



Red Hat OpenStack Platform 12

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 12 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 11 (Ocata) から 12 (Pike) 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. アップグレードの目標	3
1.2. アップグレードパス	3
1.3. リポジトリ	4
第2章 OPENSTACK PLATFORM のアップグレードの準備	6
2.1. サポートステートメント	6
2.2. アップグレードに関する一般的なアドバイス	6
2.3. アップグレード前のアンダークラウドの検証	7
2.4. アップグレード前のオーバークラウドの検証	8
2.5. アンダークラウドのバックアップ	10
2.6. 現行バージョンのアンダークラウドパッケージの更新	11
2.7. 現行バージョンのオーバークラウドイメージの更新	12
2.8. 現行バージョンのオーバークラウドパッケージの更新	13
第3章 アンダークラウドのアップグレード	15
3.1. アンダークラウドノードのアップグレード	15
3.2. オーバークラウドイメージのアップグレード	16
3.3. 以前のテンプレートバージョンの比較	17
第4章 オーバークラウドのアップグレードの準備	19
4.1. オーバークラウドの登録情報の準備	19
4.2. コンテナ化されたサービスの準備	19
4.3. 新規コンポーザブルサービスの作成	20
4.4. コンポーザブルネットワークの準備	21
4.5. 非推奨パラメーターの準備	22
4.6. CEPH STORAGE ノードのアップグレードの準備	23
4.7. HYPER-CONVERGED INFRASTRUCTURE (HCI) のアップグレードの準備	24
4.8. SSL/TLS を介してアンダークラウドのパブリック API にアクセスするための準備	26
4.9. 事前にプロビジョニングされたノードのアップグレードの準備	27
4.10. NFV を設定したオーバークラウドの準備	28
4.11. オーバークラウドのアップグレードの一般的な考慮事項	29
第5章 オーバークラウドのアップグレード	31
5.1. オーバークラウドノードのアップグレード	31
5.2. OBJECT STORAGE ノードのアップグレード	32
5.3. コンピュートノードのアップグレード	33
5.4. アップグレードの最終処理	35
第6章 アップグレード後のステップの実行	37
6.1. 新規オーバークラウドノードへのアンダークラウド CA の追加	37
6.2. オーバークラウドのアップグレード後の一般的な考慮事項	38
第7章 OPENSTACK PLATFORM の最新状態の維持	39
7.1. 更新前のアンダークラウドの検証	39
7.2. 更新前のオーバークラウドの検証	40
7.3. アンダークラウドの最新状態の維持	42
7.4. オーバークラウドイメージの最新状態の維持	43
7.5. オーバークラウドの最新状態の維持	44

第1章 はじめに

本書には、Red Hat OpenStack Platform 環境を最新のメジャーバージョンにアップグレードし、そのバージョンのマイナーリリースで最新状態に維持するのに役立つワークフローを記載しています。

1.1. アップグレードの目標

Red Hat OpenStack Platform は、現在の環境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提供しています。本ガイドでは、お使いの環境を最新の **Red Hat OpenStack Platform 12 (Pike)** リリースにアップグレードおよび更新することを目的としています。

このアップグレードでは、以下もアップグレードされます。

- **オペレーティングシステム:** Red Hat Enterprise Linux 7.4
- **ネットワーク:** Open vSwitch 2.6
- **Ceph Storage:** このプロセスで、Red Hat Ceph Storage 2 は最新バージョンにアップグレードされ、**ceph-ansible** デプロイメントに切り替えられます。



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. アップグレードパス

Red Hat OpenStack Platform 環境のアップグレードパスを以下に示します。

表1.1 OpenStack Platform のアップグレードパス

タスク		バージョン	回数
1	現行バージョンのアンダークラウドとオーバークラウドをバックアップします。	Red Hat OpenStack Platform 11	1 回
2	現行バージョンのアンダークラウドとオーバークラウドを最新のマイナーリリースに更新します。	Red Hat OpenStack Platform 11	1 回
3	現在のアンダークラウドを最新のメジャーリリースにアップグレードします。	Red Hat OpenStack Platform 11 および 12	1 回
4	オーバークラウドの準備をします。これには、関連するカスタム設定の更新などが含まれます。	Red Hat OpenStack Platform 11 および 12	1 回

タスク	バージョン	回数
5	現行バージョンのオーバークラウドを最新のメジャーリリースにアップグレードします。	Red Hat OpenStack Platform 11 および 12 1 回
6	アンダークラウドとオーバークラウドを最新のマイナーリリースに定期的に更新します。	Red Hat OpenStack Platform 12 継続的

1.3. リポジトリ

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	x86_64 システム用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 Enterprise Linux OpenStack Platform 12 for RHEL 7 (RPMS)	rhel-7-server-openstack-12-rpms	Red Hat OpenStack Platform のコアリポジトリ。Red Hat OpenStack Platform director のパッケージも含まれます。
Red Hat Ceph Storage OSD 2 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。

名前	リポジトリ	説明
Red Hat Ceph Storage MON 2 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 Server (RPMs)	rhel-7-server-rhceph-2-tools-rpms	Ceph Storage クラスターと通信するためのノード用のツールを提供します。このリポジトリは、Ceph Storage クラスターを使用するオーバークラウドをデプロイする際に、全ノードに有効化する必要があります。



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。

第2章 OPENSTACK PLATFORM のアップグレードの準備

以下の手順では、OpenStack Platform 環境を全面的に更新するための準備を行います。

2.1. 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 を使用する場合は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。
- 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 12 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2.1 Red Hat OpenStack Platform director 12 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リング
Red Hat OpenStack Platform 12	Red Hat OpenStack Platform 12 および 11	Red Hat OpenStack Platform 12 および 11	Red Hat OpenStack Platform 12 および 11

2.2. アップグレードに関する一般的なアドバイス

アップグレードに役立つアドバイスを以下に示します。

- 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。

2.3. アップグレード前のアンダークラウドの検証

Red Hat OpenStack Platform 11 のアンダークラウドをアップグレードする前に機能を確認する手順を以下に示します。

手順

1. アンダークラウドのアクセス情報を読み込みます。

```
$ source ~/stackrc
```

2. エラーが発生している Systemd サービスがあるかどうかを確認します。

```
(undercloud) $ sudo systemctl list-units --state=failed 'openstack*'
'neutron*' 'httpd' 'docker'
```

3. アンダークラウドの空き領域を確認します。

```
(undercloud) $ df -h
```

4. アンダークラウドでクロックが同期されていることを確認します。

```
(undercloud) $ sudo ntpstat
```

5. アンダークラウドのネットワークサービスを確認します。

```
(undercloud) $ openstack network agent list
```

全エージェントが **Alive** で、それらの状態が **UP** である必要があります。

6. アンダークラウドの Compute サービスを確認します。

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

全エージェントのステータスが **enabled** で、状態が **up** である必要があります。

7. アンダークラウドのボリュームサービスを確認します。

```
(undercloud) $ openstack volume service list
```

全エージェントのステータスが **enabled** で、状態が **up** である必要があります。

関連情報

- OpenStack Orchestration (heat) のデータベースで削除済みとマークされている stack のエントリを完全削除する方法は <https://access.redhat.com/solutions/2215131> のソリューションに記載されています。

2.4. アップグレード前のオーバークラウドの検証

Red Hat OpenStack Platform 11 のオーバークラウドをアップグレードする前に機能を確認する手順を以下に示します。

手順

1. アンダークラウドのアクセス情報を読み込みます。

```
$ source ~/stackrc
```

2. ベアメタルノードのステータスを確認します。

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

全ノードの電源状態が有効で (**on**)、かつメンテナンスモードが **false** である必要があります。

3. エラーが発生している Systemd サービスがあるかどうかを確認します。

```
(undercloud) $ for NODE in $(openstack server list -f value -c Networks | cut -d= -f2); do echo "=== $NODE ===" ; ssh heat-admin@$NODE "sudo systemctl list-units --state=failed 'openstack*' 'neutron*' 'httpd' 'docker' 'ceph*'" ; done
```

4. 全サービスへの HAProxy 接続をチェックします。コントロールプレーンの仮想 IP アドレスと **haproxy.stats** サービスの認証情報を取得します。

```
(undercloud) $ NODE=$(openstack server list --name controller-0 -f value -c Networks | cut -d= -f2); ssh heat-admin@$NODE sudo 'grep "listen haproxy.stats" -A 6 /etc/haproxy/haproxy.cfg'
```

以下の cURL 要求でそれらの情報を使用します。

```
(undercloud) $ curl -s -u admin:<PASSWORD> "http://<IP ADDRESS>:1993/;csv" | egrep -vi "(frontend|backend)" | awk -F',' '{ print $1" "$2" "$18 }'
```

<PASSWORD> と **<IP ADDRESS>** は、**haproxy.stats** サービスからのそれぞれの情報に置き換えます。その結果表示される一覧には、各ノード上の OpenStack Platform サービスとそれらの接続ステータスが表示されます。

5. オーバークラウドデータベースのレプリケーションの正常性をチェックします。

```
(undercloud) $ for NODE in $(openstack server list --name controller -f value -c Networks | cut -d= -f2); do echo "=== $NODE ===" ; ssh heat-admin@$NODE "sudo clustercheck" ; done
```

6. RabbitMQ クラスターの正常性を確認します。

```
(undercloud) $ for NODE in $(openstack server list --name controller
-f value -c Networks | cut -d= -f2); do echo "=== $NODE ===" ; ssh
heat-admin@$NODE "sudo rabbitmqctl node_health_check" ; done
```

7. Pacemaker リソースの正常性を確認します。

```
(undercloud) $ NODE=$(openstack server list --name controller-0 -f
value -c Networks | cut -d= -f2); ssh heat-admin@$NODE "sudo pcs
status"
```

以下の点を確認します。

- 全クラスターノードが **online** であること。
- いずれのクラスターノード上でも **stopped** のリソースがないこと。
- pacemaker で **failed** のアクションがないこと。

8. 各オーバークラウドノードでディスク領域を確認します。

```
(undercloud) $ for NODE in $(openstack server list -f value -c
Networks | cut -d= -f2); do echo "=== $NODE ===" ; ssh heat-
admin@$NODE "sudo df -h --output=source,fstype,avail -x overlay -x
tmpfs -x devtmpfs" ; done
```

9. オーバークラウドの Ceph Storage クラスターの正常性を確認します。以下のコマンドを使用すると、コントローラーノード上で **ceph** ツールが実行されて、クラスターをチェックします。

```
(undercloud) $ NODE=$(openstack server list --name controller-0 -f
value -c Networks | cut -d= -f2); ssh heat-admin@$NODE "sudo ceph -
s"
```

10. Ceph Storage OSD に空き領域があるかどうかを確認します。以下のコマンドを使用すると、コントローラーノード上で **ceph** ツールが実行され、空き領域をチェックします。

```
(undercloud) $ NODE=$(openstack server list --name controller-0 -f
value -c Networks | cut -d= -f2); ssh heat-admin@$NODE "sudo ceph
df"
```

11. オーバークラウドノードでクロックが同期されていることを確認します。

```
(undercloud) $ for NODE in $(openstack server list -f value -c
Networks | cut -d= -f2); do echo "=== $NODE ===" ; ssh heat-
admin@$NODE "sudo ntpstat" ; done
```

12. オーバークラウドのアクセス情報を読み込みます。

```
(undercloud) $ source ~/overcloudrc
```

13. オーバークラウドのネットワークサービスを確認します。

```
(overcloud) $ openstack network agent list
```

全エージェントが **Alive** で、それらの状態が **UP** である必要があります。

14. オーバークラウドの Compute サービスを確認します。

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

全エージェントのステータスが **enabled** で、状態が **up** である必要があります。

15. オーバークラウドのボリュームサービスを確認します。

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

全エージェントのステータスが **enabled** で、状態が **up** である必要があります。

関連情報

- [「How can I verify my OpenStack environment is deployed with Red Hat recommended configurations?」](#)の記事を参照してください。この記事には、Red Hat OpenStack Platform 環境をチェックして、Red Hat の推奨値に合わせて調整する方法が記載されています。
- [「Database Size Management for Red Hat Enterprise Linux OpenStack Platform」](#)の記事を参照して、オーバークラウド上の OpenStack サービスの未使用のデータベースレコードをチェックしてクリーニングします。

2.5. アンダークラウドのバックアップ

完全なアンダークラウドのバックアップには、以下のデータベースおよびファイルが含まれます。

- アンダークラウドノード上の MariaDB データベース
- (データベースを正確に復元できるように) アンダークラウド上の MariaDB 設定ファイル
- /srv/node の swift データすべて
- stack ユーザーのホームディレクトリー (/home/stack) にあるデータすべて
- アンダークラウドの SSL 証明書
 - /etc/pki/ca-trust/source/anchors/ca.crt.pem
 - /etc/pki/instack-certs/undercloud.pem



注記

バックアッププロセスを実行する前に、利用可能なディスク容量が十分にあることを確認します。tarball は、最低でも 3.5 GB になることが予想されますが、それ以上になる可能性が高くなります。

手順

1. アンダークラウドに **root** ユーザーとしてログインします。
2. データベースをバックアップします。

```
# mysqldump --opt --all-databases > /root/undercloud-all-databases.sql
```

- データベースのバックアップと設定ファイルをアーカイブします。

```
# tar --xattrs -czf undercloud-backup-`date +%F`.tar.gz
/root/undercloud-all-databases.sql /etc/my.cnf.d/server.cnf
/srv/node /home/stack /etc/pki/instack-certs/undercloud.pem
/etc/pki/ca-trust/source/anchors/ca.crt.pem
```

このコマンドにより、**undercloud-backup-[timestamp].tar.gz** という名前のファイルが作成されます。

関連情報

- アンダークラウドのバックアップを復元する必要がある場合には、『**director のアンダークラウドのバックアップと復元**』ガイドの「[復元](#)」の章を参照してください。

2.6. 現行バージョンのアンダークラウドパッケージの更新

director では、アンダークラウドノード上の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。これにより、OpenStack Platform 環境の現在のバージョン内のマイナー更新を実行することができます。これは、**Red Hat OpenStack Platform 11** 内でのマイナー更新です。

前提条件

- アンダークラウドのバックアップを実行済みであること

手順

- director に **stack** ユーザーとしてログインします。
- 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 python-tripleoclient
```

- 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。

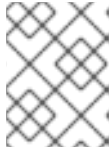
```
$ openstack undercloud upgrade
```

- ノードを再起動します。

```
$ sudo reboot
```

- ノードが起動するまで待ちます。
- 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。

7. 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
```

2.7. 現行バージョンの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。このプロセスにより、**Red Hat OpenStack Platform 11** 内のアンダークラウドでそれらのイメージが更新されます。

前提条件

- 現行バージョンのアンダークラウドの最新のマイナーリリースに更新済みであること

手順

1. **yum** ログをチェックして、新規イメージのアーカイブが利用可能かどうかを確認します。

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

2. 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/*
```

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-11.0.tar /usr/share/rhosp-director-images/ironic-python-agent-latest-11.0.tar; do tar -xvf $i; done
```

4. 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 /home/stack/images/
$ openstack overcloud node configure $(openstack baremetal node list -c UUID -f csv --quote none | sed "1d" | paste -s -d " ")
```

5. 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。

2.8. 現行バージョンのオーバークラウドパッケージの更新

director では、全オーバークラウドノード上の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。これにより、OpenStack Platform 環境の現在のバージョン内のマイナー更新を実行することができます。これは、**Red Hat OpenStack Platform 11** 内でのマイナー更新です。

前提条件

- 現行バージョンのアンダークラウドの最新のマイナーリリースに更新済みであること
- オーバークラウドのバックアップを実行済みであること

手順

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
```

-i のオプションを指定すると、各ノードは対話モードで更新されます。更新プロセスによりノードの更新が完了すると、スクリプトにより、確認のためのブレイクポイントが提供されます。**-i** オプションを使用しなかった場合には、最初のブレイクポイントで更新が一時停止されたままとなるため、**-i** オプションの指定は必須です。



注記

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、一連のブレイ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、ブレイ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イ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イクポイントを通過したり、複数のノードで一度にブレイ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イ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 を更新した場合には、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 アンダークラウドのアップグレード

このプロセスにより、アンダークラウドと、そのオーバークラウドのイメージが **Red Hat OpenStack Platform 12** にアップグレードされます。

3.1. アンダークラウドノードのアップグレード

オーバークラウドをアップグレードする前にアンダークラウドをアップグレードする必要があります。この手順では、アンダークラウドのツールセットとコア Heat テンプレートコレクションをアップグレードします。

このプロセスにより、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。

前提条件

- アップグレードのサポートステートメントを確認済みであること
- アンダークラウドのバージョンを最新のマイナーバージョンに更新済みであること

手順

1. director に **stack** ユーザーとしてログインします。
2. 現行バージョンの OpenStack Platform リポジトリを無効にします。

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

3. 新しい OpenStack Platform リポジトリを有効にします。

```
$ sudo subscription-manager repos --enable=rhel-7-server-openstack-12-rpms
```

4. **yum** コマンドを実行して、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 python-tripleoclient
```

5. **/home/stack/undercloud.conf** ファイルを編集して、**enabled_drivers** パラメーターに **pxe_ssh** ドライバーが含まれていないことを確認します。Virtual Baseboard Management Controller (VBMC) が推奨されるようになったため、このドライバーは非推奨となり、Red Hat OpenStack Platform から削除されました。**pxe_ssh** ノードを VBMC に切り替えるための説明は、『**director のインストールと使用方法**』ガイドの「[Virtual Baseboard Management Controller \(VBMC\)](#)」の項を参照してください。

6. 以下のコマンドを実行して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。

7. ノードを再起動します。

```
$ sudo reboot
```

8. ノードが起動するまで待ちます。

9. 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。

10. 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
```

3.2. オーバークラウドイメージのアップグレード

現行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。新しいイメージにより、director は最新バージョンの OpenStack Platform ソフトウェアを使用してノードのイントロスペクションとプロビジョニングを行うことができますようになります。

前提条件

- アンダークラウドが最新バージョンにアップグレードされていること

手順

1. **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/*
```

2. アーカイブを展開します。

```
$ cd ~/images
$ for i in /usr/share/rhosp-director-images/overcloud-full-latest-12.0.tar /usr/share/rhosp-director-images/ironic-python-agent-latest-12.0.tar; do tar -xvf $i; done
$ cd ~
```

3. director に最新のイメージをインポートします。

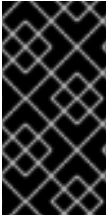
```
$ openstack overcloud image upload --update-existing --image-path /home/stack/images/
```

4. 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)
```

5. 新規イメージが存在することを確認します。

```
$ openstack image list
$ ls -l /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、その Heat テンプレートバージョンに対応していることを確認してください。たとえば、OpenStack Platform 12 の Heat テンプレートには、OpenStack Platform 12 のイメージのみを使用してください。

3.3. 以前のテンプレートバージョンの比較

アップグレードプロセスにより、最新のオーバークラウドバージョンに対応したコア Heat テンプレートの新しいセットがインストールされます。Red Hat OpenStack Platform のリポジトリには、**openstack-tripleo-heat-templates-compat** パッケージ内のコアテンプレートコレクションの以前のバージョンが維持されています。以下の手順では、オーバークラウドのアップグレードに影響する可能性のある変更点を確認することができるように、それらのバージョンを比較する方法について説明します。

手順

1. **openstack-tripleo-heat-templates-compat** パッケージをインストールします。

```
$ sudo yum install openstack-tripleo-heat-templates-compat
```

これにより、Heat テンプレートコレクションの **compat** ディレクトリー (**/usr/share/openstack-tripleo-heat-templates/compat**) に以前のテンプレートがインストールされ、以前のバージョン (**ocata**) から命名された **compat** へのリンクが作成されます。これらのテンプレートは、アップグレードされた **director** との後方互換性があるので、最新バージョンの **director** を使用して以前のバージョンのオーバークラウドをインストールすることができます。

2. コア Heat テンプレートの一時的なコピーを作成します。

```
$ cp -a /usr/share/openstack-tripleo-heat-templates /tmp/osp12
```

3. 以前のバージョンをそれ独自のディレクトリーに移動します。

```
$ mv /tmp/osp12/compat /tmp/osp11
```

4. 両ディレクトリーのコンテンツに対して **diff** を実行します。

```
$ diff -urN /tmp/osp11 /tmp/osp12
```

このコマンドにより、バージョン間におけるコアテンプレートの変更が表示されます。この内容を確認すると、オーバークラウドのアップグレード中にどのような動作が行われるかがわかります。

第4章 オーバークラウドのアップグレードの準備

以下の手順では、アップグレードのプロセスに向けたオーバークラウドの準備を行います。

前提条件

- アンダークラウドが最新バージョンにアップグレードされていること

4.1. オーバークラウドの登録情報の準備

オーバークラウドに最新のサブスクリプション情報を提供して、アップグレードプロセスの実行中にオーバークラウドが最新のパッケージを使用するようにする必要があります。

前提条件

- 最新の OpenStack Platform リポジトリが含まれたサブスクリプション
- 登録のためのアクティベーションキーを使用する場合には、新しい OpenStack Platform リポジトリを含む新規アクティベーションキーを作成すること

手順

1. 登録情報が記載された環境ファイルを編集します。以下に例を示します。

```
$ vi ~/templates/rhel-registration/environment-rhel-  
registration.yaml
```

2. 以下のパラメーター値を編集します。

rhel_reg_repos

Red Hat OpenStack Platform 12 の新しいリポジトリが含まれるように更新します。

rhel_reg_activation_key

Red Hat OpenStack Platform 12 のリポジトリにアクセスするためのアクティベーションキーを更新します。

rhel_reg_sat_repo

Red Hat Satellite 6 のより新しいバージョンを使用する場合には、Satellite 6 の管理ツールが含まれているリポジトリを更新します。

3. 環境ファイルを保存します。

関連情報

- 登録のパラメーターに関する詳しい説明は、『**Advanced Overcloud Customizations**』ガイドの「[Registering the Overcloud with an Environment File](#)」の項を参照してください。

4.2. コンテナ化されたサービスの準備

Red Hat OpenStack Platform は、コンテナを使用して OpenStack サービスをホストおよび実行するようになりました。これには、以下の操作を実行する必要があります。

- レジストリーなどのコンテナイメージソースの設定

- イメージソース上のイメージの場所を指定した環境ファイルの生成
- オーバークラウドデプロイメントへの環境ファイルの追加

この環境ファイルを異なるソース種別用に生成する方法についての全手順は、『**director のインストールと使用方法**』ガイドの「[コンテナレジストリー情報の設定](#)」を参照してください。

生成される環境ファイル (`/home/stack/templates/overcloud_images.yaml`) には、各サービスのコンテナイメージをポイントするパラメーターが記載されます。今後実行する全デプロイメント操作でこの環境ファイルを指定してください。

4.3. 新規コンポーザブルサービスの作成

Red Hat OpenStack Platform の今回のバージョンには、新たなコンポーザブルサービスが含まれています。カスタムの **roles_data** ファイルを使用する場合には、これらの新しいサービスを該当するロールに追加してください。

全ロール

以下の新規サービスは全ロールに適用されます。

OS::TripleO::Services::CertmongerUser

オーバークラウドが Certmonger からの証明書を必要とするようにすることができます。TLS/SSL 通信を有効にしている場合にのみ使用されます。

OS::TripleO::Services::Docker

コンテナ化されたサービスを管理するために **docker** をインストールします。

OS::TripleO::Services::MySQLClient

オーバークラウドのデータベースクライアントツールをインストールします。

OS::TripleO::Services::ContainersLogrotateCron

コンテナログ用の **logrotate** サービスをインストールします。

OS::TripleO::Services::Securetty

ノード上で **securetty** の設定ができるようにします。 **environments/securetty.yaml** 環境ファイルで有効化します。

OS::TripleO::Services::Tuned

Linux のチューニングデーモン (**tuned**) を有効化して設定します。

特定のロール

以下の新規サービスは特定のロールに適用されます。

OS::TripleO::Services::Clustercheck

OS::TripleO::Services::MySQL service, such as the **Controller** またはスタンドアロンの **Database** ロールなど、**OS::TripleO::Services::MySQL** サービスも使用するロールに必要です。

OS::TripleO::Services::Iscsid

Controller ロール、**Compute** ロール、**BlockStorage** ロールで **iscsid** サービスを設定します。

OS::TripleO::Services::NovaMigrationTarget

コンピュート ノード上でマイグレーションターゲットサービスを設定します。

カスタムの **roles_data** ファイルを使用する場合には、必要なロールにこれらのサービスを追加してください。

上記に加えて、特定のカスタムロール向けのサービスの最新のリストは、『**Advanced Overcloud Customization**』ガイドの「[Service Architecture: Standalone Roles](#)」の項を参照してください。

4.4. コンポーザブルネットワークの準備

Red Hat OpenStack Platform の今回のバージョンでは、コンポーザブルネットワーク向けの新機能が導入されています。カスタムの **roles_data** ファイルを使用する場合には、ファイルを編集して、コンポーザブルネットワークを各ロールに追加します。コントローラーノードの場合の例を以下に示します。

```
- name: Controller
  networks:
    - External
    - InternalApi
    - Storage
    - StorageMgmt
    - Tenant
```

その他の構文例については、デフォルトの **/usr/share/openstack-tripleo-heat-templates/roles_data.yaml** ファイルを確認してください。また、ロールの例のスニペットについては、**/usr/share/openstack-tripleo-heat-templates/roles** を確認してください。

カスタムのスタンドアロンロールとコンポーザブルネットワークの対応表を以下に示します。

ロール	必要なネットワーク
Ceph Storage Monitor	Storage、StorageMgmt
Ceph Storage OSD	Storage、StorageMgmt
Ceph Storage RadosGW	Storage、StorageMgmt
Cinder API	InternalApi
Compute	InternalApi、Tenant、Storage
Controller	External、InternalApi、Storage、StorageMgmt、Tenant
データベース	InternalApi
Glance	InternalApi
Heat	InternalApi
Horizon	InternalApi

ロール	必要なネットワーク
Ironic	必要なネットワークはなし。API には、プロビジョニング/コントロールプレーンネットワークを使用します。
Keystone	InternalApi
Load Balancer	External、InternalApi、Storage、StorageMgmt、Tenant
Manila	InternalApi
メッセージバス	InternalApi
Networker	InternalApi、Tenant
Neutron API	InternalApi
Nova	InternalApi
OpenDaylight	External、InternalApi、Tenant
Redis	InternalApi
Sahara	InternalApi
Swift API	Storage
Swift ストレージ	StorageMgmt
Telemetry	InternalApi

4.5. 非推奨パラメーターの準備

以下のパラメーターは非推奨となり、ロール固有のパラメーターに置き換えられた点に注意してください。

旧パラメーター	新規パラメーター
controllerExtraConfig	ControllerExtraConfig
OvercloudControlFlavor	OvercloudControllerFlavor
controllerImage	ControllerImage

旧パラメーター	新規パラメーター
NovaImage	ComputeImage
NovaComputeExtraConfig	ComputeExtraConfig
NovaComputeServerMetadata	ComputeServerMetadata
NovaComputeSchedulerHints	ComputeSchedulerHints
NovaComputeIPs	ComputeIPs
SwiftStorageServerMetadata	ObjectStorageServerMetadata
SwiftStorageIPs	ObjectStorageIPs
SwiftStorageImage	ObjectStorageImage
OvercloudSwiftStorageFlavor	OvercloudObjectStorageFlavor

カスタムの環境ファイルでこれらのパラメーターを更新します。

4.6. CEPH STORAGE ノードのアップグレードの準備

コンテナ化されたサービスにアップグレードされたため、Ceph Storage ノードのインストールと更新の方法が変わりました。Ceph Storage の設定では、**ceph-ansible** パッケージ内の Playbook のセットを使用するようになりました。このパッケージはアンダークラウドにインストールします。

前提条件

- オーバークラウドに、director で管理されている Ceph Storage クラスタがあること

手順

1. **ceph-ansible** パッケージをアンダークラウドにインストールします。

```
[stack@director ~]$ sudo yum install -y ceph-ansible
```

2. ストレージの環境ファイルで、最新のリソースと設定を使用するようにします。これは、以下のように変更する必要があります。
 - a. **resource_registry** はコア Heat テンプレートコレクションの **docker/services** サブディレクトリーからコンテナ化されたサービスを使用します。以下に例を示します。

```
resource_registry:
  OS::Triple0::Services::CephMon: ../docker/services/ceph-ansible/ceph-mon.yaml
  OS::Triple0::Services::CephOSD: ../docker/services/ceph-ansible/ceph-
```

```
osd.yaml
```

```
OS::TripleO::Services::CephClient: ../docker/services/ceph-ansible/ceph-client.yaml
```

- a. 新しい **CephAnsibleDisksConfig** パラメーターを使用して、ディスクのマッピングの方法を定義します。以前のバージョンの Red Hat OpenStack Platform では、**ceph::profile::params::osds** hieradata を使用して OSD レイアウトを定義していました。この hieradata を **CephAnsibleDisksConfig** パラメーターの構成に変換します。たとえば、hieradata に以下の内容が記載されているとします。

```
parameter_defaults:
  ExtraConfig:
    ceph::profile::params::osd_journal_size: 512
    ceph::profile::params::osds:
      '/dev/sdb': {}
      '/dev/sdc': {}
      '/dev/sdd': {}
```

この場合には、**CephAnsibleDisksConfig** は以下のようになります。

```
parameters_default:
  CephAnsibleDisksConfig:
    devices:
      - /dev/sdb
      - /dev/sdc
      - /dev/sdd
    journal_size: 512
    osd_scenario: collocated
```

ceph-ansible に使用する OSD ディスクレイアウトオプションの完全な一覧は、**/usr/share/ceph-ansible/group_vars/osds.yml.sample** のサンプルファイルを参照してください。

関連情報

- ハイパーコンバージドインフラストラクチャーを使用する環境には、追加の設定が必要である点に注意してください。『[Hyper-Converged Infrastructure \(HCI\) のアップグレードの準備](#)』を参照してください。
- OpenStack Platform director を使用した **ceph-ansible** の管理に関する詳しい情報は、『**Deploying an Overcloud with Containerized Red Hat Ceph**』ガイドを参照してください。

4.7. HYPER-CONVERGED INFRASTRUCTURE (HCI) のアップグレードの準備

ハイパーコンバージドインフラストラクチャー (HCI) では、Ceph Storage と Compute サービスが単一のロール内に配置されます。ただし、HCI ノードは通常のコンピュータードと同様にアップグレードします。HCI を使用している場合には、コアパッケージのインストールが完了してコンテナサービスが有効化されるまで、Ceph Storage サービスをコンテナ化されたサービスへの移行は遅らせてください。

前提条件

- オーバークラウドで、Compute と Ceph Storage の両方が配置されたロールを使用していること

手順

1. Ceph Storage の設定が含まれた環境ファイルを編集します。
2. **resource_registry** が Puppet リソースを使用するように設定します。以下に例を示します。

```
resource_registry:
  OS::Triple0::Services::CephMon: ../puppet/services/ceph-mon.yaml
  OS::Triple0::Services::CephOSD: ../puppet/services/ceph-osd.yaml
  OS::Triple0::Services::CephClient: ../puppet/services/ceph-client.yaml
```



注記

/usr/share/openstack-tripleo-heat-templates/environments/puppet-ceph.yaml の内容を例として使用します。

3. 「[オーバークラウドノードのアップグレード](#)」の手順に従って、コントローラーベースのノードをコンテナ化されたサービスにアップグレードします。
4. 「[コンピュートノードのアップグレード](#)」の手順に従って HCI ノードをアップグレードします。
5. Ceph Storage 設定の **resource_registry** を編集して、コンテナ化されたサービスを使用するようにします。

```
resource_registry:
  OS::Triple0::Services::CephMon: ../docker/services/ceph-ansible/ceph-mon.yaml
  OS::Triple0::Services::CephOSD: ../docker/services/ceph-ansible/ceph-osd.yaml
  OS::Triple0::Services::CephClient: ../docker/services/ceph-ansible/ceph-client.yaml
```



注記

/usr/share/openstack-tripleo-heat-templates/environments/storage-environment.yaml の内容を例として使用します。

6. ストレージの環境ファイルの **parameter_defaults** セクションに **CephAnsiblePlaybook** パラメーターを追加します。

```
CephAnsiblePlaybook: /usr/share/ceph-ansible/infrastructure-playbooks/switch-from-non-containerized-to-containerized-ceph-daemons.yaml
```

7. ストレージの環境ファイルの **parameter_defaults** セクションに **CephAnsibleDisksConfig** パラメーターを追加して、ディスクレイアウトを定義します。以下に例を示します。

```
CephAnsibleDisksConfig:
  devices:
    - /dev/vdb
    - /dev/vdc
    - /dev/vdd
  journal_size: 512
  osd_scenario: collocated
```

8. 「[アップグレードの最終処理](#)」の手順に従って、オーバークラウドのアップグレードの最終処理を実行します。

関連情報

- OpenStack Platform director を使用した **ceph-ansible** の管理の設定に関する詳しい情報は、『**Deploying an Overcloud with Containerized Red Hat Ceph**』ガイドを参照してください。

4.8. SSL/TLS を介してアンダークラウドのパブリック API にアクセスするための準備

オーバークラウドは、アップグレード中にアンダークラウドの OpenStack Object Storage (swift) のパブリック API にアクセスする必要があります。アンダークラウドで自己署名証明書を使用している場合には、アンダークラウドの認証局を各オーバークラウドノードに追加する必要があります。

前提条件

- アンダークラウドで、パブリック API に SSL/TLS を使用していること

手順

1. director の動的な Ansible スクリプトが OpenStack Platform 12 バージョンに更新され、オーバークラウドプラン内の **RoleNetHostnameMap** Heat パラメーターを使用してインベントリを定義するようになりました。ただし、オーバークラウドは現在 OpenStack Platform 11 のテンプレートバージョンを使用しており、これには **RoleNetHostnameMap** パラメーターがないため、一時的な静的インベントリファイルを作成する必要があります。このファイルは、以下のコマンドを実行すると生成することができます。

```
$ openstack server list -c Networks -f value | cut -d"=" -f2 >
overcloud_hosts
```

2. 以下の内容を記述した Ansible Playbook (**undercloud-ca.yml**) を作成します。

```
---
- name: Add undercloud CA to overcloud nodes
  hosts: all
  user: heat-admin
  become: true
  tasks:
    - name: Copy undercloud CA
```

```

copy:
  src: ca.crt.pem
  dest: /etc/pki/ca-trust/source/anchors/
- name: Update trust
  command: "update-ca-trust extract"
- name: Get the hostname of the undercloud
  delegate_to: 127.0.0.1
  command: hostname
  register: undercloud_hostname
- name: Verify URL
  uri:
    url: https://{ undercloud_hostname.stdout
    }}:13808/healthcheck
    return_content: yes
    register: verify
- name: Report output
  debug:
    msg: "{{ ansible_hostname }}" can access the undercloud's
    Public API"
  when: verify.content == "OK"

```

この Playbook には複数のタスクが含まれており、各ノードで以下の操作を実行します。

- アンダークラウドの認証局のファイル (**ca.crt.pem**) をオーバークラウドノードにコピーします。このファイルの名前と場所は、設定によって異なる場合があります。この例では、自己署名証明書の手順 (『**director のインストールと使用方法**』の「[SSL/TLS 証明書の設定](#)」を参照) で定義された名前と場所を使用しています。
- オーバークラウドノードで、CA トラストデータベースを更新するコマンドを実行します。
- オーバークラウドノードから、アンダークラウドの Object Storage Public API をチェックして、成功したかどうかを報告します。

3. 以下のコマンドで Playbook を実行します。

```
$ ansible-playbook -i overcloud_hosts undercloud-ca.yml
```

ここでは、一時インベントリを使用して、Ansible にオーバークラウドノードを指定します。

4. その結果、Ansible の出力には、ノードのデバッグメッセージが表示されます。以下に例を示します。

```

ok: [192.168.24.100] => {
  "msg": "overcloud-controller-0 can access the undercloud's
  Public API"
}

```

関連情報

- オーバークラウドでの Ansible 自動化の実行に関する詳しい情報は、『**director のインストールと使用方法**』ガイドの「[Ansible 自動化の実行](#)」を参照してください。

4.9. 事前にプロビジョニングされたノードのアップグレードの準備

事前にプロビジョニング済みのノードは、director の管理外で作成されたノードです。事前にプロビジョニング済みのノードを使用するオーバークラウドには、アップグレードの前に追加のステップがいくつか必要です。

前提条件

- オーバークラウドが事前にプロビジョニング済みのノードを使用していること

手順

1. 以下のコマンドを実行して、**OVERCLOUD_HOSTS** 環境変数内のノードの IP アドレスの一覧を保存します。

```
$ source ~/stackrc
$ export OVERCLOUD_HOSTS=$(openstack server list -f value -c
Networks | cut -d "=" -f 2 | tr '\n' ' ')
```

2. 以下のスクリプトを実行します。

```
$ /usr/share/openstack-tripleo-heat-templates/deployed-
server/scripts/enable-ssh-admin.sh
```

3. アップグレードを開始します。
4. コンピュートノードまたは Object Storage ノードをアップグレードする場合には、以下を使用します。
 - a. **upgrade-non-controller.sh** スクリプトに **-U** オプションを使用して **stack** ユーザーを指定します。事前にプロビジョニング済みのノードのデフォルトユーザーは **heat-admin** ではなく **stack** であることがその理由です。
 - b. **--upgrade** オプションでノードの IP アドレスを使用します。これは、ノードが director の Compute (Nova) と Bare Metal (ironic) のサービスで管理されるのでノード名がないためです。
以下に例を示します。

```
$ upgrade-non-controller.sh -U stack --upgrade 192.168.24.100
```

関連情報

- 事前にプロビジョニングされたノードに関する詳しい情報は、『**director のインストールと使用方法**』ガイドの「[事前にプロビジョニングされたノードを使用した基本的なオーバークラウドの設定](#)」を参照してください。

4.10. NFV を設定したオーバークラウドの準備

Red Hat OpenStack Platform 11 から Red Hat OpenStack Platform 12 にアップグレードすると、OVS パッケージもバージョン 2.6 から 2.7 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 12 は OVS クライアントモードで稼働します。

前提条件

- オーバークラウドでネットワーク機能仮想化 (NFV) を使用していること

手順

オーバークラウドを Red Hat OpenStack Platform 11 から OVS-DPDK を設定した Red Hat OpenStack Platform 12 にアップグレードする場合には、環境ファイルに以下の追加パラメーターを設定する必要があります。

1. **parameter_defaults** セクションで、アップグレードプロセス中に **os-net-config** を実行して、OVS 2.7 PCI アドレスを DPDK ポートに関連付けするためのネットワークデプロイメントパラメーターを追加します。

```
parameter_defaults:
  ComputeNetworkDeploymentActions: ['CREATE', 'UPDATE']
```

パラメーター名は、DPDK のデプロイに使用するロール名と一致する必要があります。この例では、ロール名は **Compute** なので、パラメーター名は **ComputeNetworkDeploymentActions** となっています。



注記

初回のアップグレードの後には、このパラメーターは必要ないので、環境ファイルから削除すべきです。

2. **resource_registry** セクションで、**ComputeNeutronOvsAgent** サービスを **neutron-ovs-dpdk-agent puppet** サービスに上書きします。

```
resource_registry:
  OS::TripleO::Services::ComputeNeutronOvsAgent:
    /usr/share/openstack-tripleo-heat-templates/puppet/services/neutron-ovs-dpdk-agent.yaml
```

Red Hat OpenStack Platform 12 は、新しい **ComputeOvsDpdk** ロールの追加をサポートするための新しいサービス (**OS::TripleO::Services::ComputeNeutronOvsDpdk**) を追加しました。上記の例では、このサービスをアップグレード向けに外部にマッピングしています。

編集した環境ファイルは、「[オーバークラウドノードのアップグレード](#)」の **openstack overcloud deploy** コマンドの一部として追加します。

4.11. オーバークラウドのアップグレードの一般的な考慮事項

オーバークラウドのアップグレードを実行する前に考慮すべき一般的な注意事項は以下の通りです。

Custom ServiceNetMap

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「[Isolating Networks](#)」のセクションを参照してください。

外部のロードバランシング

外部のロードバランシングを使用している場合には、ロードバランサーに追加する新規サービスがあるかどうかを確認してください。サービスの設定については、『**External Load Balancing for the Overcloud**』ガイドの「[Configuring Load Balancing Options](#)」の項を参照してください。

非推奨となったデプロイメントオプション

`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。

第5章 オーバークラウドのアップグレード

以下の手順では、オーバークラウドをアップグレードします。

前提条件

- アンダークラウドが最新バージョンにアップグレードされていること
- アップグレードでの変更に対応するためのカスタム環境ファイルの準備が完了していること

5.1. オーバークラウドノードのアップグレード

major-upgrade-composable-steps-docker.yaml 環境ファイルにより、**roles_data** ファイル内で **disable_upgrade_deployment: True** と指定されているロールを除くすべてのカスタムロール上の全コンポーザブルサービスがアップグレードされます。

前提条件

- アンダークラウドが最新バージョンにアップグレードされていること
- アップグレードでの変更に対応するためのカスタム環境ファイルの準備が完了していること

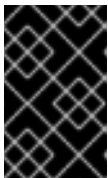
手順

1. **openstack overcloud deploy** コマンドに以下の項目を指定して実行します。

- ネットワーク分離やストレージなど、お使いの環境に関連したすべてのオプションおよびカスタムの環境ファイル
- 「[アップグレードの最終処理](#)」で生成された **overcloud_images.yaml** 環境ファイル
- **major-upgrade-composable-steps-docker.yaml** 環境ファイル
以下に例を示します。

```
$ openstack overcloud deploy --templates \
  -e /home.stack/templates/node_count.yaml \
  -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 /home/stack/templates/network_environment.yaml \
  -e /home/stack/templates/overcloud_images.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/major-
upgrade-composable-steps-docker.yaml \
  --ntp-server pool.ntp.org
```

2. 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

アップグレードにより OpenStack Networking (neutron) サーバーと L3 エージェントが無効化されるので、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。この間には、インスタンスへのアクセスは引き続き可能です。

3. 全サービスがアクティブかどうかを確認します。たとえば、コントローラーノード上のサービスを確認するには、以下のコマンドを実行します。

```
[stack@director ~]$ ssh heat-admin@192.168.24.10
[heat-admin@overcloud-controller-0 ~]$ sudo pcs status
[heat-admin@overcloud-controller-0 ~]$ sudo docker ps
```

関連情報

- このステップの完了後に何らかの問題が発生した場合には、Red Hat に連絡して、アドバイスおよびサポートを依頼してください。

5.2. OBJECT STORAGE ノードのアップグレード

スタンドアロンの Object Storage ノードは、個別に更新してサービスが有効な状態を維持する必要があります。オーバークラウドの主要なアップグレードプロセスには含まれません。director には、個々の Object Storage ノードでアップグレードを実行するためのスクリプトが含まれています。

前提条件

- **major-upgrade-composable-steps-docker.yaml** 環境ファイルを指定して **openstack overcloud deploy** を実行し、主要なカスタムロールとそれらのコンポーザブルサービスをアップグレード済みであること

手順

1. Object Storage ノードの一覧を取得します。

```
$ openstack server list -c Name -f value --name objectstorage
```

2. 一覧内の各 Object Storage ノードで以下のステップを実行します。

- a. アップグレードするノードを特定するためのノード名を使用して **upgrade-non-controller.sh** スクリプトを実行します。

```
$ upgrade-non-controller.sh --upgrade overcloud-objectstorage-0
```



注記

事前にプロビジョニング済みのノードインフラストラクチャーを使用している場合には、このコマンドでの変更について、[「アップグレードの最終処理」](#)を参照してください。

- b. Object Storage ノードのアップグレードが完了するまで待ちます。
- c. Object Storage ノードを再起動します。

```
$ openstack server reboot overcloud-objectstorage-0
```

- d. Object Storage ノードの再起動が完了するまで待ちます。

関連情報

- このステップの完了後に何らかの問題が発生した場合には、Red Hat に連絡して、アドバイスおよびサポートを依頼してください。

5.3. 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は、オーバークラウドの主要なアップグレードプロセスには含まれていません。インスタンスのアップタイムが最大限となるようにするには、ノードをアップグレードする前に 各インスタンスをコンピュートノードから移行してください。このため、コンピュートノードのアップグレードプロセスでは以下のステップが必要となります。

1. アップグレードするコンピュートノードの選択
2. 別のコンピュートノードへの移行
3. 空のコンピュートノードのアップグレード
4. アップグレードしたコンピュートノードの再起動と有効化

前提条件

- **major-upgrade-composable-steps-docker.yaml** 環境ファイルを指定して「openstack overcloud deploy」コマンドを事前に実行済みであること。このコマンドは主要なカスタムロールとそれらのコンポーザブルをアップグレードします。

手順

アップグレードするコンピュートノードの選択

1. 全コンピュートノードを一覧表示します。

```
$ source ~/stackrc
$ openstack server list -c Name -f value --name compute
```

2. アップグレードするコンピュートノードを選択して、UUID と名前を書き留めておきます。

別のコンピュートノードへの移行

1. アンダークラウドから、再起動する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
```

2. 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
```

3. 以下のコマンドの 1 つを使用して、インスタンスを移行します。

- a. 選択した特定のホストにインスタンスを移行します。

```
(overcloud) $ 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。

```
(overcloud) $ nova live-migration [instance-id]
```

- c. 一度にすべてのインスタンスのライブマイグレーションを行います。

```
$ 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、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. 正常に移行したことを確認します。

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

6. 選択した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。

空のコンピュートノードのアップグレード

1. アップグレードするノードを特定するためのノード名を使用して **upgrade-non-controller.sh** スクリプトを実行します。

```
$ upgrade-non-controller.sh --upgrade overcloud-compute-0
```



注記

事前にプロビジョニング済みのノードインフラストラクチャーを使用している場合には、このコマンドでの変更について、「[アップグレードの最終処理](#)」を参照してください。

2. コンピュート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トノードの再起動と有効化

1. コンピュートノードにログインして、再起動します。

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

2. ノードが起動するまで待ちます。
3. 再度コンピュートノードを有効化します。

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

4. コンピュートノードが有効化されているかどうかを確認します。

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

アップグレードする次のノードを選択します。アップグレードを開始する前には、[そのノード上のインスタンスを別のコンピュートノードに移行します](#)。全コンピュートノードのアップグレードが完了するまで、このプロセスを繰り返してください。

関連情報

- このステップの完了後に何らかの問題が発生した場合には、Red Hat に連絡して、アドバイスおよびサポートを依頼してください。

5.4. 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-docker.yaml**) を指定する必要があります。

前提条件

- 全ノードのアップグレードが完了していること

手順

1. **openstack overcloud deploy** コマンドに以下の項目を指定して実行します。

- ネットワーク分離やストレージなど、お使いの環境に関連したすべてのオプションおよびカスタムの環境ファイル
- [「アップグレードの最終処理」](#) で生成された **overcloud_images.yaml** 環境ファイル
- major-upgrade-converge-docker.yaml** 環境ファイル
以下に例を示します。

```
$ openstack overcloud deploy --templates \
  -e /home.stack/templates/node_count.yaml \
  -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 /home/stack/templates/network_environment.yaml \
  -e /home/stack/templates/overcloud_images.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/major-
upgrade-converge-docker.yaml \
  --ntp-server pool.ntp.org
```

2. 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。
3. 全サービスがアクティブかどうかを確認します。たとえば、コントローラーノード上のサービスを確認するには、以下のコマンドを実行します。

```
[stack@director ~]$ ssh heat-admin@192.168.24.10
[heat-admin@overcloud-controller-0 ~]$ sudo pcs status
[heat-admin@overcloud-controller-0 ~]$ sudo systemctl list-units
```

```
| 'openstack-*' 'neutron-*' 'httpd*'
```

関連情報

- このステップの完了後に何らかの問題が発生した場合には、Red Hat に連絡して、アドバイスおよびサポートを依頼してください。

第6章 アップグレード後のステップの実行

以下の手順では、主要なアップグレードプロセスが完了した後の最終ステップを実行します。

前提条件

- オーバークラウドを最新のメジャーリリースにアップグレードする作業が完了していること

6.1. 新規オーバークラウドノードへのアンダークラウド **CA** の追加

「[SSL/TLS を介してアンダークラウドのパブリック API にアクセスするための準備](#)」では、既存の全オーバークラウドノードでアンダークラウドの認証局 (CA) を追加しました。スケーリングまたは置き換えによって追加された新規ノードでも CA が必要です。これにより、新しいオーバークラウドノードが OpenStack Object Storage (swift) のパブリック API にアクセスできるようになります。以下の手順では、すべての新しいオーバークラウドノードでアンダークラウドの CA を追加する方法について説明します。

前提条件

- Red Hat OpenStack Platform 12 へのアップグレードが完了していること
- アンダークラウドでパブリック API に SSL/TLS を使用していること

手順

1. 環境ファイルを新規作成するか、既存のファイルを編集します。この例では **undercloud-ca-map.yaml** というファイル名を使用しています。
2. 環境ファイルの **parameter_defaults** セクションに **CAMap** パラメーターを追加します。以下の構文を例として使用してください。

```
parameter_defaults:
  CAMap:
    undercloud-ca: ①
    content: | ②
      -----BEGIN CERTIFICATE-----
      MIIDlTCCAn2gAwIBAgIJAOnPtx2hHEhrMA0GCSqGSIsb3D
      BAYTAlVTMQswCQYDVQQIDAJQZzEQMA4GA1UEBwwHUmFsZ
      UmVkJIEhhdDELMAkGA1UECwwCUUUxFDASBgNVBAMMCzE5M
      ...
      -----END CERTIFICATE-----
```

- ① これは、各オーバークラウドノードのトラストデータベースで CA を識別する名前です。
- ② **content** セクションは、実際の CA 証明書です。このセクションに CA の内容をコピーして貼り付けます。CA の ID が YAML 構文の要件に合っていることを確認してください。

3. このファイルを保存します。
4. このファイルは、**openstack overcloud deploy** コマンドを次回に実行する際に指定してください。

関連情報

- オーバークラウドでの CA トラストの設定については、『Advanced Overcloud Customization』ガイドの「[Enabling SSL/TLS on the Overcloud](#)」を参照してください。

6.2. オーバークラウドのアップグレード後の一般的な考慮事項

以下の項目は、オーバークラウドのアップグレード後の一般的な考慮事項です。

- 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。パッケージをアップグレードすると、各サービスのアップグレードされたバージョンに適した **.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。
- 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
```

第7章 OPENSTACK PLATFORM の最新状態の維持

このプロセスでは、OpenStack Platform 環境のマイナーバージョンを最新の状態に維持する方法について説明します。これは、**Red Hat OpenStack Platform 12** 内のマイナー更新です。

前提条件

- オーバークラウドが **Red Hat OpenStack Platform 12** にアップグレードされていること
- 新しいパッケージとコンテナイメージが **Red Hat OpenStack Platform 12** 内で利用可能であること

7.1. 更新前のアンダークラウドの検証

Red Hat OpenStack Platform 12 のアンダークラウドを更新する前に機能を確認する手順を以下に示します。

手順

1. アンダークラウドのアクセス情報を読み込みます。

```
$ source ~/stackrc
```

2. エラーが発生している Systemd サービスがあるかどうかを確認します。

```
(undercloud) $ sudo systemctl list-units --state=failed 'openstack*' 'neutron*' 'httpd' 'docker'
```

3. アンダークラウドの空き領域を確認します。

```
(undercloud) $ df -h
```

4. アンダークラウドでクロックが同期されていることを確認します。

```
(undercloud) $ sudo ntpstat
```

5. アンダークラウドのネットワークサービスを確認します。

```
(undercloud) $ openstack network agent list
```

全エージェントが **Alive** で、それらの状態が **UP** である必要があります。

6. アンダークラウドの Compute サービスを確認します。

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

全エージェントのステータスが **enabled** で、状態が **up** である必要があります。

7. アンダークラウドのボリュームサービスを確認します。

```
(undercloud) $ openstack volume service list
```

全エージェントのステータスが **enabled** で、状態が **up** である必要があります。

関連情報

- OpenStack Orchestration (heat) のデータベースで削除済みとマークされている stack のエントリを完全削除する方法は <https://access.redhat.com/solutions/2215131> のソリューションに記載されています。

7.2. 更新前のオーバークラウドの検証

Red Hat OpenStack Platform 12 のオーバークラウドを更新する前に機能を確認する手順を以下に示します。

手順

1. アンダークラウドのアクセス情報を読み込みます。

```
$ source ~/stackrc
```

2. ベアメタルノードのステータスを確認します。

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

全ノードの電源状態が有効で (**on**)、かつメンテナンスモードが **false** である必要があります。

3. エラーが発生している Systemd サービスがあるかどうかを確認します。

```
(undercloud) $ for NODE in $(openstack server list -f value -c Networks | cut -d= -f2); do echo "=== $NODE ===" ; ssh heat-admin@$NODE "sudo systemctl list-units --state=failed 'openstack*' 'neutron*' 'httpd' 'docker' 'ceph*'" ; done
```

4. エラーが発生しているコンテナ化されたサービスがあるかどうかを確認します。

```
(undercloud) $ for NODE in $(openstack server list -f value -c Networks | cut -d= -f2); do echo "=== $NODE ===" ; ssh heat-admin@$NODE "sudo docker ps -f 'exited=1' --all" ; done
(undercloud) $ for NODE in $(openstack server list -f value -c Networks | cut -d= -f2); do echo "=== $NODE ===" ; ssh heat-admin@$NODE "sudo docker ps -f 'status=dead' -f 'status=restarting'" ; done
```

5. 全サービスへの HAProxy 接続をチェックします。コントロールプレーンの仮想 IP アドレスと **haproxy.stats** サービスの認証情報を取得します。

```
(undercloud) $ NODE=$(openstack server list --name controller-0 -f value -c Networks | cut -d= -f2); ssh heat-admin@$NODE sudo 'grep "listen haproxy.stats" -A 6 /var/lib/config-data/puppet-generated/haproxy/etc/haproxy/haproxy.cfg'
```

以下の cURL 要求でそれらの情報を使用します。

```
(undercloud) $ curl -s -u admin:<PASSWORD> "http://<IP
ADDRESS>:1993/;csv" | egrep -vi "(frontend|backend)" | awk -F',' '{
print $1" "$2" "$18 }'
```

<PASSWORD> と <IP ADDRESS> は、haproxy.stats サービスからのそれぞれの情報に置き換えます。その結果表示される一覧には、各ノード上の OpenStack Platform サービスとそれらの接続ステータスが表示されます。

6. オーバークラウドデータベースのレプリケーションの正常性をチェックします。

```
(undercloud) $ for NODE in $(openstack server list --name controller
-f value -c Networks | cut -d= -f2); do echo "=== $NODE ===" ; ssh
heat-admin@$NODE "sudo docker exec clustercheck clustercheck" ; done
```

7. RabbitMQ クラスターの正常性を確認します。

```
(undercloud) $ for NODE in $(openstack server list --name controller
-f value -c Networks | cut -d= -f2); do echo "=== $NODE ===" ; ssh
heat-admin@$NODE "sudo docker exec $(ssh heat-admin@$NODE "sudo
docker ps -f 'name=.*rabbitmq.*' -q") rabbitmqctl node_health_check"
; done
```

8. Pacemaker リソースの正常性を確認します。

```
(undercloud) $ NODE=$(openstack server list --name controller-0 -f
value -c Networks | cut -d= -f2); ssh heat-admin@$NODE "sudo pcs
status"
```

以下の点を確認します。

- 全クラスターノードが **online** であること。
- いずれのクラスターノード上でも **stopped** のリソースがないこと。
- pacemaker で **failed** のアクションがないこと。

9. 各オーバークラウドノードでディスク領域を確認します。

```
(undercloud) $ for NODE in $(openstack server list -f value -c
Networks | cut -d= -f2); do echo "=== $NODE ===" ; ssh heat-
admin@$NODE "sudo df -h --output=source,fstype,avail -x overlay -x
tmpfs -x devtmpfs" ; done
```

10. オーバークラウドの Ceph Storage クラスターの正常性を確認します。以下のコマンドを使用すると、コントローラーノード上で **ceph** ツールが実行されて、クラスターをチェックします。

```
(undercloud) $ NODE=$(openstack server list --name controller-0 -f
value -c Networks | cut -d= -f2); ssh heat-admin@$NODE "sudo ceph -
s"
```

11. Ceph Storage OSD に空き領域があるかどうかを確認します。以下のコマンドを使用すると、コントローラーノード上で **ceph** ツールが実行され、空き領域をチェックします。

■

```
(undercloud) $ NODE=$(openstack server list --name controller-0 -f value -c Networks | cut -d= -f2); ssh heat-admin@$NODE "sudo ceph df"
```

12. オーバークラウドノードでクロックが同期されていることを確認します。

```
(undercloud) $ for NODE in $(openstack server list -f value -c Networks | cut -d= -f2); do echo "=== $NODE ===" ; ssh heat-admin@$NODE "sudo ntpstat" ; done
```

13. オーバークラウドのアクセス情報を読み込みます。

```
(undercloud) $ source ~/overcloudrc
```

14. オーバークラウドのネットワークサービスを確認します。

```
(overcloud) $ openstack network agent list
```

全エージェントが **Alive** で、それらの状態が **UP** である必要があります。

15. オーバークラウドの Compute サービスを確認します。

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

全エージェントのステータスが **enabled** で、状態が **up** である必要があります。

16. オーバークラウドのボリュームサービスを確認します。

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

全エージェントのステータスが **enabled** で、状態が **up** である必要があります。

関連情報

- [「How can I verify my OpenStack environment is deployed with Red Hat recommended configurations?」](#)の記事を参照してください。この記事には、Red Hat OpenStack Platform 環境をチェックして、Red Hat の推奨値に合わせて調整する方法が記載されています。

7.3. アンダークラウドの最新状態の維持

director では、アンダークラウドノード上の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。これにより、OpenStack Platform 環境の現在のバージョン内のマイナー更新を実行することができます。これは、**Red Hat OpenStack Platform 12** 内でのマイナー更新です。

前提条件

- **Red Hat OpenStack Platform 12** を使用していること
- アンダークラウドのバックアップを実行済みであること

手順

1. director に **stack** ユーザーとしてログインします。

2. **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 python-tripleoclient
```

3. **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
```

4. ノードを再起動します。

```
$ sudo reboot
```

5. ノードが起動するまで待ちます。

6. 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。

7. 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
```

オーバークラウドのイメージを最新の状態に維持して、イメージの設定が最新の **openstack-tripleo-heat-template** パッケージの要件と一致するようにします。デプロイメントを成功させ、将来オペレーションをスケーリングできるようにするには、[「オーバークラウドイメージの最新状態の維持」](#)に記載の方法に従ってオーバークラウドのイメージを更新します。

7.4. オーバークラウドイメージの最新状態の維持

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。このプロセスにより、**Red Hat OpenStack Platform 12** 内のアンダークラウドでそれらのイメージが更新されます。

前提条件

- **Red Hat OpenStack Platform 12** を使用していること
- 現行バージョンのアンダークラウドの最新のマイナーリリースに更新済みであること

手順

1. **yum** ログをチェックして、新規イメージのアーカイブが利用可能かどうかを確認します。

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

2. 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/*
```

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-12.0.tar /usr/share/rhosp-director-images/ironic-python-agent-latest-12.0.tar; do tar -xvf $i; done
```

4. 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 /home/stack/images/
$ openstack overcloud node configure $(openstack baremetal node list -c UUID -f csv --quote none | sed "1d" | paste -s -d " ")
```

5. 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。

7.5. オーバークラウドの最新状態の維持

director では、全オーバークラウドノード上の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。これにより、OpenStack Platform 環境の現在のバージョン内のマイナー更新を実行することができます。これは、**Red Hat OpenStack Platform 12** 内でのマイナー更新です。

前提条件

- **Red Hat OpenStack Platform 12** を使用していること
- 現行バージョンのアンダークラウドの最新のマイナーリリースに更新済みであること
- オーバークラウドのバックアップを実行済みであること

手順

1. コンテナ化されたサービスのイメージの最新タグを確認します。


```
$ openstack overcloud container image tag discover \
  --image registry.access.redhat.com/rhosp12/openstack-base:latest \
  --tag-from-label version-release
```

最も新しいタグを書き留めておきます。

2. **openstack overcloud container image prepare** コマンドで、コンテナイメージのソース用の更新された環境ファイルを作成します。たとえば、**registry.access.redhat.com** からのイメージを使用する場合は、以下のコマンドを実行します。

```
$ openstack overcloud container image prepare \
  --namespace=registry.access.redhat.com/rhosp12 \
  --prefix=openstack- \
  --tag [TAG] \ ①
  --set ceph_namespace=registry.access.redhat.com/rhceph \
  --set ceph_image=rhceph-2-rhel7 \
  --set ceph_tag=latest \
  --env-file=/home/stack/templates/overcloud_images.yaml \
  -e /home/stack/templates/custom_environment_file.yaml ②
```

① **[TAG]** の箇所は、前のステップで取得したタグに置き換えます。

② **-e** パラメーターで追加の環境ファイルを指定します。director は指定されているすべての環境ファイルでカスタムリソースを確認して、コンテナ化されたサービスに必要なコンテナイメージを特定します。

この環境ファイルを異なるソース種別用に生成する方法についての詳しい情報は、『**director のインストールと使用方法**』ガイドの「[コンテナレジストリー情報の設定](#)」を参照してください。

3. **openstack overcloud update stack** コマンドを実行して、オーバークラウド内のコンテナイメージの場所を更新します。

```
$ openstack overcloud update stack --init-minor-update \
  --container-registry-file
  /home/stack/templates/overcloud_images.yaml
```

--init-minor-update はオーバークラウドスタック内のパラメーターの更新のみを実行します。実際のパッケージやコンテナの更新は行いません。このコマンドが完了するまで待ちます。

4. **openstack overcloud update** コマンドを使用してパッケージとコンテナの更新を実行します。**--nodes** オプションを使用して、各ロールのノードをアップグレードします。たとえば、以下のコマンドは、**Controller** ロールのノードを更新します。

```
$ openstack overcloud update stack --nodes Controller
```

以下の順序で、各ロールグループにこのコマンドを実行します。

- **Controller**
- **CephStorage**

- **Compute**
 - **ObjectStorage**
 - **Database、MessageBus、Networker** などのカスタムのロール
5. 選択したロールの更新プロセスが開始します。director は Ansible playbook を使用して更新を実行し、各タスクの出力を表示します。
 6. 次のロールグループを更新します。全ノードの更新が完了するまで作業を繰り返してください。
 7. 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 を更新した場合には、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。