Red Hat OpenStack Platform 10

Upgrading Red Hat OpenStack Platform

Upgrading a Red Hat OpenStack Platform environment

Last Updated: 2021-05-21

Red Hat OpenStack Platform 10 Upgrading Red Hat OpenStack Platform

Upgrading a Red Hat OpenStack Platform environment

OpenStack Team
rhos-docs@redhat.com

Legal Notice

Abstract

This document lays out the different methods through which users can upgrade from Red Hat OpenStack Platform 9 (Mitaka) to 10 (Newton). These methods assume that you will be upgrading to and from an OpenStack deployment installed on Red Hat Enterprise Linux 7.

Table of Contents

CHAPTER 1. INTRODUCTION

This document provides processes for keeping Red Hat OpenStack Platform up-to-date. This document focuses on upgrades and updates that targets **Red Hat OpenStack Platform 10 (Newton)**

Red Hat only supports upgrades to Red Hat OpenStack Platform 10 on Red Hat Enterprise Linux 7.3. In addition, Red Hat recommends the following different scenarios based on whether:

- You are using the director-based Overcloud or a manually created environment.
- You are using high availability tools to manage a set of Controller nodes in a cluster.

The Section 1.1, “Upgrade Scenario Comparison” provides descriptions of all upgrade scenarios. These scenarios allow you to upgrade to a working Red Hat OpenStack Platform 10 release and provide minor updates within that version.

1.1. UPGRADE SCENARIO COMPARISON

Red Hat recommends the following upgrade scenarios for Red Hat OpenStack Platform 10. The following table provides a brief description of each.

WARNING

Do not upgrade to the Red Hat Enterprise Linux 7.3 kernel without also upgrading from Open vSwitch (OVS) 2.4.0 to OVS 2.5.0. If only the kernel is upgraded, then OVS will stop functioning.

Table 1.1. Upgrade Scenarios

Method	Description
Director-Based Environments: Performing Updates to Minor Versions	This scenario is for updating from one minor version of Red Hat OpenStack Platform 10 to a newer version of Red Hat OpenStack Platform 10. This involves updating the director packages, then using the director to launch a package update on all nodes in the Overcloud.
Director-Based Environments: Performing Upgrades to Major Versions	This scenario is for upgrading from a major versions of Red Hat OpenStack Platform. In this case, the procedure upgrades from version 9 to version 10. This involves updating the director packages, then using the director to provide a set of upgrade scripts on each node, and then performing an upgrade of the Overcloud stack.

Method	Description
Non-Director Environments: Upgrading OpenStack Services Simultaneously	This scenario is for upgrading all packages in a Red Hat OpenStack Platform environment that does not use the director for management (i.e. environments created manually). In this scenario, all packages are upgraded simultaneously.
Non-Director Environments: Upgrading Individual OpenStack Services (Live Compute) in a Standard Environment	This scenario is for upgrading all packages in a Red Hat OpenStack Platform environment that does not use the director for management (i.e. environments created manually). In this scenario, you update each OpenStack service individually.
Non-Director Environments: Upgrading Individual OpenStack Services (Live Compute) in a High Availability Environment	This scenario is for upgrading all packages in a Red Hat OpenStack Platform environment that does not use the director for management (i.e. environments created manually) and are using high availability tools for Controller-based OpenStack services. In this scenario, you update each OpenStack service individually.

For all methods:

- Ensure you have enabled the correct repositories for this release on all hosts.
- The upgrade will involve some service interruptions.
- Running instances will not be affected by the upgrade process unless you either reboot a Compute node or explicitly shut down an instance.

WARNING

Red Hat does not support upgrading any Beta release of Red Hat OpenStack Platform to any supported release.

1.2. REPOSITORY REQUIREMENTS

Both the undercloud and overcloud require access to Red Hat repositories either through the Red Hat Content Delivery Network, or through Red Hat Satellite 5 or 6. If using a Red Hat Satellite Server, synchronize the required repositories to your OpenStack Platform environment. Use the following list of CDN channel names as a guide:

Table 1.2. OpenStack Platform Repositories

Name	Repository	Description of Requirement

Red Hat Enterprise Linux 7 Server (RPMs)	**rhel-7-server-rpms**	Base operating system repository.
Red Hat Enterprise Linux 7 Server - Extras (RPMs)	**rhel-7-server-extras-rpms**	Contains Red Hat OpenStack Platform dependencies.
Red Hat Enterprise Linux 7 Server - RH Common (RPMs)	**rhel-7-server-rh-common-rpms**	Contains tools for deploying and configuring Red Hat OpenStack Platform.
Red Hat Satellite Tools for RHEL 7 Server RPMs x86_64	**rhel-7-server-satellite-tools-6.2-rpms**	Tools for managing hosts with Red Hat Satellite 6.
Red Hat Enterprise Linux High Availability (for RHEL 7 Server) (RPMs)	**rhel-ha-for-rhel-7-server-rpms**	High availability tools for Red Hat Enterprise Linux. Used for Controller node high availability.
Red Hat Enterprise Linux OpenStack Platform 10 for RHEL 7 (RPMs)	**rhel-7-server-openstack-10-rpms**	Core Red Hat OpenStack Platform repository. Also contains packages for Red Hat OpenStack Platform director.
Red Hat Ceph Storage OSD 2 for Red Hat Enterprise Linux 7 Server (RPMs)	**rhel-7-server-rhceph-2-osd-rpms**	(For Ceph Storage Nodes) Repository for Ceph Storage Object Storage daemon. Installed on Ceph Storage nodes.
Red Hat Ceph Storage MON 2 for Red Hat Enterprise Linux 7 Server (RPMs)	**rhel-7-server-rhceph-2-mon-rpms**	(For Ceph Storage Nodes) Repository for Ceph Storage Monitor daemon. Installed on Controller nodes in OpenStack environments using Ceph Storage nodes.
Red Hat Ceph Storage Tools 2 for Red Hat Enterprise Linux 7 Server (RPMs)	**rhel-7-server-rhceph-2-tools-rpms**	Provides tools for nodes to communicate with the Ceph Storage cluster. This repository should be enabled for all overcloud nodes when deploying an overcloud with a Ceph Storage cluster.

NOTE

To configure repositories for your Red Hat OpenStack Platform environment in an offline network, see "Configuring Red Hat OpenStack Platform Director in an Offline Environment" on the Red Hat Customer Portal.

1.3. BEFORE STARTING THE UPGRADE

Apply any firmware updates to your hardware before performing the upgrade.

CHAPTER 2. DIRECTOR-BASED ENVIRONMENTS: PERFORMING UPDATES TO MINOR VERSIONS

This section explores how to update packages for your Red Hat OpenStack Platform environment within the same version. In this case, it is updates within Red Hat OpenStack Platform 10. This includes updating aspects of both the Undercloud and Overcloud.

Prerequisites

- Before you can perform a minor update on OSP10, you must edit the **storage-environment.yaml** file and set the UUID in **NodeDataLookup** to lowercase. For more information, see the Red Hat Knowledgebase solution minor update on OSP10 fails on ceph .

WARNING

With High Availability for Compute instances (or Instance HA, as described in High Availability for Compute Instances), upgrades or scale-up operations are not possible. Any attempts to do so will fail.

If you have Instance HA enabled, disable it before performing an upgrade or scale-up. To do so, perform a *rollback* as described in Rollback.

This procedure for both situations involves the following workflow:

1. Update the Red Hat OpenStack Platform director packages
2. Update the Overcloud images on the Red Hat OpenStack Platform director
3. Update the Overcloud packages using the Red Hat OpenStack Platform director

2.1. PRE-UPDATE NOTES

2.1.1. General Recommendations

Before performing the update, Red Hat advises the following:

- Perform a backup of your Undercloud node before starting any steps in the update procedure. See the Back Up and Restore the Director Undercloud guide for backup procedures.
- Red Hat OpenStack Platform 10 ELS is only supported on Red Hat Enterprise Linux (RHEL) 7.7, therefore, it is important that you set the RHEL version to RHEL 7.7 before you perform an update.
- Before you can perform a minor update on OSP10, you must edit the **storage-environment.yaml** file and set the UUID in **NodeDataLookup** to lowercase. For more information, see the Red Hat Knowledgebase solution minor update on OSP10 fails on ceph .
- Run the update procedure in a test environment that includes all of the changes made before running the procedure in your production environment.

- If necessary, please contact Red Hat and request any guidance and assistance for performing an update.

2.1.2. NFV Pre-Configuration

An Overcloud with Network Functions Virtualization (NFV) enabled requires some additional preparation to accommodate any updates to the Open vSwitch (OVS) package. To support this transition when you have OVS-DPDK configured, follow these guidelines.

> **NOTE**
>
> Red Hat OpenStack Platform 10 with OVS 2.9 operates in OVS client mode for OVS-DPDK deployments.

1. Change the vhost user socket directory in a custom environment file, for example, **network-environment.yaml**:

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

2. Add the **ovs-dpdk-permissions.yaml** file to your **openstack overcloud deploy** command to configure the qemu group setting as **hugetlbfs** for OVS-DPDK:

   ```
    -e /usr/share/openstack-tripleo-heat-templates/environments/ovs-dpdk-permissions.yaml
   ```

2.2. UPDATING RED HAT OPENSTACK PLATFORM

2.2.1. Updating the Undercloud Packages

The director relies on standard RPM methods to update your environment. This involves ensuring your director’s host uses the latest packages through **yum**.

1. Log into the director as the **stack** user.

2. Stop the main OpenStack Platform services:

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

   > **NOTE**
   >
   > This causes a short period of downtime for the undercloud. The overcloud is still functional during the undercloud update.

3. Set the RHEL version to RHEL 7.7:

   ```
   $ sudo subscription-manager release --set=7.7
   ```

4. Update the **python-tripleoclient** package and its dependencies to ensure you have the latest scripts for the minor version update:

   ```
   $ sudo yum update python-tripleoclient
   ```

5. The director uses the **openstack undercloud upgrade** command to update the Undercloud environment. Run the command:

    ```
    $ openstack undercloud upgrade
    ```

Major and minor version updates to the kernel or Open vSwitch require a reboot, such as when your undercloud operating system updates from Red Hat Enterprise Linux 7.2 to 7.3, or Open vSwitch from version 2.4 to 2.5. Check the **/var/log/yum.log** file on the director node to see if either the **kernel** or **openvswitch** packages have updated their major or minor versions. If they have, perform a reboot of each node:

1. Reboot the node:

    ```
    $ sudo reboot
    ```

2. Wait until the node boots.

3. When the node boots, check the status of all services:

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

    NOTE

    It might take approximately 10 minutes for the **openstack-nova-compute** to become active after a reboot.

4. Verify the existence of your Overcloud and its nodes:

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

It is important to keep your overcloud images up to date to ensure the image configuration matches the requirements of the latest **openstack-tripleo-heat-template** package. To ensure successful deployments and scaling operations in the future, update your overclouds images using the instructions in Section 2.2.2, “Updating the Overcloud Images”.

2.2.2. Updating the Overcloud Images

The Undercloud update process might download new image archives from the **rhosp-director-images** and **rhosp-director-images-ipa** packages. Check the **yum** log to determine if new image archives are available:

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

If new archives are available, replace your current images with new images. To install the new images, first remove any existing images from the **images** directory on the **stack** user’s home (**/home/stack/images**):

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

Extract the archives:

```
$ cd ~/images
$ for i in /usr/share/rhosp-director-images/overcloud-full-latest-10.0.tar /usr/share/rhosp-director-images/ironic-python-agent-latest-10.0.tar; do tar -xvf $i; done
```

Import the latest images into the director and configure nodes to use the new images

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

To finalize the image update, verify the existence of the new images:

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

The director is now updated and using the latest images. You do not need to restart any services after the update.

2.2.3. Updating the Overcloud Packages

The Overcloud relies on standard RPM methods to update the environment. This involves the following steps:

1. Check your subscription management configuration for the **rhel_reg_release** parameter. If this parameter is not set, you must include it and set it to version 7.7:

   ```
   parameter_defaults:
     ...
     rhel_reg_release: "7.7"
   ```

   Ensure that you save the changes to the overcloud subscription management environment file.

2. Update the current plan using your original **openstack overcloud deploy** command and including the **--update-plan-only** option. For example:

   ```
   $ 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>|...]
   ```

   The **--update-plan-only** only updates the Overcloud plan stored in the director. Use the **-e** to include environment files relevant to your Overcloud and its update path. The order of the environment files is important as the parameters and resources defined in subsequent environment files take precedence. Use the following list as an example of the environment file order:

   - Any network isolation files, including the initialization file (**environments/network-isolation.yaml**) from the heat template collection and then your custom NIC configuration file.

- Any external load balancing environment files.
- Any storage environment files.
- Any environment files for Red Hat CDN or Satellite registration.
- Any other custom environment files.

3. Perform a package update on all nodes using the **openstack overcloud update** command. For example:

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

   Running an update on all nodes in parallel might cause problems. For example, an update of a package might involve restarting a service, which can disrupt other nodes. This is why the process updates each node using a set of breakpoints. This means nodes are updated one by one. When one node completes the package update, the update process moves to the next node. The update process also requires the **-i** option, which puts the command in an interactive mode that requires confirmation at each breakpoint. Without the **-i** option, the update remains paused at the first breakpoint.

This starts update process. During this process, the director reports an **IN_PROGRESS** status and periodically prompts you to clear breakpoints. For example:

```
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:
```

Press Enter to clear the breakpoint from last node on the **on_breakpoint** list. This begins the update for that node. You can also type a node name to clear a breakpoint on a specific node, or a Python-based regular expression to clear breakpoints on multiple nodes at once. However, it is not recommended to clear breakpoints on multiple controller nodes at once. Continue this process until all nodes have completed their update.

The update command reports a **COMPLETE** status when the update completes:

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

If you configured fencing for your Controller nodes, the update process might disable it. When the update process completes, reenable fencing with the following command on one of the Controller nodes:

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

The update process does not reboot any nodes in the Overcloud automatically. Major and minor version updates to the kernel or Open vSwitch require a reboot, such as when your overcloud operating system updates from Red Hat Enterprise Linux 7.2 to 7.3, or Open vSwitch from version 2.4 to 2.5. Check the

/var/log/yum.log file on each node to see if either the **kernel** or **openvswitch** packages have updated their major or minor versions. If they have, perform a reboot of each node using the "Rebooting the Overcloud" procedures in the *Director Installation and Usage* guide.

2.3. POST-UPDATE NOTES

2.3.1. Sshd Composable Service

The latest update of Red Hat OpenStack Platform 10 includes the **OS::TripleO::Services::Sshd** composable service, which is required for live migration capabilities. The director's core template collection did not include this service in the initial release but is now included in the **openstack-tripleo-heat-templates-5.2.0-12** package and later versions.

The default roles data file includes this service and the director installs the service on the overcloud on update.

If using a custom roles data file, include the **OS::TripleO::Services::Sshd** service on each overcloud role, then update your overcloud stack to include the new service.

For more information, see "Red Hat OpenStack Platform director (TripleO) CVE-2017-2637 bug and Red Hat OpenStack Platform ".

2.3.2. NFV Post-Configuration

If your Overcloud uses Network Functions Virtualization (NFV), follow this procedure to finish the update.

Procedure

You need to migrate your existing OVS-DPDK instances to ensure that the vhost socket mode changes from **dkdpvhostuser** to **dkdpvhostuserclient** mode in the OVS ports. We recommend that you snapshot existing instances and rebuild a new instance based on that snapshot image. See Manage Instance Snapshots for complete details on instance snapshots.

To snapshot an instance and boot a new instance from the snapshot:

1. Find the server ID for the instance you want to take a snapshot of:

   ```
   # openstack server list
   ```

2. Shut down the source instance before you take the snapshot to ensure that all data is flushed to disk:

   ```
   # openstack server stop SERVER_ID
   ```

3. Create the snapshot image of the instance:

   ```
   # openstack image create --id SERVER_ID SNAPSHOT_NAME
   ```

4. Boot a new instance with this snapshot image:

   ```
   # openstack server create --flavor DPDK_FLAVOR --nic net-id=DPDK_NET_ID --image
   SNAPSHOT_NAME INSTANCE_NAME
   ```

5. Optionally, verify that the new instance status is **ACTIVE**:

   ```
   # openstack server list
   ```

Repeat this procedure for all instances that you need to snapshot and relaunch.

CHAPTER 3. DIRECTOR-BASED ENVIRONMENTS: PERFORMING UPGRADES TO MAJOR VERSIONS

WARNING

Before performing an upgrade to the latest major version, ensure the undercloud and overcloud are updated to the latest minor versions. This includes both OpenStack Platform services and the base operating system. For the process on performing a minor version update, see "Director-Based Environments: Performing Updates to Minor Versions" in the Red Hat OpenStack Platform 9 *Upgrading Red Hat OpenStack Platform* guide. Performing a major version upgrade without first performing a minor version update can cause failures in the upgrade process.

WARNING

With High Availability for Compute instances (or Instance HA, as described in High Availability for Compute Instances), upgrades or scale-up operations are not possible. Any attempts to do so will fail.

If you have Instance HA enabled, disable it before performing an upgrade or scale-up. To do so, perform a *rollback* as described in Rollback.

This chapter explores how to upgrade your environment. This includes upgrading aspects of both the Undercloud and Overcloud. This upgrade process provides a means for you to move to the next major version. In this case, it is a upgrade from Red Hat OpenStack Platform 9 to Red Hat OpenStack Platform 10.

This procedure for both situations involves the following workflow:

1. Upgrade the Red Hat OpenStack Platform director packages
2. Upgrade the Overcloud images on the Red Hat OpenStack Platform director
3. Upgrade the Overcloud stack and its packages using the Red Hat OpenStack Platform director

3.1. UPGRADE SUPPORT STATEMENT

A successful upgrade process requires some preparation to accommodate changes from one major version to the next. Read the following support statement to help with Red Hat OpenStack Platform upgrade planning.

Upgrades in Red Hat OpenStack Platform director require full testing with specific configurations before performed on any live production environment. Red Hat has tested most use cases and combinations offered as standard options through the director. However, due to the number of possible combinations, this is never a fully exhaustive list. In addition, if the configuration has been modified from the standard deployment, either manually or through post configuration hooks, testing upgrade features in a non-production environment is critical. Therefore, we advise you to:

- Perform a backup of your Undercloud node before starting any steps in the upgrade procedure. See the Back Up and Restore the Director Undercloud guide for backup procedures.

- Run the upgrade procedure with your customizations in a test environment before running the procedure in your production environment.

- If you feel uncomfortable about performing this upgrade, contact Red Hat's support team and request guidance and assistance on the upgrade process before proceeding.

The upgrade process outlined in this section only accommodates customizations through the director. If you customized an Overcloud feature outside of director then:

- Disable the feature

- Upgrade the Overcloud

- Re-enable the feature after the upgrade completes

This means the customized feature is unavailable until the completion of the entire upgrade.

Red Hat OpenStack Platform director 10 can manage previous Overcloud versions of Red Hat OpenStack Platform. See the support matrix below for information.

Table 3.1. Support Matrix for Red Hat OpenStack Platform director 10

Version	Overcloud Updating	Overcloud Deploying	Overcloud Scaling
Red Hat OpenStack Platform 10	Red Hat OpenStack Platform 9 and 10	Red Hat OpenStack Platform 9 and 10	Red Hat OpenStack Platform 9 and 10

If managing an older Overcloud version, use the following Heat template collections: * For Red Hat OpenStack Platform 9: **/usr/share/openstack-tripleo-heat-templates/mitaka/**

For example:

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

The following are some general upgrade tips:

- After each step, run the **pcs status** command on the Controller node cluster to ensure no resources have failed.

- Please contact Red Hat and request guidance and assistance on the upgrade process before proceeding if you feel uncomfortable about performing this upgrade.

3.2. IMPORTANT PRE-UPGRADE NOTES

- Red Hat OpenStack Platform 10 uses some new kernel parameters now available in Red Hat Enterprise Linux 7.3. Make sure that you have upgraded your undercloud and overcloud to Red Hat Enterprise Linux 7.3 and Open vSwitch 2.5. See "Director-Based Environments: Performing

Updates to Minor Versions" for instructions on performing a package update to your undercloud and overcloud. When you have updated the kernel to the latest version, perform a reboot so that the new kernel parameters take effect.

- The OpenStack Platform 10 upgrade procedure migrates to a new composable architecture. This means many services that Pacemaker managed in previous versions now use **systemd** management. This results in a reduced number of Pacemaker managed resources.

- In previous versions of Red Hat OpenStack Platform, OpenStack Telemetry (ceilometer) used its database for metrics storage. Red Hat OpenStack Platform 10 uses OpenStack Telemetry Metrics (gnocchi) as a default backend for OpenStack Telemetry. If using an external Ceph Storage cluster for the metrics data, create a new pool on your external Ceph Storage cluster before upgrading to Red Hat OpenStack 10. The name for this pool is set with the **GnocchiRbdPoolName** parameter and the default pool name is **metrics**. If you use the **CephPools** parameter to customize your list of pools, add the metrics pool to the list. Note that there is no data migration plan for the metrics data. For more information, see Section 3.5.4, “OpenStack Telemetry Metrics”.

- Combination alarms in OpenStack Telemetry Alarms (aodh) are deprecated in favor of composite alarms. Note that:

  - Aodh does not expose combination alarms by default.

  - A new parameter, **EnableCombinationAlarms**, enables combination alarms in an Overcloud. This defaults to **false**. Set to **true** to continue using combination alarms in OpenStack Platform 10.

  - OpenStack Platform 10 includes a migration script (**aodh-data-migration**) to move to composite alarms. This guide contains instructions for migrating this data in Section 3.6.9, “Migrating the OpenStack Telemetry Alarming Database”. Make sure to run this script and convert your alarms to composite.

  - Combination alarms support will be removed in the next release.

3.3. CHECKING THE OVERCLOUD

Check your overcloud is stable before performing the upgrade. Run the following steps on the director to ensure all services in your overcloud are running:

1. Check the status of the high availability services:

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

   Replace **[CONTROLLER_IP]** with the IP address of a Controller node. This command refreshes the overcloud’s Pacemaker cluster, waits 60 seconds, then reports the status of the cluster.

2. Check for any failed OpenStack Platform **systemd** services on overcloud nodes. The following command checks for failed services on all nodes:

   ```
   $ 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. Check that **os-collect-config** is running on each node. The following command checks this service on each node:

```
$ 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
```

3.4. UNDERCLOUD UPGRADE

3.4.1. Upgrading the Director

To upgrade the Red Hat OpenStack Platform director, follow this procedure:

1. Log into the director as the **stack** user.

2. Update the OpenStack Platform repository:

   ```
   $ sudo subscription-manager repos --disable=rhel-7-server-openstack-9-rpms --disable=rhel-7-server-openstack-9-director-rpms
   $ sudo subscription-manager repos --enable=rhel-7-server-openstack-10-rpms
   ```

   This sets **yum** to use the latest repositories.

3. Stop the main OpenStack Platform services:

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

   NOTE

   This causes a short period of downtime for the undercloud. The overcloud is still functional during the undercloud upgrade.

4. Use **yum** to upgrade the director:

   ```
   $ sudo yum update python-tripleoclient
   ```

5. Use the following command to upgrade the undercloud:

   ```
   $ openstack undercloud upgrade
   ```

   This command upgrades the director's packages, refreshes the director's configuration, and populates any settings that are unset since the version change. This command does not delete any stored data, such Overcloud stack data or data for existing nodes in your environment.

Review the resulting configuration files for each service. The upgraded packages might have installed **.rpmnew** files appropriate to the Red Hat OpenStack Platform 10 version of each service.

Check the **/var/log/yum.log** file on the undercloud node to see if either the **kernel** or **openvswitch** packages have updated their major or minor versions. If they have, perform a reboot of the undercloud:

1. Reboot the node:

   ```
   $ sudo reboot
   ```

2. Wait until the node boots.

3. When the node boots, check the status of all services:

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

NOTE

It might take approximately 10 minutes for the **openstack-nova-compute** to become active after a reboot.

4. Verify the existence of your Overcloud and its nodes:

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

IMPORTANT

If using customized core Heat templates, make sure to check for differences between the updated core Heat templates and your current set. Red Hat provides updates to the Heat template collection over subsequent releases. Using a modified template collection can lead to a divergence between your custom copy and the original copy in **/usr/share/openstack-tripleo-heat-templates**. Run the following command to see differences between your custom Heat template collection and the updated original version:

```
# diff -Nar /usr/share/openstack-tripleo-heat-templates/ ~/templates/my-overcloud/
```

Make sure to either apply these updates to your custom Heat template collection, or create a new copy of the templates in **/usr/share/openstack-tripleo-heat-templates/** and apply your customizations.

3.4.2. Upgrading the Overcloud Images on the Director

This procedure ensures you have the latest images for node discovery and Overcloud deployment. The new images from the **rhosp-director-images** and **rhosp-director-images-ipa** packages are already updated from the Undercloud upgrade.

Remove any existing images from the **images** directory on the **stack** user's home (**/home/stack/images**):

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

Extract the archives:

```
$ cd ~/images
$ for i in /usr/share/rhosp-director-images/overcloud-full-latest-10.0.tar /usr/share/rhosp-director-images/ironic-python-agent-latest-10.0.tar; do tar -xvf $i; done
```

Import the latest images into the director and configure nodes to use the new images

```
$ openstack overcloud image upload --update-existing --image-path /home/stack/images/.
$ openstack baremetal configure boot
```

To finalize the image update, verify the existence of the new images:

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

The director is now upgraded with the latest images.

IMPORTANT

Make sure the Overcloud image version corresponds to the Undercloud version.

3.4.3. Using and Comparing Previous Template Versions

The upgrade process installs a new set of core Heat templates that correspond to the latest overcloud version. Red Hat OpenStack Platform’s repository retains the previous version of the core template collection in the **openstack-tripleo-heat-templates-compat** package. You install this package with the following command:

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

This installs the previous templates in the **compat** directory of your Heat template collection (**/usr/share/openstack-tripleo-heat-templates/compat**) and also creates a link to **compat** named after the previous version (**mitaka**). These templates are backwards compatible with the upgraded director, which means you can use the latest version of the director to install an overcloud of the previous version.

Comparing the previous version with the latest version helps identify changes to the overcloud during the upgrade. If you need to compare the current template collection with the previous version, use the following process:

1. Create a temporary copy of the core Heat templates:

   ```
   $ cp -a /usr/share/openstack-tripleo-heat-templates /tmp/osp10
   ```

2. Move the previous version into its own directory:

   ```
   $ mv /tmp/osp10/compat /tmp/osp9
   ```

3. Perform a **diff** on the contents of both directories:

   ```
   $ diff -urN /tmp/osp9 /tmp/osp10
   ```

This shows the core template changes from one version to the next. These changes provide an idea of what should occur during the overcloud upgrade.

3.5. OVERCLOUD PRE-UPGRADE CONFIGURATION

3.5.1. Red Hat Subscription Details

If using an environment file for Satellite registration, make sure to update the following parameters in the environment file:

- **rhel_reg_repos** - Repositories to enable for your Overcloud, including the new Red Hat OpenStack Platform 10 repositories. See Section 1.2, “Repository Requirements” for repositories to enable.
- **rhel_reg_activation_key** - The new activation key for your Red Hat OpenStack Platform 10 repositories.
- **rhel_reg_sat_repo** - A new parameter that defines the repository containing Red Hat Satellite 6’s management tools, such as **katello-agent**. Make sure to add this parameter if registering to Red Hat Satellite 6.

3.5.2. SSL Configuration

If upgrading an overcloud that uses SSL, be aware of the following:

- The network configuration requires a **PublicVirtualFixedIPs** parameter in the following format:

  ```
  PublicVirtualFixedIPs: [{'ip_address':'192.168.200.180'}]
  ```

  Include this in the **parameter_defaults** section of your network environment file.
- A new environment file with SSL endpoints. This file depends on whether accessing the overcloud through IP addresses or DNS.
  - If using IP addresses, use **/usr/share/openstack-tripleo-heat-templates/environments/tls-endpoints-public-ip.yaml**.
  - If using DNS, use **/usr/share/openstack-tripleo-heat-templates/environments/tls-endpoints-public-dns.yaml**.
- For more information about SSL/TLS configuration, see "Enabling SSL/TLS on the Overcloud" in the *Red Hat OpenStack Platform Advanced Overcloud Customization* guide.

3.5.3. Ceph Storage

If you are using a custom **storage-environment.yaml** file, check that the **resource_registry** section includes the following new resources:

```
resource_registry:
  OS::TripleO::Services::CephMon: /usr/share/openstack-tripleo-heat-templates/puppet/services/ceph-mon.yaml
  OS::TripleO::Services::CephOSD: /usr/share/openstack-tripleo-heat-templates/puppet/services/ceph-osd.yaml
  OS::TripleO::Services::CephClient: /usr/share/openstack-tripleo-heat-templates/puppet/services/ceph-client.yaml
```

You must also add the following Ceph Storage parameters to make sure that RBD clients work with previous Ceph Storage software releases.

```
parameter_defaults:
  ExtraConfig:
    ceph::conf::args:
```

```
client/rbd_default_features:
  value: "3"
```

These resources ensure the Ceph Storage composable services are enabled for Red Hat OpenStack Platform 10. The default **storage-environment.yaml** file for Red Hat OpenStack Platform 10 is now updated to include these resources.

3.5.4. OpenStack Telemetry Metrics

Red Hat OpenStack Platform 10 introduces a new component to store metrics data. If using a Red Hat Ceph Storage cluster deployed with a custom **storage-environment.yaml** file, check the file’s **parameters_default** section for the following new parameters:

- **GnocchiBackend** - The backend to use. Set this **rbd** (Ceph Storage). Other options include **swift** or **file**.

- **GnocchiRbdPoolName** - The name of the Ceph Storage pool to use for metrics data. The default is **metrics**.

If using an external Ceph Storage cluster (i.e. one not managed with director), you must manually add the pool defined in **GnocchiRbdPoolName** (for example, the default is **metrics**) before performing the upgrade.

3.5.5. Overcloud Parameters

Note the following information about overcloud parameters for upgrades:

- The default timezone for Red Hat OpenStack Platform 10 is UTC. If necessary, include an environment file to specify the timezone.

- If upgrading an Overcloud with a custom **ServiceNetMap**, ensure to include the latest **ServiceNetMap** for the new services. The default list of services is defined with the **ServiceNetMapDefaults** parameter located in the **network/service_net_map.j2.yaml** file. For information on using a custom **ServiceNetMap**, see Isolating Networks in *Advanced Overcloud Customization*.

- Due to the new composable service architecture, the parameters for configuring the NFS backend for OpenStack Image Storage (Glance) have changed. The new parameters are:

  GlanceNfsEnabled

  Enables Pacemaker to manage the share for image storage. If disabled, the Overcloud stores images in the Controller node’s file system. Set to **true**.

  GlanceNfsShare

  The NFS share to mount for image storage. For example, 192.168.122.1:/export/glance.

  GlanceNfsOptions

  The NFS mount options for the image storage.

- Due to the new composable service architecture, the syntax for some of the configuration hooks have changes. If using pre or post configuration hooks to provide custom scripts to your environment, check the syntax of your custom environment files. Use the following sections from the *Advanced Overcloud Customization* guide:

  - Customizing Overcloud Pre-Configuration

  - Customizing Overcloud Post-Configuration

- Some composable services include new parameters that configure Puppet hieradata. If you used hieradata to configure these parameters in the past, the overcloud update might report a **Duplicate declaration** error. If this situation, use the composable service parameter. For example, instead of the following:

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

  Use the following:

  ```
  parameter_defaults:
    HeatWorkers: 1
  ```

3.5.6. Custom Core Templates

IMPORTANT

This section is only required if using a modified version of the core Heat template collection. This is because the copy is a static snapshot of the original core Heat template collection from **/usr/share/openstack-tripleo-heat-templates/**. If using an unmodified core Heat template collection for your overcloud, you can skip this section.

To update your modified template collection, you need to:

1. Backup your existing custom template collection:

   ```
   $ mv ~/templates/my-overcloud/ ~/templates/my-overcloud.bak
   ```

2. Replace the new version of the template collection from /usr/share/openstack-tripleo-heat-templates:

   ```
   $ sudo cp -rv /usr/share/openstack-tripleo-heat-templates ~/templates/my-overcloud/
   ```

3. Check for differences between the old and new custom template collection. To see changes between the two, use the following diff command:

   ```
   $ diff -Nar ~/templates/my-overcloud.bak/ ~/templates/my-overcloud/
   ```

This helps identify customizations from the old template collection that you can incorporate into the new template collection. Incorporate customization into the new custom template collection.

IMPORTANT

Red Hat provides updates to the Heat template collection over subsequent releases. Using a modified template collection can lead to a divergence between your custom copy and the original copy in **/usr/share/openstack-tripleo-heat-templates**. To customize your overcloud, Red Hat recommends using the Configuration Hooks from the *Advanced Overcloud Customization* guide. If creating a copy of the Heat template collection, you should track changes to the templates using a version control system such as git.

3.6. UPGRADING THE OVERCLOUD

3.6.1. Overview and Workflow

This section details the steps required to upgrade the Overcloud. Make sure to follow each section in order and only apply the sections relevant to your environment.

This process requires you to run your original **openstack overcloud deploy** command multiple times to provide a staged method of upgrading. Each time you run the command, you include a different upgrade environment file along with your existing environment files. These new upgrade environments files are:

- **major-upgrade-ceilometer-wsgi-mitaka-newton.yaml** - Converts OpenStack Telemetry (‘Ceilometer’) to a WSGI service.
- **major-upgrade-pacemaker-init.yaml** - Provides the initialization for the upgrade. This includes updating the Red Hat OpenStack Platform repositories on each node in your Overcloud and provides special upgrade scripts to certain nodes.
- **major-upgrade-pacemaker.yaml** - Provides an upgrade for the Controller nodes.
- (Optional) **major-upgrade-remove-sahara.yaml** - Removes OpenStack Clustering (**sahara**) from the Overcloud. This accommodates a difference between OpenStack Platform 9 and 10. See Section 3.6.5, “Upgrading Controller Nodes” for more information.
- **major-upgrade-pacemaker-converge.yaml** - The finalization for the Overcloud upgrade. This aligns the resulting upgrade to match the contents for the director’s latest Heat template collection.
- **major-upgrade-aodh-migration.yaml** - Migrates the OpenStack Telemetry Alarming (**aodh**) service’s database from MongoDB to MariaDB

In between these deployment commands, you run the **upgrade-non-controller.sh** script on various node types. This script upgrades the packages on a non-Controller node.

Workflow

The Overcloud upgrade process uses the following workflow:

1. Run your deployment command including the **major-upgrade-ceilometer-wsgi-mitaka-newton.yaml** environment file.
2. Run your deployment command including the **major-upgrade-pacemaker-init.yaml** environment file.
3. Run the **upgrade-non-controller.sh** on each Object Storage node.
4. Run your deployment command including the **major-upgrade-pacemaker.yaml** and the optional **major-upgrade-remove-sahara.yaml** environment file.
5. Run the **upgrade-non-controller.sh** on each Ceph Storage node.
6. Run the **upgrade-non-controller.sh** on each Compute node.
7. Run your deployment command including the **major-upgrade-pacemaker-converge.yaml** environment file.

8. Run your deployment command including the **major-upgrade-aodh-migration.yaml** environment file.

3.6.2. Upgrading OpenStack Telemetry to a WSGI Service

This step upgrades the OpenStack Telemetry (**ceilometer**) service to run as a Web Server Gateway Interface (WSGI) applet under **httpd** instead of a standalone service. This process automatically disables the standalone **openstack-ceilometer-api** service and installs the necessary configuration to enable the WSGI applet.

Run the **openstack overcloud deploy** from your Undercloud and include the **major-upgrade-ceilometer-wsgi-mitaka-newton.yaml** environment file. Make sure you also include all options and custom environment files relevant to your environment, such as network isolation and storage.

This following is an example of an **openstack overcloud deploy** command with the added file:

```
$ 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 /home/stack/templates/network_env.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/major-upgrade-ceilometer-wsgi-
mitaka-newton.yaml \
  --ntp-server pool.ntp.org
```

Wait until the Overcloud updates with the new environment file's configuration.

NOTE

Login to a Controller node and run the **pcs status** command to check if all resources are active in the Controller cluster. If any resource have failed, run **pcs resource cleanup**, which cleans the errors and sets the state of each resource to **Started**. If any errors persist, contact Red Hat and request guidance and assistance.

3.6.3. Installing the Upgrade Scripts

This step installs scripts on each non-Controller node. These script perform the major version package upgrades and configuration. Each script differs depending on the node type. For example, Compute nodes receive different upgrade scripts to Ceph Storage nodes.

This initialization step also updates enabled repositories on all overcloud nodes. This means you do not need to disable old repositories and enable new repositories manually.

Run the **openstack overcloud deploy** from your Undercloud and include the **major-upgrade-pacemaker-init.yaml** environment file. Make sure you also include all options and custom environment files relevant to your environment, such as network isolation and storage.

This following is an example of an **openstack overcloud deploy** command with the added file:

```
$ 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 /home/stack/templates/network_env.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/major-upgrade-pacemaker-init.yaml \
--ntp-server pool.ntp.org
```

Wait until the Overcloud updates with the new environment file's configuration.

NOTE

Login to a Controller node and run the **pcs status** command to check if all resources are active in the Controller cluster. If any resource have failed, run **pcs resource cleanup**, which cleans the errors and sets the state of each resource to **Started**. If any errors persist, contact Red Hat and request guidance and assistance.

3.6.4. Upgrading Object Storage Nodes

The director uses the **upgrade-non-controller.sh** command to run the upgrade script passed to each non-Controller node from the **major-upgrade-pacemaker-init.yaml** environment file. For this step, upgrade each Object Storage node using the following command:

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

Wait until each Object Storage node completes its upgrade.

Check the **/var/log/yum.log** file on each Object Storage node to see if either the **kernel** or **openvswitch** packages have updated their major or minor versions. If so, perform a reboot of each Object Storage node:

1. Select a Object Storage node to reboot. Log into it and reboot it:

   ```
   $ sudo reboot
   ```

2. Wait until the node boots.

3. Log into the node and check the status:

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

4. Log out of the node and repeat this process on the next Object Storage node.

NOTE

Login to a Controller node and run the **pcs status** command to check if all resources are active in the Controller cluster. If any resource have failed, run **pcs resource cleanup**, which cleans the errors and sets the state of each resource to **Started**. If any errors persist, contact Red Hat and request guidance and assistance.

3.6.5. Upgrading Controller Nodes

Upgrading the Controller nodes involves including another environment file (**major-upgrade-pacemaker.yaml**) that provides a full upgrade to Controller nodes running high availability tools.

Run the **openstack overcloud deploy** from your Undercloud and include the **major-upgrade-pacemaker.yaml** environment file. Remember to include all options and custom environment files relevant to your environment, such as network isolation and storage.

Your Controller nodes might require an additional file depending on whether you aim to keep the OpenStack Data Processing (**sahara**) service enabled. OpenStack Platform 9 automatically installed OpenStack Data Processing for a default Overcloud. In OpenStack Platform 10, the user needs to explicitly include environment files to enable OpenStack Data Processing. This means:

- If you no longer require OpenStack Data Processing, include **major-upgrade-remove-sahara.yaml** file in the deployment.

- If you aim to keep OpenStack Data Processing, do not include the **major-upgrade-remove-sahara.yaml** file in the deployment. After completing the Overcloud upgrade, make sure to include the **/usr/share/openstack-tripleo-heat-templates/environments/services/sahara.yaml** to keep the service enabled and configured.

The following is an example of an **openstack overcloud deploy** command with both the required and optional files:

```
$ 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-pacemaker.yaml
  -e /usr/share/openstack-tripleo-heat-templates/environments/major-upgrade-remove-sahara.yaml \
  --ntp-server pool.ntp.org
```

Wait until the Overcloud updates with the new environment file’s configuration.

IMPORTANT

Note the following:

- This step disables the Neutron server and L3 Agent during the Controller upgrade. This means floating IP address are unavailable during this step.

- This step revises the Pacemaker configuration for the Controller cluster. This means the upgrade disables certain high availability functions temporarily.

Check the **/var/log/yum.log** file on each Controller node to see if either the **kernel** or **openvswitch** packages have updated their major or minor versions. If so, perform a reboot of each Controller node:

1. Select a node to reboot. Log into it and stop the cluster before rebooting:

   ```
   $ sudo pcs cluster stop
   ```

2. Reboot the cluster:

   ```
   $ sudo reboot
   ```

   The remaining Controller Nodes in the cluster retain the high availability services during the reboot.

3. Wait until the node boots.

4. Re-enable the cluster for the node:

```
$ sudo pcs cluster start
```

5. Log into the node and check the cluster status:

```
$ sudo pcs status
```

The node rejoins the cluster.

NOTE

If any services fail after the reboot, run sudo **pcs resource cleanup**, which cleans the errors and sets the state of each resource to **Started**. If any errors persist, contact Red Hat and request guidance and assistance.

6. Check all **systemd** services on the Controller Node are active:

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

7. Log out of the node, select the next Controller Node to reboot, and repeat this procedure until you have rebooted all Controller Nodes.

IMPORTANT

The OpenStack Platform 10 upgrade procedure migrates to a new composable architecture. This means many services that Pacemaker managed in previous versions now use **systemd** management. This results in a reduced number of Pacemaker managed resources.

3.6.6. Upgrading Ceph Storage Nodes

The director uses the **upgrade-non-controller.sh** command to run the upgrade script passed to each non-Controller node from the **major-upgrade-pacemaker-init.yaml** environment file. For this step, upgrade each Ceph Storage node with the following command:

Upgrade each Ceph Storage nodes:

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

Check the /**var**/**log**/**yum.log** file on each Ceph Storage node to see if either the **kernel** or **openvswitch** packages have updated their major or minor versions. If so, perform a reboot of each Ceph Storage node:

1. Log into a Ceph MON or Controller node and disable Ceph Storage cluster rebalancing temporarily:

```
$ sudo ceph osd set noout
$ sudo ceph osd set norebalance
```

2. Select the first Ceph Storage node to reboot and log into it.

3. Reboot the node:

   ```
   $ sudo reboot
   ```

4. Wait until the node boots.

5. Log into the node and check the cluster status:

   ```
   $ sudo ceph -s
   ```

   Check that the **pgmap** reports all **pgs** as normal (**active+clean**).

6. Log out of the node, reboot the next node, and check its status. Repeat this process until you have rebooted all Ceph storage nodes.

7. When complete, log into a Ceph MON or Controller node and enable cluster rebalancing again:

   ```
   $ sudo ceph osd unset noout
   $ sudo ceph osd unset norebalance
   ```

8. Perform a final status check to verify the cluster reports **HEALTH_OK**:

   ```
   $ sudo ceph status
   ```

NOTE

Login to a Controller node and run the **pcs status** command to check if all resources are active in the Controller cluster. If any resource have failed, run **pcs resource cleanup**, which cleans the errors and sets the state of each resource to **Started**. If any errors persist, contact Red Hat and request guidance and assistance.

3.6.7. Upgrading Compute Nodes

Upgrade each Compute node individually and ensure zero downtime of instances in your OpenStack Platform environment. This involves the following workflow:

1. Select a Compute node to upgrade

2. Migrate its instances to another Compute node

3. Upgrade the empty Compute node

List all Compute nodes and their UUIDs:

```
$ openstack server list | grep "compute"
```

Select a Compute node to upgrade and first migrate its instances using the following process:

1. From the undercloud, select a Compute Node to reboot and disable it:

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

2. List all instances on the Compute node:

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

3. Migrate each instance from the disabled host. Use one of the following commands:

   a. Migrate the instance to a specific host of your choice:

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

   b. Let **nova-scheduler** automatically select the target host:

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

      NOTE

      The **nova** command might cause some deprecation warnings, which are safe to ignore.

4. Wait until migration completes.

5. Confirm the instance has migrated from the Compute node:

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

6. Repeat this step until you have migrated all instances from the Compute Node.

IMPORTANT

For full instructions on configuring and migrating instances, see "Migrating Virtual Machines Between Compute Nodes" in the *Director Installation and Usage* guide.

The director uses the **upgrade-non-controller.sh** command to run the upgrade script passed to each non-Controller node from the **major-upgrade-pacemaker-init.yaml** environment file. Upgrade each Compute node with the following command:

```
$ source ~/stackrc
$ upgrade-non-controller.sh --upgrade NODE_UUID
```

Replace *NODE_UUID* with the UUID of the chosen Compute node. Wait until the Compute node completes its upgrade.

Check the **/var/log/yum.log** file on the Compute node you have upgraded to see if either the **kernel** or **openvswitch** packages have updated their major or minor versions. If so, perform a reboot of the Compute node:

1. Log into the Compute Node and reboot it:

```
$ sudo reboot
```

2. Wait until the node boots.

3. Enable the Compute Node again:

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

4. Select the next node to reboot.

Repeat the migration and reboot process for each node individually until you have rebooted all nodes.

> NOTE
>
> Login to a Controller node and run the **pcs status** command to check if all resources are active in the Controller cluster. If any resource have failed, run **pcs resource cleanup**, which cleans the errors and sets the state of each resource to **Started**. If any errors persist, contact Red Hat and request guidance and assistance.

3.6.8. Finalizing the Upgrade

The director needs to run through the upgrade finalization to ensure the Overcloud stack is synchronized with the current Heat template collection. This involves an environment file (**major-upgrade-pacemaker-converge.yaml**), which you include using the **openstack overcloud deploy** command.

> IMPORTANT
>
> If your Red Hat OpenStack Platform 9 environment is integrated with an external Ceph Storage Cluster from an earlier version (that is, Red Hat Ceph Storage 1.3), you must enable backwards compatibility. To do so, create an environment file (for example, **/home/stack/templates/ceph-backwards-compatibility.yaml**) containing the following:
>
> ```
> parameter_defaults:
>   ExtraConfig:
>     ceph::conf::args:
>       client/rbd_default_features:
>         value: "1"
> ```
>
> > NOTE
> >
> > Set **rbd_default_features** to **1** if you require backward compatibility with external Red Hat Ceph Storage version 1.3. If the external Red Hat Ceph Storage cluster is version 2.0 or later, or if you used director to deploy your Red Hat OpenStack Platform environment and director upgraded the environment to Red Hat Ceph Storage version 2.0 or later, set **rbd_default_features** to **63**.
>
> Then, include this file in when you run **openstack overcloud deploy** in the next step.

Run the **openstack overcloud deploy** from your Undercloud and include the **major-upgrade-**

pacemaker-converge.yaml environment file. Make sure you also include all options and custom environment files relevant to your environment, such as backwards compatibility for Ceph (if applicable), network isolation, and storage.

This following is an example of an **openstack overcloud deploy** command with the added **major-upgrade-pacemaker-converge.yaml** file:

```
$ 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-pacemaker-
converge.yaml \
  --ntp-server pool.ntp.org
```

Wait until the Overcloud updates with the new environment file’s configuration.

NOTE

Log in to a Controller node and run the **pcs status** command to check if all resources are active in the Controller cluster. If any resource have failed, run **pcs resource cleanup**, which cleans the errors and sets the state of each resource to **Started**. If any errors persist, contact Red Hat and request guidance and assistance.

3.6.9. Migrating the OpenStack Telemetry Alarming Database

This step migrates the OpenStack Telemetry Alarming (**aodh**) service’s database from MongoDB to MariaDB. This process automatically performs the migration.

Run the **openstack overcloud deploy** from your Undercloud and include the **major-upgrade-aodh-migration.yaml** environment file. Make sure you also include all options and custom environment files relevant to your environment, such as network isolation and storage.

This following is an example of an **openstack overcloud deploy** command with the added file:

```
$ 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 /home/stack/templates/network_env.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/major-upgrade-aodh-migration.yaml \
  --ntp-server pool.ntp.org
```

Wait until the Overcloud updates with the new environment file’s configuration.

NOTE

Login to a Controller node and run the **pcs status** command to check if all resources are active in the Controller cluster. If any resource have failed, run **pcs resource cleanup**, which cleans the errors and sets the state of each resource to **Started**. If any errors persist, contact Red Hat and request guidance and assistance.

This completes the Overcloud upgrade procedure.

3.7. POST-UPGRADE NOTES FOR THE OVERCLOUD

Be aware of the following notes after upgrading the Overcloud to Red Hat OpenStack Platform 10:

- Review the resulting configuration files for each service. The upgraded packages might have installed **.rpmnew** files appropriate to the Red Hat OpenStack Platform 10 version of each service.

- If you did not include optional **major-upgrade-remove-sahara.yaml** file from Section 3.6.5, "Upgrading Controller Nodes", make sure to include the **/usr/share/openstack-tripleo-heat-templates/environments/services/sahara.yaml** to ensure OpenStack Clustering (**sahara**) stays enabled in the overcloud.

- The Compute nodes might report a failure with **neutron-openvswitch-agent**. If this occurs, log into each Compute node and restart the service. For example:

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

- The upgrade process does not reboot any nodes in the Overcloud automatically. If required, perform a reboot manually after the upgrade command completes. Make sure to reboot cluster-based nodes (such as Ceph Storage nodes and Controller nodes) individually and wait for the node to rejoin the cluster. For Ceph Storage nodes, check with the **ceph health** and make sure the cluster status is **HEALTH OK**. For Controller nodes, check with the **pcs resource** and make sure all resources are running for each node.

- In some circumstances, the **corosync** service might fail to start on IPv6 environments after rebooting Controller nodes. This is due to Corosync starting before the Controller node configures the static IPv6 addresses. In these situations, restart Corosync manually on the Controller nodes:

  ```
  $ sudo systemctl restart corosync
  ```

- If you configured fencing for your Controller nodes, the upgrade process might disable it. When the upgrade process completes, reenable fencing with the following command on one of the Controller nodes:

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

- The next time you update or scale the Overcloud stack (i.e. running the **openstack overcloud deploy** command), you need to reset the identifier that triggers package updates in the Overcloud. Add a blank **UpdateIdentifier** parameter to an environment file and include it when you run the **openstack overcloud deploy** command. The following is an example of such an environment file:

  ```
  parameter_defaults:
    UpdateIdentifier:
  ```

CHAPTER 4. NON-DIRECTOR ENVIRONMENTS: UPGRADING OPENSTACK SERVICES SIMULTANEOUSLY

This scenario upgrades from Red Hat OpenStack Platform 9 to Red Hat OpenStack Platform 10 in environments that **do not use the director**. This procedure upgrades all services on all nodes. This involves the following workflow:

1. Disabling all OpenStack services
2. Performing a package upgrade
3. Performing synchronization of all databases
4. Enabling all OpenStack services

NOTE

The procedures in this chapter follow the architectural naming convention followed by all Red Hat OpenStack Platform documentation. If you are unfamiliar with this convention, refer to Architecture Guide available at Red Hat OpenStack Platform Documentation Suite before proceeding.

4.1. DISABLING ALL OPENSTACK SERVICES

The first step to performing a complete upgrade of Red Hat OpenStack Platform on a node involves shutting down all OpenStack services. This step differs based on whether the node OpenStack uses high availability tools for management (e.g. using Pacemaker on Controller nodes). This step contains instructions for both node types.

Standard Nodes

Before updating, take a **systemd** snapshot of the OpenStack Platform services.

```
# systemctl snapshot openstack-services
```

Disable the main OpenStack Platform services:

```
# systemctl stop 'openstack-*'
# systemctl stop 'neutron-*'
# systemctl stop 'openvswitch'
```

High Availability Nodes

You need to disable all OpenStack services but leave the database and load balancing services active. For example, switch the HAProxy, Galera, and MongoDB services to unmanaged in Pacemaker:

```
# pcs resource unmanage haproxy
# pcs resource unmanage galera
# pcs resource unmanage mongod
```

Disable the remaining Pacemaker-managed resources by setting the **stop-all-resources** property on the cluster. Run the following on a single member of your Pacemaker cluster:

```
# pcs property set stop-all-resources=true
```

- Wait until all Pacemaker-managed resources have stopped. Run the **pcs status** command to see the status of each resource.

```
# pcs status
```

IMPORTANT

HAProxy might show a broadcast message for unavailable services. This is normal behavior.

4.2. PERFORMING A PACKAGE UPGRADE

The next step upgrades all packages on a node. Perform this step on each node with OpenStack services.

Change to the Red Hat OpenStack Platform 10 repository using the **subscription-manager** command:

```
# subscription-manager repos --disable=rhel-7-server-openstack-9-rpms
# subscription-manager repos --enable=rhel-7-server-openstack-10-rpms
```

Run the **yum update** command on the node:

```
# yum update
```

Wait until the package upgrade completes.

Review the resulting configuration files. The upgraded packages will have installed **.rpmnew** files appropriate to the Red Hat OpenStack Platform 10 version of the service. New versions of OpenStack services may deprecate certain configuration options. You should also review your OpenStack logs for any deprecation warnings, because these may cause problems during future upgrades. For more information on the new, updated and deprecated configuration options for each service, see Configuration Reference available from Red Hat OpenStack Platform Documentation Suite .

Perform the package upgrade on each node in your environment.

4.3. PERFORMING SYNCHRONIZATION OF ALL DATABASES

The next step upgrades the database for each service.

NOTE

Flush expired tokens in the Identity service to decrease the time required to synchronize the database.

```
# keystone-manage token_flush
```

Upgrade the database schema for each service that uses the database. Run the following commands on the node hosting the service’s database.

Table 4.1. Commands to Synchronize OpenStack Service Databases

Service	Project name	Command
Identity	keystone	**# su -s /bin/sh -c "keystone-manage db_sync" keystone**
Image Service	glance	**# su -s /bin/sh -c "glance-manage db_sync" glance**
Block Storage	cinder	**# su -s /bin/sh -c "cinder-manage db sync" cinder**
Orchestration	heat	**# su -s /bin/sh -c "heat-manage db_sync" heat**
Compute	nova	**# su -s /bin/sh -c "nova-manage api_db sync" nova** **# su -s /bin/sh -c "nova-manage db sync" nova**
Telemetry	ceilometer	**# ceilometer-dbsync**
Telemetry Alarming	aodh	**# aodh-dbsync**
Telemetry Metrics	gnocchi	**# gnocchi-upgrade**
Clustering	sahara	**# su -s /bin/sh -c "sahara-db-manage upgrade heads" sahara**
Networking	neutron	**# su -s /bin/sh -c "neutron-db-manage upgrade heads" neutron**

4.4. ENABLING ALL OPENSTACK SERVICES

The final step enables the OpenStack services on the node. This step differs based on whether the node OpenStack uses high availability tools for management. For example, using Pacemaker on Controller nodes. This step contains instructions for both node types.

Standard Nodes

Restart all OpenStack services:

```
# systemctl isolate openstack-services.snapshot
```

High Availability Nodes

Restart your resources through Pacemaker. Reset the **stop-all-resources** property on a single member of your Pacemaker cluster. For example:

```
# pcs property set stop-all-resources=false
```

Wait until all resources have started. Run the **pcs status** command to see the status of each resource:

```
# pcs status
```

Enable Pacemaker management for any unmanaged resources, such as the databases and load balancer:

```
# pcs resource manage haproxy
# pcs resource manage galera
# pcs resource manage mongod
```

NOTE

This upgrade scenario has some additional upgrade procedures detailed in Chapter 7, *Additional Procedures for Non-Director Environments*. These additional procedures are optional for manual environments but help align with the current OpenStack Platform recommendations.

4.5. POST-UPGRADE NOTES

New versions of OpenStack services may deprecate certain configuration options. You should also review your OpenStack logs for any deprecation warnings, because these may cause problems during a future upgrade. For more information on the new, updated and deprecated configuration options for each service, see Configuration Reference available from Red Hat OpenStack Platform Documentation Suite.

CHAPTER 5. NON-DIRECTOR ENVIRONMENTS: UPGRADING INDIVIDUAL OPENSTACK SERVICES (LIVE COMPUTE) IN A STANDARD ENVIRONMENT

This section describes the steps you should follow to upgrade your cloud deployment by updating one service at a time with live compute in a non High Availability (HA) environment. This scenario upgrades from Red Hat OpenStack Platform 9 to Red Hat OpenStack Platform 10 in environments that **do not use the director**.

A live Compute upgrade minimizes interruptions to your Compute service, with only a few minutes for the smaller services, and a longer migration interval for the workloads moving to newly-upgraded Compute hosts. Existing workloads can run indefinitely, and you do not need to wait for a database migration.

IMPORTANT

Due to certain package dependencies, upgrading the packages for one OpenStack service might cause Python libraries to upgrade before other OpenStack services upgrade. This might cause certain services to fail prematurely. In this situation, continue upgrading the remaining services. All services should be operational upon completion of this scenario.

NOTE

This method may require additional hardware resources to bring up the Compute nodes.

NOTE

The procedures in this chapter follow the architectural naming convention followed by all Red Hat OpenStack Platform documentation. If you are unfamiliar with this convention, refer to Architecture Guide available at Red Hat OpenStack Platform Documentation Suite before proceeding.

5.1. PRE-UPGRADE TASKS

On each node, change to the Red Hat OpenStack Platform 10 repository using the **subscription-manager** command:

```
# subscription-manager repos --disable=rhel-7-server-openstack-9-rpms
# subscription-manager repos --enable=rhel-7-server-openstack-10-rpms
```

Before updating, take a **systemd** snapshot of the OpenStack Platform services.

```
# systemctl snapshot openstack-services
```

Disable the main OpenStack Platform services:

```
sudo systemctl stop 'openstack-*'
sudo systemctl stop 'neutron-*'
sudo systemctl stop 'openvswitch'
```

Upgrade the **openstack-selinux** package:

```
# yum upgrade openstack-selinux
```

This is necessary to ensure that the upgraded services will run correctly on a system with SELinux enabled.

5.2. UPGRADING WSGI SERVICES

Disable the Identity service and the Dashboard WSGI applets:

```
# systemctl stop httpd
```

Update the packages for both services:

```
# yum -d1 -y upgrade \*keystone\*
# yum -y upgrade \*horizon\* \*openstack-dashboard\*
# yum -d1 -y upgrade \*horizon\* \*python-django\*
```

It is possible that the Identity service's token table has a large number of expired entries. This can dramatically increase the time it takes to complete the database schema upgrade. To flush expired tokens from the database and alleviate the problem, the **keystone-manage** command can be used before running the Identity database upgrade.

```
# keystone-manage token_flush
# su -s /bin/sh -c "keystone-manage db_sync" keystone
```

This flushes expired tokens from the database. You can arrange to run this command periodically using **cron**.

Restart the **httpd** service.

```
# systemctl start httpd
```

5.3. UPGRADING OBJECT STORAGE (SWIFT)

On your Object Storage hosts, run:

```
# systemctl stop '*swift*'
# yum -d1 -y upgrade \*swift\*
# systemctl start openstack-swift-account-auditor \
    openstack-swift-account-reaper \
    openstack-swift-account-replicator \
    openstack-swift-account \
    openstack-swift-container-auditor \
    openstack-swift-container-replicator \
    openstack-swift-container-updater \
    openstack-swift-container \
    openstack-swift-object-auditor \
    openstack-swift-object-replicator \
    openstack-swift-object-updater \
    openstack-swift-object \
    openstack-swift-proxy
```

5.4. UPGRADING IMAGE SERVICE (GLANCE)

On your Image Service host, run:

```
# systemctl stop '*glance*'
# yum -d1 -y upgrade \*glance\*
# su -s /bin/sh -c "glance-manage db_sync" glance
# systemctl start openstack-glance-api \
    openstack-glance-registry
```

5.5. UPGRADING BLOCK STORAGE (CINDER)

On your Block Storage host, run:

```
# systemctl stop '*cinder*'
# yum -d1 -y upgrade \*cinder\*
# su -s /bin/sh -c "cinder-manage db sync" cinder
# systemctl start openstack-cinder-api \
    openstack-cinder-scheduler \
    openstack-cinder-volume
```

5.6. UPGRADING ORCHESTRATION (HEAT)

On your Orchestration host, run:

```
# systemctl stop '*heat*'
# yum -d1 -y upgrade \*heat\*
# su -s /bin/sh -c "heat-manage db_sync" heat
# systemctl start openstack-heat-api-cfn \
    openstack-heat-api-cloudwatch \
    openstack-heat-api \
    openstack-heat-engine
```

5.7. UPGRADING TELEMETRY (CEILOMETER)

NOTE

This component has some additional upgrade procedures detailed in Chapter 7, *Additional Procedures for Non-Director Environments*. These additional procedures are optional for manual environments but help align with the current OpenStack Platform recommendations.

1. On all nodes hosting Telemetry component services, run:

```
# systemctl stop '*ceilometer*'
# systemctl stop '*aodh*'
# systemctl stop '*gnocchi*'
# yum -d1 -y upgrade \*ceilometer\* \*aodh\* \*gnocchi\*
```

2. On the controller node, where database is installed, run:

```
# ceilometer-dbsync
# aodh-dbsync
# gnocchi-upgrade
```

3. After completing the package upgrade, restart the Telemetry service by running the following command on all nodes hosting Telemetry component services:

```
# systemctl start openstack-ceilometer-api \
    openstack-ceilometer-central \
    openstack-ceilometer-collector \
    openstack-ceilometer-notification \
    openstack-aodh-evaluator \
    openstack-aodh-listener \
    openstack-aodh-notifier \
    openstack-gnocchi-metricd \
    openstack-gnocchi-statsd
```

5.8. UPGRADING COMPUTE (NOVA)

1. If you are performing a rolling upgrade of your compute hosts you need to set explicit API version limits to ensure compatibility in your environment.
   Before starting Compute services on Controller or Compute nodes, set the **compute** option in the **[upgrade_levels]** section of **nova.conf** to the previous Red Hat OpenStack Platform version (**mitaka**):

   ```
   # crudini --set /etc/nova/nova.conf upgrade_levels compute mitaka
   ```

   You need to make this change on your Controller and Compute nodes.

   You should undo this operation after upgrading all of your Compute nodes.

2. On your Compute host, run:

   ```
   # systemctl stop '*nova*'
   # yum -d1 -y upgrade \*nova\*
   # su -s /bin/sh -c "nova-manage api_db sync" nova
   # su -s /bin/sh -c "nova-manage db sync" nova
   ```

3. After you have upgraded all of your hosts, you will want to remove the API limits configured in the previous step. On all of your hosts:

   ```
   # crudini --del /etc/nova/nova.conf upgrade_levels compute
   ```

4. Restart the Compute service on all the Controller and Compute nodes:

   ```
   # systemctl start openstack-nova-api \
       openstack-nova-conductor \
       openstack-nova-consoleauth \
       openstack-nova-novncproxy \
       openstack-nova-scheduler
   ```

5.9. UPGRADING CLUSTERING (SAHARA)

1. On all nodes hosting Clustering component services, run:

   ```
   # systemctl stop '*sahara*'
   # yum -d1 -y upgrade \*sahara\*
   ```

2. On the controller node, where database is installed, run:

   ```
   # su -s /bin/sh -c "sahara-db-manage upgrade heads" sahara
   ```

3. After completing the package upgrade, restart the Clustering service by running the following command on all nodes hosting Clustering component services:

   ```
   # systemctl start openstack-sahara-api \
       openstack-sahara-engine
   ```

5.10. UPGRADING OPENSTACK NETWORKING (NEUTRON)

1. On your OpenStack Networking host, run:

   ```
   # systemctl stop '*neutron*'
   # yum -d1 -y upgrade \*neutron\*
   ```

2. On the same host, update the OpenStack Networking database schema:

   ```
   # su -s /bin/sh -c "neutron-db-manage upgrade heads" neutron
   ```

3. Restart the OpenStack Networking service:

   ```
   # systemctl start neutron-dhcp-agent \
       neutron-l3-agent \
       neutron-metadata-agent \
       neutron-openvswitch-agent \
       neutron-server
   ```

NOTE

Start any additional OpenStack Networking services enabled in your environment.

5.11. POST-UPGRADE TASKS

After completing all of your individual service upgrades, you should perform a complete package upgrade on all of your systems:

```
# yum upgrade
```

This will ensure that all packages are up-to-date. You may want to schedule a restart of your OpenStack hosts at a future date in order to ensure that all running processes are using updated versions of the underlying binaries.

Review the resulting configuration files. The upgraded packages will have installed **.rpmnew** files appropriate to the Red Hat OpenStack Platform 10 version of the service.

New versions of OpenStack services may deprecate certain configuration options. You should also review your OpenStack logs for any deprecation warnings, because these may cause problems during a future upgrade. For more information on the new, updated and deprecated configuration options for each service, see Configuration Reference available from Red Hat OpenStack Platform Documentation Suite.

CHAPTER 6. NON-DIRECTOR ENVIRONMENTS: UPGRADING INDIVIDUAL OPENSTACK SERVICES (LIVE COMPUTE) IN A HIGH AVAILABILITY ENVIRONMENT

This chapter describes the steps you should follow to upgrade your cloud deployment by updating one service at a time with live compute in a High Availability (HA) environment. This scenario upgrades from Red Hat OpenStack Platform 9 to Red Hat OpenStack Platform 10 in environments that **do not use the director**.

A live Compute upgrade minimizes interruptions to your Compute service, with only a few minutes for the smaller services, and a longer migration interval for the workloads moving to newly-upgraded Compute hosts. Existing workloads can run indefinitely, and you do not need to wait for a database migration.

IMPORTANT

Due to certain package dependencies, upgrading the packages for one OpenStack service might cause Python libraries to upgrade before other OpenStack services upgrade. This might cause certain services to fail prematurely. In this situation, continue upgrading the remaining services. All services should be operational upon completion of this scenario.

NOTE

This method may require additional hardware resources to bring up the Compute nodes.

NOTE

The procedures in this chapter follow the architectural naming convention followed by all Red Hat OpenStack Platform documentation. If you are unfamiliar with this convention, refer to Architecture Guide available at Red Hat OpenStack Platform Documentation Suite before proceeding.

6.1. PRE-UPGRADE TASKS

On each node, change to the Red Hat OpenStack Platform 10 repository using the **subscription-manager** command:

```
# subscription-manager repos --disable=rhel-7-server-openstack-9-rpms
# subscription-manager repos --enable=rhel-7-server-openstack-10-rpms
```

Upgrade the **openstack-selinux** package:

```
# yum upgrade openstack-selinux
```

This is necessary to ensure that the upgraded services will run correctly on a system with SELinux enabled.

6.2. UPGRADING MARIADB

Perform the follow steps on each host running MariaDB. Complete the steps on one host before starting the process on another host.

1. Stop the service from running on the local node:

   ```
   # pcs resource ban galera-master $(crm_node -n)
   ```

2. Wait until **pcs status** shows that the service is no longer running on the local node. This may take a few minutes. The local node transitions to slave mode:

   ```
   Master/Slave Set: galera-master [galera]
   Masters: [ overcloud-controller-1 overcloud-controller-2 ]
   Slaves: [ overcloud-controller-0 ]
   ```

   The node eventually transitions to stopped:

   ```
   Master/Slave Set: galera-master [galera]
   Masters: [ overcloud-controller-1 overcloud-controller-2 ]
   Stopped: [ overcloud-controller-0 ]
   ```

3. Upgrade the relevant packages:

   ```
   # yum upgrade '*mariadb*' '*galera*'
   ```

4. Allow Pacemaker to schedule the **galera** resource on the local node:

   ```
   # pcs resource clear galera-master
   ```

5. Wait until **pcs status** shows that the galera resource is running on the local node as a master. The **pcs status** command should provide output similar to the following:

   ```
   Master/Slave Set: galera-master [galera]
   Masters: [ overcloud-controller-0 overcloud-controller-1 overcloud-controller-2 ]
   ```

Perform this procedure on each node individually until the MariaDB cluster completes a full upgrade.

6.3. UPGRADING MONGODB

This procedure upgrades MongoDB, which acts as the backend database for the OpenStack Telemetry service.

1. Remove the **mongod** resource from Pacemaker's control:

   ```
   # pcs resource unmanage mongod-clone
   ```

2. Stop the service on all Controller nodes. On each Controller node, run the following:

   ```
   # systemctl stop mongod
   ```

3. Upgrade the relevant packages:

   ```
   # yum upgrade 'mongodb*' 'python-pymongo*'
   ```

4. Reload **systemd** to account for updated unit files:

```
# systemctl daemon-reload
```

5. Restart the **mongod** service on your controllers by running, on each controller:

```
# systemctl start mongod
```

6. Clean up the resource:

```
# pcs resource cleanup mongod-clone
```

7. Return the resource to Pacemaker control:

```
# pcs resource manage mongod-clone
```

8. Wait until the output of **pcs status** shows that the above resources are running.

6.4. UPGRADING WSGI SERVICES

This procedure upgrades the packages for the WSGI services on all Controller nodes simultaneously. This includes OpenStack Identity (keystone) and OpenStack Dashboard (horizon).

1. Remove the service from Pacemaker's control:

```
# pcs resource unmanage httpd-clone
```

2. Stop the **httpd** service by running the following on each Controller node:

```
# systemctl stop httpd
```

3. Upgrade the relevant packages:

```
# yum -d1 -y upgrade \*keystone\*
# yum -y upgrade \*horizon\* \*openstack-dashboard\* httpd
# yum -d1 -y upgrade \*horizon\* \*python-django\*
```

4. Reload **systemd** to account for updated unit files on each Controller node:

```
# systemctl daemon-reload
```

5. Earlier versions of the installer may not have configured your system to automatically purge expired Keystone token, it is possible that your token table has a large number of expired entries. This can dramatically increase the time it takes to complete the database schema upgrade.
   Flush expired tokens from the database to alleviate the problem. Run the **keystone-manage** command before running the Identity database upgrade.

```
# keystone-manage token_flush
```

   This flushes expired tokens from the database. You can arrange to run this command periodically (e.g., daily) using **cron**.

6. Update the Identity service database schema:

```
# su -s /bin/sh -c "keystone-manage db_sync" keystone
```

7. Restart the service by running the following on each Controller node:

   ```
   # systemctl start httpd
   ```

8. Clean up the Identity service using Pacemaker:

   ```
   # pcs resource cleanup httpd-clone
   ```

9. Return the resource to Pacemaker control:

   ```
   # pcs resource manage httpd-clone
   ```

10. Wait until the output of **pcs status** shows that the above resources are running.

6.5. UPGRADING IMAGE SERVICE (GLANCE)

This procedure upgrades the packages for the Image service on all Controller nodes simultaneously.

1. Stop the Image service resources in Pacemaker:

   ```
   # pcs resource disable openstack-glance-registry-clone
   # pcs resource disable openstack-glance-api-clone
   ```

2. Wait until the output of **pcs status** shows that both services have stopped running.

3. Upgrade the relevant packages:

   ```
   # yum upgrade '*glance*'
   ```

4. Reload **systemd** to account for updated unit files:

   ```
   # systemctl daemon-reload
   ```

5. Update the Image service database schema:

   ```
   # su -s /bin/sh -c "glance-manage db_sync" glance
   ```

6. Clean up the Image service using Pacemaker:

   ```
   # pcs resource cleanup openstack-glance-api-clone
   # pcs resource cleanup openstack-glance-registry-clone
   ```

7. Restart Image service resources in Pacemaker:

   ```
   # pcs resource enable openstack-glance-api-clone
   # pcs resource enable openstack-glance-registry-clone
   ```

8. Wait until the output of **pcs status** shows that the above resources are running.

6.6. UPGRADING BLOCK STORAGE SERVICE (CINDER)

This procedure upgrades the packages for the Block Storage service on all Controller nodes simultaneously.

1. Stop all Block Storage service resources in Pacemaker:

   ```
   # pcs resource disable openstack-cinder-api-clone
   # pcs resource disable openstack-cinder-scheduler-clone
   # pcs resource disable openstack-cinder-volume
   ```

2. Wait until the output of **pcs status** shows that the above services have stopped running.

3. Upgrade the relevant packages:

   ```
   # yum upgrade '*cinder*'
   ```

4. Reload **systemd** to account for updated unit files:

   ```
   # systemctl daemon-reload
   ```

5. Update the Block Storage service database schema:

   ```
   # su -s /bin/sh -c "cinder-manage db sync" cinder
   ```

6. Clean up the Block Storage service using Pacemaker:

   ```
   # pcs resource cleanup openstack-cinder-volume
   # pcs resource cleanup openstack-cinder-scheduler-clone
   # pcs resource cleanup openstack-cinder-api-clone
   ```

7. Restart all Block Storage service resources in Pacemaker:

   ```
   # pcs resource enable openstack-cinder-volume
   # pcs resource enable openstack-cinder-scheduler-clone
   # pcs resource enable openstack-cinder-api-clone
   ```

8. Wait until the output of **pcs status** shows that the above resources are running.

6.7. UPGRADING ORCHESTRATION (HEAT)

This procedure upgrades the packages for the Orchestration service on all Controller nodes simultaneously.

1. Stop Orchestration resources in Pacemaker:

   ```
   # pcs resource disable openstack-heat-api-clone
   # pcs resource disable openstack-heat-api-cfn-clone
   # pcs resource disable openstack-heat-api-cloudwatch-clone
   # pcs resource disable openstack-heat-engine-clone
   ```

2. Wait until the output of **pcs status** shows that the above services have stopped running.

3. Upgrade the relevant packages:

```
# yum upgrade '*heat*'
```

4. Reload **systemd** to account for updated unit files:

```
# systemctl daemon-reload
```

5. Update the Orchestration database schema:

```
# su -s /bin/sh -c "heat-manage db_sync" heat
```

6. Clean up the Orchestration service using Pacemaker:

```
# pcs resource cleanup openstack-heat-clone
# pcs resource cleanup openstack-heat-api-cloudwatch-clone
# pcs resource cleanup openstack-heat-api-cfn-clone
# pcs resource cleanup openstack-heat-api-clone
```

7. Restart Orchestration resources in Pacemaker:

```
# pcs resource enable openstack-heat-clone
# pcs resource enable openstack-heat-api-cloudwatch-clone
# pcs resource enable openstack-heat-api-cfn-clone
# pcs resource enable openstack-heat-api-clone
```

8. Wait until the output of **pcs status** shows that the above resources are running.

6.8. UPGRADING TELEMETRY (CEILOMETER)

This procedure upgrades the packages for the Telemetry service on all Controller nodes simultaneously.

NOTE

This component has some additional upgrade procedures detailed in Chapter 7, *Additional Procedures for Non-Director Environments*. These additional procedures are optional for manual environments but help align with the current OpenStack Platform recommendations.

1. Stop all Telemetry resources in Pacemaker:

```
# pcs resource disable openstack-ceilometer-api-clone
# pcs resource disable openstack-ceilometer-collector-clone
# pcs resource disable openstack-ceilometer-notification-clone
# pcs resource disable openstack-ceilometer-central-clone
# pcs resource disable openstack-aodh-evaluator-clone
# pcs resource disable openstack-aodh-listener-clone
# pcs resource disable openstack-aodh-notifier-clone
# pcs resource disable openstack-gnocchi-metricd-clone
# pcs resource disable openstack-gnocchi-statsd-clone
# pcs resource disable delay-clone
```

2. Wait until the output of **pcs status** shows that the above services have stopped running.

3. Upgrade the relevant packages:

   ```
   # yum upgrade '*ceilometer*' '*aodh*' '*gnocchi*'
   ```

4. Reload **systemd** to account for updated unit files:

   ```
   # systemctl daemon-reload
   ```

5. Use the following command to update Telemetry database schema.

   ```
   # ceilometer-dbsync
   # aodh-dbsync
   # gnocchi-upgrade
   ```

6. Clean up the Telemetry service using Pacemaker:

   ```
   # pcs resource cleanup delay-clone
   # pcs resource cleanup openstack-ceilometer-api-clone
   # pcs resource cleanup openstack-ceilometer-collector-clone
   # pcs resource cleanup openstack-ceilometer-notification-clone
   # pcs resource cleanup openstack-ceilometer-central-clone
   # pcs resource cleanup openstack-aodh-evaluator-clone
   # pcs resource cleanup openstack-aodh-listener-clone
   # pcs resource cleanup openstack-aodh-notifier-clone
   # pcs resource cleanup openstack-gnocchi-metricd-clone
   # pcs resource cleanup openstack-gnocchi-statsd-clone
   ```

7. Restart all Telemetry resources in Pacemaker:

   ```
   # pcs resource enable delay-clone
   # pcs resource enable openstack-ceilometer-api-clone
   # pcs resource enable openstack-ceilometer-collector-clone
   # pcs resource enable openstack-ceilometer-notification-clone
   # pcs resource enable openstack-ceilometer-central-clone
   # pcs resource enable openstack-aodh-evaluator-clone
   # pcs resource enable openstack-aodh-listener-clone
   # pcs resource enable openstack-aodh-notifier-clone
   # pcs resource enable openstack-gnocchi-metricd-clone
   # pcs resource enable openstack-gnocchi-statsd-clone
   ```

8. Wait until the output of **pcs status** shows that the above resources are running.

IMPORTANT

Previous versions of the Telemetry service used an value for the **rpc_backend** parameter that is now deprecated. Check the **rpc_backend** parameter in the **/etc/ceilometer/ceilometer.conf** file is set to the following:

```
rpc_backend=rabbit
```

6.9. UPGRADING THE COMPUTE SERVICE (NOVA) ON CONTROLLER NODES

This procedure upgrades the packages for the Compute service on all Controller nodes simultaneously.

1. Stop all Compute resources in Pacemaker:

```
# pcs resource disable openstack-nova-novncproxy-clone
# pcs resource disable openstack-nova-consoleauth-clone
# pcs resource disable openstack-nova-conductor-clone
# pcs resource disable openstack-nova-api-clone
# pcs resource disable openstack-nova-scheduler-clone
```

2. Wait until the output of **pcs status** shows that the above services have stopped running.

3. Upgrade the relevant packages:

```
# yum upgrade '*nova*'
```

4. Reload **systemd** to account for updated unit files:

```
# systemctl daemon-reload
```

5. Update the Compute database schema:

```
# su -s /bin/sh -c "nova-manage api_db sync" nova
# su -s /bin/sh -c "nova-manage db sync" nova
```

6. If you are performing a rolling upgrade of your compute hosts you need to set explicit API version limits to ensure compatibility between your Mitaka and Newton environments.
   Before starting Compute services on Controller or Compute nodes, set the **compute** option in the **[upgrade_levels]** section of **nova.conf** to the previous Red Hat OpenStack Platform version (**mitaka**):

```
# crudini --set /etc/nova/nova.conf upgrade_levels compute mitaka
```

   This ensures the Controller node can still communicate to the Compute nodes, which are still using the previous version.

   You will need to first unmanage the Compute resources by running **pcs resource unmanage** on one Controller node:

```
# pcs resource unmanage openstack-nova-novncproxy-clone
# pcs resource unmanage openstack-nova-consoleauth-clone
# pcs resource unmanage openstack-nova-conductor-clone
# pcs resource unmanage openstack-nova-api-clone
# pcs resource unmanage openstack-nova-scheduler-clone
```

   Restart all the services on all controllers:

```
# openstack-service restart nova
```

You should return control to the Pacemaker after upgrading all of your compute hosts to Red Hat OpenStack Platform 10.

```
# pcs resource manage openstack-nova-scheduler-clone
# pcs resource manage openstack-nova-api-clone
# pcs resource manage openstack-nova-conductor-clone
# pcs resource manage openstack-nova-consoleauth-clone
# pcs resource manage openstack-nova-novncproxy-clone
```

7. Clean up all Compute resources in Pacemaker:

```
# pcs resource cleanup openstack-nova-scheduler-clone
# pcs resource cleanup openstack-nova-api-clone
# pcs resource cleanup openstack-nova-conductor-clone
# pcs resource cleanup openstack-nova-consoleauth-clone
# pcs resource cleanup openstack-nova-novncproxy-clone
```

8. Restart all Compute resources in Pacemaker:

```
# pcs resource enable openstack-nova-scheduler-clone
# pcs resource enable openstack-nova-api-clone
# pcs resource enable openstack-nova-conductor-clone
# pcs resource enable openstack-nova-consoleauth-clone
# pcs resource enable openstack-nova-novncproxy-clone
```

9. Wait until the output of **pcs status** shows that the above resources are running.

6.10. UPGRADING CLUSTERING SERVICE (SAHARA)

This procedure upgrades the packages for the Clustering service on all Controller nodes simultaneously.

1. Stop all Clustering service resources in Pacemaker:

```
# pcs resource disable openstack-sahara-api-clone
# pcs resource disable openstack-sahara-engine-clone
```

2. Wait until the output of **pcs status** shows that the above services have stopped running.

3. Upgrade the relevant packages:

```
# yum upgrade '*sahara*'
```

4. Reload **systemd** to account for updated unit files:

```
# systemctl daemon-reload
```

5. Update the Clustering service database schema:

```
# su -s /bin/sh -c "sahara-db-manage upgrade heads" sahara
```

6. Clean up the Clustering service using Pacemaker:

```
# pcs resource cleanup openstack-sahara-api-clone
# pcs resource cleanup openstack-sahara-engine-clone
```

7. Restart all Block Storage service resources in Pacemaker:

```
# pcs resource enable openstack-sahara-api-clone
# pcs resource enable openstack-sahara-engine-clone
```

8. Wait until the output of **pcs status** shows that the above resources are running.

6.11. UPGRADING OPENSTACK NETWORKING (NEUTRON)

This procedure upgrades the packages for the Networking service on all Controller nodes simultaneously.

1. Prevent Pacemaker from triggering the OpenStack Networking cleanup scripts:

```
# pcs resource unmanage neutron-ovs-cleanup-clone
# pcs resource unmanage neutron-netns-cleanup-clone
```

2. Stop OpenStack Networking resources in Pacemaker:

```
# pcs resource disable neutron-server-clone
# pcs resource disable neutron-openvswitch-agent-clone
# pcs resource disable neutron-dhcp-agent-clone
# pcs resource disable neutron-l3-agent-clone
# pcs resource disable neutron-metadata-agent-clone
```

3. Upgrade the relevant packages:

```
# yum upgrade 'openstack-neutron*' 'python-neutron*'
```

4. Update the OpenStack Networking database schema:

```
# su -s /bin/sh -c "neutron-db-manage upgrade heads" neutron
```

5. Clean up OpenStack Networking resources in Pacemaker:

```
# pcs resource cleanup neutron-metadata-agent-clone
# pcs resource cleanup neutron-l3-agent-clone
# pcs resource cleanup neutron-dhcp-agent-clone
# pcs resource cleanup neutron-openvswitch-agent-clone
# pcs resource cleanup neutron-server-clone
```

6. Restart OpenStack Networking resources in Pacemaker:

```
# pcs resource enable neutron-metadata-agent-clone
# pcs resource enable neutron-l3-agent-clone
# pcs resource enable neutron-dhcp-agent-clone
# pcs resource enable neutron-openvswitch-agent-clone
# pcs resource enable neutron-server-clone
```

7. Return the cleanup agents to Pacemaker control:

   ```
   # pcs resource manage neutron-ovs-cleanup-clone
   # pcs resource manage neutron-netns-cleanup-clone
   ```

8. Wait until the output of **pcs status** shows that the above resources are running.

6.12. UPGRADING COMPUTE (NOVA) NODES

This procedure upgrades the packages for on a single Compute node. Run this procedure on each Compute node individually.

If you are performing a rolling upgrade of your compute hosts you need to set explicit API version limits to ensure compatibility between your Mitaka and Newton environments.

Before starting Compute services on Controller or Compute nodes, set the **compute** option in the **[upgrade_levels]** section of **nova.conf** to the previous Red Hat OpenStack Platform version (**mitaka**):

```
# crudini --set /etc/nova/nova.conf upgrade_levels compute mitaka
```

Before updating, take a **systemd** snapshot of the OpenStack Platform services.

```
# systemctl snapshot openstack-services
```

This ensures the Controller node can still communicate to the Compute nodes, which are still using the previous version.

1. Stop all OpenStack services on the host:

   ```
   # systemctl stop 'openstack*' '*nova*'
   ```

2. Upgrade all packages:

   ```
   # yum upgrade
   ```

3. Start all OpenStack services on the host:

   ```
   # openstack-service start
   ```

4. After you have upgraded all of your hosts, remove the API limits configured in the previous step. On all of your hosts:

   ```
   # crudini --del /etc/nova/nova.conf upgrade_levels compute
   ```

5. Restart all OpenStack services on the host:

   ```
   # systemctl isolate openstack-services.snapshot
   ```

6.13. POST-UPGRADE TASKS

After completing all of your individual service upgrades, you should perform a complete package upgrade on all nodes:

```
# yum upgrade
```

This will ensure that all packages are up-to-date. You may want to schedule a restart of your OpenStack hosts at a future date in order to ensure that all running processes are using updated versions of the underlying binaries.

Review the resulting configuration files. The upgraded packages will have installed **.rpmnew** files appropriate to the Red Hat OpenStack Platform 10 version of the service.

New versions of OpenStack services may deprecate certain configuration options. You should also review your OpenStack logs for any deprecation warnings, because these may cause problems during a future upgrade. For more information on the new, updated and deprecated configuration options for each service, see Configuration Reference available from Red Hat OpenStack Platform Documentation Suite.

CHAPTER 7. ADDITIONAL PROCEDURES FOR NON-DIRECTOR ENVIRONMENTS

The follow sections outline some additional procedures for Red Hat OpenStack Platform environments not managed with director. These steps accommodate changes within the OpenStack Platform ecosystem and are best performed after an upgrade to Red Hat OpenStack Platform 10.

7.1. UPGRADING OPENSTACK TELEMETRY API TO A WSGI SERVICE

This step upgrades the OpenStack Telemetry (**ceilometer**) API to run as a Web Server Gateway Interface (WSGI) applet under **httpd** instead of a standalone service. This process disables the standalone **openstack-ceilometer-api** service and installs the necessary configuration to enable the WSGI applet.

1. Disable the OpenStack Telemetry service. This step varies based on whether you use highly available controller nodes or not.

   - For environments without high availability:

     ```
     $ sudo systemctl stop openstack-ceilometer-api
     ```

   - For environments with high availability:

     ```
     $ sudo pcs resource disable openstack-ceilometer-api
     ```

2. On each controller, copy the OpenStack Telemetry service WSGI applet (**/lib/python2.7/site-packages/ceilometer/api/app.wsgi**) to a new directory in **/var/www/cgi-bin/**. For example:

   ```
   $ sudo mkdir /var/www/cgi-bin/ceilometer
   $ cp /lib/python2.7/site-packages/ceilometer/api/app.wsgi /var/www/cgi-bin/ceilometer/app
   ```

3. On each controller, create a virtual host configuration file (**10-ceilometer_wsgi.conf**) for the OpenStack Telemetry service. Save this file in **/etc/httpd/conf.d/**. The contents of the virtual host file should resemble the following:

   ```
   Listen 8777

   <VirtualHost *:8777>
     DocumentRoot "/var/www/cgi-bin/ceilometer"

     <Directory "/var/www/cgi-bin/ceilometer">
       Options Indexes FollowSymLinks MultiViews
       AllowOverride None
       Require all granted
     </Directory>

     ErrorLog "/var/log/httpd/ceilometer_wsgi_error.log"
     ServerSignature Off
     CustomLog "/var/log/httpd/ceilometer_wsgi_access.log" combined

     SetEnvIf X-Forwarded-Proto https HTTPS=1
     WSGIApplicationGroup %{GLOBAL}
     WSGIDaemonProcess ceilometer group=ceilometer processes=1 threads=4
   ```

```
user=ceilometer
  WSGIProcessGroup ceilometer
  WSGIScriptAlias / "/var/www/cgi-bin/ceilometer/app"
</VirtualHost>
```

4. Restart the **httpd** service. This step varies based on whether you use highly available controller nodes or not.

   - For environments without high availability:

     ```
     $ sudo systemctl restart httpd
     ```

   - For environments with high availability:

     ```
     $ sudo pcs resource restart httpd
     ```

7.2. MIGRATING THE OPENSTACK TELEMETRY ALARMING DATABASE

As a part of the upgrade, **aodh-dbsync** tool creates a new MariaDB database. However, the previous database used MongoDB. This procedure migrates the past OpenStack Telemetry Alarming (**aodh**) service's database from MongoDB to MariaDB. Perform this step after upgrading your environment.

Edit the **/etc/aodh/aodh.conf** configuration file and change the database connection to a MariaDB database. For example:

```
[database]
connection = mysql+pymysql://username:password@host/aodh?charset=utf8
```

Run the following command to perform the migration:

```
$ sudo /usr/bin/aodh-data-migration \
  --nosql-conn `crudini --get /etc/ceilometer/ceilometer.conf database connection` \
  --sql-conn `crudini --get /etc/aodh/aodh.conf database connection`
```

This command migrates data from MongoDB (through **--nosql-conn**) to MariaDB (through **--sql-conn**).

CHAPTER 8. TROUBLESHOOTING DIRECTOR-BASED UPGRADES

This section provides advice for troubleshooting issues with both.

8.1. UNDERCLOUD UPGRADES

In situations where an Undercloud upgrade command (**openstack undercloud upgrade**) fails, use the following advice to locate the issue blocking upgrade progress:

- The **openstack undercloud upgrade** command prints out a progress log while it runs. If an error occurs at any point in the upgrade process, the command halts at the point of error. Use this information to identify any issues impeding upgrade progress.

- The **openstack undercloud upgrade** command runs Puppet to configure Undercloud services. This generates useful Puppet reports in the following directories:

  - **/var/lib/puppet/state/last_run_report.yaml** – The last Puppet reports generated for the Undercloud. This file shows any causes of failed Puppet actions.

  - **/var/lib/puppet/state/last_run_summary.yaml** – A summary of the **last_run_report.yaml** file.

  - **/var/lib/puppet/reports** – All Puppet reports for the Undercloud.
    Use this information to identify any issues impeding upgrade progress.

- Check for any failed services:

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

  If any services have failed, check their corresponding logs. For example, if **openstack-ironic-api** failed, use the following commands to check the logs for that service:

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

After correcting the issue impeding the Undercloud upgrade, rerun the upgrade command:

```
$ openstack undercloud upgrade
```

The upgrade command begins again and configures the Undercloud.

8.2. OVERCLOUD UPGRADES

In situations where an Overcloud upgrade process fails, use the following advice to locate the issue blocking upgrade progress:

- Check the Heat stack listing and identify any stacks that have an **UPDATE_FAILED** status. The following command identifies these stacks:

  ```
  $ heat stack-list --show-nested | awk -F "|" '{ print $3,$4 }' | grep "UPDATE_FAILED" | column
  -t
  ```

View the failed stack and its template to identify how the stack failed:

```
$ heat stack-show overcloud-Controller-qyoy54dyhrll-1-gtwy5bgta3np
$ heat template-show overcloud-Controller-qyoy54dyhrll-1-gtwy5bgta3np
```

- Check that Pacemaker is running correctly on all Controller nodes. If necessary, log into a Controller node and restart the Controller cluster:

```
$ sudo pcs cluster start
```

After correcting the issue impeding the Overcloud upgrade, rerun the **openstack overcloud deploy** command for the failed upgrade step you attempted. This following is an example of the first **openstack overcloud deploy** command in the upgrade process, which includes the **major-upgrade-pacemaker-init.yaml**:

```
$ openstack overcloud deploy --templates \
  -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-pacemaker-init.yaml
```

The **openstack overcloud deploy** retries the Overcloud stack update.

APPENDIX A. SAMPLE YAML FILES FOR NFV UPDATE

These sample YAML files upgrade OVS for an OVS-DPDK deployment.

A.1. RED HAT OPENSTACK PLATFORM 10 OVS 2.9 UPDATE FILES

A.1.1. post-install.yaml

```
heat_template_version: 2014-10-16

description: >
  Example extra config for post-deployment

parameters:
  servers:
    type: json
  ComputeHostnameFormat:
    type: string
    default: ""

resources:
  ExtraDeployments:
    type: OS::Heat::StructuredDeployments
    properties:
      servers:  {get_param: servers}
      config: {get_resource: ExtraConfig}
      # Do this on CREATE/UPDATE (which is actually the default)
      actions: ['CREATE', 'UPDATE']

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

```
        if hiera -c /etc/puppet/hiera.yaml service_names | grep -q -e neutron_ovs_dpdk_agent -e
neutron_sriov_agent -e neutron_sriov_host_config; then
            tuned_service_dependency
        fi
```