Storage アーキテクチャーガイド』**の**「Ceph ハートビート」**セクションを参照してください。

- 『Red Hat Ceph Storage Troubleshooting Guide』の「Slow requests or requests are blocked」を参照してください。
- スクラビングプロセスをチューニングするには、「How to reduce scrub impact in a Red Hat Ceph Storage cluster?」を参照してください。

5.2.8. 遅いリクエストまたはブロックされるリクエスト

ceph-osd デーモンは要求に応答するのに時間がかかり、**ceph health detail** コマンドは以下のようなエラーメッセージを返します。

```
HEALTH_WARN 30 requests are blocked > 32 sec; 3 osds have slow requests
30 ops are blocked > 268435 sec
1 ops are blocked > 268435 sec on osd.11
1 ops are blocked > 268435 sec on osd.18
28 ops are blocked > 268435 sec on osd.39
3 osds have slow requests
```

また、Ceph ログには、以下のようなエラーメッセージが記録されます。

```
2015-08-24 13:18:10.024659 osd.1 127.0.0.1:6812/3032 9 : cluster [WRN] 6 slow requests, 6 included below; oldest blocked for > 61.758455 secs
```

```
2016-07-25 03:44:06.510583 osd.50 [WRN] slow request 30.005692 seconds old, received at {date-time}: osd_op(client.4240.0:8 benchmark_data_ceph-1_39426_object7 [write 0~4194304] 0.69848840) v4 currently waiting for subops from [610]
```

エラー内容:

要求が遅い OSD は、**osd_op_complaint_time** パラメーターで定義される時間内にキュー内の 1 秒あたりの I/O 操作 (IOPS) を処理しないすべての OSD です。デフォルトでは、このパラメーターは 30 秒に設定されています。

OSD のリクエストが遅い主な原因は次のとおりです。

- ディスクドライブ、ホスト、ラック、ネットワークスイッチなどの基礎となるハードウェアに関する問題
- ネットワークに関する問題。これらの問題は、通常、OSD のフラップに関連しています。詳細は、「OSD のフラッピング」を参照してください。
- システムの負荷

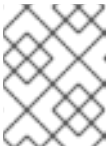
以下の表は、遅いリクエストのタイプを示しています。**dump_historic_ops** 管理ソケットコマンドを使用して、低速な要求のタイプを判断します。管理ソケットの詳細は、Red Hat Ceph Storage 4 の『管理ガイド』の「Ceph 管理ソケットの使用」セクションを参照してください。

遅いリクエストのタイプ	詳細
waiting for rw locks	OSD は、操作のために配置グループのロックの取得を待っています。

遅いリクエストのタイプ	詳細
waiting for subops	OSD は、レプリカ OSD が操作をジャーナルに適用するのを待っています。
no flag points reached	OSD は、主要な操作マイルストーンに到達しませんでした。
waiting for degraded object	OSD はまだオブジェクトを指定された回数複製していません。

この問題を解決するには、以下を行います。

1. 遅いリクエストまたはブロックされたリクエストのある OSD がディスクドライブ、ホスト、ラック、ネットワークスイッチなど、共通のハードウェアを共有しているかどうかを判断します。
2. OSD がディスクを共有する場合は、以下を実行します。
   a. **smartmontools** ユーティリティーを使用して、ディスクまたはログの状態をチェックして、ディスクのエラーを確認します。

   注記

   smartmontools ユーティリティーは、**smartmontools** パッケージに含まれています。

   b. **iostat** ユーティリティーを使用して OSD ディスクの I/O 待機レポート (**%iowai**) を取得し、ディスク負荷が大きいかどうかを判断します。

   注記

   iostat ユーティリティーは、**sysstat** パッケージに含まれています。

3. OSD が他のサービスとノードを共有している場合:
   a. RAM および CPU の使用率を確認します。
   b. **netstat** ユーティリティーを使用して、ネットワークインターフェースコントローラー (NIC) のネットワーク統計を確認し、ネットワークの問題のトラブルシューティングを行います。詳細は、**「ネットワーク問題のトラブルシューティング」**を参照してください。
4. OSD がラックを共有している場合は、ラックのネットワークスイッチを確認します。たとえば、ジャンボフレームを使用する場合は、パスの NIC にジャンボフレームが設定されていることを確認します。
5. リクエストが遅い OSD が共有している共通のハードウェアを特定できない場合や、ハードウェアやネットワークの問題をトラブルシューティングして解決できない場合は、サポートチケットを作成します。詳細は、**「サービスについて Red Hat サポートへの問い合わせ」**を参照してください。

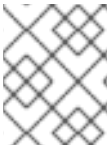
関連情報

- 詳細は、『Red Hat Ceph Storage 管理ガイド』の「Ceph 管理ソケットの使用」セクションを参照してください。

5.3. リバランスの停止および開始

OSD の失敗や停止時に、CRUSH アルゴリズムはリバランスプロセスを自動的に開始し、残りの OSD 間でデータを再分配します。

リバランスには時間とリソースがかかるため、トラブルシューティングや OSD のメンテナンス時にはリバランスの中止を検討してください。



注記

トラブルシューティングおよびメンテナンス時に、停止された OSD 内の配置グループは **degraded** します。

前提条件

- Ceph Monitor ノードへのルートレベルのアクセス。

手順

1. これを行うには、OSD を停止する前に **noout** フラグを設定します。

```
[root@mon ~]# ceph osd set noout
```

2. トラブルシューティングまたはメンテナンスが完了したら、**noout** フラグの設定を解除して、リバランスを開始します。

```
[root@mon ~]# ceph osd unset noout
```

関連情報

- 『Red Hat Ceph Storage アーキテクチャーガイド』の「リバランスおよび復元」セクションを参照してください。

5.4. OSD データパーティションのマウント

OSD データパーティションが正しくマウントされていない場合は、**ceph-osd** デーモンを起動することができません。パーティションが想定どおりにマウントされていないことが検出された場合は、本セクションの手順に従ってマウントします。

前提条件

- **ceph-osd** デーモンへのアクセス
- Ceph Monitor ノードへのルートレベルのアクセス。

手順

1. パーティションをマウントします。

```
[root@ceph-mon]# mount -o noatime PARTITION
/var/lib/ceph/osd/CLUSTER_NAME-OSD_NUMBER
```

PARTITION は、OSD データ専用の OSD ドライブのパーティションへのパスに置き換えます。クラスター名と OSD 番号を指定します。

例

```
[root@ceph-mon]# mount -o noatime /dev/sdd1 /var/lib/ceph/osd/ceph-0
```

2. 失敗した **ceph-osd** デーモンの起動を試みます。

```
[root@ceph-mon]# systemctl start ceph-osd@OSD_NUMBER
```

OSD_NUMBER を、OSD の ID に置き換えます。

例

```
[root@ceph-mon]# systemctl start ceph-osd@0
```

関連情報

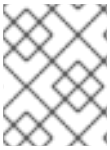
- 詳細は、**『Red Hat Ceph Storage トラブルシューティングガイド』**の「Down OSD」を参照してください。

5.5. OSD ドライブの交換

Ceph は耐障害性を確保できるように設計されているため、データを損失せずに動作が **degraded** の状態になっています。そのため、データストレージドライブに障害が発生しても、Ceph は動作します。障害が発生したドライブのコンテキストでは、パフォーマンスが **degraded** した状態は、他の OSD に保存されているデータの追加コピーが、クラスター内の他の OSD に自動的にバックフィルされることを意味します。ただし、このような場合は、障害の発生した OSD ドライブを交換し、手動で OSD を再作成します。

ドライブに障害が発生すると、Ceph は OSD を **down** として報告します。

```
HEALTH_WARN 1/3 in osds are down
osd.0 is down since epoch 23, last address 192.168.106.220:6800/11080
```



注記

Ceph は、ネットワークやパーミッションの問題により OSD を **down** とマークすることもできます。詳細は、「Down OSD」を参照してください。

最近のサーバーは、ホットスワップ対応のドライブを搭載しているのが一般的であり、ノードをダウンさせることなく、障害が発生したドライブを抜き取り、新しいドライブと交換することができます。手順全体には、以下のステップが含まれます。

1. Ceph クラスターから OSD を取り除きます。詳細は、**「Ceph クラスターからの OSD の削除」**の手順を参照してください。
2. ドライブを交換します。詳細は、**物理ドライブの置き換え** セクションを参照してください。

3. OSD をクラスターに追加します。詳細は、**「OSD の Ceph クラスターへの追加」**の手順を参照してください。

前提条件

- Ceph Monitor ノードへのルートレベルのアクセス。
- **down** になっている OSD を特定します。

```
[root@mon ~]# ceph osd tree | grep -i down
ID WEIGHT  TYPE NAME      UP/DOWN REWEIGHT PRIMARY-AFFINITY
 0 0.00999         osd.0     down  1.00000          1.00000
```

- OSD プロセスが停止していることを確認します。OSD ノードから以下のコマンドを使用します。

```
[root@mon ~]# systemctl status ceph-osd@_OSD_NUMBER_
```

- **OSD_NUMBER** を **down** とマークされた OSD の ID に置き換えます。以下に例を示します。

```
[root@mon ~]# systemctl status ceph-osd@osd.0
...
   Active: inactive (dead)
```

ceph-osd デーモンが実行しているかどうか。**down** とマークされているが対応する **ceph-osd** デーモンが実行している OSD のトラブルシューティングに関する詳細は、「Down OSDs」を参照してください。

手順: Ceph クラスターからの OSD の削除

1. OSD を **out** としてマークを付けます。

```
[root@mon ~]# ceph osd out osd.OSD_NUMBER
```

OSD_NUMBER を、**down** とマークされている OSD の ID に置き換えます。以下に例を示します。

```
[root@mon ~]# ceph osd out osd.0
marked out osd.0.
```

注記

OSD が **down** している場合、OSD からハートビートパケットを受信しない場合、Ceph は 600 秒後に自動的に **out** とマークします。この場合、障害が発生した OSD データのコピーを持つ他の OSD がバックフィルを開始し、クラスター内部に必要な数のコピーが存在するようにします。クラスターがバックフィル状態である間、クラスターの状態は **degraded** します。

2. 障害が発生した OSD がバックフィルされていることを確認します。出力には、以下のような情報が含まれます。

```
[root@mon ~]# ceph -w | grep backfill
2017-06-02 04:48:03.403872 mon.0 [INF] pgmap v10293282: 431 pgs: 1
```

```
active+undersized+degraded+remapped+backfilling, 28 active+undersized+degraded, 49
active+undersized+degraded+remapped+wait_backfill, 59 stale+active+clean, 294
active+clean; 72347 MB data, 101302 MB used, 1624 GB / 1722 GB avail; 227 kB/s rd, 1358
B/s wr, 12 op/s; 10626/35917 objects degraded (29.585%); 6757/35917 objects misplaced
(18.813%); 63500 kB/s, 15 objects/s recovering
2017-06-02 04:48:04.414397 mon.0 [INF] pgmap v10293283: 431 pgs: 2
active+undersized+degraded+remapped+backfilling, 75
active+undersized+degraded+remapped+wait_backfill, 59 stale+active+clean, 295
active+clean; 72347 MB data, 101398 MB used, 1623 GB / 1722 GB avail; 969 kB/s rd, 6778
B/s wr, 32 op/s; 10626/35917 objects degraded (29.585%); 10580/35917 objects misplaced
(29.457%); 125 MB/s, 31 objects/s recovering
2017-06-02 04:48:00.380063 osd.1 [INF] 0.6f starting backfill to osd.0 from (0'0,0'0] MAX to
2521'166639
2017-06-02 04:48:00.380139 osd.1 [INF] 0.48 starting backfill to osd.0 from (0'0,0'0] MAX to
2513'43079
2017-06-02 04:48:00.380260 osd.1 [INF] 0.d starting backfill to osd.0 from (0'0,0'0] MAX to
2513'136847
2017-06-02 04:48:00.380849 osd.1 [INF] 0.71 starting backfill to osd.0 from (0'0,0'0] MAX to
2331'28496
2017-06-02 04:48:00.381027 osd.1 [INF] 0.51 starting backfill to osd.0 from (0'0,0'0] MAX to
2513'87544
```

3. CRUSH マップから OSD を削除します。

```
[root@mon ~]# ceph osd crush remove osd.OSD_NUMBER
```

OSD_NUMBER を、**down** とマークされている OSD の ID に置き換えます。以下に例を示します。

```
[root@mon ~]# ceph osd crush remove osd.0
removed item id 0 name 'osd.0' from crush map
```

4. OSD に関連する認証キーを削除します。

```
[root@mon ~]# ceph auth del osd.OSD_NUMBER
```

OSD_NUMBER を、**down** とマークされている OSD の ID に置き換えます。以下に例を示します。

```
[root@mon ~]# ceph auth del osd.0
updated
```

5. Ceph Storage Cluster から OSD を削除します。

```
[root@mon ~]# ceph osd rm osd.OSD_NUMBER
```

OSD_NUMBER を、**down** とマークされている OSD の ID に置き換えます。以下に例を示します。

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

OSD を正常に削除した場合は、以下のコマンドの出力には表示されません。

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

6. 障害が発生したドライブをアンマウントします。

```
[root@mon ~]# umount /var/lib/ceph/osd/CLUSTER_NAME-OSD_NUMBER
```

クラスターの名前と OSD の ID を指定します。以下に例を示します。

```
[root@mon ~]# umount /var/lib/ceph/osd/ceph-0/
```

ドライブを正常にアンマウントした場合は、次のコマンドの出力には表示されません。

```
[root@mon ~]# df -h
```

手順: 物理ドライブの交換

物理ドライブの交換方法の詳細については、ハードウェアノードのマニュアルを参照してください。

a. ドライブがホットスワップ可能な場合は、故障したドライブを新しいものと交換します。

b. ドライブがホットスワップに対応しておらず、ノードに複数の OSD が含まれる場合は、ノード全体をシャットダウンして物理ドライブを交換する必要がある場合があります。クラスターのバックフィルを防ぐことを検討してください。詳細は、**『Red Hat Ceph Storage トラブルシューティングガイド』**の**「リバランスの停止および開始」**の章を参照してください。

c. ドライブが **/dev/** ディレクトリー配下に表示されたら、ドライブパスを書き留めます。

d. OSD を手動で追加する必要がある場合には、OSD ドライブを見つけ、ディスクをフォーマットします。

手順: OSD の Ceph クラスターへの追加

1. OSD を再度追加します。

   a. Ansible を使用してクラスターをデプロイしている場合は、Ceph 管理サーバーから Playbook **ceph-ansible** を再度実行します。

   ```
   [root@mon ~]# ansible-playbook /usr/share/ceph-ansible site.yml
   ```

   b. OSD を手動で追加した場合には、**Red Hat Ceph Storage 4 Operations Guide**の **「コマンドラインインターフェースで Ceph OSD の追加」**セクションを参照してください。

2. CRUSH 階層が正確であることを確認します。

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

3. CRUSH 階層の OSD の場所が適切でない場合は、OSD を希望の場所に移動します。

```
[root@mon ~]# ceph osd crush move BUCKET_TO_MOVE
BUCKET_TYPE=PARENT_BUCKET
```

たとえば、**sdd:row1** にあるバケットを root バケットに移動するには、以下を実行します。

```
[root@mon ~]# ceph osd crush move ssd:row1 root=ssd:root
```

関連情報

- 『Red Hat Ceph Storage トラブルシューティングガイド』の「Down OSD」セクションを参照してください。
- 『Red Hat Ceph Storage オペレーションガイド』の「ストレージクラスターサイズの管理」の章を参照してください。
- 『Red Hat Ceph Storage インストールガイド』

5.6. PID 数の増加

12 個以上の Ceph OSD が含まれるノードがある場合、特にリカバリー時にデフォルトの最大スレッド数 (PID数) では不十分になることがあります。これにより、一部の **ceph-osd** デーモンが終了して再起動に失敗する可能性があります。このような場合は、許容されるスレッドの最大数を増やします。

手順

一時的に数を増やすには、以下を実行します。

```
[root@mon ~]# sysctl -w kernel.pid.max=4194303
```

数値を永続的に増やすには、以下のように **/etc/sysctl.conf** ファイルを更新します。

```
kernel.pid.max = 4194303
```

5.7. 満杯のストレージクラスターからのデータの削除

Ceph は、**mon_osd_full_ratio** パラメーターで指定された容量に到達した OSD の I/O 操作を自動的に防ぎ、**full osds** エラーメッセージを返します。

この手順では、このエラーを修正するために不要なデータを削除する方法を説明します。

注記

mon_osd_full_ratio パラメーターは、クラスターの作成時に **full_ratio** パラメーターの値を設定します。その後は、**mon_osd_full_ratio** の値を変更することはできません。**full_ratio** 値を一時的に増やすには、代わりに **set-full-ratio** を増やします。

前提条件

- Ceph Monitor ノードへのルートレベルのアクセス。

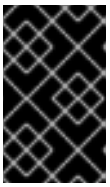
手順

1. **full_ratio** の現在の値を判別します。デフォルトでは **0.95** に設定されます。

```
[root@mon ~]# ceph osd dump | grep -i full
full_ratio 0.95
```

2. **set-full-ratio** の値を **0.97** に一時的に増やします。

```
[root@mon ~]# ceph osd set-full-ratio 0.97
```



重要

Red Hat は、**set-full-ratio** を 0.97 を超える値に設定しないことを強く推奨します。このパラメーターを高い値に設定すると、リカバリーが難しくなります。その結果、OSD を完全に復元できなくなる可能性があります。

3. パラメーターを **0.97** に正常に設定していることを確認します。

```
[root@mon ~]# ceph osd dump | grep -i full
full_ratio 0.97
```

4. クラスターの状態を監視します。

```
[root@mon ~]# ceph -w
```

クラスターの状態が **full** から **nearfull** に変わると、不要なデータが削除されます。

5. **full_ratio** の値を **0.95** に設定します。

```
[root@mon ~]# ceph osd set-full-ratio 0.95
```

6. パラメーターを **0.95** に正常に設定していることを確認します。

```
[root@mon ~]# ceph osd dump | grep -i full
full_ratio 0.95
```

関連情報

- **Red Hat Ceph Storage トラブルシューティングガイド**』の「フル OSD」セクション
- 『**Red Hat Ceph Storage トラブルシューティング**』の「Nearfull OSD」セクション

第6章 マルチサイト CEPH OBJECT GATEWAY のトラブルシューティング

本章では、マルチサイト Ceph Object Gateway の設定および操作状態に関連する最も一般的なエラーを修正する方法を説明します。

6.1. 前提条件

- 稼働中の Red Hat Ceph Storage クラスターがある。
- 稼働中の Ceph Object Gateway。

6.2. CEPH OBJECT GATEWAY のエラーコード定義

Ceph Object Gateway ログには、お使いの環境でのトラブルシューティングに役立つエラーおよび警告メッセージが含まれます。一般的なメッセージとその解決策を以下に示します。

一般的なエラーメッセージ

data_sync: ERROR: a sync operation returned error

これは、下位のバケット同期プロセスでエラーが返されたことを伝える上位のデータ同期プロセスです。このメッセージは詳細で、バケットの同期エラーがログで上に表示されます。

data sync: ERROR: failed to sync object: BUCKET_NAME:_OBJECT_NAME_

プロセスがリモートゲートウェイから HTTP 経由での必要なオブジェクトの取得に失敗したか、プロセスが RADOS へのオブジェクトの書き込みに失敗したかのいずれかであり、再試行されます。

data sync: ERROR: failure in sync, backing out (sync_status=2)

上記の条件の 1 つを反映した低レベルのメッセージ。同期前にデータが削除され、**-2 ENOENT** ステータスが表示されます。

data sync: ERROR: failure in sync, backing out (sync_status=-5)

上記の条件の 1 つを反映した低レベルのメッセージ。特に、そのオブジェクトを RADOS に書き込みに失敗し、**-5 EIO** が示されます。

ERROR: failed to fetch remote data log info: ret=11

これは、別のゲートウェイからのエラー状態を反映した **libcurl** の **EAGAIN** 汎用エラーコードです。デフォルトでは再度試行されます。

meta sync: ERROR: failed to read mdlog info with (2) No such file or directory

mdlog のシャードが作成されず、同期するものはありません。

エラーメッセージの同期

failed to sync object

プロセスがリモートゲートウェイから HTTP 経由でのオブジェクトの取得に失敗したか、そのオブジェクトの RADOS への書き込みに失敗したかのいずれかであり、再試行されます。

failed to sync bucket instance: (11) Resource temporarily unavailable

プライマリーゾーンとセカンダリーゾーン間の接続の問題。

failed to sync bucket instance: (125) Operation canceled

同じ RADOS オブジェクトへの書き込みの間に競合が発生します。

関連情報

- その他のサポートは、Red Hat サポート にお問い合わせください。

6.3. マルチサイト CEPH OBJECT GATEWAY の同期

マルチサイトの同期は、他のゾーンから変更ログを読み取ります。メタデータおよびデータログから同期の進捗の概要を取得するには、以下のコマンドを使用できます。

```
radosgw-admin sync status
```

このコマンドは、ソースゾーンの背後にあるログシャードがあれば、それを一覧表示します。

上記で実行した同期ステータスの結果がログシャードのレポートより遅れている場合は、shard-id を **X** に置き換えて次のコマンドを実行します。

```
radosgw-admin data sync status --shard-id=X
```

以下を置き換えます。

X はシャードの ID 番号に置き換えます。

例

```
[root@rgw ~]# radosgw-admin data sync status --shard-id=27
{
  "shard_id": 27,
  "marker": {
        "status": "incremental-sync",
        "marker": "1_1534494893.816775_131867195.1",
        "next_step_marker": "",
        "total_entries": 1,
        "pos": 0,
        "timestamp": "0.000000"
   },
   "pending_buckets": [],
   "recovering_buckets": [
        "pro-registry:4ed07bb2-a80b-4c69-aa15-fdc17ae6f5f2.314303.1:26"
   ]
}
```

出力には、次に同期されるバケットが表示され、あれば以前のエラーによりリトライされるバケットが表示されます。

X をバケット ID に置き換えて、次のコマンドを使用して個々のバケットのステータスを検査します。

```
radosgw-admin bucket sync status --bucket=X.
```

以下を置き換えます。

x はバケットの ID 番号に置き換えます。

その結果、ソースゾーンの背後にあるバケットインデックスログシャードが表示されます。

同期の一般的なエラーは **EBUSY** です。これは同期がすでに進行中であることを意味します。多くの場合は別のゲートウェイで行われます。同期エラーログに書き込まれたエラーを読み取ります。これは以下のコマンドで読み取りできます。

```
radosgw-admin sync error list
```

同期プロセスは成功するまで再試行されます。介入が必要なエラーが発生することもあります。

6.3.1. マルチサイトの Ceph Object Gateway データ同期のパフォーマンスカウンター

Ceph Object Gateway のマルチサイト設定では、データの同期を測定するために以下のパフォーマンスカウンターが使用できます。

- **poll_latency** は、リモートレプリケーションログに対する要求のレイテンシーを測定します。
- **fetch_bytes** は、データ同期によってフェッチされるオブジェクト数およびバイト数を測定します。

パフォーマンスカウンターの現在のメトリックデータを表示するには、**ceph daemon** コマンドを使用します。

```
ceph daemon /var/run/ceph/RGW.asok
```

例

```
{
    "data-sync-from-us-west": {
        "fetch bytes": {
            "avgcount": 54,
            "sum": 54526039885
        },
        "fetch not modified": 7,
        "fetch errors": 0,
        "poll latency": {
            "avgcount": 41,
            "sum": 2.533653367,
            "avgtime": 0.061796423
        },
        "poll errors": 0
    }
}
```

注記

デーモンを実行するノードから **ceph daemon** コマンドを実行する必要があります。

関連情報

- パフォーマンスカウンターの詳細は『Red Hat Ceph Storage 管理ガイド』の「パフォーマンスカウンター」の章を参照してください。

第7章 CEPH ISCSI ゲートウェイのトラブルシューティング

ストレージ管理者は、Ceph iSCSI ゲートウェイを使用する場合に発生する可能性のあるほとんどの一般的なエラーをトラブルシューティングすることができます。以下のような一般的なエラーが発生する可能性があります。

- iSCSI ログインの問題。
- さまざまな接続障害が発生する VMware ESXi。
- タイムアウトエラー。

7.1. 前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- 実行中の Ceph iSCSI ゲートウェイ。
- ネットワーク接続を確認します。

7.2. VMWARE ESXI でストレージ障害の原因となる切断された接続の情報収集

システムおよびディスク情報を収集すると、接続が切断され、ストレージ障害の原因となっている可能性がある iSCSI ターゲットを特定できます必要であれば、この情報を Red Hat のグローバルサポートサービスに提供して、Ceph iSCSI ゲートウェイの問題のトラブルシューティングに役立てることもできます。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- iSCSI ターゲットとなる実行中の Ceph iSCSI ゲートウェイ。
- iSCSI イニシエーターである実行中の VMware ESXi 環境。
- VMware ESXi ノードへの root レベルのアクセス。

手順

1. VWware ESXi ノードで、カーネルログを開きます。

```
[root@esx:~]# more /var/log/vmkernel.log
```

2. VMware ESXi カーネルログの以下のエラーメッセージから情報を収集します。

   例

```
2020-03-30T11:07:07.570Z cpu32:66506)iscsi_vmk:
iscsivmk_ConnRxNotifyFailure: Sess [ISID: 00023d000005 TARGET:
iqn.2017-12.com.redhat.iscsi-gw:ceph-igw TPGT: 3 TSIH: 0]
```

このメッセージから、**ISID** 番号、**TARGET** 名、および Target Portal Group Tag (**TPGT**) 番号をメモします。この例では、以下のようになります。

```
ISID: 00023d000005
TARGET: iqn.2017-12.com.redhat.iscsi-gw:ceph-igw
TPGT: 3
```

例

```
2020-03-30T11:07:07.570Z cpu32:66506)iscsi_vmk:
iscsivmk_ConnRxNotifyFailure: vmhba64:CH:4 T:0 CN:0: Connection rx
notifying failure: Failed to Receive. State=Bound
```

このメッセージから、アダプターチャネル (**CH**) 番号を書き留めます。この例では、以下のようになります。

```
vmhba64:CH:4 T:0
```

3. Ceph iSCSI ゲートウェイノードのリモートアドレスを検索するには、以下を実行します。

```
[root@esx:~]# esxcli iscsi session connection list
```

例

```
...
vmhba64,iqn.2017-12.com.redhat.iscsi-gw:ceph-igw,00023d000003,0
   Adapter: vmhba64
   Target: iqn.2017-12.com.redhat.iscsi-gw:ceph-igw (1)
   ISID: 00023d000003 (2)
   CID: 0
   DataDigest: NONE
   HeaderDigest: NONE
   IFMarker: false
   IFMarkerInterval: 0
   MaxRecvDataSegmentLength: 131072
   MaxTransmitDataSegmentLength: 262144
   OFMarker: false
   OFMarkerInterval: 0
   ConnectionAddress: 10.2.132.2
   RemoteAddress: 10.2.132.2 (3)
   LocalAddress: 10.2.128.77
   SessionCreateTime: 03/28/18 21:45:19
   ConnectionCreateTime: 03/28/18 21:45:19
   ConnectionStartTime: 03/28/18 21:45:19
   State: xpt_wait
...
```

コマンド出力から、以前に収集された **ISID** 値と **TARGET** の名前値を一致させ、**RemoteAddress** 値を書き留めます。この例では、以下のようになります。

```
Target: iqn.2017-12.com.redhat.iscsi-gw:ceph-igw
ISID: 00023d000003
RemoteAddress: 10.2.132.2
```

これで、Ceph iSCSI ゲートウェイノードからより多くの情報を収集し、問題のトラブルシューティングを行うことができます。

a. **RemoteAddress** の値に示される Ceph iSCSI ゲートウェイノードで **sosreport** を実行して、システム情報を収集します。

```
[root@igw ~]# sosreport
```

4. デッドステートになったディスクを検索するには、以下を行います。

```
[root@esx:~]# esxcli storage nmp device list
```

例

```
...
iqn.1998-01.com.vmware:d04-nmgjd-pa-zyc-sv039-rh2288h-xnh-732d78fd-
00023d000004,iqn.2017-12.com.redhat.iscsi-gw:ceph-igw,t,3-
naa.60014054a5d46697f85498e9a257567c
   Runtime Name: vmhba64:C4:T0:L4 (1)
   Device: naa.60014054a5d46697f85498e9a257567c (2)
   Device Display Name: LIO-ORG iSCSI Disk
(naa.60014054a5d46697f85498e9a257567c)
   Group State: dead (3)
   Array Priority: 0
   Storage Array Type Path Config:
{TPG_id=3,TPG_state=ANO,RTP_id=3,RTP_health=DOWN} (4)
   Path Selection Policy Path Config: {non-current path; rank: 0}
...
```

コマンド出力から、以前に収集された **CH** 番号および **TPGT** 番号が一致し、**Device** の値を書き留めます。この例では、以下のようになります。

```
vmhba64:C4:T0
Device: naa.60014054a5d46697f85498e9a257567c
TPG_id=3
```

デバイス名を使用すると、各 iSCSI ディスクの追加情報を **dead** 状態で収集できます。

a. iSCSI ディスクの詳細情報を収集します。

構文

```
esxcli storage nmp path list -d ISCSI_DISK_DEVICE >
/tmp/esxcli_storage_nmp_path_list.txt
esxcli storage core device list -d ISCSI_DISK_DEVICE >
/tmp/esxcli_storage_core_device_list.txt
```

例

```
[root@esx:~]# esxcli storage nmp path list -d naa.60014054a5d46697f85498e9a257567c
> /tmp/esxcli_storage_nmp_path_list.txt
[root@esx:~]# esxcli storage core device list -d
```

```
naa.60014054a5d46697f85498e9a257567c > /tmp/esxcli_storage_core_device_list.txt
```

5. VMware ESXi 環境に関する追加情報を収集します。

```
[root@esx:~]# esxcli storage vmfs extent list > /tmp/esxcli_storage_vmfs_extent_list.txt
[root@esx:~]# esxcli storage filesystem list > /tmp/esxcli_storage_filesystem_list.txt
[root@esx:~]# esxcli iscsi session list > /tmp/esxcli_iscsi_session_list.txt
[root@esx:~]# esxcli iscsi session connection list >
/tmp/esxcli_iscsi_session_connection_list.txt
```

6. iSCSI ログインの潜在的な問題の有無を確認します。
   - iSCSI ログインデータは送信されていませんか？
   - iSCSI ログインのタイムアウトが発生したか、ポータルグループが見つからないか？

関連情報

- Red Hat グローバルサポートサービス向けに **sosreport** を作成する方法 は Red Hat ナレッジベースソリューションを参照してください。
- Red Hat グローバルサポートサービスの ファイルのアップロード に関する Red Hat ナレッジベースソリューションを参照してください。
- カスタマーポータルで Red Hat サポートケース を作成する方法

7.3. データが送信されなかった場合の ISCSI ログイン失敗の確認

iSCSI ゲートウェイノードでは、システムログに一般的なログインネゴシエーション失敗のメッセージが、デフォルトで **/var/log/messages** に記録される場合があります。

例

```
Apr  2 23:17:05 osd1 kernel: rx_data returned 0, expecting 48.
Apr  2 23:17:05 osd1 kernel: iSCSI Login negotiation failed.
```

システムがこの状態である間、この手順で提案されているシステム情報の収集を開始します。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- iSCSI ターゲットとなる実行中の Ceph iSCSI ゲートウェイ。
- iSCSI イニシエーターである実行中の VMware ESXi 環境。
- Ceph iSCSI ゲートウェイノードへの root レベルのアクセス。
- VMware ESXi ノードへの root レベルのアクセス。

手順

1. 追加のロギングを有効にします。

```
[root@igw ~]# echo "module iscsi_target_mod +p" >
/sys/kernel/debug/dynamic_debug/control
[root@igw ~]# echo "module target_core_mod +p" >
/sys/kernel/debug/dynamic_debug/control
```

2. 追加のデバッグ情報がシステムログに反映されるまで、数分待ちます。

3. 追加のロギングを無効にします。

```
[root@igw ~]# echo "module iscsi_target_mod -p" >
/sys/kernel/debug/dynamic_debug/control
[root@igw ~]# echo "module target_core_mod -p" >
/sys/kernel/debug/dynamic_debug/control
```

4. **sosreport** を実行して、システム情報を収集します。

```
[root@igw ~]# sosreport
```

5. Ceph iSCSI ゲートウェイと VMware ESXi ノードのネットワークトラフィックを同時にキャプチャーします。

   構文

```
tcpdump -s0 -i NETWORK_INTERFACE -w OUTPUT_FILE_PATH
```

   例

```
[root@igw ~]# tcpdump -s 0 -i eth0 -w /tmp/igw-eth0-tcpdump.pcap
```

注記

ポート 3260 のトラフィックを検索します。

   a. ネットワークパケットキャプチャーファイルは大きくなる可能性があるため、Red Hat グローバルサポートサービスにファイルをアップロードする前に iSCSI ターゲットおよびイニシエーターからの **tcpdump** 出力を圧縮します。

      構文

```
gzip OUTPUT_FILE_PATH
```

      例

```
[root@igw ~]# gzip /tmp/igw-eth0-tcpdump.pcap
```

6. VMware ESXi 環境に関する追加情報を収集します。

```
[root@esx:~]# esxcli iscsi session list > /tmp/esxcli_iscsi_session_list.txt
[root@esx:~]# esxcli iscsi session connection list >
/tmp/esxcli_iscsi_session_connection_list.txt
```

a. 各 iSCSI ディスクの詳細情報を一覧表示し、収集します。

構文

```
esxcli storage nmp path list -d ISCSI_DISK_DEVICE >
/tmp/esxcli_storage_nmp_path_list.txt
```

例

```
[root@esx:~]# esxcli storage nmp device list
[root@esx:~]# esxcli storage nmp path list -d naa.60014054a5d46697f85498e9a257567c
> /tmp/esxcli_storage_nmp_path_list.txt
[root@esx:~]# esxcli storage core device list -d
naa.60014054a5d46697f85498e9a257567c > /tmp/esxcli_storage_core_device_list.txt
```

関連情報

- Red Hat グローバルサポートサービス向けに **sosreport** を作成する方法 は Red Hat ナレッジベースソリューションを参照してください。
- Red Hat グローバルサポートサービスの ファイルのアップロード に関する Red Hat ナレッジベースソリューションを参照してください。
- 詳細は、「tcpdump を使用してネットワークのパケットをキャプチャする方法」の Red Hat ナレッジベースのソリューションを参照してください。
- カスタマーポータルで Red Hat サポートケース を作成する方法

7.4. タイムアウトまたはポータルグループが見つからないことによる ISCSI ログイン失敗の確認

iSCSI ゲートウェイノードでは、デフォルトで **/var/log/messages** でタイムアウトが発生したり、システムログでターゲットポータルグループメッセージが見つからないことがあります。

例

```
Mar 28 00:29:01 osd2 kernel: iSCSI Login timeout on Network Portal 10.2.132.2:3260
```

または

例

```
Mar 23 20:25:39 osd1 kernel: Unable to locate Target Portal Group on iqn.2017-12.com.redhat.iscsi-gw:ceph-igw
```

システムがこの状態である間、この手順で提案されているシステム情報の収集を開始します。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- 実行中の Ceph iSCSI ゲートウェイ。

- Ceph iSCSI ゲートウェイノードへの root レベルのアクセス。

手順

1. 待機中のタスクのダンプを有効にして、ファイルに書き込みます。

```
[root@igw ~]# dmesg -c ; echo w > /proc/sysrq-trigger ; dmesg -c > /tmp/waiting-tasks.txt
```

2. 待機中のタスクのリストを見て、以下のメッセージを確認します。

   - **iscsit_tpg_disable_portal_group**
   - **core_tmr_abort_task**
   - **transport_generic_free_cmd**

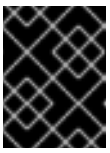
   待機タスクリストにこれらのメッセージが表示されると、**tcmu-runner** サービスで何らかの問題が発生したことを示します。**tcmu-runner** サービスが正常に再起動しなかったか、**tcmu-runner** サービスがクラッシュしました。

3. **tcmu-runner** サービスが実行中かどうかを確認します。

```
[root@igw ~]# systemctl status tcmu-runner
```

   a. **tcmu-runner** サービスが実行していない場合は、**tcmu-runner** サービスを再起動する前に **rbd-target-gw** サービスを停止します。

```
[root@igw ~]# systemctl stop rbd-target-gw
[root@igw ~]# systemctl stop tcmu-runner
[root@igw ~]# systemctl start tcmu-runner
[root@igw ~]# systemctl start rbd-target-gw
```



重要

Ceph iSCSI ゲートウェイを停止すると、まず **tcmu-runner** サービスがダウンしても IO がスタックできなくなります。

   b. **tcmu-runner** サービスが実行されている場合には、これは新しいバグである可能性があります。新しい Red Hat サポートケースを作成します。

関連情報

- Red Hat グローバルサポートサービス向けに **sosreport** を作成する方法 は Red Hat ナレッジベースソリューションを参照してください。
- Red Hat グローバルサポートサービスの ファイルのアップロード に関する Red Hat ナレッジベースソリューションを参照してください。
- カスタマーポータルで Red Hat サポートケース を作成する方法

7.5. タイムアウトコマンドエラー

システムログで SCSI コマンドが失敗した場合に、Ceph iSCSI ゲートウェイがコマンドタイムアウトエラーを報告する可能性があります。

例

```
Mar 23 20:03:14 igw tcmu-runner: 2018-03-23 20:03:14.052 2513 [ERROR]
tcmu_rbd_handle_timedout_cmd:669 rbd/rbd.gw1lun011: Timing out cmd.
```

または

例

```
Mar 23 20:03:14 igw tcmu-runner: tcmu_notify_conn_lost:176 rbd/rbd.gw1lun011: Handler
connection lost (lock state 1)
```

エラー内容:

他に処理待ちのタスクがあり、応答をタイムリーに受信できなかったために SCSI コマンドがタイムアウトした可能性があります。これらのエラーメッセージが表示されるもう 1 つの理由は、Red Hat Ceph Storage クラスターの健全性が低いことに関連している可能性があります。

この問題を解決するには、以下を行います。

1. 停滞しているタスクがないか確認します。
2. Red Hat Ceph Storage クラスターの健全性を確認します。
3. Ceph iSCSI ゲートウェイノードから iSCSI イニシエータノードへのパスにある各デバイスからシステム情報を収集します。

関連情報

- 待機中のタスクを表示する方法の詳細は、**『Red Hat Ceph Storage トラブルシューティングガイド』**の**「タイムアウトやポータルグループが見つからないなどの理由で iSCSI ログインに失敗した場合の確認」**セクションを参照してください。
- ストレージクラスターの健全性の確認に関する詳細は、**『Red Hat Ceph Storage トラブルシューティングガイド』**の**「ストレージクラスターの健常性の診断」**セクションを参照してください。
- 必要な情報を収集する方法の詳細は、**『Red Hat Ceph Storage トラブルシューティングガイド』**の**「VMware ESXi におけるストレージの原因となる接続障害情報を収集」**セクションを参照してください。

7.6. タスクのエラーの中止

Ceph iSCSI ゲートウェイが、システムログにアボートタスクエラーを報告する可能性があります。

例

```
Apr  1 14:23:58 igw kernel: ABORT_TASK: Found referenced iSCSI task_tag: 1085531
```

エラー内容:

スイッチの故障やポートの不良など、他のネットワーク障害がエラーメッセージの原因となる可能性があります。また、Red Hat Ceph Storageクラスターの健全性が低い可能性もあります。

この問題を解決するには、以下を行います。

この問題を解決するには、以下を行います。

1. 環境内のネットワークの障害を確認します。
2. Red Hat Ceph Storage クラスターの健全性を確認します。
3. Ceph iSCSI ゲートウェイノードから iSCSI イニシエータノードへのパスにある各デバイスからシステム情報を収集します。

関連情報

- ストレージクラスターの健全性の確認に関する詳細は、**『Red Hat Ceph Storage トラブルシューティングガイド』**の**「ストレージクラスターの健常性の診断」**セクションを参照してください。
- 必要な情報を収集する方法の詳細は、**『Red Hat Ceph Storage トラブルシューティングガイド』**の**「VMware ESXi におけるストレージの原因となる接続障害情報を収集」**セクションを参照してください。

7.7. 関連情報

- Ceph iSCSI ゲートウェイの詳細は、**『Red Hat Ceph Storage ブロックデバイスガイド』**を参照してください。
- 詳しくは、*3章 ネットワークの問題のトラブルシューティング*を参照してください。

第8章 CEPH 配置グループのトラブルシューティング

本セクションには、Ceph Placement Group (PG) に関連する最も一般的なエラーを修正するための情報が含まれています。

8.1. 前提条件

- ネットワーク接続を確認します。
- Monitor がクォーラムを形成できることを確認します。
- すべての正常な OSD が **up** して **in** であり、バックフィルおよびリカバリープロセスが完了したことを確認します。

8.2. 最も一般的な CEPH 配置グループエラー

以下の表では、**ceph health details** コマンドで返される最も一般的なエラーメッセージを一覧表示しています。この表には、エラーを説明し、問題を修正するための特定の手順を示す、対応セクションへのリンクがあります。

さらに、最適でない状態に陥っている配置グループをリストできます。詳しくは、「配置グループの一覧表示 (**stale**、**inactive**、または **unclean** 状態)」 を参照してください。

8.2.1. 前提条件

- 稼働中の Red Hat Ceph Storage クラスターがある。
- 稼働中の Ceph Object Gateway。

8.2.2. 配置グループのエラーメッセージ

一般的な配置グループエラーメッセージの表およびその修正方法。

エラーメッセージ	参照
HEALTH_ERR	
pgs down	配置グループが **down** している
pgs inconsistent	一貫性のない配置グループ
scrub errors	一貫性のない配置グループ
HEALTH_WARN	
pgs stale	古い配置グループ
unfound	不明なオブジェクト

8.2.3. 古い配置グループ

ceph health コマンドは、一部の配置グループ (PG) を **stale** 一覧で表示します。

```
HEALTH_WARN 24 pgs stale; 3/300 in osds are down
```

エラー内容:

モニターは、配置グループが動作しているセットのプライマリー OSD からステータスの更新を受け取らない場合や、プライマリー OSD が **down** していると他の OSD が報告されない場合に、配置グループを **stale** とマークします。

通常、PG はストレージクラスターを起動し、ピアリングプロセスが完了するまで、**stale** 状態になります。ただし、PG が想定よりも **stale** である (古くなっている) 場合は、PG のプライマリー OSD が **ダウン** しているか、または PG 統計をモニターに報告していないことを示す可能性があります。**古い** PG を保存するプライマリー OSD が **up** に戻ると、Ceph は PG の復元を開始します。

mon_osd_report_timeout の設定は、OSD が PG の統計をモニターに報告する頻度を決定します。デフォルトでは、このパラメーターは **0.5** に設定されています。これは、OSD が 0.5 秒ごとに統計を報告することを意味します。

この問題を解決するには、以下を行います。

1. **古い** PG とそれらが保存される OSD を特定します。エラーメッセージには、以下の例のような情報が含まれます。

   例

   ```
   # ceph health detail
   HEALTH_WARN 24 pgs stale; 3/300 in osds are down
   ...
   pg 2.5 is stuck stale+active+remapped, last acting [2,0]
   ...
   osd.10 is down since epoch 23, last address 192.168.106.220:6800/11080
   osd.11 is down since epoch 13, last address 192.168.106.220:6803/11539
   osd.12 is down since epoch 24, last address 192.168.106.220:6806/11861
   ```

2. **down** とマークされている OSD の問題のトラブルシューティング。詳細は、「Down OSD」を参照してください。

関連情報

- Red Hat Ceph Storage 4 **『管理ガイド』**の「配置グループセットの監視」セクション

8.2.4. 一貫性のない配置グループ

一部の配置グループは **active + clean + inconsistent** とマークされ、**ceph health detail** は以下のようなエラーメッセージを返します。

```
HEALTH_ERR 1 pgs inconsistent; 2 scrub errors
pg 0.6 is active+clean+inconsistent, acting [0,1,2]
2 scrub errors
```

エラー内容:

Ceph は、配置グループ内のオブジェクトの 1 つ以上のレプリカで不整合を検出すると、配置グループに **inconsistent** のマークを付けます。最も一般的な不整合は以下のとおりです。

- オブジェクトのサイズが正しくない。
- リカバリーが終了後、あるレプリカのオブジェクトが失われた。

ほとんどの場合、スクラビング中のエラーが原因で、配置グループ内の不整合が発生します。

この問題を解決するには、以下を行います。

1. どの配置グループが **一貫性のない** 状態かを決定します。

```
# ceph health detail
HEALTH_ERR 1 pgs inconsistent; 2 scrub errors
pg 0.6 is active+clean+inconsistent, acting [0,1,2]
2 scrub errors
```

2. 配置グループに **inconsistent** な理由を決定します。
   a. 配置グループでディープスクラビングプロセスを開始します。

```
[root@mon ~]# ceph pg deep-scrub ID
```

   ID を、以下のように **inconsistent** 配置グループの ID に置き換えます。

```
[root@mon ~]# ceph pg deep-scrub 0.6
instructing pg 0.6 on osd.0 to deep-scrub
```

   b. **ceph -w** の出力で、その配置グループに関連するメッセージを探します。

```
ceph -w | grep ID
```

   ID を、以下のように **inconsistent** 配置グループの ID に置き換えます。

```
[root@mon ~]# ceph -w | grep 0.6
2015-02-26 01:35:36.778215 osd.106 [ERR] 0.6 deep-scrub stat mismatch, got 636/635
objects, 0/0 clones, 0/0 dirty, 0/0 omap, 0/0 hit_set_archive, 0/0 whiteouts,
1855455/1854371 bytes.
2015-02-26 01:35:36.788334 osd.106 [ERR] 0.6 deep-scrub 1 errors
```

3. 出力に以下のようなエラーメッセージが含まれる場合は、**inconsistent** 配置グループを修復できます。詳細は、**「一貫性のない配置グループの修正」**を参照してください。

```
PG.ID shard OSD: soid OBJECT missing attr , missing attr _ATTRIBUTE_TYPE
PG.ID shard OSD: soid OBJECT digest 0 != known digest DIGEST, size 0 != known size
SIZE
PG.ID shard OSD: soid OBJECT size 0 != known size SIZE
PG.ID deep-scrub stat mismatch, got MISMATCH
PG.ID shard OSD: soid OBJECT candidate had a read error, digest 0 != known digest
DIGEST
```

4. 出力に以下のようなエラーメッセージが含まれる場合は、データが失われる可能性があるため、**inconsistent** のない配置グループを修正しても安全ではありません。この場合、サポートチケットを作成します。詳細は「Red Hat サポートへの問い合わせ」を参照してください。

```
PG.ID shard OSD: soid OBJECT digest DIGEST != known digest DIGEST
PG.ID shard OSD: soid OBJECT omap_digest DIGEST != known omap_digest DIGEST
```

関連情報

- 『Red Hat Ceph Storage トラブルシューティングガイド』の「配置グループの不整合の知覧表示」。
- 『Red Hat Ceph Storage アーキテクチャーガイド』の「Ceph データ整合性」セクションを参照してください。
- 『Red Hat Ceph Storage 設定ガイド』の 「OSD のスクラブ」セクション。

8.2.5. 不適切な配置グループ

ceph health コマンドは、以下のようなエラーメッセージを返します。

```
HEALTH_WARN 197 pgs stuck unclean
```

エラー内容:

Ceph 設定ファイルの **mon_pg_stuck_threshold** パラメーターで指定された秒数について、**active+clean** の状態を満たさない場合には、Ceph 配置グループは **unclean** とマーク付けされます。**mon_pg_stuck_threshold** のデフォルト値は **300** 秒です。

配置グループが **unclean** である場合は、**osd_pool_default_size** パラメーターで指定された回数複製されないオブジェクトが含まれます。**osd_pool_default_size** のデフォルト値は **3** で、Ceph はレプリカを 3 つ作成します。

通常、**unclean** 配置グループは、一部の OSD が **down** している可能性があることを意味します。

この問題を解決するには、以下を行います。

1. **down** になっている OSD を特定します。

```
# ceph osd tree
```

2. OSD の問題をトラブルシューティングし、修正します。詳細は、「Down OSD」を参照してください。

関連情報

- 配置グループの一覧表示が、古い非アクティブな状態または不完全な状態

8.2.6. 非アクティブな配置グループ

ceph health コマンドは、以下のようなエラーメッセージを返します。

```
HEALTH_WARN 197 pgs stuck inactive
```

エラー内容:

Ceph 設定ファイルの **mon_pg_stuck_threshold** パラメーターで指定された秒数について、配置グループが非表示になっていない場合、Ceph はその配置グループを **inactive** とマークします。**mon_pg_stuck_threshold** のデフォルト値は **300** 秒です。

通常、**inactive** な配置グループは一部の OSD が **down** となっている可能性があることを示します。

この問題を解決するには、以下を行います。

1. **down** になっている OSD を特定します。

```
# ceph osd tree
```

2. OSD の問題をトラブルシューティングし、修正します。

関連情報

- 古くて非アクティブまたは不完全な状態になっている配置グループの一覧表示
- 詳細は、「Down OSD」を参照してください。

8.2.7. down している配置グループ

ceph health detail コマンドは、一部の配置グループが **down** していると報告します。

```
HEALTH_ERR 7 pgs degraded; 12 pgs down; 12 pgs peering; 1 pgs recovering; 6 pgs stuck
unclean; 114/3300 degraded (3.455%); 1/3 in osds are down
...
pg 0.5 is down+peering
pg 1.4 is down+peering
...
osd.1 is down since epoch 69, last address 192.168.106.220:6801/8651
```

エラー内容:

場合によっては、ピアリングプロセスがブロックされ、配置グループがアクティブになって使用できなくなることがあります。通常、OSD の障害が原因でピアリングの障害が発生します。

この問題を解決するには、以下を行います。

ピアリング処理をブロックしている原因を判断します。

```
[root@mon ~]# ceph pg ID query
```

ID を、**down** する配置グループの ID に置き換えます。以下に例を示します。

```
[root@mon ~]# ceph pg 0.5 query

{ "state": "down+peering",
  ...
  "recovery_state": [
      { "name": "Started\/Primary\/Peering\/GetInfo",
        "enter_time": "2012-03-06 14:40:16.169679",
        "requested_info_from": []},
```

```
        { "name": "Started\/Primary\/Peering",
          "enter_time": "2012-03-06 14:40:16.169659",
          "probing_osds": [
                0,
                1],
          "blocked": "peering is blocked due to down osds",
          "down_osds_we_would_probe": [
                1],
          "peering_blocked_by": [
                { "osd": 1,
                  "current_lost_at": 0,
                  "comment": "starting or marking this osd lost may let us proceed"}]},
        { "name": "Started",
          "enter_time": "2012-03-06 14:40:16.169513"}
    ]
}
```

recovery_state セクションには、ピアリングプロセスがブロックされた理由が含まれます。

- 出力には **peering is blocked due to down osds** エラーメッセージが含まれているため「Down OSD」を参照してください。
- 他のエラーメッセージが表示された場合は、サポートチケットを作成します。詳細は、「Red Hat サポートサービスへの問い合わせ」を参照してください。

関連情報

- 『Red Hat Ceph Storage Administration Guide』の「Ceph OSD ピアリング」セクション

8.2.8. 不明なオブジェクト

ceph health コマンドは、**unfound** キーワードを含む以下のようなエラーメッセージを返します。

```
HEALTH_WARN 1 pgs degraded; 78/3778 unfound (2.065%)
```

エラー内容:

これらのオブジェクトまたは新しいコピーが分かっている場合には、Ceph のマークは **unfound** とマークしますが、オブジェクトが見つからないと判断できません。そのため、Ceph はそのようなオブジェクトを回復できず、リカバリープロセスを続行できません。

状況例

配置グループは、**osd.1** および **osd.2** にデータを格納します。

1. **osd.1** は **down** します。
2. **osd.2** は一部の書き込み操作を処理します。
3. **osd.1** が **up** とないります。
4. **osd.1** と **osd.2** の間のピアリングプロセスは開始し、**osd.1** にないオブジェクトはリカバリーのためにキューに置かれます。
5. Ceph が新規オブジェクトをコピーする前に、**osd.2** が **down** となります。

その結果、**osd.1** はこれらのオブジェクトが存在することを認識しますが、オブジェクトのコピーを持つ OSD はありません。

このシナリオでは、Ceph は障害が発生したノードが再びアクセス可能になるのを待機しており、未使用の **unfound** によりリカバリープロセスがブロックされます。

この問題を解決するには、以下を行います。

1. **unfound** オブジェクトが含まれる配置グループを決定します。

```
[root@mon ~]# ceph health detail
HEALTH_WARN 1 pgs recovering; 1 pgs stuck unclean; recovery 5/937611 objects
degraded (0.001%); 1/312537 unfound (0.000%)
pg 3.8a5 is stuck unclean for 803946.712780, current state active+recovering, last acting
[320,248,0]
pg 3.8a5 is active+recovering, acting [320,248,0], 1 unfound
recovery 5/937611 objects degraded (0.001%); **1/312537 unfound (0.000%)**
```

2. 配置グループに関する詳細情報を表示します。

```
[root@mon ~]# ceph pg ID query
```

ID を、以下のように、**unfound** オブジェクトを含む配置グループの ID に置き換えます。

```
[root@mon ~]# ceph pg 3.8a5 query
{ "state": "active+recovering",
  "epoch": 10741,
  "up": [
        320,
        248,
        0],
  "acting": [
        320,
        248,
        0],
<snip>
  "recovery_state": [
        { "name": "Started\/Primary\/Active",
          "enter_time": "2015-01-28 19:30:12.058136",
          "might_have_unfound": [
                { "osd": "0",
                  "status": "already probed"},
                { "osd": "248",
                  "status": "already probed"},
                { "osd": "301",
                  "status": "already probed"},
                { "osd": "362",
                  "status": "already probed"},
                { "osd": "395",
                  "status": "already probed"},
                { "osd": "429",
                  "status": "osd is down"}],
          "recovery_progress": { "backfill_targets": [],
              "waiting_on_backfill": [],
              "last_backfill_started": "0\/\/0\/\/-1",
```

```
        "backfill_info": { "begin": "0\/\/0\/\/-1",
            "end": "0\/\/0\/\/-1",
            "objects": []},
        "peer_backfill_info": [],
        "backfills_in_flight": [],
        "recovering": [],
        "pg_backend": { "pull_from_peer": [],
            "pushing": []}},
      "scrub": { "scrubber.epoch_start": "0",
        "scrubber.active": 0,
        "scrubber.block_writes": 0,
        "scrubber.finalizing": 0,
        "scrubber.waiting_on": 0,
        "scrubber.waiting_on_whom": []}},
    { "name": "Started",
      "enter_time": "2015-01-28 19:30:11.044020"}],
```

might_have_unfound セクションには、Ceph が **unfound** オブジェクトの検索を試行する OSD が含まれます。

- **already probed** ステータスは、Ceph が OSD 内で **unfound** オブジェクトを検出できないことを示します。
- **osd is down** 状態は、Ceph が OSD と通信できないことを示します。

3. **down** とマークされている OSD のトラブルシューティング詳細は、「Down OSD」を参照してください。
4. OSD が **down** となる問題を修正できない場合は、サポートチケットを作成してください。詳細は、「サービスについて Red Hat サポートへの問い合わせ」を参照してください。

8.3. 配置グループの一覧表示 (STALE、 INACTIVE、 または UNCLEAN 状態)

失敗した後、配置グループは **degraded** や **peering** などの状態になります。この状態は、障害リカバリープロセスが正常に進行していることを示しています。

しかし、ある配置グループが予想よりも長い期間これらの状態のいずれかになる場合、より大きな問題の兆候である可能性があります。配置グループが最適ではない状態のままになると、Monitor が報告します。

Ceph 設定ファイルの **mon_pg_stuck_threshold** オプションにより、配置グループが **inactive**、**unclean**、または **stale** とみなされるまでの秒数を決定します。

以下の表は、これらの状態と簡単な説明を示しています。

状態	意味	最も一般的な原因	参照
inactive	PG は読み取り/書き込み要求に対応できません。	• ピアリングの問題	非アクティブな配置グループ

状態	意味	最も一般的な原因	参照
unclean	PG には、必要な回数を複製されていないオブジェクトが含まれます。何かが PG の回復を妨げている。	• **unfound** オブジェクト • OSD が **down** している • 不適切な設定	不適切な配置グループ
stale	PG のステータスは、**ceph-osd** デーモンによって更新されていません。	• OSD が **down** している	古い配置グループ

前提条件

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

手順

1. スタックした PG をリストします。

```
[root@mon ~]# ceph pg dump_stuck inactive
[root@mon ~]# ceph pg dump_stuck unclean
[root@mon ~]# ceph pg dump_stuck stale
```

関連情報

- 『Red Hat Ceph Storage 管理ガイド』の「配置グループの状態」セクションを参照してください。

8.4. 配置グループ不整合の一覧表示

rados ユーティリティーを使用して、オブジェクトのさまざまなレプリカで不整合を一覧表示します。より詳細な出力を一覧表示するには、**--format=json-pretty** オプションを使用します。

本セクションでは、以下を取り上げます。

- **プールへの一貫性のない配置グループ**
- **配置グループの一貫性のないオブジェクト**
- **配置グループにおける一貫性のないスナップショットセット**

前提条件

- 健全な状態で稼働中の Red Hat Ceph Storage クラスター。

- ノードへのルートレベルのアクセス。

手順

```
rados list-inconsistent-pg POOL --format=json-pretty
```

たとえば、**data** という名前のプール内の一貫性のない配置グループの一覧を表示します。

```
# rados list-inconsistent-pg data --format=json-pretty
[0.6]
```

```
rados list-inconsistent-obj PLACEMENT_GROUP_ID
```

たとえば、ID **0.6** の配置グループに一貫性のないオブジェクトの一覧を表示します。

```
# rados list-inconsistent-obj 0.6
{
    "epoch": 14,
    "inconsistents": [
        {
            "object": {
                "name": "image1",
                "nspace": "",
                "locator": "",
                "snap": "head",
                "version": 1
            },
            "errors": [
                "data_digest_mismatch",
                "size_mismatch"
            ],
            "union_shard_errors": [
                "data_digest_mismatch_oi",
                "size_mismatch_oi"
            ],
            "selected_object_info": "0:602f83fe:::foo:head(16'1 client.4110.0:1
dirty|data_digest|omap_digest s 968 uv 1 dd e978e67f od ffffffff alloc_hint [0 0 0])",
            "shards": [
                {
                    "osd": 0,
                    "errors": [],
                    "size": 968,
                    "omap_digest": "0xffffffff",
                    "data_digest": "0xe978e67f"
                },
                {
                    "osd": 1,
                    "errors": [],
                    "size": 968,
                    "omap_digest": "0xffffffff",
                    "data_digest": "0xe978e67f"
                },
                {
                    "osd": 2,
```

```
                "errors": [
                    "data_digest_mismatch_oi",
                    "size_mismatch_oi"
                ],
                "size": 0,
                "omap_digest": "0xffffffff",
                "data_digest": "0xffffffff"
            }
        ]
    }
  ]
}
```

不整合の原因を特定するには、以下のフィールドが重要になります。

- **name**: 一貫性のないレプリカを持つオブジェクトの名前。
- **nSpace**: プールを論理的に分離する名前空間。デフォルトでは空です。
- **Static**: 配置のオブジェクト名の代わりに使用されるキー。
- **snap**: オブジェクトのスナップショット ID。オブジェクトの書き込み可能な唯一のバージョンは **head** と呼ばれます。オブジェクトがクローンの場合、このフィールドにはそのシーケンシャル ID が含まれます。
- **version**: 一貫性のないレプリカを持つオブジェクトのバージョン ID。オブジェクトへの書き込み操作ごとにインクリメントされます。
- **errors**: シャードの不一致を判別することなくシャード間の不整合を示すエラーの一覧。エラーをさらに調べるには、**shard** アレイを参照してください。
  - **data_digest_mismatch**: 1 つの OSD から読み取られるレプリカのダイジェストは他の OSD とは異なります。
  - **size_mismatch**: クローンのサイズまたは **head** オブジェクトが期待したサイズと一致しない。
  - **read_error**: このエラーは、ディスクエラーが発生したために不整合が発生したことを示しています。
- **union_shard_error**: シャードに固有のすべてのエラーの結合。これらのエラーは、問題のあるシャードに関連しています。**oi** で終わるエラーは、障害のあるオブジェクトからの情報と、選択したオブジェクトとの情報を比較する必要があることを示しています。エラーをさらに調べるには、**shard** アレイを参照してください。
  上記の例では、**osd.2** に保存されているオブジェクトレプリカは、**osd.0** および **osd.1** に保存されているレプリカとは異なるダイジェストを持ちます。具体的には、レプリカのダイジェストは、**osd.2** から読み取るシャードから計算した **0xffffffff** ではなく、**0xe978e67f** です。さらに、**osd.2** から読み込むレプリカのサイズは 0 ですが、**osd.0** および **osd.1** によって報告されるサイズは 968 です。

```
rados list-inconsistent-snapset PLACEMENT_GROUP_ID
```

たとえば、ID **0.23** の配置グループにおける一貫性のないスナップショット (**snapsets**) の一覧を表示します。

```
# rados list-inconsistent-snapset 0.23 --format=json-pretty
```

```
{
    "epoch": 64,
    "inconsistents": [
        {
            "name": "obj5",
            "nspace": "",
            "locator": "",
            "snap": "0x00000001",
            "headless": true
        },
        {
            "name": "obj5",
            "nspace": "",
            "locator": "",
            "snap": "0x00000002",
            "headless": true
        },
        {
            "name": "obj5",
            "nspace": "",
            "locator": "",
            "snap": "head",
            "ss_attr_missing": true,
            "extra_clones": true,
            "extra clones": [
                2,
                1
            ]
        }
    ]
```

このコマンドは、以下のエラーを返します。

- **ss_attr_missing**: 1 つ以上の属性がありません。属性とは、スナップショットに関する情報で、キーと値のペアのリストとしてスナップショットセットにエンコードされます。
- **ss_attr_corrupted**: 1 つ以上の属性がデコードできません。
- **clone_missing**: クローンがありません。
- **snapset_mismatch**: スナップショットセット自体に一貫性がありません。
- **head_mismatch**: スナップショットセットは、**head** が存在するか、または存在しない場合はスクラブ結果を報告します。
- **headless**: スナップショットセットの **head** がありません。
- **size_mismatch**: クローンのサイズまたは **head** オブジェクトが期待したサイズと一致しない。

関連情報

- 『Red Hat Ceph Storage トラブルシューティングガイド』の「一貫性のない配置グループ」セクション。
- 『Red Hat Ceph Storage トラブルシューティングガイド』の「一貫性のない配置グループ」セクション。

8.5. 不整合な配置グループの修正

ディープスクラビング中のエラーにより、一部の配置グループの整合性が失われる可能性があります。Ceph は、配置グループの **inconsistent** をとります。

```
HEALTH_ERR 1 pgs inconsistent; 2 scrub errors
pg 0.6 is active+clean+inconsistent, acting [0,1,2]
2 scrub errors
```

警告

特定の不整合のみを修復できます。

Ceph のログに以下のエラーが含まれている場合は、配置グループを修復しないでください。

```
_PG_._ID_ shard _OSD_: soid _OBJECT_ digest _DIGEST_ != known digest _DIGEST_
_PG_._ID_ shard _OSD_: soid _OBJECT_ omap_digest _DIGEST_ != known omap_digest
_DIGEST_
```

代わりにサポートチケットを作成してください。詳細は、**「サービスについて Red Hat サポートへの問い合わせ」**を参照してください。

前提条件

- Ceph Monitor ノードへのルートレベルのアクセス。

手順

1. **inconsistent** 配置グループを修復します。

```
[root@mon ~]# ceph pg repair ID
```

1. **ID** を、**inconsistent** 配置グループの ID に置き換えます。

関連情報

- **『Red Hat Ceph Storage トラブルシューティングガイド』**の**「一貫性のない配置グループ」**セクション。
- **「配置グループの不整合の一覧表示」 Red Hat Ceph Storage トラブルシューティングガイド。**

8.6. 配置グループの増加

配置グループ (PG) 数が十分でないと、Ceph クラスターおよびデータ分散のパフォーマンスに影響します。これは、**nearfull osds** エラーメッセージの主な原因の 1 つです。

推奨される比率は、OSD 1つに対して 100 から 300 個の PG です。この比率は、OSD をクラスターに追加すると減らすことができます。

pg_num パラメーターおよび **pgp_num** パラメーターにより、PG 数が決まります。これらのパラメーターは各プールごとに設定されるため、PG 数が少ないプールは個別に調整する必要があります。

重要

PG 数を増やすことは、Ceph クラスターで実行できる最も負荷のかかる処理です。このプロセスは、ゆっくりと計画的に行わないと、パフォーマンスに深刻な影響を与える可能性があります。**pgp_num** を増やすと、プロセスを停止したり元に戻したりすることはできず、完了する必要があります。ビジネスクリティカルな処理時間の割り当て以外で PG 数を増やすことを検討し、パフォーマンスに影響を与える可能性があることをすべてのクライアントに警告します。クラスターが **HEALTH_ERR** 状態にある場合は、PG 数を変更しないでください。

前提条件

- 健全な状態で稼働中の Red Hat Ceph Storage クラスター。
- ノードへのルートレベルのアクセス。

手順

1. データの再分配やリカバリーが個々の OSD や OSD ホストに与える影響を軽減します。
   a. **osd max backfills**、**osd_recovery_max_active**、および **osd_recovery_op_priority** パラメーターの値を減らします。

```
[root@mon ~]# ceph tell osd.* injectargs '--osd_max_backfills 1 --
osd_recovery_max_active 1 --osd_recovery_op_priority 1'
```

   b. シャローおよびディープスクラビングを無効にします。

```
[root@mon ~]# ceph osd set noscrub
[root@mon ~]# ceph osd set nodeep-scrub
```

2. Ceph Placement Groups (PGs) per Pool Calculatorを使用して、**pg_num** パラメーターおよび **pgp_num** パラメーターの最適な値を計算します。
3. 必要な値に達するまで、**pg_num** の値を少し増やします。
   a. インクリメントの開始値を決定します。2の累乗である非常に低い値を使用し、クラスターへの影響を判断して増やします。最適な値は、プールサイズ、OSD 数、クライアント I/O 負荷によって異なります。
   b. **pg_num** の値を増やします。

```
ceph osd pool set POOL pg_num VALUE
```

   プール名と新しい値を指定します。例を以下に示します。

```
# ceph osd pool set data pg_num 4
```

c. クラスターのステータスを監視します。

```
# ceph -s
```

PG の状態は、**creating** から **active+clean** に変わります。すべての PG が **active+clean** の状態になるまで待ちます。

4. 必要な値に達するまで、**pgp_num** の値を少し増やします。

a. インクリメントの開始値を決定します。2の累乗である非常に低い値を使用し、クラスターへの影響を判断して増やします。最適な値は、プールサイズ、OSD 数、クライアント I/O 負荷によって異なります。

b. **pgp_num** の値を増やします。

```
ceph osd pool set POOL pgp_num VALUE
```

プール名と新しい値を指定します。例を以下に示します。

```
# ceph osd pool set data pgp_num 4
```

c. クラスターのステータスを監視します。

```
# ceph -s
```

PG の状態は、**peering**、**wait_backfill**、**backfilling**、**recover** などによって変わります。すべての PG が **active+clean** の状態になるまで待ちます。

5. PG 数が不足しているすべてのプールに対して、前の手順を繰り返します。

6. **osd max backfills**、**osd_recovery_max_active**、および **osd_recovery_op_priority** をデフォルト値に設定します。

```
# ceph tell osd.* injectargs '--osd_max_backfills 1 --osd_recovery_max_active 3 --
osd_recovery_op_priority 3'
```

7. シャローおよびディープスクラビングを有効にします。

```
# ceph osd unset noscrub
# ceph osd unset nodeep-scrub
```

関連情報

- Nearfull OSD
- Red Hat Ceph Storage 4 の**『管理ガイド』**の**「配置グループ設定の監視」**セクション

8.7. 関連情報

- 詳しくは、*3章ネットワークの問題のトラブルシューティング*を参照してください。
- Ceph Monitor に関連する最も一般的なエラーのトラブルシューティングについては、*4章Ceph Monitor のトラブルシューティング*を参照してください。

- Ceph OSD に関連する最も一般的なエラーのトラブルシューティングに関する情報は、5 章*Ceph OSD のトラブルシューティング*を参照してください。

第9章 CEPH オブジェクトのトラブルシューティング

ストレージ管理者は、**ceph-objectstore-tool** ユーティリティーを使用して高レベルまたは低レベルのオブジェクト操作を実行することができます。**ceph-objectstore-tool** ユーティリティーは、特定の OSD または配置グループ内のオブジェクトに関する問題のトラブルシューティングに役立ちます。

重要

オブジェクトを操作すると、回復不能なデータ損失が発生する可能性があります。**ceph-objectstore-tool** ユーティリティーを使用する前に、Red Hat サポートにお問い合わせください。

9.1. 前提条件

- ネットワーク関連の問題がないことを確認します。

9.2. ハイレベルなオブジェクト操作のトラブルシューティング

ストレージ管理者は、**ceph-objectstore-tool** ユーティリティーを使用して高レベルのオブジェクト操作を実行することができます。**ceph-objectstore-tool** ユーティリティーは、以下の高レベルのオブジェクト操作をサポートします。

- オブジェクトの一覧表示
- 失われたオブジェクトの一覧表示
- 失われたオブジェクトの修正

重要

オブジェクトを操作すると、回復不能なデータ損失が発生する可能性があります。**ceph-objectstore-tool** ユーティリティーを使用する前に、Red Hat サポートにお問い合わせください。

9.2.1. 前提条件

- Ceph OSD ノードへのルートレベルのアクセス。

9.2.2. オブジェクトの一覧表示

OSD には、ゼロ対多の配置グループを含めることができ、1 つの配置グループ (PG) 内にゼロ対多のオブジェクトを含めることができます。**ceph-objectstore-tool** ユーティリティーでは、OSD に保存されているオブジェクトを一覧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。
- **ceph-osd** デーモンの停止。

手順

1. 適切な OSD がダウンしていることを確認します。

```
[root@osd ~]# systemctl status ceph-osd@OSD_NUMBER
```

例

```
[root@osd ~]# systemctl status ceph-osd@1
```

2. 配置グループに関係なく、OSD 内のすべてのオブジェクトを特定します。

```
[root@osd ~]# ceph-objectstore-tool --data-path PATH_TO_OSD --op list
```

例

```
[root@osd ~]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --op list
```

3. 配置グループ内のすべてのオブジェクトを特定します。

```
[root@osd ~]# ceph-objectstore-tool --data-path PATH_TO_OSD --pgid PG_ID --op list
```

例

```
[root@osd ~]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --pgid 0.1c --op list
```

4. オブジェクトが属する PG を特定します。

```
[root@osd ~]# ceph-objectstore-tool --data-path PATH_TO_OSD --op list OBJECT_ID
```

例

```
[root@osd ~]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --op list
default.region
```

- OSD の停止に関する詳細は、**『Red Hat Ceph Storage 管理ガイド』**の**「インスタンスごとの Ceph デーモンの開始、停止、および再起動」**セクションを参照してください。

9.2.3. 失われたオブジェクトの修正

ceph-objectstore-tool ユーティリティーを使用して、Ceph OSD に保存されている **失われたオブジェクトおよび存在しないオブジェクト** を一覧表示し、修正することができます。この手順は、レガシーオブジェクトにのみ適用されます。

前提条件

- Ceph OSD ノードへのルートレベルのアクセス。
- **ceph-osd** デーモンの停止。

手順

1. 適切な OSD がダウンしていることを確認します。

構文

```
[root@osd ~]# systemctl status ceph-osd@OSD_NUMBER
```

例

```
[root@osd ~]# systemctl status ceph-osd@1
```

2. 失われたレガシーオブジェクトをすべて一覧表示します。

構文

```
ceph-objectstore-tool --data-path PATH_TO_OSD --op fix-lost --dry-run
```

例

```
[root@osd ~]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --op fix-lost --dry-run
```

3. **ceph-objectstore-tool** ユーティリティーを使用して、**失われたおよび未使用** のオブジェクトを修正します。適切な状況を選択します。

   a. 失われたオブジェクトをすべて修正します。

   構文

   ```
   [root@osd ~]# ceph-objectstore-tool --data-path PATH_TO_OSD --op fix-lost
   ```

   例

   ```
   [root@osd ~]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --op fix-lost
   ```

   b. 配置グループ内の失われたオブジェクトをすべて修正します。

   構文

   ```
   [root@osd ~]# ceph-objectstore-tool --data-path PATH_TO_OSD --pgid PG_ID --op fix-lost
   ```

   例

   ```
   [root@osd ~]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --pgid 0.1c --op fix-lost
   ```

   c. 失われたオブジェクトを識別子で修正します。

   構文

   ```
   [root@osd ~]# ceph-objectstore-tool --data-path PATH_TO_OSD --op fix-lost OBJECT_ID
   ```

   例

```
[root@osd ~]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --op fix-lost
default.region
```

- OSD の停止に関する詳細は、**『Red Hat Ceph Storage 管理ガイド』**の**「インスタンスごとの Ceph デーモンの開始、停止、および再起動」**セクションを参照してください。

9.3. 低レベルのオブジェクト操作のトラブルシューティング

ストレージ管理者は、**ceph-objectstore-tool** ユーティリティーを使用して低レベルのオブジェクト操作を実行することができます。**ceph-objectstore-tool** ユーティリティーは、以下の低レベルのオブジェクト操作をサポートします。

- オブジェクトの内容の操作
- オブジェクトの削除
- オブジェクトマップ (OMAP) の一覧表示
- OMAP ヘッダーの操作
- OMAP キーの操作
- オブジェクトの属性の一覧表示
- オブジェクトの属性キーの操作

重要

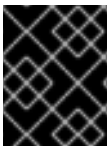
オブジェクトを操作すると、回復不能なデータ損失が発生する可能性があります。**ceph-objectstore-tool** ユーティリティーを使用する前に、Red Hat サポートにお問い合わせください。

9.3.1. 前提条件

- Ceph OSD ノードへのルートレベルのアクセス。

9.3.2. オブジェクトの内容の操作

ceph-objectstore-tool ユーティリティーを使用すると、オブジェクトのバイトを取得または設定できます。



重要

オブジェクトにバイト数を設定すると、回復できないデータ損失が発生する可能性があります。データの損失を防ぐには、オブジェクトのバックアップコピーを作成します。

前提条件

- Ceph OSD ノードへのルートレベルのアクセス。
- **ceph-osd** デーモンの停止。

手順

1. 適切な OSD がダウンしていることを確認します。

```
[root@osd ~]# systemctl status ceph-osd@$OSD_NUMBER
```

例

```
[root@osd ~]# systemctl status ceph-osd@1
```

2. OSD または配置グループ (PG) のオブジェクトを一覧表示してオブジェクトを見つけます。

3. オブジェクトにバイトを設定する前に、そのオブジェクトのバックアップと作業コピーを作成します。

```
[root@osd ~]# ceph-objectstore-tool --data-path PATH_TO_OSD --pgid PG_ID \
OBJECT \
get-bytes > OBJECT_FILE_NAME

[root@osd ~]# ceph-objectstore-tool --data-path PATH_TO_OSD --pgid PG_ID \
OBJECT \
get-bytes > OBJECT_FILE_NAME
```

例

```
[root@osd ~]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --pgid 0.1c \
'{"oid":"zone_info.default","key":"","snapid":-
2,"hash":235010478,"max":0,"pool":11,"namespace":""}'  \
get-bytes > zone_info.default.backup

[root@osd ~]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --pgid 0.1c \
'{"oid":"zone_info.default","key":"","snapid":-
2,"hash":235010478,"max":0,"pool":11,"namespace":""}'  \
get-bytes > zone_info.default.working-copy
```

4. 作業コピーオブジェクトファイルを編集し、それに応じてオブジェクトの内容を変更します。

5. オブジェクトのバイトを設定します。

```
[root@osd ~]# ceph-objectstore-tool --data-path PATH_TO_OSD --pgid PG_ID \
OBJECT \
set-bytes < OBJECT_FILE_NAME
```

例

```
[root@osd ~]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --pgid 0.1c \
'{"oid":"zone_info.default","key":"","snapid":-
2,"hash":235010478,"max":0,"pool":11,"namespace":""}' \
set-bytes < zone_info.default.working-copy
```

- OSD の停止に関する詳細は、**『Red Hat Ceph Storage 管理ガイド』**の**「インスタンスごとの Ceph デーモンの開始、停止、および再起動」**セクションを参照してください。

9.3.3. オブジェクトの削除

ceph-objectstore-tool ユーティリティーを使用してオブジェクトを削除します。オブジェクトを削除すると、そのコンテンツと参照は配置グループ (PG) から削除されます。

重要

オブジェクトが削除されると、再作成できません。

前提条件

- Ceph OSD ノードへのルートレベルのアクセス。
- **ceph-osd** デーモンの停止。

手順

1. オブジェクトの削除

```
[root@osd ~]# ceph-objectstore-tool --data-path PATH_TO_OSD --pgid PG_ID \
OBJECT \
remove
```

例

```
[root@osd ~]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --pgid 0.1c \
'{"oid":"zone_info.default","key":"","snapid":-
2,"hash":235010478,"max":0,"pool":11,"namespace":""}' \
remove
```

- OSD の停止に関する詳細は、**『Red Hat Ceph Storage 管理ガイド』**の**「インスタンスごとの Ceph デーモンの開始、停止、および再起動」**セクションを参照してください。

9.3.4. オブジェクトマップの一覧表示

ceph-objectstore-tool ユーティリティーを使用して、オブジェクトマップ (OMAP) の内容を一覧表示します。この出力では、キーの一覧が表示されます。

前提条件

- Ceph OSD ノードへのルートレベルのアクセス。
- **ceph-osd** デーモンの停止。

手順

1. 適切な OSD がダウンしていることを確認します。

```
[root@osd ~]# systemctl status ceph-osd@OSD_NUMBER
```

例

```
[root@osd ~]# systemctl status ceph-osd@1
```

2. オブジェクトマップを一覧表示します。

```
[root@osd ~]# ceph-objectstore-tool --data-path PATH_TO_OSD --pgid PG_ID \
OBJECT \
list-omap
```

例

```
[root@osd ~]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --pgid 0.1c \
'{"oid":"zone_info.default","key":"","snapid":-
2,"hash":235010478,"max":0,"pool":11,"namespace":""}' \
list-omap
```

- OSD の停止に関する詳細は、**『Red Hat Ceph Storage 管理ガイド』**の**「インスタンスごとの Ceph デーモンの開始、停止、および再起動」**セクションを参照してください。

9.3.5. オブジェクトマップヘッダーの操作

ceph-objectstore-tool ユーティリティーは、オブジェクトのキーに関連付けられた値と共にオブジェクトマップ (OMAP) ヘッダーを出力します。

注記

FileStore を OSD バックエンドオブジェクトストアとして使用する場合は、オブジェクトマップヘッダーの取得および設定時に **--journal-path PATH_TO_JOURNAL** に引数を追加します。**PATH_TO_JOURNAL** 変数は OSD ジャーナルへの絶対パスです (例: **/var/lib/ceph/osd/ceph-0/journal**)。

前提条件

- Ceph OSD ノードへのルートレベルのアクセス。
- **ceph-osd** デーモンの停止。

手順

1. 適切な OSD がダウンしていることを確認します。

```
[root@osd ~]# systemctl status ceph-osd@OSD_NUMBER
```

例

```
[root@osd ~]# systemctl status ceph-osd@1
```

- オブジェクトマップヘッダーを取得します。

```
[root@osd ~]# ceph-objectstore-tool --data-path PATH_TO_OSD \
--pgid PG_ID OBJECT \
get-omaphdr > OBJECT_MAP_FILE_NAME
```

例

```
[root@osd ~]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
--pgid 0.1c '{"oid":"zone_info.default","key":"","snapid":-
```

```
2,"hash":235010478,"max":0,"pool":11,"namespace":""}'  \
get-omaphdr > zone_info.default.omaphdr.txt
```

- オブジェクトマップヘッダーを設定します。

```
[root@osd ~]# ceph-objectstore-tool --data-path PATH_TO_OSD \
--pgid PG_ID OBJECT \
get-omaphdr < OBJECT_MAP_FILE_NAME
```

例

```
[root@osd ~]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
--pgid 0.1c '{"oid":"zone_info.default","key":"","snapid":-
2,"hash":235010478,"max":0,"pool":11,"namespace":""}'  \
set-omaphdr < zone_info.default.omaphdr.txt
```

- OSD の停止に関する詳細は、**『Red Hat Ceph Storage 管理ガイド』**の**「インスタンスごとの Ceph デーモンの開始、停止、および再起動」**セクションを参照してください。

9.3.6. オブジェクトマップキーの操作

ceph-objectstore-tool ユーティリティーを使用して、オブジェクトマップ (OMAP) キーを変更します。OMAP では、データパス、配置グループ識別子 (PG ID)、オブジェクト、およびキーを指定する必要があります。

注記

FileStore を OSD バックエンドオブジェクトストアとして使用する場合は、オブジェクトマップキーの取得、設定、または削除を行う際に **--journal-path PATH_TO_JOURNAL** を追加します。**PATH_TO_JOURNAL** 変数は OSD ジャーナルへの絶対パスです (例: **/var/lib/ceph/osd/ceph-0/journal**)。

前提条件

- Ceph OSD ノードへのルートレベルのアクセス。
- **ceph-osd** デーモンの停止。

手順

- オブジェクトマップキーを取得します。

```
[root@osd ~]# ceph-objectstore-tool --data-path PATH_TO_OSD \
--pgid PG_ID OBJECT \
get-omap KEY > OBJECT_MAP_FILE_NAME
```

例

```
[root@osd ~]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
--pgid 0.1c '{"oid":"zone_info.default","key":"","snapid":-
2,"hash":235010478,"max":0,"pool":11,"namespace":""}'  \
get-omap "" > zone_info.default.omap.txt
```

- オブジェクトマップキーを設定します。

```
[root@osd ~]# ceph-objectstore-tool --data-path PATH_TO_OSD \
--pgid PG_ID OBJECT \
set-omap KEY < OBJECT_MAP_FILE_NAME
```

例

```
[root@osd ~]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
--pgid 0.1c '{"oid":"zone_info.default","key":"","snapid":-
2,"hash":235010478,"max":0,"pool":11,"namespace":""}'  \
set-omap "" < zone_info.default.omap.txt
```

- オブジェクトマップキーを削除します。

```
[root@osd ~]# ceph-objectstore-tool --data-path PATH_TO_OSD \
--pgid PG_ID OBJECT \
rm-omap KEY
```

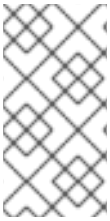
例

```
[root@osd ~]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
--pgid 0.1c '{"oid":"zone_info.default","key":"","snapid":-
2,"hash":235010478,"max":0,"pool":11,"namespace":""}'  \
rm-omap ""
```

- OSD の停止に関する詳細は、**『Red Hat Ceph Storage 管理ガイド』**の**「インスタンスごとの Ceph デーモンの開始、停止、および再起動」**セクションを参照してください。

9.3.7. オブジェクトの属性の一覧表示

ceph-objectstore-tool ユーティリティーを使用して、オブジェクトの属性を一覧表示します。この出力には、オブジェクトのキーと値が表示されます。



注記

FileStore を OSD バックエンドオブジェクトストアとして使用する場合は、オブジェクトの属性を一覧表示する際に **--journal-path PATH_TO_JOURNAL** 引数を追加します。**PATH_TO_JOURNAL** 変数は OSD ジャーナルへの絶対パスです (例: **/var/lib/ceph/osd/ceph-0/journal**)。

前提条件

- Ceph OSD ノードへのルートレベルのアクセス。
- **ceph-osd** デーモンの停止。

手順

1. 適切な OSD がダウンしていることを確認します。

```
[root@osd ~]# systemctl status ceph-osd@OSD_NUMBER
```

例

```
[root@osd ~]# systemctl status ceph-osd@1
```

2. オブジェクトの属性を一覧表示します。

```
ceph-objectstore-tool --data-path PATH_TO_OSD \
--pgid PG_ID OBJECT \
list-attrs
```

例

```
[root@osd ~]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
--pgid 0.1c '{"oid":"zone_info.default","key":"","snapid":-
2,"hash":235010478,"max":0,"pool":11,"namespace":""}' \
list-attrs
```

- OSD の停止に関する詳細は、『Red Hat Ceph Storage 管理ガイド』の「インスタンスごとの Ceph デーモンの開始、停止、および再起動」セクションを参照してください。

9.3.8. オブジェクト属性キーの操作

ceph-objectstore-tool ユーティリティーを使用してオブジェクトの属性を変更します。オブジェクトの属性を操作するには、オブジェクトの属性のデータとジャーナルパス、配置グループ識別子 (PG ID)、オブジェクト、およびキーが必要です。

注記

FileStore を OSD バックエンドオブジェクトストアとして使用する場合は、オブジェクトの属性の取得、設定、または削除を行う際に **--journal-path PATH_TO_JOURNAL** 引数を追加します。**PATH_TO_JOURNAL** 変数は OSD ジャーナルへの絶対パスです (例: **/var/lib/ceph/osd/ceph-0/journal**)。

前提条件

- Ceph OSD ノードへのルートレベルのアクセス。
- **ceph-osd** デーモンの停止。

手順

1. 適切な OSD がダウンしていることを確認します。

```
[root@osd ~]# systemctl status ceph-osd@OSD_NUMBER
```

例

```
[root@osd ~]# systemctl status ceph-osd@1
```

- オブジェクトの属性を取得します。

```
[root@osd ~]# ceph-objectstore-tool --data-path PATH_TO_OSD \
--pgid PG_ID OBJECT \
get-attrs KEY > OBJECT_ATTRS_FILE_NAME
```

例

```
[root@osd ~]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
--pgid 0.1c '{"oid":"zone_info.default","key":"","snapid":-
2,"hash":235010478,"max":0,"pool":11,"namespace":""}' \
get-attrs "oid" > zone_info.default.attr.txt
```

- オブジェクトの属性を設定します。

```
[root@osd ~]# ceph-objectstore-tool --data-path PATH_TO_OSD \
--pgid PG_ID OBJECT \
set-attrs KEY < OBJECT_ATTRS_FILE_NAME
```

例

```
[root@osd ~]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
--pgid 0.1c '{"oid":"zone_info.default","key":"","snapid":-
2,"hash":235010478,"max":0,"pool":11,"namespace":""}' \
set-attrs "oid" < zone_info.default.attr.txt
```

- オブジェクトの属性を削除します。

```
[root@osd ~]# ceph-objectstore-tool --data-path PATH_TO_OSD \
--pgid PG_ID OBJECT  \
rm-attrs KEY
```

例

```
[root@osd ~]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
--pgid 0.1c '{"oid":"zone_info.default","key":"","snapid":-
2,"hash":235010478,"max":0,"pool":11,"namespace":""}' \
rm-attrs "oid"
```

- OSD の停止に関する詳細は、**『Red Hat Ceph Storage 管理ガイド』**の**「インスタンスごとの Ceph デーモンの開始、停止、および再起動」**セクションを参照してください。

9.4. 関連情報

- Red Hat Ceph Storage のサポートについては、Red Hat カスタマーポータル を参照してください。

第10章 RED HAT サポートへのサービスの問い合わせ

本章では、本ガイドの情報で問題が解決しなかった場合に Red Hat のサポートサービスに連絡する方法を説明します。

10.1. 前提条件

- Red Hat サポートアカウント

10.2. RED HAT サポートエンジニアへの情報提供

Red Hat Ceph Storage に関連する問題を解決できない場合は、Red Hat サポートサービスに連絡し、サポートエンジニアが迅速にトラブルシューティングできるように多くの情報を提供します。

前提条件

- ノードへのルートレベルのアクセス。
- Red Hat サポートアカウント

手順

1. Red Hat カスタマーポータル でサポートチケットを作成します。
2. 理想的には、**sosreport** をチケットに割り当てます。詳細は、「Red Hat Enterprise Linux 4.6 以降での sosreport の役割と取得方法」を参照してください。
3. Ceph デーモンにセグメンテーション違反で失敗した場合には、人間が判読できるコアダンプファイルの生成を検討してください。詳細は「読み取り可能なコアダンプファイルの生成」を参照してください。

10.3. 判読可能なコアダンプファイルの生成

Ceph デーモンがセグメンテーション違反で突然終了した場合は、その障害に関する情報を収集し、Red Hat サポートエンジニアに提供します。

このような情報は初期調査を迅速化します。また、サポートエンジニアは、コアダンプファイルの情報を Red Hat Ceph Storage クラスターの既知の問題と比較できます。

10.3.1. 前提条件

1. **ceph-debuginfo** パッケージがインストールされていない場合はインストールします。
   a. **ceph-debuginfo** パッケージを含むリポジトリーを有効にします。

   Red Hat Enterprise Linux 7:

   ```
   subscription-manager repos --enable=rhel-7-server-rhceph-4-DAEMON-debug-rpms
   ```

   Ceph ノードの種別に応じて、**DAEMON** を **osd** または **mon** に置き換えます。

   Red Hat Enterprise Linux 8:

```
subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-debug-rpms
```

b. **ceph-debuginfo** パッケージをインストールします。

```
[root@mon ~]# yum install ceph-debuginfo
```

2. **gdb** パッケージがインストールされていることを確認します。インストールされていない場合は、インストールします。

```
[root@mon ~]# yum install gdb
```

デプロイメントのタイプに基づいて、手順を続けます。

- 「ベアメタルデプロイメントでの判読可能なコアダンプファイルの生成」
- 「コンテナー化されたデプロイメントでの判読可能なコアダンプファイルの生成」

10.3.2. ベアメタルデプロイメントでの判読可能なコアダンプファイルの生成

ベアメタルで Red Hat Ceph Storage を使用する場合、以下の手順に従ってコアダンプファイルを生成します。

手順

1. Ceph のコアダンプファイルの生成を有効にします。

a. **/etc/systemd/system.conf** ファイルに以下のパラメーターを追加して、コアダンプファイルに適切な **ulimits** を設定します。

```
DefaultLimitCORE=infinity
```

b. デフォルトでは、**/lib/systemd/system/CLUSTER_NAME-DAEMON@.service** にある Ceph デーモンサービスファイルの **PrivateTmp=true** パラメーターをコメントアウトします。

```
[root@mon ~]# PrivateTmp=true
```

c. **suid_dumpable** フラグを **2** に設定して、Ceph デーモンがダンプコアファイルを生成できるようにします。

```
[root@mon ~]# sysctl fs.suid_dumpable=2
```

d. コアダンプファイルの場所を調整します。

```
[root@mon ~]# sysctl kernel.core_pattern=/tmp/core
```

e. **systemd** サービスを再読み込みし、変更を反映します。

```
[root@mon ~]# systemctl daemon-reload
```

f. 変更を反映するために、Ceph デーモンを再起動します。

```
[root@mon ~]# systemctl restart ceph-DAEMON@ID
```

デーモンタイプ (**osd** または **mon**) とその ID (OSD の場合は数値、または Monitors の短縮ホスト名) を指定します。以下に例を示します。

```
[root@mon ~]# systemctl restart ceph-osd@1
```

2. デーモンを再度起動するなど、障害を再現します。
3. GNU Debugger (GDB) を使用して、アプリケーションのコアダンプファイルから判読可能なバックトレースを生成します。

```
gdb /usr/bin/ceph-DAEMON /tmp/core.PID
```

以下のように、デーモンのタイプと失敗したプロセスの PID を指定します。

```
$ gdb /usr/bin/ceph-osd /tmp/core.123456
```

GDB コマンドプロンプトで **set pag off** 子マンとおよび **set log on** コマンドを入力し、ページングを無効にし、ファイルへのロギングを有効にします。

```
(gdb) set pag off
(gdb) set log on
```

backtrace を入力して、**thr a a bt full** コマンドをプロセスのすべてのスレッドに適用します。

```
(gdb) thr a a bt full
```

バックトレースが生成されたら、**set log off** を入力して電源をオフにします。

```
(gdb) set log off
```

4. Red Hat カスタマーポータルにアクセスするシステムに **gdb.txt** ログファイルを転送して、サポートチケットにアタッチします。

10.3.3. コンテナー化されたデプロイメントでの判読可能なコアダンプファイルの生成

Red Hat Ceph Storage をコンテナーで使用する場合は、以下の手順に従ってコアダンプファイルを生成します。この手順では、コアダンプファイルを取得する 2 つのシナリオが関係します。

- SIGILL、SIGTRAP、SIGABRT、または SIGSEGV エラーにより、Ceph プロセスが予期せず終了した場合。

または

- 手動の場合。たとえば、Ceph プロセスが高い CPU サイクルを消費したり、応答がないなど、問題を手動でデバッグする場合。

前提条件

- Ceph コンテナーを実行するコンテナーノードへの root レベルのアクセス。
- 適切なデバッグパッケージのインストール

- GNU Project Debugger (**gdb**) パッケージのインストール。

手順

1. SIGILL、SIGTRAP、SIGABRT、または SIGSEGV エラーにより、Ceph プロセスが予期せず終了した場合。

   a. 障害の発生した Ceph プロセスのあるコンテナーが実行しているノードの **systemd-coredump** サービスにコアパターンを設定します。以下に例を示します。

   ```
   [root@mon]# echo "| /usr/lib/systemd/systemd-coredump %P %u %g %s %t %e" >
   /proc/sys/kernel/core_pattern
   ```

   b. Ceph プロセスが原因でコンテナーに関する次の障害の有無を確認し、**/var/lib/systemd/coredump/** ディレクトリーでコアダンプファイルを検索します。以下に例を示します。

   ```
   [root@mon]# ls -ltr /var/lib/systemd/coredump
   total 8232
   -rw-r-----. 1 root root 8427548 Jan 22 19:24 core.ceph-
   osd.167.5ede29340b6c4fe4845147f847514c12.15622.1584573794000000.xz
   ```

2. **Ceph Monitors** および **Ceph Managers** のコアダンプファイルを手動でキャプチャーするには、以下を実行します。

   a. コンテナーから Ceph デーモンの **ceph-mon** パッケージの詳細を取得します。

   Red Hat Enterprise Linux 7:

   ```
   [root@mon]# docker exec -it NAME /bin/bash
   [root@mon]# rpm -qa | grep ceph
   ```

   Red Hat Enterprise Linux 8:

   ```
   [root@mon]# podman exec -it NAME /bin/bash
   [root@mon]# rpm -qa | grep ceph
   ```

   NAME を、Ceph コンテナーの名前に置き換えます。

   b. バックアップコピーを作成し、**ceph-mon@.service** ファイルを編集するために開きます。

   ```
   [root@mon]# cp /etc/systemd/system/ceph-mon@.service /etc/systemd/system/ceph-
   mon@.service.orig
   ```

   c. **ceph-mon@.service** ファイルで、これらの 3 つのオプションを **[Service]** セクションに追加します。各オプションは 1 行ずつ追加します。

   ```
   --pid=host \
   --ipc=host \
   --cap-add=SYS_PTRACE \
   ```

   例

```
[Unit]
Description=Ceph Monitor
After=docker.service

[Service]
EnvironmentFile=-/etc/environment
ExecStartPre=-/usr/bin/docker rm ceph-mon-%i
ExecStartPre=/bin/sh -c '"$(command -v mkdir)" -p /etc/ceph /var/lib/ceph/mon'
ExecStart=/usr/bin/docker run --rm --name ceph-mon-%i \
  --memory=924m \
--cpu-quota=100000 \
-v /var/lib/ceph:/var/lib/ceph:z \
  -v /etc/ceph:/etc/ceph:z \
  -v /var/run/ceph:/var/run/ceph:z \
-v /etc/localtime:/etc/localtime:ro \
--net=host \
--privileged=true \
--ipc=host \ 1
--pid=host \ 2
--cap-add=SYS_PTRACE \ 3
-e IP_VERSION=4 \
        -e MON_IP=10.74.131.17 \
      -e CLUSTER=ceph \
  -e FSID=9448efca-b1a1-45a3-bf7b-b55cba696a6e \
  -e CEPH_PUBLIC_NETWORK=10.74.131.0/24 \
  -e CEPH_DAEMON=MON \
   \
  registry.access.redhat.com/rhceph/rhceph-3-rhel7:latest
ExecStop=-/usr/bin/docker stop ceph-mon-%i
ExecStopPost=-/bin/rm -f /var/run/ceph/ceph-mon.pd-cephcontainer-mon01.asok
Restart=always
RestartSec=10s
TimeoutStartSec=120
TimeoutStopSec=15

[Install]
WantedBy=multi-user.target
```

d. Ceph Monitor デーモンを再起動します。

構文

```
systemctl restart ceph-mon@MONITOR_ID
```

MONITOR_ID を Ceph Monitor の ID 番号に置き換えます。

例

```
[root@mon]# systemctl restart ceph-mon@1
```

e. Ceph Monitor コンテナーに **gdb** パッケージをインストールします。

Red Hat Enterprise Linux 7:

```
[root@mon]# docker exec -it ceph-mon-MONITOR_ID /bin/bash
sh $ yum install gdb
```

Red Hat Enterprise Linux 8:

```
[root@mon]# podman exec -it ceph-mon-MONITOR_ID /bin/bash
sh $ yum install gdb
```

MONITOR_ID を Ceph Monitor の ID 番号に置き換えます。

f. プロセス ID を検索します。

構文

```
ps -aef | grep PROCESS | grep -v run
```

PROCESS は、失敗したプロセス名 (例: **ceph-mon**) に置き換えます。

例

```
[root@mon]# ps -aef | grep ceph-mon | grep -v run
ceph       15390   15266  0 18:54 ?        00:00:29 /usr/bin/ceph-mon --cluster ceph --
setroot ceph --setgroup ceph -d -i 5
ceph       18110   17985  1 19:40 ?        00:00:08 /usr/bin/ceph-mon --cluster ceph --
setroot ceph --setgroup ceph -d -i 2
```

g. コアダンプファイルを生成します。

構文

```
gcore ID
```

ID を、前の手順で取得した失敗したプロセスの ID に置き換えます (例: **18110**)。

例

```
[root@mon]# gcore 18110
warning: target file /proc/18110/cmdline contained unexpected null characters
Saved corefile core.18110
```

h. コアダンプファイルが正しく生成されたことを確認します。

例

```
[root@mon]# ls -ltr
total 709772
-rw-r--r--. 1 root root 726799544 Mar 18 19:46 core.18110
```

i. Ceph Monitor コンテナー外部でコアダンプファイルをコピーします。

Red Hat Enterprise Linux 7:

```
[root@mon]# docker cp ceph-mon-MONITOR_ID:/tmp/mon.core.MONITOR_PID /tmp
```

Red Hat Enterprise Linux 8:

```
[root@mon]# podman cp ceph-mon-MONITOR_ID:/tmp/mon.core.MONITOR_PID /tmp
```

MONITOR_ID を Ceph Monitor の ID 番号に置き換え、 **MONITOR_PID** をプロセス ID 番号に置き換えます。

j. **ceph-mon@.service** ファイルのバックアップコピーを復元します。

```
[root@mon]# cp /etc/systemd/system/ceph-mon@.service.orig
/etc/systemd/system/ceph-mon@.service
```

k. Ceph Monitor デーモンを再起動します。

構文

```
systemctl restart ceph-mon@MONITOR_ID
```

MONITOR_ID を Ceph Monitor の ID 番号に置き換えます。

例

```
[root@mon]# systemctl restart ceph-mon@1
```

l. Red Hat サポートが分析するコアダンプファイルをアップロードする場合は、ステップ 4 を参照してください。

3. **Ceph OSD** のコアダンプファイルを手動でキャプチャーするには、以下を実行します。

a. コンテナーから Ceph デーモンの **ceph-osd** パッケージの詳細を取得します。

Red Hat Enterprise Linux 7:

```
[root@osd]# docker exec -it NAME /bin/bash
[root@osd]# rpm -qa | grep ceph
```

Red Hat Enterprise Linux 8:

```
[root@osd]# podman exec -it NAME /bin/bash
[root@osd]# rpm -qa | grep ceph
```

NAME を、Ceph コンテナーの名前に置き換えます。

b. Ceph コンテナーが実行しているノードに、同じバージョンの **ceph-osd** パッケージ用の Ceph パッケージをインストールします。

Red Hat Enterprise Linux 7:

```
[root@osd]# yum install ceph-osd
```

Red Hat Enterprise Linux 8:

```
[root@osd]# dnf install ceph-osd
```

必要に応じて、適切なリポジトリーを最初に有効にします。詳細は、『インストールガイド』の**「Red Hat Ceph Storage の有効化」**セクションを参照してください。

c. 障害が発生したプロセスの ID を検索します。

```
ps -aef | grep PROCESS | grep -v run
```

PROCESS は、失敗したプロセス名 (例: **ceph-osd**) に置き換えます。

```
[root@osd]# ps -aef | grep ceph-osd | grep -v run
ceph       15390   15266  0 18:54 ?        00:00:29 /usr/bin/ceph-osd --cluster ceph --
setroot ceph --setgroup ceph -d -i 5
ceph       18110   17985  1 19:40 ?        00:00:08 /usr/bin/ceph-osd --cluster ceph --
setroot ceph --setgroup ceph -d -i 2
```

d. コアダンプファイルを生成します。

```
gcore ID
```

ID を、前の手順で取得した失敗したプロセスの ID に置き換えます (例: **18110**)。

```
[root@osd]# gcore 18110
warning: target file /proc/18110/cmdline contained unexpected null characters
Saved corefile core.18110
```

e. コアダンプファイルが正しく生成されたことを確認します。

```
[root@osd]# ls -ltr
total 709772
-rw-r--r--. 1 root root 726799544 Mar 18 19:46 core.18110
```

f. Red Hat サポートが分析するコアダンプファイルをアップロードする場合は、次のステップを参照してください。

4. Red Hat サポートケースに分析用のコアダンプファイルをアップロードします。詳細は、**「Red Hat サポートエンジニアへの情報の提供」**を参照してください。

10.3.4. 関連情報

- Red Hat Customer Portal の「gdb を使用して、アプリケーションコアから読み取り可能なバックトレースを生成する方法」
- Red Hat カスタマーポータルの「アプリケーションがクラッシュまたはセグメンテーション違反が発生した時にコアファイルのダンプを有効にする」

付録A CEPH サブシステムのデフォルトログレベルの値

さまざまな Ceph サブシステムにおけるデフォルトのログレベル値の表

サブシステム	ログレベル	メモリーレベル
asok	1	5
auth	1	5
buffer	0	0
client	0	5
context	0	5
crush	1	5
default	0	5
filer	0	5
bluestore	1	5
finisher	1	5
heartbeatmap	1	5
javaclient	1	5
journaler	0	5
journal	1	5
lockdep	0	5
mds balancer	1	5
mds locker	1	5
mds log expire	1	5
mds log	1	5
mds migrator	1	5
mds	1	5

サブシステム	ログレベル	メモリーレベル
monc	0	5
mon	1	5
ms	0	5
objclass	0	5
objectcacher	0	5
objecter	0	0
optracker	0	5
osd	0	5
paxos	0	5
perfcounter	1	5
rados	0	5
rbd	0	5
rgw	1	5
throttle	1	5
timer	0	5
tp	0	5