



Red Hat OpenStack Platform 11

Red Hat OpenStack Platform のアップグレード

Red Hat OpenStack Platform 環境のアップグレード

Red Hat OpenStack Platform 11 Red Hat OpenStack Platform のアップグレード

Red Hat OpenStack Platform 環境のアップグレード

OpenStack Team
rhos-docs@redhat.com

法律上の通知

Copyright © 2018 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

本ドキュメントでは、Red Hat OpenStack Platform 10 (Newton) から 11 (Ocata) にアップグレードする複数の異なる方法について説明します。これらの方法は、アップグレード元およびアップグレード先のいずれも Red Hat Enterprise Linux 7 をベースにインストールされたデプロイメントであることを前提としています。

目次

第1章 はじめに	3
1.1. アップグレードシナリオの比較	3
1.2. リポジトリの要件	4
パート I. DIRECTOR ベースの環境	6
第2章 DIRECTOR ベースの環境: マイナーバージョンへの更新の実行	7
2.1. DIRECTOR パッケージの更新	7
2.2. オーバークラウドイメージの更新	8
2.3. オーバークラウドパッケージの更新	9
第3章 DIRECTOR ベースの環境: メジャーバージョンのアップグレードの実行	12
3.1. アップグレードのサポートステートメント	13
3.2. ライブマイグレーションの更新	14
3.3. オーバークラウドの確認	14
3.4. アンダークラウドのアップグレード	15
3.4.1. director のアップグレード	15
3.4.2. オーバークラウドイメージのアップグレード	17
3.5. オーバークラウドのアップグレード前の設定	18
3.5.1. Red Hat サブスクリプションの詳細	18
3.5.2. 非推奨および新規のコンポーザブルサービス	18
3.5.3. ロールの手動アップグレード	19
3.5.4. ストレージバックエンド	20
3.5.5. NFV の設定	21
3.5.6. オーバークラウドのパラメーター	21
3.5.7. カスタムのコアテンプレート	22
3.6. オーバークラウドのアップグレード	22
3.6.1. オーバークラウドノードのアップグレード	23
3.6.2. Object Storage ノードのアップグレード	23
3.6.3. コンピュートノードのアップグレード	24
3.6.4. アップグレードの最終処理	26
3.7. オーバークラウドのアップグレード後の注意事項	27
第4章 DIRECTOR ベースのアップグレードのトラブルシューティング	29
4.1. アンダークラウドのアップグレード	29
4.2. オーバークラウドのアップグレード	29
パート II. DIRECTOR 以外の環境	32
第5章 RED HAT OPENSTACK PLATFORM 環境の手動アップグレード	33
パート III. 付録	34
付録A NFV アップグレードファイルのサンプル	35

第1章 はじめに

本ガイドは、Red Hat OpenStack Platform を最新の状態に保つためのプロセスについて説明します。アップグレードおよび更新は、**Red Hat OpenStack Platform 11 (Ocata)** をターゲットとします。

Red Hat は、Red Hat Enterprise Linux 7.3 をベースとする Red Hat OpenStack Platform 11 へのアップグレードのみをサポートしており、以下のいずれかの条件に基づいた異なるシナリオを推奨しています。

- director ベースのオーバークラウドまたは手動で作成した環境を使用している。
- 1 クラスター内で複数のコントローラーノードを管理する高可用性ツールを使用している。

「[アップグレードシナリオの比較](#)」には、すべてのアップグレードシナリオについての説明を記載しています。これらのシナリオにより、正常に機能する Red Hat OpenStack Platform 11 へのアップグレードと、バージョン 11 内でのマイナーな更新が可能となります。

1.1. アップグレードシナリオの比較

Red Hat では、Red Hat OpenStack Platform 11 には以下のアップグレードシナリオを推奨しています。以下の表には、各シナリオについての説明をまとめています。

表1.1 アップグレードシナリオ

メソッド	説明
director ベースの環境: マイナーバージョンへの更新の実行	<p>このシナリオでは、Red Hat OpenStack Platform 11 のマイナーバージョン間の更新を行います。これには、以下の作業が必要となります。</p> <ul style="list-style-type: none"> ● director パッケージの更新 ● director を使用した、オーバークラウド内の全ノードでのパッケージ更新の起動
director ベースの環境: メジャーバージョンへのアップグレードの実行	<p>このシナリオでは、Red Hat OpenStack Platform のメジャーバージョンからのアップグレードを行います。この場合は、バージョン 10 から 11 にアップグレードする手順です。これには、以下の作業が必要となります。</p> <ul style="list-style-type: none"> ● director パッケージの更新 ● director を使用した、各ノード上のアップグレードスクリプトセットの提供 ● オーバークラウドスタックのアップグレードの実行

すべての方法に共通する注意事項

- 全ホスト上でこのリリースの正しいリポジトリが有効化されていることを確認します。
- アップグレード時には、一部のサービスを停止する必要があります。

- コンピュートノードを再起動したり、インスタンスを明示的にシャットダウンしたりしない限りは、アップグレードプロセスは、実行中のインスタンスには影響を及ぼしません。



警告

Red Hat は、Red Hat OpenStack Platform のベータリリースからサポート対象リリースへのアップグレードは一切サポートしていません。

1.2. リポジトリの要件

アンダークラウドおよびオーバークラウドにはいずれも、Red Hat コンテンツ配信ネットワーク (CDN) か Red Hat Satellite 5 または 6 を使用した Red Hat リポジトリへのアクセスが必要です。Red Hat Satellite サーバーを使用する場合は、必要なリポジトリをお使いの OpenStack Platform 環境に同期します。以下の CDN チャンネル名一覧を参考にしてください。

表1.2 OpenStack Platform リポジトリ

名前	リポジトリ	説明
Red Hat Enterprise Linux 7 Server (RPMs)	rhel-7-server-rpms	ベースオペレーティングシステムのリポジトリ
Red Hat Enterprise Linux 7 Server - Extras (RPMs)	rhel-7-server-extras-rpms	Red Hat OpenStack Platform の依存関係が含まれます。
Red Hat Enterprise Linux 7 Server - RH Common (RPMs)	rhel-7-server-rh-common-rpms	Red Hat OpenStack Platform のデプロイと設定ツールが含まれます。
Red Hat Satellite Tools for RHEL 7 Server RPMs x86_64	rhel-7-server-satellite-tools-6.2-rpms	Red Hat Satellite 6 でのホスト管理ツール
Red Hat Enterprise Linux High Availability (for RHEL 7 Server) (RPMs)	rhel-ha-for-rhel-7-server-rpms	Red Hat Enterprise Linux の高可用性ツール。コントローラーノードの高可用性に使用します。
Red Hat OpenStack Platform 11 for RHEL 7 (RPMs)	rhel-7-server-openstack-11-rpms	Red Hat OpenStack Platform のコアリポジトリ

Ceph クラスターを使用している場合には、以下のリポジトリも必要となります。

Red Hat Ceph Storage OSD 2.0 for Red Hat Enterprise Linux 7 Server (RPMs)	rhel-7-server-rhceph-2-osd-rpms	(Ceph Storage ノード向け) Ceph Storage Object Storage デーモンのリポジトリ。Ceph Storage ノードにインストールします。
---	--	---

Red Hat Ceph Storage MON 2.0 for Red Hat Enterprise Linux 7 Server (RPMs)	rhel-7-server-rhceph-2-mon-rpms	(Ceph Storage ノード向け) Ceph Storage Monitor デーモンのリポジトリ。Ceph Storage ノードを使用して OpenStack 環境にあるコントローラーノードにインストールします。
Red Hat Ceph Storage Tools 2 for Red Hat Enterprise Linux 7 Workstation (RPMs)	rhel-7-server-rhceph-2-tools-rpms	(Ceph Storage ノード向け) Ceph オブジェクトストレージに必要な Rados REST ゲートウェイを提供します。



注記

ネットワークがオフラインの Red Hat OpenStack Platform 環境向けにリポジトリを設定するには、[「オフライン環境で Red Hat OpenStack Platform Director を設定する」](#)の記事を参照してください。

パート I. DIRECTOR ベースの環境

第2章 DIRECTOR ベースの環境: マイナーバージョンへの更新の実行

本項では、同じバージョンの Red Hat OpenStack Platform 環境の更新方法について考察します。この場合は、Red Hat OpenStack Platform 11 が対象です。これには、アンダークラウドとオーバークラウドの両方の機能の更新が含まれます。



警告

インスタンスの高可用性を実装している場合には、アップグレードやスケールアップはできないので (『[コンピュータインスタンスの高可用性](#)』で説明しています)、操作を試みても失敗してしまいます。

また、インスタンスの高可用性を有効にすると、将来、director を使用したアップグレードはできなくなります。これは、メジャーアップグレードとマイナーアップグレードの両方に該当します。詳しくは、<https://access.redhat.com/ja/solutions/2898841> を参照してください。

この手順は、いずれの場合も以下のワークフローに従って作業を行います。

1. Red Hat OpenStack Platform director パッケージの更新
2. Red Hat OpenStack Platform director でのオーバークラウドイメージの更新
3. Red Hat OpenStack Platform director を使用したオーバークラウドパッケージの更新

Red Hat は、更新を実行する前に以下の作業を済ませておくことを推奨します。

- アップグレード手順のステップを開始する前に、アンダークラウドノードのバックアップを実行してください。バックアップの手順については、『[director のアンダークラウドのバックアップと復元](#)』ガイドを参照してください。
- 実稼働環境で手順を実行する前に、すべての変更が加えられたテスト環境で更新の手順を実行します。
- 更新を実行するにあたってアドバイスやサポートが必要な場合には、Red Hat までご連絡ください。

2.1. DIRECTOR パッケージの更新

director は、標準の RPM メソッドを使用して環境を更新します。このメソッドでは、**yum** コマンドを実行して、director のホストが確実に最新のパッケージを使用するようにします。

1. director に **stack** ユーザーとしてログインします。
2. 主要な OpenStack Platform サービスを停止します。

```
$ sudo systemctl stop 'openstack-*' 'neutron-*' httpd
```



注記

これにより、アンダークラウドで短時間のダウンタイムが生じます。アンダークラウドの更新中もオーバークラウドは引き続き機能します。

3. **python-tripleoclient** パッケージと依存関係を更新し、マイナーバージョンの更新向けの最新のスクリプトを使用できるようにします。

```
$ sudo yum update -y instack-undercloud openstack-puppet-modules
openstack-tripleo-common python-tripleoclient
```

4. **director** は **openstack undercloud upgrade** コマンドを使用して、アンダークラウドの環境を更新します。以下のコマンドを実行します。

```
$ openstack undercloud upgrade
```

ノードを再起動して、新しいシステム設定を有効にし、全アンダークラウドサービスを更新します。

1. ノードを再起動します。

```
$ sudo reboot
```

2. ノードが起動するまで待ちます。

ノードが起動したら、全サービスのステータスを確認します。

```
$ sudo systemctl list-units "openstack*" "neutron*" "openvswitch"
```



注記

再起動後に **openstack-nova-compute** が有効になるまでに約 10 分かかる場合があります。

オーバークラウドとそのノードが存在しているかどうかを確認します。

```
$ source ~/stackrc
$ openstack server list
$ openstack baremetal node list
$ openstack stack list
```

2.2. オーバークラウドイメージの更新

アンダークラウドの更新プロセスにより、**rhosp-director-images** および **rhosp-director-images-ipa** パッケージから新規イメージアーカイブがダウンロードされる可能性があります。**yum** ログをチェックして、新規イメージアーカイブが利用可能かどうかを確認してください。

```
$ sudo grep "rhosp-director-images" /var/log/yum.log
```

新規アーカイブが利用可能な場合には、現在のイメージを新規イメージに置き換えてください。新しいイメージをインストールするには、最初に **stack** ユーザーの **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/*
```

アーカイブを展開します。

```
$ cd ~/images
$ for i in /usr/share/rhosp-director-images/overcloud-full-latest-11.0.tar
  /usr/share/rhosp-director-images/ironic-python-agent-latest-11.0.tar; do
  tar -xvf $i; done
```

最新のイメージを director にインポートして、ノードがこれらの新規イメージを使用するように設定します。

```
$ cd ~
$ openstack overcloud image upload --update-existing --image-path ~/images
$ openstack overcloud node configure $(openstack baremetal node list -c
  UUID -f csv --quote none | sed "1d" | paste -s -d " ")
```

新規イメージの存在をチェックして、イメージの更新を最終確認します。

```
$ openstack image list
$ ls -l/httpboot
```

director が更新され、最新のイメージを使用するようになりました。この更新の後にはサービスを再起動する必要はありません。

2.3. オーバークラウドパッケージの更新

オーバークラウドでは、環境の更新に標準の RPM メソッドを使用します。これには、以下の2つのステップを実行する必要があります。

1. 元の **openstack overcloud deploy** コマンドに **--update-plan-only** オプションを追加して、現在のプランを更新します。以下に例を示します。

```
$ openstack overcloud deploy --update-plan-only \
  --templates \
  -e /usr/share/openstack-tripleo-heat-
  templates/environments/network-isolation.yaml \
  -e /home/stack/templates/network-environment.yaml \
  -e /home/stack/templates/storage-environment.yaml \
  -e /home/stack/templates/rhel-registration/environment-rhel-
  registration.yaml \
  [-e <environment_file>|...]
```

--update-plan-only のオプションを指定すると、director に保管されているオーバークラウドのプランのみが更新されます。**-e** オプションを使用して、オーバークラウドと関連のある環境ファイルとその更新パスを追加します。後で実行される環境ファイルで定義されているパラメーターとリソースが優先されることになるため、環境ファイルの順序は重要となります。以下の一覧は、環境ファイルの順序の例です

- Heat テンプレートコレクションの初期化ファイル (**environments/network-isolation.yaml**) を含むネットワーク分離ファイルと、次にカスタムの NIC 設定ファイル

- 外部のロードバランシングの環境ファイル
- ストレージの環境ファイル
- Red Hat CDN または Satellite 登録用の環境ファイル
- その他のカスタム環境ファイル

2. **openstack overcloud update** コマンドを使用して、全ノードでパッケージの更新を実行します。以下に例を示します。

```
$ openstack overcloud update stack -i overcloud
```

全ノードで並行して更新を実行すると問題が発生する可能性があります。たとえば、パッケージの更新には、サービスの再起動が必要となる場合があります、その操作によって他のノードが中断される可能性があります。そのため、このプロセスでは、一連のブレークポイントを設けて、ノードごとに更新します。1つのノードでパッケージの更新が完了すると、更新プロセスは次のノードに移ります。更新のプロセスには、各ブレークポイントで確認が要求される対話モードでコマンドを実行するための **-i** オプションも必要です。**-i** オプションを使用しない場合には、更新は最初のブレークポイントで一時停止の状態のままとなります。

これで更新のプロセスが開始します。このプロセス中に、director は **IN_PROGRESS** のステータスを報告して、ブレークポイントを通過するように定期的に要求します。以下に例を示します。

```
not_started: [u'overcloud-controller-0', u'overcloud-controller-1',
u'overcloud-controller-2']
on_breakpoint: [u'overcloud-compute-0']
Breakpoint reached, continue? Regexp or Enter=proceed, no=cancel update,
C-c=quit interactive mode:
```

Enter を押すと、**on_breakpoint** 一覧の最後のノードからブレークポイントを通過します。これで、そのノードの更新が開始します。また、ノード名を入力して特定のノードでブレークポイントを通過したり、複数のノードで一度にブレークポイントを通過するための Python ベースの正規表現を入力することも可能です。ただし、複数のコントローラーノードで同時にブレークポイントを通過することはお勧めしません。全ノードが更新を完了するまで、このプロセスを継続します。

更新が完了すると、コマンドにより **COMPLETE** のステータスが報告されます。

```
...
IN_PROGRESS
IN_PROGRESS
IN_PROGRESS
COMPLETE
update finished with status COMPLETE
```

コントローラーノードにフェンシングを設定している場合には、更新プロセスによってその設定が無効になる場合があります。更新プロセスの完了時には、コントローラーノードの1つで以下のコマンドを実行してフェンシングを再度有効にします。

```
$ sudo pcs property set stonith-enabled=true
```

更新プロセスを実行しても、オーバークラウド内のノードは自動的に再起動しません。カーネルまたは Open vSwitch のメジャー/マイナーバージョンが更新された場合 (例: オーバークラウドのオペレーティングシステムが Red Hat Enterprise Linux 7.2 から 7.3 に更新された場合や、Open vSwitch がバージョン 2.4 から 2.5 に更新された場合など) には再起動が必要です。各ノードの **/var/log/yum.log**

ファイルをチェックして、**kernel** または **openvswitch** のパッケージのメジャー/マイナーバージョンが更新されているかどうかを確認します。更新されている場合には、『**director** インストールと使用方法』ガイドの「[ノードの再起動](#)」の手順に従って各ノードを再起動します。

第3章 DIRECTOR ベースの環境: メジャーバージョンのアップグレードの実行



警告

最新のメジャーバージョンにアップグレードする前に、アンダークラウドとオーバークラウドが最新のマイナーバージョンに更新されていることを確認してください。これには、OpenStack Platform サービスとベースオペレーティングシステムの両方が含まれます。マイナーバージョンを更新するプロセスについては、Red Hat OpenStack Platform 10 『Red Hat OpenStack Platform のアップグレード』の「[director ベースの環境: マイナーバージョンへの更新の実行](#)」の章を参照してください。先にマイナーバージョンを更新せずに、メジャーバージョンをアップグレードすると、アップグレードプロセスが失敗してしまう可能性があります。



警告

インスタンスの高可用性を実装している場合には、アップグレードやスケールアップはできないので (『[コンピュータインスタンスの高可用性](#)』で説明しています)、操作を試みても失敗してしまいます。

また、インスタンスの高可用性を有効にすると、将来、director を使用したアップグレードはできなくなります。これは、メジャーアップグレードとマイナーアップグレードの両方に該当します。詳しくは、<https://access.redhat.com/ja/solutions/2898841> を参照してください。

本章では、アンダークラウドとオーバークラウドを次のメジャーバージョンにアップグレードする方法を考察します。今回は、Red Hat OpenStack Platform 10 から Red Hat OpenStack Platform 11 へのアップグレードです。

この手順では以下のワークフローに従って作業を行います。

1. Red Hat OpenStack Platform director パッケージのアップグレード
2. Red Hat OpenStack Platform director でのオーバークラウドイメージのアップグレード
3. カスタムの Heat テンプレートや環境ファイルなど、オーバークラウドのカスタマイズの更新
4. コンポーザブルサービスのアップグレードをサポートする全ノードのアップグレード
5. Object Storage の個別アップグレード
6. コンピュートノードの個別アップグレード
7. オーバークラウドのアップグレードの最終処理

3.1. アップグレードのサポートステートメント

アップグレードプロセスを成功させるには、メジャーバージョン間の変更に対応するための準備が必要です。以下のサポートステートメントを確認して、Red Hat OpenStack Platform のアップグレードのプランニングに役立ててください。

Red Hat OpenStack Platform director でのアップグレードは、ライブの実稼働環境で実行する前に、その環境固有の設定で全面的にテストする必要があります。Red Hat では、director を使用する場合の標準のオプションとして提供されているユースケースの大半とそれらの組み合わせを検証済みですが、可能な組み合わせの数が多いため、それらは完全に網羅されません。さらに、標準デプロイメントの設定が変更された場合には、手動または設定後のフックを使用して、実稼働用以外の環境でアップグレード機能をテストすることが極めて重要です。そのため、以下を実行することを推奨します。

- アンダークラウドノードのバックアップを実行してから、アップグレードの手順のステップを開始します。バックアップの手順は、『[director のアンダークラウドのバックアップと復元](#)』ガイドを参照してください。
- カスタマイズされた設定を使用するアップグレード手順は、実稼働環境で実行する前にテスト環境で実行してください。
- このアップグレードの実行するにあたって懸念がある場合には、作業を開始する前に Red Hat のサポートチームに連絡して、アップグレードのプロセスについてのアドバイスおよびサポートを依頼してください。

本項で説明するアップグレードプロセスは、director を使ったカスタマイズにのみ対応しています。director を使用せずにオーバークラウドの機能をカスタマイズした場合は、以下のステップを実行してください。

- その機能を無効にします。
- オーバークラウドをアップグレードします。
- アップグレードの完了後に機能を再度有効にします。

これは、アップグレードがすべて完了するまで、カスタマイズされた機能が使用できないことを意味します。

Red Hat OpenStack Platform director 11 は、Red Hat OpenStack Platform の以前のオーバークラウドバージョンを管理できます。詳しい情報は、以下のサポートマトリックスを参照してください。

表3.1 Red Hat OpenStack Platform director 11 のサポートマトリックス

バージョン	オーバークラウドの更新	オーバークラウドのデプロイ	オーバークラウドのスケールリング
Red Hat OpenStack Platform 11	Red Hat OpenStack Platform 11 および 10	Red Hat OpenStack Platform 11 および 10	Red Hat OpenStack Platform 11 および 10

旧バージョンのオーバークラウドを管理している場合には、以下の Heat テンプレートコレクションを使用してください。

- Red Hat OpenStack Platform 10: `/usr/share/openstack-tripleo-heat-templates/newton/`

以下に例を示します。

```
$ openstack overcloud deploy --templates /usr/share/openstack-tripleo-heat-templates/newton/ [OTHER_OPTIONS]
```

アップグレードの一般的なヒントは以下のとおりです。

- 各ステップの後には、コントローラーノードのクラスターで **pcs status** コマンドを実行して、リソースにエラーが発生していないことを確認します。
- このアップグレードの実行に関して何らかの懸念がある場合は、作業を開始する前に Red Hat に連絡して、アップグレードプロセスについてのアドバイスおよびサポートを依頼してください。

3.2. ライブマイグレーションの更新

コンピュートノードをアップグレードするには、ライブマイグレーションを有効化して、アップグレード中にインスタンスを利用可能な状態に確保する必要があります。これには、**OS::TripleO::Services::Sshd** サービスが必要です。このサービスは、Red Hat OpenStack Platform 10 の最新版にデフォルトのロールとして追加された新規サービスです。Red Hat OpenStack Platform 11 へのアップグレード中にライブマイグレーションが有効化されるようにするには、以下の手順を実行します。

- アンダークラウドを最新バージョンの Red Hat OpenStack Platform 10 に更新します。
- デフォルトのロールデータファイルを使用する場合には、各ロールに **OS::TripleO::Services::Sshd** サービスが含まれていることを確認してください。カスタムのロールデータファイルを使用する場合には、この新しいサービスを各ロールに追加します。
- **OS::TripleO::Services::Sshd** サービスが含まれている最新バージョンの Red Hat OpenStack Platform 10 にオーバークラウドを更新します。
- Red Hat OpenStack Platform 11 へのアップグレードを開始します。

これにより、コンピュートノードが相互に SSH アクセスできるようになります。このアクセスは、ライブマイグレーションのプロセスで必要となります。

重要

以前のバージョンの Red Hat OpenStack Platform では、最近のセキュリティー修正プログラム (CVE-2017-2637) によってライブマイグレーションが無効化されました。**OS::TripleO::Services::Sshd** サービスは、Red Hat OpenStack Platform 10 以降のバージョンでこの問題を解決します。詳しくは、以下のリンクを参照してください。

- [ナレッジベース: 「Red Hat OpenStack Platform director \(TripleO\) CVE-2017-2637 bug and Red Hat OpenStack Platform」](#)
- [「CVE-2017-2637」](#)

3.3. オーバークラウドの確認

アップグレードを実行する前に、オーバークラウドが安定した状態であることをチェックします。director で以下のステップを実行して、オーバークラウド内の全サービスが稼働していることを確認してください。

1. 高可用性サービスのステータスを確認します。

```
ssh heat-admin@[CONTROLLER_IP] "sudo pcs resource cleanup ; sleep 60 ; sudo pcs status"
```

[CONTROLLER_IP] はコントローラーノードの IP アドレスに置き換えます。このコマンドは、オーバークラウドの Pacemaker クラスターを最新の状態に更新し、60 秒待ってからそのクラスターのステータスを報告します。

2. オーバークラウドノードで、失敗した OpenStack Platform **systemd** サービスがあるかどうかを確認します。以下のコマンドは、失敗したサービスを全ノードで確認します。

```
$ for IP in $(openstack server list -c Networks -f csv | sed '1d' | sed 's/"//g' | cut -d '=' -f2) ; do echo "Checking systemd services on $IP" ; ssh heat-admin@$IP "sudo systemctl list-units 'openstack-*' 'neutron-*' --state=failed --no-legend" ; done
```

3. 各ノードで **os-collect-config** が実行中であることを確認します。以下のコマンドで、このサービスのステータスを各ノードでチェックします。

```
$ for IP in $(openstack server list -c Networks -f csv | sed '1d' | sed 's/"//g' | cut -d '=' -f2) ; do echo "Checking os-collect-config on $IP" ; ssh heat-admin@$IP "sudo systemctl list-units 'os-collect-config.service' --no-legend" ; done
```

重要

OpenStack Platform 10 でスタンドアロンの Keystone ノードを使用している場合には、**keystone** と **gnocchi** の間の競合が原因となって「openstack-gnocchi-statsd」サービスが正しく起動していない可能性があります。コントローラーノードまたは Telemetry ノードで「openstack-gnocchi-statsd」サービスを確認して、失敗した場合には、オーバークラウドのアップグレード前にサービスを再起動します。この問題は、[BZ#1447422](#) で対処しています。

3.4. アンダークラウドのアップグレード

3.4.1. director のアップグレード

Red Hat OpenStack Platform director をアップグレードするには、以下の手順に従ってください。

1. director に **stack** ユーザーとしてログインします。
2. OpenStack Platform のリポジトリを更新します。

```
$ sudo subscription-manager repos --disable=rhel-7-server-openstack-10-rpms
$ sudo subscription-manager repos --enable=rhel-7-server-openstack-11-rpms
```

上記のコマンドは、**yum** が最新のリポジトリを使用するように設定します。

3. 主要な OpenStack Platform サービスを停止します。

```
$ sudo systemctl stop 'openstack-*' 'neutron-*' httpd
```

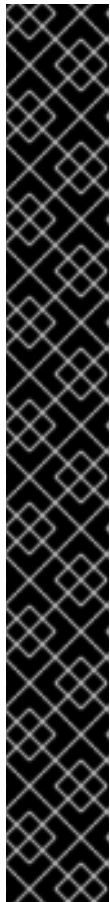


注記

これにより、アンダークラウドで短時間のダウンタイムが生じます。アンダークラウドのアップグレード中もオーバークラウドは引き続き機能します。

4. **yum** で **director** の主要なパッケージをアップグレードします。

```
$ sudo yum update -y instack-undercloud openstack-puppet-modules
openstack-tripleo-common python-tripleoclient
```



重要

デフォルトのプロビジョニング/コントロールプレーンネットワークは、**192.0.2.0/24** から **192.168.24.0/24** に変更されています。このネットワークにデフォルト値を使用している場合には、**undercloud.conf** で以下のパラメーターをチェックします。

- **local_ip**
- **network_gateway**
- **undercloud_public_vip**
- **undercloud_admin_vip**
- **network_cidr**
- **masquerade_network**
- **dhcp_start**
- **dhcp_end**

openstack undercloud upgrade コマンドを実行する前に、ネットワーク設定が適切に設定されていることを確認します。

5. 以下のコマンドでアンダークラウドをアップグレードします。

```
$ openstack undercloud upgrade
```

このコマンドにより、**director** のパッケージがアップグレードされ、**director** の設定が最新の状態に更新されて、バージョンの変更後に指定されていない設定内容が追加されます。このコマンドによって、オーバークラウドのスタックデータや環境内の既存のノードのデータなど、保存されたデータは削除されません。

ノードを再起動して、新しいシステム設定を有効にし、全アンダークラウドサービスを更新します。

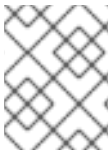
1. ノードを再起動します。

```
$ sudo reboot
```

2. ノードが起動するまで待ちます。

ノードが起動したら、全サービスのステータスを確認します。

```
$ sudo systemctl list-units "openstack*" "neutron*" "openvswitch"
```



注記

再起動後に **openstack-nova-compute** が有効になるまでに約 10 分かかる場合があります。

オーバークラウドとそのノードが存在しているかどうかを確認します。

```
$ source ~/stackrc
$ openstack server list
$ openstack baremetal node list
$ openstack stack list
```

必要な場合には `director` で設定ファイルを確認します。アップグレードされたパッケージで、Red Hat OpenStack Platform 11 バージョンのサービスに適した `.rpmnew` ファイルがインストールされている可能性があります。

3.4.2. オーバークラウドイメージのアップグレード

以下の手順では、ノードの検出とオーバークラウドのデプロイメントに向け最新のイメージが確保されるようにします。**rhosp-director-images** および **rhosp-director-images-ipa** のパッケージからの新規イメージは、アンダークラウドのアップグレードですでに更新されています。

`stack` ユーザーの `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/*
```

アーカイブを展開します。

```
$ cd ~/images
$ for i in /usr/share/rhosp-director-images/overcloud-full-latest-11.0.tar
  /usr/share/rhosp-director-images/ironic-python-agent-latest-11.0.tar; do
  tar -xvf $i; done
```

最新のイメージを `director` にインポートして、ノードがこれらの新規イメージを使用するように設定します。

```
$ cd ~
$ openstack overcloud image upload --update-existing --image-path ~/images
$ openstack overcloud node configure $(openstack baremetal node list -c
  UUID -f csv --quote none | sed "1d" | paste -s -d " ")
```

新規イメージの存在をチェックして、イメージの更新を最終確認します。

```
$ openstack image list
$ ls -l /httpboot
```

director は、最新のイメージを使用してアップグレードされました。



重要

オーバークラウドのイメージのバージョンがアンダークラウドのバージョンに対応していることを確認してください。

3.5. オーバークラウドのアップグレード前の設定

3.5.1. Red Hat サブスクリプションの詳細

オーバークラウドのアップグレードを実行する前には、そのサブスクリプション情報を更新して、環境に最新のリポジトリが使用されるようにします。Satellite の登録に環境ファイルを使用している場合には、その環境ファイルで以下のパラメーターを更新してください。

- **rhel_reg_repos**: オーバークラウドを有効化するためのリポジトリ。新しい Red Hat OpenStack Platform 11 リポジトリを含みます。有効化するリポジトリについては、「[リポジトリの要件](#)」を参照してください。
- **rhel_reg_activation_key**: Red Hat OpenStack Platform 11 リポジトリ用の新規アクティベーションキー
- **rhel_reg_sat_repo**: **katello-agent** など、Red Hat Satellite 6 の管理ツールが含まれるリポジトリを定義する新たなパラメーター。Red Hat Satellite 6 に登録する場合はこのパラメーターを更新するようにしてください。

環境ファイルのフォーマットについての詳しい情報および例は、『[オーバークラウドの高度なカスタマイズ](#)』ガイドの「[オーバークラウドの登録](#)」を参照してください。

3.5.2. 非推奨および新規のコンポーザブルサービス

以下の各項目は、カスタムの **roles_data.yaml** ファイルを使用してオーバークラウドロールを定義している場合に該当します。

カスタムの **roles_data.yaml** ファイルから以下の非推奨サービスを削除します。

サービス	説明
OS::TripleO::Services::Core	このサービスは、他の Pacemaker サービスのコアの依存関係として機能していましたが、高可用性のコンポーザブルサービスに対応するために削除されました。
OS::TripleO::Services::GlanceRegistry	このサービスは、OpenStack Image Storage (glance) のイメージメタデータを提供していましたが、glance v2 API で非推奨となったため削除されました。
OS::TripleO::Services::VipHosts	このサービスは、ノードのホスト名と IP アドレスで /etc/hosts ファイルを設定していましたが、director の Heat テンプレートに直接統合されました。

`roles_data.yaml` ファイルに以下の新しいサービスを追加します。

サービス	説明
<code>OS::TripleO::Services::MySQLClient</code>	他のコンポーザブルサービス用のデータベース設定を提供する MariaDB クライアントをノード上で設定します。このサービスは、スタンドアロンのコンポーザブルサービスを使用する全ロールに追加してください。
<code>OS::TripleO::Services::NovaPlacement</code>	OpenStack Compute (nova) Placement API を設定します。現在のオーバークラウドでスタンドアロンの Nova ロールを使用している場合には、そのロールにこのサービスを追加します。そうでない場合には、このサービスを Controller ロールに追加してください。
<code>OS::TripleO::Services::PankoApi</code>	OpenStack Telemetry Event Storage (panko) サービスを設定します。現在のオーバークラウドでスタンドアロンの Telemetry ロールを使用している場合には、このサービスをそのロールに追加します。そうでない場合には、このサービスを Controller ロールに追加してください。
<code>OS::TripleO::Services::Sshd</code>	全ノードに SSH アクセスを設定します。このアクセスは、インスタンスの移行に使用されます。

これらの新規サービスを必要とする可能性のある、オーバークラウドの別の箇所を更新します。以下に例を示します。

- **カスタムの ServiceNetMap パラメーター:** カスタムの **ServiceNetMap** を指定してオーバークラウドをアップグレードする場合には、新規サービス向けに最新の **ServiceNetMap** が含まれるようにしてください。デフォルトのサービス一覧は、`network/service_net_map.j2.yaml` ファイルの **ServiceNetMapDefaults** パラメーターで定義されます。カスタムの **ServiceNetMap** の使用方法については、『[オーバークラウドの高度なカスタマイズ](#)』の「[ネットワークの分離](#)」のセクションを参照してください。
- **外部ロードバランサー:** 外部ロードバランサーを使用する場合には、新規サービスを外部ロードバランサーの設定に追加します。詳しくは、『[オーバークラウド向けの外部ロードバランシング](#)』を参照してください。

3.5.3. ロールの手動アップグレード

アップグレードプロセスにより、全ロール上のコンポーザブルサービスが段階的にアップグレードされます。ただし、一部のロールには、ノードの個別アップグレードを行って、インスタンスおよびサービスの可用性を確保する必要があります。以下のロールが対象となります。

ロール	説明
-----	----

ロール	説明
Compute	インスタンスの稼動状態を確保するために、ノードごとに個別のアップグレードを実行する必要があります。これらのノードのプロセスでは、アップグレードの前にそのノードからインスタンスを移行しておく必要があります。
ObjectStorage	オーバークラウドが Object Storage サービス (swift) を利用できる状態を確保するために、ノードごとに個別のアップグレードを実行する必要があります。

以下のアップグレードプロセスでは **upgrade-non-controller.sh** コマンドを使用して、上記のロールのノードをアップグレードします。

Red Hat OpenStack Platform 11 のデフォルトの **roles_data.yaml** ファイルでは、これらのロールに **disable_upgrade_deployment: True** とマークされ、主要なコンポーザブルサービスのアップグレードプロセスで除外されます。この設定により、これらのロールが割り当てられたノードを個別にアップグレードすることが可能となります。ただし、これらのロールが含まれたカスタムの **roles_data.yaml** ファイルを使用する場合には、**Compute** および **ObjectStorage** のロールの定義に **disable_upgrade_deployment: True** パラメーターを必ず記載してください。以下に例を示します。

```
- name: Compute
  CountDefault: 1
  disable_upgrade_deployment: True
  ServicesDefault:
    - OS::Triple0::Services::CACerts
    - OS::Triple0::Services::CephClient
    - OS::Triple0::Services::CephExternal
  ...
```

Compute ロールなど、手動のアップグレードが必要な他のカスタムロールには、**disable_upgrade_deployment** を使用します。

3.5.4. ストレージバックエンド

一部のストレージバックエンドは、設定フックではなく、独自のコンポーザブルサービスを使用するように変更されました。カスタムストレージバックエンドを使用する場合には、**environments** ディレクトリで関連する環境ファイルに新規パラメーターとリソースが含まれているかどうかを確認してください。バックエンド用のカスタム環境ファイルを更新します。以下に例を示します。

- **NetApp Block Storage (cinder)** バックエンドの場合は、デプロイメント内の新しい **environments/cinder-netapp-config.yaml** を使用してください。
- **Dell EMC Block Storage (cinder)** バックエンドの場合は、デプロイメント内の新しい **environments/cinder-dellsc-config.yaml** を使用してください。
- **Dell EqualLogic Block Storage (cinder)** バックエンドの場合は、デプロイメント内の新しい **environments/cinder-dellps-config.yaml** を使用してください。

NetApp Block Storage (cinder) バックエンドは、それぞれのバージョンに以下のリソースを使用していました。

- OpenStack Platform 10 以前: **OS::TripleO::ControllerExtraConfigPre**:
../puppet/extraconfig/pre_deploy/controller/cinder-netapp.yaml
- OpenStack Platform 11: **OS::TripleO::Services::CinderBackendNetApp**:
../puppet/services/cinder-backend-netapp.yaml

今回の変更の結果、このバックエンドには新しい

OS::TripleO::Services::CinderBackendNetApp リソースと、関連付けられたサービステンプレートを使用するようになりました。

3.5.5. NFV の設定

Red Hat OpenStack Platform 10 から Red Hat OpenStack Platform 11 にアップグレードすると、OVS パッケージもバージョン 2.5 から 2.6 に更新されます。OVS-DPDK が設定されている場合にこの移行をサポートするには、以下のガイドラインに従ってください。



注記

Red Hat OpenStack Platform 11 は OVS クライアントモードで稼働します。

『[Red Hat OpenStack Platform のアップグレード](#)』を参照してアップグレードの準備を開始します。「[コンポーザブルサービスのアップグレード](#)」に従ってオーバークラウドのデプロイする前に、以下の手順に実行してください。

1. 既存の **post-install.yaml** ファイルを、「[付録A NFV アップグレードファイルのサンプル](#)」に記載のサンプルファイルに置き換えます。
2. **network-environment.yaml** ファイルに以下の変更を適用します。
 - a. 構成に応じて **HostCpuList** を変更します。このパラメーターには、yaml ファイルで一重引用符と二重引用符の両方を必ず使用してください。

```
HostCpusList: "'3,5,7,19,21,23'"
```

- b. 構成に応じて **NeutronDpdkSocketMemory** を変更します。このパラメーターには、yaml ファイルで二重引用符のみを使用してください。

```
NeutronDpdkSocketMemory: "2048,2048"
```

- c. **NeutronVhostuserSocketDir** を以下のように変更します。

```
NeutronVhostuserSocketDir: "/var/lib/vhost_sockets"
```

3.5.6. オーバークラウドのパラメーター

アップグレードを実行するにあたっては、オーバークラウドパラメーターについての以下の情報に注意してください。

- カスタムの **ServiceNetMap** を指定してオーバークラウドをアップグレードする場合には、新規サービス向けに最新の **ServiceNetMap** が含まれるようにしてください。デフォルトのサー

ビス一覧は、`network/service_net_map.j2.yaml` ファイルの `ServiceNetMapDefaults` パラメーターで定義されます。カスタムの `ServiceNetMap` の使用方法については、『[オーバークラウドの高度なカスタマイズ](#)』の「[ネットワークの分離](#)」のセクションを参照してください。

- オーバークラウドネットワーク用の固定の仮想 IP アドレスは、新しいパラメーターと構文を使用します。『[オーバークラウドの高度なカスタマイズ](#)』ガイドの「[予測可能な仮想 IP の割り当て](#)」を参照してください。外部のロードバランシングを使用している場合には、『[オーバークラウド向けの外部のロードバランシング](#)』ガイドの「[ロードバランシングのオプションの設定](#)」も参照してください。
- `openstack overcloud deploy` コマンドの一部のオプションは非推奨になりました。これらのオプションの代わりに、同等の Heat パラメーターを使用すべきです。これらのパラメーターの対照表は、『[director のインストールと使用方法](#)』ガイドの「[CLI ツールを使用したオーバークラウドの作成](#)」を参照してください。
- 一部のコンポーザブルロールには Puppet の `hieradata` を設定する新規パラメーターが含まれています。以前 `hieradata` を使用してこれらのパラメーターを設定した場合には、オーバークラウドの更新で **Duplicate declaration** エラーが表示される可能性があります。このような場合には、コンポーザブルサービスパラメーターを使用します。たとえば、以下の設定は使用しないようにします。

```
parameter_defaults:
  controllerExtraConfig:
    heat::config::heat_config:
      DEFAULT/num_engine_workers:
        value: 1
```

その代わりに、以下の設定を使用します。

```
parameter_defaults:
  HeatWorkers: 1
```

使用可能なパラメーターについては、『[オーバークラウドのパラメーター](#)』を参照してください。

3.5.7. カスタムのコアテンプレート

Red Hat OpenStack Platform 10 のコア Heat テンプレートコレクションの変更されたバージョンを使用している場合には、Red Hat OpenStack Platform 11 バージョンのコピーにカスタマイズを再度適用する必要があります。これには、『[オーバークラウドの高度なカスタマイズ](#)』ガイドの「[カスタムのコア Heat テンプレートの使用](#)」に記載した方法と同じように `git` バージョン管理システムを使用してください。

Red Hat は、今後のリリースで Heat テンプレートコレクションの更新を提供しています。バージョン管理システムなしで変更されたテンプレートコレクションを使用すると、カスタムのコピーと `/usr/share/openstack-tripleo-heat-templates` にあるオリジナルのコピーとの間に相違が生じる可能性があります。

カスタムの Heat テンプレートコレクションを使用する代わりとして、Red Hat では『[オーバークラウドの高度なカスタマイズ](#)』ガイドの「[設定フック](#)」に記載の方法を推奨しています。

3.6. オーバークラウドのアップグレード

3.6.1. オーバークラウドノードのアップグレード

オーバークラウドのアップグレードには、追加の環境ファイル (**major-upgrade-composable-steps.yaml**) がデプロイメントに必要です。このファイルにより、**disable_upgrade_deployment: True** パラメーターでマークされているロールを除いた全ノードに完全なアップグレードが提供されます。

アンダークラウドから **major-upgrade-composable-steps.yaml** の環境ファイルを指定して **openstack overcloud deploy** コマンドを実行します。ネットワークの分離やストレージなど環境に関連するオプションとカスタムの環境ファイルをすべて指定してください。

openstack overcloud deploy コマンドで必須およびオプションのファイルを指定する例を以下に示します。

```
$ openstack overcloud deploy --templates \
  --control-scale 3 \
  --compute-scale 3 \
  -e /usr/share/openstack-tripleo-heat-templates/environments/network-
  isolation.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/net-single-
  nic-with-vlans.yaml \
  -e network_env.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/major-
  upgrade-composable-steps.yaml \
  --ntp-server pool.ntp.org
```

新しい環境ファイルの設定でオーバークラウドが更新されるまで待ちます。



重要

このステップは、コントローラーのアップグレードの際に Neutron サーバーおよび L3 エージェントを無効化します。そのため、このステップの実行中には新規ルーターの作成はできません。

各ノードで **/var/log/yum.log** ファイルをチェックして、**kernel** または **openvswitch** のパッケージのメジャー/マイナーバージョンが更新されているかどうかを確認します。更新されている場合には、『**director** のインストールと使用方法』ガイドの「**ノードの再起動**」に記載の手順に従って、各ノードを再起動します。

major-upgrade-composable-steps.yaml 環境ファイルのデプロイ中に

は、**disable_upgrade_deployment: True** パラメーターが指定されているロールの各ノードに対して **director** が特別なスクリプトを渡します。以下の項では、アンダークラウドからこのスクリプトを起動して残りのロールをアップグレードする方法について説明します。



注記

このコマンドは、非推奨となったコンポーザブルサービスを削除して、Red Hat OpenStack Platform 11 向けの新規サービスをインストールします。非推奨および新しいサービスの一覧は、「**非推奨および新規のコンポーザブルサービス**」を参照してください。

3.6.2. Object Storage ノードのアップグレード

director は **upgrade-non-controller.sh** コマンドを使用してアップグレードのスクリプトを実行し

ます。このスクリプトは、**major-upgrade-pacemaker-init.yaml** 環境ファイルから、Object Storage ノードに渡されます。このステップでは、各 Object Storage ノードを以下のコマンドでアップグレードします。

```
$ for NODE in `openstack server list -c Name -f value --name
objectstorage`; do upgrade-non-controller.sh --upgrade $NODE ; done
```

各 Object Storage ノードを個別にアップグレードすることにより、アップグレード中にサービスの稼働状態を維持します。

各 Object Storage ノードのアップグレードが完了するまで待ちます。

各ノードで **/var/log/yum.log** ファイルをチェックして、**kernel** または **openvswitch** のパッケージのメジャー/マイナーバージョンが更新されているかどうかを確認します。更新されている場合には、各ノードを再起動します。

1. 再起動する Object Storage ノードを選択します。そのノードにログインして再起動します。

```
$ sudo reboot
```

2. ノードが起動するまで待ちます。
3. ノードにログインして、ステータスを確認します。

```
$ sudo systemctl list-units "openstack-swift*"
```

4. ノードからログアウトして、次の Object Storage ノードでこのプロセスを繰り返します。



注記

コントローラーノードにログインして **pcs status** コマンドを実行し、コントローラークラスター内の全リソースがアクティブかどうかを確認します。いずれかのリソースが失敗した場合には、**pcs resource cleanup** を実行してエラーをクリーニングし、各リソースの状態を **Started** に設定します。エラーが引き続き発生する場合には、Red Hat にヘルプ/サポートをリクエストしてください。

3.6.3. コンピュートノードのアップグレード

コンピュートノードを個別にアップグレードして、OpenStack Platform 環境のインスタンスのダウンタイムがゼロになるようにします。この操作は、以下のワークフローに従って実行します。

1. アップグレードするコンピュートノードを選択します。
2. インスタンスを別のコンピュートノードに移行します。
3. 空のコンピュートノードをアップグレードします。

全コンピュートノードとその UUID を一覧表示します。

```
$ source ~/stackrc
$ openstack server list | grep "compute"
```

アップグレードするコンピュートノードを選択してから、まず最初に以下の手順に従ってそのノードのインスタンスを移行します。

1. アンダークラウドから、再起動するコンピュータノードを選択し、そのノードを無効にします。

```
$ source ~/overcloudrc
$ openstack compute service list
$ openstack compute service set [hostname] nova-compute --disable
```

2. コンピュータノード上の全インスタンスを一覧表示します。

```
$ openstack server list --host [hostname] --all-projects
```

3. 無効にしたホストから各インスタンスを移行します。以下のコマンドの1つを使用します。

- a. 選択した特定のホストにインスタンスを移行します。

```
$ openstack server migrate [instance-id] --live [target-host]--wait
```

- b. **nova-scheduler** により対象のホストが自動的に選択されるようにします。

```
$ nova live-migration [instance-id]
```



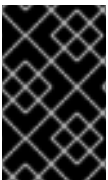
注記

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、安全に無視することができます。

4. 移行が完了するまで待ちます。
5. インスタンスがコンピュータノードから移行されたことを確認します。

```
$ openstack server list --host [hostname] --all-projects
```

6. コンピュータノードからすべてのインスタンスが移行されるまで、このステップを繰り返します。



重要

インスタンスの設定と移行の詳しい手順は、『**director のインストールと使用方法**』の「[オーバークラウドのコンピュータノードからの仮想マシンの移行](#)」を参照してください。

director は **upgrade-non-controller.sh** コマンドを使用してアップグレードのスクリプトを実行します。このスクリプトは、**major-upgrade-composable-steps.yaml** 環境ファイルから、コントローラー以外の各ノードに渡されます。以下のコマンドを実行して、各コンピュータノードをアップグレードします。

```
$ source ~/stackrc
$ upgrade-non-controller.sh --upgrade [NODE]
```

[**NODE**] は選択したコンピュータノードの UUID または名前に置き換えます。コンピュータノードのアップグレードが完了するまで待ちます。

アップグレードしたコンピュータード上の `/var/log/yum.log` ファイルをチェックして、以下のいずれかのパッケージのメジャー/マイナーバージョンが更新されているかを確認します。

- **kernel**
- **openvswitch**
- **ceph-osd** (ハイパーコンバージド環境)

更新されている場合には、ノードの再起動を実行します。

1. コンピュータードにログインして、再起動します。

```
$ sudo reboot
```

2. ノードが起動するまで待ちます。

3. 再度コンピュータードを有効化します。

```
$ source ~/overcloudrc
$ openstack compute service set [hostname] nova-compute --enable
```

4. コンピュータードが有効化されているかどうかを確認します。

```
$ openstack compute service list
```

全ノードのアップグレードと再起動が完了するまで、各ノードでこのプロセスを個別に繰り返します。

全コンピュータードをアップグレードした後は、**stackrc** アクセスの情報を元に戻します。

```
$ source ~/stackrc
```



注記

コントローラードにログインして **pcs status** コマンドを実行し、コントローラードクラスター内の全リソースがアクティブかどうかを確認します。いずれかのリソースが失敗した場合には、**pcs resource cleanup** を実行してエラーをクリーニングし、各リソースの状態を **Started** に設定します。エラーが引き続き発生する場合には、Red Hat にヘルプ / サポートをリクエストしてください。

3.6.4. アップグレードの最終処理

`director` には、アップグレードの最終処理を最後まで実行して、オーバークラウドスタックが現在の Heat テンプレートコレクションと確実に同期されるようにする必要があります。そのためには、**openstack overcloud deploy** コマンドで環境ファイル (`major-upgrade-converge.yaml`) を指定する必要があります。

 重要

Red Hat OpenStack Platform の環境が、以前のバージョン(例: Red Hat Ceph Storage 1.3) の外部の Ceph Storage Cluster と統合されている場合には、後方互換性を有効にする必要があります。そのためには、環境ファイル (例: `/home/stack/templates/ceph-backwards-compatibility.yaml`) を作成して、以下の内容を記載します。

```
parameter_defaults:
  RbdDefaultFeatures: 1
```

このファイルの作成が完了したら、次のステップで `openstack overcloud deploy` を実行する際に指定してください。

アンダークラウドから `openstack overcloud deploy` を実行して、`major-upgrade-converge.yaml` 環境ファイルを指定します。Ceph の後方互換性 (該当する場合)、ネットワークの分離、ストレージなど、お使いの環境に関連するオプションおよびカスタム環境ファイルもすべて必ず追加するようにしてください。

`major-upgrade-converge.yaml` ファイルを追加した `openstack overcloud deploy` コマンドの一例を以下に示します。

```
$ openstack overcloud deploy --templates \
  --control-scale 3 \
  --compute-scale 3 \
  -e /usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/net-single-nic-with-vlans.yaml \
  -e network_env.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/major-upgrade-converge.yaml \
  --ntp-server pool.ntp.org
```

新しい環境ファイルの設定でオーバークラウドが更新されるまで待ちます。

 注記

コントローラーノードにログインして `pcs status` コマンドを実行し、コントローラークラスター内の全リソースがアクティブかどうかを確認します。いずれかのリソースが失敗した場合には、`pcs resource cleanup` を実行してエラーをクリーニングし、各リソースの状態を `Started` に設定します。エラーが引き続き発生する場合には、Red Hat にヘルプ/サポートをリクエストしてください。

3.7. オーバークラウドのアップグレード後の注意事項

オーバークラウドを Red Hat OpenStack Platform 11 にアップグレードした後は以下の点に注意してください。

- 必要な場合にはオーバークラウドノードで作成された設定ファイルを確認します。アップグレードされたパッケージで、Red Hat OpenStack Platform 11 バージョンのサービスに適した `.rpmnew` ファイルがインストールされている可能性があります。
- Red Hat OpenStack Platform 11 では、OpenStack Block Storage (cinder) API は新しい

sizelimit フィルターを使用します。ただし、以下に記載する OpenStack Platform 9、OpenStack Platform 10、OpenStack Platform 11 の順でアップグレードを行うアップグレードパスでは、この新規フィルターが更新されない可能性があります。コントローラーまたは Cinder API ノードで以下のコマンドを実行してフィルターを確認します。

```
$ grep filter:sizelimit /etc/cinder/api-paste.ini -A1
```

フィルターは以下のように表示されるはずですが、

```
[filter:sizelimit]
paste.filter_factory =
oslo_middleware.sizelimit:RequestBodySizeLimiter.factory
```

このように表示されない場合は、各コントローラーまたは Cinder API ノードの **/etc/cinder/api-paste.ini** ファイルの値を置き換えて、**httpd** サービスを再起動します。

```
$ sudo sed -i
s/cinder.api.middleware.sizelimit/oslo_middleware.sizelimit/
/etc/cinder/api-paste.ini
$ sudo systemctl restart httpd
```

- コンピュートノードが **neutron-openvswitch-agent** の問題をレポートする可能性があります。これが発生した場合には、各コンピュートノードにログインして、このサービスを再起動します。以下のようなコマンドを実行します。

```
$ sudo systemctl restart neutron-openvswitch-agent
```

- 状況によっては、コントローラーノードの再起動後に IPv6 環境で **corosync** サービスの起動に失敗する可能性があります。これは、コントローラーノードが静的な IPv6 アドレスを設定する前に Corosync が起動してしまうことが原因です。このような場合は、コントローラーノードで Corosync を手動で再起動してください。

```
$ sudo systemctl restart corosync
```

- コントローラーノードにフェンシングを設定している場合には、アップグレードプロセスによってその設定が無効になる場合があります。アップグレードプロセスの完了時には、コントローラーノードの 1 つで以下のコマンドを実行してフェンシングを再度有効にします。

```
$ sudo pcs property set stonith-enabled=true
```


第4章 DIRECTOR ベースのアップグレードのトラブルシューティング

本項では、アンダークラウドとオーバークラウドの両シナリオで問題が発生した場合のトラブルシューティングのアドバイスを記載します。

4.1. アンダークラウドのアップグレード

アンダークラウドのアップグレードコマンド (**openstack undercloud upgrade**) が失敗した場合には、以下のアドバイスに従って、アップグレードの進捗の妨げとなっている問題を特定してください。

- **openstack undercloud upgrade** コマンドは、実行中に進捗状況のログを出力し、**.instack/install-undercloud.log** に保存します。アップグレードプロセスのいずれかの時点でエラーが発生した場合には、コマンドはその時点で停止します。この情報は、アップグレードの進行を妨げている問題の特定に使用してください。
- **openstack undercloud upgrade** コマンドは、Puppet を実行してアンダークラウドサービスを設定します。これにより、以下のディレクトリーで便利な Puppet のレポートが生成されます。
 - **/var/lib/puppet/state/last_run_report.yaml**: 最後の Puppet レポートは、アンダークラウド向けに生成されます。このファイルは、問題のある Puppet のアクションの原因を表示します。
 - **/var/lib/puppet/state/last_run_summary.yaml**: **last_run_report.yaml** ファイルのサマリー
 - **/var/lib/puppet/reports**: アンダークラウドの全 Puppet レポート
この情報を使用して、アップグレードプロセスを妨げている問題を特定します。
- エラーが発生しているサービスがあるかどうかを確認します。

```
$ sudo systemctl -t service
```

エラーが発生しているサービスがある場合は、対応するログを確認します。たとえば、**openstack-ironic-api** でエラーが発生している場合には、以下のコマンドを使用してこのサービスのログを確認します。

```
$ sudo journalctl -xe -u openstack-ironic-api  
$ sudo tail -n 50 /var/log/ironic/ironic-api.log
```

アンダークラウドのアップグレードを妨げていた問題を修正した後に、アップグレードのコマンドを再度実行します。

```
$ openstack undercloud upgrade
```

アップグレードのコマンドをもう 1 度開始して、アンダークラウドを設定します。

4.2. オーバークラウドのアップグレード

オーバークラウドのアップグレードプロセスで障害が発生した場合には、以下のアドバイスに従ってアップグレードプロセスを妨げている問題を特定します。

- Heat スタックの一覧を確認して、**UPDATE_FAILED** のステータスがあるスタックを特定します。以下のコマンドで、失敗したスタックを特定します。

```
$ openstack stack failures list overcloud
```

エラーの発生したスタックとそのスタックのテンプレートを表示して、スタックが失敗した原因を究明します。

```
$ openstack stack show overcloud-Controller-qyoy54dyhr11-1-
gtwy5bgta3np
$ openstack stack template show overcloud-Controller-qyoy54dyhr11-1-
gtwy5bgta3np
```

- 全コントローラーノード上で Pacemaker が正しく実行されていることを確認します。必要な場合は、コントローラーノードにログインして、コントローラークラスターを再起動します。

```
$ sudo pcs cluster start
```

- 設定のログファイルでエラーがあるかどうかを確認します。各ノードの `/var/run/heat-config/` ディレクトリにはこれらのログが含まれています。これらのファイル名は日付順にリストされ、標準出力 (`*-stdout.log`) およびエラー出力 (`*-stderr.log`) に分けられています。



注記

director はアップグレードプロセスの前に一式の検証チェックを実行し、オーバークラウドが正常な状態であることを確認します。アップグレードが失敗して、再試行する場合には、これらの検証チェックを無効にする必要がある場合があります。チェックを無効にするには、オーバークラウドに含まれている環境ファイルの `parameter_defaults` セクションに `SkipUpgradeConfigTags: [validation]` を一時的に追加します。

オーバークラウドのアップグレードを妨げていた問題を修正した後は、**IN_PROGRESS** ステータスのリソースがないことを確認します。

```
$ openstack stack resource list overcloud -n5 --filter
status='*IN_PROGRESS'
```

いずれかのリソースが **IN_PROGRESS** のステータスの場合には、そのプロセスが完了または失敗するまで待ちます。

アップグレードを試みて失敗したステップで `openstack overcloud deploy` コマンドを再度実行します。アップグレードプロセスの最初の `openstack overcloud deploy` コマンドの例を以下に示します。このコマンドでは `major-upgrade-composable-steps.yaml` を指定しています。

```
$ openstack overcloud deploy --templates \
--control-scale 3 \
--compute-scale 3 \
-e /usr/share/openstack-tripleo-heat-templates/environments/network-
isolation.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/net-single-
nic-with-vlans.yaml \
-e network_env.yaml \
```

```
-e /usr/share/openstack-tripleo-heat-templates/environments/major-  
upgrade-composable-steps.yaml \  
--ntp-server pool.ntp.org
```

openstack overcloud deploy は、オーバークラウドのスタックの更新を再試行します。

パート II. DIRECTOR 以外の環境

第5章 RED HAT OPENSTACK PLATFORM 環境の手動アップグレード

director を使用しない環境の手動のアップグレード手順は、**openstack-nova-placement-api** パッケージ内の Apache 設定の既知の問題 ([BZ#1434944](#)) が原因で、現在利用できない状態です。この問題のため、コンピュートを手動でアップグレードすると、コンピュートノードが Placement API に対して報告を試みる際に、以下のようなエラーメッセージが表示されます。

```
You don't have permission to access /resource_providers on this server.
```

director ベースのデプロイメントは、この問題による影響は受けません。

パート III. 付録

付録A NFV アップグレードファイルのサンプル

```

heat_template_version: 2014-10-16

description: >
  Example extra config for post-deployment

parameters:
  servers:
    type: json
  HostCpusList:
    description: >
      List of logical cores to be used by ovs-dpdk processes (dpdk-
lcore-mask)
    type: string
  NeutronDpdkCoreList:
    description: >
      List of logical cores for PMD threads. (pmd-cpu-mask)
    type: string
resources:

  ExtraDeployments:
    type: OS::Heat::StructuredDeployments
    properties:
      servers: {get_param: servers}
      config: {get_resource: ExtraConfig}
      actions: ['CREATE', 'UPDATE']

  ExtraConfig:
    type: OS::Heat::SoftwareConfig
    properties:
      group: script
      config:
        str_replace:
          template: |
            #!/bin/bash
            set -x
            function tuned_service_dependency() {
              tuned_service=/usr/lib/systemd/system/tuned.service
              grep -q "network.target" $tuned_service
              if [ "$?" -eq 0 ]; then
                sed -i '/After=.*s/network.target//g' $tuned_service
              fi
              grep -q "Before=.*network.target" $tuned_service
              if [ ! "$?" -eq 0 ]; then
                grep -q "Before=.*" $tuned_service
                if [ "$?" -eq 0 ]; then
                  sed -i 's/^\(Before=.*\)\/\1 network.target
openvswitch.service/g' $tuned_service
                else
                  sed -i '/After/i Before=network.target
openvswitch.service' $tuned_service
                fi
              fi
            }

```

```

function ovs_permission_fix() {
    ovs_service_path="/usr/lib/systemd/system/ovs-
vswitchd.service"
    grep -q "RuntimeDirectoryMode=.*" $ovs_service_path
    if [ "$?" -eq 0 ]; then
        sed -i
's/RuntimeDirectoryMode=.*RuntimeDirectoryMode=0775/' $ovs_service_path
    else
        echo "RuntimeDirectoryMode=0775" >>
$ovs_service_path
    fi
    grep -Fxq "Group=qemu" $ovs_service_path
    if [ ! "$?" -eq 0 ]; then
        echo "Group=qemu" >> $ovs_service_path
    fi
    grep -Fxq "UMask=0002" $ovs_service_path
    if [ ! "$?" -eq 0 ]; then
        echo "UMask=0002" >> $ovs_service_path
    fi
    ovs_ctl_path="/usr/share/openvswitch/scripts/ovs-ctl"
    grep -q "umask 0002 \&\& start_daemon
\"$OVS_VSWITCHD_PRIORITY\" \"$ovs_ctl_path
    if [ ! "$?" -eq 0 ]; then
        sed -i 's/start_daemon
\"$OVS_VSWITCHD_PRIORITY.*umask 0002 \&\& start_daemon
\"$OVS_VSWITCHD_PRIORITY\" \"$OVS_VSWITCHD_WRAPPER\" \"$@\"/' $ovs_ctl_path
    fi
}

function set_ovs_socket_dir {
    # TODO: Hardcoding it here as this directory is fixed,
because it requires SELinux permissions
    NEUTRON_VHOSTUSER_SOCKET_DIR="/var/lib/vhost_sockets"
    mkdir -p $NEUTRON_VHOSTUSER_SOCKET_DIR
    chown -R qemu:qemu $NEUTRON_VHOSTUSER_SOCKET_DIR
    restorecon $NEUTRON_VHOSTUSER_SOCKET_DIR
}

get_mask()
{
    local list=$1
    local mask=0
    declare -a bm
    max_idx=0
    for core in $(echo $list | sed 's/,/ /g')
    do
        index=$((core/32))
        bm[$index]=0
        if [ $max_idx -lt $index ]; then
            max_idx=$((index))
        fi
    done
    for ((i=$max_idx;i>=0;i--));
    do
        bm[$i]=0
    done
}

```



```
    for core in $(echo $list | sed 's/,/ /g')
    do
        index=$((score/32))
        temp=$((1<<$((score % 32)))
        bm[$index]=$({bm[$index]} | $temp)
    done
    printf -v mask "%x" "${bm[$max_idx]}"
    for ((i=$max_idx-1;i>=0;i--));
    do
        printf -v hex "%08x" "${bm[$i]}"
        mask+=$hex
    done
    printf "%s" "$mask"
}

    if hiera -c /etc/puppet/hiera.yaml service_names | grep -q
neutron_ovs_dpdk_agent; then
        pmd_cpu_mask=$( get_mask $PMD_CORES )
        host_cpu_mask=$( get_mask $LCORE_LIST )
        ovs-vsctl --no-wait set Open_vSwitch .
other_config:pmd-cpu-mask=$pmd_cpu_mask
        ovs-vsctl --no-wait set Open_vSwitch .
other_config:dpdk-lcore-mask=$host_cpu_mask
        tuned_service_dependency
        ovs_permission_fix
        set_ovs_socket_dir
    fi
params:
    $LCORE_LIST: {get_param: HostCpusList}
    $PMD_CORES: {get_param: NeutronDpdkCoreList}
```