



Red Hat Ceph Storage 4

設定ガイド

Red Hat Ceph Storage の設定

Red Hat Ceph Storage 4 設定ガイド

Red Hat Ceph Storage の設定

法律上の通知

Copyright © 2023 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

このドキュメントでは、ブート時と実行時に Red Hat Ceph Storage を設定する手順を説明します。また、設定の参考情報も掲載しています。Red Hat では、コード、ドキュメント、Web プロパティーにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、弊社の CTO、Chris Wright のメッセージを参照してください。

目次

第1章 CEPH 設定の基本	4
1.1. 前提条件	4
1.2. CEPH の設定	4
1.3. CEPH 設定データベース	4
1.4. CEPH 設定ファイル	5
1.5. CEPH メタ変数の使用	9
1.6. ランタイム時に CEPH 設定を一覧表示	9
1.7. 実行時における特定設定の表示	10
1.8. 実行時における特定設定の定義	10
1.9. OSD メモリーターゲット	11
1.10. MDS メモリーキャッシュの制限	12
1.11. 関連情報	13
第2章 CEPH ネットワーク設定	14
2.1. 前提条件	14
2.2. CEPH のネットワーク設定	14
2.3. CEPH デーモンの設定要件	16
2.4. CEPH ネットワークメッセンジャー	17
2.5. パブリックネットワークの設定	18
2.6. プライベートネットワークの設定	18
2.7. ファイアウォール設定の確認	19
2.8. CEPH MONITOR ノードのファイアウォール設定	19
2.9. CEPH OSD のファイアウォール設定	20
2.10. MTU 値の確認および設定	21
2.11. 関連情報	23
第3章 CEPH MONITOR の設定	24
3.1. 前提条件	24
3.2. CEPH MONITOR の設定	24
3.3. CEPH クラスタマップ	24
3.4. CEPH MONITOR クォーラム	25
3.5. CEPH MONITOR の一貫性	25
3.6. CEPH MONITOR のブートストラップ	26
3.7. 設定ファイルの CEPH MONITOR セクション	26
3.8. CEPH MONITOR の最小設定	27
3.9. CEPH の一意の識別子	28
3.10. CEPH MONITOR のデータストア	28
3.11. CEPH ストレージの容量	29
3.12. CEPH ハートビート	30
3.13. CEPH MONITOR の同期ロール	30
3.14. CEPH の時刻同期	31
3.15. 関連情報	32
第4章 CEPH の認証設定	33
4.1. 前提条件	33
4.2. CEPHX 認証	33
4.3. CEPHX の有効化	33
4.4. CEPHX の無効化	35
4.5. CEPHX ユーザーキーリング	36
4.6. CEPHX デーモンのキーリング	36
4.7. CEPHX イメージの署名	37
4.8. 関連情報	37

第5章 プール、配置グループ、および CRUSH の設定	38
5.1. 前提条件	38
5.2. プール、配置グループ、および CRUSH	38
5.3. 関連情報	38
第6章 CEPH OBJECT STORAGE DAEMON (OSD) の設定	39
6.1. 前提条件	39
6.2. CEPH OSD の設定	39
6.3. OSD のスクラブ	39
6.4. OSD のバックフィル	40
6.5. OSD リカバリー	40
6.6. 関連情報	40
第7章 CEPH MONITOR と OSD の連動設定	41
7.1. 前提条件	41
7.2. CEPH MONITOR と OSD の連動	41
7.3. OSD ハートビート	41
7.4. OSD がダウンであることの報告	42
7.5. ピアリングの失敗の報告	43
7.6. OSD の報告状況	43
7.7. 関連情報	44
第8章 CEPH のデバッグとロギングの設定	45
8.1. 前提条件	45
8.2. CEPH のデバッグとロギング	45
8.3. 関連情報	45
付録A 一般的な設定オプション	46
付録B CEPH のネットワーク設定オプション	48
付録C CEPH MONITOR の設定オプション	55
付録D CEPHX の設定オプション	71
付録E プール、配置グループ、および CRUSH の設定オプション	75
付録F OBJECT STORAGE DAEMON (OSD) の設定オプション	81
付録G CEPH MONITOR と OSD の設定オプション	96
付録H CEPH のデバッグとロギングの設定オプション	101
付録I CEPH のスクラブオプション	107
付録J BLUESTORE の設定オプション	111

第1章 CEPH 設定の基本

ストレージ管理者としては、Ceph の設定を表示する方法と、Red Hat Ceph Storage クラスターの Ceph 設定オプションを設定する方法について、基本的な理解が必要です。実行時に Ceph の設定オプションを表示、設定することができます。

1.1. 前提条件

- Red Hat Ceph Storage ソフトウェアのインストール

1.2. CEPH の設定

すべての Red Hat Ceph Storage クラスターには、以下の項目を定義する設定があります。

- クラスター ID
- 認証設定
- Ceph デーモン
- ネットワーク設定
- ノード名およびアドレス
- キーリングへのパス
- OSD ログファイルへのパス
- 他のランタイムオプション

Ansible などのデプロイメントツールは、通常、初期の Ceph 設定ファイルを作成します。ただし、デプロイメントツールを使用して Red Hat Ceph Storage クラスターをブートストラップする場合には、独自に作成することができます。

1.3. CEPH 設定データベース

Ceph Monitor は、Ceph オプションの設定データベースを管理します。これにより、ストレージクラスター全体の設定オプションを格納することで、設定の管理を一元化します。Ceph の設定をデータベースに一元化することで、ストレージクラスターの管理に役立ちます。ローカルの Ceph 設定ファイル (デフォルトでは `/etc/ceph/ceph.conf`) で定義できる Ceph オプションはまだいくつかあります。これらのいくつかの Ceph 設定オプションは、他の Ceph コンポーネントが Ceph Monitor に接続して認証し、データベースから設定情報を取得する方法を制御します。

Ceph では、実行時にデーモンの設定を変更することができます。この機能は、デバッグ設定の有効化/無効化によりログ出力を増減する場合に役立ちます。さらに、ランタイムの最適化にも使用できます。



注記

同じオプションが設定データベースと Ceph 設定ファイルに存在する場合、設定データベースのオプションの優先順位は Ceph 設定ファイルで設定されているものよりも低くなります。

セクションおよびマスク

Ceph 設定ファイルで Ceph オプションをグローバルに、デーモンタイプごとに、または特定のデーモンごとに設定できるのと同様に、これらのセクションに従って設定データベースで Ceph オプションを設定することもできます。Ceph の設定オプションには、マスクを関連付けることができます。これらのマスクは、オプションを適用するデーモンやクライアントをさらに制限することができます。

マスクには 2 つの形式があります。

type:location

type は CRUSH プロパティーで、例えば **rack** や **host** などです。**location** は、プロパティータイプの値です。例えば、**host:foo** では、特定のノード (この例では **foo**) で動作しているデーモンまたはクライアントにのみオプションを制限します。

class:device-class

device-class は、**hdd** や **ssd** など、CRUSH デバイスクラスの名前です。たとえば、**class:ssd** は、ソリッドステートドライブ (SSD) ベースの Ceph OSD にのみオプションを制限します。このマスクは、クライアントの非 OSD デーモンには影響しません。

管理コマンド

Ceph 設定データベースは、サブコマンド **ceph config ACTION** で管理できます。実施できるアクションは以下のとおりです。

dump

ストレージクラスターのオプションの設定データベース全体をダンプします。

get WHO

特定のデーモンまたはクライアントの設定をダンプします。例えば、**WHO** は **mds.a** のようなデーモンになります。

set WHO OPTION VALUE

Ceph 設定データベースの設定オプションを設定します。

show WHO

実行中のデーモンについて、報告された実行中の設定を表示します。ローカル設定ファイルが使用されていたり、コマンドラインや実行時にオプションが上書きされていたりすると、これらのオプションは Ceph Monitor が保存するオプションとは異なる場合があります。また、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。

assimilate-conf -i INPUT_FILE -o OUTPUT_FILE

INPUT_FILE から設定ファイルを同化し、有効なオプションを Ceph Monitor の設定データベースに移動します。認識できない、無効な、または Ceph Monitor で制御できないオプションは、**OUTPUT_FILE** に格納された省略された設定ファイルで返されます。このコマンドは、従来の設定ファイルから一元化された設定データベースに移行する際に便利です。

help OPTION -f json-pretty

特定の **OPTION** のヘルプを JSON 形式の出力で表示します。

1.4. CEPH 設定ファイル

Ceph 設定ファイルは起動時に Ceph デーモンを設定し、これによりデフォルト値が上書きされます。

ヒント

各 Ceph デーモンには、**ceph/src/common/config_opts.h** ファイルで設定されるデフォルト値が連続して設定されます。

Ceph のデフォルト設定ファイルの場所は **/etc/ceph/ceph.conf** です。以下を使用して、別のパスを設定してその場所を変更できます。

- **\$CEPH_CONF** 環境変数のパスの設定。
- **-c** コマンドライン引数 例: **-c path/ceph.conf**) を指定します。

Ceph 設定ファイルには、**ini** スタイルの構文を使用します。コメントの前にシャープ記号 (#) またはセミコロン (;) を記入して、コメントを追加できます。

例

```
# <--A pound sign (#) sign precedes a comment.
# Comments always follow a semi-colon (;) or a pound (#) on each line.
# The end of the line terminates a comment.
# We recommend that you provide comments in your configuration file(s).
; A comment may be anything.
```

設定ファイルは、Ceph ストレージクラスター内のすべての Ceph デーモン、または特定タイプのすべての Ceph デーモンを起動時に設定できます。一連のデーモンを設定するには、以下のように設定を受け取るプロセスのセクションに設定を含める必要があります。

[global]

詳細

[global] の設定は、Ceph Storage クラスターのすべてのデーモンに影響します。

例

```
auth supported = cephx
```

[osd]

詳細

[osd] の設定は、Ceph Storage クラスター内のすべての **ceph-osd** デーモンに影響し、**[global]** で同じ設定を上書きします。

[mon]

詳細

[mon] の下にある設定は、Ceph Storage クラスター内のすべての **ceph-mon** デーモンに影響し、**[global]** で同じ設定を上書きします。

例

```
[mon.host01]
`host = host01`
`mon_addr = 10.0.0.101`
[mon.host02]
`host = host02`
`mon_addr = 10.0.0.102`
```

[client]

詳細

[client] の下にある設定は、すべての Ceph クライアントに影響します。たとえば、マウントされた Ceph ブロックデバイス、Ceph Object Gateway などです。

例

```
log file = /var/log/ceph/radosgw.log
```

グローバル設定は、Ceph ストレージクラスターの全デーモンのすべてのインスタンスに影響します。**[global]** の見出しは、Ceph Storage クラスターのすべてのデーモンに共通する値に使用します。以下のように各 **[global]** オプションを上書きできます。

- 特定のプロセスタイプのオプションを変更する。

例

```
[osp], [mon]
```

または

- 特定プロセスのオプションを変更する。

例

```
[osd.1]
```

特定のデーモンでオーバーライドするプロセスを除き、グローバル設定を上書きすると、すべての子プロセスが影響を受けます。

一般的なグローバル設定には、認証のアクティブ化が含まれます。

例

```
[global]
#Enable authentication between hosts within the cluster.
auth_cluster_required = cephx
auth_service_required = cephx
auth_client_required = cephx
```

特定の種類のデーモンに適用される設定を指定できます。特定のインスタンスを指定せずに **[osd]** または **[mon]** で設定を指定すると、設定はすべての OSD またはモニターのデーモンにそれぞれ適用されます。デーモン全体の設定の一例としては、osd メモリーターゲットがあります。

例

```
[osd]
osd_memory_target = 5368709120
```

デーモンの特定インスタンスの設定を指定できます。タイプとインスタンス ID をピリオド (.) で区切って入力することにより、インスタンスを指定することができます。Ceph OSD デーモンのインスタンス ID は常に数値ですが、Ceph モニターの場合は英数字である場合があります。

例

```
[osd.1]
# settings affect osd.1 only.

[mon.a]
# settings affect mon.a only.
```

一般的な Ceph 設定ファイルには、少なくとも以下の設定があります。

```
[global]
fsid = UNIQUE_CLUSTER_ID
mon_initial_members = NODE_NAME[, NODE_NAME]
mon_host = IP_ADDRESS[, IP_ADDRESS]

#All clusters have a front-side public network.
#If you have two NICs, you can configure a back side cluster
#network for OSD object replication, heart beats, backfilling,
#recovery, and so on
public_network = PUBLIC_NET[, PUBLIC_NET]
#cluster_network = PRIVATE_NET[, PRIVATE_NET]

#Clusters require authentication by default.
auth_cluster_required = cephx
auth_service_required = cephx
auth_client_required = cephx

#Choose reasonable numbers for your number of replicas
#and placement groups.
osd_pool_default_size = NUM # Write an object n times.
osd_pool_default_min_size = NUM # Allow writing n copy in a degraded state.
osd_pool_default_pg_num = NUM
osd_pool_default_pgp_num = NUM

#Choose a reasonable crush leaf type.
#0 for a 1-node cluster.
#1 for a multi node cluster in a single rack
#2 for a multi node, multi chassis cluster with multiple hosts in a chassis
#3 for a multi node cluster with hosts across racks, and so on
osd_crush_chooseleaf_type = NUM
```

例

```
[global]
cluster network = 10.74.250.101/21
fsid = 3e07d43f-688e-4284-bfb7-3e6ed5d3b77b
mon host = [v2:10.0.0.101:3300/0,v1:10.0.0.101:6789/0] [v2:10.0.0.102:3300/0,v1:10.0.0.102:6789/0]
[v2:10.0.0.103:3300/0,v1:10.0.0.103:6789/0]
mon initial members = host01, host02, host03
osd pool default crush rule = -1
public network = 10.74.250.101/21

[osd]
osd memory target = 4294967296

[mon]
[mon.host01]
host = host01
mon_addr = 10.0.0.101
[mon.host02]
host = host02
mon_addr = 10.0.0.102
```

1.5. CEPH メタ変数の使用

メタ変数は、Ceph ストレージクラスターの設定を大幅に簡素化します。メタ変数が設定値に設定されると、Ceph はそのメタ変数を具体的な値に展開します。

メタ変数は、Ceph 設定ファイルの **[global]** セクション、**[osd]** セクション、**[mon]** セクション、または **[client]** セクション内で使用すると非常に強力です。しかし、管理用ソケットでも使用可能です。Ceph メタ変数は、Bash のシェル拡張に似ています。

Ceph は以下のメタ変数をサポートしています。

\$cluster

詳細

Ceph ストレージクラスター名に展開します。同じハードウェアで複数の Ceph ストレージクラスターを実行する場合に便利です。

例

`/etc/ceph/$cluster.keyring`

デフォルト

`ceph`

\$type

詳細

インスタンスデーモンのタイプに応じて、**osd** または **mon** のいずれかに展開します。

例

`/var/lib/ceph/$type`

\$id

詳細

デーモン識別子に拡張します。**osd.0** の場合、これは **0** になります。

例

`/var/lib/ceph/$type/$cluster-$id`

\$host

詳細

インスタンスデーモンのホスト名に拡張します。

\$name

詳細

\$type.\$id まで展開します。

例

`/var/run/ceph/$cluster-$name.asok`

1.6. ランタイム時に CEPH 設定を一覧表示

Ceph 設定ファイルは、ブート時および実行時に表示することができます。

別添表

- Ceph OSD ノードへのルートレベルのアクセス。
- 管理キーリングへのアクセス。

手順

1. ランタイム設定を表示するには、デーモンを実行している Ceph ノードにログインして以下を実行します。

構文

```
ceph daemon DAEMON_TYPE.ID config show
```

osd.0 の設定を確認するには、**osd.0** を含むノードにログインして以下のコマンドを実行します。

例

```
[root@osd ~]# ceph daemon osd.0 config show
```

2. 追加のオプションについては、デーモンと **help** を指定します。

例

```
[root@osd ~]# ceph daemon osd.0 help
```

1.7. 実行時における特定設定の表示

Red Hat Ceph Storage の設定設定は、Ceph Monitor ノードから実行時に確認することができます。

前提条件

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

手順

1. Ceph ノードにログインして以下を実行します。

構文

```
ceph daemon DAEMON_TYPE.ID config get PARAMETER
```

例

```
[root@mon ~]# ceph daemon osd.0 config get public_addr
```

1.8. 実行時における特定設定の定義

ランタイム設定を指定する一般的な方法は2つあります。

- Ceph Monitor の使用
- Ceph 管理ソケットの使用

tell および **injectargs** コマンドを使用してモニターと通信すると、Ceph ランタイム設定オプションを設定できます。

前提条件

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

手順

1. オプションを挿入して Ceph Monitor を使用します。

```
ceph tell DAEMON_TYPE.DAEMON_ID or * injectargs --NAME VALUE [--NAME VALUE]
```

DAEMON_TYPE は **osd** または **mon** のいずれかに置き換えます。

ランタイム設定は、*を使用して特定タイプのすべてのデーモンに適用するか、特定の **DAEMON_ID** (数字または名前) を指定できます。

たとえば、**osd.0** という名前の **ceph-osd** デーモンのデバッグロギングを **0/5** に変更するには、以下のコマンドを実行します。

```
[root@osd ~]# ceph tell osd.0 injectargs '--debug-osd 0/5'
```



注記

tell コマンドは複数の引数を取るため、**tell** の各引数は一重引用符で囲み、設定の前に2つのダッシュを付けます ('--**NAME VALUE** [--**NAME VALUE**] ['--**NAME VALUE** [--**NAME VALUE**]]')。 **ceph tell** コマンドはモニターを通過します。

モニターにバインドできない場合は、Ceph 管理ソケットを使用して引き続き変更を加えることができます。

2. 設定を変更するデーモンのノードにログインします。

- a. Ceph デーモンに直接設定変更を発行します。

```
[root@osd ~]# ceph osd.0 config set debug_osd 0/5
```



注記

引用符には引数を1つしか指定しないため、**daemon** コマンドには引用符は必要ありません。

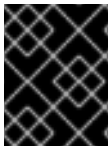
1.9. OSD メモリーターゲット

BlueStore は、**osd_memory_target** 設定オプションを使用して、OSD ヒープメモリーの使用を指定されたターゲットサイズで保持します。

osd_memory_target オプションは、システムで利用可能な RAM に基づいて OSD メモリーを設定します。デフォルトでは、Ansible は値を 4 GB に設定します。デーモンをデプロイする際に、`/usr/share/ceph-ansible/group_vars/all.yml` ファイルで、バイト単位で示している値を変更できます。**ceph.conf** ファイルの Ceph オーバーライドを使用して、**osd memory target** をたとえば 6 GB に手動で設定することもできます。

例

```
ceph_conf_overrides:
  osd:
    osd memory target: 6442450944
```



重要

Ceph オーバーライドを使用してオプションを設定する場合は、アンダースコアなしでオプションを使用します。

Ceph OSD のメモリーキャッシングは、ブロックデバイスが低速である場合に重要となります (例えば、従来のハードドライブの場合)。キャッシュヒットのメリットがソリッドステートドライブの場合よりもはるかに大きいからです。ただし、ハイパーコンバージドインフラストラクチャー (HCI) や他のアプリケーションなど、他のサービスと OSD を共存させる場合には、この点を考慮する必要があります。



注記

osd_memory_target の値は、従来のハードドライブデバイス用のデバイスごとに1つの OSD、NVMe SSD デバイス用のデバイスごとに2つの OSD です。**osds_per_device** は `group_vars/osds.yml` ファイルで定義されます。

関連情報

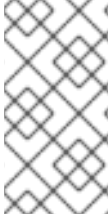
- **osd_memory_target** の設定については、[RHCS3 - osd_memory_target what it is and how ceph-ansible calculates the value](#) を参照してください。

1.10. MDS メモリーキャッシュの制限

MDS サーバーは、そのメタデータを別のストレージプール (**cephfs_metadata**) に保持し、Ceph OSD のユーザーです。Ceph File System の場合、MDS サーバーはストレージクラスター内の単一のストレージデバイスだけでなく、Red Hat Ceph Storage クラスター全体をサポートする必要があるため、特にワークロードが小/中サイズのファイルで設定されている場合 (データに対するメタデータの比率が高い)、メモリー要件が大きくなる可能性があります。

例: **mds_cache_memory_limit** を 2 GiB に設定します

```
ceph_conf_overrides:
  osd:
    mds_cache_memory_limit: 2147483648
```

注記

メタデータを多用するワークロードを持つ大規模な Red Hat Ceph Storage クラスターでは、MDS サーバーを他のメモリーを多用するサービスと同じノードに置かないでください。そうすることで、より多くのメモリー (たとえば 100 GB を超えるサイズ) を MDS に割り当てることができます。

関連情報

- Red Hat Ceph Storage File システムガイドの [MDS キャッシュサイズ制限の理解](#) を参照してください。

1.11. 関連情報

- オプションの詳細や使用方法は、[付録 A](#) の一般的な Ceph 設定オプションを参照してください。

第2章 CEPH ネットワーク設定

ストレージ管理者は、Red Hat Ceph Storage クラスターが動作するネットワーク環境を理解し、それに応じて Red Hat Ceph Storage を設定する必要があります。Ceph のネットワークオプションを理解して設定することで、ストレージクラスター全体のパフォーマンスと信頼性を最適化することができます。

2.1. 前提条件

- ネットワーク接続
- Red Hat Ceph Storage ソフトウェアのインストール

2.2. CEPH のネットワーク設定

高性能な Red Hat Ceph Storage クラスターを構築するには、ネットワークの設定が重要です。Ceph ストレージクラスターは、Ceph クライアントに代わって要求のルーティングやディスパッチを実行しません。代わりに、Ceph クライアントは Ceph OSD デーモンに直接要求を出します。Ceph OSD は Ceph クライアントに代わってデータレプリケーションを実行するため、レプリケーションおよび他の要素によって Ceph ストレージクラスターのネットワークに追加の負荷がかかります。

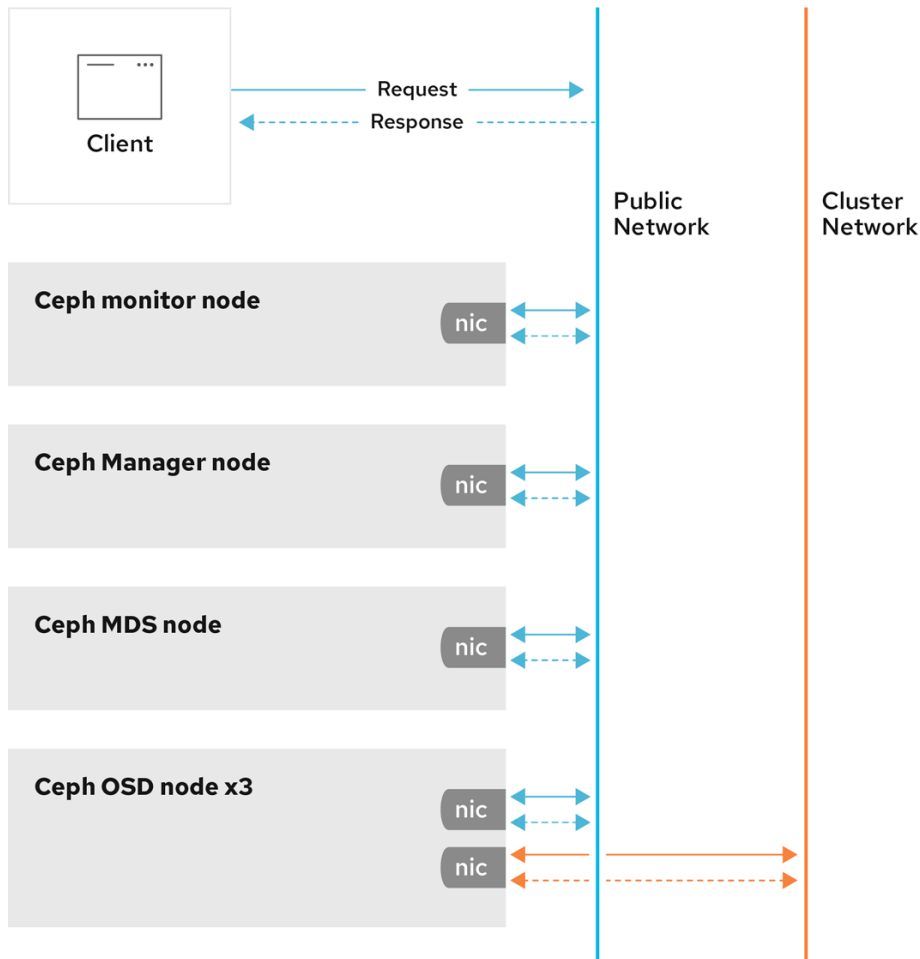
すべての Ceph クラスターは、パブリックネットワークを使用する必要があります。ただし、内部のクラスターネットワークを指定しない限り、Ceph は1つのパブリックネットワークを想定します。Ceph はパブリックネットワークでのみ機能しますが、大規模なストレージクラスターの場合は、クラスター関連のトラフィックのみを伝送する第2のプライベートネットワークを使用すると、パフォーマンスが大幅に向上します。



重要

Red Hat では、Ceph ストレージクラスターを2つのネットワークで運用することを推奨しています (1つのパブリックネットワークと1つのプライベートネットワーク)。

2つのネットワークをサポートするには、各 Ceph Node に複数のネットワークインターフェイスカード (NIC) が必要になります。



110_Ceph_0720

2つの別々のネットワークを運用することを検討する理由はいくつかあります。

- **パフォーマンス:** Ceph OSD は Ceph クライアントのデータレプリケーションを処理します。Ceph OSD がデータを複数回複製すると、Ceph OSD 間のネットワーク負荷は、Ceph クライアントと Ceph ストレージクラスター間のネットワーク負荷をすぐに阻害してしまいます。これによりレイテンシーが発生し、パフォーマンスに問題が生じます。リカバリーやリバランシングを行うと、パブリックネットワーク上で大きなレイテンシーが発生します。
- **セキュリティ:** 通常、多くのユーザーはサービス拒否 (DoS) 攻撃と呼ばれる攻撃に関与します。Ceph OSD 間のトラフィックが中断されると、ピアリングが失敗し、配置グループが **active + clean** 状態を反映しなくなり、ユーザーがデータを読み書きできなくなる可能性があります。この種の攻撃に対抗するには、インターネットに直接接続しない、完全に独立したクラスターネットワークを維持することが有効です。

ネットワーク設定の定義は必要ありません。Ceph はパブリックネットワークでのみ機能するので、Ceph デーモンを実行するすべてのホストでパブリックネットワークが設定されている必要があります。しかし、Ceph では、複数の IP ネットワークやサブネットマスクなど、より具体的な条件をパブリックネットワークに設定することができます。また、OSD ハートビート、オブジェクトのレプリケーション、およびリカバリートラフィックを処理するために、別のクラスターネットワークを構築することもできます。

設定で定義する IP アドレスと、ネットワーククライアントがサービスにアクセスする際に使用する公開用の IP アドレスを混同しないようにしてください。通常、内部 IP ネットワークは **192.168.0.0** または **10.0.0.0** です。



注記

Ceph はサブネットに CIDR 表記を使用します (例: **10.0.0.0/24**)。



重要

パブリックネットワークまたはプライベートネットワークのいずれかに複数の IP アドレスとサブネットマスクを指定する場合、ネットワーク内のサブネットは相互にルーティング可能でなければなりません。さらに、各 IP アドレスとサブネットを IP テーブルに含め、必要に応じてポートを開くようにしてください。

ネットワークの設定が完了したら、クラスターの再起動や各デーモンの再起動を行います。Ceph デーモンは動的にバインドするので、ネットワーク設定を変更してもクラスター全体を一度に再起動する必要はありません。

2.3. CEPH デーモンの設定要件

Ceph には、すべてのデーモンに適用される 1 つのネットワーク設定要件があります。Ceph 設定ファイルは、各デーモンに **host** を指定する必要があります。



重要

デプロイメントユーティリティによっては、設定ファイルを作成してくれる場合があります。デプロイメントユーティリティがこれらの値を設定する場合は、設定しないでください。



重要

host オプションは、FQDN ではなく、ノードの短縮名です。IP アドレスではありません。

ホスト名を指定して、デーモンがある場所の **host** 名と IP アドレスを設定できます。

例

```
[mon.a]
  host = host01
  mon_addr = 10.0.0.101:6789, 10.0.0.101:3300

[osd.0]
  host = host02
```

デーモンにノード IP アドレスを設定する必要はありません。これは任意の設定です。静的 IP 設定でパブリックネットワークとプライベートネットワークの両方を実行している場合、Ceph 設定ファイルで各デーモンのノードの IP アドレスを指定する場合があります。デーモンに静的 IP アドレスを設定するには、Ceph 設定ファイルのデーモンインスタンスセクションに記述する必要があります。

例

```
[osd.0]
  public_addr = 10.74.250.101/21
  cluster_addr = 10.74.250.101/21
```

2つのネットワークを持つクラスターに、単一の NIC を持つ OSD ホストを強制的にデプロイすることができます。Ceph 設定ファイルの `[osd.n]` セクションに **public addr** エントリーを追加することにより、OSD ホストを強制的にパブリックネットワークで操作できます。**n** は、1つの NIC を持つ OSD の数を示します。さらに、パブリックネットワークとクラスターネットワークは互いにトラフィックをルーティングできる必要がありますが、セキュリティ上の理由から Red Hat は推奨していません。



重要

Red Hat は、セキュリティ上の理由から、単一の NIC で2つのネットワークに接続する OSD ノードをデプロイすることを推奨しません。

関連情報

- 具体的なオプションの説明や使用方法は、Red Hat Ceph Storage 設定ガイドの [付録 B](#) のホストオプションを参照してください。
- 具体的なオプションの説明や使用方法は、Red Hat Ceph Storage 設定ガイドの [付録 B](#) の共通オプションを参照してください。

2.4. CEPH ネットワークメッセンジャー

メッセンジャーは Ceph ネットワーク層の実装です。Red Hat は2種類のメッセンジャーをサポートしています。

- **simple**
- **async**

Red Hat Ceph Storage 3 以降では、**async** がデフォルトの messenger タイプです。messenger タイプを変更するには、Ceph 設定ファイルの `[global]` セクションに **ms_type** 設定を指定します。



注記

async messenger では、Red Hat は **posix** トランスポートタイプをサポートしますが、現在 **rdma** または **dpdk** をサポートしていません。デフォルトでは、Red Hat Ceph Storage 3 以降の **ms_type** 設定は **async+posix** を反映します。ここで、**async** は messenger タイプで、**posix** はトランスポートタイプになります。

SimpleMessenger

SimpleMessenger 実装は、1ソケットあたり2つのスレッドを持つ TCP ソケットを使用します。Ceph は、各論理セッションを接続に関連付けます。パイプは、各メッセージの入力と出力を含む接続を処理します。**SimpleMessenger** は、**posix** トランスポートタイプに有効ですが、**rdma**、**dpdk** などの他のトランスポートタイプには有効ではありません。

AsyncMessenger

したがって、**AsyncMessenger** は、Red Hat Ceph Storage 3 以降のデフォルトのメッセンジャータイプです。Red Hat Ceph Storage 3 以降では、**AsyncMessenger** 実装は、接続用に固定サイズのスレッドプールを持つ TCP ソケットを使用します。これは、レプリカまたはイレイジャーコードチャンクの最大数と同じでなければなりません。CPU 数が少なかったり、サーバーあたりの OSD 数が多かったりしてパフォーマンスが低下する場合は、スレッドカウントを低い値に設定することができます。



注記

現時点で、Red Hat は **rdma**、**dppk** などの他のトランスポートタイプをサポートしていません。

関連情報

- 具体的なオプションの説明と使用方法は、Red Hat Ceph Storage 設定の [付録 B](#) の AsyncMessenger オプションを参照してください。
- Ceph messenger バージョン 2 プロトコルを使用した [ネットワーク上の暗号化](#) の使用に関する詳細は、Red Hat Ceph Storage アーキテクチャーガイドを参照してください。

2.5. パブリックネットワークの設定

パブリックネットワークの設定では、特にパブリックネットワークの IP アドレスとサブネットを定義することができます。特定のデーモンの **public addr** 設定を使用して、静的 IP アドレスまたは **public network** 設定をオーバーライドできます。

前提条件

- Red Hat Ceph Storage ソフトウェアのインストール

手順

1. Ceph 設定ファイルの **[global]** セクションに、以下の設定を追加します。

```
[global]
...
public_network = PUBLIC-NET/NETMASK
```

関連情報

- 具体的なオプションの説明や使用方法は、Red Hat Ceph Storage 設定ガイドの [付録 B](#) の共通オプションを参照してください。

2.6. プライベートネットワークの設定

クラスターネットワークを宣言した場合、OSD はハートビート、オブジェクトのレプリケーション、およびリカバリトラフィックをクラスターネットワーク上でルーティングします。これにより、単一のネットワークを使用する場合と比較して、パフォーマンスが向上します。



重要

セキュリティ強化のためは、クラスターネットワークにはパブリックネットワークやインターネットからアクセスできないようにすることが望ましいです。

クラスターネットワーク設定により、クラスターネットワークを宣言し、特にクラスターネットワークの IP アドレスおよびサブネットを定義できます。特定の OSD デーモンの **cluster addr** 設定を使用して、静的 IP アドレスを割り当てるか、**cluster network** 設定を上書きすることができます。

前提条件

- 稼働中の Red Hat Ceph Storage クラスタがある。
- Ceph ソフトウェアリポジトリへのアクセス。

手順

1. Ceph 設定ファイルの **[global]** セクションに、以下の設定を追加します。

```
[global]
...
cluster_network = CLUSTER-NET/NETMASK
```

2.7. ファイアウォール設定の確認

デフォルトでは、デーモンは **6800:7100** 範囲内のポートにバインドされます。この範囲は、ユーザーの判断で設定することができます。ファイアウォールを設定する前に、デフォルトのファイアウォール設定を確認してください。

前提条件

- 稼働中の Red Hat Ceph Storage クラスタがある。
- Ceph ソフトウェアリポジトリへのアクセス。
- Ceph Monitor ノードへのルートレベルのアクセス。

手順

1. この範囲は、ユーザーの判断で設定することができます。

```
[root@mon ~]# sudo iptables -L
```

2. **firewalld** デーモンの場合は、以下のコマンドを実行します。

```
[root@mon ~]# firewall-cmd --list-all-zones
```

一部の Linux ディストリビューションには、すべてのネットワークインターフェイスからの SSH を除くすべてのインバウンドリクエストを拒否するルールが含まれています。

例

```
REJECT all -- anywhere anywhere reject-with icmp-host-prohibited
```

2.8. CEPH MONITOR ノードのファイアウォール設定

Ceph モニターはデフォルトでポート **3300** および **6789** をリスンします。さらに、Ceph モニターは常にパブリックネットワーク上で動作します。

前提条件

- 稼働中の Red Hat Ceph Storage クラスタがある。

- Ceph ソフトウェアリポジトリへのアクセス。
- Ceph Monitor ノードへのルートレベルのアクセス。

手順

1. 以下の例を使用してルールを追加します。

```
[root@mon ~]# sudo iptables -A INPUT -i IFACE -p tcp -s IP-ADDRESS/NETMASK --dport 6789 -j ACCEPT
[root@mon ~]# sudo iptables -A INPUT -i IFACE -p tcp -s IP-ADDRESS/NETMASK --dport 3300 -j ACCEPT
```

- a. **IFACE** は、パブリックネットワークインターフェイスに置き換えます。たとえば、**eth0**、**eth1** などです。
- b. **IP-ADDRESS** は、パブリックネットワークの IP アドレスに、**NETMASK** は、パブリックネットワークのネットマスクに置き換えます。

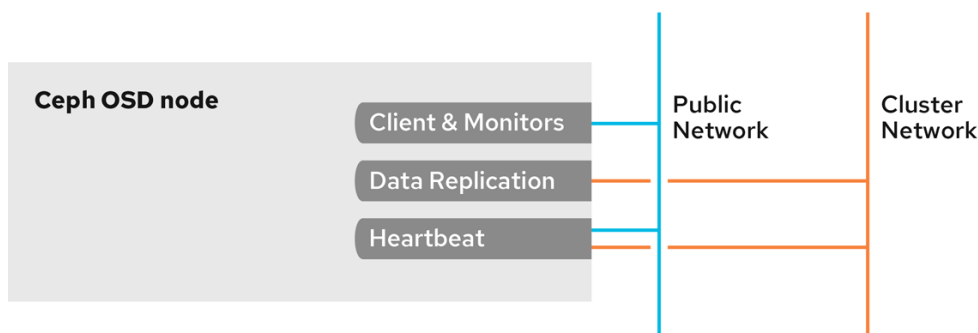
2. **firewalld** デーモンの場合は、以下のコマンドを実行します。

```
[root@mon ~]# firewall-cmd --zone=public --add-port=6789/tcp
[root@mon ~]# firewall-cmd --zone=public --add-port=6789/tcp --permanent
[root@mon ~]# firewall-cmd --zone=public --add-port=3300/tcp
[root@mon ~]# firewall-cmd --zone=public --add-port=3300/tcp --permanent
```

2.9. CEPH OSD のファイアウォール設定

デフォルトでは、Ceph OSD は、ポート 6800 から順に Ceph ノードで最初に利用可能なポートにバインドします。ノード上で動作する各 OSD に対して、6800 番ポート以降の少なくとも 4 つのポートを開くようにしてください。

- 1つはパブリックネットワーク上のクライアントおよびモニターとの通信用
- 1つはクラスターネットワーク上の他の OSD へのデータ送信用
- 2つはクラスターネットワーク上でのハートビートパケット送信用



110_Ceph_0720

ポートはノードごとに異なります。ただし、プロセスが再起動されてバインドされたポートが解放されない場合には、その Ceph ノードで実行されている Ceph デーモンが必要とするポート数よりも多くのポートを開く必要があるかもしれません。デーモンに障害が発生し、ポートを解放せずに再起動した場合に、再起動したデーモンが新しいポートにバインドするように、さらにいくつかのポートを開くことを検討してください。また、各 OSD ノードでポート範囲 **6800:7300** を開くことを検討してください。

パブリックネットワークとクラスターネットワークを別々に設定した場合、クライアントはパブリックネットワークを使用して接続し、他の Ceph OSD デーモンはクラスターネットワークを使用して接続するため、パブリックネットワークとクラスターネットワークの両方にルールを追加する必要があります。

前提条件

- 稼働中の Red Hat Ceph Storage クラスタがある。
- Ceph ソフトウェアリポジトリへのアクセス。
- Ceph OSD ノードへのルートレベルのアクセス。

手順

1. 以下の例を使用してルールを追加します。

```
[root@mon ~]# sudo iptables -A INPUT -i IFACE -m multiport -p tcp -s IP-ADDRESS/NETMASK --dports 6800:6810 -j ACCEPT
```

- a. **IFACE** は、パブリックネットワークインターフェイス (例: **eth0**、**eth1** など) に置き換えます。
- b. **IP-ADDRESS** は、パブリックネットワークの IP アドレスに、**NETMASK** は、パブリックネットワークのネットマスクに置き換えます。

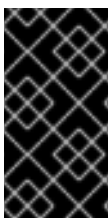
2. **firewalld** デーモンの場合は、次のコマンドを実行します。

```
[root@mon ~] # firewall-cmd --zone=public --add-port=6800-6810/tcp
[root@mon ~] # firewall-cmd --zone=public --add-port=6800-6810/tcp --permanent
```

クラスターネットワークを別のゾーンに配置した場合は、そのゾーン内のポートを適切に開きます。

2.10. MTU 値の確認および設定

最大伝送単位 (MTU) 値は、リンク層で送信される最大パケットのサイズ (バイト単位) です。MTU のデフォルト値は 1500 バイトです。Red Hat は、Red Hat Ceph Storage クラスタには、MTU 値が 9000 バイトのジャンボフレームを使用することを推奨します。



重要

Red Hat Ceph Storage では、パブリックネットワークとクラスターネットワークの両方で、通信パスにあるすべてのネットワークデバイスに同じ MTU 値がエンドツーエンドで必要となります。Red Hat Ceph Storage クラスタを実稼働環境で使用する前に、環境内のすべてのノードとネットワーク機器で MTU 値が同じであることを確認します。



注記

ネットワークインターフェイスをボンディングする場合には、MTU の値はボンディングされたインターフェイスでのみ設定する必要があります。新しい MTU 値は、ボンディングデバイスから下層のネットワークデバイスに伝播します。

前提条件

- ノードへのルートレベルのアクセス。

手順

1. 現在の MTU 値を確認します。

例

```
[root@mon ~]# ip link list
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode
DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp22s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
mode DEFAULT group default qlen 1000
    link/ether 40:f2:e9:b8:a0:48 brd ff:ff:ff:ff:ff:ff
```

この例では、ネットワークインターフェイスは **enp22s0f0** で、MTU の値は **1500** です。

2. オンラインで MTU 値を **一時的に** 変更するには、以下を実行します。

構文

```
ip link set dev NET_INTERFACE mtu NEW_MTU_VALUE
```

例

```
[root@mon ~]# ip link set dev enp22s0f0 mtu 9000
```

3. MTU 値を **永続的に** 変更するには、以下を行います。

- a. その特定のネットワークインターフェイスのネットワーク設定ファイルを編集するために開きます。

構文

```
vim /etc/sysconfig/network-scripts/ifcfg-NET_INTERFACE
```

例

```
[root@mon ~]# vim /etc/sysconfig/network-scripts/ifcfg-enp22s0f0
```

- b. 新しい行で、**MTU=9000** オプションを追加します。

例

```
NAME="enp22s0f0"
DEVICE="enp22s0f0"
MTU=9000 1
ONBOOT=yes
NETBOOT=yes
UUID="a8c1f1e5-bd62-48ef-9f29-416a102581b2"
```

```
IPV6INIT=yes  
BOOTPROTO=dhcp  
TYPE=Ethernet
```

c. network サービスを再起動します。

例

```
[root@mon ~]# systemctl restart network
```

関連情報

- 詳細は、Red Hat Enterprise Linux 8 の [Configuring and Managing Networking](#) を参照してください。
- 詳細は、Red Hat Enterprise Linux 7 の [Networking Guide](#) を参照してください。

2.11. 関連情報

- オプションの詳細や使用方法は、[付録 B](#) の Red Hat Ceph Storage ネットワーク設定オプションを参照してください。
- Ceph messenger バージョン 2 プロトコルを使用した [ネットワーク上の暗号化](#) の使用に関する詳細は、[Red Hat Ceph Storage アーキテクチャーガイド](#)を参照してください。

第3章 CEPH MONITOR の設定

ストレージ管理者として、Ceph Monitor のデフォルト設定値を使用することも、目的のワークロードに応じてカスタマイズすることもできます。

3.1. 前提条件

- Red Hat Ceph Storage ソフトウェアのインストール

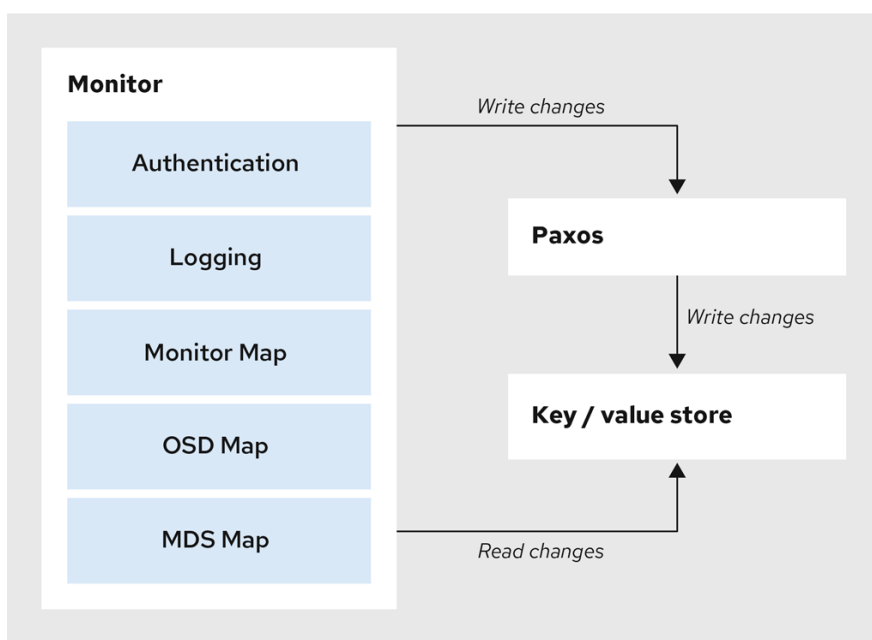
3.2. CEPH MONITOR の設定

Ceph Monitor の設定方法を理解することは、信頼性の高い Red Hat Ceph Storage クラスタを構築する上で重要なことです。すべてのストレージクラスターには少なくとも1つのモニターがあります。通常、Ceph Monitor の設定はほぼ一定のままですが、ストレージクラスター内の Ceph Monitor を追加、削除、または交換することができます。

Ceph モニターは、クラスターマップのマスターコピーを維持します。つまり、1つの Ceph モニターに接続して最新のクラスターマップを取得するだけで、Ceph クライアントはすべての Ceph モニターと Ceph OSD の位置を把握することができます。

Ceph クライアントが Ceph OSD に対して読み取り/書き込みを行うには、まず Ceph Monitor に接続する必要があります。クラスターマップの現在のコピーと CRUSH アルゴリズムを使用して、Ceph クライアントは任意のオブジェクトの位置を計算できます。オブジェクトの位置を計算できることで、Ceph クライアントは Ceph OSD と直接対話できます。このことは、Ceph の高いスケーラビリティとパフォーマンスを実現する上で非常に重要な要素となります。

Ceph Monitor の主なロールは、クラスターマップのマスターコピーを維持することです。Ceph Monitor は、認証とログサービスも提供します。Ceph Monitor は、モニターサービスのすべての変更を1つの Paxos インスタンスに書き込み、Paxos はその変更をキー/値ストアに書き込んで強い一貫性を持たせます。Ceph Monitor は、同期操作中にクラスターマップの最新バージョンにクエリーを行うことができます。Ceph Monitor は、**rocksdb** データベースを使用したキー値ストアのスナップショットやイテレーターを使用して、ストア全体の同期を実行します。



110_Ceph_0720

3.3. CEPH クラスターマップ

クラスターマップは、モニターマップ、OSD マップ、および配置グループマップなどのマップを合成したものです。クラスターマップは、多くの重要なイベントを追跡します。

- どのプロセスが Red Hat Ceph Storage クラスター内 (**in**) にあるか。
- Red Hat Ceph Storage クラスター内 **in** にあるプロセスが **up** で稼働しているか、**down** であるか。
- 配置グループが **active** または **inactive** で **clean** な、または他の一部の状態にあるかどうか。
- クラスターの現状を反映したその他の詳細情報。これには以下が含まれます。
 - ストレージ容量の合計、または
 - 使用されているストレージ容量の合計

例えば、Ceph OSD がダウンしたり、配置グループがデグレード状態に陥ったりするなど、クラスターの状態に大きな変化があった場合。クラスターマップが更新され、クラスターの現在の状態が反映されます。さらに、Ceph モニターはクラスターの以前の状態の履歴も保持します。モニターマップ、OSD マップ、および配置グループマップは、それぞれのマップバージョンの履歴を保持します。各バージョンは **エポック** と呼ばれます。

Red Hat Ceph Storage クラスターを運用する場合、これらの状態を追跡することはクラスター管理の重要な部分です。

3.4. CEPH MONITOR クォーラム

クラスターは、1 台のモニターで十分に動作します。しかし、1 台のモニターは単一故障点になります。本番環境の Ceph ストレージクラスターで高可用性を確保するには、複数のモニターで Ceph を実行し、1 つのモニターの故障がストレージクラスター全体の障害にならないようにします。

Ceph ストレージクラスターが高可用性のために複数の Ceph Monitor を実行している場合、Ceph Monitor は Paxos アルゴリズムを使用してマスタークラスターマップに関する合意を確立します。コンセンサスを得るには、大半のモニターが動作していて、クラスターマップに関するコンセンサスのためのクォーラムを確立する必要があります。例えば、1、3 つのうちの 2 つ、5 つのうちの 3 つ、6 つのうちの 4 つ等。

Red Hat では、高可用性を確保するために、少なくとも 3 つの Ceph Monitor で本番環境の Red Hat Ceph Storage クラスターを実行することを推奨しています。複数のモニターを実行する場合、クォーラムを確立するためにストレージクラスターのメンバーでなければならない初期モニターを指定することができます。これにより、ストレージクラスターがオンラインになるまでの時間が短縮される場合があります。

```
[mon]
mon_initial_members = a,b,c
```



注記

クォーラムを確立するには、ストレージクラスター内のモニターの **大半** が相互に到達できる必要があります。**mon_initial_members** オプションでクォーラムを確立するモニターの最初の数減らすことができます。

3.5. CEPH MONITOR の一貫性

Ceph 設定ファイルにモニター設定を追加する場合、Ceph Monitor モニターのアーキテクチャ的な側

面をいくつか知っておく必要があります。Ceph は、クラスター内で別の Ceph Monitor を検出する際に、Ceph Monitor に厳格な一貫性要件を課します。Ceph クライアントおよびその他の Ceph デーモンは、Ceph 設定ファイルを使用してモニターを検出し、Ceph 設定ファイルではなくモニターマップ (**monmap**) を使用して相互を検出します。

Ceph Monitor が Red Hat Ceph Storage クラスター内の他の Ceph Monitor を検出する場合、常にモニターマップのローカルコピーを参照します。Ceph 設定ファイルではなくモニターマップを使用することで、クラスターが壊れる可能性のあるエラーを回避できます。例えば、Ceph 設定ファイルでモニターのアドレスやポートを指定する際のタイプミスなどです。モニターは検出のためにモニターマップを使用し、クライアントや他の Ceph デーモンとモニターマップを共有するため、モニターマップは、モニターのコンセンサスが有効であることをモニターに対して厳格に保証します。

モニターマップへの更新適用時の厳格な一貫性

Ceph Monitor の他の更新と同様に、モニターマップへの変更は常に Paxos と呼ばれる分散型コンセンサスアルゴリズムを介して行われます。Ceph Monitor は、Ceph Monitor の追加や削除など、モニターマップへの各更新について合意し、クォーラムの各モニターが同じバージョンのモニターマップを持つようにする必要があります。モニターマップへの更新はインクリメンタルに行われるため、Ceph Monitor は最新の合意バージョンと以前のバージョンのセットを持つことになります。

履歴の維持

履歴を維持することで、古いバージョンのモニターマップを持つ Ceph Monitor が、Red Hat Ceph Storage クラスターの現在の状態に追いつくことができます。

Ceph Monitor がモニターマップではなく Ceph 設定ファイルを介してお互いを検出する場合、Ceph 設定ファイルは自動的に更新および配布されないため、新たなリスクが発生する可能性があります。Ceph Monitor が誤って古い Ceph 設定ファイルを使用し、Ceph Monitor の識別に失敗し、クォーラムから外れたり、Paxos がシステムの現在の状態を正確に判断できなかったりする状況が発生する可能性があります。

3.6. CEPH MONITOR のブートストラップ

ほとんどの設定とデプロイメントの場合、Ansible などの Ceph をデプロイするツールは、モニターマップを生成して Ceph モニターのブートストラップを支援することがあります。

Ceph モニターには、いくつかの明示的な設定が必要です。

- **ファイルシステム ID: fsid** は、オブジェクトストアの一意識別子です。同じハードウェア上で複数のストレージクラスターを稼働させることができるため、モニターのブートストラップを行う場合には、オブジェクトストアの一意の ID を指定する必要があります。Ansible などのデプロイメントツールを使用すると、ファイルシステムの識別子が生成されますが、**fsid** も手動で指定できます。
- **モニター ID:** モニター ID は、クラスター内の各モニターに割り当てられる一意の ID です。ID は英数字の値で、慣例的には通常、アルファベットのインクリメントに従います。たとえば、**a**、**b** などです。これは Ceph 設定ファイルで設定できます。例: **[mon.a]**、**[mon.b]** など、デプロイメントツール、または **ceph** コマンドの使用です。
- **キー:** モニターには秘密鍵が必要です。

3.7. 設定ファイルの CEPH MONITOR セクション

クラスター全体に設定を適用するには、**[global]** セクションに設定設定を入力します。クラスター内のすべてのモニターに設定を適用するには、**[mon]** セクションに設定設定を入力します。特定のモニターに設定を適用するには、モニターインスタンスを指定します。

例

[mon.a]

慣習的に、モニターインスタンス名にはアルファ表記が使用されます。

[global]

[mon]

[mon.a]

[mon.b]

[mon.c]

3.8. CEPH MONITOR の最小設定

Ceph 設定ファイルの Ceph モニターの最低限のモニター設定には、各モニターのホスト名 (DNS に設定されていない場合) とモニターアドレスが含まれます。これらの設定は、**[mon]** の下、または特定のモニターのエントリーの下で設定できます。

```
[mon]
mon_host = hostname1,hostname2,hostname3
mon_addr =
10.0.0.10:6789,10.0.0.11:6789,10.0.0.12:6789,10.0.0.10:3300,10.0.0.11:3300,10.0.0.12:3300
```

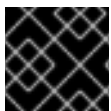
または、以下を実行します。

```
[mon.a]
host = hostname1
mon_addr = 10.0.0.10:6789, 10.0.0.10:3300
```



注記

このモニターの最小設定は、デプロイメントツールが **fsid** と **mon.** キーを生成することを前提としています。



重要

Ceph クラスターをデプロイしたら、モニターの IP アドレスを変更しないでください。

DNS ルックアップ用に Ceph クラスターを設定するには、Ceph 設定ファイルの **mon_dns_srv_name** 設定を設定します。

設定が完了したら、DNS の設定を行います。DNS ゾーンにモニターの IPv4 (A) または IPv6 (AAAA) いずれかのレコードを作成します。

例

```
#IPv4
mon1.example.com. A 192.168.0.1
mon2.example.com. A 192.168.0.2
```

```
mon3.example.com. A 192.168.0.3
```

```
#IPv6
```

```
mon1.example.com. AAAA 2001:db8::100
```

```
mon2.example.com. AAAA 2001:db8::200
```

```
mon3.example.com. AAAA 2001:db8::300
```

ここで、**example.com** は DNS 検索ドメインになります。

次に、3つのモニターを指す **mon_dns_srv_name** 設定名で SRV TCP レコードを作成します。以下の例では、デフォルトの **ceph-mon** 値を使用しています。

例

```
_ceph-mon._tcp.example.com. 60 IN SRV 10 60 6789 mon1.example.com.
```

```
_ceph-mon._tcp.example.com. 60 IN SRV 10 60 6789 mon2.example.com.
```

```
_ceph-mon._tcp.example.com. 60 IN SRV 10 60 6789 mon3.example.com.
```

```
_ceph-mon._tcp.example.com. 60 IN SRV 10 60 3300 mon1.example.com.
```

```
_ceph-mon._tcp.example.com. 60 IN SRV 10 60 3300 mon2.example.com.
```

```
_ceph-mon._tcp.example.com. 60 IN SRV 10 60 3300 mon3.example.com.
```

モニターはデフォルトでポート **6789** と **3300** で稼働し、その優先度と重みはすべて前述の例でそれぞれ **10** および **60** に設定されます。

3.9. CEPH の一意の識別子

各 Red Hat Ceph Storage クラスターには固有の ID (**fsid**) があります。指定した場合には、通常は設定ファイルの **[global]** セクションに表示されます。デプロイメントツールは通常、**fsid** を生成してモニターマップに保存するため、値は設定ファイルに表示されない可能性があります。**fsid** を使用すると、同じハードウェア上で複数のクラスターに対してデーモンを実行できます。



注記

値を設定するデプロイメントツールを使用している場合は、この値を設定しないでください。

3.10. CEPH MONITOR のデータストア

Ceph では、Ceph モニターがデータを保存するデフォルトのパスが用意されています。

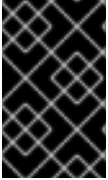


重要

Red Hat では、本番環境の Red Hat Ceph Storage クラスターで最適なパフォーマンスを得るために、Ceph モニターを Ceph OSD とは別のホストとドライブで実行することをお勧めします。

Ceph モニターは **fsync()** 関数を頻繁に呼び出します。これは、Ceph OSD ワークロードに干渉する可能性があります。

Ceph モニターは、データをキー/値ペアとして保存します。データストアを使用すると、他のメリットに加えて、復旧中の Ceph モニターが Paxos と通じて破損したバージョンを実行することを防ぎ、1つのアトミックバッチで複数の修正操作が可能になります。

**重要**

Red Hat はデフォルトのデータの場所を変更することを推奨しません。デフォルトの場所を変更する場合は、設定ファイルの **[mon]** セクションにそれを設定して、Ceph モニター全体で統一します。

**重要**

モニターデータベースを保存するディスクは、暗号化する必要があります。詳細は、[LUKS disk encryption](#) を参照してください。

3.11. CEPH ストレージの容量

Red Hat Ceph Storage クラスターが最大容量 (**mon_osd_full_ratio** パラメーターにより指定) に近くなると、データの損失を防ぐために安全対策として Ceph OSD への書き込みができなくなります。そのため、本番環境の Red Hat Ceph Storage クラスターをそのフル比率に近づけてしまうことは、高可用性が犠牲になってしまうのでグッドプラクティスとは言えません。デフォルトのフル比率は、**.95** (容量の 95%) です。これは、OSD の数が少ないテストクラスター用の非常に厳しい設定です。

ヒント

クラスターをモニタリングする際に、**nearfull** な比率に関連する警告にアラートしてください。つまり、1つまたは複数の OSD が故障した場合、一部の OSD の障害により一時的にサービスが中断される可能性があります。ストレージの容量を増やすために、OSD の増設を検討してください。

テストクラスターの一般的なシナリオでは、システム管理者が Red Hat Ceph Storage クラスターから Ceph OSD を削除してクラスターの再バランスを観察します。その後、別の Ceph OSD を削除し、Red Hat Ceph Storage クラスターが最終的にフル比率に達してロックアップするまでこれを繰り返します。

**重要**

Red Hat では、テストクラスターであっても、多少の容量計画を立てることを推奨しています。計画を立てることで、高可用性を維持するためにどれだけの予備容量が必要なのかを把握することができます。

理想的には、Ceph OSD を直ちに置き換えることなく、クラスターが **active + clean** な状態に復元できる Ceph OSD の一連の障害を計画する必要があります。クラスターを **active + degraded** の状態で実行できますが、これは通常の動作条件には理想的ではありません。

次の図は、33 台の Ceph Node が含まれる単純化した Red Hat Ceph Storage クラスターを示しています。ホストごとに1つの Ceph OSD があり、各 Ceph OSD デーモンは 3 TB のドライブに対して読み取りおよび書き込みを行います。つまり、この例の Red Hat Ceph Storage クラスターの最大実容量は 99 TB です。**mon_osd_full_ratio** が **0.95** の場合は、Red Hat Ceph Storage クラスターが空き容量が 5 TB になると、Ceph クライアントはデータの読み取りと書き込みを許可しません。そのため、Red Hat Ceph Storage クラスターの運用上の容量は 99 TB ではなく 95 TB となります。



110_Ceph_0720

このようなクラスターでは、1つまたは2つの OSD が故障するのが普通です。頻度は低いが妥当なシナリオとしては、ラックのルーターや電源が故障し、複数の OSD が同時にダウンすることが挙げられます (例: OSD 7-12)。このようなシナリオでは、さらに OSD のあるホストを短い順序で追加する場合でも、動作し続け、**active + clean** な状態を実現するクラスターを試す必要があります。容量利用率が高すぎると、データを失うことはないかもしれませんが、クラスターの容量利用率がフル比率を超えた場合、障害ドメイン内の障害を解決している間データの可用性が犠牲になる可能性があります。このため、Red Hat では、少なくとも大まかな容量計画を立てることを推奨しています。

クラスターに関する 2 つの数字を把握します。

- OSD の数
- クラスターの総容量

クラスター内の OSD の平均容量を求めるには、クラスターの総容量をクラスター内の OSD の数で割ります。この数に、通常の運用で同時に故障すると予想される OSD の数 (比較的小さい数) を乗じます。最後に、クラスターの容量にフル比率を掛けて、運用上の最大容量を算出します。そして、失敗すると予想される OSD からデータ量を差し引いて、合理的なフル比率を算出します。前述のプロセスを、より多くの OSD 故障数 (例えば、OSD のラック) で繰り返し、ほぼフル比率のための妥当な数を算出します。

3.12. CEPH ハートビート

Ceph モニターは、各 OSD からのレポートを要求し、隣接する OSD の状態に関するレポートを OSD から受け取ることで、クラスターについて把握します。Ceph では、モニターと OSD の間の相互作用について妥当なデフォルト設定が用意されていますが、必要に応じて変更することができます。

3.13. CEPH MONITOR の同期ロール

複数のモニターを持つ本番環境用のクラスターを運用する場合 (推奨される設定)、各モニターは隣接するモニターがより新しいバージョンのクラスターマップを持っているかどうかを確認します。例えば、隣接するモニターのマップのエポックナンバーが、インスタントモニターのマップの最新のエポックよ

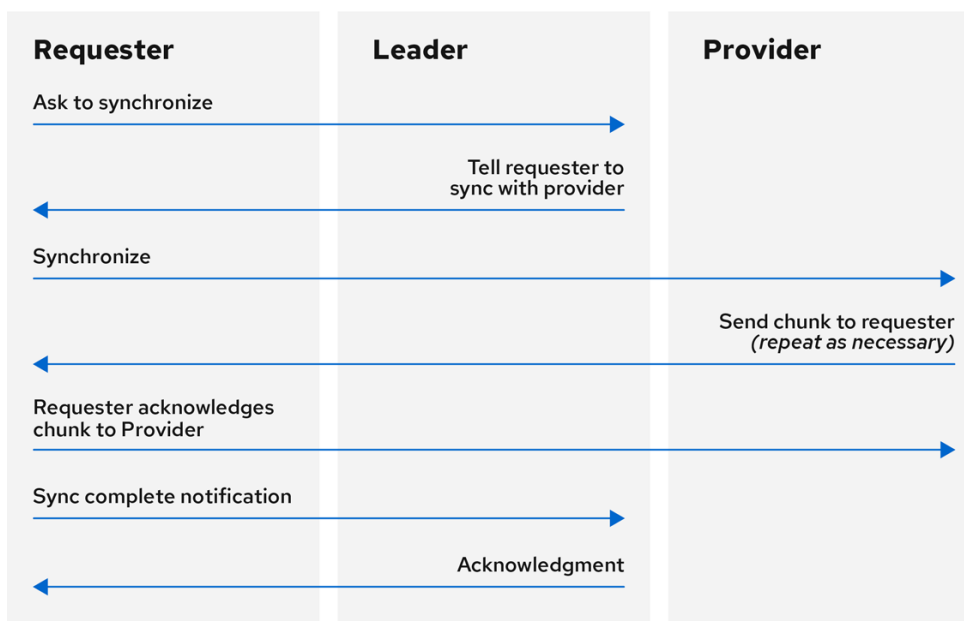
り1つ以上高い場合。定期的に、クラスター内のあるモニターが他のモニターから遅れをとることがあります。その場合、そのモニターはクォーラムから離脱し、同期をとってクラスターに関する最新の情報を取得した後、再びクォーラムに参加しなければなりません。

同期ルール

同期のために、モニターは以下の3つのロールのいずれかを取ります。

- **リーダー**: リーダーは、クラスターマップの最新の Paxos バージョンを実現する最初のモニターです。
- **プロバイダー**: プロバイダーは最新バージョンのクラスターマップを持つモニターですが、最新バージョンを最初に達成したわけではありません。
- **リクエスター**: リクエスターはリーダーの背後に置かれたモニターで、クォーラムに再参加する前にクラスターに関する最新情報を取得するために同期する必要があります。

これらのロールにより、リーダーは同期のタスクをプロバイダーに委譲することができ、同期の要求によりリーダーが過負荷になることを防ぎ、パフォーマンスが向上します。次の図では、リクエスターが他のモニターに遅れをとっていることを認識しています。リクエスターはリーダーに同期を依頼し、リーダーはリクエスターにプロバイダーとの同期を指示します。



110_Ceph_0720

モニターの同期

新しいモニターがクラスターに参加すると、常に同期が行われます。実行時の運用において、モニターは異なるタイミングでクラスターマップへの更新を受け取る場合があります。つまり、リーダーとプロバイダーのロールが、モニター間で移動する可能性があるということです。例えば、同期中にこれが起こると、プロバイダーはリーダーから遅れてしまい、プロバイダーはリクエスターとの同期を終了することができます。

同期が完了すると、Ceph ではクラスター全体のトリミングが必要になります。トリミングを行うには、配置グループが **active + clean** である必要があります。

3.14. CEPH の時刻同期

Ceph デーモンは、クリティカルなメッセージを相互に渡します。このメッセージは、デーモンがタイムアウトのしきい値に達する前に処理する必要があります。Ceph モニターのクロックが同期していないと、さまざまな異常が発生する可能性があります。

以下に例を示します。

- デーモンが受信したメッセージを無視する (タイムスタンプが古いなど)。
- メッセージ受信のタイミングが適切でない場合、タイムアウトの発生が早すぎたり遅すぎたりする。

ヒント

Ceph モニターホストに NTP をインストールして、モニタークラスターのクロックが同期した状態で動作するようにします。

NTP では、遅れによる悪影響が出ていなくても、クロックドリフトが目立つことがあります。NTP が適切なレベルの同期を維持していても、Ceph のクロックドリフトとクロックスキューの警告が発生することがあります。このような状況では、クロックドリフトを増やすことが許容できるかもしれませんが、しかし、ワークロード、ネットワークレイテンシー、デフォルトのタイムアウトに対するオーバーライド設定、およびその他の同期オプションなど、多くの要因が、Paxos の保証を損なうことなく許容できるクロックドリフトのレベルに影響を与えます。

関連情報

- 詳細は、[Ceph の時刻同期](#) セクションを参照してください。

3.15. 関連情報

- 特定のオプションの説明や使用方法は、[付録 C](#) に記載されるすべての Red Hat Ceph Storage Monitor 設定オプションを参照してください。

第4章 CEPH の認証設定

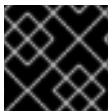
ストレージ管理者として、ユーザーとサービスを認証することは、Red Hat Ceph Storage クラスターのセキュリティーにとって重要です。Red Hat Ceph Storage には、デフォルトで暗号認証用の Cephx プロトコルと、ストレージクラスターで認証を管理するツールが含まれています。

4.1. 前提条件

- Red Hat Ceph Storage ソフトウェアのインストール

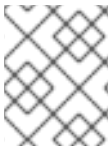
4.2. CEPHX 認証

cephx プロトコルはデフォルトで有効になっています。暗号認証には多少の計算コストがかかりますが、一般的には非常に低いものです。クライアントとホストを結ぶネットワーク環境が安全と考えられ、認証の計算コストがかかけられない場合は、無効にすることができます。Ceph Storage クラスターをデプロイする際に、デプロイメントツールは **client.admin** ユーザーおよびキーリングを作成します。



重要

Red Hat では認証の使用を推奨しています。



注記

認証を無効にすると、中間者攻撃によってクライアントとサーバーのメッセージが改ざんされる危険性があり、重大なセキュリティー問題に発展する可能性があります。

Cephx の有効化と無効化

Cephx を有効にするには、Ceph Monitor と OSD 用のキーをデプロイする必要があります。Cephx 認証のオン/オフを切り替える場合は、デプロイメント手順を繰り返す必要はありません。

4.3. CEPHX の有効化

cephx が有効な場合には、Ceph はデフォルトの検索パス **/etc/ceph/\$cluster.\$name.keyring** を含む) でキーリングを探します。Ceph 設定ファイルの **[global]** セクションに **keyring** オプションを追加することで、この場所を上書きすることができますが、これは推奨されません。

認証が無効になっているクラスターで **cephx** を有効にするには、以下の手順を実行します。ご自身またはデプロイメントユーティリティーがすでにキーを生成している場合は、キーの生成に関する手順を省略できます。

前提条件

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

手順

1. **client.admin** キーを作成し、クライアントホストのキーのコピーを保存します。

```
[root@mon ~]# ceph auth get-or-create client.admin mon 'allow *' osd 'allow *' -o
/etc/ceph/ceph.client.admin.keyring
```



警告

これにより、既存の **/etc/ceph/client.admin.keyring** ファイルの内容が消去されます。すでにデプロイメントツールがこの作業を行っている場合は、この手順を実行しないでください。

2. モニタークラスター用のキーリングを作成し、モニターシークレットキーを生成します。

```
[root@mon ~]# ceph-authtool --create-keyring /tmp/ceph.mon.keyring --gen-key -n mon. --
cap mon 'allow *'
```

3. すべてのモニターの **mon data** ディレクトリーの **ceph.mon.keyring** ファイルにモニターキーリングをコピーします。たとえば、これをクラスター **ceph** の **mon.a** にコピーするには、以下のコマンドを使用します。

```
[root@mon ~]# cp /tmp/ceph.mon.keyring /var/lib/ceph/mon/ceph-a/keyring
```

4. すべての OSD に秘密鍵を生成します。ここで、**ID** は OSD 番号です。

```
ceph auth get-or-create osd.ID mon 'allow rwx' osd 'allow *' -o
/var/lib/ceph/osd/ceph-ID/keyring
```

5. デフォルトでは、**cephx** 認証プロトコルは有効になっています。



注記

認証オプションを **none** に設定して **cephx** 認証プロトコルが無効にされていた場合には、Ceph 設定ファイル (**/etc/ceph/ceph.conf**) の **[global]** セクションの下にある以下の行を削除して、**cephx** 認証プロトコルを再度有効にします。

```
auth_cluster_required = none
auth_service_required = none
auth_client_required = none
```

6. Ceph Storage クラスターを起動または再起動します。

重要

cephx を有効にするには、クラスターを完全に再起動する必要があるか、クライアントの I/O が無効になったときにシャットダウンしてから起動する必要があるため、ダウンタイムが必要です。

これらのフラグは、ストレージクラスターを再起動またはシャットダウンする前に設定する必要があります。

```
[root@mon ~]# ceph osd set noout
[root@mon ~]# ceph osd set norecover
[root@mon ~]# ceph osd set norebalance
[root@mon ~]# ceph osd set nobackfill
[root@mon ~]# ceph osd set nodown
[root@mon ~]# ceph osd set pause
```

cephx が有効になり、すべての PG がアクティブかつクリーンな状態になったら、フラグの設定を解除します。

```
[root@mon ~]# ceph osd unset noout
[root@mon ~]# ceph osd unset norecover
[root@mon ~]# ceph osd unset norebalance
[root@mon ~]# ceph osd unset nobackfill
[root@mon ~]# ceph osd unset nodown
[root@mon ~]# ceph osd unset pause
```

4.4. CEPHX の無効化

以下の手順では、Cephx を無効にする方法を説明します。クラスター環境が比較的安全であれば、認証を実行するための計算コストを相殺することができます。

重要

Red Hat では認証を有効にすることを推奨しています。

しかし、セットアップやトラブルシューティングの際には、一時的に認証を無効にした方が簡単な場合もあります。

前提条件

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

手順

- Ceph 設定ファイルの **[global]** セクションに以下のオプションを設定して、**cephx** 認証を無効にします。

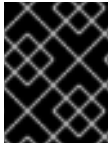
```
auth_cluster_required = none
auth_service_required = none
auth_client_required = none
```

2. Ceph Storage クラスターを起動または再起動します。

4.5. CEPHX ユーザーキーリング

認証が有効な Ceph を実行する場合には、Ceph Storage クラスターにアクセスするために **ceph** 管理コマンドおよび Ceph クライアントに認証キーが必要です。

ceph 管理コマンドおよびクライアントにこれらの鍵を提供する最も一般的な方法は、**/etc/ceph/** ディレクトリーの下に Ceph キーリングを追加することです。ファイル名は通常 **ceph.client.admin.keyring** または **\$cluster.client.admin.keyring** です。**/etc/ceph/** ディレクトリーにキーリングを含める場合は、Ceph 設定ファイルで **keyring** エントリーを指定する必要はありません。



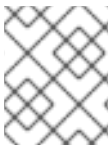
重要

Red Hat は、**client.admin** キーが含まれるため、Red Hat Ceph Storage クラスターのキーリングファイルを管理コマンドを実行するノードにコピーすることを推奨します。

それを行うには、以下のコマンドを実行します。

```
# scp USER@HOSTNAME:/etc/ceph/ceph.client.admin.keyring /etc/ceph/ceph.client.admin.keyring
```

USER を、ホストで使用されるユーザー名に **client.admin** キーを使用し、**HOSTNAME** をそのホストのホスト名に置き換えます。



注記

ceph.keyring ファイルに、クライアントマシンに適切なパーミッションが設定されていることを確認します。

推奨されていない **key** 設定を使用して、Ceph 設定ファイルにキー自体を指定したり、**keyfile** 設定を使用してキーファイルへのパスを指定することができます。

4.6. CEPHX デーモンのキーリング

管理ユーザーやデプロイメントツールは、ユーザーキーリングの生成と同じ方法で、デーモンキーリングを生成することがあります。デフォルトでは、Ceph はデーモンのキーリングをデータディレクトリー内に保存します。デフォルトのキーリングの場所や、デーモンが機能するために必要な機能など。



注記

モニターキーリングにはキーが含まれていますが、機能はなく、Ceph Storage クラスターの **auth** データベースの一部ではありません。

デーモンデータのディレクトリーの位置は、デフォルトでは以下の形式のディレクトリーになります。

```
/var/lib/ceph/$type/CLUSTER-ID
```

例

```
/var/lib/ceph/osd/ceph-12
```


これらの場所を上書きすることもできますが、お勧めできません。

4.7. CEPHX イメージの署名

Ceph にはきめ細かな制御機能があり、クライアントと Ceph の間のサービスメッセージの署名を有効または無効にすることができます。Ceph デーモン間のメッセージに対する署名を有効または無効にすることができます。



重要

Red Hat では、最初の認証のために設定されたセッションキーを使用して、Ceph がエンティティー間のすべての進行中のメッセージを認証することを推奨しています。



注記

Ceph のカーネルモジュールは、まだ署名をサポートしていません。

4.8. 関連情報

- 特定のオプションの説明や使用方法は、[付録 D](#) の Red Hat Ceph Storage Cephx のすべての設定オプションを参照してください。

第5章 プール、配置グループ、および CRUSH の設定

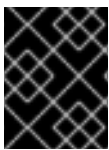
ストレージ管理者として、プール、配置グループ、および CRUSH アルゴリズムに Red Hat Ceph Storage のデフォルトオプションを使用するか、目的のワークロードに合わせてカスタマイズするかを選択することができます。

5.1. 前提条件

- Red Hat Ceph Storage ソフトウェアのインストール

5.2. プール、配置グループ、および CRUSH

プールを作成し、プールの配置グループの数を設定するとき、特にデフォルトをオーバーライドしない場合、Ceph はデフォルト値を使用します。



重要

Red Hat では、いくつかのデフォルトを上書きすることをお勧めします。具体的には、プールのレプリカサイズを設定し、デフォルトの配置グループ数を上書きします。

これらの値は、pool コマンドの実行時に設定できます。Ceph 設定ファイルの **[global]** セクションに新規のものを追加して、デフォルト値を上書きすることもできます。

例

```
[global]

# By default, Ceph makes 3 replicas of objects. If you want to set 4
# copies of an object as the default value--a primary copy and three replica
# copies--reset the default values as shown in 'osd pool default size'.
# If you want to allow Ceph to write a lesser number of copies in a degraded
# state, set 'osd pool default min size' to a number less than the
# 'osd pool default size' value.

osd_pool_default_size = 4 # Write an object 4 times.
osd_pool_default_min_size = 1 # Allow writing one copy in a degraded state.

# Ensure you have a realistic number of placement groups. We recommend
# approximately 100 per OSD. E.g., total number of OSDs multiplied by 100
# divided by the number of replicas (i.e., osd pool default size). So for
# 10 OSDs and osd pool default size = 4, we'd recommend approximately
# (100 * 10) / 4 = 250.

osd_pool_default_pg_num = 250
osd_pool_default_pgp_num = 250
```

5.3. 関連情報

- 特定のオプションの説明と使用方法は、[付録 E](#) の Red Hat Ceph Storage プール、配置グループ、および CRUSH 設定オプションをすべて参照してください。

第6章 CEPH OBJECT STORAGE DAEMON (OSD) の設定

ストレージ管理者として、Ceph Object Storage Daemon (OSD) を設定して、意図するワークロードに基づいて冗長化と最適化を行うことができます。

6.1. 前提条件

- Red Hat Ceph Storage ソフトウェアのインストール

6.2. CEPH OSD の設定

すべての Ceph クラスターには、以下の項目を定義する設定があります。

- クラスター ID
- 認証設定
- クラスター内の Ceph daemon のメンバーシップ
- ネットワーク設定
- ホスト名およびアドレス
- キーリングへのパス
- OSD ログファイルへのパス
- 他のランタイムオプション

Red Hat Ceph Storage Console または Ansible などのデプロイメントツールは、通常、初期の Ceph 設定ファイルを作成します。ただし、デプロイメントツールを使用してクラスターをブートストラップする場合には、独自に作成することができます。

便宜上、各デーモンにはデフォルト値のセットがあります。その多くは **ceph/src/common/config_opts.h** スクリプトで設定されます。これらの設定は、Ceph 設定ファイル、またはランタイム時にモニターの **tell** コマンドを使用するか、Ceph ノード上のデーモンソケットに直接接続して上書きできます。



重要

Red Hat では、Ceph を後でトラブルシューティングする際に問題が生じるため、デフォルトのパスを変更することは推奨していません。

6.3. OSD のスクラブ

Ceph は、オブジェクトの複数のコピーを作成するだけでなく、配置グループをスクラビングすることでデータの整合性を確保します。Ceph のスクラブは、オブジェクトストレージ層の **fsck** コマンドに似ています。

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atalogを生成し、各プライマリー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。

ライトスクラビング (毎日) では、オブジェクトのサイズや属性をチェックします。ディープスクラビング (毎週) は、データを読み込んでチェックサムでデータの整合性を確保します。

スクラビングはデータの整合性を保つために重要ですが、パフォーマンスを低下させる可能性があります。以下の設定を調整して、スクラブ動作を増減させます。

関連情報

- 特定のオプションの説明や使用方法は、[付録 I](#) の Red Hat Ceph Storage Ceph スクラブオプションを参照してください。

6.4. OSD のバックフィル

Ceph OSD をクラスターに追加したり、クラスターから削除したりすると、CRUSH アルゴリズムは、配置グループを Ceph OSD に移動させたり、Ceph OSD から移動させたりしてバランスを回復させ、クラスターのバランスを取り戻します。配置グループとそれに含まれるオブジェクトを移行するプロセスは、クラスターの運用パフォーマンスを大幅に低下させます。運用パフォーマンスを維持するために、Ceph はこの移行をバックフィルプロセスで実行します。これにより、Ceph はバックフィル操作をデータの読み取りまたは書き込みの要求よりも低い優先度に設定できます。

6.5. OSD リカバリー

クラスターが起動したとき、または Ceph OSD が予期せず終了して再起動したとき、OSD は書き込み操作を行う前に他の Ceph OSD とのピアリングを開始します。

Ceph OSD がクラッシュしてオンラインに戻ると、通常、配置グループのオブジェクトのより新しいバージョンが含まれる他の Ceph OSD との同期が取れなくなります。このような場合、Ceph OSD はリカバリーモードに入り、データの最新コピーを取得してマップを最新の状態に戻そうとします。Ceph OSD が停止していた時間によっては、OSD のオブジェクトや配置グループが大幅に古くなっている可能性があります。また、障害ドメイン (例: ラックなど) ダウンした場合、複数の Ceph OSD が同時にオンラインに戻る可能性があります。そのため、復旧作業には時間とリソースが必要になります。

運用パフォーマンスを維持するために、Ceph はリカバリー要求数、スレッド数、およびオブジェクトチャンクサイズを制限してリカバリーを実行し、これにより Ceph は劣化した状態でも適切なパフォーマンスを発揮することができます。

6.6. 関連情報

- 特定のオプションの説明や使用方法は、[付録 F](#) の Red Hat Ceph Storage Ceph OSD 設定オプションをすべて参照してください。

第7章 CEPH MONITOR と OSD の連動設定

ストレージ管理者としては、安定した動作環境を確保するために、Ceph Monitor と OSD の相互作用を適切に設定する必要があります。

7.1. 前提条件

- Red Hat Ceph Storage ソフトウェアのインストール

7.2. CEPH MONITOR と OSD の連動

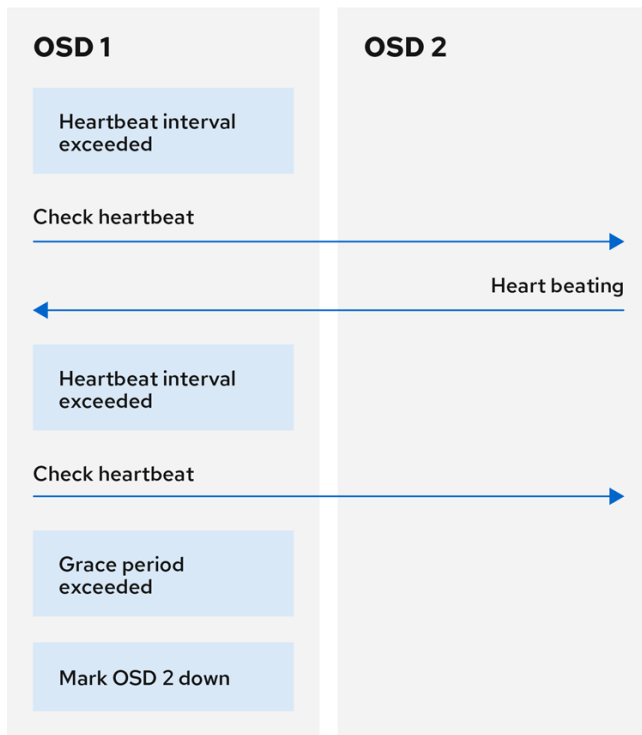
Ceph の初期設定が完了したら、Ceph をデプロイして実行することができます。**ceph health**、**ceph -s** などのコマンドを実行すると、Ceph Monitor は Ceph Storage クラスターの現在の状態を報告します。Ceph Monitor は、各 Ceph OSD デーモンからのレポートを要求し、隣接する Ceph OSD デーモンの状態に関するレポートを Ceph OSD デーモンから受け取ることで、Ceph ストレージクラスターについて把握します。Ceph Monitor がレポートを受信しない場合、または Ceph ストレージクラスターの変更のレポートを受信した場合、Ceph Monitor は Ceph クラスターマップのステータスを更新します。

Ceph では、Ceph Monitor と OSD の連携について妥当なデフォルト設定が用意されています。ただし、デフォルト値を上書きできます。以下のセクションでは、Ceph ストレージクラスターを監視する目的で、Ceph Monitor と Ceph OSD デーモンがどのように相互作用するかを説明します。

7.3. OSD ハートビート

各 Ceph OSD デーモンは、6 秒ごとに他の Ceph OSD デーモンのハートビートをチェックします。ハートビートの間隔を変更するには、Ceph 設定ファイルの **[osd]** セクションに **osd heartbeat interval** 設定を追加するか、ランタイム時にその値を変更します。

近傍の Ceph OSD デーモンが 20 秒の猶予期間内にハートビートパケットを送信しない場合、Ceph OSD デーモンは近傍の Ceph OSD デーモンが **down** であるとみなされる可能性があります。それを Ceph Monitor に報告して、Ceph クラスターマップを更新することができます。この猶予期間を変更するには、Ceph 設定ファイルの **[osd]** セクションに **osd heartbeat grace** 設定を追加するか、ランタイム時にその値を設定します。



110_Ceph_0720

7.4. OSD がダウンであることの報告

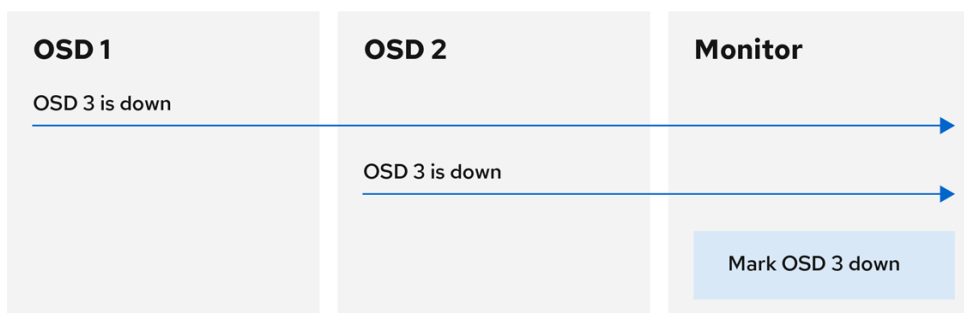
デフォルトでは、異なるホストの2つの Ceph OSD デーモンは、報告された Ceph OSD デーモンが **down** していることを Ceph モニターが確認する前に、別の Ceph OSD デーモンが **down** していることを Ceph モニターに報告する必要があります。

しかし、障害を報告するすべての OSD が、ラック内の異なるホストに設置されており、スイッチ不良により OSD 間の接続に問題が生じる場合があります。

誤報を避けるために、Ceph は障害を報告したピアを、同様に遅延しているサブクラスターの代理として考えます。これは必ずしもそうとは限りませんが、管理者が、パフォーマンスの低下しているシステムのサブセットに局所的に適切な補正を適用するのに役立つ場合があります。

Ceph は `mon_osd_reporter_subtree_level` 設定を使用して、CRUSH マップの共通の先復元タイプでピアを subcluster にグループ化します。

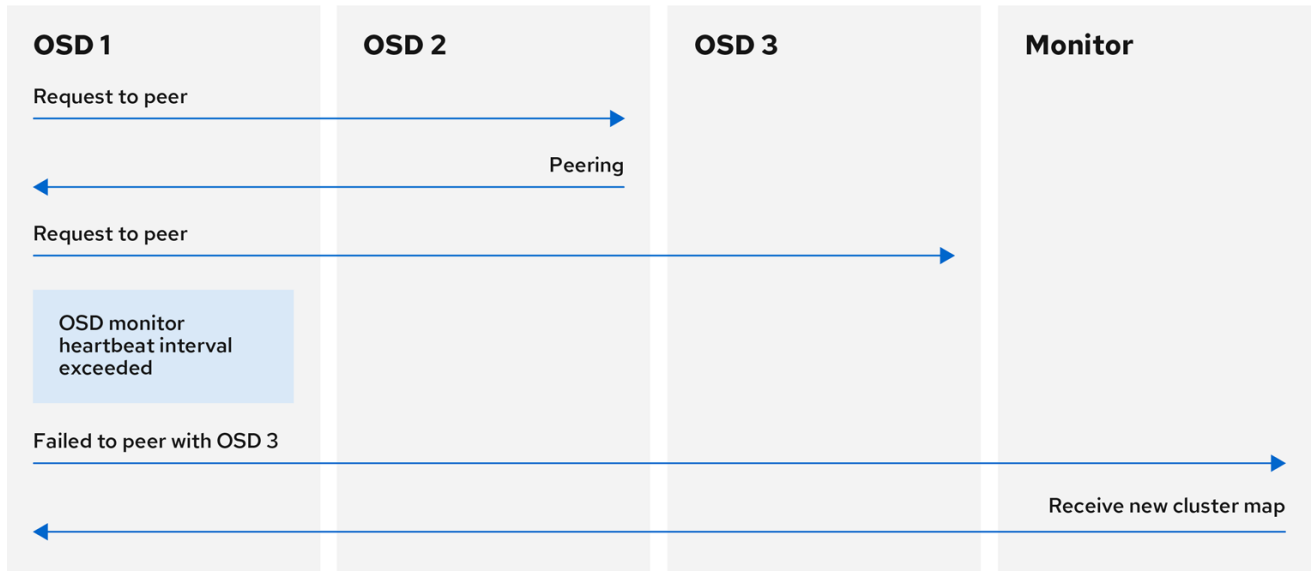
デフォルトでは、異なるサブツリーからわずか2つのレポートは、他の Ceph OSD デーモン **down** を報告する必要があります。管理者は、Ceph 設定ファイルの `[mon]` セクションの下で、`mon_osd_min_down_reporters` 設定および `mon_osd_reporter_subtree_level` 設定を追加するか、ランタイム時に値を設定することで、Ceph Monitor に Ceph OSD Daemon **down** を報告するために必要な固有のサブツリーと共通の祖先型からレポーターの数を変更することができます。



110_Ceph_0720

7.5. ピアリングの失敗の報告

Ceph OSD デーモンが、その Ceph 設定ファイルまたはクラスターマップで定義された Ceph OSD デーモンのいずれともピアリングできない場合、30 秒ごとにクラスターマップの最新コピーを求めて Ceph Monitor に ping を実行します。Ceph Monitor ハートビートの間隔は、Ceph 設定ファイルの **[osd]** セクションに **osd mon heartbeat interval** 設定を追加するか、ランタイムに値を設定して変更できます。

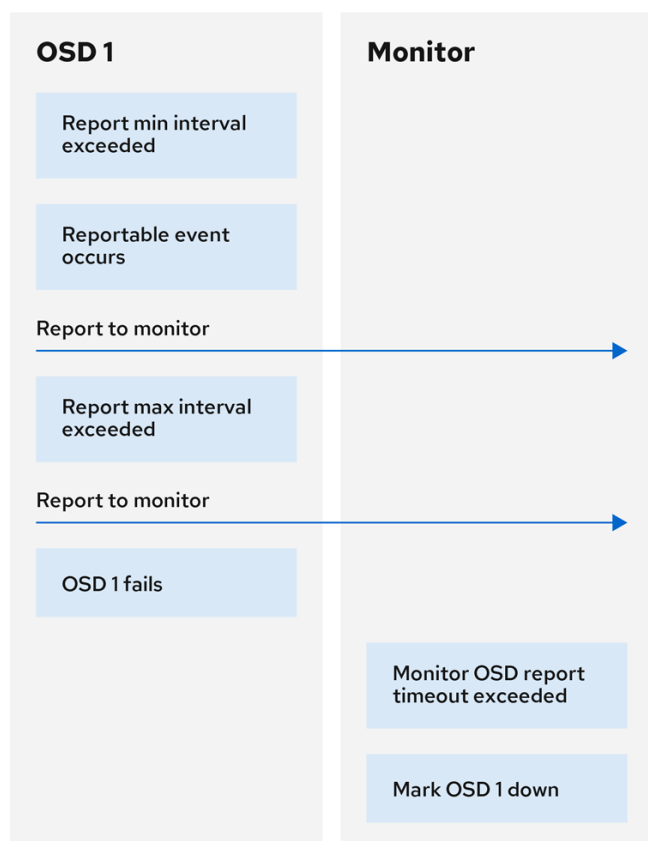


ll0_Ceph_0720

7.6. OSD の報告状況

Ceph OSD デーモンが Ceph Monitor に報告しない場合、Ceph Monitor は **mon osd report timeout** 後に **down** した Ceph OSD デーモンを考慮します。Ceph OSD デーモンは、障害、配置グループ統計の変更、**up_thru** の変更、または 5 秒以内にブートするなどの報告可能なイベント時に、Ceph Monitor にレポートを送信します。Ceph OSD Daemon の最小レポート間隔を変更するには、Ceph 設定ファイルの **[osd]** セクションに **osd mon report interval min** 設定を追加するか、ランタイムに値を設定します。

Ceph OSD デーモンは、目立った変更があったかどうかにかかわらず、120 秒ごとに Ceph Monitor にレポートを送信します。Ceph Monitor のレポート間隔を変更するには、Ceph 設定ファイルの **[osd]** セクションに **osd mon report interval max** 設定を追加するか、値をランタイムに設定します。



110_Ceph_0720

7.7. 関連情報

- 特定のオプションの説明や使用方法は、[付録 G](#) にあるすべての Red Hat Ceph Storage Ceph Monitor および OSD 設定オプションを参照してください。

第8章 CEPH のデバッグとロギングの設定

ストレージ管理者として、デバッグとログ情報の量を増やして、Red Hat Ceph Storage の問題を診断するのに役立てることができます。

8.1. 前提条件

- Red Hat Ceph Storage ソフトウェアのインストール

8.2. CEPH のデバッグとロギング

デバッグ設定は Ceph 設定ファイルでは必要ありませんが、ロギングを最適化するために追加することができます。Ceph のロギング設定の変更は通常、問題が発生した場合にランタイム時に発生しますが、Ceph 設定ファイルで変更することもできます。たとえば、クラスターの起動時に問題が発生した場合には、Ceph 設定ファイルでログ設定を増やすことを検討してください。問題が解決されたら、設定を削除したり、ランタイム操作に最適な設定を復元したりします。

デフォルトでは、`/var/log/ceph` の下にある Ceph ログファイルを表示します。

ヒント

デバッグ出力によりクラスターが遅くなる場合、レイテンシーで競合状態が非表示になる可能性があります。

ロギングはリソース集約型です。クラスターの特定のエリアに問題がある場合は、クラスターのそのエリアのロギングを有効にします。たとえば、OSD は問題なく動作しているが、Ceph Object Gateway に問題がある場合は、問題が発生している特定のゲートウェイインスタンスのデバッグロギングを有効にして開始します。必要に応じて、各サブシステムのロギングを増減します。



重要

詳細なロギングは、1時間あたり1GB を超えるデータを生成することがあります。OS ディスクが容量に達すると、ノードが機能しなくなります。

Ceph ロギングが有効またはロギングレートを増やす場合は、OS ディスクの容量が十分にあることを確認してください。

クラスターが正常に実行されている場合は、不要なデバッグ設定を削除して、クラスターが最適に実行されるようにします。デバッグ出力メッセージのロギングは比較的遅くなります。また、クラスターの操作時にリソースが無駄になります。

8.3. 関連情報

- 特定のオプションの説明や使用方法は、[付録 J](#) の Red Hat Ceph Storage Ceph のデバッグおよびログの設定オプションをすべて参照してください。

付録A 一般的な設定オプション

Ceph の一般的な設定オプションを以下に示します。



注記

通常は、Ansible などのデプロイメントツールによって自動的に設定されます。

fsid

詳細

ファイルシステム ID です。クラスターごとに1つになります。

型

UUID

必須

いいえ

デフォルト

該当なし。通常、デプロイメントツールによって生成されます。

admin_socket

詳細

Ceph モニターがクォーラムを確立しているかどうかにかかわらず、デーモンの管理コマンドを実行するためのソケット

型

文字列

必須

いいえ

デフォルト

`/var/run/ceph/$cluster-$name.asok`

pid_file

詳細

モニターや OSD が自分の PID を書き込むためのファイル。たとえば、`/var/run/$cluster/$type.$id.pid` は、**ceph** クラスターで実行している id **a** を持つ **mon** の `/var/run/ceph/mon.a.pid` を作成します。**pid file** は、デーモンが正常に停止すると削除されます。プロセスがデーモン化されていない場合 (つまり、**-f** オプションまたは **-d** オプションで実行)、**pid file** は作成されません。

型

文字列

必須

いいえ

デフォルト

いいえ

chdir

詳細

Ceph デーモンが起動してから変更するディレクトリー。デフォルトの / ディレクトリーが推奨されます。

型

文字列

必須

いいえ

デフォルト

/

max_open_files**詳細**

これが設定されている場合には、Red Hat Ceph Storage クラスターが起動すると Ceph は OS レベルで **max_open_fds** を設定します (つまりファイル記述子の最大数 #)。これにより、Ceph OSD がファイル記述子を使い果たすのを防ぐことができます。

型

64 ビット整数

必須

いいえ

デフォルト

0

fatal_signal_handlers**詳細**

設定されていると、SEGV、ABRT、BUS、ILL、FPE、XCPU、XFSZ、SYS シグナルのシグナルハンドラーをインストールして、有用なログメッセージを生成します。

型

ブール値

デフォルト

true

付録B CEPH のネットワーク設定オプション

Ceph の共通的なネットワーク設定オプションを以下に示します。

public_network

詳細

パブリック (フロントエンド) ネットワークの IP アドレスとネットマスク (例: **192.168.0.0/24**)。[global] に設定します。コンマ区切りのサブネットを指定できます。

型

<ip-address>/<netmask> [, <ip-address>/<netmask>]

必須

いいえ

デフォルト

該当なし

public_addr

詳細

パブリック (フロントサイド) ネットワークの IP アドレスです。各デーモンのセット。

型

IP アドレス

必須

いいえ

デフォルト

該当なし

cluster_network

詳細

クラスターネットワークの IP アドレスとネットマスク (例: **10.0.0.0/24**)。[global] に設定します。コンマ区切りのサブネットを指定できます。

型

<ip-address>/<netmask> [, <ip-address>/<netmask>]

必須

いいえ

デフォルト

該当なし

cluster_addr

詳細

クラスターネットワークの IP アドレスです。各デーモンのセット。

型

アドレス

必須

いいえ

デフォルト

該当なし

ms_type

詳細

ネットワークトランスポート層のメッセージャータイプです。Red Hat は、**posix** セマンティクスを使用した、messenger タイプ **simple** および **async** をサポートします。

型

文字列。

必須

いいえ

デフォルト

async+posix

ms_public_type

詳細

パブリックネットワークのネットワークトランスポート層のメッセージャータイプです。これは **ms_type** と同じように動作しますが、パブリックネットワークまたはフロントエンドネットワークにのみ適用されます。この設定により、Ceph はパブリックまたはフロントエンドまたはバックサイドのネットワークに異なるメッセージャータイプを使用できます。

型

文字列。

必須

いいえ

デフォルト

なし。

ms_cluster_type

詳細

クラスターネットワークのネットワークトランスポート層のメッセージャータイプです。これは **ms_type** と同じように動作しますが、クラスターまたはバックサイドネットワークにのみ適用されます。この設定により、Ceph はパブリックまたはフロントエンドまたはバックサイドのネットワークに異なるメッセージャータイプを使用できます。

型

文字列。

必須

いいえ

デフォルト

なし。

ホストオプション

宣言された各モニターの下に **mon addr** 設定を指定して、Ceph 設定ファイル内で少なくとも1つの Ceph Monitor を宣言する必要があります。Ceph では、Ceph 設定ファイルの宣言されたモニター、メタデータサーバー、および OSD の下に **host** の設定が必要です。



重要

localhost は使用しないでください。完全修飾ドメイン名 (FQDN) ではなく、ノードの短縮名を使用してください。ノード名を取得するサードパーティーのデプロイメントシステムを使用する場合は、**host** の値を指定しないでください。

mon_addr

詳細

クライアントが Ceph モニターへの接続に使用できる **<hostname>:<port>** エントリーの一覧。設定していない場合には、Ceph は **[mon.*]** セクションを検索します。

型

文字列

必須

いいえ

デフォルト

該当なし

host

詳細

ホスト名です。この設定は、特定のデーモンインスタンス (**[osd.0]** など) に使用します。

型

文字列

必須

デーモンインスタンスの場合は Yes。

デフォルト

localhost

TCP オプション

Ceph はデフォルトで TCP バッファリングを無効にします。

ms_tcp_nodelay

詳細

Ceph は **ms_tcp_nodelay** を有効化して、各リクエストが即時に送信されます (バッファなし)。Nagle アルゴリズムを無効にすると、ネットワークのトラフィックが増加し、混雑の原因となります。小さいパケットが多数ある場合は、**ms_tcp_nodelay** を無効にしてみてください。ただし、通常はこれを無効にすると待ち時間が長くなることに注意してください。

型

ブール値

必須

いいえ

デフォルト

true

ms_tcp_rcvbuf

詳細

ネットワーク接続の受信側のソケットバッファのサイズです。デフォルトでは無効です。

型

32 ビット整数

必須

いいえ

デフォルト

0

ms_tcp_read_timeout

詳細

クライアントまたはデーモンが別の Ceph デーモンへの要求を行い、未使用の接続を解除しない場合、**tcp read timeout** は、指定した秒数後に接続をアイドル状態として定義します。

型

未署名の 64 ビット整数

必須

いいえ

デフォルト

900 15 分。

バインドオプション

バインドオプションは、Ceph OSD デーモンのデフォルトのポート範囲を設定します。デフォルトの範囲は **6800:7100** です。また、Ceph デーモンが IPv6 アドレスにバインドするように設定することもできます。



重要

ファイアウォールの設定で、設定したポート範囲を使用できることを確認してください。

ms_bind_port_min

詳細

OSD デーモンがバインドする最小のポート番号。

型

32 ビット整数

デフォルト

6800

必須

いいえ

ms_bind_port_max

詳細

OSD デーモンがバインドする最大のポート番号。

型

32 ビット整数

デフォルト

7300

必須

いいえ

ms_bind_ipv6

詳細

Ceph デーモンが IPv6 アドレスにバインドするように設定します。

型

ブール値

デフォルト

false

必須

いいえ

非同期型メッセージャーオプション

これらの Ceph messenger オプションは、**AsyncMessenger** の動作を設定します。

ms_async_transport_type

詳細

AsyncMessenger が使用するトランスポートタイプ。Red Hat は **posix** 設定をサポートしますが、現時点では **dpdk** 設定または **rdma** 設定をサポートしません。POSIX は標準的な TCP/IP ネットワークを使用しており、デフォルト値です。その他のトランスポートタイプは実験的なもので、サポートされて **いません**。

型

文字列

必須

いいえ

デフォルト

posix

ms_async_op_threads

詳細

各 **AsyncMessenger** インスタンスによって使用されるワーカースレッドの初期数。この設定は、レプリカまたはイレイジャーコードチャンクの数に等しく **なければならない** が、CPU コア数が低い場合や、単一のサーバー上での OSD の数が高い場合には低く設定することもできます。

型

64 ビット未署名の整数

必須

いいえ

デフォルト

3

ms_async_max_op_threads

詳細

各 **AsyncMessenger** インスタンスによって使用されるワーカースレッドの最大数。OSD ホストの CPU 数が制限されている場合は低い値に設定し、Ceph が CPU を十分に活用していない場合は高い値に設定します。

型

64 ビット未署名の整数

必須

いいえ

デフォルト

5

ms_async_set_affinity

詳細

AsyncMessenger ワーカーを特定の CPU コアにバインドするには、**true** に設定します。

型

ブール値

必須

いいえ

デフォルト

true

ms_async_affinity_cores

詳細

ms_async_set_affinity が **true** の場合、この文字列は **AsyncMessenger** ワーカーを CPU コアにバインドする方法を指定します。たとえば、**0,2** はそれぞれワーカー #1 と #2 を CPU コア #0 および #2 にバインドします。**注記:** アフィニティーを手動で設定する場合は、ハイパースレッディングや同様のテクノロジーが原因で作成された仮想 CPU にワーカーを割り当てないようにしてください。これは、物理 CPU コアよりも遅いためです。

型

文字列

必須

いいえ

デフォルト

(empty)

ms_async_send_inline

詳細

キューイングや **AsyncMessenger** スレッドから送信せずに、生成したスレッドからメッセージを直接送信します。このオプションは、CPU コア数の多いシステムではパフォーマンスが低下することが知られているため、デフォルトでは無効になっています。

型

ブール値

必須

いいえ

デフォルト

false

付録C CEPH MONITOR の設定オプション

デプロイメント時に設定可能な Ceph モニターの設定オプションを以下に示します。

mon_initial_members

詳細

起動時のクラスター内の最初のモニターの ID です。指定すると、Ceph は最初のクォーラムを形成するための奇数の数のモニターを必要とします (たとえば、3)。

型

文字列

デフォルト

なし

mon_force_quorum_join

詳細

過去にマップから削除されたモニターでも、強制的にクォーラムに参加させます。

型

ブール値

デフォルト

False

mon_dns_srv_name

詳細

モニターのホスト/アドレスを DNS にクエリーする際に使用するサービス名です。

型

文字列

デフォルト

ceph-mon

fsid

詳細

クラスター ID です。クラスターごとに1つになります。

型

UUID

必須

Yes

デフォルト

該当なし。指定されていない場合は、デプロイメントツールによって生成されます。

mon_data

詳細

モニターのデータの場所です。

型

文字列

デフォルト

`/var/lib/ceph/mon/$cluster-$id`

`mon_data_size_warn`

詳細

Ceph は、モニターのデータストアがこのしきい値に達すると、クラスターログで **HEALTH_WARN** ステータスを発行します。デフォルト値は 15GB です。

型

整数

デフォルト

`15*1024*1024*1024*`

`mon_data_avail_warn`

詳細

Ceph は、モニターのデータストアで利用可能なディスク領域がこの割合以下になると、クラスターログに **HEALTH_WARN** ステータスを発行します。

型

整数

デフォルト

`30`

`mon_data_avail_crit`

詳細

Ceph は、モニターのデータストアで利用可能なディスク領域がこの割合以下になると、クラスターログに **HEALTH_ERR** ステータスを発行します。

型

整数

デフォルト

`5`

`mon_warn_on_cache_pools_without_hit_sets`

詳細

キャッシュプールに **hit_set_type** パラメーターが設定されていないと、Ceph はクラスターログで **HEALTH_WARN** ステータスを発行します。

型

ブール値

デフォルト

`True`

`mon_warn_on_crush_straw_calc_version_zero`

詳細

CRUSH の **straw_calc_version** がゼロの場合、Ceph はクラスターログの **HEALTH_WARN** ステータスを発行します。詳細は、[CRUSH 設定可能なパラメーター](#) を参照してください。

型

ブール値

デフォルト

True

mon_warn_on_legacy_crush_tunables

詳細

CRUSH の調整可能なパラメーターが古くなり過ぎた場合 (**mon_min_crush_required_version** よりも古い場合)、Ceph はクラスターログで **HEALTH_WARN** ステータスを発行します。

型

ブール値

デフォルト

True

mon_crush_min_required_version

詳細

この設定では、クラスターが必要とする最小のチューナブルプロファイルバージョンを定義します。

型

文字列

デフォルト

firefly

mon_warn_on_osd_down_out_interval_zero

詳細

mon_osd_down_out_interval 設定がゼロの場合、Ceph はクラスターログで **HEALTH_WARN** ステータスを発行します。これは、**noout** フラグが設定されている場合にもリーダーと同様の動作をするためです。管理者は、**noout** フラグを設定してクラスターのトラブルシューティングが容易になります。Ceph は、管理者が設定がゼロであることを認識するために警告を発します。

型

ブール値

デフォルト

True

mon_cache_target_full_warn_ratio

詳細

cache_target_full と **target_max_object** の比率で、Ceph により警告が表示されます。

型

浮動小数点 (Float)

デフォルト

0.66

mon_health_data_update_interval

詳細

クォーラム内のモニターがピアとヘルスステータスを共有する頻度 (秒単位)。マイナスの数値を入力すると、ヘルス更新が無効になります。

型

浮動小数点 (Float)

デフォルト

60

mon_health_to_clog**詳細**

この設定により、Ceph が定期的にクラスターログにヘルスサマリーを送信することができます。

型

ブール値

デフォルト

True

mon_health_detail_to_clog**詳細**

この設定により、Ceph が定期的にクラスターログにヘルス詳細を送信することができます。

タイプ

ブール値

デフォルト

True

mon_op_complaint_time**詳細**

更新が行われなかった後、Ceph Monitor 操作がブロックされたと見なされるまでの秒数。

タイプ

整数

デフォルト

30

mon_health_to_clog_tick_interval**詳細**

モニターが正常性の要約をクラスターログに送信する頻度 (秒単位)。正数以外の数値を指定すると、この設定は無効になります。現在のヘルスサマリーが空であったり、前回と同じであったりする場合、モニターはステータスをクラスターログに送信しません。

型

整数

デフォルト

3600

mon_health_to_clog_interval**詳細**

モニターが正常性の要約をクラスターログに送信する頻度 (秒単位)。正数以外の数値を指定すると、この設定は無効になります。モニターは常にクラスターログにサマリーを送信します。

型

整数

デフォルト

60

mon_osd_full_ratio**詳細**OSD が **full** とみなされるまでのディスク領域のパーセンテージ。**型**

浮動小数点

デフォルト**.95****mon_osd_nearfull_ratio****詳細**OSD がほぼ **nearfull** とみなされるまでのディスク領域のパーセンテージ。**型**

浮動小数点 (Float)

デフォルト**.85****mon_sync_trim_timeout****詳細, 型**

double

デフォルト**30.0****mon_sync_heartbeat_timeout****詳細, 型**

double

デフォルト**30.0****mon_sync_heartbeat_interval****詳細, 型**

double

デフォルト**5.0****mon_sync_backoff_timeout****詳細, 型**

double

デフォルト

30.0

`mon_sync_timeout`

詳細

モニターが、更新メッセージをあきらめて再びブートストラップを行うまで、同期プロバイダーから次のメッセージを待つ秒数。

型

double

デフォルト

30.0

`mon_sync_max_retries`

詳細, 型

整数

デフォルト

5

`mon_sync_max_payload_size`

詳細

同期ペイロードの最大サイズ (単位: バイト) です。

型

32 ビット整数

デフォルト

1045676

`paxos_max_join_drift`

詳細

モニターデータストアを最初に同期させるまでの、Paxos 最大反復回数です。モニターは、ピアが自分よりも先に進んでいると判断すると、先に進む前にまずデータストアと同期します。

型

整数

デフォルト

10

`paxos_stash_full_interval`

詳細

PaxosService の状態のフルコピーを隠す頻度 (コミット数)。現在、この設定は **mds**、**mon**、**auth**、および **mgr** PaxosServices のみに影響します。

型

整数

デフォルト

25

`paxos_propose_interval`

詳細

この時間間隔で更新情報を集めてから、マップの更新を提案します。

型

double

デフォルト

1.0

paxos_min**詳細**

維持する paxos の状態の最小数

型

整数

デフォルト

500

paxos_min_wait**詳細**

活動していない期間の後に更新を収集するための最小時間。

型

double

デフォルト

0.05

paxos_trim_min**詳細**

トリミング前に許容される追加提案の数

型

整数

デフォルト

250

paxos_trim_max**詳細**

一度にトリミングする追加提案の最大数

型

整数

デフォルト

500

paxos_service_trim_min**詳細**

トリムのトリガーとなる最小のバージョン数 (0 であれば無効)

型

整数

デフォルト

250

paxos_service_trim_max

詳細

1 回の提案中にトリミングするバージョン数の最大値 (0 であれば無効)

型

整数

デフォルト

500

mon_max_log_epochs

詳細

1 回の提案中にトリミングするログエポック数の最大値

型

整数

デフォルト

500

mon_max_pgmap_epochs

詳細

1 回の提案中にトリミングする pgmap エポック数の最大値

型

整数

デフォルト

500

mon_mds_force_trim_to

詳細

モニターがこのポイントまで mdsmaps をトリミングするのを強制します (0 は無効、危険なので使用には注意が必要)。

型

整数

デフォルト

0

mon_osd_force_trim_to

詳細

指定したエポックでクリーンではない PG があっても、モニターがこのポイントまで osdmaps をトリミングするのを強制します (0 は無効、危険なので使用には注意が必要)。

型

整数

デフォルト

0

mon_osd_cache_size**詳細**

基礎となるストアのキャッシュに依存しない、osdmaps のキャッシュサイズ

型

整数

デフォルト

10

mon_election_timeout**詳細**

選択の提案側で、すべての ACK を待つ最長の時間 (秒単位)

型

浮動小数点 (Float)

デフォルト

5

mon_lease**詳細**

モニターのバージョンのリース期間 (秒単位)

型

浮動小数点 (Float)

デフォルト

5

mon_lease_renew_interval_factor**詳細**

mon lease * mon lease renew interval factor は、リーダーが他のモニターのリースを更新する間隔になります。係数は **1.0** 未満でなければなりません。

型

浮動小数点 (Float)

デフォルト

0.6

mon_lease_ack_timeout_factor**詳細**

リーダーは、プロバイダーがリース拡張を承認するまで **mon lease * mon lease ack timeout factor** を待機します。

型

浮動小数点 (Float)

デフォルト

2.0

mon_accept_timeout_factor

詳細

Leader は **mon lease * mon accept timeout factor** を待ち、リクエスターが Paxos の更新を受け入れるのを待機します。また、Paxos の回復期にも同様の目的で使用されます。

型

浮動小数点 (Float)

デフォルト

2.0

mon_min_osdmap_epochs

詳細

常時保持する OSD マップエポックの最小数

型

32 ビット整数

デフォルト

500

mon_max_pgmap_epochs

詳細

モニターが保持すべき PG マップエポックの最大数

型

32 ビット整数

デフォルト

500

mon_max_log_epochs

詳細

モニターが保持すべきログエポックの最大数

型

32 ビット整数

デフォルト

500

clock_offset

詳細

システムクロックをどれだけオフセットするか。詳細は、**Clock.cc** を参照してください。

型

double

デフォルト

0

mon_tick_interval

詳細

モニターの目盛りの間隔 (秒単位)

型

32 ビット整数

デフォルト

5

mon_clock_drift_allowed

詳細

モニター間で許容されるクロックドリフト (秒単位)

型

浮動小数点 (Float)

デフォルト

.050

mon_clock_drift_warn_backoff

詳細

クロックドリフト警告のための指数バックオフ

型

浮動小数点 (Float)

デフォルト

5

mon_timecheck_interval

詳細

リーダーの時刻チェック (クロックドリフトチェック) 間隔 (秒単位)

型

浮動小数点 (Float)

デフォルト

300.0

mon_timecheck_skew_interval

詳細

スキューがあった場合のリーダーの時刻チェック (クロックドリフトチェック) 間隔 (秒単位)

型

浮動小数点 (Float)

デフォルト

30.0

mon_max_osd

詳細

クラスターで許容される OSD の最大数

型

32 ビット整数

デフォルト**10000****mon_globalid_prealloc****詳細**

クラスター内のクライアントおよびデーモンに事前に割り当てるグローバル ID の数

型

32 ビット整数

デフォルト**100****mon_sync_fs_threshold****詳細**

指定された数のオブジェクトを書き込む際に、ファイルシステムと同期します。無効にするには **0** に設定します。

型

32 ビット整数

デフォルト**5****mon_subscribe_interval****詳細**

サブスクリプションの更新間隔 (秒単位)。サブスクリプションメカニズムにより、クラスターマップやログ情報を取得することができます。

型

double

デフォルト**300****mon_stat_smooth_intervals****詳細**

最後の **N** PG マップに対する統計は、Ceph によりスムーズになります。

型

整数

デフォルト**2****mon_probe_timeout****詳細**

モニターがブートストラップを行うまで、ピアを探すために待機する秒数

型

double

デフォルト**2.0**

mon_daemon_bytes**詳細**

メタデータサーバーおよび OSD メッセージのメッセージメモリー容量 (単位: バイト)

型

64 ビット整数未署名

デフォルト

400ul << 20

mon_max_log_entries_per_event**詳細**

1 イベントあたりのログエントリーの最大数

型

整数

デフォルト

4096

mon_osd_prime_pg_temp**詳細**

クラスター外の OSD がクラスターに戻ってきたときに、以前の OSD で PGMap のプライミングを行うことを有効または無効にします。**true** 設定では、クライアントは、PG のピア化として OSD で新たに実行するまで、以前の OSD を引き続き使用します。

型

ブール値

デフォルト

true

mon_osd_prime_pg_temp_max_time**詳細**

クラスター外の OSD がクラスターに戻ってきたときに、モニターが PGMAP のプライミングを試みる時間 (秒単位)

型

浮動小数点 (Float)

デフォルト

0.5

mon_osd_prime_pg_temp_max_time_estimate**詳細**

すべての PG を並行してプライミングするまでに、各 PG での時間の最大推定値

型

浮動小数点 (Float)

デフォルト

0.25

mon_osd_allow_primary_affinity

詳細

osdmap で **primary_affinity** を設定できるようにします。

型

ブール値

デフォルト

False

mon_osd_pool_ec_fast_read**詳細**

プールでの高速読み込みオンにするかどうか。作成時に **fast_read** が指定されていない場合に、新たに作成されたイレイジャープールのデフォルト設定として使用します。

型

ブール値

デフォルト

False

mon_mds_skip_sanity**詳細**

バグ発生に関わらず続行したい際に、FSMap の安全アサーションをスキップします。FSMap のサニティーチェックに失敗すると Monitor は終了しますが、このオプションを有効にすることでそれを無効にすることができます。

型

ブール値

デフォルト

False

mon_max_mdsmmap_epochs**詳細**

1 回の提案中にトリミングする mdsmmap エポック数の最大値

型

整数

デフォルト

500

mon_config_key_max_entry_size**詳細**

config-key エントリーの最大サイズ (単位: バイト)

型

整数

デフォルト

4096

mon_scrub_interval**詳細**

保存されているチェックサムと、保存されているすべての鍵の計算されたチェックサムを比較して、モニターがストアをスクラブする頻度 (秒単位)

型

整数

デフォルト

3600*24

mon_scrub_max_keys**詳細**

都度スクラブするキーの最大数

型

整数

デフォルト

100

mon_compact_on_start**詳細**

ceph-mon の起動時に Ceph Monitor ストアとして使用されるデータベースを圧縮します。手動コンパクションは、通常のコンパクションが機能しない場合に、モニターデータベースを縮小し、パフォーマンスを向上させるのに役立ちます。

型

ブール値

デフォルト

False

mon_compact_on_bootstrap**詳細**

ブートストラップ時に Ceph Monitor ストアとして使用されるデータベースを圧縮します。ブートストラップ後に、モニターはクォーラムを作るためにお互いにプロービングを開始します。クォーラムに参加する前にタイムアウトした場合は、やり直して、再びブートストラップを行います。

型

ブール値

デフォルト

False

mon_compact_on_trim**詳細**

古い状態をトリミングする際に、ある接頭辞 (paxos を含む) をコンパクト化します。

型

ブール値

デフォルト

True

mon_cpu_threads

詳細

モニター上で CPU 負荷の高い作業を行うためのスレッドの数

型

ブール値

デフォルト

True

mon_osd_mapping_pgs_per_chunk**詳細**

配置グループから OSD へのマッピングをチャンクで計算します。このオプションで、チャンクごとの配置グループ数を指定します。

型

整数

デフォルト

4096

mon_osd_max_split_count**詳細**

分割を作成させるための関係する OSD ごとの最大の PG 数。プールの **pg_num** を増やすと、配置グループは、そのプールを提供するすべての OSD で分割されます。PG を分割する際、極端な倍数は避けるべきです。

型

整数

デフォルト

300

rados_mon_op_timeout**詳細**

rados 操作からのエラーを返す前に、モニターからの応答を待つ時間 (秒数)。0 は制限、または待ち時間がないことを意味します。

型

double

デフォルト

0

関連情報

- [プール値](#)
- [CRUSH の調整可能パラメーター](#)

付録D CEPHX の設定オプション

デプロイメント時に設定可能な Cephx の設定オプションを以下に示します。

auth_cluster_required

詳細

これが有効な場合には、Red Hat Ceph Storage クラスターデーモン **ceph-mon** および **ceph-osd** は相互に認証する必要があります。有効な設定は **cephx** または **none** です。

型

文字列

必須

いいえ

デフォルト

cephx.

auth_service_required

詳細

有効にすると、Red Hat Ceph Storage クラスターデーモンは、Ceph サービスにアクセスするために、Ceph クライアントが Red Hat Ceph Storage クラスターと認証することを要求します。有効な設定は **cephx** または **none** です。

型

文字列

必須

いいえ

デフォルト

cephx.

auth_client_required

詳細

有効にすると、Ceph クライアントは、Red Hat Ceph Storage クラスターが Ceph クライアントと認証することを要求します。有効な設定は **cephx** または **none** です。

型

文字列

必須

いいえ

デフォルト

cephx.

keyring

詳細

キーリングファイルのパス

型

文字列

必須

いいえ

デフォルト

/etc/ceph/\$cluster.\$name.keyring,/etc/ceph/\$cluster.keyring,/etc/ceph/keyring,/etc/ceph/keyring.bin

keyfile

詳細

キーファイル (つまり、キーのみを含むファイル) へのパス

型

文字列

必須

いいえ

デフォルト

なし

key

詳細

キー (つまり、キーそのもののテキスト文字列)。推奨されません。

型

文字列

必須

いいえ

デフォルト

なし

ceph-mon

場所

\$mon_data/keyring

ケイパビリティー

mon 'allow *'

ceph-osd

場所

\$osd_data/keyring

ケイパビリティー

mon 'allow profile osd' osd 'allow *'

radosgw

場所

\$rgw_data/keyring

ケイパビリティー

mon 'allow rwx' osd 'allow rwx'

cephx_require_signatures

詳細

true に設定した場合には、Ceph クライアントと Red Hat Ceph Storage クラスター間の全メッセージトラフィック、および Red Hat Ceph Storage クラスターを設定するデーモン間での署名が必要です。

型

ブール値

必須

いいえ

デフォルト

false

cephx_cluster_require_signatures**詳細**

true に設定した場合には、Ceph では、Red Hat Ceph Storage クラスターを設定する Ceph デーモン間のすべてのメッセージトラフィックに対する署名が必要です。

型

ブール値

必須

いいえ

デフォルト

false

cephx_service_require_signatures**詳細**

true に設定した場合には、Ceph クライアントと Red Hat Ceph Storage クラスター間のすべてのメッセージトラフィックに対する署名が必要です。

型

ブール値

必須

いいえ

デフォルト

false

cephx_sign_messages**詳細**

Ceph のバージョンがメッセージ署名をサポートしている場合、Ceph はすべてのメッセージに署名し、メッセージが偽装されないようにします。

型

ブール値

デフォルト

true

auth_service_ticket_ttl**詳細**

Red Hat Ceph Storage クラスターが Ceph クライアントに認証用のチケットを送信すると、クラスターはそのチケットに生存時間を割り当てます。

型

double

デフォルト

60*60

関連情報

- <additional resource 1>
- <additional resource 2>

付録E プール、配置グループ、および CRUSH の設定オプション

プール、配置グループ、および CRUSH アルゴリズムを管理する Ceph のオプションです。

mon_allow_pool_delete

詳細

モニターがプールを削除することができます。RHCS 3 以降のリリースでは、データ保護のための追加措置として、モニターはデフォルトでプールを削除できません。

型

ブール値

デフォルト

false

mon_max_pool_pg_num

詳細

プールあたりの配置グループの最大数

型

整数

デフォルト

65536

mon_pg_create_interval

詳細

同じ Ceph OSD デーモンでの PG 作成の間の秒数

型

浮動小数点 (Float)

デフォルト

30.0

mon_pg_stuck_threshold

詳細

PG がスタックしていると判断できるまでの秒数

型

32 ビット整数

デフォルト

300

mon_pg_min_inactive

詳細

Ceph は、**mon_pg_stuck_threshold** より長く非アクティブのままの PG の数がこの設定を超える場合に、クラスターログに **HEALTH_ERR** ステータスを発行します。デフォルト設定は1つの PG です。正数以外の数値を指定すると、この設定は無効になります。

型

整数

デフォルト

1

mon_pg_warn_min_per_osd

詳細

Ceph は、クラスター内の OSD ごとの PG の平均数がこの設定よりも小さい場合に、クラスターログで **HEALTH_WARN** ステータスを発行します。正数以外の数値を指定すると、この設定は無効になります。

型

整数

デフォルト

30

mon_pg_warn_max_per_osd

詳細

Ceph は、クラスター内の OSD ごとの PG の平均数がこの設定よりも大きい場合に、クラスターログの **HEALTH_WARN** ステータスを発行します。正数以外の数値を指定すると、この設定は無効になります。

型

整数

デフォルト

300

mon_pg_warn_min_objects

詳細

クラスター内のオブジェクトの総数がこの数以下の場合は警告を発生しません。

型

整数

デフォルト

1000

mon_pg_warn_min_pool_objects

詳細

オブジェクト数がこの数以下のプールには警告を発生しません。

型

整数

デフォルト

1000

mon_pg_check_down_all_threshold

詳細

down OSD のしきい値 (パーセント) で、Ceph はすべての PG をチェックして、それらがスタックまたは古くなっていることを確認します。

型

浮動小数点 (Float)

デフォルト

0.5

mon_pg_warn_max_object_skew

詳細

プール内のオブジェクトの平均数 **mon pg warn max object skew** を超える場合、Ceph はクラスターログで **HEALTH_WARN** ステータスを発行します。正数以外の数値を指定すると、この設定は無効になります。

型

浮動小数点 (Float)

デフォルト

10

mon_delta_reset_interval

詳細

Ceph が PG デルタをゼロにリセットするまでの非アクティブ時の秒数。Ceph は、各プールの使用済み容量のデルタを追跡し、管理者がリカバリーの進捗状況やパフォーマンスを評価するのに役立てます。

型

整数

デフォルト

10

mon_osd_max_op_age

詳細

HEALTH_WARN ステータスを発行する前に操作が完了するまでの最大期間 (秒単位)。

型

浮動小数点 (Float)

デフォルト

32.0

osd_pg_bits

詳細

Ceph OSD デーモンごとの配置グループのビット数

型

32 ビット整数

デフォルト

6

osd_pgp_bits

詳細

配置目的の配置グループ (PGP) の Ceph OSD デーモンあたりのビット数

型

32 ビット整数

デフォルト

6

`osd_crush_chooseleaf_type`

詳細

CRUSH ルールで **chooseleaf** に使用するバケットタイプ。名前ではなく従来のランクを使用します。

型

32 ビット整数

デフォルト

1.通常は、1つまたは複数の Ceph OSD デーモンを含むホストです。

`osd_pool_default_crush_replicated_ruleset`

詳細

レプリケートされたプールを作成する際に使用するデフォルトの CRUSH ルールセット

型

8 ビット整数

デフォルト

0

`osd_pool_erasure_code_stripe_unit`

詳細

イレイジャーコード化されたプールのオブジェクトストライプのチャンクのデフォルトサイズをバイト単位で設定します。サイズ *S* のすべてのオブジェクトは *N* ストライプとして格納され、各データチャンクは **stripe unit** バイトを受け取ります。***N* * stripe unit** バイトの各ストライプは、個別にエンコード/エンコードされます。このオプションは、イレイジャーコードプロファイルの **stripe_unit** 設定で上書きできます。

型

32 ビット符号なし整数

デフォルト

4096

`osd_pool_default_size`

詳細

プール内のオブジェクトのレプリカ数を設定します。デフォルト値は、**ceph osd pool set {pool-name} size {size}** と同じです。

型

32 ビット整数

デフォルト

3

`osd_pool_default_min_size`

詳細

プール内のオブジェクトに対して、クライアントへの書き込み操作を確認するための、書き込み

済みレプリカの最小数を設定します。最小値が満たされていない場合、Ceph はクライアントへの書き込みを確認しません。この設定により、**degraded** モードで動作している場合にレプリカの最小数を確保できます。

型

32 ビット整数

デフォルト

0 (これは、特定の最小値がないことを意味します)**0** の場合、最小は **size - (size / 2)** になります。

osd_pool_default_pg_num**詳細**

プールの配置グループのデフォルト数。デフォルト値は、**mkpool** で **pg_num** と同じです。

型

32 ビット整数

デフォルト

8

osd_pool_default_pgp_num**詳細**

プールに対する配置の配置グループのデフォルト数です。デフォルト値は、**mkpool** で **pgp_num** と同じです。PG と PGP は同じであるべきです。

型

32 ビット整数

デフォルト

8

osd_pool_default_flags**詳細**

新しいプールのデフォルトフラグ

型

32 ビット整数

デフォルト

0

osd_max_pgls**詳細**

リストアップする配置グループの最大数。大きな数を要求するクライアントは、Ceph OSD デーモンを拘束できます。

型

未署名の 64 ビット整数

デフォルト

1024

備考

デフォルトで問題ありません。

osd_min_pg_log_entries

詳細

ログファイルをトリミングする際に維持する配置グループログの最小数

型

32 ビット符号なし整数

デフォルト

1000

osd_deep_scrub_large_omap_object_value_sum_threshold

説明

RADOS オブジェクトが使用できる omap キーの数のしきい値を設定します。omap キーの数がしきい値を超えると、RADOS オブジェクトを含む配置グループは、グループがディープスクラブされたときにメッセージをログに記録します。クラスターログ **ceph.log** に1つのメッセージが記録されます。警告 **cluster [WRN] Large omap object found**、キーの数、およびオブジェクトのサイズ (バイト単位) が含まれます。2 番目のメッセージは、クラスターステータスメッセージ **Large OMAP count** を HEALTH_WARN に追加します。詳細は、<https://access.redhat.com/solutions/3660171> を参照してください。

タイプ

64 ビット未署名の整数

デフォルト

20000

付録F OBJECT STORAGE DAEMON (OSD) の設定オプション

デプロイメント時に設定可能な Ceph Object Storage Daemon (OSD) の設定オプションを以下に示します。

osd_uuid

詳細

Ceph OSD の Universally Unique Identifier (UUID)

型

UUID

デフォルト

UUID

備考

osd uuid は単一の Ceph OSD に適用されます。**fsid** はクラスター全体に適用されます。

osd_data

詳細

OSD のデータへのパスCeph のデプロイ時にディレクトリーを作成する必要があります。OSD データ用のドライブをこのマウントポイントにマウントします。

IMPORTANT: Red Hat does not recommend changing the default.

型

文字列

デフォルト

/var/lib/ceph/osd/\$cluster-\$id

osd_max_write_size

詳細

書き込みの最大サイズ (メガバイト)

型

32 ビット整数

デフォルト

90

osd_client_message_size_cap

詳細

メモリー上で許可される最大のクライアントデータメッセージ

型

64 ビット整数未署名

デフォルト

500 MB のデフォルト**500*1024L*1024L**

osd_class_dir

詳細

RADOS クラスのプラグインのクラスパス

型

文字列

デフォルト

\$libdir/rados-classes

osd_max_scrubs

詳細

Ceph OSD ごとの同時スクラブ操作の最大数

型

32 ビット整数

デフォルト

1

osd_scrub_thread_timeout

詳細

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)

型

32 ビット整数

デフォルト

60

osd_scrub_finalize_thread_timeout

詳細

スクラブ最終スレッドがタイムアウトするまでの最大時間 (秒単位)

型

32 ビット整数

デフォルト

60*10

osd_scrub_begin_hour

詳細

軽いスクラブや深いスクラブを始めることができる最も早い時間。これは、スクラビングの時間枠を定義するために **osd scrub end hour** パラメーターと共に使用し、スクラビングをオフピーク時間に制限できるようにします。設定は、24 時間サイクルの時間を指定するために整数を取ります。たとえば、**0** は午前 12:01 から午前 1:00 までを表し、**13** は午後 1:01 から午後 2:00 までを表します。

型

32 ビット整数

デフォルト

0 (午前 12:01 から 1:00)

osd_scrub_end_hour

詳細

軽いスクラブや深いスクラブを始めることができる最も遅い時間。これは、**osd scrub begin hour** パラメーターとともに使用してスクラブタイムウィンドウを定義し、スクラブをオフピーク時間に制限します。設定は、24 時間サイクルの時間を指定するために整数を取ります。たとえば、**0** は午前 12:01 から午前 1:00 までを表し、**13** は午後 1:01 から午後 2:00 までを表します。**end** 時間は、**begin** 時間よりも大きくなければなりません。

型

32 ビット整数

デフォルト

24 (午後 11:01 から午前 12:00)

osd_scrub_load_threshold

詳細

最大の負荷。(getloadavg() 関数で定義された) システムの負荷がこの数値よりも大きい場合、Ceph はスクラブを実行しません。デフォルトは **0.5** です。

型

浮動小数点 (Float)

デフォルト

0.5

osd_scrub_min_interval

詳細

Red Hat Ceph Storage クラスターの負荷が低いときに、Ceph OSD をスクラブする最小の間隔 (秒単位)

型

浮動小数点 (Float)

デフォルト

1 日 1 回。 **60*60*24**

osd_scrub_max_interval

詳細

クラスター負荷に関わらず Ceph OSD をスクラビングする最大の間隔 (秒単位)。

型

浮動小数点 (Float)

デフォルト

1 週間に 1 回になります。 **7*60*60*24**

osd_scrub_interval_randomize_ratio

詳細

比率を取り、**osd scrub min interval** および **osd scrub max interval** の間隔の間でスケジュールされたスクラブをランダム化します。

型

浮動小数点 (Float)

デフォルト

0.5。

mon_warn_not_scrubbed

詳細

スクラブされていない PG について警告する **osd_scrub_interval** からの秒数。

型

整数

デフォルト

0 (警告なし)。

osd_scrub_chunk_min

詳細

オブジェクトストアは、ハッシュの境界で終わるチャンクに分割されています。チャンキースクラブの場合、Ceph はオブジェクトを1チャンクずつスクラブし、そのチャンクへの書き込みをブロックします。**osd scrub chunk min** 設定は、スクラビングするチャンクの最小数を表します。

型

32 ビット整数

デフォルト

5

osd_scrub_chunk_max

詳細

スクラブするチャンクの最大数

型

32 ビット整数

デフォルト

25

osd_scrub_sleep

詳細

ディープスクラブ操作の間のスリープ時間

型

浮動小数点 (Float)

デフォルト

0 (またはオフ)

osd_scrub_during_recovery

詳細

リカバリー時のスクラブを可能にします。

型

ブール (Bool)

デフォルト

false

osd_scrub_invalid_stats

詳細

無効と判定された統計情報を修正するために、強制的に追加のスクラブを実行します。

型

ブール (Bool)

デフォルト

true

osd_scrub_priority**詳細**

クライアント I/O に対するスクラブ操作のキューの優先順位を制御します。

型

32 ビット符号なし整数

デフォルト

5

osd_scrub_cost**詳細**

キューのスケジューリングのために、スクラブ操作のコストをメガバイト単位で表したもの。

型

32 ビット符号なし整数

デフォルト

50 << 20

osd_deep_scrub_interval**詳細**

すべてのデータを完全に読み込むディープスクラビングのための間隔。**osd scrub load threshold** パラメーターは、この設定には影響を与えません。

型

浮動小数点 (Float)

デフォルト

1 週間に 1 回になります。**60*60*24*7**

osd_deep_scrub_stride**詳細**

ディープスクラブを実施する際の読み取りサイズ

型

32 ビット整数

デフォルト

512 KB。**524288**

mon_warn_not_deep_scrubbed**詳細**

スクラビングされていない PG について警告する **osd_deep_scrub_interval** からの秒数。

型

整数

デフォルト**0** (警告なし)。**osd_deep_scrub_randomize_ratio****詳細**

スクラブが無作為にディープスクラビングになる変化 (**osd_deep_scrub_interval** が経過する可能性も)

型

浮動小数点 (Float)

デフォルト**0.15** または 15%。**osd_deep_scrub_update_digest_min_age****詳細**

スクラブがオブジェクト全体のダイジェストを更新するまでに、オブジェクトが何秒経過していなければならないか。

型

整数

デフォルト**120** (2 時間)。**osd_op_num_shards****詳細**

クライアント操作のためのシャード数

型

32 ビット整数

デフォルト**0****osd_op_num_threads_per_shard****詳細**

クライアント操作のためのシャードあたりのスレッド数

型

32 ビット整数

デフォルト**0****osd_op_num_shards_hdd****詳細**

HDD 操作のためのシャード数

型

32 ビット整数

デフォルト**5****osd_op_num_threads_per_shard_hdd****詳細**

HDD 操作のためのシャードあたりのスレッド数

型

32 ビット整数

デフォルト**1****osd_op_num_shards_ssd****詳細**

SSD 操作のためのシャード数

型

32 ビット整数

デフォルト**8****osd_op_num_threads_per_shard_ssd****詳細**

SSD 操作のためのシャードあたりのスレッド数

型

32 ビット整数

デフォルト**2****osd_client_op_priority****詳細**

クライアントの操作に設定されている優先順位。これは、**osd recovery op priority** と相対的になります。

型

32 ビット整数

デフォルト**63****有効な範囲**

1-63

osd_recovery_op_priority**詳細**

復元の操作に設定されている優先順位。これは、**osd client op priority** と相対的になります。

型

32 ビット整数

デフォルト**3****有効な範囲**

1-63

osd_op_thread_timeout**詳細**

Ceph OSD 操作スレッドのタイムアウト (秒単位)

型

32 ビット整数

デフォルト**30****osd_op_complaint_time****詳細**

指定された秒数が経過すると、クレームに値する操作になります。

型

浮動小数点 (Float)

デフォルト**30****osd_disk_threads****詳細**

スクラビングやスナップトリミングなど、バックグラウンドでのディスクを多用する OSD 操作に使用されるディスクスレッドの数

型

32 ビット整数

デフォルト**1****osd_op_history_size****詳細**

追跡する完了した操作の最大数

型

32 ビット未署名の整数

デフォルト**20****osd_op_history_duration****詳細**

追跡する最も古い完了した操作

型

32 ビット未署名の整数

デフォルト**600****osd_op_log_threshold****詳細**

一度に表示する操作ログの数

型

32 ビット整数

デフォルト**5****osd_op_timeout****詳細**

実行中の OSD 操作がタイムアウトするまでの時間 (秒)

型

整数

デフォルト**0****重要**

クライアントが結果に対応できない限り、**osd op timeout** オプションを設定しないでください。例えば、仮想マシン上で動作するクライアントにこのパラメーターを設定すると、仮想マシンがこのタイムアウトをハードウェアの故障と解釈するため、データの破損につながる可能性があります。

osd_max_backfills**詳細**

1つの OSD に対して、または1つの OSD から許容されるバックフィル操作の最大数

型

64 ビット未署名の整数

デフォルト**1****osd_backfill_scan_min****詳細**

バックフィルスキャン1回あたりのオブジェクトの最小数

型

32 ビット整数

デフォルト**64****osd_backfill_scan_max****詳細**

バックフィルスキャン1回あたりのオブジェクトの最大数

型

32 ビット整数

デフォルト

512

osd_backfillfull_ratio

説明

Ceph OSD のフル比率がこの値以上の場合、バックフィル要求の受け入れを拒否します。

型

浮動小数点 (Float)

デフォルト

0.85

osd_backfill_retry_interval

詳細

バックフィル要求を再試行するまでの待ち時間 (秒数)

型

double

デフォルト

10.0

osd_map_dedup

詳細

OSD マップの重複の削除を有効にします。

型

ブール値

デフォルト

true

osd_map_cache_size

詳細

OSD マップキャッシュのサイズ (メガバイト)

型

32 ビット整数

デフォルト

50

osd_map_cache_bl_size

詳細

OSD デーモンのメモリー内 OSD マップキャッシュのサイズ

型

32 ビット整数

デフォルト**50****osd_map_cache_bl_inc_size****詳細**

OSD デーモンのメモリー内 OSD マップキャッシュの増分サイズ

型

32 ビット整数

デフォルト**100****osd_map_message_max****詳細**

MOSDMap メッセージごとに許容される最大のマップエントリー数

型

32 ビット整数

デフォルト**40****osd_snap_trim_thread_timeout****詳細**

スナップトリムスレッドがタイムアウトするまでの最大時間 (秒単位)

型

32 ビット整数

デフォルト**60*60*1****osd_pg_max_concurrent_snap_trims****詳細**

PG ごとの並列スナップトリムの最大数。PG ごとに何個のオブジェクトを一度にトリミングするかを制御します。

型

32 ビット整数

デフォルト**2****osd_snap_trim_sleep****詳細**

PG が発行する各トリム操作の間にスリープを挿入します。

型

32 ビット整数

デフォルト**0**

osd_max_trimming_pgs**詳細**

トリミング PG の最大数

型

32 ビット整数

デフォルト

2

osd_backlog_thread_timeout**詳細**

バックログスレッドがタイムアウトするまでの最大時間 (秒単位)

型

32 ビット整数

デフォルト

60*60*1

osd_default_notify_timeout**詳細**

OSD デフォルト通知のタイムアウト (単位: 秒)

型

32 ビット符号なし整数

デフォルト

30

osd_check_for_log_corruption**詳細**

ログファイルが破損していないか確認します。計算量が多くなる可能性があります。

型

ブール値

デフォルト

false

osd_remove_thread_timeout**詳細**

OSD 削除スレッドがタイムアウトするまでの最大時間 (秒単位)

型

32 ビット整数

デフォルト

60*60

osd_command_thread_timeout**詳細**

コマンドスレッドがタイムアウトするまでの最大時間 (秒単位)

型

32 ビット整数

デフォルト**10*60****osd_command_max_records****詳細**

失ったオブジェクトを返す際の数制限します。

型

32 ビット整数

デフォルト**256****osd_auto_upgrade_tmap****詳細**古いオブジェクトの **omap** に **tmap** を使用します。**型**

ブール値

デフォルト**true****osd_tmapput_sets_users_tmap****詳細**デバッグにだけ **tmap** を使用します。**型**

ブール値

デフォルト**false****osd_preserve_trimmed_log****詳細**

トリミングされたログファイルは保持されますが、より多くのディスク容量を使用します。

型

ブール値

デフォルト**false****osd_recovery_delay_start****詳細**

ピアリングが完了すると、Ceph は指定された秒数だけ遅延してからオブジェクトの回復を開始します。

型

浮動小数点 (Float)

デフォルト**0****osd_recovery_max_active****詳細**

OSD ごとに一度のアクティブな復旧要求の数。リクエストが増えれば復旧も早くなりますが、その分クラスターへの負荷も大きくなります。

型

32 ビット整数

デフォルト**3****osd_recovery_max_chunk****詳細**

復元したデータチャンクをプッシュする際の最大サイズ

型

64 ビット整数未署名

デフォルト**8 << 20****osd_recovery_threads****詳細**

データを復元するためのスレッド数

型

32 ビット整数

デフォルト**1****osd_recovery_thread_timeout****詳細**

復元スレッドがタイムアウトするまでの最大時間 (秒単位)

型

32 ビット整数

デフォルト**30****osd_recover_clone_overlap****詳細**

復元時のクローンのオーバーラップを保持します。常に **true** に設定する必要があります。

型

ブール値

デフォルト**true**

`rados_osd_op_timeout`

詳細

RADOS 操作からのエラーを返す前に、RADOS が OSD からの応答を待つ時間 (秒数)。値が 0 の場合は制限がないことを意味します。

型

double

デフォルト

0

付録G CEPH MONITOR と OSD の設定オプション

ハートビート設定を変更する際には、Ceph 設定ファイルの **[global]** セクションにその設定を含めます。

mon_osd_min_up_ratio

詳細

Ceph が Ceph OSD デーモンを **down** とマークする前に **up** となる Ceph OSD デーモンの最小比率。

型

double

デフォルト

.3

mon_osd_min_in_ratio

詳細

Ceph が Ceph OSD デーモンを **out** とマークを付ける前に **in** となる Ceph OSD デーモンの最小比率。

型

double

デフォルト

0.75

mon_osd_laggy_halflife

詳細

laggy 予測の秒数が減ります。

型

整数

デフォルト

60*60

mon_osd_laggy_weight

詳細

laggy 予測の減少時の新しいサンプルの重み。

型

double

デフォルト

0.3

mon_osd_laggy_max_interval

詳細

ラグ推定値の **laggy_interval** の最大値 (秒単位)。モニターは適応アプローチを使用して特定の OSD の **laggy_interval** を評価します。この値は、その OSD の猶予時間を算出するために使用されます。

型

整数

デフォルト

300

mon_osd_adjust_heartbeat_grace

詳細

true に設定すると、Ceph は **laggy** 推定値に基づいてスケーリングします。

型

ブール値

デフォルト

true

mon_osd_adjust_down_out_interval

詳細

true に設定すると、Ceph は **laggy** 推定値に基づいてスケーリングされます。

型

ブール値

デフォルト

true

mon_osd_auto_mark_in

詳細

Ceph は、Ceph OSD デーモンのブートを、Ceph Storage Cluster の **in** とマークします。

型

ブール値

デフォルト

false

mon_osd_auto_mark_auto_out_in

詳細

Ceph は、Ceph Storage クラスターから自動的に **out** とマーク付けされた Ceph OSD デーモンの起動が、クラスター内 **in** にあるとマークされます。

型

ブール値

デフォルト

true

mon_osd_auto_mark_new_in

詳細

Ceph は、新しい Ceph OSD デーモンのブートを Ceph Storage Cluster の **in** とマークします。

型

ブール値

デフォルト

true

mon_osd_down_out_interval

詳細

Ceph が Ceph OSD デーモンを **down** および **out** マークした後に応答しない場合には、Ceph が待機する秒数。

型

32 ビット整数

デフォルト

600

mon_osd_downout_subtree_limit

詳細

Ceph が自動的に **out** とマークアウトする最大の CRUSH ユニットタイプ。

型

文字列

デフォルト

rack

mon_osd_reporter_subtree_level

詳細

この設定は、報告する OSD の親 CRUSH ユニットタイプを定義します。OSD は、応答しないピアを見つけた場合、モニターに障害レポートを送信します。モニターは報告された OSD の数を **down** とマークし、猶予期間後に **out** になる可能性があります。

型

文字列

デフォルト

host

mon_osd_report_timeout

詳細

応答しない Ceph OSD デーモンが **down** するまでの猶予期間 (秒単位)。

型

32 ビット整数

デフォルト

900

mon_osd_min_down_reporters

詳細

down な Ceph OSD デーモンの報告に必要な Ceph OSD デーモンの最小数。

型

32 ビット整数

デフォルト

2

osd_heartbeat_address

詳細

ハートビート用の Ceph OSD デーモンのネットワークアドレス

型

アドレス

デフォルト

ホストアドレス

osd_heartbeat_interval

詳細

Ceph OSD デーモンがピアに ping を実行する頻度 (秒単位)

型

32 ビット整数

デフォルト

6

osd_heartbeat_grace

詳細

Ceph OSD デーモンに Ceph Storage Cluster が **down** とみなすハートビートが表示されなかった場合の経過時間。

型

32 ビット整数

デフォルト

20

osd_mon_heartbeat_interval

詳細

Ceph OSD デーモンピアがない場合に、Ceph OSD デーモンが Ceph Monitor に ping を実行する頻度

型

32 ビット整数

デフォルト

30

osd_mon_report_interval_max

詳細

Ceph OSD デーモンが Ceph Monitor に報告しなければならなくなるまでに待機できる最大時間 (秒)

型

32 ビット整数

デフォルト

120

osd_mon_report_interval_min

詳細

Ceph OSD デーモンが起動またはその他の報告可能なイベントから Ceph Monitor に報告するまでに待機する最小秒数

型

32 ビット整数

デフォルト

5

有効な範囲

osd_mon_report_interval_max 未満である必要があります。

osd_mon_ack_timeout**詳細**

Ceph Monitor が統計情報の要求を確認するまでの待ち時間 (秒数)

型

32 ビット整数

デフォルト

30

付録H CEPH のデバッグとロギングの設定オプション

Ceph 設定ファイルでロギングおよびデバッグの設定は必要ありませんが、必要に応じてデフォルト設定を上書きできます。

このオプションは、チャンネルに関係なく、すべてのデーモンのデフォルトであると仮定される単一の項目を取ります。たとえば、info の指定は default=info と解釈されます。ただし、オプションはキーと値のペアを取ることもできます。たとえば、default=daemon audit=local0 はすべてのデーモンのデフォルトで audit を local0 で上書きすると解釈されます。

log_file

詳細

クラスターのロギングファイルの場所

型

文字列

必須

いいえ

デフォルト

`/var/log/ceph/$cluster-$name.log`

mon_cluster_log_file

詳細

モニタークラスターのログファイルの場所

型

文字列

必須

いいえ

デフォルト

`/var/log/ceph/$cluster.log`

log_max_new

詳細

新規ログファイルの最大数

型

整数

必須

いいえ

デフォルト

1000

log_max_recent

詳細

ログファイルに追加する最近のイベントの最大数

型

整数

必須

いいえ

デフォルト

10000

log_flush_on_exit

詳細

終了後に Ceph がログファイルをフラッシュするかどうかを決定します。

型

ブール値

必須

いいえ

デフォルト

true

mon_cluster_log_file_level

詳細

モニタークラスターのファイルロギングのレベル。有効な設定には、"debug"、"info"、"sec"、"warn"、および "error" が含まれます。

型

文字列

デフォルト

"info"

log_to_stderr

詳細

ロギングメッセージが標準エラー (**stderr**) で表示されるかどうかを確認します。

型

ブール値

必須

いいえ

デフォルト

true

err_to_stderr

詳細

エラーメッセージが標準エラー (**stderr**) で表示されるかどうかを確認します。

型

ブール値

必須

いいえ

デフォルト

true

log_to_syslog**詳細**

ロギングメッセージが **syslog** に表示されるかどうかを決定します。

型

ブール値

必須

いいえ

デフォルト

false

err_to_syslog**詳細**

エラーメッセージが **syslog** に表示されるかどうかを確認します。

型

ブール値

必須

いいえ

デフォルト

false

clog_to_syslog**詳細**

clog メッセージが **syslog** に送信されるかどうかを決定します。

型

ブール値

必須

いいえ

デフォルト

false

mon_cluster_log_to_syslog**詳細**

クラスターログが **syslog** に出力されるかどうかを確認します。

型

ブール値

必須

いいえ

デフォルト

false

mon_cluster_log_to_syslog_level**詳細**

モニタークラスターの syslog ロギングのレベル。有効な設定には、"debug"、"info"、"sec"、"warn"、および "error" が含まれます。

型

文字列

デフォルト

"info"

mon_cluster_log_to_syslog_facility**詳細**

syslog 出力を生成するファシリティ。通常、これは Ceph デーモンの daemon に設定されます。

型

文字列

デフォルト

"daemon"

clog_to_monitors**詳細**

clog メッセージをモニターに送信するかどうかを決定します。

型

ブール値

必須

いいえ

デフォルト

true

mon_cluster_log_to_graylog**詳細**

クラスターがログメッセージを graylog に出力するかどうかを決定します。

型

文字列

デフォルト

"false"

mon_cluster_log_to_graylog_host**詳細**

graylog ホストの IP アドレス。graylog ホストがモニターホストと異なる場合は、適切な IP アドレスでこの設定を上書きします。

型

文字列

デフォルト

"127.0.0.1"

mon_cluster_log_to_graylog_port

詳細

graylog ログは、このポートに送信されます。データの受信用にポートが開いていることを確認します。

型

文字列

デフォルト

"12201"

osd_preserve_trimmed_log**詳細**

トリミング後にトリミングされたログを保持します。

型

ブール値

必須

いいえ

デフォルト

false

osd_tmapput_sets_uses_tmap**詳細**

tmap を使用します。デバッグ用途のみ。

型

ブール値

必須

いいえ

デフォルト

false

osd_min_pg_log_entries**詳細**

配置グループのログエントリーの最小数

型

32 ビット未署名の整数

必須

いいえ

デフォルト

1000

osd_op_log_threshold**詳細**

1つのパスで表示する op ログメッセージの数

型

整数

必須

いいえ

デフォルト

5

付録I CEPH のスクラブオプション

Ceph は配置グループをスクラブすることでデータの整合性を確保します。以下は、スクラブ操作の増減用に調整できる Ceph スクラブオプションです。

osd_max_scrubs

説明

Ceph OSD Deamon ごとの同時スクラブ操作の最大数

タイプ

integer

デフォルト

1

osd_scrub_begin_hour

説明

スクラブが開始される特定の時間。**osd_scrub_end_hour** とともに、スクラブが発生する時間枠を定義できます。**osd_scrub_begin_hour = 0** および **osd_scrub_end_hour = 0** を使用して、1 日中スクラブできるようにします。

タイプ

integer

デフォルト

0

許容範囲

[0, 23]

osd_scrub_end_hour

説明

スクラブが終了する特定の時間。**osd_scrub_begin_hour** とともに、スクラブが発生する時間枠を定義できます。**osd_scrub_begin_hour = 0** および **osd_scrub_end_hour = 0** を使用して、1 日住スクラブできるようにします。

タイプ

integer

デフォルト

0

許容範囲

[0, 23]

osd_scrub_begin_week_day

説明

スクラブが開始される特定の日。0 = 日曜日、1 は月曜日など「osd_scrub_end_week_day」とともに、スクラブが発生する時間枠を定義できます。**osd_scrub_begin_week_day = 0** および **osd_scrub_end_week_day = 0** を使用して、週全体のスクラブを許可します。

タイプ

integer

デフォルト

0

許容範囲**[0, 6]****osd_scrub_end_week_day****説明**

これは、スクラビングが終了する日を定義します。0 = 日曜日、1 は月曜日など **osd_scrub_begin_hour** とともに、スクラブが発生する時間枠を定義できます。 **osd_scrub_begin_week_day = 0** および **osd_scrub_end_week_day = 0** を使用して、週全体のスクラブを許可します。

タイプ

integer

デフォルト

0

許容範囲**[0, 6]****osd_scrub_during_recovery****説明**

復元中のスクラブを許可します。これを **false** に設定すると、復元中のものがあれば、新しいスクラブとディープ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継続されます。これは、ビジー状態のストレージクラスターの負荷を減らすのに役立ちます。

タイプ

boolean

デフォルト

false

osd_scrub_load_threshold**説明**

正規化された最大負荷。getloadavg ()/オンライン CPU の数で定義されるシステム負荷が、この定義された数よりも高い場合、スクラブは行われません。

タイプ

float

デフォルト

0.5

osd_scrub_min_interval**説明**

Ceph Storage クラスターの負荷が低い場合に Ceph OSD デーモンをスクラブする最小間隔 (秒)。

タイプ

float

デフォルト

1 day

osd_scrub_max_interval

説明

クラスターの負荷に関係なく、Ceph OSD デーモンをスクラブする最大間隔 (秒単位)。

タイプ

float

デフォルト

7 日

osd_scrub_chunk_min

説明

1 回の操作でスクラブするオブジェクトストアチャンクの最小数。Ceph は、スクラブ中に単一のチャンクへの書き込みをブロックします。

type

integer

デフォルト

5

osd_scrub_chunk_max

説明

1 回の操作でスクラブするオブジェクトストアチャンクの最大数。

type

integer

デフォルト

25

osd_scrub_sleep

説明

次のチャンクをスクラブする前にスリープ状態になる時間。この値を増やすと、スクラブの全体的な速度が遅くなるため、クライアント操作の影響は低くなります。

type

float

デフォルト

0.0

osd_deep_scrub_interval

説明

すべてのデータを完全に読み取る **deep** スクラビングの間隔。**osd_scrub_load_threshold** はこの設定には影響しません。

type

float

デフォルト

7 日

osd_scrub_interval_randomize_ratio

説明

配置グループの次のスクラブジョブをスケジュールする際に、**osd_scrub_min_interval** に無作為に遅延を追加します。遅延は **osd_scrub_min_interval** * **osd_scrub_interval_randomized_ratio** 未満のランダムな値です。デフォルト設定では、1、1.5 * **osd_scrub_min_interval** の許容時間枠でスクラブが分散されます。

type

float

デフォルト

0.5

osd_deep_scrub_stride**詳細**

ディープスクラブを実施する際の読み取りサイズ

type

size

デフォルト

512 KB

osd_scrub_auto_repair_num_errors**説明**

これ以上多くエラーが見つかると、自動修復は発生しません。

type

integer

デフォルト

5

osd_scrub_auto_repair**説明**

これを **true** に設定すると、スクラブまたはディープスクラブによってエラーが見つかった場合に配置グループ (PG) の自動修復が有効になります。ただし、**osd_scrub_auto_repair_num_errors** を超えるエラーが見つかった場合、修復は実行されません。このオプションは定期的なスクラブ用で、Operator が開始するスクラブではありません。

type

boolean

デフォルト

false

付録J BLUESTORE の設定オプション

デプロイメント時に設定可能な Ceph BlueStore の設定オプションを以下に示します。



注記

このリストは完全ではありません。

rocksdb_cache_size

詳細

RocksDB キャッシュのサイズ (単位: MB)

型

32 ビット整数

デフォルト

512