



Red Hat OpenStack Platform 11

Upgrading Red Hat OpenStack Platform

Upgrading a Red Hat OpenStack Platform environment

Red Hat OpenStack Platform 11 Upgrading Red Hat OpenStack Platform

Upgrading a Red Hat OpenStack Platform environment

OpenStack Team

rhos-docs@redhat.com

Legal Notice

Copyright © 2018 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Abstract

This document lays out the different methods through which users can upgrade from Red Hat OpenStack Platform 10 (Newton) to 11 (Ocata). 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	3
1.1. UPGRADE SCENARIO COMPARISON	3
1.2. REPOSITORY REQUIREMENTS	4
PART I. DIRECTOR-BASED ENVIRONMENTS	6
CHAPTER 2. DIRECTOR-BASED ENVIRONMENTS: PERFORMING UPDATES TO MINOR VERSIONS	7
2.1. PRE-UPDATE NOTES	7
2.1.1. General Recommendations	7
2.2. UPDATING RED HAT OPENSTACK PLATFORM	7
2.2.1. Updating the Undercloud Packages	7
2.2.2. Updating the Overcloud Images	8
2.2.3. Updating the Overcloud Packages	9
CHAPTER 3. DIRECTOR-BASED ENVIRONMENTS: PERFORMING A MAJOR VERSION UPGRADE	12
3.1. UPGRADE SUPPORT STATEMENT	12
3.2. LIVE MIGRATION UPDATES	14
3.3. CHECKING THE OVERCLOUD	14
3.4. UNDERCLOUD UPGRADE	15
3.4.1. Upgrading the Director	15
3.4.2. Upgrading the Overcloud Images	17
3.4.3. Using and Comparing Previous Template Versions	17
3.5. OVERCLOUD PRE-UPGRADE CONFIGURATION	18
3.5.1. Red Hat Subscription Details	18
3.5.2. Deprecated and New Composable Services	18
3.5.3. Manual Role Upgrades	20
3.5.4. Storage Backends	20
3.5.5. NFV Configuration	21
3.5.6. Overcloud Parameters	21
3.5.7. Custom Core Templates	22
3.6. OVERCLOUD UPGRADE	22
3.6.1. Upgrading Overcloud Nodes	22
3.6.2. Upgrading Object Storage Nodes	23
3.6.3. Upgrading Compute Nodes	24
3.6.4. Finalizing the Upgrade	26
3.7. POST-UPGRADE NOTES FOR THE OVERCLOUD	27
CHAPTER 4. TROUBLESHOOTING DIRECTOR-BASED UPGRADES	29
4.1. UNDERCLOUD UPGRADES	29
4.2. OVERCLOUD UPGRADES	29
PART II. NON-DIRECTOR ENVIRONMENTS	31
CHAPTER 5. UPGRADING RED HAT OPENSTACK PLATFORM MANUALLY	32
PART III. APPENDICES	33
APPENDIX A. SAMPLE NFV UPGRADE FILE	34

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 11 (Ocata)**.

Red Hat only supports upgrades to Red Hat OpenStack Platform 11 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 11 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 11. The following table provides a brief description of each.

Table 1.1. Upgrade Scenarios

Method	Description
Director-Based Environments: Performing Updates to Minor Versions	<p>This scenario is for updating from one minor version of Red Hat OpenStack Platform 11 to a newer version of Red Hat OpenStack Platform 11. This involves:</p> <ul style="list-style-type: none"> • Updating the director packages • Using the director to launch a package update on all nodes in the Overcloud.
Director-Based Environments: Performing Upgrades to Major Versions	<p>This scenario is for upgrading from a major versions of Red Hat OpenStack Platform. In this case, the procedure upgrades from version 10 to version 11. This involves:</p> <ul style="list-style-type: none"> • Updating the director packages • Using the director to provide a set of upgrade scripts on each node • Performing an upgrade of the Overcloud stack.

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 11 for RHEL 7 (RPMs)	rhel-7-server-openstack-11-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 Workstation (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 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.

PART I. DIRECTOR-BASED ENVIRONMENTS

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 11. This includes updating aspects of both the Undercloud and Overcloud.



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.
- 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.

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

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

4. The director uses the `openstack undercloud upgrade` command to update the Undercloud environment. Run the command:

```
$ openstack undercloud upgrade
```

Perform a reboot of the node to enable new system settings and refresh all undercloud services:

1. Reboot the node:

```
$ sudo reboot
```

2. Wait until the node boots.

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.

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

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

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

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 two steps:

1. Updating 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.
2. Performing 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, reboot each node using the ["Rebooting the Overcloud"](#) procedures in the *Director Installation and Usage* guide.

CHAPTER 3. DIRECTOR-BASED ENVIRONMENTS: PERFORMING A MAJOR VERSION UPGRADE



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 10 *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 undercloud and overcloud to the next major version. In this case, it is a upgrade from Red Hat OpenStack Platform 10 to Red Hat OpenStack Platform 11.

This procedure 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. Update any overcloud customizations, such as custom Heat templates and environment files.
4. Upgrade all nodes that support composable service upgrades.
5. Upgrade Object Storage nodes individually.
6. Upgrade Compute nodes individually.
7. Perform overcloud upgrade finalization.

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 11 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 11

Version	Overcloud Updating	Overcloud Deploying	Overcloud Scaling
Red Hat OpenStack Platform 11	Red Hat OpenStack Platform 11 and 10	Red Hat OpenStack Platform 11 and 10	Red Hat OpenStack Platform 11 and 10

If managing an older Overcloud version, use the following Heat template collections:

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

For example:

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

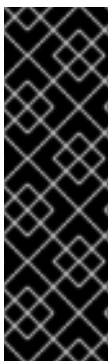
- Please contact Red Hat and request guidance and assistance on the upgrade process before proceeding if you feel uncomfortable about performing this upgrade.

3.2. LIVE MIGRATION UPDATES

Upgrading the Compute nodes requires live migration to ensure instances remain available during the upgrade. This requires the `OS::TripleO::Services::Sshd` service, which is a new service added to the default roles in latest version of Red Hat OpenStack Platform 10. To ensure live migration is enabled during the upgrade to Red Hat OpenStack Platform 11:

- Update to the undercloud to latest version of Red Hat OpenStack Platform 10.
- If using the default roles data file, check that each role includes the `OS::TripleO::Services::Sshd` service. If using a custom roles data file, add this new service to each role.
- Update the overcloud to latest version Red Hat OpenStack Platform 10 with the `OS::TripleO::Services::Sshd` service included.
- Start the upgrade to Red Hat OpenStack Platform 11.

This ensures the Compute nodes have SSH access to each other, which is required for the live migration process.



IMPORTANT

A recent security fix (CVE-2017-2637) disables live migration for previous versions of Red Hat OpenStack Platform. The `OS::TripleO::Services::Sshd` service resolves this issue for Red Hat OpenStack Platform 10 and later. For more information, see:

- ["KBase: Red Hat OpenStack Platform director \(TripleO\) CVE-2017-2637 bug and Red Hat OpenStack Platform"](#)
- ["CVE-2017-2637"](#)

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"
```

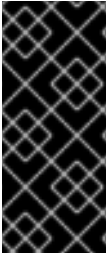
Replace `[CONTROLLER_IP]` with the IP address of a Controller node. This command refreshes the overcloud's Pacemaker cluster, waits 60 seconds, then reports the status of the cluster.

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
```



IMPORTANT

If using a standalone Keystone node on OpenStack Platform 10, the 'openstack-gnocchi-statsd' service might not have started correctly due to a race condition between keystone and gnocchi. Check the 'openstack-gnocchi-statsd' service on either Controller or Telemetry nodes and if it has failed, restart the service before upgrading the overcloud. This issue is addressed in [BZ#1447422](#).

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

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

3. Stop the main OpenStack Platform services:

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

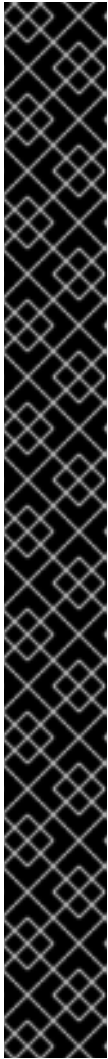


NOTE

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's main packages:

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



IMPORTANT

The default Provisioning/Control Plane network has changed from `192.0.2.0/24` to `192.168.24.0/24`. If you used default network values in your previous `undercloud.conf` file, your Provisioning/Control Plane network is set to `192.0.2.0/24`. This means you need to set certain parameters in your `undercloud.conf` file to continue using the `192.0.2.0/24` network. These parameters are:

- `local_ip`
- `network_gateway`
- `undercloud_public_vip`
- `undercloud_admin_vip`
- `network_cidr`
- `masquerade_network`
- `dhcp_start`
- `dhcp_end`.

Set the network values in `undercloud.conf` to ensure continued use of the `192.0.2.0/24` CIDR during future upgrades. Ensure your network configuration set correctly before running the `openstack undercloud upgrade` command.

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 as Overcloud stack data or data for existing nodes in your environment.

Perform a reboot of the node to enable new system settings and refresh all undercloud services:

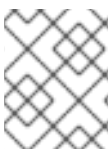
1. Reboot the node:

```
$ sudo reboot
```

2. Wait until the node boots.

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.

Verify the existence of your Overcloud and its nodes:

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

If necessary, review the configuration files on the director. The upgraded packages might have installed `.rpmnew` files appropriate to the Red Hat OpenStack Platform 11 version of the service.

3.4.2. Upgrading the Overcloud Images

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

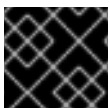
Import the latest images into the director and configure nodes to use the new images

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

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 (`newton`). 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/osp11
```

2. Move the previous version into its own directory:

```
$ mv /tmp/osp11/compat /tmp/osp10
```

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

```
$ diff -urN /tmp/osp10 /tmp/osp11
```

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

Before upgrading the overcloud, update its subscription details to ensure your environment uses the latest repositories. If using an environment file for Satellite registration, update the following parameters in the environment file:

- `rhel_reg_repos` - Repositories to enable for your Overcloud, including the new Red Hat OpenStack Platform 11 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 11 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 update this parameter if registering to Red Hat Satellite 6.

For more information and examples of the environment file format, see ["Overcloud Registration"](#) in the *Advanced Overcloud Customization* guide.

3.5.2. Deprecated and New Composable Services

The following sections apply if using a custom `roles_data.yaml` file to define your overcloud roles.

Remove the following deprecated services from your custom `roles_data.yaml` file:

Service	Description
<code>OS::TripleO::Services::Core</code>	This service acted as a core dependency for other Pacemaker services. This service has been removed to accommodate high availability composable services.
<code>OS::TripleO::Services::GlanceRegistry</code>	This service provided image metadata for OpenStack Image Storage (glance). This service has been removed due to its deprecation in the glance v2 API.
<code>OS::TripleO::Services::VipHosts</code>	This service configured the <code>/etc/hosts</code> file with node hostnames and IP addresses. This service is now integrated directly into the director's Heat templates.

Add the following new services to your custom `roles_data.yaml` file:

Service	Description
<code>OS::TripleO::Services::MySQLClient</code>	Configures the MariaDB client on a node, which provides database configuration for other composable services. Add this service to all roles with standalone composable services.
<code>OS::TripleO::Services::NovaPlacement</code>	Configures the OpenStack Compute (nova) Placement API. If using a standalone Nova role in your current overcloud, add this service to the role. Otherwise, add the service to the Controller role.
<code>OS::TripleO::Services::PankoApi</code>	Configures the OpenStack Telemetry Event Storage (panko) service. If using a standalone Telemetry role in your current overcloud, add this service to the role. Otherwise, add the service to the Controller role.
<code>OS::TripleO::Services::Sshd</code>	Configures SSH access across all nodes. Used for instance migration.

Update any additional parts of the overcloud that might require these new services such as:

- Custom ServiceNetMap Parameter** - 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*.

- **External Load Balancer** - If using an external load balancer, include the new services as a part of the external load balancer configuration. For more information, see [External Load Balancing for the Overcloud](#).

3.5.3. Manual Role Upgrades

The upgrade process provides staged upgrades for each composable service on all roles. However, some roles require an individual upgrade of their nodes to ensure availability of instances and service. These roles are:

Role	Description
Compute	Requires an individual upgrade for each node to ensure instance remain available. The process for these nodes involves migrating instances from each node before upgrading it.
ObjectStorage	Requires an individual upgrade for each node to ensure the Object Storage Service (swift) remains available to the overcloud.

This upgrade process uses the `upgrade-non-controller.sh` command to upgrade nodes in these roles.

The default `roles_data.yaml` file for Red Hat OpenStack Platform 11 marks these roles with the `disable_upgrade_deployment: True`, which excludes these roles during the main composable service upgrade process. This provides a method for upgrading the nodes in these roles individually. However, if using a custom `roles_data.yaml` file that contains these roles, make sure the `Compute` and `ObjectStorage` role definitions contain the `disable_upgrade_deployment: True` parameter. For example:

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

Use the `disable_upgrade_deployment` for other custom roles that require a manual upgrade, such as custom Compute roles.

3.5.4. Storage Backends

Some storage backends have changed from using configuration hooks to their own composable service. If using a custom storage backend, check the associated environment file in the `environments` directory for new parameters and resources. Update any custom environment files for your backends. For example:

- For the **NetApp Block Storage (cinder)** backend, use the new `environments/cinder-netapp-config.yaml` in your deployment.

- For the **Dell EMC Block Storage (cinder)** backend, use the new `environments/cinder-dellsc-config.yaml` in your deployment.
- For the **Dell EqualLogic Block Storage (cinder)** backend, use the new `environments/cinder-dellps-config.yaml` in your deployment.

For example, the **NetApp Block Storage (cinder)** backend used the following resources for these respective versions:

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

As a result, you now use the new `OS::TripleO::Services::CinderBackendNetApp` resource and its associated service template for this backend.

3.5.5. NFV Configuration

Follow these guidelines to upgrade from Red Hat OpenStack Platform 10 to Red Hat OpenStack Platform 11 when you have OVS-DPDK configured.



NOTE

Red Hat OpenStack Platform 11 operates in OVS client mode.

Begin your upgrade preparations based on the guidance from [Upgrading Red Hat OpenStack Platform](#) . Before you deploy the overcloud based on [Upgrading Composable Services](#), follow these steps:

1. Add the content from this sample [Appendix A, Sample NFV Upgrade File](#) file to any existing `post-install.yaml` file.
2. If your overcloud includes OVS version 2.5, modify the following parameters in your `.yaml` file, for example, in a `network-environment.yaml` file. This change is not in the `post-install.yaml` file:
 - a. Modify `HostCpuList` and `NeutronDpdkCoreList` to match your configuration. Ensure that you use only double quotation marks in the `yaml` file for these parameters.

```
HostCpusList: "0,16,8,24"
NeutronDpdkCoreList: "1,17,9,25"
```

- b. Modify `NeutronDpdkSocketMemory` to match your configuration. Ensure that you use only double quotation marks in the `yaml` file for this parameter.

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

- c. Modify `NeutronVhostuserSocketDir` as follows:

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

3.5.6. Overcloud Parameters

Note the following information about overcloud parameters for upgrades:

- 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*.
- Fixed VIP addresses for overcloud networks use new parameters and syntax. See ["Assigning Predictable Virtual IPs"](#) in the *Advanced Overcloud Customization* guide. If using external load balancing, see also ["Configuring Load Balancing Options"](#) in the *External Load Balancing for the Overcloud* guide.
- Some options for the `openstack overcloud deploy` command are now deprecated. You should substitute these options for their Heat parameter equivalents. For these parameter mappings, see ["Creating the Overcloud with the CLI Tools"](#) in the *Director Installation and Usage* guide.
- 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 available parameters, see the [Overcloud Parameters](#) guide.

3.5.7. Custom Core Templates

If using a modified version of the core Heat template collection from Red Hat OpenStack Platform 10, you need to re-apply your customizations to a copy of the Red Hat OpenStack Platform 11 version. To do this, use a `git` version control system similar to the one outlined in ["Using Customized Core Heat Templates"](#) from the *Advanced Overcloud Customization* guide.

Red Hat provides updates to the Heat template collection over subsequent releases. Using a modified template collection without a version control system can lead to a divergence between your custom copy and the original copy in `/usr/share/openstack-tripleo-heat-templates`.

As an alternative to using a custom Heat template collection, Red Hat recommends using the [Configuration Hooks](#) from the *Advanced Overcloud Customization* guide.

3.6. OVERCLOUD UPGRADE

3.6.1. Upgrading Overcloud Nodes

An overcloud upgrade requires an additional environment file (`major-upgrade-composable-steps.yaml`) to your deployment. This file provides a full upgrade to all nodes except roles marked with the `disable_upgrade_deployment: True` parameter.

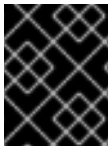
Run the `openstack overcloud deploy` command from your undercloud and include the `major-upgrade-composable-steps.yaml` environment file. Include all options and custom environment files relevant to your environment, such as network isolation and storage.

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-composable-steps.yaml \
--ntp-server pool.ntp.org
```

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



IMPORTANT

This step disables the Neutron server and L3 Agent during the upgrade. This means you cannot create new routers during this step.

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 so, perform a reboot of each node, use the reboot instructions from ["Rebooting the Overcloud"](#) in the *Director Installation and Usage* guide

During the deployment of the `major-upgrade-composable-steps.yaml` environment file, the director passes a special upgrade script to each node in roles marked with the `disable_upgrade_deployment: True` parameter. The next few sections show how to invoke this script from the undercloud and upgrade the remaining roles.



NOTE

This command removes deprecated composable services and installs new services for Red Hat OpenStack Platform 11. See [Section 3.5.2, "Deprecated and New Composable Services"](#) for a list of deprecated and new services.

3.6.2. Upgrading Object Storage Nodes

The director uses the `upgrade-non-controller.sh` command to run the upgrade script passed to the Object Storage nodes from the `major-upgrade-composable-steps.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
```

Upgrading each Object Storage node individually ensures the service remains available during the upgrade.

Wait until each Object Storage node completes its upgrade.

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 so, perform a reboot of each 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.

3.6.3. 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:

```
$ source ~/stackrc
$ 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 VMs from an Overcloud Compute Node](#)" 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-composable-steps.yaml` environment file. Upgrade each Compute node with the following command:

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

Replace `[NODE]` with the UUID or name 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 one of the following packages have updated their major or minor versions:

- `kernel`
- `openvswitch`
- `ceph-osd` (Hyper-converged environments)

If so, perform a reboot of the 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. Check whether the Compute node is enabled:

```
$ openstack compute service list
```

Repeat this process for each node individually until you have upgraded and rebooted all nodes.

After upgrading all Compute nodes, revert back to the `stackrc` access details:

```
$ source ~/stackrc
```

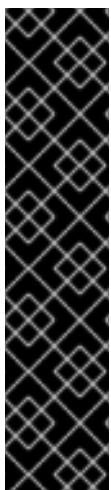


NOTE

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. 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-converge.yaml`), which you include using the `openstack overcloud deploy` command.



IMPORTANT

If your Red Hat OpenStack Platform environment is integrated with an external Ceph Storage Cluster from an earlier version (for example, Red Hat Ceph Storage 1.3), you need to enable backwards compatibility. To do so, create an environment file (for example, `/home/stack/templates/ceph-backwards-compatibility.yaml`) containing the following:

```
parameter_defaults:
  RbdDefaultFeatures: 1
```

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-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-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-converge.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.

3.7. POST-UPGRADE NOTES FOR THE OVERCLOUD

Be aware of the following notes after upgrading the Overcloud to Red Hat OpenStack Platform 11:

- If necessary, review the resulting configuration files on the overcloud nodes. The upgraded packages might have installed `.rpmnew` files appropriate to the Red Hat OpenStack Platform 11 version of the service.
- OpenStack Block Storage (cinder) API uses a new `sizelimit` filter in Red Hat OpenStack Platform 11. However, following an upgrade path from OpenStack Platform 9 to OpenStack Platform 10 to OpenStack Platform 11 might not update to this new filter. Check the filter on Controller or Cinder API nodes using the following command:

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

The filter should appear as the following:

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

If not, replace the value in the `/etc/cinder/api-paste.ini` file of each Controller or Cinder API node and restart the `httpd` service:

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

- 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
```

- 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
```


CHAPTER 4. TROUBLESHOOTING DIRECTOR-BASED UPGRADES

This section provides advice for troubleshooting issues with both the undercloud and overcloud.

4.1. 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 and saves it to `.instack/install-undercloud.log`. 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.

4.2. OVERCLOUD UPGRADES

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

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

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

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

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

- Check the configuration log files for any failures. The `/var/run/heat-config/deployed/` directory on each node contains these logs. These files are named in date order and are separated into standard output (`*-stdout.log`) and error output (`*-stderr.log`).



NOTE

The director performs a set of validation checks before the upgrade process to make sure the overcloud is in a good state. If the upgrade has failed and you want to retry, you might need to disable these validation checks. To disable these checks, temporarily add the `SkipUpgradeConfigTags: [validation]` to the `parameter_defaults` section of an environment file included with your overcloud.

After correcting the issue impeding the Overcloud upgrade, check that no resources have an **IN_PROGRESS** status:

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

If any resources have an **IN_PROGRESS** status, wait until they either complete or fail.

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-composable-steps.yaml`:

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

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

PART II. NON-DIRECTOR ENVIRONMENTS

CHAPTER 5. UPGRADING RED HAT OPENSTACK PLATFORM MANUALLY

Instructions for upgrading non-director environments manually are not available at the moment due to a known Apache configuration issue in the `openstack-nova-placement-api` package ([BZ#1434944](#)). Because of this issue, manually upgrading Compute results in the following error message when Compute nodes attempt to report to the Placement API:

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

Director-based deployments are not affected by this issue.

PART III. APPENDICES

APPENDIX A. SAMPLE NFV UPGRADE FILE

```

heat_template_version: 2014-10-16

description: >
  Example extra config for post-deployment

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

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

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

```

```

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

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

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

```

```

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

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

```