Red Hat Ceph Storage 7

エッジガイド

Red Hat Ceph Storage の Edge クラスターに関するガイド

Last Updated: 2024-03-28

Red Hat Ceph Storage 7 エッジガイド

Red Hat Ceph Storage の Edge クラスターに関するガイド

法律上の通知

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at
http://creativecommons.org/licenses/by-sa/3.0/
. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

このドキュメントでは、コスト効率の高いオブジェクトストレージ設定のソリューションである Edge クラスターに関する情報を提供します。 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、Red Hat CTO である Chris Wright のメッセージ をご覧ください。

目次

第1章 EDGE クラスター

Edge クラスターは、コスト効率の高いオブジェクトストレージ設定のためのソリューションです。

Red Hat は、次の Red Hat Ceph Storage クラスターの最小設定をサポートします。

- SSD の 2 つのレプリカを持つ 3 ノードクラスター。
- HDD のレプリカが 3 つある 4 ノードクラスター。
- 2+2 設定の EC プールを備えた 4 ノードクラスター。

クラスターが小さくなると、使用量および復元力損失のため、使用率が低下します。

第2章 プールの概要

Ceph クライアントは、データをプールに保存します。プールの作成時に、クライアントがデータを保存するための I/O インターフェイスを作成します。

Ceph クライアント、つまりブロックデバイス、ゲートウェイ、その他の観点から見ると、Ceph Storage クラスターとの対話は非常に簡単です。

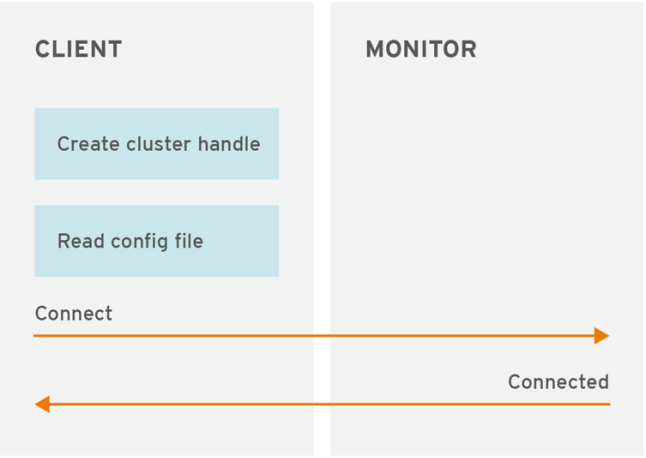
- クラスターハンドルを作成します。
- クラスターハンドルをクラスターに接続します。
- オブジェクトとその拡張属性を読み書きするための I/O コンテキストを作成します。

クラスターハンドルの作成とクラスターへの接続

Ceph Storage クラスターに接続するには、Ceph クライアントに次の詳細が必要です。

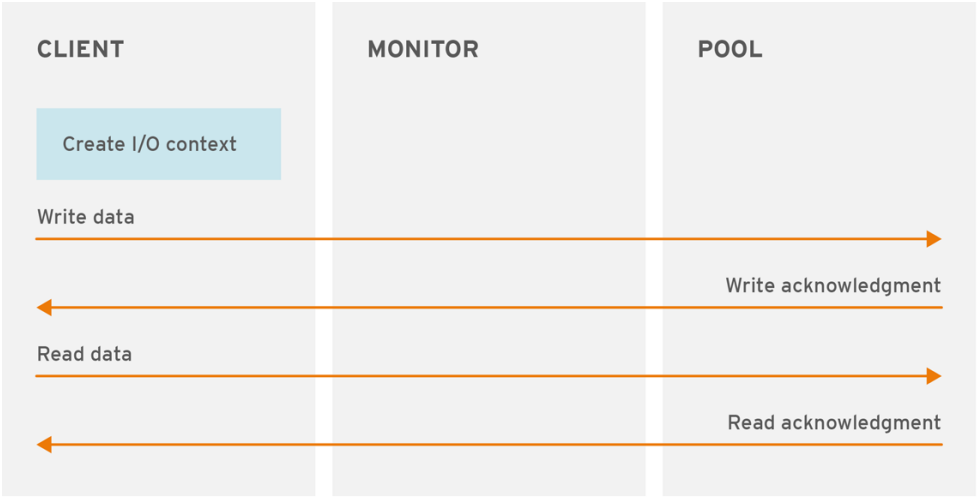
- クラスター名 (デフォルトでは Ceph) - 曖昧に聞こえるため、通常は使用しません。
- 初期モニターアドレス。

通常、Ceph クライアントは Ceph 設定ファイルのデフォルトパスを使用してパラメーターを取得し、ファイルからファイルを読み取りますが、コマンドラインでパラメーターを指定することもできます。Ceph クライアントは、ユーザー名と秘密鍵も提供します。認証はデフォルトで **On** になっています。次に、クライアントは Ceph monitor クラスターに接続し、モニター、OSD、およびプールを含むクラスターマップの最新コピーを取得します。



プール I/O コンテキストの作成

データを読み書きするために、Ceph クライアントは Ceph Storage クラスター内の特定のプールへの I/O コンテキストを作成します。指定したユーザーにプールのパーミッションがある場合は、Ceph クライアントは指定されたプールから読み取り/書き込みを行うことができます。



Ceph のアーキテクチャーを使用することで、ストレージクラスターは、プール名を指定して簡単に定義し、I/O コンテキストの作成で簡単に定義するストレージストラテジーのいずれかをクライアントが選択できるように、ストレージクラスターを Ceph クライアントに提供することができます。ストレージストラテジーはすべて、容量およびパフォーマンスにおいて Ceph クライアントを認識しません。同様に、Ceph クライアントの複雑性 (例: ブロックデバイス表現へのオブジェクトのマッピング、S3/Swift RESTful サービスの提供) は Ceph ストレージクラスターに見えません。

プールは、復元力、配置グループ、CRUSH ルール、およびクォータを提供します。

- **耐障害性:** データを損失せずに失敗した OSD の数を設定できます。複製されたプールの場合、これはオブジェクトのコピーまたはレプリカの任意数です。通常の設定では、オブジェクトと 1 つの追加コピー (例: **size = 2**) が保存されますが、コピーまたはレプリカの数を決定できます。イレイジャーコードプールの場合は、コーディングしたチャンクの数です (例: **イレイジャーコードプロファイル** の **m=2**)。

- **配置グループ**: プールの配置グループの数を設定できます。典型的な設定では、OSD ごとに約 50-100 の配置グループを使用して、最高のコンピューティングリソースを使用せずに最適なバランスを提供します。複数のプールを設定する場合は、全体としてプールとクラスターの両方に妥当な配置グループ数を設定するように注意してください。

- **CRUSH ルール**: プールにデータをプールに保存すると、CRUSH ルールがプールにマッピングされた CRUSH ルールにより、CRUSH が各オブジェクトとそのレプリカ (またはイレイジャーコード化されたプールのチャンク) の配置のルールを特定できます。プールにカスタム CRUSH ルールを作成できます。

- **クォータ**: **ceph osd pool set-quota** コマンドを使用してプールにクォータを設定すると、指定したプールに保存されるオブジェクトの最大数または最大バイト数が制限される場合があります。

第3章 耐久性のあるデータプールと耐久性のないデータプール

耐久性のあるプールと耐久性のなるプールとは何ですか?

耐久性のあるデータプールにより、データの複製やエンコードが可能になります。

耐久性のないデータプールでは、データの複製やエンコードができません。

耐久性のないプールは、**replica1** とも呼ばれ、データ損失から保護されません。耐久性のないデータプールを持つ小規模クラスターは独自のレプリケーションを実行し、独自のレプリケーションを実行するため Red Hat Ceph Storage からのレプリケーションを必要としません。

データプールを使用したクラスターの使用率

設定が小さくなると、耐久性が失われるため、クラスターの使用率が低下します。回復はホストの使用率によって制限され、実稼働環境の 1 秒あたりの入出力操作数 (IOPS) に影響を与える可能性があります。1 つのノードに障害が発生した場合は、3 番目のノードを追加してホストの使用率を制限し、Red Hat Ceph Storage を完全なレプリケーションに回復できます。

第4章 CEPH イレイジャーコーディング

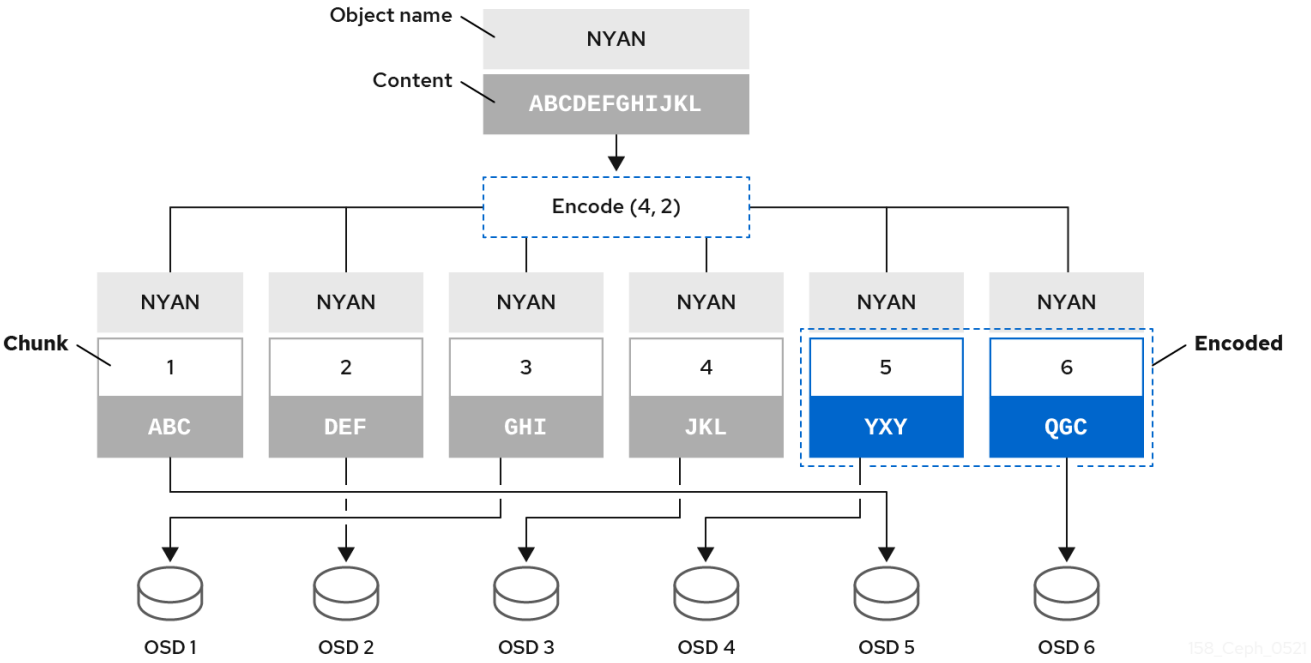
Ceph は、多くのイレイジャーコードアルゴリズムのいずれかを読み込むことができます。**Reed-Solomon** アルゴリズムが最も早く一般的に使用されるものになります。イレイジャーコードは、実際には前方誤り訂正 (FEC: forward error correction) コードです。FEC コードは、**K** チャンクのメッセージを、**N** チャンクのコードワードと呼ばれる長いメッセージに変換し、Ceph が **N** チャンクのサブセットから元のメッセージを復元できるようにします。

具体的には、変数 **K** が元のデータチャンク量である N = **N = K+M** です。変数 **M** は、余分または冗長なチャンクを表し、イレイジャーコードアルゴリズムが障害から保護します。変数 **N** は、イレイジャーコーディングプロセス後に作成されたチャンクの合計数です。**M** の値は **N-K** です。これは、アルゴリズムが **K** の元のデータチャンクから **N-K** 冗長チャンクを計算することを意味します。このアプローチにより、Ceph が元のデータすべてにアクセスできることを保証します。システムが任意の **N-K** の障害に対して回復性があります。たとえば、16 の **N** 設定のうち 10 **K**、またはイレイジャーコーディング **10/16** の場合、イレイジャーコードアルゴリズムは 10 ベースチャンク **K** に 6 つの追加チャンクを追加します。たとえば、**M = K-N** または **16-10 = 6** の設定では、Ceph は 16 チャンク **N** を 16 OSD に分散します。元のファイルは、6 つの OSD に障害が発生した場合でも、検証済みの 10 個の **N** チャンクから再構築できます。これにより、Red Hat Ceph Storage クラスターがデータを失うことがなくなり、非常に高いレベルのフォールトトレランスが保証されます。

複製されたプールと同様に、イレイジャーコーディングされたプールでは、セットアップ内のプライマリー OSD がすべての書き込み操作を受け取ります。複製されたプールでは、Ceph はセットのセカンダリー OSD 上の配置グループで各オブジェクトのディープコピーを作成します。イレイジャーコーディングの場合、プロセスは少し異なります。コード化されたプールは各オブジェクトを **K+M** チャンクとして格納します。これは **K** データチャンクと **M** コーディングチャンクに分割されます。プールには **K+M** のサイズが設定され、これにより Ceph が各チャンクを動作セットの OSD に保管することができます。Ceph は、チャンクのランクをオブジェクトの属性として保存します。プライマリー OSD はペイロードを **K+M** チャンクにエンコードし、それらを他の OSD に送信します。プライマリー OSD は、配置グループログの権威バージョンを維持するロールも果たします。

たとえば、一般的な設定では、システム管理者は 6 つの OSD を使用し、そのうちの 2 つの OSD の損失を維持するために、イレイジャーコード化されたプールを作成します。つまり、(**K+M = 6**) であり、(**M = 2**) になります。

Ceph が **ABCDEFGHIJKL** を含むオブジェクト **NYAN** をプールに書き込む場合、イレイジャーエンコーディングアルゴリズムは、コンテンツを **ABC**、**DEF**、**GHI**、および **JKL** の 4 つの部分に分割するだけで、コンテンツを 4 つのデータチャンクに分割します。コンテンツの長さが **K** の倍数でない場合は、アルゴリズムによりコンテンツをパディングします。この関数は、2 つのコーディングチャンクも作成します。4 つ目は **YXY**、5 つ目は **QGC** が付きます。Ceph は、動作セット内の OSD 上にそれぞれのチャンクを保存します。ここで、**NYAN** の名前を持つオブジェクトにチャンクが保管されますが、異なる OSD にあります。アルゴリズムは、名前に加えて、チャンクをオブジェクト **shard_t** の属性として作成した順番を保持する必要があります。たとえば、チャンク 1 には **ABC** が含まれ、Ceph はこれを **OSD5** に格納されます。一方、チャンク 5 には **YXY** が含まれ、**OSD4** に格納されます。



リカバリーのシナリオでは、クライアントはチャンク 1 から 6 を読み取ることで、イレイジャーコーディングされたプールからオブジェクト **NYAN** を読み込もうとします。OSD は、2 と 6 のチャンクがないことをアルゴリズムに通知します。これらのチャンクはイレイジャーと呼ばれます。たとえば、**OSD6** が除外されているため、プライマリー OSD はチャンク 6 を読み取ることができませんでした。また、**OSD2** は最も遅く、そのチャンクを考慮していなかったため、チャンク 2 を読み取ることができませんでした。ただし、アルゴリズムに 4 つのチャンクがあるとすぐに、**ABC** を含むチャンク 1、**GHI** を含むチャンク 3、**JKL** を含むチャンク 4、および **YXY** を含むチャンク 5 の 4 つのチャンクが読み取られます。次に、オブジェクト **ABCDEFGHIJKL** の元のコンテンツと、**QGC** を含むチャンク 6 の元のコンテンツを再構築します。

データをチャンクに分割することは、オブジェクトの配置とは無関係です。CRUSH ルールセットとイレイジャーコーディングされたプールプロファイルにより、OSD 上のチャンクの配置が決定されます。たとえば、イレイジャーコードプロファイルで Locally Repairable Code (**lrc**) プラグインを使用すると、追加のチャンクが作成され、回復に必要な OSD が少なくなります。たとえば、**lrc** プロファイル設定 **K=4 M=2 L=3** では、アルゴリズムは、**jerasure** プラグインと同じように 6 つのチャンク (**K+M**) を作成しますが、局所性の値 (**L=3**) では、アルゴリズムはさらに 2 つのチャンクをローカルに作成する必要があります。アルゴリズムは、**(K+M)/L** などの追加チャンクを作成します。チャンク 0 を含む OSD が失敗すると、チャンク 1、2、および最初のローカルチャンクを使用してこのチャンクを回復できます。この場合、アルゴリズムは 5 つではなく 3 つのチャンクのみを回復に必要とします。

注記

イレイジャーコーディングされたプールを使用すると、オブジェクトマップが無効になります。

重要

2+2 設定の消去符号化プールの場合は、入力文字列を **ABCDEFGHIJKL** から **ABCDEF** に置き換え、コーディングチャンクを **4** から **2** に置き換えます。

関連情報

- CRUSH、イレイジャーコーディングプロファイル、およびプラグインの詳細は、Red Hat Ceph Storage 7 の ストレージストラテジーガイド を参照してください。

- オブジェクトマップの詳細については、**Ceph クライアントオブジェクトマップ** のセクションを参照してください。

第5章 イレイジャーコードプールの概要

Ceph はデフォルトで複製されたプールを使用します。これにより、Ceph はすべてのオブジェクトをプライマリー OSD ノードから 1 つ以上のセカンダリー OSD にコピーします。イレイジャーコーディングされたプールは、データの持続性を確保するのに必要なディスク容量を減らしますが、レプリケーションよりもコストが高くなります。

Ceph ストレージストラテジーには、データの持続性要件を定義します。データの持続性とは、データが失われることなく、1 つまたは複数の OSD の損失を持続させることができることを意味します。

Ceph は、データをプールに保存します。プールには 2 種類のプールがあります。

- **replicated**
- **erasure-coded**

イレイジャーコーディングは、Ceph ストレージクラスターにオブジェクトを大幅に格納する方法であり、イレイジャーコードアルゴリズムによりオブジェクトがデータチャンク (**k**)、およびコーディングチャンク (**m**) に分割され、これらのチャンクを異なる OSD に保存します。

OSD に障害が発生すると、Ceph は他の OSD から残りのデータ (**k**) およびコーディング (**m**) チャンクを取得し、イレイジャーコードアルゴリズムはこれらのチャンクからオブジェクトを復元します。

注記

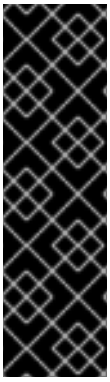
Red Hat は、書き込みやデータの損失を防ぐために、イレイジャーコーディングされたプールの **min_size** を **K+1** 以上にすることを推奨します。

イレイジャーコーディングは、レプリケーションよりもストレージ容量をより効率的に使用します。n 個のレプリケーションアプローチは、オブジェクトの **n** 個のコピーを維持するのに対し (Ceph のデフォルトは 3x)、イレイジャーコーディングは **k** + **m** チャンクのみを保持します。たとえば、3 データと 2 つのブロックのチャンクは、元のオブジェクトの 1.5x のストレージ領域を使用します。

イレイジャーコーディングはレプリケーションと比べてストレージのオーバーヘッドが少なく、イレイジャーコードアルゴリズムは、オブジェクトへのアクセスや復旧時に、レプリケーションよりも多くの RAM および CPU を使用します。イレイジャーコーディングは、データストレージが永続的であり、耐障害性になければならないものの、高速な読み取り (たとえば、コールドマイグレーションストレージ、履歴レコードなど) を必要としない場合に役立ちます。

Ceph でイレイジャーコードがどのように機能するかの詳細は、Red Hat Ceph Storage 7 の **アーキテクチャーガイド** の Ceph イレイジャーコーディング セクションを参照してください。

k=2 および **m=2** を使用してクラスターを初期化する際に、Ceph は **デフォルト** のイレイジャーコードプロファイルを作成します。つまり、Ceph は 3 つの OSD (**k+m == 4**) にオブジェクトデータを分散し、Ceph がこれらの OSD のいずれかを、データを失うことなく失う可能性があることを意味します。イレイジャーコードのプロファイリングの詳細は、イレイジャーコードプロファイル セクションを参照してください。



重要

.rgw.buckets プールのみをイレイジャーコーディング済みとして設定し、その他のすべての Ceph Object Gateway プールをレプリケート済みとして設定すると、新しいバケットを作成しようとすると以下のエラーで失敗します。

```
set_req_state_err err_no=95 resorting to 500
```

このため、イレイジャーコーディングされたプールは **omap** 操作をサポートしません。特定の Ceph Object Gateway メタデータプールには **omap** サポートが必要です。

5.1. イレイジャーコーディングされたプールのサンプルの作成

イレイジャーコーディングプールを作成し、配置グループを指定します。

最も簡単なイレイジャーコードプールは RAID5 と同等で、少なくとも 4 台のホストが必要です。2+2 プロファイルを使用してイレイジャーコーディングされたプールを作成できます。

手順

1. 2+2 設定の 4 つのノード上のイレイジャーコーディングされたプールに対して次の設定を設定します。

   構文

   ```
   ceph config set mon mon_osd_down_out_subtree_limit host
   ceph config set osd osd_async_recovery_min_cost 1099511627776
   ```

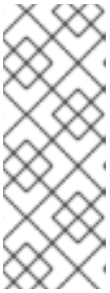
   重要

   一般に、イレイジャーコーディングプールではこれは必要ありません。

   重要

   非同期リカバリーのコストは、レプリカ上で遅れている PG ログエントリーの数と、失われたオブジェクトの数です。**osd_target_pg_log_entries_per_osd** は **30000** です。したがって、単一の PG を持つ OSD には **30000** のエントリーが存在する可能性があります。**osd_async_recovery_min_cost** は 64 ビットの整数であるため、2+2 設定の EC プールの場合は、**osd_async_recovery_min_cost** の値を **1099511627776** に設定します。

   注記

   4 つのノードを持つ EC クラスターの場合、K+M の値は 2+2 です。ノードに完全な障害が発生した場合、ノードは 4 つのチャンクとして回復されず、使用できるノードは 3 つだけになります。**mon_osd_down_out_subtree_limit** の値を **host** に設定すると、ホストのダウンシナリオ中に OSD がマークアウトされなくなり、データの再バランシングやノードが再び起動するまでの待機が防止されます。

2. 2+2 設定のイレイジャーコーディングされたプールの場合は、プロファイルを設定します。

   構文

```
ceph osd erasure-code-profile set ec22 k=2 m=2 crush-failure-domain=host
```

例

```
[ceph: root@host01 /]# ceph osd erasure-code-profile set ec22 k=2 m=2 crush-failure-domain=host

Pool : ceph osd pool create test-ec-22 erasure ec22
```

3. イレイジャーコーディングされたプールを作成します。

例

```
[ceph: root@host01 /]# ceph osd pool create ecpool 32 32 erasure

pool 'ecpool' created
$ echo ABCDEFGHI | rados --pool ecpool put NYAN -
$ rados --pool ecpool get NYAN -
ABCDEFGHI
```

32 は配置グループの数です。

第6章 バックエンド圧縮

圧縮オプションを使用して、より小さい容量のエッジクラスターを圧縮します。

BlueStore では 2 種類の圧縮が可能です。

- 一般的なワークロードの BlueStore レベルの圧縮。
- S3 ワークロードの Ceph Object Gateway の圧縮レベル。

圧縮アルゴリズムの詳細は、プール値 を参照してください。

圧縮を有効にし、プールで圧縮を有効にしたときにクラスターでクラッシュが発生しないようにする必要があります。

次の方法で、エッジクラスターのプールで圧縮を有効にできます。

- 次のコマンドを使用して、サポートされている圧縮アルゴリズム (snappy、zlib、zstd など) を有効にし、サポートされている圧縮モード (**None**、**passive**、**aggressive**、**force** など) を有効にします。

  構文

  ```
  ceph osd pool set POOL_NAME compression_algorithm ALGORITHM
  ceph osd pool set POOL_NAME compression_mode MODE
  ```

- 次のコマンドを使用して、さまざまな圧縮率を有効にします。

  構文

  ```
  ceph osd pool set POOL_NAME compression_required_ratio RATIO
  ceph osd pool set POOL_NAME compression_min_blob_size SIZE
  ceph osd pool set POOL_NAME compression_max_blob_size SIZE
  ```

- 3 つのプールを作成し、それらのプールで異なる圧縮を有効にして、プールで IO 停止が発生しないようにします。
- プールに圧縮を作成せずに 4 番目のプールを作成します。圧縮を使用してプールと同じ量のデータを書き込みます。圧縮を使用したプールは、圧縮を使用しないプールよりも使用する RAW スペースが少なくなります。

これらのアルゴリズムが設定されていることを **確認する** には、**ceph osd pool get POOL_NAME OPTION_NAME** コマンドを使用します。

これらのアルゴリズムの **設定を解除する** には、適切なオプションを指定して **ceph osd pool unset POOL_NAME OPTION_NAME** コマンドを使用します。

第7章 クラスタートポロジーとコロケーション

必要なトポロジーと、エッジクラスターのコロケーションで考慮すべき要素を理解します。

クラスタートポロジー、OpenStack とのハイパーコンバージェンス、OpenStack 上のノードのコロケーション、および OpenStack の最小設定の制限は、HCI の Ceph 設定オーバーライド を参照してください。

コロケーションの詳細は、コロケーション を参照してください。