



Red Hat OpenStack Platform 17.0

**director と共に Red Hat Ceph Storage および
Red Hat OpenStack Platform をデプロイする**

Red Hat Ceph Storage クラスターをデプロイして使用するための director の設定

Red Hat OpenStack Platform 17.0 director と共に Red Hat Ceph Storage および Red Hat OpenStack Platform をデプロイする

Red Hat Ceph Storage クラスターをデプロイして使用するための director の設定

OpenStack Team
rhos-docs@redhat.com

法律上の通知

Copyright © 2023 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

本ガイドでは、Red Hat OpenStack Platform director を使用して、Red Hat Ceph Storage クラスターを持つオーバークラウドを作成する方法について説明します。これには、director を介して Red Hat Ceph Storage クラスターをカスタマイズするための手順が含まれています。

目次

多様性を受け入れるオープンソースの強化	5
RED HAT ドキュメントへのフィードバック (英語のみ)	6
第1章 オーバークラウドと RED HAT CEPH STORAGE のデプロイ	7
1.1. RED HAT CEPH STORAGE クラスター	7
1.2. RED HAT CEPH STORAGE ノードの要件	7
1.3. CEPH STORAGE ノードと RHEL の互換性	7
1.4. RED HAT CEPH STORAGE の導入	7
1.5. RED HAT CEPH STORAGE のデプロイメント要件	8
1.6. デプロイ後の検証	9
第2章 デプロイメント用の CEPH STORAGE ノードの準備	10
2.1. CEPH STORAGE ノードのディスクのクリーニング	10
2.2. ノードの登録	10
2.3. 利用可能な RED HAT CEPH STORAGE パッケージの確認	13
2.4. マルチディスク CEPH クラスターのルートディスクの定義	13
2.5. OVERCLOUD-MINIMAL イメージの使用による RED HAT サブスクリプションエンタイトルメントの使用回避	16
2.6. RED HAT CEPH STORAGE のノードの指定	17
第3章 RED HAT CEPH STORAGE クラスターの設定	20
3.1. OPENSTACK OVERCLOUD CEPH DEPLOY コマンド	20
3.2. CEPH 設定ファイル	20
3.3. 時刻同期の設定	21
3.4. RED HAT CEPH STORAGE クラスター名の設定	22
3.5. ネットワークデータファイルを使用したネットワークオプションの設定	23
3.6. 設定ファイルを使用したネットワークオプションの設定	24
3.7. OSD の CRUSH 階層の設定	25
3.8. CEPH サービス配置オプションの設定	26
3.9. CEPH ノードの SSH ユーザーオプションの設定	27
3.10. CEPH STORAGE コンテナへのアクセス	29
第4章 RED HAT CEPH STORAGE クラスターのカスタマイズ	31
4.1. 設定オプション	31
4.2. サービス仕様の生成 (オプション)	31
4.3. RED HAT OPENSTACK PLATFORM と RED HAT CEPH STORAGE の CEPH コンテナ	32
4.4. 高度な OSD 仕様の設定	32
4.5. ノード固有のオーバーライドからの移行	33
4.6. CEPH 伝送時暗号化の有効化	33
第5章 ストレージサービスのカスタマイズ	34
5.1. カスタム環境ファイルの設定	34
5.2. RED HAT CEPH STORAGE 配置グループ	34
5.3. CEPH METADATA SERVER の有効化	36
5.4. CEPH OBJECT GATEWAY オブジェクトストレージ	36
5.5. RED HAT OPENSTACK PLATFORM オブジェクトストレージのデプロイメントオプション	37
5.6. CEPH を使用するための BLOCK STORAGE BACKUP SERVICE の設定	38
5.7. CEPH ノード向けの複数のボンディングされたインターフェイスの設定	38
第6章 ネイティブ CEPHFS を使用する SHARED FILE SYSTEMS サービスのデプロイ	39
6.1. ネイティブドライバを使用する CEPHFS	39
6.2. ネイティブ CEPHFS バックエンドセキュリティー	40

6.3. ネイティブ CEPHFS デプロイメント	41
6.4. 要件	42
6.5. ファイル共有	42
6.6. ネイティブ CEPHFS 分離ネットワーク	43
6.7. ネイティブ CEPHFS 環境のデプロイ	43
6.8. NATIVE CEPHFS バックエンド環境ファイル	44
第7章 CEPHFS-NFS を使用した SHARED FILE SYSTEMS サービスのデプロイ	46
7.1. NFS ドライバーを使用した CEPHFS	46
7.2. CEPH サービスとクライアントアクセス	47
7.3. NFS バックエンドに CEPHFS を使用した SHARED FILE SYSTEMS サービスの耐障害性	48
7.4. NFS-GANESHA インストールによる CEPHFS	48
7.5. NFS-GANESHA のインストール要件による CEPHFS	49
7.6. ファイル共有	50
7.7. カスタムロールファイルの生成	50
7.8. 更新された環境のデプロイ	51
第8章 オーバークラウドデプロイメントの開始	55
8.1. オーバークラウドデプロイメントの開始	55
第9章 DIRECTOR を使用した多様なワークロードのパフォーマンス層の定義	57
9.1. パフォーマンス層の設定	57
9.2. CRUSH ルールとプールの確認	60
第10章 RED HAT CEPH STORAGE DASHBOARD のオーバークラウドデプロイメントへの追加	62
10.1. CEPH DASHBOARD における TLS EVERYWHERE	63
10.2. CEPH DASHBOARD に必要なコンテナの追加	64
10.3. CEPH DASHBOARD のデプロイ	65
10.4. コンポーザブルネットワークを使用した CEPH DASHBOARD のデプロイ	66
10.5. デフォルト権限の変更	67
10.6. CEPH DASHBOARD へのアクセス	67
第11章 RED HAT CEPH STORAGE クラスターを管理するためのデプロイメント後の操作	69
11.1. 設定オーバーライドの無効化	69
11.2. オーバークラウドへのアクセス	69
11.3. RED HAT CEPH STORAGE ノードのモニタリング	69
11.4. BLOCK STORAGE (CINDER) 種別の新しい CEPH プールへのマッピング	70
第12章 ネイティブ CEPHFS デプロイ後の設定と検証	72
12.1. ストレージプロバイダーネットワークの作成	72
12.2. ストレージプロバイダーネットワークの設定	73
12.3. ストレージプロバイダーネットワークにおけるロールベースアクセス制御の設定	73
12.4. デフォルトのファイル共有種別の設定	74
12.5. 分離ストレージネットワークが作成されていることの確認	75
12.6. CEPH MDS サービスの確認	75
12.7. CEPH クラスターのステータスの確認	76
12.8. MANILA-SHARE サービスのステータスの確認	77
12.9. MANILA-API サービスがスケジューラーおよびファイル共有サービスを認識していることの確認	77
第13章 CEPHFS NFS デプロイ後の設定と検証	78
13.1. ストレージプロバイダーネットワークの作成	78
13.2. 共有プロバイダー STORAGENFS ネットワークの設定	79
13.3. デフォルトのファイル共有種別の設定	80
13.4. STORAGENFS 分離ネットワークが作成されていることの確認	80
13.5. CEPH MDS サービスの確認	80

13.6. CEPH クラスターのステータスの確認	81
13.7. NFS-GANESHA および MANILA-SHARE サービスのステータスの確認	82
13.8. MANILA-API サービスがスケジューラーおよびファイル共有サービスを認識していることの確認	82
第14章 環境のリブート	83
14.1. CEPH STORAGE (OSD) クラスターのリブート	83
14.2. CEPH STORAGE OSD の再起動による CEPH MONITOR サービスへの接続の有効化	84
第15章 CEPH STORAGE クラスターのスケーリング	85
15.1. CEPH STORAGE クラスターのスケールアップ	85
15.2. CEPH STORAGE ノードのスケールダウンと置き換え	86
第16章 障害のあるディスクの置き換え	90
16.1. ディスクの交換	90

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、[Red Hat CTO である Chris Wright のメッセージ](#) をご覧ください。

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat 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問題の詳細を確認する際に使用できるメールアドレスを記入してください。
7. **Submit** をクリックします。

第1章 オーバークラウドと RED HAT CEPH STORAGE のデプロイ

Red Hat OpenStack Platform (RHOSP) director は、オーバークラウドとも呼ばれるクラウド環境と Red Hat Ceph Storage をデプロイします。Director は、**tripleo-ansible** パッケージを通じて提供される Ansible Playbook を使用して、Ceph Storage クラスターをデプロイします。director は、Ceph Storage クラスターの設定およびスケーリング操作も管理します。

Red Hat Ceph Storage に関する詳細は、[Red Hat Ceph Storage Architecture Guide](#) を参照してください。

Red Hat OpenStack Platform のサービスに関する詳細は、[Director Installation and Usage の Configuring a basic overcloud with the CLI tools](#) を参照してください。

1.1. RED HAT CEPH STORAGE クラスター

Red Hat Ceph Storage は、パフォーマンス、信頼性、およびスケーラビリティのために設計された分散データオブジェクトストアです。分散オブジェクトストアは、非構造化データを使用して、最新のオブジェクトインターフェイスと従来のオブジェクトインターフェイスを同時に処理します。

Ceph Storage はクラスターとしてデプロイされます。Ceph Storage クラスターは、次の2つの主要なタイプのデーモンで設定されます。

- Ceph Object Storage Daemon (CephOSD) - CephOSD は、データストレージ、データレプリケーション、リバランス、リカバリー、モニタリング、およびレポートのタスクを実行します。
- Ceph Monitor (CephMon) - CephMon は、クラスターの現在の状態でクラスターマップのプライマリーコピーを維持します。

Red Hat Ceph Storage に関する詳細は、[Red Hat Ceph Storage アーキテクチャーガイド](#) を参照してください。

1.2. RED HAT CEPH STORAGE ノードの要件

director を使用して Ceph Storage クラスターを作成するには、追加のノード要件があります。

- プロセッサー、メモリー、ネットワークインターフェイスカードの選択、ディスクレイアウトなどのハードウェア要件は、[Red Hat Ceph Storage Hardware Guide](#) で確認できます。
- 各 Ceph Storage ノードにも、Intelligent Platform Management Interface (IPMI) 機能などのサポート対象の電源管理インターフェイスがサーバーのマザーボードに搭載されている必要があります。
- 各 Ceph Storage ノードには、少なくとも2つのディスクが必要です。RHOSP director は **cephadm** を使用して Ceph Storage クラスターをデプロイします。cephadm 機能は、ノードのルートディスクへの Ceph OSD のインストールをサポートしていません。

1.3. CEPH STORAGE ノードと RHEL の互換性

RHOSP 17.0 は RHEL 9.0 でサポートされています。ただし、Ceph Storage ロールにマップされているホストは、最新のメジャー RHEL リリースに更新されます。アップグレードする前に、Red Hat ナレッジベースの記事 [Red Hat Ceph Storage: Supported configurations](#) を確認してください。

1.4. RED HAT CEPH STORAGE の導入

以下の 2 つのフェーズで Red Hat Ceph Storage をデプロイします。

- オーバークラウドをデプロイする前に、Red Hat Ceph Storage クラスターを作成します。
- オーバークラウドのデプロイメント中に Red Hat Ceph Storage クラスターを設定します。

Ceph RADOS Block Device (RBD) サービスを提供する準備が整った Ceph Storage クラスターが作成されます。さらに、次のサービスが適切なノードで実行されています。

- Ceph モニター (CephMon)
- Ceph マネージャー (CephMgr)
- Ceph OSD (CephOSD)

プールと `cephx` キーは、設定フェーズで作成されます。

以下の Ceph Storage コンポーネントは、設定フェーズが完了するまで使用できません。

- Ceph ダッシュボード (CephDashboard)
- Ceph オブジェクトゲートウェイ (CephRGW)
- Ceph MDS (CephMds)

Red Hat Ceph Storage クラスター設定は、オーバークラウドのデプロイ中に最終決定されます。Ceph Object Gateway や Ceph Dashboard などのデーモンおよびサービスは、オーバークラウドの定義に従ってデプロイされます。Red Hat OpenStack Platform (RHOSP) サービスは、Ceph Storage クラスタークライアントとして設定されます。

1.5. RED HAT CEPH STORAGE のデプロイメント要件

Ceph Storage クラスターを作成する前に、ネットワークリソースとベアメタルインスタンスのプロビジョニングが必要です。Red Hat Ceph Storage クラスターを作成する前に、以下を設定します。

- **openstack overcloud network provision** コマンドと **cli-overcloud-network-provision.yaml** ansible playbook を使用してネットワークをプロビジョニングします。
- **openstack overcloud node provision** コマンドでベアメタルインスタンスをプロビジョニングし、**cli-overcloud-node-provision.yaml** ansible playbook を使用してベアメタルインスタンスをプロビジョニングします。

これらのタスクの詳細については、次を参照してください。

- [ネットワークガイド](#)
- [Bare Metal Provisioning](#)

Ceph Storage クラスター設定を完了するには、以下の要素がオーバークラウド環境に存在する必要があります。

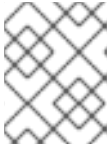
- アンダークラウドホストにインストールされた Red Hat OpenStack Platform director。Director Installation and Usage の Installing director を参照してください。
- Red Hat Ceph Storage をサポートするための推奨ハードウェアのインストール。推奨されるハードウェアの詳細は、Red Hat Ceph Storage Hardware Guide を参照してください。

1.6. デプロイ後の検証

Director は、**cephadm** コマンドによって実行される **tripleo-ansible** ロールを使用して、Ceph RADOS Block Device (RBD) を提供する準備ができている Ceph Storage クラスターをデプロイします。

cephadm が Ceph Storage デプロイメントを完了した後、以下が整っていることを確認します。

- **sudo cephadm shell** コマンドを使用するための CephMon サービスノードへの SSH アクセス。
- すべての OSD が動作しています。



注記

動作していない OSD をチェックして、クリーンアップされていないディスクなどの環境問題を確認します。

- CephMon サービスノードの **/etc/ceph** ディレクトリーにある Ceph 設定ファイルとクライアント管理キーリングファイル。
- Ceph Storage クラスターは、RBD を提供する準備ができています。

プール、cephx キー、CephDashboard、および CephRGW は、**openstack overcloud deploy** command によるオーバークラウドのデプロイメント中に設定されます。これには2つの理由があります。

- Dashboard および RGW サービスは、**haproxy** と統合する必要があります。これは、オーバークラウドとともにデプロイされます。
- プールと cephx キーの作成は、デプロイされている OpenStack クライアントによって異なります。

これらのリソースは、クライアント管理キーリングファイルと、**openstack overcloud ceph deploy** コマンドによって出力される **~/deployed_ceph.yaml** ファイルを使用して、Ceph Storage クラスターに作成されます。

cephadm の詳細については、[Red Hat Ceph Storage Installation Guide](#) を参照してください。

第2章 デプロイメント用の CEPH STORAGE ノードの準備

Red Hat Ceph Storage ノードは、IPMI 電源管理を備えたベアメタルシステムです。director は、各ノードに Red Hat Enterprise Linux をインストールします。

director は、イントロスペクションおよびプロビジョニングプロセス中に、プロビジョニングネットワークで各ノードと通信します。すべてのノードは、ネイティブ VLAN でプロビジョニングネットワークに接続します。

オーバークラウドをデプロイする前のベアメタルのプロビジョニングに関する詳細は、**director のインストールと使用** ガイドの [オーバークラウドのプロビジョニングとデプロイ](#) を参照してください。

ベアメタルプロビジョニングの完全なガイドについては、[ベアメタルプロビジョニング](#) を参照してください。

2.1. CEPH STORAGE ノードのディスクのクリーニング

Ceph Storage OSD とジャーナルパーティションには、ファクトリークリーンディスクが必要です。Ceph OSD サービスをインストールする前に、Bare Metal Provisioning サービス (ironic) によって、これらのディスクからすべてのデータとメタデータを消去する必要があります。

Bare Metal Provisioning サービスを使用して、デフォルトですべてのディスクデータとメタデータを削除するように、director を設定できます。director がこのタスクを実行するように設定されている場合、Bare Metal Provisioning サービスは、ノードが **available** に設定されるたびに、ノードを起動する追加の手順を実行します。



警告

Bare Metal Provisioning サービスは、**wipefs --force --all** コマンドを使用します。このコマンドは、ディスク上のすべてのデータとメタデータを削除しますが、セキュアな消去は実行しません。Secure Erase には非常に長い時間がかかります。

手順

1. **/home/stack/undercloud.conf** を開き、次のパラメーターを追加します。

```
clean_nodes=true
```

2. **/home/stack/undercloud.conf** を保存します。
3. アンダークラウド設定を更新します。

```
openstack undercloud install
```

2.2. ノードの登録

director と通信できるように、ノードを登録します。

手順

1. **/home/stack** にノードインベントリー JSON ファイルを作成します。
2. ノードごとのハードウェアおよび電源管理の詳細を入力します。
以下に例を示します。

```
{
  "nodes":[
    {
      "mac":[
        "b1:b1:b1:b1:b1:b1"
      ],
      "cpu":"4",
      "memory":"6144",
      "disk":"40",
      "arch":"x86_64",
      "pm_type":"ipmi",
      "pm_user":"admin",
      "pm_password":"p@55w0rd!",
      "pm_addr":"192.0.2.205"
    },
    {
      "mac":[
        "b2:b2:b2:b2:b2:b2"
      ],
      "cpu":"4",
      "memory":"6144",
      "disk":"40",
      "arch":"x86_64",
      "pm_type":"ipmi",
      "pm_user":"admin",
      "pm_password":"p@55w0rd!",
      "pm_addr":"192.0.2.206"
    },
    {
      "mac":[
        "b3:b3:b3:b3:b3:b3"
      ],
      "cpu":"4",
      "memory":"6144",
      "disk":"40",
      "arch":"x86_64",
      "pm_type":"ipmi",
      "pm_user":"admin",
      "pm_password":"p@55w0rd!",
      "pm_addr":"192.0.2.207"
    },
    {
      "mac":[
        "c1:c1:c1:c1:c1:c1"
      ],
      "cpu":"4",
      "memory":"6144",
      "disk":"40",
      "arch":"x86_64",
      "pm_type":"ipmi",
      "pm_user":"admin",
```

```

    "pm_password":"p@55w0rd!",
    "pm_addr":"192.0.2.208"
  },
  {
    "mac":[
      "c2:c2:c2:c2:c2:c2"
    ],
    "cpu":"4",
    "memory":"6144",
    "disk":"40",
    "arch":"x86_64",
    "pm_type":"ipmi",
    "pm_user":"admin",
    "pm_password":"p@55w0rd!",
    "pm_addr":"192.0.2.209"
  },
  {
    "mac":[
      "c3:c3:c3:c3:c3:c3"
    ],
    "cpu":"4",
    "memory":"6144",
    "disk":"40",
    "arch":"x86_64",
    "pm_type":"ipmi",
    "pm_user":"admin",
    "pm_password":"p@55w0rd!",
    "pm_addr":"192.0.2.210"
  },
  {
    "mac":[
      "d1:d1:d1:d1:d1:d1"
    ],
    "cpu":"4",
    "memory":"6144",
    "disk":"40",
    "arch":"x86_64",
    "pm_type":"ipmi",
    "pm_user":"admin",
    "pm_password":"p@55w0rd!",
    "pm_addr":"192.0.2.211"
  },
  {
    "mac":[
      "d2:d2:d2:d2:d2:d2"
    ],
    "cpu":"4",
    "memory":"6144",
    "disk":"40",
    "arch":"x86_64",
    "pm_type":"ipmi",
    "pm_user":"admin",
    "pm_password":"p@55w0rd!",
    "pm_addr":"192.0.2.212"
  },
  {

```

```

        "mac":[
            "d3:d3:d3:d3:d3:d3"
        ],
        "cpu":"4",
        "memory":"6144",
        "disk":"40",
        "arch":"x86_64",
        "pm_type":"ipmi",
        "pm_user":"admin",
        "pm_password":"p@55w0rd!",
        "pm_addr":"192.0.2.213"
    }
}
}

```

3. 新しいファイルを保存します。
4. スタックユーザーを初期化します。

```
$ source ~/stackrc
```

5. JSON インベントリーファイルを director にインポートし、ノードを登録する

```
$ openstack overcloud node import <inventory_file>
```

<inventory_file> を最初の手順で作成したファイルの名前に置き換えます。

6. 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>
```

2.3. 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の失敗を回避するために、必要なすべてのパッケージが利用可能であることを確認します。

2.3.1. cephadm パッケージのインストールの確認

cephadm パッケージが1つ以上のオーバークラウドノードにインストールされていることを確認します。**cephadm** パッケージは、Ceph Storage クラスターの最初のノードをブートストラップするために、使用されます。

cephadm パッケージは **overcloud-hardened-uefi-full.qcow2** イメージに含まれています。**tripleo_cephadm** ロールは、Ansible パッケージモジュールを使用して、イメージ内に存在することを確認します。

2.4. マルチディスク CEPH クラスターのルートディスクの定義

Ceph Storage ノードは通常、複数のディスクを使用します。Director は、複数のディスク設定でルートディスクを識別する必要があります。オーバークラウドイメージは、プロビジョニングプロセス中にルートディスクに書き込まれます。

ハードウェアプロパティは、ルートディスクを識別するために使用されます。ルートディスクの識別に使用できるプロパティの詳細については、「[ルートディスクを識別するプロパティ](#)」を参照してください。

手順

1. 各ノードのハードウェアイントロスペクションからのディスク情報を確認します。

```
(undercloud)$ openstack baremetal introspection data save <node_uuid> | --file
<output_file_name>
```

- **<node_uuid>** をノードの UUID に置き換えます。
- **<output_file_name>** を、ノードイントロスペクションの出力を含むファイルの名前に置き換えます。
たとえば、1つのノードのデータで3つのディスクが表示される場合があります。

```
[
  {
    "size": 299439751168,
    "rotational": true,
    "vendor": "DELL",
    "name": "/dev/sda",
    "wwn_vendor_extension": "0x1ea4dcc412a9632b",
    "wwn_with_extension": "0x61866da04f3807001ea4dcc412a9632b",
    "model": "PERC H330 Mini",
    "wwn": "0x61866da04f380700",
    "serial": "61866da04f3807001ea4dcc412a9632b"
  }
  {
    "size": 299439751168,
    "rotational": true,
    "vendor": "DELL",
    "name": "/dev/sdb",
    "wwn_vendor_extension": "0x1ea4e13c12e36ad6",
    "wwn_with_extension": "0x61866da04f380d001ea4e13c12e36ad6",
    "model": "PERC H330 Mini",
    "wwn": "0x61866da04f380d00",
    "serial": "61866da04f380d001ea4e13c12e36ad6"
  }
  {
    "size": 299439751168,
    "rotational": true,
    "vendor": "DELL",
    "name": "/dev/sdc",
    "wwn_vendor_extension": "0x1ea4e31e121cfb45",
    "wwn_with_extension": "0x61866da04f37fc001ea4e31e121cfb45",
    "model": "PERC H330 Mini",
    "wwn": "0x61866da04f37fc00",
    "serial": "61866da04f37fc001ea4e31e121cfb45"
  }
]
```

2. 一意のハードウェアプロパティを使用して、ノードのルートディスクを設定します。

```
(undercloud)$ openstack baremetal node set --property root_device='{<property_value>}'
<node-uuid>
```

- **<property_value>** を、ルートディスクの設定に使用するイントロスペクションデータからの一意のハードウェアプロパティ値に置き換えます。
- **<node_uuid>** をノードの UUID に置き換えます。



注記

一意のハードウェアプロパティは、ディスクを一意に識別するハードウェアイントロスペクションステップからの任意のプロパティです。たとえば、次のコマンドは、ディスクのシリアル番号を使用してルートディスクを設定します。

```
(undercloud)$ openstack baremetal node set --property
root_device='{ "serial": "61866da04f380d001ea4e13c12e36ad6" }'
1a4e30da-b6dc-499d-ba87-0bd8a3819bc0
```

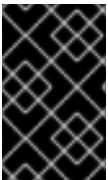
3. 最初にネットワークから起動し、次にルートディスクから起動するように、各ノードの BIOS を設定します。

director は、ルートディスクとして使用する特定のディスクを把握します。**openstack overcloud node provision** コマンドを実行すると、director はオーバークラウドをプロビジョニングし、ルートディスクにオーバークラウドのイメージを書き込みます。

2.4.1. ルートディスクを識別するプロパティ

以下の属性を定義すると、director がルートディスクを特定するのに役立ちます。

- **model** (文字列): デバイスの ID
- **vendor** (文字列): デバイスのベンダー
- **serial** (文字列): ディスクのシリアル番号
- **hctl** (文字列): SCSI のホスト、チャンネル、ターゲット、Lun
- **size** (整数): デバイスのサイズ (GB 単位)
- **wwn** (文字列): 一意のストレージ ID
- **wwn_with_extension** (文字列): ベンダー拡張子を追加した一意のストレージ ID
- **wwn_vendor_extension** (文字列): 一意のベンダーストレージ ID
- **rotational** (ブール値): 回転式デバイス (HDD) には true、そうでない場合 (SSD) には false
- **name** (文字列): デバイス名 (例: /dev/sdb1)



重要

永続的な名前を持つデバイスには **name** プロパティを使用します。ノードの起動時に値が変更される可能性があるため、**name** プロパティを使用して永続的な名前を持たないデバイスのルートディスクを設定しないでください。

2.5. OVERCLOUD-MINIMAL イメージの使用による RED HAT サブスクリプションエンタイトルメントの使用回避

Red Hat OpenStack Platform (RHOSP) デプロイメントのデフォルトイメージは **overcloud-hardened-uefi-full.qcow2** です。**overcloud-hardened-uefi-full.qcow2** イメージは、有効な Red Hat OpenStack Platform (RHOSP) サブスクリプションを使用します。サブスクリプションのエンタイトルメントを消費したくない場合は **overcloud-minimal** イメージを使用して、有償の Red Hat サブスクリプションが限度に達するのを回避できます。これは、たとえば Ceph デーモンのみでノードをプロビジョニングする場合や、他の OpenStack サービスを実行したくないベアオペレーティングシステム (OS) をプロビジョニングする場合に役立ちます。**overcloud-minimal** イメージの取得方法についての情報は、[Obtaining images for overcloud nodes](#) を参照してください。



注記

overcloud-minimal イメージでは、標準の Linux ブリッジのみサポートされます。**overcloud-minimal** イメージでは、Open vSwitch (OVS) はサポートされません。OVS は、Red Hat OpenStack Platform サブスクリプションエンタイトルメントを必要とする OpenStack サービスだからです。Ceph Storage ノードをデプロイするのに OVS は必要ありません。**ovs_bond** を使用してボンディングを定義する代わりに、**linux_bond** を使用します。

手順

1. `/home/stack/templates/overcloud-baremetal-deploy.yaml` ファイルを開きます。
2. **overcloud-minimal** イメージを使用するノードの **image** プロパティを追加または更新します。特定のノードまたはロールのすべてのノードでイメージを **overcloud-minimal** に設定できます。



注記

オーバークラウドの最小イメージは、ディスクイメージ全体ではありません。カーネルと RAM ディスクは `/home/stack/templates/overcloud-baremetal-deploy.yaml` ファイルで指定する必要があります。

特定のノード

```
- name: Ceph
  count: 3
  instances:
    - hostname: overcloud-ceph-0
      name: node00
      image:
        href: file:///var/lib/ironic/images/overcloud-minimal.raw
        kernel: file:///var/lib/ironic/images/overcloud-minimal.vmlinuz
        ramdisk: file:///var/lib/ironic/images/overcloud-minimal.initrd
    - hostname: overcloud-ceph-1
      name: node01
      image:
        href: file:///var/lib/ironic/images/overcloud-minimal.raw
        kernel: file:///var/lib/ironic/images/overcloud-minimal.vmlinuz
        ramdisk: file:///var/lib/ironic/images/overcloud-minimal.initrd
    - hostname: overcloud-ceph-2
      name: node02
```

```
image:
  href: file:///var/lib/ironic/images/overcloud-minimal.raw
  kernel: file:///var/lib/ironic/images/overcloud-minimal.vmlinuz
  ramdisk: file:///var/lib/ironic/images/overcloud-minimal.initrd
```

特定のロールのすべてのノード

```
- name: Ceph
  count: 3
  defaults:
    image:
      href: file:///var/lib/ironic/images/overcloud-minimal.raw
      kernel: file:///var/lib/ironic/images/overcloud-minimal.vmlinuz
      ramdisk: file:///var/lib/ironic/images/overcloud-minimal.initrd
  instances:
    - hostname: overcloud-ceph-0
      name: node00
    - hostname: overcloud-ceph-1
      name: node01
    - hostname: overcloud-ceph-2
      name: node02
```

3. **roles_data.yaml** ロール定義ファイルで、**rhsm_enforce** パラメーターを **False** に設定します。

```
rhsm_enforce: False
```

4. プロビジョニングコマンドを実行します。

```
(undercloud)$ openstack overcloud node provision \
--stack overcloud \
--output /home/stack/templates/overcloud-baremetal-deployed.yaml \
/home/stack/templates/overcloud-baremetal-deploy.yaml
```

5. **overcloud-baremetal-deployed.yaml** 環境ファイルを **openstack overcloud ceph deploy** コマンドに渡します。

2.6. RED HAT CEPH STORAGE のノードの指定

Red Hat Ceph Storage のノードを指定するには、新しいロールファイルを作成して **CephStorage** ロールを設定し、ベアメタルノードを **CephStorage** のリソースクラスで設定する必要があります。

手順

1. アンダークラウドに **stack** ユーザーとしてログインします。
2. **stackrc** ファイルを取得します。

```
[stack@director ~]$ source ~/stackrc
```

3. **Controller**、**Compute**、および **CephStorage** ロールを含む、**roles_data.yaml** という名前の新しいロールデータファイルを生成します。

```
(undercloud)$ openstack overcloud roles \
generate Controller Compute CephStorage -o /home/stack/templates/roles_data.yaml \
```

4. **roles_data.yaml** を開き、次のパラメーターとセクションがあることを確認します。

セクション/パラメーター	値
ロールのコメント	Role: CephStorage
ロール名	name: CephStorage
description	Ceph node role
HostnameFormatDefault	%stackname%-novaceph-%index%
deprecated_nic_config_name	ceph.yaml

5. ノード定義テンプレートに追加して、オーバークラウドの Ceph ノードを登録します。
6. ノードのハードウェアを検査します。

```
(undercloud)$ openstack overcloud node introspect --all-manageable --provide
```

7. カスタム Ceph リソースクラスを使用して、Ceph 用に指定する各ベアメタルノードにタグを付けます。

```
(undercloud)$ openstack baremetal node set \
--resource-class baremetal.CEPH <node>
```

<node> をベアメタルノードの ID に置き換えてください。

8. **CephStorage** ロールを **overcloud-baremetal-deploy.yaml** ファイルに追加し、ノードに割り当てる予測可能なノード配置、リソースクラス、またはその他の属性を定義します。

```
- name: Controller
  count: 3
- name: Compute
  count: 3
- name: CephStorage
  count: 5
  defaults:
    resource_class: baremetal.CEPH
```

9. プロビジョニングコマンドを実行します。

```
(undercloud)$ openstack overcloud node provision \
--stack stack \
--output /home/stack/templates/overcloud-baremetal-deployed.yaml \
/home/stack/templates/overcloud-baremetal-deploy.yaml
```

10. 別のターミナルでプロビジョニングの進捗をモニターリングします。プロビジョニングが成功すると、ノードの状態が **available** から **active** に変わります。

```
(undercloud)$ watch openstack baremetal node list
```

関連情報

- ノード登録の詳細については、「[ノードの登録](#)」を参照してください。
- ノードハードウェアの検査の詳細については、**Director のインストールと使用** ガイドの [ベアメタルノードハードウェアのインベントリーの作成](#) を参照してください。

第3章 RED HAT CEPH STORAGE クラスターの設定

Red Hat OpenStack Platform 環境に Red Hat Ceph Storage クラスターをデプロイするには、最初に環境の Red Hat Ceph Storage クラスターオプションを設定する必要があります。

Red Hat Ceph Storage クラスターオプションを設定します。

- [時間同期の設定](#)
- [Red Hat Ceph Storage クラスター名の設定](#)
- [ネットワークデータファイルを使用したネットワークオプションの設定](#)
- [設定ファイルを使用したネットワークオプションの設定](#)
- [OSD の CRUSH 階層の設定](#)
- [Ceph サービス配置オプションの設定](#)
- [Ceph ノードの SSH ユーザーオプションの設定](#)
- [コンテナーレジストリーの設定](#)

前提条件

Red Hat Ceph Storage クラスターを設定してデプロイする前に、Bare Metal Provisioning サービス (ironic) を使用して、ベアメタルインスタンスとネットワークをプロビジョニングします。詳細は、[ベアメタルプロビジョニング](#) を参照してください。

3.1. OPENSTACK OVERCLOUD CEPH DEPLOY コマンド

director を使用して Ceph クラスターをデプロイする場合は、**openstack overcloud ceph deploy** コマンドを使用する必要があります。コマンドオプションおよびパラメーターの完全な一覧は、コマンドラインインターフェイスリファレンスの [openstack overcloud ceph deploy](#) を参照してください。

コマンド **openstack overcloud ceph deploy --help** は、環境で使用可能な現在のオプションとパラメーターを提供します。

3.2. CEPH 設定ファイル

標準形式の初期化ファイルは、Ceph クラスター設定を実行する1つの方法です。この初期化ファイルは、Ceph クラスターを設定するために使用されます。このファイルを使用するには、* **cephadm bootstrap --config <file_name>** * **openstack overcloud ceph deploy --config <file_name>** コマンドのいずれかを使用します。

例

以下の例では、**initial-ceph.conf** という単純な初期化ファイルを作成し、**openstack overcloud ceph deploy** コマンドを使用して Ceph クラスターを設定します。ネットワーク上を通過するすべてのデータを暗号化するセキュアモードを使用するようにメッセージャー v2 プロトコルを設定する方法を示します。

```
$ cat <<EOF > initial-ceph.conf
[global]
ms_cluster_mode: secure
ms_service_mode: secure
```

```
ms_client_mode: secure
EOF
$ openstack overcloud ceph deploy --config initial-ceph.conf ...
```

3.3. 時刻同期の設定

時刻同期サービス (chrony) は、デフォルトで時刻同期が有効になっています。以下のタスクを実行してサービスを設定できます。

- [区切りリストを使用した時刻同期の設定](#)
- [環境ファイルを使用した時刻同期の設定](#)
- [時刻同期の無効化](#)



注記

時刻同期は、区切り文字付きリストまたは環境ファイルを使用して設定されます。管理に最適な手順を使用してください。

3.3.1. 区切りリストを使用した時刻同期の設定

区切りリストを使用して NTP サーバーを設定するように時刻同期サービス (chrony) を設定できます。

手順

1. アンダークラウドノードに **stack** ユーザーとしてログインします。
2. 区切りリストを使用して NTP サーバーを設定します。

```
openstack overcloud ceph deploy \
  --ntp-server "<ntp_server_list>"
```

<ntp_server_list> をコンマ区切りのサーバーのリストに置き換えます。

```
openstack overcloud ceph deploy \
  --ntp-server "0.pool.ntp.org,1.pool.ntp.org"
```

3.3.2. 環境ファイルを使用した時刻同期の設定

NTP サーバーを定義する環境ファイルを使用するように時刻同期サービス (chrony) を設定できます。

手順

1. アンダークラウドノードに **stack** ユーザーとしてログインします。
2. **/home/stack/templates/ntp-parameters.yaml** などの環境ファイルを作成して、NTP サーバー設定を含めます。
3. **NtpServer** パラメーターを追加します。 **NtpServer** パラメーターには、NTP サーバーのコンマ区切りリストが含まれます。

```
parameter_defaults:
  NtpServer: 0.pool.ntp.org,1.pool.ntp.org
```

4. 環境ファイルを使用して NTP サーバーを設定します。

```
openstack overcloud ceph deploy \
  --ntp-heat-env-file "<ntp_file_name>"
```

<ntp_file_name> を、作成した環境ファイルの名前に置き換えます。

```
openstack overcloud ceph deploy \
  --ntp-heat-env-file "/home/stack/templates/ntp-parameters.yaml"
```

3.3.3. 時刻同期の無効化

時刻同期サービス (chrony) はデフォルトで有効になっています。サービスを使用したくない場合は、サービスを無効にすることができます。

手順

1. アンダークラウドノードに **stack** ユーザーとしてログインします。
2. 時刻同期サービス (chrony) を無効にします。

```
openstack overcloud ceph deploy \
  --skip-ntp
```

3.4. RED HAT CEPH STORAGE クラスター名の設定

設定した名前で Red Hat Ceph Storage クラスターをデプロイできます。デフォルト名は **ceph** です。

手順

1. アンダークラウドノードに **stack** ユーザーとしてログインします。
2. 次のコマンドを使用して、Ceph Storage クラスターの名前を設定します。

```
openstack overcloud ceph deploy \ --cluster <cluster_name>
```

```
$ openstack overcloud ceph deploy \ --cluster central \
```

注記

キーリングファイルは、現時点では作成されません。キーリングファイルは、オーバークラウドのデプロイメント中に作成されます。キーリングファイルは、この手順で設定されたクラスター名を継承します。オーバークラウドのデプロイメントの詳細は、「[オーバークラウドデプロイメントの開始](#)」を参照してください。

上記の例では、Ceph クラスターの名前は **central** です。**central** の Ceph クラスターの設定ファイルとキーリングファイルは、デプロイプロセス中に **/etc/ceph** に作成されます。

```
[root@oc0-controller-0 ~]# ls -l /etc/ceph/
total 16
-rw-----. 1 root root 63 Mar 26 21:49 central.client.admin.keyring
-rw-----. 1 167 167 201 Mar 26 22:17 central.client.openstack.keyring
-rw-----. 1 167 167 134 Mar 26 22:17 central.client.radosgw.keyring
-rw-r--r--. 1 root root 177 Mar 26 21:49 central.conf
```

トラブルシューティング

Ceph Storage クラスターのカスタム名を設定すると、次のエラーが表示される場合があります。

monclient: get_monmap_and_config cannot identify monitors to contact because

このエラーが表示された場合は、Ceph のデプロイ後に次のコマンドを使用します。

cephadm shell --config <configuration_file> --keyring <keyring_file>

たとえば、クラスター名を **central** に設定したときにこのエラーが表示された場合は、次のコマンドを使用します。

```
cephadm shell --config /etc/ceph/central.conf \
--keyring /etc/ceph/central.client.admin.keyring
```

代わりに次のコマンドを使用することもできます。

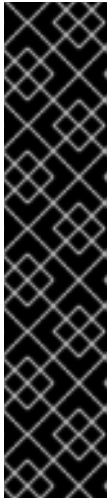
```
cephadm shell --mount /etc/ceph:/etc/ceph
export CEPH_ARGS='--cluster central'
```

3.5. ネットワークデータファイルを使用したネットワークオプションの設定

ネットワークデータファイルには、Red Hat Ceph Storage クラスターが使用するネットワークが記述されています。

手順

1. アンダークラウドノードに **stack** ユーザーとしてログインします。
2. **network_data.yaml** というカスタムネットワーク属性を定義する YAML 形式のファイルを作成します。



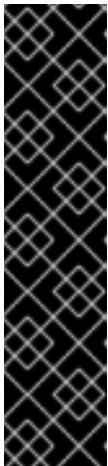
重要

ネットワーク分離を使用すると、標準のネットワークデプロイメントは、2つの Ceph ネットワークにマッピングされる2つのストレージネットワークで設定されます。

- ストレージネットワーク **storage** は、Ceph ネットワーク **public_network** にマップされます。このネットワークは、コンピュータノードから Ceph クラスタへの RBD トラフィックなどのストレージトラフィックを処理します。
- ストレージネットワーク **storage_mgmt** は、Ceph ネットワーク **cluster_network** にマップされます。このネットワークは、Ceph OSD 間のデータレプリケーションなどのストレージ管理トラフィックを処理します。

3. **openstack overcloud ceph deploy** コマンドと **--config** オプションを使用して、設定をデプロイします。

```
openstack overcloud ceph deploy \
  deployed_metal.yaml \
  -o deployed_ceph.yaml \
  --network-data network_data.yaml
```



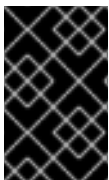
重要

openstack overcloud ceph deploy コマンドは、**--network-data** オプションで指定されたネットワークデータファイルを使用して、**public_network** および **cluster_network** として使用するネットワークを決定します。コマンドは、**--public-network-name** および **--cluster-network-name** オプションで別の名前が指定されていない限り、これらのネットワークがネットワークデータファイルで **storage** および **storage_mgmt** という名前であると想定します。

ネットワーク分離を使用してデプロイする場合は、**--network-data** オプションを使用する必要があります。このオプションを使用しない場合、デフォルトのアンダークラウド (192.168.24.0/24) が **public_network** と **cluster_network** の両方に使用されます。

3.6. 設定ファイルを使用したネットワークオプションの設定

ネットワークオプションは、ネットワークデータファイルの代わりに設定ファイルで指定できます。



重要

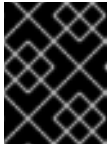
この方法を使用してネットワークオプションを設定すると、**network_data.yaml** で自動的に生成された値が上書きされます。このネットワーク設定方法を使用する場合は、必ず4つの値をすべて設定してください。

手順

1. アンダークラウドノードに **stack** ユーザーとしてログインします。
2. Ceph クラスタを設定する標準形式の初期化ファイルを作成します。他の設定オプションを含むファイルをすでに作成している場合は、それにネットワーク設定を追加できます。

3. ファイルの **[global]** セクションに次のパラメーターを追加します。

- **public_network**
- **cluster_network**
- **ms_bind_ipv4**



重要

public_network および **cluster_network** が **storage** および **storage_mgmt** と同じネットワークにマップされていることを確認します。

以下は、複数のサブネットとカスタムネットワーク名を持つネットワーク設定の設定ファイルエントリーの例です。

```
[global]
public_network = 172.16.14.0/24,172.16.15.0/24
cluster_network = 172.16.12.0/24,172.16.13.0/24
ms_bind_ipv4 = True
ms_bind_ipv6 = False
```

4. コマンド **openstack overcloud ceph deploy** を **--config** オプションとともに使用して、設定ファイルをデプロイします。

```
$ openstack overcloud ceph deploy \
  --config initial-ceph.conf --network-data network_data.yaml
```

3.7. OSD の CRUSH 階層の設定

OSD デプロイメント中にカスタム Controlled Replication Under Scalable Hashing (CRUSH) 階層を設定して、OSD **location** 属性を Ceph Storage クラスター **hosts** 仕様に追加できます。**location** 属性は、OSD が CRUSH 階層内のどこに配置されるかを設定します。



注記

location 属性は、最初の CRUSH ロケーションのみを設定します。その後の属性変更は無視されます。

手順

1. アンダークラウドノードに **stack** ユーザーとしてログインします。
2. **stackrc** アンダークラウド認証情報ファイルを入手します。
\$ source ~/stackrc
3. カスタム CRUSH 階層を定義する設定ファイルを作成します (例: **crush_hierarchy.yaml**)。
4. そのファイルに以下の設定を追加します。

```
ceph_crush_hierarchy:
  <osd_host>:
    root: default
```

```

    rack: <rack_num>
  <osd_host>:
    root: default
    rack: <rack_num>
  <osd_host>:
    root: default
    rack: <rack_num>

```

- **<osd_host>** を、OSD がデプロイされているノードのホスト名 (**ceph-0** など) に置き換えます。
- **<rack_num>** を、OSD がデプロイメントされているラックの番号 (**r0** など) に置き換えます。

5. カスタム OSD レイアウトを使用して Ceph クラスタをデプロイします。

```

openstack overcloud ceph deploy \
  deployed_metal.yaml \
  -o deployed_ceph.yaml \
  --osd-spec osd_spec.yaml \
  --crush-hierarchy crush_hierarchy.yaml

```

Ceph クラスタは、カスタム OSD レイアウトで作成されます。

上記のサンプルファイルは、次の OSD レイアウトになります。

ID	CLASS	WEIGHT	TYPE NAME	STATUS	REWEIGHT	PRI-AFF
-1		0.02939	root default			
-3		0.00980	rack r0			
-2		0.00980	host ceph-node-00			
0	hdd	0.00980	osd.0	up	1.00000	1.00000
-5		0.00980	rack r1			
-4		0.00980	host ceph-node-01			
1	hdd	0.00980	osd.1	up	1.00000	1.00000
-7		0.00980	rack r2			
-6		0.00980	host ceph-node-02			
2	hdd	0.00980	osd.2	up	1.00000	1.00000



注記

デバイスクラスは Ceph によって自動的に検出されますが、CRUSH ルールはプールに関連付けられています。プールは、引き続きオーバークラウドのデプロイメント中に **CephCrushRules** パラメーターを使用して定義および作成されます。

関連情報

詳細は、Red Hat Ceph Storage Installation Guideの [Red Hat Ceph Storage workload considerations](#) を参照してください。

3.8. CEPH サービス配置オプションの設定

カスタムロールファイルを使用して、どのノードがどの Ceph サービスを実行するかを定義できます。カスタムロールファイルは、環境が理由でデフォルトのロール割り当てが使用されない場合にのみ必要です。たとえば、ハイパーコンバージドノードをデプロイする場合、事前にデプロイされたコンピュー

トノードは、サービスタイプが **osd** の **osd** としてラベル付けされ、コンピュートインスタンスのリストを含む配置リストを持つ必要があります。

roles_data.yaml ファイルのサービス定義は、どのベアメタルインスタンスがどのサービスを実行するかを決定します。デフォルトでは、Controller ロールには CephMon および CephMgr サービスがあり、CephStorage ロールには CephOSD サービスがあります。ほとんどのコンポーザブルサービスとは異なり、Ceph サービスでは、サービスの設定方法を決定するために heat 出力は必要ありません。デプロイされた Ceph プロセスが Heat の実行前に発生する場合でも、**roles_data.yaml** ファイルが常に Ceph サービスの配置を決定します。

手順

1. アンダークラウドノードに **stack** ユーザーとしてログインします。
2. カスタムロールを定義する YAML 形式のファイルを作成します。
3. 設定ファイルをデプロイします。

```
$ openstack overcloud ceph deploy \
    deployed_metal.yaml \
    -o deployed_ceph.yaml \
    --roles-data custom_roles.yaml
```

3.9. CEPH ノードの SSH ユーザーオプションの設定

openstack overcloud ceph deploy コマンドはユーザーとキーを作成し、それらをホストに配布するため、このセクションの手順を実行する必要はありません。ただし、これはサポート対象のオプションです。

Cephadm は、SSH を使用してすべての管理対象のリモート Ceph ノードに接続します。Red Hat Ceph Storage クラスターのデプロイメントプロセスにより、すべてのオーバークラウド Ceph ノードでアカウントと SSH キーのペアが作成されます。その後、鍵ペアが Cephadm に渡され、ノードと通信できるようになります。

3.9.1. Red Hat Ceph Storage クラスター作成前の SSH ユーザーの作成

openstack overcloud ceph user enable コマンドを使用して、Ceph クラスターを作成する前に SSH ユーザーを作成できます。

手順

1. アンダークラウドノードに **stack** ユーザーとしてログインします。
2. SSH ユーザーを作成します。

```
$ openstack overcloud ceph user enable
```



注記

デフォルトのユーザー名は **ceph-admin** です。別のユーザー名を指定するには、**--cephadm-ssh-user** オプションを使用して別のユーザー名を指定します。

openstack overcloud ceph user enable --cephadm-ssh-user <custom_user_name>

--cephadm-ssh-user パラメーターを使用せずに、デフォルト名を使用することが推奨されます。

ユーザーが事前に作成されている場合は、**openstack overcloud ceph deploy** を実行するときにパラメーター **--skip-user-create** を使用します。

3.9.2. SSH ユーザーの無効化

SSH ユーザーを無効にすると、Cephadm が無効になります。Cephadm を無効にすると、Ceph クラスターを管理するサービスの機能が削除され、関連するコマンドが機能しなくなります。また、Ceph ノードのオーバークラウドのスケーリング操作も防ぎます。また、すべての公開および秘密 SSH キーも削除されます。

手順

1. アンダークラウドノードに **stack** ユーザーとしてログインします。
2. コマンド **openstack overcloud ceph user disable --fsid <FSID> ceph_spec.yaml** を使用して、SSH ユーザーを無効にします。



注記

FSID は、**deployed_ceph.yaml** 環境ファイルにあります。



重要

Cephadm を無効にする必要がない限り、**openstack overcloud ceph user disable** コマンドはお勧めしません。



重要

無効にした SSH ユーザーと Cephadm サービスを有効にするには、**openstack overcloud ceph user enable --fsid <FSID> ceph_spec.yaml** コマンドを使用します。



注記

このコマンドでは、以下を判別するために Ceph 仕様ファイルへのパスが必要です。

- SSH ユーザーを必要とするホスト。
- `_admin` ラベルがあり、プライベート SSH キーを必要とするホスト。
- 公開 SSH キーを必要とするホスト。

仕様ファイルとその生成方法について、詳しくはサービス仕様の生成を参照してください。

3.10. CEPH STORAGE コンテナへのアクセス

コンテナ化サービスへの移行 ガイドの [コンテナイメージの取得と変更](#) には、コンテナイメージを使用するためにレジストリーとアンダークラウドおよびオーバークラウド設定を準備する方法に関する手順と情報が含まれています。このセクションの情報をを使用して、これらの手順を Ceph Storage コンテナにアクセスするように調整します。

オーバークラウドから Ceph Storage コンテナにアクセスするには 2 つのオプションがあります。

- [リモートレジストリーからコンテナを直接ダウンロードする](#)
- [アンダークラウド上のコンテナのキャッシュ](#)

3.10.1. リモートレジストリーからコンテナを直接ダウンロードする

リモートレジストリーからコンテナを直接ダウンロードするように Ceph を設定できます。

手順

1. [コンテナイメージの準備](#) の手順を使用して、`containers-prepare-parameter.yaml` ファイルを作成します。
2. [プライベートレジストリーからコンテナイメージを取得する](#) の説明に従って、`ContainerImageRegistryCredentials` パラメーターを使用して、リモートレジストリーの認証情報を `containers-prepare-parameter.yaml` ファイルに追加します。
3. Ceph をデプロイするときは、`openstack overcloud cephdeploy` コマンドを使用して、`containers-prepare-parameter.yaml` ファイルを渡します。

```
openstack overcloud ceph deploy \
  --container-image-prepare containers-prepare-parameter.yaml
```



注記

[アンダークラウドでのコンテナのキャッシュ](#) で説明されているように、アンダークラウドでコンテナをキャッシュしない場合は、Ceph をデプロイするときに同じ `containers-prepare-parameter.yaml` ファイルを `openstack overcloud cephdeploy` コマンドに渡す必要があります。これにより、コンテナがアンダークラウドにキャッシュされます。

結果

Containers-prepare-parameter.yaml 内の認証情報は、**cephadm** コマンドによってリモートレジストリーに対する認証と Ceph Storage コンテナのダウンロードに使用されます。

3.10.2. アンダークラウド上のコンテナのキャッシュ

[準備中のイメージの変更](#) 手順では、次のコマンドの使用について説明しています。

```
sudo openstack tripleo container image prepare \  
-e ~/containers-prepare-parameter.yaml \  

```

[リモートレジストリーからのコンテナの直接ダウンロード](#)で説明されているように、**--container-image-prepare** オプションを使用しないで **openstack overcloud cephdeploy** コマンドに認証認証情報を提供し、リモートレジストリーから Ceph コンテナを直接 ダウンロードする場合は、Ceph をデプロイする前に **sudo openstack tripleo container image prepare** コマンドを実行する必要があります。

第4章 RED HAT CEPH STORAGE クラスターのカスタマイズ

director は、デフォルト設定で Red Hat Ceph Storage をデプロイします。このデフォルト設定をカスタマイズできます。

前提条件

- ストレージネットワークが設定された状態でデプロイされた Ceph Storage ノード。
- **openstack overcloud node provision -o ~/deployed_metal.yaml ...** によって出力されたデプロイされたベアメタルファイル。

4.1. 設定オプション

Red Hat Ceph Storage クラスターを設定するには、複数のオプションがあります。

手順

1. アンダークラウドノードに **stack** ユーザーとしてログインします。
2. オプション: 標準形式の初期化 (ini) ファイルを使用して Ceph クラスターを設定します。
 - a. 設定オプションを含むファイルを作成します。
以下は、単純な設定ファイルの例です。

```
[global]
osd crush chooseleaf type = 0
log_file = /var/log/ceph/$cluster-$type.$id.log

[mon]
mon_cluster_log_to_syslog = true
```

- b. 設定ファイルを作成します。
- c. **openstack overcloud ceph deploy --config <configuration_file_name>** コマンドを使用して設定をデプロイします。
<configuration_file_name> を作成したファイルの名前に置き換えます。

```
$ openstack overcloud ceph deploy --config initial-ceph.conf
```

3. オプション: 設定値を **cephadm bootstrap** コマンドに送信します。
openstack overcloud ceph deploy --force \ --cephadm-extra-args '<optional_arguments>' \ **<optional_arguments>** を設定値に置き換えて、基になるコマンドに提供します。



注記

引数 **--log-to-file** および **--skip-prepare-host** を使用する場合は、コマンド **openstack overcloud ceph deploy --force \ --cephadm-extra-args '--log-to-file --skip-prepare-host'** が使用されます。

4.2. サービス仕様の生成 (オプション)

Red Hat Ceph Storage クラスターサービス仕様は、Ceph Storage サービスのデプロイメントを記述する YAML ファイルです。Ceph Storage クラスターがデプロイされる前に、**tripleo** によって自動的に生成されます。通常は、個別に生成する必要はありません。

カスタムサービス仕様を作成して、Red Hat Ceph Storage クラスターをカスタマイズできます。

手順

1. アンダークラウドノードに **stack** ユーザーとしてログインします。

2. 仕様ファイルを生成します。

```
openstack overcloud ceph spec -o '<specification_file>'
```

<specification_file> を現在のサービス仕様で生成するファイルの名前に置き換えます。

```
openstack overcloud ceph spec -o '~/ceph_spec.yaml'
```

3. 生成されたファイルを編集して必要な設定を追加します。

4. カスタムサービス仕様をデプロイします。

```
openstack overcloud ceph deploy \ deployed_metal.yaml \ -o deployed_ceph.yaml \ --  
ceph-spec <specification_file>
```

<specification_file> をカスタムサービス仕様ファイルの名前に置き換えます。

```
openstack overcloud ceph deploy \ deployed_metal.yaml \ -o deployed_ceph.yaml \ --  
ceph-spec ~/ceph_spec.yaml
```

4.3. RED HAT OPENSTACK PLATFORM と RED HAT CEPH STORAGE の CEPH コンテナ

NFS Ganesha で Red Hat Ceph Storage を使用するように、Red Hat Openstack Platform (RHOSP) を設定するには、Ceph Storage コンテナが必要です。外部 Ceph Storage クラスターがブロック (RBD 経由)、オブジェクト (RGW 経由)、またはファイル (ネイティブ CephFS 経由) ストレージのみを提供する場合、Ceph Storage コンテナは不要です。

RHOSP 17.0 では、Red Hat Enterprise Linux 9 と互換性を保つために、Red Hat Ceph Storage 5.x (Ceph パッケージ 16.x) 以降が必要です。Ceph Storage 5.x コンテナは、認証が必要なレジストリーである **registry.redhat.io** でホスティングされます。詳細は [コンテナイメージを準備するためのパラメーター](#) を参照してください。

4.4. 高度な OSD 仕様の設定

デフォルトの仕様では、Ceph Storage クラスターに必要な機能が提供されない場合、高度な OSD 仕様を設定します。

手順

1. アンダークラウドノードに **stack** ユーザーとしてログインします。

2. 高度な OSD 仕様を定義する YAML 形式のファイルを作成します。

以下は、カスタム OSD 仕様の例です。

```
data_devices:
```

```
rotational: 1
db_devices:
  rotational: 0
```

この例では、すべてのローテーションデバイスがデータデバイスになり、すべての非ローテーションデバイスが共有デバイスとして使用される OSD 仕様を作成します。動的 Ceph サービス仕様が構築されると、**service_type** が **osd** の場合、仕様ファイルにあるものは、すべて仕様のセクションに追加されます。

3. 仕様ファイルを保存します。

4. 仕様をデプロイします。

```
openstack overcloud ceph deploy \ --osd-spec <osd_specification_file>
```

<osd_specification_file> を作成した仕様ファイルの名前に置き換えます。

```
$ openstack overcloud ceph deploy \ --osd-spec osd_spec.yaml \
```

関連情報

サービス仕様で OSD を設定するために使用される OSD 関連の属性のリストは、**Red Hat Ceph Storage 操作ガイド** の [OSD をデプロイするための高度なサービス仕様およびフィルター](#) を参照してください。

4.5. ノード固有のオーバーライドからの移行

Red Hat OpenStack Platform 17.0 より前は、ノード固有のオーバーライドを使用して、同種ではないサーバーハードウェアを管理していました。これは、カスタム OSD 仕様ファイルで実行されるようになりました。カスタム OSD 仕様ファイルの作成方法については、[Configuring advanced OSD specifications](#) を参照してください。

4.6. CEPH 伝送時暗号化の有効化

メッセンジャーバージョン 2 プロトコルの **secure mode** を使用して、すべての Ceph Storage トラフィックの暗号化を有効にします。Red Hat Ceph Storage [データセキュリティおよび強化](#) ガイドの [暗号化と鍵管理](#) の説明に従って、Ceph Storage を設定し、Ceph オンワイヤー暗号化を有効にします。

関連情報

Ceph 伝送時暗号化に関する詳しい情報は、Red Hat Ceph Storage **Architecture Guide** の [Ceph on-wire encryption](#) を参照してください。

第5章 ストレージサービスのカスタマイズ

director の heat テンプレートコレクションには、基本的な Ceph Storage 設定を有効にするために必要なテンプレートと環境ファイルが含まれています。

director は `/usr/share/openstack-tripleo-heat-templates/environments/cephadm/cephadm.yaml` 環境ファイルを使用して、**openstack overcloud ceph deploy** によってデプロイされた Ceph Storage クラスタに設定を追加し、デプロイ中にそれをオーバークラウドと統合します。

5.1. カスタム環境ファイルの設定

Director は、デプロイされた Red Hat Ceph Storage クラスタに基本的なデフォルト設定を適用します。カスタム環境ファイルで追加の設定を定義する必要があります。

手順

1. アンダークラウドに **stack** ユーザーとしてログインします。
2. カスタム設定を定義するファイルを作成します。
vi /home/stack/templates/storage-config.yaml
3. ファイルに **parameter_defaults** セクションを追加します。
4. カスタム設定パラメーターを追加します。パラメーター定義の詳細については、[オーバークラウドのパラメーター](#) を参照してください。

```
parameter_defaults:
  CinderEnableiscsiBackend: false
  CinderEnableRbdBackend: true
  CinderBackupBackend: ceph
  NovaEnableRbdBackend: true
  GlanceBackend: rbd
```



注記

カスタム設定ファイルで定義されたパラメーターは、`/usr/share/openstack-tripleo-heat-templates/environments/cephadm/cephadm.yaml` の対応するデフォルト設定をオーバーライドします。

5. ファイルを保存します。

関連情報

カスタム設定は、オーバークラウドのデプロイ中に、適用されます。

5.2. RED HAT CEPH STORAGE 配置グループ

配置グループ (PG) は、大規模な動的で効率的なオブジェクトトラッキングを容易にします。OSD 障害または Ceph Storage クラスタの再調整が発生した場合、Ceph は配置グループと配置グループの内容を移動または複製できます。これにより、Ceph Storage クラスタは効率的に再調整および回復できます。

次のパラメーターが Ceph 設定ファイルに含まれていないかぎり、配置グループとレプリカ数の設定はデフォルトから変更されません。

- **osd_pool_default_size**
- **osd_pool_default_pg_num**
- **osd_pool_default_pgp_num**

openstack overcloud deploy コマンドを使用して、オーバークラウドをデプロイすると、有効な Red Hat OpenStack Platform サービスごとにプールが作成されます。たとえば、次のコマンドは、Compute サービス (nova)、Block Storage サービス (cinder)、および Image サービス (glance) のプールを作成します。

```
openstack overcloud deploy --templates \
  -e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/cephadm-rbd-only.yaml
```

コマンドに **-e environments/cinder-backup.yaml** を追加すると、**backups** という名前のプールが作成されます。

```
openstack overcloud deploy --templates \
  -e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/cephadm-rbd-only.yaml
  -e environments/cinder-backup.yaml
```

プールごとに配置グループ番号を設定する必要はありません。**pg_autoscale_mode** 属性はデフォルトで有効になっています。ただし、**target_size_ratio** または **pg_num** 属性を設定することを推奨します。これにより、データの再調整が最小限に抑えられます。

プールごとに **target_size_ratio** 属性を設定するには、次の例のような設定ファイルエントリーを使用します。

```
parameter_defaults:
  CephPools:
    - name: volumes
      target_size_ratio: 0.4
      application: rbd
    - name: images
      target_size_ratio: 0.1
      application: rbd
    - name: vms
      target_size_ratio: 0.3
      application: rbd
```

この例では、サービスごとに使用されるデータの割合は次のようになります。

- Cinder ボリューム - 40%
- Glance イメージ - 10%
- Nova vms - 30%
- 他のプール用の空き容量 - 20%

これらの値は、予想される使用量に基づいて設定してください。**CephPools** パラメーターをオーバーライドしない場合、各プールはデフォルトの配置グループ番号を使用します。オートスケーラーは使用状況に基づいて時間の経過とともにこの数を自動的に調整しますが、データは Ceph クラスター内で移動されます。これは、計算リソースを使用します。

ターゲットサイズ比ではなく、配置グループ数を設定する場合は、例の **target_size_ratio** を **pg_num** に置き換えます。予想される使用量に基づいて、プールごとに異なる整数を使用してください。

Red Hat Ceph Storage プロセッサ、ネットワークインターフェイスカード、および電源管理インターフェイスに関する推奨事項は、[Red Hat Ceph Storage Hardware Guide](#) を参照してください。

5.3. CEPH METADATA SERVER の有効化

Ceph Metadata Server (MDS) は **ceph-mds** デーモンを実行します。このデーモンは、CephFS に保存されているファイルに関連するメタデータを管理します。CephFS は、ネイティブまたは NFS プロトコルで使用できます。



注記

Red Hat は、Shared File Systems サービス (manila) のネイティブ CephFS および CephFS NFS バックエンドを使用した Ceph MDS のデプロイをサポートしています。

手順

- Ceph MDS を有効にするには、オーバークラウドのデプロイ時、以下の環境ファイルを使用します。

```
/usr/share/openstack-tripleo-heat-templates/environments/cephadm/ceph-mds.yaml
```



注記

デフォルトでは、Ceph MDS はコントローラーノードにデプロイされます。Ceph MDS を独自の専用ノードにデプロイできます。

関連情報

- [Red Hat Ceph Storage ファイルシステム ガイド](#)

5.4. CEPH OBJECT GATEWAY オブジェクトストレージ

Ceph Object Gateway (RGW) は、Red Hat Ceph Storage クラスター内のオブジェクトストレージ機能にアクセスするためのインターフェイスを提供します。

director を使用して、Ceph をデプロイすると、director は RGW を自動的に有効にします。これは、Object Storage サービス (swift) を直接置き換えるものです。通常、Object Storage サービスを使用するサービスは、追加の設定なしで、代わりに RGW を使用できます。Object Storage サービスは、アップグレードされた Ceph クラスターのオブジェクトストレージオプションとして引き続き利用できます。

個別の RGW 環境ファイルを有効にする必要はありません。その他のオブジェクトストレージオプションの環境ファイルの詳細については、「[Red Hat OpenStack Platform オブジェクトストレージのデプロイメントオプション](#)」を参照してください。

デフォルトでは、Ceph Storage は Object Storage Daemon (OSD) ごとに 250 の配置グループを許可します。RGW を有効にすると、Ceph Storage は RGW に必要な次の 6 つの追加プールを作成します。

- **.rgw.root**
- **<zone_name>.rgw.control**

- `<zone_name>.rgw.meta`
- `<zone_name>.rgw.log`
- `<zone_name>.rgw.buckets.index`
- `<zone_name>.rgw.buckets.data`



注記

デプロイメントでは、`<zone_name>` は、プールが属するゾーンの名前に置き換えられます。

関連情報

- RGW の詳細は、Red Hat Ceph Storage [Object Gateway ガイド](#) を参照してください。
- Swift の代わりに RGW を使用方法の詳細については、[Block Storage Backup Guide](#) を参照してください。

5.5. RED HAT OPENSTACK PLATFORM オブジェクトストレージのデプロイメントオプション

オーバークラウドのオブジェクトストレージをデプロイするには、3つのオプションがあります。

- Ceph オブジェクトゲートウェイ (RGW)
「[Ceph Object Gateway オブジェクトストレージ](#)」の説明に従って、RGW をデプロイするには、オーバークラウドのデプロイ時、以下の環境ファイルを含めます。

```
-e environments/cephadm/cephadm.yaml
```

この環境ファイルは、Ceph ブロックストレージ (RBD) と RGW の両方を設定します。

- Object Storage サービス (swift)
RGW ではなく、Object Storage サービス (swift) をデプロイするには、オーバークラウドのデプロイ時、以下の環境ファイルを含めます。

```
-e environments/cephadm/cephadm-rbd-only.yaml
```

cephadm-rbd-only.yaml ファイルは Ceph RBD を設定しますが、RGW は設定しません。



注記

Red Hat Ceph Storage クラスターをアップグレードする前に、Object Storage サービス (swift) を使用した場合は、アップグレード中に、**environments/ceph-ansible/ceph-ansible.yaml** ファイルを **environment/cephadm/cephadm-rbd-only.yaml** に置き換えると、RGW ではなく、Object Storage サービス (swift) を引き続き使用できます。詳細は、[Keeping Red Hat OpenStack Platform Updated](#) を参照してください。

Red Hat OpenStack Platform は、Object Storage サービス (swift) から Ceph Object Gateway (RGW) への移行をサポートしていません。

- オブジェクトストレージなし
RGW または Object Storage サービス (swift) ではなく、RBD で Ceph をデプロイするには、オーバークラウドのデプロイ時、以下の環境ファイルを含めます。

```
-e environments/cephadm/cephadm-rbd-only.yaml
-e environments/disable-swift.yaml
```

cephadm-rbd-only.yaml ファイルは RBD を設定しますが、RGW は設定しません。**disable-swift.yaml** ファイルは、Object Storage サービス (swift) がデプロイされないようにします。

5.6. CEPH を使用するための BLOCK STORAGE BACKUP SERVICE の設定

Block Storage Backup サービス (cinder-backup) はデフォルトで無効になっています。Ceph で使用するには、有効にする必要があります。

手順

Block Storage Backup サービス (cinder-backup) を有効にするには、オーバークラウドのデプロイ時、以下の環境ファイルを使用します。

```
`/usr/share/openstack-tripleo-heat-templates/environments/cinder-backup.yaml`.
```

5.7. CEPH ノード向けの複数のボンディングされたインターフェイスの設定

ボンディングされたインターフェイスを使用して、複数の NIC を組み合わせ、ネットワーク接続に冗長性を追加します。Ceph ノードに十分な NIC がある場合には、各ノードに複数のボンディングされたインターフェイスを作成して冗長化機能を拡張することができます。

ノードが必要とするネットワーク接続ごとに結合インターフェイスを使用します。これにより、冗長性と各ネットワーク専用の接続の両方が提供されます。

詳細および手順については、**director のインストールおよび使用方法** ガイドの [オーバークラウドネットワークのプロビジョニング](#) を参照してください。

第6章 ネイティブ CEPHFS を使用する SHARED FILE SYSTEMS サービスのデプロイ

CephFS は、統合分散ストレージプラットフォームである Red Hat Ceph Storage の拡張性の高いオープンソースの分散ファイルシステムコンポーネントです。Ceph Storage は、Reliable Autonomic Distributed Object Store (RADOS) を使用してオブジェクトストレージ、ブロックストレージ、およびファイルストレージを実装します。POSIX に対応した CephFS は、Ceph Storage クラスタへのファイルアクセスを提供します。

Shared File Systems サービス (manila) を使用すると、ユーザーは CephFS で共有を作成し、ネイティブの Ceph FS プロトコルを使用して、それらにアクセスできます。Shared File Systems サービスは、OpenStack 内からこれらのファイル共有のライフサイクルを管理します。

今回のリリースでは、director はネイティブ CephFS バックエンドを使用する Shared File Systems をオーバークラウドにデプロイすることができます。



重要

この章は、ネイティブ CephFS NAS プロトコルによって Red Hat OpenStack Platform (RHOSP) クラウドでセルフサービスの共有ファイルシステムサービスを提供するためのネイティブ CephFS のデプロイと使用に関連しています。このタイプのデプロイメントでは、ゲスト仮想マシンが Ceph パブリックネットワークおよびインフラストラクチャーにアクセスする必要があります。汎用の OpenStack Platform デプロイメントには適していない許容信頼モデルが必要なため、信頼済み OpenStack Platform テナントでのみネイティブ CephFS をデプロイしてください。従来のテナント信頼モデルを使用する汎用の OpenStack Platform デプロイメントでは、NFS プロトコルを介して CephFS をデプロイすることができます。

6.1. ネイティブドライバを使用する CEPHFS

CephFS ネイティブドライバは、OpenStack Shared File Systems サービス (manila) と Red Hat Ceph Storage を結び付けます。Red Hat OpenStack (RHOSP) director を使用する場合、コントローラーノードはマネージャー、メタデータサーバー (MDS)、および モニター (MON) 等の Ceph デーモンならびに Shared File Systems サービスをホストします。

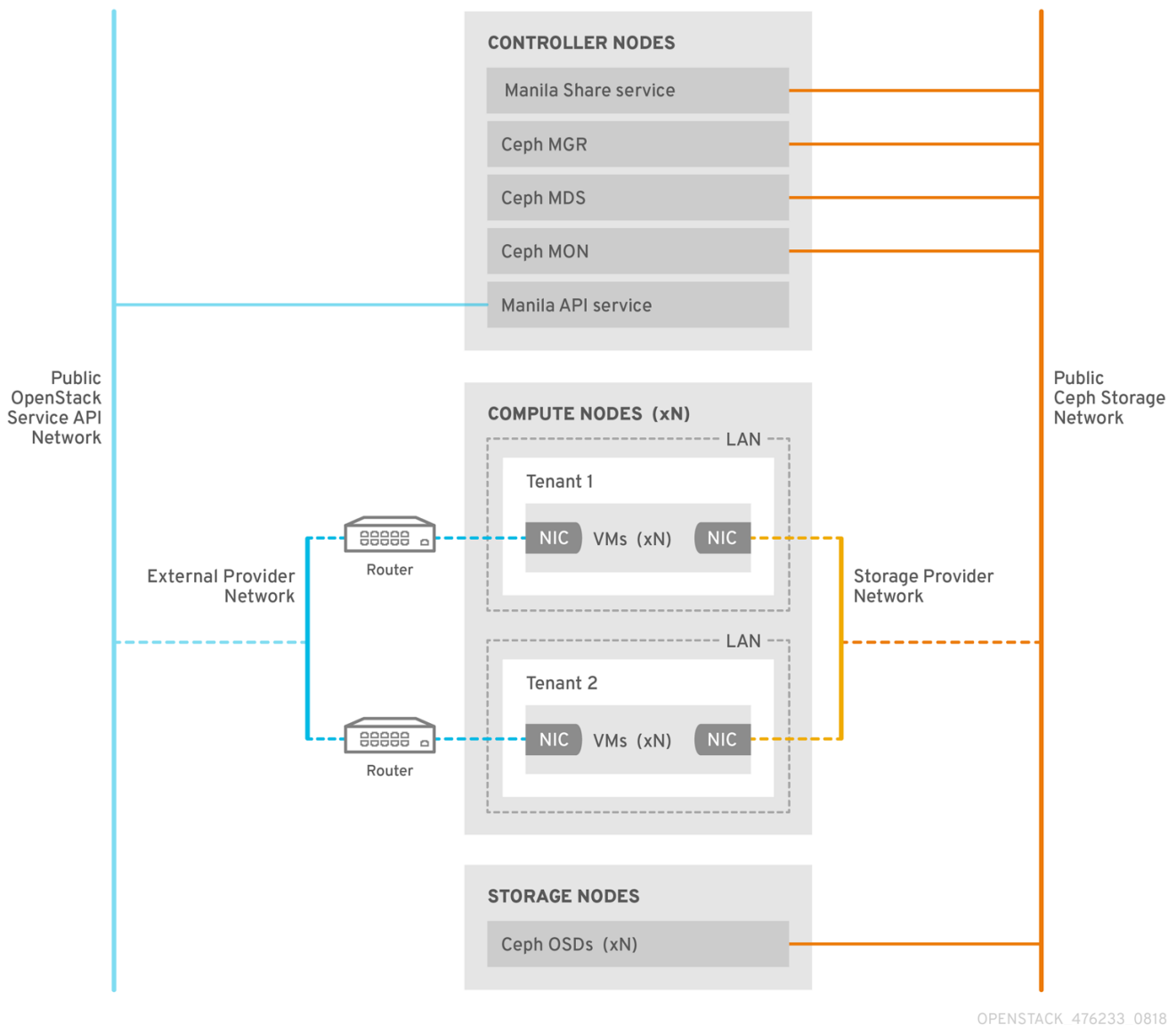
コンピュータノードは1つまたは複数のプロジェクトをホストすることができます。以前はテナントと呼ばれていたプロジェクトは、以下の図では白い長方形で表示されます。プロジェクトには、ユーザー管理の仮想マシンが含まれています。これは、2つの NIC を持つグレーのボックスで表されます。ceph および manila デーモンプロジェクトにアクセスするには、パブリック Ceph ストレージネットワーク経由でデーモンに接続します。

このネットワーク上で、Ceph オブジェクトストレージデーモン (OSD) の提供するストレージノードに保管されたデータにアクセスすることができます。プロジェクトがホストするインスタンスまたは仮想マシン (VM) は、2つの NIC と共に起動します。1つの NIC はストレージプロバイダーネットワーク専用で、2つ目の NIC は外部プロバイダーネットワークに接続するためのプロジェクト所有のルーター専用です。

ストレージプロバイダーネットワークは、プロジェクト上で動作する仮想マシンをパブリック Ceph ストレージネットワークに接続します。Ceph パブリックネットワークは、Ceph オブジェクトストレージノード、メタデータサーバー (MDS)、およびコントローラーノードへのバックエンドアクセスを提供します。

ネイティブドライバを使用する場合、CephFS では、クォータの適用、確実なプロジェクト間の分離、およびセキュリティ確保のために、クライアントおよびサーバーとの協調が必要になります。ネイティブドライバを使用する CephFS は、プライベートクラウド上の信頼済みエンドユーザーが定義

された環境で適切に機能します。この設定での協調および適切な動作のためには、ソフトウェアはユーザーの管理下で動作している必要があります。



6.2. ネイティブ CEPHFS バックエンドセキュリティー

ネイティブ CephFS バックエンドには、Red Hat OpenStack Platform (RHOSP) テナントに許容信頼モデルが必要です。この信頼モデルは、OpenStack Platform が提供するサービスの背後にあるインフラストラクチャーにユーザーが直接アクセスするのを意図的にブロックする汎用の OpenStack Platform クラウドには適切ではありません。

ネイティブ CephFS を使用する場合、ユーザーコンピュートインスタンスは Ceph サービスのデーモンが公開される Ceph パブリックネットワークに直接接続されます。ユーザー仮想マシンで実行される CephFS クライアントは Ceph サービスデーモンと連携し、ファイルデータブロックの読み取りおよび書き込みのために RADOS と直接対話します。

Shared File Systems (manila) 共有サイズを適用する CephFS クォータは、(RHOSP) ユーザーが所有する VM などのクライアント側で適用されます。ユーザー仮想マシンのクライアント側ソフトウェアは最新ではない可能性があるため、重要なクラウドインフラストラクチャーが Ceph サービスポートをターゲットにする悪意のあるまたは意図的ではないが有害なソフトウェアに対して脆弱になる可能性があります。

信頼できるユーザーがクライアント側ソフトウェアを最新の状態に維持する環境でのみ、ネイティブ CephFS をバックエンドとしてデプロイします。Red Hat Ceph Storage インフラストラクチャーに影響を与える可能性があるソフトウェアが VM で実行されないようにしてください。

多くの信頼されていないユーザーにサービスを提供する汎用の RHOSP デプロイメントの場合は、NFS で CephFS をデプロイします。NFS による CephFS の使用に関する詳細は、[Red Hat Ceph Storage および Red Hat OpenStack Platform を director とともにデプロイする](#) を参照してください。

ユーザーがクライアント側のソフトウェアを最新版に維持できない、また仮想マシンから有害なソフトウェアを除外できない状況でも、NFS バックエンドに CephFS を使用する設定を使用すると、ユーザーは NFS サーバーの公開側だけにアクセスでき、Ceph インフラストラクチャー自体にはアクセスできません。NFS は、同じ種類の協調クライアントを必要としないため、最悪の場合は、ユーザー VM からの攻撃によって、背後にある Ceph Storage インフラストラクチャーに損傷を与えずに、NFS ゲートウェイに損傷を与える可能性があります。

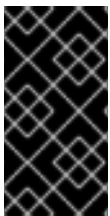
ネイティブ CephFS バックエンドを信頼できる全ユーザーに公開できますが、以下のセキュリティー対策を実施する必要があります。

- ストレージネットワークをプロバイダーネットワークとして設定する。
- ロールベースのアクセス制御 (RBAC) ポリシーを適用して、Storage プロバイダーネットワークのセキュリティーを保護する。
- プライベートファイル共有種別を作成します。

6.3. ネイティブ CEPHFS デプロイメント

Red Hat OpenStack Platform (RHOSP) 環境への一般的なネイティブ Ceph ファイルシステム (CephFS) インストールには、以下のコンポーネントが含まれます。

- コンテナ化された Ceph メタデータサーバー (MDS)、Ceph モニター (MON)、および Shared File Systems (manila) サービスを実行する RHOSP コントローラーノード。これらのサービスのいくつかは、同じノードに共存する場合や、1つまたは複数の専用のノードを持つ場合があります。
- Ceph Storage ノードで実行されるコンテナ化されたオブジェクトストレージデーモン (OSD) を持つ Ceph Storage クラスター。
- クライアントが Ceph サービスデーモンと通信できる Ceph パブリックネットワークとして機能する分離ストレージネットワーク。これを容易にするため、ユーザーはストレージネットワークをプロバイダーネットワークとして使用し、仮想マシンを接続して CephFS 共有をマウントします。



重要

CephFS ネイティブドライバーと共に Shared File Systems サービス (manila) を使用して、Manila CSI で OpenShift Container Platform にファイル共有を提供することはできません。Red Hat ではこのタイプのデプロイメントをサポートしないためです。詳細は、Red Hat のサポートにお問い合わせください。

Shared File Systems (manila) サービスの提供する API により、テナントはファイルシステムの共有を要求することができます。これは、ドライバーモジュールにより実施されます。Red Hat CephFS のドライバー (**manila.share.drivers.cephfs.driver.CephFSDriver**) により、Shared File System サービスはネイティブ CephFS をバックエンドとして使用することができます。director の管理する統合デプロイメントにおいて、ネイティブ CephFS をインストールすることができます。

director がオーバークラウドに CephFS バックエンドを持つ Shared File Systems サービスをデプロイする場合、これにより必要なデータセンターのストレージネットワークが自動的に作成されます。ただし、オーバークラウド上に対応するストレージプロバイダーネットワークを作成する必要があります。

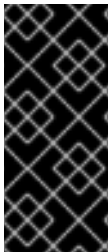
ネットワーク計画の詳細については、**director のインストールと使用** の [オーバークラウドネットワーク](#) を参照してください。

ノードの `/var/lib/config-data/puppet-generated/manila/etc/manila/manila.conf` ファイルを編集して Shared File Systems サービスを手動で設定することができますが、今後のオーバークラウド更新時に、Red Hat OpenStack Platform director を使用して任意の設定を上書きすることができます。Red Hat は、director が管理する Shared File Systems サービスのデプロイメントのみをサポートします。

6.4. 要件

以下の要件を満たす場合には、新規または既存の Red Hat OpenStack Platform (RHOSP) 環境と共にネイティブ CephFS バックエンドをデプロイすることができます。

- Red Hat OpenStack Platform バージョン 17.0 以降を使用する。
- ネイティブ CephFS バックエンドと同時に、新しい Red Hat Ceph Storage クラスターを設定する。Ceph Storage のデプロイ方法については、[director を使用した Red Hat Ceph Storage と Red Hat OpenStack Platform のデプロイ](#) を参照してください。



重要

RHOSP Shared File Systems サービス (manila) とネイティブ CephFS バックエンドの組み合わせは、Red Hat Ceph Storage バージョン 5.2 以降との使用でサポートされます。システムにインストールされている Ceph Storage のバージョンを確認する方法についての詳細は、[What are the Red Hat Ceph Storage releases and corresponding Ceph package versions?](#) を参照してください。

- Shared File Systems サービスをコントローラーノードにインストールします。これがデフォルトの動作です。
- Shared File Systems サービスには、CephFS バックエンドのインスタンスを1つだけ使用してください。

6.5. ファイル共有

Shared File Systems サービス (manila)、Ceph File System (CephFS)、および NFS バックエンドに CephFS を使用する設定では、ファイル共有の取り扱いが異なります。

Shared File Systems サービスが提供するそれぞれのファイル共有は、個別のファイルシステムの名前空間であり、また特定のサイズを持つストレージのユニットです。元来、共有ファイルシステムのストレージでは、複数のクライアントが任意のファイル共有に接続してデータの読み取りおよび書き込みが可能です。クライアントが接続できるためには、Shared File Systems サービスのアクセス制御 API を通じて各クライアントにファイル共有へのアクセス権限を付与する必要があります。

CephFS では、ファイル共有は、特定のストレージプールまたは名前空間をポイントする、クォータおよびレイアウトが定義されたディレクトリーとして扱われます。CephFS のクォータは、ディレクトリーのサイズを Shared File Systems サービスが作成するファイル共有のサイズに制限します。Ceph ファイル共有へのアクセス可否は、MDS の認証機能により決定されます。

ネイティブ CephFS では、ファイル共有のプロビジョニングおよびアクセスに CephFS プロトコルが使用されます。アクセス制御は、CephFS ユーザー名を使用する CephX 認証スキームで実行されます。

6.6. ネイティブ CEPHFS 分離ネットワーク

ネイティブ CephFS デプロイメントでは、director が Ceph パブリックネットワークとしてデプロイする分離ストレージネットワークを使用されます。クライアントはこのネットワークを使用して、さまざまな Ceph インフラストラクチャサービスデーモンと通信します。ネットワークの分離に関する詳細は、[Director のインストールおよび使用方法](#) の [ネットワークの分離](#) を参照してください。

6.7. ネイティブ CEPHFS 環境のデプロイ

環境をデプロイする準備が整ったら、ネイティブ CephFS バックエンドの設定に必要なカスタム環境およびロールを指定して **openstack overcloud deploy** コマンドを使用します。

openstack overcloud deploy コマンドでは、その他の必要なオプションに加えて以下のオプションを指定します。

Action	オプション	追加情報
network_data.yaml を使用してネットワーク設定を指定します。	<code>[filename] -n /usr/share/openstack-tripleo-heat-templates/network_data.yaml</code>	カスタム環境ファイルを使用して、このネットワークデータ環境ファイルで指定したデフォルトネットワークの値を上書きすることができます。これは、分離ネットワークを使用する際に利用可能なデフォルトのネットワークデータファイルです。このファイルは、簡潔にするために openstack overcloud deploy コマンドから省略することができます。
Ceph デーモンをデプロイします。	<code>-e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/cephadm.yaml</code>	director を使用した Red Hat Ceph Storage および Red Hat OpenStack Platform のデプロイメントの オーバークラウドデプロイメントの開始
ceph-mds.yaml を使用して Ceph メタデータサーバーをデプロイする。	<code>-e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/ceph-mds.yaml</code>	director を使用した Red Hat Ceph Storage および Red Hat OpenStack Platform のデプロイメントの オーバークラウドデプロイメントの開始
ネイティブ CephFS バックエンドを使用する manila サービスをデプロイする	<code>-e /usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsnative-config.yaml</code>	環境ファイル

以下の例は、Ceph クラスター、Ceph MDS、ネイティブ CephFS バックエンド、および Ceph クラスターに必要なネットワークをデプロイするオプションが含まれる **openstack overcloud deploy command** コマンドを示しています。

```
[stack@undercloud ~]$ openstack overcloud deploy \
...
-n /usr/share/openstack-tripleo-heat-templates/network_data.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml \
-e /home/stack/network-environment.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/cephadm.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/ceph-mds.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsnative-config.yaml
```

openstack overcloud deploy コマンドについての詳しい情報は、[Director Installation and Usage のオーバークラウドのプロビジョニングおよびデプロイ](#) を参照してください。

6.8. NATIVE CEPHFS バックエンド環境ファイル

ネイティブ CephFS バックエンドを定義するための環境ファイルである **manila-cephfsnative-config.yaml** は、アンダークラウドノードのパス **/usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsnative-config.yaml** にあります。

manila-cephfsnative-config.yaml 環境ファイルには、Shared File Systems サービスのデプロイに関する設定が含まれます。バックエンドのデフォルト設定は、ほとんどの環境で機能するはずです。

Shared File Systems サービスをデプロイする際に director が使用するデフォルト値を例で示します。

```
[stack@undercloud ~]$ cat /usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsnative-config.yaml

# A Heat environment file which can be used to enable a
# a Manila CephFS Native driver backend.
resource_registry:
  OS::TripleO::Services::ManilaApi: ../deployment/manila/manila-api-container-puppet.yaml
  OS::TripleO::Services::ManilaScheduler: ../deployment/manila/manila-scheduler-container-puppet.yaml
  # Only manila-share is pacemaker managed:
  OS::TripleO::Services::ManilaShare: ../deployment/manila/manila-share-pacemaker-puppet.yaml
  OS::TripleO::Services::ManilaBackendCephFs: ../deployment/manila/manila-backend-cephfs.yaml

parameter_defaults:
  ManilaCephFSBackendName: cephfs ①
  ManilaCephFSDriverHandlesShareServers: false ②
  ManilaCephFSCephFSAuthId: 'manila' ③
  ManilaCephFSCephFSEnableSnapshots: true ④
  ManilaCephFSCephFSVolumeMode: '0755' ⑤
  # manila cephfs driver supports either native cephfs backend - 'CEPHFS'
  # (users mount shares directly from ceph cluster), or nfs-ganesha backend -
  # 'NFS' (users mount shares through nfs-ganesha server)
  ManilaCephFSCephFSProtocolHelperType: 'CEPHFS' ⑥
```

parameter_defaults ヘッダーから設定が始まります。具体的には、このヘッダーからの設定により、**resource_registry** で設定したデフォルト値を上書きすることができます。これには、CephFS バックエンドのデフォルトを設定する **OS::TripleO::Services::ManilaBackendCephFs** で定義した値

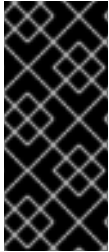
も含まれます。

- 1 **ManilaCephFSBackendName:** CephFS バックエンドの manila 設定の名前を定義します。ここでは、デフォルトのバックエンド名は **cephfs** です。
- 2 **ManilaCephFSDriverHandlesShareServers:** 共有用サーバーのライフサイクルをコントロールします。**false** に設定すると、ドライバーはライフサイクルを処理しません。CephFS バックエンドでサポートされるオプションはこれだけです。
- 3 **ManilaCephFSCephFSAuthId:** Ceph クラスタにアクセスするために director が manila サービスに作成する Ceph 認証 ID を定義します。
- 4 **ManilaCephFSCephFSEnableSnapshots:** スナップショットのアクティブ化をコントロールします。Ceph Storage 4.1 以降ではスナップショットがサポートされますが、このパラメーターのデフォルト値は **false** です。この値を **true** に設定すると、ドライバーは **snapshot_support** 機能を manila スケジューラーに報告します。
- 5 **ManilaCephFSCephVolumeMode** は、ネイティブ CephFS バックエンド上に作成される manila 共有に設定する UNIX パーミッションを制御します。デフォルト値は **755** です。
- 6 ネイティブ CephFS ドライバーを使用するには、**ManilaCephFSCephFSProtocolHelperType** を **CEPHFS** に設定する必要があります。

環境ファイルについての詳細は、Director のインストールと使用方法 の [環境ファイル](#) を参照してください。

第7章 CEPHFS-NFS を使用した SHARED FILE SYSTEMS サービスのデプロイ

NFS による Ceph File System (CephFS) を含む Shared File Systems サービス (manila) を使用すると、ブロックおよびオブジェクトストレージに使用すると同じ Red Hat Ceph Storage クラスタを使用して、NFS プロトコルを介してファイル共有を提供できます。詳しくは、[ストレージガイドの Shared File Systems サービス \(manila\) の設定](#) を参照してください。



重要

RHOSP 17.0 以降の NFS による CephFS を含む RHOSP Shared File Systems サービスは、Red Hat Ceph Storage バージョン 5.x 以降での使用がサポートされています。システムにインストールされている Ceph Storage のバージョンを確認する方法についての詳細は、[What are the Red Hat Ceph Storage releases and corresponding Ceph package versions?](#) を参照してください。

CephFS は、高度にスケーラブルな Red Hat Ceph Storage のオープンソース分散ファイルシステムのコンポーネントで、統一された分散ストレージプラットフォームです。Ceph Storage は、Reliable Autonomic Distributed Object Store (RADOS) を使用してオブジェクトストレージ、ブロックストレージ、およびファイルストレージを実装します。POSIX に対応した CephFS は、Ceph ストレージクラスターへのファイルアクセスを提供します。

Shared File Systems サービスにより、ユーザーは CephFS で共有を作成し、NFS-Ganesha によって NFS 4.1 でそれらにアクセスできます。NFS-Ganesha はファイル共有へのアクセスをコントロールし、NFS 4.1 プロトコルを使用してファイル共有をクライアントにエクスポートします。

Shared File Systems サービスは、RHOSP 内からこれらのファイル共有のライフサイクルを管理します。NFS バックエンドに CephFS を使用するようにサービスを設定した場合には、これらのファイル共有は CephFS クラスタから提供されますが、通常の NFS 共有として作成およびアクセスされます。

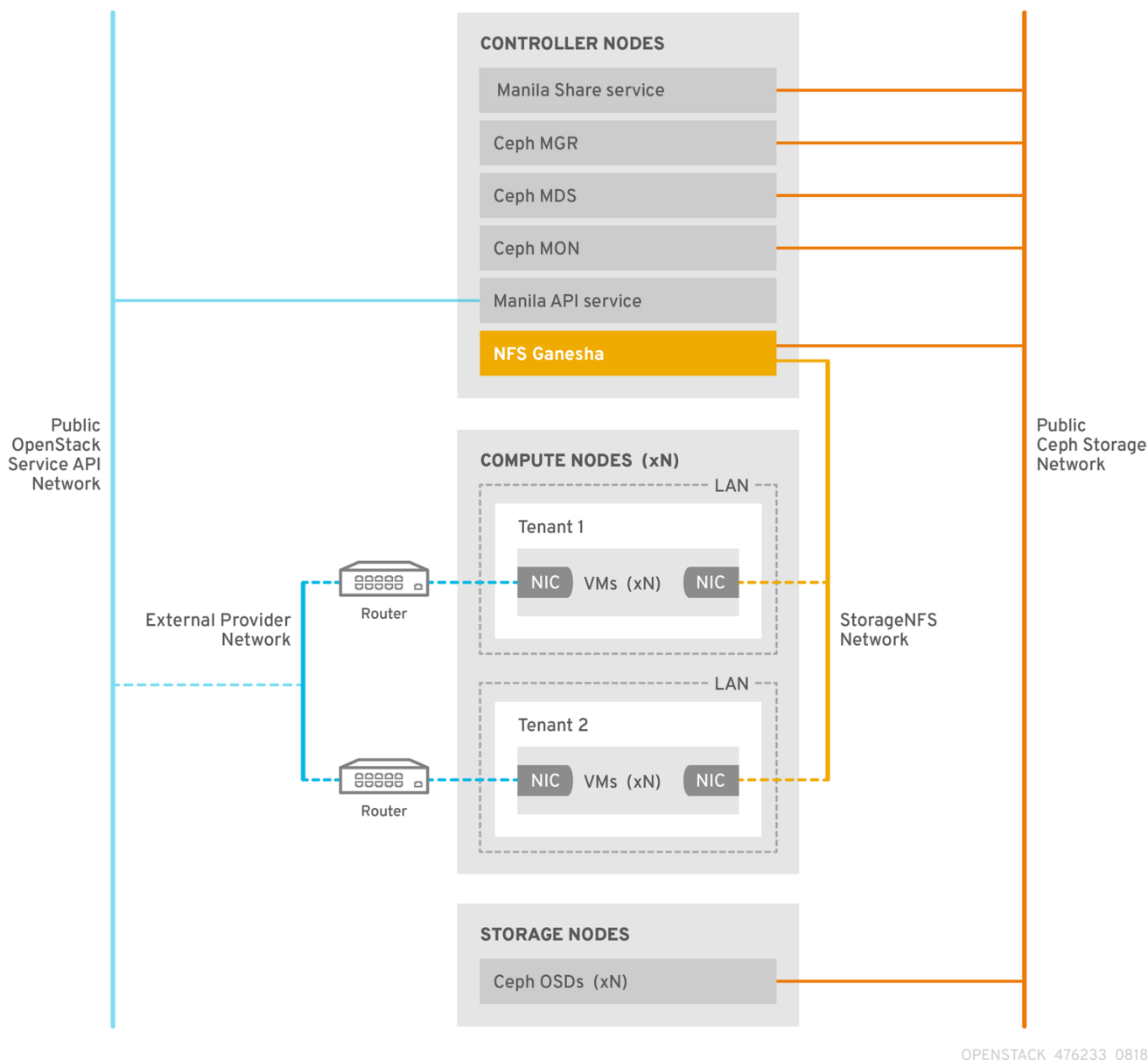
Shared File Systems サービスの詳細については、[ストレージガイドの Shared File Systems サービス \(manila\) の設定](#) を参照してください。

7.1. NFS ドライバーを使用した CEPHFS

Shared File Systems サービス (manila) の NFS バックエンドに CephFS を使用する設定には、Ceph メタデータサーバー (MDS)、NFS バックエンドに CephFS を使用するためのゲートウェイ (NFS-Ganesha)、および Ceph クラスタサービスのコンポーネントが含まれます。Shared File Systems サービスの CephFS NFS ドライバーは、NFS-Ganesha ゲートウェイを使用して NFSv4 プロトコルによる CephFS ファイル共有へのアクセスを提供します。Ceph MDS サービスは、ファイルシステムのディレクトリーおよびファイル名を RADOS クラスタに保管されるオブジェクトにマッピングします。NFS ゲートウェイは、Ceph 等の異なるストレージバックエンドを使用して NFS ファイル共有を提供します。NFS-Ganesha サービスは、Ceph サービスと共にコントローラーノード上で実行されます。

インスタンスは、少なくとも 2 つの NIC と共に起動します。1 つの NIC はプロジェクトのルーターに接続され、2 つ目の NIC は StorageNFS ネットワークに接続され、そこから直接 NFS-Ganesha ゲートウェイに接続されます。インスタンスは、NFS プロトコルを使用してファイル共有をマウントします。Ceph OSD ノードがホストする CephFS ファイル共有は、NFS ゲートウェイを通じて提供されます。

NFS-Ganesha により、ユーザーのインスタンスは MDS および他の Ceph サービスに直接アクセスしなくなるので、セキュリティが向上します。インスタンスは、Ceph デーモンに直接アクセスしません。



7.2. CEPH サービスとクライアントアクセス

Ceph がオブジェクトストレージおよびブロックストレージを提供する際にデプロイされるモニター、OSD、Rados Gateway (RGW)、およびマネージャーサービスに加えて、CephFS には Ceph メタデータサービス (MDS) が必要です。また、NFS プロトコルを使用するネイティブ CephFS へのゲートウェイとして NFS-Ganesha サービスが必要です。ユーザーが直接アクセスできるオブジェクトストレージの場合には、RGW サービスもデプロイされます。ゲートウェイは Ceph パブリックネットワークにアクセスするために CephFS クライアントを実行し、エンドユーザーではなくシステムの管理下に置かれます。

NFS-Ganesha は、Ceph パブリックネットワークおよび新たな分離ネットワーク (StorageNFS) の両方とインターフェイスする専用のコンテナ内で実行されます。Red Hat OpenStack Platform (RHOSP) director のコンポーザブルネットワーク機能が、この分離ネットワークをデプロイしてコントローラーノードに接続します。クラウド管理者はネットワークを Networking (neutron) プロバイダーネットワークとして設定することができます。

NFS-Ganesha は Ceph パブリックネットワークを通じて CephFS にアクセスし、StorageNFS ネットワーク上のアドレスを使用してその NFS サービスをバインドします。

NFS 共有にアクセスするためには、StorageNFS ネットワークに接続される新たな NIC と共に、ユーザー仮想マシン (Compute (nova) インスタンス) をプロビジョニングします。CephFS ファイル共有のエクスポート場所は、StorageNFS ネットワーク上の NFS-Ganesha サーバーの仮想 IP を使用する標準的な NFS IP:<path> タプルとして表示されます。ネットワークはユーザー仮想マシンの IP アドレスを使用して、NFS 共有でのアクセス制御を行います。

Networking (neutron) セキュリティーグループが、プロジェクト 1 に属するユーザー仮想マシンが StorageNFS ネットワークを通じてプロジェクト 2 に属するユーザー仮想マシンにアクセスするのを防ぎます。プロジェクトは同じ CephFS ファイルシステムを共有しますが、ユーザー仮想マシンはエクスポートツリー下のファイル (/path/to/share1/..., /path/to/share2/...) にしかアクセスできないので、プロジェクトデータパスの分離が確保されます。

7.3. NFS バックエンドに CEPHFS を使用した SHARED FILE SYSTEMS サービスの耐障害性

Red Hat OpenStack Platform (RHOSP) director が Ceph サービスデーモンを起動すると、これらのデーモンは自己の高可用性 (HA) 状態を管理し、一般的に、これらのデーモン用に複数のインスタンスが実行されます。これとは対照的に、本リリースでは、ファイル共有を提供することのできる NFS-Ganesha 用インスタンスは、常に 1 つだけです。

NFS バックエンドに CephFS を使用したファイル共有では、データパスに単一障害点が生じるのを避けるために、NFS-Ganesha はアクティブ/パッシブ設定 (Pacemaker-Corosync クラスタが管理) の RHOSP コントローラーノード上で実行されます。NFS-Ganesha は、複数のコントローラーノードに渡って仮想サービス IP アドレスを持つ仮想サービスとして機能します。

コントローラーノードに障害が発生した、あるいは特定のコントローラーノード上のサービスに障害が発生し、そのノードで復帰できない場合には、Pacemaker-Corosync は同じ仮想 IP アドレスを使用して新たな NFS-Ganesha インスタンスを別のコントローラーノード上で起動します。既存クライアントのマウントはファイル共有のエクスポート場所の仮想 IP アドレスを使用するので、これらのマウントは維持されます。

デフォルトの NFS マウントオプション設定および NFS 4.1 以降を使用している場合には、障害発生後に TCP 接続がリセットされ、クライアントが再接続されます。フェイルオーバー中は I/O 操作が一時的に応答しなくなりますが、機能は失われません。アプリケーション I/O も応答しなくなりますが、フェイルオーバーが完了すると処理が再開されます。

最大 90 秒の猶予期間 (クライアントがロックを再要求するのをサーバーが待機する期間) が経過するまで、新規接続や新たなロック状態などは拒否されます。NFS-Ganesha はクライアントの一覧を維持し、すべてのクライアントがロックを再要求すると、その時点で猶予期間を終了します。



注記

猶予期間のデフォルト値は 90 秒です。この値を変更するには、NFSv4 **Grace_Period** 設定オプションを編集します。

7.4. NFS-GANESHA インストールによる CEPHFS

NFS バックエンドに Ceph File System (CephFS) を使用する典型的な Red Hat OpenStack Platform (RHOSP) 環境は、以下の要素で設定されます。

- コンテナ化された Ceph メタデータサーバー (MDS)、Ceph モニター (MON)、manila、および NFS-Ganesha サービスを実行する OpenStack コントローラーノード。これらのサービスのいくつかは、同じノードに共存する場合や、専用のノードを持つ場合があります。

- Ceph ストレージノード上で動作するコンテナ化されたオブジェクトストレージデーモン (OSD) を持つ Ceph ストレージクラスター
- NFS 共有のプロビジョニング用に、プロジェクトから NFS-Ganesha サービスへのアクセスを提供する StorageNFS 分離ネットワーク



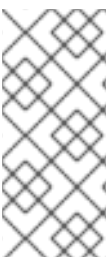
重要

NFS バックエンドに CephFS を使用した Shared File Systems サービス (manila) は、Manila CSI を使用した Red Hat OpenShift Container Platform へのファイル共有の提供を完全にサポートします。このソリューションは、大規模なデプロイメントを対象としていません。重要な推奨事項は、<https://access.redhat.com/articles/6667651> を参照してください。

Shared File Systems サービス (manila) の提供する API により、プロジェクトはファイルシステムの共有を要求することができます。これは、ドライバーモジュールにより実施されます。Red Hat CephFS のドライバー **manila.share.drivers.cephfs.driver.CephFSDriver** により、Shared File Systems サービスを CephFS バックエンドとして使用することができます。RHOSP director は、NFS-Ganesha ゲートウェイをデプロイするようにドライバーを設定します。これにより、NFS 4.1 プロトコルを使用して CephFS ファイル共有が提供されます。

RHOSP director を使用してオーバークラウドに CephFS バックエンドを持つ Shared File Systems サービスをデプロイすると、heat テンプレートで定義された必要なストレージネットワークが自動的に作成されます。ネットワークプランニングに関する詳しい情報は、**Director のインストールと使用方法のオーバークラウドネットワーク** を参照してください。

ノードの **/etc/manila/manila.conf** ファイルを編集して Shared File Systems サービスを手動で設定することができますが、今後のオーバークラウド更新時に、RHOSP director はあらゆる設定を上書きすることができます。Shared File Systems のバックエンドを設定する場合には、director を使用する方法を推奨します。RHOSP director を使用して、ストレージトラフィック用の追加の StorageNFS ネットワークを作成します。



注記

director により設定されていない外部にデプロイされた Ceph クラスターに NFS バックエンドに CephFS を使用する設定を追加する設定はサポートの対象です。現在、director では CephFS バックエンドを1つだけ定義できます。詳細は、オーバークラウドと **既存の Red Hat Ceph Storage クラスターとオーバークラウドの統合** の **オーバークラウドと Ceph Storage の統合** を参照してください。

7.5. NFS-GANESHA のインストール要件による CEPHFS

NFS バックエンドに CephFS を使用する設定は、Red Hat OpenStack Platform (RHOSP) バージョン 13 以降、完全にサポートされています。RHOSP 17.0 以降では、NFS バックエンドに CephFS を使用した RHOSP Shared File Systems サービスは、Red Hat Ceph Storage バージョン 5.2 以降との組み合わせがサポート対象の設定です。システムにインストールされている Ceph Storage のバージョンを確認する方法についての詳細は、[What are the Red Hat Ceph Storage releases and corresponding Ceph package versions?](#) を参照してください。

前提条件

- デフォルトの動作と同様に、Shared File Systems サービスをコントローラーノードにインストールする。

- NFS-Ganesha ゲートウェイサービスをコントローラーノードの Pacemaker クラスターにインストールする。
- 1つの CephFS バックエンドのインスタンスだけが Shared File Systems サービスを使用するように設定する。その他の CephFS 以外のバックエンドは、1つの CephFS バックエンドと共に使用することができます。

7.6. ファイル共有

OpenStack Shared File Systems サービス (manila)、Ceph File System (CephFS)、および NFS バックエンドに Ceph を使用する設定では、ファイル共有の取り扱いが異なります。

Shared File Systems サービスが提供するそれぞれのファイル共有は、個別のファイルシステムの名前空間であり、また特定のサイズを持つストレージのユニットです。元来、共有ファイルシステムのストレージでは、複数のクライアントが任意のファイル共有に接続してデータの読み取りおよび書き込みが可能です。クライアントが接続できるためには、Shared File Systems サービスのアクセス制御 API を通じて各クライアントにファイル共有へのアクセス権限を付与する必要があります。

CephFS では、ファイル共有は、特定のストレージプールまたは名前空間をポイントする、クォータおよびレイアウトが定義されたディレクトリーとして扱われます。CephFS のクォータは、ディレクトリーのサイズを Shared File Systems サービスが作成するファイル共有のサイズに制限します。クライアントの IP アドレスを指定して、NFS 共有を介した CephFS へのアクセスを提供します。

NFS バックエンドに CephFS を使用する設定では、ファイル共有のプロビジョニングおよびアクセスに NFS プロトコルが使用されます。NFS プロトコルはセキュリティーにも対応します。

7.7. カスタムロールファイルの生成

セキュリティーのために、NFS を介して CephFS を使用する場合は、分離されたネットワークを介してのみ Ceph NFS サーバーにアクセスできるように、NFS トラフィックを別のネットワークに分離します。デプロイ担当者は、分離されたネットワークをクラウド内の選択したプロジェクトグループに制限できます。Red Hat OpenStack (RHOSP) director には、専用の StorageNFS ネットワークをデプロイするためのサポートが付属しています。StorageNFS ネットワークを設定して使用するには、カスタムコントローラーロールが必要です。

重要

NFS トラフィック用の分離されたネットワークの作成を省略することができます。ただし、信頼できないクライアントが存在する実稼働環境用のデプロイメントで StorageNFS ネットワークを省略した場合、director は、外部ネットワークなどの分離されていない共有ネットワーク上の Ceph NFS サーバーに接続できます。分離されていない共有ネットワークは、通常、クラウド内のすべてのユーザープライベートネットワークにルーティングできます。NFS サーバーが分離されていないネットワーク上にある場合、クライアント IP アクセスルールを適用して共有ファイルシステムサービス (manila) 共有へのアクセスを制御することはできません。ユーザーは、汎用 **0.0.0.0/0** IP を使用して共有へのアクセスを許可する必要があります。一般的な IP のため、エクスポートパスを見つけた人は誰でも共有をマウントできます。

ControllerStorageNFS カスタムロールファイルを使用して、隔離された StorageNFS ネットワークを設定します。このロールファイルは、デフォルトの **Controller.yaml** ロールファイルに似ていますが、**StorageNFS** ネットワークが追加されています。

```
[stack@undercloud ~]$ cd /usr/share/openstack-tripleo-heat-templates/roles
[stack@undercloud roles]$ diff Controller.yaml ControllerStorageNfs.yaml
```

```
16a17
> - StorageNFS
50a45
```

openstack overcloud roles generate コマンドにより、**-o** 以降に指定したサービスが含まれるカスタム **roles_data.yaml** ファイルが作成されます。以下の例では、作成される **roles_data.yaml** ファイルには、**ControllerStorageNfs**、**Compute**、および **CephStorage** のサービスが含まれます。

openstack overcloud roles generate コマンドの詳細情報は、**Director** のインストールおよび使用方法の [コンポーザブルサービスとカスタムロール](#) を参照してください。



注記

既存の **roles_data.yaml** ファイルがある場合には、それを変更して設定ファイルに **ControllerStorageNfs**、**Compute**、および **CephStorage** サービスを追加します。

手順

1. アンダークラウドホストに **stack** ユーザーとしてログインします。
2. **stackrc** アンダークラウド認証情報ファイルを入手します。

```
$ source ~/stackrc
```

3. **roles_data.yaml** ファイルを作成します。

```
[stack@undercloud ~]$ openstack overcloud roles generate \
--roles-path /usr/share/openstack-tripleo-heat-templates/roles \
-o /home/stack/roles_data.yaml ControllerStorageNfs Compute CephStorage
```

7.8. 更新された環境のデプロイ

環境をデプロイする準備が整ったら、NFS-Ganesha と共に CephFS を実行するのに必要なカスタム環境およびロールを指定して、**openstack overcloud deploy** コマンドを使用します。

オーバークラウドのデプロイコマンドでは、その他の必要なオプションに加えて以下のオプションを指定します。

機能	オプション	補足情報
network_data_ganesha.yaml を使用して新たな StorageNFS ネットワークを追加する。	-n /usr/share/openstack-tripleo-heat-templates/network_data_ganesha.yaml	NFS トラフィックを別のネットワークに分離したくない場合は、このオプションを省略できます。
前のセクションで作成した roles_data.yaml ファイルにより定義したカスタムロールを追加する。	-r /home/stack/roles_data.yaml	NFS トラフィックを別のネットワークに分離したくない場合は、このオプションを省略できます。

機能	オプション	補足情報
Ceph デーモンをデプロイします。	-e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/cephadm.yaml	director を使用した Red Hat Ceph Storage および Red Hat OpenStack Platform のデプロイメントの オーバークラウドデプロイメントの開始
ceph-mds.yaml を使用して Ceph メタデータサーバーをデプロイする。	-e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/ceph-mds.yaml	director を使用した Red Hat Ceph Storage および Red Hat OpenStack Platform のデプロイメントの オーバークラウドデプロイメントの開始
CephFS-NFS バックエンドを使用して Shared File Systems サービス (manila) をデプロイします。director と共に NFS-Ganesha を設定する。	-e /usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsganesha-config.yaml	manila-cephfsganesha-config.yaml 環境ファイル

以下は、CephFS を使用するための NFS-Ganesha、Ceph Storage クラスター、Ceph MDS、および StorageNFS 分離ネットワークをデプロイするオプションを指定した **openstack overcloud deploy** コマンドの例です。

```
[stack@undercloud ~]$ openstack overcloud deploy \
--templates /usr/share/openstack-tripleo-heat-templates \
-n /usr/share/openstack-tripleo-heat-templates/network_data_ganesha.yaml \
-r /home/stack/roles_data.yaml \
-e /home/stack/containers-default-parameters.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml \
-e /home/stack/network-environment.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/cephadm.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/ceph-mds.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsganesha-config.yaml
```

openstack overcloud deploy コマンドについての詳しい情報は、[Director Installation and Usage](#) の [オーバークラウドのプロビジョニングおよびデプロイ](#) を参照してください。

7.8.1. StorageNFS および network_data_ganesha.yaml ファイル

コンポーザブルネットワークを使用して、カスタムネットワークを定義して、それを任意のロールに割り当てます。標準の **network_data.yaml** ファイルを使用する代わりに、**network_data_ganesha.yaml** ファイルを使用して StorageNFS コンポーザブルネットワークを設定することができます。これらのロールは、共に **/usr/share/openstack-tripleo-heat-templates** ディレクトリーから利用可能です。

重要

Storage NFS ネットワークを定義しない場合、director はデフォルトで外部ネットワークに設定されます。外部ネットワークは、テストおよびプロトタイプ環境で便利ですが、外部ネットワークのセキュリティは実稼働環境には十分ではありません。たとえば、外部ネットワークで NFS サービスを公開する場合、サービス拒否 (DoS) 攻撃により、NFS 共有のコンシューマーだけでなく、すべてのクラウドユーザーへのコントローラー API アクセスが中断される可能性があります。これとは対照的に、専用の Storage NFS ネットワークに NFS サービスをデプロイすると、DoS 攻撃で、ク

ラウド内の NFS 共有だけを対象にすることができてしまいます。潜在的なセキュリティリスクに加えて、NFS サービスを外部ネットワークにデプロイする際に、ファイル共有へのアクセス制御に追加のルーティング設定が必要になります。ただし、Storage NFS ネットワークでは、ネットワーク上でクライアント IP アドレスを使用して、正確なアクセス制御を行うことができます。

network_data_ganesha.yaml ファイルには、StorageNFS 分離ネットワークを定義する追加のセクションが含まれます。デフォルトの設定がほとんどのインストールで機能しますが、YAML ファイルを編集してご自分のネットワーク設定 (VLAN ID、サブネット、その他の設定など) を追加する必要があります。

```
name: StorageNFS
enabled: true
vip: true
name_lower: storage_nfs
vlan: 70
ip_subnet: '172.17.0.0/20'
allocation_pools: [{'start': '172.17.0.4', 'end': '172.17.0.250'}]
ipv6_subnet: 'fd00:fd00:fd00:7000::/64'
ipv6_allocation_pools: [{'start': 'fd00:fd00:fd00:7000::4', 'end': 'fd00:fd00:fd00:7000::ffff'}]
```

コンポーザブルネットワークについての詳しい情報は、**Director のインストールおよび使用方法** の [コンポーザブルネットワーク](#) を参照してください。

7.8.2. CephFS-NFS バックエンド環境ファイル

CephFS-NFS バックエンドを定義するための環境ファイル **manila-cephfsganesha-config.yaml** は、アンダークラウドノードのパス **/usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsganesha-config.yaml** にあります。

manila-cephfsganesha-config.yaml 環境ファイルには、Shared File Systems サービス (manila) のデプロイメントに関する設定が含まれます。バックエンドのデフォルト設定は、ほとんどの環境で機能します。Shared File Systems サービスをデプロイする際に director が使用するデフォルト値を以下の例で示します。

```
[stack@undercloud ~]$ cat /usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsganesha-config.yaml
# A Heat environment file which can be used to enable a
# a Manila CephFS-NFS driver backend.
resource_registry:
  OS::TripleO::Services::ManilaApi: ../deployment/manila/manila-api-container-puppet.yaml
  OS::TripleO::Services::ManilaScheduler: ../deployment/manila/manila-scheduler-container-puppet.yaml
  # Only manila-share is pacemaker managed:
  OS::TripleO::Services::ManilaShare: ../deployment/manila/manila-share-pacemaker-puppet.yaml
  OS::TripleO::Services::ManilaBackendCephFs: ../deployment/manila/manila-backend-cephfs.yaml
  # ceph-nfs (ganesha) service is installed and configured by Director
  # but it's still managed by pacemaker
  OS::TripleO::Services::CephNfs: ../deployment/cephadm/ceph-nfs.yaml

parameter_defaults:
  ManilaCephFSBackendName: cephfs ❶
  ManilaCephFSDriverHandlesShareServers: false ❷
  ManilaCephFSCephFSAuthId: 'manila' ❸
  # manila cephfs driver supports either native cephfs backend - 'CEPHFS'
```

```
# (users mount shares directly from ceph cluster), or nfs-ganesha backend -
# 'NFS' (users mount shares through nfs-ganesha server)
ManilaCephFSCephFSProtocolHelperType: 'NFS'
```

parameter_defaults ヘッダーから設定が始まります。**resource_registry** に設定されたデフォルト値を上書きするには、この **manila-cephfsganesha-config.yaml** 環境ファイルをローカルの環境ファイルディレクトリー **/home/stack/templates/** にコピーし、お使いの環境で必要なパラメーターの設定を編集します。これには、CephFS バックエンドのデフォルトを設定する

OS::Tripleo::Services::ManilaBackendCephFs で定義した値も含まれます。

- 1 **ManilaCephFSBackendName:** CephFS バックエンドの manila 設定の名前を定義します。ここでは、デフォルトのバックエンド名は **cephfs** です。
- 2 **ManilaCephFSDriverHandlesShareServers:** 共有用サーバーのライフサイクルをコントロールします。**false** に設定すると、ドライバーはライフサイクルを処理しません。サポートされるオプションはこれだけです。
- 3 **ManilaCephFSCephFSAuthId:** Ceph クラスターにアクセスするために director が **manila** サービス用に作成する Ceph 認証 ID を定義します。

環境ファイルの詳細は、**Director のインストールと使用方法** の [環境ファイル](#) を参照してください。

第8章 オーバークラウドデプロイメントの開始

サービスの初期設定とカスタマイズが完了したら、オーバークラウドをデプロイします。

8.1. オーバークラウドデプロイメントの開始

オーバークラウドをデプロイして、Red Hat OpenStack Platform (RHOSP) 環境の設定を実装します。

前提条件

- 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。



注記

オーバークラウドのデプロイメント中に Ceph Dashboard を追加する場合は、[10章 Red Hat Ceph Storage Dashboard のオーバークラウドデプロイメントへの追加](#) を参照してください。

手順

openstack overcloud deploy コマンドを使用して、オーバークラウドをデプロイします。すべてのコマンド引数の完全なリストについては、コマンドラインインターフェイスリファレンスの **openstack overcloud deploy** を参照してください。

コマンドの使用例を次に示します。

```
$ openstack overcloud deploy --templates -r /home/stack/templates/roles_data_custom.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/cephadm.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/ceph-mds.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/cinder-backup.yaml \
-e /home/stack/templates/storage-config.yaml \
-e /home/stack/templates/deployed-ceph.yaml \
-e /home/stack/templates/networks-deployed.yaml \
-e /home/stack/templates/deployed-metal.yaml \
-e /home/stack/templates/deployed-vips.yaml \
--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。
- r /home/stack/templates/roles_data_custom.yaml**
 - カスタマイズされたロール定義ファイルを指定します。
- e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/cephadm.yaml**
 - 以前にデプロイされた Ceph Storage クラスターをファイナライズするように、director を設定します。この環境ファイルは、デフォルトで RGW をデプロイします。また、プール、キー、およびデーモンも作成します。RGW またはオブジェクトストレージをデプロイしな

い場合は、「[Red Hat OpenStack Platform オブジェクトストレージのデプロイメントオプション](#)」で説明されているオプションを参照してください。

- **-e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/ceph-mds.yaml**
 - 「[Ceph Metadata Server の有効化](#)」で説明されているように、Ceph メタデータサーバーを有効にします。
- **-e /usr/share/openstack-tripleo-heat-templates/environments/cinder-backup.yaml**
 - 「[Ceph を使用するための Block Storage Backup Service の設定](#)」で説明されているように、Block Storage Backup サービス (**cinder-backup**) を有効にします。
- **-e /home/stack/templates/storage-config.yaml**
 - 「[カスタム環境ファイルの設定](#)」で説明されているように、カスタム Ceph Storage 設定を含む環境ファイルを追加します。
- **-e /home/stack/templates/deployed-ceph.yaml**
 - 以前に実行した **openstack overcloud ceph deploy** コマンドによる出力として、Ceph クラスター設定を含む環境ファイルを追加します。
- **-e /home/stack/templates/networks-deployed.yaml**
 - **openstack overcloud network provision** による出力として、Ceph クラスターのネットワーク設定を含む環境ファイルを追加します。
- **-e /home/stack/templates/deployed-metal.yaml**
 - **openstack overcloud node provision** による出力として、Ceph クラスターノード設定を含む環境ファイルを追加します。
- **-e /home/stack/templates/deployed-vips.yaml**
 - **openstack overcloud network vip provision** による出力として、Ceph クラスターのネットワーク VIP 設定を含む環境ファイルを追加します。
- **--ntp-server pool.ntp.org**
 - NTP サーバーを設定します。

第9章 DIRECTOR を使用した多様なワークロードのパフォーマンス層の定義

Red Hat OpenStack Platform (RHOSP) director は、Red Hat Ceph Storage パフォーマンス層をデプロイします。Ceph Storage CRUSH ルールを **CephPools** パラメーターと組み合わせて、デバイスクラスの機能を使用します。これにより、さまざまなパフォーマンス要件を持つワークロードに対応するために、さまざまな層が構築されます。

たとえば、通常のワークロード用に HDD クラスと、高パフォーマンスのロードのために SSD 上でのみデータを分散する SSD クラスを定義できます。このシナリオでは、新規の Block Storage ポリュームを作成する場合には、HDD または SSD のいずれかのパフォーマンス層を選択することができます。

CRUSH ルールの作成の詳細については、[CRUSH 階層の設定](#) を参照してください。



警告

既存の環境でパフォーマンス層を定義すると、Ceph Storage クラスターでデータが移動する可能性があります。director は、スタックの更新中、**cephadm** を使用します。**cephadm** アプリケーションには、プールが存在し、データが含まれているかどうかを確認するロジックがありません。プールに関連付けられているデフォルトの CRUSH ルールを変更すると、データが移動します。プールに大量のデータが含まれている場合、そのデータは移動されます。

ノードの追加または削除に関する支援または推奨事項が必要な場合は、Red Hat サポートにお問い合わせください。

Ceph Storage は、ディスクタイプを自動的に検出し、対応するデバイスクラス HDD、SSD、または NVMe に割り当てます。Linux カーネルによって公開されるハードウェアプロパティに基づいています。

前提条件

- 新しいデプロイメントの場合は、Red Hat Ceph Storage (RHCS) バージョン 5.2 以降を使用してください。

9.1. パフォーマンス層の設定

異なる Red Hat Ceph Storage パフォーマンス層をデプロイするには、CRUSH マップの詳細が含まれる新規の環境ファイルを作成し、これをデプロイメントコマンドに追加します。Director は、この機能の特定のパラメーターを公開しませんが、想定される **tripleo-ansible** 変数を生成できます。



注記

パフォーマンス層の設定は、CRUSH 階層と組み合わせることができます。CRUSH ルールの作成は、[Configuring CRUSH hierarchies](#) を参照してください。

手順の例では、各 Ceph Storage ノードには OSD が 3 つ含まれます。**sdb** および **sdc** は動作中のディスクで、**sdc** は SSD です。Ceph は正しいディスク種別を自動的に検出します。次に、HDD および SSD の 2 つの CRUSH ルールを設定し、2 つのデバイスクラスにそれぞれマッピングします。



注記

HDD ルールはデフォルトで、別のルールでプールを設定しない限り、すべてのプールに適用されます。

最後に、**fastpool** と呼ばれる追加のプールを作成し、SSD ルールにマッピングします。このプールは、最終的に Block Storage (cinder) バックエンドを通じて公開されます。この Block Storage バックエンドを使用するすべてのワークロードは、パフォーマンスを高速にする場合にのみ SSD によってサポートされます。これは、データまたはボリュームから起動 (boot from volume) のいずれかに活用できます。

警告

既存の環境でパフォーマンス層を定義すると、Ceph クラスター内で大量のデータが移動する場合があります。スタックの更新時に director がトリガーする **cephadm** には、プールが Ceph クラスターにすでに定義されているかどうかや、データが含まれるかどうかを確認するロジックはありません。つまり、プールに関連付けられたデフォルトの CRUSH ルールを変更すると、データの移動が行われるため、既存の環境でパフォーマンス層を定義することは危険となる可能性があります。ノードの追加または削除に関する支援または推奨事項が必要な場合は、Red Hat サポートにお問い合わせください。

手順

1. アンダークラウドノードに **stack** ユーザーとしてログインします。
2. Ceph config パラメーターおよびデバイスクラス変数を含む環境ファイル (`/home/stack/templates/ceph-config.yaml` 等) を作成します。あるいは、既存の環境ファイルに以下の設定を追加することができます。
3. **CephCrushRules** パラメーターを追加します。**crush_rules** パラメーターには、定義した各クラスまたは Ceph が自動検出する各クラスにルールが含まれている必要があります。新しいプールを作成する際にルールが指定されていない場合は、Ceph が使用するルールがデフォルトとして選択されます。

```
CephCrushRules:
  crush_rules:
    - name: HDD
      root: default
      type: host
      class: hdd
      default: true
    - name: SSD
      root: default
      type: host
      class: ssd
      default: false
```

4. **CephPools** パラメーターを追加します。
 - **rule_name** パラメーターを使用して、デフォルトのルールを使用しない各プールの層を指定します。以下の例では、**fastpool** プールは、fast tier として設定された SSD デバイスクラスを使用して Block Storage ボリュームを管理します。
 - **CinderRbdExtraPools** パラメーターを使用して、**fastpool** を Block Storage バックエンドとして設定します。

```
CephPools:
- name: fastpool
  rule_name: SSD
  application: rbd
CinderRbdExtraPools: fastpool
```

5. 以下の例を使用して、環境ファイルに正しい値が含まれることを確認します。

```
parameter_defaults:
  crush_rules:
    - name: HDD
      root: default
      type: host
      class: hdd
      default: true
    - name: SSD
      root: default
      type: host
      class: ssd
      default: false
  CinderRbdExtraPools: fastpool
  CephPools:
    - name: fastpool
      rule_name: SSD
      application: rbd
```

6. **openstack overcloud deploy** コマンドで新規の環境ファイルを指定します。

```
$ openstack overcloud deploy \
--templates \
...
-e <other_overcloud_environment_files> \
-e /home/stack/templates/ceph-config.yaml \
...
```

<other_overcloud_environment_files> をデプロイメントに追加する他の環境ファイルの一覧に置き換えます。

重要

既存の Ceph クラスターに環境ファイルを適用すると、既存の Ceph プールは新しいルールで更新されません。このため、デプロイメントの完了後に以下のコマンドを入力して、指定したプールにルールを設定する必要があります。

```
$ ceph osd pool set <pool> crush_rule <rule>
```

- <pool> を新しいルールを適用するプールの名前に置き換えます。
- <rule> を **crush_rules** パラメーターで指定したルール名の1つに置き換えます。

このコマンドを使用してルールを変更するたびに、既存のエントリーを更新するか、既存のテンプレートの **CephPools** パラメーターに新しいエントリーを追加します。

```
CephPools:
- name: <pool>
  rule_name: <rule>
  application: rbd
```

9.2. CRUSH ルールとプールの確認

CRUSH ルールとプールの設定を確認します。

警告

既存の環境でパフォーマンス層を定義すると、Ceph クラスター内で大量のデータが移動する場合があります。スタックの更新時に director がトリガーする **tripleo-ansible** には、プールが Ceph クラスターにすでに定義されているかどうかや、データが含まれるかどうかを確認するロジックはありません。つまり、プールに関連付けられたデフォルトの CRUSH ルールを変更すると、データの移動が行われるため、既存の環境でパフォーマンス層を定義することは危険となる可能性があります。ノードの追加または削除に関する支援または推奨事項が必要な場合は、Red Hat サポートにお問い合わせください。

手順

1. **tripleo-admin** ユーザーとしてオーバークラウドのコントローラーノードにログインします。
2. OSD 層が正常に設定されていることを確認するには、以下のコマンドを入力します。

```
$ sudo cephadm shell ceph osd tree
```

3. 作成されるツリービューで、各 OSD に設定したデバイスクラスが **CLASS** コラムに正しく表示されることを確認します。
4. また、以下のコマンドで、OSD がデバイスクラスに正しく割り当てられていることを確認します。

```
$ sudo cephadm shell ceph osd crush tree --show-shadow
```

5. 作成された階層を以下のコマンドによる結果と比較し、ルールごとに同じ値が適用されることを確認します。

```
$ sudo cephadm shell ceph osd crush rule dump <rule_name>
```

- <rule_name> を、チェックするルールの名前に置き換えます。
6. 作成したルール名と ID が、デプロイメント中に使用した **crush_rules** パラメーターに準じて正しいことを確認します。

```
$ sudo cephadm shell ceph osd crush rule dump | grep -E "rule_(id|name)"
```

7. Ceph プールが、ステップ 3 で取得した正しい CRUSH ルール ID に関連付けられていることを確認します。

```
$ sudo cephadm shell -- ceph osd dump | grep pool
```

8. 各プールについて、ルール ID が想定するルール名と一致することを確認してください。

第10章 RED HAT CEPH STORAGE DASHBOARD のオーバークラウドデプロイメントへの追加

Red Hat Ceph Storage Dashboard はデフォルトで無効になっていますが、Red Hat OpenStack Platform (RHOSP) director を使用してオーバークラウドで有効にすることができます。Ceph Dashboard は組み込みの Web ベースの Ceph 管理および監視アプリケーションであり、Ceph クラスター内のさまざまな側面およびオブジェクトを管理します。Red Hat Ceph Storage Dashboard は、以下のコンポーネントで設定されています。

- Ceph Dashboard Manager モジュール: ユーザーインターフェイスを提供し、プラットフォームフロントエンド Grafana が組み込まれています。
- Prometheus: モニターリングプラグイン
- Alertmanager: アラートを Dashboard に送信します。
- Node Exporters: Ceph クラスターデータを Dashboard にエクスポートします。



注記

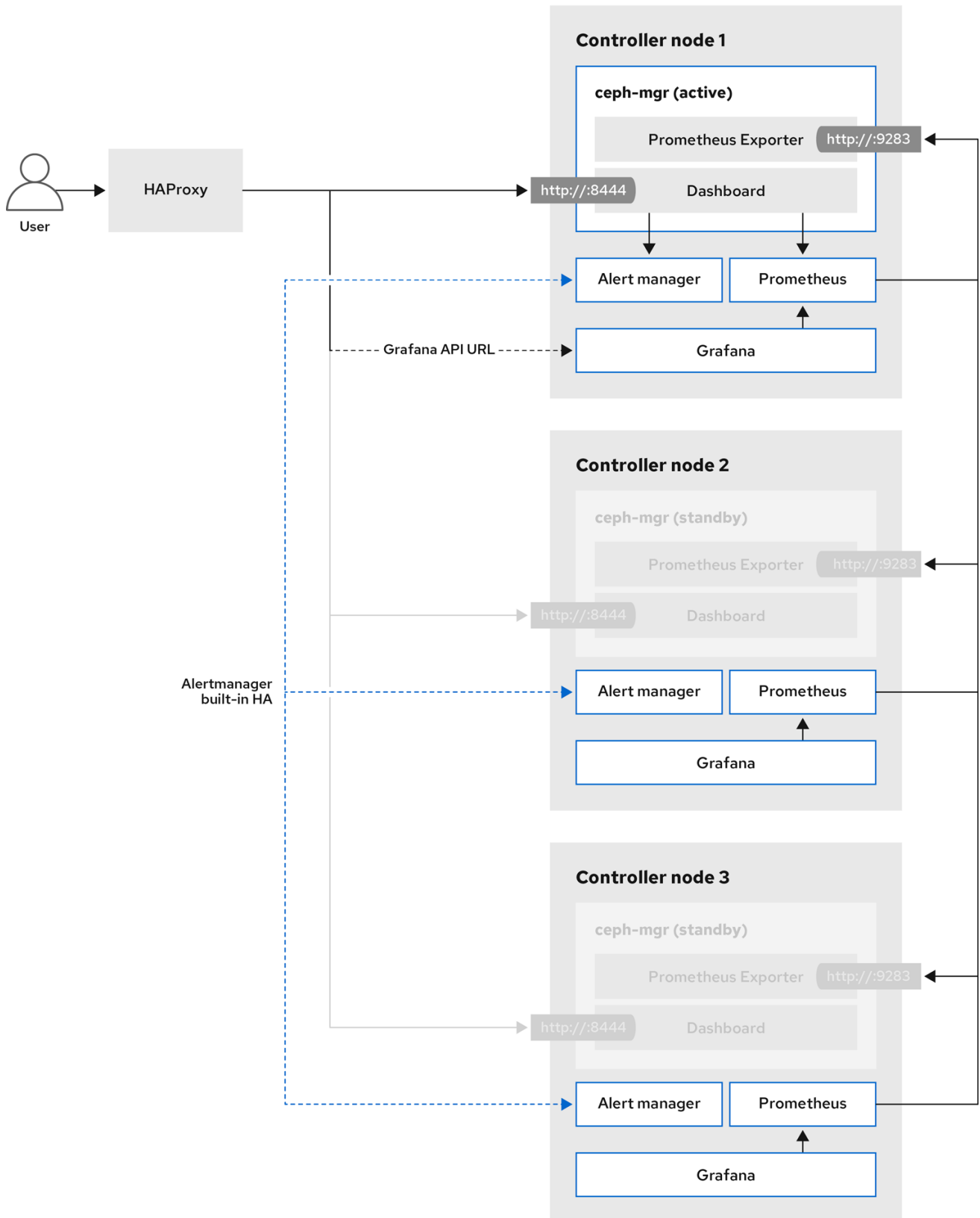
この機能は、Ceph Storage 4.1 以降でサポートされます。システムにインストールされている Ceph Storage のバージョンを確認する方法についての詳細は、[What are the Red Hat Ceph Storage releases and corresponding Ceph package versions?](#) を参照してください。



注記

Red Hat Ceph Storage Dashboard は、常に他の Ceph Manager コンポーネントと同じノード上に配置されます。

以下の図は、Red Hat OpenStack Platform 上の Ceph Dashboard のアーキテクチャーを示しています。



89_Ceph_0520

Dashboard およびその機能と制限についての詳細は、Red Hat Ceph Storage Dashboard Guideの [Dashboard features](#) を参照してください。

10.1. CEPH DASHBOARD における TLS EVERYWHERE

Dashboard フロントエンドは、TLS everywhere フレームワークと完全に統合されています。必要な環

境ファイルがあり、そのファイルが `overcloud deploy` コマンドに含まれている場合は、TLS everywhere を有効にすることができます。これにより、Grafana と Ceph Dashboard の両方の証明書要求がトリガーされ、生成された証明書とキーファイルがオーバークラウドのデプロイメント時に **cephadm** に渡されます。Dashboard およびその他の RHOSP サービス向けに TLS を有効化する手順と詳細は、**Advanced Overcloud Customization** ガイドの以下の章を参照してください。



注記

Ceph Dashboard に到達するポートは、TLS-everywhere コンテキストでも同じになります。

10.2. CEPH DASHBOARD に必要なコンテナの追加

Ceph Dashboard テンプレートをオーバークラウドに追加する前に、**containers-prepare-parameter.yaml** ファイルを使用して必要なコンテナを追加する必要があります。コンテナイメージを準備するために **containers-prepare-parameter.yaml** ファイルを生成するには、以下の手順を実行します。

手順

1. アンダークラウドホストに **stack** ユーザーとしてログインします。
2. デフォルトのコンテナイメージ準備ファイルを生成します。

```
$ sudo openstack tripleo container image prepare default \
  --local-push-destination \
  --output-env-file containers-prepare-parameter.yaml
```

3. **containers-prepare-parameter.yaml** ファイルを編集し、要件に合わせて変更を加えます。以下の **containers-prepare-parameter.yaml** ファイルのサンプルには、Grafana、Prometheus、Alertmanager、および Node Exporter などの Dashboard サービスに関連するイメージの場所およびタグが含まれます。特定のシナリオに応じて値を編集します。

```
parameter_defaults:
  ContainerImagePrepare:
    - push_destination: true
      set:
        ceph_alertmanager_image: ose-prometheus-alertmanager
        ceph_alertmanager_namespace: registry.redhat.io/openshift4
        ceph_alertmanager_tag: v4.1
        ceph_grafana_image: rhceph-5-dashboard-rhel8
        ceph_grafana_namespace: registry.redhat.io/rhceph
        ceph_grafana_tag: 4
        ceph_image: rhceph-5-rhel8
        ceph_namespace: registry.redhat.io/rhceph
        ceph_node_exporter_image: ose-prometheus-node-exporter
        ceph_node_exporter_namespace: registry.redhat.io/openshift4
        ceph_node_exporter_tag: v4.1
        ceph_prometheus_image: ose-prometheus
        ceph_prometheus_namespace: registry.redhat.io/openshift4
        ceph_prometheus_tag: v4.1
        ceph_tag: latest
```

containers-prepare-parameter.yaml ファイルを使用したレジストリーおよびイメージ設定についての詳細は、[Transitioning to Containerized Services](#) ガイドの [Container image preparation parameters](#) を参照してください。

10.3. CEPH DASHBOARD のデプロイ

Ceph ダッシュボードをデプロイするには、ceph-dashboard 環境ファイルを含めます。



注記

コンポーザブルネットワークを使用して Ceph Dashboard をデプロイする場合は、「[コンポーザブルネットワークを使用した Ceph Dashboard のデプロイ](#)」を参照してください。



注記

Ceph Dashboard の管理ユーザーロールは、デフォルトで読み取り専用モードに設定されています。Ceph Dashboard のデフォルトの管理モードを変更するには、「[デフォルト権限の変更](#)」を参照してください。

手順

1. アンダークラウドノードに **stack** ユーザーとしてログインします。
2. オプション: Ceph ダッシュボードネットワークは、デフォルトでプロビジョニングネットワークに設定されます。Ceph ダッシュボードをデプロイし、別のネットワーク経由でアクセスする場合は、環境ファイル (例: **ceph_dashboard_network_override.yaml**) を作成します。**CephDashboardNetwork** を既存のオーバークラウドでルーティングされたネットワークの1つ (例: **external**) に設定します。

```
parameter_default
ServiceNetMap:
  CephDashboardNetwork: external
```



重要

初期デプロイメント後は、別のネットワークから Ceph Dashboard にアクセスするための **CephDashboardNetwork** 値の変更はサポートされません。

3. 以下の環境ファイルを **openstack overcloud deploy** コマンドに含めます。デプロイメントの一部であるすべての環境ファイルと、デフォルトのネットワークを変更する場合は **ceph_dashboard_network_override.yaml** ファイルを含めます。

```
$ openstack overcloud deploy \
--templates \
-e <overcloud_environment_files> \
-e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/cephadm.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/ceph-
dashboard.yaml
-e ceph_dashboard_network_override.yaml
```

<overcloud_environment_files> をデプロイメントに追加する環境ファイルの一覧に置き換えます。

結果

これにより、デプロイメントは grafana、prometheus、alertmanager、および node-exporter コンテナが含まれる外部スタックを設定します。Ceph Dashboard Manager モジュールは、このスタックのバックエンドで、grafana レイアウトを組み込むことで、Ceph クラスター固有のメトリックをエンドユーザーに提供します。

10.4. コンポーザブルネットワークを使用した CEPH DASHBOARD のデプロイ

デフォルトのプロビジョニングネットワークではなく、コンポーザブルネットワークに Ceph Dashboard をデプロイすることができます。これにより、プロビジョニングネットワークに Ceph Dashboard サービスを公開する必要がなくなります。Dashboard をコンポーザブルネットワークにデプロイする際に、別個の承認プロファイルを実装することもできます。

最初にオーバークラウドをデプロイする場合にのみ Dashboard を新規ネットワークに適用することができるので、デプロイ前に使用するネットワークを選択する必要があります。デプロイ前にコンポーザブルネットワークを選択するには、以下の手順を使用します。

手順

1. アンダークラウドに stack ユーザーとしてログインします。
2. Controller 固有のロールを生成して、Dashboard コンポーザブルネットワークを追加します。

```
$ openstack overcloud roles generate -o /home/stack/roles_data_dashboard.yaml
ControllerStorageDashboard Compute BlockStorage ObjectStorage CephStorage
```

結果

- 新しい **ControllerStorageDashboard** ロールは、コマンドの出力として定義される **roles_data.yaml** 内に生成されます。overcloud deploy コマンドを使用する場合には、このファイルをテンプレート一覧に含める必要があります。
注: **ControllerStorageDashboard** ロールには、**CephNFS** も **network_data_dashboard.yaml** も含まれていません。
 - director は、コンポーザブルネットワークを定義するネットワーク環境ファイルを提供します。このファイルのデフォルトの場所は、**/usr/share/openstack-tripleo-heat-templates/network_data_dashboard.yaml** です。overcloud deploy コマンドを使用する場合には、このファイルをオーバークラウドのテンプレート一覧に含める必要があります。
3. **openstack overcloud deploy** コマンドに、デプロイメントに含まれるすべての環境ファイルと共に以下の環境ファイルを追加します。

```
$ openstack overcloud deploy \
--templates \
-r /home/stack/roles_data.yaml \
-n /usr/share/openstack-tripleo-heat-templates/network_data_dashboard.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/network-environment.yaml \
-e <overcloud_environment_files> \
-e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/cephadm.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/ceph-
dashboard.yaml
```

-

<overcloud_environment_files> をデプロイメントに追加する環境ファイルの一覧に置き換えます。

結果

これにより、デプロイメントは grafana、prometheus、alertmanager、および node-exporter コンテナが含まれる外部スタックを設定します。Ceph Dashboard Manager モジュールは、このスタックのバックエンドで、grafana レイアウトを組み込むことで、Ceph クラスター固有のメトリックをエンドユーザーに提供します。

10.5. デフォルト権限の変更

Ceph クラスターの安全な監視用に、Ceph Dashboard の管理ユーザーロールはデフォルトで読み取り専用モードに設定されます。管理ユーザーが Dashboard を使用して Ceph クラスターの要素を変更できるように管理者権限を昇格させるには、**CephDashboardAdminRO** パラメーターを使用してデフォルトの管理者権限を変更します。



警告

完全な権限を持つユーザーは、director が設定する Ceph クラスターの要素を変更する可能性があります。これは、スタックの更新の実行時に、director が設定したオプションと競合する可能性があります。この問題を回避するには、Ceph Dashboard で director の設定オプション (Ceph OSP プール属性など) を変更しないでください。

手順

1. アンダークラウドに **stack** ユーザーとしてログインします。
2. 以下の **ceph_dashboard_admin.yaml** 環境ファイルを作成します。

```
parameter_defaults:
  CephDashboardAdminRO: false
```

3. overcloud deploy コマンドを実行して、既存のスタックを更新し、既存のデプロイメントに含まれるその他すべての環境ファイルと共に作成した環境ファイルを追加します。

```
$ openstack overcloud deploy \
--templates \
-e <existing_overcloud_environment_files> \
-e ceph_dashboard_admin.yml
```

<existing_overcloud_environment_files> を既存のデプロイメントに含まれる環境ファイルの一覧に置き換えます。

10.6. CEPH DASHBOARD へのアクセス

Ceph Dashboard が正常に実行されていることを確認するには、以下の検証手順を実施してアクセスし、Ceph クラスターから表示されるデータが正しいことを確認します。

手順

1. アンダークラウドノードに **stack** ユーザーとしてログインします。
2. Dashboard の管理ログイン認証情報を取得します。

```
[stack@undercloud ~]$ grep tripleo_cephadm_dashboard_admin_password <config-download>/<stack>/cephadm/cephadm-extra-vars-heat.yml
```

3. Ceph Dashboard にアクセスするための仮想 IP アドレスを取得します。

```
[stack@undercloud-0 ~]$ grep tripleo_cephadm_dashboard_frontend_vip <config-download>/<stack>/cephadm/cephadm-extra-vars-ansible.yml
```

4. Web ブラウザーを使用してフロントエンドの仮想 IP をポイントし、Dashboard にアクセスします。director はプロビジョニングネットワーク上で Dashboard を設定して公開するので、取得した仮想 IP を使用して、TCP ポート 8444 の Dashboard に直接アクセスできます。以下の条件を満たしていることを確認します。

- Web クライアントホストは、プロビジョニングネットワークに接続されたレイヤー 2 です。
- プロビジョニングネットワークは適切にルーティングまたはプロキシされ、Web クライアントホストからアクセスできます。これらの条件が満たされていない場合は、SSH トンネルを開いて、オーバークラウド上の Dashboard の仮想 IP に到達することができます。

```
client_host$ ssh -L 8444:<dashboard_vip>:8444 stack@<your undercloud>
```

<dashboard_vip> を取得したコントロールプレーンの仮想 IP の IP アドレスに置き換えます。

5. Dashboard にアクセスするには、Web ブラウザーで <http://localhost:8444> にアクセスし、以下の詳細でログインします。
- **cephadm** が作成するデフォルトのユーザー: **admin**。
 - **<config-download>/<stack>/cephadm/cephadm-extra-vars-heat.yml** のパスワード。

結果

- Ceph Dashboard にアクセスできます。
- Dashboard に表示される数字およびグラフには、CLI コマンドの **ceph -s** が返すのと同じ Ceph クラスターステータスが反映されます。

Red Hat Ceph Storage Dashboard に関する詳細は、[Red Hat Ceph Storage Dashboard Guide](#)を参照してください。

第11章 RED HAT CEPH STORAGE クラスターを管理するためのデプロイメント後の操作

コンテナ化された Red Hat Ceph Storage を使用して Red Hat OpenStack Platform (RHOSP) 環境をデプロイした後、Ceph Storage クラスターを管理するために使用できる操作がいくつかあります。

11.1. 設定オーバーライドの無効化

Ceph Storage クラスターが最初にデプロイされた後、オーバークラウドのデプロイ中、RGW などのサービスをセットアップできるように、クラスターが設定されます。オーバークラウドのデプロイが完了したら、クラスターをスケールアップする場合を除き、director を使用して、クラスター設定を変更しないでください。クラスター設定の変更は、Ceph コマンドを使用して、実行する必要があります。

手順

1. アンダークラウドノードに **stack** ユーザーとしてログインします。
2. ファイル **deploy_ceph.yaml** または環境で使用するファイルを開いて、Ceph Storage クラスター設定を定義します。
3. **ApplyCephConfigOverridesOnUpdate** パラメーターを見つけます。
4. **ApplyCephConfigOverridesOnUpdate** パラメーター値を **false** に変更します。
5. ファイルを保存します。

関連情報

ApplyCephConfigOverridesOnUpdate および **CephConfigOverrides** パラメーターの詳細については、[オーバークラウドのパラメーター](#) を参照してください。

11.2. 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を行い、認証をサポートするスクリプトを作成します。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. 次のコマンドを実行して、ファイルのソースを明らかにします。

```
$ 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ために必要な環境変数が読み込まれます。

2. アンダークラウドとの対話に戻るには、以下のコマンドを実行します。

```
$ source ~/stackrc
```

11.3. RED HAT CEPH STORAGE ノードのモニタリング

オーバークラウドを作成したら、Ceph クラスターのステータスをチェックして、正常に機能していることを確認します。

手順

1. **tripleo-admin** ユーザーとしてコントローラーノードにログインします。

```
$ nova list
$ ssh tripleo-admin@192.168.0.25
```

2. Ceph クラスターの状態を確認します。

```
$ sudo cephadm shell -- ceph health
```

Ceph クラスターに問題がない場合は、上記のコマンドにより、**HEALTH_OK** というレポートが返されます。これは、Ceph クラスターが安全に使用できることを意味します。

3. Ceph monitor サービスを実行するオーバークラウドノードにログインし、Ceph クラスター内の全 OSD のステータスを確認します。

```
$ sudo cephadm shell -- ceph osd tree
```

4. Ceph Monitor クォーラムのステータスを確認します。

```
$ sudo cephadm shell -- ceph quorum_status
```

これにより、クォーラムに参加するモニターとどれがリーダーであるかが表示されます。

5. すべての Ceph OSD が動作中であることを確認します。

```
$ sudo cephadm shell -- ceph osd stat
```

Ceph クラスターのモニタリングの詳細については、**Red Hat Ceph Storage Administration Guide** の [Monitoring a Ceph Storage cluster](#) を参照してください。

11.4. BLOCK STORAGE (CINDER) 種別の新しい CEPH プールへのマッピング

設定手順を完了したら、Block Storage (cinder) を使用して作成した **fastpool** 層にマッピングされた種別を作成して、RHOSP テナントにパフォーマンス層の機能を使用できるようにします。

手順

1. アンダークラウドノードに **stack** ユーザーとしてログインします。
2. source コマンドで **overcloudrc** ファイルを読み込みます。

```
$ source overcloudrc
```

3. Block Storage ボリュームの既存種別を確認します。

```
$ cinder type-list
```

4. 新規の Block Storage ボリューム `fast_tier` を作成します。

```
$ cinder type-create fast_tier
```

5. Block Storage 種別が作成されていることを確認します。

```
$ cinder type-list
```

6. **fast_tier** Block Storage 種別が利用可能な場合は、作成した新しい層の Block Storage ボリュームバックエンドとして **fastpool** を設定します。

```
$ cinder type-key fast_tier set volume_backend_name=tripleo_ceph_fastpool
```

7. 新しい層を使用して、新しいボリュームを作成します。

```
$ cinder create 1 --volume-type fast_tier --name fastdisk
```



注記

Red Hat Ceph Storage のドキュメントには、Ceph Storage クラスターの継続的なメンテナンスと操作に関する追加情報と手順が記載されています。[Product Documentation for Red Hat Ceph Storage 5](#) を参照してください。

第12章 ネイティブ CEPHFS デプロイ後の設定と検証

CephFS 共有を作成し、ユーザーにアクセス権限を付与し、CephFS 共有をマウントする前に、いくつかのデプロイメント後設定タスクを完了する必要があります。

- Networking サービス (neutron) ストレージネットワークを分散データセンターストレージネットワークにマッピングします。
- カスタムロールベースアクセス制御 (RBAC) のみ、ストレージプロバイダーネットワークを信頼済みテナントで利用できるようにします。ストレージプロバイダーネットワークはグローバルに共有しないでください。
- プライベートファイル共有種別を作成します。
- 特定の信頼済みテナントにアクセス権限を付与します。

これらのステップを完了したら、テナントコンピュートインスタンスはネイティブ CephFS 共有を作成し、そのアクセスを許可し、マウントすることができます。

Shared File Systems サービス (manila) のバックエンドとしてネイティブ CephFS をデプロイすると、オーバークラウド環境に以下に示す新たな要素が追加されます。

- ストレージプロバイダーネットワーク
- コントローラーノード上の Ceph MDS サービス

ユーザーに利用を許可する前に、クラウド管理者はネイティブ CephFS 環境が安定して動作することを確認する必要があります。

ネイティブ CephFS で共有ファイルシステムサービスを使用する方法の詳細については、[ストレージガイド](#) の [共有ファイルシステムサービスの設定 \(manila\)](#) を参照してください。

12.1. ストレージプロバイダーネットワークの作成

新しい分離ストレージネットワークを Networking (neutron) プロバイダーネットワークにマッピングする必要があります。ネイティブ CephFS 共有のエクスポート場所にアクセスするために、コンピュートノードの仮想マシンをネットワークにアタッチします。

Shared File Systems サービスのネットワークセキュリティに関する情報は、[Red Hat OpenStack Platform の強化](#) の [Shared File System サービスの強化](#) を参照してください。

手順

openstack network create コマンドにより、ストレージ neutron ネットワークの設定を定義します。

1. source コマンドでオーバークラウドの認証情報ファイルを読み込みます。

```
$ source ~/<credentials_file>
```

- **<credentials_file>** を認証情報ファイルの名前 (**overcloudrc** など) に置き換えます。

2. アンダークラウドノードで、ストレージネットワークを作成します。

```
(overcloud) [stack@undercloud-0 ~]$ openstack network create Storage --provider-network-type vlan --provider-physical-network datacentre --provider-segment 30
```

以下のオプションを設定してこのコマンドを入力することができます。

- **--provider-physical-network** オプション: tripleo-heat-templates の NeutronBridgeMappings で別途 br-isolated ブリッジのタグを設定していない限り、デフォルト値 **datacentre** を使用します。
- **--provider-segment** オプション: ネットワーク環境ファイルで Storage 分離ネットワークに設定した値を使用します。これがカスタマイズされない場合、デフォルトの環境ファイルは `/usr/share/openstack-tripleo-heat-templates/network_data.yaml` になります。分離ネットワークの定義を変更していない限り、Storage ネットワークに関連付けられた VLAN の値は **30** です。
- **--provider-network-type** オプション: 値 **vlan** を使用します。

12.2. ストレージプロバイダーネットワークの設定

neutron プロバイダーネットワーク上に対応する **StorageSubnet** を作成します。アンダークラウドの **storage_subnet** と同じサブネットを設定します。また、ストレージサブネットと対応するアンダークラウドのサブネットの割り当て範囲が重複しないようにしてください。

要件

- 割り当てプールの IP 範囲 (開始および終了アドレス)
- サブネットの IP 範囲

手順

1. アンダークラウドノードから、以下のコマンドを入力します。

```
[stack@undercloud ~]$ source ~/overcloudrc
```

2. 下記のコマンド例を使用して、ネットワークをプロビジョニングします。値を実際の環境に応じて更新します。

```
(overcloud) [stack@undercloud-0 ~]$ openstack subnet create \
--allocation-pool start=172.17.3.10,end=172.17.3.149 \
--dhcp \
--network Storage \
--subnet-range 172.17.3.0/24 \
--gateway none StorageSubnet
```

- **--allocation-pool** オプションの場合は、**start=172.17.3.10,end=172.17.3.149** IP の値を、実際のネットワークの IP の値に置き換えます。
- **--subnet-range** オプション: **172.17.3.0/24** のサブネット範囲を実際のネットワークのサブネット範囲に置き換えます。

12.3. ストレージプロバイダーネットワークにおけるロールベースアクセス制御の設定

ストレージネットワークを使用することができる信頼済みテナントまたはプロジェクトを特定した後、Networking サービス (neutron) を使用してそれらのロールベースアクセス制御 (RBAC) ルールを設定します。

要件

ストレージネットワークへのアクセスを必要とするプロジェクトの名前

手順

1. アンダークラウドノードから、以下のコマンドを入力します。

```
[stack@undercloud ~]$ source ~/overcloudrc
```

2. アクセスが必要なプロジェクトを特定します。

```
(overcloud) [stack@undercloud-0 ~]$ openstack project list
+-----+
| ID                | Name      |
+-----+
| 06f1068f79d2400b88d1c2c33eacea87 | demo      |
| 5038dde12dfb44fdaa0b3ee4bfe487ce | service   |
| 820e2d9c956644c2b1530b514127fd0d | admin     |
+-----+
```

3. 必要なプロジェクトでネットワーク RBAC ルールを作成します。

```
(overcloud) [stack@undercloud-0 ~]$ openstack network rbac create \
--action access_as_shared Storage \
--type network \
--target-project demo
```

ストレージネットワークへのアクセスを必要とするすべてのプロジェクトについて、このステップを繰り返します。

12.4. デフォルトのファイル共有種別の設定

Shared File Systems サービス (manila) を使用して、特定の設定でファイル共有を作成するための共有種別を定義できます。ファイル共有の種別は、Block Storage のボリューム種別に類似した機能を持ちます。それぞれの種別には、追加の仕様などの関連する設定があります。ファイル共有の作成時に種別を呼び出すと、その設定が共有ファイルシステムに適用されます。

信頼できないユーザーからネイティブ CephFS バックエンドを保護するために、デフォルトの共有種別は作成しないでください。デフォルトの共有種別が存在しない場合、ユーザーは共有種別を指定することを強制され、信頼できるユーザーは排他的アクセス権限を持つカスタムのプライベート共有種別を使用できます。

信頼されていないテナントのデフォルトの共有種別を作成する必要がある場合、ネイティブ CephFS バックエンドからプロビジョニングを分離することができます。

手順

1. source コマンドでオーバークラウドの認証情報ファイルを読み込みます。

```
$ source ~/<credentials_file>
```

- **<credentials_file>** を認証情報ファイルの名前 (**overcloudrc** など) に置き換えます。

- 共有種別に追加の仕様を設定します。

```
(overcloud) [stack@undercloud-0 ~]$ manila type-create default false
(overcloud) [stack@undercloud-0 ~]$ manila type-key default set share_backend_name='s!=cephfs'
```

- プライベートの共有種別を作成して、信頼済みテナントにこの共有種別へのアクセス権限を付与します。

```
(overcloud) [stack@undercloud-0 ~]$ manila type-create --is-public false nativecephfstype false
(overcloud) [stack@undercloud-0 ~]$ manila type-key nativecephfstype set share_backend_name='cephfs'
(overcloud) [stack@undercloud-0 ~]$ manila type-access-add nativecephfstype <trusted_tenant_project_id>
```

- **<trusted_tenant_project_id>** を信頼されたテナントの ID に置き換えます。

共有タイプの詳細については、ストレージガイドの [共有タイプの作成](#) を参照してください。

12.5. 分離ストレージネットワークが作成されていることの確認

Shared File Systems サービスのバックエンドとしてネイティブ CephFS をデプロイするのに使用する **network_data.yaml** ファイルにより、ストレージ VLAN が作成されます。以下の手順を使用して、ストレージ VLAN が正常に作成されていることを確認します。

手順

- オーバークラウドのコントローラーノードのいずれかにログインします。
- 接続されたネットワークを確認し、**network_data.yaml** ファイルにより設定したとおりの VLAN が存在することを確認します。

```
$ ip a
8: vlan30: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default qlen 1000
    link/ether 52:9c:82:7a:d4:75 brd ff:ff:ff:ff:ff:ff
    inet 172.17.3.144/24 brd 172.17.3.255 scope global vlan30
        valid_lft forever preferred_lft forever
    inet6 fe80::509c:82ff:fe7a:d475/64 scope link
        valid_lft forever preferred_lft forever
```

12.6. CEPH MDS サービスの確認

Ceph MDS サービスのステータスを確認するには、**systemctl status** コマンドを使用します。

手順

- すべてのコントローラーノードで以下のコマンドを入力して、MDS コンテナのステータスを確認します。

```
$ systemctl status ceph-mds<@CONTROLLER-HOST>
```

以下に例を示します。

```
$ systemctl status ceph-mds@controller-0.service

ceph-mds@controller-0.service - Ceph MDS
  Loaded: loaded (/etc/systemd/system/ceph-mds@.service; enabled; vendor preset: disabled)
  Active: active (running) since Tue 2018-09-18 20:11:53 UTC; 6 days ago
  Main PID: 65066 (common)
  Tasks: 16 (limit: 204320)
  Memory: 38.2M
  CGroup: /system.slice/system-ceph\x2dmmds.slice/ceph-mds@controller-0.service
          └─60921 /usr/bin/podman run --rm --net=host --memory=32000m --cpus=4 -v
/var/lib/ceph:/var/lib/ceph:z -v /etc/ceph:/etc/ceph:z -v
/var/run/ceph:/var/run/ceph:z -v /etc/localtime:/etc/localtime:ro>
```

12.7. CEPH クラスターのステータスの確認

Ceph クラスターのステータスを確認し、クラスターがアクティブであることを確認します。

手順

1. 任意のコントローラーノードにログインします。
2. Ceph monitor デーモンから、以下のコマンドを入力します。

```
$ sudo podman exec ceph-mon-controller-0 ceph -s
cluster:
  id: 670dc288-cd36-4772-a4fc-47287f8e2ebf
  health: HEALTH_OK

services:
  mon: 3 daemons, quorum controller-1,controller-2,controller-0 (age 14h)
  mgr: controller-1(active, since 8w), standbys: controller-0, controller-2
  mds: cephfs:1 {0=controller-2=up:active} 2 up:standby
  osd: 15 osds: 15 up (since 8w), 15 in (since 8w)

task status:
  scrub status:
    mds.controller-2: idle

data:
  pools: 6 pools, 192 pgs
  objects: 309 objects, 1.6 GiB
  usage: 21 GiB used, 144 GiB / 165 GiB avail
  pgs: 192 active+clean
```



注記

1つのアクティブな MDS と 2つのスタンバイ状態の MDS があります。

3. Ceph File System の詳細ステータスを表示するには、以下のコマンドを入力します。

```
$ sudo ceph fs ls
```

```
name: cephfs metadata pool: manila_metadata, data pools: [manila_data]
```



注記

以下の出力例では、**cephfs** は、ユーザーが Shared File Systems サービスを使用して作成する CephFS ファイル共有をホストするために director が作成する Ceph File System の名前です。

12.8. MANILA-SHARE サービスのステータスの確認

manila-share サービスのステータスを確認します。

手順

1. いずれかのコントローラーノードから、**openstack-manila-share** が起動していることを確認します。

```
$ sudo pcs status resources | grep manila
```

```
* Container bundle: openstack-manila-share [cluster.common.tag/rhosp16-openstack-manila-share:pcmklatest]:
```

```
* openstack-manila-share-podman-0 (ocf::heartbeat:podman): Started controller-0
```

12.9. MANILA-API サービスがスケジューラーおよびファイル共有サービスを認識していることの確認

以下の手順を実施して、**manila-api** サービスがスケジューラーおよびファイル共有サービスを認識していることを確認します。

手順

1. アンダークラウドにログインします。
2. 以下のコマンドを入力します。

```
$ source /home/stack/overcloudrc
```

3. 以下のコマンドを入力して、**manila-scheduler** および **manila-share** が有効であることを確認します。

```
$ manila service-list
```

```
| Id | Binary          | Host          | Zone | Status | State | Updated_at |
```

```
| 2 | manila-scheduler | hostgroup     | nova | enabled | up    | 2018-08-08T04:15:03.000000 |
```

```
| 5 | manila-share     | hostgroup@cephfs | nova | enabled | up    | 2018-08-08T04:15:03.000000 |
```

第13章 CEPHFS NFS デプロイ後の設定と検証

NFS 共有を作成し、ユーザーにアクセス権限を付与し、NFS 共有をマウントする前に、2 つのデプロイメント後設定タスクを完了する必要があります。

- Networking サービス (neutron) の StorageNFS ネットワークをデータセンターの StorageNFS 分離ネットワークにマッピングする。NFS トラフィックを別のネットワークに分離したくない場合は、このオプションを省略できます。
- デフォルトのファイル共有種別を作成する。

これらのステップを完了したら、テナントコンピュートインスタンスは NFS 共有を作成し、そのアクセスを許可し、マウントすることができます。

Shared File Systems サービス (manila) の NFS バックエンドに CephFS を使用する設定をデプロイする場合、オーバークラウド環境に以下に示す新たな要素を追加します。

- StorageNFS ネットワーク
- コントローラー上の Ceph MDS サービス
- コントローラー上の NFS-Ganesha サービス

ユーザーにサービスの利用を許可する前に、クラウド管理者は NFS バックエンドに CephFS を使用する環境が安定して動作することを確認する必要があります。

13.1. ストレージプロバイダーネットワークの作成

新しい StorageNFS 分離ネットワークを Networking (neutron) プロバイダーネットワークにマッピングする必要があります。NFS-Ganesha ゲートウェイの提供するファイル共有のエクスポート場所にアクセスするために、コンピュートノードの仮想マシンをネットワークにアタッチします。

Shared File Systems サービスのネットワークセキュリティに関する情報は、[Red Hat OpenStack Platform の強化の Shared File System サービスの強化](#) を参照してください。

手順

openstack network create コマンドにより、StorageNFS neutron ネットワークの設定を定義します。

1. source コマンドでオーバークラウドの認証情報ファイルを読み込みます。

```
$ source ~/<credentials_file>
```

- **<credentials_file>** を認証情報ファイルの名前 (**overcloudrc** など) に置き換えます。

2. アンダークラウドノードで、StorageNFS ネットワークを作成します。

```
(overcloud) [stack@undercloud-0 ~]$ openstack network create StorageNFS --share --
provider-network-type vlan --provider-physical-network datacentre --provider-segment 70
```

以下のオプションを設定してこのコマンドを入力することができます。

- **--provider-physical-network** オプション: tripleo-heat-templates の NeutronBridgeMappings で別途 br-isolated ブリッジのタグを設定していない限り、デフォルト値 **datacentre** を使用します。

- **--provider-segment** オプション: heat テンプレート (`/usr/share/openstack-tripleo-heat-templates/network_data_ganesha.yaml`) で StorageNFS 分離ネットワークに設定した VLAN の値を使用します。デプロイ時に分離ネットワークの定義を変更していない限り、この値は 70 です。
- **--provider-network-type** オプション: 値 **vlan** を使用します。

13.2. 共有プロバイダー STORAGENFS ネットワークの設定

neutron 共有プロバイダーネットワークに対応する **StorageNFSSubnet** を作成します。サブネットが **network_data.yml** ファイルの **storage_nfs** ネットワーク定義と同じであることを確認し、**StorageNFS** サブネットと対応するアンダークラウドのサブネットの割り当て範囲が重複しないようにしてください。**StorageNFS** サブネットは NFS 共有の提供用であるため、ゲートウェイは必要ありません。

前提条件

- 割り当てプールの IP 範囲 (開始および終了アドレス)
- サブネットの IP 範囲

13.2.1. 共有プロバイダー StorageNFS IPv4 ネットワークの設定

neutron 共有 IPv4 プロバイダーネットワークに対応する **StorageNFSSubnet** を作成します。

手順

1. オーバークラウドノードにログインします。
2. `source` コマンドでオーバークラウドの認証情報を読み込みます。
3. 下記のコマンド例を使用して、以下の変更を行いネットワークをプロビジョニングします。
 - a. **start=172.17.0.4,end=172.17.0.250** の IP の値を、実際のネットワークの IP の値に置き換えます。
 - b. **172.17.0.0/20** のサブネット範囲を、実際のネットワークのサブネット範囲に置き換えます。

```
[stack@undercloud-0 ~]$ openstack subnet create --allocation-pool
start=172.17.0.4,end=172.17.0.250 \
--dhcp --network StorageNFS --subnet-range 172.17.0.0/20 \
--gateway none StorageNFSSubnet
```

13.2.2. 共有プロバイダー StorageNFS IPv6 ネットワークの設定

neutron 共有 IPv6 プロバイダーネットワークに対応する **StorageNFSSubnet** を作成します。

手順

1. オーバークラウドノードにログインします。
2. 下記のコマンド例を使用して、必要に応じて値の変更を行いネットワークをプロビジョニングします。

- **fd00:fd00:fd00:7000::/64** のサブネット範囲を、実際のネットワークのサブネット範囲に置き換えます。

```
[stack@undercloud-0 ~]$ openstack subnet create --ip-version 6 --dhcp --network StorageNFS --
subnet-range fd00:fd00:fd00:7000::/64 --gateway none --ipv6-ra-mode dhcpv6-stateful --ipv6-
address-mode dhcpv6-stateful StorageNFSSubnet -f yaml
```

13.3. デフォルトのファイル共有種別の設定

Shared File Systems サービス (manila) を使用して、特定の設定でファイル共有を作成するための共有種別を定義できます。ファイル共有の種別は、Block Storage のボリューム種別に類似した機能を持ちます。それぞれの種別には、追加の仕様などの関連する設定があります。ファイル共有の作成時に種別を呼び出すと、その設定が共有ファイルシステムに適用されます。

Red Hat OpenStack Platform (RHOSP) director を使用する場合、ユーザーにクラウドへのアクセスを許可する前に、デフォルトのファイル共有種別を作成する必要があります。

手順

- NFS を使用する CephFS のデフォルトの共有種別を作成します。

```
$ manila type-create default false
```

共有タイプの詳細については、ストレージガイドの [共有タイプの作成](#) を参照してください。

13.4. STORAGENFS 分離ネットワークが作成されていることの確認

Shared File Systems サービスの NFS バックエンドに CephFS を使用する設定をデプロイするのに使用する **network_data_ganesha.yaml** ファイルにより、StorageNFS VLAN が作成されます。StorageNFS 分離ネットワークが存在することを確認するには、以下の手順を実施します。

手順

1. オーバークラウドのコントローラーのいずれかにログインします。
2. 以下のコマンドを入力して接続されたネットワークを確認し、**network_data_ganesha.yaml** により設定したとおりの VLAN が存在することを確認します。

```
$ ip a
15: vlan310: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state
UNKNOWN group default qlen 1000
    link/ether 32:80:cf:0e:11:ca brd ff:ff:ff:ff:ff:ff
    inet 172.16.4.4/24 brd 172.16.4.255 scope global vlan310
        valid_lft forever preferred_lft forever
    inet 172.16.4.7/32 brd 172.16.4.255 scope global vlan310
        valid_lft forever preferred_lft forever
    inet6 fe80::3080:cfff:fe0e:11ca/64 scope link
        valid_lft forever preferred_lft forever
```

13.5. CEPH MDS サービスの確認

Ceph MDS サービスのステータスを確認するには、**systemctl status** コマンドを使用します。

手順

- すべてのコントローラーノードで以下のコマンドを入力して、MDS コンテナのステータスを確認します。

```
$ systemctl status ceph-mds<@CONTROLLER-HOST>
```

以下に例を示します。

```
$ systemctl status ceph-mds@controller-0.service

ceph-mds@controller-0.service - Ceph MDS
  Loaded: loaded (/etc/systemd/system/ceph-mds@.service; enabled; vendor preset: disabled)
  Active: active (running) since Tue 2018-09-18 20:11:53 UTC; 6 days ago
    Main PID: 65066 (common)
      Tasks: 16 (limit: 204320)
     Memory: 38.2M
    CGroup: /system.slice/system-ceph\x2dmmds.slice/ceph-mds@controller-0.service
            └─60921 /usr/bin/podman run --rm --net=host --memory=32000m --cpus=4 -v
/var/lib/ceph:/var/lib/ceph:z -v /etc/ceph:/etc/ceph:z -v
/var/run/ceph:/var/run/ceph:z -v /etc/localtime:/etc/localtime:ro>
```

13.6. CEPH クラスターのステータスの確認

Ceph クラスターのステータスを確認するには、以下の手順を実施します。

手順

1. アクティブなコントローラーノードにログインします。
2. 以下のコマンドを入力します。

```
$ sudo ceph -s

cluster:
  id: 3369e280-7578-11e8-8ef3-801844eeec7c
  health: HEALTH_OK

services:
  mon: 3 daemons, quorum overcloud-controller-1,overcloud-controller-2,overcloud-controller-0
  mgr: overcloud-controller-1(active), standbys: overcloud-controller-2, overcloud-controller-0
  mds: cephfs-1/1/1 up {0=overcloud-controller-0=up:active}, 2 up:standby
  osd: 6 osds: 6 up, 6 in
```

1つのアクティブな MDS と 2つのスタンバイ状態の MDS があります。

3. Ceph ファイルシステムのステータスをより詳細に確認するには、以下のコマンドを入力します。<cephfs> は、Ceph ファイルシステムの名前に置き換えてください。

```
$ sudo ceph fs ls

name: cephfs, metadata pool: manila_metadata, data pools: [manila_data]
```

13.7. NFS-GANESHA および MANILA-SHARE サービスのステータスの確認

NFS-Ganesha および manila-share サービスのステータスを確認するには、以下の手順を実施します。

手順

1. いずれかのコントローラーノードから以下のコマンドを入力して、**ceph-nfs** および **openstack-manila-share** が起動していることを確認します。

```
$ pcs status

ceph-nfs      (systemd:ceph-nfs@pacemaker):  Started overcloud-controller-1

podman container: openstack-manila-share [192.168.24.1:8787/rhosp-rhel8/openstack-manila-share:pcmklatest]
  openstack-manila-share-podman-0    (ocf::heartbeat:podman):    Started overcloud-controller-1
```

13.8. MANILA-API サービスがスケジューラーおよびファイル共有サービスを認識していることの確認

以下の手順を実施して、**manila-api** サービスがスケジューラーおよびファイル共有サービスを認識していることを確認します。

手順

1. アンダークラウドにログインします。
2. 以下のコマンドを入力します。

```
$ source /home/stack/overcloudrc
```

3. 以下のコマンドを入力して、**manila-scheduler** および **manila-share** が有効であることを確認します。

```
$ manila service-list

| Id | Binary          | Host          | Zone | Status | State | Updated_at |
| 2 | manila-scheduler | hostgroup     | nova | enabled | up    | 2018-08-08T04:15:03.000000 |
| 5 | manila-share     | hostgroup@cephfs | nova | enabled | up    | 2018-08-08T04:15:03.000000 |
```

第14章 環境のリブート

環境の再起動が必要になる場合があります。たとえば、物理サーバーを変更したり、停電から回復したりする必要がある場合です。このような状況では、Ceph Storage ノードの正常な起動を確認することが重要です。

ノードは、次の順序で起動する必要があります。

1. **すべての Ceph Monitor ノードを最初に起動します**: これにより、高可用性 Ceph クラスター内の Ceph Monitor サービスを確実にアクティブにします。デフォルトでは、Ceph Monitor サービスは、コントローラーノードにインストールされます。Ceph Monitor がカスタムロールで Controller とは別の場合には、このカスタムの Ceph Monitor ロールを必ずアクティブにしてください。
2. **すべての Ceph Storage ノードを起動します**: これにより、Ceph OSD クラスターはコントローラーノード上のアクティブな Ceph Monitor クラスターに接続できるようになります。

14.1. CEPH STORAGE (OSD) クラスターのリブート

Ceph Storage (OSD) ノードのクラスターを再起動するには、以下の手順を実施します。

前提条件

- **ceph-mon** サービスを実行している Ceph Monitor または Controller ノードで、Red Hat Ceph Storage クラスターのステータスが正常であり、pg ステータスが **active+clean** であることを確認する。

```
$ sudo cephadm -- shell ceph status
```

Ceph クラスターが正常な場合、**HEALTH_OK** のステータスが返されます。

Ceph クラスターのステータスが異常な場合、**HEALTH_WARN** または **HEALTH_ERR** のステータスが返されます。トラブルシューティングのガイダンスについては、[Red Hat Ceph Storage 5 トラブルシューティングガイド](#) を参照してください。

手順

1. **ceph-mon** サービスを実行している Ceph Monitor または Controller ノードにログインし、Ceph Storage クラスターのリバランスを一時的に無効にします。

```
$ sudo cephadm shell -- ceph osd set noout
$ sudo cephadm shell -- ceph osd set norebalance
```



注記

マルチスタックまたは分散計算ノード (DCN) アーキテクチャを使用している場合は、**noout** フラグと **norebalance** フラグの設定時に Ceph クラスター名を指定する必要があります。例: **sudo cephadm shell -c /etc/ceph/<cluster>.conf -k /etc/ceph/<cluster>.client.keyring**。

2. 再起動する最初の Ceph Storage ノードを選択し、そのノードにログインします。
3. ノードを再起動します。

■

```
$ sudo reboot
```

4. ノードが起動するまで待ちます。
5. ノードにログインし、Ceph クラスターのステータスを確認します。

```
$ sudo cephadm -- shell ceph status
```

pgmap により、すべての **pgs** が正常な状態 (**active+clean**) として報告されることを確認します。

6. ノードからログアウトして、次のノードを再起動し、ステータスを確認します。全 Ceph Storage ノードが再起動されるまで、このプロセスを繰り返します。
7. 完了したら、**ceph-mon** サービスを実行している Ceph Monitor または Controller ノードにログインし、クラスターのリバランスを有効にします。

```
$ sudo cephadm shell -- ceph osd unset noout
$ sudo cephadm shell -- ceph osd unset norebalance
```



注記

マルチスタックまたは分散計算ノード (DCN) アーキテクチャを使用している場合は、**noout** フラグと **norebalance** フラグの設定解除時に Ceph クラスター名を指定する必要があります。例: **sudo cephadm shell -c /etc/ceph/<cluster>.conf -k /etc/ceph/<cluster>.client.keyring**。

8. 最終のステータスチェックを実行して、クラスターが **HEALTH_OK** を報告していることを確認します。

```
$ sudo cephadm shell ceph status
```

14.2. CEPH STORAGE OSD の再起動による CEPH MONITOR サービスへの接続の有効化

すべてのオーバークラウドノードが同時に起動する状況が発生した場合には、Ceph OSD サービスが Ceph Storage ノード上で正常に起動されない場合があります。そのような場合には、Ceph Storage OSD をリブートして、Ceph Monitor サービスに接続できるようにします。

手順

- Ceph Storage ノードクラスターの **HEALTH_OK** ステータスを確認します。

```
$ sudo ceph status
```

第15章 CEPH STORAGE クラスターのスケーリング

ストレージノードを追加または削除することで、Ceph Storage クラスターのサイズをスケーリングできます。

15.1. CEPH STORAGE クラスターのスケールアップ

容量とパフォーマンスの要件が変化するにつれ、需要の増加に対応するために Ceph Storage クラスターをスケールアップできます。この操作を実行する前に、更新するデプロイメント用に十分なノードがあることを確認してください。その後、Red Hat OpenStack Platform (RHOSP) 環境で新しいノードを登録してタグ付けできます。

新しい Ceph Storage ノードを director に登録するには、この手順を完了します。

手順

1. アンダークラウドノードに **stack** ユーザーとしてログインします。
2. `~/overcloud-baremetal-deploy.yaml` を変更して、デプロイメントに CephStorage ノードを追加します。
次のサンプルファイルは、3 つの CephStorage ノードを持つ元のデプロイメントを表しています。

```
- name: CephStorage
  count: 3
  instances:
    - hostname: ceph-0
      name: ceph-0
    - hostname: ceph-1
      name: ceph-2
    - hostname: ceph-2
      name: ceph-2
```

次の例では、このファイルを変更して 3 つのノードを追加します。

```
- name: CephStorage
  count: 6
  instances:
    - hostname: ceph-0
      name: ceph-0
    - hostname: ceph-1
      name: ceph-2
    - hostname: ceph-2
      name: ceph-2
    - hostname: ceph-3
      name: ceph-3
    - hostname: ceph-4
      name: ceph-4
    - hostname: ceph-5
      name: ceph-5
```

3. 更新された `~/overcloud-baremetal-deploy.yaml` ファイルで **openstack overcloud node provision** コマンドを使用します。

```

openstack overcloud node provision \
  --stack overcloud \
  --network-config \
  --output ~/overcloud-baremetal-deployed.yaml \
  ~/overcloud-baremetal-deploy.yaml

```



注記

このコマンドは、設定済みのノードをプロビジョニングし、~/overcloud-baremetal-deployed.yaml の更新済みコピーを出力します。新しいバージョンは **CephStorage** を更新します。**DeployedServerPortMap** と **HostnameMap** にも、新しいストレージノードが含まれています。

- 更新された ~/overcloud-baremetal-deployed.yaml ファイルで **openstack overcloud deploy** コマンドを使用します。

```

openstack overcloud deploy --templates \
  -e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/cephadm.yaml \
  -e deployed_ceph.yaml
  -e overcloud-baremetal-deploy.yaml

```

結果

openstack overcloud deploy コマンドを実行すると、以下のアクションが発生します。

- ストレージネットワークとファイアウォールルールは、新しい **CephStorage** ノードで設定されます。
- ceph-admin** ユーザーが新しい **CephStorage** ノードに作成されます。
- ceph-admin** ユーザーのパブリック SSH キーが新しい **CephStorage** ノードに配布されるため、**cephadm** は SSH を使用してノードを追加できます。
- 新しい **CephMon** または **CephMgr** ノードが追加されると、**ceph-admin** プライベート SSH キーもそのノードに配布されます。
- 更新された Ceph 仕様が生成され、ブートストラップノードにインストールされます。この更新された仕様は通常、ブートストラップノードの /home/ceph-admin/specs/ceph_spec.yaml で利用できます。
- cephadm ls** は Ceph コンテナが既に実行中であることを示すため、**cephadm** ブートストラッププロセスは省略されます。
- 更新された Ceph 仕様が適用され、**cephadm** は新しいノードが Ceph クラスターに参加するようにスケジュールします。

15.2. CEPH STORAGE ノードのスケールダウンと置き換え

場合によっては、Ceph Storage クラスターをスケールダウンするか、Ceph Storage ノードを置き換える必要があります。いずれの場合も、データの損失を防ぐために、オーバークラウドから削除する Ceph Storage ノードを無効にして再調整する必要があります。



手順

Ceph Storage クラスターに OSD を失うだけの容量がない場合は、この手順を続行しないでください。

1. **tripleo-admin** ユーザーとしてオーバークラウドのコントローラーノードにログインします。
2. **sudo cephadm shell** コマンドを使用して、Ceph シェルを開始します。
3. **ceph osd tree** コマンドを使用して、サーバーによって削除される OSD を識別します。
次の例では、**ceph-2** ホストの OSD を識別します。

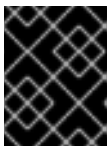
```
[ceph: root@oc0-controller-0 /]# ceph osd tree
ID CLASS WEIGHT  TYPE NAME        STATUS REWEIGHT PRI-AFF
-1      0.58557 root default
-7      0.19519 host ceph-2
  5  hdd  0.04880   osd.5      up    1.00000 1.00000
  7  hdd  0.04880   osd.7      up    1.00000 1.00000
  9  hdd  0.04880   osd.9      up    1.00000 1.00000
 11  hdd  0.04880   osd.11     up    1.00000 1.00000
```

4. Ceph クラスター仕様を YAML ファイルにエクスポートします。

```
[ceph: root@oc0-controller-0 /]# ceph orch ls --export > spec.yml
```

5. エクスポートされた仕様ファイルを編集して、該当するホストが **service-type: osd hosts** リストから削除され、該当するホストの **placement: hosts** 値が削除されるようにします。
6. 編集したファイルを保存します。
7. 変更した Ceph 仕様ファイルを適用します。

```
[ceph: root@oc0-controller-0 /]# ceph orch apply -i spec.yml
```



重要

OSD を削除する前に Ceph 仕様ファイルをエクスポートして編集しない場合、Ceph Manager は OSD を再作成しようとします。

8. OSD を削除するには、コマンド **ceph orch osd rm --zap <osd_list>** を使用します。

```
[ceph: root@oc0-controller-0 /]# ceph orch osd rm --zap 5 7 9 11
Scheduled OSD(s) for removal
[ceph: root@oc0-controller-0 /]# ceph orch osd rm status
OSD_ID HOST  STATE  PG_COUNT REPLACE FORCE DRAIN_STARTED_AT
7  ceph-2 draining 27    False  False 2021-04-23 21:35:51.215361
9  ceph-2 draining 8     False  False 2021-04-23 21:35:49.111500
11 ceph-2 draining 14    False  False 2021-04-23 21:35:50.243762
```

9. コマンド **ceph orch osd status** を使用して、OSD 削除のステータスを確認します。

```
[ceph: root@oc0-controller-0 /]# ceph orch osd rm status
OSD_ID HOST STATE PG_COUNT REPLACE FORCE DRAIN_STARTED_AT
7 ceph-2 draining 34 False False 2021-04-23 21:35:51.215361
11 ceph-2 draining 14 False False 2021-04-23 21:35:50.243762
```



警告

このコマンドが結果を返さなくなるまで、次のステップに進まないでください。

10. コマンド **ceph orch host drain <HOST>** を使用して、残りのデーモンをドレインします。

```
[ceph: root@oc0-controller-0 /]# ceph orch host drain ceph-2
```

11. コマンド **ceph orch host rm <HOST>** を使用して、ホストを削除します。

```
[ceph: root@oc0-controller-0 /]# ceph orch host rm ceph-2
```

12. Ceph シェルセッションを終了します。

13. **tripleo-admin** アカウントからログアウトします。

14. アンダークラウドノードに **stack** ユーザーとしてログインします。

15. **~/overcloud-baremetal-deploy.yaml** を以下の方法で変更します。

- スケールダウンするロールの **count** 属性を減らします。
 - プロビジョニングを解除する各ノードの **instances** エントリーを追加します。各エントリーには以下が含まれている必要があります。
 - ベアメタルノードの **name**。
 - そのノードに割り当てられた **hostname**。
 - **provisioned: false** 値。
- 以下の例では、ノード **overcloud-compute-1** を削除します。

```
- name: Compute
  count: 1
  instances:
    - hostname: overcloud-compute-0
      name: node10
      # Removed from deployment due to disk failure
      provisioned: false
    - hostname: overcloud-compute-1
      name: node11
```

16. ノードを削除するには、**openstack overcloud node delete** コマンドを使用します。

```
openstack overcloud node delete \
--stack overcloud \
--baremetal-deployment ~/overcloud-baremetal-deploy.yaml
```



注記

ノードが削除される前に、削除するノードのリストが確認プロンプトとともに提供されます。



注記

Ceph クラスターのスケールダウンが一時的であり、削除されたノードが後で復元される場合、スケールアップアクションは **count** を増やし、以前は **provisioned: false** に設定されていたノードで **provisioned: true** を設定できます。ノードが再利用されない場合は、**provisioned: false** を無期限で設定し、スケールアップアクションで新しいインスタンスエントリを指定できます。

+ 次のファイルサンプルは、各インスタンスの例をいくつか示しています。

+

```
- name: Compute
  count: 2
  instances:
    - hostname: overcloud-compute-0
      name: node10
      # Removed from deployment due to disk failure
      provisioned: false
    - hostname: overcloud-compute-1
      name: node11
    - hostname: overcloud-compute-2
      name: node12
```

第16章 障害のあるディスクの置き換え

Ceph Storage クラスター内のディスクに障害が発生した場合は、交換が可能です。

16.1. ディスクの交換

障害が発生したディスクの交換については、Red Hat Ceph Storage Installation Guideの [Adding OSDs](#) を参照してください。