



Red Hat OpenStack Platform 16.1

Framework for Upgrades (13 to 16.1)

In-place upgrades from Red Hat OpenStack Platform 13 to 16.1

Red Hat OpenStack Platform 16.1 Framework for Upgrades (13 to 16.1)

In-place upgrades from Red Hat OpenStack Platform 13 to 16.1

OpenStack Team
rhos-docs@redhat.com

Legal Notice

Copyright © 2020 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, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Abstract

This guide contains information on the framework for the in-place upgrades across long-life versions. This framework includes tools to upgrade your OpenStack Platform environment from one long life version to the next long life version. In this case, the guide focuses on upgrading from Red Hat OpenStack Platform 13 (Queens) to 16.1 (Train).

Table of Contents

CHAPTER 1. ABOUT THE RED HAT OPENSTACK PLATFORM FRAMEWORK FOR UPGRADES	5
1.1. UPGRADE FRAMEWORK FOR LONG LIFE VERSIONS	5
1.2. LIFECYCLE SUPPORT FOR LONG LIFE VERSIONS	5
1.3. UPGRADE PATHS FOR LONG LIFE RELEASE	5
CHAPTER 2. PLANNING AND PREPARATION FOR AN IN-PLACE UPGRADE	7
2.1. FAMILIARIZE YOURSELF WITH RED HAT OPENSTACK PLATFORM 16.1	7
2.2. HIGH LEVEL CHANGES IN RED HAT OPENSTACK PLATFORM 16.1	7
2.3. CHANGES IN RED HAT ENTERPRISE LINUX 8	8
2.4. LEAPP UPGRADE USAGE IN RED HAT OPENSTACK PLATFORM	9
2.5. SUPPORTED UPGRADE SCENARIOS	9
2.6. KNOWN ISSUES THAT MIGHT BLOCK AN UPGRADE	10
2.7. BACKUP AND RESTORE	11
2.8. MINOR VERSION UPDATE	11
2.9. PROXY CONFIGURATION	11
2.10. VALIDATING RED HAT OPENSTACK PLATFORM 13 BEFORE THE UPGRADE	12
CHAPTER 3. REPOSITORIES	14
3.1. UNDERCLOUD REPOSITORIES	14
3.2. OVERCLOUD REPOSITORIES	16
3.3. RED HAT SATELLITE 6 CONSIDERATIONS	19
PART I. UPGRADING THE UNDERCLOUD	21
CHAPTER 4. PREPARING FOR THE UNDERCLOUD UPGRADE	22
4.1. NEW MEMORY REQUIREMENTS	22
4.2. USING PREDICTABLE NIC NAMES FOR THE UNDERCLOUD NODE	22
4.3. SETTING THE SSH ROOT PERMISSION PARAMETER ON THE UNDERCLOUD	24
CHAPTER 5. UPGRADING THE UNDERCLOUD OPERATING SYSTEM	26
5.1. REMOVING RED HAT OPENSTACK PLATFORM DIRECTOR PACKAGES	26
5.2. PERFORMING A LEAPP UPGRADE ON THE UNDERCLOUD	26
CHAPTER 6. UPGRADING DIRECTOR	29
6.1. LOCKING THE ENVIRONMENT TO A RED HAT ENTERPRISE LINUX RELEASE	29
6.2. ENABLING REPOSITORIES FOR THE UNDERCLOUD	29
6.3. INSTALLING DIRECTOR PACKAGES	30
6.4. PREPARING CONTAINER IMAGES	30
6.5. CONTAINER IMAGE PREPARATION PARAMETERS	30
6.6. OBTAINING CONTAINER IMAGES FROM PRIVATE REGISTRIES	34
6.7. OBTAINING TRANSITIONAL CONTAINERS FOR UPGRADES	35
6.8. UPDATING THE UNDERCLOUD.CONF FILE	36
6.9. DIRECTOR CONFIGURATION PARAMETERS	36
6.10. RUNNING THE DIRECTOR UPGRADE	42
PART II. PREPARING FOR OVERCLOUD UPGRADE	44
CHAPTER 7. INITIAL STEPS FOR OVERCLOUD PREPARATION	45
7.1. PREPARING FOR OVERCLOUD SERVICE DOWNTIME	45
7.2. SELECTING COMPUTE NODES FOR UPGRADE TESTING	45
7.3. VALIDATING THE PRE-UPGRADE REQUIREMENTS	45
7.4. DISABLING FENCING IN THE OVERCLOUD	46
7.5. CREATING AN OVERCLOUD INVENTORY FILE	47

CHAPTER 8. CONFIGURING THE OVERCLOUD FOR A LEAPP UPGRADE	48
8.1. CREATING AN UPGRADES ENVIRONMENT FILE	48
8.2. UPGRADE PARAMETERS	48
8.3. COPYING THE LEAPP DATA TO THE OVERCLOUD NODES	49
8.4. USING PREDICTABLE NIC NAMES FOR OVERCLOUD NODES	50
8.5. SETTING THE SSH ROOT PERMISSION PARAMETER ON THE OVERCLOUD	50
CHAPTER 9. UPDATING COMPOSABLE SERVICES AND PARAMETERS	52
9.1. UPDATING COMPOSABLE SERVICES IN CUSTOM ROLES_DATA FILES	52
9.2. UPDATING COMPOSABLE SERVICES IN CUSTOM ENVIRONMENT FILES	55
9.3. CONFIGURING ACCESS TO THE UNDERCLOUD REGISTRY	56
9.4. DEPRECATED AND REMOVED FILTERS FOR THE NOVASCHEDULERDEFAULTFILTERS PARAMETER	57
9.5. SETTING THE COMPUTE NAME FORMAT	58
9.6. UPDATING YOUR SSL/TLS CONFIGURATION	58
9.7. UPDATING POST CONFIGURATION TEMPLATES	59
CHAPTER 10. UPDATING OVERCLOUD REGISTRATION TO THE RED HAT CUSTOMER PORTAL	61
10.1. RED HAT SUBSCRIPTION MANAGER (RHSM) COMPOSABLE SERVICE	61
10.2. RHSMVARS SUB-PARAMETERS	61
10.3. SWITCHING TO THE RHSM COMPOSABLE SERVICE	62
10.4. RHEL-REGISTRATION TO RHSM MAPPINGS	63
10.5. REGISTERING THE OVERCLOUD WITH THE RHSM COMPOSABLE SERVICE	64
10.6. APPLYING THE RHSM COMPOSABLE SERVICE TO DIFFERENT ROLES	65
CHAPTER 11. UPDATING OVERCLOUD REGISTRATION TO RED HAT SATELLITE	67
11.1. RED HAT SUBSCRIPTION MANAGER (RHSM) COMPOSABLE SERVICE	67
11.2. RHSMVARS SUB-PARAMETERS	67
11.3. SWITCHING TO THE RHSM COMPOSABLE SERVICE	68
11.4. RHEL-REGISTRATION TO RHSM MAPPINGS	69
11.5. REGISTERING THE OVERCLOUD TO RED HAT SATELLITE	70
11.6. PREPARING LEAPP TO USE SATELLITE SERVER	71
CHAPTER 12. PREPARING FOR A CEPH STORAGE UPGRADE	73
12.1. UNDERSTANDING THE CEPH STORAGE NODE UPGRADE PROCESS AT A HIGH LEVEL	73
12.2. CHECKING YOUR CEPH-ANSIBLE VERSION	74
12.3. SETTING THE CEPH-ANSIBLE REPOSITORY	75
12.4. CHECKING CEPH CLUSTER STATUS BEFORE AN UPGRADE	75
CHAPTER 13. UPDATING NETWORK CONFIGURATION	77
13.1. UPDATING NETWORK INTERFACE TEMPLATES	77
13.2. MAINTAINING OPEN VSWITCH COMPATIBILITY DURING THE UPGRADE	78
13.3. MAINTAINING COMPOSABLE NETWORK COMPATIBILITY DURING THE UPGRADE	79
CHAPTER 14. PREPARING NETWORK FUNCTIONS VIRTUALIZATION (NFV)	80
14.1. NETWORK FUNCTIONS VIRTUALIZATION (NFV) ENVIRONMENT FILES	80
CHAPTER 15. FINAL REVIEW BEFORE UPGRADE	81
15.1. NEW ENVIRONMENT FILES TO INCLUDE WITH YOUR DEPLOYMENT	81
15.2. ENVIRONMENT FILES TO REMOVE FROM YOUR DEPLOYMENT	81
15.3. UPGRADE CHECKLIST	82
PART III. UPGRADING THE OVERCLOUD	83
CHAPTER 16. UPGRADE COMMAND OVERVIEW	84
16.1. OPENSTACK OVERCLOUD UPGRADE PREPARE	84

16.2. OPENSTACK OVERCLOUD UPGRADE RUN	84
16.3. OPENSTACK OVERCLOUD EXTERNAL-UPGRADE RUN	84
16.4. OPENSTACK OVERCLOUD UPGRADE CONVERGE	85
16.5. OVERCLOUD NODE UPGRADE WORKFLOW	85
CHAPTER 17. UPGRADING A STANDARD OVERCLOUD	87
17.1. RUNNING THE OVERCLOUD UPGRADE PREPARATION	87
17.2. UPGRADING CONTROLLER NODES	88
17.3. UPGRADING THE OPERATING SYSTEM FOR CEPH STORAGE NODES	91
17.4. UPGRADING COMPUTE NODES	94
17.5. SYNCHRONIZING THE OVERCLOUD STACK	95
CHAPTER 18. UPGRADING A SPLIT CONTROLLER OVERCLOUD	97
18.1. RUNNING THE OVERCLOUD UPGRADE PREPARATION	97
18.2. UPGRADING PACEMAKER-BASED NODES	98
18.3. UPGRADING NON-PACEMAKER CONTROLLER NODES	100
18.4. UPGRADING THE OPERATING SYSTEM FOR CEPH MON NODES	101
18.5. UPGRADING THE OPERATING SYSTEM FOR CEPH STORAGE NODES	103
18.6. UPGRADING COMPUTE NODES	106
18.7. SYNCHRONIZING THE OVERCLOUD STACK	107
CHAPTER 19. UPGRADING AN OVERCLOUD WITH HYPER-CONVERGED INFRASTRUCTURE	109
19.1. RUNNING THE OVERCLOUD UPGRADE PREPARATION	109
19.2. UPGRADING CONTROLLER NODES	110
19.3. UPGRADING COMPUTE NODES WITH HYPER-CONVERGED INFRASTRUCTURE (HCI)	113
19.4. SYNCHRONIZING THE OVERCLOUD STACK	115
PART IV. CONCLUDING THE OVERCLOUD UPGRADE	117
CHAPTER 20. UPGRADING TO RED HAT CEPH STORAGE 4	118
20.1. INSTALLING CEPH-ANSIBLE	118
20.2. UPGRADING TO CEPH STORAGE 4	118
CHAPTER 21. PERFORMING POST-UPGRADE ACTIONS	119
21.1. REMOVING UNNECESSARY PACKAGES FROM THE UNDERCLOUD	119
21.2. VALIDATING THE POST-UPGRADE FUNCTIONALITY	119
21.3. UPGRADING THE OVERCLOUD IMAGES	119
21.4. MIGRATING TO OPEN VIRTUAL NETWORK (OVN)	120
PART V. TROUBLESHOOTING	122
CHAPTER 22. TROUBLESHOOTING UPGRADE ISSUES	123
22.1. CORRECTING ENVIRONMENT FILES	123

CHAPTER 1. ABOUT THE RED HAT OPENSTACK PLATFORM FRAMEWORK FOR UPGRADES

The Red Hat OpenStack Platform framework for upgrades is a workflow to upgrade your Red Hat OpenStack Platform environment from one long life version to the next long life version. This workflow is an in-place solution and the upgrade occurs within your existing environment.

1.1. UPGRADE FRAMEWORK FOR LONG LIFE VERSIONS

You can use the Red Hat OpenStack Platform **upgrade framework** to perform an in-place upgrade path through multiple versions of the overcloud. The goal is to provide you with an opportunity to remain on certain OpenStack versions that are considered **long life versions** and upgrade when the next long life version is available.

This guide provides an upgrade framework through the following versions:

Current Version	Target Version
Red Hat OpenStack Platform 13	Red Hat OpenStack Platform 16.1

1.2. LIFECYCLE SUPPORT FOR LONG LIFE VERSIONS

For detailed support dates and information on the lifecycle support for Red Hat OpenStack Platform, see ["Red Hat OpenStack Platform Life Cycle"](#).

1.3. UPGRADE PATHS FOR LONG LIFE RELEASE

Red Hat provides two options for upgrading your environment to the next long life release:

In-place upgrade

Perform an upgrade of the services in your existing environment. This guide primarily focuses on this option.

Parallel migration

Create a new Red Hat OpenStack Platform 16.1 environment and migrate your workloads from your current environment to the new environment. For more information about Red Hat OpenStack Platform parallel migration, contact Red Hat Global Professional Services.



IMPORTANT

The durations in this table are minimal estimates based on internal testing and might not apply to all productions environments. For example, if your hardware has low specifications or an extended boot period, allow for more time with these durations. To accurately gauge the upgrade duration for each task, perform these procedures in a test environment with hardware similar to your production environment.

Table 1.1. Impact and duration of upgrade paths

	In-place upgrade	Parallel migration
Upgrade duration for undercloud	<p>Estimated duration for each major action includes the following:</p> <ul style="list-style-type: none"> • 30 minutes for Leapp upgrade command • 30 minutes for Leapp reboot • 40 minutes for director upgrade 	None. You are creating a new undercloud in addition to your existing undercloud.
Upgrade duration for overcloud control plane	<p>Estimates for each Controller node:</p> <ul style="list-style-type: none"> • 60 minutes for Leapp upgrade and reboot • 60 minutes for service upgrade 	None. You are creating a new control plane in addition to your existing control plane.
Outage duration for control plane	The duration of the service upgrade of the bootstrap Controller node, which is approximately 60 minutes.	None. Both overclouds are operational during the workload migration.
Consequences of control plane outage	You cannot perform OpenStack operations during the outage.	No outage.
Upgrade duration for overcloud data plane	<p>Estimates for each Compute node and Ceph Storage node:</p> <ul style="list-style-type: none"> • 60 minutes for Leapp upgrade and reboot • 30 minutes for service upgrade 	None. You are creating a new data plane in addition to your existing data plane.
Outage duration for data plane	The outage is minimal due to workload migration from node to node.	The outage is minimal due to workload migration from overcloud to overcloud.
Additional hardware requirements	No additional hardware is required.	Additional hardware is required to create a new undercloud and overcloud.

CHAPTER 2. PLANNING AND PREPARATION FOR AN IN-PLACE UPGRADE

Before you conduct an in-place upgrade of your OpenStack Platform environment, create a plan for the upgrade and accommodate any potential obstacles that might block a successful upgrade.

2.1. FAMILIARIZE YOURSELF WITH RED HAT OPENSTACK PLATFORM 16.1

Before you perform an upgrade, familiarize yourself with Red Hat OpenStack Platform 16.1 to help you understand the resulting environment and any potential version-to-version changes that might affect your upgrade. To familiarize yourself with Red Hat OpenStack Platform 16.1, follow these suggestions:

- Read the release notes for all versions across the upgrade path and identify any potential aspects that require planning:
 - Components that contain new features
 - Known issues

Open the release notes for each version using these links:

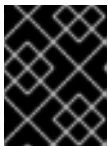
- [Red Hat OpenStack Platform 13](#), which is your current version
- [Red Hat OpenStack Platform 14](#)
- [Red Hat OpenStack Platform 15](#)
- [Red Hat OpenStack Platform 16.0](#)
- [Red Hat OpenStack Platform 16.1](#), which is your target version
- Read the [Director Installation and Usage](#) guide for version 16.1 and familiarize yourself with any new requirements and processes in this guide.
- Install a proof-of-concept Red Hat OpenStack Platform 16.1 undercloud and overcloud. Develop hands-on experience of the target OpenStack Platform version and investigate potential differences between the target version and your current version.

2.2. HIGH LEVEL CHANGES IN RED HAT OPENSTACK PLATFORM 16.1

The following high-level changes occur during the upgrade to Red Hat OpenStack Platform 16.1:

- OpenStack Platform director 16.1 configures the overcloud using an Ansible-driven method called **config-download**. This replaces the standard heat-based configuration method. Director still uses heat to orchestrate provisioning operations.
- The director installation uses the same method as the overcloud deployment. Therefore, the undercloud also uses **openstack-tripleo-heat-templates** as a blueprint for installing and configuring each service.
- The undercloud runs OpenStack services in containers.

- The undercloud pulls and stores container images through a new method. Instead of pulling container images before deploying the overcloud, the undercloud pulls all relevant container images during the deployment process.
- The overcloud deployment process includes an Advanced Subscription Management method to register nodes. This method incorporates an Ansible role to register OpenStack Platform nodes. The new method also applies different subscriptions to different node roles if necessary.
- The overcloud now uses Open Virtual Network (OVN) as the default ML2 mechanism driver. It is possible to migrate your Open vSwitch (OVS) service to OVN, which you perform after the completion of a successful upgrade.
- The undercloud and overcloud both run on Red Hat Enterprise Linux 8.
- **openstack-tripleo-heat-templates** includes a unified composable service template collection in the **deployment** directory. This directory now includes templates with merged content from both the containerized service and Puppet-based composable service templates.
- The OpenStack Data Processing service (sahara) is no longer supported.



IMPORTANT

If you have sahara enabled in your Red Hat OpenStack Platform 13 environment, do not continue with this upgrade and contact Red Hat Global Support Services.

- The OpenStack Telemetry components are deprecated in favor of the Service Telemetry Framework (STF).

2.3. CHANGES IN RED HAT ENTERPRISE LINUX 8

The undercloud and overcloud both run on Red Hat Enterprise Linux 8. This includes new tools and functions relevant to the undercloud and overcloud:

- The undercloud and overcloud use the Red Hat Container Toolkit. Instead of **docker** to build and control the container lifecycle, Red Hat Enterprise Linux 8 includes **buildah** to build new container images and **podman** for container management.
- Red Hat Enterprise Linux 8 does not include the **docker-distribution** package. The undercloud now includes a private HTTP registry to provide container images to overcloud nodes.
- The upgrade process from Red Hat Enterprise Linux 7 to 8 uses the **leapp** tool.
- Red Hat Enterprise Linux 8 does not use the **ntp** service. Instead, Red Hat Enterprise Linux 8 uses **chronyd**.
- Red Hat Enterprise Linux 8 includes new versions of high availability tools.

The Red Hat OpenStack Platform 16.1 uses Red Hat Enterprise Linux 8.2 as the base operating system. As a part of the upgrade process, you will upgrade the base operating system of nodes to Red Hat Enterprise Linux 8.2.

For more information about the key differences between Red Hat Enterprise Linux 7 and 8, see ["Considerations in adopting RHEL 8"](#). For general information about Red Hat Enterprise Linux 8, see [Product Documentation for Red Hat Enterprise Linux 8](#).

2.4. LEAPP UPGRADE USAGE IN RED HAT OPENSTACK PLATFORM

The long-life Red Hat OpenStack Platform upgrade also requires an upgrade from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8. Red Hat Enterprise Linux 7 includes a tool named **leapp**, which performs the upgrade to Red Hat Enterprise Linux 8. Both the undercloud and overcloud use a separate process for performing the operating system upgrade.

Undercloud process

Run the **leapp** upgrade manually before you run the **openstack undercloud upgrade** command. The undercloud upgrade includes instructions for performing the **leapp** upgrade.

Overcloud process

The overcloud upgrade framework automatically runs the **leapp** upgrade.

Limitations

For information of potential limitations that might affect your upgrade, see the following sections from the *Upgrading from RHEL 7 to RHEL 8* guide:

- ["Planning an upgrade"](#)
- ["Known issues"](#)

In particular, you cannot perform a Leapp upgrade on nodes that use encryption of the whole disk or a partition, such as LUKS encryption, or file-system encryption. This limitation affects Ceph OSD nodes that you have configured with the **dmccrypt: true** parameter.

If any known limitations affect your environment, seek advice from the [Red Hat Technical Support Team](#).

2.5. SUPPORTED UPGRADE SCENARIOS

Before proceeding with the upgrade, check that your overcloud is supported.



NOTE

If you are uncertain whether a particular scenario not mentioned in these lists is supported, seek advice from the [Red Hat Technical Support Team](#).

Supported scenarios

The following in-place upgrade scenarios are tested and supported.

- Standard environments with default role types: Controller, Compute, and Ceph Storage OSD
- Split-Controller composable roles
- Ceph Storage composable roles
- Hyper-Converged Infrastructure: Compute and Ceph Storage OSD services on the same node
- Environments with Network Functions Virtualization (NFV) technologies: Single-root input/output virtualization (SR-IOV) and Data Plane Development Kit (DPDK)

Technology preview scenarios

The framework for upgrades is considered a Technology Preview when you use it in conjunction with these features, and therefore is not fully supported by Red Hat. You should only test this scenario in a proof-of-concept environment and not upgrade in a production environment. For more information about Technology Preview features, see [Scope of Coverage Details](#).

- Edge and Distributed Compute Node (DCN) scenarios

Untested and unsupported

The following in-place upgrade scenarios are either untested or unsupported. If your overcloud configuration matches any of the scenarios on this list, postpone your upgrade and seek advice from the [Red Hat Technical Support Team](#).

- Environments with Instance HA enabled

2.6. KNOWN ISSUES THAT MIGHT BLOCK AN UPGRADE

Review the following known issues that might affect a successful upgrade.

[BZ#1852360 - swift_rsync and swift_rsync_healthcheck moved in failed state after undercloud upgrade](#)

The containerized **tripleo_swift_rsync** service on the Red Hat OpenStack Platform 16.1 undercloud reports a failed state. This is due to a conflict with the Red Hat OpenStack Platform 16.1 version running the OpenStack Object Storage (swift) **rsync** daemon through **xinetd**. This guide includes **xinetd** in the list of packages that you must remove before to the undercloud upgrade. Red Hat aims to resolve this issue in the next 16.1 minor release update.

[BZ#1866479 - container-tools module stream not enabled correctly](#)

The Red Hat Enterprise Linux 8 AppStream repository uses **container-tools:rhel8** as a default **dnf** module. The overcloud and undercloud require a change to the **container-tools:2.0** module to use the correct version of Podman for Red Hat OpenStack Platform 16.1 support. You change this module manually on the undercloud but director must change this module on the overcloud automatically during the upgrade process. This guide includes a workaround that uses the **UpgradelnitCommand** parameter to perform this module change on the overcloud.

[BZ#1871834 - ovn controllers cannot connect ovn dbs during 13-16.1 FFU upgrade](#)

Due to an VIP address change for Open Virtual Network (OVN), OVN controllers can no longer connect to the OVN database during an upgrade. To work around this issue:

1. After upgrading Controller nodes, obtain the new OVN VIP:

```
[root@controller-0 ~]# ovs-vsctl get open . external_ids:ovn-remote | sed -e 's/^//g'
```

This produces output in the following format.

```
tcp:<NewOvnVIP>:6642
```

2. Before you start the Compute upgrade process, log in to each Compute node and set the OVN VIP:

```
[root@compute-0 ~]# sudo docker exec -uroot ovn_controller ovs-vsctl set Open_Vswitch . external_ids:ovn-remote="tcp:<NewOvnVIP>:6642"
```

Red Hat aims to resolve this issue in the next 16.1 minor release update.

Red Hat Ceph Storage Issues

BZ#1855813 - Ceph tools repository should be switched from RHCS3 to RHCS4 only after converge, before running external-upgrade

The **ceph-ansible** playbook collection on the undercloud deploys Red Hat Ceph Storage containers on the overcloud. To upgrade your environment, you must have Red Hat Ceph Storage 3 version of **ceph-ansible** to maintain Ceph Storage 3 containers through the upgrade. This guide includes instructions on how to retain **ceph-ansible** version 3 over the course of the upgrade until you are ready to upgrade to Ceph Storage 4. Before performing the 13 to 16.1 upgrade, you must perform a minor version update of your Red Hat OpenStack Platform 13 environment and ensure you have **ceph-ansible** version 3.2.46 or later.

BZ#1870617 - noout ceph flag not unset during the FFU workflow

This guide includes commands to set **noout** and **norebalance** flags before the Ceph Storage OSD cluster upgrade and unset them after you complete the Ceph Storage OSD cluster upgrade. Red Hat aims to automate this procedure in the next 16.1 minor release update.

Red Hat Enterprise Linux Issues

BZ#1861977 - RHSA-2020:3216 grub2 security update renders system unbootable

Applying the RHSA-2020:3216 security update stops the grub menu from loading. This issue occurs when using the **shim-x64-15-14.el8_2** package. The **shim-x64-15-15.el8_2** package and later versions of this package corrects this issue. If the grub menu does not appear when rebooting nodes, see the "[System hangs after POST and the grub menu never loads after applying the RHSA-2020:3216 or RHSA-2020:3217](#)" solution.

2.7. BACKUP AND RESTORE

Before you upgrade your Red Hat OpenStack Platform 13 environment, back up the undercloud and overcloud control plane. For more information about backing up nodes with the Relax-and-recover (ReaR) utility, see the *Undercloud and Control Plane Back Up and Restore* guide.

- Back up your nodes before you perform an upgrade. For more information on backing up nodes prior to upgrade, see "[Red Hat OpenStack Platform 13 Undercloud and Control Plane Back Up and Restore](#)".
- You can back up each up node after it has been upgraded. For more information on backing up upgraded nodes, see "[Red Hat OpenStack Platform 16.1 Undercloud and Control Plane Back Up and Restore](#)".

2.8. MINOR VERSION UPDATE

Before you upgrade your Red Hat OpenStack Platform environment, update the environment to the latest minor version of your current release. For example, perform an update of your Red Hat OpenStack Platform 13 environment to the latest 13 before running the upgrade to Red Hat OpenStack Platform 16.1

For instructions on performing a minor version update for Red Hat OpenStack Platform 13, see "[Keeping Red Hat OpenStack Platform Updated](#)".

2.9. PROXY CONFIGURATION

If you use a proxy with your Red Hat OpenStack Platform 13 environment, the proxy configuration in the **/etc/environment** file will persist past the operating system upgrade and the Red Hat OpenStack Platform 16.1 upgrade.

- For more information about proxy configuration for Red Hat OpenStack Platform 13, see [Configuring an undercloud proxy](#).
- For more information about proxy configuration for Red Hat OpenStack Platform 16.1, see [Configuring an undercloud proxy](#).

2.10. VALIDATING RED HAT OPENSTACK PLATFORM 13 BEFORE THE UPGRADE

Before you upgrade to Red Hat OpenStack Platform 16.1, validate your undercloud and overcloud with the **tripleo-validations** playbooks. In Red Hat OpenStack Platform 13, you run these playbooks through the OpenStack Workflow Service (mistral).

Procedure

1. Log in to the undercloud as the **stack** user.

2. Source the **stackrc** file:

```
$ source ~/stackrc
```

3. Create a bash script called **pre-upgrade-validations.sh** and include the following content in the script:

```
#!/bin/bash
for VALIDATION in $(openstack action execution run tripleo.validations.list_validations
'{"groups": ["pre-upgrade"]}' | jq ".result[] | .id")
do
    echo "=== Running validation: $VALIDATION ==="
    STACK_NAME=$(openstack stack list -f value -c 'Stack Name')
    ID=$(openstack workflow execution create -f value -c ID tripleo.validations.v1.run_validation
{"validation_name": $VALIDATION, "plan": "$STACK_NAME"})
    while [ $(openstack workflow execution show $ID -f value -c State) == "RUNNING" ]
    do
        sleep 1
    done
    echo ""
    openstack workflow execution output show $ID | jq -r ".stdout"
    echo ""
done
```

4. Add permission to run the script:

```
$ chmod +x pre-upgrade-validations.sh
```

5. Run the script:

```
$ ./pre-upgrade-validations.sh
```

Review the script output to determine which validations succeed and fail:

=== Running validation: "check-ftype" ===

Success! The validation passed for all hosts:

* undercloud

CHAPTER 3. REPOSITORIES

This section contains the repositories for the undercloud and overcloud. Refer to this section when you need to enable repositories in certain situations:

- Enabling repositories when registering to the Red Hat Customer Portal.
- Enabling and synchronizing repositories to your Red Hat Satellite server.
- Enabling repositories when registering to your Red Hat Satellite server.

3.1. UNDERCLOUD REPOSITORIES

Red Hat OpenStack Platform 16.1 runs on Red Hat Enterprise Linux 8.2. As a result, you must lock the content from these repositories to the respective Red Hat Enterprise Linux version.



NOTE

If you synchronize repositories with Red Hat Satellite, you can enable specific versions of the Red Hat Enterprise Linux repositories. However, the repository remains the same despite the version you choose. For example, you can enable the 8.2 version of the BaseOS repository, but the repository name is still **rhel-8-for-x86_64-baseos-eus-rpms** despite the specific version you choose.



WARNING

Any repositories outside the ones specified here are not supported. Unless recommended, do not enable any other products or repositories outside the ones listed in the following tables or else you might encounter package dependency issues. Do not enable Extra Packages for Enterprise Linux (EPEL).

Core repositories

The following table lists core repositories for installing the undercloud.

Name	Repository	Description of requirement
Red Hat Enterprise Linux 8 for x86_64 - BaseOS (RPMs) Extended Update Support (EUS)	rhel-8-for-x86_64-baseos-eus-rpms	Base operating system repository for x86_64 systems.
Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)	rhel-8-for-x86_64-appstream-eus-rpms	Contains Red Hat OpenStack Platform dependencies.
Red Hat Enterprise Linux 8 for x86_64 - High Availability (RPMs) Extended Update Support (EUS)	rhel-8-for-x86_64-highavailability-eus-rpms	High availability tools for Red Hat Enterprise Linux. Used for Controller node high availability.

Name	Repository	Description of requirement
Red Hat Ansible Engine 2.9 for RHEL 8 x86_64 (RPMs)	ansible-2.9-for-rhel-8-x86_64-rpms	Ansible Engine for Red Hat Enterprise Linux. Used to provide the latest version of Ansible.
Advanced Virtualization for RHEL 8 x86_64 (RPMs)	advanced-virt-for-rhel-8-x86_64-rpms	Provides virtualization packages for OpenStack Platform.
Red Hat Satellite Tools for RHEL 8 Server RPMs x86_64	satellite-tools-6.5-for-rhel-8-x86_64-rpms	Tools for managing hosts with Red Hat Satellite 6.
Red Hat OpenStack Platform 16.1 for RHEL 8 (RPMs)	openstack-16.1-for-rhel-8-x86_64-rpms	Core Red Hat OpenStack Platform repository, which contains packages for Red Hat OpenStack Platform director.
Red Hat Fast Datapath for RHEL 8 (RPMs)	fast-datapath-for-rhel-8-x86_64-rpms	Provides Open vSwitch (OVS) packages for OpenStack Platform.

Ceph repositories

The following table lists Ceph Storage related repositories for the undercloud.

Name	Repository	Description of Requirement
Red Hat Ceph Storage Tools 4 for RHEL 8 x86_64 (RPMs)	rhceph-4-tools-for-rhel-8-x86_64-rpms	Provides tools for nodes to communicate with the Ceph Storage cluster. The undercloud requires the ceph-ansible package from this repository if you plan to use Ceph Storage in your overcloud.

IBM POWER repositories

The following table contains a list of repositories for Red Hat Openstack Platform on POWER PC architecture. Use these repositories in place of equivalents in the Core repositories.

Name	Repository	Description of requirement
Red Hat Enterprise Linux for IBM Power, little endian – BaseOS (RPMs)	rhel-8-for-ppc64le-baseos-rpms	Base operating system repository for ppc64le systems.
Red Hat Enterprise Linux 8 for IBM Power, little endian – AppStream (RPMs)	rhel-8-for-ppc64le-appstream-rpms	Contains Red Hat OpenStack Platform dependencies.

Name	Repository	Description of requirement
Red Hat Enterprise Linux 8 for IBM Power, little endian – High Availability (RPMs)	rhel-8-for-ppc64le-highavailability-rpms	High availability tools for Red Hat Enterprise Linux. Used for Controller node high availability.
Red Hat Ansible Engine 2.8 for RHEL 8 IBM Power, little endian (RPMs)	ansible-2.8-for-rhel-8-ppc64le-rpms	Ansible Engine for Red Hat Enterprise Linux. Provides the latest version of Ansible.
Red Hat OpenStack Platform 16.1 for RHEL 8 (RPMs)	openstack-16.1-for-rhel-8-ppc64le-rpms	Core Red Hat OpenStack Platform repository for ppc64le systems.

3.2. OVERCLOUD REPOSITORIES

Red Hat OpenStack Platform 16.1 runs on Red Hat Enterprise Linux 8.2. As a result, you must lock the content from these repositories to the respective Red Hat Enterprise Linux version.



NOTE

If you synchronize repositories with Red Hat Satellite, you can enable specific versions of the Red Hat Enterprise Linux repositories. However, the repository remains the same despite the version you choose. For example, you can enable the 8.2 version of the BaseOS repository, but the repository name is still **rhel-8-for-x86_64-baseos-eus-rpms** despite the specific version you choose.



WARNING

Any repositories outside the ones specified here are not supported. Unless recommended, do not enable any other products or repositories outside the ones listed in the following tables or else you might encounter package dependency issues. Do not enable Extra Packages for Enterprise Linux (EPEL).

Core repositories

The following table lists core repositories for installing the overcloud.

Name	Repository	Description of requirement
Red Hat Enterprise Linux 8 for x86_64 - BaseOS (RPMs) Extended Update Support (EUS)	rhel-8-for-x86_64-baseos-eus-rpms	Base operating system repository for x86_64 systems.
Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)	rhel-8-for-x86_64-appstream-eus-rpms	Contains Red Hat OpenStack Platform dependencies.
Red Hat Enterprise Linux 8 for x86_64 - High Availability (RPMs) Extended Update Support (EUS)	rhel-8-for-x86_64-highavailability-eus-rpms	High availability tools for Red Hat Enterprise Linux.
Red Hat Ansible Engine 2.9 for RHEL 8 x86_64 (RPMs)	ansible-2.9-for-rhel-8-x86_64-rpms	Ansible Engine for Red Hat Enterprise Linux. Used to provide the latest version of Ansible.
Advanced Virtualization for RHEL 8 x86_64 (RPMs)	advanced-virt-for-rhel-8-x86_64-rpms	Provides virtualization packages for OpenStack Platform.
Red Hat Satellite Tools for RHEL 8 Server RPMs x86_64	satellite-tools-6.5-for-rhel-8-x86_64-rpms	Tools for managing hosts with Red Hat Satellite 6.
Red Hat OpenStack Platform 16.1 for RHEL 8 (RPMs)	openstack-16.1-for-rhel-8-x86_64-rpms	Core Red Hat OpenStack Platform repository.
Red Hat Fast Datapath for RHEL 8 (RPMs)	fast-datapath-for-rhel-8-x86_64-rpms	Provides Open vSwitch (OVS) packages for OpenStack Platform.

Ceph repositories

The following table lists Ceph Storage related repositories for the overcloud.

Name	Repository	Description of Requirement
Red Hat Enterprise Linux 8 for x86_64 - BaseOS (RPMs)	rhel-8-for-x86_64-baseos-rpms	Base operating system repository for x86_64 systems.
Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)	rhel-8-for-x86_64-appstream-rpms	Contains Red Hat OpenStack Platform dependencies.
Red Hat Enterprise Linux 8 for x86_64 - High Availability (RPMs)	rhel-8-for-x86_64-highavailability-rpms	High availability tools for Red Hat Enterprise Linux.
Red Hat Ansible Engine 2.9 for RHEL 8 x86_64 (RPMs)	ansible-2.9-for-rhel-8-x86_64-rpms	Ansible Engine for Red Hat Enterprise Linux. Used to provide the latest version of Ansible.

Name	Repository	Description of Requirement
Red Hat OpenStack Platform 16.1 Director Deployment Tools for RHEL 8 x86_64 (RPMs)	openstack-16.1-deployment-tools-for-rhel-8-x86_64-rpms	Packages to help director configure Ceph Storage nodes.
Red Hat Ceph Storage OSD 4 for RHEL 8 x86_64 (RPMs)	rhceph-4-osd-for-rhel-8-x86_64-rpms	(For Ceph Storage Nodes) Repository for Ceph Storage Object Storage daemon. Installed on Ceph Storage nodes.
Red Hat Ceph Storage MON 4 for RHEL 8 x86_64 (RPMs)	rhceph-4-mon-for-rhel-8-x86_64-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 4 for RHEL 8 x86_64 (RPMs)	rhceph-4-tools-for-rhel-8-x86_64-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.

Real Time repositories

The following table lists repositories for Real Time Compute (RTC) functionality.

Name	Repository	Description of requirement
Red Hat Enterprise Linux 8 for x86_64 - Real Time (RPMs)	rhel-8-for-x86_64-rt-rpms	Repository for Real Time KVM (RT-KVM). Contains packages to enable the real time kernel. Enable this repository for all Compute nodes targeted for RT-KVM. NOTE: You need a separate subscription to a Red Hat OpenStack Platform for Real Time SKU to access this repository.

Name	Repository	Description of requirement
Red Hat Enterprise Linux 8 for x86_64 – Real Time for NFV (RPMs)	rhel-8-for-x86_64-nfv-rpms	Repository for Real Time KVM (RT-KVM) for NFV. Contains packages to enable the real time kernel. Enable this repository for all NFV Compute nodes targeted for RT-KVM. NOTE: You need a separate subscription to a Red Hat OpenStack Platform for Real Time SKU to access this repository.

IBM POWER repositories

The following table lists repositories for Openstack Platform on POWER PC architecture. Use these repositories in place of equivalents in the Core repositories.

Name	Repository	Description of requirement
Red Hat Enterprise Linux for IBM Power, little endian – BaseOS (RPMs)	rhel-8-for-ppc64le-baseos-rpms	Base operating system repository for ppc64le systems.
Red Hat Enterprise Linux 8 for IBM Power, little endian – AppStream (RPMs)	rhel-8-for-ppc64le-appstream-rpms	Contains Red Hat OpenStack Platform dependencies.
Red Hat Enterprise Linux 8 for IBM Power, little endian – High Availability (RPMs)	rhel-8-for-ppc64le-highavailability-rpms	High availability tools for Red Hat Enterprise Linux. Used for Controller node high availability.
Red Hat Ansible Engine 2.8 for RHEL 8 IBM Power, little endian (RPMs)	ansible-2.8-for-rhel-8-ppc64le-rpms	Ansible Engine for Red Hat Enterprise Linux. Used to provide the latest version of Ansible.
Red Hat OpenStack Platform 16.1 for RHEL 8 (RPMs)	openstack-16.1-for-rhel-8-ppc64le-rpms	Core Red Hat OpenStack Platform repository for ppc64le systems.

3.3. RED HAT SATELLITE 6 CONSIDERATIONS

If you use Red Hat Satellite 6 to host RPMs and container images for your Red Hat OpenStack Platform environment, you must account for certain considerations when you use Satellite 6 to deliver content during the Red Hat OpenStack Platform 16.1 upgrade.

Assumptions about your current environment

- Your Satellite server already hosts Red Hat OpenStack Platform 13 RPMs and container images.
- You have already registered all nodes in your Red Hat OpenStack Platform 13 environment to the Satellite server. For example, you previously used an activation key linked to an Red Hat OpenStack Platform 13 content view to register nodes to OpenStack Platform 13 content.

Recommendations for Red Hat OpenStack Platform upgrades

- Enable and synchronize the necessary RPM repositories for both the Red Hat OpenStack Platform 13 undercloud and overcloud. This includes the necessary Red Hat Enterprise Linux 8.2 repositories.
- Create custom products on the Satellite server to host container images for the following Red Hat OpenStack Platform versions:
 - Red Hat OpenStack Platform 16.1
 - Red Hat OpenStack Platform 15
- Create and promote a content view for Red Hat OpenStack Platform 16.1 upgrade and include the following content in the content view:
 - All undercloud and overcloud RPM repositories, including Red Hat Enterprise Linux 8.2 repositories.
 - Red Hat OpenStack Platform 16.1 container images.
 - Red Hat OpenStack Platform 15 container images.
- Create an activation key for the upgrade to Red Hat OpenStack Platform 16.1 and associate the activation key with the Red Hat OpenStack Platform 16.1 content view. Pin this activation key to Red Hat Enterprise Linux 8.2.
- Check that no node has the **katello-host-tools-fact-plugin** package installed. The Leapp upgrade does not upgrade this package and leaving this package on a Red Hat Enterprise Linux 8.2 system causes **subscription-manager** to report errors.

PART I. UPGRADING THE UNDERCLOUD

CHAPTER 4. PREPARING FOR THE UNDERCLOUD UPGRADE

Before you perform the undercloud upgrade, you must complete some preparation steps so that the undercloud upgrade runs successfully.

4.1. NEW MEMORY REQUIREMENTS

In Red Hat OpenStack Platform 16.1, the undercloud has new memory requirements:

Red Hat OpenStack Platform 13	Red Hat OpenStack Platform 16.1
16 GB RAM	24 GB RAM

Ensure that your undercloud meets these new requirements before you proceed with the upgrade.

4.2. USING PREDICTABLE NIC NAMES FOR THE UNDERCLOUD NODE

Before you run the Leapp upgrade on the undercloud node, you must check for kernel-based NIC names, which usually contain a **eth** prefix. These NIC names are usually unpredictable in terms of NIC assignments. The Leapp upgrade fails if any NICs on the node use the **eth** prefix.

Procedure

1. Log in to the undercloud as the **stack** user.
2. Create an Ansible playbook named **playbook-nics.yaml** and copy the following content into the playbook:

```
---
- name: Rename eth devices
  hosts: all
  become: yes
  vars:
    prefix: "em"
  tasks:
    - name: Update udev rules
      lineinfile:
        line: "SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="
{{ ansible_facts[item]['perm_macaddress'] | default(ansible_facts[item]['macaddress']) }}",
NAME="{{ item|replace('eth',prefix) }}"
        path: /etc/udev/rules.d/70-rhosp-persistent-net.rules
        create: True
        with_items: "{{ ansible_interfaces }}"
        when: item.startswith("eth")
    - name: Rename eth files
      block:
        - name: Check that eth files exists
          stat:
            path: /etc/sysconfig/network-scripts/ifcfg-{{ item }}
          register: nic_result
          with_items: "{{ ansible_interfaces }}"
          when: item.startswith("eth")
```

```

- name: Copy nic files using the new prefix
  copy:
    remote_src: True
    src: "{{ item.stat.path }}"
    dest: "{{ item.stat.path|replace('eth',prefix) }}"
    with_items: "{{ nic_result.results }}"
    when: item.item.startswith("eth") and item.stat.exists
- name: Edit NAME in new network-script files
  lineinfile:
    regexp: "^NAME=.*"
    line: "NAME={{ item.item|replace('eth',prefix) }}"
    path: "{{ item.stat.path|replace('eth',prefix) }}"
    with_items: "{{ nic_result.results }}"
    when: item.item.startswith("eth") and item.stat.exists
- name: Edit DEVICE in new network-script files
  lineinfile:
    regexp: "^DEVICE=.*"
    line: "DEVICE={{ item.item|replace('eth',prefix) }}"
    path: "{{ item.stat.path|replace('eth',prefix) }}"
    with_items: "{{ nic_result.results }}"
    when: item.item.startswith("eth") and item.stat.exists
- name: Backup old eth network-script files
  copy:
    remote_src: True
    src: "{{ item.stat.path }}"
    dest: "{{ item.stat.path }}.bak"
    with_items: "{{ nic_result.results }}"
    when: item.item.startswith("eth") and item.stat.exists
- name: Remove old eth network-script files
  file:
    path: "{{ item.stat.path }}"
    state: absent
    with_items: "{{ nic_result.results }}"
    when: item.item.startswith("eth") and item.stat.exists
- name: Rename route files
  block:
    - name: Check that route files exists
      stat:
        path: /etc/sysconfig/network-scripts/route-{{ item }}
        register: route_result
        with_items: "{{ ansible_interfaces }}"
        when: item.startswith("eth")
    - name: Copy route files using the new prefix
      copy:
        remote_src: True
        src: "{{ item.stat.path }}"
        dest: "{{ item.stat.path|replace('eth',prefix) }}"
        with_items: "{{ route_result.results }}"
        when: item.item.startswith("eth") and item.stat.exists
    - name: Update prefix in route files that use IP command arguments format
      replace:
        regexp: "eth"
        replace: "{{ prefix }}"
        path: "{{ item.stat.path|replace('eth',prefix) }}"
        with_items: "{{ route_result.results }}"
        when: item.item.startswith("eth") and item.stat.exists

```

```

- name: Backup old route files
  copy:
    remote_src: True
    src: "{{ item.stat.path }}"
    dest: "{{ item.stat.path }}.bak"
  with_items: "{{ route_result.results }}"
  when: item.item.startswith("eth") and item.stat.exists
- name: Remove old route files
  file:
    path: "{{ item.stat.path }}"
    state: absent
  with_items: "{{ route_result.results }}"
  when: item.item.startswith("eth") and item.stat.exists

```

**NOTE**

You will use this playbook to rename the overcloud NICs at a later stage in the upgrade process.

3. Run the **playbook-nics.yaml** playbook on the undercloud:

```
$ ansible-playbook -c local -i localhost, playbook-nics.yaml
```

The playbook sets the new NIC prefix to **em**. To set a different NIC prefix, set the **prefix** variable when running the playbook:

```
$ ansible-playbook -c local -i localhost, -e prefix="mynic" ~/playbook-nics.yaml
```

4. Reboot the undercloud node using the standard reboot procedures. For more information, see ["Rebooting nodes"](#).

Resources

- ["How to perform an in-place upgrade to RHEL 8 when using kernel NIC names on RHEL 7"](#)
- ["Consistent Network Device Naming"](#)
- ["Understanding the Predictable Network Interface Device Names"](#)

4.3. SETTING THE SSH ROOT PERMISSION PARAMETER ON THE UNDERCLOUD

The Leapp upgrade checks whether the **PermitRootLogin** parameter exists in the **/etc/ssh/sshd_config** file. You must explicitly set this parameter to either **yes** or **no**.

For security purposes, set this parameter to **no** to disable SSH access to the root user on the undercloud.

Procedure

1. Log in to the undercloud as the **stack** user.
2. Check the **/etc/ssh/sshd_config** file for the **PermitRootLogin** parameter:

```
$ sudo grep PermitRootLogin /etc/ssh/sshd_config
```

3. If the parameter is not in the **/etc/ssh/sshd_config** file, edit the file and set the **PermitRootLogin** parameter:

```
PermitRootLogin no
```

4. Save the file.

CHAPTER 5. UPGRADING THE UNDERCLOUD OPERATING SYSTEM

Before you upgrade director, you must upgrade the undercloud operating system from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8. As a part of this operating system upgrade, you must remove the Red Hat OpenStack Platform 13 packages and then run the Leapp utility to upgrade the system packages. This package removal and operating system upgrade does not affect the undercloud database. After you complete the operating system upgrade, reinstall the Red Hat OpenStack Platform 16.1 versions of the director packages.

5.1. REMOVING RED HAT OPENSTACK PLATFORM DIRECTOR PACKAGES

Before you run the Leapp utility, remove the Red Hat OpenStack Platform 13 packages tied to Red Hat Enterprise Linux 7. These package names use a release suffix of **el7ost**. Some **el7ost** remain on the system as dependencies for **subscription-manager** and the Leapp utility.

Procedure

1. Log in to the undercloud as the **stack** user.
2. Disable the main OpenStack services on the undercloud:

```
$ sudo systemctl stop openstack-* httpd haproxy mariadb rabbitmq* docker xinetd
```

3. Remove the main OpenStack services from the undercloud, except OpenvSwitch and certain Python 2 packages that are required for the upgrade:

```
$ sudo yum -y remove *el7ost* galera* haproxy* \
  httpd mysql* pacemaker* xinetd python-jsonpointer \
  qemu-kvm-common-rhev qemu-img-rhev rabbit* \
  redis* \
  -- \
  -*openvswitch* -python-docker -python-PyMySQL \
  -python-pysocks -python2-asn1crypto -python2-babel \
  -python2-cffi -python2-cryptography -python2-dateutil \
  -python2-idna -python2-ipaddress -python2-jinja2 \
  -python2-jsonpatch -python2-markupsafe -python2-pyOpenSSL \
  -python2-requests -python2-six -python2-urllib3 \
  -python-httpplib2 -python-passlib -python2-netaddr -ceph-ansible
```

4. Remove the content from the **/etc/httpd** and **/var/lib/docker** directories:

```
$ sudo rm -rf /etc/httpd /var/lib/docker
```

5.2. PERFORMING A LEAPP UPGRADE ON THE UNDERCLOUD

Install and run the Leapp utility to upgrade the operating system to Red Hat Enterprise Linux 8.

Procedure

1. Log in to the undercloud as the **stack** user.

2. Install the Leapp utility:

```
$ sudo yum install leapp
```

3. Download the additional required data files (RPM package changes and RPM repository mapping) attached to the Knowledge Base article ["Data required by the Leapp utility for an in-place upgrade from RHEL 7 to RHEL 8"](#) and place these files in the `/etc/leapp/files/` directory.

4. Update your Red Hat subscription:

- If your undercloud uses the Red Hat Customer Portal for registration, refresh your current subscription to obtain access to the Red Hat Enterprise Linux 8.2 content:

```
$ sudo subscription-manager refresh
```

- If your undercloud uses a Red Hat Satellite server for registration, re-register the undercloud to a content view associated with your Red Hat OpenStack Platform 16.1 activation key.

```
$ sudo subscription-manager register --force --org ORG --activationkey  
ACTIVATION_KEY --release 8.2
```



NOTE

The content view that you create for Red Hat OpenStack Platform 16.1 should contain content for Red Hat Enterprise Linux 8.2.

5. Red Hat OpenStack Platform 16.1 uses a newer version of Open vSwitch`. Substitute the Open vSwitch version through the **to_remove** and **to_install** transaction files:

```
$ echo 'openvswitch2.11' | sudo tee -a /etc/leapp/transaction/to_remove  
$ echo 'openvswitch2.13' | sudo tee -a /etc/leapp/transaction/to_install
```

6. Retain the Red Hat Ceph Storage 3 version of **ceph-ansible** through the upgrade with the **to_keep** transaction file:

```
$ echo 'ceph-ansible' | sudo tee -a /etc/leapp/transaction/to_keep
```

7. Start the Leapp upgrade process:

```
$ sudo leapp upgrade --debug --enablerepo rhel-8-for-x86_64-baseos-eus-rpms --  
enablerepo rhel-8-for-x86_64-appstream-eus-rpms --enablerepo fast-datapath-for-rhel-8-  
x86_64-rpms --enablerepo ansible-2.9-for-rhel-8-x86_64-rpms
```

Use the **--enablerepo** option to set the repositories that you want to enable during the Leapp upgrade process. You must include these repositories to facilitate the Red Hat OpenStack Platform 16.1 transition, especially with the newer version of Open vSwitch.

8. Wait for the **leapp upgrade** command to successfully complete.

9. Create an empty **.autorelabel** file in your root directory:

```
$ sudo touch /.autorelabel
```

After a reboot, SELinux detects this file and automatically relabels the file system.

10. Reboot the undercloud:

```
$ sudo reboot
```

Resources

- ["Data required by the Leapp utility for an in-place upgrade from RHEL 7 to RHEL 8"](#)
- ["How to perform an in-place upgrade to RHEL 8 when using kernel NIC names on RHEL 7"](#)
- ["Upgrading to RHEL 8"](#)

CHAPTER 6. UPGRADING DIRECTOR

After you complete the undercloud operating system upgrade, upgrade director. The database from the previous Red Hat OpenStack Platform 13 undercloud remains on host after the operating system upgrade. Install the new Red Hat OpenStack Platform 16.1 packages and configure the new source for Red Hat OpenStack Platform 16.1 container images before you run the **openstack undercloud upgrade** command.

6.1. LOCKING THE ENVIRONMENT TO A RED HAT ENTERPRISE LINUX RELEASE

Red Hat OpenStack Platform 16.1 is supported on Red Hat Enterprise Linux 8.2. Before you perform the update, you must lock the undercloud repositories to the Red Hat Enterprise Linux 8.2 release to avoid upgrading the operating system to a newer minor release.

Procedure

1. Log in to the undercloud as the **stack** user.
2. Lock the undercloud to a specific version with the **subscription-manager release** command:

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

6.2. ENABLING REPOSITORIES FOR THE UNDERCLOUD

Enable the repositories that are required for the undercloud, and update the system packages to the latest versions.

Procedure

1. Log in to your undercloud as the **stack** user.
2. Disable all default repositories, and enable the required Red Hat Enterprise Linux repositories:

```
[stack@director ~]$ sudo subscription-manager repos --disable=*
[stack@director ~]$ sudo subscription-manager repos --enable=rhel-8-for-x86_64-baseos-
eus-rpms --enable=rhel-8-for-x86_64-appstream-eus-rpms --enable=rhel-8-for-x86_64-
highavailability-eus-rpms --enable=ansible-2.9-for-rhel-8-x86_64-rpms --enable=openstack-
16.1-for-rhel-8-x86_64-rpms --enable=fast-datapath-for-rhel-8-x86_64-rpms --
enable=advanced-virt-for-rhel-8-x86_64-rpms
```

These repositories contain packages that the director installation requires.

3. Set the **container-tools** repository module to version **2.0**:

```
[stack@director ~]$ sudo dnf module disable -y container-tools:rhel8
[stack@director ~]$ sudo dnf module enable -y container-tools:2.0
```

4. Set the **virt** repository module to version **8.2**:

```
[stack@director ~]$ sudo dnf module disable -y virt:rhel
[stack@director ~]$ sudo dnf module enable -y virt:8.2
```

5. Synchronize the operating system to ensure that your system packages match the operating system version:

```
[stack@director ~]$ sudo dnf distro-sync -y
[stack@director ~]$ sudo reboot
```

6.3. INSTALLING DIRECTOR PACKAGES

Install packages relevant to Red Hat OpenStack Platform director.

Procedure

1. Install the command line tools for director installation and configuration:

```
[stack@director ~]$ sudo dnf install -y python3-tripleoclient
```

6.4. PREPARING CONTAINER IMAGES

The undercloud installation requires an environment file to determine where to obtain container images and how to store them. Generate and customize this environment file that you can use to prepare your container images.

Procedure

1. Log in to your undercloud host as the **stack** user.
2. Generate the default container image preparation file:

```
$ openstack tripleo container image prepare default \
  --local-push-destination \
  --output-env-file containers-prepare-parameter.yaml
```

This command includes the following additional options:

- **--local-push-destination** sets the registry on the undercloud as the location for container images. With this option, director pulls the necessary images from the Red Hat Container Catalog and pushes the images to the registry on the undercloud. Director uses the undercloud registry as the container image source. To pull container images directly from the Red Hat Container Catalog, omit this option.
- **--output-env-file** specifies an environment file that includes the parameters for preparing your container images. In this example, the name of the file is **containers-prepare-parameter.yaml**.



NOTE

You can use the same **containers-prepare-parameter.yaml** file to define a container image source for both the undercloud and the overcloud.

3. Modify the **containers-prepare-parameter.yaml** to suit your requirements.

6.5. CONTAINER IMAGE PREPARATION PARAMETERS

The default file for preparing your containers (**containers-prepare-parameter.yaml**) contains the **ContainerImagePrepare** heat parameter. This parameter defines a list of strategies for preparing a set of images:

```
parameter_defaults:
  ContainerImagePrepare:
    - (strategy one)
    - (strategy two)
    - (strategy three)
    ...
```

Each strategy accepts a set of sub-parameters that defines which images to use and what to do with the images. The following table contains information about the sub-parameters you can use with each **ContainerImagePrepare** strategy:

Parameter	Description
excludes	List of image name substrings to exclude from a strategy.
includes	List of image name substrings to include in a strategy. At least one image name must match an existing image. All excludes are ignored if includes is specified.
modify_append_tag	String to append to the tag for the destination image. For example, if you pull an image with the tag 14.0-89 and set the modify_append_tag to -hotfix , the director tags the final image as 14.0-89-hotfix .
modify_only_with_labels	A dictionary of image labels that filter the images that you want to modify. If an image matches the labels defined, the director includes the image in the modification process.
modify_role	String of ansible role names to run during upload but before pushing the image to the destination registry.
modify_vars	Dictionary of variables to pass to modify_role .

Parameter	Description
push_destination	<p>Defines the namespace of the registry that you want to push images to during the upload process.</p> <ul style="list-style-type: none"> ● If set to true, the push_destination is set to the undercloud registry namespace using the hostname, which is the recommended method. ● If set to false, the push to a local registry does not occur and nodes pull images directly from the source. ● If set to a custom value, director pushes images to an external local registry. <p>Do not set this parameter to false in production environments. If the push_destination parameter is set to false or is not defined and the remote registry requires authentication, set the ContainerImageRegistryLogin parameter to true and include the credentials with the ContainerImageRegistryCredentials parameter.</p>
pull_source	The source registry from where to pull the original container images.
set	A dictionary of key: value definitions that define where to obtain the initial images.
tag_from_label	Defines the label pattern to tag the resulting images. Usually sets to {version}-{release} .



IMPORTANT

When you push images to the undercloud, use **push_destination: true** instead of **push_destination: UNDERCLOUD_IP:PORT**. The **push_destination: true** method provides a level of consistency across both IPv4 and IPv6 addresses.

The **set** parameter accepts a set of **key: value** definitions:

Key	Description
ceph_image	The name of the Ceph Storage container image.
ceph_namespace	The namespace of the Ceph Storage container image.
ceph_tag	The tag of the Ceph Storage container image.

Key	Description
name_prefix	A prefix for each OpenStack service image.
name_suffix	A suffix for each OpenStack service image.
namespace	The namespace for each OpenStack service image.
neutron_driver	The driver to use to determine which OpenStack Networking (neutron) container to use. Use a null value to set to the standard neutron-server container. Set to ovn to use OVN-based containers.
tag	The tag that the director uses to identify the images to pull from the source registry. You usually keep this key set to the default value, which is the Red Hat OpenStack Platform version number.

**NOTE**

The container images use multi-stream tags based on Red Hat OpenStack Platform version. This means there is no longer a **latest** tag.

The **ContainerImageRegistryCredentials** parameter maps a container registry to a username and password to authenticate to that registry.

If a container registry requires a username and password, you can use **ContainerImageRegistryCredentials** to include credentials with the following syntax:

```
ContainerImagePrepare:
- push_destination: true
  set:
    namespace: registry.redhat.io/...
    ...
ContainerImageRegistryCredentials:
  registry.redhat.io:
    my_username: my_password
```

In the example, replace **my_username** and **my_password** with your authentication credentials. Instead of using your individual user credentials, Red Hat recommends creating a registry service account and using those credentials to access **registry.redhat.io** content. For more information, see ["Red Hat Container Registry Authentication"](#).

The **ContainerImageRegistryLogin** parameter is used to control the registry login on the systems being deployed. This must be set to **true** if **push_destination** is set to false or not used.

```
ContainerImagePrepare:
- set:
    namespace: registry.redhat.io/...
    ...
ContainerImageRegistryCredentials:
```

```
registry.redhat.io:
  my_username: my_password
ContainerImageRegistryLogin: true
```

If you have configured **push_destination**, do not set **ContainerImageRegistryLogin** to **true**. If you set this option to **true** and the overcloud nodes do not have network connectivity to the registry hosts defined in **ContainerImageRegistryCredentials**, the deployment might fail when trying to perform a login.

6.6. OBTAINING CONTAINER IMAGES FROM PRIVATE REGISTRIES

Some container image registries require authentication to access images. In this situation, use the **ContainerImageRegistryCredentials** parameter in your **containers-prepare-parameter.yaml** environment file.

```
parameter_defaults:
  ContainerImagePrepare:
    - (strategy one)
    - (strategy two)
    - (strategy three)
  ContainerImageRegistryCredentials:
    registry.example.com:
      username: "p@55w0rd!"
```

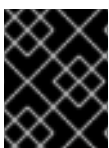


IMPORTANT

Private registries require **push_destination** set to **true** for their respective strategy in the **ContainerImagePrepare**.

The **ContainerImageRegistryCredentials** parameter uses a set of keys based on the private registry URL. Each private registry URL uses its own key and value pair to define the username (key) and password (value). This provides a method to specify credentials for multiple private registries.

```
parameter_defaults:
  ...
  ContainerImageRegistryCredentials:
    registry.redhat.io:
      myuser: 'p@55w0rd!'
    registry.internalsite.com:
      myuser2: '0th3rp@55w0rd!'
    '192.0.2.1:8787':
      myuser3: '@n0th3rp@55w0rd!'
```



IMPORTANT

The default **ContainerImagePrepare** parameter pulls container images from **registry.redhat.io**, which requires authentication.

The **ContainerImageRegistryLogin** parameter is used to control whether the system needs to log in to the remote registry to fetch the containers.

```
parameter_defaults:
...
ContainerImageRegistryLogin: true
```



IMPORTANT

You must set **ContainerImageRegistryLogin** to **true** if **push_destination** is not configured for a given strategy. If **push_destination** is configured in a **ContainerImagePrepare** strategy and the **ContainerImageRegistryCredentials** parameter is configured, the system logs in to fetch the containers and pushes them to the remote system. If the overcloud nodes do not have network connectivity to the registry hosts defined in the **ContainerImageRegistryCredentials**, set **push_destination** to **true** and **ContainerImageRegistryLogin** to **false**.

6.7. OBTAINING TRANSITIONAL CONTAINERS FOR UPGRADES

The upgrade requires containers from previous versions of Red Hat OpenStack Platform and Red Hat Ceph Storage. These containers help transition to Red Hat OpenStack Platform 16.1.

Procedure

1. Log in to your undercloud host as the **stack** user.
2. Edit the **containers-prepare-parameter.yaml** file.
3. Add the following parameters to the **set** in the **ContainerImagePrepare** parameter:

```
parameter_defaults:
  ContainerImagePrepare:
    - push_destination: true
    set:
      ...
      name_prefix_stein: openstack-
      name_suffix_stein: "
      namespace_stein: registry.redhat.io/rhosp15-rhel8
      tag_stein: 15.0
      ceph3_namespace: registry.redhat.io/rhceph
      ceph3_tag: latest
      ceph3_image: rhceph-3-rhel7
      ...
```

- The ***_stein** parameters define the container images for Red Hat OpenStack Platform 15, which the upgrade process uses for database migration.
 - The **ceph3_*** parameters define the current Red Hat Ceph Storage container images that the overcloud uses. The overcloud requires these containers until the transition from Red Hat Ceph Storage 3 to 4.
 - If you use Red Hat Satellite for container image storage, set the namespaces to the image locations on the Red Hat Satellite server.
4. Change the **neutron_driver** parameter to **openvswitch**:

```
parameter_defaults:
```

```

ContainerImagePrepare:
- push_destination: true
  set:
    ...
    neutron_driver: openvswitch
    ...

```

The upgrade retains Open vSwitch compatibility throughout the process. After you complete the upgrade to Red Hat OpenStack Platform 16.1, you migrate the overcloud from Open vSwitch to Open Virtual Network (OVN).

5. Save the **containers-prepare-parameter.yaml** file.

6.8. UPDATING THE UNDERCLOUD.CONF FILE

You can continue using the original **undercloud.conf** file from your Red Hat OpenStack Platform 13 environment, but you must modify the file to retain compatibility with Red Hat OpenStack Platform 16.1.

Procedure

1. Log in to your undercloud host as the **stack** user.
2. Edit the **undercloud.conf** file.
3. Add the following parameter to the **DEFAULT** section in the file:

```
container_images_file = /home/stack/containers-prepare-parameter.yaml
```

This parameter defines the location of the **containers-prepare-parameter.yaml** environment file so that director pulls container images for the undercloud from the correct location.

4. Check the **generate_service_certificate** parameter. The default for this parameter changes from **false** to **true**, which enables SSL/TLS on your undercloud, during the upgrade.
5. Check the **local_interface** parameter if you have migrated to a predictable NIC naming convention.
6. If you set the **masquerade_network** parameter in Red Hat OpenStack Platform 13, remove this parameter and set **masquerade = true** for each subnet.
7. Check all other parameters in the file for any changes.
8. Save the file.

6.9. DIRECTOR CONFIGURATION PARAMETERS

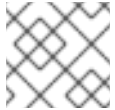
The following list contains information about parameters for configuring the **undercloud.conf** file. Keep all parameters within their relevant sections to avoid errors.

Defaults

The following parameters are defined in the **[DEFAULT]** section of the **undercloud.conf** file:

additional_architectures

A list of additional (kernel) architectures that an overcloud supports. Currently the overcloud supports **ppc64le** architecture.



NOTE

When you enable support for ppc64le, you must also set **ipxe_enabled** to **False**

certificate_generation_ca

The **certmonger** nickname of the CA that signs the requested certificate. Use this option only if you have set the **generate_service_certificate** parameter. If you select the **local** CA, certmonger extracts the local CA certificate to **/etc/pki/ca-trust/source/anchors/cm-local-ca.pem** and adds the certificate to the trust chain.

clean_nodes

Defines whether to wipe the hard drive between deployments and after introspection.

cleanup

Cleanup temporary files. Set this to **False** to leave the temporary files used during deployment in place after you run the deployment command. This is useful for debugging the generated files or if errors occur.

container_cli

The CLI tool for container management. Leave this parameter set to **podman**. Red Hat Enterprise Linux 8.2 only supports **podman**.

container_healthcheck_disabled

Disables containerized service health checks. Red Hat recommends that you enable health checks and leave this option set to **false**.

container_images_file

Heat environment file with container image information. This file can contain the following entries:

- Parameters for all required container images
- The **ContainerImagePrepare** parameter to drive the required image preparation. Usually the file that contains this parameter is named **containers-prepare-parameter.yaml**.

container_insecure_registries

A list of insecure registries for **podman** to use. Use this parameter if you want to pull images from another source, such as a private container registry. In most cases, **podman** has the certificates to pull container images from either the Red Hat Container Catalog or from your Satellite server if the undercloud is registered to Satellite.

container_registry_mirror

An optional **registry-mirror** configured that **podman** uses.

custom_env_files

Additional environment files that you want to add to the undercloud installation.

deployment_user

The user who installs the undercloud. Leave this parameter unset to use the current default user **stack**.

discovery_default_driver

Sets the default driver for automatically enrolled nodes. Requires the **enable_node_discovery** parameter to be enabled and you must include the driver in the **enabled_hardware_types** list.

**enable_ironic; enable_ironic_inspector; enable_mistral; enable_nova; enable_tempest;
enable_validations; enable_zaqar**

Defines the core services that you want to enable for director. Leave these parameters set to **true**.

enable_node_discovery

Automatically enroll any unknown node that PXE-boots the introspection ramdisk. New nodes use the **fake_pxe** driver as a default but you can set **discovery_default_driver** to override. You can also use introspection rules to specify driver information for newly enrolled nodes.

enable_novajoin

Defines whether to install the **novajoin** metadata service in the undercloud.

enable_routed_networks

Defines whether to enable support for routed control plane networks.

enable_swift_encryption

Defines whether to enable Swift encryption at-rest.

enable_telemetry

Defines whether to install OpenStack Telemetry services (gnocchi, aodh, panko) in the undercloud. Set the **enable_telemetry** parameter to **true** if you want to install and configure telemetry services automatically. The default value is **false**, which disables telemetry on the undercloud. This parameter is required if you use other products that consume metrics data, such as Red Hat CloudForms.

enabled_hardware_types

A list of hardware types that you want to enable for the undercloud.

generate_service_certificate

Defines whether to generate an SSL/TLS certificate during the undercloud installation, which is used for the **undercloud_service_certificate** parameter. The undercloud installation saves the resulting certificate **/etc/pki/tls/certs/undercloud-[undercloud_public_vip].pem**. The CA defined in the **certificate_generation_ca** parameter signs this certificate.

heat_container_image

URL for the heat container image to use. Leave unset.

heat_native

Run host-based undercloud configuration using **heat-all**. Leave as **true**.

hieradata_override

Path to **hieradata** override file that configures Puppet hieradata on the director, providing custom configuration to services beyond the **undercloud.conf** parameters. If set, the undercloud installation copies this file to the **/etc/puppet/hieradata** directory and sets it as the first file in the hierarchy. For more information about using this feature, see [Configuring hieradata on the undercloud](#).

inspection_extras

Defines whether to enable extra hardware collection during the inspection process. This parameter requires the **python-hardware** or **python-hardware-detect** packages on the introspection image.

inspection_interface

The bridge that director uses for node introspection. This is a custom bridge that the director configuration creates. The **LOCAL_INTERFACE** attaches to this bridge. Leave this as the default **br-ctlplane**.

inspection_runbench

Runs a set of benchmarks during node introspection. Set this parameter to **true** to enable the benchmarks. This option is necessary if you intend to perform benchmark analysis when inspecting the hardware of registered nodes.

ipa_otp

Defines the one-time password to register the undercloud node to an IPA server. This is required when **enable_novajoin** is enabled.

ipv6_address_mode

IPv6 address configuration mode for the undercloud provisioning network. The following list contains the possible values for this parameter:

- `dhcpv6-stateless` - Address configuration using router advertisement (RA) and optional information using DHCPv6.
- `dhcpv6-stateful` - Address configuration and optional information using DHCPv6.

ipxe_enabled

Defines whether to use iPXE or standard PXE. The default is **true**, which enables iPXE. Set this parameter to **false** to use standard PXE.

local_interface

The chosen interface for the director Provisioning NIC. This is also the device that director uses for DHCP and PXE boot services. Change this value to your chosen device. To see which device is connected, use the **ip addr** command. For example, this is the result of an **ip addr** command:

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 52:54:00:75:24:09 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.178/24 brd 192.168.122.255 scope global dynamic eth0
        valid_lft 3462sec preferred_lft 3462sec
    inet6 fe80::5054:ff:fe75:2409/64 scope link
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noop state DOWN
    link/ether 42:0b:c2:a5:c1:26 brd ff:ff:ff:ff:ff:ff
```

In this example, the External NIC uses **eth0** and the Provisioning NIC uses **eth1**, which is currently not configured. In this case, set the **local_interface** to **eth1**. The configuration script attaches this interface to a custom bridge defined with the **inspection_interface** parameter.

local_ip

The IP address defined for the director Provisioning NIC. This is also the IP address that director uses for DHCP and PXE boot services. Leave this value as the default **192.168.24.1/24** unless you use a different subnet for the Provisioning network, for example, if this IP address conflicts with an existing IP address or subnet in your environment.

local_mtu

The maximum transmission unit (MTU) that you want to use for the **local_interface**. Do not exceed 1500 for the undercloud.

local_subnet

The local subnet that you want to use for PXE boot and DHCP interfaces. The **local_ip** address should reside in this subnet. The default is **ctlplane-subnet**.

net_config_override

Path to network configuration override template. If you set this parameter, the undercloud uses a JSON format template to configure the networking with **os-net-config** and ignores the network parameters set in **undercloud.conf**. Use this parameter when you want to configure bonding or add an option to the interface. See **/usr/share/instack-undercloud/templates/net-config.json.template** for an example.

networks_file

Networks file to override for **heat**.

output_dir

Directory to output state, processed heat templates, and Ansible deployment files.

overcloud_domain_name

The DNS domain name that you want to use when you deploy the overcloud.



NOTE

When you configure the overcloud, you must set the **CloudDomain** parameter to a matching value. Set this parameter in an environment file when you configure your overcloud.

roles_file

The roles file that you want to use to override the default roles file for undercloud installation. It is highly recommended to leave this parameter unset so that the director installation uses the default roles file.

scheduler_max_attempts

The maximum number of times that the scheduler attempts to deploy an instance. This value must be greater or equal to the number of bare metal nodes that you expect to deploy at once to avoid potential race conditions when scheduling.

service_principal

The Kerberos principal for the service using the certificate. Use this parameter only if your CA requires a Kerberos principal, such as in FreeIPA.

subnets

List of routed network subnets for provisioning and introspection. The default value includes only the **ctlplane-subnet** subnet. For more information, see [Subnets](#).

templates

Heat templates file to override.

undercloud_admin_host

The IP address or hostname defined for director Admin API endpoints over SSL/TLS. The director configuration attaches the IP address to the director software bridge as a routed IP address, which uses the **/32** netmask.

undercloud_debug

Sets the log level of undercloud services to **DEBUG**. Set this value to **true** to enable **DEBUG** log level.

undercloud_enable_selinux

Enable or disable SELinux during the deployment. It is highly recommended to leave this value set to **true** unless you are debugging an issue.

undercloud_hostname

Defines the fully qualified host name for the undercloud. If set, the undercloud installation configures all system host name settings. If left unset, the undercloud uses the current host name, but you must configure all system host name settings appropriately.

undercloud_log_file

The path to a log file to store the undercloud install and upgrade logs. By default, the log file is **install-undercloud.log** in the home directory. For example, **/home/stack/install-undercloud.log**.

undercloud_nameservers

A list of DNS nameservers to use for the undercloud hostname resolution.

undercloud_ntp_servers

A list of network time protocol servers to help synchronize the undercloud date and time.

undercloud_public_host

The IP address or hostname defined for director Public API endpoints over SSL/TLS. The director configuration attaches the IP address to the director software bridge as a routed IP address, which uses the **/32** netmask.

undercloud_service_certificate

The location and filename of the certificate for OpenStack SSL/TLS communication. Ideally, you obtain this certificate from a trusted certificate authority. Otherwise, generate your own self-signed certificate.

undercloud_timezone

Host timezone for the undercloud. If you do not specify a timezone, director uses the existing timezone configuration.

undercloud_update_packages

Defines whether to update packages during the undercloud installation.

Subnets

Each provisioning subnet is a named section in the **undercloud.conf** file. For example, to create a subnet called **ctlplane-subnet**, use the following sample in your **undercloud.conf** file:

```
[ctlplane-subnet]
cidr = 192.168.24.0/24
dhcp_start = 192.168.24.5
dhcp_end = 192.168.24.24
inspection_iprange = 192.168.24.100,192.168.24.120
gateway = 192.168.24.1
masquerade = true
```

You can specify as many provisioning networks as necessary to suit your environment.

cidr

The network that director uses to manage overcloud instances. This is the Provisioning network, which the undercloud **neutron** service manages. Leave this as the default **192.168.24.0/24** unless you use a different subnet for the Provisioning network.

masquerade

Defines whether to masquerade the network defined in the **cidr** for external access. This provides the Provisioning network with a degree of network address translation (NAT) so that the Provisioning network has external access through director.



NOTE

The director configuration also enables IP forwarding automatically using the relevant **sysctl** kernel parameter.

dhcp_start; dhcp_end

The start and end of the DHCP allocation range for overcloud nodes. Ensure that this range contains enough IP addresses to allocate your nodes.

dhcp_exclude

IP addresses to exclude in the DHCP allocation range.

dns_nameservers

DNS nameservers specific to the subnet. If no nameservers are defined for the subnet, the subnet uses nameservers defined in the **undercloud_nameservers** parameter.

gateway

The gateway for the overcloud instances. This is the undercloud host, which forwards traffic to the External network. Leave this as the default **192.168.24.1** unless you use a different IP address for director or want to use an external gateway directly.

host_routes

Host routes for the Neutron-managed subnet for the overcloud instances on this network. This also configures the host routes for the **local_subnet** on the undercloud.

inspection_iprange

Temporary IP range for nodes on this network to use during the inspection process. This range must not overlap with the range defined by **dhcp_start** and **dhcp_end** but must be in the same IP subnet.

6.10. RUNNING THE DIRECTOR UPGRADE

Complete the following steps to upgrade the director.

Procedure

1. Run the following command to upgrade the director on the undercloud:

```
$ openstack undercloud upgrade
```

This command launches the director configuration script. The director upgrades its packages and configures its services to suit the settings in the **undercloud.conf**. This script takes several minutes to complete.



NOTE

The director configuration script prompts for confirmation before proceeding. Bypass this confirmation using the **-y** option:

```
$ openstack undercloud upgrade -y
```

2. The script also starts all OpenStack Platform service containers on the undercloud automatically. You manage each service through a **systemd** resource. Check the **systemd** resources:

```
$ sudo systemctl list-units "tripleo_*.service"
```

Each **systemd** service controls a container. Check the enabled containers using the following command:

```
$ sudo podman ps
```

3. The script adds the **stack** user to the **docker** group to ensure that the **stack** user has access to container management commands. Refresh the **stack** user permissions with the following command:

```
$ exec su -l stack
```

The command prompts you to log in again. Enter the stack user password.

4. To initialize the **stack** user to use the command line tools, run the following command:

```
$ source ~/stackrc
```

The prompt now indicates OpenStack commands authenticate and execute against the undercloud;

```
(undercloud) $
```

The director upgrade is complete.

PART II. PREPARING FOR OVERCLOUD UPGRADE

CHAPTER 7. INITIAL STEPS FOR OVERCLOUD PREPARATION

You must complete some initial steps to prepare for the overcloud upgrade.

7.1. PREPARING FOR OVERCLOUD SERVICE DOWNTIME

The overcloud upgrade process disables the main services at key points. This means you cannot use any overcloud services to create new resources during the upgrade duration. Workloads running in the overcloud remain active during this period, which means instances continue to run through the upgrade duration.

It is important to plan a maintenance window to ensure no users can access the overcloud services during the upgrade duration.

Affected by overcloud upgrade

- OpenStack Platform services

Unaffected by overcloud upgrade

- Instances running during the upgrade
- Ceph Storage OSDs (backend storage for instances)
- Linux networking
- Open vSwitch networking
- Undercloud

7.2. SELECTING COMPUTE NODES FOR UPGRADE TESTING

The overcloud upgrade process allows you to either:

- Upgrade all nodes in a role
- Individual nodes separately

To ensure a smooth overcloud upgrade process, it is useful to test the upgrade on a few individual Compute nodes in your environment before upgrading all Compute nodes. This ensures no major issues occur during the upgrade while maintaining minimal downtime to your workloads.

Use the following recommendations to help choose test nodes for the upgrade:

- Select two or three Compute nodes for upgrade testing
- Select nodes without any critical instances running
- If necessary, migrate critical instances from the selected test Compute nodes to other Compute nodes

7.3. VALIDATING THE PRE-UPGRADE REQUIREMENTS

Run the **pre-upgrade** validation group to check the pre-upgrade requirements.

Procedure

1. Source the **stackrc** file.

```
$ source ~/stackrc
```

2. Run the **openstack tripleo validator run** command with the **--group pre-upgrade** option:

```
$ openstack tripleo validator run --group pre-upgrade
```

3. Review the results of the validation report. To view detailed output from a specific validation, run the **openstack tripleo validator show run** command against the UUID of the specific validation from the report:

```
$ openstack tripleo validator show run <UUID>
```



IMPORTANT

A **FAILED** validation does not prevent you from deploying or running Red Hat OpenStack Platform. However, a **FAILED** validation can indicate a potential issue with a production environment.

7.4. DISABLING FENCING IN THE OVERCLOUD

When you upgrade the overcloud, you upgrade each Controller node individually to retain high availability functionality. During this period, the overcloud might detect certain nodes as disabled and attempt fencing operations, which can cause unintended behaviour.

If you have enabled fencing in the overcloud, you must temporarily disable fencing for the duration of the upgrade to avoid any unintended behaviour.



NOTE

If you use an environment file to enable fencing in your overcloud, the director re-enables fencing during the re-creation of the Controller node cluster.

Procedure

1. Log in to the undercloud as the **stack** user.
2. Source the **stackrc** file.

```
$ source ~/stackrc
```

3. Log in to a Controller node and run the Pacemaker command to disable fencing:

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

Additional Resources

- ["Fencing Controller nodes with STONITH"](#)

7.5. CREATING AN OVERCLOUD INVENTORY FILE

Generate an Ansible inventory file of all nodes in your environment with the **tripleo-ansible-inventory** command.

Procedure

1. Log in to the undercloud as the **stack** user.
2. Source the **stackrc** file.

```
$ source ~/stackrc
```

3. Create a static inventory file of all nodes:

```
$ tripleo-ansible-inventory --static-yaml-inventory ~/inventory.yaml --stack STACK_NAME
```

If you are not using the default **overcloud** stack name, set your stack name with the **--stack *STACK NAME*** option replacing ***STACK NAME*** with the name of your stack.

4. To execute Ansible playbooks on your environment, run the **ansible-playbook** command and include the full path of the dynamic inventory tool using the **-i** option. For example:

```
(undercloud) $ ansible-playbook -i ~/inventory.yaml PLAYBOOK
```

CHAPTER 8. CONFIGURING THE OVERCLOUD FOR A LEAPP UPGRADE

You must complete some Leapp configuration to prepare for the overcloud upgrade.

8.1. CREATING AN UPGRADES ENVIRONMENT FILE

The upgrade process uses an environment file to enable the upgrade process and configure specific upgrade parameters.

Procedure

1. Log in to the undercloud as the **stack** user.
2. Create an environment file called **upgrades-environment.yaml** in your **templates** directory:

```
$ touch templates/upgrades-environment.yaml
```

3. Edit the file and add the following mandatory content:

```
parameter_defaults:
  UpgradeInitCommand: |
    sudo dnf module disable -y container-tools:rhel8
    sudo dnf module enable -y container-tools:2.0
    sudo dnf module disable -y virt:rhel
    sudo dnf module enable -y virt:8.2
  UpgradeLeappCommandOptions: "--enablerepo rhel-8-for-x86_64-baseos-eus-rpms --
enablerepo rhel-8-for-x86_64-appstream-eus-rpms --enablerepo fast-datapath-for-rhel-8-
x86_64-rpms"
```

- **UpgradeInitCommand** sets a series of bash commands to run that director runs before starting the upgrade on a node. In this context, set this parameter to use the **dnf** commands to switch from the default **container-tools:rhel8** module to **container-tools:2.0**.
- **UpgradeLeappCommandOptions** sets additional options to pass to the Leapp tool. In this context, set this parameter to enable Red Hat OpenStack Platform repositories to help with the Leapp transition.

4. Save the **upgrades-environment.yaml** file.

8.2. UPGRADE PARAMETERS

Parameter	Description
UpgradeInitCommand	Command or script snippet to run on all overcloud nodes to initialize the upgrade process. For example, a repository switch.

Parameter	Description
UpgradeInitCommonCommand	Common commands required by the upgrades process. This should not normally be modified by the operator and is set and unset in the major-upgrade-composable-steps.yaml and major-upgrade-converge.yaml environment files.
UpgradeLeappCommandOptions	Additional command line options to append to the Leapp command.
UpgradeLeappDebug	Print debugging output when running Leapp. The default value is True .
UpgradeLeappDevelSkip	Skip Leapp checks by setting env variables when running Leapp in development/testing. For example, LEAPP_DEVEL_SKIP_RHSM=1.
UpgradeLeappEnabled	Use Leapp for operating system upgrade. The default value is False .
UpgradeLeappRebootTimeout	Timeout (seconds) for the OS upgrade phase via Leapp. The default value is 3600 .
UpgradeLeappToInstall	List of packages to install after Leapp upgrade.
UpgradeLeappToRemove	List of packages to remove during Leapp upgrade.

8.3. COPYING THE LEAPP DATA TO THE OVERCLOUD NODES

Each overcloud node requires the Leapp data files. The director automatically copies the data files in the **/etc/leapp/files** directory on the undercloud to the same location on each overcloud. Check that the **pes-events.json** and **repomap.csv** files still exist in the **/etc/leapp/files** directory of the undercloud:

Procedure

1. Log in to the undercloud as the **stack** user.
2. Check the **/etc/leapp/files** directory on the undercloud:

```
$ sudo ls /etc/leapp/files/
```



IMPORTANT

If these files were removed, download these files attached to the Knowledge Base article ["Data required by the Leapp utility for an in-place upgrade from RHEL 7 to RHEL 8"](#) and place these files in the **/etc/leapp/files/** directory on the undercloud.

8.4. USING PREDICTABLE NIC NAMES FOR OVERCLOUD NODES

Before you run the Leapp upgrade on overcloud nodes, you must check for kernel-based NIC names, which usually contain a **eth** prefix. These NIC names are usually unpredictable in terms of NIC assignments. The Leapp upgrade fails if any NICs on the node use the **eth** prefix.

Prerequisites

- The **playbook-nics.yaml** playbook created during the undercloud preparation process. The **playbook-nics.yaml** playbook accommodates most overcloud networking scenarios that use Ethernet devices, bridges, and Linux bonds. If your environment requires additional configuration beyond these device types, follow these recommendations before proceeding:
 - Test the playbook on a separate system with a similar networking configuration to your overcloud nodes
 - Modify the playbook to accommodate renaming the **eth** prefix within the configuration of other device types
 - Check the networking configuration of your overcloud nodes after you complete this procedure

Procedure

1. Log in to the undercloud as the **stack** user.
2. Run the **playbook-nics.yaml** playbook on all overcloud nodes:

```
$ ansible-playbook -i ~/inventory.yaml playbook-nics.yaml
```

The playbook sets the new NIC prefix to **em**. To set a different NIC prefix, set the **prefix** variable when running the playbook:

```
$ ansible-playbook -i ~/inventory.yaml -e prefix="mynic" playbook-nics.yaml
```

3. Reboot overcloud nodes using the standard reboot procedures. For more information, see ["Rebooting nodes"](#).

Resources

- ["How to perform an in-place upgrade to RHEL 8 when using kernel NIC names on RHEL 7"](#)
- ["Consistent Network Device Naming"](#)
- ["Understanding the Predictable Network Interface Device Names"](#)

8.5. SETTING THE SSH ROOT PERMISSION PARAMETER ON THE OVERCLOUD

The Leapp upgrade checks whether the **PermitRootLogin** parameter exists in the **/etc/ssh/sshd_config** file on each node. You must explicitly set this parameter to either **yes** or **no**.

For security purposes, set this parameter to **no** to disable SSH access to the root user on overcloud nodes.

Prerequisites

- An Ansible inventory file for your overcloud nodes.

Procedure

1. Log in to the undercloud as the **stack** user.
2. Create a file called **playbook-ssh.yaml** and paste the following content in the file:

```
---
- name: Configure SSH PermitRootLogin parameter
  hosts: overcloud
  become: yes
  tasks:
    - name: Set the PermitRootLogin parameter to no
      lineinfile:
        path: /etc/ssh/sshd_config
        state: present
        line: "PermitRootLogin no"
```

3. Run the playbook:

```
$ ansible-playbook -i ~/inventory.yaml playbook-ssh.yaml
```

CHAPTER 9. UPDATING COMPOSABLE SERVICES AND PARAMETERS

You must complete some composable service configuration to prepare for the overcloud upgrade.

9.1. UPDATING COMPOSABLE SERVICES IN CUSTOM ROLES_DATA FILES

This section contains information about new and deprecated composable services.

- If you use the default **roles_data** file, these services are included automatically.
- If you use a custom **roles_data** file, add the new services and remove the deprecated services for each relevant role.

Controller Nodes

The following services have been deprecated for Controller nodes. Remove them from your Controller role.

Service	Reason
OS::TripleO::Services::AodhApi OS::TripleO::Services::AodhEvaluator OS::TripleO::Services::AodhListener OS::TripleO::Services::AodhNotifier OS::TripleO::Services::CeilometerApi OS::TripleO::Services::CeilometerCollector OS::TripleO::Services::CeilometerExpirer OS::TripleO::Services::MongoDb OS::TripleO::Services::PankoApi	OpenStack Platform no longer includes OpenStack Telemetry services and now uses Service Telemetry Framework (STF) for metrics and monitoring.
OS::TripleO::Services::Congress	Congress is no longer supported.
OS::TripleO::Services::GlanceRegistry	This service is no longer supported due to OpenStack Platform Image Service (glance) API v2.
OS::TripleO::Services::NeutronCorePluginP umgrid OS::TripleO::Services::NeutronCorePluginMi donet	Deprecated plug-ins for OpenStack Networking (neutron).

Service	Reason
OS::TripleO::Services::NeutronLbaasv2Agent OS::TripleO::Services::NeutronLbaasv2Api	OpenStack Networking (neutron) Load Balancing as a Service deprecated in favour of Octavia.
OS::TripleO::Services::NovaConsoleauth	This service is now deprecated.
OS::TripleO::Services::NovaPlacement	Deprecated in favor of OS::TripleO::Services::PlacementApi.
OS::TripleO::Services::OpenDaylightApi OS::TripleO::Services::OpenDaylightOvs	OpenDaylight is no longer supported.
OS::TripleO::Services::RabbitMQ	This service has been substituted for two new services: OS::TripleO::Services::OsloMessagingRpc OS::TripleO::Services::OsloMessagingNotify
OS::TripleO::Services::SkydiveAgent OS::TripleO::Services::SkydiveAnalyzer	Skydive is no longer supported.
OS::TripleO::Services::Tacker	Tacker is no longer supported.

The following services are new for Controller nodes. Add them to your Controller role.

Service	Reason
OS::TripleO::Services::CephGrafana	Tasks to enable Ceph Dashboard service.
OS::TripleO::Services::CinderBackendDellEMCPowermax OS::TripleO::Services::CinderBackendDellEMCSc OS::TripleO::Services::CinderBackendNVMeOF	New backends for Block Storage (cinder).
OS::TripleO::Services::ContainerImagePrepare	Run the commands to automatically pull and prepare container images relevant to the services in your overcloud.

Service	Reason
OS::TripleO::Services::DesignateApi OS::TripleO::Services::DesignateCentral OS::TripleO::Services::DesignateProducer OS::TripleO::Services::DesignateWorker OS::TripleO::Services::DesignateMDNS OS::TripleO::Services::DesignateSink	Services for DNS-as-a-Service (designate).
OS::TripleO::Services::IronicInspector	Service for Bare Metal Introspection for the overcloud.
OS::TripleO::Services::IronicNeutronAgent	The networking agent for OpenStack Bare Metal (ironic).
OS::TripleO::Services::NeutronAgentsIBConfig	Service for OpenStack Networking (neutron) agent for Mellanox InfiniBand.
OS::TripleO::Services::OpenStackClients	Service for installing the Red Hat OpenStack Platform command line tools.
OS::TripleO::Services::OsloMessagingRpc OS::TripleO::Services::OsloMessagingNotify	Replacement services for the OS::TripleO::Services::RabbitMQ service.
OS::TripleO::Services::PlacementApi	Service for the Placement API.

Compute Nodes

The following services have been deprecated for Compute nodes. Remove them from your Compute role.

Service	Reason
OS::TripleO::Services::OpenDaylightOvs	OpenDaylight is no longer supported.
OS::TripleO::Services::SkydiveAgent	Skydive is no longer supported.

The following services are new for Compute nodes. Add them to your Compute role.

Service	Reason
OS::TripleO::Services::NovaAZConfig	Service to configure host aggregate and availability zone in OpenStack Compute (nova).

All Nodes

The following services have been deprecated for all nodes. Remove them from all roles.

Service	Reason
OS::TripleO::Services::Docker	Replaced with Podman.
OS::TripleO::Services::Fluentd	Deprecated in favour of OS::TripleO::Services::Rsyslog .
OS::TripleO::Services::Ntp	Deprecated in favour of OS::TripleO::Services::Timesync .
OS::TripleO::Services::SensuClient	Deprecated service.
OS::TripleO::Services::Ptp	Deprecated in favour of OS::TripleO::Services::Timesync .

The following services are new for all nodes. Add them to all roles.

Service	Reason
OS::TripleO::Services::BootParams	Service to set Kernel arguments, Tuned profiles, and CPU isolation.
OS::TripleO::Services::Collectd	Service to configure Collectd.
OS::TripleO::Services::Multipathd	Provides a Multipathd service, which is disabled by default
OS::TripleO::Services::Podman	Service to install and enable Podman.
OS::TripleO::Services::Rear	Service to install and enable Relax-and-Recover (ReaR) backup and restore tools.
OS::TripleO::Services::Rsyslog	Service to configure centralized log collection.
OS::TripleO::Services::Timesync	Service to enable time synchronization method, which is Chronyd by default.

9.2. UPDATING COMPOSABLE SERVICES IN CUSTOM ENVIRONMENT FILES

If you have any custom environment files with **resource_registry** sections, check the **resource_registry** sections for changes in composable service template mappings. The composable service files for Red Hat OpenStack Platform 16.1 reside in a new location within **/usr/share/openstack-tripleo-heat-templates/**:

Red Hat OpenStack Platform 13	Red Hat OpenStack Platform 16.1
docker/services/	deployment

The **deployment** directory contains a set of sub-directories to group composable services. For example, the **keystone** sub-directory contains composable service templates for OpenStack Identity (keystone).

To remap composable services in custom environment files, check the template location for the current service mapping and edit the mapping to the new location. This procedure uses **ceph-mgr.yaml** as an example.



IMPORTANT

This procedure acts as guidance only for remapping composable services. If you are unsure of any mapping, contact [Red Hat](#) and request advice on the correct mapping.

Procedure

1. Search for custom environment files that use composable services. You usually store custom environment files in the **/home/stack/templates** directory:

```
$ cd ~/templates/
$ grep "OS::TripleO::Services" *
```

In this scenario, one of the files shows an outdated mapping:

```
OS::TripleO::Services::CephMgr: /usr/share/openstack-tripleo-heat-
templates/docker/services/ceph-ansible/ceph-mgr.yaml
```

2. Identify the new **ceph-mgr.yaml** location in **/usr/share/openstack-tripleo-heat-templates/**. This file is now located in the ``deployment/ceph-ansible'` directory:

```
$ find /usr/share/openstack-tripleo-heat-templates/ -name ceph-mgr.yaml
/usr/share/openstack-tripleo-heat-templates/deployment/ceph-ansible/ceph-mgr.yaml
```

3. Edit the service in the custom environment file:

```
resource_registry:
  OS::TripleO::Services::CephMgr: /usr/share/openstack-tripleo-heat-
templates/deployment/ceph-ansible/ceph-mgr.yaml
```

Save the file.

9.3. CONFIGURING ACCESS TO THE UNDERCLOUD REGISTRY

Add the control plane host name of the undercloud to the **DockerInsecureRegistryAddress** parameter. Place this parameter in the **containers-prepare-parameter.yaml** file to ensure that the parameter is included in future overcloud deployments.

Procedure

1. Log in to the undercloud as the **stack** user.
2. Obtain the control plane host name on the undercloud:

```
$ sudo hiera container_image_prepare_node_names
["undercloud.ctlplane.localdomain"]
```

3. Edit the **containers-prepare-parameter.yaml** file and add the **DockerInsecureRegistryAddress** parameter with a YAML list that contains the undercloud control plane host name:

```
parameter_defaults:
  DockerInsecureRegistryAddress:
    - undercloud.ctlplane.localdomain:8787
    ...
```

You must also append the port number of the overcloud registry to the host name value. The port number is **8787**.

9.4. DEPRECATED AND REMOVED FILTERS FOR THE NOVASCHEDULERDEFAULTFILTERS PARAMETER

If your environment uses a custom **NovaSchedulerDefaultFilters** parameter, edit the parameter to remove the following deprecated and removed filters:

Filter	Status
AggregateCoreFilter	Deprecated
AggregateRamFilter	Deprecated
AggregateDiskFilter	Deprecated
ExactCoreFilter	Removed
ExactRamFilter	Removed
ExactDiskFilter	Removed
CoreFilter	Removed
RamFilter	Removed
DiskFilter	Removed
RetryFilter	Deprecated



IMPORTANT

Avoid using filters marked **Deprecated**. Red Hat OpenStack Platform 16.1 still includes deprecated filters but future versions of Red Hat OpenStack Platform will not include these filters.

9.5. SETTING THE COMPUTE NAME FORMAT

Red Hat OpenStack Platform 13 uses **%stackname%-compute-%index%** as the default naming format for Compute nodes. Red Hat OpenStack Platform 16.1 uses **%stackname%-novacompute-%index%** as the default naming format for Compute nodes. Change the default naming format to retain the original Red Hat OpenStack Platform 13 naming format. If you do not use the original naming format, director configures new OpenStack Compute (nova) agents with the new naming format and keeps the existing OpenStack Compute (nova) agents with the old naming format as orphaned services.

Procedure

1. Log in to the undercloud as the **stack** user.
2. Set the Compute naming format:
 - If you use a custom **roles_data** file, edit the custom **roles_data** file and set the **HostnameFormatDefault** parameter for the **Compute** role:

```
- name: Compute
...
  HostnameFormatDefault: '%stackname%-compute-%index%'
...
```

Save the custom **roles_data** file.

- If you use the default **roles_data** file in **openstack-tripleo-heat-templates**, set the naming format in an environment file. Edit the environment file with your node counts and flavors, which is usually named **node-info.yaml**. Add the **ComputeHostnameFormat** parameter to the **parameter_defaults** section:

```
parameter_defaults:
...
  ComputeHostnameFormat: '%stackname%-compute-%index%'
...
```

Save the **node-info.yaml** file.

9.6. UPDATING YOUR SSL/TLS CONFIGURATION

If your overcloud uses SSL/TLS for public endpoints, you must unset the **InternalTLSCAFile** parameter when you upgrade to Red Hat OpenStack Platform 16.1.

For more information about SSL/TLS for public endpoints, see ["Enabling SSL/TLS on Overcloud Public Endpoints"](#).

Procedure

1. Log in to the undercloud as the **stack** user.

2. Source the **stackrc** file:

```
$ source ~/stackrc
```

3. Edit your custom overcloud SSL/TLS public endpoint file, which is usually named **~/templates/enable-tls.yaml**.
4. Remove the **NodeTLSData** resource from the `resource_registry`:

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

The overcloud deployment uses a new service in HAProxy to determine if SSL/TLS is enabled.



NOTE

If this is the only resource in the **resource_registry** section of the **enable-tls.yaml** file, remove the complete **resource_registry** section.

5. Add the **InternalTLSCAFile** parameter to the **parameter_defaults** section of the file. Set the value of **InternalTLSCAFile** to an empty string ("):

```
parameter_defaults:
  InternalTLSCAFile: "
  SSLCertificate: |
    -----BEGIN CERTIFICATE-----
    MIIDgzCCAmugAwIBAgIJAKk46qw6ncJaMA0GCSqGS
    ...
    sFW3S2roS4X0Af/kSSD8mIBBTFTCMBAj6rtLBKLaQ
    -----END CERTIFICATE-----
```

6. Save the SSL/TLS public endpoint file file.

9.7. UPDATING POST CONFIGURATION TEMPLATES

If you use the **OS::TripleO::NodeExtraConfigPost** resource to register and run a post-configuration template, you must add the **EndpointMap** parameter to the template.

Procedure

1. Edit the post-configuration template.
2. Under the **parameters** section, add the **EndpointMap** parameter and its sub-parameters:

```
parameters:
  servers:
    type: json
  nameserver_ip:
    type: string
  DeployIdentifier:
    type: string
```

```
EndpointMap:  
  default: {}  
  type: json
```

3. Save the template.

CHAPTER 10. UPDATING OVERCLOUD REGISTRATION TO THE RED HAT CUSTOMER PORTAL

Red Hat OpenStack Platform 16.1 uses Ansible-based methods to register overcloud nodes to the Red Hat Customer Portal.

10.1. RED HAT SUBSCRIPTION MANAGER (RHSM) COMPOSABLE SERVICE

The **rhsm** composable service provides a method to register overcloud nodes through Ansible. Each role in the default **roles_data** file contains a **OS::TripleO::Services::Rhsm** resource, which is disabled by default. To enable the service, register the resource to the **rhsm** composable service file:

```
resource_registry:
  OS::TripleO::Services::Rhsm: /usr/share/openstack-tripleo-heat-templates/deployment/rhsm/rhsm-
  baremetal-ansible.yaml
```

The **rhsm** composable service accepts a **RhsmVars** parameter, which allows you to define multiple sub-parameters relevant to your registration. For example:

```
parameter_defaults:
  RhsmVars:
    rhsm_repos:
      - rhel-8-for-x86_64-baseos-eus-rpms
      - rhel-8-for-x86_64-appstream-eus-rpms
      - rhel-8-for-x86_64-highavailability-eus-rpms
      ...
    rhsm_username: "myusername"
    rhsm_password: "p@55w0rd!"
    rhsm_org_id: "1234567"
    rhsm_release: 8.2
```

You can also use the **RhsmVars** parameter in combination with role-specific parameters (e.g. **ControllerParameters**) to provide flexibility when enabling specific repositories for different nodes types.

10.2. RHSMVARS SUB-PARAMETERS

See the [role documentation](#) to learn about all Ansible parameters.

rhsm	Description
rhsm_method	Choose the registration method. Either portal , satellite , or disable .
rhsm_org_id	The organization to use for registration. To locate this ID, run sudo subscription-manager orgs from the undercloud node. Enter your Red Hat credentials when prompted, and use the resulting Key value. For more information on your organization ID, see " Understanding the Red Hat Subscription Management Organization ID ".

rhsm	Description
rhsm_pool_ids	The subscription pool ID to use. Use this if not auto-attaching subscriptions. To locate this ID, run sudo subscription-manager list --available --all --matches="OpenStack" from the undercloud node, and use the resulting Pool ID value.
rhsm_activation_key	The activation key to use for registration. Does not work when rhsm_repos is configured.
rhsm_autosubscribe	Automatically attach compatible subscriptions to this system. Set to true to enable.
rhsm_baseurl	The base URL for obtaining content. The default is the Red Hat Content Delivery Network URL. If using a Satellite server, change this value to the base URL of your Satellite server content repositories.
rhsm_server_hostname	The hostname of the subscription management service for registration. The default is the Red Hat Subscription Management hostname. If using a Satellite server, change this value to your Satellite server hostname.
rhsm_repos	A list of repositories to enable. Does not work when rhsm_activation_key is configured.
rhsm_username	The username for registration. If possible, use activation keys for registration.
rhsm_password	The password for registration. If possible, use activation keys for registration.
rhsm_release	Red Hat Enterprise Linux release for pinning the repositories. This is set to 8.2 for Red Hat OpenStack Platform 16.1.
rhsm_rhsm_proxy_host name	The hostname for the HTTP proxy. For example: proxy.example.com .
rhsm_rhsm_proxy_port	The port for HTTP proxy communication. For example: 8080 .
rhsm_rhsm_proxy_user	The username to access the HTTP proxy.
rhsm_rhsm_proxy_pass word	The password to access the HTTP proxy.

10.3. SWITCHING TO THE RHSM COMPOSABLE SERVICE

The previous **rhel-registration** method runs a bash script to handle the overcloud registration. The scripts and environment files for this method are located in the core Heat template collection at **/usr/share/openstack-tripleo-heat-templates/extraconfig/pre_deploy/rhel-registration/**.

Complete the following steps to switch from the **rhel-registration** method to the **rhsm** composable service.

Procedure

1. Exclude the **rhel-registration** environment files from future deployments operations. In most cases, exclude the following files:
 - **rhel-registration/environment-rhel-registration.yaml**
 - **rhel-registration/rhel-registration-resource-registry.yaml**
2. If you use a custom **roles_data** file, ensure that each role in your **roles_data** file contains the **OS::TripleO::Services::Rhsm** composable service. For example:

```
- name: Controller
  description: |
    Controller role that has all the controller services loaded and handles
    Database, Messaging and Network functions.
  CountDefault: 1
  ...
  ServicesDefault:
    ...
    - OS::TripleO::Services::Rhsm
    ...
```

3. Add the environment file for **rhsm** composable service parameters to future deployment operations.

This method replaces the **rhel-registration** parameters with the **rhsm** service parameters and changes the Heat resource that enables the service from:

```
resource_registry:
  OS::TripleO::NodeExtraConfig: rhel-registration.yaml
```

To:

```
resource_registry:
  OS::TripleO::Services::Rhsm: /usr/share/openstack-tripleo-heat-templates/deployment/rhsm/rhsm-
  baremetal-ansible.yaml
```

You can also include the **/usr/share/openstack-tripleo-heat-templates/environments/rhsm.yaml** environment file with your deployment to enable the service.

10.4. RHEL-REGISTRATION TO RHSM MAPPINGS

rhel-registration	rhsm / RhsmVars
rhel_reg_method	rhsm_method
rhel_reg_org	rhsm_org_id

rhel-registration	rhsm / RhsmVars
rhel_reg_pool_id	rhsm_pool_ids
rhel_reg_activation_key	rhsm_activation_key
rhel_reg_auto_attach	rhsm_autosubscribe
rhel_reg_sat_url	rhsm_satellite_url
rhel_reg_repos	rhsm_repos
rhel_reg_user	rhsm_username
rhel_reg_password	rhsm_password
rhel_reg_release	rhsm_release
rhel_reg_http_proxy_host	rhsm_rhsm_proxy_hostname
rhel_reg_http_proxy_port	rhsm_rhsm_proxy_port
rhel_reg_http_proxy_username	rhsm_rhsm_proxy_user
rhel_reg_http_proxy_password	rhsm_rhsm_proxy_password

10.5. REGISTERING THE OVERCLOUD WITH THE RHSM COMPOSABLE SERVICE

Use the following procedure to create an environment file that enables and configures the **rhsm** composable service. The director uses this environment file to register and subscribe your nodes.

Procedure

1. Create an environment file (**templates/rhsm.yaml**) to store the configuration.
2. Include your configuration in the environment file. For example:

```
resource_registry:
  OS::TripleO::Services::Rhsm: /usr/share/openstack-tripleo-heat-
    templates/deployment/rhsm/rhsm-baremetal-ansible.yaml
parameter_defaults:
  RhsmVars:
    rhsm_repos:
      - rhel-8-for-x86_64-baseos-eus-rpms
      - rhel-8-for-x86_64-appstream-eus-rpms
      - rhel-8-for-x86_64-highavailability-eus-rpms
    ...
  rhsm_username: "myusername"
```

```
rhsm_password: "p@55w0rd!"
rhsm_org_id: "1234567"
rhsm_pool_ids: "1a85f9223e3d5e43013e3d6e8ff506fd"
rhsm_method: "portal"
rhsm_release: 8.2
```

- The **resource_registry** section associates the **rhsm** composable service with the **OS::TripleO::Services::Rhsm** resource, which is available on each role.
- The **RhsmVars** variable passes parameters to Ansible for configuring your Red Hat registration.

3. Save the environment file.

10.6. APPLYING THE RHSM COMPOSABLE SERVICE TO DIFFERENT ROLES

You can apply the **rhsm** composable service on a per-role basis. For example, you can apply different sets of configurations to Controller nodes, Compute nodes, and Ceph Storage nodes.

Procedure

1. Create an environment file (**templates/rhsm.yaml**) to store the configuration.
2. Include your configuration in the environment file. For example:

```
resource_registry:
  OS::TripleO::Services::Rhsm: /usr/share/openstack-tripleo-heat-
    templates/deployment/rhsm/rhsm-baremetal-ansible.yaml
parameter_defaults:
  ControllerParameters:
    RhsmVars:
      rhsm_repos:
        - rhel-8-for-x86_64-baseos-eus-rpms
        - rhel-8-for-x86_64-appstream-eus-rpms
        - rhel-8-for-x86_64-highavailability-eus-rpms
        - ansible-2.9-for-rhel-8-x86_64-rpms
        - advanced-virt-for-rhel-8-x86_64-rpms
        - openstack-16.1-for-rhel-8-x86_64-rpms
        - rhceph-4-mon-for-rhel-8-x86_64-rpms
        - rhceph-4-tools-for-rhel-8-x86_64-rpms
        - fast-datapath-for-rhel-8-x86_64-rpms
      rhsm_username: "myusername"
      rhsm_password: "p@55w0rd!"
      rhsm_org_id: "1234567"
      rhsm_pool_ids: "55d251f1490556f3e75aa37e89e10ce5"
      rhsm_method: "portal"
      rhsm_release: 8.2
  ComputeParameters:
    RhsmVars:
      rhsm_repos:
        - rhel-8-for-x86_64-baseos-eus-rpms
        - rhel-8-for-x86_64-appstream-eus-rpms
        - rhel-8-for-x86_64-highavailability-eus-rpms
        - ansible-2.9-for-rhel-8-x86_64-rpms
```

```

- advanced-virt-for-rhel-8-x86_64-rpms
- openstack-16.1-for-rhel-8-x86_64-rpms
- rhceph-4-tools-for-rhel-8-x86_64-rpms
rhsm_username: "myusername"
rhsm_password: "p@55w0rd!"
rhsm_org_id: "1234567"
rhsm_pool_ids: "55d251f1490556f3e75aa37e89e10ce5"
rhsm_method: "portal"
rhsm_release: 8.2
CephStorageParameters:
RhsmVars:
rhsm_repos:
- rhel-8-for-x86_64-baseos-rpms
- rhel-8-for-x86_64-appstream-rpms
- rhel-8-for-x86_64-highavailability-rpms
- ansible-2.9-for-rhel-8-x86_64-rpms
- openstack-16.1-deployment-tools-for-rhel-8-x86_64-rpms
- rhceph-4-osd-for-rhel-8-x86_64-rpms
- rhceph-4-tools-for-rhel-8-x86_64-rpms
rhsm_username: "myusername"
rhsm_password: "p@55w0rd!"
rhsm_org_id: "1234567"
rhsm_pool_ids: "68790a7aa2dc9dc50a9bc39fab55e0d"
rhsm_method: "portal"
rhsm_release: 8.2

```

The **resource_registry** associates the **rhsm** composable service with the **OS::TripleO::Services::Rhsm** resource, which is available on each role.

The **ControllerParameters**, **ComputeParameters**, and **CephStorageParameters** use their own **RhsmVars** parameter to pass subscription details to their respective roles.



NOTE

Set the **RhsmVars** parameter within the **CephStorageParameters** parameter to use a Red Hat Ceph Storage subscription and repositories specific to Ceph Storage. Ensure the **rhsm_repos** parameter contains the standard Red Hat Enterprise Linux repositories instead of the Extended Update Support (EUS) repositories that Controller and Compute nodes require.

3. Save the environment file.

CHAPTER 11. UPDATING OVERCLOUD REGISTRATION TO RED HAT SATELLITE

Red Hat OpenStack Platform 16.1 uses Ansible-based methods to register overcloud nodes to a Red Hat Satellite 6 server.

11.1. RED HAT SUBSCRIPTION MANAGER (RHSM) COMPOSABLE SERVICE

The **rhsm** composable service provides a method to register overcloud nodes through Ansible. Each role in the default **roles_data** file contains a **OS::TripleO::Services::Rhsm** resource, which is disabled by default. To enable the service, register the resource to the **rhsm** composable service file:

```
resource_registry:
  OS::TripleO::Services::Rhsm: /usr/share/openstack-tripleo-heat-templates/deployment/rhsm/rhsm-
  baremetal-ansible.yaml
```

The **rhsm** composable service accepts a **RhsmVars** parameter, which allows you to define multiple sub-parameters relevant to your registration. For example:

```
parameter_defaults:
  RhsmVars:
    rhsm_repos:
      - rhel-8-for-x86_64-baseos-eus-rpms
      - rhel-8-for-x86_64-appstream-eus-rpms
      - rhel-8-for-x86_64-highavailability-eus-rpms
      ...
    rhsm_username: "myusername"
    rhsm_password: "p@55w0rd!"
    rhsm_org_id: "1234567"
    rhsm_release: 8.2
```

You can also use the **RhsmVars** parameter in combination with role-specific parameters (e.g. **ControllerParameters**) to provide flexibility when enabling specific repositories for different nodes types.

11.2. RHSMVARS SUB-PARAMETERS

See the [role documentation](#) to learn about all Ansible parameters.

rhsm	Description
rhsm_method	Choose the registration method. Either portal , satellite , or disable .
rhsm_org_id	The organization to use for registration. To locate this ID, run sudo subscription-manager orgs from the undercloud node. Enter your Red Hat credentials when prompted, and use the resulting Key value. For more information on your organization ID, see " Understanding the Red Hat Subscription Management Organization ID ".

rhsm	Description
rhsm_pool_ids	The subscription pool ID to use. Use this if not auto-attaching subscriptions. To locate this ID, run sudo subscription-manager list --available --all --matches="OpenStack" from the undercloud node, and use the resulting Pool ID value.
rhsm_activation_key	The activation key to use for registration. Does not work when rhsm_repos is configured.
rhsm_autosubscribe	Automatically attach compatible subscriptions to this system. Set to true to enable.
rhsm_baseurl	The base URL for obtaining content. The default is the Red Hat Content Delivery Network URL. If using a Satellite server, change this value to the base URL of your Satellite server content repositories.
rhsm_server_hostname	The hostname of the subscription management service for registration. The default is the Red Hat Subscription Management hostname. If using a Satellite server, change this value to your Satellite server hostname.
rhsm_repos	A list of repositories to enable. Does not work when rhsm_activation_key is configured.
rhsm_username	The username for registration. If possible, use activation keys for registration.
rhsm_password	The password for registration. If possible, use activation keys for registration.
rhsm_release	Red Hat Enterprise Linux release for pinning the repositories. This is set to 8.2 for Red Hat OpenStack Platform 16.1.
rhsm_rhsm_proxy_host name	The hostname for the HTTP proxy. For example: proxy.example.com .
rhsm_rhsm_proxy_port	The port for HTTP proxy communication. For example: 8080 .
rhsm_rhsm_proxy_user	The username to access the HTTP proxy.
rhsm_rhsm_proxy_pass word	The password to access the HTTP proxy.

11.3. SWITCHING TO THE RHSM COMPOSABLE SERVICE

The previous **rhel-registration** method runs a bash script to handle the overcloud registration. The scripts and environment files for this method are located in the core Heat template collection at **/usr/share/openstack-tripleo-heat-templates/extraconfig/pre_deploy/rhel-registration/**.

Complete the following steps to switch from the **rhel-registration** method to the **rhsm** composable service.

Procedure

1. Exclude the **rhel-registration** environment files from future deployments operations. In most cases, exclude the following files:
 - **rhel-registration/environment-rhel-registration.yaml**
 - **rhel-registration/rhel-registration-resource-registry.yaml**
2. If you use a custom **roles_data** file, ensure that each role in your **roles_data** file contains the **OS::TripleO::Services::Rhsm** composable service. For example:

```
- name: Controller
  description: |
    Controller role that has all the controller services loaded and handles
    Database, Messaging and Network functions.
  CountDefault: 1
  ...
  ServicesDefault:
    ...
    - OS::TripleO::Services::Rhsm
    ...
```

3. Add the environment file for **rhsm** composable service parameters to future deployment operations.

This method replaces the **rhel-registration** parameters with the **rhsm** service parameters and changes the Heat resource that enables the service from:

```
resource_registry:
  OS::TripleO::NodeExtraConfig: rhel-registration.yaml
```

To:

```
resource_registry:
  OS::TripleO::Services::Rhsm: /usr/share/openstack-tripleo-heat-templates/deployment/rhsm/rhsm-
  baremetal-ansible.yaml
```

You can also include the **/usr/share/openstack-tripleo-heat-templates/environments/rhsm.yaml** environment file with your deployment to enable the service.

11.4. RHEL-REGISTRATION TO RHSM MAPPINGS

rhel-registration	rhsm / RhsmVars
rhel_reg_method	rhsm_method
rhel_reg_org	rhsm_org_id

rhel-registration	rhsm / RhsmVars
rhel_reg_pool_id	rhsm_pool_ids
rhel_reg_activation_key	rhsm_activation_key
rhel_reg_auto_attach	rhsm_autosubscribe
rhel_reg_sat_url	rhsm_satellite_url
rhel_reg_repos	rhsm_repos
rhel_reg_user	rhsm_username
rhel_reg_password	rhsm_password
rhel_reg_release	rhsm_release
rhel_reg_http_proxy_host	rhsm_rhsm_proxy_hostname
rhel_reg_http_proxy_port	rhsm_rhsm_proxy_port
rhel_reg_http_proxy_username	rhsm_rhsm_proxy_user
rhel_reg_http_proxy_password	rhsm_rhsm_proxy_password

11.5. REGISTERING THE OVERCLOUD TO RED HAT SATELLITE

Use the following procedure to create an environment file that enables and configures the **rhsm** composable service to register nodes to Red Hat Satellite instead of the Red Hat Customer Portal.

Procedure

1. Create an environment file (**templates/rhsm.yaml**) to store the configuration.
2. Include your configuration in the environment file. For example:

```
resource_registry:
  OS::TripleO::Services::Rhsm: /usr/share/openstack-tripleo-heat-
    templates/deployment/rhsm/rhsm-baremetal-ansible.yaml
parameter_defaults:
  RhsmVars:
    rhsm_activation_key: "myactivationkey"
    rhsm_method: "satellite"
    rhsm_org_id: "ACME"
    rhsm_server_hostname: satellite.example.com"
    rhsm_baseurl: "https://satellite.example.com/pulp/repos"
    rhsm_release: 8.2
```

The **resource_registry** associates the **rhsm** composable service with the **OS::TripleO::Services::Rhsm** resource, which is available on each role.

The **RhsmVars** variable passes parameters to Ansible for configuring your Red Hat registration.

3. Save the environment file.

11.6. PREPARING LEAPP TO USE SATELLITE SERVER

If you use Satellite 6 to host RPM content, complete these preparation steps to ensure a successful Leapp upgrade with Satellite.

Prerequisites

- A Satellite server activation key linked to repositories for Red Hat OpenStack Platform 16.1 and Red Hat Enterprise Linux 8.2.
- An Ansible inventory file for your overcloud nodes.

Procedure

1. Log in to the undercloud as the **stack** user.
2. Create a file called **playbook-satellite.yaml** and paste the following content in the file:

```
- name: Pre-install leapp
  hosts: overcloud
  become: yes
  tasks:
    - name: Pre-install leapp
      yum:
        name:
          - leapp
          - leapp-repository
        state: installed
    - name: Remove katello-host-tools-fact-plugin
      yum:
        name:
          - katello-host-tools-fact-plugin
        state: removed
    - name: Register system
      import_role:
        name: redhat-subscription
      vars:
        rhsm_activation_key: "osp16-key"
        rhsm_method: "satellite"
        rhsm_org_id: "ACME"
        rhsm_server_hostname: "satellite.example.com"
        rhsm_baseurl: "https://satellite.example.com/pulp/repos"
        rhsm_release: "8.2"
        rhsm_force_register: True
```

Modify the **rhsm_*** variables to suit your Satellite server.

3. Run the playbook:

—

```
$ ansible-playbook -i ~/inventory.yaml playbook-satellite.yaml
```

CHAPTER 12. PREPARING FOR A CEPH STORAGE UPGRADE

The upgrade process maintains the use of Red Hat Ceph Storage 3 containerized services through the duration of the upgrade to Red Hat OpenStack Platform 16.1. After you complete the Red Hat OpenStack Platform 16.1 upgrade, you upgrade the Ceph Storage services to Red Hat Ceph Storage 4.

12.1. UNDERSTANDING THE CEPH STORAGE NODE UPGRADE PROCESS AT A HIGH LEVEL

The Ceph Storage nodes continue to use Red Hat Ceph Storage 3 containers during the overcloud upgrade process. To understand how Ceph Storage nodes and services are impacted during the upgrade process, read the following summaries for each aspect of the Ceph Storage upgrade process.

ceph-ansible

ceph-ansible is a collection of roles and playbooks that director uses to install, maintain, and upgrade Ceph Storage services. When you upgraded the undercloud, you ran certain commands that ensured **ceph-ansible** remained at the latest version 3 collection after the transition to Red Hat Enterprise Linux 8.2. Version 3 of **ceph-ansible** keeps the containerized Ceph Storage services on version 3 through the duration of the overcloud upgrade. After you complete the upgrade, you enable the Red Hat Ceph Storage update the **Red Hat Ceph Storage Tools 4 for RHEL 8** repository and update **ceph-ansible** to version 4.

Migration to Podman

During the overcloud upgrade, you must run the **openstack overcloud external-upgrade run --tags ceph_systemd** command to change the **systemd** services that control Ceph Storage containerized services to use Podman instead of Docker. You run this command before performing the operating system upgrade on any node that contains Ceph Storage containerized services.

After you change the **systemd** services to use Podman on a node, you perform the operating system upgrade and the OpenStack Platform service upgrade. The Ceph Storage containers on that node will run again after the OpenStack Platform service upgrade.

Ceph Storage operating system upgrade

You follow the same workflow on Ceph Storage nodes as you do on overcloud nodes in general. When you run the **openstack overcloud upgrade run --tags system_upgrade** command against a Ceph Storage node, director runs Leapp on Ceph Storage node and upgrades the operating system to Red Hat Enterprise Linux 8.2. You then run the untagged **openstack overcloud upgrade run** command against the Ceph Storage node, which runs the following containers:

- Red Hat Ceph Storage 3 containerized services
- Red Hat OpenStack Platform 16.1 containerized services

Upgrading to Red Hat Ceph Storage 4

After you complete the Leapp upgrade and Red Hat OpenStack Platform upgrade, the Ceph Storage containerized services will still use version 3 containers. At this point, you must upgrade **ceph-ansible** to version 4 and then run the **openstack overcloud external-upgrade run --tags ceph** command that performs an upgrade of all Red Hat Ceph Storage services on all nodes to version 4.

Summary of the Ceph Storage workflow

The following list is a high level workflow for the Red Hat Ceph Storage upgrade. This workflow is integrated into the general Red Hat OpenStack Platform workflow and you run upgrade framework commands on the undercloud to perform the operations in this workflow.

1. Upgrade the undercloud but retain version 3 of **ceph-ansible**
2. Start the overcloud upgrade
3. Perform the following tasks for each node that hosts Ceph Storage containerized services:
 - a. Migrate the Ceph Storage containerized services to Podman
 - b. Upgrade the operating system
 - c. Upgrade the OpenStack Platform services, which relaunches Ceph Storage version 3 containerized services
4. Complete the overcloud upgrade
5. Upgrade **ceph-ansible** to version 4 on the undercloud
6. Upgrade to Red Hat Ceph Storage 4 on the overcloud



NOTE

This list does not capture all steps in the complete Red Hat OpenStack Platform 16.1 upgrade process but focuses only on the aspects relevant to Red Hat Ceph Storage to describe what occurs to Ceph Storage services during the upgrade process.

12.2. CHECKING YOUR CEPH-ANSIBLE VERSION

During the undercloud upgrade, you retained the Ceph Storage 3 version of the **ceph-ansible** package. This helps maintain the compatibility of the Ceph Storage 3 containers on your Ceph Storage nodes. Verify that this package remains on your undercloud.

Procedure

1. Log in to the undercloud as the **stack** user.
2. Run the **dnf** command to check the version of the **ceph-ansible** package:

```
$ sudo dnf info ceph-ansible
```

The command output shows version 3 of the **ceph-ansible** package:

```
Installed Packages
Name      : ceph-ansible
Version   : 3.xx.xx.xx
...
```



IMPORTANT

If the **ceph-ansible** package is missing or not a version 3 package, download the latest version 3 package from the [Red Hat Package Browser](#) and manually install the package on your undercloud. Note that the **ceph-ansible** version 3 package is only available from Red Hat Enterprise Linux 7 repositories and is not available in Red Hat Enterprise Linux 8 repositories. **ceph-ansible** version 3 is not supported on Red Hat Enterprise Linux 8 outside the context of the Red Hat OpenStack Platform framework for upgrades.

12.3. SETTING THE CEPH-ANSIBLE REPOSITORY

The Red Hat OpenStack Platform 16.1 validation framework tests that **ceph-ansible** is installed correctly before director upgrades the overcloud to Red Hat Ceph Storage 4. The framework uses the **CephAnsibleRepo** parameter to check that you installed **ceph-ansible** from the correct repository. Director disables the test after you run the **openstack overcloud upgrade prepare** command and this test remains disabled through the duration of the Red Hat OpenStack Platform 16.1 overcloud upgrade. Director re-enables this test after running the **openstack overcloud upgrade converge** command. However, to prepare for this validation, you must set the **CephAnsibleRepo** parameter to the **Red Hat Ceph Storage Tools 4 for RHEL 8** repository.

Procedure

1. Log in to the undercloud as the **stack** user.
2. Edit the environment file that contains your overcloud Ceph Storage configuration. This file is usually named **ceph-config.yaml** and you can find it in your **templates** directory:

```
$ vi /home/stack/templates/ceph-config.yaml
```

3. Add the **CephAnsibleRepo** parameter to the **parameter_defaults** section:

```
parameter_defaults:
...
CephAnsibleRepo: rhceph-4-tools-for-rhel-8-x86_64-rpms
...
```

CephAnsibleRepo sets the repository that includes **ceph-ansible**. The validation framework uses this parameter to check that you have installed **ceph-ansible** on the undercloud.

4. Save the **ceph-config.yaml** file.

12.4. CHECKING CEPH CLUSTER STATUS BEFORE AN UPGRADE

Before you can proceed with the overcloud upgrade, you must verify that the Ceph cluster is functioning as expected.

Procedure

1. Log in to the node that is running the **ceph-mon** service. This node is usually a Controller node or a standalone Ceph Monitor node.
2. Enter the following command to view the status of the Ceph cluster:

```
$ docker exec ceph-mon-$HOSTNAME ceph -s
```

3. Confirm that the health status of the cluster is **HEALTH_OK** and that all of the OSDs are **up**.

CHAPTER 13. UPDATING NETWORK CONFIGURATION

You must complete some network configuration to prepare for the overcloud upgrade.

13.1. UPDATING NETWORK INTERFACE TEMPLATES

Red Hat OpenStack Platform includes a script to automatically add the missing parameters to your NIC template files.

Procedure

1. Log in to the undercloud as the **stack** user.
2. Source the **stackrc** file.

```
$ source ~/stackrc
```

3. On the undercloud, create a file called **update-nic-templates.sh** and include the following content in the file:

```
#!/bin/bash
STACK_NAME="overcloud"
ROLES_DATA="/usr/share/openstack-tripleo-heat-templates/roles_data.yaml"
NETWORK_DATA="/usr/share/openstack-tripleo-heat-templates/network_data.yaml"
NIC_CONFIG_LINES=$(openstack stack environment show $STACK_NAME | grep
"::Net::SoftwareConfig" | sed -E 's/ *OS::TripleO::// ; s/::Net::SoftwareConfig::// ; s/ http.*user-
files/ /')
echo "$NIC_CONFIG_LINES" | while read LINE; do
    ROLE=$(echo "$LINE" | awk '{print $1;}')
    NIC_CONFIG=$(echo "$LINE" | awk '{print $2;}')

    if [ -f "$NIC_CONFIG" ]; then
        echo "Updating template for $ROLE role."
        python3 /usr/share/openstack-tripleo-heat-templates/tools/merge-new-params-nic-
config-script.py \
            --tgt-dir /usr/share/openstack-tripleo-heat-templates \
            --roles-data $ROLES_DATA \
            --network-data $NETWORK_DATA \
            --role-name "$ROLE" \
            --discard-comments yes \
            --template "$NIC_CONFIG"
    else
        echo "No NIC template detected for $ROLE role. Skipping $ROLE role."
    fi
done
```

- If you use a custom overcloud name, Set the **STACK_NAME** variable to the name of your overcloud. The default name for an overcloud stack is **overcloud**.
- If you use a custom **roles_data** file, set the **ROLES_DATA** variable to the location of the custom file. If you use the default **roles_data** file, leave the variable as **/usr/share/openstack-tripleo-heat-templates/roles_data.yaml**

- If you use a custom **network_data** file, set the **NETWORK_DATA** variable to the location of the custom file. If you use the default **network_data** file, leave the variable as **/usr/share/openstack-tripleo-heat-templates/network_data.yaml**
- Run **/usr/share/openstack-tripleo-heat-templates/tools/merge-new-params-nic-config-script.py -h** to see a list of options to add to the script.

4. Add executable permissions to the script:

```
$ chmod +x update-nic-templates.sh
```

5. Run the script:

```
$ ./update-nic-templates.sh
```

The script saves a copy of each custom NIC template and updates each template with the missing parameters. The script also skips any roles that do not have a custom template:

```
No NIC template detected for BlockStorage role. Skipping BlockStorage role.
Updating template for CephStorage role.
The original template was saved as: /home/stack/templates/custom-nics/ceph-storage.yaml.20200903144835
The update template was saved as: /home/stack/templates/custom-nics/ceph-storage.yaml
Updating template for Compute role.
The original template was saved as: /home/stack/templates/custom-nics/compute.yaml.20200903144838
The update template was saved as: /home/stack/templates/custom-nics/compute.yaml
Updating template for Controller role.
The original template was saved as: /home/stack/templates/custom-nics/controller.yaml.20200903144841
The update template was saved as: /home/stack/templates/custom-nics/controller.yaml
No NIC template detected for ObjectStorage role. Skipping ObjectStorage role.
```

13.2. MAINTAINING OPEN VSWITCH COMPATIBILITY DURING THE UPGRADE

Red Hat OpenStack Platform 13 uses Open vSwitch (OVS) as the default ML2 back end for OpenStack Networking (neutron). Newer versions of Red Hat OpenStack Platform use Open Virtual Network (OVN), which expands upon OVS capabilities. However, to ensure a stable upgrade, you must maintain OVS functionality during the duration of the upgrade and then migrate to OVN after you complete the upgrade.

To maintain OVS compatibility during the upgrade, include the following environment file as part of your environment file collection:

- **/usr/share/openstack-tripleo-heat-templates/environments/services/neutron-ovs.yaml**

Treat this file as part of your deployment until you have completed the migration to OVN. Include the file with all overcloud upgrade and deployment commands:

- **openstack overcloud upgrade prepare**
- **openstack overcloud upgrade converge**
- **openstack overcloud deploy**

- **openstack overcloud update prepare**
- **openstack overcloud update converge**
- Any other command that uses environment files.

13.3. MAINTAINING COMPOSABLE NETWORK COMPATIBILITY DURING THE UPGRADE

The **network_data** file format for Red Hat OpenStack Platform 16.1 includes new sections that you can use to define additional subnets and routes within a network. However, if you use a custom **network_data** file, you can still use the **network_data** file format from Red Hat OpenStack Platform 13.

- When you upgrade from Red Hat OpenStack Platform 13 to 16.1, use the Red Hat OpenStack Platform 13 **network_data** file format during and after the upgrade. For more information about Red Hat OpenStack Platform 13 composable network syntax, see "[Custom composable networks](#)".
- When you create new overclouds on Red Hat OpenStack Platform 16.1, use the Red Hat OpenStack Platform 16.1 **network_data** file format. For more information about Red Hat OpenStack Platform 16.1 composable network syntax, see "[Custom composable networks](#)".

CHAPTER 14. PREPARING NETWORK FUNCTIONS VIRTUALIZATION (NFV)

If you use network functions virtualization (NFV), you must complete some preparation for the overcloud upgrade.

14.1. NETWORK FUNCTIONS VIRTUALIZATION (NFV) ENVIRONMENT FILES

In a typical NFV-based environment, you can enable services such as the following:

- Single-root input/output virtualization (SR-IOV)
- Data Plane Development Kit (DPDK)

You do not require any specific reconfiguration to these services to accommodate the upgrade to Red Hat OpenStack Platform 16.1. However, ensure that the environment files that enable your NFV functionality meet the following requirements:

- The default environment files to enable NFV features are located in the **environments/services** directory of the Red Hat OpenStack Platform 16.1 **openstack-tripleo-heat-templates** collection. If you include the default NFV environment files from **openstack-tripleo-heat-templates** with your Red Hat OpenStack Platform 13 deployment, verify the correct environment file location for the respective feature in Red Hat OpenStack Platform 16.1:
 - Open vSwitch (OVS) networking and SR-IOV: **/usr/share/openstack-tripleo-heat-templates/environments/services/neutron-ovs-sriov.yaml**
 - Open vSwitch (OVS) networking and DPDK: **/usr/share/openstack-tripleo-heat-templates/environments/services/neutron-ovs-dpdk.yaml**
- To maintain OVS compatibility during the upgrade from Red Hat OpenStack Platform 13 to Red Hat OpenStack Platform 16.1, you must include the **/usr/share/openstack-tripleo-heat-templates/environments/services/neutron-ovs.yaml** environment file. When running deployment and upgrade commands that involve environment files, you must include any NFV-related environment files after the **neutron-ovs.yaml** file. For example, when running **openstack overcloud upgrade prepare** with OVS and NFV environment files, include the files in the following order:
 - The OVS environment file
 - The SR-IOV environment file
 - The DPDK environment file

```
$ openstack overcloud upgrade prepare \
...
-e /usr/share/openstack-tripleo-heat-templates/environments/services/neutron-ovs.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/services/neutron-ovs-
sriov.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/services/neutron-ovs-
dpdk.yaml \
...
```

CHAPTER 15. FINAL REVIEW BEFORE UPGRADE

Complete a final check of all preparation steps before you begin the upgrade.

15.1. NEW ENVIRONMENT FILES TO INCLUDE WITH YOUR DEPLOYMENT

In addition to your regular overcloud environment files, you must include new environment files to facilitate the upgrade to Red Hat OpenStack Platform 16.1

File	Notes
/home/stack/templates/upgrades-environment.yaml	This file contains the parameters specific to the upgrade. This file is necessary only for the duration of the upgrade. Discard this file after you run openstack overcloud upgrade converge .
/home/stack/templates/rhsm.yaml	This file contains registration and subscription information for the overcloud. This file registers your systems either to the Red Hat Customer Portal or a Red Hat Satellite server.
/home/stack/containers-prepare-parameter.yaml	The file that contains the source and preparation steps. This is the same file that you use with the undercloud upgrade.
/usr/share/openstack-tripleo-heat-templates/environments/services/neutron-ovs.yaml	OpenStack Platform 16.1 uses Open Virtual Network (OVN) as the default networking backend. However, OpenStack Platform 13 used Open vSwitch (OVS). Include this file to retain OVS compatibility during the upgrade. The OpenStack Platform 16.1 documentation includes instructions on migrating from OVS to OVN after the upgrade.

Add these files to the end of your environment file listing when you run the following commands:

- **openstack overcloud upgrade prepare**
- **openstack overcloud upgrade converge**
- **openstack overcloud deploy**

15.2. ENVIRONMENT FILES TO REMOVE FROM YOUR DEPLOYMENT

Remove any environment files specific to your OpenStack Platform Red Hat OpenStack Platform 13:

- Red Hat OpenStack Platform 13 container image list
- Red Hat OpenStack Platform 13 Customer Portal or Satellite **rhel-registration** scripts

Remove these files from the list of environment files you include when you run the following commands:

- **openstack overcloud upgrade prepare**
- **openstack overcloud upgrade converge**
- **openstack overcloud deploy**

15.3. UPGRADE CHECKLIST

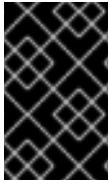
Use the following checklist to determine your readiness to upgrade the overcloud:

Item	Complete
Validated a working overcloud.	Y / N
Performed a Relax-and-Recover (ReaR) backup of the overcloud control plane.	Y / N
Implemented all preparation for Leapp, including: <ul style="list-style-type: none"> • Leapp environment file • Leapp data copied to the overcloud nodes • All overcloud nodes are now using predictable NIC names. • All overcloud nodes have PermitRootLogin set 	Y / N
Updated your registration details to Red Hat OpenStack Platform 16.1 repositories and converted your environment file to use the Ansible-based method.	Y / N
Updated your network configuration templates.	Y / N
Updated your environment file list with new environment files for Red Hat OpenStack Platform 16.1.	Y / N
Removed old environment files only relevant to Red Hat OpenStack Platform 13, such as old Red Hat registration and container image location files.	Y / N

PART III. UPGRADING THE OVERCLOUD

CHAPTER 16. UPGRADE COMMAND OVERVIEW

The upgrade process involves different commands that you run at certain stages of process.



IMPORTANT

This section only contains information about each command. You must run these commands in a specific order and provide options specific to your overcloud. Wait until you receive instructions to run these commands at the appropriate step.

16.1. OPENSTACK OVERCLOUD UPGRADE PREPARE

This command performs the initial preparation steps for the overcloud upgrade, which includes replacing the current overcloud plan on the undercloud with the new OpenStack Platform 16.1 overcloud plan and your updated environment files. This command functions similar to the **openstack overcloud deploy** command and uses many of the same options.

16.2. OPENSTACK OVERCLOUD UPGRADE RUN

This command performs the upgrade process. Director creates a set of Ansible playbooks based on the new OpenStack Platform 16.1 overcloud plan and runs the fast forward tasks on the entire overcloud. This includes running the upgrade process through each OpenStack Platform version from 13 to 16.1.

In addition to the standard upgrade process, this command can perform a Leapp upgrade of the operating system on overcloud nodes. Run these tasks using the **--tags** option.

Upgrade task tags for Leapp

system_upgrade

Task that combines tasks from **system_upgrade_prepare**, **system_upgrade_run**, and **system_upgrade_reboot**.

system_upgrade_prepare

Tasks to prepare for the operating system upgrade with Leapp.

system_upgrade_run

Tasks to run Leapp and upgrade the operating system.

system_upgrade_reboot

Tasks to reboot a system and complete the operating system upgrade.

Upgrade task tags for workload migration

nova_hybrid_state

Task that sets up temporary OpenStack Platform 16.1 containers on Compute nodes to facilitate workload migration during the upgrade.

16.3. OPENSTACK OVERCLOUD EXTERNAL-UPGRADE RUN

This command performs upgrade tasks outside the standard upgrade process. Director creates a set of Ansible playbooks based on the new OpenStack Platform 16.1 overcloud plan and you run specific tasks using the **--tags** option.

External task tags for container management

container_image_prepare

Tasks for pulling container images to the undercloud registry and preparing the images for the overcloud to use.

External task tags for Ceph Storage upgrades

ceph

Tasks to install Red Hat Ceph Storage using **ceph-ansible** playbooks.

ceph_systemd

Tasks to convert Red Hat Ceph Storage systemd unit files to use **podman** management.

External task tags for database transfer

system_upgrade_cleanup

Tasks to clean storage directories related to **system_upgrade_transfer_data** tasks.

system_upgrade_stop_services

Tasks to shut down all services.

system_upgrade_transfer_data

Tasks to shut down all services and perform a database transfer to the bootstrap node.

16.4. OPENSTACK OVERCLOUD UPGRADE CONVERGE

This command performs the final step in the overcloud upgrade. This final step synchronizes the overcloud heat stack with the OpenStack Platform 16.1 overcloud plan and your updated environment files. This process ensures that the resulting overcloud matches the configuration of a new OpenStack Platform 16.1 overcloud. This command is similar to the **openstack overcloud deploy** command and uses many of the same options.

16.5. OVERCLOUD NODE UPGRADE WORKFLOW

When you perform an upgrade on each overcloud node, you must consider the following aspects to determine the correct commands to run at the relevant stage in the upgrade:

Controller Services

- **Does the node contain Pacemaker services?** You must first upgrade the bootstrap node, which includes a database transfer and launching temporary containers to facilitate migration during the transition from Red Hat OpenStack 13 to 16.1. After upgrading the bootstrap node, upgrade each additional node with pacemaker services and ensure the each node joins the new Pacemaker cluster started with the bootstrap node. The process for upgrading split-service Controller nodes without Pacemaker does not require these additional steps.

Compute Services

- **Is the node a Compute node?** If the node does contain Compute services, you must migrate virtual machines from the node to ensure maximum availability. A Compute node in this situation includes any node designed to host virtual machines. This definition includes the follow Compute nodes types:

- Regular Compute nodes
- Compute nodes with Hyper-Converged Infrastructure (HCI)
- Compute nodes with Network Function Virtualization technologies such as Data Plane Development Kit (DPDK) or Single Root Input/Output Virtualization (SR-IOV)
- Real Time Compute nodes

Ceph Storage Services

- **Does the node contain any Ceph Storage services?** You must convert the **systemd** unit files for any containerized Ceph Storage services on the node to use **podman** instead of **docker**. This applies to the following node types:
 - Ceph Storage OSD nodes
 - Controller nodes with Ceph MON services
 - Split-Controller Ceph MON nodes
 - Compute nodes with Hyper-Converged Infrastructure (HCI)

Workflow

Use the following workflow diagram to identify the correct update path for specific nodes:



CHAPTER 17. UPGRADING A STANDARD OVERCLOUD

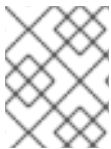
This scenario contains an example upgrade process for a standard overcloud environment, which includes the following node types:

- Three Controller nodes
- Three Ceph Storage nodes
- Multiple Compute nodes

17.1. RUNNING THE OVERCLOUD UPGRADE PREPARATION

The upgrade requires running **openstack overcloud upgrade prepare** command, which performs the following tasks:

- Updates the overcloud plan to OpenStack Platform 16.1
- Prepares the nodes for the upgrade



NOTE

If you are not using the default stack name (**overcloud**), set your stack name with the **--stack *STACK NAME*** option replacing ***STACK NAME*** with the name of your stack.

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

2. Run the upgrade preparation command:

```
$ openstack overcloud upgrade prepare \
  --stack STACK NAME \
  --templates \
  -e ENVIRONMENT FILE
...
-e /home/stack/templates/upgrades-environment.yaml \
-e /home/stack/templates/rhsm.yaml \
-e /home/stack/containers-prepare-parameter.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/services/neutron-ovs.yaml \
...
```

Include the following options relevant to your environment:

- The environment file (**upgrades-environment.yaml**) with the upgrade-specific parameters (**-e**).
- The environment file (**rhsm.yaml**) with the registration and subscription parameters (**-e**).
- The environment file (**containers-prepare-parameter.yaml**) with your new container image locations (**-e**). In most cases, this is the same environment file that the undercloud uses.
- The environment file (**neutron-ovs.yaml**) to maintain OVS compatibility.

- Any custom configuration environment files (**-e**) relevant to your deployment.
 - If applicable, your custom roles (**roles_data**) file using **--roles-file**.
 - If applicable, your composable network (**network_data**) file using **--networks-file**.
 - If you use a custom stack name, pass the name with the **--stack** option.
3. Wait until the upgrade preparation completes.
 4. Download the container images:

```
$ openstack overcloud external-upgrade run --stack STACK NAME --tags
container_image_prepare
```

17.2. UPGRADING CONTROLLER NODES

Upgrade all the Controller nodes to OpenStack Platform 16.1. This process involves upgrading each Controller node starting with the bootstrap Controller node.

In this example, the controller nodes are named using the default **overcloud-controller-NODEID** convention. This includes the following three controller nodes:

- **overcloud-controller-0**
- **overcloud-controller-1**
- **overcloud-controller-2**

Substitute these values for your own node names where applicable.



NOTE

If you are not using the **overcloud** default stack name, set your stack name with the **--stack *STACK NAME*** option replacing ***STACK NAME*** with the name of your stack.

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

2. Identify the bootstrap Controller node by running the following command on a Controller node:

```
$ ssh heat-admin@CONTROLLER_IP "sudo hiera -c /etc/puppet/hiera.yaml
pacemaker_short_bootstrap_node_name"
```

This example uses **overcloud-controller-0** as the value of the bootstrap node. Substitute this value for your own bootstrap node.

3. Upgrade the bootstrap Controller node:
 - a. If your overcloud includes a director-deployed Ceph Storage cluster, run the external upgrade command with the **ceph_systemd** tag:

```
$ openstack overcloud external-upgrade run --stack STACK NAME --tags ceph_systemd
-e ceph_ansible_limit=overcloud-controller-0
```

This command performs the following functions:

- Changes the systemd units that control the Ceph Storage containers to use Podman management.
- Limits actions to the selected Controller node using the **ceph_ansible_limit** variable.

This step is a preliminary measure to prepare the Ceph Storage services for The **leapp** upgrade.

- b. Run the upgrade command with the **system_upgrade** tag:

```
$ openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit
overcloud-controller-0
```

This command performs the following actions:

- Performs a Leapp upgrade of the operating system.
- Performs a reboot as a part of the Leapp upgrade.



IMPORTANT

The next command causes an outage on the control plane. You cannot perform any standard operations on the overcloud during the next few steps.

- c. Run the external upgrade command with the **system_upgrade_transfer_data** tag:

```
$ openstack overcloud external-upgrade run --stack STACK NAME --tags
system_upgrade_transfer_data
```

This command copies the latest version of the database from an existing node to the bootstrap node.

- d. Run the upgrade command with the **nova_hybrid_state** tag and run only the **upgrade_steps_playbook.yaml** playbook:

```
$ openstack overcloud upgrade run --stack STACK NAME --playbook
upgrade_steps_playbook.yaml --tags nova_hybrid_state --limit all
```

This command launches temporary 16.1 containers on Compute nodes to help facilitate workload migration when you upgrade Compute nodes at a later step.

- e. Run the upgrade command with no tags:

```
$ openstack overcloud upgrade run --stack STACK NAME --limit overcloud-controller-0
```

This command performs the Red Hat OpenStack Platform upgrade.



IMPORTANT

The control plane becomes active when this command finishes. You can perform standard operations on the overcloud again.

4. Upgrade the next Controller node:

- a. If your overcloud includes a director-deployed Ceph Storage cluster, run the external upgrade command with the **ceph_systemd** tag:

```
$ openstack overcloud external-upgrade run --stack STACK NAME --tags ceph_systemd
-e ceph_ansible_limit=overcloud-controller-1
```

This command performs the following functions:

- Changes the systemd units that control the Ceph Storage containers to use Podman management.
- Limits actions to the selected Controller node using the **ceph_ansible_limit** variable.

This step is a preliminary measure to prepare the Ceph Storage services for The **leapp** upgrade.

- b. Run the upgrade command with the **system_upgrade** tag on the next Controller node:

```
$ openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit
overcloud-controller-1
```

This command performs the following actions:

- Performs a Leapp upgrade of the operating system.
- Performs a reboot as a part of the Leapp upgrade.

- c. Run the upgrade command with no tags:

```
$ openstack overcloud upgrade run --stack STACK NAME --limit overcloud-controller-
0,overcloud-controller-1
```

This command performs the Red Hat OpenStack Platform upgrade. In addition to this node, include the previously upgraded bootstrap node in the **--limit** option.

5. Upgrade the final Controller node:

- a. If your overcloud includes a director-deployed Ceph Storage cluster, run the external upgrade command with the **ceph_systemd** tag:

```
$ openstack overcloud external-upgrade run --stack STACK NAME --tags ceph_systemd
-e ceph_ansible_limit=overcloud-controller-2
```

This command performs the following functions:

- Changes the systemd units that control the Ceph Storage containers to use Podman management.
- Limits actions to the selected Controller node using the **ceph_ansible_limit** variable.

This step is a preliminary measure to prepare the Ceph Storage services for The **leapp** upgrade.

- b. Run the upgrade command with the **system_upgrade** tag:

```
$ openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit overcloud-controller-2
```

This command performs the following actions:

- Performs a Leapp upgrade of the operating system.
- Performs a reboot as a part of the Leapp upgrade.

- c. Run the upgrade command with no tags:

```
$ openstack overcloud upgrade run --stack STACK NAME --limit overcloud-controller-0,overcloud-controller-1,overcloud-controller-2
```

This command performs the Red Hat OpenStack Platform upgrade. Include all Controller nodes in the **--limit** option.

17.3. UPGRADING THE OPERATING SYSTEM FOR CEPH STORAGE NODES

Upgrade the operating system for each Ceph Storage nodes.



NOTE

If you are not using the default stack name (**overcloud**), set your stack name with the **--stack *STACK NAME*** option replacing ***STACK NAME*** with the name of your stack.

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

2. Log in to a node with Ceph MON services:
3. Disable OSD exclusion and rebalancing:

```
$ sudo docker ps | grep ceph-mon
$ sudo docker exec -it CONTAINER ceph osd set noout
$ sudo docker exec -it CONTAINER ceph osd set norebalance
```

Replace *CONTAINER* with the name of the container running Ceph MON.

4. Log out of the node with Ceph MON services and return to the undercloud.
5. Select a Ceph Storage node and upgrade the operating system:
 - a. Run the external upgrade command with the **ceph_systemd** tag:

```
$ openstack overcloud external-upgrade run --stack STACK NAME --tags ceph_systemd
-e ceph_ansible_limit=overcloud-cephstorage-0
```

This command performs the following functions:

- Changes the systemd units that control the Ceph Storage containers to use Podman management.
- Limits actions to the selected node using the **ceph_ansible_limit** variable.

This step is a preliminary measure to prepare the Ceph Storage services for The **leapp** upgrade.

- b. Run the upgrade command with the **system_upgrade** tag:

```
$ openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit
overcloud-cephstorage-0
```

This command performs the following actions:

- Performs a Leapp upgrade of the operating system.
- Performs a reboot as a part of the Leapp upgrade.

- c. Run the upgrade command with no tags:

```
$ openstack overcloud upgrade run --stack STACK NAME --limit overcloud-cephstorage-
0
```

This command runs the **config-download** playbooks and configures the composable services on the Ceph Storage node. This step does not upgrade the Ceph Storage nodes to Red Hat Ceph Storage 4. The Red Hat Ceph Storage 4 upgrade occurs in a later procedure.

6. Select the next Ceph Storage node and upgrade the operating system:

- a. Run the external upgrade command with the **ceph_systemd** tag:

```
$ openstack overcloud external-upgrade run --stack STACK NAME --tags ceph_systemd
-e ceph_ansible_limit=overcloud-cephstorage-1
```

This command performs the following functions:

- Changes the systemd units that control the Ceph Storage containers to use Podman management.
- Limits actions to the selected node using the **ceph_ansible_limit** variable.

This step is a preliminary measure to prepare the Ceph Storage services for The **leapp** upgrade.

- b. Run the upgrade command with the **system_upgrade** tag:

```
$ openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit
overcloud-cephstorage-1
```

This command performs the following actions:

- Performs a Leapp upgrade of the operating system.
- Performs a reboot as a part of the Leapp upgrade.

c. Run the upgrade command with no tags:

```
$ openstack overcloud upgrade run --stack STACK NAME --limit overcloud-cephstorage-1
```

This command runs the **config-download** playbooks and configures the composable services on the Ceph Storage node. This step does not upgrade the Ceph Storage nodes to Red Hat Ceph Storage 4. The Red Hat Ceph Storage 4 upgrade occurs in a later procedure.

7. Select the final Ceph Storage node and upgrade the operating system:

a. Run the external upgrade command with the **ceph_systemd** tag:

```
$ openstack overcloud external-upgrade run --stack STACK NAME --tags ceph_systemd -e ceph_ansible_limit=overcloud-cephstorage-2
```

This command performs the following functions:

- Changes the systemd units that control the Ceph Storage containers to use Podman management.
- Limits actions to the selected node using the **ceph_ansible_limit** variable.

This step is a preliminary measure to prepare the Ceph Storage services for The **leapp** upgrade.

b. Run the upgrade command with the **system_upgrade** tag:

```
$ openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit overcloud-cephstorage-2
```

This command performs the following actions:

- Performs a Leapp upgrade of the operating system.
- Performs a reboot as a part of the Leapp upgrade.

c. Run the upgrade command with no tags:

```
$ openstack overcloud upgrade run --stack STACK NAME --limit overcloud-cephstorage-2
```

This command runs the **config-download** playbooks and configures the composable services on the Ceph Storage node. This step does not upgrade the Ceph Storage nodes to Red Hat Ceph Storage 4. The Red Hat Ceph Storage 4 upgrade occurs in a later procedure.

8. After you upgrade all HCI nodes, log into a node that hosts Ceph MON services.

9. Enable OSD exclusion and rebalancing:

```
$ sudo podman ps | grep ceph-mon
$ sudo podman exec -it CONTAINER ceph osd unset noout
$ sudo podman exec -it CONTAINER ceph osd unset norebalance
```

Replace *CONTAINER* with the name of the container running Ceph MON.

10. Log out of the node with Ceph MON services and return to the undercloud.

17.4. UPGRADING COMPUTE NODES

Upgrade all the Compute nodes to OpenStack Platform 16.1.



NOTE

If you are not using the default stack name (**overcloud**), set your stack name with the **--stack *STACK NAME*** option replacing ***STACK NAME*** with the name of your stack.

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

2. Migrate your instances. For more information on migration strategies, see ["Migrating virtual machines between Compute nodes"](#)
3. Run the upgrade command with the **system_upgrade** tag:

```
$ openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit
overcloud-compute-0
```

This command performs the following actions:

- Performs a Leapp upgrade of the operating system.
- Performs a reboot as a part of the Leapp upgrade.

4. Run the upgrade command with no tags:

```
$ openstack overcloud upgrade run --stack STACK NAME --limit overcloud-compute-0
```

This command performs the Red Hat OpenStack Platform upgrade.

5. To upgrade multiple Compute nodes in parallel, set the **--limit** option to a comma-separated list of nodes that you want to upgrade. First perform the **system_upgrade** task:

```
$ openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit
overcloud-compute-0,overcloud-compute-1,overcloud-compute-2
```

Then perform the standard OpenStack service upgrade:

```
$ openstack overcloud upgrade run --stack STACK NAME --limit overcloud-compute-
0,overcloud-compute-1,overcloud-compute-2
```

17.5. SYNCHRONIZING THE OVERCLOUD STACK

The upgrade requires an update the overcloud stack to ensure that the stack resource structure and parameters align with a fresh deployment of OpenStack Platform 16.1.



NOTE

If you are not using the default stack name (**overcloud**), set your stack name with the **--stack *STACK NAME*** option replacing ***STACK NAME*** with the name of your stack.

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

2. Edit the **containers-prepare-parameter.yaml** file and remove the following parameters and their values:

- **ceph3_namespace**
- **ceph3_tag**
- **ceph3_image**
- **name_prefix_stein**
- **name_suffix_stein**
- **namespace_stein**
- **tag_stein**

3. Run the upgrade finalization command:

```
$ openstack overcloud upgrade converge \
  --stack STACK NAME \
  --templates \
  -e ENVIRONMENT FILE
...
-e /home/stack/templates/upgrades-environment.yaml \
-e /home/stack/templates/rhsm.yaml \
-e /home/stack/containers-prepare-parameter.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/services/neutron-ovs.yaml \
...
```

Include the following options relevant to your environment:

- The environment file (**upgrades-environment.yaml**) with the upgrade-specific parameters (**-e**).
- The environment file (**rhsm.yaml**) with the registration and subscription parameters (**-e**).
- The environment file (**containers-prepare-parameter.yaml**) with your new container image locations (**-e**). In most cases, this is the same environment file that the undercloud uses.

- The environment file (**neutron-ovs.yaml**) to maintain OVS compatibility.
 - Any custom configuration environment files (**-e**) relevant to your deployment.
 - If applicable, your custom roles (**roles_data**) file using **--roles-file**.
 - If applicable, your composable network (**network_data**) file using **--networks-file**.
 - If you use a custom stack name, pass the name with the **--stack** option.
4. Wait until the stack synchronization completes.



IMPORTANT

You do not need the **upgrades-environment.yaml** file for any further deployment operations.

CHAPTER 18. UPGRADING A SPLIT CONTROLLER OVERCLOUD

This scenario contains an example upgrade process for an overcloud with Controller node services split on to multiple nodes. This includes the following node types:

- Multiple split high availability services using Pacemaker
- Multiple split Controller services
- Three Ceph MON nodes
- Three Ceph Storage nodes
- Multiple Compute nodes

18.1. RUNNING THE OVERCLOUD UPGRADE PREPARATION

The upgrade requires running **openstack overcloud upgrade prepare** command, which performs the following tasks:

- Updates the overcloud plan to OpenStack Platform 16.1
- Prepares the nodes for the upgrade



NOTE

If you are not using the default stack name (**overcloud**), set your stack name with the **--stack *STACK NAME*** option replacing ***STACK NAME*** with the name of your stack.

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

2. Run the upgrade preparation command:

```
$ openstack overcloud upgrade prepare \
  --stack STACK NAME \
  --templates \
  -e ENVIRONMENT FILE \
  ...
  -e /home/stack/templates/upgrades-environment.yaml \
  -e /home/stack/templates/rhsm.yaml \
  -e /home/stack/containers-prepare-parameter.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/services/neutron-ovs.yaml \
  ...
```

Include the following options relevant to your environment:

- The environment file (**upgrades-environment.yaml**) with the upgrade-specific parameters (**-e**).

- The environment file (**rhsm.yaml**) with the registration and subscription parameters (**-e**).
 - The environment file (**containers-prepare-parameter.yaml**) with your new container image locations (**-e**). In most cases, this is the same environment file that the undercloud uses.
 - The environment file (**neutron-ovs.yaml**) to maintain OVS compatibility.
 - Any custom configuration environment files (**-e**) relevant to your deployment.
 - If applicable, your custom roles (**roles_data**) file using **--roles-file**.
 - If applicable, your composable network (**network_data**) file using **--networks-file**.
 - If you use a custom stack name, pass the name with the **--stack** option.
3. Wait until the upgrade preparation completes.
 4. Download the container images:

```
$ openstack overcloud external-upgrade run --stack STACK NAME --tags
container_image_prepare
```

18.2. UPGRADING PACEMAKER-BASED NODES

Upgrade all nodes that host Pacemaker services to OpenStack Platform 16.1. The following roles include Pacemaker-based services:

- Controller
- Database (MySQL, Galera)
- Messaging (RabbitMQ)
- Load Balancing (HAProxy)
- Any other role that contains the following services:
 - **OS::TripleO::Services::Pacemaker**
 - **OS::TripleO::Services::PacemakerRemote**

This process involves upgrading each node starting with the bootstrap node.



NOTE

If you are not using the default stack name (**overcloud**), set your stack name with the **--stack *STACK NAME*** option replacing ***STACK NAME*** with the name of your stack.

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

2. Identify the bootstrap node by running the following command on a Controller-based node:

```
-
```

```
$ ssh heat-admin@CONTROLLER_IP "sudo hiera -c /etc/puppet/hiera.yaml
pacemaker_short_bootstrap_node_name"
```

3. Upgrade the bootstrap node:

- a. If the node any contains Ceph Storage containers, run the external upgrade command with the **ceph_systemd** tag:

```
$ openstack overcloud external-upgrade run --stack STACK_NAME --tags ceph_systemd
-e ceph_ansible_limit=overcloud-controller-0
```

This command performs the following functions:

- Changes the systemd units that control the Ceph Storage containers to use Podman management.
- Limits actions to the selected node using the **ceph_ansible_limit** variable.

This step is a preliminary measure to prepare the Ceph Storage services for The **leapp** upgrade.

- b. Run the upgrade command with the **system_upgrade** tag:

```
$ openstack overcloud upgrade run --stack STACK_NAME --tags system_upgrade --limit
overcloud-controller-0
```

This command performs the following actions:

- Performs a Leapp upgrade of the operating system.
- Performs a reboot as a part of the Leapp upgrade.

- c. Run the external upgrade command with the **system_upgrade_transfer_data** tag:

```
$ openstack overcloud external-upgrade run --stack STACK_NAME --tags
system_upgrade_transfer_data
```

This command copies the latest version of the database from an existing node to the bootstrap node.

- d. Run the upgrade command with the **nova_hybrid_state** tag and run only the **upgrade_steps_playbook.yaml** playbook:

```
$ openstack overcloud upgrade run --stack STACK_NAME --playbook
upgrade_steps_playbook.yaml --tags nova_hybrid_state --limit all
```

This command launches temporary 16.1 containers on Compute nodes to help facilitate workload migration when you upgrade Compute nodes at a later step.

- e. Run the upgrade command with no tags:

```
$ openstack overcloud upgrade run --stack STACK_NAME --limit overcloud-controller-0
```

This command performs the Red Hat OpenStack Platform upgrade.

4. Upgrade each Pacemaker-based node:

- a. If the node any contains Ceph Storage containers, run the external upgrade command with the **ceph_systemd** tag:

```
$ openstack overcloud external-upgrade run --stack STACK NAME --tags ceph_systemd
-e ceph_ansible_limit=overcloud-database-0
```

This command performs the following functions:

- Changes the systemd units that control the Ceph Storage containers to use Podman management.
- Limits actions to the selected node using the **ceph_ansible_limit** variable.

This step is a preliminary measure to prepare the Ceph Storage services for The **leapp** upgrade.

- b. Run the upgrade command with the **system_upgrade** tag on the next node:

```
$ openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit
overcloud-database-0
```

This command performs the following actions:

- Performs a Leapp upgrade of the operating system.
- Performs a reboot as a part of the Leapp upgrade.

- c. Run the upgrade command with no tags:

```
$ openstack overcloud upgrade run --stack STACK NAME --limit overcloud-controller-
0,overcloud-database-0
```

This command performs the Red Hat OpenStack Platform upgrade. In addition to this node, include any previously upgraded node in the **--limit** option.

5. Repeat the upgrade process on each Pacemaker-based node until you have upgraded all Pacemaker-based node.

18.3. UPGRADING NON-PACEMAKER CONTROLLER NODES

Upgrade all nodes without Pacemaker-based services to OpenStack Platform 16.1. These nodes usually contain a specific OpenStack service. Examples of roles without Pacemaker-based services include the following:

- Networker
- Ironic Conductor
- Object Storage
- Any custom roles with services split or scaled from standard Controller nodes

Do not include the following nodes in this grouping:

- Any Compute nodes
- Any Ceph Storage nodes

This process involves upgrading each node.



NOTE

If you are not using the default stack name (**overcloud**), set your stack name with the **--stack *STACK NAME*** option replacing ***STACK NAME*** with the name of your stack.

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

2. Run the upgrade command with the **system_upgrade** tag:

```
$ openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit overcloud-networker-0
```

This command performs the following actions:

- Performs a Leapp upgrade of the operating system.
- Performs a reboot as a part of the Leapp upgrade.

3. Run the upgrade command with no tags:

```
$ openstack overcloud upgrade run --stack STACK NAME --limit overcloud-networker-0
```

This command performs the Red Hat OpenStack Platform upgrade.

4. Repeat the upgrade process on each node until you have upgraded all Controller-based node.

18.4. UPGRADING THE OPERATING SYSTEM FOR CEPH MON NODES

Upgrade the operating system for each Ceph MON node. It is recommended to upgrade each Ceph MON node individually to maintain a quorum among the nodes.



NOTE

If you are not using the default stack name (**overcloud**), set your stack name with the **--stack *STACK NAME*** option replacing ***STACK NAME*** with the name of your stack.

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

2. Select a Ceph MON node and upgrade the operating system:

- a. Run the external upgrade command with the **ceph_systemd** tag:

```
$ openstack overcloud external-upgrade run --stack STACK NAME --tags ceph_systemd
-e ceph_ansible_limit=overcloud-cephmon-0
```

This command performs the following functions:

- Changes the systemd units that control the Ceph Storage containers to use Podman management.
- Limits actions to the selected node using the **ceph_ansible_limit** variable.

This step is a preliminary measure to prepare the Ceph Storage services for The **leapp** upgrade.

- b. Run the upgrade command with the **system_upgrade** tag:

```
$ openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit
overcloud-cephmon-0
```

This command performs the following actions:

- Performs a Leapp upgrade of the operating system.
- Performs a reboot as a part of the Leapp upgrade.

- c. Run the upgrade command with no tags:

```
$ openstack overcloud upgrade run --stack STACK NAME --limit overcloud-cephmon-0
```

This command runs the **config-download** playbooks and configures the composable services on the Ceph MON node. This step does not upgrade the Ceph MON nodes to Red Hat Ceph Storage 4. The Red Hat Ceph Storage 4 upgrade occurs in a later procedure.

3. Select the next Ceph MON node and upgrade the operating system:

- a. Run the external upgrade command with the **ceph_systemd** tag:

```
$ openstack overcloud external-upgrade run --stack STACK NAME --tags ceph_systemd
-e ceph_ansible_limit=overcloud-cephmon-1
```

This command performs the following functions:

- Changes the systemd units that control the Ceph Storage containers to use Podman management.
- Limits actions to the selected node using the **ceph_ansible_limit** variable.

This step is a preliminary measure to prepare the Ceph Storage services for The **leapp** upgrade.

- b. Run the upgrade command with the **system_upgrade** tag:

```
$ openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit
overcloud-cephmon-1
```

This command performs the following actions:

- Performs a Leapp upgrade of the operating system.
- Performs a reboot as a part of the Leapp upgrade.

c. Run the upgrade command with no tags:

```
$ openstack overcloud upgrade run --stack STACK NAME --limit overcloud-cephmon-1
```

This command runs the **config-download** playbooks and configures the composable services on the Ceph MON node. This step does not upgrade the Ceph MON nodes to Red Hat Ceph Storage 4. The Red Hat Ceph Storage 4 upgrade occurs in a later procedure.

4. Select the final Ceph MON node and upgrade the operating system:

a. Run the external upgrade command with the **ceph_systemd** tag:

```
$ openstack overcloud external-upgrade run --stack STACK NAME --tags ceph_systemd  
-e ceph_ansible_limit=overcloud-cephmon-2
```

This command performs the following functions:

- Changes the systemd units that control the Ceph Storage containers to use Podman management.
- Limits actions to the selected node using the **ceph_ansible_limit** variable.

This step is a preliminary measure to prepare the Ceph Storage services for The **leapp** upgrade.

b. Run the upgrade command with the **system_upgrade** tag:

```
$ openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit  
overcloud-cephmon-2
```

This command performs the following actions:

- Performs a Leapp upgrade of the operating system.
- Performs a reboot as a part of the Leapp upgrade.

c. Run the upgrade command with no tags:

```
$ openstack overcloud upgrade run --stack STACK NAME --limit overcloud-cephmon-2
```

This command runs the **config-download** playbooks and configures the composable services on the Ceph MON node. This step does not upgrade the Ceph MON nodes to Red Hat Ceph Storage 4. The Red Hat Ceph Storage 4 upgrade occurs in a later procedure.

18.5. UPGRADING THE OPERATING SYSTEM FOR CEPH STORAGE NODES

Upgrade the operating system for each Ceph Storage nodes.

**NOTE**

If you are not using the default stack name (**overcloud**), set your stack name with the **--stack *STACK NAME*** option replacing ***STACK NAME*** with the name of your stack.

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

2. Log in to a node with Ceph MON services:
3. Disable OSD exclusion and rebalancing:

```
$ sudo docker ps | grep ceph-mon
$ sudo docker exec -it CONTAINER ceph osd set noout
$ sudo docker exec -it CONTAINER ceph osd set norebalance
```

Replace *CONTAINER* with the name of the container running Ceph MON.

4. Log out of the node with Ceph MON services and return to the undercloud.
5. Select a Ceph Storage node and upgrade the operating system:

- a. Run the external upgrade command with the **ceph_systemd** tag:

```
$ openstack overcloud external-upgrade run --stack STACK NAME --tags ceph_systemd
-e ceph_ansiible_limit=overcloud-cephstorage-0
```

This command performs the following functions:

- Changes the systemd units that control the Ceph Storage containers to use Podman management.
- Limits actions to the selected node using the **ceph_ansiible_limit** variable.

This step is a preliminary measure to prepare the Ceph Storage services for The **leapp** upgrade.

- b. Run the upgrade command with the **system_upgrade** tag:

```
$ openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit
overcloud-cephstorage-0
```

This command performs the following actions:

- Performs a Leapp upgrade of the operating system.
- Performs a reboot as a part of the Leapp upgrade.

- c. Run the upgrade command with no tags:

```
$ openstack overcloud upgrade run --stack STACK NAME --limit overcloud-cephstorage-
0
```

This command runs the **config-download** playbooks and configures the composable services on the Ceph Storage node. This step does not upgrade the Ceph Storage nodes to Red Hat Ceph Storage 4. The Red Hat Ceph Storage 4 upgrade occurs in a later procedure.

6. Select the next Ceph Storage node and upgrade the operating system:

- a. Run the external upgrade command with the **ceph_systemd** tag:

```
$ openstack overcloud external-upgrade run --stack STACK NAME --tags ceph_systemd
-e ceph_ansible_limit=overcloud-cephstorage-1
```

This command performs the following functions:

- Changes the systemd units that control the Ceph Storage containers to use Podman management.
- Limits actions to the selected node using the **ceph_ansible_limit** variable.

This step is a preliminary measure to prepare the Ceph Storage services for The **leapp** upgrade.

- b. Run the upgrade command with the **system_upgrade** tag:

```
$ openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit
overcloud-cephstorage-1
```

This command performs the following actions:

- Performs a Leapp upgrade of the operating system.
- Performs a reboot as a part of the Leapp upgrade.

- c. Run the upgrade command with no tags:

```
$ openstack overcloud upgrade run --stack STACK NAME --limit overcloud-cephstorage-1
```

This command runs the **config-download** playbooks and configures the composable services on the Ceph Storage node. This step does not upgrade the Ceph Storage nodes to Red Hat Ceph Storage 4. The Red Hat Ceph Storage 4 upgrade occurs in a later procedure.

7. Select the final Ceph Storage node and upgrade the operating system:

- a. Run the external upgrade command with the **ceph_systemd** tag:

```
$ openstack overcloud external-upgrade run --stack STACK NAME --tags ceph_systemd
-e ceph_ansible_limit=overcloud-cephstorage-2
```

This command performs the following functions:

- Changes the systemd units that control the Ceph Storage containers to use Podman management.
- Limits actions to the selected node using the **ceph_ansible_limit** variable.

This step is a preliminary measure to prepare the Ceph Storage services for The **leapp** upgrade.

- b. Run the upgrade command with the **system_upgrade** tag:

```
$ openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit overcloud-cephstorage-2
```

This command performs the following actions:

- Performs a Leapp upgrade of the operating system.
- Performs a reboot as a part of the Leapp upgrade.

- c. Run the upgrade command with no tags:

```
$ openstack overcloud upgrade run --stack STACK NAME --limit overcloud-cephstorage-2
```

This command runs the **config-download** playbooks and configures the composable services on the Ceph Storage node. This step does not upgrade the Ceph Storage nodes to Red Hat Ceph Storage 4. The Red Hat Ceph Storage 4 upgrade occurs in a later procedure.

8. After you upgrade all HCI nodes, log into a node that hosts Ceph MON services.
9. Enable OSD exclusion and rebalancing:

```
$ sudo podman ps | grep ceph-mon
$ sudo podman exec -it CONTAINER ceph osd unset noout
$ sudo podman exec -it CONTAINER ceph osd unset norebalance
```

Replace *CONTAINER* with the name of the container running Ceph MON.

10. Log out of the node with Ceph MON services and return to the undercloud.

18.6. UPGRADING COMPUTE NODES

Upgrade all the Compute nodes to OpenStack Platform 16.1.



NOTE

If you are not using the default stack name (**overcloud**), set your stack name with the **--stack *STACK NAME*** option replacing ***STACK NAME*** with the name of your stack.

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

2. Migrate your instances. For more information on migration strategies, see ["Migrating virtual machines between Compute nodes"](#)
3. Run the upgrade command with the **system_upgrade** tag:

■

```
$ openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit
overcloud-compute-0
```

This command performs the following actions:

- Performs a Leapp upgrade of the operating system.
- Performs a reboot as a part of the Leapp upgrade.

4. Run the upgrade command with no tags:

```
$ openstack overcloud upgrade run --stack STACK NAME --limit overcloud-compute-0
```

This command performs the Red Hat OpenStack Platform upgrade.

5. To upgrade multiple Compute nodes in parallel, set the **--limit** option to a comma-separated list of nodes that you want to upgrade. First perform the **system_upgrade** task:

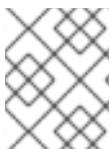
```
$ openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit
overcloud-compute-0,overcloud-compute-1,overcloud-compute-2
```

Then perform the standard OpenStack service upgrade:

```
$ openstack overcloud upgrade run --stack STACK NAME --limit overcloud-compute-
0,overcloud-compute-1,overcloud-compute-2
```

18.7. SYNCHRONIZING THE OVERCLOUD STACK

The upgrade requires an update the overcloud stack to ensure that the stack resource structure and parameters align with a fresh deployment of OpenStack Platform 16.1.



NOTE

If you are not using the default stack name (**overcloud**), set your stack name with the **--stack *STACK NAME*** option replacing ***STACK NAME*** with the name of your stack.

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

2. Edit the **containers-prepare-parameter.yaml** file and remove the following parameters and their values:

- **ceph3_namespace**
- **ceph3_tag**
- **ceph3_image**
- **name_prefix_stein**
- **name_suffix_stein**

- **namespace_stein**
- **tag_stein**

3. Run the upgrade finalization command:

```
$ openstack overcloud upgrade converge \
  --stack STACK NAME \
  --templates \
  -e ENVIRONMENT FILE \
  ...
  -e /home/stack/templates/upgrades-environment.yaml \
  -e /home/stack/templates/rhsm.yaml \
  -e /home/stack/containers-prepare-parameter.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/services/neutron-ovs.yaml \
  ...
```

Include the following options relevant to your environment:

- The environment file (**upgrades-environment.yaml**) with the upgrade-specific parameters (**-e**).
- The environment file (**rhsm.yaml**) with the registration and subscription parameters (**-e**).
- The environment file (**containers-prepare-parameter.yaml**) with your new container image locations (**-e**). In most cases, this is the same environment file that the undercloud uses.
- The environment file (**neutron-ovs.yaml**) to maintain OVS compatibility.
- Any custom configuration environment files (**-e**) relevant to your deployment.
- If applicable, your custom roles (**roles_data**) file using **--roles-file**.
- If applicable, your composable network (**network_data**) file using **--networks-file**.
- If you use a custom stack name, pass the name with the **--stack** option.

4. Wait until the stack synchronization completes.



IMPORTANT

You do not need the **upgrades-environment.yaml** file for any further deployment operations.

CHAPTER 19. UPGRADING AN OVERCLOUD WITH HYPER-CONVERGED INFRASTRUCTURE

This scenario contains an example upgrade process for an overcloud with hyper-converged infrastructure (HCI), which includes the following node types:

- Three Controller nodes
- Multiple HCI Compute nodes, which contain combined Compute and Ceph OSD services

19.1. RUNNING THE OVERCLOUD UPGRADE PREPARATION

The upgrade requires running **openstack overcloud upgrade prepare** command, which performs the following tasks:

- Updates the overcloud plan to OpenStack Platform 16.1
- Prepares the nodes for the upgrade



NOTE

If you are not using the default stack name (**overcloud**), set your stack name with the **--stack *STACK NAME*** option replacing ***STACK NAME*** with the name of your stack.

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

2. Run the upgrade preparation command:

```
$ openstack overcloud upgrade prepare \
  --stack STACK NAME \
  --templates \
  -e ENVIRONMENT FILE
...
-e /home/stack/templates/upgrades-environment.yaml \
-e /home/stack/templates/rhsm.yaml \
-e /home/stack/containers-prepare-parameter.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/services/neutron-ovs.yaml \
...
```

Include the following options relevant to your environment:

- The environment file (**upgrades-environment.yaml**) with the upgrade-specific parameters (**-e**).
- The environment file (**rhsm.yaml**) with the registration and subscription parameters (**-e**).
- The environment file (**containers-prepare-parameter.yaml**) with your new container image locations (**-e**). In most cases, this is the same environment file that the undercloud uses.
- The environment file (**neutron-ovs.yaml**) to maintain OVS compatibility.

- Any custom configuration environment files (**-e**) relevant to your deployment.
 - If applicable, your custom roles (**roles_data**) file using **--roles-file**.
 - If applicable, your composable network (**network_data**) file using **--networks-file**.
 - If you use a custom stack name, pass the name with the **--stack** option.
3. Wait until the upgrade preparation completes.
 4. Download the container images:

```
$ openstack overcloud external-upgrade run --stack STACK NAME --tags
container_image_prepare
```

19.2. UPGRADING CONTROLLER NODES

Upgrade all the Controller nodes to OpenStack Platform 16.1. This process involves upgrading each Controller node starting with the bootstrap Controller node.

In this example, the controller nodes are named using the default **overcloud-controller-NODEID** convention. This includes the following three controller nodes:

- **overcloud-controller-0**
- **overcloud-controller-1**
- **overcloud-controller-2**

Substitute these values for your own node names where applicable.



NOTE

If you are not using the **overcloud** default stack name, set your stack name with the **--stack *STACK NAME*** option replacing ***STACK NAME*** with the name of your stack.

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

2. Identify the bootstrap Controller node by running the following command on a Controller node:

```
$ ssh heat-admin@CONTROLLER_IP "sudo hiera -c /etc/puppet/hiera.yaml
pacemaker_short_bootstrap_node_name"
```

This example uses **overcloud-controller-0** as the value of the bootstrap node. Substitute this value for your own bootstrap node.

3. Upgrade the bootstrap Controller node:
 - a. If your overcloud includes a director-deployed Ceph Storage cluster, run the external upgrade command with the **ceph_systemd** tag:

```
$ openstack overcloud external-upgrade run --stack STACK NAME --tags ceph_systemd
-e ceph_ansible_limit=overcloud-controller-0
```

This command performs the following functions:

- Changes the systemd units that control the Ceph Storage containers to use Podman management.
- Limits actions to the selected Controller node using the **ceph_ansible_limit** variable.

This step is a preliminary measure to prepare the Ceph Storage services for The **leapp** upgrade.

- b. Run the upgrade command with the **system_upgrade** tag:

```
$ openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit
overcloud-controller-0
```

This command performs the following actions:

- Performs a Leapp upgrade of the operating system.
- Performs a reboot as a part of the Leapp upgrade.



IMPORTANT

The next command causes an outage on the control plane. You cannot perform any standard operations on the overcloud during the next few steps.

- c. Run the external upgrade command with the **system_upgrade_transfer_data** tag:

```
$ openstack overcloud external-upgrade run --stack STACK NAME --tags
system_upgrade_transfer_data
```

This command copies the latest version of the database from an existing node to the bootstrap node.

- d. Run the upgrade command with the **nova_hybrid_state** tag and run only the **upgrade_steps_playbook.yaml** playbook:

```
$ openstack overcloud upgrade run --stack STACK NAME --playbook
upgrade_steps_playbook.yaml --tags nova_hybrid_state --limit all
```

This command launches temporary 16.1 containers on Compute nodes to help facilitate workload migration when you upgrade Compute nodes at a later step.

- e. Run the upgrade command with no tags:

```
$ openstack overcloud upgrade run --stack STACK NAME --limit overcloud-controller-0
```

This command performs the Red Hat OpenStack Platform upgrade.



IMPORTANT

The control plane becomes active when this command finishes. You can perform standard operations on the overcloud again.

4. Upgrade the next Controller node:

- a. If your overcloud includes a director-deployed Ceph Storage cluster, run the external upgrade command with the **ceph_systemd** tag:

```
$ openstack overcloud external-upgrade run --stack STACK NAME --tags ceph_systemd
-e ceph_ansible_limit=overcloud-controller-1
```

This command performs the following functions:

- Changes the systemd units that control the Ceph Storage containers to use Podman management.
- Limits actions to the selected Controller node using the **ceph_ansible_limit** variable.

This step is a preliminary measure to prepare the Ceph Storage services for The **leapp** upgrade.

- b. Run the upgrade command with the **system_upgrade** tag on the next Controller node:

```
$ openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit
overcloud-controller-1
```

This command performs the following actions:

- Performs a Leapp upgrade of the operating system.
- Performs a reboot as a part of the Leapp upgrade.

- c. Run the upgrade command with no tags:

```
$ openstack overcloud upgrade run --stack STACK NAME --limit overcloud-controller-
0,overcloud-controller-1
```

This command performs the Red Hat OpenStack Platform upgrade. In addition to this node, include the previously upgraded bootstrap node in the **--limit** option.

5. Upgrade the final Controller node:

- a. If your overcloud includes a director-deployed Ceph Storage cluster, run the external upgrade command with the **ceph_systemd** tag:

```
$ openstack overcloud external-upgrade run --stack STACK NAME --tags ceph_systemd
-e ceph_ansible_limit=overcloud-controller-2
```

This command performs the following functions:

- Changes the systemd units that control the Ceph Storage containers to use Podman management.
- Limits actions to the selected Controller node using the **ceph_ansible_limit** variable.

This step is a preliminary measure to prepare the Ceph Storage services for The **leapp** upgrade.

- b. Run the upgrade command with the **system_upgrade** tag:

```
$ openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit overcloud-controller-2
```

This command performs the following actions:

- Performs a Leapp upgrade of the operating system.
- Performs a reboot as a part of the Leapp upgrade.

- c. Run the upgrade command with no tags:

```
$ openstack overcloud upgrade run --stack STACK NAME --limit overcloud-controller-0,overcloud-controller-1,overcloud-controller-2
```

This command performs the Red Hat OpenStack Platform upgrade. Include all Controller nodes in the **--limit** option.

19.3. UPGRADING COMPUTE NODES WITH HYPER-CONVERGED INFRASTRUCTURE (HCI)

Upgrade HCI Compute nodes to OpenStack Platform 16.1.



NOTE

If you are not using the default stack name (**overcloud**), set your stack name with the **--stack *STACK NAME*** option replacing ***STACK NAME*** with the name of your stack.

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

2. Migrate your instances. For more information on migration strategies, see ["Migrating virtual machines between Compute nodes"](#)
3. Log in to a node with Ceph MON services:
4. Disable OSD exclusion and rebalancing:

```
$ sudo docker ps | grep ceph-mon
$ sudo docker exec -it CONTAINER ceph osd set noout
$ sudo docker exec -it CONTAINER ceph osd set norebalance
```

Replace *CONTAINER* with the name of the container that runs Ceph MON.

5. Log out of the node with Ceph MON services and return to the undercloud.
6. Run the external upgrade command with the **ceph_systemd** tag:

```
$ openstack overcloud external-upgrade run --stack STACK NAME --tags ceph_systemd -e
ceph_ansible_limit=overcloud-computehci-0
```

This command performs the following functions:

- Changes the systemd units that control the Ceph Storage containers to use Podman management.
- Limits actions to the selected Ceph Storage node using the **ceph_ansible_limit** variable.

This step is a preliminary measure to prepare the Ceph Storage services for the **leapp** upgrade.

7. Run the upgrade command with the **system_upgrade** tag:

```
$ openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit
overcloud-computehci-0
```

This command performs the following actions:

- Performs a Leapp upgrade of the operating system.
- Performs a reboot as a part of the Leapp upgrade.

8. Run the upgrade command with no tags:

```
$ openstack overcloud upgrade run --stack STACK NAME --limit overcloud-computehci-0
```

This command performs the Red Hat OpenStack Platform upgrade.

9. To upgrade multiple Compute nodes in parallel, set the **--limit** option to a comma-separated list of nodes that you want to upgrade. First run the external upgrade command with the **ceph_systemd** tag:

```
$ openstack overcloud external-upgrade run --stack STACK NAME --tags ceph_systemd -e
ceph_ansible_limit=overcloud-computehci-0
```

Then perform the **system_upgrade** task:

```
$ openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit
overcloud-computehci-0,overcloud-computehci-1,overcloud-computehci-2
```

Then perform the standard OpenStack service upgrade:

```
$ openstack overcloud upgrade run --stack STACK NAME --limit overcloud-compute-
0,overcloud-computehci-1,overcloud-computehci-2
```

10. After you have upgraded all HCI nodes, log in to a node with Ceph MON services.
11. Enable OSD exclusion and rebalancing:

```
$ sudo podman ps | grep ceph-mon
$ sudo podman exec -it CONTAINER ceph osd unset noout
$ sudo podman exec -it CONTAINER ceph osd unset norebalance
```

Replace *CONTAINER* with the name of the container that runs Ceph MON.

12. Log out of the node with Ceph MON services and return to the undercloud.

19.4. SYNCHRONIZING THE OVERCLOUD STACK

The upgrade requires an update the overcloud stack to ensure that the stack resource structure and parameters align with a fresh deployment of OpenStack Platform 16.1.



NOTE

If you are not using the default stack name (**overcloud**), set your stack name with the **--stack *STACK NAME*** option replacing ***STACK NAME*** with the name of your stack.

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

2. Edit the **containers-prepare-parameter.yaml** file and remove the following parameters and their values:

- **ceph3_namespace**
- **ceph3_tag**
- **ceph3_image**
- **name_prefix_stein**
- **name_suffix_stein**
- **namespace_stein**
- **tag_stein**

3. Run the upgrade finalization command:

```
$ openstack overcloud upgrade converge \
  --stack STACK NAME \
  --templates \
  -e ENVIRONMENT FILE \
  ...
  -e /home/stack/templates/upgrades-environment.yaml \
  -e /home/stack/templates/rhsm.yaml \
  -e /home/stack/containers-prepare-parameter.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/services/neutron-ovs.yaml \
  ...
```

Include the following options relevant to your environment:

- The environment file (**upgrades-environment.yaml**) with the upgrade-specific parameters (**-e**).

- The environment file (**rhsm.yaml**) with the registration and subscription parameters (**-e**).
 - The environment file (**containers-prepare-parameter.yaml**) with your new container image locations (**-e**). In most cases, this is the same environment file that the undercloud uses.
 - The environment file (**neutron-ovs.yaml**) to maintain OVS compatibility.
 - Any custom configuration environment files (**-e**) relevant to your deployment.
 - If applicable, your custom roles (**roles_data**) file using **--roles-file**.
 - If applicable, your composable network (**network_data**) file using **--networks-file**.
 - If you use a custom stack name, pass the name with the **--stack** option.
4. Wait until the stack synchronization completes.



IMPORTANT

You do not need the **upgrades-environment.yaml** file for any further deployment operations.

PART IV. CONCLUDING THE OVERCLOUD UPGRADE

CHAPTER 20. UPGRADING TO RED HAT CEPH STORAGE 4

After you upgrade the overcloud, upgrade your Red Hat Ceph Storage cluster to version 4.

20.1. INSTALLING CEPH-ANSIBLE

The **ceph-ansible** package is required when you use Ceph Storage with Red Hat OpenStack Platform.

If you use Red Hat Ceph Storage, or if your deployment uses an external Ceph Storage cluster, install the **ceph-ansible** package. For more information about integrating with an existing Ceph Storage cluster, see [Integrating an Overcloud with an Existing Red Hat Ceph Cluster](#) .

Procedure

1. Enable the Ceph Tools repository:

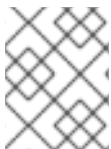
```
[stack@director ~]$ sudo subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

2. Install the **ceph-ansible** package:

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

20.2. UPGRADING TO CEPH STORAGE 4

Upgrade the Ceph Storage nodes from version 3 to version 4.



NOTE

If you are not using the default stack name (**overcloud**), set your stack name with the **--stack *STACK NAME*** option replacing ***STACK NAME*** with the name of your stack.

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

2. Run the Ceph Storage external upgrade process with the **ceph** tag:

```
$ openstack overcloud external-upgrade run --stack STACK NAME --tags ceph
```

3. Wait until the Ceph Storage upgrade completes.

CHAPTER 21. PERFORMING POST-UPGRADE ACTIONS

After you have completed the overcloud upgrade, you must perform some post-upgrade configuration to ensure that your environment is fully supported and ready for future operations.

21.1. REMOVING UNNECESSARY PACKAGES FROM THE UNDERCLOUD

After the Leapp upgrade, some unnecessary packages remain on the undercloud. Remove these packages.

Procedure

1. Remove the unnecessary packages

```
$ sudo dnf -y remove python2*
```

21.2. VALIDATING THE POST-UPGRADE FUNCTIONALITY

Run the **post-upgrade** validation group to check the post-upgrade functionality.

Procedure

1. Source the **stackrc** file.

```
$ source ~/stackrc
```

2. Run the **openstack tripleo validator run** command with the **--group post-upgrade** option:

```
$ openstack tripleo validator run --group post-upgrade
```

3. Review the results of the validation report. To view detailed output from a specific validation, run the **openstack tripleo validator show run** command against the UUID of the specific validation from the report:

```
$ openstack tripleo validator show run <UUID>
```



IMPORTANT

A **FAILED** validation does not prevent you from deploying or running Red Hat OpenStack Platform. However, a **FAILED** validation can indicate a potential issue with a production environment.

21.3. UPGRADING THE OVERCLOUD IMAGES

You must replace your current overcloud images with new versions. The new images ensure that 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. Log in to the undercloud as the **stack** user.

2. Source the **stackrc** file.

```
$ source ~/stackrc
```

3. Install the packages containing the overcloud QCOW2 archives:

```
$ sudo dnf install rhosp-director-images rhosp-director-images-ipa
```

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

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

5. Extract the archives:

```
$ cd ~/images
$ for i in /usr/share/rhosp-director-images/overcloud-full-latest-16.1.tar /usr/share/rhosp-director-images/ironic-python-agent-latest-16.1.tar; do tar -xvf $i; done
$ cd ~
```

6. Import the latest images into the director:

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

7. Configure your nodes to use the new images:

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



IMPORTANT

When you deploy overcloud nodes, ensure that the overcloud image version corresponds to the respective heat template version. For example, use the OpenStack Platform 16.1 images only with the OpenStack Platform 16.1 heat templates.



IMPORTANT

The new **overcloud-full** image replaces the old **overcloud-full** image. If you made changes to the old image, you must repeat the changes in the new image, especially if you want to deploy new nodes in the future.

21.4. MIGRATING TO OPEN VIRTUAL NETWORK (OVN)

Open Virtual Network (OVN) is an Open vSwitch-based software-defined networking (SDN) solution for supplying network services to instances. OVN provides platform-neutral support for the full OpenStack Networking API. You can use OVN to connect groups of guest instances programmatically into private L2 and L3 networks. OVN uses a standard approach to virtual networking that can extend to other Red Hat platforms and solutions.

After you upgrade to Red Hat Enterprise Linux 16.1, migrate the OpenStack Networking (neutron) service on your overcloud from the ML2/OVS driver to the ML2/OVN driver. For more information on migrating to ML2/OVN, see ["Migrating from ML2/OVS to ML2/OVN"](#).

PART V. TROUBLESHOOTING

CHAPTER 22. TROUBLESHOOTING UPGRADE ISSUES

If you experience any issues with during the upgrade process, refer to the advice in this section.

22.1. CORRECTING ENVIRONMENT FILES

If you have made a mistake with any parameters in any custom environment files, you can correct the environment file and run the **openstack overcloud upgrade prepare** command at any time during the upgrade. This command uploads a new version of your overcloud plan to director, which will generate a new set of **config-download** playbooks.

This example contains a repository name mistake in the **upgrades-environment.yaml** file:

```
parameter_defaults:
  UpgradeLeappEnabled: true
  UpgradeLeappCommandOptions: "--enablerepo rhel-7-for-x86_64-baseos-eus-rpms --enablerepo
rhel-8-for-x86_64-appstream-eus-rpms --enablerepo fast-datapath-for-rhel-8-x86_64-rpms"
  CephAnsibleRepo: rhceph-4-tools-for-rhel-8-x86_64-rpms
```

This mistake causes an issue during the Leapp upgrade for the Controller node. To rectify this issue, correct the mistake and run the **openstack overcloud upgrade prepare** command.

Procedure

1. Correct the mistake in the file:

```
parameter_defaults:
  UpgradeLeappEnabled: true
  UpgradeLeappCommandOptions: "--enablerepo rhel-8-for-x86_64-baseos-eus-rpms --
enablerepo rhel-8-for-x86_64-appstream-eus-rpms --enablerepo fast-datapath-for-rhel-8-
x86_64-rpms"
  CephAnsibleRepo: rhceph-4-tools-for-rhel-8-x86_64-rpms
```

2. Run the upgrade preparation command with the corrected file:

```
$ openstack overcloud upgrade prepare \
  --stack STACK NAME \
  --templates \
  -e ENVIRONMENT FILE
...
-e /home/stack/templates/upgrades-environment.yaml \
...
```

Wait until the overcloud stack update completes.

3. Continue with the upgrade operation step that failed.