



Red Hat OpenStack Platform 16.1

Red Hat OpenStack Platform の最新状態の維持

Red Hat OpenStack Platform のマイナーアップデートの実施

Red Hat OpenStack Platform 16.1 Red Hat OpenStack Platform の最新状態の維持

Red Hat OpenStack Platform のマイナーアップデートの実施

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

法律上の通知

Copyright © 2021 | You need to change the HOLDER entity in the en-US/Keeping_Red_Hat_OpenStack_Platform_Updated.ent file |.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

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

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

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

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

All other trademarks are the property of their respective owners.

概要

本書では、お使いの Red Hat OpenStack Platform 環境のマイナーアップデートを行う手順について説明します。

目次

前書き	3
多様性を受け入れるオープンソースの強化	4
RED HAT ドキュメントへのフィードバック (英語のみ)	5
第1章 はじめに	6
1.1. ワークフローの概要	6
1.2. 更新を妨げる可能性のある既知の問題	6
第2章 マイナーアップデートの準備	8
2.1. 環境の RED HAT ENTERPRISE LINUX リリースへのロック	8
2.2. EXTENDED UPDATE SUPPORT (EUS) リポジトリへの変更	9
2.3. RED HAT OPENSTACK PLATFORM および ANSIBLE リポジトリの更新	11
2.4. CONTAINER-TOOLS および VIRT モジュールバージョンの設定	12
2.5. コンテナイメージ準備ファイルの更新	13
2.6. SSL/TLS 設定の更新	14
2.7. オーバークラウドでのフェンシングの無効化	14
第3章 アンダークラウドの更新	16
3.1. コンテナ化されたアンダークラウドのマイナーアップデートの実施	16
3.2. オーバークラウドイメージの更新	16
3.3. アンダークラウドアップグレード後の注意事項	17
3.4. 次のステップ	17
第4章 オーバークラウドの更新	18
4.1. オーバークラウドの更新準備の実行	18
4.2. コンテナイメージ準備タスクの実行	19
4.3. 全コントローラーノードの更新	19
4.4. すべてのコンピュートノードの更新	20
4.5. 全 HCI コンピュートノードの更新	21
4.6. すべての CEPH STORAGE ノードの更新	21
4.7. データベースのオンライン更新の実施	22
4.8. 更新の最終処理	22
第5章 オーバークラウドのリポート	24
5.1. コントローラーノードおよびコンポーザブルノードのリポート	24
5.2. CEPH STORAGE (OSD) クラスターのリポート	24
5.3. コンピュートノードのリポート	25

前書き

多様性を受け入れるオープンソースの強化

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

RED HAT ドキュメントへのフィードバック (英語のみ)

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

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

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

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

第1章 はじめに

本書で説明するワークフローは、お使いの Red Hat OpenStack Platform 16.1 環境が最新のパッケージおよびコンテナで更新された状態を維持するのに役立ちます。

本ガイドは、以下のバージョンのアップグレードパスを提供します。

現在の OpenStack バージョン	新しい OpenStack バージョン
Red Hat OpenStack Platform 16.0	Red Hat OpenStack Platform 16.1.z
Red Hat OpenStack Platform 16.1	Red Hat OpenStack Platform 16.1.z

1.1. ワークフローの概要

以下の表には、アップグレードのプロセスに必要なステップの概要をまとめています。

ステップ	説明
アンダークラウドの更新	アンダークラウドを最新の OpenStack Platform 16.1.z バージョンに更新します。
オーバークラウドの更新	オーバークラウドを最新の OpenStack Platform 16.1.z バージョンに更新します。
Ceph Storage ノードの更新	すべての Ceph Storage サービスをアップグレードします。
アップグレードの最終段階	コンバージェンスのコマンドを実行して、オーバークラウドスタックをリフレッシュします。

1.2. 更新を妨げる可能性のある既知の問題

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

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

ノードのクラスターをシャットダウンする際に生じる競合状態により、Pacemaker バージョン **2.0.3-5.el8_2.4** を実行するオーバークラウドノードで更新に失敗する場合があります。

現在オーバークラウドノードのいずれかに Pacemaker バージョン **2.0.3-5.el8_2.4** がインストールされている場合、[BZ#1973660](#) を避けるためにオーバークラウドノードを更新する前に Pacemaker をアップグレードする必要があります。詳細は、以下の Red Hat ナレッジベースのソリューション「[Update from OSP16.1 to OSP16.2 might fail to update certain HA containers](#)」を参照してください。

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

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

第2章 マイナーアップデートの準備

Red Hat OpenStack Platform 16.1を最新のマイナーリリースに更新するプロセスを開始する前に、アンダークラウドおよびオーバークラウドで準備手順を実施する必要があります。

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

Red Hat OpenStack Platform 16.1は Red Hat Enterprise Linux 8.2 でサポートされています。更新を実施する前に、オペレーティングシステムをより新しいマイナーリリースにアップグレードしないように、アンダークラウドおよびオーバークラウドのリポジトリを Red Hat Enterprise Linux 8.2 リリースにロックします。

手順

1. アンダークラウドに **stack** ユーザーとしてログインします。
2. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

3. **RhsmVars** パラメーターが含まれるオーバークラウドのサブスクリプション管理用環境ファイルを編集します。通常、このファイルのデフォルト名は **rhsm.yml** です。
4. サブスクリプション管理の設定で **rhsm_release** パラメーターを確認します。このパラメーターが設定されていない場合は、このパラメーターを追加してパラメーターを 8.2 に設定します。

```
parameter_defaults:
  RhsmVars:
    ...
    rhsm_username: "myusername"
    rhsm_password: "p@55w0rd!"
    rhsm_org_id: "1234567"
    rhsm_pool_ids: "1a85f9223e3d5e43013e3d6e8ff506fd"
    rhsm_method: "portal"
    rhsm_release: "8.2"
```

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

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

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

7. すべてのノードで、オペレーティングシステムのバージョンを Red Hat Enterprise Linux 8.2 にロックするタスクが含まれる Playbook を作成します。

```
$ cat > ~/set_release.yaml <<'EOF'
- hosts: all
  gather_facts: false
  tasks:
```

```
- name: set release to 8.2
  command: subscription-manager release --set=8.2
  become: true
EOF
```

8. **set_release.yaml** Playbook を実行します。

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

すべての Red Hat OpenStack Platform ノードにコンテンツを適用するには、**--limit** オプションを使用します。Ceph Storage ノードに対してこの Playbook を実行しないでください。これらのノードには、別のサブスクリプションを使用している可能性が高いためです。



注記

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

```
$ sudo subscription-manager release --set=8.2
```

2.2. EXTENDED UPDATE SUPPORT (EUS) リポジトリへの変更

Red Hat OpenStack Platform サブスクリプションには、Red Hat Enterprise Linux 8.2 Extended Update Support (EUS) のリポジトリが含まれます。EUS リポジトリには、Red Hat Enterprise Linux 8.2 の最新のセキュリティーパッチおよびバグ修正が含まれています。更新を実行する前に、以下のリポジトリに切り替えます。

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



重要

特定バージョンの Podman との互換性を維持するには、EUS リポジトリを使用する必要があります。より新しいバージョンの Podman は Red Hat OpenStack Platform の本リリースに対してテストされておらず、予期せぬ結果を招く可能性があります。

手順

1. アンダークラウドに **stack** ユーザーとしてログインします。
2. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

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

```
parameter_defaults:
  RhsmVars:
    rhsm_repos:
      - rhel-8-for-x86_64-baseos-eus-rpms
      - rhel-8-for-x86_64-appstream-eus-rpms
      - rhel-8-for-x86_64-highavailability-eus-rpms
      - ansible-2.9-for-rhel-8-x86_64-rpms
      - advanced-virt-for-rhel-8-x86_64-rpms
      - openstack-16.1-for-rhel-8-x86_64-rpms
      - rhceph-4-tools-for-rhel-8-x86_64-rpms
      - fast-datapath-for-rhel-8-x86_64-rpms
```

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

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

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

7. すべてのノードで、リポジトリを Red Hat Enterprise Linux 8.2 EUS に設定するタスクが含まれる Playbook を作成します。

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

8. **change_eus.yaml** Playbook を実行します。

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

すべての Red Hat OpenStack Platform ノードにコンテンツを適用するには、**--limit** オプションを使用します。Ceph Storage ノードに対してこの Playbook を実行しないでください。これらのノードには、別のサブスクリプションを使用している可能性が高いためです。

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

Red Hat OpenStack Platform 16.1 パッケージおよび Ansible 2.9 パッケージを使用するようにリポジトリを更新します。

手順

1. アンダークラウドに **stack** ユーザーとしてログインします。
2. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

3. **RhsmVars** パラメーターが含まれるオーバークラウドのサブスクリプション管理用環境ファイルを編集します。通常、このファイルのデフォルト名は **rhsm.yml** です。
4. サブスクリプション管理の設定で **rhsm_repos** パラメーターを確認します。 **rhsm_repos** パラメーターで Red Hat OpenStack Platform 16.0 リポジトリおよび Ansible 2.9 リポジトリが使用されている場合は、リポジトリを正しいバージョンに変更します。

```
parameter_defaults:
  RhsmVars:
    rhsm_repos:
      - rhel-8-for-x86_64-baseos-eus-rpms
      - rhel-8-for-x86_64-appstream-eus-rpms
      - rhel-8-for-x86_64-highavailability-eus-rpms
      - ansible-2.9-for-rhel-8-x86_64-rpms
      - advanced-virt-for-rhel-8-x86_64-rpms
      - openstack-16.1-for-rhel-8-x86_64-rpms
      - rhceph-4-osd-for-rhel-8-x86_64-rpms
      - rhceph-4-mon-for-rhel-8-x86_64-rpms
      - rhceph-4-tools-for-rhel-8-x86_64-rpms
      - fast-datapath-for-rhel-8-x86_64-rpms
```

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

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

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

7. すべてのノードで、リポジトリを Red Hat OpenStack Platform 16.1 に設定するタスクが含まれる Playbook を作成します。

```
$ cat > ~/update_rhosp_repos.yaml <<'EOF'
- hosts: all
  gather_facts: false
  tasks:
    - name: change osp repos
      command: subscription-manager repos --disable=openstack-16-for-rhel-8-x86_64-rpms --
enable=openstack-16.1-for-rhel-8-x86_64-rpms --disable=ansible-2.8-for-rhel-8-x86_64-rpms
```

```
--enable=ansible-2.9-for-rhel-8-x86_64-rpms
  become: true
EOF
```

8. **update_rhosp_repos.yaml** Playbook を実行します。

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

すべての Red Hat OpenStack Platform ノードにコンテンツを適用するには、**--limit** オプションを使用します。Ceph Storage ノードに対してこの Playbook を実行しないでください。これらのノードには、別のサブスクリプションを使用している可能性が高いためです。

9. すべてのノードで、リポジトリを Red Hat OpenStack Platform 16.1 に設定するタスクが含まれる Playbook を作成します。

```
$ cat > ~/update_ceph_repos.yaml <<'EOF'
- hosts: all
  gather_facts: false
  tasks:
    - name: change ceph repos
      command: subscription-manager repos --disable=openstack-16-deployment-tools-for-
rhel-8-x86_64-rpms --enable=openstack-16.1-deployment-tools-for-rhel-8-x86_64-rpms --
disable=ansible-2.8-for-rhel-8-x86_64-rpms --enable=ansible-2.9-for-rhel-8-x86_64-rpms
      become: true
EOF
```

10. **update_ceph_repos.yaml** Playbook を実行します。

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

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

2.4. CONTAINER-TOOLS および VIRT モジュールバージョンの設定

すべてのノードで正しいパッケージバージョンが使用されるように、**container-tools** モジュールをバージョン **2.0** に、**virt** モジュールを **8.2** に、それぞれ設定します。

手順

1. アンダークラウドに **stack** ユーザーとしてログインします。
2. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

3. オーバークラウドの静的なインベントリーファイルを作成します。

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

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

- すべてのノードで **container-tools** モジュールをバージョン **2.0** に設定するタスクが含まれる Playbook を作成します。

```
$ cat > ~/container-tools.yaml <<'EOF'
- hosts: all
  gather_facts: false
  tasks:
    - name: disable default dnf module for container-tools
      command: dnf module disable -y container-tools:rhel8
      become: true
    - name: set dnf module for container-tools:2.0
      command: dnf module enable -y container-tools:2.0
      become: true

- hosts: undercloud,Compute,Controller
  gather_facts: false
  tasks:
    - name: disable default dnf module for virt
      command: dnf module disable -y virt:rhel
      become: true
    - name: disable 8.1 dnf module for virt
      command: dnf module disable -y virt:8.1
      become: true
    - name: set dnf module for virt:8.2
      command: dnf module enable -y virt:8.2
      become: true
EOF
```

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

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

2.5. コンテナイメージ準備ファイルの更新

コンテナ準備ファイルは、**ContainerImagePrepare** パラメーターが含まれるファイルです。このファイルを使用して、アンダークラウドおよびオーバークラウドのコンテナイメージを取得する際のルールを定義します。環境を更新する前に、ファイルを確認して正しいイメージバージョンを取得するようにしてください。

手順

- コンテナ準備ファイルを編集します。通常、このファイルのデフォルト名は **containers-prepare-parameter.yaml** です。
- それぞれのルールセットについて、**tag** パラメーターが **16.1** に設定されていることを確認します。

```
parameter_defaults:
  ContainerImagePrepare:
    - push_destination: true
      set:
        ...
        tag: '16.1'
        tag_from_label: '{version}-{release}'
```



注記

更新に特定のタグ (16.1 や 16.1.2 等) を使用しない場合は、**tag** キーと値のペアを削除し、**tag_from_label** のみを指定します。これにより、更新プロセスの一部として使用するタグの値を決定する際に、インストールされた Red Hat OpenStack Platform バージョンが使用されます。

1. このファイルを保存します。

2.6. SSL/TLS 設定の更新

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

手順

1. アンダークラウドに **stack** ユーザーとしてログインします。
2. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

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

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

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



注記

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

5. SSL/TLS パブリックエンドポイントファイルを保存します。

2.7. オーバークラウドでのフェンシングの無効化

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

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

オーバークラウドでフェンシングを有効にしている場合には、意図しない結果を防ぐために、更新期間中フェンシングを一時的に無効にする必要があります。



注記

オーバークラウドのフェンシングを再度有効にするには、**openstack overcloud upgrade prepare** コマンドの実行時に **fencing.yaml** 環境ファイルを追加します。director は、新しいコントローラーノードクラスターを作成する際に、オーバークラウドのフェンシングを有効にします。

手順

1. アンダークラウドに **stack** ユーザーとしてログインします。
2. source コマンドで **stackrc** ファイルを読み込みます。

```
$ source ~/stackrc
```

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

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

関連資料

- [「STONITH を使用したコントローラノードのフェンシング」](#)

第3章 アンダークラウドの更新

以下の手順では、アンダークラウドおよびそのオーバークラウドイメージを最新の Red Hat OpenStack Platform 16.1 バージョンに更新します。

3.1. コンテナ化されたアンダークラウドのマイナーアップデートの実施

director では、アンダークラウドノード上の主要なパッケージを更新するためのコマンドが提供されています。これにより、OpenStack Platform 環境の現行バージョン内のマイナーアップデートを実行することができます。

手順

1. director に **stack** ユーザーとしてログインします。
2. **dnf** 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 dnf update -y python3-tripleoclient* openstack-tripleo-common openstack-tripleo-heat-templates tripleo-ansible ansible
```

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

```
$ sudo reboot
```

6. ノードがブートするまで待ちます。

3.2. オーバークラウドイメージの更新

現在のオーバークラウドイメージを新しいバージョンに置き換える必要があります。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更新されていること

手順

1. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

2. **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-16.1.tar /usr/share/rhosp-director-images/ironic-python-agent-latest-16.1.tar; do tar -xvf $i; done
$ cd ~
```

4. director に最新のイメージをインポートします。

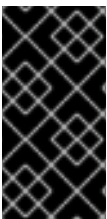
```
$ openstack overcloud image upload --update-existing --image-path /home/stack/images/
```

5. ノードが新しいイメージを使用するように設定します。

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

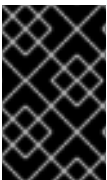
6. 新規イメージが存在することを確認します。

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



重要

オーバークラウドノードをデプロイする際には、オーバークラウドイメージのバージョンが、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 16.1 の heat テンプレートには、OpenStack Platform 16.1 のイメージのみを使用してください。



重要

新しい **overcloud-full** イメージは、古い **overcloud-full** イメージを置き換えます。古いイメージに変更を加えた場合、特に今後新規ノードをデプロイする場合には、新しいイメージで変更を繰り返す必要があります。

3.3. アンダークラウドアップグレード後の注意事項

- **stack** ユーザーのホームディレクトリーでコアテンプレートのローカルセットを使用している場合には、『[Advanced Overcloud Customization](#)』の「[Using Customized Core Heat Templates](#)」に記載の推奨ワークフローを使用して、テンプレートを更新するようにしてください。オーバークラウドをアップグレードする前に、ローカルコピーを更新する必要があります。

3.4. 次のステップ

アンダークラウドのアップグレードが完了しました。これでオーバークラウドを更新できるようになりました。

第4章 オーバークラウドの更新

以下の手順では、オーバークラウドを更新します。

前提条件

- アンダークラウドが最新バージョンに更新されていること

4.1. オーバークラウドの更新準備の実行

更新プロセスに向けてオーバークラウドを準備するには、**openstack overcloud update prepare** のコマンドを実行して、以下のタスクを実施します。

- オーバークラウドのプランを OpenStack Platform 16.1 に更新する。
- 更新に向けてノードを準備する

前提条件

- Ceph のサブスクリプションを使用し、Ceph ストレージノード用に **overcloud-minimal** イメージを使用するように **director** を設定している場合、**roles_data.yaml** ロール定義ファイルで **rism_enforce** パラメーターが **False** に設定されていることを確認する必要があります。

手順

1. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

2. 更新準備コマンドを実行します。

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

以下のオプションの中で、お使いの環境に適切なオプションを追加します。

- オーバークラウドスタックの名前がデフォルトの名前 **overcloud** とは異なる場合は、更新の準備コマンドに **--stack** オプションを追加し、**<stack_name>** を実際のスタック名に置き換えます。
- 専用のカスタムロールを使用する場合は、カスタムロール (**<roles_data>**) のファイルを追加します (**-r**)
- カスタムネットワークを使用する場合は、コンポーザブルネットワーク (**<network_data>**) のファイル_を追加します (**-n**)
- すべてのカスタム設定環境ファイル (**-e**)

3. 更新の準備が完了するまで待ちます。

4.2. コンテナイメージ準備タスクの実行

オーバークラウドの更新を実行するには、最新の OpenStack Platform 16.1 のコンテナイメージが必要です。そのためには、**container_image_prepare** 外部更新プロセスを実行します。このプロセスを実行するには、**container_image_prepare** にタグ付けされたタスクに対して **openstack overcloud external-update run** コマンドを実行します。これらのタスクにより以下のアクションが実施されます。

- お使いの環境に該当するすべてのコンテナイメージ設定を自動的に準備する。
- 事前にこのオプションを無効にしていない限り、該当するコンテナイメージをアンダークラウドにプルする。



注記

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

手順

1. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

2. **container_image_prepare** にタグ付けされたタスクに対して **openstack overcloud external-update run** コマンドを実行します。

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

4.3. 全コントローラーノードの更新

以下の手順では、すべてのコントローラーノードを最新バージョンの OpenStack Platform 16.1 に更新します。このプロセスでは、**--limit Controllerer** オプションを指定して **openstack overcloud update run** コマンドを実行し、操作をコントローラーノードだけに制限します。



重要

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



注記

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

手順

1. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

2. 更新コマンドを実行します。

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

3. コントローラーノードの更新が完了するまで待ちます。

4.4. すべてのコンピュートノードの更新

以下の手順では、すべてのコンピュートノードを最新バージョンの OpenStack Platform 16.1 に更新します。このプロセスは、**openstack overcloud update run** コマンドに **--limit Compute** オプションを指定して、操作をコンピュートノードのみに制限して実行する必要があります。

並列処理に関する考慮事項

多数のコンピュートノードをアップグレードする場合は、パフォーマンスを向上させるために、**--limit Compute** オプションを指定して **openstack overcloud upgrade run** コマンドを実行し、20 ノードのバッチを並行して処理することができます。たとえば、デプロイメントに 80 のコンピュートノードがある場合、次のコマンドを実行して、コンピュートノードを並行して更新できます。

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

'**Compute[0:19]**'、'**Compute[20:39]**'、'**Compute[40:59]**'、および '**Compute[60:79]**' のノード領域分割方法はランダムで、更新されるノードを制御することはできません。

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

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



注記

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

手順

1. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

2. 更新コマンドを実行します。

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


3. コンピュートノードの更新が完了するまで待ちます。

4.5. 全 HCI コンピュートノードの更新

以下の手順では、ハイパーコンバージドインフラストラクチャー (HCI) コンピュートノードを更新します。このプロセスには、以下の操作が必要です。

- **openstack overcloud update run** コマンドに **--nodes ComputeHCI** オプションを指定して、操作を HCI ノードのみに制限して実行する
- **openstack overcloud external-update run --tags ceph** コマンドを実行し、コンテナ化された Red Hat Ceph Storage 4 クラスタへの更新を実施する



注記

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

手順

1. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

2. 更新コマンドを実行します。

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

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

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

5. コンピュート HCI ノードの更新が完了するまで待ちます。

4.6. すべての CEPH STORAGE ノードの更新

以下の手順では、Ceph Storage ノードを更新します。このプロセスでは、以下の操作を行います。

- **openstack overcloud update run** コマンドに **--limit CephStorage** オプションを指定して、操作を Ceph Storage ノードのみに制限して実行する
- **openstack overcloud external-update run** コマンドを実行して外部プロセスとして **ceph-ansible** を実行し、Red Hat Ceph Storage 3 コンテナを更新する。



注記

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

手順

1. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

2. 更新コマンドを実行します。

```
$ openstack overcloud update run --stack <stack_name> --limit CephStorage --playbook all
```

3. ノードの更新が完了するまで待ちます。

4. Ceph Storage コンテナの更新コマンドを実行します。

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

5. Ceph Storage コンテナの更新が完了するまで待ちます。

4.7. データベースのオンライン更新の実施

一部のオーバークラウドコンポーネントでは、データベーステーブルのオンラインアップグレード(または移行)が必要です。そのためには、**online_upgrade** 外部更新プロセスを実行します。このプロセスを実行するには、**online_upgrade** にタグ付けされたタスクに対して **openstack overcloud external-update run** コマンドを実行します。これにより、以下のコンポーネントに対してデータベースのオンライン更新が実行されます。

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

手順

1. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

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

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

4.8. 更新の最終処理

更新には、オーバークラウドスタックを更新する最終ステップが必要です。これにより、スタックのソース構造が OpenStackPlatform 16.1 の標準のデプロイメントと一致し、今後、通常の **openstack overcloud deploy** の機能を実行できるようになります。

手順

1. **stackrc** ファイルを取得します。

```
$ source ~/stackrc
```

2. 更新の最終処理のコマンドを実行します。

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

以下のオプションの中で、お使いの環境に適切なオプションを追加します。

- オーバークラウドスタックの名前がデフォルトの名前 **overcloud** とは異なる場合は、更新の準備コマンドに **--stack** オプションを追加し、**<stack_name>** を実際のスタック名に置き換えます。
- 専用のカスタムロールを使用する場合は、カスタムロール (**<roles_data>**) のファイルを追加します (**-r**)
- カスタムネットワークを使用する場合は、コンポーザブルネットワーク (**<network_data>**) のファイルを追加します (**-n**)
- すべてのカスタム設定環境ファイル (**-e**)

3. 更新の最終処理が完了するまで待ちます。

第5章 オーバークラウドのリブート

Red Hat OpenStack バージョンのマイナーアップデート後に、オーバークラウドをリブートします。リブートにより、関連付けられたカーネル、システムレベル、およびコンテナコンポーネントの更新と共にノードがリフレッシュされます。これらの更新により、パフォーマンスとセキュリティ上のメリットが得られます。

ダウンタイムを計画して、以下のリブート手順を実施します。

5.1. コントローラーノードおよびコンポーザブルノードのリブート

コントローラーノードおよびコンポーザブルロールに基づくスタンドアロンのノードをリブートするには、以下の手順を実施します。ただし、これにはコンピュータードおよび Ceph Storage ノードは含まれません。

手順

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

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

3. ノードをリブートします。

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

4. ノードがブートするまで待ちます。
5. サービスを確認します。以下に例を示します。
 - a. ノードが Pacemaker サービスを使用している場合には、ノードがクラスターに再度加わったかどうかを確認します。

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

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

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

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

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

5.2. CEPH STORAGE (OSD) クラスターのリブート

Ceph Storage (OSD) ノードのクラスターをリブートするには、以下の手順を実施します。

手順

1. Ceph MON またはコントローラーノードにログインして、Ceph Storage クラスターのリバランスを一時的に無効にします。

```
$ sudo podman exec -it ceph-mon-controller-0 ceph osd set noout
$ sudo podman exec -it ceph-mon-controller-0 ceph osd set norebalance
```

2. リブートする最初の Ceph Storage ノードを選択し、そのノードにログインします。
3. ノードをリブートします。

```
$ sudo reboot
```

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

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

pgmap により、すべての **pgs** が正常な状態 (**active+clean**) として報告されることを確認します。

6. ノードからログアウトして、次のノードをリブートし、ステータスを確認します。全 Ceph Storage ノードがリブートされるまで、このプロセスを繰り返します。
7. 完了したら、Ceph MON またはコントローラーノードにログインして、クラスターのリバランスを再度有効にします。

```
$ sudo podman exec -it ceph-mon-controller-0 ceph osd unset noout
$ sudo podman exec -it ceph-mon-controller-0 ceph osd unset norebalance
```

8. 最終のステータスチェックを実行して、クラスターが **HEALTH_OK** を報告していることを確認します。

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

5.3. コンピュートノードのリブート

コンピュートノードをリブートするには、以下の手順を実施します。Red Hat OpenStack Platform 環境内のインスタンスのダウンタイムを最小限に抑えるために、この手順には、リブートするコンピュートノードからインスタンスを移行するステップも含まれています。これは、以下のワークフローを伴います。

- コンピュートノードをリブートする前に、インスタンスを別のノードに移行するかどうかを決定する。
- リブートするコンピュートノードを選択して無効にし、新規インスタンスをプロビジョニングしないようにする。
- インスタンスを別のコンピュートノードに移行する。
- 空のコンピュートノードをリブートする。
- 空のコンピュートノードを有効にする。

前提条件

コンピューターノードをリブートする前に、ノードをリブートする間インスタンスを別のコンピューターノードに移行するかどうかを決定する必要があります。

コンピューターノード間で仮想マシンインスタンスを移行する際に受ける可能性のある移行の制約の一覧を確認してください。詳細については、『[インスタンス作成のための Compute サービスの設定](#)』の「[移行の制約](#)」を参照してください。

インスタンスを移行することができない場合は、以下のコアテンプレートパラメーターを設定して、コンピューターノードリブート後のインスタンスの状態を制御することができます。

NovaResumeGuestsStateOnHostBoot

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

NovaResumeGuestsShutdownTimeout

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

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

手順

1. アンダークラウドに **stack** ユーザーとしてログインします。
2. 全コンピューターノードとその UUID を一覧表示します。

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

リブートするコンピューターノードの UUID を特定します。

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

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

4. コンピューターノード上の全インスタンスを一覧表示します。

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

5. インスタンスを移行しない場合は、[このステップ](#)に進みます。
6. インスタンスを別のコンピューターノードに移行する場合には、以下のコマンドのいずれかを使用します。

- インスタンスを別のホストに移行する。

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

- **nova-scheduler** により対象のホストが自動的に選択されるようにする。

-

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

- 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う。

```
$ 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、無視して問題ありません。

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

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

9. 選択したコンピューターノードのインスタンスがなくなるまで、移行を続けます。
10. コンピューターノードにログインして、ノードをリブートします。

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

11. ノードがブートするまで待ちます。
12. コンピューターノードを再度有効にします。

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

13. コンピューターノードが有効であることを確認します。

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