



## Red Hat OpenStack Platform 16.2

### オーバークラウドの既存 Red Hat Ceph Storage クラスターとの統合

オーバークラウドでスタンドアロンの Red Hat Ceph Storage を使用するための設定



# Red Hat OpenStack Platform 16.2 オーバークラウドの既存 Red Hat Ceph Storage クラスターとの統合

---

オーバークラウドでスタンドアロンの Red Hat Ceph Storage を使用するための設定

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/Integrating\_an\_Overcloud\_with\_an\_Existing\_Red\_Hat\_Ceph\_Storage\_Cluster.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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

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

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

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

Node.js<sup>®</sup> 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<sup>®</sup> 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

Red Hat OpenStack Platform (RHOSP) director を使用して、オーバークラウドを既存のスタンダードな Red Hat Ceph Storage クラスタと統合することができます。

## 目次

前書き .....	3
多様性を受け入れるオープンソースの強化 .....	4
RED HAT ドキュメントへのフィードバック (英語のみ) .....	5
第1章 オーバークラウドと CEPH STORAGE との統合 .....	6
1.1. 外部の CEPHFS を使用する SHARED FILE SYSTEMS サービス (MANILA) のデプロイ	6
1.2. 外部の CEPH OBJECT GATEWAY を使用するための CEPH OBJECT STORE の設定	7
第2章 オーバークラウドノードの準備 .....	8
2.1. 利用可能な RED HAT CEPH STORAGE パッケージの確認	8
2.1.1. ceph-ansible パッケージバージョンの確認	8
2.1.2. 事前にプロビジョニングされたノード用のパッケージの確認	8
2.2. 既存の CEPH STORAGE クラスターの設定	8
2.3. STACK ユーザーの初期化	10
2.4. ノードの登録	11
2.5. ノードの手動でのタグ付け	13
第3章 既存の CEPH STORAGE クラスターとの統合 .....	14
3.1. CEPH-ANSIBLE パッケージのインストール	14
3.2. カスタム環境ファイルの作成	14
3.3. ロールへのノードとフレーバーの割り当て	16
3.4. CEPH STORAGE を使用する RED HAT OPENSTACK PLATFORM 向けの CEPH コンテナ	16
3.5. オーバークラウドのデプロイ	17
3.5.1. CephFS による Shared File Systems サービスの新たな環境ファイルの追加	17
3.5.2. オブジェクトストレージ用外部 Ceph Object Gateway (RGW) の新規環境ファイルの追加	18
3.5.3. テンプレートおよび環境ファイルの呼び出し	20
3.5.4. openstack overcloud deploy コマンドのオプション	20
3.5.5. オーバークラウドの作成ステータスの表示	21
第4章 外部 CEPH STORAGE クラスターとの統合の確認 .....	22
4.1. ID の収集	22
4.2. CEPH STORAGE クラスターの確認	23
4.3. 検証の失敗のトラブルシューティング	25
第5章 オーバークラウドへのアクセス .....	27



# 前書き

## 多様性を受け入れるオープンソースの強化

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



## RED HAT ドキュメントへのフィードバック (英語のみ)

弊社ドキュメントに対するご意見をお聞かせください。ドキュメントの改善点があればお知らせください。

### ドキュメントへのダイレクトフィードバック (DDF) 機能の使用 (英語版のみ)

特定の文章、段落、またはコードブロックに対して直接コメントを送付するには、DDF の **Add Feedback** 機能を使用してください。なお、この機能は英語版のドキュメントでのみご利用いただけます。

1. **Multi-page HTML** 形式でドキュメントを表示します。
2. ドキュメントの右上隅に **Feedback** ボタンが表示されていることを確認してください。
3. コメントするテキスト部分をハイライト表示します。
4. **Add Feedback** をクリックします。
5. **Add Feedback** フィールドにコメントを入力します。
6. (オプション) ドキュメントチームが連絡を取り問題についてお伺いできるように、ご自分のメールアドレスを追加します。
7. **Submit** をクリックします。

## 第1章 オーバークラウドと CEPH STORAGE との統合

Red Hat OpenStack Platform director は、オーバークラウドと呼ばれるクラウド環境を作成します。director を使用して、Red Hat Ceph Storage との統合など、オーバークラウドの追加機能を設定することができます。オーバークラウドを、director で作成した Ceph Storage クラスターまたは既存の Ceph Storage クラスターと統合することができます。

Red Hat Ceph Storage についての詳しい情報は、『[Red Hat Ceph Storage アーキテクチャーガイド](#)』を参照してください。

### 1.1. 外部の CEPHFS を使用する SHARED FILE SYSTEMS サービス (MANILA) のデプロイ

Red Hat OpenStack Platform director は、CephFS を使用する Shared File Systems サービス (manila) をデプロイすることができます。CephFS は、ネイティブ CephFS プロトコルまたは NFS プロトコルを通じて使用できます。

これらのストレージプロトコルについての詳しい情報は、『[Deploying the Shared File Systems service with CephFS through NFS](#)』の「[CephFS with native driver](#)」および「[CephFS through NFS](#)」を参照してください。



#### 重要

NFS バックエンドに CephFS を使用した Shared File Systems (manila) サービスを使用して、Manila CSI により OpenShift Container Platform にファイル共有を提供する場合、Red Hat ではサポート例外を要求します。詳細は、Red Hat のサポートにお問い合わせください。



#### 重要

ネイティブ CephFS 共有ファイルシステムを使用するには、クライアントが Ceph パブリックネットワークにアクセスする必要があります。オーバークラウドを既存の Ceph クラスターと統合する際に、director は Ceph パブリックネットワークとして指定する分離ストレージネットワークを作成しません。このネットワークがすでに存在していることを前提とします。Ceph パブリックネットワークへの直接アクセスを許可しないでください。その代わりに、テナントが Ceph パブリックネットワークに接続するためのルーターを作成するのを許可します。

セキュリティに関する考慮事項に関する詳しい情報は、『[CephFS Back End Guide for the Shared Files Systems Service](#)』の「[Native CephFS back end security](#)」を参照してください。

NFS プロトコルを通じて CephFS を使用する場合、director は Pacemaker (PCS) が管理するコントローラーノードに NFS-Ganesha ゲートウェイをデプロイします。PCS は、アクティブ/パッシブ構成を使用してクラスターの可用性を管理します。



## 注記

この機能は、Ceph 4 サイクルの Ceph Storage 4.1 以降、または Ceph 5 サイクルの Ceph Storage 5.0 以降でサポートされます。ceph-ansible パッケージの最新バージョンをアンダークラウドにインストールする必要があります。お使いのシステムにインストールされている Ceph Storage がどのリリースかを判断する方法は、「[Red Hat Ceph Storage releases and corresponding Ceph package versions](#)」を参照してください。

アンダークラウド上で **ceph-ansible** パッケージを更新する方法は、「[ceph -ansible パッケージのインストール](#)」を参照してください。

## 前提条件

外部の Ceph Storage クラスタを使用して Shared File Systems サービスを設定するには、以下の前提条件を満たす必要があります。

- 外部の Ceph Storage クラスタにはアクティブな MDS が必要です。
- 外部の Ceph Storage クラスタには、CephFS データ (**ManilaCephFSDataPoolName**) および CephFS メタデータプール (**ManilaCephFSMetadataPoolName**) の値に基づいた CephFS ファイルシステムが必要です。詳細は、「[カスタム環境ファイルの作成](#)」を参照してください。
- 外部の Ceph Storage クラスタには、Shared File Systems サービス用の **cephx** クライアント名およびキーが必要です。詳細は、「[カスタム環境ファイルの作成](#)」を参照してください。

Red Hat Ceph Storage の詳細は、『[Red Hat Ceph Storage ファイルシステムガイド](#)』を参照してください。

## 1.2. 外部の CEPH OBJECT GATEWAY を使用するための CEPH OBJECT STORE の設定

Red Hat OpenStack Platform (RHOSP) director は、外部の Ceph Object Gateway (RGW) を Object Store サービスとして設定することをサポートしています。外部 RGW サービスで認証するには、Identity サービス (keystone) のユーザーとそのロールを確認するように RGW を設定する必要があります。

外部 Ceph Object Gateway の設定方法に関する詳細は、『[Ceph Object Gateway を使用した Keystone の使用ガイド](#)』の「[Keystone 認証を使用するように Ceph Object Gateway を設定](#)」を参照してください。

## 第2章 オーバークラウドノードの準備

Red Hat Ceph Storage クラスターとの統合方法を説明するのに使用されるオーバークラウドのデプロイメントは、6つのノードで構成されます。

- 高可用性のコントローラーノード 3 台
- コンピュートノード 3 台

Red Hat Ceph Storage クラスターには、director ではなく Ceph 管理ツールを使用してオーバークラウドとは独立に管理する独自のノードがあります。詳しい情報は、[Red Hat Ceph Storage](#) のドキュメントを参照してください。

### 2.1. 利用可能な RED HAT CEPH STORAGE パッケージの確認

オーバークラウドのデプロイメントが失敗しないようにするには、必要なパッケージがサーバーに存在することを確認します。

#### 2.1.1. ceph-ansible パッケージバージョンの確認

アンダークラウドには Ansible ベースの検証が含まれ、これを実行してオーバークラウドをデプロイする前に潜在的な問題を特定することができます。これらの検証は、典型的な問題が発生する前にそれらを特定し、オーバークラウドのデプロイメントの失敗を回避するのに役立ちます。

##### 手順

- 必要な **ceph-ansible** パッケージ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。

```
$ ansible-playbook -i /usr/bin/tripleo-ansible-inventory /usr/share/ansible/validation-playbooks/ceph-ansible-installed.yaml
```

#### 2.1.2. 事前にプロビジョニングされたノード用のパッケージの確認

Red Hat Ceph Storage (RHCS) は、特定のパッケージセットを持つオーバークラウドノードにのみサービスを提供することができます。事前にプロビジョニングされたノードを使用する場合には、これらのパッケージが存在することを確認することができます。

事前にプロビジョニングされたノードの詳細は、「[Configuring a basic overcloud with pre-provisioned nodes](#)」を参照してください。

##### 手順

- 事前にプロビジョニングされたノードに必要なパッケージが含まれていることを確認します。

```
ansible-playbook -i /usr/bin/tripleo-ansible-inventory /usr/share/ansible/validation-playbooks/ceph-dependencies-installed.yaml
```

### 2.2. 既存の CEPH STORAGE クラスターの設定

Ceph Storage クラスターの OSD プールを作成し、機能を定義し、キーおよび ID を作成します。

##### 手順

## 1. お使いの環境に適した Ceph クラスタに以下のプールを作成します。

- OpenStack Block Storage (cinder) のストレージ:

```
[root@ceph ~]# ceph osd pool create volumes <pgnum>
```

- OpenStack Image Storage (glance) のストレージ:

```
[root@ceph ~]# ceph osd pool create images <pgnum>
```

- インスタンスのストレージ:

```
[root@ceph ~]# ceph osd pool create vms <pgnum>
```

- OpenStack Block Storage Backup (cinder-backup) のストレージ:

```
[root@ceph ~]# ceph osd pool create backups <pgnum>
```

- OpenStack Telemetry Metrics (gnocchi) のストレージ:

```
[root@ceph ~]# ceph osd pool create metrics <pgnum>
```

オーバークラウドが CephFS がサポートする Shared File Systems サービス (manila) をデプロイする場合には、CephFS データおよびメタデータプールも作成します。

```
[root@ceph ~]# ceph osd pool create manila_data <pgnum>
[root@ceph ~]# ceph osd pool create manila_metadata <pgnum>
```

**<pgnum>** は配置グループの数に置き換えます。1 OSD あたり約 100 個の配置グループがベストプラクティスです。たとえば、OSD の合計数を 100 で乗算して、レプリカ数で除算します (**osd pool default size**)。適切な値を判断するには [Ceph Placement Groups \(PGs\) per Pool Calculator](#) を使用することを推奨します。

2. 以下の機能を指定して **client.openstack** ユーザーを Ceph クラスタに作成します。

- **cap\_mgr**: "allow \*"
- **cap\_mon**: profile rbd
- **cap\_osd**: profile rbd pool=volumes, profile rbd pool=vms, profile rbd pool=images, profile rbd pool=backups, profile rbd pool=metrics

```
[root@ceph ~]# ceph auth add client.openstack mgr 'allow *' mon 'profile rbd' osd 'profile
rbd pool=volumes, profile rbd pool=vms, profile rbd pool=images, profile rbd
pool=backups, profile rbd pool=metrics'
```

3. **client.openstack** ユーザー向けに作成された Ceph クライアントキーをメモします。

```
[root@ceph ~]# ceph auth list
...
[client.openstack]
key = AQC+vYNXgDAgAhAAc8UoYt+OTz5uhV7ltLdwUw==
caps mgr = "allow *"
caps mon = "profile rbd"
```

```
caps osd = "profile rbd pool=volumes, profile rbd pool=vms, profile rbd pool=images, profile
rbd pool=backups, profile rbd pool=metrics"
```

```
...
```

この例の **key** 値 (**AQC+vYNXgDAgAhAAc8UoYt+OTz5uhV7ItLdwUw==**) は Ceph クライアントキーです。**ceph-ansible** パッケージをインストールするには、このキーが必要です。

4. オーバークラウドが CephFS でサポートされる Shared File System サービスをデプロイする場合は、以下の機能を備えた **client.manila** ユーザーを Ceph クラスターに作成します。

- **cap\_mds: allow \***
- **cap\_mgr: allow \***
- **cap\_mon: allow r, allow command "auth del", allow command "auth caps", allow command "auth get", allow command "auth get-or-create"**
- **cap\_osd: allow rw**

```
[root@ceph ~]# ceph auth add client.manila mon 'allow r, allow command "auth del",
allow command "auth caps", allow command "auth get", allow command "auth get-or-
create"' osd 'allow rw' mds 'allow *' mgr 'allow *'
```

5. manila クライアント名およびオーバークラウドデプロイメントテンプレートで使用するキー値をメモします。

```
[root@ceph ~]# ceph auth get-key client.manila
AQDQ991cAAAAABAA0aXFrTnjH9aO39P0iVvYyg==
```

6. Ceph Storage クラスターのファイルシステム ID をメモします。この値は、**[global]** のセクションで、クラスターの設定ファイルにある **fsid** の設定で指定されています。

```
[global]
fsid = 4b5c8c0a-ff60-454b-a1b4-9747aa737d19
...
```



### 注記

Ceph Storage クラスターの設定ファイルに関する詳細は、『**Red Hat Ceph Storage Configuration Guide**』の「[Ceph configuration](#)」を参照してください。

ceph-ansible パッケージをインストールする際に、Ceph クライアントキーおよびファイルシステム ID、および manila クライアント ID およびキーを使用します。詳細は、「[ceph-ansible パッケージのインストール](#)」を参照してください。

## 2.3. STACK ユーザーの初期化

stack ユーザーを初期化し、director CLI ツールにアクセスするのに使用する認証情報を設定します。

### 手順

1. director ホストに **stack** ユーザーとしてログインします。
2. 以下のコマンドを入力して director の設定を初期化します。

```
$ source ~/stackrc
```

## 2.4. ノードの登録

インベントリーファイルには、ノードに関するハードウェア情報および電源管理情報が含まれます。director でノードを設定して登録するためのインベントリーファイルを作成します。

### 手順

1. インベントリーファイルを作成します。ノード定義のテンプレートの例 **instackenv.json** を参考として使用してください。

```
{
  "nodes":[
    {
      "mac":[
        "bb:bb:bb:bb:bb:bb"
      ],
      "cpu":"4",
      "memory":"6144",
      "disk":"40",
      "arch":"x86_64",
      "pm_type":"pxe_ipmitool",
      "pm_user":"admin",
      "pm_password":"p@55w0rd!",
      "pm_addr":"192.0.2.205"
    },
    {
      "mac":[
        "cc:cc:cc:cc:cc:cc"
      ],
      "cpu":"4",
      "memory":"6144",
      "disk":"40",
      "arch":"x86_64",
      "pm_type":"pxe_ipmitool",
      "pm_user":"admin",
      "pm_password":"p@55w0rd!",
      "pm_addr":"192.0.2.206"
    },
    {
      "mac":[
        "dd:dd:dd:dd:dd:dd"
      ],
      "cpu":"4",
      "memory":"6144",
      "disk":"40",
      "arch":"x86_64",
      "pm_type":"pxe_ipmitool",
      "pm_user":"admin",
      "pm_password":"p@55w0rd!",
      "pm_addr":"192.0.2.207"
    },
    {
      "mac":[
```

```

        "ee:ee:ee:ee:ee:ee"
    ],
    "cpu": "4",
    "memory": "6144",
    "disk": "40",
    "arch": "x86_64",
    "pm_type": "pxe_ipmitool",
    "pm_user": "admin",
    "pm_password": "p@55w0rd!",
    "pm_addr": "192.0.2.208"
  }
  {
    "mac": [
      "ff:ff:ff:ff:ff:ff"
    ],
    "cpu": "4",
    "memory": "6144",
    "disk": "40",
    "arch": "x86_64",
    "pm_type": "pxe_ipmitool",
    "pm_user": "admin",
    "pm_password": "p@55w0rd!",
    "pm_addr": "192.0.2.209"
  }
  {
    "mac": [
      "gg:gg:gg:gg:gg:gg"
    ],
    "cpu": "4",
    "memory": "6144",
    "disk": "40",
    "arch": "x86_64",
    "pm_type": "pxe_ipmitool",
    "pm_user": "admin",
    "pm_password": "p@55w0rd!",
    "pm_addr": "192.0.2.210"
  }
]
}

```

- そのファイルを stack ユーザーのホームディレクトリー (`/home/stack/instackenv.json`) に保存します。
- instackenv.json** インベントリーファイルを director にインポートします。

```
$ openstack overcloud node import ~/instackenv.json
```

**openstack overcloud node import** コマンドは、インベントリーファイルをインポートし、各ノードを director に登録します。

- カーネルと ramdisk イメージを各ノードに割り当てます。

```
$ openstack overcloud node configure <node>
```

director でのノードの登録、設定が完了しました。



## 2.5. ノードの手動でのタグ付け

各ノードの登録後、ハードウェアを検査して、ノードを特定のプロファイルにタグ付けする必要があります。プロファイルタグを使用してノードをフレーバーに照合してから、フレーバーをデプロイメントロールに割り当てます。

### 手順

1. ハードウェアのイントロスペクションをトリガーして、各ノードのハードウェア属性を取得します。

```
$ openstack overcloud node introspect --all-manageable --provide
```

- **--all-manageable** オプションを使用して、管理状態にあるノードのみをイントロスペクションします。ここでは、すべてのノードが管理状態にあります。
- **--provide** オプションは、イントロスペクション後に全ノードを **active** の状態にリセットします。



### 重要

このプロセスが正常に完了したことを確認します。ベアメタルノードの場合には、通常 15 分ほどかかります。

2. ノード一覧を取得して UUID を識別します。

```
$ openstack baremetal node list
```

3. 各ノードの **properties/capabilities** パラメーターに **profile** オプションを追加して、ノードを特定のプロファイルに手動でタグ付けします。 **profile** オプションを追加すると、適切なプロファイルにノードをタグ付けします。

手動でのタグ付けの代わりに、Automated Health Check (AHC) ツールを設定し、ベンチマークデータに基づいて、多数のノードに自動でタグ付けすることができます。

たとえば、3つのノードが **control** プロファイルを使用し、別の3つのノードが **compute** プロファイルを使用するようにタグ付けするには、以下の **profile** オプションを作成します。

```
$ ironic node-update 1a4e30da-b6dc-499d-ba87-0bd8a3819bc0 add
properties/capabilities='profile:control,boot_option:local'
$ ironic node-update 6faba1a9-e2d8-4b7c-95a2-c7fbd12129a add
properties/capabilities='profile:control,boot_option:local'
$ ironic node-update 5e3b2f50-fcd9-4404-b0a2-59d79924b38e add
properties/capabilities='profile:control,boot_option:local'
$ ironic node-update 484587b2-b3b3-40d5-925b-a26a2fa3036f add
properties/capabilities='profile:compute,boot_option:local'
$ ironic node-update d010460b-38f2-4800-9cc4-d69f0d067efe add
properties/capabilities='profile:compute,boot_option:local'
$ ironic node-update d930e613-3e14-44b9-8240-4f3559801ea6 add
properties/capabilities='profile:compute,boot_option:local'
```

## 第3章 既存の CEPH STORAGE クラスターとの統合

Red Hat OpenStack Platform を既存の Ceph Storage クラスターと統合するには、**ceph-ansible** パッケージをインストールする必要があります。その後、カスタムの環境ファイルを作成して、ロールにノードとフレーバーを割り当てることができます。

### 3.1. CEPH-ANSIBLE パッケージのインストール

Red Hat OpenStack Platform director は **ceph-ansible** を使用して既存の Ceph Storage クラスターと統合しますが、デフォルトでは **ceph-ansible** はアンダークラウドにインストールされていません。

#### 手順

- 以下のコマンドを入力して、アンダークラウドに **ceph-ansible** パッケージをインストールしてください。

```
$ sudo dnf install -y ceph-ansible
```

### 3.2. カスタム環境ファイルの作成

director は **ceph-ansible** にパラメーターを提供し、環境ファイルを使用して外部の Ceph Storage クラスターと統合します。

- **/usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml**

外部の CephFS を使用して Shared File Systems サービス (manila) をデプロイする場合、別の環境ファイルで追加のパラメーターを提供します。

- ネイティブ CephFS の場合、環境ファイルは **/usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsnative-config.yaml** です。
- NFS バックエンドに CephFS を使用する場合、環境ファイルは **/usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsganeshasha-config.yaml** です。

director はこれらの環境ファイルをデプロイメント時に呼び出し、既存の Ceph Storage クラスターをオーバークラウドに統合します。詳しい情報は、「[オーバークラウドのデプロイ](#)」を参照してください。

統合を設定するには、Ceph Storage クラスターの詳細を director に提供する必要があります。そのため、カスタム環境ファイルを使用してデフォルト設定を上書きします。

#### 手順

1. カスタム環境ファイルを作成します。  
**/home/stack/templates/ceph-config.yaml**
2. ファイルに **parameter\_defaults:** セクションを追加します。

```
parameter_defaults:
```

3. **parameter\_defaults** を使用して、**/usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml** で上書きするすべてのパラメーターを設定します。少なくとも以下のパラメーターを設定する必要があります。

- **CephClientKey:** Ceph Storage クラスターの Ceph クライアントキー。これは、「既存の [Ceph Storage クラスターの設定](#)」で取得した **key** の値です。たとえば、**AQDLOh1VgEp6FRAAFzT7Zw+Y9V6JJExQAsRnRQ==** です。
- **CephClusterFSID:** Ceph Storage クラスターのファイルシステム ID。これは、「既存の [Ceph Storage クラスターの設定](#)」で取得した Ceph Storage クラスターの設定ファイルの **fsid** の値です。たとえば、**4b5c8c0a-ff60-454b-a1b4-9747aa737d19** です。
- **CephExternalMonHost:** Ceph Storage クラスターの全 MON ホストの IP をコンマ区切りにしたリスト (例: **172.16.1.7, 172.16.1.8**)。以下は例になります。

```
parameter_defaults:
  CephClientKey: AQDLOh1VgEp6FRAAFzT7Zw+Y9V6JJExQAsRnRQ==
  CephClusterFSID: 4b5c8c0a-ff60-454b-a1b4-9747aa737d19
  CephExternalMonHost: 172.16.1.7, 172.16.1.8
```

4. (オプション) 実際の Ceph Storage クラスターに一致するように、デフォルトのプール名または RHOSP クライアントユーザー名を上書きできます。

- **CephClientUserName: openstack**
- **NovaRbdPoolName: vms**
- **CinderRbdPoolName: volumes**
- **GlanceRbdPoolName: images**
- **CinderBackupRbdPoolName: backups**
- **GnocchiRbdPoolName: metrics**

5. CephFS がサポートする Shared File Systems サービスをデプロイする場合は、データおよびメタデータプールの名前を設定します。

```
ManilaCephFSDataPoolName: manila_data
ManilaCephFSMetadataPoolName: manila_metadata
```



### 注記

これらの名前が、作成済みのプールの名前と一致していることを確認します。

6. Shared File Systems サービス用に作成したクライアントキーとそのキーの Ceph ユーザーの名前を設定します。

```
ManilaCephFSCephFSAuthId: 'manila'
CephManilaClientKey: 'AQDQ991cAAAAABAA0aXFrTnjH9aO39P0iVvYyg=='
```



### 注記

上書きしない限り、デフォルトのクライアントユーザー名 **ManilaCephFSCephFSAuthId** は **manila** です。**CephManilaClientKey** は常に必要です。

7. カスタム環境ファイルにオーバークラウドパラメーターを追加することができます。たとえば、**neutron** のネットワーク種別を **vxlan** に設定するには、**parameter\_defaults** に以下の設定を追加します。

```
NeutronNetworkType: vxlan
```

カスタム環境ファイルを作成したら、オーバークラウドのデプロイ時にこのファイルを追加する必要があります。オーバークラウドのデプロイに関する詳細は、「[オーバークラウドのデプロイ](#)」を参照してください。

### 3.3. ロールへのノードとフレーバーの割り当て

オーバークラウドのデプロイメントプランニングでは、各ロールに割り当てるノード数とフレーバーを指定する必要があります。すべての heat テンプレートのパラメーターと同様に、これらのロールの仕様はカスタム環境ファイル (ここでは `/home/stack/templates/ceph-config.yaml`) の **parameter\_defaults** セクションで宣言する必要があります。

以下の表のパラメーターを使用して、ロールを設定します。

表3.1 オーバークラウドノードのロールとフレーバー

heat テンプレートのパラメーター	説明
ControllerCount	スケールアウトするコントローラーノード数
OvercloudControlFlavor	コントローラーノードに使用するフレーバー ( <b>control</b> )
ComputeCount	スケールアウトするコンピュートノード数
OvercloudComputeFlavor	コンピュートノード ( <b>compute</b> ) に使用するフレーバー

たとえば、オーバークラウドが各ロール (Controller および Compute) に 3 つずつノードをデプロイするように設定するには、**parameter\_defaults** に以下の設定を追加します。

```
parameter_defaults:
  ControllerCount: 3
  ComputeCount: 3
  OvercloudControlFlavor: control
  OvercloudComputeFlavor: compute
```



#### 注記

heat テンプレートパラメーターのより詳細な一覧は、『[Director Installation and Usage](#)』の「[Creating the Overcloud with the CLI Tools](#)」を参照してください。

### 3.4. CEPH STORAGE を使用する RED HAT OPENSTACK PLATFORM 向けの CEPH コンテナ

RHOSP が Ceph を使用するように設定するには、Ceph コンテナが必要です。これは、外部の Ceph

クラスターの場合でも同じです。Red Hat Enterprise Linux 8 と互換性を持たせるには、Red Hat OpenStack Platform (RHOSP) 16 には Red Hat Ceph Storage 4 が必要です。Ceph Storage 4 コンテナは、[registry.redhat.io](https://registry.redhat.io) (認証が必要なレジストリー) でホストされます。

heat パラメータ **ContainerImageRegistryCredentials** を使用して、[registry.redhat.io](https://registry.redhat.io) で認証することができます。詳細は、「[Container image preparation parameters](#)」を参照してください。

### 3.5. オーバークラウドのデプロイ

作成した環境ファイルを使用して、オーバークラウドをデプロイします。



#### 注記

アンダークラウドのインストール時に、**undercloud.conf** ファイルに **generate\_service\_certificate=false** を設定します。設定しない場合は、オーバークラウドのデプロイ時にトラストアンカーを挿入する必要があります。トラストアンカーの挿入方法についての詳細は、『[Advanced Overcloud Customization](#)』の「[Enabling SSL/TLS on Overcloud Public Endpoints](#)」を参照してください。

#### 手順

- オーバークラウドの作成には、**openstack overcloud deploy** コマンドに追加の引数を指定する必要があります。

```
$ openstack overcloud deploy --templates \
  -e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml \
  -e /home/stack/templates/ceph-config.yaml \
  -e --ntp-server pool.ntp.org \
```

上記のコマンド例は、以下のオプションを使用します。

- **--templates** - デフォルトの heat テンプレートコレクション (`/usr/share/openstack-tripleo-heat-templates/`) からオーバークラウドを作成します。
- **-e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml**: 既存の Ceph クラスターをオーバークラウドに統合するように director を設定します。
- **-e /home/stack/templates/ceph-config.yaml**: **-e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml** の設定するデフォルトを上書きするためのカスタム環境ファイルを追加します。ここでは、「[ceph-ansible パッケージのインストール](#)」で作成したカスタム環境ファイルです。
- **--ntp-server pool.ntp.org**: NTP サーバーを設定します。

#### 3.5.1. CephFS による Shared File Systems サービスの新たな環境ファイルの追加

CephFS による Shared File Systems サービスを使用するオーバークラウドをデプロイする場合には、新たな環境ファイルを追加する必要があります。

#### 手順

1. 以下のいずれかのオプションを使用して、新たな環境ファイルを作成して追加します。

- ネイティブ CephFS バックエンドドライバーを使用するオーバークラウドをデプロイする場合には、`/usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsnative-config.yaml` を使用します。
  - NFS バックエンドに CephFS を使用するオーバークラウドをデプロイする場合は、`/usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsganesha-config.yaml` を使用します。  
NFS バックエンドに CephFS を使用する場合には、カスタムの Controller ロールをデプロイして、Ganesha CephFS を NFS ゲートウェイで実行する必要があります。また、このロールは StorageNFS 分離ネットワークを設定して、ファイル共有をクライアントに提供します。StorageNFS ネットワークおよびカスタム Controller ロールに関する詳細は、『[Deploying the Shared File Systems service with CephFS through NFS](#)』の「[Deploying the updated environment](#)」を参照してください。
2. 使用する CephFS バックエンドに応じて、`openstack overcloud deploy` コマンドの形式を変更します。

- ネイティブ CephFS の場合:

```
$ openstack overcloud deploy --templates \
-e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-
ansible-external.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsnative-
config.yaml \
-e /home/stack/templates/ceph-config.yaml \
-e --ntp-server pool.ntp.org
```

- NFS バックエンドに CephFS を使用する場合:

```
$ openstack overcloud deploy --templates \
-n /usr/share/openstack-tripleo-heat-templates/network_data_ganesha.yaml \
-r /home/stack/custom_roles.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-
ansible-external.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsganesha-
config.yaml \
-e /home/stack/templates/ceph-config.yaml \
-e --ntp-server pool.ntp.org
```



### 注記

カスタムの `ceph-config.yaml` 環境ファイルは、`ceph-ansible-external.yaml` ファイル、および `manila-cephfsnative-config.yaml` ファイルまたは `manila-cephfsganesha-config.yaml` ファイルのいずれかのパラメーターを上書きします。したがって、`ceph-ansible-external.yaml`、および `manila-cephfsnative-config.yaml` または `manila-cephfsganesha-config.yaml` のいずれかの後に、デプロイメントコマンドにカスタムの `ceph-config.yaml` 環境ファイルを追加します。

## 3.5.2. オブジェクトストレージ用外部 Ceph Object Gateway (RGW) の新規環境ファイルの追加

オブジェクトストレージ用に既存の RGW サービスを使用するオーバークラウドをデプロイする場合には、さらに環境ファイルを追加する必要があります。

## 手順

1. カスタム環境ファイル (**swift-external-params.yaml** 等) に以下の **parameter\_defaults** を追加し、実際のデプロイメントに合わせて値を調整します。

```
parameter_defaults:
  ExternalSwiftPublicUrl: 'http://<Public RGW endpoint or
loadbalancer>:8080/swift/v1/AUTH_%(project_id)s'
  ExternalSwiftInternalUrl: 'http://<Internal RGW endpoint>:8080/swift/v1/AUTH_%(project_id)s'
  ExternalSwiftAdminUrl: 'http://<Admin RGW endpoint>:8080/swift/v1/AUTH_%(project_id)s'
  ExternalSwiftUserTenant: 'service'
  SwiftPassword: 'choose_a_random_password'
```

## 注記

サンプルコードスニペットには、お使いの環境で使用する値とは異なるパラメーター値が含まれる場合があります。

- リモート RGW インスタンスがリスンするデフォルトのポートは **8080** です。外部 RGW の設定方法によっては、ポートが異なる場合があります。
- オーバークラウドで作成した **swift** ユーザーは、**SwiftPassword** パラメーターで定義したパスワードを使用します。**rgw\_keystone\_admin\_password** を使用し、Identity サービスに対する認証に同じパスワードを使用するように外部 RGW インスタンスを設定する必要があります。

2. Ceph 設定ファイルに以下のコードを追加して、Identity サービスを使用するように RGW を設定します。変数の値を実際の環境に応じて置き換えます。

```
rgw_keystone_api_version = 3
rgw_keystone_url = http://<public Keystone endpoint>:5000/
rgw_keystone_accepted_roles = member, Member, admin
rgw_keystone_accepted_admin_roles = ResellerAdmin, swiftoperator
rgw_keystone_admin_domain = default
rgw_keystone_admin_project = service
rgw_keystone_admin_user = swift
rgw_keystone_admin_password =
<password_as_defined_in_the_environment_parameters>
rgw_keystone_implicit_tenants = true
rgw_keystone_revocation_interval = 0
rgw_s3_auth_use_keystone = true
rgw_swift_versioning_enabled = true
rgw_swift_account_in_url = true
```





## 注記

デフォルトでは、director は Identity サービスに以下のロールとユーザーを作成します。

- rgw\_keystone\_accepted\_admin\_roles: ResellerAdmin, swiftoperator
- rgw\_keystone\_admin\_domain: default
- rgw\_keystone\_admin\_project: service
- rgw\_keystone\_admin\_user: swift

3. デプロイメントに該当するその他の環境ファイルと共に、追加の環境ファイルを指定して、オーバークラウドをデプロイします。

```
openstack overcloud deploy --templates \
-e <your_environment_files>
-e /usr/share/openstack-tripleo-heat-templates/environments/swift-external.yaml
-e swift-external-params.yaml
```

### 3.5.3. テンプレートおよび環境ファイルの呼び出し

アンサーファイルを使用して、すべてのテンプレートおよび環境ファイルを呼び出すこともできます。たとえば、以下のコマンドを使用して、同一のオーバークラウドをデプロイすることができます。

```
$ openstack overcloud deploy \
--answers-file /home/stack/templates/answers.yaml \
--ntp-server pool.ntp.org
```

この場合、アンサーファイル **/home/stack/templates/answers.yaml** の内容は以下のようになります。

```
templates: /usr/share/openstack-tripleo-heat-templates/
environments:
- /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml
\
- /home/stack/templates/ceph-config.yaml \
```

詳しくは、『[Director Installation and Usage](#)』の「[Including environment files in an overcloud deployment](#)」を参照してください。

### 3.5.4. openstack overcloud deploy コマンドのオプション

オプションの **openstack overcloud deploy** コマンド引数を使用すると、オーバークラウドのデプロイ方法をより柔軟に制御できるようになります。

#### 手順

- 以下のコマンドを入力して、**openstack overcloud deploy** コマンドで使用できるオプションの完全リストを確認できます。

```
$ openstack help overcloud deploy
```



詳細は、『[Director Installation and Usage](#)』の「[Configuring a basic overcloud with the CLI tools](#)」を参照してください。

### 3.5.5. オーバークラウドの作成ステータスの表示

オーバークラウドの作成プロセスが開始され、director によりノードがプロビジョニングされます。このプロセスは完了するまで多少時間がかかります。

#### 手順

- オーバークラウドの作成のステータスを確認するには、**stack** ユーザーとして別のターミナルを開き、以下のコマンドを入力します。

```
$ source ~/stackrc
$ openstack stack list --nested
```

## 第4章 外部 CEPH STORAGE クラスターとの統合の確認

オーバークラウドをデプロイしたら、Red Hat OpenStack Platform (RHOSP) サービスが Red Hat Ceph Storage クラスターに書き込むことができることを確認します。



### 警告

RHOSP では、Ceph clone 形式 v2 以降の使用はサポートされません。Ceph clone 形式の v2 が有効な Ceph クラスターからイメージまたはボリュームを削除すると、予測できない動作が発生し、データが失われる可能性があります。そのため、Ceph クローンの形式 v2 を有効にする以下のいずれかの方法を使用しないでください。

- `rbd デフォルトクローン形式の設定 = 2`
- `ceph osd set-require-min-compat-client mimic` の実行

### 4.1. ID の収集

Red Hat Ceph Storage クラスターが統合されていることを確認するには、まずイメージ、コンピュートインスタンス、およびボリュームを作成し、それぞれの ID を収集する必要があります。

#### 手順

1. **Image サービス (glance) でイメージを作成します。**

イメージの作成方法についての詳細は、『[Creating and Managing Images](#)』の「[Import an image](#)」を参照してください。

2. **後で使用できるようにイメージ ID を記録します。**

3. **Compute (nova) インスタンスを作成します。**

インスタンスの作成方法についての詳細は、『[インスタンスの作成および管理](#)』の「[インスタンスの作成](#)」を参照してください。

4. **後で使用できるようにインスタンス ID を記録します。**

5. **Block Storage (cinder) ボリュームを作成します。**

**Block Storage** ボリュームの作成方法についての詳細は、『Storage Guide』の「[Create a volume](#)」を参照してください。

6. 後で使用できるようにボリューム ID を記録します。

## 4.2. CEPH STORAGE クラスターの確認

外部 **Ceph Storage** クラスターを設定する場合、プールおよび **client.openstack** ユーザーを作成して、これらのプールにアクセスします。オーバークラウドをデプロイしたら、**client.openstack** ユーザーの認証情報が含まれるファイルを使用して、Red Hat OpenStack Platform (RHOSP) プールの内容を一覧表示できます。

プールの内容を一覧表示し、**Image** サービス (**glance**) イメージ、インスタンス、およびボリュームの ID が **Ceph Storage** クラスターに存在することを確認します。

### 手順

1. **source** コマンドでアンダークラウドの認証情報を読み込みます。

```
[stack@undercloud-0 ~]$ source stackrc
```

2. 利用可能なサーバーを一覧表示して、システム上のノードの IP アドレスを取得します。

```
(undercloud) [stack@undercloud-0 ~]$ openstack server list
```

```
+-----+-----+-----+
| ID | Name | Status | Networks | Image | Flavor |
+-----+-----+-----+
| d5a621bd-d109-41ae-a381-a42414397802 | compute-0 | ACTIVE | ctlplane=192.168.24.31 |
| overcloud-full | compute |
| 496ab196-d6cb-447d-a118-5bafc5166cf2 | controller-0 | ACTIVE | ctlplane=192.168.24.37 |
| overcloud-full | controller |
| c01e730d-62f2-426a-a964-b31448f250b3 | controller-2 | ACTIVE | ctlplane=192.168.24.55 |
| overcloud-full | controller |
| 36df59b3-66f3-452e-9aec-b7e7f7c54b86 | controller-1 | ACTIVE | ctlplane=192.168.24.39 |
| overcloud-full | controller |
| f8f00497-246d-4e40-8a6a-b5a60fa66483 | compute-1 | ACTIVE | ctlplane=192.168.24.10 |
| overcloud-full | compute |
```

3. SSH を使用して、任意のコンピュータノードにログインします。

```
(undercloud) [stack@undercloud-0 ~]$ ssh heat-admin@192.168.24.31
```

4. root ユーザーに変更します。

```
[heat-admin@compute-0 ~]$ sudo su -
```

5. `/etc/ceph/ceph.conf` ファイルおよび `/etc/ceph/ceph.client.openstack.keyring` ファイルが存在することを確認します。

```
[root@compute-0 ~]# ls -l /etc/ceph/ceph.conf
-rw-r--r--. 1 root root 1170 Sep 29 23:25 /etc/ceph/ceph.conf
[root@compute-0 ~]# ls -l /etc/ceph/ceph.client.openstack.keyring
-rw-----. 1 ceph ceph 253 Sep 29 23:25 /etc/ceph/ceph.client.openstack.keyring
```

6. 以下のコマンドを入力して、`nova_compute` コンテナが `rbd` コマンドを使用して適切なプールの内容を一覧表示するように強制します。

```
# podman exec nova_compute /usr/bin/rbd --conf /etc/ceph/ceph.conf --keyring
/etc/ceph/ceph.client.openstack.keyring --cluster ceph --id openstack ls vms
```

プール名は、Ceph Storage クラスタの設定時に作成したイメージ、仮想マシン、およびボリュームのプール名と一致している必要があります。詳しくは、「[既存の Ceph Storage クラスタの設定](#)」を参照してください。イメージ、コンピュータインスタンス、およびボリュームの ID は、「[ID の収集](#)」で記録した ID と一致する必要があります。



#### 注記

`ceph-common` パッケージで提供される `/usr/bin/rbd` はデフォルトではオーバークラウドノードにインストールされないため、コマンドの例には `podman exec nova_compute` というプレフィックスが付けられています。ただし、これは `nova_compute` コンテナで利用できます。このコマンドは、ブロックデバイスイメージを一覧表示します。詳細は、『[Ceph Storage ブロックデバイスガイド](#)』の「[ブロックデバイスイメージの一覧表示](#)」を参照してください。

以下の例は、「[ID の収集](#)」で取得した ID を使用して、各プールに各サービスの ID が存在するかどうかを確認する方法を示しています。

```
# podman exec nova_compute /usr/bin/rbd --conf /etc/ceph/ceph.conf --keyring
/etc/ceph/ceph.client.openstack.keyring --cluster ceph --id openstack ls images | grep
4485d4c0-24c3-42ec-a158-4d3950fa020b

# podman exec nova_compute /usr/bin/rbd --conf /etc/ceph/ceph.conf --keyring
/etc/ceph/ceph.client.openstack.keyring --cluster ceph --id openstack ls vms | grep
64bcb731-e7a4-4dd5-a807-ee26c669482f

# podman exec nova_compute /usr/bin/rbd --conf /etc/ceph/ceph.conf --keyring
/etc/ceph/ceph.client.openstack.keyring --cluster ceph --id openstack ls volumes | grep
aeac15e8-b67f-454f-9486-46b3d75daff4
```

### 4.3. 検証の失敗のトラブルシューティング

検証手順が失敗した場合には、`openstack.client` ユーザーの Ceph キーおよび Ceph Storage モニター IP またはホスト名と一緒に使用し、Red Hat OpenStack Platform (RHOSP) 用に作成した Ceph Storage プールから読み取り、書き込み、および削除を行うことができることを確認します。

#### 手順

1. この手順で実行する必要のある入力の量を短くするには、コンピュータノードにログインして `rbd` コマンドのエイリアスを作成します。

```
# alias rbd="podman exec nova_compute /usr/bin/rbd --conf /etc/ceph/ceph.conf --keyring
/etc/ceph/ceph.client.openstack.keyring --cluster ceph --id openstack"
```

2. テストデータを新規オブジェクトとしてプールに書き込むことができることを確認します。

```
# rbd create --size 1024 vms/foo
```

3. テストデータが表示されることを確認します。

```
# rbd ls vms | grep foo
```

4. テストデータを削除します。

```
# rbd rm vms/foo
```



## 注記

この手順が失敗した場合は、**Ceph Storage** の管理者にお問い合わせください。この手順は成功するが、コンピュートインスタンス、**glance** イメージ、または **cinder** ボリュームを作成できない場合は、**Red Hat** サポートにお問い合わせください。

## 第5章 オーバークラウドへのアクセス

**director** は、アンダークラウドからオーバークラウドと対話するための設定および認証を行うスクリプトを生成します。**director** は、このファイル **overcloudrc** を **stack** ユーザーのホームディレクトリーに保存します。

### 手順

1. 以下のコマンドを入力して **overcloudrc** ファイルを使用します。

```
$ source ~/overcloudrc
```

これにより、アンダークラウド CLI からオーバークラウドと対話するのに必要な環境変数が読み込まれます。

2. アンダークラウドに戻るには、以下のコマンドを入力します。

```
$ source ~/stackrc
```