



Red Hat OpenStack Platform 16.2

保持 Red Hat OpenStack Platform 更新

对 Red Hat OpenStack Platform 执行次要更新

Red Hat OpenStack Platform 16.2 保持 Red Hat OpenStack Platform 更新

对 Red Hat OpenStack Platform 执行次要更新

OpenStack Team
rhos-docs@redhat.com

法律通告

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.

摘要

您可以对 Red Hat OpenStack Platform (RHOSP) 环境执行次要更新，使其使用最新的软件包和容器进行更新。

目录

使开源包含更多	3
对红帽文档提供反馈	4
第 1 章 准备次版本更新	5
已知的可能会阻止更新的问题	5
流程	7
1.1. 长生命版本的升级路径	8
1.2. 将环境锁定到 RED HAT ENTERPRISE LINUX 版本	9
1.3. 切换到 TUS 软件仓库	11
1.4. 更新 RED HAT OPENSTACK PLATFORM 和 ANSIBLE 存储库	13
1.5. 设置 CONTAINER-TOOLS 模块版本	14
1.6. 更新容器镜像准备文件	15
1.7. 更新 SSL/TLS 配置	15
1.8. 在 OVERCLOUD 中禁用隔离	16
第 2 章 更新 UNDERCLOUD	18
先决条件	18
2.1. 对容器化 UNDERCLOUD 执行次要更新	18
2.2. 更新 OVERCLOUD 镜像	18
第 3 章 更新 OVERCLOUD	20
先决条件	20
流程	20
3.1. 运行 OVERCLOUD 更新准备	20
3.2. 运行容器镜像准备	21
3.3. 可选：在所有 OVERCLOUD 服务器上更新 OVN-CONTROLLER 容器	22
3.4. 更新所有 CONTROLLER 节点	22
3.5. 更新所有 COMPUTE 节点	23
3.6. 更新所有 HCI COMPUTE 节点	24
3.7. 更新所有 DISTRIBUTEDCOMPUTEHCI 节点	24
3.8. 更新所有 CEPH STORAGE 节点	25
3.9. 执行在线数据库更新	26
3.10. 最终更新	27
第 4 章 重新引导 OVERCLOUD	29
4.1. 重新引导 CONTROLLER 和可组合节点	29
4.2. 重新引导 CEPH STORAGE (OSD) 集群	30
4.3. 重新引导 COMPUTE 节点	31

使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。详情请查看 [CTO Chris Wright 的信息](#)。

对红帽文档提供反馈

我们感谢您对文档提供反馈信息。与我们分享您的成功秘诀。

在 JIRA 中提供文档反馈

使用 [Create Issue](#) 表单对文档提供反馈。JIRA 问题将在 Red Hat OpenStack Platform Jira 项目中创建，您可以在其中跟踪您的反馈进度。

1. 确保您已登录到 JIRA。如果您没有 JIRA 帐户，请创建一个帐户来提交反馈。
2. 点击以下链接打开 **Create Issue** 页面：[Create Issue](#)
3. 完成 **Summary** 和 **Description** 字段。在 **Description** 字段中，包含文档 URL、章节或章节号以及问题的详细描述。不要修改表单中的任何其他字段。
4. 点 **Create**。

第 1 章 准备次版本更新

使用最新的软件包和容器保持 Red Hat OpenStack Platform (RHOSP) 16.2 环境进行更新。

您可以更新以下版本：

旧的 RHOSP 版本	新的 RHOSP 版本
Red Hat OpenStack Platform 16.1.z	Red Hat OpenStack Platform 16.2 最新
Red Hat OpenStack Platform 16.2.z	Red Hat OpenStack Platform 16.2 最新

RHOSP 次要更新过程 workflow

您必须完成以下步骤来更新 RHOSP 环境：

1. 为 RHOSP 次要更新准备您的环境。
2. 将 undercloud 更新至最新的 OpenStack 16.2.z 版本。
3. 将 overcloud 更新至最新的 OpenStack 16.2.z 版本。
4. 升级所有 Red Hat Ceph Storage 服务。
5. 运行聚合命令以刷新 overcloud 堆栈。

如果您有多堆栈基础架构，请一次完全更新每个 overcloud 堆栈。如果您有分布式计算节点(DCN)基础架构，请在中央位置完全更新 overcloud，然后在每个边缘站点更新 overcloud，一次更新 overcloud。

在更新 RHOSP 环境前的注意事项

要帮助您在更新过程中指南，请考虑以下信息：

- 红帽建议备份 undercloud 和 overcloud control plane。有关备份节点的更多信息，请参阅 [备份和恢复 undercloud 和 control plane 节点](#)。
- 熟悉可能会阻止更新的已知问题。
- 在开始更新前，熟悉可能的更新和升级路径。更多信息请参阅 [第 1.1 节“长生命版本的升级路径”](#)。
- 要识别您当前的维护发行版本，请运行 `$ cat /etc/rhosp-release`。您还可以在更新环境后运行此命令以验证更新。

已知的可能会阻止更新的问题

查看可能影响成功次版本更新的已知问题。

[BZ39\) 73660 - \(更新\) 从 16.1 升级到 16.2 会破坏尝试配置 rabbitmq 服务。](#)

运行 Pacemaker 版本 **2.0.3-5.el8_2.4** 的 overcloud 节点可能无法成功更新，因为节点上关闭集群时出现的竞争条件。

如果当前在任何 overcloud 节点上安装了 Pacemaker 版本 **2.0.3-5.el8_2.4**，则必须升级 Pacemaker，然后才能更新 overcloud 节点。如需更多信息，请参阅以下红帽知识库解决方案 [从 OSP16.1 更新至 OSP16.2 可能无法更新某些 HA 容器](#)。

BZ the240 - 当启用 tsx 标志时，从 16.1 更新至 16.2 时，计算节点会在更新和 ping 丢失期间重启

从 Red Hat Enterprise Linux (RHEL) 版本 8.3 开始，默认禁用对 Intel 事务同步扩展 (TSX) 功能的支持。这会导致以下迁移场景中在主机间实例实时迁移问题：

- 从启用了 TSX 内核参数的主机迁移到禁用 TSX 内核参数的主机。

在支持 TSX 功能的 Intel 主机中，实时迁移可能会失败。有关受此问题影响的 CPU 的更多信息，请参阅[受影响的配置](#)。

有关更多信息，请参阅以下红帽知识库解决方案有关[Intel TSX 对 OpenStack 客户机的影响的指南](#)。

BZ the the2404 - 并行重启节点，同时保持仲裁会导致意外的节点关闭

对于运行 RHEL 8.4 的节点，且基于可组合角色，您必须先更新 **Database** 角色，然后才能更新任何其他角色。

BZ11427787 - undercloud 升级到 16.2 会失败，因为缺少了 swtpm 的依赖项

advanced-virt-for-rhel-8-x86_64-eus-rpms 和 **advanced-virt-for-rhel-8-x86_64-rpms** 软件仓库存在一个已知问题，可防止成功升级。要在升级前禁用这些软件仓库，请参阅 [OSP 16.2 不再需要红帽知识库解决方案 advanced-virt-for-rhel-8-x86_64-rpms](#)。

BZ the106 - tripleo_nova_libvirt 重启两次后的 podman panic

从 RHOSP 16.1 升级到 16.2 时存在一个已知问题，并从 RHOSP 16.2.1 升级到 16.2.2，与 Podman 和 libvirt 服务的更改相关。如果您没有在升级前迁移工作负载，则升级可能会失败。

BZ the9445 - 16.2.0 known issue nova_libvirt tag: 16.2.0-55.1638436404 libvirt version in Compatibility left instances in unmanageable state after minor update from RHOSP 16.2.0 to RHOSP 16.2.x

在评估对 libvirt 版本不兼容的风险之前，不要从 RHOSP 16.2.0 更新至 16.2.2 或 16.2.3。

要评估您的风险，请完成以下步骤：

1. 检查所有 Compute 节点上的 **nova_libvirt** 容器中的 libvirt 软件包：

```
$ sudo podman exec nova_libvirt rpm -qa libvirt-*
```

2. 检查 **nova_compute** 容器的 libvirt 版本：

```
$ sudo podman exec nova_compute rpm -qa libvirt-*
```

如果 libvirt 版本是 7.0，则部署不受这个程序错误的影响。您可以执行更新。

如果 libvirt 版本为 7.6，则部署会受到这个程序错误的影响。您的更新存在风险。要更新您的部署，请按照[更新 RHOSP 16.2.0 时的 libvirt 版本不兼容问题\(bug 2109350\)](#)中的内容进行操作。

BZ1142181566 - 16.2.3 上的 Overcloud 服务器显示简短的 fqdn 主机名

在 Red Hat OpenStack Platform (RHOSP) 16.2 中，引入了 **nova::dhcp_domain** 参数。如果您从 RHOSP 16.1 更新至任何 16.2 版本，并且您的自定义模板包含旧的 **nova::metadata::dhcp_domain** 参数，则与 **nova::dhcp_domain** 参数冲突。因此，主机名不会在 Compute 节点上生成。要避免这个问

题，请选择以下选项之一：

- 将旧的 `nova::metadata::dhcp_domain` 和 `nova::dhcp_domain` 参数设置为相同的值。
- 等待更新。RHOSP 16.2.6 中计划有一个修复。

[BZ#2255233 - \(OSP 16\) minor update: pacemaker_restart_bundle.sh 在聚合步骤中调用](#)

在从 16.2 升级到 16.2.4、16.2.5 或 16.2.6 的期间，Pacemaker 控制的服务在以下情况下意外重启：

- 您可以运行 `openstack overcloud update converge`，而无需更改 Pacemaker 控制的服务。
- 您不运行 `openstack overcloud update converge`，但无论您对 Pacemaker 控制的服务进行了配置更改，您都运行 `overcloud` 部署。

红帽工程团队正在调查此问题。要避免这个问题，请不要执行以下操作：

- 不要运行 `openstack overcloud update converge`。
- 不要修改 overcloud 环境。
- 不要扩展 overcloud 节点。

[BZ#2078793 - Unable 在 OSP16.1→16.2 升级后删除 LB 成员](#)

如果更新包含 OVN DB 模式升级，请不要在从 RHOSP 16.1 更新至 16.2 时更改 OVN DB 条目。这样做可能会导致错误配置和数据丢失。

如果您在更新过程中更改 OVN DB，其中包括 OVN DB 模式升级和 OpenShift、Kuryr 和负载均衡服务 (octavia)，您可能无法删除负载均衡实体。

临时解决方案： 如果您在更新过程中更改了 OVN DB，其中包括 OVN DB 模式升级和 OpenShift、Kuryr 和负载均衡服务，且无法删除负载均衡实体，请执行以下步骤：

1. 访问 mysql octavia DB。
2. 将实体的 `provisioning_status` 更改为 `DELETED`。

如果在更新过程中更改 OVN DB 后在更改 OVN DB 后对任何其他 OVN DB 实体发生，请运行 `neutron-db-sync` 工具。

流程

要为次要更新准备 RHOSP 环境，请完成以下步骤：

1. [第 1.2 节 “将环境锁定到 Red Hat Enterprise Linux 版本”](#)
2. [第 1.3 节 “切换到 TUS 软件仓库”](#)
3. [第 1.4 节 “更新 Red Hat Openstack Platform 和 Ansible 存储库”](#)
4. [第 1.5 节 “设置 container-tools 模块版本”](#)
5. [第 1.6 节 “更新容器镜像准备文件”](#)
6. [第 1.7 节 “更新 SSL/TLS 配置”](#)
7. [第 1.8 节 “在 overcloud 中禁用隔离”](#)

1.1. 长生命版本的升级路径

在开始更新或升级前，熟悉可能的更新和升级路径。



注意

您可以在 `/etc/rhosp-release` 和 `/etc/redhat-release` 文件中查看当前的 RHOSP 和 RHEL 版本。

表 1.1. 更新版本路径

当前版本	目标版本
RHOSP 10.0.x on RHEL 7.x	RHEL 7.7 latest 上的 RHOSP 10.0 latest
RHOSP 13.0.x on RHEL 7.x	RHEL 7.9 latest 上的 RHOSP 13.0 最新
RHOSP 16.1.x on RHEL 8.2	RHEL 8.2 最新上的 RHOSP 16.1 最新
RHOSP 16.1.x on RHEL 8.2	RHEL 8.4 最新上的 RHOSP 16.2 最新
RHOSP 16.2.x on RHEL 8.4	RHEL 8.4 最新上的 RHOSP 16.2 最新

表 1.2. 升级版本路径

当前版本	目标版本
RHEL 7.7 上的 RHOSP 10	RHEL 7.9 latest 上的 RHOSP 13 最新
RHOSP 13 on RHEL 7.9	RHEL 8.2 最新上的 RHOSP 16.1 最新
RHOSP 13 on RHEL 7.9	RHEL 8.4 最新上的 RHOSP 16.2 最新

如需更多信息，请参阅升级 [框架\(13 到 16.2\)](#)

红帽提供了两个选项，将您的环境升级到下一个长生命版本：

原位升级

在现有环境中执行服务升级。本指南主要侧重于此选项。

并行迁移

创建新的 Red Hat OpenStack Platform 16.2 环境，并将工作负载从您当前的环境迁移到新环境。有关 Red Hat OpenStack Platform 并行迁移的更多信息，请联系红帽全局专业服务。



重要

此表中的持续时间是基于内部测试的最小估算，可能不适用于所有生产环境。例如，如果您的硬件具有低规格或延长引导周期，则允许更长时间使用这些持续时间。要准确化每个任务的升级持续时间，请在测试环境中执行与生产环境类似的硬件的步骤。

表 1.3. 升级路径的影响和持续时间

	原位升级	并行迁移
undercloud 的升级持续时间	每个主要操作估计持续时间包括： <ul style="list-style-type: none"> ● Leapp upgrade 命令的 30 分钟 ● Leapp 重启 30 分钟 ● director 升级需要 40 分钟 	无。除了现有的 undercloud 外，您还将创建新的 undercloud。
overcloud control plane 的升级持续时间	每个 Controller 节点的估算： <ul style="list-style-type: none"> ● 60 分钟用于 Leapp 升级并重启 ● 服务升级 60 分钟 	无。除了现有的 control plane 外，您还会创建新的 control plane。
control plane 的中断持续时间	bootstrap Controller 节点的服务升级的持续时间，大约为 60 分钟。	无。这两个 overcloud 在工作负载迁移过程中都可以正常工作。
control plane 中断的结果	您不能在中断期间执行 OpenStack 操作。	没有中断。
overcloud 数据平面的升级持续时间	估算每个 Compute 节点和 Ceph Storage 节点： <ul style="list-style-type: none"> ● 60 分钟用于 Leapp 升级并重启 ● 服务升级的 30 分钟 	无。除了现有的数据平面外，您还会创建新的数据平面。
data plane 的中断持续时间	因为工作负载从节点迁移到节点，所以中断的最小。	因为从 overcloud 迁移到 overcloud 的工作负载，所以中断最小。
其他硬件要求	不需要额外的硬件。	要创建新的 undercloud 和 overcloud，需要额外的硬件。

1.2. 将环境锁定到 RED HAT ENTERPRISE LINUX 版本

Red Hat OpenStack Platform (RHOSP) 16.2 支持 Red Hat Enterprise Linux (RHEL) 8.4。在进行更新前，将 undercloud 和 overcloud 存储库锁定到 RHEL 8.4 版本，以避免将操作系统升级到较新的次版本。

流程

1. 以 **stack** 用户的身份登录 undercloud。

2. Source **stackrc** 文件：

```
$ source ~/stackrc
```

3. 编辑 overcloud 订阅管理环境文件，这是包含 **RhsmVars** 参数的文件。此文件的默认名称为 **rhsm.yml**。
4. 检查您的订阅管理配置是否包含 **rhsm_release** 参数。如果 **rhsm_release** 参数不存在，请添加它并将其设置为 8.4：

```
parameter_defaults:
  RhsmVars:
    ...
    rhsm_username: "myusername"
    rhsm_password: "p@55w0rd!"
    rhsm_org_id: "1234567"
    rhsm_pool_ids: "1a85f9223e3d5e43013e3d6e8ff506fd"
    rhsm_method: "portal"
    rhsm_release: "8.4"
```

5. 保存 overcloud 订阅管理环境文件。
6. 创建 overcloud 的静态清单文件：

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

如果您使用与 **overcloud** 默认 overcloud 名称不同的 overcloud 名称，请使用 **--plan** 选项设置 overcloud 的名称。

7. 创建一个 playbook，其中包含一个任务，将操作系统版本锁定到所有节点上的 RHEL 8.4：

```
$ cat > ~/set_release.yaml <<'EOF'
- hosts: all
  gather_facts: false
  tasks:
    - name: set release to 8.4
      command: subscription-manager release --set=8.4
      become: true
EOF
```

8. 运行 **set_release.yaml** playbook：

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

- 使用 **--limit** 选项将内容应用到所有 RHOSP 节点。将 **<undercloud>** 替换为包含这些节点的环境中的 Ansible 组。
- 如果您为这些节点使用不同的订阅，则无法针对 Ceph Storage 节点运行此 playbook。



注意

要手动将节点锁定到某个版本，请登录到节点并运行 **subscription-manager release** 命令：

```
$ sudo subscription-manager release --set=8.4
```

1.3. 切换到 TUS 软件仓库

除了标准软件仓库外，您的 Red Hat OpenStack Platform (RHOSP) 订阅还包括 Red Hat Enterprise Linux (RHEL) 8.4 延长更新支持(EUS)的软件仓库。2023 年 5 月 30 日后，您必须启用 RHEL 8.4 Telecommunications Update Service (TUS) 软件仓库进行维护支持。TUS 软件仓库包括 RHEL 8.4 的最新安全补丁和程序错误修复。

在进行更新前，将存储库切换到所需的 TUS 存储库。

表 1.4. 从 EUS 软件仓库切换到 TUS 软件仓库

EUS 软件仓库	TUS 软件仓库
rhel-8-for-x86_64-baseos-eus-rpms	rhel-8-for-x86_64-baseos-tus-rpms
rhel-8-for-x86_64-appstream-eus-rpms	rhel-8-for-x86_64-appstream-tus-rpms
rhel-8-for-x86_64-highavailability-eus-rpms	rhel-8-for-x86_64-highavailability-tus-rpms

表 1.5. 从标准软件仓库切换到 TUS 软件仓库

标准软件仓库	TUS 软件仓库
rhel-8-for-x86_64-baseos-rpms	rhel-8-for-x86_64-baseos-tus-rpms
rhel-8-for-x86_64-appstream-rpms	rhel-8-for-x86_64-appstream-tus-rpms
rhel-8-for-x86_64-highavailability-rpms	rhel-8-for-x86_64-highavailability-tus-rpms



重要

您必须使用 TUS 软件仓库来保持与特定版本的 Podman 的兼容性。Podman 的后续版本在 Red Hat OpenStack Platform 16.2 中未经测试，并可能导致意外的结果。

流程

1. 以 **stack** 用户的身份登录 undercloud。
2. Source **stackrc** 文件：

```
$ source ~/stackrc
```

3. 编辑 overcloud 订阅管理环境文件，这是包含 **RhsmVars** 参数的文件。此文件的默认名称为 **rhsm.yml**。
4. 检查您的订阅管理配置中的 **rhsm_repos** 参数。如果此参数不包含 TUS 软件仓库，请将相关的软件仓库改为 TUS 版本：

```
parameter_defaults:
  RhsmVars:
    rhsm_repos:
      - rhel-8-for-x86_64-baseos-tus-rpms
      - rhel-8-for-x86_64-appstream-tus-rpms
      - rhel-8-for-x86_64-highavailability-tus-rpms
      - ansible-2.9-for-rhel-8-x86_64-rpms
      - openstack-16.2-for-rhel-8-x86_64-rpms
      - rhceph-4-tools-for-rhel-8-x86_64-rpms
      - fast-datapath-for-rhel-8-x86_64-rpms
```

5. 保存 overcloud 订阅管理环境文件。
6. 创建 overcloud 的静态清单文件：

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

如果您使用与 **overcloud** 默认 overcloud 名称不同的 overcloud 名称，请使用 **--plan** 选项设置 overcloud 的名称。

7. 创建一个 playbook，其中包含将软件仓库设置为所有节点上的 RHEL 8.4 TUS 的任务：

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

- 如果您的环境包含标准软件仓库，请禁用以下软件仓库：
 - rhel-8-for-x86_64-baseos-rpms
 - rhel-8-for-x86_64-appstream-rpms
 - rhel-8-for-x86_64-highavailability-rpms

8. 运行 **change_tus.yaml** playbook:

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


- 使用 `--limit` 选项将内容应用到所有 Red Hat OpenStack Platform 节点。将 `<undercloud>` ; , `<Controller>` , `<Compute>` 替换为包含这些节点的环境中的 Ansible 组。
- 如果您为这些节点使用不同的订阅，则无法针对 Ceph Storage 节点运行此 playbook。

1.4. 更新 RED HAT OPENSTACK PLATFORM 和 ANSIBLE 存储库

更新您的软件仓库，以使用 Red Hat OpenStack Platform (RHOSP) 16.2 和 Ansible 2.9 软件包。

流程

1. 以 **stack** 用户的身份登录 `undercloud`。

2. Source **stackrc** 文件：

```
$ source ~/stackrc
```

3. 编辑 `overcloud` 订阅管理环境文件，这是包含 **RhsmVars** 参数的文件。此文件的默认名称为 **rhsm.yml**。
4. 检查您的订阅管理配置中的 **rhsm_repos** 参数。如果 **rhsm_repos** 参数使用 RHOSP 16.1 和 Ansible 2.8 存储库，请将存储库改为正确的版本：

```
parameter_defaults:
  RhsmVars:
    rhsm_repos:
      - rhel-8-for-x86_64-baseos-tus-rpms
      - rhel-8-for-x86_64-appstream-tus-rpms
      - rhel-8-for-x86_64-highavailability-tus-rpms
      - ansible-2.9-for-rhel-8-x86_64-rpms
      - openstack-16.2-for-rhel-8-x86_64-rpms
      - fast-datapath-for-rhel-8-x86_64-rpms
```

5. 保存 `overcloud` 订阅管理环境文件。
6. 创建 `overcloud` 的静态清单文件：

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

如果您使用与 **overcloud** 默认 `overcloud` 名称不同的 `overcloud` 名称，请使用 `--plan` 选项设置 `overcloud` 的名称。

7. 创建包含在所有 RHOSP 节点上将存储库设置为 RHOSP 16.2 的任务的 playbook：

```
$ cat > ~/update_rhosp_repos.yaml <<'EOF'
- hosts: all
  gather_facts: false
  tasks:
    - name: change osp repos
      command: subscription-manager repos --disable=openstack-16.1-for-rhel-8-x86_64-rpms
--enable=openstack-16.2-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. 运行 **update_rhosp_repos.yaml** playbook :

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

- 使用 **--limit** 选项将内容应用到所有 RHOSP 节点。将 **<undercloud>** ; , **<Controller>** , **<Compute>** 替换为包含这些节点的环境中的 Ansible 组。
- 如果您为这些节点使用不同的订阅，则无法针对 Ceph Storage 节点运行此 playbook。

9. 创建一个包含在所有 Ceph Storage 节点上将存储库设置为 RHOSP 16.2 的任务的 playbook :

```
$ 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-16.2-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. 运行 **update_ceph_repos.yaml** playbook :

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

使用 **--limit** 选项将内容应用到 Ceph Storage 节点。

1.5. 设置 CONTAINER-TOOLS 模块版本

将 **container-tools** 模块设置为 **3.0** 版本，以确保在所有节点上使用正确的软件包版本。

流程

1. 以 **stack** 用户的身份登录 undercloud。
2. Source **stackrc** 文件 :

```
$ source ~/stackrc
```

3. 创建 overcloud 的静态清单文件 :

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

如果您使用与 **overcloud** 默认 overcloud 名称不同的 overcloud 名称，请使用 **--plan** 选项设置 overcloud 的名称。

4. 创建一个 playbook，其中包含一个任务，将 **container-tools** 模块设置为所有节点上的 **3.0** 版本：

```
$ cat > ~/container-tools.yaml <<'EOF'
- hosts: all
  gather_facts: false
  tasks:
    - name: disable default dnf module for container-tools
      command: dnf module reset -y container-tools
      become: true
    - name: set dnf module for container-tools:3.0
      command: dnf module enable -y container-tools:3.0
      become: true
EOF
```

5. 针对所有节点运行 **container-tools.yaml** playbook：

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

1.6. 更新容器镜像准备文件

容器准备文件是包含 **ContainerImagePrepare** 参数的文件。您可以使用此文件为 undercloud 和 overcloud 获取容器镜像的规则。

在更新环境前，请检查该文件以确保获取正确的镜像版本。

流程

1. 编辑容器准备文件。此文件的默认名称为 **containers-prepare-parameter.yaml**。
2. 检查每个规则集的 **tag** 参数设置为 **16.2**：

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



注意

如果您不想将特定的标签用于更新，如 **16.2** 或 **16.2.2**，请删除 **tag** 键-值对，并只指定 **tag_from_label**。这使用安装的 Red Hat OpenStack Platform 版本来确定要用作更新过程一部分的标签值。

3. 保存这个文件。

1.7. 更新 SSL/TLS 配置

从 **resource_registry** 中删除 **NodeTLSData** 资源，以更新 SSL/TLS 配置。

流程

1. 以 **stack** 用户的身份登录 undercloud。

2. Source **stackrc** 文件：

```
$ source ~/stackrc
```

3. 编辑自定义 overcloud SSL/TLS 公共端点文件，该文件通常命名为 **~/templates/enable-tls.yaml**。

4. 从 **resource_registry** 中删除 **NodeTLSData** 资源：

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

overcloud 部署使用 HAProxy 中的新服务来确定是否启用了 SSL/TLS。



注意

如果这是 **enable-tls.yaml** 文件的 **resource_registry** 部分中的唯一资源，请删除完整的 **resource_registry** 部分。

5. 保存 SSL/TLS 公共端点文件。
6. 如果您要从 Red Hat OpenStack Platform 16.1 更新，您必须更新 Red Hat Identity Manager (IdM) 中的权限，以便所有预更新检查都通过。使用 **ssh** 登录到运行 IdM 的服务器，然后运行以下命令：

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

1.8. 在 OVERCLOUD 中禁用隔离

在更新 overcloud 之前，请确保禁用隔离。

如果在 Controller 节点更新过程中在环境中部署隔离功能，overcloud 可能会检测到某些节点被禁用并尝试隔离操作，这可能导致意外的结果。

如果您在 overcloud 中启用了隔离，则必须在更新期间临时禁用隔离，以避免任何意外的结果。

流程

1. 以 **stack** 用户的身份登录 undercloud。

2. source **stackrc** 文件：

```
$ source ~/stackrc
```

3. 登录到 Controller 节点并运行 Pacemaker 命令来禁用隔离：

■

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

将 **<controller_ip>** 替换为 Controller 节点的 IP 地址。您可以使用 **openstack server list** 命令查找 Controller 节点的 IP 地址。

4. 在 **fence.yaml** 环境文件中，将 **EnableFencing** 参数设置为 **false**，以确保隔离在更新过程中保持禁用。

其它资源

- [使用 STONITH 的隔离 Controller 节点](#)

第 2 章 更新 UNDERCLOUD

您可以使用 director 更新 undercloud 节点上的主软件包。要将 undercloud 及其 overcloud 镜像更新至最新的 Red Hat OpenStack Platform (RHOSP) 16.2 版本，请完成以下步骤：

1. [第 2.1 节 “对容器化 undercloud 执行次要更新”](#)
2. [第 2.2 节 “更新 overcloud 镜像”](#)

先决条件

- 在将 undercloud 更新至最新的 RHOSP 16.2 版本前，请确保完成所有更新准备步骤。更多信息请参阅 [第 1 章 准备次版本更新](#)。

2.1. 对容器化 UNDERCLOUD 执行次要更新

director 提供更新 undercloud 节点上主软件包的命令。使用 director 在 RHOSP 环境当前版本中执行次要更新。

流程

1. 在 undercloud 节点上，以 **stack** 用户身份登录。
2. Source **stackrc** 文件：

```
$ source ~/stackrc
```

3. 使用 **dnf update** 命令更新 director 主软件包：

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

4. 使用 **openstack undercloud upgrade** 命令更新 undercloud 环境：

```
$ openstack undercloud upgrade
```

5. 等待 undercloud 更新过程完成。
6. 重启 undercloud 以更新操作系统的内核和其他系统软件包：

```
$ sudo reboot
```

7. 稍等片刻，直到节点启动。

2.2. 更新 OVERCLOUD 镜像

您必须将您当前的 overcloud 镜像替换为新版本，以确保 director 可以使用最新版本的 RHOSP 软件内省和置备节点。如果使用预置备节点，则不需要这一步。

先决条件

- 您已将 undercloud 节点更新至最新版本。更多信息请参阅 [第 2.1 节 “对容器化 undercloud 执行次要更新”](#)。

流程

1. Source **stackrc** 文件：

```
$ source ~/stackrc
```

2. 从 **stack** 用户的主目录（**/home/stack/images**）的 **images** 中删除任何存在的镜像。

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

3. 提取存档：

```
$ cd ~/images
$ for i in /usr/share/rhosp-director-images/overcloud-full-latest-16.2.tar /usr/share/rhosp-director-images/ironic-python-agent-latest-16.2.tar; do tar -xvf $i; done
$ cd ~
```

4. 将最新的镜像导入到 director：

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

5. 将您的节点配置为使用新镜像：

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

6. 验证新镜像是否存在：

```
$ openstack image list
$ ls -l /var/lib/ironic/httpboot
```

重要

- 部署 overcloud 节点时，请确保 overcloud 镜像版本对应于对应的 heat 模板版本。例如，只使用 RHOSP 16.2 heat 模板的 RHOSP 16.2 镜像。
- 如果您部署了使用红帽客户门户网站或 Red Hat Satellite Server 的连接环境，则 overcloud 镜像和软件包存储库版本可能会不同步。为确保 overcloud 镜像和软件包存储库版本匹配，您可以使用 **virt-customize** 工具。有关更多信息，请参阅红帽知识库解决方案，[使用 virt-customize 修改 Red Hat Linux OpenStack Platform Overcloud 镜像](#)。
- 新的 **overcloud-full** 镜像替换旧的 **overcloud-full** 镜像。如果对旧镜像进行了更改，则必须在新镜像中重复更改，特别是在将来部署新节点时。

第 3 章 更新 OVERCLOUD

更新 undercloud 后，您可以通过运行 overcloud 和容器镜像准备命令、更新节点并运行 **overcloud update converge** 命令来更新 overcloud。control plane API 在次版本更新过程中完全可用。

先决条件

- 您已将 undercloud 节点更新至最新版本。更多信息请参阅 [第 2 章 更新 undercloud](#)。
- 如果您在 **stack** 用户主目录中使用一组本地的核心模板，请确保更新模板并使用高级 *Overcloud 自定义指南中的使用自定义核心 Heat 模板中的推荐工作流*。在升级 overcloud 前，您必须更新本地副本。

流程

要更新 overcloud，您必须完成以下步骤：

1. [第 3.1 节 “运行 overcloud 更新准备”](#)
2. [第 3.2 节 “运行容器镜像准备”](#)
3. [第 3.3 节 “可选：在所有 overcloud 服务器上更新 ovn-controller 容器”](#)
4. [第 3.4 节 “更新所有 Controller 节点”](#)
5. [第 3.5 节 “更新所有 Compute 节点”](#)
6. [第 3.6 节 “更新所有 HCI Compute 节点”](#)
7. [第 3.8 节 “更新所有 Ceph Storage 节点”](#)
8. [第 3.9 节 “执行在线数据库更新”](#)
9. [第 3.10 节 “最终更新”](#)

3.1. 运行 OVERCLOUD 更新准备

要为更新过程准备 overcloud，您必须运行 **openstack overcloud update prepare** 命令，该命令将 overcloud 计划更新至 Red Hat OpenStack Platform (RHOSP) 16.2，并为更新准备节点。

先决条件

- 如果使用 Ceph 订阅并将 director 配置为使用 Ceph 存储节点的 **overcloud-minimal** 镜像，则必须在 **roles_data.yaml** 角色定义文件中确保 **rism_enforce** 参数设置为 **False**。
- 如果呈现自定义 NIC 模板，则必须使用 **openstack-tripleo-heat-templates** 集合的更新版本重新生成模板，以避免与 overcloud 版本不兼容。有关自定义 NIC 模板的更多信息，[请参阅高级 Overcloud 自定义指南中的对默认网络接口模板进行定制](#)。



注意

对于使用 OVN 部署的分布式计算节点（边缘）架构，您必须为带有 Compute、distributedCompute 或 DistributedComputeHCI 节点的每个堆栈完成此流程，然后才能继续 [在所有 overcloud 服务器上更新 ovn-controller 容器](#)。

流程

1. Source **stackrc** 文件：

```
$ source ~/stackrc
```

2. 运行 update 准备命令：

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

包括与您的环境相关的以下选项：

- 如果 **overcloud** 堆栈的名称与默认名称 **overcloud** 不同，请在更新准备命令中包含 **--stack** 选项，并将 **<stack_name>** 替换为您的堆栈的名称。
- 如果您使用自己的自定义角色，请包含自定义角色(**<roles_data>**)文件(**-r**)。
- 如果使用自定义网络，请包含您的可组合网络(**<network_data>**)文件(**-n**)。
- 如果您部署高可用性集群，请在 **update prepare** 命令中包含 **--ntp-server** 选项，或者在环境文件中包含 **NtpServer** 参数和值。
- 任何自定义配置环境文件(**-e**)。

3. 等待更新准备过程完成。

3.2. 运行容器镜像准备

在更新 **overcloud** 前，您必须准备环境所需的所有容器镜像配置，并将最新的 RHOSP 16.2 容器镜像拉取到 **undercloud**。

要完成容器镜像准备，您必须针对具有 **container_image_prepare** 标签的任务运行 **openstack overcloud external-update run** 命令。



注意

如果您不使用默认的堆栈名称（即 **overcloud**），请将堆栈名称设置为 **--stack <stack_name>** 选项，并将 **<stack_name>** 替换为堆栈的名称。

流程

1. Source **stackrc** 文件：

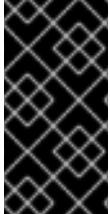
```
$ source ~/stackrc
```

2. 针对具有 **container_image_prepare** 标签的任务运行 **openstack overcloud external-update run** 命令：

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

3.3. 可选：在所有 OVERCLOUD 服务器上更新 OVN-CONTROLLER 容器

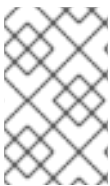
如果您使用 Modular Layer 2 Open Virtual Network 机制驱动程序(ML2/OVN)部署 overcloud，请将 ovn-controller 容器更新至最新的 RHOSP 16.2 版本。该更新会在运行 ovn-controller 容器的每个 overcloud 服务器上运行。



重要

以下流程在被分配了 Controller 角色的服务器上更新被分配 Compute 角色的服务器上的 ovn-controller 容器。

如果您在遵循此流程前意外更新 ovn-northd 服务，则可能无法访问虚拟机或创建新虚拟机或虚拟网络。以下流程恢复连接。



注意

对于分布式计算节点（边缘）架构，您必须为带有 Compute、distributedCompute 或 DistributedComputeHCI 节点的每个堆栈完成此流程，然后才能继续 [更新所有 Controller 节点](#)。

流程

1. 以 **stack** 用户的身份登录 undercloud。

2. Source **stackrc** 文件：

```
$ source ~/stackrc
```

3. 针对具有 ovn 标签的任务运行 openstack overcloud external-update run 命令：

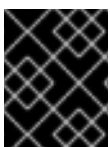
```
$ openstack overcloud external-update run --stack <stack_name> --tags ovn
```

- 如果 overcloud 堆栈的名称与默认堆栈名称 overcloud 不同，请将堆栈名称设置为 **--stack** 选项，并将 **<stack_name>** 替换为您的堆栈的名称。

4. 等待 ovn-controller 容器更新完成。

3.4. 更新所有 CONTROLLER 节点

将所有 Controller 节点更新至最新的 RHOSP 16.2 版本。运行 **openstack overcloud update run** 命令，并包含 **--limit Controller** 选项，以仅限对 Controller 节点的操作。control plane API 在次版本更新过程中完全可用。



重要

在解决 [BZ#1872404](#) 之前，对于基于可组合角色的节点，您必须先更新 **Database** 角色，然后再更新 **Controller, Messaging, Compute, Ceph** 和其他角色。



注意

如果您不使用默认的堆栈名称（即 **overcloud**），请将堆栈名称设置为 **--stack <stack_name>** 选项，并将 **<stack_name>** 替换为堆栈的名称。

流程

1. Source **stackrc** 文件：

```
$ source ~/stackrc
```

2. 运行 **update** 命令：

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

3. 等待 Controller 节点更新完成。

3.5. 更新所有 COMPUTE 节点

将所有 Compute 节点更新至最新的 RHOSP 16.2 版本。若要更新 Compute 节点，请运行 **openstack overcloud update run** 命令，并包含 **--limit Compute** 选项，以仅限制对 Compute 节点的操作。

并行化注意事项

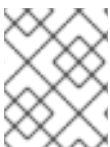
当您更新大量 Compute 节点时，要提高性能，您可以在后台运行多个更新任务，并配置每个任务来更新单独的 20 个节点组。例如，如果您的部署中有 80 个 Compute 节点，您可以运行以下命令来并行更新 Compute 节点：

```
$ 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 &
```

这种分区节点空间的方法是随机的，您没有控制要更新哪些节点。节点的选择基于在运行 **tripleo-ansible-inventory** 命令时生成的清单文件。

要更新特定的 Compute 节点，请列出您要在用逗号分开的批处理中更新的节点：

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



注意

如果您不使用默认的堆栈名称（即 **overcloud**），请将堆栈名称设置为 **--stack <stack_name>** 选项，并将 **<stack_name>** 替换为堆栈的名称。

流程

1. Source **stackrc** 文件：

```
$ source ~/stackrc
```

2. 运行 **update** 命令：

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

3. 等待 Compute 节点更新完成。

3.6. 更新所有 HCI COMPUTE 节点

将 Hyperconverged Infrastructure (HCI) Compute 节点更新至最新的 RHOSP 16.2 版本。要更新 HCI Compute 节点，请运行 **openstack overcloud update run** 命令，并包含 **--limit ComputeHCI** 选项来将操作限制为 HCI 节点。您还必须运行 **openstack overcloud external-update run --tags ceph** 命令，以对容器化 Red Hat Ceph Storage 4 集群执行更新。

先决条件

- 在运行 **ceph-mon** 服务的 Ceph Monitor 或 Controller 节点上，检查 Red Hat Ceph Storage 集群状态是否健康，pg 状态为 **active+clean**：

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

如果 Ceph 集群处于健康状态，它会返回 **HEALTH_OK** 状态。

如果 Ceph 集群状态不健康，它会返回 **HEALTH_WARN** 或 **HEALTH_ERR** 的状态。有关故障排除指南，请参阅 [Red Hat Ceph Storage 4 故障排除指南](#)。

流程

1. Source **stackrc** 文件：

```
$ source ~/stackrc
```

2. 运行 update 命令：

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

- 将 **<stack_name>** 替换为堆栈的名称。如果没有指定，则默认为 **overcloud**。

3. 等待节点更新完成。

4. 运行 Ceph Storage update 命令：

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

5. 等待 Compute HCI 节点更新完成。

3.7. 更新所有 DISTRIBUTEDCOMPUTEHCI 节点

更新特定于分布式计算节点架构的角色。当您升级分布式计算节点时，请先更新 **DistributedComputeHCI** 节点，然后更新 **DistributedComputeHCIScaleOut** 节点。



注意

如果您不使用默认的堆栈名称（即 **overcloud**），请将堆栈名称设置为 **--stack <stack_name>** 选项，并将 **<stack_name>** 替换为堆栈的名称。

先决条件

- 在运行 **ceph-mon** 服务的 Ceph Monitor 或 Controller 节点上，检查 Red Hat Ceph Storage 集群状态是否健康，pg 状态为 **active+clean**：

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

如果 Ceph 集群处于健康状态，它会返回 **HEALTH_OK** 状态。

如果 Ceph 集群状态不健康，它会返回 **HEALTH_WARN** 或 **HEALTH_ERR** 的状态。有关故障排除指南，请参阅 [Red Hat Ceph Storage 4 故障排除指南](#)。

流程

1. Source **stackrc** 文件：

```
$ source ~/stackrc
```

2. 运行 update 命令：

```
$ openstack overcloud update run --stack <stack_name> --limit DistributedComputeHCI
```

3. 等待 **DistributedComputeHCI** 节点更新完成。

4. 运行 Ceph Storage update 命令：

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

5. 等待 **DistributedComputeHCI** 节点更新完成。

6. 使用相同的进程来更新 **DistributedComputeHCIScaleOut** 节点。

3.8. 更新所有 CEPH STORAGE 节点

将 Red Hat Ceph Storage 节点更新至最新的 RHOSP 16.2 版本。



重要

RHEL 8.4 支持 RHOSP 16.2。但是，映射到 Ceph Storage 角色的主机会更新到最新的主 RHEL 版本。有关更多信息，请参阅 [Red Hat Ceph Storage: 支持的配置](#)。

先决条件

- 在运行 **ceph-mon** 服务的 Ceph Monitor 或 Controller 节点上，检查 Red Hat Ceph Storage 集群状态是否健康，pg 状态为 **active+clean**：

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

如果 Ceph 集群处于健康状态，它会返回 **HEALTH_OK** 状态。

如果 Ceph 集群状态不健康，它会返回 **HEALTH_WARN** 或 **HEALTH_ERR** 的状态。有关故障排除指南，请参阅 [Red Hat Ceph Storage 4 故障排除指南](#)。

流程

步骤

1. Source **stackrc** 文件：

```
$ source ~/stackrc
```

2. 更新组节点。
更新组中的所有节点：

```
$ openstack overcloud update run --limit <GROUP_NAME>
```

更新组中的单一节点：

```
$ openstack overcloud update run --limit <GROUP_NAME> [NODE_INDEX]
```

**注意**

如果选择单独更新节点，请确保更新所有节点。

组中的第一个节点的索引为零(0)。例如，要更新名为 **CephStorage** 的组中的第一个节点：

```
openstack overcloud update run --limit CephStorage[0]
```

3. 等待节点更新完成。
4. 运行 Ceph Storage 容器更新命令，以外部进程运行 **ceph-ansible**，并更新 Red Hat Ceph Storage 4 容器：

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

5. 等待 Ceph Storage 容器更新完成。

3.9. 执行在线数据库更新

有些 overcloud 组件需要在线更新或迁移其数据库表。要执行在线数据库更新，请针对具有 **online_upgrade** 标签的任务运行 **openstack overcloud external-update run** 命令。

在线数据库更新适用于以下组件：

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

流程

1. Source **stackrc** 文件：

```
$ source ~/stackrc
```

2. 针对使用 **online_upgrade** 标签的任务运行 **openstack overcloud external-update run** 命令：

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

3.10. 最终更新



注意

运行 **openstack overcloud update converge** 命令不再需要。但是，如果您禁用了隔离并计划跳过聚合步骤，您必须手动重新启用隔离。

您可以将 overcloud 堆栈更新至最新的 RHOSP 16.2 版本。这样可确保堆栈资源结构与 OSP 16.2 标准部署一致，您可以在以后执行常规的 **openstack overcloud deploy** 功能。

流程

1. 以 **stack** 用户的身份登录 undercloud。

2. Source **stackrc** 文件：

```
$ source ~/stackrc
```

3. 如果禁用了隔离，并且您不运行 **openstack overcloud update converge**，则必须重新启用隔离：

- a. 登录到 Controller 节点并运行 Pacemaker 命令重新启用隔离：

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

- 将 **<controller_ip>** 替换为 Controller 节点的 IP 地址。您可以使用 **openstack server list** 命令查找 Controller 节点的 IP 地址。

- b. 在 **fencing.yaml** 环境文件中，将 **EnableFencing** 参数的值设置为 **true**。

4. 可选：运行 update finalization 命令：

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

包括与您环境相关的以下选项：

- **fence.yaml** 环境文件，并将 **EnableFencing** 参数设置为 **true**。
- 如果 **overcloud** 堆栈的名称与默认名称 **overcloud** 不同，请在更新准备命令中包含 **--stack** 选项，并将 **<stack_name>** 替换为您的堆栈的名称。
- 如果使用自定义角色，请包含自定义角色(**<roles_data>**)文件(**-r**)。
- 如果使用自定义网络，请包含您的可组合网络(**<network_data>**)文件(**-n**)。
- 任何自定义配置环境文件(**-e**)。

等待更新最终完成。

第 4 章 重新引导 OVERCLOUD

在将一个次版本 Red Hat OpenStack Platform (RHOSP)更新至最新的 16.2 版本后，重启 overcloud。重启会使用任何关联的内核、系统级和容器组件更新刷新节点。这些更新提供了性能和安全优势。计划停机时间来执行重启过程。

使用以下指导了解如何重新引导不同的节点类型：

- 如果重新引导一个角色中的所有节点，请单独重新引导每个节点。如果您同时重新引导角色中的所有节点，在重新引导操作过程中可能会发生服务停机。
- 按照以下顺序在节点上完成重启步骤：
 1. [第 4.1 节 “重新引导 Controller 和可组合节点”](#)
 2. [第 4.2 节 “重新引导 Ceph Storage \(OSD\) 集群”](#)
 3. [第 4.3 节 “重新引导 Compute 节点”](#)

4.1. 重新引导 CONTROLLER 和可组合节点

根据可组合角色重新引导 Controller 节点和独立节点，并排除 Compute 节点和 Ceph Storage 节点。

流程

1. 登录您要重新引导的节点。
2. 可选：如果节点使用 Pacemaker 资源，请停止集群：

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

3. 重新引导节点：

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

4. 稍等片刻，直到节点启动。

verification

1. 验证服务是否已启用。
 - a. 如果该节点使用 Pacemaker 服务，请检查该节点是否已重新加入集群：

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

- b. 如果该节点使用 Systemd 服务，请检查是否所有服务都已启用：

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

- c. 如果该节点使用容器化服务，则检查节点上的所有容器是否已激活：

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

4.2. 重新引导 CEPH STORAGE (OSD) 集群

完成以下步骤以重新引导 Ceph Storage (OSD) 节点集群。

先决条件

- 在运行 **ceph-mon** 服务的 Ceph Monitor 或 Controller 节点上，检查 Red Hat Ceph Storage 集群状态是否健康，pg 状态为 **active+clean**：

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

如果 Ceph 集群处于健康状态，它会返回 **HEALTH_OK** 状态。

如果 Ceph 集群状态不健康，它会返回 **HEALTH_WARN** 或 **HEALTH_ERR** 的状态。有关故障排除指南，请参阅 [Red Hat Ceph Storage 4 故障排除指南](#)。

步骤

1. 登录到运行 **ceph-mon** 服务的 Ceph Monitor 或 Controller 节点，并临时禁用 Ceph Storage 集群重新平衡：

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



注意

如果您有多堆栈或分布式计算节点(DCN)架构，您必须在设置 **noout** 和 **norebalance** 标志时指定集群名称。例如：**sudo podman exec -it ceph-mon-controller-0 ceph osd set noout --cluster <cluster_name>**

2. 选择第一个要重新引导的 Ceph Storage 节点并登录到该节点。

3. 重新引导节点：

```
$ sudo reboot
```

4. 稍等片刻，直到节点启动。

5. 登录到节点，并检查集群的状态：

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

确认 **pgmap** 报告的所有 **pgs** 的状态是否都正常 (**active+clean**)。

6. 注销节点，重新引导下一个节点，并检查其状态。重复此过程，直到您已重新引导所有 Ceph Storage 节点。

7. 完成后，登录到运行 **ceph-mon** 服务的 Ceph Monitor 或 Controller 节点，然后重新启用集群重新平衡：

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



注意

如果您有多堆栈或分布式计算节点(DCN)架构，您必须在取消设置 **noout** 和 **norebalance** 标志时指定集群名称。例如：**sudo podman exec -it ceph-mon-controller-0 ceph osd set noout --cluster <cluster_name>**

8. 执行最后的状态检查，确认集群报告 **HEALTH_OK**：

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

4.3. 重新引导 COMPUTE 节点

为确保 Red Hat OpenStack Platform 环境中实例的最小停机时间，[迁移实例 workflow](#) 概述了您要从您要重新引导的 Compute 节点迁移实例所需的步骤。



注意

如果您没有将实例从源 Compute 节点迁移到另一个 Compute 节点，则实例可能会在源 Compute 节点上重启，这可能会导致升级失败。这与 Podman 和 libvirt 服务更改的已知问题相关：

- [BZ the106 - tripleo_nova_libvirt 重启两次后的 podman panic](#)
- [BZ11410135 - tripleo_nova_libvirt 重启两次后的 podman panic](#)

迁移实例 workflow

1. 决定是否在重新引导节点前将实例迁移到另一个 Compute 节点。
2. 选择并禁用您要重新引导的 Compute 节点，使其不调配新实例。
3. 将实例迁移到另一个 Compute 节点中。
4. 重新引导空的 Compute 节点。
5. 启用空的 Compute 节点。

先决条件

- 重启 Compute 节点之前，必须决定是否在节点重启过程中将实例迁移到另一个 Compute 节点。查看在 Compute 节点之间迁移虚拟机实例时可能会遇到的迁移约束列表。如需更多信息，请参阅[为实例创建配置 Compute Service](#) 中的[迁移限制](#)。
- 如果您无法迁移实例，则可设置以下核心模板参数以在 Compute 节点重启后控制实例的状态：

NovaResumeGuestsStateOnHostBoot

确定重新引导后是否将实例返回 Compute 节点上的相同状态。设为 **False** 时，实例保持关闭，必须手动启动。默认值为 **False**。

NovaResumeGuestsShutdownTimeout

重启前等待实例被关闭的时间（以秒为单位）。建议不要将此值设置为 **0**。默认值为 **300**。有关 overcloud 参数及其用法的更多信息，请参阅[Overcloud 参数](#)。

流程

1. 以 **stack** 用户的身份登录 undercloud。

2. 列出所有的 Compute 节点及其 UUID：

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

识别您要重新引导的 Compute 节点的 UUID。

3. 从 undercloud 中，选择一个 Compute 节点并禁用它：

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

4. 列出 Compute 节点上的所有实例：

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

5. 可选：如果您决定将实例迁移到另一个 Compute 节点，请完成以下步骤：

a. 如果您决定将实例迁移至另一个 Compute 节点，则使用以下命令之一：

o 要将实例迁移到其他主机，请运行以下命令：

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

o 让 **nova-scheduler** 自动选择目标主机：

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

o 一次性实时迁移所有实例：

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



注意

nova 命令可能会引发一些弃用警告，这些警告信息可以被安全忽略。

b. 稍等片刻，直至迁移完成。

c. 确认迁移成功完成：

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

d. 继续迁移实例，直到 Compute 节点上不剩任何实例。

6. 登录到 Compute 节点并重新引导节点：

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

7. 稍等片刻，直到节点启动。

8. 重新启用 Compute 节点：

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

9. 确认是否已启用 Compute 节点：

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