



Red Hat OpenStack Platform 8

オーバークラウド向けの Red Hat Ceph Storage

オーバークラウドで Red Hat Ceph Storage を使用するための設定

Red Hat OpenStack Platform 8 オーバークラウド向けの Red Hat Ceph Storage

オーバークラウドで Red Hat Ceph Storage を使用するための設定

OpenStack Team
rhos-docs@redhat.com

法律上の通知

Copyright © 2017 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, 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 Software Collections 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 Ceph Storage の環境の推奨事項や Ceph Storage ノードでオーバークラウドを実装する方法など、Red Hat OpenStack Platform director を使用して、Red Hat Ceph Storage を使用するオーバークラウドを作成する方法を説明します。

目次

第1章 はじめに	3
1.1. CEPH STORAGE の定義	3
1.2. RED HAT OPENSTACK PLATFORM での RED HAT CEPH STORAGE の使用	3
1.3. 設定要件	3
1.4. シナリオの定義	4
第2章 CEPH STORAGE ノードでのオーバークラウドの作成	6
2.1. STACK ユーザーの初期化	6
2.2. ノードの登録	7
2.3. ノードのハードウェアの検査	9
2.4. ノードの手動でのタグ付け	9
2.5. CEPH STORAGE ノードの ROOT ディスクの定義	10
2.6. オーバークラウドでの CEPH STORAGE の有効化	11
2.7. CEPH STORAGE ノードディスクのレイアウトのマッピング	12
2.8. CEPH STORAGE ノードディスクの GPT への変換	13
2.9. オーバークラウドの作成	14
2.10. オーバークラウドへのアクセス	15
2.11. CEPH STORAGE ノードの監視	15
2.12. CEPH STORAGE ノードの置き換え	16
第3章 既存の CEPH STORAGE CLUSTER のオーバークラウドへの統合	19
3.1. 既存の CEPH STORAGE CLUSTER の設定	19
3.2. STACK ユーザーの初期化	20
3.3. ノードの登録	20
3.4. ノードのハードウェアの検査	22
3.5. ノードの手動でのタグ付け	22
3.6. 既存の CEPH STORAGE CLUSTER との統合の有効化	23
3.7. オーバークラウドの作成	23
3.8. オーバークラウドへのアクセス	25
第4章 まとめ	26

第1章 はじめに

Red Hat OpenStack Platform director は、オーバークラウドと呼ばれるクラウド環境を作成します。director を使用して、オーバークラウドの追加機能を設定することができます。これらの追加機能の1つに、director で作成した Ceph Storage Cluster や、既存の Ceph Storage Cluster などの Red Hat Ceph Storage との統合が含まれます。本ガイドには、director を使用してオーバークラウドに Ceph Storage を統合する方法についての説明と設定例を記載します。

1.1. CEPH STORAGE の定義

Red Hat Ceph Storage は、優れたパフォーマンス、信頼性、スケーリングを提供するように設計された、分散型のデータオブジェクトストレージです。分散型のオブジェクトストアは、非構造化データに対応し、クライアントは新しいタイプのオブジェクトインターフェースと従来のインターフェースを同時に使用できるため、今後のストレージのあり方です。Ceph デプロイメントはすべて、2種類のデーモンで構成される **Ceph Storage Cluster** を中心とします。

Ceph OSD (Object Storage Daemon)

Ceph OSD は、Ceph クライアントの代わりにデータを格納します。また、Ceph ノードの CPU とメモリーを使用して、データの複製、リバランス、復旧、監視、レポート作成を実行します。

Ceph モニター

Ceph モニターは、ストレージクラスターの現在のステータスを含む Ceph ストレージクラスターのマッピングのマスターコピーを管理します。

Red Hat Ceph Storage に関する詳しい情報は、『[Red Hat Ceph Storage Architecture Guide](#)』を参照してください。



重要

本ガイドでは、Ceph Block Storage のみを統合します。Ceph Object (RGW) または Ceph File (CephFS) ストレージの統合については記載していません。

1.2. RED HAT OPENSTACK PLATFORM での RED HAT CEPH STORAGE の使用

Red Hat OpenStack Platform director で Red Hat Ceph Storage をオーバークラウドに統合するには、主に2つの方法があります。

Ceph Storage Cluster でのオーバークラウドの作成

director には、オーバークラウドの作成中に Ceph Storage Cluster を作成する機能があります。director は、データの格納に Ceph OSD を使用する Ceph Storage ノードセットを作成します。さらに、director は、オーバークラウドのコントローラーノードに Ceph Monitor サービスをインストールします。このため、組織が高可用性のコントローラーノード3台で構成されるオーバークラウドを作成する場合には、Ceph Monitor も高可用性サービスになります。

既存の Ceph Storage のオーバークラウドへの統合

既存の Ceph Storage Cluster がある場合には、オーバークラウドのデプロイメント時に統合できます。これは、オーバークラウドの設定以外のクラスターの管理やスケーリングが可能であることを意味します。

1.3. 設定要件

本ガイドは、『[Red Hat OpenStack Platform 8 director インストールと使用方法](#)』ガイドの補足情報としての役割を果たします。これは、『[2章 要件](#)』で指定されている同じ要件が本ガイドにも適用される

ことを意味します。必要に応じて、この要件を実装してください。

Red Hat OpenStack Platform director を使用して Ceph Storage ノードを作成する場合は、それらのノードに対する以下の要件に注意してください。

プロセッサ

Intel 64 または AMD64 CPU 拡張機能のサポートがある 64 ビットの x86 プロセッサ

メモリー

メモリー要件はストレージ容量によって異なります。ハードディスク容量 1 TB あたり最小で 1 GB のメモリーを使用するのが理想的です。

ディスク領域

ストレージ要件はメモリーの容量によって異なります。ハードディスク容量 1 TB あたり最小で 1 GB のメモリーを使用するのが理想的です。

ディスクのレイアウト

推奨される Red Hat Ceph Storage ノードの設定には、以下のようなディスクレイアウトが必要です。

- **/dev/sda:** root ディスク。director は、主なオーバークラウドイメージをディスクにコピーします。
- **/dev/sdb:** ジャーナルディスク。このディスクは、**/dev/sdb1**、**/dev/sdb2**、**/dev/sdb3** などのように、Ceph OSD ジャーナル向けにパーティションを分割します。ジャーナルディスクは通常、システムパフォーマンス向上に役立つ **Solid State Drive (SSD)** です。
- **/dev/sdc** 以降: OSD ディスク。ストレージ要件で必要な数のディスクを使用します。



重要

Ceph Storage OSD およびジャーナルのパーティションには、GPT ディスクラベルが必要です。このラベルも、カスタマイズの前に設定する必要があります。

ネットワークインターフェースカード

最小で 1 x 1 Gbps ネットワークインターフェースカード (実稼動環境では、最低でも NIC を 2 つ以上使用することを推奨します)。ボンディングインターフェース向けの場合や、タグ付けされた VLAN トラフィックを委譲する場合には、追加のネットワークインターフェースを使用します。特に大量のトラフィックにサービスを提供する OpenStack Platform 環境を構築する場合には、ストレージノードには 10 Gbps インターフェースを使用することを推奨します。

Intelligent Platform Management Interface (IPMI)

各 Ceph ノードには、サーバーのマザーボード上に IPMI 機能が必要です。

本ガイドでは、以下の要件も満たす必要があります。

- Red Hat OpenStack Platform director でインストールしたアンダークラウドホスト。[「3章 アンダークラウドのインストール」](#)を参照してください。
- Red Hat Ceph Storage のハードウェアの追加の推奨事項。これらの推奨事項については『[Red Hat Ceph Storage Hardware Guide](#)』を参照してください。

1.4. シナリオの定義

本ガイドでは、2 つのシナリオを使用します。

- 最初のシナリオでは、Ceph Storage Cluster でオーバークラウドを作成します。これは、director が Ceph Storage Cluster をデプロイすることを意味します。
- 2 番目のシナリオでは、既存の Ceph Storage Cluster とオーバークラウドを統合します。これは、オーバークラウドの管理と Ceph Storage Cluster の管理を分けることを意味します。

第2章 CEPH STORAGE ノードでのオーバークラウドの作成

このシナリオでは、**director** を使用して、独自の **Ceph Storage Cluster** が含まれるオーバークラウドを作成する方法を確認します。このシナリオでは、オーバークラウドは 9 台のノードで構成されます。

- 高可用性のコントローラーノード 3 台。各ノードに **Ceph Monitor** サービスが含まれます。
- クラスタ内に **Red Hat Ceph Storage** ノード 3 台。これらのノードには、**Ceph OSD** が含まれ、実際のストレージとしての役割を果たします。
- コンピュートノード 3 台

すべてのマシンは、電源管理に **IPMI** を使用したベアメタルマシンです。**director** により **Red Hat Enterprise Linux 7** のイメージが各ノードにコピーされるため、これらのノードではオペレーティングシステムは必要ありません。

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を使用して各ノードと通信します。すべてのノードは、ネイティブの **VLAN** 経由でネットワークに接続します。この例では、以下の **IP** アドレスの割り当てで、プロビジョニングサブネットとして **192.0.2.0/24** を使用します。

ノード名	IP アドレス	MAC アドレス	IPMI IP アドレス
director	192.0.2.1	aa:aa:aa:aa:aa:aa	
コントローラー 1	定義済みの DHCP	b1:b1:b1:b1:b1:b1	192.0.2.205
コントローラー 2	定義済みの DHCP	b2:b2:b2:b2:b2:b2	192.0.2.206
コントローラー 3	定義済みの DHCP	b3:b3:b3:b3:b3:b3	192.0.2.207
コンピュート 1	定義済みの DHCP	c1:c1:c1:c1:c1:c1	192.0.2.208
コンピュート 2	定義済みの DHCP	c2:c2:c2:c2:c2:c2	192.0.2.209
コンピュート 3	定義済みの DHCP	c3:c3:c3:c3:c3:c3	192.0.2.210
Ceph 1	定義済みの DHCP	d1:d1:d1:d1:d1:d1	192.0.2.211
Ceph 2	定義済みの DHCP	d2:d2:d2:d2:d2:d2	192.0.2.212
Ceph 3	定義済みの DHCP	d3:d3:d3:d3:d3:d3	192.0.2.213

2.1. STACK ユーザーの初期化

stack ユーザーとして **director** ホストにログインし、以下のコマンドを実行して **director** の設定を初期化します。

```
$ source ~/stackrc
```

このコマンドでは、`director` の CLI ツールにアクセスする認証情報が含まれる環境変数を設定します。

2.2. ノードの登録

ノード定義のテンプレート (`instackenv.json`) は JSON ファイル形式で、ノード登録用のハードウェアおよび電源管理の情報が含まれています。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": "pxe_ipmitool",
      "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": "pxe_ipmitool",
      "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": "pxe_ipmitool",
      "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": "pxe_ipmitool",
    "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": "pxe_ipmitool",
    "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": "pxe_ipmitool",
    "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": "pxe_ipmitool",
    "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": "pxe_ipmitool",
    "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": "pxe_ipmitool",
        "pm_user": "admin",
        "pm_password": "p@55w0rd!",
        "pm_addr": "192.0.2.213"
    }
]
}

```

テンプレートの作成後に、**stack** ユーザーのホームディレクトリー (`/home/stack/instackenv.json`) にファイルを保存してから、**director** にインポートします。これには、以下のコマンドを実行します。

```
$ openstack baremetal import --json ~/instackenv.json
```

このコマンドでテンプレートをインポートして、テンプレートから **director** に各ノードを登録します。カーネルと **ramdisk** イメージを全ノードに割り当てます。

```
$ openstack baremetal configure boot
```

director でのノードの登録、設定が完了しました。

2.3. ノードのハードウェアの検査

ノードの登録後には、各ノードのハードウェア属性を検査します。各ノードのハードウェア属性を検査するには、以下のコマンドを実行します。

```
$ openstack baremetal introspection bulk start
```



重要

このプロセスが最後まで実行されて正常に終了したことを確認してください。ベアメタルの場合には、通常 15 分ほどかかります。

2.4. ノードの手動でのタグ付け

各ノードのハードウェアを登録、検査した後は、特定のプロファイルにノードをタグ付けします。これらのプロファイルタグによりノードとフレーバーが照合され、フレーバーがデプロイメントロールに割り当てられます。

ノード一覧を取得して **UUID** を識別します。

```
$ ironic node-list
```

特定のプロファイルにノードを手動でタグ付けする場合には、各ノードの **properties/capabilities** パラメーターに **profile** オプションを追加します。たとえば、2 台のノードをタグ付けしてコントローラープロファイルとコンピュータープロファイルをそれぞれ使用するには、以下のコマンドを実行します。

```
$ ironic node-update 1a4e30da-b6dc-499d-ba87-0bd8a3819bc0 add
properties/capabilities='profile:control,boot_option:local'
$ ironic node-update 6faba1a9-e2d8-4b7c-95a2-c7fbdc12129a add
properties/capabilities='profile:control,boot_option:local'
$ ironic node-update 5e3b2f50-fcd9-4404-b0a2-59d79924b38e add
properties/capabilities='profile:control,boot_option:local'
$ ironic node-update 484587b2-b3b3-40d5-925b-a26a2fa3036f add
properties/capabilities='profile:compute,boot_option:local'
$ ironic node-update d010460b-38f2-4800-9cc4-d69f0d067efe add
properties/capabilities='profile:compute,boot_option:local'
$ ironic node-update d930e613-3e14-44b9-8240-4f3559801ea6 add
properties/capabilities='profile:compute,boot_option:local'
$ ironic node-update da0cc61b-4882-45e0-9f43-fab65cf4e52b add
properties/capabilities='profile:ceph-storage,boot_option:local'
$ ironic node-update b9f70722-e124-4650-a9b1-aade8121b5ed add
properties/capabilities='profile:ceph-storage,boot_option:local'
$ ironic node-update 68bf8f29-7731-4148-ba16-efb31ab8d34f add
properties/capabilities='profile:ceph-storage,boot_option:local'
```

profile オプションを追加すると、適切なプロファイルにノードをタグ付けします。



注記

手動でのタグ付けの代わりに、**Automated Health Check (AHC)** ツールを使用し、ベンチマークデータに基づいて、多数のノードに自動でタグ付けします。

2.5. CEPH STORAGE ノードの ROOT ディスクの定義

Ceph Storage ノードの大半では、複数のディスクを使用します。つまり、Ceph Storage ノードをプロビジョニングする際に、**director** は **root** ディスクに使用するディスクを特定する必要があるという意味です。

- **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
- **wwn** (文字列): 一意のストレージ ID
- **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)

以下の例では、**root** デバイスを特定するディスクのシリアル番号を使用して、オーバークラウドイメージをデプロイするドライブを指定します。

まず、各ノードに使用予定の **root** ドライブのシリアル番号を検索します。ノードごとに **ironic node-show** コマンドを使用して、**extra** セクションから利用可能なブロックデバイスを特定します。たとえば、以下を使用して、全ノードとそのブロックデバイスを一覧表示します。

```
$ for uuid in `ironic node-list | awk '{print $2}'`; do echo "Node ID: $uuid"; ironic node-show $uuid | grep 'properties\|extra ' -A3; done
```

この例では、出力には以下が含まれます。

```
...
Node ID: 97e3f7b3-5629-473e-a187-2193ebe0b5c7
| extra          | {u'newly_discovered': u'true', u'block_devices':      |
|                | {u'serials': [u'100000000', u'100000001',           |
|                | u'100000002', u'100000003', u'100000004',           |
|                | u'100000005', u'100000006', u'100000007',           |
|--
| properties    | {u'cpu_arch': u'x86_64', u'root_device': {u'serial':  |
|                | u'100000005'}}, u'cpus': u'16', u'capabilities':     |
|                | u'profile:ceph-storage,boot_option:local',           |
|                | u'memory_mb': u'65536', u'local_gb': u'3725'}        |
...

```

properties の **capabilities** セクションには、**profile:ceph-storage** が含まれており、UUID が **97e3f7b3-5629-473e-a187-2193ebe0b5c7** のノードは Ceph Storage ノードということが分かります。**block_devices** のパラメーターや各デバイスのシリアル番号の一覧で示されているように、このノードには、複数のブロックデバイスも含まれています。

現在、**root_device** のシリアル番号は **1000000005** に指定されています。この例で、**root** デバイスのシリアル番号を **1000000000** に指定しましょう。これにより、ノードの定義の **root_device** パラメーターに変更を加える必要があります。

```
$ ironic node-update 97e3f7b3-5629-473e-a187-2193ebe0b5c7 add
properties/root_device='{ "serial": "1000000000" }'
```

これにより、**director** が **root** ディスクとして使用する特定のディスクを識別しやすくなります。オーバークラウドの作成の開始時には、**director** はこのノードをプロビジョニングして、オーバークラウドのイメージをこのディスクに書き込みます。

2.6. オーバークラウドでの CEPH STORAGE の有効化

オーバークラウドのイメージにはすでに、Ceph サービスと必要な Puppet モジュールが含まれており、自動的に Ceph OSD ノードと Ceph Monitor をコントローラークラスター上に設定します。オーバークラウドの **heat** テンプレートコレクションには、Ceph Storage 設定を有効化するのに必要な手順も含まれていますが、**director** では Ceph Storage を有効化して、対象の設定に指定するための情報が必要です。この情報を指定するには、**storage-environment.yaml** の環境ファイルを **stack** ユーザーの **templates** ディレクトリーにコピーします。

```
$ cp /usr/share/openstack-tripleo-heat-templates/environments/storage-environment.yaml ~/templates/.
```

storage-environment.yaml のコピーで以下のオプションを変更します。

CinderEnableScsiBackend は iSCSI バックエンドを有効化します。この値は **false** に設定します。**CinderEnableRbdBackend** は Ceph Storage バックエンドを有効化します。この値は **true** に設定します。**CinderEnableNfsBackend** は NFS バックエンドを有効化します。この値は **false** に設定します。

`NovaEnableRbdBackend` は Nova の一時ストレージ用の Ceph Storage を有効化します。この値は `true` に設定します。`GlanceBackend` はバックエンドで Glance を使用するよう定義します。この値は、`rbd` に設定して、イメージに Ceph Storage を使用するようになります。



注記

`storage-environment.yaml` には、Heat を直接使用して Ceph Storage を設定するためのオプションも含まれています。ただし、`director` はこれらのノードを作成して、このシナリオでは、自動的に設定値を定義するので、これらのオプションは必要ありません。

2.7. CEPH STORAGE ノードディスクのレイアウトのマッピング

デフォルトマッピングは、Ceph Storage に root ディスクを使用しますが、実稼動環境の多くは、ストレージやジャーナリング用のパーティションに別個のディスクを複数使用します。このような状況では、以前にコピーした `storage-environment.yaml` ファイルの一部として、ストレージマップを定義します。

`storage-environment.yaml` ファイルを編集してセクションを追加し、以下の内容を記載します。

```
parameter_defaults:
  ExtraConfig:
    ceph::profile::params::osds:
```

このセクションにより、オーバークラウドに Hiera データが追加されます。Puppet は、設定時にこのデータをカスタムのパラメーターとして使用します。`ceph::profile::params::osds` パラメーターを使用して、適切なディスクとジャーナルパーティションをマッピングします。たとえば、4つのディスクがある Ceph ノードでは以下のように割り当てられます。

- `/dev/sda`: オーバークラウドのイメージを含む root ディスク
- `/dev/sdb`: ジャーナルのパーティションが含まれるディスク。これは通常、システムパフォーマンスをソリッドステートドライブ (SSD) です。
- `/dev/sdc` および `/dev/sdd`: OSD のディスク

この例では、以下のような内容が含まれるマッピングとなります。

```
ceph::profile::params::osds:
  '/dev/sdc':
    journal: '/dev/sdb'
  '/dev/sdd':
    journal: '/dev/sdb'
```

ジャーナル用に別のディスクを使用しない場合には、OSD ディスクに併置されているジャーナルを使用します。`journal` パラメーターには空の値を渡します。

```
ceph::profile::params::osds:
  '/dev/sdb': {}
  '/dev/sdc': {}
  '/dev/sdd': {}
```


変更が完了したら、オーバークラウドのデプロイ時に **Ceph Storage** ノードにディスクマッピングが使用されるように **storage-environment.yaml** を保存します。ストレージで必要な設定が開始されるように、デプロイメント内にこのファイルを追加します。

2.8. CEPH STORAGE ノードディスクの GPT への変換

Ceph Storage OSD およびジャーナルのパーティションには、**GPT** ディスクラベルが必要です。これは、**Ceph Storage** にディスクを追加すると、**Ceph OSD** をインストールする前に **GPT** ラベルへ変換する必要があります。これには、ノードでスクリプトを実行して、初回起動時にこの操作が実行されるようにする必要があります。このスクリプトは、オーバークラウドの作成時に **Heat** テンプレートの一部として追加します。たとえば、以下の **heat** テンプレート (**wipe-disks.yaml**) は、**Ceph Storage** ノードの全ディスクをチェックして、全ノード (**root** ファイルシステムを含むディスク以外) を **GPT** に変換します。

```
heat_template_version: 2014-10-16

description: >
  Wipe and convert all disks to GPT (except the disk containing the root
  file system)

resources:
  userdata:
    type: OS::Heat::MultipartMime
    properties:
      parts:
        - config: {get_resource: wipe_disk}

  wipe_disk:
    type: OS::Heat::SoftwareConfig
    properties:
      config: {get_file: wipe-disk.sh}

outputs:
  OS::stack_id:
    value: {get_resource: userdata}
  ....
```

This Heat template makes reference to a Bash script called ``wipe-disk.sh``. This script contains your procedure to wipe the non-root disks. The following script is an example of ``wipe-disk.sh`` that wipes all disks except for the root disk:

```
....
#!/bin/bash
if [[ `hostname` = *"ceph"* ]]
then
  echo "Number of disks detected: $(lsblk -no NAME,TYPE,MOUNTPOINT | grep
"disk" | awk '{print $1}' | wc -l)"
  for DEVICE in `lsblk -no NAME,TYPE,MOUNTPOINT | grep "disk" | awk '{print
$1}'`
  do
    ROOTFOUND=0
    echo "Checking /dev/$DEVICE..."
    echo "Number of partitions on /dev/$DEVICE: $(expr $(lsblk -n
/dev/$DEVICE | awk '{print $7}' | wc -l) - 1)"
    for MOUNTS in `lsblk -n /dev/$DEVICE | awk '{print $7}'`
```

```

do
  if [ "$MOUNTS" = "/" ]
  then
    ROOTFOUND=1
  fi
done
if [ $ROOTFOUND = 0 ]
then
  echo "Root not found in /dev/${DEVICE}"
  echo "Wiping disk /dev/${DEVICE}"
  sgdisk -Z /dev/${DEVICE}
  sgdisk -g /dev/${DEVICE}
else
  echo "Root found in /dev/${DEVICE}"
fi
done
fi

```

環境内に Heat テンプレートを追加するには、**storage-environment.yaml** ファイルにテンプレートを **NodeUserData** リソースとして登録します。

```

resource_registry:
  OS::TripleO::NodeUserData: /home/stack/templates/firstboot/wipe-
  disks.yaml

```



重要

このセクションは、専用のジャーナルを使用した Ceph Storage ノードには適用されません。

2.9. オーバークラウドの作成

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

```

$ openstack overcloud deploy --templates -e /home/stack/templates/storage-
  environment.yaml --control-scale 3 --compute-scale 3 --ceph-storage-scale
  3 --control-flavor control --compute-flavor compute --ceph-storage-flavor
  ceph-storage --neutron-network-type vxlan --ntp-server clock.redhat.com

```

上記のコマンドは、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からオーバークラウドを作成します。
- **-e /home/stack/templates/storage-environment.yaml:** 別の環境ファイルをオーバークラウドデプロイメントに追加します。この場合は、Ceph Storage の設定を含むストレージ環境ファイルです。
- **--control-scale 3:** コントローラーノードを 3 台にスケールリングします。
- **--compute-scale 3:** コンピュートノードを 3 台にスケールリングします。
- **--ceph-storage-scale 3:** Ceph Storage ノードを 3 台にスケールリングします。

- **--control-flavor control**: 対象のコントローラーノードに特定のフレーバーを使用します。
- **--compute-flavor compute**: コンピュートノードに特定のフレーバーを使用します。
- **--ceph-storage-flavor ceph-storage**: コンピュートノードに特定のフレーバーを使用します。
- **--neutron-network-type vxlan**: ネットワーク種別を **neutron** に設定します。
- **--ntp-server clock.redhat.com**: NTP サーバーを設定します。



注記

オプションの完全な一覧を表示するには、以下を実行します。

```
$ openstack help overcloud deploy
```

パラメーターの例については、『Red Hat OpenStack Platform 7 director インストールと使用方法』ガイドの「付録 H. デプロイメントのパラメーター」も参照してください。

オーバークラウドの作成プロセスが開始され、**director** によりノードがプロビジョニングされます。このプロセスは完了するまで多少時間がかかります。オーバークラウドの作成のステータスを確認するには、**stack** ユーザーとして別のターミナルを開き、以下を実行します。

```
$ source ~/stackrc
$ heat stack-list --show-nested
```

2.10. オーバークラウドへのアクセス

director は、**director** ホストがオーバークラウドと対話するための設定と認証を行うスクリプトを生成します。このファイル (**overcloudrc**) は、**stack** ユーザーのホームディレクトリーに保存されます。このファイルを使用するには、以下のコマンドを実行します。

```
$ source ~/overcloudrc
```

これにより、**director** ホストの 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。**director** ホストとの対話に戻るには、以下のコマンドを実行します。

```
$ source ~/stackrc
```

2.11. CEPH STORAGE ノードの監視

オーバークラウドの作成が完了したら、正しく機能していることを確認するため、**Ceph Storage Cluster** のステータスをチェックすることを推奨します。これには、**director** から **heat-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 heat-admin@192.168.0.25
```

クラスターの正常性を確認します。

```
$ sudo ceph health
```

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、クラスターを問題なく使用できることを意味します。

Ceph モニタークォーラムのステータスを確認します。

```
$ sudo 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。

全 Ceph OSD が実行中であるかどうかを確認します。

```
$ ceph osd stat
```

Ceph Storage Cluster の監視に関する詳しい情報は、『[Red Hat Ceph Storage Administration Guide](#)』の「[Chapter 3. Monitoring](#)」を参照してください。

2.12. CEPH STORAGE ノードの置き換え

Ceph Storage ノードに障害が発生する可能性があります。このような状況では、データが失われないように、問題のあるノードを無効化してリバランスしてから、オーバークラウドから削除するようにしてください。以下の手順では、Ceph Storage ノードを置き換えるプロセスについて説明します。



注記

以下の手順では、『[Red Hat Ceph Storage Administration Guide](#)』からの手順を使用して、手動で Ceph Storage ノードを削除します。Ceph Storage ノードの手動での削除に関する詳しい情報は、『[Red Hat Ceph Storage Administration Guide](#)』の「[Chapter 15. Removing OSDs \(Manual\)](#)」を参照してください。

heat-admin ユーザーとして、コントローラーノードまたは Ceph Storage ノードにログインします。**director** の **stack** ユーザーには、**heat-admin** ユーザーにアクセスするための SSH キーがあります。

OSD ツリーを一覧表示して、ノードの OSD を検索します。たとえば、削除するノードには、以下の OSD が含まれる場合があります。

```
-2 0.09998 host overcloud-cephstorage-0 0 0.04999 osd.0 up 1.00000 1.00000 1 0.04999 osd.1 up 1.00000 1.00000
```

Ceph Storage ノードの OSD を無効化します。今回は、OSD ID は 0 と 1 です。

```
[heat-admin@overcloud-controller-0 ~]$ sudo ceph osd out 0
[heat-admin@overcloud-controller-0 ~]$ sudo ceph osd out 1
```

Ceph Storage クラスターがリバランスを開始します。このプロセスが完了するまで待機してください。以下のコマンドを使用して、ステータスを確認できます。

```
[heat-admin@overcloud-controller-0 ~]$ sudo ceph -w
```

Ceph クラスターのリバランスが完了したら、**heat-admin** ユーザーとして、問題のある Ceph Storage ノードにログインして、このノードを停止します。

```
[heat-admin@overcloud-cephstorage-0 ~]$ sudo /etc/init.d/ceph stop osd.0
[heat-admin@overcloud-cephstorage-0 ~]$ sudo /etc/init.d/ceph stop osd.1
```

これ以上データを受信しないように、CRUSH マップからこの Ceph Storage ノードを削除します。

```
[heat-admin@overcloud-cephstorage-0 ~]$ sudo ceph osd crush remove osd.0
[heat-admin@overcloud-cephstorage-0 ~]$ sudo ceph osd crush remove osd.1
```

OSD 認証キーを削除します。

```
[heat-admin@overcloud-cephstorage-0 ~]$ sudo ceph auth del osd.0
[heat-admin@overcloud-cephstorage-0 ~]$ sudo ceph auth del osd.1
```

クラスターから OSD を削除します。

```
[heat-admin@overcloud-cephstorage-0 ~]$ sudo ceph osd rm 0
[heat-admin@overcloud-cephstorage-0 ~]$ sudo ceph osd rm 1
```

ノードからログアウトして、**stack** ユーザーとして **director** ホストに戻ります。

```
[heat-admin@overcloud-cephstorage-0 ~]$ exit
[stack@director ~]$
```

director が再度プロビジョニングしないように、Ceph Storage ノードを無効にします。

```
[stack@director ~]$ ironic node-list
[stack@director ~]$ ironic node-set-maintenance [UUID] true
```

Ceph Storage ノードを削除するには、ローカルのテンプレートファイルを使用して **overcloud** スタックへの更新が必要です。最初に、オーバークラウドスタックの UUID を特定します。

```
$ heat stack-list
```

削除する Ceph Storage ノードの UUID を特定します。

```
$ nova list
```

以下のコマンドを実行してスタックからノードを削除し、それに応じてプランを更新します。

```
$ openstack overcloud node delete --stack [STACK_UUID] --templates -e
[ENVIRONMENT_FILE] [NODE_UUID]
```



重要

オーバークラウドの作成時に追加の環境ファイルを渡した場合には、予定外の変更がオーバークラウドに加えられないように、ここで **-e** または **--environment-file** オプションを使用して環境ファイルを再度渡します。

stack が更新を完了するまで待機します。**heat stack-list --show-nested** を使用して、**stack** の更新を監視します。

`director` のノードプールに新しいノードを追加して、**Ceph Storage** ノードとしてデプロイします。--**ceph-storage-scale** を使用して、オーバークラウド内の合計 **Ceph Storage** ノード数を定義します。たとえば、3つのノードから構成されるクラスターから問題のあるノードを削除して置き換える場合は、--**ceph-storage-scale 3** を使用して、**Ceph Storage** ノードの数を元の値に戻します。

```
$ openstack overcloud deploy --templates --ceph-storage-scale 3 -e  
[ENVIRONMENT_FILES]
```



重要

オーバークラウドの作成時に追加の環境ファイルを渡した場合には、予定外の変更がオーバークラウドに加えられないように、ここで **-e** または **--environment-file** オプションを使用して環境ファイルを再度渡します。

`director` は、新しいノードをプロビジョニングして、新しいノードの詳細を用いて **stack** 全体を更新します。

heat-admin ユーザーとしてコントローラーノードにログインして、**Ceph Storage** ノードのステータスを確認します。以下に例を示します。

```
[heat-admin@overcloud-controller-0 ~]$ sudo ceph status
```

osdmap セクションの値が、クラスターに設定するノード数と一致していることを確認します。

エラーの発生した **Ceph Storage** ノードが新規ノードに置き換えられました。

第3章 既存の CEPH STORAGE CLUSTER のオーバークラウドへの統合

以下のシナリオでは、既存の Ceph Storage Cluster をオーバークラウドのバックエンドとして使用するステップを説明します。以下のシナリオでは、オーバークラウドは 6 台のノードで構成されます。

- 高可用性のコントローラーノード 3 台
- コンピュートノード 3 台

director は、別個の Ceph Storage Cluster と独自のノードをオーバークラウドに統合します。このクラスターは、オーバークラウドから独立して、管理されます。たとえば、Ceph Storage Cluster は、OpenStack Platform **director** ではなく Ceph 管理ツールを使用してスケールリングします。

すべてのマシンは、電源管理に IPMI を使用したベアメタルマシンです。**director** により Red Hat Enterprise Linux 7 のイメージが各ノードにコピーされるため、これらのノードではオペレーティングシステムは必要ありません。

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を使用してコントローラーノードとコンピュートノードと通信します。すべてのノードは、ネイティブの VLAN 経由でネットワークに接続します。この例では、以下の IP アドレスの割り当てで、プロビジョニングサブネットとして 192.0.2.0/24 を使用します。

ノード名	IP アドレス	MAC アドレス	IPMI IP アドレス
director	192.0.2.1	aa:aa:aa:aa:aa:aa	
コントローラー 1	定義済みの DHCP	b1:b1:b1:b1:b1:b1	192.0.2.205
コントローラー 2	定義済みの DHCP	b2:b2:b2:b2:b2:b2	192.0.2.206
コントローラー 3	定義済みの DHCP	b3:b3:b3:b3:b3:b3	192.0.2.207
コンピュート 1	定義済みの DHCP	c1:c1:c1:c1:c1:c1	192.0.2.208
コンピュート 2	定義済みの DHCP	c2:c2:c2:c2:c2:c2	192.0.2.209
コンピュート 3	定義済みの DHCP	c3:c3:c3:c3:c3:c3	192.0.2.210

3.1. 既存の CEPH STORAGE CLUSTER の設定

Ceph クラスターに、**volumes**、**images**、**vms** の 3 つのプールを作成します。以下のコマンドは指標として使用してください。

```
[root@ceph ~]# ceph osd pool create volumes <pg-num>
[root@ceph ~]# ceph osd pool create images <pg-num>
[root@ceph ~]# ceph osd pool create vms <pg-num>
```



注記

配置グループ (**pg-num**) の数には現実的な値を使用します。1 OSDにつき 100 程度を推奨します。たとえば、OSD の合計数を 100 で乗算して、レプリカ数で除算します (**osd pool default size**)。

Ceph クラスターに、以下の機能を持つ **client.openstack** ユーザーを作成します。

- **cap_mon: allow r**
- **cap_osd: allow class-read object_prefix rbd_children, allow rwx pool=volumes, allow rwx pool=vms, allow rwx pool=images**

以下のコマンドは指標として使用してください。

```
[root@ceph ~]# ceph auth add client.openstack mon 'allow r' osd 'allow
class-read object_prefix rbd_children, allow rwx pool=volumes, allow rwx
pool=vms, allow rwx pool=images'
```

3.2. STACK ユーザーの初期化

stack ユーザーとして **director** ホストにログインし、以下のコマンドを実行して **director** の設定を初期化します。

```
$ source ~/stackrc
```

このコマンドでは、**director** の CLI ツールにアクセスする認証情報が含まれる環境変数を設定します。

3.3. ノードの登録

ノード定義のテンプレート (**instackenv.json**) は JSON ファイル形式で、ノード登録用のハードウェアおよび電源管理の情報が含まれています。以下に例を示します。

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



```
    "memory": "6144",
    "disk": "40",
    "arch": "x86_64",
    "pm_type": "pxe_ipmitool",
    "pm_user": "admin",
    "pm_password": "p@55w0rd!",
    "pm_addr": "192.0.2.206"
  },
  {
    "mac": [
      "dd:dd:dd:dd:dd:dd"
    ],
    "cpu": "4",
    "memory": "6144",
    "disk": "40",
    "arch": "x86_64",
    "pm_type": "pxe_ipmitool",
    "pm_user": "admin",
    "pm_password": "p@55w0rd!",
    "pm_addr": "192.0.2.207"
  },
  {
    "mac": [
      "ee:ee:ee:ee:ee:ee"
    ],
    "cpu": "4",
    "memory": "6144",
    "disk": "40",
    "arch": "x86_64",
    "pm_type": "pxe_ipmitool",
    "pm_user": "admin",
    "pm_password": "p@55w0rd!",
    "pm_addr": "192.0.2.208"
  }
  {
    "mac": [
      "ff:ff:ff:ff:ff:ff"
    ],
    "cpu": "4",
    "memory": "6144",
    "disk": "40",
    "arch": "x86_64",
    "pm_type": "pxe_ipmitool",
    "pm_user": "admin",
    "pm_password": "p@55w0rd!",
    "pm_addr": "192.0.2.209"
  }
  {
    "mac": [
      "gg:gg:gg:gg:gg:gg"
    ],
    "cpu": "4",
    "memory": "6144",
    "disk": "40",
    "arch": "x86_64",
    "pm_type": "pxe_ipmitool",
```

```

        "pm_user": "admin",
        "pm_password": "p@55w0rd!",
        "pm_addr": "192.0.2.210"
    }
]
}

```

テンプレートの作成後に、**stack** ユーザーのホームディレクトリー (`/home/stack/instackenv.json`) にファイルを保存してから、**director** にインポートします。これには、以下のコマンドを実行します。

```
$ openstack baremetal import --json ~/instackenv.json
```

このコマンドでテンプレートをインポートして、テンプレートから **director** に各ノードを登録します。カーネルと **ramdisk** イメージを全ノードに割り当てます。

```
$ openstack baremetal configure boot
```

director でのノードの登録、設定が完了しました。

3.4. ノードのハードウェアの検査

ノードの登録後には、各ノードのハードウェア属性を検査します。各ノードのハードウェア属性を検査するには、以下のコマンドを実行します。

```
$ openstack baremetal introspection bulk start
```



重要

このプロセスが最後まで実行されて正常に終了したことを確認してください。ベアメタルの場合には、通常 15 分ほどかかります。

3.5. ノードの手動でのタグ付け

各ノードのハードウェアを登録、検査した後は、特定のプロファイルにノードをタグ付けします。これらのプロファイルタグによりノードとフレーバーが照合され、フレーバーがデプロイメントロールに割り当てられます。

ノード一覧を取得して **UUID** を識別します。

```
$ ironic node-list
```

特定のプロファイルにノードを手動でタグ付けする場合には、各ノードの **properties/capabilities** パラメーターに **profile** オプションを追加します。たとえば、2 台のノードをタグ付けしてコントローラープロファイルとコンピュータープロファイルをそれぞれ使用するには、以下のコマンドを実行します。

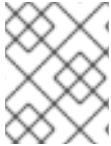
```

$ ironic node-update 1a4e30da-b6dc-499d-ba87-0bd8a3819bc0 add
properties/capabilities='profile:control,boot_option:local'
$ ironic node-update 6faba1a9-e2d8-4b7c-95a2-c7fbdc12129a add
properties/capabilities='profile:control,boot_option:local'
$ ironic node-update 5e3b2f50-fcd9-4404-b0a2-59d79924b38e add

```

```
properties/capabilities='profile:control,boot_option:local'
$ ironic node-update 484587b2-b3b3-40d5-925b-a26a2fa3036f add
properties/capabilities='profile:compute,boot_option:local'
$ ironic node-update d010460b-38f2-4800-9cc4-d69f0d067efe add
properties/capabilities='profile:compute,boot_option:local'
$ ironic node-update d930e613-3e14-44b9-8240-4f3559801ea6 add
properties/capabilities='profile:compute,boot_option:local'
```

profile オプションを追加すると、適切なプロファイルにノードをタグ付けします。



注記

手動でのタグ付けの代わりに、**Automated Health Check (AHC)** ツールを使用し、ベンチマークデータに基づいて、多数のノードに自動でタグ付けします。

3.6. 既存の CEPH STORAGE CLUSTER との統合の有効化

stack ユーザーのホームディレクトリ内のディレクトリに `/usr/share/openstack-tripleo-heat-templates/environments/puppet-ceph-external.yaml` のコピーを作成します。

```
[stack@director ~]# mkdir templates
[stack@director ~]# cp /usr/share/openstack-tripleo-heat-
templates/environments/puppet-ceph-external.yaml ~/templates/.
```

このファイルを編集して、下記のパラメーターを設定します。

- 絶対パスにリソース定義を設定します。

```
OS::TripleO::CephClusterConfig::SoftwareConfig: /usr/share/openstack-
tripleo-heat-templates/puppet/extraconfig/ceph/ceph-external-config.yaml
```

- ファイルを編集して、**Ceph Storage** 環境の情報を使用して、以下の3つのパラメーターを設定します。
- **CephClientKey**
- **CephExternalMonHost**
- **CephClusterFSID**
- 必要な場合は、以下のパラメーターと値を使用して、**OpenStack** プールとクライアントユーザーの名前を設定します。
- **CephClientUserName: openstack**
- **NovaRbdPoolName: vms**
- **CinderRbdPoolName: volumes**
- **GlanceRbdPoolName: images**

3.7. オーバークラウドの作成

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

必要があります。以下に例を示します。

```
$ openstack overcloud deploy --templates -e /home/stack/templates/puppet-
ceph-external.yaml --control-scale 3 --compute-scale 3 --ceph-storage-
scale 0 --control-flavor control --compute-flavor compute --neutron-
network-type vxlan --ntp-server clock.redhat.com
```

上記のコマンドは、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からオーバークラウドを作成します。
- **-e /home/stack/templates/puppet-ceph-external.yaml:** 別の環境ファイルをオーバークラウドデプロイメントに追加します。この場合は、既存の Ceph Storage Cluster の設定を含むストレージ環境ファイルです。
- **--control-scale 3:** コントローラーノードを 3 台にスケールリングします。
- **--compute-scale 3:** コンピュートノードを 3 台にスケールリングします。
- **--ceph-storage-scale 0:** Ceph Storage ノードを 0 にスケールリングします。このように指定すると、director により Ceph Storage ノードが作成されないようになります。
- **--control-flavor control:** 対象のコントローラーノードに特定のフレーバーを使用します。
- **--compute-flavor compute:** コンピュートノードに特定のフレーバーを使用します。
- **--neutron-network-type vxlan:** ネットワーク種別を **neutron** に設定します。
- **--ntp-server clock.redhat.com:** NTP サーバーを設定します。

注記

オプションの完全な一覧を表示するには、以下を実行します。

```
$ openstack help overcloud deploy
```

パラメーターの例については、『Red Hat OpenStack Platform 7 director インストールと使用方法』ガイドの「[付録 H. デプロイメントのパラメーター](#)」も参照してください。

オーバークラウドの作成プロセスが開始され、director によりノードがプロビジョニングされます。このプロセスは完了するまで多少時間がかかります。オーバークラウドの作成のステータスを確認するには、**stack** ユーザーとして別のターミナルを開き、以下を実行します。

```
$ source ~/stackrc
$ heat stack-list --show-nested
```

この設定では、オーバークラウドが外部の Ceph Storage Cluster を使用するように設定します。このクラスターは、オーバークラウドから独立して、管理される点に注意してください。たとえば、Ceph Storage Cluster は、OpenStack Platform director ではなく Ceph 管理ツールを使用してスケールリングします。

3.8. オーバークラウドへのアクセス

director は、**director** ホストがオーバークラウドと対話するための設定と認証を行うスクリプトを生成します。このファイル (**overcloudrc**) は、**stack** ユーザーのホームディレクトリーに保存されます。このファイルを使用するには、以下のコマンドを実行します。

```
$ source ~/overcloudrc
```

これにより、**director** ホストの 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。**director** ホストとの対話に戻るには、以下のコマンドを実行します。

```
$ source ~/stackrc
```

第4章 まとめ

これで、Red Hat Ceph Storage を使用したオープンクラウドの作成および設定が完了しました。オープンクラウド作成後の一般的な機能については、『Red Hat OpenStack Platform 8 director インストールと使用方法』の「[8章 オープンクラウドの作成後のタスク実行](#)」を参照してください。