



Red Hat OpenStack Platform 17.1

Framework for upgrades (16.2 to 17.1)

In-place upgrades from Red Hat OpenStack Platform 16.2 to 17.1

Red Hat OpenStack Platform 17.1 Framework for upgrades (16.2 to 17.1)

In-place upgrades from Red Hat OpenStack Platform 16.2 to 17.1

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 16.2 (Train) to 17.1 (Wallaby).

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE	4
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION	5
CHAPTER 1. ABOUT THE RED HAT OPENSTACK PLATFORM FRAMEWORK FOR UPGRADES	6
1.1. HIGH-LEVEL CHANGES IN RED HAT OPENSTACK PLATFORM 17.1	6
1.2. CHANGES IN RED HAT ENTERPRISE LINUX 9	6
1.3. UPGRADE FRAMEWORK FOR LONG LIFE VERSIONS	7
1.4. UPGRADE DURATION AND IMPACT	8
1.5. PLANNING AND PREPARATION FOR AN IN-PLACE UPGRADE	10
1.5.1. Familiarize yourself with Red Hat OpenStack Platform 17.1	10
1.5.2. Minor version update requirement	11
1.5.3. Leapp upgrade usage in Red Hat OpenStack Platform	11
1.5.4. Storage driver certification	11
1.5.5. Supported upgrade scenarios	11
1.5.6. Red Hat Virtualization upgrade process	12
1.5.7. Known issues that might block an upgrade	12
1.5.8. Backup and restore	14
1.5.9. Proxy configuration	14
1.5.10. Planning for a Compute node upgrade	15
1.6. REPOSITORIES	16
1.6.1. Undercloud repositories	16
1.6.2. Overcloud repositories	17
1.6.3. Red Hat Satellite Server 6 considerations	20
CHAPTER 2. UPGRADING THE UNDERCLOUD	23
2.1. ENABLING REPOSITORIES FOR THE UNDERCLOUD	23
2.2. VALIDATING RHOSP BEFORE THE UPGRADE	23
2.3. PREPARING CONTAINER IMAGES	24
2.4. GUIDELINES FOR CONTAINER IMAGE TAGGING	25
2.5. OBTAINING CONTAINER IMAGES FROM PRIVATE REGISTRIES	27
2.6. UPDATING THE UNDERCLOUD.CONF FILE	29
2.7. NETWORK CONFIGURATION FILE CONVERSION	29
2.8. RUNNING THE DIRECTOR UPGRADE	30
CHAPTER 3. UPGRADING WITH EXTERNAL CEPH DEPLOYMENTS	32
3.1. UPDATING CEPH CLIENT CONFIGURATION FOR RHOSP 17.1	32
CHAPTER 4. PREPARING FOR AN OVERCLOUD UPGRADE	35
4.1. PREPARING FOR OVERCLOUD SERVICE DOWNTIME	35
4.2. DISABLING FENCING IN THE OVERCLOUD	35
4.3. UNDERCLOUD NODE DATABASE BACKUP	36
4.4. UPDATING COMPOSABLE SERVICES IN CUSTOM ROLES_DATA FILES	36
4.5. CHECKING CUSTOM PUPPET PARAMETERS	39
4.6. FINAL REVIEW BEFORE UPGRADE	40
4.6.1. Upgrade command overview	40
4.6.1.1. openstack overcloud upgrade prepare	40
4.6.1.2. openstack overcloud upgrade run	40
4.6.1.3. openstack overcloud external-upgrade run	41
4.6.2. Upgrade Parameters	41
4.6.3. Custom files to include in your deployment	42
4.6.4. New environment files to include with your deployment	42

4.6.5. Environment files to remove from your deployment	42
4.6.6. Upgrading IPA services	43
4.6.7. Upgrade checklist	43
CHAPTER 5. OVERCLOUD ADOPTION AND PREPARATION	45
5.1. PERFORMING THE OVERCLOUD ADOPTION AND PREPARATION	45
5.2. OVERCLOUD ADOPTION FOR MULTI-CELL ENVIRONMENTS	53
CHAPTER 6. UPGRADING AN OVERCLOUD WITH DIRECTOR-DEPLOYED CEPH DEPLOYMENTS	55
6.1. INSTALLING CEPH-ANSIBLE	55
6.2. UPGRADING TO RED HAT CEPH STORAGE 5	55
CHAPTER 7. PREPARING NETWORK FUNCTIONS VIRTUALIZATION (NFV)	60
7.1. NETWORK FUNCTIONS VIRTUALIZATION (NFV) ENVIRONMENT FILES	60
CHAPTER 8. UPGRADING THE OVERCLOUD	62
8.1. UPGRADING RHOSP ON ALL NODES IN EACH STACK	62
CHAPTER 9. UPGRADING THE UNDERCLOUD OPERATING SYSTEM	63
9.1. SETTING THE SSH ROOT PERMISSION PARAMETER ON THE UNDERCLOUD	63
9.2. VALIDATING YOUR SSH KEY SIZE	63
9.3. PERFORMING THE UNDERCLOUD SYSTEM UPGRADE	64
CHAPTER 10. UPGRADING THE CONTROL PLANE OPERATING SYSTEM	66
10.1. UPGRADING THE CONTROL PLANE NODES	66
CHAPTER 11. UPGRADING THE COMPUTE NODE OPERATING SYSTEM	70
11.1. SELECTING COMPUTE NODES FOR UPGRADE TESTING	70
11.2. UPGRADING ALL COMPUTE NODES TO RHEL 9.2	71
11.3. UPGRADING COMPUTE NODES TO A MULTI-RHEL ENVIRONMENT	72
11.3.1. Creating roles for Multi-RHEL Compute nodes	73
11.3.2. Upgrading the Compute node operating system	76
CHAPTER 12. PERFORMING POST-UPGRADE ACTIONS	78
12.1. UPGRADING THE OVERCLOUD IMAGES	78
12.2. UPDATING CPU PINNING PARAMETERS	78
12.3. UPDATING THE DEFAULT MACHINE TYPE FOR HOSTS AFTER AN UPGRADE TO RHOSP 17	81
12.4. RE-ENABLING FENCING IN THE OVERCLOUD	83
CHAPTER 13. UPGRADING RED HAT CEPH STORAGE 5 TO 6	85
13.1. DIRECTOR-DEPLOYED RED HAT CEPH STORAGE ENVIRONMENTS	85
13.1.1. Updating the cephadm client	85
13.1.2. Updating the Red Hat Ceph Storage container image	86
13.1.3. Running the container image prepare	86
13.1.4. Upgrading to Red Hat Ceph Storage 6 with Orchestrator	87
13.1.5. Upgrading NFS Ganesha when moving from Red Hat Ceph Storage 5 to 6	88
13.2. EXTERNAL RED HAT CEPH STORAGE CLUSTER ENVIRONMENT	89
13.2.1. Updating the Red Hat Ceph Storage container image	89
13.2.2. Running the container image prepare	89
13.2.3. Upgrading NFS Ganesha when moving from Red Hat Ceph Storage 5 to 6	90

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. HIGH-LEVEL CHANGES IN RED HAT OPENSTACK PLATFORM 17.1

The following high-level changes occur during the upgrade to Red Hat OpenStack Platform (RHOSP) 17.1:

- The RHOSP upgrade and the operating system upgrade are separated into two distinct phases. You upgrade RHOSP first, then you upgrade the operating system.
- You can upgrade a portion of your Compute nodes to RHEL 9.2 while the rest of your Compute nodes remain on RHEL 8.4. This is called a Multi-RHEL environment.
- With an upgrade to Red Hat Ceph Storage 5, **cephadm** now manages Red Hat Ceph Storage. Previous versions of Red Hat Ceph Storage were managed by **ceph-ansible**. You can upgrade your Red Hat Ceph Storage nodes from version 5 to version 6 after the upgrade to RHOSP 17.1 is complete. Otherwise, Red Hat Ceph Storage nodes can remain on version 5 with RHOSP 17.1 until the end of the Red Hat Ceph Storage 5 lifecycle. To upgrade from Red Hat Ceph Storage version 5 to version 6, begin with one of the following procedures for your environment:
 - Director-deployed Red Hat Ceph Storage environments: [Updating the **cephadm** client](#)
 - External Red Hat Ceph Storage cluster environments: [Updating the Red Hat Ceph Storage container image](#)
- By default, the RHOSP overcloud uses Open Virtual Network (OVN) as the default ML2 mechanism driver in versions 16.2 and 17.1. If your RHOSP 16.2 deployment uses the OVS mechanism driver, you must upgrade to 17.1 with the OVS mechanism driver. Do not attempt to change the mechanism driver during the upgrade. After the upgrade, you can migrate from the OVS to the OVN mechanism driver. See [Migrating to the OVN mechanism driver](#).
- In ML2/OVN deployments, you can enable egress minimum and maximum bandwidth policies for hardware offloaded ports. For more information, see [Configuring the Networking service for QoS policies](#) in *Configuring Red Hat OpenStack Platform networking*.
- The undercloud and overcloud both run on Red Hat Enterprise Linux 9.

1.2. CHANGES IN RED HAT ENTERPRISE LINUX 9

The Red Hat OpenStack Platform 17.1 uses Red Hat Enterprise Linux (RHEL) 9.2 as the base operating system. As a part of the upgrade process, you will upgrade the base operating system of nodes to RHEL 9.2.

Before beginning the upgrade, review the following information to familiarize yourself with RHEL 9:

- For more information about the latest changes in RHEL 9, see the [Red Hat Enterprise Linux 9.2 Release Notes](#).

- For more information about the key differences between Red Hat Enterprise Linux 8 and 9, see [Considerations in adopting RHEL 9](#).
- For general information about Red Hat Enterprise Linux 9, see [Product Documentation for Red Hat Enterprise Linux 9](#).
- For more information about upgrading from RHEL 8 to RHEL 9, see [Upgrading from RHEL 8 to RHEL 9](#).

1.3. 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 16.2.4 and later	Red Hat OpenStack Platform 17.1 latest

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

Upgrade paths for long life releases

Familiarize yourself with the possible update and upgrade paths before you begin an upgrade. If you are using an environment that is earlier than RHOSP 16.2.4, before you upgrade from major version to major version, you must first update your existing environment to the latest minor release.

For example, if your current deployment is Red Hat OpenStack Platform (RHOSP) 16.2.1 on Red Hat Enterprise Linux (RHEL) 8.4, you must perform a minor update to the latest RHOSP 16.2 version before you upgrade to RHOSP 17.1.



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 16.2.x on RHEL 8.4	RHOSP 16.2 latest on RHEL 8.4 latest
RHOSP 17.0.x on RHEL 9.0	RHOSP 17.0 latest on RHEL 9.0 latest
RHOSP 17.0.x on RHEL 9.0	RHOSP 17.1 latest on RHEL 9.2 latest

RHOSP 17.1.x on RHEL 9.2	RHOSP 17.1 latest on RHEL 9.2 latest
--------------------------	--------------------------------------

For more information, see [Performing a minor update of Red Hat OpenStack Platform](#) .

Table 1.2. Upgrades version path

Current version	Target version
RHOSP 16.2 on RHEL 8.4	RHOSP 17.1 latest on RHEL 9.2 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 RHOSP 17.1 environment and migrate your workloads from your current environment to the new environment. For more information about RHOSP parallel migration, contact Red Hat Global Professional Services.

1.4. UPGRADE DURATION AND IMPACT

The durations in the following table were recorded in a test environment that consisted of an overcloud with 200 nodes, and 9 Ceph Storage hosts with 17 object storage daemons (OSDs) each. The durations in the table might not apply to all production environments. For example, if your hardware has low specifications or an extended boot period, allow for more time with these durations. Durations also depend on network I/O to container and package content, and disk I/O on the host.

To accurately estimate the upgrade duration for each task, perform these procedures in a test environment with hardware that is similar to your production environment.

Table 1.3. Duration and impact of In-place upgrade

	Duration	Notes
Undercloud upgrade	<ul style="list-style-type: none"> 30 minutes 	<ul style="list-style-type: none"> No disruption to the overcloud.
Overcloud adoption and preparation	<ul style="list-style-type: none"> 10 minutes for bare metal adoption 30 minutes for upgrade prepare 	<ul style="list-style-type: none"> Duration scales based on the size of the overcloud. No disruption to the overcloud.

	Duration	Notes
Red Hat Ceph Storage upgrade	<ul style="list-style-type: none"> ● Ceph upgrade ansible run: 90 minutes total, 10 minutes per node ● Ceph ansible run for cephadm adoption: 30 minutes total, 3 minutes per node ● Post ceph upgrade and adoption overcloud upgrade prepare: 20 minutes 	<ul style="list-style-type: none"> ● Duration scales based on the number of Storage hosts and OSDs. ● Storage performance is degraded.
Overcloud OpenStack upgrade	<ul style="list-style-type: none"> ● 120 minutes 	<ul style="list-style-type: none"> ● Duration scales based on the size of the overcloud. ● During this process, agents are restarted and API transactions might be lost. Disable client access to the OpenStack API during this stage.
Undercloud system upgrade	<ul style="list-style-type: none"> ● 40 minutes 	<ul style="list-style-type: none"> ● Includes multiple reboots. Reboot times are hardware dependent. ● Includes SELinux relabeling. Hosts with large numbers of files take significantly longer. ● No disruption to the overcloud.
Overcloud non-Compute host system upgrade	<ul style="list-style-type: none"> ● 30 minutes for upgrade prepare ● 40 minutes per host system upgrade 	<ul style="list-style-type: none"> ● Includes multiple reboots. Reboot times are hardware dependent. ● Includes SELinux relabeling. Hosts with large numbers of files take significantly longer. ● Performance is degraded.

	Duration	Notes
Overcloud Compute host upgrade	<ul style="list-style-type: none"> • 40 minutes per batch of hosts • 30 minutes for upgrade prepare 	<ul style="list-style-type: none"> • You run upgrade prepare on select batches of Compute nodes. The duration depends on the number of Compute nodes in each batch. There is no outage. • Includes multiple reboots. Reboot times are hardware dependent. • Includes SELinux relabeling. Hosts with large numbers of files take significantly longer. • To prevent workload outages during the reboot, you can migrate the workloads to another host beforehand.

1.5. 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.

1.5.1. Familiarize yourself with Red Hat OpenStack Platform 17.1

Before you perform an upgrade, familiarize yourself with Red Hat OpenStack Platform 17.1 to help you understand the resulting environment and any potential version-to-version changes that might affect your upgrade. To familiarize yourself with Red Hat OpenStack Platform 17.1, follow these suggestions:

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

Open the release notes for each version using these links:

- [Red Hat OpenStack Platform 16.2](#) , which is your source version
- [Red Hat OpenStack Platform 17.1](#) which is your target version
- Read the [Installing and managing Red Hat OpenStack Platform with director](#) guide for version 17.1 and familiarize yourself with any new requirements and processes in this guide.
- Install a proof-of-concept Red Hat OpenStack Platform 17.1 undercloud and overcloud. Develop hands-on experience of the target OpenStack Platform version and investigate potential differences between the target version and your current version.

1.5.2. Minor version update requirement

To upgrade from Red Hat OpenStack Platform (RHOSP) 16.2 to 17.1, your environment must be running RHOSP version 16.2.4 or later. If you are using a version of RHOSP that is earlier than 16.2.4, update the environment to the latest minor version of your current release. For example, update your Red Hat OpenStack Platform 16.2.3 environment to the latest 16.2 version before upgrading to Red Hat OpenStack Platform 17.1.

For instructions on performing a minor version update for Red Hat OpenStack Platform 16.2, see [Keeping Red Hat OpenStack Platform Updated](#).

1.5.3. 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 (RHEL) 8.4 to RHEL 9.2. The upgrade process uses the Leapp utility to perform the upgrade to RHEL 9.2. However, some aspects of the Leapp upgrade are customized to ensure that you are upgrading specifically from RHEL 8.4 to RHEL 9.2. To upgrade your operating system to RHEL 9.2, see [Performing the undercloud system upgrade](#).

Limitations

For information on potential limitations that might affect your upgrade, see the following sections from the *Upgrading from RHEL 8 to RHEL 9* guide:

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

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 8 to RHEL 9*.

1.5.4. Storage driver certification

Before you upgrade, confirm deployed storage drivers are certified for use with Red Hat OpenStack Platform 17.1.

For information on software certified for use with Red Hat OpenStack Platform 17.1, see [Software certified for Red Hat OpenStack Platform 17](#).

1.5.5. Supported upgrade scenarios

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



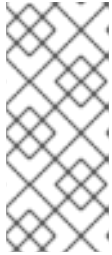
NOTE

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

Supported scenarios

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

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

1.5.6. Red Hat Virtualization upgrade process

If you are running your control plane on Red Hat Virtualization, there is no effect on the Red Hat OpenStack Platform (RHOSP) upgrade process. The RHOSP upgrade is the same regardless of whether or not an environment is running on Red Hat Virtualization.

1.5.7. Known issues that might block an upgrade

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

[BZ#2224085 - Leapp is stuck in Interim System when --debug is specified](#)

If you upgrade your operating system from RHEL 7.x to RHEL 8.x, or from RHEL 8.x to RHEL 9.x, do not run a Leapp upgrade with the **--debug** option. The system remains in the **early console in setup code** state and does not reboot automatically. To avoid this issue, the **UpgradeLeappDebug** parameter is set to **false** by default. Do not change this value in your templates.

[BZ#2203785 - Collectd sensubility stops working after overcloud node was rebooted.](#)

After rebooting an overcloud node, a permission issue causes collectd-sensubility to stop working. As a result, collectd-sensubility stops reporting container health. During an upgrade from Red Hat OpenStack Platform (RHOSP) 16.2 to RHOSP 17.1, overcloud nodes are rebooted as part of the Leapp upgrade. To ensure that collectd-sensubility continues to work, run the following command:

```
sudo podman exec -it collectd setfacl -R -m u:collectd:rwx /run/podman
```


BZ#2222683 - (Edge+DirOp+OSasInfra) FFU & Multi-Rhel Test Execution

The following deployments are not supported in a Multi-RHEL environment:

- ShiftOnStack
- DirectorOperator

You must upgrade all Compute hosts to RHEL 9.2, or keep all Compute hosts running on RHEL 8.4.

Distributed Compute Node (DCN) deployments do not support a Multi-RHEL environment, nor an upgrade from RHOSP 16.2 to RHOSP 17.1.

BZ#2180542 - After upgrade, manila ceph-nfs fails to start with error: ceph-nfs start on leaf1-controller-0 returned 'error' because 'failed'

The Pacemaker-controlled **ceph-nfs** resource requires a runtime directory to store some process data. The directory is created when you install or upgrade RHOSP. Currently, a reboot of the Controller nodes removes the directory, and the **ceph-nfs** service does not recover when the Controller nodes are rebooted. If all Controller nodes are rebooted, the **ceph-nfs** service fails permanently.

You can apply the following workaround:

1. If you reboot a Controller node, log in to the Controller node and create a **/var/run/ceph** directory:

```
$ mkdir -p /var/run/ceph
```
2. Repeat this step on all Controller nodes that have been rebooted. If the **ceph-nfs-pacemaker** service has been marked as failed, after creating the directory, run the following command from any of the Controller nodes:

```
$ pcs resource cleanup
```

BZ#2210873 - assimilate_{{ tripleo_cephadm_cluster }}.conf required if--crush-hierarchy is used

If the **CephPools** parameter is defined with a set of pool overrides, you must add **rule_name: replicated_rule** to the definition to avoid pool creation failures during an upgrade from RHOSP 16.2 to 17.1.

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.

[BZ#2246409 - \(OSP16.2→17.1\) Cinder volume NFS mounts on compute nodes are preventing leapp upgrade](#)

During an upgrade from RHOSP 16.2 to 17.1, the operating system upgrade from RHEL 8.4 to RHEL 9.2 fails if Cinder volume NFS mounts are present on Compute nodes. Contact your Red Hat support representative for a workaround.

[BZ#2266025 - FFU 16to17. System upgrade process is interrupted after undercloud reboot if OS was created from default RHEL 8.4 image](#)

There is an issue where Red Hat Enterprise Linux (RHEL) 8.4 images do not have the **"GRUB_DEFAULT=saved"** definition in the **/etc/default/grub** file. If you downloaded a RHEL 8.4 KVM Guest Image from the Red Hat Customer Portal to deploy your undercloud, and you are upgrading from Red Hat OpenStack Platform 16.2 to 17.1, the following issues occur:

- The system upgrade fails to update the grub menu properly.
- After a system reboot, director boots RHEL 8 instead of RHEL 9 on the nodes.

For a workaround for this issue, see the Red Hat Knowledgebase solution [FFU 16to17. System upgrade process is interrupted after undercloud reboot](#).

1.5.8. Backup and restore

Before you upgrade your Red Hat OpenStack Platform (RHOSP) 16.2 environment, back up the undercloud and overcloud control plane by using one of the following options:

- Back up your nodes before you perform an upgrade. For more information about backing up nodes before you upgrade, see [Red Hat OpenStack Platform 16.2 Backing up and restoring the undercloud and control plane nodes](#).
- Back up the undercloud node after you perform the undercloud upgrade and before you perform the overcloud upgrade. For more information about backing up the undercloud, see [Creating a backup of the undercloud node](#) in the *Red Hat OpenStack Platform 17.1 Backing up and restoring the undercloud and control plane nodes*.
- Use a third-party backup and recovery tool that suits your environment. For more information about certified backup and recovery tools, see the [Red Hat Ecosystem catalog](#).

1.5.9. Proxy configuration

If you use a proxy with your Red Hat OpenStack Platform 16.2 environment, the proxy configuration in the **/etc/environment** file will persist past the operating system upgrade and the Red Hat OpenStack Platform 17.1 upgrade.

- For more information about proxy configuration for Red Hat OpenStack Platform 16.2, see [Considerations when running the undercloud with a proxy](#) in *Installing and managing Red Hat OpenStack Platform with director*.
- For more information about proxy configuration for Red Hat OpenStack Platform 17.1, see [Considerations when running the undercloud with a proxy](#) in *Installing and managing Red Hat OpenStack Platform with director*.

1.5.10. Planning for a Compute node upgrade

After you upgrade your Compute nodes from Red Hat OpenStack Platform (RHOSP) 16.2 to RHOSP 17.1, you can choose one of the following options to upgrade the Compute host operating system:

- Keep a portion of your Compute nodes on Red Hat Enterprise Linux (RHEL) 8.4, and upgrade the rest to RHEL 9.2. This is referred to as a Multi-RHEL environment.
- Upgrade all Compute nodes to RHEL 9.2, and complete the upgrade of the environment.
- Keep all Compute nodes on RHEL 8.4. The lifecycle of RHEL 8.4 applies.

Benefits of a Multi-RHEL environment

You must upgrade all of your Compute nodes to RHEL 9.2 to take advantage of any hardware-related features that are only supported in RHOSP 17.1, such as vTPM and Secure Boot. However, you might require that some or all of your Compute nodes remain on RHEL 8.4. For example, if you certified an application for RHEL 8, you can keep your Compute nodes running on RHEL 8.4 to support the application without blocking the entire upgrade.

The option to upgrade part of your Compute nodes to RHEL 9.2 gives you more control over your upgrade process. You can prioritize upgrading the RHOSP environment within a planned maintenance window and defer the operating system upgrade to another time. Less downtime is required, which minimizes the impact to end users.



NOTE

If you plan to upgrade from RHOSP 17.1 to RHOSP 18.0, you must upgrade all hosts to RHEL 9.2. If you continue to run RHEL 8.4 on your hosts beyond the Extended Life Cycle Support phase, you must obtain a TUS subscription.

Limitations of a Multi-RHEL environment

The following limitations apply in a Multi-RHEL environment:

- Compute nodes running RHEL 8 cannot consume NVMe-over-TCP Cinder volumes.
- You cannot use different paths for socket files on RHOSP 16.2 and 17.1 for collectd monitoring.
- You cannot mix ML2/OVN and ML2/OVS mechanism drivers. For example, if your RHOSP 16.2 deployment included ML2/OVN, your Multi-RHEL environment must use ML2/OVN.
- FIPS is not supported in a Multi-RHEL environment. FIPs deployment is a Day 1 operation. FIPS is not supported in RHOSP 16.2. As a result, when you upgrade from RHOSP 16.2 to 17.1, the 17.1 environment does not include FIPS.
- Edge topologies are currently not supported.



IMPORTANT

All HCI nodes in supported Hyperconverged Infrastructure environments must use the same version of Red Hat Enterprise Linux as the version used by the Red Hat OpenStack Platform controllers. If you wish to use multiple Red Hat Enterprise versions in a hybrid state on HCI nodes in the same Hyperconverged Infrastructure environment, contact the [Red Hat Customer Experience and Engagement](#) team to discuss a support exception.

Upgrading Compute nodes

Use one of the following options to upgrade your Compute nodes:

- To perform a Multi-RHEL upgrade of your Compute nodes, see [Upgrading Compute nodes to a Multi-RHEL environment](#).
- To upgrade all Compute nodes to RHEL 9.2, see [Upgrading Compute nodes to RHEL 9.2](#).
- If you are keeping all of your Compute nodes on RHEL 8.4, no additional configuration is required.

1.6. 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.

1.6.1. Undercloud repositories

You run Red Hat OpenStack Platform (RHOSP) 17.1 on Red Hat Enterprise Linux (RHEL) 9.2. RHEL 8.4 Compute nodes are also supported in a Multi-RHEL environment when upgrading from RHOSP 16.2.



NOTE

If you synchronize repositories with Red Hat Satellite, you can enable specific versions of the Red Hat Enterprise Linux repositories. However, the repository remains the same despite the version you choose. For example, you can enable the 9.2 version of the BaseOS repository, but the repository name is still **rhel-9-for-x86_64-baseos-eus-rpms** despite the specific version you choose.



WARNING

Any repositories except the ones specified here are not supported. Unless recommended, do not enable any other products or repositories except 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 9 for x86_64 - BaseOS (RPMs) Extended Update Support (EUS)	rhel-9-for-x86_64-baseos-eus-rpms	Base operating system repository for x86_64 systems.
Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)	rhel-9-for-x86_64-appstream-eus-rpms	Contains Red Hat OpenStack Platform dependencies.
Red Hat Enterprise Linux 9 for x86_64 - High Availability (RPMs) Extended Update Support (EUS)	rhel-9-for-x86_64-highavailability-eus-rpms	High availability tools for Red Hat Enterprise Linux. Used for Controller node high availability.
Red Hat OpenStack Platform for RHEL 9 (RPMs)	openstack-17.1-for-rhel-9-x86_64-rpms	Core Red Hat OpenStack Platform repository, which contains packages for Red Hat OpenStack Platform director.
Red Hat Fast Datapath for RHEL 9 (RPMs)	fast-datapath-for-rhel-9-x86_64-rpms	Provides Open vSwitch (OVS) packages for OpenStack Platform.
Red Hat Enterprise Linux 8.4 for x86_64 - BaseOS (RPMs) Telecommunications Update Service (TUS)	rhel-8-for-x86_64-baseos-tus-rpms	Base operating system repository for x86_64 systems.
Red Hat Enterprise Linux 8.4 for x86_64 - AppStream (RPMs)	rhel-8-for-x86_64-appstream-tus-rpms	Contains Red Hat OpenStack Platform dependencies.
Red Hat OpenStack Platform for RHEL 8 x86_64 (RPMs)	openstack-17.1-for-rhel-8-x86_64-rpms	Core Red Hat OpenStack Platform repository.

1.6.2. Overcloud repositories

You run Red Hat OpenStack Platform (RHOSP) 17.1 on Red Hat Enterprise Linux (RHEL) 9.2. RHEL 8.4 Compute nodes are also supported in a Multi-RHEL environment when upgrading from RHOSP 16.2.



NOTE

If you synchronize repositories with Red Hat Satellite, you can enable specific versions of the Red Hat Enterprise Linux repositories. However, the repository remains the same despite the version you choose. For example, you can enable the 9.2 version of the BaseOS repository, but the repository name is still **rhel-9-for-x86_64-baseos-eus-rpms** despite the specific version you choose.

**WARNING**

Any repositories except the ones specified here are not supported. Unless recommended, do not enable any other products or repositories except 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 9 for x86_64 - BaseOS (RPMs) Extended Update Support (EUS)	rhel-9-for-x86_64-baseos-eus-rpms	Base operating system repository for x86_64 systems.
Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)	rhel-9-for-x86_64-appstream-eus-rpms	Contains Red Hat OpenStack Platform dependencies.
Red Hat Enterprise Linux 9 for x86_64 - High Availability (RPMs) Extended Update Support (EUS)	rhel-9-for-x86_64-highavailability-eus-rpms	High availability tools for Red Hat Enterprise Linux.
Red Hat OpenStack Platform for RHEL 9 x86_64 (RPMs)	openstack-17.1-for-rhel-9-x86_64-rpms	Core Red Hat OpenStack Platform repository.
Red Hat Fast Datapath for RHEL 9 (RPMs)	fast-datapath-for-rhel-9-x86_64-rpms	Provides Open vSwitch (OVS) packages for OpenStack Platform.
Red Hat Ceph Storage Tools 6 for RHEL 9 x86_64 (RPMs)	rhceph-6-tools-for-rhel-9-x86_64-rpms	Tools for Red Hat Ceph Storage 6 for Red Hat Enterprise Linux 9.
Red Hat Enterprise Linux 8.4 for x86_64 - BaseOS (RPMs) Telecommunications Update Service (TUS)	rhel-8-for-x86_64-baseos-tus-rpms	Base operating system repository for x86_64 systems.
Red Hat Enterprise Linux 8.4 for x86_64 - AppStream (RPMs)	rhel-8-for-x86_64-appstream-tus-rpms	Contains Red Hat OpenStack Platform dependencies.
Red Hat Enterprise Linux 8 for x86_64 - High Availability (RPMs) Telecommunications Update Service (TUS)	rhel-8-for-x86_64-highavailability-tus-rpms	High availability tools for Red Hat Enterprise Linux.

Name	Repository	Description of requirement
Red Hat OpenStack Platform for RHEL 8 x86_64 (RPMs)	openstack-17.1-for-rhel-8-x86_64-rpms	Core Red Hat OpenStack Platform repository.

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 9 for x86_64 - BaseOS (RPMs) Extended Update Support (EUS)	rhel-9-for-x86_64-baseos-eus-rpms	Base operating system repository for x86_64 systems.
Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)	rhel-9-for-x86_64-appstream-eus-rpms	Contains Red Hat OpenStack Platform dependencies.
Red Hat Enterprise Linux 9 for x86_64 - High Availability (RPMs) Extended Update Support (EUS)	rhel-9-for-x86_64-highavailability-eus-rpms	High availability tools for Red Hat Enterprise Linux.
Red Hat OpenStack Platform for RHEL 9 x86_64 (RPMs)	openstack-17.1-for-rhel-9-x86_64-rpms	Core Red Hat OpenStack Platform repository.
Red Hat Fast Datapath for RHEL 9 (RPMs)	fast-datapath-for-rhel-9-x86_64-rpms	Provides Open vSwitch (OVS) packages for OpenStack Platform.
Red Hat Ceph Storage Tools 6 for RHEL 9 x86_64 (RPMs)	rhceph-6-tools-for-rhel-9-x86_64-rpms	Tools for Red Hat Ceph Storage 6 for Red Hat Enterprise Linux 9.
Red Hat Enterprise Linux 8.4 for x86_64 - BaseOS (RPMs) Telecommunications Update Service (TUS)	rhel-8-for-x86_64-baseos-tus-rpms	Base operating system repository for x86_64 systems.
Red Hat Enterprise Linux 8.4 for x86_64 - AppStream (RPMs)	rhel-8-for-x86_64-appstream-tus-rpms	Contains Red Hat OpenStack Platform dependencies.
Red Hat OpenStack Platform for RHEL 8 x86_64 (RPMs)	openstack-17.1-for-rhel-8-x86_64-rpms	Core Red Hat OpenStack Platform 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 9 for x86_64 - BaseOS (RPMs)	rhel-9-for-x86_64-baseos-rpms	Base operating system repository for x86_64 systems.
Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)	rhel-9-for-x86_64-appstream-rpms	Contains Red Hat OpenStack Platform dependencies.
Red Hat OpenStack Platform Deployment Tools for RHEL 9 x86_64 (RPMs)	openstack-17.1-deployment-tools-for-rhel-9-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-17.1-for-rhel-9-x86_64-rpms repository.
Red Hat OpenStack Platform for RHEL 9 x86_64 (RPMs)	openstack-17.1-for-rhel-9-x86_64-rpms	Packages to help director configure Ceph Storage nodes. This repository is included with combined Red Hat OpenStack Platform and Red Hat Ceph Storage subscriptions. If you use a standalone Red Hat Ceph Storage subscription, use the openstack-17.1-deployment-tools-for-rhel-9-x86_64-rpms repository.
Red Hat Ceph Storage Tools 6 for RHEL 9 x86_64 (RPMs)	rhceph-6-tools-for-rhel-9-x86_64-rpms	Provides tools for nodes to communicate with the Ceph Storage cluster.
Red Hat Fast Datapath for RHEL 9 (RPMs)	fast-datapath-for-rhel-9-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.

1.6.3. 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 (RHOSP) environment and you plan to use Satellite 6 to deliver content during the RHOSP 17.1 upgrade, the following must be true:

- Your Satellite Server hosts RHOSP 16.2 RPMs and container images.
- You have registered all nodes in your RHOSP 16.2 environment to your Satellite Server.

For example, you used an activation key linked to a RHOSP 16.2 content view to register nodes to RHOSP 16.2 content.

Recommendations for RHOSP upgrades

- Enable and synchronize the necessary RPM repositories for both the RHOSP 16.2 undercloud and overcloud. This includes the necessary Red Hat Enterprise Linux (RHEL) 9.2 repositories.
- Create custom products on your Satellite Server to host container images for RHOSP 17.1.
- Create and promote a content view for RHOSP 17.1 upgrade and include the following content in the content view:
 - RHEL 8 repositories:
 - Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)


```
rhel-8-for-x86_64-appstream-tus-rpms
```
 - Red Hat Enterprise Linux 8 for x86_64 - BaseOS (RPMs)


```
rhel-8-for-x86_64-baseos-tus-rpms
```
 - Red Hat Enterprise Linux 8 for x86_64 - High Availability (RPMs)


```
rhel-8-for-x86_64-highavailability-eus-rpms
```
 - Red Hat Fast Datapath for RHEL 8 (RPMs)


```
fast-datapath-for-rhel-8-x86_64-rpms
```
 - RHEL 9 repositories:
 - Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)


```
rhel-9-for-x86_64-appstream-eus-rpms
```
 - Red Hat Enterprise Linux 9 for x86_64 - BaseOS (RPMs)


```
rhel-9-for-x86_64-baseos-eus-rpms
```
 - All undercloud and overcloud RPM repositories, including RHEL 9.2 repositories. To avoid issues enabling the RHEL repositories, ensure that you include the correct version of the RHEL repositories, which is 9.2.
 - RHOSP 17.1 container images.
- Associate an activation key with the RHOSP 17.1 content view that you have created for the RHOSP 17.1 upgrade.
- Check that no node has the **katello-host-tools-fact-plugin** package installed. The Leapp upgrade does not upgrade this package. Leaving this package on a RHEL 9.2 system causes **subscription-manager** to report errors.

- You can configure Satellite Server to host RHOSP 17.1 container images. To upgrade from RHOSP 16.2 to 17.1, you need the following container images:
 - Container images that are hosted on the **rhosp-rhel8** namespace:
 - **rhosp-rhel8/openstack-collectd**
 - **rhosp-rhel8/openstack-nova-libvirt**
 - Container images that are hosted on the **rhosp-rhel9** namespace. For information about configuring the **rhosp-rhel9** namespace container images, see [Preparing a Satellite server for container images](#) in *Installing and managing Red Hat OpenStack Platform with director* .
- If you use a Red Hat Ceph Storage subscription and have configured director to use the **overcloud-minimal** image for Red Hat Ceph Storage nodes, on your Satellite Server you must create a content view and add the following RHEL 9.2 repositories to it:
 - Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)

```
rhel-9-for-x86_64-appstream-eus-rpms
```
 - Red Hat Enterprise Linux 9 for x86_64 - BaseOS (RPMs)

```
rhel-9-for-x86_64-baseos-eus-rpms
```

For more information, see [Importing Content](#) and [Managing Content Views](#) in the *Red Hat Satellite Managing Content* guide.

CHAPTER 2. UPGRADING THE UNDERCLOUD

Upgrade the undercloud to Red Hat OpenStack Platform 17.1. The undercloud upgrade uses the running Red Hat OpenStack Platform 16.2 undercloud. The upgrade process exports heat stacks to files, and converts heat to ephemeral heat while upgrading the rest of the services on your nodes.

For information about the duration and impact of this upgrade procedure, see [Upgrade duration and impact](#).

2.1. 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 (RHEL) 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=openstack-17.1-for-rhel-8-x86_64-rpms \
--enable=fast-datapath-for-rhel-8-x86_64-rpms
```

3. Switch the **container-tools** module version to RHEL 8 on all nodes:

```
[stack@director ~]$ sudo dnf -y module switch-to container-tools:rhel8
```

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

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

2.2. VALIDATING RHOSP BEFORE THE UPGRADE

Before you upgrade to Red Hat OpenStack Platform (RHOSP) 17.1, validate your undercloud and overcloud with the **tripleo-validations** playbooks. In RHOSP 16.2, you run these playbooks through the OpenStack Workflow Service (mistral).

Procedure

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

```
$ source ~/stackrc
```

3. Adjust the permissions of the **/var/lib/mistral/.ssh** directory:

■

```
$ sudo chmod +x /var/lib/mistral/.ssh/
```

4. Install the packages for validation:

```
$ sudo dnf -y update openstack-tripleo-validations python3-validations-libs validations-common
```

5. Copy the inventory from mistral:

```
$ sudo chown stack:stack /var/lib/mistral/.ssh/tripleo-admin-rsa
$ sudo cat /var/lib/mistral/<stack>/tripleo-ansible-inventory.yaml > inventory.yaml
```

- Replace <stack> with the name of the stack.

6. Run the validation:

```
$ validation run -i inventory.yaml --group pre-upgrade
```

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

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

Success! The validation passed for all hosts:
* undercloud
```

2.3. 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 the 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 the undercloud host as the **stack** user.
2. Optional: Back up the 16.2 **containers-prepare-parameter.yaml** file:

```
$ cp containers-prepare-parameter.yaml \
  containers-prepare-parameter.yaml.orig
```

3. Generate the default container image preparation file:

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

This command includes the following additional options:

- **--local-push-destination** sets the registry on the undercloud as the location for container images. 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.

4. Modify the **containers-prepare-parameter.yaml** to suit your requirements. For more information about container image parameters, see [Container image preparation parameters](#).
5. If your deployment includes Red Hat Ceph Storage, update the Red Hat Ceph Storage container image parameters in the **containers-prepare-parameter.yaml** file for the version of Red Hat Ceph Storage that your deployment uses:

```
ceph_namespace: registry.redhat.io/rhceph
ceph_image: <ceph_image_file>
ceph_tag: latest
ceph_grafana_image: <grafana_image_file>
ceph_grafana_namespace: registry.redhat.io/rhceph
ceph_grafana_tag: latest
```

- Replace **<ceph_image_file>** with the name of the image file for the version of Red Hat Ceph Storage that your deployment uses:
 - Red Hat Ceph Storage 5: **rhceph-5-rhel8**
 - Red Hat Ceph Storage 6: **rhceph-6-rhel9**
- Replace **<grafana_image_file>** with the name of the image file for the version of Red Hat Ceph Storage that your deployment uses:
 - Red Hat Ceph Storage 5: **rhceph-5-dashboard-rhel8**
 - Red Hat Ceph Storage 6: **rhceph-6-dashboard-rhel9**

2.4. 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 17.1.0 and the release for the container image is **5.161**, then the resulting tag for the container image is 17.1.0-5.161.

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 17.1 and 17.1.0 link to the latest **release** in the 17.1.0 container stream. If a new minor release of 17.1 occurs, the 17.1 tag links to the latest **release** for the new minor release stream while the 17.1.0 tag continues to link to the latest **release** within the 17.1.0 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 17.1. This always corresponds to the latest minor version and release.

```
parameter_defaults:
  ContainerImagePrepare:
    - set:
      ...
      tag: 17.1
      ...
```

- To change to a specific minor version for OpenStack Platform container images, set the tag to a minor version. For example, to change to 17.1.2, set **tag** to 17.1.2.

```
parameter_defaults:
  ContainerImagePrepare:
    - set:
      ...
      tag: 17.1.2
      ...
```

- 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: 17.1
      ...
      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": "17.1.0",
```

```

| ...
| }

```

- 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 17.1.0 set for **version** and 5.161 set for **release**, the resulting tag for the container image is 17.1.0-5.161.
- 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.

2.5. 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**:

```

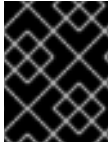
parameter_defaults:
  ContainerImagePrepare:
    - push_destination: true
    set:
      namespace: registry.redhat.io/...
    ...
  - push_destination: true

```

```

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
```

2.6. UPDATING THE UNDERCLOUD.CONF FILE

You can continue using the original **undercloud.conf** file from your Red Hat OpenStack Platform 16.2 environment, but you must modify the file to retain compatibility with Red Hat OpenStack Platform 17.1. For more information about parameters for configuring the **undercloud.conf** file, see [Undercloud configuration parameters](#) in *Installing and managing Red Hat OpenStack Platform with director*.

Procedure

1. Log in to your undercloud host as the **stack** user.
2. Create a file called **skip_rhel_release.yaml** and set the **SkipRhelEnforcement** parameter to **true**:

```
parameter_defaults:
  SkipRhelEnforcement: true
```

3. Open the **undercloud.conf** file and add the following parameters to the **DEFAULT** section in the file:

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

- Add any additional custom environment files to the **custom_env_files** parameter. The **custom_env_files** parameter defines the location of the **skip_rhel_release.yaml** file that is required for the upgrade.
- The **container_images_file** 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.



NOTE

If your original **undercloud.conf** file includes the **CertmongerKerberosRealm** parameter in the **/home/stack/custom-kerberos-params.yaml** file, you must replace the **CertmongerKerberosRealm** parameter with the **HAProxyCertificatePrincipal** parameter. The **CertmongerKerberosRealm** parameter causes the undercloud upgrade to fail.

4. Check all other parameters in the file for any changes.
5. Save the file.

2.7. NETWORK CONFIGURATION FILE CONVERSION

If your network configuration templates include the following functions, you must manually convert your NIC templates to Jinja2 Ansible format before you upgrade the undercloud. The following functions are not supported with automatic conversion:

- 'get_file'
- 'get_resource'
- 'digest'
- 'repeat'
- 'resource_facade'
- 'str_replace'
- 'str_replace_strict'
- 'str_split'
- 'map_merge'
- 'map_replace'
- 'yaql'
- 'equals'
- 'if'
- 'not'
- 'and'
- 'or'
- 'filter'
- 'make_url'
- 'contains'

For more information about manually converting your NIC templates, see [Manually converting NIC templates to Jinja2 Ansible format](#) in *Customizing your Red Hat OpenStack Platform deployment*.

2.8. RUNNING THE DIRECTOR UPGRADE

Upgrade director on the undercloud.

Prerequisites

- Confirm that the **tripleo_mysql.service** is running:

```
$ systemctl status tripleo_mysql
```

If the service is not running, start the service:

```
$ sudo systemctl start tripleo_mysql
```

- If your network configuration templates include certain functions, ensure that you manually convert your NIC templates to Jinja2 Ansible format. For a list of those functions and a link to the manual procedure, see [Network configuration file conversion](#).

Procedure

- Launch the director configuration script to upgrade director:

```
$ openstack undercloud upgrade
```

The director configuration script upgrades director packages and configures director services to match the settings in the **undercloud.conf** file. This script takes several minutes to complete.



NOTE

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

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

CHAPTER 3. UPGRADING WITH EXTERNAL CEPH DEPLOYMENTS

If your Red Hat OpenStack Platform (RHOSP) deployment uses an externally deployed Red Hat Ceph Storage cluster, you might need to upgrade your Red Hat Ceph Storage cluster before continuing with your RHOSP upgrade.

If your Red Hat Ceph Storage cluster is currently on Release 4, perform the following tasks:

1. Upgrade the Red Hat Ceph Storage cluster from Release 4 to Release 5.
2. Upgrade your RHOSP deployment from Release 16.2 to Release 17.1.
3. Upgrade the Red Hat Ceph Storage cluster from Release 5 to Release 6.

If your Red Hat Ceph Storage cluster is currently on Release 5, perform the following tasks:

1. Upgrade your RHOSP deployment from Release 16.2 to Release 17.1.
2. Upgrade the Red Hat Ceph Storage cluster from Release 5 to Release 6.

For more information about upgrading your Red Hat Ceph Storage cluster, see the following guides:

- [Red Hat Ceph Storage 5 Upgrade Guide](#)
- [Red Hat Ceph Storage 6 Upgrade Guide](#)

After you upgrade your Red Hat Ceph Storage cluster, you must migrate from the ceph-ansible **ceph-client** role to the tripleo-ansible **tripleo_ceph_client** role.

3.1. UPDATING CEPH CLIENT CONFIGURATION FOR RHOSP 17.1

Before Red Hat OpenStack Platform (RHOSP) 17.1, for external Red Hat Ceph Storage environments, OpenStack Ceph Clients were configured by the ceph-ansible **ceph-client** role. In RHOSP 17.1, OpenStack Ceph Clients are configured by the tripleo-ansible **tripleo_ceph_client** role. Before you run the overcloud upgrade in [Performing the overcloud adoption and preparation](#), you must replace the tripleo-heat-templates environment file that is used to configure the OpenStack services with an external Ceph cluster.

Procedure

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

```
$ source ~/stackrc
```

3. If you included the **environments/ceph-ansible/ceph-ansible-external.yaml** file in the following commands, you must replace the file with the **environments/external-ceph.yaml** file.
 - **openstack overcloud upgrade prepare**
 - **openstack overcloud deploy**
For example, replace

```
$ openstack overcloud deploy
...
-e environments/ceph-ansible/ceph-ansible-external.yaml
...
```

with

```
$ openstack overcloud deploy
...
-e environments/external-ceph.yaml
...
```

4. Create a file called **ceph_params.yaml** and include the following content:

```
parameter_defaults:
  CephClusterFSID: <fsid>
  CephClientKey: <key>
  CephExternalMonHost: <mon ip addresses>
  CephSpecFqdn: <true/false>
  CephConfigPath: "/etc/ceph"
  DeployedCeph: false
```

- Replace **<fsid>** with the UUID of your Red Hat Ceph Storage cluster.
- Replace **<key>** with your Ceph client key.
- Replace **<mon ip addresses>** with a list of your Ceph Mon Host IPs.
- Replace **<true/false>** with the value that applies to your environment.



NOTE

If your Red Hat Ceph Storage deployment includes short names, you must set the **CephSpecFqdn** parameter to false. If set to true, the inventory generates with both the short names and domain names, causing the Red Hat Ceph Storage upgrade to fail.

5. Include the **ceph_params.yaml** file in the overcloud deployment command:

```
$ openstack overcloud deploy \
...
-e ~/environments/ceph_params.yaml \
```



IMPORTANT

Do not remove the **ceph_params.yaml** file after the RHOSP upgrade is complete. This file must be present in external Red Hat Ceph Storage environments. Additionally, any time you run **openstack overcloud deploy**, you must include the **ceph_params.yaml** file, for example, **-e ceph_params.yaml**.

Next steps

You include the **ceph_params.yaml** file in the overcloud upgrade preparation script that you create when you perform the overcloud adoption and preparation procedure. For more information, see [Performing the overcloud adoption and preparation](#).

CHAPTER 4. PREPARING FOR AN OVERCLOUD UPGRADE

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

4.1. 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

4.2. 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.



NOTE

When you complete the upgrade of your Red Hat OpenStack Platform environment, you must re-enable fencing in the overcloud. For more information about re-enabling fencing, see [Re-enabling fencing in the overcloud](#).

Procedure

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

```
$ source ~/stackrc
```

- For each Controller node, log in to the Controller node and run the Pacemaker command to disable fencing:

```
$ ssh tripleo-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 at **/etc/hosts** or **/var/lib/mistral**.
- 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](#)

4.3. UNDERCLOUD NODE DATABASE BACKUP

You can use the **openstack undercloud backup --db-only** command to create a standalone database backup that runs on the undercloud node. You can also use that backup to recover the state of the database in the event that it becomes corrupted. For more information about backing up the undercloud database, see [Creating a standalone database backup of the undercloud nodes](#) in *Red Hat OpenStack Platform 17.1 Backing up and restoring the undercloud and control plane nodes*.

4.4. UPDATING COMPOSABLE SERVICES IN CUSTOM ROLES_DATA FILES

This section contains information about new and deprecated composable services.

All nodes

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

Service	Reason
OS::TripleO::Services::CinderBackendDellEMCXTREMIOISCSI	Deprecated services.
OS::TripleO::Services::CinderBackendDellIPs	
OS::TripleO::Services::CinderBackendVRTS HyperScale	
OS::TripleO::Services::Ec2Api	Deprecated service.
OS::TripleO::Services::Fluentd	Deprecated in favour of OS::TripleO::Services::Rsyslog .
OS::TripleO::Services::FluentdAlt	Deprecated service.

Service	Reason
OS::TripleO::Services::Keepalived	Deprecated service.
OS::TripleO::Services::MistralApi OS::TripleO::Services::MistralEngine OS::TripleO::Services::MistralEventEngine OS::TripleO::Services::MistralExecutor	Deprecated services.
OS::TripleO::Services::NeutronLbaasv2Agent OS::TripleO::Services::NeutronLbaasv2Api	OpenStack Networking (neutron) Load Balancing as a Service deprecated in favour of Octavia.
OS::TripleO::Services::NeutronML2FujitsuCfab OS::TripleO::Services::NeutronML2FujitsuFossw	Deprecated services.
OS::TripleO::Services::NeutronSriovHostConfig	Deprecated service.
OS::TripleO::Services::NovaConsoleauth	This service is removed.
OS::TripleO::Services::Ntp	Deprecated in favor of OS::TripleO::Services::Timesync .
OS::TripleO::Services::OpenDaylightApi OS::TripleO::Services::OpenDaylightOvs	OpenDaylight is no longer supported.
OS::TripleO::Services::OpenShift::GlusterFS OS::TripleO::Services::OpenShift::Infra OS::TripleO::Services::OpenShift::Master OS::TripleO::Services::OpenShift::Worker	Deprecated services.
OS::TripleO::Services::PankoApi	The OpenStack Telemetry services are deprecated in favor of Service Telemetry Framework (STF) for metrics and monitoring. The legacy telemetry services are only available in RHOSP 17.1 to help facilitate the transition to STF and will be removed in a future version of RHOSP.
OS::TripleO::Services::Rear	Deprecated service.

Service	Reason
OS::TripleO::Services::SaharaApi OS::TripleO::Services::SaharaEngine	Deprecated services.
OS::TripleO::Services::SensuClient OS::TripleO::Services::SensuClientAlt	Deprecated services.
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::TripleoUI	Deprecated service.
OS::TripleO::Services::UndercloudMinionMessaging OS::TripleO::Services::UndercloudUpgradeEphemeralHeat	Deprecated services.
OS::TripleO::Services::Zaqar	Deprecated service.

Controller nodes

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

Service	Reason
OS::TripleO::Services::GlanceApilInternal	Service for the internal instance of the Image service (glance) API to provide location data to administrators and services that require it, such as the Block Storage service (cinder) and the Compute service (nova).

Compute nodes

By default, 17.1 Compute nodes run the **OS::TripleO::Services::NovaLibvirt** service. However, if you perform the RHOSP upgrade with the Compute nodes running the **OS::TripleO::Services::NovaLibvirt** service, the virtual machine instances appear as shut off. To prevent this issue, all Compute nodes that are on RHEL 8.4 must run the **OS::TripleO::Services::NovaLibvirtLegacy** service, and the container image must be based on UBI-8.

After the RHOSP upgrade, if you want to upgrade your Compute nodes to RHEL 9.2, your Compute nodes must run the **OS::TripleO::Services::NovaLibvirt** service and the container image must be based on UBI-9, or your virtual machine instances appear as shut off.

For more information about upgrading the operating system on Compute nodes, see [Upgrading all Compute nodes to RHEL 9.2](#) and [Upgrading Compute nodes to a Multi-RHEL environment](#).

4.5. 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 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 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'

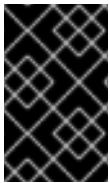
```

4.6. FINAL REVIEW BEFORE UPGRADE

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

4.6.1. Upgrade command overview

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

4.6.1.1. 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 17.1 overcloud plan and your updated environment files. This command functions similar to the **openstack overcloud deploy** command and uses many of the same options.

Before you run the **openstack overcloud upgrade prepare** command, you must perform the overcloud adoption. For more information about overcloud adoption, see [Performing the overcloud adoption and preparation](#).

4.6.1.2. openstack overcloud upgrade run

This command performs the upgrade process. Director creates a set of Ansible playbooks based on the new OpenStack Platform 17.1 overcloud plan and runs the fast forward tasks on the entire overcloud. This includes running the upgrade process through each OpenStack Platform version from 16.2 to 17.1.

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

Upgrade task tags for Leapp

system_upgrade

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

system_upgrade_prepare

Tasks to prepare for the operating system upgrade with Leapp.

system_upgrade_run

Tasks to run Leapp and upgrade the operating system.

system_upgrade_reboot

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

4.6.1.3. 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 17.1 overcloud plan and you run specific tasks using the **--tags** option.

External task tags for container management**container_image_prepare**

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

4.6.2. 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.
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 false .
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 .

Parameter	Description
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.

4.6.3. 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** command.

4.6.4. 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) 17.1.

File	Notes
/home/stack/templates/upgrades-environment.yaml	This file contains the parameters specific to the upgrade. This file is necessary only for the duration of the upgrade.
/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.
/home/stack/templates/ceph.yaml	This file contains the parameters that Ceph Storage is required to override.

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

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

4.6.5. Environment files to remove from your deployment

Remove any environment files specific to your OpenStack Platform Red Hat OpenStack Platform 16.2:

- Red Hat OpenStack Platform 16.2 container image list
- Red Hat OpenStack Platform 16.2 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 deploy**

4.6.6. Upgrading IPA services

If TLS everywhere is enabled in your environment, add an additional permission to the Nova Host Manager role to allow the creation of DNS zone entries.

Prerequisites

Check whether the Nova Host Management permission is included in your environment:

```
$ ipa privilege-show "Nova Host Management"
```

If you already have this permission, skip the following procedure.

Procedure

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

```
$ source ~/stackrc
```

3. Add the **Nova Host Management** permission:

```
$ kinit admin
$ ipa privilege-add-permission 'Nova Host Management' --permission 'System: Modify Realm Domains'
```

4. Create an environment file called **ipa_environment.yaml** and include the following configuration:

```
resource_registry:
  OS::TripleO::Services::IpaClient: /usr/share/openstack-tripleo-heat-
  templates/deployment/ipa/ipaservices-baremetal-ansible.yaml

parameter_defaults:
  IdMServer: $IPA_FQDN
  IdMDomain: $IPA_DOMAIN
  IdMInstallClientPackages: False
```

5. Save the environment file.

4.6.7. Upgrade checklist

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

Item	Complete
Validated a working overcloud.	Y / N

Item	Complete
Performed a Relax-and-Recover (ReaR) backup of the overcloud control plane. For more information, see Red Hat OpenStack Platform 16.2 Backing up and restoring the undercloud and control plane nodes .	Y / N
Created a backup of the database that runs on the undercloud node. For more information, see Creating a backup of the undercloud node in Red Hat OpenStack Platform 17.1 Backing up and restoring the undercloud and control plane nodes .	Y / N
Updated your registration details to Red Hat OpenStack Platform 17.1 repositories and converted your environment file to use the Ansible-based method.	Y / N
Updated your network configuration templates.	Y / N
Updated your environment file list with new environment files for Red Hat OpenStack Platform 17.1.	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 command.	Y / N
Removed old environment files only relevant to Red Hat OpenStack Platform 16.2, such as old Red Hat registration and container image location files.	Y / N

CHAPTER 5. OVERCLOUD ADOPTION AND PREPARATION

Perform the overcloud adoption and upgrade preparation on each stack in your environment.

For information about the duration and impact of this upgrade procedure, see [Upgrade duration and impact](#).

5.1. PERFORMING THE OVERCLOUD ADOPTION AND PREPARATION

You must perform the following tasks for overcloud adoption:

- On each stack, adopt the network and host provisioning configuration exports into the overcloud.
- Define new containers and additional compatibility configuration.

After adoption, you must run the upgrade preparation script, which performs the following tasks:

- Updates the overcloud plan to OpenStack Platform 17.1
- Prepares the nodes for the upgrade

For information about the duration and impact of this upgrade procedure, see [Upgrade duration and impact](#).

Prerequisites

- All nodes are in the **ACTIVE** state:

```
$ openstack baremetal node list
```

If any nodes are in the **MAINTENANCE** state, set them to **ACTIVE**:

```
$ openstack baremetal node maintenance unset <node_uuid>
```

- Replace **<node_uuid>** with the UUID of the node.

Procedure

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

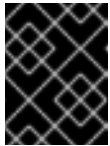
```
$ source ~/stackrc
```

3. Verify that the following files that were exported during the undercloud upgrade contain the expected configuration for the overcloud upgrade. You can find the following files in the **~/overcloud-deploy** directory:
 - **tripleo-<stack>-passwords.yaml**
 - **tripleo-<stack>-network-data.yaml**
 - **tripleo-<stack>-virtual-ips.yaml**

- **tripleo-<stack>-baremetal-deployment.yaml**

**NOTE**

If the files were not generated after the undercloud upgrade, contact Red Hat Support.

**IMPORTANT**

If you have a multi-cell environment, review [Overcloud adoption for multi-cell environments](#) for an example of copying the files to each cell stack.

4. On the main stack, copy the **passwords.yaml** file to the `~/overcloud-deploy/$(<stack>)` directory. Repeat this step on each stack in your environment:

```
$ cp ~/overcloud-deploy/<stack>/tripleo-<stack>-passwords.yaml ~/overcloud-deploy/<stack>/<stack>-passwords.yaml
```

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

5. On the main stack, copy the **network-data.yaml** file to the stack user's home directory and deploy the networks. Repeat this step on each stack in your environment:

```
$ cp ~/overcloud-deploy/<stack>/tripleo-<stack>-network-data.yaml ~/
$ mkdir ~/overcloud_adopt
$ openstack overcloud network provision --debug \
--output /home/stack/overcloud_adopt/generated-networks-deployed.yaml tripleo-<stack>-network-data.yaml
```

For more information, see [Provisioning and deploying your overcloud](#) in *Installing and managing Red Hat OpenStack Platform with director*.

6. On the main stack, copy the **virtual-ips.yaml** file to the stack user's home directory and provision the network VIPs. Repeat this step on each stack in your environment:

```
$ cp ~/overcloud-deploy/<stack>/tripleo-<stack>-virtual-ips.yaml ~/
$ openstack overcloud network vip provision --debug \
--stack <stack> --output \
/home/stack/overcloud_adopt/generated-vip-deployed.yaml tripleo-<stack>-virtual-ips.yaml
```

7. On the main stack, copy the **baremetal-deployment.yaml** file to the stack user's home directory and provision the overcloud nodes. Repeat this step on each stack in your environment:

```
$ cp ~/overcloud-deploy/<stack>/tripleo-<stack>-baremetal-deployment.yaml ~/
$ openstack overcloud node provision --debug --stack <stack> \
--output /home/stack/overcloud_adopt/baremetal-deployment.yaml \
tripleo-<stack>-baremetal-deployment.yaml
```

**NOTE**

This is the final step of the overcloud adoption. If your overcloud adoption takes longer than 10 minutes to complete, contact Red Hat Support.

8. Complete the following steps to prepare the containers:

- a. Back up the **containers-prepare-parameter.yaml** file that you used for the undercloud upgrade:

```
$ cp containers-prepare-parameter.yaml \
containers-prepare-parameter.yaml.orig
```

- b. Define the following environment variables before you run the script to update the **containers-prepare-parameter.yaml** file:

- **NAMESPACE:** The namespace for the UBI9 images. For example, **NAMESPACE="namespace":example.redhat.com:5002",'**
- **EL8_NAMESPACE:** The namespace for the UBI8 images.
- **NEUTRON_DRIVER:** The driver to use and determine which OpenStack Networking (neutron) container to use. Set to the type of containers you used to deploy the original stack. For example, set to **NEUTRON_DRIVER="neutron_driver":"ovn",'** to use OVN-based containers.
- **EL8_TAGS:** The tags of the UBI8 images, for example, **EL8_TAGS="tag":"17.1",'**.
 - Replace **"17.1"**, with the tag that you use in your content view.
- **EL9_TAGS:** The tags of the UBI9 images, for example, **EL9_TAGS="tag":"17.1",'**.
 - Replace **"17.1"**, with the tag that you use in your content view.
 - For more information about the **tag** parameter, see [Container image preparation parameters](#) in *Customizing your Red Hat OpenStack Platform deployment*.
- **CONTROL_PLANE_ROLES:** The list of control plane roles using the **--role** option, for example, **--role ControllerOpenstack, --role Database, --role Messaging, --role Networker, --role CephStorage**. To view the list of control plane roles in your environment, run the following command:

```
$ export STACK=<stack> \
$ sudo awk '/tripleo_role_name/ {print "--role " $2}' \
/var/lib/mistral/${STACK}/tripleo-ansible-inventory.yaml \
| grep -vi compute
```

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

- **COMPUTE_ROLES:** The list of Compute roles using the **--role** option, for example, **--Compute-1**. To view the list of Compute roles in your environment, run the following command:

```
$ sudo awk '/tripleo_role_name/ {print "--role " $2}' \
/var/lib/mistral/${STACK}/tripleo-ansible-inventory.yaml \
| grep -i compute
```

- **CEPH_OVERRIDE:** If you deployed Red Hat Ceph Storage, specify the Red Hat Ceph Storage 5 container images. For example: **CEPH_OVERRIDE="ceph_image":"rhceph-5-rhel8","ceph_tag":"<latest>",'**
 - Replace **<latest>** with the latest **ceph_tag** version, for example, **5-499**.

The following is an example of the **containers-prepare-parameter.yaml** file configuration:

```

NAMESPACE="namespace":'registry.redhat.io/rhosp-rhel9','
EL8_NAMESPACE="namespace":'registry.redhat.io/rhosp-rhel8','
NEUTRON_DRIVER="neutron_driver":'ovn','
EL8_TAGS="tag":'17.1','
EL9_TAGS="tag":'17.1','
CONTROL_PLANE_ROLES="--role Controller"
COMPUTE_ROLES="--role Compute"
CEPH_TAGS="ceph_tag":'5','

```

- c. Run the following script to to update the **containers-prepare-parameter.yaml** file:



WARNING

If you deployed Red Hat Ceph Storage, ensure that the **CEPH_OVERRIDE** environment variable is set to the correct values before executing the following command. Failure to do so results in issues when upgrading Red Hat Ceph Storage.

```

$ python3 /usr/share/openstack-tripleo-heat-templates/tools/multi-rhel-container-image-prepare.py \
  ${COMPUTE_ROLES} \
  ${CONTROL_PLANE_ROLES} \
  --enable-multi-rhel \
  --excludes collectd \
  --excludes nova-libvirt \
  --minor-override "
  ${EL8_TAGS}${EL8_NAMESPACE}${CEPH_OVERRIDE}${NEUTRON_DRIVER}\no_tag\":"not_used\" \
  --major-override "
  ${EL9_TAGS}${NAMESPACE}${CEPH_OVERRIDE}${NEUTRON_DRIVER}\no_tag\":"not_used\" \
  --output-env-file \
  /home/stack/containers-prepare-parameter.yaml

```

The **multi-rhel-container-image-prepare.py** script supports the following parameters:

--output-env-file

Writes the environment file that contains the default **ContainerImagePrepare** value.

--local-push-destination

Triggers an upload to a local registry.

--enable-registry-login

Enables the flag that allows the system to attempt to log in to a remote registry prior to pulling the containers. Use this flag when **--local-push-destination** is not used and the target systems have network connectivity to remote registries. Do not use this flag for an overcloud that might not have network connectivity to a remote registry.

--enable-multi-rhel

Enables multi-rhel.

--excludes

Lists the services to exclude.

--major-override

Lists the override parameters for a major release.

--minor-override

Lists the override parameters for a minor release.

--role

The list of roles.

--role-file

The **role_data.yaml** file.

- d. If you deployed Red Hat Ceph Storage, open the **containers-prepare-parameter.yaml** file to confirm that the Red Hat Ceph Storage 5 container images are specified and that there are no references to Red Hat Ceph Storage 6 container images.
9. If you have a director-deployed Red Hat Ceph Storage deployment, create a file called **ceph_params.yaml** and include the following content:

```
parameter_defaults:
  CephSpecFqdn: true
  CephConfigPath: "/etc/ceph"
  CephAnsibleRepo: "rhceph-5-tools-for-rhel-8-x86_64-rpms"
  DeployedCeph: true
```

**IMPORTANT**

Do not remove the **ceph_params.yaml** file after the RHOSP upgrade is complete. This file must be present in director-deployed Red Hat Ceph Storage environments. Additionally, any time you run **openstack overcloud deploy**, you must include the **ceph_params.yaml** file, for example, **-e ceph_params.yaml**.

**NOTE**

If your Red Hat Ceph Storage deployment includes short names, you must set the **CephSpecFqdn** parameter to **false**. If set to **true**, the inventory generates with both the short names and domain names, causing the Red Hat Ceph Storage upgrade to fail.

10. Create an environment file called **upgrades-environment.yaml** in your templates directory and include the following content:

```
parameter_defaults:
  ExtraConfig:
    nova::workarounds::disable_compute_service_check_for_ffu: true
  DnsServers: ["<dns_servers>"]
  DockerInsecureRegistryAddress: <undercloud_FQDN>
  UpgradeInitCommand: |
    sudo subscription-manager repos --disable=*
    if $( grep -q 9.2 /etc/os-release )
```

```

then
    sudo subscription-manager repos --enable=rhel-9-for-x86_64-baseos-eus-rpms --
enable=rhel-9-for-x86_64-appstream-eus-rpms --enable=rhel-9-for-x86_64-highavailability-
eus-rpms --enable=openstack-17.1-for-rhel-9-x86_64-rpms --enable=fast-datapath-for-rhel-9-
x86_64-rpms
    sudo podman ps | grep -q ceph && subscription-manager repos --enable=rhceph-5-
tools-for-rhel-9-x86_64-rpms
    sudo subscription-manager release --set=9.2
else
    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=openstack-17.1-for-rhel-8-x86_64-rpms --enable=fast-datapath-for-rhel-8-
x86_64-rpms
    sudo podman ps | grep -q ceph && subscription-manager repos --enable=rhceph-5-
tools-for-rhel-8-x86_64-rpms
    sudo subscription-manager release --set=8.4
fi

if $(sudo podman ps | grep -q ceph )
then
    sudo dnf -y install cephadm
fi

```

- Replace **<dns_servers>** with a comma-separated list of your DNS server IP addresses, for example, **["10.0.0.36", "10.0.0.37"]**.
 - Replace **<undercloud_FQDN>** with the fully qualified domain name (FQDN) of the undercloud host, for example, **"undercloud-0.ctlplane.redhat.local:8787"**.
For more information about the upgrade parameters that you can configure in the environment file, see [Upgrade parameters](#).
11. On the undercloud, create a file called **overcloud_upgrade_prepare.sh** in your templates directory. You must create this file for each stack in your environment. This file includes the original content of your overcloud deploy file and the environment files that are relevant to your environment. For example:

```

#!/bin/bash
openstack overcloud upgrade prepare --yes \
--timeout 460 \
--templates /usr/share/openstack-tripleo-heat-templates \
--ntp-server 192.168.24.1 \
--stack <stack> \
-r /home/stack/roles_data.yaml \
-e /home/stack/templates/internal.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/services/neutron-ovs.yaml \
-e /home/stack/templates/network/network-environment.yaml \
-e /home/stack/templates/inject-trust-anchor.yaml \
-e /home/stack/templates/hostnames.yml \
-e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-
ansible.yaml \
-e /home/stack/templates/nodes_data.yaml \
-e /home/stack/templates/debug.yaml \
-e /home/stack/templates/firstboot.yaml \
-e /home/stack/templates/upgrades-environment.yaml \
-e /home/stack/overcloud-params.yaml \
-e /home/stack/overcloud-deploy/overcloud/overcloud-network-environment.yaml \

```

```
-e /home/stack/overcloud_adopt/baremetal-deployment.yaml \
-e /home/stack/overcloud_adopt/generated-networks-deployed.yaml \
-e /home/stack/overcloud_adopt/generated-vip-deployed.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/nova-hw-machine-type-
upgrade.yaml \
-e /home/stack/skip_rhel_release.yaml \
-e ~/containers-prepare-parameter.yaml
```



NOTE

If you have a multi-cell environment, review [Overcloud adoption for multi-cell environments](#) for an example of creating the `overcloud_upgrade_prepare.sh` file for each cell stack.

- a. In the original `network-environment.yaml` file (`/home/stack/templates/network/network-environment.yaml`), remove all the `resource_registry` resources that point to `OS::TripleO::*::Net::SoftwareConfig`.
- b. In the `overcloud_upgrade_prepare.sh` file, include the following options relevant to your environment:
 - The environment file (`upgrades-environment.yaml`) with the upgrade-specific 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 (`skip_rhel_release.yaml`) with the release parameters (`-e`).
 - Any custom configuration environment files (`-e`) relevant to your deployment.
 - If applicable, your custom roles (`roles_data`) file by using `--roles-file`.
 - For Ceph deployments, the environment file (`ceph_params.yaml`) with the Ceph parameters (`-e`).
 - The files that were generated during overcloud adoption (`networks-deployed.yaml`, `vip-deployed.yaml`, `baremetal-deployment.yaml`) (`-e`).
 - If applicable, the environment file (`ipa-environment.yaml`) with your IPA service (`-e`).
 - If you are using composable networks, the (`network_data`) file by using `--network-file`.



NOTE

Do not include the `network-isolation.yaml` file in your overcloud deploy file or the `overcloud_upgrade_prepare.sh` file. Network isolation is defined in the `network_data.yaml` file.

- If you use a custom stack name, pass the name with the `--stack` option.

**NOTE**

You must include the **nova-hw-machine-type-upgrade.yaml** file in your templates until all of your RHEL 8 Compute nodes are upgraded to RHEL 9 in the environment. If this file is excluded, an error appears in the **nova_compute.log** in the **/var/log/containers/nova** directory. After you upgrade all of your RHEL 8 Compute nodes to RHEL 9, you can remove this file from your configuration and update the stack.

- c. In the director-deployed Red Hat Ceph Storage use case, if you enabled the Shared File Systems service (manila) with CephFS through NFS on the deployment that you are upgrading, you must specify an additional environment file at the end of the **overcloud_upgrade_prepare.sh** script file. You must add the environment file at the end of the script because it overrides another environment file that is specified earlier in the script:

```
-e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/manila-
cephfsganesha-config.yaml
```

- d. In the external Red Hat Ceph Storage use case, if you enabled the Shared File Systems service (manila) with CephFS through NFS on the deployment that you are upgrading, you must check that the associated environment file in the **overcloud_upgrade_prepare.sh** script points to the tripleo-based **ceph-nfs** role. If present, remove the following environment file:

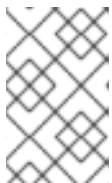
```
-e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/manila-
cephfsganesha-config.yaml
```

And add the following environment file:

```
-e /usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsganesha-
config.yaml
```

12. Run the upgrade preparation script for each stack in your environment:

```
$ source stackrc
$ chmod 755 /home/stack/overcloud_upgrade_prepare.sh
$ sh /home/stack/overcloud_upgrade_prepare.sh
```

**NOTE**

If you have a multi-cell environment, you must run the script for each **overcloud_upgrade_prepare.sh** file that you created for each cell stack. For an example, see [Overcloud adoption for multi-cell environments](#).

13. Wait until the upgrade preparation completes.

14. Download the container images:

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


5.2. OVERCLOUD ADOPTION FOR MULTI-CELL ENVIRONMENTS

Overcloud adoption involves copying the following files that were exported during the undercloud upgrade into the stack user's home directory:

- **network-data.yaml**
- **virtual-ips.yaml**
- **baremetal-deployment.yaml**

You must copy the files to the overcloud stack first and then copy them to each cell stack.



NOTE

The **network-data.yaml** file is available only on the overcloud stack. You must copy the file from the overcloud stack to all the other cell stacks.

The following example copies the **virtual-ips.yaml** file:

- Overcloud stack:

```
$ cp ~/overcloud-deploy/<overcloud>/tripleo-<overcloud>-virtual-ips.yaml ~/ \
$ cd ~/ \
$ openstack overcloud network vip provision \
--debug --stack <overcloud> \
--output /home/stack/overcloud_adopt/generated-vip-deployed.yaml \ tripleo-<overcloud>-
virtual-ips.yaml
```

- Cell stack 1:

```
$ cp ~/overcloud-deploy/<stack1>/tripleo-<stack1>-virtual-ips.yaml ~/ \
$ cd ~/ \
$ openstack overcloud network vip provision \
--debug --stack <stack1> \
--output /home/stack/overcloud_adopt/generated-<stack1>-vip-deployed.yaml \ tripleo-
<stack1>-virtual-ips.yaml
```

- Cell stack 2:

```
$ cp ~/overcloud-deploy/<stack2>/tripleo-<stack2>-virtual-ips.yaml ~/ \
$ cd ~/ \
$ openstack overcloud network vip provision \
--debug --stack <stack2> \
--output /home/stack/overcloud_adopt/generated-<stack2>-vip-deployed.yaml \ tripleo-
<stack2>-virtual-ips.yaml
```

Upgrade preparation

Performing the upgrade prepare procedure for a multi-cell environment requires the following steps:

1. Create the **overcloud_upgrade_prepare.sh** file for each cell stack, starting with the overcloud stack.

2. Include the generated output file that you created for the cell stack in the **overcloud_upgrade_prepare.sh** file. Ensure that you include the environment files that are specific to each cell stack in the **overcloud_upgrade_prepare.sh** file.
3. Run the **overcloud_upgrade_prepare.sh** script for each cell stack.

The following example adds the **generated-vip-deployed.yaml** files that were generated per cell stack during overcloud adoption:

- Overcloud stack:

```
#!/bin/bash
openstack overcloud upgrade prepare --yes \
...
-e /home/stack/templates/upgrades-environment.yaml \
-e /home/stack/overcloud-params.yaml \
-e/home/stack/overcloud_adopt/generated-vip-deployed.yaml \
...
```

Run the **overcloud_upgrade_prepare.sh** script for the overcloud stack.

- Cell stack 1:

```
#!/bin/bash
openstack overcloud upgrade prepare --yes \
...
-e /home/stack/templates/upgrades-environment.yaml \
-e /home/stack/stack1-params.yaml \
-e/home/stack/overcloud_adopt/generated-<stack1>-vip-deployed.yaml \
...
```

Run the **overcloud_upgrade_prepare.sh** script for cell stack 1.

- Cell stack 2:

```
#!/bin/bash
openstack overcloud upgrade prepare --yes \
...
-e /home/stack/templates/upgrades-environment.yaml \
-e /home/stack/stack2-params.yaml \
-e/home/stack/overcloud_adopt/generated-<stack2>-vip-deployed.yaml \
...
```

Run the **overcloud_upgrade_prepare.sh** script for cell stack 2.

For more information about the overcloud adoption and preparation process, see [Running the overcloud upgrade preparation](#).

CHAPTER 6. UPGRADING AN OVERCLOUD WITH DIRECTOR-DEPLOYED CEPH DEPLOYMENTS

If your environment includes director-deployed Red Hat Ceph Storage deployments with or without hyperconverged infrastructure (HCI) nodes, you must upgrade your deployments to Red Hat Ceph Storage 5. With an upgrade to version 5, **cephadm** now manages Red Hat Ceph Storage instead of **ceph-ansible**.

6.1. INSTALLING CEPH-ANSIBLE

If you deployed Red Hat Ceph Storage using director, you must complete this procedure. The **ceph-ansible** package is required to upgrade Red Hat Ceph Storage with Red Hat OpenStack Platform.

Procedure

1. Enable the Ceph 5 Tools repository:

```
[stack@director ~]$ sudo subscription-manager repos --enable=rhceph-5-tools-for-rhel-8-x86_64-rpms
```

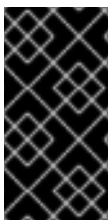
2. Install the **ceph-ansible** package:

```
[stack@director ~]$ sudo dnf update -y ceph-ansible
```

6.2. UPGRADING TO RED HAT CEPH STORAGE 5

Upgrade the following nodes from Red Hat Ceph Storage version 4 to version 5:

- Red Hat Ceph Storage nodes
- Hyperconverged infrastructure (HCI) nodes, which contain combined Compute and Ceph OSD services



IMPORTANT

Do not perform this procedure if the Red Hat Ceph Storage nodes were deployed using the **overcloud-minimal** image as described in [Using the overcloud-minimal image to avoid using a Red Hat subscription entitlement](#). This procedure will support **overcloud-minimal** image deployments when [BZ#2257738](#) is complete.

For information about the duration and impact of this upgrade procedure, see [Upgrade duration and impact](#).



NOTE

Red Hat Ceph Storage 5 uses Prometheus v4.10, which has the following known issue: If you enable Red Hat Ceph Storage dashboard, two data sources are configured on the dashboard. For more information about this known issue, see [BZ#2054852](#).

Red Hat Ceph Storage 6 uses Prometheus v4.12, which does not include this known issue. Red Hat recommends upgrading from Red Hat Ceph Storage 5 to Red Hat Ceph Storage 6 after the upgrade from Red Hat OpenStack Platform (RHOSP) 16.2 to 17.1 is complete. To upgrade from Red Hat Ceph Storage version 5 to version 6, begin with one of the following procedures for your environment:

- Director-deployed Red Hat Ceph Storage environments: [Updating the `cephadm` client](#)
- External Red Hat Ceph Storage cluster environments: [Updating the Red Hat Ceph Storage container image](#)

Procedure

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

```
$ source ~/stackrc
```

3. Run the Red Hat Ceph Storage external upgrade process with the **ceph** tag:

```
$ openstack overcloud external-upgrade run \
  --skip-tags "ceph_health,opendev-validation,ceph_ansible_remote_tmp" \
  --stack <stack> \
  --tags ceph,facts 2>&1
```

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

4. Create the ceph-admin user and distribute the appropriate keyrings:

```
ANSIBLE_LOG_PATH=/home/stack/cephadm_enable_user_key.log \
ANSIBLE_HOST_KEY_CHECKING=false \
ansible-playbook -i /home/stack/overcloud-deploy/<stack>/config-download/<stack>/tripleo-
ansible-inventory.yaml \
  -b -e ansible_python_interpreter=/usr/libexec/platform-python /usr/share/ansible/tripleo-
playbooks/ceph-admin-user-playbook.yml \
  -e tripleo_admin_user=ceph-admin \
  -e distribute_private_key=true \
  --limit ceph_osd,ceph_mon,Undercloud
```

5. Update the packages on the Red Hat Ceph Storage nodes:

```
$ openstack overcloud upgrade run \
  --stack <stack> \
  --skip-tags ceph_health,opendev-validation,ceph_ansible_remote_tmp \
  --tags setup_packages --limit ceph_osd,ceph_mon,Undercloud \
  --playbook /home/stack/overcloud-deploy/<stack>/config-
download/<stack>/upgrade_steps_playbook.yaml 2>&1
```

**NOTE**

The Ceph Monitor service (CephMon) runs on Controller nodes. This command includes the **ceph_mon** tag, which also updates the packages on the Controller nodes.

6. Configure the Red Hat Ceph Storage nodes to use **cephadm**:

```
$ openstack overcloud external-upgrade run \
  --skip-tags ceph_health,opendev-validation,ceph_ansible_remote_tmp \
  --stack <stack> \
  --tags cephadm_adopt 2>&1
```

7. Modify the **overcloud_upgrade_prepare.sh** file to replace the **ceph-ansible** file with a **cephadm** heat environment file.

```
#!/bin/bash
openstack overcloud upgrade prepare --yes \
  --timeout 460 \
  --templates /usr/share/openstack-tripleo-heat-templates \
  --ntp-server 192.168.24.1 \
  --stack <stack> \
  -r /home/stack/roles_data.yaml \
  -e /home/stack/templates/internal.yaml \
  ...
  -e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/cephadm-rbd-
only.yaml \
  -e ~/containers-prepare-parameter.yaml
```

**NOTE**

This example uses the **environments/cephadm/cephadm-rbd-only.yaml** file because RGW is not deployed. If you plan to deploy RGW, use **environments/cephadm/cephadm.yaml** after you finish upgrading your RHOSP environment, and then run a stack update.

8. Modify the **overcloud_upgrade_prepare.sh** file to remove the following environment file if you added it earlier when you ran the overcloud upgrade preparation:

```
-e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/manila-
cephsganesha-config.yaml
```

9. Save the file.

10. Run the upgrade preparation command:

```
$ source stackrc
$ chmod 755 /home/stack/overcloud_upgrade_prepare.sh
sh /home/stack/overcloud_upgrade_prepare.sh
```

11. If your deployment includes HCI nodes, create a temporary **hci.conf** file in a **cephadm** container of a Controller node:

- a. Log in to a Controller node:

```
$ ssh cloud-admin@<controller_ip>
```

- Replace **<controller_ip>** with the IP address of the Controller node.

b. Retrieve a **cephadm** shell from the Controller node:

Example

```
[cloud-admin@controller-0 ~]$ sudo cephadm shell
```

c. In the **cephadm** shell, create a temporary **hci.conf** file:

Example

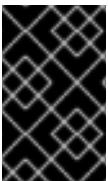
```
[ceph: root@edpm-controller-0 /]# cat <<EOF > hci.conf
[osd]
osd_memory_target_autotune = true
osd_numa_auto_affinity = true
[mgr]
mgr/cephadm/autotune_memory_target_ratio = 0.2
EOF
...
```

d. Apply the configuration:

Example

```
[ceph: root@edpm-controller-0 /]# ceph config assimilate-conf -i hci.conf
```

For more information about adjusting the configuration of your HCI deployment, see [Ceph configuration overrides for HCI](#) in *Deploying a hyperconverged infrastructure*.



IMPORTANT

You must upgrade the operating system on all HCI nodes to RHEL 9. For more information on upgrading Compute and HCI nodes, see [Upgrading Compute nodes to RHEL 9.2](#).

The Red Hat Ceph Storage cluster is now upgraded to version 5. This has the following implications:

- You no longer use **ceph-ansible** to manage Red Hat Ceph Storage. Instead, the Ceph Orchestrator manages the Red Hat Ceph Storage cluster. For more information about the Ceph Orchestrator, see [The Ceph Operations Guide](#).
- You no longer need to perform stack updates to make changes to the Red Hat Ceph Storage cluster in most cases. Instead, you can run day two Red Hat Ceph Storage operations directly on the cluster as described in [The Ceph Operations Guide](#). You can also scale Red Hat Ceph Storage cluster nodes up or down as described in [Scaling the Ceph Storage cluster](#) in *Deploying Red Hat Ceph Storage and Red Hat OpenStack Platform together with director*.
- Inspect the Red Hat Ceph Storage cluster's health. For more information about monitoring your cluster's health, see [Monitoring Red Hat Ceph Storage nodes](#) in *Deploying Red Hat Ceph Storage and Red Hat OpenStack Platform together with director*.

- Do not include environment files, such as **environments/ceph-ansible/ceph-ansible.yaml**, in openstack deployment commands such as **openstack overcloud deploy**. If your deployment includes **ceph-ansible** environment files, replace them with one of the following options:

Red Hat Ceph Storage deployment	Original ceph-ansible file	Cephadm file replacement
Ceph RADOS Block Device (RBD) only	Any ceph-ansible environment file	environments/cephadm/cephadm-rbd-only.yaml
RBD and the Ceph Object Gateway (RGW)	Any ceph-ansible environment file	environments/cephadm/cephadm.yaml
Ceph Dashboard	environments/ceph-ansible/ceph-dashboard.yaml	Respective file in environments/cephadm/
Ceph MDS	environments/ceph-ansible/ceph-mds.yaml	Respective file in environments/cephadm/

CHAPTER 7. PREPARING NETWORK FUNCTIONS VIRTUALIZATION (NFV)

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

7.1. NETWORK FUNCTIONS VIRTUALIZATION (NFV) ENVIRONMENT FILES

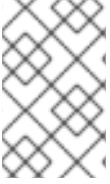
In a typical NFV-based environment, you can enable services such as the following:

- 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 17.1. However, ensure that the environment files that enable your NFV functionality meet the following requirements:

- The default environment files to enable NFV features are located in the **environments/services** directory of the Red Hat OpenStack Platform 17.1 **openstack-tripleo-heat-templates** collection. If you include the default NFV environment files from **openstack-tripleo-heat-templates** with your Red Hat OpenStack Platform 16.2 deployment, verify the correct environment file location for the respective feature in Red Hat OpenStack Platform 17.1:
 - 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 16.2 to Red Hat OpenStack Platform 17.1, you must include the **/usr/share/openstack-tripleo-heat-templates/environments/services/neutron-ovs.yaml** environment file. When running deployment and upgrade commands that involve environment files, you must include any NFV-related environment files after the **neutron-ovs.yaml** file. For example, when running **openstack overcloud upgrade prepare** with OVS and NFV environment files, include the files in the following order:
 - The OVS environment file
 - The SR-IOV environment file
 - The DPDK environment file

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


**NOTE**

There is a 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 8. UPGRADING THE OVERCLOUD

Upgrade Red Hat OpenStack Platform content across the whole overcloud on each stack in your environment.

8.1. UPGRADING RHOSP ON ALL NODES IN EACH STACK

Upgrade all overcloud nodes to Red Hat OpenStack Platform (RHOSP) 17.1 for each stack, starting with the main stack.



NOTE

You must ensure that the pacemaker is running on all controllers before you upgrade the overcloud nodes.

For information about the duration and impact of this upgrade procedure, see [Upgrade duration and impact](#).

Procedure

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

```
$ source ~/stackrc
```

3. Upgrade RHOSP on all nodes in your main stack:

```
$ openstack overcloud upgrade run --yes --stack <stack> --debug --limit  
allovercloud,undercloud --playbook all
```

- Replace `<stack>` with the name of the overcloud stack that you want to upgrade the nodes on.
Repeat this step for each stack in your RHOSP deployment.



NOTE

If you have a multi-cell environment, you must upgrade RHOSP on the cell stacks before you upgrade RHOSP on the overcloud stack.

CHAPTER 9. UPGRADING THE UNDERCLOUD OPERATING SYSTEM

You must upgrade the undercloud operating system from Red Hat Enterprise Linux 8.4 to Red Hat Enterprise Linux 9.2. The system upgrade performs the following tasks:

- Ensures that network interface naming remains consistent after the system upgrade
- Uses Leapp to upgrade RHEL in-place
- Reboots the undercloud

9.1. 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.

9.2. VALIDATING YOUR SSH KEY SIZE

Starting with Red Hat Enterprise Linux (RHEL) 9.1, a minimum SSH key size of 2048 bits is required. If your current SSH key on Red Hat OpenStack Platform (RHOSP) director is less than 2048 bits, you can lose access to the overcloud. You must verify that your SSH key meets the required bit size.

Procedure

1. Validate your SSH key size:

```
ssh-keygen -l -f /home/stack/overcloud-deploy/overcloud/ssh_private_key
```

Example output:

```
1024 SHA256:Xqz0Xz0/aJua6B3qRD7VsLr6n/V3zhmnGskcFR6FIJw
stack@director.example.local (RSA)
```

- If your SSH key is less than 2048 bits, you must rotate out the SSH key before continuing. For more information, see [Updating SSH keys in your OpenStack environment](#) in *Hardening Red Hat OpenStack Platform*.

9.3. PERFORMING THE UNDERCLOUD SYSTEM UPGRADE

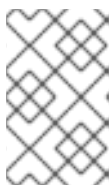
Upgrade your undercloud operating system to Red Hat Enterprise Linux (RHEL) 9.2. As part of this upgrade, you create a file named **system_upgrade.yaml**, which you use to enable the appropriate repositories and required Red Hat OpenStack Platform options and content to install Leapp. You use this file to also upgrade your control plane nodes and Compute nodes.

For information about the duration and impact of this upgrade procedure, see [Upgrade duration and impact](#).

Procedure

- Log in to the undercloud as the **stack** user.
- Create a file named **system_upgrade.yaml** in your templates directory and include the following content:

```
parameter_defaults:
  UpgradeLeappDevelSkip: "LEAPP_UNSUPPORTED=1
  LEAPP_DEVEL_SKIP_CHECK_OS_RELEASE=1 LEAPP_NO_NETWORK_RENAMING=1
  LEAPP_DEVEL_TARGET_RELEASE=9.2"
  UpgradeLeappDebug: false
  UpgradeLeappEnabled: true
  LeappActorsToRemove:
  ['checkifcfg','persistentnetnamesdisable','checkinstalledkernels','biosdevname']
  LeappRepolnitCommand: |
    subscription-manager repos --disable=*
    subscription-manager repos --enable rhel-8-for-x86_64-baseos-tus-rpms --enable rhel-8-
for-x86_64-appstream-tus-rpms --enable openstack-17.1-for-rhel-8-x86_64-rpms
    subscription-manager release --set=8.4
  UpgradeLeappCommandOptions: "--enablerepo=rhel-9-for-x86_64-baseos-eus-rpms --
enablerepo=rhel-9-for-x86_64-appstream-eus-rpms --enablerepo=rhel-9-for-x86_64-
highavailability-eus-rpms --enablerepo=openstack-17.1-for-rhel-9-x86_64-rpms --
enablerepo=fast-datapath-for-rhel-9-x86_64-rpms"
  UpgradeLeappToRemove: ['irb']
```



NOTE

If your deployment includes Red Hat Ceph Storage nodes, you must add the **CephLeappRepolnitCommand** parameter and specify the source OS version of your Red Hat Ceph Storage nodes. For example:

```
CephLeappRepolnitCommand:
...
subscription-manager release --set=8.6
```



IMPORTANT

Removing the **ruby-irb** package is mandatory to avoid a conflict between the RHEL 8 ruby-irb directory and the RHEL 9 symlink. For more information, see the Red Hat Knowledgebase solution [leapp upgrade RHEL8 to RHEL9 fails with error "rubygem-irb-1.3.5-160.el9_0.noarch conflicts with file from package ruby-irb-2.5.9-110.module+el8.6.0+15956+aa803fc1.noarch"](#).

3. Add the **LeappInitCommand** parameter to your **system_upgrade.yaml** file to specify additional requirements applicable to your environment, for example, if you need to define role-based overrides:

```
LeappInitCommand: |
  sudo subscription-manager repos --disable=*
  sudo subscription-manager repos
  --enable=rhel-9-for-x86_64-baseos-eus-rpms --enable=rhel-9-for-x86_64-appstream-eus-
  rpms --enable=rhel-9-for-x86_64-highavailability-eus-rpms --enable=openstack-17.1-for-rhel-
  9-x86_64-rpms --enable=fast-datapath-for-rhel-9-x86_64-rpms

  leapp answer --add --section check_vdo.confirm=True
```

4. If you use kernel-based NIC names, add the following parameter to the **system_upgrade.yaml** file to ensure that the NIC names persist throughout the upgrade process:

```
parameter_defaults:
  NICsPrefixesToUdev: ['en']
  ...
```

5. Run the Leapp upgrade:

```
$ openstack undercloud upgrade --yes --system-upgrade \
/home/stack/system_upgrade.yaml
```



NOTE

If you need to run the Leapp upgrade again, you must first reset the repositories to RHEL 8.

6. Reboot the undercloud:

```
$ sudo reboot
```

CHAPTER 10. UPGRADING THE CONTROL PLANE OPERATING SYSTEM

Upgrade the operating system on your control plane nodes. The upgrade includes the following tasks:

- Running the overcloud upgrade prepare command with the system upgrade parameters
- Running the overcloud system upgrade, which uses Leapp to upgrade RHEL in-place
- Rebooting the nodes

10.1. UPGRADING THE CONTROL PLANE NODES

To upgrade the control plane nodes in your environment to Red Hat Enterprise Linux 9.2, you must upgrade one-third of your control plane nodes at a time, starting with the bootstrap nodes.

You upgrade your control plane nodes by using the **openstack overcloud upgrade run** command. This command performs the following actions:

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

Each node is rebooted during the system upgrade. The performance of the Pacemaker cluster and the Red Hat Ceph Storage cluster is degraded during this downtime, but there is no outage.

This example includes the following nodes with composable roles:

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

For information about the duration and impact of this upgrade procedure, see [Upgrade duration and impact](#).

Procedure

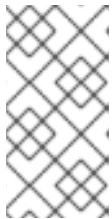
1. Log in to the undercloud host as the **stack** user.

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

```
$ source ~/stackrc
```

3. Run the following script without the **CONTROL_PLANE_ROLES** parameter. Ensure that you include the variables that you used to prepare the containers in [Running the overcloud upgrade preparation](#).

```
python3 \
/usr/share/openstack-tripleo-heat-templates/tools/multi-rhel-container-image-prepare.py \
  ${COMPUTE_ROLES} \
  --enable-multi-rhel \
  --excludes collectd \
  --excludes nova-libvirt \
  --minor-override \
  "
  ${EL8_TAGS}${EL8_NAMESPACE}${CEPH_OVERRIDE}${NEUTRON_DRIVER}"no_tag":
  \
  "not_used" \
  --major-override \
  "
  ${EL9_TAGS}${NAMESPACE}${CEPH_OVERRIDE}${NEUTRON_DRIVER}"no_tag":\
  \
  "not
  _used" \
  --output-env-file \
  /home/stack/containers-prepare-parameter.yaml
```



NOTE

The **CONTROL_PLANE_ROLES** parameter defines the list of your control plane roles. Removing this parameter from the script prepares the control plane roles for an upgrade to RHEL 9.2. If the **CONTROL_PLANE_ROLES** parameter is included in the script, the control plane roles remain on RHEL 8.4.

4. In the **skip_rhel_release.yaml** file, set the **SkipRhelEnforcement** parameter to **false**:

```
parameter_defaults:
  SkipRhelEnforcement: false
```

5. Update the **overcloud_upgrade_prepare.sh** file:

```
$ openstack overcloud upgrade prepare --yes \
...
-e /home/stack/system_upgrade.yaml \
-e /home/stack/containers-prepare-parameter.yaml \
-e /home/stack/skip_rhel_release.yaml \
...
```

- Include the **system_upgrade.yaml** file with the upgrade-specific parameters (-e).
- Include the **containers-prepare-parameter.yaml** file with the control plane roles removed (-e).
- Include the **skip_rhel_release.yaml** file with the release parameters (-e).

6. Run the **overcloud_upgrade_prepare.sh** script:

```
$ sh /home/stack/overcloud_upgrade_prepare.sh
```

7. Fetch any new or modified containers that you require for the system upgrade:

```
$ openstack overcloud external-upgrade run \
  --stack <stack> \
  --tags container_image_prepare 2>&1
```

8. Upgrade the first one-third of the control plane nodes:

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

- Replace **<stack>** with the name of your stack.
- Replace **<controller-0>,<database-0>,<messaging-0>,<networker-0>,<ceph-0>** with your own node names.

9. Log in to each upgraded node and verify that the cluster in each node is running:

```
$ sudo pcs status
```

Repeat this verification step after you upgrade the second one-third of your control plane nodes, and after you upgrade the last one-third of your control plane nodes.

10. Upgrade the second one-third of the control plane nodes:

```
$ openstack overcloud upgrade run --yes \
  --stack <stack> \
  --tags system_upgrade \
  --limit <controller-1>,<database-1>,<messaging-1>,<networker-1>,<ceph-1>
```

- Replace **<controller-1>,<database-1>,<messaging-1>,<networker-1>,<ceph-1>** with your own node names.

11. Upgrade the last one-third of the control plane nodes:

```
$ openstack overcloud upgrade run --yes \
  --stack <stack> \
  --tags system_upgrade \
  --limit <controller-2>,<database-2>,<messaging-2>,<networker-2>,<ceph-2>
```

- Replace **<controller-2>,<database-2>,<messaging-2>,<networker-2>,<ceph-2>** with your own node names.

12. If you enabled STF, run the upgrade command with no tags. Run this command after the operating system upgrade to update the **collectd** container on all nodes.

```
$ openstack overcloud upgrade run --yes \
  --stack <stack> \
```


`--limit <controller-0>,<controller-1>,<controller-2>,<database-0>,<database-1>,<database-2>,<networker-0>,<networker-1>,<networker-2>,<ceph-0>,<ceph-1>,<ceph-2>`

- Replace **<controller-0>,<controller-1>,<controller-2>,<database-0>,<database-1>,<database-2>,<networker-0>,<networker-1>,<networker-2>,<ceph-0>,<ceph-1>,<ceph-2>** with your own node names.

CHAPTER 11. UPGRADING THE COMPUTE NODE OPERATING SYSTEM

You can upgrade the operating system on all of your Compute nodes to RHEL 9.2, or upgrade some Compute nodes while the rest remain on RHEL 8.4.



IMPORTANT

If your deployment includes hyperconverged infrastructure (HCI) nodes, you must upgrade all HCI nodes to RHEL 9. For more information about upgrading to RHEL 9, see [Upgrading Compute nodes to RHEL 9.2](#).

For information about the duration and impact of this upgrade procedure, see [Upgrade duration and impact](#).

Prerequisites

- Review [Planning for a Compute node upgrade](#).

11.1. SELECTING COMPUTE NODES FOR UPGRADE TESTING

The overcloud upgrade process allows you to either:

- Upgrade all nodes in a role.
- Upgrade 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. Review which migration scenarios are supported:

Source Compute node RHEL version	Destination Compute node RHEL version	Supported/Not supported
RHEL 8	RHEL 8	Supported
RHEL 8	RHEL 9	Supported
RHEL 9	RHEL 9	Supported
RHEL 9	RHEL 8	Not supported

11.2. UPGRADING ALL COMPUTE NODES TO RHEL 9.2

Upgrade all your Compute nodes to RHEL 9.2 to take advantage of the latest features and to reduce downtime.

Prerequisites

- If your deployment includes hyper-converged infrastructure (HCI) nodes, place hosts in maintenance mode to prepare the Red Hat Ceph Storage cluster on each HCI node for reboot. For more information, see [Placing hosts in the maintenance mode using the Ceph Orchestrator in *The Ceph Operations Guide*](#).

Procedure

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

```
$ source ~/stackrc
```

3. In the **container-image-prepare.yaml** file, ensure that only the tags specified in the **ContainerImagePrepare** parameter are included, and the **MultiRhelRoleContainerImagePrepare** parameter is removed. For example:

```
parameter_defaults:
  ContainerImagePrepare:
    - tag_from_label: "{version}-{release}"
  set:
    namespace:
    name_prefix:
    name_suffix:
    tag:
    rhel_containers: false
    neutron_driver: ovn
    ceph_namespace:
    ceph_image:
    ceph_tag:
```

4. In the **roles_data.yaml** file, replace the **OS::TripleO::Services::NovaLibvirtLegacy** service with the **OS::TripleO::Services::NovaLibvirt** service that is required for RHEL 9.2.
5. Include the **-e system_upgrade.yaml** argument and the other required **-e** environment file arguments in the **overcloud_upgrade_prepare.sh** script as shown in the following example:

```
$ openstack overcloud upgrade prepare --yes
...
-e /home/stack/system_upgrade.yaml
...
```

6. Run the **overcloud_upgrade_prepare.sh** script.
7. Upgrade the operating system on the Compute nodes to RHEL 9.2. Use the **--limit** option with a comma-separated list of nodes that you want to upgrade. The following example upgrades the **compute-0**, **compute-1**, and **compute-2** nodes.

■

```
$ openstack overcloud upgrade run --yes --tags system_upgrade --stack <stack> --limit
compute-0,compute-1,compute-2
```

- Replace **<stack>** with the name of your stack.
8. Upgrade the containers on the Compute nodes to RHEL 9.2. Use the **--limit** option with a comma-separated list of nodes that you want to upgrade. The following example upgrades the **compute-0**, **compute-1**, and **compute-2** nodes.

```
$ openstack overcloud upgrade run --yes --stack <stack> --limit compute-0,compute-
1,compute-2
```

11.3. UPGRADING COMPUTE NODES TO A MULTI-RHEL ENVIRONMENT

You can upgrade a portion of your Compute nodes to RHEL 9.2 while the rest of your Compute nodes remain on RHEL 8.4. This upgrade process involves the following fundamental steps:

1. Plan which nodes you want to upgrade to RHEL 9.2, and which nodes you want to remain on RHEL 8.4. Choose a role name for each role that you are creating for each batch of nodes, for example, **ComputeRHEL-9.2** and **ComputeRHEL-8.4**.
2. Create roles that store the nodes that you want to upgrade to RHEL 9.2, or the nodes that you want to stay on RHEL 8.4. These roles can remain empty until you are ready to move your Compute nodes to a new role. You can create as many roles as you need and divide nodes among them any way you decide. For example:
 - If your environment uses a role called **ComputeSRIOV** and you need to run a canary test to upgrade to RHEL 9.2, you can create a new **ComputeSRIOVRHEL9** role and move the canary node to the new role.
 - If your environment uses a role called **ComputeOffload** and you want to upgrade most nodes in that role to RHEL 9.2, but keep a few nodes on RHEL 8.4, you can create a new **ComputeOffloadRHEL8** role to store the RHEL 8.4 nodes. You can then select the nodes in the original **ComputeOffload** role to upgrade to RHEL 9.2.
3. Move the nodes from each Compute role to the new role.
4. Upgrade the operating system on specific Compute nodes to RHEL 9.2. You can upgrade nodes in batches from the same role or multiple roles.



NOTE

In a Multi-RHEL environment, the deployment should continue to use the pc-i440fx machine type. Do not update the default to Q35. Migrating to the Q35 machine type is a separate, post-upgrade procedure to follow after all Compute nodes are upgraded to RHEL 9.2. For more information about migrating the Q35 machine type, see [Updating the default machine type for hosts after an upgrade to RHOSP 17](#).

Use the following procedures to upgrade Compute nodes to a Multi-RHEL environment:

- [Creating roles for Multi-RHEL Compute nodes](#)

- [Upgrading the Compute node operating system](#)

11.3.1. Creating roles for Multi-RHEL Compute nodes

Create new roles to store the nodes that you are upgrading to RHEL 9.2 or that are staying on RHEL 8.4, and move the nodes into the new roles.

Procedure

1. Create the relevant roles for your environment. In the **role_data.yaml** file, copy the source Compute role to use for the new role.
Repeat this step for each additional role required. Roles can remain empty until you are ready to move your Compute nodes to the new roles.

- If you are creating a RHEL 8 role:

```
name: <ComputeRHEL8>
description: |
  Basic Compute Node role
CountDefault: 1
rhsm_enforce_multios: 8.4
...
ServicesDefault:
...
- OS::TripleO::Services::NovaLibvirtLegacy
```



NOTE

Roles that contain nodes remaining on RHEL 8.4 must include the **NovaLibvirtLegacy** service.

- Replace **<ComputeRHEL8>** with the name of your RHEL 8.4 role.
- If you are creating a RHEL 9 role:

```
name: <ComputeRHEL9>
description: |
  Basic Compute Node role
CountDefault: 1
...
ServicesDefault:
...
- OS::TripleO::Services::NovaLibvirt
```



NOTE

Roles that contain nodes being upgraded to RHEL 9.2 must include the **NovaLibvirt** service. Replace **OS::TripleO::Services::NovaLibvirtLegacy** with **OS::TripleO::Services::NovaLibvirt**.

- Replace **<ComputeRHEL9>** with the name of your RHEL 9.2 role.
2. Copy the **overcloud_upgrade_prepare.sh** file to the **copy_role_Compute_param.sh** file:

```
$ cp overcloud_upgrade_prepare.sh copy_role_Compute_param.sh
```

3. Edit the **copy_role_Compute_param.sh** file to include the **copy_role_params.py** script. This script generates the environment file that contains the additional parameters and resources for the new role. For example:

```
/usr/share/openstack-tripleo-heat-templates/tools/copy_role_params.py --rolename-src
<Compute_source_role> --rolename-dst <Compute_destination_role> \
-o <Compute_new_role_params.yaml> \

-e /home/stack/templates/internal.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/services/neutron-ovs.yaml \
-e /home/stack/templates/network/network-environment.yaml \
-e /home/stack/templates/inject-trust-anchor.yaml \
-e /home/stack/templates/hostnames.yml \
-e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible.yaml
\
-e /home/stack/templates/nodes_data.yaml \
-e /home/stack/templates/debug.yaml \
-e /home/stack/templates/firstboot.yaml \
  -e /home/stack/overcloud-params.yaml \
-e /home/stack/overcloud-deploy/overcloud/overcloud-network-environment.yaml \
-e /home/stack/overcloud_adopt/baremetal-deployment.yaml \
-e /home/stack/overcloud_adopt/generated-networks-deployed.yaml \
-e /home/stack/overcloud_adopt/generated-vip-deployed.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/nova-hw-machine-type-
upgrade.yaml \
-e ~/containers-prepare-parameter.yaml
```

- Replace **<Compute_source_role>** with the name of your source Compute role that you are copying.
 - Replace **<Compute_destination_role>** with the name of your new role.
 - Use the **-o** option to define the name of the output file that includes all the non-default values of the source Compute role for the new role. Replace **<Compute_new_role_params.yaml>** with the name of your output file.
4. Run the **copy_role_Compute_param.sh** script:

```
$ sh /home/stack/copy_role_Compute_param.sh
```

5. Move the Compute nodes from the source role to the new role:

```
python3
/usr/share/openstack-tripleo-heat-templates/tools/baremetal_transition.py --baremetal-
deployment /home/stack/tripleo-<stack>-baremetal-deployment.yaml --src-role
<Compute_source_role> --dst-role <Compute_destination_role> <Compute-0> <Compute-
1> <Compute-2>
```

**NOTE**

This tool includes the original `/home/stack/tripleo-<stack>-baremetal-deployment.yaml` file that you exported during the undercloud upgrade. The tool copies and renames the source role definition in the `/home/stack/tripleo-<stack>-baremetal-deployment.yaml` file. Then, it changes the `hostname_format` to prevent a conflict with the newly created destination role. The tool then moves the node from the source role to the destination role and changes the `count` values.

- Replace `<stack>` with the name of your stack.
 - Replace `<Compute_source_role>` with the name of the source Compute role that contains the nodes that you are moving to your new role.
 - Replace `<Compute_destination_role>` with the name of your new role.
 - Replace `<Compute-0>` `<Compute-1>` `<Compute-2>` with the names of the nodes that you are moving to your new role.
6. Reprovision the nodes to update the environment files in the stack with the new role location:

```
$ openstack overcloud node provision --stack <stack> --output
/home/stack/overcloud_adopt/baremetal-deployment.yaml /home/stack/tripleo-<stack>-
baremetal-deployment.yaml
```

**NOTE**

The output `baremetal-deployment.yaml` file is the same file that is used in the `overcloud_upgrade_prepare.sh` file during overcloud adoption.

7. Include any Compute roles that are remaining on RHEL 8.4 in the `COMPUTE_ROLES` parameter, and run the following script. For example, if you have a role called `ComputeRHEL8` that contains the nodes that are remaining on RHEL 8.4, `COMPUTE_ROLES = --role ComputeRHEL8`.

```
python3
/usr/share/openstack-tripleo-heat-templates/tools/multi-rhel-container-image-prepare.py \
  ${COMPUTE_ROLES} \
  --enable-multi-rhel \
  --excludes collectd \
  --excludes nova-libvirt \
  --minor-override "
  ${EL8_TAGS}${EL8_NAMESPACE}${CEPH_OVERRIDE}${NEUTRON_DRIVER}"no_tag":
  "not_used" \
  --major-override "
  ${EL9_TAGS}${EL9_NAMESPACE}${CEPH_OVERRIDE}${NEUTRON_DRIVER}"no_tag":
  "not
  _used" \
  --output-env-file \
  /home/stack/containers-prepare-parameter.yaml
```

8. Repeat this procedure to create additional roles and to move additional Compute nodes to those new roles.

11.3.2. Upgrading the Compute node operating system

Upgrade the operating system on selected Compute nodes to RHEL 9.2. You can upgrade multiple nodes from different roles at the same time.

Prerequisites

Ensure that you have created the necessary roles for your environment. For more information about creating roles for a Multi-RHEL environment, see [Creating roles for Multi-RHEL Compute nodes](#).

Procedure

1. In the **skip_rhel_release.yaml** file, set the **SkipRhelEnforcement** parameter to **false**:

```
parameter_defaults:
  SkipRhelEnforcement: false
```

2. Include the **-e system_upgrade.yaml** argument and the other required **-e** environment file arguments in the **overcloud_upgrade_prepare.sh** script as shown in the following example:

```
$ openstack overcloud upgrade prepare --yes \
...
-e /home/stack/system_upgrade.yaml \
-e /home/stack/<Compute_new_role_params.yaml> \
...
```

- Include the **system_upgrade.yaml** file with the upgrade-specific parameters (-e).
 - Include the environment file that contains the parameters needed for the new role (-e). Replace **<Compute_new_role_params.yaml>** with the name of the environment file you created for your new role.
 - If you are upgrading nodes from multiple roles at the same time, include the environment file for each new role that you created.
3. Optional: Migrate your instances. For more information on migration strategies, see [Migrating virtual machines between Compute nodes](#) and [Preparing to migrate](#).
 4. Run the **overcloud_upgrade_prepare.sh** script.
 5. Upgrade the operating system on specific Compute nodes. Use the **--limit** option with a comma-separated list of nodes that you want to upgrade. The following example upgrades the **computerhel9-0**, **computerhel9-1**, **computerhel9-2**, and **computesriov-42** nodes from the **ComputeRHEL9** and **ComputeSRIOV** roles.

```
$ openstack overcloud upgrade run --yes --tags system_upgrade --stack <stack> --limit
computerhel9-0,computerhel9-1,computerhel9-2,computesriov-42
```

- Replace **<stack>** with the name of your stack.
6. Upgrade the containers on the Compute nodes to RHEL 9.2. Use the **--limit** option with a comma-separated list of nodes that you want to upgrade. The following example upgrades the **computerhel9-0**, **computerhel9-1**, **computerhel9-2**, and **computesriov-42** nodes from the **ComputeRHEL9** and **ComputeSRIOV** roles.


```
$ openstack overcloud upgrade run --yes --stack <stack> --limit computerhel9-0,computerhel9-1,computerhel9-2,computesriov-42
```

CHAPTER 12. 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.

12.1. 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 Red Hat OpenStack Platform software.



NOTE

You must use the new version of the overcloud images if you redeploy your overcloud. For more information on installing overcloud images, see [Installing the overcloud images](#) in *Installing and managing Red Hat OpenStack Platform with director*.

Prerequisites

- You have upgraded the undercloud to the latest version.

Procedure

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

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

- Extract the archives:

```
$ cd ~/images
$ for i in /usr/share/rhosp-director-images/overcloud-full-latest-17.1.tar /usr/share/rhosp-director-images/ironic-python-agent-latest-17.1.tar; do tar -xvf $i; done
$ cd ~
```

- Import the images into director:

```
(undercloud) [stack@director images]$ openstack overcloud image upload --image-path /home/stack/images/ --update-existing
```

The command completes the following tasks:

- Converts the image format from QCOW to RAW.
- Provides status updates about the upload of the image.

12.2. UPDATING CPU PINNING PARAMETERS

You must migrate the CPU pinning configuration from the **NovaVcpuPinSet** parameter to the following parameters after completing the upgrade to Red Hat OpenStack Platform 17.1:

NovaComputeCpuDedicatedSet

Sets the dedicated (pinned) CPUs.

NovaComputeCpuSharedSet

Sets the shared (unpinned) CPUs.

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:

- Create a new flavor that does not set the **hw:cpu_thread_policy** thread policy and resize the instances with that flavor:

- i. Source your overcloud authentication file:

```
$ source ~/overcloudrc
```

- ii. Create a flavor with the default thread policy, **prefer**:

```
(overcloud) $ openstack flavor create <flavor>
```



NOTE

When you resize an instance, you must use a new flavor. You cannot reuse the current flavor. For more information, see [Resizing an instance](#) in the *Creating and managing instances* guide.

- iii. Convert the instances to use the new flavor:

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

- iv. Repeat this step for all pinned instances that use 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](#)

12.3. UPDATING THE DEFAULT MACHINE TYPE FOR HOSTS AFTER AN UPGRADE TO RHOSP 17

The machine type of an instance is a virtual chipset that provides certain default devices, such as a PCIe graphics card or Ethernet controller. Cloud users can specify the machine type for their instances by using an image with the **hw_machine_type** metadata property that they require.

Cloud administrators can use the Compute parameter **NovaHWMachineType** to configure each Compute node architecture with a default machine type to apply to instances hosted on that architecture. If the **hw_machine_type** image property is not provided when launching the instance, the default machine type for the host architecture is applied to the instance. Red Hat OpenStack Platform (RHOSP) 17 is based on RHEL 9. The **pc-i440fx** QEMU machine type is deprecated in RHEL 9, therefore the default machine type for **x86_64** instances that run on RHEL 9 has changed from **pc** to **q35**. Based on this change in RHEL 9, the default value for machine type **x86_64** has also changed from **pc** in RHOSP 16 to **q35** in RHOSP 17.

From RHOSP 16.2 and later, the Compute service records the instance machine type within the system metadata of the instance when it launches an instance. This means that it is now possible to change the **NovaHWMachineType** during the lifetime of a RHOSP deployment without affecting the machine type of existing instances.

The Compute service records the machine type of instances that are not in a **SHELVED_OFFLOADED** state. Therefore, after an upgrade to RHOSP 17 you must manually record the machine type of instances that are in **SHELVED_OFFLOADED** state, and verify that all instances within the environment or specific cell have had a machine type recorded. After you have updated the system metadata for each instance with the machine types, you can update the **NovaHWMachineType** parameter to the RHOSP 17 default, **q35**, without affecting the machine type of existing instances.



NOTE

From RHOSP OSP17.0 onwards, Q35 is the default machine type. The Q35 machine type uses PCIe ports. You can manage the number of PCIe port devices by configuring the heat parameter **NovaLibvirtNumPciePorts**. The number of devices that can attach to a PCIe port is fewer than instances running on previous versions. If you want to use more devices, you must use the **hw_disk_bus=scsi** or **hw_scsi_model=virtio-scsi** image property. For more information, see [Metadata properties for virtual hardware](#).

Prerequisites

- Upgrade all Compute nodes to RHEL 9.2. For more information about upgrading Compute nodes, see [Upgrading all Compute nodes to RHEL 9.2](#).

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 as the **heat-admin** user:

```
(undercloud)$ metalsmith list
$ ssh heat-admin@<controller_ip>
```

Replace **<controller_ip>** with the IP address of the Controller node.

4. Retrieve the list of instances that have no machine type set:

```
[heat-admin@<controller_ip> ~]$ sudo podman exec -i -u root nova_api \
nova-manage libvirt list_unset_machine_type
```

5. Check the **NovaHWMachineType** parameter in the **nova-hw-machine-type-upgrade.yaml** file for the default machine type for the instance host. The default value for the **NovaHWMachineType** parameter in RHOSP 16.2 is as follows:

```
x86_64=pc-i440fx-rhel7.6.0,aarch64=virt-rhel7.6.0,ppc64=pseries-
rhel7.6.0,ppc64le=pseries-rhel7.6.0
```

6. Update the system metadata of each instance with the default instance machine type:

```
[heat-admin@<controller_ip> ~]$ sudo podman exec -i -u root nova_api \
nova-manage libvirt update_machine_type <instance_uuid> <machine_type>
```

- Replace **<instance_uuid>** with the UUID of the instance.
- Replace **<machine_type>** with the machine type to record for the instance.

7. Confirm that the machine type is recorded for all instances:

```
[heat-admin@<controller_ip> ~]$ sudo podman exec -i -u root nova_api \
nova-status upgrade check
```

This command returns a warning if an instance is found without a machine type. If you get this warning, repeat this procedure from step 4.

8. Change the default value of **NovaHWMachineType** in a Compute environment file to **x86_64=q35** and deploy the overcloud.

Verification

1. Create an instance that has the default machine type:

```
(overcloud)$ openstack server create --flavor <flavor> \
--image <image> --network <network> \
--wait defaultMachineTypeInstance
```

- Replace **<flavor>** with the name or ID of a flavor for the instance.
- Replace **<image>** with the name or ID of an image that does not set **hw_machine_type**.
- Replace **<network>** with the name or ID of the network to connect the instance to.

2. Verify that the instance machine type is set to the default value:

```
[heat-admin@<controller_ip> ~]$ sudo podman exec -i -u root nova_api \
nova-manage libvirt get_machine_type <instance_uuid>
```

Replace **<instance_uuid>** with the UUID of the instance.

3. Hard reboot an instance with a machine type of **x86_64=pc-i440fx**:

```
(overcloud)$ openstack server reboot --hard <instance_uuid>
```

Replace **<instance_uuid>** with the UUID of the instance.

4. Verify that the instance machine type has not been changed:

```
[heat-admin@<controller_ip> ~]$ sudo podman exec -i -u root nova_api \
nova-manage libvirt get_machine_type <instance_uuid>
```

Replace **<instance_uuid>** with the UUID of the instance.

12.4. RE-ENABLING FENCING IN THE OVERCLOUD

Before you upgraded the overcloud, you disabled fencing in [Disabling fencing in the overcloud](#). After you upgrade your environment, re-enable fencing to protect your data if a node fails.

Procedure

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

```
$ source ~/stackrc
```

3. Log in to a Controller node and run the Pacemaker command to re-enable fencing:

```
$ ssh tripleo-admin@<controller_ip> "sudo pcs property set stonith-enabled=true"
```

- 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 **true**.

Additional Resources

- [Fencing Controller nodes with STONITH](#)

CHAPTER 13. UPGRADING RED HAT CEPH STORAGE 5 TO 6

You can upgrade the Red Hat Ceph Storage cluster from Release 5 to 6 after all other upgrade tasks are completed.

Prerequisites

- The upgrade from Red Hat OpenStack Platform 16.2 to 17.1 is complete.
- All Controller nodes are upgraded to Red Hat Enterprise Linux 9. In HCI environments, all Compute nodes must also be upgraded to RHEL 9.
- The current Red Hat Ceph Storage 5 cluster is healthy.

13.1. DIRECTOR-DEPLOYED RED HAT CEPH STORAGE ENVIRONMENTS

Perform the following tasks if Red Hat Ceph Storage is director-deployed in your environment.

13.1.1. Updating the cephadm client

Before you upgrade the Red Hat Ceph Storage cluster, you must update the **cephadm** package in the overcloud nodes to Release 6.

Prerequisites

Confirm that the health status of the Red Hat Ceph Storage cluster is **HEALTH_OK**. Log in to a Controller node and use the command **sudo cephadm shell — ceph -s** to confirm the cluster health. If the status is not **HEALTH_OK**, correct any issues before continuing with this procedure.

Procedure

1. Create a playbook to enable the Red Hat Ceph Storage (tool only) repositories in the Controller nodes. It should contain the following information:

```
- hosts: all
gather_facts: false
tasks:
  - name: Enable RHCS 6 tools repo
    ansible.builtin.command: |
      subscription-manager repos --disable=rhceph-5-tools-for-rhel-9-x86_64-rpms
      subscription-manager repos --enable=rhceph-6-tools-for-rhel-9-x86_64-rpms
    become: true
  - name: Update cephadm
    ansible.builtin.package:
      name: cephadm
      state: latest
    become: true
```

2. Run the playbook:


```
ansible-playbook -i ~/overcloud-deploy/<stack>/tripleo-ansible-inventory.yaml
<playbook_file_name> --limit <controller_role>
```

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

- Replace `<playbook_file_name>` with the name of the playbook created in the previous step.
 - Replace `<controller_role>` with the role applied to Controller nodes.
 - Use the `--limit` option to apply the content to Controller nodes only.
3. Log in to a Controller node.
 4. Verify that the `cephadm` package is updated to Release 6:

```
$ sudo dnf info cephadm | grep -i version
```

13.1.2. Updating the Red Hat Ceph Storage container image

The `container-image-prepare.yaml` file contains the `ContainerImagePrepare` parameter and defines the Red Hat Ceph Storage containers. This file is used by the `tripleo-container-image prepare` command to define the rules for obtaining container images for the undercloud and overcloud. Update this file with the correct image version before updating your environment.

Procedure

1. Locate your container preparation file. The default name of this file is `containers-prepare-parameter.yaml`.
2. Edit the container preparation file.
3. Locate the `ceph_tag` parameter. The current entry should be similar to the following example:

```
ceph_namespace: registry.redhat.io
ceph_image: rhceph
ceph_tag: '5'
```

4. Update the `ceph_tag` parameter to a value of `6`:

```
ceph_namespace: registry.redhat.io
ceph_image: rhceph
ceph_tag: '6'
```

5. Save the file.

13.1.3. Running the container image prepare

Complete the container image preparation process by running the director container preparation command. This prepares all container image configurations for the overcloud and retrieves the latest Red Hat Ceph Storage 6 container image.

Procedure

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

```
$ source ~/stackrc
```

3. Run the container preparation command:
\$ openstack tripleo container image prepare -e <container_preparation_file>
 - Replace **<container_preparation_file>** with the name of your file. The default file is **containers-prepare-parameter.yaml**.
4. Verify the new Red Hat Ceph Storage image is present in the undercloud registry:
\$ openstack tripleo container image list -f value | awk -F '/' '{print \$2}'

13.1.4. Upgrading to Red Hat Ceph Storage 6 with Orchestrator

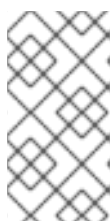
Upgrade to Red Hat Ceph Storage 6 by using the Orchestrator capabilities of the **cephadm** command.

Prerequisites

- On a Monitor or Controller node that is running the **ceph-mon** service, confirm the Red Hat Ceph Storage cluster status by using the **sudo cephadm --shell ceph status** command. This command returns one of three responses:
 - **HEALTH_OK** - The cluster is healthy. Proceed with the cluster upgrade.
 - **HEALTH_WARN** - The cluster is unhealthy. Do not proceed with the cluster upgrade before the blocking issues are resolved. For troubleshooting guidance, see [Red Hat Ceph Storage 5 Troubleshooting Guide](#).
 - **HEALTH_ERR** - The cluster is unhealthy. Do not proceed with the cluster upgrade before the blocking issues are resolved. For troubleshooting guidance, see [Red Hat Ceph Storage 5 Troubleshooting Guide](#).

Procedure

1. Log in to a Controller node.
2. Upgrade the cluster to the latest Red Hat Ceph Storage version by using [Upgrade a Red Hat Ceph Storage cluster using cephadm](#) in the *Red Hat Ceph Storage 6 Upgrade Guide*.



NOTE

The Red Hat Ceph Storage upgrade documentation describes using the **cephadm-preflight** Ansible playbook as part of the upgrade procedure. For more information about the **cephadm-preflight** playbook, see [The cephadm-ansible playbooks](#) in the *Red Hat Ceph Storage 6 Installation Guide*.

3. Create an Ansible inventory location file for the **cephadm-preflight** Ansible playbook as described in [Configuring Ansible inventory location](#) in the *Red Hat Ceph Storage 6 Installation Guide*. This file is required during the Red Hat Ceph Storage upgrade process. The **cephadm-preflight** playbook does not use an inventory created by director.
4. Wait until the Red Hat Ceph Storage container upgrade completes.



NOTE

Monitor the status upgrade by using the command **sudo cephadm shell — ceph orch upgrade status**.

13.1.5. Upgrading NFS Ganesha when moving from Red Hat Ceph Storage 5 to 6

When Red Hat Ceph Storage is upgraded from Release 4 to 5, NFS Ganesha is not adopted by the Orchestrator. This means it remains under director control and must be moved manually to Release 6.

Red Hat Ceph Storage 5 based NFS Ganesha with a Red Hat Ceph Storage 6 cluster is only supported during the upgrade period. Once you upgrade the Red Hat Ceph Storage cluster to 6, you must upgrade NFS Ganesha to use a Release 6 based container image.



NOTE

This procedure only applies to environments that are using the Shared File Systems service (manila) with CephFS NFS. Upgrading the Red Hat Ceph Storage container for NFS Ganesha is mandatory in these environments.

Procedure

1. Log in to a Controller node.
2. Inspect the **ceph-nfs** service:
\$ sudo pcs status | grep ceph-nfs
3. Inspect the **ceph-nfs systemd** unit to confirm that it contains the Red Hat Ceph Storage 5 container image and tag:

```
$ cat /etc/systemd/system/ceph-nfs@.service | grep -i container_image
```

4. Create a file called **/home/stack/ganesha_update_extravars.yml** with the following content:

```
tripleo_cephadm_container_image: <ceph_image_name>
tripleo_cephadm_container_ns: <ceph_image_namespace>
tripleo_cephadm_container_tag: <ceph_image_tag>
```

- Replace **<ceph_image_name>** with the name of the Red Hat Ceph Storage container image.
- Replace **<ceph_image_namespace>** with the name of the Red Hat Ceph Storage container namespace.
- Replace **<ceph_image_tag>** with the name of the Red Hat Ceph Storage container tag. For example, in a typical environment, this content would have the following values:

```
tripleo_cephadm_container_image: rhceph
tripleo_cephadm_container_ns: undercloud-0.ctlplane.redhat.local:8787
tripleo_cephadm_container_tag: '6'
```

5. Save the file.
6. Run the **ceph-update-ganesha.yml** playbook and provide the **ganesha_update_extravars.yml** playbook for additional command parameters:

```
ansible-playbook -i /usr/share/ansible/tripleo-playbooks/ceph-update-ganesha.yml \
-e @$HOME/overcloud-deploy/<stack>/config-download/<stack>/global_vars.yml \
-e @$HOME/overcloud-deploy/<stack>/config-download/<stack>/cephadm/cephadm-extra-
```

```
vars-heat.yml \
-e @$HOME/ganesha_update_extravars.yaml
```

- Replace **<stack>** with the name of the overcloud stack.
7. Verify that the **ceph-nfs** service is running:
\$ sudo pcs status | grep ceph-nfs
 8. Verify that the **ceph-nfs systemd** unit contains the Red Hat Ceph Storage 6 container image and tag:
\$ cat /etc/systemd/system/ceph-nfs@.service | grep rhceph

13.2. EXTERNAL RED HAT CEPH STORAGE CLUSTER ENVIRONMENT

Perform the following tasks if your Red Hat Ceph Storage cluster is external to your Red Hat OpenStack Platform deployment in your environment.

13.2.1. Updating the Red Hat Ceph Storage container image

The **container-image-prepare.yaml** file contains the **ContainerImagePrepare** parameter and defines the Red Hat Ceph Storage containers. This file is used by the **tripleo-container-image prepare** command to define the rules for obtaining container images for the undercloud and overcloud. Update this file with the correct image version before updating your environment.

Procedure

1. Locate your container preparation file. The default name of this file is **containers-prepare-parameter.yaml**.
2. Edit the container preparation file.
3. Locate the **ceph_tag** parameter. The current entry should be similar to the following example:

```
ceph_namespace: registry.redhat.io
ceph_image: rhceph
ceph_tag: '5'
```

4. Update the **ceph_tag** parameter to a value of **6**:

```
ceph_namespace: registry.redhat.io
ceph_image: rhceph
ceph_tag: '6'
```

5. Save the file.

13.2.2. Running the container image prepare

Complete the container image preparation process by running the director container preparation command. This prepares all container image configurations for the overcloud and retrieves the latest Red Hat Ceph Storage 6 container image.

Procedure

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

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

```
$ source ~/stackrc
```

3. Run the container preparation command:

```
$ openstack tripleo container image prepare -e <container_preparation_file>
```

- Replace **<container_preparation_file>** with the name of your file. The default file is **containers-prepare-parameter.yaml**.

4. Verify the new Red Hat Ceph Storage image is present in the undercloud registry:

```
$ openstack tripleo container image list -f value | awk -F '/' '{print $2}'
```

13.2.3. Upgrading NFS Ganesha when moving from Red Hat Ceph Storage 5 to 6

When Red Hat Ceph Storage is upgraded from Release 4 to 5, NFS Ganesha is not adopted by the Orchestrator. This means it remains under director control and must be moved manually to Release 6.

Red Hat Ceph Storage 5 based NFS Ganesha with a Red Hat Ceph Storage 6 cluster is only supported during the upgrade period. Once you upgrade the Red Hat Ceph Storage cluster to 6, you must upgrade NFS Ganesha to use a Release 6 based container image.



NOTE

This procedure only applies to environments that are using the Shared File Systems service (manila) with CephFS NFS. Upgrading the Red Hat Ceph Storage container for NFS Ganesha is mandatory in these environments.

Procedure

1. Log in to a Controller node.
2. Inspect the **ceph-nfs** service:
\$ sudo pcs status | grep ceph-nfs
3. Inspect the **ceph-nfs systemd** unit to confirm that it contains the Red Hat Ceph Storage 5 container image and tag:

```
$ cat /etc/systemd/system/ceph-nfs@.service | grep -i container_image
```

4. Create a file called **/home/stack/ganesha_update_extravars.yaml** with the following content:

```
tripleo_cephadm_container_image: <ceph_image_name>
tripleo_cephadm_container_ns: <ceph_image_namespace>
tripleo_cephadm_container_tag: <ceph_image_tag>
```

- Replace **<ceph_image_name>** with the name of the Red Hat Ceph Storage container image.
- Replace **<ceph_image_namespace>** with the name of the Red Hat Ceph Storage container namespace.
- Replace **<ceph_image_tag>** with the name of the Red Hat Ceph Storage container tag. For example, in a typical environment, this content would have the following values:

```
tripleo_cephadm_container_image: rhceph
tripleo_cephadm_container_ns: undercloud-0.ctlplane.redhat.local:8787
tripleo_cephadm_container_tag: '6'
```

5. Save the file.
6. Run the **ceph-update-ganesha.yml** playbook and provide the **ganesha_update_extravars.yml** playbook for additional command parameters:

```
ansible-playbook -i /usr/share/ansible/tripleo-playbooks/ceph-update-ganesha.yml \
-e @$HOME/overcloud-deploy/<stack>/config-download/<stack>/global_vars.yml \
-e @$HOME/overcloud-deploy/<stack>/config-download/<stack>/cephadm/cephadm-extra-
vars-heat.yml \
-e @$HOME/ganesha_update_extravars.yml
```

- Replace **<stack>** with the name of the overcloud stack.
7. Verify that the **ceph-nfs** service is running:
\$ sudo pcs status | grep ceph-nfs
 8. Verify that the **ceph-nfs systemd** unit contains the Red Hat Ceph Storage 6 container image and tag:
\$ cat /etc/systemd/system/ceph-nfs@.service | grep rhceph