



SUSE Enterprise Storage 7

# 運用と管理ガイド

# 運用と管理ガイド

## SUSE Enterprise Storage 7


著者: Tomáš Bažant、Alexandra Settle、Liam Proven

発行日: 11/12/2025

<https://documentation.suse.com> 

Copyright © 2020–2025 SUSE LLC and contributors. All rights reserved.

特に明記されている場合を除き、本書はクリエイティブ・コモンズ表示-継承4.0国際(CC-BY-SA 4.0)に基づいてライセンスされています。 <https://creativecommons.org/licenses/by-sa/4.0/legalcode>  を参照してください。

SUSEの商標については、<http://www.suse.com/company/legal/> を参照してください。サードパーティ各社とその製品の商標は、所有者であるそれぞれの会社に所属します。商標記号(®、™など)は、SUSEおよび関連会社の商標を示します。アスタリスク(\*)は、第三者の商標を示します。

本書のすべての情報は、細心の注意を払って編集されています。しかし、このことは絶対に正確であることを保証するものではありません。SUSE LLC、その関係者、著者、翻訳者のいずれも誤りまたはその結果に対して一切責任を負いかねます。

# 目次

## このガイドについて xvii

- 1 利用可能なマニュアル xvii
- 2 フィードバックの提供 xviii
- 3 マニュアルの表記規則 xix
- 4 製品のライフサイクルとサポート xxi  
SUSEによるサポートの定義 xxi • SUSE Enterprise Storageのサポートステートメント xxi • 技術レビュー xxii
- 5 Cephの貢献者 xxiii
- 6 このガイドで使用するコマンドとコマンドプロンプト xxiii  
Salt関連のコマンド xxiii • Ceph関連のコマンド xxiv • 一般的なLinuxコマンド xxv • 追加情報 xxv

## I Cephダッシュボード 1

### 1 Cephダッシュボードについて 2

### 2 ダッシュボードのWebユーザインタフェース 3

- 2.1 ログイン 3
- 2.2 ユーティリティメニュー 4
- 2.3 メインメニュー 5
- 2.4 コンテンツペイン 6
- 2.5 Web UIの共通機能 6
- 2.6 ダッシュボードウィジェット 6  
ステータスウィジェット 7 • 容量のウィジェット 8 • パフォーマンスウィジェット 8

## **3 Cephダッシュボードユーザと役割の管理 10**

- 3.1 ユーザの一覧 10
- 3.2 新しいユーザの追加 10
- 3.3 ユーザの編集 11
- 3.4 ユーザの削除 11
- 3.5 ユーザの役割の一覧 12
- 3.6 カスタム役割の追加 12
- 3.7 カスタム役割の編集 14
- 3.8 カスタム役割の削除 14

## **4 クラスタの内部情報の表示 15**

- 4.1 クラスタノードの表示 15
- 4.2 クラスタのインベントリへのアクセス 15
- 4.3 Ceph Monitorの表示 16
- 4.4 サービスの表示 17
- 4.5 Ceph OSDの表示 18  
OSDの追加 21
- 4.6 クラスタ設定の表示 24
- 4.7 CRUSHマップの表示 24
- 4.8 マネージャモジュールの表示 25
- 4.9 ログの表示 26
- 4.10 監視の表示 26

## **5 プールの管理 27**

- 5.1 新しいプールの追加 28
- 5.2 プールの削除 28
- 5.3 プールのオプションの編集 28

## 6 RADOS Block Deviceの管理 30

- 6.1 RBDに関する詳細の表示 30
- 6.2 RBDの設定の表示 31
- 6.3 RBDの作成 32
- 6.4 RBDの削除 34
- 6.5 RADOS Block Deviceのスナップショットの作成 34
- 6.6 RBDミラーリング 35
  - プライマリクラスタとセカンダリクラスタの設定 36 • rbd-mirrorデーモンの有効化 36 • ミラーリングの無効化 37 • ピアのブートストラップ処理 38 • クラスタピアの削除 39 • Cephダッシュボードによるプールのレプリケーション設定 39 • RBDイメージレプリケーションが機能することの確認 43
- 6.7 iSCSI Gatewayの管理 46
  - iSCSIターゲットの追加 47 • iSCSIターゲットの編集 49 • iSCSIターゲットの削除 49
- 6.8 RBD QoS (サービス品質) 49
  - オプションのグローバルな設定 50 • 新しいプールでのオプションの設定 50 • 既存のプールでのオプションの設定 51 • 設定オプション 51 • 新しいRBDイメージを使用したRBD QoSオプションの作成 52 • 既存のイメージでのRBD QoSの編集 52 • イメージをコピーまたは複製する際の設定オプションの変更 52

## 7 NFS Ganeshaの管理 53

- 7.1 NFSエクスポートの作成 53
- 7.2 NFSエクスポートの削除 55
- 7.3 NFSエクスポートの編集 55

## 8 CephFSの管理 57

- 8.1 CephFSの概要の表示 57

## 9 Object Gatewayの管理 59

- 9.1 Object Gatewayの表示 59

- 9.2 Object Gatewayユーザの管理 60  
新しいゲートウェイユーザの追加 61 • ゲートウェイユーザの削除 63 • ゲートウェイユーザの詳細の編集 63
- 9.3 Object Gatewayバケットの管理 63  
新しいバケットの追加 63 • バケットの詳細の表示 64 • バケットの編集 64 • バケットの削除 65

## 10 手動設定 66

- 10.1 TLS/SSLサポートの設定 66  
自己署名証明書の作成 67 • CA署名証明書の使用 67
- 10.2 ホスト名とポート番号の変更 68
- 10.3 ユーザ名とパスワードの調整 69
- 10.4 Object Gateway管理フロントエンドの有効化 69
- 10.5 iSCSI管理の有効化 70
- 10.6 Single Sign-Onを有効にする 71

## 11 コマンドラインによるユーザと役割の管理 73

- 11.1 パスワードポリシーの管理 73
- 11.2 ユーザアカウントの管理 74
- 11.3 ユーザの役割と許可 74  
セキュリティスコープの定義 75 • ユーザの役割の指定 76
- 11.4 プロキシ設定 78  
リバースプロキシによるダッシュボードへのアクセス 79 • リダイレクションの無効化 79 • エラーステータスコードの設定 79 • HAProxyの設定例 79
- 11.5 API要求の監査 80
- 11.6 CephダッシュボードによるNFS Ganeshaの設定 81  
複数のNFS Ganeshaクラスタの設定 82
- 11.7 デバッグ用プラグイン 82

## II クラスタの運用 83

### 12 クラスタの状態の判断 84

- 12.1 クラスタの状態の確認 84
- 12.2 クラスタのヘルスの確認 86
- 12.3 クラスタの使用量統計の確認 95
- 12.4 OSDの状態の確認 97
- 12.5 満杯のOSDの確認 97
- 12.6 Monitorの状態の確認 98
- 12.7 配置グループの状態の確認 99
- 12.8 ストレージの容量 99
- 12.9 OSDと配置グループの監視 101
  - OSDの監視 102 • 配置グループセットの割り当て 103 • ピアリング 104 • 配置グループの状態の監視 105 • オブジェクトの場所の検索 110

### 13 運用タスク 112

- 13.1 クラスタ設定の変更 112
- 13.2 ノードの追加 112
- 13.3 ノードの削除 113
- 13.4 OSDの管理 115
  - ディスクデバイスの一覧 115 • ディスクデバイスの消去 115 • DriveGroups仕様を用いたOSDの追加 116 • OSDの削除 125 • OSDの交換 126
- 13.5 新しいノードへのSalt Masterの移動 127
- 13.6 クラスタノードの更新 129
  - ソフトウェアリポジトリ 129 • リポジトリのステージング 129 • Cephサービスのダウンタイム 129 • 更新の実行 129
- 13.7 Cephの更新 130
  - 更新の開始 130 • 更新の監視 130 • 更新のキャンセル 131

13.8 クラスタの停止または再起動 131

13.9 Cephクラスタ全体の削除 132

## 14 Cephサービスの運用 133

14.1 個別のサービスの運用 133

14.2 サービスタイプの運用 134

14.3 単一のノードでサービスを運用する 134

サービスとターゲットの確認 134 • ノード上のすべてのサービスの運用 135 • ノード上の個別のサービスの運用 135 • サービス状態のクエリ 136

14.4 Cephクラスタ全体のシャットダウンと再起動 136

## 15 バックアップおよび復元 138

15.1 クラスタ設定とデータのバックアップ 138

ceph-salt設定のバックアップ 138 • Ceph設定のバックアップ 138 • Salt設定のバックアップ 138 • カスタム設定のバックアップ 139

15.2 Cephノードの復元 139

## 16 監視とアラート 141

16.1 カスタムイメージまたはローカルイメージの設定 142

16.2 監視サービスのアップデート 143

16.3 監視の無効化 144

16.4 Grafanaの設定 144

16.5 Prometheus Manager Moduleの設定 145

ネットワークインタフェースの設定 145 • scrape\_intervalの設定 146 • キャッシュの設定 146 • RBDイメージ監視の有効化 147

16.6 Prometheusのセキュリティモデル 147

16.7 Prometheus Alertmanager SNMPウェブフック 148

prometheus-webhook-snmpサービスの設定 149 • Prometheus AlertmanagerのSNMP用設定 150

### III クラスタへのデータ保存 151

## 17 保存データの管理 152

### 17.1 OSDデバイス 153

デバイスクラス 153

### 17.2 バケット 160

### 17.3 ルールセット 163

ノードツリーの反復処理 165 • firstnとindep 167

### 17.4 配置グループ 168

配置グループの使用 168 • PG\_NUMの値の決定 170 • 配置グループ数の設定 171 • 配置グループ数の確認 172 • クラスタの配置グループの統計情報の確認 172 • スタックしている配置グループの統計情報の確認 172 • 配置グループマップの検索 172 • 配置グループの統計情報の取得 173 • 配置グループのスクラブ 173 • 配置グループのバックフィルと回復の優先度の設定 173 • 失われたオブジェクトを元に戻す 174 • 配置グループの自動拡張の有効化 174

### 17.5 CRUSHマップの操作 175

CRUSHマップの編集 175 • OSDの追加または移動 177 • `ceph osd reweight`と`ceph osd crush reweight`の違い 177 • OSDの削除 178 • バケットの追加 178 • バケットの移動 179 • バケットの削除 179

### 17.6 配置グループのスクラブ 179

## 18 ストレージプールの管理 182

### 18.1 プールの作成 182

### 18.2 プールの一覧 184

### 18.3 プールの名前変更 184

### 18.4 プールの削除 184

### 18.5 その他の操作 185

プールとアプリケーションの関連付け 185 • プールのクォータの設定 186 • プールの統計情報の表示 186 • プールから値を取得 188 • プールに値を設定 189 • オブジェクトレプリカの数の設定 192

- 18.6 プールのマイグレーション 194  
制限 194 • キャッシュ層を使用した移行 195 • RBDイメージの移行 197
- 18.7 プールのスナップショット 198  
プールのスナップショットの作成 198 • プールのスナップショットの一覧 198 • プールのスナップショットの削除 198
- 18.8 データ圧縮 199  
圧縮の有効化 199 • プール圧縮オプション 199 • グローバル圧縮オプション 200
- 19 イレージャコーディングプール 202**
- 19.1 イレージャコーディングプールの前提条件 202
- 19.2 サンプルのイレージャコーディングプールの作成 202
- 19.3 イレージャコードプロファイル 203  
新しいイレージャコードプロファイルの作成 206 • イレージャコードプロファイルの削除 207 • イレージャコードプロファイルの詳細の表示 207 • イレージャコードプロファイルの一覧 208
- 19.4 イレージャコーディングプールをRADOS Block Deviceとしてマーク付け 208
- 20 RADOS Block Device 209**
- 20.1 Block Deviceのコマンド 209  
複製プールでのBlock Deviceイメージの作成 209 • イレージャコーディングプールでのBlock Deviceイメージの作成 210 • Block Deviceイメージの一覧 210 • イメージ情報の取得 211 • Block Deviceイメージのサイズの変更 211 • Block Deviceイメージの削除 211
- 20.2 マウントとアンマウント 211  
Cephユーザアカウントの作成 212 • ユーザ認証 212 • RADOS Block Deviceを使用するための準備 213 • **rbdmmap**: ブート時のRBDデバイスのマッピング 215 • RBDデバイスのサイズを増やす 216
- 20.3 スナップショット 216  
cephxの有効化と設定 217 • スナップショットの基本 217 • スナップショットの階層化 219

- 20.4 RBDイメージのミラーリング 223
  - プールの設定 224 • RBDイメージの設定 228 • ミラーリング状態の確認 233
- 20.5 キャッシュの設定 233
- 20.6 QoS設定 235
- 20.7 先読み設定 236
- 20.8 拡張機能 237
- 20.9 古いカーネルクライアントを使用したRBDのマッピング 239
- 20.10 Block DeviceとKubernetesの有効化 240
  - Kubernetes環境におけるCeph Block Deviceの利用 242

#### IV クラスタデータへのアクセス 246

### 21 Ceph Object Gateway 247

- 21.1 Object Gatewayの制約と命名の制限 247
  - バケットの制限 247 • 保存オブジェクトの制限 247 • HTTPヘッダの制限 248
- 21.2 Object Gatewayの展開 248
- 21.3 Object Gatewayサービスの操作 248
- 21.4 設定オプション 248
- 21.5 Object Gatewayのアクセスの管理 248
  - Object Gatewayへのアクセス 249 • S3およびSwiftアカウントの管理 251
- 21.6 HTTPフロントエンド 254
- 21.7 Object GatewayでのHTTPS/SSLの有効化 255
  - 自己署名証明書の作成 255 • SSLを使用するようにObject Gatewayを設定する 256
- 21.8 同期モジュール 256
  - 同期モジュールの設定 257 • ゾーンの同期 258 • ElasticSearch同期モジュール 259 • クラウド同期モジュール 262 • アーカイブ同期モジュール 267

- 21.9 LDAP 認証 267
  - 認証メカニズム 268 • 要件 268 • LDAP認証を使用するためのObject Gatewayの設定 269 • カスタム検索フィルタを使用したユーザアクセスの制限 269 • LDAP認証用アクセストークンの生成 270
- 21.10 バケットインデックスのシャーディング 271
  - バケットインデックスのリシャーディング 271 • 新しいバケットのバケットインデックスシャーディング 274
- 21.11 OpenStack Keystoneの統合 275
  - OpenStackの設定 275 • Ceph Object Gatewayの設定 276
- 21.12 プールの配置とストレージクラス 278
  - 配置ターゲットの表示 278 • ストレージクラス 279 • ゾーングループおよびゾーンの設定 279 • 配置のカスタマイズ 281 • ストレージクラスの使用 283
- 21.13 マルチサイトObject Gateway 283
  - 要件と前提 284 • マスタゾーンの設定 284 • セカンダリゾーンの設定 290 • Object Gatewayの一般的な保守 296 • フェールオーバーと障害復旧機能を提供 297
- 22 Ceph iSCSI Gateway 299**
  - 22.1 ceph-iscsi管理対象ターゲット 299
    - open-iscsiに接続 299 • Microsoft Windows (Microsoft iSCSIイニシエータ)に接続 302 • VMwareの接続 310
  - 22.2 結論 315
- 23 クラスタファイルシステム 316**
  - 23.1 CephFSのマウント 316
    - クライアントの準備 316 • シークレットファイルの作成 317 • CephFSのマウント 317
  - 23.2 CephFSのアンマウント 319
  - 23.3 /etc/fstabでのCephFSのマウント 319

- 23.4 複数のアクティブMDSデーモン(アクティブ-アクティブMDS) 319  
アクティブ-アクティブMDSの使用 319 • MDSのアクティブクラスタサイズの増加 320 • ランク数の減少 320 • ランクへのディレクトリツリーの手動固定 321
- 23.5 フェールオーバーの管理 322  
スタンバイ再生の設定 322
- 23.6 CephFSのクォータの設定 322  
CephFSのクォータの制限 322 • CephFSのクォータの設定 323
- 23.7 CephFSスナップショットの管理 324  
スナップショットの作成 325 • スナップショットの削除 326
- 24 Sambaを介したCephデータのエクスポート 327**
  - 24.1 Samba共有を介したCephFSのエクスポート 327  
Sambaパッケージの設定とエクスポート 327 • ゲートウェイが1つの場合の例 327 • 高可用性の設定 330
  - 24.2 SambaゲートウェイとActive Directoryの参加 336  
Sambaのインストール準備 336 • DNSの検証 337 • SRVレコードの解決 337 • Kerberos の設定 338 • ローカルホスト名の解決 338 • Sambaの設定 339 • Active Directoryドメインへの参加 342 • ネームサービススイッチの設定 342 • サービスの起動 343 • winbindd接続のテスト 343
- 25 NFS Ganesha 345**
  - 25.1 NFSサービスの作成 346
  - 25.2 NFS Ganeshaの起動または再起動 346
  - 25.3 NFS回復プールのオブジェクトの一覧 347
  - 25.4 NFSエクスポートの作成 347
  - 25.5 NFSエクスポートの確認 347
  - 25.6 NFSエクスポートのマウント 348
  - 25.7 複数のNFS Ganeshaクラスタ 348

## V 仮想化ツールとの統合 349

### 26 libvirtとCeph 350

- 26.1 libvirtで使用するためのCephの設定 350
- 26.2 VMマネージャの準備 351
- 26.3 VMの作成 352
- 26.4 VMの設定 352
- 26.5 まとめ 355

### 27 QEMU KVMインスタンスのバックエンドとしてのCephの使用 356

- 27.1 qemu-block-rbdのインストール 356
- 27.2 QEMUの使用 356
- 27.3 QEMUでのイメージの作成 357
- 27.4 QEMUでのイメージのサイズ変更 357
- 27.5 QEMUでのイメージ情報の取得 357
- 27.6 RBDでのQEMUの実行 358
- 27.7 discardおよびTRIMの有効化 358
- 27.8 QEMUのキャッシュオプションの設定 359

## VI クラスタの設定 360

### 28 Cephクラスタの設定 361

- 28.1 ceph.confファイルの設定 361
  - コンテナイメージ内のceph.confへのアクセス 361
- 28.2 設定データベース 362
  - セクションとマスクの設定 362 • 設定オプションの設定と読み取り 362 • ランタイム中のデーモンの設定 363
- 28.3 config-keyの保存 365
  - iSCSI Gateway 366

- 28.4 Ceph OSDとBlueStore 367  
自動キャッシュサイズ調整の設定 367
- 28.5 Ceph Object Gateway 368  
一般的な設定 368 • HTTPフロントエンドの設定 377

## **29 Ceph Managerモジュール 380**

- 29.1 バランサ 380  
「crush-compat」モード 381 • データバランシングの計画と実行 381
- 29.2 テレメトリモジュールの有効化 382

## **30 cephxを使用した認証 385**

- 30.1 認証アーキテクチャ 385
- 30.2 キー管理 388  
予備知識 389 • ユーザの管理 392 • キーリングの管理 396 • コマンド  
ラインの使用 398

## **A アップストリーム「Octopus」ポイントリリースに基づくCeph保守更新 400**

## **B マニュアルの更新 403**

## **用語集 404**

# このガイドについて

このガイドでは、基本的なCephクラスタを展開した後に、管理者として実行する必要があるルーチンタスク(日常的な管理)に焦点を当てて説明しています。また、サポートされている、Cephクラスタに保存されたデータにアクセスする方法をすべて説明しています。

SUSE Enterprise Storage 7はSUSE Linux Enterprise Server 15 SP2の拡張機能です。Ceph (<http://ceph.com/>) ストレージプロジェクトの機能に、SUSEのエンタープライズエンジニアリングとサポートが組み合わされています。SUSE Enterprise Storage 7により、IT組織は、コモディティハードウェアプラットフォームを使用して多様な使用事例に対応できる分散ストレージアーキテクチャを展開できます。

## 1 利用可能なマニュアル



### 注記: オンラインマニュアルと最新のアップデート

製品に関するマニュアルは、<https://documentation.suse.com> からご利用いただけます。最新のアップデートもご利用いただけるほか、マニュアルをさまざまな形式でブラウズおよびダウンロードすることができます。最新のマニュアルアップデートは英語版で検索できます。

また、製品マニュアルは、`/usr/share/doc/manual`の下にあるインストール済みシステムから入手できます。製品マニュアルは `ses-manual_LANG_CODE` という名前のRPMパッケージに含まれています。システム上にマニュアルが存在しない場合は、たとえば次のコマンドを使用してインストールしてください。

```
root # zypper install ses-manual_en
```


この製品の次のマニュアルを入手できます。

導入ガイド (<https://documentation.suse.com/ses/html/ses-all/book-storage-deployment.html>)


このガイドでは基本的なCephクラスタの展開方法と、追加のサービスの展開方法に焦点を当てて説明しています。また、旧バージョンの製品をSUSE Enterprise Storage 7にアップグレードする手順についても説明しています。

**運用と管理ガイド** (<https://documentation.suse.com/ses/html/ses-all/book-storage-admin.html>) 

このガイドでは、基本的なCephクラスタを展開した後に、管理者として実行する必要があるルーチンタスク(日常的な管理)に焦点を当てて説明しています。また、サポートされている、Cephクラスタに保存されたデータにアクセスする方法をすべて説明しています。

**Security Hardening Guide** (<https://documentation.suse.com/ses/html/ses-all/book-storage-security.html>) 

このガイドでは、クラスタのセキュリティを確保する方法に焦点を当てて説明しています。

**トラブルシューティングガイド** (<https://documentation.suse.com/ses/html/ses-all/book-storage-troubleshooting.html>) 

このガイドでは、SUSE Enterprise Storage 7を実行する際のさまざまな一般的な問題と、CephやObject Gatewayのような関連コンポーネントに関する問題について説明しています。


**SUSE Enterprise Storage for Windows Guide** (<https://documentation.suse.com/ses/html/ses-all/book-storage-windows.html>) 


このガイドでは、Windowsドライバを使用したMicrosoft Windows環境とSUSE Enterprise Storageの統合、インストール、および設定について説明しています。

## 2 フィードバックの提供


このドキュメントに対するフィードバックや貢献を歓迎します。次のチャンネルがあります。

### サービス要求およびサポート

ご使用の製品に利用できるサービスとサポートのオプションについては、<http://www.suse.com/support/> を参照してください。

サービス要求を開くには、SUSE Customer Centerに登録されたSUSEの購読が必要です。<https://scc.suse.com/support/requests> に移動して、ログインし、新規作成をクリックします。

### バグレポート

<https://bugzilla.suse.com/> にあるドキュメントで問題を報告します。レポートिंगの問題には、Bugzillaアカウントが必要です。

このプロセスを簡略化するために、このドキュメントのHTMLバージョンの見出しの横にあるReport Documentation Bug (ドキュメントバグの報告)リンクを使用できます。これらにより、Bugzillaで適切な製品とカテゴリが事前に選択され、現在のセクションへのリンクが追加されます。バグレポートの入力を直ちに開始できます。

## 貢献内容

このドキュメントに貢献するには、このドキュメントのHTMLバージョンの見出しの横にあるEdit Source (ソースの編集)リンクを使用してください。GitHubのソースコードに移動し、そこでプル要求を開くことができます。貢献にはGitHubアカウントが必要です。

このドキュメントに使用されるドキュメント環境に関する詳細については、<https://github.com/SUSE/doc-ses>にあるリポジトリのREADMEを参照してください。

## メール

ドキュメントに関するエラーの報告やフィードバックは[doc-team@suse.com](mailto:doc-team@suse.com)宛に送信してください。ドキュメントのタイトル、製品のバージョン、およびドキュメントの発行日を記載してください。また、関連するセクション番号とタイトル(またはURL)、問題の簡潔な説明も記載してください。

# 3 マニュアルの表記規則

このマニュアルでは、次の通知と表記規則が使用されています。

- `/etc/passwd`: ディレクトリ名とファイル名
- `PLACEHOLDER`: `PLACEHOLDER`は、実際の値で置き換えられます。
- `PATH`: 環境変数
- `ls`、`--help`: コマンド、オプション、およびパラメータ
- `user`: ユーザまたはグループの名前
- `package_name`: ソフトウェアパッケージの名前
- `Alt`、`Alt + F1`: 押すキーまたはキーの組み合わせ。キーはキーボードのように大文字で表示されます。
- ファイル、ファイル > 名前を付けて保存: メニュー項目、ボタン
- `AMD/Intel` この説明は、Intel 64/AMD64アーキテクチャにのみ当てはまります。矢印は、テキストブロックの先頭と終わりを示します。◁

**IBM Z, POWER** この説明は、IBM ZおよびPOWERの各アーキテクチャにのみ当てはまります。矢印は、テキストブロックの先頭と終わりを示します。◁□

- 第1章、「章の例」：このガイドの別の章への相互参照。
- root特権で実行する必要があるコマンド。多くの場合、これらのコマンドの先頭にsudoコマンドを置いて、特権のないユーザとしてコマンドを実行することもできます。

```
root # command
tux > sudo command
```

- 特権のないユーザでも実行できるコマンド。

```
tux > command
```

- 通知



### 警告: 警告の通知

続行する前に知っておくべき、無視できない情報。セキュリティ上の問題、データ損失の可能性、ハードウェアの損傷、または物理的な危険について警告します。



### 重要: 重要な通知

続行する前に知っておくべき重要な情報です。



### 注記: メモの通知

追加情報。たとえば、ソフトウェアバージョンの違いに関する情報です。



### ヒント: ヒントの通知

ガイドラインや実地的なアドバイスなどの役に立つ情報です。

- コンパクトな通知



追加情報。たとえば、ソフトウェアバージョンの違いに関する情報です。



ガイドラインや実地的なアドバイスなどの役に立つ情報です。

## 4 製品のライフサイクルとサポート

SUSE製品が異なれば、製品のライフサイクルも異なります。SUSE Enterprise Storageのライフサイクルの正確な日付を確認するには、<https://www.suse.com/lifecycle/>を参照してください。

### 4.1 SUSEによるサポートの定義

サポートポリシーとオプションについては、<https://www.suse.com/support/policy.html>および<https://www.suse.com/support/programs/long-term-service-pack-support.html>を参照してください。

### 4.2 SUSE Enterprise Storageのサポートステートメント

サポートを受けるには、SUSEの適切な購読が必要です。利用可能なサポートサービスを具体的に確認するには、<https://www.suse.com/support/>にアクセスして製品を選択してください。

サポートレベルは次のように定義されます。

#### L1

問題の判別。互換性情報、使用サポート、継続的な保守、情報収集、および利用可能なドキュメントを使用した基本的なトラブルシューティングを提供するように設計されたテクニカルサポートを意味します。

#### L2

問題の切り分け。データの分析、お客様の問題の再現、問題領域の特定、レベル1で解決できない問題の解決、またはレベル3の準備を行うように設計されたテクニカルサポートを意味します。

#### L3

問題解決。レベル2サポートで特定された製品の欠陥を解決するようにエンジニアリングに依頼して問題を解決するように設計されたテクニカルサポートを意味します。

契約されているお客様およびパートナーの場合、SUSE Enterprise Storageでは、次のものを除くすべてのパッケージに対してL3サポートを提供します。

- 技術レビュー
- サウンド、グラフィック、フォント、およびアートワーク
- 追加の顧客契約が必要なパッケージ
- モジュール「Workstation Extension」の一部として出荷される一部のパッケージは、L2サポートのみです。
- 名前が `-devel` で終わるパッケージ(ヘッダファイルなどの開発用リソースが含まれるパッケージ)のサポートを受けるには、そのメインパッケージが必要です。

SUSEは、元のパッケージの使用のみをサポートします。つまり、変更も、再コンパイルもされないパッケージをサポートします。

## 4.3 技術レビュー

技術レビューとは、今後のイノベーションを垣間見ていただくための、SUSEによって提供されるパッケージ、スタック、または機能を意味します。技術レビューは、ご利用中の環境で新しい技術をテストする機会を参考までに提供する目的で収録されています。私たちはフィードバックを歓迎しています。技術レビューをテストする場合は、SUSEの担当者に連絡して、経験や使用例をお知らせください。ご入力いただいた内容は今後の開発のために役立たせていただきます。

技術レビューには、次の制限事項があります。

- 技術レビューはまだ開発中です。したがって、機能が不完全であったり、不安定であったり、何らかの理由で運用環境での使用には適していなかったりする場合があります。
- 技術レビューにはサポートが提供されません。
- 技術レビューは、特定のハードウェアアーキテクチャでしか利用できないことがあります。

- 技術プレビューの詳細および機能は、変更される場合があります。そのため、今後リリースされる技術プレビューへのアップグレードができない場合や、再インストールが必要となる場合があります。
- 技術プレビューは製品から予告なく削除される可能性があります。将来的にこうした技術に対応したバージョンを提供することを、SUSEはお約束しません。たとえば、プレビューがお客様や市場のニーズを満たしていない、またはエンタープライズ基準に準拠していないとSUSEが判断した場合です。

ご使用の製品に付属している技術プレビューの概要については、[https://www.suse.com/releases/notes/x86\\_64/SUSE-Enterprise-Storage/7](https://www.suse.com/releases/notes/x86_64/SUSE-Enterprise-Storage/7)にあるリリースノートを参照してください。

## 5 Cephの貢献者

Cephプロジェクトとそのドキュメントは、数百人の貢献者と組織の作業の結果です。詳しくは「<https://ceph.com/contributors/>」を参照してください。

## 6 このガイドで使用されるコマンドとコマンドプロンプト

Cephクラスタ管理者は、特定のコマンドを実行して、クラスタの動作を設定および調整します。必要になるコマンドには、次のようにいくつかの種類があります。

### 6.1 Salt関連のコマンド

これらのコマンドは、Cephクラスタノードを展開する場合や、クラスタノードの一部(または全部)で同時にコマンドを実行する場合、クラスタノードを追加または削除する場合に役立ちます。最も頻繁に使用されるコマンドは**ceph-salt**と**ceph-salt config**です。Salt Masterノードでは、Saltコマンドは`root`として実行する必要があります。これらのコマンドは、次のプロンプトで示されます。

```
root@master #
```

例:

```
root@master # ceph-salt config ls
```

## 6.2 Ceph関連のコマンド

これらは、**ceph**、**cephadm**、**rbd**、または**radosgw-admin**など、コマンドラインでクラスタとそのゲートウェイのすべての側面を設定および微調整するための下位レベルのコマンドです。

Ceph関連のコマンドを実行するには、Cephキーの読み取りアクセス権が必要です。このキーの機能により、Ceph環境内におけるユーザの特権が定義されます。1つのオプションは、**root**として(または**sudo**を使用して)Cephコマンドを実行し、制限のないデフォルトのキーリング「**ceph.client.admin.keyring**」を使用します。

より安全な推奨オプションは、各管理者ユーザに対してより制限の厳しい個別のキーを作成し、そのキーを、各ユーザが読み取ることができるディレクトリに保存することです。次に例を示します。

```
~/ceph/ceph.client.USERNAME.keyring
```



### ヒント: Cephキーのパス

カスタムの管理者ユーザとキーリングを使用するには、**ceph**コマンドを実行するたびに、**-n client.USER\_NAME**オプションと**--keyring PATH/TO/KEYRING**オプションを使用して、ユーザ名とプールのパスを指定する必要があります。

これを回避するには、個々のユーザの**~/.bashrc**ファイルで**CEPH\_ARGS**変数にこれらのオプションを含めてください。

Ceph関連のコマンドは任意のクラスタノードで実行できますが、管理ノードで実行することをお勧めします。このドキュメントでは、**cephadm**ユーザを使用してコマンドを実行するので、コマンドは次のプロンプトが表示されます。

```
cephuser@adm >
```

例:

```
cephuser@adm > ceph auth list
```



### ヒント: 特定のノード用のコマンド

クラスタノードに対して特定の役割でコマンドを実行するようドキュメントで指示されている場合は、プロンプトによって示されます。以下に例を示します。

```
cephuser@mon >
```

### 6.2.1 ceph-volumeの実行

SUSE Enterprise Storage 7から、Cephサービスはコンテナ化された状態で実行されます。OSDノード上で**ceph-volume**を実行する必要がある場合は、**cephadm**コマンドに付加する必要があります。たとえば、次のようになります。

```
cephuser@adm > cephadm ceph-volume simple scan
```

## 6.3 一般的なLinuxコマンド

**mount**、**cat**、または**openssl**など、Cephに関連しないLinuxコマンドは、関連するコマンドに必要な特権に応じて、**cephuser@adm >**または**root #**のいずれかで導入されます。

## 6.4 追加情報

Cephのキー管理の詳細については、[30.2項「キー管理」](#)を参照してください。

# I Cephダッシュボード

- 1 Cephダッシュボードについて 2
- 2 ダッシュボードのWebユーザインタフェース 3
- 3 Cephダッシュボードユーザと役割の管理 10
- 4 クラスタの内部情報の表示 15
- 5 プールの管理 27
- 6 RADOS Block Deviceの管理 30
- 7 NFS Ganeshaの管理 53
- 8 CephFSの管理 57
- 9 Object Gatewayの管理 59
- 10 手動設定 66
- 11 コマンドラインによるユーザと役割の管理 73

# 1 Cephダッシュボードについて

CephダッシュボードはWebベースのCeph管理/監視用ビルトインアプリケーションで、クラスタの様々な側面とオブジェクトを管理します。『導入ガイド』、第5章「cephadmによる展開」、5.3項「Cephクラスタの展開」で基本的なクラスタが展開されると、ダッシュボードは自動的に有効化されます。

SUSE Enterprise Storage 7用のCephダッシュボードには、Webベースの管理機能が追加され、Ceph Managerの監視やアプリケーション管理などのCephの管理が容易になりました。Ceph関連の複雑なコマンドを知らなくても、Cephクラスタを管理および監視できるようになりました。Cephダッシュボードの直感的なインターフェース、またはその組み込みREST APIのいずれかを使用できます。

Cephダッシュボードモジュールは、`ceph-mgr`がホストされたWebサーバを使用して、Cephクラスタに関する情報と統計を視覚化します。Ceph Managerの詳細については、『導入ガイド』、第1章「SESとCeph」、1.2.3項「Cephのノードとデーモン」を参照してください。

## 2 ダッシュボードのWebユーザインタフェース

### 2.1 ログイン

Cephダッシュボードにログインするには、ポート番号を含むCephダッシュボードのURLをブラウザに与えます。アドレスを確認するには、次のコマンドを実行します。

```
cephuser@adm > ceph mgr services | grep dashboard
"dashboard": "https://host:port/",
```

このコマンドはCephダッシュボードが置かれている場所のURLを返します。このコマンドの使用に関して問題が生じた場合、『Troubleshooting Guide』、第10章「Troubleshooting the Ceph Dashboard」、10.1項「Locating the Ceph Dashboard」を参照してください。



図 2.1: CEPHダッシュボードのログイン画面

クラスターの展開時に作成した資格情報を使用してログインします(『導入ガイド』、第5章「cephadmによる展開」、5.3.2.9項「Cephダッシュボードログインアカウント情報の設定」を参照してください)。



## ヒント: カスタムユーザアカウント

Cephダッシュボードへアクセスするのにデフォルトの「admin」アカウントを使用しない場合は、管理者特権を持つカスタムユーザアカウントを作成します。詳細については、[第11章「コマンドラインによるユーザと役割の管理」](#)を参照してください。

ダッシュボードのユーザインターフェースはいくつかの「ブロック」に別れています。すなわち、画面右上の「ユーティリティメニュー」、画面左側の「メインメニュー」、そして、中央の「コンテンツペイン」です。



図 2.2: CEPHダッシュボードのホームページ

## 2.2 ユーティリティメニュー

画面の右上にはユーティリティメニューがあります。ここには、Cephクラスタよりもダッシュボードとの関連性が高い一般的なタスクが含まれます。オプションをクリックすることで、次のトピックにアクセスできます。

- ダッシュボードの言語インターフェースの変更: チェコ語、ドイツ語、英語、スペイン語、フランス語、インドネシア語、イタリア語、日本語、韓国語、ポーランド語、ポルトガル語(ブラジル)、中国語に対応しています。
- タスクと通知
- ドキュメント、REST APIに関する情報、ダッシュボードに関する詳細を表示します。

- ユーザ管理とテレメトリ設定。



## 注記

ユーザの役割に関するコマンドラインの説明の詳細については、[第11章「コマンドラインによるユーザと役割の管理」](#)を参照してください。

- ログイン設定。パスワードの変更やサインアウトができます。

## 2.3 メインメニュー

ダッシュボードのメインメニューは、画面の左側にあります。これは次のトピックに対応します。

### ダッシュボード

Cephダッシュボードのホームページに戻ります。

### クラスタ

ホスト、インベントリ、Ceph Monitor、サービス、Ceph OSD、クラスタ設定、CRUSHマップ、Ceph Managerモジュール、ログ、および監視についての詳細情報を表示します。

### プール

クラスタプールを表示および管理します。

### ブロック

RADOS Block Deviceのイメージ、ミラーリング、iSCSIの詳細情報の表示と管理を行います。

### NFS

NFS Ganeshaの展開を表示および管理します。



## 注記

NFS Ganeshaが展開されていない場合は、参考情報が表示されます。[11.6項「CephダッシュボードによるNFS Ganeshaの設定」](#)を参照してください。

### ファイルシステム

CephFSを表示および管理します。

## Object Gateway

Object Gatewayのデーモン、ユーザ、およびバケットを表示および管理します。



### 注記


オブジェクトゲートウェイが展開されていない場合は、通知が表示されません。[10.4項「Object Gateway管理フロントエンドの有効化」](#)を参照してください。

## 2.4 コンテンツペイン


コンテンツペインは、ダッシュボードの画面のメイン部分を占めます。ダッシュボードのホームページには、多数の便利なウィジェットが表示されており、クラスタの現在のステータス、容量、およびパフォーマンス情報について簡潔に通知します。


## 2.5 Web UIの共通機能

Cephダッシュボードでは、プールのリスト、OSDノードのリスト、RBDデバイスのリストなど、「リスト」を操作することがよくあります。すべてのリストは、デフォルトでは5秒ごとに自動更新されます。次の共通ウィジェットは、これらのリストを管理または調整する場合に役立ちます。

 をクリックして、リストの手動更新をトリガします。

 をクリックして、個々のテーブル列の表示/非表示を切り替えます。

 をクリックして、1ページに表示する行数を入力(または選択)します。

 の内部をクリックし、検索する文字列を入力して行をフィルタします。

リストが複数のページにわたる場合は、 を使用して、現在表示されているページを変更します。

## 2.6 ダッシュボードウィジェット

各ダッシュボードウィジェットには、実行中のCephクラスタの特定の側面に関連する特定のステータス情報が表示されます。一部のウィジェットはアクティブなリンクになっており、クリックすると、そのウィジェットが表すトピックに関連する詳細ページにリダイレクトされます。



## ヒント: マウスポインタを合わせて詳細を表示

一部のグラフィカルウィジェットでは、マウスポインタを合わせると詳細が表示されます。

### 2.6.1 ステータスウィジェット

ステータスウィジェットは、クラスタの現在のステータスに関する簡単な概要を提供します。

ステータス



図 2.3: ステータスウィジェット

#### クラスタのステータス

クラスタのヘルスに関する基本的な情報が示されます。

#### ホスト

クラスタノードの合計数が表示されます。

#### モニター

実行中のモニターとその定数の数が表示されます。

#### OSD

OSDの合計数と、「up」および「in」のOSDの数が表示されます。

#### マネージャ

アクティブおよびスタンバイ状態のCeph Managerデーモンの数が表示されます。

#### Object Gateway

実行中のObject Gatewayの数が表示されます。

#### メタデータサーバ

メタデータサーバの数が表示されます。

#### iSCSI Gateway

設定されているiSCSI Gatewayの数が表示されます。

## 2.6.2 容量のウィジェット

容量ウィジェットには、ストレージ容量に関する簡単な情報が表示されます。

容量



図 2.4: 容量のウィジェット

### 未フォーマット時の容量

使用済みの容量と使用可能な未フォーマット時のストレージ容量の比率が表示されます。

### オブジェクト

クラスタに保存されているデータオブジェクトの数が表示されます。

### 配置グループのステータス

配置グループのチャートがステータスに従って表示されます。

### プール

クラスタのプールの数が表示されます。

### OSDあたりの配置グループ数

OSDあたりの配置グループの平均数が表示されます。

## 2.6.3 パフォーマンスウィジェット

パフォーマンスウィジェットは、Cephクライアントの基本的なパフォーマンスを参照します。

パフォーマンス



図 2.5: パフォーマンスウィジェット

### クライアントの読み取り/書き込み

1秒あたりのクライアントの読み取り/書き込み操作の量。

### クライアントのスループット

1秒あたりのCephクライアントとの間で転送されるデータ量(バイト単位)。

### 回復スループット

1秒あたりに回復されるデータのスループット。

### スクラブ


スクラビング([17.4.9項「配置グループのスクラブ」](#)を参照してください)のステータスを表示します。ステータスは[非アクティブ](#)、[有効化済み](#)、[アクティブ](#)のいずれかです。

## 3 Cephダッシュボードユーザと役割の管理

コマンドラインでCephコマンドを使用して実行するダッシュボードユーザ管理については、第11章「コマンドラインによるユーザと役割の管理」ですでに紹介しています。

このセクションでは、ダッシュボードWebユーザインタフェースを使用してユーザアカウントを管理する方法について説明します。

### 3.1 ユーザの一覧

ユーティリティメニューの  をクリックし、ユーザ管理を選択します。

リストに含まれる情報は、各ユーザのユーザ名、フルネーム、メールアドレス、割り当てられた役割の一覧、役割が有効かどうか、および、パスワード失効日です。



ユーザ名	名前	電子メール	役割	有効化済み	Password expiration date
admin			administrator	✓	
Alex	Alexandra Settle	tux@example.com	cluster-manager, pool-manager	✓	
dashboard user 1	Dashboard User1	du1@example.com		✓	
rgw user	RGW User	rgw@example.com	pool-manager, rgw-manager	✓	

図 3.1: ユーザの管理

### 3.2 新しいユーザの追加

新しいユーザを追加するには、テーブル見出しの左上の作成をクリックします。ユーザ名、パスワード、およびオプションで氏名と電子メールを入力します。

作成 ユーザ

ユーザ名 \*

potato

✓

パスワード ?

.....

✓

👁

パスワードの確認入力

.....

✓

👁

Password expiration date ?

Password expiration date...

✕

氏名

Mr. Potato

✓

電子メール

potato@example.com

✓

役割

✎ There are no roles.

✓ 有効化済み

✓ User must change password at next login

作成 ユーザ

キャンセル

図 3.2: ユーザの追加

事前定義された役割をユーザに割り当てるには、小さいペンのアイコンをクリックします。Create User (ユーザの作成) をクリックして確認します。


## 3.3 ユーザの編集

ユーザのテーブル行をクリックして選択を強調表示します。編集を選択して、ユーザに関する詳細を編集します。ユーザの編集をクリックして確認します。

## 3.4 ユーザの削除

ユーザのテーブル行をクリックして選択を強調表示します。編集の横にあるドロップダウンボタンを選択します。リストから削除を選択してユーザアカウントを削除します。はいチェックボックスをオンにし、Delete User (ユーザの削除) をクリックして確認します。

## 3.5 ユーザの役割の一覧

ユーティリティメニューの  をクリックし、ユーザ管理を選択します。続いて、役割タブをクリックします。

このリストには、各役割の名前、説明、およびシステム役割であるかどうかが含まれます。



名前	説明	システム役割
administrator	Administrator	✓
block-manager	Block Manager	✓
cephfs-manager	CephFS Manager	✓
cluster-manager	Cluster Manager	✓
ganesha-manager	NFS Ganesha Manager	✓
pool-manager	Pool Manager	✓
read-only	Read-Only	✓
rgw-manager	RGW Manager	✓

図 3.3: ユーザの役割

## 3.6 カスタム役割の追加

新しいカスタム役割を追加するには、テーブル見出しの左上の作成をクリックします。名前と説明を入力し、許可の横で、適切な許可を選択します。



### ヒント: カスタム役割の消去

カスタムユーザ役割を作成し、後で **ceph-salt purge** コマンドを使ってCephクラスタを削除する場合、まずカスタム役割を消去する必要があります。詳細については、[13.9 項「Cephクラスタ全体の削除」](#)を参照してください。

作成 Role

名前 \*

ganesha pool user ✓

説明

a user that can only manage ganesha and pools ✓

許可

<input type="checkbox"/> すべて	<input type="checkbox"/> 読み取り	<input type="checkbox"/> 作成	<input type="checkbox"/> 更新	<input type="checkbox"/> 削除
<input type="checkbox"/> cephfs	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> config-opt	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> dashboard-settings	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> grafana	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> hosts	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> iscsi	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> log	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> manager	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> monitor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> nfs-ganesha	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> osd	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> pool	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> prometheus	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> rbd-image	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> rbd-mirroring	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> rgw	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> user	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

作成 Role キャンセル

図 3.4: 役割の追加



## ヒント: 複数の有効化

トピック名の前にあるチェックボックスをオンにすると、そのトピックの許可がすべて有効になります。すべてチェックボックスをオンにすると、すべてのトピックの許可がすべて有効になります。

Create Role (役割の作成)をクリックして確認します。

## 3.7 カスタム役割の編集

カスタムの役割の説明と許可を編集するには、ユーザのテーブル行をクリックして選択を強調表示し、テーブル見出しの左上の編集を選択します。役割の編集をクリックして確認します。

## 3.8 カスタム役割の削除

役割のテーブル行をクリックして選択を強調表示してから、編集の横にあるドロップダウンボタンを選択します。リストから削除を選択して役割を削除します。はいチェックボックスをオンにし、Delete Role (役割の削除)をクリックして確認します。

## 4 クラスタの内部情報の表示

クラスタメニュー項目を使用すると、Cephクラスタホスト、インベントリ、Ceph Monitor、サービス、OSD、設定、CRUSHマップ、Ceph Manager、ログ、および監視ファイルに関する詳細情報を表示できます。

### 4.1 クラスタノードの表示

クラスタノードのリストを表示するには、クラスタ > ホストをクリックします。



The screenshot shows the 'Hosts' page in the Ceph web interface. At the top, there are tabs for 'Hosts List' and 'Overall Performance'. Below the tabs is a table with columns: Host Name, Service, Labels, and Version. The table lists four nodes: 'master', 'node1', 'node2', and 'node3'. Each node has a dropdown arrow next to its name. The 'node1' row is highlighted. Below the table, it says '0 selected / 4 total'.

ホスト名	サービス	Labels	バージョン
master			
node1	mgr,node1.wbmq,mon,node1.osd.0,osd.3		15.2.4-557-g4ac763f0b3
node2	mgr,node2.qcwalx,mon,node2.osd.1,osd.4		15.2.4-557-g4ac763f0b3
node3	mgr,node3.rhkzy,mon,node3.osd.2,osd.5		15.2.4-557-g4ac763f0b3

図 4.1: ホスト

ノードのパフォーマンスの詳細を表示するには、ホスト名列のノード名の横にあるドロップダウン矢印をクリックします。

サービス列には、関連する各ノードで実行されているすべてのデーモンが一覧にされます。デーモン名をクリックすると、その詳細な設定が表示されます。

### 4.2 クラスタのインベントリへのアクセス

デバイスのリストを表示するには、クラスタ > インベントリをクリックします。リストにはデバイスのパス、種類、利用可能かどうか、ベンダー、モデル、サイズ、OSDが含まれます。

ホスト名列のノード名をクリックして選択します。選択した状態で、識別をクリックすると、そのホストを実行しているデバイスが特定されます。このとき、デバイスにLEDを点滅させるよう指示されます。このアクションの時間は、1、2、5、10、15分から選択します。実行をクリックします。

ホスト名	Device path	タイプ	Available	Vendor	Model	サイズ	OSD
master	/dev/vda	HDD		Ox1af4		42 GiB	
node1	/dev/vda	HDD		Ox1af4		42 GiB	
node1	/dev/vdb	HDD		Ox1af4		8 GiB	osd.0
node1	/dev/vdc	HDD		Ox1af4		8 GiB	osd.3
node2	/dev/vda	HDD		Ox1af4		42 GiB	
node2	/dev/vdb	HDD		Ox1af4		8 GiB	osd.1
node2	/dev/vdc	HDD		Ox1af4		8 GiB	osd.4
node3	/dev/vda	HDD		Ox1af4		42 GiB	
node3	/dev/vdb	HDD		Ox1af4		8 GiB	osd.2
node3	/dev/vdc	HDD		Ox1af4		8 GiB	osd.5

0 選択済み / 10 合計

図 4.2: サービス

## 4.3 Ceph Monitorの表示

実行中のCeph Monitorが存在するクラスタノードのリストを表示するには、クラスターモニターをクリックします。コンテンツペインは2つのビューに分割されます。1つはステータスビューで、もう1つは定数内と非定数内のビューです。

ステータステーブルには、実行中のCeph Monitorに関する一般的な統計情報が表示されます。内容は次の通りです。

- クラスターID
- monmapが変更されました
- monmapエポック
- 定数con
- 定数mon
- 必須のcon
- 必須のmon

定数内と非定数内のペインには、各Monitorの名前、ランク数、パブリックIPアドレス、開いているセッション数が含まれます。

関連するCeph Monitorの設定を表示するには、名前列のノード名をクリックします。

ステータス	
クラスターID	06766fa4-a9a7-11eb-9e46-625400b22828
monmapが変更されました	2021-04-30T11:27:20.465652Z
monmapエポック	1
定数con	4540138292840890367
定数mon	kraken,luminous,mimic,osdmap-prune,nautilus,octopus
必須のcon	2449958747315978244
必須のmon	kraken,luminous,mimic,osdmap-prune,nautilus,octopus

#### 定数内

名前	ランク	パブリックアドレス	セッションの開始
node1	0	10.20.156.201:6789/0	.....
node2	2	10.20.156.202:6789/0	.....
node3	1	10.20.156.203:6789/0	.....
3 合計			

#### 非定数内

名前	ランク	パブリックアドレス
No data to display		
0 合計		

図 4.3: CEPH MONITOR

## 4.4 サービスの表示

利用可能な各サービスの詳細を表示するには、クラスター > サービスをクリックします。サービスの例としては、crash、Ceph Manager、Ceph Monitorなどがあります。リストにはコンテナイメージの名前、コンテナイメージのID、実行中の内容のステータス、サイズ、最後に更新された日時が含まれます。

デーモンの詳細を表示するには、サービス列のサービス名の横にあるドロップダウン矢印をクリックします。詳細リストには、ホスト名、デーモンの種類、デーモンID、コンテナID、コンテナイメージ名、コンテナイメージID、バージョン番号、ステータス、最後に更新された日時が含まれます。

Cluster > Services

× 削除

🔄

🗑

10

🔍

×

サービス	Container image name	Container image ID	Placement	Running	サイズ	Last Refreshed
▼ crash	registry.suse.de/devel/storage/7.0/containers/ses/7/ceph/ceph:latest	6549871c3f67			4	2020-08-14T13:37:34.148847

デーモン

🔄

🗑

10

🔍

×

📉

ホスト名

Any

ホスト名	Daemon type	Daemon ID	Container ID	Container image name	Container image ID	バージョン	ステータス	Last Refreshed
master	crash	master	3acfc11b607e	registry.suse.de/devel/storage/7.0/containers/ses/7/ceph/ceph:latest	6549871c3f67	15.2.4.557	running	2020-08-14T13:37:34.148847
node1	crash	node1	3d56e2a421eb	registry.suse.de/devel/storage/7.0/containers/ses/7/ceph/ceph:latest	6549871c3f67	15.2.4.557	running	2020-08-14T13:37:35.371944
node2	crash	node2	8fa9790b9a51	registry.suse.de/devel/storage/7.0/containers/ses/7/ceph/ceph:latest	6549871c3f67	15.2.4.557	running	2020-08-14T13:37:35.208871
node3	crash	node3	b047531bf2a	registry.suse.de/devel/storage/7.0/containers/ses/7/ceph/ceph:latest	6549871c3f67	15.2.4.557	running	2020-08-14T13:37:35.965886
4 合計								

>	mgr	registry.suse.de/devel/storage/7.0/containers/ses/7/ceph/ceph:latest	dcfacef0831b	master	1	1	2021-08-04T13:45:07.130845Z
>	mon	registry.suse.de/devel/storage/7.0/containers/ses/7/ceph/ceph:latest	dcfacef0831b	master:10.20.165.200	1	1	2021-08-04T13:45:07.131382Z
>	node-exporter	registry.suse.com/caasp/v4.5/prometheus-node-exporter:0.18.1	a149a78bcd37	master	1	1	2021-08-04T13:45:07.132807Z
>	osd.sesdev_osd_deployer	registry.suse.de/devel/storage/7.0/containers/ses/7/ceph/ceph:latest	dcfacef0831b	master	4	4	2021-08-04T13:45:07.131429Z

1 選択済み / 5 合計

図 4.4: サービス

## 4.5 Ceph OSDの表示

実行中のOSDデーモンが存在するノードのリストを表示するには、クラスタ>OSDをクリックします。このリストには、各ノードの名前、ID、ステータス、デバイスクラス、配置グループの数、サイズ、使用状況、時間内の読み込み/書き込みチャート、および1秒あたりの読み込み/書き込み操作の速度が含まれます。



The screenshot shows the 'OSD List' page in a monitoring tool. It features a table with columns for Host, ID, Status, Device class, Config Group, Size, Flags, Usage, Read/Write charts, and Read/Write rates. There are 6 OSDs listed across 3 nodes (node1, node2, node3). Each OSD is in a 'In Up' state. The usage for each OSD is 13%.

ホスト	ID	ステータス	Device class	配置グループ数	サイズ	フラグ	使用量	読み取りバイト数	書き込みバイト数	読み取り操作数	書き込み操作数
node1	0	In Up	hdd	0	8 GiB		13%			0 /s	0 /s
node2	1	In Up	hdd	1	8 GiB		13%			0 /s	0 /s
node3	2	In Up	hdd	1	8 GiB		13%			0 /s	0 /s
node1	3	In Up	hdd	1	8 GiB		13%			0 /s	0 /s
node2	4	In Up	hdd	1	8 GiB		13%			0 /s	0 /s
node3	5	In Up	hdd	0	8 GiB		13%			0 /s	0 /s

図 4.5: CEPH OSD

ポップアップウィンドウを開くには、テーブル見出しのクラスタ全体の設定ドロップダウンメニューから、フラグを選択します。このウィンドウにはクラスタ全体に適用されるフラグの一覧が表示されます。個々のフラグを有効または無効にし、送信をクリックして確認できます。

クラスタ全体のOSDフラグ

☐ インなし

以前にアウトとしてマークされたOSDは、それらの始動時にインとしてマークされることはありません

☐ アウトなし

OSDは、設定済みの間隔が経過した後に自動的にアウトとしてマークされます

☐ アップなし

OSDを始動することは許可されていません

☐ ダウンなし

OSDの障害レポートは無視されているため、OSDはモニターによってダウンとしてマークされません

☐ 一時停止

読み取りと書き込みを一時停止します

☐ スクラブなし

スクラブ処理は無効化されています

☐ ディープスクラブなし

ディープスクラブ処理は無効化されています

☐ バックフィルなし

配置グループのバックフィルは中断されています

☐ No Rebalance

OSD will choose not to backfill unless PG is also degraded

☐ 回復なし

配置グループの回復は中断されています

☒ ビット単位のソート

ビット単位のソートを使用する

☒ 消去されたスナップディレクトリ

OSDによってスナップセットが変換されました

送信

キャンセル

図 4.6: OSDフラグ

ポップアップウィンドウを開くには、テーブル見出しのクラスタ全体の設定ドロップダウンメニューから、回復優先度を選択します。このウィンドウにはクラスタ全体に適用されるOSDの回復優先度の一覧が表示されます。希望する優先度プロファイルを有効にし、その下の個々の値を微調整できます。送信をクリックして確認します。

OSD回復優先度

優先度 \*

カスタム

☒ 優先度値のカスタマイズ

最大バックフィル数 \* ?

1

回復最大アクティブ \* ?

0

回復最大単一始動 \*

1

回復スリープ \* ?

0

送信

キャンセル

図 4.7: OSD回復優先度

デバイスの設定とパフォーマンスに関する詳細が含まれる拡張テーブルを表示するには、ホスト列のノード名の横にあるドロップダウン矢印をクリックします。複数のタブを参照すると、属性、メタデータ、デバイスヘルス、パフォーマンスカウンタ、読み込みと書き込みをグラフィカルに表示するヒストグラム、およびパフォーマンスの詳細の各リストを確認できます。

OSDリスト

全体的なパフォーマンス

編集

Cluster-wide configuration

10

Q

	ホスト	ID	ステータス	Device class	配置グループ数	サイズ	フラグ	使用量	読み取りバイト数	書き込みバイト数	読み取り操作数	書き込み操作数
<input checked="" type="checkbox"/>	master	0	<div><div>In</div><div>up</div></div>	<div><div>hdd</div></div>	171	8 GiB		<div><div></div><div>12%</div></div>	<div><div></div></div>	<div><div></div></div>	1,799,664,569,364,049 /s	0 /s

Devices

属性(OSDマップ)

メタデータ

Device health

パフォーマンスカウンタ

ヒストグラム

パフォーマンスの詳細

10

Q

名前	説明	値
bluefs.bytes_written_slow	Bytes written to WAL/SSTs at slow device	0
bluefs.bytes_written_sst	Bytes written to SSTs	0
bluefs.bytes_written_wal	Bytes written to WAL	0
bluefs.db_total_bytes	Total bytes (main db device)	1073741824
bluefs.db_used_bytes	Used bytes (main db device)	301268992
bluefs.log_bytes	Size of the metadata log	4517888
bluefs.logged_bytes	Bytes written to the metadata log	0
bluefs.num_files	File count	12
bluefs.read_bytes	Bytes requested in buffered read mode	0
bluefs.read_prefetch_bytes	Bytes requested in prefetch read mode	0

112 合計

<

<<

1

2

3

4

5

>

>>

図 4.8: OSDの詳細



## ヒント: OSDでの固有のタスクの実行

OSDノード名をクリックすると、テーブルの行が強調表示されます。これは、そのノードでタスクを実行できることを表すものです。以下のアクションのいずれかを選択して実行できます。編集、作成、スクラブ、ディープスクラブ、再重みづけ、アウトとしてマーク、インとしてマーク、ダウンとしてマーク、喪失としてマーク、消去、破棄、削除。

作成ボタン隣のテーブル見出しの左上にある下矢印をクリックし、実行したいタスクを選択します。

### 4.5.1 OSDの追加

新しいOSDを追加するには、次の手順に従います。

1. ステータスが使用可能のストレージデバイスを持つクラスタノードが存在することを確認します。テーブル見出しの左上にある下矢印をクリックして、作成を選択します。これにより、OSDの作成ウィンドウが開きます。

図 4.9: OSDの作成

2. OSDにプライマリストレージデバイスを追加するには、追加をクリックします。ストレージデバイスを追加する前に、プライマリデバイステーブルの右上でフィルタ条件を指定する必要があります。たとえば、タイプ hdd など。追加をクリックして確認します。

プライマリデバイス

10

タイプ

hdd

タイプ: hdd

フィルタをクリア

ホスト名	デバイスパス	タイプ	ベンダ名	モデル	サイズ
doc-ses-min1	/dev/vdb	HDD	0x1af4		12GiB
doc-ses-min1	/dev/vdc	HDD	0x1af4		12GiB
合計2					

デバイス数: 2。未フォーマット時の容量: 24GiB。

追加

キャンセル

図 4.10: プライマリデバイスの追加

- 更新されたOSDの作成ウィンドウで、必要に応じて共有WALとBDデバイスを追加したり、デバイスの暗号化を有効化したりします。

OSDの作成

プライマリデバイス

タイプ: hdd

クリア

ホスト名	デバイスパス	タイプ	ベンダ名	モデル	サイズ
doc-ses-min1	/dev/vdb	HDD	0x1af4		12GiB
doc-ses-min1	/dev/vdc	HDD	0x1af4		12GiB
合計2					

未フォーマット時の容量: 24GiB

共有デバイス

WALデバイス

+

追加

DBデバイス

+

追加

設定

機能

☐ 暗号化

プレビュー

キャンセル

図 4.11: プライマリデバイスを追加したOSDの作成

- 以前追加したデバイスのDriveGroup仕様のプレビューを表示するには、プレビューをクリックします。作成をクリックして確認します。

OSD作成プレビュー

DriveGroups

```
[
  {
    "service_type": "osd",
    "service_id": "dashboard-admin-1600784434446",
    "host_pattern": "**",
    "data_devices": {
      "rotational": true
    }
  }
]
```

作成 キャンセル

図 4.12:

- OSDのリストに新しいデバイスが追加されます。

OSDリスト 全体的なパフォーマンス

+ 作成

Cluster-wide configuration

10

Q

X

	ホスト	ID	ステータス	Device class	配置グループ数	サイズ	フラグ	使用量	読み取りバイト数	書き込みバイト数	読み取り操作数	書き込み操作数
<input type="checkbox"/>	> doc-ses-min2	0	<span>in</span> <span>up</span>	hdd	119	10 GiB		11%			0.7999105934891158 /s	0 /s
<input type="checkbox"/>	> doc-ses-min3	1	<span>in</span> <span>up</span>	hdd	108	10 GiB		11%			1.5998816768416986 /s	0 /s
<input type="checkbox"/>	> doc-ses-min4	2	<span>in</span> <span>up</span>	hdd	126	10 GiB		11%			0 /s	0 /s
<input type="checkbox"/>	> doc-ses-min1	3	<span>in</span> <span>up</span>	hdd	96	12 GiB		9%			0.399945526088382 /s	0 /s
<input type="checkbox"/>	> doc-ses-min1	4	<span>in</span> <span>up</span>	hdd	76	12 GiB		9%			1.9995708432976873 /s	0 /s

0 選択済み / 5 合計

図 4.13: 新しく追加されたOSD

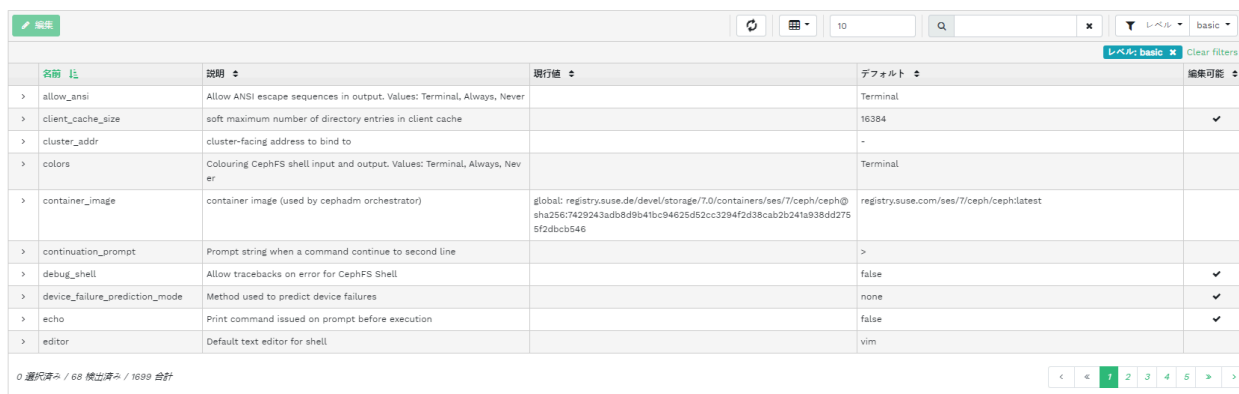


## 注記

OSD作成プロセスの進捗を視覚化する機能はありません。実際に作成されるまでには、多少の時間がかかります。OSDは展開が完了すると、リストに表示されます。展開のステータスを確認したい場合は、[クラスターログ](#)をクリックして、ログを表示します。

## 4.6 クラスタ設定の表示

Cephクラスタ設定オプションの完全なリストを表示するには、**クラスタ > 設定**をクリックします。リストにはオプション名、簡単な説明、現在値とデフォルト値、およびオプションを編集可能かどうかが含まれます。



The screenshot shows the Ceph cluster settings interface. At the top, there is a search bar and a filter level set to 'basic'. Below this is a table with the following columns: '名前' (Name), '説明' (Description), '現在値' (Current Value), 'デフォルト' (Default), and '編集可能' (Editable). The table lists various configuration options such as 'allow\_ansi', 'client\_cache\_size', 'cluster\_addr', 'colors', 'container\_image', 'continuation\_prompt', 'debug\_shell', 'device\_failure\_prediction\_mode', 'echo', and 'editor'. Each row includes a chevron icon to the left of the name, a detailed description, the current value, the default value, and a checkmark indicating if it is editable.

名前	説明	現在値	デフォルト	編集可能
> allow_ansi	Allow ANSI escape sequences in output. Values: Terminal, Always, Never		Terminal	
> client_cache_size	soft maximum number of directory entries in client cache		16384	✓
> cluster_addr	cluster-facing address to bind to		-	
> colors	Colouring CephFS shell input and output. Values: Terminal, Always, Never		Terminal	
> container_image	container image (used by cephadm orchestrator)	global: registry.suse.de/devrel/storage/7.0/containers/ses/7/ceph/ceph@sha256:7429243adb8d9b41bc94625d5cc3294f2d38cab2b241a938dd2755f2dbcb546	registry.suse.com/ses/7/ceph/ceph:latest	
> continuation_prompt	Prompt string when a command continue to second line		>	
> debug_shell	Allow tracebacks on error for CephFS Shell		false	✓
> device_failure_prediction_mode	Method used to predict device failures		none	✓
> echo	Print command issued on prompt before execution		false	✓
> editor	Default text editor for shell		vim	

図 4.14: クラスタの設定

オプションに関する詳細情報が含まれる拡張テーブルを表示するには、名前列の設定オプションの隣にあるドロップダウン矢印をクリックします。詳細情報としては、値の種類、最小許容値と最大許容値、実行中に更新可能か、などが含まれます。

特定のオプションを強調表示した後で、テーブル見出しの左上の編集ボタンをクリックして、その値を編集できます。保存をクリックして変更を確定します。

## 4.7 CRUSHマップの表示

クラスタのCRUSHマップを表示するには、**クラスタ > CRUSHマップ**をクリックします。CRUSHマップの全般的な情報については、17.5項「**CRUSHマップの操作**」を参照してください。

ルート、ノード、または個々のOSDをクリックすると、CRUSHの重み、マップツリー内の深さ、OSDのデバイスクラスなどの詳細情報が表示されます。

CRUSHマップビューア																	
<div>▼ default (root)</div> <div>▼ node3 (host)</div> <div> <div>up</div> <div>osd.2 (osd)</div> </div> <div> <div>up</div> <div>osd.5 (osd)</div> </div> <div>▼ node1 (host)</div> <div> <div>up</div> <div>osd.0 (osd)</div> </div> <div> <div>up</div> <div>osd.3 (osd)</div> </div> <div>▼ node2 (host)</div> <div> <div>up</div> <div>osd.1 (osd)</div> </div> <div> <div>up</div> <div>osd.4 (osd)</div> </div>	<div>osd.2 (osd)</div> <table> <tr><td>crush_weight</td><td>0.0077972412109375</td></tr> <tr><td>depth</td><td>2</td></tr> <tr><td>device_class</td><td>hdd</td></tr> <tr><td>exists</td><td>1</td></tr> <tr><td>id</td><td>2</td></tr> <tr><td>primary_affinity</td><td>1</td></tr> <tr><td>reweight</td><td>1</td></tr> <tr><td>type_id</td><td>0</td></tr> </table>	crush_weight	0.0077972412109375	depth	2	device_class	hdd	exists	1	id	2	primary_affinity	1	reweight	1	type_id	0
crush_weight	0.0077972412109375																
depth	2																
device_class	hdd																
exists	1																
id	2																
primary_affinity	1																
reweight	1																
type_id	0																

図 4.15: CRUSHマップ

## 4.8 マネージャモジュールの表示

使用可能なCeph Managerモジュールのリストを表示するには、クラスタ>マネージャモジュールをクリックします。各行は、モジュール名と、そのモジュールが現在有効かどうかに関する情報で構成されます。

編集

🔄

📅

10

🔍

✕

名前	有効化済み
ansible	
balancer	
crash	
dashboard	✓
deepsea	
devicehealth	
diskprediction_local	
influx	
insights	
iostat	✓

1個選択/合計24個

<

<<

1

2

3

>>

>

図 4.16: マネージャモジュール

詳細設定を含む拡張テーブルを、下の詳細テーブルに表示するには、名前列のモジュールの隣にあるドロップダウン矢印をクリックします。設定を編集するには、テーブル見出しの左上の編集ボタンをクリックします。更新をクリックして変更を確定します。

モジュールを有効化または無効化するには、テーブル見出しの左上にある編集ボタンの隣のドロップダウン矢印をクリックします。

## 4.9 ログの表示

クラスタの最近のログエントリのリストを表示するには、クラスタ > ログをクリックします。各行は、タイムスタンプ、ログエントリのタイプ、およびログに記録されたメッセージ自体で構成されます。

監査サブシステムのログエントリを表示するには、監査ログタブをクリックします。監査を有効または無効にするためのコマンドについては、11.5項「API要求の監査」を参照してください。



図 4.17: ログ

## 4.10 監視の表示

Prometheusアラートの詳細の管理と表示を行うには、クラスタ > 監視をクリックします。

アクティブなPrometheusが存在するなら、このコンテンツペインの有効なアラート、すべてのアラート、またはサイレンスから詳細な情報を表示できます。



### 注記

Prometheusが展開されていない場合、表示される情報バナーから関連するドキュメントを表示できます。

## 5 プールの管理



### ヒント: プールについての詳細

Cephのプールの全般的な情報については、[第18章「ストレージプールの管理」](#)を参照してください。イレージャコーディングプールに固有の情報については、[第19章「イレージャコーディングプール」](#)を参照してください。

利用可能なすべてのプールを一覧にするには、メインメニューからPools (プール)をクリックします。

このリストには、各プールの名前、タイプ、関連するアプリケーション、配置グループのステータス、レプリカサイズ、最後の変更、イレージャコード化済みプロファイル、使用量、Crushルールセット、および読み書きの統計情報が表示されます。

プールリスト

全体的なパフォーマンス

+ 追加

10

Q

×

名前	タイプ	アプリケーション	配置グループのステータス	レプリカサイズ	最後の更新	イレージョコード化済みプロフィール	Crushルールセット	使用率	読み取りバイト数	書き込みバイト数	読み取り操作数	書き込み操作数
.rgw.root	複製	rgw	active+clean 8	3	22		replicated_rule	0%	<div></div>	<div></div>	0 / 秒	0 / 秒
cephfs_data	複製	cephfs	active+clean 256	3	209		replicated_rule	0%	<div></div>	<div></div>	0 / 秒	0 / 秒
cephfs_metadata	複製	cephfs	active+clean 64	3	210		replicated_rule	0%	<div></div>	<div></div>	0 / 秒	0 / 秒
default.rgw.buckets.index	複製	rgw	active+clean 8	3	75		replicated_rule	0%	<div></div>	<div></div>	0 / 秒	0 / 秒
default.rgw.control	複製	rgw	active+clean 8	3	25		replicated_rule	0%	<div></div>	<div></div>	0 / 秒	0 / 秒
default.rgw.log	複製	rgw	active+clean 8	3	30		replicated_rule	0%	<div></div>	<div></div>	0 / 秒	0 / 秒
default.rgw.meta	複製	rgw	active+clean 8	3	28		replicated_rule	0%	<div></div>	<div></div>	0 / 秒	0 / 秒
family_photos	複製	cephfs	active+clean 128	3	226		replicated_rule	0%	<div></div>	<div></div>	0 / 秒	0 / 秒
testing_rbd_pool	複製	cephfs, rbd	active+clean 128	3	76		replicated_rule	0%	<div></div>	<div></div>	0.8 / 秒	0 / 秒

0個選択/合計9個

図 5.1: プールのリスト

名前列のプール名の隣にあるドロップダウン矢印をクリックして、プールの詳細情報が含まれる拡張テーブルを表示してください。たとえば、一般的な詳細、パフォーマンスの詳細、設定などが表示されます。

## 5.1 新しいプールの追加

新しいプールを追加するには、プールテーブルの左上の作成をクリックします。プールフォームには、プール名、タイプ、アプリケーション、圧縮モード、最大バイト数と最大オブジェクト数のクォータを入力できます。この特定のプールに最適な配置グループの数は、プールのフォームそのものによって事前に計算されます。この計算は、クラスタ内のOSDの量、および選択したプールタイプとその特定の設定に基づきます。配置グループ数を手動で設定すると、すぐに計算された数に置き換えられます。プールの作成をクリックして確認します。

作成 Pool

名前 \* potato-pool ✓

プールタイプ \* replicated ✓ ▾

PG Autoscale on ✓ ▾

複製されたサイズ \* 3

アプリケーション cephfs ✕

CRUSH

Crushルールセット replicated\_rule ▾ ⓘ + ▢

圧縮

モード none ✓ ▾

Quotas

Max bytes ⓘ 例: 10GiB

Max objects ⓘ 0

作成 Pool キャンセル

図 5.2: 新しいプールの追加

## 5.2 プールの削除

プールを削除するには、そのテーブル行のプールを選択します。作成ボタンの横にあるドロップダウン矢印をクリックして、削除をクリックします。

## 5.3 プールのオプションの編集

プールのオプションを編集するには、そのテーブル行を選択してプールテーブルの左上の編集をクリックします。

プールの名前を変更したり、配置グループの数を増やしたり、プールのアプリケーションのリストや圧縮設定を変更したりできます。プールの編集をクリックして確認します。

## 6 RADOS Block Deviceの管理

利用可能なすべてのRBD (RADOS Block Devices)を一覧にするには、メインメニューからブロック > イメージをクリックします。

このリストにはデバイスの概要が含まれます。たとえば、デバイス名、関連するプール名、ネームスペース、デバイスのサイズ、デバイスのオブジェクト数とオブジェクトサイズ、デバイスのプロビジョニングの詳細、親などです。

イメージNamespacesごみ箱全体的なパフォーマンス

+ 作成

10

Q

名前	プール	Namespace	サイズ	オブジェクト数	オブジェクトサイズ	プロビジョニング読み込み	プロビジョニング読み込み合計	親
> example_rbd_device	rbd		4 MiB	1	4 MiB	0 B	0 B	-
> potato_rbd	rbd		10 MiB	3	4 MiB	0 B	0 B	-

0 選択済み / 2 合計

図 6.1: RBDイメージのリスト

### 6.1 RBDに関する詳細の表示

デバイスに関する詳細情報を表示するには、テーブルでそのデバイスの行をクリックします。

詳細	スナップショット	設定	パフォーマンス
名前	example_rbd_device		
プール	rbd		
データプール	-		
作成済み	2021/04/30 15:02:38		
サイズ	4 MiB		
オブジェクト数	1		
オブジェクトサイズ	4 MiB		
機能	<div> <span>deep-flatten</span> <span>exclusive-lock</span> <span>fast-diff</span> <span>layering</span> <span>object-map</span> </div>		
プロビジョニング済み	0 B		
プロビジョニング済み合計	0 B		
ストライピング単位	4 MiB		
ストライピング数	1		
親	-		
ブロック名のプレフィックス	rbd_data.37ff2b17a0d1		
順番	22		
Format Version	2		

図 6.2: RBDの詳細

## 6.2 RBDの設定の表示

デバイスの詳細な設定を表示するには、テーブルでそのデバイスの行をクリックし、続いてその下のテーブルで設定タブをクリックします。

名前	説明	キー	ソース
BPSバースト	希望する入出力バイト数のバースト上限。	rbd_qos_bps_burst	グローバル
BPS制限	希望する秒あたり入出力バイト数の上限。	rbd_qos_bps_limit	グローバル
IOPSバースト	希望する入出力操作数のバースト上限。	rbd_qos_iops_burst	グローバル
IOPS制限	希望する秒あたり入出力操作数の上限。	rbd_qos_iops_limit	グローバル
読み取りBPSバースト	希望する読み取りバイト数のバースト上限。	rbd_qos_read_bps_burst	グローバル
読み取りBPS制限	希望する秒あたり読み取りバイト数の上限。	rbd_qos_read_bps_limit	グローバル
読み取りIOPSバースト	希望する読み取り操作数のバースト上限。	rbd_qos_read_iops_burst	グローバル
読み取りIOPS制限	希望する秒あたり読み取り操作数の上限。	rbd_qos_read_iops_limit	グローバル
書き込みBPSバースト	希望する書き込みバイト数のバースト上限。	rbd_qos_write_bps_burst	グローバル
書き込みBPS制限	希望する秒あたり書き込みバイト数の上限。	rbd_qos_write_bps_limit	グローバル

図 6.3: RBD設定

## 6.3 RBDの作成

新しいデバイスを追加するには、テーブル見出しの左上の作成をクリックして、RBDの作成画面で次の操作を行います。

作成 RBD

名前 \*

example\_rbd\_device

プール \*

rbd

✓

☐ 専用のデータプールを使用してください

サイズ \*

例: 10GiB

✓

機能

☐ ディープフラット化

☒ 階層化

☐ 排他ロック

☐ オブジェクトマップ(排他ロックが必要)

☐ ジャーナリング(排他ロックが必要)

☐ Fast diff (interlocked with object-map)

詳細...

作成 RBD

キャンセル

図 6.4: 新しいRBDの追加

1. 新しいデバイスの名前を入力します。命名の制限については、『導入ガイド』、第2章「ハードウェア要件と推奨事項」、2.11項「名前の制限」を参照してください。
2. 新しいRBDデバイスの作成元となる、rbdアプリケーションが割り当てられたプールを選択します。
3. 新しいデバイスのサイズを指定します。
4. デバイスの追加オプションを指定します。デバイスのパラメータを微調整するには、詳細をクリックし、オブジェクトサイズ、ストライプユニット、またはストライプ数の値を入力します。QoS (サービス品質)の制限を入力するには、サービス品質をクリックし、制限値を入力します。

5. Create RBD (RBDの作成)をクリックして確認します。

## 6.4 RBDの削除

デバイスを削除するには、そのテーブル行を選択します。作成ボタンの横にあるドロップダウン矢印をクリックして、削除をクリックします。Delete RBD (RBDの削除)をクリックして削除を確認します。



### ヒント: RBDのごみ箱への移動

RBDの削除は元に戻せないアクションです。代わりにごみ箱に移動するを実行すると、後でメインテーブルのごみ箱タブでデバイスを選択し、テーブル見出しの左上の復元をクリックしてデバイスを復元できます。

## 6.5 RADOS Block Deviceのスナップショットの作成

RADOS Block Deviceのスナップショットを作成するには、そのテーブル行を選択してください。設定内容ペインが表示されます。スナップショットタブを選択し、テーブル見出しの左上にある作成をクリックします。スナップショットの名前を入力し、RBDスナップショットの作成をクリックして確認します。

スナップショットを選択した後で、デバイスの名前の変更、保護、複製、コピー、削除などの追加のアクションを実行できます。ロールバックは、現在のスナップショットからデバイスの状態を復元します。

詳細

スナップショット

設定

+ 作成

10

名前	サイズ	プロビジョニング済み	状態	作成済み
testing_rbd-20190215T095402Z	10MiB	0B	未保護	2/15/19 10:54:08 AM
testing_rbd-20190405T074138Z	10MiB	0B	未保護	4/5/19 9:41:42 AM

0個選択/合計2個

図 6.5: RBDのスナップショット

## 6.6 RBDミラーリング

RADOS Block Deviceイメージを2つのCephクラスタ間で非同期にミラーリングできます。Cephダッシュボードを使用して、2つ以上のクラスタ間でRBDイメージを複製するよう設定できます。この機能には2つのモードがあります。

### ジャーナルベース

このモードは、RBDイメージのジャーナリング機能を使用して、クラスタ間でクラッシュコンシステントなレプリケーションを保証します。

### スナップショットベース

このモードでは、RBDイメージのミラースナップショットを使用して、クラッシュコンシステントなRBDイメージをクラスタ間で複製します。ミラースナップショットはスケジュールに沿って定期的に作成するか、手動で作成します。

ジャーナルベースのミラーリングだけを使用している場合、ミラーリングはピアクラスタ内のプールごとに設定されます。また、プール内の特定のイメージサブセットに設定することや、プール内のすべてのイメージを自動的にミラーリングするように設定することもできます。

ミラーリングは`rbd`コマンドを使用して設定します。`rbd`はデフォルトでSUSE Enterprise Storage 7にインストール済みです。`rbd-mirror`デーモンは、リモートのピアクラスタからイメージの更新を取得して、ローカルクラスタ内のイメージに適用する処理を受け持ちます。`rbd-mirror`デーモンの有効化の詳細については、[6.6.2項「`rbd-mirror`デーモンの有効化](#)」を参照してください。

レプリケーションに対する要望に応じて、RADOS Block Deviceのミラーリングは単方向レプリケーション用または双方向レプリケーション用に設定できます。

### 単方向レプリケーション

データがプライマリクラスタからセカンダリクラスタにミラーリングされるだけであれば、`rbd-mirror`デーモンはセカンダリクラスタ上でのみ実行されます。

### 双方向レプリケーション

データが、あるクラスタのプライマリイメージから別のクラスタの非プライマリイメージにミラーリングされる場合(逆も同様)、`rbd-mirror`デーモンは両方のクラスタで実行されます。

## ！ 重要

`rbd-mirror`デーモンの各インスタンスは、ローカルとリモート両方のCephクラスタを同時に接続できなければなりません。たとえば、すべてのMonitorホストとOSDホストを接続する必要があります。さらに、ミラーリングのワークロードを扱うため、ネットワークの2つのデータセンター間には十分な帯域幅が必要です。

## 💡 ヒント: 一般情報

RADOS Block Deviceのミラーリングの一般情報とコマンドラインによるアプローチについては、[20.4項「RBDイメージのミラーリング」](#)を参照してください。

### 6.6.1 プライマリクラスタとセカンダリクラスタの設定

「プライマリ」クラスタは、イメージを含む元のプールが作成される場所です。「セカンダリ」クラスタは、「プライマリ」クラスタからプールやイメージが複製される場所です。

## 📄 注記: 相対命名

「プライマリ」および「セカンダリ」という用語は、クラスタよりも個々のプールとの関連性が高いため、レプリケーションのコンテキストでは相対的に使用できます。たとえば、双方向レプリケーションにおいて、あるプールを「プライマリ」クラスタから「セカンダリ」クラスタにミラーリングし、別のプールを「セカンダリ」クラスタから「プライマリ」クラスタにミラーリングできます。

### 6.6.2 `rbd-mirror`デーモンの有効化

次の手順では、`rbd`コマンドを使用してミラーリングを設定するための基本的な管理タスクを実行する方法を説明します。ミラーリングは、Cephクラスタ内のプールごとに設定します。プールの設定手順は、両方のピアクラスタで行う必要があります。これから説明する手順では、わかりやすくするため「プライマリ」および「セカンダリ」という名前の2つのクラスタが1つのホストからアクセス可能であることを想定しています。

`rbd-mirror`デーモンは、クラスタデータの実際のレプリケーションを実行します。

1. `ceph.conf`とキーリングファイルの名前を変更し、プライマリホストからセカンダリホストにコピーします。

```
cephuser@secondary > cp /etc/ceph/ceph.conf /etc/ceph/primary.conf
cephuser@secondary > cp /etc/ceph/ceph.admin.client.keyring \
/etc/ceph/primary.client.admin.keyring
cephuser@secondary > scp PRIMARY_HOST:/etc/ceph/ceph.conf \
/etc/ceph/secondary.conf
cephuser@secondary > scp PRIMARY_HOST:/etc/ceph/ceph.client.admin.keyring \
/etc/ceph/secondary.client.admin.keyring
```

2. **rbd**を使用してプールのミラーリングを有効にするには、**mirror pool enable**、プール名、ミラーリングモードを指定します。

```
cephuser@adm > rbd mirror pool enable POOL_NAME MODE
```



## 注記

ミラーリングモードはimageまたはpoolを指定できます。以下に例を示します。

```
cephuser@secondary > rbd --cluster primary mirror pool enable image-pool image
cephuser@secondary > rbd --cluster secondary mirror pool enable image-pool
image
```

3. Cephダッシュボードでブロック > ミラーリングに移動します。左側のデーモンテーブルに、アクティブに実行されている**rbd-mirror**デーモンとそのヘルスが表示されます。

## デーモン

<div> <div>🔄</div> <div>📊</div> <div>10</div> <div>🔍</div> <div>✕</div> </div>				
インスタンス ⇅	ID 📄	ホスト名 ⇅	バージョン ⇅	ヘルス ⇅
292255	test	doc-ses-min4	14.2.2-354-g8878cf2360	OK
合計1個				

図 6.6: **rbd-mirror**デーモンの実行

## 6.6.3 ミラーリングの無効化

**rbd**を使用してプールのミラーリングを無効化するには、**mirror pool disable**コマンドとプール名を指定します。

```
cephuser@adm > rbd mirror pool disable POOL_NAME
```

この方法でプールのミラーリングを無効にした場合、ミラーリングを明示的に有効にしたイメージ(プール内)のミラーリングも無効になります。

## 6.6.4 ピアのブートストラップ処理

`rbd-mirror`がピアクラスタを発見するためには、ピアをプールに登録し、ユーザアカウントを作成する必要があります。このプロセスは`rbd`コマンドとともに`mirror pool peer bootstrap create`コマンドと`mirror pool peer bootstrap import`コマンドを使用することで自動化できます。

`rbd`を使用して新しいブートストラップトークンを手動で作成するには、`mirror pool peer bootstrap create`コマンドとプール名に加えて、ローカルクラスタを記述するためにオプションのサイト名を指定します。

```
cephuser@adm > rbd mirror pool peer bootstrap create [--site-name local-site-name] pool-name
```

`mirror pool peer bootstrap create`コマンドの出力はトークンです。このトークンを`mirror pool peer bootstrap import`コマンドに提供する必要があります。たとえば、プライマリクラスタで次のコマンドを実行します。

```
cephuser@adm > rbd --cluster primary mirror pool peer bootstrap create --site-name primary
image-pool
eyJmc2lkIjo1OWY1MjgyZGI0Yjg5ODU0NTk2LTgwOTgtMzIwYzFmYzY5MmYzIiwia2V5Ijo1QVFBUnczOWQwdkhvQmhBQVlMM1I4RmR5dHNJQU50bkFTZ0l0TVE9PSIsIm1vb19ob3N0I
WlpcnJvcilwZWVyIiwia2V5Ijo1QVFBUnczOWQwdkhvQmhBQVlMM1I4RmR5dHNJQU50bkFTZ0l0TVE9PSIsIm1vb19ob3N0I
joiW3YyOjE5MjE4xNjguMS4zOjY4MjAsdjE6MTkyLjE2OC4xLjM6NjgyMV0ifQ==
```

`rbd`コマンドを使用して別のクラスタが作成したブートストラップトークンを手動でインポートするには、次の項目を指定します。`mirror pool peer bootstrap import`コマンド、プール名、作成されたトークンへのファイルパス(標準入力から読み込む場合は「-」)、および、オプションでローカルクラスタとミラーリング方向を記述するサイト名(デフォルトでは双方向ミラーリングを表す`rx-tx`に設定されていますが、単方向ミラーリングを表す`rx-only`にも設定できます)。

```
cephuser@adm > rbd mirror pool peer bootstrap import [--site-name local-site-name] \
[--direction rx-only or rx-tx] pool-name token-path
```

たとえば、セカンダリクラスタで次のコマンドを実行します。

```
cephuser@adm > cat >>EOF < token
```

```
eyJmc2lkIjojOWY1MjgyZGItYjg5OjU0NTk2LTgwOTgtMzIwYzFmYzM5NmYzIiwia2xpZW50X2lkIjoicmJkLWlpcn
\
JvcilwZWVyIiwia2V5IjojQVFBUnczOWQwdkhvQmhBQVlMM1I4RmR5dHNJQU50bkFTZ010TVE9PSIsIm1vbl9ob3N0I
\
joiW3YyOjE5Mi4xNjguMS4zOjY4MjAsdjE6MTkyLjE2OC4xLjM6NjgyMV0ifQ==
EOF
cephuser@adm > rbd --cluster secondary mirror pool peer bootstrap import --site-name
secondary image-pool token
```

## 6.6.5 クラスタピアの削除

**rbd**コマンドを使用して、ミラーリングピアCephクラスタを削除するには、**mirror pool peer remove**コマンド、プール名、ピアのUUIDを指定します(UUIDは**rbd mirror pool info**コマンドにより取得できます)。

```
cephuser@adm > rbd mirror pool peer remove pool-name peer-uuid
```

## 6.6.6 Cephダッシュボードによるプールのレプリケーション設定

RBDイメージをミラーリングできるようにするには、**rbd-mirror**デーモンがプライマリクラスタへのアクセス権を持っている必要があります。作業を始める前に、6.6.4項「ピアのブートストラップ処理」の手順に従って設定したかを確認してください。

1. 「プライマリ」および「セカンダリ」クラスタの両方で同じ名前のプールを作成し、プールに**rbd**アプリケーションを割り当てます。新しいプールの作成の詳細については、5.1項「新しいプールの追加」を参照してください。

作成 Pool

名前 \*

mirrored-pool

プールタイプ \*

replicated ✓ ⇅

PG Autoscale

off ✓ ⇅

配置グループ \*

4 ✓

計算のヘルプ

複製されたサイズ \*

3 ✓

アプリケーション

✎ rbd ✕

CRUSH

Crushルールセット

replicated\_rule ⇅ ⓘ + 削除

圧縮

モード

none ⇅

Quotas

Max bytes ⓘ

例: 10GiB

Max objects ⓘ

0

RBD設定

サービス品質 +

作成 Pool キャンセル

2. 「プライマリ」および「セカンダリ」クラスタの両方のダッシュボードで、ブロック > ミラーリングに移動します。右側のプールテーブルで、複製するプールの名前をクリックし、モードの編集をクリックした後で、レプリケーションモードを選択します。この例では、「プール」レプリケーションモードを使用します。つまり、指定したプール内のすべてのイメージが複製されます。更新をクリックして確認します。

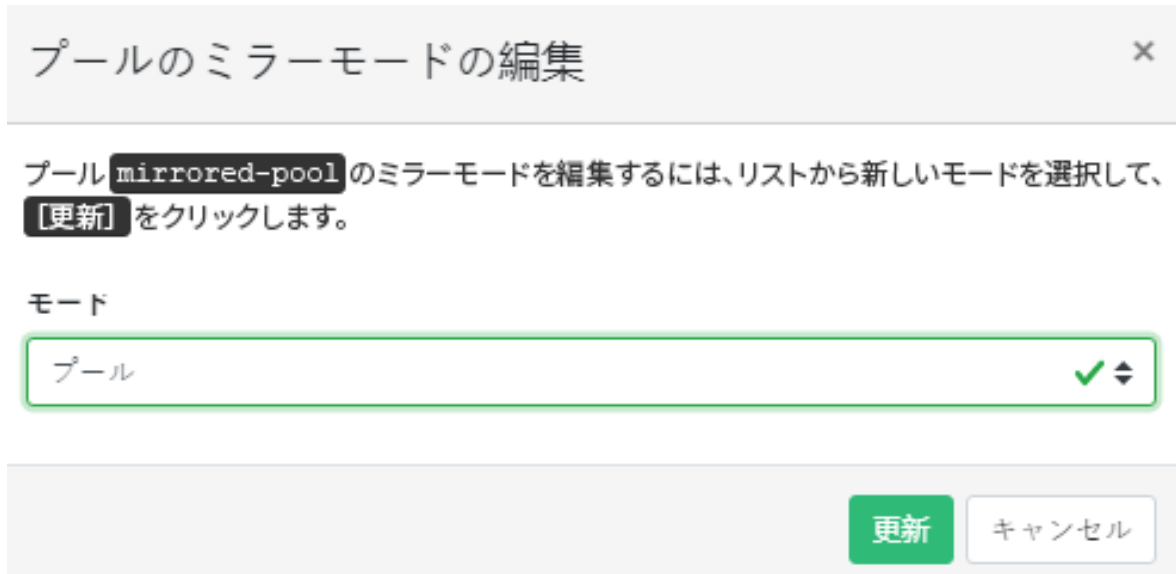


図 6.8: レプリケーションモードの設定

### ！ 重要: プライマリクラスタでのエラーまたは警告

レプリケーションモードを更新すると、対応する右側の列にエラーまたは警告フラグが表示されます。これは、まだプールにレプリケーション用のピアユーザーが割り当てられていないためです。ピアユーザーは「セカンダリ」クラスタにのみ割り当てるので、「プライマリ」クラスタではこのフラグは無視してください。

3. 「セカンダリ」クラスタのダッシュボードで、ブロック > ミラーリングに移動します。ピアの追加を選択してプールミラーピアを追加します。「プライマリ」クラスタの詳細を指定します。

プールのミラーピアの追加

プール `mirrored-pool` のプールのミラーピア属性を追加し、`[送信]` をクリックします。

クラスタ名 \*

プライマリ

CephX ID \*

rbd-mirror-peer

モニターアドレス

10.100.24.60,10.100.24.61,10.100.24.62

CephXキー

AQAIr5Vd4y/UMRAATF8ee/wnPF2x3P9DtmEP2Q==

送信

キャンセル

図 6.9: ピア資格情報の追加

#### クラスタ名

プライマリクラスタを識別する任意の固有の文字列(「primary」など)。このクラスタ名は、実際のセカンダリクラスタの名前とは異なる必要があります。

#### CephX ID

ミラーリングピアとして作成したCephユーザID。この例では、「rbd-mirror-peer」です。

#### モニターアドレス

プライマリクラスタのCeph MonitorノードのIPアドレスをカンマ区切りリスト。

#### CephXキー

ピアユーザIDに関連したキー。このキーを取得するには、プライマリクラスタで次のサンプルコマンドを実行します。

```
cephuser@adm > ceph auth print_key pool-mirror-peer-name
```

送信をクリックして確認します。

## プール

モードの編集

↺

📊

10

🔍

✕

名前	モード	リーダー	# ローカル	# リモート	ヘルス
example_rbd_pool	プール	292255	2	2	OK
mirrored-pool	プール	292255	0	0	OK
pool3	イメージ	292255	2	2	OK
pool4	プール	292255	1	1	OK

1個選択/合計4個

図 6.10: 複製されたプールのリスト

### 6.6.7 RBDイメージレプリケーションが機能することの確認

rbid-mirrorデーモンが実行されていて、CephダッシュボードでRBDイメージレプリケーションが設定されている場合、ここで、レプリケーションが実際に機能するかどうかを確認します。

1. 「プライマリ」 クラスタのCephダッシュボードで、RBDイメージを作成し、その親プールが、レプリケーション用にすでに作成済みのプールになるようにします。イメージに対して「排他ロック」機能と「ジャーナリング」機能を有効にします。RBDイメージの作成方法の詳細については、6.3項「RBDの作成」を参照してください。

## RBDの作成

名前 \*

mirrored-image1

プール \*

mirrored-pool

☐ 専用のデータプールを使用してください

サイズ \*

60GiB

機能

- ☐ ディープフラット化
- ☒ 階層化
- ☒ 排他ロック
- ☐ オブジェクトマップ(排他ロックが必要)
- ☒ ジャーナリング(排他ロックが必要)
- ☐ Fast diff (オブジェクトマップが必要)

[詳細...](#)

RBDの作成

キャンセル

2. 複製するイメージを作成した後で、「セカンダリ」クラスタのCephダッシュボードを開き、ブロック>ミラーリングに移動します。右側のプールテーブルは、#リモートイメージの数の変更を反映し、#ローカルイメージの数を同期します。

## プール

モードの編集					
名前	モード	リーダー	#ローカル	#リモート	ヘルス
example_rbd_pool	プール	292255	2	2	OK
mirrored-pool	プール	292255	1	1	OK
pool3	イメージ	292255	2	2	OK
pool4	プール	292255	1	1	OK
1個選択/合計4個					

図 6.12: 新しいRBDイメージの同期



## ヒント: レプリケーションの進行状況

ページ下部のイメージテーブルには、RBDイメージのレプリケーションのステータスが表示されます。問題タブには、考えられる問題が含まれており、同期中タブには、イメージのレプリケーションの進行状況が表示され、準備完了タブには、レプリケーションが正常に完了したすべてのイメージが一覧にされます。

### イメージ

問題 同期中 準備完了			
プール	イメージ	説明	状態
mirrored-pool	mirrored-image1	replaying, master_position=[object_number=3, tag_tid=1, entry_tid=3], mirror_position=[object_number=3, tag_tid=1, entry_tid=3], entries_behind_master=0	再生中
pool3	img1	replaying, master_position=[object_number=1, tag_tid=2, entry_tid=6401], mirror_position=[object_number=1, tag_tid=2, entry_tid=6401], entries_behind_master=0	再生中
pool3	new_image1	replaying, master_position=[object_number=1, tag_tid=3, entry_tid=641], mirror_position=[object_number=1, tag_tid=3, entry_tid=641], entries_behind_master=0	再生中
pool4	img4	replaying, master_position=[object_number=3, tag_tid=1, entry_tid=3], mirror_position=[object_number=3, tag_tid=1, entry_tid=3], entries_behind_master=0	再生中
合計6個			

図 6.13: RBDイメージのレプリケーションステータス

3. 「プライマリ」クラスタで、データをRBDイメージに書き込みます。「セカンダリ」クラスタのCephダッシュボードで、ブロック>イメージに移動して、プライマリクラスタのデータが書き込まれるにつれて、対応するイメージのサイズが大きくなっているかどうかを監視します。

## 6.7 iSCSI Gatewayの管理



### ヒント: iSCSI Gatewayについての詳細

iSCSI Gatewayの全般的な情報については、第22章「Ceph iSCSI Gateway」を参照してください。

利用可能なすべてのゲートウェイとマップ済みイメージを一覧にするには、メインメニューからブロック>iSCSIをクリックします。概要タブを開くと、現在設定されているiSCSI Gatewayとマップ済みのRBDイメージが一覧にされます。

ゲートウェイテーブルには、各ゲートウェイの状態、iSCSIターゲットの数、およびセッションの数が一覧にされます。イメージテーブルには、各マップ済みイメージの名前、関連するプール名のバックストアタイプ、および他の統計情報の詳細が一覧にされます。

ターゲットタブには、現在設定されているiSCSIターゲットが一覧にされます。

概要 ターゲット				
+ 作成 Discovery authentication				
ターゲット	ポータル	イメージ	# Sessions	
iqn.2001-07.com.ceph:1619785904397	master.ses7-mini.test:10.20.165.200	rbd/example_rbd_device_potato	0	
iqn.2001-07.com.ceph:1619785974221	master.ses7-mini.test:192.168.121.185	rbd/potato-rbd	0	
0 選択済み / 2 合計				

図 6.14: iSCSIターゲットのリスト

ターゲットのより詳細な情報を表示するには、そのテーブル行のドロップダウン矢印をクリックしてください。ツリー構造のスキーマが開き、ディスク、ポータル、イニシエータ、およびグループが一覧にされます。項目をクリックして展開し、その詳細コンテンツを表示します。オプションで右側の表に関連する設定が表示されます。

概要

ターゲット

作成

検出認証の

10

Q

×

ターゲット	ポータル	イメージ	# Sessions
iqn.2001-07.com.ceph:1597683071527	node1.asettle-dashbord.test:10.20.164.201	rbd/example_rbd_device_potato	0

iSCSIトポロジ

rbd/example\_rbd\_device\_potato

iqn.2001-07.com.ceph:1597683071527

Disks

rbd/example\_rbd\_device\_potato

Portals

node1.asettle-dashbord.test:10.20.164.201

Initiators

Groups

名前

現在の

デフォルト

バックストア	ユーザー: rbd (tcmu-runner)	rbd
hw_max_sectors	1024	1024
lun	0	
max_data_area_mb	8	8
osd_op_timeout	30	30
qfull_timeout	5	5
wwn	bf60abfd-9159-4098-bc9b-2be4daefa5c	
7 合計		

1 選択済み / 2 合計

図 6.15: iSCSIターゲット詳細

### 6.7.1 iSCSIターゲットの追加

新しいiSCSIターゲットを追加するには、ターゲットテーブルの左上の作成をクリックして、必要な情報を入力します。

作成 Target

ターゲットIQN \*

iqn.2001-07.com.ceph:1620133668760

ポータル \*

アイテムが選択されていません。

+ ポータルの追加

イメージ

アイテムが選択されていません。

+ イメージの追加

☐ ACL認証

ユーザ

パスワード

相互ユーザ

相互パスワード

作成 Target

キャンセル

図 6.16: 新しいターゲットの追加

1. 新しいゲートウェイのターゲットアドレスを入力します。
2. Add portal (ポータルの追加)をクリックして、リストから1つまたは複数のiSCSIポータルを選択します。

3. Add image (イメージの追加)をクリックして、ゲートウェイのRBDイメージを1つまたは複数選択します。
4. 認証を使用してゲートウェイにアクセスする必要がある場合は、ACL認証チェックボックスをオンにして資格情報を入力します。Mutual authentication (相互認証)およびDiscovery authentication (ディスカバリ認証)を有効にすると、より高度な認証オプションが表示されます。
5. Create Target (ターゲットの作成)をクリックして確認します。

## 6.7.2 iSCSIターゲットの編集

既存のiSCSIターゲットを編集するには、ターゲットテーブルでその行をクリックし、テーブルの左上の編集をクリックします。

その後、iSCSIターゲットの追加、ポータルの追加または削除、および関連するRBDイメージの追加または削除を行うことができます。ゲートウェイの認証情報を調整することもできます。

## 6.7.3 iSCSIターゲットの削除

iSCSIターゲットを削除するには、テーブル行を選択してから編集ボタンの隣にあるドロップダウン矢印をクリックして、削除を選択します。はいを有効にして、iSCSIターゲットの削除をクリックして確認します。

# 6.8 RBD QoS (サービス品質)



### ヒント: 詳細の参照先

RBD QoS設定オプションの全般的な情報と説明については、[20.6項「QoS設定」](#)を参照してください。

QoSオプションは次の複数のレベルで設定できます。

- グローバル
- プールごと
- イメージごと

「グローバル」の設定はリストの一番上にあり、新しく作成されたすべてのRBDイメージと、プールまたはRBDイメージ層でこれらの値を上書きしないイメージに使用されます。グローバルに指定されたオプション値をプールごとまたはイメージごとに上書きできます。プールに対して指定されたオプションは、そのプールのすべてのRBDイメージに適用されます。ただし、イメージに対して設定された設定オプションが優先される場合を除きます。イメージに対して指定されたオプションは、プールに対して指定されたオプションを上書きし、グローバルに指定されたオプションを上書きします。

このようにして、デフォルトをグローバルに定義して、特定のプールのすべてのRBDイメージに適合させ、個々のRBDイメージのプール設定を上書きできます。

## 6.8.1 オプションのグローバルな設定

RADOS Block Deviceをグローバルに設定するには、メインメニューからクラスタ > 設定を選択してください。

1. レベルの横に、すべての利用可能なグローバル設定オプションを一覧にするには、ドロップダウンメニューから詳細を選択します。
2. 検索フィールドで`rbd_qos`をフィルタリングして、テーブルの結果をフィルタします。これにより、QoSで使用可能なすべての設定オプションが一覧にされます。
3. 値を変更するには、テーブルでその行をクリックし、テーブルの左上の編集を選択します。編集ダイアログには、値を指定するための6つの異なるフィールドが含まれます。mgrテキストボックスには、RBD設定オプションの値を入力する必要があります。



### 注記

他のダイアログとは異なり、このダイアログでは接頭辞付きの単位で値を指定できません。編集するオプションによっては、バイトまたはIOPSのいずれかでこれらの値を設定する必要があります。

## 6.8.2 新しいプールでのオプションの設定

新しいプールを作成し、そのプールにRBD設定オプションを設定するには、プール > 作成をクリックします。プールタイプとして`replicated` (複製)を選択します。RBD QoSオプションを設定できるようにするには、プールに`rbd`アプリケーションタグを追加する必要があります。



## 注記

イレージャコーディングプールにRBD QoS設定オプションを設定することはできません。イレージャコーディングプールにRBD QoSオプションを設定するには、RBDイメージの複製されたメタデータプールを編集する必要があります。これにより、そのイメージのイレージャコーディングデータプールに設定が適用されます。

### 6.8.3 既存のプールでのオプションの設定

既存のプールにRBD QoSオプションを設定するには、プールをクリックし、プールのテーブル行をクリックして、テーブルの左上の編集を選択します。

ダイアログにRBD設定セクションが表示され、その後にサービス品質セクションが表示されます。



## 注記

RBD設定セクションもサービス品質セクションも表示されない場合は、RBD設定オプションの設定に使用できない「イレージャコーディング」プールを編集しているか、RBDイメージで使用するようプールが設定されていない可能性があります。後者の場合は、RBDアプリケーションタグをプールに割り当てます。これにより、対応する設定セクションが表示されます。

### 6.8.4 設定オプション

設定オプションを展開するには、サービス品質+をクリックします。使用可能なすべてのオプションのリストが表示されます。設定オプションの単位は、テキストボックスにすでに表示されています。BPS (1秒あたりのバイト数)オプションの場合は、「1M」や「5G」などのショートカットを自由に使用できます。これらはそれぞれ自動的に「1 MB/秒」と「5 GB/秒」に変換されます。

各テキストボックスの右にあるリセットボタンをクリックすると、プールに設定されている値が削除されます。グローバルに設定されたオプションやRBDイメージに設定されたオプションの設定値は削除されません。

## 6.8.5 新しいRBDイメージを使用したRBD QoSオプションの作成

イメージにRBD QoSオプションが設定された状態でRBDイメージを作成するには、ブロックイメージを選択し、作成をクリックします。詳細設定セクションを展開するには、詳細...をクリックします。使用可能なすべての設定オプションを開くには、サービス品質+をクリックします。

## 6.8.6 既存のイメージでのRBD QoSの編集

既存のイメージでRBD QoSオプションを編集するには、ブロックイメージを選択し、プールのテーブル行をクリックして、最後に編集をクリックします。編集ダイアログが表示されます。詳細設定セクションを展開するには、詳細...をクリックします。使用可能なすべての設定オプションを開くには、サービス品質+をクリックします。

## 6.8.7 イメージをコピーまたは複製する際の設定オプションの変更

RBDイメージを複製またはコピーする場合、デフォルトでは、その特定のイメージに設定された値もコピーされます。コピーまたは複製時にRBDイメージを変更する場合は、RBDイメージを作成または編集する場合と同様に、コピー/複製ダイアログで、更新された設定値を指定できます。この場合、コピーまたは複製するRBDイメージの値のみが設定(またはリセット)されます。この操作では、ソースRBDイメージの設定もグローバル設定も変更されません。

コピー/複製時にオプションの値をリセットすることを選択した場合、そのオプションの値は、そのイメージに設定されません。つまり、親プールに値が設定されている場合、親プールに指定されたそのオプションの値が使用されます。親プールに値が設定されていない場合は、グローバルなデフォルトが使用されます。

## 7 NFS Ganeshaの管理



### ヒント: NFS Ganeshaについての詳細

NFS Ganeshaの全般的な情報については、[第25章「NFS Ganesha」](#)を参照してください。

利用可能なすべてのNFSエクスポートを一覧にするには、メインメニューからNFSをクリックします。

各エクスポートのディレクトリ、デーモンのホスト名、ストレージバックエンドのタイプ、およびアクセスタイプがリストに表示されます。

+ 作成						
	パス	疑似	クラスタ	デーモン	ストレージバックエンド	アクセスタイプ
>	/potato/potato	/exportimus-maximus	ganesha-sesdev_nfs		CephFS	MDONLY_RO
>	/root	/exportcephfs	ganesha-sesdev_nfs		CephFS	RW
>	/root/potato	/exportpotato	ganesha-sesdev_nfs		CephFS	MDONLY
選択0/合計3						

図 7.1: NFSエクスポートのリスト

NFSエクスポートの詳細情報を表示するには、テーブルでそのテーブル行をクリックします。

詳細	
クライアント(0)	
アクセスタイプ	RW
CephFSファイルシステム	sesdev_fs
CephFSユーザ	admin
クラスタ	ganesha-sesdev_nfs
デーモン	
NFSプロトコル	NFSv3, NFSv4
パス	/root
疑似	/exportcephfs
Squash	no_root_squash
ストレージバックエンド	CephFS
トランスポート	TCP, UDP

図 7.2: NFSエクスポートの詳細

## 7.1 NFSエクスポートの作成

新しいNFSエクスポートを追加するには、エクスポートテーブルの左上にある作成をクリックして、必要な情報を入力します。

NFSエクスポートの作成

クラスタ *	ganesha-sesdev_nfs	⬆
デーモン	アイテムが選択されていません。 <span>+ デーモンの追加</span>	
ストレージバックエンド *	CephFS	✓ ⬆
CephFSユーザID *	admin	✓ ⬆
CephFS名 *	sesdev_fs	✓ ⬆
セキュリティラベル	<input type="checkbox"/> セキュリティラベルの有効化	
CephFSパス *	/root	✓
	新しいディレクトリが作成されます	
NFSプロトコル *	<input checked="" type="checkbox"/> NFSv3 <input checked="" type="checkbox"/> NFSv4	
NFSタグ ?		
疑似 * ?	/exportcephfs	✓
アクセスタイプ *	RW	✓ ⬆
	すべての操作を許可します	
スカッシュ *	no_root_squash	✓ ⬆
トランスポートプロトコル *	<input checked="" type="checkbox"/> UDP <input checked="" type="checkbox"/> TCP	
クライアント	任意のクライアントからアクセスできます <span>+ クライアントの追加</span>	

NFSエクスポートの作成
キャンセル

図 7.3: 新しいNFSエクスポートの追加

1. エクスポートを実行するNFS Ganeshaデーモンを1つ以上選択します。
2. ストレージのバックエンドを選択します。



## 重要

現時点でサポートされているのは、CephFSにより支援されているNFSエクスポートだけです。

3. ユーザIDと、その他のバックエンド関連オプションを選択します。
4. NFSエクスポートのディレクトリパスを入力します。指定したディレクトリがサーバ上に存在しない場合、新しく作成されます。
5. 他のNFS関連オプションを指定します。たとえば、サポートされるNFSプロトコルのバージョン、疑似、アクセスタイプ、スカッシュ、トランスポートプロトコルなどです。
6. アクセスを特定のクライアントだけに制限する必要がある場合、クライアントの追加をクリックして、クライアントのIPアドレスと共に、アクセスタイプと(ルート権限無効化オプションを指定します。
7. NFSエクスポートの作成をクリックして確認します。

## 7.2 NFSエクスポートの削除

エクスポートを削除するには、そのテーブル行でエクスポートを選択して強調表示します。編集ボタンの横にあるドロップダウン矢印をクリックして、削除を選択します。はいチェックボックスをオンにし、NFSエクスポートの削除をクリックして確認します。

## 7.3 NFSエクスポートの編集

既存のエクスポートを編集するには、そのテーブル行でエクスポートを選択して強調表示し、エクスポートテーブルの左上にある編集をクリックします。

その後、NFSエクスポートのすべての詳細を調整できます。

NFSエクスポートの編集

クラスタ *	ganesha-sesdev_nfs	
デーモン	アイテムが選択されていません。 <span>+ デーモンの追加</span>	
ストレージ バックエンド *	CephFS	
CephFS ユーザID *	admin	
CephFS名 *	sesdev_fs	
セキュリティ ティラベル	<input type="checkbox"/> セキュリティラベルの有効化	
CephFSパス *	/root	
NFS プロトコル *	<input checked="" type="checkbox"/> NFSv3 <input checked="" type="checkbox"/> NFSv4	
NFSタグ ?		
疑似 ? *	/exportcephfs	
アクセス タイプ *	RW すべての操作を許可します	
スカッシュ *	no_root_squash	
トランスポート プロトコル *	<input checked="" type="checkbox"/> UDP <input checked="" type="checkbox"/> TCP	
クライアント	任意のクライアントからアクセスできます <span>+ クライアントの追加</span>	

NFSエクスポートの編集
キャンセル

図 7.4: NFSエクスポートの編集

## 8 CephFSの管理



### ヒント: 詳細の参照先

CephFSの詳細情報については、[第23章「クラスタファイルシステム」](#)を参照してください。

## 8.1 CephFSの概要の表示

設定されているファイルシステムの概要を表示するには、メインメニューからファイルシステムをクリックします。メインテーブルには、各ファイルシステムの名前、作成日、および有効かどうか表示されます。

ファイルシステムのテーブル行をクリックすると、そのランクと、ファイルシステムに追加されたプールの詳細が表示されます。

名前	作成済み	有効化済み
sesdev_fs	2021/04/30 12:30:19	✓

ランク	状態	デーモン	アクティビティ	dエントリ	iノード
0	active	sesdev_fs.master.ingr	Reqs: 0 /s	10	13

プール	タイプ	サイズ	使用量
cephfs.sesdev_fs. data		13.1 GiB	0%
cephfs.sesdev_fs. metadata		13.1 GiB	0%

図 8.1: CEPHFSの詳細

画面の下部に、関連するMDS iノードとクライアント要求の数をリアルタイムで収集した統計情報カウントが表示されます。

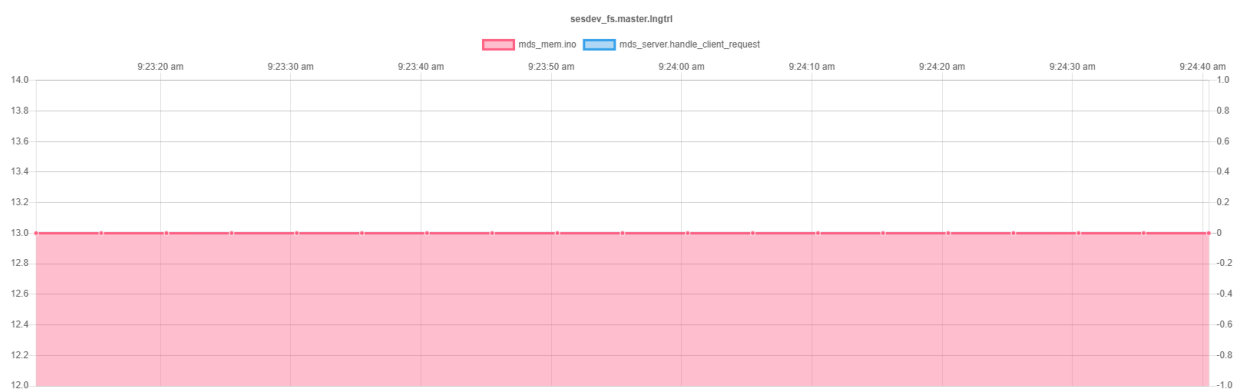


図 8.2: CEPHFSの詳細

## 9 Object Gatewayの管理

### ！ 重要

Cephダッシュボード上でObject Gatewayのフロントエンドにアクセスしようとする  
と、次の通知が表示される場合があります。

#### Information

No RGW credentials found, please consult the documentation on how to enable RGW for the dashboard.

Please consult the documentation on how to configure and enable the Object Gateway management functionality.

これは、cephadmによるObject GatewayのCephダッシュボード向け自動設定が行われていないためです。この通知が表示された場合、[10.4項「Object Gateway管理フロントエンドの有効化」](#)の手順に従って、Object GatewayのCephダッシュボード用フロントエンドを手動で有効化してください。

### 💡 ヒント: Object Gatewayの詳細

Object Gatewayの全般的な情報については、[第21章「Ceph Object Gateway」](#)を参照してください。

## 9.1 Object Gatewayの表示

設定されているObject Gatewayのリストを表示するには、オブジェクトゲートウェイ・デーモンをクリックします。このリストには、ゲートウェイのID、ゲートウェイデーモンが実行されているクラスタノードのホスト名、およびゲートウェイのバージョン番号が含まれます。

ゲートウェイの詳細情報を表示するには、ゲートウェイ名の横にあるドロップダウン矢印をクリックします。パフォーマンスカウンタタブには、読み込み/書き込み操作とキャッシュ統計情報の詳細が表示されます。

詳細	パフォーマンスカウンタ	パフォーマンスの詳細
arch	x86_64	
ceph_release	octopus	
ceph_version	cephバージョン15.2.4-557-g4ac763f0b3 (4ac763f0b3864d9168bc4a46fef26d7fa759545e) octopus (安定)	
ceph_version_short	15.2.4-557-g4ac763f0b3	
container_hostname	node1	
container_image	registry.suse.de/devel/storage/7.0/containers/ses/7/ceph/ceph	
cpu	Intel Coreプロセッサ(Haswell、TSXなし)	
distro	sles	
distro_description	SUSE Linux Enterprise Server 15 SP2	
distro_version	15.2	
frontend_config#0	beast port=80	
frontend_type#0	beast	
hostname	node1	
kernel_description	#1 SMP Wed Jul 29 18:54:11 UTC 2020 (dbe0add)	
kernel_version	5.3.18-24.9-default	
mem_swap_kb	0	
mem_total_kb	4020668	
num_handles	1	
os	Linux	
pid	1	
zone_id	2a664005-94ad-432a-b873-d563fed68496	
zone_name	デフォルト	
zonegroup_id	cc4ec3c6-c611-4bfd-a155-e3e05552d5cd	
zonegroup_name	デフォルト	

図 9.1: ゲートウェイの詳細

## 9.2 Object Gatewayユーザの管理

既存のObject Gatewayユーザのリストを表示するには、オブジェクトゲートウェイ ユーザをクリックします。

ユーザアカウントの詳細を表示するには、ユーザ名の横にあるドロップダウン矢印をクリックします。表示される情報は、ステータス情報やユーザとバケットクォータの詳細などです。

詳細

キー

ユーザ名	rgw-admin
氏名	admin
中断済み	いいえ
システム	はい
最大バケット数	1000

ユーザクォータ

有効化済み	いいえ
最大サイズ	-
最大オブジェクト数	-

バケットクォータ

有効化済み	いいえ
最大サイズ	-
最大オブジェクト数	-

図 9.2: ゲートウェイユーザ

## 9.2.1 新しいゲートウェイユーザの追加

新しいゲートウェイユーザを追加するには、テーブル見出しの左上の作成をクリックします。資格情報、S3キー、ユーザとバケットクォータの詳細を入力し、ユーザの作成をクリックして確認します。

ユーザの作成

ユーザ名 \*

example\_rgw\_user

✓

氏名 \*

Example User

✓

電子メール  
アドレス

example@user.com

✓

最大  
バケット数

カスタム

✓ ⇅

1000

☐ 中断済み

S3キー

☒ キーの自動生成

ユーザクォータ

☐ 有効化済み

バケットクォータ

☒ 有効化済み

☒ 無制限のサイズ

☒ 無制限のオブジェクト数

ユーザの作成

キャンセル

図 9.3: 新しいゲートウェイユーザの追加

## 9.2.2 ゲートウェイユーザの削除

ゲートウェイユーザを削除するには、ユーザを選択して強調表示します。編集の横のドロップダウンボタンをクリックし、リストから削除を選択してユーザアカウントを削除します。はいチェックボックスをオンにし、ユーザの削除をクリックして確認します。

## 9.2.3 ゲートウェイユーザの詳細の編集

ゲートウェイユーザの詳細を変更するには、ユーザを選択して強調表示します。テーブル見出しの左上にある編集をクリックします。

機能、キー、サブユーザ、クォータ情報など、基本または追加のユーザ情報を変更します。ユーザの編集をクリックして確認します。

キータブには、ゲートウェイユーザ、およびそのアクセスキーと秘密鍵の読み込み専用リストが含まれます。キーを表示するには、リストでユーザ名をクリックし、テーブル見出しの左上の表示を選択します。S3キーダイアログで、「目」のアイコンをクリックしてキーを表示するか、クリップボードアイコンをクリックして関連するキーをクリップボードにコピーします。

# 9.3 Object Gatewayバケットの管理

OGW (Object Gateway)バケットは、OpenStack Swiftコンテナの機能を実装しています。Object Gatewayバケットは、データオブジェクトを保存するためのコンテナとして機能します。

オブジェクトゲートウェイバケットをクリックして、オブジェクトゲートウェイバケットのリストを表示します。

## 9.3.1 新しいバケットの追加

新しいオブジェクトゲートウェイバケットを追加するには、テーブル見出しの左上の作成をクリックします。バケット名を入力し、所有者を選択し、配置ターゲットを設定します。バケットの作成をクリックして確認します。



### 注記

この時点で有効化済みを選択してロックを有効化できますが、作成後にも設定可能です。詳細については、[9.3.3項「バケットの編集」](#)を参照してください。

### 9.3.2 バケットの詳細の表示

Object Gatewayバケットの詳細情報を表示するには、バケット名の隣にあるドロップダウン矢印をクリックします。

詳細	
名前	エクスポート
ID	2a664005-94ad-432a-b873-d563fed68496.14523.1
所有者	rgw-admin
インデックスタイプ	通常
配置ルール	default-placement
マーカー	2a664005-94ad-432a-b873-d563fed68496.14523.1
最大マーカー	0#,1#,2#,3#,4#,5#,6#,7#,8#,9#,10#
バージョン	0#1,1#,2#,3#,4#,5#,6#,7#,8#,9#,10#1
マスタバージョン	0#0,1#,2#,3#,4#,5#,6#,7#,8#,9#,10#0
変更時間	8/24/20 1:24:34 PM
ゾーングループ	cc4ec3c6-c611-4bfd-a155-e3e05552d5cd
バージョンニング	中断済み
MFA削除	無効化済み
バケットクォータ	
有効化済み	いいえ
最大サイズ	無制限
最大オブジェクト数	無制限
ロック	
有効化済み	いいえ

図 9.4: ゲートウェイバケットの詳細



#### ヒント: バケットクォータ

詳細テーブルの下で、バケットクォータとロック設定の詳細を確認できます。

### 9.3.3 バケットの編集

バケットを選択して強調表示してから、テーブル見出しの左上にある編集をクリックします。

バケットの所有者の更新と、バージョニング、多要素認証、ロックの有効化ができます。変更後、バケットの編集をクリックして確認します。

バケットの編集

Id

eaf156f1-e787-4c5c-8e86-06cba6481d65.44187.1

名前

root

所有者 \*

asettle

✓

配置ターゲット

default-placement

バージョニング

☐ 有効化済み ⓘ

多要素認証

☐ 削除が有効 ⓘ

ロック

☐ 有効化済み ⓘ

バケットの編集

キャンセル

図 9.5: バケットの詳細の編集

### 9.3.4 バケットの削除

Object Gatewayバケットを削除するには、バケットを選択して強調表示します。編集の横のドロップダウンボタンをクリックし、リストから削除を選択してバケットを削除します。はいチェックボックスをオンにし、バケットの削除をクリックして確認します。

## 10 手動設定

このセクションでは、コマンドラインでダッシュボードを手動で設定したいユーザ向けの詳細情報を紹介します。

### 10.1 TLS/SSLサポートの設定

ダッシュボードとのすべてのHTTP接続は、デフォルトでTLS/SSLによって保護されています。セキュア接続にはSSL証明書が必要です。自己署名証明書を使用することも、証明書を生成して既知のCA (認証局)で署名することもできます。



#### ヒント: SSLの無効化

特定の理由がある場合、SSLのサポートを無効にできます。たとえば、SSLをサポートしないプロキシの後方でダッシュボードを実行する場合などです。

SSLを無効にする場合は注意してください。「ユーザ名とパスワード」は「暗号化されずに」ダッシュボードに送信されます。

SSLを無効にするには、次のコマンドを実行します。

```
cephuser@adm > ceph config set mgr mgr/dashboard/ssl false
```



#### ヒント: Ceph Managerプロセスの再起動

SSL証明書とキーを変更した後で、Ceph Managerプロセスを手動で再起動する必要があります。このためには、次のコマンドを実行します。

```
cephuser@adm > ceph mgr fail ACTIVE-MANAGER-NAME
```

または、ダッシュボードモジュールを無効化して再度有効化することもできます。この場合、マネージャが自身を再起動します。

```
cephuser@adm > ceph mgr module disable dashboard  
cephuser@adm > ceph mgr module enable dashboard
```

## 10.1.1 自己署名証明書の作成

セキュア通信用の自己署名証明書の作成は簡単です。これにより、ダッシュボードの実行を高速化できます。



### 注記: Webブラウザの警告

ほとんどのWebブラウザでは、自己署名証明書を使用すると警告が表示され、ダッシュボードへのセキュアな接続を確立するには明示的に確認する必要があります。

自己署名証明書を生成してインストールするには、次の組み込みコマンドを使用します。

```
cephuser@adm > ceph dashboard create-self-signed-cert
```

## 10.1.2 CA署名証明書の使用

ダッシュボードへの接続を適切に保護し、Webブラウザで自己署名証明書に関する警告が表示されないようにするため、CAによって署名された証明書を使用することをお勧めします。次のようなコマンドを使用して証明書キーペアを生成できます。

```
root # openssl req -new -nodes -x509 \
-subj "/O=IT/CN=ceph-mgr-dashboard" -days 3650 \
-keyout dashboard.key -out dashboard.crt -extensions v3_ca
```

上のコマンドは、`dashboard.key`ファイルと`dashboard.crt`ファイルを出力します。`dashboard.crt`ファイルがCAによって署名されたら、次のコマンドを実行して、すべてのCeph Managerインスタンスに対してその証明書を有効にします。

```
cephuser@adm > ceph config-key set mgr/dashboard/crt -i dashboard.crt
cephuser@adm > ceph config-key set mgr/dashboard/key -i dashboard.key
```



### ヒント: 各マネージャインスタンスに対して異なる証明書が必要な場合

Ceph Managerの各インスタンスに対して異なる証明書が必要な場合は、次のようにコマンドを変更してインスタンスの名前を含めます。NAMEは、Ceph Managerインスタンスの名前(通常は関連するホスト名)で置き換えます。

```
cephuser@adm > ceph config-key set mgr/dashboard/NAME/crt -i dashboard.crt
cephuser@adm > ceph config-key set mgr/dashboard/NAME/key -i dashboard.key
```

## 10.2 ホスト名とポート番号の変更

Cephダッシュボードは特定のTCP/IPアドレスとTCPポートにバインドされます。デフォルトでは、ダッシュボードをホストする現在アクティブなCeph Managerは、TCPポート8443 (SSLが無効な場合は8080)にバインドされます。



### 注記

Ceph Manager(および、Cephダッシュボード)を実行しているホストでファイアウォールが有効化されている場合、ポートにアクセスできるように設定変更が必要な場合があります。Ceph向けファイアウォール設定の詳細については、『Troubleshooting Guide』、第13章「Hints and tips」、13.7項「Firewall settings for Ceph」を参照してください。

Cephダッシュボードは、デフォルトで「::」にバインドされます。これは、使用可能なすべてのIPv4およびIPv6アドレスに対応します。次のコマンドを使用すると、すべてのCeph Managerインスタンスに適用されるようにWebアプリケーションのIPアドレスとポート番号を変更できます。

```
cephuser@adm > ceph config set mgr mgr/dashboard/server_addr IP_ADDRESS
cephuser@adm > ceph config set mgr mgr/dashboard/server_port PORT_NUMBER
```



### ヒント: Ceph Managerインスタンスの個別の設定

各ceph-mgrデーモンは専用のダッシュボードインスタンスをホストするため、インスタンスを個別に設定しなければならない場合があります。次のコマンドを使用して、特定のマネージャインスタンスのIPアドレスとポート番号を変更します(NAMEをceph-mgrのIDで置き換えます)。

```
cephuser@adm > ceph config set mgr mgr/dashboard/NAME/server_addr IP_ADDRESS
cephuser@adm > ceph config set mgr mgr/dashboard/NAME/server_port PORT_NUMBER
```



### ヒント: 設定済みエンドポイントの一覧

**ceph mgr services**コマンドは、現在設定されているすべてのエンドポイントを表示します。dashboardキーを検索して、ダッシュボードにアクセスするためのURLを取得します。

## 10.3 ユーザ名とパスワードの調整

デフォルトの管理者アカウントを使用しない場合は、別のユーザアカウントを作成して、それを少なくとも1つの役割に関連付けます。事前定義済みの一連のシステム役割が用意されており、これらの役割を使用できます。詳細については、[第11章「コマンドラインによるユーザと役割の管理」](#)を参照してください。

管理者特権を持つユーザを作成するには、次のコマンドを使用します。

```
cephuser@adm > ceph dashboard ac-user-create USER_NAME PASSWORD administrator
```

## 10.4 Object Gateway管理フロントエンドの有効化

ダッシュボードのObject Gateway管理機能を使用するには、`system`フラグが有効なユーザのログインアカウント情報を指定する必要があります。

1. `system`フラグが設定されたユーザがない場合は、作成します。

```
cephuser@adm > radosgw-admin user create --uid=USER_ID --display-name=DISPLAY_NAME --system
```

コマンドが出力する`access_key`と`secret_key`を記録します。

2. `radosgw-admin`コマンドを使用して、既存のユーザの資格情報を取得することもできます。

```
cephuser@adm > radosgw-admin user info --uid=USER_ID
```

3. 受け取った資格情報をダッシュボードに提供します。

```
cephuser@adm > ceph dashboard set-rgw-api-access-key ACCESS_KEY  
cephuser@adm > ceph dashboard set-rgw-api-secret-key SECRET_KEY
```



### 注記

SUSE Linux Enterprise Server 15 SP2では、デフォルトでファイアウォールが有効です。ファイアウォール設定の詳細については、『Troubleshooting Guide』、第13章「Hints and tips」、13.7項「Firewall settings for Ceph」を参照してください。

考慮すべき点があります。

- Object Gatewayのホスト名とポート番号は自動的に決定されます。
- 複数のゾーンを使用している場合、マスタゾーングループとマスタゾーン内のホストは自動的に決定されます。ほとんどのセットアップではこれで十分ですが、状況によってはホスト名とポートを手動で設定したい場合があります。

```
cephuser@adm > ceph dashboard set-rgw-api-host HOST
cephuser@adm > ceph dashboard set-rgw-api-port PORT
```

- 次の追加設定が必要になる場合があります。

```
cephuser@adm > ceph dashboard set-rgw-api-scheme SCHEME # http or https
cephuser@adm > ceph dashboard set-rgw-api-admin-resource ADMIN_RESOURCE
cephuser@adm > ceph dashboard set-rgw-api-user-id USER_ID
```

- Object Gatewayのセットアップで自己署名証明書(10.1項「TLS/SSLサポートの設定」)を使用している場合は、不明なCAによって署名された証明書、またはホスト名が一致しないことによって接続が拒否されないようにするため、ダッシュボードで証明書の検証を無効にします。

```
cephuser@adm > ceph dashboard set-rgw-api-ssl-verify False
```

- Object Gatewayで要求の処理に時間がかかりすぎて、ダッシュボードがタイムアウトする場合は、タイムアウト値を調整できます(デフォルトは45秒)。

```
cephuser@adm > ceph dashboard set-rest-requests-timeout SECONDS
```

## 10.5 iSCSI管理の有効化

Cephダッシュボードは、iSCSIターゲットを管理します。これには、Ceph iSCSIゲートウェイの `rbd-target-api` サービスが提供するREST APIを使用します。REST APIがインストール済みで、iSCSIゲートウェイ上で有効化されていることを確認します。



### 注記

CephダッシュボードのiSCSI管理機能は、`ceph-iscsi` プロジェクトの最新版であるバージョン3に依存しています。使用しているオペレーティングシステムが適切なバージョンであることを確認してください。さもなければ、CephダッシュボードはiSCSI管理機能を使用できません。

ceph-iscsi REST APIがHTTPSモードに設定されており、自己署名証明書を使用している場合、ceph-iscsi APIにアクセスした際のSSL証明書の検証を回避するようにダッシュボードを設定します。

API SSL検証を無効化します。

```
cephuser@adm > ceph dashboard set-iscsi-api-ssl-verification false
```

利用可能なiSCSIゲートウェイを定義します。

```
cephuser@adm > ceph dashboard iscsi-gateway-list
cephuser@adm > ceph dashboard iscsi-gateway-add scheme://username:password@host[:port]
cephuser@adm > ceph dashboard iscsi-gateway-rm gateway_name
```

## 10.6 Single Sign-Onを有効にする

SSO (「シングルサインオン」)は、ユーザが複数のアプリケーションに1つのIDとパスワードで同時にログインできるアクセス制御方法です。

Cephダッシュボードは、SAML 2.0プロトコルを介したユーザの外部認証をサポートしています。「権限付与」は引き続きダッシュボードによって実行されるため、まずユーザアカウントを作成し、それを目的の役割に関連付ける必要があります。ただし、「認証」プロセスは、既存のIdP (「Identity Provider」)によって実行できます。

シングルサインオンを設定するには、次のコマンドを使用します。

```
cephuser@adm > ceph dashboard sso setup saml2 CEPH_DASHBOARD_BASE_URL \
  IDP_METADATA IDP_USERNAME_ATTRIBUTE \
  IDP_ENTITY_ID SP_X_509_CERT \
  SP_PRIVATE_KEY
```

パラメータは次の通りです。

### CEPH\_DASHBOARD\_BASE\_URL

Cephダッシュボードにアクセス可能なベースURL(たとえば、「https://cephdashboard.local」)。

### IDP\_METADATA

IdPメタデータXMLのURL、ファイルパス、または内容(たとえば、「https://myidp/metadata」)。

### IDP\_USERNAME\_ATTRIBUTE

オプション。認証応答からユーザ名を取得するために使用される属性。デフォルトは「uid」。

IDP\_ENTITY\_ID

オプション。IdPメタデータに複数のエンティティIDが存在する場合に使用します。

SP\_X\_509\_CERT / SP\_PRIVATE\_KEY

オプション。署名と暗号化のためにCephダッシュボード(サービスプロバイダ)によって使用される証明書のファイルパスまたは内容。



## 注記: SAML要求

SAML要求の発行者の値は次のパターンに従います。

```
CEPH_DASHBOARD_BASE_URL/auth/saml2/metadata
```

SAML 2.0の現在の設定を表示するには、次のコマンドを実行します。

```
cephuser@adm > ceph dashboard sso show saml2
```

シングルサインオンを無効にするには、次のコマンドを実行します。

```
cephuser@adm > ceph dashboard sso disable
```

SSOが有効かどうかを確認するには、次のコマンドを実行します。

```
cephuser@adm > ceph dashboard sso status
```

SSOを有効にするには、次のコマンドを実行します。

```
cephuser@adm > ceph dashboard sso enable saml2
```

## 11 コマンドラインによるユーザと役割の管理

このセクションでは、Cephダッシュボードによって使用されるユーザアカウントを管理する方法について説明します。これは、ユーザアカウントを作成または変更する場合や、適切なユーザ役割と許可を設定する場合に役立ちます。

### 11.1 パスワードポリシーの管理

デフォルトでは、次のチェックを含むパスワードポリシー機能が有効になっています。

- パスワードはN文字より長い。
- 古いパスワードと新しいパスワードが同じか。

次のコマンドにより、パスワードポリシー機能全体をON/OFFに切り替えることができます。

```
cephuser@adm > ceph dashboard set-pwd-policy-enabled true|false
```

次のチェック項目を個別にON/OFFに切り替えることも可能です。

```
cephuser@adm > ceph dashboard set-pwd-policy-check-length-enabled true|false
cephuser@adm > ceph dashboard set-pwd-policy-check-oldpwd-enabled true|false
cephuser@adm > ceph dashboard set-pwd-policy-check-username-enabled true|false
cephuser@adm > ceph dashboard set-pwd-policy-check-exclusion-list-enabled true|false
cephuser@adm > ceph dashboard set-pwd-policy-check-complexity-enabled true|false
cephuser@adm > ceph dashboard set-pwd-policy-check-sequential-chars-enabled true|false
cephuser@adm > ceph dashboard set-pwd-policy-check-repetitive-chars-enabled true|false
```

また、パスワードポリシーの動作設定に次のオプションを利用できます。

- 最小のパスワード長(デフォルトは8文字)。

```
cephuser@adm > ceph dashboard set-pwd-policy-min-length N
```

- 最低限のパスワードの複雑さ(デフォルトは10)。

```
cephuser@adm > ceph dashboard set-pwd-policy-min-complexity N
```

パスワードの複雑さは、パスワードに含まれる各文字を分類することで算出されます。

- パスワードに使用できない単語をカンマで区切ったリスト。

```
cephuser@adm > ceph dashboard set-pwd-policy-exclusion-list word[,...]
```

## 11.2 ユーザアカウントの管理

Cephダッシュボードは、複数のユーザアカウントの管理をサポートしています。各ユーザアカウントは、ユーザ名、パスワード(bcryptを使用して暗号化された形式で保存)、オプションの名前、およびオプションの電子メールアドレスで構成されます。

ユーザアカウントはCeph Monitorの設定データベースに保存され、すべてのCeph Managerインスタンス間でグローバルに共有されます。

ユーザアカウントを管理するには、次のコマンドを使用します。

### 既存のユーザの表示

```
cephuser@adm > ceph dashboard ac-user-show [USERNAME]
```

### 新しいユーザの作成

```
cephuser@adm > ceph dashboard ac-user-create USERNAME [PASSWORD] [ROLENAME] [NAME] [EMAIL]
```

### ユーザの削除

```
cephuser@adm > ceph dashboard ac-user-delete USERNAME
```

### ユーザのパスワードの変更

```
cephuser@adm > ceph dashboard ac-user-set-password USERNAME PASSWORD
```

### ユーザの名前と電子メールの変更

```
cephuser@adm > ceph dashboard ac-user-set-info USERNAME NAME EMAIL
```

### ユーザの無効化

```
cephuser@adm > ceph dashboard ac-user-disable USERNAME
```

### ユーザの有効化

```
cephuser@adm > ceph dashboard ac-user-enable USERNAME
```

## 11.3 ユーザの役割と許可

このセクションでは、ユーザの役割に割り当てることができるセキュリティスコープ、ユーザの役割を管理する方法、およびユーザの役割をユーザアカウントに割り当てする方法について説明します。

### 11.3.1 セキュリティスコープの定義

ユーザアカウントは、ユーザがダッシュボードのどの部分にアクセスできるかを定義する一連の役割に関連付けられます。ダッシュボードの各部分は、「セキュリティ」スコープ内でグループ化されます。セキュリティスコープは事前定義されていて静的です。現在のところ、次のセキュリティスコープを使用できます。

#### **hosts**

ホストメニューエントリに関連するすべての機能が含まれます。

#### **config-opt**

Ceph設定オプションの管理に関連するすべての機能が含まれます。

#### **pool**

プールの管理に関連するすべての機能が含まれます。

#### **osd**

Ceph OSDの管理に関連するすべての機能が含まれます。

#### **monitor**

Ceph Monitorの管理に関連するすべての機能が含まれます。

#### **rbd-image**

RADOS Block Deviceイメージの管理に関連するすべての機能が含まれます。

#### **rbd-mirroring**

RADOS Block Deviceのミラーリングの管理に関連するすべての機能が含まれます。

#### **iscsi**

iSCSIの管理に関連するすべての機能が含まれます。

#### **rgw**

Object Gatewayの管理に関連するすべての機能が含まれます。

#### **cephfs**

CephFSの管理に関連するすべての機能が含まれます。

#### **manager (マネージャ)**

Ceph Managerの管理に関連するすべての機能が含まれます。

#### **log**

Cephのログの管理に関連するすべての機能が含まれます。

## grafana

Grafanaプロキシに関連するすべての機能が含まれます。

## prometheus

Prometheusアラート管理に関連するすべての機能が含まれます。

## dashboard-settings

ダッシュボードの設定を変更できます。

## 11.3.2 ユーザの役割の指定

「役割」は、「セキュリティスコープ」と一連の「許可」との間の一連のマッピングを指定するものです。許可には、「read」、「create」、「update」、および「delete」の4つのタイプがあります。

次の例では、ユーザがプールの管理に関連する機能に対して「read」許可と「create」許可を持ち、RBDイメージの管理に関連する機能に対して完全な許可を持つ役割を指定します。

```
{
  'role': 'my_new_role',
  'description': 'My new role',
  'scopes_permissions': {
    'pool': ['read', 'create'],
    'rbd-image': ['read', 'create', 'update', 'delete']
  }
}
```

ダッシュボードには、「システム役割」と呼ばれる、事前定義済みの一連の役割があらかじめ用意されています。これらの役割は、Cephダッシュボードを新規インストールした後ですぐに使用できます。

### 管理者

すべてのセキュリティスコープに対する完全な許可を提供します。

### 読み込み専用

ダッシュボード設定を除くすべてのセキュリティスコープの読み込み許可を提供します。

### block-manager

「rbd-image」、「rbd-mirroring」、および「iscsi」のスコープに対する完全な許可を提供します。

### rgw-manager

「rgw」スコープに対する完全な許可を提供します。

## cluster-manager

「hosts」、「osd」、「monitor」、「manager」、および「config-opt」のスコープに対する完全な許可を提供します。

## pool-manager

「pool」スコープに対する完全な許可を提供します。

## cephfs-manager

「cephfs」スコープに対する完全な許可を提供します。

### 11.3.2.1 カスタム役割の管理

次のコマンドを使用して、新しいユーザの役割を作成できます。

#### 新規ロール(役割)を作成します:

```
cephuser@adm > ceph dashboard ac-role-create ROLENAME [DESCRIPTION]
```

#### 役割の削除:

```
cephuser@adm > ceph dashboard ac-role-delete ROLENAME
```

#### 役割へのスコープ許可の追加

```
cephuser@adm > ceph dashboard ac-role-add-scope-perms ROLENAME SCOPENAME PERMISSION  
[PERMISSION...]
```

#### 役割からのスコープ許可の削除

```
cephuser@adm > ceph dashboard ac-role-del-perms ROLENAME SCOPENAME
```

### 11.3.2.2 ユーザアカウントへの役割の割り当て

役割をユーザに割り当てるには、次のコマンドを使用します。

#### ユーザの役割の設定

```
cephuser@adm > ceph dashboard ac-user-set-roles USERNAME ROLENAME [ROLENAME ...]
```

#### ユーザへの追加の役割の追加

```
cephuser@adm > ceph dashboard ac-user-add-roles USERNAME ROLENAME [ROLENAME ...]
```

## ユーザからの役割の削除

```
cephuser@adm > ceph dashboard ac-user-del-roles USERNAME ROLENAME [ROLENAME ...]
```



### ヒント: カスタム役割の消去

カスタム役割を作成し、後で**ceph.purge**ランナを使ってCephクラスタを削除する場合、まずカスタム役割を消去する必要があります。詳細については、[13.9項「Cephクラスタ全体の削除」](#)を参照してください。

### 11.3.2.3 例: ユーザとカスタム役割の作成

このセクションでは、RBDイメージの管理、Cephプールの表示と作成、および他のスコープへの読み込み専用アクセスを行うことができるユーザアカウントの作成手順について説明します。

1. tuxという名前の新しいユーザを作成します。

```
cephuser@adm > ceph dashboard ac-user-create tux PASSWORD
```

2. 役割を作成し、スコープ許可を指定します。

```
cephuser@adm > ceph dashboard ac-role-create rbd/pool-manager
cephuser@adm > ceph dashboard ac-role-add-scope-perms rbd/pool-manager \
  rbd-image read create update delete
cephuser@adm > ceph dashboard ac-role-add-scope-perms rbd/pool-manager pool read
  create
```

3. 役割をtuxユーザに関連付けます。

```
cephuser@adm > ceph dashboard ac-user-set-roles tux rbd/pool-manager read-only
```

## 11.4 プロキシ設定

Cephダッシュボードへの固定URLを定めたい場合や、管理ノードへの直接接続を許可したくない場合は、プロキシを設定できます。このプロキシは受け取った要求現在をアクティブなceph-mgrインスタンスへ自動的に転送します。

## 11.4.1 リバースプロキシによるダッシュボードへのアクセス

リバースプロキシ設定を使用してダッシュボードにアクセスしている場合は、URLプレフィックスを使用してダッシュボードを提供しなければならないことがあります。ダッシュボードがプレフィックスを含むハイパーリンクを使用できるようにするには、`url_prefix`設定を設定できます。

```
cephuser@adm > ceph config set mgr mgr/dashboard/url_prefix URL_PREFIX
```

これにより、`http://HOST_NAME:PORT_NUMBER/URL_PREFIX/`でダッシュボードにアクセスできます。

## 11.4.2 リダイレクションの無効化

CephダッシュボードがHAProxyのようなロードバランシングプロキシの背後にある場合、リダイレクション動作を無効化することで、内部URL(解決できないURL)がフロントエンドのクライアントに公開されないようにします。次のコマンドを使用することで、ダッシュボードがアクティブなダッシュボードへリダイレクトするのではなく、HTTPエラー(デフォルトでは500)を返すようになります。

```
cephuser@adm > ceph config set mgr mgr/dashboard/standby_behaviour "error"
```

デフォルトのリダイレクション動作に設定をリセットするには、次のコマンドを使用します。

```
cephuser@adm > ceph config set mgr mgr/dashboard/standby_behaviour "redirect"
```

## 11.4.3 エラーステータスコードの設定

リダイレクション動作を無効化する場合、スタンバイダッシュボードのHTTPステータスコードをカスタマイズする必要があります。そのためには、次のコマンドを実行します。

```
cephuser@adm > ceph config set mgr mgr/dashboard/standby_error_status_code 503
```

## 11.4.4 HAProxyの設定例

以下に示す設定は、HAProxyを使用したTLS/SSLパススルーの例です。



## 注記

この設定が効果を発揮する状況は次の通りです。ダッシュボードがフェールオーバーした際に、フロントエンドのクライアントはHTTPリダイレクト応答(303)を受け取る可能性があります。この場合、クライアントは解決不能ホストにリダイレクトされてしまいます。

このような状況は、HAProxyの2つのヘルスチェック中にフェールオーバーが発生した場合に生じます。これは、それまでアクティブだったダッシュボードが、新しいアクティブノードをリダイレクト先とする303エラーで応答してしまうためです。このような状況を避けるため、スタンバイノードのリダイレクション動作を無効化することを検討する必要があります。

```
defaults
    log global
    option log-health-checks
    timeout connect 5s
    timeout client 50s
    timeout server 450s

frontend dashboard_front
    mode http
    bind *:80
    option httplog
    redirect scheme https code 301 if !{ ssl_fc }

frontend dashboard_front_ssl
    mode tcp
    bind *:443
    option tcplog
    default_backend dashboard_back_ssl

backend dashboard_back_ssl
    mode tcp
    option httpchk GET /
    http-check expect status 200
    server x HOST:PORT ssl check verify none
    server y HOST:PORT ssl check verify none
    server z HOST:PORT ssl check verify none
```

## 11.5 API要求の監査

CephダッシュボードのREST APIは、PUT、POST、およびDELETEの各要求をCeph監査ログに記録できます。ログはデフォルトでは無効ですが、次のコマンドで有効にできます。

```
cephuser@adm > ceph dashboard set-audit-api-enabled true
```

ログが有効な場合、要求ごとに次のパラメータがログに記録されます。

**from**

「https://[::1]:44410」などの要求の発信元。

**path**

/api/authなどのREST APIパス。

**method**

「PUT」、「POST」、または「DELETE」。

**user**

ユーザの名前(または「None」)。

次に、ログエントリの例を示します。

```
2019-02-06 10:33:01.302514 mgr.x [INF] [DASHBOARD] \
from='https://[::ffff:127.0.0.1]:37022' path='/api/rgw/user/exu' method='PUT' \
user='admin' params='{ "max_buckets": "1000", "display_name": "Example User", "uid":
"exu", "suspended": "0", "email": "user@example.com" }'
```



### ヒント: 要求ペイロードのログの無効化

要求ペイロード(引数とその値のリスト)のログは、デフォルトで有効になっています。これは、次のコマンドを使用して無効にできます。

```
cephuser@adm > ceph dashboard set-audit-api-log-payload false
```

## 11.6 CephダッシュボードによるNFS Ganeshaの設定

Cephダッシュボードは、CephFSまたはObject Gatewayをバックストアとして使用するNFS Ganeshaエクスポートを管理できます。ダッシュボードはCephFSクラスタのRADOSオブジェクトに保存されたNFS Ganeshaの設定ファイルを管理します。NFS Ganeshaは設定の一部をCephクラスタに保存する必要があります。

NFS Ganeshaの設定オブジェクトの場所を設定するには、次のコマンドを実行します。

```
cephuser@adm > ceph dashboard set-ganesha-clusters-rados-pool-
namespace pool_name[/namespace]
```

これで、Cephダッシュボードを使用してNFS Ganeshaのエクスポートを管理できます。

### 11.6.1 複数のNFS Ganeshaクラスタの設定

Cephダッシュボードは、異なるNFS Ganeshaクラスタに属するNFS Ganeshaエクスポートの管理をサポートしています。各NFS Ganeshaクラスタの設定オブジェクトを異なるRADOSプール/ネームスペースに保存することをお勧めします。これは、設定を互いに分離するためです。

各NFS Ganeshaクラスタの設定の場所を指定するには、次のコマンドを使用します。

```
cephuser@adm > ceph dashboard set-ganesha-clusters-rados-pool-namespace cluster_id:pool_name[/namespace](,cluster_id:pool_name[/namespace])*
```

`Cluster_id`は任意の文字列で、NFS Ganeshaクラスタを一意に識別します。

Cephダッシュボードと複数のNFS Ganeshaクラスタを設定した場合、エクスポートがどのクラスタに属しているかをWeb UIによって自動的に選択できるようになります。

## 11.7 デバッグ用プラグイン

Cephダッシュボードのプラグインはダッシュボードの機能を拡張します。デバッグプラグインを使用すると、デバッグモードによってダッシュボードの動作をカスタマイズできるようになります。次のコマンドによって、デバッグモードの有効化、無効化、確認ができます。

```
cephuser@adm > ceph dashboard debug status
Debug: 'disabled'
cephuser@adm > ceph dashboard debug enable
Debug: 'enabled'
cephuser@adm > dashboard debug disable
Debug: 'disabled'
```

デフォルトでは、この機能は無効です。このモードは運用環境の展開に推奨される設定です。必要に応じて、再起動せずにデバッグモードを有効化できます。

## II クラスタの運用

- 12 クラスタの状態の判断 84
- 13 運用タスク 112
- 14 Cephサービスの運用 133
- 15 バックアップおよび復元 138
- 16 監視とアラート 141

## 12 クラスタの状態の判断

実行中のクラスタがある場合、**ceph**ツールを使用して監視できます。一般的に、クラスタの状態の判断には、Ceph OSD、Ceph Monitor、配置グループ、およびメタデータサーバのステータスを確認します。



### ヒント: 対話モード

**ceph**ツールをインタラクティブモードで実行するには、コマンドラインで引数を付けずに「**ceph**」と入力します。インタラクティブモードは、1行に多くの**ceph**コマンドを入力する場合に便利です。例:

```
cephuser@adm > ceph
ceph> health
ceph> status
ceph> quorum_status
ceph> mon stat
```

## 12.1 クラスタの状態の確認

**ceph status**か**ceph -s**を使用して、クラスタの直近の状態を確認できます。

```
cephuser@adm > ceph -s
cluster:
  id:      b4b30c6e-9681-11ea-ac39-525400d7702d
  health:  HEALTH_OK

services:
  mon: 5 daemons, quorum ses-min1,ses-master,ses-min2,ses-min4,ses-min3 (age 2m)
  mgr: ses-min1.gpijpm(active, since 3d), standbys: ses-min2.oopvyh
  mds: my_cephfs:1 {0=my_cephfs.ses-min1.oterul=up:active}
  osd: 3 osds: 3 up (since 3d), 3 in (since 11d)
  rgw: 2 daemons active (myrealm.myzone.ses-min1.kwwazo, myrealm.myzone.ses-min2.jngabw)

task status:
  scrub status:
    mds.my_cephfs.ses-min1.oterul: idle

data:
  pools:   7 pools, 169 pgs
  objects: 250 objects, 10 KiB
  usage:   3.1 GiB used, 27 GiB / 30 GiB avail
```

出力に表示される情報は、次のとおりです。

- クラスタID
- クラスタのヘルス状態
- Monitorマップのエポック、およびMonitor定数の状態
- Monitorマップのエポック、およびOSDの状態
- Ceph Managerのステータス
- Object Gatewayのステータス
- 配置グループのマップバージョン
- 配置グループとプールの数
- 保存データの「名目上」の量と、保存オブジェクトの数
- 保存データの合計量



### ヒント: Cephによるデータ使用量の計算方法

usedの値は、未加工ストレージの実際の使用量を反映します。xxx GB / xxx GBの値は、クラスタの全体的なストレージ容量の利用可能な量(より少ない数)を意味します。名目上の数は、複製、クローン作成、またはスナップショット作成前の保存データのサイズを反映します。したがって、実際の保存データの量は名目上の量より大きくなるのが一般的です。Cephは、データのレプリカを作成し、クローンやスナップショットの作成にもストレージ容量を使用することがあるためです。

直近の状態を表示するその他のコマンドは次のとおりです。

- ceph pg stat
- ceph osd pool stats
- ceph df
- ceph df detail

リアルタイムに更新された情報を取得するには、次のコマンドのいずれか(ceph -sを含む)をwatchコマンドの引数として実行します。

```
root # watch -n 10 'ceph -s'
```

監視を終了する場合は、**Ctrl-C** キーを押します。

## 12.2 クラスタのヘルスの確認

クラスタの起動後、データの読み込みや書き込みを開始する前に、クラスタのヘルスを確認します。

```
cephuser@adm > ceph health
HEALTH_WARN 10 pgs degraded; 100 pgs stuck unclean; 1 mons down, quorum 0,2 \
node-1,node-2,node-3
```



### ヒント

設定またはキーリングにデフォルト以外の場所を指定した場合、その場所を指定できません。

```
cephuser@adm > ceph -c /path/to/conf -k /path/to/keyring health
```

Cephクラスタは、次のいずれかのヘルスコードを返します。

#### OSD\_DOWN

1つ以上のOSDにダウン状態を示すマークが付いています。OSDデーモンが停止されているか、ピアOSDがネットワーク経由でOSDに接続できない可能性があります。一般的な原因として、デーモンの停止またはクラッシュ、ホストのダウン、ネットワークの停止などがあります。

ホストが正常である場合、デーモンは起動していて、ネットワークは機能しています。デーモンがクラッシュした場合は、そのデーモンのログファイル(/var/log/ceph/ceph-osd.\*)にデバッグ情報が記述されていることがあります。

#### OSD\_crush\_type\_DOWN (例: OSD\_HOST\_DOWN)

特定のCRUSHサブツリー内のすべてのOSD (たとえば、特定のホスト上のすべてのOSD) にダウン状態を示すマークが付いています。

#### OSD\_ORPHAN

OSDがCRUSHマップ階層で参照されていますが、存在しません。次のコマンドを使用して、OSDをCRUSH階層から削除できます。

```
cephuser@adm > ceph osd crush rm osd.ID
```

## OSD\_OUT\_OF\_ORDER\_FULL

「backfillfull」(デフォルトは0.90)、「nearfull」(デフォルトは0.85)、「full」(デフォルトは0.95)、または「failsafe\_full」、あるいはこれらすべての使用量のしきい値が昇順になっていません。特に、**backfillfull < nearfull**、**nearfull < full**、**full < failsafe\_full**となっている必要があります。

現在の値を読み込むには、次のコマンドを実行します。

```
cephuser@adm > ceph health detail
HEALTH_ERR 1 full osd(s); 1 backfillfull osd(s); 1 nearfull osd(s)
osd.3 is full at 97%
osd.4 is backfill full at 91%
osd.2 is near full at 87%
```

次のコマンドを使用してしきい値を調整できます。

```
cephuser@adm > ceph osd set-backfillfull-ratio ratio
cephuser@adm > ceph osd set-nearfull-ratio ratio
cephuser@adm > ceph osd set-full-ratio ratio
```

## OSD\_FULL

1つ以上のOSDが**full**のしきい値を超えており、クラスタは書き込みを実行できません。次のコマンドを使用して、プールごとの使用量を確認できます。

```
cephuser@adm > ceph df
```

次のコマンドを使用して、現在定義されている**full**の比率を確認できます。

```
cephuser@adm > ceph osd dump | grep full_ratio
```

書き込み可用性を復元するための短期的な回避策は、fullのしきい値を少し高くすることです。

```
cephuser@adm > ceph osd set-full-ratio ratio
```

さらにOSDを展開してクラスタに新しいストレージを追加するか、既存のデータを削除して領域を解放します。

## OSD\_BACKFILLFULL

1つ以上のOSDが**backfillfull**のしきい値を超えており、データをこのデバイスにリバランスできません。これは、リバランスを完了できないこと、およびクラスタが満杯に近付いていることを示す早期警告です。次のコマンドを使用して、プールごとの使用量を確認できます。

```
cephuser@adm > ceph df
```

## OSD\_NEARFULL

1つ以上のOSDが**nearfull**のしきい値を超えています。これは、クラスタが満杯に近付いていることを示す早期警告です。次のコマンドを使用して、プールごとの使用量を確認できます。

```
cephuser@adm > ceph df
```

## OSDMAP\_FLAGS

関心のあるクラスタフラグが1つ以上設定されています。「full」を除き、これらのフラグは次のコマンドを使用して設定またはクリアできます。

```
cephuser@adm > ceph osd set flag  
cephuser@adm > ceph osd unset flag
```

次のようなフラグがあります。

### full

クラスタにfullのフラグが付いており、書き込みを実行できません。

### pauserd、pausewr

読み込みまたは書き込みを一時停止しました。

### noup

OSDの起動が許可されていません。

### nodown

OSD障害レポートが無視されているため、MonitorはOSDに「down」のマークを付けません。

### noin

以前に「out」のマークが付けられているOSDには、起動時に「in」のマークは付けられません。

### noout

設定した間隔が経過した後、「down」状態のOSDに自動的に「out」のマークは付けられません。

### nobackfill、norecover、norebalance

回復またはデータリバランスは中断されます。

### noscrub、nodeep\_scrub

スクラブ(17.6項「[配置グループのスクラブ](#)」を参照)は無効化されます。

## **notieragent**

キャッシュ階層化アクティビティは中断されます。

## **OSD\_FLAGS**

1つ以上のOSDに、関心のあるOSDごとのフラグが付いています。次のようなフラグがあります。

### **noup**

OSDの起動が許可されていません。

### **nodown**

このOSD障害レポートは無視されます。

### **noin**

障害後に、このOSDにすでに自動的に「out」のマークが付けられている場合、起動時に「in」のマークは付けられません。

### **noout**

このOSDがダウンしている場合、設定した間隔が経過した後に自動的に「out」のマークは付けられません。

次のコマンドを使用して、OSDごとのフラグを設定およびクリアできます。

```
cephuser@adm > ceph osd add-flag osd-ID  
cephuser@adm > ceph osd rm-flag osd-ID
```

## **OLD\_CRUSH\_TUNABLES**

CRUSHマップは非常に古い設定を使用しており、更新する必要があります。

このヘルス警告をトリガさせることなく使用できる最も古い調整可能パラメータ(すなわち、クラスタに接続できる最も古いクライアントバージョン)は、mon\_crush\_min\_required\_version設定オプションで指定します。

## **OLD\_CRUSH\_STRAW\_CALC\_VERSION**

CRUSHマップは、strawバケットの中間重み値の計算に、最適ではない古い手法を使用しています。新しい手法を使用するようCRUSHマップを更新する必要があります(straw\_calc\_version=1)。

## **CACHE\_POOL\_NO\_HIT\_SET**

使用量を追跡するためのヒットセットが1つ以上のキャッシュプールに設定されておらず、階層化エージェントはキャッシュからフラッシュまたは削除するコールドオブジェクトを識別できません。次のコマンドを使用して、キャッシュプールにヒットセットを設定できます。

```
cephuser@adm > ceph osd pool set poolname hit_set_type type
```

```
cephuser@adm > ceph osd pool set poolname hit_set_period period-in-seconds
cephuser@adm > ceph osd pool set poolname hit_set_count number-of-hitsets
cephuser@adm > ceph osd pool set poolname hit_set_fpp target-false-positive-rate
```

## OSD\_NO\_SORTBITWISE

Luminous v12より前のOSDは実行されていませんが、`sortbitwise`フラグが付いていません。Luminous v12以上のOSDを起動する前に、`sortbitwise`フラグを設定する必要があります。

```
cephuser@adm > ceph osd set sortbitwise
```

## POOL\_FULL

1つ以上のプールがクォータに達しており、書き込みをこれ以上許可していません。次のコマンドを使用して、プールのクォータと使用量を設定できます。

```
cephuser@adm > ceph df detail
```

次のコマンドを使用して、プールのクォータを増加できます。

```
cephuser@adm > ceph osd pool set-quota poolname max_objects num-objects
cephuser@adm > ceph osd pool set-quota poolname max_bytes num-bytes
```

または、既存のデータを削除して使用量を削減できます。

## PG\_AVAILABILITY

データ可用性が低下しています。つまり、クラスタは、クラスタ内の一部のデータに対する潜在的な読み込みまたは書き込み要求を実行できません。具体的には、1つ以上のPGがI/O要求の実行を許可していない状態です。問題があるPGの状態には、「peering」、「stale」、「incomplete」、および「active」の欠如などがあります(これらの状態がすぐにはクリアされない場合)。影響を受けるPGについての詳しい情報は、次のコマンドを使用して参照できます。

```
cephuser@adm > ceph health detail
```

ほとんどの場合、根本原因は、現在1つ以上のOSDがダウンしていることです。次のコマンドを使用して、問題がある特定のPGの状態を問い合わせることができます。

```
cephuser@adm > ceph tell pgid query
```

## PG\_DEGRADED

一部のデータのデータ冗長性が低下しています。つまり、一部のデータについて必要な数のレプリカがクラスタにないか(複製プールの場合)、イレージャコードのフラグメントがクラスタにありません(イレージャコーディングプールの場合)。具体的には、1つ以上のPGに「degraded」または「undersized」のフラグが付いているか(クラスタ内にそ

の配置グループの十分なインスタンスがありません)、またはしばらくの間「clean」フラグが付いていません。影響を受けるPGについての詳しい情報は、次のコマンドを使用して参照できます。

```
cephuser@adm > ceph health detail
```

ほとんどの場合、根本原因は、現在1つ以上のOSDがダウンしていることです。次のコマンドを使用して、問題がある特定のPGの状態を問い合わせることができます。

```
cephuser@adm > ceph tell pgid query
```

## PG\_DEGRADED\_FULL

クラスタに空き領域がないため、一部のデータのデータ冗長性が低下しているか、危険な状態である可能性があります。具体的には、1つ以上のPGに「backfill\_toofull」または「recovery\_toofull」のフラグが付いています。つまり、1つ以上のOSDが**backfillfull**のしきい値を超えているため、クラスタはデータを移行または回復できません。

## PG\_DAMAGED

データスクラブ(17.6項「[配置グループのスクラブ](#)」を参照)によってクラスタのデータ整合性に問題が検出されました。具体的には、1つ以上のPGに「inconsistent」または「snaptrim\_error」のフラグが付いています。これは、前のスクラブ操作で問題が見つかったか、「repair」フラグが設定されていることを示しており、現在その不整合の修復が進行中であることを意味します。

## OSD\_SCRUB\_ERRORS

最近のOSDスクラブで不整合が発見されました。

## CACHE\_POOL\_NEAR\_FULL

キャッシュ層プールがほぼ満杯です。このコンテキストにおける「満杯」は、キャッシュプールの「target\_max\_bytes」および「target\_max\_objects」のプロパティによって判断されます。プールがターゲットしきい値に達した場合、データがキャッシュからフラッシュまたは削除される間、プールへの書き込み要求がブロックされることがあり、通常はレイテンシが非常に高くなり、パフォーマンスが低下する状態になります。次のコマンドを使用して、キャッシュプールのターゲットサイズを調整できます。

```
cephuser@adm > ceph osd pool set cache-pool-name target_max_bytes bytes
cephuser@adm > ceph osd pool set cache-pool-name target_max_objects objects
```

通常のキャッシュフラッシュおよび削除アクティビティは、基本層の可用性またはパフォーマンスの低下、あるいはクラスタ全体の負荷によっても低速になることがあります。

## TOO\_FEW\_PGS

使用中のPGの数が、設定可能なしきい値である、OSDあたりの `mon_pg_warn_min_per_osd` のPG数未満です。このため、クラスタ内のOSDへのデータの分散とバランスが最適ではなくなり、全体的なパフォーマンスが低下します。

## TOO\_MANY\_PGS

使用中のPGの数が、設定可能なしきい値である、OSDあたりの `mon_pg_warn_max_per_osd` のPG数を超過しています。このため、OSDデーモンのメモリ使用量が増加する、クラスタの状態が変化(たとえば、OSDの再起動、追加、削除)した後にはピアリングの速度が低下する、Ceph ManagerとCeph Monitorの負荷が増加するなどの可能性があります。

既存のプールの `pg_num` 値を減らすことはできませんが、`pgp_num` 値を減らすことは可能です。これによって、同じOSDセットの複数のPGを効果的に一緒に配置して、前に説明した悪影響を多少緩和できます。次のコマンドを使用して、`pgp_num` の値を調整できます。

```
cephuser@adm > ceph osd pool set pool pgp_num value
```

## SMALLER\_PGP\_NUM

1つ以上のプールの `pgp_num` の値が `pg_num` 未満です。これは通常、配置動作を同時に増やさずにPG数を増やしたことを示します。通常は、次のコマンドを使用して、`pgp_num` を `pg_num` に一致するよう設定し、データマイグレーションをトリガすることによって解決します。

```
cephuser@adm > ceph osd pool set pool pgp_num pg_num_value
```

## MANY\_OBJECTS\_PER\_PG

1つ以上のプールで、PGあたりの平均オブジェクト数がクラスタの全体の平均を大幅に超過しています。具体的なしきい値は、`mon_pg_warn_max_object_skew` の設定値で制御します。これは通常、クラスタ内のほとんどのデータを含むプールのPGが少なすぎるか、それほど多くのデータを含まない他のプールのPGが多すぎるか、またはその両方であることを示します。Monitorの `mon_pg_warn_max_object_skew` 設定オプションを調整することによって、しきい値を上げてヘルス警告を停止できます。

## POOL\_APP\_NOT\_ENABLED

1つ以上のオブジェクトが含まれるプールが存在しますが、特定のアプリケーション用のタグが付けられていません。この警告を解決するには、プールにアプリケーション用のラベルを付けます。たとえば、プールがRBDによって使用される場合は、次のコマンドを実行します。

```
cephuser@adm > rbd pool init pool_name
```

プールがカスタムアプリケーション「foo」によって使用されている場合、次の低レベルのコマンドを使用してラベルを付けることもできます。

```
cephuser@adm > ceph osd pool application enable foo
```

## POOL\_FULL

1つ以上のプールがクォータに達しています(またはクォータに非常に近付いています)。このエラー条件をトリガするためのしきい値は、`mon_pool_quota_crit_threshold`設定オプションで制御します。次のコマンドを使用して、プールクォータを増減(または削除)できます。

```
cephuser@adm > ceph osd pool set-quota pool max_bytes bytes
cephuser@adm > ceph osd pool set-quota pool max_objects objects
```

クォータの値を0に設定すると、クォータは無効になります。

## POOL\_NEAR\_FULL

1つ以上のプールがクォータに近付いています。この警告条件をトリガするためのしきい値は、`mon_pool_quota_warn_threshold`設定オプションで制御します。次のコマンドを使用して、プールクォータを増減(または削除)できます。

```
cephuser@adm > ceph osd osd pool set-quota pool max_bytes bytes
cephuser@adm > ceph osd osd pool set-quota pool max_objects objects
```

クォータの値を0に設定すると、クォータは無効になります。

## OBJECT\_MISPLACED

クラスタ内の1つ以上のオブジェクトが、クラスタで保存場所に指定されているノードに保存されていません。これは、クラスタに最近加えられた変更によって発生したデータマイグレーションがまだ完了していないことを示します。データの誤配置そのものは危険な状態ではありません。データ整合性は危険な状態ではなく、必要な数の新しいコピーが(必要な場所に)存在する限り、オブジェクトの古いコピーが削除されることはありません。

## OBJECT\_UNFOUND

クラスタ内の1つ以上のオブジェクトが見つかりません。具体的には、OSDはオブジェクトの新しいコピーまたは更新されたコピーが存在していることを認識していますが、現在動作しているOSD上にオブジェクトのそのバージョンのコピーが見つかりません。「見つからない」オブジェクトに対する読み込みまたは書き込み要求はブロックされます。検出されなかったオブジェクトの最新のコピーがある、ダウンしているOSDを稼働状態に戻すのが理想的です。見つからないオブジェクトを受け持っているPGのピアリング状態から、候補のOSDを特定できます。

```
cephuser@adm > ceph tell pgid query
```

## REQUEST\_SLOW

OSDの1つ以上の要求の処理に長い時間がかかっています。これは、極端な負荷、低速なストレージデバイス、またはソフトウェアのバグを示している可能性があります。OSDホストから次のコマンドを実行して、対象のOSDの要求キューを問い合わせることができます。

```
cephuser@adm > ceph daemon osd.id ops
```

最も低速な最近の要求のサマリが表示されます。

```
cephuser@adm > ceph daemon osd.id dump_historic_ops
```

次のコマンドを使用して、OSDの場所を特定できます。

```
cephuser@adm > ceph osd find osd.id
```

## REQUEST\_STUCK

1つ以上のOSD要求が比較的長時間ブロックされています(たとえば、4096秒)。これは、クラスタが長時間にわたって正常でないか(たとえば、十分な数のOSDが実行されていないか、PGが非アクティブ)、OSDに何らかの内部的な問題があることを示します。

## PG\_NOT\_SCRUBBED

最近、1つ以上のPGがスクラブ(17.6項「[配置グループのスクラブ](#)」を参照)されていません。PGは通常、`mon_scrub_interval`の秒数ごとにスクラブされ、スクラブなしに`mon_warn_not_scrubbed`の間隔が経過した場合、この警告がトリガされます。cleanフラグが付いていない場合、PGはスクラブされません。これは、PGが誤配置されているか機能が低下している場合に発生することがあります(前のPG\_AVAILABILITYおよびPG\_DEGRADEDを参照してください)。次のコマンドを使用して、クリーンなPGのスクラブを手動で開始できます。

```
cephuser@adm > ceph pg scrub pgid
```

## PG\_NOT\_DEEP\_SCRUBBED

最近、1つ以上のPGが詳細スクラブ(17.6項「[配置グループのスクラブ](#)」を参照)されていません。PGは通常、`osd_deep_mon_scrub_interval`の秒数ごとにスクラブされ、スクラブなしに`mon_warn_not_deep_scrubbed`秒が経過した場合、この警告がトリガされます。cleanフラグが付いていない場合、PGは詳細スクラブされません。これは、PGが誤配置されているか機能が低下している場合に発生することがあります(前のPG\_AVAILABILITYおよびPG\_DEGRADEDを参照してください)。次のコマンドを使用して、クリーンなPGのスクラブを手動で開始できます。

```
cephuser@adm > ceph pg deep-scrub pgid
```



## ヒント

設定またはキーリングにデフォルト以外の場所を指定した場合、その場所を指定できません。

```
root # ceph -c /path/to/conf -k /path/to/keyring health
```

## 12.3 クラスタの使用量統計の確認

クラスタのデータ使用量とプール間での分散を確認するには、**ceph df** コマンドを使用します。詳細を取得するには、**ceph df detail** を使用します。

```
cephuser@adm > ceph df
--- RAW STORAGE ---
CLASS  SIZE      AVAIL    USED      RAW USED  %RAW USED
hdd    30 GiB    27 GiB   121 MiB   3.1 GiB   10.40
TOTAL  30 GiB    27 GiB   121 MiB   3.1 GiB   10.40

--- POOLS ---
POOL                                ID  STORED  OBJECTS  USED      %USED  MAX AVAIL
device_health_metrics              1      0 B         0      0 B        0    8.5 GiB
cephfs.my_cephfs.meta              2    1.0 MiB      22    4.5 MiB    0.02    8.5 GiB
cephfs.my_cephfs.data              3      0 B         0      0 B        0    8.5 GiB
.rgw.root                          4    1.9 KiB      13    2.2 MiB    0      8.5 GiB
myzone.rgw.log                     5    3.4 KiB     207     6 MiB    0.02    8.5 GiB
myzone.rgw.control                 6      0 B         8      0 B        0    8.5 GiB
myzone.rgw.meta                    7      0 B         0      0 B        0    8.5 GiB
```

出力の **RAW STORAGE** セクションには、クラスタがデータに使用しているストレージの量の概要が表示されます。

- **CLASS**: デバイスのストレージクラス。デバイスクラスの詳細については、17.1.1項「**デバイスクラス**」を参照してください。
- **SIZE**: クラスタの全体的なストレージ容量。
- **AVAIL**: クラスタで利用可能な空き領域の量。
- **USED**: ブロックデバイスに保持されている、純粋にデータオブジェクト用として割り当てられている領域(すべてのOSDの累積)。

- RAW USED: 「USED」領域と、Ceph用としてブロックデバイスで割り当て/予約されている領域(BlueStoreのBlueFSの部分など)。
- % RAW USED: 使用済みの未加工ストレージの割合。この数字をfull\_ratioおよびnear\_full\_ratioと組み合わせて使用して、クラスタの容量に達していないことを確認します。詳細については、[12.8項「ストレージの容量」](#)を参照してください。



## 注記: クラスタの充足レベル

未加工ストレージの充足レベルが100%に近付いている場合は、クラスタに新しいストレージを追加する必要があります。使用量がさらに多くなると、1つのOSDが満杯になり、クラスタのヘルスに問題が発生することがあります。

すべてのOSDの充足レベルを一覧にするには、コマンド `ceph osd df tree` を使用します。

出力の POOL セクションには、プールのリストと各プールの名目上の使用量が表示されます。このセクションの出力には、レプリカ、クローン、またはスナップショットは反映されて「いません」。たとえば、1MBのデータを持つオブジェクトを保存した場合、名目上の使用量は1MBですが、実際の使用量は、レプリカ、クローン、およびスナップショットの数によっては2MB以上になることがあります。

- POOL: プールの名前。
- ID: プールのID。
- STORED: ユーザが保存したデータの量。
- OBJECTS: プールごとの保存オブジェクトの名目上の数。
- USED: すべてのOSDノードによって純粋にデータ用として割り当てられている領域の量 (KB単位)。
- %USED: プールごとの使用済みストレージの名目上の割合。
- MAX AVAIL: 特定のプールで利用可能な最大領域。



## 注記

POOLS セクションの数字は名目上の数字です。レプリカ、スナップショット、またはクローンの数は含まれません。そのため、USEDの量や%USEDの量を合計しても、出力の RAW STORAGE セクションの RAW USED の量や%RAW USED の量にはなりません。

## 12.4 OSDの状態の確認

次のコマンドを実行して、OSDが動作中であることを確認します。

```
cephuser@adm > ceph osd stat
```

あるいは、

```
cephuser@adm > ceph osd dump
```

CRUSHマップ内での位置に従ってOSDを表示することもできます。

**ceph osd tree**コマンドを実行すると、CRUSHツリーと共に、ホスト、そのOSD、動作中かどうか、および重みが出力されます。

```
cephuser@adm > ceph osd tree
```

ID	CLASS	WEIGHT	TYPE NAME	STATUS	REWEIGHT	PRI-AFF
-1	3	0.02939	root default			
-3	3	0.00980	rack mainrack			
-2	3	0.00980	host osd-host			
0	1	0.00980	osd.0	up	1.00000	1.00000
1	1	0.00980	osd.1	up	1.00000	1.00000
2	1	0.00980	osd.2	up	1.00000	1.00000

## 12.5 満杯のOSDの確認

Cephでは、データが失われないようにするため、満杯のOSDに書き込むことはできません。運用クラスタでは、クラスタが満杯率に近付くと警告が表示されます。**mon osd full ratio**のデフォルトは0.95です。すなわち、容量の95%に達すると、クライアントによるデータの書き込みが停止されます。**mon osd nearfull ratio**のデフォルトは0.85です。すなわち、容量の85%に達するとヘルス警告が生成されます。

満杯のOSDノードは**ceph health**でレポートされます。

```
cephuser@adm > ceph health
HEALTH_WARN 1 nearfull osds
osd.2 is near full at 85%
```

あるいは、

```
cephuser@adm > ceph health
HEALTH_ERR 1 nearfull osds, 1 full osds
osd.2 is near full at 85%
osd.3 is full at 97%
```

満杯のクラスタへの最適な対応方法は、新しいOSDホスト/ディスクを追加して、クラスタが新しく利用可能になったストレージにデータを再分散できるようにする方法です。



## ヒント: OSDが満杯になることを防ぐ

OSDが満杯になると(ディスク領域の100%を使用すると)、通常は警告なしにすぐにクラッシュします。次に、OSDノードを管理する際に覚えておくべきヒントをいくつか示します。

- 各OSDのディスク領域(通常は/var/lib/ceph/osd/osd-{1,2..}にマウント)は、基礎となる専用のディスクまたはパーティションに配置する必要があります。
- Ceph設定ファイルを確認して、CephがOSD専用のディスク/パーティションにログファイルを保存しないようにします。
- 他のプロセスがOSD専用のディスク/パーティションに書き込まないようにします。

## 12.6 Monitorの状態の確認

クラスタの起動後、最初にデータを読み書きする前に、Ceph Monitorの定数のステータスを確認します。クラスタがすでに要求を処理している場合は、Ceph Monitorのステータスを定期的に確認して、それらが実行されていることを確認します。

Monitorマップを表示するには、次のコマンドを実行します。

```
cephuser@adm > ceph mon stat
```

あるいは、

```
cephuser@adm > ceph mon dump
```

Monitorクラスタの定数の状態を確認するには、次のコマンドを実行します。

```
cephuser@adm > ceph quorum_status
```

定数の状態が返されます。たとえば、3つのMonitorで構成されるCephクラスタは、次の状態を返す場合があります。

```
{ "election_epoch": 10,
  "quorum": [
    0,
    1,
    2],
  "monmap": { "epoch": 1,
    "fsid": "444b489c-4f16-4b75-83f0-cb8097468898",
    "modified": "2011-12-12 13:28:27.505520",
    "created": "2011-12-12 13:28:27.505520",
```

```

    "mons": [
      { "rank": 0,
        "name": "a",
        "addr": "192.168.1.10:6789\0"},
      { "rank": 1,
        "name": "b",
        "addr": "192.168.1.11:6789\0"},
      { "rank": 2,
        "name": "c",
        "addr": "192.168.1.12:6789\0"}
    ]
  }
}

```

## 12.7 配置グループの状態の確認

配置グループはオブジェクトをOSDにマップします。配置グループを監視する場合、配置グループが`active`および`clean`である必要があります。詳細については、[12.9項「OSDと配置グループの監視」](#)を参照してください。

## 12.8 ストレージの容量

Cephストレージクラスタがその最大容量に近付くと、Cephは、データ損失を防止するための安全対策として、Ceph OSDに対して書き込みや読み込みを行えないようにします。したがって、運用クラスタをその満杯率に近付けることは、高可用性が犠牲になるため、適切な方法ではありません。デフォルトの満杯率は0.95、つまり容量の95%に設定されています。これは、OSDの数が少ないテストクラスタでは非常に積極的な設定です。



### ヒント: ストレージ容量の増加

クラスタを監視する際には、`nearfull`の比率に関連する警告に注意してください。これは、1つ以上のOSDに障害が発生している場合に、複数のOSDに障害が発生すると、サービスが一時的に中断する可能性があることを意味します。OSDを追加してストレージ容量を増やすことを検討してください。

テストクラスタの一般的なシナリオには、システム管理者がCephストレージクラスタからCeph OSDを1つ削除して、クラスタの再バランスを監視することが含まれます。次に、別のCeph OSDを削除し、最終的にクラスタが満杯率に達してロックするまで削除を続けます。テストクラスタでも、何らかの容量計画を立てることをお勧めします。計画を立てることにより、高可用性を維持するために必要なスペア容量を見積もることができます。理

想的には、Ceph OSDに一連の障害が発生した場合に、これらのCeph OSDをすぐに交換しなくてもクラスタがactive + cleanの状態に回復できるような計画を立てます。active + degradedの状態でクラスタを実行することはできますが、これは通常の動作条件には適しません。

次の図は、ホストあたり1つのCeph OSDが存在する33個のCephノードで構成されるシンプルなCephストレージクラスタを表していて、各ノードは3TBドライブに対して読み書きを行います。この例で使用するクラスタの実際の最大容量は99TBです。mon osd full ratioオプションは0.95に設定されています。クラスタの残り容量が5TBまで低下すると、クライアントはデータを読み書きできなくなります。したがって、このストレージクラスタの動作容量は、99TBではなく95TBです。

ラック1	ラック2	ラック3	ラック4	ラック5	ラック6
OSD 1	OSD 7	OSD 13	OSD 19	OSD 25	OSD 31
OSD 2	OSD 8	OSD 14	OSD 20	OSD 26	OSD 32
OSD 3	OSD 9	OSD 15	OSD 21	OSD 27	OSD 33
OSD 4	OSD 10	OSD 16	OSD 22	OSD 28	スベア
OSD 5	OSD 11	OSD 17	OSD 23	OSD 29	スベア
OSD 6	OSD 12	OSD 18	OSD 24	OSD 30	スベア

図 12.1: CEPHクラスタ

このようなクラスタで、1つまたは2つのOSDに障害が発生することはよくあります。それほど頻繁ではないものの妥当なシナリオとして、ラックのルータまたは電源装置に障害が発生して、同時に複数のOSD (OSD 7~12など)がダウンする状況があります。このようなシナリオでは、たとえ数台のホストと追加のOSDを早急に追加することになるとしても、稼働状態を維持してactive + clean状態を達成できるクラスタを目指す必要があります。容量使用率が高すぎても、データが失われることはありません。ただし、クラスタの容量使用率が満杯率を超える場合は、障害ドメイン内の停止を解決しても、データ可用性が犠牲になる可能性が依然としてあります。この理由のため、少なくとも大まかな容量計画を立てることをお勧めします。

クラスタの次の2つ数量を特定します。

1. OSDの数。
2. クラスタの合計容量。

クラスタの合計容量をクラスタのOSDの数で割ると、クラスタ内のOSDの平均容量がわかります。この平均容量に、通常の操作時に同時に障害が発生すると予想するOSDの数(比較的少数)を掛けることを検討してください。最後に、クラスタの容量に満杯率を掛ると、最大動作

容量が得られます。次に、妥当な満杯率に達しないと予想されるOSDのデータの量を引きます。OSD障害(OSDのラック)の数を増やして上の手順を繰り返し、ほぼ満杯率に近い妥当な数を得ます。

次の設定はクラスタの作成時にのみ適用され、その後OSDマップに保存されます。

```
[global]
mon osd full ratio = .80
mon osd backfillfull ratio = .75
mon osd nearfull ratio = .70
```



## ヒント

これらの設定はクラスタの作成中にのみ適用されます。後で、**`ceph osd set-nearfull-ratio`**コマンドと**`ceph osd set-full-ratio`**コマンドを使用して、OSDマップで変更する必要があります。

### **mon osd full ratio**

OSDをfullと見なす使用済みディスク容量の割合。デフォルトは0.95です。

### **mon osd backfillfull ratio**

OSDが過剰にfullであるためバックフィルできないと見なす使用済みディスク容量の割合。デフォルトは0.90です。

### **mon osd nearfull ratio**

OSDをnearfullと見なす使用済みディスク容量の割合。デフォルトは0.85です。



## ヒント: OSDの重みの確認

一部のOSDはnearfullであるのに他のOSDには十分な容量がある場合、nearfullのOSDのCRUSHの重みに問題がある可能性があります。

## 12.9 OSDと配置グループの監視

高可用性と高信頼性を実現するには、ハードウェアとソフトウェアの問題を管理する耐障害性を持つアプローチが必要です。Cephには単一障害点がなく、データに対する要求を「degraded」モードで処理できます。Cephのデータ配置には、データが特定のOSDアドレスに直接バインドされないようにする間接層が導入されています。つまり、システム障害を追跡するには、問題の根本にある配置グループとその基礎となるOSDを見つける必要があります。



## ヒント: 障害が発生した場合のアクセス

クラスタの一部に障害が発生すると、特定のオブジェクトにアクセスできなくなる場合があります。これは、他のオブジェクトにアクセスできないという意味ではありません。障害が発生したら、OSDおよび配置グループを監視するための手順に従います。次にトラブルシューティングを開始します。

Cephは通常、自己修復します。ただし、問題が続く場合は、OSDと配置グループを監視すると問題を特定するのに役立ちます。

### 12.9.1 OSDの監視

OSDのステータスは、「クラスタ内」(「in」)または「クラスタ外」(「out」)のいずれかです。それと同時に、「稼働中」(「up」)または「ダウンしていて実行中でない」(「down」)のいずれかにもなります。OSDが「up」の場合、クラスタ内(データを読み書きできる)またはクラスタ外のいずれかの可能性があります。OSDがクラスタ内に存在していて、最近クラスタ外に移動した場合、Cephは配置グループを他のOSDに移行します。OSDがクラスタ外の場合、CRUSHは配置グループをOSDに割り当てません。OSDは、「down」の場合、「out」にもなります。



## 注記: 正常でない状態

OSDが「down」で「in」の場合は問題があり、クラスタは正常な状態ではありません。

`ceph health`、`ceph -s`、`ceph -w`などのコマンドを実行する場合、クラスタは常にHEALTH OKをエコーバックするわけではないことに気付くことがあります。OSDに関しては、次の状況ではクラスタがHEALTH OKをエコーしないことを予期する必要があります。

- まだクラスタを起動していない(クラスタは応答しない)。
- クラスタを起動または再起動したが、配置グループは作成中で、OSDはピアリングプロセス中であるため、まだ準備ができていない。
- OSDを追加または削除した。
- クラスタマップを変更した。

OSDの監視の重要な側面は、クラスタが稼働中であるときに、クラスタ内のすべてのOSDも稼働中であることを確認することです。すべてのOSDが実行中であるかどうかを確認するには、次のコマンドを実行します。

```
root # ceph osd stat
x osds: y up, z in; epoch: eNNNN
```

この結果から、OSDの合計数(x)、「up」のOSDの数(y)、「in」のOSDの数(z)、およびマップのエポック(eNNNN)がわかります。クラスタ内にある「in」のOSDの数が「up」であるOSDの数より多い場合は、次のコマンドを実行して、実行中でないceph-osdデーモンを特定します。

```
root # ceph osd tree
#ID CLASS WEIGHT  TYPE NAME                STATUS REWEIGHT PRI-AFF
-1          2.00000 pool openstack
-3          2.00000 rack dell-2950-rack-A
-2          2.00000 host dell-2950-A1
0  ssd 1.00000    osd.0                    up  1.00000 1.00000
1  ssd 1.00000    osd.1                    down 1.00000 1.00000
```

たとえば、IDが1のOSDがdownしている場合は、次のコマンドを実行して起動します。

```
cephuser@osd > sudo systemctl start ceph-CLUSTER_ID@osd.0.service
```

停止しているOSDや再起動しないOSDに関連する問題については、『Troubleshooting Guide』、第4章「Troubleshooting OSDs」、4.3項「OSDs not running」を参照してください。

## 12.9.2 配置グループセットの割り当て

CRUSHが配置グループをOSDに割り当てる場合、CRUSHはプールのレプリカの数を確認し、配置グループの各レプリカが異なるOSDに割り当てられるように配置グループをOSDに割り当てます。たとえば、プールに配置グループのレプリカが3つ必要な場合、CRUSHはこれらをそれぞれosd.1、osd.2、osd.3に割り当てることがあります。CRUSHは実際には、CRUSHマップで設定した障害ドメインを考慮した擬似的にランダムな配置にしようとします。そのため、大規模なクラスタで複数の配置グループが最も近隣にあるOSDに割り当てられることはほとんどありません。特定の配置グループのレプリカを含む必要があるOSDのセットを「動作セット」と呼びます。動作セットのOSDがダウンしているか、または他の理由で配置グループのオブジェクトの要求を処理できない場合があります。このような状況が生じた場合は、次のシナリオの1つに一致する可能性があります。

- OSDを追加または削除した。これにより、CRUSHが配置グループを他のOSDに再割り当てしたため、「動作セット」の構成が変更され、「バックフィル」プロセスによってデータのマイグレーションが実行された。
- OSDが「down」状態であったため再起動され、現在回復中である。
- 「動作セット」のOSDが「down」状態であるか、要求を処理できないため、別のOSDが一時的にその権限を引き継いだ。

Cephは、要求を実際に処理するOSDのセットである「アップセット」を使用してクライアント要求を処理します。ほとんどの場合、「アップセット」と「動作セット」は事実上同一です。これらが同一ではない場合、Cephがデータを移行中であるか、OSDが回復中であるか、または問題があることを示している場合があります(たとえば、このようなシナリオでは、Cephでは通常HEALTH\_WARN状態と「stuck stale」メッセージをエコーします)。

配置グループのリストを取得するには、次のコマンドを実行します。

```
cephuser@adm > ceph pg dump
```

指定した配置グループの「動作セット」または「アップセット」内にあるOSDを表示するには、次のコマンドを実行します。

```
cephuser@adm > ceph pg map PG_NUM
osdmap eNNN pg RAW_PG_NUM (PG_NUM) -> up [0,1,2] acting [0,1,2]
```

この結果から、osdmapエポック(eNNN)、配置グループの数(PG\_NUM)、「アップセット」(「up」)のOSD、および「動作セット」(「acting」)のOSDがわかります。



### ヒント: クラスタの問題の指標

「アップセット」と「動作セット」が一致しない場合、これはクラスタの再バランスそのものか、クラスタの潜在的な問題の指標である可能性があります。

## 12.9.3 ピアリング

配置グループにデータを書き込む場合、データの状態はactiveでなければならない、さらにclean状態である必要があります。Cephが配置グループの現在の状態を判断するために、配置グループのプライマリOSD(「動作セット」の最初のOSD)は、セカンダリおよび3番目のOSDとピアリングして、配置グループの現在の状態に関する合意を確立します(PGの3つのレプリカがプールにあることを想定)。

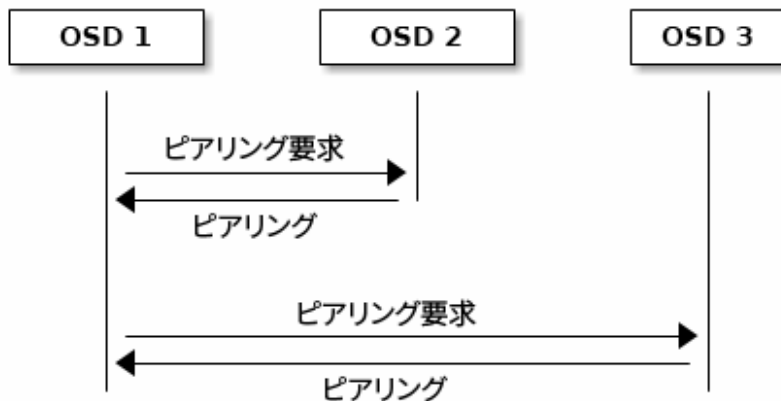


図 12.2: ピアリングスキーマ

## 12.9.4 配置グループの状態の監視

`ceph health`、`ceph -s`、`ceph -w`などのコマンドを実行する場合、クラスタは常にHEALTH OKメッセージをエコーバックするわけでないことに気付くことがあります。OSDが実行中であるかどうかを確認した後で、配置グループの状態も確認する必要があります。

配置グループのピアリングに関係する次のような多くの状況では、クラスタはHEALTH OKをエコー「しない」ことを予期してください。

- プールを作成したが、配置グループはまだピアリングされていない。
- 配置グループが回復中である。
- クラスタにOSDを追加したか、クラスタからOSDを削除した。
- CRUSHマップを変更したが、配置グループは移行中である。
- 配置グループの異なるレプリカに整合性のないデータがある。
- Cephが配置グループのレプリカをスクラブしている。
- Cephにバックフィル操作を完了するための十分なストレージ容量がない。

上のいずれかの状況でCephがHEALTH WARNをエコーしても慌てないでください。多くの場合、クラスタは単独で回復します。場合によっては、対処が必要です。配置グループの監視の重要な側面は、クラスタが稼働しているときに、すべての配置グループが「active」であり、できれば「clean」状態であることを確認することです。すべての配置グループのステータスを確認するには、次のコマンドを実行します。

```
cephuser@adm > ceph pg stat
x pgs: y active+clean; z bytes data, aa MB used, bb GB / cc GB avail
```

この結果から、配置グループの合計数(x)、特定の状態(「active+clean」など)である配置グループの数(y)、および保存データの量(z)がわかります。

配置グループの状態に加えて、Cephでは、使用済みのストレージ容量(aa)、残りのストレージ容量(bb)、および配置グループの合計ストレージ容量もエコーバックされます。次のようないくつかのケースでは、これらの数値が重要になる可能性があります。

- near full ratioまたはfull ratioに達しつつある。
- CRUSH設定のエラーのため、データがクラスタ間で分散されている。



## ヒント: 配置グループID

配置グループIDは、プール番号(プール名ではない)、それに続くピリオド(.)、および配置グループID (16進数)で構成されます。プール番号とプールの名前は、**ceph osd lspools**の出力で参照できます。たとえば、デフォルトのプール**rbld**はプール番号0に対応します。完全修飾形式の配置グループIDは次の形式です。

```
POOL_NUM.PG_ID
```

これは通常、次のようになります。

```
0.1f
```

配置グループのリストを取得するには、次のコマンドを実行します。

```
cephuser@adm > ceph pg dump
```

出力をJSON形式でフォーマットしてファイルに保存することもできます。

```
cephuser@adm > ceph pg dump -o FILE_NAME --format=json
```

特定の配置グループに対してクエリを実行するには、次のコマンドを実行します。

```
cephuser@adm > ceph pg POOL_NUM.PG_ID query
```

次のリストでは、配置グループの一般的な状態について詳しく説明します。

## 作成中

プールを作成すると、指定した数の配置グループが作成されます。Cephは、1つ以上の配置グループを作成している場合に「creating」をエコーします。配置グループが作成されると、その配置グループの「動作セット」に属するOSDがピアリングされます。ピ

アリングが完了したら、配置グループのステータスは「active+clean」になります。これは、Cephクライアントがその配置グループへの書き込みを開始できることを意味します。

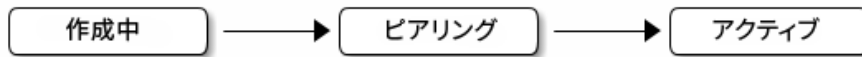


図 12.3: 配置グループのステータス

## ピアリング

Cephは、配置グループのピアリング中に、配置グループのレプリカを保存するOSDを、その配置グループ内にあるオブジェクトとメタデータの状態について合意状態にします。つまり、Cephがピアリングを完了すると、その配置グループを保存するOSDは配置グループの現在の状態に関して合意したことになります。ただし、ピアリングプロセスが完了しても、各レプリカの内容が最新であるという意味には「なりません」。



### 注記: 信頼できる履歴

Cephでは、「動作セット」のすべてのOSDが書き込み操作を永続化するまで、クライアントへの書き込み操作を確認「しません」。この動作により、ピアリング操作が最後に正常に完了した後に確認されたすべての書き込み操作のレコードが、「動作セット」の少なくとも1つのメンバーに確実に存在するようになります。

確認された書き込み操作それぞれの正確なレコードにより、Cephは、配置グループの信頼できる新しい履歴を構築および拡張できます。この履歴には、一連の操作の順序が完全かつ全面的に記録されているため、この履歴を実行すれば、配置グループのOSDのコピーを最新の状態にすることができます。

## アクティブ

Cephがピアリングプロセスを完了すると、配置グループはactiveになります。active状態とは、プライマリ配置グループとレプリカで配置グループのデータを読み込み操作と書き込み操作に一般的に使用できることを意味します。

## クリーン

配置グループがclean状態である場合、プライマリOSDとレプリカOSDは正常にピアリングされていて、その配置グループに未処理のレプリカは存在しません。配置グループ内のすべてのオブジェクトは、Cephによって正確な回数複製されています。

## DEGRADED

クライアントがプライマリOSDにオブジェクトを書き込む場合、プライマリOSDが、レプリカをレプリカOSDに書き込む責任を持ちます。プライマリOSDがオブジェクトをストレージに書き込んだ後も、配置グループは「degraded」状態のままです。この状態は、Cephがレプリカオブジェクトを正常に作成したという確認をプライマリOSDがレプリカOSDから受け取るまで続きます。

配置グループが「active+degraded」になる可能性がある理由は、OSDにまだオブジェクトがすべて格納されていなくてもOSDが「active」になる可能性があるためです。OSDがダウンした場合、CephはそのOSDに割り当てられた各配置グループを「degraded」としてマークします。OSDが稼働状態に戻ったら、OSDをもう一度ピアリングする必要があります。ただし、「degraded」状態の配置グループが「active」であれば、クライアントは引き続きその配置グループに新しいオブジェクトを書き込むことができます。

OSDが「down」で、「degraded」状態が解決しない場合、Cephは、ダウンしているOSをクラスタの「out」としてマークし、「down」状態のOSDから別のOSDにデータを再マップします。「down」とマークされてから「out」とマークされるまでの時間は、`mon osd down out interval`オプションで制御します。デフォルトでは600秒に設定されています。

配置グループに含める必要がある1つ以上のオブジェクトをCephが見つけないという理由で、配置グループが「degraded」状態になる可能性もあります。見つからないオブジェクトに対して読み書きを行うことはできませんが、「degraded」状態の配置グループ内にある他のすべてのオブジェクトには今までどおりアクセスできます。

## 回復中

Cephは、ハードウェアやソフトウェアの問題が継続している場合に大規模な耐障害性を実現するように設計されています。OSDが「down」になった場合、その内容が、配置グループ内にある他のレプリカの現在の状態よりも遅れることがあります。OSDが「up」に戻ったら、現在の状態を反映するように配置グループの内容を更新する必要があります。その間、OSDに「recovering」状態が反映される場合があります。

ハードウェアの障害によって複数のOSDのカスケード障害が発生する可能性があるため、回復は常に簡単であるとは限りません。たとえば、ラックまたはキャビネットのネットワークスイッチに障害が発生することがあります。これにより、多数のホストマシンのOSDがクラスタの現在の状態より遅れる可能性があります。障害が解決されたら、各OSDを回復する必要があります。

Cephには、新しいサービス要求と、データオブジェクトを回復して配置グループを現在の状態に復元する必要性との間のリソース競合のバランスを取るための設定が多数用意されています。`osd recovery delay start`設定を使用すると、回復プロセスを開始する前に、OSDを再起動して再ピアリングを行い、さらに一部の再生要求を処理するこ

ともできます。`osd recovery thread timeout`は、スレッドのタイムアウトを設定します。これは、複数のOSDが失敗した場合に、時間差で再起動および再ピアリングするためです。`osd recovery max active`設定は、OSDが同時に処理する回復要求の数を制限して、OSDが要求を処理できなくなるのを防ぎます。`osd recovery max chunk`設定は、回復されたデータチャンクのサイズを制限し、ネットワークの輻輳を防ぎます。

## バックフィル

新しいOSDがクラスタに参加すると、CRUSHは、クラスタ内のOSDから、新しく追加されたOSDに配置グループを再割り当てします。再割り当てされた配置グループをただちに受け入れるよう新しいOSDに強制すると、新しいOSDに過度な負荷がかかる可能性があります。OSDに配置グループをバックフィルすることで、このプロセスをバックグラウンドで開始できます。バックフィルが完了して準備が整ったら、新しいOSDは処理を開始します。

バックフィル操作中に、次のいずれかの状態が表示されることがあります。

「backfill\_wait」は、バックフィル操作が保留中で、まだ進行中ではないことを示します。「backfill」は、バックフィル操作が進行中であることを示します。

「backfill\_too\_full」は、バックフィル操作が要求されたものの、ストレージ容量が十分でないため完了できなかったことを示します。配置グループをバックフィルできない場合、「incomplete」と見なされることがあります。

Cephには、配置グループをOSD (特に新しいOSD)に再割り当てすることに伴う負荷を管理するための設定が多数用意されています。デフォルトでは、`osd max backfills`は、OSDに対する同時バックフィルの最大数を10に設定します。`backfill full ratio`を使用すると、OSDがその満杯率(デフォルトでは90%)に近付いている場合にバックフィル要求を拒否できます。変更するには、`ceph osd set-backfillfull-ratio`コマンドを使用します。OSDがバックフィル要求を拒否する場合、`osd backfill retry interval`を使用すると、OSDは要求を再試行できます(デフォルトでは10秒後)。OSDに`osd backfill scan min`および`osd backfill scan max`を設定して、スキャン間隔を管理することもできます(デフォルトでは64と512)。

## REMAPPED (再マップ)

配置グループの変更を処理する「動作セット」が変更されると、古い「動作セット」から新しい「動作セット」にデータが移行されます。新しいプライマリOSDが要求を処理するまでにしばらく時間がかかる場合があります。そのため、配置グループのマイグレーションが完了するまで、古いプライマリに対して引き続き要求を処理するよう求めることができます。データマイグレーションが完了したら、新しい「動作セット」のプライマリOSDがマッピングで使用されます。

## STALE

Cephは、ハートビートを使用して、ホストとデーモンが確実に実行されるようにしていますが、`ceph-osd`デーモンが「stuck」状態になり、統計情報がタイムリーにレポートされないこともあります(たとえば、一時的なネットワーク障害)。デフォルトでは、OSDデーモンはその配置グループ、およびブートと障害の統計情報を0.5秒ごとにレポートします。これは、ハートビートのしきい値よりも高い頻度です。配置グループの「動作セット」のプライマリOSDがモニターにレポートしない場合、または他のOSDが、プライマリOSDが「down」しているとレポートした場合、モニターは配置グループを「stale」としてマークします。

クラスタの起動時には、ピアリングプロセスが完了するまで「stale」状態が表示されることがよくあります。クラスタがしばらく動作した後で配置グループが「stale」状態になっている場合は、その配置グループのプライマリOSDがダウンしているか、配置グループの統計情報をモニターにレポートしていないことを示します。

## 12.9.5 オブジェクトの場所の検索

オブジェクトデータをCephオブジェクトストアに保存するには、Cephクライアントは、オブジェクトを名を設定して、関連するプールを指定する必要があります。Cephクライアントは最新のクラスタマップを取得し、CRUSHアルゴリズムは、オブジェクトを配置グループにマップする方法を計算し、続いて配置グループをOSDに動的に割り当てる方法を計算します。オブジェクトの場所を見つけるには、オブジェクト名とプール名があれば十分です。例:

```
cephuser@adm > ceph osd map POOL_NAME OBJECT_NAME [NAMESPACE]
```

### 例 12.1: オブジェクトの特定

一例として、オブジェクトを作成しましょう。コマンドラインで**`rados put`**コマンドを使用して、オブジェクト名「test-object-1」、いくつかのオブジェクトデータを含むサンプルファイル「testfile.txt」へのパス、およびプール名「data」を指定します。

```
cephuser@adm > rados put test-object-1 testfile.txt --pool=data
```

Cephオブジェクトストアにオブジェクトが保存されたことを確認するため、次のコマンドを実行します。

```
cephuser@adm > rados -p data ls
```

続いて、オブジェクトの場所を特定します。Cephによってオブジェクトの場所が出力されます。

```
cephuser@adm > ceph osd map data test-object-1
```

```
osdmap e537 pool 'data' (0) object 'test-object-1' -> pg 0.d1743484 \
(0.4) -> up ([1,0], p0) acting ([1,0], p0)
```

サンプルオブジェクトを削除するには、単に**rados rm**コマンドを使用して削除します。

```
cephuser@adm > rados rm test-object-1 --pool=data
```

## 13 運用タスク

### 13.1 クラスタ設定の変更

既存のCephクラスタの設定を変更するには、次の手順に従います。

1. 現在のクラスタの設定をファイルにエクスポートします。

```
cephuser@adm > ceph orch ls --export --format yaml > cluster.yaml
```

2. 設定が記載されたファイルを編集し、関連する行を更新します。仕様の例については、『導入ガイド』、第5章「cephadmによる展開」、5.4項「サービスとゲートウェイの展開」と13.4.3項「DriveGroups仕様を用いたOSDの追加」を参照してください。
3. 新しい設定を適用します。

```
cephuser@adm > ceph orch apply -i cluster.yaml
```

### 13.2 ノードの追加

Cephクラスタに新しいノードを追加するには、次の手順に従います。

1. SUSE Linux Enterprise ServerとSUSE Enterprise Storageを新規ホストにインストールします。詳細については、『導入ガイド』、第5章「cephadmによる展開」、5.1項「SUSE Linux Enterprise Serverのインストールと設定」を参照してください。
2. ホストを既存のSalt MasterのSalt Minionとして設定します。詳細については、『導入ガイド』、第5章「cephadmによる展開」、5.2項「Saltの展開」を参照してください。
3. 新しいホストをceph-saltに追加し、cephadmにホストを認識させます。たとえば、次のコマンドを実行します。

```
root@master # ceph-salt config /ceph_cluster/minions add ses-min5.example.com
root@master # ceph-salt config /ceph_cluster/roles/cephadm add ses-min5.example.com
```

詳細については、『導入ガイド』、第5章「cephadmによる展開」、5.3.2.2項「Salt Minionの追加」を参照してください。

4. ceph-saltにノードが追加されたことを確認します。

```
root@master # ceph-salt config /ceph_cluster/minions ls
```

```
o- minions ..... [Minions: 5]
[...]
o- ses-min5.example.com ..... [no roles]
```

5. 新しいクラスタホストに設定を適用します。

```
root@master # ceph-salt apply ses-min5.example.com
```

6. 新しく追加したホストがcephadm環境に属していることを確認します。

```
cephuser@adm > ceph orch host ls
HOST                ADDR                LABELS    STATUS
[...]
ses-min5.example.com  ses-min5.example.com
```

## 13.3 ノードの削除



### ヒント: OSDの削除

削除しようとしているノードがOSDを実行している場合、まずOSDを削除してからそのノード上でOSDが実行されていないことを確認してください。OSDを削除する方法の詳細については、[13.4.4項「OSDの削除」](#)を参照してください。

クラスタからノードを削除するには、次の手順に従います。

1. `node-exporter`と`crash`を除くすべてのCephサービスタイプについて、ノードのホスト名をクラスタの配置仕様ファイル(`cluster.yml`など)から削除します。詳細については、『導入ガイド』、第5章「cephadmによる展開」、5.4.2項「サービス仕様と配置仕様」を参照してください。たとえば、`ses-min2`という名前のホストを削除する場合、すべての`placement:`セクションから、`- ses-min2`という記載をすべて削除します。

この状態から

```
service_type: rgw
service_id: EXAMPLE_NFS
placement:
  hosts:
    - ses-min2
    - ses-min3
```

次のように変更してください。

```
service_type: rgw
```

```
service_id: EXAMPLE_NFS
placement:
  hosts:
    - ses-min3
```

変更内容を設定ファイルに適用します。

```
cephuser@adm > ceph orch apply -i rgw-example.yaml
```

## 2. cephadm環境からノードを削除します。

```
cephuser@adm > ceph orch host rm ses-min2
```

## 3. ノードがcrash.osd.1とcrash.osd.2というサービスを実行している場合、ホスト上で次のコマンドを実行してサービスを削除します。

```
root@minion > cephadm rm-daemon --fsid CLUSTER_ID --name SERVICE_NAME
```

例:

```
root@minion > cephadm rm-daemon --fsid b4b30c6e... --name crash.osd.1
root@minion > cephadm rm-daemon --fsid b4b30c6e... --name crash.osd.2
```

## 4. 削除したいミニオンからすべての役割を削除します。

```
cephuser@adm > ceph-salt config /ceph_cluster/roles/tuned/throughput remove ses-min2
cephuser@adm > ceph-salt config /ceph_cluster/roles/tuned/latency remove ses-min2
cephuser@adm > ceph-salt config /ceph_cluster/roles/cephadm remove ses-min2
cephuser@adm > ceph-salt config /ceph_cluster/roles/admin remove ses-min2
```

削除したいミニオンがブートストラップミニオンの場合、ブートストラップの役割も削除する必要があります。

```
cephuser@adm > ceph-salt config /ceph_cluster/roles/bootstrap reset
```

## 5. 1つのホストからすべてのOSDを削除したら、そのホストをCRUSHマップから削除します。

```
cephuser@adm > ceph osd crush remove bucket-name
```



### 注記

バケット名はホスト名と同じであるはずですが。

## 6. これで、クラスタからミニオンを削除できるようになります。

```
cephuser@adm > ceph-salt config /ceph_cluster/minions remove ses-min2
```

## ！ 重要

障害イベント中で、削除しようとしているミニオンが永続的な電源オフ状態である場合、Salt Masterからノードを削除する必要があります。

```
root@master # salt-key -d minion_id
```

その後、`pillar_root/ceph-salt.sls`からノードを手動で削除します。このファイルは通常、`/srv/pillar/ceph-salt.sls`に置かれています。

## 13.4 OSDの管理

このセクションではCephクラスタにOSDを追加、消去、削除する方法を説明します。

### 13.4.1 ディスクデバイスの一覧

すべてのクラスタノード上のディスクデバイスが使用中か未使用かを確認するには、次のコマンドを実行してディスクを一覧にしてください。

```
cephuser@adm > ceph orch device ls
```

HOST	PATH	TYPE	SIZE	DEVICE	AVAIL	REJECT	REASONS
ses-master	/dev/vda	hdd	42.0G		False	locked	
ses-min1	/dev/vda	hdd	42.0G		False	locked	
ses-min1	/dev/vdb	hdd	8192M	387836	False	locked, LVM detected, Insufficient space (<5GB) on vgs	
ses-min2	/dev/vdc	hdd	8192M	450575	True		

### 13.4.2 ディスクデバイスの消去

ディスクデバイスを再利用するには、まず内容を消去する必要があります。

```
ceph orch device zap HOST_NAME DISK_DEVICE
```

例:

```
cephuser@adm > ceph orch device zap ses-min2 /dev/vdc
```



## 注記

`unmanaged` フラグが設定されておらず、以前 OSD を展開した際に DriveGroups または `--all-available-devices` オプションを使用していた場合、消去後に cephadm が自動的にこれらの OSD を展開します。

### 13.4.3 DriveGroups仕様を用いたOSDの追加

「DriveGroups」では、Ceph クラスタの OSD のレイアウトを指定します。レイアウトは単独の YAML ファイルで定義します。このセクションでは、例として `drive_groups.yml` を使用します。

管理者は、相互に関連する OSD のグループ (HDD と SSD の混成環境に展開されるハイブリッド OSD) を手動で指定するか、同じ展開オプション (たとえば、同じオブジェクトストア、同じ暗号化オプション、スタンドアロン OSD など) を共有する必要があります。デバイスが明示的に一覧にされないようにするため、DriveGroups では、`ceph-volume` インベントリレポートで選択した数個のフィールドに対応するフィルタ項目のリストを使用します。cephadm では、これらの DriveGroups を実際のデバイスリストに変換してユーザが調べられるようにするコードを提供します。

OSD の仕様をクラスタに適用するには、次のコマンドを実行します。

```
cephuser@adm > ceph orch apply osd -i drive_groups.yml
```

アクションのプレビューを確認する場合や、アプリケーションをテストする場合には、`--dry-run` オプションを付加した `ceph orch apply osd` コマンドを使用できます。次に例を示します。

```
cephuser@adm > ceph orch apply osd -i drive_groups.yml --dry-run
...
+-----+-----+-----+-----+-----+-----+
|SERVICE|NAME  |HOST  |DATA      |DB  |WAL  |
+-----+-----+-----+-----+-----+-----+
|osd      |test  |mgr0  |/dev/sda  |-   |-   |
|osd      |test  |mgr0  |/dev/sdb  |-   |-   |
+-----+-----+-----+-----+-----+-----+
```

`--dry-run` の出力が想定通りなら、`--dry-run` オプションを外して再度コマンドを実行するだけです。

### 13.4.3.1 アンマネージドOSD

DriveGroups仕様と一致する、利用可能なすべてのクリーンディスクデバイスは、クラスタに追加すると自動的にOSDとして使用されます。この動作を「マネージド」モードと呼びます。

「マネージド」モードを無効化するには、`unmanaged: true`行を関連する仕様に追加してください。次に例を示します。

```
service_type: osd
service_id: example_drvgrp_name
placement:
  hosts:
    - ses-min2
    - ses-min3
encrypted: true
unmanaged: true
```



#### ヒント

すでに展開済みのOSDを「マネージド」モードから「アンマネージド」モードに切り替えるには、[13.1項「クラスタ設定の変更」](#)で説明した手順の中で`unmanaged: true`行を追加してください。

### 13.4.3.2 DriveGroups仕様

DriveGroups仕様ファイルの例を次に示します。

```
service_type: osd
service_id: example_drvgrp_name
placement:
  host_pattern: '*'
data_devices:
  drive_spec: DEVICE_SPECIFICATION
db_devices:
  drive_spec: DEVICE_SPECIFICATION
wal_devices:
  drive_spec: DEVICE_SPECIFICATION
block_wal_size: '5G' # (optional, unit suffixes permitted)
block_db_size: '5G' # (optional, unit suffixes permitted)
encrypted: true      # 'True' or 'False' (defaults to 'False')
```



## 注記

かつてのDeepSeaで「encryption」と呼ばれていたオプションは、「encrypted」に名前が変更されました。SUSE Enterprise Storage 7にDriveGroupに適用する場合は、サービス仕様にこの新しい用語が使われているかを確認してください。さもなければ、**ceph orch apply**操作は失敗します。

### 13.4.3.3 一致するディスクデバイス

次のフィルタを使用して指定を記述できます。

- ディスクモデル別。

```
model: DISK_MODEL_STRING
```

- ディスクベンダー別。

```
vendor: DISK_VENDOR_STRING
```



## ヒント

DISK\_VENDOR\_STRINGは必ず小文字で入力してください。

ディスクモデルとディスクベンダーの詳細を取得するには、次のコマンドの出力を確認してください。

```
cephuser@adm > ceph orch device ls
HOST    PATH    TYPE  SIZE DEVICE_ID          MODEL          VENDOR
ses-min1 /dev/sdb ssd   29.8G SATA_SSD_AF34075704240015  SATA SSD      ATA
ses-min2 /dev/sda ssd   223G Micron_5200_MTFDDAK240TDN  Micron_5200_MTFD ATA
[...]
```

- ディスクが回転型かどうか。SSDとNVMeドライブは回転型ではありません。

```
rotational: 0
```

- OSDで使用可能な「すべての」ドライブを使用してノードを展開します。

```
data_devices:
  all: true
```

- また、一致するディスクの数を制限します。

```
limit: 10
```

#### 13.4.3.4 サイズによるデバイスのフィルタリング

ディスクデバイスをサイズでフィルタできます(正確なサイズ、またはサイズの範囲)。size:パラメータには、次の形式の引数を指定できます。

- '10G' - 正確にこのサイズのディスクを含めます。
- '10G:40G' - この範囲内のサイズのディスクを含めます。
- ':10G' - サイズが10GB以下のディスクを含めます。
- '40G:' - サイズが40GB以上のディスクを含めます。

##### 例 13.1: ディスクサイズによる一致

```
service_type: osd
service_id: example_drvgrp_name
placement:
  host_pattern: '*'
data_devices:
  size: '40TB:'
db_devices:
  size: ':2TB'
```



#### 注記: 引用符が必要

区切り文字「:」を使用する場合は、サイズを引用符で囲む必要があります。そうしないと、「:」記号は新しい設定のハッシュであると解釈されます。



#### ヒント: 単位のショートカット

ギガバイト(G)の代わりに、メガバイト(M)やテラバイト(T)でもサイズを指定できます。

#### 13.4.3.5 DriveGroupsの例

このセクションでは、さまざまなOSDセットアップの例を示します。

##### 例 13.2: 単純なセットアップ

この例では、同じセットアップを使用する2つのノードについて説明します。

- 20台のHDD

- ベンダー: Intel
- モデル: SSD-123-foo
- サイズ: 4TB
- 2台のSSD
  - ベンダー: Micron
  - モデル: MC-55-44-ZX
  - サイズ: 512GB

対応するdrive\_groups.ymlファイルは次のようになります。

```
service_type: osd
service_id: example_drvgrp_name
placement:
  host_pattern: '*'
data_devices:
  model: SSD-123-foo
db_devices:
  model: MC-55-44-XZ
```

このような設定は単純で有効です。問題は、管理者が将来、別のベンダーのディスクを追加することがあっても、それらのディスクが含まれない点です。この設定を向上させるには、ドライブのコアプロパティのフィルタを減らします。

```
service_type: osd
service_id: example_drvgrp_name
placement:
  host_pattern: '*'
data_devices:
  rotational: 1
db_devices:
  rotational: 0
```

前の例では、回転型デバイスはすべて「データデバイス」として宣言し、非回転型デバイスはすべて「共有デバイス」(wal、db)として使用します。

2TBを超えるドライブが常に低速のデータデバイスであることがわかっている場合は、サイズでフィルタできます。

```
service_type: osd
service_id: example_drvgrp_name
placement:
  host_pattern: '*'
data_devices:
```

```
size: '2TB:'  
db_devices:  
  size: ':2TB'
```

### 例 13.3: 詳細セットアップ

この例では、2つの別個のセットアップについて説明します。20台のHDDで2台のSSDを共有するセットアップと、10台のSSDで2台のNVMeを共有するセットアップです。

- 20台のHDD
  - ベンダー: Intel
  - モデル: SSD-123-foo
  - サイズ: 4TB
- 12台のSSD
  - ベンダー: Micron
  - モデル: MC-55-44-ZX
  - サイズ: 512GB
- 2つのNVMe
  - ベンダー: Samsung
  - モデル: NVME-QQQQ-987
  - サイズ: 256GB

このようなセットアップは、次のような2つのレイアウトで定義できます。

```
service_type: osd  
service_id: example_drvgrp_name  
placement:  
  host_pattern: '*'  
data_devices:  
  rotational: 0  
db_devices:  
  model: MC-55-44-XZ
```

```
service_type: osd  
service_id: example_drvgrp_name2  
placement:  
  host_pattern: '*'  
data_devices:  
  model: MC-55-44-XZ
```

```
db_devices:
  vendor: samsung
  size: 256GB
```

#### 例 13.4: 不均一なノードを使用した詳細セットアップ

前の例では、すべてのノードに同じドライブがあることを想定しています。ただし、常にこれが当てはまるとは限りません。

ノード1～5:

- 20台のHDD
  - ベンダー: Intel
  - モデル: SSD-123-foo
  - サイズ: 4TB
- 2台のSSD
  - ベンダー: Micron
  - モデル: MC-55-44-ZX
  - サイズ: 512GB

ノード6～10:

- 5つのNVMe
  - ベンダー: Intel
  - モデル: SSD-123-foo
  - サイズ: 4TB
- 20台のSSD
  - ベンダー: Micron
  - モデル: MC-55-44-ZX
  - サイズ: 512GB

レイアウトに「target」キーを使用して、特定のノードをターゲットに設定できます。Saltのターゲット表記を使用すると、内容をシンプルに保つことができます。

```
service_type: osd
service_id: example_drvgrp_one2five
```

```
placement:
  host_pattern: 'node[1-5]'
data_devices:
  rotational: 1
db_devices:
  rotational: 0
```

続いて以下を設定します。

```
service_type: osd
service_id: example_drvgrp_rest
placement:
  host_pattern: 'node[6-10]'
data_devices:
  model: MC-55-44-XZ
db_devices:
  model: SSD-123-foo
```

#### 例 13.5: エキスパートセットアップ

前の事例はすべて、WALとDBが同じデバイスを使用することを想定していました。ただし、WALを専用のデバイスに展開することもできます。

- 20台のHDD
  - ベンダー: Intel
  - モデル: SSD-123-foo
  - サイズ: 4TB
- 2台のSSD
  - ベンダー: Micron
  - モデル: MC-55-44-ZX
  - サイズ: 512GB
- 2つのNVMe
  - ベンダー: Samsung
  - モデル: NVME-YYYY-987
  - サイズ: 256GB

```
service_type: osd
service_id: example_drvgrp_name
placement:
```

```
host_pattern: '*'
data_devices:
  model: MC-55-44-XZ
db_devices:
  model: SSD-123-foo
wal_devices:
  model: NVME-QQQQ-987
```

#### 例 13.6: 複雑な(可能性が低い)セットアップ

次のセットアップでは、以下を定義してみます。

- 1つのNVMeを利用する20台のHDD
- 1台のSSD (db)と1つのNVMe (wal)を利用する2台のHDD
- 1つのNVMeを利用する8台のSSD
- 2台SSDスタンドアロン(暗号化)
- 1台のHDDはスペアで、展開しない

使用するドライブの概要は次のとおりです。

- 23台のHDD
  - ベンダー: Intel
  - モデル: SSD-123-foo
  - サイズ: 4TB
- 10台のSSD
  - ベンダー: Micron
  - モデル: MC-55-44-ZX
  - サイズ: 512GB
- 1つのNVMe
  - ベンダー: Samsung
  - モデル: NVME-QQQQ-987
  - サイズ: 256GB

DriveGroupsの定義は次のようになります。

```
service_type: osd
```

```
service_id: example_drvgrp_hdd_nvme
placement:
  host_pattern: '*'
data_devices:
  rotational: 0
db_devices:
  model: NVME-QQQQ-987
```

```
service_type: osd
service_id: example_drvgrp_hdd_ssd_nvme
placement:
  host_pattern: '*'
data_devices:
  rotational: 0
db_devices:
  model: MC-55-44-XZ
wal_devices:
  model: NVME-QQQQ-987
```

```
service_type: osd
service_id: example_drvgrp_ssd_nvme
placement:
  host_pattern: '*'
data_devices:
  model: SSD-123-foo
db_devices:
  model: NVME-QQQQ-987
```

```
service_type: osd
service_id: example_drvgrp_standalone_encrypted
placement:
  host_pattern: '*'
data_devices:
  model: SSD-123-foo
encrypted: True
```

ファイルが上から下へ解析されると、HDDが1台残ります。

## 13.4.4 OSDの削除

クラスタからOSDノードを削除する前に、クラスタの空きディスク容量が、削除予定のOSDディスクの容量以上であることを確認してください。OSDを削除すると、クラスタ全体のリバランスが発生することに注意してください。

1. IDを取得して、削除するOSDを特定します。

```
cephuser@adm > ceph orch ps --daemon_type osd
```

NAME	HOST	STATUS	REFRESHED	AGE	VERSION
osd.0	target-ses-090	running (3h)	7m ago	3h	15.2.7.689 ...
osd.1	target-ses-090	running (3h)	7m ago	3h	15.2.7.689 ...
osd.2	target-ses-090	running (3h)	7m ago	3h	15.2.7.689 ...
osd.3	target-ses-090	running (3h)	7m ago	3h	15.2.7.689 ...

## 2. クラスタから1つ以上のOSDを削除します。

```
cephuser@adm > ceph orch osd rm OSD1_ID OSD2_ID ...
```

例:

```
cephuser@adm > ceph orch osd rm 1 2
```

## 3. 削除操作の状態をクエリすることができます。

```
cephuser@adm > ceph orch osd rm status
```

OSD_ID	HOST	STATE	PG_COUNT	REPLACE	FORCE	STARTED_AT
2	cephadm-dev	done, waiting for purge	0	True	False	2020-07-17 13:01:43.147684
3	cephadm-dev	draining	17	False	True	2020-07-17 13:01:45.162158
4	cephadm-dev	started	42	False	True	2020-07-17 13:01:45.162158

### 13.4.4.1 OSD削除の中止

OSDの削除をスケジュールした後、必要に応じて削除を停止できます。次のコマンドを実行すると、OSDの初期状態がリセットされ、キューから削除されます。

```
cephuser@adm > ceph orch osd rm stop OSD_SERVICE_ID
```

## 13.4.5 OSDの交換

IDを維持したままOSDを交換するには、次のコマンドを実行します。

```
cephuser@adm > ceph orch osd rm OSD_SERVICE_ID --replace
```

例:

```
cephuser@adm > ceph orch osd rm 4 --replace
```

OSDの交換は、OSDがCRUSH階層から永久に削除されることがなく、代わりにdestroyedフラグが割り当てられることを除けば、OSDの削除と同じです(詳細については、[13.4.4項「OSDの削除」](#)を参照してください)。

`destroyed`フラグは、次回OSDを展開した際に再利用するOSD IDを特定するために使用されます。新しく追加されたディスクがDriveGroups仕様と一致する場合、そのディスクには交換されたディスクのOSD IDが割り当てられます(詳細については、[13.4.3項「DriveGroups仕様を用いたOSDの追加」](#)を参照してください)。



## ヒント

`--dry-run`オプションを付加すると、実際の交換は行われませんが、通常発生する手順をプレビューします。

## 13.5 新しいノードへのSalt Masterの移動

Salt Masterのホストを新しいものに交換する必要がある場合、次の手順に従います。

1. クラスタ設定をエクスポートし、エクスポートされたJSONファイルをバックアップします。詳細については、『導入ガイド』、第5章「cephadmによる展開」、5.3.2.15項「クラスタ設定のエクスポート」を参照してください。
2. 古いSalt Masterがクラスタで唯一の管理ノードでもある場合、`/etc/ceph/ceph.client.admin.keyring`と`/etc/ceph/ceph.conf`を新しいSalt Masterに手動で移動します。
3. 古いSalt Masterノードで、Salt Masterの`systemd`サービスを停止し無効化します。

```
root@master # systemctl stop salt-master.service
root@master # systemctl disable salt-master.service
```

4. 古いSalt Masterノードがすでにクラスタ内に存在しない場合、Salt Minionの`systemd`サービスも停止し無効化します。

```
root@master # systemctl stop salt-minion.service
root@master # systemctl disable salt-minion.service
```



## 警告

古いSalt MasterノードがいずれかのCephデーモン(MON、MGR、OSD、MDS、ゲートウェイ、監視)を実行している場合は、`salt-minion.service`の停止と無効化を行わないでください。

5. SUSE Linux Enterprise Server 15 SP2を新しいSalt Masterにインストールします。手順は『導入ガイド』、第5章「cephadmによる展開」、5.1項「SUSE Linux Enterprise Serverのインストールと設定」を参照してください。



## ヒント: Salt Minionの移行

Salt Minionを新しいSalt Masterへ簡単に移行するには、各Minionから元のSalt Masterの公開鍵を削除します。

```
root@minion > rm /etc/salt/pki/minion/minion_master.pub
root@minion > systemctl restart salt-minion.service
```

6. 新しいSalt Masterに `salt-master` パッケージをインストールし、必要に応じて `salt-minion` パッケージもインストールします。
7. 新しいSalt Masterノードに `ceph-salt` をインストールします。

```
root@master # zypper install ceph-salt
root@master # systemctl restart salt-master.service
root@master # salt '*' saltutil.sync_all
```



## 重要

次の手順に進む前に、3つすべてのコマンドを実行したか確認してください。これらのコマンドはべき等です。つまり、同じコマンドを複数回実行しても問題ありません。

8. 新しいSalt Masterをクラスタに取り込みます。手順は『導入ガイド』、第5章「cephadmによる展開」、5.3.1項「ceph-saltのインストール」と『導入ガイド』、第5章「cephadmによる展開」、5.3.2.2項「Salt Minionの追加」、『導入ガイド』、第5章「cephadmによる展開」、5.3.2.4項「管理ノードの指定」を参照してください。
9. バックアップしたクラスタ設定をインポートして適用します。

```
root@master # ceph-salt import CLUSTER_CONFIG.json
root@master # ceph-salt apply
```



## 重要

インポートする前に、エクスポートされた `CLUSTER_CONFIG.json` ファイルに記載されたSalt Masterの `minion id` の名前を変更します。

## 13.6 クラスタノードの更新

ローリングアップデートを定期的に適用して、Cephクラスタノードを最新の状態に保ちます。

### 13.6.1 ソフトウェアリポジトリ

最新のソフトウェアパッケージのパッチをクラスタに適用する前に、クラスタのすべてのノードが関連するリポジトリにアクセスできることを確認します。必要なリポジトリの完全なリストについては、『導入ガイド』、第7章「前回リリースからのアップグレード」、7.1.5.1項「ソフトウェアリポジトリ」を参照してください。

### 13.6.2 リポジトリのステージング

クラスタノードにソフトウェアリポジトリを提供するステージングツール(SUSE Manager、RMT (Repository Management Tool)など)を使用する場合、SUSE Linux Enterprise ServerとSUSE Enterprise Storageの両方の「更新」リポジトリのステージが同じ時点で作成されていることを確認します。

ステージングツールを使用して、パッチレベルが`frozen`または`staged`のパッチを適用することを強く推奨します。これにより、クラスタに参加している新しいノードと、クラスタですでに動作しているノードが確実に同じパッチレベルになるようにします。また、新しいノードがクラスタに参加する前に、クラスタのすべてのノードに最新のパッチを適用する必要がなくなります。

### 13.6.3 Cephサービスのダウンタイム

設定によっては、更新中にクラスタノードが再起動される場合があります。Object Gateway、Samba Gateway、NFS Ganesha、iSCSIなど、サービスの単一障害点があると、再起動されるノードに存在するサービスからクライアントマシンが一時的に切断される場合があります。

### 13.6.4 更新の実行

すべてのクラスタノードでソフトウェアパッケージを最新バージョンに更新するには、次のコマンドを実行します。

```
root@master # ceph-salt update
```

## 13.7 Cephの更新

cephadmに対して、Cephを更新して各バグフィックスリリースを適用するように指示できます。Cephサービスの自動更新は、推奨される順番を尊重します。つまり、まずはCeph ManagerとCeph Monitorから開始し、その後、Ceph OSD、メタデータサーバ、Object Gatewayなどのその他のサービスに進みます。クラスタの利用を継続できることをCephが示すまで、各デーモンは再起動されません。



### 注記

以下の更新手順では、**ceph orch upgrade**コマンドを使用します。以下の手順は、ある製品バージョンのCephクラスタを更新する方法を詳細に説明したもので(たとえば、保守更新)、クラスタをある製品バージョンから別の製品バージョンにアップグレードする方法を説明したものではないことに注意してください。

### 13.7.1 更新の開始

更新を開始する前に、すべてのノードが現在オンラインであり、クラスタが正常な状態であることを確認してください。

```
cephuser@adm > cephadm shell -- ceph -s
```

特定のCephリリースに更新するには、次のコマンドを使用します。

```
cephuser@adm > ceph orch upgrade start --image REGISTRY_URL
```

例:

```
cephuser@adm > ceph orch upgrade start --image registry.suse.com/ses/7/ceph/ceph:latest
```

ホスト上でパッケージをアップグレードします。

```
cephuser@adm > ceph-salt update
```

### 13.7.2 更新の監視

更新が進行中かどうかを確認するには、次のコマンドを実行します。

```
cephuser@adm > ceph orch upgrade status
```

更新が進行中であれば、Cephの状態出力でプログレスバーを確認できます。

```
cephuser@adm > ceph -s
[...]
progress:
  Upgrade to registry.suse.com/ses/7/ceph/ceph:latest (00h 20m 12s)
  [=====.....] (time remaining: 01h 43m 31s)
```

cephadmのログを監視することもできます。

```
cephuser@adm > ceph -W cephadm
```

### 13.7.3 更新のキャンセル

更新処理はいつでも中止できます。

```
cephuser@adm > ceph orch upgrade stop
```

## 13.8 クラスタの停止または再起動

場合によっては、クラスタ全体を停止または再起動しなければならないことがあります。実行中のサービスの依存関係を入念に確認することをお勧めします。次の手順では、クラスタの停止と起動の概要を説明します。

1. OSDにoutのマークを付けないようCephクラスタに指示します。

```
cephuser@adm > ceph osd set noout
```

2. 次の順序でデーモンとノードを停止します。

1. ストレージクライアント
2. ゲートウェイ(たとえば、NFS Ganesha、Object Gateway)
3. メタデータサーバ
4. Ceph OSD
5. Ceph Manager
6. Ceph Monitor

3. 必要に応じて、保守タスクを実行します。
4. ノードとサーバをシャットダウンプロセスの逆の順序で起動します。
  1. Ceph Monitor
  2. Ceph Manager
  3. Ceph OSD
  4. メタデータサーバ
  5. ゲートウェイ(たとえば、NFS Ganesha、Object Gateway)
  6. ストレージクライアント
5. nooutフラグを削除します。

```
cephuser@adm > ceph osd unset noout
```

## 13.9 Cephクラスタ全体の削除

**ceph-salt purge** コマンドを実行すると、Cephクラスタ全体が削除されます。複数のCephクラスタを展開している場合は、**ceph -s** コマンドでレポートされるクラスタがパージされます。これにより、異なる設定をテストする際にクラスタ環境をクリーンにすることができます。

誤って削除されないことがないよう、オーケストレーションは、セキュリティ対策が解除されているかどうかをチェックします。次のコマンドを実行して、セキュリティ対策を解除してCephクラスタを削除できます。

```
root@master # ceph-salt disengage-safety
root@master # ceph-salt purge
```

## 14 Cephサービスの運用

Cephサービスは、デーモンレベル、ノードレベル、またはクラスタレベルで運用できます。必要なアプローチに応じて、`cephadm`か`systemctl`コマンドを使用してください。

### 14.1 個別のサービスの運用

個別のサービスを運用する必要がある場合は、まずそのサービスを確認します。

```
cephuser@adm > ceph orch ps
```

NAME	HOST	STATUS	REFRESHED	[...]
mds.my_cephfs.ses-min1.oterul	ses-min1	running (5d)	8m ago	
mgr.ses-min1.gpijpm	ses-min1	running (5d)	8m ago	
mgr.ses-min2.oopvyh	ses-min2	running (5d)	8m ago	
mon.ses-min1	ses-min1	running (5d)	8m ago	
mon.ses-min2	ses-min2	running (5d)	8m ago	
mon.ses-min4	ses-min4	running (5d)	7m ago	
osd.0	ses-min2	running (61m)	8m ago	
osd.1	ses-min3	running (61m)	7m ago	
osd.2	ses-min4	running (61m)	7m ago	
rgw.myrealm.myzone.ses-min1.kkwazo	ses-min1	running (5d)	8m ago	
rgw.myrealm.myzone.ses-min2.jngabw	ses-min2	error	8m ago	

特定のノードのサービスを確認するには、次のコマンドを実行します。

```
ceph orch ps NODE_HOST_NAME
```

以下に例を示します。

```
cephuser@adm > ceph orch ps ses-min2
```

NAME	HOST	STATUS	REFRESHED
mgr.ses-min2.oopvyh	ses-min2	running (5d)	3m ago
mon.ses-min2	ses-min2	running (5d)	3m ago
osd.0	ses-min2	running (67m)	3m ago



#### ヒント

`ceph orch ps`コマンドはいくつかの出力フォーマットをサポートしています。フォーマットを変更するには`--format FORMAT`オプションを付加してください。[ここ](#)で、`FORMAT`は`json`、`json-pretty`、または`yaml`のいずれかです。例:

```
cephuser@adm > ceph orch ps --format yaml
```

サービスの名前を確認することで、サービスの起動、再起動、停止が可能になります。

```
ceph orch daemon COMMAND SERVICE_NAME
```

たとえば、IDが0のOSDサービスを再起動するには、次のコマンドを実行します。

```
cephuser@adm > ceph orch daemon restart osd.0
```

## 14.2 サービスタイプの運用

Cephクラスタ全体で特定のタイプのサービスを運用する必要がある場合は、次のコマンドを使用します。

```
ceph orch COMMAND SERVICE_TYPE
```

`COMMAND`は`start`、`stop`、または`restart`のいずれかで置き換えます。

たとえば、クラスタのすべてのMONを再起動するには次のコマンドを使用します。このコマンドでは、MONが実際にはどのノードで実行されているかを考慮しません。

```
cephuser@adm > ceph orch restart mon
```

## 14.3 単一のノードでサービスを運用する

`systemctl`コマンドを使用することで、Cephに関連した`systemd`サービスとターゲットを単一のノードで運用できます。

### 14.3.1 サービスとターゲットの確認

Cephに関連した`systemd`サービスとターゲットを運用する前に、これらのユニットファイルのファイル名を確認する必要があります。サービスのファイル名には次のようなパターンがあります。

```
ceph-FSID@SERVICE_TYPE.ID.service
```

例:

```
ceph-b4b30c6e-9681-11ea-ac39-525400d7702d@mon.doc-ses-min1.service
```

```
ceph-b4b30c6e-9681-11ea-ac39-525400d7702d@rgw.myrealm.myzone.doc-ses-min1.kwwazo.service
```

## FSID

Cephクラスタの固有のIDです。`ceph fsid`コマンドの出力で確認できます。

## SERVICE\_TYPE

サービスの種類です。たとえば、`osd`、`mon`、`rgw`などがあります。

## ID

サービスの識別文字列です。OSDの場合はサービスのID番号です。他のサービスでは、ノードのホスト名の場合と、サービスタイプに関連した追加の文字列の場合があります。



## ヒント

`SERVICE_TYPE.ID`の部分は、`ceph orch ps`コマンドの出力のNAME列の内容と同一です。

## 14.3.2 ノード上のすべてのサービスの運用

Cephのsystemdターゲットを使用することで、ノード上の「すべて」のサービスを同時に運用することができます。あるいは、`FSID`で識別された「クラスタに属する」すべてのサービスを同時に運用することもできます。

たとえば、サービスがどのクラスタに所属しているかを考慮せずにノード上のCephサービスをすべて停止するには、次のコマンドを実行します。

```
root@minion > systemctl stop ceph.target
```

IDが**b4b30c6e-9681-11ea-ac39-525400d7702d**のCephクラスタに属するすべてのサービスを再起動するには、次のコマンドを実行します。

```
root@minion > systemctl restart ceph-b4b30c6e-9681-11ea-ac39-525400d7702d.target
```

## 14.3.3 ノード上の個別のサービスの運用

特定のサービスの名前を確認したら、次のように運用します。

```
systemctl COMMAND SERVICE_NAME
```

たとえば、IDが**b4b30c6e-9681-11ea-ac39-525400d7702d**のクラスタ上にある、IDが1のOSDサービスを単独で再起動する場合は、次のコマンドを実行します。

```
root # systemctl restart ceph-b4b30c6e-9681-11ea-ac39-525400d7702d@osd.1.service
```

### 14.3.4 サービス状態のクエリ

サービスの状態をsystemdに問い合わせることができます。例:

```
root # systemctl status ceph-b4b30c6e-9681-11ea-ac39-525400d7702d@osd.0.service
```

## 14.4 Cephクラスタ全体のシャットダウンと再起動

クラスタのシャットダウンと再起動は、計画停電の際に必要となる場合があります。すべてのCeph関連サービスを停止し、正常に再起動させるには、以下の手順に従います。

#### 手順 14.1: CEPHクラスタ全体のシャットダウン

1. クラスタにアクセスしているすべてのクライアントをシャットダウンするか、切断します。
2. CRUSHが自動的にクラスタをリバランスしないように、クラスタをnooutに設定します。

```
cephuser@adm > ceph osd set noout
```

3. すべてのクラスタノードのすべてのCephサービスを停止します。

```
root@master # ceph-salt stop
```

4. すべてのクラスタノードの電源を切ります。

```
root@master # salt -G 'ceph-salt:member' cmd.run "shutdown -h"
```

#### 手順 14.2: CEPHクラスタ全体の起動

1. 管理ノードの電源を入れます。
2. Ceph Monitorノードの電源を入れます。
3. Ceph OSDノードの電源を入れます。
4. 以前に設定したnooutフラグの設定を解除します。

```
root@master # ceph osd unset noout
```

5. すべての設定されているゲートウェイの電源を入れます。
6. クラスタのクライアントの電源を入れるか、接続します。

## 15 バックアップおよび復元

この章では、Cephクラスタの機能を復元できるようにするには、クラスタのどの部分をバックアップする必要があるかについて説明します。

### 15.1 クラスタ設定とデータのバックアップ

#### 15.1.1 ceph-salt設定のバックアップ

クラスタ設定をエクスポートします。詳細情報については、『導入ガイド』、第5章「cephadmによる展開」、5.3.2.15項「クラスタ設定のエクスポート」を参照してください。

#### 15.1.2 Ceph設定のバックアップ

/etc/cephディレクトリをバックアップします。ここには、重要なクラスタ設定が含まれています。たとえば、管理ノードを交換する必要がある場合、/etc/cephのバックアップが必要になります。

#### 15.1.3 Salt設定のバックアップ

/etc/saltディレクトリをバックアップする必要があります。ここには、Salt Masterのキーや受け付けたクライアントキーなど、Saltの各種設定ファイルが含まれています。

管理ノードをバックアップする場合にSaltのファイルは厳密には必須ではありませんが、バックアップしておくとSaltクラスタの再展開が容易になります。これらのファイルのバックアップがないと、新しい管理ノードで再度Salt Minionを登録する必要があります。



#### 注記: Salt Masterの秘密鍵のセキュリティ

Salt Masterの秘密鍵のバックアップは必ず安全な場所に保存してください。Salt Masterのキーを使用すると、すべてのクラスタノードを操作できます。

## 15.1.4 カスタム設定のバックアップ

- Prometheusのデータおよびカスタマイズ。
- Grafanaのカスタマイズ。
- iSCSI設定の手動変更。
- Cephの各種キー。
- CRUSHマップおよびCRUSHルール。次のコマンドを実行して、CRUSHルールを含む、逆コンパイルされたCRUSHマップを`crushmap-backup.txt`に保存します。

```
cephuser@adm > ceph osd getcrushmap | crushtool -d - -o crushmap-backup.txt
```

- Samba Gatewayの設定。ゲートウェイを1つ使用している場合は、`/etc/samba/smb.conf`をバックアップします。HAセットアップを使用している場合は、CTDB設定ファイルとPacemaker設定ファイルもバックアップします。Samba Gatewayで使用する設定の詳細については、[第24章「Sambaを介したCephデータのエクスポート」](#)を参照してください。
- NFS Ganeshaの設定。HAセットアップの使用時にのみ必要です。NFS Ganeshaで使用する設定の詳細については、[第25章「NFS Ganesha」](#)を参照してください。

## 15.2 Cephノードの復元

バックアップからノードを復元する手順では、ノードを再インストールして設定ファイルを置き換えた後、置換ノードが再度追加されるようにクラスタを再度オーケストレーションします。

管理ノードの再展開が必要な場合は、[13.5項「新しいノードへのSalt Masterの移動」](#)を参照してください。

ミニオンについては、単に再構築と再展開を行った方が簡単な場合が多いです。

1. ノードを再インストールします。詳細については、『導入ガイド』、第5章「cephadmによる展開」、5.1項「SUSE Linux Enterprise Serverのインストールと設定」を参照してください。
2. Saltをインストールします。詳細については『導入ガイド』、第5章「cephadmによる展開」、5.2項「Saltの展開」を参照してください。
3. `/etc/salt`ディレクトリがバックアップから復元されると、関連するSaltサービスを再開できるようになります。以下に例を示します。

```
root@master # systemctl enable salt-master  
root@master # systemctl start salt-master  
root@master # systemctl enable salt-minion  
root@master # systemctl start salt-minion
```

4. すべてのミニオンから古いSalt Masterノード用の公開マスター鍵を削除します。

```
root@master # rm /etc/salt/pki/minion/minion_master.pub  
root@master # systemctl restart salt-minion
```

5. 管理ノードのローカルに置かれていたファイルをすべて復元します。
6. 事前にエクスポートしたJSONファイルからクラスタ設定をインポートします。詳細については、『導入ガイド』、第5章「cephadmによる展開」、5.3.2.15項「クラスタ設定のエクスポート」を参照してください。
7. インポートしたクラスタ設定を適用します。

```
root@master # ceph-salt apply
```

## 16 監視とアラート

SUSE Enterprise Storage 7では、cephadmが監視スタックとアラートスタックを展開します。ユーザはcephadmを使用して展開したいサービス(Prometheus、Alertmanager、Grafanaなど)をYAML設定ファイルで指定する必要がありますが、CLIを使用してサービスを展開することもできます。同じ種類のサービスが複数展開される場合は、高可用性セットアップが展開されます。ノードエクスポートはこのルールの例外です。

cephadmを使用して以下の監視サービスを展開できます。

- 「Prometheus」は監視およびアラートツールキットです。このツールはPrometheusエクスポートからのデータを収集し、事前に定義されたしきい値に達した場合は事前に設定されたアラートを発します。
- 「Alertmanager」はPrometheusサーバによって送信されるアラートを処理します。このツールはアラートの重複排除、グループ化、ルーティングを行って適切な受信者に届けます。デフォルトでは、自動的にCephダッシュボードが受信者に設定されます。
- 「Grafana」は視覚化およびアラートソフトウェアです。この監視スタックではGrafanaのアラート機能を使用しません。そのかわり、アラートにはAlertmanagerを使用します。
- 「ノードエクスポート」はPrometheus用エクスポートで、インストールしたノードに関するデータを提供します。すべてのノードにノードエクスポートをインストールすることをお勧めします。

Prometheus Manager ModuleはPrometheusエクスポートを提供し、`ceph-mgr`収集ポイントのCephパフォーマンスカウンタを伝達します。

「スクレイピング」対象(メトリクスを提供するデーモン)などの、Prometheus設定はcephadmが自動的に設定します。cephadmはデフォルトアラートのリストも展開します。たとえば、`health error`、`10% OSDs down`、`pgs inactive`などです。

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。ユーザは手持ちのTLS証明書を支給するか、自己署名証明書を利用できます。Grafanaが展開される前にカスタム証明書を設定していない場合は、自動的に自己署名証明書が作成され、Grafana用に設定されます。

Grafana用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。

```
cephuser@adm > ceph config-key set mgr/cephadm/grafana_key -i $PWD/key.pem
cephuser@adm > ceph config-key set mgr/cephadm/grafana_cert -i $PWD/certificate.pem
```

AlertmanagerはPrometheusサーバによって送信されるアラートを処理します。重複排除、グループ化、正しいレシーバへのルーティングを行います。アラートはAlertmanagerを用いて停止できますが、Cephダッシュボードからも管理できます。

すべてのノードにノードエクスポートを展開することをお勧めします。展開にはnode-exporterサービスタイプを記載したmonitoring.yamlファイルを使用できます。サービスの展開の詳細については、『導入ガイド』、第5章「cephadmによる展開」、5.4.3.8項「監視スタックの展開」を参照してください。

## 16.1 カスタムイメージまたはローカルイメージの設定



### ヒント

このセクションでは、サービスの展開やアップグレードを行う際に使用するコンテナイメージの設定を変更する方法を説明します。サービスの展開または再展開に必要なコマンドは含まれません。

監視スタックを展開する方法としては、『導入ガイド』、第5章「cephadmによる展開」、5.4.3.8項「監視スタックの展開」に記載されているような、監視スタックの仕様を適用する方法をお勧めします。

カスタムイメージまたはローカルコンテナイメージを展開するには、イメージをcephadmに設定しなければなりません。そのためには、次のコマンドを実行する必要があります。

```
cephuser@adm > ceph config set mgr mgr/cephadm/OPTION_NAME VALUE
```

OPTION\_NAMEには、次のいずれかの名前が入ります。

- container\_image\_prometheus
- container\_image\_node\_exporter
- container\_image\_alertmanager
- container\_image\_grafana

オプションが設定されていないか、設定が削除されている場合は、以下のイメージをVALUEとして使用します。

- registry.suse.com/caasp/v4.5/prometheus-server:2.18.0
- registry.suse.com/caasp/v4.5/prometheus-node-exporter:0.18.1

- [registry.suse.com/caasp/v4.5/prometheus-alertmanager:0.16.2](https://registry.suse.com/caasp/v4.5/prometheus-alertmanager:0.16.2)
- [registry.suse.com/ses/7/ceph/grafana:7.0.3](https://registry.suse.com/ses/7/ceph/grafana:7.0.3)

以下に例を示します。

```
cephuser@adm > ceph config set mgr mgr/cephadm/container_image_prometheus prom/  
prometheus:v1.4.1
```



## 注記

カスタムイメージを設定すると、デフォルトの値は無視されます(ただし、上書きはされません)。デフォルトの値はアップデートが利用可能になった際に変更されます。カスタムイメージを設定すると、カスタムイメージを設定したコンポーネントの自動アップデートができなくなります。アップデートをインストール可能にするには、手動で設定(イメージ名とタグ)をアップデートする必要があります。

推奨設定を使用することを選択する場合、以前に設定したカスタムイメージをリセットすることができます。リセット後は再びデフォルト値が使用されます。設定オプションをリセットするには、**`ceph config rm`**を使用してください。

```
cephuser@adm > ceph config rm mgr mgr/cephadm/OPTION_NAME
```

以下に例を示します。

```
cephuser@adm > ceph config rm mgr mgr/cephadm/container_image_prometheus
```

## 16.2 監視サービスのアップデート

16.1項「カスタムイメージまたはローカルイメージの設定」で述べたように、cephadmは推奨されるテスト済みのコンテナイメージのURLが設定された状態で提供されます。これらのイメージはデフォルト値として使用されています。

Cephパッケージをアップデートすると、これらのURLの新しいバージョンが提供される場合があります。この場合、コンテナイメージの取得元がアップデートされるだけで、サービスはアップデートされません。

16.1項「カスタムイメージまたはローカルイメージの設定」で述べた手動によるアップデート、またはCephパッケージのアップデートにともなう自動アップデートによって、コンテナイメージのURLが最新版にアップデートされると、監視サービスをアップデートできるようになります。

監視サービスのアップデートには`ceph orch redeploy`を使用します。以下に例を示します。

```
cephuser@adm > ceph orch redeploy node-exporter
cephuser@adm > ceph orch redeploy prometheus
cephuser@adm > ceph orch redeploy alertmanager
cephuser@adm > ceph orch redeploy grafana
```

今のところ、1つのコマンドで存在するすべての監視サービスをアップデートすることはできません。監視サービスをアップデートする順番は重要ではありません。



## 注記

カスタムコンテナイメージを使用している場合、Cephパッケージがアップデートされても監視サービス用のURLは自動的に変更されません。カスタムコンテナイメージを指定している場合は、手動で新しいコンテナイメージのURLを指定する必要があります。たとえば、ローカルコンテナレジストリを使用する場合に、このようなケースが生じます。

使用をお勧めするコンテナイメージのURLについては、[16.1項「カスタムイメージまたはローカルイメージの設定」](#)セクションを参照してください。

## 16.3 監視の無効化

監視スタックを無効化するには、次のコマンドを実行します。

```
cephuser@adm > ceph orch rm grafana
cephuser@adm > ceph orch rm prometheus --force # this will delete metrics data collected so far
cephuser@adm > ceph orch rm node-exporter
cephuser@adm > ceph orch rm alertmanager
cephuser@adm > ceph mgr module disable prometheus
```

## 16.4 Grafanaの設定

CephダッシュボードのバックエンドはGrafana URLを必要とします。これは、フロントエンドがGrafanaダッシュボードの有無を確認してからロードできるようにするためです。CephダッシュボードにGrafanaを実装している方法の性質から、CephダッシュボードでGrafanaのグラフを確認できるようにするには、2つの作業を結びつける必要があります。

- バックエンド(Ceph MGRモジュール)は要求されたグラフの存在を確認する必要があります。この要求が正常に完了した場合、バックエンドはフロントエンドに対してGrafanaに安全にアクセスできることを通知します。
- その後、フロントエンドは`iframe`を使用してユーザのブラウザから直接Grafanaのグラフを要求します。Cephダッシュボードを通じて、迂回することなくGrafanaのインスタンスに直接アクセスします。

お客様の環境によっては、ユーザのブラウザがCephダッシュボードで設定したURLに直接アクセスすることが難しい場合もあります。対策としては、フロントエンド(ユーザのブラウザ)がGrafanaへのアクセスに使うべきURLを伝えるためだけに使用する、別のURLを設定する方法があります。

フロントエンドに返答するためのURLを変更するには、次のコマンドを実行します。

```
cephuser@adm > ceph dashboard set-grafana-frontend-api-url GRAFANA-SERVER-URL
```

コマンド中でオプションの値を指定していない場合は、`GRAFANA_API_URL`オプションの値が利用されます。この値は、`cephadm`のアップデートにより自動的かつ定期的に設定されます。オプションの値を指定した場合は、ブラウザが指定したURLを使ってGrafanaにアクセスするように設定されます。

## 16.5 Prometheus Manager Moduleの設定

Prometheus Manager ModuleはCephの内部モジュールの1つで、Cephの機能を拡張します。このモジュールはCephから(メタ)データを読み込み、Cephの状態やヘルスに関する情報を取得します。そして、Prometheusが利用できるフォーマットで(スクレイピングされた)データを提供します。



### 注記

設定変更を適用するには、Prometheus Manager Moduleを再起動する必要があります。

### 16.5.1 ネットワークインタフェースの設定

デフォルトでは、Prometheus Manager ModuleはIPv4およびIPv6の全アドレスからのHTTPリクエストをホストのポート9283番で受け付けます。ポートとリッスンアドレスは`ceph config-key set mgr/prometheus/server_addr`キーと`mgr/prometheus/server_port`キーを使用して設定可能です。このポートはPrometheusのレジストリに登録されます。

server\_addrをアップデートするには、次のコマンドを実行します。

```
cephuser@adm > ceph config set mgr mgr/prometheus/server_addr 0.0.0.0
```

server\_portをアップデートするには、次のコマンドを実行します。

```
cephuser@adm > ceph config set mgr mgr/prometheus/server_port 9283
```

## 16.5.2 scrape\_intervalの設定

デフォルトでは、Prometheus Manager Moduleのスクレイピング間隔は15秒に設定されています。スクレイピング間隔を10秒未満にすることはお勧めしません。Prometheusモジュールに別のスクレイピング間隔を設定するには、scrape\_intervalを希望する値に設定してください。



### 重要

正常な動作と不具合の防止のため、このモジュールのscrape\_intervalとPrometheusのスクレイピング間隔は常に一致するように設定する必要があります。

```
cephuser@adm > ceph config set mgr mgr/prometheus/scrape_interval 15
```

## 16.5.3 キャッシュの設定

大規模なクラスター(OSDが1000を超えるクラスター)では、メトリクスの取得にかかる時間が重要になる場合があります。キャッシュを使用しないと、Prometheus Manager Moduleがマネージャを過負荷状態に陥らせ、Ceph Managerインスタンスが応答しなくなったり、クラッシュするおそれがあります。そのため、キャッシュ機能はデフォルトで有効に設定してあり、無効にできません。しかしその結果、キャッシュが古くなる可能性があります。Cephからのメトリクスの取得にかかる時間がscrape\_intervalの設定を超えると、キャッシュは古くなったと見なされます。

この場合、警告が記録されるとともに、モジュールが次のいずれかの動作をします。

- HTTPステータスコード503(service unavailable)を返します。
- キャッシュが古くなっている可能性を承知の上で、キャッシュのコンテンツを返します。

この動作は、**ceph config set**コマンドにより設定できます。

古くなっている可能性のあるデータを返すことをモジュールに指示するには、returnに設定してください。

```
cephuser@adm > ceph config set mgr mgr/prometheus/stale_cache_strategy return
```

service unavailableを返すことをモジュールに指示するには、failに設定してください。

```
cephuser@adm > ceph config set mgr mgr/prometheus/stale_cache_strategy fail
```

## 16.5.4 RBDイメージ監視の有効化

必要に応じて、Prometheus Manager ModuleはRBDイメージごとのIO統計状態を収集できます。そのためには、動的OSDパフォーマンスカウンタを有効化します。mgr/prometheus/rbd\_stats\_pools設定パラメータで指定したプールのすべてのイメージについて、統計情報が収集されます。

パラメータはpool[/namespace]エントリのカンマスペースで区切ったリストです。namespaceを指定しない場合は、プールのすべてのネームスペースについて統計情報が収集されます。

以下に例を示します。

```
cephuser@adm > ceph config set mgr mgr/prometheus/rbd_stats_pools "pool1,pool2,poolN"
```

モジュールは指定されたプールとネームスペースをスキャンして、利用可能なイメージのリストを作成します。さらに、モジュールはリストを定期的に更新します。更新間隔はmgr/prometheus/rbd\_stats\_pools\_refresh\_intervalパラメータにより設定可能です(秒単位)。デフォルト値は300秒(5分)です。

たとえば、同期間隔を10分に変更するには次のコマンドを実行します。

```
cephuser@adm > ceph config set mgr mgr/prometheus/rbd_stats_pools_refresh_interval 600
```

## 16.6 Prometheusのセキュリティモデル

Prometheusのセキュリティモデルは、信頼されていないユーザがPrometheus HTTPのエンドポイントとログにアクセスできる場合を仮定しています。この条件では、信頼されていないユーザが、データベースに格納されているPrometheusが収集したすべての(メタ)データや、さまざまな運用情報とデバッグ情報にアクセスできます。

しかし、PrometheusのHTTP APIは読み込み専用の動作しかできないように制限されています。APIにより設定を変更することはできないため、機密情報は持ち出せません。さらに、PrometheusにはDoS攻撃(Denial-of-Service Attack)の被害を軽減するための対策が組み込まれています。

## 16.7 Prometheus Alertmanager SNMPウェブフック

SNMPトラップを使用してPrometheusアラートに関する通知を受け取りたい場合は、cephadmを使用してPrometheus Alertmanager SNMPウェブフックをインストールできます。そのためには、以下の内容を含むサービス仕様と配置仕様を記載するファイルを作成する必要があります。



### 注記

サービス仕様と配置仕様を記載するファイルの詳細については、『導入ガイド』、第5章「cephadmによる展開」、5.4.2項「サービス仕様と配置仕様」を参照してください。

```
service_type: container
service_id: prometheus-webhook-snmp
placement:
  ADD_PLACEMENT_HERE
image: registry.suse.com/ses/7/prometheus-webhook-snmp:latest
args:
  - "--publish 9099:9099"
envs:
  - ARGS="--debug --snmp-host=ADD_HOST_GATEWAY_HERE"
  - RUN_ARGS="--metrics"
EOF
```

サービスをデフォルト設定で動作させるには、上記のサービス仕様を使用してください。Prometheus受信者がリスンするポートを公開する必要があります。そのためには、サービスの実行中にコマンドライン引数`--publish HOST_PORT:CONTAINER_PORT`を使用します。これは、コンテナがポートを自動的に公開しないためです。仕様に以下の行を追加する方法でも、ポートを公開できます。

```
args:
  - "--publish 9099:9099"
```

あるいは、コマンドライン引数 `--network=host` を使用して、ホストネットワークにコンテナを接続する方法もあります。

```
args:
  - "--network=host"
```

SNMPトラップ受信者がコンテナと同じホストにインストールされていない場合、SNMPホストのFQDNも指定する必要があります。container/hostの外部からSNMPトラップを受信できるように、コンテナのネットワークゲートウェイを使用してください。

```
envs:
  - ARGS="--debug --snmp-host=CONTAINER_GATEWAY"
```

## 16.7.1 prometheus-webhook-snmpサービスの設定

コンテナは環境変数または設定ファイルを用いて設定できます。

環境変数を使用する場合は、`ARGS`によりグローバルオプションを設定し、`RUN_ARGS`により`run`コマンドのオプションを指定します。サービス仕様は以下のように編集する必要があります。

```
envs:
  - ARGS="--debug --snmp-host=CONTAINER_GATEWAY"
  - RUN_ARGS="--metrics --port=9101"
```

設定ファイルを使用する場合、サービス仕様を以下のように編集する必要があります。

```
files:
  etc/prometheus-webhook-snmp.conf:
    - "debug: True"
    - "snmp_host: ADD_HOST_GATEWAY_HERE"
    - "metrics: True"
volume_mounts:
  etc/prometheus-webhook-snmp.conf: /etc/prometheus-webhook-snmp.conf
```

展開には次のコマンドを実行します。

```
cephuser@adm > ceph orch apply -i SERVICE_SPEC_FILE
```

詳細については、『導入ガイド』、第5章「cephadmによる展開」、5.4.3項「Cephサービスの展開」を参照してください。

## 16.7.2 Prometheus AlertmanagerのSNMP用設定

最後に、Prometheus AlertmanagerをSNMPトラップ用に設定する必要があります。このサービスをまだ展開していない場合は、サービス仕様ファイルを作成してください。Prometheus Alertmanager SNMPウェブフックがインストールされているホストのIPアドレスかFQDNでIP\_OR\_FQDNを置き換える必要があります。以下に例を示します。



### 注記

すでにこのサービスを展開している場合は、AlertmanagerがSNMP用に正しくセットアップされていることを確認し、以下の設定で再展開してください。

```
service_type: alertmanager
placement:
  hosts:
    - HOSTNAME
webhook_configs:
  - 'http://IP_OR_FQDN:9099/'
```

次のコマンドを実行して、サービス仕様を適用します。

```
cephuser@adm > ceph orch apply -i SERVICE_SPEC_FILE
```

### III クラスタへのデータ保存

- 17 保存データの管理 152
- 18 ストレージプールの管理 182
- 19 イレージャコーディングプール 202
- 20 RADOS Block Device 209

## 17 保存データの管理

CRUSHアルゴリズムは、データの保存場所を計算することによって、データの保存と取得の方法を決定します。CRUSHにより、Cephクライアントは、中央サーバやブローカ経由ではなく直接OSDと通信できるようになります。アルゴリズムで決定されるデータの保存と取得の方法を使用することで、Cephは、SPOF (single point of failure)、パフォーマンスのボトルネック、およびスケーラビリティの物理的な制限を解消します。

CRUSHはクラスタのマップを必要とし、そのCRUSHマップを使用して、クラスタ全体に均等に分散したデータを擬似ランダムにOSDに保存および取得します。

CRUSHマップには、OSDのリスト、デバイスを物理的な場所に集約するための「バケット」のリスト、およびCephクラスタのプール内でデータをどのように複製するかをCRUSHに指示するルールが含まれます。インストールの基礎になっている物理的な組織を反映することで、CRUSHは、相関するデバイス障害の潜在的な原因をモデル化し、これによってその原因に対応できます。原因としては、物理的な距離の近さ、共有電源、共有ネットワークなどが代表的です。この情報をクラスタマップにエンコードすることにより、CRUSHの配置ポリシーは、オブジェクトのレプリカを異なる障害ドメインに分離しながら、必要な分散を維持できます。たとえば、発生する可能性がある同時障害に対応するため、データレプリカを、異なるシェルフ、ラック、電源、コントローラ、または物理的な場所を使用するデバイスに配置することが望ましい場合があります。

Cephクラスタの展開後、デフォルトのCRUSHマップが生成されます。Cephサンドボックス環境にはこれで十分です。ただし、大規模なデータクラスタを展開する場合は、カスタムCRUSHマップの作成を積極的に検討する必要があります。カスタムCRUSHマップは、Cephクラスタの管理、パフォーマンスの向上、およびデータの安全性の確保に役立つためです。

たとえば、OSDがダウンしてオンサイトでのサポートやハードウェアの交換が必要になった場合、CRUSHマップがあれば、ホストの物理的なデータセンター、ルーム、列、およびラックの場所を容易に特定できます。

同様に、障害の特定の迅速化にも役立つことがあります。たとえば、特定のラックのOSDすべてが同時にダウンした場合、OSD自体の障害ではなく、ネットワークスイッチ、あるいはラックまたはネットワークスイッチの電源の障害であることがあります。

カスタムCRUSHマップは、障害が発生したホストに関連付けられた配置グループ(17.4項「[配置グループ](#)」を参照)が機能低下状態になった場合に、Cephによってデータの冗長コピーが保存される物理的な場所を特定するのにも役立ちます。

CRUSHマップには主なセクションが3つあります。

- **OSDデバイス**は、`ceph-osd`デーモンに対応するオブジェクトストレージデバイスで構成されます。
- **バケット**は、ストレージの場所の階層的な集約構造(列、ラック、ホストなど)と、それらに割り当てられている重みで構成されます。
- **ルールセット**は、バケットの選択方法で構成されます。

## 17.1 OSDデバイス

配置グループをOSDにマップするため、CRUSHマップにはOSDデバイスのリスト(OSDデーモンの名前)が必要です。デバイスのリストはCRUSHマップの先頭に記述されます。

```
#devices
device NUM osd.OSD_NAME class CLASS_NAME
```

以下に例を示します。

```
#devices
device 0 osd.0 class hdd
device 1 osd.1 class ssd
device 2 osd.2 class nvme
device 3 osd.3class ssd
```

一般的な規則として、OSDデーモンは1つのディスクにマップされます。

### 17.1.1 デバイスクラス

CRUSHマップによってデータ配置を柔軟に制御できる点は、Cephの強みの1つです。これは、クラスタの管理において最も困難な部分の1つでもあります。「デバイスクラス」を使用すると、これまで管理者が手動で行う必要があった、CRUSHマップに対して特によく行われる変更を自動化できます。

#### 17.1.1.1 CRUSHの管理の問題

多くの場合、Cephクラスタは、HDD、SSD、NVMeなど複数のタイプのストレージデバイスで構築し、これらの種類を混在させる場合もあります。ここでは、このようなさまざまなタイプのストレージデバイスを「デバイスクラス」と呼びます。これは、CRUSHバケットの「タイプ」プロパティを混同しないようにするためです(たとえば、ホスト、ラック、列など。詳細については[17.2項「バケット」](#)を参照してください)。SSDでサポートされているCeph OSDは、回転型のディスクでサポートされているものよりはるかに高速であるため、特定のワーク

ロードに適しています。Cephを使用すると、異なるデータセットやワークロード用のRADOSプールを簡単に作成したり、これらのプールのデータ配置を制御するために異なるCRUSHルールを簡単に割り当てたりすることができます。

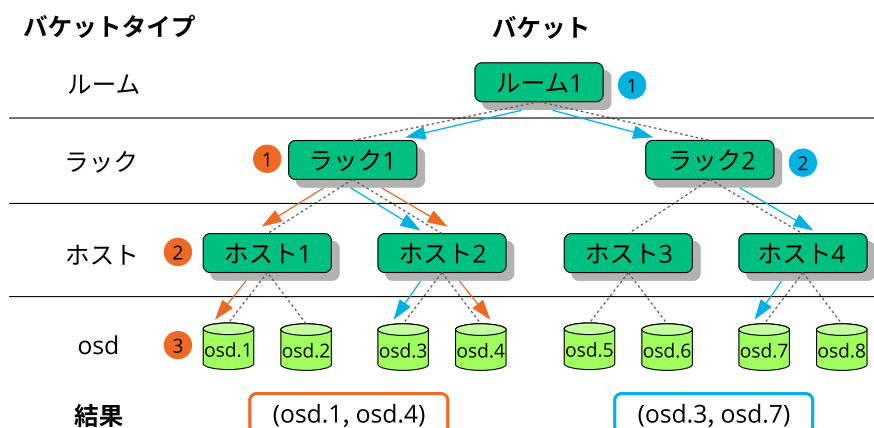


図 17.1: 複数のデバイスクラスが混在するOSD

ただし、データを特定のデバイスクラスにのみ配置するようにCRUSHルールを設定するのは面倒です。ルールはCRUSH階層の点から見ると有効ですが、(上記のサンプル階層のように)複数のデバイスが同じホストやラックに混在する場合、これらのデバイスは混在して階層の同じサブツリーに表示されます(デフォルト)。以前のバージョンのSUSE Enterprise Storageでは、これらを手動で別個のツリーに分離するには、デバイスクラスごとに1つずつ、複数のバージョンの中間ノードを作成する必要がありました。

### 17.1.1.2 デバイスクラス

Cephが提供する優れた解決策は、各OSDに「デバイスクラス」というプロパティを追加することです。デフォルトで、OSDは、Linuxカーネルによって公開されるハードウェアプロパティに基づいて、そのデバイスクラスを「hdd」、「ssd」、または「nvme」のいずれかに自動的に設定します。これらのデバイスクラスは`ceph osd tree`コマンド出力の新しい列でレポートされます。

```
cephuser@adm > ceph osd tree
ID CLASS WEIGHT  TYPE NAME        STATUS REWEIGHT PRI-AFF
-1          83.17899 root default
-4          23.86200 host cpach
2  hdd  1.81898    osd.2      up  1.00000 1.00000
3  hdd  1.81898    osd.3      up  1.00000 1.00000
4  hdd  1.81898    osd.4      up  1.00000 1.00000
5  hdd  1.81898    osd.5      up  1.00000 1.00000
6  hdd  1.81898    osd.6      up  1.00000 1.00000
7  hdd  1.81898    osd.7      up  1.00000 1.00000
8  hdd  1.81898    osd.8      up  1.00000 1.00000
```

15	hdd	1.81898	osd.15	up	1.00000	1.00000
10	nvme	0.93100	osd.10	up	1.00000	1.00000
0	ssd	0.93100	osd.0	up	1.00000	1.00000
9	ssd	0.93100	osd.9	up	1.00000	1.00000

たとえば、デバイスドライバがデバイスに関する情報を `/sys/block` を介して適切に公開していないためにデバイスクラスの自動検出に失敗する場合、コマンドラインからデバイスクラスを調整できます。

```
cephuser@adm > ceph osd crush rm-device-class osd.2 osd.3
done removing class of osd(s): 2,3
cephuser@adm > ceph osd crush set-device-class ssd osd.2 osd.3
set osd(s) 2,3 to class 'ssd'
```

### 17.1.1.3 CRUSH配置ルールの設定

CRUSHルールにより、特定のデバイスクラスへの配置を制限できます。たとえば、次のコマンドを実行して、SSD上にのみデータを分散する高速な「複製」プールを作成できます。

```
cephuser@adm > ceph osd crush rule create-
replicated RULE_NAME ROOT FAILURE_DOMAIN_TYPE DEVICE_CLASS
```

以下に例を示します。

```
cephuser@adm > ceph osd crush rule create-replicated fast default host ssd
```

「fast\_pool」というプールを作成し、それを「fast」ルールに割り当てます。

```
cephuser@adm > ceph osd pool create fast_pool 128 128 replicated fast
```

「イレージャコード」ルールを作成するプロセスはわずかに異なります。まず、目的のデバイスクラスのプロパティを含むイレージャコードプロファイルを作成します。その後、イレージャコーディングプールを作成する際にそのプロファイルを使用します。

```
cephuser@adm > ceph osd erasure-code-profile set myprofile \
k=4 m=2 crush-device-class=ssd crush-failure-domain=host
cephuser@adm > ceph osd pool create mypool 64 erasure myprofile
```

CRUSHマップを手動で編集してルールをカスタマイズする必要がある場合に備えて、デバイスクラスを指定できるように構文が拡張されています。たとえば、上のコマンドによって生成されたCRUSHルールは次のようになります。

```
rule ecpool {
  id 2
  type erasure
  min_size 3
  max_size 6
  step set_chooseleaf_tries 5
```

```
step set_choose_tries 100
step take default class ssd
step chooseleaf indep 0 type host
step emit
}
```

ここでの重要な違いは、「take」コマンドに追加のサフィックス「class CLASS\_NAME」が含まれている点です。

#### 17.1.1.4 追加のコマンド

CRUSHマップで使用されるデバイスクラスを一覧にするには、次のコマンドを実行します。

```
cephuser@adm > ceph osd crush class ls
[
  "hdd",
  "ssd"
]
```

既存のCRUSHルールを一覧にするには、次のコマンドを実行します。

```
cephuser@adm > ceph osd crush rule ls
replicated_rule
fast
```

「fast」という名前のCRUSHルールの詳細を表示するには、次のコマンドを実行します。

```
cephuser@adm > ceph osd crush rule dump fast
{
  "rule_id": 1,
  "rule_name": "fast",
  "ruleset": 1,
  "type": 1,
  "min_size": 1,
  "max_size": 10,
  "steps": [
    {
      "op": "take",
      "item": -21,
      "item_name": "default~ssd"
    },
    {
      "op": "chooseleaf_firstn",
      "num": 0,
      "type": "host"
    },
    {
      "op": "emit"
    }
  ]
}
```

```
]
}
```

「ssd」クラスに属するOSDを一覧にするには、次のコマンドを実行します。

```
cephuser@adm > ceph osd crush class ls-osd ssd
0
1
```

### 17.1.1.5 古いSSDルールからデバイスクラスへの移行

バージョン5より前のSUSE Enterprise Storageでは、SSDなどのデバイスに適用されるルールを作成するには、CRUSHマップを手動で編集し、特殊なデバイスタイプ(SSDなど)それぞれに対して並列階層を維持する必要がありました。SUSE Enterprise Storage 5から、この処理は、デバイスクラス機能によって透過的に有効になりました。

**crushtool** コマンドを使用して、古いルールと階層を新しいクラスベースのルールに変換できます。次のように複数のタイプの変換が可能です。

**crushtool --reclassify-root** ROOT\_NAME DEVICE\_CLASS

このコマンドは、以下を使用して、ROOT\_NAMEの下階層にあるものをすべて取得し、そのルートを参照するルールをすべて調整します。

```
take ROOT_NAME
```

これを代わりに以下に調整します。

```
take ROOT_NAME class DEVICE_CLASS
```

さらに、指定したクラスの「シャドウツリー」に古いIDを使用するよう、バケットの番号を再割り当てします。そのため、データの移動は発生しません。

#### 例 17.1: **crushtool --reclassify-root**

次のような既存のルールについて考えてみてください。

```
rule replicated_ruleset {
  id 0
  type replicated
  min_size 1
  max_size 10
  step take default
  step chooseleaf firstn 0 type rack
  step emit
}
```

ルート「default」をクラス「hdd」として再分類した場合、このルールは次のようになります。

```
rule replicated_ruleset {
    id 0
    type replicated
    min_size 1
    max_size 10
    step take default class hdd
    step chooseleaf firstn 0 type rack
    step emit
}
```

### **crushtool --set-subtree-class** BUCKET\_NAME DEVICE\_CLASS

この方法は、BUCKET\_NAMEをルートとするサブツリー内にあるすべてのデバイスに、指定したデバイスクラスのマークを付けます。

--set-subtree-classは通常、--reclassify-rootオプションと組み合わせて使用し、そのルートにあるすべてのデバイスに正しいクラスのラベルが付けられるようにします。ただし、これらのデバイスによっては意図的に異なるクラスを使用しているものがあるため、再度ラベルを付けたくない場合があります。このような場合は、--set-subtree-classオプションを除外してください。このような再マッピングは完全ではないことに注意してください。以前のルールは複数のクラスのデバイスにわたって分散されますが、調整されたルールは指定したデバイスクラスのデバイスにのみマップされるためです。

### **crushtool --reclassify-bucket** MATCH\_PATTERN DEVICE\_CLASS DEFAULT\_PATTERN

この方法では、並列タイプに固有の階層を通常の階層にマージできます。たとえば、多くのユーザが次のようなCRUSHマップを使用しています。

#### 例 17.2: **crushtool --reclassify-bucket**

```
host node1 {
    id -2          # do not change unnecessarily
    # weight 109.152
    alg straw
    hash 0 # rjenkins1
    item osd.0 weight 9.096
    item osd.1 weight 9.096
    item osd.2 weight 9.096
    item osd.3 weight 9.096
    item osd.4 weight 9.096
    item osd.5 weight 9.096
    [...]
}

host node1-ssd {
    id -10         # do not change unnecessarily
    # weight 2.000
```

```

    alg straw
    hash 0 # rjenkins1
    item osd.80 weight 2.000
    [...]
}

root default {
    id -1          # do not change unnecessarily
    alg straw
    hash 0 # rjenkins1
    item node1 weight 110.967
    [...]
}

root ssd {
    id -18         # do not change unnecessarily
    # weight 16.000
    alg straw
    hash 0 # rjenkins1
    item node1-ssd weight 2.000
    [...]
}

```

この機能は、指定したパターンに一致する各バケットを再分類します。パターンは`%suffix`または`prefix%`のようになります。上の例では、パターン`%-ssd`を使用します。一致した各バケットに対し、ワイルドカード「%」に一致する、名前の残りの部分にベースバケットが指定されます。一致したバケットのすべてのデバイスに指定したデバイスクラスのラベルが付けられ、デバイスがベースバケットに移動されます。ベースバケットが存在しない場合(たとえば、「node12-ssd」は存在するものの「node12」は存在しない場合)、指定したデフォルトの親バケットの下にベースバケットが作成されてリンクされます。古いバケットIDは新しいシャドウバケット用に保持されるため、データの移動は行われません。古いバケットを参照する`take`ステップが含まれるルールが調整されます。

**crushtool --reclassify-bucket** `BUCKET_NAME` `DEVICE_CLASS` `BASE_BUCKET`

`--reclassify-bucket` オプションをワイルドカードなしでを使用して、単一のバケットをマップできます。たとえば、前の例では、「ssd」バケットをデフォルトのバケットにマッピングしたいと考えています。

上のフラグメントで構成されるマップを変換する最後のコマンドは、次のとおりです。

```

cephuser@adm > ceph osd getcrushmap -o original
cephuser@adm > crushtool -i original --reclassify \
  --set-subtree-class default hdd \
  --reclassify-root default hdd \
  --reclassify-bucket %-ssd ssd default \
  --reclassify-bucket ssd ssd default \

```

```
-o adjusted
```

正しく変換されたことを確認するために、`--compare`オプションがあります。このオプションは、CRUSHマップへの大量の入力サンプルをテストし、同じ結果が返されるかどうかを比較するものです。これらの入力、`--test`に適用されるオプションと同じオプションで制御します。上の例では、コマンドは次のようになります。

```
cephuser@adm > crushtool -i original --compare adjusted
rule 0 had 0/10240 mismatched mappings (0)
rule 1 had 0/10240 mismatched mappings (0)
maps appear equivalent
```



## ヒント

違いがあった場合は、再マップされる入力の比率がカッコ内に表示されます。

調整されたCRUSHマップに問題がなければ、マップをクラスタに適用できます。

```
cephuser@adm > ceph osd setcrushmap -i adjusted
```

### 17.1.1.6 詳細の参照先

CRUSHマップの詳細については、[17.5項「CRUSHマップの操作」](#)を参照してください。

Cephプールの一般的な詳細については、[第18章「ストレージプールの管理」](#)を参照してください。

イレージャコーディングプールの詳細については、[第19章「イレージャコーディングプール」](#)を参照してください。

## 17.2 バケット

CRUSHマップにはOSDのリストが含まれており、これをツリー構造のバケット配置に編成してデバイスを物理的な場所に集約できます。個々のOSDはツリーの葉にあたります。

0	osd	特定のデバイスまたはOSD( <a href="#">osd.1</a> 、 <a href="#">osd.2</a> 、など)。
1	ホスト	1つ以上のOSDを含むホストの名前。
2	シャーシ	ラック内のどのシャーシに <a href="#">ホスト</a> が存在するかを識別するID。

3	ラック	コンピュータラック。デフォルトはunknownrackです。
4	列	一連のラックの列。
5	pdu	「Power Distribution Unit」(配電ユニット)の略語。
6	ポッド	「Point of Delivery」の略語。ここでは、ひとかたまりのPDUやラック列を指します。
7	ルーム	ラック列が設置されている部屋。
8	データセンター	1つ以上のルームを含む、物理的なデータセンター。
9	地域	世界の地理的地域(たとえば、NAM、LAM、EMEA、APACなど)。
10	root	OSDバケットツリーのルートノード(通常は、defaultに設定されます)。



## ヒント

既存のタイプを変更して独自のバケットタイプを作成できます。

Cephの展開ツールは、各ホストのバケットと「default」という名前のrootが含まれるCRUSHマップを生成します。これは、デフォルトのrbdプールで役立ちます。残りのバケットタイプは、ノード/バケットの物理的な場所の情報を保存するための手段を提供します。OSD、ホスト、またはネットワークハードウェアが正常に機能しておらず、管理者が物理的なハードウェアにアクセスする必要がある場合、これによってクラスタ管理が大幅に容易になります。

バケットには、タイプ、固有の名前(文字列)、負の整数で表される固有のID、項目の合計容量/機能を基準にした相対的な重み、バケットアルゴリズム(デフォルトはstraw2)、およびハッシュ(デフォルトは0で、CRUSHハッシュrjenkins1を反映)が含まれます。1つのバケットには1つ以上の項目を含めることができます。項目は他のバケットやOSDで構成できます。また、項目には、その項目の相対的な重みを反映した重みを設定できます。

```
[bucket-type] [bucket-name] {
  id [a unique negative numeric ID]
  weight [the relative capacity/capability of the item(s)]
  alg [the bucket type: uniform | list | tree | straw2 | straw ]
  hash [the hash type: 0 by default]
  item [item-name] weight [weight]
```

```
}
```

次の例は、バケットを使用して、プールと、データセンター、ルーム、ラック、列などの物理的な場所をどのように集約できるかを示しています。

```
host ceph-osd-server-1 {
    id -17
    alg straw2
    hash 0
    item osd.0 weight 0.546
    item osd.1 weight 0.546
}

row rack-1-row-1 {
    id -16
    alg straw2
    hash 0
    item ceph-osd-server-1 weight 2.00
}

rack rack-3 {
    id -15
    alg straw2
    hash 0
    item rack-3-row-1 weight 2.00
    item rack-3-row-2 weight 2.00
    item rack-3-row-3 weight 2.00
    item rack-3-row-4 weight 2.00
    item rack-3-row-5 weight 2.00
}

rack rack-2 {
    id -14
    alg straw2
    hash 0
    item rack-2-row-1 weight 2.00
    item rack-2-row-2 weight 2.00
    item rack-2-row-3 weight 2.00
    item rack-2-row-4 weight 2.00
    item rack-2-row-5 weight 2.00
}

rack rack-1 {
    id -13
    alg straw2
    hash 0
    item rack-1-row-1 weight 2.00
    item rack-1-row-2 weight 2.00
    item rack-1-row-3 weight 2.00
    item rack-1-row-4 weight 2.00
    item rack-1-row-5 weight 2.00
}
```

```

}

room server-room-1 {
    id -12
    alg straw2
    hash 0
    item rack-1 weight 10.00
    item rack-2 weight 10.00
    item rack-3 weight 10.00
}

datacenter dc-1 {
    id -11
    alg straw2
    hash 0
    item server-room-1 weight 30.00
    item server-room-2 weight 30.00
}

root data {
    id -10
    alg straw2
    hash 0
    item dc-1 weight 60.00
    item dc-2 weight 60.00
}

```

## 17.3 ルールセット

CRUSHマップは、プールのデータ配置を決定するルールである「CRUSHルール」の概念をサポートしています。大規模クラスタでは、ほとんどの場合、プールを大量に作成し、各プールが専用のCRUSHルールセットとルールを持つようにします。デフォルトのCRUSHマップには、デフォルトのルート用のルールがあります。ルートやルールがさらに必要な場合は、後で作成する必要があります。作成しない場合、新しいプールを作成するときに自動的に作成されます。



### 注記

ほとんどの場合、デフォルトのルールを変更する必要はありません。新しいプールを作成する場合、そのデフォルトのルールセットは0です。

ルールは次の形式を取ります。

```
rule rulename {
```

```

ruleset ruleset
type type
min_size min-size
max_size max-size
step step
}

```

## ruleset

整数。ルールを、ルールのセットに属しているものとして分類します。プールでルールセットを設定することによって有効にします。このオプションは必須です。デフォルトは0です。

## type

文字列。「複製」プールまたは「イレージャ」コーディングプールのいずれかのルールを記述します。このオプションは必須です。デフォルトはreplicatedです。

## min\_size

整数。プールグループが作成するレプリカがこの数より少ない場合、CRUSHはこのルールを選択しません。このオプションは必須です。デフォルトは2です。

## max\_size

整数。プールグループが作成するレプリカがこの数より多い場合、CRUSHはこのルールを選択しません。このオプションは必須です。デフォルトは10です。

## step take bucket

名前で指定したバケットを取ります。ツリーの下方へ反復処理を開始します。このオプションは必須です。ツリー全体の反復処理の詳細については、[17.3.1項「ノードツリーの反復処理」](#)を参照してください。

## step targetmodenum type bucket-type

targetはchooseまたはchooseleafのいずれかにできます。chooseに設定すると、大量のバケットが選択されます。chooseleafは、バケットセット内の各バケットのサブツリーからOSD (リーフノード)を直接選択します。

modeはfirstnまたはindepのいずれかにできます。[17.3.2項「firstnとindep」](#)を参照してください。

特定のタイプのバケットの数を指定します。Nを利用可能なオプションの数とすると、num > 0 && < Nの場合、それと同じ数のバケットを選択します。num < 0の場合、N - numを意味します。num == 0の場合、N個のバケット(利用可能なものすべて)を選択します。step takeまたはstep chooseに従います。

## step emit

現在の値を出力してスタックを空にします。一般的にはルールの中で使用しますが、同じルール内の別のツリーを形成する場合にも使用できます。step chooseに従います。

### 17.3.1 ノードツリーの反復処理

バケットで定義された構造はノードツリーと見なすことができます。このツリーのバケットがノードで、OSDがリーフに当たります。

CRUSHマップのルールは、このツリーからどのような方法でOSDを選択するかを定義します。ルールは特定のノードから処理を始めて、ツリーの方へと反復処理を行い、OSDのセットを返します。どのブランチを選択する必要があるかを定義することはできません。その代わりに、CRUSHアルゴリズムにより、OSDのセットがレプリケーション要件を満足し、データを均等に分散するよう保証されます。

step take bucketを使用すると、ノードツリー全体の反復処理は、指定したバケット(バケットタイプではありません)から始まります。ツリー内のすべてのブランチのOSDを返す場合は、指定したバケットがルートバケットである必要があります。そうしないと、以降の手順はサブツリーでのみ反復処理されます。

ルール定義のstep takeの後にはstep chooseエントリが1つ以上続きます。それぞれのstep chooseは、直前に選択されていた上位ノードから、定義された数のノード(またはブランチ)を選択します。

最後に、step emitで、選択されたOSDが返されます。

step chooseleafは、指定したバケットのブランチからOSDを直接選択する便利な機能です。

図17.2「ツリーの例」に、stepを使用してツリー全体で反復処理を行う例を示します。次のルール定義では、オレンジ色の矢印と番号はexample1aとexample1bに対応し、青色はexample2に対応します。

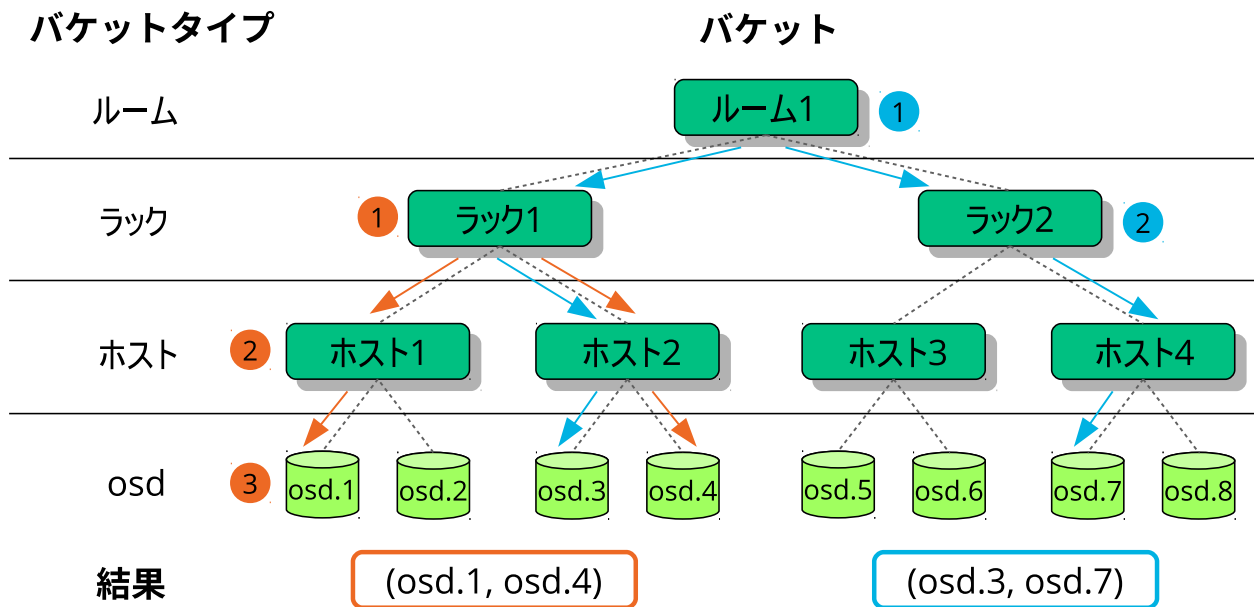


図 17.2: ツリーの例

```

# orange arrows
rule example1a {
    ruleset 0
    type replicated
    min_size 2
    max_size 10
    # orange (1)
    step take rack1
    # orange (2)
    step choose firstn 0 host
    # orange (3)
    step choose firstn 1 osd
    step emit
}

rule example1b {
    ruleset 0
    type replicated
    min_size 2
    max_size 10
    # orange (1)
    step take rack1
    # orange (2) + (3)
    step chooseleaf firstn 0 host
    step emit
}

# blue arrows
rule example2 {

```

```
ruleset 0
type replicated
min_size 2
max_size 10
# blue (1)
step take room1
# blue (2)
step chooseleaf firstn 0 rack
step emit
}
```

### 17.3.2 firstnとindep

CRUSHルールは、障害ノードまたはOSDの置換を定義します(17.3項「ルールセット」を参照してください)。キーワード`step`には、パラメータとして`firstn`または`indep`が必要です。図 17.3「ノードの置換方法」に例を示します。

`firstn`は、アクティブノードのリストの最後に置換ノードを追加します。障害ノードの場合、以降の正常なノードが左側に移動されて、障害ノードの隙間を埋めます。これは、「複製プール」に対するデフォルトかつ適切な方法です。2つ目のノードにはすでにすべてのデータがあるため、プライマリノードの権限をただちに引き継ぐことができます。

`indep`は、各アクティブノードに対して修復済みの置換ノードを選択します。障害ノードを置換する際に、残りのノードの順序は変更されません。これは「イレージャコーディングプール」にとって適切です。イレージャコーディングプールでは、ノードに保存されるデータは、ノード選択時の位置によって異なります。ノードの順序が変更されると、影響を受けるノードのすべてのデータを再配置しなければなりません。

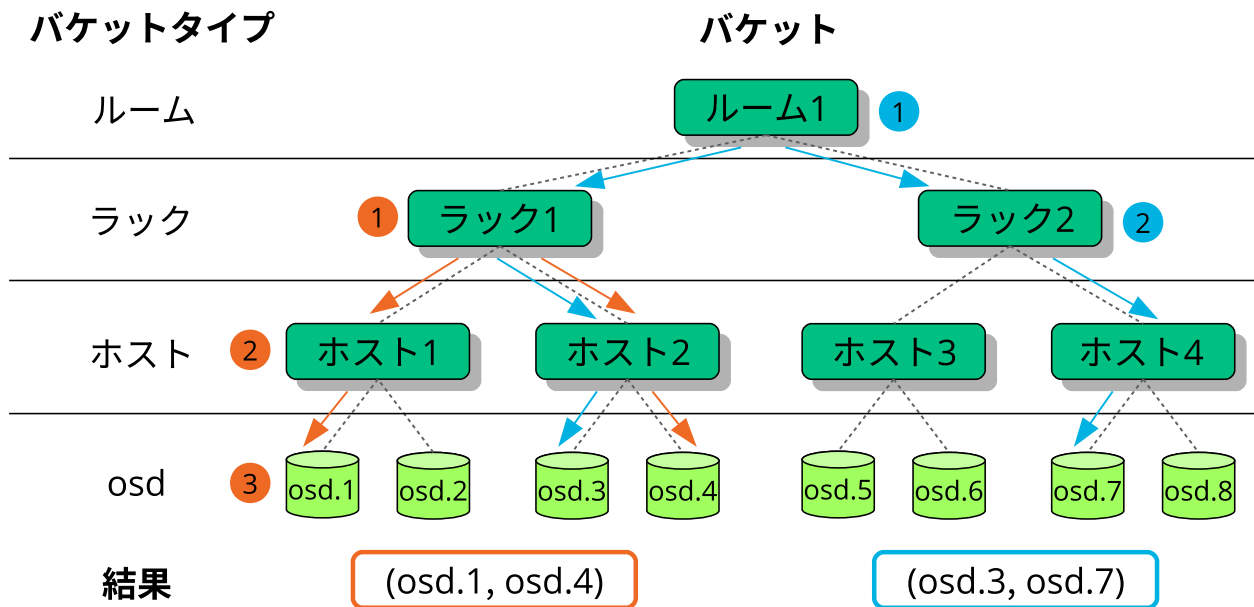


図 17.3: ノードの置換方法

## 17.4 配置グループ

Cephは、オブジェクトをPG (配置グループ)にマップします。配置グループは、オブジェクトをグループとしてOSDに配置する、論理オブジェクトプールのシャードまたはフラグメントです。配置グループにより、CephがOSDにデータを保存する際のオブジェクトごとのメタデータの量が削減されます。配置グループの数が多いほど(たとえば、OSDあたり100など)、バランスが向上します。

### 17.4.1 配置グループの使用

PG (配置グループ)は複数のオブジェクトを1つのプール内に集約します。この主な理由は、オブジェクトごとにオブジェクトの配置とメタデータを追跡すると、計算コストが高くなるためです。たとえば、何百万ものオブジェクトが存在するシステムでは、その各オブジェクトの配置を直接追跡することはできません。

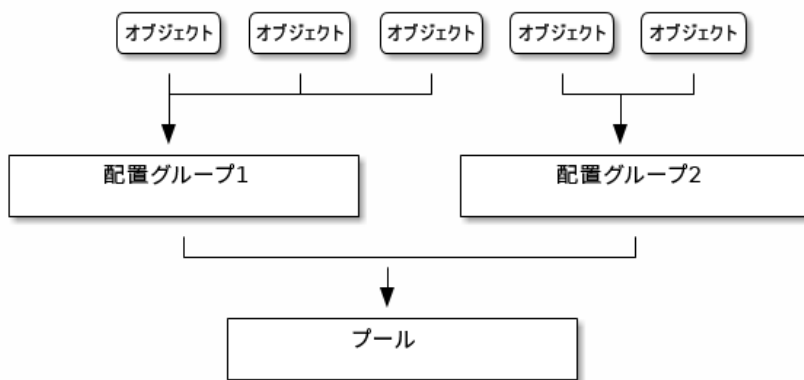


図 17.4: プール内の配置グループ

Cephクライアントは、オブジェクトが属する配置グループを計算します。このために、オブジェクトIDをハッシュし、定義されたプール内のPGの数と、プールのIDに基づいて操作を適用します。

配置グループ内のオブジェクトのコンテンツは、一連のOSDに保存されます。たとえば、サイズが2の複製プールでは、各配置グループは次のように2つのOSDにオブジェクトを保存します。

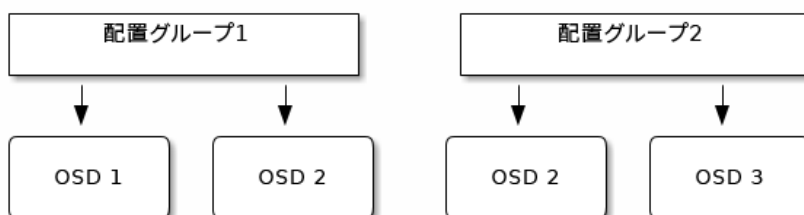


図 17.5: 配置グループとOSD

OSD #2に障害が発生した場合、別のOSDが配置グループ#1に割り当てられ、OSD #1内にあるすべてのオブジェクトのコピーで埋められます。プールサイズを2から3に変更すると、追加のOSDが配置グループに割り当てられ、配置グループ内にあるすべてのオブジェクトのコピーを受け取ります。

配置グループはOSDを所有するのではなく、同じプール(他のプールの場合もあり)の他の配置グループとOSDを共有するものです。OSD #2に障害が発生した場合、配置グループ#2は、OSD #3を使用してオブジェクトのコピーを復元する必要があります。

配置グループの数が増えると、新しい配置グループにOSDが割り当てられます。CRUSH機能の結果も変化し、以前の配置グループの一部のオブジェクトは新しい配置グループにコピーされ、古い配置グループから削除されます。

## 17.4.2 PG\_NUMの値の決定



### 注記

Ceph Nautilus (v14.x)以降はCeph Managerのpg\_autoscalerモジュールを使用することで、必要に応じて配置グループを自動拡張できます。この機能を有効にしたい場合は、『Deploying and Administering SUSE Enterprise Storage with Rook』、第6章「Configuration」、6.1.1.1項「Default PG and PGP counts」を参照してください。

新しいプールを作成するときに、PG\_NUMの値を従来通り手動で選択することも可能です。

```
root # ceph osd pool create POOL_NAME PG_NUM
```

PG\_NUMを自動的に計算することはできません。次に、クラスタ内のOSDの数に応じた、一般的に使用される値をいくつか示します。

#### OSDが5未満の場合:

PG\_NUMを128に設定します。

#### OSDが5～10の場合:

PG\_NUMを512に設定します。

#### OSDが10～50の場合:

PG\_NUMを1024に設定します。

OSDの数が増えるにつれて、PG\_NUMの適切な値を選択することがより重要になってきます。PG\_NUMは、OSDに障害が発生した場合のクラスタの動作とデータの耐久性に強く影響します。

### 17.4.2.1 OSDが50を超える場合の配置グループの計算

OSDが50未満の場合は、17.4.2項「PG\_NUMの値の決定」で説明されている事前選択値を使用してください。OSDが50を超える場合は、リソース使用量、データ耐久性、および分散のバランスが取れるよう、OSDあたり約50～100の配置グループを推奨します。単一プールのオブジェクトの場合、次の式を使用してベースラインを取得できます。

```
total PGs = (OSDs * 100) / POOL_SIZE
```

ここで、POOL\_SIZEは複製プールのレプリカ数か、ceph osd erasure-code-profile getコマンドによって返されるイレージャコーディングプールの「k」+「m」の合計です。結果は、最も近い2の累乗値まで切り上げる必要があります。CRUSHアルゴリズムが配置グループ間でオブジェクト数をバランスよく均等に配置できるよう、切り上げを推奨します。

たとえば、OSDの数が200で、プールサイズが3つのレプリカであるクラスタの場合、次のようにPGの数を見積もります。

$$(200 * 100) / 3 = 6667$$

最も近い2の累乗値は「8192」です。

複数のデータプールを使用してオブジェクトを保存する場合、プールあたりの配置グループの数と、OSDあたりの配置グループの数のバランスを取る必要があります。システムリソースの過剰な使用やピアリングプロセスの大幅な低速化を招くことなく、OSDごとの違いが適度に小さくなるような妥当な配置グループ合計数を算出する必要があります。

たとえば、10個のプールで構成されるクラスタがあり、各プールについて10個のOSDに512個の配置グループが存在する場合、合計5,120個の配置グループが10個のOSDに分散されます。つまり、OSDあたり512個の配置グループになります。このようなセットアップでは、リソースを使用し過ぎることはありません。ただし、それぞれに512個の配置グループが含まれるプールを1,000個作成した場合、OSDはそれぞれ約50,000個の配置グループを処理することになり、ピアリングに必要なリソースと時間が大幅に増えます。

### 17.4.3 配置グループ数の設定



#### 注記

Ceph Nautilus (v14.x)以降はCeph Managerのpg\_autoscalerモジュールを使用することで、必要に応じて配置グループを自動拡張できます。この機能を有効にしたい場合は、『Deploying and Administering SUSE Enterprise Storage with Rook』、第6章「Configuration」、6.1.1.1項「Default PG and PGP counts」を参照してください。

従来通りプールの配置グループ数を手動で指定する必要がある場合は、プールの作成時に指定する必要があります(18.1項「プールの作成」を参照してください)。プールに配置グループを設定した後であれば、次のコマンドを実行して配置グループ数を増やすことができます。

```
root # ceph osd pool set POOL_NAME pg_num PG_NUM
```

配置グループの数を増やした後、クラスタの再バランスを行う前に、配置対象の配置グループの数(PG\_NUM)を増やす必要もあります。PG\_NUMは、配置の際にCRUSHアルゴリズムによって考慮される配置グループの数です。PG\_NUMを増やすと配置グループが分割されますが、データはPG\_NUMを増やすまで新しい配置グループに移行されません。PG\_NUMはPG\_NUMと等しくなければなりません。配置対象の配置グループの数を増やすには、次のコマンドを実行します。

```
root # ceph osd pool set POOL_NAME pgp_num PGP_NUM
```

#### 17.4.4 配置グループ数の確認

プールの配置グループ数を確認するには、次の`get`コマンドを実行します。

```
root # ceph osd pool get POOL_NAME pg_num
```

#### 17.4.5 クラスタの配置グループの統計情報の確認

クラスタの配置グループの統計情報を確認するには、次のコマンドを実行します。

```
root # ceph pg dump [--format FORMAT]
```

有効な形式は「plain」(デフォルト)と「json」です。

#### 17.4.6 スタックしている配置グループの統計情報の確認

指定した状態でスタックしているすべての配置グループの統計情報を確認するには、次のコマンドを実行します。

```
root # ceph pg dump_stuck STATE \
[--format FORMAT] [--threshold THRESHOLD]
```

`STATE`は、「inactive」(PGは最新のデータが含まれるOSDが起動するのを待機しているため、読み込みまたは書き込みを処理できない)、「unclean」(必要な回数複製されていないオブジェクトがPGに含まれている)、「stale」(PGは不明な状態で、それらのPGをホストしているOSDが`mon_osd_report_timeout`オプションで指定された時間間隔でモニタクラスタにレポートしていない)、「undersized」、または「degraded」のいずれかです。

有効な形式は「plain」(デフォルト)と「json」です。

このしきい値は、配置グループがスタックしてから、返される統計情報にそれを含めるまでの最小秒数を定義します(デフォルトでは300秒)。

#### 17.4.7 配置グループマップの検索

特定の配置グループの配置グループマップを検索するには、次のコマンドを実行します。

```
root # ceph pg map PG_ID
```

Cephは、配置グループマップ、配置グループ、およびOSDステータスを返します。

```
root # ceph pg map 1.6c
osdmap e13 pg 1.6c (1.6c) -> up [1,0] acting [1,0]
```

## 17.4.8 配置グループの統計情報の取得

特定の配置グループの統計情報を取得するには、次のコマンドを実行します。

```
root # ceph pg PG_ID query
```

## 17.4.9 配置グループのスクラブ

配置グループをスクラブ(17.6項「配置グループのスクラブ」)するには、次のコマンドを実行します。

```
root # ceph pg scrub PG_ID
```

Cephは、プライマリノードとレプリカノードを確認し、配置グループ内にあるすべてのオブジェクトのカタログを生成し、それらを比較して、欠落しているオブジェクトや一致しないオブジェクトがないかと、コンテンツに整合性があるかどうかを確認します。レプリカがすべて一致していれば、最後にセマンティックを一括処理して、スナップショットに関連するすべてのオブジェクトメタデータに整合性があることを確認します。エラーはログでレポートされます。

## 17.4.10 配置グループのバックフィルと回復の優先度の設定

複数の配置グループで回復やバックフィルが必要になった場合に、一部のグループに他のグループよりも重要なデータが格納されている状況が発生することがあります。たとえば、一部のPGには稼働中のマシンで使用されているイメージのデータが格納されていて、他のPGは非アクティブなマシンや関連性の低いデータで使用されている場合があります。このような場合、これらのグループの回復の優先度を設定して、該当するグループに保存されているデータのパフォーマンスと可用性を先に復元できます。バックフィルまたは回復中に特定の配置グループに優先のマークを付けるには、次のコマンドを実行します。

```
root # ceph pg force-recovery PG_ID1 [PG_ID2 ... ]
```

```
root # ceph pg force-backfill PG_ID1 [PG_ID2 ... ]
```

これにより、Cephは、他の配置グループより先に、まず指定した配置グループに対して回復またはバックフィルを実行します。これは、現在進行中のバックフィルや回復を中断するのではなく、指定したPGをできるだけ早く処理するものです。考えが変わった場合、または間違ったグループに優先度を設定した場合は、次のコマンドを使用して、優先度の設定をキャンセルします。

```
root # ceph pg cancel-force-recovery PG_ID1 [PG_ID2 ... ]
root # ceph pg cancel-force-backfill PG_ID1 [PG_ID2 ... ]
```

**cancel-\***コマンドを使用すると、PGから「force」フラグが削除され、PGはデフォルトの順序で処理されるようになります。このコマンドも、現在処理中の配置グループには影響せず、まだキューに入っている配置グループにのみ影響します。グループの回復またはバックフィルが完了すると、「force」フラグは自動的にクリアされます。

### 17.4.11 失われたオブジェクトを元に戻す

クラスタで1つ以上のオブジェクトが失われ、失われたデータの検索を中止する場合は、見つからないオブジェクトに「喪失」のマークを付ける必要があります。

可能性がある場所すべてに対してクエリを実行してもまだオブジェクトが失われた状態である場合、失われたオブジェクトを放棄しなければならないことがあります。これは、書き込みそのものが回復される前に実行された書き込みをクラスタが認識できるような複数の障害がまれな組み合わせで起きた場合に発生する可能性があります。

現在サポートされている唯一のオプションは「revert」で、以前のバージョンのオブジェクトにロールバックするか、新しいオブジェクトの場合はその情報を完全に消去します。「見つからない」オブジェクトに「喪失」のマークを付けるには、次のコマンドを実行します。

```
cephuser@adm > ceph pg PG_ID mark_unfound_lost revert|delete
```

### 17.4.12 配置グループの自動拡張の有効化

配置グループ(PG)とは、Cephのデータ分散方法を内部で実装している詳細部分です。配置グループの自動拡張を有効化することで、クラスタの使用方法に応じてクラスタが配置グループの作成や自動調整を行えるようにします。

システム上の各プールには`pg_autoscale_mode`というプロパティがあり、`off`、`on`、`warn`に設定することが可能です。

自動拡張はプールごとに設定します。また、次の3つのモードで動作します。

## off

このプールの自動拡張を無効化します。管理者の判断で、各プールに適切な数の配置グループを選択してください。

## on

対象プールで配置グループ数の自動調整を有効化します。

## warn

配置グループ数を調整する必要がある場合に、ヘルスアラートを発します。

既存のプールに自動拡張モードを設定するには、次のコマンドを実行します。

```
cephuser@adm > ceph osd pool set POOL_NAME pg_autoscale_mode mode
```

デフォルトの`pg_autoscale_mode`を設定することもできます。この設定は今後作成されるすべてのプールに適用されます。コマンドは次の通りです。

```
cephuser@adm > ceph config set global osd_pool_default_pg_autoscale_mode MODE
```

次のコマンドを実行することで、各プール、その相対的な使用率、推奨される配置グループ数の変更を確認できます。

```
cephuser@adm > ceph osd pool autoscale-status
```

## 17.5 CRUSHマップの操作

このセクションでは、CRUSHマップの編集やパラメータの変更、OSDの追加/移動/削除など、CRUSHマップの基本的な操作方法を紹介します。

### 17.5.1 CRUSHマップの編集

既存のCRUSHマップを編集するには、次の手順に従います。

1. CRUSHマップを取得します。クラスタのCRUSHマップを取得するには、次のコマンドを実行します。

```
cephuser@adm > ceph osd getcrushmap -o compiled-crushmap-filename
```

Cephにより、指定したファイル名に、コンパイル済みのCRUSHマップが出力(`-o`)されます。CRUSHマップはコンパイル済み形式なので、編集する前に逆コンパイルする必要があります。

2. CRUSHマップを逆コンパイルします。CRUSHマップを逆コンパイルするには、次のコマンドを実行します。

```
cephuser@adm > crushtool -d compiled-crushmap-filename \  
-o decompiled-crushmap-filename
```

Cephにより、コンパイル済みのCRUSHマップが逆コンパイル(-d)されて、指定したファイル名に出力(-o)されます。

3. デバイス、バケット、およびルールのパラメータを少なくとも1つ編集します。
4. CRUSHマップをコンパイルします。CRUSHマップをコンパイルするには、次のコマンドを実行します。

```
cephuser@adm > crushtool -c decompiled-crush-map-filename \  
-o compiled-crush-map-filename
```

Cephにより、指定したファイル名に、コンパイル済みのCRUSHマップが保存されます。

5. CRUSHマップを設定します。クラスタのCRUSHマップを設定するには、次のコマンドを実行します。

```
cephuser@adm > ceph osd setcrushmap -i compiled-crushmap-filename
```

Cephにより、指定したファイル名のコンパイル済みCRUSHマップがクラスタのCRUSHマップとして入力されます。



## ヒント: バージョン管理システムの使用

エクスポートおよび変更したCRUSHマップファイルには、gitやsvnなどのバージョン管理システムを使用します。これにより、ロールバックを簡単に行うことができます。



## ヒント: 新しいCRUSHマップのテスト

調整した新しいCRUSHマップは、**crushtool --test**コマンドを使用してテストし、新しいCRUSHマップを適用する前の状態と比較します。次のコマンドスイッチが役立つ場合があります。**--show-statistics**、**--show-mappings**、**--show-bad-mappings**、**--show-utilization**、**--show-utilization-all**、**--show-choose-tries**

## 17.5.2 OSDの追加または移動

実行中のクラスタのCRUSHマップでOSDを追加または移動するには、次のコマンドを実行します。

```
cephuser@adm > ceph osd crush set id_or_name weight root=pool-name  
bucket-type=bucket-name ...
```

### id

整数。OSDの数値ID。このオプションは必須です。

### name

文字列。OSDの完全な名前。このオプションは必須です。

### weight

倍精度。OSDのCRUSHの重み。このオプションは必須です。

### root

キー/値のペア。CRUSH階層には、デフォルトでそのルートとしてプールが含まれます。このオプションは必須です。

### bucket-type

キー/値のペア。CRUSH階層におけるOSDの場所を指定できます。

次の例は、osd.0を階層に追加するか、前の場所からそのOSDを移動します。

```
cephuser@adm > ceph osd crush set osd.0 1.0 root=data datacenter=dc1 room=room1 \  
row=foo rack=bar host=foo-bar-1
```

## 17.5.3 ceph osd reweightとceph osd crush reweightの違い

Ceph OSDの「重み」を変更する、類似するコマンドが2つあります。これらを使用するコンテキストは異なるため、混乱を招く可能性があります。

### 17.5.3.1 ceph osd reweight

使用方法:

```
cephuser@adm > ceph osd reweight OSD_NAME NEW_WEIGHT
```

**ceph osd reweight**はCeph OSDに上書きの重みを設定します。この値は0～1の範囲です。それ以外の値に設定すると、CRUSHは、通常であればこのドライブ上に存続するデータを強制的に再配置します。OSD上のバケットに割り当てられた重みは変更「しません」。これは、CRUSHによる通常の分散が適切に機能しない場合の修正手段です。たとえば、OSDの1つが90%で、その他が40%の場合、この重みを減らして補正を試してみることができます。



### 注記: OSDの重みは一時的である

**ceph osd reweight**は永続的な設定ではないことに注意してください。あるOSDにマークを付けるとその重みは0に設定され、もう一度マークを付けるとその重みは1に変更されます。

## 17.5.3.2 **ceph osd crush reweight**

使用方法:

```
cephuser@adm > ceph osd crush reweight OSD_NAME NEW_WEIGHT
```

**ceph osd crush reweight**は、OSDの「CRUSH」の重みを設定します。この重みは任意の値（一般的にはディスクのTB単位のサイズ）で、システムがOSDに割り当てようとするデータの量を制御します。

## 17.5.4 **OSDの削除**

実行中のクラスタのCRUSHマップからOSDを削除するには、次のコマンドを実行します。

```
cephuser@adm > ceph osd crush remove OSD_NAME
```

## 17.5.5 **バケットの追加**

実行中のクラスタのCRUSHマップにバケットを追加するには、**ceph osd crush add-bucket**コマンドを実行します。

```
cephuser@adm > ceph osd crush add-bucket BUCKET_NAME BUCKET_TYPE
```

## 17.5.6 バケットの移動

CRUSHマップ階層の別の場所または位置へバケットを移動するには、次のコマンドを実行します。

```
cephuser@adm > ceph osd crush move BUCKET_NAME BUCKET_TYPE=BUCKET_NAME [...]
```

以下に例を示します。

```
cephuser@adm > ceph osd crush move bucket1 datacenter=dc1 room=room1 row=foo rack=bar  
host=foo-bar-1
```

## 17.5.7 バケットの削除

CRUSHマップ階層からバケットを削除するには、次のコマンドを実行します。

```
cephuser@adm > ceph osd crush remove BUCKET_NAME
```



### 注記: 空のバケットのみ

CRUSH階層からバケットを削除する前に、バケットを空にする必要があります。

## 17.6 配置グループのスクラブ

オブジェクトの複数のコピーを作成するほかに、Cephは、配置グループを「スクラブ」することによってデータの整合性を保証します(配置グループの詳細については、『導入ガイド』、第1章「SESとCeph」、1.3.2項「配置グループ」を参照)。Cephのスクラブは、Object Storage層に対して**fsck**を実行することに似ています。各配置グループについて、Cephは、すべてのオブジェクトのカタログを生成し、各プライマリオブジェクトとそのレプリカを比較して、オブジェクトの欠落や不一致がないことを確認します。日次の軽量スクラブではオブジェクトのサイズと属性を確認するのに対し、週次の詳細スクラブではデータを読み込み、チェックサムを使用してデータの整合性を保証します。

スクラブはデータの整合性を維持するために重要ですが、パフォーマンスを低下させる可能性があります。次の設定を調整して、スクラブ操作を増減できます。

osd max scrubs

Ceph OSDの同時スクラブ操作の最大数。デフォルトは1です。

#### osd scrub begin hour、osd scrub end hour

スクラブを実行可能な時間枠を定義する時間(0~24)。デフォルトでは、0から始まり24に終了します。

### 重要

配置グループのスクラブ間隔が`osd scrub max interval`の設定を超えている場合、スクラブは、定義した時間枠とは関係なく実行されます。

#### osd scrub during recovery

回復時のスクラブを許可します。「false」に設定すると、アクティブな回復がある間は、新しいスクラブのスケジューリングが無効になります。すでに実行中のスクラブは続行されます。このオプションは、高負荷のクラスタの負荷を下げる場合に役立ちます。デフォルトは「true」です。

#### osd scrub thread timeout

スクラブスレッドがタイムアウトするまでの最大時間(秒単位)。デフォルトは60です。

#### osd scrub finalize thread timeout

スクラブ最終処理スレッドがタイムアウトするまでの最大時間(秒単位)。デフォルトは60\*10です。

#### osd scrub load threshold

正規化された最大負荷。システム負荷(`getloadavg()` / `online_cpus`の数の比率で定義)がこの数字を超えた場合、Cephはスクラブを実行しません。デフォルトは0.5です。

#### osd scrub min interval

Cephクラスタの負荷が低い場合にCeph OSDをスクラブする最小間隔(秒単位)。デフォルトは60\*60\*24 (1日1回)です。

#### osd scrub max interval

クラスタの負荷に関係なくCeph OSDをスクラブする最大間隔(秒単位)。デフォルトは7\*60\*60\*24 (週1回)です。

#### osd scrub chunk min

1回の操作でスクラブするObject Storeチャンクの最大数。スクラブ中、Cephは1つのチャンクへの書き込みをブロックします。デフォルトは5です。

#### osd scrub chunk max

1回の操作でスクラブするObject Storeチャンクの最大数。デフォルトは25です。

#### osd scrub sleep

チャンクの次のグループをスクラブするまでのスリープ時間。この値を増やすとスクラブ操作全体の速度が低下しますが、クライアントの操作への影響は少なくなります。デフォルトは0です。

#### osd deep scrub interval

「詳細」スクラブ(すべてのデータを完全に読み込み)の間隔。osd scrub load thresholdオプションはこの設定に影響しません。デフォルトは $60 \times 60 \times 24 \times 7$  (週1回)です。

#### osd scrub interval randomize ratio

配置グループに対して次のスクラブジョブをスケジューリングする際に、osd scrub min intervalの値にランダムな遅延を追加します。この遅延は、osd scrub min interval \* osd scrub interval randomized ratioの結果よりも小さいランダムな値です。したがって、デフォルト設定では、許可された時間枠である $[1, 1.5] \times \text{osd scrub min interval}$ の中で実質的にランダムにスクラブが分散されます。デフォルトは0.5です。

#### osd deep scrub stride

詳細スクラブ実行時の読み込みサイズ。デフォルトは524288 (512KB)です。

## 18 ストレージプールの管理

Cephはデータをプール内に保存します。プールは、オブジェクトを保存するための論理グループです。プールを作成せずに初めてクラスタを展開した場合、Cephはデフォルトのプールを使用してデータを保存します。次の重要な特徴はCephプールに関連するものです。

- 「災害耐性」：Cephプールは、プール内のデータを複製またはエンコードすることで、災害耐性をもたらします。各プールは複製プール(replicated)、またはイレージャコーディングプール(erasure coding)に設定できます。複製プールの場合、プール内の各データオブジェクトが持つレプリカ(コピー)の数をさらに設定します。失っても問題ないコピー(OSD、CRUSHバケット/リーフ)の数は、レプリカの数 - 1個までです。イレージャコーディングを使用する場合、値 $k$ と値 $m$ を設定します。値 $k$ はデータチャンクの数で、値 $m$ はコーディングチャンクの数です。イレージャコーディングプールの場合、失ってもデータに問題が生じないOSD(CRUSHバケット/リーフ)の数は、コーディングチャンクの数により決まります。
- 「配置グループ」：プールの配置グループの数を設定できます。一般的な設定では、OSDあたり約100個の配置グループを使用し、大量のコンピューティングリソースを使用することなく最適なバランスを提供します。複数のプールを設定する場合は、プールとクラスタ全体の両方にとって適切な数の配置グループを設定するよう注意してください。
- 「CRUSHルール」：プールにデータを保存する場合、オブジェクトとそのレプリカ(またはイレージャコーディングプールの場合はチャンク)は、プールにマップされたCRUSHルールに従って配置されます。ご使用のプールに対してカスタムCRUSHルールを作成できます。
- 「スナップショット」：`ceph osd pool mksnap`を使用してスナップショットを作成すると、特定のプールのスナップショットが効果的に作成されます。

データをプールに編成するために、プールを一覧、作成、および削除できます。各プールの使用量統計を表示することもできます。

### 18.1 プールの作成

オブジェクトのコピーを複数保持することによってOSDの損失から回復するにはreplicated、汎用RAID5またはRAID6機能を利用するにはerasureを指定して、プールを作成できます。必要な未加工ストレージは、複製プールでは多く、イレージャコーディングプールでは少なくなります。デフォルトの設定値はreplicatedです。イレージャコーディングプールの詳細については、[第19章「イレージャコーディングプール」](#)を参照してください。

複製プールを作成するには、次のコマンドを実行します。

```
cephuser@adm > ceph osd pool create POOL_NAME
```



## 注記

これ以外のオプション引数については、自動拡張機能で処理されます。詳細については、[17.4.12項「配置グループの自動拡張の有効化」](#)を参照してください。

イレージャコーディングプールを作成するには、次のコマンドを実行します。

```
cephuser@adm > ceph osd pool create POOL_NAME erasure CRUSH_RULESET_NAME \
EXPECTED_NUM_OBJECTS
```

OSDあたりの配置グループの制限を超える場合、**ceph osd pool create**コマンドは失敗する可能性があります。この制限はオプション`mon_max_pg_per_osd`で設定します。

### POOL\_NAME

プールの名前。固有である必要があります。このオプションは必須です。

### POOL\_TYPE

プールタイプ。オブジェクトのコピーを複数保持することによってOSDの損失から回復するには「replicated」、一種の汎用RAID5機能を利用するには「erasure」を指定できます。複製プールの場合、必要な未加工ストレージが増えますが、Cephのすべての操作が実装されます。イレージャプールの場合、必要な未加工ストレージは減りますが、利用可能な操作のサブセットのみが実装されます。デフォルトのPOOL\_TYPEはreplicatedです。

### CRUSH\_RULESET\_NAME

このプールのCRUSHルールセットの名前。指定したルールセットが存在しない場合、複製プールの作成は-ENOENTで失敗します。複製プールでは`osd pool default CRUSH replicated ruleset`設定変数で指定されるルールセットです。このルールセットは存在している必要があります。イレージャプールでは、デフォルトのイレージャコードプロファイルを使用する場合は「erasure-code」、それ以外の場合はPOOL\_NAMEです。このルールセットは、まだ存在しない場合は暗黙的に作成されます。

### erasure\_code\_profile=profile

イレージャコーディングプール専用。イレージャコードプロファイルを使用します。`osd erasure-code-profile set`で定義した既存のプロファイルである必要があります。



## 注記

何らかの理由でプールの自動拡張が無効化(`pg_autoscale_mode`がoffに設定)されている場合は、手動でPG数を計算して設定できます。プールに適した配置グループ数の計算の詳細については、「[17.4項「配置グループ」](#)」を参照してください。

### EXPECTED\_NUM\_OBJECTS

このプールの想定オブジェクト数。この値を(負の`filestore merge threshold`とともに)設定すると、プールの作成時にPGフォルダが分割されます。これにより、ランタイム時のフォルダ分割によるレイテンシの影響が避けられます。

## 18.2 プールの一覧

クラスタのプールを一覧にするには、次のコマンドを実行します。

```
cephuser@adm > ceph osd pool ls
```

## 18.3 プールの名前変更

プールの名前を変更するには、次のコマンドを実行します。

```
cephuser@adm > ceph osd pool rename CURRENT_POOL_NAME NEW_POOL_NAME
```

プールの名前を変更する場合に、認証ユーザ用のプールごとのケーパビリティがあるときは、そのユーザのケーパビリティを新しいプール名で更新する必要があります。

## 18.4 プールの削除



### 警告: プールの削除は元に戻せない

プールには重要なデータが収められている場合があります。プールを削除すると、プール内のすべてのデータが消え、回復する方法はありません。

誤ってプールを削除することはきわめて危険であるため、Cephには、プールの削除を防止するメカニズムが2つ実装されています。プールを削除するには、両方のメカニズムを無効にする必要があります。

1つ目のメカニズムはNODELETEフラグです。各プールにこのフラグがあり、デフォルト値は「false」です。プールのこのフラグのデフォルト値を確認するには、次のコマンドを実行します。

```
cephuser@adm > ceph osd pool get pool_name nodelete
```

nodelete: trueが出力される場合、次のコマンドを使用してフラグを変更しない限り、プールを削除できません。

```
cephuser@adm > ceph osd pool set pool_name nodelete false
```

2つ目のメカニズムは、クラスタ全体の設定パラメータmon allow pool deleteで、デフォルトは「false」です。つまり、デフォルトではプールを削除できません。表示されるエラーメッセージは次のとおりです。

```
Error EPERM: pool deletion is disabled; you must first set the
mon_allow_pool_delete config option to true before you can destroy a pool
```

この安全設定に関係なくプールを削除するには、mon allow pool deleteを一時的に「true」に設定してプールを削除し、その後、パラメータを「false」に戻します。

```
cephuser@adm > ceph tell mon.* injectargs --mon-allow-pool-delete=true
cephuser@adm > ceph osd pool delete pool_name pool_name --yes-i-really-really-mean-it
cephuser@adm > ceph tell mon.* injectargs --mon-allow-pool-delete=false
```

**injectargs** コマンドを実行すると、次のメッセージが表示されます。

```
injectargs:mon_allow_pool_delete = 'true' (not observed, change may require restart)
```

これは単にコマンドが正常に実行されたことを確認するものです。エラーではありません。作成したプール用に独自のルールセットとルールを作成した場合、プールがなくなったらルールセットとルールを削除することをお勧めします。

## 18.5 その他の操作

### 18.5.1 プールとアプリケーションの関連付け

プールを使用する前に、プールをアプリケーションに関連付ける必要があります。CephFSで使用されるプール、またはObject Gatewayによって自動的に作成されるプールは自動的に関連付けられます。

それ以外の場合は、自由な形式のアプリケーション名を手動でプールに関連付けることができます。

```
cephuser@adm > ceph osd pool application enable POOL_NAME APPLICATION_NAME
```



## ヒント: デフォルトのアプリケーション名

アプリケーション名として、CephFSは`cephfs`、RADOS Block Deviceは`rbd`、Object Gatewayは`rgw`をそれぞれ使用します。

1つのプールを複数のアプリケーションに関連付けて、各アプリケーションで専用のメタデータを使用できます。プールに関連付けられたアプリケーションを一覧にするには、次のコマンドを実行します。

```
cephuser@adm > ceph osd pool application get pool_name
```

## 18.5.2 プールのクォータの設定

最大バイト数、またはプールあたりのオブジェクトの最大数に対してプールクォータを設定できます。

```
cephuser@adm > ceph osd pool set-quota POOL_NAME MAX_OBJECTS OBJ_COUNT MAX_BYTES BYTES
```

以下に例を示します。

```
cephuser@adm > ceph osd pool set-quota data max_objects 10000
```

クォータを削除するには、値を0に設定します。

## 18.5.3 プールの統計情報の表示

プールの使用量統計を表示するには、次のコマンドを実行します。

```
cephuser@adm > rados df
```

POOL_NAME	DEGRADED	RD_OPS	RD	WR_OPS	WR	USED	CLONES	COPIES	MISSING_ON_PRIMARY	UNFOUND
							USED	COMPR	UNDER	COMPR
.rgw.root				768 KiB	4	0	12		0	0
0	44	44 KiB	4	4 KiB	0 B		0 B			
cephfs_data				960 KiB	5	0	15		0	0
0	5502	2.1 MiB	14	11 KiB	0 B		0 B			

cephfs_metadata	1.5 MiB	22	0	66	0	0
0 26 78 KiB	176 147 KiB	0 B	0 B	0 B		
default.rgw.buckets.index	0 B	1	0	3	0	0
0 4 4 KiB	1 0 B	0 B	0 B	0 B		
default.rgw.control	0 B	8	0	24	0	0
0 0 0 B	0 0 B	0 B	0 B	0 B		
default.rgw.log	0 B	207	0	621	0	0
0 5372132 5.1 GiB	3579618 0 B	0 B	0 B	0 B		
default.rgw.meta	961 KiB	6	0	18	0	0
0 155 140 KiB	14 7 KiB	0 B	0 B	0 B		
example_rbd_pool	2.1 MiB	18	0	54	0	0
0 3350841 2.7 GiB	118 98 KiB	0 B	0 B	0 B		
iscsi-images	769 KiB	8	0	24	0	0
0 1559261 1.3 GiB	61 42 KiB	0 B	0 B	0 B		
mirrored-pool	1.1 MiB	10	0	30	0	0
0 475724 395 MiB	54 48 KiB	0 B	0 B	0 B		
pool2	0 B	0	0	0	0	0
0 0 0 B	0 0 B	0 B	0 B	0 B		
pool3	333 MiB	37	0	111	0	0
0 3169308 2.5 GiB	14847 118 MiB	0 B	0 B	0 B		
pool4	1.1 MiB	13	0	39	0	0
0 1379568 1.1 GiB	16840 16 MiB	0 B	0 B	0 B		

個々の列の説明は次のとおりです。

## USED

プールによって使用されているバイトの数。

## OBJECTS

プールに保存されているオブジェクトの数。

## CLONES

プールに保存されているクローンの数。スナップショットが作成されてオブジェクトに書き込まれる場合、元のオブジェクトは変更されずにそのクローンが作成されるため、元のスナップショットオブジェクトの内容は変更されません。

## COPIES

オブジェクトレプリカの数。たとえば、レプリケーション係数3の複製プールに「x」個のオブジェクトがある場合、コピーの数は通常、3 \* x個になります。

## MISSING\_ON\_PRIMARY

プライマリOSDにコピーが見つからないときに劣化状態になっているオブジェクトの数 (すべてのコピーが存在するわけではない)。

## UNFOUND

見つからないオブジェクトの数。

## DEGRADED

劣化したオブジェクトの数。

## RD\_OPS

このプールに対して要求された読み込み操作の合計数。

## RD

このプールから読み込まれたバイトの合計数。

## WR\_OPS

このプールに対して要求された書き込み操作の合計数。

## WR

プールに書き込まれたバイトの合計数。同じオブジェクトに何度も書き込むことができるため、これはプールの使用率と同じではないことに注意してください。その結果、プールの使用率は同じままであっても、プールに書き込まれたバイト数は大きくなります。

## USED COMPR

圧縮データに割り当てられているバイトの数。

## UNDER COMPR

圧縮データが非圧縮時に使用するバイトの数。

## 18.5.4 プールから値を取得

プールから値を取得するには、次のように`get`コマンドを実行します。

```
cephuser@adm > ceph osd pool get POOL_NAME KEY
```

18.5.5項「プールに値を設定」に示すキーと、次のキーの値を取得できます。

## PG\_NUM

プールの配置グループの数。

## PGP\_NUM

データ配置を計算する際に使用する配置グループの有効数。有効な範囲は`PG_NUM`以下です。



## ヒント: プールのすべての値

特定のプールに関連するすべての値を一覧にするには、次のコマンドを実行します。

```
cephuser@adm > ceph osd pool get POOL_NAME all
```

## 18.5.5 プールに値を設定

プールに値を設定するには、次のコマンドを実行します。

```
cephuser@adm > ceph osd pool set POOL_NAME KEY VALUE
```

プールタイプごとにソートしたプールの値のリストを以下に示します。

### 共通するプールの値

#### crash\_replay\_interval

確認済みであるもののコミットされていない要求の再生をクライアントに許可する秒数。

#### pg\_num

プールの配置グループの数。新しいクラスタにOSDを追加する場合は、新しいOSDの対象に指定されたすべてのプール上にある配置グループの値を確認します。

#### pgp\_num

データ配置を計算する際に使用する配置グループの有効数。

#### crush\_ruleset

クラスタ内のオブジェクト配置のマッピングに使用するルールセット。

#### hashpspool

指定したプールに対してHASHPSPOOLフラグを設定(1)または設定解除(0)します。このフラグを有効にすると、PGをOSDに効率的に分散するためにアルゴリズムが変更されます。HASHPSPOOLフラグがデフォルトの0に設定されたプールでこのフラグを有効にすると、クラスタは、すべてのPGをもう一度正しく配置するためにバックフィルを開始します。これはクラスタに多大なI/O負荷をかける可能性があるため、非常に負荷が高い運用クラスタでは、0～1のフラグを有効にしないでください。

#### nodelete

プールの削除を防止します。

#### nopgchange

プールの`pg_num`および`pgp_num`の変更を防止します。

## **noscrub、nodeep-scrub**

I/Oの一時的な高負荷を解決するため、特定のプールに対してデータの(ディープ)スクラブを無効にします。

## **write\_fadvise\_dontneed**

指定したプールの読み込み/書き込み要求のWRITE\_FADVISE\_DONTNEEDフラグを設定または設定解除して、キャッシュへのデータ格納をバイパスします。デフォルトはfalseです。複製プールとECプールの両方に適用されます。

## **scrub\_min\_interval**

クラスタの負荷が低い場合にプールをスクラブする最小間隔(秒単位)。デフォルトの0は、Ceph設定ファイルのosd\_scrub\_min\_intervalの値が使用されることを意味します。

## **scrub\_max\_interval**

クラスタの負荷に関係なくプールをスクラブする最大間隔(秒単位)。デフォルトの0は、Ceph設定ファイルのosd\_scrub\_max\_intervalの値が使用されることを意味します。

## **deep\_scrub\_interval**

プールの「詳細」スクラブの間隔(秒単位)。デフォルトの0は、Ceph設定ファイルのosd\_deep\_scrubの値が使用されることを意味します。

## **複製プールの値**

### **size**

プール内のオブジェクトのレプリカ数を設定します。詳細については、18.5.6項「オブジェクトレプリカの数設定」を参照してください。複製プール専用です。

### **min\_size**

I/Oに必要なレプリカの最小数を設定します。詳細については、18.5.6項「オブジェクトレプリカの数設定」を参照してください。複製プール専用です。

## **nosizechange**

プールのサイズの変更を防止します。プールが作成された際にデフォルト値としてosd\_pool\_default\_flag\_nosizechangeパラメータの値を取得します。このパラメータはデフォルトではfalseです。複製プールにのみ適用できます。ECプールはサイズを変更できないためです。

### hit\_set\_type

キャッシュプールのヒットセットの追跡を有効にします。詳細については、「[Bloom Filter \(http://en.wikipedia.org/wiki/Bloom\\_filter\)](http://en.wikipedia.org/wiki/Bloom_filter)」を参照してください。このオプションに設定できる値は、`bloom`、`explicit_hash`、または`explicit_object`です。デフォルトは`bloom`で、他の値はテスト専用です。

### hit\_set\_count

キャッシュプールに関して保存するヒットセットの数。値を増やすほど、`ceph-osd`デーモンのRAM消費量が増えます。デフォルトは0です。

### hit\_set\_period

キャッシュプールのヒットセットの期間(秒単位)。値を増やすほど、`ceph-osd`デーモンのRAM消費量が増えます。プールが作成された際にデフォルト値として`osd_tier_default_cache_hit_set_period`パラメータの値を取得します。このパラメータのデフォルト値は1200です。複製プールにのみ適用できます。ECプールはキャッシュ層として使用できないためです。

### hit\_set\_fpp

`bloom`ヒットセットタイプの誤検知確率。詳細については、「[Bloom Filter \(http://en.wikipedia.org/wiki/Bloom\\_filter\)](http://en.wikipedia.org/wiki/Bloom_filter)」を参照してください。有効な範囲は0.0～1.0で、デフォルトは0.05です。

### use\_gmt\_hitset

キャッシュ階層化のヒットセットを作成する際に、GMT (グリニッジ標準時)のタイムスタンプを使用するようOSDに強制します。これにより、異なるタイムゾーンにあるノードが同じ結果を返すようにします。デフォルトは1です。この値は変更できません。

### cache\_target\_dirty\_ratio

キャッシュプールに含まれる変更済みオブジェクトの割合で、この割合を超えると、キャッシュ階層化エージェントは変更済み(ダーティ)オブジェクトをバックグストレージプールにフラッシュします。デフォルトは0.4です。

### cache\_target\_dirty\_high\_ratio

キャッシュプールに含まれる変更済みオブジェクトの割合で、この割合を超えると、キャッシュ階層化エージェントは変更済み(ダーティ)オブジェクトをより高速なバックグストレージプールにフラッシュします。デフォルトは0.6です。

### cache\_target\_full\_ratio

キャッシュプールに含まれる未変更オブジェクトの割合で、この割合を超えると、キャッシュ階層化エージェントは未変更(クリーン)オブジェクトをキャッシュプールから削除します。デフォルトは0.8です。

### **target\_max\_bytes**

max\_bytesのしきい値がトリガされた場合、Cephはオブジェクトのフラッシュまたは削除を開始します。

### **target\_max\_objects**

max\_objectsのしきい値がトリガされた場合、Cephはオブジェクトのフラッシュまたは削除を開始します。

### **hit\_set\_grade\_decay\_rate**

連続する2つのhit\_set間の温度減衰率。デフォルトは20です。

### **hit\_set\_search\_last\_n**

温度を計算するために、hit\_set内で最大N個の出現をカウントします。デフォルトは1です。

### **cache\_min\_flush\_age**

キャッシュ階層化エージェントがオブジェクトをキャッシュプールからストレージプールへフラッシュするまでの時間(秒単位)。

### **cache\_min\_evict\_age**

キャッシュ階層化エージェントがオブジェクトをキャッシュプールから削除するまでの時間(秒単位)。

## イレージャコーディングプールの値

### **fast\_read**

イレージャコーディングプールでこのフラグが有効な場合、読み込み要求は、すべてのシャードに対してサブ読み込みを発行し、クライアントの要求を実行するためにデコードする十分なシャードを受け取るまで待機します。イレージャプラグインが**jerasure**および**isa**の場合、最初のK個の応答が返された時点で、これらの応答からデコードされたデータを使用してただちにクライアントの要求が実行されます。このアプローチでは、CPUの負荷が増え、ディスク/ネットワークの負荷は減ります。現在のところ、このフラグはイレージャコーディングプールでのみサポートされます。デフォルトは0です。

## 18.5.6 オブジェクトレプリカ数の設定

複製プール上のオブジェクトレプリカ数を設定するには、次のコマンドを実行します。

```
cephuser@adm > ceph osd pool set poolname size num-replicas
```

`num-replicas`にはオブジェクトそのものも含まれます。たとえば、オブジェクトとそのオブジェクトの2つのコピーで合計3つのオブジェクトインスタンスが必要な場合、3を指定します。



### 警告: 3つ未満のレプリカを設定しない

`num-replicas`を2に設定した場合、データのコピーは「1つ」だけになります。1つのオブジェクトインスタンスが失われた場合、たとえば回復中の前回のスクラブ以降に、他のコピーが壊れていないことを信頼する必要があります(詳細については、[17.6項「配置グループのスクラブ」](#)を参照)。

プールを1つのレプリカに設定することは、プール内にデータオブジェクトのインスタンスが「1つ」だけ存在することを意味します。OSDに障害発生すると、データは失われます。レプリカが1つのプールの使用法としては、一時データを短時間保存することが考えられます。



### ヒント: 3つを超えるレプリカの設定

1つのプールに対して4つのレプリカを設定すると、信頼性が25%向上します。

2つのデータセンターの場合、各データセンターで2つのコピーを使用できるよう、1つプールに対してレプリカを4つ以上設定します。これにより、一方のデータセンターが失われてもまだ2つのコピーが存在し、さらにディスクが1つ失われてもデータが失われないようにします。



### 注記

1つのオブジェクトが、機能低下モードにおいてレプリカが`pool size`未満の状態ではI/Oを受け付ける場合があります。I/Oに必要なレプリカの最小数を設定するには、`min_size`設定を使用する必要があります。次に例を示します。

```
cephuser@adm > ceph osd pool set data min_size 2
```

これにより、データプール内のオブジェクトはレプリカが`min_size`未満の場合、I/Oを受け取らなくなります。



### ヒント: オブジェクトレプリカの数取得

オブジェクトレプリカの数取得するには、次のコマンドを実行します。

```
cephuser@adm > ceph osd dump | grep 'replicated size'
```

`replicated size`属性が強調表示された状態でプールが一覧にされます。デフォルトでは、Cephはオブジェクトのレプリカを2つ作成します(合計で3つのコピー、またはサイズ3)。

## 18.6 プールのマイグレーション

プールを作成する際(18.1項「プールの作成」を参照)、プールのタイプや配置グループの数など、初期パラメータを指定する必要があります。後でこれらのパラメータのいずれかを変更する場合(たとえば、複製プールをイレージャコーディングプールに変換したり、配置グループの数を減らしたりする場合)、プールのデータを、展開に適したパラメータを持つ別のプールに移行する必要があります。

このセクションでは、2つのマイグレーション方法を説明します。1つは「キャッシュ層」を使う方法で一般的なプールのデータマイグレーションに使用します。もう1つは`rbd migrate`サブコマンドを使用する方法で、RBDイメージを新しいプールに移行します。どちらの方法にもその詳細と制限があります。

### 18.6.1 制限

- 「キャッシュ層」の方法を使用して、複製プールからECプールまたは別の複製プールに移行できます。ECプールからの移行はサポートされていません。
- RBDイメージとCephFSエクスポートを複製プールからECプールに移行することはできません。その理由として、RBDとCephFSは`omap`を使用してメタデータを保存していますが、ECプールは`omap`をサポートしていないためです。たとえば、RBDのヘッダオブジェクトはフラッシュできません。それでも、メタデータを複製プールに残したまま、データをECプールに移行することは可能です。
- `rbd migration`による方法を使用すると、クライアントのダウンタイムを最小限に抑えてイメージを移行できます。必要なのは、`prepare`ステップの前にクライアントを停止して、後でクライアントを起動することだけです。`prepare`ステップの直後にイメージを開くことができるのは、この機能をサポートする`librbd`クライアント(Ceph Nautilus以降)のみであり、それ以前の`librbd`クライアントや`krbd`クライアントは`commit`ステップが実行されるまでイメージを開くことができないことに注意してください。

## 18.6.2 キャッシュ層を使用した移行

原理は単純で、移行する必要があるプールを逆の順番でキャッシュ層に含めます。次の例では、「testpool」という名前の複製プールをイレージャコーディングプールに移行します。

### 手順 18.1: 複製プールからイレージャコーディングプールへの移行

1. 「newpool」という名前の新しいイレージャコーディングプールを作成します。プールの作成パラメータの詳細な説明については、18.1項「プールの作成」を参照してください。

```
cephuser@adm > ceph osd pool create newpool erasure default
```

使用されているクライアントキーリングが「testpool」と少なくとも同じ機能を「newpool」に提供することを確認します。

これでプールが2つできました。データが入った元の複製プール「testpool」と、新しい空のイレージャコーディングプール「newpool」です。

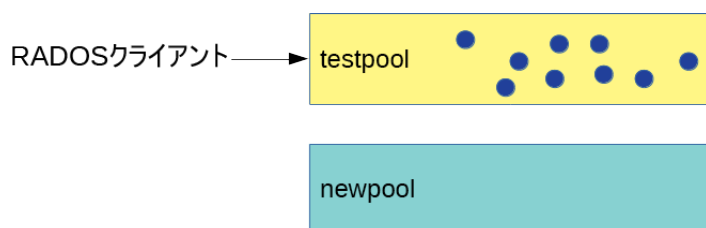


図 18.1: マイグレーション前のプール

2. キャッシュ層をセットアップして、複製プール「testpool」をキャッシュプールとして設定します。-force-nonemptyオプションを使用すると、プールにすでにデータがある場合にもキャッシュ層を追加できます。

```
cephuser@adm > ceph tell mon.* injectargs \
'--mon_debug_unsafe_allow_tier_with_nonempty_snaps=1'
cephuser@adm > ceph osd tier add newpool testpool --force-nonempty
cephuser@adm > ceph osd tier cache-mode testpool proxy
```

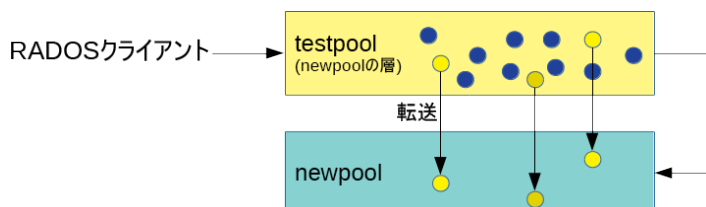


図 18.2: キャッシュ層のセットアップ

3. キャッシュプールからすべてのオブジェクトを新しいプールに強制的に移動します。

```
cephuser@adm > rados -p testpool cache-flush-evict-all
```

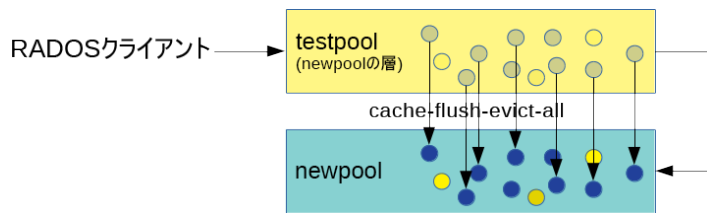


図 18.3: データのフラッシュ

- すべてのデータが新しいイレージャコーディングプールにフラッシュされるまでは、オーバーレイを指定してオブジェクトが古いプールで検索されるようにする必要があります。

```
cephuser@adm > ceph osd tier set-overlay newpool testpool
```

このオーバーレイにより、すべての操作が古い複製プール「testpool」に転送されます。

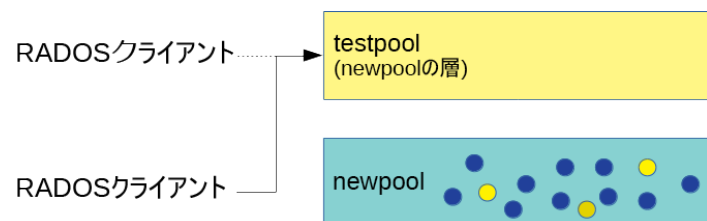


図 18.4: オーバーレイの設定

これで、新しいプールのオブジェクトにアクセスするようすべてのクライアントを切り替えることができます。

- すべてのデータがイレージャコーディングプール「newpool」に移行されたら、オーバーレイと古いキャッシュプール「testpool」を削除します。

```
cephuser@adm > ceph osd tier remove-overlay newpool  
cephuser@adm > ceph osd tier remove newpool testpool
```

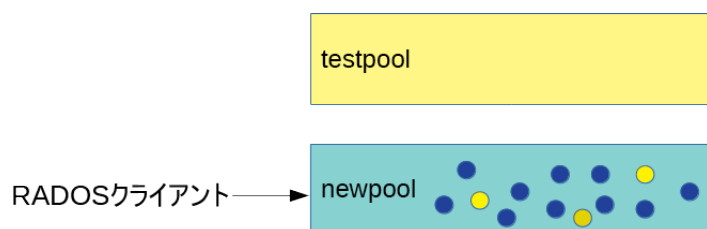


図 18.5: マイグレーションの完了

## 6. 実行

```
cephuser@adm > ceph tell mon.* injectargs \  
'--mon_debug_unsafe_allow_tier_with_nonempty_snaps=0'
```

### 18.6.3 RBDイメージの移行

次に、RBDイメージを1つの複製プールから別の複製プールに移行する場合に推奨する方法を示します。

1. クライアント(仮想マシンなど)がRBDイメージにアクセスしないようにします。
2. 新しいイメージをターゲットプール内に作成し、親をソースイメージに設定します。

```
cephuser@adm > rbd migration prepare SRC_POOL/IMAGE TARGET_POOL/IMAGE
```



#### ヒント: イレージャコーディングプールにデータだけを移行する

イメージデータのみを新しいECプールに移行し、メタデータを元の複製プールに残す必要がある場合は、代わりに次のコマンドを実行します。

```
cephuser@adm > rbd migration prepare SRC_POOL/IMAGE \  
--data-pool TARGET_POOL/IMAGE
```

3. クライアントがターゲットプール内のイメージにアクセスできるようにします。
4. データをターゲットプールに移行します。

```
cephuser@adm > rbd migration execute SRC_POOL/IMAGE
```

5. 古いイメージを削除します。

```
cephuser@adm > rbd migration commit SRC_POOL/IMAGE
```

## 18.7 プールのスナップショット

プールのスナップショットは、Cephのプール全体の状態のスナップショットです。プールのスナップショットにより、プールの状態の履歴を保持できます。プールのスナップショットを作成すると、プールサイズに比例したストレージ領域が消費されます。プールのスナップショットを作成する前に、必ず関連するストレージに十分なディスク領域があることを確認してください。

### 18.7.1 プールのスナップショットの作成

プールのスナップショットを作成するには、次のコマンドを実行します。

```
cephuser@adm > ceph osd pool mksnap POOL-NAME SNAP-NAME
```

以下に例を示します。

```
cephuser@adm > ceph osd pool mksnap pool1 snap1  
created pool pool1 snap snap1
```

### 18.7.2 プールのスナップショットの一覧

プールの既存のスナップショットを一覧にするには、次のコマンドを実行します。

```
cephuser@adm > rados lssnap -p POOL_NAME
```

以下に例を示します。

```
cephuser@adm > rados lssnap -p pool1  
1 snap1 2018.12.13 09:36:20  
2 snap2 2018.12.13 09:46:03  
2 snaps
```

### 18.7.3 プールのスナップショットの削除

プールのスナップショットを削除するには、次のコマンドを実行します。

```
cephuser@adm > ceph osd pool rmsnap POOL-NAME SNAP-NAME
```

## 18.8 データ圧縮

BlueStore (詳細については、『導入ガイド』、第1章「SESとCeph」、1.4項「BlueStore」を参照)は、オンザフライでデータを圧縮してディスク容量を節約できます。圧縮率は、システムに保存されるデータによって異なります。圧縮/圧縮解除には、追加のCPUパワーが必要になることに注意してください。

データ圧縮をグローバルに設定し(18.8.3項「グローバル圧縮オプション」を参照)、その後、個々のプールに対して固有の圧縮設定を上書きできます。

プールにデータが含まれるかどうかに関係なく、プールのデータ圧縮を有効/無効にしたり、圧縮アルゴリズムやモードをいつでも変更したりできます。

プールの圧縮を有効にすると、既存のデータに圧縮は適用されなくなります。

プールの圧縮を無効にすると、そのプールのすべてのデータの圧縮が解除されます。

### 18.8.1 圧縮の有効化

`POOL_NAME`という名前のプールのデータ圧縮を有効にするには、次のコマンドを実行します。

```
cephuser@adm > ceph osd pool set POOL_NAME compression_algorithm COMPRESSION_ALGORITHM
cephuser@adm > ceph osd pool set POOL_NAME compression_mode COMPRESSION_MODE
```



#### ヒント: プール圧縮の無効化

プールのデータ圧縮を無効にするには、圧縮アルゴリズムとして「none」を使用します。

```
cephuser@adm > ceph osd pool set POOL_NAME compression_algorithm none
```

### 18.8.2 プール圧縮オプション

次に、すべての圧縮設定のリストを示します。

#### **compression\_algorithm**

使用可能な値は、`none`、`zstd`、`snappy`です。デフォルトは`snappy`です。

どの圧縮アルゴリズムを使用するかは、特定の使用事例によって異なります。次に、推奨事項をいくつか示します。

- 変更する妥当な理由がない限り、デフォルトの `snappy` を使用してください。
- `zstd` は、圧縮率は優れていますが、少量のデータを圧縮する場合には CPU オーバーヘッドが高くなります。
- クラスタの CPU とメモリの使用量に注意しながら、実際のデータのサンプルに対してこれらのアルゴリズムのベンチマークを実行します。

### **compression\_mode**

使用可能な値は、`none`、`aggressive`、`passive`、`force` です。デフォルトは `none` です。

- `none`: 圧縮しません。
- `passive`: `COMPRESSIBLE` と表示されている場合、圧縮します。
- `aggressive`: `INCOMPRESSIBLE` と表示されている場合以外、圧縮します。
- `force`: 常に圧縮します。

### **compression\_required\_ratio**

値: 倍精度、比率 =  $\text{SIZE\_COMPRESSED} / \text{SIZE\_ORIGINAL}$ 。デフォルトは `0.875` です。これは、占有されている容量が圧縮によって 12.5% 以上削減されない場合は、オブジェクトは圧縮されないことを意味します。

この率を上回るオブジェクトは、圧縮効果が低いため圧縮状態では保存されません。

### **compression\_max\_blob\_size**

値: 符号なし整数、バイト単位のサイズ。デフォルト: `0`。

圧縮されるオブジェクトの最大サイズ。

### **compression\_min\_blob\_size**

値: 符号なし整数、バイト単位のサイズ。デフォルト: `0`。

圧縮されるオブジェクトの最小サイズ。

## 18.8.3 グローバル圧縮オプション

次の設定オプションは Ceph 設定で指定でき、1 つのプールだけでなくすべての OSD に適用されます。18.8.2 項「プール圧縮オプション」に一覧にされているプール固有の設定が優先されます。

### **bluestore\_compression\_algorithm**

`compression_algorithm` を参照してください。

### **bluestore\_compression\_mode**

[compression\\_mode](#)を参照してください。

### **bluestore\_compression\_required\_ratio**

[compression\\_required\\_ratio](#)を参照してください。

### **bluestore\_compression\_min\_blob\_size**

値: 符号なし整数、バイト単位のサイズ。デフォルト: 0。

圧縮されるオブジェクトの最大サイズ。この設定はデフォルト

では無視され、[bluestore\\_compression\\_min\\_blob\\_size\\_hdd](#)と

[bluestore\\_compression\\_min\\_blob\\_size\\_ssd](#)が優先されます。0以外の値に設定した場合は、この設定が優先されます。

### **bluestore\_compression\_max\_blob\_size**

値: 符号なし整数、バイト単位のサイズ。デフォルト: 0。

圧縮されるオブジェクトの最大サイズ。このサイズを超えると、オ

ブジェクトはより小さいチャンクに分割されます。この設定はデフォ

ルトでは無視され、[bluestore\\_compression\\_max\\_blob\\_size\\_hdd](#)と

[bluestore\\_compression\\_max\\_blob\\_size\\_ssd](#)が優先されます。0以外の値に設定した場合は、この設定が優先されます。

### **bluestore\_compression\_min\_blob\_size\_ssd**

値: 符号なし整数、バイト単位のサイズ。デフォルト: 8K。

圧縮してソリッドステートドライブに保存されるオブジェクトの最小サイズ。

### **bluestore\_compression\_max\_blob\_size\_ssd**

値: 符号なし整数、バイト単位のサイズ。デフォルト: 64K。

圧縮してソリッドステートドライブに保存されるオブジェクトの最大サイズ。このサイズを超えると、オブジェクトはより小さいチャンクに分割されます。

### **bluestore\_compression\_min\_blob\_size\_hdd**

値: 符号なし整数、バイト単位のサイズ。デフォルト: 128K。

圧縮してハードディスクに保存されるオブジェクトの最小サイズ。

### **bluestore\_compression\_max\_blob\_size\_hdd**

値: 符号なし整数、バイト単位のサイズ。デフォルト: 512K。

圧縮してハードディスクに保存されるオブジェクトの最大サイズ。このサイズを超えると、オブジェクトはより小さいチャンクに分割されます。

## 19 イレージャコーディングプール

Cephでは、プール内のデータの通常のレプリケーションに代わる代替手段が提供されています。これを「イレージャ」または「イレージャコーディング」プールと呼びます。イレージャプールは「複製」プールのすべての機能を備えているわけではありません(たとえば、RBDプールのメタデータを保存することはできません)、必要な未加工ストレージが少なく済みます。1TBのデータを保存可能なデフォルトのイレージャプールでは、1.5TBの未加工ストレージが必要で、これによって単一ディスク障害を許容することができます。複製プールでは同じ目的に対して2TBの未加工ストレージが必要であるため、比較しても遜色ありません。

イレージャコードの背景情報については、[https://en.wikipedia.org/wiki/Erasure\\_code](https://en.wikipedia.org/wiki/Erasure_code)を参照してください。

ECプールに関連するプール値のリストについては、[イレージャコーディングプールの値](#)を参照してください。

### 19.1 イレージャコーディングプールの前提条件

イレージャコーディングを利用するには、以下を行う必要があります。

- CRUSHマップでイレージャルールを定義する
- 使用するコーディングアルゴリズムを指定するイレージャコードプロファイルを定義する
- 前述したルールとプロファイルを使用してプールを作成する

プールを作成してデータを保存した後でプロファイルや、プロファイル内の詳細を変更することはできないことに注意してください。

「イレージャプール」のCRUSHルールでは`step`に`indep`を使用するようにしてください。詳細については、[17.3.2項「firstnとindep」](#)を参照してください。

### 19.2 サンプルのイレージャコーディングプールの作成

最もシンプルなイレージャコーディングプールはRAID5と同等で、少なくとも3つのホストを必要とします。この手順では、テスト用のプールを作成する方法について説明します。

1. コマンド `ceph osd pool create` を使用して、タイプが「erasure」のプールを作成します。12は、配置グループの数を表します。デフォルトのパラメータの場合、このプールは1つのOSDの障害に対応できます。

```
cephuser@adm > ceph osd pool create ecpool 12 12 erasure
pool 'ecpool' created
```

2. 文字列 `ABCDEFGHI` を `NYAN` という名前のオブジェクトに書き込みます。

```
cephuser@adm > echo ABCDEFGHI | rados --pool ecpool put NYAN -
```

3. これで、テストのためにOSDを無効にできます。たとえば、OSDをネットワークから接続解除します。
4. プールがデバイスの障害に対応できるかどうかをテストするため、`rados` コマンドを使用してファイルの内容にアクセスできます。

```
cephuser@adm > rados --pool ecpool get NYAN -
ABCDEFGHI
```

## 19.3 イレージャコードプロファイル

`ceph osd pool create` コマンドを起動して「イレージャプール」を作成する場合、別のプロファイルを指定しない限り、デフォルトのプロファイルが使用されます。プロファイルはデータの冗長性を定義します。このためには、任意に `k` および `m` という名前が付けられた2つのパラメータを定義します。`k` および `m` は、1つのデータを何個の `chunks` (チャンク) に分割するかと、何個のコーディングチャンクを作成するかを定義します。これにより、冗長チャンクは異なるOSDに保存されます。

イレージャプールプロファイルに必要な定義は次のとおりです。

### chunk

エンコーディング関数を呼び出すと、同じサイズの複数のチャンクが返されます。連結して元のオブジェクトを再構成できるデータチャンクと、失われたチャンクの再構築に使用できるコーディングチャンクです。

## k

データチャンクの数。これは、元のオブジェクトが分割されるチャンクの数です。たとえば、 $k = 2$ の場合、10KBのオブジェクトはそれぞれが5KBの $k$ 個のオブジェクトに分割されます。イレージャコーディングプールのデフォルトの`min_size`は、 $k + 1$ です。ただし、書き込みおよびデータが失われるのを防ぐため、`min_size`を $k + 2$ 以上にするをお勧めします。

## m

コーディングチャンクの数。これは、エンコーディング関数によって計算される追加チャンクの数です。コーディングチャンクが2つある場合、2つのOSDに障害が発生してもデータが失われないことを意味します。

## crush-failure-domain

チャンクの分散先のデバイスを定義します。値としてバケットタイプを設定する必要があります。すべてのバケットタイプについては、[17.2項「バケット」](#)を参照してください。障害ドメインがrackの場合、ラック障害時の災害耐性を向上させるため、チャンクは別のラックに保存されます。このためには $k+m$ 個のラックが必要なことに注意してください。

[19.2項「サンプルのイレージャコーディングプールの作成」](#)で使用されているデフォルトのイレージャコードプロファイルでは、1つのOSDまたはホストに障害が発生した場合は、クラスタデータは失われません。したがって、1TBのデータを保存するには、さらに0.5TBの未加工ストレージが必要です。つまり、1TBのデータには1.5TBの未加工ストレージが必要です( $k=2$ 、 $m=1$ であるため)。これは、一般的なRAID 5設定と同等です。比較すると、複製プールでは1TBのデータを保存するのに2TBの未加工ストレージが必要です。

デフォルトのプロファイルの設定は次のコマンドで表示できます。

```
cephuser@adm > ceph osd erasure-code-profile get default
directory=.libs
k=2
m=1
plugin=jerasure
crush-failure-domain=host
technique=reed_sol_van
```

プール作成後はプロファイルを変更できないため、適切なプロファイルを選択することが重要です。異なるプロファイルを持つ新しいプールを作成し、前のプールにあるオブジェクトをすべて新しいプールに移動する必要があります([18.6項「プールのマイグレーション」](#)を参照)。

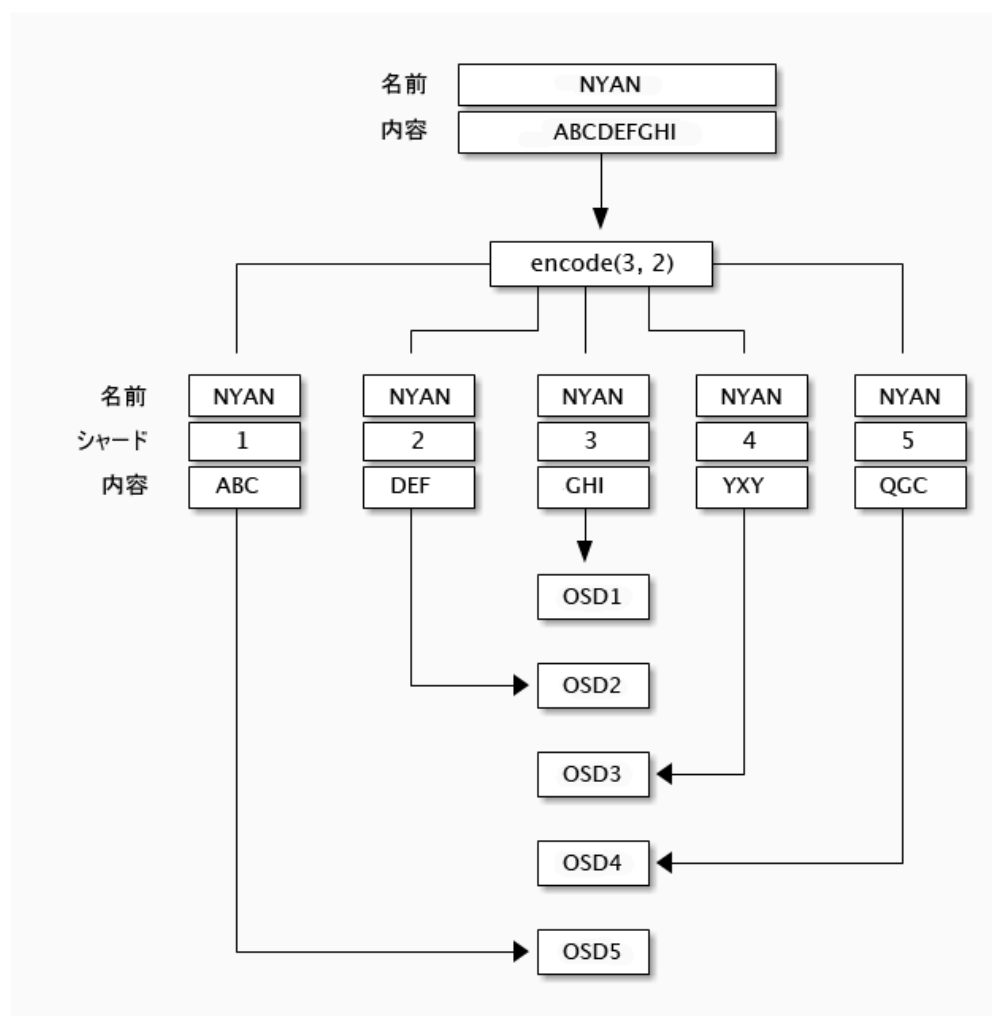
k、m、および`crush-failure-domain`のパラメータは、ストレージのオーバーヘッドとデータの持続性を定義するので、プロファイルの中で最も重要なパラメータです。たとえば、目的のアーキテクチャが66%のストレージオーバーヘッドでラック2台分の損失に耐える必要がある場合、次のプロファイルを定義できます。これは「ラック」タイプのバケットが設定されたCRUSHマップでのみ有効であることに注意してください。

```
cephuser@adm > ceph osd erasure-code-profile set myprofile \
  k=3 \
  m=2 \
  crush-failure-domain=rack
```

この新しいプロファイルで、[19.2項「サンプルのイレージャコーディングプールの作成」](#)の例をもう一度使用できます。

```
cephuser@adm > ceph osd pool create ecpool 12 12 erasure myprofile
cephuser@adm > echo ABCDEFGHI | rados --pool ecpool put NYAN -
cephuser@adm > rados --pool ecpool get NYAN -
ABCDEFGHI
```

NYANオブジェクトは3つ(k=3)に分割され、2つの追加チャンク(m=2)が作成されます。mの値は、データを失うことなく何個のOSDが同時に失われても構わないかを定義します。`crush-failure-domain=rack`は、2つのチャンクが同じラックに保存されないようにするCRUSHルールセットを作成します。



### 19.3.1 新しいイレージャコードプロファイルの作成

次のコマンドで新しいイレージャコードプロファイルを作成します。

```
root # ceph osd erasure-code-profile set NAME \
  directory=DIRECTORY \
  plugin=PLUGIN \
  stripe_unit=STRIPE_UNIT \
  KEY=VALUE ... \
  --force
```

#### DIRECTORY

オプション。イレージャコードプラグインをロードするディレクトリの名前を設定します。デフォルトは /usr/lib/ceph/erasure-code です。

## PLUGIN

オプション。イレージャコードプラグインを使用して、コーディングチャンクを計算し、欠落しているチャンクを回復します。使用可能なプラグインは、「jerasure」、「isa」、「lrc」、および「shes」です。デフォルトは「jerasure」です。

## STRIPE\_UNIT

オプション。ストライプあたりの、データチャンクのデータ量。たとえば、2つのデータチャンクとstripe\_unit=4Kが設定されたプロファイルでは、チャンク0に0~4K、チャンク1に4K~8K、続いてもう一度チャンク0に8K~12Kの範囲が設定されます。最高のパフォーマンスを得るためには、4Kの倍数にする必要があります。デフォルト値は、プールの作成時にモニタ設定オプションosd\_pool\_erasure\_code\_stripe\_unitから取得されます。このプロファイルを使用するプールの「stripe\_width」は、データチャンクの数にこの「stripe\_unit」を掛けた値になります。

## KEY=VALUE

選択したイレージャコードプラグインに固有のオプションのキー/値のペア。

## --force

オプション。既存のプロファイルを同じ名前で上書きし、4K以外で配置されたstripe\_unitを設定できるようにします。

## 19.3.2 イレージャコードプロファイルの削除

次のコマンドは、NAMEで指定したイレージャコードプロファイルを削除します。

```
root # ceph osd erasure-code-profile rm NAME
```



### 重要

プロファイルがプールによって参照されている場合、削除は失敗します。

## 19.3.3 イレージャコードプロファイルの詳細の表示

次のコマンドは、NAMEで指定したイレージャコードプロファイルの詳細を表示します。

```
root # ceph osd erasure-code-profile get NAME
```

## 19.3.4 イレージャコードプロファイルの一覧

次のコマンドは、すべてのイレージャコードプロファイルの名前を一覧にします。

```
root # ceph osd erasure-code-profile ls
```

## 19.4 イレージャコーディングプールをRADOS Block Deviceとしてマーク付け

ECプールにRBDプールとしてマークを付けるには、適切にタグを付けます。

```
cephuser@adm > ceph osd pool application enable rbd ec_pool_name
```

RBDはECプールにイメージの「データ」を保存できます。ただし、イメージのヘッダとメタデータは引き続き複製プールに保存する必要があります。このための「rbd」という名前のプールがあると想定した場合、次のコマンドを使用します。

```
cephuser@adm > rbd create rbd/image_name --size 1T --data-pool ec_pool_name
```

このイメージは他のイメージと同じように通常の方法で使用できますが、すべてのデータは「rbd」プールではなく ec\_pool\_name プールに保存される点が異なります。

## 20 RADOS Block Device

ブロックとは連続するバイトのことで、たとえば4MBブロックのデータなどです。ブロックベースのストレージインタフェースは、ハードディスク、CD、フロッピーディスクなどの回転型媒体にデータを保存する最も一般的な方法です。Block Deviceインタフェースはあらゆる場所で利用されているため、仮想ブロックデバイスは、Cephのような大容量データストレージシステムを操作するための理想的な候補です。

Ceph Block Deviceは物理リソースを共有でき、サイズの変更が可能です。データはCephクラスタ内の複数のOSD上にストライプされて保存されます。Ceph Block Deviceは、スナップショットの作成、レプリケーション、整合性などのRADOSの機能を利用します。CephのRBD (RADOS Block Device)は、カーネルモジュールまたはlibrbdライブラリを使用してOSDと対話します。



図 20.1: RADOSプロトコル

Cephのブロックデバイスは、高いパフォーマンスと無限のスケーラビリティをカーネルモジュールに提供します。これらは、QEMUなどの仮想化ソリューションや、OpenStackなど、libvirtに依存するクラウドベースのコンピューティングシステムをサポートします。同じクラスタを使用して、Object Gateway、CephFS、およびRADOS Block Deviceを同時に運用できます。

### 20.1 Block Deviceのコマンド

**rbd**コマンドを使用して、Block Deviceイメージを作成、一覧、イントロスペクト、および削除できます。さらに、イメージのクローン作成、スナップショットの作成、スナップショットへのイメージのロールバック、スナップショットの表示などの操作にも使用できます。

#### 20.1.1 複製プールでのBlock Deviceイメージの作成

Block Deviceをクライアントに追加する前に、既存のプール内に、関連するイメージを作成する必要があります(第18章「ストレージプールの管理」を参照)。

```
cephuser@adm > rbd create --size MEGABYTES POOL-NAME/IMAGE-NAME
```

たとえば、「mypool」という名前のプールに情報を保存する「myimage」という名前の1GBのイメージを作成するには、次のコマンドを実行します。

```
cephuser@adm > rbd create --size 1024 mypool/myimage
```



## ヒント: イメージサイズの単位

サイズの単位のショートカット(「G」または「T」)を省略した場合、イメージのサイズはメガバイト単位になります。ギガバイトまたはテラバイトを指定するには、サイズの数字の後に「G」または「T」を使用します。

## 20.1.2 イレージャコーディングプールでのBlock Deviceイメージの作成

Block DeviceイメージのデータをEC (イレージャコーディング)プールに直接保存できます。RADOS Block Deviceイメージは、「データ」部分と「メタデータ」部分で構成されます。ECプールには、RADOS Block Deviceイメージの「データ」部分のみを保存できます。プールは`overwrite`フラグが**true**に設定されている必要があります。このように設定できるのは、プールが保存されているすべてのOSDがBlueStoreを使用している場合のみです。

ECプールにイメージの「メタデータ」の部分を保存することはできません。`rbd create`コマンドの`--pool=`オプションを使用してイメージのメタデータを保存する複製プールを指定するか、イメージ名のプレフィックスとして`pool/`を指定できます。

ECプールを作成します。

```
cephuser@adm > ceph osd pool create EC_P00L 12 12 erasure
cephuser@adm > ceph osd pool set EC_P00L allow_ec_overwrites true
```

メタデータを保存する複製プールを指定します。

```
cephuser@adm > rbd create IMAGE_NAME --size=1G --data-pool EC_P00L --pool=P00L
```

または:

```
cephuser@adm > rbd create P00L/IMAGE_NAME --size=1G --data-pool EC_P00L
```

## 20.1.3 Block Deviceイメージの一覧

「mypool」という名前のプール内のBlock Deviceを一覧にするには、次のコマンドを実行します。

```
cephuser@adm > rbd ls mypool
```

### 20.1.4 イメージ情報の取得

「mypool」という名前のプール内のイメージ「myimage」から情報を取得するには、次のコマンドを実行します。

```
cephuser@adm > rbd info mypool/myimage
```

### 20.1.5 Block Deviceイメージのサイズの変更

RADOS Block Deviceイメージはシンプロビジョニングされます。つまり、そこにデータを保存し始めるまでは、実際に物理ストレージを使用しません。ただし、`--size`オプションで設定する最大容量があります。イメージの最大サイズを増やす(または減らす)場合、次のコマンドを実行します。

```
cephuser@adm > rbd resize --size 2048 POOL_NAME/IMAGE_NAME # to increase  
cephuser@adm > rbd resize --size 2048 POOL_NAME/IMAGE_NAME --allow-shrink # to decrease
```

### 20.1.6 Block Deviceイメージの削除

「mypool」という名前のプール内にあるイメージ「myimage」に対応するBlock Deviceを削除するには、次のコマンドを実行します。

```
cephuser@adm > rbd rm mypool/myimage
```

## 20.2 マウントとアンマウント

RADOS Block Deviceを作成した後は、他のディスクデバイスと同じように使用できます。デバイスをフォーマットし、マウントしてファイルを交換できるようにし、完了したらアンマウントできます。

デフォルトでは、`rbd`コマンドはCephの`admin`ユーザアカウントを使用してクラスタにアクセスします。このアカウントはクラスタに対する完全な管理アクセス権限を持ちます。そのため、Linuxワークステーションに`root`でログインした場合と同様に、誤って損害を発生させてしまうリスクが発生します。したがって、特権を制限したユーザアカウントを作成して、RADOS Block Deviceの通常の読み込み/書き込みアクセスに使用することが望ましいです。

## 20.2.1 Cephユーザアカウントの作成

Ceph Manager、Ceph Monitor、Ceph OSDのキーパビリティを使用して新しいユーザアカウントを作成するには、**ceph**コマンドと**auth get-or-create**サブコマンドを使用します。

```
cephuser@adm > ceph auth get-or-create client.ID mon 'profile rbd' osd 'profile profile
name \
[pool=pool-name] [, profile ...]' mgr 'profile rbd [pool=pool-name]'
```

たとえば、vmsプールへの読み込み/書き込みアクセスと、imagesプールへの読み込み専用アクセスができる、qemuという名前のユーザを作成するには、次のコマンドを実行します。

```
ceph auth get-or-create client.qemu mon 'profile rbd' osd 'profile rbd pool=vms, profile
rbd-read-only pool=images' \
mgr 'profile rbd pool=images'
```

**ceph auth get-or-create**コマンドは特定のユーザ用のキーリングを出力します。また、キーリングを`/etc/ceph/ceph.client.ID.keyring`に書き込むことができます。



### 注記

**rbd**コマンドを使用する場合、オプションの**--id ID**引数を与えることでユーザIDを指定できます。

Cephユーザアカウントの管理の詳細については、[第30章「cephxを使用した認証」](#)を参照してください。

## 20.2.2 ユーザ認証

ユーザ名を指定するには、**--id user-name**を使用します。**cephx**認証を使用する場合は、秘密を指定する必要もあります。秘密は、キーリング、または秘密が含まれるファイルから取得できます。

```
cephuser@adm > rbd device map --pool rbd myimage --id admin --keyring /path/to/keyring
```

あるいは、

```
cephuser@adm > rbd device map --pool rbd myimage --id admin --keyfile /path/to/file
```

## 20.2.3 RADOS Block Deviceを使用するための準備

1. Cephクラスタに、マップするディスクイメージが存在するプールが含まれることを確認します。プールは`mypool`、イメージは`myimage`という名前であると想定します。

```
cephuser@adm > rbd list mypool
```

2. イメージを新しいBlock Deviceにマップします。

```
cephuser@adm > rbd device map --pool mypool myimage
```

3. すべてのマップ済みデバイスを一覧にします。

```
cephuser@adm > rbd device list
id pool  image  snap device
0  mypool myimage -    /dev/rbd0
```

作業対象のデバイスは`/dev/rbd0`です。



### ヒント: RBDデバイスのパス

`/dev/rbdDEVICE_NUMBER`の代わりに、永続的なデバイスパスとして`/dev/rbd/POOL_NAME/IMAGE_NAME`を使用できます。例:

```
/dev/rbd/mypool/myimage
```

4. `/dev/rbd0`デバイス上にXFSファイルシステムを作成します。

```
root # mkfs.xfs /dev/rbd0
log stripe unit (4194304 bytes) is too large (maximum is 256KiB)
log stripe unit adjusted to 32KiB
meta-data=/dev/rbd0          isize=256    agcount=9, agsize=261120 blks
=                               sectsz=512   attr=2, projid32bit=1
=                               crc=0      finobt=0
data      =                   bsize=4096   blocks=2097152, imaxpct=25
=                               sunit=1024  swidth=1024 blks
naming    =version 2          bsize=4096   ascii-ci=0 ftype=0
log       =internal log      bsize=4096   blocks=2560, version=2
=                               sectsz=512   sunit=8 blks, lazy-count=1
realtime  =none              extsz=4096   blocks=0, rtextents=0
```

5. 使用するマウントポイントに`/mnt`を置き換えてから、デバイスをマウントして正しくマウントされたかを確認します。

```
root # mount /dev/rbd0 /mnt
```

```
root # mount | grep rbd0
/dev/rbd0 on /mnt type xfs (rw,relatime,attr2,inode64,sunit=8192,...
```

これで、ローカルディレクトリと同じように、このデバイスとの間でデータを移動できます。



## ヒント: RBDデバイスのサイズを増やす

RBDデバイスのサイズが十分ではなくなった場合、簡単にサイズを増やすことができます。

1. RBDイメージのサイズを、たとえば10GBに増やします。

```
cephuser@adm > rbd resize --size 10000 mypool/myimage
Resizing image: 100% complete...done.
```

2. デバイスの新しいサイズ全体を使用するようファイルシステムを拡張します。

```
root # xfs_growfs /mnt
[...]
data blocks changed from 2097152 to 2560000
```

6. デバイスへのアクセスが終わったら、デバイスをマップ解除してアンマウントできます。

```
cephuser@adm > rbd device unmap /dev/rbd0
root # umount /mnt
```



## ヒント: 手動によるマウントとアンマウント

ブート後にRBDのマッピングとマウントを行い、シャットダウン前にRBDをアンマウントするプロセスをスムーズにするため、**rbdmmap**スクリプトと**systemd**ユニットが提供されています。20.2.4項「**rbdmmap**: ブート時のRBDデバイスのマッピング」を参照してください。

## 20.2.4 **rbdmmap**: ブート時のRBDデバイスのマッピング

**rbdmmap**は、1つ以上のRBDイメージに対する**rbdm map**および**rbdm device unmap**の操作を自動化するシェルスクリプトです。このスクリプトはいつでも手動で実行できますが、ブート時にRBDイメージを自動的にマッピングしてマウント(シャットダウン時にはアンマウントしてマッピング解除)するのが主な利点です。これはInitシステムによってトリガされます。このために、**ceph-common**パッケージに**systemd**のユニットファイルである**rbdmmap.service**が含まれています。

このスクリプトは引数を1つ取り、**map**または**unmap**のどちらかを指定できます。どちらの場合も、スクリプトは設定ファイルを解析します。デフォルトは**/etc/ceph/rbdmap**ですが、環境変数**RBDMAPFILE**で上書きできます。設定ファイルの各行が、マッピングまたはマッピング解除する1つのRBDイメージに対応します。

構成ファイルは次のような形式になっています。

```
image_specification rbd_options
```

image\_specification

プール内のイメージのパス。 pool\_name/image\_nameとして指定します。

rbd\_options

基礎となる**rbdm device map**コマンドに渡されるパラメータのオプションのリスト。これらのパラメータとその値をコンマ区切り文字列として指定する必要があります。次に例を示します。

```
PARAM1=VAL1,PARAM2=VAL2,...
```

次の例では、**rbdmmap**スクリプトで次のコマンドを実行します。

```
cephuser@adm > rbd device map POOL_NAME/IMAGE_NAME --PARAM1 VAL1 --PARAM2 VAL2
```

次の例では、ユーザ名とキーリングを対応する秘密とともに指定する方法を確認できます。

```
cephuser@adm > rbdmap device map mypool/myimage id=rbd_user,keyring=/etc/ceph/ceph.client.rbd.keyring
```

**rbdmmap map**として実行すると、設定ファイルを解析し、指定されているRBDイメージそれぞれに対して、最初にイメージをマッピング(**rbdm device map**を使用)、次にイメージをマウントしようと試みます。

**rbdmmap unmap**として実行すると、設定ファイルに一覧にされているイメージがアンマウントされてマッピング解除されます。

**rbdmmap unmap-all**は、設定ファイルに一覧にされているかどうかに関係なく、現在マップされているRBDイメージをすべてアンマウントし、その後マップ解除しようと試みます。

成功した場合、イメージは**rbd device map**操作によって/dev/rbdXデバイスにマップされます。この時点でudevルールがトリガされ、実際にマップされたデバイスを指すフレンドリデバイス名のシンボリックリンク/dev/rbd/pool\_name/image\_nameが作成されます。

正常にマウントおよびアンマウントするには、/etc/fstabに「フレンドリ」デバイス名に対応するエントリが必要です。RBDイメージの/etc/fstabエントリを記述する場合、「noauto」(または「nofail」)マウントオプションを指定します。**rbdmmap.service**は一般的にブートシーケンスのかなり遅い段階でトリガされるため、このオプションを指定することによって、Initシステムが、対象デバイスがまだ存在しない早すぎるタイミングでデバイスをマウントしないようにします。

**rbd**オプションの完全なリストについては、**rbd**のマニュアルページ(**man 8 rbd**)を参照してください。

**rbdmmap**の使用法の例については、**rbdmmap**のマニュアルページ(**man 8 rbdmap**)を参照してください。

## 20.2.5 RBDデバイスのサイズを増やす

RBDデバイスのサイズが十分ではなくなった場合、簡単にサイズを増やすことができます。

1. RBDイメージのサイズを、たとえば10GBに増やします。

```
cephuser@adm > rbd resize --size 10000 mypool/myimage
Resizing image: 100% complete...done.
```

2. デバイスの新しいサイズ全体を使用するようファイルシステムを拡張します。

```
root # xfs_growfs /mnt
[...]
data blocks changed from 2097152 to 2560000
```

## 20.3 スナップショット

RBDのスナップショットは、RADOS Block Deviceイメージのスナップショットです。スナップショットにより、イメージの状態の履歴を保持します。Cephはスナップショットの階層化もサポートしており、VMイメージのクローンを素早く簡単に作成できます。**rbid**コマンド、およびさまざまな高レベルのインタフェース(QEMU、**libvirt**、OpenStack、CloudStackなど)を使用したBlock Deviceのスナップショットをサポートしています。



## 注記

イメージのスナップショットを作成する前に、入出力操作を停止し、保留中の書き込みをすべてフラッシュする必要があります。イメージにファイルシステムが含まれる場合、スナップショットの作成時に、そのファイルシステムが整合性のある状態である必要があります。

### 20.3.1 cephxの有効化と設定

cephxが有効な場合、ユーザ名またはIDと、そのユーザに対応する鍵が含まれるキーリングのパスを指定する必要があります。詳しくは「[第30章「cephxを使用した認証」](#)」を参照してください。以降のパラメータを再入力せずに済むよう、`CEPH_ARGS`環境変数を追加することもできます。

```
cephuser@adm > rbd --id user-ID --keyring=/path/to/secret commands
cephuser@adm > rbd --name username --keyring=/path/to/secret commands
```

例:

```
cephuser@adm > rbd --id admin --keyring=/etc/ceph/ceph.keyring commands
cephuser@adm > rbd --name client.admin --keyring=/etc/ceph/ceph.keyring commands
```



## ヒント

`CEPH_ARGS`環境変数にユーザと秘密を追加して、毎回入力しなくて済むようにします。

### 20.3.2 スナップショットの基本

次の手順では、コマンドラインで**rbid**を使用して、スナップショットを作成、一覧、および削除する方法を説明します。

#### 20.3.2.1 スナップショットの作成

**rbid**を使用してスナップショットを作成するには、`snap create`オプション、プール名、およびイメージ名を指定します。

```
cephuser@adm > rbd --pool pool-name snap create --snap snap-name image-name
cephuser@adm > rbd snap create pool-name/image-name@snap-name
```

例:

```
cephuser@adm > rbd --pool rbd snap create --snap snapshot1 image1
cephuser@adm > rbd snap create rbd/image1@snapshot1
```

### 20.3.2.2 スナップショットの一覧

イメージのスナップショットを一覧にするには、プール名とイメージ名を指定します。

```
cephuser@adm > rbd --pool pool-name snap ls image-name
cephuser@adm > rbd snap ls pool-name/image-name
```

例:

```
cephuser@adm > rbd --pool rbd snap ls image1
cephuser@adm > rbd snap ls rbd/image1
```

### 20.3.2.3 スナップショットのロールバック

**rbd**を使用して特定のスナップショットにロールバックするには、`snap rollback`オプション、プール名、イメージ名、およびスナップショット名を指定します。

```
cephuser@adm > rbd --pool pool-name snap rollback --snap snap-name image-name
cephuser@adm > rbd snap rollback pool-name/image-name@snap-name
```

例:

```
cephuser@adm > rbd --pool pool1 snap rollback --snap snapshot1 image1
cephuser@adm > rbd snap rollback pool1/image1@snapshot1
```



## 注記

イメージをスナップショットにロールバックすることは、イメージの現在のバージョンをスナップショットのデータで上書きすることを意味します。ロールバックの実行にかかる時間は、イメージのサイズに応じて長くなります。イメージをスナップショットに「ロールバック」するよりもスナップショットから「クローンを作成する方が高速」であり、以前の状態に戻す場合はこの方法をお勧めします。

### 20.3.2.4 スナップショットの削除

**rbd**を使用してスナップショットを削除するには、`snap rm`オプション、プール名、イメージ名、およびユーザ名を指定します。

```
cephuser@adm > rbd --pool pool-name snap rm --snap snap-name image-name
cephuser@adm > rbd snap rm pool-name/image-name@snap-name
```

例:

```
cephuser@adm > rbd --pool pool1 snap rm --snap snapshot1 image1
cephuser@adm > rbd snap rm pool1/image1@snapshot1
```



## 注記

Ceph OSDはデータを非同期で削除するので、スナップショットを削除してもディスク領域はすぐには解放されません。

### 20.3.2.5 スナップショットの消去

**rbd**を使用してイメージのすべてのスナップショットを削除するには、**snap purge**オプションとイメージ名を指定します。

```
cephuser@adm > rbd --pool pool-name snap purge image-name
cephuser@adm > rbd snap purge pool-name/image-name
```

例:

```
cephuser@adm > rbd --pool pool1 snap purge image1
cephuser@adm > rbd snap purge pool1/image1
```

### 20.3.3 スナップショットの階層化

Cephでは、Block DeviceスナップショットのCOW (コピーオンライト)クローンを複数作成できます。スナップショットの階層化により、Ceph Block Deviceのクライアントはイメージを非常に素早く作成できます。たとえば、Linux VMが書き込まれたBlock Deviceイメージを作成してから、そのイメージのスナップショットを作成し、スナップショットを保護して、コピーオンライトクローンを必要な数だけ作成できます。スナップショットは読み込み専用なので、スナップショットのクローンを作成することでセマンティクスが簡素化され、クローンを素早く作成できます。



## 注記

次のコマンドラインの例で使われている「親」および「子」という用語は、Ceph Block Deviceのスナップショット(親)と、そのスナップショットから作成された対応するクローンイメージ(子)を意味します。

クローンイメージ(子)にはその親イメージへの参照が保存されており、これによってクローンイメージから親のスナップショットを開いて読み込むことができます。

スナップショットのCOWクローンは、他のCeph Block Deviceイメージとまったく同じように動作します。クローンイメージに対して読み書きを行ったり、クローンを作成したり、サイズを変更したりできます。クローンイメージに特別な制約はありません。ただし、スナップショットのコピーオンライトクローンはスナップショットを参照するので、クローンを作成する前に「必ず」スナップショットを保護する必要があります。



### 注記: `--image-format 1`はサポートされない

非推奨の`rbd create --image-format 1`オプションを使用して作成されたイメージのスナップショットを作成することはできません。Cephでサポートされているのは、デフォルトの「format 2」のイメージのクローン作成のみです。

#### 20.3.3.1 階層化の基本事項

Ceph Block Deviceの階層化は簡単なプロセスです。まずイメージを用意する必要があります。続いて、イメージのスナップショットを作成し、スナップショットを保護する必要があります。これらの手順を実行した後、スナップショットのクローンの作成を開始できます。

クローンイメージは親スナップショットへの参照を持ち、プールID、イメージID、およびスナップショットIDを含みます。プールIDが含まれることは、あるプールから別のプール内のイメージへスナップショットのクローンを作成できることを意味します。

- 「イメージテンプレート」：Block Deviceの階層化の一般的な使用事例は、マスタイメージと、クローンのテンプレートとして機能するスナップショットを作成することです。たとえば、Linux配布パッケージ(たとえば、SUSE Linux Enterprise Server)のイメージを作成して、そのスナップショットを作成できます。定期的にイメージを更新して新しいスナップショットを作成できます(たとえば、`zypper ref && zypper patch`の後に`rbd snap create`を実行します)。イメージが完成したら、いずれかのスナップショットのクローンを作成できます。
- 「拡張テンプレート」：より高度な使用事例として、ベースイメージより多くの情報を提供するテンプレートイメージを拡張することがあります。たとえば、イメージ(VMテンプレート)のクローンを作成して、他のソフトウェア(たとえば、データベース、コンテンツ管理システム、分析システム)をインストールしてから、拡張イメージのスナップショットを作成でき、このスナップショットそのものをベースイメージと同じ方法で更新できます。

- 「テンプレートプール」: Block Deviceの階層化を使用する方法の1つが、テンプレートとして機能するマスタイメージと、それらのテンプレートの各スナップショットが含まれるプールを作成することです。その後、読み込み専用特権をユーザに拡張し、プール内での書き込みまたは実行の能力を持たなくても、スナップショットのクローンを作成できるようにします。
- 「イメージのマイグレーション/回復」: Block Deviceの階層化を使用する方法の1つが、あるプールから別のプールへデータを移行または回復することです。

### 20.3.3.2 スナップショットの保護

クローンは親スナップショットにアクセスします。ユーザが誤って親スナップショットを削除すると、すべてのクローンが壊れます。データの損失を防ぐため、クローンを作成する前に、スナップショットを保護する必要があります。

```
cephuser@adm > rbd --pool pool-name snap protect \
--image image-name --snap snapshot-name
cephuser@adm > rbd snap protect pool-name/image-name@snapshot-name
```

例:

```
cephuser@adm > rbd --pool pool1 snap protect --image image1 --snap snapshot1
cephuser@adm > rbd snap protect pool1/image1@snapshot1
```



### 注記

保護されたスナップショットは削除できません。

### 20.3.3.3 スナップショットのクローンの作成

スナップショットのクローンを作成するには、親プール、イメージ、スナップショット、子プール、およびイメージ名を指定する必要があります。クローンを作成する前に、スナップショットを保護する必要があります。

```
cephuser@adm > rbd clone --pool pool-name --image parent-image \
--snap snap-name --dest-pool pool-name \
--dest child-image
cephuser@adm > rbd clone pool-name/parent-image@snap-name \
pool-name/child-image-name
```

例:

```
cephuser@adm > rbd clone pool1/image1@snapshot1 pool1/image2
```



## 注記

あるプールから別のプール内のイメージへスナップショットのクローンを作成できます。たとえば、一方のプール内に読み込み専用のイメージとスナップショットをテンプレートとして維持しておき、別のプール内に書き込み可能クローンを維持できます。

### 20.3.3.4 スナップショットの保護の解除

スナップショットを削除するには、まず保護を解除する必要があります。また、クローンから参照されているスナップショットは削除「できません」。スナップショットを削除する前に、スナップショットの各クローンをフラット化する必要があります。

```
cephuser@adm > rbd --pool pool-name snap unprotect --image image-name \
--snap snapshot-name
cephuser@adm > rbd snap unprotect pool-name/image-name@snapshot-name
```

例:

```
cephuser@adm > rbd --pool pool1 snap unprotect --image imagel --snap snapshot1
cephuser@adm > rbd snap unprotect pool1/imagel@snapshot1
```

### 20.3.3.5 スナップショットの子の一覧

スナップショットの子を一覧にするには、次のコマンドを実行します。

```
cephuser@adm > rbd --pool pool-name children --image image-name --snap snap-name
cephuser@adm > rbd children pool-name/image-name@snapshot-name
```

例:

```
cephuser@adm > rbd --pool pool1 children --image imagel --snap snapshot1
cephuser@adm > rbd children pool1/imagel@snapshot1
```

### 20.3.3.6 クローンイメージのフラット化

クローンイメージは親スナップショットへの参照を保持しています。子クローンから親スナップショットへの参照を削除する場合、スナップショットからクローンへ情報をコピーすることによって効果的にイメージを「フラット化」します。クローンのフラット化にかかる時間は、スナップショットのサイズに応じて長くなります。スナップショットを削除するには、まず子イメージをフラット化する必要があります。

```
cephuser@adm > rbd --pool pool-name flatten --image image-name
cephuser@adm > rbd flatten pool-name/image-name
```

例:

```
cephuser@adm > rbd --pool pool1 flatten --image image1
cephuser@adm > rbd flatten pool1/image1
```



## 注記

フラット化されたイメージにはスナップショットからの情報がすべて含まれるため、階層化されたクローンよりも多くのストレージ領域を使用します。

## 20.4 RBDイメージのミラーリング

RBDイメージを2つのCephクラスタ間で非同期でミラーリングできます。この機能には2つのモードがあります。

### ジャーナルベース

このモードは、RBDイメージのジャーナリング機能を使用して、クラスタ間でクラッシュコンシステントなレプリケーションを保証します。RBDイメージに対する書き込みが発生すると、まず関連するジャーナルに記録されてから、イメージが実際に変更されます。remoteクラスタはジャーナルを読み込み、イメージのローカルコピーに更新内容を再現します。RBDイメージのジャーナリング機能を使用すると、RBDイメージに1回書き込むたびに2回の書き込みが発生するため、書き込みに伴う遅延が約2倍となることが見込まれます。

### スナップショットベース

このモードは、RBDイメージのミラースナップショットを使用して、クラッシュコンシステントなRBDイメージをクラスタ間で複製します。このミラースナップショットはスケジュールに沿って定期的に作成するか、手動で作成します。remoteクラスタは2つのミラースナップショットの間でなんらかのデータかメタデータが更新されているかを判定し、イメージのローカルコピーに差分をコピーします。RBDのfast-diffイメージ機能のおかげで、更新されたデータブロックは素早く計算できます。RBDイメージ全体をスキャンする必要はありません。このモードには時間的な整合性がないため、フェールオーバーシナリオの際に利用するには事前にスナップショットの差分全体を同期する必要があります。部分的に適用されたスナップショット差分については、使用する前に最新の完全に同期されたスナップショットまでロールバックします。

ミラーリングは、ピアクラスタ内のプールごとに設定します。ジャーナルベースのミラーリングだけを使用している場合、プール内の特定のイメージサブセットに設定することも、プール内のすべてのイメージを自動的にミラーリングするように設定することもできます。ミラーリングは`rbd`コマンドを使用して設定します。`rbd-mirror`デーモンは、`remote`のピアクラスタからイメージの更新を取得して、`local`クラスタ内のイメージに適用する処理を受け持ちます。

レプリケーションに対する要望に応じて、RBDミラーリングは単方向レプリケーション用または双方向レプリケーション用に設定できます。

### 単方向レプリケーション

データがプライマリクラスタからセカンダリクラスタにミラーリングされるだけであれば、`rbd-mirror`デーモンはセカンダリクラスタ上でのみ実行されます。

### 双方向レプリケーション

データが、あるクラスタのプライマリイメージから別のクラスタの非プライマリイメージにミラーリングされる場合(逆も同様)、`rbd-mirror`デーモンは両方のクラスタで実行されます。

## ！ 重要

`rbd-mirror`デーモンの各インスタンスは、`local` Cephクラスタと`remote` Cephクラスタへ同時に接続できる必要があります。たとえば、すべてのMonitorホストとOSDホストに接続できる必要があります。さらに、ミラーリングのワークロードを扱うため、ネットワークの2つのデータセンター間には十分な帯域幅が必要です。

## 20.4.1 プールの設定

次の手順では、`rbd`コマンドを使用してミラーリングを設定するための基本的な管理タスクを実行する方法を説明します。ミラーリングは、Cephクラスタ内のプールごとに設定します。これらのプール設定手順は、両方のピアクラスタで実行する必要があります。これらの手順では、わかりやすくするため、`local`および`remote`という名前の2つのクラスタが1つのホストからアクセス可能であることを想定しています。

異なるCephクラスタに接続する方法の詳細については、`rbd`のマニュアルページ(`man 8 rbd`)を参照してください。



## ヒント: 複数のクラスタ

以下の例におけるクラスタ名は、同名のCeph設定ファイルである`/etc/ceph/remote.conf`と、同名のCephキーリングファイルである`/etc/ceph/remote.client.admin.keyring`に対応しています。

### 20.4.1.1 プールのミラーリングの有効化

プールのミラーリングを有効にするには、**`mirror pool enable`**サブコマンド、プール名、およびミラーリングモードを指定します。ミラーリングモードは`pool`または`image`にすることができます。

#### **pool**

ジャーナリング機能が有効な、プール内のすべてのイメージをミラーリングします。

#### **image**

各イメージに対して明示的にミラーリングを有効にする必要があります。詳細については、20.4.2.1項「イメージミラーリングの有効化」を参照してください。

例:

```
cephuser@adm > rbd --cluster local mirror pool enable POOL_NAME pool
cephuser@adm > rbd --cluster remote mirror pool enable POOL_NAME pool
```

### 20.4.1.2 ミラーリングの無効化

プールのミラーリングを無効にするには、**`mirror pool disable`**サブコマンドとプール名を指定します。この方法でプールのミラーリングを無効にした場合、ミラーリングを明示的に有効にしたイメージ(プール内)のミラーリングも無効になります。

```
cephuser@adm > rbd --cluster local mirror pool disable POOL_NAME
cephuser@adm > rbd --cluster remote mirror pool disable POOL_NAME
```

### 20.4.1.3 ピアのブートストラップ処理

`rbd-mirror`デーモンがピアクラスタを検出するためには、ピアをプールに登録し、ユーザアカウントを作成する必要があります。このプロセスは`rbd`とともに**`mirror pool peer bootstrap create`**と**`mirror pool peer bootstrap import`**のコマンドを使用することで自動化できます。

```
cephuser@local > rbd mirror pool peer bootstrap create \
  [--site-name LOCAL_SITE_NAME] POOL_NAME
```

```
cephuser@local > rbd --cluster local mirror pool peer bootstrap create --site-name local
image-pool
eyJmc2lkIjo0Y0Y1MjgyZGI0Yjg5ODU0NTk2LTgwTgtMzIwYzFmYzM5NmYzIiwia2xpZW50X2lkIjoicmJkLWlpcnJ
\
joiQVFBUnczOWQwdkhvQmhBQVlMM1I4RmR5dHJQU50bkFTZ0l0TVE9PSIsIm1vbl9ob3N0Ijo0Y0Y1MjgyZGI0Yjg5ODU0NTk2LTgwTgtMzIwYzFmYzM5NmYzIiwia2xpZW50X2lkIjoicmJkLWlpcnJ
```

```
rbd mirror pool peer bootstrap import \
  [--site-name LOCAL_SITE_NAME] \
  [--direction DIRECTION] \
  POOL_NAME TOKEN_PATH
```

LOCAL\_SITE\_NAME

DIRECTION

POOL NAME

TOKEN PATH

たとえば、remoteクラスタで次のコマンドを実行します。

```
cephuser@remote > cat <<EOF > token
eyJmc2lkIjoj0WY1MjgyZGIyYjg5ODS00NTk2LTgw0TgtMzIwYzFmYzM5NmYzIiwiaY2xpZW50X2lkIjoicmJkLW1pcnJ
```

```
EOF
```

```
cephuser@adm > rbd --cluster remote mirror pool peer bootstrap import \  
--site-name remote image-pool token
```

## 20.4.1.4 クラスタピアの手動追加

20.4.1.3項「ピアのブートストラップ処理」に記載したピアのブートストラップ方法の代わりに、手動でピアを指定することもできます。ミラーリングを実行するには、リモートの`rbd-mirror`デーモンがローカルクラスタにアクセスする必要があります。リモートの`rbd-mirror`デーモンが使用する新しいローカルCephユーザを作成します。次の例では`rbd-mirror-peer`です。

```
cephuser@adm > ceph auth get-or-create client.rbd-mirror-peer \  
mon 'profile rbd' osd 'profile rbd'
```

`rbd`コマンドにより、ミラーリングピアのCephクラスタを追加するには、次の構文を使用します。

```
rbd mirror pool peer add POOL_NAME CLIENT_NAME@CLUSTER_NAME
```

例:

```
cephuser@adm > rbd --cluster site-a mirror pool peer add image-pool client.rbd-mirror-  
peer@site-b  
cephuser@adm > rbd --cluster site-b mirror pool peer add image-pool client.rbd-mirror-  
peer@site-a
```

デフォルトでは、`rbd-mirror`デーモンが、`/etc/ceph/.CLUSTER_NAME.conf`に置かれたCeph設定ファイルにアクセスする必要があります。この設定ファイルにはピアクラスタのMONのIPアドレスと、デフォルトまたはカスタムのキーリング検索パスに配置された`CLIENT_NAME`という名前のクライアント用キーリングが含まれます。キーリング検索パスの例は、`/etc/ceph/CLUSTER_NAME.CLIENT_NAME.keyring`などです。

もしくは、ピアクラスタのMONとクライアントキーの両方または一方を、ローカルのCeph設定キーストアに安全に保存することもできます。ミラーリングピアを追加する際にピアクラスタの接続属性を指定するには、`--remote-mon-host`オプションと`--remote-key-file`オプションを使用します。例:

```
cephuser@adm > rbd --cluster site-a mirror pool peer add image-pool \  
client.rbd-mirror-peer@site-b --remote-mon-host 192.168.1.1,192.168.1.2 \  
--remote-key-file /PATH/TO/KEY_FILE  
cephuser@adm > rbd --cluster site-a mirror pool info image-pool --all  
Mode: pool
```

Peers:				
UUID	NAME	CLIENT	MON_HOST	KEY
587b08db...	site-b	client.rbd-mirror-peer	192.168.1.1,192.168.1.2	AQAeuZdb...

### 20.4.1.5 クラスピアの削除

ミラーリングピアクラスタを削除するには、`mirror pool peer remove`サブコマンド、プール名、およびピアのUUID (`rbd mirror pool info`コマンドで参照可能)を指定します。

```
cephuser@adm > rbd --cluster local mirror pool peer remove POOL_NAME \
55672766-c02b-4729-8567-f13a66893445
cephuser@adm > rbd --cluster remote mirror pool peer remove POOL_NAME \
60c0e299-b38f-4234-91f6-eed0a367be08
```

### 20.4.1.6 データプール

宛先クラスタにイメージを作成する場合、`rbd-mirror`は次のようにデータプールを選択します。

- 宛先クラスタにデフォルトのデータプールが設定されている場合は、そのプールを使用します(設定には、`rbd_default_data_pool`設定オプションを使用します)。
- デフォルトのデータプールが設定されていない場合、ソースイメージが別のデータプールを使用しており、宛先クラスタに同じ名前のプールが存在するなら、そのプールを使用します。
- これら2つの条件が満たされない場合、データプールは設定されません。

## 20.4.2 RBDイメージの設定

プール設定と異なり、イメージ設定はミラーリングピアの1つのCephクラスタのみで実行する必要があります。

ミラーリングされたRBDイメージは、「プライマリ」または「非プライマリ」のいずれかとして指定されます。これはイメージのプロパティであり、プールのプロパティではありません。非プライマリとして指定されたイメージは変更できません。

イメージに対して初めてミラーリングを有効にすると、イメージは自動的にプライマリに昇格します(プールのミラーモードが「pool」で、イメージのジャーナリング機能が有効な場合、ミラーリングは暗黙的に有効になります。または、`rbd`コマンドによって明示的に有効にします(20.4.2.1項「イメージミラーリングの有効化」を参照してください)。

### 20.4.2.1 イメージミラーリングの有効化

ミラーリングがimageモードで設定されている場合、プール内の各イメージに対して明示的にミラーリングを有効にする必要があります。`rbd`コマンドを使用して特定のイメージのミラーリングを有効にするには、`mirror image enable`サブコマンドと共にプール名とイメージ名を指定します。

```
cephuser@adm > rbd --cluster local mirror image enable \  
POOL_NAME/IMAGE_NAME
```

イメージのミラーリングモードは、`journal`または`snapshot`のどちらかを使用できます。

#### journal(デフォルト)

`journal`モードに設定した場合、ミラーリング処理にRBDイメージのジャーナリング機能を使用して、イメージ内容の複製を行います。イメージに対するRBDイメージのジャーナリング機能が有効化されていなかった場合、自動的に有効化されます。

#### スナップショット

`snapshot`モードに設定した場合、ミラーリング処理にRBDイメージのミラースナップショット機能を使用して、イメージ内容の複製を行います。有効化した場合、最初のミラースナップショットが自動的に作成されます。RBDイメージのミラースナップショットを`rbd`コマンドにより追加で作成することもできます。

例:

```
cephuser@adm > rbd --cluster local mirror image enable image-pool/image-1 snapshot  
cephuser@adm > rbd --cluster local mirror image enable image-pool/image-2 journal
```

### 20.4.2.2 イメージジャーナリング機能の有効化

RBDのミラーリングは、RBDのジャーナリング機能を使用して、複製イメージが常にクラッシュコンシステントな状態を保つようにします。`image`ミラーリングモードを使用する場合、イメージのミラーリングを有効にするとジャーナリング機能が自動的に有効になります。`pool`ミラーリングモードを使用する場合、イメージをピアクラスタにミラーリングするには、RBDイメージジャーナリング機能を有効にする必要があります。この機能は、イメージの作成時に`rbd`コマンドで`--image-feature exclusive-lock,journaling`オプションを指定することによって有効にできます。

または、既存のRBDイメージに対して動的にジャーナリング機能を有効にすることもできます。ジャーナリングを有効にするには、`feature enable`サブコマンド、プール名、イメージ名、および機能名を指定します。

```
cephuser@adm > rbd --cluster local feature enable POOL_NAME/IMAGE_NAME exclusive-lock
cephuser@adm > rbd --cluster local feature enable POOL_NAME/IMAGE_NAME journaling
```



## 注記: オプションの依存関係

journaling機能はexclusive-lock機能に依存します。exclusive-lock機能がまだ有効になっていない場合は、有効にしてからjournaling機能を有効にする必要があります。



## ヒント

デフォルトですべての新規イメージのジャーナリングを有効にできます。Ceph設定ファイルに`rbd default features = layering,exclusive-lock,object-map,deep-flatten,journaling`を追加してください。

### 20.4.2.3 イメージのミラースナップショットの作成

スナップショットベースのミラーリングを使用する場合、RBDイメージの変更内容のミラーリングが必要になるたびに、ミラースナップショットを作成する必要があります。`rbd`コマンドを使用してミラースナップショットを手動で作成するには、`mirror image snapshot`コマンドと共に、プール名とイメージ名を指定します。

```
cephuser@adm > rbd mirror image snapshot POOL_NAME/IMAGE_NAME
```

例:

```
cephuser@adm > rbd --cluster local mirror image snapshot image-pool/image-1
```

デフォルトでイメージごとに作成されるミラースナップショットは3つだけです。上限に達した場合は、最新のミラースナップショットが自動的に削除されます。`rbd_mirroring_max_mirroring_snapshots`設定オプションにより、必要に応じて上限を上書きできます。また、イメージが削除された場合やミラーリングが無効化された場合には、ミラースナップショットが自動的に削除されます。

ミラースナップショットのスケジュールを定義することで、定期的にミラースナップショットを自動作成することも可能です。ミラースナップショットのスケジュールは、グローバル、プールごと、イメージごとのレベルで設定できます。どのレベルにも複数のミラースナップショットのスケジュールを設定できます。ただし、個別のミラーイメージに適用される最も詳細なスナップショットスケジュールだけが実行されます。

**rbd**コマンドを使用してミラースナップショットスケジュールを作成するには、**mirror snapshot schedule add**コマンドと、プール名またはイメージ名(オプション)、スナップショット間隔、開始時間(オプション)を指定します。

スナップショット間隔はサフィックス**d**、**h**、**m**を使用することでそれぞれ日、時、分の単位で指定できます。必要に応じて、開始時間をISO 8601日時フォーマットにより指定できます。次に例を示します。

```
cephuser@adm > rbd --cluster local mirror snapshot schedule add --pool image-pool 24h 14:00:00-05:00
cephuser@adm > rbd --cluster local mirror snapshot schedule add --pool image-pool --image image1 6h
```

**rbd**コマンドを使用してミラースナップショットスケジュールを削除するには、**mirror snapshot schedule remove**コマンドと、対応するスケジュールの追加コマンドに一致するオプションを指定します。

**rbd**コマンドを使用して特定のレベル(グローバル、プール、イメージ)のすべてのスナップショットスケジュールを一覧にするには、**mirror snapshot schedule ls**コマンドと、必要に応じてプール名またはイメージ名を指定します。また、**--recursive**オプションを指定すると、指定したレベル以下のすべてのスケジュールを一覧にできます。次に例を示します。

```
cephuser@adm > rbd --cluster local mirror schedule ls --pool image-pool --recursive
POOL      NAMESPACE IMAGE  SCHEDULE
image-pool -      -      every 1d starting at 14:00:00-05:00
image-pool      image1 every 6h
```

**rbd**コマンドを使用して、スナップショットベースのミラーリングRBDイメージが次はいつ作成されるかを確認するには、**mirror snapshot schedule status**コマンドとプール名またはイメージ名(オプション)を指定します。次に例を示します。

```
cephuser@adm > rbd --cluster local mirror schedule status
SCHEDULE TIME      IMAGE
2020-02-26 18:00:00 image-pool/image1
```

#### 20.4.2.4 イメージミラーリングの無効化

特定のイメージのミラーリングを無効にするには、**mirror image disable**サブコマンドと共にプール名とイメージ名を指定します。

```
cephuser@adm > rbd --cluster local mirror image disable POOL_NAME/IMAGE_NAME
```

### 20.4.2.5 イメージの昇格と降格

プライマリ指定をピアクラスタ内のイメージに移動する必要があるフェールオーバーシナリオの場合、プライマリイメージへのアクセスを停止し、現在のプライマリイメージを降格してから、新しいプライマリイメージを昇格し、代替クラスタ上のイメージへのアクセスを再開する必要があります。



#### 注記: 強制昇格

`--force` オプションを使用して昇格を強制できます。強制昇格は、降格をピアクラスタに伝搬できない場合(たとえば、クラスタ障害や通信停止が発生した場合)に必要です。この結果、2つのピア間でスプリットブレインシナリオが発生し、**`resync`** サブコマンドを発行するまでイメージは同期されなくなります。

特定のイメージを非プライマリに降格するには、**`mirror image demote`** サブコマンドと共にプール名とイメージ名を指定します。

```
cephuser@adm > rbd --cluster local mirror image demote POOL_NAME/IMAGE_NAME
```

プール内のすべてのプライマリイメージを非プライマリに降格するには、**`mirror pool demote`** サブコマンドと共にプール名を指定します。

```
cephuser@adm > rbd --cluster local mirror pool demote POOL_NAME
```

特定のイメージをプライマリに昇格するには、**`mirror image promote`** サブコマンドと共にプール名とイメージ名を指定します。

```
cephuser@adm > rbd --cluster remote mirror image promote POOL_NAME/IMAGE_NAME
```

プール内のすべての非プライマリイメージをプライマリに昇格するには、**`mirror pool promote`** サブコマンドと共にプール名を指定します。

```
cephuser@adm > rbd --cluster local mirror pool promote POOL_NAME
```



#### ヒント: I/O 負荷の分割

プライマリまたは非プライマリの状態はイメージごとなので、2つのクラスタでI/O負荷を分割したり、フェールオーバーまたはフェールバックを実行したりできます。

### 20.4.2.6 イメージの再同期の強制

`rbd-mirror`デーモンがスプリットブレインイベントを検出した場合、このデーモンは、イベントが修正されるまで、影響を受けるイメージのミラーリングを試行しません。イメージのミラーリングを再開するには、まず、古いと判定されたイメージを降格してから、プライマリイメージへの再同期を要求します。イメージの再同期を要求するには、**`mirror image resync`**サブコマンドと共にプール名とイメージ名を指定します。

```
cephuser@adm > rbd mirror image resync POOL_NAME/IMAGE_NAME
```

### 20.4.3 ミラーリング状態の確認

ピアクラスタのレプリケーションの状態は、ミラーリングされたすべてのプライマリイメージについて保存されます。この状態は、**`mirror image status`**および**`mirror pool status`**の各サブコマンドを使用して取得できます。

ミラーイメージの状態を要求するには、**`mirror image status`**サブコマンドと共にプール名とイメージ名を指定します。

```
cephuser@adm > rbd mirror image status POOL_NAME/IMAGE_NAME
```

ミラープールのサマリ状態を要求するには、**`mirror pool status`**サブコマンドと共にプール名を指定します。

```
cephuser@adm > rbd mirror pool status POOL_NAME
```



#### ヒント:

**`mirror pool status`**サブコマンドに`--verbose`オプションを追加すると、プール内にあるすべてのミラーリングイメージについて状態の詳細も出力されます。

## 20.5 キャッシュの設定

Ceph Block Device (`librbd`)のユーザスペースの実装では、Linuxページキャッシュを利用できません。したがって、Ceph Block Deviceには独自のインメモリキャッシングが含まれます。RBDのキャッシュはハードディスクキャッシュと同様に動作します。OSがバリア要求またはフラッシュ要求を送信すると、すべての「ダーティ」データがOSDに書き込まれます。つまり、ライトバックキャッシュを使用することは、フラッシュを適切に送信するVMで正常

に動作する物理ハードディスクを使用することと同様に安全です。キャッシュはLRU (「Least Recently Used」) アルゴリズムを使用しており、ライトバックモードでは、隣接する要求をマージしてスループットを向上させることができます。

Cephは、RBDのライトバックキャッシュをサポートしています。RBDのライトバックキャッシュを有効にするには、次のコマンドを実行します。

```
cephuser@adm > ceph config set client rbd_cache true
```

デフォルトでは、librbdはキャッシュを実行しません。書き込みと読み込みはストレージクラスに直接送信され、書き込みはデータがすべてのレプリカのディスク上にある場合にのみ返されます。キャッシュを有効にすると、rbd\_cache\_max\_dirtyオプションで設定されている値より多くの未フラッシュバイトがある場合を除いて、書き込みはすぐに返されます。このような場合、書き込みはライトバックをトリガし、十分なバイトがフラッシュされるまでブロックされます。

CephはRBDのライトスルーキャッシュをサポートします。キャッシュのサイズを設定したり、ターゲットと制限を設定して、ライトバックキャッシュからライトスルーキャッシュに切り替えたりすることができます。ライトスルーモードを有効にするには、次のコマンドを実行します。

```
cephuser@adm > ceph config set client rbd_cache_max_dirty 0
```

つまり、書き込みはデータがすべてのレプリカのディスク上にある場合にのみ返されますが、読み込みはキャッシュから行われる場合があります。キャッシュはクライアントのメモリ内にあり、各RBDイメージは専用のキャッシュを持ちます。キャッシュはクライアントに対してローカルであるため、イメージにアクセスする他のユーザがいる場合、整合性はありません。キャッシュが有効な場合、RBDに加えてGFSまたはOCFSを実行することはできません。以下のパラメータがRADOS Block Deviceの動作に影響します。これらのパラメータを設定するには、clientカテゴリを使用します。

```
cephuser@adm > ceph config set client PARAMETER VALUE
```

#### rbd\_cache

RBD (RADOS Block Device)のキャッシュを有効にします。デフォルトは「true」です。

#### rbd\_cache\_size

RBDキャッシュのサイズ(バイト単位)。デフォルトは32MBです。

#### rbd\_cache\_max\_dirty

キャッシュがライトバックをトリガする「ダーティ」の制限(バイト単位)。rbd\_cache\_max\_dirtyは、rbd\_cache\_sizeより小さくする必要があります。0に設定すると、ライトスルーキャッシュを使用します。デフォルトは24MBです。

#### rbd cache target dirty

キャッシュがデータをデータストレージに書き込み始めるまでの「ダーティターゲット」。キャッシュへの書き込みはブロックしません。デフォルトは16MBです。

#### rbd cache max dirty age

ライトバックの開始前にダーティデータがキャッシュ内に存在する秒数。デフォルトは1です。

#### rbd cache writethrough until flush

ライトスルーモードで開始し、最初のフラッシュ要求を受信したらライトバックに切り替えます。rbdで実行されている仮想マシンが古すぎてフラッシュを送信できない場合(たとえば、カーネル2.6.32より前のLinuxのvirtioドライバ)は、この設定を有効にするのは消極的ですが安全です。デフォルトは「true」です。

## 20.6 QoS設定

一般的に、QoS (サービスの品質)とは、トラフィックの優先順位付けとリソース予約の方法のことを指します。これは特に、特別な要件を持つトラフィックを転送するために重要です。



### 重要: iSCSIではサポートされない

次のQoS設定は、ユーザスペースのRBD実装であるlibrbdによってのみ使用され、kRBD実装では使用されません。iSCSIはkRBDを使用するため、QoS設定を使用しません。ただし、iSCSIでは、標準のカーネル機能を使用して、カーネルブロックデバイス層でQoSを設定できます。

#### rbd qos iops limit

希望する秒あたりI/O操作数の上限。デフォルトは0 (制限なし)です。

#### rbd qos bps limit

希望する秒あたりI/Oバイト数の上限。デフォルトは0 (制限なし)です。

#### rbd qos read iops limit

希望する秒あたり読み取り操作数の上限。デフォルトは0 (制限なし)です。

#### rbd qos write iops limit

希望する秒あたり書き込み操作数の上限。デフォルトは0 (制限なし)です。

#### rbd qos read bps limit

希望する秒あたり読み取りバイト数の上限。デフォルトは0 (制限なし)です。

rbd qos write bps limit

希望する秒あたり書き込みバイト数の上限。デフォルトは0 (制限なし)です。

rbd qos iops burst

希望するI/O操作数のバースト上限。デフォルトは0 (制限なし)です。

rbd qos bps burst

希望するI/Oバイト数のバースト上限。デフォルトは0 (制限なし)です。

rbd qos read iops burst

希望する読み取り操作数のバースト上限。デフォルトは0 (制限なし)です。

rbd qos write iops burst

希望する書き込み操作数のバースト上限。デフォルトは0 (制限なし)です。

rbd qos read bps burst

希望する読み取りバイト数のバースト上限。デフォルトは0 (制限なし)です。

rbd qos write bps burst

希望する書き込みバイト数のバースト上限。デフォルトは0 (制限なし)です。

rbd qos schedule tick min

QoSの最小スケジューリングティック(ミリ秒)。デフォルトは50です。

## 20.7 先読み設定

RADOS Block Deviceは、先読み/プリフェッチをサポートしており、小容量の順次読み込みが最適化されます。これは、仮想マシンの場合は通常はゲストOSによって処理されますが、ブートローダは効率的な読み込みを発行できません。キャッシュが無効な場合、先読みは自動的に無効になります。



### 重要: iSCSIではサポートされない

次の先読み設定は、ユーザスペースのRBD実装であるlibrbdによってのみ使用され、kRBD実装では使用されません。iSCSIはkRBDを使用するため、先読み設定を使用しません。ただし、iSCSIでは、標準のカーネル機能を使用して、カーネルブロックデバイス層で先読みを設定できます。

rbd readahead trigger requests

先読みをトリガするために必要な順次読み込み要求の数。デフォルトは10です。

### rbd readahead max bytes

先読み要求の最大サイズ。0に設定すると、先読みは無効になります。デフォルトは512KBです。

### rbd readahead disable after bytes

この量のバイトRBDイメージから読み込みを行った後は、そのイメージが閉じられるまで先読みは無効になります。これにより、ゲストOSは起動時に先読みを引き継ぐことができます。0に設定すると、先読みは有効なままです。デフォルトは50MBです。

## 20.8 拡張機能

RADOS Block Deviceは、RBDイメージの機能を拡張する拡張機能をサポートしています。RBDイメージの作成時にコマンドラインで機能を指定すること、rbd\_default\_featuresオプションを使用してCeph設定ファイルで機能を指定することもできます。

rbd\_default\_featuresオプションの値は、次の2つの方法で指定できます。

- 機能の内部値の合計として指定する。各機能には独自の内部値があります。たとえば、「layering」は1で、「fast-diff」は16です。したがって、これらの2つの機能をデフォルトで有効にするには、以下を含めます。

```
rbd_default_features = 17
```

- 機能のカンマ区切りリストとして指定する。この場合、前の例は次のようになります。

```
rbd_default_features = layering,fast-diff
```



### 注記: iSCSIではサポートされない機能

deep-flatten、object-map、journaling、fast-diff、およびstripingの機能を使用するRBDイメージは、iSCSIではサポートされません。

次に、RBDの拡張機能のリストを示します。

### layering

階層化により、クローン作成を使用できます。  
内部値は1で、デフォルトは「yes」です。

### striping

ストライピングは、複数のオブジェクトにデータを分散し、順次読み込み/書き込みワークロードの並列処理に役立ちます。これにより、大容量またはビジー状態のRADOS Block Deviceにおいて単一ノードのボトルネックを防ぎます。

内部値は2で、デフォルトは「yes」です。

### exclusive-lock

有効にすると、クライアントは書き込みを行う前にオブジェクトのロックを取得する必要があります。単一のクライアントが同時に1つのイメージにアクセスしている場合にのみ、排他ロックを有効にします。内部値は4です。デフォルトは「yes」です。

### object-map

オブジェクトマップのサポートは、排他ロックのサポートに依存します。ブロックデバイスはシンプロビジョニングされます。つまり、実際に存在するデータのみを保存します。オブジェクトマップのサポートは、どのオブジェクトが実際に存在するか(ドライブに保存されたデータを持つか)を追跡するのに役立ちます。オブジェクトマップのサポートを有効にすると、クローン作成、保存密度の低いイメージのインポートとエクスポート、および削除のI/O操作が高速化されます。

内部値は8で、デフォルトは「yes」です。

### fast-diff

Fast-diffのサポートは、オブジェクトマップのサポートと排他ロックのサポートに依存します。これは、オブジェクトマップに別のプロパティを追加することで、イメージのスナップショットと、スナップショットの実際のデータ使用と間の差分を生成する速度が大幅に向上します。

内部値は16で、デフォルトは「yes」です。

### deep-flatten

ディープフラット化は、**rbid flatten** (20.3.3.6項「クローンイメージのフラット化」を参照)を、イメージそのものの以外にイメージのすべてのスナップショットでも機能するようにします。ディープフラット化がなければ、イメージのスナップショットは引き続き親に依存するため、スナップショットが削除されるまで親イメージを削除することはできません。ディープフラット化は、スナップショットがある場合でも、親をそのクローンから独立させます。

内部値は32で、デフォルトは「yes」です。

## ジャーナリング

ジャーナリングのサポートは排他ロックに依存します。ジャーナリングは、イメージに対するすべての変更を発生順に記録します。RBDミラーリング(20.4項「RBDイメージのミラーリング」を参照)では、ジャーナルを使用してクラッシュ整合イメージをremoteクラスタに複製します。

内部値は64で、デフォルトは「no」です。

## 20.9 古いカーネルクライアントを使用したRBDのマッピング

SUSE Enterprise Storage 7を使用して展開したクラスタでは、古いクライアント(SLE11 SP4など)でサポートされない複数の機能(RBDイメージレベルの機能とRADOSレベルの機能の両方)が強制的に適用されるため、これらの古いクライアントはRBDイメージをマップできない場合があります。これが発生した場合、OSDログに次のようなメッセージが表示されます。

```
2019-05-17 16:11:33.739133 7fcb83a2e700 0 -- 192.168.122.221:0/1006830 >> \
192.168.122.152:6789/0 pipe(0x65d4e0 sd=3 :57323 s=1 pgs=0 cs=0 l=1 c=0x65d770).connect \
protocol feature mismatch, my 2fffffffffffff < peer 4010ff8ffacffff missing 401000000000000
```



### 警告: CRUSHマップのバケットタイプを変更すると大規模なリバランスが発生する

CRUSHマップのバケットタイプを「straw」と「straw2」の間で切り替える場合は、計画的に行ってください。バケットタイプを変更するとクラスタの大規模なリバランスが発生するため、クラスタノードに重大な影響があることを想定しておいてください。

1. サポートされていないRBDイメージ機能を無効にします。例:

```
cephuser@adm > rbd feature disable pool1/image1 object-map
cephuser@adm > rbd feature disable pool1/image1 exclusive-lock
```

2. CRUSHマップのバケットタイプを「straw2」から「straw」に変更します。

- a. CRUSHマップを保存します。

```
cephuser@adm > ceph osd getcrushmap -o crushmap.original
```

- b. CRUSHマップを逆コンパイルします。

```
cephuser@adm > crushtool -d crushmap.original -o crushmap.txt
```

c. CRUSHマップを編集して、「straw2」を「straw」に置き換えます。

d. CRUSHマップを再コンパイルします。

```
cephuser@adm > crushtool -c crushmap.txt -o crushmap.new
```

e. 新しいCRUSHマップを設定します。

```
cephuser@adm > ceph osd setcrushmap -i crushmap.new
```

## 20.10 Block DeviceとKubernetesの有効化

Ceph RBDとKubernetes v1.13以上を`ceph-csi`ドライバにより併用できます。このドライバはRBDイメージを動的にプロビジョニングしてKubernetesボリュームを支援します。また、RBDに支援されたボリュームを参照するポッドを実行中のワーカーノードで、RBDイメージをBlock Deviceとしてマッピングします。

Ceph Block DeviceとKubernetesを併用するには、Kubernetes環境に`ceph-csi`をインストールして設定する必要があります。

### ！ 重要

`ceph-csi`はデフォルトではRBDカーネルモジュールを使用します。しかし、このモジュールがCeph CRUSHの調整可能パラメータや、RBDイメージ機能をすべてサポートしているとは限りません。

1. デフォルトでは、Ceph Block DeviceはRBDプールを使用します。Kubernetesボリュームストレージ用のプールを作成します。Cephクラスタが実行中であることを確認してから、プールを作成します。

```
cephuser@adm > ceph osd pool create kubernetes
```

2. RBDツールを使用してプールを初期化します。

```
cephuser@adm > rbd pool init kubernetes
```

3. Kubernetesと`ceph-csi`用の新しいユーザを作成します。次のコマンドを実行し、生成されたキーを記録します。

```
cephuser@adm > ceph auth get-or-create client.kubernetes mon 'profile rbd' osd 'profile rbd pool=kubernetes' mgr 'profile rbd pool=kubernetes'
```

```
[client.kubernetes]
key = AQD9o0Fd6hQRChAAt7fMaSZXduT3NWEqylNpmg==
```

4. `ceph-csi`は、Cephクラスタ用のCeph Monitorアドレスを定義するために、Kubernetesに保存されたConfigMapオブジェクトを必要とします。Cephクラスタの固有fsidとMonitorアドレスの両方を収集します。

```
cephuser@adm > ceph mon dump
<...>
fsid b9127830-b0cc-4e34-aa47-9d1a2e9949a8
<...>
0: [v2:192.168.1.1:3300/0,v1:192.168.1.1:6789/0] mon.a
1: [v2:192.168.1.2:3300/0,v1:192.168.1.2:6789/0] mon.b
2: [v2:192.168.1.3:3300/0,v1:192.168.1.3:6789/0] mon.c
```

5. 次の例に示すようなcsi-config-map.yamlファイルを生成します。なお、`clusterID`はFSIDで、`monitors`はMonitorアドレスで置き換えてください。

```
kubectl@adm > cat <<EOF > csi-config-map.yaml
---
apiVersion: v1
kind: ConfigMap
data:
  config.json: |-
    [
      {
        "clusterID": "b9127830-b0cc-4e34-aa47-9d1a2e9949a8",
        "monitors": [
          "192.168.1.1:6789",
          "192.168.1.2:6789",
          "192.168.1.3:6789"
        ]
      }
    ]
metadata:
  name: ceph-csi-config
EOF
```

6. 作成されたら、Kubernetesに新しいConfigMapオブジェクトを保存します。

```
kubectl@adm > kubectl apply -f csi-config-map.yaml
```

7. `ceph-csi`はCephクラスタとの通信にcephx資格情報を必要とします。新しく作成したKubernetesユーザIDとcephxキーを使用して、次の例に示すようなcsi-rbd-secret.yamlファイルを生成します。

```
kubectl@adm > cat <<EOF > csi-rbd-secret.yaml
---
```

```
apiVersion: v1
kind: Secret
metadata:
  name: csi-rbd-secret
  namespace: default
stringData:
  userID: kubernetes
  userKey: AQD9o0Fd6hQRChAA7fMaSZXduT3NWEqylNpmg==
EOF
```

8. 生成されたら、Kubernetesに新しいシークレットオブジェクトを保存します。

```
kubectl@adm > kubectl apply -f csi-rbd-secret.yaml
```

9. 必要となるServiceAccountとRBAC ClusterRole/ClusterRoleBindingのKubernetesオブジェクトを作成します。これらのオブジェクトを、お客様のKubernetes環境に合わせてカスタマイズする必要はありません。そのため、ceph-csi展開用のYAMLファイルから、オブジェクトを直接利用できます。

```
kubectl@adm > kubectl apply -f https://raw.githubusercontent.com/ceph/ceph-csi/master/deploy/rbd/kubernetes/csi-provisioner-rbac.yaml
kubectl@adm > kubectl apply -f https://raw.githubusercontent.com/ceph/ceph-csi/master/deploy/rbd/kubernetes/csi-nodeplugin-rbac.yaml
```

10. ceph-csiプロビジョナとノードプラグインを作成します。

```
kubectl@adm > wget https://raw.githubusercontent.com/ceph/ceph-csi/master/deploy/rbd/kubernetes/csi-rbdplugin-provisioner.yaml
kubectl@adm > kubectl apply -f csi-rbdplugin-provisioner.yaml
kubectl@adm > wget https://raw.githubusercontent.com/ceph/ceph-csi/master/deploy/rbd/kubernetes/csi-rbdplugin.yaml
kubectl@adm > kubectl apply -f csi-rbdplugin.yaml
```



### 重要

デフォルトでは、プロビジョナとノードプラグインのYAMLファイルは、ceph-csiコンテナの開発版リリースを取得します。リリース版を使用するように、YAMLファイルをアップデートする必要があります。

## 20.10.1 Kubernetes環境におけるCeph Block Deviceの利用

Kubernetes StorageClassはストレージクラスを定義します。複数のStorageClassオブジェクトを作成して、多様なサービス品質レベルと機能をマッピングできます。例としては、NVMeをベースのプールとHDDベースのプールの併用などです。

作成済みのKubernetesプールをマッピングするceph-csi StorageClassを作成するには、次のYAMLファイルを使用できます。ただし、作成前にclusterIDプロパティと使用するCephクラスタのFSIDが一致することを確認してください。

```
kubect@adm > cat <<EOF > csi-rbd-sc.yaml
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-rbd-sc
provisioner: rbd.csi.ceph.com
parameters:
  clusterID: b9127830-b0cc-4e34-aa47-9d1a2e9949a8
  pool: kubernetes
  csi.storage.k8s.io/provisioner-secret-name: csi-rbd-secret
  csi.storage.k8s.io/provisioner-secret-namespace: default
  csi.storage.k8s.io/node-stage-secret-name: csi-rbd-secret
  csi.storage.k8s.io/node-stage-secret-namespace: default
reclaimPolicy: Delete
mountOptions:
  - discard
EOF
kubect@adm > kubectl apply -f csi-rbd-sc.yaml
```

PersistentVolumeClaimはユーザからの抽象化ストレージリソースに対する要求です。要求後、PersistentVolumeClaimはポッドのリソースに関連付けられ、PersistentVolumeをプロビジョニングします。これは、Cephブロックイメージに支援されます。必要に応じてvolumeModeを付加することで、マウント済みのファイルシステム(デフォルト)と、Block DeviceベースのRAWボリュームを選択できます。

ceph-csiを使用する場合、volumeModeにFilesystemを指定すると、ReadWriteOnce accessMode要求とReadOnlyMany accessMode要求の両方をサポートできます。また、volumeModeにBlockを指定すると、ReadWriteOnce accessMode要求、ReadWriteMany accessMode要求、ReadOnlyMany accessMode要求をサポートできます。

たとえば、前の手順で作成したceph-csi-based StorageClassを使用する、ブロックベースのPersistentVolumeClaimを作成するには、次のYAMLファイルを使用できます。これにより、csi-rbd-sc StorageClassからRAWブロックストレージを要求します。

```
kubect@adm > cat <<EOF > raw-block-pvc.yaml
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: raw-block-pvc
spec:
  accessModes:
```

```

- ReadWriteOnce
volumeMode: Block
resources:
  requests:
    storage: 1Gi
storageClassName: csi-rbd-sc
EOF
kubectl@adm > kubectl apply -f raw-block-pvc.yaml

```

次に示すのは、前に述べたPersistentVolumeClaimをRAW Block Deviceとしてポッドのソースにバインドする例です。

```

kubectl@adm > cat <<EOF > raw-block-pod.yaml
---
apiVersion: v1
kind: Pod
metadata:
  name: pod-with-raw-block-volume
spec:
  containers:
    - name: fc-container
      image: fedora:26
      command: ["/bin/sh", "-c"]
      args: ["tail -f /dev/null"]
      volumeDevices:
        - name: data
          devicePath: /dev/xvda
  volumes:
    - name: data
      persistentVolumeClaim:
        claimName: raw-block-pvc
EOF
kubectl@adm > kubectl apply -f raw-block-pod.yaml

```

前の手順で作成した ceph-csi-based StorageClassを使用する、ファイルシステムベースのPersistentVolumeClaimを作成するには、次のYAMLファイルを使用できます。これにより、csi-rbd-sc StorageClassからマウント済みのファイルシステム(RBDイメージにより支援)を要求します。

```

kubectl@adm > cat <<EOF > pvc.yaml
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: rbd-pvc
spec:
  accessModes:
    - ReadWriteOnce
  volumeMode: Filesystem

```

```
resources:
  requests:
    storage: 1Gi
    storageClassName: csi-rbd-sc
EOF
kubectl@adm > kubectl apply -f pvc.yaml
```

次に示すのは、前に述べたPersistentVolumeClaimをマウント済みのファイルシステムとしてポッドのリソースにバインドする例です。

```
kubectl@adm > cat <<EOF > pod.yaml
---
apiVersion: v1
kind: Pod
metadata:
  name: csi-rbd-demo-pod
spec:
  containers:
    - name: web-server
      image: nginx
      volumeMounts:
        - name: mypvc
          mountPath: /var/lib/www/html
  volumes:
    - name: mypvc
      persistentVolumeClaim:
        claimName: rbd-pvc
        readOnly: false
EOF
kubectl@adm > kubectl apply -f pod.yaml
```

## IV クラスタデータへのアクセス

- 21 Ceph Object Gateway **247**
- 22 Ceph iSCSI Gateway **299**
- 23 クラスタファイルシステム **316**
- 24 Sambaを介したCephデータのエクスポート **327**
- 25 NFS Ganesha **345**

## 21 Ceph Object Gateway

この章では、サービスの状態の確認、アカウントの管理、マルチサイトゲートウェイ、LDAP認証など、Object Gatewayに関連する管理タスクについて詳しく説明します。

### 21.1 Object Gatewayの制約と命名の制限

次に、Object Gatewayの重要な制限のリストを示します。

#### 21.1.1 バケットの制限

S3 API経由でObject Gatewayに接続する場合、バケット名はDNSに準拠した名前(ダッシュ文字「-」は使用可能)に制限されます。Swift API経由でObject Gatewayに接続する場合は、UTF-8でサポートされている文字(スラッシュ文字「/」を除く)を自由に組み合わせて使用できます。バケット名の最大長は255文字です。バケット名は固有でなければなりません。



#### ヒント: DNSに準拠したバケット名

Swift API経由ではUTF-8ベースのバケット名を使用できますが、同じバケットにS3 API経由でアクセスする際に問題が起きないように、バケットにはS3の命名制限に従った名前を付けることをお勧めします。

#### 21.1.2 保存オブジェクトの制限

##### ユーザあたりのオブジェクトの最大数

デフォルトでは制限はありません(最大 $2^{63}$ に制限)。

##### バケットあたりのオブジェクトの最大数

デフォルトでは制限はありません(最大 $2^{63}$ に制限)。

##### アップロード/保存するオブジェクトの最大サイズ

1回のアップロードは5GBに制限されます。これより大きいオブジェクトサイズにはマルチパートを使用してください。マルチパートチャンクの最大数は10000です。

### 21.1.3 HTTPヘッダの制限

HTTPヘッダと要求の制限は、使用するWebフロントエンドによって異なります。デフォルトのBeastでは、HTTPヘッダのサイズは16kBに制限されています。

## 21.2 Object Gatewayの展開

Ceph Object Gatewayの展開方法は、その他のCephサービスの展開手順と同じ、つまり `cephadm` を使用します。詳細については、『導入ガイド』、第5章「`cephadm`による展開」、5.4.2項「サービス仕様と配置仕様」、また、具体的には『導入ガイド』、第5章「`cephadm`による展開」、5.4.3.4項「Object Gatewayの展開」を参照してください。

## 21.3 Object Gatewayサービスの操作

その他のCephサービスと同じようにObject Gatewayを操作できます。その場合、まず、`ceph orch ps` コマンドを使用してサービス名を特定し、次のコマンドを実行してサービスを操作します。次に例を示します。

```
ceph orch daemon restart OGW_SERVICE_NAME
```

Cephサービスの操作の詳細については、第14章「Cephサービスの運用」を参照してください。

## 21.4 設定オプション

Object Gatewayの設定オプションのリストについては、28.5項「Ceph Object Gateway」を参照してください。

## 21.5 Object Gatewayのアクセスの管理

S3またはSwiftと互換性のあるインタフェースを使用してObject Gatewayと通信できます。S3インタフェースは、Amazon S3 RESTful APIの大規模なサブセットと互換性があります。Swiftインタフェースは、OpenStack Swift APIの大規模なサブセットと互換性があります。

どちらのインタフェースも、ユーザの秘密鍵を使用してゲートウェイと通信するには、特定のユーザを作成し、関連するクライアントソフトウェアをインストールする必要があります。

## 21.5.1 Object Gatewayへのアクセス

### 21.5.1.1 S3インタフェースへのアクセス

S3インタフェースにアクセスするには、RESTクライアントが必要です。S3cmdはコマンドラインのS3クライアントです。これは、[OpenSUSEビルドサービス \(https://build.opensuse.org/package/show/Cloud:Tools/s3cmd\)](https://build.opensuse.org/package/show/Cloud:Tools/s3cmd) にあります。このリポジトリには、SUSE Linux EnterpriseベースおよびopenSUSEベースの両方の配布パッケージ用のバージョンがあります。

S3インタフェースへのアクセスをテストする場合、簡単なPythonスクリプトを作成することもできます。このスクリプトは、Object Gatewayに接続して新しいバケットを作成し、すべてのバケットを一覧にします。aws\_access\_key\_idおよびaws\_secret\_access\_keyの値は、[21.5.2.1項「S3およびSwiftユーザの追加」](#)のradosgw\_adminコマンドによって返されるaccess\_keyおよびsecret\_keyの値から取得されます。

1. python-botoパッケージをインストールします。

```
root # zypper in python-boto
```

2. 次の内容で、s3test.pyという名前の新しいPythonスクリプトを作成します。

```
import boto
import boto.s3.connection
access_key = '11BS02LGFB6AL6H1ADMW'
secret_key = 'vzCEkuryfn060dfef4fgQPqFrncKEIkh3Zcd0ANY'
conn = boto.connect_s3(
    aws_access_key_id = access_key,
    aws_secret_access_key = secret_key,
    host = 'HOSTNAME',
    is_secure=False,
    calling_format = boto.s3.connection.OrdinaryCallingFormat(),
)
bucket = conn.create_bucket('my-new-bucket')
for bucket in conn.get_all_buckets():
    print "NAME\tCREATED".format(
        name = bucket.name,
        created = bucket.creation_date,
    )
```

HOSTNAMEは、Object Gatewayサービスを設定したホストのホスト名に置き換えてください。たとえば、gateway\_hostです。

3. スクリプトを実行します。

```
python s3test.py
```

次のような内容が出力されます。

```
my-new-bucket 2015-07-22T15:37:42.000Z
```

### 21.5.1.2 Swiftインタフェースへのアクセス

SwiftインタフェースでObject Gatewayにアクセスするには、**swift** コマンドラインクライアントが必要です。コマンドラインオプションについては、マニュアルページ [man 1 swift](#) を参照してください。

このパッケージは、SP3以降のSUSE Linux Enterprise 12およびSUSE Linux Enterprise 15の「パブリッククラウド」モジュールに含まれています。パッケージをインストールする前に、このモジュールを有効にしてソフトウェアリポジトリを更新する必要があります。

```
root # SUSEConnect -p sle-module-public-cloud/12/SYSTEM-ARCH
sudo zypper refresh
```

または

```
root # SUSEConnect -p sle-module-public-cloud/15/SYSTEM-ARCH
root # zypper refresh
```

**swift** コマンドをインストールするには、次のコマンドを実行します。

```
root # zypper in python-swiftclient
```

Swiftにアクセスするには、次の構文を使用します。

```
tux > swift -A http://IP_ADDRESS/auth/1.0 \
-U example_user:swift -K 'SWIFT_SECRET_KEY' list
```

IP\_ADDRESSは、ゲートウェイサーバのIPアドレスに置き換えてくださ

い。SWIFT\_SECRET\_KEYは、[21.5.2.1項「S3およびSwiftユーザの追加」](#)で**swift**ユーザを対象に実行した**radosgw-admin key create**コマンドの出力の値に置き換えてください。

例:

```
tux > swift -A http://gateway.example.com/auth/1.0 -U example_user:swift \
-K 'r5wWlXj0CeE07DixD1FjTLmNYIViaC6JVhi3013h' list
```

出力は次のとおりです。

## 21.5.2 S3およびSwiftアカウントの管理

### 21.5.2.1 S3およびSwiftユーザの追加

エンドユーザがゲートウェイを操作できるようにするには、ユーザ、アクセスキー、および秘密を作成する必要があります。ユーザには、「ユーザ」と「サブユーザ」の2種類があります。「ユーザ」はS3インタフェースを操作する場合に使用し、「サブユーザ」はSwiftインタフェースのユーザです。各サブユーザは特定のユーザに関連付けられます。

Swiftユーザを作成するには、次の手順に従います。

1. Swiftユーザ(ここでの用語では「サブユーザ」)を作成するために、まず関連付けられた「ユーザ」を作成する必要があります。

```
cephuser@adm > radosgw-admin user create --uid=USERNAME \
--display-name="DISPLAY-NAME" --email=EMAIL
```

例:

```
cephuser@adm > radosgw-admin user create \
--uid=example_user \
--display-name="Example User" \
--email=penguin@example.com
```

2. このユーザのサブユーザ(Swiftインタフェース)を作成するために、ユーザID (--uid=USERNAME)、サブユーザID、およびサブユーザのアクセスレベルを指定する必要があります。

```
cephuser@adm > radosgw-admin subuser create --uid=UID \
--subuser=UID \
--access=[ read | write | readwrite | full ]
```

例:

```
cephuser@adm > radosgw-admin subuser create --uid=example_user \
--subuser=example_user:swift --access=full
```

3. ユーザの秘密鍵を生成します。

```
cephuser@adm > radosgw-admin key create \
--gen-secret \
```

```
--subuser=example_user:swift \  
--key-type=swift
```

4. どちらのコマンドでも、ユーザの状態を示すJSON形式のデータが出力されます。次の行に注意し、secret\_keyの値を覚えめす。

```
"swift_keys": [  
  { "user": "example_user:swift",  
    "secret_key": "r5wWIXj0CeE07DixD1FjTlMNYIViaC6JVhi3013h"}],
```

S3インタフェースを介してObject Gatewayにアクセスする場合、次のコマンドを実行してS3ユーザを作成する必要があります。

```
cephuser@adm > radosgw-admin user create --uid=USERNAME \  
--display-name="DISPLAY-NAME" --email=EMAIL
```

例:

```
cephuser@adm > radosgw-admin user create \  
--uid=example_user \  
--display-name="Example User" \  
--email=penguin@example.com
```

このコマンドは、ユーザのアクセスキーと秘密鍵も生成します。access\_keyおよびsecret\_keyのキーワードの出力とそれらの値を確認します。

```
[...]
"keys": [  
  { "user": "example_user",  
    "access_key": "11BS02LGFB6AL6H1ADMW",  
    "secret_key": "vzCEkuryfn060dfee4fgQPqFrncKEIkh3Zcd0ANY"}],  
[...]
```

### 21.5.2.2 S3およびSwiftユーザの削除

ユーザを削除する手順は、S3ユーザでもSwiftユーザでも同様です。ただし、Swiftユーザの場合、そのサブユーザを含むユーザを削除する必要があります。

S3またはSwiftユーザ(その全サブユーザを含む)を削除するには、次のコマンドでuser rmとユーザIDを指定します。

```
cephuser@adm > radosgw-admin user rm --uid=example_user
```

サブユーザを削除するには、subuser rmとサブユーザIDを指定します。

```
cephuser@adm > radosgw-admin subuser rm --uid=example_user:swift
```

次のオプションを利用できます。

#### **--purge-data**

ユーザIDに関連付けられたすべてのデータをパージします。

#### **--purge-keys**

ユーザIDに関連付けられたすべてのキーをパージします。



### ヒント: サブユーザの削除

サブユーザを削除すると、Swiftインタフェースへのアクセスも削除されます。そのユーザはシステムに残ります。

## 21.5.2.3 S3およびSwiftユーザのアクセスキーと秘密鍵の変更

`access_key`および`secret_key`のパラメータは、ゲートウェイにアクセスする際にObject Gatewayユーザを識別します。既存のユーザのキーを削除すると、古いキーは上書きされます。そのため、これは新しいユーザを作成することと同じです。

S3ユーザの場合、次のコマンドを実行します。

```
cephuser@adm > radosgw-admin key create --uid=EXAMPLE_USER --key-type=s3 --gen-access-key --gen-secret
```

Swiftユーザの場合、次のコマンドを実行します。

```
cephuser@adm > radosgw-admin key create --subuser=EXAMPLE_USER:swift --key-type=swift --gen-secret
```

#### --key-type=TYPE

キーのタイプを指定します。`swift`または`s3`です。

#### --gen-access-key

ランダムなアクセスキーを生成します(デフォルトではS3ユーザ用)。

#### --gen-secret

ランダムな秘密鍵を生成します。

#### --secret=KEY

秘密鍵を指定します。たとえば、手動で生成した秘密鍵を指定します。

### 21.5.2.4 ユーザクォータの管理の有効化

Ceph Object Gatewayでは、ユーザと、ユーザが所有するバケットにクォータを設定できます。バケット内のオブジェクトの最大数や、最大ストレージサイズ(メガバイト単位)などのクォータがあります。

ユーザクォータを有効にする前に、まずそのパラメータを設定する必要があります。

```
cephuser@adm > radosgw-admin quota set --quota-scope=user --uid=EXAMPLE_USER \
--max-objects=1024 --max-size=1024
```

#### --max-objects

オブジェクトの最大数を指定します。負の値を指定すると、クォータの確認が無効になります。

#### --max-size

最大バイト数を指定します。負の値を指定すると、クォータの確認が無効になります。

#### --quota-scope

クォータのスコープを設定します。オプションはbucketおよびuserです。バケットクォータは、ユーザが所有するバケットに適用されます。ユーザクォータはユーザに適用されます。

ユーザクォータを選択したら、そのクォータを有効にできます。

```
cephuser@adm > radosgw-admin quota enable --quota-scope=user --uid=EXAMPLE_USER
```

クォータを無効にするには、次のコマンドを実行します。

```
cephuser@adm > radosgw-admin quota disable --quota-scope=user --uid=EXAMPLE_USER
```

クォータの設定を一覧にするには、次のコマンドを実行します。

```
cephuser@adm > radosgw-admin user info --uid=EXAMPLE_USER
```

クォータの統計を更新するには、次のコマンドを実行します。

```
cephuser@adm > radosgw-admin user stats --uid=EXAMPLE_USER --sync-stats
```

## 21.6 HTTPフロントエンド

Ceph Object Gatewayは、2つの埋め込みHTTPフロントエンド(「Beast」と「Civetweb」)をサポートしています。

Beastフロントエンドは、HTTPの解析のためにBoost.Beastライブラリを使用し、非同期ネットワークI/OのためにBoost.Asioライブラリを使用します。

Civetwebフロントエンドは、MongooseのフォークであるCivetweb HTTPライブラリを使用します。

これらは`rgw_frontends`オプションを使用して設定できます。設定オプションのリストについては、[28.5項「Ceph Object Gateway」](#)を参照してください。

## 21.7 Object GatewayでのHTTPS/SSLの有効化

Object Gatewayを有効にしてSSLで安全に通信するには、CAによって発行された証明書を持っているか、自己署名証明書を作成する必要があります。

### 21.7.1 自己署名証明書の作成



#### ヒント

CAによって署名された有効な証明書をすでに持っている場合、このセクションはスキップしてください。

次の手順では、Salt Master上で自己署名SSL証明書を生成する方法について説明します。

1. Object Gatewayを追加のサブジェクトIDで認識する必要がある場合は、それらを`/etc/ssl/openssl.cnf`ファイルの`[v3_req]`セクションの`subjectAltName`オプションに追加します。

```
[...]
[ v3_req ]
subjectAltName = DNS:server1.example.com DNS:server2.example.com
[...]
```



#### ヒント: subjectAltNameのIPアドレス

`subjectAltName`オプションのドメイン名の代わりにIPアドレスを使用するには、上の例に示されている行を以下で置き換えてください。

```
subjectAltName = IP:10.0.0.10 IP:10.0.0.11
```

2. **openssl**を使用して、キーと証明書を作成します。証明書に含める必要があるデータをすべて入力します。FQDNを一般名として入力することをお勧めします。証明書に署名する前に、「Requested Extensions」に「X509v3 Subject Alternative Name:」が含まれていること、および生成された証明書に「X509v3 Subject Alternative Name:」が設定されていることを確認します。

```
root@master # openssl req -x509 -nodes -days 1095 \  
-newkey rsa:4096 -keyout rgw.key \  
-out rgw.pem
```

3. キーを証明書ファイルに追加します。

```
root@master # cat rgw.key >> rgw.pem
```

## 21.7.2 SSLを使用するようにObject Gatewayを設定する

SSL証明書を使用するようにObject Gatewayを設定するには、`rgw_frontends`オプションを使用します。例:

```
cephuser@adm > ceph config set WHO rgw_frontends \  
beast ssl_port=443 ssl_certificate=config://CERT ssl_key=config://KEY
```

CERTとKEYの設定キーを指定しない場合、Object Gatewayサービスは、次の設定キーの下にあるSSL証明書およびキーを探します。

```
rgw/cert/RGW_REALM/RGW_ZONE.key  
rgw/cert/RGW_REALM/RGW_ZONE.crt
```

デフォルトのSSLキーおよび証明書の場所を上書きする場合、次のコマンドを使用してそれらを設定データベースにインポートします。

```
ceph config-key set CUSTOM_CONFIG_KEY -i PATH_TO_CERT_FILE
```

`config://`ディレクティブを使用してカスタム設定キーを使用します。

## 21.8 同期モジュール

Object Gatewayは、マルチサイトサービスとして展開されます。また、データおよびメタデータをゾーン間でミラーリングできます。「同期モジュール」は、データとメタデータを別の外部層へ転送できるようにするマルチサイトフレームワーク上に構築されています。同期

モジュールにより、データの変更が発生した場合に一連のアクションを実行できます(たとえば、バケットやユーザの作成などのメタデータの操作)。Object Gatewayでのマルチサイトの変更は最終的にリモートサイトで一貫性が保たれるので、変更は非同期で伝搬されます。これは、外部のクラウドクラスタへのObject Storageのバックアップ、テープドライブを使用するカスタムバックアップソリューション、ElasticSearchでのメタデータのインデックス作成といった使用事例に対応しています。

## 21.8.1 同期モジュールの設定

すべての同期モジュールは同様の方法で設定します。新しいゾーンを作成し(詳細については[21.13項「マルチサイトObject Gateway」](#)を参照)、その`--tier_type`オプションを設定する必要があります。たとえば、クラウド同期モジュールの場合は、`--tier-type=cloud`に設定します。

```
cephuser@adm > radosgw-admin zone create --rgw-zonegroup=ZONE-GROUP-NAME \
--rgw-zone=ZONE-NAME \
--endpoints=http://endpoint1.example.com,http://endpoint2.example.com, [...] \
--tier-type=cloud
```

次のコマンドを使用して特定の層を設定できます。

```
cephuser@adm > radosgw-admin zone modify --rgw-zonegroup=ZONE-GROUP-NAME \
--rgw-zone=ZONE-NAME \
--tier-config=KEY1=VALUE1,KEY2=VALUE2
```

設定の`KEY`には更新する設定変数を指定し、`VALUE`にその新しい値を指定します。ネストされた値にはピリオドを使用してアクセスできます。例:

```
cephuser@adm > radosgw-admin zone modify --rgw-zonegroup=ZONE-GROUP-NAME \
--rgw-zone=ZONE-NAME \
--tier-config=connection.access_key=KEY,connection.secret=SECRET
```

参照されているエントリに角括弧「[]」を追加して、配列エントリにアクセスできます。角括弧「[]」を使用して、新しい配列エントリを追加できます。インデックス値の-1は、配列の最後のエントリを参照します。新しいエントリを作成し、同じコマンドで再びそれを参照することはできません。たとえば、`PREFIX`で始まるバケットの新しいプロファイルを作成するコマンドは次のとおりです。

```
cephuser@adm > radosgw-admin zone modify --rgw-zonegroup=ZONE-GROUP-NAME \
--rgw-zone=ZONE-NAME \
--tier-config=profiles[].source_bucket=PREFIX'*'
cephuser@adm > radosgw-admin zone modify --rgw-zonegroup=ZONE-GROUP-NAME \
--rgw-zone=ZONE-NAME \
```

```
--tier-config=profiles[-1].connection_id=CONNECTION_ID,profiles[-1].acls_id=ACLS_ID
```



## ヒント: 設定エントリの追加と削除

`--tier-config-add=KEY=VALUE`パラメータを使用して、新しい層の設定エントリを追加できます。

`--tier-config-rm=KEY`を使用して、既存のエントリを削除できます。

## 21.8.2 ゾーンの同期

同期モジュール設定はゾーンにローカルです。同期モジュールは、ゾーンがデータをエクスポートするのか、それとも別のゾーンで変更されたデータを使用できるだけかを判断します。Luminousの時点でサポートされている同期プラグインは、`ElasticSearch`、`rgw` (ゾーン間でデータを同期するデフォルトの同期プラグイン)、および`log` (リモートゾーンで実行されるメタデータ操作を記録する単純な同期プラグイン)です。以降のセクションでは、`ElasticSearch`同期モジュールを使用するゾーンを例にして説明します。他の同期プラグインでも設定のプロセスは同様です。



## 注記: デフォルトの同期プラグイン

`rgw`はデフォルトの同期プラグインで、明示的な設定は必要はありません。

### 21.8.2.1 要件と前提

21.13項「マルチサイトObject Gateway」で説明されているような、2つのゾーン`us-east`と`us-west`で構成される単純なマルチサイト設定を前提にしましょう。ここでは、他のサイトからのメタデータのみを処理するゾーンである3つ目のゾーン`us-east-es`を追加します。このゾーンは、`us-east`と同じCephクラスタにあっても、別のクラスタにあっても構いません。このゾーンは他のゾーンからのメタデータのみを使用し、このゾーンのObject Gatewayはエンドユーザの要求を直接実行することはありません。

### 21.8.2.2 ゾーンの設定

1. 21.13項「マルチサイトObject Gateway」で説明されているものと同様の3つ目のゾーンを作成します。次に例を示します。

```
cephuser@adm > radosgw-admin zone create --rgw-zonegroup=us --rgw-zone=us-east-es \
--access-key=SYSTEM-KEY --secret=SECRET --endpoints=http://rgw-es:80
```

2. 次のコマンドを使用して、このゾーンに対して同期モジュールを設定できます。

```
cephuser@adm > radosgw-admin zone modify --rgw-zone=ZONE-NAME --tier-type=TIER-TYPE \
--tier-config={set of key=value pairs}
```

3. たとえば、ElasticSearch同期モジュールでは、次のコマンドを実行します。

```
cephuser@adm > radosgw-admin zone modify --rgw-zone=ZONE-NAME --tier-
type=elasticsearch \
--tier-config=endpoint=http://localhost:9200,num_shards=10,num_replicas=1
```

サポートされているさまざまなtier-configオプションについては、[21.8.3項「ElasticSearch同期モジュール」](#)を参照してください。

4. 最後にピリオドを更新します。

```
cephuser@adm > radosgw-admin period update --commit
```

5. 続いて、ゾーンでObject Gatewayを起動します。

```
cephuser@adm > ceph orch start rgw.REALM-NAME.ZONE-NAME
```

## 21.8.3 ElasticSearch同期モジュール

この同期モジュールは、他のゾーンからElasticSearchにメタデータを書き込みます。Luminousの時点では、これは、現在ElasticSearchに保存しているJSON形式のデータフィールドです。

```
{
  "_index" : "rgw-gold-ee5863d6",
  "_type" : "object",
  "_id" : "34137443-8592-48d9-8ca7-160255d52ade.34137.1:object1:null",
  "_score" : 1.0,
  "_source" : {
    "bucket" : "testbucket123",
    "name" : "object1",
    "instance" : "null",
    "versioned_epoch" : 0,
    "owner" : {
      "id" : "user1",
```

```

    "display_name" : "user1"
  },
  "permissions" : [
    "user1"
  ],
  "meta" : {
    "size" : 712354,
    "mtime" : "2017-05-04T12:54:16.462Z",
    "etag" : "7ac66c0f148de9519b8bd264312c4d64"
  }
}
}

```

### 21.8.3.1 Elasticsearchの層タイプの設定パラメータ

#### endpoint

アクセスするElasticSearchサーバエンドポイントを指定します。

#### num\_shards

(整数)データ同期初期化時にElasticSearchに設定するシャードの数。初期化後は変更できないことに注意してください。ここで変更を行った場合、ElasticSearchのインデックスの再構築と、データ同期プロセスの再初期化が必要になります。

#### num\_replicas

(整数)データ同期初期化時にElasticSearchに設定するレプリカの数。

#### explicit\_custom\_meta

(true | false)すべてのユーザカスタムメタデータのインデックスを作成するか、それともカスタムメタデータエントリのインデックスを作成する対象をユーザが(バケットレベルで)設定する必要があるかを指定します。デフォルトではfalseになっています。

#### index\_buckets\_list

(文字列のコンマ区切りリスト)空の場合、すべてのバケットのインデックスが作成されます。空でない場合、ここで指定したバケットのインデックスのみが作成されます。バケットのプレフィックス(「foo\*」など)またはサフィックス(「\*bar」など)を指定できます。

#### approved\_owners\_list

(文字列のコンマ区切りリスト)空の場合、すべての所有者のバケットのインデックスが作成されます(他の制約に依存)。空でない場合、指定した所有者が所有するバケットのインデックスのみが作成されます。サフィックスとプレフィックスを指定することもできます。

## override\_index\_path

(文字列)空でない場合、この文字列がElasticSearchのインデックスパスとして使用されます。空の場合、インデックスパスは同期初期化時に決定されて生成されます。

## ユーザ名

認証が必要な場合にElasticSearchのユーザ名を指定します。

## password

認証が必要な場合にElasticSearchのパスワードを指定します。

### 21.8.3.2 メタデータクエリ

ElasticSearchクラスタにオブジェクトメタデータが保存されるようになったので、ElasticSearchエンドポイントを一般に公開しないようにし、エンドポイントにはクラスタ管理者のみがアクセスできるようにすることが重要です。ユーザが自身のメタデータのみを問い合わせ、他のユーザのメタデータは問い合わせないようにするため、メタデータクエリをエンドユーザそのものに公開すると、問題が発生します。このためには、ElasticSearchクラスタでもRGWと同様の方法でユーザを認証する必要がありますが、これが問題になります。

Luminousから、メタデータマスタゾーンのRGWでエンドユーザの要求を実行できるようになりました。これにより、ElasticSearchエンドポイントを一般に公開しないようにできると同時に、RGWそのものがエンドユーザの要求を認証できるので、認証と権限付与の問題も解決します。このために、RGWでは、バケットAPIにElasticSearchの要求を実行できる新しいクエリが導入されています。これらの要求はすべてメタデータマスタゾーンに送信する必要があります。

## ElasticSearchクエリの取得

```
GET /BUCKET?query=QUERY-EXPR
```

要求パラメータ:

- max-keys: 返されるエントリの最大数
- marker: ページ分割マーカ

```
expression := ([(<arg> <op> <value> [])[<and|or> ...]
```

演算子は、<、<=、==、>=、>の1つです。

例:

```
GET /?query=name==foo
```

ユーザが読み込み許可を持つ、「foo」という名前のインデックス作成済みキーをすべて返します。出力は、S3のバケット一覧の応答に似たXML形式のキーのリストになります。

### カスタムメタデータフィールドの設定

インデックスの作成が必要なカスタムメタデータエントリ(指定したバケットの下層)と、これらのキーのタイプを定義します。カスタムメタデータのインデックス作成が明示的に設定されている場合、rgwによって指定のカスタムメタデータ値のインデックスが作成されるようにするため、これが必要になります。それ以外の場合は、インデックスが作成されるメタデータキーのタイプが文字列以外のときに必要です。

```
POST /BUCKET?mdsearch
x-amz-meta-search: <key [; type]> [, ...]
```

複数のメタデータフィールドはコンマで区切る必要があります。「;」を使用して、フィールドに対してタイプを強制的に適用できます。現在許可されているタイプは、文字列(デフォルト)、整数、および日付です。たとえば、カスタムオブジェクトメタデータx-amz-meta-yearを整数、x-amz-meta-dateを日付タイプ、およびx-amz-meta-titleを文字列としてインデックスを作成する場合、次のように指定します。

```
POST /mybooks?mdsearch
x-amz-meta-search: x-amz-meta-year;int, x-amz-meta-release-date;date, x-amz-meta-title;string
```

### カスタムメタデータ設定の削除

カスタムメタデータのバケット設定を削除します。

```
DELETE /BUCKET?mdsearch
```

### カスタムメタデータ設定の取得

カスタムメタデータのバケット設定を取得します。

```
GET /BUCKET?mdsearch
```

## 21.8.4 クラウド同期モジュール

このセクションでは、ゾーンデータをリモートクラウドサービスに同期するモジュールについて説明します。同期は単方向のみです。日付はリモートゾーンから同期されません。このモジュールの主な目的は、データを複数のクラウドサービスプロバイダと同期できるようにすることです。現在のところ、AWS (S3)と互換性のあるクラウドプロバイダがサポートされています。

データをリモートクラウドサービスに同期するには、ユーザ資格情報を設定する必要があります。多くのクラウドサービスでは、各ユーザが作成できるバケットの数に制限を導入しているため、ソースオブジェクトとバケット、異なるターゲットから異なるバケットとバケットプレフィックスへのマッピングを設定できます。ソースのアクセスリスト(ACL)は保持されないことに注意してください。特定のソースユーザの許可を特定の宛先ユーザにマッピングできます。

APIの制限のため、元のオブジェクト変更時刻とHTTP ETag (エンティティタグ)を保持する方法はありません。クラウド同期モジュールは、これらを宛先オブジェクトのメタデータ属性として保存します。

#### 21.8.4.1 クラウド同期モジュールの設定

以下はクラウド同期モジュールの単純な設定と複雑な設定の例です。単純な設定は複雑な設定と競合する可能性があることに注意してください。

##### 例 21.1: 単純な設定

```
{
  "connection": {
    "access_key": ACCESS,
    "secret": SECRET,
    "endpoint": ENDPOINT,
    "host_style": path | virtual,
  },
  "acls": [ { "type": id | email | uri,
    "source_id": SOURCE_ID,
    "dest_id": DEST_ID } ... ],
  "target_path": TARGET_PATH,
}
```

##### 例 21.2: 複雑な設定

```
{
  "default": {
    "connection": {
      "access_key": ACCESS,
      "secret": SECRET,
      "endpoint": ENDPOINT,
      "host_style" path | virtual,
    },
    "acls": [
      {
        "type": id | email | uri,  # optional, default is id
        "source_id": ID,
        "dest_id": ID
      } ... ]
    }
  }
```

```

    "target_path": PATH # optional
  },
  "connections": [
  {
    "connection_id": ID,
    "access_key": ACCESS,
    "secret": SECRET,
    "endpoint": ENDPOINT,
    "host_style": path | virtual, # optional
  } ... ],
  "acl_profiles": [
  {
    "acls_id": ID, # acl mappings
    "acls": [ {
      "type": id | email | uri,
      "source_id": ID,
      "dest_id": ID
    } ... ]
  }
  ],
  "profiles": [
  {
    "source_bucket": SOURCE,
    "connection_id": CONNECTION_ID,
    "acls_id": MAPPINGS_ID,
    "target_path": DEST, # optional
  } ... ],
}

```

使用される設定用語の説明は次のとおりです。

### 接続

リモートクラウドサービスへの接続を表します。「connection\_id」、  
「access\_key」、「secret」、「endpoint」、および「host\_style」が含まれます。

### access\_key

特定の接続に使用されるリモートクラウドアクセスキー。

### secret

リモートクラウドサービスの秘密鍵。

### endpoint

リモートクラウドサービスエンドポイントのURL。

### host\_style

リモートクラウドエンドポイントにアクセスする際に使用されるホストスタイルのタイプ(「path」または「virtual」)。デフォルトは「path」です。

## **acls**

アクセスリストマッピングの配列。

## **acl\_mapping**

各「acl\_mapping」構造には、「type」、「source\_id」、および「dest\_id」が含まれます。これらは、各オブジェクトのACL変換を定義します。ACL変換により、ソースユーザIDを宛先IDに変換できます。

## **type**

ACLのタイプ: 「id」はユーザIDを定義し、「email」は電子メールでユーザを定義し、「uri」はuri (グループ)でユーザを定義します。

## **source\_id**

ソースゾーンでのユーザのID。

## **dest\_id**

宛先でのユーザのID。

## **target\_path**

ターゲットパスの作成方法を定義する文字列。ターゲットパスは、ソースオブジェクト名の追加先のプレフィックスを指定します。ターゲットパスの設定可能項目には、次の任意の変数を含めることができます。

## **SID**

同期インスタンスIDを表す固有の文字列。

## **ZONEGROUP**

ゾーングループの名前。

## **ZONEGROUP\_ID**

ゾーングループのID。

## **ZONE**

ゾーンの名前。

## **ZONE\_ID**

ゾーンのID。

## **BUCKET**

ソースバケットの名前。

## **OWNER**

ソースバケット所有者のID。

例: target\_path = rgwx-ZONE-SID/OWNER/BUCKET

## **acl\_profiles**

アクセスリストプロファイルの配列。

## **acl\_profile**

各プロファイルには、プロファイルを表す「acls\_id」と、「acl\_mapping」のリストを格納する「acls」配列が含まれます。

## **プロファイル**

プロファイルのリスト。各プロファイルには以下が含まれます。

### **source\_bucket**

バケット名、またはこのプロファイルのソースバケットを定義するバケットプレフィックス(\*で終わる場合)のいずれか。

### **target\_path**

説明については上記を参照。

### **connection\_id**

このプロファイルに使用する接続のID。

### **acls\_id**

このプロファイルに使用するACLのプロファイルのID。

## 21.8.4.2 S3固有の設定

クラウド同期モジュールは、AWS S3と互換性のあるバックエンドでのみ機能します。S3クラウドサービスにアクセスする場合、その動作を微調整するために使用できる設定可能項目がいくつかあります。

```
{
  "multipart_sync_threshold": OBJECT_SIZE,
  "multipart_min_part_size": PART_SIZE
}
```

### **multipart\_sync\_threshold**

サイズがこの値以上のオブジェクトは、マルチパートアップロードを使用してクラウドサービスと同期されます。

### **multipart\_min\_part\_size**

マルチパートアップロードを使用してオブジェクトを同期する際に使用する最小パーツサイズ。

## 21.8.5 アーカイブ同期モジュール

**アーカイブ同期モジュール**は、Object GatewayのS3オブジェクトのバージョン管理機能を利用します。**アーカイブゾーン**を設定することで、時間の経過とともに他のゾーンで異なるバージョンのS3オブジェクトが発生した場合にそれらをキャプチャできます。アーカイブゾーンが保持するバージョンの履歴は、アーカイブゾーンに関連付けられているゲートウェイを介してのみ削減できます。

このようなアーキテクチャにより、バージョン管理されていない複数のゾーンがゾーンゲートウェイを介してデータとメタデータをミラーリングし、エンドユーザーに高可用性を提供できると同時に、アーカイブゾーンはすべてのデータ更新をキャプチャし、それらをS3オブジェクトのバージョンとして統合します。

マルチゾーン設定にアーカイブゾーンを含めることにより、一方のゾーンでS3オブジェクト履歴の柔軟性を利用しながら、残りのゾーンでは、バージョン管理されたS3オブジェクトのレプリカが使用する領域を節約できます。

### 21.8.5.1 アーカイブ同期モジュールの設定



#### ヒント: 詳細

マルチサイトゲートウェイの設定の詳細については、[21.13項「マルチサイトObject Gateway」](#)を参照してください。

同期モジュールの設定の詳細については、[21.8項「同期モジュール」](#)を参照してください。

アーカイブ同期モジュールを使用するには、層タイプが`archive`に設定された新しいゾーンを作成する必要があります。

```
cephuser@adm > radosgw-admin zone create --rgw-zonegroup=ZONE_GROUP_NAME \
--rgw-zone=OGW_ZONE_NAME \
--endpoints=http://OGW_ENDPOINT1_URL[,http://OGW_ENDPOINT2_URL,...]
--tier-type=archive
```

## 21.9 LDAP 認証

デフォルトのローカルユーザ認証とは別に、Object GatewayでLDAPサーバのサービスを使用してユーザを認証することもできます。

## 21.9.1 認証メカニズム

Object GatewayがトークンからユーザのLDAP資格情報を抽出します。ユーザ名から検索フィルタが構成されます。Object Gatewayは、設定済みのユーザアカウントを使用して、一致するエントリをディレクトリで検索します。エントリが見つかった場合、Object Gatewayは、トークンから抽出したパスワードを使用して、見つかった識別名へのバインドを試みます。資格情報が有効であれば、バインドが成功し、Object Gatewayはアクセスを許可します。

許可するユーザを制限するには、検索ベースを特定の部門に設定するか、カスタム検索フィルタを指定します。たとえば、特定のグループメンバーシップ、カスタムオブジェクトクラス、またはカスタム属性を要求できます。

## 21.9.2 要件

- 「LDAPまたはActive Directory」 : Object Gatewayがアクセス可能な動作中のLDAPインスタンス。
- 「サービスアカウント」 : Object Gatewayが検索許可と共に使用するLDAP資格情報。
- 「ユーザアカウント」 : LDAPディレクトリ内の1つ以上のユーザアカウント。



### 重要: LDAPユーザとローカルユーザを重複させない

ローカルユーザの名前と、LDAPを使用して認証するユーザの名前に同じユーザ名を使用することはできません。Object Gatewayはこれらのユーザを区別できず、同じユーザとして扱います。



### ヒント: 正常性チェック

サービスアカウントまたはLDAP接続を確認するには、**ldapsearch**ユーティリティを使用します。例:

```
tux > ldapsearch -x -D "uid=ceph,ou=system,dc=example,dc=com" -W \  
-H ldaps://example.com -b "ou=users,dc=example,dc=com" 'uid=*' dn
```

想定される問題を排除するため、必ずCephの設定ファイルと同じLDAPパラメータを使用してください。

## 21.9.3 LDAP認証を使用するためのObject Gatewayの設定

次のパラメータはLDAP認証と関連があります。

### rgw\_ldap\_uri

使用するLDAPサーバを指定します。プレーンテキストの資格情報がオープンに転送されるのを避けるため、必ず ldaps://FQDN:PORT パラメータを使用してください。

### rgw\_ldap\_binddn

Object Gatewayが使用するサービスアカウントの識別名(DN)。

### rgw\_ldap\_secret

サービスアカウントのパスワード。

### rgw\_ldap\_searchdn

ユーザを検索するための、ディレクトリ情報ツリーのベースを指定します。users部門または具体的なOU(部門)にすることができます。

### rgw\_ldap\_dnattr

ユーザ名を照合するために、構成された検索フィルタで使用される属性。DIT (ディレクトリ情報ツリー)に応じて、uidまたはcnになります。

### rgw\_search\_filter

指定されていない場合、Object Gatewayは自動的に rgw\_ldap\_dnattr 設定を使用して検索フィルタを構成します。このパラメータは、許可するユーザのリストを非常に柔軟な方法で絞り込む場合に使用します。詳細については、[21.9.4項「カスタム検索フィルタを使用したユーザアクセスの制限」](#)を参照してください。

## 21.9.4 カスタム検索フィルタを使用したユーザアクセスの制限

rgw\_search\_filter パラメータは2つの方法で使用できます。

### 21.9.4.1 構成された検索フィルタをさらに制限するための部分フィルタ

次に、部分フィルタの例を示します。

```
"objectclass=inetorgperson"
```

Object Gatewayは、トークンから抽出したユーザ名と値`rgw_ldap_dnattr`を使用して通常の方法で検索フィルタを生成します。続いて、構成されたフィルタが`rgw_search_filter`属性の部分フィルタと結合されます。ユーザ名と設定によっては、最終的な検索フィルタは次のようになります。

```
"(&(uid=hari)(objectclass=inetorgperson))"
```

この場合、ユーザ「hari」がLDAPディレクトリで見つかり、オブジェクトクラス「inetorgperson」を持っていて、有効なパスワードを指定したときにのみ、このユーザにアクセスが許可されます。

#### 21.9.4.2 完全なフィルタ

完全なフィルタには、認証試行中にユーザ名に置き換えられるUSERNAMEトークンが含まれる必要があります。この場合、`rgw_ldap_dnattr`パラメータは使用されなくなります。たとえば、有効なユーザを特定のグループに制限するには、次のフィルタを使用します。

```
"(&(uid=USERNAME)(memberOf=cn=ceph-users,ou=groups,dc=mycompany,dc=com))"
```



#### 注記: memberOf属性

LDAP検索で`memberOf`属性を使用するには、特定のLDAPサーバ実装からのサーバサイドのサポートが必要です。

### 21.9.5 LDAP認証用アクセストークンの生成

`radosgw-token`ユーティリティは、LDAPユーザ名とパスワードに基づいてアクセストークンを生成します。実際のアクセストークンであるBase-64エンコード文字列を出力します。好みのS3クライアント(21.5.1項「Object Gatewayへのアクセス」を参照)を使用し、このトークンをアクセスキーとして指定し、空の秘密鍵を使用します。

```
tux > export RGW_ACCESS_KEY_ID="USERNAME"
tux > export RGW_SECRET_ACCESS_KEY="PASSWORD"
cephuser@adm > radosgw-token --encode --ttype=ldap
```



#### 重要: クリアテキスト資格情報

アクセストークンはBase-64エンコードのJSON構造で、LDAP資格情報がクリアテキストで含まれます。



## 注記: Active Directory

Active Directoryでは、`--ttype=ad`パラメータを使用します。

## 21.10 バケットインデックスのシャーディング

Object Gatewayは、バケットインデックスデータをインデックスプール(デフォルトでは`.rgw.buckets.index`)に保存します。1つのバケットに配置するオブジェクトが多すぎる(数十万個)場合、バケットあたりのオブジェクトの最大数のクォータ(`rgw bucket default quota max objects`)を設定しないと、インデックスプールのパフォーマンスが低下することがあります。「バケットインデックスのシャーディング」は、このようなパフォーマンス低下を防止し、各バケットで大量のオブジェクトを使用できるようになります。

### 21.10.1 バケットインデックスのリシャーディング

バケットが大容量になり、初期設定が十分ではなくなった場合、バケットのインデックスプールをリシャーディングする必要があります。オンラインの自動バケットインデックスリシャーディングを使用することも(21.10.1.1項「動的リシャーディング」を参照)、バケットインデックスをオフラインで手動でリシャーディングすることもできます(21.10.1.2項「手動リシャーディング」を参照)。

#### 21.10.1.1 動的リシャーディング

SUSE Enterprise Storage 5から、オンラインのバケットリシャーディングがサポートされています。これは、バケットあたりのオブジェクト数が一定のしきい値に達しているかどうかを検出し、バケットインデックスで使用されるシャードの数を自動的に増やします。このプロセスにより、各バケットインデックスシャードのエントリの数が減ります。

検出プロセスは次の条件で実行されます。

- バケットに新しいオブジェクトが追加された場合。
- すべてのバケットを定期的にスキャンするバックグラウンドプロセス内。これは、更新されない既存のバケットに対応するために必要です。

リシャーディングが必要なバケットは`reshard_log`キューに追加され、後でリシャーディングするようスケジュールされます。リシャードスレッドはバックグラウンドで動作し、スケジュールされたリシャーディングを一度に1つずつ実行します。

## 動的リシャードニングの設定

### rgw\_dynamic\_resharding

動的バケットインデックスリシャードニングを有効/無効にします。設定可能な値は「true」または「false」です。デフォルトは「true」です。

### rgw\_reshard\_num\_logs

リシャードニングログの対象にするシャードの数。デフォルトは16です。

### rgw\_reshard\_bucket\_lock\_duration

リシャードニング中のバケットオブジェクトに対するロック期間。デフォルトは120秒です。

### rgw\_max\_objs\_per\_shard

バケットインデックスシャードあたりのオブジェクトの最大数。デフォルトは100000オブジェクトです。

### rgw\_reshard\_thread\_interval

リシャードスレッド処理のラウンド間の最大時間。デフォルトは600秒です。

## リシャードニングプロセスを管理するためのコマンド

### リシャードニングキューへのバケットの追加

```
cephuser@adm > radosgw-admin reshard add \  
--bucket BUCKET_NAME \  
--num-shards NEW_NUMBER_OF_SHARDS
```

### リシャードニングキューの一覧

```
cephuser@adm > radosgw-admin reshard list
```

### バケットリシャードニングの処理/スケジュール

```
cephuser@adm > radosgw-admin reshard process
```

### バケットリシャードニングの状態の表示

```
cephuser@adm > radosgw-admin reshard status --bucket BUCKET_NAME
```

### 保留中のバケットリシャードニングのキャンセル

```
cephuser@adm > radosgw-admin reshard cancel --bucket BUCKET_NAME
```

## 21.10.1.2 手動リシャーディング

21.10.1.1項「動的リシャーディング」で説明されている動的リシャーディングは、単純な Object Gateway設定でのみサポートされます。マルチサイト設定の場合は、このセクションで説明する手動リシャーディングを使用します。

バケットインデックスをオフラインで手動でリシャーディングするには、次のコマンドを使用します。

```
cephuser@adm > radosgw-admin bucket reshard
```

**bucket reshard** コマンドは次の処理を実行します。

- 指定したオブジェクトに対してバケットインデックスオブジェクトの新しいセットを作成する
- これらのインデックスオブジェクトのすべてのエントリを分散する
- 新しいバケットインスタンスを作成する
- 新しいバケットインスタンスをバケットにリンクし、すべての新規インデックス操作が新しいバケットインデックスを経由するようにする
- 新旧のバケットIDを標準出力に出力する



### ヒント

多数のシャードを選択する場合、以下に注意をし、シャードあたり100000エントリ以下を目指してください。素数であるバケットインデックスのシャードは、シャード間で均等に分散しているバケットインデックスエントリで優れた動作になる傾向があります。たとえば、503個のバケットインデックスのシャードは素数であるため、500個のシャードよりも優れています。

#### 手順 21.1: バケットインデックスのリシャーディング

1. バケットに対するすべての操作が停止していることを確認します。
2. 元のバケットインデックスをバックアップします。

```
cephuser@adm > radosgw-admin bi list \  
--bucket=BUCKET_NAME \  
> BUCKET_NAME.list.backup
```

3. バケットインデックスをリシャーディングします。

```
cephuser@adm > radosgw-admin bucket reshard \  
--bucket=BUCKET_NAME \  
--num-shards=NEW_SHARDS_NUMBER
```



## ヒント: 古いバケットID

このコマンドは、出力の一部として新旧のバケットIDも出力します。

## 21.10.2 新しいバケットのバケットインデックスシャーディング

バケットインデックスシャーディングを制御するオプションは2つあります。

- 単純な設定の場合は、rgw\_override\_bucket\_index\_max\_shardsオプションを使用します。
- マルチサイト設定の場合は、bucket\_index\_max\_shardsオプションを使用します。

これらのオプションを0に設定すると、バケットインデックスシャーディングが無効になります。0より大きい値にすると、バケットインデックスシャーディングが有効になり、シャードの最大数が設定されます。

シャードの推奨数を計算するには、次の式が役立ちます。

```
number_of_objects_expected_in_a_bucket / 100000
```

シャードの最大数は7877であることに注意してください。

### 21.10.2.1 マルチサイト設定

マルチサイト設定では、フェールオーバーを管理するために別のインデックスプールを設定できます。1つのゾングループ内のゾーン全体に一貫したシャード数を設定するには、ゾングループの設定でbucket\_index\_max\_shardsオプションを設定します。

1. ゾングループの設定を zonegroup.json ファイルにエクスポートします。

```
cephuser@adm > radosgw-admin zonegroup get > zonegroup.json
```

2. zonegroup.json ファイルを編集して、指定した各ゾーンに対して bucket\_index\_max\_shards オプションを設定します。

3. ゾーングループをリセットします。

```
cephuser@adm > radosgw-admin zonegroup set < zonegroup.json
```

4. ピリオドを更新します。21.13.2.6項「ピリオドの更新」を参照してください。

## 21.11 OpenStack Keystoneの統合

OpenStack Keystoneは、OpenStack製品の識別情報サービスです。Object GatewayをKeystoneと統合して、Keystoneの認証トークンを受け付けるゲートウェイを設定できます。Keystoneによってゲートウェイにアクセスする権限が付与されたユーザは、Ceph Object Gateway側で確認され、必要であれば自動的に作成されます。Object Gatewayは、取り消されたトークンのリストを定期的にKeystoneに問い合わせます。

### 21.11.1 OpenStackの設定

Ceph Object Gatewayを設定する前に、Swiftサービスを有効にしてCeph Object Gatewayを指すようにOpenStack Keystoneを設定する必要があります。

1. Swiftサービスを設定します。OpenStackを使用してSwiftユーザを検証するには、まずSwiftサービスを作成します。

```
tux > openstack service create \  
  --name=swift \  
  --description="Swift Service" \  
  object-store
```

2. エンドポイントを設定します。Swiftサービスを作成した後、Ceph Object Gatewayを指すようにします。REGION\_NAMEは、ゲートウェイのゾーングループ名またはリージョン名に置き換えます。

```
tux > openstack endpoint create --region REGION_NAME \  
  --publicurl "http://radosgw.example.com:8080/swift/v1" \  
  --adminurl "http://radosgw.example.com:8080/swift/v1" \  
  --internalurl "http://radosgw.example.com:8080/swift/v1" \  
  swift
```

3. 設定を確認します。Swiftサービスを作成してエンドポイントを設定した後、エンドポイントを表示して、すべての設定が正しいことを確認します。

```
tux > openstack endpoint show object-store
```

## 21.11.2 Ceph Object Gatewayの設定

### 21.11.2.1 SSL証明書の設定

Ceph Object Gatewayは、取り消されたトークンのリストを定期的にKeystoneに問い合わせます。これらの要求はエンコードおよび署名されています。同様にエンコードおよび署名された自己署名トークンを提供するようにKeystoneを設定することもできます。これらの署名されたメッセージをデコードして検証できるようゲートウェイを設定する必要があります。したがって、Keystoneが要求を作成するために使用するOpenSSL証明書を「nss db」フォーマットに変換する必要があります。

```
root # mkdir /var/ceph/nss
root # openssl x509 -in /etc/keystone/ssl/certs/ca.pem \
    -pubkey | certutil -d /var/ceph/nss -A -n ca -t "TCu,Cu,Tuw"
root # openssl x509 -in /etc/keystone/ssl/certs/signing_cert.pem \
    -pubkey | certutil -A -d /var/ceph/nss -n signing_cert -t "P,P,P"
```

Ceph Object GatewayがOpenStack Keystoneと対話できるようにするため、OpenStack Keystoneで自己署名SSL証明書を使用できます。Ceph Object Gatewayが実行されているノードにKeystoneのSSL証明書をインストールするか、オプション`rgw keystone verify ssl`の値を「false」に設定します。`rgw keystone verify ssl`を「false」に設定すると、ゲートウェイが証明書の検証を試行しないことを意味します。

### 21.11.2.2 Object Gatewayのオプションの設定

次のオプションを使用してKeystone統合を設定できます。

`rgw keystone api version`

Keystone APIのバージョン。有効なオプションは2または3です。デフォルトは2です。

`rgw keystone url`

Keystoneサーバの管理RESTful APIのURLとポート番号。`SERVER_URL:PORT_NUMBER`というパターンに従います。

`rgw keystone admin token`

管理要求に対してKeystone内部で設定されるトークンと共有シークレット。

`rgw keystone accepted roles`

要求を実行するために必要な役割。デフォルトは「Member, admin」です。

rgw keystone accepted admin roles

ユーザが管理特権を得られるようにする役割のリスト。

rgw keystone token cache size

Keystoneトークンキャッシュのエントリの最大数。

rgw keystone revocation interval

拒否されたトークンを確認するまでの秒数。デフォルトは15 \* 60です。

rgw keystone implicit tenants

新しいユーザを同じ名前の専用のテナント内に作成します。デフォルトは「false」です。

rgw s3 auth use keystone

「true」に設定すると、Ceph Object GatewayはKeystoneを使用してユーザを認証します。デフォルトは「false」です。

nss db path

NSSデータベースのパス。

OpenStackサービスの設定と同様の方法で、Keystoneサービステナント、およびKeystoneのユーザとパスワード(OpenStack Identity APIのバージョン2.0の場合)を設定することもできます。これにより、設定ファイルで共有シークレット rgw keystone admin token を設定するのを避けることができます。このような設定は運用環境では無効にする必要があります。サービステナントの資格情報には管理者特権を含める必要があります。詳細については、[公式のOpenStack Keystoneのドキュメント \(https://docs.openstack.org/keystone/latest/#setting-up-projects-users-and-roles\)](https://docs.openstack.org/keystone/latest/#setting-up-projects-users-and-roles) を参照してください。関連する設定オプションは次のとおりです。

rgw keystone admin user

Keystone管理者ユーザの名前。

rgw keystone admin password

Keystone管理者ユーザのパスワード。

rgw keystone admin tenant

Keystoneバージョン2.0の管理者ユーザのテナント。

Ceph Object GatewayのユーザはKeystoneテナントにマップされます。Keystoneユーザには、多くの場合、複数のテナントで複数の役割が割り当てられます。Ceph Object Gatewayは、チケットを取得すると、そのチケットに割り当てられているテナントとユーザの役割を確認し、`rgw keystone accepted roles`オプションの設定に従って要求を受け入れるか拒否します。



## ヒント: OpenStackテナントのマッピング

SwiftテナントはデフォルトでObject Gatewayユーザにマップされますが、`rgw keystone implicit tenants`オプションを使用してOpenStackテナントにマップすることもできます。これにより、コンテナは、Object GatewayのデフォルトであるS3同様のグローバルネームスペースではなく、テナントのネームスペースを使用ようになります。混乱を避けるため、計画段階でマッピング方法を決定することをお勧めします。その理由は、後でこのオプションを切り替えた場合、テナント下にマッピングされた新しい要求のみが対象となり、前に作成された古いバケットはグローバルネームスペースに存在し続けるためです。

OpenStack Identity APIのバージョン3では、`rgw keystone admin tenant`オプションを次の内容に置き換える必要があります。

`rgw keystone admin domain`

Keystone管理者ユーザのドメイン。

`rgw keystone admin project`

Keystone管理者ユーザのプロジェクト。

## 21.12 プールの配置とストレージクラス

### 21.12.1 配置ターゲットの表示

配置ターゲットは、特定のバケットにどのプールを関連付けるかを制御します。バケットの配置ターゲットは作成時に選択し、変更することはできません。次のコマンドを実行して、その`placement_rule`を表示できます。

```
cephuser@adm > radosgw-admin bucket stats
```

ゾーングループ設定には、「default-placement」という名前の初期ターゲットが設定された配置ターゲットのリストが含まれています。ゾーン設定により、各ゾーングループの配置ターゲットの名前がそのローカルストレージにマップされます。このゾーン配置情報には、バケットインデックスの「index\_pool」の名前、不完全なマルチパートアップロードに関するメタデータの「data\_extra\_pool」の名前、および各ストレージクラスの「data\_pool」の名前が含まれています。

## 21.12.2 ストレージクラス

ストレージクラスは、オブジェクトデータの配置をカスタマイズするのに役立ちます。S3バケットのライフサイクルルールを使用すると、ストレージクラス間でのオブジェクトの移行を自動化できます。

ストレージクラスは、配置ターゲットの観点から定義します。各ゾーングループ配置ターゲットには、使用可能なストレージクラスが「STANDARD」という名前の初期クラスで一覧にされます。ゾーン設定は、各ゾーングループのストレージクラスに「data\_pool」プール名を提供する処理を受け持ちます。

## 21.12.3 ゾーングループおよびゾーンの設定

ゾーングループおよびゾーンに対して **radosgw-admin** コマンドを使用し、その配置を設定します。次のコマンドを使用して、ゾーングループ配置設定を問い合わせることができます。

```
cephuser@adm > radosgw-admin zonegroup get
{
  "id": "ab01123f-e0df-4f29-9d71-b44888d67cd5",
  "name": "default",
  "api_name": "default",
  ...
  "placement_targets": [
    {
      "name": "default-placement",
      "tags": [],
      "storage_classes": [
        "STANDARD"
      ]
    }
  ],
  "default_placement": "default-placement",
  ...
}
```

ゾーン配置設定を問い合わせるには、次のコマンドを実行します。

```
cephuser@adm > radosgw-admin zone get
{
  "id": "557cdcee-3aae-4e9e-85c7-2f86f5eddb1f",
  "name": "default",
  "domain_root": "default.rgw.meta:root",
  ...
  "placement_pools": [
    {
      "key": "default-placement",
      "val": {
        "index_pool": "default.rgw.buckets.index",
        "storage_classes": {
          "STANDARD": {
            "data_pool": "default.rgw.buckets.data"
          }
        },
        "data_extra_pool": "default.rgw.buckets.non-ec",
        "index_type": 0
      }
    }
  ],
  ...
}
```



### 注記: 以前のマルチサイト設定がない場合

以前にマルチサイト設定を実行したことがない場合は、「デフォルト」のゾーンとゾーングループが自動的に作成され、このゾーン/ゾーングループへの変更はCeph Object Gatewaysを再起動するまで有効になりません。マルチサイトのレルムを作成している場合は、ゾーン/ゾーングループの変更は、**`radosgw-admin period update --commit`** コマンドで変更をコミットした後で有効になります。

#### 21.12.3.1 配置ターゲットの追加

「temporary」という名前の新しい配置ターゲットを作成するには、まずそれをゾーングループに追加します。

```
cephuser@adm > radosgw-admin zonegroup placement add \
  --rgw-zonegroup default \
  --placement-id temporary
```

次に、そのターゲットのゾーン配置情報を指定します。

```
cephuser@adm > radosgw-admin zone placement add \
  --rgw-zone default \
```

```
--placement-id temporary \  
--data-pool default.rgw temporary.data \  
--index-pool default.rgw temporary.index \  
--data-extra-pool default.rgw temporary.non-ec
```

### 21.12.3.2 ストレージクラスの追加

「COLD」という名前の新しいストレージクラスを「default-placement」ターゲットに追加するには、まずそれをゾーングループに追加します。

```
cephuser@adm > radosgw-admin zonegroup placement add \  
--rgw-zonegroup default \  
--placement-id default-placement \  
--storage-class COLD
```

次に、そのストレージクラスのゾーン配置情報を指定します。

```
cephuser@adm > radosgw-admin zone placement add \  
--rgw-zone default \  
--placement-id default-placement \  
--storage-class COLD \  
--data-pool default.rgw.cold.data \  
--compression lz4
```

## 21.12.4 配置のカスタマイズ

### 21.12.4.1 ゾーングループのデフォルトの配置の編集

デフォルトでは、新しいバケットはそのゾーングループのdefault\_placementターゲットを使用します。次のコマンドを使用して、このゾーングループ設定を変更できます。

```
cephuser@adm > radosgw-admin zonegroup placement default \  
--rgw-zonegroup default \  
--placement-id new-placement
```

### 21.12.4.2 ユーザのデフォルトの配置の編集

Ceph Object Gatewayユーザは、ユーザ情報に空ではないdefault\_placementフィールドを設定することにより、ゾーングループのデフォルトの配置ターゲットを上書きすることができます。同様に、default\_storage\_classを設定すると、デフォルトでオブジェクトに適用されるSTANDARDストレージクラスを上書きできます。

```
cephuser@adm > radosgw-admin user info --uid testid
{
  ...
  "default_placement": "",
  "default_storage_class": "",
  "placement_tags": [],
  ...
}
```

ゾーングループの配置ターゲットにタグが含まれている場合、ユーザは、その配置ターゲットを使用してバケットを作成できません。ただし、そのユーザ情報の「placement\_tags」フィールドに、一致するタグが少なくとも1つ含まれている場合を除きます。これは、特定のタイプのストレージへのアクセスを制限するのに役立つ場合があります。

**radosgw-admin** コマンドでこれらのフィールドを直接変更することはできません。そのため、JSONフォーマットを手動で編集する必要があります。

```
cephuser@adm > radosgw-admin metadata get user:USER-ID > user.json
tux > vi user.json      # edit the file as required
cephuser@adm > radosgw-admin metadata put user:USER-ID < user.json
```

### 21.12.4.3 S3のデフォルトのバケットの配置の編集

S3プロトコルでバケットを作成する場合、配置ターゲットを LocationConstraint の一部として指定し、ユーザとゾーングループのデフォルトの配置ターゲットを上書きすることができます。

通常、LocationConstraint は、ゾーングループの api\_name に一致する必要があります。

```
<LocationConstraint>default</LocationConstraint>
```

カスタム配置ターゲットを、コロンに続く api\_name に追加できます。

```
<LocationConstraint>default:new-placement</LocationConstraint>
```

### 21.12.4.4 Swiftのバケットの配置の編集

Swiftプロトコルでバケットを作成する場合、HTTPヘッダの X-Storage-Policy で配置ターゲットを指定できます。

```
X-Storage-Policy: NEW-PLACEMENT
```

## 21.12.5 ストレージクラスの使用

すべての配置ターゲットには、新しいオブジェクトにデフォルトで適用されるSTANDARDストレージクラスがあります。このデフォルトを、その`default_storage_class`で上書きできません。

デフォルト以外のストレージクラスにオブジェクトを作成するには、要求のHTTPヘッダでそのストレージクラス名を指定します。S3プロトコルでは`X-Amz-Storage-Class`ヘッダを使用し、Swiftプロトコルでは`X-Object-Storage-Class`ヘッダを使用します。

Transitionアクションを使用してストレージクラス間でオブジェクトデータを移動するには、**S3オブジェクトライフサイクル管理**を使用できます。

## 21.13 マルチサイトObject Gateway

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

### マルチゾーン

ゾーングループと複数のゾーンから構成されている設定で、それぞれのゾーンに1つまたは複数の`ceph-radosgw`インスタンスがあります。それぞれのゾーンは、その独自のCeph Storage Clusterを利用しています。ゾーンの1つで重大な障害が発生した場合、ゾーングループ内の複数のゾーンがゾーングループに障害復旧を提供します。それぞれのゾーンがアクティブで、書き込み操作を受け付ける場合があります。障害復旧に加えて、複数のアクティブゾーンがコンテンツ配信ネットワークの基盤としても動作する場合があります。

### マルチゾーングループ

Ceph Object Gatewayは、複数のゾーングループをサポートしていて、それぞれのゾーングループに1つまたは複数のゾーンがあります。同じレルム内の1つのゾーングループのゾーンに別のゾーングループとして保存されているオブジェクトは、グローバルオブジェクトネームスペースを共有するため、ゾーングループおよびゾーン間で固有のオブジェクトIDになります。



### 注記

ゾーングループはそのゾーングループ内のメタデータのみを同期するというこ  
とに留意することが重要です。データおよびメタデータは、ゾーングループ内  
のゾーンの間で複製されます。データやメタデータはレルム間では共有されま  
せん。

## 複数のレルム

Ceph Object Gatewayは、グローバルに固有のネームスペースであるレルムという概念をサポートしています。1つまたは複数のゾーングループを含む複数のレルムがサポートされます。

アクティブ-アクティブゾーン設定で動作するようにそれぞれのObject Gatewayを設定し、非マスタゾーンへの書き込みを許可できます。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、および複数のエポックを含むピリオドを保存し、設定変更を追跡します。rgwデーモンが同期を処理するため、個別の同期エージェントは不要です。この同期方法では、Ceph Object Gatewayは、アクティブ-パッシブ設定ではなくアクティブ-アクティブ設定で動作できます。

### 21.13.1 要件と前提

マルチサイト設定では、2つ以上のCephストレージクラスタおよび2つ以上のCeph Object Gatewayインスタンス(各Cephストレージクラスタに1つ)が必要です。次の設定は、地理的に離れた場所に2つ以上のCephストレージクラスタがあることを想定しています。ただし、設定は同じサイトで動作できます。たとえば、rgw1とrgw2という名前であるとして。

マルチサイト設定では、マスタゾーングループおよびマスタゾーンが必要です。マスタゾーンは、マルチサイトクラスタのすべてのメタデータ操作に関する真実を語る資料です。また、それぞれのゾーングループにはマスタゾーンが必要です。ゾーングループには、1つまたは複数のセカンダリゾーンまたは非マスタゾーンがある場合があります。このガイドでは、rgw1ホストがマスタゾーングループのマスタゾーンとして動作し、rgw2ホストがマスタゾーングループのセカンダリゾーンとして動作します。

### 21.13.2 マスタゾーンの設定

マルチサイト設定のすべてのゲートウェイは、マスタゾーングループおよびマスタゾーン内のホストのceph-radosgwデーモンから設定を取得します。マルチサイト設定でゲートウェイを設定するには、ceph-radosgwインスタンスを選択してマスタゾーングループおよびマスタゾーンを設定します。

#### 21.13.2.1 レルムの作成

レルムは、1つまたは複数のゾーンを含む1つまたは複数のゾーングループから構成されるグローバルに固有のネームスペースを表します。ゾーンにはバケットが含まれていて、バケットにはオブジェクトが含まれています。レルムは、Ceph Object Gatewayを有効にし、同じハー

ドウェアの複数のネームスペースおよびその設定をサポートします。レルムにはピリオドの概念が含まれています。それぞれのピリオドは、ゾーングループおよびゾーンの設定の状態を時間で表します。ゾーングループまたはゾーンを変更するたびに、ピリオドを更新し、コミットします。デフォルトでは、Ceph Object Gatewayは、下位互換性を満たすためにレルムを作成しません。ベストプラクティスとして、新しいクラスタにはレルムを作成することをお勧めします。

マルチサイト設定用にgoldという新しいレルムを作成します。そのためには、マスタゾーングループおよびゾーンで使用するホストでコマンドラインインタフェースを開きます。次のコマンドを実行します。

```
cephuser@adm > radosgw-admin realm create --rgw-realm=gold --default
```

クラスタにレルムが1つある場合、`--default`フラグを指定します。`--default`が指定されると、**`radosgw-admin`**はデフォルトでこのレルムを使用します。`--default`を指定しない場合、ゾーングループおよびゾーンを追加するには、これらを追加するときに`--rgw-realm`フラグまたは`--realm-id`フラグのいずれかを指定する必要があります。

レルムを作成した後、**`radosgw-admin`**は、レルムの設定を返します。

```
{
  "id": "4a367026-bd8f-40ee-b486-8212482ddcd7",
  "name": "gold",
  "current_period": "09559832-67a4-4101-8b3f-10dfcd6b2707",
  "epoch": 1
}
```



## 注記

Cephは、レルム用に固有のIDを生成します。必要に応じてレルムの名前を変更できません。

### 21.13.2.2 マスタゾーングループの作成

レルムには、レルムのマスタゾーングループとして動作する1つ以上のゾーングループが必要です。マルチサイト設定用に新しいマスタゾーングループを作成します。そのためには、マスタゾーングループおよびゾーンで使用するホストでコマンドラインインタフェースを開きます。次のコマンドを実行してusというマスタゾーングループを作成します。

```
cephuser@adm > radosgw-admin zonegroup create --rgw-zonegroup=us \
--endpoints=http://rgw1:80 --master --default
```

レルムにゾーングループが1つしかない場合、`--default`フラグを指定します。`--default`が指定されると、**radosgw-admin**は、新しいゾーンを追加するときにデフォルトでこのゾーングループを使用します。`--default`を指定しない場合、ゾーンを追加するには、ゾーンを追加または変更するときに`--rgw-zonegroup`フラグまたは`--zonegroup-id`フラグのいずれかを指定してゾーングループを特定する必要があります。

マスタゾーングループを作成した後、**radosgw-admin**は、ゾーングループの設定を返します。例:

```
{
  "id": "d4018b8d-8c0d-4072-8919-608726fa369e",
  "name": "us",
  "api_name": "us",
  "is_master": "true",
  "endpoints": [
    "http://\//rgw1:80"
  ],
  "hostnames": [],
  "hostnames_s3website": [],
  "master_zone": "",
  "zones": [],
  "placement_targets": [],
  "default_placement": "",
  "realm_id": "4a367026-bd8f-40ee-b486-8212482ddcd7"
}
```

### 21.13.2.3 マスタゾーンの作成

#### ! 重要

ゾーンは、ゾーン内に配置するCeph Object Gatewayノードに作成する必要があります。

マルチサイト設定用に新しいマスタゾーンを作成します。そのためには、マスタゾーングループおよびゾーンで使用するホストでコマンドラインインタフェースを開きます。次のコマンドを実行します。

```
cephuser@adm > radosgw-admin zone create --rgw-zonegroup=us --rgw-zone=us-east-1 \
--endpoints=http://rgw1:80 --access-key=SYSTEM_ACCESS_KEY --secret=SYSTEM_SECRET_KEY
```



## 注記

上の例では、`--access-key`オプションと`--secret`オプションを指定していません。これらの設定は、次のセクションでユーザを作成したときにゾーンに追加されます。

マスタゾーンを作成した後、`radosgw-admin`は、ゾーンの設定を返します。例:

```
{
  "id": "56dfabbb-2f4e-4223-925e-de3c72de3866",
  "name": "us-east-1",
  "domain_root": "us-east-1.rgw.meta:root",
  "control_pool": "us-east-1.rgw.control",
  "gc_pool": "us-east-1.rgw.log:gc",
  "lc_pool": "us-east-1.rgw.log:lc",
  "log_pool": "us-east-1.rgw.log",
  "intent_log_pool": "us-east-1.rgw.log:intent",
  "usage_log_pool": "us-east-1.rgw.log:usage",
  "reshard_pool": "us-east-1.rgw.log:reshard",
  "user_keys_pool": "us-east-1.rgw.meta:users.keys",
  "user_email_pool": "us-east-1.rgw.meta:users.email",
  "user_swift_pool": "us-east-1.rgw.meta:users.swift",
  "user_uid_pool": "us-east-1.rgw.meta:users.uid",
  "otp_pool": "us-east-1.rgw.otp",
  "system_key": {
    "access_key": "1555b35654ad1656d804",
    "secret_key": "h7GhxuBLTrlhVUyxSPUKUV8r/2EI4ngqJxD7iBdBYLhwluN30JaT3Q=="
  },
  "placement_pools": [
    {
      "key": "us-east-1-placement",
      "val": {
        "index_pool": "us-east-1.rgw.buckets.index",
        "storage_classes": {
          "STANDARD": {
            "data_pool": "us-east-1.rgw.buckets.data"
          }
        },
        "data_extra_pool": "us-east-1.rgw.buckets.non-ec",
        "index_type": 0
      }
    }
  ],
  "metadata_heap": "",
  "realm_id": ""
}
```

## 21.13.2.4 デフォルトのゾーンとグループの削除

### ！ 重要

次の手順では、まだデータを保存していない新しくインストールしたシステムを使用するマルチサイト設定を想定しています。デフォルトのゾーンおよびそのプールを使用してデータを保存済みの場合、これらを**削除しないでください**。削除するとデータが削除され、回復できなくなります。

Object Gatewayをデフォルトインストールすると、`default`という名前のデフォルトのゾーングループが作成されます。デフォルトのゾーンがある場合、削除します。まず、デフォルトのゾーングループからデフォルトのゾーンを削除します。

```
cephuser@adm > radosgw-admin zonegroup delete --rgw-zonegroup=default
```

Cephストレージクラスタにデフォルトのプールがある場合、削除します。

### ！ 重要

次の手順では、現時点でデータが保存されていない新しくインストールしたシステムを使用するマルチサイト設定を想定しています。デフォルトのゾーングループを使用してデータを保存済みの場合、これらを**削除しないでください**。

```
cephuser@adm > ceph osd pool rm default.rgw.control default.rgw.control --yes-i-really-really-mean-it
cephuser@adm > ceph osd pool rm default.rgw.data.root default.rgw.data.root --yes-i-really-really-mean-it
cephuser@adm > ceph osd pool rm default.rgw.gc default.rgw.gc --yes-i-really-really-mean-it
cephuser@adm > ceph osd pool rm default.rgw.log default.rgw.log --yes-i-really-really-mean-it
cephuser@adm > ceph osd pool rm default.rgw.meta default.rgw.meta --yes-i-really-really-mean-it
```

### ⚠ 警告

デフォルトのゾーングループを削除すると、システムユーザも削除されます。管理ユーザキーが伝搬されない場合、CephダッシュボードのObject Gateway管理機能は動作しません。この手順を続行する場合、次のセクションに進み、システムユーザを再作成します。

### 21.13.2.5 システムユーザの作成

レームおよびピリオドの情報を取得するにはその前に、`ceph-radosgw`デーモンを認証する必要があります。マスタゾーンで、システムユーザを作成し、デーモン間の認証を簡略化します。

```
cephuser@adm > radosgw-admin user create --uid=zone.user \  
--display-name="Zone User" --access-key=SYSTEM_ACCESS_KEY \  
--secret=SYSTEM_SECRET_KEY --system
```

セカンダリゾーンでは`access_key`と`secret_key`をマスタゾーンで認証するため、これらをメモします。

システムユーザをマスタゾーンに追加します。

```
cephuser@adm > radosgw-admin zone modify --rgw-zone=us-east-1 \  
--access-key=ACCESS-KEY --secret=SECRET
```

ピリオドを更新して変更を有効にします。

```
cephuser@adm > radosgw-admin period update --commit
```

### 21.13.2.6 ピリオドの更新

マスタゾーンの設定を更新した後、ピリオドを更新します。

```
cephuser@adm > radosgw-admin period update --commit
```

ピリオドを更新した後、`radosgw-admin`は、ピリオドの設定を返します。例:

```
{  
  "id": "09559832-67a4-4101-8b3f-10dfcd6b2707", "epoch": 1, "predecessor_uuid": "",  
  "sync_status": [], "period_map":  
  {  
    "id": "09559832-67a4-4101-8b3f-10dfcd6b2707", "zonegroups": [], "short_zone_ids": []  
  }, "master_zonegroup": "", "master_zone": "", "period_config":  
  {  
    "bucket_quota": {  
      "enabled": false, "max_size_kb": -1, "max_objects": -1  
    }, "user_quota": {  
      "enabled": false, "max_size_kb": -1, "max_objects": -1  
    }  
  }, "realm_id": "4a367026-bd8f-40ee-b486-8212482ddcd7", "realm_name": "gold",  
  "realm_epoch": 1  
}
```



## 注記

ピリオドを更新すると、エポックが変更され、その他のゾーンが更新した設定を受け取るようになります。

### 21.13.2.7 Gatewayの起動

Object Gatewayホストで、Ceph Object Gatewayサービスを起動し、有効にします。クラスタ固有のFSIDを識別するには、**ceph fsid**を実行します。Object Gatewayのデーモン名を識別するには、**ceph orch ps --hostname HOSTNAME**を実行します。

```
cephuser@ogw > systemctl start ceph-FSID@DAEMON_NAME  
cephuser@ogw > systemctl enable ceph-FSID@DAEMON_NAME
```

### 21.13.3 セカンダリゾーンの設定

ゾーングループ内のゾーンは、各ゾーンに同じデータが存在するようにするため、すべてのデータを複製します。セカンダリゾーンを作成する場合は、セカンダリゾーンにサービスを提供するよう指定されたホスト上で次の操作をすべて実行します。



## 注記

3つ目のゾーンを追加するには、セカンダリゾーンの追加と同じ手順を実行します。異なるゾーン名を使用してください。



## 重要

ユーザ作成などのメタデータ操作は、マスタゾーン内のホストで実行する必要があります。マスタゾーンおよびセカンダリゾーンは、バケット操作を受け取ることができますが、セカンダリゾーンは、バケット操作をマスタゾーンにリダイレクトします。マスタゾーンがダウンしている場合、バケット操作は失敗します。

### 21.13.3.1 レルムのインポート

URLのパス、アクセスキー、およびマスタゾーングループのマスタゾーンの秘密を使用して、レルムの設定をホストにインポートします。デフォルト以外のレルムをインポートするには、`--rgw-realm`設定オプションまたは`--realm-id`設定オプションを使用してレルムを指定します。

```
cephuser@adm > radosgw-admin realm pull --url=url-to-master-zone-gateway --access-key=access-key --secret=secret
```



#### 注記

レルムをインポートすると、リモートホストの現在のピリオドの設定も取得され、また、それがこのホストの現在のピリオドになります。

このレルムがデフォルトのレルムまたは唯一のレルムの場合、そのレルムをデフォルトのレルムにします。

```
cephuser@adm > radosgw-admin realm default --rgw-realm=REALM-NAME
```

### 21.13.3.2 セカンダリゾーンの作成

マルチサイト設定用にセカンダリゾーンを作成します。そのためには、セカンダリゾーンで使用するホストでコマンドラインインタフェースを開きます。ゾーングループID、新しいゾーン名およびゾーンのエンドポイントを指定します。`--master`フラグは使用しないでください。デフォルトでは、すべてのゾーンがアクティブ-アクティブ設定で動作します。セカンダリゾーンが書き込み操作を受け付けられないようにする場合、`--read-only`フラグを指定して、マスタゾーンとセカンダリゾーンの間にアクティブ-パッシブ設定を作成します。また、マスタゾーングループのマスタゾーンに保存されている、生成済みのシステムユーザの`access_key`と`secret_key`を指定します。次のコマンドを実行します。

```
cephuser@adm > radosgw-admin zone create --rgw-zonegroup=ZONE-GROUP-NAME \
--rgw-zone=ZONE-NAME --endpoints=URL \
--access-key=SYSTEM-KEY --secret=SECRET \
--endpoints=http://FQDN:80 \
[--read-only]
```

例:

```
cephuser@adm > radosgw-admin zone create --rgw-zonegroup=us --endpoints=http://rgw2:80 \
--rgw-zone=us-east-2 --access-key=SYSTEM_ACCESS_KEY --secret=SYSTEM_SECRET_KEY \
{
```

```

"id": "950c1a43-6836-41a2-a161-64777e07e8b8",
"name": "us-east-2",
"domain_root": "us-east-2.rgw.data.root",
"control_pool": "us-east-2.rgw.control",
"gc_pool": "us-east-2.rgw.gc",
"log_pool": "us-east-2.rgw.log",
"intent_log_pool": "us-east-2.rgw.intent-log",
"usage_log_pool": "us-east-2.rgw.usage",
"user_keys_pool": "us-east-2.rgw.users.keys",
"user_email_pool": "us-east-2.rgw.users.email",
"user_swift_pool": "us-east-2.rgw.users.swift",
"user_uid_pool": "us-east-2.rgw.users.uid",
"system_key": {
  "access_key": "1555b35654ad1656d804",
  "secret_key": "h7GhxuBLTrlhVUyxSPUKUV8r\2EI4ngqJxD7iBdBYLhwluN30JaT3Q=="
},
"placement_pools": [
  {
    "key": "default-placement",
    "val": {
      "index_pool": "us-east-2.rgw.buckets.index",
      "data_pool": "us-east-2.rgw.buckets.data",
      "data_extra_pool": "us-east-2.rgw.buckets.non-ec",
      "index_type": 0
    }
  }
],
"metadata_heap": "us-east-2.rgw.meta",
"realm_id": "815d74c2-80d6-4e63-8cfc-232037f7ff5c"
}

```

## 重要

次の手順では、まだデータが保存されていない新しくインストールしたシステムを使用するマルチサイト設定を想定しています。デフォルトのゾーンおよびそのプールを使用してデータを保存済みの場合、これらを**削除しないでください**。削除するとデータが失われ、回復できなくなります。

必要に応じてデフォルトのゾーンを削除します。

```
cephuser@adm > radosgw-admin zone rm --rgw-zone=default
```

必要に応じてCephストレージクラスタのデフォルトのプールを削除します。

```

cephuser@adm > ceph osd pool rm default.rgw.control default.rgw.control --yes-i-really-really-mean-it
cephuser@adm > ceph osd pool rm default.rgw.data.root default.rgw.data.root --yes-i-really-really-mean-it

```

```
cephuser@adm > ceph osd pool rm default.rgw.gc default.rgw.gc --yes-i-really-really-mean-it
cephuser@adm > ceph osd pool rm default.rgw.log default.rgw.log --yes-i-really-really-mean-it
cephuser@adm > ceph osd pool rm default.rgw.users.uid default.rgw.users.uid --yes-i-really-really-mean-it
```

### 21.13.3.3 Ceph設定ファイルの更新

セカンダリゾーンのホストでCeph設定ファイルを更新します。そのためには、`rgw_zone`設定オプションとセカンダリゾーンの名前をインスタンスのエントリに追加します。

そのためには、次のコマンドを実行します。

```
cephuser@adm > ceph config set SERVICE_NAME rgw_zone us-west
```

### 21.13.3.4 ピリオドの更新

マスタゾーンの設定を更新した後、ピリオドを更新します。

```
cephuser@adm > radosgw-admin period update --commit
{
  "id": "b5e4d3ec-2a62-4746-b479-4b2bc14b27d1",
  "epoch": 2,
  "predecessor_uuid": "09559832-67a4-4101-8b3f-10dfcd6b2707",
  "sync_status": [ "[...]"
],
  "period_map": {
    "id": "b5e4d3ec-2a62-4746-b479-4b2bc14b27d1",
    "zonegroups": [
      {
        "id": "d4018b8d-8c0d-4072-8919-608726fa369e",
        "name": "us",
        "api_name": "us",
        "is_master": "true",
        "endpoints": [
          "http://\rgw1:80"
        ],
        "hostnames": [],
        "hostnames_s3website": [],
        "master_zone": "83859a9a-9901-4f00-aa6d-285c777e10f0",
        "zones": [
          {
            "id": "83859a9a-9901-4f00-aa6d-285c777e10f0",
            "name": "us-east-1",
            "endpoints": [
              "http://\rgw1:80"
            ]
          }
        ]
      }
    ]
  }
}
```

```

    ],
    "log_meta": "true",
    "log_data": "false",
    "bucket_index_max_shards": 0,
    "read_only": "false"
  },
  {
    "id": "950c1a43-6836-41a2-a161-64777e07e8b8",
    "name": "us-east-2",
    "endpoints": [
      "http://\rgw2:80"
    ],
    "log_meta": "false",
    "log_data": "true",
    "bucket_index_max_shards": 0,
    "read_only": "false"
  }
],
"placement_targets": [
  {
    "name": "default-placement",
    "tags": []
  }
],
"default_placement": "default-placement",
"realm_id": "4a367026-bd8f-40ee-b486-8212482ddcd7"
}
],
"short_zone_ids": [
  {
    "key": "83859a9a-9901-4f00-aa6d-285c777e10f0",
    "val": 630926044
  },
  {
    "key": "950c1a43-6836-41a2-a161-64777e07e8b8",
    "val": 4276257543
  }
]
},
"master_zonegroup": "d4018b8d-8c0d-4072-8919-608726fa369e",
"master_zone": "83859a9a-9901-4f00-aa6d-285c777e10f0",
"period_config": {
  "bucket_quota": {
    "enabled": false,
    "max_size_kb": -1,
    "max_objects": -1
  },
  "user_quota": {
    "enabled": false,

```

```

        "max_size_kb": -1,
        "max_objects": -1
    },
    "realm_id": "4a367026-bd8f-40ee-b486-8212482ddcd7",
    "realm_name": "gold",
    "realm_epoch": 2
}

```



## 注記

ピリオドを更新すると、エポックが変更され、その他のゾーンが更新した設定を受け取るようになります。

### 21.13.3.5 Object Gatewayの起動

Object Gatewayホストで、Ceph Object Gatewayサービスを起動し、有効にします。

```
cephuser@adm > ceph orch start rgw.us-east-2
```

### 21.13.3.6 同期の状態の確認

セカンダリゾーンが稼働しているときに、同期の状態を確認します。同期では、マスタゾーンで作成したユーザとバケットがセカンダリゾーンにコピーされます。

```
cephuser@adm > radosgw-admin sync status
```

同期操作の状態が出力に表示されます。例:

```

realm f3239bc5-e1a8-4206-a81d-e1576480804d (gold)
  zonegroup c50dbb7e-d9ce-47cc-a8bb-97d9b399d388 (us)
    zone 4c453b70-4a16-4ce8-8185-1893b05d346e (us-west)
metadata sync syncing
  full sync: 0/64 shards
  metadata is caught up with master
  incremental sync: 64/64 shards
data sync source: lee9da3e-114d-4ae3-a8a4-056e8a17f532 (us-east)
  syncing
  full sync: 0/128 shards
  incremental sync: 128/128 shards
  data is caught up with source

```



## 注記

セカンダリゾーンは、バケット操作を受け付けますが、これをマスタゾーンにリダイレクトし、マスタゾーンと同期を取り、バケット操作の結果を受け取ります。マスタゾーンがダウンしている場合、セカンダリゾーンで実行されたバケット操作は失敗しますが、オブジェクト操作は成功します。

## 21.13.4 Object Gatewayの一般的な保守

### 21.13.4.1 同期の状態の確認

次のコマンドを使用して、ゾーンの複製の状態に関する情報を問い合わせることができます。

```
cephuser@adm > radosgw-admin sync status
  realm b3bc1c37-9c44-4b89-a03b-04c269bea5da (gold)
  zonegroup f54f9b22-b4b6-4a0e-9211-fa6ac1693f49 (us)
    zone adce11c9-b8ed-4a90-8bc5-3fc029ff0816 (us-west)
      metadata sync syncing
        full sync: 0/64 shards
        incremental sync: 64/64 shards
        metadata is behind on 1 shards
        oldest incremental change not applied: 2017-03-22 10:20:00.0.881361s
  data sync source: 341c2d81-4574-4d08-ab0f-5a2a7b168028 (us-east)
    syncing
      full sync: 0/128 shards
      incremental sync: 128/128 shards
      data is caught up with source
    source: 3b5d1a3f-3f27-4e4a-8f34-6072d4bb1275 (us-3)
      syncing
        full sync: 0/128 shards
        incremental sync: 128/128 shards
        data is caught up with source
```

### 21.13.4.2 メタデータマスタゾーンの変更



## 重要

メタデータマスタにするゾーンを変更する際には注意してください。ゾーンで現在のマスタゾーンからのメタデータの同期が完了していない場合、マスタに昇格するときに残っていたエントリを操作できず、これらの変更は失われます。そのため、ゾーン

の **radosgw-admin** の同期の状態がメタデータの同期に追いつくまで待機してからそのゾーンをマスタに昇格することをお勧めします。同様に、メタデータの変更が現在のマスタゾーンで処理されていて、別のゾーンがマスタに昇格している場合、これらの変更は失われる可能性が高いです。これを回避するには、前のマスタゾーンの Object Gateway インスタンスをシャットダウンすることをお勧めします。別のゾーンを昇格した後、その新しいピリオドは **radosgw-admin** ピリオドの取得によってフェッチでき、そのゲートウェイを再起動できます。

ゾーン(たとえばゾングループ **us** のゾーン **us-west**)をメタデータマスタに昇格するには、そのゾーンで次のコマンドを実行します。

```
cephuser@ogw > radosgw-admin zone modify --rgw-zone=us-west --master
cephuser@ogw > radosgw-admin zonegroup modify --rgw-zonegroup=us --master
cephuser@ogw > radosgw-admin period update --commit
```

新しいピリオドが生成され、ゾーン **us-west** の Object Gateway インスタンスによって、このピリオドがその他のゾーンに送信されます。

### 21.13.5 フェールオーバーと障害復旧機能を提供

マスタゾーンに障害が発生した場合、障害復旧のためにセカンダリゾーンにフェールオーバーします。

1. セカンダリゾーンをマスタおよびデフォルトのゾーンにします。例:

```
cephuser@adm > radosgw-admin zone modify --rgw-zone=ZONE-NAME --master --default
```

デフォルトでは、Ceph Object Gateway はアクティブ-アクティブ設定で動作します。クラスタがアクティブ-パッシブ設定で動作するように設定されていた場合、セカンダリゾーンは読み込み専用ゾーンになっています。 **--read-only** の状態を削除して、このゾーンが書き込み操作を受け付けることができるようにします。例:

```
cephuser@adm > radosgw-admin zone modify --rgw-zone=ZONE-NAME --master --default \
--read-only=false
```

2. ピリオドを更新して変更を有効にします。

```
cephuser@adm > radosgw-admin period update --commit
```

3. Ceph Object Gateway を再起動します。

```
cephuser@adm > ceph orch restart rgw
```

前のマスタゾーンが復旧したら、操作を元に戻します。

1. 復旧したゾーンで、現在のマスタゾーンから最新のレルムの設定をインポートします。

```
cephuser@adm > radosgw-admin realm pull --url=URL-TO-MASTER-ZONE-GATEWAY \
--access-key=ACCESS-KEY --secret=SECRET
```

2. 復旧したゾーンをマスタおよびデフォルトのゾーンにします。

```
cephuser@adm > radosgw-admin zone modify --rgw-zone=ZONE-NAME --master --default
```

3. ピリオドを更新して変更を有効にします。

```
cephuser@adm > radosgw-admin period update --commit
```

4. 復旧したゾーンでCeph Object Gatewayを再起動します。

```
cephuser@adm > ceph orch restart rgw@rgw
```

5. セカンダリゾーンを読み込み専用設定にする必要がある場合、セカンダリゾーンを更新します。

```
cephuser@adm > radosgw-admin zone modify --rgw-zone=ZONE-NAME --read-only
```

6. ピリオドを更新して変更を有効にします。

```
cephuser@adm > radosgw-admin period update --commit
```

7. セカンダリゾーンでCeph Object Gatewayを再起動します。

```
cephuser@adm > ceph orch restart@rgw
```

## 22 Ceph iSCSI Gateway

この章では、iSCSI Gatewayに関連する管理タスクに焦点を当てて説明します。展開手順については、『導入ガイド』、第5章「cephadmによる展開」、5.4.3.5項「iSCSI Gatewayの展開」を参照してください。

### 22.1 ceph-iscsi管理対象ターゲット

この章では、Linux、Microsoft Windows、またはVMwareが実行されているクライアントからceph-iscsi管理対象ターゲットに接続する方法について説明します。

#### 22.1.1 open-iscsiに接続

ceph-iscsiを利用するiSCSIターゲットにopen-iscsiで接続するのは、2段階のプロセスです。まず、イニシエータがゲートウェイホスト上で利用可能なiSCSIターゲットを検出し、ログインして、利用可能なLU (論理ユニット)をマップする必要があります。

どちらの手順でもopen-iscsiデーモンが実行されている必要があります。open-iscsiデーモンの起動方法はLinux配布パッケージによって異なります。

- SLES (SUSE Linux Enterprise Server)およびRHEL (Red Hat Enterprise Linux)のホストでは、**systemctl start iscsid** (または、**systemctl**が利用できない場合は**service iscsid start**)を実行します。
- DebianおよびUbuntuのホストでは、**systemctl start open-iscsi** (または**service open-iscsi start**)を実行します。

イニシエータホストでSUSE Linux Enterprise Serverが実行されている場合、iSCSIターゲットへの接続方法については、<https://documentation.suse.com/sles/15-SP2/html/SLES-all/cha-iscsi.html#sec-iscsi-initiator>を参照してください。

open-iscsiをサポートするその他のLinux配布パッケージでは、次に進んでceph-iscsiゲートウェイ上でターゲットを検出します(この例では、iscsi1.example.comをポータルアドレスとして使用します。マルチパスアクセスの場合は、iscsi2.example.comでこれらの手順を繰り返します)。

```
root # iscsiadm -m discovery -t sendtargets -p iscsi1.example.com
192.168.124.104:3260,1 iqn.2003-01.org.linux-iscsi.iscsi.SYSTEM-ARCH:testvol
```

続いて、ポータルにログインします。ログインが正常に完了した場合、ポータル上でRBDを利用する論理ユニットは、ただちにシステムSCSIバス上で利用可能になります。

```
root # iscsiadm -m node -p iscsi1.example.com --login
Logging in to [iface: default, target: iqn.2003-01.org.linux-iscsi.iscsi.SYSTEM-ARCH:testvol, portal: 192.168.124.104,3260] (multiple)
Login to [iface: default, target: iqn.2003-01.org.linux-iscsi.iscsi.SYSTEM-ARCH:testvol, portal: 192.168.124.104,3260] successful.
```

他のポータルIPアドレスまたはホストに対して、このプロセスを繰り返します。

システムに`lsscsi`ユーティリティがインストールされている場合は、そのユーティリティを使用して、システムで利用可能なSCSIデバイスを列挙します。

```
lsscsi
[8:0:0:0]    disk      SUSE      RBD          4.0    /dev/sde
[9:0:0:0]    disk      SUSE      RBD          4.0    /dev/sdf
```

マルチパス設定(接続されている2台のiSCSIデバイスが1台の同一のLUを表す)では、`multipath`ユーティリティでマルチパスデバイスの状態を調べることもできます。

```
root # multipath -ll
360014050cf9dcfcb2603933ac3298dca dm-9 SUSE,RBD
size=49G features='0' hwhandler='0' wp=rw
|-+- policy='service-time 0' prio=1 status=active
|  `-- 8:0:0:0 sde 8:64 active ready running
`--+- policy='service-time 0' prio=1 status=enabled
    `-- 9:0:0:0 sdf 8:80 active ready running
```

これで、このマルチパスデバイスをBlock Deviceと同じように使用できます。たとえば、デバイスをLinux LVM (論理ボリューム管理)の物理ボリュームとして使用したり、単にデバイス上にファイルシステムを作成したりできます。次の例は、新しく接続されたマルチパスiSCSIボリューム上にXFSファイルシステムを作成する方法を示しています。

```
root # mkfs -t xfs /dev/mapper/360014050cf9dcfcb2603933ac3298dca
log stripe unit (4194304 bytes) is too large (maximum is 256KiB)
log stripe unit adjusted to 32KiB
meta-data=/dev/mapper/360014050cf9dcfcb2603933ac3298dca isize=256    agcount=17,
  agsize=799744 blks
       =                          sectsz=512   attr=2, projid32bit=1
       =                          crc=0         finobt=0
data     =                          bsize=4096   blocks=12800000, imaxpct=25
       =                          sunit=1024    swidth=1024 blks
naming   =version 2                bsize=4096   ascii-ci=0 ftype=0
log      =internal log             bsize=4096   blocks=6256, version=2
       =                          sectsz=512    sunit=8 blks, lazy-count=1
realtime =none                     extsz=4096   blocks=0, rtextents=0
```

XFSは非クラスタ化ファイルシステムであるため、特定の時点で1つのiSCSIイニシエータノードにのみマウントできます。

特定のターゲットに関連付けられているiSCSI LUの使用を中止したい場合は、次のコマンドを実行します。

```
root # iscsiadm -m node -p iscsi1.example.com --logout
Logging out of session [sid: 18, iqn.2003-01.org.linux-iscsi.iscsi.SYSTEM-ARCH:testvol,
portal: 192.168.124.104,3260]
Logout of [sid: 18, target: iqn.2003-01.org.linux-iscsi.iscsi.SYSTEM-ARCH:testvol, portal:
192.168.124.104,3260] successful.
```

ディスカバリおよびログインの場合と同様に、ポータルのすべてのIPアドレスまたはホスト名に対してログアウト手順を繰り返す必要があります。

### 22.1.1.1 マルチパスの設定

マルチパス設定はクライアントまたはイニシエータ上で管理され、`ceph-iscsi`設定とは無関係です。ブロックストレージを使用する前にストラテジーを選択します。`/etc/multipath.conf`を編集した後、次のコマンドを使用して`multipathd`を再起動します。

```
root # systemctl restart multipathd
```

フレンドリ名を使用するアクティブ-パッシブ設定では、次の記述を設定ファイルに追加します。

```
defaults {
    user_friendly_names yes
}
```

追加先のファイルは`/etc/multipath.conf`です。ターゲットに正常に接続したら、次のコマンドを実行します。

```
root # multipath -ll
mpathd (36001405dbb561b2b5e439f0aed2f8e1e) dm-0 SUSE,RBD
size=2.0G features='0' hwhandler='0' wp=rw
|+- policy='service-time 0' prio=1 status=active
| `-- 2:0:0:3 sdl 8:176 active ready running
|+- policy='service-time 0' prio=1 status=enabled
| `-- 3:0:0:3 sdj 8:144 active ready running
`+- policy='service-time 0' prio=1 status=enabled
  `-- 4:0:0:3 sdk 8:160 active ready running
```

各リンクの状態に注意してください。アクティブ-アクティブ設定の場合は、次の記述を設定ファイルに追加します。

```
defaults {
    user_friendly_names yes
}

devices {
    device {
        vendor "(LIO-ORG|SUSE)"
        product "RBD"
        path_grouping_policy "multibus"
        path_checker "tur"
        features "0"
        hardware_handler "1 alua"
        prio "alua"
        failback "immediate"
        rr_weight "uniform"
        no_path_retry 12
        rr_min_io 100
    }
}
```

追加先のファイルは/etc/multipath.confです。multipathdを再起動して、次のコマンドを実行します。

```
root # multipath -ll
mpathd (36001405dbb561b2b5e439f0aed2f8e1e) dm-3 SUSE,RBD
size=2.0G features='1 queue_if_no_path' hwhandler='1 alua' wp=rw
`-+- policy='service-time 0' prio=50 status=active
   |- 4:0:0:3 sdj 8:144 active ready running
   |- 3:0:0:3 sdk 8:160 active ready running
   `-- 2:0:0:3 sdl 8:176 active ready running
```

## 22.1.2 Microsoft Windows (Microsoft iSCSIイニシエータ)に接続

Windows 2012サーバからSUSE Enterprise StorageのiSCSIターゲットに接続するには、次の手順に従います。

1. Windowsサーバ マネージャーを開きます。ダッシュボードから、ツール > iSCSI イニシエータを選択します。iSCSI イニシエータのプロパティダイアログが表示されます。探索タブを選択します。

ターゲット

検出

お気に入りのターゲット

ボリュームとデバイス

RADIUS

設定

ターゲットポータル

システムは次のポータルでターゲットを検索します。

更新

アドレス	ポート	アダプタ	IPアドレス

ターゲットポータルを追加するには、[ポータルの検出]をクリックします。

ポータルの検出...

ターゲットポータルを削除するには、上のアドレスを選択して、[削除]をクリックします。

削除

iSNSサーバ

システムは次のiSNSサーバに登録されています。

更新

名前

iSNSサーバを追加するには、[サーバの追加]を選択します。

サーバの追加...

iSNSサーバを削除するには、上のサーバを選択して、[削除]をクリックします。

削除

OK

キャンセル

適用

図 22.1: iSCSIイニシエータのプロパティ

- ターゲット ポータルの探索ダイアログで、ターゲットフィールドにターゲットのホスト名またはIPアドレスを入力して、OKをクリックします。

追加するポータルのIPアドレスまたはDNS名およびポート番号を入力します。

ターゲットポータルの検出のデフォルト設定を変更するには、[詳細] ボタンをクリックします。

IPアドレスまたはDNS名:                      ポート: (デフォルトは3260です。)

192.168.124.104                      3260

詳細...                      OK                      キャンセル

図 22.2: ターゲットポータルの探索

3. 他のすべてのゲートウェイホストの名前またはIPアドレスに対して、このプロセスを繰り返します。完了したら、ターゲット ポータルリストを確認します。

ターゲット

検出

お気に入りのターゲット

ボリュームとデバイス

RADIUS

設定

ターゲットポータル

システムは次のポータルでターゲットを検索します。

更新

アドレス	ポート	アダプタ	IPアドレス
192.168.124.104	3260	デフォルト	デフォルト
192.168.124.105	3260	デフォルト	デフォルト

ターゲットポータルを追加するには、[ポータルの検出]をクリックします。

ポータルの検出...

ターゲットポータルを削除するには、上のアドレスを選択して、[削除]をクリックします。

削除

iSNSサーバ

システムは次のiSNSサーバに登録されています。

更新

名前

iSNSサーバを追加するには、[サーバの追加]を選択します。

サーバの追加...

iSNSサーバを削除するには、上のサーバを選択して、[削除]をクリックします。

削除

OK

キャンセル

適用

図 22.3: ターゲットポータル

- 次に、ターゲットタブに切り替えて、検出されたターゲットを確認します。

305

Microsoft Windows (Microsoft iSCSI イニシエータ) に接続 | SES 7



図 22.4: ターゲット

- ターゲットタブで接続をクリックします。ターゲットへの接続ダイアログが表示されます。複数パスを有効にするチェックボックスをオンにしてMPIO (マルチパスI/O)を有効にして、OKをクリックします。
- ターゲットへの接続ダイアログが閉じたら、プロパティを選択して、ターゲットのプロパティを確認します。

セッション

ポータルグループ

更新

識別子

☒ fffff00103669020-400001370000000f  
☒ fffff00103669020-40000137000000010

セッションを追加するには、[セッションの追加] をクリックします。

セッションの追加

1つ以上のセッションを接続解除するには、各セッションを選択して、[接続解除] をクリックします。

接続解除

セッションに関連付けられたデバイスを表示するには、セッションを選択して、[デバイス] をクリックします。

デバイス...

セッション情報

ターゲットポータルグループタグ:

1

ステータス:

接続

接続数:

1

最大許容接続数:

1

認証:

指定なし

ヘッダーダイジェスト:

指定なし

データダイジェスト:

指定なし

複数接続セッション(MCS)の設定

セッションに接続を追加する、または選択済みセッションのMCSポリシーを設定するには、[MCS] をクリックします。

MCS...

OK

キャンセル

図 22.5: iSCSIターゲットのプロパティ

7. デバイスを選択し、MPIOをクリックしてマルチパスI/Oの設定を確認します。

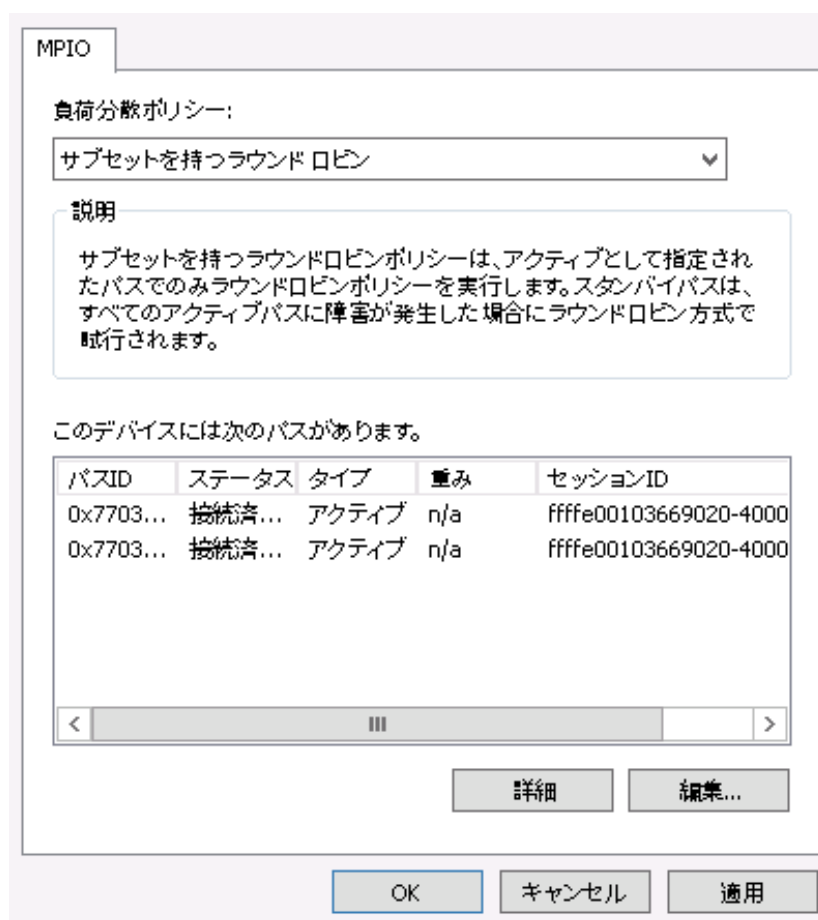


図 22.6: デバイスの詳細

デフォルトの負荷分散ポリシーはRound Robin With Subset (サブセット付きラウンドロビン)です。純粋なフェールオーバー設定が必要な場合は、Fail Over Only (フェールオーバーのみ)に変更します。

これでiSCSIイニシエータの設定は終了です。iSCSIボリュームを他のSCSIデバイスと同じように利用できるようになり、初期化してボリュームやドライブとして使用できます。OKをクリックしてiSCSIイニシエータのプロパティダイアログを閉じて、サーバマネージャーダッシュボードからファイルサービスと記憶域サービスの役割に進みます。

新しく接続されたボリュームを確認します。これはiSCSIバス上で「SUSE RBD SCSI Multi-Path Drive (SUSE RBD SCSIマルチパスデバイス)」として識別されており、初期状態では、状態に「オフライン」、パーティションテーブルタイプに「不明」のマークが付いています。新しいボリュームがすぐに表示されない場合は、タスクドロップダウンボックスから記憶域の再スキャンを選択して、iSCSIバスを再スキャンします。

1. iSCSIボリュームを右クリックして、コンテキストメニューからボリュームの新規作成を選択します。新しいボリューム ウィザードが表示されます。次へをクリックして、新しく接続されたiSCSIボリュームを強調表示し、次へをクリックして開始します。

### サーバとディスクの選択

開始する前に

- サーバとディスク
- サイズ
- ドライブ文字またはフォルダ
- ファイルシステム設定
- 確認
- 結果

サーバ:

プロビジョニング先	ステータス	クラスタの役割	宛先
WIN-U3AILLIMUEE	オンライン	非クラスタ化	ローカル

更新 再スキャン

ディスク:

ディスク	仮想ディスク	容量	空き容量	サブシステム
ディスク0		234GB	233GB	
ディスク1		234GB	234GB	
ディスク3		48.8GB	48.8GB	

< 前へ 次へ > 作成 キャンセル

図 22.7: 新しいボリュームウィザード

2. 初期状態では、デバイスは空でパーティションテーブルを含みません。プロンプトが表示されたら、ボリュームがGPTパーティションテーブルで初期化されることを示すダイアログを確認します。

**i** 選択したディスクがオンラインになり、GPTディスクとして初期化されます。続行するには、[OK]をクリックします。別のディスクを選択するか、新しい仮想ディスクを作成するには、[キャンセル]をクリックします。

OK キャンセル

図 22.8: オフラインディスクのプロンプト

3. ボリュームサイズを選択します。通常は、デバイスの全容量を使用します。続いて、新しく作成されたボリュームを利用できるドライブ文字またはディレクトリ名を割り当てます。次に、新しいボリューム上に作成するファイルシステムを選択します。最後に、選択内容を確認して作成をクリックし、ボリュームの作成を完了します。

## 選択内容の確認

開始する前に

サーバとディスク

サイズ

ドライブ文字またはフォルダ

ファイルシステム設定

**確認**

結果

以下の設定が正しいことを確認して、[作成]をクリックします。

ボリュームの場所

サーバ: WIN-U3AILLIMUEE

ディスク: ディスク3

空き容量: 48.8GB

ボリュームのプロパティ

ボリュームサイズ: 48.8GB

ドライブ文字またはフォルダ: D:\

ボリュームラベル: 新しいボリューム

ファイルシステム設定

ファイルシステム: NTFS

短いファイル名の作成: 無効

割り当てユニットサイズ: デフォルト

< 前へ

次へ >

作成

キャンセル

図 22.9: ボリュームの選択内容の確認

プロセスが完了したら、結果を確認し、閉じるをクリックしてドライブの初期化を完了します。初期化が完了すると、このボリューム(およびそのNTFSファイルシステム)は、新しく初期化されたローカルドライブと同じように利用可能になります。

### 22.1.3 VMwareの接続

1. `ceph-iscsi`で管理されているiSCSIボリュームに接続するには、設定済みのiSCSIソフトウェアアダプタが必要です。現在のvSphere設定でこのアダプタが利用できない場合は、構成 > ストレージ アダプタ > 追加 > iSCSI Software initiator (iSCSIソフトウェアイニシエータ)の順に選択して作成します。
2. アダプタが利用可能になったら、アダプタを右クリックしてコンテキストメニューからプロパティを選択し、アダプタのプロパティを選択します。

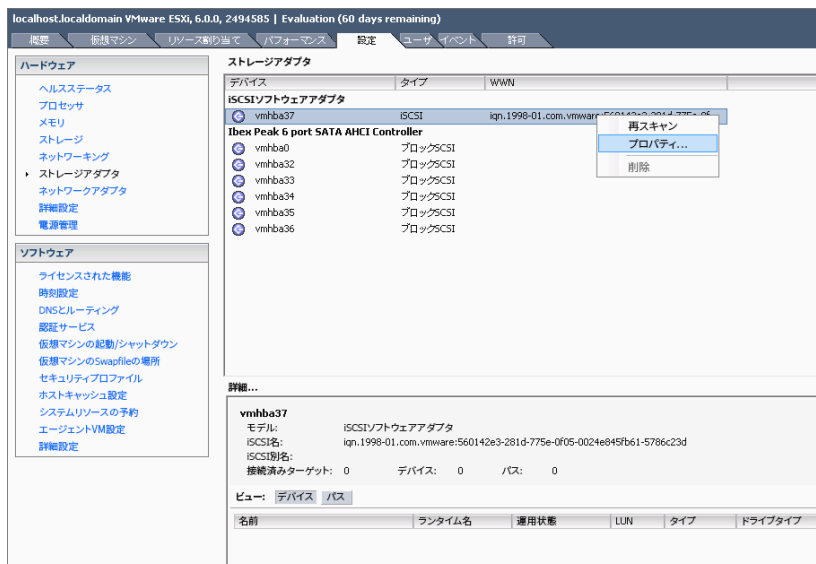


図 22.10: iSCSIイニシエータのプロパティ

3. iSCSI Software Initiator (iSCSIソフトウェアイニシエータ)ダイアログで、構成ボタンをクリックします。続いて、動的検出タブに移動して、追加を選択します。
4. `ceph-iscsi` iSCSI GatewayのIPアドレスまたはホスト名を入力します。複数のiSCSI Gatewayをフェールオーバー設定で実行する場合は、運用するゲートウェイの数だけこの手順を繰り返します。

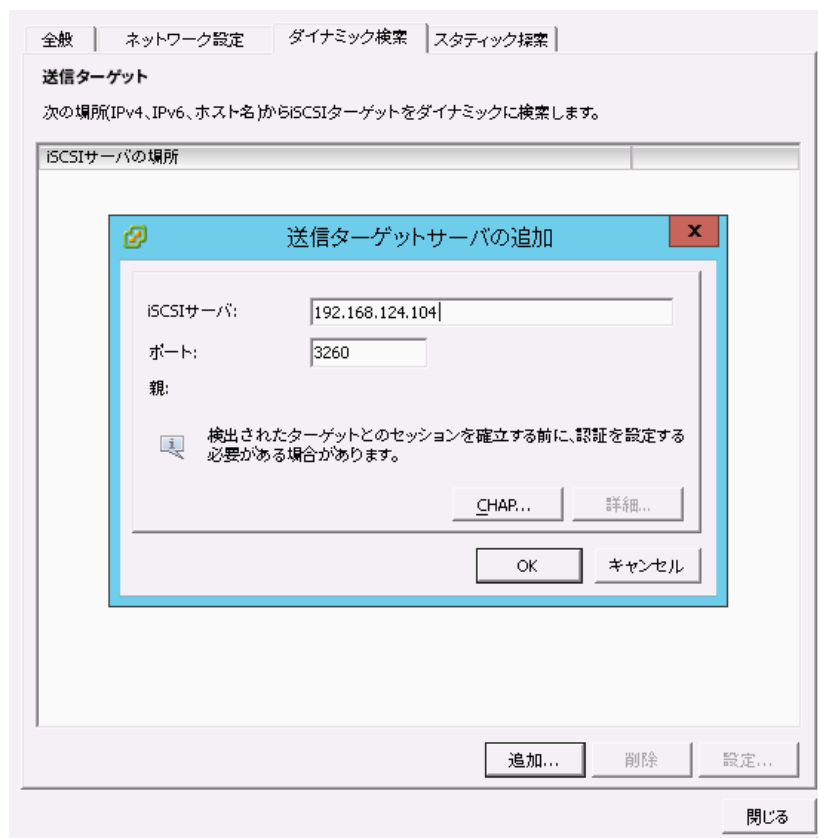


図 22.11: ターゲットサーバの追加

すべてのiSCSI Gatewayを入力したら、ダイアログでOKをクリックして、iSCSIアダプタの再スキャンを開始します。

5. 再スキャンが完了すると、新しいiSCSIデバイスが詳細ペインのストレージ アダプタリストの下に表示されます。マルチパスデバイスの場合は、アダプタを右クリックして、コンテキストメニューからパスの管理を選択します。



図 22.12: マルチパスデバイスの管理

これで、ステータスにすべてのパスが緑色のマークで表示されます。パスの1つに有効(I/O)のマークが付いていて、他のすべてには単にアクティブのマークが付いている必要があります。

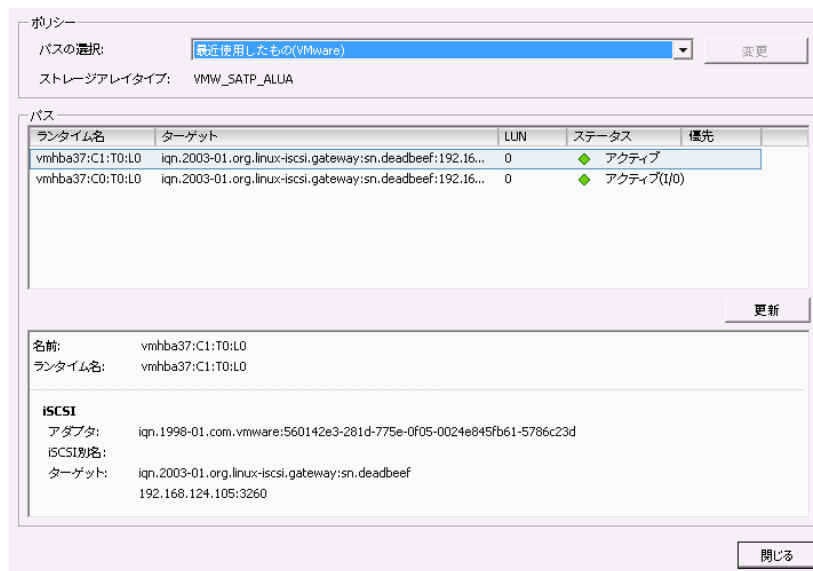


図 22.13: マルチパスのパスの一覧

6. ストレージ アダプタからステータスというラベルの項目に切り替えることができます。このペインの右上隅にあるストレージの追加...を選択して、Add Storage (ストレージの追加)ダイアログを表示します。続いて、ディスク/LUNを選択して、次へをクリックします。新しく追加されたiSCSIデバイスがディスクまたはLUNの選択リストに表示されます。デバイスを選択し、次へをクリックして続行します。

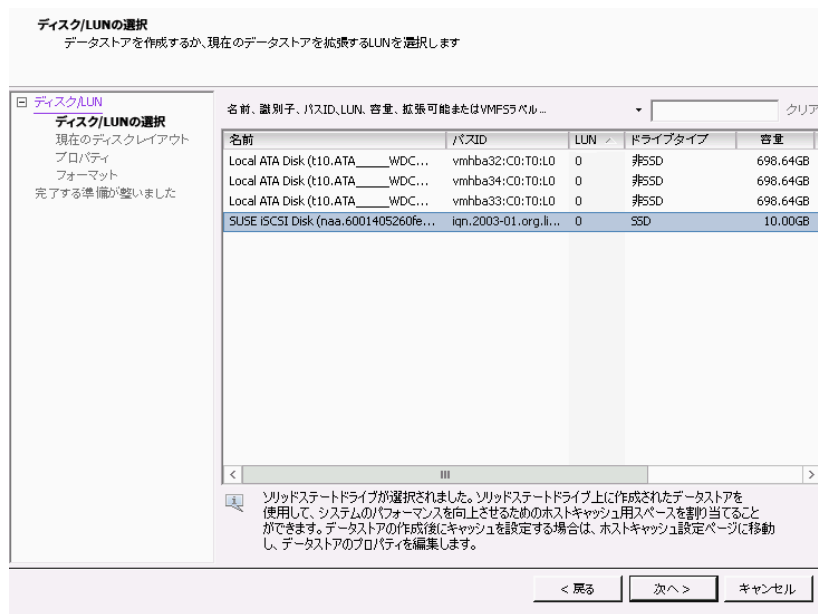


図 22.14: ストレージの追加ダイアログ

次へをクリックして、デフォルトのディスクレイアウトをそのまま使用します。

7. プロパティペインで、新しいデータストアに名前を割り当てて、次へをクリックします。ボリュームの全領域をこのデータストアに使用する場合はデフォルト設定をそのまま使用し、データストアの領域を減らす場合はカスタム領域設定を選択します。

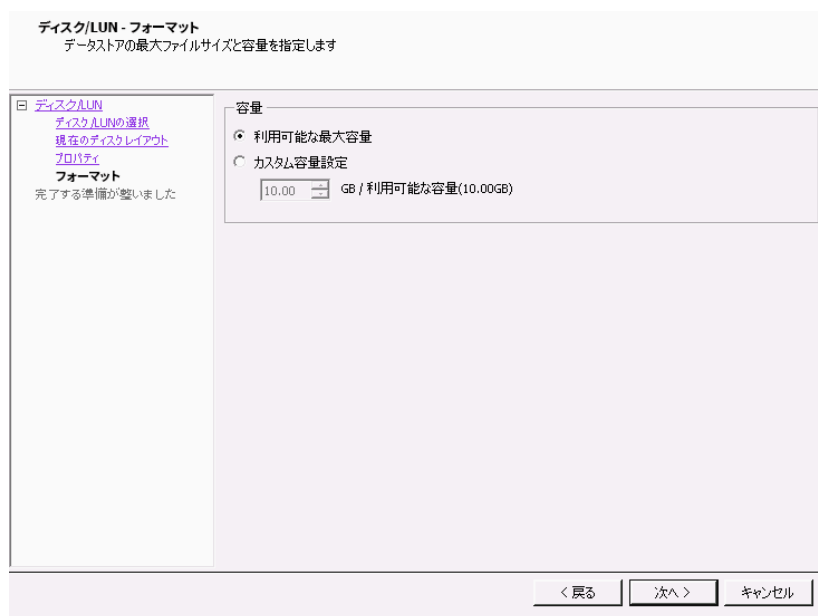


図 22.15: カスタム領域設定

終了をクリックしてデータストアの作成を完了します。

新しいデータストアがデータストアのリストに表示され、そのデータストアを選択して詳細を取得できます。これで、vSphereの他のデータストアと同じように、ceph-iscsiを利用するiSCSIボリュームを使用できるようになります。

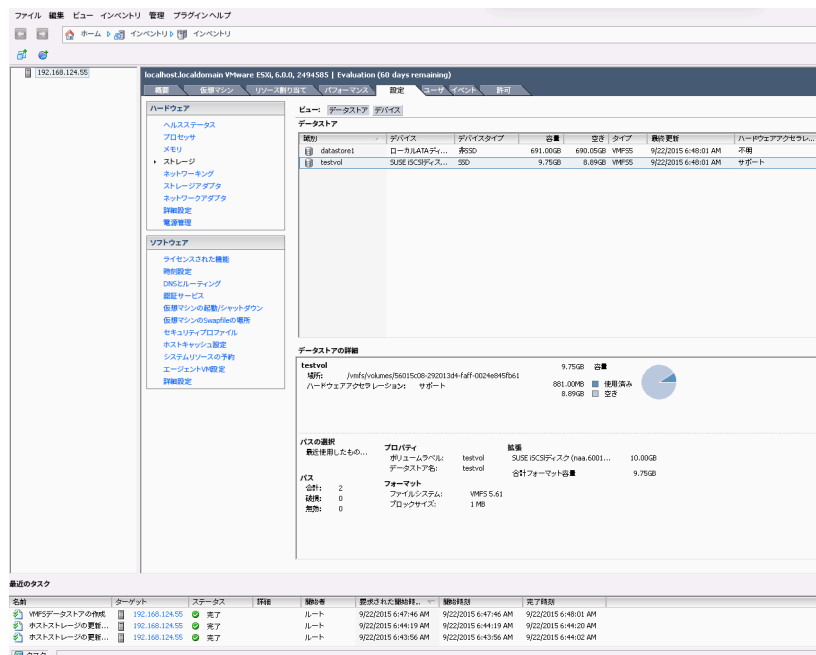


図 22.16: iSCSIデータストアの概要

## 22.2 結論

ceph-iscsiはSUSE Enterprise Storage 7において鍵となるコンポーネントで、iSCSIプロトコルに対応した任意のサーバやクライアントから分散型の高可用性ブロックストレージへのアクセスを可能にします。1つ以上のiSCSI Gatewayホストでceph-iscsiを使用することにより、Ceph RBDイメージはiSCSIターゲットに関連付けられたLU (論理ユニット)として利用可能になります。オプションで、負荷分散された可用性が高い方法でアクセスすることもできます。

ceph-iscsiのすべての設定はCeph RADOS Object Storeに保存されるので、ceph-iscsiゲートウェイホストは本質的に永続状態を持ちません。したがって、自由に置換、増強、または縮小できます。その結果、SUSE Enterprise Storage 7により、SUSEのお客様は、コモディティハードウェアと完全なオープンソースプラットフォーム上で、真に分散化され、高い可用性、災害耐性、自己修復機能を併せ持つエンタープライズストレージ技術を運用できます。

## 23 クラスタファイルシステム

この章では、通常はクラスタの設定とCephFSのエクスポート後に実行する管理タスクについて説明します。CephFSの設定の詳細については、『導入ガイド』、第5章「cephadmによる展開」、5.4.3.3項「メタデータサーバの展開」を参照してください。

### 23.1 CephFSのマウント

ファイルシステムが作成されてMDSがアクティブになったら、クライアントホストからファイルシステムをマウントできます。

#### 23.1.1 クライアントの準備

クライアントホストがSUSE Linux Enterprise 12 SP2以降を実行している場合、システムはCephFSをそのまますぐにマウントできます。

クライアントホストがSUSE Linux Enterprise 12 SP1を実行している場合は、CephFSをマウントする前にすべての最新パッチを適用する必要があります。

いずれの場合も、CephFSをマウントするのに必要なものはすべてSUSE Linux Enterpriseに付属しています。SUSE Enterprise Storage 7製品は必要ありません。

完全な`mount`構文をサポートするには、CephFSのマウントを試みる前に、`ceph-common` パッケージ(SUSE Linux Enterpriseに付属)をインストールする必要があります。

#### ！ 重要

パッケージ `ceph-common` がない場合(つまり、`mount.ceph`ヘルパーがない場合)、モニターのホスト名ではなく、IPアドレスが必要です。これは、カーネルクライアントがネームレゾリューションを実行できないためです。

基本的な構文は、次の通りです。

```
root # mount -t ceph MON1_IP[:PORT],MON2_IP[:PORT],...:CEPHFS_MOUNT_TARGET \
MOUNT_POINT -o name=CEPHX_USER_NAME,secret=SECRET_STRING
```

## 23.1.2 シークレットファイルの作成

Cephクラスタは、デフォルトで認証がオンの状態で動作します。秘密鍵(キーリングそのものではない)を保存するファイルを作成する必要があります。特定のユーザの秘密鍵を入手してファイルを作成するには、次の操作を行います。

### 手順 23.1: 秘密鍵の作成

1. キーリングファイル内の特定のユーザの鍵を表示します。

```
cephuser@adm > cat /etc/ceph/ceph.client.admin.keyring
```

2. マウントしたCephFS (Ceph File System)を使用するユーザの鍵をコピーします。通常、鍵は次のような形式です。

```
AQCj2YpRiAe6CxAA7/ETt7Hcl9IyxyYciVs47w==
```

3. ファイル名の部分にユーザ名を使用してファイルを作成します。たとえば、ユーザ**admin**の場合は、/etc/ceph/admin.secretのようになります。
4. 前の手順で作成したファイルに鍵の値を貼り付けます。
5. ファイルに適切なアクセス権を設定します。このユーザは、ファイルを読み込める唯一のユーザである必要があります。ほかのユーザは一切アクセス権を持つことはできません。

## 23.1.3 CephFSのマウント

**mount** コマンドでCephFSをマウントできます。Monitorのホスト名またはIPアドレスを指定する必要があります。SUSE Enterprise Storageでは**cephx**認証がデフォルトで有効になっているため、ユーザ名とその関連シークレットも指定する必要があります。

```
root # mount -t ceph ceph_mon1:6789:/ /mnt/cephfs \  
-o name=admin,secret=AQATSKdNGBnwLhAAAnNDKnH65FmVKpXZJVasUeQ==
```

以前のコマンドはシェルの履歴に残るため、ファイルからシークレットを読み込むアプローチの方が安全です。

```
root # mount -t ceph ceph_mon1:6789:/ /mnt/cephfs \  
-o name=admin,secretfile=/etc/ceph/admin.secret
```

シークレットファイルには実際のキーリングシークレットだけが含まれる必要があることに注意してください。この例では、ファイルに含まれるのは次の行だけです。



## ヒント: 複数のMonitorの指定

マウント時に特定のMonitorがダウンしている事態に備え、**mount**コマンドラインで複数のMonitorをコンマで区切って指定することをお勧めします。各Monitorのアドレスは`host[:port]`という形式です。ポートを指定しない場合は、デフォルトで6789が使用されます。

ローカルホストでマウントポイントを作成します。

```
root # mkdir /mnt/cephfs
```

CephFSをマウントします。

```
root # mount -t ceph ceph_mon1:6789:/ /mnt/cephfs \
-o name=admin,secretfile=/etc/ceph/admin.secret
```

ファイルシステムのサブセットをマウントする場合は、サブディレクトリ subdir を指定できます。

```
root # mount -t ceph ceph_mon1:6789:/subdir /mnt/cephfs \
-o name=admin,secretfile=/etc/ceph/admin.secret
```

**mount** コマンドで複数のMonitorホストを指定できます。

```
root # mount -t ceph ceph_mon1,ceph_mon2,ceph_mon3:6789:/ /mnt/cephfs \
-o name=admin,secretfile=/etc/ceph/admin.secret
```



## 重要: ルートディレクトリに対する読み込みアクセス

パス制約付きのクライアントを使用する場合は、MDSのケーパビリティにルートディレクトリに対する読み込みアクセスを含める必要があります。たとえば、キーリングは次のようになります。

```
client.bar
key: supersecretkey
caps: [mds] allow rw path=/barjail, allow r path=/
caps: [mon] allow r
caps: [osd] allow rwx
```

`allow r path=`の部分は、パス制約付きのクライアントは、ルートボリュームを表示できても書き込みはできないことを意味します。これは、完全な分離が要件である使用事例で問題になることがあります。

## 23.2 CephFSのアンマウント

CephFSをアンマウントするには、`umount`コマンドを使用します。

```
root # umount /mnt/cephfs
```

## 23.3 /etc/fstabでのCephFSのマウント

クライアントの起動時にCephFSを自動的にマウントするには、対応する行をファイルシステムテーブル/`/etc/fstab`に挿入します。

```
mon1:6790,mon2:/subdir /mnt/cephfs ceph name=admin,secretfile=/etc/ceph/  
secret.key,noatime,_netdev 0 2
```

## 23.4 複数のアクティブMDSデーモン(アクティブ-アクティブMDS)

CephFSは、デフォルトでは単一のアクティブMDSデーモン用に設定されています。大規模システム用にメタデータのパフォーマンスを拡張する場合、複数のアクティブMDSデーモンを有効にできます。これにより、各デーモンがお互いにメタデータワークロードを共有します。

### 23.4.1 アクティブ-アクティブMDSの使用

デフォルトの単一のMDSではメタデータのパフォーマンスがボトルネックになる場合、複数のアクティブMDSデーモンの使用を検討します。

デーモンを追加しても、すべてのワークロードタイプのパフォーマンスが向上するわけではありません。たとえば、単一のクライアント上で動作している単一のアプリケーションの場合、そのアプリケーションが大量のメタデータ操作を並列で実行していない限り、MDSデーモンの数を増やしてもメリットはありません。

一般的に大量のアクティブMDSデーモンのメリットを受けられるワークロードは、クライアントが複数あり、多数の別個のディレクトリを操作する可能性が高いワークロードです。

## 23.4.2 MDSのアクティブクラスタサイズの増加

各CephFSファイルシステムには、作成するランクの数を制御する`max_mds`設定があります。ファイルシステム内の実際のランク数は、新しいランクを引き受けるスペアデーモンが利用可能な場合にのみ増やされます。たとえば、実行中のMDSデーモンが1つだけで、`max_mds`が2に設定されている場合、2番目のランクは作成されません。

次の例では、`max_mds`オプションを2に設定して、デフォルトのランクとは別の新しいランクを作成します。変更を確認するには、`max_mds`の設定前と設定後に`ceph status`を実行し、`fsmap`が含まれる行を確認します。

```
cephuser@adm > ceph status
[...]
```

services:

```
[...]
mds: cephfs-1/1/1 up {0=node2=up:active}, 1 up:standby
[...]
```

cephuser@adm > ceph fs set cephfs max\_mds 2

cephuser@adm > ceph status

```
[...]
```

services:

```
[...]
mds: cephfs-2/2/2 up {0=node2=up:active,1=node1=up:active}
[...]
```

新しく作成されたランク(1)は、「creating (作成中)」状態を経由して「active (アクティブ)」状態になります。

### ！ 重要: スタンバイデーモン

複数のアクティブMDSデーモンを使用している場合、高可用性システムには、アクティブデーモンを実行するサーバに障害が発生した場合に処理を引き継ぐスタンバイデーモンも必要です。

そのため、高可用性システムの`max_mds`の実用的な最大数は、システムのMDSサーバの合計数から1を引いた数になります。複数のサーバ障害時に可用性を維持するには、切り抜ける必要があるサーバ障害の数に一致するようにシステムのスタンバイデーモンの数を増やします。

## 23.4.3 ランク数の減少

最初に、すべてのランク(削除するランクを含む)がアクティブになっている必要があります。つまり、少なくとも`max_mds` MDSデーモンが利用可能である必要があります。

最初に、`max_mds`をより低い数字に設定します。たとえば、単一のアクティブMDSに戻します。

```
cephuser@adm > ceph status
[...]
services:
  [...]
  mds: cephfs-2/2/2 up {0=node2=up:active,1=node1=up:active}
  [...]
cephuser@adm > ceph fs set cephfs max_mds 1
cephuser@adm > ceph status
[...]
services:
  [...]
  mds: cephfs-1/1/1 up {0=node2=up:active}, 1 up:standby
  [...]
```

## 23.4.4 ランクへのディレクトリツリーの手動固定

複数のアクティブメタデータサーバ設定では、バランサが動作し、メタデータの負荷をクラスターに均等に分散します。これは通常、ほとんどのユーザにとって十分有効に機能しますが、メタデータを特定のランクに明示的にマッピングして動的バランサを無効にした方が良い場合もあります。これにより、管理者やユーザは、アプリケーションの負荷を均等に分散したり、ユーザのメタデータ要求によるクラスター全体への影響を抑えたりできます。

このために提供されているメカニズムを「エクスポートピン」と呼びます。これはディレクトリの拡張属性です。この拡張属性の名前は`ceph.dir.pin`です。標準のコマンドを使用して、この属性を設定できます。

```
root # setfattr -n ceph.dir.pin -v 2 /path/to/dir
```

拡張属性の値(-v)は、ディレクトリサブツリーの割り当て先となるランクです。デフォルト値-1は、ディレクトリが固定されないことを示します。

ディレクトリのエクスポートピンは、設定されているエクスポートピンを持つ最も近い親から継承されます。したがって、ディレクトリにエクスポートピンを設定すると、そのすべての子に影響します。ただし、子ディレクトリのエクスポートピンを設定して親のピンを上書きできます。例:

```
root # mkdir -p a/b                                     # "a" and "a/b" start with no export pin set.
setfattr -n ceph.dir.pin -v 1 a/                         # "a" and "b" are now pinned to rank 1.
setfattr -n ceph.dir.pin -v 0 a/b                       # "a/b" is now pinned to rank 0
                                                         # and "a/" and the rest of its children
                                                         # are still pinned to rank 1.
```

## 23.5 フェールオーバーの管理

MDSデーモンがMonitorとの通信を停止した場合、そのMonitorは`mds_beacon_grace`の秒数(デフォルトは15秒)待機してから、デーモンを「遅延」としてマークします。MDSデーモンのフェールオーバー中に処理を引き継ぐ「スタンバイ」デーモンを1つ以上設定できます。

### 23.5.1 スタンバイ再生の設定

各CephFSファイルシステムはスタンバイ再生デーモンを追加するように設定することもできます。こうしたスタンバイデーモンはアクティブMDSのメタデータジャーナルを追跡して、アクティブMDSが利用不能になるようなイベントが発生した場合のフェールオーバー時間を短縮します。各アクティブMDSに設定できる、追跡用のスタンバイ再生デーモンは一つだけです。

ファイルシステムにスタンバイ再生を設定するには、次のコマンドを使用します。

```
cephuser@adm > ceph fs set FS-NAME allow_standby_replay B00L
```

設定された場合、モニターは使用可能なスタンバイデーモンをファイルシステムのアクティブMDSを追跡するように割り当てます。

ひとたびMDSがスタンバイ再生状態になると、そのMDSは追跡しているランクのスタンバイとしてのみ使用されます。別のランクが失敗し、他に利用可能なスタンバイがない場合でも、このスタンバイ再生デーモンは代わりとして使用されません。そのため、スタンバイ再生機能を使用する場合は、すべてのアクティブMDSにスタンバイ再生デーモンを設定することをお勧めします。

## 23.6 CephFSのクォータの設定

Cephファイルシステムの任意のサブディレクトリにクォータを設定できます。クォータは、ディレクトリ階層の指定したポイントの下層に保存される「バイト」または「ファイル」の数を制限します。

### 23.6.1 CephFSのクォータの制限

CephFSでのクォータの使用には、次の制限があります。

## クォータは協調的で非競合

Cephクォータは、ファイルシステムをマウントしているクライアントに依存し、制限に達すると書き込みを停止します。サーバ側では、悪意のあるクライアントが必要なだけデータを書き込むのを防止することはできません。クライアントが完全に信頼されていない環境では、ファイルシステムがいっぱいになるのを防ぐため、クォータを使用しないでください。

## クォータは正確ではない

ファイルシステムへの書き込み中のプロセスは、クォータ制限に達した直後に停止されます。そのため必然的に、設定された制限を超える量のデータを書き込むことができます。クライアントのライタは、設定された制限を超えてから1/10秒以内に停止されません。

## バージョン4.17からクォータはカーネルクライアントに実装される

クォータは、ユーザスペースクライアント(libcephfs、ceph-fuse)によってサポートされます。Linuxカーネルクライアント4.17以降は、SUSE Enterprise Storage 7クラスタ上のCephFSクォータをサポートします。カーネルクライアントが最新バージョンであっても、クォータ拡張属性を設定できても、古いクラスタ上のクォータは処理できません。SLE12-SP3以降のカーネルは、クォータの処理に必要なバックポートをすでに備えています。

## パスベースのマウント制限とともに使用する場合はクォータを慎重に設定する

クライアントは、クォータを適用するには、クォータが設定されているディレクトリiノードにアクセスできる必要があります。クライアントがMDSの機能に基づいて特定のパス(たとえば、/home/user)へのアクセスを制限されている場合に、そのクライアントがアクセスできない祖先ディレクトリ(/home)にクォータが設定されているときは、クライアントはクォータを適用しません。パスベースのアクセス制限を使用する場合、クライアントがアクセスできるディレクトリにクォータを設定することを忘れないでください(たとえば、/home/userや/home/user/quota\_dir)。

## 23.6.2 CephFSのクォータの設定

仮想拡張属性を使用して、CephFSクォータを設定できます。

ceph.quota.max\_files

「ファイル」制限を設定します。

ceph.quota.max\_bytes

「バイト」制限を設定します。

これらの属性がディレクトリiノード上に存在する場合、そこにクォータが設定されています。存在しない場合は、そのディレクトリにクォータは設定されていません(ただし、親ディレクトリに設定されている場合があります)。

100MBのクォータを設定するには、次のコマンドを実行します。

```
cephuser@mds > setfattr -n ceph.quota.max_bytes -v 100000000 /SOME/DIRECTORY
```

10,000ファイルのクォータを設定するには、次のコマンドを実行します。

```
cephuser@mds > setfattr -n ceph.quota.max_files -v 10000 /SOME/DIRECTORY
```

クォータ設定を表示するには、次のコマンドを実行します。

```
cephuser@mds > getfattr -n ceph.quota.max_bytes /SOME/DIRECTORY
```

```
cephuser@mds > getfattr -n ceph.quota.max_files /SOME/DIRECTORY
```



### 注記: クォータが設定されない

拡張属性の値が「0」の場合、クォータは設定されません。

クォータを削除するには、次のコマンドを実行します。

```
cephuser@mds > setfattr -n ceph.quota.max_bytes -v 0 /SOME/DIRECTORY
cephuser@mds > setfattr -n ceph.quota.max_files -v 0 /SOME/DIRECTORY
```

## 23.7 CephFSスナップショットの管理

CephFSスナップショットは、スナップショットを作成した時点でのファイルシステムの読み込み専用ビューを作成します。スナップショットは任意のディレクトリに作成できます。スナップショットでは、ファイルシステムの指定ディレクトリの下層にあるすべてのデータが対象になります。スナップショットの作成後、バッファされたデータはさまざまなクライアントから非同期にフラッシュされます。その結果、スナップショットの作成は非常に高速です。



### 重要: 複数のファイルシステム

複数のCephFSファイルシステムが(ネームスペースを介して)1つのプールを共有している場合、それらのスナップショットは競合し、1つのスナップショットを削除すると同じプールを共有している他のスナップショットのファイルデータがなくなります。

## 23.7.1 スナップショットの作成

CephFSスナップショット機能は、新しいファイルシステムではデフォルトで有効になっています。既存のファイルシステムでこの機能を有効にするには、次のコマンドを実行します。

```
cephuser@adm > ceph fs set CEPHFS_NAME allow_new_snaps true
```

スナップショットを有効にすると、CephFSのすべてのディレクトリに特別な `.snap` サブディレクトリが作成されます。



### 注記

これは「仮想」サブディレクトリです。このサブディレクトリは親ディレクトリのディレクトリ一覧には表示されませんが、`.snap` という名前はファイル名やディレクトリ名として使用できません。`.snap` ディレクトリにアクセスするには、明示的にアクセスする必要があります。たとえば、次のようなコマンドを使用します。

```
tux > ls -la /CEPHFS_MOUNT/.snap/
```



### 重要: カーネルクライアントの制限

CephFSカーネルクライアントには、1つのファイルシステム内で400を超えるスナップショットを処理できないという制限があります。スナップショットの数は、使用しているクライアントに関係なく、常にこの制限を下回るようにする必要があります。SLE12-SP3などの古いCephFSクライアントを使用する場合は、スナップショットが400を超えるとクライアントがクラッシュするため、操作に害を及ぼすことに注意してください。



### ヒント: カスタムスナップショットサブディレクトリ名

`client snapdir` 設定により、スナップショットサブディレクトリに異なる名前を設定できます。

スナップショットを作成するには、`.snap` ディレクトリに、カスタム名を持つサブディレクトリを作成します。たとえば、`/CEPHFS_MOUNT/2/3/` ディレクトリのスナップショットを作成するには、次のコマンドを実行します。

```
tux > mkdir /CEPHFS_MOUNT/2/3/.snap/CUSTOM_SNAPSHOT_NAME
```

## 23.7.2 スナップショットの削除

スナップショットを削除するには、.snapディレクトリ内にあるそのサブディレクトリを削除します。

```
tux > rmdir /CEPHFS_MOUNT/2/3/.snap/CUSTOM_SNAPSHOT_NAME
```

## 24 Sambaを介したCephデータのエクスポート

この章では、Cephクラスタに保存されたデータをSamba/CIFS共有を介してエクスポートし、Windows\*クライアントマシンからデータに簡単にアクセスできるようにする方法について説明します。また、Ceph Sambaゲートウェイを設定してWindows\*ドメインのActive Directoryに参加し、ユーザを認証および承認するのに役立つ情報も含まれています。



### 注記: Sambaゲートウェイのパフォーマンス

プロトコルオーバーヘッドが増加し、クライアントとストレージ間の余分なネットワークホップによって追加の遅延が発生するため、Sambaゲートウェイ経由でCephFSにアクセスすると、ネイティブのCephFSクライアントと比較して、アプリケーションのパフォーマンスが大幅に低下する場合があります。

## 24.1 Samba共有を介したCephFSのエクスポート



### 警告: クロスプロトコルアクセス

ネイティブのCephFSおよびNFSクライアントは、Sambaを介して取得されるファイルロックによる制限を受けません。また、その逆も同様です。クロスプロトコルファイルロックに依存するアプリケーションでは、CephFSを利用するSamba共有パスに他の手段でアクセスした場合、データの破壊が発生することがあります。

### 24.1.1 Sambaパッケージの設定とエクスポート

Samba共有を設定およびエクスポートするには、パッケージ `samba-ceph` および `samba-winbind` をインストールします。これらのパッケージがインストールされていない場合、インストールします。

```
cephuser@smb > zypper install samba-ceph samba-winbind
```

### 24.1.2 ゲートウェイが1つの場合の例

Samba共有をエクスポートする準備として、Sambaゲートウェイとして動作する適切なノードを選択します。このノードは、Cephクライアントネットワークに加え、十分なCPU、メモリ、およびネットワーキングリソースにアクセスする必要があります。

フェールオーバー機能は、CTDBとSUSE Linux Enterprise High Availability Extensionで提供できます。HAセットアップの詳細については、[24.1.3項「高可用性の設定」](#)を参照してください。

1. クラスタ内に動作中のCephFSがすでに存在することを確認します。
2. Ceph管理ノード上にSambaゲートウェイに固有のキーリングを作成して、両方のSambaゲートウェイノードにコピーします。

```
cephuser@adm > ceph auth get-or-create client.samba.gw mon 'allow r' \
osd 'allow *' mds 'allow *' -o ceph.client.samba.gw.keyring
cephuser@adm > scp ceph.client.samba.gw.keyring SAMBA_NODE:/etc/ceph/
```

SAMBA\_NODEは、Sambaゲートウェイノードの名前に置き換えます。

3. Sambaゲートウェイノードで次の手順を実行します。Ceph統合パッケージとともにSambaをインストールします。

```
cephuser@smb > sudo zypper in samba samba-ceph
```

4. /etc/samba/smb.confファイルのデフォルトの内容を以下に置き換えます。

```
[global]
netbios name = SAMBA-GW
clustering = no
idmap config * : backend = tdb2
passdb backend = tdbsam
# disable print server
load printers = no
smbd: backgroundqueue = no

[SHARE_NAME]
path = CEPHFS_MOUNT
read only = no
oplocks = no
kernel share modes = no
```

先のCEPHFS\_MOUNTパスは、カーネルCephFS共有設定でSambaを起動する前にマウントする必要があります。[23.3項「/etc/fstabでのCephFSのマウント」](#)を参照してください。

この共有設定はLinuxカーネルのCephFSクライアントを使用しています。パフォーマンス上の理由から、この設定をお勧めします。もしくは、Samba vfs\_cephモジュールをCephクラスタとの通信に使用することもできます。設定内容を以下に示します。この方法は旧来の用途向けであり、新しいSambaの展開にはお勧めしません。

```
[SHARE_NAME]
```

```
path = /
vfs objects = ceph
ceph: config_file = /etc/ceph/ceph.conf
ceph: user_id = samba.gw
read only = no
oplocks = no
kernel share modes = no
```



## ヒント: Oplocksと共有モード

oplocks (SMB2+ leasesとしても知られている)は、積極的なクライアントキャッシングによってパフォーマンスを向上させることができますが、現在のところ、Sambaが他のCephFSクライアント(カーネルmount.cephfs、FUSE、NFS Ganeshaなど)とともに展開されている場合は安全ではありません。

すべてのCephFSファイルシステムパスアクセスをSambaで排他的に処理する場合は、oplocksパラメータを有効化しても安全です。

現在のところ、ファイルサービスを正しく動作させるには、CephFS vfsモジュールで実行されている共有では、kernel share modesを無効にする必要があります。



## 重要: アクセスの許可

SambaはSMBユーザとグループをローカルアカウントにマッピングします。次のコマンドにより、Samba共有アクセス用のパスワードをローカルユーザに割り当てることができます。

```
root # smbpasswd -a USERNAME
```

I/Oを正常に実行するには、共有パスのアクセス制御リスト(ACL)を使用して、Samba経由で接続されているユーザへのアクセスを許可する必要があります。ACLを変更するには、CephFSカーネルクライアントを介して一時的にマウントし、共有パスに対してchmod、chown、またはsetfaclのユーティリティを使用します。たとえば、すべてのユーザに対してアクセスを許可するには、次のコマンドを実行します。

```
root # chmod 777 MOUNTED_SHARE_PATH
```

### 24.1.2.1 Sambaサービスの起動

次のコマンドを使用して、スタンドアロンのSambaサービスを起動または再起動します。

```
root # systemctl restart smb.service
root # systemctl restart nmb.service
root # systemctl restart winbind.service
```

Sambaサービスがブート時に起動するようにするには、次のコマンドで有効化します。

```
root # systemctl enable smb.service
root # systemctl enable nmb.service
root # systemctl enable winbind.service
```



#### ヒント: オプションのnmbサービスとwinbindサービス

ネットワーク共有ブラウジングが不要な場合は、nmbサービスの有効化と起動は不要です。

winbindサービスは、Active Directoryドメインメンバーとして設定する場合にのみ必要です。24.2項「SambaゲートウェイとActive Directoryの参加」を参照してください。

### 24.1.3 高可用性の設定



#### 重要: 透過的なフェールオーバーはサポートされない

Samba + CTDBのマルチノード展開では単一ノードと比較して可用性が高くなりますが(第24章「Sambaを介したCephデータのエクスポート」を参照)、クライアント側の透過的なフェールオーバーはサポートされていません。Sambaゲートウェイノードで障害が発生した場合、アプリケーションが短時間停止する可能性があります。

このセクションでは、Sambaサーバの2ノード高可用性設定の方法について例を使って説明します。このセットアップでは、SUSE Linux Enterprise High Availability Extensionが必要です。これら2つのノードは、earth (192.168.1.1)およびmars (192.168.1.2)という名前です。

SUSE Linux Enterprise High Availability Extensionの詳細については、<https://documentation.suse.com/sle-ha/15-SP1/>を参照してください。

さらに、2つの浮動仮想IPアドレスにより、実行している物理ノードがどれであれ、クライアントからの該当サービスへの接続が可能になります。Hawk2でのクラスタ管理には192.168.1.10を使用し、CIFSエクスポートには192.168.2.1を排他的に使用します。これにより、後で簡単にセキュリティ制約を適用できます。

次の手順では、インストールの例について説明します。詳細については、<https://documentation.suse.com/sle-ha/15-SP2/html/SLE-HA-all/art-sleha-install-quick.html> を参照してください。

1. 管理ノード上にSambaゲートウェイに固有のキーリングを作成して、両方のノードにコピーします。

```
cephuser@adm > ceph auth get-or-create client.samba.gw mon 'allow r' \
    osd 'allow *' mds 'allow *' -o ceph.client.samba.gw.keyring
cephuser@adm > scp ceph.client.samba.gw.keyring earth:/etc/ceph/
cephuser@adm > scp ceph.client.samba.gw.keyring mars:/etc/ceph/
```

2. SLE-HAセットアップには、アクティブクラスタノードが非同期状態になった場合に「スプリットブレイン」状態に陥ることを避けるため、フェンシングデバイスが必要です。このため、SBD (Stonith Block Device)を使用したCeph RBDイメージを使用できます。詳細については、<https://documentation.suse.com/sle-ha/15-SP2/html/SLE-HA-all/cha-ha-storage-protect.html#sec-ha-storage-protect-fencing-setup> を参照してください。

RBDイメージが存在しない場合は、`rbd`という名前のRBDプールを作成し(18.1項「プールの作成」を参照してください)、`rbd`を関連付けます(18.5.1項「プールとアプリケーションの関連付け」を参照してください)。そして、`sbd01`という名前の関連するRBDイメージを作成します。

```
cephuser@adm > ceph osd pool create rbd
cephuser@adm > ceph osd pool application enable rbd rbd
cephuser@adm > rbd -p rbd create sbd01 --size 64M --image-shared
```

3. `earth`および`mars`を、Sambaサービスをホストするように準備します。

- a. 次のパッケージがインストールされていることを確認してから進んでください:  
`ctdb`、`tdb-tools`、および `samba`をインストールします。

```
root # zypper in ctdb tdb-tools samba samba-ceph
```

- b. SambaサービスとCTDBサービスが停止され、無効化されていることを確認します。

```
root # systemctl disable ctdb
root # systemctl disable smb
root # systemctl disable nmb
root # systemctl disable winbind
root # systemctl stop ctdb
root # systemctl stop smb
root # systemctl stop nmb
```

```
root # systemctl stop winbind
```

- c. すべてのノードのファイアウォールのポート4379を開きます。これは、CTDBが他のクラスタノードと通信するために必要です。
4. earth上にSambaの設定ファイルを作成します。これらは、後で自動的にmarsに同期します。
  - a. Sambaゲートウェイノード上のプライベートIPアドレスのリストを/etc/ctdb/nodesファイルに挿入します。詳細については、ctdbのマニュアルのページ([man 7 ctdb](#))を参照してください。

```
192.168.1.1
192.168.1.2
```

- b. Sambaを設定します。/etc/samba/smb.confの[global]セクションに次の行を追加します。CTDB-SERVERの代わりに、任意のホスト名を使用します(クラスタ内のすべてのノードは、この名前を持つ1つの大きなノードとして表示されます)。共有の定義も追加します。一例として、SHARE\_NAMEを検討してください。

```
[global]
netbios name = SAMBA-HA-GW
clustering = yes
idmap config * : backend = tdb2
passdb backend = tdbsam
ctdbd socket = /var/lib/ctdb/ctdb.socket
# disable print server
load printers = no
smbd: backgroundqueue = no

[SHARE_NAME]
path = /
vfs objects = ceph
ceph: config_file = /etc/ceph/ceph.conf
ceph: user_id = samba.gw
read only = no
oplocks = no
kernel share modes = no
```

すべてのSambaゲートウェイノードで/etc/ctdb/nodesのファイルと/etc/samba/smb.confのファイルが一致している必要があることに注意してください。

5. SUSE Linux Enterprise High Availabilityクラスタをインストールして起動します。
  - a. earthおよびmarsでSUSE Linux Enterprise High Availability Extensionを登録します。

```
root@earth # SUSEConnect -r ACTIVATION_CODE -e E_MAIL
```

```
root@mars # SUSEConnect -r ACTIVATION_CODE -e E_MAIL
```

- b. 両方のノードに `ha-cluster-bootstrap` をインストールします。

```
root@earth # zypper in ha-cluster-bootstrap
```

```
root@mars # zypper in ha-cluster-bootstrap
```

- c. RBDイメージ `sbd01` を両方のSambaゲートウェイに `rbdmmap.service` を介してマッピングします。

`/etc/ceph/rbdmap` を編集して、SBDイメージのエントリを追加します。

```
rbd/sbd01 id=samba.gw,keyring=/etc/ceph/ceph.client.samba.gw.keyring
```

`rbdmmap.service` を有効化し、起動します。

```
root@earth # systemctl enable rbdmap.service && systemctl start rbdmap.service
root@mars # systemctl enable rbdmap.service && systemctl start rbdmap.service
```

`/dev/rbd/rbd/sbd01` のデバイスは両方のSambaゲートウェイで使用可能である必要があります。

- d. `earth` のクラスタを初期化し、`mars` を参加させます。

```
root@earth # ha-cluster-init
```

```
root@mars # ha-cluster-join -c earth
```

## ！ 重要

クラスタの初期化および参加の処理中に、SBDを使用するかどうか対話形式で確認されます。`y` で確定し、ストレージデバイスのパスとして `/dev/rbd/rbd/sbd01` を指定してください。

6. クラスタの状態を確認します。クラスタに2つのノードが追加されたことがわかります。

```
root@earth # crm status
2 nodes configured
1 resource configured
```

```
Online: [ earth mars ]
```

```
Full list of resources:
```

```
admin-ip          (ocf::heartbeat:IPAddr2):      Started earth
```

## 7. earthで次のコマンドを実行して、CTDBリソースを設定します。

```
root@earth # crm configure
crm(live)configure# primitive ctdb ocf:heartbeat:CTDB params \
    ctdb_manages_winbind="false" \
    ctdb_manages_samba="false" \
    ctdb_recovery_lock="!/usr/lib64/ctdb/ctdb_mutex_ceph_rados_helper
    ceph client.samba.gw cephfs_metadata ctdb-mutex"
    ctdb_socket="/var/lib/ctdb/ctdb.socket" \
    op monitor interval="10" timeout="20" \
    op start interval="0" timeout="200" \
    op stop interval="0" timeout="100"
crm(live)configure# primitive smb systemd:smb \
    op start timeout="100" interval="0" \
    op stop timeout="100" interval="0" \
    op monitor interval="60" timeout="100"
crm(live)configure# primitive nmb systemd:nmb \
    op start timeout="100" interval="0" \
    op stop timeout="100" interval="0" \
    op monitor interval="60" timeout="100"
crm(live)configure# primitive winbind systemd:winbind \
    op start timeout="100" interval="0" \
    op stop timeout="100" interval="0" \
    op monitor interval="60" timeout="100"
crm(live)configure# group g-ctdb ctdb winbind nmb smb
crm(live)configure# clone cl-ctdb g-ctdb meta interleave="true"
crm(live)configure# commit
```



### ヒント: オプションのnmbプリミティブとwinbindプリミティブ

ネットワーク共有ブラウジングが不要な場合は、nmbプリミティブの追加は不要です。

winbindプリミティブは、Active Directoryドメインメンバーとして設定する場合にのみ必要です。24.2項「[SambaゲートウェイとActive Directoryの参加](#)」を参照してください。

設定オプション `ctdb_recovery_lock` のバイナリ `/usr/lib64/ctdb/ctdb_mutex_rados_helper` には、パラメータ `CLUSTER_NAME`、`CEPHX_USER`、`CEPH_POOL`、および `RADOS_OBJECT` がこの順序で指定されています。

追加の `lock-timeout` パラメータを付加して、使用されているデフォルト値(10秒)を上書きできます。値を大きくすると、CTDB回復マスタのフェールオーバー時間が長くなり、値を小さくすると、回復マスタがダウンとして不正確に検出され、フラッピングフェールオーバーがトリガされる可能性があります。

## 8. クラスタ対応のIPアドレスを追加します。

```
crm(live)configure# primitive ip ocf:heartbeat:IPAddr2
  params ip=192.168.2.1 \
    unique_clone_address="true" \
    op monitor interval="60" \
    meta resource-stickiness="0"
crm(live)configure# clone cl-ip ip \
  meta interleave="true" clone-node-max="2" globally-unique="true"
crm(live)configure# colocation col-with-ctdb 0: cl-ip cl-ctdb
crm(live)configure# order o-with-ctdb 0: cl-ip cl-ctdb
crm(live)configure# commit
```

`unique_clone_address` が `true` に設定されている場合、`IPAddr2` リソースエージェントはクローンIDを指定のアドレスに追加し、3つの異なるIPアドレスを設定します。これらは通常必要とされませんが、負荷分散に役立ちます。この項目の詳細については、<https://documentation.suse.com/sle-ha/15-SP2/html/SLE-HA-all/cha-ha-lb.html> を参照してください。

## 9. 結果を確認します。

```
root@earth # crm status
Clone Set: base-clone [dlm]
  Started: [ factory-1 ]
  Stopped: [ factory-0 ]
Clone Set: cl-ctdb [g-ctdb]
  Started: [ factory-1 ]
  Started: [ factory-0 ]
Clone Set: cl-ip [ip] (unique)
  ip:0      (ocf:heartbeat:IPAddr2):      Started factory-0
  ip:1      (ocf:heartbeat:IPAddr2):      Started factory-1
```

## 10. クライアントコンピュータからテストを行います。次のコマンドをLinuxクライアントで実行して、システムからファイルをコピーしたり、システムにファイルをコピーできるかどうか確認します。

```
root # smbclient //192.168.2.1/myshare
```

### 24.1.3.1 HA Sambaリソースの再起動

SambaまたはCTDBの設定に何らかの変更を加えた場合、変更を有効にするためHAリソースの再起動が必要になる場合があります。再起動は次のコマンドにより実行できます。

```
root # crm resource restart cl-ctdb
```

## 24.2 SambaゲートウェイとActive Directoryの参加

Ceph Sambaゲートウェイを、AD (Active Directory)をサポートするSambaドメインのメンバーになるように設定できます。Sambaドメインのメンバーは、エクスポートされたCephFSのファイルとディレクトリで、ローカルACL (アクセスリスト)のドメインユーザとグループを使用できます。

### 24.2.1 Sambaのインストール準備

このセクションでは、Samba自体を設定する前に、注意する必要がある準備手順について説明します。クリーンな環境から開始することで、混乱を防ぎ、以前のSambaインストールのファイルが新しいドメインメンバーのインストールと混在しないようにします。



#### ヒント: クロックの同期

すべてのSambaゲートウェイノードのクロックをActive Directoryドメインコントローラと同期する必要があります。クロックスキューがあると、認証が失敗する場合があります。

Sambaまたは名前キャッシュプロセスが実行されていないことを確認します。

```
cephuser@smb > ps ax | egrep "samba|smbd|nmbd|winbindd|nscd"
```

この出力に、samba、smbd、nmbd、winbindd、またはnscdのプロセスが一覧にされる場合は、これらを停止します。

以前にこのホストでSambaのインストールを実行したことがある場合は、`/etc/samba/smb.conf` ファイルを削除します。また、`*.tdb` ファイル、`*.ldb` ファイルなど、Sambaデータベースファイルもすべて削除します。Sambaデータベースが含まれるディレクトリを一覧にするには、次のコマンドを実行します。

```
cephuser@smb > smb -b | egrep "LOCKDIR|STATEDIR|CACHEDIR|PRIVATE_DIR"
```

## 24.2.2 DNSの検証

AD (Active Directory)は、DNSを使用して、Kerberosなどの他のDC (ドメインコントローラ)とサービスを検索します。したがって、ADドメインメンバーとサーバがAD DNSゾーンを解決できる必要があります。

DNSが正しく設定されていること、および前方参照と逆引き参照の両方が正しく解決されることを確認します。次に例を示します。

```
cephuser@adm > nslookup DC1.domain.example.com
Server:          10.99.0.1
Address:         10.99.0.1#53

Name:   DC1.domain.example.com
Address: 10.99.0.1
```

```
cephuser@adm > 10.99.0.1
Server:          10.99.0.1
Address: 10.99.0.1#53

1.0.99.10.in-addr.arpa name = DC1.domain.example.com.
```

## 24.2.3 SRVレコードの解決

ADは、SRVレコードを使用して、KerberosやLDAPなどのサービスを検索します。SRVレコードが正しく解決されることを確認するには、次のように`nslookup`の対話型シェルを使用します。

```
cephuser@adm > nslookup
Default Server:  10.99.0.1
Address: 10.99.0.1

> set type=SRV
> _ldap._tcp.domain.example.com.
Server:  UnKnown
```

```
Address: 10.99.0.1
```

```
_ldap._tcp.domain.example.com  SRV service location:
    priority = 0
    weight   = 100
    port     = 389
    svr hostname = dc1.domain.example.com
domain.example.com  nameserver = dc1.domain.example.com
dc1.domain.example.com  internet address = 10.99.0.1
```

## 24.2.4 Kerberos の設定

Sambaは、HeimdalおよびMIT Kerberosのバックエンドをサポートしています。ドメインメンバーにKerberosを設定するには、`/etc/krb5.conf`ファイルに以下を設定します。

```
[libdefaults]
    default_realm = DOMAIN.EXAMPLE.COM
    dns_lookup_realm = false
    dns_lookup_kdc = true
```

前の例では、DOMAIN.EXAMPLE.COMレルムに対してKerberosを設定します。`/etc/krb5.conf`ファイルには、他のパラメータを設定しないことをお勧めします。`/etc/krb5.conf`に`include`行が含まれる場合、この行は機能しません。この行を削除する「必要があります」。

## 24.2.5 ローカルホスト名の解決

ホストをドメインに参加させる場合、Sambaは、そのホスト名をAD DNSゾーンに登録しようとします。このため、**net**ユーティリティで、DNSまたは`/etc/hosts`ファイルの正しいエントリを使用してホスト名を解決する必要があります。

ホスト名が正しく解決されることを確認するには、**getent hosts**コマンドを使用します。

```
cephuser@adm > getent hosts example-host
10.99.0.5      example-host.domain.example.com  example-host
```

ホスト名とFQDNは、IPアドレス127.0.0.1、またはドメインメンバーのLANインタフェースで使用するアドレス以外のIPアドレスに解決されてはなりません。出力が表示されないか、またはホストが間違ったIPアドレスに解決される場合に、DHCPを使用しないときは、`/etc/hosts`ファイルに正しいエントリを設定します。

```
127.0.0.1      localhost
```



## ヒント: DHCPと/etc/hosts

DHCPを使用する場合、`/etc/hosts`には「127.0.0.1」の行のみが含まれていることを確認します。問題が続く場合は、DHCPサーバの管理者に連絡してください。

マシンのホスト名にエイリアスを追加する必要がある場合は、「127.0.0.1」の行ではなく、マシンのIPアドレスで始まる行の終わりに追加します。

## 24.2.6 Sambaの設定

このセクションでは、Sambaの設定に含める必要がある特定の設定オプションについて説明します。

Active Directoryのドメインメンバーシップの主要な設定を行うには、`/etc/samba/smb.conf`の`[global]`セクションで、`security = ADS`に加えて、適切なKerberosレルムとIDマッピングパラメータを設定します。

```
[global]
security = ADS
workgroup = DOMAIN
realm = DOMAIN.EXAMPLE.COM
...
```

### 24.2.6.1 winbinddでのIDマッピングのバックエンドの選択

ユーザが異なるログインシェルやUnixホームディレクトリパスを使用する必要がある場合、またはユーザにすべての場所で同じIDを使用させたい場合は、winbindの「ad」バックエンドを使用し、RFC2307の属性をADに追加する必要があります。



## 重要: RFC2307の属性とID番号

ユーザまたはグループの作成時に、RFC2307属性は自動的に追加されません。

DCで見つかるID番号(3000000の範囲の番号)は、RFC2307の属性では「ない」ので、Unixドメインメンバーでは使用されません。すべての場所で同じID番号が必要な場合は、`uidNumber`属性と`gidNumber`属性をADに追加し、Unixドメインメンバーで「ad」バックエンドを使用します。`uidNumber`属性と`gidNumber`属性をADに追加する場合は、3000000の範囲の番号を使用しないでください。

ユーザがSamba AD DCを認証にのみ使用し、Samba AD DCにデータを保存したりログインしたりしない場合は、「ind」バックエンドを使用できます。この場合、ユーザとグループのIDはWindows\* RIDから計算されます。すべてのUnixドメインメンバーでsmb.confの同じ[global]セクションを使用している場合は、同じIDが取得されます。「rid」バックエンドを使用する場合は、ADに何も追加する必要はなく、RFC2307の属性は無視されます。

「rid」バックエンドを使用する場合、smb.confでtemplate shellパラメータとtemplate homedirパラメータを設定します。これらの設定はグローバルで、全員が同じログインシェルとUnixホームディレクトリパスを取得します(個別のUnixホームディレクトリパスとシェルを設定可能なRFC2307の属性とは異なります)。

Sambaを設定する方法はもう1つあります。この方法は、すべての場所でユーザとグループに同じIDを設定する必要があるものの、ユーザに同じログインシェルを設定し、ユーザが同じUnixホームディレクトリパスを使用するだけで良い場合に使用できます。このためには、winbindの「rid」バックエンドとsmb.confのテンプレート行を使用します。このように、uidNumber属性とgidNumber属性をADに追加するだけで済みます。



## ヒント: IDマッピングのためのバックエンドの詳細情報

IDマッピングに使用可能なバックエンドの詳細については、関連するマニュアルページのman 8 idmap\_ad、man 8 idmap\_rid、およびman 8 idmap\_autoridを参照してください。

### 24.2.6.2 ユーザとグループのID範囲の設定

使用するwinbindバックエンドを決定した後、smb.confのidmap configオプションで、使用する範囲を指定する必要があります。Unixドメインメンバーでは、複数のブロックのユーザIDとグループIDが最初から予約されています。

表 24.1: ユーザとグループのデフォルトのIDブロック

ID	範囲
0~999	ローカルシステムユーザとグループ
1000から開始	ローカルUnixユーザとグループ
10000から開始	DOMAINユーザとグループ

上の範囲からわかるように、「\*」または「DOMAIN」の範囲を999以下から始まるように設定しないでください。この範囲は、ローカルシステムユーザとグループに干渉するためです。また、ローカルUnixユーザとグループの領域も残しておく必要があるため、`idmap config`の範囲を3000にするのが適切な妥協点である可能性があります。

「DOMAIN」の規模がどのくらい拡大する可能性があるか、および信頼できるドメインを使用する計画があるかどうかを判断する必要があります。その後、`idmap config`の範囲を次のように設定できます。

表 24.2: ID範囲

Domain	範囲
*	3000～7999
DOMAIN	10000～999999
TRUSTED	1000000～9999999

### 24.2.6.3 ローカルrootユーザへのドメイン管理者アカウントのマッピング

Sambaでは、ドメインアカウントをローカルアカウントにマップすることができます。この機能を使用して、クライアントで操作を要求したアカウントとは異なるユーザとして、ドメインメンバーのファイルシステムでファイル操作を実行します。



#### ヒント: ドメイン管理者のマッピング(オプション)

ドメイン管理者をローカル`root`アカウントにマッピングするかどうかはオプションです。このマッピングを設定するのは、ドメイン管理者が`root`許可を使用してドメインメンバーでファイル操作を実行できる必要がある場合だけにしてください。また、管理者を`root`アカウントにマッピングしても、「管理者」としてUnixドメインメンバーにログインすることはできないことに注意してください。

ドメイン管理者をローカル`root`アカウントにマップするには、次の手順に従います。

1. `smb.conf`ファイルの`[global]`セクションに次のパラメータを追加します。

```
username map = /etc/samba/user.map
```

2. 次の内容で`/etc/samba/user.map`ファイルを作成します。

```
!root = DOMAIN\Administrator
```

## ❗ 重要

「ad」のIDマッピングバックエンドを使用する場合は、ドメイン管理者アカウントにuidNumber属性を設定しないでください。アカウントにこの属性が設定されていると、その値によってrootユーザのローカルUID「0」が上書きされるため、マッピングが失敗します。

詳細については、[smb.conf](#)マニュアルページの[username map](#)パラメータ([man 5 smb.conf](#))を参照してください。

## 24.2.7 Active Directory ドメインへの参加

ホストをActive Directoryドメインに参加させるには、次のコマンドを実行します。

```
cephuser@smb > net ads join -U administrator
Enter administrator's password: PASSWORD
Using short domain name -- DOMAIN
Joined EXAMPLE-HOST to dns domain 'DOMAIN.example.com'
```

## 24.2.8 ネームサービススイッチの設定

ドメインユーザとグループをローカルシステムで利用できるようにするには、NSS (ネームサービススイッチ)ライブラリを有効にする必要があります。[/etc/nsswitch.conf](#)ファイルの次のデータベースに[winbind](#)のエントリを追加します。

```
passwd: files winbind
group: files winbind
```

## ！ 重要: 考慮すべきポイント

- 両方のデータベースに対し、filesエントリを最初のソースのままにします。これにより、NSSは、winbindサービスに照会する前に、/etc/passwdファイルと/etc/groupファイルからドメインユーザとグループを検索できます。
- NSS shadowデータベースにはwinbindエントリを追加しないでください。これにより、wbinfoユーティリティが失敗する可能性があります。
- ローカルの/etc/passwdファイルで、ドメイン内のファイルと同じユーザ名を使用しないでください。

## 24.2.9 サービスの起動

設定の変更後、24.1.2.1項「Sambaサービスの起動」または24.1.3.1項「HA Sambaリソースの再起動」に従って、Sambaサービスを再起動してください。

## 24.2.10 winbindd接続のテスト

### 24.2.10.1 winbinddのping送信

winbinddサービスがAD DC (ドメインコントローラ)またはPDC (プライマリドメインコントローラ)に接続できるかどうかを確認するには、次のコマンドを入力します。

```
cephuser@smb > wbinfo --ping-dc
checking the NETLOGON for domain[DOMAIN] dc connection to "DC.DOMAIN.EXAMPLE.COM"
succeeded
```

上のコマンドが失敗する場合は、winbinddサービスが実行されていること、およびsmb.confファイルが正しく設定されていることを確認します。

### 24.2.10.2 ドメインユーザとグループの検索

libnss\_winbindライブラリでは、ドメインユーザとグループを検索できます。たとえば、ドメインユーザ「DOMAIN\demo01」を検索するには、次のコマンドを実行します。

```
cephuser@smb > getent passwd DOMAIN\\demo01  
DOMAIN\demo01:*:10000:10000:demo01:/home/demo01:/bin/bash
```

ドメイングループ「Domain Users」を検索するには、次のコマンドを実行します。

```
cephuser@smb > getent group "DOMAIN\\Domain Users"  
DOMAIN\domain users:x:10000:
```

### 24.2.10.3 ドメインユーザとグループへのファイル許可の割り当て

NSS (ネームサービススイッチ)ライブラリでは、ドメインユーザアカウントとグループをコマンドで使用できます。たとえば、ファイルの所有者を「demo01」ドメインユーザに設定し、グループを「Domain Users」ドメイングループに設定するには、次のコマンドを入力します。

```
cephuser@smb > chown "DOMAIN\\demo01:DOMAIN\\domain users" file.txt
```

## 25 NFS Ganesha

NFS Ganeshaは、オペレーティングシステムカーネルの一部としてではなく、ユーザアドレススペースで動作するNFSサーバです。NFS Ganeshaを使用することで、Cephなど独自のストレージメカニズムをプラグインして、任意のNFSクライアントからアクセスできます。詳細については、『導入ガイド』、第5章「cephadmによる展開」、5.4.3.6項「NFS Ganeshaの展開」を参照してください。



### 注記: NFS Ganeshaのパフォーマンス

NFSゲートウェイ経由でCephにアクセスすると、ネイティブのCephFSと比較してアプリケーションのパフォーマンスが大幅に低下する場合があります。これは、プロトコルオーバーヘッドが増加し、クライアントとストレージ間の余分なネットワークホップによって追加の遅延が発生するためです。

各NFS Ganeshaサービスは次のような階層状の設定から構成されます。

- ブートストラップ用の`ganesh.conf`
- サービスごとのRADOS共通設定オブジェクト
- エクスポートごとのRADOS設定オブジェクト

ブートストラップ設定は、コンテナ内で`nfs-ganesha`デーモンを起動するために最小限必要な設定です。各ブートストラップ設定には`%url`ディレクティブが含まれます。このディレクティブにより、必要に応じてRADOS共通設定オブジェクトから追加の設定が読み込まれます。共通設定オブジェクトには追加の`%url`ディレクティブを含めることができます。このディレクティブはエクスポートRADOS設定オブジェクトで定義された各NFSエクスポートを対象とするものです。

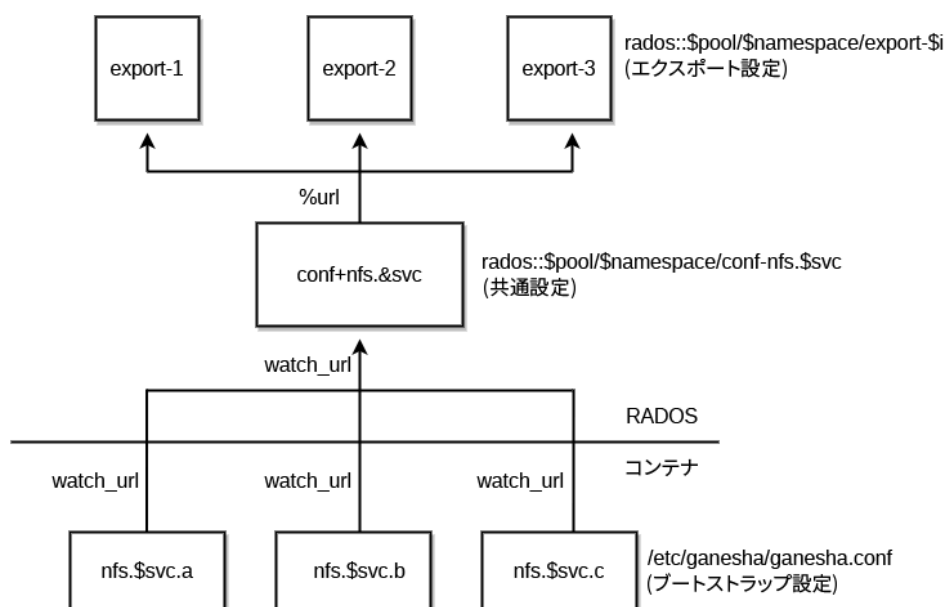


図 25.1: NFS GANESHAの構造

## 25.1 NFSサービスの作成

Cephサービスの展開内容を指定する方法としては、YAMLフォーマットのファイルを作成して、展開したいサービスの仕様を記載することをお勧めします。サービスの種類ごとに個別の仕様ファイルを作成できます。また、複数(もしくは、すべて)の種類のサービスを1つのファイルで指定することも可能です。

選択した方法に応じて、NFS Ganeshaサービスを作成するために関連するYAMLフォーマットのファイルのアップデートまたは作成が必要になります。ファイルの作成の詳細については、『導入ガイド』、第5章「cephadmによる展開」、5.4.2項「サービス仕様と配置仕様」を参照してください。

ファイルのアップデートまたは作成が完了したら、次のコマンドを実行してnfs-ganeshaサービスを作成してください。

```
cephuser@adm > ceph orch apply -i FILE_NAME
```

## 25.2 NFS Ganeshaの起動または再起動

NFS Ganeshaサービスを起動するには、次のコマンドを実行します。

```
cephuser@adm > ceph orch start nfs.SERVICE_ID
```

NFS Ganeshaサービスを再起動するには、次のコマンドを実行します。

```
cephuser@adm > ceph orch restart nfs.SERVICE_ID
```

単一のNFS Ganeshaデーモンを再起動したいだけなら、次のコマンドを実行します。

```
cephuser@adm > ceph orch daemon restart nfs.SERVICE_ID
```

NFS Ganeshaが起動または再起動した時点では、NFS v4に90秒の猶予タイムアウトが設定されています。猶予期間中、クライアントからの新しい要求はアクティブに拒否されます。したがって、NFSが猶予期間の場合、クライアントで要求の低速化が発生することがあります。

## 25.3 NFS回復プールのオブジェクトの一覧

NFS回復プールのオブジェクトを一覧にするには、次のコマンドを実行します。

```
cephuser@adm > rados --pool POOL_NAME --namespace NAMESPACE_NAME ls
```

## 25.4 NFSエクスポートの作成

NFSエクスポートを作成するには、次のコマンドを実行します。



### 注記

希望するcephxユーザIDとシークレットアクセスキーが含まれるように、FSALブロックを修正する必要があります。

```
cephuser@adm > rados --pool POOL_NAME --namespace NAMESPACE_NAME put export-1 export-1
```

## 25.5 NFSエクスポートの確認

NFS v4は疑似ファイルシステムのルートにエクスポートのリストを作成します。NFS Ganeshaサーバノードの/をマウントすることで、NFS共有がエクスポートされたことを確認できます。

```
root # mount -t nfs nfs_ganesha_server_hostname:/ /path/to/local/mountpoint
root # ls /path/to/local/mountpoint cephfs
```



## 注記: NFS Ganeshaはv4のみ

デフォルトでは、cephadmがNFS v4サーバを設定します。NFS v4は`rpcbind`デーモンとも`mountd`デーモンとも対話しません。`showmount`などのNFSクライアントツールは設定済みエクスポートを表示しません。

## 25.6 NFSエクスポートのマウント

エクスポートされたNFS共有をクライアントホストにマウントするには、次のコマンドを実行します。

```
root # mount -t nfs nfs_ganesha_server_hostname:/ /path/to/local/mountpoint
```

## 25.7 複数のNFS Ganeshaクラスタ

複数のNFS Ganeshaクラスタを定義できます。これにより、次のことが実現できます。

- CephFSへのアクセスのために分離されたNFS Ganeshaクラスタ。

## V 仮想化ツールとの統合

26    libvirtとCeph **350**

27    QEMU KVMインスタンスのバックエンドとしてのCephの使用 **356**

## 26 libvirtとCeph

libvirtライブラリは、ハイパーバイザーインタフェースと、それらを使用するソフトウェアアプリケーションとの間に仮想マシン抽象化層を作成します。libvirtを使用することにより、開発者やシステム管理者は、QEMU/KVM、Xen、LXC、VirtualBoxなど、さまざまなハイパーバイザーに対する共通管理フレームワーク、共通API、および共通シェルインタフェース(**virsh**)に集中できます。

Ceph Block DeviceはQEMU/KVMをサポートします。libvirtと連動するソフトウェアでCeph Block Deviceを使用できます。クラウドソリューションはlibvirtを使用してQEMU/KVMと対話し、QEMU/KVMはlibrbdを介してCeph Block Deviceと対話します。

Ceph Block Deviceを使用するVMを作成するには、以降のセクションの手順を使用します。これらの例では、プール名にlibvirt-pool、ユーザ名にclient.libvirt、イメージ名にnew-libvirt-imageをそれぞれ使用しています。好きな値を使用できますが、後続の手順でコマンドを実行する際に値を置き換えるようにしてください。

### 26.1 libvirtで使用するためのCephの設定

libvirtで使用するためにCephを設定するには、次の手順を実行します。

1. プールを作成します。次の例では、プール名libvirt-poolと128の配置グループを使用しています。

```
cephuser@adm > ceph osd pool create libvirt-pool 128 128
```

プールが存在することを確認します。

```
cephuser@adm > ceph osd lspools
```

2. Cephユーザを作成します。次の例では、Cephユーザ名client.libvirtを使用し、libvirt-poolを参照しています。

```
cephuser@adm > ceph auth get-or-create client.libvirt mon 'profile rbd' osd \
'profile rbd pool=libvirt-pool'
```

名前が存在することを確認します。

```
cephuser@adm > ceph auth list
```



### 注記: ユーザ名またはID

`libvirt`は、Cephユーザ名`client.libvirt`ではなくID `libvirt`を使用してCephにアクセスします。IDと名前の違いの詳細については、[30.2.1.1項「ユーザ」](#)を参照してください。

3. QEMUを使用してRBDプール内にイメージを作成します。次の例では、イメージ名`new-libvirt-image`を使用し、`libvirt-pool`を参照しています。



### ヒント: キーリングファイルの場所

`libvirt`ユーザキーは、`/etc/ceph`ディレクトリに配置されたキーリングファイルに保存されます。キーリングファイルには、それが属するCephクラスタの名前が含まれた適切な名前が付いている必要があります。デフォルトのクラスタ名「ceph」の場合、キーリングファイル名は`/etc/ceph/ceph.client.libvirt.keyring`になります。

キーリングが存在しない場合は、次のコマンドで作成します。

```
cephuser@adm > ceph auth get client.libvirt > /etc/ceph/  
ceph.client.libvirt.keyring
```

```
root # qemu-img create -f raw rbd:libvirt-pool/new-libvirt-image:id=libvirt 2G
```

イメージが存在することを確認します。

```
cephuser@adm > rbd -p libvirt-pool ls
```

## 26.2 VMマネージャの準備

`libvirt`はVMマネージャなしでも使用できますが、最初のドメインは`virt-manager`で作成する方が簡単です。

1. 仮想マシンマネージャをインストールします。

```
root # zypper in virt-manager
```

2. 仮想化して実行するシステムのOSイメージを準備/ダウンロードします。

3. 仮想マシンマネージャを起動します。

```
virt-manager
```

## 26.3 VMの作成

**virt-manager**でVMを作成するには、次の手順を実行します。

1. リストから接続を選択し、右クリックしてNew (新規作成)を選択します。
2. Import existing disk image (既存のディスクイメージのインポート)を選択し、既存のストレージのパスを入力して、既存のディスクイメージをインポートします。OSタイプとメモリ設定を指定し、名前に仮想マシンの名前を入力します。たとえば、`libvirt-virtual-machine`です。
3. 設定を完了してVMを起動します。
4. `sudo virsh list`を使用して、新しく作成したドメインが存在することを確認します。必要に応じて、次のように接続文字列を指定します。

```
virsh -c qemu+ssh://root@vm_host_hostname/system list
Id      Name                                     State
-----
[...]
9       libvirt-virtual-machine                running
```

5. VMにログインし、Cephで使用するために設定する前にVMを停止します。

## 26.4 VMの設定

この章では、**virsh**を使用してCephとの統合用にVMを設定する方法に焦点を当てて説明します。多くの場合、**virsh**コマンドにはルート特権(**sudo**)が必要で、ルート特権がないと適切な結果が返されません。また、ルート特権が必要なことは通知されません。**virsh**コマンドのリファレンスについては、`man 1 virsh`を参照してください(パッケージ `libvirt-client` をインストールする必要があります)。

1. `virsh edit vm-domain-name`を使用して、設定ファイルを開きます。

```
root # virsh edit libvirt-virtual-machine
```

## 2. <devices>の下位に<disk>エントリが存在する必要があります。

```
<devices>
  <emulator>/usr/bin/qemu-system-SYSTEM-ARCH</emulator>
  <disk type='file' device='disk'>
    <driver name='qemu' type='raw'/>
    <source file='/path/to/image/recent-linux.img'/>
    <target dev='vda' bus='virtio'/>
    <address type='drive' controller='0' bus='0' unit='0'/>
  </disk>
```

/path/to/image/recent-linux.imgは、OSイメージのパスに置き換えてください。

### ！ 重要

テキストエディタではなく、**sudo virsh edit**を使用してください。/  
etc/libvirt/qemuにある設定ファイルをテキストエディタで編集した場合、libvirtが変更を認識しないことがあります。/etc/libvirt/qemuにあるXMLファイルの内容と**sudo virsh dumpxml vm-domain-name**の結果に違いがある場合、VMが適切に動作しないことがあります。

## 3. 前に作成したCeph RBDイメージを<disk>エントリとして追加します。

```
<disk type='network' device='disk'>
  <source protocol='rbd' name='libvirt-pool/new-libvirt-image'>
    <host name='monitor-host' port='6789'/>
  </source>
  <target dev='vda' bus='virtio'/>
</disk>
```

monitor-hostを実際のホスト名に置き換え、必要に応じてプールまたはイメージ、あるいはその両方の名前を置き換えてください。Ceph Monitor用に複数の<host>エントリを追加できます。dev属性は、VMの/devディレクトリに表示される論理デバイス名です。オプションのbus属性は、エミュレートするディスクデバイスのタイプを示します。有効な設定はドライバ固有です(たとえば、ide、scsi、virtio、xen、usb、sataなど)。

## 4. ファイルを保存します。

## 5. Cephクラスタで認証が有効になっている場合(デフォルト)、秘密を生成する必要があります。好みのエディタを開き、次の内容でsecret.xmlという名前のファイルを作成します。

```
<secret ephemeral='no' private='no'>
```

```
<usage type='ceph'>
  <name>client.libvirt secret</name>
</usage>
</secret>
```

## 6. 秘密を定義します。

```
root # virsh secret-define --file secret.xml
<uuid of secret is output here>
```

## 7. `client.libvirt`の鍵を取得して、鍵の文字列をファイルに保存します。

```
cephuser@adm > ceph auth get-key client.libvirt | sudo tee client.libvirt.key
```

## 8. 秘密のUUIDを設定します。

```
root # virsh secret-set-value --secret uuid of secret \
--base64 $(cat client.libvirt.key) && rm client.libvirt.key secret.xml
```

さらに、前に入力した`<disk>`要素に次の`<auth>`エントリを追加することにより、秘密を手動で設定する必要があります(uuidの値は、上のコマンドライン例の結果に置き換えます)。

```
root # virsh edit libvirt-virtual-machine
```

続いて、ドメイン設定ファイルに`<auth></auth>`要素を追加します。

```
...
</source>
<auth username='libvirt'>
  <secret type='ceph' uuid='9ec59067-fdbc-a6c0-03ff-df165c0587b8' />
</auth>
<target ...
```



## 注記

この例で使用しているIDは`libvirt`で、26.1項「`libvirt`で使用するためのCephの設定」の手順2で生成したCephユーザ名`client.libvirt`ではありません。生成したCephユーザ名のID部分を使用するようにしてください。何らかの理由で秘密を再生成する必要がある場合は、もう一度`sudo virsh secret-set-value`を実行する前に、`sudo virsh secret-undefine uuid`を実行する必要があります。

## 26.5 まとめ

Cephで使用するためにVMを設定したら、VMを起動できます。VMとCephが通信していることを確認するには、次の手順を実行できます。

1. Cephが動作しているかどうかを確認します。

```
cephuser@adm > ceph health
```

2. VMが動作しているかどうかを確認します。

```
root # virsh list
```

3. VMがCephと通信しているかどうかを確認します。vm-domain-nameはVMドメインの名前に置き換えてください。


```
root # virsh qemu-monitor-command --hmp vm-domain-name 'info block'
```

4. `&target dev='hdb' bus='ide' />`のデバイスが/devまたは/proc/partitionsに表示されるかどうかを確認します。

```
tux > ls /dev  
tux > cat /proc/partitions
```

## 27 QEMU KVMインスタンスのバックエンドとしてのCephの使用

Cephの最も一般的な使用事例として、仮想マシンにBlock Deviceイメージを提供することがあります。たとえば、理想的な設定のOSと関連ソフトウェアを使用して「ゴールデン」イメージを作成できます。続いて、そのイメージのスナップショットを作成します。最後に、スナップショットのクローンを作成します(通常は複数回。詳細については、[20.3項「スナップショット」](#)を参照してください)。スナップショットのコピーオンライトクローンを作成できるということは、新しい仮想マシンを起動するたびにクライアントがイメージ全体をダウンロードしないで済むため、CephはBlock Deviceイメージを仮想マシンに素早くプロビジョニングできることを意味します。

Ceph Block DeviceをQEMU仮想マシンと統合できます。QEMU KVMの詳細については、<https://documentation.suse.com/sles/15-SP2/html/SLES-all/part-virt-qemu.html>  を参照してください。

### 27.1 qemu-block-rbdのインストール

Ceph Block Deviceを使用するには、QEMUに適切なドライバがインストールされている必要があります。`qemu-block-rbd`パッケージがインストールされているかどうかを確認し、必要に応じてインストールします。

```
root # zypper install qemu-block-rbd
```

### 27.2 QEMUの使用

QEMUのコマンドラインで、プール名とイメージ名を指定する必要があります。スナップショット名も指定できます。

```
qemu-img command options \  
rbd:pool-name/image-name@snapshot-name:option1=value1:option2=value2...
```

たとえば、`id`および`conf`のオプションを指定すると、次のようになります。

```
qemu-img command options \  
rbd:pool_name/image_name:id=glance:conf=/etc/ceph/ceph.conf
```

## 27.3 QEMUでのイメージの作成

QEMUからBlock Deviceイメージを作成できます。rbd、プール名、および作成するイメージの名前を指定する必要があります。イメージのサイズも指定する必要があります。

```
qemu-img create -f raw rbd:pool-name/image-name size
```

例:

```
qemu-img create -f raw rbd:pool1/image1 10G
Formatting 'rbd:pool1/image1', fmt=raw size=10737418240 nocow=off cluster_size=0
```



### 重要

RBDで使用するフォーマットオプションとして実用的なものは、実際のところ`raw`データフォーマットだけです。技術的には、`qcow2`などQEMUでサポートされている他のフォーマットを使用できますが、そうするとオーバーヘッドが追加されるほか、キャッシングが有効な場合に、仮想マシンのライブマイグレーションにおいてボリュームが不安定になります。

## 27.4 QEMUでのイメージのサイズ変更

QEMUからBlock Deviceイメージのサイズを変更できます。rbd、プール名、およびサイズを変更するイメージの名前を指定する必要があります。イメージのサイズも指定する必要があります。

```
qemu-img resize rbd:pool-name/image-name size
```

例:

```
qemu-img resize rbd:pool1/image1 9G
Image resized.
```

## 27.5 QEMUでのイメージ情報の取得

QEMUからBlock Deviceイメージの情報を取得できます。rbd、プール名、およびイメージの名前を指定する必要があります。

```
qemu-img info rbd:pool-name/image-name
```

例:

```
qemu-img info rbd:pool1/image1
image: rbd:pool1/image1
file format: raw
virtual size: 9.0G (9663676416 bytes)
disk size: unavailable
cluster_size: 4194304
```

## 27.6 RBDでのQEMUの実行

QEMUは、librbdを介して仮想Block Deviceとしてイメージに直接アクセスできます。これにより、追加のコンテキストスイッチを避け、RBDキャッシングを利用できます。

**qemu-img**を使用して、既存の仮想マシンイメージをCeph Block Deviceイメージに変換できます。たとえば、qcow2イメージがある場合、次のコマンドを実行できます。

```
qemu-img convert -f qcow2 -O raw sles12.qcow2 rbd:pool1/sles12
```

そのイメージから仮想マシンの起動を実行するには、次のコマンドを実行できます。

```
root # qemu -m 1024 -drive format=raw,file=rbd:pool1/sles12
```

RBDキャッシングはパフォーマンスを大幅に向上できます。QEMUのキャッシュオプションでlibrbdのキャッシングを制御します。

```
root # qemu -m 1024 -drive format=rbd,file=rbd:pool1/sles12,cache=writeback
```

RBDキャッシングの詳細については、[20.5項「キャッシュの設定」](#)を参照してください。

## 27.7 discardおよびTRIMの有効化

Ceph Block Deviceはdiscard操作をサポートしています。つまり、ゲストはTRIM要求を送信してCeph Block Deviceに未使用領域を解放させることができます。ゲストでこれを有効にするには、discardオプションを指定してXFSをマウントします。

ゲストがこれを利用できるようにするには、Block Deviceに対して明示的に有効にする必要があります。このためには、ドライブに関連付けられているdiscard\_granularityを指定する必要があります。

```
root # qemu -m 1024 -drive format=raw,file=rbd:pool1/sles12,id=drive1,if=none \
-device driver=ide-hd,drive=drive1,discard_granularity=512
```



## 注記

上の例では、IDEドライバを使用しています。virtioドライブはdiscardをサポートしません。

libvirtを使用する場合、**virsh edit**を使用してlibvirtドメインの設定ファイルを編集し、`xmlns:qemu`の値を含めます。その後、`qemu:commandline block`をそのドメインの子として追加します。次の例に、`qemu id=`を使用して2台のデバイスを異なる`discard_granularity`の値に設定する方法を示します。

```
<domain type='kvm' xmlns:qemu='http://libvirt.org/schemas/domain/qemu/1.0'>
  <qemu:commandline>
    <qemu:arg value='-set' />
    <qemu:arg value='block.scsi0-0-0.discard_granularity=4096' />
    <qemu:arg value='-set' />
    <qemu:arg value='block.scsi0-0-1.discard_granularity=65536' />
  </qemu:commandline>
</domain>
```

## 27.8 QEMUのキャッシュオプションの設定

QEMUのキャッシュオプションは、次のCeph RBDキャッシュ設定に対応します。

ライトバック:

```
rbd_cache = true
```

ライトスルー:

```
rbd_cache = true
rbd_cache_max_dirty = 0
```

None:

```
rbd_cache = false
```

QEMUのキャッシュ設定は、Cephのデフォルト設定(Cephの設定ファイルで明示的に設定されていない設定)を上書きします。Cephの設定ファイルでRBDキャッシュ設定を明示的に設定した場合(20.5項「[キャッシュの設定](#)」を参照)、Cephの設定がQEMUのキャッシュ設定を上書きします。QEMUのコマンドラインでキャッシュ設定を行った場合、QEMUのコマンドラインの設定がCephの設定ファイルの設定を上書きします。

## VI クラスタの設定

- 28 Cephクラスタの設定 **361**
- 29 Ceph Managerモジュール **380**
- 30 cephxを使用した認証 **385**

## 28 Cephクラスタの設定

この章では、設定オプションを使用してCephクラスタを設定する方法を説明します。

### 28.1 ceph.confファイルの設定

cephadmは基本的なceph.confファイルを使用します。このファイルにはMONに接続し、設定情報を認証して取得するための最小限のオプションセットだけが含まれます。ほとんどの場合、使用するのはmon\_hostオプションだけに限られます(しかしながら、DNSのSRVレコードを使用することで、このオプションも不要となります)。

#### ！ 重要

もはやceph.confファイルはクラスタ設定を保存する中心的役割を担うことはなくなっており、主に設定データベースが使用されます(28.2項「設定データベース」を参照してください)。

従来通り、ceph.confファイルを使用してクラスタ設定を変更する必要がある場合(たとえば、設定データベースからのオプションの読み取りをサポートしていないクライアントを使用する場合)、次のコマンドを実行し、クラスタ全体でceph.confファイルの保守と配布を行う必要があります。

```
cephuser@adm > ceph config set mgr mgr/cephadm/manage_etc_ceph_ceph_conf false
```

#### 28.1.1 コンテナイメージ内のceph.confへのアクセス

Cephデーモンはコンテナ内で実行されますが、従来通りceph.conf設定ファイルにアクセスすることができます。設定ファイルはホストシステム上で次のファイルとして「バインドマウント」されています。

```
/var/lib/ceph/CLUSTER_FSID/DAEMON_NAME/config
```

CLUSTER\_FSIDは、**ceph fsid**コマンドで取得できる、実行中のクラスタの固有FSIDで置き換えます。DAEMON\_NAMEは**ceph orch ps**コマンドにより一覧にされる固有のデーモン名で置き換えます。以下に例を示します。

```
/var/lib/ceph/b4b30c6e-9681-11ea-ac39-525400d7702d/osd.2/config
```

デーモンの設定を変更するには、デーモンのconfigファイルを編集し、再起動します。

```
root # systemctl restart ceph-CLUSTER_FSID-DAEMON_NAME
```

以下に例を示します。

```
root # systemctl restart ceph-b4b30c6e-9681-11ea-ac39-525400d7702d-osd.2
```



## 重要

cephadmがデーモンを再展開すると、すべてのカスタム設定は失われます。

## 28.2 設定データベース

Ceph Monitorは、クラスタ全体の動作に影響する設定オプションの中央データベースを管理します。

### 28.2.1 セクションとマスクの設定

MONに保存された設定オプションは「グローバル」セクション、「デーモンタイプ」セクション、または、「特定のデーモン」セクションに記録できます。また、オプションは関連する「マスク」を持つ場合もあり、オプションの適用対象となるデーモンやクライアントを細かく制限できます。マスクには2つの形式があります。

- `TYPE:LOCATION` この場合`TYPE`は`rack`や`host`などのCRUSHプロパティで、`LOCATION`はそのプロパティの値です。  
たとえば、`host:example_host`は特定のホストで実行されるデーモンまたはクライアントだけにオプションを制限します。
- `CLASS:DEVICE_CLASS`の場合。`DEVICE_CLASS`は`hdd`や`ssd`などのCRUSHデバイスクラスの名前です。たとえば、`class:ssd`はSSDにより支援されるOSDだけにオプションを制限します。このマスクはOSDでないデーモンやクライアントには影響しません。

### 28.2.2 設定オプションの設定と読み取り

クラスタの設定オプションの設定または読み取りを行うには、次のコマンドを使用してください。`WHO`パラメータは、セクション名、マスク、またはその両方をスラッシュ(/)記号で区切って組み合わせたものを使用できます。たとえば、`osd/rack:foo`は`foo`という名前のラックに含まれるすべてのOSDデーモンを表します。

### **ceph config dump**

クラスタ全体を対象とする設定データベース全体をダンプします。

### **ceph config get WHO**

設定データベースに保存されている、特定のデーモンまたはクライアント(たとえば、`mds.a`)の設定をダンプします。

### **ceph config set WHO OPTION VALUE**

設定データベースの設定オプションに指定した値を設定します。

### **ceph config show WHO**

報告された実行中のデーモンに関する現在の設定を表示します。ローカルの設定ファイルが同時に使用されている場合や、オプションがランタイム中にオーバーライドされていたり、コマンドラインでオーバーライドされている場合、これらの設定はMonitorが保存した値と異なることもあります。オプション値のソースは出力の一部として報告されます。

### **ceph config assimilate-conf -i INPUT\_FILE -o OUTPUT\_FILE**

`INPUT_FILE`で指定した設定ファイルをインポートし、すべての有効なオプションを設定データベースに保存します。認識されないか、無効であるか、Monitorが制御できない設定は、`OUTPUT_FILE`という名前で保存される、簡略化された設定ファイルに返されます。このコマンドは、旧来の設定ファイルから、中央化されたMonitorベースの設定に移行する際に便利です。

## 28.2.3 ランタイム中のデーモンの設定

ほとんどの場合、Cephによりランタイム中のデーモンの設定を変更できます。これは、ログ出力の量を増減する必要がある場合や、ランタイム中にクラスタの最適化を行う場合に有用です。

次のコマンドを使用して、設定オプションの値を更新できます。

```
cephuser@adm > ceph config set DAEMON OPTION VALUE
```

たとえば、特定のOSDでデバッグログのレベルを調整するには、次のコマンドを実行します。

```
cephuser@adm > ceph config set osd.123 debug_ms 20
```



### 注記

同じオプションがローカルの設定ファイルでもカスタマイズされている場合、Monitorの設定は無視されます。これは、設定ファイルよりも優先度が低いからです。

### 28.2.3.1 値のオーバーライド

**tell**または**daemon**サブコマンドを使用して、オプションの値を一時的に変更することができます。この変更は実行中のプロセスだけに影響し、デーモンやプロセスが再起動すると破棄されます。

値をオーバーライドする方法は2つあります。

- **tell**サブコマンドを使用して、いずれかのクラスターノードから特定のデーモンにメッセージを送る方法。

```
cephuser@adm > ceph tell DAEMON config set OPTION VALUE
```

以下に例を示します。

```
cephuser@adm > ceph tell osd.123 config set debug_osd 20
```



#### ヒント

**tell**サブコマンドはデーモンの識別にワイルドカードを使用できます。たとえば、すべてのOSDデーモンでデバッグレベルを調整するには、次のコマンドを実行します。

```
cephuser@adm > ceph tell osd.* config set debug_osd 20
```

- **daemon**サブコマンドを使用して、デーモンのプロセスを実行中のノードから`/var/run/ceph`のソケットを介して特定のデーモンのプロセスに接続する方法。

```
cephuser@adm > ceph daemon DAEMON config set OPTION VALUE
```

以下に例を示します。

```
cephuser@adm > ceph daemon osd.4 config set debug_osd 20
```



#### ヒント

**ceph config show**コマンドを使用してランタイム中の設定を表示すると(28.2.3.2項「ランタイム設定の表示」を参照してください)、一時的にオーバーライドされた値は、overrideをソースとして表示されます。

### 28.2.3.2 ランタイム設定の表示

デーモンに設定されているすべてのオプションを表示するには、次のコマンドを実行します。

```
cephuser@adm > ceph config show-with-defaults osd.0
```

デーモンに設定されている、デフォルトではないすべてのオプションを表示するには、次のコマンドを実行します。

```
cephuser@adm > ceph config show osd.0
```

特定のオプションを調べるには、次のコマンドを実行します。

```
cephuser@adm > ceph config show osd.0 debug_osd
```

デーモンのプロセスを実行しているノードから実行中のデーモンに接続し、設定を確認することも可能です。

```
cephuser@adm > ceph daemon osd.0 config show
```

デフォルトではない設定だけを表示するには、次のコマンドを実行します。

```
cephuser@adm > ceph daemon osd.0 config diff
```

特定のオプションを調べるには、次のコマンドを実行します。

```
cephuser@adm > ceph daemon osd.0 config get debug_osd
```

## 28.3 config-keyの保存

config-keyはCeph Monitorが提供する多目的サービスです。キーと値のペアを永続的に保存することで、設定キーの管理を容易にします。config-keyを使用するのは主にCephのツールとデーモンです。



### ヒント

キーの追加や既存のキーの変更を行ったら、影響を受けるサービスを再起動して変更を有効にします。Cephサービスの操作の詳細については、[第14章「Cephサービスの運用」](#)を参照してください。

**Config-key**コマンドを使用して、config-keyストアを操作します。**config-key**コマンドと共に、以下のサブコマンドを使用します。

#### **`ceph config-key rm KEY`**

指定したキーを削除します。

#### **`ceph config-key exists KEY`**

指定したキーの有無をチェックします。

#### **`ceph config-key get KEY`**

指定したキーの値を取得します。

#### **`ceph config-key ls`**

すべてのキーを一覧にします。

#### **`ceph config-key dump`**

すべてのキーとその値をダンプします。

#### **`ceph config-key set KEY VALUE`**

指定したキーとその値を保存します。

## 28.3.1 iSCSI Gateway

iSCSI Gatewayは`config-key`ストアを使用して、設定オプションの保存と読み取りを行います。すべてのiSCSI Gatewayに関連するキーはプレフィックスとして`iscsi`という文字列がきます。次に例を示します。

```
iscsi/trusted_ip_list
iscsi/api_port
iscsi/api_user
iscsi/api_password
iscsi/api_secure
```

たとえば、2セットの設定オプションが必要な場合は、別の記述キーワードでプレフィックスを拡張します。たとえば、`datacenterA`と`datacenterB`を追加すると次のようになります。

```
iscsi/datacenterA/trusted_ip_list
iscsi/datacenterA/api_port
[...]
iscsi/datacenterB/trusted_ip_list
iscsi/datacenterB/api_port
[...]
```

## 28.4 Ceph OSDとBlueStore

### 28.4.1 自動キャッシュサイズ調整の設定

`tc_malloc`がメモリアロケータとして設定されていて、`bluestore_cache_autotune`設定が有効になっている場合、BlueStoreを、そのキャッシュサイズを自動的に調整するように設定できます。このオプションは現在、デフォルトで有効です。BlueStoreは、`osd_memory_target`設定オプションを使用して、OSDのヒープメモリ使用量を指定されたターゲットサイズに維持しようとします。これはベストエフォート型のアルゴリズムで、`osd_memory_cache_min`で指定されている量よりも小さいサイズにキャッシュが縮小されることはありません。キャッシュ比率は、優先度の階層に基づいて選択されます。優先度情報が使用できない場合は、代わりに`bluestore_cache_meta_ratio`オプションと`bluestore_cache_kv_ratio`オプションが使用されます。

#### **bluestore\_cache\_autotune**

最小値を優先しながら、異なるBlueStoreキャッシュに割り当てられる比率を自動的に調整します。デフォルトは`True`です。

#### **osd\_memory\_target**

`tc_malloc`および`bluestore_cache_autotune`が有効な場合、この量のバイトをメモリ内マップした状態を保持しようとします。



#### **注記**

これは、プロセスのRSSメモリ使用量と正確には一致しない場合があります。通常、プロセスによってマップされたヒープメモリの合計量は、このターゲットに近い値を維持しますが、マップ解除済みのメモリをカーネルが実際に再利用する保証はありません。

#### **osd\_memory\_cache\_min**

`tc_malloc`および`bluestore_cache_autotune`が有効な場合に、キャッシュに使用するメモリの最小量を設定します。



#### **注記**

この値を低く設定しすぎると、多大なキャッシュスラッシングが発生する可能性があります。

## 28.5 Ceph Object Gateway

いくつかのオプションによりObject Gatewayの動作を間接的に操作することができます。オプションを指定しない場合は、そのデフォルト値が使用されます。次に、Object Gatewayのすべてのオプションのリストを示します。

### 28.5.1 一般的な設定

#### **rgw\_frontends**

HTTPフロントエンドを設定します。複数のフロントエンドがある場合は、各項目のカンマ区切りリストを指定します。各フロントエンド設定には、スペースで区切ったオプションのリストを含めることができます。この場合、各オプションは「キー=値」または「キー」の形式になります。デフォルトは`beast port=7480`です。

#### **rgw\_data**

Object Gatewayのデータファイルの場所を設定します。デフォルトは`/var/lib/ceph/radosgw/CLUSTER_ID`です。

#### **rgw\_enable\_apis**

指定したAPIを有効にします。デフォルトは「s3, swift, swift\_auth, admin All APIs」です。

#### **rgw\_cache\_enabled**

Object Gatewayキャッシュを有効/無効にします。デフォルトは`true`です。

#### **rgw\_cache\_lru\_size**

Object Gatewayキャッシュのエントリの数。デフォルトは10000です。

#### **rgw\_socket\_path**

ドメインソケットのソケットパス。`FastCgiExternalServer`は、このソケットを使用します。ソケットパスを指定しない場合、Object Gatewayは外部サーバとして実行されません。ここで指定するパスは、`rgw.conf`ファイルで指定するパスと同じである必要があります。

#### **rgw\_fcgi\_socket\_backlog**

fcgiのソケットバックログ。デフォルトは1024です。

#### **rgw\_host**

Object Gatewayインスタンスのホスト。IPアドレスまたはDNS名を指定できます。デフォルトは`0.0.0.0`です。

**rgw\_port**

インスタンスが要求をリスンするポート番号。指定されていない場合、Object Gatewayは外部のFastCGIを実行します。

**rgw\_dns\_name**

サービス対象ドメインのDNS名。

**rgw\_script\_uri**

SCRIPT\_URIが要求で設定されていない場合の代替値。

**rgw\_request\_uri**

REQUEST\_URIが要求で設定されていない場合の代替値。

**rgw\_print\_continue**

100-continueが使用可能な場合、これを有効にします。デフォルトはtrueです。

**rgw\_remote\_addr\_param**

リモートアドレスパラメータ。たとえば、リモートアドレスが含まれるHTTPフィールド、またはリバースプロキシが使用可能な場合はX-Forwarded-Forアドレス。デフォルトはREMOTE\_ADDRです。

**rgw\_op\_thread\_timeout**

開いているスレッドのタイムアウト(秒)。デフォルトは600です。

**rgw\_op\_thread\_suicide\_timeout**

Object Gatewayプロキシが停止するまでのタイムアウト(秒)。0 (デフォルト)に設定すると無効になります。

**rgw\_thread\_pool\_size**

Beastサーバのスレッドの数。より多くの要求を実行する必要がある場合は、値を増やします。デフォルトは100スレッドです。

**rgw\_num\_rados\_handles**

Object GatewayのRADOSクラスタハンドルの数。Object Gatewayの各ワーカスレッドは、その有効期間中、RADOSハンドルを選択するようになりました。このオプションは今後のリリースで廃止され、削除される可能性があります。デフォルトは1です。

**rgw\_num\_control\_oids**

異なるObject Gatewayインスタンス間のキャッシュ同期に使用する通知オブジェクトの数。デフォルトは8です。

**rgw\_init\_timeout**

Object Gatewayが初期化を中止するまでの秒数。デフォルトは30です。

### **rgw\_mime\_types\_file**

MIMEタイプのパスと場所。Swiftによるオブジェクトタイプの自動検出に使用します。  
デフォルトは`/etc/mime.types`です。

### **rgw\_gc\_max\_objs**

1つのガベージコレクション処理サイクルでガベージコレクションによって処理できるオブジェクトの最大数。デフォルトは32です。

### **rgw\_gc\_obj\_min\_wait**

ガベージコレクション処理によってオブジェクトを削除および処理するまでの最大待機時間。デフォルトは $2 * 3600$ です。

### **rgw\_gc\_processor\_max\_time**

2つの連続するガベージコレクション処理サイクルを開始する間隔の最大時間。デフォルトは3600です。

### **rgw\_gc\_processor\_period**

ガベージコレクション処理のサイクル時間。デフォルトは3600です。

### **rgw\_s3\_success\_create\_obj\_status**

`create-obj`に対する代替の成功ステータス応答。デフォルトは0です。

### **rgw\_resolve\_cname**

Object Gatewayが要求ホスト名フィールドのDNS CNAMEレコードを使用する必要があるかどうか(ホスト名がObject Gateway DNS名に等しくない場合)。デフォルトは`false`です。

### **rgw\_obj\_stripe\_size**

Object Gatewayオブジェクトのオブジェクトストライプのサイズ。デフォルトは`4 << 20`です。

### **rgw\_extended\_http\_attrs**

エンティティ(たとえば、ユーザ、バケット、オブジェクト)に設定できる一連の新しい属性を追加します。これらの追加属性は、エンティティを配置したり、POSTメソッドを使用して変更したりする場合に、HTTPヘッダフィールドによって設定できます。設定されている場合、これらの属性は、エンティティに対してGET/HEADを要求したときにHTTPフィールドとして返されます。デフォルトは`content_foo, content_bar, x-foo-bar`です。

### **rgw\_exit\_timeout\_secs**

プロセスを待機してから無条件に終了するまでの秒数。デフォルトは120です。

### **rgw\_get\_obj\_window\_size**

1つのオブジェクト要求のウィンドウサイズ(バイト単位)。デフォルトは16 << 20です。

### **rgw\_get\_obj\_max\_req\_size**

Ceph Storage Clusterに送信される1つのGET操作の最大要求サイズ。デフォルトは4 << 20です。

### **rgw\_relaxed\_s3\_bucket\_names**

USリージョンのバケットに対して、あいまいなS3バケット名を有効にします。デフォルトはfalseです。

### **rgw\_list\_buckets\_max\_chunk**

ユーザバケットを一覧にする際に1つの操作で取得するバケットの最大数。デフォルトは1000です。

### **rgw\_override\_bucket\_index\_max\_shards**

バケットインデックスオブジェクトのシャードの数を表します。0 (デフォルト)の設定は、シャーディングがないことを示します。バケットの一覧のコストが増加するため、大きすぎる値(たとえば、1000)を設定しないことをお勧めします。この変数は、自動的にradosgw-adminコマンドに適用されるよう、クライアントまたはグローバルセクションで設定する必要があります。

### **rgw\_curl\_wait\_timeout\_ms**

特定のcurl呼び出しのタイムアウト(ミリ秒)。デフォルトは1000です。

### **rgw\_copy\_obj\_progress**

コピー操作に時間がかかる場合に、オブジェクトの進行状況の出力を有効にします。デフォルトはtrueです。

### **rgw\_copy\_obj\_progress\_every\_bytes**

コピーの進行状況出力間の最大バイト数。デフォルトは1024 \* 1024です。

### **rgw\_admin\_entry**

管理要求URLのエントリポイント。デフォルトはadminです。

### **rgw\_content\_length\_compat**

CONTENT\_LENGTHとHTTP\_CONTENT\_LENGTHの両方が設定されたFCGI要求の互換処理を有効にします。デフォルトはfalseです。

### **rgw\_bucket\_quota\_ttl**

キャッシュされたクォータ情報を信頼する時間の量(秒単位)。このタイムアウトを過ぎると、クォータ情報はクラスタから再フェッチされます。デフォルトは600です。

### **rgw\_user\_quota\_bucket\_sync\_interval**

バケットクォータ情報が蓄積されてからクラスタと同期するまでの時間(秒単位)。この時間の間、他のObject Gatewayインスタンスは、このインスタンスに対する操作に関連したバケットクォータ統計情報の変更を確認しません。デフォルトは180です。

### **rgw\_user\_quota\_sync\_interval**

ユーザクォータ情報が蓄積されてからクラスタと同期するまでの時間(秒数)。この時間の間、他のObject Gatewayインスタンスは、このインスタンスに対する操作に関連したユーザクォータ統計情報の変更を確認しません。デフォルトは180です。

### **rgw\_bucket\_default\_quota\_max\_objects**

バケットあたりのオブジェクトのデフォルトの最大数。他のクォータが指定されていない場合、新しいユーザに対して設定され、既存のユーザには影響しません。この変数は、自動的に**radosgw-admin**コマンドに適用されるよう、クライアントまたはグローバルセクションで設定する必要があります。デフォルトは-1です。

### **rgw\_bucket\_default\_quota\_max\_size**

バケットあたりのデフォルトの最大容量(バイト単位)。他のクォータが指定されていない場合、新しいユーザに対して設定され、既存のユーザには影響しません。デフォルトは-1です。

### **rgw\_user\_default\_quota\_max\_objects**

ユーザのオブジェクトのデフォルトの最大数。これには、ユーザが所有するすべてのバケット内にあるすべてのオブジェクトが含まれます。他のクォータが指定されていない場合、新しいユーザに対して設定され、既存のユーザには影響しません。デフォルトは-1です。

### **rgw\_user\_default\_quota\_max\_size**

他のクォータが指定されていない場合に新しいユーザに対して設定されるユーザの最大クォータサイズ(バイト単位)。既存のユーザには影響しません。デフォルトは-1です。

### **rgw\_verify\_ssl**

要求の実行中にSSL証明書を検証します。デフォルトはtrueです。

### **rgw\_max\_chunk\_size**

1回の操作で読み込むデータチャンクの最大サイズ。値を4MB (4194304)に増やすと、大容量オブジェクトの処理時にパフォーマンスが向上します。デフォルトは128KB (131072)です。

## マルチサイト設定

### rgw\_zone

ゲートウェイインスタンスのゾーンの名前。ゾーンが設定されていない場合は、radosgw-admin zone default コマンドでクラスタ全体のデフォルト値を設定できます。

### rgw\_zonegroup

ゲートウェイインスタンスのゾーングループの名前。ゾーングループが設定されていない場合は、radosgw-admin zonegroup default コマンドでクラスタ全体のデフォルト値を設定できます。

### rgw\_realm

ゲートウェイインスタンスのレルムの名前。レルムが設定されていない場合は、radosgw-admin realm default コマンドでクラスタ全体のデフォルト値を設定できます。

### rgw\_run\_sync\_thread

レルム内に同期元となる他のゾーンがある場合は、データとメタデータの同期を処理するためのスレッドを生成します。デフォルトはtrueです。

### rgw\_data\_log\_window

データログエントリのウィンドウ(秒単位)。デフォルトは30です。

### rgw\_data\_log\_changes\_size

データ変更ログ用に保持するメモリ内エントリの数。デフォルトは1000です。

### rgw\_data\_log\_obj\_prefix

データログのオブジェクト名のプレフィックス。デフォルトは「data\_log」です。

### rgw\_data\_log\_num\_shards

データ変更ログを保持するシャード(オブジェクト)の数。デフォルトは128です。

### rgw\_md\_log\_max\_shards

メタデータログのシャードの最大数。デフォルトは64です。

## SWIFT設定

### rgw\_enforce\_swift\_acls

SwiftのACL (アクセス制御リスト)設定を適用します。デフォルトはtrueです。

### rgw\_swift\_token\_expiration

Swiftのトークンを期限切れにする時間(秒単位)。デフォルトは24 \* 3600です。

## rgw\_swift\_url

Ceph Object Gateway Swift APIのURL。

## rgw\_swift\_url\_prefix

「/v1」の部分の前に配置するSwift StorageURLのURLプレフィックス。これにより、同じホスト上で複数のゲートウェイインスタンスを実行できます。互換性のため、この設定変数を空に設定すると、デフォルトの「/swift」が使用されます。StorageURLをルートから開始するには、明示的なプレフィックス「/」を使用します。



### 警告

S3 APIが有効な場合、このオプションを「/」に設定しても機能しません。S3を無効にすると、マルチサイト設定でObject Gatewayを展開できなくなることに注意してください。

## rgw\_swift\_auth\_url

内部Swift認証が使用されていない場合にv1認証トークンを検証するためのデフォルトのURL。

## rgw\_swift\_auth\_entry

Swift認証URLのエントリポイント。デフォルトはauthです。

## rgw\_swift\_versioning\_enabled

OpenStack Object Storage APIのオブジェクトのバージョン管理を有効にします。これにより、クライアントは、バージョンを管理する必要があるコンテナにX-Versions-Location属性を設定できます。この属性では、アーカイブされたバージョンを保存するコンテナの名前を指定します。これは、アクセス制御の検証のため、バージョン管理されたコンテナと同じユーザが所有する必要があります。ACLは考慮「されません」。これらのコンテナは、S3のバージョン管理メカニズムではバージョン管理できません。デフォルトはfalseです。

## ログ設定

### rgw\_log\_nonexistent\_bucket

存在しないバケットに対する要求をObject Gatewayがログに記録できるようにします。デフォルトはfalseです。

### rgw\_log\_object\_name

オブジェクト名のログ書式。書式指定子の詳細については、マニュアルページ [man 1 date](#) を参照してください。デフォルトは%Y-%m-%d-%H-%i-%nです。

**rgw\_log\_object\_name\_utc**

ログに記録するオブジェクト名にUTC時刻を含めるかどうか。 false(デフォルト)に設定すると、ローカル時刻が使用されます。

**rgw\_usage\_max\_shards**

使用状況ログ用のシャードの最大数。デフォルトは32です。

**rgw\_usage\_max\_user\_shards**

1人のユーザの使用状況ログに使用するシャードの最大数。デフォルトは1です。

**rgw\_enable\_ops\_log**

Object Gatewayの正常な操作それぞれに対してログを有効にします。デフォルトは false です。

**rgw\_enable\_usage\_log**

使用状況ログを有効にします。デフォルトは false です。

**rgw\_ops\_log\_rados**

操作ログをCeph Storage Clusterバックエンドに書き込むかどうか。デフォルトは true です。

**rgw\_ops\_log\_socket\_path**

操作ログを書き込むためのUnixドメインソケット。

**rgw\_ops\_log\_data\_backlog**

Unixドメインソケットに書き込まれる操作ログのデータバックログの最大データサイズ。デフォルトは5<<20です。

**rgw\_usage\_log\_flush\_threshold**

同期的にフラッシュするまでの、使用状況ログ内のマージされたダーティエントリの数。デフォルトは1024です。

**rgw\_usage\_log\_tick\_interval**

保留中の使用状況ログデータを「n」秒ごとにフラッシュします。デフォルトは30です。

**rgw\_log\_http\_headers**

ログエントリに含めるHTTPヘッダのカンマ区切りリスト。ヘッダ名では大文字と小文字は区別されず、各単語を下線で区切った完全なヘッダ名を使用します。たとえば、「http\_x\_forwarded\_for」「http\_x\_special\_k」のようにします。

**rgw\_intent\_log\_object\_name**

インテントログオブジェクト名のログ書式。書式指定子の詳細については、マニュアルページ man 1 date を参照してください。デフォルトは「%Y-%m-%d-%i-%n」です。

### **rgw\_intent\_log\_object\_name\_utc**

インテントログオブジェクト名にUTC時刻を含めるかどうか。 `false`(デフォルト)に設定すると、ローカル時刻が使用されます。

## **KEYSTONE設定**

### **rgw\_keystone\_url**

KeystoneサーバのURL。

### **rgw\_keystone\_api\_version**

Keystoneサーバと通信するために使用するOpenStack Identity APIのバージョン(2または3)。デフォルトは2です。

### **rgw\_keystone\_admin\_domain**

OpenStack Identity API v3を使用する場合に管理者特権を持つOpenStackドメインの名前。

### **rgw\_keystone\_admin\_project**

OpenStack Identity API v3を使用する場合に管理者特権を持つOpenStackプロジェクトの名前。設定されていない場合は、代わりに **rgw\_keystone\_admin\_tenant** の値が使用されます。

### **rgw\_keystone\_admin\_token**

Keystone管理者トークン(共有シークレット)。Object Gatewayでは、管理者トークンを使用した認証は、管理者資格情報を使用した認証よりも優先されます(オプション `rgw_keystone_admin_user`、`rgw_keystone_admin_password`、`rgw_keystone_admin_tenant`、`rgw_keystone_admin_project`、および `rgw_keystone_admin_domain`)。管理者トークン機能は非推奨と見なされています。

### **rgw\_keystone\_admin\_tenant**

OpenStack Identity API v2を使用する場合に管理者特権を持つOpenStackテナント(サービステナント)の名前。

### **rgw\_keystone\_admin\_user**

OpenStack Identity API v2を使用する場合にKeystone認証用の管理者特権を持つOpenStackユーザ(サービスユーザ)の名前。

### **rgw\_keystone\_admin\_password**

OpenStack Identity API v2を使用する場合のOpenStack管理者ユーザのパスワード。

### **rgw\_keystone\_accepted\_roles**

要求を実行するために必要な役割。デフォルトは「Member, admin」です。

### **rgw\_keystone\_token\_cache\_size**

各Keystoneトークンキャッシュ内のエントリの最大数。デフォルトは10000です。

### **rgw\_keystone\_revocation\_interval**

トークンの失効を確認する間隔の秒数。デフォルトは15 \* 60です。

### **rgw\_keystone\_verify\_ssl**

Keystoneへのトークン要求の実行中にSSL証明書を検証します。デフォルトはtrueです。

## 28.5.1.1 追加の注意事項

### **rgw\_dns\_name**

クライアントがvhost形式のバケットを使用できるようにします。

vhost形式のアクセスは、bucketname.s3-endpoint/object-pathを使用することを意味します。これに対してpath形式のアクセスはs3-endpoint/bucket/objectを使用します。

rgw dns nameが設定されている場合、S3クライアントがrgw dns nameで指定されるエンドポイントに直接要求するように設定されているか、確認してください。

## 28.5.2 HTTPフロントエンドの設定

### 28.5.2.1 Beast

#### **port、ssl\_port**

IPv4およびIPv6のリスンポート番号。複数のポート番号を指定できます。

```
port=80 port=8000 ssl_port=8080
```

デフォルトは80です。

#### **endpoint、ssl\_endpoint**

「address[:port]」の形式のリスンアドレス。アドレスは、ドット区切りの10進数形式のIPv4アドレス文字列、または角括弧で囲んだ16進数形式のIPv6アドレスです。IPv6エンドポイントを指定すると、IPv6のみがリスンされます。オプションのポート番号は、endpointの場合は80、ssl\_endpointの場合は443にデフォルトで設定されます。複数のアドレスを指定できます。

```
endpoint=[::1] endpoint=192.168.0.100:8000 ssl_endpoint=192.168.0.100:8080
```

### ssl\_private\_key

SSLが有効なエンドポイントに対して使用する秘密鍵のパス(オプション)。指定されていない場合、ssl\_certificateファイルが秘密鍵として使用されます。

### tcp\_nodelay

指定されている場合、ソケットオプションにより、接続時にNagleのアルゴリズムが無効化されます。つまり、パケットは、バッファがいっぱいになるかタイムアウトになるまで待つのではなく、できるだけ早く送信されます。

「1」は、すべてのソケットに対してNagleのアルゴリズムを無効にします。

「0」は、Nagleのアルゴリズムを有効なままにします(デフォルト)。

#### 例 28.1: BEASTの設定例

```
cephuser@adm > ceph config set rgw.myrealm.myzone.ses-min1.kwwazo \  
rgw_frontends beast port=8000 ssl_port=443 \  
ssl_certificate=/etc/ssl/ssl.crt \  
error_log_file=/var/log/radosgw/beast.error.log
```

## 28.5.2.2 CivetWeb

### ポート

リスンするポート番号。SSLが有効なポートには、サフィックス「s」を追加します(たとえば、「443s」)。特定のIPv4またはIPv6アドレスをバインドするには、「address:port」の形式を使用します。複数のエンドポイントを指定するには、各エンドポイントを「+」で結合するか、複数のオプションを指定します。

```
port=127.0.0.1:8000+443s  
port=8000 port=443s
```

デフォルトは7480です。

### num\_threads

受信HTTP接続を処理するためにCivetwebによって生成されるスレッドの数。これは実質的に、フロントエンドが処理できる同時接続数を制限します。

デフォルトは、rgw\_thread\_pool\_sizeオプションで指定されている値です。

### request\_timeout\_ms

Civetwebが他の受信データを待機してから中止するまでの時間(ミリ秒単位)。

デフォルトは30,000ミリ秒です。

### **access\_log\_file**

アクセスログファイルのパス。フルパス、または現在の作業ディレクトリを基準とした相対パスのいずれかを指定できます。指定されていない場合(デフォルト)、アクセスはログに記録されません。

### **error\_log\_file**

エラーログファイルのパス。フルパス、または現在の作業ディレクトリを基準とした相対パスのいずれかを指定できます。指定されていない場合(デフォルト)、エラーはログに記録されません。

#### **例 28.2: /etc/ceph/ceph.confのCIVETWEB設定の例**

```
cephuser@adm > ceph config set rgw.myrealm.myzone.ses-min2.ingabw \
rgw_frontends civetweb port=8000+443s request_timeout_ms=30000 \
error_log_file=/var/log/radosgw/civetweb.error.log
```

## 28.5.2.3 共通オプション

### **ssl\_certificate**

SSLが有効なエンドポイントに対して使用するSSL証明書ファイルのパス。

### **prefix**

すべての要求のURLに挿入するプレフィックス文字列。たとえば、Swift専用のフロントエンドでは、URLプレフィックス/swiftを指定できます。

## 29 Ceph Managerモジュール

Ceph Managerのアーキテクチャ(概要については、『導入ガイド』、第1章「SESとCeph」、1.2.3項「Cephのノードとデーモン」を参照)では、「ダッシュボード」(パートI「Cephダッシュボード」を参照)、「prometheus」(第16章「監視とアラート」を参照)、「バランサ」などの「モジュール」によって機能を拡張できます。

使用可能なすべてのモジュールを一覧にするには、次のコマンドを実行します。

```
cephuser@adm > ceph mgr module ls
{
  "enabled_modules": [
    "restful",
    "status"
  ],
  "disabled_modules": [
    "dashboard"
  ]
}
```

特定のモジュールを有効または無効にするには、次のコマンドを実行します。

```
cephuser@adm > ceph mgr module enable MODULE-NAME
```

以下に例を示します。

```
cephuser@adm > ceph mgr module disable dashboard
```

有効なモジュールが提供するサービスを一覧にするには、次のコマンドを実行します。

```
cephuser@adm > ceph mgr services
{
  "dashboard": "http://myserver.com:7789/",
  "restful": "https://myserver.com:8789/"
}
```

### 29.1 バランサ

バランサモジュールは、OSD間でのPG (配置グループ)の分散を最適化して展開のバランスを確保します。このモジュールはデフォルトで有効になっていますが、非アクティブです。サポートされているモードは、crush-compatとupmapの2つです。



#### ヒント: バランサの現在のステータスと設定

バランサの現在のステータスと設定に関する情報を表示するには、次のコマンドを実行します。

```
cephuser@adm > ceph balancer status
```

### 29.1.1 「crush-compat」モード

「crush-compat」モードでは、バランサは、OSDのreweight-setsを調整して、データが適切に分散されるようにします。これは、OSDの間でPGを移動するので、PGの配置が正しくなくなり、一時的にクラスタ状態がHEALTH\_WARNになります。



#### ヒント: モードのアクティベーション

「crush-compat」はデフォルトのモードですが、SUSEでは、これを明示的に有効にすることをお勧めします。

```
cephuser@adm > ceph balancer mode crush-compat
```

### 29.1.2 データバランシングの計画と実行

バランサモジュールを使用して、データバランシングの計画を作成できます。その後、計画を手動で実行することも、バランサでPGのバランスを継続的に調整することもできます。

バランサを手動モードまたは自動モードのどちらで実行するか判断は、現在のデータの不均衡、クラスタサイズ、PG数、I/Oアクティビティなど、さまざまな要因に依存します。初期計画を作成し、クラスタのI/O負荷が低いときに実行することをお勧めします。この理由は、初期の不均衡はかなり大きくなる可能性があるため、クライアントへの影響を低く抑えることをお勧めします。初期の手動実行後、自動モードを有効にして、通常のI/O負荷でリバランストラフィックを監視することを検討します。PG分散の向上は、バランサが原因で発生するリバランストラフィックに対して比較検討する必要があります。



#### ヒント: PG (配置グループ)の移動可能な割合

バランスプロセス中に、バランサモジュールは、PGの設定可能な部分のみが移動されるようPGの動きを制限します。デフォルトは5%ですが、次のコマンドを実行することで、この割合をたとえば9%に調整できます。

```
cephuser@adm > ceph config set mgr target_max_misplaced_ratio .09
```

バランス計画を作成および実行するには、次の手順に従います。

1. クラスタの現在のスコアを確認します。

```
cephuser@adm > ceph balancer eval
```

2. 計画を作成します。たとえば、「great\_plan」などです。

```
cephuser@adm > ceph balancer optimize great_plan
```

3. 「great\_plan」がどのような変化をもたらすかを確認します。

```
cephuser@adm > ceph balancer show great_plan
```

4. 「great\_plan」を適用することを決定した場合は、クラスタの潜在的なスコアを確認します。

```
cephuser@adm > ceph balancer eval great_plan
```

5. 「great\_plan」を1回だけ実行します。

```
cephuser@adm > ceph balancer execute great_plan
```

6. **ceph -s** コマンドを使用して、クラスタのバランスを確認します。結果に問題がなければ、自動バランスを有効にします。

```
cephuser@adm > ceph balancer on
```

後で自動バランスを無効にする場合は、次のコマンドを実行します。

```
cephuser@adm > ceph balancer off
```



### ヒント: 初期計画なしの自動バランス

初期計画を実行しないで自動バランスを有効にすることができます。このような場合、配置グループのリバランスが長時間実行される可能性があることを予期してください。

## 29.2 テレメトリモジュールの有効化

テレメトリプラグインは、プラグインが実行されているクラスタに関するCephプロジェクトの匿名データを送信します。

この(オプトイン)コンポーネントには、クラスタの展開方法、Cephのバージョン、ホストの分散、およびプロジェクトでCephの使用方法についての理解を深めるのに役立つ他のパラメータに関するカウンタと統計情報が含まれています。プール名、オブジェクト名、オブジェクトの内容、ホスト名などの機密データは含まれません。

テレメトリモジュールの目的は、開発者に自動フィードバックループを提供し、これによって、導入率の定量化や追跡を行ったり、望ましくない結果を避けるために設定時により適切に説明または検証する必要がある事項を指摘したりできるようにすることです。



## 注記

テレメトリモジュールを使用するには、Ceph Managerノードに、HTTPSでアップストリームサーバにデータをプッシュする機能が必要です。企業のファイアウォールでこのアクションが許可されるようにします。

1. テレメトリモジュールを有効にするには、次のコマンドを実行します。

```
cephuser@adm > ceph mgr module enable telemetry
```



## 注記

このコマンドでは、ローカルでのみデータを表示できます。Cephコミュニティとデータを共有することはできません。

2. テレメトリモジュールがデータの共有を開始できるようにするには、次のコマンドを実行します。

```
cephuser@adm > ceph telemetry on
```

3. テレメトリデータ共有を無効にするには、次のコマンドを実行します。

```
cephuser@adm > ceph telemetry off
```

4. 印刷可能なJSONレポートを生成するには、次のコマンドを実行します。

```
cephuser@adm > ceph telemetry show
```

5. 連絡先と説明をレポートに追加するには、次のコマンドを実行します。

```
cephuser@adm > ceph config set mgr mgr/telemetry/contact John Doe  
john.doe@example.com  
cephuser@adm > ceph config set mgr mgr/telemetry/description 'My first Ceph cluster'
```

6. このモジュールは、デフォルトでは24時間ごとに新しいレポートを作成して送信します。この間隔を調整するには、次のコマンドを実行します。

```
cephuser@adm > ceph config set mgr mgr/telemetry/interval HOURS
```

## 30 cephxを使用した認証

クライアントを識別して中間者攻撃を防御するため、Cephはcephx認証システムを提供します。このコンテキストの「クライアント」とは、人間のユーザ(管理者ユーザなど)またはCeph関連サービス/デーモン(たとえば、OSD、Monitor、Object Gateway)のどちらかです。



### 注記

cephxプロトコルは、TLS/SSLと異なり、転送中のデータ暗号化には対応していません。

### 30.1 認証アーキテクチャ

cephxは、認証に共有秘密鍵を使用します。つまり、クライアントとCeph Monitorの両方がクライアントの秘密鍵のコピーを持ちます。この認証プロトコルでは、実際に鍵を公開することなく、両者が鍵のコピーを持っていることをお互いに証明できます。これによって相互認証が提供されます。つまり、クラスタはユーザが秘密鍵を所有していることを信頼し、ユーザはクラスタが秘密鍵を持っていることを信頼します。

Cephの重要なスケーラビリティ機能は、Ceph Object Storeへの中央インタフェースの必要がないことです。つまり、CephクライアントはOSDと直接対話できます。データを保護するため、Cephはcephx認証システムを備えており、これによってCephクライアントを認証します。

各Monitorでクライアントを認証して鍵を配布することができ、この結果cephxの使用時にSPOF (single point of failure)やボトルネックがなくなります。Monitorは、Cephサービスを利用する際に使用するセッションキーが含まれる認証データ構造を返します。このセッションキー自体がクライアントの永続的な秘密鍵で暗号化されているため、そのクライアントのみがCeph Monitorにサービスを要求できます。その後、クライアントはセッションキーを使用して必要なサービスをMonitorに要求し、Monitorは、データを実際に処理するOSDに対してクライアントを認証するチケットをクライアントに提供します。CephのMonitorとOSDは秘密を共有するので、クライアントは、Monitorが提供するチケットをクラスタ内の任意のOSDまたはメタデータサーバで使用できます。cephxチケットは期限切れになるため、攻撃者が不正に入手した期限切れのチケットやセッションキーを使用することはできません。

cephxを使用するには、まず管理者がクライアント/ユーザをセットアップする必要があります。次の図では、`client.admin`ユーザがコマンドラインから`ceph auth get-or-create-key`を呼び出して、ユーザ名と秘密鍵を生成します。Cephの`auth`サブシステムは、ユーザ名と鍵を生成してMonitorにコピーを保存し、ユーザの秘密を`client.admin`ユーザに戻します。つまり、クライアントとMonitorが秘密鍵を共有します。

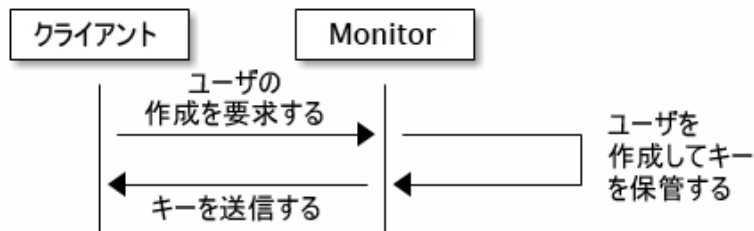


図 30.1: 基本的なcephx認証

Monitorで認証する場合、クライアントはMonitorにユーザ名を渡します。Monitorは、セッションキーを生成して、それをユーザ名に関連付けられている秘密鍵で暗号化し、暗号化されたチケットをクライアントに戻します。その後、クライアントは共有秘密鍵でデータを復号化して、セッションキーを取得します。このセッションキーは、現在のセッション中、ユーザを識別します。次にクライアントは、このセッションキーによって署名された、ユーザに関連するチケットを要求します。Monitorはチケットを生成し、ユーザの秘密鍵で暗号化してクライアントに戻します。クライアントはチケットを復号化し、そのチケットを使用して、クラスタ全体のOSDとメタデータサーバに対する要求に署名します。

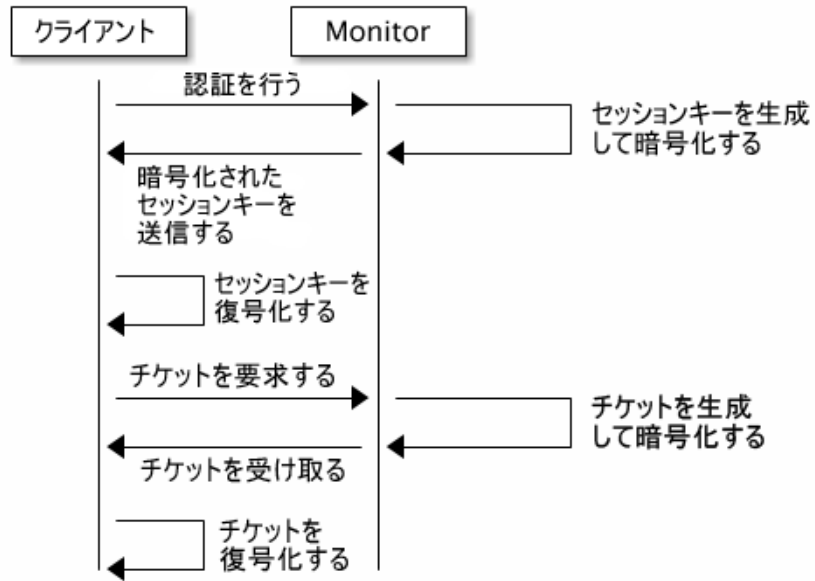


図 30.2: cephx認証

cephx プロトコルは、クライアントマシンとCephサーバとの間で進行中の通信を認証します。初期認証後にクライアントとサーバとの間で送信される各メッセージは、Monitor、OSD、およびメタデータサーバがその共有秘密で検証できるチケットを使用して署名されます。

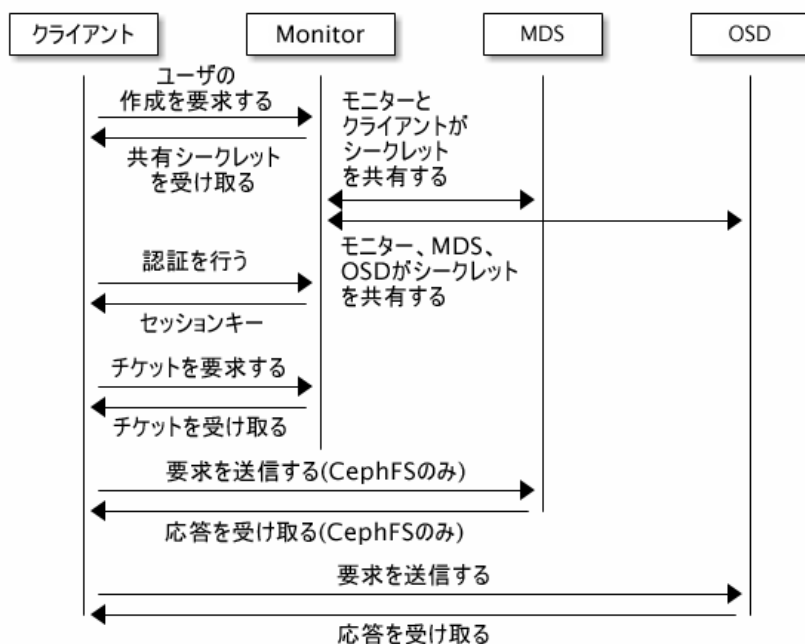


図 30.3: cephx認証 - MDSとOSD

## ！ 重要

この認証は、CephクライアントとCephクラスタホストとの間の保護を提供します。Cephクライアントより先は認証されません。ユーザがリモートホストからCephクライアントにアクセスする場合、ユーザのホストとクライアントホストとの間の接続にはCeph認証は適用されません。

## 30.2 キー管理

このセクションでは、Cephクライアントユーザ、およびCeph Storage Clusterでの認証と権限付与について説明します。「ユーザ」とは、個人、またはCephクライアントを使用してCeph Storage Clusterデーモンと対話するシステムアクタ(アプリケーションなど)のいずれかです。

認証と権限付与を有効にして(デフォルトで有効)Cephを実行している場合、ユーザ名と、指定したユーザの秘密鍵が含まれるキーリングを指定する必要があります(通常はコマンドラインを使用)。ユーザ名を指定しない場合、`client.admin`がデフォルトのユーザ名として使用さ

れます。キーリングを指定しない場合、Ceph設定ファイルのキーリング設定を使用してキーリングが検索されます。たとえば、ユーザ名またはキーリングを指定せずに`ceph health`コマンドを実行すると、Cephはコマンドを次のように解釈します。

```
cephuser@adm > ceph -n client.admin --keyring=/etc/ceph/ceph.client.admin.keyring health
```

または、`CEPH_ARGS`環境変数を使用して、ユーザ名と秘密の再入力を避けることができます。

## 30.2.1 予備知識

Cephクライアントのタイプ(たとえば、Block Device、Object Storage、File System、ネイティブAPI)に関係なく、Cephはすべてのデータを「プール」内にオブジェクトとして保存します。Cephユーザがデータを読み書きするには、プールに対するアクセスが必要です。さらに、Cephの管理コマンドを使用するための実行許可も必要です。Cephのユーザ管理を理解するには、次の概念が役立ちます。

### 30.2.1.1 ユーザ

ユーザとは、個人またはシステムアクタ(アプリケーションなど)のいずれかです。ユーザを作成することによって、誰が(または何が)Ceph Storage Cluster、そのプール、およびプール内のデータにアクセスできるかを制御できます。

Cephでは、ユーザの「タイプ」を使用します。ユーザ管理の目的では、タイプは常に`client`です。Cephは、ユーザタイプとユーザIDで構成される、ピリオド(.)区切り形式でユーザを識別します。たとえば、`TYPE.ID`、`client.admin`、`client.user1`などです。ユーザタイプを使用する理由は、Ceph Monitor、OSD、およびメタデータサーバもcephxプロトコルを使用しますが、これらはクライアントではないためです。ユーザタイプを区別すると、クライアントユーザと他のユーザを容易に区別でき、アクセス制御、ユーザのモニタリング、および追跡可能性が効率化されます。

場合によっては、Cephのユーザタイプがわかりにくいことがあります。Cephのコマンドラインでは、コマンドラインの使用法に応じて、タイプを指定しても指定しなくてもユーザを指定できるためです。`--user`または`--id`を指定する場合は、タイプを省略できます。そのため、`client.user1`を単に`user1`として入力できます。`--name`または`-n`を指定する場合は、`client.user1`のようにタイプと名前を指定する必要があります。ベストプラクティスとして、可能な限りタイプと名前を使用することをお勧めします。



## 注記

Ceph Storage Clusterユーザは、Ceph Object StorageユーザやCeph File Systemユーザと同じではありません。Ceph Object Gatewayは、ゲートウェイデーモンとStorage Clusterとの間の通信にCeph Storage Clusterユーザを使用しますが、ゲートウェイにはエンドユーザ向けの独自のユーザ管理機能があります。Ceph File SystemはPOSIXセマンティクスを使用します。そこに関連付けられているユーザスペースは、Ceph Storage Clusterユーザと同じではありません。

### 30.2.1.2 権限付与とケーパビリティ

Cephでは、認証ユーザにMonitor、OSD、およびメタデータサーバの機能を実行する権限を付与することを説明するために、「ケーパビリティ(cap)」という用語を使用します。ケーパビリティはプールまたはプールネームスペース内のデータへのアクセスを制限することもできます。ユーザの作成または更新時にCeph管理者ユーザがユーザのケーパビリティを設定します。

ケーパビリティの構文は次の形式に従います。

```
daemon-type 'allow capability' [...]
```

次に、各サービスタイプのケーパビリティのリストを示します。

#### Monitorのケーパビリティ

r、w、x、およびallow profile capを含めます。

```
mon 'allow rwx'
mon 'allow profile osd'
```

#### OSDのケーパビリティ

r、w、x、class-read、class-write、およびprofile osdを含めます。さらに、プールとネームスペースも設定できます。

```
osd 'allow capability' [pool=poolname] [namespace=namespace-name]
```

#### MDSのケーパビリティ

allowまたは空白のみが必要です。

```
mds 'allow'
```

次のエントリで各ケーパビリティについて説明します。

## **allow**

デーモンのアクセス設定の前に付けます。MDSの場合にのみ、暗黙的に rw を示します。

## **r**

ユーザに読み込みアクセスを付与します。CRUSHマップを取得するためにMonitorが必要です。

## **w**

ユーザにオブジェクトへの書き込みアクセスを付与します。

## **x**

クラスメソッドを呼び出すためのケーパビリティ(読み込みと書き込みの両方)、およびMonitorで auth 操作を実行するためのケーパビリティをユーザに付与します。

## **class-read**

クラス読み込みメソッドを呼び出すためのケーパビリティをユーザに付与します。xのサブセットです。

## **class-write**

クラス書き込みメソッドを呼び出すためのケーパビリティをユーザに付与します。xのサブセットです。

## **\***

特定のデーモン/プールに対する読み込み、書き込み、および実行の許可と、管理コマンドの実行機能をユーザに付与します。

## **profile osd**

他のOSDまたはMonitorにOSDとして接続するための許可をユーザに付与します。OSDがレプリケーションハートビートトラフィックと状態レポートを処理できるようにするために、OSDに付与されます。

## **profile mds**

他のMDSまたはMonitorにMDSとして接続するための許可をユーザに付与します。

## **profile bootstrap-osd**

OSDをブートするための許可をユーザに付与します。展開ツールに対して委任され、OSDのブート時にキーを追加するための許可を展開ツールに付与します。

## **profile bootstrap-mds**

メタデータサーバをブートするための許可をユーザに付与します。展開ツールに対して委任され、メタデータサーバのブート時にキーを追加するための許可を展開ツールに付与します。

### 30.2.1.3 プール

プールは、ユーザがデータを保存する論理パーティションです。Cephの展開環境では、一般的に、類似するデータタイプ用の論理パーティションとしてプールを作成します。たとえば、CephをOpenStackのバックエンドとして展開する場合、標準的な展開には、ボリューム、イメージ、バックアップ、および仮想マシン用のプールと、`client.glance`や`client.cinder`などのユーザが存在します。

## 30.2.2 ユーザの管理

ユーザ管理機能により、Cephクラスタ管理者は、Cephクラスタ内でユーザを直接作成、更新、および削除できます。

Cephクラスタ内でユーザを作成または削除する場合、キーをクライアントに配布して、キーリングに追加できるようにする必要があります。詳細については[30.2.3項「キーリングの管理」](#)を参照してください。

### 30.2.2.1 ユーザの一覧

クラスタ内のユーザを一覧にするには、次のコマンドを実行します。

```
cephuser@adm > ceph auth list
```

クラスタ内のすべてのユーザが一覧にされます。たとえば、2ノードのクラスタでは、`ceph auth list`の出力は次のようになります。

```
installed auth entries:

osd.0
  key: AQCvCbTToC6MDhAATtuT70Sl+DymPCfDSsyV4w==
  caps: [mon] allow profile osd
  caps: [osd] allow *
osd.1
  key: AQC4CbTTCFJBChAAVq5spj0ff4eHZICxIOVZeA==
  caps: [mon] allow profile osd
  caps: [osd] allow *
client.admin
  key: AQBHCbtT6APDHhAA5W00cBchwKQjh3dkKsyPjw==
  caps: [mds] allow
  caps: [mon] allow *
  caps: [osd] allow *
client.bootstrap-mds
  key: AQBICbtT0K9uGBAAdbE5zcIGHZL3T/u2g6EBww==
  caps: [mon] allow profile bootstrap-mds
```

```
client.bootstrap-osd
  key: AQBHCbtT4Gxq0RAADE5u7RkpCN/oo4e5W0uBtw==
  caps: [mon] allow profile bootstrap-osd
```



### 注記: TYPE.ID表記

ユーザのTYPE.ID表記は、osd.0の場合、タイプがosdのユーザを指定し、そのIDが0になるように適用されることに注意してください。client.adminの場合は、タイプがclientのユーザで、そのIDはadminです。さらに、各エントリにkey: valueのエントリと1つ以上のcaps:エントリもあることに注意してください。

-o filename オプションと **ceph auth list** を使用して、出力をファイルに保存できます。

#### 30.2.2.2 ユーザに関する情報の取得

特定のユーザ、キー、およびケーパビリティを取得するには、次のコマンドを実行します。

```
cephuser@adm > ceph auth get TYPE.ID
```

例:

```
cephuser@adm > ceph auth get client.admin
exported keyring for client.admin
[client.admin]
  key = AQA19uZUqIwkHxAAFuUwvq0eJD4S173oFRxe0g==
  caps mds = "allow"
  caps mon = "allow *"
  caps osd = "allow *"
```

開発者の場合は次のコマンドを実行することもできます。

```
cephuser@adm > ceph auth export TYPE.ID
```

**auth export** コマンドは **auth get** と同じですが、内部の認証IDも出力します。

#### 30.2.2.3 ユーザの追加

ユーザを追加すると、ユーザの作成に使用するコマンドで指定したユーザ名(TYPE.ID)、秘密鍵、およびケーパビリティが作成されます。

ユーザのキーにより、ユーザはCeph Storage Clusterで認証できます。ユーザのケーパビリティは、Ceph Monitor (mon)、Ceph OSD (osd)、またはCeph Metadata Server (mds) に対する読み込み、書き込み、および実行の各権限をユーザに付与します。

次のように、ユーザを追加する場合、いくつかのコマンドを利用できます。

### **ceph auth add**

ユーザを追加する場合の標準の方法です。ユーザを作成してキーを生成し、指定したケーパビリティを追加します。

### **ceph auth get-or-create**

このコマンドはユーザ名(カッコ内)とキーが含まれるキーファイルフォーマットを返すため、多くの場合、ユーザを作成する場合に最も便利な方法です。ユーザがすでに存在する場合は、単にユーザ名とキーをキーファイルフォーマットで返します。 `-o filename` オプションを使用して、出力をファイルに保存できます。

### **ceph auth get-or-create-key**

ユーザを作成して、そのユーザのキー(のみ)を返す場合に便利です。キーのみが必要なクライアント(たとえば、`libvirt`)で役立ちます。ユーザがすでに存在する場合は、単にキーを返します。 `-o filename` オプションを使用して、出力をファイルに保存できます。

クライアントユーザを作成する際に、ケーパビリティを持たないユーザを作成できます。ケーパビリティを持たないユーザは、認証はできますが、それ以上の操作は実行できません。このようなクライアントはMonitorからクラスタマップを取得できません。ただし、ケーパビリティの追加を延期する場合、後で `ceph auth caps` コマンドを使用して、ケーパビリティを持たないユーザを作成できます。

通常のユーザは、少なくともCeph Monitorに対する読み込みケーパビリティと、Ceph OSDに対する読み込みおよび書き込みのケーパビリティを持ちます。さらに、ほとんどの場合、ユーザのOSD許可は特定のプールへのアクセスに制限されます。

```
cephuser@adm > ceph auth add client.john mon 'allow r' osd \
'allow rw pool=liverpool'
cephuser@adm > ceph auth get-or-create client.paul mon 'allow r' osd \
'allow rw pool=liverpool'
cephuser@adm > ceph auth get-or-create client.george mon 'allow r' osd \
'allow rw pool=liverpool' -o george.keyring
cephuser@adm > ceph auth get-or-create-key client.ringo mon 'allow r' osd \
'allow rw pool=liverpool' -o ringo.key
```

## 重要

OSDに対するケーパビリティをユーザに提供しながら、特定のプールにアクセスを制限「しない」場合、そのユーザはクラスタ内の「すべて」のプールへのアクセスを持ちます。

### 30.2.2.4 ユーザのケーパビリティの変更

**ceph auth caps** コマンドを使用して、ユーザを指定してそのユーザのケーパビリティを変更できます。新しいケーパビリティを設定すると、現在のケーパビリティは上書きされます。現在のケーパビリティを表示するには、**ceph auth get USERTYPE.USERID**。ケーパビリティを追加するには、次の形式を使用する際に既存のケーパビリティを指定する必要もあります。

```
cephuser@adm > ceph auth caps USERTYPE.USERID daemon 'allow [r|w|x|*|...] \
[pool=pool-name] [namespace=namespace-name]' [daemon 'allow [r|w|x|*|...] \
[pool=pool-name] [namespace=namespace-name]']
```

例:

```
cephuser@adm > ceph auth get client.john
cephuser@adm > ceph auth caps client.john mon 'allow r' osd 'allow rw pool=prague'
cephuser@adm > ceph auth caps client.paul mon 'allow rw' osd 'allow r pool=prague'
cephuser@adm > ceph auth caps client.brian-manager mon 'allow *' osd 'allow *'
```

ケーパビリティを削除する場合、ケーパビリティをリセットできます。すでに設定されている、特定のデーモンへのアクセスをユーザに付与しない場合、空の文字列を指定します。

```
cephuser@adm > ceph auth caps client.ringo mon ' ' osd ' '
```

### 30.2.2.5 ユーザの削除

ユーザを削除するには、**ceph auth del**を使用します。

```
cephuser@adm > ceph auth del TYPE.ID
```

ここで、TYPEはclient、osd、mon、またはmdsのいずれかで、IDはデーモンのユーザ名またはIDです。

存在しなくなったプール専用の許可を持つユーザを作成した場合は、それらのユーザの削除も検討することをお勧めします。

### 30.2.2.6 ユーザのキーの出力

ユーザの認証キーを標準出力に出力するには、次のコマンドを実行します。

```
cephuser@adm > ceph auth print-key TYPE.ID
```

ここで、TYPEはclient、osd、mon、またはmdsのいずれかで、IDはデーモンのユーザ名またはIDです。

ユーザのキーを出力すると、クライアントソフトウェアにユーザのキー(libvirtなど)を入力する必要がある場合に便利です。次に例を示します。

```
root # mount -t ceph host:/ mount_point \  
-o name=client.user,secret=`ceph auth print-key client.user`
```

### 30.2.2.7 ユーザのインポート

1人以上のユーザをインポートするには、**`ceph auth import`**を使用し、キーリングを指定します。

```
cephuser@adm > ceph auth import -i /etc/ceph/ceph.keyring
```



#### 注記

Ceph Storage Clusterは、新しいユーザ、キー、およびキーパビリティを追加し、既存のユーザ、キー、およびキーパビリティを更新します。

## 30.2.3 キーリングの管理

CephクライアントでCephにアクセスすると、クライアントはローカルキーリングを検索します。Cephにより、デフォルトで次の4つのキーリング名を使用してキーリング設定が事前設定されるため、デフォルトを上書きする場合を除き、ユーザがCeph設定ファイルでキーリングを設定する必要はありません。

```
/etc/ceph/cluster.name.keyring  
/etc/ceph/cluster.keyring  
/etc/ceph/keyring  
/etc/ceph/keyring.bin
```

`cluster`メタ変数は、Ceph設定ファイルの名前で定義されているCephクラスタ名です。`ceph.conf`は、クラスタ名が`ceph`であるため、`ceph.keyring`になることを意味します。`name`メタ変数は、ユーザタイプとユーザID(たとえば、`client.admin`)であるため、`ceph.client.admin.keyring`になります。

ユーザ(たとえば、`client.ringo`)を変更した後、キーを取得してCephクライアント上のキーリングに追加し、ユーザがCeph Storage Clusterにアクセスできるようにする必要があります。

Ceph Storage Cluster内でユーザを直接一覧、取得、追加、変更、および削除する方法の詳細については、[30.2項「キー管理」](#)を参照してください。ただし、Cephには、Cephクライアントからキーリングを管理できる**`ceph-authtool`**ユーティリティも用意されています。

### 30.2.3.1 キーリングの作成

30.2項「キー管理」の手順を使用してユーザを作成した場合、ユーザのキーをCephクライアントに提供し、クライアントが指定のユーザのキーを取得してCeph Storage Clusterで認証できるようにする必要があります。Cephクライアントは、キーリングにアクセスしてユーザ名を検索し、ユーザのキーを取得します。

```
cephuser@adm > ceph-authtool --create-keyring /path/to/keyring
```

複数のユーザが含まれるキーリングを作成する場合は、キーリングのファイル名にクラスタ名(たとえば、`cluster.keyring`)を使用して、`/etc/ceph`ディレクトリに保存することをお勧めします。これにより、Ceph設定ファイルのローカルコピーで指定しなくても、キーリング設定のデフォルト設定でこのファイル名が選択されます。たとえば、次のコマンドを実行して、`ceph.keyring`を作成します。

```
cephuser@adm > ceph-authtool -C /etc/ceph/ceph.keyring
```

1人のユーザが含まれるキーリングを作成する場合は、クラスタ名、ユーザタイプ、およびユーザ名を指定して、`/etc/ceph`ディレクトリに保存することをお勧めします。たとえば、ユーザが`client.admin`の場合は、`ceph.client.admin.keyring`とします。

### 30.2.3.2 キーリングにユーザを追加

Ceph Storage Clusterにユーザを追加する場合(30.2.2.3項「ユーザの追加」を参照)、ユーザ、キー、および権限を取得して、そのユーザをキーリングに保存できます。

1つのキーリングにつき1人のユーザのみを使用する場合は、**`ceph auth get`**コマンドを`-o`オプションと共に使用して、出力をキーリングファイルフォーマットで保存します。たとえば、`client.admin`ユーザのキーリングを作成するには、次のコマンドを実行します。

```
cephuser@adm > ceph auth get client.admin -o /etc/ceph/ceph.client.admin.keyring
```

キーリングにユーザをインポートする場合は、**`ceph-authtool`**を使用して、インポート先のキーリングとインポート元のキーリングを指定します。

```
cephuser@adm > ceph-authtool /etc/ceph/ceph.keyring \  
--import-keyring /etc/ceph/ceph.client.admin.keyring
```

## ！ 重要

キーリングがセキュリティ侵害を受けた場合は、`/etc/ceph`ディレクトリからキーを削除し、30.2.3.1項「キーリングの作成」と同じ手順を使用して新しいキーを再作成してください。

### 30.2.3.3 ユーザの作成

Cephでは、Ceph Storage Cluster内でユーザを直接作成するための`ceph auth add`コマンドが提供されています。ただし、Cephクライアントのキーリング上でユーザ、キー、およびケーパビリティを直接作成することもできます。その後、ユーザをCeph Storage Clusterにインポートできます。

```
cephuser@adm > ceph-authtool -n client.ringo --cap osd 'allow rwx' \
--cap mon 'allow rwx' /etc/ceph/ceph.keyring
```

キーリングの作成と、そのキーリングに新しいユーザを追加する操作を同時に行うこともできます。

```
cephuser@adm > ceph-authtool -C /etc/ceph/ceph.keyring -n client.ringo \
--cap osd 'allow rwx' --cap mon 'allow rwx' --gen-key
```

前のシナリオでは、新しいユーザ`client.ringo`はキーリング内にのみ存在します。新しいユーザをCeph Storage Clusterに追加するには、同じようにクラスタに新しいユーザを追加する必要があります。

```
cephuser@adm > ceph auth add client.ringo -i /etc/ceph/ceph.keyring
```

### 30.2.3.4 ユーザの変更

キーリング内のユーザレコードのケーパビリティを変更するには、キーリングとユーザ、およびその後にケーパビリティを指定します。

```
cephuser@adm > ceph-authtool /etc/ceph/ceph.keyring -n client.ringo \
--cap osd 'allow rwx' --cap mon 'allow rwx'
```

変更したユーザをCephクラスタ環境内で更新するには、Cephクラスタ内でキーリングからユーザエントリに変更をインポートする必要があります。

```
cephuser@adm > ceph auth import -i /etc/ceph/ceph.keyring
```

キーリングからCeph Storage Clusterユーザを更新する方法の詳細については、[30.2.2.7項「ユーザのインポート」](#)を参照してください。

## 30.2.4 コマンドラインの使用

`ceph`コマンドは、ユーザ名と秘密の操作に関連する次のオプションをサポートします。

### --idまたは--user

Cephは、ユーザをタイプとIDで識別します(TYPE.ID。たとえば、client.adminやclient.user1)。id、name、および-nのオプションを使用して、ユーザ名のID部分を指定できます(たとえば、adminやuser1)。--idでユーザを指定して、タイプを省略できます。たとえば、ユーザclient.fooを指定するには、次のコマンドを入力します。

```
cephuser@adm > ceph --id foo --keyring /path/to/keyring health
cephuser@adm > ceph --user foo --keyring /path/to/keyring health
```

### --nameまたは-n

Cephは、ユーザをタイプとIDで識別します(TYPE.ID。たとえば、client.adminやclient.user1)。--nameおよび-nのオプションを使用して、完全修飾ユーザ名を指定できます。ユーザタイプ(通常はclient)とユーザIDを指定する必要があります。

```
cephuser@adm > ceph --name client.foo --keyring /path/to/keyring health
cephuser@adm > ceph -n client.foo --keyring /path/to/keyring health
```

### --keyring

1つ以上のユーザ名と秘密が含まれるキーリングのパス。--secretオプションも同じ機能を提供しますが、Object Gatewayでは動作しません。Object Gatewayでは、--secretは別の目的で使用されます。**ceph auth get-or-create**を使用してキーリングを取得し、ローカルに保存できます。キーリングのパスを切り替えずにユーザ名を切り替えることができるため、お勧めの方法です。

```
cephuser@adm > rbd map --id foo --keyring /path/to/keyring mypool/myimage
```

## A アップストリーム「Octopus」ポイントリリースに基づくCeph保守更新

SUSE Enterprise Storage 7のいくつかの主要パッケージは、CephのOctopusリリースシリーズに基づいています。Cephプロジェクト(<https://github.com/ceph/ceph>)がOctopusシリーズの新しいポイントリリースを公開した場合、SUSE Enterprise Storage 7は、アップストリームの最新のバグ修正や機能のバックポートのメリットを得られるように更新されます。この章には、製品に組み込み済みか、組み込みが予定されている各アップストリームポイントリリースに含まれる重要な変更点についての概要が記載されています。

### Octopus 15.2.5ポイントリリース

Octopusのポイントリリース15.2.5では、以下の修正とその他の変更が行われました。

- CephFS: 新しい拡張ディレクトリ属性である、分散型エフェメラルピンニングとランダムエフェメラルピンニングを使用して、自動静的サブツリーパーティショニングポリシーを自動で設定できるようになりました。詳細については、次のドキュメントを参照してください。 <https://docs.ceph.com/docs/master/cephfs/multimds/>
- Monitorに`mon_osd_warn_num_repaired`設定オプションが追加されました。デフォルトで10に設定されています。いずれかのOSDで、保存データのI/Oエラーを修復した数がこの値を超えた場合、`OSD_T00_MANY_REPAIRS`ヘルス警告が発生します。
- グローバルまたはプールごとに`no_scrub`フラグと`no_deep_scrub`フラグの両方またはいずれかが設定されていない場合、無効化されたタイプのスケジュール済みスクラブは中止されます。ユーザが開始したスクラブはすべて中断されません。
- 正常なクラスタでosdmapが最適化されない問題を修正しました。

### Octopus 15.2.4ポイントリリース

Octopusのポイントリリース15.2.4では、以下の修正とその他の変更が行われました。

- CVE-2020-10753: rgw: s3 CORSConfigurationのExposeHeaderで改行をサニタイズ
- Object Gateway: オーフアンを処理する`radosgw-admin`サブコマンド(`radosgw-admin orphans find`、`radosgw-admin orphans finish`、`radosgw-admin orphans list-jobs`)は非推奨となりました。これらのサブコマンドはあまり保守されておらず、またク

ラストに中間結果を保存するため空き容量の少ないクラスタを満杯にしてしまう可能性があります。そのため、現在、現在実験的と見なされている **rgw-orphan-list** ツールに置き換えられました。

- RBD: RBDのごみ箱を空にするスケジュールを保存するために使用される RBD プールオブジェクトの名前が `rbd_trash_trash_purge_schedule` から `rbd_trash_purge_schedule` に変更されました。ユーザが RBDのごみ箱を空にするスケジュール機能の使用を開始しており、プールごとまたはネームスペースごとにスケジュールを設定している場合は、`rbd_trash_trash_purge_schedule` オブジェクトを `rbd_trash_purge_schedule` にコピーしてからアップグレードして `rbd_trash_purge_schedule` を削除する必要があります。この削除には、ごみ箱を空にするスケジュールが設定されていたすべての RBD プールとネームスペースに対して次のコマンドを使用します。

```
rados -p pool-name [-N namespace] cp rbd_trash_trash_purge_schedule
      rbd_trash_purge_schedule
rados -p pool-name [-N namespace] rm rbd_trash_trash_purge_schedule
```

もしくは、他の便利な方法を使用して、アップグレード後にスケジュールを復元してください。

## Octopus 15.2.3 ポイントリリース

- Octopus ポイントリリース 15.2.3  
は、`bluefs_preextend_wal_files` と `bluefs_buffered_io` を同時に有効にした場合に WAL の破損が見られる問題に対処するためのホットフィックスリリースでした。15.2.3 での修正はあくまで応急処置です (`bluefs_preextend_wal_files` のデフォルト値を `false` に変更)。恒久的な修正では `bluefs_preextend_wal_files` オプションを完全に削除することになります。この修正は 15.2.6 ポイントリリースに含まれる可能性が高いです。

## Octopus 15.2.2 ポイントリリース

Octopus のポイントリリース 15.2.2 では、1 つのセキュリティ脆弱性にパッチを適用しました。

- CVE-2020-10736: MONとMGRでの認証バイパスを修正

## Octopus 15.2.1ポイントリリース

Octopusのポイントリリース15.2.1では、Luminous (SES5.5)からNautilus (SES6)さらにOctopus (SES7)に素早くアップグレードした際に、OSDがクラッシュする問題を修正しました。また、最初のOctopus (15.2.0)リリースに存在した2つのセキュリティ脆弱性にパッチを適用しました。

- CVE-2020-1759: msgr V2セキュアモードでのナンスの再利用を修正
- CVE-2020-1760: RGW GetObjectヘッダー分割によるXSSを修正

## B マニュアルの更新

この章には、SUSE Enterprise Storageの現行リリースに適用される、マニュアルの更新内容の一覧を記載しています。

- Cephダッシュボードの章(パートI「Cephダッシュボード」)を1つ上のレベルに移し、目次からトピックの詳細を直接検索できるようにしました。
- 『運用と管理ガイド』の構成を見直し、現行のガイドの範囲と一致させました。一部の章は、他のガイドに移動させました(jsc#SES-1397)。

# 用語集

## 一般

### Alertmanager

Prometheusサーバによって送信されるアラートを処理し、エンドユーザーに通知する単一のバイナリ。

### Ceph Manager

Ceph Manager (MGR)は、Cephの管理ソフトウェアです。クラスタ全体からすべての状態を一か所に収集します。

### Ceph Monitor

Ceph Monitor (MON)は、Cephのモニターソフトウェアです。

### Ceph Object Storage

オブジェクトストレージの「製品」、サービス、またはケーパビリティです。Ceph Storage ClusterとCeph Object Gatewayから構成されます。

### Ceph OSDデーモン

`ceph-osd`デーモンは、ローカルファイルシステムにオブジェクトを保存し、ネットワーク経由でそれらにアクセスできるようにするCephのコンポーネントです。

### Ceph Storage Cluster

ユーザのデータを保存するストレージソフトウェアのコアセット。1つのセットは複数のCeph Monitorと複数のOSDで構成されます。

### `ceph-salt`

Saltを使用して、`cephadm`に管理されるCephクラスタを展開するツールを提供します。

### `cephadm`

`cephadm`はCephクラスタを展開、管理します。その手段として、SSHを使用してマネージャデーモンからホストに接続し、Cephデーモンコンテナを追加、削除、更新します。

### CephFS

Cephのファイルシステム。

## CephX

Cephの認証プロトコル。CephXはKerberosのように動作しますが、単一障害点がありません。

## Cephクライアント

Ceph Storage Clusterにアクセスできる、Cephコンポーネントのコレクション。たとえば、Object Gateway、Ceph Block Device、CephFS、およびこれらに関連するライブラリ、カーネルモジュール、FUSEクライアントなどがあります。

## Cephダッシュボード

WebベースのCeph管理/監視用ビルトインアプリケーションで、クラスタの様々な側面とオブジェクトを管理します。このダッシュボードはCeph Managerモジュールとして実装されます。

## CRUSH、CRUSHマップ

「Controlled Replication Under Scalable Hashing」：データの保存場所を計算することによって、データの保存と取得の方法を決定するアルゴリズム。CRUSHは、クラスタ全体に均等に分散したデータを擬似ランダムにOSDで保存および取得するために、クラスタのマップを必要とします。

## CRUSHルール

特定のプール(単数または複数)に適用される、CRUSHのデータ配置ルール。

## DriveGroups

DriveGroupsは、物理ドライブにマッピングできる1つ以上のOSDレイアウトの宣言です。OSDレイアウトはCephが指定された基準を満たすようにOSDストレージをメディア上に物理的に割り当てる方法を定義します。

## Grafana

データベース分析および監視ソリューション。

## Metadata Server

メタデータサーバ(MDS)は、Cephのメタデータソフトウェアです。

## Multi-zone

## Object Gateway

Ceph Object Store用のS3/Swiftゲートウェイコンポーネント。RADOS Gateway (RGW)とも呼ばれます。

## OSD

「Object Storage Device」：物理ストレージユニットまたは論理ストレージユニット。

## **OSDノード**

データの保存、データレプリケーションの処理、回復、バックフィル、およびリバランスを実行し、他のCeph OSDデーモンを確認することによってCeph Monitorにモニタリング情報を提供するクラスタノード。

## **PG**

配置グループ: 「プール」を細分化したもので、パフォーマンスを調整するために使用します。

## **Prometheus**

システム監視およびアラートツールキット。

## **RADOS Block Device (RBD)**

Cephのブロックストレージコンポーネント。Cephブロックデバイスとも呼ばれます。

## **Reliable Autonomic Distributed Object Store (RADOS)**

ユーザのデータを保存するストレージソフトウェアのコアセット(MON+OSD)。

## **Samba**

Windowsとの統合ソフトウェア。

## **Sambaゲートウェイ**

SambaゲートウェイはWindowsドメインのActive Directoryに参加し、ユーザの認証と権限の付与を行います。

## **アーカイブ同期モジュール**

S3オブジェクトのバージョン履歴を保持するためのObject Gatewayゾーンを作成できるモジュール。

## **ゾーングループ**

## **ノード**

Cephクラスタ内の1つのマシンまたはサーバ。

## **バケット**

他のノードを物理的な場所の階層に集約するポイント。

## **プール**

ディスクイメージなどのオブジェクトを保存するための論理パーティション。

## **ポイントリリース**

バグ修正やセキュリティ上の修正だけを含む、応急措置的なリリース。

### **ルーティングツリー**

受信者が実行できるさまざまなルートを示す図に付けられる用語。

### **ルールセット**

プールのデータ配置を決定するためのルール。

### **管理ノード**

このホストからCeph関連のコマンドを実行して、クラスタのホストを管理します。