



Red Hat OpenStack Platform 12

Upgrading Red Hat OpenStack Platform

Upgrading a Red Hat OpenStack Platform environment

Red Hat OpenStack Platform 12 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 11 (Ocata) to 12 (Pike). 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 GOALS	3
1.2. UPGRADE PATH	3
1.3. REPOSITORIES	4
CHAPTER 2. PREPARING FOR AN OPENSTACK PLATFORM UPGRADE	6
2.1. SUPPORT STATEMENT	6
2.2. GENERAL UPGRADE TIPS	6
2.3. VALIDATING THE UNDERCLOUD BEFORE AN UPGRADE	7
2.4. VALIDATING THE OVERCLOUD BEFORE AN UPGRADE	7
2.5. BACKING UP THE UNDERCLOUD	10
2.6. UPDATING THE CURRENT UNDERCLOUD PACKAGES	11
2.7. UPDATING THE CURRENT OVERCLOUD IMAGES	12
2.8. UPDATING THE CURRENT OVERCLOUD PACKAGES	12
CHAPTER 3. UPGRADING THE UNDERCLOUD	15
3.1. UPGRADING THE UNDERCLOUD NODE	15
3.2. UPGRADING THE OVERCLOUD IMAGES	16
3.3. COMPARING PREVIOUS TEMPLATE VERSIONS	17
CHAPTER 4. PREPARING FOR THE OVERCLOUD UPGRADE	18
4.1. PREPARING OVERCLOUD REGISTRATION DETAILS	18
4.2. PREPARING FOR CONTAINERIZED SERVICES	18
4.3. PREPARING FOR NEW COMPOSABLE SERVICES	19
4.4. PREPARING FOR COMPOSABLE NETWORKS	20
4.5. PREPARING FOR DEPRECATED PARAMETERS	21
4.6. PREPARING FOR CEPH STORAGE NODE UPGRADES	22
4.7. PREPARING FOR HYPER-CONVERGED INFRASTRUCTURE (HCI) UPGRADES	23
4.8. PREPARING ACCESS TO THE UNDERCLOUD'S PUBLIC API OVER SSL/TLS	25
4.9. PREPARING FOR PRE-PROVISIONED NODES UPGRADE	26
4.10. PREPARING AN NFV-CONFIGURED OVERCLOUD	27
4.11. GENERAL CONSIDERATIONS FOR OVERCLOUD UPGRADES	28
CHAPTER 5. UPGRADING THE OVERCLOUD	29
5.1. UPGRADING THE OVERCLOUD NODES	29
5.2. UPGRADING THE OBJECT STORAGE NODES	30
5.3. UPGRADING THE COMPUTE NODES	31
5.4. FINALIZING THE UPGRADE	33
CHAPTER 6. EXECUTING POST UPGRADE STEPS	35
6.1. INCLUDING THE UNDERCLOUD CA ON NEW OVERCLOUD NODES	35
6.2. GENERAL CONSIDERATIONS AFTER AN OVERCLOUD UPGRADE	36
CHAPTER 7. KEEPING OPENSTACK PLATFORM UPDATED	37
7.1. VALIDATING THE UNDERCLOUD BEFORE AN UPDATE	37
7.2. VALIDATING THE OVERCLOUD BEFORE AN UPDATE	38
7.3. KEEPING THE UNDERCLOUD UPDATED	40
7.4. KEEPING THE OVERCLOUD IMAGES UPDATED	41
7.5. KEEPING THE OVERCLOUD UPDATED	42

CHAPTER 1. INTRODUCTION

This document provides a workflow to help upgrade your Red Hat OpenStack Platform environment to the latest major version and keep it updated with minor releases of that version.

1.1. UPGRADE GOALS

Red Hat OpenStack Platform provides a method to upgrade your current environment to the next major version. This guide aims to upgrade and update your environment to the latest **Red Hat OpenStack Platform 12 (Pike)** release.

The upgrade also upgrades to the following:

- **Operating System:** Red Hat Enterprise Linux 7.4
- **Networking:** Open vSwitch 2.6
- **Ceph Storage:** The process upgrades to the latest version of Red Hat Ceph Storage 2 and switches to a **ceph-ansible** deployment.



WARNING

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

1.2. UPGRADE PATH

The following represents the upgrade path of a Red Hat OpenStack Platform environment:

Table 1.1. OpenStack Platform Upgrade Path

	Task	Version	When
1	Backup your current undercloud and overcloud.	Red Hat OpenStack Platform 11	Once
2	Update your current undercloud and overcloud to the latest minor release.	Red Hat OpenStack Platform 11	Once
3	Upgrade your current undercloud to the latest major release.	Red Hat OpenStack Platform 11 to 12	Once
4	Prepare your overcloud, including updating any relevant custom configuration.	Red Hat OpenStack Platform 11 to 12	Once
5	Upgrade your current overcloud to the latest major release.	Red Hat OpenStack Platform 11 to 12	Once

Task	Version	When	
6	Update your undercloud and overcloud to the latest minor release on a regular basis.	Red Hat OpenStack Platform 12	Ongoing

1.3. REPOSITORIES

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 for x86_64 systems.
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 12 for RHEL 7 (RPMs)	rhel-7-server-openstack-12-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.

Name	Repository	Description of Requirement
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 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.

CHAPTER 2. PREPARING FOR AN OPENSTACK PLATFORM UPGRADE

This process prepares your OpenStack Platform environment for a full update.

2.1. 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 being 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.
- 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 12 can manage previous Overcloud versions of Red Hat OpenStack Platform. See the support matrix below for information.

Table 2.1. Support Matrix for Red Hat OpenStack Platform director 12

Version	Overcloud Updating	Overcloud Deploying	Overcloud Scaling
Red Hat OpenStack Platform 12	Red Hat OpenStack Platform 12 and 11	Red Hat OpenStack Platform 12 and 11	Red Hat OpenStack Platform 12 and 11

2.2. GENERAL UPGRADE TIPS

The following are some tips to help with your upgrade:

- 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.

2.3. VALIDATING THE UNDERCLOUD BEFORE AN UPGRADE

The following is a set of steps to check the functionality of your **Red Hat OpenStack Platform 11** undercloud before an upgrade.

Procedure

1. Source the undercloud access details:

```
$ source ~/stackrc
```

2. Check for failed Systemd services:

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

3. Check the undercloud free space:

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

Use the "[Undercloud Requirements](#)" as a basis to determine if you have adequate free space.

4. Check that clocks are synchronized on the undercloud:

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

5. Check the undercloud network services:

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

All agents should be **Alive** and their state should be **UP**.

6. Check the undercloud compute services:

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

All agents' status should be **enabled** and their state should be **up**

Related Information

- The following solution article shows how to remove deleted stack entries in your OpenStack Orchestration (heat) database: <https://access.redhat.com/solutions/2215131>

2.4. VALIDATING THE OVERCLOUD BEFORE AN UPGRADE

The following is a set of steps to check the functionality of your **Red Hat OpenStack Platform 11** overcloud before an upgrade.

Procedure

1. Source the undercloud access details:

```
$ source ~/stackrc
```

2. Check the status of your bare metal nodes:

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

All nodes should have a valid power state (**on**) and maintenance mode should be **false**.

3. Check for failed Systemd services:

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

4. Check the HAProxy connection to all services. Obtain the Control Plane VIP address and authentication details for the **haproxy.stats** service:

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

Use these details in the following cURL request:

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

Replace **<PASSWORD>** and **<IP ADDRESS>** details with the respective details from the **haproxy.stats** service. The resulting list shows the OpenStack Platform services on each node and their connection status.

5. Check overcloud database replication health:

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

6. Check RabbitMQ cluster health:

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

7. Check Pacemaker resource health:

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

Look for:

- All cluster nodes **online**.
- No resources **stopped** on any cluster nodes.
- No **failed** pacemaker actions.

8. Check the disk space on each overcloud node:

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

9. Check overcloud Ceph Storage cluster health. The following command runs the **ceph** tool on a Controller node to check the cluster:

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

10. Check Ceph Storage OSD for free space. The following command runs the **ceph** tool on a Controller node to check the free space:

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

11. Check that clocks are synchronized on overcloud nodes

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

12. Source the overcloud access details:

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

13. Check the overcloud network services:

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

All agents should be **Alive** and their state should be **UP**.

14. Check the overcloud compute services:

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

All agents' status should be **enabled** and their state should be **up**

15. Check the overcloud volume services:

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

All agents' status should be **enabled** and their state should be **up**.

Related Information

- Review the article ["How can I verify my OpenStack environment is deployed with Red Hat recommended configurations?"](#). This article provides some information on how to check your Red Hat OpenStack Platform environment and tune the configuration to Red Hat's recommendations.
- Review the article ["Database Size Management for Red Hat Enterprise Linux OpenStack Platform"](#) to check and clean unused database records for OpenStack Platform services on the overcloud.

2.5. BACKING UP THE UNDERCLOUD

A full undercloud backup includes the following databases and files:

- All MariaDB databases on the undercloud node
- MariaDB configuration file on the undercloud (so that you can accurately restore databases)
- All swift data in `/srv/node`
- All data in the stack user home directory: `/home/stack`
- The undercloud SSL certificates:
 - `/etc/pki/ca-trust/source/anchors/ca.crt.pem`
 - `/etc/pki/instack-certs/undercloud.pem`



NOTE

Confirm that you have sufficient disk space available before performing the backup process. The tarball can be expected to be at least 3.5 GB, but this is likely to be larger.

Procedure

1. Log into the undercloud as the **root** user.
2. Back up the database:

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

3. Archive the database backup and the configuration files:

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

This creates a file named **undercloud-backup-[timestamp].tar.gz**.

Related Information

- If you need to restore the undercloud backup, see the ["Restore"](#) chapter in the *Back Up and Restore the Director Undercloud* guide.

2.6. UPDATING THE CURRENT UNDERCLOUD PACKAGES

The director provides commands to update the packages on the undercloud node. This allows you to perform a minor update within the current version of your OpenStack Platform environment. This is a minor update within **Red Hat OpenStack Platform 11**.

Prerequisites

- You have performed a backup of the undercloud.

Procedure

1. Log into the director as the **stack** user.
2. Update the **python-tripleoclient** package and its dependencies to ensure you have the latest scripts for the minor version update:

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

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

```
$ openstack undercloud upgrade
```

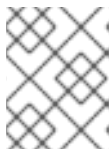
4. Reboot the node:

```
$ sudo reboot
```

5. Wait until the node boots.

6. 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.

7. Verify the existence of your overcloud and its nodes:

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

2.7. UPDATING THE CURRENT OVERCLOUD IMAGES

The undercloud update process might download new image archives from the **rhosp-director-images** and **rhosp-director-images-ipa** packages. This process updates these images on your undercloud within **Red Hat OpenStack Platform 11**.

Prerequisites

- You have updated to the latest minor release of your current undercloud version.

Procedure

1. Check the **yum** log to determine if new image archives are available:

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

2. 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/*
```

3. 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
```

4. 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 " ")
```

5. 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.

2.8. UPDATING THE CURRENT OVERCLOUD PACKAGES

The director provides commands to update the packages on all overcloud nodes. This allows you to perform a minor update within the current version of your OpenStack Platform environment. This is a minor update within **Red Hat OpenStack Platform 11**.

Prerequisites

- You have updated to the latest minor release of your current undercloud version.
- You have performed a backup of the overcloud.

Procedure

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

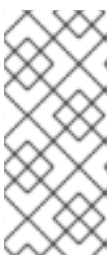
```
$ openstack overcloud deploy --update-plan-only \
  --templates \
  -e /usr/share/openstack-tripleo-heat-
templates/environments/network-isolation.yaml \
  -e /home/stack/templates/network-environment.yaml \
  -e /home/stack/templates/storage-environment.yaml \
  -e /home/stack/templates/rhel-registration/environment-rhel-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

The **--update-plan-only** only updates the Overcloud plan stored in the director. Use the **-e** option 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. Perform a package update on all nodes using the **openstack overcloud update** command. For example:

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

The **-i** runs an interactive mode to update each node. When the update process completes a node update, the script provides a breakpoint for you to confirm. Without the **-i** option, the update remains paused at the first breakpoint. Therefore, it is mandatory to include the **-i** option.



NOTE

Running an update on all nodes in parallel can 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.

3. The update process starts. 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.

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

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

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

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

6. The update process does not reboot any nodes in the Overcloud automatically. Updates to the kernel or Open vSwitch require a reboot. 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 Nodes"](#) procedures in the *Director Installation and Usage* guide.

CHAPTER 3. UPGRADING THE UNDERCLOUD

This process upgrades the undercloud and its overcloud images to **Red Hat OpenStack Platform 12**.

3.1. UPGRADING THE UNDERCLOUD NODE

You need to upgrade the undercloud before upgrading the overcloud. This procedure upgrades the undercloud toolset and the core Heat template collection.

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

Prerequisites

- You have read the upgrade support statement.
- You have updated to the latest minor version of your undercloud version.

Procedure

1. Log into the director as the **stack** user.
2. Disable the current OpenStack Platform repository:

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

3. Enable the new OpenStack Platform repository:

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

4. Run **yum** to upgrade the director's main packages:

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

5. Edit the **/home/stack/undercloud.conf** file and check that the **enabled_drivers** parameter does not contain the **pxe_ssh** driver. This driver is deprecated in favor of the Virtual Bare Metal Controller (VBMC) and removed from Red Hat OpenStack Platform. For information on switching **pxe_ssh** nodes to VBMC, see ["Virtual Bare Metal Controller \(VBMC\)"](#) in the *Director Installation and Usage* guide.

6. Run the following command to upgrade the undercloud:

```
$ openstack undercloud upgrade
```

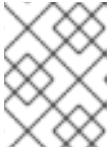
This command upgrades the director's packages, refreshes the director's configuration, and populates any settings that are unset since the version change. This command does not delete any stored data, such as Overcloud stack data or data for existing nodes in your environment.

7. Reboot the node:

```
$ sudo reboot
```

8. Wait until the node boots.
9. 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.

10. Verify the existence of your overcloud and its nodes:

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

3.2. UPGRADING THE OVERCLOUD IMAGES

You need to replace your current overcloud images with new versions. The new images ensure the director can introspect and provision your nodes using the latest version of OpenStack Platform software.

Prerequisites

- You have upgraded the undercloud to the latest version.

Procedure

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

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

2. Extract the archives:

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

3. Import the latest images into the director:

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

4. Configure your nodes to use the new images:

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

5. Verify the existence of the new images:

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



IMPORTANT

When deploying overcloud nodes, ensure the Overcloud image version corresponds to the respective Heat template version. For example, only use the OpenStack Platform 12 images with the OpenStack Platform 12 Heat templates.

3.3. 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. This procedure shows how to compare these versions so you can identify changes that might affect your overcloud upgrade.

Procedure

1. Install the **openstack-tripleo-heat-templates-compat** package:

```
$ 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 (**ocata**). 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.

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

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

3. Move the previous version into its own directory:

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

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

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

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.

CHAPTER 4. PREPARING FOR THE OVERCLOUD UPGRADE

This process prepares the overcloud for the upgrade process.

Prerequisites

- You have upgraded the undercloud to the latest version.

4.1. PREPARING OVERCLOUD REGISTRATION DETAILS

You need to provide the overcloud with the latest subscription details to ensure the overcloud consumes the latest packages during the upgrade process.

Prerequisites

- A subscription containing the latest OpenStack Platform repositories.
- If using activation keys for registration, create a new activation key including the new OpenStack Platform repositories.

Procedure

1. Edit the environment file containing your registration details. For example:

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

2. Edit the following parameter values:

rhel_reg_repos

Update to include the new repositories for Red Hat OpenStack Platform 12.

rhel_reg_activation_key

Update the activation key to access the Red Hat OpenStack Platform 12 repositories.

rhel_reg_sat_repo

If using a newer version of Red Hat Satellite 6, update the repository containing Satellite 6's management tools.

3. Save the environment file.

Related Information

- For more information about registration parameters, see "[Registering the Overcloud with an Environment File](#)" in the *Advanced Overcloud Customizations* guide.

4.2. PREPARING FOR CONTAINERIZED SERVICES

Red Hat OpenStack Platform now uses containers to host and run OpenStack services. This requires you to:

- Configure a container image source, such as a registry
- Generate an environment file with image locations on your image source

- Add the environment file to your overcloud deployment

For full instructions about generating this environment file for different use case, see "[Configuring Container Registry Details](#)" in the *Director Installation and Usage* guide.

The resulting environment file (`/home/stack/templates/overcloud_images.yaml`) contains parameters that point to the container image locations for each service. Include this file in all future deployment operations.

4.3. PREPARING FOR NEW COMPOSABLE SERVICES

This version of Red Hat OpenStack Platform contains new composable services. If using a custom `roles_data` file, include these new services in their applicable roles.

All Roles

The following new services apply to all roles.

OS::TripleO::Services::CertmongerUser

Allows the overcloud to require certificates from Certmonger. Only used if enabling TLS/SSL communication.

OS::TripleO::Services::Docker

Installs **docker** to manage containerized services.

OS::TripleO::Services::MySQLClient

Installs the overcloud database client tool.

OS::TripleO::Services::ContainersLogrotateCron

Installs the **logrotate** service for container logs.

OS::TripleO::Services::Securetty

Allows configuration of **securetty** on nodes. Enabled with the `environments/securetty.yaml` environment file.

OS::TripleO::Services::Tuned

Enables and configures the Linux tuning daemon (**tuned**).

Specific Roles

The following new services apply to specific roles:

OS::TripleO::Services::Clustercheck

Required on any role that also uses the **OS::TripleO::Services::MySQL** service, such as the **Controller** or standalone **Database** role.

OS::TripleO::Services::Iscsid

Configures the **iscsid** service on the **Controller**, **Compute**, and **BlockStorage** roles.

OS::TripleO::Services::NovaMigrationTarget

Configures the migration target service on **Compute** nodes.

If using a custom `roles_data` file, add these services to required roles.

In addition, see the "[Service Architecture: Standalone Roles](#)" section in the *Advanced Overcloud Customization* guide for updated lists of services for specific custom roles.

4.4. PREPARING FOR COMPOSABLE NETWORKS

This version of Red Hat OpenStack Platform introduces a new feature for composable networks. If using a custom **roles_data** file, edit the file to add the composable networks to each role. For example, for Controller nodes:

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

Check the default **/usr/share/openstack-tripleo-heat-templates/roles_data.yaml** file for further examples of syntax. Also check the example role snippets in **/usr/share/openstack-tripleo-heat-templates/roles**.

The following table provides a mapping of composable networks to custom standalone roles:

Role	Networks Required
Ceph Storage Monitor	Storage, StorageMgmt
Ceph Storage OSD	Storage, StorageMgmt
Ceph Storage RadosGW	Storage, StorageMgmt
Cinder API	InternalApi
Compute	InternalApi, Tenant, Storage
Controller	External, InternalApi, Storage, StorageMgmt, Tenant
Database	InternalApi
Glance	InternalApi
Heat	InternalApi
Horizon	InternalApi
Ironic	None required. Uses the Provisioning/Control Plane network for API.
Keystone	InternalApi
Load Balancer	External, InternalApi, Storage, StorageMgmt, Tenant
Manila	InternalApi

Role	Networks Required
Message Bus	InternalApi
Networker	InternalApi, Tenant
Neutron API	InternalApi
Nova	InternalApi
OpenDaylight	External, InternalApi, Tenant
Redis	InternalApi
Sahara	InternalApi
Swift API	Storage
Swift Storage	StorageMgmt
Telemetry	InternalApi

4.5. PREPARING FOR DEPRECATED PARAMETERS

Note that the following parameters are deprecated and have been replaced with role-specific parameters:

Old Parameter	New Parameter
controllerExtraConfig	ControllerExtraConfig
OvercloudControlFlavor	OvercloudControllerFlavor
controllerImage	ControllerImage
NovaImage	ComputeImage
NovaComputeExtraConfig	ComputeExtraConfig
NovaComputeServerMetadata	ComputeServerMetadata
NovaComputeSchedulerHints	ComputeSchedulerHints
NovaComputeIPs	ComputeIPs
SwiftStorageServerMetadata	ObjectStorageServerMetadata

Old Parameter	New Parameter
SwiftStorageIPs	ObjectStorageIPs
SwiftStorageImage	ObjectStorageImage
OvercloudSwiftStorageFlavor	OvercloudObjectStorageFlavor

Update these parameters in your custom environment files.

4.6. PREPARING FOR CEPH STORAGE NODE UPGRADES

Due to the upgrade to containerized services, the method for installing and updating Ceph Storage nodes has changed. Ceph Storage configuration now uses a set of playbooks in the **ceph-ansible** packages, which you install on the undercloud.

Prerequisites

- Your overcloud has a director-managed Ceph Storage cluster.

Procedure

1. Install the **ceph-ansible** package to the undercloud:

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

2. Check that you are using the latest resources and configuration in your storage environment file. This requires the following changes:
 - a. The **resource_registry** uses containerized services from the **docker/services** subdirectory of your core Heat template collection. For example:

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

- a. Use the new **CephAnsibleDisksConfig** parameter to define how your disks are mapped. Previous versions of Red Hat OpenStack Platform used the **ceph::profile::params::osds** hieradata to define the OSD layout. Convert this hieradata to the structure of the **CephAnsibleDisksConfig** parameter. For example, if your hieradata contained the following:

```
parameter_defaults:
  ExtraConfig:
    ceph::profile::params::osd_journal_size: 512
    ceph::profile::params::osds:
```

```
'/dev/sdb': {}
'/dev/sdc': {}
'/dev/sdd': {}
```

Then the **CephAnsibleDisksConfig** would look like this:

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

For a full list of OSD disk layout options used in **ceph-ansible**, view the sample file in **/usr/share/ceph-ansible/group_vars/osds.yml.sample**.

Related Information

- Note that environments using Hyper-Converged Infrastructure require some additional configuration. See [Section 4.7, “Preparing for Hyper-Converged Infrastructure \(HCI\) Upgrades”](#).
- For more information about **ceph-ansible** management with OpenStack Platform director, see the *Deploying an Overcloud with Containerized Red Hat Ceph* guide.

4.7. PREPARING FOR HYPER-CONVERGED INFRASTRUCTURE (HCI) UPGRADES

On a Hyper-Converged Infrastructure (HCI), the Ceph Storage and Compute services are collocated within a single role. However, you upgrade the HCI nodes the same way as a regular Compute nodes. In this situation, you delay migrating the Ceph Storage services to containerized services until the core packages have been installed and the container services enabled.

Prerequisites

- Your overcloud uses a collocated role containing Compute and Ceph Storage.

Procedure

1. Edit the environment file containing your Ceph Storage configuration.
2. Ensure the **resource_registry** uses the Puppet resources. For example:

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

**NOTE**

Use the contents of the `/usr/share/openstack-tripleo-heat-templates/environments/puppet-ceph.yaml` as an example.

3. Upgrade your Controller-based nodes to containerized services using the instructions in [Section 5.1, “Upgrading the Overcloud Nodes”](#).
4. Upgrade your HCI nodes using the instructions in [Section 5.3, “Upgrading the Compute Nodes”](#)
5. Edit the **resource_registry** in your Ceph Storage configuration to use the containerized services:

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

**NOTE**

Use the contents of the `/usr/share/openstack-tripleo-heat-templates/environments/storage-environment.yaml` as an example.

6. Add the **CephAnsiblePlaybook** parameter to the **parameter_defaults** section of your storage environment file:

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

7. Add the **CephAnsibleDisksConfig** parameter to the **parameter_defaults** section of your storage environment file and define the disk layout. For example:

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

8. Finalize the upgrade of your overcloud using the instructions in [Section 5.4, “Finalizing the Upgrade”](#).

Related Information

- For more information about configuring **ceph-ansible** management with OpenStack Platform director, see the *Deploying an Overcloud with Containerized Red Hat Ceph* guide.

4.8. PREPARING ACCESS TO THE UNDERCLOUD'S PUBLIC API OVER SSL/TLS

The overcloud requires access to the undercloud's OpenStack Object Storage (swift) Public API during the upgrade. If your undercloud uses a self-signed certificate, you need to add the undercloud's certificate authority to each overcloud node.

Prerequisites

- The undercloud uses SSL/TLS for its Public API

Procedure

1. The director's dynamic Ansible script has updated to the OpenStack Platform 12 version, which uses the **RoleNetHostnameMap** Heat parameter in the overcloud plan to define the inventory. However, the overcloud currently uses the OpenStack Platform 11 template versions, which do not have the **RoleNetHostnameMap** parameter. This means you need to create a temporary static inventory file, which you can generate with the following command:

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

2. Create an Ansible playbook (**undercloud-ca.yml**) that contains the following:

```
---
- name: Add undercloud CA to overcloud nodes
  hosts: all
  user: heat-admin
  become: true
  tasks:
    - name: Copy undercloud CA
      copy:
        src: ca.crt.pem
        dest: /etc/pki/ca-trust/source/anchors/
    - name: Update trust
      command: "update-ca-trust extract"
    - name: Get the hostname of the undercloud
      delegate_to: 127.0.0.1
      command: hostname
      register: undercloud_hostname
    - name: Verify URL
      uri:
        url: https://{{ undercloud_hostname.stdout
}}:13808/healthcheck
        return_content: yes
        register: verify
    - name: Report output
      debug:
        msg: "{{ ansible_hostname }}" can access the undercloud's
Public API"
        when: verify.content == "OK"
```

This playbook contains multiple tasks that perform the following on each node:

- Copy the undercloud's certificate authority file (**ca.crt.pem**) to the overcloud node. The name of this file and its location might vary depending on your configuration. This example uses the name and location defined during the self-signed certificate procedure (see ["SSL/TLS Certificate Configuration"](#) in the *Director Installation and Usage* guide).
 - Execute the command to update the certificate authority trust database on the overcloud node.
 - Checks the undercloud's Object Storage Public API from the overcloud node and reports if successful.
3. Run the playbook with the following command:

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

This uses the temporary inventory to provide Ansible with your overcloud nodes.

4. The resulting Ansible output should show a debug message for node. For example:

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

Related Information

- For more information on running Ansible automation on your overcloud, see ["Running Ansible Automation"](#) in the *Director Installation and Usage* guide.

4.9. PREPARING FOR PRE-PROVISIONED NODES UPGRADE

Pre-provisioned nodes are nodes created outside of the director's management. An overcloud using pre-provisioned nodes requires some additional steps prior to upgrading.

Prerequisites

- The overcloud uses pre-provisioned nodes.

Procedure

1. Run the following commands to save a list of node IP addresses in the **OVERCLOUD_HOSTS** environment variable:

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

2. Run the following script:

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

3. Proceed with the upgrade.

4. When upgrading Compute or Object Storage nodes, use the following:
 - a. Use the **-U** option with the **upgrade-non-controller.sh** script and specify the **stack** user. This is because the default user for pre-provisioned nodes is **stack** and not **heat-admin**.
 - b. Use the node's IP address with the **--upgrade** option. This is because the node are not managed with the director's Compute (nova) and Bare Metal (ironic) services and do not have a node name.
For example:

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

Related Information

- For more information on pre-provisioned nodes, see ["Configuring a Basic Overcloud using Pre-Provisioned Nodes"](#) in the *Director Installation and Usage* guide.

4.10. PREPARING AN NFV-CONFIGURED OVERCLOUD

When you upgrade from Red Hat OpenStack Platform 11 to Red Hat OpenStack Platform 12, the OVS package also upgrades from version 2.6 to version 2.7. To support this transition when you have OVS-DPDK configured, follow these guidelines.



NOTE

Red Hat OpenStack Platform 12 operates in OVS client mode.

Prerequisites

- Your overcloud uses Network Functions Virtualization (NFV).

Procedure

When you upgrade the Overcloud from Red Hat OpenStack Platform 11 to Red Hat OpenStack Platform 12 with OVS-DPDK configured, you must set the following additional parameters in an environment file.

- In the **parameter_defaults** section, add a network deployment parameter to run **os-net-config** during the upgrade process to associate the OVS 2.7 PCI address with DPDK ports:

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

The parameter name must match the name of the role you use to deploy DPDK. In this example, the role name is **Compute** so the parameter name is **ComputeNetworkDeploymentActions**.



NOTE

This parameter is not needed after the initial upgrade and should be removed from the environment file.

- In the **resource_registry** section, override the **ComputeNeutronOvsAgent** service to the **neutron-ovs-dpdk-agent** puppet service:

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

Red Hat OpenStack Platform 12 added a new service

(**OS::TripleO::Services::ComputeNeutronOvsDpdk**) to support the addition of the new **ComputeOvsDpdk** role. The example above maps this externally for upgrades.

Include the resulting environment file as part of the **openstack overcloud deploy** command in [Section 5.1, “Upgrading the Overcloud Nodes”](#).

4.11. GENERAL CONSIDERATIONS FOR OVERCLOUD UPGRADES

The following items are a set of general reminders to consider before upgrading the overcloud:

Custom ServiceNetMap

If upgrading an Overcloud with a custom **ServiceNetMap**, ensure you 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 Balancing

If using external load balancing, check for any new services to add to your load balancer. See also ["Configuring Load Balancing Options"](#) in the *External Load Balancing for the Overcloud* guide for service configuration.

Deprecated Deployment Options

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.

CHAPTER 5. UPGRADING THE OVERCLOUD

This process upgrades the overcloud.

Prerequisites

- You have upgraded the undercloud to the latest version.
- You have prepared your custom environment files to accommodate the changes in the upgrade.

5.1. UPGRADING THE OVERCLOUD NODES

The **major-upgrade-composable-steps-docker.yaml** environment file upgrades all composable services on all custom roles, except for any roles with **disable_upgrade_deployment: True** in the **roles_data** file. These nodes are updated with a separate process.

Prerequisites

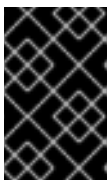
- You have upgraded the undercloud to the latest version.
- You have prepared your custom environment files to accommodate the changes in the upgrade.

Procedure

1. Run the **openstack overcloud deploy** command and include:
 - All options and custom environment files relevant to your environment, such as network isolation and storage.
 - The **overcloud_images.yaml** environment file generated in [Section 4.2, “Preparing for Containerized Services”](#).
 - The **major-upgrade-composable-steps-docker.yaml** environment file.
 For example:

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

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



IMPORTANT

The upgrade disables the OpenStack Networking (neutron) server and L3 Agent. This means you cannot create new routers during this step. You can still access instances during this period.

3. Check if all services are active. For example, to check services on a Controller node:

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

Related Information

- If you encounter any issues after completing this step, please contact Red Hat and request guidance and assistance.

5.2. UPGRADING THE OBJECT STORAGE NODES

Standalone Object Storage nodes are not included in the main overcloud node upgrade process because you need to update each node individually to retain the services' activity. The director contains a script to execute the upgrade on an individual Object Storage nodes.

Prerequisites

- You have previously run **openstack overcloud deploy** with the **major-upgrade-composable-steps-docker.yaml** environment file. This upgrades the main custom roles and their composable services.

Procedure

1. Obtain a list of Object Storage nodes:

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

2. Perform the following steps for each Object Storage node in the list:

- a. Run the **upgrade-non-controller.sh** script using the node name to identify the node to upgrade:

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



NOTE

If using pre-provisioned node infrastructure, see [Section 4.9, “Preparing for Pre-Provisioned Nodes Upgrade”](#) for changes with this command.

- b. Wait until the Object Storage node completes the upgrade.
- c. Reboot the Object Storage node:

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

- d. Wait until the Object Storage node completes the reboot.

Related Information

- If you encounter any issues after completing this step, please contact Red Hat and request guidance and assistance.

5.3. UPGRADING THE COMPUTE NODES

Compute nodes are not included in the main overcloud node upgrade process. To ensure maximum uptime of instances, you migrate each instance from a Compute node before upgrading the node. This means the Compute node upgrade process involves the following steps:

1. [Select a Compute node to upgrade](#)
2. [Migrate instances to another Compute node](#)
3. [Upgrade the empty Compute node](#)
4. [Reboot and enable the upgraded Compute node](#)

Prerequisites

- You have previously run 'openstack overcloud deploy' with the **major-upgrade-composable-steps-docker.yaml** environment file. This upgrades the main custom roles and their composable services.

Procedure

Select a Compute node to upgrade:

1. List all Compute nodes:

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

2. Select a Compute node to upgrade and note its UUID and name.

Migrate instances to another Compute node:

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

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

2. List all instances on the Compute node:

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

3. Use one of the following commands to migrate your instances:

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

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

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

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

- c. Live migrate all instances at once:

```
$ nova host-evacuate-live [hostname]
```



NOTE

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

4. Wait until migration completes.
5. Confirm the migration was successful:

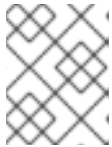
```
(overcloud) $ openstack server list --host [hostname] --all-projects
```

6. Continue migrating instances until none remain on the chosen Compute Node.

Upgrade the empty Compute node:

1. Run the **upgrade-non-controller.sh** script using the node name to identify the node to upgrade:

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



NOTE

If using pre-provisioned node infrastructure, see [Section 4.9, “Preparing for Pre-Provisioned Nodes Upgrade”](#) for changes with this command.

2. Wait until the Compute node completes the upgrade.

Reboot and enable the upgraded Compute node:

1. Log into the Compute Node and reboot it:

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

2. Wait until the node boots.
3. Enable the Compute Node again:

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

4. Check whether the Compute node is enabled:

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

Select the next node to upgrade. [Migrate its instances to another Compute node](#) before performing the upgrade. Repeat this process until you have upgraded all Compute nodes.

Related Information

- If you encounter any issues after completing this step, please contact Red Hat and request guidance and assistance.

5.4. 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-docker.yaml**), which you include using the **openstack overcloud deploy** command.

Prerequisites

- You have upgraded all nodes.

Procedure

1. Run the **openstack overcloud deploy** command and include:
 - All options and custom environment files relevant to your environment, such as network isolation and storage.
 - The **overcloud_images.yaml** environment file generated in [Section 4.2, “Preparing for Containerized Services”](#).
 - The **major-upgrade-converge-docker.yaml** environment file.
For example:

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

2. Wait until the overcloud updates with the new environment file's configuration.
3. Check if all services are active. For example, to check services on a Controllor node:

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

Related Information

- If you encounter any issues after completing this step, please contact Red Hat and request guidance and assistance.

CHAPTER 6. EXECUTING POST UPGRADE STEPS

This process implements final steps after completing the main upgrade process.

Prerequisites

- You have completed the overcloud upgrade to the latest major release.

6.1. INCLUDING THE UNDERCLOUD CA ON NEW OVERCLOUD NODES

In [Section 4.8, “Preparing Access to the Undercloud’s Public API over SSL/TLS”](#) we added the undercloud certificate authority (CA) on all existing overcloud nodes. New nodes added to the environment, either through scaling or replacement, also require the CA so that the new overcloud node has access to the OpenStack Object Storage (swift) Public API. This procedure shows how to include the undercloud CA on all new overcloud nodes.

Prerequisites

- You have upgraded to Red Hat OpenStack Platform 12.
- Your undercloud uses SSL/TLS for its Public API.

Procedure

1. Create a new or edit an existing environment file. This example uses the filename **undercloud-ca-map.yaml**.
2. Add the **CAMap** parameter to the **parameter_defaults** section of the environment file. Use the following syntax as an example:

```
parameter_defaults:
  CAMap:
    undercloud-ca: ❶
    content: | ❷
      -----BEGIN CERTIFICATE-----
      MIIDlTCCAn2gAwIBAgIJA0nPtx2hHEhrMA0GCSqGSIb3D
      BAYTA1VTMQswCQYDVQQIDAJQzEQMA4GA1UEBwwHUmFsZ
      UmVkiehhdDELMAkGA1UECwwCUUUxFDASBgNVBAMMCzE5M
      ...
      -----END CERTIFICATE-----
```

- ❶ This is the name that identifies the CA in each overcloud node’s trust database.
- ❷ The **content** section is the actual CA certificate. Copy and paste the CA content in this section. Ensure the CA’s indentation matches the requirements for YAML syntax.

3. Save this file.
4. Include this file with subsequent execution of the **openstack overcloud deploy** command.

Related Information

- See ["Enabling SSL/TLS on the Overcloud"](#) in the Advanced Overcloud Customization guide for more information about configuring certificate authority trusts on the overcloud.

6.2. GENERAL CONSIDERATIONS AFTER AN OVERCLOUD UPGRADE

The following items are general considerations after an overcloud upgrade:

- If necessary, review the resulting configuration files on the overcloud nodes. The upgraded packages might have installed **.rpmnew** files appropriate to the upgraded version of each service.
- 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
```


CHAPTER 7. KEEPING OPENSTACK PLATFORM UPDATED

This process provides instructions on how to keep your OpenStack Platform environment updated from minor versions to minor version. This is a minor update within **Red Hat OpenStack Platform 12**.

Prerequisites

- You have upgraded the overcloud to **Red Hat OpenStack Platform 12**.
- New packages and container images are available within **Red Hat OpenStack Platform 12**.

7.1. VALIDATING THE UNDERCLOUD BEFORE AN UPDATE

The following is a set of steps to check the functionality of your **Red Hat OpenStack Platform 12** undercloud before an update.

Procedure

1. Source the undercloud access details:

```
$ source ~/stackrc
```

2. Check for failed Systemd services:

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

3. Check the undercloud free space:

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

+Use the "[Undercloud Requirements](#)" as a basis to determine if you have adequate free space.

4. Check that clocks are synchronized on the undercloud:

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

5. Check the undercloud network services:

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

All agents should be **Alive** and their state should be **UP**.

6. Check the undercloud compute services:

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

All agents' status should be **enabled** and their state should be **up**

Related Information

- The following solution article shows how to remove deleted stack entries in your OpenStack Orchestration (heat) database: <https://access.redhat.com/solutions/2215131>

7.2. VALIDATING THE OVERCLOUD BEFORE AN UPDATE

The following is a set of steps to check the functionality of your **Red Hat OpenStack Platform 12** overcloud before an update.

Procedure

1. Source the undercloud access details:

```
$ source ~/stackrc
```

2. Check the status of your bare metal nodes:

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

All nodes should have a valid power state (**on**) and maintenance mode should be **false**.

3. Check for failed Systemd services:

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

4. Check for failed containerized services:

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

5. Check the HAProxy connection to all services. Obtain the Control Plane VIP address and authentication details for the **haproxy.stats** service:

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

Use these details in the following cURL request:

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

Replace **<PASSWORD>** and **<IP ADDRESS>** details with the respective details from the **haproxy.stats** service. The resulting list shows the OpenStack Platform services on each node and their connection status.

6. Check overcloud database replication health:

-

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

7. Check RabbitMQ cluster health:

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

8. Check Pacemaker resource health:

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

Look for:

- All cluster nodes **online**.
- No resources **stopped** on any cluster nodes.
- No **failed** pacemaker actions.

9. Check the disk space on each overcloud node:

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

10. Check overcloud Ceph Storage cluster health. The following command runs the **ceph** tool on a Controller node to check the cluster:

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

11. Check Ceph Storage OSD for free space. The following command runs the **ceph** tool on a Controller node to check the free space:

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

12. Check that clocks are synchronized on overcloud nodes

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

13. Source the overcloud access details:

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

14. Check the overcloud network services:

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

All agents should be **Alive** and their state should be **UP**.

15. Check the overcloud compute services:

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

All agents' status should be **enabled** and their state should be **up**

16. Check the overcloud volume services:

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

All agents' status should be **enabled** and their state should be **up**.

Related Information

- Review the article ["How can I verify my OpenStack environment is deployed with Red Hat recommended configurations?"](#). This article provides some information on how to check your Red Hat OpenStack Platform environment and tune the configuration to Red Hat's recommendations.

7.3. KEEPING THE UNDERCLOUD UPDATED

The director provides commands to update the packages on the undercloud node. This allows you to perform a minor update within the current version of your OpenStack Platform environment. This is a minor update within **Red Hat OpenStack Platform 12**.

Prerequisites

- You are using **Red Hat OpenStack Platform 12**.
- You have performed a backup of the undercloud.

Procedure

1. Log into the director as the **stack** user.
2. Update the **python-tripleoclient** package and its dependencies to ensure you have the latest scripts for the minor version update:

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

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

```
$ openstack undercloud upgrade
```

4. Reboot the node:

```
$ sudo reboot
```

5. Wait until the node boots.

6. 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.

7. 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 7.4, “Keeping the Overcloud Images Updated”](#).

7.4. KEEPING THE OVERCLOUD IMAGES UPDATED

The undercloud update process might download new image archives from the **rhosp-director-images** and **rhosp-director-images-ipa** packages. This process updates these images on your undercloud within **Red Hat OpenStack Platform 12**.

Prerequisites

- You are using **Red Hat OpenStack Platform 12**.
- You have updated to the latest minor release of your current undercloud.

Procedure

1. Check the **yum** log to determine if new image archives are available:

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

2. 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/*
```

3. Extract the archives:

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

4. 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 " ")
```

5. 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.

7.5. KEEPING THE OVERCLOUD UPDATED

The director provides commands to update the packages on all overcloud nodes. This allows you to perform a minor update within the current version of your OpenStack Platform environment. This is a minor update within **Red Hat OpenStack Platform 12**.

Prerequisites

- You are using **Red Hat OpenStack Platform 12**.
- You have updated to the latest minor release of your current undercloud.
- You have performed a backup of the overcloud.

Procedure

1. Find the latest tag for the containerized service images:

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

Make a note of the most recent tag.

2. Create an updated environment file for your container image source. Run using the **openstack overcloud container image prepare** command. For example, to use images from **registry.access.redhat.com**:

```
$ openstack overcloud container image prepare \
```

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

- ❶ Replace **[TAG]** with the tag obtained from the previous step.
- ❷ Include all additional environment files with the **-e** parameter. The director checks the custom resources in all included environment files and identifies the container images required for the containerized services.

For more information about generating this environment file for different source types, see ["Configuring Container Registry Details"](#) in the *Director Installation and Usage* guide.

3. Run the **openstack overcloud update stack** command to update the container image locations in your overcloud:

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

The **--init-minor-update** only performs an update of the parameters in the overcloud stack. It does not perform the actual package or container update. Wait until this command completes.

4. Perform a package and container update using the **openstack overcloud update** command. Using the **--nodes** option to upgrade node for each role. For example, the following command updates nodes in the **Controller** role

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

Run this command for each role group in the following order:

- **Controller**
 - **CephStorage**
 - **Compute**
 - **ObjectStorage**
 - Any custom roles such as **Database**, **MessageBus**, **Networker**, and so forth.
5. The update process starts for the chosen role starts. The director uses an Ansible playbook to perform the update and displays the output of each task.
 6. Update the next role group. Repeat until you have updated all nodes.
 7. The update process does not reboot any nodes in the Overcloud automatically. Updates to the kernel or Open vSwitch require a reboot. 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 Nodes"](#) procedures in the *Director Installation and Usage* guide.