Red Hat OpenStack Platform 16.2

Framework for Upgrades (13 to 16.2)

In-place upgrades from Red Hat OpenStack Platform 13 to 16.2

Last Updated: 2024-01-08

Red Hat OpenStack Platform 16.2 Framework for Upgrades (13 to 16.2)

In-place upgrades from Red Hat OpenStack Platform 13 to 16.2

OpenStack Team
rhos-docs@redhat.com

Legal Notice

Copyright © 2024 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 16.2 (Train).

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE 6

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION 7

CHAPTER 1. ABOUT THE RED HAT OPENSTACK PLATFORM FRAMEWORK FOR UPGRADES 8
1.1. UPGRADE FRAMEWORK FOR LONG LIFE VERSIONS 8
1.2. LIFECYCLE SUPPORT FOR LONG LIFE VERSIONS 8
1.3. UPGRADE PATHS FOR LONG LIFE RELEASES 8

CHAPTER 2. PLANNING AND PREPARATION FOR AN IN-PLACE UPGRADE 11
2.1. FAMILIARIZE YOURSELF WITH RED HAT OPENSTACK PLATFORM 16.2 11
2.2. HIGH LEVEL CHANGES IN RED HAT OPENSTACK PLATFORM 16.2 11
2.3. CHANGES IN RED HAT ENTERPRISE LINUX 8 12
2.4. LEAPP UPGRADE USAGE IN RED HAT OPENSTACK PLATFORM 13
2.5. SUPPORTED UPGRADE SCENARIOS 14
2.6. CONSIDERATIONS FOR UPGRADING WITH EXTERNAL CEPH DEPLOYMENTS 15
2.7. KNOWN ISSUES THAT MIGHT BLOCK AN UPGRADE 15
2.8. BACKUP AND RESTORE 18
2.9. MINOR VERSION UPDATE 19
2.10. PROXY CONFIGURATION 19
2.11. DELETING RHEL REGISTRATION RESOURCES 19
2.12. VALIDATING RED HAT OPENSTACK PLATFORM 13 BEFORE THE UPGRADE 19

CHAPTER 3. REPOSITORIES 21
3.1. UNDERCLOUD REPOSITORIES 21
3.2. OVERCLOUD REPOSITORIES 23
3.3. RED HAT SATELLITE SERVER 6 CONSIDERATIONS 27

CHAPTER 4. PREPARING FOR THE UNDERCLOUD UPGRADE 30
4.1. UPGRADING WITH EXTERNAL CEPH PREREQUISITE 30
4.2. NEW MEMORY REQUIREMENTS 30
4.3. USING PREDICTABLE NIC NAMES FOR THE UNDERCLOUD NODE 30
4.4. SETTING THE SSH ROOT PERMISSION PARAMETER ON THE UNDERCLOUD 34
4.5. CONVERTING TO NEXT GENERATION POWER MANAGEMENT DRIVERS 34

CHAPTER 5. UPGRADING THE UNDERCLOUD OPERATING SYSTEM 36
5.1. REMOVING RED HAT OPENSTACK PLATFORM DIRECTOR PACKAGES 36
5.2. PERFORMING A LEAPP UPGRADE ON THE UNDERCLOUD 36
5.3. CUSTOMIZING THE BASE PACKAGES AFTER A LEAPP UPGRADE 39

CHAPTER 6. UPGRADING DIRECTOR 41
6.1. LOCKING THE ENVIRONMENT TO A RED HAT ENTERPRISE LINUX RELEASE 41
6.2. ENABLING REPOSITORIES FOR THE UNDERCLOUD 41
6.3. INSTALLING DIRECTOR PACKAGES 42
6.4. PREPARING CONTAINER IMAGES 42
6.5. CONTAINER IMAGE PREPARATION PARAMETERS 43
6.6. GUIDELINES FOR CONTAINER IMAGE TAGGING 46
6.7. OBTAINING CONTAINER IMAGES FROM PRIVATE REGISTRIES 47
6.8. OBTAINING TRANSITIONAL CONTAINERS FOR UPGRADES 49
6.9. UPDATING THE UNDERCLOUD.CONF FILE 51
6.10. DIRECTOR CONFIGURATION PARAMETERS 51
6.11. RUNNING THE DIRECTOR UPGRADE 58

CHAPTER 7. INITIAL STEPS FOR OVERCLOUD PREPARATION 60
7.1. PREPARING FOR OVERCLOUD SERVICE DOWNTIME 60
7.2. SELECTING COMPUTE NODES FOR UPGRADE TESTING 60
7.3. CREATING AN OVERCLOUD INVENTORY FILE 60
7.4. VALIDATING THE PRE-UPGRADE REQUIREMENTS 61
7.5. DISABLING FENCING IN THE OVERCLOUD 62
7.6. CHECKING CUSTOM PUPPET PARAMETERS 62

CHAPTER 8. CONFIGURING THE OVERCLOUD FOR A LEAPP UPGRADE 64
8.1. CREATING AN UPGRADES ENVIRONMENT FILE 64
8.2. UPGRADE PARAMETERS 64
8.3. USING PREDICTABLE NIC NAMES FOR OVERCLOUD NODES 65

CHAPTER 9. UPDATING COMPOSABLE SERVICES AND PARAMETERS 67
9.1. UPDATING COMPOSABLE SERVICES IN CUSTOM ROLES_DATA FILES 67
9.2. UPDATING COMPOSABLE SERVICES IN CUSTOM ENVIRONMENT FILES 72
9.3. CONFIGURING ACCESS TO THE UNDERCLOUD REGISTRY 73
9.4. DEPRECATED AND REMOVED FILTERS FOR THE NOVASCHEDULERDEFAULTFILTERS PARAMETER 73
9.5. SETTING THE COMPUTE NAME FORMAT 74
9.6. CONFIGURING THE INSTANCE SERIAL NUMBER 75
9.7. UPDATING THE SSL/TLS CONFIGURATION 75
9.8. UPDATING POST CONFIGURATION TEMPLATES 76
9.9. CONSIDERATIONS WHEN RETAINING LEGACY TELEMETRY SERVICES 76

CHAPTER 10. UPDATING OVERCLOUD REGISTRATION TO THE RED HAT CUSTOMER PORTAL 77
10.1. RED HAT SUBSCRIPTION MANAGER (RHSM) COMPOSABLE SERVICE 77
10.2. RHSMVARS SUB-PARAMETERS 77
10.3. SWITCHING TO THE RHSM COMPOSABLE SERVICE 78
10.4. RHEL-REGISTRATION TO RHSM MAPPINGS 79
10.5. REGISTERING THE OVERCLOUD WITH THE RHSM COMPOSABLE SERVICE 80
10.6. APPLYING THE RHSM COMPOSABLE SERVICE TO DIFFERENT ROLES 81

CHAPTER 11. UPDATING OVERCLOUD REGISTRATION TO RED HAT SATELLITE SERVER 83
11.1. RED HAT SUBSCRIPTION MANAGER (RHSM) COMPOSABLE SERVICE 83
11.2. RHSMVARS SUB-PARAMETERS 83
11.3. SWITCHING TO THE RHSM COMPOSABLE SERVICE 84
11.4. RHEL-REGISTRATION TO RHSM MAPPINGS 85
11.5. REGISTERING THE OVERCLOUD TO RED HAT SATELLITE SERVER 86
11.6. PREPARING LEAPP TO USE SATELLITE SERVER 87

CHAPTER 12. PREPARING FOR A DIRECTOR-DEPLOYED CEPH STORAGE UPGRADE 89
12.1. UNDERSTANDING THE CEPH STORAGE NODE UPGRADE PROCESS AT A HIGH LEVEL 89
12.2. CHECKING YOUR CEPH-ANSIBLE VERSION 90
12.3. SETTING THE CEPH-ANSIBLE REPOSITORY 91
12.4. CHECKING CEPH CLUSTER STATUS BEFORE AN UPGRADE 92

CHAPTER 13. PREPARING FOR UPGRADING WITH EXTERNAL CEPH DEPLOYMENTS 93
13.1. INSTALLING CEPH-ANSIBLE 93
13.2. SETTING THE CEPH-ANSIBLE REPOSITORY 93

CHAPTER 14. UPDATING NETWORK CONFIGURATION 95
14.1. UPDATING NETWORK INTERFACE TEMPLATES 95
14.2. MAINTAINING OPEN VSWITCH COMPATIBILITY DURING THE UPGRADE 96
14.3. MAINTAINING COMPOSABLE NETWORK COMPATIBILITY DURING THE UPGRADE 97

CHAPTER 15. PREPARING NETWORK FUNCTIONS VIRTUALIZATION (NFV) 98
15.1. NETWORK FUNCTIONS VIRTUALIZATION (NFV) ENVIRONMENT FILES 98

CHAPTER 16. FINAL REVIEW BEFORE UPGRADE 100
16.1. CUSTOM FILES TO INCLUDE IN YOUR DEPLOYMENT 100
16.2. NEW ENVIRONMENT FILES TO INCLUDE WITH YOUR DEPLOYMENT 100
16.3. ENVIRONMENT FILES TO REMOVE FROM YOUR DEPLOYMENT 100
16.4. UPGRADE CHECKLIST 101

CHAPTER 17. UPGRADE COMMAND OVERVIEW 103
17.1. OPENSTACK OVERCLOUD UPGRADE PREPARE 103
17.2. OPENSTACK OVERCLOUD UPGRADE RUN 103
17.3. OPENSTACK OVERCLOUD EXTERNAL-UPGRADE RUN 103
17.4. OPENSTACK OVERCLOUD UPGRADE CONVERGE 104
17.5. OVERCLOUD NODE UPGRADE WORKFLOW 104

CHAPTER 18. UPGRADING A STANDARD OVERCLOUD 107
18.1. RUNNING THE OVERCLOUD UPGRADE PREPARATION 107
18.2. UPGRADING CONTROLLER NODES 108
18.3. UPGRADING CONTROLLER NODES WITH DIRECTOR-DEPLOYED CEPH STORAGE 110
18.4. UPGRADING THE OPERATING SYSTEM FOR CEPH STORAGE NODES 113
18.5. UPGRADING COMPUTE NODES 116
18.6. SYNCHRONIZING THE OVERCLOUD STACK 116

CHAPTER 19. UPGRADING AN OVERCLOUD WITH EXTERNAL CEPH DEPLOYMENTS 119
19.1. RUNNING THE OVERCLOUD UPGRADE PREPARATION 119
19.2. UPGRADING CONTROLLER NODES WITH EXTERNAL CEPH DEPLOYMENTS 120
19.3. UPGRADING COMPUTE NODES 123
19.4. SYNCHRONIZING THE OVERCLOUD STACK 123

CHAPTER 20. SPEEDING UP AN OVERCLOUD UPGRADE 126
20.1. RUNNING THE OVERCLOUD UPGRADE PREPARATION 126
20.2. UPGRADING THE CONTROL PLANE NODES 127
20.3. UPGRADING COMPUTE NODES IN PARALLEL 131
20.4. SYNCHRONIZING THE OVERCLOUD STACK 133
20.5. UPGRADING THE ENTIRE OVERCLOUD AT ONCE 134

CHAPTER 21. UPGRADING A SPLIT CONTROLLER OVERCLOUD 136
21.1. RUNNING THE OVERCLOUD UPGRADE PREPARATION 136
21.2. UPGRADING PACEMAKER-BASED NODES 137
21.3. UPGRADING NON-PACEMAKER CONTROLLER NODES 139
21.4. UPGRADING THE OPERATING SYSTEM FOR CEPH MON NODES 140
21.5. UPGRADING THE OPERATING SYSTEM FOR CEPH STORAGE NODES 142
21.6. UPGRADING COMPUTE NODES 145
21.7. SYNCHRONIZING THE OVERCLOUD STACK 146

CHAPTER 22. UPGRADING AN OVERCLOUD WITH HYPER-CONVERGED INFRASTRUCTURE 148
22.1. RUNNING THE OVERCLOUD UPGRADE PREPARATION 148
22.2. UPGRADING CONTROLLER NODES WITH DIRECTOR-DEPLOYED CEPH STORAGE 149
22.3. UPGRADING COMPUTE NODES WITH HYPER-CONVERGED INFRASTRUCTURE (HCI) 152
22.4. SYNCHRONIZING THE OVERCLOUD STACK 154

CHAPTER 23. UPGRADING A DIRECTOR-DEPLOYED CEPH STORAGE CLUSTER TO RED HAT CEPH STORAGE 4 156
23.1. INSTALLING CEPH-ANSIBLE 156

23.2. UPGRADING TO CEPH STORAGE 4 156

CHAPTER 24. OSD MIGRATION FROM FILESTORE TO BLUESTORE 158
24.1. CHECKING THAT YOUR CLUSTER RUNS FILESTORE AND THEREFORE REQUIRES MIGRATION 158
24.2. MIGRATING OSDS FROM FILESTORE TO BLUESTORE 158
24.3. VERIFYING YOUR FILESTORE TO BLUESTORE MIGRATION 160

CHAPTER 25. PERFORMING POST-UPGRADE ACTIONS 162
25.1. REMOVING UNNECESSARY PACKAGES AND DIRECTORIES FROM THE UNDERCLOUD 162
25.2. DELETING USERS OF REDUNDANT TELEMETRY SERVICES 162
25.3. VALIDATING THE POST-UPGRADE FUNCTIONALITY 163
25.4. UPGRADING THE OVERCLOUD IMAGES 163
25.5. UPDATING CPU PINNING PARAMETERS 164
25.6. MIGRATING USERS TO THE MEMBER ROLE 167

CHAPTER 26. TROUBLESHOOTING UPGRADE ISSUES 168
26.1. CORRECTING ENVIRONMENT FILES 168

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message.

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Tell us how we can make it better.

Providing documentation feedback in Jira

Use the Create Issue form to provide feedback on the documentation. The Jira issue will be created in the Red Hat OpenStack Platform Jira project, where you can track the progress of your feedback.

1. Ensure that you are logged in to Jira. If you do not have a Jira account, create an account to submit feedback.
2. Click the following link to open a the **Create Issue** page: Create Issue
3. Complete the **Summary** and **Description** fields. In the **Description** field, include the documentation URL, chapter or section number, and a detailed description of the issue. Do not modify any other fields in the form.
4. Click **Create**.

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

The Red Hat OpenStack Platform (RHOSP) framework for upgrades is a workflow to upgrade your RHOSP 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 (RHOSP) **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.

The Red Hat OpenStack Platform upgrade process also upgrades the version of Red Hat Enterprise Linux (RHEL) on your nodes.

This guide provides an upgrade framework through the following versions:

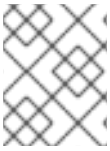
Current Version	Target Version
Red Hat OpenStack Platform 13 latest	Red Hat OpenStack Platform 16.2 latest

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 RELEASES

Familiarize yourself with the possible update and upgrade paths before you begin an update or an upgrade.



NOTE

You can view your current RHOSP and RHEL versions in the **/etc/rhosp-release** and **/etc/redhat-release** files.

Table 1.1. Updates version path

Current version	Target version
RHOSP 10.0.x on RHEL 7.x	RHOSP 10.0 latest on RHEL 7.7 latest
RHOSP 13.0.x on RHEL 7.x	RHOSP 13.0 latest on RHEL 7.9 latest
RHOSP 16.1.x on RHEL 8.2	RHOSP 16.1 latest on RHEL 8.2 latest
RHOSP 16.1.x on RHEL 8.2	RHOSP 16.2 latest on RHEL 8.4 latest

Current version	Target version
RHOSP 16.2.x on RHEL 8.4	RHOSP 16.2 latest on RHEL 8.4 latest

For more information, see Keeping Red Hat OpenStack Platform Updated .

Table 1.2. Upgrades version path

Current version	Target version
RHOSP 10 on RHEL 7.7	RHOSP 13 latest on RHEL 7.9 latest
RHOSP 13 on RHEL 7.9	RHOSP 16.1 latest on RHEL 8.2 latest
RHOSP 13 on RHEL 7.9	RHOSP 16.2 latest on RHEL 8.4 latest

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 16.2 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 1.3. Impact and duration of upgrade paths

	In-place upgrade	Parallel migration

	In-place upgrade	Parallel migration
Upgrade duration for undercloud	Estimated duration for each major action includes the following: • 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	Estimates for each Controller node: • 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	Estimates for each Compute node and Ceph Storage node: • 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 16.2

Before you perform an upgrade, familiarize yourself with Red Hat OpenStack Platform 16.2 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 16.2, 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
  - Red Hat OpenStack Platform 16.2 , which is your target version
- Read the Director Installation and Usage guide for version 16.2 and familiarize yourself with any new requirements and processes in this guide.
- Install a proof-of-concept Red Hat OpenStack Platform 16.2 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 16.2

The following high-level changes occur during the upgrade to Red Hat OpenStack Platform 16.2:

- OpenStack Platform director 16.2 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).
- Starting with Red Hat Enterprise Linux (RHEL) version 8.3, support for the Intel Transactional Synchronization Extensions (TSX) feature is disabled by default. This causes issues with instance live migration between hosts when migrating from hosts that run Red Hat OpenStack Platform 13 with RHEL version 8.2, to hosts that run Red Hat OpenStack Platform 16.2 with RHEL version 8.4.
  Instance live migration fails after you reboot the Compute nodes. To ensure that the upgraded nodes are booted with the TSX feature enabled and that you can successfully live migrate your instances, add **tsx=off** to your **KernelArgs** role parameter for the Compute node and reboot the node.

  For more information, see the Red Hat Knowledgebase solution Guidance on Intel TSX impact on OpenStack guests (applies for RHEL 8.3 and above).

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 16.2 uses Red Hat Enterprise Linux 8.4 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 8.4.

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 requires a base operating system upgrade from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8. Red Hat Enterprise Linux 7 uses the Leapp utility to perform the upgrade to Red Hat Enterprise Linux 8. To ensure that Leapp and its dependencies are available, verify that the following Red Hat Enterprise Linux 7 repositories are enabled:

- Red Hat Enterprise Linux 7 Server RPMs x86_64 7Server or Red Hat Enterprise Linux 7 Server RPMs x86_64 7.9

```
rhel-7-server-rpms
x86_64 7Server
or:
rhel-7-server-rpms
x86_64 7.9
```

- Red Hat Enterprise Linux 7 Server - Extras RPMs x86_64

```
rhel-7-server-extras-rpms
x86_64
```

For more information, see Preparing a RHEL 7 system for the upgrade .

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 **dmcrypt: true** parameter.

If any known limitations affect your environment, seek advice from the Red Hat Technical Support Team.

Troubleshooting

For information about troubleshooting potential Leapp issues, see Troubleshooting in *Upgrading from RHEL 7 to RHEL 8*.

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, including Ceph Storage custom configurations, such as **CephConfigOverrides** and **CephAnsibleExtraConfig**
- 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)
- Environments with Instance HA enabled

NOTE

During an upgrade procedure, nova live migrations are supported. However, evacuations initiated by Instance HA are not supported. When you upgrade a Compute node, the node is shut down cleanly and any workload running on the node is not evacuated by Instance HA automatically. Instead, you must perform live migration manually.

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

2.6. CONSIDERATIONS FOR UPGRADING WITH EXTERNAL CEPH DEPLOYMENTS

If you have deployed a Red Hat Ceph Storage system separately and then used director to deploy and configure OpenStack, you can use the Red Hat OpenStack Platform framework for upgrades to perform an in-place upgrade with external Ceph deployments. This scenario is different from upgrading a Ceph cluster that was deployed using director.

The differences that you must take into account when planning and preparing for an in-place upgrade with external Ceph deployments are the following:

1. Before you can upgrade your Red Hat OpenStack Platform deployment from version 13 to version 16.2, you must upgrade your Red Hat Ceph Storage cluster from version 3 to version 4. For more information, see Upgrading a Red Hat Ceph Storage cluster in the *Red Hat Ceph Storage 4 Installation Guide*.

2. After you upgrade your Red Hat Ceph Storage cluster from version 3 to version 4, Red Hat OpenStack Platform 13 might still run RHCSv3 client components, however these are compatible against the RHCSv4 cluster.

3. You can follow the upgrade path described in the Framework For Upgrades (13 to 16.2) document, and where applicable, you must complete the conditional steps that support this particular scenario. A conditional step starts with the following statement: "If you are upgrading with external Ceph deployments".

4. When you upgrade with external Ceph deployments, you install RHCSv4 **ceph-ansible** as part of the overcloud upgrade process. When you upgrade a Ceph cluster that was deployed using director, you install RHCSv4 **ceph-ansible** after the overcloud upgrade process is complete.

IMPORTANT

When you upgrade a Red Hat Ceph Storage cluster from a previous supported version to version 4.2z2, the upgrade completes with the storage cluster in a **HEALTH_WARN** state with a warning message that states **monitors are allowing insecureglobal_id reclaim**. This is due to the patched CVE (CVE-2021-20288), see Ceph HEALTH_WARN with 'mons are allowing insecure global_id reclaim' after install/upgrade to RHCS 4.2z2 (or newer).

Because the **HEALTH_WARN** state is displayed due to the CVE, it is possible to mute health warnings temporarily. However, there is a risk that if you mute warnings you do not have visibility about potential older and unpatched clients connected to your cluster. For more information about muting health warnings, see Upgrading a Red Hat Ceph Storage cluster in the *Red Hat Ceph Storage* documentation.

2.7. KNOWN ISSUES THAT MIGHT BLOCK AN UPGRADE

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

BZ#2228414 - Missing service_user for nova_compute causes nova hybrid state to fail

A service token is now required for OpenStack Compute (nova) and Openstack Block (cinder) services. During an upgrade from Red Hat OpenStack Platform (RHOSP) 13 to 16.2, if the service token is not configured, live migrations fail with the following error in **nova-compute.log**: **"2023-xx-xx xx:xx:xx.xxx 8 ERROR oslo_messaging.rpc.server [...] Exception during message handling: cinderclient.exceptions.ClientException: ConflictNovaUsingAttachment: Detach volume from instance XXXXXX using the Compute API (HTTP 409) (Request-ID: req-XXXXXX)"**

To avoid this issue, apply the fix from RHBA-2023:5163 - Bug Fix Advisory. You must apply the fix after the undercloud upgrade, but before starting the overcloud adoption.

BZ#1902849 - osp13-osp16.1 ffu fails on clusters previously upgraded from osp8, osp10

Red Hat OpenStack Platform (RHOSP) environments that have been previously upgraded from version RHOSP 10, require the **python-docker** package to avoid BZ#1902849. For more information, see the Red Hat Knowledgebase solution osp13-osp16.1 ffu fails on older environments missing python-docker package.

BZ#1925078 - RHOSP13-16.1 FFU: Overcloud upgrade hangs in controller after failed attempt with reference to wrong ceph image

Systems that use UEFI boot and a UEFI bootloader in OSP13 might run into an UEFI issue that results in:

- **/etc/fstab** not being updated
- grub-install is incorrectly used on EFI system

For more information, see the Red Hat Knowledgebase solution FFU 13 to 16.1: Leapp fails to update the kernel on UEFI based systems and /etc/fstab does not contain the EFI partition.

If your systems use UEFI, contact Red Hat Technical Support.

BZ#1895887 - ovs+dpdk fail to attach device OvsDpdkHCI

After upgrading with the Leapp utility, the Compute node with OVS-DPDK workload does not function properly. To resolve this issue, perform one of the following steps:
Remove the **/etc/modules-load.d/vfio-pci.conf** file before you upgrade the Compute node.

or

Restart **ovs-vswitchd** service on the Compute node after you upgrade the Compute node.

This issue affects RHOSP 16.1.3. For more information, see the Red Hat Knowledgebase solution OVS-DPDK errors after Framework Upgrade from OSP 13 to 16.1 on HCI compute node .

BZ#1923165 - OSP-16.2 (Upgrades)(TripleO) Add a config to disable Intel "TSX" on RHEL-8.3 kernel

Starting with Red Hat Enterprise Linux (RHEL) version 8.3, support for the Intel Transactional Synchronization Extensions (TSX) feature is disabled by default. This causes issues with instance live migration between hosts in the following migration scenario:

- Migrating from hosts where the TSX kernel argument is enabled to hosts where the TSX kernel argument is disabled.

Live migration can be unsuccessful in Intel hosts that support the TSX feature. For more information about the CPUs that are affected by this issue, see Affected Configurations.

For more information, review the following Red Hat Knowledgebase solution Guidance on Intel TSX impact on OpenStack guests.

BZ#2016144 - FFU 13-16.1: During Leapp upgrade reboot, openvswitch failed to start with error Starting ovsdb-server ovsdb-server: /var/run/openvswitch/ovsdb-server.pid.tmp: create failed (Permission denied)

Red Hat OpenStack Platform (RHOSP) environments that have been upgraded from previous versions might contain unnecessary files in **/etc/systemd/system/ovs***. You must remove these files before you begin the overcloud upgrade process from RHOSP 13 to RHOSP 16.2.

BZ#2021525 - openstack overcloud upgrade run times out / HAProxy container fails to start

An upgrade from Red Hat OpenStack Platform (RHOSP) 13 to RHOSP 16.2 might fail during the deployment step because of invalid SELinux labels. For a resolution and more information, see the Red Hat Knowledgebase solution Pacemaker managed services might not restart during an OSP13 - OSP16.x FFU.

BZ#2027787 - Undercloud upgrade to 16.2 fails because of missing dependencies of swtpm

There is a known issue with the **advanced-virt-for-rhel-8-x86_64-rpms** and **advanced-virt-for-rhel-8-x86_64-eus-rpms** repositories that prevents a successful upgrade. To disable these repositories before upgrading, see the Red Hat Knowledgebase solution advanced-virt-for-rhel-8-x86_64-rpms are no longer required in OSP 16.2.

BZ#2024447 - Identity service (keystone) password for the placement user was overridden by NovaPassword during FFU RHOSP 13 to 16

During an upgrade from Red Hat OpenStack Platform 13 to 16.2, if you define a value for the **NovaPassword** parameter but not the **PlacementPassword** parameter, the **NovaPassword** parameter overrides the OpenStack Identity service (keystone) password for the placement user. To preserve the Identity service password, do not set the **NovaPassword** or the **PlacementPassword** in the **parameter_defaults** section.
If you set both passwords in the **parameter_defaults** section, the Compute nodes might not be able to communicate with the control plane until they are upgraded. For more information about upgrading Compute nodes, see Upgrading Compute nodes.

Additionally, if you deployed the overcloud on RHOSP 13 by using the **NovaPassword**, **PlacementPassword**, or both, you must remove those passwords from the template and run the **openstack overcloud deploy** command on RHOSP 13 before upgrading to RHOSP 16.2.

BZ#2141186 - Live migration fails due to qemu error during in-place upgrade

During or after an in-place upgrade from Red Hat OpenStack Platform (RHOSP) 13 to RHOSP 16.2, live migration between 16.2 Compute nodes fails on instances with the following configuration:

- Multi-queue is enabled.
- The number of allocated vcps is 9 or more.
- The instance is running on RHOSP 13.

To successfully migrate your Compute nodes during an upgrade, add the following parameter to your custom environment file:

```
parameter_defaults:
  ComputeExtraConfig:
    nova::compute::libvirt::max_queues: 8
```

Include your updated custom environment file when you run the following commands during the upgrade:

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

Optionally, after you complete the upgrade, include the custom environment file with the parameter when you run the **openstack overcloud deploy** command.

For more information, see the Red Hat Knowledgebase solution Live migration fails due to qemu error in in-place upgrades environment.

BZ#2141393 - cephvolumescan actor fails

If your environment includes both Ceph and non-Ceph containers, the Leapp upgrade fails because the **cephvolumescan** actor cannot retrieve the ceph volumes list.
To disable the **cephvolumescan** actor and complete the Leapp upgrade, add the following parameter to your template:

```
parameter_defaults:
  LeappActorsToRemove: ['cephvolumescan']
```

BZ#2164396 - FFU: Redhat satellite tools repository to be enabled for FFU (13 to 16.2)

If you are using Satellite version 6.7, the upgrade fails when you enable the Red Hat Satellite Tools for RHEL 8 Server RPMs x86_64 repository. The failure occurs because the appropriate packages cannot be installed. The Red Hat engineering team is investigating a solution to this issue.

BZ#2245602 - Upgrade (OSP16.2 →OSP17.1) controller-0 does not perform leapp upgrade due to packages missing ovn2.15 openvswitch2.15

If you upgrade from Red Hat OpenStack Platform (RHOSP) 13 to 16.1 or 16.2, or from RHOSP 16.2 to 17.1, do not include the **system_upgrade.yaml** file in the **--answers-file answer-upgrade.yaml** file. If the **system_upgrade.yaml** file is included in that file, the **environments/lifecycle/upgrade-prepare.yaml** file overwrites the parameters in the **system_upgrade.yaml** file. To avoid this issue, append the **system_upgrade.yaml** file to the **openstack overcloud upgrade prepare** command. For example:

```
$ openstack overcloud upgrade prepare --answers-file answer-upgrade.yaml /
-r roles-data.yaml /
-n networking-data.yaml /
-e system_upgrade.yaml /
-e upgrade_environment.yaml /
```

With this workaround, the parameters that are configured in the **system_upgrade.yaml** file overwrite the default parameters in the **environments/lifecycle/upgrade-prepare.yaml** file.

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 16.2 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.

2.8. 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 about backing up nodes before you upgrade, see Red Hat OpenStack Platform 13 Undercloud and Control Plane Back Up and Restore.
- You can back up each node after it has been upgraded. For more information about backing up upgraded nodes, see Red Hat OpenStack Platform 16.2 Backing up and restoring the undercloud and control plane nodes.
- You can back up the database that runs on the undercloud node after you perform the undercloud upgrade and before you perform the overcloud upgrade. For more information about backing up the undercloud database, see Creating a database backup of the undercloud node in the *Red Hat OpenStack Platform 16.2 Backing up and restoring the undercloud and control plane nodes* guide.

2.9. 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 16.2.

For instructions on performing a minor version update for Red Hat OpenStack Platform 13, see Keeping Red Hat OpenStack Platform Updated.

2.10. 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 16.2 upgrade.

- For more information about proxy configuration for Red Hat OpenStack Platform 13, see Considerations when running the undercloud with a proxy .
- For more information about proxy configuration for Red Hat OpenStack Platform 16.2, see Considerations when running the undercloud with a proxy .

2.11. DELETING RHEL REGISTRATION RESOURCES

If the **DeleteOnRHELUnregistration** parameter is set to **true** in an existing environment file or the **rhel-registration.yaml** template, the overcloud upgrade cannot proceed. In this case, when you perform a minor update to the latest Red Hat OpenStack Platform 13z version, set the **DeleteOnRHELUnregistration** parameter to **false**.

Procedure

1. In the **parameter_defaults** section of your environment file, if the **DeleteOnRHELUnregistration** parameter is set to **true**, set the parameter to **false**.
2. Run the **openstack overcloud update prepare** command.
3. Run the **openstack undercloud upgrade** command.

2.12. VALIDATING RED HAT OPENSTACK PLATFORM 13 BEFORE THE UPGRADE

Before you upgrade to Red Hat OpenStack Platform 16.2, 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).

NOTE

If you use CDN or Satellite as repository sources, the validation fails. To resolve this issue, see the Red Hat Knowledgebase solution, repos validation fails because of SSL certificate error.

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

RHOSP (RHOSP) 16.2 runs on Red Hat Enterprise Linux (RHEL) 8.4. As a result, you must lock the content from these repositories to the respective RHEL version.

NOTE

- If you synchronize repositories by using Red Hat Satellite, you can enable specific versions of the RHEL repositories. However, the repository label remains the same despite the version you choose. For example, if you enable the 8.4 version of the BaseOS repository, the repository name includes the specific version that you enabled, but the repository label is still **rhel-8-for-x86_64-baseos-eus-rpms**.
- The **advanced-virt-for-rhel-8-x86_64-rpms** and **advanced-virt-for-rhel-8-x86_64-eus-rpms** repositories are no longer required. To disable these repositories, see the Red Hat Knowledgebase solution advanced-virt-for-rhel-8-x86_64-rpms are no longer required in OSP 16.2.

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 RHOSP dependencies.

Name	Repository	Description of requirement
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 RHEL. Used for Controller node high availability.
Red Hat Ansible Engine 2.9 for RHEL 8 x86_64 (RPMs)	**ansible-2.9-for-rhel-8-x86_64-rpms**	Ansible Engine for RHEL. Used to provide the latest version of Ansible.
RHOSP 16.2 for RHEL 8 (RPMs)	**openstack-16.2-for-rhel-8-x86_64-rpms**	Core RHOSP repository, which contains packages for RHOSP 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 or if you want to integrate with an existing Ceph Storage cluster.

IBM POWER repositories

The following table contains a list of repositories for RHOSP 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 RHOSP 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 RHEL. Used for Controller node high availability.
Red Hat Fast Datapath for RHEL 8 IBM Power, little endian (RPMS)	**fast-datapath-for-rhel-8-ppc64le-rpms**	Provides Open vSwitch (OVS) packages for OpenStack Platform.
Red Hat Ansible Engine 2.9 for RHEL 8 IBM Power, little endian (RPMs)	**ansible-2.9-for-rhel-8-ppc64le-rpms**	Ansible Engine for RHEL. Provides the latest version of Ansible.
Red Hat OpenStack Platform 16.2 for RHEL 8 (RPMs)	**openstack-16.2-for-rhel-8-ppc64le-rpms**	Core RHOSP repository for ppc64le systems.

3.2. OVERCLOUD REPOSITORIES

Red Hat OpenStack Platform (RHOSP) 16.2 runs on Red Hat Enterprise Linux (RHEL) 8.4. As a result, you must lock the content from these repositories to the respective RHEL version.

NOTE

- If you synchronize repositories by using Red Hat Satellite, you can enable specific versions of the RHEL repositories. However, the repository label remains the same despite the version you choose. For example, if you enable the 8.4 version of the BaseOS repository, the repository name includes the specific version that you enabled, but the repository label is still **rhel-8-for-x86_64-baseos-eus-rpms**.
- The **advanced-virt-for-rhel-8-x86_64-rpms** and **advanced-virt-for-rhel-8-x86_64-eus-rpms** repositories are no longer required. To disable these repositories, see the Red Hat Knowledgebase solution advanced-virt-for-rhel-8-x86_64-rpms are no longer required in OSP 16.2.

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).

Controller node repositories

The following table lists core repositories for Controller nodes in 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 RHOSP 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 RHEL.
Red Hat Ansible Engine 2.9 for RHEL 8 x86_64 (RPMs)	**ansible-2.9-for-rhel-8-x86_64-rpms**	Ansible Engine for RHEL. Used to provide the latest version of Ansible.
Red Hat OpenStack Platform 16.2 for RHEL 8 (RPMs)	**openstack-16.2-for-rhel-8-x86_64-rpms**	Core RHOSP 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.
Red Hat Ceph Storage Tools 4 for RHEL 8 x86_64 (RPMs)	**rhceph-4-tools-for-rhel-8-x86_64-rpms**	Tools for Red Hat Ceph Storage 4 for RHEL 8.

Compute and ComputeHCI node repositories

The following table lists core repositories for Compute and ComputeHCI nodes in 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 RHOSP 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 RHEL.
Red Hat Ansible Engine 2.9 for RHEL 8 x86_64 (RPMs)	**ansible-2.9-for-rhel-8-x86_64-rpms**	Ansible Engine for RHEL. Used to provide the latest version of Ansible.

Name	Repository	Description of requirement
Red Hat OpenStack Platform 16.2 for RHEL 8 (RPMs)	**openstack-16.2-for-rhel-8-x86_64-rpms**	Core RHOSP 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.
Red Hat Ceph Storage Tools 4 for RHEL 8 x86_64 (RPMs)	**rhceph-4-tools-for-rhel-8-x86_64-rpms**	Tools for Red Hat Ceph Storage 4 for RHEL 8.

Real Time Compute 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.
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.

Ceph Storage node 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.

Name	Repository	Description of requirement
Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)	**rhel-8-for-x86_64-appstream-rpms**	Contains RHOSP 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 RHEL. NOTE: If you used the **overcloud-full** image for your Ceph Storage role, you must enable this repository. Ceph Storage roles should use the **overcloud-minimal** image, which does not require this repository.
Red Hat Ansible Engine 2.9 for RHEL 8 x86_64 (RPMs)	**ansible-2.9-for-rhel-8-x86_64-rpms**	Ansible Engine for RHEL. Used to provide the latest version of Ansible.
Red Hat OpenStack Platform 16.2 Director Deployment Tools for RHEL 8 x86_64 (RPMs)	**openstack-16.2-deployment-tools-for-rhel-8-x86_64-rpms**	Packages to help director configure Ceph Storage nodes. This repository is included with standalone Ceph Storage subscriptions. If you use a combined OpenStack Platform and Ceph Storage subscription, use the **openstack-16.2-for-rhel-8-x86_64-rpms** repository.
Red Hat OpenStack Platform 16.2 for RHEL 8 (RPMs)	**openstack-16.2-for-rhel-8-x86_64-rpms**	Packages to help director configure Ceph Storage nodes. This repository is included with combined OpenStack Platform and Ceph Storage subscriptions. If you use a standalone Ceph Storage subscription, use the **openstack-16.2-deployment-tools-for-rhel-8-x86_64-rpms** repository.
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.
Red Hat Fast Datapath for RHEL 8 (RPMS)	**fast-datapath-for-rhel-8-x86_64-rpms**	Provides Open vSwitch (OVS) packages for OpenStack Platform. If you are using OVS on Ceph Storage nodes, add this repository to the network interface configuration (NIC) templates.

IBM POWER repositories

The following table lists repositories for RHOSP 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 RHOSP dependencies.
Red Hat Enterprise Linux 8 for IBM Power, little endian - High Availability (RPMs)	**rhel-8-for-ppc64le-highavailability-rpms**	High availability tools for RHEL. Used for Controller node high availability.
Red Hat Fast Datapath for RHEL 8 IBM Power, little endian (RPMS)	**fast-datapath-for-rhel-8-ppc64le-rpms**	Provides Open vSwitch (OVS) packages for OpenStack Platform.
Red Hat Ansible Engine 2.9 for RHEL 8 IBM Power, little endian (RPMs)	**ansible-2.9-for-rhel-8-ppc64le-rpms**	Ansible Engine for RHEL. Used to provide the latest version of Ansible.
Red Hat OpenStack Platform 16.2 for RHEL 8 (RPMs)	**openstack-16.2-for-rhel-8-ppc64le-rpms**	Core RHOSP repository for ppc64le systems.

3.3. RED HAT SATELLITE SERVER 6 CONSIDERATIONS

If you use Red Hat Satellite Server 6 to host RPMs and container images for your Red Hat OpenStack Platform environment, you must account for certain considerations when you use Satellite Server 6 to deliver content during the Red Hat OpenStack Platform 16.2 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 your 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 8.4 repositories.
- Create custom products on your Satellite Server to host container images for the following Red Hat OpenStack Platform versions:

- Red Hat OpenStack Platform 16.2
- Red Hat OpenStack Platform 15

- Create and promote a content view for Red Hat OpenStack Platform 16.2 upgrade and include the following content in the content view:
  - The following Red Hat Enterprise Linux 7 repositories:
    - Red Hat Enterprise Linux 7 Server RPMs x86_64 7Server or Red Hat Enterprise Linux 7 Server RPMs x86_64 7.9

      ```
      rhel-7-server-rpms
      x86_64 7Server
      or:
      rhel-7-server-rpms
      x86_64 7.9
      ```

    - Red Hat Enterprise Linux 7 Server - Extras RPMs x86_64

      ```
      rhel-7-server-extras-rpms
      x86_64
      ```

  - All undercloud and overcloud RPM repositories, including Red Hat Enterprise Linux 8.4 repositories. Ensure that you include the correct version of the Red Hat Enterprise Linux repositories, which is 8.4. If you don't include the correct version, you might run into issues with enabling RHEL 8 repositories. For more information, see the Red Hat Knowledgebase solution RHEL 7 to RHEL 8 LEAPP Upgrade Failing When Using Red Hat Satellite .
  - Red Hat OpenStack Platform 16.2 container images.
  - Red Hat OpenStack Platform 15 container images.

- Associate an activation key with the Red Hat OpenStack Platform 16.2 content view that you have created for the Red Hat OpenStack Platform 16.2 upgrade.

- 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 8.4 system causes **subscription-manager** to report errors.

- You can configure Satellite Server to host Red Hat OpenStack Platform 16.2 container images. For more information, see Preparing a Satellite server for container images in *Director Installation and Usage*.

- If you use a Ceph subscription and have configured director to use the **overcloud-minimal** image for Ceph storage nodes, on your Satellite Server you must create a Content View and add the following Red Hat Enterprise Linux (RHEL) 8.4 repositories to it:
  - Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)

    ```
    rhel-8-for-x86_64-appstream-rpms
    x86_64 8.4
    ```

  - Red Hat Enterprise Linux 8 for x86_64 - BaseOS (RPMs)

```
rhel-8-for-x86_64-baseos-rpms
x86_64 8.4
```

For more information, see Importing Red Hat Content and Managing Content Views in the *Red Hat Satellite Content Management Guide*.

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. UPGRADING WITH EXTERNAL CEPH PREREQUISITE

If you are upgrading with external Ceph deployments, before you can upgrade your Red Hat OpenStack Platform deployment, you must upgrade your Red Hat Ceph Storage cluster from version 3 to version 4. For more information, see Upgrading a Red Hat Ceph Storage cluster in the *Red Hat Ceph Storage 4 Installation Guide* .

4.2. NEW MEMORY REQUIREMENTS

In Red Hat OpenStack Platform 16.2, the undercloud has new memory requirements:

Red Hat OpenStack Platform 13	Red Hat OpenStack Platform 16.2
16 GB RAM	24 GB RAM

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

4.3. 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 an **eth** prefix. These NIC names are usually unpredictable in terms of NIC assignments.

You can run the **playbook-nics.yaml** playbook to rename NIC names to use the **em** NIC prefix. You can also set a different NIC prefix, by modifying the **prefix** variable when running the playbook. However, the NIC changes are only applied after the Leapp upgrade process completes and the node is rebooted.

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"
    undercloud_conf: "/home/stack/undercloud.conf"
    osnet_conf: "/etc/os-net-config/config.json"
  tasks:
    - set_fact:
        eth_interfaces: "{{ ansible_interfaces | select('match','eth.*') | list }}"
    - debug:
        msg: "{{ eth_interfaces }}"
```

```
    - 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: "{{ eth_interfaces }}"
    - name: Rename eth files
      block:
        - name: Check that eth files exists
          stat:
            path: /etc/sysconfig/network-scripts/ifcfg-{{ item }}
          register: nic_result
          with_items: "{{ eth_interfaces }}"
        - 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.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.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.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.stat.exists
        - name: Remove old eth network-script files
          file:
            path: "{{ item.stat.path }}"
            state: absent
          with_items: "{{ nic_result.results }}"
          when: 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: "{{ eth_interfaces }}"
```

```
    - 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.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.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.stat.exists
    - name: Remove old route files
      file:
        path: "{{ item.stat.path }}"
        state: absent
      with_items: "{{ route_result.results }}"
      when: item.stat.exists
- name: Perform a final regex for any remaining eth prefixes in ifcfg files
  block:
    - name: Get a list of all ifcfg files
      find:
        paths: /etc/sysconfig/network-scripts/
        patterns: 'ifcfg-*'
        excludes: '*.bak'
      register: ifcfg_files
    - name: Perform final regex on ifcfg files
      replace:
        path: "{{ item[0].path }}"
        regexp: "{{ item[1] }}"
        replace: "{{ item[1]|replace('eth',prefix) }}"
      with_nested:
        - "{{ ifcfg_files.files }}"
        - "{{ eth_interfaces }}"
- name: Replace interface name in files referencing old eth interface
  block:
    - name: Check if undercloud.conf exists
      stat:
        path: "{{ undercloud_conf }}"
      register: undercloud_conf_stat
    - name: Replace interface name in undercloud.conf
      replace:
        path: "{{ undercloud_conf }}"
        regexp: 'eth(\d+)'
        replace: "{{ prefix }}\\1"
      when: undercloud_conf_stat.stat.exists
    - name: Check if os-net-config's config.json exists
```

```
      stat:
        path: "{{ osnet_conf }}"
      register: osnet_conf_stat
    - name: Replace interface name in config.json
      replace:
        path: "{{ osnet_conf }}"
        regexp: 'eth(\d+)'
        replace: "{{ prefix }}\\1"
      when: osnet_conf_stat.stat.exists
- name: Patch vlan devices
  block:
    - name: Check that vlan files exists
      stat:
        path: /etc/sysconfig/network-scripts/ifcfg-{{ item }}
      register: nic_result
      when: item.startswith("vlan")
      with_items: "{{ ansible_interfaces }}"
    - name: Backup old vlan network-script files
      copy:
        remote_src: True
        src: "{{ item.stat.path }}"
        dest: "{{ item.stat.path }}.bak"
      when: item.item.startswith("vlan") and item.stat.exists
      with_items: "{{ nic_result.results }}"
    - name: Edit PHYSDEV in new network-script files
      replace:
        path: "{{ item.stat.path }}"
        regexp: "^PHYSDEV=eth"
        replace: "PHYSDEV={{ prefix }}"
      when: item.item.startswith("vlan") and item.stat.exists
      with_items: "{{ nic_result.results }}"
```

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
```

The NIC changes are only applied after the Leapp upgrade process completes and the node is rebooted.

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

4.4. 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.

4.5. CONVERTING TO NEXT GENERATION POWER MANAGEMENT DRIVERS

Red Hat OpenStack Platform now uses next generation drivers, also known as *hardware types*, that replace older drivers.

The following table shows an analogous comparison between older drivers with their next generation hardware type equivalent:

Old Driver	New Hardware Type
pxe_ipmitool	**ipmi**
pxe_drac	**idrac**
pxe_ilo	**ilo**
pxe_irmc	**irmc**
VBMC (**pxe_ipmitool**)	**ipmi**
fake_pxe	**fake-hardware**

In OpenStack Platform 15, these older drivers have been removed and are no longer accessible. You must change to hardware types **before** upgrading to OpenStack Platform 16.2.

Procedure

1. Check the current list of hardware types enabled:

```
$ source ~/stackrc
$ openstack baremetal driver list --type dynamic
```

2. If you use a hardware type driver that is not enabled, enable the driver using the **enabled_hardware_types** parameter in the **undercloud.conf** file:

```
enabled_hardware_types = ipmi,redfish,idrac
```

3. Save the file and refresh the undercloud:

```
$ openstack undercloud install
```

4. Run the following commands, substituting the **OLDDRIVER** and **NEWDRIVER** variables for your power management type:

```
$ source ~/stackrc
$ OLDDRIVER="pxe_ipmitool"
$ NEWDRIVER="ipmi"
$ for NODE in $(openstack baremetal node list --driver $OLDDRIVER -c UUID -f value) ; do
openstack baremetal node set $NODE --driver $NEWDRIVER; done
```

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 16.2 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-httplib2 -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 (RHEL) 8.

Prerequisites

- Before you install and run Leapp, ensure that you have familiarized yourself with the Section 2.4, "Leapp upgrade usage in Red Hat OpenStack Platform" section.
- Before you perform the Leapp upgrade, ensure that you complete the Section 4.3, "Using predictable NIC names for the undercloud node" section. If you do not rename the network interface names before you perform the Leapp upgrade process, the interface names might change after the upgrade to RHEL 8.2 is complete.

Procedure

1. Log in to the undercloud as the **stack** user.
2. Install the Leapp utility and jq:

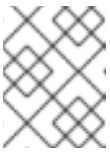
```
$ sudo yum install leapp
$ sudo yum install jq
```

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 8.4 content:

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

   - If your undercloud uses Red Hat Satellite Server for registration, re-register the undercloud to a content view associated with your Red Hat OpenStack Platform (RHOSP)16.2 activation key.

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



NOTE

The content view that you create for Red Hat OpenStack Platform 16.2 must contain content for Red Hat Enterprise Linux 8.4.

5. Red Hat OpenStack Platform 16.2 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 'openvswitch2.15' | 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. Adjust the kernel modules that are no longer supported in RHEL 8:

```
$ if [ -f /usr/share/leapp-repository/repositories/system_upgrade/el7toel8/actors/kernel/checkkerneldrivers/files/removed_drivers.txt ]; then
    for module in pata_acpi floppy; do
        sudo sed -i "/^${module}$/d" /usr/share/leapp-repository/repositories/system_upgrade/el7toel8/actors/kernel/checkkerneldrivers/files/removed_drivers.txt
    done
else
    for module in pata_acpi floppy; do
        jq ". | del(.data[] | select(.driver_name == \"${module}\"))" /etc/leapp/files/device_driver_deprecation_data.json | sudo tee /etc/leapp/files/device_driver_deprecation_data.json_modified
        mv /etc/leapp/files/device_driver_deprecation_data.json_modified /etc/leapp/files/device_driver_deprecation_data.json
    done
fi
```

8. Remove the **pam_pkcs11** module:

```
$ sudo leapp answer --add --section remove_pam_pkcs11_module_check.confirm=True
```

9. Optional: If your environment is deployed with a TLS-Everywhere architecture and it uses the deprecated **authconfig** utility to configure authentication on your system, configure your RHEL 8 system with the **authselect** utility:

```
$ sudo leapp answer --add --section authselect_check.confirm=True
```

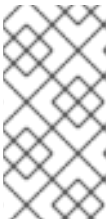
   For more information about authentication configuration during the Leapp upgrade process, see Known issues in *Upgrading from RHEL 7 to RHEL 8* .

10. Set the **LEAPP_DEVEL_TARGET_RELEASE** and **LEAPP_UNSUPPORTED** environment variables to specify the RHEL 8 minor version that you want to upgrade to. For RHOSP 16.2, you must set the RHEL 8 minor version to **8.4**:

```
$ export LEAPP_UNSUPPORTED=1
$ export LEAPP_DEVEL_TARGET_RELEASE=8.4
```

    You must use the **LEAPP_UNSUPPORTED** environment variable every time you use a environment variable with the **LEAPP_DEVEL** prefix.

11. Remove the persistent network names actor from the Leapp process:



NOTE

If you do not rename the network interface names before you perform the Leapp upgrade process, the interface names might change after the upgrade to RHEL 8.2 is complete. For more information about renaming the network interface names, see Section 4.3, “Using predictable NIC names for the undercloud node” .

```
$ sudo rm -f /usr/share/leapp-repository/repositories/system_upgrade/el7toel8/actors/persistentnetnamesdisable/actor.py
```

12. Start the Leapp upgrade process:

```
$ sudo -E 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 16.2 transition, especially with the newer version of Open vSwitch.

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

14. 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.

15. Reboot the undercloud:

```
$ sudo reboot
```

16. Remove the Leapp packages from the transaction exclusion that is defined in the DNF configuration:

```
$ sudo dnf config-manager --save --setopt exclude=''
```

Additional 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
- Performing the upgrade from RHEL 7 to RHEL 8

5.3. CUSTOMIZING THE BASE PACKAGES AFTER A LEAPP UPGRADE

You can specify additional packages to install after you upgrade your host from Red Hat Enterprise Linux (RHEL) 7.9 to RHEL 8.4. You can customize your base packages on specific roles by using the **BaseTripleoPackages** variable.

For example, after you upgrade a Ceph Storage node to RHEL 8.4, the **python3-openstackclient** package is no longer required. However, if your deployment requires this package, you can include it in the **BaseTripleoPackages** variable in your custom template.

Procedure

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

2. Source the **stackrc** undercloud credentials file:

```
$ source ~/stackrc
```

3. In your custom template, add the **BaseTripleoPackages** variable to include additional packages in your deployment. For example:

```
BaseTripleoPackages:
- jq
- lvm2
- net-snmp
- openstack-selinux
- os-net-config
- puppet-tripleo
- python3-heat-agent*
- python3-openstackclient
```

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 16.2 packages and configure the new source for Red Hat OpenStack Platform 16.2 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 16.2 is supported on Red Hat Enterprise Linux 8.4. Before you perform the update, you must lock the undercloud repositories to the Red Hat Enterprise Linux 8.4 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=8.4
```

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-tus-rpms \
--enable=rhel-8-for-x86_64-appstream-tus-rpms \
--enable=rhel-8-for-x86_64-highavailability-tus-rpms \
--enable=ansible-2.9-for-rhel-8-x86_64-rpms \
--enable=openstack-16.2-for-rhel-8-x86_64-rpms \
--enable=fast-datapath-for-rhel-8-x86_64-rpms
```

   These repositories contain packages that the director installation requires.

3. Set the **container-tools** repository module to version **3.0**:

```
[stack@director ~]$ sudo dnf module reset container-tools
[stack@director ~]$ sudo dnf module enable -y container-tools:3.0
```

4. 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.

NOTE

If you need to configure specific container image versions for your undercloud, you must pin the images to a specific version. For more information, see Pinning container images for the undercloud.

Procedure

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

```
$ sudo 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. This means that director pulls the necessary images from the Red Hat Container Catalog and pushes them to the registry on the undercloud. Director uses this registry as the container image source. To pull directly from the Red Hat Container Catalog, omit this option.
- **--output-env-file** is an environment file name. The contents of this file include the parameters for preparing your container images. In this case, 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

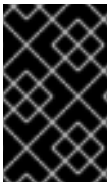
The default file for preparing your containers (**containers-prepare-parameter.yaml**) contains the **ContainerImagePrepare** heat parameter. This parameter defines a list of strategies for preparing a set of images:

```
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 that you can use with each **ContainerImagePrepare** strategy:

Parameter	Description
excludes	List of regular expressions to exclude image names from a strategy.
includes	List of regular expressions 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 16.2.3-5.161 and set the **modify_append_tag** to **-hotfix**, the director tags the final image as 16.2.3-5.161-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	Defines the namespace of the registry that you want to push images to during the upload process. • 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. If you set this parameter to **false** in production environments while pulling images directly from Red Hat Container Catalog, all overcloud nodes will simultaneously pull the images from the Red Hat Container Catalog over your external connection, which can cause bandwidth issues. Only use **false** to pull directly from a Red Hat Satellite Server hosting the container images. 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.
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	Use the value of specified container image metadata labels to create a tag for every image and pull that tagged image. For example, if you set **tag_from_label: {version}-{release}**, director uses the **version** and **release** labels to construct a new tag. For one container, **version** might be set to 16.2.3 and **release** might be set to **5.161**, which results in the tag 16.2.3-5.161. Director uses this parameter only if you have not defined **tag** in the **set** dictionary.



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.
ceph_alertmanager_image **ceph_alertmanager_namespace** **ceph_alertmanager_tag**	The name, namespace, and tag of the Ceph Storage Alert Manager container image.
ceph_grafana_image **ceph_grafana_namespace** **ceph_grafana_tag**	The name, namespace, and tag of the Ceph Storage Grafana container image.
ceph_node_exporter_image **ceph_node_exporter_namespace** **ceph_node_exporter_tag**	The name, namespace, and tag of the Ceph Storage Node Exporter container image.
ceph_prometheus_image **ceph_prometheus_namespace** **ceph_prometheus_tag**	The name, namespace, and tag of the Ceph Storage Prometheus container image.
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.

Key	Description
tag	Sets a specific tag for all images from the source. If not defined, director uses the Red Hat OpenStack Platform version number as the default value. This parameter takes precedence over the **tag_from_label** value.

NOTE

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

6.6. GUIDELINES FOR CONTAINER IMAGE TAGGING

The Red Hat Container Registry uses a specific version format to tag all Red Hat OpenStack Platform container images. This format follows the label metadata for each container, which is **version-release**.

version

Corresponds to a major and minor version of Red Hat OpenStack Platform. These versions act as streams that contain one or more releases.

release

Corresponds to a release of a specific container image version within a version stream.

For example, if the latest version of Red Hat OpenStack Platform is 16.2.3 and the release for the container image is **5.161**, then the resulting tag for the container image is 16.2.3-5.161.

The Red Hat Container Registry also uses a set of major and minor **version** tags that link to the latest release for that container image version. For example, both 16.2 and 16.2.3 link to the latest **release** in the 16.2.3 container stream. If a new minor release of 16.2 occurs, the 16.2 tag links to the latest **release** for the new minor release stream while the 16.2.3 tag continues to link to the latest **release** within the 16.2.3 stream.

The **ContainerImagePrepare** parameter contains two sub-parameters that you can use to determine which container image to download. These sub-parameters are the **tag** parameter within the **set** dictionary, and the **tag_from_label** parameter. Use the following guidelines to determine whether to use **tag** or **tag_from_label**.

- The default value for **tag** is the major version for your OpenStack Platform version. For this version it is 16.2. This always corresponds to the latest minor version and release.

```
parameter_defaults:
  ContainerImagePrepare:
  - set:
      ...
      tag: 16.2
      ...
```

- To change to a specific minor version for OpenStack Platform container images, set the tag to a minor version. For example, to change to 16.2.2, set **tag** to 16.2.2.

```
parameter_defaults:
```

```
ContainerImagePrepare:
- set:
    ...
    tag: 16.2.2
    ...
```

- When you set **tag**, director always downloads the latest container image **release** for the version set in **tag** during installation and updates.
- If you do not set **tag**, director uses the value of **tag_from_label** in conjunction with the latest major version.

```
parameter_defaults:
  ContainerImagePrepare:
  - set:
      ...
      # tag: 16.2
      ...
    tag_from_label: '{version}-{release}'
```

- The **tag_from_label** parameter generates the tag from the label metadata of the latest container image release it inspects from the Red Hat Container Registry. For example, the labels for a certain container might use the following **version** and **release** metadata:

```
"Labels": {
  "release": "5.161",
  "version": "16.2.3",
  ...
}
```

- The default value for **tag_from_label** is **{version}-{release}**, which corresponds to the version and release metadata labels for each container image. For example, if a container image has 16.2.3 set for **version** and 5.161 set for **release**, the resulting tag for the container image is 16.2.3-5.161.
- The **tag** parameter always takes precedence over the **tag_from_label** parameter. To use **tag_from_label**, omit the **tag** parameter from your container preparation configuration.
- A key difference between **tag** and **tag_from_label** is that director uses **tag** to pull an image only based on major or minor version tags, which the Red Hat Container Registry links to the latest image release within a version stream, while director uses **tag_from_label** to perform a metadata inspection of each container image so that director generates a tag and pulls the corresponding image.

6.7. OBTAINING CONTAINER IMAGES FROM PRIVATE REGISTRIES

The **registry.redhat.io** registry requires authentication to access and pull images. To authenticate with **registry.redhat.io** and other private registries, include the **ContainerImageRegistryCredentials** and **ContainerImageRegistryLogin** parameters in your **containers-prepare-parameter.yaml** file.

ContainerImageRegistryCredentials

Some container image registries require authentication to access images. In this situation, use the **ContainerImageRegistryCredentials** parameter in your **containers-prepare-parameter.yaml** environment file. The **ContainerImageRegistryCredentials** parameter uses a set of keys based on the

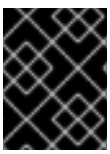
private registry URL. Each private registry URL uses its own key and value pair to define the username (key) and password (value). This provides a method to specify credentials for multiple private registries.

```
parameter_defaults:
  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.

To specify authentication details for multiple registries, set multiple key-pair values for each registry in **ContainerImageRegistryCredentials**:

```
parameter_defaults:
  ContainerImagePrepare:
  - push_destination: true
    set:
      namespace: registry.redhat.io/...
      ...
  - push_destination: true
    set:
      namespace: registry.internalsite.com/...
      ...
  ...
  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.

For more information, see Red Hat Container Registry Authentication .

ContainerImageRegistryLogin

The **ContainerImageRegistryLogin** parameter is used to control whether an overcloud node system needs to log in to the remote registry to fetch the container images. This situation occurs when you want the overcloud nodes to pull images directly, rather than use the undercloud to host images.

You must set **ContainerImageRegistryLogin** to **true** if **push_destination** is set to false or not used for a given strategy.

```
parameter_defaults:
  ContainerImagePrepare:
  - push_destination: false
    set:
      namespace: registry.redhat.io/...
      ...
  ...
  ContainerImageRegistryCredentials:
    registry.redhat.io:
      myuser: 'p@55w0rd!'
  ContainerImageRegistryLogin: true
```

However, if the overcloud nodes do not have network connectivity to the registry hosts defined in **ContainerImageRegistryCredentials** and you set **ContainerImageRegistryLogin** to **true**, the deployment might fail when trying to perform a login. 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** so that the overcloud nodes pull images from the undercloud.

```
parameter_defaults:
  ContainerImagePrepare:
  - push_destination: true
    set:
      namespace: registry.redhat.io/...
      ...
  ...
  ContainerImageRegistryCredentials:
    registry.redhat.io:
      myuser: 'p@55w0rd!'
  ContainerImageRegistryLogin: false
```

6.8. 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 16.2.

Procedure

1. Log in to your undercloud host as the **stack** user.
2. Edit the **containers-prepare-parameter.yaml** file.
3. Add the transitional container parameters to **set** in the **ContainerImagePrepare** parameter. Set the parameters in one of the following ways depending on your type of deployment.
   - If your deployment uses a Red Hat Ceph Storage cluster that was deployed using director, add the following parameters:

     ```
     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 both the **ceph3_*** and **ceph_*** parameters for the transition from Red Hat Ceph Storage 3 to 4.
- If you use Red Hat Satellite Server for container image storage, set the namespaces to the image locations on your Red Hat Satellite Server.

- If your deployment uses an external Ceph Storage cluster, add the following parameters:

```
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
      ceph_namespace: registry.redhat.io/rhceph
      ceph_tag: latest
      ceph_image: rhceph-4-rhel8
      ...
```

- The ***_stein** parameters define the container images for Red Hat OpenStack Platform 15, which the upgrade process uses for database migration.
- The **ceph_*** parameters define the current Red Hat Ceph Storage 4 container images that the overcloud uses.
- If you use Red Hat Satellite Server for container image storage, set the namespaces to the image locations on your 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 16.2, you migrate the overcloud from Open vSwitch to Open Virtual Network (OVN).

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

6.9. 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 16.2.

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.

6.10. 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.

IMPORTANT

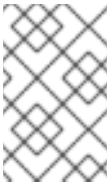
At minimum, you must set the **container_images_file** parameter to the environment file that contains your container image configuration. Without this parameter properly set to the appropriate file, director cannot obtain your container image rule set from the **ContainerImagePrepare** parameter nor your container registry authentication details from the **ContainerImageRegistryCredentials** parameter.

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 in addition to the default **x86_64** architecture.



NOTE

When you enable support for ppc64le, you must also set **ipxe_enabled** to **False**. For more information on configuring your undercloud with multiple CPU architectures, see Configuring a multiple CPU architecture overcloud .

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

Delete temporary files. Set this to **False** to retain the temporary files used during deployment. The temporary files can help you debug the deployment if errors occur.

container_cli

The CLI tool for container management. Leave this parameter set to **podman**. Red Hat Enterprise Linux 8.4 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** 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.

WARNING

RBAC is not supported by every component. The Alarming service (aodh) and Gnocchi do not take secure RBAC rules into account.

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. For PowerPC deployments, or for hybrid PowerPC and x86 deployments, set this value to **false**.

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: em0: <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 em0
       valid_lft 3462sec preferred_lft 3462sec
    inet6 fe80::5054:ff:fe75:2409/64 scope link
       valid_lft forever preferred_lft forever
3: em1: <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 **em0** and the Provisioning NIC uses **em1**, which is currently not configured. In this case, set the **local_interface** to **em1**. 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.
For IPv6, the local IP address prefix length must be **/64** to support both stateful and stateless connections.

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 or YAML 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. For more information about customizing undercloud network interfaces, see Configuring undercloud network interfaces.

networks_file

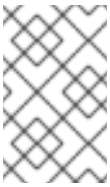
Networks file to override for **heat**.

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.

If the **undercloud_admin_host** is not in the same IP network as the **local_ip**, you must set the **ControlVirtualInterface** parameter to the interface on which you want the admin APIs on the undercloud to listen. By default, the admin APIs listen on the **br-ctlplane** interface. Set the

ControlVirtualInterface parameter in a custom environment file, and include the custom environment file in the **undercloud.conf** file by configuring the **custom_env_files** parameter.

For information about customizing undercloud network interfaces, see Configuring undercloud network interfaces.

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.
If the **undercloud_public_host** is not in the same IP network as the **local_ip**, you must set the **PublicVirtualInterface** parameter to the public-facing interface on which you want the public APIs on the undercloud to listen. By default, the public APIs listen on the **br-ctlplane** interface. Set the **PublicVirtualInterface** parameter in a custom environment file, and include the custom environment file in the **undercloud.conf** file by configuring the **custom_env_files** parameter.

For information about customizing undercloud network interfaces, see Configuring undercloud network interfaces.

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.

IMPORTANT

Director cannot change the IP addresses for a subnet after director creates the subnet.

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 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 to your nodes. If not specified for the subnet, director determines the allocation pools by removing the values set for the **local_ip**, **gateway**, **undercloud_admin_host**, **undercloud_public_host**, and **inspection_iprange** parameters from the subnets full IP range. You can configure non-contiguous allocation pools for undercloud control plane subnets by specifying a list of start and end address pairs. Alternatively, you can use the **dhcp_exclude** option to exclude IP addresses within an IP address range. For example, the following configurations both create allocation pools **172.20.0.100-172.20.0.150** and **172.20.0.200-172.20.0.250**:

Option 1

```
dhcp_start = 172.20.0.100,172.20.0.200
dhcp_end = 172.20.0.150,172.20.0.250
```

Option 2

```
dhcp_start = 172.20.0.100
dhcp_end = 172.20.0.250
dhcp_exclude = 172.20.0.151-172.20.0.199
```

dhcp_exclude

IP addresses to exclude in the DHCP allocation range. For example, the following configuration excludes the IP address **172.20.0.105** and the IP address range **172.20.0.210-172.20.0.219**:

```
dhcp_exclude = 172.20.0.105,172.20.0.210-172.20.0.219
```

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.

6.11. 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_*"
```

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.

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 control plane services at key points. You cannot use any overcloud services to create new resources when these key points are reached. Workloads that are running in the overcloud remain active during the upgrade process, which means instances continue to run during the upgrade of the control plane. During an upgrade of Compute nodes, these workloads can be live migrated to Compute nodes that are already upgraded.

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

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. 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>
 --ansible_ssh_user heat-admin
```

   If you are not using the default **overcloud** stack name, replace ***<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>
```

7.4. VALIDATING THE PRE-UPGRADE REQUIREMENTS

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

For more information about the Red Hat OpenStack Platform (RHOSP) validation framework, see Using the validation framework in the *Director Installation and Usage* guide.

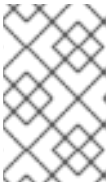
Procedure

1. Source the **stackrc** file.

```
$ source ~/stackrc
```

2. Run the **openstack tripleo validator run** command with the **--group pre-upgrade** option and include the **/usr/libexec/platform-python** python runtime environment:

```
$ openstack tripleo validator run --group pre-upgrade --python-interpreter
/usr/libexec/platform-python -i inventory.yaml
```

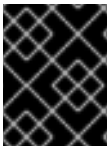


NOTE

Ensure that you include the list of packages that you want to check. You can **use --extra-vars** or **--extra-vars-file** in the command to supply the list through the CLI. For more information, see Command arguments for tripleo validator run .

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

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



IMPORTANT

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

7.5. DISABLING FENCING IN THE OVERCLOUD

Before you upgrade the overcloud, ensure that fencing is disabled.

When you upgrade the overcloud, you upgrade each Controller node individually to retain high availability functionality. If fencing is deployed in your environment, the overcloud might detect certain nodes as disabled and attempt fencing operations, which can cause unintended results.

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

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"
```

   Replace **<controller_ip>** with the IP address of a Controller node. You can find the IP addresses of your Controller nodes with the **openstack server list** command.

4. In the **fencing.yaml** environment file, set the **EnableFencing** parameter to **false** to ensure that fencing stays disabled during the upgrade process.

Additional Resources

- Fencing Controller nodes with STONITH

7.6. CHECKING CUSTOM PUPPET PARAMETERS

If you use the **ExtraConfig** interfaces for customizations of Puppet parameters, Puppet might report duplicate declaration errors during the upgrade. This is due to changes in the interfaces provided by the puppet modules themselves.

This procedure shows how to check for any custom **ExtraConfig** hieradata parameters in your environment files.

Procedure

1. Select an environment file and the check if it has an **ExtraConfig** parameter:

```
$ grep ExtraConfig ~/templates/custom-config.yaml
```

2. If the results show an **ExtraConfig** parameter for any role (e.g. **ControllerExtraConfig**) in the chosen file, check the full parameter structure in that file.

3. If the parameter contains any puppet Hierdata with a **SECTION/parameter** syntax followed by a **value**, it might have been been replaced with a parameter with an actual Puppet class. For example:

```
parameter_defaults:
  ExtraConfig:
    neutron::config::dhcp_agent_config:
      'DEFAULT/dnsmasq_local_resolv':
        value: 'true'
```

4. Check the director's Puppet modules to see if the parameter now exists within a Puppet class. For example:

```
$ grep dnsmasq_local_resolv
```

   If so, change to the new interface.

5. The following are examples to demonstrate the change in syntax:

   - Example 1:

     ```
     parameter_defaults:
       ExtraConfig:
         neutron::config::dhcp_agent_config:
           'DEFAULT/dnsmasq_local_resolv':
             value: 'true'
     ```

     Changes to:

     ```
     parameter_defaults:
       ExtraConfig:
         neutron::agents::dhcp::dnsmasq_local_resolv: true
     ```

   - Example 2:

     ```
     parameter_defaults:
       ExtraConfig:
         ceilometer::config::ceilometer_config:
           'oslo_messaging_rabbit/rabbit_qos_prefetch_count':
             value: '32'
     ```

     Changes to:

     ```
     parameter_defaults:
       ExtraConfig:
         oslo::messaging::rabbit::rabbit_qos_prefetch_count: '32'
     ```

CHAPTER 8. CONFIGURING THE OVERCLOUD FOR A LEAPP UPGRADE

The long-life Red Hat OpenStack Platform (RHOSP) upgrade requires a base operating system upgrade from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8. Red Hat Enterprise Linux 7 uses the Leapp utility to perform the upgrade to Red Hat Enterprise Linux 8. For more information about Leapp and its dependencies, see Preparing a RHEL 7 system for the upgrade .

The overcloud upgrade framework automatically runs the leapp upgrade. To ensure a successful RHOSP upgrade, it is recommended that you manually run a pre-upgrade report to identify and resolve any potential problems. Run pre-upgrade reports for at least one host of each of the Compute, Controller, and Ceph Storage roles. For more information about Leapp pre-upgrade reports, see Reviewing the pre-upgrade report.

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:
  UpgradeLeappCommandOptions: " --enablerepo rhel-8-for-x86_64-baseos-eus-rpms --enablerepo rhel-8-for-x86_64-appstream-eus-rpms --enablerepo rhel-8-for-x86_64-highavailability-eus-rpms --enablerepo ansible-2.9-for-rhel-8-x86_64-rpms --enablerepo fast-datapath-for-rhel-8-x86_64-rpms "
```

   For more information about the upgrade parameters that you can configure in the environment file, see Upgrade parameters.

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

8.2. UPGRADE PARAMETERS

You can modify the behavior of the upgrade process with 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**.
UpgradeLeappPostRebootDelay	Maximum (seconds) to wait for machine to reboot and respond to a test command. The default value is **120**.
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. 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 an **eth** prefix. These NIC names are usually unpredictable in terms of NIC assignments.

You can run the **playbook-nics.yaml** playbook to rename NIC names to use the **em** NIC prefix. You can also set a different NIC prefix, by modifying the **prefix** variable when running the playbook. However, the NIC changes are only applied after the Leapp upgrade process completes and the node is rebooted.

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
```

The NIC changes are only applied after the Leapp upgrade process completes and the node is rebooted.

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

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.

IMPORTANT

If any overcloud nodes in your deployment are dedicated Object Storage (swift) nodes, you must copy the default **roles_data.yaml** file and edit **ObjectStorage** to remove the following line: **deprecated_server_resource_name: 'SwiftStorage'**

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**	The OpenStack Telemetry services are deprecated in favour of Service Telemetry Framework (STF) for metrics and monitoring. The legacy telemetry services are only available in RHOSP 16.2 to help facilitate the transition to STF and will be removed in a future version of RHOSP. **NOTE** If you use auto scaling, do not remove these services from your Controller nodes.

Service	Reason
OS::TripleO::Services::PankoApi	The OpenStack Telemetry services are deprecated in favour of Service Telemetry Framework (STF) for metrics and monitoring. The legacy telemetry services are only available in RHOSP 16.2 to help facilitate the transition to STF and will be removed in a future version of RHOSP. **NOTE** If you use CloudForms, do not remove these services from your Controller nodes.
OS::TripleO::Services::CeilometerApi **OS::TripleO::Services::CeilometerCollector** **OS::TripleO::Services::CeilometerExpirer** **OS::TripleO::Services::MongoDb**	These services are no longer supported. Remove these services from your Controller nodes.
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::NeutronCorePluginPlumgrid **OS::TripleO::Services::NeutronCorePluginMidonet**	Deprecated plug-ins for OpenStack Networking (neutron).
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 removed.
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.

Service	Reason
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.
OS::TripleO::Services::gnocchi_metricd **OS::TripleO::Services::gnocchi_statsd** **OS::TripleO::Services::gnocchi_api**	Gnocchi is deprecated and will be removed in a later RHOSP release.
OS::TripleO::Services::panko_api_cron	Panko is deprecated and will be removed in a later RHOSP release.

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).

Service	Reason

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 16.2 reside in a new location within **/usr/share/openstack-tripleo-heat-templates/**:

Red Hat OpenStack Platform 13	Red Hat OpenStack Platform 16.2
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

To configure access to the undercloud registry, add the control plane host name of the undercloud and the IP address of the undercloud on the provisioning network 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 control plane host name of the undercloud and the IP address of the undercloud on the provisioning network:

```
parameter_defaults:
  DockerInsecureRegistryAddress:
  - undercloud.ctlplane.localdomain:8787
  - 192.168.24.1:8787
  ...
```

   You must also append the port number of the overcloud registry to the host name and IP address values. 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

Filter	Status
RamFilter	Removed
DiskFilter	Removed
RetryFilter	Deprecated

IMPORTANT

Avoid using filters marked **Deprecated**. Red Hat OpenStack Platform 16.2 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 16.2 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. CONFIGURING THE INSTANCE SERIAL NUMBER

In Red Hat OpenStack Platform 13, the serial number for instances that are stored in the virtual BIOS of the virtual machine is based on the serial number of the host.

In Red Hat OpenStack Platform 16.2, the serial number for instances that are stored in the virtual BIOS of the virtual machine is based on the UUID of the instance by default.

If you want to retain the behavior of your Red Hat OpenStack Platform 13 deployment when you upgrade to Red Hat OpenStack Platform 16.2, you must configure **[libvirt]sysinfo_serial**.

Procedure

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

```
[stack@director ~]$ source ~/stackrc
```

3. Open an environment file.
4. Add the following configuration to your environment file to specify that the instance serial number is based on the serial number of the host:

```
parameter_defaults:
  <Role>ExtraConfig:
    nova::compute::libvirt::sysinfo_serial: auto
```

5. Save the updates to your environment file.
6. Add the file to your overcloud upgrade and deployment commands.

9.7. UPDATING THE SSL/TLS CONFIGURATION

Remove the **NodeTLSData** resource from the **resource_registry** to update your SSL/TLS configuration.

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. Save the SSL/TLS public endpoint file.

9.8. 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.

9.9. CONSIDERATIONS WHEN RETAINING LEGACY TELEMETRY SERVICES

In Red Hat OpenStack Platform (RHOSP) 16.2, the OpenStack Telemetry components are deprecated in favor of the Service Telemetry Framework (STF), therefore the legacy telemetry components are not enabled after the upgrade.

If you use auto scaling or CloudForms services, you must retain the legacy telemetry services.

To retain legacy RHOSP 13 telemetry services, include the **/usr/share/openstack-tripleo-heat-templates/environments/enable-legacy-telemetry.yaml** environment file when you run the **openstack overcloud upgrade prepare** and **openstack overcloud upgrade converge** commands.

You must also include the **enable-legacy-telemetry.yaml** environment file every time you update the overcloud after the upgrade.

The legacy telemetry services are only available to help facilitate the transition to STF and will be removed in a future version of RHOSP.

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

Red Hat OpenStack Platform 16.2 uses Ansible-based methods to register overcloud nodes to the Red Hat Customer Portal.

10.1. RED HAT SUBSCRIPTION MANAGER (RHSM) COMPOSABLE SERVICE

You can use the **rhsm** composable service 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 you can use to define multiple sub-parameters relevant to your registration:

```
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: 8.4
```

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

10.2. RHSMVARS SUB-PARAMETERS

Use the following sub-parameters as part of the **RhsmVars** parameter when you configure the **rhsm** composable service. For more information about the Ansible parameters that are available, see the role documentation.

rhsm	Description
rhsm_method	Choose the registration method. Either **portal**, **satellite**, or **disable**.
rhsm_org_id	The organization that you want to use for registration. To locate this ID, run **sudo subscription-manager orgs** from the undercloud node. Enter your Red Hat credentials at the prompt, 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 that you want to use. Use this parameter if you do not want to auto-attach subscriptions. To locate this ID, run **sudo subscription-manager list --available --all --matches="Red Hat OpenStack"** from the undercloud node, and use the resulting**Pool ID** value.
rhsm_activation_key	The activation key that you want to use for registration.
rhsm_autosubscribe	Use this parameter to attach compatible subscriptions to this system automatically. Set the value to **true** to enable this feature.
rhsm_baseurl	The base URL for obtaining content. The default URL is the Red Hat Content Delivery Network. If you use 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 you use a Satellite server, change this value to your Satellite server hostname.
rhsm_repos	A list of repositories that you want to enable.
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 8.4 for Red Hat OpenStack Platform
rhsm_rhsm_proxy_hostname	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_password	The password to access the HTTP proxy.

IMPORTANT

You can use **rhsm_activation_key** and **rhsm_repos** together only if **rhsm_method** is set to **portal**. If **rhsm_method** is set to 'satellite', you can only use either **rhsm_activation_key** or **rhsm_repos**.

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

To help transition your details from the **rhel-registration** method to the **rhsm** method, use the following table to map your parameters and values.

rhel-registration	rhsm / RhsmVars
rhel_reg_method	rhsm_method
rhel_reg_org	rhsm_org_id
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

Create an environment file that enables and configures the **rhsm** composable service. Director uses this environment file to register and subscribe your nodes.

Procedure

1. Create an environment file named **templates/rhsm.yml** 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: 8.4
```

- 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 named **templates/rhsm.yml** 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
        - openstack-16.2-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: 8.4
  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
      - openstack-16.2-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: 8.4
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-16.2-deployment-tools-for-rhel-8-x86_64-rpms
    rhsm_username: "myusername"
    rhsm_password: "p@55w0rd!"
    rhsm_org_id: "1234567"
    rhsm_pool_ids: "68790a7aa2dc9dc50a9bc39fabc55e0d"
    rhsm_method: "portal"
    rhsm_release: 8.4
```

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** parameters each use a separate **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 SERVER

Red Hat OpenStack Platform 16.2 uses Ansible-based methods to register overcloud nodes to Red Hat Satellite Server 6.

11.1. RED HAT SUBSCRIPTION MANAGER (RHSM) COMPOSABLE SERVICE

You can use the **rhsm** composable service 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 you can use to define multiple sub-parameters relevant to your registration:

```
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: 8.4
```

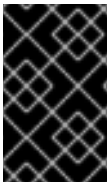
You can also use the **RhsmVars** parameter in combination with role-specific parameters, for example, **ControllerParameters**, to provide flexibility when enabling specific repositories for different nodes types.

11.2. RHSMVARS SUB-PARAMETERS

Use the following sub-parameters as part of the **RhsmVars** parameter when you configure the **rhsm** composable service. For more information about the Ansible parameters that are available, see the role documentation.

rhsm	Description
rhsm_method	Choose the registration method. Either **portal**, **satellite**, or **disable**.
rhsm_org_id	The organization that you want to use for registration. To locate this ID, run **sudo subscription-manager orgs** from the undercloud node. Enter your Red Hat credentials at the prompt, 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 that you want to use. Use this parameter if you do not want to auto-attach subscriptions. To locate this ID, run **sudo subscription-manager list --available --all --matches="Red Hat OpenStack"** from the undercloud node, and use the resulting**Pool ID** value.
rhsm_activation_key	The activation key that you want to use for registration.
rhsm_autosubscribe	Use this parameter to attach compatible subscriptions to this system automatically. Set the value to **true** to enable this feature.
rhsm_baseurl	The base URL for obtaining content. The default URL is the Red Hat Content Delivery Network. If you use 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 you use a Satellite server, change this value to your Satellite server hostname.
rhsm_repos	A list of repositories that you want to enable.
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 8.4 for Red Hat OpenStack Platform
rhsm_rhsm_proxy_hostname	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_password	The password to access the HTTP proxy.



IMPORTANT

You can use **rhsm_activation_key** and **rhsm_repos** together only if **rhsm_method** is set to **portal**. If **rhsm_method** is set to 'satellite', you can only use either **rhsm_activation_key** or **rhsm_repos**.

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

To help transition your details from the **rhel-registration** method to the **rhsm** method, use the following table to map your parameters and values.

rhel-registration	rhsm / RhsmVars
rhel_reg_method	rhsm_method
rhel_reg_org	rhsm_org_id
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 SERVER

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 named **templates/rhsm.yml** 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: 8.4
```

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

Prerequisites

- Create a Satellite Server activation key that is linked to repositories for Red Hat OpenStack Platform 16.2 and Red Hat Enterprise Linux 8.4.
- Generate an Ansible inventory file for your overcloud nodes.
- On your Satellite Server, create and promote a content view for Red Hat OpenStack Platform 16.2 upgrade and include the repositories that required for the upgrade. For more information, see Red Hat Satellite Server 6 considerations .
- If you use a Ceph subscription and have configured director to use the **overcloud-minimal** image for Ceph storage nodes, on your Satellite Server you must create a Content View and add the following Red Hat Enterprise Linux (RHEL) 8.4 repositories to it:
  - Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)

    ```
    rhel-8-for-x86_64-appstream-rpms
    x86_64 8.4
    ```

  - Red Hat Enterprise Linux 8 for x86_64 - BaseOS (RPMs)

    ```
    rhel-8-for-x86_64-baseos-rpms
    x86_64 8.4
    ```

    For more information, see Importing Red Hat Content and Managing Content Views in the *Red Hat Satellite Content Management Guide* .

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
    redhat_subscription:
      activationkey: "osp16-key"
      org_id: "ACME"
      server_hostname: "satellite.example.com"
      rhsm_baseurl: "https://satellite.example.com/pulp/repos"
      force_register: True
```

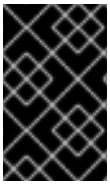
Modify the **redhat_subscription** variables to suit your Satellite Server.

3. Run the playbook:

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

CHAPTER 12. PREPARING FOR A DIRECTOR-DEPLOYED CEPH STORAGE UPGRADE

If your deployment uses a director-deployed Red Hat Ceph Storage cluster, you must complete the procedures included in this section.



IMPORTANT

RHOSP 16.2 is supported on RHEL 8.4. However, hosts that are mapped to the Ceph Storage role update to the latest major RHEL release. For more information, see Red Hat Ceph Storage: Supported configurations.

NOTE

If you are upgrading with external Ceph deployments, you must skip the procedures included in this section and continue to Chapter 13, *Preparing for upgrading with external Ceph deployments*.

The upgrade process maintains the use of Red Hat Ceph Storage 3 containerized services during the upgrade to Red Hat OpenStack Platform 16.2. After you complete the Red Hat OpenStack Platform 16.2 upgrade, you upgrade the Ceph Storage services to Red Hat Ceph Storage 4.

You cannot provision new shares with the Shared File Systems service (manila) until you complete both the Red Hat OpenStack Platform 16.2 upgrade and the Ceph Storage services upgrade to Red Hat Ceph Storage 4.

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

The director-deployed 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 8.4. 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 8.4. 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 16.2 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 16.2 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.

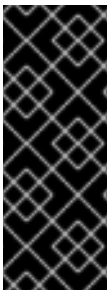
Procedure

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 16.2 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 16.2 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

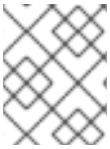
1. Log in to the node that is running the **ceph-mon** service. This node is usually a Controller node or a standalone Ceph Monitor node.
2. Enter the following command to view the status of the Ceph cluster:

```
$ 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. PREPARING FOR UPGRADING WITH EXTERNAL CEPH DEPLOYMENTS

If you are upgrading with external Ceph deployments, you must complete the procedures included in this section.



NOTE

If your deployment does not use an external Ceph Storage cluster, you must skip the procedures included in this section and continue to the next section.

13.1. INSTALLING CEPH-ANSIBLE

If you are upgrading with external Ceph deployments, you must complete this procedure.

The **ceph-ansible** package is required when you use Ceph Storage with Red Hat OpenStack Platform.

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
```

13.2. SETTING THE CEPH-ANSIBLE REPOSITORY

The Red Hat OpenStack Platform 16.2 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 16.2 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.

CHAPTER 14. UPDATING NETWORK CONFIGURATION

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

14.1. 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 \
            --tht-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. Optional: If you use a spine-leaf network topology for your RHOSP environment, check the **roles_data.yaml** file and ensure that it uses the correct role names for the NIC templates for your deployment. The script uses the value of the **deprecated_nic_config_name** parameter in the **roles_data.yaml** file.

6. 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.
```

14.2. 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**

NOTE

When you include the **neutron-ovs.yaml** environment file, check if the **neutron-ovs-dvr.yaml** environment file is included in your environment file collection. You must include the **neutron-ovs.yaml** environment file before the **neutron-ovs-dvr.yaml** file, to avoid failures during the upgrade.

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.

Troubleshooting OVS compatibility

If the upgrade process fails because the parameters defined in the **neutron-ovs.yaml** file are overwriting the parameters defined in the **neutron-ovs-dvr.yaml**, change the order in which you include these files and run the **openstack overcloud upgrade prepare** and **openstack overcloud upgrade run** again on the affected nodes. If one of the affected nodes is a Compute node, remove the **openstack-neutron*** packages from that node.

14.3. MAINTAINING COMPOSABLE NETWORK COMPATIBILITY DURING THE UPGRADE

The **network_data** file format for Red Hat OpenStack Platform 16.2 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 16.2, 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 16.2, use the Red Hat OpenStack Platform 16.2 **network_data** file format. For more information about Red Hat OpenStack Platform 16.2 composable network syntax, see Custom composable networks.

CHAPTER 15. PREPARING NETWORK FUNCTIONS VIRTUALIZATION (NFV)

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

15.1. 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 16.2. 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 16.2 **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 16.2:
  - Open vSwitch (OVS) networking and SR-IOV: **/usr/share/openstack-tripleo-heat-templates/environments/services/neutron-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 16.2, 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-sriov.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/services/neutron-ovs-dpdk.yaml \
  ...
```

NOTE

During an upgrade, you can migrate instances between RHOSP 13 and RHOSP 16.2 Compute nodes only when the RHOSP 13 Compute nodes are in the **hybrid** state. For more information, see Migration constraints in the *Configuring the Compute Service for Instance Creation* guide.

There is an additional migration constraint for NFV workloads: you cannot live migrate instances from OVS-DPDK Compute nodes during an upgrade. Alternatively, you can cold migrate instances from OVS-DPDK Compute nodes during an upgrade.

CHAPTER 16. FINAL REVIEW BEFORE UPGRADE

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

16.1. CUSTOM FILES TO INCLUDE IN YOUR DEPLOYMENT

If any overcloud nodes in your deployment are dedicated Object Storage (swift) nodes, you must copy the default **roles_data.yaml** file and edit **ObjectStorage** to remove **deprecated_server_resource_name: 'SwiftStorage'**. Then use the **--roles-file** option to pass the file to the **openstack overcloud upgrade prepare** or the **openstack overcloud upgrade converge** command.

16.2. 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 (RHOSP) 16.2.

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 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 16.2 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 16.2 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**

16.3. 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**

16.4. UPGRADE CHECKLIST

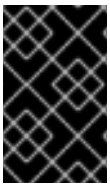
Use the following checklist to determine your readiness to upgrade the overcloud:

Item	Complete
Validated a working overcloud.	Y / N
Performed a Relax-and-Recover (ReaR) backup of the overcloud control plane. For more information, see Red Hat OpenStack Platform 13 Undercloud and Control Plane Back Up and Restore.	Y / N
Created a backup of the database that runs on the undercloud node. For more information, see Creating a database backup of the undercloud node in the *Red Hat OpenStack Platform 16.2 Backing up and restoring the undercloud and control plane nodes* guide.	Y / N
Implemented all preparation for Leapp, including: • Leapp environment file • All overcloud nodes are now using predictable NIC names.	Y / N
Updated your registration details to Red Hat OpenStack Platform 16.2 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 16.2.	Y / N
Optional: If your deployment includes dedicated Object Storage (swift) nodes: Copied the **roles_data.yaml** file, removed **deprecated_server_resource_name: 'SwiftStorage'**, and passed the file to the **openstack overcloud upgrade prepare** or the **openstack overcloud upgrade converge** command.	Y / N

Item	Complete
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

CHAPTER 17. 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.

17.1. 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 16.2 overcloud plan and your updated environment files. This command functions similar to the **openstack overcloud deploy** command and uses many of the same options.

17.2. OPENSTACK OVERCLOUD UPGRADE RUN

This command performs the upgrade process. Director creates a set of Ansible playbooks based on the new OpenStack Platform 16.2 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 16.2.

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 16.2 containers on Compute nodes to facilitate workload migration during the upgrade.

17.3. 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 16.2 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

- If your deployment uses a Red Hat Ceph Storage cluster that was deployed using director, you can use the following tags:

  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.

- If you are upgrading with external Ceph deployments, you can skip the tasks that use the **ceph** and **ceph_systemd** tags.

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.

17.4. 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 16.2 overcloud plan and your updated environment files. This process ensures that the resulting overcloud matches the configuration of a new OpenStack Platform 16.2 overcloud. This command is similar to the **openstack overcloud deploy** command and uses many of the same options.

17.5. 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 in order to start a database transfer and launch temporary containers that facilitate migration during the transition from Red Hat OpenStack 13 to 16.2. During the bootstrap Controller node upgrade process, a new Pacemaker cluster is created and new Red Hat OpenStack 16.2 containers are started on the node, while the remaining Controller nodes are still running on Red Hat OpenStack 13. After upgrading the bootstrap node, you must upgrade each additional node with Pacemaker services and ensure that 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

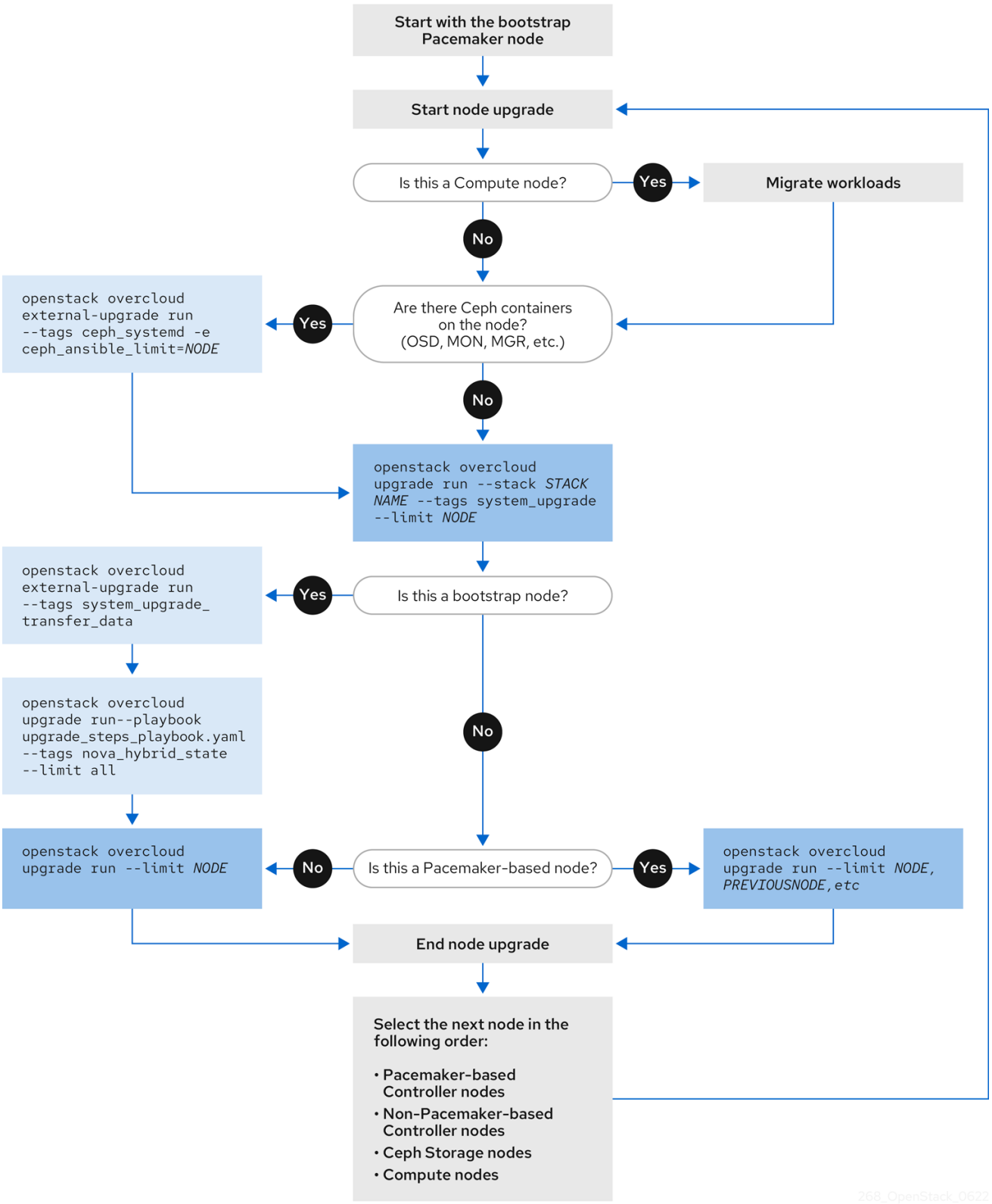
- **Is the node a Compute node?**If the node does contain Compute services, you must migrate virtual machines from the node to ensure maximum availability. A Compute node in this situation includes any node designed to host virtual machines. This definition includes the follow Compute nodes types:
 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


Start with the bootstrap Pacemaker node
Start node upgrade
Is this a Compute node?
Yes
Migrate workloads
No
Are there Ceph containers on the node? (OSD, MON, MGR, etc.)
Yes
openstack overcloud external-upgrade run --tags ceph_systemd -e ceph_ansible_limit=NODE
No
openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit NODE
Is this a bootstrap node?
Yes
openstack overcloud external-upgrade run --tags system_upgrade_ transfer_data
openstack overcloud upgrade run--playbook upgrade_steps_playbook.yaml --tags nova_hybrid_state --limit all
No
Is this a Pacemaker-based node?
Yes
openstack overcloud upgrade run --limit NODE, PREVIOUSNODE,etc
No
openstack overcloud upgrade run --limit NODE
End node upgrade
Select the next node in the following order:
• Pacemaker-based Controller nodes
• Non-Pacemaker-based Controller nodes
• Ceph Storage nodes
• Compute nodes
268_OpenStack_0622

CHAPTER 18. 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

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 16.2
- 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 CONTROLLER NODES

To upgrade all the Controller nodes to Red Hat OpenStack Platform (RHOSP) 16.2, you must upgrade each Controller node starting with the bootstrap Controller node.

If your deployment uses a Red Hat Ceph Storage cluster that was deployed using director, follow the procedure in Upgrading Controller nodes with director-deployed Ceph Storage .

During the bootstrap Controller node upgrade process, a new Pacemaker cluster is created and new RHOSP 16.2 containers are started on the node, while the remaining Controller nodes are still running on RHOSP 13.

After upgrading the bootstrap node, you must upgrade each additional node with Pacemaker services and ensure that each node joins the new Pacemaker cluster started with the bootstrap node. For more information, see Overcloud node upgrade workflow.

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

2. On the undercloud node, identify the bootstrap Controller node:

```
$ tripleo-ansible-inventory --list [--stack <stack_name>] |jq .overcloud_Controller.hosts[0]
```

- Replace **<stack_name>** with the name of your stack.

3. Upgrade the bootstrap Controller node:
   a. Perform a Leapp upgrade of the operating system on the bootstrap Controller node:

```
$ openstack overcloud upgrade run [--stack <stack>] --tags system_upgrade --limit
<bootstrap_controller_node>
```

- Replace **<bootstrap_controller_node>** with the host name of the bootstrap Controller node in your environment, for example, **overcloud-controller-0**.
- If you are not using the default overcloud stack name, **overcloud**, include the **--stack** optional argument and replace **<stack>** with the name of your overcloud stack.

The bootstrap Controller node is rebooted as part of the Leapp upgrade.

b. Copy the latest version of the database from an existing node to the bootstrap node:

```
$ openstack overcloud external-upgrade run [--stack <stack>] --tags
system_upgrade_transfer_data
```

IMPORTANT

This command causes an outage on the control plane. You cannot perform any standard operations on the overcloud until the RHOSP upgrade is complete and the control plane is active again.

c. Launch temporary 16.2 containers on Compute nodes to help facilitate workload migration when you upgrade Compute nodes at a later step:

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

d. Upgrade the overcloud with no tags:

```
$ openstack overcloud upgrade run --stack <stack> --limit <bootstrap_controller_node>
```

e. Verify that after the upgrade, the new Pacemaker cluster is started and that the control plane services such as **galera**, **rabbit**, **haproxy**, and **redis** are running:

```
$ sudo pcs status
```

4. Upgrade the next Controller node:

a. Verify that the old cluster is no longer running:

```
$ sudo pcs status
```

An error similar to the following is displayed when the cluster is not running:

```
Error: cluster is not currently running on this node
```

b. Perform a Leapp upgrade of the operating system on the Controller node:

```
$ openstack overcloud upgrade run --stack <stack> --tags system_upgrade --limit
<controller_node>
```

- Replace **<controller_node>** with the host name of the Controller node to upgrade, for example, **overcloud-controller-1**.
  The Controller node is rebooted as a part of the Leapp upgrade.

c. Upgrade the Controller node, adding it to the previously upgraded nodes in the new Pacemaker cluster:

```
$ openstack overcloud upgrade run --stack <stack> --limit
<bootstrap_controller_node,controller_node_1,controller_node_n>
```

- Replace **<bootstrap_controller_node,controller_node_1,controller_node_n>** with a comma-separated list of the Controller nodes that you have upgraded so far, and the additional Controller node that you want to add to the Pacemaker cluster, for example, **overcloud-controller-0,overcloud-controller-1, overcloud-controller-2**.

18.3. UPGRADING CONTROLLER NODES WITH DIRECTOR-DEPLOYED CEPH STORAGE

If your deployment uses a Red Hat Ceph Storage cluster that was deployed using director, you must complete this procedure.

To upgrade all the Controller nodes to OpenStack Platform 16.2, you must upgrade each Controller node starting with the bootstrap Controller node.

During the bootstrap Controller node upgrade process, a new Pacemaker cluster is created and new Red Hat OpenStack 16.2 containers are started on the node, while the remaining Controller nodes are still running on Red Hat OpenStack 13.

After upgrading the bootstrap node, you must upgrade each additional node with Pacemaker services and ensure that each node joins the new Pacemaker cluster started with the bootstrap node. For more information, see Overcloud node upgrade workflow.

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.

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

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

```
$ tripleo-ansible-inventory --list [--stack <stack_name>] |jq .overcloud_Controller.hosts[0]
```

   - Optional: Replace **<stack_name>** with the name of the stack. If not specified, the default is **overcloud**.

3. Upgrade the bootstrap Controller node:

   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-controller-0
```

   Replace **<stack_name>** with the name of your stack.

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:

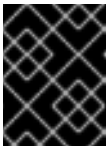
```
$ openstack overcloud upgrade run [--stack <stack_name>] --playbook upgrade_steps_playbook.yaml --tags nova_hybrid_state --limit all
```

This command launches temporary 16.2 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.

f. Verify that after the upgrade, the new Pacemaker cluster is started and that the control plane services such as galera, rabbit, haproxy, and redis are running:

```
$ sudo pcs status
```

4. Upgrade the next Controller node:

a. Verify that the old cluster is no longer running:

```
$ sudo pcs status
```

An error similar to the following is displayed when the cluster is not running:

```
Error: cluster is not currently running on this node
```

b. 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.

c. 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.

d. 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. Verify that the old cluster is no longer running:

```
$ sudo pcs status
```

An error similar to the following is displayed when the cluster is not running:

```
Error: cluster is not currently running on this node
```

b. 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.

c. 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.

d. 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.

18.4. UPGRADING THE OPERATING SYSTEM FOR CEPH STORAGE NODES

If your deployment uses a Red Hat Ceph Storage cluster that was deployed using director, you must 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. 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.

3. 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.

4. 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.

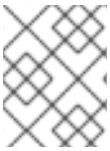
c. Run the upgrade command with no tags:

```
$ 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.

18.5. UPGRADING COMPUTE NODES

Upgrade all the Compute nodes to OpenStack Platform 16.2.



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:

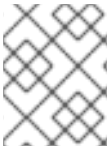
```
$ openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit
overcloud-compute-0,overcloud-compute-1,overcloud-compute-2
```

Then perform the standard OpenStack service upgrade:

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

18.6. 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 16.2.



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
```

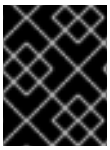
2. Edit the **containers-prepare-parameter.yaml** file and remove the following parameters and their values:
   - **ceph3_namespace**
   - **ceph3_tag**
   - **ceph3_image**
   - **name_prefix_stein**
   - **name_suffix_stein**
   - **namespace_stein**
   - **tag_stein**
3. To re-enable fencing in your overcloud, set the **EnableFencing** parameter to **true** in the **fencing.yaml** environment file.
4. 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 (**fencing.yaml**) with the **EnableFencing** parameter set to **true**.
   - 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.

5. 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 EXTERNAL CEPH DEPLOYMENTS

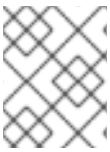
This scenario contains an example upgrade process for an overcloud environment with external Ceph deployments, which includes the following node types:

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

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 16.2
- 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 WITH EXTERNAL CEPH DEPLOYMENTS

If you are upgrading with external Ceph deployments, you must complete this procedure.

To upgrade all the Controller nodes to OpenStack Platform 16.2, you must upgrade each Controller node starting with the bootstrap Controller node.

During the bootstrap Controller node upgrade process, a new Pacemaker cluster is created and new Red Hat OpenStack 16.2 containers are started on the node, while the remaining Controller nodes are still running on Red Hat OpenStack 13.

After upgrading the bootstrap node, you must upgrade each additional node with Pacemaker services and ensure that each node joins the new Pacemaker cluster started with the bootstrap node. For more information, see Overcloud node upgrade workflow.

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.

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

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

```
$ tripleo-ansible-inventory --list [--stack <stack_name>] |jq .overcloud_Controller.hosts[0]
```

- Optional: Replace **<stack_name>** with the name of the stack. If not specified, the default is **overcloud**.

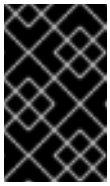
3. Upgrade the bootstrap Controller node:

   a. 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.

   b. 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.

   c. 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 16.2 containers on Compute nodes to help facilitate workload migration when you upgrade Compute nodes at a later step.

   d. 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.

   e. Verify that after the upgrade, the new Pacemaker cluster is started and that the control plane services such as galera, rabbit, haproxy, and redis are running:

   ```
   $ sudo pcs status
   ```

4. Upgrade the next Controller node:

   a. Verify that the old cluster is no longer running:

      ```
      $ sudo pcs status
      ```

      An error similar to the following is displayed when the cluster is not running:

      ```
      Error: cluster is not currently running on this node
      ```

   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. Verify that the old cluster is no longer running:

      ```
      $ sudo pcs status
      ```

      An error similar to the following is displayed when the cluster is not running:

      ```
      Error: cluster is not currently running on this node
      ```

   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

Upgrade all the Compute nodes to OpenStack Platform 16.2.

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
```

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 16.2.

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. To re-enable fencing in your overcloud, set the **EnableFencing** parameter to **true** in the **fencing.yaml** environment file.
4. 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 (**fencing.yaml**) with the **EnableFencing** parameter set to **true**.
   - 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.

5. Wait until the stack synchronization completes.

IMPORTANT

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

CHAPTER 20. SPEEDING UP AN OVERCLOUD UPGRADE

To speed up the overcloud upgrade process, you can upgrade the control plane incrementally, or upgrade all of your nodes at once.

Upgrade incrementally

You can upgrade 1/3 of your control plane at a time. After you upgrade the first 1/3 of the control plane, you can move your environment into mixed-mode where the control plane APIs are running and the cloud is operational. High availability operational performance can be resumed only after the entire control plane has been upgraded.

Sections 20.1 to 20.4 contain an example upgrade process for an overcloud environment that includes the following node types with composable roles:

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

Upgrade the entire overcloud at once

By upgrading the whole overcloud at once, you complete the upgrade much faster than upgrading incrementally. Note that this option requires you to take your control plane and data plane offline.

To upgrade the entire overcloud, see Section 20.5, “Upgrading the entire overcloud at once” .

20.1. 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 16.2
- 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
```

20.2. UPGRADING THE CONTROL PLANE NODES

To upgrade the control plane nodes in your environment to OpenStack Platform 16.2, you must upgrade 1/3 of your control plane nodes at a time, starting with the bootstrap nodes.

During the bootstrap Controller node upgrade process, a new Pacemaker cluster is created and new Red Hat OpenStack 16.2 containers are started on the node, while the remaining Controller nodes continue to run on Red Hat OpenStack 13.

This example includes the following node types with composable roles. The control plane nodes are named using the default **overcloud-*ROLE-NODEID*** convention:

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

- **overcloud-database-1**
- **overcloud-database-2**
- **overcloud-networker-0**
- **overcloud-networker-1**
- **overcloud-networker-2**
- **overcloud-ceph-0**
- **overcloud-ceph-1**
- **overcloud-ceph-2**

Replace these values for your own node names where applicable.

After you upgrade the **overcloud-controller-0**, **overcloud-database-0**, **overcloud-networker-0**, and **overcloud-ceph-0** bootstrap nodes, which comprise the first 1/3 of your control plane nodes, you must upgrade each additional 1/3 of the nodes with Pacemaker services and ensure that each node joins the new Pacemaker cluster started with the bootstrap node. Therefore, you must upgrade **overcloud-controller-1**, **overcloud-database-1**, **overcloud-networker-1**, and **overcloud-ceph-1** before you upgrade **overcloud-controller-2**, **overcloud-database-2**, **overcloud-networker-2**, and **overcloud-ceph-2**.

Procedure

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

```
$ source ~/stackrc
```

3. On the undercloud node, identify the bootstrap Controller node:

```
$ tripleo-ansible-inventory --list [--stack <stack_name>] |jq .overcloud_Controller.hosts[0]
```

   - Optional: Replace **<stack_name>** with the name of the stack. If not specified, the default is **overcloud**.

4. Upgrade the **overcloud-controller-0**, **overcloud-database-0**, **overcloud-networker-0**, and **overcloud-ceph-0** control plane nodes:

   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-controller-0,overcloud-database-0,overcloud-networker-
0,overcloud-ceph-0
```

   This command performs the following actions:

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

This step is a preliminary measure to prepare the Ceph Storage services for the Leapp upgrade.

b. Perform a Leapp upgrade of the operating system on each control plane node:

```
$ openstack overcloud upgrade run --stack <stack_name> --tags system_upgrade --limit
overcloud-controller-0,overcloud-database-0,overcloud-networker-0,overcloud-ceph-0
```

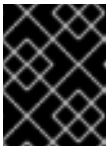
The control plane nodes are rebooted as a part of the Leapp upgrade.

IMPORTANT

If the Ceph node upgrade fails, ensure that **controller-0** has finished upgrading before you proceed with the rest of the upgrade.

c. Copy the latest version of the database from an existing node to the bootstrap node:

```
$ openstack overcloud external-upgrade run --stack <stack_name> --tags
system_upgrade_transfer_data
```



IMPORTANT

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

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

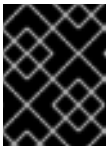
```
$ openstack overcloud upgrade run --stack <stack_name> \
 --playbook upgrade_steps_playbook.yaml \
 --tags nova_hybrid_state --limit all
```

This command launches temporary 16.2 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,overcloud-database-0,overcloud-networker-0,overcloud-ceph-0 --playbook all
```

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.

f. Optional: On the bootstrap Controller node, verify that after the upgrade, the new Pacemaker cluster is started and that the control plane services such as galera, rabbit, haproxy, and redis are running:

```
$ sudo pcs status
```

5. Upgrade the **overcloud-controller-1**, **overcloud-database-1**, **overcloud-networker-1**, and **overcloud-ceph-1** control plane nodes:

   a. Log in to the **overcloud-controller-1** node and verify that the old cluster is no longer running:

   ```
   $ sudo pcs status
   ```

   An error similar to the following is displayed when the cluster is not running:

   ```
   Error: cluster is not currently running on this node
   ```

   b. 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,overcloud-database-1,overcloud-networker-
   1,overcloud-ceph-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 nodes using the **ceph_ansible_limit** variable.

   This step is a preliminary measure to prepare the Ceph Storage services for the Leapp upgrade.

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

   ```
   $ openstack overcloud upgrade run --stack <stack_name> --tags system_upgrade --limit
   overcloud-controller-1,overcloud-database-1,overcloud-networker-1,overcloud-ceph-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.

   d. Run the upgrade command with no tags:

   ```
   $ openstack overcloud upgrade run --stack <stack_name> --limit overcloud-controller-
   0,overcloud-controller-1,overcloud-database-0,overcloud-database-1,overcloud-
   networker-0,overcloud-networker-1,overcloud-ceph-0,overcloud-ceph-1
   ```

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

6. Upgrade the **overcloud-controller-2**, **overcloud-database-2**, **overcloud-networker-2**, and **overcloud-ceph-2** control plane nodes:

   a. Log in to the **overcloud-controller-2** node and verify that the old cluster is no longer running:

   ```
   $ sudo pcs status
   ```

An error similar to the following is displayed when the cluster is not running:

```
Error: cluster is not currently running on this node
```

b. 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,overcloud-database-2,overcloud-networker-
2,overcloud-ceph-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 nodes using the **ceph_ansible_limit** variable.

This step is a preliminary measure to prepare the Ceph Storage services for the Leapp upgrade.

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

```
$ openstack overcloud upgrade run --stack <stack_name> --tags system_upgrade --limit
overcloud-controller-2,overcloud-database-2,overcloud-networker-2,overcloud-ceph-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.

d. Run the upgrade command with no tags:

```
$ openstack overcloud upgrade run --stack <stack_name> --limit overcloud-controller-
0,overcloud-controller-1,overcloud-controller-2,overcloud-database-0,overcloud-
database-1,overcloud-database-2,overcloud-networker-0,overcloud-networker-
1,overcloud-networker-2,overcloud-ceph-0,overcloud-ceph-1,overcloud-ceph-2
```

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

20.3. UPGRADING COMPUTE NODES IN PARALLEL

To upgrade a large number of Compute nodes to OpenStack Platform 16.2, you can run the **openstack overcloud upgrade run** command with the **--limit Compute** option in parallel on groups of 20 nodes.

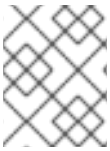
You can run multiple upgrade tasks in the background, where each task upgrades a separate group of 20 nodes. When you use this method to upgrade Compute nodes in parallel, you cannot select which nodes you upgrade. The selection of nodes is based on the inventory file that you generate when you run the **tripleo-ansible-inventory** command. For example, if you have 80 Compute nodes in your deployment, you can run the following commands to update the Compute nodes in parallel:

```
$ openstack overcloud upgrade run -y --limit 'Compute[0:19]' > upgrade-compute-00-19.log 2>&1 &
$ openstack overcloud upgrade run -y --limit 'Compute[20:29]' > upgrade-compute-20-29.log 2>&1 &
```

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

To upgrade specific Compute nodes, use a comma-separated list of nodes:

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



NOTE

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

Procedure

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

```
$ source ~/stackrc
```

3. Migrate your instances. For more information on migration strategies, see Migrating virtual machines between Compute nodes.
4. Run the upgrade command with the **system_upgrade** tag:

```
$ openstack overcloud upgrade run -y --stack STACK NAME --tags system_upgrade --limit
'Compute[0:19]' > upgrade-compute-00-19.log 2>&1 &
$ openstack overcloud upgrade run -y --stack STACK NAME --tags system_upgrade --limit
'Compute[20:29]' > upgrade-compute-20-29.log 2>&1 &
$ openstack overcloud upgrade run -y --stack STACK NAME --tags system_upgrade --limit
'Compute[40:59]' > update-compute-40-59.log 2>&1 &
$ openstack overcloud upgrade run -y --stack STACK NAME --tags system_upgrade --limit
'Compute[60:79]' > update-compute-60-79.log 2>&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.

5. Run the upgrade command with no tags:

```
$ openstack overcloud upgrade run -y --stack STACK NAME --limit 'Compute[0:19]' >
upgrade-compute-00-19.log 2>&1 &
$ openstack overcloud upgrade run -y --stack STACK NAME --limit 'Compute[20:29]' >
upgrade-compute-20-29.log 2>&1 &
$ openstack overcloud upgrade run -y --stack STACK NAME --limit 'Compute[40:59]' >
update-compute-40-59.log 2>&1 &
$ openstack overcloud upgrade run -y --stack STACK NAME --limit 'Compute[60:79]' >
update-compute-60-79.log 2>&1 &
```

   This command performs the Red Hat OpenStack Platform upgrade.

6. Optional: To upgrade selected Compute nodes, use the **--limit** option with a comma-separated list of nodes that you want to upgrade. The following example upgrades the **overcloud-compute-0**, **overcloud-compute-1**, **overcloud-compute-2** nodes in parallel.

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

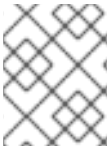
   ```
   $ openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit overcloud-compute-0,overcloud-compute-1,overcloud-compute-2
   ```

   b. Run the upgrade command with no tags:

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

20.4. 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 16.2.



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. To re-enable fencing in your overcloud, set the **EnableFencing** parameter to **true** in the **fencing.yaml** environment file.

4. 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 (**fencing.yaml**) with the **EnableFencing** parameter set to **true**.
- 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.

5. Wait until the stack synchronization completes.

IMPORTANT

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

20.5. UPGRADING THE ENTIRE OVERCLOUD AT ONCE

This upgrade process requires you to first shut down all workloads that are running in the overcloud. You then upgrade all of your overcloud systems, and restart the workloads afterward. This process is driven from the undercloud.

You can also upgrade Compute nodes that include Red Hat Ceph Storage as part of this procedure, or upgrade them separately after you upgrade all the other nodes.

Prerequisites

- In the **upgrades-environment.yaml** file, include the following parameter in **parameter_defaults**:

```
AllInOneUpgrade: true
```

Procedure

1. Shut down your workloads.
2. If you deployed director-integrated Ceph, switch the Ceph systemd files to podman:

```
$ openstack overcloud external-upgrade run --stack overcloud --tags ceph_systemd -e
ceph_ansible_limit=controller-0,controller-1,controller-2,ceph-0,ceph-1,ceph-2,networker-
0,networker-1
```

   - Replace **controller-0**,**controller-1**,**controller-2**,**ceph-0**,**ceph-1**,**ceph-2**,**networker-0**,**networker-1** with the server names for the roles in your environment.
   - To upgrade Compute nodes that contain Ceph, include the hostname of the Compute node in the **openstack overcloud external-upgrade run** command. For example:

```
$ openstack overcloud upgrade run --stack overcloud --tags ceph_systemd -e
ceph_ansible_limit=overcloud-compute-0,overcloud-compute-1
```

     Additionally, include the hostname of the Compute node in the commands in steps 4 and 5.

3. Stop all RHOSP services on the nodes:

```
$ openstack overcloud external-upgrade run --stack overcloud --tags
system_upgrade_stop_services
```

4. Run the system upgrade on all nodes. If you deployed director-integrated Ceph, include all Ceph nodes in the same --limit command:

```
$ openstack overcloud upgrade run --stack overcloud --tags system_upgrade --limit
controller-0,controller-1,controller-2,ceph-0,ceph-1,ceph-2,networker-0,networker-1
```

5. Run the upgrade without tags:

```
$ openstack overcloud upgrade run --stack overcloud --limit controller-0,controller-
1,controller-2,ceph-0,ceph-1,ceph-2,networker-0,networker-1
```

Next steps

- Continue with the upgrade from Synchronizing the overcloud stack.

CHAPTER 21. UPGRADING A SPLIT CONTROLLER OVERCLOUD

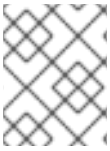
This scenario contains an example upgrade process for an overcloud with Controller node services split on to multiple nodes. This includes the following node types:

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

21.1. 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 16.2
- 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
```

21.2. UPGRADING PACEMAKER-BASED NODES

Upgrade all nodes that host Pacemaker services to OpenStack Platform 16.2. 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.

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

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

```
$ tripleo-ansible-inventory --list [--stack <stack_name>] |jq .overcloud_Controller.hosts[0]
```

- Optional: Replace **<stack_name>** with the name of the stack. If not specified, the default is **overcloud**.

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
   ```

   Replace **<stack_name>** with the name of your stack.

   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 16.2 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.

21.3. UPGRADING NON-PACEMAKER CONTROLLER NODES

Upgrade all nodes without Pacemaker-based services to OpenStack Platform 16.2. 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.

21.4. 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.

21.5. UPGRADING THE OPERATING SYSTEM FOR CEPH STORAGE NODES

If your deployment uses a Red Hat Ceph Storage cluster that was deployed using director, you must 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. 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.

3. 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.

4. 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.

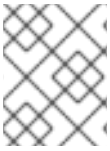
c. Run the upgrade command with no tags:

```
$ 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.

21.6. UPGRADING COMPUTE NODES

Upgrade all the Compute nodes to OpenStack Platform 16.2.



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:

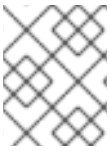
```
$ openstack overcloud upgrade run --stack STACK NAME --tags system_upgrade --limit overcloud-compute-0,overcloud-compute-1,overcloud-compute-2
```

Then perform the standard OpenStack service upgrade:

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

21.7. 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 16.2.



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. To re-enable fencing in your overcloud, set the **EnableFencing** parameter to **true** in the **fencing.yaml** environment file.
4. 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 (**fencing.yaml**) with the **EnableFencing** parameter set to **true**.
- 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.

5. Wait until the stack synchronization completes.

IMPORTANT

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

CHAPTER 22. UPGRADING AN OVERCLOUD WITH HYPER-CONVERGED INFRASTRUCTURE

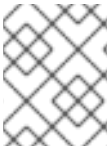
This scenario contains an example upgrade process for an overcloud with hyper-converged infrastructure (HCI), which includes the following node types:

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

22.1. 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 16.2
- 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
```

22.2. UPGRADING CONTROLLER NODES WITH DIRECTOR-DEPLOYED CEPH STORAGE

If your deployment uses a Red Hat Ceph Storage cluster that was deployed using director, you must complete this procedure.

To upgrade all the Controller nodes to OpenStack Platform 16.2, you must upgrade each Controller node starting with the bootstrap Controller node.

During the bootstrap Controller node upgrade process, a new Pacemaker cluster is created and new Red Hat OpenStack 16.2 containers are started on the node, while the remaining Controller nodes are still running on Red Hat OpenStack 13.

After upgrading the bootstrap node, you must upgrade each additional node with Pacemaker services and ensure that each node joins the new Pacemaker cluster started with the bootstrap node. For more information, see Overcloud node upgrade workflow.

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.

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

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

```
$ tripleo-ansible-inventory --list [--stack <stack_name>] |jq .overcloud_Controller.hosts[0]
```

- Optional: Replace **<stack_name>** with the name of the stack. If not specified, the default is **overcloud**.

3. Upgrade the bootstrap Controller node:

   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-controller-0
   ```

   Replace **<stack_name>** with the name of your stack.

   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:

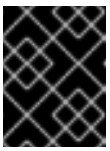
   ```
   $ openstack overcloud upgrade run [--stack <stack_name>] --playbook
   upgrade_steps_playbook.yaml --tags nova_hybrid_state --limit all
   ```

   This command launches temporary 16.2 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.

f. Verify that after the upgrade, the new Pacemaker cluster is started and that the control plane services such as galera, rabbit, haproxy, and redis are running:

```
$ sudo pcs status
```

4. Upgrade the next Controller node:

a. Verify that the old cluster is no longer running:

```
$ sudo pcs status
```

An error similar to the following is displayed when the cluster is not running:

```
Error: cluster is not currently running on this node
```

b. 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.

c. 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.

d. 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. Verify that the old cluster is no longer running:

   ```
   $ sudo pcs status
   ```

   An error similar to the following is displayed when the cluster is not running:

   ```
   Error: cluster is not currently running on this node
   ```

   b. 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.

   c. 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.

   d. 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.

22.3. UPGRADING COMPUTE NODES WITH HYPER-CONVERGED INFRASTRUCTURE (HCI)

Upgrade HCI Compute nodes to OpenStack Platform 16.2.

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 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.

4. 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.

5. 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.

6. 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,overcloud-computehci-1,overcloud-computehci-
2
```

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-computehci-
0,overcloud-computehci-1,overcloud-computehci-2
```

22.4. 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 16.2.

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. To re-enable fencing in your overcloud, set the **EnableFencing** parameter to **true** in the **fencing.yaml** environment file.
4. Run the upgrade finalization command:

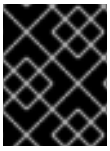
```
$ openstack overcloud upgrade converge \
  --stack STACK NAME \
  --templates \
  -e ENVIRONMENT FILE
  …
  -e /home/stack/templates/upgrades-environment.yaml \
  -e /home/stack/templates/rhsm.yaml \
```

```
-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 (**fencing.yaml**) with the **EnableFencing** parameter set to **true**.
- 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.

5. Wait until the stack synchronization completes.

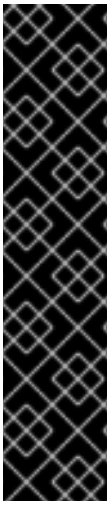


IMPORTANT

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

CHAPTER 23. UPGRADING A DIRECTOR-DEPLOYED CEPH STORAGE CLUSTER TO RED HAT CEPH STORAGE 4

If your deployment uses a Red Hat Ceph Storage cluster that was deployed using director, you must complete the procedures included in this section.



IMPORTANT

When you upgrade a Red Hat Ceph Storage cluster from a previous supported version to version 4.2z2, the upgrade completes with the storage cluster in a **HEALTH_WARN** state with a warning message that states **monitors are allowing insecureglobal_id reclaim**. This is due to the patched CVE (CVE-2021-20288), see Ceph HEALTH_WARN with 'mons are allowing insecure global_id reclaim' after install/upgrade to RHCS 4.2z2 (or newer).

Because the **HEALTH_WARN** state is displayed due to the CVE, it is possible to mute health warnings temporarily. However, there is a risk that if you mute warnings you do not have visibility about potential older and unpatched clients connected to your cluster. For more information about muting health warnings, see Upgrading a Red Hat Ceph Storage cluster in the *Red Hat Ceph Storage* documentation.

NOTE

If you are upgrading with external Ceph deployments, you must skip the procedures included in this section and continue to the next section.

After you upgrade the overcloud, upgrade your director-deployed Ceph Storage cluster to Red Hat Ceph Storage cluster to version 4.

23.1. INSTALLING CEPH-ANSIBLE

If your deployment uses a Red Hat Ceph Storage cluster that was deployed using director, you must complete this procedure.

The **ceph-ansible** package is required when you use Ceph Storage with Red Hat OpenStack Platform.

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
```

23.2. 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 24. OSD MIGRATION FROM FILESTORE TO BLUESTORE

After you complete and verify the upgrade process, you must migrate your FileStore OSDs to BlueStore. You must complete the migration one node at a time. The following procedure uses **ceph-ansible** to complete the migration. This procedure only applies if the Ceph cluster is deployed by director.

24.1. CHECKING THAT YOUR CLUSTER RUNS FILESTORE AND THEREFORE REQUIRES MIGRATION

Procedure

1. Log in as the **heat-admin** user on a node with Ceph MON containers, such as Controller nodes or standalone Ceph MON nodes. For example, in a standard overcloud deployment, **overcloud-controller-1** uses Ceph MON containers.

2. Query the Ceph cluster to see what driver is in use by the OSDs:

```
[heat-admin@overcloud-controller-1 ~]$ sudo -i
[root@overcloud-controller-1 ~]# podman exec -it ceph-mon-overcloud-controller-1 sh -c
"ceph -f json osd metadata" | jq -c 'sort_by(.hostname) | .[] | ["host", .hostname, "osd_id", .id,
"objectstore", .osd_objectstore]'
[root@overcloud-controller-1 ~]#
```

3. If any line returns **"objectstore": "filestore"**, that node requires OSD migration.

WARNING

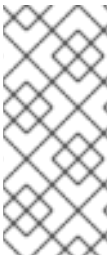
The migration time can vary depending on the size of your cluster. If you have a very large cluster, the migration time is proportional to the number of OSDs in that cluster and the amount of data stored. Ensure that you complete the migration as soon as possible so that your environment is not in a mixed architecture scenario, which can impact performance.

WARNING

Because managing FileStore-based OSDs with Red Hat Ceph Storage (RHCS) 4 versions of **ceph-ansible** is not supported, complete the migration before you run any stack updates.

24.2. MIGRATING OSDS FROM FILESTORE TO BLUESTORE

To migrate from FileStore to BlueStore, director uses Ansible to delete and recreate all OSDs on the node. Director performs a capacity check before the migration process. Finally, director redploys the OSDs with the BlueStore back end.



NOTE

You can expedite the migration from FileStore to BlueStore by increasing the throttling on backfill and recovery operations. For more information about throttling, see the Red Hat Knowledgebase article Ceph: Throttling down or up backfill and recovery and rebalance. Before beginning the throttling procedure, contact the Red Hat Ceph Storage team to ensure that the procedure is safe to use in your environment.

Prerequisites

- A functional and running Red Hat Ceph Storage (RHCS) 4 cluster. You can check the cluster by entering the following command in a ceph MON container on a Controller or Standalone Ceph MON node:

```
[root@overcloud-controller-1 ~]# podman exec -it ceph-mon-overcloud-controller-1 sh -c
"ceph  -s"
```

Procedure

1. Ensure that **osd_objectstore** in the **CephAnsibleDisksConfig** parameter does not default to **filestore**. If the **osd_objectstore** parameter is present in any of your custom heat environment files, you must define the value **bluestore** explicitly or remove it:

```
parameter_defaults:
  CephAnsibleDisksConfig:
    devices:
      - /dev/sdb
      - /dev/sdc
      - /dev/sdd
    osd_scenario: lvm
   osd_objectstore: bluestore
```

NOTE

If you have any specific FileStore configuration with, for example, journals, ensure that you update the configuration accordingly. For more information about advanced configurations, see Mapping the Ceph Storage Node Disk Layout in the *Deploying an overcloud with containerized Red Hat Ceph* guide.

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

3. Enter the **openstack overcloud external-upgrade run** command with the **ceph_fstobs** tag. Replace **<NODE_NAME>** with the name of the Ceph OSD node you want to upgrade. You can use the **openstack server list** command to find the node name.

```
[stack@undercloud ~] $ openstack overcloud external-upgrade run --tags ceph_fstobs -e
ceph_ansible_limit=<NODE_NAME> | tee oc-fstobs.log
```

4. Log in to a node that has Ceph MON services and query the Ceph cluster to check the status of the OSDs of the node you have upgraded. Before you can start the migration of the next OSD node, you must ensure that the one you have upgraded is successfully migrated:

```
[heat-admin@overcloud-controller-1 ~]$ sudo -i
[root@overcloud-controller-1 ~]# podman exec -it ceph-mon-overcloud-controller-1 sh -c
"ceph -f json osd metadata" | jq -c '.[] | select(.hostname == "<NODE_NAME>") | ["host",
.hostname, "osd_id", .id, "objectstore", .osd_objectstore]'
[root@overcloud-controller-1 ~]# exit
```

   Replace <NODE_NAME> with the name of the node that was migrated. If the result shows that the OSDs use BlueStore, its migration is successful.

5. Optional: To view additional details about a specific OSD, enter the following command:

```
[root@overcloud-controller-1 ~]# podman exec -it ceph-mon-overcloud-controller-1 sh -c
"ceph osd metadata <ID>"
```

   Replace <ID> with the ID of the OSD you want to query.

6. Before you can start the migration process on the next node, you must wait for the cluster to synchronize.

```
[root@overcloud-controller-1 ~]# podman exec -it ceph-mon-overcloud-controller-1 sh -c
"ceph  -s"
```

```
Review the command output and ensure that the health of the cluster is `HEALTH_OK` and
the PGs are in the `active+clean` state.
```

7. Before you can start the migration process on the next node, you must wait for the cluster rebalancing process to complete. To follow the status, run the following command:

```
[heat-admin@overcloud-controller-0 ~]$ sudo podman exec ceph-mon-<NODE_NAME>
ceph -w
```

   Replace <NODE_NAME> with the name of the node that was migrated.

8. Repeat the migration process for each node in the storage cluster.

For more information about migration from FileStore to BlueStore, see BlueStore in the Red Hat Ceph Storage *Administration Guide*.

24.3. VERIFYING YOUR FILESTORE TO BLUESTORE MIGRATION

You can check the status of an OSD to ensure that you have successfully completed the migration.

Procedure

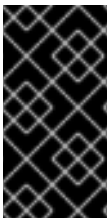
1. Log in as the **heat-admin** user to a ceph-mon container that is hosted on one of the Controller nodes.

2. Query the Ceph cluster:

```
[heat-admin@overcloud-controller-1 ~]$ sudo -i
```

```
[root@overcloud-controller-1 ~]# podman exec -it ceph-mon-overcloud-controller-1 sh -c
"ceph -f json osd metadata" | jq -c 'sort_by(.hostname) | .[] | ["host", .hostname, "osd_id", .id,
"objectstore", .osd_objectstore]'
[root@overcloud-controller-1 ~]#
```

If the configuration shows that all the OSDs across the cluster use BlueStore, the migration is successful.



IMPORTANT

A recommended best practice is to run an idempotent stack update to ensure that the configuration definition and the actual configuration match. The stack update duration varies depending on the size of your system, so to reduce downtime you can plan to complete the migration during a maintenance window.

CHAPTER 25. 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.

25.1. REMOVING UNNECESSARY PACKAGES AND DIRECTORIES FROM THE UNDERCLOUD

After the Leapp upgrade, remove the unnecessary packages and directories that remain on the undercloud.

Procedure

1. Remove the unnecessary packages

```
$ sudo dnf -y remove --exclude=python-pycadf-common python2*
```

2. Remove the content from the **/httpboot** and **/tftpboot** directories that includes old images used in Red Hat OpenStack 13:

```
$ sudo rm -rf /httpboot /tftpboot
```

25.2. DELETING USERS OF REDUNDANT TELEMETRY SERVICES

The telemetry endpoints are disabled by default. You can use this procedure to remove any telemetry endpoints that remain after the upgrade.

Prerequisites

- You have telemetry endpoints remaining after the upgrade. You can use the procedure to identify the remaining telemetry endpoints.

Procedure

1. Log in to your undercloud and source the overcloud authentication file:

```
$ source ~/overcloudrc
```

2. Identify the telemetry endpoints that remain after the upgrade:

```
$ openstack endpoint list | grep -i -e aodh -e gnocchi -e panko
```

3. Delete the users of the missing endpoints:

```
$ openstack user delete aodh gnocchi panko
```

Verification

- Verify that the endpoint users are absent:

```
$ openstack user list
```

25.3. 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. If no inventory file exists, you must generate a static inventory file:

```
$ tripleo-ansible-inventory
--static-yaml-inventory ~$HOME/config-download/<stack_name>/tripleo-ansible-inventory.yaml
--stack <stack_name>
--ansible_ssh_user heat-admin
```

   If you are not using the default overcloud stack name, replace <stack_name> with the name of your stack.

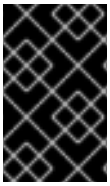
3. Run the **openstack tripleo validator run** command with the **--group post-upgrade** option:

```
$ openstack tripleo validator run --group {validation} --inventory ~$HOME/config-download/<stack_name>/tripleo-ansible-inventory.yaml
```

   If you are not using the default overcloud stack name, replace <stack_name> with the name of your stack.

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

```
$ openstack tripleo validator show run --full <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.

25.4. 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-x86_64
```

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-16.2.tar /usr/share/rhosp-director-images/ironic-python-agent-latest-16.2.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 16.2 images only with the OpenStack Platform 16.2 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.

25.5. UPDATING CPU PINNING PARAMETERS

Red Hat OpenStack Platform 16.2 uses new parameters for CPU pinning:

NovaComputeCpuDedicatedSet

Sets the dedicated (pinned) CPUs.

NovaComputeCpuSharedSet

Sets the shared (unpinned) CPUs.

You must migrate the CPU pinning configuration from the **NovaVcpuPinSet** parameter to the **NovaComputeCpuDedicatedSet** and **NovaComputeCpuSharedSet** parameters after completing the upgrade to Red Hat OpenStack Platform 16.2.

Procedure

1. Log in to the undercloud as the **stack** user.
2. If your Compute nodes support simultaneous multithreading (SMT) but you created instances with the **hw:cpu_thread_policy=isolated** policy, you must perform one of the following options:
   - Unset the **hw:cpu_thread_policy** thread policy and resize the instances:
     i. Source your overcloud authentication file:

     ```
     $ source ~/overcloudrc
     ```

     ii. Unset the **hw:cpu_thread_policy** property of the flavor:

     ```
     (overcloud) $ openstack flavor unset --property hw:cpu_thread_policy <flavor>
     ```

     NOTE

     - Unsetting the **hw:cpu_thread_policy** attribute sets the policy to the default **prefer** policy, which sets the instance to use an SMT-enabled Compute node if available. You can also set the **hw:cpu_thread_policy** attribute to **require**, which sets a hard requirements for an SMT-enabled Compute node.
     - If the Compute node does not have an SMT architecture or enough CPU cores with available thread siblings, scheduling will fail. To prevent this, set **hw:cpu_thread_policy** to **prefer** instead of **require**. The default **prefer** policy ensures that thread siblings are used when available.
     - If you use **hw:cpu_thread_policy=isolate**, you must have SMT disabled or use a platform that does not support SMT.

     iii. Convert the instances to use the new thread policy.

     ```
     (overcloud) $ openstack server resize --flavor <flavor> <server>
     (overcloud) $ openstack server resize confirm <server>
     ```

     Repeat this step for all pinned instances using the **hw:cpu_thread_policy=isolated** policy.

   - Migrate instances from the Compute node and disable SMT on the Compute node:
     i. Source your overcloud authentication file:

     ```
     $ source ~/overcloudrc
     ```

     ii. Disable the Compute node from accepting new virtual machines:

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

iii. Migrate all instances from the Compute node. For more information on instance migration, see Migrating virtual machine instances between Compute nodes .

iv. Reboot the Compute node and disable SMT in the BIOS of the Compute node.

v. Boot the Compute node.

vi. Re-enable the Compute node:

```
(overcloud) $ openstack compute service set <hostname> nova-compute --enable
```

3. Source the **stackrc** file:

```
$ source ~/stackrc
```

4. Edit the environment file that contains the **NovaVcpuPinSet** parameter.

5. Migrate the CPU pinning configuration from the **NovaVcpuPinSet** parameter to **NovaComputeCpuDedicatedSet** and **NovaComputeCpuSharedSet**:

- Migrate the value of **NovaVcpuPinSet** to **NovaComputeCpuDedicatedSet** for hosts that were previously used for pinned instances.
- Migrate the value of **NovaVcpuPinSet** to **NovaComputeCpuSharedSet** for hosts that were previously used for unpinned instances.
- If there is no value set for NovaVcpuPinSet, then all Compute node cores should be assigned to either **NovaComputeCpuDedicatedSet** or **NovaComputeCpuSharedSet**, depending on the type of instances you intend to host on the nodes.

For example, your previous environment file might contain the following pinning configuration:

```
parameter_defaults:
  ...
  NovaVcpuPinSet: 1,2,3,5,6,7
  ...
```

To migrate the configuration to a pinned configuration, set the **NovaComputeCpuDedicatedSet** parameter and unset the **NovaVcpuPinSet** parameter:

```
parameter_defaults:
  ...
  NovaComputeCpuDedicatedSet: 1,2,3,5,6,7
  NovaVcpuPinSet: ""
  ...
```

To migrate the configuration to an unpinned configuration, set the **NovaComputeCpuSharedSet** parameter and unset the **NovaVcpuPinSet** parameter:

```
parameter_defaults:
  ...
```

```
NovaComputeCpuSharedSet: 1,2,3,5,6,7
NovaVcpuPinSet: ""
...
```

IMPORTANT

Ensure the configuration of either **NovaComputeCpuDedicatedSet** or **NovaComputeCpuSharedSet** matches the configuration defined in **NovaVcpuPinSet**. To change the configuration for either of these, or to configure both **NovaComputeCpuDedicatedSet** or **NovaComputeCpuSharedSet**, ensure the Compute nodes with the pinning configuration are not running any instances before updating the configuration.

6. Save the file.
7. Run the deployment command to update the overcloud with the new CPU pinning parameters.

```
(undercloud) $ openstack overcloud deploy \
  --stack _STACK NAME_ \
  --templates \
  ...
  -e /home/stack/templates/<compute_environment_file>.yaml
  ...
```

Additional resources

- Configuring CPU pinning on Compute nodes

25.6. MIGRATING USERS TO THE MEMBER ROLE

In Red Hat OpenStack Platform 13, the default member role is called **_member_**.
In Red Hat OpenStack Platform 16.2, the default member role is called **member**.

When you complete the upgrade from Red Hat OpenStack Platform 13 to Red Hat OpenStack Platform 16.2, users that you assigned to the **_member_** role still have that role. You can migrate all of the users to the **member** role by using the following steps.

Prerequisites

- You have upgraded the overcloud to the latest version.

Procedure

1. List all of the users on your cloud that have the **_member_** role:

```
openstack role assignment list --names --role _member_ --sort-column project
```

2. For each user, remove the **_member_** role, and apply the **member** role:

```
openstack role remove --user <user> --project <project>  _member_
openstack role add --user <user> --project <project>  member
```

CHAPTER 26. TROUBLESHOOTING UPGRADE ISSUES

If you experience any issues with during the upgrade process, refer to the advice in this section.

26.1. 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.