



Red Hat Ceph Storage 5

オペレーションガイド

Red Hat Ceph Storage の操作タスク

Red Hat Ceph Storage 5 オペレーションガイド

Red Hat Ceph Storage の操作タスク

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 © 2022 | 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

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

目次

第1章 CEPH ORCHESTRATOR の紹介	5
1.1. CEPH ORCHESTRATOR の使用	5
第2章 CEPH ORCHESTRATOR を使用したサービスの管理	7
2.1. CEPH ORCHESTRATOR の配置指定	7
2.2. コマンドラインインターフェースを使用した CEPH デーモンのデプロイ	7
2.3. コマンドラインインターフェースを使用したホストのサブセットへの CEPH デーモンのデプロイ	9
2.4. CEPH ORCHESTRATOR のサービス指定	10
2.5. サービス指定を使用した CEPH デーモンのデプロイ	11
第3章 CEPH ORCHESTRATOR を使用したホストの管理	14
3.1. 前提条件	14
3.2. CEPH ORCHESTRATOR を使用したホストの追加	14
3.3. CEPH ORCHESTRATOR を使用した複数ホストの追加	16
3.4. CEPH ORCHESTRATOR を使用したホストの一覧表示	18
3.5. CEPH ORCHESTRATOR を使用したホストへのラベルの追加	18
3.6. CEPH ORCHESTRATOR を使用したホストの削除	20
3.7. CEPH ORCHESTRATOR を使用したホストのメンテナンスモード	22
第4章 CEPH ORCHESTRATOR を使用したモニターの管理	24
4.1. CEPH MONITOR	24
4.2. モニタリング選択ストラテジーの設定	25
4.3. コマンドラインインターフェースを使用した CEPH モニターデーモンのデプロイ	25
4.4. サービス指定を使用した CEPH モニターデーモンのデプロイ	28
4.5. CEPH ORCHESTRATOR を使用した特定ネットワークでのモニターデーモンのデプロイ	29
4.6. CEPH ORCHESTRATOR を使用したモニターデーモンの削除	31
4.7. 異常なストレージクラスターからの CEPH MONITOR の削除	32
第5章 CEPH ORCHESTRATOR を使用したマネージャーの管理	34
5.1. 前提条件	34
5.2. CEPH ORCHESTRATOR を使用したマネージャーデーモンのデプロイ	34
5.3. CEPH ORCHESTRATOR を使用したマネージャーデーモンの削除	35
5.4. CEPH MANAGER バランサーモジュールの使用	36
5.5. CEPHMANAGER アラートモジュールの使用	40
5.6. CEPH MANAGER クラッシュモジュールの使用	42
第6章 CEPH ORCHESTRATOR を使用した OSD の管理	47
6.1. CEPH OSD	47
6.2. CEPH OSD ノードの設定	47
6.3. OSD メモリーの自動チューニング	47
6.4. CEPH OSD デプロイメント用のデバイスの一覧表示	49
6.5. CEPH OSD デプロイメントのデバイスの消去	51
6.6. すべての利用可能なデバイスへの CEPH OSD のデプロイ	52
6.7. 特定のデバイスおよびホストへの CEPH OSD のデプロイ	54
6.8. OSD をデプロイするための高度なサービス指定およびフィルター	55
6.9. 高度なサービス指定を使用した CEPH OSD のデプロイ	58
6.10. CEPH ORCHESTRATOR を使用した OSD デーモンの削除	62
6.11. CEPH ORCHESTRATOR を使用した OSD の置き換え	63
6.12. CEPH ORCHESTRATOR を使用した OSD の削除停止	65
6.13. CEPH ORCHESTRATOR を使用した OSD のアクティブ化	66
6.13.1. データ移行の監視	67
6.14. 配置グループの再計算	67

第7章 CEPH ORCHESTRATOR を使用したモニタリングスタックの管理	69
7.1. CEPH ORCHESTRATOR を使用したモニタリングスタックのデプロイ	70
7.2. CEPH ORCHESTRATOR を使用したモニタリングスタックの削除	72
第8章 基本的な RED HAT CEPH STORAGE のクライアント設定	74
8.1. クライアントマシンでのファイルの設定	74
8.2. クライアントマシンでのキーリングの設定	74
第9章 CEPH ORCHESTRATOR を使用した MDS サービスの管理	76
9.1. 前提条件	76
9.2. コマンドラインインターフェースを使用した MDS サービスのデプロイ	76
9.3. サービス指定を使用した MDS サービスのデプロイ	78
9.4. CEPH ORCHESTRATOR を使用した MDS サービスの削除	80
第10章 CEPH ORCHESTRATOR を使用した CEPH オブジェクトゲートウェイの管理	83
10.1. 前提条件	83
10.2. コマンドラインインターフェースを使用した CEPH OBJECT GATEWAY のデプロイ	83
10.3. サービス仕様を使用した CEPH OBJECT GATEWAY のデプロイ	86
10.4. CEPH ORCHESTRATOR を使用したマルチサイト CEPH OBJECT GATEWAY のデプロイ	88
10.5. CEPH ORCHESTRATOR を使用した CEPH OBJECT GATEWAY の削除	94
第11章 CEPH ORCHESTRATOR を使用した NFS-GANESHA ゲートウェイの管理 (利用制限あり)	96
11.1. 前提条件	96
11.2. CEPH ORCHESTRATOR を使用した NFS-GANESHA クラスターの作成	96
11.3. コマンドラインインターフェースを使用した NFS-GANESHA ゲートウェイのデプロイ	98
11.4. サービス指定を使用した NFS-GANESHA ゲートウェイのデプロイ	100
11.5. CEPH ORCHESTRATOR を使用した NFS-GANESHA クラスターの更新	102
11.6. CEPH ORCHESTRATOR を使用した NFS-GANESHA クラスター情報の確認	103
11.7. CEPH ORCHESTRATOR を使用した NFS-GANESHA CLUTSER ログの取得	104
11.8. CEPH ORCHESTRATOR を使用したカスタム NFS-GANESHA 設定の設定	105
11.9. CEPH ORCHESTRATOR を使用したカスタム NFS-GANESHA 設定のリセット	109
11.10. CEPH ORCHESTRATOR を使用した NFS-GANESHA クラスターの削除	110
11.11. CEPH ORCHESTRATOR を使用した NFS-GANESHA ゲートウェイの削除	111
第12章 CEPH ORCHESTRATOR を使用した ISCSI ゲートウェイの管理 (利用制限あり)	113
12.1. 前提条件	113
12.2. コマンドラインインターフェースを使用した ISCSI ゲートウェイのデプロイ	113
12.3. サービス指定を使用した ISCSI ゲートウェイのデプロイ	114
12.4. CEPH ORCHESTRATOR を使用した ISCSI ゲートウェイの削除	116
第13章 SNMP トラップの設定	118
13.1. 簡易ネットワーク管理プロトコル	118
13.2. SNMPTRAPD の設定	119
13.3. SNMP ゲートウェイのデプロイ	122
第14章 ノードの障害の処理	127
14.1. 前提条件	128
14.2. ノードの追加または削除前の考慮事項	128
14.3. パフォーマンスに関する考慮事項	128
14.4. ノードの追加または削除に関する推奨事項	129
14.5. CEPH OSD ノードの追加	130
14.6. CEPH OSD ノードの削除	132
14.7. ノードの障害のシミュレーション	133
第15章 データセンター障害の処理	135

15.1. 前提条件	135
15.2. データセンター障害の回避	135
15.3. データセンター障害の処理	136

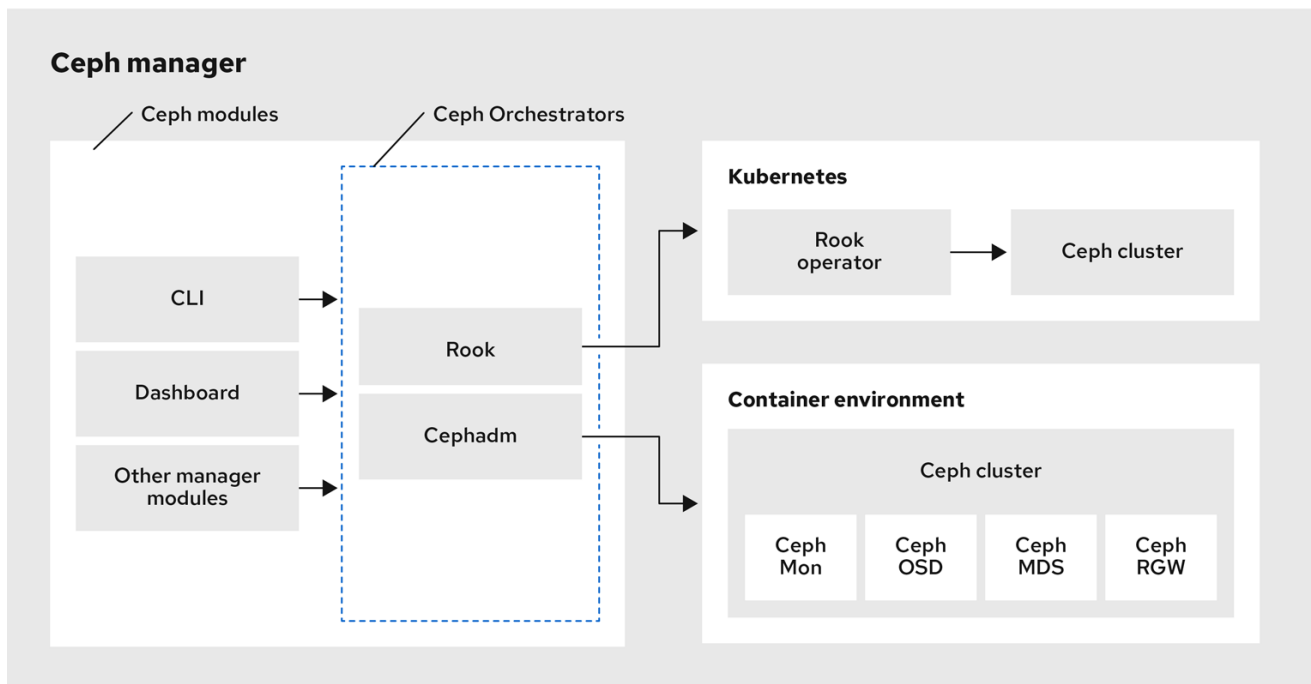
第1章 CEPH ORCHESTRATOR の紹介

ストレージ管理者として、Red Hat Ceph Storage クラスタでデバイスを検出し、サービスを作成する機能を提供する Ceph Orchestrator with Cephadm ユーティリティーを使用できます。

1.1. CEPH ORCHESTRATOR の使用

Red Hat Ceph Storage Orchestrators は、主に Red Hat Ceph Storage クラスタと、Rook や Cephadm などのデプロイメントツール間のブリッジとして機能するマネージャーモジュールです。また、Ceph コマンドラインインターフェースおよび Ceph Dashboard と統合します。

以下は、Ceph Orchestrator のワークフローの図です。



153_Ceph_0421

Red Hat Ceph Storage Orchestrators の種類

Red Hat Ceph Storage Orchestrators には、主に 3 つのタイプがあります。

- **Orchestrator CLI:** これらは Orchestrators で使用される一般的な API で、実装できるコマンドセットが含まれます。これらの API は、**ceph-mgr** モジュールを外部オーケストレーションサービスとオーケストレートする一般的なコマンドラインインターフェース (CLI) も提供しています。以下は、Ceph Orchestrator で使用される用語です。
 - **ホスト:** これは、物理ホストのホスト名で、コンテナ内の Pod 名、DNS 名、コンテナ名、またはホスト名ではありません。
 - **サービスタイプ:** これは、nfs、mds、osd、mon、ragw、mgr、iscsi などのサービスのタイプです。
 - **サービス:** モニターサービス、マネージャーサービス、OSD サービス、Ceph Object Gateway サービス、NFS サービスなど、Ceph Storage クラスタが提供する機能サービス。

- デーモン: Ceph Object Gateway サービスなどの1つ以上のホストによってデプロイされるサービスの特定のインスタンスでは、3つの異なるホストで異なる Ceph Object Gateway デーモンを実行することができます。
- **Cephadm Orchestrator:** これは、Rook または Ansible などの外部ツールに依存しない Ceph Orchestrator モジュールであり、SSH 接続を確立し、明示的な管理コマンドを実行することでクラスタのノードを管理します。このモジュールは Day One および Day Two 操作を対象としています。

Cephadm Orchestrator の使用は、Ansible などのデプロイメントフレームワークを使用せずに Ceph Storage クラスタをインストールする推奨方法です。これは、ストレージデバイスのインベントリを作成、OSD のデプロイメントと交換、Ceph デーモンの起動と停止など、あらゆる管理操作を実行するために、クラスタ内のすべてのノードに接続できる SSH 設定および鍵にアクセスできるマネージャーデーモンを提供するものです。さらに、Cephadm Orchestrator は、同一の場所に配置されたサービスの独立したアップグレードを可能にするために、**systemd** によって管理されるコンテナイメージをデプロイします。

また、このオーケストレーターでは、Ceph Monitor および Ceph Manager を実行する最低限のクラスタをブートストラップするコマンドなど、現在のホストでコンテナイメージベースのサービスのデプロイメントを管理するために、必要なすべての操作をカプセル化するツールが重要なポイントとなるでしょう。

- **Rook Orchestrator:** Rook は、Kubernetes Rook operator を使用して Kubernetes クラスタ内で実行される Ceph ストレージクラスタを管理するオーケストレーションツールです。rook モジュールは、Ceph の Orchestrator フレームワークと Rook との間の統合を提供します。Rook は、Kubernetes のオープンソースクラウドネイティブストレージ operator です。Rook は「operator」モデルに従い、ここでは Ceph ストレージクラスタおよびその望ましい状態を記述するためにカスタムリソース定義 (CRD) オブジェクトが Kubernetes で定義されます。rook operator デーモンは、現在のクラスタ状態と望ましい状態を比較する制御ループで実行され、それらを収束させるための手順を実行します。Ceph の望ましい状態を記述する主なオブジェクトは Ceph ストレージクラスタ CRD で、どのデバイスが OSD によって消費されるか、実行すべきモニターの数、使用すべき Ceph バージョンなどの情報が含まれます。Rook は、RBD プール、CephFS ファイルシステムなどを記述するために他の複数の CRD を定義します。

Rook Orchestrator モジュールは、**ceph-mgr** デーモンで実行されるグルーで、必要なクラスタの状態を記述する Kubernetes で Ceph ストレージクラスタに変更を加えて Ceph オーケストレーション API を実装します。Rook クラスタの **ceph-mgr** デーモンは Kubernetes Pod として実行されているため、rook モジュールは明示的な設定がなくても Kubernetes API に接続できます。

第2章 CEPH ORCHESTRATOR を使用したサービスの管理

ストレージ管理者は、Red Hat Ceph Storage クラスターのインストール後に、Ceph Orchestrator を使用してストレージクラスターのサービスを監視および管理できます。サービスは、一緒に設定されるデーモンのグループです。

本セクションでは、以下の管理情報について説明します。

- [Ceph Orchestrator の配置指定](#)。
- [コマンドラインインターフェースを使用した Ceph デーモンのデプロイ](#)。
- [コマンドラインインターフェースを使用したホストのサブセットでの Ceph デーモンのデプロイ](#)。
- [Ceph Orchestrator のサービス仕様](#)。
- [サービス仕様を使用した Ceph デーモンのデプロイ](#)。

2.1. CEPH ORCHESTRATOR の配置指定

Ceph Orchestrator を使用して、**osds**、**mons**、**mgrs**、**mds**、**rgw**、および **iSCSI** サービスをデプロイできます。Red Hat は、配置指定を使用してサービスをデプロイすることを推奨します。Ceph Orchestrator を使用して、サービスをデプロイするためにデプロイする必要があるデーモンの場所と数を把握する必要があります。配置の仕様は、コマンドライン引数または **yaml** ファイルのサービス仕様として渡すことができます。

配置指定を使ってサービスをデプロイする方法は2つあります。

- コマンドラインインターフェースで配置指定を直接使用します。たとえば、ホストに3つのモニターをデプロイする場合は、以下のコマンドを実行して **host01**、**host02**、および **host03** に3つのモニターをデプロイします。

例

```
[ceph: root@host01 /]# ceph orch apply mon --placement="3 host01 host02 host03"
```

- YAML ファイルでの配置指定の使用。たとえば、すべてのホストに **node-exporter** をデプロイする場合には、**yaml** ファイルで以下を指定できます。

例

```
service_type: node-exporter
placement:
host_pattern: '*'
```

2.2. コマンドラインインターフェースを使用した CEPH デーモンのデプロイ

Ceph Orchestrator を使用すると、**ceph orch** コマンドを使用して Ceph Manager、Ceph Monitors、Ceph OSD、モニタリングスタックなどのデーモンをデプロイできます。配置の指定は、Orchestrator コマンドの **--placement** 引数として渡されます。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- ホストがストレージクラスターに追加されている。

手順

1. Cephadm シェルにログインします。

例

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

2. 以下のいずれかの方法を使用して、ホストにデーモンをデプロイします。

- **方法 1:** デーモンの数とホスト名を指定します。

構文

```
ceph orch apply SERVICE_NAME --placement="NUMBER_OF_DAEMONS  
HOST_NAME_1 HOST_NAME_2 HOST_NAME_3"
```

例

```
[ceph: root@host01 /]# ceph orch apply mon --placement="3 host01 host02 host03"
```

- **方法 2:** ラベルをホストに追加してから、ラベルを使用してデーモンをデプロイします。
 - a. ホストにラベルを追加します。

構文

```
ceph orch host label add HOSTNAME_1 LABEL
```

例

```
[ceph: root@host01 /]# ceph orch host label add host01 mon
```

- b. ラベルでデーモンをデプロイします。

構文

```
ceph orch apply DAEMON_NAME label:LABEL
```

例

```
ceph orch apply mon label:mon
```

- **方法 3:** ラベルをホストに追加し、**--placement** 引数を使用してデプロイします。
 - a. ホストにラベルを追加します。

構文

```
ceph orch host label add HOSTNAME_1 LABEL
```

例

```
[ceph: root@host01 /]# ceph orch host label add host01 mon
```

- b. ラベルの配置指定を使用してデーモンをデプロイします。

構文

```
ceph orch apply DAEMON_NAME --placement="label:LABEL"
```

例

```
ceph orch apply mon --placement="label:mon"
```

検証

- サービスを一覧表示します。

例

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

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

```
ceph orch ps --daemon_type=DAEMON_NAME
ceph orch ps --service_name=SERVICE_NAME
```

例

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=mon
[ceph: root@host01 /]# ceph orch ps --service_name=mon
```

関連情報

- 『Red Hat Ceph Storage Operations Guide』の「[Adding hosts using the Ceph Orchestrator](#)」セクションを参照してください。

2.3. コマンドラインインターフェースを使用したホストのサブセットへの CEPH デーモンのデプロイ

--placement オプションを使用して、ホストのサブセットにデーモンをデプロイできます。デーモンをデプロイするホストの名前で、配置指定のデーモン数を指定できます。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- ホストがクラスターに追加されている。

手順

1. Cephadm シェルにログインします。

例

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

2. Ceph デーモンをデプロイするホストを一覧表示します。

例

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

3. デーモンをデプロイします。

構文

```
ceph orch apply SERVICE_NAME --placement="NUMBER_OF_DAEMONS  
HOST_NAME_1 _HOST_NAME_2 HOST_NAME_3"
```

例

```
ceph orch apply mgr --placement="2 host01 host02 host03"
```

この例では、**mgr** デーモンは、2つのホストにのみデプロイされます。

検証

- ホストを一覧表示します。

例

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

関連情報

- 『Red Hat Ceph Storage Operations Guide』の「[Listing hosts using the Ceph Orchestrator](#)」セクションを参照してください。

2.4. CEPH ORCHESTRATOR のサービス指定

サービス指定は、Ceph サービスのデプロイに使用されるサービス属性および設定を指定するデータ構造です。以下は、サービス指定を指定するためのマルチドキュメント YAML ファイル **cluster.yml** の例です。

例

```

service_type: mon
placement:
  host_pattern: "mon*"
---
service_type: mgr
placement:
  host_pattern: "mgr*"
---
service_type: osd
service_id: default_drive_group
placement:
  host_pattern: "osd*"
data_devices:
  all: true

```

以下は、サービス指定のプロパティが以下のように定義されるパラメーターのリストです。

- **service_type**: サービスのタイプ
 - mon、crash、mds、mgr、osd、rbd、rbd-mirror などの Ceph サービス。
 - nfs や rgw などの Ceph ゲートウェイ。
 - Alertmanager、Prometheus、Grafana、または Node-exporter などのモニタリングスタック。
 - カスタムコンテナのコンテナ。
- **service_id**: サービスの一意名
- **placement**: これは、デーモンの場所とデプロイ方法を定義するために使用されます。
- **unmanaged: true** に設定すると、Orchestrator は、このサービスに関連付けられたデーモンをデプロイまたは削除しません。

Orchestrator のステートレスサービス

ステートレスサービスは、状態の情報を利用可能にする必要がないサービスです。たとえば、**rgw** サービスを起動するには、サービスの起動または実行に追加情報は必要ありません。**rgw** サービスは、機能を提供するためにこの状態に関する情報を作成しません。**rgw** サービスを開始されるタイミングに関係なく、状態は同じになります。

2.5. サービス指定を使用した CEPH デーモンのデプロイ

Ceph Orchestrator を使用すると、YAML ファイルのサービス指定を使用して Ceph Manager、Ceph Monitors、Ceph OSD、モニタリングスタックなどのデーモンをデプロイできます。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- すべてのノードへの root レベルのアクセス。

手順

1. Cephadm シェルにログインします。

例

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

- 以下のディレクトリーに移動します。

構文

```
cd /var/lib/ceph/DAEMON_PATH/
```

例

```
[ceph: root@host01 /]# cd /var/lib/ceph/mon/
```

- yml** ファイルを作成します。

例

```
[ceph: root@host01 mon]# touch mon.yml
```

- このファイルは2つの方法で設定できます。
 - ファイルを編集して、配置の指定にホストの詳細を含めます。

構文

```
service_type: SERVICE_NAME
placement:
  hosts:
    - HOST_NAME_1
    - HOST_NAME_2
```

例

```
service_type: mon
placement:
  hosts:
    - host01
    - host02
    - host03
```

- ファイルを編集し、ラベルの詳細を配置指定に含めます。

構文

```
service_type: SERVICE_NAME
placement:
  label: "_LABEL_1"
```

例


```
service_type: mon
placement:
  label: "mon"
```

5. サービス指定を使用して Ceph デーモンをデプロイします。

構文

```
ceph orch apply -i FILE_NAME.yaml
```

例

```
[ceph: root@host01 mon]# ceph orch apply -i mon.yaml
```

検証

- サービスを一覧表示します。

例

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

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

```
ceph orch ps --daemon_type=DAEMON_NAME
```

例

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=mon
```

関連情報

- 『Red Hat Ceph Storage Operations Guide』の「[Listing hosts using the Ceph Orchestrator](#)」セクションを参照してください。

第3章 CEPH ORCHESTRATOR を使用したホストの管理

ストレージ管理者は、バックエンドで Cephadm で Ceph Orchestrator を使用し、既存の Red Hat Ceph Storage クラスタでホストを追加、一覧表示、および削除できます。

ホストにラベルを追加することもできます。ラベルは自由形式であり、特別な意味はありません。各ホストに複数のラベルを指定できます。たとえば、Monitor デーモンがデプロイされたすべてのホストに **mon** ラベルを適用し、Manager デーモンがデプロイされたすべてのホストに **mgr** を適用し、Ceph オブジェクトゲートウェイに **rgw** を適用します。

ストレージクラスター内のすべてのホストにラベルを付けると、各ホストで実行されているデーモンをすばやく識別できるため、システム管理タスクが簡素化されます。さらに、Ceph Orchestrator または YAML ファイルを使用して、特定のホストラベルを持つホストにデーモンをデプロイまたは削除できます。

本セクションでは、以下の管理タスクを説明します。

- [Ceph Orchestrator を使用したホストの追加](#)
- [Ceph Orchestrator を使用した複数ホストの追加](#)
- [Ceph Orchestrator を使用したホストの一覧表示](#)
- [Ceph Orchestrator を使用したホストへのラベルの追加](#)
- [Ceph Orchestrator を使用したホストの削除](#)
- [Ceph Orchestrator を使用したホストのメンテナンスモード](#)

3.1. 前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- すべてのノードへの root レベルのアクセス。
- 新しいホストの IP アドレスは `/etc/hosts` ファイルで更新する必要があります。

3.2. CEPH ORCHESTRATOR を使用したホストの追加

バックエンドで Cephadm で Ceph Orchestrator を使用して、ホストを既存の Red Hat Ceph Storage クラスタに追加できます。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- ストレージクラスター内のすべてのノードへの root レベルのアクセス。
- ノードを CDN に登録して、サブスクリプションを割り当てます。
- ストレージクラスター内のすべてのノードへの sudo アクセスおよびパスワードなしの **ssh** アクセスのある Ansible ユーザー。

手順

1. Ceph 管理ノードから、Cephadm シェルにログインします。

例

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

2. クラスターの SSH 公開鍵をフォルダーに展開します。

構文

```
ceph cephadm get-pub-key > ~/PATH
```

例

```
[ceph: root@host01 /]# ceph cephadm get-pub-key > ~/ceph.pub
```

3. Ceph クラスターの SSH 公開鍵を、新たなホストの root ユーザーの **authorized_keys** ファイルにコピーします。

構文

```
ssh-copy-id -f -i ~/PATH root@HOST_NAME_2
```

例

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

4. Ansible 管理ノードから、新しいホストを Ansible インベントリーファイルに追加します。ファイルのデフォルトの場所は **/usr/share/cephadm-ansible/hosts/** です。以下の例は、一般的なインベントリーファイルの構造を示しています。

例

```
node1
node2
node3

[admin]
node0
```



注記

新しいホストを Ansible インベントリーファイルに追加し、ホストでプリフライト Playbook を実行している場合は、ステップ 6 に進みます。

5. **--limit** オプションを指定して、プリフライト Playbook を実行します。

構文

```
ansible-playbook -i INVENTORY_FILE cephadm-preflight.yml --limit NEWHOST
```

例

```
[ansible@admin cephadm-ansible]$ ansible-playbook -i hosts cephadm-preflight.yml --extra-vars "ceph_origin=rhcs" --limit host02
```

プリフライト Playbook は、新しいホストに **podman**、**lvm2**、**chronyd**、および **cephadm** をインストールします。インストールが完了すると、**cephadm** は **/usr/sbin/** ディレクトリーに配置されます。

6. Ceph 管理ノードから、**cephadm**オーケストレーターを使用して、ホストをストレージクラスターに追加します。

構文

```
ceph orch host add HOST_NAME IP_ADDRESS_OF_HOST
```

例

```
[ceph: root@host01 /]# ceph orch host add host02 10.69.265.25
```

検証

- ホストを一覧表示します。

例

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

関連情報

- 『Red Hat Ceph Storage Operations Guide』の「[Listing hosts using the Ceph Orchestrator](#)」セクションを参照してください。
- **cephadm-preflight** Playbook の詳細は、「[Running the preflight playbook](#)」を参照してください。
- 『Red Hat Ceph Storage Installation Guide』の「[Registering Red Hat Ceph Storage nodes to the CDN and attaching subscriptions](#)」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の [sudo アクセスが設定された Ansible ユーザーの作成](#) セクションを参照してください。

3.3. CEPH ORCHESTRATOR を使用した複数ホストの追加

バックエンドで Cephadm で Ceph Orchestrator を使用して、YAML ファイル形式でサービス指定を使用して、同時に Red Hat Ceph Storage クラスターに複数のホストを追加できます。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。

手順

1. Cephadm シェルにログインします。

例

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

2. 以下のディレクトリーに移動します。

構文

```
cd /var/lib/ceph/DAEMON_PATH/
```

例

```
[ceph: root@host01 /]# cd /var/lib/ceph/hosts/
```

3. **hosts.yml** ファイルを作成します。

例

```
[ceph: root@host01 hosts]# touch hosts.yml
```

4. **hosts.yml** ファイルを編集し、以下の詳細を追加します。

例

```
service_type: host
addr: host01
hostname: host01
labels:
- mon
- osd
- mgr
---
service_type: host
addr: host02
hostname: host02
labels:
- mon
- osd
- mgr
---
service_type: host
addr: host03
hostname: host03
labels:
- mon
- osd
```

5. サービス指定を使用してホストをデプロイします。

構文

```
ceph orch apply -i FILE_NAME.yml
```

例

```
[ceph: root@host01 hosts]# ceph orch apply -i hosts.yml
```

検証

- ホストを一覧表示します。

例

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

関連情報

- 『Red Hat Ceph Storage Operations Guide』の「[Listing hosts using the Ceph Orchestrator](#)」セクションを参照してください。

3.4. CEPH ORCHESTRATOR を使用したホストの一覧表示

Ceph クラスターのホストを Ceph Orchestrator で一覧表示できます。



注記

ホストの **STATUS** は、**ceph orch host ls** コマンドの出力では空白になります。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- ホストがストレージクラスターに追加されている。

手順

1. Cephadm シェルにログインします。

例

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

2. クラスターのホストを一覧表示します。

例

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

ホストの **STATUS** が空白であることを確認できます。これは想定内です。

3.5. CEPH ORCHESTRATOR を使用したホストへのラベルの追加

Ceph Orchestrator を使用して、既存の Red Hat Ceph Storage クラスター内のホストにラベルを追加できます。ラベルの例の一部は、ホストにデプロイされるサービスに基づいて、**mgr**、**mon**、および **osd** になります。

cephadm に特別な意味を持ち、**_** で始まる以下のホストラベルを追加することもできます。

- **_no_schedule**: このラベルは、**cephadm** がホスト上でデーモンをスケジュールまたはデプロイすることを阻止します。すでに Ceph デーモンが含まれている既存のホストに追加されると、これにより、**cephadm** は、自動的に削除されない OSD を除いて、それらのデーモンを別の場所に移動します。ホストに **_no_schedule** ラベルが追加されると、デーモンはそのホストにデプロイされません。ホストが削除される前にデーモンがドレインされると、そのホストに **_no_schedule** ラベルが設定されます。
- **_no_autotune_memory**: このラベルは、ホスト上のメモリーを自動調整しません。そのホスト上の1つ以上のデーモンに対して、**osd_memory_target_autotune** オプションまたは他の同様のオプションが有効になっている場合でも、デーモンメモリーが調整されることを阻止します。
- **_admin**: デフォルトでは、**_admin** ラベルはストレージクラスター内のブートストラップされたホストに適用され、**client.admin** キーは、**ceph orch client-keyring {ls|set|rm}** 関数でそのホストに配布されるように設定されます。このラベルを追加のホストに追加すると、通常、**cephadm** は設定ファイルとキーリングファイルを **/etc/ceph** にデプロイします。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- ホストがストレージクラスターに追加される。

手順

1. Cephadm シェルにログインします。

例

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

2. ホストにラベルを追加します。

構文

```
ceph orch host label add HOST_NAME LABEL_NAME
```

例

```
[ceph: root@host01 /]# ceph orch host label add host02 mon
```

検証

- ホストを一覧表示します。

例

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

-

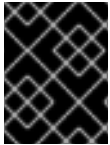
3.6. CEPH ORCHESTRATOR を使用したホストの削除

Ceph Orchestrator で、Ceph クラスターのホストを削除できます。また、ホストでコンテナを保持しないようにするため、ホストを削除する際に **node-exporter** および **crash** サービスを削除する必要があります。



注記

ストレージクラスターからホストを削除する前に、マネージャー、モニター、OSD などのすべてのサービスを手動で削除してください。



重要

ブートストラップホストを削除する場合は、ホストを削除する前に、必ず管理キーリングと設定ファイルをストレージクラスター内の別のホストにコピーしてください。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- すべてのノードへの root レベルのアクセス。
- ホストがストレージクラスターに追加されている。
- すべてのサービスがデプロイされている。
- Cephadm は、サービスを削除する必要があるノードにデプロイされます。

手順

1. Cephadm シェルにログインします。

例

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

2. **node-exporter** および **crash** を除くすべての Ceph サービスに対して、ホストを配置指定ファイルから削除します。

例

```
service_type: rgw
placement:
  hosts:
    - host01
    - host02
```

この例では、**host03** は Ceph オブジェクトゲートウェイサービスの配置指定を削除します。ホストにデプロイされたすべてのサービスに対して上記の手順を実行する必要があります。

- a. OSD の削除には **--unmanaged=True** で OSD をデプロイします。

例


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



注記

これにより、デバイスでの OSD の自動デプロイメントが阻止されます。

- b. OSD を削除します。

構文

```
for osd_id in $(ceph orch ps HOST_NAME --daemon_type osd | grep osd | awk '{print $1}' | cut -c 5-); do ceph orch osd rm $osd_id; done
```

例

```
[ceph: root@host01 /]# for osd_id in $(ceph orch ps host03 --daemon_type osd | grep osd | awk '{print $1}' | cut -c 5-); do ceph orch osd rm $osd_id; done
```

- c. ホストを削除します。

構文

```
ceph orch host rm HOST_NAME
```

例

```
[ceph: root@host01 /]# ceph orch host rm host03
```

3. **node-exporter** および **crash** サービスを削除する必要があるノードで、以下のコマンドを実行します。

- a. root ユーザーとして、Cephadm シェル外部で、クラスタの **fsid** の詳細とサービス名を取得します。

例

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

- b. **fsid** および **node-exporter** サービスの **name** をコピーします。

- c. サービスを削除します。

構文

```
cephadm rm-daemon --fsid CLUSTER_ID --name SERVICE_NAME
```

例

```
[root@host03 ~]# cephadm rm-daemon --fsid a34c81a0-889b-11eb-af98-001a4a00063d --name node-exporter.host03
```

検証

- **cephadm ls** コマンドを実行して、サービスの削除を確認します。

例

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

- ホスト、デーモン、およびプロセスを一覧表示します。

例

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

関連情報

- 詳細は『Red Hat Ceph Storage Operations Guide』の「[Adding hosts using the Ceph Orchestrator](#)」セクションを参照してください。
- 詳細は『Red Hat Ceph Storage Operations Guide』の「[Listing hosts using the Ceph Orchestrator](#)」セクションを参照してください。

3.7. CEPH ORCHESTRATOR を使用したホストのメンテナンスモード

Ceph Orchestrator を使用して、ホストのメンテナンスモードを切り替えられます。これにより、ホスト上のすべての Ceph デーモンが停止します。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- すべてのノードへの root レベルのアクセス。
- クラスタに追加されたホスト。

手順

1. Cephadm シェルにログインします。

例

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

2. ホストをメンテナンスモードにしたり、メンテナンスモードから解除したりできます。

- ホストをメンテナンスモードにします。

構文

```
ceph orch host maintenance enter HOST_NAME [--force]
```

例

```
[ceph: root@host01 /]# ceph orch host maintenance enter host02 --force
```

--force フラグを使用すると、ユーザーは警告を回避できますが、アラートは回避できません。

- ホストをメンテナンスモードから解除します。

構文

```
ceph orch host maintenance exit HOST_NAME
```

例

```
[ceph: root@host01 /]# ceph orch host maintenance exit host02
```

検証

- ホストを一覧表示します。

例

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

第4章 CEPH ORCHESTRATOR を使用したモニターの管理

ストレージ管理者は、配置指定を使用して追加のモニターをデプロイし、サービス指定を使用してモニターを追加し、サブネット設定にモニターを追加し、特定のホストにモニターを追加できます。さらに、Ceph Orchestrator を使用してモニターを削除できます。

デフォルトでは、一般的な Red Hat Ceph Storage クラスタには、3 つまたは 5 つのモニターデーモンが異なるホストにデプロイされます。

Red Hat は、クラスタに 5 つ以上のノードがある場合に、5 つのモニターをデプロイすることを推奨します。

Ceph は、クラスタの拡大とともにモニターデーモンを自動的にデプロイし、クラスタの縮小とともにモニターデーモンを自動的にスケールバックします。このような自動拡大および縮小のスムーズな実行は、適切なサブネット設定によります。

モニターノードまたはクラスタ全体が単一のサブネットにある場合、Cephadm は新規ホストをクラスタに追加する際に最大 5 つのモニターデーモンを自動的に追加します。Cephadm は、新しいホストでモニターデーモンを自動的に設定します。新しいホストは、ストレージクラスタのブートストラップされたホストと同じサブネットにあります。

Cephadm はストレージクラスタのサイズの変更に対応するようモニターをデプロイし、スケーリングすることもできます。

4.1. CEPH MONITOR

Ceph Monitor は、ストレージクラスタマップのマスターコピーを維持する軽量プロセスです。すべての Ceph クライアントは Ceph モニターに問い合わせ、ストレージクラスタマップの現在のコピーを取得し、クライアントがプールにバインドし、読み取りと書き込みを可能にします。

Ceph Monitor は Paxos プロトコルのバリエーションを使用して、ストレージクラスタ全体でマップやその他の重要な情報について合意を確立します。Paxos の性質上、Ceph は、クォーラムを確立するためにモニターの大部分を実行する必要があり、合意を確立します。



重要

Red Hat では、実稼働クラスタのサポートを受け取るために、別のホストで少なくとも 3 つのモニターが必要になります。

Red Hat は、奇数のモニターをデプロイすることを推奨します。奇数の Ceph モニターは、偶数のモニターよりも障害に対する回復性が高くなっています。たとえば、2 つのモニターデプロイメントでクォーラムを維持するには、Ceph は障害を許容できません。3 つのモニターでは障害を 1 つ、4 つのモニターでは障害を 1 つ、5 つのモニターでは障害を 2 つ許容します。このため、奇数も推奨されています。要約すると、Ceph は、モニターの大部分 (3 つのうち 2 つ、4 つのうち 3 つなど) が実行され、相互に通信できるようにする必要があります。

マルチノードの Ceph ストレージクラスタの初回のデプロイには、Red Hat では 3 つのモニターが必要です。3 つ以上のモニターが有効な場合には、一度に数を 2 つ増やします。

Ceph Monitor は軽量であるため、OpenStack ノードと同じホストで実行できます。ただし、Red Hat は、別のホストでモニターを実行することを推奨します。



重要

Red Hat は、コンテナ化された環境における Ceph サービスを共存させることのみをサポートしています。

ストレージクラスターからモニターを削除する場合、Ceph Monitor は Paxos プロトコルを使用して、マスターストレージクラスターマップに関する合意を確立することを検討してください。クォーラムを確立するには、十分な数の Ceph モニターが必要です。

関連情報

- サポートされているすべての Ceph 設定については、ナレッジベースアティクル「[Red Hat Ceph Storage でサポートされる構成](#)」を参照してください。

4.2. モニタリング選択ストラテジーの設定

モニター選択ストラテジーは、ネット分割を識別し、障害を処理します。選択モニターストラテジーは、3つの異なるモードで設定できます。

1. **classic** - これは、2つのサイト間のエレクターモジュールに基づいて、最も低いランクのモニターが投票されるデフォルトのモードです。
2. **disallow** - このモードでは、モニターを不許可とマークできます。この場合、モニターはクォーラムに参加してクライアントにサービスを提供しますが、選出されたリーダーになることはできません。これにより、許可されていないリーダーの一覧にモニターを追加できます。モニターが許可されていないリストにある場合、そのモニターは常に別のモニターに先送りされます。
3. **connectivity** - このモードは、主にネットワークの不一致を解決するために使用されます。ピアに対して各モニターによって提供される接続スコアを評価し、最も接続されたモニターと信頼できるモニターをリーダーに選択します。このモードは、クラスターが複数のデータセンターにまたがっている場合や影響を受けやすい場合に発生する可能性のあるネット分割を処理するように設計されています。このモードでは接続スコア評価が組み込まれ、最良スコアのモニターが選択されます。

他のモードで機能が不要でない限り、Red Hat は、**classic** モードに留まります。

クラスターを構築する前に、以下のコマンドで **election_strategy** を **classic**、**disallow**、または **connectivity** に変更します。

構文

```
ceph mon set election_strategy {classic|disallow|connectivity}
```

4.3. コマンドラインインターフェースを使用した CEPH モニターデーモンのデプロイ

Ceph Orchestrator はデフォルトで1つのモニターデーモンをデプロイします。コマンドラインインターフェースで **placement** 指定を使用して、追加のモニターデーモンをデプロイできます。異なる数のモニターデーモンをデプロイするには、別の数を指定します。モニターデーモンがデプロイされるホストを指定しないと、Ceph Orchestrator はホストをランダムに選択し、モニターデーモンをそれらにデプロイします。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- ホストがクラスタに追加されている。

手順

1. Cephadm シェルにログインします。

例

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

2. Ceph モニターデーモンをデプロイするには、以下の 4 つの方法があります。

方法 1

- 配置指定を使用して、ホストにモニターをデプロイします。



注記

Red Hat は **--placement** オプションを使用して特定のホストにデプロイすることを推奨します。

構文

```
ceph orch apply mon --placement="HOST_NAME_1 HOST_NAME_2 HOST_NAME_3"
```

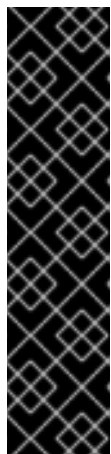
例

```
[ceph: root@host01 /]# ceph orch apply mon --placement="host01 host02 host03"
```



注記

コマンドの最初のノードとしてブートストラップノードが含まれるようにしてください。



重要

モニターを個別に追加しないでください。**ceph orch apply mon** は置き換えを行うため、モニターはすべてのホストに追加されません。たとえば、以下のコマンドを実行すると、最初のコマンドが **host01** にモニターを作成します。2 番目のコマンドは **host01** のモニターの代わりに **host02** にモニターを作成します。3 番目のコマンドは **host02** のモニターの代わりに **host03** にモニターを作成します。最終的には、3 番目のホストにのみモニターが存在することになります。

```
# ceph orch apply mon host01  
# ceph orch apply mon host02  
# ceph orch apply mon host03
```

方法 2

- 配置指定を使用して、ラベルの付いた特定ホストに特定数のモニターをデプロイします。
 - a. ホストにラベルを追加します。

構文

```
ceph orch host label add HOSTNAME_1 LABEL
```

例

```
[ceph: root@host01 /]# ceph orch host label add host01 mon
```

- b. デーモンをデプロイします。

構文

```
ceph orch apply mon --placement="HOST_NAME_1:mon HOST_NAME_2:mon  
HOST_NAME_3:mon"
```

例

```
[ceph: root@host01 /]# ceph orch apply mon --placement="host01:mon host02:mon  
host03:mon"
```

方法 3

- 配置指定を使用して、特定ホストに特定数のモニターをデプロイします。

構文

```
ceph orch apply mon --placement="NUMBER_OF_DAEMONS HOST_NAME_1  
HOST_NAME_2 HOST_NAME_3"
```

例

```
[ceph: root@host01 /]# ceph orch apply mon --placement="3 host01 host02 host03"
```

方法 4

- ストレージクラスターのホストにモニターデーモンを無作為にデプロイします。

構文

```
ceph orch apply mon NUMBER_OF_DAEMONS
```

例

```
[ceph: root@host01 /]# ceph orch apply mon 3
```

検証

- サービスを一覧表示します。

例

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

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

```
ceph orch ps --daemon_type=DAEMON_NAME
```

例

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=mon
```

4.4. サービス指定を使用した CEPH モニターデーモンのデプロイ

Ceph Orchestrator はデフォルトで1つのモニターデーモンをデプロイします。YAML形式のファイルなどのサービス指定を使用して、追加のモニターデーモンをデプロイできます。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- ホストがクラスターに追加されている。

手順

1. Cephadm シェルにログインします。

例

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

2. 以下のディレクトリーに移動します。

構文

```
cd /var/lib/ceph/DAEMON_PATH/
```

例

```
[ceph: root@host01 /]# cd /var/lib/ceph/mon/
```

3. **mon.yml** ファイルを作成します。

例


```
[ceph: root@host01 mon]# touch mon.yml
```

4. **mon.yml** ファイルを編集し、以下の詳細が含まれるようにします。

構文

```
service_type: mon
placement:
  hosts:
    - HOST_NAME_1
    - HOST_NAME_2
```

例

```
service_type: mon
placement:
  hosts:
    - host01
    - host02
```

5. モニターデーモンをデプロイします。

構文

```
ceph orch apply -i FILE_NAME.yml
```

例

```
[ceph: root@host01 mon]# ceph orch apply -i mon.yml
```

検証

- サービスを一覧表示します。

例

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

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

```
ceph orch ps --daemon_type=DAEMON_NAME
```

例

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=mon
```

4.5. CEPH ORCHESTRATOR を使用した特定ネットワークでのモニターデーモンのデプロイ

Ceph Orchestrator はデフォルトで1つのモニターデーモンをデプロイします。各モニターに IP アドレスまたは CIDR ネットワークを明示的に指定し、各モニターが配置される場所を制御できます。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- ホストがクラスタに追加されている。

手順

1. Cephadm シェルにログインします。

例

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

2. 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
```

3. 特定ネットワーク上のホストにモニターをデプロイします。

構文

```
ceph orch daemon add mon HOST_NAME_1:_IP_OR_NETWORK_
```

例

```
[ceph: root@host01 /]# ceph orch daemon add mon host03:10.1.2.123
```

検証

- サービスを一覧表示します。

例

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

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

```
ceph orch ps --daemon_type=DAEMON_NAME
```

例

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=mon
```

4.6. CEPH ORCHESTRATOR を使用したモニターデーモンの削除

ホストからモニターデーモンを削除するには、他のホストにモニターデーモンを再デプロイします。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- ホストがクラスタに追加されている。
- ホストにデプロイされたモニターデーモン1つ以上。

手順

1. Cephadm シェルにログインします。

例

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

2. **ceph orch apply** コマンドを実行して、必要なモニターデーモンをデプロイします。

構文

```
ceph orch apply mon "NUMBER_OF_DAEMONS_HOST_NAME_1 HOST_NAME_3"
```

host02 から monitor デーモンを削除する場合は、他のホストにモニターを再デプロイします。

例

```
[ceph: root@host01 /]# ceph orch apply mon "2 host01 host03"
```

検証

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

```
ceph orch ps --daemon_type=DAEMON_NAME
```

例

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=mon
```

関連情報

- 詳細は、『Red Hat Ceph Storage Operations Guide』の「[Deploying the Ceph monitor daemons using the command line interface](#)」セクションを参照してください。
- 詳細は、『Red Hat Ceph Storage Operations Guide』の「[Deploying the Ceph monitor daemons using the service specification](#)」セクションを参照してください。

4.7. 異常なストレージクラスターからの CEPH MONITOR の削除

正常でないストレージクラスターから、**ceph-mon** デーモンを削除できます。正常でないストレージクラスターとは、配置グループが永続的に **active + clean** 状態ではないストレージクラスターのことです。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- Ceph Monitor ノードへのルートレベルのアクセス。
- Ceph Monitor ノードが少なくとも 1 台実行している。

手順

1. 存続しているモニターを特定し、ホストにログインします。

構文

```
ssh root@MONITOR_ID
```

例

```
[root@node0 ~]# ssh root@node0
```

2. 各 Ceph Monitor ホストにログインし、すべての Ceph Monitor を停止します。

構文

```
cephadm unit --name DAEMON_NAME.HOSTNAME stop
```

例

```
[root@node0 ~]# cephadm unit --name mon.node0 stop
```

3. 拡張デーモンのメンテナンスに適した環境と、デーモンを対話的に実行するための環境を設定します。

構文

```
cephadm shell --name DAEMON_NAME.HOSTNAME
```

例

```
[root@node0 ~]# cephadm shell --name mon.node0
```

4. **monmap** ファイルのコピーを抽出します。

構文

```
ceph-mon -i HOSTNAME --extract-monmap TEMP_PATH
```

-

例

```
[ceph: root@node0 /]# ceph-mon -i node1 --extract-monmap /tmp/monmap
2022-01-05T11:13:24.440+0000 7f7603bd1700 -1 wrote monmap to /tmp/monmap
```

5. Ceph Monitor 以外を削除します。

構文

```
monmaptool TEMPORARY_PATH --rm HOSTNAME
```

例

```
[ceph: root@node0 /]# monmaptool /tmp/monmap --rm node1
```

6. 削除されたモニターを含む存続しているモニターマップを、存続している Ceph モニターに挿入します。

構文

```
ceph-mon -i HOSTNAME --inject-monmap TEMP_PATH
```

例

```
[ceph: root@node0 /]# ceph-mon -i node1 --inject-monmap /tmp/monmap
```

7. 存続しているモニターのみを起動します。

構文

```
cephadm unit --name DAEMON_NAME.HOSTNAME start
```

例

```
[root@node0 ~]# cephadm unit --name mon.node0 start
```

8. モニターがクォーラムを形成していることを確認します。

例

```
[ceph: root@node0 /]# ceph -s
```

9. オプション: 削除された Ceph Monitor のデータディレクトリーを `/var/lib/ceph/CLUSTER_FSID/mon.HOSTNAME` ディレクトリーにアーカイブします。

第5章 CEPH ORCHESTRATOR を使用したマネージャーの管理

ストレージ管理者は、Ceph Orchestrator を使用して追加のマネージャーデーモンをデプロイできます。Cephadm は、ブートストラッププロセス中にブートストラップノードにマネージャーデーモンを自動的にインストールします。

本セクションでは、以下の管理タスクを説明します。

- [Ceph Orchestrator を使用したマネージャーデーモンのデプロイ](#)
- [Ceph Orchestrator を使用したマネージャーデーモンの削除](#)

5.1. 前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- すべてのノードへの root レベルのアクセス。
- ホストがクラスタに追加されている。

5.2. CEPH ORCHESTRATOR を使用したマネージャーデーモンのデプロイ

Ceph Orchestrator はデフォルトで 2 つの Manager デーモンをデプロイします。コマンドラインインターフェースで **placement** 指定を使用して、追加のマネージャーデーモンをデプロイできます。異なる数の Manager デーモンをデプロイするには、別の数を指定します。Manager デーモンがデプロイされるホストを指定しないと、Ceph Orchestrator はホストをランダムに選択し、Manager デーモンをそれらにデプロイします。



注記

デプロイメントごとに少なくとも 3 つの Ceph Manager がデプロイメントに含まれるようにします。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- ホストがクラスタに追加されている。

手順

1. Cephadm シェルにログインします。

例

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

2. マネージャーデーモンをデプロイする方法は 2 つあります。

方法 1

- 特定のホストセットに配置指定を使用してマネージャーデーモンをデプロイします。



注記

Red Hat は **--placement** オプションを使用して特定のホストにデプロイすることを推奨します。

構文

```
ceph orch apply mgr --placement=" HOST_NAME_1 HOST_NAME_2 HOST_NAME_3"
```

例

```
[ceph: root@host01 /]# ceph orch apply mgr --placement="host01 host02 host03"
```

方法 2

- ストレージクラスターのホストにマネージャーデーモンを無作為にデプロイします。

構文

```
ceph orch apply mgr NUMBER_OF_DAEMONS
```

例

```
[ceph: root@host01 /]# ceph orch apply mgr 3
```

検証

- サービスを一覧表示します。

例

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

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

```
ceph orch ps --daemon_type=DAEMON_NAME
```

例

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=mgr
```

5.3. CEPH ORCHESTRATOR を使用したマネージャーデーモンの削除

ホストからマネージャーデーモンを削除するには、他のホストにデーモンを再デプロイします。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。

- すべてのノードへの root レベルのアクセス。
- ホストがクラスターに追加されている。
- ホストにデプロイされたマネージャーデーモン1つ以上。

手順

1. Cephadm シェルにログインします。

例

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

2. **ceph orch apply** コマンドを実行して、必要なマネージャーデーモンを再デプロイします。

構文

```
ceph orch apply mgr "NUMBER_OF_DAEMONS _HOST_NAME_1 HOST_NAME_3"
```

host02 からマネージャーデーモンを削除する場合は、他のホストにマネージャーデーモンを再デプロイします。

例

```
[ceph: root@host01 /]# ceph orch apply mgr "2 host01 host03"
```

検証

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

```
ceph orch ps --daemon_type=DAEMON_NAME
```

例

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=mgr
```

関連情報

- 詳細は『Red Hat Ceph Storage Operations Guide』の「[Deploying the manager daemons using the Ceph Orchestrator](#)」セクションを参照してください。

5.4. CEPH MANAGER バランサーモジュールの使用

バランサーは、Ceph Manager (**ceph-mgr**) のモジュールで、OSD 全体の配置グループ (PG) の配置を最適化することで、自動または監視された方法でバランスの取れた分散を実現します。

現在、バランサーモジュールを無効にできません。これをオフにして設定をカスタマイズすることしかできません。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。

手順

1. balancer モジュールが有効になっていることを確認します。

例

```
[ceph: root@host01 /]# ceph mgr module enable balancer
```

2. balancer モジュールをオンにします。

例

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

3. デフォルトのモードは **crush-compat** です。モードは以下のように変更できます。

例

```
[ceph: root@host01 /]# ceph balancer mode upmap
```

または

例

```
[ceph: root@host01 /]# ceph balancer mode crush-compat
```

Status

balanサーの現在のステータスは、以下を実行していつでも確認できます。

例

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

自動バランシング

デフォルトでは、balancer モジュールをオンにする場合、自動分散が使用されます。

例

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

以下を使用して、balancer を再度オフにできます。

例

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

これには、古いクライアントと後方互換性があり、時間の経過とともにデータディストリビューションに小さな変更を加えて、OSD を同等に利用されるようにする **crush-compat** モードを使用します。

スロットリング

たとえば、OSD が失敗し、システムがまだ修復していない場合などに、クラスターのパフォーマンスが低下する場合は、PG ディストリビューションには調整は行われません。

クラスターが正常な場合、 balancer は変更を調整して、置き間違えた、または移動する必要のある PG の割合がデフォルトで 5% のしきい値を下回るようにします。このパーセンテージは、 **max_misplaced** 設定を使用して調整できます。たとえば、しきい値を 7% に増やすには、次のコマンドを実行します。

例

```
[ceph: root@host01 /]# ceph config-key set mgr/balancer/target_max_misplaced .07
```

監視付き最適化

balancer 操作はいくつかの異なるフェーズに分類されます。

1. プラン の構築。
2. 現在の PG ディストリビューションまたは プラン 実行後に得られる PG ディストリビューションに対するデータ分散の品質の評価。
3. プラン の実行。
 - 現在のディストリビューションを評価し、スコアを付けます。

例

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

- 単一プールのディストリビューションを評価するには、以下を実行します。

構文

```
ceph balancer eval POOL_NAME
```

例

```
[ceph: root@host01 /]# ceph balancer eval rbd
```

- 評価の詳細を表示するには、以下を実行します。

例

```
[ceph: root@host01 /]# ceph balancer eval-verbose ...
```

- 現在設定されているモードを使用してプランを生成するには、以下を実行します。

構文

```
ceph balancer optimize PLAN_NAME
```

PLAN_NAME は、カスタムプラン名に置き換えます。

例

```
[ceph: root@host01 /]# ceph balancer optimize rbd_123
```

- プランの内容を表示するには、以下を実行します。

構文

```
ceph balancer show PLAN_NAME
```

例

```
[ceph: root@host01 /]# ceph balancer show rbd_123
```

- 古いプランを破棄するには、以下を実行します。

構文

```
ceph balancer rm PLAN_NAME
```

例

```
[ceph: root@host01 /]# ceph balancer rm rbd_123
```

- 現在記録されているプランを表示するには、status コマンドを使用します。

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

- プラン実行後に生じるディストリビューションの品質を計算するには、以下を実行します。

構文

```
ceph balancer eval PLAN_NAME
```

例

```
[ceph: root@host01 /]# ceph balancer eval rbd_123
```

- プランを実行するには、以下を実行します。

構文

```
ceph balancer execute PLAN_NAME
```

例

```
[ceph: root@host01 /]# ceph balancer execute rbd_123
```



注記

ディストリビューションの改善が想定される場合にのみ、プランを実行します。実行後、プランは破棄されます。

5.5. CEPHMANAGER アラートモジュールの使用

Ceph Manager アラートモジュールを使用して、Red Hat Ceph Storage クラスターの健全性に関する簡単なアラートメッセージを電子メールで送信できます。



注記

このモジュールは、堅牢な監視ソリューションを目的としたものではありません。Ceph クラスター自体の一部として実行されるため、**ceph-mgr** デーモンに障害が発生するとアラートが送信されないという根本的な制約があります。ただし、このモジュールは、既存の監視インフラストラクチャーが存在しない環境に存在するスタンドアロンクラスターには役立ちます。

前提条件

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

手順

1. Cephadm シェルにログインします。

例

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

2. アラートモジュールを有効にします。

例

```
[ceph: root@host01 /]# ceph mgr module enable alerts
```

3. アラートモジュールが有効になっていることを確認します。

例

```
[ceph: root@host01 /]# ceph mgr module ls | more
{
  "always_on_modules": [
    "balancer",
    "crash",
    "devicehealth",
    "orchestrator",
    "pg_autoscaler",
```

```

    "progress",
    "rbd_support",
    "status",
    "telemetry",
    "volumes"
  ],
  "enabled_modules": [
    "alerts",
    "cephadm",
    "dashboard",
    "iostat",
    "nfs",
    "prometheus",
    "restful"
  ]
]

```

4. Simple Mail Transfer Protocol (SMTP) を設定します。

構文

```

ceph config set mgr mgr/alerts/smtp_host SMTP_SERVER
ceph config set mgr mgr/alerts/smtp_destination RECEIVER_EMAIL_ADDRESS
ceph config set mgr mgr/alerts/smtp_sender SENDER_EMAIL_ADDRESS

```

例

```

[ceph: root@host01 /]# ceph config set mgr mgr/alerts/smtp_host smtp.example.com
[ceph: root@host01 /]# ceph config set mgr mgr/alerts/smtp_destination
example@example.com
[ceph: root@host01 /]# ceph config set mgr mgr/alerts/smtp_sender
example2@example.com

```

5. オプション: デフォルトでは、アラートモジュールは SSL とポート 465 を使用します。これを変更するには、**smtp_ssl** を **false** に設定します。

構文

```

ceph config set mgr mgr/alerts/smtp_ssl false
ceph config set mgr mgr/alerts/smtp_port PORT_NUMBER

```

例

```

[ceph: root@host01 /]# ceph config set mgr mgr/alerts/smtp_ssl false
[ceph: root@host01 /]# ceph config set mgr mgr/alerts/smtp_port 587

```

6. SMTP サーバーへの認証:

構文

```

ceph config set mgr mgr/alerts/smtp_user USERNAME
ceph config set mgr mgr/alerts/smtp_password PASSWORD

```

例

```
[ceph: root@host01 /]# ceph config set mgr mgr/alerts/smtp_user admin1234
[ceph: root@host01 /]# ceph config set mgr mgr/alerts/smtp_password admin1234
```

- オプション: デフォルトでは、SMTP **From** 名は **Ceph** です。これを変更するには、**smtp_from_name** パラメーターを設定します。

構文

```
ceph config set mgr mgr/alerts/smtp_from_name CLUSTER_NAME
```

例

```
[ceph: root@host01 /]# ceph config set mgr mgr/alerts/smtp_from_name 'Ceph Cluster Test'
```

- オプション: デフォルトでは、アラートモジュールはストレージクラスターの健全性を毎分チェックし、クラスターの健全ステータスに変更があった場合にメッセージを送信します。頻度を変更するには、**interval** パラメーターを設定します。

構文

```
ceph config set mgr mgr/alerts/interval INTERVAL
```

例

```
[ceph: root@host01 /]# ceph config set mgr mgr/alerts/interval "5m"
```

この例では、間隔は 5 分に設定されています。

- オプション: アラートをすぐに送信します。

例

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

関連情報

- Ceph の健全性メッセージの詳細は、『[Red Hat Ceph Storage Troubleshooting Guide](#)』の「[Health messages of a Ceph cluster](#)」セクションを参照してください。

5.6. CEPH MANAGER クラッシュモジュールの使用

Ceph Manager クラッシュモジュールを使用すると、デーモンの `crashdump` に関する情報を収集し、これを Red Hat Ceph Storage クラスターに保存して詳細な分析を行うことができます。

デフォルトでは、デーモンの `crashdump` は `/var/lib/ceph/crash` にダンプされます。**crash_dir** オプションを使用して設定できます。クラッシュディレクトリーの名前は、時間、日付、およびランダムに生成される UUID で名前が付けられ、同じ **crash_id** を持つメタデータファイルの **meta** と最近のログファイルが含まれます。

ceph-crash.service を使用して、これらのクラッシュを自動的に送信し、Ceph Monitor で永続化することができます。**ceph-crash.service** は `crashdump` ディレクトリーを監視し、それらを **ceph crash post** でアップロードします。

RECENT_CRASH ヘルスメッセージは、Ceph クラスタ内の最も一般的なヘルスマッセージのいずれかとなります。このヘルスマッセージは、1つ以上の Ceph デーモンが最近クラッシュし、そのクラッシュが管理者によってアーカイブまたは確認されていないことを意味します。これは、ソフトウェアのバグや、障害のあるディスクなどのハードウェアの問題があることを示している可能性があります。オプション **mgr/crash/warn_recent_interval** は、最新の方法の期間 (デフォルトでは 2 週間) を制御します。以下のコマンドを実行して警告を無効にすることができます。

例

```
[ceph: root@host01 /]# ceph config set mgr/crash/warn_recent_interval 0
```

mgr/crash/retain_interval オプションは、自動的にパージされるまでクラッシュレポートを保持する期間を制御します。このオプションのデフォルトは1年です。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。

手順

1. crash モジュールが有効になっていることを確認します。

例

```
[ceph: root@host01 /]# ceph mgr module ls | more
{
  "always_on_modules": [
    "balancer",
    "crash",
    "devicehealth",
    "orchestrator_cli",
    "progress",
    "rbd_support",
    "status",
    "volumes"
  ],
  "enabled_modules": [
    "dashboard",
    "pg_autoscaler",
    "prometheus"
  ]
}
```

2. クラッシュダンプの保存: **meta** ファイルは、メタとして crash ディレクトリーに保存されている JSON blob です。ceph コマンド **-i** オプションを呼び出すことができます。これは、stdin から読み取ります。

例

```
[ceph: root@host01 /]# ceph crash post -i meta
```

3. 新しいクラッシュ情報およびアーカイブされたすべてのクラッシュ情報のタイムスタンプまたは UUID クラッシュ ID を表示します。

例

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

- すべての新規クラッシュ情報のタイムスタンプまたは UUID クラッシュ ID を一覧表示します。

例

```
[ceph: root@host01 /]# ceph crash ls-new
```

- すべての新規クラッシュ情報のタイムスタンプまたは UUID クラッシュ ID を一覧表示します。

例

```
[ceph: root@host01 /]# ceph crash ls-new
```

- 保存されたクラッシュ情報の概要を経過時間別にグループ化して一覧表示します。

例

```
[ceph: root@host01 /]# ceph crash stat
8 crashes recorded
8 older than 1 days old:
2021-05-20T08:30:14.533316Z_4ea88673-8db6-4959-a8c6-0eea22d305c2
2021-05-20T08:30:14.590789Z_30a8bb92-2147-4e0f-a58b-a12c2c73d4f5
2021-05-20T08:34:42.278648Z_6a91a778-bce6-4ef3-a3fb-84c4276c8297
2021-05-20T08:34:42.801268Z_e5f25c74-c381-46b1-bee3-63d891f9fc2d
2021-05-20T08:34:42.803141Z_96adfc59-be3a-4a38-9981-e71ad3d55e47
2021-05-20T08:34:42.830416Z_e45ed474-550c-44b3-b9bb-283e3f4cc1fe
2021-05-24T19:58:42.549073Z_b2382865-ea89-4be2-b46f-9a59af7b7a2d
2021-05-24T19:58:44.315282Z_1847afbc-f8a9-45da-94e8-5aef0738954e
```

- 保存したクラッシュの詳細を表示します。

構文

```
ceph crash info CRASH_ID
```

例

```
[ceph: root@host01 /]# ceph crash info 2021-05-24T19:58:42.549073Z_b2382865-ea89-4be2-b46f-9a59af7b7a2d
{
  "assert_condition": "session_map.sessions.empty()",
  "assert_file": "/builddir/build/BUILD/ceph-16.1.0-486-g324d7073/src/mon/Monitor.cc",
  "assert_func": "virtual Monitor::~Monitor()",
  "assert_line": 287,
  "assert_msg": "/builddir/build/BUILD/ceph-16.1.0-486-g324d7073/src/mon/Monitor.cc: In function 'virtual Monitor::~Monitor()' thread 7f67a1aeb700 time 2021-05-24T19:58:42.545485+0000\n/builddir/build/BUILD/ceph-16.1.0-486-g324d7073/src/mon/Monitor.cc: 287: FAILED\nceph_assert(session_map.sessions.empty())\n",
  "assert_thread_name": "ceph-mon",
  "backtrace": [
    "/lib64/libpthread.so.0(+0x12b30) [0x7f679678bb30]",
```



```

    "gsignal()",
    "abort()",
    "(ceph::__ceph_assert_fail(char const*, char const*, int, char const*)+0x1a9)
[0x7f6798c8d37b]",
    "/usr/lib64/ceph/libceph-common.so.2(+0x276544) [0x7f6798c8d544]",
    "(Monitor::~Monitor()+0xe30) [0x561152ed3c80]",
    "(Monitor::~Monitor()+0xd) [0x561152ed3cdd]",
    "main()",
    "__libc_start_main()",
    "_start()"
  ],
  "ceph_version": "16.1.0-486.el8cp",
  "crash_id": "2021-05-24T19:58:42.549073Z_b2382865-ea89-4be2-b46f-9a59af7b7a2d",
  "entity_name": "mon.ceph-adm4",
  "os_id": "rhel",
  "os_name": "Red Hat Enterprise Linux",
  "os_version": "8.3 (Ootpa)",
  "os_version_id": "8.3",
  "process_name": "ceph-mon",
  "stack_sig":
"957c21d558d0cba4cee9e8aaf9227b3b1b09738b8a4d2c9f4dc26d9233b0d511",
  "timestamp": "2021-05-24T19:58:42.549073Z",
  "utsname_hostname": "host02",
  "utsname_machine": "x86_64",
  "utsname_release": "4.18.0-240.15.1.el8_3.x86_64",
  "utsname_sysname": "Linux",
  "utsname_version": "#1 SMP Wed Feb 3 03:12:15 EST 2021"
}

```

8. **KEEP** の日数より古い保存済みクラッシュを削除します。ここで、**KEEP** は整数である必要があります。

構文

```
ceph crash prune KEEP
```

例

```
[ceph: root@host01 /]# ceph crash prune 60
```

9. **RECENT_CRASH** ヘルスチェックで考慮されなくなり、**crash ls-new** の出力に表示されないようにクラッシュレポートをアーカイブします。**crash ls** に表示されます。

構文

```
ceph crash archive CRASH_ID
```

例

```
[ceph: root@host01 /]# ceph crash archive 2021-05-24T19:58:42.549073Z_b2382865-ea89-4be2-b46f-9a59af7b7a2d
```

10. すべてのクラッシュレポートをアーカイブします。

例

```
[ceph: root@host01 /]# ceph crash archive-all
```

11. クラッシュダンプを削除します。

構文

```
ceph crash rm CRASH_ID
```

例

```
[ceph: root@host01 /]# ceph crash rm 2021-05-24T19:58:42.549073Z_b2382865-ea89-4be2-b46f-9a59af7b7a2d
```

関連情報

- Ceph の健全性メッセージの詳細は、『Red Hat Ceph Storage Troubleshooting Guide』の「[Health messages of a Ceph cluster](#)」セクションを参照してください。

第6章 CEPH ORCHESTRATOR を使用した OSD の管理

ストレージ管理者は、Ceph Orchestrators を使用して Red Hat Ceph Storage クラスターの OSD を管理できます。

6.1. CEPH OSD

Red Hat Ceph Storage クラスターが稼働している場合は、ランタイム時に OSD をストレージクラスターに追加できます。

Ceph OSD は、通常1つのストレージドライブおよびノード内の関連付けられたジャーナル用に1つの **ceph-osd** デーモンで構成されます。ノードに複数のストレージドライブがある場合は、ドライブごとに1つの **ceph-osd** デーモンをマッピングします。

Red Hat は、クラスターの容量を定期的に確認して、ストレージ容量の最後に到達するかどうかを確認することを推奨します。ストレージクラスターが **ほぼ完全** の比率に達すると、1つ以上の OSD を追加してストレージクラスターの容量を拡張します。

Red Hat Ceph Storage クラスターのサイズを縮小したり、ハードウェアを置き換える場合は、ランタイム時に OSD を削除することも可能です。ノードに複数のストレージドライブがある場合には、そのドライブ用に **ceph-osd** デーモンのいずれかを削除する必要もあります。通常、ストレージクラスターの容量を確認して、容量の上限に達したかどうかを確認することが推奨されます。ストレージクラスターが **ほぼ完全** の比率ではないことを OSD を削除する場合。



重要

OSD を追加する前に、ストレージクラスターが **完全な** 比率を超えないようにします。ストレージクラスターが **ほぼ完全な** 比率に達した後に OSD の障害が発生すると、ストレージクラスターが **完全な** 比率を超過する可能性があります。Ceph は、ストレージ容量の問題を解決するまでデータを保護するための書き込みアクセスをブロックします。完全な比率の影響を考慮せずに OSD を削除しないでください。

6.2. CEPH OSD ノードの設定

OSD を使用するプールのストレージストラテジーとして同様に Ceph OSD とサポートするハードウェアを設定します。Ceph は、一貫性のあるパフォーマンスプロファイルを確保するために、プール全体でハードウェアを統一します。最適なパフォーマンスを得るには、同じタイプまたはサイズのドライブのある CRUSH 階層を検討してください。

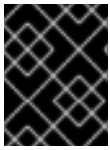
異なるサイズのドライブを追加する場合は、それに応じて重量を調整してください。OSD を CRUSH マップに追加する場合は、新規 OSD の重みを考慮してください。ハードドライブの容量は、1年あたり約 40% 増加するため、新しい OSD ノードはストレージクラスターの古いノードよりも大きなハードドライブを持つ可能性があります。つまり、重みが大きくなる可能性があります。

新たにインストールを行う前に、『[Installation Guide](#)』の「Requirements for Installing Red Hat Ceph Storage」の章を確認します。

6.3. OSD メモリーの自動チューニング

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

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



重要

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

構文

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

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

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

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

構文

$$\text{osd_memory_target} = \frac{\text{TOTAL_RAM_OF_THE_OSD} * (1048576) * (\text{autotune_memory_target_ratio})}{\text{NUMBER_OF_OSDS_IN_THE_OSD_NODE} - (\text{SPACE_ALLOCATED_FOR_OTHER_DAEMONS})}$$

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

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

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

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



注記

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

例

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

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

例

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

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

構文

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

例

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



注記

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

構文

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

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

例

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

6.4. CEPH OSD デプロイメント用のデバイスの一覧表示

Ceph Orchestrator を使用して OSD をデプロイする前に、利用可能なデバイスの一覧を確認することができます。コマンドは、Cephadm によって検出可能なデバイスの一覧を出力するために使用されます。以下の条件がすべて満たされると、ストレージデバイスが利用可能であるとみなされます。

- デバイスにはパーティションがない。
- デバイスは LVM 状態でない。
- デバイスをマウントしてはいけない。
- デバイスにはファイルシステムを含めることはできない。
- デバイスには Ceph BlueStore OSD を含めることはできない。
- デバイスは 5 GB を超える必要がある。



注記

Ceph は、利用できないデバイスに OSD をプロビジョニングしません。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- ホストがクラスタに追加されている。
- すべてのマネージャーおよびモニターデーモンがデプロイされている。

手順

1. Cephadm シェルにログインします。

例

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

2. OSD をデプロイするために利用可能なデバイスを一覧表示します。

構文

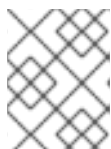
```
ceph orch device ls [--hostname=HOSTNAME_1 HOSTNAME_2] [--wide] [--refresh]
```

例

```
[ceph: root@host01 /]# ceph orch device ls --wide --refresh
```

--wide オプションを使用すると、デバイスが OSD として使用できない可能性がある理由など、デバイスに関連するすべての詳細が提供されます。このオプションは、NVMe デバイスをサポートしません。

3. 任意手順: **ceph orch device ls** の出力で **Health**、**Ident** および **Fault** フィールドを有効にするには、以下のコマンドを実行します。



注記

これらのフィールドは **libstoragegmt** ライブラリーによってサポートされ、現時点では SCSI、SAS、SATA デバイスをサポートします。

- a. root ユーザーとして、ハードウェアと **libstoragegmt** ライブラリーとの互換性を確認して、計画外のサービスの中断が発生しないようにします。

例

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

この出力で、それぞれの SCSI VPD 0x83 ID で **Health Status** が **Good** と表示されます。



注記

この情報を取得しないと、フィールドを有効にした場合にデバイスで異常な挙動が発生する可能性があります。

- b. **libstoragegmt** サポートを有効にします。

例

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

これを有効にし、**ceph orch device ls** を実行すると、**Health** フィールドの出力が **Good** になります。

検証

- デバイスを一覧表示します。

例

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

6.5. CEPH OSD デプロイメントのデバイスの消去

OSD をデプロイする前に、利用可能なデバイスの一覧を確認する必要があります。デバイスに空き容量がない場合は、そのデバイスを消去してデバイス上のデータを消去します。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- ホストがクラスタに追加されている。
- すべてのマネージャーおよびモニターデーモンがデプロイされている。

手順

1. Cephadm シェルにログインします。

例

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

- OSD をデプロイするために利用可能なデバイスを一覧表示します。

構文

```
ceph orch device ls [--hostname=HOSTNAME_1 HOSTNAME_2] [--wide] [--refresh]
```

例

```
[ceph: root@host01 /]# ceph orch device ls --wide --refresh
```

- デバイスのデータをクリアします。

構文

```
ceph orch device zap HOSTNAME FILE_PATH --force
```

例

```
[ceph: root@host01 /]# ceph orch device zap host02 /dev/sdb --force
```

検証

- デバイスに容量があることを確認します。

例

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

Available の下のフィールドが **Yes** であることを確認できます。

関連情報

- 詳細は、『Red Hat Ceph Storage Operations Guide』の「[Listing devices for Ceph OSD deployment](#)」セクションを参照してください。

6.6. すべての利用可能なデバイスへの CEPH OSD のデプロイ

すべての OSDs を利用可能なすべてのデバイスにデプロイできます。Cephadm により、Ceph Orchestrator は利用可能な未使用のストレージデバイスで OSD を検出およびデプロイできます。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- ホストがクラスタに追加されている。
- すべてのマネージャーおよびモニターデーモンがデプロイされている。

手順

1. Cephadm シェルにログインします。

例

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

2. OSD をデプロイするために利用可能なデバイスを一覧表示します。

構文

```
ceph orch device ls [--hostname=HOSTNAME_1 HOSTNAME_2] [--wide] [--refresh]
```

例

```
[ceph: root@host01 /]# ceph orch device ls --wide --refresh
```

3. すべての利用可能なデバイスに OSD をデプロイします。

例

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

ceph orch apply の影響は永続的であり、Orchestrator は自動的にデバイスを見つけ、それをクラスターに追加し、新しい OSD を作成します。これは、以下の条件下で実行されます。

- 新しいディスクまたはドライブがシステムに追加される。
- 既存のディスクまたはドライブは消去される。
- OSD は削除され、デバイスは消去される。
--unmanaged パラメーターを使用して、利用可能なすべてのデバイスで OSD の自動作成を無効にできます。

例

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

パラメーター **--unmanaged** を **true** に設定すると、OSD の作成が無効になり、新しい OSD サービスを適用しても変更はありません。



注記

コマンド **ceph orch daemon add** は、新しい OSD を作成しますが、OSD サービスを追加しません。

検証

- サービスを一覧表示します。

例

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

- ノードとデバイスの詳細を表示します。

例

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

関連情報

- 『Red Hat Ceph Storage Operations Guide』の「[Listing devices for Ceph OSD deployment](#)」セクションを参照してください。

6.7. 特定のデバイスおよびホストへの CEPH OSD のデプロイ

Ceph Orchestrator を使用して、特定のデバイスおよびホストにすべての Ceph OSD をデプロイできます。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- ホストがクラスタに追加されている。
- すべてのマネージャーおよびモニターデーモンがデプロイされている。

手順

1. Cephadm シェルにログインします。

例

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

2. OSD をデプロイするために利用可能なデバイスを一覧表示します。

構文

```
ceph orch device ls [--hostname=HOSTNAME_1 HOSTNAME_2] [--wide] [--refresh]
```

例

```
[ceph: root@host01 /]# ceph orch device ls --wide --refresh
```

3. 特定のデバイスおよびホストに OSD をデプロイします。

構文

```
ceph orch daemon add osd HOSTNAME:_DEVICE_PATH_
```

例

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

検証

- サービスを一覧表示します。

例

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

- ノードとデバイスの詳細を表示します。

例

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

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

```
ceph orch ps --service_name=SERVICE_NAME
```

例

```
[ceph: root@host01 /]# ceph orch ps --service_name=osd
```

関連情報

- 『Red Hat Ceph Storage Operations Guide』の「[Listing devices for Ceph OSD deployment](#)」セクションを参照してください。

6.8. OSD をデプロイするための高度なサービス指定およびフィルター

タイプ OSD のサービス指定は、ディスクのプロパティを使用してクラスターレイアウトを記述する 1 つの方法です。これは、デバイスの名前やパスの指定を把握せずに、必要な設定でどのディスクを OSD にするか Ceph に知らせる抽象的な方法をユーザーに提供します。各デバイスおよび各ホストに対して、**yaml** ファイルまたは **json** ファイルを定義します。

OSD 仕様の一般設定

- **service_type**: 'osd': これは、OSDS の作成には必須です。
- **service_id**: 希望するサービス名または ID を使用します。OSD のセットは、指定ファイルを使用して作成されます。この名前は、すべての OSD を管理し、Orchestrator サービスを表すために使用されます。
- **placement**: これは、OSD をデプロイする必要があるホストを定義するために使用されます。以下のオプションで使用できます。
 - **host_pattern**: '*' - ホストの選択に使用するホスト名パターン。
 - **Label**: 'osd_host' - OSD をデプロイする必要があるホストで使用されるラベル。

- **hosts:** 'host01', 'host02': OSD をデプロイする必要があるホスト名の明示的なリスト。
- **selection of devices:** OSD が作成されるデバイス。これにより、OSD が異なるデバイスから分離されます。3つのコンポーネントを持つ BlueStore OSD のみを作成できます。
 - OSD データ: すべての OSD データが含まれます。
 - WAL: BlueStore 内部ジャーナルまたはログ先行書き込み
 - DB: BlueStore 内部メタデータ
- **data_devices:** OSD をデプロイするデバイスを定義します。この場合、OSD は同じ場所に配置されるスキーマに作成されます。フィルターを使用して、デバイスおよびフォルダーを選択できます。
- **wal_devices:** WAL OSD に使用されるデバイスを定義します。フィルターを使用して、デバイスおよびフォルダーを選択できます。
- **db_devices:** DB OSD のデバイスを定義します。フィルターを使用して、デバイスおよびフォルダーを選択できます。
- **encrypted:** **True** または **False** のいずれかに設定される OSD の情報を暗号化するオプションのパラメーター
- **unmanaged:** オプションのパラメーターで、デフォルトでは **False** に設定されます。Orchestrator で OSD サービスを管理しない場合は、これを **True** に設定します。
- **block_wal_size:** ユーザー定義の値 (バイト単位)。
- **block_db_size:** ユーザー定義の値 (バイト単位)。
- **osds_per_device:** デバイスごとに複数の OSD をデプロイするためのユーザー定義値。

デバイスを指定するためのフィルター

フィルターは、**data_devices**、**wal_devices**、および **db_devices** パラメーターとともに使用されます。

フィルターの名前	詳細	構文	例
Model	ターゲット固有のディスク。 lsblk -o NAME,FSTYPE,LABEL,MOUNTPOINT,SIZE,MODEL コマンドまたは smartctl -i /DEVIVE_PATH を実行すると、モデルの詳細を取得できます。	Model: DISK_MODEL_NAME	Model: MC-55-44-XZ
Vendor	ターゲット固有のディスク	Vendor: DISK_VENDOR_NAME	Vendor: Vendor Cs
Size Specification	正確なサイズのディスク	size: EXACT	size: '10G'

Size Specification	範囲内にあるディスクサイズ	size: LOW:HIGH	size: '10G:40G'
Size Specification	サイズがこれ以下のディスク	size: :HIGH	size: ':10G'
Size Specification	サイズがこれ以上のディスク	size: LOW:	size: '40G:'
Rotational	ディスクの回転属性。1はローテーションされるすべてのディスクと一致し、0はローテーションされないすべてのディスクと一致します。 rotational=0の場合、OSDはSSDまたはNVMEで設定されます。rotational=1の場合、OSDはHDDで設定されます。	rotational: 0 or 1	rotational: 0
All	利用可能なディスクをすべて考慮します。	all: true	all: true
Limiter	有効なフィルターが指定されていても、一致するディスクの量を制限する場合は、'limit' ディレクティブを使用できます。これは、最後の手段としてのみ使用してください。	limit: NUMBER	limit: 2



注記

同じホストに同じ場所のないコンポーネントを持つ OSD を作成するには、使用される異なるタイプのデバイスを指定し、デバイスが同じホスト上になければなりません。



注記

OSD のデプロイに使用されるデバイスは、**libstoragegmt** によってサポートされる必要があります。

関連情報

- 『Red Hat Ceph Storage Operations Guide』の「[Deploying Ceph OSDs using the advanced specifications](#)」セクションを参照してください。
- **libstoragegmt** の詳細は、『Red Hat Ceph Storage Operations Guide』の「[Listing devices for Ceph OSD deployment](#)」セクションを参照してください。

6.9. 高度なサービス指定を使用した CEPH OSD のデプロイ

タイプ OSD のサービス指定は、ディスクのプロパティを使用してクラスターレイアウトを記述する 1 つの方法です。これは、デバイスの名前やパスの指定を把握せずに、必要な設定でどのディスクを OSD にするか Ceph に知らせる抽象的な方法をユーザーに提供します。

各デバイスおよび各ホストに OSD をデプロイするには、**yml** ファイルまたは **json** ファイルを定義します。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- ホストがクラスターに追加されている。
- すべてのマネージャーおよびモニターデーモンがデプロイされている。

手順

1. Cephadm シェルにログインします。

例

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

2. モニターノードで、以下のディレクトリーに移動します。

構文

```
cd /var/lib/ceph/DAEMON_PATH/
```

例

```
[ceph: root@host01 /]# cd /var/lib/ceph/osd/
```

3. **osd_spec.yml** ファイルを作成します。

例

```
[ceph: root@host01 osd]# touch osd_spec.yml
```

4. **osd_spec.yml** ファイルを編集し、以下の詳細が含まれるようにします。

構文

```
service_type: osd
service_id: SERVICE_ID
placement:
  host_pattern: '*' # optional
data_devices: # optional
  model: DISK_MODEL_NAME # optional
paths:
  - /DEVICE_PATH
```

```
osds_per_device: NUMBER_OF_DEVICES # optional
db_devices: # optional
  size: # optional
  all: true # optional
paths:
  - /DEVICE_PATH
encrypted: true
```

- a. 単純なシナリオ: この場合、すべてのノードが同じ設定になります。

例

```
service_type: osd
service_id: osd_spec_default
placement:
  host_pattern: '*'
data_devices:
  all: true
paths:
  - /dev/sdb
encrypted: true
```

例

```
service_type: osd
service_id: osd_spec_default
placement:
  host_pattern: '*'
data_devices:
  size: '80G'
db_devices:
  size: '40G:'
paths:
  - /dev/sdc
```

- b. 高度なシナリオ: これにより、すべての HDD を専用の DB または WAL デバイスとして割り当てられた 2 つの SSD のある **data_devices** として使用し、目的のレイアウトを作成します。残りの SSD は、NVME ベンダーが専用の DB または WAL デバイスとして割り当てられている **data_devices** です。

例

```
service_type: osd
service_id: osd_spec_hdd
placement:
  host_pattern: '*'
data_devices:
  rotational: 0
db_devices:
  model: Model-name
  limit: 2
---
service_type: osd
service_id: osd_spec_ssd
```

```
placement:
  host_pattern: '*'
data_devices:
  model: Model-name
db_devices:
  vendor: Vendor-name
```

- c. 不均一のノードを使用する高度なシナリオ: これは、host_pattern キーに応じて、異なる OSD 仕様を異なるホストに適用します。

例

```
service_type: osd
service_id: osd_spec_node_one_to_five
placement:
  host_pattern: 'node[1-5]'
data_devices:
  rotational: 1
db_devices:
  rotational: 0
---
service_type: osd
service_id: osd_spec_six_to_ten
placement:
  host_pattern: 'node[6-10]'
data_devices:
  model: Model-name
db_devices:
  model: Model-name
```

- d. 専用の WAL および DB デバイスを使用した高度なシナリオ:

例

```
service_type: osd
service_id: osd_using_paths
placement:
  hosts:
    - host01
    - host02
data_devices:
  paths:
    - /dev/sdb
db_devices:
  paths:
    - /dev/sdc
wal_devices:
  paths:
    - /dev/sdd
```

- e. デバイスごとに複数の OSD がある高度なシナリオ:

例


```

service_type: osd
service_id: multiple_osds
placement:
  hosts:
    - host01
    - host02
osds_per_device: 4
data_devices:
  paths:
    - /dev/sdb

```

- OSD をデプロイする前にドライランを実行します。



注記

この手順は、デーモンをデプロイしなくても、デプロイメントのプレビューを提供します。

例

```
[ceph: root@host01 osd]# ceph orch apply -i osd_spec.yml --dry-run
```

- サービス指定を使用して OSD をデプロイします。

構文

```
ceph orch apply -i FILE_NAME.yml
```

例

```
[ceph: root@host01 osd]# ceph orch apply -i osd_spec.yml
```

検証

- サービスを一覧表示します。

例

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

- ノードとデバイスの詳細を表示します。

例

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

関連情報

- 『Red Hat Ceph Storage Operations Guide』の「[Advanced service specifications and filters for deploying OSDs](#)」セクションを参照してください。

6.10. CEPH ORCHESTRATOR を使用した OSD デーモンの削除

Cephadm を使用して、OSD をクラスターから削除できます。

クラスターから OSD を削除するには、2つの手順を実行します。

1. クラスターからすべての配置グループ (PG) を退避します。
2. PG のない OSD をクラスターから削除します。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- ホストがクラスターに追加されている。
- Ceph Monitor、Ceph Manager、および Ceph OSD デーモンがストレージクラスターにデプロイされます。

手順

1. Cephadm シェルにログインします。

例

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

2. OSD を削除する必要があるデバイスとノードを確認します。

例

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

3. OSD を削除します。

構文

```
ceph orch osd rm OSD_ID [--replace] [--force]
```

例

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



注記

--replace などのオプションを指定せずにストレージクラスターから OSD を削除すると、デバイスはストレージクラスターから完全に削除されます。OSD のデプロイに同じデバイスを使用する必要がある場合は、ストレージクラスターにデバイスを追加する前に、最初にデバイスをザッピングする必要があります。

4. オプション: 特定のノードから複数の OSD を削除するには、次のコマンドを実行します。

構文

```
ceph orch osd rm OSD_ID OSD_ID
```

例

```
[ceph: root@host01 /]# ceph orch osd rm 2 5
```

- OSD の削除のステータスを確認します。

例

```
[ceph: root@host01 /]# ceph orch osd rm status
OSD_ID HOST STATE PG_COUNT REPLACE FORCE DRAIN_STARTED_AT
2 host01 draining 124 False False 2021-09-07 16:26:07.142980
5 host01 draining 107 False False 2021-09-07 16:26:08.330371
```

OSD に PG が残っていない場合は廃止され、クラスターから削除されます。

検証

- Ceph OSD が削除されたデバイスおよびノードの詳細を確認します。

例

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

関連情報

- 詳細は、『Red Hat Ceph Storage Operations Guide』の「[Deploying Ceph OSDs on all available devices](#)」セクションを参照してください。
- 詳細は、『Red Hat Ceph Storage Operations Guide』の「[Deploying Ceph OSDs on specific devices and hosts](#)」セクションを参照してください。
- デバイスのスペースをクリアする方法の詳細については、Red Hat Ceph Storage Operations Guide の [Zapping devices for Ceph OSD deployment](#) セクションを参照してください。

6.11. CEPH ORCHESTRATOR を使用した OSD の置き換え

ディスクに障害が発生した場合は、物理ストレージデバイスを交換し、同じ OSD ID を再利用することで、CRUSH マップを再設定する必要がなくなります。

`ceph orch rm` コマンドを使用して OSD ID を保持することで、クラスターから OSD を置き換えることができます。OSD は CRUSH 階層から永続的に削除されませんが、**destroyed** フラグが割り当てられます。このフラグは、次の OSD デプロイメントで再利用できる OSD ID を判断するために使用されます。**destroyed** フラグは、次の OSD デプロイメントで再利用される OSD ID を判断するために使用されます。

デプロイメントに OSD 仕様を使用する場合、新たに追加されたディスクには置き換えられたディスクの OSD ID が割り当てられます。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。

- ホストがクラスターに追加されている。
- Monitor、Manager、および OSD デーモンはストレージクラスターにデプロイされます。

手順

1. Cephadm シェルにログインします。

例

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

2. OSD を置き換える必要のあるデバイスとノードを確認します。

例

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

3. OSD を置き換えます。

構文

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

例

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

4. OSD 置き換えのステータスを確認します。

例

```
[ceph: root@host01 /]# ceph orch osd rm status
```

検証

- Ceph OSD が置き換えられるデバイスとノードの詳細を確認します。

例

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

同じホストで実行している置き換えられた ID と同じ ID の OSD が表示されます。

関連情報

- 詳細は、『Red Hat Ceph Storage Operations Guide』の「[Deploying Ceph OSDs on all available devices](#)」セクションを参照してください。
- 詳細は、『Red Hat Ceph Storage Operations Guide』の「[Deploying Ceph OSDs on specific devices and hosts](#)」セクションを参照してください。

6.12. CEPH ORCHESTRATOR を使用した OSD の削除停止

削除用にキューに置かれた OSD のみの削除を停止することができます。これにより、OSD の初期状態がリセットされ、削除キューから削除されます。

OSD の削除処理中である場合は、プロセスを停止することはできません。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- ホストがクラスターに追加されている。
- Monitor、Manager デーモン、および OSD デーモンがクラスターにデプロイされている。
- 開始した OSD プロセスを削除する。

手順

1. Cephadm シェルにログインします。

例

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

2. 削除するために開始された OSD のデバイスとノードを確認します。

例

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

3. キューに置かれた OSD の削除を停止します。

構文

```
ceph orch osd rm stop OSD_ID
```

例

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

4. OSD の削除のステータスを確認します。

例

```
[ceph: root@host01 /]# ceph orch osd rm status
```

検証

- Ceph OSD を削除するためにキューに入れられたデバイスおよびノードの詳細を確認します。

例

■

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

関連情報

- 詳細は、『Red Hat Ceph Storage Operations Guide』の「[Removing the OSD daemons using the Ceph Orchestrator](#)」セクションを参照してください。

6.13. CEPH ORCHESTRATOR を使用した OSD のアクティブ化

ホストのオペレーティングシステムが再インストールされた場合に、クラスターで OSD をアクティブにすることができます。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- ホストがクラスターに追加されている。
- Monitor、Manager、および OSD デーモンがストレージクラスターにデプロイされている。

手順

1. Cephadm シェルにログインします。

例

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

2. ホストのオペレーティングシステムを再インストールしたら、OSD をアクティベートします。

構文

```
ceph cephadm osd activate HOSTNAME
```

例

```
[ceph: root@host01 /]# ceph cephadm osd activate host03
```

検証

- サービスを一覧表示します。

例

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

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

```
ceph orch ps --service_name=SERVICE_NAME
```

例

```
[ceph: root@host01 /]# ceph orch ps --service_name=osd
```

6.13.1. データ移行の監視

OSD を CRUSH マップに追加または削除すると、Ceph は配置グループを新規または既存の OSD に移行してデータのリバランスを開始します。**ceph-w** コマンドを使用して、データの移行を確認できます。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- 最近 OSD を追加または削除した。

手順

1. データの移行を確認するには、以下を実行します。

例

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

2. 配置グループのステータスが **active+clean** から **active, some degraded objects** し、最後に移行の完了時に **active+clean** に変わるのを確認します。
3. ユーティリティを終了するには、**Ctrl + C** を押します。

6.14. 配置グループの再計算

配置グループ (PG) は、利用可能な OSD にまたがるプールデータの分散を定義します。配置グループは、使用する冗長性アルゴリズムに基づいて構築されます。3 方向のレプリケーションでは、冗長性は 3 つの異なる OSD を使用するように設定されます。イレイジャーコーディングされたプールの場合、使用する OSD の数はチャンクの数で定義されます。

プールを定義する場合、配置グループの数によって、使用可能なすべての OSD にデータが分散される粒度のグレードが定義されます。数値が大きいほど、容量負荷の均等化が向上します。ただし、データを再構築する場合は配置グループの処理も重要であるため、事前に慎重に選択する必要があります。計算をサポートするには、アジャイル環境の生成に利用できるツールを使用できます。

ストレージクラスタの有効期間中、プールが最初に予想される制限を上回る可能性があります。ドライブの数が増えると、再計算が推奨されます。OSD ごとの配置グループの数は約 100 である必要があります。ストレージクラスタに OSD を追加すると、OSD あたりの PG の数は時間の経過とともに減少します。ストレージクラスタで最初に 120 ドライブを使用し、プールの **pg_num** を 4000 に設定すると、レプリケーション係数が 3 の場合に、OSD ごとに 100 PG に設定されます。時間の経過とともに、OSD の数が 10 倍になると、OSD あたりの PG の数は 10 になります。OSD ごとの PG の数が少ないと、容量が不均一に分散される傾向があるため、プールごとの PG を調整することを検討してください。

配置グループ数の調整は、オンラインで実行できます。再計算は PG 番号の再計算だけでなく、データの再配置が関係し、長いプロセスになります。ただし、データの可用性はいつでも維持されます。

OSD ごとの非常に高い PG は、失敗した OSD 上でのすべての PG を再構築すると同時に起動されるた

め、回避する必要があります。時間内に再構築を実行するには、多くの IOPS が必要ですが、これは利用できない可能性があります。これにより、I/O キューが深くなり、待ち時間が長くなり、ストレージクラスターが使用できなくなったり、修復時間が長くなったりします。

関連情報

- 特定のユースケースで値を計算する場合は、[PG calculator](#) を参照してください。
- 詳細は、『Red Hat Ceph Storage Strategies Guide』の「[Erasure Code Pools](#)」の章を参照してください。

第7章 CEPH ORCHESTRATOR を使用したモニタリングスタックの管理

ストレージ管理者は、バックエンドにて Cephadm と Ceph Orchestrator を使用して、モニタリングおよびアラートスタックをデプロイできます。モニタリングスタックは、Prometheus、Prometheus エクスポート、Prometheus Alertmanager、および Grafana で構成されます。ユーザーは、YAML 設定ファイルの Cephadm でこれらのサービスを定義するか、またはコマンドラインインターフェースを使用してそれらをデプロイする必要があります。同じタイプの複数のサービスがデプロイされると、可用性の高い設定がデプロイされます。ノードエクスポートはこのルールの例外です。



注記

Red Hat Ceph Storage 5.0 は、Prometheus、Grafana、Alertmanager、および node-exporter などのモニタリングサービスをデプロイするためのカスタムイメージをサポートしません。

以下のモニタリングサービスは、Cephadm でデプロイすることができます。

- Prometheus はモニタリングおよびアラートツールキットです。これは、Prometheus エクスポートによって提供されるデータを収集し、事前に定義されたしきい値に達した場合に事前設定されたアラートを実行します。Prometheus マネージャーモジュールは、**ceph-mgr** のコレクションポイントから Ceph パフォーマンスカウンターに渡す Prometheus エクスポートを提供します。

デーモンを提供するメトリクスなど、スクレーパーターゲットなどの Prometheus 設定は、Cephadm によって自動的に設定されます。Cephadm は、デフォルトのアラートの一覧 (例: health error、10% OSDs down、pgs inactive) もデプロイします。
- Alertmanager は、Prometheus サーバーによって送信されるアラートを処理します。これは、正しい受信側にアラートを複製解除、グループ化、およびルーティングします。デフォルトでは、Ceph ダッシュボードは受信側として自動的に設定されます。Alertmanager は、Prometheus サーバーによって送信されるアラートを処理します。アラートは Alertmanager を使用して無音にすることができますが、無音は Ceph Dashboard を使用して管理することもできます。
- Grafana は可視化およびアラートソフトウェアです。Grafana のアラート機能は、このモニタリングスタックによって使用されません。アラートについては、Alertmanager が使用されます。

デフォルトでは、Grafana へのトラフィックは TLS で暗号化されます。独自の TLS 証明書を指定するか、自己署名の証明書を使用できます。Grafana をデプロイする前にカスタム証明書が設定されていない場合、自己署名証明書が自動的に作成され、Grafana に設定されます。Grafana のカスタム証明書は、以下のコマンドを使用して設定できます。

構文

```
ceph config-key set mgr/cephadm/grafana_key -i
PRESENT_WORKING_DIRECTORY/key.pem
ceph config-key set mgr/cephadm/grafana_cert -i
PRESENT_WORKING_DIRECTORY/certificate.pem
```

ノードエクスポートは Prometheus のエクスポートで、インストールされているノードに関するデータを提供します。ノードエクスポートをすべてのノードにインストールすることが推奨されます。これは、node-exporter サービスタイプの monitoring.yml ファイルを使用して実行できます。

7.1. CEPH ORCHESTRATOR を使用したモニタリングスタックのデプロイ

モニタリングスタックは、Prometheus、Prometheus エクスポーター、Prometheus Alertmanager、および Grafana で構成されます。Ceph Dashboard はこれらのコンポーネントを使用して、クラスターの使用状況およびパフォーマンスの詳細なメトリクスを保存し、可視化します。

YAML ファイル形式のサービス指定を使用して、モニタリングスタックをデプロイできます。すべてのモニタリングサービスのバインド先のネットワークおよびポートは **yml** ファイルで設定できます。

前提条件

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

手順

1. Ceph Manager デーモンで prometheus モジュールを有効にします。これにより、内部 Ceph メトリクスが公開され、Prometheus がそれらを読み取りできます。

例

```
[ceph: root@host01 /]# ceph mgr module enable prometheus
```



重要

このコマンドは、Prometheus のデプロイ前に実行されるようにします。デプロイメントの前にコマンドが実行されなかった場合は、Prometheus を再デプロイして設定を更新する必要があります。

```
ceph orch redeploy prometheus
```

2. 以下のディレクトリーに移動します。

構文

```
cd /var/lib/ceph/DAEMON_PATH/
```

例

```
[ceph: root@host01 mds/]# cd /var/lib/ceph/monitoring/
```



注記

ディレクトリー **monitoring** が存在しない場合は、作成します。

3. **monitoring.yml** ファイルを作成します。

例

```
[ceph: root@host01 monitoring]# touch monitoring.yml
```

- 以下の例のような内容で指定ファイルを編集します。

例

```
service_type: prometheus
service_name: prometheus
placement:
  hosts:
  - host01
networks:
- 192.169.142.0/24
---
service_type: node-exporter
---
service_type: alertmanager
service_name: alertmanager
placement:
  hosts:
  - host03
networks:
- 192.169.142.03/24
---
service_type: grafana
service_name: grafana
placement:
  hosts:
  - host02
networks:
- 192.169.142.02/24
```

- モニタリングサービスを適用します。

例

```
[ceph: root@host01 monitoring]# ceph orch apply -i monitoring.yml
```

検証

- サービスを一覧表示します。

例

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

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

```
ceph orch ps --service_name=SERVICE_NAME
```

例

```
[ceph: root@host01 /]# ceph orch ps --service_name=prometheus
```



重要

Prometheus、Grafana、および Ceph ダッシュボードはすべて相互に対話するように自動設定されるため、Ceph ダッシュボードで Grafana 統合が完全に機能します。

7.2. CEPH ORCHESTRATOR を使用したモニタリングスタックの削除

`ceph orch rm` コマンドを使用してモニタリングスタックを削除できます。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。

手順

- Cephadm シェルにログインします。

例

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

- `ceph orch rm` コマンドを使用してモニタリングスタックを削除します。

構文

```
ceph orch rm SERVICE_NAME --force
```

例

```
[ceph: root@host01 /]# ceph orch rm grafana
[ceph: root@host01 /]# ceph orch rm prometheus
[ceph: root@host01 /]# ceph orch rm node-exporter
[ceph: root@host01 /]# ceph orch rm alertmanager
[ceph: root@host01 /]# ceph mgr module disable prometheus
```

- プロセスのステータスを確認します。

例

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

検証

- サービスを一覧表示します。

例

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

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

```
ceph orch ps
```

例

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

関連情報

- 詳細は、『Red Hat Ceph Storage Operations Guide』の「[Deploying the monitoring stack using the Ceph Orchestrator](#)」セクションを参照してください。

第8章 基本的な RED HAT CEPH STORAGE のクライアント設定

ストレージ管理者は、ストレージクラスターと対話するために基本設定でクライアントマシンを設定する必要があります。ほとんどのクライアントマシンには **ceph-common package** とその依存関係のみがインストールされている必要があります。これは、基本的な **ceph** コマンドおよび **rados** コマンドと、**mount.ceph** や **rbd** などのその他のコマンドを提供します。

8.1. クライアントマシンでのファイルの設定

通常、クライアントマシンには完全なストレージクラスターメンバーよりも小さな設定ファイルが必要になります。Ceph モニターに到達するためにクライアントに情報を提供する最小限の設定ファイルを生成できます。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- ノードへの root アクセス。

手順

1. ファイルを設定するノードで、**/etc** フォルダーにディレクトリー **ceph** を作成します。

例

```
[root@host01 ~]# mkdir /etc/ceph/
```

2. **/etc/ceph** ディレクトリーに移動します。

例

```
[root@host01 ~]# cd /etc/ceph/
```

3. **ceph** ディレクトリーで設定ファイルを生成します。

例

```
[root@host01 ceph]# ceph config generate-minimal-conf

# minimal ceph.conf for 417b1d7a-a0e6-11eb-b940-001a4a000740
[global]
fsid = 417b1d7a-a0e6-11eb-b940-001a4a000740
mon_host = [v2:10.74.249.41:3300/0,v1:10.74.249.41:6789/0]
```

このファイルのコンテンツは、**/etc/ceph/ceph.conf** パスにインストールする必要があります。この設定ファイルを使用して、Ceph モニターに到達できます。

8.2. クライアントマシンでのキーリングの設定

ほとんどの Ceph クラスターは認証が有効な状態で実行され、クライアントはクラスターマシンと通信するために鍵が必要です。Ceph モニターに到達するためにクライアントに詳細を提供できるキーリングを生成できます。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- ノードへの root アクセス。

手順

1. キーリングを設定するノードで、**/etc** フォルダにディレクトリー **ceph** を作成します。

例

```
[root@host01 ~]# mkdir /etc/ceph/
```

2. **ceph** ディレクトリーの **/etc/ceph** ディレクトリーに移動します。

例

```
[root@host01 ~]# cd /etc/ceph/
```

3. クライアントのキーリングを生成します。

構文

```
ceph auth get-or-create client.CLIENT_NAME -o /etc/ceph/NAME_OF_THE_FILE
```

例

```
[root@host01 ceph]# ceph auth get-or-create client.fs -o /etc/ceph/ceph.keyring
```

4. **ceph.keyring** ファイルの出力を確認します。

例

```
[root@host01 ceph]# cat ceph.keyring
```

```
[client.fs]
key = AQAvoH5gkUCsExAATz3xCBLd4n6B6jRv+Z7CVQ==
```

出力は **/etc/ceph/ceph.keyring** などのキーリングファイルに配置する必要があります。

第9章 CEPH ORCHESTRATOR を使用した MDS サービスの管理

ストレージ管理者は、バックエンドにて Cephadm と Ceph Orchestrator を使用して MDS サービスをデプロイできます。デフォルトでは、Ceph File System (CephFS) はアクティブな MDS デーモンを1つだけ使用します。ただし、多くのクライアントがあるシステムでは複数のアクティブな MDS デーモンを使用する利点があります。

本セクションでは、以下の管理タスクを説明します。

- [コマンドラインインターフェースを使用した MDS サービスのデプロイ。](#)
- [サービス指定を使用した MDS サービスのデプロイ。](#)
- [Ceph Orchestrator を使用した MDS サービスの削除。](#)

9.1. 前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- すべてのノードへの root レベルのアクセス。
- ホストがクラスタに追加されている。
- すべてのマネージャー、モニター、および OSD デーモンがデプロイされている。

9.2. コマンドラインインターフェースを使用した MDS サービスのデプロイ

Ceph Orchestrator を使用すると、コマンドラインインターフェースで **placement** 仕様を使用して、Metadata Server (MDS) サービスをデプロイできます。Ceph ファイルシステム (CephFS) には、1つ以上の MDS が必要です。



注記

最低でも、Ceph ファイルシステム (CephFS) データ用のプール1つと CephFS メタデータ用のプール1つの2つのプールがあるようにしてください。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- ホストがクラスタに追加されている。
- すべてのマネージャー、モニター、および OSD デーモンがデプロイされます。

手順

1. Cephadm シェルにログインします。

例

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

2. 配置指定を使用して MDS デーモンをデプロイする方法は2つあります。

方法 1

- **ceph fs volume** を使用して MDS デーモンを作成します。これにより、CephFS ボリューム、CephFS に関連付けられたプールが作成され、ホスト上で MDS サービスも起動します。

構文

```
ceph fs volume create FILESYSTEM_NAME --placement="NUMBER_OF_DAEMONS
HOST_NAME_1 HOST_NAME_2 HOST_NAME_3"
```



注記

デフォルトでは、このコマンドに対してレプリケートされたプールが作成されま
す。

例

```
[ceph: root@host01 /]# ceph fs volume create test --placement="2 host01 host02"
```

方法 2

- プール、CephFS を作成してから、配置指定を使用して MDS サービスをデプロイします。
 - a. CephFS のプールを作成します。

構文

```
ceph osd pool create DATA_POOL
ceph osd pool create METADATA_POOL
```

例

```
[ceph: root@host01 /]# ceph osd pool create cephfs_data
[ceph: root@host01 /]# ceph osd pool create cephfs_metadata
```

- b. データプールおよびメタデータプールのファイルシステムを作成します。

構文

```
ceph fs new FILESYSTEM_NAME METADATA_POOL DATA_POOL
```

例

```
[ceph: root@host01 /]# ceph fs new test cephfs_metadata cephfs_data
```

- c. **ceph orch apply** コマンドを使用して MDS サービスをデプロイします。

構文

```
ceph orch apply mds FILESYSTEM_NAME --placement="NUMBER_OF_DAEMONS
HOST_NAME_1 HOST_NAME_2 HOST_NAME_3"
```

例

```
[ceph: root@host01 /]# ceph orch apply mds test --placement="2 host01 host02"
```

検証

- サービスを一覧表示します。

例

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

- CephFS のステータスを確認します。

例

```
[ceph: root@host01 /]# ceph fs ls  
[ceph: root@host01 /]# ceph fs status
```

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

```
ceph orch ps --daemon_type=DAEMON_NAME
```

例

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=mds
```

関連情報

- Ceph ファイルシステム (CephFS) の作成に関する詳細は、[Red Hat Ceph Storage File System Guide](#) を参照してください。

9.3. サービス指定を使用した MDS サービスのデプロイ

Ceph Orchestrator を使用すると、サービス指定を使用して MDS サービスをデプロイできます。



注記

少なくとも 2 つのプールがあることを確認してください。1 つは Ceph ファイルシステム (CephFS) データ用で、もう 1 つは CephFS メタデータ用です。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- ホストがクラスタに追加されている。
- すべてのマネージャー、モニター、および OSD デーモンがデプロイされます。

手順

1. Cephadm シェルにログインします。

例

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

2. 以下のディレクトリーに移動します。

構文

```
cd /var/lib/ceph/DAEMON_PATH/
```

例

```
[ceph: root@host01 /]# cd /var/lib/ceph/mds/
```



注記

ディレクトリー **mds** が存在しない場合は、作成します。

3. **mds.yml** ファイルを作成します。

例

```
[ceph: root@host01 mds]# touch mds.yml
```

4. **mds.yml** ファイルを編集し、以下の詳細が含まれるようにします。

構文

```
service_type: mds
service_id: FILESYSTEM_NAME
placement:
  hosts:
    - HOST_NAME_1
    - HOST_NAME_2
    - HOST_NAME_3
```

例

```
service_type: mds
service_id: fs_name
placement:
  hosts:
    - host01
    - host02
```

5. サービス指定を使用して MDS サービスをデプロイします。

構文

```
ceph orch apply -i FILE_NAME.yml
```

例

```
[ceph: root@host01 mds]# ceph orch apply -i mds.yml
```

6. MDS サービスがデプロイされ、機能したら、CephFS を作成します。

構文

```
ceph fs new CEPHFS_NAME METADATA_POOL DATA_POOL
```

例

```
[ceph: root@host01 /]# ceph fs new test metadata_pool data_pool
```

検証

- サービスを一覧表示します。

例

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

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

```
ceph orch ps --daemon_type=DAEMON_NAME
```

例

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=mds
```

関連情報

- Ceph ファイルシステム (CephFS) の作成に関する詳細は、[Red Hat Ceph Storage File System Guide](#) を参照してください。

9.4. CEPH ORCHESTRATOR を使用した MDS サービスの削除

`ceph orch rm` コマンドを使用してサービスを削除できます。または、ファイルシステムおよび関連するプールを削除できます。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- すべてのノードへの root レベルのアクセス。
- ホストがクラスタに追加されている。

- ホストにデプロイされた MDS デーモン1つ以上。

手順

- MDS デーモンをクラスターから削除する方法は2つあります。

方法1

- CephFS ボリューム、関連するプール、およびサービスを削除します。
 - a. Cephadm シェルにログインします。

例

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

- b. 設定パラメーター `mon_allow_pool_delete` を `true` に設定します。

例

```
[ceph: root@host01 /]# ceph config set mon mon_allow_pool_delete true
```

- c. ファイルシステムを削除します。

構文

```
ceph fs volume rm FILESYSTEM_NAME --yes-i-really-mean-it
```

例

```
[ceph: root@host01 /]# ceph fs volume rm cephfs-new --yes-i-really-mean-it
```

このコマンドは、ファイルシステム、そのデータ、メタデータプールを削除します。また、有効な `ceph-mgr` Orchestrator モジュールを使用して MDS を削除しようとします。

方法2

- `ceph orch rm` コマンドを使用して、クラスター全体から MDS サービスを削除します。
 - a. サービスを一覧表示します。

例

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

- b. サービスの削除

構文

```
ceph orch rm SERVICE_NAME
```

例

```
[ceph: root@host01 /]# ceph orch rm mds.test
```

検証

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

```
ceph orch ps
```

例

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

関連情報

- 詳細は、『Red Hat Ceph Storage Operations Guide』の「[Deploying the MDS service using the command line interface](#)」セクションを参照してください。
- 詳細は、『Red Hat Ceph Storage Operations Guide』の「[Deploying the MDS service using the service specification](#)」セクションを参照してください。

第10章 CEPH ORCHESTRATOR を使用した CEPH オブジェクトゲートウェイの管理

ストレージ管理者は、コマンドラインインターフェースまたはサービス指定を使用して、Ceph オブジェクトゲートウェイをデプロイできます。

また、マルチサイトオブジェクトゲートウェイを設定し、Ceph Orchestrator を使用して Ceph オブジェクトゲートウェイを削除することもできます。

Cephadm は、Ceph オブジェクトゲートウェイを、単一クラスターデプロイメントを管理するデーモンのコレクションまたはマルチサイトデプロイメントの特定のレルムおよびゾーンを管理するデーモンのコレクションとしてデプロイします。



注記

Cephadm では、オブジェクトゲートウェイデーモンは、**ceph.conf** やコマンドラインではなく、モニター設定データベースを使用して設定されます。この設定が **client.rgw** セクションにない場合は、オブジェクトゲートウェイデーモンがデフォルト設定で起動し、ポート **80** にバインドします。



注記

.default.rgw.buckets.index プールは、Ceph Object Gateway でバケットが作成された後にのみ作成されます。一方、データはバケットにアップロードされた後に **.default.rgw.buckets.data** プールが作成されます。

本セクションでは、以下の管理タスクを説明します。

- [コマンドラインインターフェースを使用した Ceph オブジェクトゲートウェイのデプロイ](#)。
- [サービス仕様を使用した Ceph オブジェクトゲートウェイのデプロイ](#)。
- [Ceph Orchestrator を使用したマルチサイト Ceph オブジェクトゲートウェイのデプロイ](#)。
- [Ceph Orchestrator を使用した Ceph オブジェクトゲートウェイの削除](#)

10.1. 前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- すべてのノードへの root レベルのアクセス。
- ホストがクラスターに追加されている。
- すべてのマネージャー、モニター、および OSD がストレージクラスターにデプロイされます。

10.2. コマンドラインインターフェースを使用した CEPH OBJECT GATEWAY のデプロイ

Ceph Orchestrator を使用すると、コマンドラインインターフェースで **ceph orch** コマンドを使用して Ceph オブジェクトゲートウェイをデプロイすることができます。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- すべてのノードへの root レベルのアクセス。
- ホストがクラスターに追加されている。
- すべてのマネージャー、モニター、および OSD デーモンがデプロイされている。

手順

1. Cephadm シェルにログインします。

例

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

2. Ceph オブジェクトゲートウェイデーモンは、以下の 3 つの方法でデプロイできます。

方法 1

- レルム、ゾングループ、ゾーンを作成し、ホスト名で配置指定を使用します。
 - a. レルムを作成します。

構文

```
radosgw-admin realm create --rgw-realm=REALM_NAME --default
```

例

```
[ceph: root@host01 /]# radosgw-admin realm create --rgw-realm=test_realm --default
```

- b. ゾングループを作成します。

構文

```
radosgw-admin zonegroup create --rgw-zonegroup=ZONE_GROUP_NAME --master --default
```

例

```
[ceph: root@host01 /]# radosgw-admin zonegroup create --rgw-zonegroup=default --master --default
```

- c. ゾーンを作成します。

構文

```
radosgw-admin zone create --rgw-zonegroup=ZONE_GROUP_NAME --rgw-zone=ZONE_NAME --master --default
```

例

-


```
[ceph: root@host01 /]# radosgw-admin zone create --rgw-zonegroup=default --rgw-zone=test_zone --master --default
```

- d. 変更をコミットします。

構文

```
radosgw-admin period update --rgw-realm=REALM_NAME --commit
```

例

```
[ceph: root@host01 /]# radosgw-admin period update --rgw-realm=test_realm --commit
```

- e. **ceph orch apply** コマンドを実行します。

構文

```
ceph orch apply rgw NAME [--realm=REALM_NAME] [--zone=ZONE_NAME] --placement="NUMBER_OF_DAEMONS [HOST_NAME_1 HOST_NAME_2]"
```

例

```
[ceph: root@host01 /]# ceph orch apply rgw test --realm=test_realm --zone=test_zone --placement="2 host01 host02"
```

方法 2

- 任意のサービス名を使用して、単一のクラスターデプロイメント用に 2 つの Ceph オブジェクトゲートウェイデーモンをデプロイします。

構文

```
ceph orch apply rgw SERVICE_NAME
```

例

```
[ceph: root@host01 /]# ceph orch apply rgw foo
```

方法 3

- ホストのラベル付きセットで任意のサービス名を使用します。

構文

```
ceph orch host label add HOST_NAME_1 LABEL_NAME
ceph orch host label add HOSTNAME_2 LABEL_NAME
ceph orch apply rgw SERVICE_NAME '--placement=label:_LABEL_NAME_ count-per-host:_NUMBER_OF_DAEMONS_' --port=8000
```



注記

`NUMBER_OF_DAEMONS` は、各ホストにデプロイされる Ceph オブジェクトゲートウェイの数を制御します。追加のコストを発生させずに最高のパフォーマンスを実現するには、この値を 2 に設定します。

例

```
[ceph: root@host01 /]# ceph orch host label add host01 rgw # the 'rgw' label can be anything
[ceph: root@host01 /]# ceph orch host label add host02 rgw
[ceph: root@host01 /]# ceph orch apply rgw foo '--placement=label:rgw count-per-host:2' --port=8000
```

検証

- サービスを一覧表示します。

例

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

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

```
ceph orch ps --daemon_type=DAEMON_NAME
```

例

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=rgw
```

10.3. サービス仕様を使用した CEPH OBJECT GATEWAY のデプロイ

Ceph Orchestrator を使用すると、サービス仕様を使用して Ceph オブジェクトゲートウェイをデプロイすることができます。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- すべてのノードへの root レベルのアクセス。
- ホストがクラスタに追加されている。
- すべてのマネージャー、モニター、および OSD デーモンがデプロイされている。

手順

1. Cephadm シェルにログインします。

例

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

- 以下のディレクトリーに移動します。

構文

```
cd /var/lib/ceph/DAEMON_PATH/
```

例

```
[ceph: root@host01 nfs/]# cd /var/lib/ceph/rgw/
```

rgw ディレクトリーが存在しない場合は、パスにディレクトリーを作成します。

- rgw.yml** ファイルを作成します。

例

```
[ceph: root@host01 rgw]# touch rgw.yml
```

- rgw.yml** ファイルを編集して、以下の詳細を追加します。

構文

```
service_type: rgw
service_id: REALM_NAME.ZONE_NAME
placement:
  hosts:
    - HOST_NAME_1
    - HOST_NAME_2
  count-per-host: NUMBER_OF_DAEMONS
spec:
  rgw_realm: REALM_NAME
  rgw_zone: ZONE_NAME
networks:
  - NETWORK_IP_ADDRESS # RGW service binds to a specific network
```



注記

NUMBER_OF_DAEMONS は、各ホストにデプロイされる Ceph オブジェクトゲートウェイの数を制御します。追加のコストを発生させずに最高のパフォーマンスを実現するには、この値を 2 に設定します。

例

```
service_type: rgw
service_id: test_realm.test_zone
placement:
  hosts:
    - host01
    - host02
    - host03
```

```
count-per-host: 2
spec:
  rgw_realm: test_realm
  rgw_zone: test_zone
networks:
  - 192.169.142.0/24
```

5. サービス仕様を使用して Ceph オブジェクトゲートウェイをデプロイします。

構文

```
ceph orch apply -i FILE_NAME.yaml
```

例

```
[ceph: root@host01 rgw]# ceph orch apply -i rgw.yaml
```

検証

- サービスを一覧表示します。

例

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

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

```
ceph orch ps --daemon_type=DAEMON_NAME
```

例

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=rgw
```

10.4. CEPH ORCHESTRATOR を使用したマルチサイト CEPH OBJECT GATEWAY のデプロイ

Ceph Orchestrator は、Ceph Object Gateway のマルチサイト設定オプションをサポートしています。

各オブジェクトゲートウェイを active-active ゾーン設定で機能するように設定すると、非プライマリーゾーンへの書き込みが可能になります。マルチサイト設定は、レルムと呼ばれるコンテナ内に保存されます。

レルムは、ゾーングループ、ゾーン、および期間を保存します。**rgw** デーモンは同期を処理し、個別の同期エージェントが不要になるため、アクティブ-アクティブ構成で動作します。

コマンドラインインターフェース (CLI) を使用してマルチサイトゾーンをデプロイすることもできます。



注記

以下の設定では、少なくとも2つの Red Hat Ceph Storage クラスタが地理的に別々の場所であることを前提としています。ただし、この設定は同じサイトでも機能します。

前提条件

- 少なくとも2つの実行中の Red Hat Ceph Storage クラスタ
- 少なくとも2つの Ceph Object Gateway インスタンス (各 Red Hat Ceph Storage クラスタに1つ)。
- すべてのノードへの root レベルのアクセス。
- ノードまたはコンテナがストレージクラスタに追加されます。
- すべての Ceph Manager、Monitor、および OSD デーモンがデプロイされます。

手順

1. **cephadm** シェルで、プライマリーゾーンを設定します。

- a. レルムを作成します。

構文

```
radosgw-admin realm create --rgw-realm=REALM_NAME --default
```

例

```
[ceph: root@host01 /]# radosgw-admin realm create --rgw-realm=test_realm --default
```

ストレージクラスタに単一のレルムがある場合は、**--default** フラグを指定します。

- b. プライマリーゾーングループを作成します。

構文

```
radosgw-admin zonegroup create --rgw-zonegroup=ZONE_GROUP_NAME --  
endpoints=http://RGW_PRIMARY_HOSTNAME:RGW_PRIMARY_PORT_NUMBER_1  
_ --master --default
```

例

```
[ceph: root@host01 /]# radosgw-admin zonegroup create --rgw-zonegroup=us --  
endpoints=http://rgw1:80 --master --default
```

- c. プライマリーゾーンを作成します。

構文

```
radosgw-admin zone create --rgw-zonegroup=PRIMARY_ZONE_GROUP_NAME --rgw-  
zone=PRIMARY_ZONE_NAME --  
endpoints=http://RGW_PRIMARY_HOSTNAME:RGW_PRIMARY_PORT_NUMBER_1
```

```
_ --access-key=SYSTEM_ACCESS_KEY --secret=SYSTEM_SECRET_KEY
```

例

```
[ceph: root@host01 /]# radosgw-admin zone create --rgw-zonegroup=us --rgw-zone=us-east-1 --endpoints=http://rgw1:80
```

- d. 必要に応じて、デフォルトゾーン、ゾーングループ、および関連するプールを削除します。

重要

デフォルトゾーンおよびゾーングループを使用してデータを保存する場合は、デフォルトゾーンとそのプールは削除しないでください。また、デフォルトのゾーングループを削除して、システムユーザーを削除します。

例

```
[ceph: root@host01 /]# radosgw-admin zonegroup delete --rgw-zonegroup=default
[ceph: root@host01 /]# ceph osd pool rm default.rgw.log default.rgw.log --yes-i-really-really-mean-it
[ceph: root@host01 /]# ceph osd pool rm default.rgw.meta default.rgw.meta --yes-i-really-really-mean-it
[ceph: root@host01 /]# ceph osd pool rm default.rgw.control default.rgw.control --yes-i-really-really-mean-it
[ceph: root@host01 /]# ceph osd pool rm default.rgw.data.root default.rgw.data.root --yes-i-really-really-mean-it
[ceph: root@host01 /]# ceph osd pool rm default.rgw.gc default.rgw.gc --yes-i-really-really-mean-it
```

- e. システムユーザーを作成します。

構文

```
radosgw-admin user create --uid=USER_NAME --display-name="USER_NAME" --access-key=SYSTEM_ACCESS_KEY --secret=SYSTEM_SECRET_KEY --system
```

例

```
[ceph: root@host01 /]# radosgw-admin user create --uid=zone.user --display-name="Zone user" --system
```

access_key および **secret_key** を書き留めておきます。

- f. アクセスキーとシステムキーをプライマリゾーンに追加します。

構文

```
radosgw-admin zone modify --rgw-zone=PRIMARY_ZONE_NAME --access-key=ACCESS_KEY --secret=SECRET_KEY
```

例

```
[ceph: root@host01 /]# radosgw-admin zone modify --rgw-zone=us-east-1 --access-
key=NE48APYCAODEPLKBCZVQ--
secret=u24GHQWRE3yxNBNFBzjM4jn14mFlckQ4EKL6LoW
```

- g. 変更をコミットします。

構文

```
radosgw-admin period update --commit
```

例

```
[ceph: root@host01 /]# radosgw-admin period update --commit
```

- h. **cephadm** シェルの外部で、ストレージクラスターおよびプロセスの **FSID** を取得します。

例

```
[root@host01 ~]# systemctl list-units | grep ceph
```

- i. Ceph Object Gateway デーモンを起動します。

構文

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

例

```
[root@host01 ~]# systemctl start ceph-62a081a6-88aa-11eb-a367-
001a4a000672@rgw.test_realm.us-east-1.host01.ahdtsw.service
[root@host01 ~]# systemctl enable ceph-62a081a6-88aa-11eb-a367-
001a4a000672@rgw.test_realm.us-east-1.host01.ahdtsw.service
```

2. Cephadm シェルで、セカンダリーゾーンを設定します。

- a. ホストからプライマリーレルム設定をプルします。

構文

```
radosgw-admin realm pull --url=URL_TO_PRIMARY_ZONE_GATEWAY --access-
key=ACCESS_KEY --secret-key=SECRET_KEY
```

例

```
[ceph: root@host04 /]# radosgw-admin realm pull --url=http://10.74.249.26:80 --access-
key=LIPEYZJLTWXRKXS9LPJC --secret-
key=lsAje0AVDNXNw48LjMAimpCpl7VaxJYSnfD0FFKQ
```

- b. ホストからプライマリー期間設定をプルします。

構文

```
radosgw-admin period pull --url=URL_TO_PRIMARY_ZONE_GATEWAY --access-
key=ACCESS_KEY --secret-key=SECRET_KEY
```

例

```
[ceph: root@host04 /]# radosgw-admin period pull --url=http://10.74.249.26:80 --access-
key=LIPEYZJLTWXRKXS9LPJC --secret-
key=lsAje0AVDNXNw48LjMAimpCpl7VaxJYSnfD0FFKQ
```

- c. セカンダリーゾーンを設定します。

構文

```
radosgw-admin zone create --rgw-zonegroup=ZONE_GROUP_NAME \
--rgw-zone=SECONDARY_ZONE_NAME --
endpoints=http://RGW_SECONDARY_HOSTNAME:_RGW_PRIMARY_PORT_NUMBE
R_1_ \
--access-key=SYSTEM_ACCESS_KEY --secret=SYSTEM_SECRET_KEY \
--endpoints=http://FQDN:80 \
[--read-only]
```

例

```
[ceph: root@host04 /]# radosgw-admin zone create --rgw-zonegroup=us --rgw-zone=us-
east-2 --endpoints=http://rgw2:80 --access-key=LIPEYZJLTWXRKXS9LPJC --secret-
key=lsAje0AVDNXNw48LjMAimpCpl7VaxJYSnfD0FFKQ
```

- d. 必要に応じて、デフォルトゾーンを削除します。



重要

デフォルトゾーンおよびゾーングループを使用してデータを保存する場合は、デフォルトゾーンとそのプールは削除しないでください。

例

```
[ceph: root@host04 /]# radosgw-admin zone rm --rgw-zone=default
[ceph: root@host04 /]# ceph osd pool rm default.rgw.log default.rgw.log --
yes-i-really-really-mean-it
[ceph: root@host04 /]# ceph osd pool rm default.rgw.meta
default.rgw.meta --yes-i-really-really-mean-it
[ceph: root@host04 /]# ceph osd pool rm default.rgw.control
default.rgw.control --yes-i-really-really-mean-it
[ceph: root@host04 /]# ceph osd pool rm default.rgw.data.root
default.rgw.data.root --yes-i-really-really-mean-it
[ceph: root@host04 /]# ceph osd pool rm default.rgw.gc default.rgw.gc --
yes-i-really-really-mean-it
```


- e. Ceph 設定データベースを更新します。

構文

```
ceph config set _SERVICE_NAME_ rgw_zone _SECONDARY_ZONE_NAME_
```

例

```
[ceph: root@host04 /]# ceph config set rgw rgw_zone us-east-2
```

- f. 変更をコミットします。

構文

```
radosgw-admin period update --commit
```

例

```
[ceph: root@host04 /]# radosgw-admin period update --commit
```

- g. Cephadm シェルの外部で、ストレージクラスターおよびプロセスの FSID を取得します。

例

```
[root@host04 ~]# systemctl list-units | grep ceph
```

- h. Ceph Object Gateway デモンを起動します。

構文

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

例

```
[root@host04 ~]# systemctl start ceph-62a081a6-88aa-11eb-a367-  
001a4a000672@rgw.test_realm.us-east-2.host04.ahdtsw.service  
[root@host04 ~]# systemctl enable ceph-62a081a6-88aa-11eb-a367-  
001a4a000672@rgw.test_realm.us-east-2.host04.ahdtsw.service
```

3. 必要に応じて、配置仕様を使用して、マルチサイトの Ceph Object Gateway をデプロイします。

構文

```
ceph orch apply rgw _NAME_ --realm=_REALM_NAME_ --  
zone=_PRIMARY_ZONE_NAME_ --placement="_NUMBER_OF_DAEMONS_  
_HOST_NAME_1_ _HOST_NAME_2_"
```

例

```
[ceph: root@host04 /]# ceph orch apply rgw east --realm=test_realm --zone=us-east-1 --placement="2 host01 host02"
```

検証

- 同期のステータスを確認してデプロイメントを確認します。

例

```
[ceph: root@host04 /]# radosgw-admin sync status
```

10.5. CEPH ORCHESTRATOR を使用した CEPH OBJECT GATEWAY の削除

`ceph orch rm` コマンドを使用して、Ceph オブジェクトゲートウェイデーモンを削除できます。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- すべてのノードへの root レベルのアクセス。
- ホストがクラスタに追加されている。
- ホストにデプロイされた Ceph オブジェクトゲートウェイデーモンが少なくとも1つ含まれます。

手順

1. Cephadm シェルにログインします。

例

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

2. サービスを一覧表示します。

例

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

3. サービスの削除

構文

```
ceph orch rm SERVICE_NAME
```

例

```
[ceph: root@host01 /]# ceph orch rm rgw.test_realm.test_zone_bb
```

検証

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

```
ceph orch ps
```

例

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

関連情報

- 詳細は、『Red Hat Ceph Storage Operations Guide』の「[Deploying the Ceph object gateway using the command-line interface](#)」セクションを参照してください。
- 詳細は、『Red Hat Ceph Storage 操作ガイド』の「[サービス仕様を使用して Ceph オブジェクトゲートウェイのデプロイ](#)」セクションを参照してください。

第11章 CEPH ORCHESTRATOR を使用した NFS-GANESHA ゲートウェイの管理 (利用制限あり)

ストレージ管理者は、バックエンドにて Cephadm と Orchestrator を使用して、NFS-Ganesha ゲートウェイをデプロイできます。Cephadm は、事前定義された RADOS プールおよびオプションの namespace を使用して NFS Ganesha をデプロイします。



注記

この技術には、利用制限があります。詳細は、[Deprecated functionality](#) の章を参照してください。



注記

Red Hat は、NFS v4.0+ プロトコルでのみ CephFS エクスポートをサポートします。

本セクションでは、以下の管理タスクを説明します。

- [Ceph Orchestrator を使用した NFS-Ganesha クラスターの作成](#)
- [コマンドラインインターフェースを使用した NFS-Ganesha ゲートウェイのデプロイ。](#)
- [サービス指定を使用した NFS-Ganesha ゲートウェイのデプロイ。](#)
- [Ceph Orchestrator を使用した NFS-Ganesha クラスター情報の確認](#)
- [Ceph Orchestrator を使用した NFS-Ganesha cluster ログの取得](#)
- [Ceph Orchestrator を使用したカスタム NFS-Ganesha 設定](#)
- [Ceph Orchestrator を使用したカスタム NFS-Ganesha 設定のリセット。](#)
- [Ceph Orchestrator を使用した NFS-Ganesha クラスターの削除](#)
- [Ceph Orchestrator を使用した NFS Ganesha ゲートウェイの削除](#)

11.1. 前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- すべてのノードへの root レベルのアクセス。
- ホストがクラスターに追加されている。
- すべてのマネージャー、モニター、および OSD デーモンがデプロイされている。

11.2. CEPH ORCHESTRATOR を使用した NFS-GANESHA クラスターの作成

Ceph Orchestrator の `mgr/nfs` モジュールを使用して、NFS-Ganesha クラスターを作成できます。このモジュールは、バックエンドで Cephadm を使用して NFS クラスターをデプロイします。

これにより、すべての NFS-Ganesha デーモンの共通のリカバリープール、**clusterid** に基づく新規ユーザー、および共通の NFS-Ganesha 設定 RADOS オブジェクトが作成されます。

デーモンごとに、新しいユーザーおよび共通の設定がプールに作成されます。すべてのクラスターにはクラスター名に関して異なる namespace がありますが、それらは同じリカバリープールを使用します。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- ホストがクラスターに追加されている。
- すべてのマネージャー、モニター、および OSD デーモンがデプロイされている。

手順

1. Cephadm シェルにログインします。

例

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

2. **mgr/nfs** モジュールを有効にします。

例

```
[ceph: root@host01 /]# ceph mgr module enable nfs
```

3. クラスターを作成します。

構文

```
ceph nfs cluster create CLUSTER_NAME ["HOST_NAME_1_ HOST_NAME_2  
HOST_NAME_3"]
```

CLUSTER_NAME は任意の文字列です。**HOST_NAME_1** は、NFS-Ganesha デーモンをデプロイするためにホストを表す任意の文字列です。

例

```
[ceph: root@host01 /]# ceph nfs cluster create nfsganesha "host01 host02"  
NFS Cluster Created Successful
```

これにより、**host01** および **host02** の1つのデーモンと、NFS_Ganesha クラスター **nfsganesha** が作成されます。

検証

- クラスターの詳細を一覧表示します。

例

```
[ceph: root@host01 /]# ceph nfs cluster ls
```

- NFS-Ganesha クラスター情報を表示します。

構文

```
ceph nfs cluster info CLUSTER_NAME
```

例

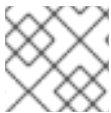
```
[ceph: root@host01 /]# ceph nfs cluster info nfsganesha
```

関連情報

- 詳細は、『Red Hat Ceph Storage File System guide』の「[Exporting Ceph File System namespaces over the NFS protocol](#)」セクションを参照してください。
- 詳細は、『Red Hat Ceph Storage Operations Guide』の「[Deploying the Ceph daemons using the placement specification](#)」セクションを参照してください。

11.3. コマンドラインインターフェースを使用した NFS-GANESHA ゲートウェイのデプロイ

バックエンドで Cephadm と Ceph Orchestrator を使用し、配置指定を使用して NFS-Ganesha ゲートウェイをデプロイできます。この場合、RADOS プールを作成し、namespace を作成してから、ゲートウェイをデプロイする必要があります。



注記

Red Hat は、NFS v4.0+ プロトコルでのみ CephFS エクスポートをサポートします。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- ホストがクラスターに追加されている。
- すべてのマネージャー、モニター、および OSD デーモンがデプロイされている。

手順

1. Cephadm シェルにログインします。

例

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

2. RADOS プール namespace を作成し、アプリケーションを有効にします。

構文

```
ceph osd pool create POOL_NAME _
ceph osd pool application enable POOL_NAME freeform/rgw/rbd/cephfs/nfs
rbd pool init -p POOL_NAME
```

例

```
[ceph: root@host01 /]# ceph osd pool create nfs-ganesha
[ceph: root@host01 /]# ceph osd pool application enable nfs-ganesha nfs
[ceph: root@host01 /]# rbd pool init -p nfs-ganesha
```

3. コマンドラインインターフェースで配置指定を使用して NFS-Ganesha ゲートウェイをデプロイします。

構文

```
ceph orch apply nfs SERVICE_ID --pool POOL_NAME --namespace NAMESPACE --
placement="NUMBER_OF_DAEMONS HOST_NAME_1 HOST_NAME_2 HOST_NAME_3"
```

例

```
[ceph: root@host01 /]# ceph orch apply nfs foo --pool nfs-ganesha --namespace nfs-ns --
placement="2 host01 host02"
```

これにより、**host01** および **host02** の1つのデーモンと、NFS-Ganesha クラスター **nfs-ganesha** がデプロイされます。

検証

- サービスを一覧表示します。

例

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

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

```
ceph orch ps --daemon_type=DAEMON_NAME
```

例

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=nfs
```

関連情報

- 詳細は、『Red Hat Ceph Storage Operations Guide』の「[Deploying the Ceph daemons using the command line interface](#)」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「[Creating a block device pool](#)」セクションを参照してください。

11.4. サービス指定を使用した NFS-GANESHA ゲートウェイのデプロイ

バックエンドで Cephadm と Ceph Orchestrator を使用し、サービス指定を使用して NFS-Ganesha ゲートウェイをデプロイできます。この場合、RADOS プールを作成し、namespace を作成してから、ゲートウェイをデプロイする必要があります。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- ホストがクラスタに追加されている。

手順

1. Cephadm シェルにログインします。

例

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

2. RADOS プールと namespace を作成し、RBD を有効にします。

構文

```
ceph osd pool create POOL_NAME _  
ceph osd pool application enable POOL_NAME rbd  
rbd pool init -p POOL_NAME
```

例

```
[ceph: root@host01 /]# ceph osd pool create nfs-ganesha  
[ceph: root@host01 /]#ceph osd pool application enable nfs-ganesha rbd  
[ceph: root@host01 /]#rbd pool init -p nfs-ganesha
```

3. 以下のディレクトリーに移動します。

構文

```
cd /var/lib/ceph/DAEMON_PATH/
```

例

```
[ceph: root@host01 nfs/]# cd /var/lib/ceph/nfs/
```

nfs ディレクトリーが存在しない場合は、パス上に作成します。

4. **nfs.yml** ファイルを作成します。

例

```
[ceph: root@host01 nfs]# touch nfs.yml
```


5. **nfs.yml** ファイルを編集し、以下の詳細が含まれるようにします。

構文

```
service_type: nfs
service_id: SERVICE_ID
placement:
  hosts:
    - HOST_NAME_1
    - HOST_NAME_2
spec:
  pool: POOL_NAME
  namespace: NAMESPACE
```

例

```
service_type: nfs
service_id: foo
placement:
  hosts:
    - host01
    - host02
spec:
  pool: nfs-ganesha
  namespace: nfs-ns
```

6. サービス指定を使用して NFS-Ganesha ゲートウェイをデプロイします。

構文

```
ceph orch apply -i FILE_NAME.yml
```

例

```
[ceph: root@host01 nfs]# ceph orch apply -i nfs.yml
```

検証

- サービスを一覧表示します。

例

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

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

```
ceph orch ps --daemon_type=DAEMON_NAME
```

例

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=nfs
```

関連情報

- 詳細は、『Red Hat Ceph Storage Block Device Guide』の「[Creating a block device pool](#)」セクションを参照してください。

11.5. CEPH ORCHESTRATOR を使用した NFS-GANESHA クラスターの更新

バックエンドで Ceph Orchestrator と Cephadm を使用して、ホスト上のデーモンの配置を変更して、NFS-Ganesha クラスターを更新できます。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- ホストがクラスターに追加されている。
- すべてのマネージャー、モニター、および OSD デーモンがデプロイされている。
- **mgr/nfs** モジュールを使用して作成された NFS-Ganesha クラスター。

手順

1. Cephadm シェルにログインします。

例

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

2. クラスターを更新します。

構文

```
ceph orch apply nfs CLUSTER_NAME  
["HOST_NAME_1,HOST_NAME_2,HOST_NAME_3"]
```

CLUSTER_NAME は任意の文字列です。**HOST_NAME_1** は、デプロイされた NFS-Ganesha デーモンをするためにホストを表す任意の文字列です。

例

```
[ceph: root@host01 /]# ceph orch apply nfs nfsganesha "host02"
```

これにより、**host02** 上の **nfsganesha** クラスターが更新されます。

検証

- クラスターの詳細を一覧表示します。

例

```
[ceph: root@host01 /]# ceph nfs cluster ls
```

- NFS-Ganesha クラスター情報を表示します。

構文

```
ceph nfs cluster info CLUSTER_NAME
```

例

```
[ceph: root@host01 /]# ceph nfs cluster info nfsganesha
```

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

```
ceph orch ps --daemon_type=DAEMON_NAME
```

例

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=nfs
```

関連情報

- 詳細は、『Red Hat Ceph Storage Operations Guide』の「[Creating the NFS-Ganesha cluster using the Ceph Orchestrator](#)」セクションを参照してください。

11.6. CEPH ORCHESTRATOR を使用した NFS-GANESHA クラスター情報の確認

Ceph Orchestrator を使用して、NFS-Ganesha クラスターの情報を確認できます。すべての NFS Ganesha クラスターや、ポート、IP アドレス、およびクラスターが作成されたホストの名前のある特定のクラスターに関する情報を取得できます。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- ホストがクラスターに追加されている。
- すべてのマネージャー、モニター、および OSD デーモンがデプロイされている。
- **mgr/nfs** モジュールを使用して作成された NFS-Ganesha クラスター。

手順

1. Cephadm シェルにログインします。

例

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

2. NFS-Ganesha クラスター情報を表示します。

構文

```
ceph nfs cluster info CLUSTER_NAME
```

例

```
[ceph: root@host01 /]# ceph nfs cluster info nfsganesha

{
  "nfsganesha": [
    {
      "hostname": "host02",
      "ip": [
        "10.74.251.164"
      ],
      "port": 2049
    }
  ]
}
```

関連情報

- 詳細は、『Red Hat Ceph Storage Operations Guide』の「[Creating the NFS-Ganesha cluster using the Ceph Orchestrator](#)」セクションを参照してください。

11.7. CEPH ORCHESTRATOR を使用した NFS-GANESHA CLUSTER ログの取得

Ceph Orchestrator を使用すると、NFS-Ganesha クラスターログを取得できます。サービスがデプロイされているノードにいる必要があります。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- NFS がデプロイされたノードに Cephadm がインストールされている。
- すべてのノードへの root レベルのアクセス。
- ホストがクラスターに追加されている。
- **mgr/nfs** モジュールを使用して作成された NFS-Ganesha クラスター。

手順

1. root ユーザーとして、ストレージクラスターの **FSID** を取得します。

例

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

FSID とサービスの名前をコピーします。

2. ログを取得します。

構文

```
cephadm logs --fsid FSID --name SERVICE_NAME
```

例

```
[root@host03 ~]# cephadm logs --fsid 499829b4-832f-11eb-8d6d-001a4a000635 --name
nfs.foo.host03
```

関連情報

- 詳細は、『Red Hat Ceph Storage Operations Guide』の「[Deploying the Ceph daemons using the placement specification](#)」セクションを参照してください。

11.8. CEPH ORCHESTRATOR を使用したカスタム NFS-GANESHA 設定の設定

NFS-Ganesha クラスタはデフォルトの設定ブロックで定義されます。Ceph Orchestrator を使用すると、設定をカスタマイズでき、デフォルトの設定ブロックよりも優先されます。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- ホストがクラスタに追加されている。
- すべてのマネージャー、モニター、および OSD デーモンがデプロイされている。
- **mgr/nfs** モジュールを使用して作成された NFS-Ganesha クラスタ。

手順

1. Cephadm シェルにログインします。

例

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

2. 以下は、NFS-Ganesha クラスタのデフォルト設定の例です。

例

```
# {{ cephadm_managed }}
NFS_CORE_PARAM {
    Enable_NLM = false;
    Enable_RQUOTA = false;
    Protocols = 4;
}
```

```

MDCACHE {
    Dir_Chunk = 0;
}

EXPORT_DEFAULTS {
    Attr_Expiration_Time = 0;
}

NFSv4 {
    Delegations = false;
    RecoveryBackend = 'rados_cluster';
    Minor_Versions = 1, 2;
}

RADOS_KV {
    UserId = "{{ user }}";
    nodeid = "{{ nodeid }}";
    pool = "{{ pool }}";
    namespace = "{{ namespace }}";
}

RADOS_URLS {
    UserId = "{{ user }}";
    watch_url = "{{ url }}";
}

RGW {
    cluster = "ceph";
    name = "client.{{ rgw_user }}";
}

%url {{ url }}

```

3. NFS-Ganesha クラスタ設定をカスタマイズします。以下は、設定をカスタマイズする2つの例です。

- ログレベルを変更します。

例

```

LOG {
    COMPONENTS {
        ALL = FULL_DEBUG;
    }
}

```

- カスタムエクスポートブロックを追加します。
 - a. ユーザーを作成します。



注記

FSAL ブロックで指定されたユーザーは、NFS-Ganesha デーモンが Ceph クラスタにアクセスするための適切な上限を設定する必要があります。

構文

```
ceph auth get-or-create client.USER_NAME mon 'allow r' osd 'allow rw
pool=POOL_NAME namespace=NFS_CLUSTER_NAME, allow rw tag cephfs
data=FILE_SYSTEM_NAME' mds 'allow rw path=EXPORT_PATH'
```

例

```
[ceph: root@host01 /]# ceph auth get-or-create client.nfstest1 mon 'allow r' osd 'allow
rw pool=nfsganesh namespace=nfs_cluster_name, allow rw tag cephfs
data=filesystem_name' mds 'allow rw path=export_path
```

- b. 以下のディレクトリーに移動します。

構文

```
cd /var/lib/ceph/DAEMON_PATH/
```

例

```
[ceph: root@host01 /]# cd /var/lib/ceph/nfs/
```

nfs ディレクトリーが存在しない場合は、パス上に作成します。

- c. 新しい設定ファイルを作成します。

構文

```
touch PATH_TO_CONFIG_FILE
```

例

```
[ceph: root@host01 nfs]# touch nfs-ganesha.conf
```

- d. カスタムエクスポートブロックを追加して設定ファイルを編集します。1つのエクスポートを作成し、これは Ceph NFS エクスポートインターフェースによって管理されます。

構文

```
EXPORT {
  Export_Id = NUMERICAL_ID;
  Transports = TCP;
  Path = PATH_WITHIN_CEPHFS;
  Pseudo = BINDING;
  Protocols = 4;
  Access_Type = PERMISSIONS;
  Attr_Expiration_Time = 0;
  Squash = None;
  FSAL {
    Name = CEPH;
    Filesystem = "FILE_SYSTEM_NAME";
```

```

    User_Id = "USER_NAME";
    Secret_Access_Key = "USER_SECRET_KEY";
  }
}

```

例

```

EXPORT {
  Export_Id = 100;
  Transports = TCP;
  Path = /;
  Pseudo = /ceph/;
  Protocols = 4;
  Access_Type = RW;
  Attr_Expiration_Time = 0;
  Squash = None;
  FSAL {
    Name = CEPH;
    Filesystem = "filesystem name";
    User_Id = "user id";
    Secret_Access_Key = "secret key";
  }
}

```

4. 新しい設定をクラスターに適用します。

構文

```
ceph nfs cluster config set _CLUSTER_NAME_ -i _PATH_TO_CONFIG_FILE_
```

例

```
[ceph: root@host01 nfs]# ceph nfs cluster config set nfs-ganesha -i /root/nfs-ganesha.conf
```

これにより、カスタム設定のサービスも再起動されます。

検証

- サービスを一覧表示します。

例

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

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

```
ceph orch ps --daemon_type=DAEMON_NAME
```

例


```
[ceph: root@host01 /]# ceph orch ps --daemon_type=nfs
```

- カスタム設定を確認します。

構文

```
./bin/rados -p POOL_NAME -N CLUSTER_NAME get userconf-nfs.CLUSTER_NAME -
```

例

```
[ceph: root@host01 /]# ./bin/rados -p nfs-ganesha -N nfsganesha get userconf-nfs.nfsganesha -
```

関連情報

- 詳細は、『Red Hat Ceph Storage Operations Guide』の「[Resetting custom NFS-Ganesha configuration using the Ceph Orchestrator](#)」セクションを参照してください。

11.9. CEPH ORCHESTRATOR を使用したカスタム NFS-GANESHA 設定のリセット

Ceph Orchestrator を使用すると、ユーザー定義の設定をデフォルト設定にリセットできます。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- ホストがクラスタに追加されている。
- すべてのマネージャー、モニター、および OSD デーモンがデプロイされている。
- **mgr/nfs** モジュールを使用してデプロイされた NFS-Ganesha
- カスタム NFS クラスタの設定がセットアップされている。

手順

1. Cephadm シェルにログインします。

例

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

2. NFS_Ganesha 設定をリセットします。

構文

```
ceph nfs cluster config reset CLUSTER_NAME
```

例

```
[ceph: root@host01 /]# ceph nfs cluster config reset nfsganesha
```

検証

- サービスを一覧表示します。

例

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

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

```
ceph orch ps --daemon_type=DAEMON_NAME
```

例

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=nfs
```

- カスタム設定が削除されていることを確認します。

構文

```
./bin/rados -p POOL_NAME -N CLUSTER_NAME get userconf-nfs.CLUSTER_NAME -
```

例

```
[ceph: root@host01 /]# ./bin/rados -p nfs-ganesha -N nfsganesha get userconf-nfs.nfsganesha -
```

関連情報

- 詳細は、『Red Hat Ceph Storage Operations Guide』の「[Creating the NFS-Ganesha cluster using the Ceph Orchestrator](#)」セクションを参照してください。
- 詳細は、『Red Hat Ceph Storage Operations Guide』の「[Setting custom NFS-Ganesha configuration using the Ceph Orchestrator](#)」セクションを参照してください。

11.10. CEPH ORCHESTRATOR を使用した NFS-GANESHA クラスターの削除

バックエンドで Cephadm と Ceph Orchestrator を使用すると、NFS-Ganesha クラスターを削除できます。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- ホストがクラスターに追加されている。

- すべてのマネージャー、モニター、および OSD デーモンがデプロイされている。
- **mgr/nfs** モジュールを使用して作成された NFS-Ganesha クラスター。

手順

1. Cephadm シェルにログインします。

例

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

2. クラスターを削除します。

構文

```
ceph nfs cluster delete CLUSTER_NAME
```

CLUSTER_NAME は任意の文字列です。

例

```
[ceph: root@host01 /]# ceph nfs cluster delete nfsganesha  
NFS Cluster Deleted Successfully
```

検証

- クラスターの詳細を一覧表示します。

例

```
[ceph: root@host01 /]# ceph nfs cluster ls
```

関連情報

- 詳細は、『Red Hat Ceph Storage File System guide』の「[Exporting Ceph File System namespaces over the NFS protocol](#)」セクションを参照してください。
- 詳細は、『Red Hat Ceph Storage Operations Guide』の「[Creating the NFS-Ganesha cluster using the Ceph Orchestrator](#)」セクションを参照してください。

11.11. CEPH ORCHESTRATOR を使用した NFS-GANESHA ゲートウェイの削除

ceph orch rm コマンドを使用して、NFS-Ganesha ゲートウェイを削除できます。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- すべてのノードへの root レベルのアクセス。

- ホストがクラスターに追加されている。
- ホストにデプロイされた1つ以上の NFS-Ganesha ゲートウェイ。

手順

1. Cephadm シェルにログインします。

例

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

2. サービスを一覧表示します。

例

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

3. サービスの削除

構文

```
ceph orch rm SERVICE_NAME
```

例

```
[ceph: root@host01 /]# ceph orch rm nfs.foo
```

検証

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

```
ceph orch ps
```

例

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

関連情報

- 詳細は、『Red Hat Ceph Storage Operations Guide』の「[Deploying the Ceph daemons using the placement specification](#)」セクションを参照してください。
- 詳細は、『Red Hat Ceph Storage Operations Guide』の「[Deploying the NFS-Ganesha gateway using the service specification](#)」セクションを参照してください。
- 詳細は、『Red Hat Ceph Storage Operations Guide』の「[Deploying the NFS-Ganesha gateway using the placement specification](#)」セクションを参照してください。

第12章 CEPH ORCHESTRATOR を使用した iSCSI ゲートウェイの管理 (利用制限あり)

ストレージ管理者は、Ceph Orchestrator を使用して iSCSI ゲートウェイをデプロイできます。iSCSI ゲートウェイは、RADOS Block Device (RBD) イメージを SCSI ディスクとしてエクスポートする高可用性 (HA) iSCSI ターゲットを提供します。

YAML ファイルのように、配置指定またはサービス指定のいずれかを使用して iSCSI ゲートウェイをデプロイできます。



注記

この技術には、利用制限があります。詳細は、[Deprecated functionality](#) の章を参照してください。

本セクションでは、以下の管理タスクを説明します。

- [配置指定を使用した iSCSI ゲートウェイのデプロイ](#)。
- [サービス指定を使用した iSCSI ゲートウェイのデプロイ](#)。
- [Ceph Orchestrator を使用した iSCSI ゲートウェイの削除](#)。

12.1. 前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- すべてのノードへの root レベルのアクセス。
- ホストがクラスタに追加されている。
- すべてのマネージャー、モニター、および OSD デーモンがデプロイされている。

12.2. コマンドラインインターフェースを使用した iSCSI ゲートウェイのデプロイ

Ceph Orchestrator を使用すると、コマンドラインインターフェースで `ceph orch` コマンドを使用して iSCSI ゲートウェイをデプロイできます。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- ホストがクラスタに追加されている。
- すべてのマネージャー、モニター、および OSD デーモンがデプロイされている。

手順

1. `Cephadm` シェルにログインします。

例

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

2. プールを作成します。

構文

```
ceph osd pool create POOL_NAME
```

例

```
[ceph: root@host01 /]# ceph osd pool create mypool
```

3. コマンドラインインターフェースを使用して iSCSI ゲートウェイをデプロイします。

構文

```
ceph orch apply iscsi POOLNAME admin admin --placement='NUMBER_OF_DAEMONS  
HOST_NAME_1 HOST_NAME_2'
```

例

```
[ceph: root@host01 /]# ceph orch apply iscsi mypool admin admin --placement="1 host01"
```

検証

- サービスを一覧表示します。

例

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

- ホストおよびプロセスを一覧表示します。

構文

```
ceph orch ps --daemon_type=DAEMON_NAME
```

例

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=iscsi
```

12.3. サービス指定を使用した iSCSI ゲートウェイのデプロイ

Ceph Orchestrator を使用すると、サービス指定を使用して iSCSI ゲートウェイをデプロイできます。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- ホストがクラスターに追加されている。

- すべてのマネージャー、モニター、および OSD デーモンがデプロイされている。

手順

1. Cephadm シェルにログインします。

例

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

2. 以下のディレクトリーに移動します。

構文

```
cd /var/lib/ceph/DAEMON_PATH/
```

例

```
[ceph: root@host01 /]# cd /var/lib/ceph/iscsi/
```

注記: iscsi ディレクトリーが存在しない場合は作成します。

3. **iscsi.yml** ファイルを作成します。

例

```
[ceph: root@host01 iscsi]# touch iscsi.yml
```

4. **iscsi.yml** ファイルを編集して、以下の詳細が含まれるようにします。

構文

```
service_type: iscsi
service_id: iscsi
placement:
  hosts:
    - HOST_NAME_1
    - HOST_NAME_2
spec:
  pool: POOL_NAME # RADOS pool where ceph-iscsi config data is stored.
  trusted_ip_list: "IP_ADDRESS_1,IP_ADDRESS_2" # optional
  api_port: ... # optional
  api_user: API_USERNAME # optional
  api_password: API_PASSWORD # optional
  api_secure: true/false # optional
  ssl_cert: | # optional
  ...
  ssl_key: | # optional
  ...
```

例

```
service_type: iscsi
```

```
service_id: iscsi
placement:
  hosts:
    - host01
spec:
  pool: mypool
```

5. サービス指定を使用して iSCSI ゲートウェイをデプロイします。

構文

```
ceph orch apply -i FILE_NAME.yml
```

例

```
[ceph: root@host01 iscsi]# ceph orch apply -i iscsi.yml
```

検証

- サービスを一覧表示します。

例

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

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

```
ceph orch ps --daemon_type=DAEMON_NAME
```

例

```
[ceph: root@host01 /]# ceph orch ps --daemon_type=iscsi
```

12.4. CEPH ORCHESTRATOR を使用した iSCSI ゲートウェイの削除

`ceph orch rm` コマンドを使用して、iSCSI ゲートウェイデーモンを削除できます。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- すべてのノードへの root レベルのアクセス。
- ホストがクラスタに追加されている。
- ホストにデプロイされた1つ以上の iSCSI ゲートウェイデーモン。

手順

1. Cephadm シェルにログインします。

例

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

2. サービスを一覧表示します。

例

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

- a. サービスの削除

構文

```
ceph orch rm SERVICE_NAME
```

例

```
[ceph: root@host01 /]# ceph orch rm iscsi.iscsi
```

検証

- ホスト、デーモン、およびプロセスを一覧表示します。

構文

```
ceph orch ps
```

例

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

関連情報

- 詳細は、『Red Hat Ceph Storage Operations Guide』の「[Deploying the iSCSI gateway using the command line interface](#)」セクションを参照してください。
- 詳細は、『Red Hat Ceph Storage Operations Guide』の「[Deploying the iSCSI gateway using the service specification](#)」セクションを参照してください。

第13章 SNMP トラップの設定

ストレージ管理者は、Red Hat Ceph Storage クラスタに簡易ネットワーク管理プロトコル (SNMP) ゲートウェイをデプロイおよび設定して、Prometheus Alertmanager からアラートを受信し、SNMP トラップとしてそれらをクラスタにルーティングできます。

13.1. 簡易ネットワーク管理プロトコル

簡易ネットワーク管理プロトコル (SNMP) は、さまざまなハードウェアおよびソフトウェアプラットフォームにまたがる分散システムおよびデバイスを監視するためのもので、最も広く使用されているオープンプロトコルの1つです。Ceph の SNMP 統合は、Prometheus Alertmanager クラスタからゲートウェイデーモンへのアラートの転送に重点を置いています。ゲートウェイデーモンはアラートを SNMP 通知に変換し、これを指定された SNMP 管理プラットフォームに送信します。ゲートウェイデーモンは `snmp_notifier_project` からのものであり、認証と暗号化による SNMP V2c および V3 サポートを提供します。

Red Hat Ceph Storage SNMP ゲートウェイサービスは、デフォルトでゲートウェイの1つのインスタンスをデプロイします。配置情報を提供することで、これを増やすことができます。ただし、複数の SNMP ゲートウェイデーモンを有効にすると、SNMP 管理プラットフォームは同じイベントに対して複数の通知を受け取ります。

SNMP トラップはアラートメッセージであり、Prometheus Alertmanager はこれらのアラートを SNMP 通知機能に送信し、SNMP 通知機能は指定されたアラートのラベルでオブジェクト識別子 (OID) を探します。各 SNMP トラップには一意の ID があり、ステータスが更新された追加のトラップを特定の SNMP ポーラーに送信できます。SNMP は Ceph の健全性チェックにフックして、すべての健全性警告が特定の SNMP トラップを生成するようにします。

正しく動作し、デバイスステータスに関する情報をユーザーに転送して監視するために、SNMP はいくつかのコンポーネントに依存しています。SNMP を設定する 4 つの主要なコンポーネントがあります。

- **SNMP マネージャー** - SNMP マネージャーは、管理ステーションとも呼ばれ、ネットワーク監視プラットフォームを実行するコンピューターです。SNMP 対応デバイスをポーリングし、それらからデータを取得する役割を持つプラットフォーム。SNMP マネージャーは、エージェントにクエリーを実行し、エージェントからの応答を受信し、エージェントからの非同期イベントを確認します。
- **SNMP エージェント** - SNMP エージェントは、管理対象のシステムで実行されるプログラムであり、システムの MIB データベースが含まれています。これらは、帯域幅やディスク容量などのデータを収集して集約し、Management Information Base (MIB) に送信します。
- **Management information base (MIB)** - これらは SNMP エージェントに含まれるコンポーネントです。SNMP マネージャーはこれをデータベースとして使用し、エージェントに特定の情報へのアクセスを要求します。この情報は、ネットワーク管理システム (NMS) に必要です。NMS は、エージェントをポーリングしてこれらのファイルから情報を取得してから、それらをユーザーが見ることができるグラフやディスプレイに変換する処理を行います。MIB には、ネットワークデバイスによって決定される統計値と制御値が含まれています。
- **SNMP デバイス**

以下のバージョンの SNMP は互換性があり、ゲートウェイの実装でサポートされています。

- **V2c** - 認証なしでコミュニティ文字列を使用し、外部からの攻撃に対して脆弱です。
- **V3 authNoPriv** - 暗号化せずにユーザー名とパスワードの認証を使用します。

- **V3 authPriv** - SNMP 管理プラットフォームへの暗号化を伴うユーザー名とパスワードの認証を使用します。



重要

SNMP トラップを使用する場合は、バージョン番号のセキュリティー設定が正しいことを確認して、SNMP に固有の脆弱性を最小限に抑え、許可されていないユーザーからネットワークを保護してください。

13.2. SNMPTRAPD の設定

snmptrapd デーモンには、**snmp-gateway** サービスの作成時に指定する必要がある認証設定が含まれているため、**snmp-gateway** をデプロイする前に Simple Network Management Protocol (SNMP) ターゲットを設定することが重要です。

SNMP ゲートウェイ機能は、Prometheus スタックで生成されたアラートを SNMP 管理プラットフォームに公開する手段を提供します。**snmptrapd** ツールに基づいて、宛先への SNMP トラップを設定できます。このツールを使用すると、1 つ以上の SNMP トラップリスナーを確立できます。

設定には以下のパラメーターが重要となります。

- **engine-id** は、デバイスの一意的識別子 (16 進数) であり、SNMPV3 ゲートウェイで必要とされています。Red Hat では、このパラメーターに ``8000C53F_CLUSTER_FSID_WITHOUT_DASHES_`` を使用することをお勧めしています。
- **SNMP_COMMUNITY_FOR_SNMPV2** パラメーターである **snmp-community** は、SNMPV2c ゲートウェイに対して **public** です。
- **AUTH_PROTOCOL** である **auth-protocol** は、SNMPV3 ゲートウェイでは必須であり、デフォルトでは **SHA** になります。
- SNMPV3 ゲートウェイでは、**PRIVACY_PROTOCOL** である **privacy-protocol** は必須となります。
- **PRIVACY_PASSWORD** は、暗号化された SNMPV3 ゲートウェイでは必須となります。
- **SNMP_V3_AUTH_USER_NAME** はユーザー名であり、SNMPV3 ゲートウェイでは必須となります。
- **SNMP_V3_AUTH_PASSWORD** はパスワードであり、SNMPV3 ゲートウェイでは必須となります。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- ノードへのルートレベルのアクセス。
- Red Hat Enterprise Linux システムに **firewalld** をインストールします。

手順

1. SNMP 管理ホストで、SNMP パッケージをインストールします。

例

■

```
[root@host01 ~]# dnf install -y net-snmp-utils net-snmp
```

- SNMP のポート 162 を開いて、アラートを受信します。

例

```
[root@host01 ~]# firewall-cmd --zone=public --add-port=162/udp
[root@host01 ~]# firewall-cmd --zone=public --add-port=162/udp --permanent
```

- 管理情報ベース (MIB) を実装して、SNMP 通知を理解し、宛先ホストでの SNMP サポートを強化します。メインリポジトリから raw ファイルをコピーします:

<https://github.com/ceph/ceph/blob/master/monitoring/snmp/CEPH-MIB.txt>

例

```
[root@host01 ~]# curl -o CEPH_MIB.txt -L
https://raw.githubusercontent.com/ceph/ceph/master/monitoring/snmp/CEPH-MIB.txt
[root@host01 ~]# scp CEPH_MIB.txt root@host02:/usr/share/snmp/mibs
```

- snmptrapd** ディレクトリーを作成します。

例

```
[root@host01 ~]# mkdir /root/snmptrapd/
```

- SNMP バージョンに基づいて、各プロトコルの **snmptrapd** ディレクトリーに設定ファイルを作成します。

構文

```
format2 %V\n% Agent Address: %A \n Agent Hostname: %B \n Date: %H - %J - %K - %L -
%M - %Y \n Enterprise OID: %N \n Trap Type: %W \n Trap Sub-Type: %q \n
Community/Infosec Context: %P \n Uptime: %T \n Description: %W \n PDU Attribute/Value
Pair Array:\n%v \n ----- \n
createuser -e 0x_ENGINE_ID_ SNMPV3_AUTH_USER_NAME AUTH_PROTOCOL
SNMP_V3_AUTH_PASSWORD PRIVACY_PROTOCOL PRIVACY_PASSWORD
authuser log,execute SNMP_V3_AUTH_USER_NAME
authCommunity log,execute,net SNMP_COMMUNITY_FOR_SNMPV2
```

- SNMPV2c の場合、以下のように **snmptrapd_public.conf** ファイルを作成します。

例

```
format2 %V\n% Agent Address: %A \n Agent Hostname: %B \n Date: %H - %J - %K -
%L - %M - %Y \n Enterprise OID: %N \n Trap Type: %W \n Trap Sub-Type: %q \n
Community/Infosec Context: %P \n Uptime: %T \n Description: %W \n PDU
Attribute/Value Pair Array:\n%v \n ----- \n
authCommunity log,execute,net public
```

ここでの **public** 設定は、**snmp-gateway** サービスをデプロイするときに使用される **snmp_community** 設定と一致する必要があります。

- 認証のみのSNMPV3の場合は、以下のように **snmptrapd_auth.conf** ファイルを作成します。

例

```
format2 %V\n% Agent Address: %A \n Agent Hostname: %B \n Date: %H - %J - %K -
%L - %M - %Y \n Enterprise OID: %N \n Trap Type: %W \n Trap Sub-Type: %q \n
Community/Infosec Context: %P \n Uptime: %T \n Description: %W \n PDU
Attribute/Value Pair Array:\n%v \n ----- \n
createuser -e 0x8000C53Ff64f341c655d11eb8778fa163e914bcc myuser SHA
mypassword
authuser log,execute myuser
```

0x8000C53Ff64f341c655d11eb8778fa163e914bcc 文字列は **engine_id** で、**myuser** と **mypassword** は認証情報です。パスワードのセキュリティーは、**SHA** アルゴリズムによって定義されます。

これは、**snmp-gateway** デーモンをデプロイするための設定に対応します。

例

```
snmp_v3_auth_username: myuser
snmp_v3_auth_password: mypassword
```

- 認証と暗号化を使用するSNMPV3の場合、以下のように **snmptrapd_authpriv.conf** ファイルを作成します。

例

```
format2 %V\n% Agent Address: %A \n Agent Hostname: %B \n Date: %H - %J - %K -
%L - %M - %Y \n Enterprise OID: %N \n Trap Type: %W \n Trap Sub-Type: %q \n
Community/Infosec Context: %P \n Uptime: %T \n Description: %W \n PDU
Attribute/Value Pair Array:\n%v \n ----- \n
createuser -e 0x8000C53Ff64f341c655d11eb8778fa163e914bcc myuser SHA
mypassword DES mysecret
authuser log,execute myuser
```

0x8000C53Ff64f341c655d11eb8778fa163e914bcc 文字列は **engine_id** で、**myuser** と **mypassword** は認証情報です。パスワードセキュリティーは **SHA** アルゴリズムで定義され、**DES** はプライバシー暗号化のタイプになります。

これは、**snmp-gateway** デーモンをデプロイするための設定に対応します。

例

```
snmp_v3_auth_username: myuser
snmp_v3_auth_password: mypassword
snmp_v3_priv_password: mysecret
```

6. SNMP 管理ホストでデーモンを実行します。

構文

```
/usr/sbin/snmptrapd -M /usr/share/snmp/mibs -m CEPH-MIB.txt -f -C -c
/root/snmptrapd/CONFIGURATION_FILE -Of -Lo :162
```

例

```
[root@host01 ~]# /usr/sbin/snmptrapd -M /usr/share/snmp/mibs -m CEPH-MIB.txt -f -C -c
/root/snmptrapd/snmptrapd_auth.conf -Of -Lo :162
```

- ストレージクラスターでアラートがトリガーされた場合は、SNMP 管理ホストで出力を監視できます。SNMP トラップと、MIB によってデコードされたトラップを確認します。

例

```
NET-SNMP version 5.8
Agent Address: 0.0.0.0
Agent Hostname: <UNKNOWN>
Date: 15 - 5 - 12 - 8 - 10 - 4461391
Enterprise OID: .
Trap Type: Cold Start
Trap Sub-Type: 0
Community/Infosec Context: TRAP2, SNMP v3, user myuser, context
Uptime: 0
Description: Cold Start
PDU Attribute/Value Pair Array:
.iso.org.dod.internet.mgmt.mib-2.1.3.0 = Timeticks: (292276100) 3 days, 19:52:41.00
.iso.org.dod.internet.snmpV2.snmpModules.1.1.4.1.0 = OID:
.iso.org.dod.internet.private.enterprises.ceph.cephCluster.cephNotifications.prometheus.promM
gr.promMgrPrometheusInactive
.iso.org.dod.internet.private.enterprises.ceph.cephCluster.cephNotifications.prometheus.promM
gr.promMgrPrometheusInactive.1 = STRING:
"1.3.6.1.4.1.50495.1.2.1.6.2[alertname=CephMgrPrometheusModuleInactive]"
.iso.org.dod.internet.private.enterprises.ceph.cephCluster.cephNotifications.prometheus.promM
gr.promMgrPrometheusInactive.2 = STRING: "critical"
.iso.org.dod.internet.private.enterprises.ceph.cephCluster.cephNotifications.prometheus.promM
gr.promMgrPrometheusInactive.3 = STRING: "Status: critical"
- Alert: CephMgrPrometheusModuleInactive
Summary: Ceph's mgr/prometheus module is not available
Description: The mgr/prometheus module at 10.70.39.243:9283 is unreachable. This could
mean that the module has been disabled or the mgr itself is down.
Without the mgr/prometheus module metrics and alerts will no longer function. Open a shell
to ceph and use 'ceph -s' to determine whether the mgr is active. If the mgr is not active,
restart it, otherwise you can check the mgr/prometheus module is loaded with 'ceph mgr
module ls' and if it's not listed as enabled, enable it with 'ceph mgr module enable
prometheus'"
```

上記の例では、Prometheus モジュールが無効になった後にアラートが生成されます。

関連情報

- Red Hat Ceph Storage Operations Guide の [Deploying the SNMP gateway](#) セクションを参照してください。

13.3. SNMP ゲートウェイのデプロイ

SNMPV2c または SNMPV3 のいずれかを使用して、Simple Network Management Protocol (SNMP) ゲートウェイをデプロイできます。SNMP ゲートウェイをデプロイするには、次の2つの方法があります。

1. 認証情報ファイルを作成する方法
2. すべての詳細を含む1つのサービス設定yaml ファイルを作成する方法。

以下のパラメーターを使用して、バージョンに基づいてSNMP ゲートウェイをデプロイできます。

- **service_type** は **snmp-gateway** です。
- **service_name** は、任意のユーザー定義の文字列です。
- **count** は、ストレージクラスターにデプロイされるSNMP ゲートウェイの数です。
- **snmp_destination** パラメーターは、hostname:port の形式である必要があります。
- **engine-id** は、デバイスの一意的識別子(16進数)であり、SNMPV3 ゲートウェイで必要とされています。Red Hat では、このパラメーターに ``8000C53F_CLUSTER_FSID_WITHOUT_DASHES_`` を使用することをお勧めしています。
- **snmp_community** パラメーターは、SNMPV2c ゲートウェイに対して **public** です。
- **auth-protocol** はSNMPV3 ゲートウェイでは必須であり、デフォルトでは **SHA** になります。
- **privacy-protocol** では、認証と暗号化を備えたSNMPV3 ゲートウェイが必須となります。
- デフォルトでは、ポートは **9464** です。
- シークレットとパスワードをオーケストレーターに渡すには、**-i FILENAME** を指定する必要があります。

SNMP ゲートウェイサービスがデプロイまたは更新されると、Prometheus Alertmanager 設定が自動的に更新され、objectidentifier を持つアラートがSNMP ゲートウェイデーモンに転送されてさらに処理されます。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- ノードへのルートレベルのアクセス。
- SNMP 管理ホストである宛先ホストで **snmptrapd** を設定します。

手順

1. Cephadm シェルにログインします。

例

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

2. SNMP ゲートウェイをデプロイする必要があるホストのラベルを作成します。

構文

```
ceph orch host label add HOSTNAME snmp-gateway
```

例

```
[ceph: root@host01 /]# ceph orch host label add host02 snmp-gateway
```

3. SNMP のバージョンに応じて、認証情報ファイルまたはサービス設定ファイルを作成します。

- SNMPV2c の場合、以下のようにファイルを作成します。

例

```
[ceph: root@host01 /]# cat snmp_creds.yml  
  
snmp_community: public
```

あるいは

例

```
[ceph: root@host01 /]# cat snmp-gateway.yml  
  
service_type: snmp-gateway  
service_name: snmp-gateway  
placement:  
  count: 1  
spec:  
  credentials:  
    snmp_community: public  
  port: 9464  
  snmp_destination: 192.168.122.73:162  
  snmp_version: V2c
```

- 認証のみの SNMPV3 の場合は、以下のようにファイルを作成します。

例

```
[ceph: root@host01 /]# cat snmp_creds.yml  
  
snmp_v3_auth_username: myuser  
snmp_v3_auth_password: mypassword
```

あるいは

例

```
[ceph: root@host01 /]# cat snmp-gateway.yml  
  
service_type: snmp-gateway  
service_name: snmp-gateway  
placement:  
  count: 1  
spec:
```



```

credentials:
  snmp_v3_auth_password: mypassword
  snmp_v3_auth_username: myuser
  engine_id: 8000C53F64f341c655d11eb8778fa163e914bcc
  port: 9464
  snmp_destination: 192.168.122.1:162
  snmp_version: V3

```

- 認証と暗号化を使用する SNMPV3 の場合、以下のようにファイルを作成します。

例

```
[ceph: root@host01 /]# cat snmp_creds.yml
```

```

snmp_v3_auth_username: myuser
snmp_v3_auth_password: mypassword
snmp_v3_priv_password: mysecret

```

あるいは

例

```
[ceph: root@host01 /]# cat snmp-gateway.yml
```

```

service_type: snmp-gateway
service_name: snmp-gateway
placement:
  count: 1
spec:
  credentials:
    snmp_v3_auth_password: mypassword
    snmp_v3_auth_username: myuser
    snmp_v3_priv_password: mysecret
  engine_id: 8000C53F64f341c655d11eb8778fa163e914bcc
  port: 9464
  snmp_destination: 192.168.122.1:162
  snmp_version: V3

```

4. **ceph orch** コマンドを実行します。

構文

```

ceph orch apply snmp-gateway --snmp_version=V2c_OR_V3 --
destination=SNMP_DESTINATION [--port=PORT_NUMBER]\
[--engine-id=8000C53F_CLUSTER_FSID_WITHOUT_DASHES_] [--auth-
protocol=MDS_OR_SHA] [--privacy_protocol=DES_OR_AES] -i FILENAME

```

あるいは

構文

```
ceph orch apply -i FILENAME.yml
```

- SNMPV2c の場合、**snmp_creds** ファイルを使用して、**snmp-version** を **V2c** として指定して、**ceph orch** コマンドを実行します。

例

```
[ceph: root@host01 /]# ceph orch apply snmp-gateway --snmp-version=V2c --destination=192.168.122.73:162 --port=9464 -i snmp_creds.yml
```

- SNMPV3 で認証のみの場合、**snmp_creds** ファイルを用いて、**snmp-version** を **V3** および **engine-id** として指定して、**ceph orch** コマンドを実行します。

例

```
[ceph: root@host01 /]# ceph orch apply snmp-gateway --snmp-version=V3 --engine-id=8000C53Ff64f341c655d11eb8778fa163e914bcc--destination=192.168.122.73:162 -i snmp_creds.yml
```

- 認証と暗号化を使用する SNMPV3 の場合、**snmp_creds** ファイルを使用して、**snmp-version** を **V3**、**privacy-protocol**、および **engine-id** として指定して、**ceph orch** コマンドを実行します。

例

```
[ceph: root@host01 /]# ceph orch apply snmp-gateway --snmp-version=V3 --engine-id=8000C53Ff64f341c655d11eb8778fa163e914bcc--destination=192.168.122.73:162 --privacy-protocol=AES -i snmp_creds.yml
```

あるいは

- すべての SNMP バージョンの場合、**snmp-gateway** ファイルを使用して、以下のコマンドを実行します。

例

```
[ceph: root@host01 /]# ceph orch apply -i snmp-gateway.yml
```

関連情報

- Red Hat Ceph Storage Operations Guide の [Configuring `snmptrapd`](#) のセクションを参照してください。

第14章 ノードの障害の処理

ストレージクラスター内でノード全体に障害が発生する可能性があります。ストレージ管理者が行うノード障害の処理は、ディスク障害の処理と同様です。ノードの障害としてCephが1つのディスクに対してのみ配置グループ(PG)を復元する代わりに、そのノード内のディスクのすべてのPGを復元する必要があります。CephはOSDがすべてダウンしていることを検出し、自己修復として知られる復元プロセスを自動的に開始します。

ノードの障害シナリオは3つあります。ノードを置き換える際の各シナリオにおけるハイレベルのワークフローを以下に示します。

- ノードの置き換えには、失敗したノードからroot ディスクおよびCeph OSD ディスクを使用します。
 1. バックフィルを無効にします。
 2. ノードを置き換え、古いノードからディスクを取得し、それらを新規ノードに追加します。
 3. バックフィルを有効にします。
- ノードを置き換え、オペレーティングシステムを再インストールし、障害が発生したノードからCeph OSD ディスクを使用します。
 1. バックフィルを無効にします。
 2. Ceph 設定のバックアップを作成します。
 3. ノードを置き換え、障害が発生したノードからCeph OSD ディスクを追加します。
 4. ディスクをJBODとして設定
 5. オペレーティングシステムをインストールします。
 6. Ceph の設定を復元します。
 7. Ceph Orchestrator コマンドを使用して新規ノードをストレージクラスターに追加します。Ceph デーモンが自動的に対応するノードに配置されます。
 8. バックフィルを有効にします。
- ノードを置き換え、オペレーティングシステムを再インストールし、すべての新規Ceph OSD ディスクを使用します。
 1. バックフィルを無効にします。
 2. 障害のあるノードのすべてのOSD をストレージクラスターから削除します。
 3. Ceph 設定のバックアップを作成します。
 4. ノードを置き換え、障害が発生したノードからCeph OSD ディスクを追加します。
 - a. ディスクをJBODとして設定
 5. オペレーティングシステムをインストールします。
 6. Ceph Orchestrator コマンドを使用して新規ノードをストレージクラスターに追加します。Ceph デーモンが自動的に対応するノードに配置されます。

- 7. バックフィルを有効にします。

14.1. 前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- 障害のあるノード。

14.2. ノードの追加または削除前の考慮事項

Ceph の未処理の機能の1つは、ランタイム時に Ceph OSD ノードを追加または削除できる機能です。つまり、ストレージクラスターの容量のサイズを変更したり、ストレージクラスターを縮小せずにハードウェアを置き換えることができることを意味します。

ストレージクラスターの状態が劣化 (**degraded**) している間に Ceph クライアントを提供する機能にも運用上の利点があります。たとえば、残業や週末ではなく、通常の営業時間内にハードウェアを追加、削除、または交換できます。ただし、Ceph OSD ノードの追加および削除により、パフォーマンスに大きな影響を与える可能性があります。

Ceph OSD ノードを追加または削除する前に、ストレージクラスターのパフォーマンスへの影響を考慮してください。

- ストレージクラスターの容量を拡張または縮小するか、Ceph OSD ノードを追加または削除することで、ストレージクラスターのリバランスとしてバックフィルを予測します。このリバランス期間中に、Ceph は追加のリソースを使用します。これにより、ストレージクラスターのパフォーマンスに影響する可能性があります。
- 実稼働用 Ceph Storage クラスターでは、Ceph OSD ノードに特定のタイプのストレージストラテジーを容易にする特定のハードウェア設定があります。
- Ceph OSD ノードは CRUSH 階層の一部であるため、ノードの追加または削除のパフォーマンスへの影響は通常 CRUSH ルールセットを使用するプールのパフォーマンスに影響します。

関連情報

- 詳細は、Red Hat Ceph Storage の『[Storage Strategies Guide](#)』を参照してください。

14.3. パフォーマンスに関する考慮事項

以下の要素は通常、Ceph OSD ノードの追加時または削除時のストレージクラスターのパフォーマンスに影響します。

- Ceph クライアントは、Ceph への I/O インターフェースの負荷を配置します。つまり、クライアントはプールに負荷を置きます。プールは CRUSH ルールセットにマップします。基礎となる CRUSH 階層により、Ceph は障害ドメインにデータを配置できます。基礎となる Ceph OSD ノードにクライアント負荷が高いプールが必要な場合、クライアントの負荷は復元時間に大きな影響を及ぼし、パフォーマンスが低下する可能性があります。書き込み操作には持続性のためにデータのレプリケーションが必要になるため、特に書き込み集約型クライアント負荷は、ストレージクラスターを回復するのに要する時間が長くなる可能性があります。
- 通常、追加または削除している容量は、クラスターの復元に影響を及ぼします。さらに、追加または削除するノードのストレージの密度も復元時間に影響を及ぼす可能性があります。たとえば、36 OSD を持つノードは、通常、12 OSD を持つノードよりも復元にかかる時間が長くなります。

- ノードを削除するには、十分な容量を確保して、**完全な比率**または**ほぼ完全比率**に到達しないようにします。ストレージクラスターが**フル比率**になると、Ceph は書き込み動作を一時停止してデータの損失を防ぎます。
- Ceph OSD ノードは、少なくとも1つのCeph CRUSH 階層にマッピングし、階層は少なくとも1つのプールにマップされます。CRUSH ルールセットを使用する各プールは、Ceph OSD ノードが追加または削除される際にパフォーマンスに影響が出ます。
- レプリケーションプールは、データのディープコピーを複製するネットワーク帯域幅を使用する傾向がありますが、イレイジャーコーディングプールはより多くのCPU を使用して **k+m** コーディングのチャンクを計算する傾向があります。データに存在するより多くのコピーで、ストレージクラスターを回復するのにかかる時間が長くなります。たとえば、大きなプールまたは **k+m** チャンクの数が多い場合は、同じデータのコピー数が少ないレプリケーションプールよりも復元にかかる時間が長くなります。
- ドライブ、コントローラー、およびネットワークインターフェースカードはすべて、復元時間に影響を与える可能性があるスループットの特性を持ちます。一般に、10 Gbps やSSD などのスループット特性が高いノードは、1Gbps やSATA ドライブなどのスループット特性が低いノードよりも迅速に回復します。

14.4. ノードの追加または削除に関する推奨事項

Red Hat は、ノード内の一度に1つのOSD を追加または削除して、次のOSD に進む前にストレージクラスターを回復させることを推奨します。これは、ストレージクラスターのパフォーマンスへの影響を最小限に抑えるのに役立ちます。ノードに障害が発生した場合は、一度に1つのOSD ではなく、ノード全体を一度に変更する必要がある場合があります。

OSD を削除するには、以下を実行します。

- [Ceph Orchestrator を使用した OSD デーモンの削除](#)

OSD を追加するには、以下を実行します。

- [利用可能なすべてのデバイスへの Ceph OSD のデプロイ](#)
- [高度なサービス指定を使用した Ceph OSD のデプロイ](#)
- [特定のデバイスおよびホストへの Ceph OSD のデプロイ](#)

Ceph OSD ノードを追加または削除する場合は、他の継続中のプロセスがストレージクラスターのパフォーマンスにも影響することを検討してください。クライアントI/O への影響を減らすために、Red Hat では以下を推奨します。

容量の計算

Ceph OSD ノードを削除する前に、ストレージクラスターがそのすべてのOSD の内容を**完全な比率**に到達せずにバックフィルするようにしてください。**フル比率**に達すると、ストレージクラスターは書き込み操作を拒否するようになります。

スクラビングを一時的に無効にする

スクラビングはストレージクラスターのデータの持続性を確保するために不可欠ですが、リソース集約型です。Ceph OSD ノードを追加または削除する前に、スクラビングおよびディープスクラビングを無効にして、現在のスクラビング操作を完了してから続行します。

```
ceph osd set noscrub
ceph osd set nodeep-scrub
```

Ceph OSD ノードを追加または削除すると、ストレージクラスターが **active+clean** 状態に戻り、**noscrub** および **nodeep-scrub** の設定を解除します。

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

バックフィルと復元の制限

妥当なデータの永続性がある場合は、パフォーマンスが劣化 (**degraded**) した状態での動作に問題はありません。たとえば、**osd_pool_default_size = 3** および **osd_pool_default_min_size = 2** を使用してストレージクラスターを操作できます。可能な限り早い復元時間用にストレージクラスターを調整することができますが、これにより Ceph クライアントの I/O パフォーマンスが大幅に影響を受ける可能性があります。最大の Ceph クライアント I/O パフォーマンスを維持するには、バックフィルと復元の操作を制限し、その操作に長い時間がかかる可能性があります。

```
osd_max_backfills = 1
osd_recovery_max_active = 1
osd_recovery_op_priority = 1
```

osd_recovery_sleep などの sleep パラメーターおよび delay パラメーターを設定することもできます。

配置グループの数を増やす

最後に、ストレージクラスターのサイズを拡張する場合は、配置グループの数を増やすことが必要となる場合があります。配置グループの数を拡張する必要がある場合、Red Hat はプレースメントグループの数を段階的に増やすことをお勧めします。配置グループの数を大幅に増やすと、パフォーマンスが大幅に低下します。

14.5. CEPH OSD ノードの追加

Red Hat Ceph Storage クラスターの容量を拡張するには、OSD ノードを追加します。

前提条件

- Red Hat Ceph Storage クラスターが実行中である。
- ネットワーク接続が割り当てられたプロビジョニングされたノード

手順

1. ストレージクラスターの他のノードが、短縮ホスト名で新規ノードに到達できることを確認します。
2. スクラビングを一時的に無効にします。

例

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

3. バックフィルおよび復元機能を制限します。

構文

-

```
ceph tell DAEMON_TYPE.* injectargs --OPTION_NAME VALUE [--OPTION_NAME VALUE]
```

例

```
[ceph: root@host01 /]# ceph tell osd.* injectargs --osd-max-backfills 1 --osd-recovery-max-active 1 --osd-recovery-op-priority 1
```

4. クラスターの SSH 公開鍵をフォルダーに展開します。

構文

```
ceph cephadm get-pub-key > ~/PATH
```

例

```
[ceph: root@host01 /]# ceph cephadm get-pub-key > ~/ceph.pub
```

5. ceph クラスターの SSH 公開鍵を、新たなホストの root ユーザーの **authorized_keys** ファイルにコピーします。

構文

```
ssh-copy-id -f -i ~/PATH root@HOST_NAME_2
```

例

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

6. 新規ノードを CRUSH マップに追加します。

構文

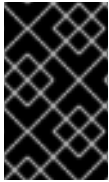
```
ceph orch host add NODE_NAME IP_ADDRESS
```

例

```
[ceph: root@host01 /]# ceph orch host add host02 168.20.20.2
```

7. ノードの各ディスクの OSD をストレージクラスターに追加します。

- [利用可能なすべてのデバイスへの Ceph OSD のデプロイ](#)
- [高度なサービス指定を使用した Ceph OSD のデプロイ](#)
- [特定のデバイスおよびホストへの Ceph OSD のデプロイ](#)



重要

OSD ノードを Red Hat Ceph Storage クラスタに追加する場合、Red Hat は、1 度に 1 つの OSD デーモンを追加し、次の OSD に進む前にクラスタを **active+clean** 状態に復元できるようにすることを推奨します。

関連情報

- 詳細は、『Red Hat Ceph Storage Configuration Guide』の「[Setting a Specific Configuration Setting at Runtime](#)」セクションを参照してください。
- CRUSH 階層の適切な場所にノードを配置するための詳細は、Red Hat Ceph Storage の『Storage Strategies Guide』の「[Adding a Bucket](#)」および「[Moving a Bucket](#)」セクションを参照してください。

14.6. CEPH OSD ノードの削除

ストレージクラスタの容量を減らすには、OSD ノードを削除します。



警告

Ceph OSD ノードを削除する前に、ストレージクラスタが**完全な比率**に到達せずすべての OSD の内容をバックフィルするようにしてください。**フル比率**に達すると、ストレージクラスタは書き込み操作を拒否するようになります。

前提条件

- Red Hat Ceph Storage クラスタが実行中である。
- ストレージクラスタ内のすべてのノードへの root レベルのアクセス。

手順

1. ストレージクラスタの容量を確認します。

構文

```
ceph df
rados df
ceph osd df
```

2. スクラビングを一時的に無効にします。

構文

```
ceph osd set noscrub
ceph osd set nodeep-scrub
```

3. バックフィルおよび復元機能を制限します。

構文

```
ceph tell DAEMON_TYPE.* injectargs --OPTION_NAME VALUE [--OPTION_NAME VALUE]
```

例

```
[ceph: root@host01 /]# ceph tell osd.* injectargs --osd-max-backfills 1 --osd-recovery-max-active 1 --osd-recovery-op-priority 1
```

4. ノード上の各 OSD をストレージクラスターから削除します。

- [Ceph Orchestrator を使用した OSD デーモンの削除](#)



重要

ストレージクラスターから OSD ノードを削除する場合、Red Hat は、ノード内の一度に1つの OSD を削除してから、次の OSD を削除する前にクラスターが **active+clean** 状態に回復できるようにすることを推奨します。

- a. OSD を削除したら、ストレージクラスターが **ほぼ完全比率** に達していないことを確認します。

構文

```
ceph -s  
ceph df
```

- b. ノードのすべての OSD がストレージクラスターから削除されるまでこの手順を繰り返します。
5. すべての OSD が削除されたら、ホストを削除します。

- [Ceph Orchestrator を使用したホストの削除](#)。

関連情報

- 詳細は、『Red Hat Ceph Storage Configuration Guide』の「[Setting a specific configuration setting at runtime](#)」セクションを参照してください。

14.7. ノードの障害のシミュレーション

ハードノードの障害をシミュレーションするには、ノードの電源をオフにし、オペレーティングシステムを再インストールします。

前提条件

- 正常かつ実行中の Red Hat Ceph Storage クラスター
- ストレージクラスター内のすべてのノードへの root レベルのアクセス。

手順

1. ストレージクラスターの容量を確認し、ノードの削除への影響を確認します。

例

```
[ceph: root@host01 /]# ceph df
[ceph: root@host01 /]# rados df
[ceph: root@host01 /]# ceph osd df
```

2. 必要に応じて、復元およびバックフィルを無効にします。

例

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

3. ノードをシャットダウンします。
4. ホスト名を変更する場合は、CRUSH マップからノードを削除します。

例

```
[ceph: root@host01 /]# ceph osd crush rm host03
```

5. ストレージクラスターのステータスを確認します。

例

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

6. ノードにオペレーティングシステムを再インストールします。
7. 新しいノードを追加します。

- [Ceph Orchestrator を使用したホストの追加](#)

8. 必要に応じて、復元およびバックフィルを有効にします。

例

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

9. Ceph のヘルスを確認します。

例

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

関連情報

- 詳細は、[『Red Hat Ceph Storage インストールガイド』](#)を参照してください。

第15章 データセンター障害の処理

ストレージ管理者は、データセンター障害を回避するために予防措置を取ることができます。これには、以下の予防措置があります。

- データセンターインフラストラクチャーの設定
- CRUSH マップ階層内に障害ドメインの設定
- ドメイン内での障害ノードの指定

15.1. 前提条件

- 正常かつ実行中の Red Hat Ceph Storage クラスタ
- ストレージクラスター内のすべてのノードへの root レベルのアクセス。

15.2. データセンター障害の回避

データセンターインフラストラクチャーの設定

ストレッチクラスター内の各データセンターには、ローカル機能および依存関係を反映するために異なるストレージクラスターの設定を指定できます。データの保存に役立つデータセンター間でレプリケーションを設定します。1つのデータセンターに障害が発生しても、ストレージクラスターの他のデータセンターにデータのコピーが含まれます。

CRUSH マップ階層内での障害ドメインの設定

障害またはフェイルオーバーのドメインは、ストレージクラスター内のドメインの冗長コピーです。アクティブなドメインが失敗すると、障害ドメインはアクティブドメインになります。

デフォルトで、CRUSH マップはフラット階層内のストレージクラスターのすべてのノードを一覧表示します。ただし、最善の結果を得るには、CRUSH マップ内に論理階層構造を作成します。階層は、各ノードが属するドメインと、障害のあるドメインを含む、ストレージクラスター内のそれらのドメイン間の関係を指定します。階層内の各ドメインの障害ドメインを定義すると、ストレージクラスターの信頼性が向上します。

複数のデータセンターを含むストレージクラスターを計画する際には、CRUSH マップ階層内にノードを配置するため、1つのデータセンターが停止した場合には、残りのストレージクラスターは稼働し続けます。

ドメイン内での障害ノードの設計

ストレージクラスター内のデータに3方向のレプリケーションを使用する予定の場合には、障害ドメイン内のノードの場所を考慮してください。データセンター内で停止が発生した場合は、一部のデータが1つのコピーにのみ存在する可能性があります。このシナリオでは、2つのオプションがあります。

- 標準設定で、データは読み取り専用ステータスのままにします。
- ライブは、停止期間に1つのコピーのみを行います。

標準設定では、ノード間でのデータ配置のランダム性のため、すべてのデータが影響を受けるわけではありませんが、一部のデータは1つのコピーしか持つことができず、ストレージクラスターは読み取り専用モードに戻ります。ただし、一部のデータが1つのコピーにのみ存在する場合、ストレージクラスターは読み取り専用モードに戻ります。

15.3. データセンター障害の処理

Red Hat Ceph Storage は、ストレッチクラスターでデータセンターのいずれかを失うなど、インフラストラクチャーに非常に致命的な障害がある場合があります。標準のオブジェクトストアのユースケースでは、3つのデータセンターすべての構成は、それらの間にレプリケーションを設定して個別に実行できます。このシナリオでは、各データセンターのストレージクラスター設定は異なり、ローカルの機能と依存関係を反映する可能性があります。

配置階層の論理構造を考慮する必要があります。適切な CRUSH マップは使用でき、インフラストラクチャー内の障害ドメインの階層構造が反映されます。論理階層定義を使用すると、標準の階層定義を使用することではなく、ストレージクラスターの信頼性が向上します。障害ドメインは CRUSH マップで定義されます。デフォルトの CRUSH マップには、フラットな階層のすべてのノードが含まれます。ストレッチクラスターなどの3つのデータセンター環境では、ノードの配置は、1つのデータセンターが停止できるように管理する必要がありますが、ストレージクラスターは稼働したままです。データに対して3方向レプリケーションを使用する場合に、ノードがある障害について検討してください。

以下の例では、作成されるマップは6つの OSD ノードを持つストレージクラスターの初期設定から派生しています。この例では、すべてのノードが1つのディスクを持つため、1つの OSD しかありません。全ノードは、階層ツリーの標準ルートであるデフォルトのルート下に分類されます。2つの OSD に重みが割り当てられているため、これらの OSD は他の OSD よりも少ないデータチャンクを受け取ります。これらのノードは、最初の OSD ディスクよりも大きなディスクを持つ後から導入されました。これは、ノードのグループが失敗しているデータ配置には影響しません。

例

```
[ceph: root@host01 /]# ceph osd tree
ID WEIGHT TYPE NAME          UP/DOWN REWEIGHT PRIMARY-AFFINITY
-1 0.33554 root default
-2 0.04779  host host03
  0 0.04779  osd.0    up 1.00000    1.00000
-3 0.04779  host host02
  1 0.04779  osd.1    up 1.00000    1.00000
-4 0.04779  host host01
  2 0.04779  osd.2    up 1.00000    1.00000
-5 0.04779  host host04
  3 0.04779  osd.3    up 1.00000    1.00000
-6 0.07219  host host06
  4 0.07219  osd.4    up 0.79999    1.00000
-7 0.07219  host host05
  5 0.07219  osd.5    up 0.79999    1.00000
```

論理階層定義を使用してノードを同じデータセンターにグループ化すると、データの配置の成熟度を実行できます。root、datacenter、rack、row、およびhostの定義タイプにより、3つのデータセンターのストッククラスターで障害ドメインを反映させることができます。

- ノード host01 および host02 はデータセンター1 (DC1) にあります。
- ノード host03 および host05 はデータセンター2 (DC2) にあります。
- ノード host04 および host06 はデータセンター3 (DC3) にあります。
- すべてのデータセンターが同じ構造に属する (全DC)

ホストのすべての OSD はホスト定義に属しているため、変更は必要ありません。その他のすべての割り当ては、ストレージクラスターの実行時に以下によって調整できます。

- 以下のコマンドでバケット構造を定義します。

```
ceph osd crush add-bucket allDC root
ceph osd crush add-bucket DC1 datacenter
ceph osd crush add-bucket DC2 datacenter
ceph osd crush add-bucket DC3 datacenter
```

- CRUSH マップを変更して、ノードをこの構造内の適切な場所に移動します。

```
ceph osd crush move DC1 root=allDC
ceph osd crush move DC2 root=allDC
ceph osd crush move DC3 root=allDC
ceph osd crush move host01 datacenter=DC1
ceph osd crush move host02 datacenter=DC1
ceph osd crush move host03 datacenter=DC2
ceph osd crush move host05 datacenter=DC2
ceph osd crush move host04 datacenter=DC3
ceph osd crush move host06 datacenter=DC3
```

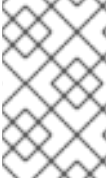
この構造内で、新しいホストや新しいディスクを追加することもできます。OSD を階層の右側に配置することにより、CRUSH アルゴリズムが冗長な部分を構造内の異なる障害ドメインに配置するように変更されます。

上記の例は、以下のようになります。

例

```
[ceph: root@host01 /]# ceph osd tree
ID WEIGHT TYPE NAME          UP/DOWN REWEIGHT PRIMARY-AFFINITY
-8 6.00000 root allDC
-9 2.00000 datacenter DC1
-4 1.00000 host host01
 2 1.00000 osd.2 up 1.00000 1.00000
-3 1.00000 host host02
 1 1.00000 osd.1 up 1.00000 1.00000
-10 2.00000 datacenter DC2
-2 1.00000 host host03
 0 1.00000 osd.0 up 1.00000 1.00000
-7 1.00000 host host05
 5 1.00000 osd.5 up 0.79999 1.00000
-11 2.00000 datacenter DC3
-6 1.00000 host host06
 4 1.00000 osd.4 up 0.79999 1.00000
-5 1.00000 host host04
 3 1.00000 osd.3 up 1.00000 1.00000
-1 0 root default
```

上記の一覧には、osd ツリーを表示することで、生成される CRUSH マップが表示されます。ホストがデータセンターにどのように属し、すべてのデータセンターが同じトップレベル構造に属しているかがわかりやすくなりましたが、場所が明確に区別されています。



注記

マップに応じてデータを適切な場所に配置すると、正常なクラスター内でのみ適切に機能します。一部の OSD が利用できない状況では、置き換えが発生する可能性があります。この誤差は、可能な場合は自動的に修正されます。

関連情報

- 詳細は、Red Hat Ceph Storage の『Storage Strategies Guide』の「[CRUSH administration](#)」の章を参照してください。