



Red Hat OpenStack Platform 16.1

Upgrading Red Hat OpenStack Platform (15 to 16.1)

In-place upgrades from Red Hat OpenStack Platform 15 to 16.1

Red Hat OpenStack Platform 16.1 Upgrading Red Hat OpenStack Platform (15 to 16.1)

In-place upgrades from Red Hat OpenStack Platform 15 to 16.1

OpenStack Team
rhos-docs@redhat.com

Legal Notice

Copyright © 2022 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

Upgrade your OpenStack Platform environment from version 15 (Stein) to 16.1 (Train).

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE	3
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION	4
CHAPTER 1. INTRODUCTION TO UPGRADING OPENSTACK PLATFORM 15 TO 16.1	5
1.1. HIGH LEVEL WORKFLOW	5
1.2. KNOWN ISSUES THAT MIGHT BLOCK AN UPGRADE	5
CHAPTER 2. PREPARING YOUR ENVIRONMENT FOR THE UPGRADE	7
2.1. LOCKING THE ENVIRONMENT TO A RED HAT ENTERPRISE LINUX RELEASE	7
2.2. CHANGING TO EXTENDED UPDATE SUPPORT (EUS) REPOSITORIES	8
2.3. UPDATING RED HAT OPENSTACK PLATFORM AND ANSIBLE REPOSITORIES	10
2.4. SETTING THE CONTAINER-TOOLS AND VIRT MODULE VERSIONS	11
2.5. UPDATING YOUR CONTAINER IMAGE PREPARATION FILE	12
2.6. DISABLING FENCING IN THE OVERCLOUD	13
CHAPTER 3. UPGRADING THE UNDERCLOUD FROM OPENSTACK PLATFORM VERSION 15 TO 16.1	15
3.1. PERFORMING A MINOR UPDATE OF A CONTAINERIZED UNDERCLOUD	15
3.2. UNDERCLOUD POST-UPGRADE NOTES	15
3.3. NEXT STEPS	15
CHAPTER 4. UPGRADING THE OVERCLOUD FROM OPENSTACK PLATFORM VERSION 15 TO 16.1	16
4.1. PREPARING CONTAINER IMAGES FOR ROLLING UPGRADE OF HA SERVICES	16
4.2. RUNNING THE OVERCLOUD UPDATE PREPARATION	18
4.3. RUNNING THE CONTAINER IMAGE PREPARATION	18
4.4. UPDATING ALL CONTROLLER NODES	19
4.5. UPDATING ALL COMPUTE NODES	20
4.6. UPDATING ALL HCI COMPUTE NODES	20
4.7. UPDATING ALL CEPH STORAGE NODES	21
4.8. PERFORMING ONLINE DATABASE UPDATES	22
4.9. FINALIZING THE UPDATE	22
CHAPTER 5. REBOOTING THE OVERCLOUD AFTER THE UPGRADE	24
5.1. REBOOTING CONTROLLER AND COMPOSABLE NODES	24
5.2. REBOOTING A CEPH STORAGE (OSD) CLUSTER	24
5.3. REBOOTING COMPUTE NODES	25

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.

Using the Direct Documentation Feedback (DDF) function

Use the **Add Feedback** DDF function for direct comments on specific sentences, paragraphs, or code blocks.

1. View the documentation in the *Multi-page HTML* format.
2. Ensure that you see the **Feedback** button in the upper right corner of the document.
3. Highlight the part of text that you want to comment on.
4. Click **Add Feedback**.
5. Complete the **Add Feedback** field with your comments.
6. Optional: Add your email address so that the documentation team can contact you for clarification on your issue.
7. Click **Submit**.

CHAPTER 1. INTRODUCTION TO UPGRADING OPENSTACK PLATFORM 15 TO 16.1

This document provides a workflow to help you upgrade your Red Hat OpenStack Platform 15 to 16.1 and update your environment with the latest packages and containers.

This guide provides an upgrade path through the following versions:

Old OpenStack Version	New OpenStack Version
Red Hat OpenStack Platform 15.0	Red Hat OpenStack Platform 16.1

1.1. HIGH LEVEL WORKFLOW

The following table provides an outline of the tasks required for the upgrade process:

Task	Description
Preparing your environment for the upgrade	Prepare and configure the repositories, modules, and containers that are required for the upgrade.
Upgrading the undercloud	Upgrade the undercloud to the latest OpenStack Platform 16.1 version.
Upgrading the overcloud	Upgrade the overcloud to the latest OpenStack Platform 16.1 version. Upgrade all Controller, Compute, and Ceph Storage services. Run the convergence command to refresh your overcloud stack.
Rebooting the overcloud	Reboot the overcloud.

1.2. KNOWN ISSUES THAT MIGHT BLOCK AN UPGRADE

Review the following list of known issues that might affect the upgrade from OpenStack Platform 15 to 16.1.

[BZ#1895220 - Network communication problems for instances using OVN provider network](#)

Updates to 16.1.3 from an earlier version (16.0.x or 16.1.x) can cause network disruption due to a database issue with Open Virtual Network (OVN). When you perform a minor version update, you normally update Controller nodes before Compute nodes. When you update Controller nodes, director updates the **ovndb-north** database schema to the latest version. The **ovn-controller** service on Compute nodes cannot interpret the newer version of the **ovndb-north** database schema and cannot obtain the correct network flow for instances. As a workaround to minimize the network disruption, you must update the **ovn_controller** service on Compute nodes before you run the **openstack overcloud update run** command and after you run the **openstack overcloud update prepare** command. For more information, see the "[OVN update in 16.1 workaround](#)" knowledgebase article. Red Hat aims to resolve this issue in the next 16.1 minor release update.

[BZ#1872404 - restarting nodes in parallel while maintaining quorum creates an unexpected node shutdown](#)

Until this issue is resolved, for nodes based on composable roles, you must update the **Database** role first, before you can update **Controller, Messaging, Compute, Ceph**, and other roles.

CHAPTER 2. PREPARING YOUR ENVIRONMENT FOR THE UPGRADE

To upgrade your environment from Red Hat OpenStack Platform 15 to Red Hat OpenStack Platform 16.1, you must configure the correct repositories, modules, and parameters, before you can upgrade the undercloud toolset and core Heat template collection.

To prepare your undercloud and overcloud nodes for the upgrade, complete the following preparation tasks:

1. [Section 2.1, "Locking the environment to a Red Hat Enterprise Linux release"](#)
2. [Section 2.2, "Changing to Extended Update Support \(EUS\) repositories"](#)
3. [Section 2.3, "Updating Red Hat Openstack Platform and Ansible repositories"](#)
4. [Section 2.4, "Setting the container-tools and virt module versions"](#)
5. [Section 2.5, "Updating your container image preparation file"](#)
6. [Section 2.6, "Disabling fencing in the overcloud"](#)

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

Red Hat OpenStack Platform 16.1 is supported on Red Hat Enterprise Linux 8.2. Prior to performing the upgrade, lock the undercloud and overcloud repositories to the Red Hat Enterprise Linux 8.2 release to avoid upgrading the operating system to a newer minor release.

Procedure

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

```
$ source ~/stackrc
```
3. Edit your overcloud subscription management environment file, which is the file that contains the **RhsmVars** parameter. The default name for this file is usually **rhsm.yml**.
4. Check your subscription management configuration for the **rhsm_release** parameter. If this parameter is not set, add this parameter and set the parameter to 8.2:

```
parameter_defaults:
  RhsmVars:
    ...
    rhsm_username: "myusername"
    rhsm_password: "p@55w0rd!"
    rhsm_org_id: "1234567"
    rhsm_pool_ids: "1a85f9223e3d5e43013e3d6e8ff506fd"
    rhsm_method: "portal"
    rhsm_release: "8.2"
```

5. Save the overcloud subscription management environment file.

6. Create a static inventory file of your overcloud:

```
$ tripleo-ansible-inventory --ansible_ssh_user heat-admin --static-yaml-inventory
~/inventory.yaml
```

If you use an overcloud name different to the default overcloud name of **overcloud**, set the name of your overcloud with the **--plan** option.

7. Create a playbook that contains a task to lock the operating system version to Red Hat Enterprise Linux 8.2 on all nodes:

```
$ cat > ~/set_release.yaml <<'EOF'
- hosts: all
  gather_facts: false
  tasks:
    - name: set release to 8.2
      command: subscription-manager release --set=8.2
      become: true
EOF
```

8. Run the **set_release.yaml** playbook:

```
$ ansible-playbook -i ~/inventory.yaml -f 25 ~/set_release.yaml --limit
undercloud,Controller,Compute
```

Use the **--limit** option to apply the content to all Red Hat OpenStack Platform nodes. Do not run this playbook against Ceph Storage nodes because you are most likely using a different subscription for these nodes.



NOTE

To manually lock a node to a version, log in to the node and run the **subscription-manager release** command:

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

2.2. CHANGING TO EXTENDED UPDATE SUPPORT (EUS) REPOSITORIES

Your Red Hat OpenStack Platform subscription includes repositories for Red Hat Enterprise Linux 8.2 Extended Update Support (EUS). The EUS repositories include the latest security patches and bug fixes for Red Hat Enterprise Linux 8.2. Switch to the following repositories before performing an upgrade.

Standard Repository	EUS Repository
rhel-8-for-x86_64-baseos-rpms	rhel-8-for-x86_64-baseos-eus-rpms
rhel-8-for-x86_64-appstream-rpms	rhel-8-for-x86_64-appstream-eus-rpms
rhel-8-for-x86_64-highavailability-rpms	rhel-8-for-x86_64-highavailability-eus-rpms



IMPORTANT

You must use EUS repositories to retain compatibility with a specific version of Podman. Later versions of Podman are untested for this Red Hat OpenStack Platform release and can cause unexpected results.

Procedure

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

```
$ source ~/stackrc
```

3. Edit your overcloud subscription management environment file, which is the file that contains the **RhsmVars** parameter. The default name for this file is usually **rhsm.yml**.
4. Check the **rhsm_repos** parameter in your subscription management configuration. If this parameter does not include the EUS repositories, change the relevant repositories to the EUS versions:

```
parameter_defaults:
  RhsmVars:
    rhsm_repos:
      - rhel-8-for-x86_64-baseos-eus-rpms
      - rhel-8-for-x86_64-appstream-eus-rpms
      - rhel-8-for-x86_64-highavailability-eus-rpms
      - ansible-2.9-for-rhel-8-x86_64-rpms
      - advanced-virt-for-rhel-8-x86_64-rpms
      - openstack-16.1-for-rhel-8-x86_64-rpms
      - rhceph-4-tools-for-rhel-8-x86_64-rpms
      - fast-datapath-for-rhel-8-x86_64-rpms
```

5. Save the overcloud subscription management environment file.
6. Create a static inventory file of your overcloud:

```
$ tripleo-ansible-inventory --ansible_ssh_user heat-admin --static-yaml-inventory
~/inventory.yaml
```

If you use an overcloud name different to the default overcloud name of **overcloud**, set the name of your overcloud with the **--plan** option.

7. Create a playbook that contains a task to set the repositories to Red Hat Enterprise Linux 8.2 EUS on all nodes:

```
$ cat > ~/change_eus.yaml <<'EOF'
- hosts: all
  gather_facts: false
  tasks:
    - name: change to eus repos
      command: subscription-manager repos --disable=rhel-8-for-x86_64-baseos-rpms --
disable=rhel-8-for-x86_64-appstream-rpms --disable=rhel-8-for-x86_64-highavailability-rpms
--enable=rhel-8-for-x86_64-baseos-eus-rpms --enable=rhel-8-for-x86_64-appstream-eus-
```

```
rpms --enable=rhel-8-for-x86_64-highavailability-eus-rpms
become: true
EOF
```

- Run the **change_eus.yaml** playbook:

```
$ ansible-playbook -i ~/inventory.yaml -f 25 ~/change_eus.yaml --limit
undercloud,Controller,Compute
```

Use the **--limit** option to apply the content to all Red Hat OpenStack Platform nodes. Do not run this playbook against Ceph Storage nodes because you are most likely using a different subscription for these nodes.

2.3. UPDATING RED HAT OPENSTACK PLATFORM AND ANSIBLE REPOSITORIES

Update your repositories to use Red Hat OpenStack Platform 16.1 and Ansible 2.9 packages.

Procedure

- Log in to the undercloud as the **stack** user.
- Source the **stackrc** file:


```
$ source ~/stackrc
```
- Edit your overcloud subscription management environment file, which is the file that contains the **RhsmVars** parameter. The default name for this file is usually **rhsm.yml**.
- Check the **rhsm_repos** parameter in your subscription management configuration. If the **rhsm_repos** parameter is using the Red Hat OpenStack Platform 13 and Ansible 2.8 repositories, change the repository to the correct versions:

```
parameter_defaults:
  RhsmVars:
    rhsm_repos:
      - rhel-8-for-x86_64-baseos-eus-rpms
      - rhel-8-for-x86_64-appstream-eus-rpms
      - rhel-8-for-x86_64-highavailability-eus-rpms
      - ansible-2.9-for-rhel-8-x86_64-rpms
      - advanced-virt-for-rhel-8-x86_64-rpms
      - openstack-16.1-for-rhel-8-x86_64-rpms
      - fast-datapath-for-rhel-8-x86_64-rpms
```

- Save the overcloud subscription management environment file.
- Create a static inventory file of your overcloud:

```
$ tripleo-ansible-inventory --ansible_ssh_user heat-admin --static-yaml-inventory
~/inventory.yaml
```

If you use an overcloud name different to the default overcloud name of **overcloud**, set the name of your overcloud with the **--plan** option.

7. Create a playbook that contains a task to set the repositories to Red Hat OpenStack Platform 16.1 on all nodes:

```
$ cat > ~/update_rhosp_repos.yaml <<'EOF'
- hosts: all
  gather_facts: false
  tasks:
    - name: change osp repos
      command: subscription-manager repos --disable=openstack-16-for-rhel-8-x86_64-rpms --
enable=openstack-16.1-for-rhel-8-x86_64-rpms --disable=ansible-2.8-for-rhel-8-x86_64-rpms
--enable=ansible-2.9-for-rhel-8-x86_64-rpms
      become: true
EOF
```

8. Run the **update_rhosp_repos.yaml** playbook:

```
$ ansible-playbook -i ~/inventory.yaml -f 25 ~/update_rhosp_repos.yaml --limit
undercloud,Controller,Compute
```

Use the **--limit** option to apply the content to all Red Hat OpenStack Platform nodes. Do not run this playbook against Ceph Storage nodes because you are most likely using a different subscription for these nodes.

9. Create a playbook that contains a task to set the repositories to Red Hat OpenStack Platform 16.1 on all nodes:

```
$ cat > ~/update_ceph_repos.yaml <<'EOF'
- hosts: all
  gather_facts: false
  tasks:
    - name: change ceph repos
      command: subscription-manager repos --disable=openstack-16-deployment-tools-for-
rhel-8-x86_64-rpms --enable=openstack-16.1-deployment-tools-for-rhel-8-x86_64-rpms --
disable=ansible-2.8-for-rhel-8-x86_64-rpms --enable=ansible-2.9-for-rhel-8-x86_64-rpms
      become: true
EOF
```

10. Run the **update_ceph_repos.yaml** playbook:

```
$ ansible-playbook -i ~/inventory.yaml -f 25 ~/update_ceph_repos.yaml --limit CephStorage
```

Use the **--limit** option to apply the content to Ceph Storage nodes.

2.4. SETTING THE CONTAINER-TOOLS AND VIRT MODULE VERSIONS

Set the **container-tools** module to version **2.0** and the **virt** module to **8.2** to ensure you use the correct package versions on all nodes.

Procedure

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

```
$ source ~/stackrc
```

3. Create a static inventory file of your overcloud:

```
$ tripleo-ansible-inventory --ansible_ssh_user heat-admin --static-yaml-inventory
~/inventory.yaml
```

If you use an overcloud name different to the default overcloud name of **overcloud**, set the name of your overcloud with the **--plan** option.

4. Create a playbook that contains a task to set the **container-tools** module to version **2.0** on all nodes:

```
$ cat > ~/container-tools.yaml <<'EOF'
- hosts: all
  gather_facts: false
  tasks:
    - name: disable default dnf module for container-tools
      command: dnf module disable -y container-tools:rhel8
      become: true
    - name: set dnf module for container-tools:2.0
      command: dnf module enable -y container-tools:2.0
      become: true

- hosts: undercloud,Compute,Controller
  gather_facts: false
  tasks:
    - name: disable default dnf module for virt
      command: dnf module disable -y virt:rhel
      become: true
    - name: disable 8.1 dnf module for virt
      command: dnf module disable -y virt:8.1
      become: true
    - name: set dnf module for virt:8.2
      command: dnf module enable -y virt:8.2
      become: true
EOF
```

5. Run the **container-tools.yaml** playbook against all nodes:

```
$ ansible-playbook -i ~/inventory.yaml -f 25 ~/container-tools.yaml
```

2.5. UPDATING YOUR CONTAINER IMAGE PREPARATION FILE

Your container preparation file is the file that contains the **ContainerImagePrepare** parameter. You use this file to define the rules for obtaining container images for the undercloud and overcloud. Before you upgrade your environment, check the file to ensure you obtain the correct image versions.

Procedure

1. Edit the container preparation file. The default name for this file is usually **containers-prepare-parameter.yaml**.

2. Set the **tag** parameter to **16.1** and update the **namespace** parameter to **registry.redhat.io/rhosp-rhel8**:

```
parameter_defaults:
  ContainerImagePrepare:
    - push_destination: true
  set:
    namespace: registry.redhat.io/rhosp-rhel8
    ...
    tag: '16.1'
    tag_from_label: '{version}-{release}'
```



NOTE

If you do not want to use a specific tag for the update, such as **16.1** or **16.1.2**, remove the **tag** key-value pair and specify **tag_from_label** only. This will use the installed Red Hat OpenStack Platform version when determining the value for the tag to use as part of the update process.

1. Save this file.

2.6. DISABLING FENCING IN THE OVERCLOUD

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

If fencing is deployed in your environment during the Controller nodes upgrade process, 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

To re-enable fencing in your overcloud, include the **fencing.yaml** environment file when you run the **openstack overcloud update prepare** command. Director enables fencing in your overcloud when you create the new Controller node cluster.

Procedure

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

```
$ source ~/stackrc
```

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

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

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

Additional Resources

- [Fencing Controller nodes with STONITH](#)

CHAPTER 3. UPGRADING THE UNDERCLOUD FROM OPENSTACK PLATFORM VERSION 15 TO 16.1

To upgrade the undercloud from OpenStack Platform version 15 to 16.1, you must perform a procedure that is used for a minor update of a containerized undercloud. During this procedure you update the undercloud toolset, core Heat template collection, and the undercloud environment.

3.1. PERFORMING A MINOR UPDATE OF A CONTAINERIZED UNDERCLOUD

Director provides commands that you can use to update the main packages on the undercloud node. This allows you to upgrade your OpenStack Platform version 15 environment to version 16.1.

Procedure

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

2. Run **dnf** to upgrade the director main packages:

```
$ sudo dnf update -y python3-tripleoclient* tripleo-ansible ansible
```

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

```
$ openstack undercloud upgrade
```

4. Wait until the undercloud upgrade process completes.
5. Reboot the undercloud to update the operating system's kernel and other system packages:

```
$ sudo reboot
```

6. Wait until the node boots.

3.2. UNDERCLOUD POST-UPGRADE NOTES

- If you use a local set of core templates in your **stack** users home directory, ensure you update the templates using the recommended workflow in [Using Customized Core Heat Templates](#) in the *Advanced Overcloud Customization* guide. You must update the local copy before upgrading the overcloud.

3.3. NEXT STEPS

The undercloud upgrade is complete. You can now upgrade the overcloud.

CHAPTER 4. UPGRADING THE OVERCLOUD FROM OPENSTACK PLATFORM VERSION 15 TO 16.1

To upgrade the overcloud, you must update the overcloud plan, prepare the nodes for the upgrade, prepare all container image configuration that applies to your environment, and upgrade the nodes.



IMPORTANT

Until [BZ#1872404](#) is resolved, for nodes based on composable roles, you must update the **Database** role first, before you can update **Controller, Messaging, Compute, Ceph**, and other roles.

Prerequisites

- You have upgraded the undercloud node from OpenStack Platform version 15 to 16.1.

To upgrade the overcloud, complete the following tasks:

1. [Section 4.1, "Preparing container images for rolling upgrade of HA services"](#)
2. [Section 4.2, "Running the overcloud update preparation"](#)
3. [Section 4.3, "Running the container image preparation"](#)
4. [Section 4.4, "Updating all Controller nodes"](#)
5. [Section 4.5, "Updating all Compute nodes"](#)
6. [Section 4.6, "Updating all HCI Compute nodes"](#)
7. [Section 4.7, "Updating all Ceph Storage nodes"](#)
8. [Section 4.8, "Performing online database updates"](#)
9. [Section 4.9, "Finalizing the update"](#)

4.1. PREPARING CONTAINER IMAGES FOR ROLLING UPGRADE OF HA SERVICES

When a change in the **namespace**, **name_prefix** or **name_suffix** occurs for a container image used by a HA service, Pacemaker automatically restarts all the instances of this service in the control plane to recreate containers with the newly configured container image name.

Because Red Hat OpenStack 15 and 16.1 use different **name_prefix** values, after the first node in the cluster is updated, the remaining nodes start using the new container image and expect it to be present locally.

To allow a rolling update of HA container image names, Red Hat OpenStack 16.1 introduces the **ClusterCommonTag** heat parameter to configure HA services to use an intermediate image name with a fixed **name_prefix** that is **cluster.common.tag** and a fixed **name_suffix** that is **pcmklatest**. Every time director pulls new container images on a node, it updates the intermediate image tags so that they point to the newly pulled images.

To transition to the OpenStack 16.1 intermediate container image naming scheme, you must create the initial container tags manually for each HA container image present on the controller nodes. The new

cluster.common.tag/rhosp16-openstack-* container tag will point to the same container image ID that is referenced by the original **registry/rhosp15-openstack-*:pcmklatest** tag.

Procedure

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

```
$ source ~/stackrc
```

3. Create a static inventory file of your overcloud:

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

4. Create the script that tags the existing image with the new name:

```
cat > pcmkr_common_tag.sh <<'EOF'
#!/bin/sh
# Due to a change in internal CI repo, we need to adjust
# the HA relate containers before running the update.
# See bz#1846042/PIDONE
OLD_PREFIX=${1:-"openstack-"}
NEW_PREFIX=${2:-"openstack-"}
TRANSFORM=s%${OLD_PREFIX}%${NEW_PREFIX}%p

# Get all images used by HA containers (disregards images with cluster common tag in their
# name)
IMAGES=$(sudo podman images | awk '$1 !~ /cluster.common.tag/ && $2 ~ /pcmklatest/
{print $1}')

if [ -n "$IMAGES" ]; then
    echo "Creating cluster common tags and linking them to current HA images"
fi

# i: 192.168.24.1:8787/rhosp15-rhel8/openstack-mariadb
# image: openstack-mariadb
# transformed: openstack-mariadb
# full_i: 192.168.24.1:8787/rhosp15-rhel8/openstack-mariadb:pcmklatest
# full_icommon: cluster.common.tag/openstack-mariadb:pcmklatest
for i in $IMAGES; do
    image=$(echo "$i" | sed 's%.*/%%')
    transformed=$(echo $image | sed -n $TRANSFORM)
    full_i=$i:pcmklatest
    full_icommon=cluster.common.tag/${transformed}:pcmklatest
    # echo $i - $TRANSFORM "=>" $image - $transformed

    # original image points to a image hash, create a new tag
    # with cluster.common.tag and make it point to the same image hash
    echo $full_i "-->" $full_icommon
    sudo podman tag $full_i $full_icommon
done
EOF
```

5. To tag the existing image with the new name, run the script:

```
$ ansible -i inventory.yaml 'overcloud' -m script -a './pcmk_r_common_tag.sh'
```

4.2. RUNNING THE OVERCLOUD UPDATE PREPARATION

To prepare the overcloud for the update process, run the **openstack overcloud update prepare** command, which performs the following tasks:

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

Prerequisites

- If you use a Ceph subscription and have configured director to use the **overcloud-minimal** image for Ceph storage nodes, you must ensure that in the **roles_data.yaml** role definition file, the **rhsm_enforce** parameter is set to **False**.

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

2. Run the update preparation command:

```
$ openstack overcloud update prepare \
  --templates \
  --stack <stack_name> \
  -r <roles_data_file> \
  -n <network_data_file> \
  -e <environment_file> \
  -e <environment_file> \
  ...
```

Include the following options relevant to your environment:

- If the name of your overcloud stack is different to the default name **overcloud**, include the **--stack** option in the update preparation command and replace **<stack_name>** with the name of your stack.
 - If using your own custom roles, include your custom roles (**<roles_data> file (-r)**)
 - If using custom networks, include your composable network (**<network_data> file (-n)**)
 - Any custom configuration environment files (**-e**)
3. Wait until the update preparation completes.

4.3. RUNNING THE CONTAINER IMAGE PREPARATION

The overcloud requires the latest OpenStack Platform 16.1 container images before performing the update. This involves executing the **container_image_prepare** external update process. To execute this process, you must run the **openstack overcloud external-update run** command against tasks tagged with the **container_image_prepare** tag. These tasks perform the following actions:

- Automatically prepare all container image configuration relevant to your environment.
- Pull the relevant container images to your undercloud, unless you have previously disabled this option.



NOTE

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

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

2. Run the **openstack overcloud external-update run** command against tasks tagged with the **container_image_prepare** tag:

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

4.4. UPDATING ALL CONTROLLER NODES

This process updates all the Controller nodes to the latest OpenStack Platform 16.1 version. The process involves running the **openstack overcloud update run** command and including the **--limit Controller** option to restrict operations to the Controller nodes only. The control plane API is fully available during the minor update.



IMPORTANT

Until [BZ#1872404](#) is resolved, for nodes based on composable roles, you must update the **Database** role first, before you can update **Controller, Messaging, Compute, Ceph**, and other roles.



NOTE

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

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

2. Run the update command:

```
$ openstack overcloud update run --stack <stack_name> --limit Controller --playbook all
```

3. Wait until the Controller node update completes.

4.5. UPDATING ALL COMPUTE NODES

This process updates all Compute nodes to the latest OpenStack Platform 16.1 version. The process involves running the **openstack overcloud update run** command and including the **--limit Compute** option to restrict operations to the Compute nodes only.

Parallelization considerations

When you update a large number of Compute nodes, to improve performance, you can run the **openstack overcloud update run** command with the **--limit Compute** option in parallel on batches of 20 nodes. For example, if you have 80 Compute nodes in your deployment, you can run the following commands to update the Compute nodes in parallel:

```
$ openstack overcloud update run -y --limit 'Compute[0:19]' > update-compute-0-19.log 2>&1 &
$ openstack overcloud update run -y --limit 'Compute[20:39]' > update-compute-20-39.log 2>&1 &
$ openstack overcloud update run -y --limit 'Compute[40:59]' > update-compute-40-59.log 2>&1 &
$ openstack overcloud update run -y --limit 'Compute[60:79]' > update-compute-60-79.log 2>&1 &
```

The '**Compute[0:19]**', '**Compute[20:39]**', '**Compute[40:59]**', and '**Compute[60:79]**' way of partitioning the nodes space is random and you don't have control over which nodes are updated.

To update specific Compute nodes, list the nodes that you want to update in a batch separated by a comma:

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



NOTE

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

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

2. Run the update command:

```
$ openstack overcloud update run --stack <stack_name> --limit Compute --playbook all
```

3. Wait until the Compute node update completes.

4.6. UPDATING ALL HCI COMPUTE NODES

This process updates the Hyperconverged Infrastructure (HCI) Compute nodes. The process involves:

- Running the **openstack overcloud update run** command and including the **--limit ComputeHCI** option to restrict operations to the HCI nodes only.
- Running the **openstack overcloud external-update run --tags ceph** command to perform an update to a containerized Red Hat Ceph Storage 4 cluster.

**NOTE**

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

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

2. Run the update command:

```
$ openstack overcloud update run --stack <stack_name> --limit ComputeHCI --playbook all
```

3. Wait until the node update completes.

4. Run the Ceph Storage update command. For example:

```
$ openstack overcloud external-update run --stack <stack_name> --tags ceph
```

5. Wait until the Compute HCI node update completes.

4.7. UPDATING ALL CEPH STORAGE NODES

This process updates the Ceph Storage nodes. The process involves:

- Running the **openstack overcloud update run** command and including the **--limit CephStorage** option to restrict operations to the Ceph Storage nodes only.
- Running the **openstack overcloud external-update run** command to run **ceph-ansible** as an external process and update the Red Hat Ceph Storage 3 containers.

**NOTE**

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

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

2. Run the update command:

```
$ openstack overcloud update run --stack <stack_name> --limit CephStorage --playbook all
```

3. Wait until the node update completes.

4. Run the Ceph Storage container update command:

```
$ openstack overcloud external-update run --tags ceph
```

5. Wait until the Ceph Storage container update completes.

4.8. PERFORMING ONLINE DATABASE UPDATES

Some overcloud components require an online upgrade (or migration) of their databases tables. This involves executing the **online_upgrade** external update process. To execute this process, run the **openstack overcloud external-update run** command against tasks tagged with the **online_upgrade** tag. This performs online database updates to the following components:

- OpenStack Block Storage (cinder)
- OpenStack Compute (nova)

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

2. Run the **openstack overcloud external-update run** command against tasks that use the **online_upgrade** tag:

```
$ openstack overcloud external-update run --tags online_upgrade
```

4.9. FINALIZING THE UPDATE

The update requires a final step to update the overcloud stack. This ensures the stack's resource structure aligns with a regular deployment of OpenStack Platform 16.1 and allows you to perform standard **openstack overcloud deploy** functions in the future.

Procedure

1. Source the **stackrc** file:

```
$ source ~/stackrc
```

2. Run the update finalization command:

```
$ openstack overcloud update converge \
  --templates \
  --stack <stack_name> \
  -r <roles_data_file> \
  -n <network_data_file> \
  -e <environment_file> \
  -e <environment_file> \
  ...
  ...
```

Include the following options relevant to your environment:

- If the name of your overcloud stack is different to the default name **overcloud**, include the **--stack** option in the update preparation command and replace **<stack_name>** with the name of your stack.

- If using your own custom roles, include your custom roles (**<roles_data>**) file (**-r**)
 - If using custom networks, include your composable network (**<network_data>**) file (**-n**)
 - Any custom configuration environment files (**-e**).
3. Wait until the update finalization completes.

CHAPTER 5. REBOOTING THE OVERCLOUD AFTER THE UPGRADE

After you have upgraded your Red Hat OpenStack environment from version 15 to 16.1, you must reboot your overcloud. The reboot refreshes the nodes with any associated kernel, system-level, and container component updates that provide performance and security benefits.

Plan downtime to perform the following reboot procedures.

To reboot your overcloud, complete the following tasks:

1. [Section 5.1, “Rebooting Controller and composable nodes”](#)
2. [Section 5.2, “Rebooting a Ceph Storage \(OSD\) cluster”](#)
3. [Section 5.3, “Rebooting Compute nodes”](#)

5.1. REBOOTING CONTROLLER AND COMPOSABLE NODES

Complete the following steps to reboot Controller nodes and standalone nodes based on composable roles, excluding Compute nodes and Ceph Storage nodes.

Procedure

1. Log in to the node that you want to reboot.
2. Optional: If the node uses Pacemaker resources, stop the cluster:

```
[heat-admin@overcloud-controller-0 ~]$ sudo pcs cluster stop
```

3. Reboot the node:

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

4. Wait until the node boots.
5. Check the services. For example:
 - a. If the node uses Pacemaker services, check that the node has rejoined the cluster:

```
[heat-admin@overcloud-controller-0 ~]$ sudo pcs status
```

- b. If the node uses Systemd services, check that all services are enabled:

```
[heat-admin@overcloud-controller-0 ~]$ sudo systemctl status
```

- c. If the node uses containerized services, check that all containers on the node are active:

```
[heat-admin@overcloud-controller-0 ~]$ sudo podman ps
```

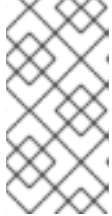
5.2. REBOOTING A CEPH STORAGE (OSD) CLUSTER

Complete the following steps to reboot a cluster of Ceph Storage (OSD) nodes.

Procedure

1. Log into a Ceph MON or Controller node and disable Ceph Storage cluster rebalancing temporarily:

```
$ sudo podman exec -it ceph-mon-controller-0 ceph osd set noout
$ sudo podman exec -it ceph-mon-controller-0 ceph osd set norebalance
```



NOTE

If you have a multistack or distributed compute node (DCN) architecture, you must specify the cluster name when you set the **noout** and **norebalance** flags. For example: **sudo podman exec -it ceph-mon-controller-0 ceph osd set noout --cluster <cluster_name>**

2. Select the first Ceph Storage node that you want to reboot and log in to the node.
3. Reboot the node:

```
$ sudo reboot
```

4. Wait until the node boots.
5. Log into the node and check the cluster status:

```
$ sudo podman exec -it ceph-mon-controller-0 ceph status
```

Check that the **pgmap** reports all **pgs** as normal (**active+clean**).

6. Log out of the node, reboot the next node, and check its status. Repeat this process until you have rebooted all Ceph storage nodes.
7. When complete, log into a Ceph MON or Controller node and re-enable cluster rebalancing:

```
$ sudo podman exec -it ceph-mon-controller-0 ceph osd unset noout
$ sudo podman exec -it ceph-mon-controller-0 ceph osd unset norebalance
```



NOTE

If you have a multistack or distributed compute node (DCN) architecture, you must specify the cluster name when you unset the **noout** and **norebalance** flags. For example: **sudo podman exec -it ceph-mon-controller-0 ceph osd set noout --cluster <cluster_name>**

8. Perform a final status check to verify that the cluster reports **HEALTH_OK**:

```
$ sudo podman exec -it ceph-mon-controller-0 ceph status
```

5.3. REBOOTING COMPUTE NODES

Complete the following steps to reboot Compute nodes. To ensure minimal downtime of instances in your Red Hat OpenStack Platform environment, this procedure also includes instructions about migrating instances from the Compute node that you want to reboot. This involves the following workflow:

- Decide whether to migrate instances to another Compute node before rebooting the node.
- Select and disable the Compute node you want to reboot so that it does not provision new instances.
- Migrate the instances to another Compute node.
- Reboot the empty Compute node.
- Enable the empty Compute node.

Prerequisites

Before you reboot the Compute node, you must decide whether to migrate instances to another Compute node while the node is rebooting.

Review the list of migration constraints that you might run into when migrating virtual machine instances between Compute nodes. For more information, see [Migration constraints](#) in *Configuring the Compute Service for Instance Creation*.

If you cannot migrate the instances, you can set the following core template parameters to control the state of the instances after the Compute node reboots:

NovaResumeGuestsStateOnHostBoot

Determines whether to return instances to the same state on the Compute node after reboot. When set to **False**, the instances remain down and you must start them manually. Default value is: **False**

NovaResumeGuestsShutdownTimeout

Number of seconds to wait for an instance to shut down before rebooting. It is not recommended to set this value to **0**. Default value is: 300

For more information about overcloud parameters and their usage, see [Overcloud Parameters](#).

Procedure

1. Log in to the undercloud as the **stack** user.
2. List all Compute nodes and their UUIDs:

```
$ source ~/stackrc
(undercloud) $ openstack server list --name compute
```

Identify the UUID of the Compute node that you want to reboot.

3. From the undercloud, select a Compute node. Disable the node:

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

4. List all instances on the Compute node:

-

```
(overcloud) $ openstack server list --host <hostname> --all-projects
```

5. If you decide not to migrate instances, skip to [this step](#).
6. If you decide to migrate the instances to another Compute node, use one of the following commands:

- Migrate the instance to a different host:

```
(overcloud) $ openstack server migrate <instance_id> --live <target_host> --wait
```

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

```
(overcloud) $ nova live-migration <instance_id>
```

- Live migrate all instances at once:

```
$ nova host-evacuate-live <hostname>
```



NOTE

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

7. Wait until migration completes.
8. Confirm that the migration was successful:

```
(overcloud) $ openstack server list --host <hostname> --all-projects
```

9. Continue to migrate instances until none remain on the chosen Compute node.

10. Log in to the Compute node and reboot the node:

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

11. Wait until the node boots.
12. Re-enable the Compute node:

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

13. Check that the Compute node is enabled:

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