



Red Hat Hyperconverged Infrastructure for Cloud 13

オペレーションガイド

Red Hat Hyperconverged Infrastructure for Cloud Solution の操作

Red Hat Hyperconverged Infrastructure for Cloud 13 オペレーションガイド

Red Hat Hyperconverged Infrastructure for Cloud Solution の操作

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

法律上の通知

Copyright © 2023 | You need to change the HOLDER entity in the en-US/Operations_Guide.ent file |.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

このドキュメントでは、Red Hat OpenStack Platform 13 および Red Hat Ceph Storage 3 を使用して、AMD64 および Intel 64 のアーキテクチャーで実行している Red Hat Hyperconverged Infrastructure for Cloud ソリューションを操作する手順について説明します。

目次

第1章 RED HAT HYPERCONVERGED INFRASTRUCTURE CLOUD 運用タスクの実行	4
第2章 オーバークラウド設定の更新	5
第3章 オーバークラウドへのノードの追加	6
第4章 オーバークラウドからのノードの削除	10
4.1. 前提条件	10
4.2. ストレージクラスターからの CEPH OSD サービスの削除	10
4.3. オーバークラウドからの NOVA コンピューティングサービスの削除	13
4.4. 関連資料	14
第5章 CEPH ブロックデバイスミラーリングの使用	16
5.1. 前提条件	16
5.2. CEPH ブロックデバイスのミラーリング	16
rbd-mirror デーモン	16
ミラーリングのモード	17
イメージの状態	17
非同期 Red Hat Ceph Storage 更新	17
5.3. 同じ名前のストレージクラスター間のミラーリングの設定	18
前提条件	18
手順	18
5.4. ミラーリングのための CEPH BLOCK DEVICE JOURNALING の有効化	19
前提条件	19
手順	19
関連情報	20
5.5. プールでの CEPH ブロックデバイスミラーリングの設定	20
5.5.1. 前提条件	20
5.5.2. プールでのミラーリングの有効化	20
前提条件	20
手順	20
関連情報	21
5.5.3. プールでのミラーリングの無効化	21
前提条件	21
手順	21
関連情報	21
5.5.4. クラスターピアの追加	22
5.5.5. クラスターピアの削除	22
前提条件	22
手順	22
関連情報	23
5.5.6. クラスターピアに関する情報の表示	23
前提条件	23
手順	23
関連情報	23
5.5.7. プールのミラーリングステータスの取得	23
前提条件	24
手順	24
関連情報	24
5.5.8. 関連情報	24
5.6. イメージでの CEPH BLOCK デバイスミラーリングの設定	24
5.6.1. 前提条件	24

5.7. イメージミラーリングの有効化	25
前提条件	25
手順	25
関連情報	25
5.7.1. 関連情報	25
5.8. CEPH BLOCK デバイスの双方向ミラーリングの設定	25
前提条件	26
手順	26
関連情報	31
5.9. ストレージクラスター間のレプリケーションの遅延	31
前提条件	31
手順	31
関連情報	32
5.10. 災害からの復旧	32
前提条件	32
手順	32
関連情報	34
5.11. 関連情報	34

第1章 RED HAT HYPERCONVERGED INFRASTRUCTURE CLOUD 運用タスクの実行

Red Hat Hyperconverged Infrastructure (RHHI) Cloud ソリューションには、次の3つの基本的な運用タスクがあります。

- オーバークラウド設定の [更新](#)
- オーバークラウドへの [ノードの追加](#)
- オーバークラウドからの [ノードの削除](#)

第2章 オーバークラウド設定の更新

Red Hat Hyperconverged Infrastructure (RHHI) for Cloud 設定を更新して新しい機能を追加したり、オーバークラウドの機能を変更したりする必要がある場合があります。

前提条件

- 実行中の RHHI for Cloud ソリューション。

手順

Red Hat OpenStack Platform director ノードで、**stack** ユーザーとして以下の手順を実施します。

1. 最初のオーバークラウドデプロイメントと同じ TripleO Heat テンプレートを使用して、**openstack overcloud deploy** コマンドを再実行します。

例

```
[stack@director ~]$ openstack overcloud deploy --templates \  
-r ~/custom-templates/custom-roles.yaml \  
-e /usr/share/openstack-tripleo-heat-templates/environments/puppet-pacemaker.yaml \  
-e /usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml \  
-e /usr/share/openstack-tripleo-heat-templates/environments/storage-environment.yaml \  
-e ~/templates/network.yaml \  
-e ~/templates/ceph.yaml \  
-e ~/templates/compute.yaml \  
-e ~/templates/layout.yaml
```



注記

新しい環境ファイルをオーバークラウドに追加する場合は、追加の **-e** 引数を **openstack overcloud deploy** コマンドに追加します。

関連情報

- [Red Hat Hyperconverged Infrastructure for Cloud デプロイメントガイド](#) .
- Red Hat OpenStack Platform 10 Director のインストールおよび使用 [ガイド](#) .

第3章 オーバークラウドへのノードの追加

オーバークラウドは、新しい Nova コンピューティングおよび Ceph OSD ノードをオーバークラウドに追加することで、需要の増加に合わせて拡張できます。

前提条件

- 実行中の RHHI クラウドソリューション。
- ネットワークインターフェイスカード (NIC) の MAC アドレス。
- IPMI ユーザー名およびパスワード

手順

Red Hat OpenStack Platform director ノードで、**stack** ユーザーとして以下の手順を実施します。

1. 新しいノードを管理するために Ironic サービスのホスト定義ファイルを作成し、設定します。
 - a. 新しい JSON ホスト定義ファイルを作成します。

```
[stack@director ~]$ touch ~/new_node.json
```

- b. 以下のテンプレートを使用して、**nodes** スタンザの角括弧 (**{ "nodes": [] }**) の間に新しいノードの定義ブロックを追加します。

```
{
  "pm_password": "IPMI_USER_PASSWORD",
  "name": "NODE_NAME",
  "pm_user": "IPMI_USER_NAME",
  "pm_addr": "IPMI_IP_ADDR",
  "pm_type": "pxe_ipmitool",
  "mac": [
    "NIC_MAC_ADDR"
  ],
  "arch": "x86_64",
  "capabilities": "node:_NODE_ROLE-INSTANCE_NUM_boot_option:local"
}
```

以下を置き換えます。

- **IPMI_USER_PASSWORD** と IPMI パスワード。
- ノードのわかりやすい名前を持つ **NODE_NAME**。これはオプションのパラメーターです。
- **IPMI_USER_NAME** は、ノードの電源をオンまたはオフにすることができる IPMI ユーザー名に置き換えます。
- **IPMI_IP_ADDR** は、IPMI IP アドレスに置き換えます。
- PXE ブートを処理するネットワークカードの MAC アドレスを持つ **NIC_MAC_ADDR**。
- **NODE_ROLE-INSTANCE_NUM** とノードのロール、およびノード番号。このソリューションでは、**controller** と **osd-compute** の 2 つのロールを使用します。

例

```
{
  "nodes": [
    {
      "pm_password": "AbC1234",
      "name": "m630_slot2",
      "pm_user": "ipmiadmin",
      "pm_addr": "10.19.143.62",
      "pm_type": "pxe_ipmitool",
      "mac": [
        "c8:1f:66:65:33:42"
      ],
      "arch": "x86_64",
      "capabilities": "node:osd-compute-3,boot_option:local"
    }
  ]
}
```

2. ノードを Ironic データベースにインポートします。

```
[stack@director ~]$ openstack baremetal import ~/new_node.json
```

- a. **openstack baremetal import** コマンドが、新しいノードで Ironic データベースに投入されていることを確認します。

```
[stack@director ~]$ openstack baremetal node list
```

3. 新しいノードをメンテナンスモードに設定します。

```
ironic node-set-maintenance $UUID true
```

以下を置き換えます。

- **\$UUID** を新しいノードの UUID に置き換えます。ステップ 2a の出力を参照して、新しいノードの UUID を取得します。

例

```
[stack@director ~]$ ironic node-set-maintenance 7250678a-a575-4159-840a-e7214e697165 true
```

4. 新しいノードのハードウェアを検査します。

```
openstack baremetal introspection start $UUID
```

以下を置き換えます。

- **\$UUID** を新しいノードの UUID に置き換えます。ステップ 2a の出力を参照して、新しいノードの UUID を取得します。

例

```
[stack@director ~]$ openstack baremetal introspection start 7250678a-a575-4159-840a-e7214e697165 true
```

- a. イントロスペクションプロセスが完了するまでに時間がかかる場合があります。イントロスペクションプロセスのステータスを確認します。

```
[stack@director ~]$ openstack baremetal introspection bulk status
```

出力例

```
+-----+-----+-----+
| Node UUID           | Finished | Error |
+-----+-----+-----+
| a94b75e3-369f-4b2d-b8cc-8ab272e23e89 | True    | None |
| 7ace7b2b-b549-414f-b83e-5f90299b4af3 | True    | None |
| 8be1d83c-19cb-4605-b91d-928df163b513 | True    | None |
| e8411659-bc2b-4178-b66f-87098a1e6920 | True    | None |
| 04679897-12e9-4637-9998-af8bee30b414 | True    | None |
| 48b4987d-e778-48e1-ba74-88a08edf7719 | True    | None |
| 7250678a-a575-4159-840a-e7214e697165 | True    | None |
+-----+-----+-----+
```

5. 新しいノードでメンテナンスモードを無効にします。

```
ironic node-set-maintenance $UUID false
```

以下を置き換えます。

- **\$UUID** を新しいノードの UUID に置き換えます。ステップ 2a の出力を参照して、新しいノードの UUID を取得します。

例

```
[stack@director ~]$ ironic node-set-maintenance 7250678a-a575-4159-840a-e7214e697165 false
```

6. 完全なオーバークラウドカーネルと RAM ディスクイメージを新しいノードに割り当てます。

```
[stack@director ~]$ openstack baremetal configure boot
```

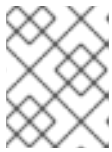
7. `~/templates/layout.yaml` ファイルを開いて編集します。
 - a. **parameter_defaults** セクションで、**OsdComputeCount** オプションを **3** から **4** に変更します。
 - b. **OsdComputeIPs** セクションで、分離されたネットワークごとに新しいノードの IP アドレスを追加します。
8. 最初のオーバークラウドデプロイメントと同じ TripleO Heat テンプレートを使用して **openstack overcloud deploy** コマンドを再実行し、新しいオーバークラウド設定を適用します。

179

```
[stack@director ~]$ openstack overcloud deploy --templates \  
-r ~/templates/custom-roles.yaml \  
-e /usr/share/openstack-tripleo-heat-templates/environments/puppet-pacemaker.yaml \  
-e /usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml \  
-e /usr/share/openstack-tripleo-heat-templates/environments/storage-environment.yaml \  
-e ~/templates/network.yaml \  
-e ~/templates/ceph.yaml \  
-e ~/templates/compute.yaml \  
-e ~/templates/layout.yaml
```

9. 新しいノードの追加を確認します。

```
[stack@director ~]$ openstack server list
```



注記

ノードのステータスが **ACTIVE** の場合、新しいノードはオーバークラウドに正常に追加されています。

第4章 オーバークラウドからのノードの削除

Red Hat OpenStack Platform director (RHOSP-d) は、Red Hat Ceph Storage (RHCS) ノードの自動削除をサポートしていません。

4.1. 前提条件

- ワークロードを処理するのに十分な CPU と RAM があることを確認します。
- 削除するノードからコンピューターワークロードを移行します。
- **HEALTH_OK** のステータスを維持するのに十分な予備ストレージ容量がストレージクラスターにあることを確認します。

4.2. ストレージクラスターからの CEPH OSD サービスの削除

この手順により、Ceph OSD サービスがストレージクラスターのメンバーから削除されます。

前提条件

- 正常な Ceph ストレージクラスター

手順

特に明記されていない限り、コントローラー/モニターノードの1つで、**root** ユーザーとして次の手順を実行します。

1. Ceph Storage クラスターのヘルスステータスを確認します。

```
[root@controller ~]# ceph health
```

この手順を続行する前に、ヘルスステータスが **HEALTH_OK** になっている必要があります。



警告

ceph health コマンドがストレージクラスターが **near full** であると報告した場合、Ceph OSD を削除すると、フル比率の制限を超えるか、それに達する可能性があります。これにより、データが失われる可能性があります。ストレージクラスターが **near full** の場合は、先に進む前に [Red Hat サポート](#) に連絡してください。

2. 削除する Ceph OSD の数を決定します。

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

出力例

```
ID WEIGHT TYPE NAME UP/DOWN REWEIGHT PRIMARY-AFFINITY
-1 52.37256 root default
```

```
-2 13.09314 host overcloud-osd-compute-3
 0 1.09109  osd.0          up 1.00000  1.00000
 4 1.09109  osd.4          up 1.00000  1.00000
 8 1.09109  osd.8          up 1.00000  1.00000
12 1.09109  osd.12         up 1.00000  1.00000
16 1.09109  osd.16         up 1.00000  1.00000
20 1.09109  osd.20         up 1.00000  1.00000
24 1.09109  osd.24         up 1.00000  1.00000
28 1.09109  osd.28         up 1.00000  1.00000
32 1.09109  osd.32         up 1.00000  1.00000
36 1.09109  osd.36         up 1.00000  1.00000
40 1.09109  osd.40         up 1.00000  1.00000
44 1.09109  osd.44         up 1.00000  1.00000
...
```

OSD の **up** と **in** の合計数を表示するには

```
[root@controller ~]# ceph osd stat
```

出力例

```
osdmap e173: 48 osds: 48 up, 48 in
      flags sortbitwise
```

- 新しいターミナルセッションから Ceph Storage クラスターをモニターします。

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

このターミナルセッションでは、OSD がストレージクラスターから削除される様子を確認できます。次のステップのために、元のターミナルセッションに戻ります。

- OSD を **out** にマークします。

```
ceph osd out $OSD_NUM
```

以下を置き換えます。

- \$OSD_NUM** を OSD 名の番号部分に置き換えます。

例

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

ノード上のすべての OSD を **out** に設定します。



注記

このステップをスクリプト化して複数の OSD を順次処理する場合は、各 **ceph osd out** コマンドの実行の間に少なくとも 10 秒の **sleep** コマンドを設定します。

- すべての配置グループが **アクティブ + クリーン** になり、ストレージクラスターが **HEALTH_OK** 状態になるまで待ちます。ステップ 3 の新しいターミナルセッションから、配置グループの移行を確認できます。このデータのリバランスは、完了するまでに時間がかかる場合があります。
- Ceph Storage クラスターのヘルスステータスを確認します。

```
[root@controller ~]# ceph health
```

- Compute/OSD ノードから、**root** ユーザーとして、すべての OSD デーモンを無効にして停止します。

```
[root@osdcompute ~]# systemctl disable ceph-osd.target  
[root@osdcompute ~]# systemctl stop ceph-osd.target
```

- CRUSH マップから OSD を削除します。

```
ceph osd crush remove osd.$OSD_NUM
```

以下を置き換えます。

- \$OSD_NUM** を OSD 名の番号部分に置き換えます。

例

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



注記

CRUSH マップから OSD を削除すると、CRUSH は配置グループを取得する OSD を再計算し、それに応じてデータをリバランスします。

- OSD 認証キーを削除します。

```
ceph auth del osd.$OSD_NUM
```

以下を置き換えます。

- \$OSD_NUM** を OSD 名の番号部分に置き換えます。

例

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

- OSD を削除します。

```
ceph osd rm $OSD_NUM
```

以下を置き換えます。

- **\$OSD_NUM** を OSD 名の番号部分に置き換えます。

例

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

4.3. オーバークラウドからの NOVA コンピューティングサービスの削除

この手順により、Nova コンピューティングサービスがオーバークラウドのメンバーから削除され、ハードウェアの電源がオフになります。

前提条件

- 実行中のインスタンスをオーバークラウド内の別のコンピュートノードに移行します。

手順

Red Hat OpenStack Platform director (RHOSP-d) ノードで、**stack** ユーザーとして以下の手順を実施します。

1. コンピュートノードのステータスを確認します。

```
[stack@director ~]$ nova service-list
```

2. コンピュートサービスを無効にします。

```
nova service-disable $HOST_NAME nova-compute
```

以下を置き換えます。

- **\$HOST_NAME** をコンピュートのホスト名に置き換えます。

例

```
[stack@director ~]$ nova service-disable overcloud-osd-compute-3.localdomain
nova-compute
+-----+-----+-----+
| Host           | Binary   | Status   |
+-----+-----+-----+
| overcloud-osd-compute-3.localdomain | nova-compute | disabled |
+-----+-----+-----+
```

3. コンピュートノードの Nova ID を収集します。

```
[stack@director ~]$ openstack server list
```

コマンド出力の最初の列にある Nova UUID を書き留めます。

4. OpenStack Platform 名を収集します。

```
[stack@director ~]$ heat stack-list
```

コマンド出力の 2 列目にある **stack_name** を書き留めます。

5. オーバークラウドから UUID でコンピュートノードを削除します。

```
openstack overcloud node delete --stack OSP_NAME NOVA_UUID
```

以下を置き換えます。

- 前のステップの **stack_name** を使用した **OSP_NAME**。
- 前の手順の Nova UUID を使用した **NOVA_UUID**。

例

```
[stack@director ~]$ openstack overcloud node delete --stack overcloud 6b2a2e71-f9c8-4d5b-aaf8-dada97c90821
deleting nodes [u'6b2a2e71-f9c8-4d5b-aaf8-dada97c90821'] from stack overcloud
Started Mistral Workflow. Execution ID: 396f123d-df5b-4f37-b137-83d33969b52b
```

6. コンピュートノードがオーバークラウドから削除されたことを確認します。

```
[stack@director ~]$ openstack server list
```

コンピュートノードが正常に削除された場合、上記のコマンド出力には表示されません。

```
[stack@director ~]$ nova service-list
```

削除された Nova コンピュートノードのステータスは、**disabled** および **down** になります。

7. Ironic がノードの電源をオフにしたことを確認します。

```
[stack@director ~]$ openstack baremetal node list
```

コンピュートノードの電源状態と可用性は、それぞれ **power off** と **available** になります。上記のコマンド出力の最初の列の値である Nova コンピュートサービス ID を書き留めます。

8. Nova スケジューラーの **nova-compute** サービスからコンピュートノードを削除します。

```
nova service-delete COMPUTE_SERVICE_ID
```

以下を置き換えます。

- **COMPUTE_SERVICE_ID** は、前のステップの Nova コンピュートサービス ID に置き換えます。

例

```
[stack@director ~]$ nova service-delete 145
```

4.4. 関連資料

- [Red Hat Ceph Storage 管理 ガイド](#) .

第5章 CEPH ブロックデバイスミラーリングの使用

技術者は、Ceph ブロックデバイスをミラーリングして、ブロックデバイスのデータストレージを保護できます。

5.1. 前提条件

- Red Hat Ceph Storage クラスターが実行されている。
- Ceph クライアントのコマンドラインインターフェイスへのアクセス。

5.2. CEPH ブロックデバイスのミラーリング

Ceph ブロックデバイスのミラーリングは、2つ以上の Ceph クラスター間での Ceph ブロックデバイスイメージの非同期レプリケーションです。

ミラーリングには次の利点があります。* 読み取りと書き込み、ブロックデバイスのサイズ変更、スナップショット、クローン、およびフラット化を含む、イメージに対するすべての変更のポイントインタイムの一貫したレプリカを確保します。* 主に災害時の復旧を目的としています。* アクティブ-パッシブまたはアクティブ-アクティブ設定で実行できます。つまり、必須の排他的ロックとジャーナリング機能を使用して、Ceph はイメージに対するすべての変更を発生順に記録します。* リモートイメージのクラッシュコンシステントミラーがローカルで利用可能であることを確認します。

イメージをピアクラスターにミラーリングする前に、ジャーナリングを有効にする必要があります。



重要

ブロックデバイスイメージをミラーリングするローカルプールとリモートプールをサポートする CRUSH 階層は、同じ容量とパフォーマンス特性を持つべきであり、過剰な遅延なしでミラーリングを保証するために十分な帯域幅を持つべきです。たとえば、プライマリークラスター内のイメージへの平均書き込みスループットが X MiB/s である場合に、ネットワークはセカンダリーサイトへのネットワーク接続で N * X スループットと、N イメージをミラーリングする安全係数 Y% に対応している必要があります。

rbd-mirror デーモン

rbd-mirror デーモンは、ある Ceph クラスターから別のクラスターへのイメージの同期を担当します。**rbd-mirror** パッケージは **rbd-mirror** デーモンを提供します。レプリケーションのタイプに応じて、**rbd-mirror** は単一のクラスターまたはミラーリングに参加するすべてのクラスターで実行されます。

一方向レプリケーション

データがプライマリークラスターからバックアップとして機能するセカンダリークラスターにミラーリングされる場合、**rbd-mirror** はバックアップクラスターでのみ実行されます。RBD ミラーリングは、アクティブ/パッシブ設定で複数のセカンダリーサイトを持つことができます。

双方向レプリケーション

データがプライマリークラスターからセカンダリークラスターにミラーリングされ、セカンダリークラスターがプライマリーおよび相互にミラーリングできる場合、両方のクラスターで **rbd-mirror** が実行されている必要があります。現在、アクティブ/アクティブ設定とも呼ばれる双方向レプリケーションは、2つのサイト間でのみサポートされています。



重要

双方向レプリケーションでは、**rbd-mirror** の各インスタンスが同時に他の Ceph クラスタに接続できる必要があります。また、ミラーリングを処理するために、ネットワークには2つのデータセンターサイトの間で十分な帯域幅が必要です。



警告

Ceph Storage クラスタごとに単一の **rbd-mirror** デーモンのみを実行します。

ミラーリングのモード

ミラーリングは、ピアクラスター内のプールごとに設定されます。Red Hat Ceph Storage は、プール内のどのイメージがミラーリングされるかに応じて、2つのモードをサポートします。

プールモード

ジャーナリング機能を有効にして、プール内のすべてのイメージをミラーリングします。

イメージモード

プール内のイメージの特定のサブセットのみがミラーリングされるため、イメージごとにミラーリングを個別に有効にする必要があります。

イメージの状態

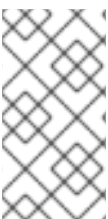
アクティブ/パッシブ設定では、ミラーリングされたイメージは次のとおりです。

- プライマリー
 - これらのミラーリングされたイメージは変更 **できます**。
- 非プライマリー
 - これらのミラーイメージは変更 **できません**。

イメージでミラーリングが最初に有効化された時点で、イメージはプライマリーに自動的にプロモートされます。イメージプロモーションは、ミラーリングモードに基づいて暗黙的または明示的に行うことができます。プールモードでミラーリングが有効になっている場合、イメージプロモーションは暗黙的に行われます。ミラーリングがイメージモードで有効になっている場合、イメージプロモーションは明示的に行われます。プライマリーイメージをデモートし、プライマリー以外のイメージをプロモートすることもできます。

非同期 Red Hat Ceph Storage 更新

Ceph Block Device ミラーリングを使用してストレージクラスターに非同期更新を行う場合は、更新のインストール手順に従ってください。更新が正常に完了したら、Ceph Block Device インスタンスを再起動します。



注記

ceph-rbd-mirror インスタンスを再起動するために必要な順序はありません。Red Hat では、プライマリーイメージを含むプールを指す **ceph-rbd-mirror** インスタンスを再起動し、続いてミラー化されたプールを指す **ceph-rbd-mirror** インスタンスを再起動することをお勧めします。

関連情報

- 詳細は、Red Hat Ceph Storage Block Device Guide の [ジャーナリングの有効化](#) セクションを参照してください。
- 詳細は、Red Hat Ceph Storage Block Device Guide の [災害からの回復](#) セクションを参照してください。
- 詳細は、Red Hat Ceph Storage Block Device Guide の [プールミラーリングの有効化](#) セクションを参照してください。
- 詳細は、Red Hat Ceph Storage ブロックデバイスガイドの [イメージでのミラーリングの有効化](#) セクションを参照してください。

5.3. 同じ名前のストレージクラスター間のミラーリングの設定

同じクラスター名 (デフォルトではストレージクラスター名は **ceph**) を使用して Ceph Storage クラスターを作成すると、Ceph ブロックデバイスのミラーリングで問題が発生する可能性があります。たとえば、一部の Ceph 関数は、**ceph** という名前のストレージクラスターを想定しています。両方のクラスターが同じ名前の場合、現在、追加の手順を実行して **rbd-mirror** を設定する必要があります。

前提条件

- 異なるサイトにある 2 つの実行中の Red Hat Ceph Storage クラスター。
- **rbd-mirror** デーモンが実行されるストレージクラスターまたはクライアントノードへのアクセス。

手順

1. **root** として、両方のストレージクラスターで、適切なファイルに **CLUSTER** オプションを追加して、ストレージクラスター名を指定します。

例

```
CLUSTER=master
```

Red Hat Enterprise Linux

`/etc/sysconfig/ceph` ファイルを編集し、値として Ceph Storage クラスター名を指定して **CLUSTER** オプションを追加します。

Ubuntu

`/etc/default/ceph` ファイルを編集し、値として Ceph Storage クラスター名を指定して **CLUSTER** オプションを追加します。

2. **root** として、**rbd-mirror** デーモンを実行しているノードに対してのみ、**ceph.conf** ファイルへのシンボリックリンクを作成します。

```
[root@monitor ~]# ln -s /etc/ceph/ceph.conf /etc/ceph/master.conf
```

3. ストレージクラスターを参照する場合は、シンボリックリンク名と **--cluster** フラグを使用します。

例

```
--cluster master
```

5.4. ミラーリングのための CEPH BLOCK DEVICE JOURNALING の有効化

Ceph Block Device ジャーナリング機能を有効にする方法は2つあります。

- イメージ作成について。
- 既存のイメージに対して動的に。



重要

ジャーナリングは、有効にする必要がある **exclusive-lock** 機能に依存します。

前提条件

- Red Hat Ceph Storage クラスタが実行されている。
- Ceph クライアントコマンドラインインターフェイスへのアクセス。

手順

イメージ作成時に有効にする

1. 通常のコマンドラインとして、次のコマンドを実行して、イメージの作成時にジャーナリングを有効にします。

```
rd create $IMAGE_NAME --size $MEGABYTES --pool $POOL_NAME --image-feature
$FEATURE_NAME[$FEATURE_NAME]
```

例

```
[user@rd-client ~]$ rbd create image-1 --size 1024 --pool pool-1 --image-feature exclusive-
lock,journaling
```

既存のイメージで有効にする

1. 通常のコマンドラインとして、次のコマンドを実行して、既存のイメージのジャーナリングを有効にします。

```
rbd feature enable $POOL_NAME/$IMAGE_NAME $FEATURE_NAME
```

例

```
[user@rd-client ~]$ rbd feature enable pool-1/image-1 exclusive-lock
[user@rd-client ~]$ rbd feature enable pool-1/image-1 journaling
```

デフォルトの設定

1. デフォルトですべての新しいイメージのジャーナリングを有効にするには、次の行を Ceph 設定ファイル `/etc/ceph/ceph.conf` にデフォルトで追加します。

■

```
rbd default features = 125
```

関連情報

- 詳細は、Red Hat Ceph Storage インストールガイドの **Ceph クライアントロール** のインストールセクションを参照してください。
 - [Red Hat Enterprise Linux](#)
 - [Ubuntu](#)

5.5. プールでの CEPH ブロックデバイスミラーリングの設定

技術者は、プールのミラーリングを有効または無効にしたり、クラスターピアを追加または削除したり、ピアとプールに関する情報を表示したりできます。

- [「プールでのミラーリングの有効化」](#)
- [「プールでのミラーリングの無効化」](#)
- [「クラスターピアの追加」](#)
- [「クラスターピアの削除」](#)
- [「クラスターピアに関する情報の表示」](#)
- [「プールのミラーリングステータスの取得」](#)

5.5.1. 前提条件

- Red Hat Ceph Storage クラスターが実行されている。
- Ceph クライアントコマンドラインインターフェイスへのアクセス。

5.5.2. プールでのミラーリングの有効化

オブジェクトプールでミラーリングを有効にする場合、使用するミラーリングモードを指定する必要があります。

前提条件

- Red Hat Ceph Storage クラスターが実行されている。
- Ceph クライアントコマンドラインインターフェイスへのアクセス。
- 既存のオブジェクトプール。

手順

1. 通常のコマンドラインとして、次のコマンドを実行して、プールでミラーリングを有効にします。

```
rbd mirror pool enable $POOL_NAME $MODE
```

例

プールモードを有効にするには:


```
[user@rbd-client ~]$ rbd mirror pool enable data pool
```

イメージモードを有効にするには:

```
[user@rbd-client ~]$ rbd mirror pool enable data image
```

関連情報

- 詳細は、Red Hat Ceph Storage インストールガイドの **Ceph クライアントロール** のインストールセクションを参照してください。
 - [Red Hat Enterprise Linux](#)
 - [Ubuntu](#)
- 詳細は、「**ミラーリングのモード**」を参照してください。

5.5.3. プールでのミラーリングの無効化

プールでミラーリングを無効にする前に、クラスターピアを削除する必要があります。



注記

プールでミラーリングを無効にすると、イメージモードで個別にミラーリングが有効にされたプール内のすべてのイメージのミラーリングも無効になります。

前提条件

- Red Hat Ceph Storage クラスターが実行されている。
- Ceph クライアントコマンドラインインターフェイスへのアクセス。
- クラスターピアとして削除されました。
- 既存のオブジェクトプール。

手順

1. 通常のコマンドラインとして、次のコマンドを実行してプールのミラーリングを無効にします。

```
rbd mirror pool disable $POOL_NAME
```

例

```
[user@rbd-client ~]$ rbd mirror pool disable data
```

関連情報

- 詳細は、Red Hat Ceph Storage インストールガイドの **Ceph クライアントロール** のインストールセクションを参照してください。
 - [Red Hat Enterprise Linux](#)
 - [Ubuntu](#)

- 詳細は、「[クラスターピアの削除](#)」を参照してください。

5.5.4. クラスターピアの追加

rbd-mirror デーモンがそのピアクラスターを検出するには、ピアをプールに登録する必要があります。

前提条件

- 異なるサイトにある 2 つの実行中の Red Hat Ceph Storage クラスター。
- Ceph クライアントコマンドラインインターフェイスへのアクセス。
- 既存のオブジェクトプール。

手順

1. 通常のコマンドラインとして、次のコマンドを実行してクラスターピアを追加します。

```
rbd --cluster $CLUSTER_NAME mirror pool peer add $POOL_NAME
$CLIENT_NAME@$TARGET_CLUSTER_NAME
```

例

remote クラスターをピアとして **local** クラスターに追加します。

```
[user@rbd-client ~]$ rbd --cluster local mirror pool peer add data client.remote@remote
```

関連情報

- 詳細は、Red Hat Ceph Storage インストールガイドの **Ceph クライアントロール** のインストールセクションを参照してください。
 - [Red Hat Enterprise Linux](#)
 - [Ubuntu](#)

5.5.5. クラスターピアの削除

rbd-mirror デーモンがそのピアクラスターを検出するには、ピアをプールに登録する必要があります。

前提条件

- 異なるサイトにある 2 つの実行中の Red Hat Ceph Storage クラスター。
- Ceph クライアントコマンドラインインターフェイスへのアクセス。
- 既存のクラスターピア。

手順

1. 次の手順で使用するために、ピアの Universally Unique Identifier (UUID) を記録します。ピアの UUID を表示するには、通常のコマンドラインとして次のコマンドを実行します。

```
rbd mirror pool info $POOL_NAME
```

2. 通常のユーザーとして、次のコマンドを実行してクラスターピアを削除します。

```
rbd mirror pool peer remove $POOL_NAME $PEER_UUID
```

例

```
[user@rbd-client ~]$ rbd mirror pool peer remove data 55672766-c02b-4729-8567-f13a66893445
```

関連情報

- 詳細は、Red Hat Ceph Storage インストールガイドの **Ceph クライアントロール** のインストールセクションを参照してください。
 - [Red Hat Enterprise Linux](#)
 - [Ubuntu](#)

5.5.6. クラスターピアに関する情報の表示

この手順を実行すると、クラスターピアに関する基本情報を表示できます。

前提条件

- 異なるサイトにある 2 つの実行中の Red Hat Ceph Storage クラスター。
- Ceph クライアントコマンドラインインターフェイスへのアクセス。
- 既存のクラスターピア。

手順

1. 通常ユーザーとして、次のコマンドを実行して、クラスターピアに関する情報を表示します。

```
rbd mirror pool info $POOL_NAME
```

例

```
[user@rbd-client ~]$ rbd mirror pool info data
Enabled: true
Peers:
  UUID                NAME      CLIENT
  786b42ea-97eb-4b16-95f4-867f02b67289 ceph-remote client.admin
```

関連情報

- 詳細は、Red Hat Ceph Storage インストールガイドの **Ceph クライアントロール** のインストールセクションを参照してください。
 - [Red Hat Enterprise Linux](#)
 - [Ubuntu](#)

5.5.7. プールのミラーリングステータスの取得

この手順を実行すると、プールの Ceph Block Device ミラーリングのステータスを表示できます。

前提条件

- Ceph クライアントコマンドラインインターフェイスへのアクセス。
- 既存のクラスターピア。
- 既存のオブジェクトストレージプール。

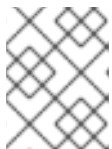
手順

1. 通常のコマンドラインユーザーとして、次のコマンドを実行して、プールのミラーリングステータスを表示します。

```
rbd mirror pool status <pool-name>
```

例

```
[user@rbd-client ~]$ rbd mirror pool status data
health: OK
images: 1 total
```



注記

プール内のすべてのミラーリングされたイメージの詳細を出力するには、**--verbose** オプションを使用します。

関連情報

- 詳細は、Red Hat Ceph Storage インストールガイドの **Ceph クライアントロール** のインストールセクションを参照してください。
 - [Red Hat Enterprise Linux](#)
 - [Ubuntu](#)

5.5.8. 関連情報

- 詳細は、Red Hat Ceph Storage インストールガイドの **Ceph クライアントロール** のインストールセクションを参照してください。
 - [Red Hat Enterprise Linux](#)
 - [Ubuntu](#)

5.6. イメージでの CEPH BLOCK デバイスマラーリングの設定

技術者は、イメージのミラーリングを有効または無効にしたり、クラスターピアを追加または削除したり、ピアとプールに関する情報を表示したりできます。

5.6.1. 前提条件

- Red Hat Ceph Storage クラスターが実行されている。

- Ceph クライアントコマンドラインインターフェイスへのアクセス。

5.7. イメージミラーリングの有効化

この手順により、イメージでの Ceph Block Device ミラーリングが有効になります。

前提条件

- Red Hat Ceph Storage クラスターが実行されている。
- Ceph クライアントコマンドラインインターフェイスへのアクセス。
- 既存のイメージ。

手順

1. 通常のコマンドラインとして、次のコマンドを実行してイメージのミラーリングを有効にします。

```
rbd mirror image enable $POOL_NAME/$IMAGE_NAME
```

例

```
[user@rbd-client ~]$ rbd mirror image enable data/image2
```

関連情報

- 詳細は、Red Hat Ceph Storage インストールガイドの **Ceph クライアントロール** のインストールセクションを参照してください。
 - [Red Hat Enterprise Linux](#)
 - [Ubuntu](#)
- 詳細は、「[ミラーリングのモード](#)」を参照してください。

included::modules/procedure_rbd_disabling-image-mirroring_en-us.adoc[leveloffset=+1]

included::modules/procedure_rbd_promoting-and-demoting-an-image_en-us.adoc[leveloffset=+1]

included::modules/procedure_rbd_resynchronizing-an-image_en-us.adoc[leveloffset=+1]

included::modules/procedure_rbd_viewing-the-mirroring-status-for-a-single-image_en-us.adoc[leveloffset=+1]

5.7.1. 関連情報

- 詳細は、Red Hat Ceph Storage インストールガイドの **Ceph クライアントロール** のインストールセクションを参照してください。
 - [Red Hat Enterprise Linux](#)
 - [Ubuntu](#)

5.8. CEPH BLOCK デバイスの双方向ミラーリングの設定

双方向ミラーリングは、自動フェイルオーバーに適した効果的なアクティブ/アクティブミラーリングソリューションです。

前提条件

- 異なるサイトにある2つの実行中の Red Hat Ceph Storage クラスタ。
 - 各ストレージクラスタには、対応する設定ファイルが `/etc/ceph/` ディレクトリーにあります。
- 両方のストレージクラスタに接続する1つの Ceph クライアント。
 - Ceph クライアントのコマンドラインインターフェイスへのアクセス。
- 既存のオブジェクトストレージプールとイメージ。
 - 各ストレージクラスタには同じオブジェクトプール名が存在します。

手順

- オブジェクトストレージプール内のすべてのイメージで **exclusive-lock** と **journaling** が有効になっていることを確認します。

```
rbd info $POOL_NAME/$IMAGE_NAME
```

例

```
[user@rbd-client ~]$ rbd info data/image1
```

- rbd-mirror** パッケージは、Red Hat Ceph Storage Tools リポジトリによって提供されます。**root** として、ローカルおよびリモートのストレージクラスタの Ceph Monitor ノードで、**rbd-mirror** パッケージをインストールします。

Red Hat Enterprise Linux

```
[root@monitor-remote ~]# yum install rbd-mirror
```

Ubuntu

```
[user@monitor-remote ~]$ sudo apt-get install rbd-mirror
```



注記

rbd-mirror デーモンは、ストレージクラスタ内の任意のノードで実行できます。Ceph Monitor または OSD ノードである必要はありません。ただし、ストレージクラスタごとに1つの **rbd-mirror** デーモンのみです。

- root** として、両方のストレージクラスタで、適切なファイルに **CLUSTER** オプションを追加して、ストレージクラスタ名を指定します。

例

```
CLUSTER=local
```

Red Hat Enterprise Linux

`/etc/sysconfig/ceph` ファイルを編集し、値として Ceph Storage クラスター名を指定して **CLUSTER** オプションを追加します。

Ubuntu

`/etc/default/ceph` ファイルを編集し、値として Ceph Storage クラスター名を指定して **CLUSTER** オプションを追加します。



注記

同じ名前の2つの Ceph Storage クラスター間で Ceph Block Device ミラーリングを処理する [手順](#) を参照してください。

4. 通常のユーザーとして、両方のストレージクラスターで、オブジェクトストレージプールにアクセスし、キーリングをファイルに出力する権限を持つユーザーを作成します。

```
ceph auth get-or-create client.$STORAGE_CLUSTER_NAME mon 'profile rbd' osd 'profile rbd pool=$POOL_NAME' -o $PATH_TO_KEYRING_FILE --cluster $STORAGE_CLUSTER_NAME
```

- a. ローカルストレージクラスターの Ceph Monitor ノードで、**client.local** ユーザーを作成し、キーリングを **local.client.local.keyring** ファイルに出力します。

例

```
[user@monitor-local ~]$ ceph auth get-or-create client.local mon 'profile rbd' osd 'profile rbd pool=data' -o /etc/ceph/local.client.local.keyring --cluster local
```

- b. リモートストレージクラスターの Ceph Monitor ノードで、**client.remote** ユーザーを作成し、キーリングを **remote.client.remote.keyring** ファイルに出力します。

例

```
[user@monitor-remote ~]$ ceph auth get-or-create client.remote mon 'profile rbd' osd 'profile rbd pool=data' -o /etc/ceph/remote.client.remote.keyring --cluster remote
```

5. **root** として、Ceph 設定ファイルとストレージクラスターごとに新しく作成されたキーリングファイルを、各ストレージクラスター間でコピーし、両方のストレージクラスター内の任意の Ceph クライアントノードにコピーします。

```
scp $PATH_TO_STORAGE_CLUSTER_CONFIG_FILE_NAME $SSH_USER_NAME@$MON_NODE:/etc/ceph/
scp $PATH_TO_STORAGE_CLUSTER_KEYRING_FILE_NAME $SSH_USER_NAME@$CLIENT_NODE:/etc/ceph/
```

ローカルからリモートへのコピーの例

```
[root@monitor-local ~]# scp /etc/ceph/local.conf example@remote:/etc/ceph/
[root@monitor-local ~]# scp /etc/ceph/local.client.local.keyring example@remote:/etc/ceph/
```

リモートからローカルへのコピーの例

```
[root@monitor-remote ~]# scp /etc/ceph/remote.conf example@local:/etc/ceph/
[root@monitor-remote ~]# scp /etc/ceph/remote.client.remote.keyring
example@local:/etc/ceph/
```

ローカルとリモートの両方をクライアントにコピーする例

```
[root@monitor-local ~]# scp /etc/ceph/local.conf example@rbd-client:/etc/ceph/
[root@monitor-local ~]# scp /etc/ceph/local.client.local.keyring example@rbd-client:/etc/ceph/
```

```
[root@monitor-remote ~]# scp /etc/ceph/remote.conf example@rbd-client:/etc/ceph/
[root@monitor-remote ~]# scp /etc/ceph/remote.client.remote.keyring example@rbd-
client:/etc/ceph/
```

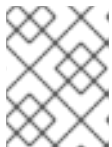
6. **root** として、両方のストレージクラスターの Ceph Monitor ノードで、**rbd-mirror** デーモンを有効にして起動します。

```
systemctl enable ceph-rbd-mirror.target
systemctl enable ceph-rbd-mirror@$CLIENT_ID
systemctl start ceph-rbd-mirror@$CLIENT_ID
```

\$CLIENT_ID は、**rbd-mirror** デーモンが使用する Ceph Storage クラスターユーザーです。

例:

```
[root@monitor-remote ~]# systemctl enable ceph-rbd-mirror.target
[root@monitor-remote ~]# systemctl enable ceph-rbd-mirror@remote
[root@monitor-remote ~]# systemctl start ceph-rbd-mirror@remote
```



注記

\$CLIENT_ID ユーザーには、ストレージクラスターへの適切な **cephx** 認証アクセスが必要です。

プールモードの双方向ミラーリングの設定

1. 通常のコピーモードとして、各ストレージクラスターにアクセスできる任意の Ceph クライアントノードから、両方のストレージクラスターに存在するオブジェクトストレージプールのプールミラーリングを有効にします。

```
rbd mirror pool enable $POOL_NAME $MIRROR_MODE --cluster
$STORAGE_CLUSTER_NAME
```

例

```
[user@rbd-client ~]$ rbd mirror pool enable data pool --cluster local
[user@rbd-client ~]$ rbd mirror pool enable data pool --cluster remote
```

- a. ミラーリングが正常に有効になっていることを確認します。

```
rbd mirror pool status $POOL_NAME
```


例

```
[user@rbd-client ~]$ rbd mirror pool status data
health: OK
images: 1 total
```

2. 通常のユーザーとして、ストレージクラスターを他のストレージクラスターのピアとして追加します。

```
rbd mirror pool peer add $POOL_NAME $CLIENT_NAME@$STORAGE_CLUSTER_NAME
--cluster $PEER_STORAGE_CLUSTER_NAME
```

例

```
[user@rbd-client ~]$ rbd mirror pool peer add data client.local@local --cluster remote
[user@rbd-client ~]$ rbd mirror pool peer add data client.remote@remote --cluster local
```

- a. ストレージクラスターピアが正常に追加されたことを確認します。

例

```
[user@rbd-client ~]$ rbd mirror pool info data --cluster local
Mode: pool
Peers:
  UUID                NAME CLIENT
  de32f0e3-1319-49d3-87f9-1fc076c83946 remote client.remote

[user@rbd-client ~]$ rbd mirror pool info data --cluster remote
Mode: pool
Peers:
  UUID                NAME CLIENT
  87ea0826-8ffe-48fb-b2e8-9ca81b012771 local client.local
```

イメージモードの一方方向ミラーリングの設定

1. 通常ユーザーとして、両方のストレージクラスターでオブジェクトストレージプールのイメージミラーリングを有効にします。

```
rbd mirror pool enable $POOL_NAME $MIRROR_MODE --cluster
$STORAGE_CLUSTER_NAME
```

例

```
[user@rbd-client ~]$ rbd mirror pool enable data image --cluster local
[user@rbd-client ~]$ rbd mirror pool enable data image --cluster remote
```

- a. ミラーリングが正常に有効になっていることを確認します。

```
rbd mirror pool status $POOL_NAME
```

例

```
[user@rbd-client ~]$ rbd mirror pool status data
health: OK
images: 1 total
```

2. 通常のユーザーとして、ストレージクラスターを他のストレージクラスターのピアとして追加します。

```
rbd mirror pool peer add $POOL_NAME $CLIENT_NAME@$STORAGE_CLUSTER_NAME
--cluster $PEER_STORAGE_CLUSTER_NAME
```

例

```
[user@rbd-client ~]$ rbd mirror pool peer add data client.local@local --cluster remote
[user@rbd-client ~]$ rbd mirror pool peer add data client.remote@remote --cluster local
```

- a. ストレージクラスターピアが正常に追加されたことを確認します。

```
rbd mirror pool info --cluster $STORAGE_CLUSTER_NAME
```

例

```
[user@rbd-client ~]$ rbd mirror pool info --cluster remote
Mode: image
Peers:
  UUID                NAME  CLIENT
  87ea0826-8ffe-48fb-b2e8-9ca81b012771 local client.local
```

```
[user@rbd-client ~]$ rbd mirror pool info --cluster local
Mode: image
Peers:
  UUID                NAME  CLIENT
  de32f0e3-1319-49d3-87f9-1fc076c83946 remote client.remote
```

3. 通常ユーザーとして、ローカルストレージクラスターで、イメージのミラーリングを明示的に有効にします。

```
rbd mirror image enable $POOL_NAME/$IMAGE_NAME --cluster
$STORAGE_CLUSTER_NAME
```

例

```
[user@rbd-client ~]$ rbd mirror image enable data/image1 --cluster local
Mirroring enabled
```

- a. ミラーリングが正常に有効になっていることを確認します。

```
[user@rbd-client ~]$ rbd mirror image status data/image1 --cluster local
image1:
  global_id: 2c928338-4a86-458b-9381-e68158da8970
  state:    up+replaying
  description: replaying, master_position=[object_number=6, tag_tid=2,
  entry_tid=22598], mirror_position=[object_number=6, tag_tid=2,
```

```
entry_tid=29598], entries_behind_master=0
last_update: 2018-04-28 18:47:39
```

```
[user@rbd-client ~]$ rbd mirror image status data/image1 --cluster remote
image1:
  global_id: 2c928338-4a86-458b-9381-e68158da8970
  state:    up+replaying
  description: replaying, master_position=[object_number=6, tag_tid=2,
entry_tid=22598], mirror_position=[object_number=6, tag_tid=2,
entry_tid=29598], entries_behind_master=0
last_update: 2018-04-28 18:47:39
```

関連情報

- 詳細は、Red Hat Ceph Storage インストールガイドの **Ceph クライアントロール** のインストールセクションを参照してください。
 - [Red Hat Enterprise Linux](#)
 - [Ubuntu](#)
- 詳細については、**ストレージ戦略** ガイドの **プール** の章を参照してください。
- 詳細については、**管理ガイド** の **ユーザー管理** の章を参照してください。
- 詳細は、「**プールでの Ceph ブロックデバイスミラーリングの設定**」を参照してください。
- 詳細は、「**イメージでの Ceph Block デバイスミラーリングの設定**」を参照してください。
- 詳細は、「**ミラーリングのための Ceph Block Device Journaling の有効化**」を参照してください。

5.9. ストレージクラスター間のレプリケーションの遅延

一方向または双方向のレプリケーションを使用しているかどうかにかかわらず、Ceph Block Device ミラーリングイメージ間のレプリケーションを遅らせることができます。プライマリーイメージへの不要な変更がレプリケートされたセカンダリーイメージに伝達されるまでの時間の緩衝として、レプリケーションの遅延ストラテジーを実装できます。レプリケーションの遅延は、グローバルまたは個々のイメージで設定でき、宛先ストレージクラスターで設定する必要があります。

前提条件

- 異なるサイトにある 2 つの実行中の Red Hat Ceph Storage クラスター。
- **rbd-mirror** デーモンが実行されるストレージクラスターまたはクライアントノードへのアクセス。

手順

レプリケーションの遅延をグローバルに設定する

1. **root** として、**rbd-mirror** デーモンを実行しているノードで Ceph 設定ファイルを編集し、次の行を追加します。

```
rbd_mirroring_replay_delay = $MINIMUM_DELAY_IN_SECONDS
```

例

```
rbd_mirroring_replay_delay = 600
```

イメージのレプリケーション遅延の設定

1. 通常のユーザーとして、Ceph クライアントノードで、特定のプライマリーイメージのレプリケーション遅延を設定し、次のコマンドを実行します。

```
rbd image-meta set $POOL_NAME/$IMAGE_NAME conf_rbd_mirroring_replay_delay
$MINIMUM_DELAY_IN_SECONDS
```

+ .例

```
[user@rbd-client ~]$ rbd image-meta set data/image1 conf_rbd_mirroring_replay_delay 600
```

関連情報

- 詳細は、「[Ceph ブロックデバイスのミラーリング](#)」を参照してください。

5.10. 災害からの復旧

次の手順は、プライマリーストレージクラスターが正常または非正常な方法で終了した後で、セカンダリーストレージクラスター上のミラー化されたデータにフェイルオーバーする方法を示しています。

前提条件

- 異なるサイトにある 2 つの実行中の Red Hat Ceph Storage クラスター。
- 両方のストレージクラスターに接続する 1 つの Ceph クライアント。
 - Ceph クライアントのコマンドラインインターフェイスへのアクセス。

手順

正常なシャットダウン後のフェイルオーバー

1. プライマリーイメージを使用するクライアントをすべて停止します。この手順は、イメージを使用しているクライアントによって異なります。
2. 通常のユーザーとして、Ceph クライアントノードで、ローカルストレージクラスターにあるプライマリーイメージをデモートします。

```
rbd mirror image demote $POOL_NAME/$IMAGE_NAME --
cluster=$STORAGE_CLUSTER_NAME
```

例

```
[user@rbd-client ~]$ rbd mirror image demote data/image1 --cluster=local
```

3. 通常のユーザーとして、Ceph クライアントノードで、リモートストレージクラスターにある非プライマリーイメージをプロモートします。

```
rbd mirror image promote $POOL_NAME/$IMAGE_NAME --
cluster=$STORAGE_CLUSTER_NAME
```

例

```
[user@rbd-client ~]$ rbd mirror image promote data/image1 --cluster=remote
```

4. ピアイメージへのアクセスを再開します。この手順は、イメージを使用しているクライアントによって異なります。

正常にシャットダウンされなかった場合のフェイルオーバー

1. プライマリストレージクラスターが停止していることを確認します。
2. プライマリイメージを使用するクライアントをすべて停止します。この手順は、イメージを使用しているクライアントによって異なります。
3. 通常のコピーとして、Ceph クライアントノードで、リモートストレージクラスターにある非プライマリイメージをプロモートします。ローカルストレージクラスターにデモットが伝播されないため、**--force** オプションを使用します。

```
rbd mirror image promote --force $POOL_NAME/$IMAGE_NAME --
cluster=$STORAGE_CLUSTER_NAME
```

例

```
$ rbd mirror image promote --force data/image1 --cluster=remote
```

4. ピアイメージへのアクセスを再開します。この手順は、イメージを使用しているクライアントによって異なります。

プライマリストレージクラスターへのフェイルバック

1. プライマリストレージクラスターが使用可能であることを確認します。
2. 通常のコピーとして、Ceph クライアントノードで正常にシャットダウンされなかった場合は、ローカルストレージクラスターにあるプライマリイメージを降格します。

```
rbd mirror image demote $POOL_NAME/$IMAGE_NAME --
cluster=$STORAGE_CLUSTER_NAME
```

例

```
[user@rbd-client ~]$ rbd mirror image demote data/image1 --cluster=local
```

3. 正常にシャットダウンされなかった場合にのみ、イメージをもう一度同期します。通常のコピーとして、Ceph クライアントノードでイメージを再同期します。

```
rbd mirror image resync $POOL_NAME/$IMAGE_NAME --
cluster=$STORAGE_CLUSTER_NAME
```

例

```
[user@rbd-client ~]$ rbd mirror image resync data/image1 --cluster=local
```

- 再同期が完了し、**up+replaying** 状態になっていることを確認します。通常のコマンドとして、Ceph クライアントノードで、イメージの再同期ステータスを確認します。

```
rbd mirror image status $POOL_NAME/$IMAGE_NAME --
cluster=$STORAGE_CLUSTER_NAME
```

例

```
[user@rbd-client ~]$ rbd mirror image status data/image1 --cluster=local
```

- 通常のコマンドとして、Ceph クライアントノードで、リモートストレージクラスターにあるセカンダリーイメージを降格します。

```
rbd mirror image demote $POOL_NAME/$IMAGE_NAME --
cluster=$STORAGE_CLUSTER_NAME
```

例

```
[user@rbd-client ~]$ rbd mirror image demote data/image1 --cluster=remote
```

- 通常のコマンドとして、Ceph クライアントノードで、ローカルストレージクラスターにある以前のプライマリーイメージをプロモートします。

```
rbd mirror image promote $POOL_NAME/$IMAGE_NAME --
cluster=$STORAGE_CLUSTER_NAME
```

例

```
$ rbd mirror image promote data/image1 --cluster=local
```

関連情報

- Red Hat OpenStack Platform ストレージガイドの [ブロックストレージおよびボリューム](#) の章を参照してください。

5.11. 関連情報

- 詳細は、Red Hat Ceph Storage インストールガイドの [Ceph クライアントロール](#) のインストールセクションを参照してください。
 - [Red Hat Enterprise Linux](#)
 - [Ubuntu](#)