



Red Hat OpenStack Platform 16.1

ハイパーコンバージドインフラストラクチャーガイド

Red Hat OpenStack Platform オーバークラウドにおけるハイパーコンバージドインフラストラクチャーの設定についての理解

Red Hat OpenStack Platform 16.1 ハイパーコンバージドインフラストラクチャーガイド

Red Hat OpenStack Platform オーバークラウドにおけるハイパーコンバージドインフラストラクチャーの設定についての理解

OpenStack Team
rhos-docs@redhat.com

法律上の通知

Copyright © 2021 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 のハイパーコンバージェンスの実装について説明します。この実装では、Compute サービスと Ceph Storage サービスが同じホストに配置されます。

目次

多様性を受け入れるオープンソースの強化	3
RED HAT ドキュメントへのフィードバックの提供	4
パート I. RED HAT OPENSTACK PLATFORM ハイパーコンバードインフラストラクチャーの設定およびデプロイ	5
第1章 前提条件	6
第2章 ハイパーコンバードノード向けのオーバークラウドロールの準備	7
2.1. マルチディスククラスタのルートディスクの定義	9
第3章 ハイパーコンバードノード上におけるリソース分離の設定	12
3.1. COMPUTE サービス用に確保する CPU およびメモリーリソースの自動生成プロセス	13
3.2. RED HAT CEPH STORAGE のバックフィルとリカバリーの操作	14
第4章 ストレージ管理ネットワークポートの NIC へのマッピング	16
第5章 CEPH STORAGE のデプロイメント前の検証	18
5.1. CEPH-ANSIBLE パッケージバージョンの確認	18
5.2. 事前にプロビジョニングされたノード用のパッケージの確認	18
第6章 HCI オーバークラウドのデプロイ	19
6.1. CEPH-ANSIBLE を実行するノードの限定	20
第7章 ハイパーコンバードノードのスケーリング	22
7.1. HCI 環境におけるハイパーコンバードノードのスケールアップ	22
7.2. HCI 環境におけるハイパーコンバードノードのスケールダウン	22
第8章 OPENSTACK WORKFLOW COMPUTE の CPU およびメモリーの計算	23
第9章 関連資料	25

多様性を受け入れるオープンソースの強化

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

RED HAT ドキュメントへのフィードバックの提供

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

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

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

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

パート I. RED HAT OPENSTACK PLATFORM ハイパーコンバージドインフラストラクチャーの設定およびデプロイ

Red Hat OpenStack Platform (RHOSP) ハイパーコンバージドインフラストラクチャー (HCI) は、ハイパーコンバージドノードで構成されます。リソースの使用率を最適化するために、サービスがこのハイパーコンバージドノード上で共存します。RHOSP HCI では、Compute サービスとストレージサービスがハイパーコンバージドノード上で共存します。ハイパーコンバージドノードのみのオーバークラウド、またはハイパーコンバージドノードを通常のコンピュータノードおよび Ceph Storage ノードと混在させたオーバークラウドをデプロイすることが可能です。



注記

Red Hat Ceph Storage をストレージプロバイダーとして使用する必要があります。

ヒント

- Ceph のメモリー設定を自動的に調整するには、ceph-ansible 3.2 以降を使用します。
- BlueStore のメモリー処理機能を利用するには、BlueStore を HCI デプロイメントのバックエンドとして使用します。

オーバークラウド上に HCI を作成およびデプロイし、オーバークラウドのネットワーク機能仮想化などのその他の機能と統合し、ハイパーコンバージドノード上の Compute サービスと Red Hat Ceph Storage サービス両方のパフォーマンスを最適な状態にするには、以下の手順を実施する必要があります。

1. ハイパーコンバージドノード向けの事前定義されたカスタムオーバークラウドロール **ComputeHCI** を準備する。
2. リソース分離を設定する。
3. ストレージ管理ネットワークのポートを NIC にマッピングする。
4. オーバークラウドをデプロイする。
5. (オプション) ハイパーコンバージドノードをスケールリングする。

第1章 前提条件

- アンダークラウドをデプロイしている。アンダークラウドのデプロイ方法についての説明は、[『Director Installation and Usage』](#) を参照してください。
- お使いの環境で、RHOSP Compute および Red Hat Ceph Storage の要件を満たすノードをプロビジョニング可能である。詳しくは、[『Director Installation and Usage』](#) の [「Basic overcloud deployment」](#) を参照してください。
- 環境内の全ノードを登録している。詳しくは、[『Deploying an overcloud with containerized Red Hat Ceph』](#) の [「Registering nodes」](#) を参照してください。
- 環境内の全ノードがタグ付けされている。詳しくは、[『Deploying an overcloud with containerized Red Hat Ceph』](#) の [「Manually tagging nodes into profiles」](#) を参照してください。
- Compute サービスおよび Ceph OSD サービスに使用するノード上のディスクをクリーンアップしている。詳しくは、[『Deploying an overcloud with containerized Red Hat Ceph』](#) の [「Cleaning Ceph Storage node disks」](#) を参照してください。
- オーバークラウドノードを Red Hat コンテンツ配信ネットワークまたは Red Hat Satellite サーバーに登録するための準備を行っている。詳しくは、[『Advanced Overcloud Customization』](#) の [「Ansible-based overcloud registration」](#) を参照してください。

第2章 ハイパーコンバインドノード向けのオーバークラウドロールの準備

ハイパーコンバインドノードを使用するには、そのノードにロールを定義する必要があります。Red Hat OpenStack Platform (RHOSP) は、ハイパーコンバインドノード向けの事前定義されたロール **ComputeHCI** を提供します。このロールにより、Compute サービスと Ceph オブジェクトストレージ デモン (OSD) サービスを共存させ、同じハイパーコンバインドノード上にまとめてデプロイすることができます。**ComputeHCI** ロールを使用するには、デプロイメントで使用するその他の全ロールに加えて、このロールが含まれるカスタムの **roles_data.yaml** ファイルを生成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. オーバークラウドに使用するその他のロールに加えて **ComputeHCI** ロールが含まれるカスタムの **roles_data.yaml** ファイルを作成します。

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

カスタムロールに関する詳しい情報は、『オーバークラウドの高度なカスタマイズ』の「[コンポーザブルサービスとカスタムロール](#)」および「[roles_data ファイルの検証](#)」を参照してください。

3. **~/templates** に新規環境ファイル (**ports.yaml**) を作成します。
4. **ports.yaml** ファイルに以下の設定を追加して、**ComputeHCI** ロールのポート割り当てを設定します。

```
resource_registry:
  OS::TripleO::ComputeHCI::Ports::ExternalPort: /usr/share/openstack-tripleo-heat-
templates/network/ports/<ext_port_file>.yaml
  OS::TripleO::ComputeHCI::Ports::InternalApiPort: /usr/share/openstack-tripleo-heat-
templates/network/ports/internal_api.yaml
  OS::TripleO::ComputeHCI::Ports::StoragePort: /usr/share/openstack-tripleo-heat-
templates/network/ports/storage.yaml
  OS::TripleO::ComputeHCI::Ports::TenantPort: /usr/share/openstack-tripleo-heat-
templates/network/ports/tenant.yaml
  OS::TripleO::ComputeHCI::Ports::StorageMgmtPort: /usr/share/openstack-tripleo-heat-
templates/network/ports/<storage_mgmt_file>.yaml
```

- **<ext_port_file>** は外部ポートファイルの名前に置き換えてください。DVR を使用している場合は「external」に設定します。それ以外の場合は「noop」に設定します。DVR の詳細は、「[分散仮想ルーター\(DVR\)の設定](#)」を参照してください。
- **<storage_mgmt_file>** は、ストレージ管理ファイルの名前に置き換えます。以下の値のいずれかに設定します。

値	説明
storage_mgmt	IP プールから選択しない場合や、環境で IPv6 アドレスを使用しない場合に使用します。

値	説明
storage_mgmt_from_pool	ComputeHCI ロールが IP プールから選択するようにする場合に使用します。
storage_mgmt_v6	環境で IPv6 アドレスを使用する場合に使用します。
storage_mgmt_from_pool_v6	ComputeHCI ロールが IPv6 アドレスプールから選択するようにする場合に使用します。

詳細は、『Advanced Overcloud Customization』の「[Basic network isolation](#)」を参照してください。

5. ComputeHCI ロール用のフレーバーを作成します。

```
(undercloud)$ openstack flavor create --id auto \
--ram 6144 --disk 40 --vcpus 4 computeHCI
```

6. フレーバーの属性を設定します。

```
(undercloud)$ openstack flavor set --property "cpu_arch"="x86_64" \
--property "capabilities:boot_option"="local" \
--property "resources:CUSTOM_BAREMETAL"="1" \
--property "resources:DISK_GB"="0" \
--property "resources:MEMORY_MB"="0" \
--property "resources:VCPU"="0" computeHCI
```

7. フレーバーを新規プロファイルにマッピングします。

```
(undercloud)$ openstack flavor set \
--property "capabilities:profile"="computeHCI" computeHCI
```

8. ノード一覧を取得して UUID を把握します。

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

9. ノードを新規プロファイルにタグ付けします。

```
(undercloud)$ openstack baremetal node set \
--property capabilities='profile:computeHCI,boot_option:local' <UUID>
```

詳細は、『Deploying an overcloud with containerized Red Hat Ceph』の「[Manually tagging nodes into profiles](#)」および「[Assigning nodes and flavors to roles](#)」を参照してください。

10. **computeHCI** フレーバーを ComputeHCI ロールに関連付けるには、以下の設定を **node-info.yaml** ファイルに追加します。

```
parameter_defaults:
  OvercloudComputeHCIFlavor: computeHCI
  ComputeHCICount: 3
```

2.1. マルチディスククラスターのルートディスクの定義

ノードで複数のディスクが使用されている場合には、director はプロビジョニング時にルートディスクを特定する必要があります。たとえば、ほとんどの Ceph Storage ノードでは、複数のディスクが使用されます。デフォルトのプロビジョニングプロセスでは、director はルートディスクにオーバークラウドイメージを書き込みます。

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): ディスクを用いるデバイス (HDD) の場合は true、それ以外 (SSD) の場合は false。
- **name** (文字列): デバイス名 (例: /dev/sdb1)



重要

name 属性は、永続的なデバイス名が付いたデバイスにのみ使用します。他のデバイスのルートディスクを設定する際に、**name** を使用しないでください。この値は、ノードのブート時に変更される可能性があります。

シリアル番号を使用してルートデバイスを指定するには、以下の手順を実施します。

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。以下のコマンドを実行して、ノードのディスク情報を表示します。

```
(undercloud) $ openstack baremetal introspection data save 1a4e30da-b6dc-499d-ba87-0bd8a3819bc0 | jq ".inventory.disks"
```

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. **openstack baremetal node set --property root_device=** を入力して、ノードのルートディスクを設定します。ルートディスクを定義するのに最も適切なハードウェア属性値を指定します。

```

(undercloud) $ openstack baremetal node set --property root_device='{ "serial":
<serial_number> }' <node-uuid>

```

たとえば、ルートデバイスをシリアル番号が **61866da04f380d001ea4e13c12e36ad6** の disk 2 に設定するには、以下のコマンドを入力します。

```

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

```



注記

各ノードの BIOS を設定して、選択したルートディスクからの起動を含めるようにします。最初にネットワークからのブートを試み、次にルートディスクからのブートを試みるように、ブート順序を設定します。

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 deploy** 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。

第3章 ハイパーコンバインドノード上におけるリソース分離の設定

ハイパーコンバインドノード上で Ceph OSD サービスと Compute サービスを共存させると、お互いが同じホスト上に存在することを認識しないため、Red Hat Ceph Storage サービスと Compute サービス間でリソースの競合が発生するリスクがあります。リソースの競合が発生すると、サービスのパフォーマンスが低下し、ハイパーコンバージェンスの利点が打ち消される可能性があります。

Ceph サービスと Compute サービスの両方にリソースの分離を設定して、競合が発生を防ぐ必要があります。

手順

1. (オプション) Compute 環境ファイルに以下のパラメーターを追加して、自動生成される Compute 設定を上書きします。

```
parameter_defaults:
  ComputeHCIParameters:
    NovaReservedHostMemory: <ram>
    NovaCPUAllocationRatio: <ratio>
```

- **<ram>** を、ハイパーコンバインドノード上で Ceph OSD サービスおよびインスタンスのオーバーヘッド用に確保する RAM 容量 (MB 単位) に置き換えます。
 - **<ratio>** を、インスタンスをデプロイするコンピューターノードを選択する際に Compute スケジューラーが使用すべき比率に置き換えます。
自動生成される Compute 設定についての詳細は、「[Compute サービス用に確保する CPU およびメモリーリソースの自動生成プロセス](#)」を参照してください。
2. Red Hat Ceph Storage 用にメモリーリソースを確保するには、`/home/stack/templates/storage-container-config.yaml` で `is_hci` パラメーターを `true` に設定します。

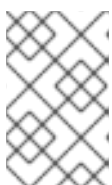
```
parameter_defaults:
  CephAnsibleExtraConfig:
    is_hci: true
```

この設定により、**ceph-ansible** は Red Hat Ceph Storage 用のメモリーリソースを確保することができ、HCI デプロイメントの `osd_memory_target` パラメーター設定を自動的に調整することで、Ceph OSD によるメモリー増大を低減することができます。



警告

Red Hat では、`ceph_osd_docker_memory_limit` パラメーターを直接上書きすることは推奨しません。



注記

FileStore または BlueStore どちらのバックエンドが使用されていても、ceph-ansible 3.2 では、`ceph_osd_docker_memory_limit` は Ansible の検出したホストの最大メモリーに自動的に設定されます。

- (オプション) デフォルトでは、**ceph-ansible** は Ceph OSD ごとに1つの仮想 CPU を確保します。Ceph OSD ごとに複数の CPU が必要な場合には、以下の設定を `/home/stack/templates/storage-container-config.yaml` に追加します。

```
parameter_defaults:
  CephAnsibleExtraConfig:
    ceph_osd_docker_cpu_limit: <cpu_limit>
```

<cpu_limit> を、各 Ceph OSD 用に確保する CPU 数に置き換えます。

ハードウェアおよびワークロードに基づいて CPU リソースを調整する方法の詳細は、『[Red Hat Ceph Storage Hardware Selection Guide](#)』を参照してください。

- (オプション) 以下のパラメーターを Ceph 環境ファイルに追加することで、Ceph OSD の削除時に Red Hat Ceph Storage のバックフィルとリカバリーの操作の優先度を低くします。

```
parameter_defaults:
  CephConfigOverrides:
    osd_recovery_op_priority: <priority_value>
    osd_recovery_max_active: <no_active_recovery_requests>
    osd_max_backfills: <max_no_backfills>
```

- <priority_value> を、OSD クライアント OP の優先度と相対的に、リカバリーの操作の優先度に置き換えます。
- <no_active_recovery_requests> を、1 OSD あたりの1回にアクティブなリカバリー要求の件数に置き換えます。
- <max_no_backfills> を、単一の OSD との間で許容されるバックフィルの最大数に置き換えます。
デフォルトの Red Hat Ceph Storage のバックフィルおよびリカバリーオプションに関する詳細は、『[Red Hat Ceph Storage のバックフィルとリカバリーの操作](#)』を参照してください。

3.1. COMPUTE サービス用に確保する CPU およびメモリーリソースの自動生成プロセス

director の提供するデフォルトのプラン環境ファイルにより、デプロイメント時のハイパーコンバージドノードのリソース制約が設定されます。このプラン環境ファイルは、OpenStack Workflow に以下のプロセスを実施するように指示します。

1. [ノードのハードウェアの検査](#) 時に収集したハードウェアイントロスペクションデータを取得する。
2. そのデータに基づき、ハイパーコンバージドノード上の Compute の最適な CPU およびメモリー割り当て負荷を算出する。
3. これらの制約を設定し Compute に CPU/メモリーリソースを確保するのに必要なパラメーターを自動生成する。これらのパラメーターは、**plan-environment-derived-params.yaml** ファイルの `hci_profile_config` セクションで定義されます。



注記

Compute の **reserved_host_memory** および **cpu_allocation_ratio** の設定値を算出するのに、各ワークロードプロファイルの **average_guest_memory_size_in_mb** および **average_guest_cpu_utilization_percentage** パラメーターが使用されます。

Compute 環境ファイルに以下のパラメーターを追加して、自動生成される Compute 設定を上書きすることができます。

自動生成される nova.conf パラメーター	Compute 環境ファイルのオーバーライド	説明
reserved_host_memory	<pre>parameter_defaults: ComputeHCIParameters: NovaReservedHostMemory: 181000</pre>	ハイパーコンバージドノード上で、Ceph OSD サービスおよびゲストインスタンスごとのオーバーヘッドに確保する RAM 容量を設定します。
cpu_allocation_ratio	<pre>parameter_defaults: ComputeHCIParameters: NovaCPUAllocationRatio: 8.2</pre>	インスタンスをデプロイするコンピュータノードを選択する際に Compute スケジューラーが使用すべき比率を設定します。

これらのオーバーライドは、ComputeHCI ロールを使用するすべてのノード (つまり、すべてのハイパーコンバージドノード) に適用されます。 **NovaReservedHostMemory** および **NovaCPUAllocationRatio** の最適な値を手動で決定する方法に関する詳しい情報は、「[8章 OpenStack Workflow Compute の CPU およびメモリーの計算](#)」を参照してください。

ヒント

以下のスクリプトを使用して、ハイパーコンバージドノードの **NovaReservedHostMemory** および **NovaCPUAllocationRatio** の適切な基準値を算出することができます。

[nova_mem_cpu_calc.py](#)

3.2. RED HAT CEPH STORAGE のバックフィルとリカバリーの操作

Ceph OSD が削除されると、Red Hat Ceph Storage はバックフィルおよびリカバリー操作を使用してクラスターをリバランスします。Red Hat Ceph Storage は配置グループポリシーに従って、データのコピーを複数保管するためにこの操作を実行します。これらの操作は、システムリソースを使用します。Red Hat Ceph Storage クラスターに負荷がかかっている場合、リソースがバックフィルおよびリカバリーに回されるので、パフォーマンスが低下します。

OSD 削除時のこのパフォーマンスへの影響を軽減するには、バックフィルおよびリカバリー操作の優先度を低くすることができます。この手法のマイナス面は、データのレプリカがより少ない状態が長くなるので、データがリスクにさらされる可能性が若干高くなることです。

以下の表で説明するパラメーターが、バックフィルおよびリカバリー操作の優先度を設定するのに使用されます。

パラメーター	説明	デフォルト値
osd_recovery_op_priority	OSD クライアントの操作に関して、リカバリー操作の優先度を設定します。	3
osd_recovery_max_active	同時に要求できる1 OSD あたりのアクティブなリカバリー要求の数を設定します。要求が増えるにつれてリカバリーは加速されますが、それらの要求によりクラスターにかかる負荷が増大します。レイテンシーを低くするには、このパラメーターを1に設定します。	3
osd_max_backfills	単一の OSD との間で許容されるバックフィルの最大数を設定します。	1

第4章 ストレージ管理ネットワークポートの NIC へのマッピング

ストレージ管理ネットワークのポートをハイパーコンバインドノードの物理 NIC にマッピングする必要があります。

手順

1. お使いの環境の `/usr/share/openstack-tripleo-heat-templates/network/config/<nic_config>/compute.yaml` ファイルを、環境ファイルのディレクトリにコピーして、その名前を変更します。

```
$ cp /usr/share/openstack-tripleo-heat-templates/network/config/<nic_config>/compute.yaml
/home/stack/templates/nic-configs/compute-hci.yaml
```

`<nic_config>` を、お使いの環境の `compute.yaml` ファイルが含まれる NIC 設定フォルダーに置き換えます。各 NIC 設定の詳細については、各フォルダーの `README.md` を参照してください。

2. `compute-hci.yaml` ファイルにまだ以下の `StorageMgmtNetworkVlanID` 定義がなければ、その定義を追加します。

```
StorageMgmtNetworkVlanID:
  default: 40
  description: Vlan ID for the storage mgmt network traffic.
  type: number
```

3. 各 HCI ノードで `StorageMgmtNetworkVlanID` を特定の NIC にマッピングします。たとえば、VLAN を単一の NIC にトランク接続する場合、`compute-hci.yaml` ファイルの `network_config:` セクションに以下のエントリーを追加します。

```
type: vlan
device: em2
mtu: 9000
use_dhcp: false
vlan_id: {get_param: StorageMgmtNetworkVlanID}
addresses:
  -
    ip_netmask: {get_param: StorageMgmtIpSubnet}
```



注記

NIC を `StorageMgmtNetworkVlanID` にマッピングする際に、MTU を 9000 (ジャンボフレーム) に設定して、Red Hat Ceph Storage のパフォーマンスを向上させます。詳細は、『ネットワークガイド』の「[director での MTU 設定の定義](#)」および『Advanced Overcloud Customization』の「[ジャンボフレームの設定](#)」を参照してください。

4. `~/templates/network.yaml` という名前でネットワークの環境ファイルを作成します。
5. オーバークラウドのデプロイメント中に Compute NIC テンプレート `compute-hci.yaml` を呼び出すには、以下の設定を `network.yaml` ファイルに追加します。

resource_registry:

OS::TripleO::ComputeHCL::Net::SoftwareConfig: /home/stack/templates/nic-configs/compute-hci.yaml

第5章 CEPH STORAGE のデプロイメント前の検証

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

5.1. CEPH-ANSIBLE パッケージバージョンの確認

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

手順

ceph-ansible パッケージの修正バージョンがインストールされていることを確認してください。

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

5.2. 事前にプロビジョニングされたノード用のパッケージの確認

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

事前にプロビジョニングされたノードの詳細は、「[事前にプロビジョニングされたノードを使用した基本的なオーバークラウドの設定](#)」を参照してください。

手順

サーバーに必要なパッケージが含まれていることを確認します。

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

第6章 HCI オーバークラウドのデプロイ

すべてのHCI設定が完了したら、オーバークラウドをデプロイする必要があります。デプロイオプションについての詳細は、『[directorのインストールと使用方法](#)』の「[CLI ツールを使用した基本的なオーバークラウドの設定](#)」を参照してください。



重要

Red Hat OpenStack Platform (RHOSP) HCI 環境をデプロイする際には、インスタンス HA を有効にしないでください。Red Hat Ceph Storage を使用したハイパーコンバージド RHOSP デプロイメントでインスタンス HA を使用する場合は、Red Hat の担当者にお問い合わせください。

前提条件

- その他すべての Red Hat Ceph Storage の設定に、別のベース環境ファイルを1つ (または複数) 使用している (例: `/home/stack/templates/storage-config.yaml`)。詳しくは、『[Deploying an overcloud with containerized Red Hat Ceph](#)』の「[Customizing the Storage service](#)」および「[Sample environment file: creating a Ceph Storage cluster](#)」を参照してください。
- ベース環境ファイルで、各ロールに割り当てるノード数を定義している。詳細は、『[Deploying an overcloud with containerized Red Hat Ceph](#)』の「[Assigning nodes and flavors to roles](#)」を参照してください。
- 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。設定していない場合は、『[Advanced Overcloud Customization](#)』の「[Enabling SSL/TLS on Overcloud Public Endpoints](#)」で説明するように、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。

手順

- その他の環境ファイルと共に新しいロールファイルおよび環境ファイルをスタックに追加して、HCI オーバークラウドをデプロイします。

```
(undercloud)$ openstack overcloud deploy --templates \
  -e [your environment files] \
  -r /home/stack/templates/roles_data_hci.yaml \
  -e /home/stack/templates/ports.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible.yaml \
  -e /home/stack/templates/storage-config.yaml \
  -e /home/stack/templates/storage-container-config.yaml \
  -e /home/stack/templates/network.yaml \
  [-e /home/stack/templates/ceph-backfill-recovery.yaml \]
  --ntp-server pool.ntp.org
```

デプロイメントコマンドに `/usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible.yaml` を指定すると、すべてのデフォルト設定と共にコンテナ化された Red Hat Ceph クラスタをデプロイするベース環境ファイルが追加されます。詳しくは、『[Deploying an overcloud with containerized Red Hat Ceph](#)』を参照してください。



注記

デプロイメントで Single Root Input/Output Virtualization (SR-IOV) を使用する場合は、デプロイメントコマンドに以下のオプションを追加します。

- **-e /usr/share/openstack-tripleo-heat-templates/environments/services/neutron-sriov.yaml**
- **-e /home/stack/templates/network-environment.yaml**

ヒント

アンサー ファイルを使用して、デプロイメントに追加する環境ファイルを指定することも可能です。詳しい情報は、『[director のインストールと使用方法](#)』の「[オーバークラウドデプロイメントへの環境ファイルの追加](#)」を参照してください。

6.1. CEPH-ANSIBLE を実行するノードの限定

ceph-ansible を実行するノードを限定することで、デプロイメントの更新時間を短縮することができます。Red Hat OpenStack Platform (RHOSP) で Ceph の設定に **config-download** が使用されている場合、デプロイメント全体に対して **config-download** および **ceph-ansible** を実行する代わりに、**--limit** オプションを使用してノードの一覧を指定することができます。この機能は、たとえばオーバークラウドをスケールアップする場合や障害の発生したディスクを置き換える場合に役立ちます。このようなシナリオでは、環境に追加する新規ノードでのみデプロイメントを実行することができます。

障害の発生したディスクの置き換え時に **--limit** を使用するシナリオの例

以下の手順例では、Ceph ストレージノード **oc0-cephstorage-0** でディスク障害が発生したため、ベンダーでフォーマット済みの新規ディスクを受け入れます。新しいディスクを OSD として使用できるように、Ansible を **oc0-cephstorage-0** ノード上で実行する必要があります。しかし、その他すべての Ceph Storage ノードで実行する必要はありません。例として記述した環境ファイルおよびノードの名前を、実際の環境に適した名前に置き換えてください。

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、**source** コマンドで **stackrc** 認証情報ファイルを読み込みます。

```
# source stackrc
```

2. 新規ディスクが不足している OSD を起動するのに使用されるように、以下の手順の1つを実施します。

- **--limit** オプションを使用して **ceph-ansible** を実行するノードを指定し、スタックの更新を実行する。

```
$ 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/ceph-ansible/ceph-ansible.yaml \
  -e ~/my-ceph-settings.yaml \
  -e <other-environment_files> \
  --limit oc0-controller-0:oc0-controller-2:oc0-controller-1:oc0-cephstorage-0:undercloud
```


この例では、Ceph mon の OSD 定義を変更するのに Ansible が必要なため、コントローラーが含まれています。

- **config-download** で **ansible-playbook-command.sh** スクリプトが生成されている場合は、**--limit** オプションを指定してスクリプトを実行して、指定したノードを **ceph-ansible** に渡すこともできます。

```
./ansible-playbook-command.sh --limit oc0-controller-0:oc0-controller-2:oc0-controller-1:oc0-cephstorage-0:undercloud
```

警告

必ずアンダークラウドを制限一覧に含める必要があります。含めないと、**--limit** を使用する際に **ceph-ansible** を実行することができません。この注意が必要なのは、アンダークラウドでのみ実行される **external_deploy_steps_tasks** Playbook により **ceph-ansible** が実行されるためです。

第7章 ハイパーコンバージドノードのスケールリング

HCI ノードをスケールアップまたはスケールダウンする場合、コンピュータードまたは Red Hat Ceph Storage ノードのスケールリングと同じ原則および手法が適用されます。

7.1. HCI 環境におけるハイパーコンバージドノードのスケールアップ

HCI 環境のハイパーコンバージドノードをスケールアップするには、ハイパーコンバージドノードではないノードをスケールアップするのと同じ手順に従います。詳しくは、『Director Installation and Usage』の「[Adding nodes to the overcloud](#)」を参照してください。



注記

新規ノードをタグ付けする場合には、必ず適切なフレーバーを使用するようにしてください。

OSD を Red Hat Ceph Storage クラスタに追加して HCI ノードをスケールアップする方法は、『[コンテナ化された Red Hat Ceph を持つオーバークラウドのデプロイ](#)』の「[OSD の Ceph Storage ノードへの追加](#)」を参照してください。

7.2. HCI 環境におけるハイパーコンバージドノードのスケールダウン

HCI 環境のハイパーコンバージドノードをスケールダウンするには、HCI ノード上の Ceph OSD サービスをリバランスし、HCI ノードからインスタンスを移行し、オーバークラウドからコンピュータードを削除する必要があります。

手順

1. HCI ノード上の Ceph OSD サービスを無効にして、リバランスします。HCI ノードまたは Red Hat Ceph Storage ノードを削除する際に、director は Red Hat Ceph Storage クラスタを自動的にリバランスしないので、このステップが必要となります。
2. HCI ノードからインスタンスを移行します。詳細については、『[インスタンス作成のための Compute サービスの設定](#)』の「[コンピュータード間の仮想マシンインスタンスの移行](#)」を参照してください。
3. オーバークラウドからコンピュータードを削除します。詳しくは、「[コンピュータードの削除](#)」を参照してください。

第8章 OPENSTACK WORKFLOW COMPUTE の CPU およびメモリーの計算

OpenStack Workflow は CPU とメモリーに最適な設定を計算し、その結果を使用してパラメーター **NovaReservedHostMemory** および **NovaCPUAllocationRatio** に反映させます。

NovaReservedHostMemory

NovaReservedHostMemory パラメーターは、ホストノードに確保するメモリー容量 (MB 単位) を設定します。ハイパーコンバージドノードの適切な値を決定するには、各 OSD が 3 GB のメモリーを消費すると仮定します。メモリーが 256 GB で OSD が 10 のノードの場合には、Ceph に 30 GB のメモリーを割り当てて、226 GB を Compute に残します。このメモリー容量のノードでは、たとえば、それぞれ 2 GB のメモリーを使用するインスタンスを 113 台ホストすることができます。

ただし、ハイパーバイザー用に、インスタンス 1 台あたりの追加のオーバーヘッドを考慮する必要があります。このオーバーヘッドを 0.5 GB と仮定すると、同じノードがホストすることのできるインスタンスは 90 台だけになります (226 GB を 2.5 GB で除算する)。ホストノードに確保するメモリー容量 (Compute サービスが使用してはならないメモリー) は以下のように算出します。

$(In * Ov) + (Os * RA)$

各オプションについての説明は以下のとおりです。

- **In**: インスタンス数
- **Ov**: インスタンス 1 台あたりに必要なオーバーヘッド用のメモリー容量
- **Os**: ノード上の OSD 数
- **RA**: 各 OSD に割り当てる必要のある RAM 容量

90 台のインスタンスの場合には、 $(90 * 0.5) + (10 * 3) = 75$ GB という計算になります。Compute サービスには、この値を MB 単位で指定します (75000)。

以下の Python コードにより、この計算を行うことができます。

```
left_over_mem = mem - (GB_per_OSD * osds)
number_of_guests = int(left_over_mem /
    (average_guest_size + GB_overhead_per_guest))
nova_reserved_mem_MB = MB_per_GB * (
    (GB_per_OSD * osds) +
    (number_of_guests * GB_overhead_per_guest))
```

NovaCPUAllocationRatio

Compute スケジューラーは、インスタンスをデプロイする Compute ノードを選択する際に **NovaCPUAllocationRatio** を使用します。デフォルトでは、この値は **16.0** (16:1) です。ノードに 56 のコアがある場合には、Compute スケジューラーは 896 の仮想 CPU を使用するのに十分なインスタンスをノードにスケジュールすることになります。この値を超えると、そのノードはそれ以上インスタンスをホストできないと見なされます。

ハイパーコンバージドノードに適切な **NovaCPUAllocationRatio** を決定するには、各 Ceph OSD が最小で 1 コアを使用すると仮定します (ワークロードが I/O 集中型で、SSD を使用しないノード上にある場合を除く)。56 コア、10 OSD のノードでは、この計算で 46 コアが Compute に確保されます。各インスタンスが割り当てられた CPU を 100 パーセント使用すと仮定すると、この比率は単にインスタン

スの仮想 CPU 数をコア数で除算した値となります (46 / 56 = 0.8)。ただし、通常インスタンスは割り当てられた CPU を 100 パーセント使用することはないため、必要なゲスト仮想 CPU の数を決定する際には、予想される使用率を考慮に入れて **NovaCPUAllocationRatio** を高くすることができます。

したがって、インスタンスが仮想 CPU の 10 パーセント (または 0.1) のみを使用すると予想できる場合には、インスタンス用の仮想 CPU 数は $46 / 0.1 = 460$ の式で表すことができます。この値をコア数 (56) で除算すると、比率は約 8 に増えます。

以下の Python コードにより、この計算を行うことができます。

```
cores_per_OSD = 1.0
average_guest_util = 0.1 # 10%
nonceph_cores = cores - (cores_per_OSD * osds)
guest_vCPUs = nonceph_cores / average_guest_util
cpu_allocation_ratio = guest_vCPUs / cores
```

第9章 関連資料

Red Hat OpenStack Platform (RHOSP) に関する詳しい情報は、以下のガイドを参照してください。

- [Director Installation and Usage](#): このガイドは、RHOSP 環境 (アンダークラウドおよびオーバークラウドの両方) のエンドツーエンドのデプロイメントに関するガイダンスを提供します。
- [Advanced Overcloud Customization](#): このガイドでは、director を使用して RHOSP の高度な機能を設定する方法について記載しています (例: カスタムロールの使用方法)。
- [Deploying an overcloud with containerized Red Hat Ceph](#) : このガイドでは、Red Hat Ceph Storage をストレージプロバイダーとして使用するオーバークラウドのデプロイ方法について記載しています。
- [Networking Guide](#): このガイドでは、RHOSP のネットワーク設定タスクについて詳細に説明しています。