



Red Hat OpenStack Platform 17.1

大規模な Red Hat OpenStack Platform のデプロイ

大規模なデプロイメントにおけるハードウェア要件および推奨事項

Red Hat OpenStack Platform 17.1 大規模な Red Hat OpenStack Platform のデプロイ

大規模なデプロイメントにおけるハードウェア要件および推奨事項

OpenStack Team
rhos-docs@redhat.com

法律上の通知

Copyright © 2024 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

このガイドには、大規模な Red Hat OpenStack Platform をデプロイする際のさまざまな推奨事項を記載します。推奨事項には、ハードウェアの推奨事項、アンダークラウドのチューニング、およびオーバークラウドの設定が含まれます。

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 (英語のみ)	4
第1章 大規模デプロイメントにおける推奨事項	5
第2章 大規模な RED HAT OPENSTACK デプロイメントで推奨される仕様	6
2.1. アンダークラウドのシステム要件	6
2.2. オーバークラウドコントローラーノードのシステム要件	6
2.3. オーバークラウドコンピュートノードのシステム要件	9
2.4. RED HAT CEPH STORAGE ノードのシステム要件	10
第3章 RED HAT OPENSTACK デプロイメントのベストプラクティス	11
3.1. RED HAT OPENSTACK デプロイメントの準備	11
3.2. RED HAT OPENSTACK のデプロイメント設定	12
3.3. アンダークラウドのチューニング	16

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、用語の置き換えは、今後の複数のリリースにわたって段階的に実施されます。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。

Jira でドキュメントのフィードバックを提供する

問題の作成 フォームを使用して、Red Hat OpenStack Services on OpenShift (RHOSO) または Red Hat OpenStack Platform (RHOSP) の以前のリリースのドキュメントに関するフィードバックを提供します。RHOSO または RHOSP ドキュメントの問題を作成すると、その問題は RHOSO Jira プロジェクトに記録され、フィードバックの進行状況を追跡できるようになります。

問題の作成 フォームを完了するには、Jira にログインしていることを確認してください。Red Hat Jira アカウントをお持ちでない場合は、<https://issues.redhat.com> でアカウントを作成できます。

1. 次のリンクをクリックして、**問題の作成** ページを開きます (**問題の作成**)。
2. **Summary** フィールドと **Description** フィールドに入力します。**Description** フィールドに、ドキュメントの URL、章またはセクション番号、および問題の詳しい説明を入力します。フォーム内の他のフィールドは変更しないでください。
3. **Create** をクリックします。

第1章 大規模デプロイメントにおける推奨事項

大規模な Red Hat OpenStack Platform (RHOSP) 環境をデプロイする際には、以下のアンダークラウドおよびオーバークラウドの推奨事項、仕様、および設定を使用します。100 を超えるオーバークラウドノードが含まれる RHOSP 17.1 デプロイメントは、大規模な環境とみなされます。Red Hat は、ミニオンを使用しない RHOSP 17.1 を使用して、750 のオーバークラウドノードが含まれる大規模な環境で、最適なパフォーマンスをテストおよび検証しています。

DCN ベースのデプロイメントの場合、中央サイトおよびエッジサイトからのノードの数が非常に大きいことがあります。DCN デプロイメントに関する推奨事項は、Red Hat Global Support Services にお問い合わせください。

第2章 大規模な RED HAT OPENSTACK デプロイメントで推奨される仕様

提供される推奨事項を使用して、大規模なクラスターデプロイメントをスケーリングできます。

以下の手順で示す値は、Red Hat OpenStack Platform Performance & Scale Team が実施したテストに基づくもので、個々の環境によって異なる可能性があります。

2.1. アンダークラウドのシステム要件

最適なパフォーマンスを得るには、物理サーバーにアンダークラウドノードをインストールします。ただし、仮想アンダークラウドノードを使用する場合、仮想マシンには、以下の表で説明されている物理マシンと同様に、十分なリソースを確保するようにしてください。

表2.1 推奨されるアンダークラウドノードの仕様

システム要件	説明
ノード数	1
CPU	32 コア、64 スレッド
ディスク	500 GB のルートディスク (1x SSD または 2x 7200 RPM のハードドライブ (RAID 1))
メモリー	256 GB
ネットワーク	25 Gbps のネットワークインターフェイスまたは 10 Gbps のネットワークインターフェイス

2.2. オーバークラウドコントローラーノードのシステム要件

すべてのコントロールプレーンサービスは、3つのノードで稼働する必要があります。通常、すべてのコントロールプレーンサービスは3つのコントローラーノードに分散してデプロイされます。

コントローラーサービスのスケーリング

コントローラーサービスで利用可能なリソースを増やすには、これらのサービスを追加のノードにスケーリングします。たとえば、**db** または **messaging** コントローラーサービスを専用ノードにデプロイして、コントローラーノードの負荷を軽減できます。

コントローラーサービスをスケーリングするには、スケーリングするサービスのセットをコンポーザブルロールを使用して定義します。コンポーザブルロールを使用する場合は、各サービスは3つの追加の専用ノードで実行される必要があります。Pacemaker のクォーラムを維持するために、コントロールプレーン内のノードの合計数を追加する必要があります。

この例のコントロールプレーンは、以下に示す9台のノードで構成されます。

- コントローラーノード 3 台
- データベースノード 3 台

- メッセージングノード 3 台

詳細は、Red Hat OpenStack Platform デプロイメントのカスタマイズの [コンポーザブルサービスとカスタムロール](#) を参照してください。

コンポーザブルロールを使用したコントローラーサービスのスケーリングに関する質問については、[Red Hat Global Consulting](#) にお問い合わせください。

ストレージに関する考慮事項

オーバークラウドデプロイメントのコントローラーノードを計画する場合は、十分なストレージを準備します。

デプロイメントに Ceph Storage が含まれていない場合は、オーバークラウドのワークロードまたは Image (glance) サービスが使用する Object Storage (swift) 用に、専用のディスクまたはノードを使用してください。コントローラーノードで Object Storage を使用する場合は、オブジェクトデータ保存時のディスク使用率を削減するために、ルートディスクとは別に NVMe デバイスを使用してください。

Block Storage サービス (cinder) では、ボリュームを Image Storage サービス (glance) にアップロードするために、大規模な同時操作を実行する必要があります。イメージはコントローラーディスクにかなりの I/O 負荷をかけます。推奨されるワークフローではありませんが、必要な場合は、コントローラーノードで SSD ディスクを使用して、一括操作のために高い IOPS を確保してください。



注記

- Ceilometer、gnocchi、および Alarming サービス (aodh) に基づく古い Telemetry サービスは、デフォルトで無効になっており、パフォーマンスへの悪影響があるため推奨されません。これらの Telemetry サービスを有効にすると、gnocchi が大量の I/O を消費し、Ceph が有効でなくてもメトリクスを Object Storage ノードに送信します。
- 大規模なテストはすべて、Director でデプロイされた Ceph クラスタを備えた環境で行われています。

CPU に関する考慮事項

コントローラーノードが受け取る API 呼び出し、AMQP メッセージ、およびデータベースクエリーの数が、コントローラーノードでの CPU メモリー消費に影響を与えます。各 Red Hat OpenStack Platform (RHOSP) コンポーネントがタスクを同時に処理および実行する能力は、個々の RHOSP コンポーネントに設定されるワーカースレッドの数によっても制限されます。パフォーマンスの低下を避けるために、コンポーネントがコントローラーノードに多数のタスクを持っている場合、そのコンポーネントのワーカースレッドの最大数が CPU 数によって制限されます。

RHOSP director がコントローラー上で設定するコンポーネントのワーカースレッドの数は、CPU 数によって制限されます。

デプロイメントで Ceph Storage ノードを使用する場合、ノード数が 700 を超える大規模な環境では以下の仕様が推奨されます。

表2.2 Ceph Storage ノードを使用する場合に推奨されるコントローラーノードの仕様

システム要件	説明
--------	----

システム要件	説明
ノード数	<p>Controller ロールに含まれるコントローラーサービスを持つ 3 台のコントローラーノード</p> <p>オプションとして、専用ノードでコントローラーサービスをスケールするには、コンポーザブルサービスを使用します。詳細は、Red Hat OpenStack Platform デプロイメントのカスタマイズガイド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。</p>
CPU	2 ソケット (それぞれ 32 コア、64 スレッド)
ディスク	<p>500 GB のルートディスク (1x SSD または 2x 7200 RPM のハードドライブ (RAID 1))</p> <p>Swift 専用の 500GB ディスク (1x SSD または 1x NVMe)</p> <p>オプション: イメージキャッシュ用の 500 GB ディスク (7200RPM の 1x SSD または 2x ハードドライブ、RAID 1)</p>
メモリー	384GB
ネットワーク	<p>25 Gbps のネットワークインターフェイスまたは 10 Gbps のネットワークインターフェイス。10 Gbps ネットワークインターフェイスを使用する場合は、ネットワークボンディングを使用して 2 つのボンディングを作成します。</p> <ul style="list-style-type: none"> ● プロビジョニング (bond0、mode4)、内部 API (bond0、mode4)、プロジェクト (bond0、mode4) ● ストレージ (bond1、mode4)、ストレージ管理 (bond1、mode4)

デプロイメントで Ceph Storage ノードを使用しない場合、ノード数が 700 を超える大規模な環境では以下の仕様が推奨されます。

表2.3 Ceph Storage ノードを使用しない場合に推奨されるコントローラーノードの仕様

システム要件	説明
--------	----

システム要件	説明
ノード数	<p>Controller ロールに含まれるコントローラーサービスを持つ 3 台のコントローラーノード</p> <p>オプションとして、専用ノードでコントローラーサービスをスケールするには、コンポーザブルサービスを使用します。詳細は、Red Hat OpenStack Platform デプロイメントのカスタマイズガイドの 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。</p>
CPU	2 ソケット (それぞれ 32 コア、64 スレッド)
ディスク	<p>500 GB ルートディスク (1x SSD)</p> <p>Swift 専用の 500GB ディスク (1x SSD または 1x NVMe)</p> <p>オプション: イメージキャッシュ用の 500 GB ディスク (7200RPM の 1x SSD または 2x ハードドライブ、RAID 1)</p>
メモリー	384GB
ネットワーク	<p>25 Gbps のネットワークインターフェイスまたは 10 Gbps のネットワークインターフェイス。10 Gbps ネットワークインターフェイスを使用する場合は、ネットワークボンディングを使用して 2 つのボンディングを作成します。</p> <ul style="list-style-type: none"> ● プロビジョニング (bond0、mode4)、内部 API (bond0、mode4)、プロジェクト (bond0、mode4) ● ストレージ (bond1、mode4)、ストレージ管理 (bond1、mode4)

2.3. オーバークラウドコンピュートノードのシステム要件

オーバークラウドのデプロイメントを計画する際には、コンピュートノードに推奨されるシステム要件を確認します。

表2.4 コンピュートノードに推奨される仕様

システム要件	説明
ノード数	Red Hat は、さまざまなコンポーザブルコンピュートロールで 750 ノードのスケールについてテストを実施しています。

システム要件	説明
CPU	2 ソケット (それぞれ 12 コア、24 スレッド)
ディスク	500 GB のルートディスク
メモリー	128 GB (NUMA ノードあたり 64 GB)。デフォルトでは 2 GB がホスト用に確保されます。 分散仮想ルーターでは、確保されるメモリーを 5 GB に増やします。
ネットワーク	25 Gbps のネットワークインターフェイスまたは 10 Gbps のネットワークインターフェイス。10 Gbps ネットワークインターフェイスを使用する場合は、ネットワークボンディングを使用して 2 つのボンディングを作成します。 <ul style="list-style-type: none"> ● プロビジョニング (bond0、mode4)、内部 API (bond0、mode4)、プロジェクト (bond0、mode4) ● ストレージ (bond1、mode4)

2.4. RED HAT CEPH STORAGE ノードのシステム要件

Ceph Storage ノードのシステム要件については、次のリソースを参照してください。

- Ceph ノードのハードウェア要件に関する詳細は、Red Hat Ceph Storage 4 **ハードウェアガイド** の [ハードウェアを選択する一般的な原則](#) を参照してください。
- Ceph ノードのデプロイメント設定の詳細は、[director を使用した Red Hat Ceph Storage および Red Hat OpenStack Platform のデプロイ](#) を参照してください。
- ストレージのレプリケーション数の変更に関する詳細は、**Red Hat Ceph Storage 設定ガイド** の [プール、配置グループ、および CRUSH 設定オプション](#) を参照してください。

第3章 RED HAT OPENSTACK デプロイメントのベストプラクティス

OpenStack のデプロイを計画して準備する場合は、以下のベストプラクティスを確認してください。お使いの環境で、これらのプラクティスの1つまたは複数を実用できます。

3.1. RED HAT OPENSTACK デプロイメントの準備

Red Hat OpenStack Platform (RHOSP) をデプロイする前に、以下のデプロイメント準備タスクのリストを確認してください。お使いの環境で、デプロイメント準備タスクの1つまたは複数を実用できます。

イントロスペクションのサブネット範囲を設定して、一度にイントロスペクションを実施する最大オーバークラウドノードに対応する

`director` を使用して RHOSP をデプロイおよび設定する場合は、コントロールプレーンネットワークに CIDR 表記を使用して、現在または今後追加するすべてのオーバークラウドノードに対応します。

優先するネットワークに対してジャンボフレームを有効にする

使用頻度の高いネットワークでジャンボフレームやより高い MTU を使用すると、ネットワークでより大きなデータグラムや TCP ペイロードを送信し、CPU オーバーヘッドを削減して帯域幅を増やすことができます。ジャンボフレームは、より高い MTU をサポートするネットワークスイッチを備えたネットワークに対してのみ有効にしてください。MTU を高くするとパフォーマンスが向上することが判明している標準ネットワークは、Tenant ネットワーク、Storage ネットワーク、および Storage Management ネットワークです。詳細は、[director を使用した Red Hat OpenStack Platform のインストールと管理](#)の [ジャンボフレームの設定](#) を参照してください。

各ノードのルートディスクヒントとして World Wide Name (WWN) を設定し、デプロイメントおよび起動時にノードが誤ったディスクを使用しないようにする

ノードに複数のディスクが含まれる場合は、イントロスペクションデータを使用し、各ノードのルートディスクヒントとして WWN を設定します。これにより、デプロイメントおよび起動時にノードが誤ったディスクを使用しないようになります。詳細は、[director を使用した Red Hat OpenStack Platform のインストールと管理](#)ガイドの [マルチディスク Ceph クラスターのルートディスクの定義](#) を参照してください。

複数のディスクを持つノードで Bare Metal サービス (ironic) の自動クリーニングを有効にする

Bare Metal サービスの自動クリーニングを使用して、複数のディスクを持ち、複数のブートローダーがある可能性のあるノードでメタデータを消去します。ディスクに複数のブートローダーが存在するため、ノードはブートディスクとの一貫性を失う可能性があり、その結果、誤った URL を使用するメタデータをプルしようとするノードのデプロイメントに失敗します。

Bare Metal サービスの自動クリーニングを有効にするには、アンダークラウドノードで `undercloud.conf` ファイルを編集し、以下の行を追加します。

```
clean_nodes = true
```

Bare Metal (ironic) イントロスペクションのノード数を制限する

すべてのノードで同時にイントロスペクションを実行する場合には、ネットワークの制約により失敗する可能性があります。一度に最大 50 のノードでイントロスペクションを実施します。

`undercloud.conf` ファイルの `dhcp_start` および `dhcp_end` の範囲が、環境にあるノードの数に対して十分な大きさになるようにしてください。

利用可能な IP が十分でない場合は、範囲のサイズを超えて発行しないでください。これにより、同時に実行するイントロスペクション操作の数が制限されます。イントロスペクションの DHCP リースが期限切れになるのを許可するには、イントロスペクションが完了してから数分間は IP アドレスをさらに発行しないでください。

3.2. RED HAT OPENSTACK のデプロイメント設定

Red Hat OpenStack Platform (RHOSP) デプロイメント設定の推奨事項に関する以下のリストを確認してください。

小規模なデプロイメントにより heat テンプレートを検証する

3つ以上のコントローラーノード、1つのコンピューターノード、および3つの Ceph Storage ノードで構成される、小規模な環境をデプロイします。この設定を使用して、すべての heat テンプレートが正しいことを確認できます。

コンピューター全体のインスタンスの分散を改善する

多数のインスタンスの作成中、Compute スケジューラーは、コンピューターノードの以前のインスタンスのリソース割り当てが確定するまで、そのコンピューターノードのリソースを認識しません。コンピューターノードの不均一な生成を避けるために、次のいずれかの操作を実行できます。

- **NovaSchedulerShuffleBestSameWeighedHosts** パラメーターの値を **true** に設定します。

```
parameter_defaults:
  NovaSchedulerShuffleBestSameWeighedHosts: `True`
```

- インスタンスによってコンピューターノードが過負荷になるのを防ぐには、**max_instances_per_host** をコンピューターノードが生成できるインスタンスの最大数に設定し、**NumInstancesFilter** パラメーターが有効になっていることを確認します。コンピューターノードがこのインスタンス数に達すると、スケジューラーがそのノードをそれ以降のインスタンス生成スケジュールの対象として選択しなくなります。



注記

NumInstancesFilter パラメーターはデフォルトで有効になっています。ただし、環境ファイルの **NovaSchedulerEnabledFilters** パラメーターを変更する場合は、必ず **NumInstancesFilter** パラメーターを有効にしてください。

```
parameter_defaults:
  ControllerExtraConfig
    nova::scheduler::filter::max_instances_per_host: <maximum_number_of_instances>
  NovaSchedulerEnabledFilters:
    - AvailabilityZoneFilter
    - ComputeFilter
    - ComputeCapabilitiesFilter
    - ImagePropertiesFilter
    - ServerGroupAntiAffinityFilter
    - ServerGroupAffinityFilter
    - NumInstancesFilter
```

- **<maximum_number_of_instances>** は、コンピューターノードが生成できるインスタンスの最大数に置き換えます。

Networking サービス (neutron) のスケール設定

パフォーマンスとスケールの安定性を向上させるために、表 3.1. の設定を、大規模な OpenStack 環境でテストおよび検証しました。

サーバー側のプローブ間隔は、**ovsdb-server** によってクライアント (**neutron**、**ovn-controller**、および **ovn-metadata-agent**) に送信されるプローブのタイムアウトを制御します。タイムアウトが経過するまでにクライアントから応答が得られない場合、クライアントとの接続が切断され、強制的に再接続されます。クライアントがタイムアウトする可能性が最も高い状況は、クライアントが

ovsdb-server に初めて接続し、データベースのコピーをメモリーにロードするときです。タイムアウトが短すぎると、データベースのダウンロード中に **ovsdb-server** がクライアントを切断します。その結果、クライアントが再接続して再試行することになり、このサイクルが永久に繰り返されます。したがって、最大タイムアウト間隔が機能しない場合は、プローブ間隔の値をゼロに設定してプローブを無効にしてください。

クライアント側のプローブ間隔が無効になっている場合、**ovsdb-server** への接続が TCP キープアライブメッセージを使用して監視されます。



注記

可能な場合は、常に tripleo heat template (THT) のパラメーターを使用して必要な設定を行ってください。THT または Puppet でデフォルト値が定義されている場合、手動で行った設定が、設定ダウンロードの実行によって上書きされるためです。さらに、手動で設定できるのは既存の環境のみであるため、設定の変更は新しいノードや置き換えられたノードには適用されません。

表3.1 Networking サービスに推奨されるスケール設定

設定	説明	手動による設定	THT のパラメーター
コンピュータノード上の OVS サーバー側の非アクティブ状態プローブ	このプローブ間隔を 5 秒から 30 秒に増やします。	<pre>ovs-vsctl set Manager . inactivity_probe=30000</pre>	
コントローラーノード上の OVN Northbound サーバー側の非アクティブ状態プローブ	このプローブ間隔を 180000 ミリ秒に増やすか、0 に設定して無効にします。	<pre>podman exec -u root ovn_controller ovn-nbctl --no-leader-only set Connection . inactivity_probe=180000</pre>	
コントローラーノード上の OVN Southbound サーバー側の非アクティブ状態プローブ	このプローブ間隔を 180000 ミリ秒に増やすか、0 に設定して無効にします。	<pre>podman exec -u root ovn_controller ovn-sbctl --no-leader-only set Connection . inactivity_probe=180000</pre>	

設定	説明	手動による設定	THT のパラメーター
コンピュータノード上の OVN コントローラーのリモートプローブ間隔	このプローブ間隔を 180000 ミリ秒に増やすか、0 に設定して無効にします。	<pre>podman exec -u root ovn_controller ovs-vsctl -- no-leader-only set Open_vSwitch . external_ids:ovn-remote- probe-interval=180000</pre>	OVNRemoteProbeInterval: 180000
コントローラーノード上の Networking サービスのクライアント側プローブ間隔	このプローブ間隔を 180000 ミリ秒に増やすか、0 に設定して無効にします。	<pre>crudini --set /var/lib/config- data/puppet- generated/neutron/etc/neutr on/plugins/ml2/ml2_conf.ini ovn ovsdb_probe_interval 180000</pre>	OVNOvsdbProbeInterval: 180000
コントローラーノード上の Networking サービスの api_workers	neutron-server の負荷に基づいて、個別の API ワーカープロセスのデフォルト数を 12 から 16 以上に増やします。	<pre>crudini --set /var/lib/config- data/puppet- generated/neutron/etc/neutr on/neutron.conf DEFAULT api_workers 16</pre>	NeutronWorkers: 16
コントローラーノード上の Networking サービスの agent_down_time	非常に大規模なクラスターの場合は、 agent_down_time を最大許容数に設定します。	<pre>crudini --set /var/lib/config- data/puppet- generated/neutron/etc/neutr on/neutron.conf DEFAULT agent_down_time 2147483</pre>	NeutronAgentDownTime: 2147483
コンピュータノード上の OVN メタデータの report_agent	大規模なインストールでは、 report_agent を無効にします。	<pre>crudini --set /var/lib/config- data/puppet- generated/neutron/etc/neutr on/neutron_ovn_metadata_a gent.ini agent report_agent false</pre>	
コンピュータノード上の OVN の metadata_workers	すべてのコンピュータノードの metadata_workers を最小限に減らし、OVN Southbound データベースへの接続を減らします。	<pre>crudini --set /var/lib/config- data/puppet- generated/neutron/etc/neutr on/neutron_ovn_metadata_a gent.ini DEFAULT metadata_workers 1</pre>	NeutronMetadataWorkers: 1

設定	説明	手動による設定	THT のパラメーター
コンピュータノード上の OVN メタデータの rpc_workers	すべてのコンピュータノードで rpc_workers を最小限に減らします。	<pre>crudini --set /var/lib/config-data/puppet-generated/neutron/etc/neutron/neutron_ovn_metadata_agent.ini DEFAULT rpc_workers 0</pre>	NeutronRpcWorkers: 0
コンピュータノード上の OVN メタデータのクライアント側プローブ間隔	このプローブ間隔を 180000 ミリ秒に増やすか、0 に設定して無効にします。	<pre>crudini --set /var/lib/config-data/puppet-generated/neutron/etc/neutron/neutron_ovn_metadata_agent.ini ovn ovnsdb_probe_interval 180000</pre>	OVNOvsdbProbeInterval: 180000

同時にプロビジョニングされるノードの数を制限する

平均的なエンタープライズレベルのラックユニット内に収まるサーバーの数は、通常 50 台です。そのため、同時にデプロイできるノードの数は、平均 1 ラック分です。

デプロイの問題を診断するのに必要なデバッグを最小限に抑えるには、一度にデプロイするノードを 50 個までにしてください。より大きな数のノードをデプロイする場合として、Red Hat は最大 100 ノードの同時テストに成功しています。

コンピュータノードをバッチでスケールするには、**--limit** オプションを指定して **openstack overcloud deploy** コマンドを使用します。これにより、時間が節約され、アンダークラウドでのリソース消費が削減されます。

未使用の NIC を無効にする

デプロイ中にオーバークラウドに未使用の NIC がある場合は、NIC 設定テンプレートで未使用のインターフェイスを定義して、インターフェイスを **use_dhcp: false** および **defroute: false** に設定する必要があります。

未使用のインターフェイスを定義しない場合は、イントロスペクションおよびスケールアップ操作中に、ルーティングの問題や IP 割り当ての問題が発生する可能性があります。デフォルトでは、NIC は **BOOTPROTO=dhcp** を設定します。つまり、未使用のオーバークラウド NIC は、PXE プロビジョニングに必要な IP アドレスを消費します。これにより、ノードで利用可能な IP アドレスのプールが減少する場合があります。

未使用の Bare Metal Provisioning (ironic) ノードの電源をオフにする

メンテナンスモードにある未使用の Bare Metal Provisioning (ironic) ノードの電源をオフにしてください。Bare Metal Provisioning は、メンテナンスモードのノードの電源状態を追跡せず、メンテナンスモードで電源オン状態のまま維持されている以前のデプロイメントのノードの電源状態を、誤ってオフとして報告します。これにより、未使用のノードのオペレーティングシステムに古い設定 (オーバークラウドネットワークの IP アドレスなど) がある場合、進行中のデプロイで問題が発生する可能性があります。デプロイが失敗した後に再デプロイする場合は、未使用のノードをすべて電源オフにしてください。

3.3. アンダークラウドのチューニング

Red Hat OpenStack Platform (RHOSP) のデプロイメントをスケーリングする予定があり、デフォルトのアンダークラウド設定を指定する場合は、このセクションを確認してください。

より大きなメッセージサイズをサポートするために HA Services を調整する

大規模なデプロイメントでは、mariadb および rabbitmq HA サービスに設定されているデフォルト値よりも大きなメッセージサイズが必要になります。アンダークラウドをデプロイする前に、カスタム環境ファイルと hieradata オーバーライドファイルを使用してこれらの値を増やします。

custom_env_files.yaml

```
parameter_defaults:
  max_message_size: 536870912
  MySQLServerOptions:
    mysqld:
      max_allowed_packet: "512M"
```

hieradata_override.yaml

```
rabbitmq_config_variables:
  max_message_size: 536870912
  cluster_partition_handling: 'ignore'
  queue_master_locator: '<<"min-masters">>'
```

undercloud.conf

```
custom_env_files = /home/stack/custom-undercloud-params.yaml
hieradata_override = /home/stack/hieradata.yaml
```

オープンファイルの制限を 4096 に増やす

/etc/security/limits.conf ファイル内の次のパラメーターを編集して、必ずアンダークラウドのオープンファイル制限を 4096 に増やしてください。

```
* soft nofile 4096
* hard nofile 4096
```

プロビジョニングプロセスと設定プロセスを分離する

- スタックおよび関連する RHOSP リソースのみを作成するには、**--stack-only** オプションを指定してデプロイメントコマンドを実行できます。
- Red Hat は、100 を超えるノードをデプロイする場合に、スタックと **config-download** の手順を分離することを推奨しています。

オーバークラウドに必要なすべての環境ファイルを追加します。

```
$ openstack overcloud deploy \
  --templates \
  -e <environment-file1.yaml> \
  -e <environment-file2.yaml> \
  ...
  --stack-only
```

- スタックのプロビジョニングが完了したら、**tripleo-admin** ユーザーのアンダークラウドからオーバークラウドへの SSH アクセスを有効にできます。**config-download** プロセスでは、**tripleo-admin** ユーザーを使用して Ansible ベースの設定を実施します。

```
$ openstack overcloud admin authorize
```

- オーバークラウドスタックの作成を無効にして、**config-download** ワークフローをソフトウェア設定にのみ適用するには、**--config-download-only** オプションを指定してデプロイコマンドを実行します。オーバークラウドに必要なすべての環境ファイルを追加します。

```
$ openstack overcloud deploy \
--templates \
-e <environment-file1.yaml> \
-e <environment-file2.yaml> \
...
--config-download-only
```

- **config-download** Playbook の実行を特定のノードまたはノードセットに制限するには、**--limit** オプションを使用します。
- スケールアップ操作の場合、新しいノードにのみソフトウェア設定を適用するには、**--limit** オプションを **--config-download-only** オプションを併せて使用します。

```
$ openstack overcloud deploy \
--templates \
-e <environment-file1.yaml> \
-e <environment-file2.yaml> \
...
--config-download-only --config-download-timeout --limit <Undercloud>,<Controller>,
<Compute-1>,<Compute-2>
```

--limit オプションを使用する場合は、必ず、リストに **<Controller>** と **<Undercloud>** を含めるようにしてください。**external_deploy_steps** インターフェイスを使用するタスク (すべての Ceph 設定など) は、**<Undercloud>** がオプションリストに含まれている場合に実行されます。すべての **external_deploy_steps** タスクはアンダークラウドで実行されます。

たとえば、スケールアップタスクを実行して Ceph への接続を必要とするコンピューターノードを追加する場合、リストに **<Undercloud>** が含まれてないと、このタスクは失敗します。Ceph 設定ファイルと **cephx** キーファイルが指定されていないためです。

--skip-tags external_deploy_steps オプションを使用しなければ、タスクは失敗します。