Red Hat Ceph Storage 3

Red Hat Enterprise Linux のインストールガイド

Red Hat Enterprise Linux への Red Hat Ceph Storage のインストール

Last Updated: 2023-02-22

Red Hat Ceph Storage 3 Red Hat Enterprise Linux のインストールガイド

Red Hat Enterprise Linux への Red Hat Ceph Storage のインストール

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律上の通知

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Installation_Guide_for_Red_Hat_Enterprise_Linux.ent file |.

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

本書では、AMD64 および Intel 64 のアーキテクチャーで実行している Red Hat Enterprise Linux 7 に Red Hat Ceph Storage をインストールする方法を説明します。

目次

第1章 RED HAT CEPH STORAGE とは 4

第2章 RED HAT CEPH STORAGE のインストール要件 6
2.1. 前提条件 6
2.2. RED HAT CEPH STORAGE をインストールするための要件チェックリスト 6
2.3. RED HAT CEPH STORAGE のオペレーティングシステム要件 7
2.4. RED HAT CEPH STORAGE ノードの CDN への登録とサブスクリプションのアタッチ 7
前提条件 8
手順 8
関連情報 9
2.5. RED HAT CEPH ストレージリポジトリーの有効化 9
前提条件 9
手順 9
関連情報 10
2.6. OSD ノードで RAID コントローラーを使用する際の考慮事項 (オプション) 10
2.7. OBJECT GATEWAY で NVME を使用する際の考慮事項 (オプション) 10
2.8. RED HAT CEPH STORAGE のネットワーク構成の確認 10
前提条件 11
手順 11
関連情報 11
2.9. RED HAT CEPH STORAGE のファイアウォールの設定 11
2.10. SUDO アクセスのある ANSIBLE ユーザーの作成 15
2.11. ANSIBLE でパスワードなしの SSH を有効にする 17
前提条件 17
手順 17
関連情報 18

第3章 RED HAT CEPH STORAGE の導入 20
3.1. 前提条件 20
3.2. RED HAT CEPH STORAGE クラスターのインストール 20
前提条件 21
手順 21
3.3. すべての NVME ストレージに OSD ANSIBLE 設定の構成 34
3.4. メタデータサーバのインストール 35
3.5. CEPH クライアントロールのインストール 36
前提条件 36
手順 36
関連情報 37
3.6. CEPH OBJECT GATEWAY のインストール 38
前提条件 38
手順 38
関連情報 39
3.6.1. マルチサイト Ceph Object Gateway の設定 39
3.7. NFS-GANESHA ゲートウェイのインストール 42
前提条件 42
手順 42
関連情報 43
3.8. LIMIT オプションについて 43
3.9. 関連情報 44

第4章 RED HAT CEPH STORAGE CLUSTER のアップグレード 45
前提条件 46

4.1. ストレージクラスターのアップグレード 47
手順 47
4.2. RED HAT CEPH STORAGE DASHBOARD のアップグレード 51

第5章 次のステップ 53

付録A トラブルシューティング 54
A.1. ANSIBLE は、予想よりも少ないデバイスを検出するため、インストールを停止します 54

付録B RED HAT CEPH STORAGE の手動インストール 55
B.1. 前提条件 55
Red Hat Ceph Storage のネットワークタイムプロトコルの設定 55
前提条件 55
手順: RHCS のネットワークタイムプロトコルを構成する 55
関連情報 56
ブートストラップの監視 56
B.2. CEPH MANAGER の手動インストール 62
OSD ブート制約 63

付録C CEPH コマンドラインインターフェースのインストール 69
前提条件 69
手順 69

付録D CEPH ブロックデバイスの手動インストール 70
前提条件 70
手順 70

付録E CEPH OBJECT GATEWAY の手動インストール 73
前提条件 73
手順 73
詳細 75

付録F CEPH のデフォルト設定の上書き 76

付録G RED HAT CEPH STORAGE 2 から 3 への手動アップグレード 77
モニタノードのアップグレード 78
手順 78
G.1. CEPH MANAGER の手動インストール 79
OSD ノードのアップグレード 81
前提条件 81
手順 81
関連情報 84
Ceph Object Gateway ノードのアップグレード 84
前提条件 84
手順 84
関連項目 85
Ceph クライアントノードのアップグレード 85
前提条件 85
手順 85

付録H バージョン 2 と 3 の間の ANSIBLE 変数の変更 87

付録I 既存の CEPH クラスターの ANSIBLE へのインポート 88

付録J ANSIBLE を使用して CEPH クラスターをパージする 89

第1章 RED HAT CEPH STORAGE とは

Red Hat Ceph Storage はスケーラブルでオープンなソフトウェア定義のストレージプラットフォームで、Ceph ストレージシステムの最も安定したバージョンと、Ceph 管理プラットフォーム、デプロイメントユーティリティー、サポートサービスを組み合わせたものです。

Red Hat Ceph Storage は、クラウドインフラストラクチャーおよび Web スケールオブジェクトストレージ用に設計されています。Red Hat Ceph Storage クラスターは、以下のタイプのノードで構成されます。

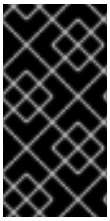
Red Hat Ceph Storage Ansible 管理ノード

このタイプのノードは、以前のバージョンの Red Hat Ceph Storage に行われた従来の Ceph 管理ノードとして機能します。このタイプのノードでは、以下の機能が提供されます。

- ストレージクラスターの一元管理
- Ceph 設定ファイルおよびキー
- 必要に応じて、セキュリティー上の理由からインターネットにアクセスできないノードに Ceph をインストールするためのローカルリポジトリー

ノードの監視

各モニターノードは、クラスターマップのマスターコピーを維持する monitor デーモン (**ceph-mon**) を実行します。クラスターマップにはクラスタートポロジーが含まれます。Ceph クラスターに接続するクライアントは、モニターからクラスターマップの現在のコピーを取得します。これにより、クライアントがクラスターへのデータの読み取りおよび書き込みが可能になります。



重要

Ceph は 1 つのモニターで実行できますが、実稼働クラスターで高可用性を確保するためには、Red Hat は少なくとも 3 つのモニターノードを持つデプロイメントのみをサポートします。Red Hat は、750 個の OSD を超えるストレージクラスターに合計 5 つの Ceph Monitor をデプロイすることを推奨します。

OSD ノード

各 Object Storage Device (OSD) ノードは Ceph OSD デーモン (**ceph-osd**) を実行し、ノードに割り当てられている論理ディスクと相互作用します。Ceph は、この OSD ノードにデータを保存します。
Ceph は、非常に少数の OSD ノード (デフォルトは 3) で実行できますが、実稼働クラスターは、ストレージクラスターで中程度のスケール (たとえば OSD が 50 個) で始まります。理想的には、Ceph クラスターに複数の OSD ノードがあり、CRUSH マップを作成して分離された障害ドメインを許可します。

MDS ノード

各 Metadata Server (MDS) ノードは、Ceph ファイルシステム (CephFS) に保存されているファイルに関連する MDS デーモン (**ceph-mds**) を実行します。MDS デーモンは、共有クラスターへのアクセスも調整します。

Object Gateway ノード

Ceph Object Gateway ノードは、Ceph RADOS Gateway デーモン (**ceph-radosgw**) を実行し、Ceph Storage クラスターへの RESTful ゲートウェイを使用するアプリケーションを提供する **librados** 上に構築されたオブジェクトストレージインターフェースです。Ceph Object Gateway は、以下の 2 つのインターフェースをサポートします。

S3

Amazon S3 RESTful API の大規模なサブセットと互換性のあるインターフェースでオブジェクトストレージ機能を提供します。

Swift

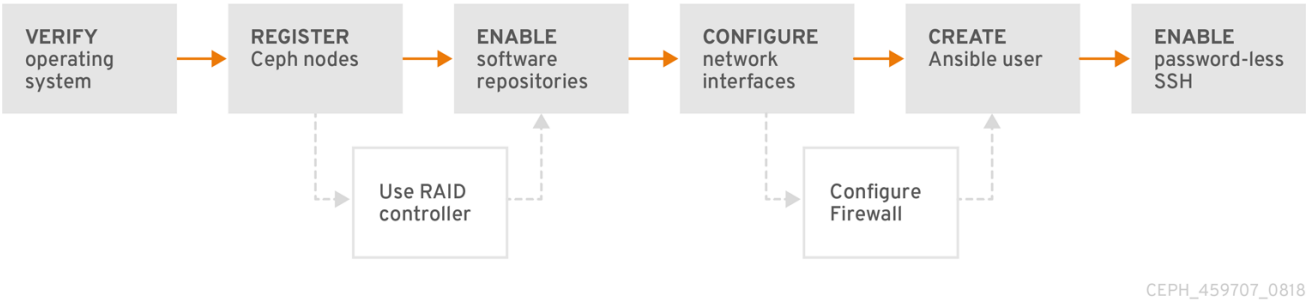
OpenStack Swift API の大規模なサブセットと互換性のあるインターフェースでオブジェクトストレージ機能を提供します。

Ceph アーキテクチャーの詳細は、Red Hat Ceph Storage3 の**アーキテクチャーガイド** を参照してください。

最低限推奨されるハードウェアは、Red Hat Ceph ストレージハードウェア選択ガイド 3 を参照してください。

第2章 RED HAT CEPH STORAGE のインストール要件

図2.1 前提条件のワークフロー



Red Hat Ceph Storage (RHCS) をインストールする前に、以下の要件をチェックして、各 Monitor、OSD、メタデータサーバー、およびクライアントノードを適宜準備します。

2.1. 前提条件

- ハードウェアが最低必要条件を満たしていることを確認してください。詳細については、Red Hat Ceph Storage3 のハードウェアガイドを参照してください。

2.2. RED HAT CEPH STORAGE をインストールするための要件チェックリスト

タスク	必須	セクション	推奨事項
オペレーティングシステムのバージョンの確認	はい	「Red Hat Ceph Storage のオペレーティングシステム要件」	
Ceph ノードの登録	はい	「Red Hat Ceph Storage ノードの CDN への登録とサブスクリプションのアタッチ」	
Ceph ソフトウェアリポジトリーの有効化	はい	「Red Hat Ceph ストレージリポジトリーの有効化」	
OSD ノードでの RAID コントローラーの使用	いいえ	「OSD ノードで RAID コントローラーを使用する際の考慮事項 (オプション)」	RAID コントローラーでライトバックキャッシュを有効にすると、OSD ノードの小規模な I/O 書き込みスループットが増大する場合があります。
ネットワークの設定	はい	「Red Hat Ceph Storage のネットワーク構成の確認」	少なくとも、パブリックネットワークが必要です。ただし、クラスター通信用のプライベートネットワークが推奨されます。

タスク	必須	セクション	推奨事項
ファイアウォールの設定	いいえ	「Red Hat Ceph Storage のファイアウォールの設定」	ファイアウォールは、ネットワークの信頼レベルを大きくすることができます。
Ansible ユーザーの作成	はい	「**sudo** アクセスのある Ansible ユーザーの作成」	すべての Ceph ノードで Ansible ユーザーを作成する必要があります。
パスワードを使用しない SSH の有効化	はい	「Ansible でパスワードなしの SSH を有効にする」	Ansible で必須。

注記

デフォルトでは、**ceph-ansible** は要件として NTP をインストールします。NTP がカスタマイズされている場合は、Red Hat Ceph Storage の手動インストール **の Red Hat Ceph Storage のネットワークタイムプロトコルの構成** を参照して、Ceph で正しく機能するように NTP を構成する方法を理解してください。

2.3. RED HAT CEPH STORAGE のオペレーティングシステム要件

Red Hat Ceph Storage 3 には、Red Hat Enterprise Linux 7, update 5 以降が必要です。クラスター内のすべてのノードで同じバージョンとアーキテクチャを使用してください。

重要

Red Hat Ceph Storage 3 は Red Hat Enterprise Linux 8 ではサポートされていません。

重要

Red Hat は異種のオペレーティングシステムやバージョンを使用したクラスターをサポートしていません。

関連情報

- Red Hat Enterprise Linux 7 の インストールガイド
- Red Hat Enterprise Linux 7 のシステム管理者ガイド

要求事項のチェックリストに戻る

2.4. RED HAT CEPH STORAGE ノードの CDN への登録とサブスクリプションのアタッチ

各 Red Hat Ceph Storage (RHCS) ノードをコンテンツ配信ネットワーク (CDN) に登録し、ノードがソフトウェアリポジトリーにアクセスできるように適切なサブスクリプションを割り当てます。各 RHCS ノードは、Red Hat Enterprise Linux 7 のベースコンテンツとエクストラリポジトリのコンテンツに完全にアクセスできる必要があります。

注記

インストール中にインターネットにアクセスできない RHCS ノードの場合は、Red Hat Satellite サーバーを使用してソフトウェアコンテンツを提供します。または、ローカルの Red Hat Enterprise Linux 7 Server ISO イメージをマウントし、RHCS ノードが ISO イメージを指すようにします。詳細は、Red Hat サポート にお問い合わせください。

Red Hat Satellite サーバーに Ceph ノードの登録に関する詳細は、Red Hat カスタマーポータルの記事「How to Register Ceph with Satellite 6」 および「How to Register Ceph with Satellite 5」を参照してください。

前提条件

- 有効な Red Hat サブスクリプション
- RHCSのノードは、インターネットに接続できる必要があります。

手順

root ユーザーとして、ストレージクラスター内のすべてのノードで次の手順を実行します。

1. ノードを登録します。プロンプトが表示されたら、Red Hat カスタマーポータルの認証情報を入力します。

```
# subscription-manager register
```

2. CDN から最新のサブスクリプションデータをプルします。

```
# subscription-manager refresh
```

3. Red Hat Ceph Storage で利用可能なサブスクリプションの一覧を表示します。

```
# subscription-manager list --available --all --matches="*Ceph*"
```

適切なサブスクリプションを特定し、プール ID を取得します。

4. サブスクリプションを割り当てます。

```
# subscription-manager attach --pool=$POOL_ID
```

置き換え

- **POOL_ID** には、前のステップで特定したプールIDを指定します。

5. デフォルトのソフトウェアリポジトリーを無効にします。その後、Red Hat Enterprise Linux 7 Server、Red Hat Enterprise Linux 7 Server Extras、RHCS の各リポジトリーを有効にします。

```
# subscription-manager repos --disable=*
# subscription-manager repos --enable=rhel-7-server-rpms
# subscription-manager repos --enable=rhel-7-server-extras-rpms
```

```
# subscription-manager repos --enable=rhel-7-server-rhceph-3-mon-els-rpms
# subscription-manager repos --enable=rhel-7-server-rhceph-3-osd-els-rpms
# subscription-manager repos --enable=rhel-7-server-rhceph-3-tools-els-rpms
```

6. 最新のパッケージを受け取るようにシステムをアップデートします。

```
# yum update
```

関連情報

- Red Hat Enterprise Linux 7 のシステム管理者ガイドのシステムの登録とサブスクリプションの管理の章を参照してください。
- 「Red Hat Ceph ストレージリポジトリーの有効化」

要求事項のチェックリストに戻る

2.5. RED HAT CEPH ストレージリポジトリーの有効化

Red Hat Ceph Storage をインストールする前に、インストール方法を選択する必要があります。Red Hat Ceph Storage では、以下の 2 つのインストール方法がサポートされます。

- コンテンツ配信ネットワーク (CDN)
  インターネットに直接接続可能な Ceph ノードを持つ Ceph Storage クラスターの場合は、Red Hat Subscription Manager を使用して必要な Ceph リポジトリーを有効にします。
- ローカルリポジトリー
  セキュリティー対策がインターネットにアクセスできない Ceph Storage クラスターでは、ISO イメージとして配信される単一のソフトウェアビルドから Red Hat Ceph Storage 3.3 をインストールします。これにより、ローカルリポジトリーをインストールできます。

前提条件

- 有効なカスタマーサブスクリプション
- CDN をインストールする場合、RHCS ノードはインターネットに接続できる必要があります。
- CDN インストールの場合は、CDN にクラスタノードを登録します。
- EPEL ソフトウェアリポジトリーを無効にします。

```
[root@monitor ~]# yum install yum-utils vim -y
[root@monitor ~]# yum-config-manager --disable epel
```

手順

CDN インストールの場合:

Ansible 管理ノード で、Red Hat Ceph Storage 3 Tools リポジトリーおよび Ansible リポジトリーを有効にします。

```
[root@admin ~]# subscription-manager repos --enable=rhel-7-server-rhceph-3-tools-els-rpms --
enable=rhel-7-server-ansible-2.6-rpms
```

ISO インストールの場合:

1. Red Hat カスタマーポータルにログインします。
2. **Downloads** をクリックして、**Software & Download** センターに移動します。
3. Red Hat Ceph Storage エリアで **Download Software** をクリックして、最新バージョンのソフトウェアをダウンロードします。

関連情報

- Red Hat Enterprise Linux の管理者ガイドのシステムの登録とサブスクリプションの管理の章

要件チェックリストに戻る

2.6. OSD ノードで RAID コントローラーを使用する際の考慮事項 (オプション)

OSD ノードに 1~2 GB のキャッシュがインストールされている RAID コントローラーがある場合は、ライトバックキャッシュを有効にすると、I/O 書き込みスループットが向上する可能性があります。ただし、キャッシュは不揮発性である必要があります。

最近の RAID コントローラーには通常、電力損失イベント中に揮発性メモリーを不揮発性 NAND メモリーに排出するのに十分な電力を供給するスーパーキャパシタがあります。電源の復旧後に、特定のコントローラーとそのファームウェアがどのように動作するかを理解することが重要です。

RAID コントローラーによっては、手動の介入が必要になります。ハードドライブは、ディスクキャッシュをデフォルトで有効または無効にすべきかどうかに関わらず、オペレーティングシステムにアドバタイズします。ただし、特定の RAID コントローラーとファームウェアは、このような情報を提供しません。ファイルシステムが破損しないように、ディスクレベルのキャッシュが無効になっていることを確認します。

ライトバックキャッシュを有効にして、各 Ceph OSD データドライブにライトバックを設定して、単一の RAID 0 ボリュームを作成します。

Serial Attached SCSI (SAS) または SATA 接続の Solid-state Drive (SSD) ディスクも RAID コントローラーに存在する場合は、コントローラーとファームウェアが **pass-through** モードをサポートしているかどうかを確認します。**pass-through** モードを有効にすると、キャッシュロジックが回避され、通常は高速メディアの待ち時間が大幅に低くなります。

要求事項のチェックリストに戻る

2.7. OBJECT GATEWAY で NVME を使用する際の考慮事項 (オプション)

Red Hat Ceph Storage の Object Gateway 機能を使用する予定で、OSD ノードに NVMe ベースの SSD または SATA SSD がある場合は、Ceph Object Gateway for Production の手順に従って、LVM で NVMe を最適に使用することを検討してください。これらの手順では、ジャーナルとバケットインデックスを SSD に一緒に配置する特別に設計された Ansible Playbook の使用方法を説明します。これにより、すべてのジャーナルを 1 つのデバイスに配置する場合に比べてパフォーマンスを向上させることができます。NVMe と LVM を最適に使用するための情報は、本インストールガイドと併せて参照してください。

要求事項のチェックリストに戻る

2.8. RED HAT CEPH STORAGE のネットワーク構成の確認

すべてのRed Hat Ceph Storage (RHCS) ノードには、パブリックネットワークが必要です。Ceph クライアントが Ceph monitor ノードおよび Ceph OSD ノードに到達できるパブリックネットワークにネットワークインターフェースカードが設定されている必要があります。

Ceph がパブリックネットワークとは別のネットワークでハートビート、ピアリング、レプリケーション、および復元を実行できるように、クラスターネットワーク用のネットワークインターフェイスカードがある場合があります。

ネットワークインターフェースを設定し、変更を永続化します。

重要

Red Hat では、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に単一のネットワークインターフェースカードを使用することは推奨していません。

前提条件

- ネットワークに接続されたネットワークインターフェースカード。

手順

ストレージクラスターのすべての RHCSノードで、**root**ユーザーとして以下の手順を実行します。

1. 以下の設定が、公開されているネットワークインターフェースカードに対応する **/etc/sysconfig/network-scripts/ifcfg-*** ファイルにあることを確認します。

   a. 静的 IP アドレスについて **BOOTPROTO** パラメーターは **none** に設定されます。

   b. **ONBOOT** パラメーターは **yes** に設定する必要があります。
   これが **no** に設定されていると、Ceph ストレージクラスターがリブート時にピアに機能しなくなる可能性があります。

   c. IPv6 アドレス指定を使用する場合には、**IPV6_FAILURE_FATAL** パラメーターを除き、**IPV6INIT** などの IPV6 パラメーターを **yes** に設定する必要があります。
   また、Ceph 構成ファイル/etc/ceph/ceph.conf を編集して、IPv6 を使用するように **Ceph** に指示します。そうでない場合、Ceph は IPv4 を使用します。

関連情報

- Red Hat Enterprise Linux 7 のネットワークインターフェイススクリプトの設定の詳細については、 Red Hat Enterprise Linux 7 の**ネットワークガイド** の **ifcfg ファイルを使用したネットワークインターフェイスの設定**の章を参照してください。

- ネットワーク設定の詳細は、Red Hat Ceph Storage 3 の『**設定ガイド**』の「**ネットワーク設定参照**」の章を参照してください。

要求事項のチェックリストに戻る

2.9. RED HAT CEPH STORAGE のファイアウォールの設定

Red Hat Ceph Storage (RHCS) は **firewalld** サービスを使用します。

Monitor デーモンは、Ceph Storage クラスター内の通信にポート **6789** を使用します。

各 Ceph OSD ノードで、OSD デーモンは範囲 **6800-7300** 内の複数のポートを使用します。

- パブリックネットワークを介してクライアントおよびモニターと通信するための 1 つ

- クラスターネットワーク上で他の OSD にデータを送信する 1 つ (利用可能な場合)。それ以外の場合は、パブリックネットワーク経由でデータを送信します。
- 可能な場合は、クラスターネットワークを介してハートビートパケットを交換するための 1 つ。それ以外の場合は、パブリックネットワーク経由

Ceph Manager (**ceph-mgr**) デーモンは、**6800-7300** 範囲内のポートを使用します。同じノード上で Ceph Monitor と **ceph-mgr** デーモンを共存させることを検討してください。

Ceph Metadata Server ノード**(ceph-mds)** は、**6800〜7300** の範囲のポートを使用します。

Ceph Object Gateway ノードは、デフォルトで **8080** を使用するように Ansible によって設定されます。ただし、デフォルトのポート (例: ポート **80**) を変更できます。

SSL/TLS サービスを使用するには、ポート **443** を開きます。

前提条件

- ネットワークハードウェアが接続されている。

手順

root ユーザーで以下のコマンドを実行してください。

1. すべての RHCS ノードで **firewalld** サービスを開始します。これを有効にして、システムの起動時に実行し、実行していることを確認します。

```
# systemctl enable firewalld
# systemctl start firewalld
# systemctl status firewalld
```

2. すべての Monitor ノードで、パブリックネットワークの **6789** ポートを開く。

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

ソースアドレスに基づいてアクセスを制限するには、以下を実行します。

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="6789" accept"
```

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="6789" accept" --permanent
```

置き換え

- **IP_address** には、Monitor ノードのネットワークアドレスを指定します。
- **netmask_prefix** には、CIDR 表記のネットマスクを指定します。

例

```
[root@monitor ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.11/24" port protocol="tcp" \
port="6789" accept"
```

```
[root@monitor ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.11/24" port protocol="tcp" \
port="6789" accept" --permanent
```

3. すべての OSD ノードで、パブリックネットワークでポート **6800-7300** を開きます。

```
[root@osd ~]# firewall-cmd --zone=public --add-port=6800-7300/tcp
[root@osd ~]# firewall-cmd --zone=public --add-port=6800-7300/tcp --permanent
```

別のクラスターネットワークがある場合には、適切なゾーンでコマンドを繰り返します。

4. すべての Ceph Manager (**ceph-mgr**) ノード (通常はMonitorのノードと同じ) で、パブリックネットワークの **6800〜7300** 番ポートを開きます。

```
[root@monitor ~]# firewall-cmd --zone=public --add-port=6800-7300/tcp
[root@monitor ~]# firewall-cmd --zone=public --add-port=6800-7300/tcp --permanent
```

別のクラスターネットワークがある場合には、適切なゾーンでコマンドを繰り返します。

5. すべての Ceph Metadata Server (**ceph-mds**) ノードにおいて、パブリックネットワークでポート **6800** を開きます。

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

別のクラスターネットワークがある場合には、適切なゾーンでコマンドを繰り返します。

6. すべての Ceph Object Gateway ノードで、パブリックネットワーク上の関連するポートを開きます。

   a. デフォルトの Ansible が設定されたポート **8080** を開くには、以下のコマンドを実行します。

```
[root@gateway ~]# firewall-cmd --zone=public --add-port=8080/tcp
[root@gateway ~]# firewall-cmd --zone=public --add-port=8080/tcp --permanent
```

ソースアドレスに基づいてアクセスを制限するには、以下を実行します。

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="8080" accept"
```

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="8080" accept" --permanent
```

置き換え

- オブジェクトゲートウェイノードのネットワークアドレスを含む **IP_address**。

- **netmask_prefix** には、CIDR 表記のネットマスクを指定します。

例

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="8080" accept"
```

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="8080" accept" --permanent
```

b. 任意。Ansible を使用して Ceph Object Gateway をインストールし、使用する Ceph Object Gateway を Ansible が構成するデフォルトのポートを **8080** からポート **80** に変更した場合は、次のポートを開きます。

```
[root@gateway ~]# firewall-cmd --zone=public --add-port=80/tcp
[root@gateway ~]# firewall-cmd --zone=public --add-port=80/tcp --permanent
```

ソースアドレスに基づいてアクセスを制限するには、以下のコマンドを実行します。

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="80" accept"
```

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="80" accept" --permanent
```

置き換え

- オブジェクトゲートウェイノードのネットワークアドレスを含む **IP_address**。
- **netmask_prefix** には、CIDR 表記のネットマスクを指定します。

例

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="80" accept"
```

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="80" accept" --permanent
```

c. 任意。SSL/TLS を使用するには、**443** ポートを開きます。

```
[root@gateway ~]# firewall-cmd --zone=public --add-port=443/tcp
[root@gateway ~]# firewall-cmd --zone=public --add-port=443/tcp --permanent
```

ソースアドレスに基づいてアクセスを制限するには、以下のコマンドを実行します。

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="443" accept"
```

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_address/netmask_prefix" port protocol="tcp" \
port="443" accept" --permanent
```

置き換え

- オブジェクトゲートウェイノードのネットワークアドレスを含む **IP_address**。
- **netmask_prefix** には、CIDR 表記のネットマスクを指定します。

例

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="443" accept"
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="443" accept" --permanent
```

関連情報

- パブリックネットワークとクラスターネットワークの詳細は、Red Hat Ceph Storage のネットワーク構成の確認を参照してください。
- **Firewalld** の詳細は、 Red Hat Enterprise Linux 7 のセキュリティーガイドの ファイアウォールの使用の章を参照してください。

要求事項のチェックリストに戻る

2.10. SUDO アクセスのある ANSIBLE ユーザーの作成

Ansible は、ソフトウェアをインストールし、パスワードを要求せずに設定ファイルを作成するための **root** 権限を持つユーザーとして、すべての Red Hat Ceph Storage (RHCS) ノードにログインできる必要があります。Ansible を使用して Red Hat Ceph Storage クラスターをデプロイおよび設定する際に、ストレージクラスター内のすべてのノードにパスワードなしの **root** アクセスで Ansible ユーザーを作成する必要があります。

前提条件

- ストレージクラスター内のすべてのノードへの **root** または **sudo** アクセスがある。

手順

1. Ceph ノードに **root** ユーザとしてログインします。

```
ssh root@$HOST_NAME
```

置き換え

- Ceph ノードのホスト名で **$HOST_NAME**。

例

```
# ssh root@mon01
```

プロンプトに従い **root** パスワードを入力します。

2. 新しい Ansible ユーザーを作成します。

```
adduser $USER_NAME
```

置き換え

- **$USER_NAME** には、Ansible ユーザーの新しいユーザー名を指定します。

例

```
# adduser admin
```

重要

ceph をユーザー名として使用しないでください。**ceph** ユーザー名は、Ceph デーモン用に予約されます。クラスター全体で統一されたユーザー名を使用すると、使いやすさが向上しますが、侵入者は通常、そのユーザー名をブルートフォース攻撃に使用するため、明白なユーザー名の使用は避けてください。

3. このユーザーに新しいパスワードを設定します。

```
# passwd $USER_NAME
```

置き換え

- **$USER_NAME** には、Ansible ユーザーの新しいユーザー名を指定します。

例

```
# passwd admin
```

プロンプトが表示されたら、新しいパスワードを 2 回入力します。

4. 新規に作成されたユーザーの **sudo** アクセスを設定します。

```
cat << EOF >/etc/sudoers.d/$USER_NAME
$USER_NAME ALL = (root) NOPASSWD:ALL
EOF
```

置き換え

- **$USER_NAME** には、Ansible ユーザーの新しいユーザー名を指定します。

例

```
# cat << EOF >/etc/sudoers.d/admin
admin ALL = (root) NOPASSWD:ALL
EOF
```

5. 正しいファイル権限を新しいファイルに割り当てます。

```
chmod 0440 /etc/sudoers.d/$USER_NAME
```

置き換え

- **$USER_NAME** には、Ansible ユーザーの新しいユーザー名を指定します。

例

```
# chmod 0440 /etc/sudoers.d/admin
```

関連情報

- Red Hat Enterprise Linux7 の**システム管理者ガイド**の新しいユーザーの追加セクション。

要件チェックリストに戻る

2.11. ANSIBLE でパスワードなしの SSH を有効にする

Ansible 管理ノードで SSH キーペアを生成し、ストレージクラスター内の各ノードに公開キーを配布して、Ansible がパスワードの入力を求められることなくノードにアクセスできるようにします。

前提条件

- **sudo** アクセス権を持つ Ansible ユーザーを作成

手順
Ansible 管理ノードから、Ansible ユーザーとして次の手順を実行します。

1. SSH キーペアを生成し、デフォルトのファイル名を受け入れ、パスフレーズを空のままにします。

```
[user@admin ~]$ ssh-keygen
```

2. 公開鍵をストレージクラスター内のすべてのノードにコピーします。

```
ssh-copy-id $USER_NAME@$HOST_NAME
```

置き換え

- **$USER_NAME** には、Ansible ユーザーの新しいユーザー名を指定します。
- Ceph ノードのホスト名で **$HOST_NAME**。

例

```
[user@admin ~]$ ssh-copy-id admin@ceph-mon01
```

3. ~/**.ssh**/**config** ファイルを作成および編集します。

重要

~/**.ssh**/**config** ファイルを作成および編集することで、**ansible-playbook** コマンドを実行するたびに **-u $USER_NAME** オプションを指定する必要はありません。

a. SSH **config** ファイルを作成します。

```
[user@admin ~]$ touch ~/.ssh/config
```

b. 編集のために **config** ファイルを開きます。ストレージクラスターの各ノードの **Hostname** と **User** オプションを設定します。

```
Host node1
  Hostname $HOST_NAME
  User $USER_NAME
Host node2
  Hostname $HOST_NAME
  User $USER_NAME
...
```

置き換え

- Ceph ノードのホスト名で **$HOST_NAME**。
- **$USER_NAME** には、Ansible ユーザーの新しいユーザー名を指定します。

例

```
Host node1
  Hostname monitor
  User admin
Host node2
  Hostname osd
  User admin
Host node3
  Hostname gateway
  User admin
```

4. ~/**.ssh**/**config** ファイルに正しいファイルパーミッションを設定します。

```
[admin@admin ~]$ chmod 600 ~/.ssh/config
```

関連情報

- **ssh_config(5)** の man ページ
- Red Hat Enterprise Linux7 の **システム管理者ガイド** の Open SSH の章

要求事項のチェックリストに戻る

第3章 RED HAT CEPH STORAGE の導入

本章では、Ansible アプリケーションを使用して Red Hat Ceph Storage クラスターおよびその他のコンポーネントをデプロイする方法を説明します (メタデータサーバーや Ceph Object Gateway など)。

- Red Hat Ceph Storage クラスターをインストールするには、「Red Hat Ceph Storage クラスターのインストール」 を参照してください。
- メタデータサーバをインストールするには、「メタデータサーバのインストール」 を参照してください。
- **ceph-client** ロールをインストールするには、「Ceph クライアントロールのインストール」 を参照してください。
- Ceph Object Gatewayをインストールするには、「Ceph Object Gateway のインストール」 を参照してください。
- マルチサイトの Ceph Object Gateway を設定するには、「マルチサイト Ceph Object Gateway の設定」 を参照してください。
- Ansibleの **--limit** オプションについては、「**limit** オプションについて」 を参照してください。

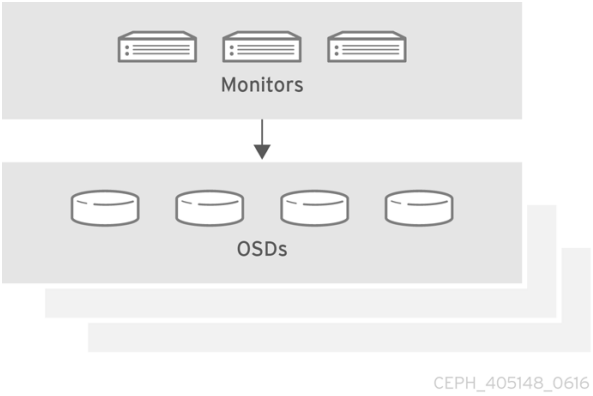
3.1. 前提条件

- 有効なカスタマーサブスクリプションを取得します。
- クラスタノードの準備各ノードで
  - コンテンツ配信ネットワーク (CDN) にノードを登録し、サブスクリプションを割り当てます。
  - 適切なソフトウェアリポジトリーを有効にします。
  - Ansible ユーザーを作成しします。
  - パスワードなしの SSH アクセスの有効化。
  - 任意。ファイアウォールの設定

3.2. RED HAT CEPH STORAGE クラスターのインストール

ceph-ansible playbook 付きの Ansible アプリケーションを使用して、Red Hat Ceph Storage 3 をインストールします。

本番の Ceph ストレージクラスターは、最低でも 3 台のモニターホストと、複数の OSD デーモンを搭載した 3 台の OSD ノードが必要です。



前提条件

- Ansible 管理ノードの root アカウントを使用して、**ceph-ansible** パッケージをインストールします。

```
[root@admin ~]# yum install ceph-ansible
```

手順
指示がない限り、Ansible の管理ノードから以下のコマンドを実行します。

1. Ansibleのユーザーとして、**ceph-ansible** Playbook で生成された一時的な値をAnsibleが保存する**ceph-ansible-keys** ディレクトリーを作成します。

```
[user@admin ~]$ mkdir ~/ceph-ansible-keys
```

2. root として、**/etc/ansible/** ディレクトリーに **/usr/share/ceph-ansible/group_vars** ディレクトリーへのシンボリックリンクを作成します。

```
[root@admin ~]# ln -s /usr/share/ceph-ansible/group_vars /etc/ansible/group_vars
```

3. **/usr/share/ceph-ansible** ディレクトリーに移動します。

```
[root@admin ~]$ cd /usr/share/ceph-ansible
```

4. **yml.sample** ファイルのコピーを新たに作成します。

```
[root@admin ceph-ansible]# cp group_vars/all.yml.sample group_vars/all.yml
[root@admin ceph-ansible]# cp group_vars/osds.yml.sample group_vars/osds.yml
[root@admin ceph-ansible]# cp site.yml.sample site.yml
```

5. コピーしたファイルを編集します。

   a. **group_vars/all.yml** ファイルを編集します。アンコメントする最も一般的な必須およびオプションのパラメータについては、以下の表を参照してください。なお、この表にはすべてのパラメータが含まれているわけではありません。

重要

カスタムクラスター名の使用はサポートされていないため、**cluster: ceph** パラメータに **ceph** 以外の値を設定しないでください。

表3.1 Ansible の一般的な設定

オプション	値	必須	備考
ceph_origin	**repository** または **distro** または **local**	はい	**repository** 値は、新しいリポジトリーで Ceph をインストールすることを意味します。**distro** の値は、個別のリポジトリーファイルが追加されず、Linux ディストリビューションに含まれる Ceph のバージョンをすべて取得することを意味します。**local** の値は、Ceph バイナリーがローカルマシンからコピーされることを意味します。
ceph_repository_type	**cdn** または **iso**	はい	
ceph_rhcs_version	**3**	はい	
ceph_rhcs_iso_path	ISO イメージへのパス	ISO イメージを使用する場合は Yes	
monitor_interface	Monitor ノードがリッスンするインターフェース	**MONITOR_INTERFACE**、**MONITOR_ADDRESS**、または **MONITOR_ADDRESS_BLOCK** が必要です。	
monitor_address	Monitor ノードがリッスンするアドレス		
monitor_address_block	Ceph のパブリックネットワークのサブネット		ノードの IP アドレスは不明だが、サブネットはわかっている場合に使用します
ip_version	**ipv6**	IPv6 アドレスを使用している場合は Yes	

オプション	値	必須	備考
public_network	IPv6 を使用する場合には、Ceph パブリックネットワークの IP アドレスとネットマスク、または対応する IPv6 アドレス	はい	「Red Hat Ceph Storage のネットワーク構成の確認」
cluster_network	Ceph クラスターネットワークの IP アドレスとネットマスク	いいえ、デフォルトは **public_network** です	
configure_firewall	Ansible は適切なファイアウォールルールの設定を試みます	いいえ。値を **true** または **false** に設定します。	

all.yml ファイルの例は次のようになります。

```
ceph_origin: distro
ceph_repository: rhcs
ceph_repository_type: cdn
ceph_rhcs_version: 3
monitor_interface: eth0
public_network: 192.168.0.0/24
```

注記

必ず **all.yml** ファイルで **ceph_origin** を **distro** に設定してください。これにより、インストールプロセスで正しいダウンロードリポジトリが使用されるようになります。

注記

ceph_rhcs_version オプションを **3** に設定すると、最新バージョンの Red Hat Ceph Storage 3 がプルされます。

警告

デフォルトでは、Ansible はインストール済みの再起動を試みますが、マスクされた **firewalld** サービスにより、Red Hat Ceph Storage デプロイメントが失敗する可能性があります。この問題を回避するには、**all.yml** ファイルで **configure_firewall** オプションを **false** に設定します。**firewalld** サービスを実行している場合は、**all.yml** ファイルで **configure_firewall** オプションを使用する必要はありません。

詳細は、**all.yml** ファイルを参照してください。

b. **group_vars/osds.yml** ファイルを編集します。アンコメントする最も一般的な必須およびオプションのパラメータについては、以下の表を参照してください。なお、この表にはすべてのパラメータが含まれているわけではありません。

重要

OSD のインストールには、OSD がインストールされている機器とは異なる物理的な機器を使用してください。オペレーティングシステムと OSD 間で同じデバイスを共有すると、パフォーマンスの問題が発生することになります。

表3.2 OSD Ansible 設定

オプション	値	必須	備考
osd_scenario	**collocated**。ログ先行書き込みとキー/値データ (Blue Store) またはジャーナル (File Store) と OSD データに同じデバイスを使用 **non-collocated**。SSD や NVMe メディアなどの専用デバイスを使用して先行書き込みログとキー/値データ (Blue Store) またはジャーナルデータ(File Store)を保存するため **lvm**: OSDデータの保存に論理ボリュームマネージャを使用する場合	はい	**osd_scenario:non-collocated**、**ceph-ansible** を使用する場合、**devices** と **dedicated_devices** の変数の数が一致することを期待します。たとえば、**devices** で 10 個のディスクを指定する場合は、**dedicated_devices** で 10 個のエントリーを指定する必要があります。
osd_auto_discovery	OSD を自動的に検出する場合は **true**	**osd_scenario: collocated** を使用している場合は Yes	**devices** 設定を使用している場合は使用できません。

オプション	値	必須	備考
devices	**ceph data** が保存されているデバイスのリスト	デバイスのリストを指定する場合は Yes	**osd_auto_discovery** 設定を使用する場合は使用できません。**osd_scenario** として **lvm** を使用し、**devices** オプションを設定する場合、**ceph-volume lvm batch** モードは最適化された OSD 構成を作成します。
dedicated_devices	**ceph journal** が保存されている非コロケーション OSD 専用デバイスのリスト	**osd_scenario: non-collocated** 場合は Yes	非分割型のデバイスであること
dmcrypt	OSD を暗号化する場合は **true**	いいえ	デフォルトは **false**

オプション	値	必須	備考
lvm_volumes	File Store または Blue Store 辞書のリスト	**osd_scenario: lvm**使用している場合、ストレージ デバイスが **devices** を使用して定義されている場合は Yes	各ディクショナリーには、**data** キー、**journal** キー、および **data_vg** キーが含まれている必要があります。論理ボリュームまたはボリュームグループはすべて、完全パスではなく、名前にする必要があります。**data** キーおよび **journal** キーは論理ボリューム (LV) またはパーティションにすることができますが、複数の **data** 論理ボリュームに 1 つのジャーナルを使用しないでください。**data_vg** キーは、**data** 論理ボリューム含むボリュームグループである必要があります。必要に応じて、**journal_vg** キーを使用して、ジャーナル LV を含むボリュームグループを指定できます (該当する場合)。サポートされているさまざまな構成については、以下の例を参照してください。
osds_per_device	デバイスごとに作成する OSD 数。	いいえ	デフォルトは **1** です
osd_objectstore	OSD の Ceph オブジェクトストアタイプ。	いいえ	デフォルトは **bluestore** です。もう一つの選択肢は、**filestore**です。アップグレードに必要です。

以下は、**collocated**、**non-collocated**、**lvm** の 3つの OSD シナリオを使用した場合の **osds.yml** ファイルの例です。指定されていない場合、デフォルトの OSD オブジェクトストアフォーマットは BlueStore です。

Collocated

```
osd_objectstore: filestore
osd_scenario: collocated
devices:
  - /dev/sda
  - /dev/sdb
```

コロケートされていない - BlueStore

```
osd_objectstore: bluestore
osd_scenario: non-collocated
devices:
 - /dev/sda
 - /dev/sdb
 - /dev/sdc
 - /dev/sdd
dedicated_devices:
 - /dev/nvme0n1
 - /dev/nvme0n1
 - /dev/nvme1n1
 - /dev/nvme1n1
```

コロケートされていない例では、デバイスごとに 1 つずつ、4 つの Blue Store OSD が作成されます。この例では、従来のハードドライブ (**sda**、**sdb**、**sdc**、**sdd**) がオブジェクトデータに使用され、ソリッドステートドライブ (SSD) (/ **dev**/**nvme0n1**、/ **dev**/**nvme1n1**) が Blue Store データベースに使用されて書き込みます-先行書き込みログ。この構成では、/**dev**/**sda** および/**dev**/**sdb** デバイスを /**dev**/**nvme0n1** デバイスとペアにし、/**dev**/**sdc** および /**dev**/**sdd** デバイスを / **dev**/**nvme1n1** デバイスとペアにします。

非コロケーション - Filestore

```
osd_objectstore: filestore
osd_scenario: non-collocated
devices:
  - /dev/sda
  - /dev/sdb
  - /dev/sdc
  - /dev/sdd
dedicated_devices:
   - /dev/nvme0n1
   - /dev/nvme0n1
   - /dev/nvme1n1
   - /dev/nvme1n1
```

LVM simple

```
osd_objectstore: bluestore
osd_scenario: lvm
devices:
  - /dev/sda
  - /dev/sdb
```

または

```
osd_objectstore: bluestore
osd_scenario: lvm
devices:
  - /dev/sda
  - /dev/sdb
  - /dev/nvme0n1
```

これらの単純な構成では **ceph-ansible** はバッチモード (**ceph-volume lvm batch**) を使用して OSD を作成します。

最初のシナリオでは、**devices** を従来のハードドライブまたは SSD の場合には、デバイスごとに OSD が 1 つ作成されます。

2 つ目のシナリオでは、従来のハードドライブと SSD が混在している場合、データは従来のハードドライブ (**sda**、**sdb**) に配置され、BlueStore データベース (**block.db**) は SSD (**nvme0n1**) にできる限り大きく作成されます。

LVM advance

```
osd_objectstore: filestore
osd_scenario: lvm
lvm_volumes:
   - data: data-lv1
     data_vg: vg1
     journal: journal-lv1
     journal_vg: vg2
   - data: data-lv2
     journal: /dev/sda
     data_vg: vg1
```

または

```
osd_objectstore: bluestore
osd_scenario: lvm
lvm_volumes:
  - data: data-lv1
    data_vg: data-vg1
    db: db-lv1
    db_vg: db-vg1
    wal: wal-lv1
    wal_vg: wal-vg1
  - data: data-lv2
    data_vg: data-vg2
    db: db-lv2
    db_vg: db-vg2
    wal: wal-lv2
    wal_vg: wal-vg2
```

これらの事前シナリオ例では、事前にボリュームグループと論理ボリュームを作成しておく必要があります。それらは **ceph-ansible** によって作成されません。

注記

すべての NVMe SSD を使用する場合は、**osd_scenario:lvm** および **osds_per_device:4** オプションを設定します。詳細は、Red Hat Ceph Storage **インストールガイド**の Red Hat Enterprise Linux 用の **すべての NVMe Storage の OSDAnsible 設定の構成**または **Ubuntu 用のすべての NVMe Storage の OSDAnsible設定の構成** を参照してください。

詳細は、**osds.yml** ファイルのコメントをご覧ください。

6. デフォルトでは **/etc/ansible/hosts** にある Ansible のインベントリファイルを編集します。サンプルホストをコメントアウトすることを忘れないでください。

   a. **[mons]** セクションの下に Monitor のノードを追加します。

   ```
   [mons]
   MONITOR_NODE_NAME1
   MONITOR_NODE_NAME2
   MONITOR_NODE_NAME3
   ```

   b. **[osds]** セクションの下に OSD ノードを追加します。ノードがシーケンシャルなネーミングの場合は、レンジの使用を検討してください。

   ```
   [osds]
   OSD_NODE_NAME1[1:10]
   ```

   注記

   新規インストールの OSD の場合、デフォルトのオブジェクトストア形式は BlueStore です。

   i. オプションで、**devices** および **dedicated_devices** オプションを使用して、OSD ノードが使用するデバイスを指定します。複数のデバイスをリストアップするには、コンマで区切ったリストを使用します。

   構文

   ```
   [osds]
   CEPH_NODE_NAME devices="['DEVICE_1', 'DEVICE_2']" dedicated_devices="
   ['DEVICE_3', 'DEVICE_4']"
   ```

   例

   ```
   [osds]
   ceph-osd-01 devices="['/dev/sdc', '/dev/sdd']" dedicated_devices="['/dev/sda',
   '/dev/sdb']"
   ceph-osd-02 devices="['/dev/sdc', '/dev/sdd', '/dev/sde']" dedicated_devices="
   ['/dev/sdf', '/dev/sdg']"
   ```

   デバイスを指定しない場合は、**osds.yml** ファイルの **osd_auto_discovery** オプションを **true** に設定してください。

注記

OSD が異なる名前の **デバイス** を使用する場合や、いずれかの OSD でデバイスのいずれかに障害が発生した場合に、devices および **dedicated_devices** パラメーターを使用すると便利です。

7. オプションで、すべてのデプロイメント(**ベアメタル** または **コンテナー** 内)でホスト固有のパラメーターを使用する場合は、ホストに固有のパラメーターを含めるために **host_vars** ディレクトリーにホストファイルを作成します。

   a. ストレージクラスターに追加される新しい Ceph OSD ノードを **/etc/ansible/host_vars/** ディレクトリーに作成します。

      構文

      ```
      touch /etc/ansible/host_vars/OSD_NODE_NAME
      ```

      例

      ```
      [root@admin ~]# touch /etc/ansible/host_vars/osd07
      ```

   b. ホスト固有のパラメーターで ファイルを更新します。**ベアメタル** デプロイメントでは、devices: および **dedicated_ devices:** セクションをファイルに追加できます。

      例

      ```
      devices:
        - /dev/sdc
        - /dev/sdd
        - /dev/sde
        - /dev/sdf

      dedicated_devices:
        - /dev/sda
        - /dev/sdb
      ```

8. オプションで、**ベアメタル**または**コンテナー内**のすべてのデプロイメントで、**ansible-playbook** を使用してカスタム CRUSH 階層を作成できます。

   a. Ansible のインベントリーファイルを設定します。**osd_crush_location** パラメーターを使用して、OSD ホストを CRUSH マップの階層内のどこに配置するかを指定します。OSD の場所を指定するために、2 つ以上の CRUSH バケットタイプを指定し、1 つのバケットの **type** をホストに指定する必要があります。デフォルトでは、これには、**root**、**datacenter**、**room**、**row**、**pod**、**pdu**、**rack**、**chassis** および **host** が含まれます。

      構文

      ```
      [osds]
      CEPH_OSD_NAME osd_crush_location="{ 'root': ROOT_BUCKET', 'rack':
      'RACK_BUCKET', 'pod': 'POD_BUCKET', 'host': 'CEPH_HOST_NAME' }"
      ```

      例

```
[osds]
ceph-osd-01 osd_crush_location="{ 'root': 'default', 'rack': 'rack1', 'pod': 'monpod', 'host':
'ceph-osd-01' }"
```

b. **crush_rule_config** パラメーターと**create_crush_tree** パラメーターを **True** に設定し、デフォルトの CRUSH ルールを使用しない場合は、少なくとも 1 つの CRUSH ルールを作成します。たとえば、**HDD** デバイスを使用している場合は、次のようにパラメーターを編集します。

```
crush_rule_config: True
crush_rule_hdd:
    name: replicated_hdd_rule
    root: root-hdd
    type: host
    class: hdd
    default: True
crush_rules:
  - "{{ crush_rule_hdd }}"
create_crush_tree: True
```

SSD デバイスを使用している場合は、以下のようにパラメーターを編集します。

```
crush_rule_config: True
crush_rule_ssd:
    name: replicated_ssd_rule
    root: root-ssd
    type: host
    class: ssd
    default: True
crush_rules:
  - "{{ crush_rule_ssd }}"
create_crush_tree: True
```

注記

ssd と **hdd** の両方の OSD がデプロイされていない場合、デフォルトの CRUSH ルールは失敗します。これは、デフォルトのルールに、定義する必要のある **class** パラメーターが含まれているためです。

注記

さらに、上記の手順で説明されているように、カスタムの CRUSH 階層を **host_vars** ディレクトリーの OSD ファイルに追加し、この設定を有効にします。

c. **group_vars/clients.yml** ファイルで作成した **crush_rules** を使用して **pools** を作成します。

例

>>>>>>> 3993c70c7f25ab628cbfd9c8e27623403ca18c99

```
copy_admin_key: True
user_config: True
```

```
pool1:
  name: "pool1"
  pg_num: 128
  pgp_num: 128
  rule_name: "HDD"
  type: "replicated"
  device_class: "hdd"
pools:
  - "{{ pool1 }}"
```

d. ツリーを表示します。

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

e. プールを検証します。

```
# for i in $(rados lspools);do echo "pool: $i"; ceph osd pool get $i crush_rule;done

pool: pool1
crush_rule: HDD
```

9. **ベアメタル** または**コンテナー内** のすべてのデプロイメントで、Ansible インベントリーファイル (デフォルトでは **/etc/ansible/hosts** ファイル) を編集するために開きます。例のホストをコメントアウトします。

   a. **[mgrs]** セクションに Ceph Manager (**ceph-mgr**) ノードを追加します。Ceph Manager デーモンを Monitor ノードにコロケーションします。

```
[mgrs]
<monitor-host-name>
<monitor-host-name>
<monitor-host-name>
```

10. Ansible ユーザーとして、Ansible が Ceph ホストに到達できることを確認します。

```
[user@admin ~]$ ansible all -m ping
```

11. 次の行を **/etc/ansible/ansible.cfg** ファイルに追加します。

```
retry_files_save_path = ~/
```

12. **root** として、**/var/log/ansible/** ディレクトリーを作成し、**ansible** ユーザーに適切な権限を割り当てます。

```
[root@admin ~]# mkdir /var/log/ansible
[root@admin ~]# chown ansible:ansible  /var/log/ansible
[root@admin ~]# chmod 755 /var/log/ansible
```

   a. 次のように **log_path** 値を更新して、**/usr/share/ceph-ansible/ansible.cfg** ファイルを編集します。

```
log_path = /var/log/ansible/ansible.log
```

13. Ansible ユーザーとして、**/usr/share/ceph-ansible/** ディレクトリーに移動します。

```
[user@admin ~]$ cd /usr/share/ceph-ansible/
```

14. **ceph-ansible** Playbook を実行します。

```
[user@admin ceph-ansible]$ ansible-playbook site.yml
```

> 注記
>
> デプロイメントの速度を増やすには、**--forks** オプションを **ansible-playbook** に指定します。デフォルトでは、**ceph-ansible** はフォークを **20** に設定します。この設定では、ノードを同時にインストールします。一度に最大 30 個のノードをインストールするには、**ansible-playbook --forks 30 PLAYBOOKFILE** を実行します。管理ノードのリソースが過剰に使用されていないことを確認するために、監視する必要があります。そうである場合は、**--forks** に渡される数を減らします。

15. モニターノードの root アカウントを使用して、Ceph クラスターのステータスを確認します。

```
[root@monitor ~]# ceph health
HEALTH_OK
```

16. **rados** を使用してクラスターが機能していることを確認します。

    a. モニターノードから、8 つの配置グループを持つテストプールを作成します。
    構文

    ```
    [root@monitor ~]# ceph osd pool create <pool-name> <pg-number>
    ```

    例

    ```
    [root@monitor ~]# ceph osd pool create test 8
    ```

    b. **hello-world.txt** というファイルを作成します。
    構文

    ```
    [root@monitor ~]# vim <file-name>
    ```

    例

    ```
    [root@monitor ~]# vim hello-world.txt
    ```

    c. オブジェクト名 **hello-world** を使用して、**hello-world.txt** をテストプールにアップロードします。
    構文

    ```
    [root@monitor ~]# rados --pool <pool-name> put <object-name> <object-file>
    ```

    例

```
[root@monitor ~]# rados --pool test put hello-world hello-world.txt
```

d. テストプールからファイル名 **fetch.txt** として **hello-world** をダウンロードします。
構文

```
[root@monitor ~]# rados --pool <pool-name> get <object-name> <object-file>
```

例

```
[root@monitor ~]# rados --pool test get hello-world fetch.txt
```

e. **fetch.txt** の内容を確認してください。

```
[root@monitor ~]# cat fetch.txt
```

出力は以下のようになります。

```
"Hello World!"
```

注記

クラスタの状態を確認するだけでなく、**ceph-medic** ユーティリティを使用して Ceph Storage Cluster を総合的に診断することができます。『Red Hat Ceph Storage 3 **管理ガイド**』 の「 **ceph-medic を使用した Ceph Storage クラスターの診断** 」の章を参照してください。

3.3. すべての NVME ストレージに OSD ANSIBLE 設定の構成

ストレージに NVMe (Non-volatile Memory Express) デバイスのみを使用する場合のパフォーマンスを最適化するには、各 NVMe デバイスに 4 つの OSD を構成します。通常、OSD はデバイスごとに1つしか設定されないため、NVMe デバイスのスループットを十分に活用できません。

注記

SSD と HDD を混在させる場合、SSD は OSD ではなくジャーナルまたは **block.db** のいずれかに使用されます。

注記

テストでは、各 NVMe デバイスに 4 つの OSD を設定すると、最適なパフォーマンスが得られます。**osds_per_device:4** を設定することをお勧めしますが、必須ではありません。他の値を設定すると、お客様の環境でより良いパフォーマンスが得られる場合があります。

前提条件

- Ceph クラスターのソフトウェアおよびハードウェアの要件をすべて満たすこと。

手順

1. **osd_scenario:lvm** および **osds_per_device:4** を **group_vars/osds.yml** に設定します。
   -

```
osd_scenario: lvm
osds_per_device: 4
```

2. **devices** の下に NVMe デバイスを一覧表示します。

```
devices:
  - /dev/nvme0n1
  - /dev/nvme1n1
  - /dev/nvme2n1
  - /dev/nvme3n1
```

3. **group_vars/osds.yml** の設定は以下のようになります。

```
osd_scenario: lvm
osds_per_device: 4
devices:
  - /dev/nvme0n1
  - /dev/nvme1n1
  - /dev/nvme2n1
  - /dev/nvme3n1
```

注記

lvm_volumes ではなく、この設定で **devices** を使用する必要があります。これは、**lvm_volumes** が、通常、作成済みの論理ボリュームで使用され、**osds_per_device** は Ceph による論理ボリュームの自動作成を意味するためです。

関連情報

- Red Hat Enterprise Linux への Red Hat Ceph ストレージクラスターのインストール
- Ubuntu への Red Hat Ceph Storage クラスターのインストール

3.4. メタデータサーバのインストール

Ansible 自動化アプリケーションを使用して Ceph Metadata Server (MDS) をインストールします。Ceph File System をデプロイするには、メタデータサーバーデーモンが必要です。

前提条件

- 稼働中の Red Hat Ceph Storage クラスター

手順

Ansible 管理ノードで以下の手順を実行します。

1. 新しいセクション **[mdss]** を **/etc/ansible/hosts** ファイルに追加します。

```
[mdss]
hostname
hostname
hostname
```

hostname は、Ceph Metadata Server をインストールするノードのホスト名に置き換えてください。

2. **/usr/share/ceph-ansible** ディレクトリーに移動します。

```
[root@admin ~]# cd /usr/share/ceph-ansible
```

3. 任意。デフォルトの変数を変更します。

   a. **mdss.yml** という名前の **group_vars/mdss.yml.sample** ファイルのコピーを作成します。

```
[root@admin ceph-ansible]# cp group_vars/mdss.yml.sample group_vars/mdss.yml
```

   b. オプションで、**mdss.yml** のパラメーターを編集します。詳細は、**mdss.yml** を参照してください。

4. Ansible のユーザーとして、Ansible のプレイブックを実行します。

```
[user@admin ceph-ansible]$ ansible-playbook site.yml --limit mdss
```

5. メタデータサーバーをインストールしたら、設定を行います。詳細は、Red Hat Ceph Storage3 の Ceph ファイルシステムガイドの**メタデータサーバーデーモンの設定**の章を参照してください。

関連情報

- Red Hat Ceph Storage 3 の**Ceph ファイルシステムガイド**
- **limit オプションについて**

3.5. CEPH クライアントロールのインストール

ceph-ansible ユーティリティーは、Ceph 設定ファイルと管理キーリングをノードにコピーする **ceph-client** ロールを提供します。さらに、このロールを使用してカスタムプールおよびクライアントを作成することができます。

前提条件

- 稼働中の Ceph ストレージクラスター (**active + clean** の状態が望ましい)。
- *2章Red Hat Ceph Storage のインストール要件* に記載されているタスクを実行します。

手順
Ansible 管理ノードで以下のタスクを実行します。

1. 新しいセクション **[clients]** を **/etc/ansible/hosts** ファイルに追加します。

```
[clients]
<client-hostname>
```

   <client-hostname> は、**ceph-client** ロールをインストールするノードのホスト名に置き換えます。

2. **/usr/share/ceph-ansible** ディレクトリーに移動します。

```
[root@admin ~]# cd /usr/share/ceph-ansible
```

3. **clients.yml** という名前の **clients.yml.sample** ファイルの新しいコピーを作成します。

```
[root@admin ceph-ansible ~]# cp group_vars/clients.yml.sample group_vars/clients.yml
```

4. **group_vars/clients.yml** ファイルを開き、以下の行をコメント解除します。

```
keys:
  - { name: client.test, caps: { mon: "allow r", osd: "allow class-read object_prefix
rbd_children, allow rwx pool=test" },  mode: "{{ ceph_keyring_permissions }}" }
```

   a. **client.test** を実際のクライアント名に置き換え、クライアントキーをクライアント定義の行に追加します。以下に例を示します。

```
key: "ADD-KEYRING-HERE=="
```

   これで、行全体の例は次のようになります。

```
- { name: client.test, key: "AQAin8tUMICVFBAALRHNrV0Z4MXupRw4v9JQ6Q==", caps:
{ mon: "allow r", osd: "allow class-read object_prefix rbd_children, allow rwx pool=test" },
mode: "{{ ceph_keyring_permissions }}" }
```

   注記

   ceph-authtool --gen-print-key コマンドは、新しいクライアントキーを生成することができます。

5. 必要に応じて、プールおよびクライアントを作成するように **ceph-client** に指示します。

   a. **clients.yml** を更新します。

      - **user_config** 設定のコメントを解除して、**true** に設定します。
      - **pools** セクションおよび **keys** セクションのコメントを解除し、必要に応じて更新します。**cephx** 機能を使用して、カスタムプールとクライアント名をまとめて定義できます。

   b. **osd_pool_default_pg_num** 設定を **all.yml** ファイルの **ceph_conf_overrides** セクションに追加します。

```
ceph_conf_overrides:
  global:
    osd_pool_default_pg_num: <number>
```

   <number> を配置グループのデフォルト数に置き換えてください。

6. Ansible Playbook の実行:

```
[user@admin ceph-ansible]$ ansible-playbook site.yml --limit clients
```

関連情報

- 「**limit** オプションについて」

3.6. CEPH OBJECT GATEWAY のインストール

Ceph Object Gateway は、RADOS ゲートウェイとも呼ばれ、**librados** API 上に構築されたオブジェクトストレージインターフェースであり、アプリケーションに Ceph ストレージクラスターへの RESTful ゲートウェイを提供します。

前提条件

- 稼働中の Red Hat Ceph Storage クラスター (**active + clean** 状態が望ましい)
- Ceph Object Gateway ノードで、*2章Red Hat Ceph Storage のインストール要件* に記載されているタスクを実行します。

手順
Ansible 管理ノードで以下のタスクを実行します。

1. **[rgws]** セクションの下の **/etc/ansible/hosts** ファイルにゲートウェイホストを追加して、それらのロールを Ansible に識別します。ホストに連続する命名がある場合は、以下のように範囲を使用します。

```
[rgws]
<rgw_host_name_1>
<rgw_host_name_2>
<rgw_host_name[3..10]>
```

2. Ansible 設定ディレクトリーに移動します。

```
[root@ansible ~]# cd /usr/share/ceph-ansible
```

3. サンプルファイルから **rgws.yml** ファイルを作成します。

```
[root@ansible ~]# cp group_vars/rgws.yml.sample group_vars/rgws.yml
```

4. **group_vars/rgws.yml** ファイルを開いて編集します。管理者キーを Ceph Object Gateway ノードにコピーするには、**copy_admin_key** オプションのコメントを解除します。

```
copy_admin_key: true
```

5. **rgws.yml** ファイルでは、デフォルトの **7480** ポートとは異なるデフォルトポートを指定することができます。以下に例を示します。

```
ceph_rgw_civetweb_port: 80
```

6. **all.yml** ファイルは **radosgw_interface** を指定する**必要があります**。以下に例を示します。

```
radosgw_interface: eth0
```

   インターフェースを指定すると、同じホストで複数のインスタンスを実行している場合に、Civetweb が別の Civetweb インスタンスと同じ IP アドレスにバインドされないようにします。

7. 通常、デフォルトの設定を変更するには、**rgw.yml** ファイル内の設定をアンコメントし、それに応じて変更します。**rgw.yml** ファイルにない設定に追加の変更を加えるには、**all.yml** ファイ

ルで **ceph_conf_overrides:** を使用します。例えば、**rgw_dns_name:** に DNS サーバーのホストを設定し、クラスタの DNS サーバーをワイルドカード用に設定して S3 サブドメインを有効にします。

```
ceph_conf_overrides:
  client.rgw.rgw1:
    rgw_dns_name: <host_name>
    rgw_override_bucket_index_max_shards: 16
    rgw_bucket_default_quota_max_objects: 1638400
```

詳細な設定の詳細は、Red Hat Ceph Storage 3 の『実稼働環境への Ceph Object Gateway ガイド』を参照してください。高度なトピックには以下が含まれます。

- Ansible グループの設定
- ストレージストラテジーの開発プールの作成方法および設定方法の詳細は、**「ルートプールの作成」**、**「システムプールの作成」**、および**「データ配置戦略の作成」**セクションを参照してください。
  バケットのシャード化の詳細は、「バケットのシャード化」を参照してください。

8. **group_vars/all.yml** ファイルの **radosgw_interface** パラメーターのコメントを外します。

```
radosgw_interface: <interface>
```

以下を置き換えます。

- Ceph Object Gateway がリッスンするインターフェースを使用する **<interface>**

詳細は、**all.yml** ファイルを参照してください。

9. Ansible Playbook の実行:

```
[user@admin ceph-ansible]$ ansible-playbook site.yml --limit rgws
```

注記

Ansible は、各 Ceph Object Gateway が確実に実行されていることを確認します。

単一サイトの構成の場合は、Ceph ObjectGateway を Ansible 構成に追加します。

マルチサイトデプロイメントでは、各ゾーンの Ansible 設定を行う必要があります。つまり、Ansible によって、そのゾーン用に Ceph Storage クラスターおよびゲートウェイインスタンスが作成されます。

マルチサイトクラスターのインストールが完了したら、マルチサイト用のクラスターの設定方法は、Red Hat Ceph Storage 4 の『**Object Gateway Guide for Red Hat Enterprise Linux**』の「マルチサイト」 の章に進んでください。

関連情報

- 「**limit** オプションについて」
- Red Hat Enterprise Linux のオブジェクトゲートウェイガイド

3.6.1. マルチサイト Ceph Object Gateway の設定

Ansibleは、マルチサイト環境の Ceph Object Gateway のレルム、ゾーングループ、マスターゾーン、セカンダリゾーンを構成します。

前提条件

- Red Hat Ceph Storage クラスターを実行する 2 つ。
- Ceph Object Gateway ノード上で、**『Red Hat Ceph Storage インストールガイド』**の「Red Hat Ceph Storage のインストール要件」に記載のタスクを実行します。
- ストレージクラスターごとに 1 つの Ceph Object Gateway をインストールして構成します。

手順

1. プライマリーストレージクラスターの Ansible ノードで以下の手順を実行します。

   a. システムキーを生成し、**multi-site-keys.txt** ファイルで出力を取得します。

   ```
   [root@ansible ~]# echo system_access_key: $(cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 20 | head -n 1) > multi-site-keys.txt
   [root@ansible ~]# echo system_secret_key: $(cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 40 | head -n 1) >> multi-site-keys.txt
   ```

   b. Ansible 設定ディレクトリー **/usr/share/ceph-ansible** に移動します。

   ```
   [root@ansible ~]# cd /usr/share/ceph-ansible
   ```

   c. **group_vars/all.yml** ファイルを開いて編集します。以下のオプションを追加し、**$ZONE_NAME**、**$ZONE_GROUP_NAME**、**$REALM_NAME**、**$ACCESS_KEY**、**$SECRET_KEY** の値を適宜更新することで、マルチサイトのサポートを有効にします。複数の Ceph Object Gateway がマスターゾーンにある場合は、**rgw_multisite_endpoints** オプションを設定する必要があります。**rgw_multisite_endpoints** オプションの値は、コンマで区切られたリストで、スペースは含みません。

   例

   ```
   rgw_multisite: true
   rgw_zone: $ZONE_NAME
   rgw_zonemaster: true
   rgw_zonesecondary: false
   rgw_multisite_endpoint_addr: "{{ ansible_fqdn }}"
   rgw_multisite_endpoints:
   http://foo.example.com:8080,http://bar.example.com:8080,http://baz.example.com:8080
   rgw_zonegroup: $ZONE_GROUP_NAME
   rgw_zone_user: zone.user
   rgw_realm: $REALM_NAME
   system_access_key: $ACCESS_KEY
   system_secret_key: $SECRET_KEY
   ```

   注記

   ansible_fqdn ドメイン名は、セカンダリーストレージクラスターから解決可能である必要があります。

注記

新しい Object Gateway を追加するときは、Ansible Playbook を実行する前に、新しい Object Gateway のエンドポイント URL を使用して **rgw_multisite_endpoints** リストの最後に追加してください。

d. Ansible Playbook の実行:

```
[user@ansible ceph-ansible]$ ansible-playbook site.yml --limit rgws
```

e. Ceph Object Gateway デーモンを再起動します。

```
[root@rgw ~]# systemctl restart ceph-radosgw@rgw.`hostname -s`
```

2. セカンダリーストレージクラスターの Ansible ノードで以下の手順を行います。

a. Ansible 設定ディレクトリー **/usr/share/ceph-ansible** に移動します。

```
[root@ansible ~]# cd /usr/share/ceph-ansible
```

b. **group_vars/all.yml** ファイルを開いて編集します。以下のオプションを追加し、**$ZONE_NAME**、**$ZONE_GROUP_NAME**、**$REALM_NAME**、**$ACCESS_KEY**、**$SECRET_KEY** の値を更新することで、マルチサイトのサポートを有効にします。**rgw_zone_user**、**system_access_key**、**system_secret_key** は、マスターゾーンの設定で使用したものと同じ値でなければなりません。**rgw_pullhost** オプションには、マスターゾーンの Ceph Object Gateway を指定する必要があります。
複数の Ceph Object Gateway がセカンダリゾーンにある場合は、**rgw_multisite_endpoints** オプションを設定する必要があります。**rgw_multisite_endpoints** オプションの値は、コンマで区切られたリストで、スペースは含みません。

例

```
rgw_multisite: true
rgw_zone: $ZONE_NAME
rgw_zonemaster: false
rgw_zonesecondary: true
rgw_multisite_endpoint_addr: "{{ ansible_fqdn }}"
rgw_multisite_endpoints:
http://foo.example.com:8080,http://bar.example.com:8080,http://baz.example.com:8080
rgw_zonegroup: $ZONE_GROUP_NAME
rgw_zone_user: zone.user
rgw_realm: $REALM_NAME
system_access_key: $ACCESS_KEY
system_secret_key: $SECRET_KEY
rgw_pull_proto: http
rgw_pull_port: 8080
rgw_pullhost: $MASTER_RGW_NODE_NAME
```

注記

ansible_fqdn ドメイン名は、プライマリーストレージクラスターから解決可能である必要があります。

注記

新しい Object Gateway を追加するときは、Ansible Playbook を実行する前に、新しい Object Gateway のエンドポイント URL を使用して **rgw_multisite_endpoints** リストの最後に追加してください。

c. Ansible Playbook の実行:

```
[user@ansible ceph-ansible]$ ansible-playbook site.yml --limit rgws
```

d. Ceph Object Gateway デーモンを再起動します。

```
[root@rgw ~]# systemctl restart ceph-radosgw@rgw.`hostname -s`
```

3. マスターおよびセカンダリストレージクラスターで Ansible プレイブックを実行すると、アクティブ-アクティブ Ceph Object Gateway 構成が実行されます。

4. マルチサイト Ceph Object Gateway の設定を確認します。

   a. 各サイトの Ceph Monitor ノードと Object Gateway ノードから、プライマリーとセカンダリが他のサイトを **curl** できる必要があります。

   b. 両方のサイトで **radosgw-admin sync status** コマンドを実行します。

3.7. NFS-GANESHA ゲートウェイのインストール

Ceph NFS Ganesha ゲートウェイは、Ceph Object Gateway 上に構築される NFS インターフェースで、ファイルシステム内のファイルを Ceph Object Storage に移行するために POSIX ファイルシステムインターフェースを使用するアプリケーションを Ceph Object Gateway に提供します。

前提条件

- 稼働中の Ceph ストレージクラスター (**active + clean** の状態が望ましい)。
- Ceph Object Gateway を実行するノードを少なくとも 1 つ。
- 開始前の手順を 実行します。

手順

Ansible 管理ノードで以下のタスクを実行します。

1. サンプルファイルから **nfss** ファイルを作成します。

```
[root@ansible ~]# cd /usr/share/ceph-ansible/group_vars
[root@ansible ~]# cp nfss.yml.sample nfss.yml
```

2. **[nfss]** グループの下にゲートウェイホストを **/etc/ansible/hosts** ファイルに追加して、Ansible へのグループメンバーシップを特定します。ホストがシーケンシャルに命名されている場合は、範囲を指定します。以下に例を示します。

```
[nfss]
<nfs_host_name_1>
<nfs_host_name_2>
<nfs_host_name[3..10]>
```

3. Ansible の設定ディレクトリである **/etc/ansible/** に移動します。

```
[root@ansible ~]# cd /usr/share/ceph-ansible
```

4. 管理者キーを Ceph Object Gateway ノードにコピーするには、**/usr/share/ceph-ansible/group_vars/nfss.yml** ファイルの **copy_admin_key** 設定をコメント解除します。

```
copy_admin_key: true
```

5. **/usr/share/ceph-ansible/group_vars/nfss.yml** ファイルの FSAL (File System Abstraction Layer) セクションを設定します。ID、S3 ユーザー ID、S3 アクセスキーおよびシークレットを提供します。NFSv4 の場合は、以下のようになります。

```
###################
# FSAL RGW Config #
###################
#ceph_nfs_rgw_export_id: <replace-w-numeric-export-id>
#ceph_nfs_rgw_pseudo_path: "/"
#ceph_nfs_rgw_protocols: "3,4"
#ceph_nfs_rgw_access_type: "RW"
#ceph_nfs_rgw_user: "cephnfs"
# Note: keys are optional and can be generated, but not on containerized, where
# they must be configered.
#ceph_nfs_rgw_access_key: "<replace-w-access-key>"
#ceph_nfs_rgw_secret_key: "<replace-w-secret-key>"
```

> **警告**
>
> アクセスおよびシークレットキーは任意で、生成できます。

6. Ansible Playbook の実行:

```
[user@admin ceph-ansible]$ ansible-playbook site-docker.yml --limit nfss
```

関連情報

- 「**limit** オプションについて」
- Red Hat Enterprise Linux のオブジェクトゲートウェイガイド

3.8. LIMIT オプションについて

本セクションでは、Ansible の **--limit** オプションを説明します。

Ansible は、インベントリーファイルの特定のセクションに **site**、**site-docker**、**rolling_upgrade** Ansible Playbook を使用できるようにする **--limit** オプションをサポートしています。

```
$ ansible-playbook site.yml|rolling_upgrade.yml|site-docker.yml --limit
osds|rgws|clients|mdss|nfss|iscsigws
```

たとえば、ベアメタルに OSD のみを再デプロイするには、Ansible ユーザーとして次のコマンドを実行します。

```
$ ansible-playbook /usr/share/ceph-ansible/site.yml --limit osds
```

重要

1 つのノードに Ceph コンポーネントを同じ場所に配置すると、**limit** オプションで指定されたコンポーネントタイプが 1 つだけであるにもかかわらず、Ansible はノード上のすべてのコンポーネントにプレイブックを適用します。たとえば、OSD とメタデータサーバー (MDS) を含むノードで **--limit osds** オプションを指定して **rolling_update** プレイブックを実行すると、Ansible は OSD と MDS の両方のコンポーネントをアップグレードします。

3.9. 関連情報

- Ansible ドキュメント

第4章 RED HAT CEPH STORAGE CLUSTER のアップグレード

このセクションでは、Red Hat Ceph Storage の新しいメジャーバージョンまたはマイナーバージョンにアップグレードする方法について説明します。

- ストレージクラスターをアップグレードするには、「ストレージクラスターのアップグレード」 を参照してください。
- Red Hat Ceph Storage Dashboard をアップグレードするには、「Red Hat Ceph Storage Dashboard のアップグレード」 を参照してください。

管理ノードの **/usr/share/ceph-ansible/infrastructure-playbooks/** ディレクトリーにある **Ansiblerolling_update.yml** Playbook を使用して、Red Hat Ceph Storage の 2 つのメジャーバージョンまたはマイナーバージョン間でアップグレードするか、非同期更新を適用します。

Ansible は Ceph ノードを以下の順序でアップグレードします。

- ノードの監視
- MGR ノード
- OSD ノード
- MDS ノード
- Ceph Object Gateway ノード
- その他すべての Ceph クライアントノード

注記

Red Hat Ceph Storage 3 では、**/usr/share/ceph-ansible/group_vars/** ディレクトリーにある Ansible 設定ファイルにいくつかの変更が導入されており、特定のパラメーターの名前が変更されたり削除されたりしています。したがって、バージョン 3 にアップグレードした後、**all.yml.sample** ファイルと **osds.yml.sample** ファイルから新しいコピーを作成する前に、**all.yml** ファイルと **osds.yml** ファイルのバックアップコピーを作成してください。変更点の詳細は、付録H *バージョン 2 と 3 の間の Ansible 変数の変更* をご覧ください。

注記

Red Hat Ceph Storage 3.1 以降では、Object Gateway および高速 NVMe ベースの SSD (および SATA SSD) を使用する場合のパフォーマンスのためにストレージを最適化するために、新しい Ansible プレイブックが導入されています。Playbook は、ジャーナルとバケットインデックスを SSD に一緒に配置することでこれを行います。これにより、すべてのジャーナルを 1 つのデバイスに配置する場合に比べてパフォーマンスを向上させることができます。これらの Playbook は、Ceph のインストール時に使用されます。既存の OSD は動作し続け、アップグレード中に追加のステップは必要ありません。このようにストレージを最適化するために OSD を同時に再設定する際に、Ceph クラスターをアップグレードする方法はありません。ジャーナルまたはバケットインデックスに異なるデバイスを使用するには、OSD を再プロビジョニングする必要があります。詳細は、Ceph Object Gateway for Production での LVM での NVMe の最適な使用を参照してください。

重要

Playbook **rolling_update.yml** には、同時に更新するノード数を調整する **シリアル** 変数が含まれます。Red Hat では、デフォルト値 (**1**) を使用することを強く推奨します。これにより、Ansible がクラスターノードを 1 つずつアップグレードします。

重要

いずれかの時点でアップグレードが失敗した場合は、**ceph status** コマンドでクラスターの状態を確認して、アップグレードの失敗理由を把握します。不具合の原因や解決方法がわからない場合は、**Red Hat サポート** にお問い合わせください。

重要

rolling_update.yml Playbook を使用して Red Hat Ceph Storage 3.x バージョンにアップグレードする場合、Ceph ファイルシステム (Ceph FS)を使用するユーザーは、Metadata Server (MDS) クラ スターを手動で更新する必要があります。これは、既知の問題によるものです。

ceph-ansible rolling-upgrade.yml を使用してクラスター全体をアップグレードする前に **/etc/ansible/hosts** の MDS ホストをコメントアウトしてから、MDS を手動でアップグレードします。**etc/ansible/hosts** ファイルでは

```
#[mdss]
#host-abc
```

MDS クラスターの更新方法など、この既知の問題の詳細については、Red Hat Ceph Storage 3.0 リリースノート を参照してください。

重要

Red Hat Ceph Storage クラスターを以前のバージョンからバージョン 3.2 にアップグレードする場合、Ceph Ansible 設定ではデフォルトのオブジェクトストアタイプが BlueStore に設定されます。OSD オブジェクトストアに FileStore を使用する場合は、Ceph Ansible 設定を明示的に FileStore に設定します。これにより、新たにデプロイされ、置き換えられた OSD は FileStore を使用します。

重要

Playbook **rolling_update.yml** を使用して Red Hat Ceph Storage 3.x バージョンにアップグレードし、マルチサイト Ceph Object Gateway 設定を使用している場合には、マルチサイト設定を指定するために **all.yml** ファイルを手動で更新する必要はありません。

前提条件

- ストレージクラスター内のすべてのノードで **root** ユーザーとしてログインします。
- ストレージクラスターのすべてのノードで、**rhel-7-server-extras-rpms** リポジトリーを有効にします。

```
# subscription-manager repos --enable=rhel-7-server-extras-rpms
```

- Ceph ノードが Red Hat コンテンツ配信ネットワーク (CDN) に接続されておらず、ISO イメージを使用して Red Hat Ceph Storage をインストールした場合は、ローカルリポジトリーを最新

バージョンの Red Hat Ceph Storage で更新します。詳しくは 「Red Hat Ceph ストレージリポジトリーの有効化」 をご覧ください。

- Red Hat Ceph Storage 2.x から 3.x にアップグレードする場合は、Ansible 管理ノードと RBD ミラーリングノードで、Red Hat Ceph Storage 3 Tools リポジトリーを有効にします。

```
# subscription-manager repos --enable=rhel-7-server-rhceph-3-tools-els-rpms
```

- Ansible 管理ノードで、Ansible リポジトリーを有効にします。

```
[root@admin ~]# subscription-manager repos --enable=rhel-7-server-ansible-2.6-rpms
```

- Ansible 管理ノードで、**ansible** パッケージおよび **ceph-ansible** パッケージの最新バージョンがインストールされていることを確認します。

```
[root@admin ~]# yum update ansible ceph-ansible
```

- **rolling_update.yml** Playbook で、**health_osd_check_retries** と **health_osd_check_delay** の値をそれぞれ **50** と **30** に変更します。

```
health_osd_check_retries: 50
health_osd_check_delay: 30
```

これらの値を設定すると、OSD ノードごとに Ansible は最大 25 分待機し、30 秒ごとにストレージクラスターの状態をチェックし、アップグレードプロセスを続行する前に待機します。

注記

ストレージクラスターで使用されているストレージ容量に基づいて、**health_osd_check_retries** オプションの値をスケールアップまたはダウンします。たとえば、436 TB 未満の 218 TB (ストレージ容量の 50%) を使用している場合は、**health_osd_check_retries** オプションを **50** に設定します。

- アップグレードするクラスターに **exclusive-lock** 機能を使用する Ceph Block Device イメージが含まれている場合には、全 Ceph Block Device ユーザーにクライアントをブラックリストに登録するパーミッションがあるようにしてください。

```
ceph auth caps client.<ID> mon 'allow r, allow command "osd blacklist"' osd '<existing-OSD-user-capabilities>'
```

4.1. ストレージクラスターのアップグレード

手順

Ansible の管理ノードから以下のコマンドを使用します。

1. **root** ユーザーとして、**/usr/share/ceph-ansible/** ディレクトリーにナビゲートします。

```
[root@admin ~]# cd /usr/share/ceph-ansible/
```

2. Red Hat Ceph Storage バージョン 3.x から最新バージョンにアップグレードする場合は、この手順をスキップします。**group_vars/all.yml** と **group_vars/osds.yml** ファイルをバックアップします。

-

```
[root@admin ceph-ansible]# cp group_vars/all.yml group_vars/all_old.yml
[root@admin ceph-ansible]# cp group_vars/osds.yml group_vars/osds_old.yml
[root@admin ceph-ansible]# cp group_vars/clients.yml group_vars/clients_old.yml
```

3. Red Hat Ceph Storage バージョン 3.x から最新バージョンにアップグレードする場合は、この手順をスキップします。Red Hat Ceph Storage 2.x から 3.x にアップグレードする場合は、**group_vars/all.yml.sample**、**group_vars/osds.yml.sample**、**group_vars/clients.yml.sample** ファイルの新しいコピーを作成して、それぞれ **group_vars/all.yml**、**group_vars/osds.yml**、**group_vars/clients.yml** に名前を変更します。それらを開いて編集します。詳しくは、*付録H バージョン2 と 3 の間の Ansible 変数の変更* と「Red Hat Ceph Storage クラスターのインストール」 をご覧ください。

```
[root@admin ceph-ansible]# cp group_vars/all.yml.sample group_vars/all.yml
[root@admin ceph-ansible]# cp group_vars/osds.yml.sample group_vars/osds.yml
[root@admin ceph-ansible]# cp group_vars/clients.yml.sample group_vars/clients.yml
```

4. Red Hat Ceph Storage バージョン 3.x から最新バージョンにアップグレードする場合は、この手順をスキップします。Red Hat Ceph Storage 2.x から 3.x にアップグレードする場合は、**group_vars/clients.yml** ファイルを開き、以下の行をアンコメントします。

```
keys:
  - { name: client.test, caps: { mon: "allow r", osd: "allow class-read object_prefix
rbd_children, allow rwx pool=test" },  mode: "{{ ceph_keyring_permissions }}" }
```

   a. **client.test** を実際のクライアント名に置き換え、クライアントキーをクライアント定義の行に追加します。以下に例を示します。

```
key: "ADD-KEYRING-HERE=="
```

   これで、行全体の例は次のようになります。

```
- { name: client.test, key: "AQAin8tUMICVFBAALRHNrV0Z4MXupRw4v9JQ6Q==", caps:
{ mon: "allow r", osd: "allow class-read object_prefix rbd_children, allow rwx pool=test" },
mode: "{{ ceph_keyring_permissions }}" }
```

注記

クライアントキーを取得するには、**ceph auth get-or-create** コマンドを実行して、指定されたクライアントのキーを表示します。

5. **group_vars/all.yml** ファイルで、**upgrade_ceph_packages** オプションのコメントを外し、**True** に設定します。

```
upgrade_ceph_packages: True
```

6. **group_vars/all.yml** ファイルで、**ceph_rhcs_version** を **3** に設定します。

```
ceph_rhcs_version: 3
```

注記

ceph_rhcs_version オプションを **3** に設定すると、最新バージョンの Red Hat Ceph Storage 3 がプルされます。

7. **group_vars/all.yml** ファイルで、**ceph_origin** パラメーターを **distro** に設定します。

```
ceph_origin: distro
```

8. **group_vars/all.yml** ファイルに **fetch_directory** パラメーターを追加してください。

```
fetch_directory: <full_directory_path>
```

以下を置き換えます。

- **<full_directory_path>** を、書き込み可能な場所 (Ansible ユーザーのホームディレクトリーなど) に置き換えます。ストレージクラスターの初期インストール時に使用した既存のパスを入力してください。

既存のパスが失われていたり、なくなっていたりする場合は、まず次のことを行ってください。

a. 既存の **group_vars/all.yml** ファイルに以下のオプションを追加します。

```
fsid: <add_the_fsid>
generate_fsid: false
```

b. **take-over-existing-cluster.yml** Ansible playbook を実行します。

```
[user@admin ceph-ansible]$ cp infrastructure-playbooks/take-over-existing-cluster.yml .
[user@admin ceph-ansible]$ ansible-playbook take-over-existing-cluster.yml
```

9. アップグレードするクラスターに Ceph Object Gateway ノードが含まれている場合は、**radosgw_interface** パラメーターを **group_vars/all.yml** ファイルに追加します。

```
radosgw_interface: <interface>
```

以下を置き換えます。

- Ceph Object Gateway がリッスンするインターフェースを使用する **<interface>**

10. Red Hat Ceph Storage 3.2 から、デフォルトの OSD オブジェクトストアは BlueStore です。従来の OSD オブジェクトストアを維持するには、**osd_objectstore** オプションを **group_vars/all.yml** ファイルの **filestore** に明示的に設定する必要があります。

```
osd_objectstore: filestore
```

注記

osd_objectstore オプションを **filestore** に設定し、OSD を置き換えると BlueStore ではなく FileStore が使用されます。

11. **/etc/ansible/hosts** にある Ansible インベントリーファイルで、**[mgrs]** セクションの下に Ceph

Manager (**ceph-mgr**) ノードを追加します。Ceph Manager デーモンを Monitor ノードにコロケーションします。バージョン 3.x から最新のバージョンにアップグレードする場合は、この手順をスキップします。

```
[mgrs]
<monitor-host-name>
<monitor-host-name>
<monitor-host-name>
```

12. **infrastructure-playbooks** ディレクトリーから現在のディレクトリーに、**rolling_update.yml** をコピーします。

```
[root@admin ceph-ansible]# cp infrastructure-playbooks/rolling_update.yml .
```

> 重要
>
> Playbook **rolling_update.yml** では、**limit** の ansible オプションを使用しないでください。

13. **/var/log/ansible/** ディレクトリーを作成し、**ansible** ユーザーに適切な権限を割り当てます。

```
[root@admin ceph-ansible]# mkdir /var/log/ansible
[root@admin ceph-ansible]# chown ansible:ansible  /var/log/ansible
[root@admin ceph-ansible]# chmod 755 /var/log/ansible
```

    a. 次のように **log_path** 値を更新して、**/usr/share/ceph-ansible/ansible.cfg** ファイルを編集します。

```
log_path = /var/log/ansible/ansible.log
```

14. Ansible ユーザーとして、Playbook を実行します。

```
[user@admin ceph-ansible]$ ansible-playbook rolling_update.yml
```

15. RBD ミラーリングデーモンノードに **root** ユーザーとしてログインしているときに、**rbd-mirror** を手動でアップグレードします。

```
# yum upgrade rbd-mirror
```

デーモンを再起動:

```
# systemctl restart  ceph-rbd-mirror@<client-id>
```

16. クラスターの状態に問題がないことを確認します。**root** ユーザーとしてモニターノードにログインし、cephstatus コマンドを実行します。

```
[root@monitor ~]# ceph -s
```

1. OpenStack 環境で動作する場合には、すべての **cephx** ユーザーがプールに RBD プロファイルを使用するように更新します。以下のコマンドは **root** ユーザーとして実行する必要があります。

- Glance ユーザー

```
ceph auth caps client.glance mon 'profile rbd' osd 'profile rbd pool=<glance-pool-name>'
```

例

```
[root@monitor ~]# ceph auth caps client.glance mon 'profile rbd' osd 'profile rbd
pool=images'
```

- Cinder ユーザー

```
ceph auth caps client.cinder mon 'profile rbd' osd 'profile rbd pool=<cinder-volume-pool-
name>, profile rbd pool=<nova-pool-name>, profile rbd-read-only pool=<glance-pool-
name>'
```

例

```
[root@monitor ~]# ceph auth caps client.cinder mon 'profile rbd' osd 'profile rbd
pool=volumes, profile rbd pool=vms, profile rbd-read-only pool=images'
```

- OpenStack の一般ユーザー

```
ceph auth caps client.openstack mon 'profile rbd' osd 'profile rbd-read-only pool=<cinder-
volume-pool-name>, profile rbd pool=<nova-pool-name>, profile rbd-read-only pool=
<glance-pool-name>'
```

例

```
[root@monitor ~]# ceph auth caps client.openstack mon 'profile rbd' osd 'profile rbd-read-
only pool=volumes, profile rbd pool=vms, profile rbd-read-only pool=images'
```

重要

ライブクライアントの移行を実行する前に、これらの CAPS 更新を行います。これにより、クライアントがメモリーで実行している新しいライブラリーを使用でき、古い CAPS 設定がキャッシュから破棄され、新しい RBD プロファイル設定が適用されるようになります。

4.2. RED HAT CEPH STORAGE DASHBOARD のアップグレード

以下の手順では、Red Hat Ceph Storage Dashboard をバージョン 3.1 から 3.2 にアップグレードする手順の概要を説明しています。

アップグレードする前に、Red Hat Ceph Storage がバージョン 3.1 から 3.2 にアップグレードされていることを確認してください。4.1 を参照してください。手順については、「ストレージクラスターのアップグレード」 を参照してください。

警告

アップグレード手順により、ストレージダッシュボードの履歴データが削除されます。

手順

1. **root** ユーザーとして、Ansible 管理ノードから **cephmetrics-ansible** パッケージを更新します。

```
[root@admin ~]# yum update cephmetrics-ansible
```

2. **/usr/share/cephmetrics-ansible** ディレクトリーに移動します。

```
[root@admin ~]# cd /usr/share/cephmetrics-ansible
```

3. 更新された Red Hat Ceph ストレージダッシュボードをインストールします。

```
[root@admin cephmetrics-ansible]# ansible-playbook -v playbook.yml
```

第5章 次のステップ

これは、最新のデータセンターの困難なストレージ要求を満たすために Red Hat Ceph Storage が実行できることの開始点にすぎません。以下は、さまざまなトピックの情報へのリンクになります。

- パフォーマンスのベンチマークとパフォーマンスカウンターへのアクセスは、Red Hat Ceph Storage 3 の『管理ガイド』の「パフォーマンスのベンチマーク」の章を参照してください。
- スナップショットの作成および管理。Red Hat Ceph Storage 3 の『ブロックデバイスガイド』の「スナップショット」の章を参照してください。
- Red Hat Ceph Storage クラスターの拡張については、Red Hat Ceph Storage 3 の『管理ガイド』の「クラスターサイズの管理」の章を参照してください。
- Ceph Block Device のミラーリングは、Red Hat Ceph Storage 3 の『ブロックデバイスガイド』の「ブロックデバイスのミラーリング」の章を参照してください。
- プロセス管理。Red Hat Ceph Storage 3 の『管理ガイド』の「プロセスの管理」の章を参照してください。
- 調整可能なパラメーター。Red Hat Ceph Storage 3 の「設定ガイド」を参照してください。
- OpenStack のバックエンドストレージとして Ceph を使用する場合には、Red Hat OpenStack Platform の 『ストレージガイド』の「バックエンド」セクションを参照してください。

付録A トラブルシューティング

A.1. ANSIBLE は、予想よりも少ないデバイスを検出するため、インストールを停止します

Ansible 自動化アプリケーションはインストールプロセスを停止し、以下のエラーを返します。

```
- name: fix partitions gpt header or labels of the osd disks (autodiscover disks)
  shell: "sgdisk --zap-all --clear --mbrtogpt -- '/dev/{{ item.0.item.key }}' || sgdisk --zap-all --clear --
mbrtogpt -- '/dev/{{ item.0.item.key }}'"
  with_together:
    - "{{ osd_partition_status_results.results }}"
    - "{{ ansible_devices }}"
  changed_when: false
  when:
    - ansible_devices is defined
    - item.0.item.value.removable == "0"
    - item.0.item.value.partitions|count == 0
    - item.0.rc != 0
```

エラー内容:

/usr/share/ceph-ansible/group_vars/osds.yml ファイルで **osd_auto_discovery** パラメーターが **true** に設定されている場合、Ansible は利用可能なすべてのデバイスを自動的に検出して設定します。このプロセス中、Ansible はすべての OSD が同じデバイスを使用することを想定します。デバイスは、Ansible が名前を検出するのと同じ順序で名前を取得します。いずれかの OSD でデバイスのいずれかが失敗すると、Ansible は障害が発生したデバイスの検出に失敗し、インストールプロセス全体を停止します。

状況例:

1. 3 つの OSD ノード (**host1**、**host2**、**host3**) は、**/dev/sdb** ディスク、**/dev/sdc** ディスク、および **dev/sdd** ディスクを使用します。
2. **host2** では、**/dev/sdc** ディスクに障害が発生し、削除されます。
3. 次回の再起動時に、Ansible は削除した **/dev/sdc** ディスクの検出に失敗し、**host2**、**/dev/sdb** および **/dev/sdc** (以前は **/dev/sdd**) には 2 つのディスクのみが使用されることを想定します。
4. Ansible はインストールプロセスを停止し、上記のエラーメッセージを返します。

この問題を修正するには、以下を実行します。

/etc/ansible/hosts ファイルで、障害が発生したディスクを持つ OSD ノードが使用するデバイスを指定します (上記の例の **host2**)。

```
[osds]
host1
host2 devices="[ '/dev/sdb', '/dev/sdc' ]"
host3
```

詳しくは *3章Red Hat Ceph Storage の導入* をご覧ください。

付録B RED HAT CEPH STORAGE の手動インストール

重要

Red Hat は、手動でデプロイしたクラスターのアップグレードをサポートしたり、テストしたりしません。したがって、Red Hat は、Ansible を使用して Red Hat Ceph Storage 3 で新規クラスターをデプロイすることを推奨します。詳しくは 3章*Red Hat Ceph Storage の導入* をご覧ください。

Yum などのコマンドラインユーティリティーを使って、手動でデプロイされたクラスターをインストールすることができます。

すべての Ceph クラスターにはモニターが少なくとも 1 つ、最低でも OSD がクラスターに保存されているオブジェクトのコピーとして必要になります。Red Hat は、実稼働環境に 3 台のモニターを使用し、少なくとも 3 つのオブジェクトストレージデバイス (OSD) を使用することを推奨します。

コマンドラインインターフェースを使用して Ceph Storage クラスターをインストールするには、以下の手順を行います。

- 最初のモニターノードをブートストラップします。
- Ceph Manager デーモンのインストール。
- オブジェクトストレージデバイス (OSD) ノードの追加。

B.1. 前提条件

Red Hat Ceph Storage のネットワークタイムプロトコルの設定
すべての Ceph Monitor および OSD ノードでは、ネットワークタイムプロトコル (NTP) を構成する必要があります。Ceph ノードが NTP ピアであることを確認します。NTP は、クロックドリフトから発生する問題を先取りするのに役立ちます。

注記

Ansible を使用して Red Hat Ceph Storage クラスターをデプロイする場合、Ansible は NTP を自動的にインストール、構成、および有効にします。

前提条件

- 有効なタイムソースへのネットワークアクセス。

手順: RHCS のネットワークタイムプロトコルを構成する
ストレージクラスターのすべての RHCS ノードで、 **root** ユーザーとして以下の手順を実行します。

1. **ntp** パッケージをインストールします。

```
# yum install ntp
```

2. NTP サービスを起動し、再起動後も持続するようにします。

```
# systemctl start ntpd
# systemctl enable ntpd
```

3. NTP が正しくクロックを同期していることを確認してください。

```
$ ntpq -p
```

関連情報

- Red Hat Enterprise Linux7 の システム管理者ガイドのntpd を使用した NTP の設定 の章。

ブートストラップの監視

Monitor のブートストラップおよび Ceph Storage クラスターの拡張には、以下のデータが必要です。

一意識別子

ファイルシステム識別子 (**fsid**) はクラスターの一意の識別子です。**fsid** は、Ceph ストレージクラスターが Ceph ファイルシステムに主に使用する場合に使用されていました。Ceph はネイティブのインターフェース、ブロックデバイス、およびオブジェクトストレージゲートウェイのインターフェースもサポートするようになり、**fsid** は一部の誤検出になります。

クラスター名

Ceph クラスターにはクラスター名があり、これはスペースを含まないシンプルな文字列です。デフォルトのクラスター名は **ceph** ですが、別のクラスター名を指定することもできます。デフォルトのクラスター名を上書きすることは、複数のクラスターを扱う場合に特に有効です。
マルチサイトアーキテクチャーで複数のクラスターを実行する場合、クラスター名 (**us-west**、**us-east** など) は、現在のコマンドラインセッションのクラスターを識別します。

注記

コマンドラインインターフェイスでクラスター名を識別するには、クラスター名を使用して Ceph 構成ファイルを指定します (例: **ceph.conf**、**us-west.conf**、**us-east.conf** など)。

たとえば、以下のようになります。

```
# ceph --cluster us-west.conf ...
```

監視名

クラスター内の各 Monitor インスタンスには一意の名前があります。一般的には、Ceph Monitor 名はノード名です。Red Hat では、ノードごとに Ceph Monitor を 1 つ推奨していますが、Ceph OSD デーモンを Ceph Monitor デーモンと同じ場所に配置しないことを推奨します。短いノード名を取得するには、**hostname -s** コマンドを使用します。

マップの監視

初期モニターのブートストラップでは、モニターマップを生成する必要があります。Monitor マップには以下が必要です。

- ファイルシステム識別子 (**fsid**)
- クラスター名、または **ceph** のデフォルトのクラスター名が使用されます。
- 1 つ以上のホスト名とその IP アドレス

キーリングの監視

モニターは、秘密鍵を使用して相互に通信します。Monitor 秘密鍵でキーリングを生成し、初期 Monitor のブートストラップ時にこれを提供する必要があります。

管理者キーリング

ceph コマンドラインインターフェースユーティリティーを使用するには、**client.admin** ユーザーを作成し、そのキーリングを生成します。また、**client.admin** ユーザーを Monitor キーリングに追加する必要があります。

前述の要件は、Ceph 設定ファイルの作成を意味するものではありません。ただし、Red Hat では、Ceph 設定ファイルを作成し、少なくとも **fsid**、**mon initial members**、および **mon host** の設定で設定することを推奨します。

実行時にすべての Monitor 設定を取得および設定できます。ただし、Ceph 設定ファイルには、デフォルト値を上書きする設定のみが含まれる場合があります。Ceph 設定ファイルに設定を追加すると、デフォルト設定が上書きされます。Ceph 設定ファイルでこれらの設定を維持すると、クラスターを簡単に維持できます。

初期モニターをブートストラップするには、以下の手順を実行します。

1. Red Hat Ceph Storage 3 Monitor リポジトリーを有効にします。

```
[root@monitor ~]# subscription-manager repos --enable=rhel-7-server-rhceph-3-mon-els-rpms
```

2. 初期 Monitor ノードで、**root** で **ceph-mon** パッケージをインストールします。

```
# yum install ceph-mon
```

3. **root** で、**/etc/ceph/** ディレクトリーに Ceph 設定ファイルを作成します。デフォルトでは、Ceph は **ceph.conf**を使用します。ここで、**ceph** はクラスター名を反映します。

   構文

```
# touch /etc/ceph/<cluster_name>.conf
```

   例

```
# touch /etc/ceph/ceph.conf
```

4. **root** でクラスターの一意の識別子を生成し、一意の ID を Ceph 設定ファイルの **[global]** セクションに追加します。

   構文

```
# echo "[global]" > /etc/ceph/<cluster_name>.conf
# echo "fsid = `uuidgen`" >> /etc/ceph/<cluster_name>.conf
```

   例

```
# echo "[global]" > /etc/ceph/ceph.conf
# echo "fsid = `uuidgen`" >> /etc/ceph/ceph.conf
```

5. 現在の Ceph 設定ファイルを表示します。

```
$ cat /etc/ceph/ceph.conf
[global]
fsid = a7f64266-0894-4f1e-a635-d0aeaca0e993
```

6. **root** として、最初の Monitor を Ceph 設定ファイルに追加します。

構文

```
# echo "mon initial members = <monitor_host_name>[,<monitor_host_name>]" >>
/etc/ceph/<cluster_name>.conf
```

例

```
# echo "mon initial members = node1" >> /etc/ceph/ceph.conf
```

7. **root** として、初期 Monitor の IP アドレスを Ceph 設定ファイルに追加します。

構文

```
# echo "mon host = <ip-address>[,<ip-address>]" >> /etc/ceph/<cluster_name>.conf
```

例

```
# echo "mon host = 192.168.0.120" >> /etc/ceph/ceph.conf
```

注記

IPv6 アドレスを使用するには、**ms bind ipv6** オプションを **true** に設定します。詳細は、Red Hat Ceph Storage 3 の『設定ガイド』の「バインド」セクションを参照してください。

8. **root** として、クラスターのキーリングを作成し、Monitor シークレットキーを生成します。

構文

```
# ceph-authtool --create-keyring /tmp/<cluster_name>.mon.keyring --gen-key -n mon. --cap
mon '<capabilites>'
```

例

```
# ceph-authtool --create-keyring /tmp/ceph.mon.keyring --gen-key -n mon. --cap mon 'allow *'
creating /tmp/ceph.mon.keyring
```

9. **root** で管理者キーリングを生成し、**<cluster_name>.client.admin.keyring** ユーザーを生成し、ユーザーをキーリングに追加します。

構文

```
# ceph-authtool --create-keyring /etc/ceph/<cluster_name>.client.admin.keyring --gen-key -n
client.admin --set-uid=0 --cap mon '<capabilites>' --cap osd '<capabilites>' --cap mds
'<capabilites>'
```

–

例

```
# ceph-authtool --create-keyring /etc/ceph/ceph.client.admin.keyring --gen-key -n
client.admin --set-uid=0 --cap mon 'allow *' --cap osd 'allow *' --cap mds 'allow'
creating /etc/ceph/ceph.client.admin.keyring
```

10. **root** として、**<cluster_name> .client.admin.keyring** キーを**<cluster_name>.mon.keyring** に追加します。

構文

```
# ceph-authtool /tmp/<cluster_name>.mon.keyring --import-keyring
/etc/ceph/<cluster_name>.client.admin.keyring
```

例

```
# ceph-authtool /tmp/ceph.mon.keyring --import-keyring /etc/ceph/ceph.client.admin.keyring
importing contents of /etc/ceph/ceph.client.admin.keyring into /tmp/ceph.mon.keyring
```

11. Monitor マップを生成します。初期 Monitor のノード名、IP アドレス、および **fsid** を使用して指定し、**/tmp/monmap** として保存します。

構文

```
$ monmaptool --create --add <monitor_host_name> <ip-address> --fsid <uuid>
/tmp/monmap
```

例

```
$ monmaptool --create --add node1 192.168.0.120 --fsid a7f64266-0894-4f1e-a635-
d0aeaca0e993 /tmp/monmap
monmaptool: monmap file /tmp/monmap
monmaptool: set fsid to a7f64266-0894-4f1e-a635-d0aeaca0e993
monmaptool: writing epoch 0 to /tmp/monmap (1 monitors)
```

12. 初期モニターノードで、**root** としてデフォルトのデータディレクトリーを作成します。

構文

```
# mkdir /var/lib/ceph/mon/<cluster_name>-<monitor_host_name>
```

例

```
# mkdir /var/lib/ceph/mon/ceph-node1
```

13. **root** として、最初の Monitor デーモンに Monitor マップとキーリングを設定します。

構文

```
# ceph-mon [--cluster <cluster_name>] --mkfs -i <monitor_host_name> --monmap
/tmp/monmap --keyring /tmp/<cluster_name>.mon.keyring
```

例

```
# ceph-mon --mkfs -i node1 --monmap /tmp/monmap --keyring /tmp/ceph.mon.keyring
ceph-mon: set fsid to a7f64266-0894-4f1e-a635-d0aeaca0e993
ceph-mon: created monfs at /var/lib/ceph/mon/ceph-node1 for mon.node1
```

14. 現在の Ceph 設定ファイルを表示します。

```
# cat /etc/ceph/ceph.conf
[global]
fsid = a7f64266-0894-4f1e-a635-d0aeaca0e993
mon_initial_members = node1
mon_host = 192.168.0.120
```

さまざまな Ceph 構成設定に関する詳細は、Red Hat Ceph Storage 3 の 『設定ガイド』 を参照してください。Ceph 設定ファイルの例では、最も一般的な構成設定の一部を示しています。

例

```
[global]
fsid = <cluster-id>
mon initial members = <monitor_host_name>[, <monitor_host_name>]
mon host = <ip-address>[, <ip-address>]
public network = <network>[, <network>]
cluster network = <network>[, <network>]
auth cluster required = cephx
auth service required = cephx
auth client required = cephx
osd journal size = <n>
osd pool default size = <n>  # Write an object n times.
osd pool default min size = <n> # Allow writing n copy in a degraded state.
osd pool default pg num = <n>
osd pool default pgp num = <n>
osd crush chooseleaf type = <n>
```

15. **root**として、**done** ファイルを作成します。

構文

```
# touch /var/lib/ceph/mon/<cluster_name>-<monitor_host_name>/done
```

例

```
# touch /var/lib/ceph/mon/ceph-node1/done
```

16. **root** として、新しく作成されたディレクトリーおよびファイルで所有者とグループのアクセス権を更新します。

構文

```
# chown -R <owner>:<group> <path_to_directory>
```

例

```
# chown -R ceph:ceph /var/lib/ceph/mon
# chown -R ceph:ceph /var/log/ceph
# chown -R ceph:ceph /var/run/ceph
# chown ceph:ceph /etc/ceph/ceph.client.admin.keyring
# chown ceph:ceph /etc/ceph/ceph.conf
# chown ceph:ceph /etc/ceph/rbdmap
```

注記

Ceph Monitor ノードが OpenStack Controller ノードと同じ場所にある場合、Glance および Cinder キーリングファイルは、それぞれ **glance** および **cinder** によって所有されている必要があります。以下に例を示します。

```
# ls -l /etc/ceph/
...
-rw-------.  1 glance glance      64 <date> ceph.client.glance.keyring
-rw-------.  1 cinder cinder      64 <date> ceph.client.cinder.keyring
...
```

17. カスタム名を持つストレージクラスターの場合、**root** として次の行を追加します。

構文

```
# echo "CLUSTER=<custom_cluster_name>" >> /etc/sysconfig/ceph
```

例

```
# echo "CLUSTER=test123" >> /etc/sysconfig/ceph
```

18. **root** として、初期モニターノードで **ceph-mon** プロセスを開始して有効にします。

構文

```
# systemctl enable ceph-mon.target
# systemctl enable ceph-mon@<monitor_host_name>
# systemctl start ceph-mon@<monitor_host_name>
```

例

```
# systemctl enable ceph-mon.target
# systemctl enable ceph-mon@node1
# systemctl start ceph-mon@node1
```

19. **root** として、monitor デーモンが実行していることを確認します。

構文

```
# systemctl status ceph-mon@<monitor_host_name>
```

例

```
# systemctl status ceph-mon@node1
● ceph-mon@node1.service - Ceph cluster monitor daemon
   Loaded: loaded (/usr/lib/systemd/system/ceph-mon@.service; enabled; vendor preset:
disabled)
   Active: active (running) since Wed 2018-06-27 11:31:30 PDT; 5min ago
 Main PID: 1017 (ceph-mon)
   CGroup: /system.slice/system-ceph\x2dmon.slice/ceph-mon@node1.service
           └─1017 /usr/bin/ceph-mon -f --cluster ceph --id node1 --setuser ceph --setgroup ceph

Jun 27 11:31:30 node1 systemd[1]: Started Ceph cluster monitor daemon.
Jun 27 11:31:30 node1 systemd[1]: Starting Ceph cluster monitor daemon...
```

Red Hat Ceph Storage Monitor をストレージクラスターに追加するには、Red Hat Ceph Storage 3 の『管理ガイド』の「モニターの追加」セクションを参照してください。

B.2. CEPH MANAGER の手動インストール

通常、Ansible 自動化ユーティリティーは、Red Hat Ceph Storage クラスターをデプロイする際に Ceph Manager デーモン (**ceph-mgr**) をインストールします。ただし、Ansible を使用して Red Hat Ceph Storage を管理しない場合は、Ceph Manager を手動でインストールすることができます。Red Hat は、Ceph Manager デーモンと Ceph Monitor デーモンを同じノードに配置することを推奨します。

前提条件

- 稼働中の Red Hat Ceph Storage クラスター
- **root** または **sudo** アクセス
- **rhel-7-server-rhceph-3-mon-els-rpms** リポジトリーが有効
- ファイアウォールを使用している場合は、パブリックネットワーク上でポート **6800-7300** を開く

手順

ceph-mgr がデプロイされるノードで、**root** ユーザーまたは **sudo** ユーティリティーで以下のコマンドを使用します。

1. **ceph-mgr** パッケージをインストールします。

   ```
   [root@node1 ~]# yum install ceph-mgr
   ```

2. **/var/lib/ceph/mgr/ceph-hostname/** ディレクトリーを作成します。

   ```
   mkdir /var/lib/ceph/mgr/ceph-hostname
   ```

   hostname を、**ceph-mgr** デーモンがデプロイされるノードのホスト名に置き換えます。以下に例を示します。

   ```
   [root@node1 ~]# mkdir /var/lib/ceph/mgr/ceph-node1
   ```

3. 新しく作成されたディレクトリーで、**ceph-mgr** デーモンの認証キーを作成します。

```
[root@node1 ~]# ceph auth get-or-create mgr.`hostname -s` mon 'allow profile mgr' osd
'allow *' mds 'allow *' -o /var/lib/ceph/mgr/ceph-node1/keyring
```

4. **/var/lib/ceph/mgr/** ディレクトリーの所有者とグループを **ceph:ceph** に変更します。

```
[root@node1 ~]# chown -R ceph:ceph /var/lib/ceph/mgr
```

5. **ceph-mgr** ターゲットを有効にします。

```
[root@node1 ~]# systemctl enable ceph-mgr.target
```

6. **ceph-mgr** インスタンスを有効にして開始します。

```
systemctl enable ceph-mgr@hostname
systemctl start ceph-mgr@hostname
```

hostname を、**ceph-mgr** をデプロイするノードのホスト名に置き換えます。以下に例を示します。

```
[root@node1 ~]# systemctl enable ceph-mgr@node1
[root@node1 ~]# systemctl start ceph-mgr@node1
```

7. **ceph-mgr** デーモンが正常に起動していることを確認します。

```
ceph -s
```

出力には、**services:** セクションの下に以下の行と同様の行が含まれます。

```
    mgr: node1(active)
```

8. 追加の **ceph-mgr** デーモンをインストールして、現在のアクティブなデーモンに障害が発生した場合にアクティブになるスタンバイデーモンとして機能します。

関連情報

- 「Red Hat Ceph Storage のインストール要件」

OSD ブート制約

モニターを最初に実行したら、オブジェクトストレージデバイス (OSD) の追加を開始できます。オブジェクトのコピー数を処理するのに十分な OSD があるまで、クラスターは **active + clean** 状態に到達できません。

オブジェクトのデフォルトのコピー数は 3 です。少なくとも 3 つの OSD ノードが必要です。ただし、オブジェクトのコピーを 2 つだけ使用する場合には、OSD ノードを 2 つだけ追加してから、Ceph 設定ファイルの **osd pool default size** および **osd pool default min size** 設定を更新します。

詳細は、Red Hat Ceph Storage 3 の『**設定**』の「OSD 設定参照」セクションを参照してください。

初期モニターのブートストラップ後に、クラスターにはデフォルトの CRUSH マップがあります。ただし、CRUSH マップには Ceph ノードにマッピングされた Ceph OSD デーモンがありません。

OSD をクラスターに追加し、デフォルトの CRUSH マップを更新するには、各 OSD ノードで以下のコマンドを実行します。

1. Red Hat Ceph Storage 3 OSD リポジトリーを有効にします。

```
[root@osd ~]# subscription-manager repos --enable=rhel-7-server-rhceph-3-osd-els-rpms
```

2. **root** で Ceph OSD ノードに **ceph-osd** パッケージをインストールします。

```
# yum install ceph-osd
```

3. Ceph 設定ファイルと管理キーリングファイルを初期 Monitor ノードから OSD ノードにコピーします。

構文

```
# scp <user_name>@<monitor_host_name>:<path_on_remote_system>
<path_to_local_file>
```

例

```
# scp root@node1:/etc/ceph/ceph.conf /etc/ceph
# scp root@node1:/etc/ceph/ceph.client.admin.keyring /etc/ceph
```

4. OSD 用の Universally Unique Identifier (UUID) を生成します。

```
$ uuidgen
b367c360-b364-4b1d-8fc6-09408a9cda7a
```

5. **root** として、OSD インスタンスを作成します。

構文

```
# ceph osd create <uuid> [<osd_id>]
```

例

```
# ceph osd create b367c360-b364-4b1d-8fc6-09408a9cda7a
0
```

注記

このコマンドは、後続のステップに必要な OSD 番号識別子を出力します。

6. **root** として、新規 OSD のデフォルトディレクトリーを作成します。

構文

```
# mkdir /var/lib/ceph/osd/<cluster_name>-<osd_id>
```

例

```
# mkdir /var/lib/ceph/osd/ceph-0
```

7. **root** として、OSD として使用するドライブを準備し、作成したディレクトリーにマウントします。Ceph データおよびジャーナル用にパーティションを作成します。ジャーナルとデータパーティションは同じディスクに配置できます。以下の例では、15 GB のディスクを使用しています。

   構文

   ```
   # parted <path_to_disk> mklabel gpt
   # parted <path_to_disk> mkpart primary 1 10000
   # mkfs -t <fstype> <path_to_partition>
   # mount -o noatime <path_to_partition> /var/lib/ceph/osd/<cluster_name>-<osd_id>
   # echo "<path_to_partition>  /var/lib/ceph/osd/<cluster_name>-<osd_id>   xfs
   defaults,noatime 1 2" >> /etc/fstab
   ```

   例

   ```
   # parted /dev/sdb mklabel gpt
   # parted /dev/sdb mkpart primary 1 10000
   # parted /dev/sdb mkpart primary 10001 15000
   # mkfs -t xfs /dev/sdb1
   # mount -o noatime /dev/sdb1 /var/lib/ceph/osd/ceph-0
   # echo "/dev/sdb1 /var/lib/ceph/osd/ceph-0  xfs defaults,noatime 1 2" >> /etc/fstab
   ```

8. **root** として、OSD データディレクトリーを初期化します。

   構文

   ```
   # ceph-osd -i <osd_id> --mkfs --mkkey --osd-uuid <uuid>
   ```

   例

   ```
   # ceph-osd -i 0 --mkfs --mkkey --osd-uuid b367c360-b364-4b1d-8fc6-09408a9cda7a
   ... auth: error reading file: /var/lib/ceph/osd/ceph-0/keyring: can't open /var/lib/ceph/osd/ceph-
   0/keyring: (2) No such file or directory
   ... created new key in keyring /var/lib/ceph/osd/ceph-0/keyring
   ```

   注記

   --mkkeyオプションを指定して **ceph-osd** を実行する前に、ディレクトリーを空にする必要があります。カスタムのクラスタ名を使用する場合、**ceph-osd** ユーティリティーには **--cluster** オプションが必要です。

9. **root** として、OSD 認証キーを登録します。クラスター名が **ceph** と異なる場合は、代わりにクラスター名を挿入してください。

   構文

   ```
   # ceph auth add osd.<osd_id> osd 'allow *' mon 'allow profile osd' -i
   /var/lib/ceph/osd/<cluster_name>-<osd_id>/keyring
   ```

例

```
# ceph auth add osd.0 osd 'allow *' mon 'allow profile osd' -i /var/lib/ceph/osd/ceph-0/keyring
added key for osd.0
```

10. **root** として、OSD ノードを CRUSH マップに追加します。

構文

```
# ceph [--cluster <cluster_name>] osd crush add-bucket <host_name> host
```

例

```
# ceph osd crush add-bucket node2 host
```

11. **root** で OSD ノードを **default** の CRUSH ツリーに配置します。

構文

```
# ceph [--cluster <cluster_name>] osd crush move <host_name> root=default
```

例

```
# ceph osd crush move node2 root=default
```

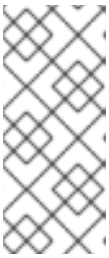
12. **root** として、OSD ディスクを CRUSH マップに追加します。

構文

```
# ceph [--cluster <cluster_name>] osd crush add osd.<osd_id> <weight> [<bucket_type>=
<bucket-name> ...]
```

例

```
# ceph osd crush add osd.0 1.0 host=node2
add item id 0 name 'osd.0' weight 1 at location {host=node2} to crush map
```



注記

CRUSH マップを逆コンパイルし、OSD をデバイス一覧に追加することもできます。OSD ノードをバケットとして追加してから、デバイスを OSD ノードの項目として追加し、OSD に重みを割り当て、CRUSH マップを再コンパイルし、CRUSH マップを設定します。詳細は、Red Hat Ceph Storage 3 の**『ストレージ戦略ガイド』**のセクション「CRUSH マップの編集」を参照してください。

13. **root** として、新しく作成されたディレクトリーおよびファイルで所有者とグループのアクセス権を更新します。

構文

```
# chown -R <owner>:<group> <path_to_directory>
```

-

例

```
# chown -R ceph:ceph /var/lib/ceph/osd
# chown -R ceph:ceph /var/log/ceph
# chown -R ceph:ceph /var/run/ceph
# chown -R ceph:ceph /etc/ceph
```

14. カスタム名を持つストレージクラスターの場合は、**root** として、次の行を **/etc/sysconfig/ceph** ファイルに追加します。

    構文

    ```
    # echo "CLUSTER=<custom_cluster_name>" >> /etc/sysconfig/ceph
    ```

    例

    ```
    # echo "CLUSTER=test123" >> /etc/sysconfig/ceph
    ```

15. OSD ノードは Ceph Storage クラスターの設定にあります。ただし、OSD デーモンは **down** および **in** です。新しい OSD は、データの受信開始前に **up** である必要があります。**root** として、OSD プロセスを有効にして開始します。

    構文

    ```
    # systemctl enable ceph-osd.target
    # systemctl enable ceph-osd@<osd_id>
    # systemctl start ceph-osd@<osd_id>
    ```

    例

    ```
    # systemctl enable ceph-osd.target
    # systemctl enable ceph-osd@0
    # systemctl start ceph-osd@0
    ```

    OSD デーモンを起動すると、これが **up** および **in** になります。

モニターと一部の OSD が稼働しています。以下のコマンドを実行して、配置グループピアを監視できます。

```
$ ceph -w
```

OSD ツリーを表示するには、以下のコマンドを実行します。

```
$ ceph osd tree
```

例

```
ID  WEIGHT    TYPE NAME        UP/DOWN  REWEIGHT  PRIMARY-AFFINITY
-1       2    root default
-2       2        host node2
```

```
 0     1        osd.0      up     1          1
-3     1     host node3
 1     1        osd.1      up     1          1
```

OSD をストレージクラスターに追加してストレージ容量を拡張するには、Red Hat Ceph Storage 3 **『管理ガイド』**の「OSD の追加」セクションを参照してください。

付録C CEPH コマンドラインインターフェースのインストール

Ceph コマンドラインインターフェース (CLI) により、管理者は Ceph 管理コマンドを実行できます。CLI は **ceph-common** パッケージにより提供され、以下のユーティリティーが含まれます。

- **ceph**
- **ceph-authtool**
- **ceph-dencoder**
- **rados**

前提条件

- 稼働中の Ceph ストレージクラスター (**active + clean** の状態が望ましい)。

手順

1. クライアントノードで、Red Hat Ceph Storage 3 Tools リポジトリーを有効にします。

```
[root@gateway ~]# subscription-manager repos --enable=rhel-7-server-rhceph-3-tools-els-rpms
```

2. クライアントノードで、**ceph-common** パッケージをインストールします。

```
# yum install ceph-common
```

3. 最初の監視ノードから、Ceph 設定ファイル (ここでは **ceph.conf**) と管理キーリングをクライアントノードにコピーします。

   構文

```
# scp /etc/ceph/<cluster_name>.conf <user_name>@<client_host_name>:/etc/ceph/
# scp /etc/ceph/<cluster_name>.client.admin.keyring
<user_name>@<client_host_name:/etc/ceph/
```

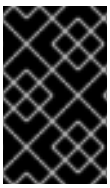
   例

```
# scp /etc/ceph/ceph.conf root@node1:/etc/ceph/
# scp /etc/ceph/ceph.client.admin.keyring root@node1:/etc/ceph/
```

   <client_host_name> を、クライアントノードのホスト名に置き換えてください。

付録D CEPH ブロックデバイスの手動インストール

以下の手順では、シンプロビジョニングされ、サイズが変更可能な Ceph ブロックデバイスをインストールおよびマウントする方法を説明します。



重要

Ceph ブロックデバイスは、Ceph Monitor ノードと OSD ノードとは別のノードにデプロイする必要があります。同じノードでカーネルクライアントとカーネルサーバーデーモンを実行すると、カーネルのデッドロックが発生する可能性があります。

前提条件

- *付録C Ceph コマンドラインインターフェースのインストール* セクションに記載されているタスクを実施するしておく。
- QEMU を使用する仮想マシンのバックエンドとして Ceph ブロックデバイスを使用する場合は、デフォルトのファイル記述子を増やします。詳細は、ナレッジベースの記事「Ceph - VM hangs when transferring large amounts of data to RBD disk」を参照してください。

手順

1. OSD ノード (**osd 'allow rwx'**) 上のファイルへの完全なパーミッションを持つ **client.rbd** という名前の Ceph Block Device ユーザーを作成し、結果をキーリングファイルに出力します。

```
ceph auth get-or-create client.rbd mon 'profile rbd' osd 'profile rbd pool=<pool_name>' \
-o /etc/ceph/rbd.keyring
```

<pool_name> を、**client.rbd** によるアクセスを許可するプールの名前 (例: **rbd**) に置き換えます。

```
# ceph auth get-or-create \
client.rbd mon 'allow r' osd 'allow rwx pool=rbd' \
-o /etc/ceph/rbd.keyring
```

ユーザーの作成に関する詳細は、Red Hat Ceph Storage 3 **『管理ガイド』**の**「ユーザー管理」**セクションを参照してください。

2. ブロックデバイスイメージを作成します。

```
rbd create <image_name> --size <image_size> --pool <pool_name> \
--name client.rbd --keyring /etc/ceph/rbd.keyring
```

<image_name>、**<image_size>**、および **<pool_name>** を指定します。以下に例を示します。

```
$ rbd create image1 --size 4096 --pool rbd \
--name client.rbd --keyring /etc/ceph/rbd.keyring
```

警告

デフォルトの Ceph 設定には、以下の Ceph ブロックデバイス機能が含まれます。

- **layering**
- **exclusive-lock**
- **object-map**
- **deep-flatten**
- **fast-diff**

カーネル RBD (**krbd**) クライアントを使用する場合、Red Hat Enterprise Linux 7.3 に含まれている現在のカーネルバージョンは**object-map**、**deep-flatten**、および **fast-diff** をサポートしていないため、ブロックデバイスイメージをマップすることはできません。

この問題を回避するには、サポートされていない機能を無効にします。これを行うには、以下のいずれかのオプションを使用します。

- サポートされていない機能を動的に無効にします。

  ```
  rbd feature disable <image_name> <feature_name>
  ```

  以下に例を示します。

  ```
  # rbd feature disable image1 object-map deep-flatten fast-diff
  ```

- **rbd create** コマンドで **--image-feature layering** オプションを使用して、新たに作成されたブロックデバイスイメージで **階層化** のみを有効にします。
- Ceph 設定ファイルで機能のデフォルトを無効にします。

  ```
  rbd_default_features = 1
  ```

これは既知の問題です。詳細は、Red Hat Ceph Storage 3 の『**リリースノート**』の「**既知の問題**」の章を参照してください。

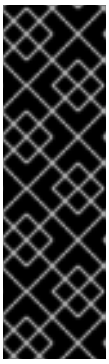
これらの機能はすべて、ユーザー空間の RBD クライアントを使用してブロックデバイスイメージにアクセスするユーザーに機能します。

3. 新規に作成されたイメージをブロックデバイスにマッピングします。

```
rbd map <image_name> --pool <pool_name>\
--name client.rbd --keyring /etc/ceph/rbd.keyring
```

以下に例を示します。

```
# rbd map image1 --pool rbd --name client.rbd \
--keyring /etc/ceph/rbd.keyring
```



重要

カーネルブロックデバイスは現在、CRUSH マップのレガシーストローバケットアルゴリズムのみをサポートしています。CRUSH チューナブルを最適に設定した場合は、それらをレガシーまたは以前のメジャーリリースに設定する必要があります。そうしなければ、イメージをマップできません。

または、CRUSH マップで **straw2** を **straw** に置き換えます。詳細は、Red Hat Ceph Storage 3 の**ストレージ戦略** ガイドの クラッシュマップの編集 の章を参照してください。

4. ファイルシステムを作成してブロックデバイスを使用します。

```
mkfs.ext4 -m5 /dev/rbd/<pool_name>/<image_name>
```

以下のように、プール名とイメージ名を指定します。

```
# mkfs.ext4 -m5 /dev/rbd/rbd/image1
```

この作業には少し時間がかかります。

5. 新しく作成されたファイルシステムをマウントします。

```
mkdir <mount_directory>
mount /dev/rbd/<pool_name>/<image_name> <mount_directory>
```

以下に例を示します。

```
# mkdir /mnt/ceph-block-device
# mount /dev/rbd/rbd/image1 /mnt/ceph-block-device
```

詳細は、Red Hat Ceph Storage 3 のブロックデバイスガイドを参照してください。

付録E CEPH OBJECT GATEWAY の手動インストール

Ceph オブジェクトゲートウェイは RADOS ゲートウェイとしても知られている **librados** API 上に構築されたオブジェクトストレージインターフェースで、RESTful ゲートウェイを Ceph ストレージクラスターに提供します。

前提条件

- 稼働中の Ceph ストレージクラスター (**active + clean** の状態が望ましい)。
- *2章Red Hat Ceph Storage のインストール要件* に記載されているタスクを実行します。

手順

1. Red Hat Ceph Storage 3 Tools リポジトリーを有効にします。

```
[root@gateway ~]# subscription-manager repos --enable=rhel-7-server-rhceph-3-tools-els-rpms
```

2. Object Gateway ノードで、**ceph-radosgw** パッケージをインストールします。

```
# yum install ceph-radosgw
```

3. 初期モニターノードで、以下の手順を実施します。

   a. 以下のように Ceph 設定ファイルを更新します。

```
[client.rgw.<obj_gw_hostname>]
host = <obj_gw_hostname>
rgw frontends = "civetweb port=80"
rgw dns name = <obj_gw_hostname>.example.com
```

   ここで、**<obj_gw_hostname>** はゲートウェイノードの短縮ホスト名です。短縮ホスト名を表示するには、**hostname -s** コマンドを使用します。

   b. 更新された設定ファイルを新しい Object Gateway ノードおよび Ceph Storage クラスターのその他のノードにコピーします。

   構文

```
# scp /etc/ceph/<cluster_name>.conf <user_name>@<target_host_name>:/etc/ceph
```

   例

```
# scp /etc/ceph/ceph.conf root@node1:/etc/ceph/
```

   c. **<cluster_name>.client.admin.keyring** ファイルを新しい Object Gateway ノードにコピーします。

   構文

```
# scp /etc/ceph/<cluster_name>.client.admin.keyring
<user_name>@<target_host_name>:/etc/ceph/
```

例

```
# scp /etc/ceph/ceph.client.admin.keyring root@node1:/etc/ceph/
```

4. Object Gateway ノードで、データディレクトリーを作成します。

構文

```
# mkdir -p /var/lib/ceph/radosgw/<cluster_name>-rgw.`hostname -s`
```

例

```
# mkdir -p /var/lib/ceph/radosgw/ceph-rgw.`hostname -s`
```

5. Object Gateway ノードで、ユーザーとキーリングを追加して、オブジェクトゲートウェイをブートストラップします。

構文

```
# ceph auth get-or-create client.rgw.`hostname -s` osd 'allow rwx' mon 'allow rw' -o
/var/lib/ceph/radosgw/<cluster_name>-rgw.`hostname -s`/keyring
```

例

```
# ceph auth get-or-create client.rgw.`hostname -s` osd 'allow rwx' mon 'allow rw' -o
/var/lib/ceph/radosgw/ceph-rgw.`hostname -s`/keyring
```

重要

ゲートウェイキーの機能を提供する場合は、読み取り機能を指定する必要があります。ただし、Monitor 書き込み機能を提供することはオプションです。指定した場合、Ceph Object Gateway はプールを自動的に作成できます。

このような場合は、プール内の配置グループの数に適切な数を指定してください。それ以外の場合、ゲートウェイはデフォルトの番号を使用しますが、これはニーズに適していない可能性があります。詳細は、「Ceph Placement Groups (PGs) per Pool Calculator」を参照してください。

6. Object Gateway ノードで、**done** ファイルを作成します。

構文

```
# touch /var/lib/ceph/radosgw/<cluster_name>-rgw.`hostname -s`/done
```

例

```
# touch /var/lib/ceph/radosgw/ceph-rgw.`hostname -s`/done
```

7. Object Gateway ノードで、所有者およびグループのパーミッションを変更します。

```
# chown -R ceph:ceph /var/lib/ceph/radosgw
```

```
# chown -R ceph:ceph /var/log/ceph
# chown -R ceph:ceph /var/run/ceph
# chown -R ceph:ceph /etc/ceph
```

8. カスタム名を持つストレージクラスターの場合、**root** として次の行を追加します。

   構文

   ```
   # echo "CLUSTER=<custom_cluster_name>" >> /etc/sysconfig/ceph
   ```

   例

   ```
   # echo "CLUSTER=test123" >> /etc/sysconfig/ceph
   ```

9. Object Gateway ノードで、TCP ポート 80 を開きます。

   ```
   # firewall-cmd --zone=public --add-port=80/tcp
   # firewall-cmd --zone=public --add-port=80/tcp --permanent
   ```

10. Object Gateway ノードで、**ceph-radosgw** プロセスを開始して有効にします。

    構文

    ```
    # systemctl enable ceph-radosgw.target
    # systemctl enable ceph-radosgw@rgw.<rgw_hostname>
    # systemctl start ceph-radosgw@rgw.<rgw_hostname>
    ```

    例

    ```
    # systemctl enable ceph-radosgw.target
    # systemctl enable ceph-radosgw@rgw.node1
    # systemctl start ceph-radosgw@rgw.node1
    ```

インストールが完了すると、書き込み機能が Monitor に設定されると、Ceph Object Gateway はプールを自動的に作成します。プールを手動で作成する方法は、ストレージ戦略ガイドのプールの章を参照してください。

詳細

- Red Hat Ceph Storage 3 Red Hat Enterprise Linux のオブジェクトゲートウェイガイド

付録F CEPH のデフォルト設定の上書き

Ansible 設定ファイルに特に指定しない限り、Ceph はデフォルト設定を使用します。

Ansible は Ceph 設定ファイルを管理するため、**/usr/share/ceph-ansible/group_vars/all.yml** ファイルを編集して Ceph の設定を変更します。**ceph_conf_overrides** の設定を使用して、デフォルトの Ceph 設定を上書きします。

Ansible は、Ceph 設定ファイル (**[global]**、**[mon]**、**[osd]**、**[mds]**、**[rgw]** など) と同じセクションをサポートします。特定の Ceph Object Gateway インスタンスなどの特定のインスタンスをオーバーライドすることもできます。以下に例を示します。

```
###################
# CONFIG OVERRIDE #
###################

ceph_conf_overrides:
  client.rgw.rgw1:
    log_file: /var/log/ceph/ceph-rgw-rgw1.log
```

注記

Ansible には、Ceph 設定ファイルの特定セクションを参照する際に中かっこが含まれません。セクション名および設定名はコロンで終了します。

重要

CONFIG OVERRIDE セクションの **cluster_network** パラメーターを使用してクラスターネットワークを設定しないでください。競合する 2 つのクラスターネットワークが Ceph 設定ファイルに設定されている可能性があるためです。

クラスターネットワークを設定するには、**CEPH CONFIGURATION** セクションで **cluster_network** パラメーターを使用します。詳細は、「Red Hat Ceph Storage クラスターのインストール」 を参照してください。

付録G RED HAT CEPH STORAGE 2 から 3 への手動アップグレード

Ceph Storage Cluster は、クラスターの実行中にローリング方式でバージョン 2 からバージョン 3 にアップグレードできます。クラスター内の各ノードを順番にアップグレードします。前のノードが完了した後でのみ、次のノードに進みます。

Red Hat では、以下の順序で Ceph コンポーネントをアップグレードすることをお勧めします。

- ノードの監視
- OSD ノード
- Ceph Object Gateway ノード
- その他すべての Ceph クライアントノード

Red Hat Ceph Storage 3 では、新しいデーモン Ceph Manager (**ceph-mgr**) が導入されています。モニターノードをアップグレードした後、**ceph-mgr** をインストールします。

Red Hat Ceph Storage 2 を 3 にアップグレードするには、2 つの方法があります。

- Red Hat のコンテンツ配信ネットワーク (CDN) の使用
- Red Hat が提供する ISO イメージファイルを使用する

ストレージクラスターをアップグレードした後、レガシーチューナブルを使用して CRUSH マップに関するヘルス警告を出すことができます。詳細は、Red Hat Ceph Storage3 のストレージ戦略ガイドの CRUSHTunables セクションを参照してください。

例

```
$ ceph -s
    cluster 848135d7-cdb9-4084-8df2-fb5e41ae60bd
     health HEALTH_WARN
            crush map has legacy tunables (require bobtail, min is firefly)
     monmap e1: 1 mons at {ceph1=192.168.0.121:6789/0}
            election epoch 2, quorum 0 ceph1
     osdmap e83: 2 osds: 2 up, 2 in
      pgmap v1864: 64 pgs, 1 pools, 38192 kB data, 17 objects
            10376 MB used, 10083 MB / 20460 MB avail
                 64 active+clean
```

重要

Red Hat は、すべての Ceph クライアントが Ceph ストレージクラスターと同じバージョンを実行することを推奨しています。

前提条件

- アップグレードするクラスターに **exclusive-lock** 機能を使用する Ceph Block Device イメージが含まれている場合には、全 Ceph Block Device ユーザーにクライアントをブラックリストに登録するパーミッションがあるようにしてください。

```
ceph auth caps client.<ID> mon 'allow r, allow command "osd blacklist"' osd '<existing-OSD-user-capabilities>'
```

モニタノードのアップグレード

この項では、Ceph Monitorノードを後のバージョンにアップグレードする手順を説明します。モニターの数は奇数でなければなりません。1 つの Monitor をアップグレードしている間、ストレージクラスターにはクォーラムがあります。

手順

ストレージクラスタの各 Monitor ノードで以下の手順を行います。一度に 1 つの Monitor ノードのみをアップグレードします。

1. ソフトウェアリポジトリーを使用して Red Hat Ceph Storage 2 をインストールした場合は、リポジトリーを無効にします。

```
# subscription-manager repos --disable=rhel-7-server-rhceph-2-mon-rpms --disable=rhel-7-server-rhceph-2-installer-rpms
```

2. Red Hat Ceph Storage 3 Monitor リポジトリーを有効にします。

```
[root@monitor ~]# subscription-manager repos --enable=rhel-7-server-rhceph-3-mon-els-rpms
```

3. **root** で Monitor プロセスを停止します。

   構文

```
# service ceph stop <daemon_type>.<monitor_host_name>
```

   例

```
# service ceph stop mon.node1
```

4. **root** で、**ceph-mon** パッケージをアップデートします。

```
# yum update ceph-mon
```

5. **root** として、所有者とグループの権限を更新します。

   構文

```
# chown -R <owner>:<group> <path_to_directory>
```

   例

```
# chown -R ceph:ceph /var/lib/ceph/mon
# chown -R ceph:ceph /var/log/ceph
# chown -R ceph:ceph /var/run/ceph
# chown ceph:ceph /etc/ceph/ceph.client.admin.keyring
# chown ceph:ceph /etc/ceph/ceph.conf
# chown ceph:ceph /etc/ceph/rbdmap
```

注記

Ceph Monitor ノードが OpenStack Controller ノードと同じ場所にある場合、Glance および Cinder キーリングファイルは、それぞれ **glance** および **cinder** によって所有されている必要があります。以下に例を示します。

```
# ls -l /etc/ceph/
...
-rw-------.  1 glance glance      64 <date> ceph.client.glance.keyring
-rw-------.  1 cinder cinder      64 <date> ceph.client.cinder.keyring
...
```

6. SELinux が強制モードまたは許容モードの場合は、次回の再起動時に SELinux コンテキストのラベルを変更します。

```
# touch /.autorelabel
```

警告

SELinux はすべてのファイルシステムをトラバースし、誤ってラベル付けされたファイルを修正する必要があるため、再ラベル付けは完了するまでに長い時間がかかる場合があります。ディレクトリーの再ラベル付けを除外するには、再起動する前にディレクトリーを **/etc/selinux/fixfiles_exclude_dirs** ファイルに追加します。

7. **root** として、**ceph-mon** プロセスを有効にします。

```
# systemctl enable ceph-mon.target
# systemctl enable ceph-mon@<monitor_host_name>
```

8. **root** で Monitor ノードを再起動します。

```
# shutdown -r now
```

9. Monitor ノードが起動したら、次の Monitor ノードに移動する前に、Ceph ストレージクラスターの状態を確認します。

```
# ceph -s
```

G.1. CEPH MANAGER の手動インストール

通常、Ansible 自動化ユーティリティーは、Red Hat Ceph Storage クラスターをデプロイする際に Ceph Manager デーモン (**ceph-mgr**) をインストールします。ただし、Ansible を使用して Red Hat Ceph Storage を管理しない場合は、Ceph Manager を手動でインストールすることができます。Red Hat は、Ceph Manager デーモンと Ceph Monitor デーモンを同じノードに配置することを推奨します。

前提条件

前提条件

- 稼働中の Red Hat Ceph Storage クラスター
- **root** または **sudo** アクセス
- **rhel-7-server-rhceph-3-mon-els-rpms** リポジトリーが有効
- ファイアウォールを使用している場合は、パブリックネットワーク上でポート **6800-7300** を開く

手順

ceph-mgr がデプロイされるノードで、**root** ユーザーまたは **sudo** ユーティリティーで以下のコマンドを使用します。

1. **ceph-mgr** パッケージをインストールします。

```
[root@node1 ~]# yum install ceph-mgr
```

2. **/var/lib/ceph/mgr/ceph-hostname/** ディレクトリーを作成します。

```
mkdir /var/lib/ceph/mgr/ceph-hostname
```

hostname を、**ceph-mgr** デーモンがデプロイされるノードのホスト名に置き換えます。以下に例を示します。

```
[root@node1 ~]# mkdir /var/lib/ceph/mgr/ceph-node1
```

3. 新しく作成されたディレクトリーで、**ceph-mgr** デーモンの認証キーを作成します。

```
[root@node1 ~]# ceph auth get-or-create mgr.`hostname -s` mon 'allow profile mgr' osd
'allow *' mds 'allow *' -o /var/lib/ceph/mgr/ceph-node1/keyring
```

4. **/var/lib/ceph/mgr/** ディレクトリーの所有者とグループを **ceph:ceph** に変更します。

```
[root@node1 ~]# chown -R ceph:ceph /var/lib/ceph/mgr
```

5. **ceph-mgr** ターゲットを有効にします。

```
[root@node1 ~]# systemctl enable ceph-mgr.target
```

6. **ceph-mgr** インスタンスを有効にして開始します。

```
systemctl enable ceph-mgr@hostname
systemctl start ceph-mgr@hostname
```

hostname を、**ceph-mgr** をデプロイするノードのホスト名に置き換えます。以下に例を示します。

```
[root@node1 ~]# systemctl enable ceph-mgr@node1
[root@node1 ~]# systemctl start ceph-mgr@node1
```

7. **ceph-mgr** デーモンが正常に起動していることを確認します。

```
ceph -s
```

出力には、**services:** セクションの下に以下の行と同様の行が含まれます。

```
    mgr: node1(active)
```

8. 追加の **ceph-mgr** デーモンをインストールして、現在のアクティブなデーモンに障害が発生した場合にアクティブになるスタンバイデーモンとして機能します。

関連情報

- 「Red Hat Ceph Storage のインストール要件」

OSD ノードのアップグレード

この項では、Ceph OSD ノードを後のバージョンにアップグレードする手順を説明します。

前提条件

OSD ノードをアップグレードすると、OSD が停止または再起動する可能性があるため、一部の配置グループがデグレードされます。Ceph がリカバリープロセスを開始しないようにするには、Monitor ノードで、**noout** および **norebalance** OSD フラグを設定します。

```
[root@monitor ~]# ceph osd set noout
[root@monitor ~]# ceph osd set norebalance
```

手順

ストレージクラスタの各 OSD ノードで以下の手順を行います。一度に 1 つの OSD ノードのみをアップグレードします。Red Hat Ceph Storage 2.3 に対して ISO ベースのインストールが実行された場合は、この最初のステップをスキップしてください。

1. **root** で、Red Hat Ceph Storage 2 のリポジトリーを無効にします。

```
# subscription-manager repos --disable=rhel-7-server-rhceph-2-osd-rpms --disable=rhel-7-server-rhceph-2-installer-rpms
```

2. Red Hat Ceph Storage 3 OSD リポジトリーを有効にします。

```
[root@osd ~]# subscription-manager repos --enable=rhel-7-server-rhceph-3-osd-els-rpms
```

3. **root** として、実行中の OSD プロセスをすべて停止します。

   構文

```
# service ceph stop <daemon_type>.<osd_id>
```

   例

```
# service ceph stop osd.0
```

4. **root** として、**ceph-osd** パッケージを更新します。

```
# yum update ceph-osd
```

5. **root** として、新しく作成されたディレクトリーおよびファイルで所有者とグループのアクセス権を更新します。

   構文

   ```
   # chown -R <owner>:<group> <path_to_directory>
   ```

   例

   ```
   # chown -R ceph:ceph /var/lib/ceph/osd
   # chown -R ceph:ceph /var/log/ceph
   # chown -R ceph:ceph /var/run/ceph
   # chown -R ceph:ceph /etc/ceph
   ```

   注記

   次の **find** コマンドを使用すると、多数のディスクがある Ceph ストレージクラスターで **chown** コマンドを並行して使用することにより、所有権を変更するプロセスが速くなる可能性があります。

   ```
   # find /var/lib/ceph/osd -maxdepth 1 -mindepth 1 -print | xargs -P12 -n1 chown
   -R ceph:ceph
   ```

6. SELinux が強制モードまたは許可モードに設定されている場合は、次の再起動のためにファイルに SELinux コンテキストの再ラベル付けを設定します。

   ```
   # touch /.autorelabel
   ```

   警告

   SELinux はすべてのファイルシステムをトラバースし、誤ってラベル付けされたファイルを修正する必要があるため、再ラベル付けは完了するまでに長い時間がかかります。ディレクトリーの再ラベル付けを除外するには、再起動する前にディレクトリーを **/etc/selinux/fixfiles_exclude_dirs** ファイルに追加します。

注記

配置グループ (PG) あたりのオブジェクト数が多い環境では、ディレクトリーの列挙速度が低下し、パフォーマンスに悪影響を及ぼします。これは、SELinux コンテキストを検証する xattr クエリが追加されたことが原因です。マウント時にコンテキストを設定すると、コンテキストに対する xattr クエリが削除され、特に低速のディスクでの全体的なディスクパフォーマンスが向上します。

/etc/ceph/ceph.conf ファイルの **[osd]** セクションに次の行を追加します。

+

```
osd_mount_options_xfs=rw,noatime,inode64,context="system_u:object_r:ceph_var_lib_t:s0"
```

7. **root** として、カーネルからデバイスイベントを再生します。

```
# udevadm trigger
```

8. **root** として、**ceph-osd** プロセスを有効にします。

```
# systemctl enable ceph-osd.target
# systemctl enable ceph-osd@<osd_id>
```

9. **root** として、OSD ノードを再起動します。

```
# shutdown -r now
```

10. 次の OSD ノードに移動します。

注記

noout フラグと **norebalance** フラグが設定されている場合、ストレージクラスターは **HEALTH_WARN** 状態になります。

```
$ ceph health
HEALTH_WARN noout,norebalance flag(s) set
```

Ceph Storage Cluster のアップグレードが完了したら、以前に設定した OSD フラグの設定を解除し、ストレージクラスターのステータスを確認します。

モニターノードで、すべての OSD ノードがアップグレードされた後、**noout** フラグと**norebalance** フラグの設定を解除します。

```
# ceph osd unset noout
# ceph osd unset norebalance
```

さらに、**ceph osd require-osd-release <release>** コマンドを実行します。このコマンドにより、Red Hat Ceph Storage 2.3 を搭載した OSD をストレージクラスターに追加できなくなります。このコマンドを実行しない場合、ストレージステータスは **HEALTH_WARN** になります。

```
# ceph osd require-osd-release luminous
```

関連情報

- ストレージクラスターに新しい OSD を追加してストレージ容量を拡張するには、Red Hat Ceph Storage3 の **管理ガイド** の **OSD の追加** セクションを参照してください。

Ceph Object Gateway ノードのアップグレード

このセクションでは、Ceph Object Gateway ノードを新しいバージョンにアップグレードする手順について説明します。

前提条件

- Red Hat は、HAProxy などのロードバランサーの背後に Ceph Object Gateway を配置することをお勧めします。ロードバランサーを使用する場合は、リクエストが処理されなくなったら、ロードバランサーから Ceph Object Gateway を削除します。

- **rgw_region_root_pool** パラメーターで指定されたリージョンプールのカスタム名を使用する場合は、Ceph 構成ファイルの **[global]** セクションに **rgw_zonegroup_root_pool** パラメーターを追加します。**rgw_zonegroup_root_pool** の値を **rgw_region_root_pool** と同じになるように設定します。次に例を示します。

```
[global]
rgw_zonegroup_root_pool = .us.rgw.root
```

手順

ストレージクラスター内の各 Ceph Object Gateway ノードで次の手順を実行します。一度に 1 つのノードのみをアップグレードします。

1. オンラインリポジトリーを使用して Red Hat Ceph Storage をインストールした場合は、2 つのリポジトリーを無効にします。

```
# subscription-manager repos --disable=rhel-7-server-rhceph-2.3-tools-rpms --disable=rhel-7-server-rhceph-2-installer-rpms
```

2. Red Hat Ceph Storage 3 Tools リポジトリーを有効にします。

```
[root@gateway ~]# subscription-manager repos --enable=rhel-7-server-rhceph-3-tools-els-rpms
```

3. Ceph Object Gateway プロセス (**ceph-radosgw**) を停止します。

```
# service ceph-radosgw stop
```

4. **ceph-radosgw** パッケージを更新します。

```
# yum update ceph-radosgw
```

5. 新しく作成された **/var/lib/ceph/radosgw/** および **/var/log/ceph/** ディレクトリーとそれらのコンテンツに対する所有者とグループの権限を **ceph** に変更します。

```
# chown -R ceph:ceph /var/lib/ceph/radosgw
# chown -R ceph:ceph /var/log/ceph
```

6. SELinux が強制モードまたは許容モードで実行するように設定されている場合は、次回の起動時に SELinux コンテキストのラベルを変更するように指示します。

```
# touch /.autorelabel
```

重要

SELinux はすべてのファイルシステムをトラバースし、誤ってラベル付けされたファイルを修正する必要があるため、再ラベル付けは完了するまでに長い時間がかかります。ディレクトリーの再ラベル付けを除外するには、再起動する前にディレクトリーを **/etc/selinux/fixfiles_exclude_dirs** ファイルに追加します。

7. **ceph-radosgw** プロセスを有効にします。

```
# systemctl enable ceph-radosgw.target
# systemctl enable ceph-radosgw@rgw.<hostname>
```

<hostname> を Ceph Object Gateway ホストの名前 (**gateway-node** など) に置き換えます。

```
# systemctl enable ceph-radosgw.target
# systemctl enable ceph-radosgw@rgw.gateway-node
```

8. Ceph Object Gateway ノードを再起動します。

```
# shutdown -r now
```

9. ロードバランサーを使用する場合は、Ceph Object Gateway ノードをロードバランサーに追加し直します。

関連項目

- Red Hat Enterprise Linux 用の Ceph Object Gateway ガイド

Ceph クライアントノードのアップグレード

Ceph のクライアントは

- Ceph ブロックデバイス
- OpenStack Nova コンピュートノード
- QEMU/KVM ハイパーバイザー
- Ceph クライアント側ライブラリーを使用するカスタムアプリケーション

Red Hat は、すべての Ceph クライアントが Ceph ストレージクラスターと同じバージョンを実行することを推奨しています。

前提条件

- パッケージのアップグレード中に Ceph クライアントノードに対するすべての I/O 要求を停止して、予期しないエラーが発生しないようにします

手順

1. ソフトウェアリポジトリーを使用して Red Hat Ceph Storage 2 クライアントをインストールした場合は、リポジトリーを無効にします。

```
# subscription-manager repos --disable=rhel-7-server-rhceph-2-tools-rpms --disable=rhel-7-server-rhceph-2-installer-rpms
```

注記

Red Hat Ceph Storage 2 クライアントに対して ISO ベースのインストールが実行された場合は、この最初のステップをスキップしてください。

2. クライアントノードで、Red Hat Ceph Storage Tools 3 リポジトリーを有効にします。

```
[root@gateway ~]# subscription-manager repos --enable=rhel-7-server-rhceph-3-tools-els-rpms
```

3. クライアントノードで、**ceph-common** パッケージを更新します。

```
# yum update ceph-common
```

ceph-common パッケージをアップグレードした後、Ceph クライアントサイドライブラリーに依存するアプリケーションを再起動します。

注記

QEMU/KVM インスタンスを実行している OpenStack Nova コンピュートノードをアップグレードする場合や、専用の QEMU/KVM クライアントを使用する場合には、インスタンスを再起動しても機能しないため、QEMU/KVM インスタンスを停止して起動してください。

付録H バージョン 2 と 3 の間の ANSIBLE 変数の変更

Red Hat Ceph Storage 3 では、**/usr/share/ceph-ansible/group_vars/** ディレクトリーにある構成ファイルの特定の変数が変更されたか、削除されました。以下の表は、すべての変更点を示しています。バージョン3へのアップグレード後は、**all.yml.sample** と **osds.yml.sample** を再度コピーし、これらの変更を反映させてください。詳細は、Red Hat Ceph ストレージクラスターのアップグレード を参照してください。

旧オプション	新しいオプション	ファイル
ceph_rhcs_cdn_install	**ceph_repository_type: cdn**	**all.yml**
ceph_rhcs_iso_install	**ceph_repository_type: iso**	**all.yml**
ceph_rhcs	**ceph_origin: repository** および **ceph_repository:rhcs** (デフォルトで有効)	**all.yml**
journal_collocation	**osd_scenario: collocated**	**osds.yml**
raw_multi_journal	**osd_scenario: non-collocated**	**osds.yml**
raw_journal_devices	**dedicated_devices**	**osds.yml**
dmcrytpt_journal_collocation	**dmcrypt: true** + **osd_scenario: collocated**	**osds.yml**
dmcrypt_dedicated_journal	**dmcrypt: true** + **osd_scenario: non-collocated**	**osds.yml**

付録I 既存の CEPH クラスターの ANSIBLE へのインポート

Ansible が Ansible なしでデプロイされたクラスターを使用するように設定することができます。たとえば、Red Hat Ceph Storage 1.3 クラスターを手動でバージョン 2 にアップグレードした場合は、以下の手順により Ansible を使用するように設定してください。

1. バージョン 1.3 からバージョン 2 に手動でアップグレードした後、管理ノードに Ansible をインストールおよび設定します。

2. Ansible 管理ノードに、クラスター内の全 Ceph ノードにパスワードレスの **ssh** アクセスがあることを確認します。詳しくは 「Ansible でパスワードなしの SSH を有効にする」 をご覧ください。

3. **root** で、**/etc/ansible/** ディレクトリーに Ansible の **group_vars** ディレクトリーへのシンボリックリンクを作成します。

```
# ln -s /usr/share/ceph-ansible/group_vars /etc/ansible/group_vars
```

4. **root** で **all.yml.sample** ファイルから **all.yml** ファイルを作成し、編集用に開きます。

```
# cd /etc/ansible/group_vars
# cp all.yml.sample all.yml
# vim all.yml
```

5. **group_vars/all.yml** で **generate_fsid** 設定を **false** に設定します。

6. **ceph fsid** を実行して、現在のクラスター **fsid** を取得します。

7. 取得した **fsid** を **group_vars/all.yml** に設定します。

8. Ceph ホストが含まれるように、**/etc/ansible/hosts** の Ansible インベントリーを変更します。**[mons]** セクションの下にモニター、**[osds]** セクションの下に OSD、および **[rgws]** セクションのゲートウェイを追加して、それらのロールをAnsible に特定します。

9. **ceph_conf_overrides** セクションが、**all.yml** ファイルの **[global]** セクション、**[osd]** セクション、**[mon]** セクション、および **[client]** セクションに使用される元の **ceph.conf** オプションで更新されていることを確認します。
   osd ジャーナル、**public_network**、**cluster_network** などのオプションはすでに **all.yml** に含まれているため、**ceph_conf_overrides** には追加しないでください。**all.yml** に含まれず、元の **ceph.conf** にあるオプションのみを **ceph_conf_overrides** に追加する必要があります。

10. **/usr/share/ceph-ansible/** ディレクトリーから Playbook を実行します。

```
# cd /usr/share/ceph-ansible/
# cp infrastructure-playbooks/take-over-existing-cluster.yml .
$ ansible-playbook take-over-existing-cluster.yml -u <username>
```

付録J ANSIBLE を使用して CEPH クラスターをパージする

Ansible を使用して Ceph クラスターをデプロイし、クラスターをパージする場合は、**infrastructure-playbooks** ディレクトリーにある **purge-cluster.yml Ansible** Playbook を使用します。

重要

Ceph クラスターをパージすると、クラスターの OSD に保存されているデータが失われます。

Ceph クラスターをパージする前に...

osds.yml ファイルの **osd_auto_discovery** オプションを確認してください。このオプションが **true** に設定されていると、パージは失敗します。この不具合を防ぐために、パージを実行する前に以下の手順を行ってください。

1. **osds.yml** ファイルで OSD デバイスを宣言します。詳しくは 「Red Hat Ceph Storage クラスターのインストール」 をご覧ください。
2. **osds.yml** ファイルの **osd_auto_discovery** オプションをコメントアウトします。

Ceph クラスターをパージするには...

1. **root** として **/usr/share/ceph-ansible/** ディレクトリーに移動します。

```
# cd /usr/share/ceph-ansible
```

2. **root** で、**purge-cluster.yml** Ansible playbook をカレントディレクトリにコピーします。

```
# cp infrastructure-playbooks/purge-cluster.yml .
```

3. **purge-cluster.yml** Ansible playbook を実行します:

```
$ ansible-playbook purge-cluster.yml
```