Red Hat Ceph Storage 7

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: 2024-03-17

Red Hat Ceph Storage 7 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

法律上の通知

概要

本書では、Red Hat Ceph Storage に関する一般的な問題を解決する方法を説明します。 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 初期のトラブルシューティング

ストレージ管理者であれば、Red Hat のサポートに連絡する前に、Red Hat Ceph Storage クラスターの初期トラブルシューティングを行うことができます。この章では、以下の内容をご紹介します。

- 問題の特定
- ストレージクラスターの正常性の診断
- Ceph の正常性の理解
- Ceph クラスターの正常性アラートのミュート
- Ceph ログについて
- `sos report` の生成

前提条件

- 稼働中の Red Hat Ceph Storage クラスターがある。

1.1. 問題の特定

Red Hat Ceph Storage クラスターでエラーの原因を確認するには、手順についてのセクションにある質問に回答します。

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場合は、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てください。
2. どの Ceph コンポーネントが問題を引き起こすかについて把握しているか？
   a. いいえ。**Red Hat Ceph Storage トラブルシューティングガイド**の 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 でサポートされる設定 について参照してください。

1.2. ストレージクラスターの正常性の診断

以下の手順では、Red Hat Ceph Storage クラスターの正常性を診断するための基本的な手順を紹介します。

前提条件

- 稼働中の Red Hat Ceph Storage クラスターがある。

手順

1. Cephadm シェルにログインします。

   例

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

2. 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。

   例

   ```
   [ceph: root@host01 /]# 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。

3. ストレージクラスターのログを監視します。

   例

   ```
   [ceph: root@host01 /]# ceph -W cephadm
   ```

4. クラスターのログをファイルに取り込むには、以下のコマンドを実行します。

   例

   ```
   [ceph: root@host01 /]# ceph config set global log_to_file true
   [ceph: root@host01 /]# ceph config set global mon_cluster_log_to_file true
   ```

   デフォルトでは、ログは **/var/log/ceph/CLUSTER_FSID/** ディレクトリーにあります。**Ceph ログについて** に記載されているエラーメッセージがないか、Ceph ログを確認します。

5. 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。

注記

実行中の **mgr** デーモンがない場合は、ヘルス警告が表示されます。Red Hat Ceph Storage クラスターの最後の **mgr** デーモンが削除された場合は、Red Hat Storage クラスターのランダムホストに **mgr** デーモンを手動でデプロイできます。**Red Hat Ceph Storage 7 管理ガイド** の mgr デーモンの手動デプロイ を参照してください。

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 クラスターの正常性アラートのミュート

特定のシナリオでは、ユーザーが一時的にいくつかの警告をミュートしたい場合があります。正常性チェックをミュートして、Ceph クラスターの報告されたステータス全体に影響を与えないようにすることができます。

アラートは正常性チェックコードで指定します。たとえば、OSD がメンテナンスのためにダウンした場合は、**OSD_DOWN** 警告が出力されることがあります。メンテナンスが終了するまで警告をミュートにすることもできます。これらの警告が出ると、メンテナンス期間中、クラスターは **HEALTH_OK** ではなく **HEALTH_WARN** になります。

アラートの範囲が悪化すると、ほとんどの正常性ミュートも消えます。たとえば、1 つの OSD がダウンしていて、アラートがミュートになっている場合、さらに 1 つ以上の OSD がダウンすると、ミュートが消えます。これは、警告やエラーの原因となっているものの量や数を示すカウントを伴う正常性アラートに当てはまります。

前提条件

- 稼働中の Red Hat Ceph Storage クラスターがある。
- ノードへのアクセスのルートレベル。
- 正常性への警告メッセージです。

手順

1. Cephadm シェルにログインします。

例

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

2. **ceph health detail** コマンドを実行して、Red Hat Ceph Storage クラスターの正常性を確認します。

例

```
[ceph: root@host01 /]# ceph health detail

HEALTH_WARN 1 osds down; 1 OSDs or CRUSH {nodes, device-classes} have
{NOUP,NODOWN,NOIN,NOOUT} flags set
[WRN] OSD_DOWN: 1 osds down
    osd.1 (root=default,host=host01) is down
[WRN] OSD_FLAGS: 1 OSDs or CRUSH {nodes, device-classes} have
{NOUP,NODOWN,NOIN,NOOUT} flags set
    osd.1 has flags noup
```

ストレージクラスターは、OSD の 1 つがダウンしているため、**HEALTH_WARN** 状態になっていることがわかります。

3. アラートをミュートします。

構文

```
ceph health mute HEALTH_MESSAGE
```

例

```
[ceph: root@host01 /]# ceph health mute OSD_DOWN
```

4. オプション: 正常性チェックのミュートに TTL (time to live) を設定することができ、指定した時間が経過するとミュートが自動的に失効します。コマンドの任意の duration 引数として TTL を指定します。

構文

```
ceph health mute HEALTH_MESSAGE DURATION
```

DURATION は、**s**、**sec**、**m**、**min**、**h**、**hour** で指定できます。

例

```
[ceph: root@host01 /]# ceph health mute OSD_DOWN 10m
```

この例では、アラート **OSD_DOWN** が 10 分間ミュートされます。

5. Red Hat Ceph Storage クラスターのステータスが **HEALTH_OK** に変更されているかどうかを確認します。

例

```
[ceph: root@host01 /]# ceph -s
  cluster:
    id:     81a4597a-b711-11eb-8cb8-001a4a000740
    health: HEALTH_OK
            (muted: OSD_DOWN(9m) OSD_FLAGS(9m))

  services:
    mon: 3 daemons, quorum host01,host02,host03 (age 33h)
    mgr: host01.pzhfuh(active, since 33h), standbys: host02.wsnngf, host03.xwzphg
    osd: 11 osds: 10 up (since 4m), 11 in (since 5d)

  data:
    pools:   1 pools, 1 pgs
    objects: 13 objects, 0 B
    usage:   85 MiB used, 165 GiB / 165 GiB avail
    pgs:     1 active+clean
```

この例では、**OSD_DOWN** および **OSD_FLAG** の警告がミュートされ、そのミュートが 9 分間有効であることがわかります。

6. オプション: ミュートを **スティッキー** にすることで、アラートが解除された後もミュートを保持することができます。

   構文

   ```
   ceph health mute HEALTH_MESSAGE DURATION --sticky
   ```

   例

   ```
   [ceph: root@host01 /]# ceph health mute OSD_DOWN 1h --sticky
   ```

7. 次のコマンドを実行して、ミュートを削除できます。

   構文

   ```
   ceph health unmute HEALTH_MESSAGE
   ```

   例

   ```
   [ceph: root@host01 /]# ceph health unmute OSD_DOWN
   ```

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 **Ceph クラスターの正常性メッセージ** セクションを参照してください。

1.5. CEPH ログについて

ファイルへのロギングが有効になると、Ceph はログを **/var/log/ceph/CLUSTER_FSID/** 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 **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** の **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。
- ファイルへのロギングを有効にするには、**Ceph デーモンログ** を参照してください。

1.6. SOS REPORT の作成

sos report コマンドを実行すると、Red Hat Enterprise Linux から Red Hat Ceph Storage クラスターの設定詳細、システム情報、診断情報を収集することができます。Red Hat サポートチームは、この情報をストレージクラスターのさらなるトラブルシューティングに使用します。

前提条件

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

手順

1. **sos** パッケージをインストールするには、以下のコマンドを実行します。

   例

   ```
   [root@host01 ~]# dnf install sos
   ```

2. **sos report** を実行して、ストレージクラスターのシステム情報を取得します。

   例

   ```
   [root@host01 ~]# sosreport -a --all-logs
   ```

   レポートは **/var/tmp** ファイルに保存されます。

   特定の Ceph デーモン情報を取得するには、次のコマンドを実行します。

   例

```
[root@host01 ~]# sos report --all-logs -e
ceph_mgr,ceph_common,ceph_mon,ceph_osd,ceph_ansible,ceph_mds,ceph_rgw
```

関連情報

- What is an sosreport and how to create one in Red Hat Enterprise Linuxを参照してください。詳細は、ナレッジベースの記事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。

前提条件

- 稼働中の Red Hat Ceph Storage クラスターがある。

2.1. 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設定されます。

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/CLUSTER_FSID/** 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	説明
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
ミリ秒	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

```
2022-05-12 12:37:04.278761 7f45a9afc700 10 mon.cephn2@0(leader).osd e322 e322: 2 osds: 2 up,
2 in
2022-05-12 12:37:04.278792 7f45a9afc700 10 mon.cephn2@0(leader).osd e322
min_last_epoch_clean 322
2022-05-12 12:37:04.278795 7f45a9afc700 10 mon.cephn2@0(leader).log v1010106 log
2022-05-12 12:37:04.278799 7f45a9afc700 10 mon.cephn2@0(leader).auth v2877 auth
2022-05-12 12:37:04.278811 7f45a9afc700 20 mon.cephn2@0(leader) e1 sync_trim_providers
2022-05-12 12:37:09.278914 7f45a9afc700 11 mon.cephn2@0(leader) e1 tick
2022-05-12 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
2022-05-12 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
2022-05-12 12:37:09.278982 7f45a9afc700 10 mon.cephn2@0(leader).osd e322 e322: 2 osds: 2 up,
2 in
2022-05-12 12:37:09.278989 7f45a9afc700  5 mon.cephn2@0(leader).paxos(paxos active c
1028850..1029466) is_readable = 1 - now=2021-08-12 12:37:09.278990 lease_expire=0.000000 has
v0 lc 1029466
....
2022-05-12 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
2022-05-12 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
2022-05-12 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
....
2022-05-12 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

```
2022-05-12 11:27:53.869151 7f5d55d84700  1 -- 192.168.17.3:0/2410 --> 192.168.17.4:6801/2801 --
osd_ping(ping e322 stamp 2021-08-12 11:27:53.869147) v2 -- ?+0 0x63baa00 con 0x578dee0
2022-05-12 11:27:53.869214 7f5d55d84700  1 -- 192.168.17.3:0/2410 --> 192.168.0.114:6801/2801
-- osd_ping(ping e322 stamp 2021-08-12 11:27:53.869147) v2 -- ?+0 0x638f200 con 0x578e040
2022-05-12 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 2021-08-12
```

```
11:27:53.869147) v2 ==== 47+0+0 (261193640 0 0) 0x63c1a00 con 0x578e040
2022-05-12 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 2021-08-12
11:27:53.869147) v2 ==== 47+0+0 (261193640 0 0) 0x6313200 con 0x578dee0
....
2022-05-12 11:28:10.432313 7f5d6e71f700  5 osd.0 322 tick
2022-05-12 11:28:10.432375 7f5d6e71f700 20 osd.0 322 scrub_random_backoff lost coin flip,
randomly backing off
2022-05-12 11:28:10.432381 7f5d6e71f700 10 osd.0 322 do_waiters -- start
2022-05-12 11:28:10.432383 7f5d6e71f700 10 osd.0 322 do_waiters -- finish
```

関連情報

- 実行時のロギング設定
- 設定ファイルでのロギングの設定

2.2. 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

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

構文

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

例

```
[ceph: root@host01 /]# 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 7 の **設定ガイド** の Ceph のデバッグおよびログ設定リファレンス の章

2.3. 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
- 実行時のロギング設定
- Red Hat Ceph Storage 7 の **設定ガイド** の Ceph のデバッグおよびログ設定リファレンス の章

2.4. 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
```

2.5. CEPH OBJECT GATEWAY の操作ログの作成と収集

ユーザー識別情報が操作ログ出力に追加されます。これは、顧客が S3 アクセスの監査のためにこの情報にアクセスできるようにするために使用されます。Ceph Object Gateway の操作ログのすべてのバージョンで、S3 リクエストによってユーザー ID を確実に追跡します。

手順

1. ログの場所を見つけます。

   構文

   ```
   logrotate -f
   ```

   例

   ```
   [root@host01 ~]# logrotate -f
   /etc/logrotate.d/ceph-12ab345c-1a2b-11ed-b736-fa163e4f6220
   ```

2. 指定された場所内のログをリスト表示します。

   構文

   ```
   ll LOG_LOCATION
   ```

   例

   ```
   [root@host01 ~]# ll /var/log/ceph/12ab345c-1a2b-11ed-b736-fa163e4f6220
    -rw-r--r--. 1 ceph ceph    412 Sep 28 09:26 opslog.log.1.gz
   ```

3. 現在のバケットをリストします。

   例

   ```
   [root@host01 ~]# /usr/local/bin/s3cmd ls
   ```

4. バケットを作成します。

   構文

   ```
   /usr/local/bin/s3cmd mb s3://NEW_BUCKET_NAME
   ```

   例

   ```
   [root@host01 ~]# /usr/local/bin/s3cmd mb s3://bucket1
   Bucket `s3://bucket1` created
   ```

5. 現在のログをリスト表示します。

   構文

   ```
   ll LOG_LOCATION
   ```

例

```
[root@host01 ~]# ll /var/log/ceph/12ab345c-1a2b-11ed-b736-fa163e4f6220
 total 852
 ...
 -rw-r--r--. 1 ceph ceph    920 Jun 29 02:17 opslog.log
 -rw-r--r--. 1 ceph ceph    412 Jun 28 09:26 opslog.log.1.gz
```

6. ログを収集します。

構文

```
tail -f LOG_LOCATION/opslog.log
```

例

```
[root@host01 ~]# tail -f /var/log/ceph/12ab345c-1a2b-11ed-b736-fa163e4f6220/opslog.log

{"bucket":"","time":"2022-09-29T06:17:03.133488Z","time_local":"2022-09-
29T06:17:03.133488+0000","remote_addr":"10.0.211.66","user":"test1",
"operation":"list_buckets","uri":"GET /
HTTP/1.1","http_status":"200","error_code":"","bytes_sent":232,
"bytes_received":0,"object_size":0,"total_time":9,"user_agent":"","referrer":
"","trans_id":"tx00000c80881a9acd2952a-006335385f-175e5-primary",
"authentication_type":"Local","access_key_id":"1234","temp_url":false}

{"bucket":"cn1","time":"2022-09-29T06:17:10.521156Z","time_local":"2022-09-
29T06:17:10.521156+0000","remote_addr":"10.0.211.66","user":"test1",
"operation":"create_bucket","uri":"PUT /cn1/
HTTP/1.1","http_status":"200","error_code":"","bytes_sent":0,
"bytes_received":0,"object_size":0,"total_time":106,"user_agent":"",
"referrer":"","trans_id":"tx0000058d60c593632c017-0063353866-175e5-primary",
"authentication_type":"Local","access_key_id":"1234","temp_url":false}
```

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) の chrony に接続するトラブルシューティング手順を説明します。

前提条件

- 稼働中の Red Hat Ceph Storage クラスターがある。

3.1. 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。

   例

   ```
   [root@host01 ~]# dnf install net-tools
   [root@host01 ~]# dnf install telnet
   ```

2. **cephadm** シェルにログインし、Ceph 設定ファイルの **public_network** パラメーターに正しい値が含まれていることを確認します。

   例

   ```
   [ceph: root@host01 /]# cat /etc/ceph/ceph.conf
   # minimal ceph.conf for 57bddb48-ee04-11eb-9962-001a4a000672
   [global]
    fsid = 57bddb48-ee04-11eb-9962-001a4a000672
    mon_host = [v2:10.74.249.26:3300/0,v1:10.74.249.26:6789/0]
   [v2:10.74.249.163:3300/0,v1:10.74.249.163:6789/0]
   [v2:10.74.254.129:3300/0,v1:10.74.254.129:6789/0]
   [mon.host01]
   public network = 10.74.248.0/21
   ```

3. シェルを終了し、ネットワークインターフェイスが起動していることを確認します。

   例

   ```
   [root@host01 ~]# 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: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode
DEFAULT group default qlen 1000
    link/ether 00:1a:4a:00:06:72 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@host01 ~]# ping host02
   ```

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お互いに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@host01 ~]# firewall-cmd --info-zone=public
   public (active)
     target: default
     icmp-block-inversion: no
     interfaces: ens3
     sources:
     services: ceph ceph-mon cockpit dhcpv6-client ssh
     ports: 9283/tcp 8443/tcp 9093/tcp 9094/tcp 3000/tcp 9100/tcp 9095/tcp
     protocols:
     masquerade: no
     forward-ports:
     source-ports:
     icmp-blocks:
     rich rules:

   [root@host01 ~]# telnet 192.168.0.22 9100
   ```

6. インターフェイ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@host01 ~]# ethtool -S ens3 | 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@host01 ~]# ifconfig
ens3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.74.249.26  netmask 255.255.248.0  broadcast 10.74.255.255
        inet6 fe80::21a:4aff:fe00:672  prefixlen 64  scopeid 0x20<link>
        inet6 2620:52:0:4af8:21a:4aff:fe00:672  prefixlen 64  scopeid 0x0<global>
        ether 00:1a:4a:00:06:72  txqueuelen 1000  (Ethernet)
        RX packets 150549316  bytes 56759897541 (52.8 GiB)
        RX errors 0  dropped 176924  overruns 0  frame 0
        TX packets 55584046  bytes 62111365424 (57.8 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 9373290  bytes 16044697815 (14.9 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9373290  bytes 16044697815 (14.9 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

c. **netstat** コマンドの使用:

例

```
[root@host01 ~]# netstat -ai
Kernel Interface table
Iface             MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-
OVR Flg
ens3             1500 311847720      0 364903 0      114341918      0      0      0 BMRU
lo              65536 19577001       0      0 0       19577001       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。

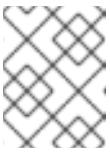
例

```
[root@host01 ~]# dnf install iperf3
```

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

例

```
[root@host01 ~]# 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@host02 ~]# 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@host01 ~]# ethtool ens3
Settings for ens3:
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。
- 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。
- 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。

3.2. 基本的な CHRONY 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 chrony NTP のトラブルシューティング手順を説明します。

前提条件

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

手順

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

   例

   ```
   [root@mon ~]# systemctl status chronyd
   ```

2. **chronyd** が実行されていない場合は、有効にして起動します。

   例

   ```
   [root@mon ~]# systemctl enable chronyd
   [root@mon ~]# systemctl start chronyd
   ```

3. **chronyd** がクロックを正しく同期していることを確認します。

   例

   ```
   [root@mon ~]# chronyc sources
   [root@mon ~]# chronyc sourcestats
   [root@mon ~]# chronyc tracking
   ```

関連情報

- 高度な chrony NTP の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 How to troubleshoot chrony issues 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。
- 詳細は、Checking if chrony is synchronized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。

前提条件

- ネットワーク接続の検証。

4.1. 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リスト表示しています。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。

前提条件

- 稼働中の Red Hat Ceph Storage クラスターがある。

4.1.1. 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

4.1.2. 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

4.1.3. 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、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。

   構文

```
systemctl status ceph-FSID@DAEMON_NAME
systemctl start ceph-FSID@DAEMON_NAME
```

例

```
[root@mon ~]# systemctl status ceph-b404c440-9e4c-11ec-a28a-
001a4a0001df@mon.host01.service
[root@mon ~]# systemctl start ceph-b404c440-9e4c-11ec-a28a-
001a4a0001df@mon.host01.service
```

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

注記

デフォルトでは、モニターログはログフォルダーに表示されません。ログがフォルダーに表示されるようにするには、ファイルへのロギングを有効にする必要があります。ファイルへのロギングを有効にするには、Ceph デーモンログ を参照してください。

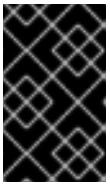
2. ログに以下のようなエラーメッセージが含まれる場合、Ceph Monitor のストアが破損している可能性があります。

```
Corruption: error in middle of record
Corruption: 1 missing files; example: /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。

```
[ceph: root@host01 /]# ceph daemon mon.host01 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。

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 管理ガイド**の **Ceph デーモンの開始、停止、再始動** セクション。
- **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

4.1.4. 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。

```
2022-05-04 07:28:32.035795 7f806062e700 0 log [WRN] : mon.a 127.0.0.1:6789/0 clock skew 0.14s
> max 0.05s
2022-05-04 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。

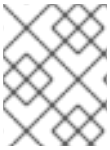
重要

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 chrony 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。
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 chrony NTP サーバーをデプロイすることを検討してください。詳細については、Red Hat Customer Portal にある、お使いの OS バージョン の製品ドキュメント (OS バージョンの os-product) 内の **基本システム設定** ガイドの **Chrony Suite を使用した NTP の設定** の章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

4.1.5. 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 RocksDB 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/CLUSTER_FSID/mon.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の詳細は、**Ceph Monitor がクォーラム外である** 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。

   構文

   ```
   du -sch /var/lib/ceph/CLUSTER_FSID/mon.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

   ```
   [root@mon ~]# du -sh  /var/lib/ceph/b341e254-b165-11ed-a564-ac1f6bb26e8c/mon.host01/
   109M /var/lib/ceph/b341e254-b165-11ed-a564-ac1f6bb26e8c/mon.host01/
   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

4.1.6. 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。

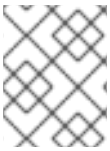
関連情報

- 詳細は、Red Hat Ceph Storage 7 の **管理ガイド**の **Ceph 管理ソケットの使用** を参照してください。
- **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。

4.2. 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。

   例

   ```
   [ceph: root@host01 /]# 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。

      構文

      ```
      systemctl stop ceph-FSID@DAEMON_NAME
      ```

      例

```
[root@mon ~]# systemctl stop ceph-b404c440-9e4c-11ec-a28a-
001a4a0001df@mon.host01.service
```

b. Ceph Monitor マップをコピーします。

構文

```
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。

例

```
[ceph: root@host01 /]#  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。

構文

```
systemctl stop ceph-FSID@DAEMON_NAME
```

例

```
[root@mon ~]# systemctl stop ceph-b404c440-9e4c-11ec-a28a-
001a4a0001df@mon.host01.service
```

4. Ceph Monitor マップを注入します。

構文

```
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。

例

```
[root@mon ~]# ceph-mon -i mon.host01 --inject-monmap /tmp/monmap
```

5. Ceph Monitor を起動します。

構文

```
systemctl start ceph-FSID@DAEMON_NAME
```

例

```
[root@mon ~]# systemctl start ceph-b404c440-9e4c-11ec-a28a-
001a4a0001df@mon.host01.service
```

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。

構文

```
systemctl start ceph-FSID@DAEMON_NAME
```

例

```
[root@mon ~]# systemctl start ceph-b404c440-9e4c-11ec-a28a-
001a4a0001df@mon.host01.service
```

関連情報

- Ceph モニターがクォーラム外にある を参照してください。
- Ceph Monitor ストアのリカバリー を参照してください

4.3. 失敗したモニターの置き換え

Ceph Monitor のストアが破損している場合は、ストレージクラスター内のモニターを交換できます。

前提条件

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

   ```
   [ceph: root@host01 /]# ceph mon remove host01 --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。

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。

4.4. 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

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

手順

1. **ceph-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。

   構文

   ```
   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。

   例

   ```
   [ceph: root@host01 /]# ceph tell mon.host01 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。

   構文

```
systemctl restart ceph-FSID@DAEMON_NAME
```

例

```
[root@mon ~]# systemctl restart ceph-b404c440-9e4c-11ec-a28a-
001a4a0001df@mon.host01.service
```

5. Monitor がクォーラムを形成することを確認します。

```
[ceph: root@host01 /]# 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。

構文

```
systemctl status ceph-FSID@DAEMON_NAME
systemctl stop ceph-FSID@DAEMON_NAME
```

例

```
[root@mon ~]# systemctl status ceph-b404c440-9e4c-11ec-a28a-
001a4a0001df@mon.host01.service
[root@mon ~]# systemctl stop ceph-b404c440-9e4c-11ec-a28a-
001a4a0001df@mon.host01.service
```

8. Monitor ストアを圧縮します。

構文

```
ceph-monstore-tool /var/lib/ceph/CLUSTER_FSID/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。

例

```
[ceph: root@host01 /]# ceph-monstore-tool /var/lib/ceph/b404c440-9e4c-11ec-a28a-
001a4a0001df/mon.host01 compact
```

9. **ceph-mon** を再度起動します。

構文

```
systemctl start ceph-FSID@DAEMON_NAME
```

例

```
[root@mon ~]# systemctl start ceph-b404c440-9e4c-11ec-a28a-
001a4a0001df@mon.host01.service
```

関連情報

- Ceph Monitor ストアが大きくなりすぎている を確認してください
- Ceph モニターがクォーラム外にある を参照してください。

4.5. 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** にデプロイメントし、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

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

2. **ceph-mgr** デーモンを再起動します。

4.6. CEPH MONITOR ストアのリカバリー

Ceph Monitor は、クラスターマップを RocksDB 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
```

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。

4.6.1. 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

- 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. 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. **ceph-volume** コマンドを使用してデータパーティションをリスト表示します。

例

```
[ceph: root@host01 /]# ceph-volume lvm 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。

構文

```
mount -t tmpfs tmpfs /var/lib/ceph/osd/ceph-$i
```

c. SELinux コンテキストを復元します。

構文

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

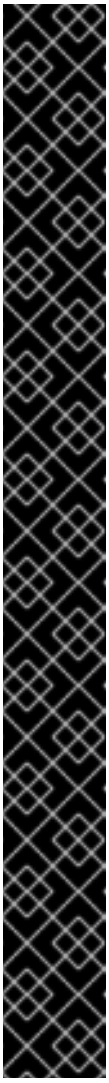
OSD_ID を、OSD ノード上の Ceph OSD ID の数値のスペース区切りリストに置き換えます。

d. 所有者とグループを **ceph:ceph** に変更します。

構文

```
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。これを行うには、以下を行います。

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 マップをデプロイメントします。

   構文

   ```
   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。

   構文

   ```
   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。

   例

```
[root@host01 ~]# cd /root/
[root@host01 ~]# ms=/tmp/monstore/
[root@host01 ~]# db=/root/db/
[root@host01 ~]# db_slow=/root/db.slow/

[root@host01 ~]# mkdir $ms
[root@host01 ~]# 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。

例

```
[ceph: root@host01 /]# ceph-authtool /etc/ceph/ceph.client.admin.keyring -n mon. --cap
mon 'allow *' --gen-key
[ceph: root@host01 /]# 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。

例

```
[ceph: root@host01 /]# 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。

例

```
[ceph: root@host01 /]# 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。

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

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。

構文

```
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。

構文

```
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。

構文

```
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. すべてのノードで一時的にマウントされたすべての OSD をアンマウントします。

例

```
[root@host01 ~]# umount /var/lib/ceph/osd/ceph-*
```

4. すべての Ceph Monitor デーモンを起動します。

構文

```
systemctl start ceph-FSID@DAEMON_NAME
```

例

```
[root@mon ~]# systemctl start ceph-b404c440-9e4c-11ec-a28a-
001a4a0001df@mon.host01.service
```

5. Monitor がクォーラムを形成できることを確認します。

構文

```
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。

   構文

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

   例

   ```
   [root@mon ~]# systemctl start ceph-b341e254-b165-11ed-a564-
   ac1f6bb26e8c@mgr.extensa003.exrqql.service
   ```

   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。クラスター上のすべての OSD に対して繰り返します。

   構文

   ```
   systemctl start ceph-FSID@osd.OSD_ID
   ```

   例

   ```
   [root@host01 ~]# systemctl start ceph-b404c440-9e4c-11ec-a28a-
   001a4a0001df@osd.0.service
   ```

8. OSD がサービスに返されることを確認します。

   例

   ```
   [ceph: root@host01 /]# ceph -s
   ```

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。
- ネットワーク関連の問題については、**Red Hat Ceph Storage トラブルシューティングガイド** のネットワークの **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。

前提条件

- ネットワーク接続を確認します。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。
- **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。詳細は、**Ceph Monitor 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 の健常性について** 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。

5.1. 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リスト表示しています。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。

前提条件

- Ceph OSD ノードへのルートレベルのアクセス。

5.1.1. 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	Full OSD
HEALTH_WARN	
backfillfull osds	backfillfull OSDS
nearfull osds	Nearfull OSD
osds are down	Down OSD 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

5.1.2. 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

5.1.3. 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。

5.1.4. backfillfull 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: HEALTH_WARN
3 backfillfull osd(s)
Low space hindering backfill (add storage if this doesn't resolve itself): 32 pgs backfill_toofull
```

詳細

1 つ以上の OSD が backfillfull しきい値を超えた場合には、Ceph は、リバランスしてこのデバイスにデータが分散されるのを防ぎます。これは、リバランスが完了していない可能性があり、クラスターがいっぱいに近づいていることを示す早期警告です。backfullfull しきい値のデフォルトは 90% です。

この問題のトラブルシューティング:

プールごとの使用率を確認します。

```
ceph df
```

%RAW USED が 70 ~ 75% を超えている場合は、次のいずれかのアクションを実行できます。

- 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。
- **backfull_toofull** に PG スタックが含まれる OSD の **backfillfull** の比率を増やし、復元プロセスを続行できるようにします。できるだけ早く新しいストレージをクラスターに追加するか、データを削除して、OSD がいっぱいになるのを防ぎます。

  構文

  ```
  ceph osd set-backfillfull-ratio VALUE
  ```

  VALUE の範囲は 0.0 から 1.0 です。

  例

  ```
  [ceph: root@host01/]# ceph osd set-backfillfull-ratio 0.92
  ```

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。

5.1.5. 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. クラスター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. 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。

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

- Full OSDs
- Red Hat Ceph Storage 7 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 7 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 **CRUSH マップの調整可能な変更が Red Hat Ceph Storage の OSD 間で PG 分散に与える影響をテストするにはどうすればよいですか?** 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。

5.1.6. Down OSD

ceph health detail 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、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。

```
[ceph: root@host01 /]# 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。**OSD_ID** をダウンしている OSD の ID に置き換えます。

構文

```
systemctl restart ceph-FSID@osd.OSD_ID
```

例

```
[root@host01 ~]# systemctl restart ceph-b404c440-9e4c-11ec-a28a-001a4a0001df@osd.0.service
```

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リスト表示してから、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。

   a. 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、ファイルへのロギングが有効になると、Ceph はデフォルトでログファイルを **/var/log/ceph/CLUSTER_FSID/** ディレクトリーに保存します。

   b. **EIO** エラーメッセージは、基盤となるディスクの障害を示します。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 ドライブの交換** を参照してください。

   c. 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")
   ```

5. **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 ドライブの交換** を参照してください。

6. 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、ファイルへのロギングが有効になると、Ceph はデフォルトでログファイルを **/var/log/ceph/CLUSTER_FSID/** 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**
- ファイルへのロギングを有効にするには、**Ceph デーモンログ** を参照してください。

5.1.7. 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
2022-05-05 06:27:20.810535 mon.0 [INF] osdmap e609: 9 osds: 8 up, 9 in
2022-05-05 06:27:24.120611 mon.0 [INF] osdmap e611: 9 osds: 7 up, 9 in
2022-05-05 06:27:25.975622 mon.0 [INF] HEALTH_WARN; 118 pgs stale; 2/9 in osds are down
2022-05-05 06:27:27.489790 mon.0 [INF] osdmap e614: 9 osds: 6 up, 9 in
2022-05-05 06:27:36.540000 mon.0 [INF] osdmap e616: 9 osds: 7 up, 9 in
2022-05-05 06:27:39.681913 mon.0 [INF] osdmap e618: 9 osds: 8 up, 9 in
2022-05-05 06:27:43.269401 mon.0 [INF] osdmap e620: 9 osds: 9 up, 9 in
2022-05-05 06:27:54.884426 mon.0 [INF] osdmap e622: 9 osds: 8 up, 9 in
2022-05-05 06:27:57.398706 mon.0 [INF] osdmap e624: 9 osds: 7 up, 9 in
2022-05-05 06:27:59.669841 mon.0 [INF] osdmap e625: 9 osds: 6 up, 9 in
2022-05-05 06:28:07.043677 mon.0 [INF] osdmap e628: 9 osds: 7 up, 9 in
2022-05-05 06:28:10.512331 mon.0 [INF] osdmap e630: 9 osds: 8 up, 9 in
2022-05-05 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。

```
2022-05-25 03:44:06.510583 osd.50 127.0.0.1:6801/149046 18992 : cluster [WRN] map e600547
wrongly marked me down
```

```
2022-05-25 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。
- ネットワークの問題。

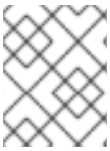
Ceph OSD は、ストレージクラスターのプライベートネットワークに障害が発生したり、クライアント向けのパブリックネットワークに大きな遅延が発生したりする状況を管理できません。

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	説明	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
```

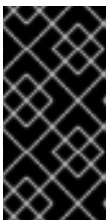
2. **down** としてマークされている OSD と、その OSD が置かれているノードを判別します。

```
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。

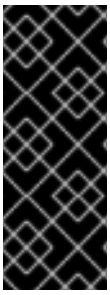
4. **noup** フラグおよび **nodown** フラグを設定して、OSD を **down** および **up** としてマークするのを停止するための一時的な強制モニターを実行できます。

```
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開くには、詳細について **Red Hat サポートへのお問い合わせ** 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 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 トラブルシューティングガイド** の **遅いリクエストまたはブロックされるリクエスト** を参照してください。

5.1.8. 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。

```
2022-05-24 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
```

```
2022-05-25 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
- ネットワークの問題。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については、Red Hat Ceph Storage 7 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	説明
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。
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イスコントローラー (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。

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。

5.2. 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. Cephadm シェルにログインします。

   例

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

2. OSD を停止する前に **noout** フラグを設定します。

   例

   ```
   [ceph: root@host01 /]# ceph osd set noout
   ```

3. 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。

例

```
[ceph: root@host01 /]# ceph osd unset noout
```

関連情報

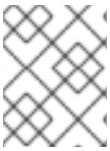
- **Red Hat Ceph Storage アーキテクチャーガイド**の **リバランスおよび復元** セクションを参照してください。

5.3. 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

- 稼働中の Red Hat Ceph Storage クラスターがある。
- Ceph Monitor ノードへの root レベルのアクセス。
- 少なくとも 1 つの OSD が **down** になっています。

Ceph クラスターからの OSD の削除

1. Cephadm シェルにログインします。

例

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

2. **down** になっている OSD を特定します。

例

```
[ceph: root@host01 /]# ceph osd tree | grep -i down
ID  CLASS  WEIGHT   TYPE NAME          STATUS  REWEIGHT  PRI-AFF
 0   hdd 0.00999        osd.0     down  1.00000          1.00000
```

3. クラスターがデータをリバランスして他の OSD にそのテータをコピーできるように、OSD を **out** としてマークします。

構文

```
ceph osd out OSD_ID.
```

例

```
[ceph: root@host01 /]# ceph osd out osd.0
marked out osd.0.
```

注記

OSD が **down** していると、**mon_osd_down_out_interval** パラメーターに基づいて OSD からハートビートパケットを受信しないと、Ceph は、600 秒後に OSD を自動的に **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。

4. 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。

例

```
[ceph: root@host01 /]# ceph -w | grep backfill
2022-05-02 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
2022-05-02 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
2022-05-02 04:48:00.380063 osd.1 [INF] 0.6f starting backfill to osd.0 from (0'0,0'0] MAX to
2521'166639
2022-05-02 04:48:00.380139 osd.1 [INF] 0.48 starting backfill to osd.0 from (0'0,0'0] MAX to
2513'43079
2022-05-02 04:48:00.380260 osd.1 [INF] 0.d starting backfill to osd.0 from (0'0,0'0] MAX to
```

```
2513'136847
2022-05-02 04:48:00.380849 osd.1 [INF] 0.71 starting backfill to osd.0 from (0'0,0'0] MAX to
2331'28496
2022-05-02 04:48:00.381027 osd.1 [INF] 0.51 starting backfill to osd.0 from (0'0,0'0] MAX to
2513'87544
```

配置グループの状態が **active+clean** から **active** になり、一部の劣化したオブジェクトに変化し、移行が完了すると最終的に **active+clean** に変化するはずです。

5. OSD を停止します。

構文

```
ceph orch daemon stop OSD_ID
```

例

```
[ceph: root@host01 /]# ceph orch daemon stop osd.0
```

6. ストレージクラスターから OSD を削除します。

構文

```
ceph orch osd rm OSD_ID --replace
```

例

```
[ceph: root@host01 /]# ceph orch osd rm 0 --replace
```

OSD_IDは保存されます。

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。

1. 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。
2. 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。
3. 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。
4. 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。

Ceph クラスターへの OSD の追加

1. 新しいドライブを挿入したら、以下のオプションを使用して OSD をデプロイすることができます。
   - OSD は、**--unmanaged** パラメーターが設定されていない場合は、Ceph Orchestrator によって自動的にデプロイされます。

例

```
[ceph: root@host01 /]# ceph orch apply osd --all-available-devices
```

- **unmanaged** パラメーターを **true** に設定して、利用可能なすべてのデバイスに OSD をデプロイします。

  例

  ```
  [ceph: root@host01 /]# ceph orch apply osd --all-available-devices --unmanaged=true
  ```

- 特定のデバイスやホストに OSD をデプロイします。

  例

  ```
  [ceph: root@host01 /]# ceph orch daemon add osd host02:/dev/sdb
  ```

2. CRUSH 階層が正確であることを確認します。

   例

   ```
   [ceph: root@host01 /]# ceph osd tree
   ```

関連情報

- **Red Hat Ceph Storage オペレーションガイド** の **すべての利用可能なデバイスへの Ceph OSD のデプロイ** 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 **特定のデバイスおよびホストへの Ceph OSD のデプロイ** セクション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 Down OSD セクションを参照してください。
- Red Hat Ceph Storage インストールガイドを参照してください。

5.4. 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
```

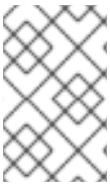
数値を永続的に増やすには、以下のように **/etc/sysctl.conf** ファイルを更新します。

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

5.5. 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. Cephadm シェルにログインします。

   例

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

2. **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。

   ```
   [ceph: root@host01 /]# ceph osd dump | grep -i full
   full_ratio 0.95
   ```

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

   ```
   [ceph: root@host01 /]# 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。

4. 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。

   ```
   [ceph: root@host01 /]# ceph osd dump | grep -i full
   full_ratio 0.97
   ```

5. クラスターの状態を監視します。

   ```
   [ceph: root@host01 /]# 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。

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

```
[ceph: root@host01 /]# ceph osd set-full-ratio 0.95
```

7. 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。

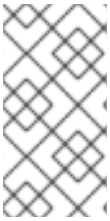
```
[ceph: root@host01 /]# ceph osd dump | grep -i full
full_ratio 0.95
```

関連情報

- **Red Hat Ceph Storage トラブルシューティングガイド**の Full OSD セクション
- **Red Hat Ceph Storage トラブルシューティングガイド** 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

この章では、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。



注記

radosgw-adminbucket sync status コマンドにより、データがマルチサイト間で一貫している場合でもバケットがシャード上で遅れていることが報告された場合は、バケットへの追加の書き込みを実行します。ステータスレポートを同期し、バケットがソースに追いついたというメッセージを表示します。

前提条件

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

6.1. 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。

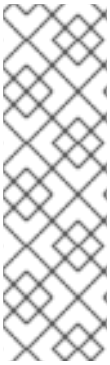
6.2. マルチサイト CEPH OBJECT GATEWAY の同期

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。

例

```
[ceph: root@host01 /]# 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、それをリスト表示します。



注記

radosgw-admin sync status コマンドを実行すると、シャードの回復が見られることがあります。データの同期の場合には、レプリケーションログのシャードが 128 個あり、それぞれ個別に処理されます。これらのレプリケーションログイベントによってトリガーされたアクションのいずれかがネットワーク、ストレージ、またはその他の場所から発生した場合、これらのエラーは追跡されるため、操作を後で再試行できます。特定のシャードに再試行が必要なエラーがある場合、**radosgw-admin sync status** コマンドはそのシャードを **回復中** として報告します。この回復は自動的に行われるため、Operator が介入して問題を解決する必要はありません。

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。

マルチサイトオブジェクト内のバケットも Ceph ダッシュボードで監視できます。詳細は、**Red Hat Ceph Storage ダッシュボードガイド** の **マルチサイトオブジェクトのバケットのモニタリング** を参照してください。

構文

```
radosgw-admin data sync status --shard-id=X --source-zone=ZONE_NAME
```

例

```
[ceph: root@host01 /]# radosgw-admin data sync status --shard-id=27 --source-zone=us-east
{
  "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。

構文

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

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. 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 --admin-daemon** コマンドを使用します。

構文

```
ceph --admin-daemon /var/run/ceph/ceph-client.rgw.RGW_ID.asok perf dump data-sync-from-ZONE_NAME
```

例

```
[ceph: root@host01 /]# ceph --admin-daemon /var/run/ceph/ceph-client.rgw.host02-rgw0.103.94309060818504.asok perf dump data-sync-from-us-west
```

```
{
    "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 --admin-daemon** コマンドを実行する必要があります。

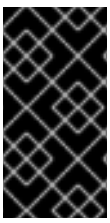
関連情報

- 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。

6.4. マルチサイトの CEPH OBJECT GATEWAY 設定でのデータ同期

ストレージクラスターのマルチサイト Ceph Object Gateway 設定では、フェイルオーバーおよびフェイルバックにより、データの同期が停止します。**radosgw-admin sync status** コマンドは、データ同期が長期間遅れていることを報告します。

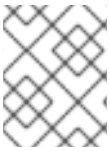
radosgw-admin data sync init コマンドを実行してサイト間でデータを同期してから、Ceph Object Gateway を再起動できます。このコマンドは実際のオブジェクトデータには触れず、指定されたソースゾーンのデータ同期を開始します。これにより、ゾーンはソースゾーンから完全同期を再開します。



重要

data sync init コマンドを実行する前に、**Red Hat サポート** にお問い合わせください。

同期を完全に再開する場合、およびソースゾーンで同期が必要なデータが大量にある場合は、帯域幅の消費が高くなるため、それに応じて計画する必要があります。



注記

ユーザーがセカンダリーサイトのバケットを誤って削除した場合は、サイトで **metadata sync init** コマンドを使用してデータを同期できます。

前提条件

- 稼働中の Red Hat Ceph Storage クラスターがある。

- 少なくとも 2 つのサイトで設定された Ceph Object Gateway。

手順

1. サイト間の同期ステータスを確認します。

   例

```
[ceph: host04 /]# radosgw-admin sync status
          realm d713eec8-6ec4-4f71-9eaf-379be18e551b (india)
      zonegroup ccf9e0b2-df95-4e0a-8933-3b17b64c52b7 (shared)
           zone 04daab24-5bbd-4c17-9cf5-b1981fd7ff79 (primary)
   current time 2022-09-15T06:53:52Z
zonegroup features enabled: resharding
  metadata sync no sync (zone is master)
      data sync source: 596319d2-4ffe-4977-ace1-8dd1790db9fb (secondary)
                        syncing
                        full sync: 0/128 shards
                        incremental sync: 128/128 shards
                        data is caught up with source
```

2. セカンダリーゾーンからデータを同期します。

   例

```
[ceph: root@host04 /]# radosgw-admin data sync init --source-zone primary
```

3. サイトですべての Ceph Object Gateway デーモンを再起動します。

   例

```
[ceph: root@host04 /]# ceph orch restart rgw.myrgw
```

第7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。

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。

7.1. 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、「配置グループのリスト表示 (**stale**、**inactive**、または **unclean** 状態)」 を参照してください。

前提条件

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

7.1.1. 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	**down** している配置グループ
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

7.1.2. 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** リストで表示します。

```
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、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、次の例のような情報が含まれています。

   例

   ```
   [ceph: root@host01 /]# 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 OSDs を参照してください。

関連情報

- Red Hat Ceph Storage 7 の**管理ガイド** の **配置グループ設定の監視** セクション

7.1.3. 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. Cephadm シェルにログインします。

   例

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

2. 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: root@host01 /]# 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
   ```

3. 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。

      構文

      ```
      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。

      ```
      [ceph: root@host01 /]# 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。

      構文

      ```
      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。

      ```
      [ceph: root@host01 /]# ceph -w | grep 0.6
      2022-05-26 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.
      2022-05-26 01:35:36.788334 osd.106 [ERR] 0.6 deep-scrub 1 errors
      ```

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、**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。

構文

```
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
```

5. 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 を参照してください。
- 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 セクションを参照してください。

7.1.4. 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: root@host01 /]# 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

7.1.5. 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

- 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
- 詳細は、Down OSD を参照してください。

7.1.6. 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。

構文

```
ceph pg ID query
```

ID を **down** している配置グループの ID に置き換えます。

例

```
[ceph: root@host01 /]#  ceph pg 0.5 query

{ "state": "down+peering",
  ...
  "recovery_state": [
      { "name": "Started\/Primary\/Peering\/GetInfo",
        "enter_time": "2021-08-06 14:40:16.169679",
        "requested_info_from": []},
      { "name": "Started\/Primary\/Peering",
        "enter_time": "2021-08-06 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": "2021-08-06 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 管理ガイドの Ceph OSD ピアリング セクション

7.1.7. 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. Cephadm シェルにログインします。

   例

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

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

   ```
   [ceph: root@host01 /]# 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%)**
   ```

3. 配置グループに関する詳細情報を表示します。

   構文

   ```
   ceph pg ID query
   ```

   ID を、**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。

   例

   ```
   [ceph: root@host01 /]# 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": "2021-08-28 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": "2021-08-28 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。

4. **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。
5. 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。

7.2. 配置グループのリスト表示 (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
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. Cephadm シェルにログインします。

   例

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

2. スタックした PG をリストします。

   例

```
[ceph: root@host01 /]# ceph pg dump_stuck inactive
[ceph: root@host01 /]# ceph pg dump_stuck unclean
[ceph: root@host01 /]# 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。

7.3. 配置グループ不整合のリスト表示

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。より詳細な出力をリスト表示するには、**--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

- プール内の一貫性のない配置グループをすべて表示します。

  構文

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

  例

  ```
  [ceph: root@host01 /]# rados list-inconsistent-pg data --format=json-pretty
  [0.6]
  ```

- ID を持つ配置グループ内の一貫性のないオブジェクトを表示します。

  構文

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

  例

  ```
  [ceph: root@host01 /]# 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リスト。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。

- 一貫性のないスナップショットのセットを一覧表示します

  構文

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

  例

  ```
  [ceph: root@host01 /]# 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、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。

7.4. 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

- **inconsistent** 配置グループを修復します。

  構文

  ```
  ceph pg repair ID
  ```

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

関連情報

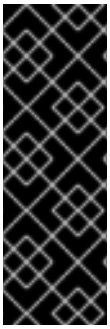
- **Red Hat Ceph Storage トラブルシューティングガイド**の **一貫性のない配置グループ** セクションを参照してください。
- **Red Hat Ceph Storage トラブルシューティングガイド** の **配置グループの不整合の一覧表示** を参照してください。

7.5. 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。

   ```
   [ceph: root@host01 /]# 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。

   ```
   [ceph: root@host01 /]# ceph osd set noscrub
   [ceph: root@host01 /]# 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。

   構文

   ```
   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。

   例

   ```
   [ceph: root@host01 /]# 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。

例

```
[ceph: root@host01 /]# 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。

   構文

   ```
   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: root@host01 /]# 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: root@host01 /]# 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: root@host01 /]# 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: root@host01 /]# ceph osd unset noscrub
   [ceph: root@host01 /]# ceph osd unset nodeep-scrub
   ```

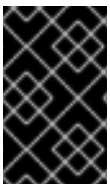
関連情報

- **ほぼ完全な OSD** を参照してください
- **Red Hat Ceph Storage 管理ガイド** の **配置グループ設定の監視** セクション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。
- 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。
- PG オートスケーラーの詳細については、Red Hat Ceph Storage ストレージ戦略ガイドの **配置グループの自動スケーリング** セクションを参照してください。

第8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。

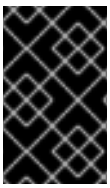
前提条件

- ネットワーク関連の問題がないことを確認します。

8.1. 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。

- オブジェクトのリスト表示
- 失われたオブジェクトのリスト表示
- 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。

前提条件

- Ceph OSD ノードへのルートレベルのアクセス。

8.1.1. オブジェクトのリスト表示

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リスト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。

構文

```
systemctl status ceph-FSID@osd.OSD_ID
```

例

```
[root@host01 ~]# systemctl status ceph-b404c440-9e4c-11ec-a28a-
001a4a0001df@osd.0.service
```

2. OSD コンテナーにログインします。

構文

```
cephadm shell --name osd.OSD_ID
```

例

```
[root@host01 ~]# cephadm shell --name osd.0
```

3. 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。

構文

```
ceph-objectstore-tool --data-path PATH_TO_OSD --op list
```

例

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --op list
```

4. 配置グループ内のすべてのオブジェクトを特定します。

構文

```
ceph-objectstore-tool --data-path PATH_TO_OSD --pgid PG_ID --op list
```

例

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --pgid 0.1c
--op list
```

5. オブジェクトが属する PG を特定します。

構文

```
ceph-objectstore-tool --data-path PATH_TO_OSD --op list OBJECT_ID
```

例

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --op list
default.region
```

8.1.2. 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** をリスト表示し、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

   ```
   systemctl status ceph-FSID@osd.OSD_ID
   ```

   例

   ```
   [root@host01 ~]# systemctl status ceph-b404c440-9e4c-11ec-a28a-001a4a0001df@osd.0.service
   ```

2. OSD コンテナーにログインします。

   構文

   ```
   cephadm shell --name osd.OSD_ID
   ```

   例

   ```
   [root@host01 ~]# cephadm shell --name osd.0
   ```

3. 失われたレガシーオブジェクトをすべてリスト表示します。

   構文

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

   例

   ```
   [ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --op fix-lost --dry-run
   ```

4. **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

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

例

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

```
ceph-objectstore-tool --data-path PATH_TO_OSD --pgid PG_ID --op fix-lost
```

例

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

```
ceph-objectstore-tool --data-path PATH_TO_OSD --op fix-lost OBJECT_ID
```

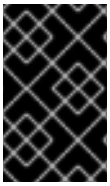
例

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --op
fix-lost default.region
```

8.2. 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) のリスト表示
- OMAP ヘッダーの操作
- OMAP キーの操作
- オブジェクトの属性のリスト表示
- オブジェクトの属性キーの操作



重要

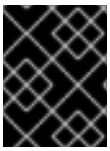
オブジェクトを操作すると、回復不能なデータ損失が発生する可能性があります。**ceph-objectstore-tool** ユーティリティーを使用する前に、Red Hat サポートにお問い合わせください。

前提条件

- Ceph OSD ノードへのルートレベルのアクセス。

8.2.1. 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。

   構文

   ```
   systemctl status ceph-FSID@osd.OSD_ID
   ```

   例

   ```
   [root@host01 ~]# systemctl status ceph-b404c440-9e4c-11ec-a28a-001a4a0001df@osd.0.service
   ```

2. OSD または配置グループ (PG) のオブジェクトをリスト表示してオブジェクトを見つけます。
3. OSD コンテナーにログインします。

   構文

   ```
   cephadm shell --name osd.OSD_ID
   ```

   例

   ```
   [root@host01 ~]# cephadm shell --name osd.0
   ```

4. 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。

構文

```
ceph-objectstore-tool --data-path PATH_TO_OSD --pgid PG_ID \
OBJECT \
get-bytes > OBJECT_FILE_NAME
```

例

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

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

5. 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。
6. オブジェクトのバイトを設定します。

構文

```
ceph-objectstore-tool --data-path PATH_TO_OSD --pgid PG_ID \
OBJECT \
set-bytes < OBJECT_FILE_NAME
```

例

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

8.2.2. 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. OSD コンテナーにログインします。

   構文

   ```
   cephadm shell --name osd.OSD_ID
   ```

   例

   ```
   [root@host01 ~]# cephadm shell --name osd.0
   ```

2. オブジェクトの削除

   構文

   ```
   ceph-objectstore-tool --data-path PATH_TO_OSD --pgid PG_ID \
   OBJECT \
   remove
   ```

   例

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

8.2.3. オブジェクトマップのリスト表示

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) の内容をリスト表示します。この出力では、キーのリストが表示されます。

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。

   構文

   ```
   systemctl status ceph-osd@OSD_ID
   ```

   例

   ```
   [root@host01 ~]# systemctl status ceph-b404c440-9e4c-11ec-a28a-
   001a4a0001df@osd.0.service
   ```

2. OSD コンテナーにログインします。

   構文

   ```
   cephadm shell --name osd.OSD_ID
   ```

   例

   ```
   [root@host01 ~]# cephadm shell --name osd.0
   ```

3. オブジェクトマップをリスト表示します。

   構文

   ```
   ceph-objectstore-tool --data-path PATH_TO_OSD --pgid PG_ID \
   OBJECT \
   list-omap
   ```

   例

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

8.2.4. 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。

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

   ```
   systemctl status ceph-FSID@osd.OSD_ID
   ```

   例

   ```
   [root@host01 ~]# systemctl status ceph-b404c440-9e4c-11ec-a28a-
   001a4a0001df@osd.0.service
   ```

2. OSD コンテナーにログインします。

構文

```
cephadm shell --name osd.OSD_ID
```

例

```
[root@host01 ~]# cephadm shell --name osd.0
```

3. オブジェクトマップヘッダーを取得します。

構文

```
ceph-objectstore-tool --data-path PATH_TO_OSD \
--pgid PG_ID OBJECT \
get-omaphdr > OBJECT_MAP_FILE_NAME
```

例

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

4. オブジェクトマップヘッダーを設定します。

構文

```
ceph-objectstore-tool --data-path PATH_TO_OSD \
--pgid PG_ID OBJECT \
get-omaphdr < OBJECT_MAP_FILE_NAME
```

例

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

8.2.5. 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。

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. OSD コンテナーにログインします。

   構文

   ```
   cephadm shell --name osd.OSD_ID
   ```

   例

   ```
   [root@host01 ~]# cephadm shell --name osd.0
   ```

2. オブジェクトマップキーを取得します。

   構文

   ```
   ceph-objectstore-tool --data-path PATH_TO_OSD \
   --pgid PG_ID OBJECT \
   get-omap KEY > OBJECT_MAP_FILE_NAME
   ```

   例

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

3. オブジェクトマップキーを設定します。

   構文

   ```
   ceph-objectstore-tool --data-path PATH_TO_OSD \
   --pgid PG_ID OBJECT \
   set-omap KEY < OBJECT_MAP_FILE_NAME
   ```

   例

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

4. オブジェクトマップキーを削除します。

   構文

   ```
   ceph-objectstore-tool --data-path PATH_TO_OSD \
   --pgid PG_ID OBJECT \
   rm-omap KEY
   ```

   例

   ```
   [ceph: root@host01 /]# 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 ""
```

8.2.6. オブジェクトの属性のリスト表示

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リスト表示します。この出力には、オブジェクトのキーと値が表示されます。

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

   ```
   systemctl status ceph-FSID@osd.OSD_ID
   ```

   例

   ```
   [root@host01 ~]# systemctl status ceph-b404c440-9e4c-11ec-a28a-
   001a4a0001df@osd.0.service
   ```

2. OSD コンテナーにログインします。

   構文

   ```
   cephadm shell --name osd.OSD_ID
   ```

   例

   ```
   [root@host01 ~]# cephadm shell --name osd.0
   ```

3. オブジェクトの属性をリスト表示します。

   構文

   ```
   ceph-objectstore-tool --data-path PATH_TO_OSD \
   --pgid PG_ID OBJECT \
   list-attrs
   ```

   例

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

8.2.7. 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。

前提条件

- Ceph OSD ノードへのルートレベルのアクセス。
- **ceph-osd** デーモンを停止します。

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

   ```
   systemctl status ceph-FSID@osd.OSD_ID
   ```

   例

   ```
   [root@host01 ~]# systemctl status ceph-b404c440-9e4c-11ec-a28a-
   001a4a0001df@osd.0.service
   ```

2. OSD コンテナーにログインします。

   構文

   ```
   cephadm shell --name osd.OSD_ID
   ```

   例

   ```
   [root@host01 ~]# cephadm shell --name osd.0
   ```

3. オブジェクトの属性を取得します。

   構文

   ```
   ceph-objectstore-tool --data-path PATH_TO_OSD \
   --pgid PG_ID OBJECT \
   get-attr KEY > OBJECT_ATTRS_FILE_NAME
   ```

   例

   ```
   [ceph: root@host01 /]# 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-attr "oid" > zone_info.default.attr.txt
   ```

4. オブジェクトの属性を設定します。

構文

```
ceph-objectstore-tool --data-path PATH_TO_OSD \
--pgid PG_ID OBJECT \
set-attr KEY < OBJECT_ATTRS_FILE_NAME
```

例

```
[ceph: root@host01 /]# 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-attr "oid"<zone_info.default.attr.txt
```

5. オブジェクトの属性を削除します。

構文

```
ceph-objectstore-tool --data-path PATH_TO_OSD \
--pgid PG_ID OBJECT  \
rm-attr KEY
```

例

```
[ceph: root@host01 /]# 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-attr "oid"
```

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9章 ストレッチモードでのクラスターのトラブルシューティング

障害が発生したタイブレーカーモニターを交換および削除できます。必要に応じて、クラスターを強制的に回復モードまたは正常モードにすることもできます。

関連情報

ストレッチモードのクラスターの詳細は、Ceph Storage のストレッチクラスターを参照してください。

9.1. タイブレーカーをクォーラム内のモニターに置き換える

タイブレーカーモニターに障害が発生した場合は、それをクォーラム内の既存のモニターに置き換えて、クラスターから削除できます。

前提条件

- 稼働中の Red Hat Ceph Storage クラスターがある。
- クラスターでストレッチモードが有効になっている

手順

1. 自動化されたモニターのデプロイメントを無効にします。

   例

   ```
   [ceph: root@host01 /]# ceph orch apply mon --unmanaged

   Scheduled mon update…
   ```

2. クォーラムでモニターを表示します。

   例

   ```
   [ceph: root@host01 /]# ceph -s

   mon: 5 daemons, quorum host01, host02, host04, host05 (age 30s), out of quorum: host07
   ```

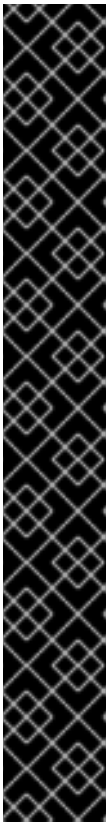
3. モニターを新しいタイブレーカーとしてクォーラムに設定します。

   構文

   ```
   ceph mon set_new_tiebreaker NEW_HOST
   ```

   例

   ```
   [ceph: root@host01 /]# ceph mon set_new_tiebreaker host02
   ```



重要

モニターが既存の非タイブレーカーモニターと同じ場所にある場合、エラーメッセージが表示されます。

例

```
[ceph: root@host01 /]# ceph mon set_new_tiebreaker host02

Error EINVAL: mon.host02 has location DC1, which matches mons host02 on the datacenter dividing bucket for stretch mode.
```

その場合は、モニターの場所を変更します。

構文

```
ceph mon set_location HOST datacenter=DATACENTER
```

例

```
[ceph: root@host01 /]# ceph mon set_location host02 datacenter=DC3
```

4. 障害が発生したタイブレーカーモニターを削除します。

   構文

   ```
   ceph orch daemon rm FAILED_TIEBREAKER_MONITOR --force
   ```

   例

   ```
   [ceph: root@host01 /]# ceph orch daemon rm mon.host07 --force

   Removed mon.host07 from host 'host07'
   ```

5. モニターがホストから削除されたら、モニターを再デプロイします。

   構文

   ```
   ceph mon add HOST IP_ADDRESS datacenter=DATACENTER
   ceph orch daemon add mon HOST
   ```

   例

   ```
   [ceph: root@host01 /]# ceph mon add host07 213.222.226.50 datacenter=DC1
   [ceph: root@host01 /]# ceph orch daemon add mon host07
   ```

6. クォーラムに 5 つのモニターがあることを確認します。

   例

```
[ceph: root@host01 /]# ceph -s

mon: 5 daemons, quorum host01, host02, host04, host05, host07 (age 15s)
```

7. すべてが正しく設定されていることを確認します。

例

```
[ceph: root@host01 /]# ceph mon dump

epoch 19
fsid 1234ab78-1234-11ed-b1b1-de456ef0a89d
last_changed 2023-01-17T04:12:05.709475+0000
created 2023-01-16T05:47:25.631684+0000
min_mon_release 16 (pacific)
election_strategy: 3
stretch_mode_enabled 1
tiebreaker_mon host02
disallowed_leaders host02
0: [v2:132.224.169.63:3300/0,v1:132.224.169.63:6789/0] mon.host02; crush_location
{datacenter=DC3}
1: [v2:220.141.179.34:3300/0,v1:220.141.179.34:6789/0] mon.host04; crush_location
{datacenter=DC2}
2: [v2:40.90.220.224:3300/0,v1:40.90.220.224:6789/0] mon.host01; crush_location
{datacenter=DC1}
3: [v2:60.140.141.144:3300/0,v1:60.140.141.144:6789/0] mon.host07; crush_location
{datacenter=DC1}
4: [v2:186.184.61.92:3300/0,v1:186.184.61.92:6789/0] mon.host03; crush_location
{datacenter=DC2}
dumped monmap epoch 19
```

8. モニターを再デプロイします。

構文

```
ceph orch apply mon --placement="HOST_1, HOST_2, HOST_3, HOST_4, HOST_5”
```

例

```
[ceph: root@host01 /]# ceph orch apply mon --placement="host01, host02, host04, host05,
host07"

Scheduled mon update...
```

9.2. タイブレーカーを新しいモニターに交換する

タイブレーカーモニターに障害が発生した場合は、それを新しいモニターに置き換えて、クラスターから削除できます。

前提条件

- 稼働中の Red Hat Ceph Storage クラスターがある。

- クラスターでストレッチモードが有効になっている

手順

1. 新しいモニターをクラスターに追加します。
   a. **crush_location** を新しいモニターに手動で追加します。

      構文

      ```
      ceph mon add NEW_HOST IP_ADDRESS datacenter=DATACENTER
      ```

      例

      ```
      [ceph: root@host01 /]# ceph mon add host06 213.222.226.50 datacenter=DC3

      adding mon.host06 at [v2:213.222.226.50:3300/0,v1:213.222.226.50:6789/0]
      ```

      注記

      新しいモニターは、既存の非タイブレーカーモニターとは別の場所に配置する必要があります。

   b. 自動化されたモニターのデプロイメントを無効にします。

      例

      ```
      [ceph: root@host01 /]# ceph orch apply mon --unmanaged

      Scheduled mon update…
      ```

   c. 新しいモニターをデプロイします。

      構文

      ```
      ceph orch daemon add mon NEW_HOST
      ```

      例

      ```
      [ceph: root@host01 /]# ceph orch daemon add mon host06
      ```

2. 6 つのモニターがあり、そのうちの 5 つがクォーラムにあることを確認します。

   例

   ```
   [ceph: root@host01 /]# ceph -s

   mon: 6 daemons, quorum host01, host02, host04, host05, host06 (age 30s), out of quorum: host07
   ```

3. 新しいモニターを新しいタイブレーカーとして設定します。

   構文

構文

```
ceph mon set_new_tiebreaker NEW_HOST
```

例

```
[ceph: root@host01 /]# ceph mon set_new_tiebreaker host06
```

4. 障害が発生したタイブレーカーモニターを削除します。

構文

```
ceph orch daemon rm FAILED_TIEBREAKER_MONITOR --force
```

例

```
[ceph: root@host01 /]# ceph orch daemon rm mon.host07 --force

Removed mon.host07 from host 'host07'
```

5. すべてが正しく設定されていることを確認します。

例

```
[ceph: root@host01 /]# ceph mon dump

epoch 19
fsid 1234ab78-1234-11ed-b1b1-de456ef0a89d
last_changed 2023-01-17T04:12:05.709475+0000
created 2023-01-16T05:47:25.631684+0000
min_mon_release 16 (pacific)
election_strategy: 3
stretch_mode_enabled 1
tiebreaker_mon host06
disallowed_leaders host06
0: [v2:213.222.226.50:3300/0,v1:213.222.226.50:6789/0] mon.host06; crush_location
{datacenter=DC3}
1: [v2:220.141.179.34:3300/0,v1:220.141.179.34:6789/0] mon.host04; crush_location
{datacenter=DC2}
2: [v2:40.90.220.224:3300/0,v1:40.90.220.224:6789/0] mon.host01; crush_location
{datacenter=DC1}
3: [v2:60.140.141.144:3300/0,v1:60.140.141.144:6789/0] mon.host02; crush_location
{datacenter=DC1}
4: [v2:186.184.61.92:3300/0,v1:186.184.61.92:6789/0] mon.host05; crush_location
{datacenter=DC2}
dumped monmap epoch 19
```

6. モニターを再デプロイします。

構文

```
ceph orch apply mon --placement="HOST_1, HOST_2, HOST_3, HOST_4, HOST_5”
```

例

```
[ceph: root@host01 /]# ceph orch apply mon --placement="host01, host02, host04, host05, host06"

Scheduled mon update…
```

9.3. ストレッチクラスターを強制的に回復モードまたは正常モードにする

Stretch Degraded モードの場合、切断されたデータセンターが復旧すると、クラスターは自動的に回復モードになります。それが起こらない場合、または回復モードを早期に有効にしたい場合は、ストレッチクラスターを強制的に回復モードにすることができます。

前提条件

- 稼働中の Red Hat Ceph Storage クラスターがある。
- クラスターでストレッチモードが有効になっている

手順

1. ストレッチクラスターを強制的に回復モードにします。

   例

   ```
   [ceph: root@host01 /]#  ceph osd force_recovery_stretch_mode --yes-i-really-mean-it
   ```

   注記

   回復状態では、クラスターは **HEALTH_WARN** 状態になります。

2. 回復モードの場合、配置グループが正常になった後、クラスターは通常のストレッチモードに戻る必要があります。それが起こらない場合は、ストレッチクラスターを強制的に正常モードにすることができます。

   例

   ```
   [ceph: root@host01 /]#  ceph osd force_healthy_stretch_mode --yes-i-really-mean-it
   ```

   注記

   クロスデータセンターピアリングを早期に強制する必要があり、データダウンタイムのリスクを許容する場合、またはすべての配置グループが完全に復旧していなくてもピアリングできることを個別に確認している場合にも、このコマンドを実行できます。

   また、回復状態によって生成される **HEALTH_WARN** 状態を削除するために、正常モードを呼び出すこともできます。

注記

force_recovery_stretch_mode および **force_recovery_healthy_mode** コマンドは、予期しない状況を管理するプロセスに含まれているため、必要ありません。

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。

前提条件

- Red Hat サポートアカウント

10.1. 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、What is an sosreport and how to create one in Red Hat Enterprise Linux? 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。

10.2. 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。

前提条件

1. 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ておく。
   a. 次のリポジトリーを有効にして、必要な debuginfo パッケージをインストールしておく。

   例

   ```
   [root@host01 ~]# subscription-manager repos --enable=rhceph-6-tools-for-rhel-9-x86_64-rpms
   [root@host01 ~]# yum --enable=rhceph-6-tools-for-rhel-9-x86_64-debug-rpms
   ```

   リポジトリーを有効にすると、サポートされるパッケージのこの一覧から必要な debuginfo パッケージをインストールできます。

```
ceph-base-debuginfo
ceph-common-debuginfo
ceph-debugsource
ceph-fuse-debuginfo
ceph-immutable-object-cache-debuginfo
ceph-mds-debuginfo
ceph-mgr-debuginfo
ceph-mon-debuginfo
ceph-osd-debuginfo
ceph-radosgw-debuginfo
cephfs-mirror-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。

   例

   ```
   [root@host01 ~]# dnf install gdb
   ```

- 「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」

10.2.1. 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 6 のコアダンプファイルを生成できます。これには、コアダンプファイルをキャプチャーする 2 つのシナリオが含まれます。

- 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。
- ホストに 8 GB 以上の RAM があることを確認します。ホストに複数のデーモンがある場合は、Red Hat は RAM を増やすことを推奨します。

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。

      例

```
[root@mon]# echo "| /usr/lib/systemd/systemd-coredump %P %u %g %s %t %c %h %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。

例

```
[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 OSD** 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. **MONITOR_ID** または **OSD_ID** を取得して、コンテナーを入力します。

構文

```
podman ps
podman exec -it MONITOR_ID_OR_OSD_ID bash
```

例

```
[root@host01 ~]# podman ps
[root@host01 ~]# podman exec -it ceph-1ca9f6a8-d036-11ec-8263-fa163ee967ad-osd-2
bash
```

b. **procps-ng** パッケージおよび **gdb** パッケージをコンテナーにインストールします。

例

```
[root@host01 ~]# dnf install procps-ng gdb
```

c. プロセス ID を検索します。

構文

```
ps -aef | grep PROCESS | grep -v run
```

PROCESS は、実行中のプロセスの名前に置き換えます (例: **ceph-mon** または **ceph-osd**)。

例

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

d. 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プロセスの ID に置き換えます (例: **18110**)。

例

```
[root@host01 ~]# 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ていることを確認します。

例

```
[root@host01 ~]# ls -ltr
total 709772
-rw-r--r--. 1 root root 726799544 Mar 18 19:46 core.18110
```

f. 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。

構文

```
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。

3. 他の Ceph デーモン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. **cephadm** シェルにログインします。

例

```
[root@host03 ~]# cephadm shell
```

b. デーモンの **ptrace** を有効にします。

例

```
[ceph: root@host01 /]# ceph config set mgr mgr/cephadm/allow_ptrace true
```

c. デーモンサービスを再デプロイします。

構文

```
ceph orch redeploy SERVICE_ID
```

例

```
[ceph: root@host01 /]# ceph orch redeploy mgr
[ceph: root@host01 /]# ceph orch redeploy rgw.rgw.1
```

d. **cephadm shell** を終了し、デーモンがデプロイされているホストにログインします。

例

```
[ceph: root@host01 /]# exit
[root@host01 ~]# ssh root@10.0.0.11
```

e. DAEMON_ID を取得して、コンテナーを入力します。

例

```
[root@host04 ~]# podman ps
[root@host04 ~]# podman exec -it ceph-1ca9f6a8-d036-11ec-8263-fa163ee967ad-rgw-rgw-1-host04 bash
```

f. **procps-ng** パッケージおよび **gdb** パッケージをインストールします。

例

```
[root@host04 /]# dnf install procps-ng gdb
```

g. プロセスの PID を取得します。

例

```
[root@host04 /]# ps aux | grep rados
ceph           6  0.3  2.8 5334140 109052 ?      Sl   May10   5:25 /usr/bin/radosgw -n client.rgw.rgw.1.host04 -f --setuser ceph --setgroup ceph --default-log-to-file=false --default-log-to-stderr=true --default-log-stderr-prefix=debug
```

h. コアダンプを収集します。

構文

```
gcore PID
```

例

```
[root@host04 /]# gcore 6
```

i. コアダンプファイルが正しく生成されていることを確認します。

例

```
[root@host04 /]# ls -ltr
total 108798
-rw-r--r--. 1 root root 726799544 Mar 18 19:46 core.6
```

-

j. コンテナー外でコアダンプファイルをコピーします。

構文

```
podman cp ceph-mon-DAEMON_ID:/tmp/mon.core.PID /tmp
```

DAEMON_ID は Ceph デーモンの ID 番号に、 **PID** はプロセス ID 番号に置き換えます。

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。

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
ミリ秒	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

付録B CEPH クラスターの正常性メッセージ

Red Hat Ceph Storage クラスターが出力する可能性のある正常性メッセージには限りがあります。これらは、固有の識別子を持つヘルスチェックとして定義されています。識別子は、ツールが正常性チェックを理解し、その意味を反映する方法でそれらを提示できるようにすることを目的とした、簡潔な疑似人間可読文字列です。

表B.1 Monitor

正常性コード	説明
DAEMON_OLD_VERSION	すべてのデーモンで古いバージョンの Ceph が実行されている場合は警告します。複数のバージョンが検出された場合は、正常性エラーが発生します。
MON_DOWN	1つまたは複数の Ceph Monitor デーモンが現在ダウンしています。
MON_CLOCK_SKEW	**ceph-mon** デーモンを実行しているノードのクロックが十分に同期されていません。**ntpd** や **chrony** でクロックを同期させることで解決します。
MON_MSGR2_NOT_ENABLED	**ms_bind_msgr2** オプションは有効ですが、1つまたは複数の Ceph Monitor がクラスターの monmap で v2 ポートにバインドするように設定されていません。**ceph mon enable-msgr2** コマンドを実行して解決します。
MON_DISK_LOW	1つまたは複数の Ceph Monitor のディスク領域が不足しています。
MON_DISK_CRIT	1つまたは複数の Ceph Monitor のディスク領域が極端に少なくなっています。
MON_DISK_BIG	1つまたは複数の Ceph Monitor のデータベースサイズが非常に大きくなっています。
AUTH_INSECURE_GLOBAL_ID_RECLAIM	Ceph Monitor への再接続時に **global_id** を安全に再要求していないクライアントまたはデーモンが1つ以上ストレージクラスターに接続されています。
AUTH_INSECURE_GLOBAL_ID_RECLAIM_ALLOWED	Ceph は現在、**auth_allow_insecure_global_id_reclaim** 設定が **true** に設定されているため、クライアントが安全でないプロセスを使用してモニターに再接続し、以前の **global_id** を再取得できるように設定されています。

表B.2 Manager

正常性コード	説明
MGR_DOWN	すべての Ceph Manager デーモンは現在ダウンしています。
MGR_MODULE_DEPENDENCY	有効な Ceph Manager モジュールが依存関係のチェックに失敗しています。
MGR_MODULE_ERROR	Ceph Manager モジュールに予期せぬエラーが発生しました。通常、これは、モジュールのサーブ関数から未処理の例外が発生したことを意味します。

表B.3 OSD

正常性コード	説明
OSD_DOWN	1つまたは複数の OSD がダウンとマークされています。
OSD_CRUSH_TYPE_DOWN	特定の CRUSH サブツリー内のすべての OSD が down とマークされています。たとえば、あるホスト上のすべての OSD が down とマークされます。たとえば、OSD_HOST_DOWN および OSD_ROOT_DOWN です。
OSD_ORPHAN	CRUSH マップの階層で OSD が参照されていますが、存在していません。**ceph osd crush rm osd._OSD_ID** コマンドを実行して、OSD を削除します。
OSD_OUT_OF_ORDER_FULL	nearfull、backfillfull、full、または failsafefull の使用率のしきい値は昇順ではありません。**ceph osd set-nearfull-ratio RATIO**、**ceph osd set-backfillfull-ratio RATIO**、および **ceph osd set-full-ratio RATIO** を実行して、しきい値を調整します。
OSD_FULL	1つ以上の OSD が完全なしきい値を超えており、ストレージクラスターが書き込みを処理できないようになっています。わずかなマージン **ceph osd set-full-ratio RATIO** で完全なしきい値を上げることにより、書き込みの可用性を復元します。
OSD_BACKFILLFULL	1つ以上の OSD が backfillfull しきい値を超えたため、データをこのデバイスにリバランスできなくなります。
OSD_NEARFULL	1つまたは複数の OSD が nearfull の閾値を超えました。

正常性コード	説明
OSDMAP_FLAGS	1つまたは複数のストレージクラスターフラグが設定されています。これらのフラグには、**full**、**pauserd**、**pausewr**、**noup**、**nodown**、**noin**、**noout**、**nobackfill**、**norecover**、**norebalance**、**noscrub**、**nodeep_scrub**、および **notieragent** が含まれます。**full** 以外は、**ceph osd set FLAG** コマンドおよび **ceph osd unset FLAG** コマンドでフラグをクリアできます。
OSD_FLAGS	1つ以上の OSD または CRUSH に対象のフラグが設定されています。これらのフラグには、**noup**、**nodown**、**noin**、および **noout** があります。
OLD_CRUSH_TUNABLES	CRUSH マップは非常に古い設定を使用しているため、更新する必要があります。
OLD_CRUSH_STRAW_CALC_VERSION	CRUSH マップでは、**straw** バケットの中間重量値を計算するのに、古くて最適ではない方法を使用しています。
CACHE_POOL_NO_HIT_SET	1つ以上のキャッシュプールは、使用率を追跡するためのヒットセットで設定されていません。これにより、階層化エージェントがコールドオブジェクトを識別してフラッシュし、キャッシュから削除することができなくなります。**ceph osd pool set_POOL_NAME_ hit_set_type TYPE**、**ceph osd pool set POOL_NAME hit_set_period PERIOD_IN_SECONDS**、**ceph osd pool set POOL_NAME hit_set_count NUMBER_OF_HIT_SETS**、および **ceph osd pool set POOL_NAME hit_set_fpp TARGET_FALSE_POSITIVE_RATE** コマンドを使用して、キャッシュプールのヒットセットを設定します。
OSD_NO_SORTBITWISE	**sortbitwise** フラグが設定されていません。**ceph osd set sortbitwise** コマンドでフラグを設定します。
POOL_FULL	1つまたは複数のプールがクォータに達し、書き込みを許可しなくなりました。**ceph osd pool set-quota POOL_NAME max_objects NUMBER_OF_OBJECTS** および **ceph osd pool set-quota POOL_NAME max_bytes BYTES** を使用してプールのクォータを増やすか、一部の既存のデータを削除して使用率を下げます。

正常性コード	説明
BLUEFS_SPILLOVER	BlueStore バックエンドを使用する 1 つ以上の OSD には db パーティションが割り当てられていますが、その領域が満杯になっているため、メタデータが通常の低速デバイスにあふれ出ています。**ceph config set osd bluestore_warn_on_bluefs_spillover false** コマンドで無効にします。
BLUEFS_AVAILABLE_SPACE	この出力では、BDEV_DB free、BDEV_SLOW free、および available_from_bluestore の 3 つの値が得られます。
BLUEFS_LOW_SPACE	BlueStore File System (BlueFS) の空き容量が少なく、**available_from_bluestore** が少ない場合は、BlueFS のアロケーションユニットのサイズを小さくすることを検討できます。
BLUESTORE_FRAGMENTATION	BlueStore が動作すると、基盤となるストレージの空き領域が断片化されます。これは正常なことであり、避けられないことですが、過度のフラグメント化は速度低下の原因となります。
BLUESTORE_LEGACY_STATFS	BlueStore は、プールごとの詳細ベースで内部使用統計を追跡し、1 つ以上の OSD に BlueStore ボリュームがあります。**ceph config set global bluestore_warn_on_legacy_statfs false** コマンドで警告を無効にします。
BLUESTORE_NO_PER_POOL_OMAP	BlueStore では、プールごとの omap 領域の使用状況を追跡しています。**ceph config set global bluestore_warn_on_no_per_pool_omap false** コマンドで警告を無効にします。
BLUESTORE_NO_PER_PG_OMAP	BlueStore では、PG による omap 領域の利用状況を把握しています。**ceph config set global bluestore_warn_on_no_per_pg_omap false** コマンドで警告を無効にします。
BLUESTORE_DISK_SIZE_MISMATCH	BlueStore を使用している 1 つまたは複数の OSD で、物理デバイスのサイズとそのサイズを追跡するメタデータの間に内部不整合があります。
BLUESTORE_NO_COMPRESSION `	1 つまたは複数の OSD が、BlueStore 圧縮プラグインを読み込むことができません。これは、**ceph-osd** バイナリーが圧縮プラグインと一致しないインストールの失敗、または **ceph-osd** デーモンの再起動を含まない最近のアップグレードが原因である可能性があります。

正常性コード	説明
BLUESTORE_SPURIOUS_READ_ERRORS	BlueStore を使用する 1 つ以上の OSD が、メインデバイスで誤った読み取りエラーを検出します。BlueStore はこれらのエラーに対して、ディスクの読み取りを再試行することで回復しました。

表B.4 デバイスの正常性

正常性コード	説明
DEVICE_HEALTH	1 つ以上のデバイスが近日中に故障することが予想され、警告のしきい値が **mgr/devicehealth/warn_threshold** 設定オプションで制御されます。データを移行するためにデバイスを **out** とマークし、ハードウェアを交換します。
DEVICE_HEALTH_IN_USE	1 つ以上のデバイスがまもなく故障すると予想され **mgr/devicehealth/mark_out_threshold** に基づいてストレージクラスターから out とマークされていますが、まだ 1 つ以上の PG に参加しています。
DEVICE_HEALTH_TOOMANY	すぐに障害が発生するデバイスが多すぎると予想され、**mgr/devicehealth/self_heal** 動作が有効になります。これにより、すべての障害のあるデバイスを out マークすると、クラスターの **mon_osd_min_in_ratio** が超えられ、OSD が多すぎて自動的に **out** とマークされなくなります。

表B.5 プールおよび配置グループ

正常性コード	説明
PG_AVAILABILITY	データの可用性が低下します。つまり、ストレージクラスター内の一部のデータに対する潜在的な読み取りまたは書き込み要求に対応できなくなります。
PG_DEGRADED	一部のデータでデータの冗長性が低下します。これは、複製プールやイレイジャーコードフラグメントについて、ストレージクラスターに必要な数の複製がないことを意味します。
PG_RECOVERY_FULL	ストレージクラスターの空き領域が不足しているため、データの冗長性が低下するか、一部のデータのリスクにさらされる可能性があります。具体的には、1 つ以上の PG に **recovery_toofull** フラグが設定されています。これは、1 つまたは複数の PG が原因でデータを移行または回復できないことを意味します。より多くの OSD が **full** しきい値を超えています。

正常性コード	説明
PG_BACKFILL_FULL	ストレージクラスターの空き領域が不足しているため、データの冗長性が低下するか、一部のデータのリスクにさらされる可能性があります。具体的には、1つ以上の PG に **backfill_toofull** フラグが設定されています。これは、1つまたは複数の PG が原因でデータを移行または回復できないことを意味します。より多くの OSD が **backfillfull** しきい値を超えています。
PG_DAMAGED	データスクラビングにより、ストレージクラスター内のデータの整合性に関するいくつかの問題が発見されました。具体的には、1つ以上の PG で不整合フラグまたは **snaptrim_error** フラグが設定されています。これは、以前のスクラブ操作で問題が検出されたこと、または **repair** フラグが設定されていることを示します。そのような矛盾は現在進行中です。
OSD_SCRUB_ERRORS	最近の OSD のスクラブでは、矛盾点が明らかになりました。
OSD_TOO_MANY_REPAIRS	読み取りエラーが発生し、別のレプリカが利用可能な場合は、そのレプリカを使用してエラーを直ちに修復し、クライアントがオブジェクトデータを取得できるようにします。
LARGE_OMAP_OBJECTS	**osd_deep_scrub_large_omap_object_key_threshold** または **osd_deep_scrub_large_omap_object_value_sum_threshold**、もしくはその両方によって決定されるように、1つまたは複数のプールに大きな omap オブジェクトが含まれています。**ceph config set osd osd_deep_scrub_large_omap_object_key_threshold KEYS** コマンドおよび **ceph config set osd osd_deep_scrub_large_omap_object_value_sum_threshold BYTES** コマンドでしきい値を調整します。
CACHE_POOL_NEAR_FULL	キャッシュ層のプールがほぼ満杯です。**ceph osd pool set CACHE_POOL_NAME target_max_bytes BYTES** コマンドおよび **ceph osd pool set CACHE_POOL_NAME target_max_bytes BYTES** コマンドを使用して、キャッシュプールのターゲットサイズを調整します。
TOO_FEW_PGS	ストレージクラスターで使用されている PG の数が、OSD ごとの **mon_pg_warn_min_per_osd** PG の設定可能なしきい値を下回っています。

正常性コード	説明
POOL_PG_NUM_NOT_POWER_OF_TWO	1 つ以上のプールの **pg_num** の値が 2 の累乗ではありません。**ceph config set global mon_warn_on_pool_pg_num_not_power_of_two false** コマンドで警告を無効にします。
POOL_TOO_FEW_PGS	プールに現在保存されているデータの量に基づいて、1 つ以上のプールにおそらくより多くの PG が必要です。**ceph osd pool set POOL_NAME pg_autoscale_mode off** コマンドで PG の自動スケーリングを無効にするか、**ceph osd pool set POOL_NAME pg_autoscale_mode on** コマンドで PG の数を自動的に調整するか、**ceph osd pool set POOL_NAME pg_num _NEW_PG_NUMBER** コマンドで PG の数を手動で設定します。
TOO_MANY_PGS	ストレージクラスターで使用されている PG の数が、OSD ごとの **mon_max_pg_per_osd** PG の設定可能なしきい値を超えています。ハードウェアを追加して、クラスター内の OSD の数を増やします。
POOL_TOO_MANY_PGS	プールに現在保存されているデータの量に基づいて、1 つ以上のプールにおそらくより多くの PG が必要です。**ceph osd pool set POOL_NAME pg_autoscale_mode off** コマンドで PG の自動スケーリングを無効にするか、**ceph osd pool set POOL_NAME pg_autoscale_mode on** コマンドで PG の数を自動的に調整するか、**ceph osd pool set POOL_NAME pg_num _NEW_PG_NUMBER** コマンドで PG の数を手動で設定します。
POOL_TARGET_SIZE_BYTES_OVERCOMMITTED	1 つまたは複数のプールに、プールの予想サイズを推定するための **target_size_bytes** プロパティーが設定されていますが、その値が利用可能なストレージの合計を超えています。**ceph osd pool set POOL_NAME target_size_bytes 0** コマンドで、プールの値を 0 に設定します。
POOL_HAS_TARGET_SIZE_BYTES_AND_RATIO	1 つ以上のプールに **target_size_bytes** and **target_size_ratio** の両方が設定されており、プールの予想されるサイズを推定しています。**ceph osd pool set POOL_NAME target_size_bytes 0** コマンドで、プールの値を 0 に設定します。
TOO_FEW_OSDS	ストレージクラスター内の OSD 数が、設定可能なしきい値である **o`sd_pool_default_size** を下回っています。

正常性コード	説明
SMALLER_PGP_NUM	1つまたは複数のプールの **pgp_num** の値が **pg_num** 未満です。これは通常、PG 数が増加しても、配置動作が増加していないことを示しています。**ceph osd pool set POOL_NAME pgp_num PG_NUM_VALUE** コマンドを使用して、**pgp_num** を **pg_num** と一致させるように設定することで、この問題を解決します。
MANY_OBJECTS_PER_PG	1つ以上のプールには、PG ごとのオブジェクトの平均数があり、ストレージクラスター全体の平均よりもはるかに高くなっています。具体的なしきい値は、**mon_pg_warn_max_object_skew** 設定値によって制御されます。
POOL_APP_NOT_ENABLED	1つまたは複数のオブジェクトを含むプールが存在しますが、特定のアプリケーションで使用するためのタグが付けられていません。この警告を解決するには、**rbd pool init POOL_NAME** コマンドでアプリケーションが使用するプールをラベル付けします。
POOL_FULL	1つ以上のプールがクォーターに達しています。このエラー状態を引き起こすための閾値は、**mon_pool_quota_crit_threshold** 設定オプションで制御されます。
POOL_NEAR_FULL	1つまたは複数のプールが、設定された満杯のしきい値に近づいています。**ceph osd pool set-quota POOL_NAME max_objects NUMBER_OF_OBJECTS** コマンドおよび **ceph osd pool set-quota POOL_NAME max_bytes BYTES** コマンドでプールのクォータを調整します。
OBJECT_MISPLACED	ストレージクラスターの1つまたは複数のオブジェクトが、ストレージクラスターが保存したいノードに保存されていません。これは、最近行われたストレージクラスターの変更によるデータの移行が完了していないことを示しています。
OBJECT_UNFOUND	ストレージクラスター内に1つ以上のオブジェクトが見つかりません。具体的には、OSD はオブジェクトの新しいコピーまたは更新されたコピーが存在する必要があることを認識していますが、そのバージョンのオブジェクトのコピーが現在オンラインの OSD で見つかりません。

正常性コード	説明
SLOW_OPS	1つ以上の OSD またはモニター要求の処理に長い時間がかかっている。これは、極端な負荷、遅いストレージデバイス、またはソフトウェアのバグを示している可能性があります。
PG_NOT_SCRUBBED	1つ以上の PG が最近スクラブされていません。PG は通常、**osd_scrub_max_interval** でグローバルに指定された設定間隔でスクラブされます。**ceph pg scrub PG_ID** コマンドでスクラブを開始します。
PG_NOT_DEEP_SCRUBBED	1つ以上の PG が最近ディープスクラビングされていません。**ceph pg deep-scrub PG_ID** コマンドでスクラブを開始します。PG は通常、**osd_deep_scrub_interval** 秒ごとにスクラブされます。この警告は、間隔の **mon_warn_pg_not_deep_scrubbed_ratio** パーセンテージが、期限が切れてからスクラブなしで経過したときにトリガーされます。
PG_SLOW_SNAP_TRIMMING	1つまたは複数の PG のスナップショットトリムキューが、設定された警告しきい値を超えました。これは、非常に多くのスナップショットが最近削除されたか、OSD が新しいスナップショットの削除率に追いつくのに十分な速さでスナップショットをトリミングできないことを示しています。

表B.6 その他

正常性コード	説明
RECENT_CRASH	1つ以上の Ceph デーモンが最近クラッシュしましたが、そのクラッシュは管理者によってまだ確認されていません。
TELEMETRY_CHANGED	テレメトリーが有効になっていますが、その時点からテレメトリーレポートの内容が変更されているため、テレメトリーレポートは送信されません。
AUTH_BAD_CAPS	1つまたは複数の認証ユーザーに、モニターが解析できない機能があります。**ceph auth ENTITY_NAME DAEMON_TYPE CAPS** コマンドでユーザーの能力を更新します。

正常性コード	説明
OSD_NO_DOWN_OUT_INTERVAL	**mon_osd_down_out_interval** オプションがゼロに設定されています。これは、OSD に障害が発生した後、システムが修復操作を自動的に実行しないことを意味します。**ceph config global mon mon_warn_on_osd_down_out_interval_zero false** コマンドで間隔をなくします。
DASHBOARD_DEBUG	ダッシュボードのデバッグモードが有効になっています。つまり、REST API 要求の処理中にエラーが発生した場合、HTTP エラーレスポンスには Python のトレースバックが含まれています。**ceph dashboard debug disable** コマンドでデバッグモードを無効にします。