



# Red Hat OpenStack Platform 16.2

## Red Hat OpenStack Platform の最新状態の維持

Red Hat OpenStack Platform のマイナー更新の実施



# Red Hat OpenStack Platform 16.2 Red Hat OpenStack Platform の最新状態の維持

---

Red Hat OpenStack Platform のマイナー更新の実施

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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

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

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

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

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

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

All other trademarks are the property of their respective owners.

## 概要

Red Hat Open Stack Platform (RHOSP) 環境のマイナー更新を実行して、最新のパッケージとコンテナで最新の状態に保つことができます。

## 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 マイナー更新の準備 .....	5
更新を妨げる可能性のある既知の問題 .....	5
手順 .....	7
1.1. ロングライフリリースのアップグレードパス .....	7
1.2. 環境の RED HAT ENTERPRISE LINUX リリースへのロック .....	9
1.3. TUS リポジトリへの切り替え .....	11
1.4. RED HAT OPENSTACK PLATFORM および ANSIBLE リポジトリの更新 .....	13
1.5. CONTAINER-TOOLS モジュールバージョンの設定 .....	15
1.6. コンテナイメージ準備ファイルの更新 .....	15
1.7. SSL/TLS 設定の更新 .....	16
1.8. オーバークラウドでのフェンシングの無効化 .....	17
第2章 アンダークラウドの更新 .....	18
前提条件 .....	18
2.1. コンテナ化されたアンダークラウドのマイナー更新を実施する .....	18
2.2. オーバークラウドイメージの更新 .....	18
第3章 オーバークラウドの更新 .....	20
前提条件 .....	20
手順 .....	20
3.1. オーバークラウドの更新準備タスクの実施 .....	20
3.2. コンテナイメージ準備タスクの実行 .....	21
3.3. オプション: すべてのオーバークラウドサーバーでの OVN-CONTROLLER コンテナの更新 .....	22
3.4. すべてのコントローラーノードの更新 .....	23
3.5. すべてのコンピュートノードの更新 .....	23
3.6. 全 HCI コンピュートノードの更新 .....	24
3.7. すべての DISTRIBUTEDCOMPUTEHCI ノードの更新 .....	25
3.8. すべての CEPH STORAGE ノードの更新 .....	26
3.9. データベースのオンライン更新の実施 .....	27
3.10. 更新の最終処理 .....	27
第4章 オーバークラウドの再起動 .....	29
4.1. コントローラーノードおよびコンポーザブルノードの再起動 .....	29
4.2. CEPH STORAGE (OSD) クラスターの再起動 .....	30
4.3. コンピュートノードの再起動 .....	31



## 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 マイナー更新の準備

お使いの Red Hat OpenStack Platform (RHOSP) 16.2 環境を、最新のパッケージおよびコンテナで更新された状態に維持してください。

更新に関するバージョンは以下のとおりです。

更新前の RHOSP バージョン	新規 RHOSP バージョン
Red Hat OpenStack Platform 16.1.z	Red Hat OpenStack Platform 16.2 (最新)
Red Hat OpenStack Platform 16.2.z	Red Hat OpenStack Platform 16.2 (最新)

### RHOSP マイナー更新プロセスワークフロー

RHOSP 環境を更新するには、次の手順を完了する必要があります。

1. RHOSP マイナー更新向けの環境を準備します。
2. アンダークラウドを最新の OpenStack 16.2.z バージョンに更新します。
3. オーバークラウドを最新の Open Stack 16.2.z バージョンに更新します。
4. すべての Red Hat Ceph Storage サービスをアップグレードします。
5. コンバージェンスのコマンドを実行して、オーバークラウドスタックをリフレッシュします。

インフラストラクチャーがマルチスタックの場合は、各オーバークラウドスタックを一度に1つずつ完全に更新します。分散コンピューティングノード (DCN) インフラストラクチャーがある場合は、中央のロケーションでオーバークラウドを完全に更新してから、各エッジサイトでオーバークラウドを一度に1つずつ更新します。

### RHOSP 環境を更新する前の考慮事項

更新プロセス中のガイドとして、次の情報を考慮してください。

- Red Hat は、アンダークラウドおよびオーバークラウドの制御プレーンをバックアップすることを推奨しています。ノードのバックアップについて詳しくは、[アンダークラウドおよびコントロールプレーンノードのバックアップと復元](#)を参照してください。
- 更新を妨げる可能性のある既知の問題を把握してください。
- 更新する前に、可能な更新パスとアップグレードパスを把握してください。詳細は、「[ロングライフリリースのアップグレードパス](#)」を参照してください。
- 現在のメンテナンスリリースを確認するには、**\$ cat /etc/rhosp-release** を実行してください。環境を更新した後にこのコマンドを実行して、更新を検証することもできます。

### 更新を妨げる可能性のある既知の問題

マイナーバージョンの更新の正常な完了に影響を及ぼす可能性のある、以下の既知の問題を確認してください。

[BZ#1973660 - 16.1 から 16.2 への更新時に rabbitmq サービスの設定を試みる操作が中断する。](#)

ノードのクラスターをシャットダウンする際に生じる競合状態により、Pacemaker バージョン **2.0.3-5.el8\_2.4** を実行するオーバークラウドノードで更新に失敗する場合があります。

現在オーバークラウドノードのいずれかに Pacemaker バージョン **2.0.3-5.el8\_2.4** がインストールされている場合、オーバークラウドノードを更新する前に Pacemaker をアップグレードする必要があります。詳細は、以下の Red Hat ナレッジベースのソリューション [Update from OSP16.1 to OSP16.2 might fail to update certain HA containers](#) を参照してください。

**BZ#1975240 - 16.1 から 16.2 への更新時に tsx フラグを有効にすると、更新中にコンピュートノードが再起動し ping の喪失が生じる**

Red Hat Enterprise Linux (RHEL) バージョン 8.3 以降、Intel Transactional Synchronization Extensions (TSX) 機能のサポートはデフォルトで無効になっています。これにより、以下の移行シナリオにおけるホスト間でのインスタンスのライブマイグレーションで問題が発生します。

- TSX カーネル引数が有効なホストから TSX カーネル引数が無効なホストへの移行。

TSX 機能をサポートする Intel ホストでは、ライブマイグレーションに失敗する場合があります。この問題の影響を受ける CPU の詳細は、[Affected Configurations](#) を参照してください。

詳細は、Red Hat ナレッジベースのソリューション [Guidance on Intel TSX impact on OpenStack guests](#) を参照してください。

**BZ#1872404: クォーラムを維持してノードを並行して再起動すると、予期せぬノードのシャットダウンが生じる**

RHEL 8.4 を実行し、コンポーザブルロールをベースとするノードについては、他のロールを更新する前に **Database** ロールを更新する必要があります。

**BZ#2027787 - Undercloud upgrade to 16.2 fails because of missing dependencies of swtpm**

**advanced-virt-for-rhel-8-x86\_64-eus-rpms** および **advanced-virt-for-rhel-8-x86\_64-rpms** リポジトリには、アップグレードが正常に行われないという既知の問題があります。アップグレード前にこれらのリポジトリを無効にするには、Red Hat ナレッジベースのソリューション [advanced-virt-for-rhel-8-x86\\_64-rpms are no longer required in OSP 16.2](#) を参照してください。

**BZ#2009106 - podman panic after tripleo\_nova\_libvirt restart two times**

RHOSP 16.1 から 16.2 へのアップグレード、および RHOSP 16.2.1 から 16.2.2 へのアップグレードには、Podman および libvirt サービスの変更に関連する既知の問題があります。アップグレードする前にワークロードを移行しないと、アップグレードが失敗する可能性があります。

**BZ#2129445 - 16.2.0 の既知の問題 nova\_libvirt タグ: 16.2.0-55.1638436404 libvirt バージョンの非互換性により、RHOSP 16.2.0 から RHOSP 16.2.x へのマイナー更新後にインスタンスが管理不能な状態のままになる**

libvirt バージョンの非互換性による重大な影響のリスクを評価するまで、RHOSP 16.2.0 から 16.2.2 または 16.2.3 に更新しないでください。この評価を完了するには、すべてのコンピュートノードの **nova\_libvirt** コンテナの libvirt パッケージを確認します。

```
$ sudo podman exec nova_libvirt rpm -q libvirt
```

libvirt のバージョンが 7.0 の場合、デプロイメントはバグの影響を受けません。更新を実行できます。

libvirt のバージョンが 7.6 の場合、デプロイメントはバグの影響を受けません。更新はリスクがあります。以下のいずれかのオプションを選択します。

- 注意: リリース時に RHOSP 16.2.4 に直接更新します。このリリースには、非互換性の問題に対する修正が含まれています。更新を延期できる場合は、このオプションを使用することが推奨されます。
- ホットフィックス: お使いの環境がこの問題を解決するホットフィックスパッチと互換性があるかどうかを確認するには、Red Hat サポート担当者にお問い合わせください。ビジネス面で即時更新が必要な場合には、このオプションを使用します。

バージョンの互換性がない状態で更新を既に実行している場合は、問題を解決するためのガイダンスについて、[Instances are not manageable after upgrading deployment from RHOSP16.2.0 to RHOSP16.2.z](#) を参照してください。

### BZ#2181566 - 16.2.3 のオーバークラウドサーバーが短い fqdn ホスト名を表示

Red Hat OpenStack Platform (RHOSP) 16.2 では、**nova::dhcp\_domain** パラメーターが導入されました。RHOSP 16.1 から 16.2 リリースに更新し、カスタムテンプレートにレガシーの **nova::metadata::dhcp\_domain** パラメーターが含まれる場合は、**nova::dhcp\_domain** パラメーターで競合が発生します。その結果、コンピュートノードではホスト名が生成されません。この問題を回避するには、次のいずれかのオプションを選択します。

- 従来の **nova::metadata::dhcp\_domain** パラメーターおよび **nova::dhcp\_domain** パラメーターを同じ値に設定します。
- 更新を待ちます。RHOSP 16.2.6 で修正が予定されています。

## 手順

マイナー更新用に RHOSP 環境を準備するには、以下の手順を実行します。

1. [「環境の Red Hat Enterprise Linux リリースへのロック」](#)
2. [「TUS リポジトリへの切り替え」](#)
3. [「Red Hat Openstack Platform および Ansible リポジトリの更新」](#)
4. [「container-tools モジュールバージョンの設定」](#)
5. [「コンテナイメージ準備ファイルの更新」](#)
6. [「SSL/TLS 設定の更新」](#)
7. [「オーバークラウドでのフェンシングの無効化」](#)

## 1.1. ロングライフリリースのアップグレードパス

更新またはアップグレードを開始する前に、可能な更新およびアップグレードパスをよく理解してください。



### 注記

RHOSP および RHEL の現行バージョンは、**/etc/rhosp-release** および **/etc/redhat-release** ファイルで確認できます。

表1.1 バージョンの更新パス

現行バージョン	更新後のバージョン
RHEL 7.x 上の RHOSP 10.0.x	最新の RHEL 7.7 における最新の RHOSP 10.0
RHEL 7.x 上の RHOSP 13.0.x	最新の RHEL 7.9 における最新の RHOSP 13.0
RHEL 8.2 上の RHOSP 16.1.x	最新の RHEL 8.2 における最新の RHOSP 16.1
RHEL 8.2 上の RHOSP 16.1.x	最新の RHEL 8.4 における最新の RHOSP 16.2
RHEL 8.4 上の RHOSP 16.2.x	最新の RHEL 8.4 における最新の RHOSP 16.2

表1.2 バージョンパスをアップグレードします

現行バージョン	更新後のバージョン
RHEL 7.7 上の RHOSP 10	最新の RHEL 7.9 における最新の RHOSP 13
RHEL 7.9 上の RHOSP 13	最新の RHEL 8.2 における最新の RHOSP 16.1
RHEL 7.9 上の RHOSP 13	RHEL 8.4 最新の RHOSP 16.2 最新

詳細については [Framework for Upgrades \(13 to 16.2\)](#) を参照してください。

Red Hat では、お使いの環境を次のロングライフリリースにアップグレードするためのオプションを 2 つ提供しています。

インプレースアップグレード

既存の環境でサービスのアップグレードを実施します。本ガイドでは、主にこのオプションを中心に説明します。

並列移行

新しい Red Hat OpenStack Platform 16.2 環境を作成し、ワークロードを現在の環境から新しい環境に移行します。Red Hat OpenStack Platform の並列移行についての詳しい情報は、Red Hat Global Professional Services にお問い合わせください。



重要

以下の表に示す時間は内部テストに基づく最短の推定値であり、すべての実稼働環境には適用されない可能性があります。たとえば、ハードウェアのスペックが低い場合やブート時間が長い場合は、これらの時間に余裕を持たせてください。各タスクのアップグレード時間を正確に測定するには、実稼働環境と類似したハードウェアを持つテスト環境でこれらの手順を実施してください。

表1.3 アップグレードパスの影響と時間

	インプレースアップグレード	並列移行
アンダークラウドのアップグレード時間	<p>それぞれの主要な操作の推定時間は以下のとおりです。</p> <ul style="list-style-type: none"> <li>● Leapp アップグレードコマンド: 30 分間</li> <li>● Leapp リブート: 30 分間</li> <li>● director のアップグレード: 40 分間</li> </ul>	なし。既存のアンダークラウドに加えて、新しいアンダークラウドを作成します。
オーバークラウドコントロールプレーンのアップグレード時間	<p>コントローラーノードごとの推定時間は以下のとおりです。</p> <ul style="list-style-type: none"> <li>● Leapp アップグレードおよびリブート: 60 分間</li> <li>● サービスのアップグレード: 60 分間</li> </ul>	なし。既存のコントロールプレーンに加えて、新しいコントロールプレーンを作成します。
コントロールプレーンの機能停止時間	ブートストラップコントローラーノードのサービスアップグレード時間: 約 60 分間	なし。ワークロードの移行中、両方のオーバークラウドは稼動状態にあります。
コントロールプレーンの機能停止による影響	機能停止時間中 OpenStack の操作を行うことはできません。	機能停止時間はありません。
オーバークラウドデータプレーンのアップグレード時間	<p>コンピュートノードおよび Ceph Storage ノードごとの推定時間は以下のとおりです。</p> <ul style="list-style-type: none"> <li>● Leapp アップグレードおよびリブート: 60 分間</li> <li>● サービスのアップグレード: 30 分間</li> </ul>	なし。既存のデータプレーンに加えて、新しいデータプレーンを作成します。
データプレーンの機能停止時間	ノード間のワークロードの移行により、機能停止時間は最小限に抑えられます。	オーバークラウド間のワークロードの移行により、機能停止時間は最小限に抑えられます。
追加ハードウェアに関する要件	追加のハードウェアは必要ありません。	新しいアンダークラウドおよびオーバークラウドを作成するために、追加のハードウェアが必要です。

## 1.2. 環境の RED HAT ENTERPRISE LINUX リリースへのロック

Red Hat OpenStack Platform (RHOSP) 16.2 は Red Hat Enterprise Linux 8.4 (RHEL) でサポートされています。更新を実行する前に、アンダークラウドおよびオーバークラウドのリポジトリを RHEL 8.4 リリースにロックして、オペレーティングシステムが新しいマイナーリリースにアップグレードされないようにします。

## 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. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

3. **RhsmVars** パラメーターが含まれるオーバークラウドのサブスクリプション管理用環境ファイルを編集します。通常、このファイルのデフォルト名は **rhsm.yml** です。
4. サブスクリプション管理の設定で、**rhsm\_release** パラメーターが含まれているかどうかを確認してください。**rhsm\_release** パラメーターが存在しない場合は、追加して 8.4 に設定します。

```
parameter_defaults:
  RhsmVars:
    ...
    rhsm_username: "myusername"
    rhsm_password: "p@55w0rd!"
    rhsm_org_id: "1234567"
    rhsm_pool_ids: "1a85f9223e3d5e43013e3d6e8ff506fd"
    rhsm_method: "portal"
    rhsm_release: "8.4"
```

5. オーバークラウドのサブスクリプション管理用環境ファイルを保存します。
6. オーバークラウドの静的なインベントリーファイルを作成します。

```
$ tripleo-ansible-inventory --ansible_ssh_user heat-admin --static-yaml-inventory
~/inventory.yaml
```

デフォルトのオーバークラウド名 **overcloud** 以外のオーバークラウド名を使用する場合は、**--plan** オプションを使用して実際のオーバークラウドの名前を設定します。

7. すべてのノードでオペレーティングシステムのバージョンを RHEL8.4 にロックするタスクを含めて、Playbook を作成します。

```
$ cat > ~/set_release.yaml <<'EOF'
- hosts: all
  gather_facts: false
  tasks:
    - name: set release to 8.4
      command: subscription-manager release --set=8.4
      become: true
EOF
```

8. **set\_release.yaml** Playbook を実行します。

```
$ ansible-playbook -i ~/inventory.yaml -f 25 ~/set_release.yaml --limit <undercloud>,<Controller>,<Compute>
```

- **--limit** オプションを使用して、コンテンツをすべての RHOSP ノードに適用します。**<undercloud>**、**<Controller>**、**<Compute>** を、それらのノードを含む環境内の Ansible グループに置き換えます。
- これらのノードに別のサブスクリプションを使用している場合は、Ceph Storage ノードに対してこの Playbook を実行することはできません。



### 注記

手動でノードを特定のバージョンにロックするには、ノードにログインして **subscription-manager release** コマンドを実行します。

```
$ sudo subscription-manager release --set=8.4
```

## 1.3. TUS リポジトリへの切り替え

Red Hat OpenStack Platform (RHOSP) サブスクリプションには、標準のリポジトリに加えて、Red Hat Enterprise Linux (RHEL) 8.4 Extended Update Support (EUS) のリポジトリが含まれています。2022 年 4 月 30 日以降、メンテナンส์サポートを受けるには、RHEL 8.2 Telecommunications Update Service (TUS) リポジトリを有効化する必要があります。EUS リポジトリには、RHEL 9.0 の最新のセキュリティパッチとバグ修正が含まれています。

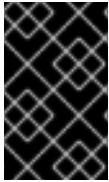
更新を実行する前に、リポジトリを必要な TUS リポジトリに切り替えます。

表1.4 EUS リポジトリから TUS リポジトリへの変更

EUS リポジトリ	TUS リポジトリ
rhel-8-for-x86_64-baseos-eus-rpms	rhel-8-for-x86_64-baseos-tus-rpms
rhel-8-for-x86_64-appstream-eus-rpms	rhel-8-for-x86_64-appstream-tus-rpms
rhel-8-for-x86_64-highavailability-eus-rpms	rhel-8-for-x86_64-highavailability-tus-rpms

表1.5 標準リポジトリから TUS リポジトリへの切り替え

標準のリポジトリ	TUS リポジトリ
rhel-8-for-x86_64-baseos-rpms	rhel-8-for-x86_64-baseos-tus-rpms
rhel-8-for-x86_64-appstream-rpms	rhel-8-for-x86_64-appstream-tus-rpms
rhel-8-for-x86_64-highavailability-rpms	rhel-8-for-x86_64-highavailability-tus-rpms



## 重要

特定バージョンの Podman との互換性を維持するには、TUS リポジトリを使用する必要があります。より新しいバージョンの Podman は、Red Hat Open Stack Platform 16.2 でテストされておらず、予期せぬ結果を招く可能性があります。

## 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。

```
$ source ~/stackrc
```

3. **RhsmVars** パラメーターが含まれるオーバークラウドのサブスクリプション管理用環境ファイルを編集します。通常、このファイルのデフォルト名は **rhsm.yml** です。
4. サブスクリプション管理の設定で **rhsm\_repos** パラメーターを確認します。このパラメーターに TUS リポジトリが含まれていない場合、該当するリポジトリを TUS バージョンに変更します。

```
parameter_defaults:
  RhsmVars:
    rhsm_repos:
      - rhel-8-for-x86_64-baseos-tus-rpms
      - rhel-8-for-x86_64-appstream-tus-rpms
      - rhel-8-for-x86_64-highavailability-tus-rpms
      - ansible-2.9-for-rhel-8-x86_64-rpms
      - openstack-16.2-for-rhel-8-x86_64-rpms
      - rhceph-4-tools-for-rhel-8-x86_64-rpms
      - fast-datapath-for-rhel-8-x86_64-rpms
```

5. オーバークラウドのサブスクリプション管理用環境ファイルを保存します。
6. オーバークラウドの静的なインベントリーファイルを作成します。

```
$ tripleo-ansible-inventory --ansible_ssh_user heat-admin --static-yaml-inventory
~/inventory.yaml
```

デフォルトのオーバークラウド名 **overcloud** 以外のオーバークラウド名を使用する場合は、**--plan** オプションを使用して実際のオーバークラウドの名前を設定します。

7. すべてのノードで、リポジトリを RHEL 8.4 EUS に設定するタスクが含まれる Playbook を作成します。

```
$ cat > ~/change_tus.yaml <<'EOF'
- hosts: all
  gather_facts: false
  tasks:
    - name: change to tus repos
      command: subscription-manager repos --disable=rhel-8-for-x86_64-baseos-eus-rpms --
disable=rhel-8-for-x86_64-appstream-eus-rpms --disable=rhel-8-for-x86_64-highavailability-
eus-rpms --enable=rhel-8-for-x86_64-baseos-tus-rpms --enable=rhel-8-for-x86_64-
```



```
appstream-tus-rpms --enable=rhel-8-for-x86_64-highavailability-tus-rpms
become: true
EOF
```

- お使いの環境に標準リポジトリが含まれている場合は、以下のリポジトリを無効にします。
  - rhel-8-for-x86\_64-baseos-rpms
  - rhel-8-for-x86\_64-appstream-rpms
  - rhel-8-for-x86\_64-highavailability-rpms

#### 8. **change\_tus.yaml** Playbook を実行します。

```
$ ansible-playbook -i ~/inventory.yaml -f 25 ~/change_tus.yaml --limit <undercloud>,<Controller>,<Compute>
```

- すべての Red Hat OpenStack Platform ノードにコンテンツを適用するには、**--limit** オプションを使用します。**<undercloud>**、**<Controller>**、**<Compute>** を、それらのノードを含む環境内の Ansible グループに置き換えます。
- これらのノードに別のサブスクリプションを使用している場合は、Ceph Storage ノードに対してこの Playbook を実行することはできません。

## 1.4. RED HAT OPENSTACK PLATFORM および ANSIBLE リポジトリの更新

Red Hat OpenStack Platform (RHOSP) 16.2 パッケージおよび Ansible 2.9 パッケージを使用するようにリポジトリを更新します。

### 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. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

3. **RhsmVars** パラメーターが含まれるオーバークラウドのサブスクリプション管理用環境ファイルを編集します。通常、このファイルのデフォルト名は **rhsm.yml** です。
4. サブスクリプション管理の設定で **rhsm\_repos** パラメーターを確認します。**rhsm\_repos** パラメーターで RHOSP 16.1 リポジトリおよび Ansible 2.8 リポジトリが使用されている場合は、リポジトリを正しいバージョンに変更します。

```
parameter_defaults:
  RhsmVars:
    rhsm_repos:
      - rhel-8-for-x86_64-baseos-tus-rpms
      - rhel-8-for-x86_64-appstream-tus-rpms
      - rhel-8-for-x86_64-highavailability-tus-rpms
      - ansible-2.9-for-rhel-8-x86_64-rpms
      - openstack-16.2-for-rhel-8-x86_64-rpms
      - fast-datapath-for-rhel-8-x86_64-rpms
```

5. オーバークラウドのサブスクリプション管理用環境ファイルを保存します。
6. オーバークラウドの静的なインベントリーファイルを作成します。

```
$ tripleo-ansible-inventory --ansible_ssh_user heat-admin --static-yaml-inventory
~/inventory.yaml
```

デフォルトのオーバークラウド名 **overcloud** 以外のオーバークラウド名を使用する場合は、**--plan** オプションを使用して実際のオーバークラウドの名前を設定します。

7. すべての RHOSP ノードで、リポジトリを RHOSP 16.2 に設定するタスクが含まれる Playbook を作成します。

```
$ cat > ~/update_rhosp_repos.yaml <<'EOF'
- hosts: all
  gather_facts: false
  tasks:
    - name: change osp repos
      command: subscription-manager repos --disable=openstack-16.1-for-rhel-8-x86_64-rpms
      --enable=openstack-16.2-for-rhel-8-x86_64-rpms --disable=ansible-2.8-for-rhel-8-x86_64-
      rpms --enable=ansible-2.9-for-rhel-8-x86_64-rpms
      become: true
EOF
```

8. **update\_rhosp\_repos.yaml** Playbook を実行します。

```
$ ansible-playbook -i ~/inventory.yaml -f 25 ~/update_rhosp_repos.yaml --limit <undercloud>,
<Controller>,<Compute>
```

- **--limit** オプションを使用して、コンテンツをすべての RHOSP ノードに適用します。**<undercloud>**、**<Controller>**、**<Compute>** を、それらのノードを含む環境内の Ansible グループに置き換えます。
  - これらのノードに別のサブスクリプションを使用している場合は、Ceph Storage ノードに対してこの Playbook を実行することはできません。
9. すべての Ceph Storage ノードで、リポジトリを RHOSP 16.2 に設定するタスクが含まれる Playbook を作成します。

```
$ cat > ~/update_ceph_repos.yaml <<'EOF'
- hosts: all
  gather_facts: false
  tasks:
    - name: change ceph repos
      command: subscription-manager repos --disable=openstack-16-deployment-tools-for-
      rhel-8-x86_64-rpms --enable=openstack-16.2-deployment-tools-for-rhel-8-x86_64-rpms --
      disable=ansible-2.8-for-rhel-8-x86_64-rpms --enable=ansible-2.9-for-rhel-8-x86_64-rpms
      become: true
EOF
```

10. **update\_ceph\_repos.yaml** Playbook を実行します。

```
$ ansible-playbook -i ~/inventory.yaml -f 25 ~/update_ceph_repos.yaml --limit CephStorage
```

**--limit** オプションを使用して、コンテンツを Ceph Storage ノードに適用します。

## 1.5. CONTAINER-TOOLS モジュールバージョンの設定

**container-tools** モジュールをバージョン **3.0** に設定して、すべてのノードで正しいパッケージバージョンを使用するようにします。

### 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. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

3. オーバークラウドの静的なインベントリーファイルを作成します。

```
$ tripleo-ansible-inventory --ansible_ssh_user heat-admin --static-yaml-inventory
~/inventory.yaml
```

デフォルトのオーバークラウド名 **overcloud** 以外のオーバークラウド名を使用する場合は、**--plan** オプションを使用して実際のオーバークラウドの名前を設定します。

4. すべてのノードで **container-tools** モジュールをバージョン **3.0** に設定するタスクが含まれる Playbook を作成します。

```
$ cat > ~/container-tools.yaml <<'EOF'
- hosts: all
  gather_facts: false
  tasks:
    - name: disable default dnf module for container-tools
      command: dnf module reset -y container-tools
      become: true
    - name: set dnf module for container-tools:3.0
      command: dnf module enable -y container-tools:3.0
      become: true
EOF
```

5. すべてのノードに対して **container-tools.yaml** Playbook を実行します。

```
$ ansible-playbook -i ~/inventory.yaml -f 25 ~/container-tools.yaml
```

## 1.6. コンテナイメージ準備ファイルの更新

コンテナ準備ファイルは、**ContainerImagePrepare** パラメーターが含まれるファイルです。このファイルを使用して、アンダークラウドおよびオーバークラウドのコンテナイメージを取得する際のルールを定義します。

環境を更新する前に、ファイルを確認して正しいイメージバージョンを取得するようにしてください。

### 手順

1. コンテナ準備ファイルを編集します。通常、このファイルのデフォルト名は **containers-prepare-parameter.yaml** です。
2. それぞれのルールセットについて、**tag** パラメーターが **16.2** に設定されていることを確認します。

```
parameter_defaults:
  ContainerImagePrepare:
    - push_destination: true
    set:
      ...
      tag: '16.2'
      tag_from_label: '{version}-{release}'
```



### 注記

更新に特定のタグ (**16.2** や **16.2.2** 等) を使用しない場合は、**tag** キーと値のペアを削除し、**tag\_from\_label** のみを指定します。これにより、更新プロセスの一部として使用するタグの値を決定する際に、インストールされた Red Hat OpenStack Platform バージョンが使用されます。

3. このファイルを保存します。

## 1.7. SSL/TLS 設定の更新

**resource\_registry** から **NodeTLSData** リソースを削除して、SSL/TLS 設定を更新します。

### 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. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

3. カスタムのオーバークラウド SSL/TLS パブリックエンドポイントファイルを編集します。通常、このファイルの名前は **~/templates/enable-tls.yaml** です。
4. **resource\_registry** から **NodeTLSData** リソースを削除します。

```
resource_registry:
  OS::TripleO::NodeTLSData: /usr/share/openstack-tripleo-heat-
    templates/puppet/extraconfig/tls/tls-cert-inject.yaml
  ...
```

オーバークラウドデプロイメントは、HAProxy の新しいサービスを使用して SSL/TLS が有効かどうかを判断します。



### 注記

これが **enable-tls.yaml** ファイルの **resource\_registry** セクションにある唯一のリソースである場合、**resource\_registry** セクションをすべて削除します。

5. SSL/TLS パブリックエンドポイントファイルを保存します。
6. Red Hat OpenStack Platform 16.1 から更新する場合、すべての更新前チェックに合格するには、Red Hat Identity Manager (IdM) でパーミッションを更新する必要があります。IdM を実行しているサーバーに **ssh** を使用してログインし、次のコマンドを実行します。

```
$ kinit admin
$ ipa privilege-add-permission 'Nova Host Management' --permission 'System: Modify Realm Domains'
```

## 1.8. オーバークラウドでのフェンシングの無効化

オーバークラウドを更新する前に、フェンシングが無効になっていることを確認します。

コントローラーノードの更新プロセス中にフェンシングが環境にデプロイされると、オーバークラウドは特定ノードが無効であることを検出し、フェンシング操作を試みる場合があります。これにより、意図しない結果が生じる可能性があります。

オーバークラウドでフェンシングを有効にしている場合には、意図しない結果を防ぐために、更新期間中フェンシングを一時的に無効に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. source コマンドで **stackrc** ファイルを読み込みます。

```
$ source ~/stackrc
```

3. コントローラーノードにログインし、Pacemaker コマンドを実行してフェンシングを無効にします。

```
$ ssh heat-admin@<controller_ip> "sudo pcs property set stonith-enabled=false"
```

**<controller\_ip>** を、コントローラーノードの IP アドレスに置き換えます。コントローラーノードの IP アドレスは、**openstack server list** コマンドで確認できます。

4. **fencing.yaml** 環境ファイルで、**EnableFencing** パラメーターを **false** に設定し、更新プロセス中にフェンシングが無効のままとなるようにします。

### 関連情報

- [STONITH を使用したコントローラーノードのフェンシング](#)

## 第2章 アンダークラウドの更新

director を使用して、アンダークラウドノード上の主要なパッケージを更新します。アンダークラウドとそのオーバークラウドイメージを最新の Red Hat Open Stack Platform (RHOSP) 16.2 バージョンに更新するには、以下の手順を実行します。

1. 「[コンテナ化されたアンダークラウドのマイナー更新を実施する](#)」
2. 「[オーバークラウドイメージの更新](#)」

### 前提条件

- アンダークラウドを最新の RHOSP 16.2 バージョンに更新する前に、すべての更新準備手順を完了していることを確認してください。詳細は、[1章 マイナー更新の準備](#) を参照してください。

### 2.1. コンテナ化されたアンダークラウドのマイナー更新を実施する

director では、アンダークラウドノード上の主要なパッケージを更新するためのコマンドが提供されています。Director を使用して、RHOSP 環境の現在のバージョン内でマイナー更新を実行します。

#### 手順

1. アンダークラウドノードで、**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。

```
$ source ~/stackrc
```

3. **dnfupdate** コマンドを使用して director メインパッケージを更新します。

```
$ sudo dnf update -y python3-tripleoclient* tripleo-ansible ansible
```

4. **openstack undercloudupgrade** コマンドを使用してアンダークラウド環境を更新します。

```
$ openstack undercloud upgrade
```

5. アンダークラウドの更新プロセスが完了するまで待ちます。
6. アンダークラウドをリブートして、オペレーティングシステムのカーネルとその他のシステムパッケージを更新します。

```
$ sudo reboot
```

7. ノードがブートするまで待ちます。

### 2.2. オーバークラウドイメージの更新

Director がノードをイントロスペクトして最新バージョンの RHOSP ソフトウェアでプロビジョニングできるようにするには、現在のオーバークラウドイメージを新しいバージョンに置き換える必要があります。事前にプロビジョニングされたノードを使用している場合は、この手順は必要ありません。

#### 前提条件

- アンダークラウドノードが最新バージョンに更新されている。詳細は、[「コンテナ化されたアンダークラウドのマイナー更新を実施する」](#) を参照してください。

## 手順

1. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

2. **stack** ユーザーのホーム下の **images** ディレクトリー (**/home/stack/images**) から既存のイメージを削除します。

```
$ rm -rf ~/images/*
```

3. アーカイブを展開します。

```
$ cd ~/images
$ for i in /usr/share/rhosp-director-images/overcloud-full-latest-16.2.tar /usr/share/rhosp-director-images/ironic-python-agent-latest-16.2.tar; do tar -xvf $i; done
$ cd ~
```

4. director に最新のイメージをインポートします。

```
$ openstack overcloud image upload --update-existing --image-path /home/stack/images/
```

5. ノードが新しいイメージを使用するように設定します。

```
$ openstack overcloud node configure $(openstack baremetal node list -c UUID -f value)
```

6. 新規イメージが存在することを確認します。

```
$ openstack image list
$ ls -l /var/lib/ironic/httpboot
```

## 重要

- オーバークラウドノードをデプロイする際には、オーバークラウドイメージのバージョンが該当する heat テンプレートバージョンに対応している状態にします。たとえば、RHOSP16.2 heat テンプレートでは RHOSP16.2 イメージのみを使用します。
- Red Hat カスタマーポータルまたは Red Hat Satellite Server を使用する接続環境をデプロイした場合、オーバークラウドのイメージとパッケージリポジトリのバージョンが同期していない可能性があります。オーバークラウドのイメージとパッケージリポジトリのバージョンが一致していることを確認するには、**virt-customize** ツールを使用できます。詳細は、Red Hat ナレッジベースで [Modifying the Red Hat Linux OpenStack Platform Overcloud Image with virt-customize](#) のソリューションを参照してください。
- 新しい **overcloud-full** イメージは、古い **overcloud-full** イメージを置き換えます。古いイメージに変更を加えた場合、特に今後新規ノードをデプロイする場合は、新しいイメージで変更を繰り返す必要があります。

## 第3章 オーバークラウドの更新

アンダークラウドを更新した後、オーバークラウドとコンテナイメージの準備コマンドを実行し、ノードを更新し、**オーバークラウドの更新収束コマンド**を実行してオーバークラウドを更新できます。コントロールプレーン API は、マイナー更新中もすべて利用できます。

### 前提条件

- アンダークラウドノードが最新バージョンに更新されている。詳細は、[2章 アンダークラウドの更新](#)を参照してください。
- stack** ユーザーのホームディレクトリーでコアテンプレートのローカルセットを使用している場合には、**オーバークラウドの高度なカスタマイズのカスタムコア Heat テンプレートの使用**に記載の推奨ワークフローを使用して、テンプレートを更新するようにしてください。オーバークラウドをアップグレードする前に、ローカルコピーを更新する必要があります。

### 手順

オーバークラウドを更新するには、次の手順を実行する必要があります。

1. [「オーバークラウドの更新準備タスクの実施」](#)
2. [「コンテナイメージ準備タスクの実行」](#)
3. [「オプション: すべてのオーバークラウドサーバーでの ovn-controller コンテナの更新」](#)
4. [「すべてのコントローラーノードの更新」](#)
5. [「すべてのコンピュートノードの更新」](#)
6. [「全 HCI コンピュートノードの更新」](#)
7. [「すべての Ceph Storage ノードの更新」](#)
8. [「データベースのオンライン更新の実施」](#)
9. [「更新の最終処理」](#)

### 3.1. オーバークラウドの更新準備タスクの実施

更新プロセスに向けてオーバークラウドを準備するには、**openstack overcloud update prepare** コマンドを実行する必要があります。このコマンドは、オーバークラウドプランを Red Hat Open Stack Platform (RHOSP)16.2 に更新して、更新用にノードを準備します。

#### 前提条件

- Ceph のサブスクリプションを使用し、Ceph ストレージノード用に **overcloud-minimal** イメージを使用するように director を設定している場合、**roles\_data.yaml** ロール定義ファイルで **rhsm\_enforce** パラメーターが **False** に設定されていることを確認する。
- カスタム NIC テンプレートをレンダリングした場合は、オーバークラウドのバージョンとの非互換性を回避するために、**openstack-tripleo-heat-templates** コレクションの更新バージョンでテンプレートを再生成する必要があります。カスタム NIC テンプレートの詳細は、**オーバークラウドの高度なカスタマイズ ガイド**の [カスタマイズのためのデフォルトのネットワークインターフェイステンプレートのレンダリング](#)を参照してください。





## 注記

OVN デプロイメントを使用する分散コンピュートノード (エッジ) アーキテクチャーの場合は、[すべてのオーバークラウドサーバーでの ovn-controller コンテナの更新](#) セクションに進む前に、Compute、DistributedCompute、または DistributedComputeHCI ノードを含むスタックごとにこの手順を完了する必要があります。

## 手順

1. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

2. 更新準備コマンドを実行します。

```
$ openstack overcloud update prepare \
  --templates \
  --stack <stack_name> \
  -r <roles_data_file> \
  -n <network_data_file> \
  -e <environment_file> \
  -e <environment_file> \
  ...
```

以下のオプションの中で、お使いの環境に適切なオプションを追加します。

- オーバークラウドスタックの名前がデフォルトの名前 **overcloud** とは異なる場合は、更新の準備コマンドに **--stack** オプションを追加し、**<stack\_name>** を実際のスタック名に置き換えます。
- 専用のカスタムロールを使用する場合は、カスタムロール (**<roles\_data>**) のファイルを追加します (**-r**)。
- カスタムネットワークを使用する場合は、コンポーザブルネットワーク (**<network\_data>**) のファイルを追加します (**-n**)
- 高可用性クラスターをデプロイする場合は、更新の準備コマンドに **--ntp-server** オプションを追加するか、または環境ファイルに **NtpServer** パラメーターおよび値を追加します。
- すべてのカスタム設定環境ファイル (**-e**)

3. 更新の準備プロセスが完了するまで待ちます。

## 3.2. コンテナイメージ準備タスクの実行

オーバークラウドを更新する前に、環境に必要なすべてのコンテナイメージ設定を準備して、最新の RHOSP16.2 コンテナイメージをアンダークラウドにプルする必要があります。

コンテナイメージの準備を完了するには、**container\_image\_prepare** タグの付いたタスクに対して **openstack overcloud external-updaterun** コマンドを実行する必要があります。



## 注記

デフォルトのスタック名 (**overcloud**) を使用していない場合は、**--stack <stack\_name>** オプションでスタック名を設定します。**<stack\_name>** は実際のスタック名に置き換えます。

## 手順

1. **stackrc** ファイルを取得します。

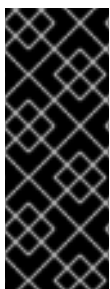
```
$ source ~/stackrc
```

2. **container\_image\_prepare** タグの付いたタスクに対して **openstack overcloud external-update run** コマンドを実行します。

```
$ openstack overcloud external-update run --stack <stack_name> --tags  
container_image_prepare
```

## 3.3. オプション: すべてのオーバークラウドサーバーでの OVN-CONTROLLER コンテナの更新

Modular Layer 2 Open Virtual Network メカニズムドライバー (ML2/OVN) を使用してオーバークラウドをデプロイした場合は、ovn-controller コンテナを最新の RHOSP16.2 バージョンに更新します。更新は、ovn-controller コンテナを実行するすべてのオーバークラウドサーバーで行われます。



## 重要

次の手順では、コントローラーのロールが割り当てられているサーバーの ovn-northd サービスを更新する前に、コンピューターのロールが割り当てられているサーバーの ovn-controller コンテナを更新します。

この手順を実行する前に誤って ovn-northd サービスを更新した場合、仮想マシンにアクセスしたり、新しい仮想マシンや仮想ネットワークを作成したりできない可能性があります。次の手順で接続を復元します。



## 注記

分散型コンピューターノード (エッジ) アーキテクチャの場合は、[すべてのコントローラーノードの更新](#) セクションに進む前に、Compute、DistributedCompute、または DistributedComputeHCI ノードを含むスタックごとにこの手順を完了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。

```
$ source ~/stackrc
```

3. ovn タグを持つタスクに対して **openstack overcloud external-update run** コマンドを実行します。

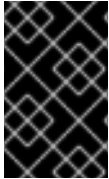
```
$ openstack overcloud external-update run --stack <stack_name> --tags ovn
```

- オーバークラウドスタックの名前がデフォルトのスタック名 **overcloud** と異なる場合は、スタック名を **--stack** オプションで設定し、**<stack\_name>** をスタックの名前に置き換えます。

4. ovn-controller コンテナの更新が完了するまで待ちます。

### 3.4. すべてのコントローラーノードの更新

すべてのコントローラーノードを最新の RHOSP16.2 バージョンに更新します。**--limit Controllerer** オプションを指定して **openstack overcloud update run** コマンドを実行し、操作をコントローラーノードだけに制限します。コントロールプレーン API は、マイナー更新中もすべて利用できます。



#### 重要

[BZ#1872404](#) が解決されるまで、コンポーザブルロールに基づくノードについては、先ず **Database** ロールを更新してから、**Controller**、**Messaging**、**Compute**、**Ceph**、およびその他のロールを更新する必要があります。



#### 注記

デフォルトのスタック名 (**overcloud**) を使用していない場合は、**--stack <stack\_name>** オプションでスタック名を設定します。**<stack\_name>** は実際のスタック名に置き換えます。

#### 手順

1. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

2. 更新コマンドを実行します。

```
$ openstack overcloud update run --stack <stack_name> --limit Controller
```

3. コントローラーノードの更新が完了するまで待ちます。

### 3.5. すべてのコンピュートノードの更新

すべてのコンピュートノードを最新の RHOSP16.2 バージョンに更新します。コンピュートノードを更新するには、**openstack overcloud update run** コマンドに **--limit Compute** オプションを指定して、操作をコンピュートノードのみに制限して実行する必要があります。

#### 並列処理に関する考慮事項

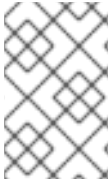
多数のコンピュートノードを更新する場合には、パフォーマンス向上のため、バックグラウンドで複数の更新タスクを実行し、ノードが 20 個含まれる別個のグループを更新するように各タスクを設定できます。たとえば、デプロイメントに 80 のコンピュートノードがある場合、次のコマンドを実行して、コンピュートノードを並行して更新できます。

```
$ openstack overcloud update run -y --limit 'Compute[0:19]' > update-compute-0-19.log 2>&1 &
$ openstack overcloud update run -y --limit 'Compute[20:39]' > update-compute-20-39.log 2>&1 &
$ openstack overcloud update run -y --limit 'Compute[40:59]' > update-compute-40-59.log 2>&1 &
$ openstack overcloud update run -y --limit 'Compute[60:79]' > update-compute-60-79.log 2>&1 &
```

ノード領域分割方法はランダムで、更新されるノードを制御することはできません。ノードの選択は、**tripleo-ansible-inventory** コマンドの実行時に生成するインベントリファイルに基づきます。

特定のコンピュートノードを更新するには、バッチで更新するノードをコンマ区切りリストで指定します。

```
$ openstack overcloud update run --limit <Compute0>,<Compute1>,<Compute2>,<Compute3>
```



#### 注記

デフォルトのスタック名 (**overcloud**) を使用していない場合は、**--stack <stack\_name>** オプションでスタック名を設定します。**<stack\_name>** は実際のスタック名に置き換えます。

#### 手順

1. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

2. 更新コマンドを実行します。

```
$ openstack overcloud update run --stack <stack_name> --limit Compute
```

3. コンピュートノードの更新が完了するまで待ちます。

## 3.6. 全 HCI コンピュートノードの更新

ハイパーコンバージドインフラストラクチャー (HCI) コンピューティングノードを最新の RHOSP16.2 バージョンに更新します。HCI Compute ノードを更新するには、**openstack overcloud update run** コマンドを実行し、**-limit Compute HCI** オプションを指定して、操作を HCI ノードのみに制限します。**openstack overcloud external-update run --tags ceph** コマンドを実行し、コンテナ化された Red Hat Ceph Storage 4 クラスターへの更新を実施する

#### 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 podman exec -it ceph-mon-controller-0 ceph -s
```

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** のステータスを返す。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 4 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. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

2. 更新コマンドを実行します。

```
$ openstack overcloud update run --stack <stack_name> --limit ComputeHCI
```

- **<stack\_name>** をスタックの名前に置き換えます。指定されていない場合、デフォルトは **overcloud** です。

3. ノードの更新が完了するまで待ちます。
4. Ceph Storage の更新コマンドを実行します。

```
$ openstack overcloud external-update run --stack <stack_name> --tags ceph
```

5. コンピュート HCI ノードの更新が完了するまで待ちます。

### 3.7. すべての DISTRIBUTEDCOMPUTEHCI ノードの更新

分散コンピュートノードのアーキテクチャーに固有のロールを更新します。分散コンピュートノードをアップグレードするときは、最初に **DistributedComputeHCI** ノードを更新してから、**DistributedComputeHCIScaleOut** ノードを更新します。



#### 注記

デフォルトのスタック名である overcloud を使用していない場合は、スタック名を **--stack <stack\_name>** オプションで設定し、**<stack\_name>** をスタックの名前に置き換えます。

#### 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 podman exec -it ceph-mon-controller-0 ceph -s
```

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** のステータスを返す。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 4 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. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

2. 更新コマンドを実行します。

```
$ openstack overcloud update run --stack <stack_name> --limit DistributedComputeHCI
```

3. **DistributedComputeHCI** ノードの更新が完了するまで待ちます。
4. Ceph Storage の更新コマンドを実行します。  

```
$ openstack overcloud external-update run --stack <stack_name> --tags ceph
```
5. **DistributedComputeHCI** ノードの更新が完了するまで待ちます。
6. 同じプロセスを使用して、**DistributedComputeHCIScaleOut** ノードを更新します。

### 3.8. すべての CEPH STORAGE ノードの更新

Red Hat Ceph Storage ノードを最新の RHOSP 16.2 バージョンに更新します。



#### 重要

RHOSP 16.2 は RHEL 8.4 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。詳細は、[Red Hat Ceph Storage: サポートされる設定](#) を参照してください。

#### 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 podman exec -it ceph-mon-controller-0 ceph -s
```

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** のステータスを返す。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 4 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. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

2. グループノードを更新します。  
グループ内のすべてのノードを更新するには、以下のコマンドを実行します。

```
$ openstack overcloud update run --limit <GROUP_NAME>
```

グループ内の単一ノードを更新するには、以下のコマンドを実行します。

```
$ openstack overcloud update run --limit <GROUP_NAME> [NODE_INDEX]
```



### 注記

ノードを個別に更新する場合は、必ずすべてのノードを更新してください。

グループ内の最初のノードのインデックスはゼロ (0) です。たとえば、**CephStorage** という名前のグループの最初のノードを更新するには、以下のコマンドを実行します。

```
openstack overcloud update run --limit CephStorage[0]
```

3. ノードの更新が完了するまで待ちます。
4. Ceph Storage container update コマンドを実行して **ceph-ansible** を外部プロセスとして実行し、Red Hat Ceph Storage 4 コンテナを更新します。

```
$ openstack overcloud external-update run --tags ceph
```

5. Ceph Storage コンテナの更新が完了するまで待ちます。

## 3.9. データベースのオンライン更新の実施

一部のオーバークラウドコンポーネントでは、データベーステーブルのオンライン更新 (または移行) が必要です。オンラインでのデータベース更新を実行するには、**online\_upgrade** タグが付いたタスクに対して **openstack overcloud external-updaterun** コマンドを実行します。

データベースのオンライン更新は、次のコンポーネントに適用されます。

- OpenStack Block Storage (cinder)
- OpenStack Compute (nova)

### 手順

1. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

2. **online\_upgrade** のタグを使用するタスクに対して **openstack overcloud external-update run** コマンドを実行します。

```
$ openstack overcloud external-update run --tags online_upgrade
```

## 3.10. 更新の最終処理

最新の RHOSP16.2 バージョンへの更新を完了するには、オーバークラウドスタックを更新する必要があります。これにより、スタックのリソース構造が OSP 16.2 の標準のデプロイメントと一致し、今後、通常の **openstack overcloud deploy** の機能を実行できるようになります。

### 手順

1. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

2. オーバークラウドでフェンシングを再度有効にするには、**fencing.yaml** 環境ファイルで、**EnableFencing** パラメーターを **true** に設定します。
3. 更新の最終処理のコマンドを実行します。

```
$ openstack overcloud update converge \  
  --templates \  
  --stack <stack_name> \  
  -r <roles_data_file> \  
  -n <network_data_file> \  
  -e <environment_file> \  
  -e <environment_file> \  
  ...  
  ...
```

以下のオプションの中で、お使いの環境に適切なオプションを追加します。

- **EnableFencing** パラメーターが **true** に設定された **fencing.yaml** 環境ファイル。
  - オーバークラウドスタックの名前がデフォルトの名前 **overcloud** とは異なる場合は、更新の準備コマンドに **--stack** オプションを追加し、**<stack\_name>** を実際のスタック名に置き換えます。
  - カスタムロールを使用する場合は、カスタムロール (**<roles\_data>**) ファイル (**-r**) を含めます。
  - カスタムネットワークを使用する場合は、コンポーザブルネットワーク (**<network\_data>**) のファイルを追加します (**-n**)
  - すべてのカスタム設定環境ファイル (**-e**)
4. 更新の最終処理が完了するまで待ちます。



## 第4章 オーバークラウドの再起動

最新の 16.2 バージョンへの Red Hat Open Stack Platform (RHOSP) のマイナー更新実行後、オーバークラウドを再起動します。リブートにより、関連付けられたカーネル、システムレベル、およびコンテナコンポーネントの更新と共にノードがリフレッシュされます。これらの更新により、パフォーマンスとセキュリティ上のメリットが得られます。ダウンタイムを計画して、リブート手順を実施します。

以下のガイドを使用して、さまざまなノードのタイプを再起動する方法を説明します。

- 1つのロールで全ノードを再起動する場合は、各ノードを個別に再起動します。ロールの全ノードを同時に再起動すると、その操作中サービスにダウンタイムが生じる場合があります。
- 次の順序でノードの再起動の手順を完了します。
  1. [「コントローラーノードおよびコンポーザブルノードの再起動」](#)
  2. [「Ceph Storage \(OSD\) クラスターの再起動」](#)
  3. [「コンピュートノードの再起動」](#)

### 4.1. コントローラーノードおよびコンポーザブルノードの再起動

設定可能なロールに基づいてコントローラーノードとスタンドアロンノードを再起動し、コンピュートノードと Ceph ストレージノードを除外します。

#### 手順

1. リブートするノードにログインします。
2. オプション: ノードが Pacemaker リソースを使用している場合は、クラスターを停止します。

```
[heat-admin@overcloud-controller-0 ~]$ sudo pcs cluster stop
```

3. ノードをリブートします。

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

4. ノードが起動するまで待ちます。

#### 検証

1. サービスが有効になっていることを確認します。
  - a. ノードが Pacemaker サービスを使用している場合は、ノードがクラスターに再度加わったか確認します。

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

- b. ノードが Systemd サービスを使用している場合は、すべてのサービスが有効化されていることを確認します。

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

- c. ノードがコンテナ化されたサービスを使用している場合は、ノード上の全コンテナがアクティブであることを確認します。

```
[heat-admin@overcloud-controller-0 ~]$ sudo podman ps
```

## 4.2. CEPH STORAGE (OSD) クラスターの再起動

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 podman exec -it ceph-mon-controller-0 ceph -s
```

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** のステータスを返す。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 4 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 podman exec -it ceph-mon-controller-0 ceph osd set noout
$ sudo podman exec -it ceph-mon-controller-0 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** フラグの設定時にクラスター名を指定する必要があります。例: **sudo podman exec -it ceph-mon-controller-0 ceph osd set noout --cluster <cluster\_name>**

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. ノードがブートするまで待ちます。
5. ノードにログインして、クラスターのステータスを確認します。

```
$ sudo podman exec -it ceph-mon-controller-0 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、クラスターの再調整を再度有効にします。

```
$ sudo podman exec -it ceph-mon-controller-0 ceph osd unset noout
$ sudo podman exec -it ceph-mon-controller-0 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クラスター名を指定する必要があります。例: **sudo podman exec -it ceph-mon-controller-0 ceph osd set noout --cluster <cluster\_name>**

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 podman exec -it ceph-mon-controller-0 ceph status
```

## 4.3. コンピュートノードの再起動

Red Hat Open Stack Platform 環境でのインスタンスのダウンタイムを最小限に抑えるために、[インスタンスの移行ワークフロー](#)では、再起動するコンピュートノードからインスタンスを移行する時に必要な手順を概説します。



### 注記

インスタンスをソースコンピュートノードから別のコンピュートノードに移行しない場合、インスタンスがソースコンピュートノードで再起動され、アップグレードが失敗する可能性があります。これは、Podman と libvirt サービスの変更に関する既知の問題に関連しています。

- [BZ#2009106 - podman panic after tripleo\\_nova\\_libvirt restart two times](#)
- [BZ#2010135 - podman panic after tripleo\\_nova\\_libvirt restart two times](#)

### インスタンスの移行ワークフロー

1. コンピュートノードを再起動する前に、インスタンスを別のノードに移行するか決定します。
2. 再起動するコンピュートノードを選択して無効にし、新規インスタンスをプロビジョニングしないようにします。
3. インスタンスを別のコンピュートノードに移行します。
4. 空のコンピュートノードを再起動します。
5. 空のコンピュートノードを有効にします。

### 前提条件

- コンピュートノードを再起動する前に、ノードの再起動中にインスタンスを別のコンピュートノードに移行するか決定する。  
コンピュートノード間で仮想マシンインスタンスを移行する際に発生する可能性のある移行の制約リストを確認する。詳細は、[インスタンス作成のための Compute サービスの設定の移行の制約](#)を参照してください。
- インスタンスを移行できない場合は、以下のコアテンプレートパラメーターを設定して、コンピュートノード再起動後のインスタンスの状態を制御する。

#### NovaResumeGuestsStateOnHostBoot

リブート後のコンピュートノードで、インスタンスを同じ状態に戻すか定義します。**False**に設定すると、インスタンスは停止した状態を維持し、手動で起動する必要があります。デフォルト値は **False** です。

#### NovaResumeGuestsShutdownTimeout

再起動する前に、インスタンスのシャットダウンを待つ秒数。この値を **0** に設定することは推奨されません。デフォルト値は **300** です。

オーバークラウドパラメーターおよびその使用方法についての詳細は、[Overcloud Parameters](#)を参照してください。

### 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. 全コンピュートノードとその UUID を一覧表示します。

```
$ source ~/stackrc
(undercloud) $ openstack server list --name compute
```

リブートするコンピュートノードの UUID を特定します。

3. アンダークラウドから、コンピュートノードを選択し、そのノードを無効にします。

```
$ source ~/overcloudrc
(overcloud) $ openstack compute service list
(overcloud) $ openstack compute service set <hostname> nova-compute --disable
```

4. コンピュートノード上の全インスタンスを一覧表示します。

```
(overcloud) $ openstack server list --host <hostname> --all-projects
```

5. オプション: インスタンスを別のコンピュートノードに移行する場合には、以下の手順を実行します。
  - a. インスタンスを別のコンピュートノードに移行する場合は、以下のコマンドのいずれかを使用します。

- インスタンスを別のホストに移行するには、次のコマンドを実行します。

```
(overcloud) $ openstack server migrate <instance_id> --live <target_host> --wait
```

- **nova-scheduler** がターゲットホストを自動的に選択できるようにします。

```
(overcloud) $ nova live-migration <instance_id>
```

- すべてのインスタンスを一度にライブマイグレーションします。

```
$ nova host-evacuate-live <hostname>
```



#### 注記

**nova** コマンドで非推奨の警告が表示される可能性がありますが、無視しても問題ありません。

- b. 移行が完了するまで待ちます。
- c. 移行が正常に完了したことを確認します。

```
(overcloud) $ openstack server list --host <hostname> --all-projects
```

- d. コンピュートノードのインスタンスがなくなるまで、移行を続けます。

6. コンピュートノードにログインして、ノードをリブートします。

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

7. ノードが起動するまで待ちます。
8. コンピュートノードを再度有効にします。

```
$ source ~/overcloudrc  
(overcloud) $ openstack compute service set <hostname> nova-compute --enable
```

9. コンピュートノードが有効であることを確認します。

```
(overcloud) $ openstack compute service list
```